Università degli studi di RomaTre

Emilio Caschera 510911

February 19, 2025

Source Distance Estimation

Evaluating Encoder Architectures: Performance Variations in Source Distance Estimation Models

Abstract

This study aims to implement and compare the performance of various encoders for input preprocessing in SeldNet, a neural network developed to address the problem of audio distance estimation. Four types of encoders based on filterbanks, implemented using the Asteroid Audio library, were evaluated: an unconstrained encoder, an unconstrained encoder with analyticity constraints, a parametric encoder, and an STFT-based encoder. Additionally, a custom encoder was developed using PyTorch, consisting of four convolutional blocks.

Experimental results demonstrated that the STFT-based encoder significantly outperformed all other approaches, showing superior accuracy in distance estimation. The unconstrained encoders, particularly the one without constraints, approached the STFT's results in certain scenarios but did not match its overall performance.

These findings highlight the validity of the STFT approach as an optimal solution for audio preprocessing in distance estimation applications, while also underscoring the limitations of more advanced, customized architectures.

Contents

Chapter 1

INTRODUCTION

1.1 Source Distance Estimation

Source Distance Estimation (SDE) is a subfield of auditory scene analysis that focuses on measuring the distance between an audio source and a receiver, typically a microphone or an array of microphones. This task is complicated by the fact that audio signals are subject to attenuation, reflection, and diffraction, which depend on environmental characteristics such as reverberation and background noise. Unlike Direction of Arrival (DoA) estimation, which concentrates on the angles of sound wave incidence, SDE aims to provide a quantitative measure of distance, a crucial element for applications in surveillance, human-computer interaction, and augmented reality. Both tasks are valuable in many practical applications, including enhancing the robustness of automatic speech recognition systems by improving the performance of acoustic echo cancellers and in autonomous robotics. Despite the fact that DoA and source distance estimation are both conducted using multichannel audio in most practical scenarios, the latter has been largely overlooked [1].

First, source distance estimation is generally considered a more challenging task due to the fact that distance cues weaken as the distance between the audio source and the receiver increases. Second, DoA provides sufficient information for many downstream tasks in spatial filtering. However, many applications, such as source separation, acoustic monitoring, and context-aware devices, would still benefit from complete information on the sound source's location, highlighting the need for further research into source distance estimation (SDE).

Most methods for DoA and distance estimation rely on arrays with more than two microphones [2]. Multichannel data enables the exploitation of spatial cues such as interchannel time differences (ITD) and interchannel level differences (ILD) to provide useful information for efficient DoA estimation, positively impacting distance estimation as well [3]. However, the use of multiple microphones comes with limitations related to budget and physical portability. To address this issue, some studies have explored the use of binaural recordings, reducing the number of channels to two and leveraging human auditory signals . Our auditory system identifies the Direction of Arrival (DoA) of a sound source—estimating the angle from which the sound originates—through the head and ear diffraction effects, which act as directional detectors [4]. More specifically, humans use binaural cues known as interaural time differences (ITD) and interaural intensity differences (IID) between the sound waves reaching the eardrums. However, binaural signals do not contain source distance information except at close range, so distance perception must rely on a combination of higher-level or multimodal cues.

The simplest of these is sound level attenuation according to the inverse distance law due to spherical wave propagation from the source, which is nonetheless a relative cue and requires familiarity with a certain source signal and environment. Additionally, auditory distance localization mainly relies on echo and reverberation cues [5] as well as audiovisual information [6]. If some or all of these cues are removed, as in monophonic headphone listening tests described in [7], human distance perception performance can be quite poor [8]. Nevertheless, the simpler scenario of single-microphone distance estimation has been largely underestimated.

Two main approaches have been adopted to address the SDE problem: regression and classification. The regression approach treats distance estimation as a continuous variable, allowing the prediction of exact distances (e.g., 1.75 meters). This method offers greater flexibility and precision but can be more complex to manage, particularly regarding generalization to new scenarios and effective model training. In contrast, the classification approach simplifies the problem by dividing distances into discrete classes, such as ”near,” ”medium,” and ”far,” or defining precise distance intervals like [0-1m], [1-2m], [2-3m]. While classification facilitates model training and often enhances robustness, it sacrifices estimation granularity, limiting the method’s applicability in contexts where high precision is required.

