THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Acoustic model adaptation from raw waveforms with Sincnet

OPEN ACCESS

ACOUSTIC MODEL ADAPTATION FROM RAW WAVEFORMS WITH SINCNET

Joachim Fainberg, Ondřej Klejch, Erfan Loweimi, Peter Bell, Steve Renals

Centre for Speech Technology Research, University of Edinburgh, United Kingdom

ABSTRACT

Raw waveform acoustic modelling has recently gained interest due to neural networks' ability to learn feature extraction, and the potential for finding better representations for a given scenario than hand-crafted features. SincNet has been proposed to reduce the number of parameters required in raw-waveform modelling, by restricting the filter functions, rather than having to learn every tap of each filter. We study the adaptation of the SincNet filter parameters from adults' to children's speech, and show that the parameterisation of the SincNet layer is well suited for adaptation in practice: we can efficiently adapt with a very small number of parameters, producing error rates comparable to techniques using orders of magnitude more parameters.

Index Terms— Acoustic model adaptation, children's speech, raw waveform, SincNet

1. INTRODUCTION

Automatic speech recognition models have in recent years obtained impressive word error rates (WERs) [1]. A key component to improving performance is to reduce mismatch between the acoustic model and test data, by explicit adaptation or normalisation of acoustic factors (e.g. [2, 3, 4, 5, 6]). Methods such as Vocal Tract Length Normalisation (VTLN) [7], which aims to mitigate large variations in individual speakers acoustics, scales the filterbank in standard feature extraction. There has, however, been a growing interest in reducing the amount of hand-crafted feature extraction that is required for acoustic modelling of speech [8, 9, 10, 11]. The motivations to learn part, or all, of the feature extractor range from aiding interpretability [8, 10], to obtaining more optimal representations for the task at hand [12]. Jaitly and Hinton [13], for example, argued that low-dimensional, hand-crafted features, such as Mel-frequency cepstral coefficients (MFCCs), may lose relevant information that is otherwise present in the original signals.

From raw time-domain waveforms, convolutional neural networks (CNN) have shown promising results [8, 9, 14, 15].

It has even been demonstrated that it is possible to learn band-pass beamformers from multi-channel raw waveforms [15], and a feature extractor learned from raw frequency representations of speech has been shown to outperform conventional methods [16]. Their interpretability, however, is sometimes limited, and it is not always clear how to apply existing adaptation techniques. In a recent approach called SincNet [10], Ravanelli and Bengio propose to constrain the CNN filters learned from raw time-domain signals, by requiring each kernel to model a rectangular band-pass filter. The authors show that this yields improved efficiency, and that the filters are more easily interpretable.

In this paper we propose to make use of these characteristics for the adaptation of raw waveform acoustic models: we would like efficient, compact representations that are quick to estimate and cheap to store. We explore whether we can obtain this by adapting the cut-off frequencies, and the gains of the filters in SincNet. This layer may be particularly well suited for speaker adaptation, as the lower layers are known to carry more speaker information than the other layers [17, 18]. We will show in Section 2.1 that adapting this parameterisation of the CNN filters has similarities with, and crucial differences from, VTLN, feature-space Maximum Likelihood Linear Regression (fMLLR) [3], and Learning Hidden Unit Contributions (LHUC) [2]. VTLN has been used to mitigate large variations in vocal tract length for the recognition of children's speech [19]. In our experiments we adapt from adults' to children's speech and show that we obtain VTLN-like scaling functions of the filter frequencies.

There are related approaches in literature that aim to learn, and update filterbanks on top of e.g. raw spectra [12, 9, 20, 21]. As argued in these papers, fixed filterbanks may not be an optimal choice for a particular task. Sailor and Patil [22] indeed showed that their proposed convolutional restricted Boltzmann machine (RBM) model learns different centre frequencies depending on the task at hand. Our work is perhaps most closely related to Seki et al. [21], who proposed to adapt a filterbank composed of differentiable functions such as Gaussian or Gammatone filters. They demonstrated more than 7% relative reductions in WER when adapting to speakers in a spontaneous Japanese speech transcription task. Our work differs in that we propose to adapt the SincNet layer, which operates on raw waveforms, rather than power spectra.

We review SincNet and related methods in Section 2. Sec-

tion 3 presents the experimental setup, with results in Section 4. Section 5 concludes the paper.

