EUROREGIO BNAM2022
Joint Acoustics Conference
9th-11th of May, Aalborg, Denmark

A parameter-conditional neural network framework for modelling parameterized auditory models

Peter Asbjørn Leer Bysted [1,3,*] **Jesper Jensen**[2,3], **Zheng-Hua Tan**[3], **Jan Østergaard**[3], **Lars Bramsløw**[1]

[1]Eriksholm Research Centre, 3070 Snekkersten, Denmark.
[2]Oticon A/S, 2765 Smørum, Denmark.
[3]Department of Electronic Systems, Aalborg University, 9000 Aalborg, Denmark
[*]pelb@eriksholm.com

Abstract

The use of auditory models is important for designing speech and audio processing algorithms for hearing assistive devices. These auditory models are often parameterized by a set of parameters relating to auditory function, e.g., hair cell loss or synaptopathy. In practice, the computational load of these auditory models can be very high thus limiting the feasibility of using the models as bio-inspired loss functions for deep learning based hearing loss compensation strategies or denoising strategies. Previous efforts have addressed this problem by training a neural network for each parameter configuration of the auditory model which greatly reduces the computation time of the auditory model but requires a new network to be trained whenever the parameterization changes. In this paper we propose an approach where a single neural network is trained, once and for all, to accurately simulate auditory models across their parameter spaces by conditioning the weights of the network on the parameters of the respective auditory models. This approach enables greater flexibility than training a single model for each parameter configuration, as any parameterization can be acquired on the fly. The accuracy of the neural network is shown to be robust across both unseen inputs and standard audiograms.

Keywords: Auditory models, deep learning, neural networks, computational modelling

1 Introduction

The mammalian auditory system is a complex, dynamic and non-linear system which processes and extracts information from acoustic signals. In order to gain understanding and predictive capability of the auditory system, there has for several decades been undertaken a large effort to model the experimental data. Traditionally the auditory system is broken down into smaller sub-systems corresponding to a functional or anatomical sub-structure, e.g. the cochlea or the auditory nerve. These models are themselves often complex and might be computationally expensive. The models are therefore limited in their use-cases for real-time signal processing or deep learning applications. Previous approaches have solved this problem by training a neural network, here denoted as an Auditory Model Simulator (AMS), to simulate the input-output relations of the sub-systems of the auditory systems [1][2], which can simulate the auditory models in real-time. The drawback of this approach is that an individual AMS has to be trained for each parameter configuration of a given auditory model. To circumvent this problem, we present a framework that can simulate the inner representation of the auditory model, by extending the previous described approach to include a Weight Generating Network (WGN), a neural network that generates the weights for the fixed structure of the AMS conditioned on the parameter space of the auditory model.

2 Methods

In this section the overall framework and the different components of the proposed framework will be introduced, including the chosen example auditory model, the architecture of the AMS and WGN, the training procedure and the parameters of the networks.

2.1. Modelling framework

We denote the auditory model, as $f_\theta : X \rightarrow I$, with X being the signal space, I the inner representation space of the model and θ the parameters of the model. We approximate this model by the structure shown in Figure 1. We generate the parameters for the AMS from the WGN: $\Theta \rightarrow W$, where Θ is the weight space of the model, and W is the parameter space of the AMS that approximates f_θ, which we denote as $\hat{f}_\theta$. There are thus two disjoint paths in the framework, each of which is represented by a neural network: The signal path (represented by the AMS) and the parameter path (represented by the WGN). By decoupling the audio and the audiogram inputs through two networks, the complexity at inference is smaller than training one large network, as the functional relationship between the auditory models and audiogram is disjoint from the audio processing path.