1.2 Related Works

In the field of audio signal processing, Sound Source Distance Estimation (SDE) and Direction of Arrival (DOA) determination represent fundamental challenges with significant practical applications. In recent years, various methodologies have been proposed to address these challenges, ranging from classical machine learning approaches to more recent deep learning architectures.

Traditional machine learning techniques have demonstrated remarkable effectiveness in addressing these tasks. Specifically, Gaussian Mixture Models (GMM) have been extensively utilized in various configurations: Brendel et al. employed GMMs [9] to estimate the coherent-to-diffuse power ratio for determining source-microphone distance, while Vesa trained these models using Magnitude Squared Coherence (MSC)-based features to incorporate channel correlation information [10] [11].

Other classical approaches have included the use of classifiers such as K-Nearest Neighbors (KNN)[12] and Linear Discriminative Analysis (LDA), often in conjunction with MSC features. A significant contribution was made by Georganti et al., who introduced the Binaural Signal Magnitude Difference Standard Deviation (BSMD-STD) as an innovative feature for training GMMs and Support Vector Machines (SVMs) [13].

Despite these successes, most of these methods rely on complex algorithms requiring precise calibration to adapt to various acoustic conditions. The exploration of Deep Neural Network (DNN)-based approaches for source distance estimation remained relatively limited until recently. Yiwere et al. proposed an approach inspired by image classification, utilizing CRNNs trained on log-mel spectrograms to classify three different distances in three distinct environments. However, while these models showed promising results within the same environment, their performance degraded significantly when applied to recordings from different environments [14].

A significant advancement was achieved by Sobhdel et al., who introduced relational networks to address this challenge through few-shot learning, demonstrating improvements over traditional Convolutional Neural Networks (CNNs) [15]. It is noteworthy that both studies conducted tests within a limited range of specific distances, generally not exceeding 3-4 meters. In the context of monaural audio, research on speaker distance estimation has been even more limited. Among the early works, some utilized low-level features such as Linear Predictive Coding (LPC), spectral skewness, and kurtosis to classify speaker distance. Venkatesan et al. proposed both monaural and binaural features to train GMMs and SVMs [16]. Regarding DNN approaches, Patterson et al. classified ”far” and ”near” speech to perform sound source separation from monaural audio.

1.3 Proposed Method

In this work, we propose a Convolutional Recurrent Neural Network (CRNN) model. This architectural typology was selected as it has demonstrated favorable results in numerous studies for Sound Event Localization and Detection (SELD) tasks.

The proposed model leverages acoustic feature extraction from monaural audio signals to capture temporal, spatial, and spectral characteristics relevant to distance estimation. Specifically, the magnitude and phase maps of the Short-Time Fourier Transform (STFT) are extracted from the audio signal. These representations contain information about the acoustic properties of the environment in which the sound propagates, proving valuable for the SDE task. To emphasize the audio signal regions most significant for distance estimation, an attention module is introduced. This module learns an attention map that is subsequently applied to the input acoustic features, enabling the model to focus on the most informative parts of the signal.

The CRNN model structure incorporates convolutional layers for feature extraction and bidirectional recurrent layers (Gated Recurrent Unit, GRU) for temporal information integration. Finally, fully connected layers are employed to perform regression and obtain the final sound source distance estimation. The model training is accomplished by minimizing a loss function that accounts for both the distance estimation for each temporal frame and the final distance estimation for the entire signal. This configuration acts as a regularization term, promoting coherent estimates at both frame and overall signal levels. The model input has been designed as a three-dimensional tensor that includes magnitude and phase information of the audio signal, represented through transforms such as the Short-Time Fourier Transform (STFT). The phase, often overlooked, has been included as it provides additional distance cues due to phase variations caused by the signal path.

