ISAC: An Invertible and Stable Auditory Filter Bank with Customizable Kernels for ML Integration

Daniel Haider*, Felix Perfler*, Peter Balazs*‡, Clara Hollomey†, and Nicki Holighaus*
*Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria
†University of Applied Sciences St. Pölten, St. Pölten, Austria
‡ Interdisciplinary Transformation University Austria (IT:U), Linz, Austria
daniel.haider@oeaw.ac.at

***Abstract*—This paper introduces ISAC, an invertible and stable, perceptually-motivated filter bank that is specifically designed to be integrated into machine learning paradigms. More precisely, the center frequencies and bandwidths of the filters are chosen to follow a non-linear, auditory frequency scale, the filter kernels have user-defined maximum temporal support and may serve as learnable convolutional kernels, and there exists a corresponding filter bank such that both form a perfect reconstruction pair. ISAC provides a powerful and user-friendly audio front-end suitable for any application, including analysis-synthesis schemes.**

***Index Terms*—filter bank learning, invertibility, stability, convolutional neural networks**

I. Introduction

Using learnable time-frequency filter banks as front-ends for neural networks can considerably increase the computational efficiency of a wide range of audio processing tasks [6]. Related approaches range from fully learnable convolutional layers with 1D kernels [1], [17], [24] to parametrized constructions where only the bandwidths and center frequencies of fixed prototype kernels are learned [22], [28]. Prior work indicates that optimizing learnable filter bank front-ends benefits from initialization as perceptually-motivated time-frequency filters, e.g., based on the mel frequency scale [29], and a wide range of related transforms has been used [12], [23], [27].

Although invertibility and numerical stability of the underlying time-frequency filter bank is of limited concern for classification tasks, it is crucial in analysis-synthesis schemes [4], [15], [17] to prevent the introduction of errors and ensure that no needed information is lost. Moreover, it has been found that stability in the form of a 1-Lipschitz or even Parseval property significantly increases robustness of the model against noise and adversarial examples [2], [9], [16], [20].

Perceptually-motivated filter banks that are perfectly invertible and numerically stabile exist [10], [19]. In particular, these constructions employ filters with bandwidths matched to the *critical bands* [31] of human hearing. However, the frequency-domain implementation in [10], [19] does not provide control of the kernel size[1], even favoring bandlimited filters, which necessarily have kernels of infinite length. Yet, control of the kernel size is a requirement for the efficient integration of any time-frequency filter bank into learnable machine learning frameworks to restrict the number of learnable parameters and employ linear convolution directly on the GPU [6], [7].

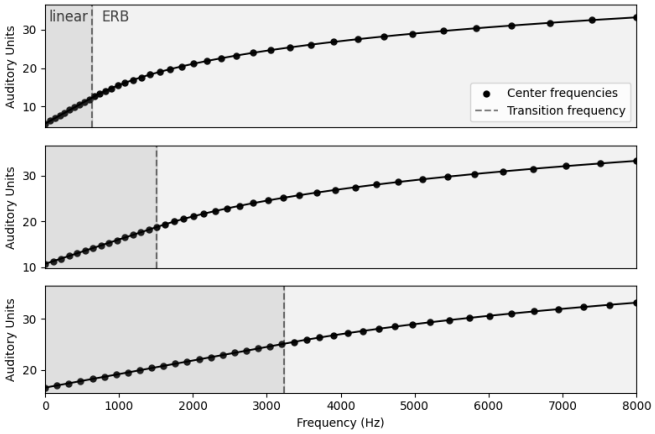

Fig. 1: Frequency scales of ISAC filter banks for kernel sizes 256, 128, and 64 (top to bottom), based on the ERB scale. The center frequencies in the dark area lie on a linear scale, and those in the light area on the ERB scale. The shorter the kernels, the larger the linear part of the scale becomes.

