FASTAUDIO: A LEARNABLE AUDIO FRONT-END FOR SPOOF SPEECH DETECTION

Quchen Fu[⋆]*, Zhongwei Teng*[⋆]*, Jules White*[⋆]*, Maria E. Powell*[†]*, Douglas C. Schmidt* [⋆]

[⋆] Dept. of Computer Science, Vanderbilt University
[†] Dept. of Otolaryngology–Head and Neck Surgery, Vanderbilt University Medical Center

ABSTRACT

Spoof speech can be used to try and fool speaker verification systems that determine the identity of the speaker based on voice characteristics. This paper compares popular learnable front-ends on this task. We categorize the front-ends by defining two generic architectures and then analyze the filtering stages of both types in terms of learning constraints. We propose replacing fixed filterbanks with a learnable layer that can better adapt to anti-spoofing tasks. The proposed FastAudio front-end is then tested with two popular back-ends to measure the performance on the Logical Access track of the ASVspoof 2019 dataset. The FastAudio front-end achieves a relative improvement of 29.7% when compared with fixed front-ends, outperforming all other learnable front-ends on this task.

Index Terms— Spoof Speech Detection, Automatic Speaker Verification, Learnable Audio Filterbanks

1. INTRODUCTION

Spoof speech detection identifies attempts to fool a speaker verification system, including Text-To-Speech (TTS), Voice Conversion (VC), and Replay Attacks. In spoof speech detection, audio is preprocessed to create a compressed representation that is smaller in size, but aims to preserve as many of the important features as possible before spoof detection is applied. The component that performs this preprocessing step is known as the front-end. Front-ends can be either handcrafted or learnable, and the process of choosing the proper handcrafted front-ends is also known as feature selection.

This paper provides three contributions to the study of how front-ends contribute to spoof speech detection performance. First, we propose a lightweight learnable front-end called FastAudio that achieved the lowest min t-DCF in spoof speech detection compared to other front-ends. Second, we provide a comparison of feature selections for spoofing countermeasures, with a special focus on learnable audio front-ends, and show how applying shape constraints can make the filterbank layer perform better while reducing the number of parameters. Third, we describe the architecture that achieved top performance on the ASVspoof 2019 [1] dataset.

The remainder of this paper is organized as follows: Section 2 summarizes the classification of audio front-ends based on structure and the background for filter learning; Section 3 discusses different constraint types regarding filterbank learning; Section 4 describes our experiment setups, including dataset, metric, and model details; Section 5 analyzes the result and describes our experiment insights regarding filter learning for spoof speech detection, and Section 6 presents concluding remarks.

2. BACKGROUND ON AUDIO FRONT-END

Spoof speech detection is a single-task classification problem, for which many front-ends have been tested, including Instantaneous Frequency (IF), Group Delay (GD), and Mel-frequency cepstral coefficients (MFCC), etc. The front-ends used in the classification of speech have been dominated by MFCC and recently Log Mel FilterBanks (FBanks), both of which are hand-crafted features that are fixed and not learnable. Constant Q Transform (CQT) [2] is another handcrafted front-end commonly used for music generation and music note recognition as it can better mimic musical scales. However, prior research reported CQT was also the best performing front-end for spoof detection [3].

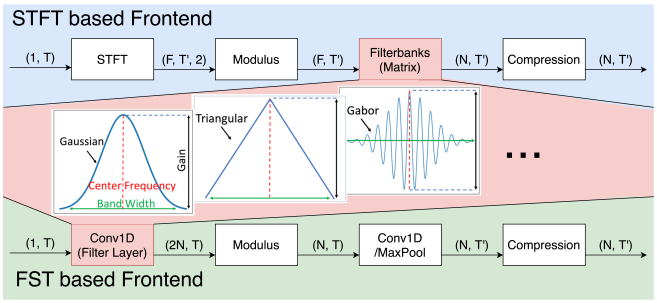

Fig. 1. FST based front-end and STFT based front-end

As shown in Figure 1, front-ends can be categorized by the procedures they perform. There are two key categories: First-order Scattering Transform (FST) [4] based front-ends and Short-Time Fourier Transform (STFT) based front-ends. Unlike STFT which multiplies a filterbank matrix with a spectrogram, FST uses a convolutional layer on the raw audio waveform to approximate the standard filtering process.

