Article

Low-order Spherical Harmonic HRTF Restoration using a Neural Network Approach

Benjamin Tsui [1,*], William A. P. Smith [2] and Gavin Kearney [1]

[1] AudioLab, Communications Technologies Research Group, Department of Electronic Engineering, University of York, UK; gavin.kearney@york.ac.uk
[2] Computer Vision and Pattern Recognition (CVPR) research group in the Department of Computer Science, University of York, UK; william.smith@york.ac.uk
[*] Correspondence: bt712@york.ac.uk

Version July 10, 2020 submitted to Journal Not Specified

Abstract: Spherical harmonic (SH) interpolation is a commonly used method to spatially up-sample sparse Head Related Transfer Function (HRTF) datasets to denser HRTF datasets. However, depending on the number of sparse HRTF measurements and SH order, this process can introduce distortions in high frequency representation of the HRTFs. This paper investigates whether it is possible to restore some of the distorted high frequency HRTF components using machine learning algorithms. A combination of Convolutional Auto-Encoder (CAE) and Denoising Auto-Encoder (DAE) models is proposed to restore the high frequency distortion in SH interpolated HRTFs. Results are evaluated using both Perceptual Spectral Difference (PSD) and localisation prediction models, both of which demonstrate significant improvement after the restoration process.

Keywords: Deep learning; Head Related Transfer Function (HRTF); Restoration; Ambisonics; Spatial Audio; Spherical harmonic; Audio signal processing; Denoising; Auto-Encoder; Neural Network

1. Introduction

Virtual Reality (VR) and Augmented Reality (AR) technologies are on the rise, through the advent of commercially available and affordable VR headsets, with applications in gaming, education, therapy, social media and digital culture amongst others. To provide a high fidelity immersive experience in virtual environments requires good quality spatial audio. To achieve this, the VR/AR technology must be able to deliver to the ears the same binaural cues as would be experienced in real life. These are Interaural Time Difference (ITD), Interaural Level Difference (ILD) and spectral cues introduced by the ear pinnae and torso. Head Related Transfer Functions (HRTFs) are sets of binaural filters that encapsulate these cues from different angles relative to the head in three dimensions. Sound sources can be spatialised by direct convolution with the a given HRTF pair representing the intended sound source direction. Alternatively a virtual loudspeaker framework can be employed, where methods such as Vector Base Amplitude Panning (VBAP) [1] or Ambisonics [2] are used to render sources between virtual loudspeaker points formed from the HRTFs [3]. Both methods typically require a high number of HRTF measurements to ensure good spatial resolution in the rendered audio [4].

However, HRTFs are highly personalised due to different head and ear shapes. Using mismatched HRTFs can affect timbre quality, localisation performance and externalisation. Currently, the main way to obtain personalised HRTFs is through physical measurements, where microphones are placed at the ear canal of a subject and the loudspeakers positioned at different angles relative to head to measure the transfer functions [5]. This measurement process is often tedious and requires substantial setup and calibration. Recent developments have been made in HRTF based selection based on anthropomorphic data extracted from photographs of the ear [6] or HRTF simulation using 3D head models [7]. However, simulation is computationally expensive and usually requires a lot of processing time [8,9].

To simplify the measurement process, different HRTF interpolation methods have been proposed to acquire dense HRTF sets from sparse HRTF measurements. The current state of the art method is Spherical Harmonic (SH) interpolation, which leverages the spatial continuity in spherical harmonics (SH) and uses it as a bridge to spatially up-sample a sparse HRTF measurement set to a denser one. Depending on the number of sparse HRTF measurement points and SH order, high frequency information can be lost in this process. Consequently, listeners may perceive timbre difference and weakened localisation performance in practical use.

Recently, developments in machine learning have shown great improvement in neural style transfer and data restoration especially in the image domain [10,11]. This paper investigates whether similar models can be used to restore the distorted high frequency data in SH interpolated HRTFs.

This paper is organised as follows: Section 2 will cover relevant background information on HRTF interpolation methods, particularly in SH interpolation. It will also discuss the motivation for using a machine learning approach along with identification of some candidate models. Section 3 will discuss the method used in this study, including the data pre-processing workflow, a baseline model and different techniques investigated on top of the baseline model. Section 4 evaluates the performance of the model based on perceptual spectral difference and localisation performance. Section 5 discusses the results and potential directions for the work. Section 6 concludes the paper.

2. Spherical Harmonic HRTF interpolation

HRTF interpolation can be done in many different ways, including using inverse-distance weighting and spherical splines in the time or frequency domains or manifold learning [12–16]. However, SH interpolation is one of the more elegant methods which has a more standardised procedure and shows promising results [16].

HRTF sets are commonly described in the SH domain due to its simplicity and ease of use in Ambisonics for spatial audio reproduction [17]. Since the SH domain describes a continuous spatial representation of the HRTFs, interpolation can be readily achieved. A given HRTF H(`,Œ) can be converted to the spherical harmonic domain at a given order M by using a re-encoding matrix C with K rows and L columns, where K is the number of SH channels calculated as $K = (M+1)^2$ and L is the number of HRTF measurements from different angles, where $L \geq K$. The coefficients Y^{σ}_{mn} in the re-encoding matrix C with SH order m and degree n are calculated by

$$\begin{aligned} Y^{\sigma}_{mn}(\theta,\phi) = & \sqrt{(2-\delta_{n,0})\,\frac{(m-n)!}{(m+n)!}}P_{mn}(\sin\phi) \\ & \times \begin{cases} \cos(n\theta), \text{ if } \sigma = +1 \\ \sin(n\theta), \text{ if } \sigma = -1 \end{cases} \end{aligned} \tag{1}$$

where $\sigma = \pm 1$, $P_{mn}(\sin\phi)$ are the Legendre functions of order m and degree n, $\delta_{n,0}$ is the Kronecker delta function:

$$\delta_{n,0} \equiv \begin{cases} 1, \text{ for } n = 0 \\ 0, \text{ for } n \neq 0 \end{cases} \tag{2}$$

This paper uses Schmidt semi-normalisation (SN3D) in the computation of Y^{σ}_{mn}. For normal SH HRTF use, a mode matching decoding matrix D can be calculated from C with the following pseudo-inverse equation:

$$\mathrm{D} = \mathrm{C}^{-1} = \mathrm{C}^{T}\left(\mathrm{C}\mathrm{C}^{T}\right)^{-1} \tag{3}$$

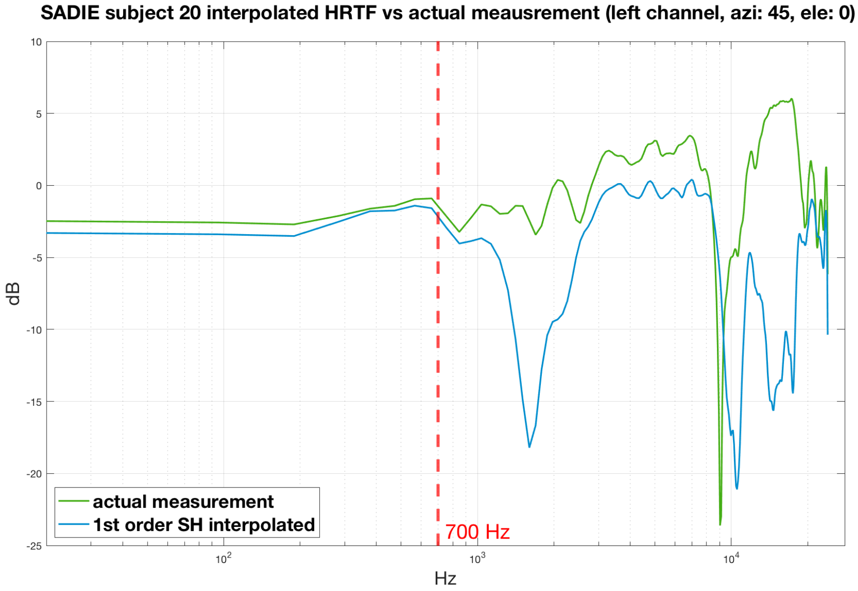


Figure 1. Actual ipsilateral HRTF measurement vs SH interpolated HRTF (green: Actual HRTF measurement, blue: SH interpolated HRTF), $M = 1$.

