Deep Binaural Direction of Arrival Estimation

An Experimental Analysis

Jago T. Reed-Jones

A thesis submitted to Liverpool John Moores University in
partial fulfilment of the requirements for the degree of
Doctor of Philosophy

June, 2025

Abstract

The objective of binaural direction of arrival (DoA) estimation is to find the DoA of a sound source by measuring the sound field with a binaural array. This field increasingly applies deep learning to this task, particularly convolutional neural networks which are trained on relatively raw representations of the binaural audio.

This work investigates the field, establishing common trends among different publications, particularly in the data preparation, scrutinising these trends for instances of the emergence of collective wisdom without empirical backing. Based on this, an experimental evaluation is performed to gain insight into the efficacy of different existing and novel techniques, based on a recurring testing framework.

Such experimental evaluations are undertaken for several topics: an analysis of acoustic conditions on the performance of binaural DoA estimation, a broad empirical study on binaural feature representations to be used with convolutional neural networks (CNNs), the proposal and comparison of convolutional recurrent neural network (CRNN) models for binaural DoA estimation, and an investigation into binaural DoA estimation in the mismatched anechoic condition; referring to a mismatch in head-related transfer function (HRTF) measurements between training and testing datasets for an identical binaural array.

The findings in this thesis lead to recommendations for more effectively using deep neural networks for binaural DoA estimation, while also demonstrating the limited ability of such systems to generalise to unseen binaural data when using simulated binaural datasets which are limited in their scope.

Declaration

I hereby declare that this thesis titled “Deep Binaural Direction of Arrival Estimation” and the work presented in it is my own, and that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Acknowledgements

I owe deep thanks to my supervisors Dr. David Ellis, Dr. John Marsland, Prof. Paul Fergus and especially my director of studies Dr. Karl Jones for their guidance in the execution of this work.

I am grateful to my examiners Dr. Jez Wells and Dr. Sebastian Chandler-Črnigoj for their thoughtful assessment of this thesis, and Dr. Ellen Dawson for chairing the examination.

My enthusiasm for audio engineering which has culminated in this PhD research is in thanks to the educators involved in my bachelors and masters studies. I am grateful to all those at Liverpool John Moores University and Universiy of West London involved in my prior education, and especially to Colin Robinson, Nino Auricchio, and Dr. Christos Chousidis who have had a particularly enduring impact.

I gratefully acknowledge the financial support of the Faculty of Engineering, Liverpool John Moores University.

Thank you to supportive and helpful research colleagues at LJMU, especially Szilárd Novoth, Xinrui Liang, Bingyu Mu, and Steve Stroud.

I have greatly appreciated the support of my colleagues at Audioscenic, and I am especially thankful to Jacob Hollebon, Daniel Wallace, and Yasmine Mellah, each of whom has offered valuable support.

The enduring support of my parents has been fundamental to my academic journey.

I am deeply grateful to my partner, Jing Guo, for her endless unwavering love and support throughout this endeavour: I am not sure I could have accomplished without you.

Contents

List of Figures

Acronyms

6DoF	6 degrees of freedom
AdaGrad	adaptive gradient algorithm
AP	affinity propagation algorithm
ASA	auditory scene analysis
ASR	automatic speech recognition
BGD	batch gradient descent
BiGRU	bidirectional gated recurent unit
BiLi	binaural listening
BiLSTM	bidirectional long short-term memory
BRIR	binaural room impulse response
BRTF	binaural room transfer function
BSSL	binaural sound source localisation
CASA	computational auditory scene analysis
CNN	convolutional neural network
CPS	cross power spectrum
CRNN	convolutional recurrent neural network
CTF	common transfer function
DBN	deep belief network
DCASE	detection and classification of acoustic scenes and events
DCT	discrete cosine transform
DFT	discrete Fourier transform
DNN	deep neural network
DoA	direction of arrival
DTF	directional transfer function
ELM	extreme learning machine

ERB	equivalent rectangular bandwidth
FEM	finite element method
FIR	finite impulse response
GCC-PHAT	generalised cross correlation phase transform
GFB	gammatone filterbank
GFC	gammatone-frequency cepstrum
GFCC	gammatone-frequency cepstral coefficient
GMM	gaussian mixture model
GRU	gated recurrent unit
HATO	head above torso orientation
HATS	head and torso simulator
HRIR	head-related impulse response
HRTF	head-related transfer function
IACCF	interaural cross correlation function
IC	interaural coherence
ICA	independent component analysis
IFT	inverse fourier transform
IIR	infinite impulse response
ILD	interaural level difference
IPD	interaural phase difference
IR	impulse response
ISM	image source method
ITD	interaural time difference
LSTM	long short-term memory
LTI	linear time-invariant

MCT	multi-conditional training
MFC	mel-frequency cepstrum
MFCC	mel-frequency cepstral coefficient
MGD	minibatch gradient descent
ML	machine learning
MLP	multi-layer perceptron
MSO	medial superior olive
NMF	non-negative matrix factorisation
ReLU	rectified linear unit
RIR	room impulse response
RMS	root mean square
RMSE	root mean square error
RMSLE	root mean square localisation error
RMSProp	root mean square propagation
RNN	recurrent neural network
SGD	stochastic gradient descent
SLP	single layer perceptron
SNR	signal-to-noise ratio
SOFA	spatially oriented format for acoustics
SSL	sound source localisation
STFT	short-time fourier transform
TDOA	time difference on arrival
TF	time-frequency
TSP	telecommunications and signal processing laboratory
VAD	voice activity detector
VR	virtual reality

1 Introduction

1.1 Motivation

sound source localisation (SSL) systems are routinely encountered in every day life. A technology which is often associated with its military and robotics applications now also finds home in many speech processing applications for a simple reason; sound localisation helps machines understand us.

The techniques used in SSL tend to rely on microphone arrays containing as large a number of transducers as possible.

It is not just machines that can perform this task. We humans, as well as much of Animalia, are able to identify the direction of a sound. For many animals this ability is significant as it allows for the detection of predators, or conversely prey. Despite this, none of these animals have more than two ears, contrary to the dependence on microphone arrays in SSL.

binaural sound source localisation (BSSL) is the field of study which seeks to replicate this within machine hearing systems. Doing so would potentially allow for reduction of the costly data acquisition associated with large arrays, and potentially be vital for unlocking some of the more remarkable elements of human sound cognition, such as the cocktail party effect (Cherry, 1953).

Interest in such technology is found in fields related to speech processing, where it is hoped that breaking down auditory scenes into individual components would provide a method by which computer understanding of speech is advanced.

There is interest in BSSL also from the field of robotics, and in particular for humanoid robots (Keyrouz, 2014; Nguyen *et al.*, 2018; Dávila-Chacón *et al.*, 2018) where BSSL can be directly applied for the same role it serves in the human audition system. Another important application of BSSLs is in hearing aids and cochlear implants. The sound localisation ability in hearing impaired listeners is limited (Durlach *et al.*, 1981), and amplification of signals alone, as done in hearing aids, is not known to improve the localisation ability (Noble and Byrne, 1990; Köbler and Rosenhall, 2002). To aid this degraded ability HRTF rendering based on a known DoA has been proposed, for which BSSL is required to find the DoA.

More typically however, it is through the use of BSSL in pursuit of expanding SNR that is helpful, as accurate DoA estimation allows for use of binaural beamforming to attenuate unwanted sound sources (Doclo *et al.*, 2010), and consequently improve speech intelligibility (Froehlich *et al.*, 2015). As will be established in this work's literature review, the field of BSSL has begun to become dominated by the application of deep learning techniques to this task; with it particularly often being reported that deep learning techniques are more capable of robust localisation in adverse acoustic conditions (Ma *et al.*, 2017).

1.2 Structure of Thesis

The main body of this work is divided into five chapters, a summary of each of these are given here.

Relevant Background

The task of SSL is typically seen as a problem in the domain of signal processing, and now typically machine learning; however, the task pertains not only to digital signals, but also on sound's propagation in the real world, and so relies heavily on knowledge drawn from acoustics. The binaural element expands this further due to its biological origin, leading to audiology and psychology to also provide important background.

The knowledge required from these fields to understand the task of BSSL is introduced in Chapter 2.

Building on this, the chapter also introduces previous work relevant to this study; this starts with an overview of HRTF measurement datasets, required due to their importance to the task, and that there exists no similar survey which is sufficiently current. Proceeding this, methods of approaching the task of BSSL are presented, beginning with legacy algorithmic and model based approaches, introducing some now more classical machine learning approaches, and finally an in-depth review of all known work on the use of deep learning for the task of BSSL.

Research Aims and Design

This chapter introduces the primary aim and the objectives of this thesis, and discusses the research methodology applied in pursuit of this aim.

Localising in Simulated Acoustic Environments

As will be shown during the literature review, the on-trend approach to Deep-BSSL is to use deep neural networks (DNNs) with convolutional layers: CNNs.

Chapter 4.1 shows the effectiveness of using such models for BSSLs in perfect acoustic conditions, which serves as an introduction to the general experimental framework applied in all chapters of this work of training and testing models on binaural datasets created under controlled conditions to gain insight over an aspect of the dataset, or the model itself. This also introduces the evaluation metrics used throughout this thesis.

Following this, the same model is evaluated on binaural datasets containing various acoustic degradations: the mismatched HRTF condition as well as additive noise in the forms of diffuse noise, interfering sound sources, and the noise mixtures which are used throughout the rest of this thesis. Following this experimentation is undertaken on BSSL in the presence of reverberation, finding that CNNs generalise poorly to unknown rooms when trained on binaural data made with either real or synthetic binaural room impulse responses (BRIRs).

These experiments address the effect of changing reverb time, finding experimental proof of the aforementioned generalisation issue, the effect of changing only room geometry, localisation performance when training and testing with real measured BRIRs, and the resolution of this issue with generalised datasets, at the expense of increased dataset complexity.

Feature Representations

As will be established in the literature review, it is typical that works on BSSL focus on novel models for achieving improved performance; one oversight occurring due to this is a lack of attention on the optimal preparation of datasets.

One such example is that previous works which apply CNN to BSSL use a number of different feature-representations of the binaural audio, with no discernible justification.

This chapter begins to address this oversight, by looking specifically at the conversion of audio in feature representations.

First, experimentation is undertaken into different magnitude based feature representations, and then on phase and time-delay based representations of the audio.

Deep Learning Architectures

This chapter addresses the design of the deep learning models rather than the datasets.

Throughout this thesis 2D convolutional layers are used in models, therefore also making all the feature representations 2D. The first experiment in this chapter makes an essential check of whether the use of 2D matrices is justified, or whether 1D representations would have been preferable.

A trend identified in the larger field of SSL, which is less common in BSSL, is the use of recurrent layers alongside convolutional layers, in models referred to as CRNNs.

This chapter seeks to answer the following questions: Does CRNN also improve performance in BSSL as it does in microphone array-based SSL, and if so, which recurrent layers are optimal to employ to achieve that objective?

Do address this, a comparison is undertaken of CRNN with a CNN, followed comparison of four CRNN models with differing recurrent layers.

Mismatched Anechoic Condition

Due to the way in which previous studies have been performed, an acoustic condition highly relevant to BSSL is the mismatched anechoic condition; being when the freefield HRTFs used in the training sets differ from those in the testing dataset. Theoretically these are identical, however small amounts of measurement noise can create generalisation issues.

This chapter investigates CNNs susceptibility to this issue, and proposes some possible methods of augmenting HRTFs to reduce this issue without requiring larger numbers of measured freefield HRTF datasets to overcome the generalisation issue.

Conclusion

An assessment of how the results presented in this work do or do not support the use of deep learning in DoA estimation.

1.3 Novel Contributions

The scope of this work includes several novel contributions. These are:

- An experimental analysis of BSSL with CNNs in a variety acoustic conditions, establishing the largest challenges in deep binaural DoA estimation.
- A broad experimental comparison of binaural feature representation techniques for deep binaural DoA estimation with CNNs to establish best practices.
- A study on the use of CRNNs for binaural DoA estimation, establishing their benefit over CNNs, and the proposal and comparison of four CRNN models leading to evidence based recommendations for CRNN design.
- Investigation of binaural DoA estimation in the mismatched anechoic condition, including the proposal, testing and evaluation of novel augmentation techniques.

1.4 Associated Publications

Selected results and work in this thesis have also been presented in the following publications.

Chapter 5 - Feature Representations
Reed-Jones, J. T., Jones, K. O., Fergus, P., Marsland, J., and Ellis, D. L. (2023). "Comparison of Performance in Binaural Sound Source Localisation using Convolutional Neural Networks for differing Feature Representations". In: Audio Engineering Society Convention 154. Audio Engineering Society.

Chapter 6 - Deep Learning Architectures
Reed-Jones, J. T., Fergus, P., Ellis, D. L., and Jones, K. O. (2024a). "A Study on the Relative Accuracy and Robustness of the Convolutional Recurrent Neural Network based approach to Binaural Sound Source Localisation". In: Audio Engineering Society Convention 157. 290. Audio Engineering Society.

Reed-Jones, J. T., Fergus, P., Ellis, D. L., Marsland, J., and Jones, K. O. (2024b). “Improving Full Horizontal Plane Binaural Sound Localization by use of BiLSTM”. In: 2024 International Conference on Information Technologies (InfoTech). IEEE, pp. 1–4.

2 Relevant Background

The task of binaural sound source localisation draws from several fields including acoustics, audio signal processing, machine learning, audiology and more. To understand the task fully it is necessary to introduce relevant knowledge from these fields, which is presented in this chapter.

2.1 Acoustics & Psychoacoustics

2.1.1 3D Space

Since this work relates to localisation of sources in space, spatial coordinates of objects will regularly be used. These can either be given as Cartesian coordinates, (x, y, z) in a 3D space, or with the polar coordinates (φ, θ, r) where φ is azimuth, θ is elevation and r is distance, a convention used consistently throughout this work. Polar coordinates are generally a more useful concept in sound localisation, as it is often the direction of arrival (DoA) of a sound being estimated, without the inclusion of distance.
The two systems can be defined in relation to each other as:

$$\begin{aligned} x &= r \sin\theta \cos\varphi \\ y &= r \sin\theta \sin\varphi \\ z &= r \cos\theta \end{aligned} \tag{1}$$

Another convention used in this work is defining azimuth (φ) on a scale from $-180°$ to $180°$ so that $0°$ is the forward direction and $180°$ is the backward direction and also equal to $-180°$. Elevation (θ) is defined between $-90°$ and $90°$ so that $-90°$ is the downwards direction, $90°$ is upwards.

Occasionally referred to in this thesis are the front and back hemifields. These refer to the range of values in front and behind the reference point, so that:

$$\begin{aligned} front &\rightarrow -90° < \varphi < 90° \\ back &\rightarrow 90° < \varphi < -90° \end{aligned} \tag{2}$$

remembering that $180°$ wraps around to $-180°$.

2.1.2 Human Anatomy

Orientation with relation to the human body can be described with several anatomical planes:

The horizontal plane otherwise known as transversal or axial plane, is the plane which divides the body into a head and tail sections.

The median plane otherwise known as the sagittal plane, is the plane which divides the body into left and right sections.

The frontal plane otherwise known as the coronal plane, is the plane which divides the body in front and back sections.

Typically these planes are imagined to converge at the centre of the body, around the abdomen. However, in sound localisation it is typically more useful to think of the centre of the head as being the centre of the body. This creates an approximation in which the ears sit on both the horizontal and frontal planes, and the median plane equally divides the axis between the ears.

This axis drawn between the ears can also be referred to as the interaural axis.

2.1.3 Wave Propagation

Sound is the name given to oscillations of pressure travelling through any elastic medium, typically air. These peaks and troughs propagate through the medium typically as transverse waves, where particles are moved to cause compressions and rarefactions in the medium (Everest, 2022).

An important feature of acoustic waves is sound pressure p, which is measured with the unit Pascals (Pa). Due to the very large total range of typical values of p, this is often converted onto the decibel scale with a reference level of 20μPa.

$$\mathrm{dB(SPL)} = 20 \log \frac{p}{2 \times 10^{-5}} \tag{3}$$

Another important parameter is sound intensity, which defines the level of power for an area. Intensity is related to pressure as:

$$I = pv \tag{4}$$

where v is particle velocity.

Geometric Spreading

Imagine sound as emanating from a point in space, or a point source. Sound will travel from this point in space at an equal speed in all directions, leading to sound spreading as a sphere. Figure 1 shows waves spreading from a point source.

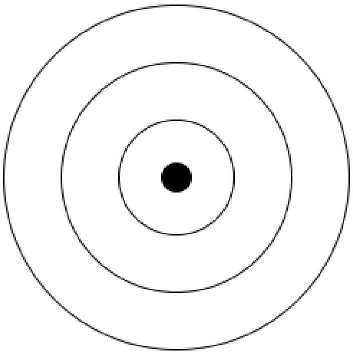

Figure 1: 2D representation of spherical spreading from a point source

As sound wave travels away from a source then it can be thought of a sphere with an increasing radius. This leads to a proportional loss of sound pressure as the radius increases.

$$p \propto \frac{1}{r} \tag{5}$$

particle velocity also has the same inversely proportional relationship

$$v \propto \frac{1}{r} \tag{6}$$

This leads to intensity being proportional to the square of the inverse, a relationship known as the inverse square law.

$$I = \frac{1}{r^2} \tag{7}$$

The Speed of Sound

The propagation speed of sound in this thesis is referred to as c_0. The speed of sound in air is dependent upon some variables, such as the temperature and humidity of the air, but in linear models of wave propagation it is presumed to be non-dependent upon any characteristic of the wave itself, and thus is treated as a constant. The assumed value of c_0 exclusively used in this work is given by Equation (8).

$$c_0 = 343m/s \tag{8}$$

Reflection

When sound interacts with a boundary one possible outcome is reflection. In the case of specular reflections, as shown in Figure 2, the angle of angle of reflections is equal to the angle of incidence. Specular reflections, however, occur only when the reflecting boundary is larger than the wavelength and is seemingly smooth compared to the wavelength. In practice, most surfaces contain irregularities which results in diffuse reflections, in which the sound wave is scattered over multiple directions.

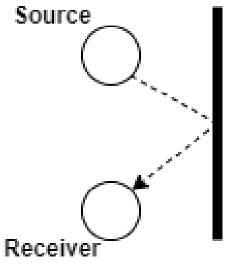

Figure 2: A reflection

Absorption

In acoustics, absorption refers to the transformation of acoustic energy into another innocuous form (Everest, 2022). This happens in air itself, leading to some attenuation by the medium. The level of this attenuation is described as an air absorption coefficient, m, which is dependent on the frequency of the sound, the temperature of the air, the humidity, and the atmospheric pressure. This attenuation is larger at higher frequencies.

When sound interacts with a reflecting boundary, the boundary also has an absorption effect on the reflected sound. This similarly is described with an absorption coefficient α, which can vary widely for different materials as well as for different frequencies of the sound, with higher frequencies typically being more absorbed.

Diffraction

When sound interacts with an obstacle, it will spread around the obstacle's edges in a process referred to as diffraction. This is shown in Figure 3 with planar waves interacting with an obstacle. The waves behind the obstacle are attenuated relative to those which have not been diffracted.

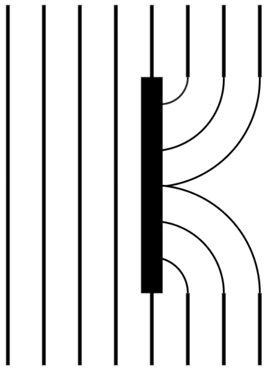

Figure 3: Diffraction around an obstacle.

The size of the obstacle defines how much disturbance to the sound field is caused, with small obstacles having little influence. Behind the obstacle, particularly for larger obstacles, the sound pressure level will be reduced, but not eliminated.

Nearfield vs Farfield

As previously established, sound waves spread from a point source spherically. As the radius of this sphere gets larger, the surface of the sphere appears flatter. This means that at an adequate distance from a sound source, travelling sound waves can be approximated as planar waves. This phenomenon is illustrated in Figure 4.

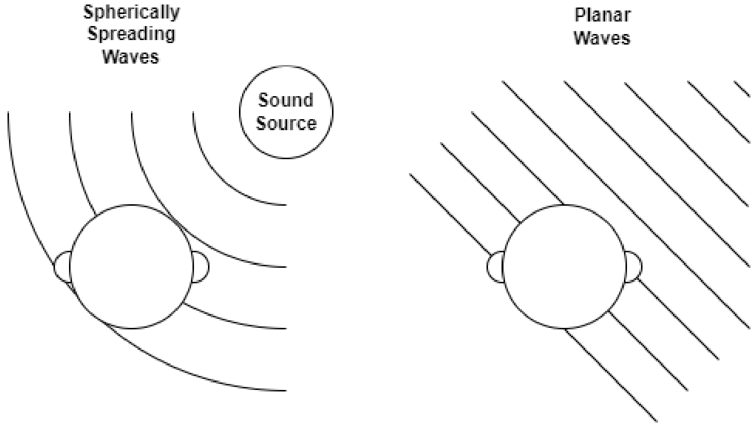

Figure 4: Illustration of Spherically Spreading and Planar Waves

The distance at which the planar wave approximation becomes accurate depends on the wavelength, aperture of the original sound source, and aperture of receiver; in the context of binaural listening this is often approximated as 1m.

2.1.4 Binaural Hearing

Binaural hearing refers to the perception of sound using the two ears, a near ubiquitous trait in the animal kingdom. This is not due to a common ancestor; different parts of the animal kingdom independently developed a binaural hearing system (Schnupp and Carr, 2009), suggesting a strong evolutionary advantage to this feature.

The range of auditory tasks which benefit from binaural hearing are vast, but a fundamental one is the ability to perceive the DoA of sound, an ability referred to as binaural sound localisation.

Based on the information already presented we can formulate an understanding of the cues used to achieve this.

First, consider the most basic binaural formulation: a sound source and two sensors. Relative to the sensors, the sound source has some angle, φ, and a distance, r. Additionally, the distance between the sensors is denoted as d.

As clearly illustrated in Figure 5, sound waves are likely to arrive at one sensor before the other. In other words there is a time difference on arrival (TDOA). This is due to the path lengths from the source to each of the sensors being different; the degree to which they are different is dependent on the DoA, with sources on the interaural axis having the maximum possible TDOA, while sources on the midsagittal plane create a symmetric image, and therefore no TDOA.

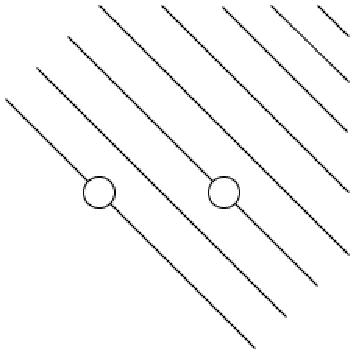

Figure 5: Binaural Freefield Model

This phenomenon, where sound arrives at different times between the ears, is the first of the binaural localiser's salient cues, and in this context is referred to as the interaural time difference (ITD).

Presuming planar waves, the relationship between ITD and azimuth can be given with:

$$\varphi = \arcsin \frac{c_0 \tau}{d} \tag{9}$$

This is not the only cue however. To uncover the next cue requires a new model: the rigid-sphere model (Rayleigh, 1907).

By imagining not just two sensors, but two sensors sitting on a sphere through which sound does not easily transmit. In this case, the head casts an acoustic shadow behind which the sound will be attenuated. If an ear is occluded by the sphere, therefore, a level difference between the ears will occur: this is the other salient cue, interaural level difference (ILD). This is illustrated in Figure 6.

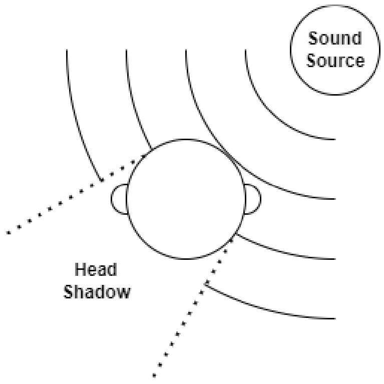

Figure 6: Illustration of the Head Shadow

The rigid sphere model also reveals a change to the freefield model of ITD: the sphere also somewhat increases the difference in path length between the ears, due to the sound now needing to travel around the sphere. This is illustrated in Figure 7.

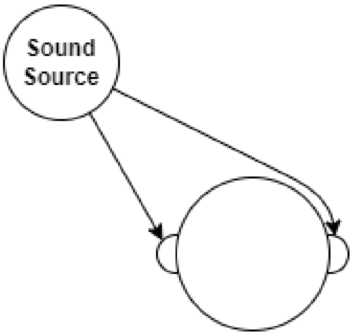

Figure 7: Shortest path length according to Rigid Sphere Model

The relation between φ and τ previously given in Equation (9), therefore, is incorrect. One method of solving for this is to scale the head diameter to add extra ITD:

$$\begin{gathered}\varphi = \arcsin \frac{c_0 \tau}{kd} \\ 1.2 \leq k \leq 1.3\end{gathered} \tag{10}$$

This is known as the sine law (Blauert, 1997).

Alternatively the Woodworth ITD model (1938) gives a more geometrically motivated approximation, derived from the path lengths around a spherical head.

$$(\varphi + \sin\varphi) = \frac{2c_0\tau}{d} \tag{11}$$

These models make no suggestion to any frequency dependence since the path length and speed of sound remain constant at all frequencies. However, in reality measured ITDs reveal a level of frequency dependence with ITD not remaining constant for altering frequency (Abbagnaro *et al.*, 1975). Based on investigation into the ITD of a mannequin head, Kuhn (1977) found that the Woodworth model becomes invalid at lower frequencies, with ITDs increasing to beyond those modelled.

The Human Hearing Apparatus

Humans hear with the ear, but that is made up of smaller parts, which can be categorised into the outer ear, middle ear and inner ear.

The outer ear consists of the pinna, and the entrance of the ear canal. The pinna, or auricle, is the visible external part of the ear. Further categorisation of the pinna can be seen in Figure 8.

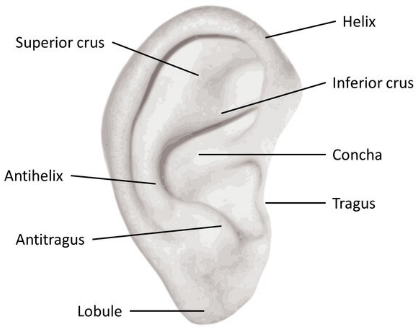

Figure 8: Anatomy of the Pinna (Hussain, 2020)

The middle ear is responsible for converting pressure waves in air into pressure waves within the cochlea. It consists of an air-filled space, sealed by the tympanic membrane (Volandri *et al.*, 2011). The tympanic membrane vibrates under pressure, which in turn incites motion in the last part of the middle ear, three bones named the ossicles. The ossicles acoustically couple the tympanic membrane to the cochlea deeper within the ear, providing the necessary change in acoustic impedance (Peake *et al.*, 1992) required to transmit vibration to the cochlea.

The inner ear consists primarily of the cochlea: the part of the inner ear responsible for converting mechanical waves into electrochemical impulses. The cochlea is a hollow tube, rolled into a spiral, and containing fluid. Within this is a tissue structure named the basilar membrane, which contains sensory hair cells; the transduction mechanism of the ear (McPherson, 2018). The cochlea duct reduces in diameter along its length, leading to different parts of the cochlea being sensitive to different frequencies (Greenwood, 1961), this being the cause of humans' ability to independently detect sound levels at different frequencies.

After the cochlea, the electrochemical signals are sent to the brain via the auditory nerve. The part of the brain responsible for auditory processing is named the auditory cortex (Gelfand and Calandruccio, 2009). The human ear does not seem to be very phase sensitive, although some reactivity has been reported to phase changes (Laitinen *et al.*, 2013). The duplex model of hearing, however, is entirely dependent on humans being sensitive to small timing differences between the two ears, and so there must be a mechanism by which interaural time differences can be sensed.

The Jeffrees model (Jeffress, 1948) is a neurocomputational model seeking to explain such phenomena, which models the onsets coming from each path as tapped delay lines, wherein the taps are joint with coincident detectors. The tap at which coincidence occurs corresponds to the interaural time difference.

Much later, evidence for such an architecture was found in barn owls (Carr and Konishi, 1990), however other strategies have been found in other avian and mammalian species (Ashida and Carr, 2011)

Cone of Confusion

A notable factor of both the shadowless head model and the freefield model is that they are both entirely symmetrical about the interaural axis. This is true also for ILD and ITD values which are equal in the front and rear hemifields. This leads to two or more possible horizontal plane positions existing for any viable ITD or ILD value.

Consider also the relation of positions in 3D space according to the rigid sphere model. The complete symmetry in all dimensions of a sphere means that as elevation is changed, the binaural cues are not affected. Similarly, assuming planar waves altering distance also has no effect upon binaural cues. Accounting for this, then, reveals a possible range of positions for a given ITD in the shape of a cone.

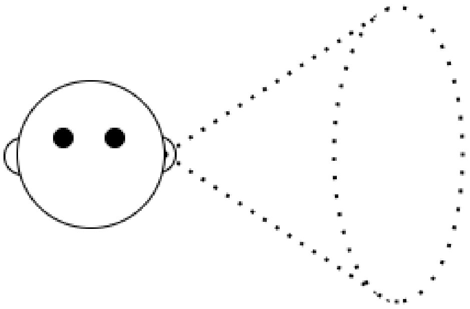

Figure 9: Cone of Confusion

This is the cone of confusion, illustrated in Figure 9. This phenomenon does occur among humans; particularly notable is front-back confusion where we incorrectly localise a sound source's mirror image on the horizontal plane. However, in most familiar environments this is not a common problem, and so this model of localisation must not yet be complete.

Monaural Cues

Instead of models, consider instead the real human head. Compared to the rigid-sphere, it is notably not symmetrical about the coronal plane nor any horizontal plane. This is particularly true when examining the outer ear, where paths from the rear are blocked by the pinnae. This does somewhat alter the binaural cues, adding a slight frequency dependence, however this is still not sufficient information for resolving the cone of confusion.

The other effect that the outer ear, and rest of the head, has on sound is imparting a unique filtering characteristic, as at different frequencies sound interacts with different parts of the head in different fashions. As this filtering is linear and time-invariant, this can be described with a set of transfer functions, which are named head-related transfer functions (HRTFs), and their time-domain counterparts head-related impulse responses (HRIRs). It is possible to measure HRTFs as per other acoustic transfer functions.

Torso Effects

Considering the human head alone still does not provide a complete representation of how humans hear sound. The torso can also impose an acoustic shadow for some source positions, but also can cause a scattering of sound, particularly at the shoulders. These cues are particularly import for median plane localisation. Torso effects can be considered through expansion of the rigid sphere model into a 'snowman' model (Duda *et al.*, 2002), as well as in real measured HRTFs.

2.1.5 Spatial Audio

Binaural Audio

The field of spatial audio aims to create immersive audio experiences through reproduction of a full sound field. One of the techniques used is through binaural audio. This can be native binaural audio, which refers to audio recorded with a binaural array. Here, a binaural array refers to a two element array which imparts binaural cues on oncoming sound. Alternatively, the audio could be a binaural render, in which audio is seemingly placed at a desired location in space through application of HRTFs. This binaural audio can then be played back through either headphones, or loudspeakers employing crosstalk cancellation processing, to give the illusion of a three-dimensional soundfield.

Head Simulators

It is possible to measure HRTFs using real human heads, by placing microphones on or in the ear. However, measurements conducted in this way are often compromised due to the necessity of the subject to remain completely motionless over a long measurement period. Additional issues are that the microphones may change position over the course of a measurement as cannot be permanently affixed, and measurements can not be taken with a microphone placed at the actual location of the tympanic membrane.

An attractive alternative is the use of head simulators; these being mannequin heads with microphones positioned in the ears of the mannequin, which are designed to accurately model the acoustic characteristics of the human head. This approach allows for the use of larger and higher quality microphones, more accurate positioning of these microphones, and the removal of unwanted movement and noise from the human body.

Binaural Rendering

In most spatial audio situations, using audio recorded with a head simulator is not practical. More typically audio is spatialised through use of HRIR/HRTF. For an audio source, $x[n]$, binaural directionality can be applied through convolution with an HRIR of the target DoA.

$$y[n, r, \theta, \varphi] = x[n] * hrir[n, r, \theta, \varphi] \tag{12}$$

or equivalently in the frequency domain:

$$Y[\omega, r, \theta, \varphi] = X[\omega] HRTF[\omega, r, \theta, \varphi] \tag{13}$$

This relationship is also fundamental to binaural sound source localisation (BSSL), as it describes the relation of binaural audio to the sound source's DoA.

The HRTFs used for rendering can be created through impulse response (IR) measurement of head simulators, IR measurement of human heads with binaural microphones, or some simulation method. These methods of acquisition are covered further in Chapter 2.1.7.

The resulting binaural audio can be used to give spatial impression if played directly over headphones, though the correct delivery of monaural cues also relies upon a completely transparent reproduction system, and so the transfer function of the headphones themselves are often also inverted as part of a binaural renderer (Sunder *et al.*, 2014).

Binaurally recreating a spatial scene with loudspeakers is more difficult, as both ears exist within the soundfield produced by both loudspeakers. To reproduce this, the soundfield control technique known as crosstalk cancellation (Nelson and Elliott, 1991), in which the unwanted contralateral path is cancelled with destructive interference.

Research Interests in Binaural Audio

While binaural spatial reproduction has been known to have been used for over a century (Boren, 2017), its adoption has only been recent with its use in video games (Farkaš, 2018) and virtual reality (VR) (Serafin *et al.*, 2018). This has led to its increased popularity as a research topic.

There are some common research themes:

Personalised HRTFs as monaural cues are dependent on the pinnae, and binaural cues are dependent on the size and shape of the head, they can vary significantly between different subjects, so HRTFs should be considered individualised (Wenzel *et al.*, 1993). As measuring HRTF is lengthy process requiring specialised equipment, there is significant interest in synthesising HRTFs through numerical calculation (Ziegelwanger *et al.*, 2015; Brinkmann *et al.*, 2023) or modelling based on anthropomorphic features (Hu *et al.*, 2008; Miccini and Spagnol, 2020).

HRTF Interpolation A limiting factor on the quality of binaural rendering is the sampling density of the datasets, which is often limited by measurement apparatus and time taken to perform measurements. When trying to render sources of directions not present in the dataset, interpolation can be used; however this is a compromise, especially when using diffuse field binaural room transfer functions (BRTFs). One common research theme emerging due to this is accurate binaural room impulse responses (BRIRs) interpolation (Garcia-Gomez and Lopez, 2018; Bruschi *et al.*, 2020; Qiao and Choueiri, 2023; Li *et al.*, 2025), and the similar task of spatial upsampling of HRTF datasets (Hogg *et al.*, 2024).

6 Degrees of Freedom another part of binaural rendering, particularly for VR applications, is trying to reproduce accurate dynamic cues based on the listeners real movements. Doing this for all directions of motion (x,y,z translation as well as x,y,z rotation) is often referred to as 6 degrees of freedom (6DoF). This is achieved by counter-rotating sound sources direction based on the the listener's head pose, while translation adaption is achieved by recalculating the receiver to source vectors based on the listener's reported position relative to a reference point (McCormack *et al.*, 2023). Research on the topic tends to focus on efficiently achieving this (Plinge *et al.*, 2018) or the listener perception of doing so (Bacila and Lee, 2023)

2.1.6 Binaural Arrays

Consider cases in which binauralised audio is desired without use of human participants. Given spatial hearing's reliance on cues imposed by the body, for these purposes it is essential to think of the human body as being a part of the human hearing apparatus, which must be replicated.

To do so, the human anatomy is broken down according to its relevance to sound localisation, which is firstly summarised in Table 2.

Table 2: Sound localisation cues imparted by different parts of human anatomy

	Inner Ear	**Outer Ear**	**Head**	**Torso**	**Rest Of Body**
Monaural Cues	×	✓	×	✓	✓
Binaural Cues	×	×	✓	×	×

Inner Ear As the inner ear does not impart direction-dependent filtering to a sound, it does not need to accurately be modelled in a binaural array. This, however, is not true for other similar but more popular fields requiring use of ear simulation, such as headphone measurement, and some of the binaural arrays discussed do simulate the inner ear.

Outer Ear (Pinna) The pinna's heavy influence in HRTFs means that for a binaural array to contain plausible monaural cues, it must accurately model the pinna. Due to this, binaural arrays typically create human-like pinnae of a soft material with acoustic properties designed to match that of the real ear. There exists a commercially popular binaural array which places pinnae in the freefield with no head[1] which trades accurate binaural cues for reduced cost and easier usability for field recording. However, these are not commonly used in research beyond testing of its accuracy

[1] 3DIO Binaural Microphones: `https://3diosound.com/collections/microphones` [Accessed 21st October 2023]

(Crnigoj, 2020). Also, there exist ear simulators, which simulate of only a single ear[2], which due to having only one receiver cannot be considered binaural arrays.

Head The head is entirely responsible for imparting the binaural cues on oncoming sound waves. Despite this importance, ITD and ILD are mostly determined by the geometry of the head rather than its material. Owing to this in head simulators the head is often just modelled as some plastic with a density similar to a head, rather than the more accurate modelling of the pinnae. An example of a binaural array which models only the head, and not the pinnae, is Rayleigh's rigid sphere (Rayleigh and Lodge, 1904).

Torso Given that the torso is known to induce monaural cues, some measurement apparatus also include torso: this apparatus is often abbreviated as head and torso simulator (HATS). Lifelike accuracy is not a great priority for the torso simulator, but it is typically desired that the head should be able to make free yaw movements relative to a fixed torso, as per human head movements.

Rest of Body Other large parts of the body such as the arms and legs must have some small impact on the HRTF, however it is reasonable to presume that this is not significant for sound localisation as this has never been proven. Other parts of the body that do have some impact include the hair, which despite its proximity to the ears, is generally not studied in relation to HRTF, and the bones inside of the head which are able to deliver sound to the ear in a process named bone conduction. Bone conduction is also not generally considered important to sound localisation.

Head Simulators used in Research

Knowledge of the head simulators used in research is important as measurements of these are often used in creating binaural DoA estimators.

Neumann KU-100

The Neumann KU-100 is a binaural microphone specifically designed for binaural

[2] GRAS Acoustics 43AG Ear Simulator: `https://www.grasacoustics.com/products/ear-simulator-kit/product/ss\_export/pdf2?product_id=737` [Accessed 21st October 2023]

recording[3]. It is heavily used in spatial audio research due to its accurate sound localisation cues, while being relatively inexpensive compared to more multi-applicable head simulators.

GRAS KEMAR

The GRAS KEMAR Head and Torso Simulator[4] also regularly features in HRTF measurement datasets. As opposed to the KU-100, the KEMAR has a torso, as well as more accurate inner-ear simulation for other measurement tasks. Notably, the KEMAR has detachable pinnae, for which there are two varieties which can be installed commonly referred to as the small pinnae and large pinnae.

B&K HATS

Brüel & Kjær manufacture a Head and Torso Simulator also designed for transfer function measurement, and which can also be used specifically for HRTF measurement.

Head Acoustics HMS Range

Head Acoustics produce a range of head simulators, both for electroacoustics measurement and binaural recording[5]. Despite its suitability, these rarely feature in HRTF related research.

FABIAN

FABIAN is a Head and Torso Simulator belonging to Technical University of Berlin for which HRTF Measurements are available (Brinkmann *et al.*, 2017).

Cortex

The Cortex HATSs is a now discontinued line of head simulator manufactured by

[3] Neumann KU-100: `https://www.neumann.com/en-en/products/microphones/ku-100/` [Accessed 21st October 2023]

[4] GRAS 45BB KEMAR: `https://www.grasacoustics.com/products/head-torso-simulators-kemar/product/ss_export/pdf2?product_id=733` [Accessed 21st October 2023]. Other similar KEMAR models exist also, but do not differ in a way that affects HRTFs

[5] Head Acoustics Artificial Heads: `https://www.head-acoustics.com/products/artificial-head-binaural-recording` [Accessed 21st October 2023]

Neutrik Cortex Instruments. Its use can be found in publications from the original time of manufacture (Maijala, 1997; Kahana, 2000), with remaining units still being used in modern work (Vicente and Lavandier, 2020).

Binaural Microphones

The real human head can be turned into an electroacoustic array by the addition of microphones at the ears. This approach is discussed further in Chapter 2.1.7, but it is important to note that this can be achieved with a pair of well matching microphones small enough to be placed on or in the ear.

2.1.7 HRTF Measurement and Simulation

There have been attempts to measure HRTFs for almost a century (Troger, 1930), however it was with the advent of digital recording and a renewed interest in binaural recording that HRTF measurements became common in audio research.

The spatially oriented format for acoustics (SOFA) (2022) is a standardised file format for storing and sharing acoustic measurement data. It has conventions for different forms of acoustic data, including several appropriate binaural data:

- SimpleFreeFieldHRIR
- SimpleFreeFieldHRTF
- SimpleFreeFieldHRSOS (Second Order Section)
- FreeFieldHRTF (Supports spatially continuous representations)
- FreefieldHRIR (Supports spatially continuous representations)
- SingleRoomSRIR (Spatial Room Impulse Response; can be used for arbitrary number of receivers)
- SingleRoomMIMOSRIR (Multiple Input Multiple Output Spatial Room Impulse Response)

2.1.8 Freefield Measurements

When measuring HRTFs, the aim is to measure a transfer function showing the effect the human body has upon a sound source. Consider, however, the case of sound propagation in a room. Thinking in the time-domain, the BRIR can be thought of as the convolution of HRIRs and the room impulse response (RIR):

$$\mathrm{brir}[n] = \mathrm{hrir}[n] * \mathrm{rir}[n] \tag{14}$$

meaning also that the resulting sound source, as acted upon by the BRIR, has the relation:

$$y[n] = \mathrm{hrir}[n] * \mathrm{rir}[n] * x[n] \tag{15}$$

showing that in order to measure only the HRIR, the RIR must equal exactly $\delta[n]$; that is to say the room must have no observable effect upon the signal. The most effective way to achieve this would be to remove all physical boundaries allowing sound to propagate freely; however, this is rarely practical. Instead, specialist rooms are designed to try to simulate this situation: these rooms being named anechoic chambers.

Anechoic chambers tend to feature walls covered in large wedges of some very absorbent material, such that upon reflection the wave receives a significant reduction in energy. Furthermore, the shape is designed to maximise the number of reflections that each path is likely to undergo before reaching the receiver.

There is also another element of soundfields which obscures the HRIR from observation also; additive noise.

$$y[n] = \mathrm{hrir}[n] * \mathrm{rir}[n] * x[n] + \eta[n] \tag{16}$$

Where $\eta[n]$ is the additive noise. This can come in different forms, but includes waves propagating from other sound sources. While it not difficult to remove other sound sources from the same room, sound can transmit well enough through typical walls so as to still be audible; another objective in anechoic chamber design deals with this issue, the elimination of acoustic transmission through the walls of the chamber. This is achieved by suspending the room within another larger room, with the impedance mismatches between the solid materials of the wall and the air acting as the attenuator.

Ideally all HRTF measurements should be carried out in an anechoic chamber for these reasons.

2.1.9 Room Auralisation

As previously established, when a wave propagates in rooms, a listener will hear not only the sound directly coming from the sound source, but also reflections of the sound off the walls and other boundaries in the room. In various applications, it is advantageous to be able to simulate such a sound field: a process here named room auralisation.

Image Source Method

A virtual scene of reflections can be efficiently calculated using geometric principles, in an approach known as the image source method (ISM) (Savioja, 1999).

Consider sound as a ray—that is, it travels only in straight lines. A reflection can then be modelled by tracing the path of such a ray as it travels from a source, to a boundary, and then to a receiver, as previously illustrated in Figure 2. As can be seen, the path of interest consists of two vectors, one from the source to the boundary, and one from the boundary to the receiver.

Finding this path is greatly aided by the geometric property that if the source is mirrored about the boundary, a single vector from the mirrored source to receiver is identical to the reflected distance in distance, as well as sharing the same point of intersection with the boundary. This is shown in Figure 10.

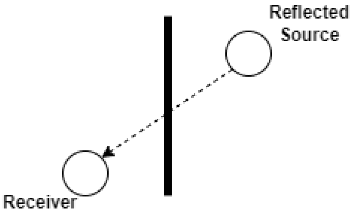

Figure 10: A mirror source image of a a reflection

Recalling that $T = d/c_0$, the propagation time T can be estimated based on an assumption of speed of sound, c_0, as d can be calculated as the euclidean distance between mirrored source and receiver.

A value for d, furthermore, allows us to estimate the wideband attenuation caused by spherical expansion, by use of inverse square law:

$$p \propto \frac{1}{d} \tag{17}$$

where p is pressure. This can be further expanded by also considering the frequency dependent attenuation effects of air absorption.

It also becomes possible to model surface absorptions effects, as this method provides knowledge of the number boundaries the ray has intersected. Consider now not only a single boundary, but an entire rectilinear room. We can apply the same technique, not just mirroring the source but also all the boundaries, which we refer to as the image space as seen in Figure 11.

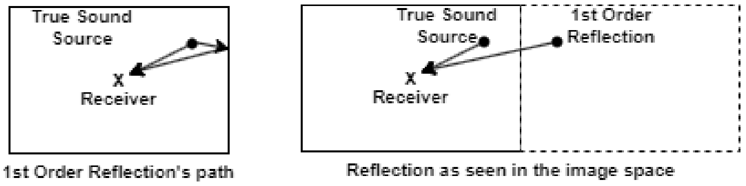

Figure 11: 1st Order Reflection as seen in a slice of the original room and the image space

Based on an estimate of the absorption coefficient, α, of that boundary, a resulting frequency dependent reduction on attenuation can be modelled for the first order reflection, that is a reflection path which intersects with a single reflection, as seen in Fig 11.

This can be further expanded for higher orders, by creating a higher order image space by further reflections, as shown in Figure 12. The resulting higher level of surface absorption effects on the signal are known through the number of intersections with boundaries in image source path, where for each intersection that boundary's α is applied.

With this information it is possible to simulate a convincing monophonic reverberation by creating a finite impulse response (FIR) filter consisting of the sum of this series

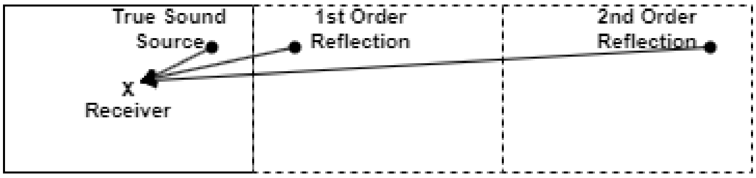

Figure 12: 2nd Order Reflection as seen in a slice of the image space

of reflections. As reverberation is an infinite system, this does require truncation of the maximum order for which this is calculated. This can be achieved either by defining maximum order as a parameter, or a maximum length of time which is used to truncate at the order in which propagation time exceeds the maximum length.

In the context of binaural audio, another useful property of reflections is present when modelling with the image source method: the direction of arrival of every reflection is known as this is the same as the angle of the receiver and mirrored source. This means that HRIR convolutions can be applied to each reflection to create a binaural room impulse simulation. However, as this involves a convolution for every direction, for higher orders this method is computationally expensive.

Scattering Delay Network

The scattering delay network (De Sena *et al.*, 2011; De Sena *et al.*, 2015) is a related concept, which simulates rooms based on real geometries and acoustic parameters, but which also introduces a trade-off of gaining increased computational efficiency for reduced accuracy.

The scattering delay network method considers only first order reflections, calling the intersection points of these first order reflections for room nodes. Path lengths are then calculated for every combination of node, source and receiver. A first order reflection can be modelled as [source → node → receiver], and a second order reflection can be modelled as [source → node → node → receiver], and so forth. These paths are illustrated in Figure 13.

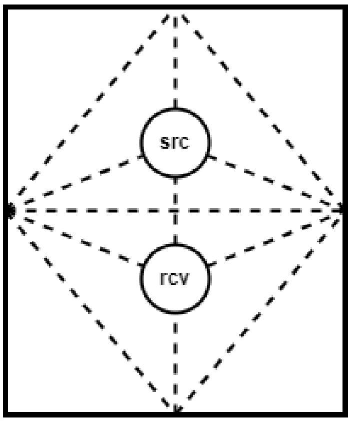

Figure 13: Paths in Scattering Delay Network.

This is computationally efficient, as this can be implemented solely through delay lines without the use of FIR filtering, however for higher orders this approximation of path length becomes increasingly inaccurate.

This can be binauralised using the directions of the nodes relative to the receiver, however this results in a relatively small number of unique directions compared to the number found when using the ISM.

Wave Based Modelling

ISM and scattering delay networks are based on geometric acoustics, which assume that sound propagates as rays. This approximation is valid primarily when surfaces are smooth and significantly larger than the wavelength of the sound, such that wave phenomena like diffraction and interference can be neglected.

A more accurate simulation of sound propagation can be achieved by solving the full wave equation, which provides a true representation of sound's behaviour, including wave phenomena such as diffraction.

There are several well established approaches to wave based modelling (Murphy *et al.*, 2007), but in the context of the simulation of room acoustics the most typical approach is finite element method (FEM) modelling. In FEM, first the target 3D space is modelled. This can include not only the boundaries used in the geometric methods to create a room, but also objects placed within that room for a more complex model. Surfaces and boundaries are assigned material properties which determine how sound will interact with them.

Following this, the model is discretised to create a mesh of elements. Then at least one node is designated a source, from which sound is emitted, and at least one node is designated a receiver, at which sound arrives.
In FEM, the wave equation is then solved to find pressure at every element, for every target frequency. The number of operations involved in this makes FEM very computationally expensive relative to the previously introduced geometric methods.
Finally, the output is the pressure at the receiver element. Combining these outputs across frequencies together constructs a transfer function which describes the difference from source to receiver.
Modelling sound as rays has the benefit of providing knowledge of the reflections' DoAs which can be used to select HRIR for spatialisation. To spatialise the output in FEM another approach must be taken: the head and its pinnae must be added to the model and the receiver elements placed at the ears. To accurately render an HRTF in this manor requires a very detailed model, and high frequency resolution, which significantly increases the computational expense of this approach.

2.2 Digital Signal Processing and Analysis

This subchapter provides a summary of standard digital signal processing formulations. These formulations are widely available in standard texts (Oppenheim and Schafer, 2009; Smith, 2007).

A continuous time-series signal, $x(t)$ can be converted into a discrete time signal, $x[n]$, through the process of sampling. The highest frequency available in the discrete time signal, otherwise named the Nyquist frequency, is determined by the sampling rate f_s by the relation:

$$f_N = \frac{f_s}{2} \tag{18}$$

The cross correlation measures the similarity between two signals as a function of time lag, computed by finding the integral of the product of the two signals. This operation can be expressed with the $\star$ operator.

$$R_{x_1x_2}(\tau) = x_1(t) \star x_2(t) = \int_{-\infty}^{\infty} \overline{x_1(t)} x_2(t+\tau).dt \tag{19}$$

where x_1 and x_2 are the two signals, $\overline{x_1(t)}$ is the complex conjugate of $x_1(t)$, and τ is the time lag. This has an equivalent expression for discrete-time signals:

$$R_{x_1x_2}[k] = x_1[n] \star x_2[n] = \sum_{n=-\infty}^{\infty} \overline{x_1[n]}x_2[n+k].dt \tag{20}$$

where k is the discrete time lag.

Equation (12) introduced the use of convolution in binaural rendering. A more complete definition of convolution can be given as:

$$(x_1 * x_2)(t) = \int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)\,d\tau \tag{21}$$

and equivalently for discrete time signals:

$$(x_1 * x_2)[n] = \sum_{m=-\infty}^{\infty} x_1[k]x_2[n-k]$$

It is notable that unlike cross-correlation, the convolution operation is commutative, that is to say:

$$(x_1 * x_2)(t) = (x_2 * x_1)(t) \tag{22}$$

All these continuous and discrete time signals exist within the time-domain. It is imperative to also consider signals in the frequency domain. Conversion between the two is achieved by Fourier transform:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \tag{23}$$

where j is the imaginary unit, and ω is angular frequency of the relation $\omega = 2\pi f$. The corresponding inverse fourier transform (IFT) is defined as:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\,d\omega \tag{24}$$

The discrete Fourier transform (DFT) meanwhile is defined as:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn} \tag{25}$$

with k in this case being an integer value representing frequency. Note that typically in this work, ω is still used as the independent variable so as to make more clear when frequency domain is being used. The inverse DFT then is defined as:

$$x[n] = \frac{1}{N}\sum_{m=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn} \tag{26}$$

If DFT is applied to two signals, $x_1[n]$ and $x_2[n]$, to give $X_1[k]$ and $X_2[k]$, the cross power spectrum (CPS) of the two signals can be obtained by multiplication of one of the spectra with the conjugate of the other.

$$S_{x1x2}[k] = \overline{X_1[k]}X_2[k] \tag{27}$$

This is to say that the CPS is the Fourier transform of the cross correlation of the two signals:

$$S_{x1x2}[m] = \text{DFT}\{R_{x1x2}[k]\} \tag{28}$$

Signals are created and altered by systems. Some important features of systems are their linearity, and time variance.