1.4 Comparative Analysis of Encoder Architecture

The selection of appropriate pre-processing filters represents a crucial aspect in Sound Source Distance Estimation (SDE) systems, as it directly influences the quality and reliability of the extracted acoustic features. The choice of filtering methodology can significantly impact the model’s ability to capture relevant distance cues from the audio signal, particularly in challenging acoustic environments. This work aims to conduct a comprehensive analysis of different types of filters that can be employed as pre-processing for SDE models. Four main filter categories are compared:

"Free" filters: where the weights of analysis and synthesis filters are learned completely freely by the model, offering maximum flexibility. "Analytic free" filters: which incorporate the Hilbert transform to ensure temporal shift invariance, as a variant of both learnable filter categories. "Parametric" filters: belonging to predefined filter families, whose parameters are optimized by the model, representing a compromise between flexibility and constraints. "Fixed" filters: consisting of predetermined transformations such as STFT, gamma-tone filters, or Mel filters, which leverage desirable theoretical properties but cannot be adapted to the specific task. Additionally, the implementation of a filter created from scratch and composed of 4 convolutional layers is explored.

This evaluation enables us to assess not only the absolute performance of each filter type but also their robustness across different acoustic conditions and their computational efficiency in real-world applications.

Chapter 2

PROPOSED METHOD

In this section, we present a detailed description of our proposed architecture for Sound Source Distance Estimation (SDE), which integrates various filtering approaches with a deep learning framework. Our methodology encompasses both the fundamental model structure and a systematic exploration of different filtering strategies.

The architecture is designed to effectively process monaural audio signals while maintaining computational efficiency and robustness across diverse acoustic environments. We first describe the core neural network architecture, followed by a comprehensive analysis of the implemented filtering mechanisms and their distinctive characteristics. Particular attention is given to the design choices that enable effective feature extraction and the integration of different filtering approaches within the learning framework.