While FST-based front-end approaches have made progress, prior research has shown they lose signal energy, which cor-

responds to information loss, since only the first-order coefficients of a scattering transform are used [4]. The FST-based approaches are also time-consuming [5] since convolution layers with large kernels are computation intensive.

STFT based front-ends remain popular, and FBanks are still the front-end for the state-of-the-art speaker identification [6] and speech recognition [7] systems. However, many STFT based front-ends are fixed and may not adapt well to certain downstream tasks [8].

Both types of front-ends employ some type of filter-like manipulations to model the non-linearity of the human ear's sensitivity to frequency. The distribution of filter center frequency is referred to as scale. Studies [9] have shown that the Mel-scale, as shown in Equation 1, can capture human perception for pitch relatively well.

$$m = 2595 \log_{10} \left(1 + \frac{f}{700}\right) \tag{1}$$

To make this manipulation domain adaptable, filters can be made learnable. As shown in Figure 1, a filterbank can learn its center frequency c_n, gain g_n, bandwidth b_n, and shape s_n. The filter properties can be summarized in Equation 2.

$$\omega_n(f) = g_n s_n(c_n; b_n; f) \tag{2}$$

3. RESEARCH QUESTIONS REGARDING LEARNABLE FILTERBANKS

Previous research has explored the feasibility of learnable filterbanks. For example, nnAudio [10] implemented a set of unconstrained learnable filterbanks. T. Sainath *et al.* [11] reported limited improvement from unconstrained filterbank learning. DNN-FBCC [12] explored some constraints over filters by adopting a mask matrix. Zhang and Wu [13] described a detailed study on the shape and positiveness constraint's effect on the filterbanks. However, no systematic study has been done on constraining the filterbank shape in the STFT-based approach used for spoof speech detection.

Type	Name	Filter/BandWidth		Center Frequency		Gain
		Shape	Clamp	Sorted	Clamp	
FST based	TD-FBanks	Gabor	-	Yes	Yes	-
	SincNet	Sinc	Yes	No	Yes	Fixed*
	LEAF	Gabor	Yes	No	Yes	Fixed
STFT based	nnAudio	-	No	No	No	-
	DNN-FBCC	-	Yes	Yes	Yes	-
	FastAudio	Triang	Yes	No	Yes	Fixed

Table 1. Filter Comparison of Learnable Front-end

As shown in Table 1, all current FST-based front-ends put shape constraints on the band-pass filters. However, STFT based front-ends, like DNN-FBCC, do not constrain the filter shape. Instead, a mask is put on the filters so that the bandwidth is clamped and the filters are sorted by center frequencies. We therefore designed a learnable front-end, called FastAudio, specifically focused on answering the following questions: 1. Is a shape constraint necessary for spoof detection, and which shape constraint has the lowest min t-DCF? 2. Should the center frequency be sorted for spoof detection? 3. What do trained filterbanks learn about spoof detection compared to handcrafted FBanks? These questions are discussed in subsection *5.3* and *5.4* of Section 5.

4. EXPERIMENT AND DATASET

The ASVspoof 2019 corpus consists of two parts: Logical Access and Physical Access. Here we focus on the Logical Access (LA) task. LA contains fake (spoof) speech generated from various text-to-speech and voice conversion techniques. The true speech audio files are referred to as *Bona fide*.

Since there are existing Automatic Speech Verification (ASV) systems that provide some protection against spoofing attacks, the goal is to design a system that can best complement existing ASV systems (the result of the existing ASV system is provided by the dataset in labels). The system we have developed is called the Countermeasures (CM). The evaluation metric is the tandem detection cost function (min t-DCF), which is designed to best reflect real-world protection effects.