However, for HRTF interpolation, another re-encoding matrix $\hat{\mathrm{C}}$ and decoding matrix $\hat{\mathrm{D}}$ with the desired target angles is required, where $\hat{L}$ is replaced as the number of HRTF measurements to be interpolated from the SH HRTFs whilst K remains the same. The interpolated HRTFs $\hat{\mathrm{H}}$(`,Œ) can be calculated with the following equation:

$$\hat{\mathrm{H}} = \hat{\mathrm{D}}(\mathrm{C}(\mathrm{H})) \tag{4}$$

The main issue caused by SH HRTF interpolation is that the interpolated HRTFs are only accurate up to the spatial aliasing frequency f_{lim}, approximated by

$$f_{lim} \approx \frac{cM}{4r(M+1)\sin(\pi/(2M+2))} \approx \frac{cM}{2\pi r} \tag{5}$$

where c is the speed of sound, approximated as 343 m/s at 20°C in air and r is the radius of the human head [18]. For 1st order, the spatial aliasing frequency is around 700Hz.

The spectral distortions will not only affect the timbre, but will also degrade the localisation performance since the important cues for identification of source elevation are changed, as shown in Figure 1.

To challenge the full potential of the use of machine learning, this paper chooses to use 1st order SH interpolation as it requires the least number of HRTF measurements. An octahedron configuration with 6 measurements is selected, which has a more stable energy distribution than other arrays for 1st order SH [19]. When using SH interpolation, the number of outputs can be flexible.

We also employ dual-band Time Alignment (TA) in the encoding of the HRTFs [20,21]. Given ITD is only effective at low frequencies, high frequency ITD can be removed when undertaking SH encoding. This is achieved by time aligning (TA) the HRTFs at high frequencies. By doing this, lower order SH HRTFs are more effective at preserving high frequency information, which improves interpolation results. In this study, the crossover frequency was set at 2.5kHz, as suggested in [21].

3. Machine learning HRTF Restoration

Research domains like speech recognition, natural language processing and computer vision have demonstrated that a more general data-driven method often beats traditional knowledge based

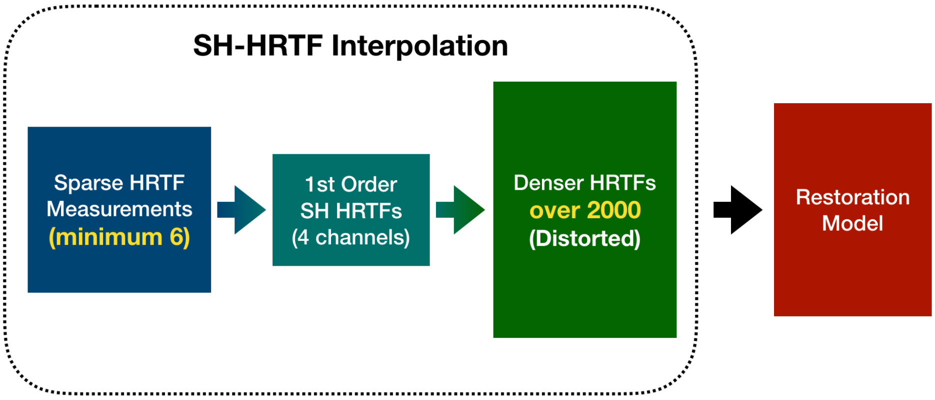

SH-HRTF Interpolation
Sparse HRTF Measurements (minimum 6)
1st Order SH HRTFs (4 channels)
Denser HRTFs over 2000 (Distorted)
Restoration Model

Figure 2. Proposed method overview

signal processing methods in the long run, as the data can keep growing in the future [22]. In image processing, recent developments in machine learning show great potential in noise reduction in images, image inpainting, colorising old photos or videos and neural style transfer [10,11,23–29]. These tasks can be considered to be quite similar to restoring distorted SH interpolated HRTFs. Candidate models include variants of fully connected Neural Networks (NN), Auto-encoder (AE) Convolution Auto-encoder (CAE) [28], Residual Network (ResNet) [30] and Generative Adversarial Networks (GANs) [31].

Most machine learning models require large amounts of labelled data to produce excellent results. However, currently the number of available HRTF databases is very limited. There are only a total number of 233 HRTF datasets freely available combined in (Spatially Oriented Format for Acoustics) SOFA format at the time when we were working on this project [32,33]. Compared to the data size used to train image processing machine learning models, which can be in the region of hundreds of thousands of images, HRTF data is far too few to generalise well or to train sophisticated models. However even with such limited data, SH interpolated HRTF restoration is potentially achievable using machine learning algorithms and is now investigated.

An overview of the proposed method is shown in Figure 2. A subset of HRTFs are selected from a database to represent a sparse HRTF measurement set. These HRTFs are then interpolated in a traditional SH HRTF interpolation manner. After the interpolation, each HRTF measurement will feed into the machine learning model for restoration. This paper chooses the output size for the interpolation process based on the number of HRTF measurements of the original dataset, which is typically over 2000 measurements depending on the dataset. The restored HRTFs output from the machine learning model can then be easily compared with the true HRTF measurements. In this section, we will first discuss the data preparation and format, then introduce a baseline model used in this study before improving it with different enhancements techniques including weight decay, dropout, early stopping etc.

3.1. Data pre-processing

The training and testing data are extracted from different HRTF databases including ARI [34], ITA [35], RIEC [36], SADIE I [37], SADIE II [38], IRCAM Listen [39] and the Bernschutz KU100 [40] database.

The SH HRTF interpolation process takes place in the time domain, as Head Related Impulse Responses (HRIRs) which are converted to the frequency domain Head Related Transfer Functions (HRTFs) after the interpolation process for input to the restoration model. Note that due to the randomness of machine learning models, there is a chance that a model could incorrectly give negative amplitude spectra in the output. To avoid this, the input and output data are scaled to decibels. The output data is then rescaled before converting back to the time domain.

Peer-reviewed version available at Appl. Sci. 2020, 10, 5764; doi:10.3390/app10175764

	azimuth	elevation
1	90.0	0.0
2	270.0	0.0
3	0.0	45.0
4	0.0	-45.0
5	180.0	45.0
6	180.0	-45.0

Table 1. Angle selection for training, validation and testing

	azimuth	elevation
1	0.0	0.0
2	180.0	0.0
3	90.0	45.0
4	90.0	-45.0
5	270.0	45.0
6	270.0	-45.0

Table 2. Angle selection for training and validation only

3.1.1. Data selection

This paper uses 6 measurements with an octahedron configuration for the sparse dataset, which is one of the more challenging cases for SH HRTF interpolation. There are two sets of configurations used, one used in both training and testing, the another one used only in training for data augmentation purpose. The angles are shown in Tables 1 and 2.

Amongst all the popular HRTF datasets, only the SADIE I [37], SADIE II [38], IRCAM Listen [39] and Bernschutz KU100 [40] databases can provide the measurements from these angles. In this paper, Subjects 19 and 20 from the SADIE II database are held out for testing and evaluation of the model and are not used in the training. SADIE II Subject 20 were tracked during the training process. This design tries to show how well the model copes with unforeseen HRTF measurements. Furthermore,the Bernschutz KU100 HRTFs were also excluded from the training and validation sets.This allows us to study the effect of alternate HRTF measurement methods of expected near-match datasets to the existing KU100 measurements in the training data.

Since only the SADIE, IRCAM and Bernshutz datasets have the required measurements for training and validation, it is challenging to produce an accurate model with such a limited variation of HRTF data. Consequently data augmentation of other HRTF datasets was undertaken to provide some extra data for training and validation. The ARI [34], ITA [35] and RIEC [36] datasets were included with modified angles - Positions with an elevation angle at -45°were changed to -30°. This modification was also undertaken for the RIEC data set, as well as positions with an elevation angle at 45°changed to 50°. The effect of this data augmentation is demonstrated in Section 3.2.

Once all the training and validation HRTFs were concatenated, 50000 measurements were randomly selected for the training and validation set with an 80:20 ratio, considering practical training time and the limitations of available computer memory (see Section 3.2).

To improve the speed and stability in the training process, it is considered good practice to standardise the input data before feeding it into the machine learning model. The standardisation equation as follows:

$$z = \frac{x - \mu}{\sigma} \tag{6}$$

where x is the input data, μ and σ are the mean and standard deviation of the training and validation data. Note that the same μ and σ should be used for test data.