A system is linear if it adheres to the superposition principle, and the homogeneity principle. The superposition principle states that the output of the system for the sum of multiple inputs is exactly equal to the sum of outputs for the same inputs processed separately:

$$S\{x_1[n] + x_2[n]\} = S\{x_1[n]\} + S\{x_2[n]\} \tag{29}$$

where S is the system.

The homogeneity principle states that if the input of a system is scaled by a constant, the corresponding output of the system must be equivalently scaled:

$$S\{a \cdot x[n]\} = a \cdot S\{x[n]\} \tag{30}$$

where a is the scalar.

A system is said to be time-invariant if its behaviour does not change over time. Such that:

$$y[n - n_0] = S\{x[n - n_0]\} \tag{31}$$

where n_0 is a time period, which could be of any positive value.

Systems which meet both these conditions, linearity and time invariance, are said to be linear time-invariant (LTI) systems. If a system is LTI, its behaviour in the time domain can be characterised by an impulse response; that is the output of the system to an impulse.

$$h[n] = S\{\delta[n]\} \tag{32}$$

where $\delta[n]$ is the unit function:

$$\delta[n] = [1, 0, 0, 0, 0, \ldots] \tag{33}$$

One property of note, is that the output of the system will be equal to convolving the input with $h[n]$:

$$y[n] = x[n] * h[n] \tag{34}$$

So that if $x[n] = \delta[n]$ then $y[n] = h[n]$.

This can also be equivalently defined in the frequency domain.

$$Y[\omega] = X[\omega] \cdot H[\omega] \tag{35}$$

$H[\omega]$ can therefore be described as the system's transfer function, as it defines the output with respect to the input. The human head is an LTI system, and as such can be described with a transfer function: the previously introduced HRTF.

$$p_{ear[ch]} = p_{src} \cdot \mathrm{HRTF}[\omega, ch, r, \theta, \varphi] \tag{36}$$

where p_{ear} is pressure at the ears, and p_{src} is pressure at the source.

Impulse responses can be either finite or infinite in duration. An FIR only feed-forwards, allowing Equation (34) to be re-expressed as:

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + \ldots + b_Nx[n-N] \tag{37}$$

This is shown as a block diagram in Figure 14.

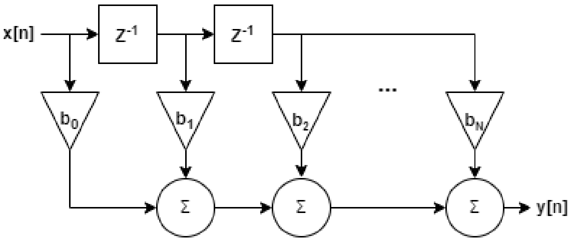

Figure 14: FIR Filter

A notable property of FIR filters it that $b_0 = h[0]$, $b_1 = h[1]$ and so forth until $b_N = h[N]$.

infinite impulse responses (IIRs), on the other hand, are infinite in duration, due to presence of feedback in the system. They can be expressed as:

$$\begin{aligned} y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + &\ldots + b_Nx[n-N] \\ -a_1y[n-1] - a_2y[n-2] - &\ldots - a_My[n-M] \end{aligned} \tag{38}$$

This is shown as a block diagram in Figure 15.

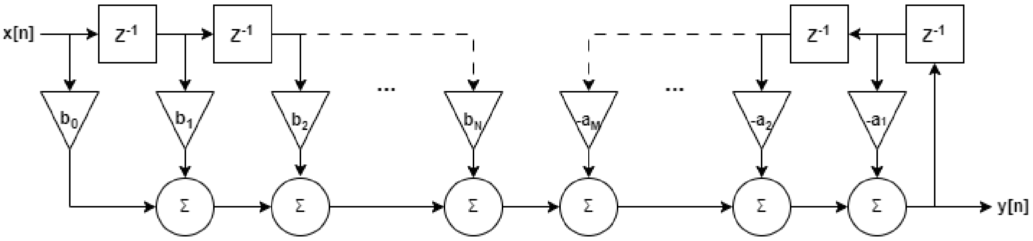

Figure 15: IIR Filter

Time domain signals can also be converted to the time-frequency (TF) domain, in which the signal takes a matrix form with one axis representing time and one dimension representing frequency. This can be achieved by a short-time fourier transform (STFT). This operates by applying Fourier transforms to discrete windows of the input signal, as given below for an input signal $x[n]$

$$X[m, \omega] = \sum_{n=-\infty}^{\infty} x[n] \, w[n - mH] \, e^{-j\omega n} \tag{39}$$

where $X[m, \omega]$ is the output matrix of complex values, m is the frame index, H is the hopsize, and $w[\cdot]$ is the windowing signal. Hopsize refers to the number of samples the centre of the window is moved each frame: a high hop size reduces the amount of overlap between between frames. This can reduce redundancy, but also gives a sparser sampling of the signal.

The TF-matrix could then be turned into a spectrogram by taking the magnitude, in order to give a way to visually interpret the magnitude response of the signal over time.

$$\text{spectrogram}\{x[n]\} = |X[N, \omega]| \tag{40}$$

Additional transformations could be applied to find different characteristics within the source signal. For example, a power spectrogram can be found by taking the square of the magnitude:

$$\text{power-spectrogram}\{x[n]\} = |X[N, \omega]|^2 \tag{41}$$

A log-magnitude spectrogram can be found by taking the logarithm of the magnitude:

$$\text{log-magnitude spectrogram}\{x[n]\} = \log(|X[N, \omega]|) \tag{42}$$

Or, the frequency make up of the spectrogram could be transformed by taking the matrix multiplication of the spectrogram and a filterbank

$$\text{filterbank-spectrogram}\{x[n]\} = |X[N, \omega]| \cdot F[\omega, K] \tag{43}$$

where $F[\cdot]$ is a bank with K number of filters.

Spectrograms provide a powerful way of visualising frequency content over time, however they can be ambiguous if the input contains multiple convolved sources of variation; a typical example of this being the vocal tract and glottal source in speech.

Cepstral analysis was developed to analyse such signals, wherein the IFT is taken of the log of the magnitude of the spectrum. In the TF domain this is given as:

$$\text{cepstrum}\{x[n]\} = \mathcal{F}^{-1}\left\{\log\left(|X(N,\omega)|\right)\right\} \tag{44}$$

where $\mathcal{F}^{-1}$ is the IFT. This IFT does not return the signal from frequency domain to time domain, but rather transforms it into the abstract quefrency domain, and so you the resulting matrix represents time and quefrency. A low quefrency relates to slow-varying components of a signal, while a high quefrency relates to quickly-varying components of a signal. This is helpful in the analysis of speech signals as the vocal tract shape is a slowly changing modifier, and the glottis is a fast-vibrating source, and so when analysed with a cepstrum these previously convolved sources are now seen as two additive components existing at different quefrency ranges.

2.2.1 Sound Source Localisation

sound source localisation (SSL) refers to estimation of a sound source's DoA based upon measurement of the sound field; an application of digital signal processing. As per human cognition, this typically involves exploiting differences seen between different sensors caused by wave propagation. Unlike the human auditory system, however, such systems are not necessarily limited to two sensors. Owing to this, it is common for conventional SSL approaches to improve localisation accuracy through use of a large number of transducers in an array.

Time Difference on Arrival Estimation

TDOA refers to the delay of a sound wave arriving at spaced sensors. This, therefore, is functionally equivalent to the ITD, however does not refer to biological systems, and is not limited to two sensors.

For a sound source, $x(t)$, we can describe the sound arriving at the transducer as:

$$y_n(t) = x(t - D_n) \tag{45}$$

where n is the transducer number, and D is the time delay, equivalent to:

$$D_n = d_n.c_0 \tag{46}$$

where d is the distance from the transducer to the sound source. For each transducer, then, time lag will be different.

The difference in τ between two sensors can be estimated using cross-correlation (Nuttall *et al.*, 1974).

$$R_{x_1x_2}(\tau) = x_1(t) \star x_2(t) \tag{47}$$

where τ is the time lag. The time delay estimate D is then found as the value of τ at which the function maximises.

$$D = \arg\max_{\tau}(R_{x_1x_2}(\tau)) \tag{48}$$

The condition in Equation (48) assumed that the signals x_1 and x_2 are highly correlated. In the context of audio, there are some conditions where this is not necessarily true, for example in the presence of high levels of noise and reverberation. Due to this the approach can lack robustness.

2.3 Machine Learning, Neural Networks, and Deep Learning

Machine learning is a field of study which aims to develop statistical models and algorithms capable of autonomously mapping inputs to outputs on unseen data, based on relationships learned from seen data. This is typically achieved through a training phase, during which a model learns from training data using an optimization algorithm, followed by an inference phase, where the trained model is used to make predictions on unseen data.

The approaches, techniques, models, and formulations presented in this subchapter are consistent with standard treatments found in literature (Bishop and Nasrabadi, 2006; Goodfellow *et al.*, 2016). Approaches to machine learning differ, and can be categorised into four major types:

Supervised learning The model learns from labelled data, mapping inputs to known outputs.

Unsupervised learning The model identifies patterns in unlabelled data.

Semi-supervised learning The model is provided both labelled and unlabelled data.

Reinforcement learning The model learns by taking actions, and is rewarded or penalised based upon its performance compared to the desired outcome.

2.3.1 Machine Learning Models

The model used in machine learning is often dependent on the desired output. To map inputs to continuous outputs, it is typical to use regression; this being a supervised learning approach wherein a dependent variable is mapped to independent variables by finding a function which best fits the relationship between the two. Most typical examples include linear regression in which the relationship is matched with a linear equation, or polynomial regression in which higher order polynomials are used instead.

Instead of a continuous variable, often the desired output is a categorical label: models designed to map inputs to these categorical outputs are called classifiers. Typically to be deemed classification, learning must be supervised. Examples of classification machine learning (ML) models include support vector machines, decision trees, and naïve Bayesian classifiers, k-Nearest Neighbours classifiers.

Clustering also maps inputs to categorical outputs, however in clustering the inputs are not labelled; and as such it is considered an unsupervised approach. Examples of clustering models include k-means clustering, and hierarchical clustering.

Another desired output could be a lower-dimensional representation of the input, for this dimensionality reduction models can be employed. A common example of this is principal component analysis.

Another desired output may be the generation of new data which resembles the data found in the training dataset, these being referred to as generative models. Examples include gaussian mixture models (GMMs) and hidden Markov models.

This list of types of machine learning model is not exhaustive, but also it is important to note that models that have here been identified with one type of output and training may be reapplied elsewhere; for example, GMMs are often associated with clustering (Najar *et al.*, 2017).

2.3.2 Feature Extraction

While different types of output have been introduced, the nature of the inputs remains to be examined. It is often unproductive to attempt to train machine learning models on raw data. A good example of this is audio. Take, for example, the task of SSL in which the aim is to map input audio to a a location in space. It would be possible to achieve this by directly training the model on raw time domain waveform audio, however doing so is not a typical approach as the waveform audio is both large in its number of data points, and lacks an inherent structure from which the model can easily recognise patterns.

Due to this, it is essential to transform the input data to a representation with lower-dimensionality. This is often achieved using signal processing and statistical methods which have been hand-picked for the task so as to highlight relevant features in the raw data; doing so reduces the complexity required in the model, allowing for the more successful learning of input-output relationships. To continue with the example of audio, an audio file initially containing several thousand samples, may be reduced to just a few features.

2.3.3 Training

Machine learning models are modified to achieve specific tasks through training; this refers to the process in which the model learns the relationships between input and outputs from a training dataset. To achieve this, a cost function is defined which describes the model's ability to capture patterns in the data: the system would be behaving perfectly when this function is minimised.

An optimisation algorithm seeks to minimise the cost function by adjusting model parameters, leading to a model trained for the task. Optimisation in ML can be closed-form, directly calculating the optimal solution, however typically cost functions are non-convex and so require an iterative approach.

2.3.4 Datasets

In the design and creation of a machine learning system, models are typically trained and then evaluated. To do this effectively, separate datasets for training and testing must be constructed. In addition to this, it is also common to create a validation dataset for use during training.

The training dataset contains the data from which the model should be able to learn to map input to output relationships. A strong training dataset should be large, highly relevant to the task, accurately annotated if labelled, diverse such that it covers a realistic variety of situations, and well balanced in terms of its distribution across possible outputs.

The testing, or evaluation, dataset is auxiliary to the testing dataset. Rather than being used in training, it is used in evaluation in which the dataset is used predict outputs on the data, and the performance of the system is measured by a relevant metric. If a system performs poorly on both the training and testing datasets, it is said to have underfit and has not successfully identified patterns between the input and outputs. If a system performs well on the training dataset, but poorly on the testing dataset, it is said to have overfit, and has learned patterns between inputs and outputs which are unique to the training dataset.

A good testing dataset should be representative of the real-world conditions in which the model will be applied, as this provides the ability to evaluate the system under realistic conditions. To be able to identify overfit, it is also important that the testing dataset contains unseen data; that being data which is completely separate and independent of the training data.

Lastly, it is also common to construct a validation dataset. A validation is a process in which between iterations during training, the model is used to predict outputs on a small validation dataset. The validation dataset can be used as proxy for the testing dataset, providing real-time insight into underfit and overfit in the system during training.

A good validation dataset should possess the same characteristics as a good testing dataset: good representation, and meet the unseen condition. The validation dataset, however, should also be independent from the testing dataset. This is because the role of validation in the training process can lead to a bias towards the validation dataset, something which can only be uncovered in final evaluation if there is adequate independence between the validation and testing datasets.

2.3.5 Evaluating Machine Learning Systems

Machine learning models are evaluated using performance metrics, which can be similar to the cost function but differ in that they are not used in the training process. These metrics vary depending on the task and type of output. Below are some common performance metrics for classification and regression models.

Classification

Below are some common categorical performance metrics. To better understand these metrics it is important to understand that all categorical predictions can have one of four outcomes

True Positive (TP) The ground truth is positive, and the prediction is positive.

True Negative (TN) The ground truth is negative, and the prediction is negative.

False Positive (FP) The ground truth is negative, but the prediction is positive.

False Negative (FN) The ground truth is positive, but the prediction is negative.

Accuracy The rate at which a models predicted category matches the ground truth.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{49}$$

Precision The proportion of true positives among all positive instances, made up of true and false positives. This is an especially useful metric when the cost of false positives is high.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{50}$$

Recall The proportion of true positives among true positives and false negatives. This is an especially useful metric when the cost of false negatives is high.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{51}$$

Regression

Below are some common performance metrics for continuous outputs.

Mean Absolute Error The mean error with directionality removed.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{52}$$

Mean Squared Error The mean of the squares of errors. The square term more heavily penalises large errors.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{53}$$

Root Mean Squared Error The square root of the mean squared error. This retains the heavy penalisation of large errors, but returns the values back to the scale of the data.

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{54}$$

2.3.6 Artificial Neural Networks

A specific type of machine learning model are artificial neural networks. These are models made up of artificial neurons, with these neurons being connected within the model much like nodes within a network. This is inspired by, and analogous to, biological neural networks, however it is important to note that this analogy is somewhat superficial: artificial neurons are not direct models of biological neurons.

Neurons take a series of inputs and calculate an input through a linear transformation and a nonlinear function. The linear transformation consists of a weighted sum and a bias. The nonlinear part is known as an activation function, and applies a pre-selected nonlinear function to the signal.

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{55}$$

where x_i is the series of inputs to the neuron, w_i is a series of weights, b is a bias, $f(\cdot)$ is the activation function, and y is the output.

These biases and weights are the neurons learnable parameters; it is these values which are updated during the training.

The activation function provides the neuron an ability to map nonlinear relationships between inputs and outputs. This is achieved not by changing the activation function itself, but instead by changing the weights and biases which determines how the input interacts with the activation function. Listed below are some common activation functions.

Sigmoid This activation function produces an S-shaped curve by using an exponential function in the denominator, mapping values to the range 0 to 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{56}$$

Hyperbolic Tangent This activation function also produces an S-shaped curve, but within the range -1 to 1

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{57}$$

rectified linear unit (ReLU) In modern ML practice ReLU is the most common activation function. It simply imposes a lower limit of 0 on the signal.

$$\text{ReLU}(x) = \max(0, x) \tag{58}$$

Softmax Unlike the other activation functions, softmax is applied across an entire layer of neurons. This is helpful at the output of a neural network for classification as it creates a probability density.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{59}$$

where the index i refers to the specific neuron, and the index j iterates over all neurons in that layer.

Neurons are then organised into neural networks. Instead of single values, such as at the input of a neuron, consider instead x as being a vector containing features calculated through feature extraction, with a length of n_{inputs}.

The most simple architecture which could be created is to combine neurons in parallel, to create a layer. This can be expressed in matrix form:

$$y = f(Wx + b) \tag{60}$$

where y is the output vector with a length of $n_{outputs}$, and W is a matrix of weights with the dimension $n_{outputs} \times n_{inputs}$. This simple model is named a single layer perceptron (SLP). The shortcoming of SLPs is that because only one linear operation is applied to the input, they are incapable of mapping complex non-linearities.

To overcome this, an additional layer can be added to create a multi-layer perceptron (MLP). This additional layer is referred to as a hidden layer, as it abstracted from the output. Letting h be the output of the hidden layer, this can then be defined as the serialisation of two SLPs:

$$\begin{aligned} h &= f_1(W_1x + b_1) \\ y &= f_2(W_2h + b_2) \end{aligned} \tag{61}$$

It is important to note that as the hidden layer is abstracted from the output, its length is no longer determined by the input or output layers and is therefore arbitrary. This then becomes a hyperparameter which can be tuned to optimise performance.

2.3.7 Cost Functions

For neural networks to be trained, it is essential to introduce a cost function by which the performance of the system can be measured in terms of the degree to which the the model's predictions have deviated from the ground truth.

For classification, this is almost exclusively cross-entropy loss, as defined below.

$$J(\xi) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C} y_c^{(i)} \log(\hat{y}_c^{(i)}) \tag{62}$$

where i is the index of the training data, c is the index of the class, $\hat{y}_c^{(i)}$ is the predicted probability that the training data instance i belongs to the class c, $y_c^{(i)}$ is a binary indicator that i is of c, C is the total number of classes, m is the total number of training examples, ξ are the neural network parameters and $J(\xi)$ is the cost function.

In the case of the MLP described in Equation (61), ξ would refer to all of the weights and biases in the system such that $\xi = W_1, W_2, b_1, b_2$.

It can be seen that this function is only suitable if a finite number of classes exist, hence the need for a classification task. Furthermore, the binary indicator means that all incorrect classifications are treated as exactly equally incorrect.

In regression based problems, a continuous function is required based on error. The functions introduced as evaluation metrics, mean absolute error (Equation (52)), mean square error (Equation (53)), or root mean square error (Equation (54)) could also be used as cost functions in this case.

2.3.8 Backpropagation

Given the cost function, the loss can be computed and used to update the weights W_1, W_2 and biases b_1, b_2 during training. To achieve this, first a forward pass is undertaken wherein an input vector is propagated through the network leading to a predicted output $\hat{y}$. This is compared to the true output as defined by a label, y, in the cost function $J(\cdot)$.

$$L = J(\hat{y}, y) \tag{63}$$

With the loss now available, the objective becomes to compute the degree to which the loss changes when each parameter is adjusted. These are the gradients, and are defined as the derivative of the loss with respect to the parameter, for example the gradient with respect to a weight is defined as $\frac{\partial L}{\partial W}$. The gradient of the output can be defined in relation to the cost function:

$$\frac{\partial J}{\partial \hat{y}} = J'(\hat{y}, y) \tag{64}$$

Then, using the chain rule, the gradients of the weights and biases of the output layer can be found:

$$\frac{\partial J}{\partial W_2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_2} = J'(\hat{y}, y) \cdot h^T f_2'(W_2 h + b_2) \tag{65}$$

$$\frac{\partial J}{\partial b_2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_2} = J'(\hat{y}, y) \cdot f_2'(W_2 h + b_2) \tag{66}$$

Loss is then propagated back to the first layer.

$$\frac{\partial J}{\partial W_1} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial W_1} = J'(\hat{y}, y) \cdot W_2^T f_2'(W_2 h + b_2) \cdot x^T f_1'(W_1 x + b_1) \tag{67}$$

$$\frac{\partial J}{\partial b_1} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial b_1} = J'(\hat{y}, y) \cdot W_2^T f_2'(W_2 h + b_2) \cdot f_1'(W_1 x + b_1) \tag{68}$$

This method of propagating loss back through the network to calculate gradients is called backpropagation.

2.3.9 Optimisation Algorithms

The parameters in the model can then be updated based upon the gradients calculated during backpropagation. This is achieved by means of an optimisation algorithm. Below some typical optimisation algorithms are introduced.

Batch Gradient Descent

Gradient descent is a very common optimisation technique in ML. This starts with the gradients of all of the parameters in the model:

$$\nabla J(\xi) = \begin{pmatrix} \frac{\partial J}{\partial \xi_0} \\ \frac{\partial J}{\partial \xi_1} \\ \frac{\partial J}{\partial \xi_2} \\ \vdots \\ \frac{\partial J}{\partial \xi_n} \end{pmatrix} \tag{69}$$

In gradient descent, steepest descent then is iteratively applied to the parameters. At each iteration, this is such that:

$$\xi^{(t+1)} = \xi^{(t)} - \alpha \nabla J(\xi^{(t)}) \tag{70}$$

where t is the current iteration, and α is a learning rate which weights the change in parameters per iteration. Specifically in batch gradient descent (BGD), these iterations occur once per every pass of the training dataset, so once per epoch.

Stochastic Gradient Descent

stochastic gradient descent (SGD) differs from BGD only in the iteration rate, parameters are updated once per data point rather than once per epoch. This typically leads to a noisier trajectory (in loss over time) but faster convergence.

Minibatch Gradient Descent

minibatch gradient descent (MGD) can be though of as a compromise between batch gradient descent and SGD, wherein parameters update once every nth data point, where n is a predetermined value; often a smaller power of two (i.e. 32, 64, 128).

The exhibited behaviour in MGD is also a compromise between BGD; it is less noisy than SGD but is also less likely to get stuck in local minima than BGD.

Adaptive Gradient Algorithm

adaptive gradient algorithm (AdaGrad) (Duchi *et al.*, 2011) is an adaption of SGD, wherein the learning rate is altered according to the gradient. Starting with a vector initialised with zeroes:

$$G = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{71}$$

at each iteration, t, the gradient of the cost function is calculated as per Equation (69), and then each component of G is updated with the square of the gradient component:

$$G_{t+1,i} = G_{t,i} + \left(\frac{\partial J}{\partial \xi_i^{(t)}} \right)^2 \tag{72}$$

and then this weighting is used to create an adaptive learning rate based upon an initial learning weight α:

$$\alpha_{t,i} = \frac{\alpha}{\sqrt{G_{t,i}}} \tag{73}$$

Root Mean Square Propagation

root mean square propagation (RMSProp) (Tieleman, 2012) addresses a shortcoming of AdaGrad; the tendency towards too heavily decreasing the learning rate. RMSProp is differentiated by finding a moving average of the square of gradient components at each iteration instead of using the square of gradient components.

$$E[g^2]_{t+1,i} = \beta E[g^2]_{t,i} + (1-\beta) \left(\frac{\partial J}{\partial \xi_i^{(t)}} \right)^2 \tag{74}$$

where β is decay rate. This moving average is then used to create an adaptive learning rate:

$$\alpha_{t,i} = \frac{\alpha}{\sqrt{E[g^2]_{t+1,i}}} \tag{75}$$

Adam

Adam optimisation (Kingma and Ba, 2014) is a widely employed technique which builds on AdaGrad and RMSProp. In the Adam optimiser, two moment estimates are made, one based upon the gradient and one based upon the square of the gradient. For a pair of decay rates β this is expressed as:

$$m_{t+1,i} = \beta_1 m_{t,i} + (1 - \beta_1) g_{t,i} \tag{76}$$

and:

$$v_{t+1,i} = \beta_2 v_{t,i} + (1 - \beta_2) g_{t,i}^2 \tag{77}$$

and the parameters are then updated based on a learning rate weighted by the two moment estimates:

$$\xi_i^{(t+1)} = \xi_i^{(t)} - \alpha \frac{m_{t+1,i}}{\sqrt{v_{t+1,i}}} \tag{78}$$

2.3.10 Recurrent Neural Networks

In SLPs and MLPs the relationship between the output is dependent on the input at that moment in time. However, in real-world scenarios dependencies often extend across time, and so the output could be influenced by past inputs. To model these relationships, recurrent neural networks (RNNs) introduce feedback and memory mechanisms which allow a neural network to draw upon past events.

A simple RNN can be built by the introduction of a hidden state to a neural network This is a set of activations dependent on the input, as well as previous activations, as defined below.

$$h_t = f(W_x x_t + W_h h_{t-1} + b) \tag{79}$$

Where h is the hidden state. This hidden state can then be introduced into the previous SLP architecture to create an RNN.

$$y_t = g(W_y h_t + b_y) \tag{80}$$

This produces a series of outputs for a series of inputs. It is possible also to make a single prediction from a series of inputs by taking only the prediction at the last moment of time in the series, as defined below.

$$y = g(W_y h_T + b_y) \tag{81}$$

2.3.11 Deep Learning

A disruptive trend within the field of machine learning is the increased popularity of its subfield deep learning.

The deep in deep learning refers to the size of the neural networks used in this technique, separating input and output with several layers of processing.

A simple deep neural network (DNN) would be to expand the MLP introduced in Equation (61) to include another layer, as shown below.

$$\begin{aligned} h_1 &= f_1(W_1 x + b_1) \\ h_2 &= f_2(W_2 h_1 + b_2) \\ y &= f_3(W_3 h_2 + b_3) \end{aligned} \tag{82}$$

The middle layer, h_2, is referred to as a hidden layer. It is named this because it is entirely abstracted from the inputs and outputs. Increasing the number of layers like this allows for mapping of more complex relationships between input and output.

The popularity of deep learning escalated with the introduction of AlexNet (Krizhevsky *et al.*, 2012), which achieved a significant improvement over previous techniques in a visual object recognition task. Alexnet used five convolutional layers and two fully connected layers, a comparatively large architecture compared to what previously would have been applied.

2.3.12 Convolutional Neural Networks

The neural network layers introduced so far have all contained a matrix multiplication in which the matrix is determined by the size of the inputs and outputs. In terms of Big O notation, this leads to a complexity of $O(N_{input} \cdot N_{output})$. This computation penalty for large inputs explains the need for heavy dimensionality reduction during the feature extraction process, however this approach relies on an ability to extract relevant features, which is not always possible.

Instead of this, it would be beneficial to have an approach which scales with input and output size linearly, as this would more easily allow for operating on raw representations of data. This can be achieved with convolutional layers.

A convolutional layer applies multiple convolutions of part of the input tensor with a kernel, iterating over the input tensor according to a stride size, a technique introduced in LeNet (LeCun *et al.*, 1998) for handwritten character recognition. This can be expressed mathematically. Defining the input to be a tensor, X, which has the size $[m, H_X, W_X, C_X]$ referring to batch size, height, width and number of channels, A convolution kernel is similarly defined as K, with a size $[m, H_K, W_K, C_X, C_Y]$ where C_Y is the number of output channels, and H_K and W_K are typically small values. This can be expressed mathematically in terms of a convolution operation with stride s, where stride is a multiplier applied to the convolution indices.

$$\mathbf{Y} = \mathbf{X} *_s \mathbf{K} \tag{83}$$

where $[m, n]$ iterate over spatial dimensions, k over channels. The output, Y, is referred to as a set of feature or activation maps.

Equation (83) reveals the source of the computational saving; while in fully connected layers each input element must have N_{output} multiplications, here each input element must only be convolved with a kernel.

Another benefit of this approach over a fully-connected layer is that it introduces translation invariance as the kernel is slid over the input, and so is able to identify similar patterns in different parts of the input.

The kernel has a higher number of dimensions than the input; this introduces an extra dimension in the output such that you obtain different a series of features maps from one input tensor. This helps convolutional neural networks (CNNs) map multiple relationships in the data.

Should stride be greater than 1, this will create a reduction in dimensionality due to a sparsity of input connections; this helps balance the increased computation from the additional kernel dimension. This also results in each element in the output being formed by a larger area in the input, which can be a desirable trait. Equation (83) was introduced with three dimensional matrices. This is not a requirement however; it would also be valid to have two dimensional convolution kernel to operate on a two dimensional input, such that $\mathbf{X} \in \mathbb{R}^{m \times H_X \times W_X}$ and $\mathbf{K} \in \mathbb{R}^{H_K \times W_K \times C_Y}$, or similarly a one dimensional filter could be used on an input vector such that $\mathbf{X} \in \mathbb{R}^{m \times N_X}$ and $\mathbf{K} \in \mathbb{R}^{N_K \times C_Y}$.

Strided convolution leads to a slight shrinkage of the width and height dimensions as there are not enough valid elements at the edges of the input to perform convolutions. To account for this, it is typical to zero pad the input to have the effect of making $H_Y = H_K$ and $W_Y = H_Y$.

CNNs layers tend to be made of more operations than just the convolution. Typically proceeding the convolution would be a normalisation operation, to improve stability in the process by reducing the variability between input tensors. This most typically would be achieved with batch normalisation, wherein each batch is normalised based upon that batch's mean and variance. The mean is given as:

$$\mu_c = \frac{1}{mH_XW_X} \sum_{b,h,w} X_{b,h,w,c} \tag{84}$$

where b, h, c, w are the batch, height, width and channel indices respectively. The variance is then given as:

$$\sigma_c^2 = \frac{1}{mH_XW_X} \sum_{b,h,w} (X_{b,h,w,c} - \mu_c)^2 \tag{85}$$

These are then used to normalise the input tensor:

$$\hat{\mathbf{X}} = \frac{\mathbf{X} - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{86}$$

Where $\hat{\mathbf{X}}$ is the output normalised tensor, and ϵ is a small offset introduced to increase numerical stability.

There is then a final operation in which learnable scaling and shifting factors γ and β are applied to the tensor

$$\mathbf{Y} = \gamma\hat{\mathbf{X}} + \beta \tag{87}$$

These factors allow the network to reintroduce a desired scale into the network. This is particularly important in the context of the next operation applied to the tensor: the activation function. This would typically be ReLU or one of the other activations functions introduced in Chapter 2.3.6.

$$\mathbf{Y} = ReLU(\mathbf{X}) \tag{88}$$

Finally, it is also typical to include a pooling operation which reduces the spatial dimensions of the feature maps. The pooling window, defined by a size of $P_H \times P_W$, slides across the feature map, and for each window, the maximum value is selected. The output tensor is constructed from these maximum values.

This process helps extract features, as the maximum operation can extract the most salient value from each pool. Simultaneous to this, it gives another method of control over feature map size, which is useful in terms of reducing computational complexity, but also because changing the number of parameters at different parts of the network can aid in avoiding underfitting or overfitting.

These four operations together make a typical convolutional layer.

$$\text{Convolution} \longrightarrow \text{Normalisation} \longrightarrow \text{Activation Function} \longrightarrow \text{Pooling}$$

A CNN would then typically cascade many of these convolutional layers together, before using fully connected layers at the end of the model to combine all the feature maps, allowing the model to use the entire receptive field for its final predictions.

2.3.13 LSTMs and GRUs

A difficulty encountered in deep learning, particularly when using RNNs, is the vanishing gradient problem. This refers to the tendency of gradients to become smaller as the loss is backpropagated through each layer of the network. In the case of RNNs, this issue is particularly prevalent as the network's output depends on previous time steps, meaning the gradients must be backpropagated through time, not just through layers. As the gradients are propagated back through many time steps, they are multiplied by the derivatives of the activation functions at each step. Since these derivatives are typically less than 1, this results in the gradients shrinking exponentially, making it difficult for the network to learn long-term dependencies.

To tackle this issue the long short-term memory (LSTM) layer was devised (Hochreiter and Schmidhuber, 1997); this being a recurrent layer which uses memory cells and gates instead of typical feedback loops, which are able to map long term dependencies without the many iterations which result in vanishing gradients.

Memory cells are stored variables which are updated based on three types of gates: forget, input and output gates. These are all have a similar definition:

$$i_t = \sigma(W_i[h_{t-1}, x] + b_i) \tag{89}$$

$$f_t = \sigma(W_f[h_{t-1}, x] + b_f) \tag{90}$$

$$o_t = \sigma(W_o[h_{t-1}, x] + b_o) \tag{91}$$

where $[i, f, o]_t$ are the hidden gates at the present time step, $\sigma(\cdot)$ is the sigmoid function, W_i, W_f, and W_t are matrices of learnable weights with a size of $N_h \times (N_h + N_x)$ where h refers to the hidden units, h_{t-1} is the hidden state at the previous time step, and b_i, b_f, and b_o are learnable biases. Due to the sigmoid function, each of these equations returns a value between 0 and 1. These values are used to determine the flow of data in and out of the LSTM over time.

First, a candidate cell state is created from the input.

$$\tilde{C}_t = \tanh{(W_C)}[h_{t-1}, x_t] + b_C \tag{92}$$

where W_C and b_C are learnable weights and biases.

The LSTM's memory cell state is then updated based upon this this candidate cell state and the previous cell state, as scaled by the input and forget gates

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{93}$$

Finally the output, referred to as the hidden state, is created from the hyperbolic tangent of the cell state, scaled by the output gate.

$$h_t = o_t \odot \tanh C_t \tag{94}$$

The cell state, as shown in Equation (92), is what mitigates the issue of vanishing gradients. As the cell state does not itself contain an activation function, which is the main cause of the attenuation of gradients, the cell state are able to store gradients over long durations without attenuation. This allows the model to learn longer term dependencies.

Compared to conventional RNNs, however, LSTMs significantly increase the complexity in terms of number of parameters and operations involved. The gated recurrent unit (GRU) is a recurrent layer also based on gating, which reduces the complexity of LSTMs.

GRUs contain two gates: the update gate and the reset gate. These are calculated similarly to the gates in LSTMs, but the input and previous hidden states are summed rather than concatenated.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{95}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{96}$$

where z_t is the update gate, and r_t is the reset gate, W_z and W_r are learnable weights for the current input, U_z and U_r are learnable weight matrices for the hidden units from previous time step, and b_z and b_r are learnable biases. A candidate hidden state is then created:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \tag{97}$$

where W_h and U_h are weight matrices for the input and hidden states. Following this the hidden state is calculated.

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{98}$$

2.3.14 Regularisation

The complexity of deep learning models causes a great likelihood of overfitting, as they are capable of very fine details in inputs to output predictions. One method of mitigating this is to simply reduce the number of parameters during training: this is achieved with dropout, a technique which sets a certain proportion of the model's parameters to zero in the forward pass so that they do not contribute to the cost function.

Additionally, it is possible to regularise a neural network by summing an additional term into the loss calculation, typically penalising large parameters. This term can be introduced to the loss calculation given in Equation (63), as shown below.

$$L = J(\hat{y}, y) + R(\xi) \tag{99}$$

where $R(\cdot)$ is the regularisation function. This regularisation function could be applied to all weights in the model as seen here, or only to specific layers. The two most common regularisation functions are L1 and L2 regularisation, which are defined below.

L1, or lasso, regularisation penalises based upon the absolute values of the parameters.

$$R_{L1}(\xi) = \lambda \sum |\xi| \tag{100}$$

where λ is a hyperparameter which controls the severity of the penalty.

Meanwhile L2, or ridge, regularisation penalises based upon the square of the parameters.

$$R_{L2}(\xi) = \lambda \sum \xi^2 \tag{101}$$

This has the effect of more considerably penalising large parameters relative to small parameters.

Large parameters tend to induce large changes in output based on small changes to the input, which is why these methods are able to tackle overfitting. However, aggressive regularisation may just as easily induce underfitting, as the model is penalised for containing anything but small weights, which are only capable of mapping basic relationships.

2.3.15 The Utility of Machine Learning

Many real-world systems are too complex, dynamic, or noisy to be described by clean, mathematical equations. Attempting to model or handcraft algorithms in these scenarios can often lead to solutions which are non-robust in real scenarios. It is in cases like these that machine learning is often best applied: when a scenario lacks an analytical solution, machine learning can be capable of finding meaningful hidden patterns in data which can be used to make predictions.

Deep learning furthers this by allowing models to make predictions from high-dimensional and unstructured representations of data, meaning that it becomes unnecessary to find features which are optimal for a task.

These capabilities have led to deep learning being widely adopted in many domains where traditional signal processing techniques were once dominant.

2.3.16 Machine Learning APIs

Software implementation and training of neural networks often happens by using toolboxes, which typically provide a way of building neural networks through common types of layers, and their associated parameters, as well as the algorithms for training and testing models, as well as tuning parameters. Common examples are listed below.

- Tensorflow (Abadi *et al.*, 2016)
- PyTorch (Paszke *et al.*, 2017)
- Scikit Learn (Pedregosa *et al.*, 2011)
- Keras (Gulli and Pal, 2017), integrated into Tensorflow
- Matlab Machine Learning Toolboxes (Paluszek and Thomas, 2016)

2.3.17 Speech Corpi

A significant application of ML is to speech processing, including but not limited to automatic speech recognition (ASR), speech synthesis, and speaker recognition. These tasks require use of large datasets of labelled speech data; speech corpi.

This thesis largely concentrates on speech localisation, and so some of these speech corpi are introduced.

TIMIT

TIMIT (Garofolo *et al.*, 1992) is a popular speech corpus containing monophonic speech of 630 speakers of American English. The audio in TIMIT is uncompressed, however is sampled at 16kHz and is therefore bandlimited to 8kHz.

Librispeech

Librispeech (Panayotov *et al.* 2015) is a large speech corpus made up of English speech samples taken from public domain audiobooks. This improves upon TIMIT in terms of scale, however it is also sampled at 16kHz.

telecommunications and signal processing laboratory (TSP)

The TSP corpus (Kabal, 2002) also consists of English language recordings. As per TIMIT, these have been deliberately recorded for the dataset. An advantage of this is that they have been able to evenly split the speakers between male and female, something which is not true of Librispeech (Garnerin *et al.*, 2021). However, the dataset suffers in variety, with only 24 speakers being present.

TSP has a higher sampling rate of 48kHz, representing the entire audible spectrum.

2.3.18 Related applications of ML

There exists other audio related tasks which similar to SSL have found success in the application of deep learning. Some of these are listed below.

Blind Source Separation

Consider an audio signal, with any number of channels. It consists of the sum of some number of individual sound sources. However once these signals have been summed, this process is not easily reversed without knowledge of at least some of the original signals. Blind source separation refers to the task of reversing this sum.

Conventional techniques for this task include independent component analysis (ICA) (Comon, 1994), and non-negative matrix factorisation (NMF) (Lee and Seung, 1999), and methods derived from these original methods (Sawada *et al.*, 2019).

This is a field where machine learning, and now especially deep learning, have been applied widely (Ansari *et al.*, 2023).

Blind source separation and SSL are interlinked by a complementary relationship; accurate sound localisation can be used to improve sound separation (Nikunen and Virtanen, 2014), and likewise accurate sound separation can be used to improve sound localisation (Pu *et al.*, 2021).

Just as BSSL is a subfield of SSL concerning binaural signals, there exists also a subfield of blind source separation concerning separation within binaural signals (Alinaghi *et al.*, 2013; Alinaghi *et al.*, 2014)

Computational Auditory Scene Analysis

There is an area of understanding of human audition referred to as auditory scene analysis (ASA). ASA addresses how humans hear complex auditory scenes, with many sound sources, each of which themselves which could be described as complex. Humans have the ability within these scenes to pay selective attention to a sound source of interest, the cocktail party effect (Cherry, 1953). How humans are able to manipulate such complex scenes is the basis of research on ASA, which looks at the underlying mechanisms, including binaural sound localisation, which can allow for such effects (Bregman, 1994).

computational auditory scene analysis (CASA), then, seeks the same end of complex sound scene understanding, by employing computational versions of the same underlying functions. Given its inspiration from human audition, it is unsurprising that this is often, although far from exclusively, achieved using binaural signals; this leads to BSSL being an important aspect of CASA (Wang and Brown, 2006).

2.4 Review of Binaural Sound Source Localisation Literature

Binaural Sound Source Localisation is an already an established task due to interest stemming from its possible applications in a number of fields; this chapter looks at relevant previous work. This begins with an overview of publicly available HRTF datasets; an important topic due to the frequency of their use in BSSL research, for which no comprehensive and current overview exists.

Following this, an overview of the works on the topic of BSSL is given. This introduces techniques used prior to the emergence of deep learning based approaches, then gives an exhaustive survey on deep learning approaches to this task, showing the themes which have emerged from the research.

2.4.1 HRTF Datasets

This work, and other works on BSSL, depend heavily on the use of HRTF measurement datasets for the synthesis of training and testing binaural datasets. To this end, a short review of published datasets is presented.

Datasets are categorised as consisting of either measurements or simulations. The measured datasets are further categorised by whether they consist of measurements of head simulators or human subjects.

For each dataset, some background on the measurement approach is presented, with any information which could make the HRTF exceptional, and plots of the source positions found on horizontal and median planes are presented.

Measured Head Simulators

The datasets presented in this subchapter were measured using artificial heads. They are listed by the institution at which they were measured, owing to the strong significance of this variable caused by the uniqueness of measurement apparatus between research institutions, which has a strong influence on the resulting datasets.

It is common for datasets containing multiple subjects to also include HRTFs of head simulators in the same dataset. Examples of these are listed in Chapter 2.4.1.

Massachusetts Institute of Technology, USA

The MIT KEMAR HRTF Measurements (Gardner, Martin, *et al.*, 1994) consist of freefield measurements of a KEMAR binaural array, measured with loudspeakers mounted on a sphere 1.4 metres from the receiver. HRIRs were measured for 710 positions, including the full azimuthal plane, with elevation restricted to -40° to +90°. These positions are shown in Figure 16. As this was a pioneering HRTF dataset, its usage in binaural research has been extensive.

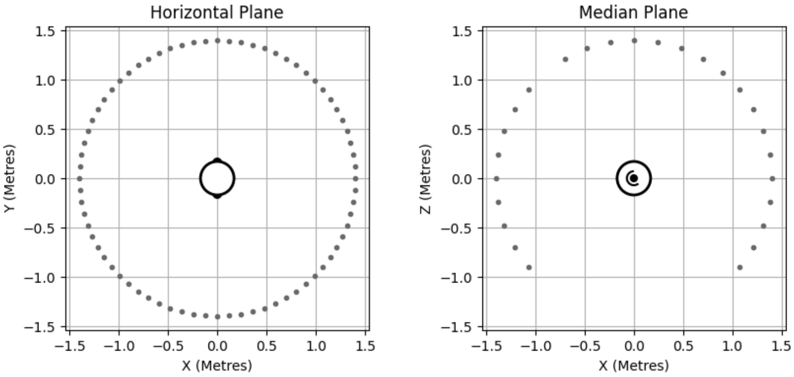

Figure 16: Horizontal and Median Plane Source Positions in MIT KEMAR HRTF Dataset

TH Köln, Germany

Researchers at TH Köln have published multiple sets of HRTF Measurements. Firstly, a set of farfield measurements of a KU100 binaural array (Bernschütz, 2013). These were measured for several sampling configurations, including a set forming a circle on the horizontal plane with 1° sampling spacing, measured by rotating the array with a turntable, as shown in Figure 18. In addition to this, three full-sphere measurement grids, measured with a robotic arm with 3 degrees of freedom. These sampling grids all represented the full sphere, with 2354, 2702, and 16020 sampling positions as shown in Figure 17.

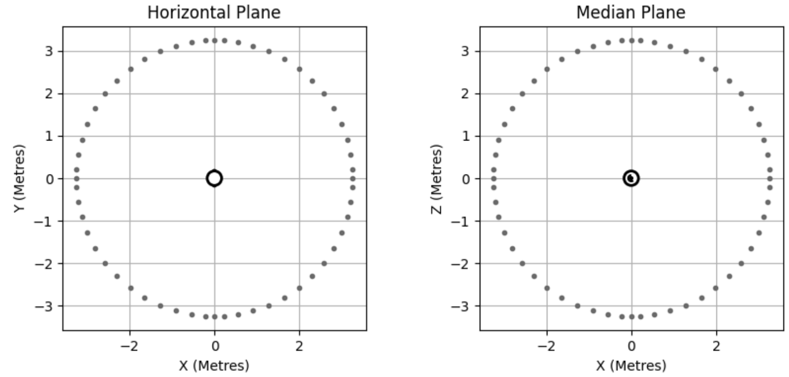

Figure 17: Horizontal and Median Plane Source Positions in Koln KU100 Spherical HRTF Datasets

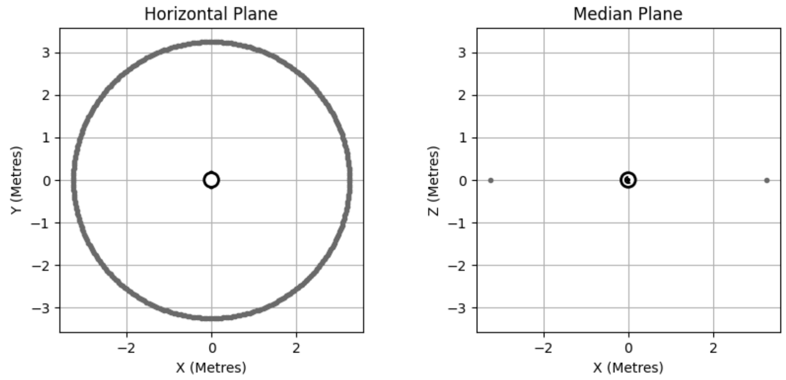

Figure 18: Horizontal and Median Plane Source Positions in Köln KU100 Horizontal Plane HRTF Dataset

A second set of KU100 HRTF measurements was later published, also including horizontal plane and full sphere sampling configurations, but now for multiple source distances (Pörschmann *et al.*, 2017). Specifically, spherical grids of 2702 positions were sampled for four radii in the range 0.5m to 1.5m, and additionally the horizontal-plane circular sample grid with sampled with 1° spacing, was sampled for five radii from 0.25m to 1.5m (Arend *et al.*, 2016) as shown in Figures 19 & 20.

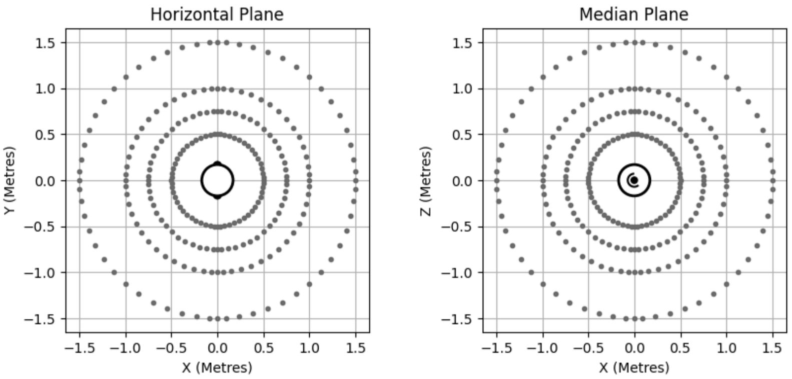

Figure 19: Horizontal and Median Plane Source Positions in Köln KU100 Nearfield Spherical HRTF Dataset

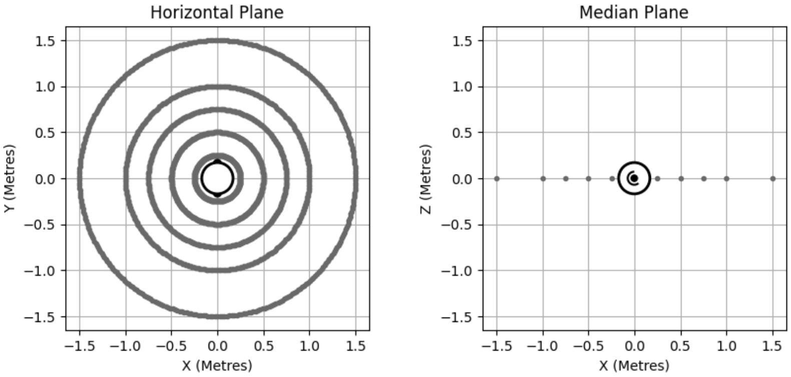

Figure 20: Horizontal and Median Plane Source Positions in Köln KU100 Nearfield Horizontal HRTF Dataset

Another similarly sampled dataset has been measured containing HRTFs of both a KU100 and a Head Acoustics simulator, but wearing various types of headwear: a cap, a helmet, a VR headset, and a pair of over-ear headphones (Pörschmann *et al.*, 2019).

TU Berlin, Germany

Two sets of HRTF measurements have been published by researchers at TU Berlin.

As at Köln, a multiple distance measurement was also made at TU Berlin, but of a KEMAR head simulator wearing the large pinnae attachments (Wierstorf *et al.*, 2011). The measurements were made in circular arrangements on the full horizontal plane, by rotating the head simulator in 1° sampling intervals, which were measured for four radii: [0.5m, 1m, 2m, 3m], as shown in Figure 21.

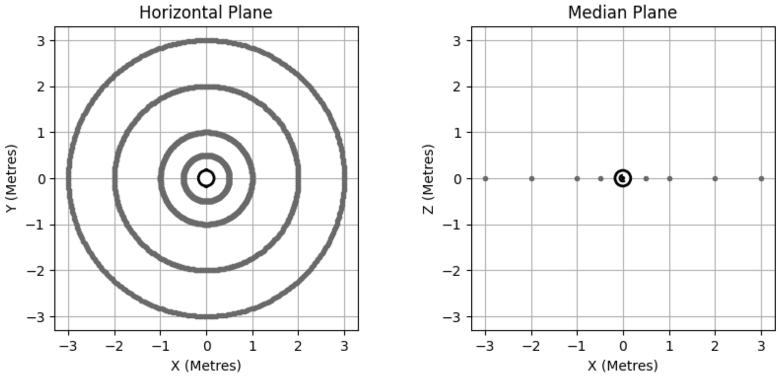

Figure 21: Horizontal and Median Plane Source Positions in TU-Berlin KEMAR HRTF Dataset

The second dataset is of measurements of the FABIAN head simulator previously introduced in Chapter 2.1.6. These differ from other datasets introduced so far in that they not only measure for different rotations of sound source, but also measure for different rotations of the simulator's head relative to the torso (Brinkmann *et al.*, 2017). 11 head orientations relative to the torso, which is termed as a head above torso orientations (HATOs), in the range -50° to 50°. At each of these orientations, the entire simulator was rotated so as to sample a full sphere sampling grid of 11950 positions. The horizontal and median plane positions for one of these head rotations are shown in Figure 22.

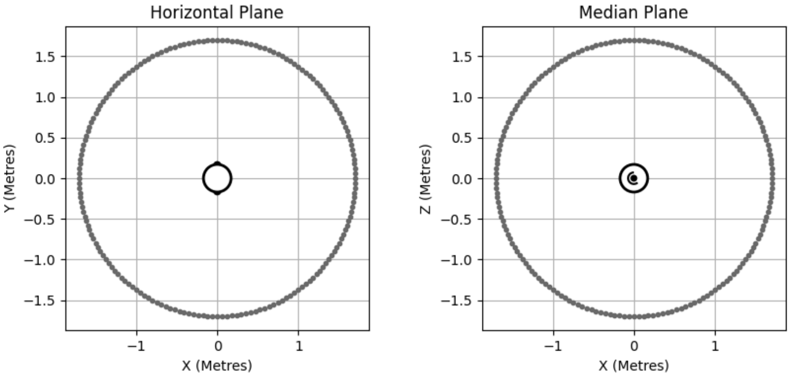

Figure 22: Horizontal and Median Plane Source Positions in TU-Berlin FABIAN HRTF Dataset

Club Fritz

Club Fritz is the name given to a round robin study on HRTF measurements in which a single KU100 head simulator was measured at various institutions with HRTF measurement apparatus (Katz and Begault, 2007; Andreopoulou *et al.*; 2015). This includes 12 measurement sets from 9 institutions:

- IRCAM, France
- University of Maryland, USA
- NASA Ames, USA
- IRCAM, France (#2)
- RWTH Aachen, Germany
- Helsinki University of Technology, Finland
- NHK Science & Technology Research Laboratories, Japan
- NICT, Japan
- Nagoya University, Japan
- Tohoku University, Japan
- IRCAM, France (#3)
- Austrian Academy of Sciences

The only standardised factors between these measurements are the head simulator itself, and that the measurements should be of freefield HRIRs. Otherwise each institution used only their typical measurement process, including the sampling positions.