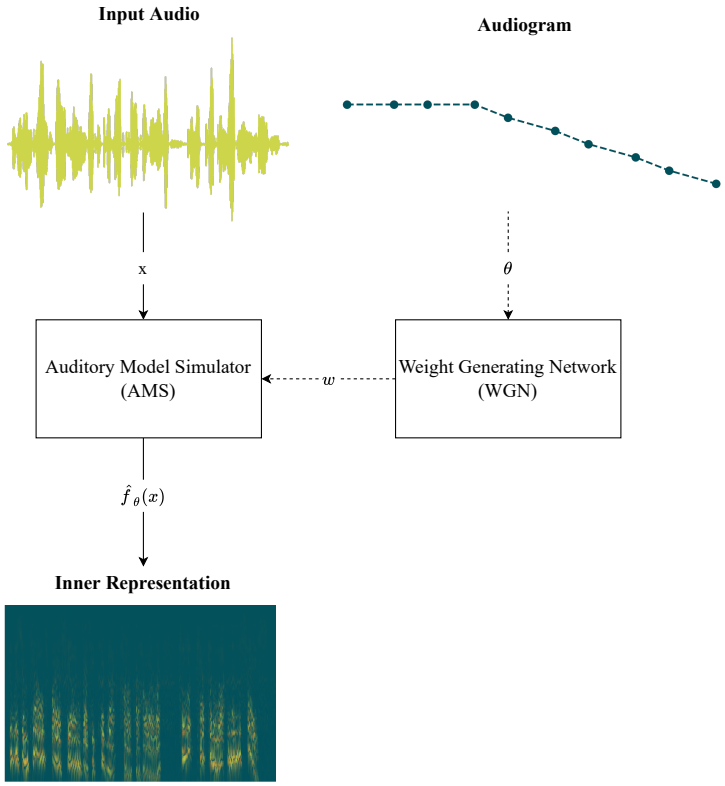

Figure 1: An acoustic input signal, x, is passed through the Auditory Model Simulator (AMS). The weights of the neural network are generated by the Weight Generating Network (WGN), that produces the neural network weights based on model parameters, θ. The bold lines denote the signal path, and the dashed lines denote the parameter path. The result is a model inner representation.

2.2. Auditory model

In this work we consider the Inner Hair Cell (IHC) stage of the UR EAR 2020b model[3][4] as an exemplary model. The model is illustrated in Figure 2. The model consists of K parallel non-linear auditory filters modelling the IHC transduction induced by the movement of the basilar membrane, with filters placed successively along the basilar membrane. The center frequencies of the filters are logartihmically distributed from 125 to 8000 Hz. For each filter, there are 2 parameters related to hearing loss, c_{OHC} and c_{IHC}, relating to outer hair cell function and inner hair cell function, respectively. The model is supplied with a function that for any given audiogram computes the full set of appropriate model parameters.

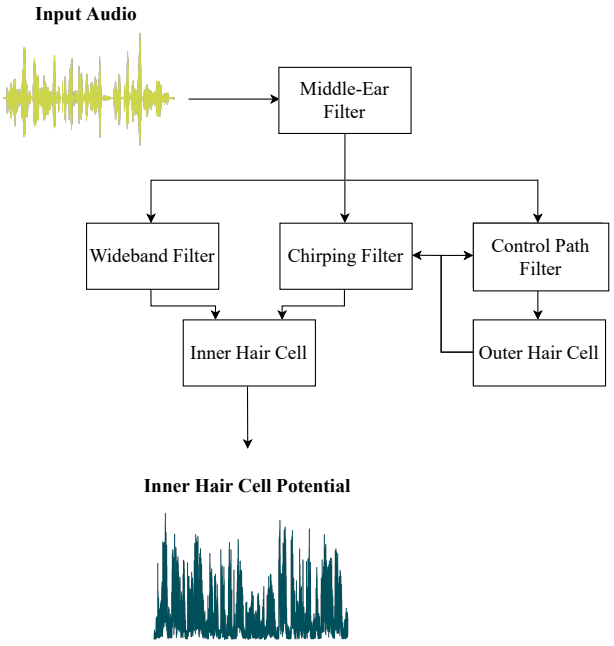

Figure 2: Example auditory model [3][4].

2.3. Auditory Model Simulator

For the AMS we use the Wave-U-Net structure [5]. The structure is a U-shaped convolutional auto-encoder, that for each layer performs a convolution, followed by downsampling or upsampling and a non-linear activation function. The network is subdivided into blocks according to their function in the network: The input block, downsampling block, embedding block, upsampling block and output block. All blocks except for the input block include a 1D-convolution operation parameterized by the WGN, resulting in a total of 2N+2 1D-convolution operations, where N is the number of downsampling blocks. The architecture is illustrated in Figure 3. The Wave-U-Net is very similar to previously used models for simulating cochlea respones [1]. In the previous work the structure was found appropriate for simulating different hearing losses, and different models, such as models of the cochlea, the inner hair cells and the auditory nerve, motivating this choice of network.