4.1. Dataset

The performance of the FastAudio learnable front-end is evaluated on the ASVspoof 2019 LA dataset. The training and development sets contain data generated from the same algorithms. However, to ensure the spoof detection system can generalize well to audio of unseen types, the evaluation set also contains attacks that are generated from different algorithms.

4.2. Metrics

The primary metric for spoof speech detection is the minimum normalized tandem detection cost function (min t-DCF), as shown in Equation 3. The min t-DCF measures the overall protection rate for combined CM and ASV systems, where β depends on application parameters (priors, costs) and ASV performance (miss, false alarm, and spoof miss rates), while $P^{\text{cm}}_{\text{miss}}(s)$ and $P^{\text{cm}}_{\text{fa}}(s)$ are the CM miss and false alarm rates at threshold s [1].

$$\text{t} - \text{DCF}^{\text{min}}_{\text{norm}} = \min_{s} \{\beta P^{\text{cm}}_{\text{miss}}(s) + P^{\text{cm}}_{\text{fa}}(s)\} \tag{3}$$

Equal error rate (EER) was used as a secondary metric to make comparison possible with earlier datasets like ASVSpoof 2017. EER is defined as the value of false acceptance rate and false rejection rates where they are equal.

4.3. Back-end

Our FastAudio front-end consists of an STFT transform followed by a learnable filterbank layer, and finally a log compression layer to mimic the non-linearity of human sensitivity to loudness. We integrated the front-ends with two of the most popular back-ends for audio classification: X-vector [14] [15]

and ECAPA-TDNN [16] [15]. The back-end turns the FBank-variant into a 256-dimensional embedding vector. The vectors are then fed into a linear classifier.

Xvector		ECAPA-TDNN	
Layer	**Output**	**Layer**	**Output**
Input	(N, T′)	Input	(N, T′)
TDNN X 5	(1500, T′)	Conv1D + ReLU + BN	(C, T′)
Stats Pool	(3000, 1)	SE-Res2Block X 3	(3, C, T′)
Linear	(256, 1)	Conv1D + ReLU	(1536, T′)
		Atten Stats Pool + BN	(3072, 1)
		FC + BN	(256, 1)

Table 2. X-vector and ECAPA-TDNN

4.4. Experimental Setup

This model was trained on 2 Nvidia 2080 Ti GPUs for 100 epochs and the batch size was set to 12 (except for TD-filterbank whose batch size was 4 to stay within memory limits). We also compared the performance of our front-end with other STFT-based and FST-based front-ends, both under learnable and fixed settings.

To make the comparison fair, we kept the hyperparameters across all experiments the same so that the front-end outputs have the same dimensions. The sampling rate was set to 16kHz, window length to 25ms, window stride to 10ms, and the number of filters to 40. All learnable front-ends were initialized to mimic Mel-FBanks, as previous research [17] has shown that random initialization has worse performance. Detailed hyperparameters and data augmentation are available on our GitHub repository.[1]

5. RESULTS AND ANALYSIS

5.1. How do learnable front-ends perform on min t-DCF compared with handcrafted front-ends for spoof speech detection?

Since recent systematic comparisons of front-ends on spoof detection were done in 2015 [18], we designed the experiments so that an updated baseline can be established, including learnable front-ends. We choose a combination of FST and STFT front-ends with both fixed and learnable settings so that the experiment is comprehensive. We found the FST-based learnable front-ends need longer training time than hand-crafted features in the spoof speech detection task and cannot beat the performance of CQT, as shown in Table 3.

5.2. Can we design an STFT-based front-end for spoof speech detection that is learnable and can it beat the performance of CQT?

Since FST-based learnable front-ends failed to beat the performance of CQT, we designed a front-end following the traditional STFT-based approach and limited the number of trainable parameters. We call this approach FastAudio since it trains faster than FST-based front-ends. We hypothesize that instead of changing the front-end architecture completely (e.g., as in FST-based approaches) we can boost the fixed STFT-based performance by making the filterbank layer learnable. We tested FastAudio under three different constraint settings and the best one achieved 29.7% decrease in min t-DCF compared to FBanks, outperforming CQT (See Table 3).