Plots of the source positions on the horizontal plane for all twelve measurement sets are shown in Figure 23.

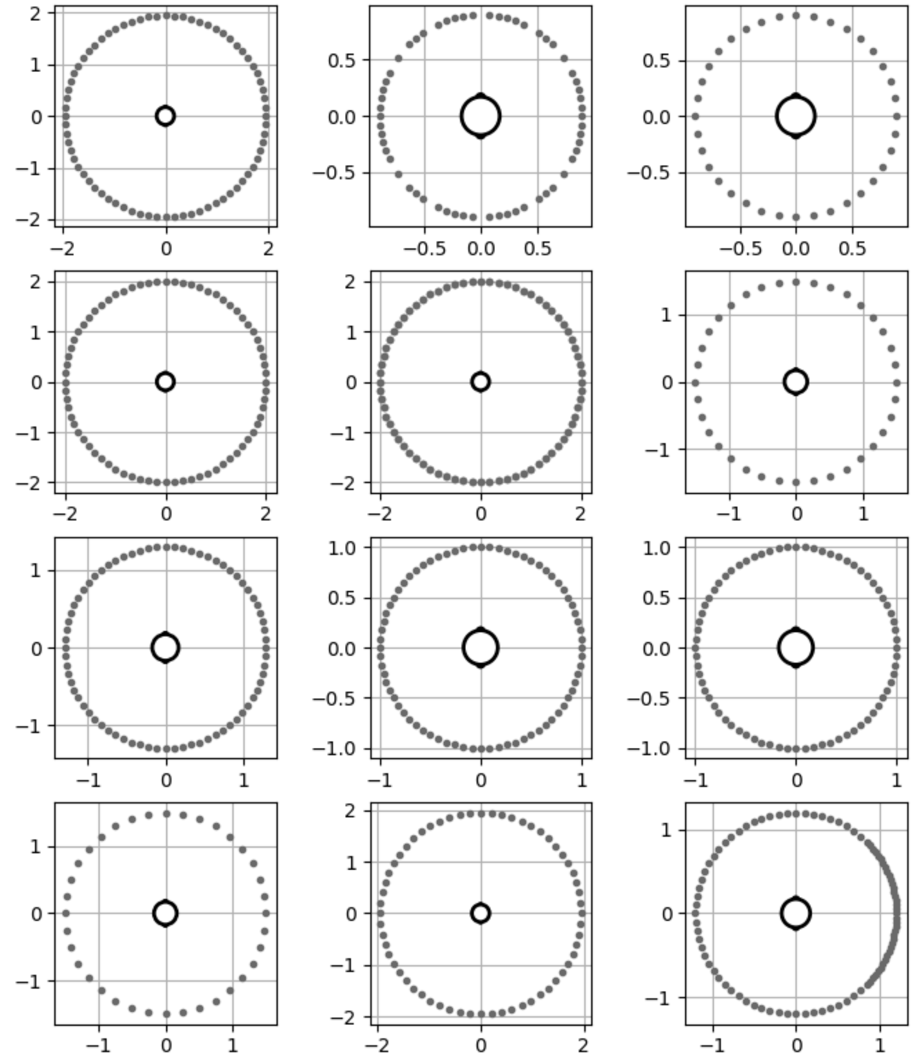

Figure 23: Source Positions on the horizontal plane for Club Fritz HRTFs

University of Iceland [Viking Dataset]

The Viking dataset contains full-sphere measurements of a KEMAR binaural array. However, a set of 20 custom pinnae models were attached the mannequin head for these measurements (Spagnol *et al.*, 2019). 19 of these pinnae were made from moulds of human subjects, and the last one was made from a mould of the KEMAR's large pinnae attachment. A spherical grid of measurements was made in a non-anechoic room, by rotating the mannequin head around its vertical axis to achieve different azimuths, and rotating the loudspeaker to achieve different elevations; the positions being shown in Figure 24. The distance from emitter to receiver was 1m.

The Viking dataset v2 furthers this by using the same moulds, but changing the material used to create the pinnae models, and taking measurements using the same measurement rig but in an anechoic chamber (Spagnol *et al.*, 2020).

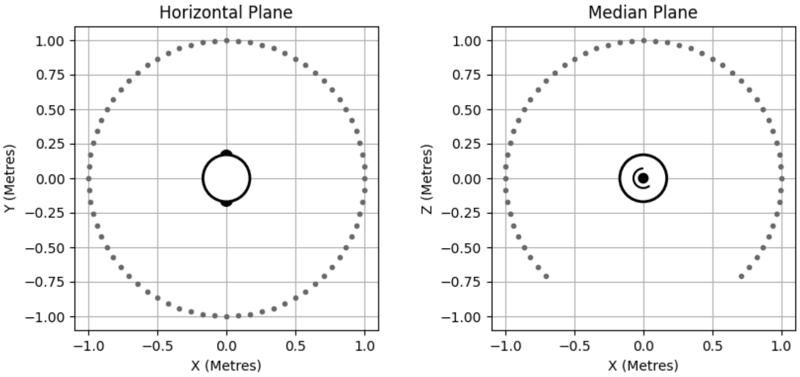

Figure 24: Horizontal and Median Plane Source Positions in Viking HRTF Dataset

Peking University, PRC [PKU-IOA Dataset]

The PKU-IOA dataset consists of freefield HRTFs measurements of a KEMAR mannequin (Qu *et al.*, 2008). These were measured with 1 degree spacing on the horizontal plane, achieved by rotating the KEMAR, while the elevation plane was sampled with 10° increments from -40° to 90°, and each of these spheres was sampled for 8 distances: 20, 30, 40, 50, 75, 100, 130 and 160cm, as shown in Figure 25.

This dataset is notable in that it foregoes the conventional loudspeaker-based HRTF measurement technique of using loudspeakers as a source, opting instead to use a spark gap to create an impulse (Qu *et al.*, 2009). This is ideal for measurements made in close proximity, as the source aperture of a spark gap is smaller than that of a loudspeaker, allowing the source to behave more closely to a point-source.

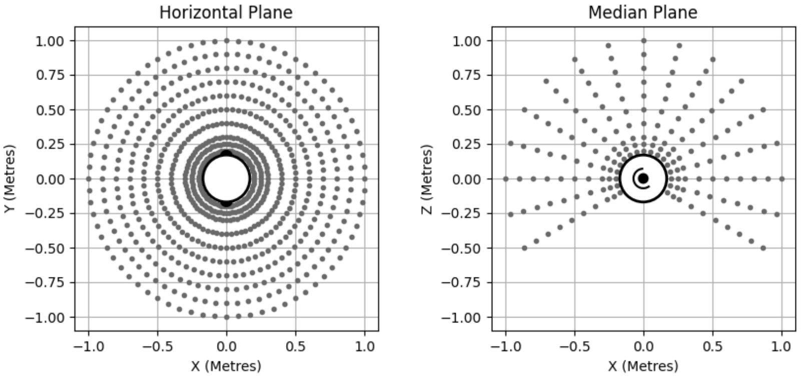

Figure 25: Horizontal and Median Plane Source Positions in Peking University KEMAR dataset

Aalto University, Finland

Marschall *et al.* (2023) measured a nearfield HRTF dataset, also using a spark gap, but for a 3D-printed mannequin head with microphones integrated at the mannequin's ears. HRTFs were measured for up to 13 azimuths, with this varying for elevation, and 6 elevations in the range -26° to 90°. This same spherical grid was measured for radii of 0.2, 0.3, 0.4, and 0.5m, as shown in Figure 26.

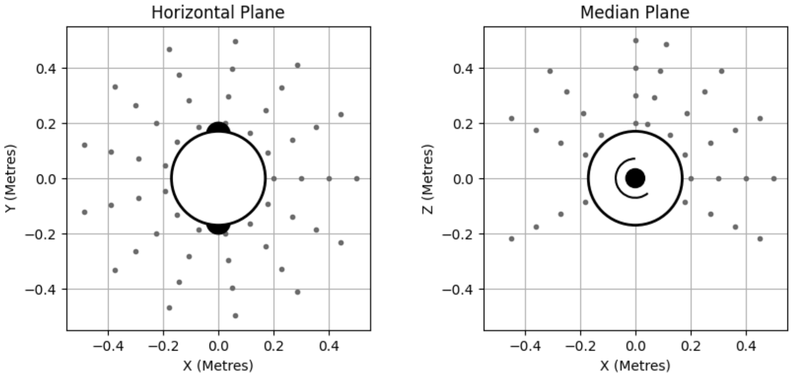

Figure 26: Horizontal and Median Plane Source Positions in Aalto University nearfielf HRTF dataset

Leibniz University Hannover, Germany

Li *et al.* (2023) also made multi-distance freefield measurements of a KEMAR, however theirs differed in that the spatial distance sampling is very dense. The spherical grid features spacing of 5° in both the azimuth and elevation dimensions achieved by yaw-rotating the mannequin on a turntable, and repositioning the loudspeaker for elevation.

This was then measured for different radii ranging from 0.2m to 1.1m, with a sampling density of 0.01m leading to a maximum of 90 distances measured. The minimum distance was not achievable for all elevations, leading to the spheres for distances under 0.3m being subsampled. The positions on the horizontal and median planes are shown in Figure 27.

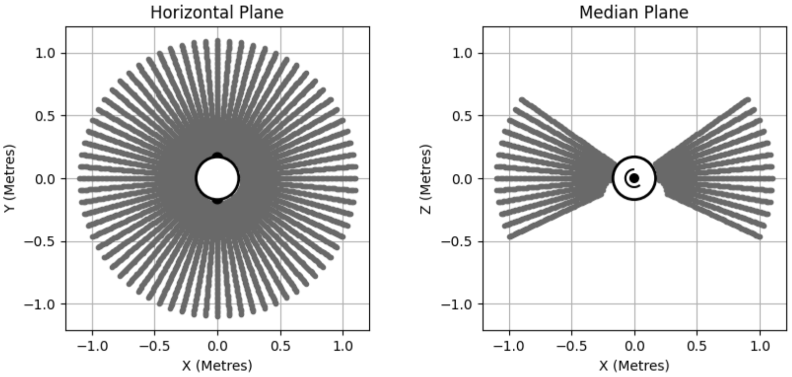

Figure 27: Horizontal and Median Plane Source Positions in Hannover Nearfield Kemar HRTF Dataset

University of Oldenburg, Germany Kayser *et al.* (2009) measured HRIRs of a B&K HATS in both anechoic and reverberant conditions. Not only were the HATS in-ear microphones used, but also three-channel microphone arrays on a pair of hearing aids for a total of eight channels. HRTFs were measured by rotating the HATS on a turntable and measuring with a single loudspeaker. The number of positions measured changed depending on the location; in the anechoic chamber the full horizontal plane was measured in 5° increments, for 4 different elevations in the range -10° to 20° and two different distances: 0.8m and 3m. Further measurements were made in two offices, a cafeteria and a courtyard; but the spatial sampling is sparse with a combined total of 69 BRIRs being measured between these four spaces.

These HRTFs and BRTFs have not been distributed in the SOFA format, and so are not plotted here.

Thiemann and Par (2019) made measurements of four head simulators, these being:

- KEMAR
- B&K 4128C HATS
- Head Acoustics HMSII HATS
- DADEC: A prototype HATS with adjustable ear canal (Hiipakka *et al.*, 2010)

The HATSs were measured in an anechoic chamber by rotating an arc upon which a loudspeaker sits, which could be driven along the arc circumference to change elevation. The spatial sampling was not entirely uniform, due to a reduction in sampling at high elevations, but mostly led to sampling of 2° in both the azimuthal and elevation dimensions, as shown in Figure 28.

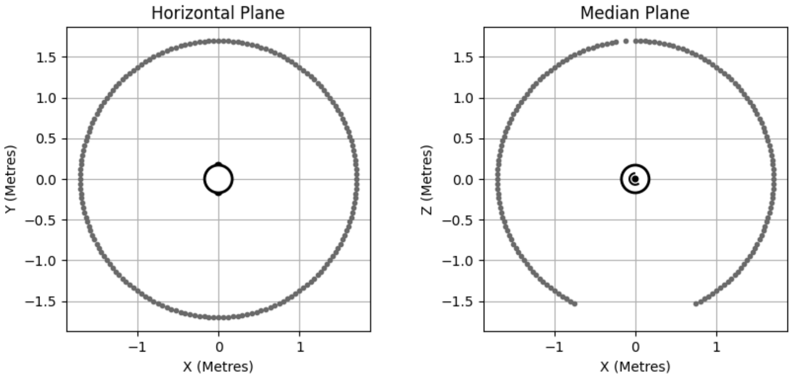

Figure 28: Horizontal and Median Plane Source Positions in Oldenburg HATS datasets

Another HRTF dataset was also measured at Oldenburg wherein a selection of in-ear hearing devices were placed in the ear of a human subject, and used to measure HRTFs from a spherical loudspeaker array (Denk *et al.*, 2018).

BRIRs

There also exists compilations of BRIRs measured in reverberant spaces. The objective of these is often for either the auralisation of signals, or in aid of research where testing under the diffuse field condition is required.

BBC R&D, UK

Pike and Romanov (2017) measured a collection of BRIRs designed specifically for loudspeaker auralisation, consisting of BRTF measurements of loudspeakers and a KU100 head simulator in a ITU-R BS.1116 (2015) compliant listening room. Each set of BRIRs was measured with a different loudspeaker layout, using the layouts defined in ITU-R BS.2051 (2022).

University of Huddersfield, UK

Bacila and Lee (2019) compiled a set of BRIRs of a KU100 head simulator inside a concert hall. A single loudspeaker remains in a constant position, while the head simulator was rotated on a turntable for 100 directions per revolution such that the horizontal plane is sampled by 3.6° increments. This is repeated for 13 different positions, 12 of which were in a grid equally spaced by 2 metres, with the last position being closer to the loudspeaker.

University of Surrey, UK

Multiple sets of BRIRs have been published by the University of Surrey.

The IoSR listening room multichannel BRIR dataset (Francombe, 2017) consists of BRIRs measured in another ITU-R BS.1116 compliant listening room. 24 loudspeakers, two of which were subwoofers, were arranged into a (9+10+3) configuration as defined in ITU-R BS.2051, and measurements were taken using a Cortex HATS which was rotated by one full revolution in 2.5° increments.

In aid of work on source separation the same HATS was used in another compilation, but of multiple rooms (Hummersone *et al.*, 2010). Notably, this included also an anechoic chamber, as well as a medium-sized office, a class room, a large lecture theatre and a small lecture theatre, all of which had an RT60 of less than 1 second. Measurements were made by placing a loudspeaker at positions on an arc so as to represent the frontal horizontal plane, such that φ was sampled in 5° increments for the range $-90° < \varphi < 90°$. Another set of BRIRs were measured using the same HATS, as well as a KU100, for other work on source separation (Remaggi *et al.*, 2019), measuring another different four rooms with RT60 values of less than 1 second. The two head simulators were used exclusively for different rooms.

The SurrRoom dataset (Cieciura *et al.*, 2023) consists of BRIRs, as well as RIR, of another 7 rooms. These rooms include the same listening room previously measured, a recording studio, and a selection of offices and classrooms. Rooms were measured using a KEMAR HATS, with the receiver being placed at five distances in front of the source, in the range 1m to 3m. The head of the HATS was then rotated to produce a selection of HATOs in 15° steps in the range -45° to 45°

TU Berlin, Germany

Erbes *et al.* (2015) measured BRIRs in a listening room, of unspecified acoustic quality. Measurements were made using a KEMAR HATS, measuring 64 loudspeakers in a square configuration, with a length and width of 4m. Measurements were made with the mannequin in the center of the array, as well as at an off-centre position.

Princeton University, USA

Qiao *et al.* (2024) present a set of BRIRs also measured in a listening room, however this dataset is differentiated from others as rather than the source loudspeakers being arranged for a standard surround reproduction, or in some spherical arrangement, the source is a line array of eight loudspeakers. A B&K HATS is translated in [x,y] positions in front of the line array to create a grid of 11x21 positions in the ranges [-0.5, 0.5]m and [0.5, 1]m relative to the line array. The HATS is also rotated to 37 azimuthal positions, moving in 5° increments from facing left relative to the array, to facing right.

Multiple Subject Compilations

A series of datasets exist which measure the HRTFs of multiple human subjects, and also often including head simulators alongside these. These datasets are typically used in aid of research on HRTF personalisation.

Austrian Academy of Sciences [ARI Dataset]

The ARI dataset consists of in-ear HRTFs measured in human listeners in a semi-anechoic chamber (2011). The listener was rotated by 2.5° increments within ±45°, and 5° outside of this range, while an arced loudspeaker array is used to measure elevations in the range -30° to 80°, as shown in Figure 29. These positions were measured for 260 listeners.

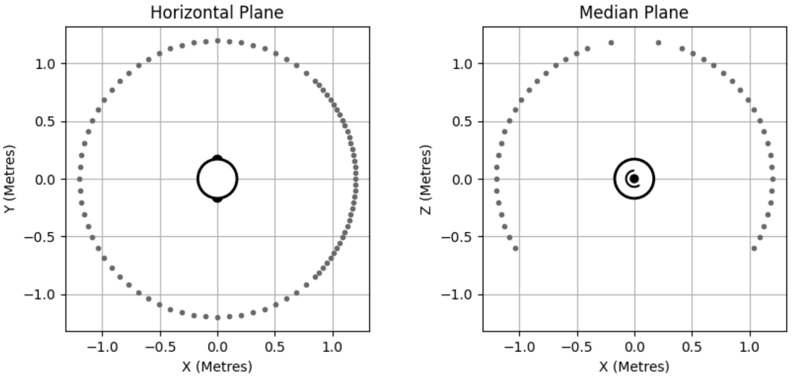

Figure 29: Horizontal and Median Plane Source Positions in ARI Dataset

University of California, Davis, USA [CIPIC Database]

The CIPIC database (Algazi *et al.*, 2001) is a very commonly used set of HRTFs. Measurements were made by rotating an arc of loudspeakers around the listeners interaural axis, rather than the more usual rotation around the vertical axis. This leads to uniformally sampled changes in elevation rather than azimuth, however as elevation was not sampled below 45°, the horizontal plane contains twice the number of positions as the vertical plane.

The loudspeakers were placed at 5° increments, however not the entire hemifield was sampled, leading to sparser spacing when approaching the interaural axis, as shown in Figure 30.

The public database consists of HRTFs for 43 human mannequins, as well as two measurements of a KEMAR wearing the large and small pinnae attachments.

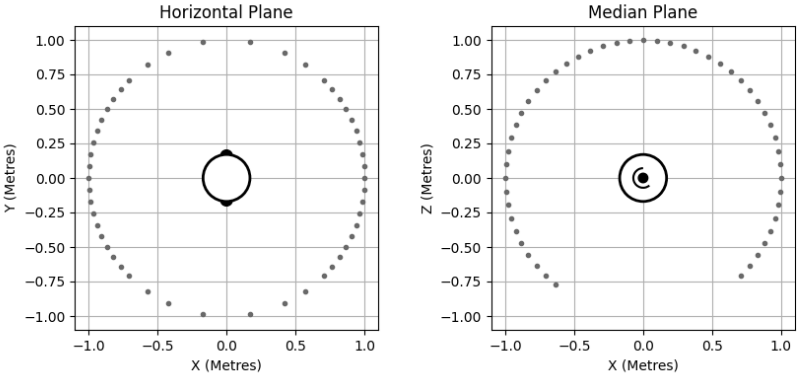

Figure 30: Horizontal and Median Plane Source Positions in CIPIC Dataset

Tohoku University, Japan [RIEC Dataset]

The RIEC dataset (Watanabe *et al.*, 2014) consists of HRTF measurements taken in an anechoic chamber by rotating a complete ring of loudspeakers around a listener. The ring had a radius of 1.5m, and with sampling points spaced by 10°, which was then rotated around the vertical axis to sample increments of 5°, as shown in Figure 31. HRTFs for 107 subjects are currently available in this dataset.

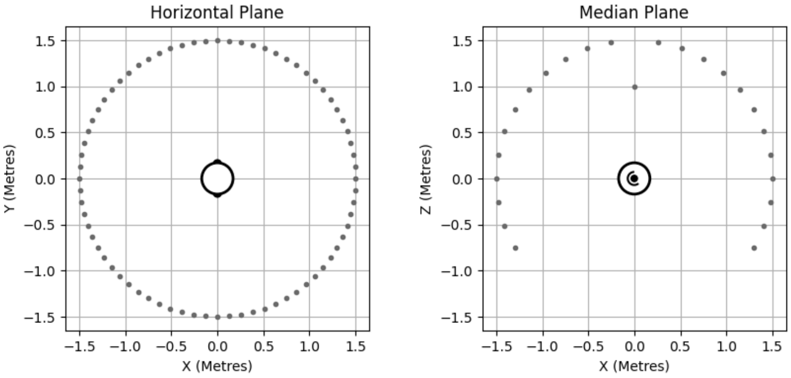

Figure 31: Horizontal and Median Plane Source Positions in RIEC Dataset

RWTCH Aachen University, Germany [ITA Database]

The ITA Database (Bomhardt *et al.*, 2017) consists of freefield HRTFs of 48 listeners. These were sampled by rotating an arc of loudspeakers around a listener in 5° increments, the arc containing 64 loudspeakers sampling the vertical plane with 2.5°

increments for elevations in the range -70° to 88°.

TU Berlin, Germany [HUTUBS]

Another dataset created at TU Berlin, HUTUBS, is a dataset of anechoic HRTFs and 3D head meshes of 93 human subjects (Brinkmann *et al.*, 2019), and two measurements of the FABIAN head simulator, with repeated measurements of one subject. Subjects wore in-ear microphones, and were positioned centrally in a vertically orientated ring of loudspeakers of radius of 1.47m. The ring allowed for spatial sampling of the entire median plane with 10° spacing, and was rotated such that the horizontal plane was also sampled with 10° spacing, as shown in Figure 32. Included also are numerically simulated HRTF based on the subjects' head meshes, which feature a much finer sampling grid.

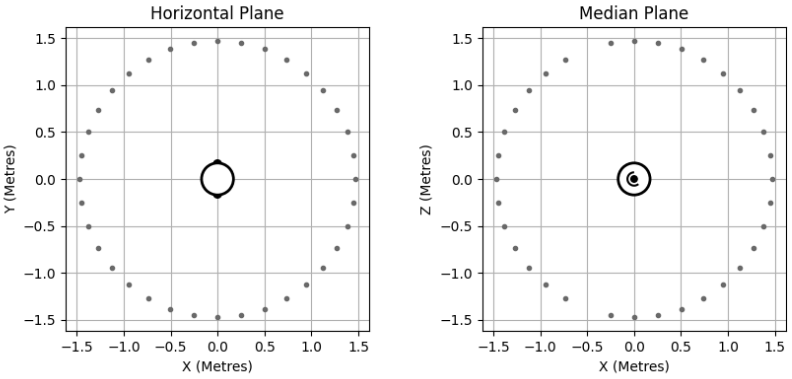

Figure 32: Horizontal and Median Plane Source Positions of measured HRTFs in HUTUBS Dataset

Princeton University, USA [3D3A]

At Princeton, a compilation of HRTFs of 38 subjects was made, together with head and torso meshes of 31 of the subjects. Subjects were measured by a vertically oriented arc of loudspeakers, affixed with 9 loudspeakers spaced near-uniformly in the range -15° to 75°. This arc was then rotated around the listener by increments of 5°, as shown in Figure 33.

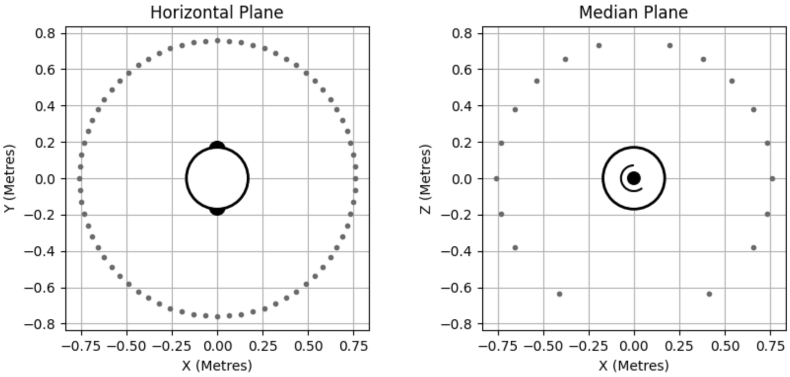

Figure 33: Horizontal and Median Plane Source Positions in Princeton 3D3A HRTF Dataset

IRCAM, France

Three HRTF measurement datasets have been published by IRCAM, each consisting of different measurement apparatus. The Listen database (Warusfel, 2003) consists of HRTF measurements of 51 subjects. Subjects had their HRTFs measured inside an anechoic chamber, using a single loudspeaker suspended by crane, while the listener was rotated on a turntable.

The crane was moved to allow for ten elevation positions equally spaced by 15° in the range -45° to 90°, while the turntable was turned in increments of 15° for most positions, though the very upper elevation positions had sparser sampling, as shown in Figure 34

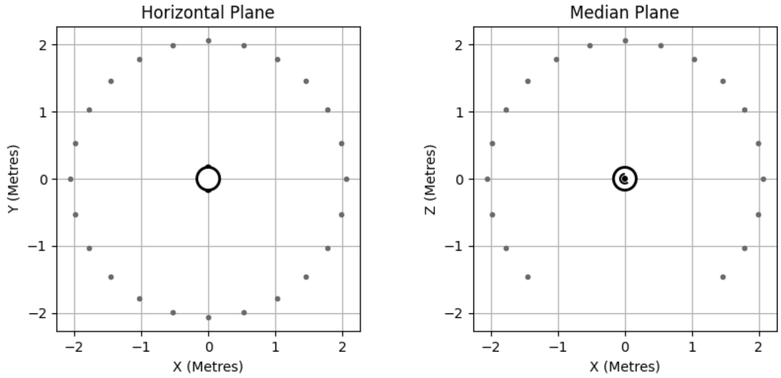

Figure 34: Horizontal and Median Plane Source Positions in IRCAM's Listen Dataset

The binaural listening (BiLi) dataset (Carpentier *et al.*, 2014) measured HRTFs of another 54 subjects, also being conducted in an anehcoic chamber. However, these measurements were performed with a denser sampling grid. Measurements were made by a pivoting arc of 4 loudspeakers.

The sampling is based on a Gaussian grid, with an elevation range with of -15° to 86°. This ordinarily would not include samples on the horizontal plane, but as these are required for many HRTFs applications, additional measurements were made at $\theta = 0°$. The extra positions measured can be clearly seen on the plotted median plane positions in Figure 35.

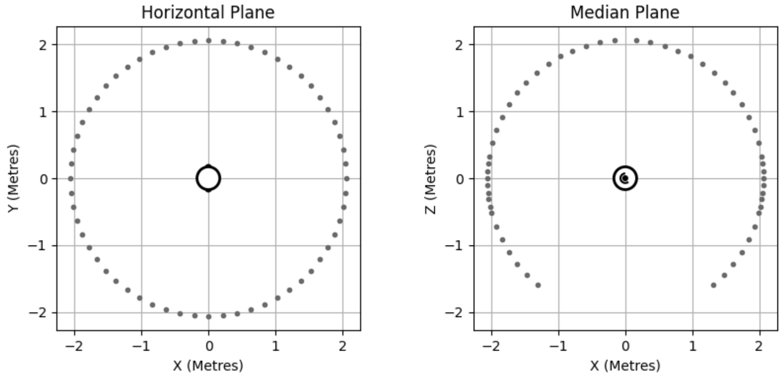

Figure 35: Horizontal and Median Plane Source Positions in IRCAM's BiLi dataset

Crossmod measures HRTFs of a further 24 listeners. Documentation for these is limited, but the description within the SOFA files confirms the dataset is also measured in the IRCAM anechoic chamber. The sampling grid employed is denser than Listen's, but sparser than BiLi's, and features a more heavily sampling of sources around the horizontal plane, as shown in Figure 36. It is not known if this also employs an arc of loudspeakers.

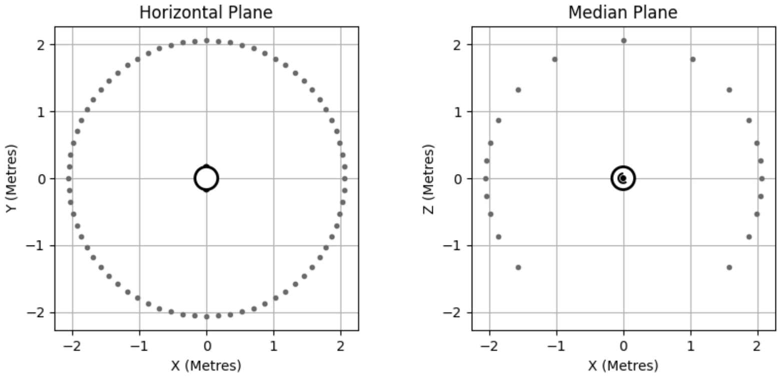

Figure 36: Horizontal and Median Plane Source Positions in IRCAM's Crossmod Dataset

University of York, UK [SADIE II]

The SADIE II database (Armstrong *et al.*, 2018) consists of anechoic HRTFs and diffuse-field BRTFs measured of 18 human subjects, as well as a KEMAR and a KU100 head simulator, as well as some anthropomorphic data. Different sampling grids were employed for the HRTFs of head simulators and human subjects, as well as for the BRTFs.

The SADIE II database samples the sphere as to achieve a distribution of multiple commonly used sampling grids used in ambisonic speaker arrangements (Armstrong *et al.*, 2017). This is achieved by rotating human listeners next to a static configuration of loudspeakers sampling a spherical segment. Horizontal and Median plane positions of this are seen in Figure 37.

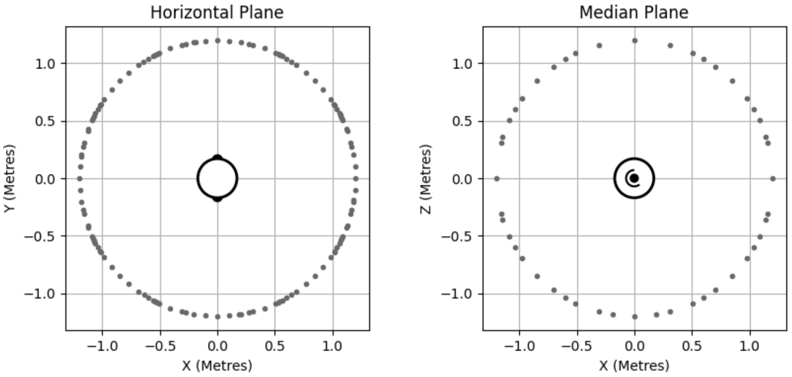

Figure 37: Horizontal and Median Plane Source Positions in SADIE Human HRTFs

The head simulators were measured using the same apparatus, however the rotation increments were shortened to 1°, resulting in a much denser sampling grid, as seen in Figure 38.

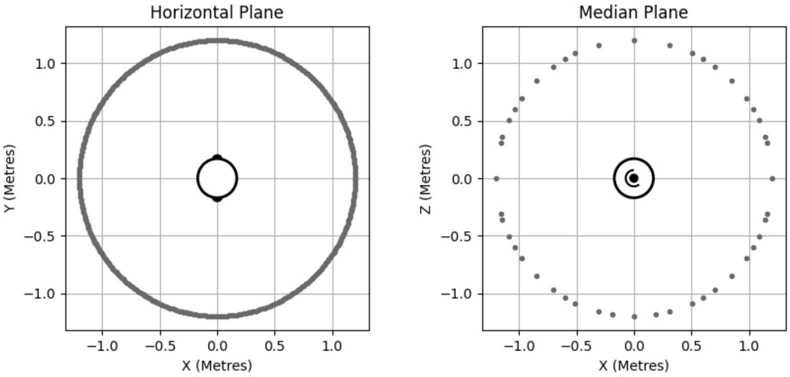

Figure 38: Horizontal and Median Plane Source Positions in SADIE Head Simulator HRTFs

The SADIE database is notable in also containing BRTF measurements of the same subjects. This was completed using a 50 speaker Lebedev grid inside a treated listening room. This looks particularly sparse in Figure 39 as not all azimuthal and elevation sampling occurs directly on the horizontal and median planes.

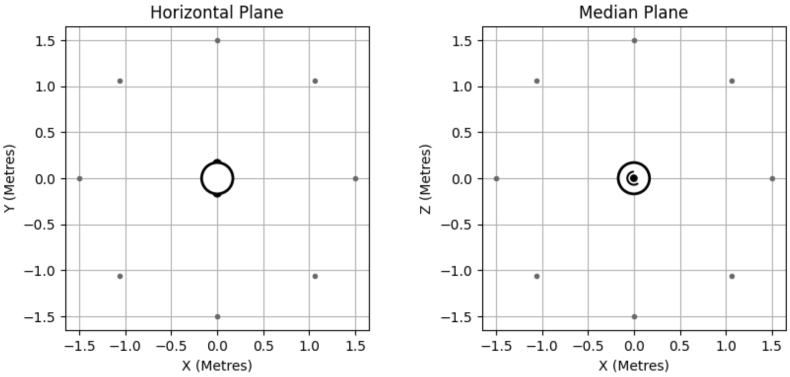

Figure 39: Horizontal and Median Plane Source Positions in SADIE database's BRTFs

South China University of Technology, PRC

Yu *et al.* (2018) compiled a nearfield set of HRTF measurements of 56 human subjects, evenly split between male and female. This was achieved by rotating the subject within an arc of loudspeakers. The loudspeakers on this arc were adjustable in distance, to achieve measurements at seven unequally spaced source distances between 0.2m and 1m, as seen in Figure 40

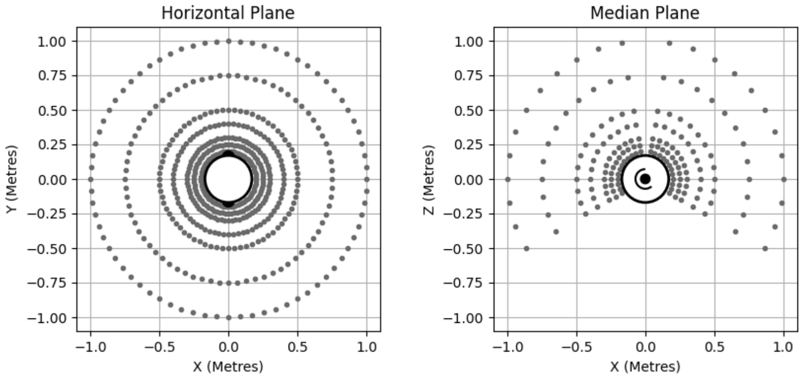

Figure 40: Horizontal and Median Plane Source Positions in SCUT Nearfield HRTFs of Human Subjects

An additional unreported HRTF measured with the same apparatus exists for the KEMAR head simulator. This features an additional set of positions on the transverse plane, as seen in Figure 41.

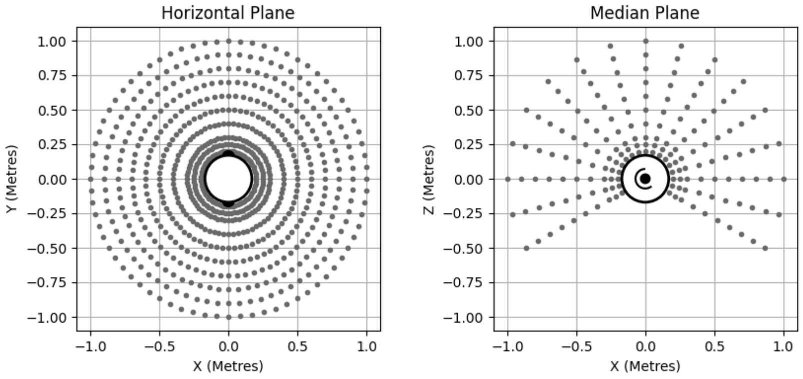

Figure 41: Horizontal and Median Plane Source Positions in SCUT Nearfield HRTFs of KEMAR head simulator

Imperial College London, UK [Sonicom]

The Sonicom HRTF dataset (Engel *et al.*, 2023) is currently the largest dataset of measured HRTFs, which at the time of writing contains HRTFs for 200 subjects, and an additional two sets of HRTFs of the KEMAR HATS with the two large and small pinna attachments. The dataset also includes 3D head meshes of all of the subjects. Horizontal and Median plane positions are seen in Figure 42.

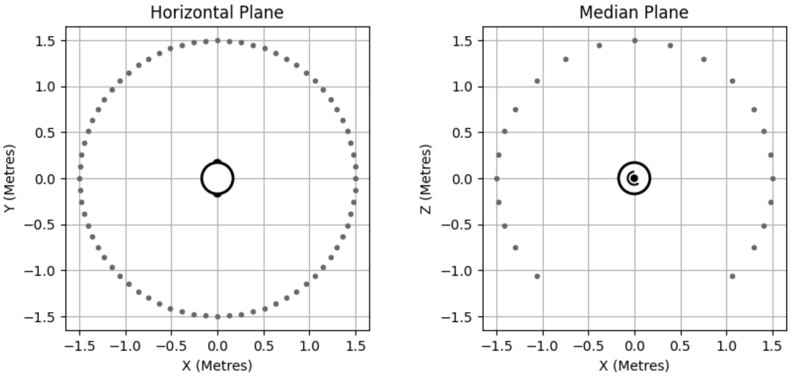

Figure 42: Horizontal and Median Plane Source Positions in Sonicom HRTFs Dataset

Meta Reality Labs, USA [Sound Sphere 2]

The Sound Sphere 2 HRTF dataset (Warnecke *et al.*, 2024) contains freefield HRTF measurements of 78 subjects. Measurements were taken in an anechoic chamber containing a 2m radius arc of 54 loudspeakers spaced by 3° elevation increments from a minimum elevation of $-69°$. The subject was rotated within this by 6° increments, resulting in the positions shown in Figure 43.

In addition to the human subjects, three head simulators were measured: a KEMAR, a B&K HATS, and a KU100.

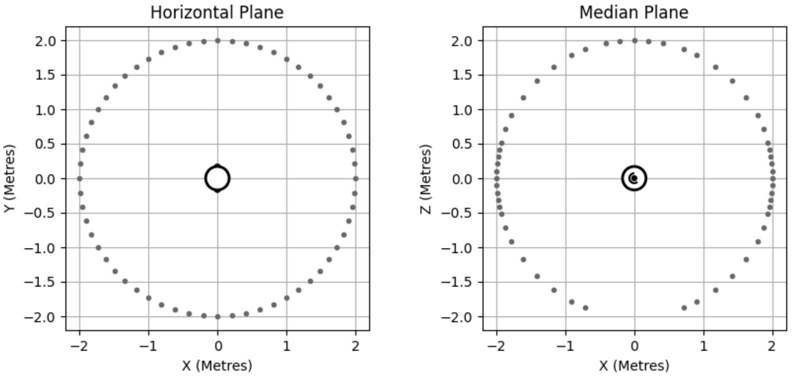

Figure 43: Horizontal and Median Plane Source Positions in Sound Sphere 2 Dataset

2.4.2 Review of Binaural Sound Source Localisation

Sound Source Localisation machinery is a well developed field, with examples of spaced microphone arrays being applied to acoustic location finding having existed for over a century (1918). Much of this work on this topic concerns sound source localisation by microphone array, but attention to replicating humans' binaural approach to sound localisation has also been long established.

This chapter will introduce and give an overview of significant work in the field of BSSL, with particular emphasis on machine learning based approaches.

Additionally, while comprehensive reviews of sound source localization are already commonplace (Rascon and Meza, 2017; Liaquat *et al.*, 2021; Desai and Mehendale, 2022), including specifically in relation to applying deep learning (Grumiaux *et al.*, 2022) to the task, selected works on SSL are still introduced here, selected for their particular relevance to the research later presented in this work.

Algorithmic and Model Based Approaches to BSSL

Macpherson (1991) introduced a model designed for estimating the azimuth of a binaural signal, for the purpose of analysing stereo speaker reproduction. A binaural signal was decomposed by filterbank into 16 bands, and for each of these bands an ITD and ILD estimate was made. ITD was estimated by finding the maximum of the interaural cross correlation function (IACCF), and ILD was estimated by measuring the ratio of energy in the two channels. Both of these cues, at every frequency band, were then mapped to an azimuth estimate. This was achieved by making measurements of a KEMAR HATS for different azimuths, and applying the model to these to find a ground truth. In the case where an ILD maps to multiple possible angles, an average of the possibilities is made. The many azimuth estimates were reduced to a single estimate by histogram analysis. The ground truths were made for 10° increments in the frontal horizontal plane, though interpolation was employed to allow for prediction with higher resolution than this. A precedent effect model was also applied, ignoring ITD measured over 1-6ms from what was identified as an initial onset. The system was tested first using the same dataset used to create the ground truth, where unsurprisingly high accuracy was reported, but then also in non-anechoic environments. The first reverberant environment was made by surrounding the binaural array by wooden panels at one metre away. This provided strong early reflections, due to the low absorption of the material. In this case, the system continued to perform strongly, only seeing significant error at large azimuths.

The next environment had all reflective surfaces, but further from the binaural array. This emphasises the later stages of the reverberation; the diffuse tail. In this case, much poorer performance was reported. The combination of these techniques suggests the strength of the precedence effect modelling.

Keyrouz *et al.* (2006) presented a model for binaural localisation in both azimuthal and elevation planes. For this technique, a database of inverse HRIRs at different positions was made. The paper describes several methods employed to reduce the size of the database, including reducing the size of the HRTFs through principal component analysis. Segments of binaural audio, as measured at the ears of a binaural array, were then convolved with all of the inverse transfer functions in the database. From the resulting signals, the correlation between the two channels was found; if the HRIR in the audio and the inverse filter are well matched then this correlation will tend towards one; so a location estimate is made based on the filter with the maximum correlation factor.

These systems were evaluated with binaural audio using the same HRTF set as used in the model: the MIT Kemar HRTFs (Gardner, Martin, *et al.*, 1994). Depending upon the degree to which HRTFs used in the inverse filters were reduced, and by which techniques, an accuracy of up to 90% was reported, with it also being seen that errors tended to occur in locations with small distance from the true positions. The disadvantage of this approach is the large number of convolutions involved; hence the need to reduce the size of the HRIRs.

An improvement to this approach was then proposed (Keyrouz and Diepold, 2006), in which complete cancellation of the original sound source is attempted by finding the ratio of the two channels in the frequency domain; if binaural audio is modelled as being the convolution product of only audio and a pair of HRIRs, this will leave the ratio of the HRTFs. A database of the ratios of HRTFs is similarly created, against which the comparison is made. An improvement in accuracy at smaller lengths was shown as compared to the first technique.

A further development to this approach was the reduction of the computation cost by reducing the search to specific regions of interest (Keyrouz, 2011).

This system was then later used as one block of a larger system (Keyrouz, 2014). This introduces a new process; microphones are placed inside both of the ear canals but also outside the pinnae on both sides of the head. The HRTF was then found as the ratio between the transfer functions of the inside and outside microphone. For both channels, this was correlated with a bank of HRTFs. A DoA estimate was either given by an average of the locations reported by the three systems, or by probabilistic model in the form of Bayesian fusion. The system was evaluated under the influence of both additive noise, and reverberation. It was shown that the combined system outperforms the individual systems, and that the addition of Bayesian fusion helps also.

Earlier Machine Learning based Approaches

May *et al.* (2011) describe training GMMs for the task of azimuthal DoA on the front horizontal plane, in the presence of reverb. A 20ms window was decomposed by gammatone filterbank (GFB) into 32 bands, from which ITD and ILD were extracted.

With the aim of achieving robustness, multi-conditional training (MCT) was applied: in this case specifically meaning the use of BRIRs in the binaural audio simulation to create reverberation, and the addition of interfering sound sources to the mixture.

The type of model was evaluated in a number of experiments. Firstly, altering the model complexity, which revealed the expected pattern of decreasing error for a larger model. Secondly, a study on which binaural cues are preferable was performed, which revealed highest performance for joint ILD and ITD with low pass filter. Thirdly, receiver position was altered. Through this it was found that multiple source distances must be considered for robustness to reverberation. Fourth, the GMM was compared to baseline systems for varying reverb time and number of sources. In all scenarios the GMM outperformed the more classical approaches. Fifth, the number of acoustic sources was estimated, by finding the number of peaks in the histogram of probabilities. This showed high performance for lower numbers of sources, particularly in less reverberant conditions.

May *et al.* (2015) later proposed another system based on GMM, but with the addition of a head rotation strategy. Recognising that a system tasked with full horizontal plane localisation when exposed to unknown rooms is likely to falsely output front-back reversals. Ambiguous azimuth estimations were identified if more peaks are found in the probability distributions than expected number of sound sources. When this occurs, the head is yawed by 30°. Comparison of the probability distributions prior and post rotation reveal the true peak, as the true source will counterrotate with respect to the yaw rotation, while the phantom source will rotate with the yaw rotation.

This strategy was simulated using TU Berlin freefield HRTF measurements of a KEMAR HATS (Wierstorf *et al.*, 2011), and the Surrey multi-room BRTFs measurements (Hummersone *et al.*, 2010). The system was evaluated on up to three individual speakers.

The results re-assert the need for MCT, with extremely poor generalisation to unseen rooms having been observed without it. The efficacy of the head rotation strategy was assessed on the basis of quadrant error rate; being the rate at which the system falsely localises with an error of larger than 90°. The head rotation strategy reveals a reduction of these errors by around 5%; which while significant, seems to be far less important than a well trained static model, as MCT reduces these same errors by 30%.

Further evaluation of this concept was reported by Ma *et al.* (2015c), where BRIR measurements of a KEMAR were made in rooms for this purpose. New rotations strategies are employed: rotating until facing the most likely source, this being justified by the observance of this behaviour in humans (Perrett and Noble, 1997), rotating towards the likely source but by a fixed maximum rotation, and rotating randomly. Additionally, this rotation was broken into multiple steps, rather than happening instantaneously. This was also evaluated for 1 to 3 speakers.

The results further support the use of a head rotation strategy. Of the proposed strategies, large (60°) yaw rotations towards the most likely target were preferred. It was shown that multiple step head rotations are preferred, as well as longer duration rotations.

Deep Learning based Binaural Sound Source Localisation

Ma *et al.* (2015b) further developed upon the work with GMMs and head rotations (May *et al.*, 2015; Ma *et al.*, 2015c) by the use of a feedforward DNN to perform DoA estimation as replacement of the GMMs. They proposed the use of a DNN trained upon frequency banded binaural cues; namely the IACCF and an ILD value. Frequency bands were made through GFB decomposition, and a separate DNN was trained at each frequency band. An azimuth estimate was then made by maximising the combined softmax outputs at all frequency bands.

This was then also tested with the same MCT approach described by May *et al.* (2011), and the head rotation strategy described by May *et al.* (2015). The system was compared to the previously described GMMs.

Across all testing conditions a limited but consistent improvement in performance is seen. The head rotation strategy also improves the results of the DNN based system compared with the GMM based system.

Performance of the same system was later reported using a different set of BRIRs measured for the full horizontal plane (Ma *et al.*, 2017). These results concur with what was already reported, of improved performance when using DNN, MCT when training, and employment of the head rotation strategy.

Lovedee-Turner and Murphy (2018) looked at application of machine learning to estimate the DoA of a sound source, but also the DoA of the source's reflections as measured in a BRIR.

In the experiment, anechoic HRIRs from the SADIE II dataset (Armstrong *et al.*, 2018) with added noise are converted directly into feature maps; first through GFB decomposition, and then the finding of frequency-banded IACCF and ILD. These were then used to train a cascade-correlation neural network; a form of adaptive neural network in which during training new units are added and trained to maximise correlation with the residual error of the network. This was reported to achieve better accuracy than a more typical MLP.

To evaluate whether this model could then be used for reflections, a dataset was made in an anechoic chamber where BRIR was recorded for head simulator, with a loudspeaker as emitter, and a reflective surface was mounted in the chamber to give a single reflection of known DoA; as calculated with ISM. This was done with both a KEMAR 45BC Head Simulator, and a KU100.

The direct path IR was separated from the reflection by windowing around identified peaks in the signal. 144 of these BRIRs were generated, representing source positions and reflections on the full horizontal plane. The results are compared to a baseline of estimating DoA based on ITD as found through IACCF.

The system reported a large improvement over the baseline method, particularly for front-back confusions which was reduced from baseline 50% to 0% and 1.39% for the direct sound, and 2.78% and 9.03% for the reflected sound: these two figures representing the two head simulators used for evaluation.

Vecchiotti *et al.* (2019) introduced an influential model, which estimates DoA directly from the raw waveform, rather than employing feature extraction techniques.

Two systems were introduced: one in which an audio signal was decomposed using GFB and fed into non-trained convolutional layers, and one where Binaural audio was created using the Surrey BRIR dataset (Hummersone *et al.*, 2010), convolved with speech signals from the TIMIT database (Garofolo *et al.*, 1992). Several training sets were created, an anechoic one, and training sets including all but one of the test rooms, which was repeated to allow for testing on each of the four rooms.

The results of the anechoic set showed significant overfit to the anechoic condition for WaveLoc-CONV, with around 40° of error for all rooms, and lesser overfit for WaveLoc-GTF with around 10° of error.

For the reverberant training sets, this is reduced to around 2° for both systems, WaveLoc-CONV outperforming WaveLoc-GTF on three of the four rooms. This suggests a tendency for CNNs to overfit on anechoic data, and for this to be significantly worsened if the input layer is unrestricted.

Xu *et al.* (2019) proposed a CNN based system for localising sound sources in the front hemifield of the horizontal plane in reverberant conditions.

The system extracts binaural cues using a cochlear model named the 'cascade of asymmetric resonators with fast-acting compression' cochlear system, originally proposed by Lyon (2017). This models frequency banding, nonlinear and feedback effects of the cochlear. The medial superior olive (MSO) is modelled by cross-correlation, and onsets are detected, at which point a correlogram is created from the previously described cochlear models, in a way that somewhat mimics the precedence effect.

These correlograms are used as the input of a relatively shallow CNN, with two convolutional layers.

A reverberant binaural dataset consisting of speech taken from the Austalk database (Estival *et al.*, 2014) was combined with BRIRs from an office. This same dataset was used to train and test a similar model, which applies extreme learning machine (ELM) to the correlograms rather than CNN. This system was used as a baseline.

The results show a noteworthy increase in performance from the CNN to the ELM, with localisation error being reduced from 19.1° to 3.7°.

Zhou *et al.* (2019) also applied CNNs to BSSL, training CNNs on a feature matrix made from the IACCF of GFB derived frequency bands. The binaural audio for this was created by filtering of speech taken from the CHAINS speech corpus (Cummins *et al.*, 2006) with BRIRs of two sources: measured BRIRs from the Surrey dataset (Hummersone *et al.* which were used exclusively for testing, 2010) as well as ISM synthesised BRIRs using HRIRs taken from the MIT KEMAR dataset (Gardner, Martin, *et al.*, 1994) which were used for both training and testing. The binaural data consisted of source positions on the frontal hemifield of the azimuthal plane, as per the Surrey BRIRs. Additionally white noise is added to the testing dataset, at various levels.

Particularly high accuracy was reported when the system was tested on the synthetic BRIRs, with root mean square localisation error (RMSLE) ranging from 1°-2° depending on reverb time and signal-to-noise ratio (SNR). More notably, however, this performance decreases only to 3° when trained and tested on unknown rooms.

Wang *et al.* (2019) and then Wang *et al.* (2020) addressed the task of localising in the mismatched HRTF condition; that is when the HRTFs used in the training dataset differ to those used in the evaluation dataset. To address this, training data was clustered into groups of a more similar HRTF.