2.3.1 Input Block

This block resamples the input to 20 kHz and crops the input such that the input length is a multiple of 2^N. This ensures that the length of the output of the network and skip connections are consistent with the input.

2.3.2 Downsampling Block

The downsampling block is a 1-dimensional convolution block, i.e. convolution over time and an activation function, followed by decimating the signal by a factor 2. Note, there is no anti-aliasing filters in this setup. Skip connections are connected to the upsampling blocks, which might help restore the phase information that might be lost during down-sampling and improve gradient flow during training.

2.3.3 Embedding block

The embedding block consists of a convolutional layer and an activation function. The parameters of the convolution is the same as of the downsampling block, but there is no skip connection and downsampling.

2.3.4 Upsampling Block

The upsampling block is structured inversely to the downsampling block. The input is upsampled by linear interpolation, followed by a concatenation with the skip connection from the downsampling block and a convolution block.

2.3.5 Output Block

The output block concatenates the input and performs a point wise convolution with the last upsampling layer and the input. The inner representation is the output of this layer.

2.4. Weight Generating Network

The WGN generates the neural weights for the AMS by a linear combination of parameter tensors. The WGN is illustrated in Figure 4. Denote the tensors containing the weights for the nth layer in the AMS by $\mathbf{w}_n$, then the WGN generates $\mathbf{w}_n$ as a function of audiogram parameters θ, by a linear combination of K tensors:

$$\mathbf{w}_n = \sum_{k=1}^{K} \alpha_{n,k} \mathbf{w}_{n,k} \tag{1}$$

The $\alpha_{n,k}$ are generated by the WGN, by feeding the model parameters found from the audiogram through a 3-layer Multi Layer Perceptron (MLP), e.g. a fully connected feedforward network with 3 layers. The output of the MLP has dimensions $(K(2N+2))$, and is reshaped into a matrix of dimension $(2N+2, K)$. This matrix is split into 2N+2 K-dimensional vectors, and the softmax function is computed across the K elements in each vector, outputting $0 \leq \alpha_{n,k} \leq 1$, which are used as in Equation (1). Note that the method is similar to the Dynamical Convolution approach in [6] and Conditional Convolution approach in [7], the difference being that these approaches generate the weights by conditioning on the input and therefore solves a different problem.

2.5. Training

In order to train the network, a dataset is created by generating 7500 input-output pairs, $\{T \coloneqq (x, f_\theta(x)) | x \in X, \theta \in \Theta\}$, where X is 7500 random sentences from the LibriTTS [8] database, and Θ is generated by sampling from 10 different standard audiograms [9] and multiplying the audiograms by a random scalar between 0.5 and 1. The audiograms are defined for 10 frequency bands and are interpolated linearly on a log/dB scale for center frequencies between these frequency bands. For the results presented here we use J=90 center frequencies, or frequency channels, of the auditory model. For all audiograms we contribute 2/3 of the threshold shift to the outer hair cells (OHC) and 1/3 to the IHC. The inputs are normalized to 80 dB SPL, and the input-output pairs