To summarise, in total there are 230 subjects used in the training, from SADIE I (18 subjects), SADIE II (18 subjects not counting the 2 hold out data), IRCAM Listen (51 subjects), ARI (60 subjects), ITA (45 subjects) and RIEC (38 subjects). Subjects 19 and 20 from SADIE II and the KU100 measurement from Bernschutz are held out as test sets. Data was standardised before training.

3.2. Baseline Model

There are numerous ML models throughout the literature that have the potential for SH interpolated HRTF restoration. Here we aim to find a model that has a simple architecture whilst able to produce viable results. The reason to use a simple model is based on the consideration of the limited number of HRTF datasets - a simpler model is less likely to over-fit the training data. For comparison,

Comparison of results between mono and stereo inputs (lower is better)						
Model	Overall Mean	Training	Validation	SADIE 20	Bernschutz	Test Mean
Baseline (mono)	50.66	31.40	33.94	62.27	75.04	68.65
Baseline (stereo)	44.46	28.17	30.17	52.34	67.17	59.75

Table 3. Comparison between stereo input and mono input with the baseline model

MSE with different loss functions (lower the better)						
Model	Overall Mean	Training	Validation	SADIE 20	Bernschutz	Test Mean
Baseline (MSE loss)	44.97	27.38	29.43	58.14	64.92	61.53
Baseline (L1 loss)	44.04	28.16	30.14	53.66	64.18	58.92
Baseline (Smooth L1 loss)	44.46	28.17	30.17	52.34	67.17	59.75

Table 4. Comparison between different loss function

all the models in this paper were trained with 500 epochs. The majority were trained with an NVIDIA Quadro P4000M GPU with 32GB of Computer RAM. For the models with extensive amount of data in subsection 3.3.5, a NVIDIA GeForce RTX2080 Ti with 40GB of Computer RAM was used.

The proposed model can be seen as a simplified version of an Inception Network [41]. Separate models for left and right channels are trained individually, whilst the input of the model takes both channels to provide additional information which improves the results as shown in Table 3. The model input size is 2 × 256 (left and right channels of the interpolated HRTFs with the length of 256 samples per channel) and the output is 256 (either left or right channel).

The model proposed here uses a combination of a Convolutional Auto-Encoder (CAE) and a Denoising Auto-Encoder (DAE) [42]. Preliminary test results showed that the DAE is better with the main contour of the frequency response (Figure 3) and CAE is better with the finer details (Figure 4). The combination of the two yields positive results. Similar results are also observed in research with image [43].

The results from the convolutional CAE and DAE are concatenated and passed through a fully connected layer for the voting process.

The complete model is shown in Figure 5. Note that batch normalisation is performed after each convolution layer and transposed convolution layer, with the exception of the very last transposed convolution layer so the output of the CAE should have similar magnitude with the DAE. The models are built and trained with PyTorch [44,45] using smooth L1 loss with Adam optimiser (learning rate: 0.000001, beta 1: 0.9, beta 2: 0.999).

Different loss functions were compared and the results are shown in Table 4. The mean square error (MSE) loss also known as the L2 loss performs worse in the test data but slightly better in the training and validation data. L1 loss, also known as mean absolute error (MAE) showed key improvements with the SADIE Subject 20 dataset and slight improvements with the Bernschutz KU100 test data. The reason L1 loss out-performs MSE loss might be because L1 loss is usually less sensitive to outliers, which is the case when there are HRTFs from different databases. Smooth L1 loss is a combination of L1 loss and MSE loss. For an error below 1, it performs as the MSE loss function; and for the rest it performs as L1 loss. Compared to L1 loss, this method has a continuous derivative at zero so it provides a smoother gradient when the error gets smaller than one. The result of smooth L1 loss further improves the SADIE 20 dataset but there is a trade off on the Bernschutz dataset. Considering the real world application, it is more practical to optimise for unforeseen HRTF measurements of different human subjects instead of different measurement methods with the same artificial head model. The same principle holds in the further optimisation techniques in the upcoming tests. Therefore, the models in this paper use smooth L1 loss as the loss function.

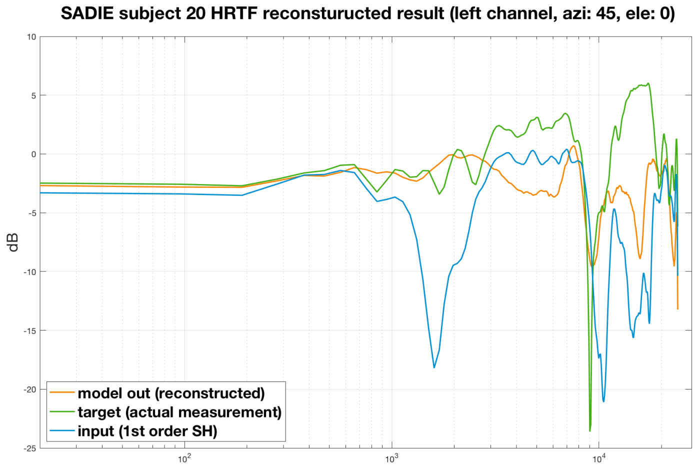


Figure 3. Restoration result example for ipsilateral HRTF response with DAE only (orange: model restored output, green: actual HRTF measurement, blue: SH interpolated HRTF.)

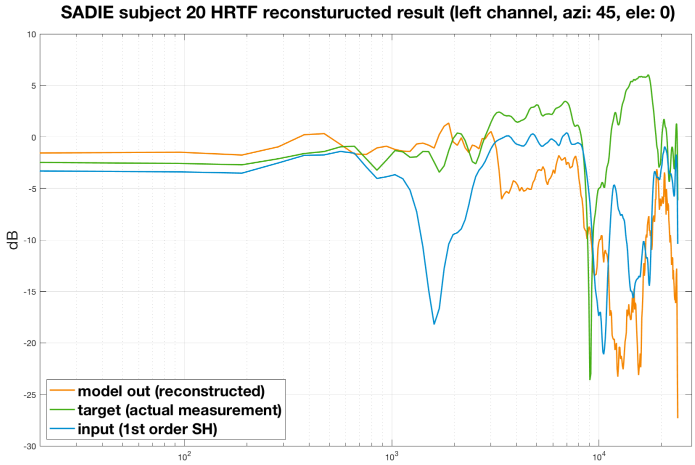


Figure 4. Restoration result example for ipsilateral HRTF response with CAE only (orange: model restored output, green: actual HRTF measurement, blue: SH interpolated HRTF).

The baseline model has been trained for 500 epochs with batch size of 8. The reason for using a small batch size is because prior research shows that a larger batch size may produce worse performance [46,47].

Figures 6 and 7 show the mean squared error (MSE) change during the training. The blue and orange lines are the training and validation results respectively, The green and red lines are the test sets of Subject 20 from SADIE II database and Bernschutz KU100 measurement accordingly. The results show that the training and validation results are trending downwards while the two test sets flatten out after the first 20 epochs. This indicates that the models are over-fitting the training data. Over-fitting is normal in this case considering the limited number of HRTF datasets in the training data (230 in total).

The effects of data augmentation are shown in Table 5, where it can be seen that there are some drawbacks with the training and validation data, and a slight drawback with the SADIE Subject 20 test data. However, the extra data provides significant improvement with the Bernschutz interpolation. This demonstrates that the extra variety of measurements helps the model generalise better across different measurement methods.

The most ideal way reduce over-fitting is to train with more data. However, given the limited HRTF measurements available, different regularisation techniques can be utilised to improve the baseline result, which will be discussed this section.