Binaural audio was created by combining speech taken from the TIMIT corpus with HRTFs taken from the CIPIC and RIEC databases. Only positions on the frontal horizontal plane were considered. From these, IACCF and ILD were extracted and used as cues for localisation.

Clustering was achieved by means of affinity propagation; a clustering method which differs from the more common k-means clustering by not requiring the number of clusters to be specified in advance and by using message passing between data points to identify exemplars and form clusters. For the clustering, a similarity matrix was created based on the localisation accuracy of the DNN when trained and evaluated on every combination of HRTF. From each cluster, a single HRTF was selected, and finally these selected HRTFs were used to train the DNN, with the rationale of reducing complexity while still allowing for full generalisation.

As a baseline, the system was compared against conventional cross-correlation based localisation, the model based methods proposed by Raspaud *et al.* (2009) and Pang *et al.* (2017), and the similar DNN proposed by Ma *et al.* (2017). An improvement of localisation accuracy was reported over the baselines.

To address the issue of front-back confusion in BSSL systems, Jiang *et al.* (2020) proposed an alternative architecture fusing a CNN responsible for distinguishing front-back hemifield, and a DNN responsible for giving an azimuth estimate which is transformed into the correct hemifield based on the result of the CNN.

The model was trained on a truncated IACCF curve, and an ILD estimate, of binaural audio. The binaural audio was created through convolution of speech samples with BRIRs taken from the AIR dataset (Jeub *et al.*, 2009), and additive noise was introduced through noise samples taken from the NOISEX-92 dataset (Varga and Steeneken, 1993). High localisation accuracy was reported in anechoic and noisy conditions, but performance suffered in reverberant conditions. Proof is shown, however, of the relative success of the CNN in resolving front-back confusion, with this being improved over baseline systems.

Zhao *et al.* (2021) proposed a DNN based system, trained on typical binaural cues but evaluated in conditions of low SNR. Binaural audio is decomposed by GFB, and then at each frequency band converted into interaural cues: ITD, ILD, interaural phase difference (IPD) and interaural coherence (IC). ITD and IC were used by a deep belief network (DBN) to localise the correct quadrant, as to resolve front-back confusion, while all four cues were input to a DNN tasked with azimuthal estimation within that quadrant. Training and testing was split into two experiments, one looking at SNR and one at reverberation. For the first test, binaural audio is created by combination of speech samples with the MIT KEMAR database (Gardner, Martin, *et al.*, 1994). Different noise samples were added to the signals, at SNR ranging from 20dB(SNR) to -10dB(SNR). Models from three other works were used as a baseline (Wu *et al.*, 2016; Zhang and Liu, 2015; Ma *et al.*, 2017). A slightly higher level of localisation error is reported over the baseline systems.

The model was then also tested in reverberant conditions, using BRIRs measured at Surrey (Hummersone *et al.*, 2010) and TU Berlin (Ma *et al.*, 2015a). In this case too, a higher robustness to reverberation is reported over the baseline systems.

The methods up to now use neural networks in the DoA estimation stage of binaural sound localisation, either predicting from features or the raw binaural audio. Another possible approach, however, is to leverage neural networks in turning binaural audio into more useful features for a more conventional sound localiser on which to perform localisation.

Yang *et al.* (2021a) presented such an approach, wherein the direct path relative-transfer function is predicted from binaural audio. This is a common feature used in array based SSL; in the context of BSSL it refers to the relative transfer function of the two HRIRs. Reverberation and noise would reduce the efficacy of such an approach, so a first stage of monaural speech enhancement takes log-spectrograms of raw speech, and applies bidirectional long short-term memory (BiLSTM) to directly predict a clean log-spectrogram.

A binaural enhancement stage separately processes intensity and phase matrices with convolutional layer, before concatenation and use of LSTM for the transfer function prediction.

A DoA estimate was given by matching the predicted transfer function to a ground truth.

A testing dataset was created by creating BRIRs by ISM simulation using HRTFs from the MIT Kemar datset. This was done for the frontal horizontal plane. These BRIRs are then combined with speech from the TIMIT corpus. In addition, noise from the NOISEX-92 Database (Varga and Steeneken, 1993) are added to the signals.

The system was compared against two neural network based models of SSL (Pak and Shin, 2019; Chakrabarty and Habets, 2019). An increase in performance was reported over the baseline methods.

More extensive testing of the same approach was later reported (Yang *et al.*, 2021b); with some changes to the model such as the use of more convolutional layers in the binaural enhancement stage, and the replacement of LSTM with GRU. A notable change in the experimental method was the introduction of an HRTF mismatch; BRIRs for training and testing are now created using HRTFs from the CIPIC database (Algazi *et al.*, 2001). The results in the mismatched condition were strong, with little reduction in accuracy being seen.

Further testing was also undertaken using real binaural recordings, taken from the LOCATA dataset (Tang *et al.*, 2019); for which the proposed model also outperformed the baseline methods.

Massicotte *et al.* (2022) introduced an LSTM based system. The system performed feature extraction on binaural audio by way of wavelet scattering, a form of signal decomposition by wavelet transform (Andén and Mallat, 2014). This was used to train and test two separate LSTM layers, and subsequent regression created estimations for azimuth and elevation.

The system was trained on speech data from the TIMIT database combined with HRTF of the KEMAR mannequin taken from the CIPIC database.

They reported high levels of accuracy, however the system was only tested in the context of SNR, and the influence of reverberation was not considered.

O'Dwyer and Boland (2022) trained a DNN for BSSL when using multiple HRTFs, but using a novel technique of sorting data in order to achieve better performance.

Binaural audio was created by combining speech samples from the McGill TSP database (Kabal, 2002) with HRTFs from the CIPIC database (Algazi *et al.*, 2001). The features extracted from this were ILD, the IACCF, and gammatone-frequency cepstral coefficients (GFCCs); which were used to train a DNN.

O'Dwyer and Boland (2022) assessed the performance of DNNs trained with one HRTF when tested using other HRTFs from the database. The poor performance typically associated with the mismatched HRTF condition was seen in this case.

Additionally, it was observed that performance was not equal for all mismatched HRTFs. Based upon the fact that performance seems to be higher for similar HRTFs, O'Dwyer and Boland (2022) proposed clustering into groups of similar HRTFs, and training and testing within these clusters. Clustering was enabled by means of the affinity propagation algorithm (AP) algorithm. The result of this was a very small decrease in performance for known HRTFs, but a significant increase in performance for unknown HRTFs.

Inspired by the efficacy of CNNs, but wishing to find a less computationally complex solution, Phokhinanan *et al.* (2023) proposed the use of vision transformer. The vision transformer (Dosovitskiy *et al.*, 2020) breaks an image into smaller tensors, named patches, with embedded information about original position in the image. Each of the patches passes through a self-attention mechanism, and feedforward neural network. In the proposed model, binaural audio is turned into ILD and IPD matrices by taking STFT of the binaural audio, and finding the ratio of the log-magnitude and phase of the two channels. Rather than embedding patches as squares, vertical lines of value are used so as to represent frequency bins.

A training dataset was made by using HRTFs from the MIT KEMAR database, (Gardner, Martin, *et al.*, 1994) for source positions in the frontal azimuthal plane. These were combined with utterances taken from the TIMIT spech corpus (Garofolo *et al.*, 1992). Noise was then added to the signals, using noise signals taken from the NOISEX-92 dataset (Varga and Steeneken, 1993).

The system was evaluated in two scenarios: for a testing dataset with varying levels of unseen noise, and a dataset with both varying levels of unseen noise and unknown speakers. Under these conditions, it outperformed a baseline of a CNN trained for the same task.

Following from this, Phokhinanan *et al.* (2024) proposed another similar system, but for the task of BSSL in the mismatched HRTF condition. The newly proposed system uses addition feature representations: matrices of the real and imaginary parts of the two signals, and a new architecture wherein each of the four features has a unique encoder block. The training and testing datasets were similarly made using speech from TIMIT and noise from NOISEX-92, but HRTFs from CIPIC (Algazi *et al.*, 2001) were used for training, while HRTFs from the RIEC dataset (Watanabe *et al.*, 2014) were used for testing.

The system was tested under three conditions; common HRTFs but mismatched noise-types, mismatched HRTFs but common noise types, and both mismatched HRTFs and noise. The results again showed an improvement over the baseline. A significant reduction in accuracy was seen in the mismatched HRTF condition for both the proposed and baseline systems.

Geva *et al.* (2024) also proposed a novel hybrid model, training a network on both time-domain waveform data and TF-domain spectrograms.

They created a dataset using HRIRs measured for 24 source directions, and convolved these with music taken from the MUSDB18 dataset; a dataset of mixed music and individual channel stems intended for music source separation tasks (Rafii *et al.*, 2017).

A two branch CNN was used, in which one branch 11,025 sample long sections of time-domain binaural audio was input into four 1D convolutional layers, and in the other spectrograms of size [129, 127, 2] were input into four 2D convolutional layers. These were then concatenated, and the output layer was a three element vector predicting Cartesian coordinates of the loudspeakers.

They used the Vecchiotti *et al.* (2019) system as a baseline. Geva *et al.* (2024) reported high levels of accuracy, with an average angular error of 0.24°. This compares to the reported 19.07° of error found from retraining and testing the baseline system on this work's dataset.

One potential issue in speaker estimation, is that speech tends not to be continuous, and in these moments of silence a DoA estimator is more likely to give an erroneous estimation.

To address this, Varzandeh *et al.* (2024) proposed the integration of a voice activity detector (VAD), detecting when speech is occurring. In the study, three baseline systems were use; these being CNNs trained upon the generalised cross correlation phase transform (GCC-PHAT) of the binaural signal, the magnitude and phase of the CPS of the binaural signal, and the real and imaginary parts of the CPS of the binaural signal. These baseline systems are augmented by the probability that speech is occurring as determined by VAD, such that no DoA estimate is made while speech is not likely to be occurring.

These baseline systems were compared against the proposed systems, which combine the previously described feature representations with another input; a periodicity cue named the periodicity degree, which was originally proposed for other application (Chen and Hohmann, 2015), wherein the signal is decomposed by a filterbank of comb filters with different delays so as to identify different fundamental periods in the signal, and the feature representation extracted is the mean amplitude of the filtered signals.

The systems were trained on binaural audio made by convolution of speech signals taken from the TIMIT corpus with HRIRs taken from the Oldenburg HATS HRTFs (Kayser *et al.*, 2009). Noise was added to this to make SNR in the range of -5dB to 20dB. Testing was done under two conditions: static and moving sound sources. In both cases, HRIRs are replaced with BRIRs measured in non-anechoic conditions.

The proposed systems outperformed the baselines. Of the tested feature representations, GCC-PHAT performed most strongly in all conditions.

Distance Estimation in Binaural Sound Source Localisation

Yiwere and Rhee (2017) tested a model designed for joint distance estimation and azimuthal direction estimation in reverberant conditions; however the range of azimuths was limited to just the three directions 0°, 30° and 60°, and four distances 1m, 1.5m, 2m and 3m. The system consisted of a DNN trained on the IACCF of the binaural signal.

Notably, for training and testing data was captured with in-situ recording. Sound samples, which are stated to include speech from the TIMIT corpus (Garofolo, 1993) as well as unreported samples, are played back through a loudspeaker at the previously described positions, and stereo recordings are taken from a spaced microphone pair. This, notably, is not a binaural array; however this may not be of significant consequence as ILD and monaural cues are not considered by this model. This method of creating audio data accounts for the small amount of positions measured in the dataset.

This was carried out for three rooms. The model was trained independently on room 1, room 2, and rooms 2 & 3 combined. These combinations were then tested separately on room 1, 2 and 3. The reported results mostly show the predicted pattern, of being higher when the training dataset contains audio from the same room as in the testing dataset; except for the model trained on rooms 2 & 3 still reported higher accuracy on room 1, suggesting favourable acoustic conditions.

Given the small number of source locations, it is difficult to draw conclusion from the classification accuracy of this system as even high accuracy represents less well performing localising than other previous models. However, this does suggest that IACCF may be a viable cue for distance estimation as well as azimuth estimation.

O'Dwyer *et al.* (2019) investigated the use of typical ML approaches to BSSL for the purpose of estimating distance.

Binaural audio was created through the measurement of BRIRs of a KEMAR HATS for two different azimuths; 0° and 30° and three 1m spaced distances from 1m to 4m. This notably is outside of the near-field, so the system will not be able to interpret distance from changes in binaural cues.

A variety of features were extracted: ITD, as being determined by the peak of the IACCF, broadband ILD, IACCF, average energy of GFB decomposed signals, and mel-frequency cepstral coefficients (MFCCs). These cues were used to train DNNs both separately, and in combination.

Unsurprisingly it was found that combination of all features performed best. Individually, the IACCF and the MFCCs performed best when $\varphi = 0°$ but gammatone energies performed better when $\varphi = 30°$.

El-Mohandes *et al.* (2023) investigated deep BSSL on ‘earable’ devices; those being electronic devices worn at the ears such as earbuds and headphones. In this case, the HRTF is dependent upon the listener’s head. To avoid HRTF mismatch, they proposed a lightweight HRTF measurement procedure in which a mobile phone is used as an emitter. Binaural audio used for training was then created based on these HRTF measurements representing a full sphere of source positions.

The binaural audio was used to create gammatone-frequency cepstrums (GFCs), and an IACCF. The GFC was used as input to a branch of convolutional layers, the IACCF was interpreted by a branch of dense layers. The branches were concatenated by further dense layers. The output is a prediction of both azimuth and elevation. Loss is calculated as the great-circle distance between the predicted and true spherical locations.

Two training schemes were proposed; subject-dependent where all weights are trained from scratch from the users’ unique HRTFs, and subject-adapted where the network was first trained on a generalised dataset, the weights of the layers in the individual branches were frozen and only the weights of the post-fusion layers are updated for the data from individualised HRTF. These were compared to a baseline of subject-independent data; the generalised training dataset.

The two proposed training schemes show a significant improvement over the baseline of a generalised dataset, showing the importance of individualised cues for BSSL. This performance is likely skewed heavily by cone-of-confusion errors, which would not have been present in the most popular work on BSSL with HRTF mismatch (Wang *et al.*, 2020) as they restrict to only the front horizontal plane.

The proposed subject-adaptive training scheme did not perform well when training the model on individualised HRTFs.

It is not only head rotation which can provide potential cues useful for BSSL; listener position translations also will have their own dynamic effects. Krause *et al.* (2024a) studied their potential use in a full position BSSL system estimating distance as well as DoA.

The authors trained a two branch convolutional recurrent neural network (CRNN), with branches representing two forms of audio representation: spectral features, and motion-based features. The spectral features consisted of mean magnitude spectrogram, IPD, and ILD. The motion-based features consisted of the listener's head rotation, represented with quaternions, as well as the listener's position, represented with Cartesian coordinates. The branches consist of 2D convolutional layers, and are concatenated before use of bidirectional gated recurent unit (BiGRU) layer. The system as first tested on separate datasets of where the listener is static, with rotating head, and with listener translation, all of which is trying to localise a single speaker. A large number, 2500, of synthesised room configurations were used in the study for reverb.

In a test only performing DoA estimation, localisation error was reduced by the inclusion of head rotation, and substantially reduced by the inclusion of listener translation and head rotation.

Works on Sound Source Localisation for Hearing Aids

As previously established, Hearing Aids greatly benefit from the ability to localise sound sources, since this allows for selective beamforming to try to replicate the cocktail party processing effect that is damaged in hearing impaired listeners (Noble and Gatehouse, 2006). These systems may not be true BSSL systems, since it is not uncommon for microphone arrays to be appended to each hearing aid to assist the beamforming part of the processing. However, in the context of the human head, these sensors are still near coincident, and so analysis of binaural cues for localisation is still similar to conventional BSSL.

Courtois *et al.* (2014) proposed a system specific to hearing aids, supposing that at the time common statistical approaches would not be possible on the devices owing to memory constraints. Instead of this, they recommend exploiting the radio transmission of a worn-microphone's audio to the receiver hearing aids, and using the difference in transmission time in order to calculate ITD.

Goli and Par (2023) proposed a CNN trained on TF representation of sound at each microphone in hearing aid based array. The matrix is created by GFB decomposition, followed by half-wave rectification and square rooting of the signal in a method meant to induce nonlinearity in an emulation of human audition. Inspired by its use in other fields (Sainath *et al.*, 2015; Park and Yoo 2020), Goli and Par (2023) tested whether learnable parameters in the GFB provide better localisation accuracy; an approach previously seen in Vecchiotti *et al.* (2019).

It was found that the learnable GFB does improve localisation, but only if the parameters are initialised with parameters based on typical GFB.

The system was tested in three different rooms, and at differing levels of SNR. It was seen that the system is able to generalise to other rooms, with the expected behaviour of reducing accuracy owing to increasing reverb time and decreasing SNR. The proposed system is able to outperform a baseline system of a MLP trained on GCC-PHAT features: though this would not be expected to work well for near-coincident channels, as in this case of monaural array.

The authors also compared results of using all microphones on both ears, deemed a binaural array, and only microphones on one ear, deemed a monaural array. The system is still able to perform monaurally, but loses robustness to reverberation and noise.

It is notable, however, that the proposed system uses a relatively large TF-representation, with a size of $[2205, 32, M]$ where M is the number of microphones. Computationally, this may not be an efficient approach if such large matrices are required.

2.4.3 State of the Field in Binaural Sound Source Localisation

From the works listed, some general trends can be seen, as outlined below.

Bias towards Speech Localisation

There is a very strong emphasis on DoA estimation of speech as a sound source. There are several reasons why this may be the case: speech is likely the most impactful application of sound localisation, as a source of audio there is an abundance of large scale speech datasets, and speech as a signal is complex, and tasks which work well for speech do often generalise well to other forms of audio.

Pivot to CNN and CRNN

Original application of DNN regularly saw use of handcrafted features. The trend now is towards end-to-end systems, with convolutional layers generally replacing this feature extraction process. Within this is another trend, increasing usage of recurrent layers with the architecture of a CNN.

Bias towards Azimuth Estimation

A large bias towards azimuth estimation over other forms of localisation is seen. A reality of this is that BSSL will always be more accurate when doing so due to the importance of binaural cues only valid for changes on the horizontal plane.

Sound Localisation as a Classification Task

Coordinates are continuous values, however in spite of this BSSL is often treated as a classification task.

Single Source Localisation

Some noteworthy study on multi-speaker localisations have been seen, but since the field's pivot towards CNN, total emphasis has been placed on providing DoA estimates for single listeners.

3 Research Aims and Design

3.1 Aim & Objectives

The primary aim of this work is to assess the utility of deep learning for the task of binaural sound source localisation.
To enable this, the following objectives were set out:

- Survey and assess existing literature on binaural sound source localisation, identifying gaps existing in current literature
- Design a framework for experimentally evaluating deep neural network performance on the task of binaural direction of arrival (DoA)-estimation
- Benchmark existing techniques of preparing binaural datasets for binaural sound source localisation tasks in different conditions
- Propose and evaluate new techniques for preparing binaural data for unexplored acoustic conditions
- Investigate convolutional neural network architectures for the task of binaural sound source localisation
- Summarise findings and propose areas for future work

3.2 Methodology

This thesis investigates the task of binaural DoA estimation using deep neural networks by conducting experiments on the preparation of simulated binaural datasets, the conversion of binaural datasets into feature representations, the choice of deep neural network architecture, and dataset pre-processing in pursuit of better generalisation.

The method is separated into four distinct phases, corresponding to the four following chapters.

Localising in Simulated Acoustic Environments: To uncover areas of particular challenge with regard to simulated acoustic data, an approach is taken in which a deep neural network trained on anechoic binaural data is tested with various different types of alterations. From these results particularly challenging acoustic environments are uncovered.

Feature Representations: Comparative experiments are conducted to assess different feature representations. Firstly, magnitude-based features are compared under controlled conditions, with consistent datasets, training regimes, and model structures. This process is then extended to a broader range of phase and time-based cues, also using controlled comparisons to isolate the effects of the feature choice.

Deep Learning Architectures: Network architectures are evaluated through controlled comparative analysis in which datasets, training process, and model size are fixed while the architectural design is varied. This phase investigates the relative merits of different convolutional and recurrent designs for binaural localisation.

Mismatched Anechoic Condition: This phase presents an investigation into a previously underdeveloped part of binaural sound source localisation (BSSL). This is achieved by first demonstrating experimental proof of the existence of this issue, before proposing possible solutions to this issue and confirming the efficacy of the proposed solutions through comparison with a baseline.

4 Localising in Simulated Acoustic Environments

4.1 BSSL with Convolutional Neural Networks

In Binaural Sound Source Localisation, high accuracy can be achieved under ideal conditions: that being a sound scene in which there is no reverberation, additive noise, or interfering sources. This ideal scenario, however, is an impossibility in the real world; any useful system also needs to perform with these degradations present.

This chapter will introduce the different acoustic features that can affect performance, and show the effect varying them has upon classification accuracy. Through this experimental analysis it will be shown which areas need to be focused upon.

The models introduced in this chapter were implemented, trained, and evaluated using the MATLAB Deep Learning Toolbox. All datasets were also generated in MATLAB, with the addition of the SOFA Toolbox MATLAB API, which is used for handling head-related impulse responses (HRIRs) throughout this thesis.

4.2 The Ideal Condition

To establish the challenges associated with the task of binaural sound source localisation (BSSL), direction of arrival (DoA) estimation in the ideal condition first needs to be considered. To do this, a simple convolutional neural network (CNN) will be trained and tested with audio data which has been simulated in the ideal condition. This will effectively show the accuracy of a BSSL system operating inside of an anechoic chamber.

4.2.1 CNN

To build a system, the actual design of the CNN itself needs consideration.

In order to minimise complexity, a single-branched CNN was employed. The result of this being that only one feature representation could be input to the system. Due to this, it was decided that magnitude would be used as it is theoretically possible to localise in the full azimuthal field based on monaural cues alone.

As this and following chapters are largely concerned with the data being used to train and test models, rather than the model architecture itself, the design parameters of the CNN are chosen due to their success in initial exploratory experimentation. A CNN of three layers was employed, each of these layers containing Convolution, Batch Normalisation and a ReLu activation function. This architecture is shown in Table 3. The growth pattern of increasing filter sizes and numbers through the layers is a typical approach for CNNs, batch normalisation is used to improve stability and max pooling is employed to restrain the size of the activations.

Table 3: CNN Architecture

Layer	**Hyperparameters**
Input Layer	(300, 6, 2)
2D Convolution	(2,2),8
Batch Normalisation	
ReLu	
Max Pooling	(2,2)
2D Convolution	(8,8),16
Batch Normalisation	
ReLu	
Max Pooling	(2,2)
2D Convolution	(16,16),32
Batch Normalisation	
ReLu	
Fully Connected Layer	72

The input layer parameter size was dictated by the dataset, and the 72 classes at the output correspond to the classes expanded upon in the following chapter.

4.2.2 Dataset and Training

The first consideration was which DoAs should be used in training and testing of the system. Azimuth is a continuous scale, in this case ranging from $-180°$ to $180°$, however the classifier inherently will quantise this as it only has a finite number of possible outputs. This can be approached by either limiting our data to a finite set of azimuthal positions, or by turning the full azimuthal plane into a series of 'bins' in the manner of a histogram.

For example if a CNN with 32 classes in its output layer was used, and these classes are evenly distributed among the full azimuthal plane such that the classes can be thought of representing:

$$\varphi_n = -180°, -170°, ..., 170° \tag{102}$$

or alternatively as representing 32 10° wide 'bins', such that:

$$\begin{aligned} bin_1 &\Rightarrow -180° \leq \varphi_1 < -170° \\ bin_2 &\Rightarrow -170° \leq \varphi_2 < -160° \\ &\qquad\qquad ... \\ bin_{32} &\Rightarrow 170° \leq \varphi_{32} < 180° \end{aligned} \tag{103}$$

While the second case may be a more useful way of thinking about these classifiers in the real world, where azimuth is a continuous scale which the classifier inherently has to quantise, under experimental conditions data can also be quantised such that only the azimuths shown in Equation (102) are used in the training and testing of the system.

This is advantageous as it reduces the number of HRIRs used, which are time-consuming to measure leading to databases tending to be limited in their number of DoA positions. This becomes even more advantageous when binaural room impulse responses (BRIRs) are considered later in this chapter.

In both cases, however, it should also be clear that there will be a trade-off between accuracy and computational resources. Increasing the number of classes reduces the azimuth bin size allowing the classifier to be more precise, however, in order to have the same amount of training files per class, the total number of training data files must increase.

In reality, as the distance between each DoA has decreased, and therefore the head-related transfer functions (HRTFs) have become more alike, the amount of training files per class needed to achieve the same level of classification accuracy will need to increase. Or, in other terms as the azimuth bin size has been reduced the classifier needs to be more accurate in order to select the new smaller azimuthal range.

In response to this trade-off, it was decided that 72 classes would be used, representing 5° increments:

$$\varphi_n = -180^\circ, -175^\circ, -170^\circ, ..., 170^\circ, 175^\circ \tag{104}$$

This has the advantage of improved comparison with other works that have also classified using 5° steps. The head, as it pertains to HRTFs, is a linear time-invariant (LTI) system. This means that binaural recordings do not need to be made with a dummy head microphone, but can instead be virtually created by convolving an audio signal with the impulse response (IR) of a dummy head. Under these ideal conditions, these can be represented as:

$$x_{L,R}[n, \varphi] = s[n] * hrir_{L,R}[n, \varphi] \tag{105}$$

where L, R are the channels left and right, $s[n]$ is the audio signal and $hrir[n, \varphi]$ is the HRIR. What needs to be considered is what audio is to be used as the signal, and how the HRIRs are to be created. Given that the azimuth has been discretised, and the channel can be thought of as a dimension to our data, this is re-expressed:

$$\begin{gathered} x_\varphi[n, i] = s[n] * hrir_\varphi[n, i] \\ i = \{L, R\} \end{gathered} \tag{106}$$

The choice of original audio signal used is likely to be significant in terms of generalisation, but in order to simplify experimentation in this and later chapters it was decided only one type of audio should be used. In Chapter 2.4 it was identified that significant interest in BSSL comes broadly with the end aim of localising and separating speech from background noise. Given this possible application, and that it will allow for fairer comparison with other systems, a heavy emphasis on localising speech is found in this work, and so speech is used as the audio signal. Specifically, speech is taken from the Librispeech corpus (Panayotov *et al.*, 2015) which contains English speech recordings taken from audiobooks, with a sample rate of 16kHz.

Sample rates of 16kHz are commonly seen in speech corpi for practical reasons, and so is a trait that the task of speaker localisation has inherited, however it does impose a Nyquist limit of 8kHz on the dataset. This is seen as acceptable in this case as in speech, most energy is below this limit, and the salient localisation cues are at lower frequencies, however it should be noted that audio above 8kHz is used by humans to resolve the cone of confusion and so a negative impact is possible.

HRIR recordings of a KEMAR mannequin were used, which were taken from the SADIE II Database (Armstrong *et al.*, 2018).

Having combined the HRIRs with the audio through convolution, so that now different versions of the same audio exist for 72 source locations, the audio is split into smaller sections. It was found during exploratory experiments that changing the length of the audio clip, and therefore the x-dimension in the subsequent time-frequency (TF)-Matrix has no significant effect upon the classification accuracy, and so a short length of 100mS (1600 samples) was chosen.

Rather than windowing the signal as a complete system may do, the audio was instead cut into 100mS files. This was done to achieve better generalisation for a given size of dataset, since no overlapping audio allows for a greater range of speech recordings.

CNNs are designed to be used with inputs of two or three dimensions in the same manner as an image file, the stereophonic audio files created, however, only exist in two: time and channels. Due to this the time series signals needs to be processed to create TF-matrices. This is achieved through use of a short-time fourier transform (STFT). The STFT was introduced in Equation (39) in Chapter 2.2, this is shown here for the specific case where azimuth and channels are considered:

$$X_{\varphi}[m, \omega, i] = \sum_{n=-\infty}^{\infty} x_{\varphi}[n, i] w[n - mH] e^{-j\omega n} \tag{107}$$

This yields a matrix of complex numbers, which was changed into a spectrogram through taking the magnitude:

$$spectrogram\{x_{\varphi}[n, i]\} = |X_{\varphi}[m, \omega, i]| \tag{108}$$

However, with regard to BSSL this spectrogram contains a large inefficiency: a large proportion of the frequency dimension ω is dedicated to the highest frequencies, which differs to human hearing which perceives frequency logarithmically. This can be improved upon with a conversion to the mel-scale: a perceptual scale closer to that of human hearing.

$$\text{mel-spectrogram}\{x_\varphi[n,i]\} = |X_\varphi[N,\omega,i]| \cdot M[\omega,k] \tag{109}$$

where M is a matrix made from a filterbank made up of triangular overlapped filters spaced across frequencies according to the Mel scale:

$$m = 2595 \log(1 + \frac{\omega}{1400\pi}) \tag{110}$$

This process was completed for 100 seconds of audio, yielding 1000 TF-Matrices per source direction, and 72,000 matrices in total to be used in training. This training set was then randomly split 80%-20% into training and validation files. The CNN was trained using stochastic gradient descent (SGD) with an initial learn rate of 0.0001, for a maximum of 50 epochs but was manually stopped at 17 epochs because loss was approaching zero.

4.2.3 Testing Method

The trained CNN was then used to classify audio taken from a new testing dataset.

The testing set was created using precisely the same method as employed for the training set, but using audio files taken from a later part of the speech corpus. This was undertaken to yield 300 audio files, and then matrices, at each source position for a total of 21,600 files.

The CNN was then used to classify these files. The first metric recorded was the Classification accuracy, which shows the ratio between successful and unsuccessful classifications (where 1 or 100% is completely successful).

$$Classification\ Rate = \frac{Correct\ Classifications}{N} \tag{111}$$

The next metric is the Front-Back Confusion Rate, which was designed to show the degree to which errors were caused by Front-Back Confusions.

This is achieved by mirroring the target value, Y_{Test}, to represent the front-back mirror value.

However, if the rate was then taken at this point in the same manner as classification rate, there would be two issues: Firstly, classifications in the bins adjacent to the front-back mirror would not count as front-back confusions, which would lead to misleading results as the front-back confusions would become lower in situations where the system is more conventionally inaccurate. Secondly, the cases where $Y_{Test} = Mirrored\{Y_{Test}\}$, being at $Y_{Test} = \pm 90^\circ$, correct classifications would be counted as front-back confusions.

To account for these issues, an algorithm was developed which introduced wider limits around $Mirrored\{Y_{Test}\}$ and ignores those false positives.

Algorithm 1 Front-Back Confusion Algorithm

```
$Y_{M\_Test} \leftarrow Mirrored\{Y_{Test}\}$
$Y_{UL} \leftarrow Y_{M\_Test} + 10^\circ$
$Y_{LL} \leftarrow Y_{M\_Test} - 10^\circ$
if $Y_{Test} \neq -90^\circ \pm 10^\circ \wedge Y_{Test} \neq 90^\circ \pm 10^\circ$ then
    if $Y_{M\_Test} \geq Y_{LL} \wedge Y_{M\_Test} \leq Y_{UL}$ then
        Front-Back Confusion
    else
        No Confusion
    end if
else
    No Confusion
end if
```

Having applied Algorithm 1 to all testing files, the rate is simply:

$$\text{Front Back Confusion Rate} = \frac{\text{Front Back Confusions}}{N} \tag{112}$$

These two metrics give insight into the performance of classification, however it is important to remember that the output of the classifier is a number on a continuous scale. To gain insight into the degree to which incorrect classifications are incorrect, the root mean square error (RMSE) was also calculated.

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N}(Y_{Pred}[n] - Y_{Test}[n])^2}{N}} \tag{113}$$

This alone, however, would induce a higher level of error than it should owing to the circular nature of the results.

Consider the case $Y_{Test} = -180°$ and $Y_{Pred} = 175°$. Calculating the error using the difference as per Equation (113) would suggest 355° of error, however the distance between these two DoAs is actually 5° as $180 = -180°$.

To tackle this, if Y_{Test} and Y_{Pred} are in different halves of the azimuthal plane, the degrees adjusted to by ±360° to the nearer point. This circular error is referred to as root mean square localisation error (RMSLE).

Algorithm 2 Algorithm for making RMSLE circular

```
if $|Y_{Pred} - Y_{Test}| > 180°$ then
    if $Y_{Test} < 0°$ then
        $Y_{Test} \leftarrow Y_{Test} + 360$
    else
        $Y_{Test} \leftarrow Y_{Test} - 360$
    end if
end if
```

Finally, given that the RMSLE could potentially be disproportionately affected by front-back confusions, since these will tend to induce high degrees of error, another metric which will be referred to as RMSLE with Mirroring is used.

RMSLE with Mirroring constrains all results to a 180° azimuthal plane through mirroring, so as to effectively eliminate all front-back confusions.

Having applied Algorithm 3 to all testing files, a value for RMSLE with Mirroring can be calculated using Equation (113).

4.2.4 Results

The results of training and evaluating the CNN on the anechoic training and testing datasets are shown in Table 4.

Algorithm 3 Mirroring Algorithm for RMSLE with Mirroring Metric

```
if $Y_{Test} < -90° \vee Y_{Test} > 90°$ then
    $Y_{Test} \leftarrow Mirrored\{Y_{Test}\}$
end if
if $Y_{Pred} < -90° \vee Y_{Pred} > 90°$ then
    $Y_{Pred} \leftarrow Mirrored\{Y_{Pred}\}$
end if
```

Table 4: Results from training and testing a CNN in ideal conditions

Classification Accuracy	99.91%
Front-Back Confusion Rate	0%
RMSLE	0.2°
RMSLE with Mirroring	0.2°

4.2.5 Discussion

It can be seen in Table 4 that the system shows exceptional performance, approaching perfect. This result, however, lacks real-world utility as it the acoustic conditions are unrealistic; it must also be established whether such a system is able to generalise to other conditions.

4.3 Diffuse Noise

Audio, as a medium, is very prone to the unintended inclusion of noise. This can be added to a signal through a variety of ways such as electric interference, from quantisation during analogue-to-digital conversion, or perhaps most significantly through the recording of unwanted noise in an acoustic environment. It is at this point, however, that it needs to be pointed out that in the task of sound localisation not all of these types of noise are the same. If one considers the process of making a binaural recording noise inflicted upon the signal after the point of transduction will not have had any directional cues imparted upon it, which in this work is referred to as diffuse noise.

Typically in other audio signal processing tasks such as automatic speech recognition (ASR), the more significant challenge is unwanted sounds which have been recorded from the acoustic environment, such as a passing car. As these sounds emanate from their own DoA, they shall be recorded with binaural cues, and so need to be thought of a separate problem, one which will be explored later in this chapter.

In this sub-chapter, a CNN which has been trained in ideal conditions shall be tested using audio which has had diffuse noise added to the signal, to show the relation between diffuse noise and system performance.

4.3.1 Testing Sets

As previously mentioned, the CNN previously trained under ideal conditions was used in this experiment, and so the only new element of the method is the synthesis of the testing set.

Each testing set will also consist of 300 files for each source direction, however in this case there were ten different testing sets to represent 10 different levels of signal-to-noise ratio (SNR). These levels represented 12 dB increments starting at 0dB, and ending at 108dB: these numbers being chosen to represent the full range from noise equalling signal in amplitude, to the theoretically lowest possible audible noise floor.

The testing set was generated in the same manner as described in Chapter 4.2.2, but prior to the Fourier transform, a pink noise signal generated on each iteration, and scaled to the correct value, was added to the signal. Pink noise was chosen as avoid for bias towards higher octave bands, as pink noise contains equal energy per octave.

To scale the signals, both the noise and speech signal were normalised to -6dB to avoid the sum's total exceeding full-scale, and then the noise was scaled by the inverse of the SNR value. This alone, however, would not yield the right result as while normalising to -6dB for the noise would also yield a dB(root mean square (RMS)) value of -6dB, for the speech, however, the peak amplitude will be much higher than the RMS. The dataset was tested to see on average what the difference between peak and RMS is for the speech samples, finding a value of -14.46dB. To account for this, then, the noise was also scaled by -14.46dB. The noise was then added to the speech, signal normalised back to 0dB, and spectrograms created.

The CNN was used to classify each testing set individually, and the same metrics as the previous sub-chapter were compiled. These are presented as plots with signal-to-noise ratio on the x-axis, and a line of best fit created through polynomial regression has been added for each result.

4.3.2 Results

The following figures show the results of testing the CNN trained in the ideal conditions tested with data with diffuse noise added. Figure 44 shows the classification rate plotted against SNR, Figure 45 shows the front-back confusion rate plotted against SNR, and Figure 46 shows the RMSLE plotted against SNR.

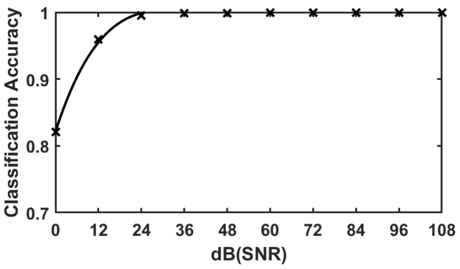

Figure 44: Classification Rate with respect to SNR

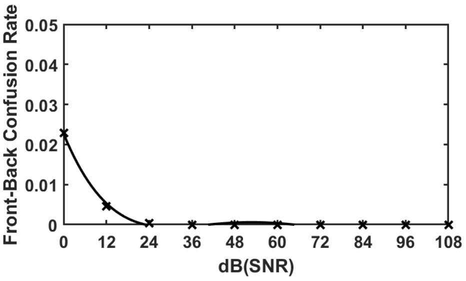

Figure 45: Front-Back Confusion Rate with respect to SNR

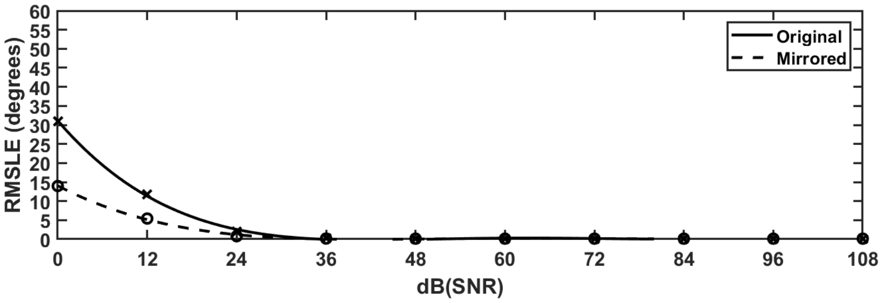

Figure 46: RMSLE with respect to SNR

4.3.3 Discussion

From Figures 44-46 it is immediately noticeable that despite the CNN not being trained for such a task, its performance in the presence of diffuse noise is still fairly robust, with a noticeable effect only being found for SNRs below $24dB$. Even at SNR of $0dB$, classification accuracy is still in excess of 80%, though the RMSLE of $30°$

suggests these errors are not tightly clustered around the target azimuth, despite the low incidence of front-back confusions.

Plotting the results as a histogram for individual azimuths can provide more insight into these errors. The classifier shows different behaviour at different azimuths; of the azimuths with higher levels of error, three behaviours can be highlighted, which are as follows.

It can be seen in Figure 47 that the errors are somewhat distributed normally around the target azimuth, however outliers explain the relatively high RMSLE value.

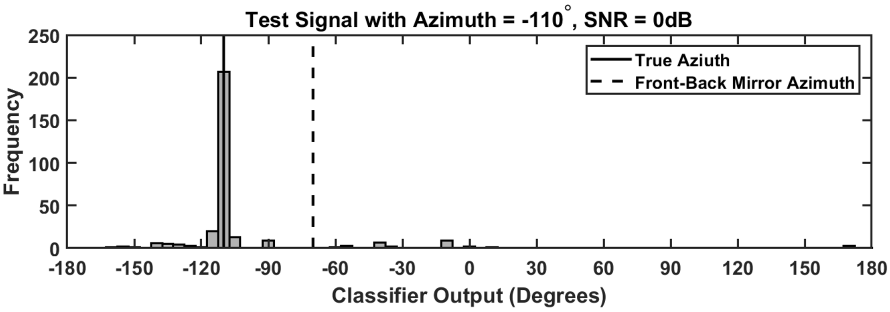

Figure 47: Classification Results for $\varphi = -110°$ and $0dB(SNR)$

Meanwhile, Figure 48 shows a clearly double peaked distribution, with one peak at the true azimuth and one at close to the front-back mirror.

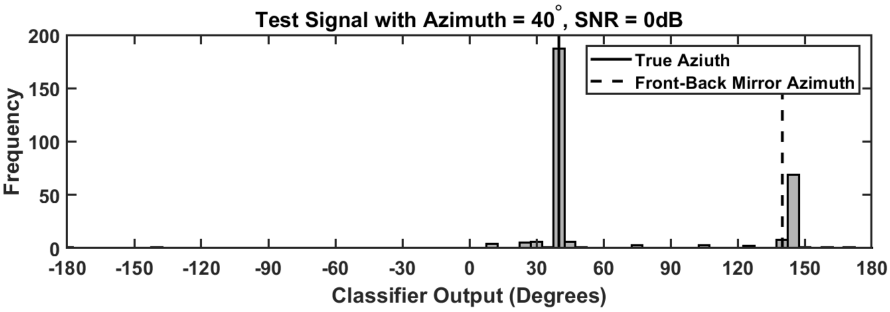

Figure 48: Classification Results for $\varphi = 40°$ and $0dB(SNR)$

This behaviour is unremarkable in that it was hypothesised and then occurred. What is significant is that the front-back reversal behaviour has occurred at a much lower rate than expected, with more error being incurred by the previous factor and the error shown in Figure 49.

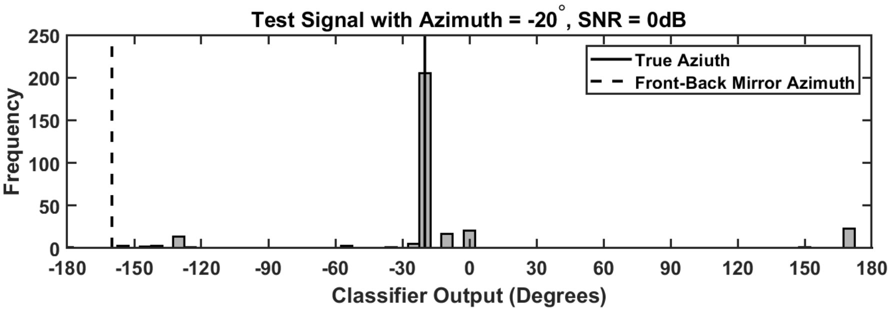

Figure 49: Classification Results for $\varphi = -20°$ and $0dB(SNR)$

Figure 49 shows to some degree the previous two errors, but also another error worth highlighting, being the spurious peak around 170°. It was found that this error sometimes occurred regardless of target azimuth. It is likely when the CNN is having difficulty classifying, some unknown bias forces this error, unfortunately inducing a high degree of error. This effect is also visible in Figure 48.

It can be seen that the RMSLE with Mirroring metric follows about half of RMSLE, despite lack of front-back confusions. This is not overly surprising, as constraining the full azimuthal field from a total of 360° to 180° halves that possible error, though another suggestion is that the errors which do occur are somewhat randomly distributed, rather than clustered around a peak.

4.4 Interfering Sound Source

As introduced in Chapter 4.3, from the perspective of BSSL unwanted noise can occur in two varieties: diffuse, or from a distinct sound source. This sub-chapter will show the result of attempting to classify a testing set using a CNN, trained under ideal conditions, in the presence of a second interfering sound source.

4.4.1 Method

The same CNN trained under ideal conditions was also used for this test, with the testing set again being changed.

It is hypothesised that for noise with the HRTF applied, the presence of these monaural cues shall change the reaction of the CNN, and so for a given SNR level, the result will not be the same. To this end, the type of noise needs more careful consideration as the degree to which the audio is similar to the training set will probably alter this effect.

With this in mind, it was decided to test for two different types of noise: a second speech signal, and a noise signal. Both these signals are tested at noise levels corresponding to the SNR values in the previous sub-chapter. This can be thought of as SNR here too, but is referred to instead as the ratio between sound source levels.

The only difference, then, in the testing set generation is the manner in which the interfering sound sources are synthesised. For the noise signal, pink noise was generated and then convolved with a randomly selected HRIR. This random selection does leave a $^1/_{72}$ chance that the noise is coincident with the original sound source, as there are 72 possible DoAs in this test, however this is adequately small to be considered negligible. The noise was then scaled in the same manner as the diffuse noise, and summed to the speech signal.

For the speech signal, ten segments of speech were taken from a much later point of the speech corpus. For each (primary) audio file, one of the ten speech files were again chosen at random, and convolved with an HRIR also chosen at random. This was then normalised to the same level as the primary audio file, and then scaled by the intended ratio. The resulting audio could then be transformed into spectrograms.

The CNN was then used to classify the testing sets. The results are shown on plots with the sound source level ratio on the x-axis, with both signals being shown on each plot, except for RMSLE which for clarity is still separated between two plots.

4.4.2 Results

The following figures show the results of testing the CNN trained in the ideal condition, tested with data with directional noise added. Figure 50 shows the classification rate plotted against SNR, Figure 51 shows the front-back confusion rate plotted against SNR, and Figure 52 shows RMSLE plotted against SNR.

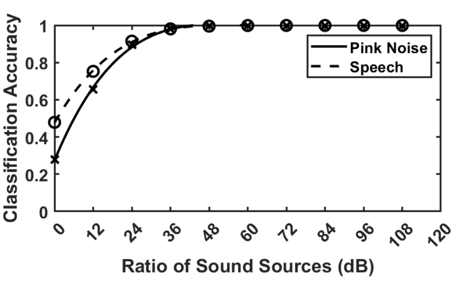

Figure 50: Classification Rate with respect to interfering sound level ratio

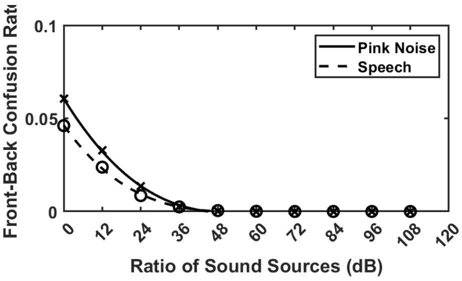

Figure 51: Front-Back Confusion Rate with respect to interfering sound level ratio

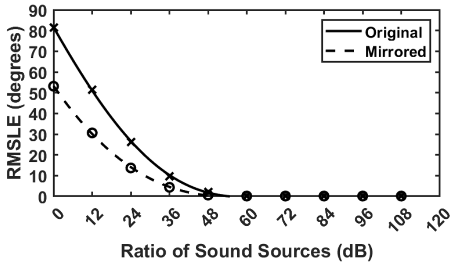

Figure 52: RMSLE for interfering pink noise with respect to interfering sound level ratio

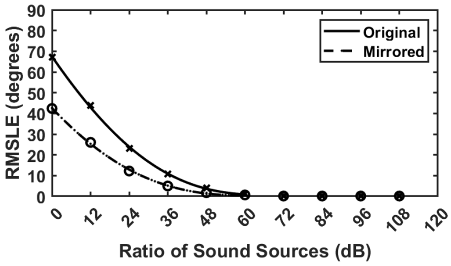

Figure 53: RMSLE for interfering speech with respect to interfering sound level ratio

4.4.3 Discussion

The most immediately noticeable finding here is that both the interfering sound sources, at equivalent sound levels, have a much more detrimental effect upon sound localisation than diffuse noise does.

Additionally, it is clear that the accuracy of the system actually performs better under the presence of an interfering speech signal rather than an interfering noise signal. This is surprising, since given that the CNN has been trained using speech, it seems logical that the system would erroneously output the azimuth of the interfering sound source more often for speech than noise.

The presence of this sort of error is supported by the classification accuracy being approximately 50% when the two speech levels are equal in level which is what would be expected, however, for the pink noise accuracy is below even that at approximately 30%, suggesting the system is suffering some sort of more severe error.

Given that the second source azimuths are randomly distributed, plotting a histogram is unhelpful, since while it does show the error being randomly distributed it is unclear if this is due to the localiser identifying the HRIR of the second source, or just general error.

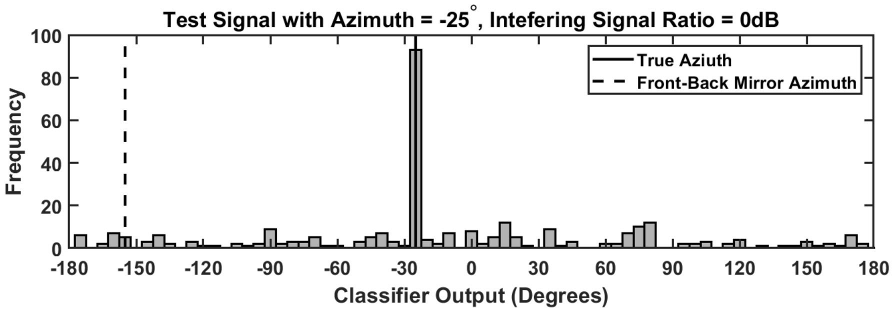

Figure 54: Histogram showing classifier output at azimuth of $-25°$ for an interfering pink noise at $0dB$

To gain further insight, another testing set was created where the azimuth of the second sound source is always 40°.

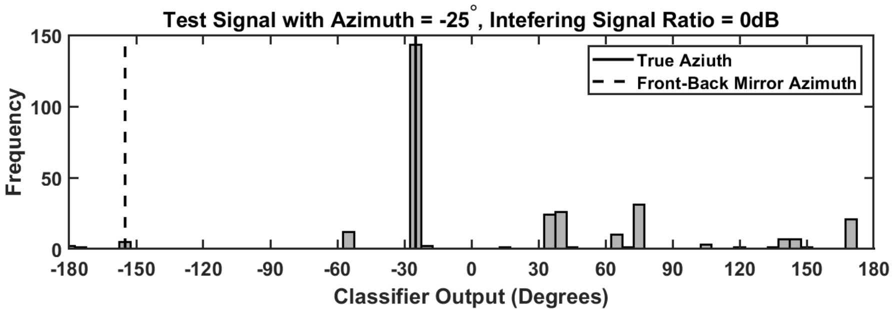

Figure 55: Histogram showing classifier output at azimuth of $-25°$ for an interfering pink noise at $0dB$ with azimuth of 40°

From Figures 54 and 55 it can be seen that indeed, to a certain degree the random distribution was being caused by the classifier identifying the secondary source's HRTF, however other spurious errors do remain.

Figure 56 contains a better demonstration of how the accuracy for the pink noise signal has dropped below 50%, with a second peak not just being seen around 40°, but at other random positions.

Given this behaviour, we can therefore expect that in the presence of a second source the effects of bad SNR, as well as of a competing sound source, will in fact sum to create even greater error.

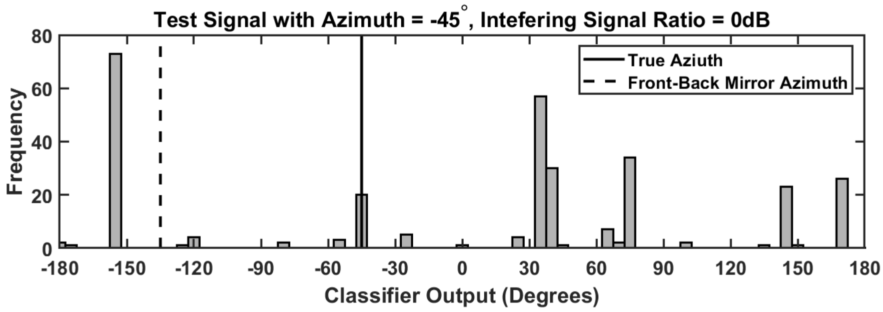

Figure 56: Histogram showing classifier output at azimuth of $-45°$ for an interfering pink noise at $0dB$ with azimuth of $40°$

Looking at the equivalent situation for the speech signal, in Figure 57, it can be seen that similar behaviour occurs but the number of mis-classifications are fewer.