2.1 Model Architecture

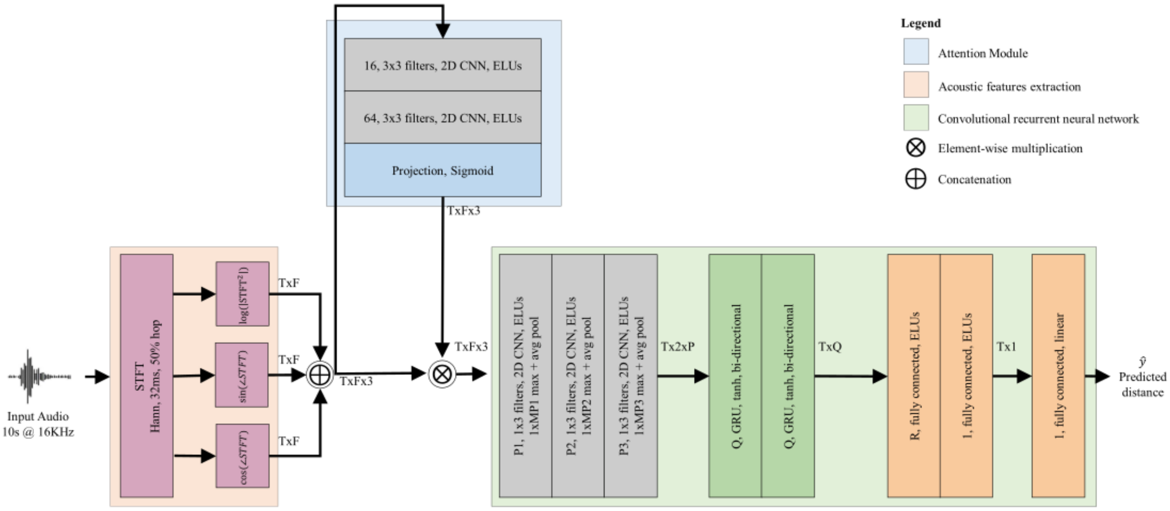

Legend
Attention Module
Acoustic features extraction
Convolutional recurrent neural network
Element-wise multiplication
Concatenation
16, 3x3 filters, 2D CNN, ELUs
64, 3x3 filters, 2D CNN, ELUs
Projection, Sigmoid
TxFx3
Input Audio
10s @ 16KHz
STFT
Hann, 32ms, 50% hop
log(|STFT^2|)
sin(∠STFT)
cos(∠STFT)
TxF
TxF
TxF
TxFx3
TxFx3
P1, 1x3 filters, 2D CNN, ELUs 1xMP1 max + avg pool
P2, 1x3 filters, 2D CNN, ELUs 1xMP2 max + avg pool
P3, 1x3 filters, 2D CNN, ELUs 1xMP3 max + avg pool
Tx2xP
Q, GRU, tanh, bi-directional
Q, GRU, tanh, bi-directional
TxQ
R, fully connected, ELUs
1, fully connected, ELUs
Tx1
1, fully connected, linear
$\hat{y}$
Predicted distance

Figure 2.1: Proposed architecture for speaker distance estimation. First, acoustic features are extracted from the single-channel audio. In more detail, 3 maps (magnitude of the short-time Fourier transform (STFT), sinus, and cosinus of the STFT phase) are obtained with shape T × F , where T and F are the time and frequency bins, respectively. Then, the maps are stacked along the channel dimension resulting in a feature tensor of size T × F × 3. To highlight the feature regions that are most informative for distance estimation, an attention map is learned from the three-channel tensor, which is then element-wise multiplied with the input feature tensor. The output is further processed by the convolutional layers with Pi 1 × 3 kernels, also denoted as frequency kernels, yielding a T × 2 × P tensor that is arranged in a T × Q matrix, where Q = 2P . Subsequently, the resulting matrix is analyzed by two gated recurrent unit (GRU) layers with Q neurons to model temporal patterns. Finally, the output from recurrent layers T × Q is fed to three fully connected layers with R, 1, and 1 neurons respectively to map the features to the predicted distance ŷ.

To process temporal, spatial, and spectral characteristics of these features, a CRNN has been employed for the experiments. This type of model has shown good results in many studies for sound event local- ization and detection (SELD) tasks [17], [18]. While the core neural architecture employed in our experiments follows the model proposed by Michael Neri et al., this study specifically focuses on modifications to the input preprocessing stage. Our investigation centers on varying the feature extraction component upstream of the network, while maintaining the subsequent architecture unchanged. This approach allows for a direct comparison of different filtering strategies while controlling for other architectural variables.

2.1.1 Acoustic Features Extraction

All the operations on the audio files are performed at 16 kHz. The selection of this sampling frequency is because the speech spectrum is mostly contained in the range 0-8 kHz [19]. In addition, a lower frequency yields a lower number of samples, reducing the computational complexity of feature extraction and distance estimation. Initially, a pre-processing stage is employed to extract the complex $\mathrm{STFT}\{x\} \in \mathbb{C}^{T\times F}$from the single-channel audio signal $\mathbf{x} \in \mathbb{R}^{1\times L}$, where T is the number of time frames, F the number of frequency bins, and L the number of samples. This transformation is computed using a Hann window of length 32 ms with 50% overlap. Subsequently, the magnitude ($|\mathrm{STFT}\{\mathbf{x}\}| \in \mathbb{R}^{T\times F}$) and phase ($\angle\mathrm{STFT}\{\mathbf{x}\} \in \mathbb{R}^{T\times F}$) components of the STFT are computed from the complex matrix. Sinus and cosinus maps of the phase spectrogram are computed by applying $\sin(\cdot)$ and $\cos(\cdot)$ functions element-wise, since the features provide a smoother continuous representation of the raw phase information. The concept of utilizing the phase spectrogram has been adopted from contemporary research on multichannel source separation [20], learning-based localization [21], and speech enhancement [22] as phase information contains cues regarding the acoustic properties of the environment in which the sound propagates [23]. Finally, the magnitude of the STFT and the sinus and cosinus maps are stacked into a $T \times F \times 3$ tensor. This representation is then fed into the attention module and the convolutional layers for further processing and analysis.