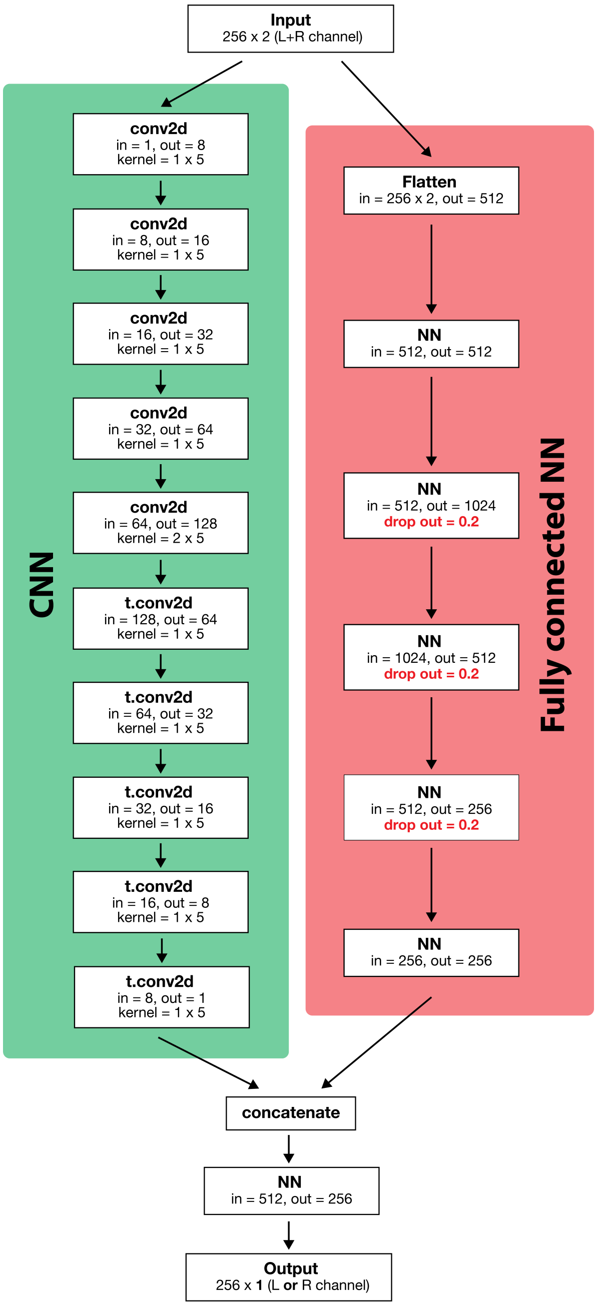


Figure 5. Baseline model and proposed model (proposed model used drop out in some NN layers (in red) among other techniques discuss in section 3.3.3)

Figures 6 and 7 show there are some difference between the SADIE hold out test data and Bernschutz KU100 data although the difference is less on the left channel. However, the average MSE test error of the two channels is approximately the same (left: 59.874, right: 59.633). Further investigations are required to establish the cause of the differences in the left and right channels.

On the other hand, it is interesting to find that the results with the Bernschutz KU100 data also suffers from over-fitting. As there are different KU100 HRTFs in the training data, and the KU100 measurements in the SADIE II database are very similar to the Bernschutz KU100 measurements, it is unexpected to see the model perform quite poorly when comparing the training and validation results. More oddly, in the later sessions, it shows that the Bernschutz KU100 measurements do not seem to benefit from any regularisation methods. One plausible hypothesis is that the current model only

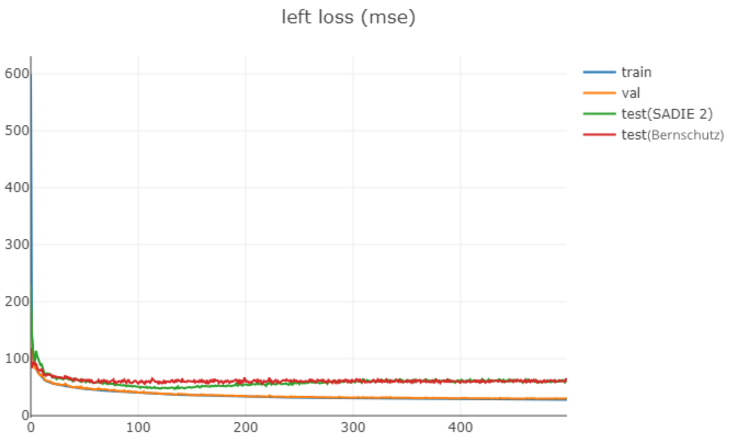


Figure 6. Left MSE during training (blue: training loss, orange: validation loss, green: hold out test data Subject 20 from SADIE II, red: Bernschutz KU100

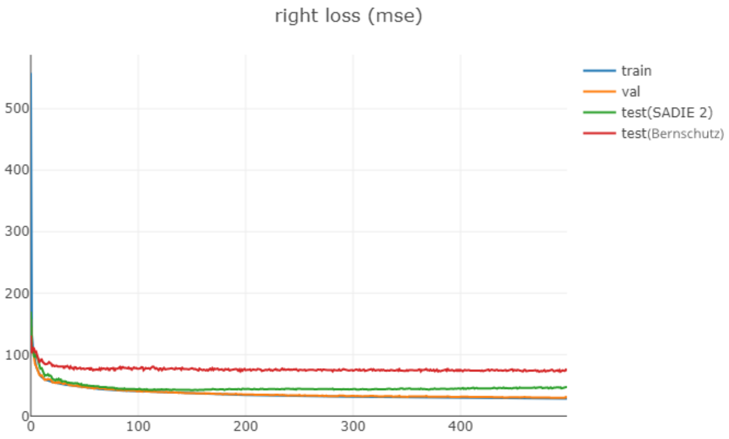


Figure 7. Right MSE during training (blue: training loss, orange: validation loss, green: hold out test data Subject 20 from SADIE II, red: Bernschutz KU100

trained with 6 HRTF databases which represents 6 different measurement setups. The model requires a larger variety of inputs to be able to generalise across different measurement setups and methods.

3.3. Model Enhancement

3.3.1. Model with weight decay

Weight decay is also known as L^2 parameter regularisation or ridge regression. This is a common regularisation method for reducing over-fitting in training. The idea of weight decay is to penalise the large weights in order to simplify the model and reduce over-fitting. This paper uses a weight decay rate of 0.001 as an experiment to see the effect of this regularisation method. The result does not seem to have any positive impact on the SADIE II Subject 20 dataset and there is a main drawback with the Bernschutz KU100 data as the error increased from 67.166 to 71.888 (table 7).

3.3.2. Model with dropout

Dropout randomly “drops out” a percentage of nodes in the neural network during training. The idea is to avoiding co-adaption between nodes by never guaranteeing that any pair will both be used

Compare the results with and without data from ARI, ITA and RIEC (lower the better)						
Model	Overall Mean	Training	Validation	SADIE 20	Bernschutz	Test Mean
Baseline (without extra data)	45.28	18.29	20.55	51.33	90.93	71.13
Baseline (with extra data)	44.46	28.17	30.17	52.34	67.17	59.75

Table 5. Mean Squared Error (MSE) between with and without data from ARI, ITA and RIEC

Compare the results from different models (lower the better)						
Model	Overall Mean	Training	Validation	SADIE 20	Bernschutz	Test Mean
Baseline	44.46	28.17	30.17	52.34	67.17	59.76
With weight decay	46.04	28.59	30.96	52.75	71.89	62.32
With dropout	46.47	29.14	30.08	54.52	72.15	63.33
With weight decay and dropout (proposed model)	45.48	29.85	30.61	47.21	74.23	60.72
With weight decay and dropout (early stopped at 111 epoch)	49.78	40.92	41.00	**47.18**	70.04	58.61
Baseline trained with extra data	39.36	19.74	20.09	59.87	57.72	58.80
With weight decay and dropout and trained with extra data	41.44	22.49	22.07	59.69	61.50	60.60
Bigger model with weight decay and trained with extra data	**31.38**	**7.83**	**10.61**	56.88	**50.22**	**53.55**

Table 6. MSE between different models

during the training process to avoid the model over-relying on a few nodes within a layer. It can also been seen it as randomly sampling from the exponential number of possible narrow sub-networks during training, then provide an average the performance of all these combinations in test time or application. Note that dropout layer can only apply on fully connected layers but not convolutional layers. The model uses a 20 percent dropout ratio on the second to forth fully connected layer.

This method produces worse results with the test data compared to the baseline model, especially with the hold out SADIE II data, but performs slightly better with the validation data. However, such slight differences may be introduced by the randomness of machine learning training. According to the result, dropout seems to have more negative impact on regularisation compared to weight decay. In theory it is possible to increase the dropout ratio or use a different configuration to increase the regularisation effect. However, finding the optimal architecture and hyper-parameters is beyond the scope of this paper and could be part of the future work.

3.3.3. Combining weight decay and dropout

Combining weight decay and dropout shows the best result in the hold out SADIE II data despite there being a noticeable trade off with the Bernschutz dataset. This result indicates that by combining weight decay and dropout, the model can generalise better across different unforeseen HRTF subjects (SADIE hold out) but not the measurement method (Bernschutz). As this model performs the best with the SADIE II test data, this will be the proposed model to be further analysed in Section 4.

It is interesting to find that the combination of the two different methods shows large difference in results, but not with either method individually. It is not clear whether the improvement comes from the combination of the techniques or it is from the cumulative regularisation power. This could be an individual research topic to be investigated in the future.