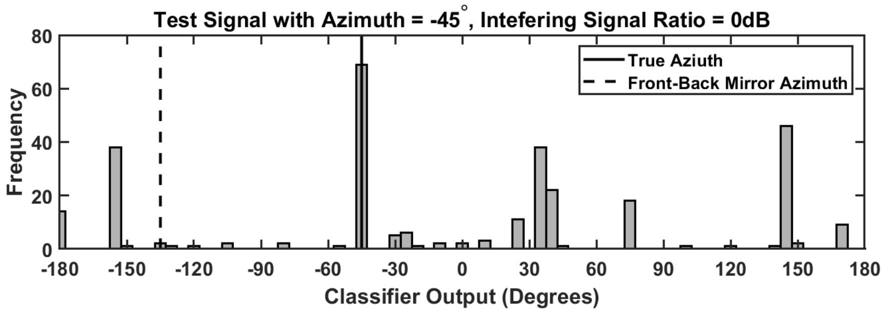

Figure 57: Histogram showing classifier output at azimuth of $-45°$ for an interfering pink noise at $0dB$ with azimuth of $40°$

4.5 Noise Mixture

Realistic sound environments would not contain just one interfering sound source, but a mixture of different sound sources of different energy. An environment like this is made up of interfering sound sources, so expecting similar results is a reasonable hypothesis, however as the number of sound sources is increased the resulting mixture of sound will become closer to that of diffuse noise.

This sub-chapter will test the performance of the CNN using testing data containing such a noise mixture, with varying SNR and number of sources. This will show whether the cocktail party phenomenon is implicit, or if such behaviour needs to be learned by the model.

4.5.1 Method

The fundamental component of the noise mixture was sound sources made by combining pink noise with an HRIRs of random azimuth, achieved in precisely the same way as in Chapter 4.4.1. This however was done for Q number of sound sources. Each of the sound sources for every mixture was normalised to a random level between 0 and 1, and summed into the mixture. The entire mixture was then normalised to the required levels to represent SNRs of $0dB, 12dB, 24dB, 36dB$.

$$\begin{aligned} y_{L,R}[n] &= B_{SNR}(\sum_{q=1}^{Q} A_{rand}(pink[n] * hrir_{L,R}[n, \varphi_{rand}])) \\ Q &= 1, 2, 3, ..., 10 \\ B &= 1, 0.25, 0.0063, 0.0016 \end{aligned} \tag{114}$$

Where $y[n]$ is the resulting noise mixture, Q is the number of sound sources, φ_{rand} is a random azimuth, A_{rand} is a random number between 0 and 1, and $pink[n]$ is a randomly generated pink noise signal.

Each of these noise mixtures can then combined with the same training files used in previous chapters.

$$x_{L,R}[n, \varphi] = (s[n] * hrir_{L,R}[n, \varphi]) + D(y_{L,R}[n]) \tag{115}$$

where D is the inverse of the average level of the speech combined with the HRIRs.

4.5.2 Results

The following figures show the results of testing the CNN trained in the ideal condition, tested with data with noise mixtures added. Figure 58 shows the classification rate plotted against SNR, Figure 59 shows the front-back confusion rate plotted against SNR, and Figure 60 shows RMSLE plotted against SNR.

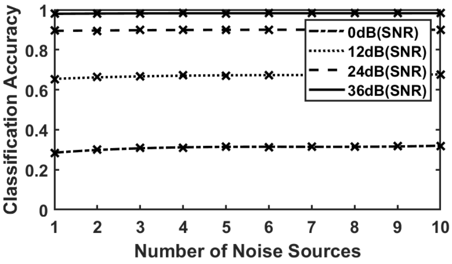

Figure 58: Classification Accuracy for Testing Set with Noise Mixtures of varying levels and Number of Sources

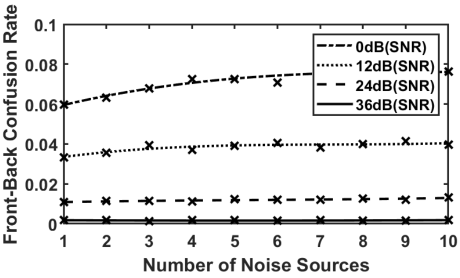

Figure 59: Confusion Rate for Testing Set with Noise Mixtures of varying levels and Number of Sources

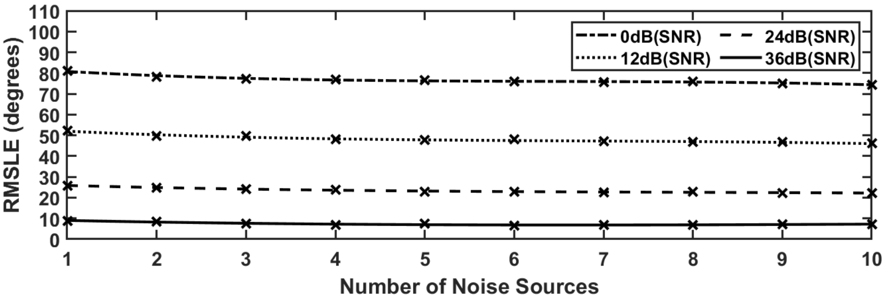

Figure 60: RMSLE for Testing Set with Noise Mixtures of varying levels and Number of Sources

4.5.3 Discussion

There is one obvious factor uncovered in the results: that the number of sound sources in the noise mixture has no significant influence on the performance of the system. It should be noted at this point, that the anechoic conditions may have caused influence here, with it being possible that a different relationship would be seen if each of the sound sources also had reflections arriving at the sensor, thus more closely resembling completely diffuse noise.

Given this, then, it can be concluded that if a system needs to display 'cocktail-party' like behaviour, where it is able to ignore unwanted elements of a sound scene, this behaviour would need to be explicitly taught to the system.

4.6 Mismatched HRTFs

In need of consideration is whether a system is only intended to be used for a specific sensor, or whether it should be able to operate using the HRTFs of different binaural arrays. The latter is referred to as the Mismatched HRTF condition, and this sub-chapter shall show the results of the CNN's DoA estimation in this condition.

4.6.1 Method

Six additional HRIRs were taken from the SADIE II database, being the subjects H3-H8. While the database contains more subjects than this, they do not include HRIRs for the azimuths and elevation combinations being used in these tests, and so to avoid having to use interpolation are not used. H3-H8 are all human subjects, and so should all have unique HRTFs from not only each other, but from the mannequin head as well.

The HRIRs were then combined with the same audio files as in the rest of the chapter to create a test set of 300 files per azimuth per head. These were again transformed into mel-spectrograms, and the same CNN used to attempt to classify.

4.6.2 Results

Given the discrete nature of the independent variable, these results are shown in column charts, with no line of best fit. Figures 61, 62, and 63 show classification rate, front-back confusion rate, and RMSLE for the different HRTFs

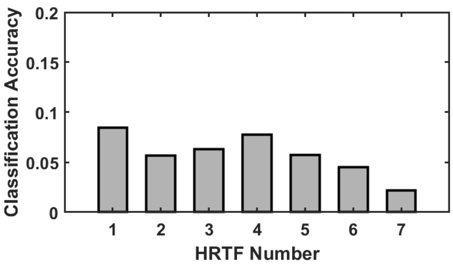

Figure 61: Classification Rate for different HRTFs

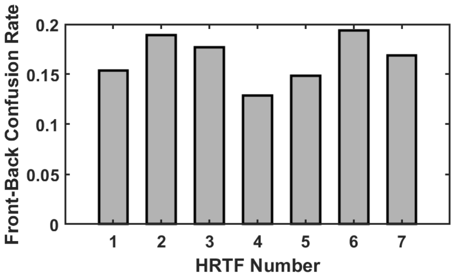

Figure 62: Front-Back Confusion Rate for different HRTFs

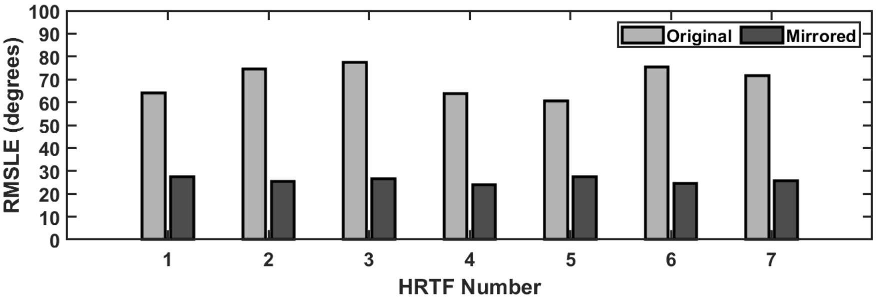

Figure 63: RMSLE for different HRTFs

4.6.3 Dicussion

In Figures 61-63 it can be clearly seen that the classification rate under the mismatched-HRTF condition is extremely poor. Noticeably, despite such a low instance of correct classifications, a high rate of front-back confusions is present, beyond the 5% figure expected had the classifications been completely random. This is insightful in showing that front-back confusions can be problematic in BSSL with CNN, despite the previous tests an additive noise not inducing such errors.

Given the very low performance, it is possible to conclude that these systems will not be suitable for use on different sensors unless particular attention is devoted to this problem.

4.7 Reverb Time

In the previous sub-chapters a commonly encountered pattern is that increasingly adverse acoustic conditions, such as an increasing noise level, result in lower level of classification performance. It is a reasonable hypothesis, then, that an increasing reverberation time, R_T, will have a similar effect. To test this, an experiment was conducted in the same manner of previous sub-chapters where a system is trained upon an anechoic dataset, and then tested upon data which has been combined with a binaural array's impulsed response in a reverberant room; a BRIR.

4.7.1 Training Dataset

The anechoic dataset for training the system was constructed in the same manner as for the anechoic dataset constructed in Chapter 4.2.2, however a different set of HRIRs were needed to be used, as the HRIRs need to match the binaural array and distance coordinates used in the BRIRs. In this case, that being Subject-21 from the CIPIC HRTF database (Algazi *et al.*, 2001), being a KEMAR mannequin with small pinnae. The CIPIC database, however, at 0° elevation only contains HRIRs for 50 azimuths, which are also spaced by 5° but do not include all of the previously used azimuths. The azimuths available can be seen in Figure 64

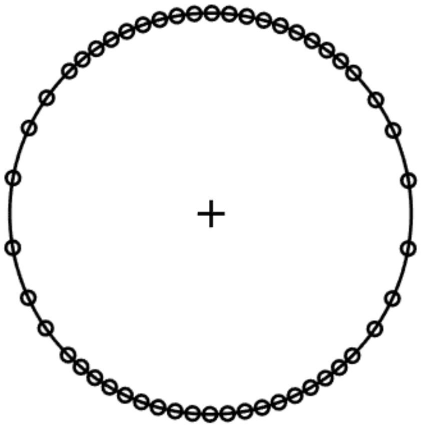

Figure 64: Source directions available in in CIPIC database

Accordingly, the dataset was constructed for 50 azimuths, and although this will mean that such a system will not be able to localise any of these missing directions, the effect upon the metrics being used to assess these systems will not be significant. As per Chapter 4.2.2, 1000 audio files per source direction were created.

4.7.2 Testing Dataset

To create the testing dataset, audio files were combined with BRIRs. Since no dataset of BRIRs recorded for sources in the full azimuthal field is known to be publicly available, BRIRs were instead generated using the image source method (ISM). ISM was used rather than wave based due to practical restraints, as accurately rendering unique room geometries for all combinations of source and receiver would have been a resource-intensive task to achieve with finite element method (FEM). To achieve this, an existing library which employs the ISM was used (Mandel, 2013) to generate the BRIRs.

BRIRs were generated for an arbitrary room size of $[4, 4, 3]$ metres. Four BRIRs were generated for this one room size so as to achieve an R_T of $0.5, 1, 1.5, 2$ seconds, where R_T is RT60, the length of time taken for reverberation to attenuate by $-60dB$. These times were chosen so as to represent a realistic range of reverb times from dull to lively.

This was achieved by altering the room's boundaries' absorption coefficients which would result in these target reverb times, as calculated by the Sabine equation:

$$R_T = \frac{0.161V}{\sum_{n=1}^{N} S_n.\alpha_n} \tag{116}$$

where V is the volume of the room, N is the number of boundaries, S_n is the surface area of a boundary, and α is the absorption coefficient of that boundary. A set of uniform absorption coefficients for a room, therefore, can be calculated with:

$$\alpha = \frac{0.161V}{(\sum S_n).R_T} \tag{117}$$

The Sabine equation for predicting reverberation time is a less accurate formulation than the Eyring equation, due to an underestimation of energy loss (Stephenson, 2012). In this case, the simpler Sabine formulation is still used as the accuracy is not seen as of high importance, as it is the general relationship between reverb time and accuracy which reveals insight into robustness of the system, and the arbitrary reverb times used are not of particular significance.

The BRIRs were then convolved with the same 300 audio files used in previous testing sets, to create four sets of testing files representing the different reverb times.

4.7.3 Results

The classification accuracy, front-back confusion rate, and RMSLE are plotted against reverb time in Figures 65-66.

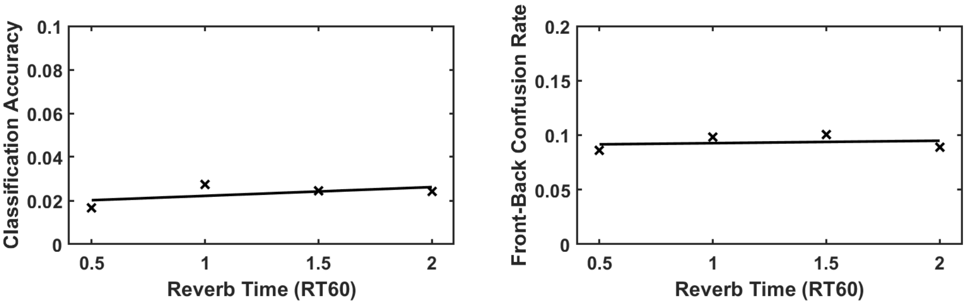

Figure 65: Classification Accuracy & Front-Back Confusion Rate with respect to R_T for a model trained on anechoic binaural audio

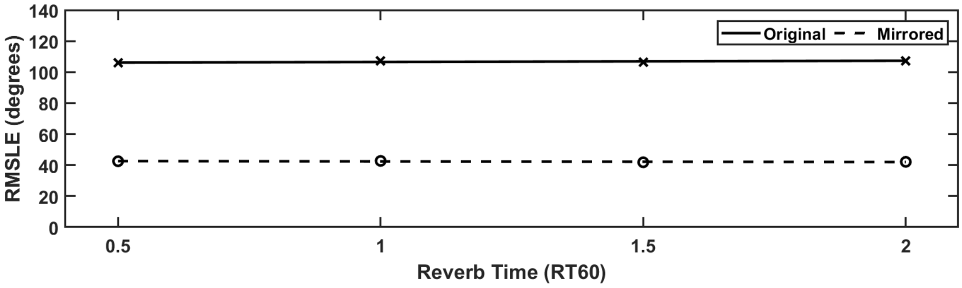

Figure 66: RMSLE with respect to R_T for model trained on anechoic binaural audio

4.7.4 Discussion

In Figures 65 - 66 it can clearly be seen that the system performs extremely poorly in the presence of the reverb, rarely correctly classifying. Notably it also does not seem to show correlation between reverb time and performance, despite the increasingly difficult acoustic conditions. This is reminiscent of previous tests where the model has failed to generalise, such as in the mismatched-HRTF condition.

4.8 Changing Reverb Time, without the Generalisation Challenge

To test whether the poor performance is the result of poor generalisation, the previous testing set is used to test a new model, one that is trained on training data created using the same BRIRs as the testing set.

4.8.1 Training Dataset

The same speech files from described in Chapter 4.2.2 are used to create the dataset, however rather than being convolved with HRIRs, the speech files were convolved with each of the BRIRs.

4.8.2 Results

The classification accuracy, front-back confusion rate, and RMSLE are plotted against reverb time in Figures 67 & 68.

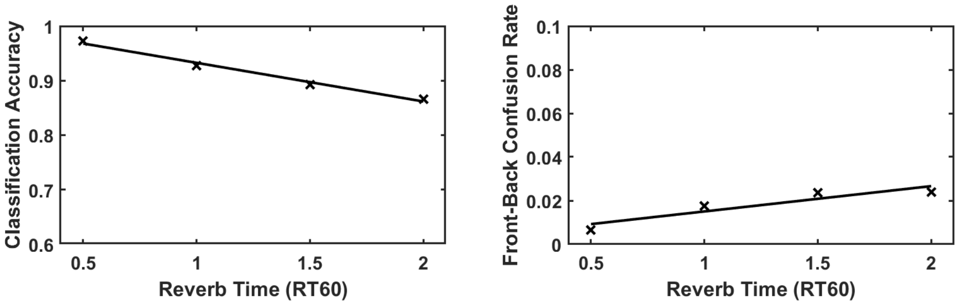

Figure 67: Classification Accuracy & Front-Back Confusion Rate with respect to R_T for a model trained using matched BRIRs

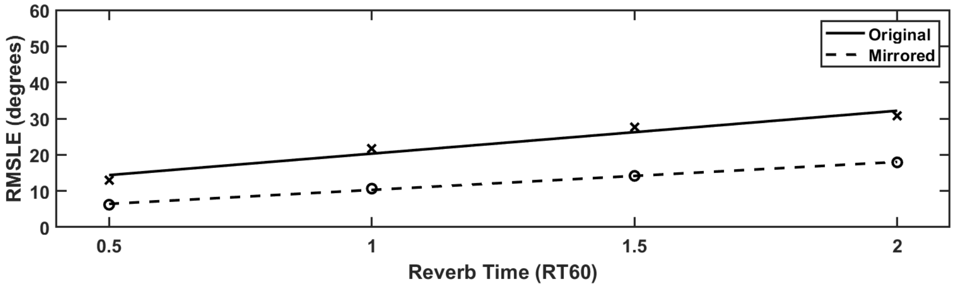

Figure 68: RMSLE with respect to R_T for a model trained using matched BRIRs

4.8.3 Discussion

It is unsurprising that the model using a training dataset which includes BRIRs shows an increased performance, however the difference is stark with the system now approaching 100% accuracy. Notable also is that the results now show the expected inverse proportionality between R_T and accuracy. This further supports the hypothesis that the challenge when using BRIRs is not just the increased complexity from the spatial domain, but that the model is unable to generalise for different BRIRs in the same way that it cannot generalise for different HRIRs.

4.9 Effect of Changing Room Dimensions

Rather than testing the effect of changing the R_T, the effect of changing the room by altering its dimensions was tested next. This was undertaken to expose whether this generalisation issue is tied to the length of R_T.

This was achieved by training two models on the same testing set, one trained on anechoic data and one trained on some of the rooms present in the testing dataset.

4.9.1 Training & Testing Datasets

BRIRs

To create the BRIRs, ISM was once again applied. However, instead of changing the absorption coefficients of each room to vary reverb time, an R_T of 0.5 seconds was used in all cases. To make each room unique each room was given unique dimensions, made from three randomly generated integers between 1-10, representing metres. These dimensions are shown in Table 5. Uniform absorption coefficients were once again calculated so as to achieve an R_T of 0.5s.

Table 5: Room Dimensions for 10 Room Dataset

Room Number	1	2	3	4	5	6	7	8	9	10
Width (m)	9	5	3	9	4	6	10	4	7	1
Length (m)	7	9	5	1	5	4	3	1	8	6
Height (m)	6	2	6	7	9	3	5	10	9	2

Anechoic Training Dataset

The anechoic dataset described in Chapter 4.7 is also appropriate for this experiment, the model trained using that dataset.

Rooms 1-5 Training Dataset

To show the effect training a model on some but not all of the rooms, a training dataset containing files which used BRIRs from the first 5 rooms was created. To achieve this, the same 1000 speech samples used in the anechoic training dataset were combined with each of BRIRs for Rooms 1-5, yielding a total of 5000 audio files per azimuth: 250,000 in total. These were then transformed into spectrograms.

The same model was trained on this training dataset for 100 epochs using a SGD optimiser.

Testing Dataset

The testing dataset was created by convolving the same 300 speech samples from previous experiments with all 10 of the sets of BRIRs, and the performance metrics for each of these testing sets were calculated.

4.9.2 Results

The classification accuracy, front-back confusion rate, and RMSLE are plotted against room number in Figures 69, 70 & 71, where for each room two different bars denote the two training datasets used to train the models: Anechoic and Rooms 1-5.

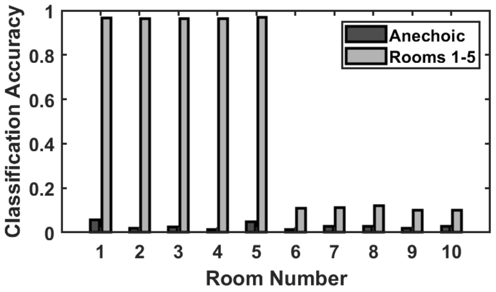

Figure 69: Classification Accuracy for different rooms when trained using anechoic data

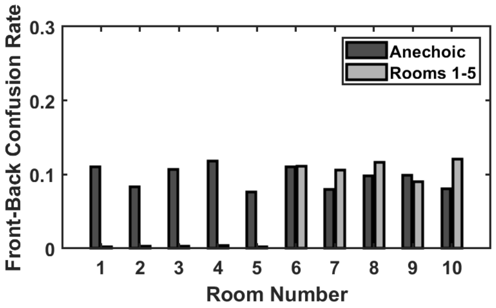

Figure 70: Confusion Rate for different rooms when trained using anechoic data

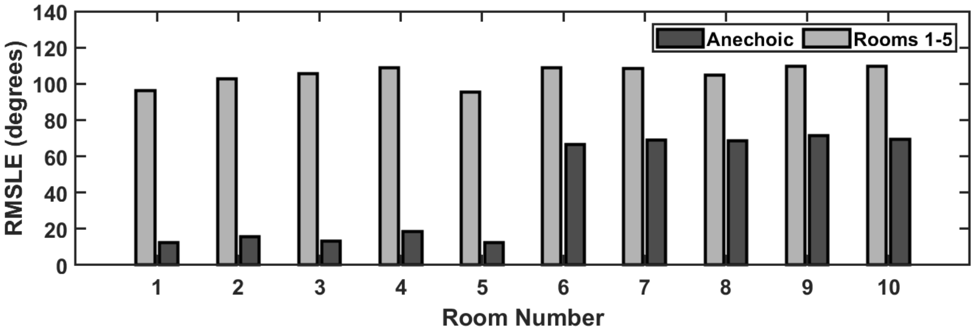

Figure 71: RMSLE for Different Rooms when trained using Anechoic Data

4.9.3 Discussion

From Figures 69 and 71, it can be seen that as expected the anechoic dataset performed poorly on all ten rooms, meanwhile the dataset trained on rooms 1-5 performed convincingly on the observed rooms, but poorly on the unobserved rooms. It is very clear that there is still a generalisation issue: the anechoic data trained model achieves very poor performance on all of the models, and the model trained on rooms 1 to 5 achieves very good performance on those rooms, but poor performance on the rest of the rooms.

It is notable that the performance of the model trained with reverberant data does achieve slightly better performance than the anechoically trained one on rooms it has not observed, suggesting improved generalisation would be possible through increasing the number of rooms the model learns from. However, this could also be simply due to the training dataset for this model just being larger causing a longer training process.

4.10 Training with Natural Binaural Data

This issue of the models not generalising to different room impulse responses (RIRs) has thus far only been substantiated using synthetic data. A reasonable hypothesis, then, is that this behaviour is caused in some way by the process of synthesising BRIRs, rather than it being a true phenomenon.

To test this hypothesis, datasets for training and testing were also created using recorded BRIRs in real rooms.

4.10.1 Training & Testing Datasets

The BRIRs were taken from the Surrey BRIR dataset (Hummersone *et al.*, 2010). This dataset contains BRIRs of 4 different rooms and an anechoic space, the BRIRs of the four rooms were used. The reverb times of these rooms can be seen in Table 6.

Table 6: Reverb Time of Rooms in the Surrey BRIR Database

Room Number	1	2	3	4
RT60	0.32	0.47	0.68	0.89

Notably, the Surrey BRIRs do not contain recordings for the entire azimuthal field, but are restrained to the frontal hemifield, such that:

$$-90^\circ \leq \varphi \leq 90^\circ \tag{118}$$

This means that performance with regard to front-back confusion cannot be measured. However, the BRIR dataset was nonetheless used as no dataset of recorded BRIRs in multiple rooms is known to the researcher.

Thus, only the BRIRs for the source directions in Figure 64 which are in frontal field were used, a total of 25 source directions. This can be seen in Figure 72.

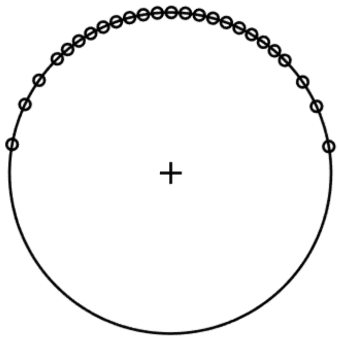

Figure 72: Source directions taken from Surrey Database

The training dataset was created by convolving the BRIRs of the first two rooms with the same 1000 audio files used previously, so a total of 2000 files per source direction.

Similarly, the testing dataset was created by combining the same 300 audio files as in the previous experiments with all the BRIRs of all four rooms at the 25 different source directions.

4.10.2 Results

Figure 73 & 74 show classification accuracy and RMSLE for the model trained on rooms 1 and 2 of the Surrey BRIRs, and evaluated with all four rooms.

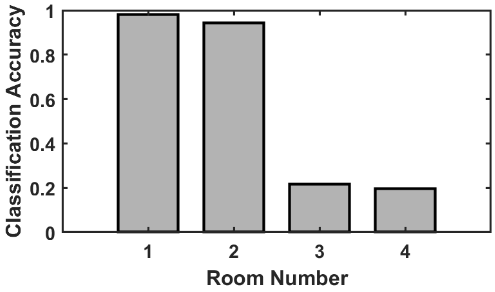

Figure 73: Classification Accuracy for Different Rooms when trained using Natural Binaural Data

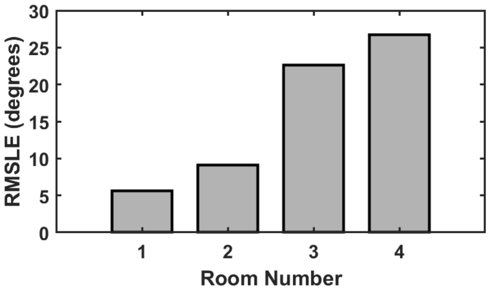

Figure 74: RMSLE for Different Rooms when trained using Natural Binaural Data

4.10.3 Dicussion

From both the results for classification accuracy and RMSLE a similar behaviour can be seen, the model is able to achieve high performance for the rooms to which the neural networks have been exposed, and much poorer performance for the ones to which they have not. This differs from the results seen when training with the synthetic binaural data only in that the accuracy for the unknown rooms is slightly higher, despite the total number of rooms in the training set being lower. This does suggest this effect is slightly mitigated when using real BRIRs rather than synthesised ones, but the degree of this effect is not significant.

4.11 Training with larger numbers of BRIRs

It has been established that CNNs have particular difficulty localising in unknown locations. Given that this is a generalisation issue, a reasonable hypothesis is that increasing the total number of rooms in the training set will increase the generalisation performance as this is typical behaviour in CNN models. This hypothesis, therefore, was tested.

4.11.1 Training and Testing Dataset

A larger number of BRIRs were created by generating more room geometries. This was achieved through the same method described in Chapter 4.9.1, using the same range of 1-10 metres. This was undertaken for create 68 room geometries. For each of these room geometries, 3 sets of BRIRs for reverb times of 0.5, 1 and 1.5 seconds, also achieved by uniformly altering the absorption coefficients. These BRIRs were then equally split into training and testing, such that 34 is dedicated to each.

For the training dataset each of the 50 DoAs, the same 1000 audio files were convolved with the BRIRs of the three reverb times for the 34 room geometries dedicated training, as well as the anechoic HRIRs. The testing dataset was similarly made by combining the same 300 audio files with the BRIRs of the three reverb times for the 34 room geometries dedicated to testing, along with the anechoic HRIRs.

In addition, another testing dataset was created in which the 300 audio files were convolved with the BRIRs of the 34 room geometries used in training. This means that there are two testing datasets, one containing rooms known to the model in that it has been trained on these, and one containing rooms unknown to the model.

4.11.2 Results

Results are presented in terms of reverb time, for both the testing datasets of known rooms and unknown rooms so as to represent the level of generalisation. This is plotted for the metrics classification accuracy, front-back confusion rate and RMSLE in Figures 75-77.

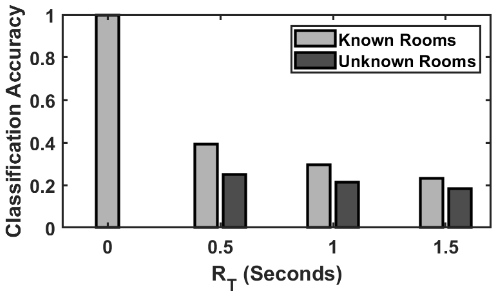

Figure 75: Classification Accuracy for Different Rooms when trained using a large number of BRIRs

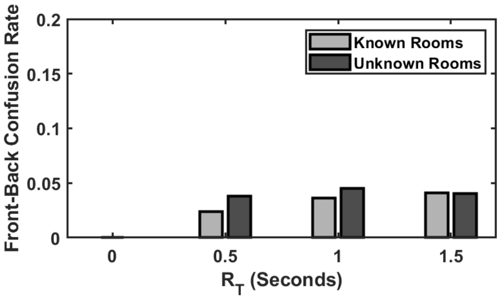

Figure 76: Confusion Rate for Different Rooms when trained using a large number of BRIRs

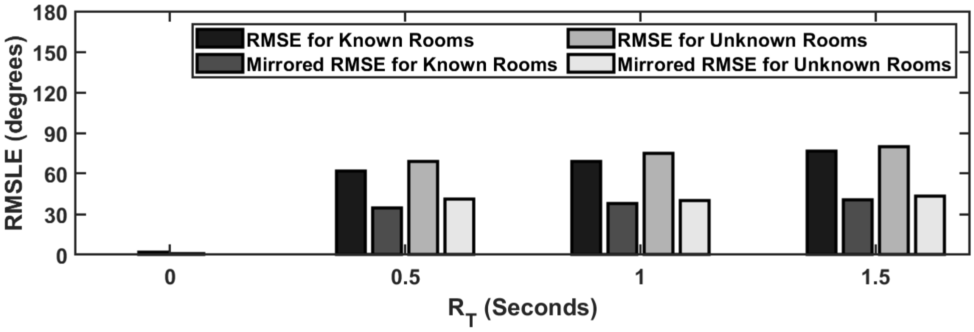

Figure 77: RMSLE for Different Rooms when trained using a large number of BRIRs

4.11.3 Discussion

It can be seen that as compared to the previous test with the network trained on five rooms, the difference between performance on known rooms and unknown rooms has been greatly reduced. This does point to increasing the generality of the training datasets as a possible solution to this problem, however it exposes another issue with this approach: a significant drop of performance from close to 100% performance among known rooms to around 20-40% classification accuracy.

This is likely due to the reduction in training data per room, a necessary reduction to avoid inflated training times or resources. This hypothesis is supported by the disproportionately high performance when $R_T = 0$, which is likely high due to all of the data for this reverb time being created with the same HRIR, as opposed to BRIR, effectively massively increasing the amount of training data available for that room.

4.12 Conclusion

In this chapter, the first CNN model has been presented, trained using data created in the ideal condition, and tested using testing sets containing data created under a variety of acoustic conditions. The testing method which will be reused throughout this work was also introduced.

Through demonstration and discussion of results, some conclusions have already become self-evident: Except under extreme conditions, noise that is both diffuse or directional does not present the challenge that it does in other audio related tasks such as ASR.

Conditions relating to generalisation, however, pose a significant problem. It was seen that a system trained on one HRTF will not generalise at all for other HRTFs. Though it is helpful to be aware of this, the rest of this work will not exercise effort in finding a solution to the mismatched HRTF condition.

From the results shown in this chapter from experiments on using reverberation it was seen that realistic reverberation environments cause an issue for BSSL systems which use CNN as the model is unable to localise in unknown rooms, that is rooms for which the BRIR has not been included in the training data.

With a view to this, models developed in this work need to be tested upon binaural audio which has been created using rooms which are not known to the trained model, so as to avoid an unrealistic view of how well the model would perform in real acoustic spaces.

A solution to this form of overfit has been proposed; simply increasing the number of known rooms. However due to finite computing resources the degree to which this will be able to be employed without sacrificing general performance is limited.

A common theme throughout the results of experiments in this chapter is that front-back confusion does not pose a significant challenge to the system in most circumstances, and that errors, when they do occur, tend to be more random. This in particular is notable, as often previous works often will constrain the azimuthal field to 180° either to avoid front-back confusion or because of the HRIRs databases available: this suggests, however, that this restriction is not necessary and wherever possible working with the full azimuthal field is encouraged.

Finally the success of using only a magnitude spectrum input for a CNN to localise needs to be pointed out. This will be addressed more fully in a later chapter, but when complexity needs to be reduced so as to improve training and testing times, this is an option which shall be reused throughout this work.

This chapter has introduced the framework by which deep-DoA estimation under a variety of conditions will be tested: training and testing datasets are created using speech and HRTFs and processed into a useful representation, a model is defined, and trained on the data, and then evaluated with the data set using a series of metrics: classification accuracy, front-back confusion, RMSLE, and RMSLE with mirroring.

5 Feature Representations

5.1 Magnitude Features

The convolutional neural networks (CNNs) in previous chapters have concerned the effect of changing the raw audio signal's content to achieve generalisation, but did not address the processing of the signal applied in creation of a time-frequency (TF)-Matrix.

While such literature exists for feature extraction for deep neural networks (DNNs) (O'Dwyer *et al.*, 2018), there is a gap in existing literature concerning the effect upon localisation accuracy of changing the type of feature a CNN is trained on.

To address this, this chapter shows the method and results of an attempt to benchmark the localisation accuracy for different feature representations.

5.1.1 Magnitude Features

Four different magnitude spectra are considered: mel-spectrograms, mel-frequency cepstrums (MFCs), gammatonegrams, and gammatone-frequency cepstrums (GFCs).

The mel-spectrogram presents a time-series signal as a matrix of values with dimensions representing time and frequency by means of a short-time fourier transform (STFT). However, as opposed to a linear-frequency spectrogram, the frequency is logarithmically scaled. This scaling more closely mimics human pitch perception, and allows for a higher amount of relevant information to be included in the spectrogram due to a greater emphasis on lower frequencies.

The MFC is a modification of the spectrogram, where a discrete cosine transform (DCT) is applied to the spectrogram to create something similar to but distinct from a spectrum; a cepstrum. Accordingly, it has retained the mel-frequency weighting of the mel-spectrogram. The mel-frequency cepstral coefficients (MFCCs), of which the MFC is composed, do not provide a particularly helpful visual representation of an audio signal, but have been widely employed as a feature for machine learning, particularly in speech recognition, where they are found to produce better results (Tiwari, 2010).

The gammatonegram calculates an intensity matrix similar to a mel-spectrogram, but having used a gammatone filterbank to discretise in the frequency dimension. The gammatone filterbank is a series of gammatone filters which have a frequency response reminiscent of the filtering that occurs within the cochlea. Due to this, they are used in models of the auditory system, as well as in features for CNNs in tasks in speech processing (Pour *et al.*, 2014).

The GFC creates a cepstrum in a manner identical to the MFC, but by applying the DCT to create a cepstrum from a gammatonegram rather than a mel-spectrogram.

The work presented in this chapter is also described in the publication 'Comparison of Performance in Binaural Sound Source Localisation using Convolutional Neural Networks for differing Feature Representations" (Reed-Jones *et al.*, 2023).

The models presented in this chapter were developed, trained and tested using the MATLAB Deep Learning Toolbox. Dataset generation was performed using MATLAB.

5.1.2 Binaural Dataset

A binaural dataset was created by combining speech utterances taken from the Librispeech corpus with head-related transfer functions (HRTFs) from, and binaural room transfer function (BRTF) made with, the CIPIC database KEMAR mannequin measurements.

For the training dataset, a total of 2000 speech samples were created by cutting 200 speech files from the corpus into 10 100ms sections. An additional 100 samples were created for the testing dataset by cutting another 100 speech files from the dataset into a single 100ms section. This is summarised in Table 7.

Table 7: Speech Files used in Magnitude Features experiment

	Librispeech Files	**No. 100mS Sections**	**Total**
Training	200	10	2000
Testing	100	1	100

BRIRs were generated through image-source simulation, for ten rooms with dimensions randomly generated in the range of 1-10m.

For each of these room dimensions, three sets of BRTFs were created with altered absorption coefficients of the boundaries, such that according to the Sabine equation (116) the room had reverb times of

$$T_{60} = [0.5, 1, 1.5] seconds \tag{119}$$

The Sabine equation is used over the Eyring formula in this case for the same reason previously outlined; by grouping data reverb times the aim is to find a general relationship between performance and reverb time, rather than a prediction of performance at specific reverb time.

The length of the binaural room impulse responses (BRIRs) was truncated by the T_{60} value, and the maximum reflection order was not truncated. This was carried out for the 50 azimuthal directions on the horizontal plane seen in the CIPIC database. Of these 30 resulting sets of BRTF, half were used for training, and half were used for testing.

In addition to this, the head-related impulse responses (HRIRs) taken directly from the CIPIC database were directly used, representing $T_{60} = 0$. In reality this is an approximation, as the HRIRs measured in the CIPIC database are not entirely anechoic. Noise mixtures were created through the method introduced in Chapter 4.5 of convolving BRIR of random directions. This was carried out for the training set with pink noise, and with a sample of room noise for the testing dataset.

The resulting noise mixtures were then scaled so as to achieve an signal-to-noise ratio (SNR) of:

$$dB(SNR) = [0, 12, 24, 36] \tag{120}$$

Each of the speech files was convolved with one of the BRIRs, and summed with one of the noise mixtures, to leave an equal distribution among reverb time and noise level. This is represented in Table 8.

Table 8: Distribution of training files per each source direction in magnitude features experiment

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	6.25%	6.25%	6.25%	6.25%
0.5	6.25%	6.25%	6.25%	6.25%
1	6.25%	6.25%	6.25%	6.25%
1.5	6.25%	6.25%	6.25%	6.25%

This was carried out for each of the 50 source directions, leading to a total of 1,600,000 training audio files (44.44 hours of audio), and 80,000 testing audio files (2.22 hours of audio).

5.1.3 TF Matrices

The training and testing audio datasets were then further processed into datasets containing four different types of magnitude TF matrices on which the CNN is to be tested and trained. As previously introduced, these four representations are:

- Mel-Spectrogram
- Mel-Frequency Cepstrum
- Gammatonegram
- Gammatone-Frequency Cepstrum

A definition of the mel-spectrogram has already been given in Equation (109). In this case, the spectrogram was created using window with a length of 465 samples, and an overlap length of 256 samples which led to the time domain being quantised to six samples. Meanwhile, the mel-filterbank was made up of 300 triangular bands in the range 100Hz to to 8kHz. This is summarised in Table 9.

Table 9: Mel-Spectrogram Parameters

Parameter	**Value**
Window Length	465
Overlap	256
N. frequency bands	300
Lowest Frequency	100 Hz
Highest Frequency	8000 Hz

The MFC is created through applying a DCT to the log of a mel-spectrogram. This differs from the cepstrum introduced in Chapter 2.2 as it does not use an inverse fourier transform (IFT). Rather than transformation to the quefrency domain, this instead finds discrete coefficients representing periodicity. Since this approach is also useful in the context of cepstral analysis, it is also considered a cepstrum.

$$\begin{bmatrix} MFCC_1 \\ MFCC_2 \\ ... \\ MFCC_N \end{bmatrix} = DCT\left\{ \begin{bmatrix} M_1 \\ M_2 \\ ... \\ M_N \end{bmatrix} \right\} \tag{121}$$

The gammatonegram is similar in conception to the mel-frequency spectrogram, differing only in that gammatone filters are not the same as mel filters. Gammatone filters are made from impulse responses derived by taking the product of a cosine and a gamma distribution term (Patterson *et al.*, 1992).

$$gam(t) = \frac{t^{n-1} \cos{(\omega_c t + \phi)}}{e^{2\pi b t}} \tag{122}$$

where n is the order, ϕ is phase shift, f is the centre frequency and b is the bandwidth. The frequencies and bandwidths of the filters are conventionally determined by the equivalent rectangular bandwidth (ERB) scale (Moore and Glasberg, 1983). Gammatone filterbanks are akin to how the cochlea decomposes frequency, which is the original application they were proposed for (Johannesma, 1972).

The creation of a gammatonegram is achieved similarly to the mel-spectrogram, by finding the product of the frequency domain responses of the binaural signal and the filters at each window. These filters values are also log scaled.

$$\begin{bmatrix} G_1 \\ G_2 \\ ... \\ G_N \end{bmatrix} = \log(\sum \left(|Y_{L,R}[k, \varphi_T]|. \begin{bmatrix} GAM_1[k] \\ GAM_2[k] \\ ... \\ GAM_N[k] \end{bmatrix} \right)) \tag{123}$$

The GFC is conceptually identical to MFC, but applying the same DCT to a gammatonegram rather than a mel-spectrogram.

$$\begin{bmatrix} GFCC_1 \\ GFCC_2 \\ ... \\ GFCC_N \end{bmatrix} = DCT \left\{ \begin{bmatrix} G_1 \\ G_2 \\ ... \\ G_N \end{bmatrix} \right\} \tag{124}$$

These four methods were applied to all training and testing datasets.

5.1.4 Model & Training

The same three layer CNN model was used to train on all four datasets, consisting of three convolutional layers with filters of increasing size. This is shown in Table 10. The architecture of this model is similar to that used in Chapter 4.1, differing only in the length of the output vector.

Table 10: CNN Architecture

Layer	Hyperparameters
Input Layer	(300, 6, 2)
Convolution Layer	(2,2),8
Batch Normalisation	
ReLu	
Max Pooling	2,2
Convolution Layer	(8,8),16
Batch Normalisation	
ReLu	
Max Pooling	2,2
Convolution Layer	(16,16),32
Batch Normalisation	
ReLu	
Dense Layer	50

stochastic gradient descent (SGD) training was employed using a learn rate of 0.001, and each model was trained for a period of 200 epochs.

5.1.5 Results

The CNNs were all then used to classify the corresponding testing datasets of TF-matrices. The classification rate, confusion rate, and root mean square localisation error (RMSLE) as defined in Chapter 4.2.3 are shown for each individual combination of RT60 and SNR. Averages are also plotted for both RT60 and SNR. The classification rate per reverb time and SNR for the four feature representations are shown in Tables 11 - 14, with averages being plotted against reverb time and signal to noise ratio in Figures 78 & 79.

The same arrangement of results is repeated for front-back confusion rate in Tables 15-17 and Figures 80 & 81, as well as for RMSLE in Tables 19 - 22 and Figures 82 & 83.

Classification Rate

Table 11: Classification rate for Mel Spectrogram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	69.3%	84.6%	93.7%	97.4%
0.5	24.7%	27.7%	29%	29.3%
1	23.3%	25.7%	25.9%	26.3%
1.5	21.2%	23.3%	22.4%	22.2%

Table 12: Classification rate for MFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	42.9%	54.4%	62.2%	66%
0.5	8.1%	9.2%	9.8%	9.5%
1	8.4%	8.9%	8.9%	9.0%
1.5	7.4%	8.4%	7.6%	7.4%

Table 13: Classification rate for Gammatonegram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	73.8%	87.3%	92.7%	94.3%
0.5	21.8%	24.7%	25.6%	25.5%
1	20.6%	23.3%	22.7%	22.7%
1.5	19.6%	21.2%	20.3%	20.4%

Table 14: Classification rate for GFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	41.6%	50.6%	55.4%	58%
0.5	8.3%	8.9%	8.9%	9.1%
1	8.3%	9.2%	9%	8.3%
1.5	8.1%	8.3%	8.7%	8.3%

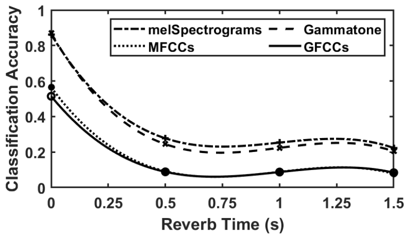

Figure 78: Classification Rate with respect to Reverb Time

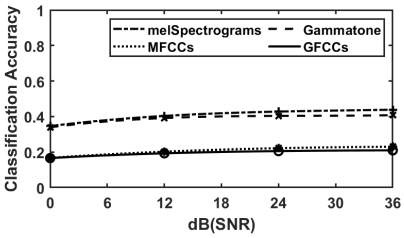

Figure 79: Classification Rate with respect to Signal to Noise Ratio

Confusion Rate

Table 15: Confusion Rate for Mel Spectrogram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	2.3%	0.8%	0.2%	0.06%
0.5	3.2%	3.3%	2.5%	2.7%
1	3.1%	2.9%	3%	2.9%
1.5	3.2%	3.1%	2.9%	3.2%

Table 16: Confusion Rate for MFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	4.6%	5%	4.5%	3.9%
0.5	5.7%	5.8%	5.8%	5.6%
1	5.7%	5.1%	5.4%	5.1%
1.5	5.6%	5.7%	5.4%	5.6%

Table 17: Confusion Rate for Gammatonegram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	2.5%	1.5%	0.8%	0.8%
0.5	3.4%	3.2%	3.3%	2.8%
1	3.4%	3.4%	3.1%	3.1%
1.5	3.2%	3.4%	3.4%	3.7%

Table 18: Confusion Rate for GFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	5.9%	6.1%	5.7%	5.6%
0.5	5.8%	5.5%	5.3%	5.7%
1	5.4%	5.6%	5.6%	5.5%
1.5	5.2%	5.8%	6.1%	5.3%

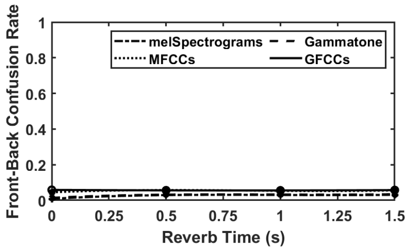

Figure 80: Confusion Rate with respect to Reverb Time

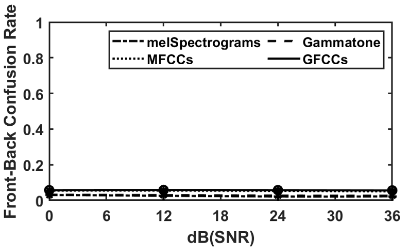

Figure 81: Classification Rate with respect to Signal to Noise Ratio

RMSLE

Table 19: RMSLE for Mel Spectrogram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	44.5°	27.7°	16°	7.76°
0.5	70.5°	67.2°	66.6°	66°
1	72.7°	69.3°	71.3°	68.1°
1.5	74.4°	73.7°	73.4°	74.8°

Table 20: RMSLE for MFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	60.5°	51.1°	44.1°	40.4°
0.5	87°	84.8°	84.2°	85.1°
1	87.6°	84.2°	84.6°	84.9°
1.5	87.6°	88.1°	88.7°	88.5°

Table 21: RMSLE for Gammatonegram

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	42.7°	27.3°	15.6%	14.1°
0.5	70.9°	69°	67.1°	66.3°
1	74.1°	70.7°	72.1°	71.3°
1.5	74.9°	75°	74.8°	74.9°

Table 22: RMSLE for MFC

	Signal to Noise Ratio (dB)			
RT60 (S)	0	12	24	36
0	60.3°	51°	45°	43.9°
0.5	86.4°	83.7°	84.7°	84.5°
1	87.9°	85.9°	87.1°	86.2°
1.5	87.9°	88.7°	88.6°	87.9°

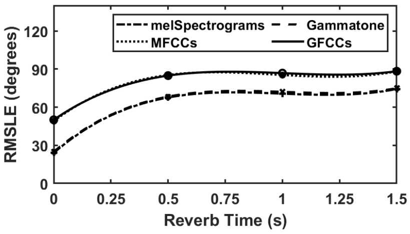

Figure 82: RMSLE with respect to Reverb Time

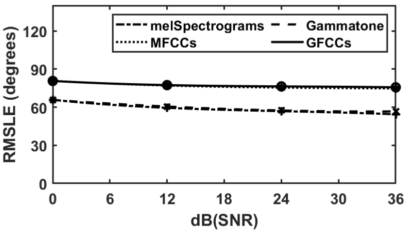

Figure 83: RMSLE with respect to Signal to Noise Ratio

5.1.6 Discussion

Looking particularly at the classification rate with respect to reverb time, shown in Figure 78, significantly better performance is seen in the anechoic condition. This is due to the overfit to observed rooms established in Chapter 4.6.3. It is unsurprising that that particular performance characteristic is retained, as the models in this chapter are similar to those in Chapter 4.6.3, and although the type is varied they remain trained just on log-magnitude matrices. This is deemed acceptable for this test, as only comparison between the feature types is of importance, but suggests what is already presumed that magnitude feature representations alone are not suitable for binaural sound source localisation (BSSL).

Among both classification rate, and RMSLE, a preference is seen towards using the log-magnitude values directly, rather than using cepstral coefficients. This confirms the apparent conventional wisdom of the field, as currently MFC and GFC for use by CNN are not typical.

Comparison of the two filterbank types, however, is not so revealing with the performance seen for mel-spectrogram and gammatonegram not having any significant difference, which is seen also for MFC and GFC. This suggests no apparent preference for gammatone decomposition, which is notable as this is very typical for deep binaural direction of arrival (DoA) estimation.

5.2 Phase and Temporal Features

Up to this point approaches taken have only made use of magnitude based feature representations. This is a naïve approach, as interaural time difference (ITD) is a salient cue for binaural sound localisation in humans. This chapter explores the use of time delay estimates and phase information in binaural DoA estimation with CNN, in a similar way to how the use of magnitude features was previously explored; through the processing of a single binaural dataset by a number of feature extraction methods, from which CNNs were trained.

5.2.1 Previous Work

There is a strong emphasis on using ITD for binaural sound source localisation with several likely reasons: low-frequency ITD is known to be the most salient cue in human binaural sound localisation (Wightman and Kistler, 1992) showing evolution has deemed this to be the most reliable cue. This idea is supported further by the supposition that early mammalian hearing systems relied solely upon interaural level difference (ILD) for sound localisation, and mechanisms for ITD based localisation were developed later (Grothe *et al.*, 2010).

ITD interpretation is identical to the popular time difference on arrival (TDOA) based estimation techniques used in microphone array based sound source localisation (SSL); Rascon and Meza's (2017) review on SSL in robotics suggests cross-correlation based TDOA estimation to be the most common approach taken.

While combining ITD and ILD is an early (Macpherson, 1991) and persistent theme in binaural sound localisation, solely ITD based analysis can still be used as a model of human localisation (Fazi and Hamdan, 2018; Simón *et al.*, 2018) preferred over ILD for its more general applicability (Pulkki and Hirvonen, 2005), though this reduces validity to only low frequencies.

Within machine learning based BSSL, ITD based upon the maxima of the interaural cross correlation function (IACCF) can be used as a feature (May *et al.*, 2011; Ma *et al.*, 2015a), or the entire IACCF can be used (Ma *et al.*, 2017). The more recent but dominant trend is training directly from the phase spectrum of the signal (Vecchiotti *et al.*, 2019), leaving the neural network to interpret the binaural cues.

Notably, it is not typical to bandlimit the signal when doing so. High frequency interaural phase difference (IPD) can become ambiguous due to spatial aliasing and it has been shown that at high frequencies humans cannot necessarily detect changes to a signal's relative phase (Wightman and Kistler, 1989), however it is also true that for high frequency signals the relative time of arrival of the amplitude envelope is still detectable. Neural networks may be able to make use of this cue.

The models presented in this sub-chapter were developed, trained and tested using the Tensorflow library for Python. Dataset generation, however, was still performed using Matlab.

5.2.2 Binaural Dataset

For this experiment, a dataset of binaural audio was created from which all of the subsequent datasets were trained. This dataset was created by combining BRIRs with speech and noise. However, there are some differences to the datasets generated for the magnitude features experiments detailed in Chapter 5.1.

This dataset makes use of the telecommunications and signal processing laboratory (TSP) corpus rather than TIMIT; which allowed for the use of a higher sample rate, resolving potential reductions in performance due to missing high frequency monaural cues.

A different set of BRIRs were generated for this experiment: 327 rectilinear room geometries were created by randomly generating three numbers scaled from 1-10 representing the length, width, and height dimensions of the room in metres. The absorption of the walls was made from uniform and randomly generated coefficients to create a room with an RT60 within the range 0.2 to 1.5 seconds.