We fill this gap by proposing ISAC, a perceptually-motivated time-frequency filter bank that is invertible and numerically stable. While ISAC is based on the construction proposed in [19], we introduce some key changes to enhance user-friendliness and facilitate an efficient integration into convolutional machine learning frameworks. On demand, the kernels of ISAC can be further optimized together with the remaining model parameters, similar to [29]. Stability and invertibility can be maintained throughout the whole training

D. Haider and F. Perfler are recipients of a DOC Fellowship (A 26355 and A 27288, respectively) of the Austrian Academy of Sciences at the Acoustics Research Institute. The work of N. Holighaus was supported by the FWF project DISCO (PAT4780023). The work of P. Balazs was supported by the FWF projects LoFT (P 34624), NoMASP (P 34922), Voice Prints (P 36446) and the WWTF project EleCom (LS23-024). C. Hollomey has been funded by the Vienna Science and Technology Fund (WWTF) and by the State of Lower Austria [Grant ID: 10.47379/LS23024]. ISAC was derived from the implementation of `audfilters` of the Large Time-Frequency Analysis Toolbox (LTFAT) (https://ltfat.org).

[1]In the signal processing literature, it is common to refer to the convolution kernel as *impulse response*. Here, we prefer the term convolution kernel, which is widely understood in the mathematics and machine learning communities.

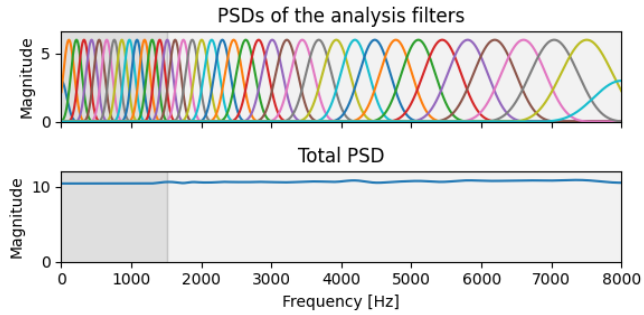

Fig. 2: Individual and total power spectral densities of an ISAC filter bank with 40 kernels of size $T^* = 128$. The flat total PSD indicates that the filter bank is well conditioned. The condition number for a decimation factor of 6 is $\kappa = 1.05$.

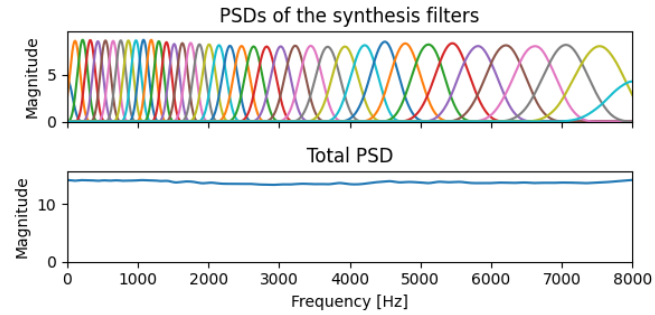

Fig. 3: Individual and total power spectral densities of learned synthesis kernels for the setting of Fig. 2. The reconstruction error is 6e-6. A regularizing term ensured that the total PSD stays flat. The condition number is $\kappa = 1.07$.

process by including a regularization term in the learning objective [2], [9], [15]. Moreover, we can compute structurally identical approximate duals, i.e., having the same kernel sizes.

The central novelty in ISAC is a user-defined maximal kernel size. In order to achieve this while maintaining bandwidth requirements and a flat total power spectral density (PSD), we linearize the auditory scale below the frequency at which the desired kernel size is first achieved. We thus obtain a set of filters with short kernels that can be used for time-frequency processing in various applications. Our implementation supports the seamless integration into machine learning frameworks and, similar to the idea in [28], it is also possible to extend the filter bank to a compressed mel-type spectrogram.