3.3.4. Early stopping

Early stopping is one of the regularisation methods that sometimes is not considered as good practise in machine learning training because it breaks the principle of orthogonalisation and makes hyper-parameter tuning difficult [48]. Another reason this method is controversial is because the result can be hard to reproduce and compare across different models. However, according to the learning curve from the model combining weight decay and dropout in Figure 8 and 9, early stopping should perform slightly better with the test data, especially with the test set from SADIE II. In order to demonstrate the effect, this paper retrained the proposed model and stopped training at 111 epochs.

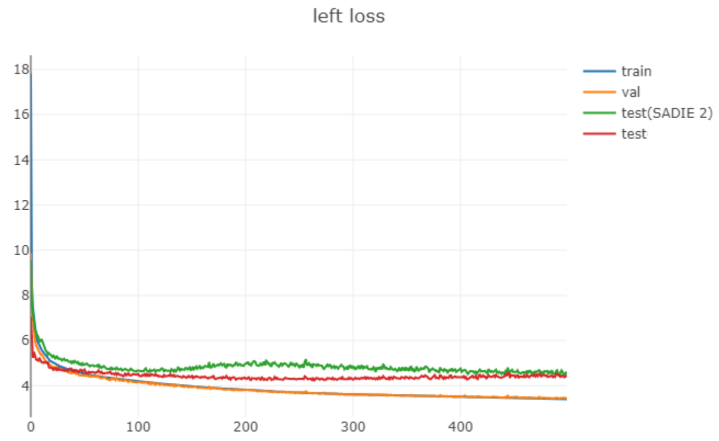


Figure 8. Left channel loss during training (blue: training loss, orange: validation loss, green: hold out test data Subject 20 from SADIE II, red: Bernschutz KU100

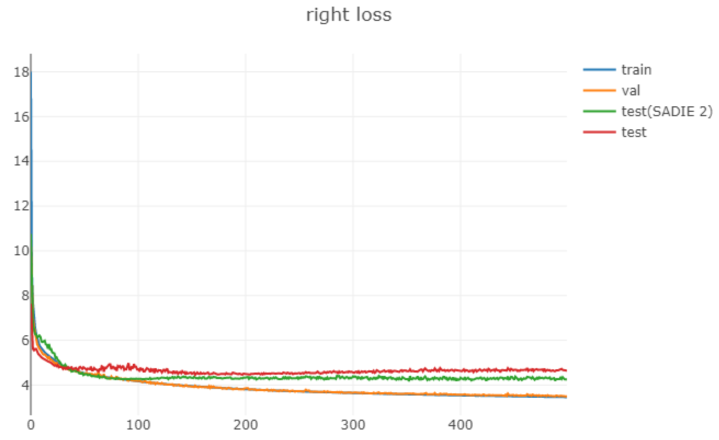


Figure 9. Right channel loss during training (blue: training loss, orange: validation loss, green: hold out test data Subject 20 from SADIE II, red: Bernschutz KU100

3.3.5. Training with more data

To shorten the training time to compare across different methods and considering the limited size of RAM, the models discussed above were trained with 50,000 randomly sampled HRTFs from different angels of the training and validation HRTF sets. However, the best way to reduce over-fitting is to increase the size of the training set. To investigate what the model capable of with more data, a baseline model was trained with 633,000 HRTF measurements. Training this amount of data can take a lot of time per epoch. To speed up the process the batch size for training increased to 32, whilst the validation and test sets remained the same at 8 for better comparison.

Two models, the baseline model and the model with weight decay and dropout were trained with extra data. The baseline model trained with extra data showed major improvement with the Bernschutz dataset, alongside the training and validation sets. However, there is also a noticeable trade off with the SADIE II Subject 20 test data. It is believed that the improvement in the Bernschutz dataset is the result of the model having more examples of the KU100 HRTFs with different measurements at different angles, so it can generalise the measurement method better. The trade off in the SADIE II Subject 20 test data may be caused by the increased batch size, which can induce worse performance [46,47]. Nevertheless, as the extra data is within the same distribution, it is quite unlikely it could provide any noticeable performance improvement with unforeseen HRTF measurement subjects.

3.3.6. Bigger model

Considering the current results and the limited number of labelled data, training a bigger model is against normal machine learning practices.

To demonstrate the potential capability of the proposed method and insight for future research, a slightly deeper model has also been trained with extra data. The goal here is to minimise the training and validation error as much as possible, neglecting the trade off in test datasets' results. To balance out the model size and training time, only the convolution neural network is changed. An extra convolution layer and transposed convolution layer pair was added in the convolution model.

As this model only focuses on the test and validation results, dropout and weight decay regularisation methods are lifted, which defeat the purpose of using a bigger neural network. The model was trained with 633,000 HRTFs with a batch size of 32 for training similar to section 3.3.5 as bigger models usually work better with more data.

Compared to the baseline model, the training time of each epoch from the bigger model increased from 6 minutes to 26 minutes. The model was trained with 500 epochs and the results are shown in Table 6. As expected, the model provides huge improvement in training and validation, but not much improvement in the SADIE II test data. On the other hand, it is interesting to note that it provides the best performance for the Bernschutz data. With extra data, the results seem to align with the training and validation data. The hypothesis is that more data with a bigger model is the key to generalise different measurement methods. However, more experiments with a wider variety of HRTF sets are needed to draw a conclusive result.

4. Evaluation

In this section, the results of the proposed model, including weight decay and dropout will be further analysed for perceptual difference and localisation performance. We utilise perceptual models based on these two criteria in order to provide more robust results for bench-marking.

4.1. Perceptual Spectral Difference

To formally estimate the perceptual performance, the results were further analysed with a Perceptual Spectral Difference (PSD) model [5]. This model calculates the difference between two binaural signals or HRTFs, and presents a more accurate perceptual comparison of spectral differences. This paper compares the difference between before and after the restoration process with the actual HRTF measurements.

The comparison between mean PSD before and after reconstruction with different HRTF datasets is shown in Table 7 and figure 11 with the minimum and maximum PSD plotted. The results show that the model provides significant improvement in PSD across all datasets. However, the minimum and maximum in Figure 11 shows that the model seems to introduce higher PSD error in some cases.

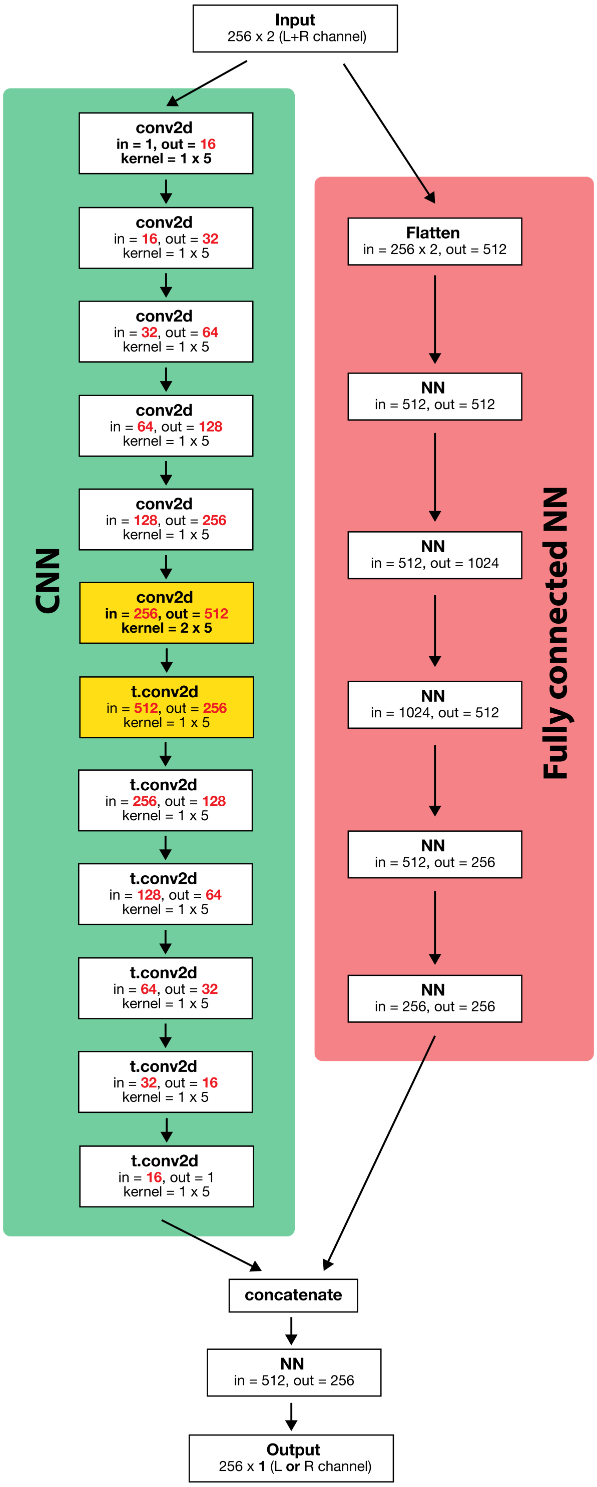


Figure 10. Wider and deeper model (highlighted the main difference comparing to the proposed model

As for most applications that use HRTFs, the smoothness across all angles is more crucial than the average performance. Further analysis with the box plot in Figure 12 shows that although the model may introduced more extreme outliers with unforeseen HRTF measurement subjects (SADIE Subject 19 and 20), the model still improves the majority of the HRTFs and reduces the interquartile range (IQR) in the result.