For each of these rooms, the listener's head was placed at exactly the centre, and BRIRs were generated for the 50 source positions present in the CIPIC HRTF database.

These BRIRs were all truncated to a length of 0.5 seconds regardless of RT60 value. While this does have the effect of eliminating an accurate representation of the reverberation's decaying tail, it increases the total number of BRIRs which can be generated, hence the larger number of rooms present in this dataset compared to previous experiments. As the later part of a BRIR is diffuse in nature, it is presumed that the addition of noise achieves a similar effect on the performance of DoA estimation.

The corpus consists of approximately 1400 speech recordings. Of these, 1200 were used for creating training data, and 50 were used for creating validation data. For the training dataset, one second sections of each of these 1200 files was independently convolved with BRIRs of all 50 source directions, for a randomised room geometry of the first 300 rooms. The 50 validation files were convolved with the BRIRs of the remaining 27 room geometries.

Noise mixtures were created by the same method used in previous experiments of convolving a noise source with 1-10 BRIRs of the same room geometry at random source directions. For the training data, this noise source was pink noise, while for validation data, this was an ambient sound recording taken from freesound. These noise mixtures were randomly scaled to the range -24 to -60 dBFS.

5.2.3 Model & Training

In addition to use of a single binaural dataset, model training was controlled for all tests concerning preparation of phase and time based cues through minimum variability between tests. For these tests, a CNN with four convolutional layers with an equal number of equally sized kernels was used, as shown in Table 23. This was used as it was found that not including growth through the system was helpful for avoiding overfit.

Table 23: CNN Model used in Phase and Time cues testing

Layer	Hyperparameters
Input Shape	Variable
2D Convolution	(3,3), 16
Max Pooling	(2,2)
2D Convolution	(3,3), 16
Max Pooling	(2,2)
2D Convolution	(3,3), 16
Max Pooling	(2,2)
2D Convolution	(3,3), 16
Dropout	0.5
Dense	50

Notably, also deployed in this model is the use of a kernel regulariser, employed in order to help penalise overfitting. In this case, L2 regularisation was employed, with a regularisation value of $\lambda = 0.01$.

The input shape could alter between experiments due to the nature of the data. To train, the audio dataset was split into batches, each containing 13 seconds of audio, where each second of audio would eventually be split into 10 feature representations such that each of the batches in fact contained 130 data points.

The network was trained using an Adam optimiser (Kingma and Ba, 2014), with a learning rate of 1e^{-3} as well as well as a decay rate of 1e^{-3}.

5.2.4 Phase vs IPD

As previously established, a trend in BSSL is to train systems directly on the phase spectrum of some binaural audio. There are some different methods for preparing the phase signal for this, but one notable question is whether to train the system directly on the phase spectra of the two channels leaving the system to interpret the ITD for itself, or whether to take the IPD of the signal and train the system on this.

As per previous chapters, classification of two-dimensional matrices was focused on here. In this test, generating these matrices was again achieved through STFTs of the training and testing audio datasets. This was carried with the parameters shown in Table 24.

Table 24: STFT parameters used in Phase vs IPD test

Parameter	**Value**
Window Type	Hanning
Window Size	512
Hop Size	256

This produced matrices of the size $[257, 2]$ of complex values. These were then turned into Phase spectra of the same size:

$$\phi_{L,R}[m, \omega] = \angle\{X_{L,R}[m, \omega]\} \tag{125}$$

where $X[m, \omega]$ is the complex spectra. This was turned into IPD spectra by finding the signed angular distance between the two phase signals.

$$\mathrm{IPD}_{L,R} = \angle\left(e^{j(\phi_L - \phi_R)}\right) \tag{126}$$

It is important to note that the phase has not been unwrapped before finding the difference. The model was then trained and tested on the resulting matrices.

Results

The metrics classification accuracy, front-back confusion rate, RMSLE and mirrored-RMSLE for the phase and IPD matrix based representations are shown for each individual reverb time in the Tables 25 - 28, and the three metrics classification accuracy, front-back confusion and RMSLE are plotted against reverb time in Figures 84-86.

Table 25: Accuracy for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	43.1%	37.3%	34.04%
Phase	56%	24.2%	33.3%

Table 26: Confusion Rate for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	2.2%	1.2%	1.2%
Phase	0.1%	1.4%	1.4%

Table 27: RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	61.6°	65.9°	71°
Phase	30.3°	55.5°	58.2°

Table 28: Mirrored RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	45.2°	51.5°	58.4°
Phase	26.3°	40.5°	46.3°

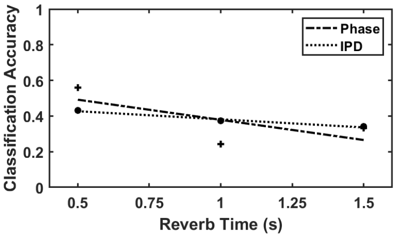

Figure 84: Classification Accuracy plotted against reverb time for IPD and Phase cues

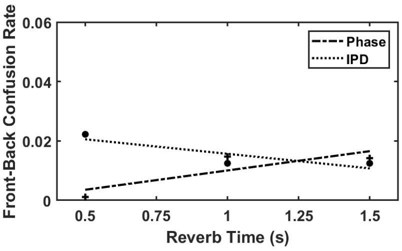

Figure 85: Front-back confusion rate plotted against reverb time for IPD and Phase cues

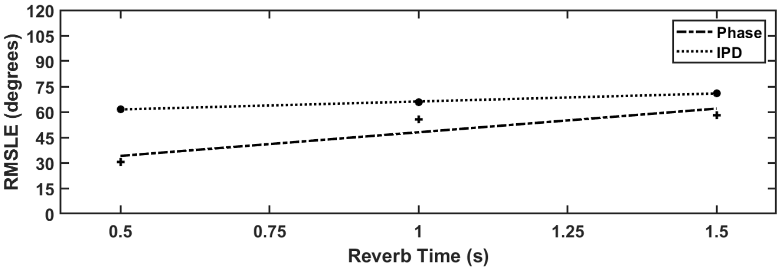

Figure 86: The RMSLE plotted against reverb time for IPD and Phase cues

Discussion

The results in Chapter 5.2.4 show the difference in performance between CNNs trained on 3D phase matrices and 2D IPD matrices made directly from the complex values with no other processing.

It can be seen in Figure 86 that according to the reported RMSLE, using 2D phase matrices is preferable to finding the IPD, with improved performance being seeing at all three reverb times. It should be noted however that in Figure 86 the gradient of the curve representing phase is steeper, suggesting that the phase representation may be less robust to reverberation.

5.2.5 Unwrapped Phase

Phase can be ambiguous due to its value always being constrained to the range $0 \leq \phi \leq 2\pi$, leading to phase ambiguity as delays beyond this range are wrapped back into the original range; this is particularly prescient for higher frequencies of the signal.

One possible solution to this is to use an unwrapping algorithm. Phase unwrapping algorithms detect phase discontinuities larger than a certain threshold, and adjust the signal accordingly

$$\begin{aligned}\Delta\phi_{\text{corrected}}[n] &= \begin{cases} \Delta\phi[\omega] - 2\pi & \text{if } \Delta\phi[\omega] > \pi \\ \Delta\phi[\omega] + 2\pi & \text{if } \Delta\phi[\omega] < -\pi \\ \Delta\phi[\omega] & \text{otherwise} \end{cases} \\ &\mathit{where} \\ &\Delta\phi[\omega] = \phi[\omega] - \phi[\omega - 1]\end{aligned} \tag{127}$$

A threshold of π is used here, but this is not necessary and can be altered depending on result. This can then be turned into an unwrapped signal through a cumulative sum.

$$\phi_{\text{unwrapped}}[\omega] = \phi[0] + \sum_{n=1}^{\omega} \Delta\phi_{\text{corrected}}[n] \tag{128}$$

This phase unwrapping is applied to each of the frequency axes of the phase spectra found in Equation (125), to find a new spectra of unwrapped phase values $\phi_{unwrapped,\{L,R\}}[m,\omega]$.

This is also used to make a new IPD spectra:

$$IPD_{unwrapped}[m,\omega] = \phi_{unwrapped,L}[m,\omega] - \phi_{unwrapped,R}[m,\omega] \tag{129}$$

These two methods were applied to create two new sets of training and testing datasets, upon which the neural network was trained and tested.

Results

From testing the models with the evaluation sets, the metrics for accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 29-32, while classification accuracy is plotted against reverb time in Figure 87.

Table 29: Accuracy for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	1.4%	1.7%	1.3%
Phase	4.5%	3.7%	3%

Table 30: Confusion Rate for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	2%	2.7%	3.4%
Phase	4.5%	3.7%	3%

Table 31: RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	101.7°	99°	101.3°
Phase	87°	83.6°	83.4°

Table 32: Mirrored RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	73.9°	99°	101.3°
Phase	47.1°	51.4°	52.2°

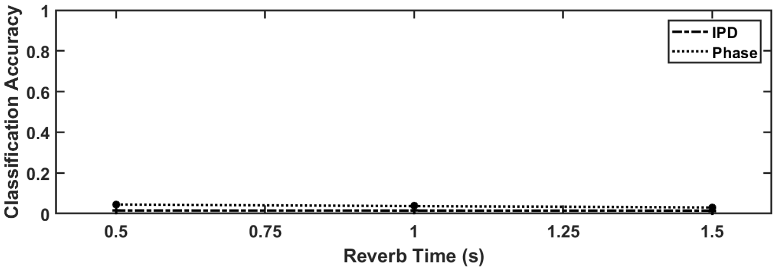

Figure 87: Classification Accuracy plotted against time for evaluation of unwrapped phase and IPD feature representations

Discussion

The results presented in Chapter 5.2.5 show performance of CNN trained and tested on matrices of unwrapped phase, and IPD based on unwrapped phase.

Tables 29-32 show very poor performance, with phase achieving only slightly better performance than random guessing, and IPD achieving below even that. This strongly suggests that unwrapping alone, at least for this style of CNN, greatly reduces the chances of the neural network being able to converge.

One possible reason for this is the much larger range of values seen in unwrapped phase, and that these ranges greatly vary between datapoints. Such data can lead to exploding gradients, leading to the system not converging during training.

5.2.6 Unwrapped Phase with Normalisation

Given the hypothesis that unwrapped phase may lead to exploding gradients due to large ranges of values, one possible solution may be normalise the values. To test this, the same training and testing dataset as in Chapter 5.2.5 was used, but with the difference that for each data point was normalised to the range 0-1.

Doing this leads to an invalid representation since it hides information relating to the absolute phase difference between the signals, which is important if such a system is expecting to be able to interpret ITD. If, however, the shape of the unwrapped phase curve is of significance, this approach is likely to yield better results than those reported in Chapter 5.2.5, which is of significance and so still tested.

Results

From evaluating the models with the testing sets, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 33-36, while classification accuracy is plotted against reverb time in Figure 88.

Table 33: Accuracy for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	4.8%	3.6%	2.7%
Phase	4%	3.5%	3.2%

Table 34: Confusion Rate for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	9.9%	8.2%	8.1%
Phase	3.8%	5.8%	7.2%

Table 35: RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	85.7°	89.6°	89.3°
Phase	83.5°	83.6°	83.3°

Table 36: Mirrored RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	57.8°	62°	62.7°
Phase	52.3°	56.4°	56.4°

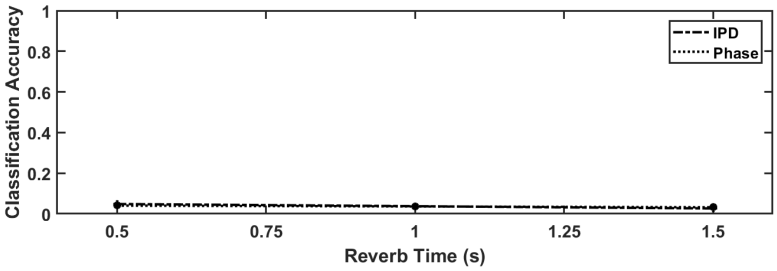

Figure 88: Classification Accuracy plotted against time for evaluation of unwrapped phase and IPD with normalisation feature representations

Discussion

The results in Chapter 5.2.6 show the performance of CNN trained on unwrapped phase matrices which have been normalised to 0-1 at every data point.

Some degree of improvement is seen in these results compared with the unwrapped phase with no normalisation. The degree to which this is significant is limited, as it is still significantly worse than directly using the still-wrapped phase. It may be that the more manageable value range leads to a system which can more easily converge, however this form of feature representation is inherently compromised and not worth pursuing further.

5.2.7 Sine Scaled Unwrapped Phase

Another way in which to re-scale input data to a more restricted range is to use a nonlinear scaling function; in this case this is achieved simply by using a sine function.

It is first aimed to rescale the data such that it falls within half a revolution of a sinusoid, specifically in the region $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. For the case of IPD, reasonable high limits can be found as the IPD of the signal at the extreme lateral positions $\varphi = -90°$ and $\varphi = 90°$. This value is found from the HRTF directly, and used to rescale the signals to a range of $-\frac{\pi}{2}$ to $\frac{\pi}{2}$.

$$\phi_{rescaled} = \frac{\pi}{2} \cdot \frac{\phi}{\phi_{Max}} \tag{130}$$

This is carried out for both for 2D phase and IPD matrices. After scaling, a sine function is applied

$$X = \sin(\phi_{rescaled}) \tag{131}$$

doing this rescales the values nonlinearly in a way which may or may not be beneficial, but significantly also means that values which greatly exceed the proposed scaling factor in either positive or negative direction get rescaled back into the range 0-1. These values may lose meaning; however they are less likely to cause issues in the training process than excessively large values.

Results

From evaluating the models with the testing sets for sine-scaled unwrapped phase and IPD representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 37-40, while RMSLE is shown plotted against reverb time in Figure 89.

Table 37: Accuracy for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	48.3%	27.2%	35.3%
Phase	65.2%	18.6%	40.7%

Table 38: Confusion Rate for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	0.8%	3.3%	2.9%
Phase	0%	0.8%	0.2%

Table 39: RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	50.8°	56.5°	53.9°
Phase	55.6°	75.8°	79.8°

Table 40: Mirrored RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	28.4°	36.6°	40.6°
Phase	49°	62.6°	68.7°

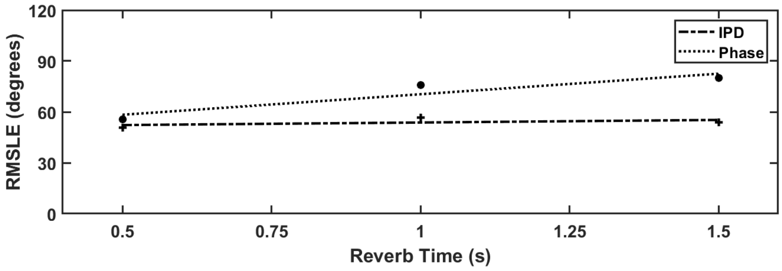

Figure 89: RMSLE plotted against time for evaluation of unwrapped phase and IPD with sine-scaling feature representations

Discussion

The results presented in Chapter 5.2.7 show the performance of CNN trained and tested with unwrapped phase and IPD matrices, rescaled by the maximum IPD, and having had a sine function applied.

The results here are an improvement of the previous form of rescaling the data, normalisation. However, while the results are now approaching that seen by unprocessed phase, they are still outperformed.

Recalling that this method rests on an assumption which is valid for IPD, but not for 2D phase matrices, it is unsurprising that a larger difference between phase and IPD is seen here. It is unexpected, however, that the 2D phase matrices are able to achieve a higher level of classification accuracy than IPD at some reverb times. This is not seen in the plot of RMSLE however, the more valid performance metric of the two.

5.2.8 Sine Scaled Phase

One of the effects of the sinusoid function is a scaling of the values, in a way that a neural network may find beneficial. It is checked then, whether applying a sine function directly to the still-wrapped phase can cause a relative improvement in performance.

$$X_{Phase} = \sin(\phi) \tag{132}$$

A notable property of Equation (132) is its similarity to the imaginary part of the original complex matrix X:

$$\text{Im}\{X\} = |X|\sin(\phi) \tag{133}$$

The removal of the magnitude term $|X|$ means that Equation (132) represents the normalised imaginary part of X. For the IPD, the sine function is applied prior to the finding of the difference.

$$X_{IPD} = \sin(\phi_L) - \sin(\phi_R) \tag{134}$$

Results

From evaluating the models with the testing sets for sine-scaled phase and IPD representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 37-40, while RMSLE is shown plotted against reverb time in Figure 89.

Table 41: Accuracy for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	56.3%	27.2%	34.9%
Phase	66.9%	29.9%	44.2%

Table 42: Confusion Rate for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	1.9%	0.6%	1.2%
Phase	0%	0.3%	0%

Table 43: RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	44.2°	46.2°	55.3°
Phase	45.4°	70.3°	74.9°

Table 44: Mirrored RMSLE for IPD and Phase

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
IPD	28.3°	36.3°	41.3°
Phase	38.5°	54.9°	66.1°

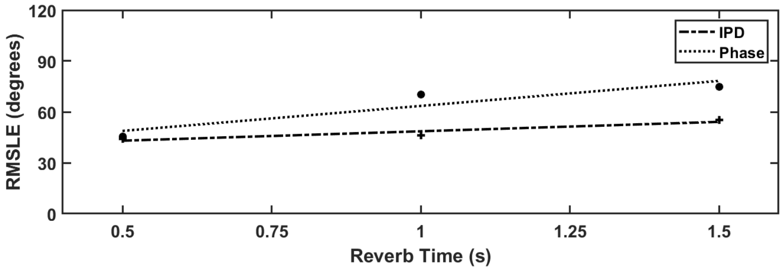

Figure 90: RMSLE plotted against time for evaluation of phase and IPD with sine-scaling feature representations

Discussion

The results presented in Chapter 5.2.8 show the performance of CNN trained and tested on still-wrapped phase and IPD matrices, modified by a sine function.

The performance seen in this test, while preferable to some previous methods, does not improve over directly using the phase. This is conclusive: rescaling of the phase matrices is not justified.

5.2.9 Downsampled Frequency-Banded IACCFs

Introduced in Chapter 2.2.1 was the idea of using cross correlation to estimate the ITD of a binaural signal. A reasonable assumption, then, is that cross-correlation could be used as a cue for binaural DoA estimation.

This has commonly been seen in conjunction with multi-layer perceptron (MLP) architectures, in which the cross-correlation is directly input to fully-connected layers (Ma *et al.*, 2015b; O'Dwyer and Boland, 2022). Its use in conjunction with 2D convolutional layers however has not been seen, with IPD typically being used instead.

Due to this, it is tested whether it is possible to prepare cross correlation signals such that they are able to outperform IPD as a cue.

The first method which was tested was to calculate only one cross-correlation function per audio file, extract the part of the signal relevant to ITD, and split into frequency bands through a filterbank.

The 100ms audio file was firstly truncated to the first 10ms; the reason for doing so is using the section of audio is overly computationally expensive: the number of operations is proportional to the product of the length of the signal and the number of lags calculated. The number of lags is truncated based upon an assumption presented in the following paragraph, but to further aid with data generation times the signal length was also reduced. Moving operations into the frequency domain would have eliminated this signal length factor, however when doing so it is not possible to truncate lags to the desired range at the time of the cross-correlation operation.

The cross correlation of the signal was then taken, to give the curve R_{LR}:

$$R_{lr}[m] = b_l[n] \star b_r[n] \tag{135}$$

Where m is the discrete lag, and l and r refer to left and right. The convention of capitalising these indices is broken here to avoid confusion with R, the cross-correlation. Relative to the length of the signal, 10ms, the length of maximum possible ITDs are small. This means that only part of the signal $R_{lr}[m]$ is relevant to ITD.

Based upon an estimate that maximum ITD of the KEMAR head simulator is less than $800\mu S$, an assumption made based on measurements presented as part of the SADIE II database (Armstrong *et al.*, 2018), the range of lags for which $R_{lr}[m]$ relevant is given as:

$$-0.0008 < \tau < 0.0008 \tag{136}$$

or, equivalently:

$$-\frac{Fs}{1250} < m < \frac{Fs}{1250} \tag{137}$$

For the sampling rate of 44100 used in binaural dataset, this results in $R_lr[n]$ being truncated to a total length of 71 samples.

71 samples represents a much greater width than the 16 samples caused by the windowing in the STFT of previous tests. While performance with the full 71 samples was later tested, firstly the performance with a reduced 16 samples was tested, created through a downsampling of the signal; an anti-aliasing filter is present in this downsampling process.

Now the resulting downsampled signal was split into frequency bands through use of a gammatone filterbank (GFB). The GFB was created with 257 filters, representing a range of 20Hz to 20kHz.

Cross correlation, and finite impulse response (FIR) filtering are linear time-invariant (LTI) operations, and accordingly the order of operations yields equivalent results.

With this, a matrix of the dimensions [16, 257] has been created representing time lag by frequency. This is used as the input to the CNN.

Results

From evaluating the models with the testing sets for downsampled frequency-banded IACCF representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in entirely within Table 45, while RMSLE is shown plotted against reverb time in Figure 91.

Table 45: Evaluation Results for Downsampled Frequency-Banded IACCF Representations

	Reverb Time		
Metric	**0.5**	**1**	**1.5**
Accuracy	10.2%	8%	6.3%
Confusion Rate	9%	9.4%	7.9%
RMSLE	73.1°	80.1°	79.4°
Mirrored RMSLE	40.5°	47.5°	48.9°

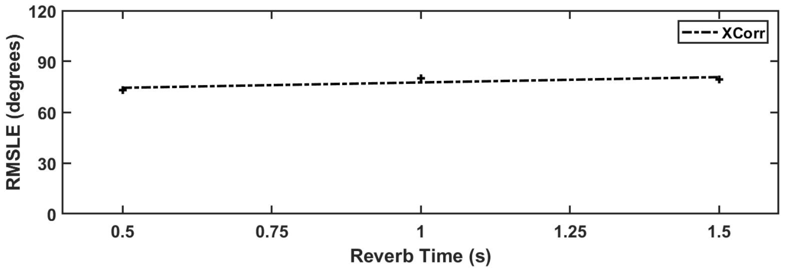

Figure 91: RMSLE plotted against time for evaluation of downsampled frequency-banded IACCF representations

Discussion

The results presented in Chapter 5.2.9 show the performance of CNN trained and tested on matrices made from frequency banded and downsampled IACCFs.

The significant result found here, is that the performance of this system is much lower than those trained with phase or IPD matrices. This supports the trend of moving away from using IACCFs with the introduction of CNN into the field; and suggests that this type of representation is not being overlooked.

5.2.10 Wideband Frequency-Banded IACCFs

Following from the test performed in Chapter 5.2.9, a similar test is performed but without the downsampling stage, leaving matrices of the size [71,257]. As the size of the matrix has by necessity increased to achieve this, the total size of the training dataset was halved; this is both as a penalty to allow for better comparison between tests, but also because the number of datapoints in the training dataset had been determined originally based on the size of each.

Results

From evaluating the models with the testing sets for wideband frequency-banded IACCF representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in entirely within Table 46, while RMSLE is shown plotted against reverb time in Figure 92

Table 46: Evaluation Results for Wideband Frequency-Banded IACCF Representations

	Reverb Time		
Metric	**0.5**	**1**	**1.5**
Accuracy	12.6%	7.3%	9.4%
Confusion Rate	11.6%	12.2%	8.8%
RMSLE	66.6°	69.5°	69.4°
Mirrored RMSLE	39.5°	43.7°	46°

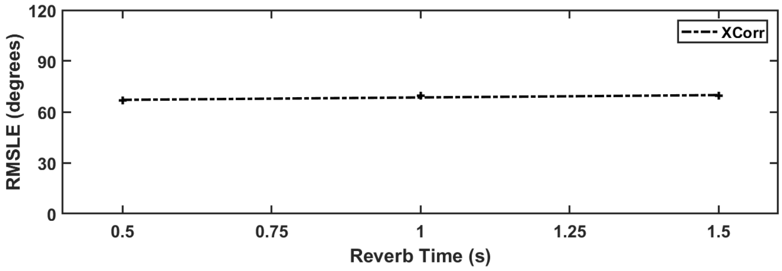

Figure 92: Classification Accuracy plotted against time for evaluation of wideband frequency-banded IACCF representations

Discussion

The results presented in Chapter 5.2.10 show the performance of CNN trained and tested on frequency-banded IACCF matrices, but without any downsampling.

Despite the much larger sized matrices, the results are only slightly improved over the downsampled matrices. This shows that while this downsampling did create some loss in accuracy, it is not to blame for the relatively poor performance of IACCF derived matrices for binaural DoA estimation with CNN.

5.2.11 IACCF derived TF-Matrix

An alternative method was tested, being to create a TF-Matrix, akin to a spectrogram, but by finding individual matrix elements through the ITD prediction method derived from IACCF.

In order to achieve this, prior to the cross correlation being made, the signal was windowed with a hanning window. The window had a size of 512 samples, and a hopsize of 256 was used, equivalent to the window used in the creation of STFTs in phase tests.

For each of the resulting windowed sections of audio, exactly the same method as Chapter 5.2.9 was employed to create the IACCF $R_{lr}[m]$, and it was also reduced to the samples relevant for binaural DoA estimation, but then the additional stage of finding the argmax of the IACCFs was employed.

$$ITD = \arg\max(R_{lr}[m]) \tag{138}$$

This was carried out for every window, at every frequency band, and the resulting ITD estimates were collated in a matrix of the size [16, 257].

Results

From evaluating the models with the testing sets for IACCF derived TF-matrix representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in entirely within Table 47, while RMSLE is shown plotted against reverb time in Figure93

Table 47: Evaluation Results for IACCF derived TF-Matrix Representations

	Reverb Time		
Metric	**0.5**	**1**	**1.5**
Accuracy	1.9%	1.7%	2.1%
Confusion Rate	5.4%	4.1%	4.1%
RMSLE	103.4°	105.3°	107.5°
Mirrored RMSLE	71.8°	75.5°	76.7°

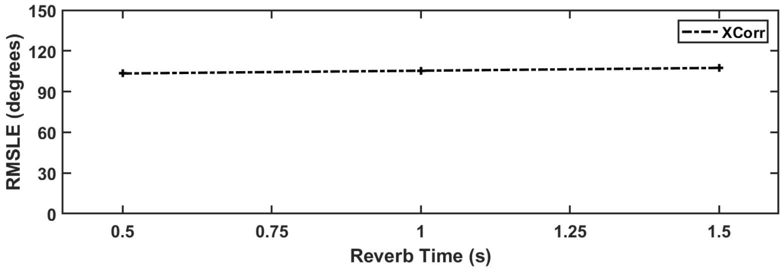

Figure 93: RMSLE plotted against time for evaluation of IACCF derived TF-Matrix representations

Discussion

The results in Chapter 5.2.11 show the performance of CNN trained and tested on matrices derived by maximising IACCF curves for different frequency bands at different windows. It can be seen that this is a negative result; the accuracy is $1/N$ and as such the system is randomly guessing. This is proof of the ineffectiveness of this approach. Based on this, it is supposed that taking the maximum of the correlation signal removes the information important for robust sound localisation.

5.2.12 Low-Frequency IACCF derived TF-Matrix

Duplex theory suggests that ITD is only significant at lower frequencies. Based on this, it is hypothesised that one way to improve the results of the TF-Matrix would be to reduce the range of frequencies.

New TF-matrices are made by reducing the maximum cutoff frequency of the filterbank to 1500Hz, however the number of filters is maintained to give a much denser sampling over the new smaller range.

Results

From evaluating the models with the testing sets for low-passed IACCF derived TF-matrix representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in entirely within Table 48, while RMSLE are shown plotted against reverb time in Figure 94

Table 48: Evaluation Results for low-passed IACCF derived TF-Matrix Representations

	Reverb Time		
Metric	**0.5**	**1**	**1.5**
Accuracy	1.9%	1.8%	1.4%
Confusion Rate	4.9%	4.9%	4.6%
RMSLE	104.5°	105.3°	106°
Mirrored RMSLE	72.2°	73.2°	74.8°

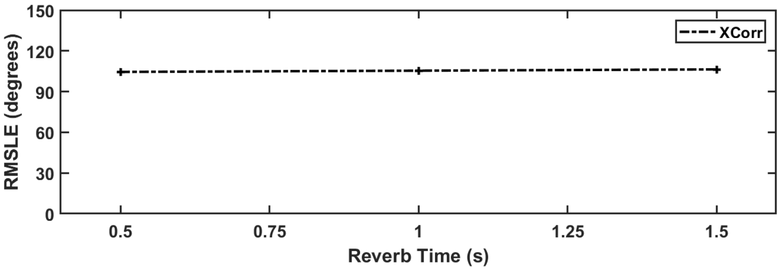

Figure 94: RMSLE plotted against time for evaluation of low-passed IACCF derived TF-Matrix representations

Discussion

The results in Chapter 5.2.12 show the performance of CNN trained and tested with frequency band limited matrices made of frequency banded IACCFs. It can be seen that this creates no improvement over the results reported in Chapter 5.2.11. Given the performance of this, and all other IACCF based matrices, it is concluded that IACCF is not suitable for use with 2D convolution layers as compared with training the model with directly with the phase.

5.2.13 Comparison with Magnitude

It is of interest to determine whether localising with phase gives better or worse performance than localising with magnitude of a signal; however direct comparison between the results reported in this Chapter with results in Chapter 5.1 is not possible owing to the differences in dataset and training conditions. To this end the binaural dataset was also turned into log-magnitude and ILDs matrices.

The log-magnitude matrix was created by taking the logarithm of the absolute of the complex matrix

$$\mathrm{Mag}_{L,R}[m,\omega] = \log_{10}(|X_{L,R}[m,\omega]|) \tag{139}$$

Meanwhile, the ILD matrix was found by taking the difference of the logarithms of the magnitudes of the two channels.

$$\mathrm{ILD}_{L,R}[m,\omega] = \log(|X_L[m,\omega]|) - \log(|X_R[m,\omega]|) \tag{140}$$

The resulting training matrices were used to train the same CNN architecture, which was then tested on the generated testing matrices.

Results

From evaluating the models with the testing sets for ILD and magnitude representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 49-52, while RMSLE is shown plotted against reverb time in Figure 95.

Table 49: Accuracy for ILD and Mag.

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
ILD	52.1%	34.9%	57%
Magnitude	56.4%	45.8%	50.5%

Table 50: Confusion Rate for ILD and Mag.

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
ILD	0.2%	3.6%	0%
Magnitude	0.2%	3%	1.4%

Table 51: RMSLE for ILD and Mag.

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
ILD	39.6°	30.4°	35.5°
Magnitude	39.6°	30.5°	39.4°

Table 52: Mirrored RMSLE for ILD and Mag.

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
ILD	26.9°	23°	27.4°
Magnitude	24.4°	19.2°	30.5°

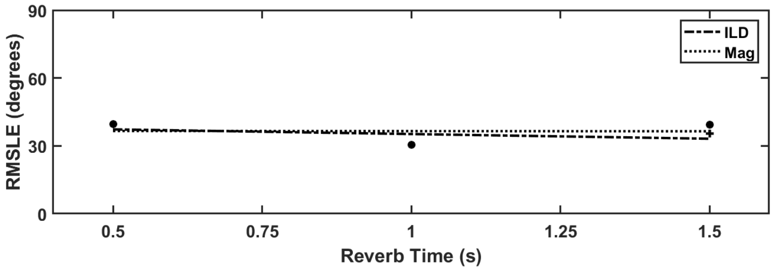

Figure 95: RMSLE plotted against time for evaluation of ILD and Magnitude representations

Discussion

The results in Chapter 5.2.13 show the performance of CNN trained and tested with magnitude and ILD matrices of the same STFT complex values used for phase and ILD matrices, for the purposes of comparison.

It can be seen that performance is similar to that of using phase, but with a slightly lower RMSLE being reported. This is a significant result, as typically ITD is preferred over ILD in binaural sound localisation systems; however it seems in the case of CNN based DoA estimation, the magnitude of HRTFs seems to be the slightly stronger cue.

5.2.14 Real and Imaginary Part

A similar approach is after performing Fourier transform on the signal to convert into the frequency domain, rather than finding the magnitude and phase of the complex matrices instead taking the real and imaginary parts. Each of the real and imaginary parts contain information that relates both to the magnitude and phase of the systems: this test will show whether this can still meaningfully interpreted by a CNN. The two matrices are simplistically defined below.

$$\begin{gathered}\mathrm{Re}(X_{L,R}[m,\omega]) \\ \mathrm{Im}(X_{L,R}[m,\omega])\end{gathered} \tag{141}$$

Results

From evaluating the models with the testing sets for real and imaginary part representations, the metrics accuracy, front-back confusion rate, RMSLE and Mirrored-RMSLE are shown in Tables 53-56, while RMSLE is shown plotted against reverb time in Figure 96.

Table 53: Accuracy for Real and Imaginary

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
Real	45.0%	42.8%	38.2%
Imaginary	43.6%	42.1%	38.4%

Table 54: Confusion Rate for Real and Imaginary

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
Real	4.34%	3.16%	2.42%
Imaginary	4.68%	4.68%	2.98%

Table 55: RMSLE for Real and Imaginary

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
Real	49.5°	51.4°	51.0°
Imaginary	49.8°	51.8°	51.5°

Table 56: Mirrored RMSLE for Real and Imaginary

	Reverb Time		
Cue	**0.5**	**1**	**1.5**
Real	40.9°	43.1°	44.5°
Imaginary	40.9°	43.3°	45.0°

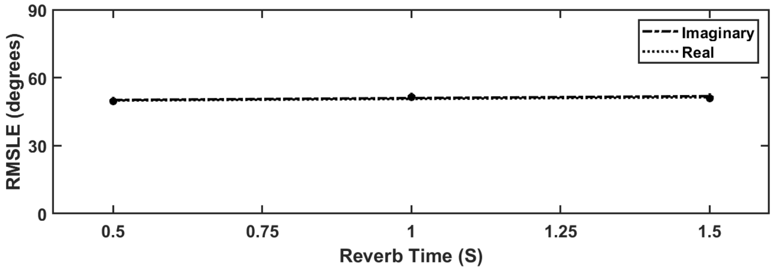

Figure 96: RMSLE plotted against time for evaluation of real and imaginary parte representations

Discussion

From all the sets of results presented in Tables 53 - 56 and Figure 96 it can be seen that when training and testing with the real or imaginary parts of the complex matrices, there is no significant difference between using either the real or imaginary part of the complex binaural signal for classification. This is unsurprising, as neither of these parts of the signal hold special significance with regard to localisation cues; the binaural and monaural cues are spread among both the real and imaginary parts.

Comparing these results with other results with those presented in Figure 95 it can be seen that performance is quite comparable but slightly worse than using the magnitude of the complex signal. Comparing with Figure 86 reveals a general improvement over localising with phase, with the exception of at low reverb times owing to the much steeper gradient in the reverb performance.

These results certainly do not negate the idea of using real and imaginary parts of the complex signal in replacement of phase and magnitude as performance is similar, however, there similarly is not a strong justification that they would be preferred.

5.3 Conclusion

This chapter addressed the use of different feature representations for the task of BSSL using CNNs. Chapter 5.1 addressed the relative performance of difference magnitude matrix feature representations of binaural audio for deep binaural DoA estimation using CNN. This was achieved by comparing four types of feature representation; mel-spectrogram, gammatonegram, MFC and GFC.

Comparison of the results shows a preference for the spectrogram representations of sound, but at the same time no preference can be well established between the different approaches to filterbank coefficients, despite the emphasis on GFB in literature due to its similarity to human cognition. Chapter 5.2 addressed the performance of CNN when trained and tested with a variety of phase or IACCF based representations of binaural signals. Additionally, these were compared to the time-frequency magnitude of the HRTF.

The most significant finding, is that directly using unprocessed phase is the optimal strategy tested, with the other proposed processing not providing improvement.

Use of matrices derived from IACCF in the context of DoA estimation by CNN is dismissed owing to poor performance.

There is evidence to suggest that when training CNN on phase and magnitude of the Fourier transform of a binaural signal, it is in fact the magnitude which is the more dominant cue. Additionally, it has been shown that similar performance can be achieved when using the real or imaginary parts of the signal, in lieu of phase or magnitude.

6 Deep Learning Architectures

Chapters 4 & 5 have introduced good practice with regard to generating datasets for the task of binaural direction of arrival (DoA) estimation. It is prudent to accompany this with insight into selection of model for this task.

This chapter contains three experiments on the theme of deep learning model architectures for binaural DoA estimation. Chapter 6.1 benchmarks the performance of 1D and 2D convolutional layers in convolutional neural networks (CNNs) for binaural sound source localisation (BSSL), Chapter 6.2 introduces a rationale for using convolutional recurrent neural networks (CRNNs) in BSSL, Chapter 6.3 compares the performance of CRNNs relative to CNNs, Chapter 6.4 compares the use of different types of recurrent layer in CRNN models for BSSL, and finally Chapter 6.5 makes overall conclusions.

6.1 1D vs 2D Convolution

Up to this point, the CNNs presented in this work have been built with 2D convolutional layers, and as such expect matrices upon which to train and test.

No obvious justification for doing so has yet been presented. While this extra dimension could present an extra cue, such as time (as in the case of short-time fourier transform (STFT) based input), CNNs are not necessarily capable of interpreting this, due to their quality of translation invariance.

While 2D Convolutional layers are not exceptional in BSSL (Xu *et al.*, 2019; Pang *et al.*, 2019), 1D convolutional layers which learn directly from the magnitude and phase of the system are common also (Vecchiotti *et al.*, 2019). The reduction of dimensionality allows for a greater number of samples, and therefore more detail, in the existing dimension which could be of greater benefit. Here, a comparison of these two approaches is undertaken to understand the relative advantage of each approach. This is achieved by the creation of a binaural dataset with reverberant conditions, for speech on the horizontal plane, and the subsequent training and testing of the two differing systems on the data.

The models presented in this chapter were developed, trained and tested using the Tensorflow library for Python, while the dataset generation was performed in

MATLAB.

6.1.1 Binaural Dataset

The binaural dataset described in Chapter 5.2.2 was also used in this experiment. This consisted of binaural room impulse responses (BRIRs) synthesised for 327 different rectilinear rooms, which were then combined with speech from the telecommunications and signal processing laboratory (TSP) Corpus, as well as summed with additive noise which itself was also binauralised, to create 60,000 second-long training files, and another 5000 second-long files used for testing.

6.1.2 Feature Representations

1-Dimensional

From each of the audio files described in Chapter 6.1.1, 10 phase and magnitude spectra were extracted. This was achieved by reducing the full signal to ten segments containing 4410 samples, and for each of these the FFT was taken to give the vector of complex frequency domain numbers.

This was reduced to the first 2206 samples to represent frequencies of only up to Nyquist frequency. From this, the logarithm of the absolute of the complex vector was taken to give two vectors representing magnitude, and the angle of the complex vector was taken to give two channels of phase.

2-Dimensional

The two-dimensional matrix was created by STFT. The sample ten sub-segments of audio were further windowed by a moving window determining the time-dimension of the spectrogram. This window had a length and hop size chosen such that the window length, and therefore number of frequency domain samples, is one tenth of the one dimension vector, and the hop size was chosen so as to create an x-dimension of 10.

Table 57: STFT Parameters

Parameter	**Value**
Window Length	440
Hop Size	400

Doing this closely approximated the total number of samples in the matrix to the vector, with the matrix containing a total of 2210 samples.

As in the case of the vector, a magnitude matrix was created by taking the log of the absolute of the time-frequency (TF)-Matrix separately for the two channels, and a phase matrix was created by taking the angle of the complex matrix.

6.1.3 Models and Training

The two models were designed to match each other; although the fundamental change in architecture complicates this. For the two dimensional matrices, 2D Convolution layers with $(3, 3)$ sized kernels, and max pooling layers with $(2, 2)$ kernels were employed as seen in Table 58. This is similar in design to the model used in Chapter 5.2, with the flat architecture also being employed to avoid overfit.

Table 58: Model for localisation of with 2D Matrices

Phase Branch		Magnitude Branch	
Layer	**Hyperparameters**	**Layer**	**Hyperparameters**
Input Layer	(221, 10, 2)	Input	(221, 10, 2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Dropout		0.5	
Dense, Softmax		72	

While for the 1D inputs, these 2D layers were replaced with 1D convolution layers with 3 sample kernel size, and 1D max pooling with 2 sample kernel size, as seen in in Table 59.

Table 59: Model for localisation of with 1D Vectors

Phase Branch		**Magnitude Branch**	
Layer	**Hyperparameters**	**Layer**	**Hyperparameters**
Input	(2207, 2)	Input	(2207, 2)
1D Convolution	3, 16	1D Convolution	3, 16
Batch Normalisation		Batch Normalisation	
Max Pooling	2	Max Pooling	2
1D Convolution	3, 16	1D Convolution	3, 16
Batch Normalisation		Batch Normalisation	
Max Pooling	2	Max Pooling	2
1D Convolution	3, 16	1D Convolution	3, 16
Batch Normalisation		Batch Normalisation	
Max Pooling	2	Max Pooling	2
1D Convolution	3, 16	1D Convolution	3, 16
Dropout		0.5	
Dense, Softmax		72	

The two models were both trained using an Adam optimiser with identical parameters, as shown in Table 60.

Table 60: Adam Optimiser Parameters for 1D vs 2D test

Parameter	**Value**
Rate	1e-3
Decay	1e-3

The models were trained for a total of 400 epochs.

6.1.4 Results

Results are presented for the two systems for the metrics classification accuracy, confusion rate and root mean square localisation error (RMSLE)

Classification Accuracy

Results of evaluation with the testing dataset using the classification accuracy for the two models are displayed for all reverb times in Table 61, and plotted against reverb time in Figure 97.

Table 61: CLASSIFICATION ACCURACY FOR 1D VS 2D TEST

	RT60		
Model	**0.5**	**1**	**1.5**
1D	48.7%	44.3%	41.2%
2D	50.5%	47.6%	43.8%

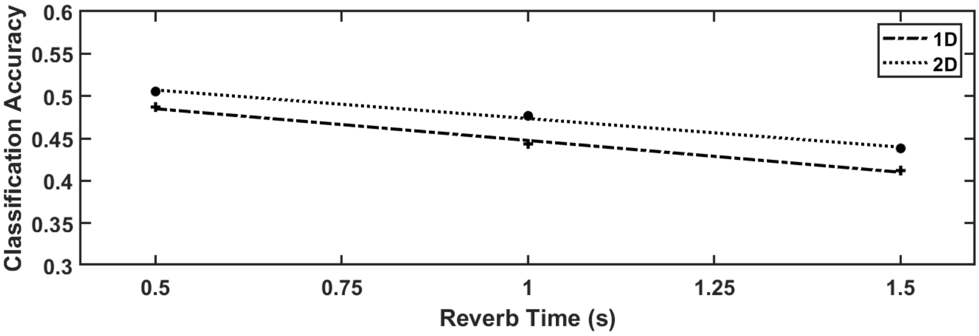

Figure 97: Classification Accuracy at different Reverb Times for 1D and 2D models

Confusion Rate

Results of evaluation with the testing dataset using the front-back confusion for the two models are displayed for all reverb times in Table 62, and plotted against reverb time in Figure 98.

Table 62: CONFUSION RATE FOR 1D VS 2D TEST

	RT60		
Model	**0.5**	**1**	**1.5**
1D	2.7%	3.5%	2.8%
2D	2.1%	2.6%	1.4%

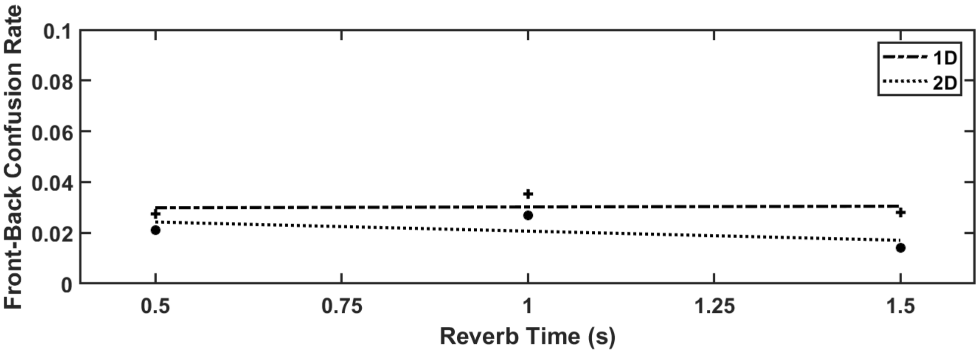

Figure 98: Confusion Rate at different Reverb Times for 1D and 2D models

RMSLE

Results of evaluation with the testing dataset using the RMSLE for the two models are displayed for all reverb times in Table 63, and plotted against reverb time in Figure 99.

Table 63: RMSLE for 1D vs 2D test

	RT60		
Model	**0.5**	**1**	**1.5**
1D	53.9°	54.1°	55.4°
2D	50.5°	48.6°	49.5°

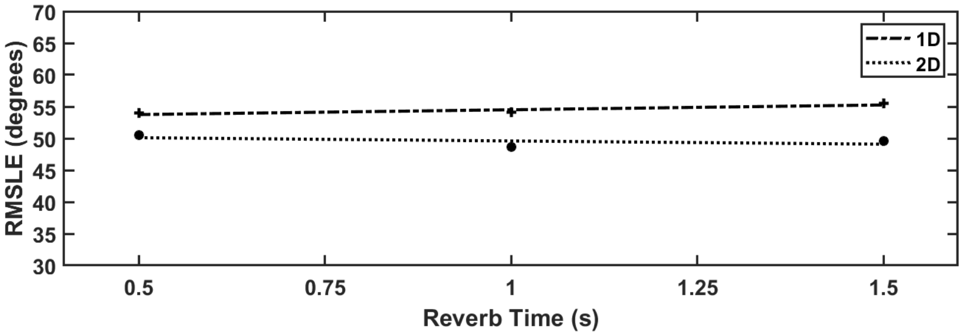

Figure 99: RMSLE at different Reverb Times for 1D and 2D models

6.1.5 Discussion

In Tables 61 - 63 and Figures 97 - 99 a clear preference for 2D convolution layers is observed, the 2D model outperforming 1D for all metrics at all reverb times, and regardless of number of epochs trained. Given this result, an emphasis on 2D convolution layers, and preparation of matrices for such, is continued throughout the rest of this work.

6.2 Introduction to Convolution Recurrent Neural Networks

Sound radiates from physical entities. The result of this is that the position of sound sources cannot show random behaviour over time, something which can be exploited by BSSL systems. This a strong rationale for the use of recurrence in BSSL models, as when we consider a sound source either moving or remaining still, this behaviour is predictable over time.

Much of the work concerning the use of CNNs for BSSL does not consider this cue over long ranges of time, instead treating oncoming audio as a series of mutually independent DoA estimation problems (Xu *et al.*, 2019; Zhou *et al.*, 2019), despite the promising employment of recurrence in CNNs used for array based sound localisation (Adavanne *et al.*, 2018). Later works do use recurrent layers, typically gated recurrent unit (GRU), in the context of a static sound localisation problem (Yang and Zheng, 2024), as well as in the context of trying to exploit dynamic cues through listener head rotation (García-Barrios *et al.*, 2022) and movement (Krause *et al.*, 2024a).

While the motivation for their inclusion is presumably improved performance, the degree to which this is true has not explicitly been studied. This chapter endeavours to do this through comparison of the performance of systems which do and do not include recurrent layers, as well as a comparison of performance between systems using different forms of recurrent layer.

The models presented in this chapter were developed, trained and tested using the MATLAB Deep Learning Toolbox, and the dataset generation was performed in MATLAB.

6.3 Convolutional Recurrent Neural Networks compared to Convolutional Neural Networks

The results in this sub-chapter were also presented in Reed-Jones *et al.* (2024b)

6.3.1 Introduction

Given the previously stated rationale, when trained on the same dataset with comparable parameters, a CRNN should be able to outperform a CNN at the task of BSSL. This hypothesis was tested experimentally, by comparing the ability of a CNN and a CRNN to localise static speech.

6.3.2 Binaural Audio Dataset

The same method of creating binaural sound scenes out of speech and noise sources as used in previous experiments was also employed for this experiment, using the TSP corpus for speech. To reflect the higher sample rate of the TSP corpus compared to TIMIT, a new set of BRIRs was generated. This was achieved using the same head-related impulse response (HRIR) of the KEMAR mannequin taken from CIPIC (Algazi *et al.* 2001), converted into BRIRs using the same image source method (ISM) library. This was undertaken for 100 rectilinear rooms with dimensions in the range of 1-10m. The absorption value was altered so as to achieve an RT60 between 0 and 1.5 seconds in length.

1000 audio samples taken from the corpus were used for training. Each of these was downsampled to 44.1kHz to match CIPIC's sampling rate, and then convolved with a random room from each of the 50 source directions. The resulting 50,000 audio files were then all truncated to 1 second in length.

Additionally, a noise mixture was added to the signal. This was again achieved through the summation of a series of BRIRs of random source directions convolved with pink noise. The resulting mixture was randomly scaled with a target signal-to-noise ratio (SNR) between 0 and 36 deciBels.

For the testing set BRIRs 10 room dimensions were generated, and for each of these sets of dimensions 3 sets of BRIRs were generated to achieve the target reverb times of:

$$T_{R60} = [0.5, 1, 1.5]\text{seconds} \tag{142}$$

The same process was undertaken with the noise mixture, with the pink noise replaced by ambient sound recordings, with the resulting mixture being scaled so as to achieve SNR values of:

$$SNR = [0, 12, 24, 36]dB(SNR) \tag{143}$$

All possible combinations of T_R60 and dB(SNR) were applied to another 100 speech files taken from the same corpus, yielding a total of 1600 audio files.

6.3.3 Feature Representations

The binaural dataset was then turned into two features representing phase and magnitude in the time-frequency domain. Two matrices were generated per 100ms of the original audio, so 10 pairs of matrices for each sequence.

For magnitude, this was achieved through the gammatone decomposition method described in Chapter 5.1. In this case, however, only 147 bands were used. A Hann window with a length 425 samples, and an overlap of 256 samples, yielded a matrix with the dimension $[147, 19, 2]$

For the phase representation, no auditory filtering was applied. Instead, a STFT was performed using the same window, and the output was downsampled to 147 frequencies. The angle of this was taken, and one channel was subtracted from the other, then projected onto a unit circle, to give a matrix representing interaural phase difference (IPD) of the size $[147, 19]$.

6.3.4 Feature Processing CNNs

The approach to CRNNs taken was to separate the system into distinct parts based on a two-stage training process, the parts being two CNNs trained on the training dataset, and a recurrent neural network (RNN) trained on the activations at the final convolution layer of the CNNs when classifying the training data. The CNNs can be thought of as a feature extraction processor, which is used to process the training and testing data.

The CNNs both have an almost identical design, differing only in the input layer size. This employs a flat architecture of three layers with eighteen 6×6 sized kernels; as per previous experiments this flat architecture is employed to avoid overfit. The CNNs were then trained on one frame of each sequence in the training set, using stochastic gradient descent (SGD) with a learning rate of 0.01 for a period of 100 epochs.

Table 64: CNN to be adapted into Convolutional Layers

Layer	**Hyperparameters**
Input Layer	
2D Convolution	(6,6), 18
Batch Normalisation	
Relu	
Max Pooling	(2,2)
2D Convolution	(6,6), 18
Batch Normalisation	
Relu	
Max Pooling	(2,2)
2D Convolution	(6,6), 18
Batch Normalisation	
Relu	
Dense	50

These CNNs were then used to classify the training set, and for each pair of matrices the activations at the final convolution layer was concatenated into sequences of the size $(5184, 10)$

6.3.5 RNN

The RNN trained on the activations was one with a shallow design, consisting of a single recurrent layer: a bidirectional long short-term memory (BiLSTM) containing 200 hidden units. The RNN was trained using SGD for 200 epochs with a learning rate of 0.0001

Table 65: CRNN Classifier

Layer	Hyperparameters
Input	(5184, 10)
BiLSTM	200
Dense	50

6.3.6 Baseline CNN

As opposed to the two CNNs already introduced, the CNN actually used as a baseline in this experiment is in fact a single layer perceptron (SLP) trained on these same activations. This is rightly called a CNN as it consists of the same layers as a typical CNN architecture, but with the added peculiarity that different parts of the network were trained separately.

Table 66: Baseline Classifier

Layer	Hyperparameters
Input	(5184, 1)
Dense	50

This SLP was trained using the same optimisation algorithm as the RNN.