II. Filter Banks and Convolutional Layers

Filter banks and *(linear) convolutional layers* are two names for the same class of linear, time-invariant (LTI) operators, possibly combined with decimation. They are well-suited for processing data that exhibits translation invariance. In practice, however, the two terms hint at different use cases and implied restrictions to the considered LTI operator. *Time-frequency transforms* are designed as filter banks to match specific time-frequency characteristics, with kernel sizes typically ranging from 10 to about 100 ms (corresponding to 48 to 4800 samples for a sampling rate $f_s = 48$ kHz) for most audio processing tasks. Learned convolutional layers, on the other hand, usually employ much shorter kernels for processing audio input. As an example, the default kernel size in the learnable encoder of ConvTasNet [17] is 16, and wav2vec [1] as well as WavLM [5] use kernel lengths between 2 and 10 samples to represent the input audio. Reasons for this comprise the decreased computational demands along with empirically observed performance improvements associated with employing shorter kernels [26], [30]. Moreover, long filters are prone to numerical instabilities [14] and may increase the risk of overfitting when training data is limited. To capture dependencies over a longer time range with short kernels, the models include different dimensionality reduction schemes and are usually designed deeper, i.e., multiple convolutional layers with short kernels are composed. Hence, integrating a learnable filter bank into this paradigm, necessitates to allow a customization of the kernel sizes to a definable maximum, similar to how we set them for time-frequency filter banks in practice.

In mathematical terms, for a finite discrete real-valued audio signal $x \in \mathbb{R}^L$ and a filter $g \in \mathbb{C}^L$ (specified by its kernel), filtering x by g equals finite circular convolution of x and g (i.e., indices are understood $\bmod L$) and decimate the output by a factor $d \geq 1$ (a.k.a. stride), classically written as

$$(x * g) \downarrow_d [n] = \sum_{k=0}^{L-1} x[k] g[dn - k]. \tag{1}$$

Assuming that a filter has kernel size $T \ll L$, by commutativity of the convolution, we may rewrite the above as

$$(x * g) \downarrow_d [n] = \sum_{k=0}^{T-1} g[k] x[dn - k]. \tag{2}$$

While (1) is how convolution is formally defined, the reduced version in (2) is how convolution (with circulant boundary conditions) is applied in a convolutional layer, where the T kernel coefficients are the trainable parameters of the network.

III. The ISAC Filter Bank Construction

The construction of the ISAC filters closely follows the one for AUDlets [19]. We recall the main idea: Given a (non-linear) auditory scale function $F_S : [0, \frac{f_s}{2}] \to S$ that maps positive frequencies (in Hz) to an auditory scale S (in auditory units), and a function $B_S : [0, \frac{f_s}{2}] \to \mathbb{R}$ that gives the associated bandwidths. For the commonly used equivalent rectangular bandwidth (ERB) auditory scale [13], [18], the functions are given by

$$F_{\text{ERB}}(f) = 9.265 \ln\left(1 + \frac{f}{228.8455}\right), \tag{3}$$

and

$$B_{\text{ERB}}(f) = 24.7 + \frac{f}{9.265}. \tag{4}$$

If there is no associated bandwidth function (e.g., for the mel scale), setting $B_S(f) = (\partial(F_S^{-1})/\partial f)(F_S(f))$ provides appropriate frequency overlap of the filters.

The AUDlet filters h_k are constructed in the Fourier domain by placing a (symmetric) prototype kernel w at desired center frequencies f_k, $k = 1, \ldots, K$ and impose the bandwidths $B_S(f_k)$ (with an optional factor $\gamma > 0$) on the chosen auditory scale S, yielding

$$\hat{h}_k(f) = \sqrt{d_k} \cdot w\left(\frac{f - f_k}{\gamma \cdot B_S(f_k)}\right). \tag{5}$$

The center frequencies f_k are spaced equidistantly on the image of $[f_{\min}, f_{\max}]$ under F_S and the decimation factors d_k are chosen to yield sufficient overlap at low to moderate redundancy, aiming at stability and invertibility. If necessary, a low- and a high-pass filter are added to cover the regions $[0, f_{\min})$ and $(f_{\max}, \frac{f_s}{2}]$. The kernels obtained from (5), however, have full length by default. Using them as kernels for a convolutional layer is therefore not practicable, especially if the signal length is very long. Hence, to efficiently employ such filters in this context, a limitable kernel size is required.