	SADIE 18 (training data)	SADIE 19 (hold out)	SADIE 20 (hold out)	Bernschutz KU100
PSD (sones) (SH input)	3.03	3.05	2.84	2.57
PSD (sones) (model output)	1.93	2.12	1.96	1.61
Frontal azimuth mean error (SH input)	20.81	25.36	30.00	39.67
Frontal azimuth mean error (model output)	19.29	17.47	15.84	18.98
Sagittal RMS error (deg) (SH input)	40.7	38.6	37.5	38.1
Sagittal RMS error (deg) (model output)	44.3	43.7	39.3	41.4
Sagittal quadrant errors (%) (SH input)	11.5	9.1	7.6	7.1
Sagittal quadrant errors (%) (model output)	24.8	25.2	14.7	12.4

Table 7. Predicted model performance with various HRTF sets

According to figure 11 and 12, Subject 19 from the SADIE II database has a worse maximum PSD and many outliers. To further investigate the cause of the result, Figure 13 shows the PSD before and after comparison of different angles in Subject 19 from the SADIE II database. The left is the SH interpolated HRTFs before restoration and the right is the one after processed with the ML model. The figure shows that most of the high PSD results were introduced in the lower frontal region. It is unclear what caused the increased error, but one hypothesis is that the abnormality was caused by the shadow effect from the knees, as the SADIE II along with most of the other databases are measured with subjects sitting on a chair.

However, figure 14 shows the PSD before and after comparison with the holdout Subject 20 from SADIE II database. Besides a small area of minor PSD increment in the very low frontal region, the restored result shows there is no significant abnormality in any region. To have a deeper understanding of the cause of the abnormality in Subject 19 holdout data, extra tests with more HRTF sets are required.

Figure 15 shows the PSD before and after comparison with the Bernschutz KU100 data. The model seems to perform better with unforeseen measurement method, as it shows improvement in the PSD at all angles. It is worth noticing that the lower frontal region in the figures does not have any oddly high PSD results, perhaps because KU100 is a head only dummy head model.

4.2. Localisation performance

The localisation performance was analysed with the May's model and Baumgatner's model in the Auditory Modelling Toolbox (AMT) [49–51]. The May's model is for frontal azimuth localisation on the horizontal plane and the Baumgatner's model is for the frontal sagittal plane.

Table 7 showed the localisation results of the proposed model. The frontal azimuth localisation mean error from the May's model show improvement in all HRTF datasets. However, for frontal sagittal plane with the Baumgatner's model, results in RMS error and quadrant errors indicate all the reconstructed HRTFs seems to perform worse in the frontal sagittal plane localisation.

The current model suffers in localisation tasks may because it used smooth L1 loss as the loss function. The smooth L1 loss only focus on the magnitude difference at each frequency point. There may not be any meaningful connection between those magnitude difference and localisation performance. Therefore, the model failed to optimised the HRTFs for the localisation performance. Future model could add some types of localisation error into the loss function to improve the localisation results.

5. Discussion and future work

The goal of this paper is to prove the hypothesis that machine learning can be used to restored distorted interpolated HRTFs. To draw a convincing argument, this paper picked one of the more challenging situations based on 1^{st} order SH and 6 measurements. Models with higher order SH and more measurements should have better performance than the current one as less data needs to be restored.

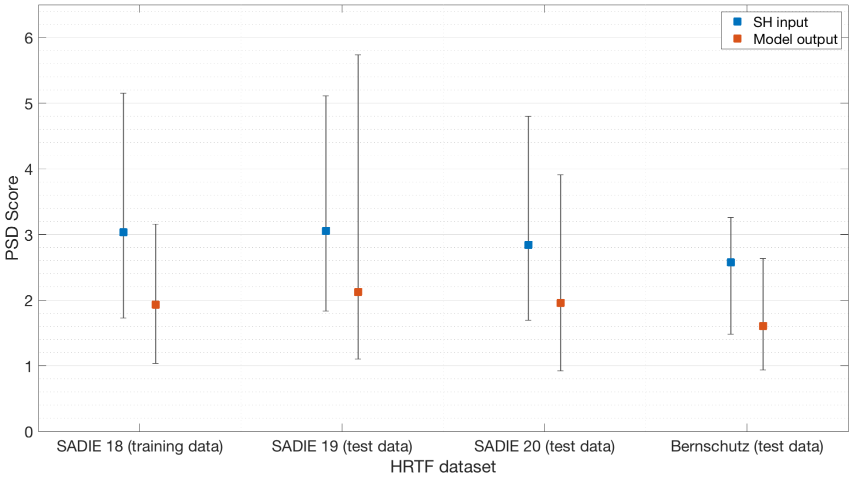


Figure 11. Minimum, maximum, and average PSD across different angle in different data sets

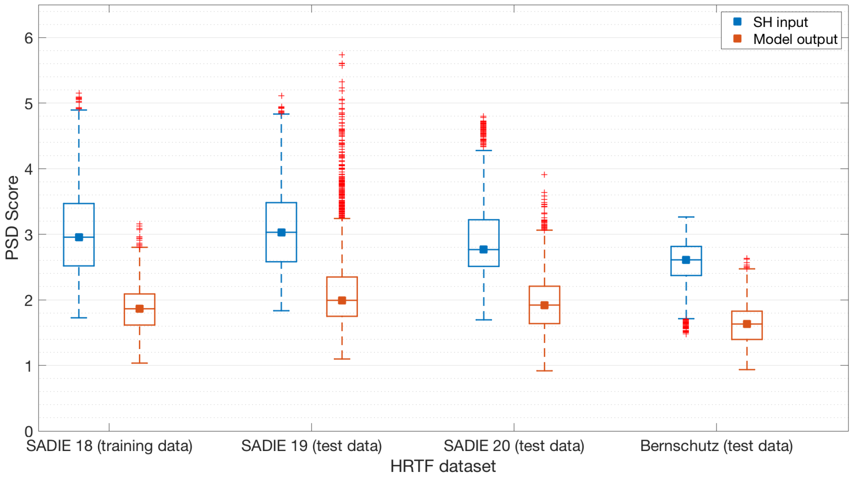


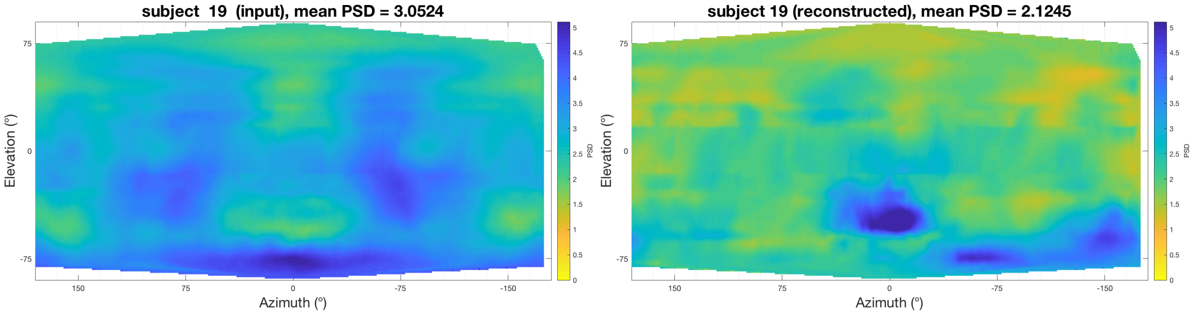
Figure 12. PSD median and box plot with whiskers with maximum 1.5 IQR

The results show that a simple ML model can be used to restore distorted SH interpolated HRTFs, although the current state of this model is far from optimised for application. It is believed that there will be significant improvement if more HRTF measurements are available for training in the future. Under the current situation, one way to improve the model is through hyper-parameter tuning, including the parameters for regularisation.