2. SINCNET

The idea of SincNet [10] is to use rectangular band-pass filters in place of standard CNN filters for raw-waveform acoustic models. A rectangular filter with lower and upper cut-off frequencies f_l and f_u has the following time-domain representation, represented as the difference between two low-pass filters:

$$g[n, f_u, f_l] = 2f_u \operatorname{sinc}(2\pi f_u n) - 2f_l \operatorname{sinc}(2\pi f_l n). \quad (1)$$

Consequently, the number of parameters per filter is reduced from having to model every tap of each filter (i.e. the filter length) to only having to model two: the cut-off frequencies of the filters, regardless of filter length. An example of learned filters are shown in Figure 1. As in [10] we use Hamming windows [23] to smooth discontinuities towards the edges:

$$w[n] = 0.54 - 0.46 \cos(\frac{2\pi n}{L}), \quad (2)$$

where L is the filter length. Consequently, the final forward pass for speech input $x[n]$ with one filter is:

$$y[n] = x[n] * g_w[n, f_u, f_l] = x[n] * w[n] \, g[n, f_u, f_l]. \quad (3)$$

A set of filters becomes a learnable filterbank of approximately rectangular filters. A related method by Seki et al. [20] replaced the standard Mel-filterbank during feature extraction of Mel-frequency cepstral coefficients (MFCCs) with differentiable Gaussian filters on top of power spectra, enabling the learning of centre frequencies, bandwidths and gain. SincNet also learns a filterbank, but in the time-domain on raw waveform features. For SincNet, Ravanelli and Bengio [10] chose not to explicitly model the gain of each filter, as it can be readily learned by later parts of the neural network.

2.1. Relationship with VTLN, fMLLR and LHUC

A learnable filterbank has close relationships with other well-known methods, as also previously highlighted by Seki et al. [20]. In this paper we suggest to update the SincNet filterbank for each speaker. This strongly resembles VTLN [7], which aims to compensate for varying vocal tract lengths among speakers. It accomplishes this by scaling, or warping, the centre frequencies of the filters in the Mel-filterbank. Consequently, adapting the parameters of the SincNet layer resembles VTLN with a few key differences:

1. SincNet operates in the time-domain, and uses corresponding rectangular filters rather than triangular filters as in the Mel-filterbank;
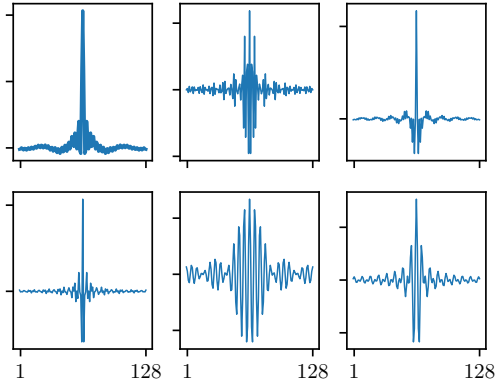

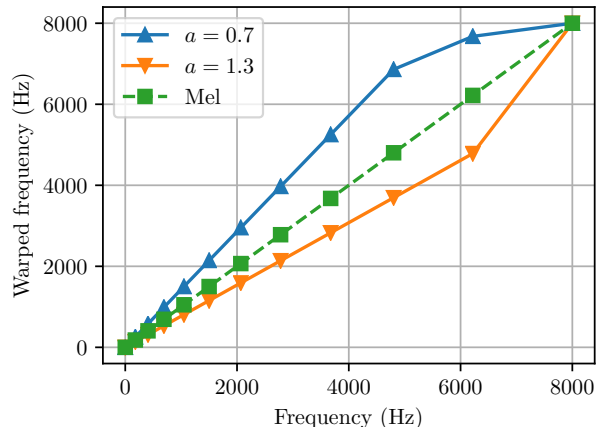**Fig. 1**. Examples of learned bandpass filters in the time-domain.

Fig. 2. Example of piece-wise linear scaling functions for VTLN. $\alpha = 1$ would equal the Mel line.

2. VTLN typically uses a scaling function that is assumed to be piece-wise linear with a single slope parameter, α (as shown in Figure 2), whilst if adapting SincNet, the effective learned scaling functions are less constrained.

3. The slope parameter α is typically determined with a grid search (although, there exist more sophisticated methods such as gradient search [24]). With SincNet we can learn the scaling function using gradient descent.