A. Restricted kernel size and a dichotomous auditory scale

For ISAC, we want to transfer the construction in (5) to the time domain and use a (symmetric) prototype kernel with restricted size. To achieve this, a few modifications are required. First, since filters with short kernels have full Fourier-domain support, there is no unique notion of bandwidth that we can match with B_S. We derive a suitable correspondence using a continuous, compactly supported prototype. Let $g \in \mathcal{C}_c(\mathbb{R})$ be supported on $[0, 1]$ with a peak-normalized frequency response ($\|\hat{g}\|_\infty = 1$). By specifying a reference bandwidth[2] $bw_R(g)$ of g we can determine the appropriate dilation that yields a filter kernel with bandwidth B_S when applied to g:

$$\tilde{B}_{S,g}(f) = \frac{B_S(f) \cdot bw_R(g)}{\|g\|^2}. \tag{6}$$

The final (discrete) ISAC kernel, associated with center frequency f and bandwidth $B_S(f)$ is obtained by sampling the dilated prototype g at sampling rate f_s, and discarding all values that are identically zero. See Eq. (8) in the next section for the precise expression. We denote the resulting kernel size by $T_{S,g}(f)$.

As an inherent property of the scaling function, the bandwidths are getting smaller in the lower frequencies, hence, the kernel sizes naturally become larger. If we want to set a maximum kernel size $T_{\max}$ while preserving the bandwidth requirements, we cannot follow the original scale function anymore. As a solution, we linearize it where necessary. Let $f^* \in [0, \frac{f_s}{2}]$ denote the transition frequency where $T_{\max} = T_{S,g}(f^*)$. We define a modified auditory frequency scale that is the original scale above f^* and its linearization at f^* below (Fig. 1). For an auditory scale S, the function is given as

$$\tilde{F}_S(f^*; f) = \begin{cases} F_S(f^*) + F_S'(f^*)(f - f^*) & \text{for } f \in [0, f^*] \\ F_S(f) & \text{for } f \in (f^*, \frac{f_s}{2}]. \end{cases}$$

[2]As a default, we use the -3 dB bandwidth of g.

This has the benefit that we can preserve the kernel shape while re-using the bandwidth at f^* (and the corresponding overlap) at all lower frequencies. Accordingly, we use the following modification of the bandwidth function,

$$\tilde{B}_{S,g}(f^*; f) = \begin{cases} \tilde{B}_{S,g}(f^*) & \text{for } f \in [0, f^*] \\ \tilde{B}_{S,g}(f) & \text{for } f \in (f^*, \frac{f_s}{2}], \end{cases} \tag{7}$$

implying that $T_{S,g}(f) = T_{\max}$ for every $f \in [0, f^*]$. If we want to set a minimum kernel size instead, the linearization can be done analogously for the upper frequency region.

B. The ISAC kernels

Let $g \in \mathcal{C}_c(\mathbb{R})$ be as before, even symmetric around $1/2$ (e.g., a Hann window) and f^* be the transition frequency for the maximal kernel size $T_{\max}$, i.e., $T_{S,g}(f^*) = T_{\max}$. For center frequencies f_k, $k = 1, \ldots, K$ obtained from equidistant samples on the image of $[0, \frac{f_s}{2}]$ under $\tilde{F}_S(f^*, f)$, the ISAC kernels are given by

$$g_k(\ell) = \frac{\sqrt{d_k}}{T_{S,g}(f_k)} \cdot g\left(\frac{\ell}{T_{S,g}(f_k)}\right) \cdot \exp(-2\pi i f_k \ell / L), \tag{8}$$

for $\ell = 0, \ldots, T_{S,g}(f_k) - 1$. To conform with standard implementations of convolutional layers, all kernels with size $T_{S,g}(f) < T_{\max}$ can be zero-padded to size $T_{\max}$ and centered. Recalling that $T_{S,g}(f_k)$ is the kernel size that corresponds to the bandwidth $\tilde{B}_{S,g}(f^*; f_k)$ we see that the above formula is the time domain-version of (5). The L_1-normalization via $T_{S,g}(f_k)^{-1}$ ensures the peak-normalization of the frequency response (provided that g is positive).