6.3.7 Results

Performance for the BiLSTM and the baseline CNN according to the metrics accuracy, front-back confusion, RMSLE and Mirrored RMSLE are presented averaged across all reverb times and noise levels in 67. Additionally, RMSLE is plotted against reverb time in Figure 100, and against signal-to-noise ratio in Figure 101.

Table 67: Performance metrics from entire testing dataset

	Accuracy	Front-Back Confusion	RMSLE	RMSLE (Mirrored)
BiLSTM	81.15%	0.52%	13.28	7.06
Baseline	76.49%	1.7%	23.15	10.3

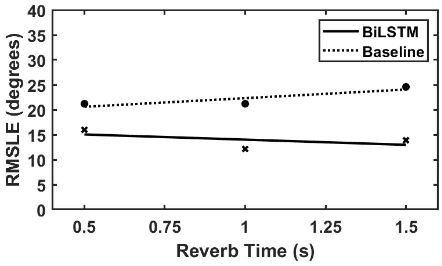

Figure 100: RMSLE plotted with respect to reverb time

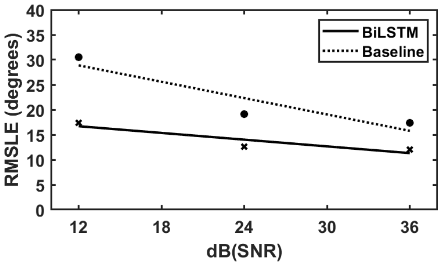

Figure 101: RMSLE plotted with respect to signal-to-noise ratio

6.3.8 Discussion

The results presented in Chapter 6.3.7 compare the relative performance of the use of a BiLSTM layer in a CNN with a baseline fully connected layer.

On all of the reported metrics in Table 67 a relative improvement of performance in the CRNN is seen over the baseline. The reported classification accuracy only improves by approximately 5%; however this corresponds to a 10° improvement in RMSLE, nearly halving the error. This suggests that use of CRNN not only improves correct classifications, but reduces the amount of error in incorrect classifications.

While front-back confusions were not exceptionally high in the baseline, they are reduced in the CRNN. Looking at the mirrored RMSLE reveals that this is not a significant contribution to the RMSLE; more likely a reduction in large spurious errors has been seen. Looking at the system's performance with respect to reverb time and SNR in Figures 100 & 101, a reduction in the gradient of the line is seen in the CRNN, suggesting an increase in the robustness of the system.

6.4 Comparison of Convolutional Recurrent Neural Network Architectures

The findings presented in this sub-chapter were also published in Reed-Jones *et al.* (2024a)

An important question currently missing from the literature is what type of recurrent layer is suitable. GRU and bidirectional gated recurent unit (BiGRU) have been applied elsewhere, but it should be established if either of these are preferred, or if long short-term memory (LSTM) and BiLSTM can achieve better performance. To do this, a comparison was undertaken where four RNNs were trained on the same activations described in Chapter 6.3.4.

The four RNNs had the same shallow design as seen in Table 65, with the recurrent layer being replaced in each case. All four used 200 hidden units. This is shown in recurrent layer in Table 68, in which the recurrent layer is substituted with GRU, BiGRU, LSTM and BiLSTM during the experiment.

Table 68: The RNN architecture used in the comparison experiment

Layer	Hyperparameters
Input	(5184, 10)
Recurrent Layer	200
Dense	50

Each of these RNNs was trained on the training set of activations using SGD with a learning rate of 0.0001 over a period of 200 epochs.

6.4.1 Results

Results are presented first in a general comparison of the four models for the four metrics classification accuracy, front-back confusion, RMSLE and mirrored RMSLE averaged across all noise levels and reverb times in Table 69. These are then also presented for each individual combination of reverb time and signal-to-noise ratio in Table 70.

In addition to this, RMSLE is also plotted in Figures 102 & 103.

Table 69: Performance metrics from entire testing dataset

	Accuracy	Front-Back Confusion	RMSLE	RMSLE (Mirrored)
GRU	70.53%	1.64%	35.18°	20.2°
BiGRU	70.48%	1.53%	35.94°	20.62°
LSTM	71.62%	1.47%	34.19°	19.54°
BiLSTM	72.67%	0.01%	34.04°	19.3°

Table 70: Complete results for all recurrent layers

GRU

	Noise Level			
RT60	0	12	24	36
0.5	67.9°	20.2°	9.1°	8.2°
1	64.6°	17°	10.4°	9.7°
1.5	64.4°	18.5°	16.1°	16.3°

BiGRU

	Noise Level			
RT60	0°	12°	24°	36°
0.5	70.1°	19.9°	7.6°	5.7°
1	66.6°	18.2°	9.8°	8.3°
1.5	66.2°	19°	14.1°	14.5°

LSTM

	Noise Level			
RT60	0	12	24	36
0.5	67.5°	19.5°	12.7°	11.2°
1	64.8°	17.5°	10.6°	9.6°
1.5	66.2°	17.1°	11.3°	11.2°

BiLSTM

	Noise Level			
RT60	0	12	24	36
0.5	66.6°	18.5°	10.5°	9.9°
1	62.8°	17.6°	11.9°	9.3°
1.5	62.7°	16.4°	11.3°	10.3°

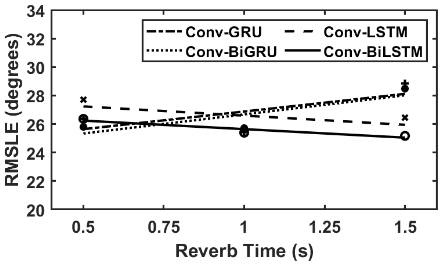

Figure 102: Localisation Error of different recurrent layers with respect to T_{R60}

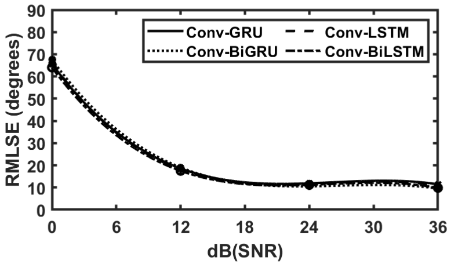

Figure 103: Localisation Error of different recurrent layers with respect to SNR

6.4.2 Discussion

The results in Chapter 6.4.1 show the relative performance of four types of recurrent layer: GRU, BiGRU, LSTM, BiLSTM.

In overall performance, as seen in Table 69, the difference between all the layers is very slight, with a total difference between the highest and lowest RMSLE only 1.14°. This makes drawing conclusion on a favoured layer questionable, as all seem capable of similar performance; and larger or different datasets may easily yield different results.

What is seen, however, is a slight preference towards bidirectional layers of single directional layers, and a preference for LSTM over GRU. This is despite the relative popularity of GRU in literature on using CRNN for BSSL (García-Barrios *et al.*, 2022; Krause *et al.*, 2024a; Krause *et al.*, 2024b).

It can be seen in Table 70 that significantly higher level of error are seen when the SNR is at 0dB. This significantly skews the results and leads to the errors of over 30° seen in Table 69, when in reality at lower noise level the RMSLE is near-always below 20°.

One shortcoming of this study is the use of static sound sources. These were used to align to the general, but not complete, emphasis on static speaker localisation identified in the current state of the field. Movement of the speaker, however, is both likely, and a significant temporal cue in the context of BSSL.

6.5 Conclusion

In this chapter, experiments were undertaken to provide insight into choice of deep learning model architecture for the task of BSSL. In Chapter 6.1, the performance of 1D and 2D convolutional layers in CNNs were compared: it was found that the use of 2D convolutional layers is preferable.

Chapters 6.3 & 6.4 presented experiments on using CRNNs for BSSL. In Chapter 6.3 a CRNN containing a BiLSTM layer was compared to a baseline CNN system, and in Chapter 6.4 the recurrent layers LSTM, BiLSTM, GRU, BiGRU were compared for their relative performance when used in a CRNN trained for for BSSL.

The most significant conclusion is a clear preference for performing BSSL with CRNN over CNN. This shows the trend in conventional sound source localisation (SSL) towards CRNN is correct for the case of BSSL as well.

In comparing the types of layers, a very soft preference is given towards BiLSTM, however the significance of this is questionable and does not suggest a strong case for using LSTM based layers over GRU based layers.

7 Mismatched Anechoic Condition

Reconsider the model of spatial hearing, for a speaker on the horizontal plane in the freefield.

$$y_{L,R}[n,\varphi] = s[n] * \mathrm{hrir}_{L,R}[n,\varphi] + \eta[n] \tag{144}$$

where $y[n]$ is the reproduced signal, which should be akin to pressure at ears, $s[n]$ is the speech signal, and $\eta[n]$ is the additive noise.

In using measured head-related impulse responses (HRIRs) to create binaural audio using this model another inaccuracy arises; measured HRIRs do differ from the true HRIR. These differences are likely to arise from:

- Room effects
- Measurement apparatus
- Measurement and processing method
- Unintended noise in measurement

'Anechoic room effects' is somewhat oxymoronic; however it is a valid observation as no rooms, including anechoic chambers, are truly anechoic. These effects can effectively be thought of as noise which has been convolved with the signal, or convolutive noise. Expanding the model with this knowledge therefore gives:

$$y_{L,R}[n,\varphi] = s[n] * \mathrm{hrir}_{L,R}[n,\varphi] * \varsigma[n] + \eta[n] \tag{145}$$

where $\varsigma[n]$ is the convolutive noise.

Another helpful distinction should be made at this point: some sources of convolutive noise will vary with position, while some will be present throughout a measurement dataset.

Consider anechoic condition experiments, such as those presented in Chapter 4, but also those found in the literature review, Chapter 2.4: it is typical that a training and testing dataset will be created from convolving HRIRs of different direction of arrivals (DoAs) with monophonic sound sources, mismatch between the training and testing data is introduced only through ensuring different speech samples are used between the training and testing datasets.

Typically it is only studies which specifically address the mismatched head-related transfer function (HRTF) condition that deviate from this (Wang *et al.*, 2019; Wang *et al.*, 2020; Qian *et al.*, 2022). This leads to an issue that even when performance is high, it is not necessarily true that the system is identifying DoA based upon the true HRIR, it may instead be identifying differences based upon the position dependent convolutive noise. Furthermore, group convolutive noise in an HRTF dataset could have the effect of rendering a dataset not robust to other datasets, and therefore real data.

This is an issue that has previously been identified by Hammond (2021), who termed it the mismatched anechoic condition, as differences between different datasets are effectively an equivalent problem to mismatched HRTFs.

It should also be noted that this is of relevance to non-anechoic systems. As previously established, most experimentation into binaural DoA estimation in the diffuse field uses room simulation to create reverberant environments. The reflections in this reverberation were typically created through convolution-delayed and weighted versions of the original signal with HRIRs taken from the original signal. Studies which have used real binaural room impulse response (BRIR) measurements are less prone to this error, however there will still be common characteristics amongst measurement apparatus and method which are not found outside of those measurements.

This chapter, then, addresses binaural DoA estimation in the mismatched anechoic condition, profiling performance of systems trained on only one set of HRTF, as well as the proposal and analysis of data augmentation methods designed to mitigate this issue.

The models presented in this chapter were developed, trained and tested using the Tensorflow library for Python, while the dataset generation was performed in Matlab.

7.1 The ‘Club Fritz’ HRTF dataset

To study the effects of the mismatched anechoic condition, it is helpful to have a compilation of anechoic measurements of the same head simulator. This exists in ‘Club

Fritz', the name given to a round-robin study of HRTF measurements of a KU100 head simulator, from which 12 sets of HRTFs measurements have been published (Katz and Begault, 2007). The institutions involved have been described in Chapter 2.4.1.

The HRTF measurements in Club Fritz are consistent only in that the same receiver is used in every instance. Beyond this, there can be mismatch in the Source Positions measured, the acoustic source, the impulse type, and more.

Theoretically for farfield measurements the binaural cues interaural time difference (ITD) and interaural level difference (ILD) should remain consistent across datasets for measurements taken at the same azimuth and elevation, as it is the head alone that is responsible for these cues. To demonstrate this, the ITD below 1500Hz, and the ILD in the range 11.25-12.75kHz are plotted with respect to azimuthal angle on the horizontal plane for all measurement sets, as seen in Figure 104.

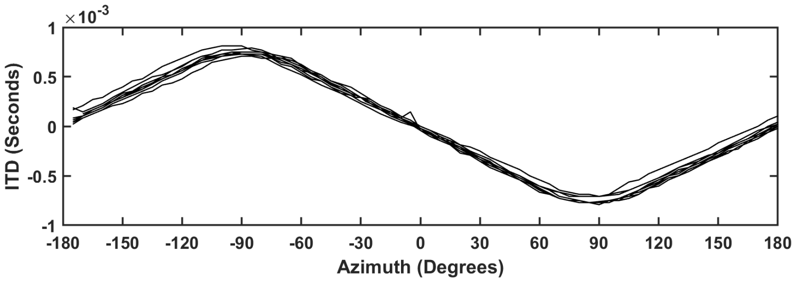

Figure 104: ITD of Horizontal Plane measurements in Club Fritz

While there is a strong trend seen between all measurement sets, a small amount of deviation in ITD between measurements is seen in Figure 104. Some of this is constant offset, which may have been caused by an offset in the rotation of the head, but there are also a pair of HRTFs measurements that have a lower maximum ITD than that seen in other measurement sets. These two HRTFs, being the ones measured at the University of Maryland and NASA, which both used loudspeakers on a measurement arc with a radius of only 0.9m. This is below what is normally seen as the requirement to meet the farfield assumption for binaural measurement, and is the most likely reason for this reduction in ITD.

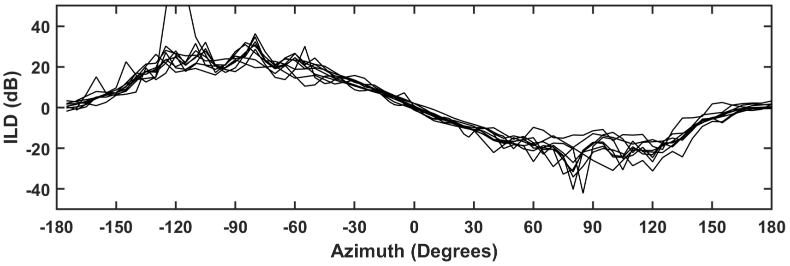

Figure 105: ILD of Horizontal Plane measurements in Club Fritz

While the expected trend is still seen in ILD, the data is much noisier and the differences between measurement datasets is much larger. This is to be expected, as ILD is much more likely to change owing to measurement noise, and because ILD is considered at higher frequencies, small differences in position between measurements are likely to become more noticeable.

Given that binaural cues do not remain entirely consistent between the measurements, it is also unlikely that monaural cues will be consistent. Recalling that monaural cues refer to the differences in the magnitude at different frequencies for different positions, it is possible to see if this is different between different measurement sets through the plotting of the common transfer function (CTF) of each measurement set; the CTF being the part of the HRTFs which is common for all DoAs. This, ideally, would be identical in all measurement sets. The CTF is found simply by finding the average of all the magnitudes of HRTFs in a set:

$$\mathrm{CTF}[\omega] = \frac{1}{K}\sum_{k=1}^{k} |\mathrm{HRTF}_k[\omega]| \tag{146}$$

where k is an index representing different DoAs.

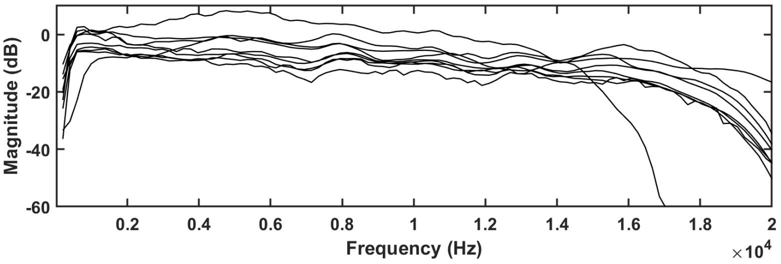

Figure 106: CTFs of different measurement sets in Club Fritz

Figure 106 shows the CTFs of the HRTFs found in Club Fritz. It can be seen that the difference is quite large. This is to say that the measurement conditions can greatly affect binaural cues.

7.2 Localisation Performance in the Mismatched Anechoic Condition

No study has been conducted showing the effects of training and testing binaural DoA estimators on mismatched anechoic HRTF datasets. To fill in this gap, therefore, a study was undertaken in which a convolutional neural network (CNN) was trained on a set of HRTF measurements of a specific head simulator, and then tested using different unmatched HRTF measurements of head simulator.

To test sound localisation ability in the mismatched anechoic condition, training and testing datasets were created using the HRIRs in the Club Fritz dataset.

7.2.1 Training and Testing Datasets

Binaural datasets were created by convolution of the Club Fritz HRIRs with speech taken from the telecommunications and signal processing laboratory (TSP) corpus. Due to source positions being determined by the measurement apparatus, they are not common between the measurement datasets, as was shown in Figure 23 in Chapter 2.4.1. For this test, for a true representation of this task, it is necessary to avoid interpolation and so the choice of azimuthal positions is not obvious. Table 71 shows that many of the source HRTF sets divide the horizontal plane by a multiple of 36. As these datasets sample the horizontal plane with equal spacing, and none begin on a number differing from 0°, a 10 degree spacing will allow for perfect coverage of 10 of the 12 HRIRs.

Table 71: Number of Horizontal Plane Source Positions of every HRTF set in Club Fritz

HRTF Number	1	2	3	4	5	6	7	8	9	10	11	12
No. Positions	72	58	36	72	144	36	72	72	72	36	60	90

HRTF set 11 equally samples the horizontal plane in 60 points, leading to 6° spacing, and so has very few common DoAs with the other sets. Given this difference, this set is completely ignored in this and following tests.

As previously established, two of the HRIRs do not meet the farfield condition and that has an effect on binaural cues. Given this, these two HRIRs, numbers 2 and 3, are also ignored for these tests. This has the added benefit that all 9 remaining HRTFs contain all 36 target DoAs.

Speech samples were taken from the TSP dataset. The first 1000 files of over a second in length in the dataset were shortened to 1 second in duration. Each of these second long sections were then convolved with the 36 HRIRs, of all 9 HRIRs sets.

A further 100 files were used to create testing dataset, with these files again being shortened to 1 second segments, and convolved with all possible combinations of azimuthal direction and HRIR.

The resulting binaural audio was then used to create magnitude and interaural phase difference (IPD) matrices.

Per each 1 second clip, 10 100ms sections are used to create 10 pairs of matrices. These are created through short-time fourier transform (STFT) of the audio, using a rectangular window of 440 samples, which was moved with a hop size of 440 samples. From these complex matrices, the magnitude and phase was taken to give two matrices per audio clip with the size $[221, 10, 2]$

Efficient dimensionality reduction was desired, leading to the choice to use an equal window size and hop size. Given that this results in no overlap between windows, a rectangular windowing function was used to window the signal. The benefit in terms of dimensionality reduction was seen as favourable compared to possible spectral artefacts caused by this type of windowing.

Table 72: STFT parameters for creating feature representations

Parameter	**Value**
Sampling Rate	48kHz
Window Size	440
Hop Size	440

The result is feature representations for all 9 sets of training, validation and testing data each representing a unique HRTF from the Club Fritz dataset

7.2.2 Model, Training and Testing

A two-branched CNN was trained and tested on the resulting data. This consisted of four 2D Convolution layers per branch, as seen in Table 73. This model is almost identical to the 2D convolution model used in Chapter 6.1, changing only in the length of the output vector. This was used for the same reason, for its ability to avoid overfit as compared to CNN with growing filter sizes. This model was trained independently on all 9 sets of training data, using an Adam optimiser for a period of 100 epochs, as seen in Table 74.

Table 73: CNN for localisation in mismatched anechoic condition

IPD Branch		**Magnitude Branch**	
Layer	**Hyperparameters**	**Layer**	**Hyperparameters**
Input	(221, 10, 2)	Input	(221, 10, 2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Batch Normalisation		Batch Normalisation	
Max Pooling	(2,2)	Max Pooling	(2,2)
2D Convolution	(3,3), 16	2D Convolution	(3,3), 16
Dropout		0.5	
Dense, Softmax		36	

Table 74: Adam Optimiser Parameters for mismatched anechoic test

Parameter	Value
Rate	1e-5
Decay	1e-3

This created 9 trained models representing each of the training datasets. Each of these models were then used to localise all 9 testing datasets, such that there were 9 × 9 sets of results in total.

7.2.3 Results

Results, in the form of classification accuracy, root mean square localisation error (RMSLE), and confusion rate are first presented in the form of a cross performance matrix, representing every combination of training and testing dataset.

The metrics are scaled from black to red with maximum ranges possible for each, according to the following ranges

Classification Accuracy 0 to 1

RMSLE 0° to 180°

Confusion Rate 0 to 1

This is shown in Figure 107.

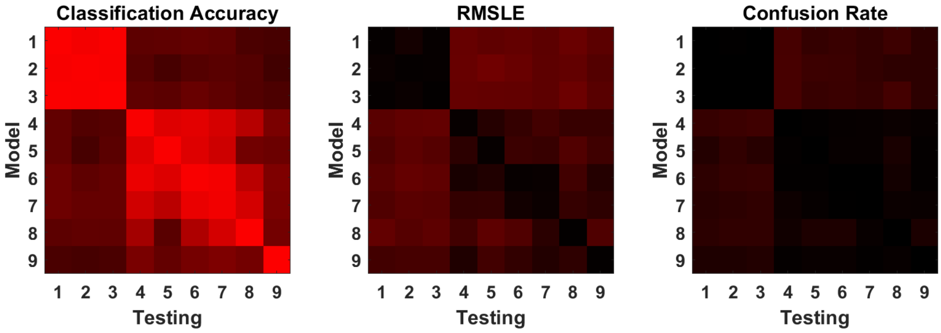

Figure 107: Inter-Measurement set Performance Matrices for CNN Trained on HRTFs in Club Fritz

In addition to this, performance is presented for every model when tested on all unseen HRTF datasets, being the positions in Figure 107 not on the main diagonal. This is shown in Table 75.

Table 75: Average Performance of CNN in mismatched anechoic condition

Model	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
Accuracy	52.8%	49.4%	51.2%	60%	52.6%	61.7%	62.6%	51.4%	39.4%	*53.5%*
RMSLE	45.6°	49.9°	47.9°	46.1°	51.8°	49.4°	42.9°	54.1°	47.9°	*48.4°*
Confusion	12.2%	14.1%	14.1%	12.5%	10.8%	11.2%	8.9%	12.2%	9.2%	*11.7%*

7.2.4 Discussion

From the results reported in Figure 107 and Table **??**, it can clearly be seen that if the CNN is trained on one HRTF set, it is not guaranteed to generalise to other measured HRTFs of the same head, despite these theoretically being identical.

For every model under test, the reported accuracy is perfect when tested with binaural data made with that same series of HRTF measurements; however examples of a model well predicting the azimuth of binaural data created with another measurements dataset are very infrequent; there just appear to be some pairs of measurement sets that are well matched.

Figure 107 reveals some clustering in the results, particularly with the first three sets of HRTFs results which are completely able to generalise to each other, and entirely unable to generalise to other other sets of HRTFs. Some models, such as 5 and 9 seem particularly poor at generalising to all other models.

Compared to previous tests, the reported confusion rate here is large, with a reported average front-back reversal rate of 16%. This shows that similar but different measurements have the potential to lead to front-back reversal, which is particularly problematic and likely is a large contributor to the very large RMSLE seen. This issue, as per many issues encountered in deep DoA estimation, as well as machine learning in general, can be categorised as a problem of overfit. It is, however, a problem of overfit that is often overlooked in research on deep learning based binaural sound source localisation (BSSL)

7.3 Increasing Number of HRTFs

It has been established that the issue of localising in the mismatched anechoic condition is an issue of overfit. Given this, it would make sense that increasing the number of HRTF measurements sets is a potential solution to this issue.

To assess this possible solution, two tests were undertaken wherein the number of HRTFs present in the training dataset is increased to 2, 4 and then 8, these numbers being chosen to make the dataset more easily divisible. The method and results for both of these tests are presented in this sub-chapter.

7.3.1 Training Datasets

Three sets of training datasets were made for this test, for training with two, four, and eight HRTFs. The HRTFs included in each test were in the range $n \leq \mathrm{hrtf}_n < n+2$ for two HRTFs, $n \leq \mathrm{hrtf}_n < n+4$ for four HRTFs and $n \leq \mathrm{hrtf}_n < n+8$ where numbers above 9 are wrapped back to 1 (so that 10 == 1). This is described fully in Table 76.

Table 76: Distribution of HRTFs between training datsets for increased HRTFs testing

Set	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
2 HRTFs	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-1
4 HRTFs	1-4	2-5	3-6	4-7	5-8	6-9	7-10	8-11	9-1	10-2	11-3
8 HRTFs	1-8	2-9	3-10	4-11	5-1	6-2	7-3	8-4	9-5	10-6	11-7

These HRTFs were convolved with the same speech samples used in baseline tests including only one HRTF: however, the number of speech samples used is reduced such that each speech sample is independently convolved with every HRIR, but the total final number of samples remains identical. This distribution is shown in Table 77.

Table 77: Number of speech samples per DoA in training set for each test

Set	Speech Samples	HRTFs	Total per DoA	Absolute Total
Baseline	1000	1	1000	36,000
2 HRTFs	500	2	1000	36,000
4 HRTFs	250	4	1000	36,000
8 HRTFs	125	8	1000	36,000

Both these new training datasets were used to train the same model used in baseline experiment described in Table 73, which was also trained on the training datasets for the same duration with identical training parameters as the baseline.

7.3.2 Model, Training & Testing

The same model and training parameters as used in Chapter 7.2.2 was used to train the 11 models based on the created training datasets. These were then tested on the exact same dataset as described in Chapter 7.2.1, so as to allow for direct comparison in the results measured on only one HRTF from the baseline system.

7.3.3 Results

Cross Performance Matrices are presented individually for 2, 4 and 8 HRTFs in Figs. 108-110. Additionally, average performance when evaluated on mismatched testing dataset is shown in Table 78.

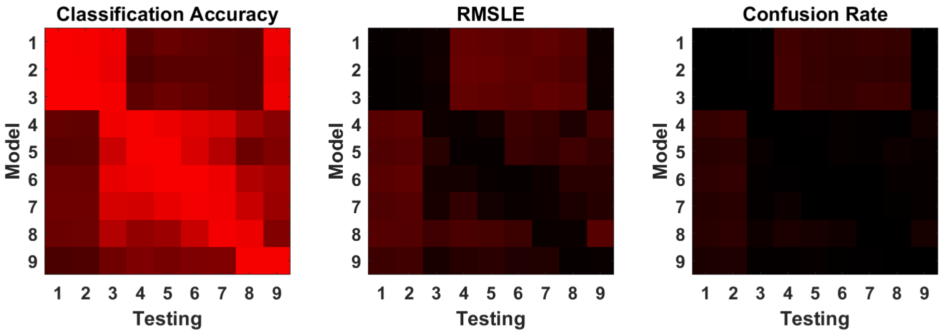

Figure 108: Inter-Measurement set Cross Performance matrices for CNN Trained on 2 HRTFs in Club Fritz

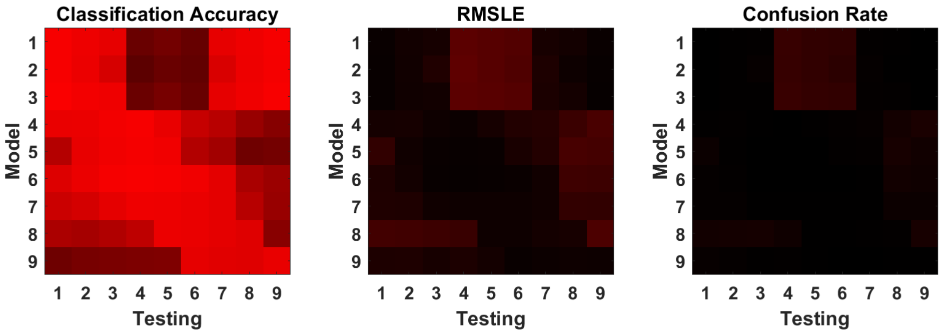

Figure 109: Inter-Measurement set Cross Performance matrices for CNN Trained on 4 HRTFs in Club Fritz

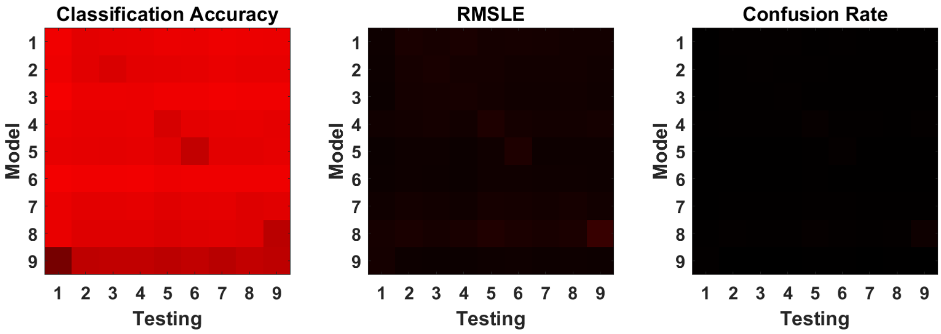

Figure 110: Inter-Measurement set Cross Performance matrices for CNN Trained on 8 HRTFs in Club Fritz

Table 78: Average Performance for each Model when Increasing Number of HRTFs Trained On

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
2 HRTFs	**Accuracy**	47.2%	47.0%	78.8%	55.4%	58.4%	58.8%	57.5%	50.2%	65.8%	*57.7%*
	RMSLE	49.3°	51.8°	20.9°	49.3°	45.0°	48.5°	43.5°	42.0°	31.1°	*42.4°*
	Confusion	49.3°	51.8°	20.9°	49.3°	45.0°	48.5°	43.5°	42.0°	31.1°	*42.4°*
4 HRTFs	**Accuracy**	69.4%	75.9%	76.1%	48.4%	54.9%	53.4%	80.0%	67.0%	54.4%	*64.4%*
	RMSLE	29.0°	23.0°	21.9°	50.9°	42.8°	43.2°	20.6°	37.0°	46.3°	*35.0°*
	Confusion	3.8%	3.0%	2.8%	14.5%	12.1%	11.5%	1.8%	6.0%	7.4%	*7.0%*
8 HRTFs	**Accuracy**	46.3%	88.8%	85.0%	92.1%	83.8%	76.1%	93.6%	87.0%	72.7%	*80.6%*
	RMSLE	15.3°	18.7°	18.0°	16.7°	21.5°	21.6°	11.2°	15.5°	39.0°	*19.7°*
	Confusion	1.5%	1.8%	1.8%	1.6%	3.0%	2.1%	0.7%	1.0%	5.7%	*2.1%*

7.3.4 Discussion

The results presented in Chapter 7.3.3 show the performance of CNNs trained on increasing numbers of different HRTF measurement sets of a single binaural array.

The expected trend is revealed: increasing the number of HRTF measurement sets also allows the model to better generalise to other unknown measurement sets. Looking only at the cross performance matrices could be misleading, as a much larger proportion of the entire matrix becomes covered by models being tested on already seen HRTF measurement sets. Table 78, which only includes the results of unknown results, confirms this is not the case however, and that increasing the number of HRTFs does improve generalisation.

Even in the case of maximum number, 8 HRTFs, the system still cannot perfectly generalise to other HRTF measurements; two systems still report a large degree of error, model number 1 and 6. It should be noted however, that these two models are tested solely on data made with HRTF sets 9 & 5, the two sets previously identified as generalising poorly to the other sets. In this case, increasing the number of HRTF has the downside that it reduces the statistical significance of the testing dataset.

Comparing these results of increased number of HRTFs to the case with single, it is seen that performance in fact decreases when increased to two HRTFs. This is not the general trend, but shows that data complexity is still a concern and there can be potential for doing more harm than good.

7.4 Converting HRTFs to Directional Transfer Functions

In the introduction of this chapter, the differences between HRTFs measurements was shown through the use of CTF; the average transfer function across all frequencies in the measurement set shown in Figure 106. Given that this difference is known, it is possible to use this as an equalisation filter, under the presumption that taking away these differing common elements will better align the HRTFs.

This creates a transfer function which is termed the directional transfer function (DTF). It is a technique used in spatial audio, under a similar assumption that the CTF describes the unwanted parts of the signal

$$\mathrm{DTF}[\omega, r, \theta, \varphi] = \frac{\mathrm{HRTF}[\omega, r, \theta, \varphi]}{\mathrm{CTF}[\omega]} \tag{147}$$

This method is similar to the diffuse-field equalisation technique commonly applied to HRTF measurements, also equalises the HRTFs by the inverse of the average, however this is more typically executed using the root mean square (RMS) of the HRTFs:

$$H_{eq}[\omega, r, \theta\, \phi] = \frac{\mathrm{HRTF}[\omega, r, \theta, \varphi]}{\sqrt{\frac{1}{K}\sum_{k=1}^{k} |\mathrm{HRTF}_k[\omega]|^2}} \tag{148}$$

Given this similarity, it is important to note then that no such equalisation has already been applied to the HRTFs distributed as part of the Club Fritz study.

7.4.1 Method

An implementation of this taken from the auditory modelling toolbox (Majdak *et al.*, 2022) was applied to the all of the sets of HRTFs. The new resulting sets of DTFs were used to create training and testing datasets in a way exactly identical to that described in Chapter 7.2. Furthermore, it was tested whether doing this for increased number of measurement sets leads to a similar trend as that presented in chapter 7.3.3, and so the number of DTFs is increased in an exactly identical manner.

Table 79: Number of speech samples per DoA in training set for each test

Set	Speech Samples	DTFs	Total per DoA	Absolute Total
1 DTF	1000	1	1000	36,000
2 DTFs	500	2	1000	36,000
4 DTFs	250	4	1000	36,000
8 DTFs	125	8	1000	36,000

7.4.2 Results

Cross Performance matrices are presented individually for 1, 2, 4 and 8 DTFs in Figs. 111-114. Additionally, performance when evaluated on mismatched testing data is seen in Table 80.

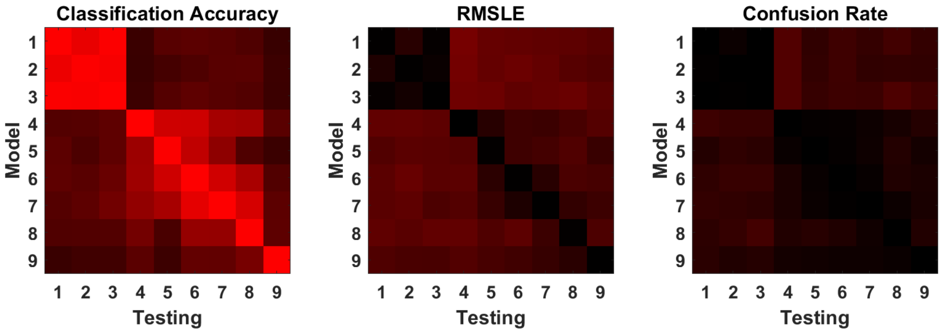

Figure 111: Inter-Measurement set Cross Performance matrices for CNN Trained on DTFs

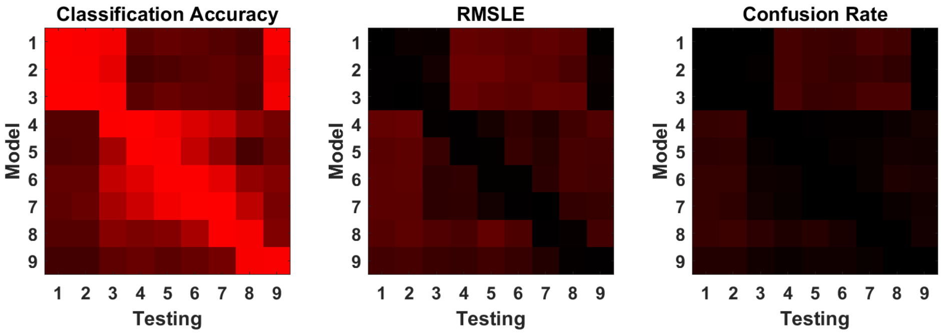

Figure 112: Inter-Measurement set Cross Performance matrices for CNN Trained on 2 DTFs

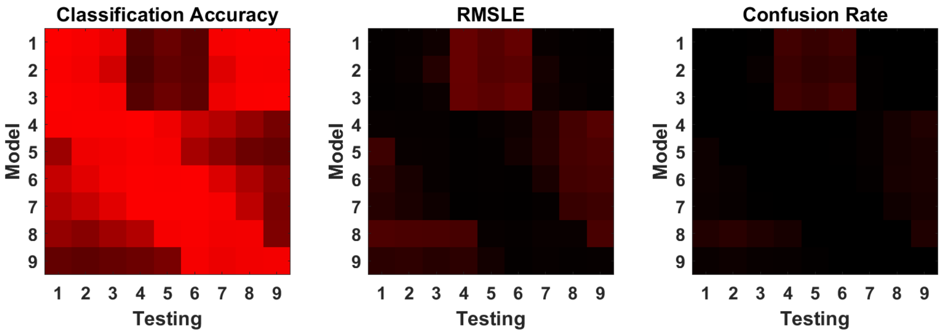

Figure 113: Inter-Measurement set Cross Performance matrices for CNN Trained on 4 DTFs

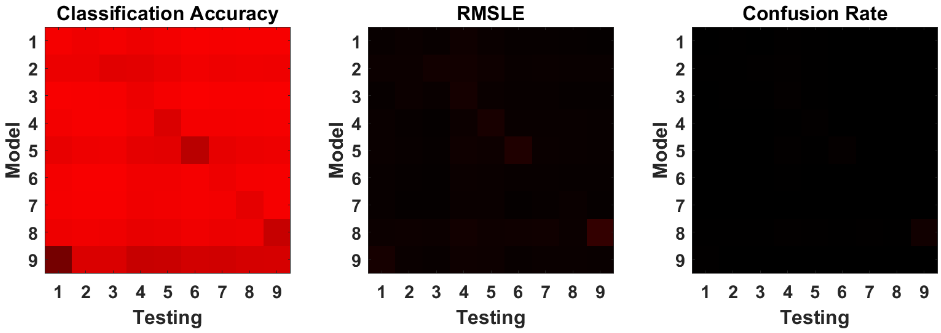

Figure 114: Inter-Measurement set Cross Performance matrices for CNN Trained on 8 DTFs

Table 80: Average Performance across models trained on DTFs of Club Fritz

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
1 DTF	**Accuracy**	47.9%	47.5%	51.4%	41.2%	46.7%	55.6%	50.1%	48.6%	28.9%	*46.4%*
	RMSLE	50.2°	53.5°	47.0°	68.9°	52.8°	53.8°	50.9°	53.1°	53.3°	*53.7°*
	Confusion	14.0%	15.0%	14.8%	17.7%	11.4%	12.1%	11.1%	15.1%	15.0%	*14.0%*
2 DTFs	**Accuracy**	42.8%	43.3%	68.8%	50.8%	55.4%	53.8%	53.9%	43.7%	60.5%	*52.5%*
	RMSLE	52.7°	55.9°	32.1°	53.5°	50.7°	51.1°	44.0°	52.1°	35.8°	*47.5°*
	Confusion	16.4%	16.6%	6.2%	15.6%	13.3%	12.4%	13.2%	14.5%	6.1%	*12.7%*
4 DTFs	**Accuracy**	61.0%	70.4%	72.8%	41.5%	52.6%	49.8%	78.3%	67.9%	46.6%	*60.1%*
	RMSLE	37.2°	25.3°	25.2°	59.2°	40.6°	46.0°	19.8°	37.9°	51.8°	*38.1°*
	Confusion	6.3%	5.5%	4.1%	17.2%	12.8%	14.5%	1.9%	6.4%	10.9%	*8.9%*
8 DTFs	**Accuracy**	45.9%	93.5%	88.2%	92.7%	85.4%	72.5%	93.0%	89.7%	77.6%	*82.1%*
	RMSLE	16.0°	8.2°	13.0°	14.8°	17.3°	21.8°	6.0°	7.3°	36.0°	*15.6°*
	Confusion	0.9%	0.7%	0.9%	1.7%	1.4%	2.0%	0.4%	0.3%	6.7%	*1.7%*

7.4.3 Discussion

The results in Chapter 7.4.2 show performance on CNN trained and tested on DTFs created from HRTFs in the Club Fritz dataset.

Comparing Table 80 to Tables 76 and 75 reveals only a small change in performance between use of HRTFs and DTFs, with HRTFs slightly outperforming DTFs for low counts of measurement datasets, and DTFs slightly outperforming HRTFs for higher counts of measurement datasets. This suggests that the difference between measurement datasets leading to lack of generalisation is not well described within the differences in CTFs, and that these differences are not relevant at all. Based on this it is concluded that conversion to DTF is not necessary.

7.5 Augmentations to HRTFs for Improved Performance

The most obvious solution to this issue would be the solution most typically applied to most overfit issues in deep learning: increase the quantity and variety in the training data. For anechoic HRTFs however, this is difficult: if it is supposed that the model is too heavily fitting to artefacts in the individual HRTF measurements, then it is difficult to acquire more of these as generally only one set of measurements of a subject is published. If the issue is differences between measurement datasets as a whole, then a similar problem exists that there are not enough institutions which have measured HRTFs to create adequate variety.

One solution may be to augment the HRTFs in ways that reduce the models' ability to overfit to misleading information.

7.5.1 Horizontal Plane Mirroring

The human head (and consequently the HRTF) are very close to but not exactly symmetrical, meaning the binaural cues in sound sources of positions reflected by the median plane are close to a perfect additive inverse, and the monaural cues are almost exactly identical.

This information is used to justify measuring the full horizontal plane, rather than only half the plane as split by the median plane. However, in this case it creates an opportunity for an augmentation; if HRTFs from one half of the median plane are mirrored onto the other half, it is possible to create two sets of HRTFs from one set of measurements.

This idea has previously been applied in deep learning with HRTFs (Pauwels and Picinali, 2023), but in the context of a test only looking at monaural cues, addressing the suitability of mismatched HRTF datasets for deep learning tasks in general.

Method

The augmentation is achieved by mirroring sources about the median plane. What this means exactly, is that the horizontal plane is split into two halves down the median plane. For one dataset HRIRs on the left half of the plane are original measurements, and HRIRs on the right half of the plane are the channel reversed version of the measurements on the left half. The two positions directly in front and behind of the listener were not altered.

This was then repeated again but such that the right half retains original measurements, and the left half now contains a channel reversed version of the right half.

These two new sets of HRTFs are then used to create training dataset for each of the original HRTF measurements of exactly equal size to those in Chapter 7.2.1. This was achieved by using the same speech samples, but alternating which of the two HRTFs from the augmented sets are used for convolution for each audio file.

The same model as described in Table 73 was trained using the same parameters described in Table 74 for the same period of 100 epochs.

Results

The cross performance matrix for models trained on horizontal plane mirroring augmented datasets are shown in Figure 115.

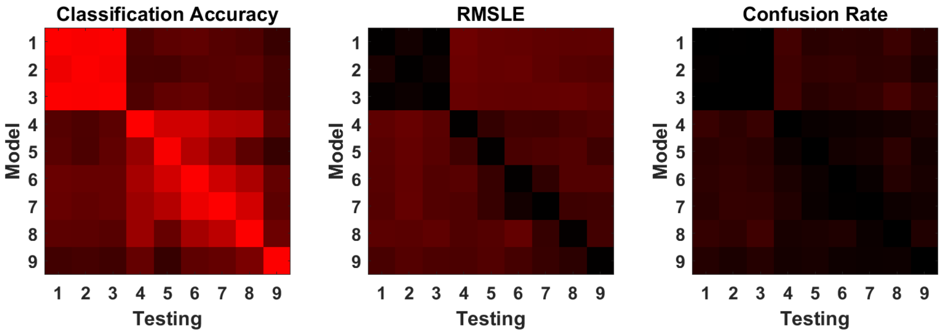

Figure 115: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF with Horizontal Plane Mirroring

Average performance is presented compared with change (Δ) of performance compared to the baseline of a single HRTF, as seen in Table 81

Table 81: Average Performance for CNN Trained on HRTFs with Horizontal Plane Mirroring

Model	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
Accuracy	52.0%	51.1%	52.4%	48.3%	51.1%	58.5%	53.8%	51.5%	32.7%	*50.2%*
Δ	4.0%	3.6%	1.0%	7.1%	4.4%	2.8%	3.7%	2.9%	3.8%	*3.7%*
RMSLE	44.5°	49.7°	42.9°	58.1°	51.5°	51.2°	47.7°	51.3°	51.3°	*49.8°*
Δ	-5.7°	-3.9°	-4.1°	-10.8°	-1.3°	-2.6°	-3.2°	-1.8°	-2.0°	*-3.9°*
Confusion	13.7%	13.3%	14.5%	15.4%	10.6%	11.0%	10.4%	14.9%	11.7%	*12.8%*
Δ	-0.4%	-1.6%	-0.3%	-2.3%	-0.8%	-1.1%	-0.6%	-0.2%	-3.4%	*-1.2%*

Discussion

Chapter 7.5.1 shows the results of training CNNs with the horizontal mirroring augmentation, tested under the mismatched anechoic condition.

Table 81 shows that performance over the baseline of a single HRTF causes an average 3.9° improvement in RMSLE. This modest improvement suggests there may be some benefit to the horizontal mirroring method. Unfortunately, however, as binaural audio is 2 channels, it is only possible to perform this augmentation once per HRTF. This, however, can still help exaggerate the diversity of small collections of HRTF measurements.

7.5.2 Horizontal-Plane Interpolation

Given the apparent improvement of performance it is supposed that there may be other methods to manipulate existing data in a set of HRTFs such that small differences are encountered, leading to potential increases in generalisation.

It is recalled that one of the hypothesised reasons a for lack of generalisation in the mismatched anechoic condition is that CNNs may overfit to convolutive noise found in individual HRTF measurements. Based upon this, it is proposed to augment HRTFs by creating new HRTFs with equivalent binaural and monaural cues, but without using the original HRTF which contains the convolutive noise.

This was achieved by interpolation between the HRTFs which straddle the target HRTF on the horizontal plane. For example, in a measurement set where the horizontal plane is sampled by 10° increments, supposing it is desired to augment the HRTF in the position $\varphi = 30°$. This would be achieved by interpolating between the HRTFs found at $\varphi = 20°$ and $\varphi = 40°$.

This interpolation is done in the frequency domain, applying linear interpolation to the magnitudes and phase of the signal.

$$\begin{aligned} |\mathrm{HRTF}_{\mathrm{aug1}}[\omega, 30°]| &= \frac{|\mathrm{HRTF}[\omega, 20°]| + |\mathrm{HRTF}[\omega, 40°]|}{2} \\ \angle\mathrm{HRTF}_{\mathrm{aug1}}[\omega, 30°] &= \frac{\angle\mathrm{HRTF}[\omega, 20°] + \angle\mathrm{HRTF}[\omega, 40°]}{2} \end{aligned} \tag{149}$$

This could then be continued to higher orders, by interpolating the next two most distant positions on the horizontal plane; in this example being 10° and 50°.

$$\begin{aligned} |\mathrm{HRTF}_{\mathrm{aug2}}[\omega, 30^\circ]| &= \frac{|\mathrm{HRTF}[\omega, 10^\circ]| + |\mathrm{HRTF}[\omega, 50^\circ]|}{2} \\ \angle\mathrm{HRTF}_{\mathrm{aug2}}[\omega, 30^\circ] &= \frac{\angle\mathrm{HRTF}[\omega, 10^\circ] + \angle\mathrm{HRTF}[\omega, 50^\circ]}{2} \end{aligned} \tag{150}$$

To test the efficacy of this augmentation, new sets of HRTFs are created using the proposed augmentation technique for every existing HRTF measurement set, up to an order of 7, being the maximum possible with the number of HRTF sets available. These were then used in conjunction with the original measurements to create training dataset of 2, 4, and 8 sets of HRTFs made from one original set of measurements.

Results

Cross Performance Matrices are presented individually for 2, 4 and 8 HRTFs in Figs. 116-118. Additionally, average performance when evaluated on mismatched testing dataset is shown in Table 82.

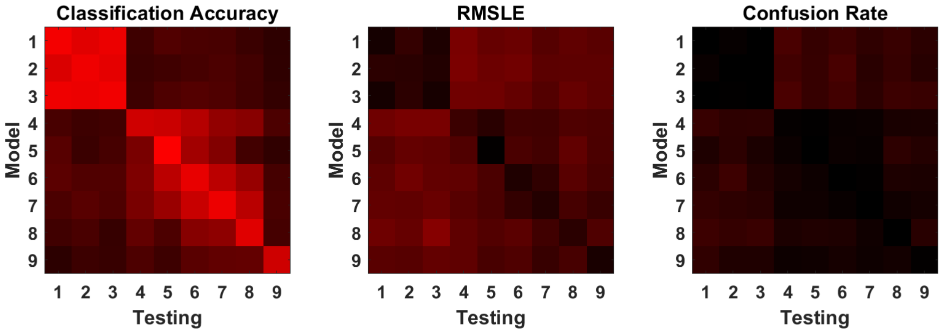

Figure 116: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF augmented once with Horizontal Plane Interpolation

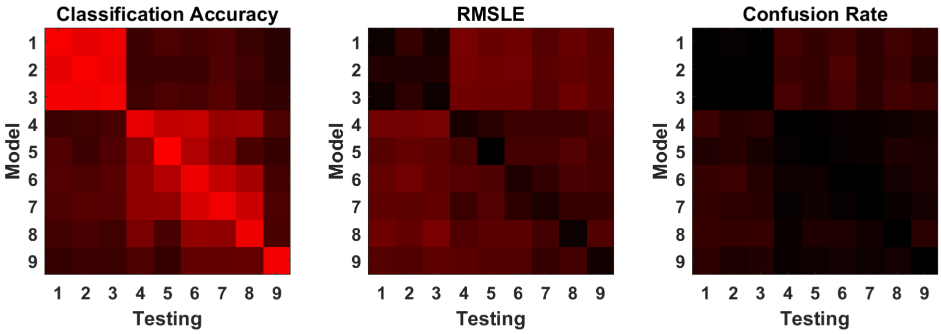

Figure 117: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF augmented three times with Horizontal Plane Interpolation

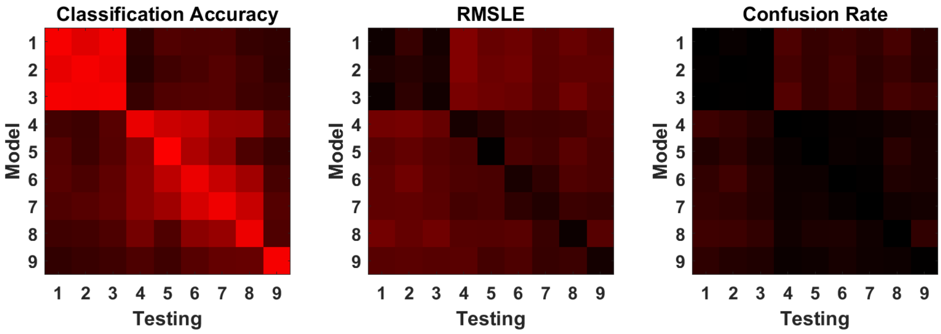

Figure 118: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF augmented seven times with Horizontal Plane Interpolation

Table 82: Average performance for CNNs trained with horizontal plane interpolation

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
Set of 2	**Accuracy**	45.2%	44.8%	44.1%	37.1%	45.7%	50.2%	47.1%	42.0%	25.1%	*42.3%*
	RMSLE	53.6°	58.9°	59.3°	67.5°	54.7°	55.9°	45.3°	57.6°	51.8°	*56.0°*
	Confusion	14.7%	14.2%	11.8%	14.9%	11.8%	13.9%	9.1%	15.1%	14.2%	*13.3%*
Set of 4	**Accuracy**	45.1%	45.0%	46.4%	41.1%	43.7%	51.9%	48.8%	44.8%	25.5%	*43.6%*
	RMSLE	52.5°	57.2°	55.9°	60.6°	57.9°	55.7°	44.9°	52.8°	49.8°	*54.2°*
	Confusion	14.7%	13.5%	11.4%	12.3%	11.7%	13.9%	9.3%	13.8%	14.1%	*12.7%*
Set of 8	**Accuracy**	45.9%	44.3%	49.6%	38.0%	46.4%	52.4%	50.0%	44.3%	27.4%	*44.2%*
	RMSLE	52.0°	59.2°	49.9°	64.3°	54.1°	55.7°	44.1°	54.2°	52.5°	*54.0°*
	Confusion	15.2%	15.5%	11.0%	14.5%	11.5%	13.5%	9.6%	15.6%	14.5%	*13.4%*

Discussion

Looking at Table 82, little improvement in performance is seen as the number of augmentations is increased. This suggests no discernible improvement to generalisation when using this augmentation technique. Based on this, this approach is not recommended. This also suggests the hypothesis that convolutive noise in single measurements is the cause of a lack of generalisation may not be correct.