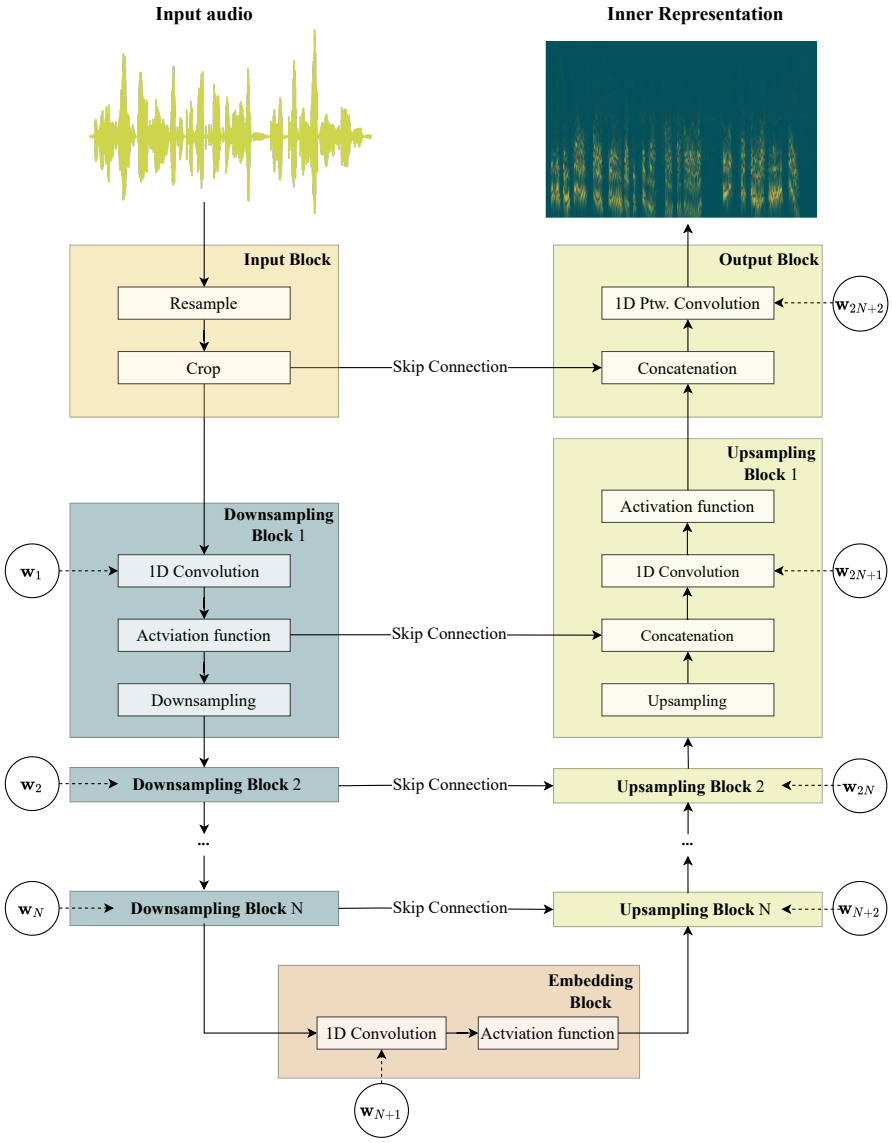

Figure 3: Overview of the auditory network simulator. The bold lines denote the signal path and the dashed lines denote the parameter path. The circles denote the weights generated by the Weight Generating Network (WGN).

are sampled at 20 kHz and segmented into windows of 2048 samples with 256 samples of temporal context on each side of the window. The AMS and WGN are trained jointly with respect to a scaled L1-loss function:

$$L(f_\theta(x), \hat{f}_\theta(x)) = \sum_{j=1}^{J} ||\beta_j f_{\theta,j}(x) - \hat{f}_{\theta,j}(x)||_1 \tag{2}$$

where j is the j-th channel of the inner representation and

$$\beta_j = \frac{1}{|T|} \sum_{(x, f_\theta(x)) \in T} \frac{1}{||f_{\theta,j}(x)||_1} \tag{3}$$

The weighting term β_j allows better optimization of the higher frequency channels, where the absolute energy is orders of magnitude lower than the lower frequency channels which affects the integration of the gradient