Due to the division of the center frequencies into an auditory and a linear part, ISAC can be interpreted as merging an AUDlet filter bank from [19] with a partial short-time Fourier transform (STFT) with kernel size $T_{\max}$ that covers the low frequency region. Note that the critical bandwidths of the human auditory sytem are assumed to be approximately constant at low frequencies and increase proportionally to frequency only at higher ranges [31]. Hence, the frequency scale for ISAC yields a reasonable approximation of auditory scales—provided that $T_{\max}$ is not too small.

Finally, by including 0 and $\frac{f_s}{2}$ as center frequencies, there is no need to add extra low- and high-pass filters, and it allows the user to conveniently choose the number of channels as a second configuration parameter, further facilitating the integration into a neural network.

C. Stability

The numerical stability of a filter bank is determined by the condition number of its frame operator matrix [8]. Let g_k, d_k for $k = 1, \ldots, K$ denote the kernels and decimation factors for a filter bank with K channels. The condition number is defined as $\kappa = B/A$, where A is the largest, and B the smallest number such that

$$A \cdot \|x\|^2 \leq \sum_{k=1}^{K} \|(x * g_k) \downarrow_{d_k}\|^2 \leq B \cdot \|x\|^2 \tag{9}$$

		kernel size			
		8†	**32**	**128**	**512**
# channels	**16**†	1.00	1.17	1.17	1.49
	40	1.00	1.04	1.05	1.08
	96	1.00	1.04	1.04	1.05
	512	1.00	1.04	1.03	1.04

TABLE I: Condition numbers of ISAC filter banks for different kernel sizes and numbers of channels. Decimation factors vary between 1 and 6. For the settings with very short filters and few channels (marked with †) we set $\gamma = 3$.

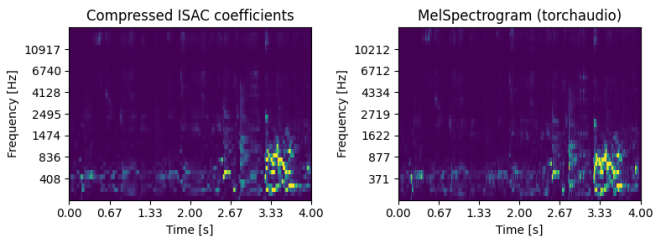

Fig. 4: Left: The compressed representation of a glitchy sound based on an ISAC filter bank ($K = 40$, $T_{\max} = 480$). Right: The mel spectrogram with the same settings. The output sizes match up to a few numbers of time bins.

for every $x \in \mathbb{R}^L$. Provided $A > 0$ exists, the filter bank is called a frame for $\mathbb{R}^L$ and the existence of a dual synthesis filter bank with perfect reconstruction is guaranteed. If $\kappa = 1$, the filter bank is said to be a tight frame and possesses convenient properties such as Parseval-type energy preservation and is its own dual filter bank. Figure 2 shows that the total PSD of the ISAC filterbank ($K = 40$, $T_{\max} = 128$), given by $\sum_{i=k}^{K} |\hat{g}_k|^2 + |\hat{\bar{g}}_k|^2$, is nearly constant, which - if decimation is chosen appropriately - provides a good indicator for κ. However, there are efficient ways to compute κ also exactly for uniform decimation [3]. Table I lists the condition numbers of ISAC filter banks for different settings, indicating that they are almost tight in all practically relevant settings.

If the kernels of the filter bank are further optimized, there are different ways to maintain a good condition number [9], [16]. One simple but effective approach [2], [20] is to add $\beta \cdot (\kappa - 1)$ as a regularizing term to the loss function, where $\beta > 0$ is a hyperparameter that controls the influence of the regularization. If the regularizer kicks in sufficiently, the filter bank can be simultaneously used as synthesis filter bank.