An alternative method that may be possible to reduce over-fitting is to use data augmentation. HRTF measurements are expensive and tedious, therefore it is not very likely there will be a huge increase in HRTF measurement data in the near future. To augment the current dataset, one possible way is to use more different sparse HRTF configurations or different SH order to train the model. Table 5, shows that even if the extra data is not perfect, it is possible that it can still improve the model performance in some cases.

Similar to data augmentation, noise injection is different regularisation method that has be shown to work better than weight decay in some cases [52–54]. By picking the right parameters, it is believed that it could generalise better across measurement methods as the model could focus on the general information across various HRTF measurements as opposed to the artefacts introduced by different measurement methods.

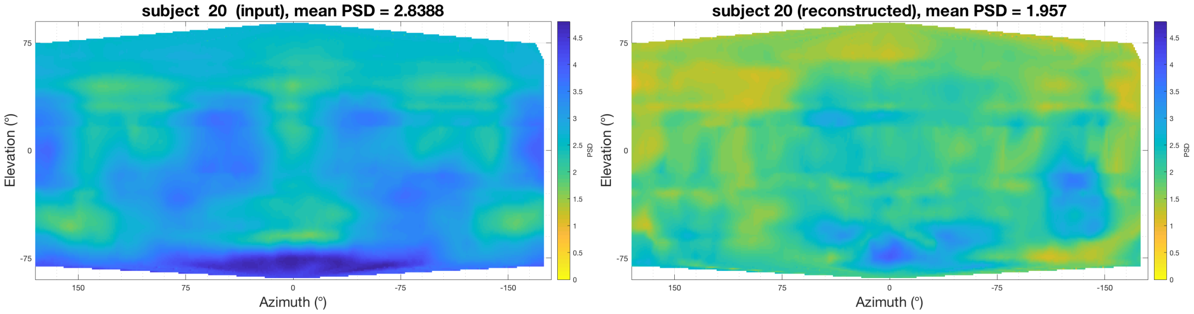
Another problem that was observed from the current model is the localisation performance. Although it shows some improvement in the horizontal plane, the sagittal error needs improvement.



(a) before reconstruction

(b) after reconstruction

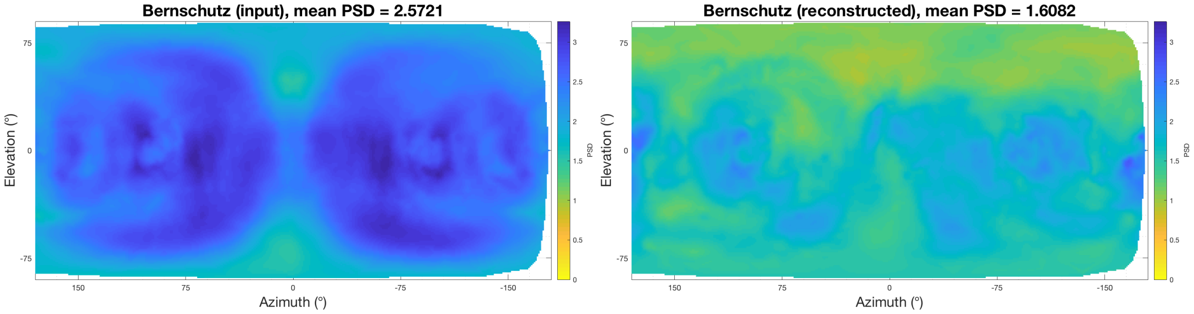
Figure 13. PSD of Subject 19 from SADIE II database



(a) before reconstruction

(b) after reconstruction

Figure 14. PSD of Subject 20 from SADIE II database



(a) before reconstruction

(b) after reconstruction

Figure 15. PSD of Bernschutz KU100 dataset

Discussed in Section 4, it is believed that the smooth L1 loss function only compares the difference in each frequency and fails to capture other useful metrics of HRTFs. Potentially, some custom loss functions can be implemented to improve the model. Gatys et al. [55] trained a separate machine learning model for content loss and implemented a special function for style loss. Similar methods like training a localisation model as localisation loss function may be able to solve the problem. Furthermore, with the recent success in generative adversarial networks (GAN), it should be possible to build a GAN based on localisation performance [29,31,56–58]. However, as a GAN can be unstable to train and it usually requires a lot of tuning, it may not be the most effective way for SH interpolated HRTF restoration.

This paper tries to show a general insight of using machine learning for HRTFs reconstruction. According to table 6, table 7 and the discussion in session 4, to apply the idea in real-life application, optimising the model for some narrative tasks should yield better performance. With a more specific application in mind, not just the parameters of the model can be changed, the model can also be trained with cleaner or more particular data specialised for the task. Alternatively, using transfer learning based on the current model can provide a head start for these applications.

6. Conclusion

HRTF interpolation in the SH domain often suffers from distortion in the high frequencies. With the recent development in machine learning algorithms, this paper has shown that it is possible to restore the distorted SH interpolated HRTFs with a ML model. Although the proposed method suffers from over-fitting, it still shows improvements in perceptual difference and localisation performance. It is believed that with more training data in the future, the model performance can be vastly improved. However, HRTF measurements can be difficult and time consuming to obtain. With the current size of available data, optimising the model with a narrative use case with some hyper-parameters tuning and data augmenting should have the best chance to put the model in real world applications.

Supplementary Materials: Supporting data and code are available at GitHub: https://github.com/Benjamin-Tsui/SH_HRTF_Restoration.

Author Contributions: Methodology, Benjamin Tsui; writing–review and editing, Benjamin Tsui, William Smith and Gavin Kearney; supervision, William Smith and Gavin Kearney

Funding: This research received no external funding.

Acknowledgments: Thanks for the technical support from Cal Armstrong and Thomas McKenzie.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SH	Spherical Harmonic
HRTF	Head Related Transfer Function
CAE	Convolutional Auto-Encoder
DAE	Denoising Auto-Encoder
PSD	Perceptual Spectral Difference
VR	Virtual Reality
AR	Augmented Reality
ITD	Interaural Time Difference
ILD	Interaural Level Difference
HRTFs	Head Related Transfer Functions
VBAP	Vector Base Amplitude Panning
TA	Time Aligning
NN	Neural Networks
AE	Auto-encoder
ResNet	Residual Network
GANs	Generative Adversarial Networks
SOFA	Spatially Oriented Format for Acoustics
MSE	Mean Square Error
MAE	Mean Absolute Error
IQR	Interquartile Range
AMT	Auditory Modelling Toolbox

References

1. Pulkki, V. Virtual sound source positioning using vector base amplitude panning. *Journal of the audio engineering society* **1997**, *45*, 456–466.
2. Gerzon, M.A. Periphony: With-height sound reproduction. *Journal of the audio engineering society* **1973**, *21*, 2–10.