In the original SincNet formulation [10] the gains of the filters are held fixed. Downstream layers can learn to scale the contributions of the filters. However, the filter gains may be suitable targets for adaptation for which we would like to attribute importance to the output of individual filters with a small number of parameters. This has similarly been done with learnable filterbanks in traditional feature extraction pipelines [21]. We also briefly note that if we were to scale the gain of each filter, then this would correspond to a version

of feature-space Maximum Likelihood Linear Regression (fMLLR) [3] with a diagonal matrix and no bias, or similarly to Learning Hidden Unit Contritutions (LHUC) [2] which scales the output of each neuron by a scalar $r^{(i)}$ for filter i:

$$y^{(i)}[n] = r^{(i)} \sum_{l=0}^{L-1} x[l] \, g_w^{(i)}[n-l]. \qquad (4)$$

Clearly, we can view the vector of scalars, $\mathbf{r}$, as either scaling the features, the gain of the filters, or the output of the layer.

3. EXPERIMENTAL SETUP

Our baseline models are built using the AMI Meeting Corpus [25], which contains about 70 hours of training data from fictitious design team meetings. We use the individual head-mounted microphone (IHM) stream, and trained HMM-GMM systems using Kaldi [26] following the recipe for AMI[1].

As adaptation data we use children's speech from the British English PF-STAR corpus [27], which in total consists of roughly 14 hours of data of read children's speech. The children are aged between 4–14, with the majority being 8–10 years old. The data contains a fair amount of mispronunciation and hesitation, making recognition challenging. It is clearly mismatched from the AMI data, in terms of what is spoken, the speaking style, and the acoustics of the speakers (see e.g. [28]).

3.1. SincNet acoustic model

The neural network acoustic model is detailed in Table 1. The first layer consists of 40 Sinc filters, each with length 129 as has been used previously in a speech recognition task [29]. We experimented with different methods of initialising the upper and lower frequencies of each filter as follows:

1. Mel-scale [30] as in Ravanelli and Bengio [10]: the lower frequencies of the filters are linearly interpolated between the corresponding mels of f_{min} and f_{max}, with $f_u[i] = f_l[i-1]$;

2. Uniformly at random in the same range. This effectively becomes linear when sorted by centre frequency: $f_l \sim \mathcal{U}(f_{min}, f_{max})$, and $f_u[i] = f_l[i-1]$;

3. Flat with $f_l = f_{min}$ and $f_u = f_{min} + b$ for each filter (randomness is induced by the layers above).

For each scheme we set $f_{min} = 30\,\mathrm{Hz}$ and $f_{max} = sr/2 - (f_{min} + b)$, where sr is the sampling rate (16 kHz in our experiments), and $b = 50\,\mathrm{Hz}$ is the minimum bandwidth. The remaining layers consist of six 1-D convolution layers with ReLUs, each with 800 units. Kernel sizes and dilation rates

are shown in Table 1. Batchnorm (BN) layers are interspersed throughout. The final softmax layer outputs to 3,976 tied states.

The models are trained for 6 epochs using Adam [31] with a batch size of 256 and a learning rate of 0.0015, unless noted otherwise. The waveforms are sampled as in [10, 29]: we use 200 ms windows with a shift of 10 ms, i.e. the input size to the network is $16000 * 0.200 = 3200$. We implemented and trained the models using Keras [32] and Tensorflow [33]. We decode and score with Kaldi [26]. Our experimental code is publically available[2].

#	Type	Dim	Size	Dil	Params
1	SincConv	40	129	-	80
-	MaxPooling	-	3	-	
2	BN(ReLU(Conv))	800	2	1	68,000
-	MaxPooling	-	3	-	
3	BN(ReLU(Conv))	800	2	3	1,284,000
-	MaxPooling	-	3	-	
4	BN(ReLU(Conv))	800	2	6	1,284,000
-	MaxPooling	-	3	-	
5	BN(ReLU(Conv))	800	2	9	1,284,000
-	MaxPooling	-	2	-	
6	BN(ReLU(Conv))	800	2	6	1,284,000
7	ReLU(Conv)	800	1	1	640,800
8	Softmax(Conv)	3976	1	1	3,184,776

Table 1. Model topology. In total there are 9,029,656 parameters (including batchnorm).