IV. Application

A. Compression for mel-type spectrograms

In [11] it was shown that a mel spectrogram can be rewritten in terms of a filter bank decomposition using adaptive filters on the mel scale followed by a time-averaging operation after taking the squared modulus of the filter bank coefficients. In short-hand notation,

$$|\mathrm{STFT}(x)|^2 \cdot \Lambda = |\mathrm{MEL}(x)|^2 * \phi, \tag{10}$$

where "$\cdot\ \Lambda$" shall describe the averaging to obtain the mel spectrogram and "$* \ \phi$" means channel-wise convolution over time with MEL being the adaptive filter bank. Although it is in general not known what the time averaging kernel ϕ is explicitly, the authors give an example (in the continuous case) where it can only be a Gaussian when a Gaussian window is used in the STFT. In the finite discrete setting, it is even less clear what ϕ should be. A practicable solution that yields a comparable compression as in the mel spectrogram is to use a rectangular window that has the same length as the filters divided by the decimation factor with 50% overlap. In Figure 4 we show the comparison between the proposed compression of a mel-based ISAC filter bank (left) and torchaudio's `MelSpectrogram` (right), both with 40 channels and a kernel size of 480. With the proposed implementation, it is easily possible to make the averaging kernel ϕ learnable, yielding an adaptive transform, similar to [28].

B. Learn a dual filter bank with the same kernel size

If a filter bank is a frame, there are infinitely many possible dual frames that provide perfect reconstruction [8]. Finding a dual frame that exhibits a filter bank structure, yet is not the canonical dual of the filter bank, is challenging, especially if additional properties are desired [21] - in our case, restricted kernel support [25]. With the integration into an automatic differentiation scheme, we can learn such a synthesis filter bank by minimizing the reconstruction error employing the same kernel size. Thereby, we may find a synthesis filter bank that yields close-to-perfect reconstruction. Since ISAC filter banks are almost tight, the duals are not "far away", and we are able to indeed find good approximate dual synthesis filter banks with the kernel size (reconstruction error $\approx 6e-6$). Similar to above, we can promote a flat total PSD via regularization. Such a filter bank is exemplarily depicted in Figure 3. In other words, we learn a left-inverse of the analysis operator of the filter bank that is not the Moore-Penrose pseudo inverse (resulting in the canonical dual filter bank which does not have the same kernel size).

This provides a setting that can be efficiently integrated into any neural network-based analysis-synthesis architecture, e.g., for speech enhancement [15].

V. Conclusion

The paper introduces ISAC, an invertible, stable, and perceptually-motivated filter bank with customizable kernel sizes, designed for the needs of modern machine learning. All transforms are implemented as PyTorch `nn.module` for a direct integration into PyTorch's automatic differentiation paradigm. With the option `is_learnable` the kernels can be set to be learnable parameters of the model. Everything is included in the PYPI package `hybra` which can be conveniently installed using `pip install hybra`. The accompanying GitHub project is found under https://github.com/danedane-haider/HybrA-filterbanks.

References

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Proc. NeurIPS*, 33:12449–12460, 2020.

[2] P. Balazs, D. Haider, V. Lostanlen, and F. Perfler. Trainable signal encoders that are robust against noise. In *Proc. INTER-NOISE*, volume 270, pages 1836–1844, 2024.

[3] H. Bölcskei, F. Hlawatsch, and H. G. Feichtinger. Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Processing*, 46(12):3256–3268, 1998.

[4] S. Braun and I. Tashev. Data augmentation and loss normalization for deep noise suppression. In *International Conference on Speech and Computer*, pages 79–86. Springer, 2020.

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[6] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans. nnAudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1D convolutional neural networks. *IEEE Access*, 8:161981–162003, 2020.