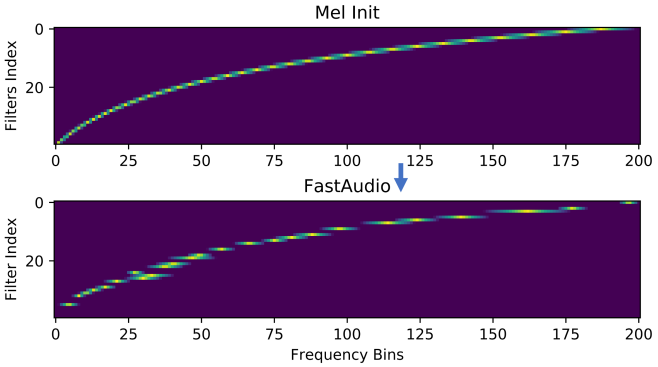

Fig. 2. Heatmap of the magnitude of the frequency response for initialization filters (up) and learned filters (down).

5.3. Which set of constraints for filterbank learning performs best in spoof speech detection?

We found the existence of shape constraint plays an important role in improving spoof detection accuracy. However, we did not find a significant difference in constraining the shape of the filters to be Gaussian or Triangular. We found that sorting the filterbanks by center frequency does not improve accuracy, which confirms the conclusion from [19]. As shown in Figure 2, the learned filterbank distribution closely follows the hand-crafted filterbanks in both center frequency and bandwidth. The similarity in c_n and b_n helps explain the strong performance of handcrafted features compared to the learnable front-end, especially compared to the FST-based front-ends.

The visualization of the front-end output is shown in Figure 3. All outputs contain "horizontal lines" that correspond to certain frequencies, which is a sign of filter selectiveness. We found that front-end output like LEAF, TD-filterbanks, and nnAudio changed greatly after training due to the number of trainable parameters. When the shape of the filters is not constrained, as shown in nnAudio, the trained front-end shows signs of over-fitting (many randomly distributed dots) and has the worst performance. Since the nnAudio has no constraint for filter shape, the learned filter shape is determined by 201 points, so t may contain sharp peaks and select frequencies of narrow ranges, thus creating the irregular dots.

5.4. What does FastAudio learn about spoof speech detection and how can we interpret what it learns?

Formants are the spectral peaks resulted from acoustic resonance of the human vocal tract. Since in English vowels contain more energy than consonants, we expect our learned

[1] https://github.com/magnumresearchgroup/Fastaudio

			ECAPA-TDNN		X-vector			
Front-end	**#Params**	**Constraint**	**EER**	**min t-DCF**	**EER**	**min t-DCF**	**MACs**	**Train Time/Epoch**
CQT	0	Fixed	1.73	0.05077	3.40	0.09510	0	10:58 min
Fbanks	0	Fixed	2.11	0.06425	2.39	0.06875	0	10:53 min
FastAudio-Tri	**80**	**Shape+Clamp**	**1.54**	**0.04514**	**1.73**	**0.04909**	**0.00GMac**	**13:02 min**
FastAudio-Gauss	80	Shape+Clamp	1.63	0.04710	1.67	0.05158	0	12:51 min
FastAudio-Sort	80	Shape+Clamp+Order	1.89	0.05204	1.69	0.05235	0	12:59 min
LEAF	282	Shape+Clamp	2.49	0.06445	3.28	0.07319	0.01GMac	34.45 min
nnAudio	8.04k	No	3.63	0.08929	5.56	0.14707	0	13:00 min
TD-filterbanks	31k	Shape+Clamp	1.83	0.05284	3.18	0.08427	1.32GMac	22.48 min
Front-end	**Name**	**Constraint**	**EER**	**min-tDCF**	**Backend**			**Baseline**
SincNet	RawNet2	Fixed	5.13	0.1175	-			-