3.2. Language model

As the acoustic and language models for AMI are greatly mismatched to PF-STAR, we interpolate the standard AMI language model based on AMI and Fisher [34] data, with the training data from PF-STAR. This is similar to other literature working with PF-STAR [35]. We note, however, that there is some overlap in the sentences between training and test sets for PF-STAR, i.e. training a LM on the training set causes some data leakage. For this paper we believe this is acceptable, given that we are interested in the acoustic model mismatch. Without the biased LM, the combined effect of a mismatched LM, and a mismatched AM, produced WERs greater than 90% in our preliminary experiments.

We estimate a 3-gram LM with Kneser-Ney discounting on the PF-STAR training set using the SRILM toolkit [36]. This is interpolated with the AMI model, giving the latter a weight of 0.7. The vocabulary is restricted to the top 150k words from an interpolated 1-gram model. Finally, we prune the interpolated model with a threshold of 10^{-7}.

4. RESULTS

The results with our models trained on AMI are shown in Table 2. The various initialisation schemes produce quite similar WERs, but, perhaps surprisingly, the Mel-initialisation performs least well. The differences are, however, less than 2% relative. Overall these numbers are roughly 5 percentage points worse than those produced with cross-entropy systems in the corresponding Kaldi recipe for AMI. A key difference may be the use of speed perturbation for data augmentation [37]. Our models are slow to train, but we propose improving training speed as an area for future work.

Figure 3 demonstrates how the initialisation schemes lead to different final responses in the filters. The flat initialisation is essentially forced to change significantly, otherwise each filter would extract identical information. After training it begins to approximate a Mel-like curve. This is in line with similar research [9, 22]. We noted in Section 3.1 that the uniform initialisation effectively creates a linear initialisation. It remains largely linear after training, with some shifts in higher frequencies, and changes to bandwidths. Note that each response is markedly different, yet the corresponding WERs are similar. This may be explained by the ability of the downstream network to learn to use the extracted features in different ways [21]. We also experimented with a larger number of filters than 40 (e.g. 128), but saw no benefit to WERs; instead, the filter bandwidths become quite erratic, as demonstrated in Figure 4.

We use the model trained from a flat initialisation for our further experiments.

Initialisation	Eval	Dev
Mel	30.6	28.0
Flat	30.2	28.0
Uni	30.3	27.9

Table 2. Results (% WER) on AMI with various filter initialisation.

4.1. Domain adaptation to children's speech

We next investigate supervised domain adaptation of the Sinc-Conv layer from AMI to PF-STAR (from adults' to children's speech). As shown in Table 3, the AMI model is initially highly mismatched with PF-STAR, with a WER of 68.19%, which aligns with what is expected from the literature [19]. For reference we include a model trained from scratch on PF-STAR, which obtains 20.46% WER. Adapting the SincConv layer of the AMI model for a single epoch to the training set of PF-STAR reduces the error rate to 31.65%. We include numbers showing the effect of updating the statistics of the batchnorm layers. Freezing the batchnorm layers demonstrates that the primary improvement comes from adapting the 80 param-
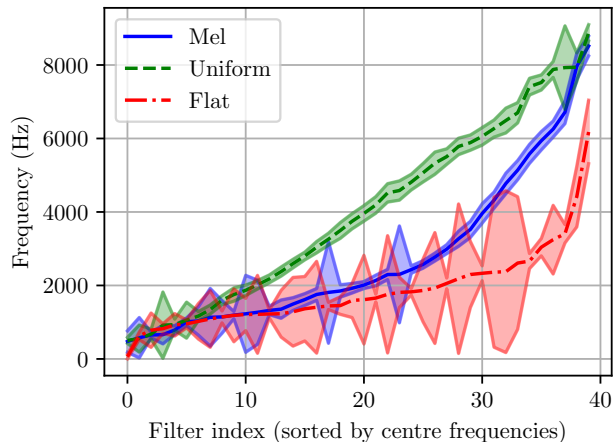

Fig. 3. Upper and lower learned frequencies per filter with different initialisation schemes, after six epochs of training on AMI. In contrast to Mel and Uniform, Flat is forced to change in order to extract different information in each filter.
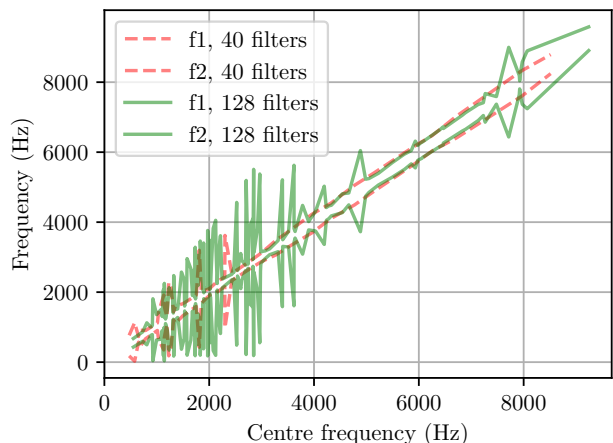