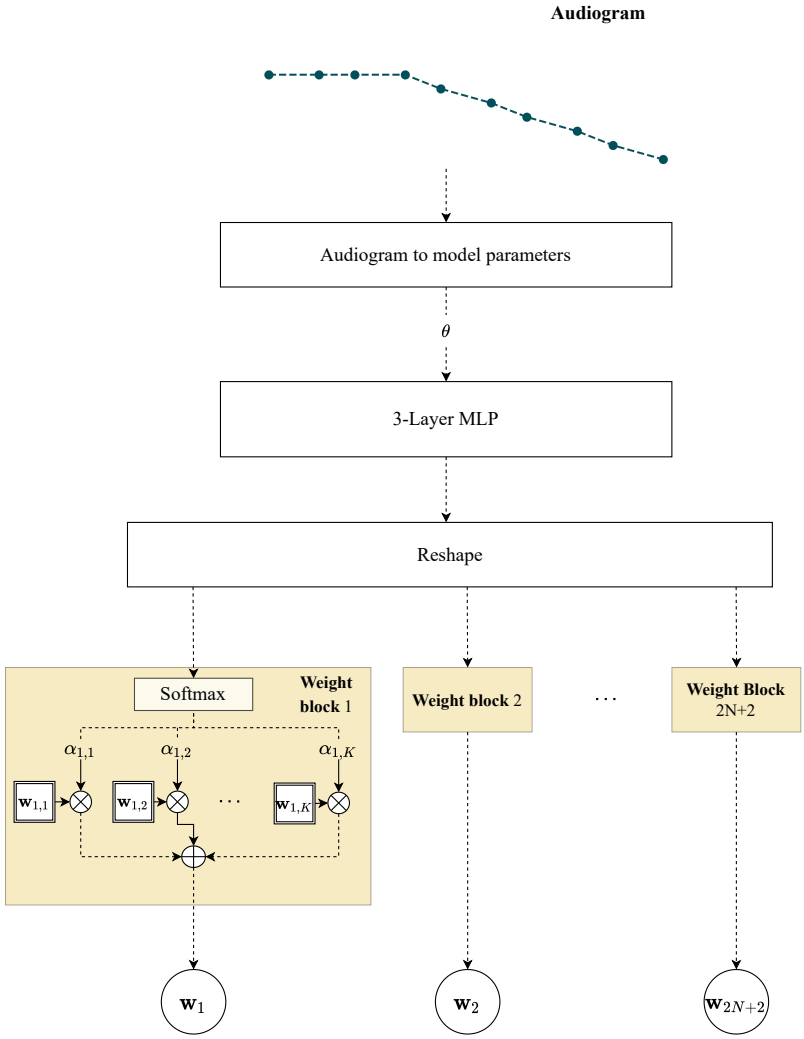

Figure 4: Overview of the Weight Generating Network. The outputs of this network is used as parameter inputs for the Auditory Model Simulator

with respect to the different channels during training. This weighting is especially important for simulating sloping hearing losses, since the shape of a conventional hearing loss exacerbates the effect. During inference, $\hat{f}_{\theta,j}$ is multiplied by the reciprocal of β_j. The ADAM [10] optimizer is used together with back propagation and gradient descent using a batch size of 256 and a learning rate of 0.0001. The network was trained for 75 epochs.

2.6. Model parameters

For the results in this paper we use the AMS and WGN structures given in Tables 1 and 2.

Table 1: Structure of the Auditory Model Simulator

N	8
Kernel size	21
Depth	160
Encoder activation	Tanh
Decoder activation	PReLU

Table 2: Structure of the Weight Generating Network

Layer 1 of MLP	(180×100)
Layer 2 of MLP	(100×100)
Layer 3 of MLP	(100×48)
Activation of MLP	Tanh
K	3

3 Results

In this section we will present the results in three different ways: (i) Through visualization of the inner representations, (ii) a comparison of the error between the training set and test set, and (iii) an error measure of the model across different audiograms and levels.

3.1. Visualizing the results

In Figure 5 we illustrate the inner representation of the ground truth (top row), our simulation (middle row) and the difference between the two (bottom row), for three different audiograms, for the same speech signal, x. From the figure it can be seen that the model fairly well approximates the general characteristics of the reference model, although discrepancies still exists.

3.2. Generalization to unseen speech

In order to test if the model can generalize to unseen speech, we compare the Mean Absolute Error (MAE) on a set of 100 audiograms and sentences of speech that was in the original training dataset with a testset consisting of the same audiograms but different sentences, with the speech scaled to the same level of the training set. The results are shown in Table 3. The test error is 7% larger than the training error, suggesting that the model is quite robust to unseen input signals.

Table 3: Training and test Mean Absolute Error (MAE) for the model, computed across different audiograms