Table 3. A stage-wise comparison of the different Front-ends' performance on the ASVspoof 2019 LA dataset

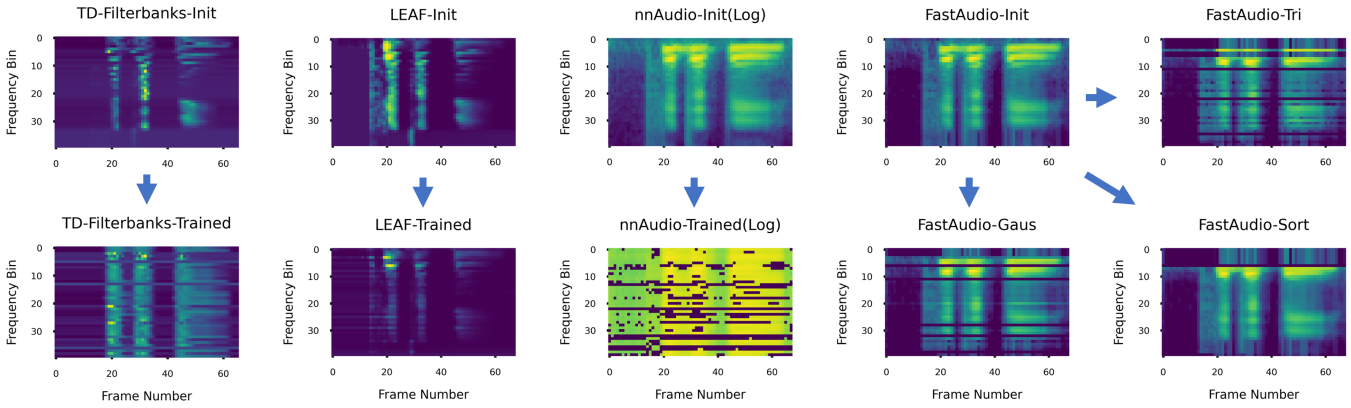

Fig. 3. Visualization of Learnable Front-ends

filters center frequencies to concentrate around the average formants of English vowels [20]. We plotted the cumulative frequency response of the FastAudio in Figure 4. We found 2 peaks in the lower frequency and 1 peak in the high frequency.

The peaks in frequencies around 320Hz∼440 Hz and 1120Hz may correspond to the $1st$ and $2nd$ formants averaged over all vowels in English [21]. This adaptation to human speech suggests FastAudio was able to successfully learn what is important for spoof speech detection tasks. Similar adaptation was also reported in the FST-based front-end for speech identification tasks [21].

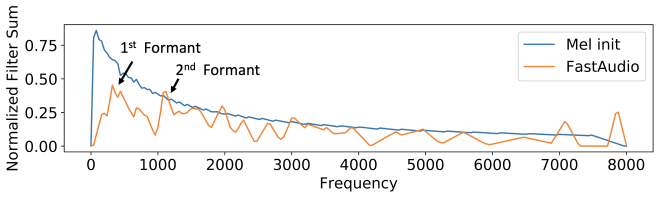

Fig. 4. Cumulative frequency response of the FastAudio filters

We also found peaks in the high pitch regions near the sampling boundary, which suggests spoof speech may differ from the real speech in frequencies that are "ignored" by scales used by handcrafted front-ends like the Mel-scale. High-frequency energy was deemed less important and subsequently underrepresented in Mel-scales. However, in the spoof detection task, we suspect that because these high frequencies are "unimportant" to human hearing, the spoof speech generator does not create realistic imitation in high frequencies. Thus, the representation of high-frequency data may be a good indicator for spoof speech detection.

Together, these findings indicated that: 1. Learned FastAudio filters are more selective than their initialization. 2. FastAudio emphasizes frequencies around 1st and 2nd formants, which may be important for distinguishing between spoof and *bona fide* speech. 3. Learned FastAudio filters are more sensitive to high-frequency energy, which may be a salient feature of spoof detection. 4. Through end-to-end training, FastAudio can adapt to spoof detection tasks. The front-end successfully adapted to the downstream task and was able to learn the phonetics of human speech.