Fig. 4. Cut-off frequencies per filter for filterbanks with 40-filters or 128-filters after training with a Mel-initialised filterbank, plotted against their respective centre frequencies for comparison. We observed similar WERs, with simply more noisy bandwidths, with an increased number of filters.

eters in the SincConv layer. We freeze the batchnorm layers in all experiments that follow. This experiment shows that we can effectively adapt a very small number of parameters in the model, improving the out-of-domain model by over 50% relative, and coming within 12 percentage points of a model trained from scratch with all 9M parameters. Adapting the SincConv layer amounts to adapting less than 0.0009% of the total number of parameters in the model (see Table 1).

Figure 5 shows that adapting the SincConv layer shifts the upper frequency distribution of the filters, and their bandwidths. This is reflected in the corresponding VTLN function. This suggests that the model has adjusted to higher fre-
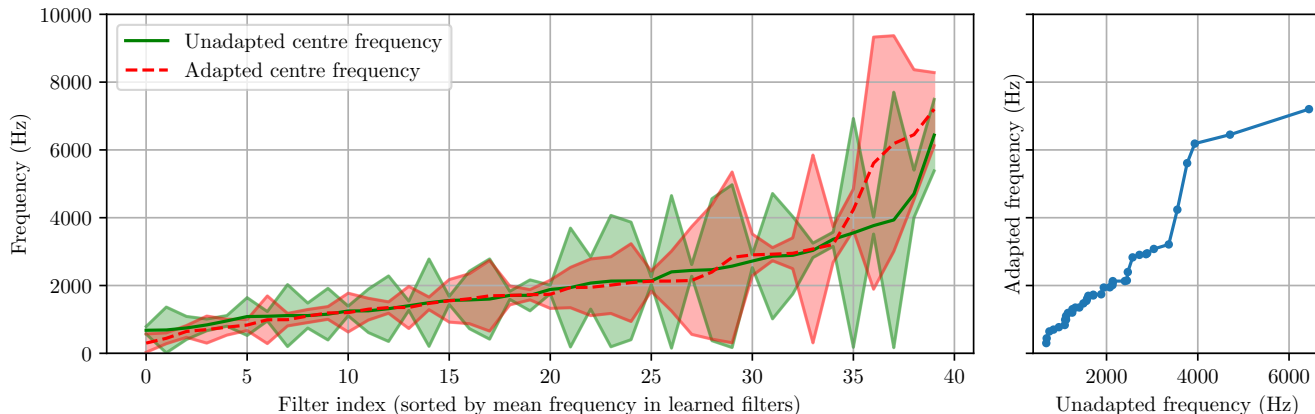
Fig. 5. (Left) SincNet filters before and after domain adaptation to PF-STAR. (Right) SincNet centre frequency adapted vs unadapted (e.g. VTLN-like function).

Model	WER (%)
AMI	68.19
PF-STAR	20.46
Adapt+batchnorm	29.90
Adapt−batchnorm	31.65

Table 3. Results (% WER) on the PF-STAR test set, after adapting the AMI model for one epoch. A PF-STAR model is shown for reference. The models have 40 filters that were originally initialised flat.

quency content in the children's speech data. Figure 6 shows average power spectra from a the corresponding log-mel filterbank features of AMI and PF-STAR which supports this notion.

We note that the VTLN-like function is nearly piecewise-linear; i.e. similar to the assumptions made during typical use of VTLN. However, it was here obtained through backpropagation instead of grid-search or other methods.

Table 4 demonstrates the effect of the number of adaptation utterances. As the amount of data increases adapting all parameters (excluding SincConv) produces lower error rates, as should be expected. The models begin to diverge at about three utterances (roughly 1 minute for PF-STAR).