3. Noisternig, M.; Musil, T.; Sontacchi, A.; Holdrich, R. 3D binaural sound reproduction using a virtual ambisonic approach. IEEE International Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS'03. 2003. IEEE, 2003, pp. 174–178.
4. Kearney, G.; Doyle, T. Height Perception in Ambisonic Based Binaural Decoding. Audio Engineering Society Convention 139, 2015.
5. Armstrong, C.; McKenzie, T.; Murphy, D.; Kearney, G. A perceptual spectral difference model for binaural signals. *145th Audio Engineering Society International Convention, AES 2018* **2018**, pp. 1–5.
6. Lee, G.W.; Kim, H.K. Personalized HRTF modeling based on deep neural network using anthropometric measurements and images of the ear. *Applied Sciences* **2018**, *8*, 2180.
7. Katz, B.F. Boundary element method calculation of individual head-related transfer function. I. Rigid model calculation. *The Journal of the Acoustical Society of America* **2001**, *110*, 2440–2448.
8. Young, K.; Kearney, G.; Tew, A.I. Loudspeaker Positions with Sufficient Natural Channel Separation for Binaural Reproduction. Audio Engineering Society International Conference on Spatial Reproduction - Aesthetics and Science, 2018.
9. Young, K.; Tew, A.I.; Kearney, G. Boundary element method modelling of KEMAR for binaural rendering: Mesh production and validation. *Interactive Audio Systems Symposium* **2016**, pp. 1–8.
10. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision* **2017**, *2017-Octob*, 2242–2251. doi:10.1109/ICCV.2017.244.
11. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* **2017**, *2017-Janua*, 5967–5976. doi:10.1109/CVPR.2017.632.
12. Gamper, H. Head-related transfer function interpolation in azimuth, elevation, and distance. *The Journal of the Acoustical Society of America* **2013**, *134*, EL547–EL553. doi:10.1121/1.4828983.
13. Grijalva, F.; Martini, L.C.; Florencio, D.; Goldenstein, S. Interpolation of Head-Related Transfer Functions Using Manifold Learning. *IEEE Signal Processing Letters* **2017**, *24*, 221–225. doi:10.1109/LSP.2017.2648794.
14. Hartung, K.; Braasch, J.; Sterbing, S.J. Comparison of different methods for the interpolation of head-related transfer functions. AES 16th International Conference: Spatial Sound Reproduction, 1999, pp. 319–329.
15. Martin, R.L.; McAnally, K. Interpolation of Head-Related Transfer Functions. *Air Operations Division Defence Science and Technology Organisation* **2007**.
16. Evans, M.J.; Angus, J.A.S.; Tew, A.I. Analyzing head-related transfer function measurements using surface spherical harmonics. *The Journal of the Acoustical Society of America* **1998**, *104*, 2400–1637. doi:10.1121/1.3336399.
17. Zotter, F.; Frank, M. *Ambisonics*, 1 ed.; Vol. 19, *Springer Topics in Signal Processing*, Springer International Publishing: Cham, 2019; pp. XIV, 210. doi:10.1007/978-3-030-17207-7.
18. Bertet, S.; Jérôme, D.; Sébastien, M. 3D Sound Field Recording with Higher Order Ambisonics – Objective Measurements and Validation of Spherical Microphone. *AES 120th Convention* **2006**, pp. 1–24.
19. Kearney, G.; Doyle, T. Height perception in ambisonic based binaural decoding. *139th Audio Engineering Society International Convention, AES 2015* **2015**, pp. 1–10.
20. Zaunschirm, M.; Schörkhuber, C.; Höldrich, R. Binaural rendering of Ambisonic signals by head-related impulse response time alignment and a diffuseness constraint. *The Journal of the Acoustical Society of America* **2018**, *143*, 3616–3627. doi:10.1121/1.5040489.
21. Mckenzie, T.; Murphy, D.T.; Kearney, G. An Evaluation of Pre-Processing Techniques for Virtual Loudspeaker Binaural Ambisonic Rendering. EAA Spatial Audio Signal Processing symposium, 2019, pp. 149–154. doi:10.25836/sasp.2019.09.
22. Sutton, R. The Bitter Lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, 2019. (Accessed on 10/29/2019).
23. Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; Li, H. High-resolution image inpainting using multi-scale neural patch synthesis. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* **2017**, *2017-Janua*, 4076–4084. doi:10.1109/CVPR.2017.434.
24. Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.Z.; Ebrahimi, M. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212* **2019**.

25. Yan, Z.; Li, X.; Li, M.; Zuo, W.; Shan, S. Shift-net: Image inpainting via deep feature rearrangement. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2018**, *11218 LNCS*, 3–19. doi:10.1007/978-3-030-01264-9{_}1.
26. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image Inpainting for Irregular Holes Using Partial Convolutions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **2018**, *11215 LNCS*, 89–105. doi:10.1007/978-3-030-01252-6{_}6.
27. Antic, J. jantic/DeOldify: A Deep Learning based project for colorizing and restoring old images (and video!). https://github.com/jantic/DeOldify, 2019. (Accessed on 10/29/2019).
28. Mao, X.; Shen, C.; Yang, Y.B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. Advances in neural information processing systems, 2016, pp. 2802–2810.
29. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016**, *2016-December*, 770–778, [1512.03385]. doi:10.1109/CVPR.2016.90.
31. Goodfellow, I.; Pouget-Abadie, J.; . . . , M.M.A.i.n.; 2014, U. Generative adversarial nets. *Papers.Nips.Cc* **2014**, pp. 1–9. doi:10.1017/CBO9781139058452.
32. General information on SOFA, 2013.
33. SOFA - Spatially Oriented Format for Acoustics, 2015.
34. Institute, A.R. ARI HRTF Database, 2014.
35. Bomhardt, R.; De La, M.; Klein, F.; Fels, J. A high-resolution head-related transfer function and three-dimensional ear model database A high-resolution head-related transfer function dataset and 3D ear model database. *Proc. Mtgs. Acoust. The Journal of the Acoustical Society of America* **2016**, *29*. doi:10.1121/1.4970409.
36. Watanabe, K.; Iwaya, Y.; Suzuki, Y.; Takane, S.; Sato, S. Dataset of head-related transfer functions measured with a circular loudspeaker array. *Acoustical Science and Technology* **2014**, *35*, 159–165. doi:10.1250/ast.35.159.
37. SADIE | Spatial Audio For Domestic Interactive Entertainment.
38. Armstrong, C.; Chadwick, A.; Thresh, L.; Murphy, D.; Kearney, G. Simultaneous HRTF Measurement of Multiple Source Configurations Utilizing Semi-Permanent Structural Mounts. Audio Engineering Society Convention 143. Audio Engineering Society, 2017.
39. Warusfel, O. Listen HRTF Database, 2003.
40. Bernschütz, B. A Spherical Far Field HRIR/HRTF Compilation of the Neumann KU 100. *Fortschritte der Akustik – AIA-DAGA 2013* **2013**, pp. 592–595.
41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2015**, *07-12-June*, 1–9. doi:10.1109/CVPR.2015.7298594.
42. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning. *Nature* **2016**, *521*, 800, [arXiv:1312.6184v5]. doi:10.1038/nmeth.3707.
43. Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F.A.; Brendel, W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* **2018**.
44. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. NIPS, 2017.
45. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; Garnett, R., Eds.; Curran Associates, Inc., 2019; pp. 8024–8035.
46. Masters, D.; Luschi, C. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612* **2018**.
47. Wilson, D.R.; Martinez, T.R. The general inefficiency of batch training for gradient descent learning. *Neural Networks* **2003**, *16*, 1429–1451. doi:10.1016/S0893-6080(03)00138-2.

48. Other regularization methods - Practical aspects of Deep Learning | Coursera. https://www.coursera.org/lecture/deep-neural-network/other-regularization-methods-Pa53F. (Accessed on 01/01/2020).
49. Baumgartner, R.; Majdak, P.; Laback, B. Modeling sound-source localization in sagittal planes for human listeners. *The Journal of the Acoustical Society of America* **2014**, *136*, 791–802. doi:10.1121/1.4887447.
50. May, T.; Van De Par, S.; Kohlrausch, A. A probabilistic model for robust localization based on a binaural auditory front-end. *IEEE Transactions on Audio, Speech and Language Processing* **2011**, *19*, 1–13. doi:10.1109/TASL.2010.2042128.
51. Søndergaard, P.; Majdak, P. The Auditory Modeling Toolbox. In *The Technology of Binaural Listening*; Blauert, J., Ed.; Springer: Berlin, Heidelberg, 2013; pp. 33–56.
52. Aggarwal, C.C. *Neural Networks and Deep Learning: A Textbook*; Springer International Publishing, 2018.
53. Zur, R.M.; Jiang, Y.; Pesce, L.L.; Drukker, K. Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. *Medical Physics* **2009**, *36*, 4810–4818. doi:10.1118/1.3213517.
54. He, Z.; Rakin, A.S.; Fan, D. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 588–597.
55. Gatys, L.; Ecker, A.; Bethge, M. A Neural Algorithm of Artistic Style. *Journal of Vision* **2016**, *16*, 326. doi:10.1167/16.12.326.
56. Zhang, M.; Zheng, Y. Hair-GANs: Recovering 3D Hair Structure from a Single Image. *arXiv preprint arXiv:1811.06229* **2018**.
57. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *Veterinary Immunology and Immunopathology* **2018**, *166*, 33–42. doi:10.1016/j.vetimm.2015.04.007.
58. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training GANs. *Advances in Neural Information Processing Systems* **2016**, pp. 2234–2242, [1606.03498].