6. CONCLUDING REMARKS

This paper investigates the performance of learnable front-ends on spoof detection and proposes an STFT-based audio front-end called FastAudio. We tested the proposed front-end under different constraint settings and found FastAudio successfully adapted to spoof detection. The proposed front-end achieves top performance on the ASVspoof 2019 dataset, beating the fixed equivalent by 29.7% and surpassing the performance of CQT, which was reported as the best hand-crafted feature for spoof speech detection.

7. REFERENCES

[1] M. Todisco, X. Wang, Ville Vestman, Md. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and Kong-Aik Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *ArXiv*, vol. abs/1904.05441, 2019.

[2] Judith C. Brown, "Calculation of a constant q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, pp. 425–434, 1991.

[3] Xu Li, N. Li, Chao Weng, Xunying Liu, Dan Su, Dong Yu, and H. Meng, "Replay and synthetic speech detection with res2net architecture," in *ICASSP*, 2021.

[4] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Transactions on Signal Processing*, vol. 62, pp. 4114–4128, 2014.

[5] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, pp. 150, 2018.

[6] Brecht Desplanques, Jenthe Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *INTERSPEECH*, 2020.

[7] J. Villalba, N. Chen, David Snyder, D. Garcia-Romero, A. McCree, Gregory Sell, Jonas Borgstrom, L. P. García-Perera, F. Richardson, R. Dehak, P. Torres-Carrasquillo, and Najim Dehak, "State-of-the-art speaker recognition with neural network embeddings in nist sre18 and speakers in the wild evaluations," *Comput. Speech Lang.*, vol. 60, 2020.

[8] Zhongwei Teng, Quchen Fu, Jules White, Maria Golla Powell, and Douglas C. Schmidt, "Complementing handcrafted features with raw waveform using a light-weight auxiliary model," *ArXiv*, vol. abs/2109.02773, 2021.

[9] R. Lippmann, "Speech recognition by machines and humans," *Speech Commun.*, vol. 22, pp. 1–15, 1997.

[10] K. Cheuk, Hans Anderson, Kat R. Agres, and Dorien Herremans, "nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks," *IEEE Access*, vol. 8, pp. 161981–162003, 2020.

[11] T. Sainath, Brian Kingsbury, Abdel rahman Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 297–302, 2013.

[12] Hong Yu, Z. Tan, Yiming Zhang, Zhanyu Ma, and Jun Guo, "Dnn filter bank cepstral coefficients for spoofing detection," *IEEE Access*, vol. 5, pp. 4779–4787, 2017.

[13] Teng Zhang and Ji Wu, "Discriminative frequency filter banks learning with neural networks," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, pp. 1–16, 2019.

[14] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[15] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, S. Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, A. Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, Franccois Grondin, William Aris, Hwidong Na, Yan Gao, R. Mori, and Yoshua Bengio, "Speechbrain: A general-purpose speech toolkit," *ArXiv*, vol. abs/2106.04624, 2021.

[16] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[17] Neil Zeghidour, Nicolas Usunier, I. Kokkinos, Thomas Schatz, Gabriel Synnaeve, and Emmanuel Dupoux, "Learning filterbanks from raw speech for phone recognition," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5509–5513, 2018.

[18] X. Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Chng Eng Siong, and Haizhou Li, "Spoofing speech detection using high dimensional magnitude and phase features: the ntu approach for asvspoof 2015 challenge," in *INTERSPEECH*, 2015.

[19] Neil Zeghidour, O. Teboul, F. D. C. Quitry, and M. Tagliasacchi, "Leaf: A learnable frontend for audio classification," *ArXiv*, vol. abs/2101.08596, 2021.

[20] B. Lindblom, "Explaining phonetic variation: A sketch of the h&h theory," 1990.

[21] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, 2018.