Adapt/utts	0	1	2	3	20
Sinc	68.19	56.67	47.87	40.36	31.06
All–Sinc	68.19	56.70	47.42	35.89	21.34

Table 4. Results (% WER) on the PF-STAR test set given number of adaptation utterances when adapting from the AMI model. As the amount of data increases, adapting all parameters surpass only adapting the SincConv layer.
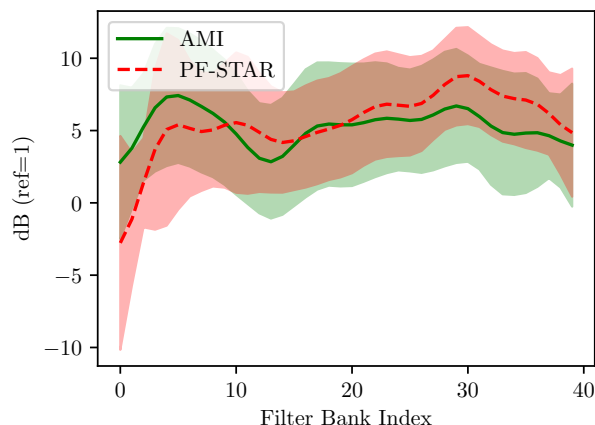

Fig. 6. Average log-mel filterbank spectra for random subsets of AMI and PF-STAR data. The shaded region denotes mean plus-minus one standard deviation.

4.2. Speaker adaptation

A more realistic, practical scenario, is to adapt to a few utterances obtained per speaker. In these experiments we adapt from the AMI model to 12 individual speakers in PF-STAR's eval/adapt set, testing on the corresponding speakers in eval/test. The results are shown in Figure 7 which shows the evolution with the number of epochs of adaptation. `LHUC0` indicates using LHUC on the output of the SincConv layer (40 parameters), and `LHUC1` is LHUC on the output of the first CNN layer (800 parameters). We use a learning rate of 0.8 for `LHUC0` and `LHUC1`, as LHUC can characteristically use very large learning rates without overfitting [2, 4]. When using `LHUC1` in combination with SincConv, we use the standard 0.0015 learning rate for SincConv, but a multiplier of 500 for LHUC. For `Sinc+LHUC0` we did not find this beneficial and used the same learning rate for both.
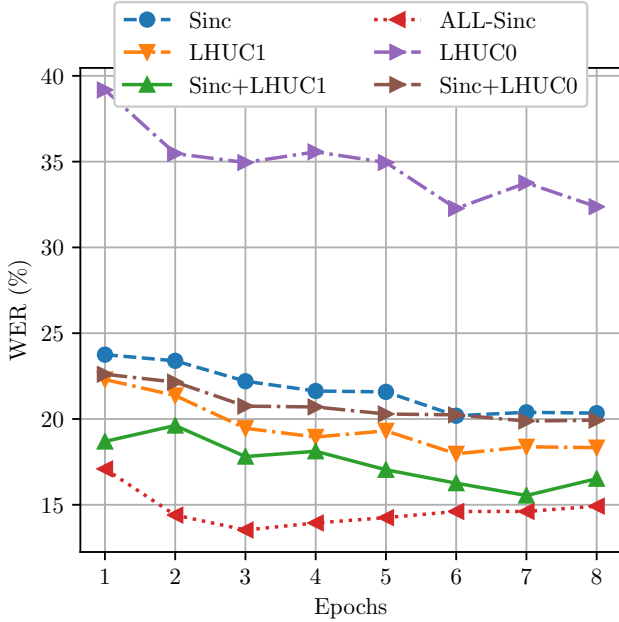
Fig. 7. Speaker adaptation over epochs with various techniques. The unadapted model obtains 59.06% WER.

The un-adapted WER is 59.06%. Adapting the 80 parameters of the SincConv layer yields only slightly worse results than `LHUC1` with 10 times fewer parameters. Interestingly, the two are complementary, as demonstrated by `Sinc+LHUC1`, and at best produces WERs similar to adapting all 9M parameters (excluding SincConv). `ALL-Sinc` is more sensitive to overfitting as evident from the figure.

Adapting the gain of the filters improves substantially over the unadapted model, but does not provide similar performance to any of the other approaches. One factor may be that the these parameters were fixed during the training of the baseline model as in [10], hence the rest of the network may have compensated by other means. It is, however, complementary with adapting the filterbank frequencies, with `Sinc+LHUC0` slightly outperforming `Sinc`. A summary of the results after adapting for eight epochs is shown in Table 5.

Figure 8 shows VTLN-like functions obtained from the adapted SincConv layer to each speaker. There is a clear difference between each function, which is in line with what one might expect given the variability of the acoustics of children's data [19].