7.5.3 Vertical Plane Interpolation

One possible cause of the lack of improvement in the horizontal-plane interpolation method is that all of these positions are eventually still included in the training dataset.

To test this, another similar augmentation scheme was proposed, in which instead of interpolation on the horizontal plane, the multi-dimensional nature of these datasets were leveraged, and instead the DoA representing next larger and smaller elevation to 0° at the desired azimuth was chosen.

$$\begin{aligned} |\mathrm{HRTF}_{\mathrm{aug1}}[\omega, 0, \varphi]| &= \frac{|\mathrm{HRTF}[\omega, \vartheta, \varphi]| + |\mathrm{HRTF}[\omega, -\vartheta, \varphi]|}{2} \\ \angle\mathrm{HRTF}_{\mathrm{aug1}}[\omega, 0, \varphi] &= \frac{\angle\mathrm{HRTF}[\omega, \vartheta, \varphi] + \angle\mathrm{HRTF}[\omega, -\vartheta, \varphi]}{2} \end{aligned} \tag{151}$$

where ϑ is some offset to θ based on the next available θ value in the sampling grid. This was also repeated for multiple orders, creating sets of [2, 4, 8] unique sets of HRTFs per original measurement set.

Results

Cross Performance Matrices are presented individually for 2 and 4 HRTFs in Figs. 119-120. Additionally, average performance when evaluated on mismatched testing datasets is shown in Table 83.

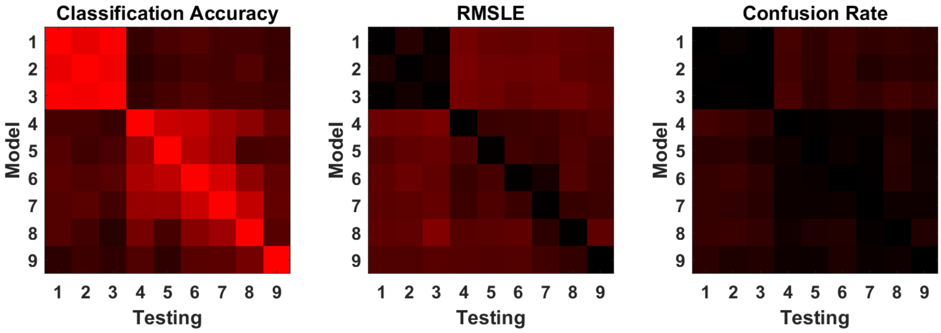

Figure 119: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF augmented once with Vertical Plane Interpolation

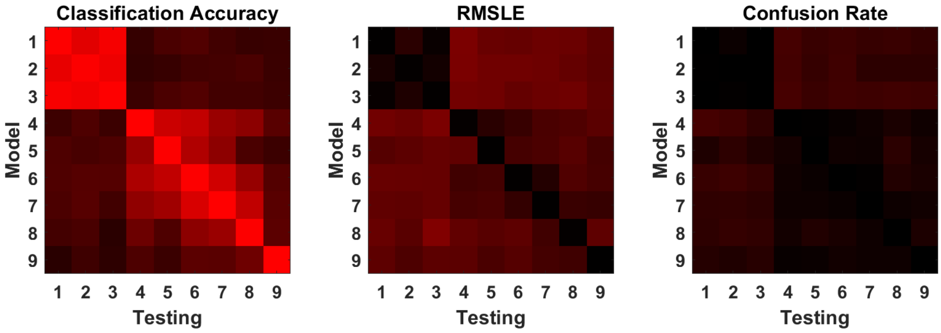

Figure 120: Mismatched Anechoic Condition Cross Performance matrices for CNN Trained on HRTF augmented three times with Vertical Plane Interpolation

Table 83: Average performance of CNN trained on dataset augmented with the vertical plane interpolation technique

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	***Avg.***
Set of 2	**Accuracy**	47.7%	45.9%	43.9%	42.2%	43.9%	52.2%	49.7%	44.7%	32.4%	*44.7%*
	RMSLE	47.5°	52.3°	54.2°	59.4°	57.5°	54.5°	46.7°	52.8°	52.9°	*53.1°*
	Confusion	15.6%	15.1%	11.9%	13.9%	10.4%	14.0%	8.4%	14.6%	12.0%	*12.9%*
Set of 4	**Accuracy**	44.8%	46.7%	43.5%	41.4%	45.4%	52.8%	47.8%	43.7%	30.2%	*44.0%*
	RMSLE	51.1°	51.8°	55.6°	64.7°	55.3°	54.9°	52.6°	56.4°	53.5°	*55.1°*
	Confusion	14.0%	15.5%	12.9%	13.9%	11.7%	13.1%	10.8%	14.5%	13.1%	*13.3%*

Discussion

It can be seen in Table 83 that the vertical plane approach offers no improvement over the horizontal plane approach. This confirms the negative result, that this interpolation based approach to HRTF augmentation is not suitable for improving generalisation of CNN for BSSL in the mismatched anechoic condition.

7.6 Conclusion

In this chapter, a short study has been conducted on the efficacy of binaural DoA-estimation using CNNs. It was found that CNNs trained on data made with a single HRTF measurement set, generalise very poorly to other HRTF measurement sets. It was shown that this could be mitigated by increasing the number of measurement sets, however this would require a similar round-robin set of measurements to be conducted on other binaural arrays to replicate.

A method of altering the HRTFs was proposed, being to find and remove the common part of the transfer function. This did not achieve an improvement.

Following this some methods of augmenting HRTFs were proposed; creating new positions with horizontal plane mirroring, horizontal plane interpolation and vertical plane interpolation. Of these methods, horizontal plane interpolation provided the only increase in performance, however this method is limited in that for each measurement set only one augmentation can be achieved.

The mismatched anechoic condition is a challenging barrier in deep-DoA estimation, which has typically been ignored in previous work in the field.

8 Conclusions and Further Work

This work has looked at Binaural direction of arrival (DoA) estimation by use of convolutional neural networks (CNNs), based on the trend towards this approach in the field of binaural sound source localisation (BSSL). To assess the suitability of this approach, and to move towards more optimal implementation of CNNs in generalised scenarios, a series of experiments was performed analysing localisation performance in scenarios identified as significant in a literature review.

- Localisation of single speaker
- DoA Estimation on the full horizontal plane
- Localisation in the presence of the reverberation of unseen rooms
- Localisation in the presence of unseen noise mixtures
- Localisation which can be generalised beyond single sets of head-related transfer function (HRTF) measurements

Based on this, experimental work was carried out on the following areas.

Localising in Simulated Acoustic Environments

A study of how different types of acoustic environments affect a simple CNN trained on magnitude of short-time fourier transform (STFT) domain binaural audio was undertaken. These tests included diffuse noise, interfering sound sources, additive noise and simulation of it,the mismatched HRTF condition, and reverberation time.

A significant pattern is identified, in which some types of acoustic degradation, particularly those which can be considered additive noise, cause some reduction in performance, while some acoustic conditions pose significant generalisation issues.

It was found here that CNNs tasked with binaural DoA estimation generalise very poorly to unknown rooms, and this is a more significant challenge than the actual level of reverberation.

Another notable conclusion here is that binaural room impulse responses (BRIRs) synthesised through image source method (ISM) work as a good proxy for performance of real measured BRIRs.

Feature Representations

It was identified that previously little attention has been placed on different types of magnitude-frequency domain feature representations, and the significance of changing the approach.

Through comparative analysis, it was found that using log-spectrograms is a reasonable approach, as compared to cepstra. Additionally it was found that the type of filterbank used in STFTs is of little significance. Attention was then placed on phase and time based feature representations, following a similar technique of comparative analysis between different approaches in controlled conditions.

The significant findings were that unwrapping phase does not improve performance, that phase is significantly preferred to cross-correlation derived representations, and that 2D phase matrices are preferred over interaural phase difference (IPD) matrices. It was also found that the magnitude of STFT domain binaural audio is the salient representation, not phase.

Deep Learning Architectures

It was tested whether 1D or 2D convolutional layers, and their correspondingly shaped inputs, are preferred.

It was shown here that 2D representations are capable of achieving better performance. Following this the increasing trend towards convolutional recurrent neural network (CRNN) was scrutinised, as it had not yet been established in literature whether the introduction of recurrent layers is capable of improving performance in BSSL. It was found that that this is the case, with CRNNs being able to outperform CNNs.

Building on this another comparison was performed investigating the preferred type of recurrent layer for use in BSSL. Out of long short-term memory (LSTM), bidirectional long short-term memory (BiLSTM), gated recurrent unit (GRU) and bidirectional gated recurent unit (BiGRU), a soft preference for BiLSTM is established.

Mismatched Anechoic Condition

It was identified that much of the previous work on BSSL evaluates systems using the

HRTFs from the measurement set as the training data, which is potentially hiding a large generalisation issue.

The heavy degree to which CNNs are not able to generalise to other measurement sets was established, and some methods of dealing with this were proposed beyond increasing the number of HRTF measurements: removal of the common transfer function (CTF) from the HRTFs, augmentation by mirroring of source positions, and two novel augmentation schemes where new HRTFs are created through selective interpolation.

It was shown that data plurality is still the most significant factor towards robust localisation under the mismatched anechoic condition, but that some improvement through augmentation is possible.

The total of these findings advances understanding towards robust binaural DoA estimation, as overlooked and ignored issues of generalisation in CNN based DoA estimation are now better understood.

The main conclusion taken from the research in this thesis is that deep learning based binaural DoA estimation, as presented in previous publications where simulated data is used as a proxy for real-world implementation, is fundamentally limited by the bottleneck of generalising to conditions otherwise unseen in the datasets. Improvement towards this aim can be made through more careful consideration of the training data, feature representations, and the model design; it is the recommendation of this thesis that future work in the pursuit of robust binaural DoA estimation concentrates on utilising realistic binaural datasets.

8.1 Areas for Future Work

This work establishes the importance of two large and under-explored generalisation issues encountered while using convolutional neural networks for binaural sound source localisation: generalising to unknown room impulse responses, and to unknown measurements of a known binaural array (the mismatched anechoic condition). In both cases, the most effective solution to this has been shown to be increasing the number of known conditions in the dataset, however this comes at the expense of increased data complexity which may lead to a general worsening of performance.

A recommended area of future work is to further explore signal processing approaches to mitigate these factors in the dataset. Humans ability to localise sound in reverberant environments is aided through an ability to differentiate between a direct path's wavefront and the reflections, in what is known as the precedence effect. A plausible hypothesis is that precedence effect modelling can help mitigate the generalisation issue; which needs to be tested.

Solutions to mitigate the issue of generalising to unknown measurements sets were proposed in Chapter 7, however most of these results were negative. A clear area for future work is the proposal of more augmentation methods which could be used in aid of this scenario. A reasonable hypothesis given the behaviour found in this section, is that a set of equalisation filters which could be applied to HRTF sets could aid in this.

A shortcoming of the framework applied in this work is that sound sources are always presumed to be non-moving. This is an unrealistic assumption for real sound sources. Moving sound sources have been considered in previous work, but there has not been significant effort in making these trajectories realistic. Another possible area for future work would be the realistic modelling of these trajectories, and investigating of relevant effects of this.

A criticism of this work is that it has concentrated solely on trends existing in the niche field of BSSL, ignoring more general trends seen in deep learning in general, and applied to other audio fields. The sudden increase in works applying CNNs to BSSL in 2019 came eight years after the publication of AlexNet (Krizhevsky *et al.*, 2012) and the popularity of CNNs in other fields. In 2017 transformers were introduced (Vaswani, 2017), and have been getting introduced to new applications since then; following the same trend, it would be reasonable to assume that transformer based architectures will also come to dominate the field of BSSL. There is evidence of this process: a transformer based model has achieved best localisation results in the sound localisation challenge in detection and classification of acoustic scenes and events (DCASE) for 2022-2024 (Wang *et al.*, 2022). A now important study is to test the application of transformer based architectures to BSSL.

This thesis concentrates on BSSL under other conditions: matched binaural array, on the horizontal plane, and of speech. A useful continuation of this work would be to apply the same testing framework to BSSL in the mismatched-HRTF condition, to median plane or full spherical DoA estimation, and finally to sound sources which are not speech.

8.2 Closing Remarks

The research presented in this thesis investigates use of CNN architectures for the task of BSSL. This is achieved by identifying and scrutinising, and expanding upon current trends on existing literature on the task of BSSL. This approach gives insight into the task of BSSL from the perspective of the data, which is not necessarily present in the more typical literature on the topic where novel systems are introduced and evaluated. Conclusions drawn in this thesis can be used for more effective study of BSSL in future work.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., *et al.* (2016). "{TensorFlow}: a system for {Large-Scale} machine learning". In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283.

Abbagnaro, L. A., Bauer, B. B., and Torick, E. L. (1975). "Measurements of diffraction and interaural delay of a progressive sound wave caused by the human head. II". In: *The Journal of the Acoustical Society of America* 58.3, pp. 693–700.

Adavanne, S., Politis, A., and Virtanen, T. (2018). "Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network". In: *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1462–1466.

AES69-2022: AES standard for file exchange - Spatial acoustic data file format (2022). Standard. Audio Engineering Society.

Algazi, V., Duda, R., Thompson, D., and Avendano, C. (2001). "The CIPIC HRTF database". In: *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, pp. 99–102.

Alinaghi, A., Jackson, P. J., Liu, Q., and Wang, W. (2014). "Joint Mixing Vector and Binaural Model Based Stereo Source Separation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.9, pp. 1434–1448.

Alinaghi, A., Wang, W., and Jackson, P. J. (2013). "Spatial and coherence cues based time-frequency masking for binaural reverberant speech separation". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, pp. 684–688.

Andén, J. and Mallat, S. (2014). "Deep Scattering Spectrum". In: *IEEE Transactions on Signal Processing* 62.16, pp. 4114–4128.

Andreopoulou, A., Begault, D. R., and Katz, B. F. G. (2015). "Inter-Laboratory Round Robin HRTF Measurement Comparison". In: *IEEE Journal of Selected Topics in Signal Processing* 9.5, pp. 895–906.

Ansari, S., Alatrany, A. S., Alnajjar, K. A., Khater, T., Mahmoud, S., Al-Jumeily, D., and Hussain, A. J. (2023). "A survey of artificial intelligence approaches in blind source separation". In: *Neurocomputing* 561, p. 126895.

Arend, J. M., Neidhardt, A., and Pörschmann, C. (2016). “Measurement and Perceptual Evaluation of a Spherical Near-Field HRTF Set (Messung und perzeptive Evaluierung eines sphärischen Satzes von Nahfeld-HRTFs)”. In: *29th Tonmeistertagung*.

ARI HRTF-Database (2011). URL: `https://www.oeaw.ac.at/isf/das-institut/software/hrtf-database` 03/19/2024/03/19/2024/03/19/2024.

Armstrong, C., Chadwick, A., Thresh, L., Murphy, D., and Kearney, G. (2017). “Simultaneous HRTF Measurement of Multiple Source Configurations Utilizing Semi-Permanent Structural Mounts”. In: *Audio Engineering Society Convention 143*. Audio Engineering Society.

Armstrong, C., Thresh, L., Murphy, D., and Kearney, G. (2018). “A Perceptual Evaluation of Individual and Non-Individual HRTFs: A Case Study of the SADIE II Database”. In: *Applied Sciences* 8.11.

Ashida, G. and Carr, C. E. (2011). “Sound localization: Jeffress and beyond”. In: *Current Opinion in Neurobiology* 21.5. Networks, circuits and computation, pp. 745–751.

Bacila, B. and Lee, H. (2023). “Perceptual dimensions of listener envelopment (LEV) in a positional and directional-varying context”. In: *Audio Engineering Society Conference: AES 2023 International Conference on Spatial and Immersive Audio*.

Bacila, B. I. and Lee, H. (2019). “360 binaural room impulse response (BRIR) database for 6DOF spatial perception research”. In: *Audio Engineering Society Convention 146*. Audio Engineering Society.

Bernschütz, B. (2013). “A spherical far field HRIR/HRTF compilation of the Neumann KU 100”. In: *Proceedings of the 40th Italian (AIA) annual conference on acoustics and the 39th German annual conference on acoustics (DAGA) conference on acoustics*. Vol. 29. German Acoustical Society (DEGA) Berlin.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Vol. 4. 4. New York: Springer.

Blauert, J. (1997). *Spatial hearing : the psychophysics of human sound localization*. eng. Rev. ed. Cambridge, Mass: MIT Press.

Bomhardt, R., Fuente Klein, M. de la, and Fels, J. (2017). “A high-resolution head-related transfer function and three-dimensional ear model database”. In: *Proceedings of Meetings on Acoustics* 29.1, p. 050002.

Boren, B. (2017). “Immersive Sound”. In: chap. History of 3D Sound, pp. 40–62.

Bregman, A. S. (1994). *Auditory scene analysis: The perceptual organization of sound.* MIT press.

Brinkmann, F., Dinakaran, M., Pelzer, R., Grosche, P., Voss, D., and Weinzierl, S. (2019). “A Cross-Evaluated Database of Measured and Simulated HRTFs Including 3D Head Meshes, Anthropometric Features, and Headphone Impulse Responses”. In: *J. Audio Eng. Soc* 67.9, pp. 705–718.

Brinkmann, F., Kreuzer, W., Thomsen, J., Dombrovskis, S., Pollack, K., Weinzierl, S., and Majdak, P. (2023). “Recent Advances in an Open Software for Numerical HRTF Calculation”. In: *J. Audio Eng. Soc* 71.7/8, pp. 502–514.

Brinkmann, F., Lindau, A., Weinzierl, S., Geissler, G., Par, S. van de, Müller-Trapet, M., Opdam, R., and Vorländer, M. (2017). “The FABIAN head-related transfer function data base”. In.

Bruschi, V., Nobili, S., Cecchi, S., and Piazza, F. (2020). “An innovative method for binaural room impulse responses interpolation”. In: *Audio Engineering Society Convention 148*. Audio Engineering Society.

Carpentier, T., Bahu, H., Noisternig, M., and Warusfel, O. (2014). “Measurement of a head-related transfer function database with high spatial resolution”. In: *7th Forum Acusticum(EAA)*. Krakow, Poland.

Carr, C. and Konishi, M. (1990). “A circuit for detection of interaural time differences in the brain stem of the barn owl”. In: *Journal of Neuroscience* 10.10, pp. 3227–3246.

Chakrabarty, S. and Habets, E. A. P. (2019). “Multi-Speaker DOA Estimation Using Deep Convolutional Networks Trained With Noise Signals”. In: *IEEE Journal of Selected Topics in Signal Processing* 13.1, pp. 8–21.

Chen, Z. and Hohmann, V. (2015). “Online Monaural Speech Enhancement Based on Periodicity Analysis and A Priori SNR Estimation”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.11, pp. 1904–1916.

Cherry, E. C. (1953). “Some Experiments on the Recognition of Speech, with One and with Two Ears”. In: *The Journal of the Acoustical Society of America* 25.5, pp. 975–979.

Cieciura, C., Volino, M., and Jackson, P. J. (2023). “SurrRoom 1.0 Dataset: Spatial Room Capture with Controlled Acoustic and Optical Measurements”. In: *Audio Engineering Society Convention 154*. Audio Engineering Society.

Comon, P. (1994). “Independent component analysis, a new concept?” In: *Signal processing* 36.3, pp. 287–314.

Courtois, G., Marmaroli, P., Lindberg, M., Oesch, Y., and Balande, W. (2014). “Implementation of a Binaural Localization Algorithm in Hearing Aids: Specifications and Achievable Solutions”. In: *Audio Engineering Society Convention 136*.

Crnigoj, S. C. (2020). “Proposing Factors towards a Standardised Testing Environment for Binaural and 3D Sound Systems”. PhD thesis. Liverpool John Moores University, United Kingdom.

Cummins, F., Grimaldi, M., Leonard, T., and Simko, J. (2006). “The chains speech corpus: Characterizing individual speakers”. In: *Proc of SPECOM*, pp. 1–6.

Dávila-Chacón, J., Liu, J., and Wermter, S. (2018). “Enhanced robot speech recognition using biomimetic binaural sound source localization”. In: *IEEE transactions on neural networks and learning systems* 30.1, pp. 138–150.

De Sena, E., Hacıhabiboğlu, H., Cvetković, Z., and Smith, J. O. (2015). “Efficient synthesis of room acoustics via scattering delay networks”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.9, pp. 1478–1492.

De Sena, E., Hacihabiboglu, H., and Cvetkovic, Z. (2011). “Scattering delay network: An interactive reverberator for computer games”. In: *Audio Engineering Society Conference: 41st International Conference: Audio for Games*. Audio Engineering Society.

Denk, F., Ernst, S., Heeren, J., Ewert, S. D., and Kollmeier, B. (2018). “The oldenburg hearing device (olhead) hrtf database”. In: *University of Oldenburg, Tech. Rep.*

Desai, D. and Mehendale, N. (2022). “A review on sound source localization systems”. In: *Archives of Computational Methods in Engineering* 29.7, pp. 4631–4642.

Doclo, S., Gannot, S., Moonen, M., and Spriet, A. (2010). “Acoustic beamforming for hearing aid applications”. In: *Handbook on array processing and sensor networks*, pp. 269–302.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.* (2020). “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929*.

Duchi, J., Hazan, E., and Singer, Y. (2011). “Adaptive subgradient methods for online learning and stochastic optimization.” In: *Journal of machine learning research* 12.7.

Duda, R. O., Algazi, V. R., and Thompson, D. M. (2002). “The Use of Head-and-Torso Models for Improved Spatial Sound Synthesis”. In: *Journal of The Audio Engineering Society*.

Durlach, N., Thompson, C., and Colburn, H. (1981). “Binaural interaction in impaired listeners: A review of past research”. In: *Audiology* 20.3, pp. 181–211.

Engel, I., Daugintis, R., Vicente, T., Hogg, A. O., Pauwels, J., Tournier, A. J., and Picinali, L. (2023). “The sonicom hrtf dataset”. In: *Journal of the Audio Engineering Society* 71.5, pp. 241–253.

Erbes, V., Geier, M., Weinzierl, S., and Spors, S. (2015). “database of single-channel and binaural room impulse responses of a 64-channel loudspeaker array”. In: *journal of the audio engineering society* 189.

Estival, D., Cassidy, S., Cox, F., and Burnham, D. (2014). “AusTalk: an audio-visual corpus of Australian English”. In.

Everest, F. A. (2022). *Master handbook of acoustics*. 7th Edition. New York: McGraw Hill.

Farkaš, T. (2018). “Binaural and ambisonic sound as the future standard of digital games”. In: *Acta Ludologica* 1.2, pp. 34–46.

Fazi, F. M. and Hamdan, E. (2018). “Stage compression in transaural audio”. In: *Audio Engineering Society Convention 144*. Audio Engineering Society.

Francombe, J. (2017). *IoSR Listening Room Multichannel BRIR dataset*.

Froehlich, M., Freels, K., and Powers, T. A. (2015). “Speech recognition benefit obtained from binaural beamforming hearing aids: Comparison to omnidirectional and individuals with normal hearing”. In: *Audiology Online* 14338, pp. 1–8.

García-Barrios, G., Krause, D. A., Politis, A., Mesaros, A., Gutiérrez-Arriola, J. M., and Fraile, R. (2022). “Binaural source localization using deep learning and head rotation information”. In: *2022 30th European Signal Processing Conference (EUSIPCO).* IEEE, pp. 36–40.

Garcia-Gomez, V. and Lopez, J. J. (2018). “Binaural room impulse responses interpolation for multimedia real-time applications”. In: *Audio Engineering Society Convention 144.* Audio Engineering Society.

Gardner, B., Martin, K., *et al.* (1994). “HRFT Measurements of a KEMAR Dummyhead Microphone”. In.

Garnerin, M., Rossato, S., and Besacier, L. (2021). “Investigating the Impact of Gender Representation in ASR Training Data: a Case Study on Librispeech”. In: *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing.* Online: Association for Computational Linguistics, pp. 86–92.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallett, D., Dahlgren, N., and Zue, V. (1992). “TIMIT Acoustic-phonetic Continuous Speech Corpus”. In: *Linguistic Data Consortium.*

Garofolo, J. S. (1993). “Timit acoustic phonetic continuous speech corpus”. In: *Linguistic Data Consortium, 1993.*

Gelfand, S. A. and Calandruccio, L. (2009). *Essentials of audiology.*

Geva, G., Warusfel, O., Dubnov, S., Dubnov, T., Amedi, A., and Hel-Or, Y. (2024). “Binaural Sound Source Localization Using a Hybrid Time and Frequency Domain Model”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8821–8825.

Goli, P. and Par, S. van de (2023). “Deep Learning-Based Speech Specific Source Localization by Using Binaural and Monaural Microphone Arrays in Hearing Aids”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31, pp. 1652–1666.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* Cambridge, MA: MIT Press.

Greenwood, D. D. (1961). "Critical bandwidth and the frequency coordinates of the basilar membrane". In: *The Journal of the Acoustical Society of America* 33.10, pp. 1344–1356.

Grothe, B., Pecka, M., and McAlpine, D. (2010). "Mechanisms of Sound Localization in Mammals". In: *Physiological Reviews* 90.3. PMID: 20664077, pp. 983–1012.

Grumiaux, P.-A., Kitić, S., Girin, L., and Guérin, A. (2022). "A survey of sound source localization with deep learning methods". In: *The Journal of the Acoustical Society of America* 152.1, pp. 107–151.

Gulli, A. and Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.

"Gun? How 63 German guns were located by sound waves alone in single day" (1918). In: *Popular Science monthly*, p. 39.

Hammond, B. R. (2021). "Methods for Robust Binaural Sound Source Localisation". PhD thesis. University of Surrey.

Hiipakka, M., Tikander, M., and Karjalainen, M. (2010). "Modeling the external ear acoustics for insert headphone usage". In: *Journal of the Audio Engineering Society* 58.4, pp. 269–281.

Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hogg, A. O. T., Jenkins, M., Liu, H., Squires, I., Cooper, S. J., and Picinali, L. (2024). "HRTF Upsampling With a Generative Adversarial Network Using a Gnomonic Equiangular Projection". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32, pp. 2085–2099.

Hu, H., Zhou, L., Ma, H., and Wu, Z. (2008). "HRTF personalization based on artificial neural network in individual virtual auditory space". In: *Applied Acoustics* 69.2, pp. 163–172.

Hummersone, C., Mason, R., and Brookes, T. (2010). "Dynamic Precedence Effect Modeling for Source Separation in Reverberant Environments". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.7, pp. 1867–1871.

Hussain, R. (2020). "Augmented reality based middle and inner ear surgical procedures". PhD thesis. Université Bourgogne Franche-Comté.

ITU-R BS.1116: Methods for the subjective assessment of small impairments in audio systems (2015). Standard. British Standards Institution (BSI).

ITU-R BS.2051: Advanced sound system for programme production (2022). Standard. British Standards Institution (BSI).

Jeffress, L. A. (1948). “A place theory of sound localization.” In: *Journal of comparative and physiological psychology* 41.1, p. 35.

Jeub, M., Schafer, M., and Vary, P. (2009). “A binaural room impulse response database for the evaluation of dereverberation algorithms”. In: *2009 16th International Conference on Digital Signal Processing*. IEEE, pp. 1–5.

Jiang, S., Wu, L., Yuan, P., Sun, Y., and Liu, H. (2020). “Deep and CNN fusion method for binaural sound source localisation”. In: *The Journal of Engineering* 2020.13, pp. 511–516.

Johannesma, P. (1972). “The pre-response stimulus ensemble of neurons in the cochlear nucleus”. In: *Symposium on Hearing Theory, 1972*. IPO.

Kabal, P. (2002). “TSP speech database”. In: *McGill University, Database Version* 1.0, pp. 09–02.

Kahana, Y. (2000). “Numerical modelling of the head-related transfer function”. PhD thesis. University of Southampton.

Katz, B. and Begault, D. (2007). “Round robin comparison of HRTF measurement systems: preliminary results”. In: pp. 1–6.

Kayser, H., Ewert, S. D., Anemüller, J., Rohdenburg, T., Hohmann, V., and Kollmeier, B. (2009). “Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses”. In: *EURASIP Journal on advances in signal processing* 2009, pp. 1–10.

Keyrouz, F., Naous, Y., and Diepold, K. (2006). “A New Method for Binaural 3-D Localization Based on Hrtfs”. In: *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. Vol. 5, pp. V–V.

Keyrouz, F. (2011). “Humanoid hearing: A novel three-dimensional approach”. In: *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pp. 214–219.

Keyrouz, F. (2014). “Advanced binaural sound localization in 3-D for humanoid robots”. In: *IEEE Transactions on Instrumentation and Measurement* 63.9, pp. 2098–2107.

Keyrouz, F. and Diepold, K. (2006). “An Enhanced Binaural 3D Sound Localization Algorithm”. In: *2006 IEEE International Symposium on Signal Processing and Information Technology*, pp. 662–665.

Kingma, D. P. and Ba, J. (2014). “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980*.

Köbler, S. and Rosenhall, U. (2002). “Horizontal localization and speech intelligibility with bilateral and unilateral hearing aid amplification: Localización horizontal y discriminación del lenguaje con adaptación unilateral y bilateral de auxiliares auditivos”. In: *International journal of audiology* 41.7, pp. 395–400.

Krause, D. A., García-Barrios, G., Politis, A., and Mesaros, A. (2024a). “Binaural Sound Source Distance Estimation and Localization for a Moving Listener”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32, pp. 996–1011.

Krause, D. A., Politis, A., and Mesaros, A. (2024b). “Sound Event Detection and Localization with Distance Estimation”. In: *arXiv preprint arXiv:2403.11827*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25.

Kuhn, G. F. (1977). “Model for the interaural time differences in the azimuthal plane”. In: *the Journal of the Acoustical Society of America* 62.1, pp. 157–167.

Laitinen, M.-V., Disch, S., and Pulkki, V. (2013). “sensitivity of human hearing to changes in phase spectrum”. In: *journal of the audio engineering society* 61 (11), pp. 860–877.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, D. D. and Seung, H. S. (1999). “Learning the parts of objects by non-negative matrix factorization”. In: *nature* 401.6755, pp. 788–791.

Li, Y., Wang, L., and Reiss, J. (2025). “Binaural room impulse responses interpolation using physics-informed neural networks in three dimensions”. In: 51st German Annual Conference on Acoustics (DAGA).

Li, Y., Cardenuto, J., Di Giusto, F., Preihs, S., and Peissig, J. (2023). "A near-field Head-Related Transfer Function data set of KEMAR with high distance resolution and multiple elevations". In: *Audio Engineering Society.*

Liaquat, M. U., Munawar, H. S., Rahman, A., Qadir, Z., Kouzani, A. Z., and Mahmud, M. P. (2021). "Localization of sound sources: A systematic review". In: *Energies* 14.13, p. 3910.

Lovedee-Turner, M. and Murphy, D. (2018). "Application of Machine Learning for the Spatial Analysis of Binaural Room Impulse Responses". In: *Applied Sciences* 8.1.

Lyon, R. F. (2017). *Human and machine hearing: extracting meaning from sound.* Cambridge University Press.

Ma, N., Brown, G. J., and Gonzalez, J. A. (2015a). "Exploiting top-down source models to improve binaural localisation of multiple sources in reverberant environments." In: *INTERSPEECH 2015: Speech beyond Speech.* ISCA, pp. 160–164.

Ma, N., Brown, G. J., and May, T. (2015b). "Exploiting deep neural networks and head movements for binaural localisation of multiple speakers in reverberant conditions". In: *INTERSPEECH 2015: Speech beyond Speech.* ISCA, pp. 3302–3306.

Ma, N., May, T., and Brown, G. J. (2017). "Exploiting Deep Neural Networks and Head Movements for Robust Binaural Localization of Multiple Sources in Reverberant Environments". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.12, pp. 2444–2453.

Ma, N., May, T., Wierstorf, H., and Brown, G. J. (2015c). "A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 2699–2703.

Macpherson, E. A. (1991). "A computer model of binaural localization for stereo imaging measurement". In: *Journal of the Audio Engineering Society* 39.9, pp. 604–622.

Maijala, P. (1997). "Better binaural recordings using the real human head". In: *INTERNOISE.* Vol. 2. NOISE CONTROL FOUNDATION, pp. 1135–1138.

Majdak, P., Hollomey, C., and Baumgartner, R. (2022). "AMT 1. x: A toolbox for reproducible research in auditory modeling". In: *Acta Acustica* 6, p. 19.

Mandel, M. (2013). *Matlab Recti-Linear Room Simulator*. `https://github.com/mim/rlrs`.

Marschall, M., Bolaños, J. G., Prepeliță, S. T., and Pulkki, V. (2023). "A database of near-field head-related transfer functions based on measurements with a laser spark source". In: *Applied Acoustics* 203, p. 109173.

Massicotte, P., Chaoui, H., and Ouameur, M. A. (2022). "LSTM with Scattering Decomposition-Based Feature Extraction for Binaural Sound Source Localization". In: *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, pp. 436–440.

May, T., Ma, N., and Brown, G. J. (2015). "Robust localisation of multiple speakers exploiting head movements and multi-conditional training of binaural cues". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2679–2683.

May, T., Par, S. van de, and Kohlrausch, A. (2011). "A Probabilistic Model for Robust Localization Based on a Binaural Auditory Front-End". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.1, pp. 1–13.

McCormack, L., Meyer-Kahlen, N., Alon, D. L., Ben-Hur, Z., Garı, S. V. A., and Robinson, P. (2023). "Six-degrees-of-freedom binaural reproduction of head-worn microphone array capture". In: *Journal of the Audio Engineering Society* 71.10, pp. 638–649.

McPherson, D. R. (2018). "Sensory Hair Cells: An Introduction to Structure and Physiology". In: *Integrative and Comparative Biology* 58.2, pp. 282–300.

Miccini, R. and Spagnol, S. (2020). "HRTF Individualization using Deep Learning". In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 390–395.

El-Mohandes, A. M., Zandi, N. H., and Zheng, R. (2023). "DeepBSL: 3-D Personalized Deep Binaural Sound Localization on Earable Devices". In: *IEEE Internet of Things Journal* 10.21, pp. 19004–19013.

Moore, B. C. and Glasberg, B. R. (1983). "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns". In: *The journal of the acoustical society of America* 74.3, pp. 750–753.

Murphy, D., Kelloniemi, A., Mullen, J., and Shelley, S. (2007). “Acoustic modeling using the digital waveguide mesh”. In: *IEEE Signal Processing Magazine* 24.2, pp. 55–66.

Najar, F., Bourouis, S., Bouguila, N., and Belghith, S. (2017). “A comparison between different gaussian-based mixture models”. In: *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, pp. 704–708.

Nelson, P. and Elliott, S. (1991). *Active control of sound*. Academic Press.

Nguyen, Q., Girin, L., Bailly, G., Elisei, F., and Nguyen, D.-C. (2018). “Autonomous sensorimotor learning for sound source localization by a humanoid robot”. In: *IROS 2018-Workshop on Crossmodal Learning for Intelligent Robotics in conjunction with IEEE/RSJ IROS*.

Nikunen, J. and Virtanen, T. (2014). “Direction of arrival based spatial covariance model for blind sound source separation”. In: *IEEE/ACM transactions on audio, speech, and language processing* 22.3, pp. 727–739.

Noble, W. and Byrne, D. (1990). “A comparison of different binaural hearing aid systems for sound localization in the horizontal and vertical planes”. In: *British Journal of Audiology* 24.5, pp. 335–346.

Noble, W. and Gatehouse, S. (2006). “Effects of bilateral versus unilateral hearing aid fitting on abilities measured by the Speech, Spatial, and Qualities of Hearing scale (SSQ) Efectos de la adaptación uni o bilateral de auxiliares auditivos en las habilidades medidas la escala de cualidades auditiva, espacial y del lenguaje (SSQ)”. In: *International Journal of Audiology* 45.3, pp. 172–181.

Nuttall, A. H., Carter, G. C., and Montavon, E. M. (1974). “Estimation of the two-dimensional spectrum of the space-time noise field for a sparse line array”. In: *The Journal of the Acoustical Society of America* 55.5, pp. 1034–1041.

O’Dwyer, H., Bates, E., and Boland, F. M. (2018). “A Machine Learning Approach to Detecting Sound-Source Elevation in Adverse Environments”. In: *Audio Engineering Society Convention 144*.

O’Dwyer, H., Csadi, S., Bates, E., and Boland, F. M. (2019). “A study in machine learning applications for sound source localization with regards to distance”. In: *Audio Engineering Society Convention 146*. Audio Engineering Society.

O’Dwyer, H. and Boland, F. (2022). “HRTF Clustering for Robust Training of a DNN for Sound Source Localization”. In: *Journal of the Audio Engineering Society* 70.12, pp. 1015–1026.

Oppenheim, A. V. and Schafer, R. (2009). *Signals & systems.* 3rd Edition. New York: Pearson Educación.

Pak, J. and Shin, J. W. (2019). “Sound Localization Based on Phase Difference Enhancement Using Deep Neural Networks”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.8, pp. 1335–1345.

Paluszek, M. and Thomas, S. (2016). *MATLAB machine learning.* Apress.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). “Librispeech: An ASR corpus based on public domain audio books”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210.

Pang, C., Liu, H., and Li, X. (2019). “Multitask Learning of Time-Frequency CNN for Sound Source Localization”. In: *IEEE Access* 7, pp. 40725–40737.

Pang, C., Liu, H., Zhang, J., and Li, X. (2017). “Binaural sound localization based on reverberation weighting and generalized parametric mapping”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.8, pp. 1618–1632.

Park, H. and Yoo, C. D. (2020). “CNN-based learnable gammatone filterbank and equal-loudness normalization for environmental sound classification”. In: *IEEE Signal Processing Letters* 27, pp. 411–415.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). “Automatic differentiation in pytorch”. In.

Patterson, R. D., Robinson, K., Holdsworth, J., McKeown, D., Zhang, C., and Allerhand, M. (1992). “Complex sounds and auditory images”. In: *Auditory physiology and perception.* Elsevier, pp. 429–446.

Pauwels, J. and Picinali, L. (2023). “On the relevance of the differences between HRTF measurement setups for machine learning”. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 1–5.

Peake, W. T., Rosowski, J. J., and Lynch, T. J. (1992). “Middle-ear transmission: Acoustic versus ossicular coupling in cat and human”. In: *Hearing Research* 57.2, pp. 245–268.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.* (2011). “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Perrett, S. and Noble, W. (1997). “The effect of head rotations on vertical plane sound localization”. In: *The Journal of the Acoustical Society of America* 102.4, pp. 2325–2332.

Phokhinanan, W., Obin, N., and Argentieri, S. (2023). “Binaural Sound Localization in Noisy Environments Using Frequency-Based Audio Vision Transformer (FAViT)”. In: *INTERSPEECH.* ISCA, pp. 3704–3708.

Phokhinanan, W., Obin, N., and Argentieri, S. (2024). “Auditory Cortex-Inspired Spectral Attention Modulation for Binaural Sound Localization in HRTF Mismatch”. In: *International Conference on Acoustics, Speech, and Signal Processing.*

Pike, C. and Romanov, M. (2017). “an impulse response dataset for dynamic data-based auralization of advanced sound systems”. In: *journal of the audio engineering society* 334.

Plinge, A., Schlecht, S. J., Thiergart, O., Robotham, T., Rummukainen, O., and Habets, E. A. P. (2018). “Six-Degrees-of-Freedom Binaural Audio Reproduction of First-Order Ambisonics with Distance Information”. In: *Audio Engineering Society Conference: 2018 AES International Conference on Audio for Virtual and Augmented Reality.*

Pörschmann, C., Arend, J. M., and Gillioz, R. (2019). “How wearing headgear affects measured head-related transfer functions”. In: *EAA Spatial Audio Signal Processing Symposium.* Paris, France, pp. 49–54.

Pörschmann, C., Arend, J. M., and Neidhardt, A. (2017). “A Spherical Near-Field HRTF Set for Auralization and Psychoacoustic Research”. In: *Audio Engineering Society Convention 142.*

Pour, A. F., Asgari, M., and Hasanabadi, M. R. (2014). "Gammatonegram based speaker identification". In: *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, pp. 52–55.

Pu, H., Cai, C., Hu, M., Deng, T., Zheng, R., and Luo, J. (2021). "Towards Robust Multiple Blind Source Localization Using Source Separation and Beamforming". In: *Sensors* 21.2.

Pulkki, V. and Hirvonen, T. (2005). "Localization of virtual sources in multichannel audio reproduction". In: *IEEE Transactions on Speech and Audio Processing* 13.1, pp. 105–119.

Qian, K., Wang, J., Yang, W., and Liu, M. (2022). "Binaural sound source localization based on neural networks in mismatched hrtf condition". In: *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*, pp. 62–67.

Qiao, Y. and Choueiri, E. (2023). "Neural modeling and interpolation of binaural room impulse responses with head tracking". In: *Audio Engineering Society Convention 155*. Audio Engineering Society.

Qiao, Y., Gonzales, R. M., and Choueiri, E. (2024). "A multi-loudspeaker binaural room impulse response dataset with high-resolution translational and rotational head coordinates in a listening room". In: *Frontiers in Signal Processing* 4, p. 1380060.

Qu, T., Xiao, Z., Gong, M., Huang, Y., Li, X., and Wu, X. (2008). "Distance dependent head-related transfer function database of KEMAR". In: pp. 466–470.

Qu, T., Xiao, Z., Gong, M., Huang, Y., Li, X., and Wu, X. (2009). "Distance-Dependent Head-Related Transfer Functions Measured With High Spatial Resolution Using a Spark Gap". In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6, pp. 1124–1132.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). *The MUSDB18 corpus for music separation.*

Rascon, C. and Meza, I. (2017). "Localization of sound sources in robotics: A review". In: *Robotics and Autonomous Systems* 96, pp. 184–210.

Raspaud, M., Viste, H., and Evangelista, G. (2009). "Binaural source localization by joint estimation of ILD and ITD". In: *Ieee transactions on audio, speech, and language processing* 18.1, pp. 68–77.

Rayleigh, L. (1907). “XII. On our perception of sound direction”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 13.74, pp. 214–232.

Rayleigh, L. and Lodge, A. (1904). “Iv. on the acoustic shadow of a sphere”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 203.359-371, pp. 87–110.

Reed-Jones, J. T., Fergus, P., Ellis, D. L., and Jones, K. O. (2024a). “a study on the relative accuracy and robustness of the convolutional recurrent neural network based approach to binaural sound source localisation”. In: *Audio Engineering Society Convention 157*. 290. Audio Engineering Society.

Reed-Jones, J. T., Fergus, P., Ellis, D. L., Marsland, J., and Jones, K. O. (2024b). “Improving Full Horizontal Plane Binaural Sound Localization by use of BiLSTM”. In: *2024 International Conference on Information Technologies (InfoTech)*. IEEE, pp. 1–4.

Reed-Jones, J. T., Jones, K. O., Fergus, P., Marsland, J., and Ellis, D. L. (2023). “Comparison of Performance in Binaural Sound Source Localisation using Convolutional Neural Networks for differing Feature Representations”. In: *Audio Engineering Society Convention 154*. Audio Engineering Society.

Remaggi, L., Jackson, P. J., and Wang, W. (2019). “Modeling the comb filter effect and interaural coherence for binaural source separation”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.12, pp. 2263–2277.

Sainath, T. N., Weiss, R. J., Senior, A. W., Wilson, K. W., and Vinyals, O. (2015). “Learning the speech front-end with raw waveform CLDNNs.” In: *Interspeech*. Dresden, Germany, pp. 1–5.

Savioja, L. (1999). “Modeling techniques for virtual acoustics”. In: *Simulation* 45.10, p. 10.

Sawada, H., Ono, N., Kameoka, H., Kitamura, D., and Saruwatari, H. (2019). “A review of blind source separation methods: two converging routes to ILRMA originating from ICA and NMF”. In: *APSIPA Transactions on Signal and Information Processing* 8, e12.

Schnupp, J. and Carr, C. (2009). “On hearing with more than one ear: lessons from evolution”. In: *Nature Neuroscience* 12.6, pp. 692–697.

Serafin, S., Geronazzo, M., Erkut, C., Nilsson, N. C., and Nordahl, R. (2018). “Sonic Interactions in Virtual Reality: State of the Art, Current Challenges, and Future Directions”. In: *IEEE Computer Graphics and Applications* 38.2, pp. 31–43.

Simón, M., Hamdan, E., Menzies, D., and Fazi, F. M. (2018). “A Study of the Effect of Head Rotation on Transaural Reproduction”. In: *Audio Engineering Society Convention 145*.

Smith, J. O. (2007). *Introduction to digital filters: with audio applications.* Vol. 2. Julius Smith.

Spagnol, S., Miccini, R., Unnthórsson, R., *et al.* (2020). “The viking HRTF dataset v2”. In.

Spagnol, S., Purkhús, K. B., Unnthórsson, R., and Björnsson, S. K. (2019). “The viking HRTF dataset”. In: *16th Sound and music computing conference.* Sound and Music Computing Network, pp. 55–60.

Stephenson, U. M. (2012). “Different assumptions-different reverberation formulae”. In: *INTER-NOISE and NOISE-CON Congress and Conference Proceedings.* Vol. 2012. 4. Institute of Noise Control Engineering, pp. 7646–7657.

Sunder, K., Tan, E.-L., and Gan, W.-S. (2014). “Effect of Headphone Equalization on Auditory Distance Perception”. In: *Audio Engineering Society Convention 137*.

Tang, Z., Kanu, J. D., Hogan, K., and Manocha, D. (2019). “Regression and classification for direction-of-arrival estimation with convolutional recurrent neural networks”. In: *arXiv preprint arXiv:1904.08452*.

Thiemann, J. and Par, S. van de (2019). “A multiple model high-resolution head-related impulse response database for aided and unaided ears”. In: *EURASIP Journal on Advances in Signal Processing* 2019, pp. 1–9.

Tieleman, T. (2012). “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2, p. 26.

Tiwari, V. (2010). “MFCC and its applications in speaker recognition”. In: *International journal on emerging technologies* 1.1, pp. 19–22.

Troger, J. (1930). “Die Schallaufnahme ¡lurch das iiussere Ohr [Collection of sound by the external ear]”. In: *Phys. Z.* 31, pp. 26–47.

Varga, A. and Steeneken, H. J. (1993). "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems". In: *Speech communication* 12.3, pp. 247–251.

Varzandeh, R., Doclo, S., and Hohmann, V. (2024). "Speech-Aware Binaural DOA Estimation Utilizing Periodicity and Spatial Features in Convolutional Neural Networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32, pp. 1198–1213.

Vaswani, A. (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems.*

Vecchiotti, P., Ma, N., Squartini, S., and Brown, G. J. (2019). "End-to-end binaural sound localisation from the raw waveform". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 451–455.

Vicente, T. and Lavandier, M. (2020). "Further validation of a binaural model predicting speech intelligibility against envelope-modulated noises". In: *Hearing Research* 390, p. 107937.

Volandri, G., Di Puccio, F., Forte, P., and Carmignani, C. (2011). "Biomechanics of the tympanic membrane". In: *Journal of biomechanics* 44.7, pp. 1219–1236.

Wang, D. and Brown, G. J. (2006). "Binaural Sound Localization". In: *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, pp. 147–185.

Wang, J., Wang, J., Yan, Z., Wang, X., and Xie, X. (2019). "DNN and Clustering Based Binaural Sound Source Localization in Mismatched HRTF Condition". In: *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP).* IEEE, pp. 1–5.

Wang, J., Wang, J., Qian, K., Xie, X., and Kuang, J. (2020). "Binaural sound localization based on deep neural network and affinity propagation clustering in mismatched HRTF condition". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2020, pp. 1–16.

Wang, Q., Chai, L., Wu, H., Nian, Z., Niu, S., Zheng, S., Wang, Y., Sun, L., Fang, Y., Pan, J., *et al.* (2022). "The nerc-slip system for sound event localization and detection of dcase2022 challenge". In: *DCASE2022 Challenge, Tech. Rep.*

Warnecke, M., Clapp, S., Ben-Hur, Z., Alon, D. L., Garí, S. V. A., and Calamia, P. (2024). "Sound Sphere 2: A High-resolution HRTF Database". In: *2024 AES 5th International Conference on Audio for Virtual and Augmented Reality (AVAR)*.

Warusfel, O. (2003). *Listen HRTF Database.* `http://recherche.ircam.fr/equipes/salles/listen/index.html`. Accessed: 2024-04-29.

Watanabe, K., Iwaya, Y., Suzuki, Y., Takane, S., and Sato, S. (2014). "Dataset of head-related transfer functions measured with a circular loudspeaker array". In: *Acoustical Science and Technology* 35.3, pp. 159–165.

Wenzel, E., Arruda, M., Kistler, D., and Wightman, F. (1993). "Localization using nonindividualized head-related transfer functions". In: *The Journal of the Acoustical Society of America* 94, pp. 111–23.

Wierstorf, H., Geier, M., and Spors, S. (2011). "A free database of head related impulse response measurements in the horizontal plane with multiple distances". In: *Audio Engineering Society Convention 130*. Audio Engineering Society.

Wightman, F. L. and Kistler, D. J. (1989). "Headphone simulation of free-field listening. II: Psychophysical validation". In: *The Journal of the Acoustical Society of America* 85.2, pp. 868–878.

Wightman, F. L. and Kistler, D. J. (1992). "The dominant role of low-frequency interaural time differences in sound localization". In: *The Journal of the Acoustical Society of America* 91.3, pp. 1648–1661.

Woodworth, R. S. and Schlosberg, H. (1938). *Experimental psychology.* Holt.

Wu, X., Talagala, D. S., Zhang, W., and Abhayapala, T. D. (2016). "Spatial feature learning for robust binaural sound source localization using a composite feature vector". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 6320–6324.

Xu, Y., Afshar, S., Singh, R. K., Wang, R., Schaik, A. van, and Hamilton, T. J. (2019). "A Binaural Sound Localization System using Deep Convolutional Neural Networks". In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5.

Yang, B., Li, X., and Liu, H. (2021a). "Supervised Direct-Path Relative Transfer Function Learning for Binaural Sound Source Localization". In: *ICASSP 2021 -*

2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 825–829.

Yang, B., Liu, H., and Li, X. (2021b). "Learning Deep Direct-Path Relative Transfer Function for Binaural Sound Source Localization". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29, pp. 3491–3503.

Yang, Q. and Zheng, Y. (2024). "DeepEar: Sound Localization With Binaural Microphones". In: *IEEE Transactions on Mobile Computing* 23.1, pp. 359–375.

Yiwere, M. and Rhee, E. J. (2017). "Distance estimation and localization of sound sources in reverberant conditions using deep neural networks". In: *Int. J. Appl. Eng. Res* 12.22, pp. 12384–12389.

Yu, G., Wu, R., Liu, Y., and Xie, B. (2018). "Near-field head-related transfer-function measurement and database of human subjects". In: *The Journal of the Acoustical Society of America* 143.3, EL194–EL198.

Zhang, J. and Liu, H. (2015). "Robust acoustic localization via time-delay compensation and interaural matching filter". In: *IEEE Transactions on Signal Processing* 63.18, pp. 4771–4783.

Zhao, F., Li, R., and Pan, D. (2021). "Deep learning for binaural sound source localization with low signal-to-noise ratio". In: *Journal of Physics: Conference Series*. Vol. 1828. 1. IOP Publishing, p. 012017.

Zhou, L., Ma, K., Wang, L., Chen, Y., and Tang, Y. (2019). "Binaural Sound Source Localization Based on Convolutional Neural Network." In: *Computers, Materials & Continua* 60.2.

Ziegelwanger, H., Kreuzer, W., and Majdak, P. (2015). "Mesh2hrtf: Open-source software package for the numerical calculation of head-related transfer functions". In: *22nd international congress on sound and vibration*.