[7] K. Choi, D. Joo, and J. Kim. Kapre: On-GPU audio preprocessing layers for a quick implementation of deep neural network models with Keras. *arXiv:1706.05781*, 2017.

[8] O. Christensen. *An Introduction to Frames and Riesz Bases*. Applied and Numerical Harmonic Analysis. Birkhäuser Boston, 2002.

[9] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. *Proc. ICML*, 2017.

[10] O. Derrien, T. Necciari, and P. Balazs. A quasi-orthogonal, invertible, and perceptually relevant time-frequency transform for audio coding. In *Proc. EUSIPCO*, pages 799–803. IEEE, 2015.

[11] M. Dörfler, T. Grill, R. Bammer, and A. Flexer. Basic filters for convolutional neural networks applied to music: Training or design? *Neural Computing and Applications*, 32:941–954, 2020.

[12] Q. Fu, Z. Teng, J. White, M. E. Powell, and D. C. Schmidt. Fastaudio: A learnable audio front-end for spoof speech detection. In *Proc. ICASSP*, pages 3693–3697. IEEE, 2022.

[13] B. R. Glasberg and B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1-2):103–138, 1990.

[14] D. Haider, V. Lostanlen, M. Ehler, and P. Balazs. Instabilities in convnets for raw audio. *IEEE Signal Processing Letters*, 31:1084–1088, 2024.

[15] D. Haider, F. Perfler, V. Lostanlen, M. Ehler, and P. Balazs. Hold me tight: Stable encoder-decoder design for speech enhancement. In *Proc. InterSpeech*. ISCA, 2024.

[16] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. Parseval proximal neural networks. *Journal of Fourier Analysis and Applications*, 26(59):1531–5851, 2020.

[17] Y. Luo and N. Mesgarani. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 27(8):1256–1266, 2019.

[18] T. Necciari, P. Balazs, N. Holighaus, and P. Søndergaard. The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction. In *Proc. ICASSP*, 2013.

[19] T. Necciari, N. Holighaus, P. Balazs, Z. Průša, P. Majdak, and O. Derrien. Audlet filter banks: A versatile analysis/synthesis framework using auditory frequency scales. *Applied Sciences*, 8(1), 2018.

[20] R. Nenov, D. Haider, and P. Balazs. (Almost) Smooth Sailing: Towards numerical stability of neural networks through differentiable regularization of the condition number. In *Proc. Differentiable Almost Everything Workshop at ICML*, 2024.

[21] N. Perraudin, N. Holighaus, P. Soendergaard, and P. Balazs. Designing Gabor windows using convex optimization. *Applied Mathematics and Computation*, 330:266–287, 2018.

[22] M. Ravanelli and Y. Bengio. Speaker recognition from raw waveform with SincNet. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 1021–1028, 2018.

[23] H. B. Sailor, D. M. Agrawal, and H. A. Patil. Unsupervised filterbank learning using convolutional restricted Boltzmann machine for environmental sound classification. In *Proc. InterSpeech*, volume 8, page 9, 2017.

[24] T. N. Sainath, B. Kingsbury, A.-R. Mohamed, and B. Ramabhadran. Learning filter banks within a deep neural network framework. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 297–302, 2013.

[25] D. T. Stoeva. On compactly supported dual windows of Gabor frames. *Journal of Mathematical Analysis and Applications*, 505(1):125436, 2022.

[26] N. Tomen and J. C. van Gemert. Spectral leakage and rethinking the kernel size in CNNs. In *Proc. ICCV*, pages 5118–5127, 2021.

[27] H. Yu, Z.-H. Tan, Y. Zhang, Z. Ma, and J. Guo. DNN filter bank cepstral coefficients for spoofing detection. *IEEE Access*, 5:4779–4787, 2017.

[28] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi. LEAF: A learnable frontend for audio classification. In *Proc. ICML*, 2021.

[29] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux. Learning filterbanks from raw speech for phone recognition. In *Proc. ICASSP*, pages 5509–5513, 2018.

[30] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4):99, 2024.

[31] E. Zwicker. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.