5. CONCLUSIONS

We have shown that adapting the filterbank frequencies from raw waveforms with SincNet is extremely parameter efficient, obtaining substantial improvements in WERs with a fraction of the total model parameters on a children's speaker adaptation task. It is also complementary with the standard

Method	WER (%)	Params
Unadapted	59.06	-
Sinc	20.34	80
LHUC0	32.37	40
Sinc+LHUC0	19.93	120
LHUC1	18.33	800
Sinc+LHUC1	16.52	880
ALL-Sinc	14.92	$\sim$ 9M

Table 5. Results (% WER) adapting from AMI to individual speakers in PF-STAR for 8 epochs.
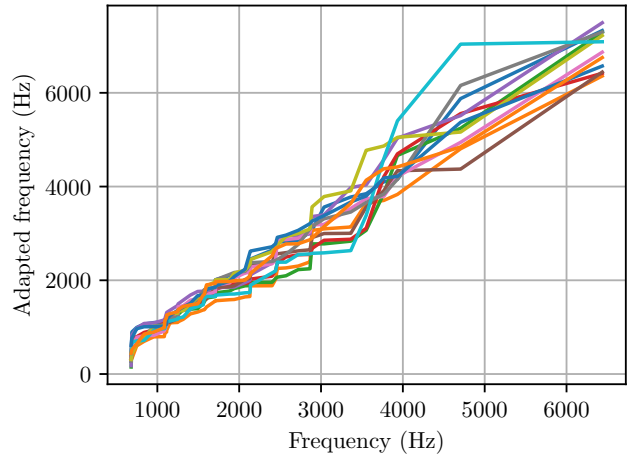

Fig. 8. Corresponding VTLN scaling functions for individual speakers (colours) adapted with `Sinc`. The majority of the scaling occurs in the higher frequencies.

LHUC technique, producing results similar to adapting all 9 million model parameters (excluding the filterbank layer). We also show that the parameterisation of SincNet affords interpretability during adaptation: during domain adaptation to children's speech, the layer learns to pay more attention to higher frequencies. Similarly for speaker adaptation, the change in the filter frequencies effectively resembles VTLN, producing individual scaling functions for each speaker. Finally, we noted that adapting the gain is related to LHUC and fMLLR, and this proved complementary to adapting the filterbank frequencies.

In future work we would like to explore the use of meta-learning (as in [4]) to learn filter-specific learning rates, as well as experimenting with unsupervised, test-time adaptation of the SincConv layer, and exploring the layer's response to noise. Finally, we would like to experiment with the inclusion of feature-based adaptation methods such as i-vectors [38], which has previously been shown to be complementary to model-based adaptation [39].

6. REFERENCES

[1] Wayne Xiong, Lingfeng Wu, Fil Alleva, Jasha Droppo, Xuedong Huang, and Andreas Stolcke, "The Microsoft 2017 conversational speech recognition system," in *ICASSP*, 2018, pp. 5934–5938.

[2] Pawel Swietojanski, Jinyu Li, and Steve Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *TASLP*, vol. 24, no. 8, pp. 1450–1463, 2016.

[3] Mark JF Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[4] Ondřej Klejch, Joachim Fainberg, and Peter Bell, "Learning to adapt: A meta-learning approach for speaker adaptation," in *Proc. Interspeech*, 2018, pp. 867–871.

[5] Khe Chai Sim, Arun Narayanan, Ananya Misra, Anshuman Tripathi, Golan Pundak, Tara Sainath, Parisa Haghani, Bo Li, and Michiel Bacchiani, "Domain adaptation using factorized hidden layer for robust automatic speech recognition," in *Proc. Interspeech*, 2018, pp. 892–896.

[6] Joachim Fainberg, Steve Renals, and Peter Bell, "Factorised representations for neural network adaptation to diverse acoustic environments," in *Proc. Interspeech*, 2017, pp. 749–753.

[7] Li Lee and Richard C. Rose, "Speaker normalization using efficient frequency warping procedures," in *ICASSP*, 1996, vol. 1, pp. 353–356.

[8] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proc. Interspeech*, 2014.

[9] Tara N. Sainath, Ron J. Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Interspeech*, 2015, pp. 1–5.

[10] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with SincNet," in *SLT*, 2018, pp. 1021–1028.

[11] Ryu Takeda, Kazuhiro Nakadai, and Kazunori Komatani, "Multi-timescale feature-extraction architecture of deep neural networks for acoustic model training from raw speech signal," in *IROS*, 2018, pp. 2503–2510.