	Train	Test
MAE	0.00081	0.00087

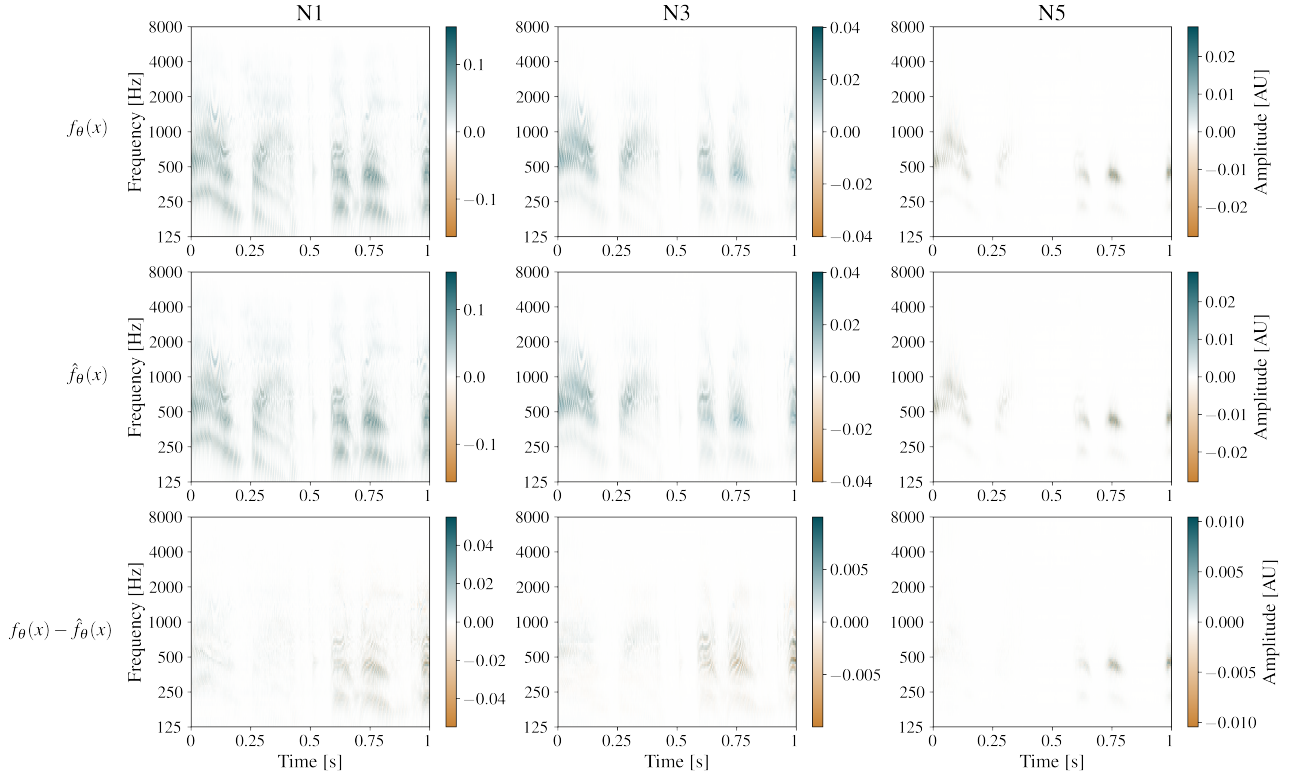

Figure 5: An example of the outputs of the Auditory Model Simulator conditioned on three different standard audiograms (N1,N3,N5) [9]. The top row is the ground truth, the middle row is the output of the AMS and the bottom row is the difference between the ground truth and our approach. The input is a speech signal that was not part of the training set, scaled to have an RMS value of 80 dB SPL. Notice different scales across audiograms and the bottom row to enhance visibility.

3.3. Testing error across audiograms and levels

In order to quantify the performance across audiograms and speech levels, we measure the error on the 10 standard audiograms [9] across 10 sentences that were not in the training set. We scale the absolute error, so that the errors are comparable across different levels and audiograms by introducing a Relative Mean Absolute Error (RMAE):

$$RMAE = \frac{|f_\theta(x) - \hat{f}_\theta(x)|_1}{|f_\theta(x)|_1} \tag{4}$$

From Table 4 it is clear that the error becomes large at low SPL levels relative to the level which was used to train the networks for audiograms with severe/profound broadband hearing losses, i.e N5-N7, meanwhile the error stays relative constant across different audiograms for the SPL levels on which the networks were trained. We note that N6 and N7 have identical errors, which is due to the parameters of the auditory model being identical for these two audiograms. It can be seen that the error is relatively large for the 60 dB SPL for the N5-N7 audiograms. Upon inspecting the output, we observe that the waveform looks to be the correct but has a large DC offset. This DC offset disappears for the 70 and 80 dB SPL condition. The DC offset could be due to low amplitudes and energy relative to the other audiograms and input levels resulting in a small effect on the gradient of the weights, and in particular the bias units, during training of the network and a higher sensitivity to small perturbations of the weights during inference. The significance of this error will largely depend on the application. We hypothesize that by using an architecture that discards the bias units, extending the training set to include additional input levels and modifying the loss function to accommodate the additional input levels,

that one can alleviate the problem of the DC offset.

Table 4: Relative Mean Absolute Error plus/minus the standard deviation for the 10 standard audiograms [9], measured across the same 10 sentences for each audiogram

Audiogram/Level dB[SPL]	60 dB	70 dB	80 dB
N1	0.23 ± 0.04	0.20 ± 0.03	0.17 ± 0.02
N2	0.21 ± 0.04	0.18 ± 0.03	0.16 ± 0.03
N3	0.21 ± 0.03	0.19 ± 0.03	0.19 ± 0.02
N4	0.36 ± 0.04	0.32 ± 0.03	0.31 ± 0.02
N5	1.7 ± 0.14	0.39 ± 0.04	0.21 ± 0.04
N6	1.84 ± 0.16	0.39 ± 0.04	0.21 ± 0.04
N7	1.84 ± 0.16	0.39 ± 0.04	0.21 ± 0.04
S1	0.23 ± 0.04	0.20 ± 0.03	0.17 ± 0.02
S2	0.20 ± 0.04	0.18 ± 0.03	0.16 ± 0.03
S3	0.27 ± 0.02	0.24 ± 0.02	0.23 ± 0.02

4 Conclusion

In this work we propose a neural network modelling framework that simulates an auditory model across its parameter space by processing the input and the parameters disjointly through two neural networks. The networks are trained on a set of speech and audiograms, and the model is shown to generalize well to unseen speech inputs. The error of the model is measured across a set of standard audiograms and input level, and is shown to have robust performance, except for a few cases where the model had a significant DC offset. Further work could investigate architectural improvements, parameter tuning and optimization techniques for this kind of model.

5 Acknowledgements

This work is partly supported by Innovation Fund Denmark Case no. 0153-00091B

References

[1] Deepak Baby, Arthur Van Den Broucke, and Sarah Verhulst. A convolutional neural-network model of human cochlear mechanics and filter tuning for real-time applications. *Nature machine intelligence*, 3(2):134, 2 2021. doi: 10.1038/S42256-020-00286-8. URL /pmc/articles/PMC7116797//pmc/articles/PMC7116797/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC7116797/.

[2] Anil Nagathil, Florian Göbel, Alexandru Nelus, and Ian C. Bruce. Computationally efficient DNN-based approximation of an auditory model for applications in speech processing. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2021-June:301–305, 2021. ISSN 15206149. doi: 10.1109/ICASSP39728.2021.9413993.

[3] Muhammad S. A. Zilany, Ian C. Bruce, and Laurel H. Carney. Updated parameters and expanded simulation options for a model of the auditory periphery. *The Journal of the Acoustical Society of America*, 2014. ISSN 0001-4966. doi: 10.1121/1.4837815.

[4] Auditory Models - Publications - Carney Lab - University of Rochester Medical Center. URL https://www.urmc.rochester.edu/labs/carney/publications-code/auditory-models.aspx.

[5] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-U-Net: A multi-scale neural network for end-to-end audio source separation. *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 334–340, 2018. doi: 10.5281/zenodo.1492417.

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic Convolution: Attention over Convolution Kernels. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11027–11036, 12 2019. ISSN 10636919. doi: 10.1109/CVPR42600.2020.01104. URL https://arxiv.org/abs/1912.03458v2.

[7] Brandon Yang, Google Brain, Gabriel Bender, Quoc V Le Google Brain, and Jiquan Ngiam. CondConv: Conditionally Parameterized Convolutions for Efficient Inference. URL https://github.com/tensorflow/tpu/tree/master/.

[8] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019-September:1526–1530, 4 2019. ISSN 19909772. doi: 10.21437/Interspeech.2019-2441. URL https://arxiv.org/abs/1904.02882v1.

[9] Nikolai Bisgaard, Marcel S M G Vlaming, and Martin Dahlquist. Standard Audiograms for the IEC 60118-15 Measurement Procedure. *Trends in Amplification*, 14(2):113–120. doi: 10.1177/1084713810379609. URL http://tia.sagepub.com.

[10] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 12 2014. URL https://arxiv.org/abs/1412.6980v9.