[12] Tara N. Sainath, Brian Kingsbury, Abdel-rahman Mohamed, and Bhuvana Ramabhadran, "Learning filter banks within a deep neural network framework," in *ASRU*, 2013, pp. 297–302.

[13] Navdeep Jaitly and Geoffrey Hinton, "Learning a better representation of speech soundwaves using restricted Boltzmann machines," in *ICASSP*, 2011, pp. 5884–5887.

[14] Dimitri Palaz, Mathew Magimai-Doss, and Ronan Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," in *Proc. Interspeech*, 2015, pp. 11–15.

[15] Yedid Hoshen, Ron J. Weiss, and Kevin W Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *ICASSP*, 2015, pp. 4624–4628.

[16] Pegah Ghahremani, Hossein Hadian, Hang Lv, Daniel Povey, and Sanjeev Khudanpur, "Acoustic modeling from frequency domain representations of speech," in *Proc. Interspeech*, 2018, pp. 1596–1600.

[17] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn, "Understanding how deep belief networks perform acoustic modelling," in *ICASSP*, 2012, pp. 4273–4276.

[18] Pawel Swietojanski and Steve Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*, 2014, pp. 171–176.

[19] Alexandros Potamianos and Shrikanth Narayanan, "Robust recognition of children's speech," *IEEE Transactions on speech and audio processing*, vol. 11, no. 6, pp. 603–616, 2003.

[20] Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa, "A deep neural network integrated with filterbank learning for speech recognition," in *ICASSP*, 2017, pp. 5480–5484.

[21] Hiroshi Seki, Kazumasa Yamamoto, Tomoyosi Akiba, and Seiichi Nakagawa, "Rapid speaker adaptation of neural network based filterbank layer for automatic speech recognition," in *SLT*, 2018, pp. 574–580.

[22] Hardik B. Sailor and Hemant A. Patil, "Novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition," *TASLP*, vol. 24, no. 12, pp. 2341–2353, 2016.

[23] Alan V Oppenheim and Ronald W Schafer, "Digital signal processing," *Prentice-Hall*, vol. 19752, pp. 26–30, 1975.

[24] Sankaran Panchapagesan and Abeer Alwan, "Multi-parameter frequency warping for VTLN by gradient search," in *ICASSP*, 2006, vol. 1, pp. I–I.

[25] Jean Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus," *LREC*, vol. 41, no. 2, pp. 181–190, 2007.

[26] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," Tech. Rep., IEEE Signal Processing Society, 2011.

[27] Anton Batliner, Mats Blomberg, Shona D'Arcy, Daniel Elenius, Diego Giuliani, Matteo Gerosa, Christian Hacker, Martin Russell, Stefan Steidl, and Michael Wong, "The PF_STAR childrens speech corpus," in *Proc. Interspeech*, 2005, pp. 2761–2764.

[28] Joachim Fainberg, Peter Bell, Mike Lincoln, and Steve Renals, "Improving children's speech recognition through out-of-domain data augmentation," in *Proc. Interspeech*, 2016, pp. 1598–1602.

[29] Erfan Loweimi, Peter Bell, and Steve Renals, "On learning interpretable CNNs with parametric modulated kernel-based filters," in *Proc. Interspeech*, 2019.

[30] Douglas O'Shaughnessy, *Speech Communication: Human and Machine*, Universities press, 1987.

[31] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[32] François Chollet et al., "Keras," `https://github.com/keras-team/keras`, 2015.

[33] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: A system for large-scale machine learning," in *OSDI*, 2016, pp. 265–283.

[34] Christopher Cieri, David Miller, and Kevin Walker, "The Fisher Corpus: a resource for the next generations of speech-to-text," in *LREC*, 2004, vol. 4, pp. 69–71.

[35] S Pavankumar Dubagunta, Selen Hande Kabil, and Mathew Magimai Doss, "Improving children speech recognition through feature learning from raw speech signal," in *ICASSP*, 2019, pp. 5736–5740.

[36] Andreas Stolcke, "SRILM-an extensible language modeling toolkit," in *ICSLP*, 2002.

[37] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015.

[38] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2013, pp. 55–59.

[39] Lahiru Samarakoon and Khe Chai Sim, "On combining i-vectors and discriminative adaptation methods for unsupervised speaker normalization in DNN acoustic models," in *ICASSP*, 2016, pp. 5275–5279.