2.1.2 Attention Module

computes an attention map $H \in \mathbb{R}^{+T\times F\times 3}$ from the audio features. The objective of this learned matrix is to emphasize the regions of the features that are most informative for the estimation of the distance. Specifically, this module is the function $f_{ATT} : \mathbb{R}^{T\times F\times 3} \to \mathbb{R}^{+T\times F\times 3}$. Its structure is composed of 2 convolutional blocks, having 16 and 64 3×3 filters, respectively. Then, a 1×1 convolutional layer with three filters, followed by a sigmoid activation, is used to map the features to yield the $T \times F \times 3$ attention map. Finally, the output acoustic features $\tilde{X} \in \mathbb{R}^{T\times F\times 3}$ are obtained by element-wise multiplication ($\otimes$) between the input acoustic features and the attention map as

$$\tilde{X} = f_{ATT}(X) \otimes X \tag{2.1}$$

2.1.3 Convolutional Layers

The architecture employs three convolutional blocks for feature extraction. In more detail, the structure of each block involves a 2D convolutional layer comprising P_i 1 × 3 filters, i.e., along the frequency axis with values of 8, 32, and 128 assigned to the respective layers. We denote these filters as $frequency\ kernels$. In fact, rectangular filters can be more parameter-efficient compared to square kernels. Since the former has fewer parameters than square kernels of the same receptive field size, they can lead to a more compact model, making training and inference more computationally efficient and potentially reducing the risk of overfitting. Following this layer, a batch normalization [24] step is applied, along with max and average pooling operations along the frequency dimension. Then, the results of which are summed. The activation function utilized after each convolutional layer is the exponential linear unit (ELU) [25], which is denoted as

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{per } x \geq 0 \\ \alpha(e^x - 1), & \text{per } x < 0 \end{cases} \tag{2.2}$$

where α is a coefficient that regularizes the saturation of negative values. Notably, each layer employs a specific pooling rate denoted by $M\ P_i$, with values of 8, 8, and 2 assigned to the respective layers.

2.1.4 Recurrent Layers

To process the feature maps from the convolutional layers, two bi-directional GRU layers are utilized with $\tanh(\cdot)$ as the activation function. These layers have exhibited promising results in audio and speech processing tasks, demonstrating parameter efficiency compared to long short-term memory (LSTM) networks [26]. The output of the CNN with shape T × 2 × P is stacked along the channel dimension to produce a T × Q matrix to be fed to the recurrent layers. Then, in the proposed configuration, the extraction of reverberation-related information primarily relies on integrating information over time with the recurrent layers. Within this implementation, two bi-directional GRUs with Q = 2P = 128 neurons each for every time frame are employed. Then, to predict the distance, three fully connected layers are employed, where an independent mapping between each time frame is performed in each layer. Firstly, the initial linear layer projects time-wise features from the last GRU onto a matrix of dimensions T × R, where R = 128. Subsequently, the second linear layer independently maps each time frame of the T × R matrix onto a vector of size T × 1, denoted as the time-wise distance estimation $\hat{\mathbf{y}}$. Specifically, this vector represents the distance estimation for each time frame. Finally, the last fully connected layer is employed to perform regression and thus estimate the predicted distance, denoted as $\hat{y} \in \mathbb{R}$.

2.2 Proposed Filters

Most state-of-the-art speech separation methods can be described using an encoding-masking-decoding framework. An encoder transforms the time-domain signal by convolving every signal frame indexed by k $\in \{0, ..., K-1\}$ with a bank of N *analysis* filters $\{u_n(t)\}_{n=0...N-1}$ of length L:

$$\mathbf{X}(k,n) = \sum_{t=0}^{L-1} x(t+kH)u_n(t), \quad n \in \{0,\ldots,N-1\} \tag{2.3}$$

where H is the hop size. After an optional non-linearity $\mathcal{G}$, $\mathbf{X}$ is then fed to the masking network $\mathcal{MN}$:

$$\mathcal{MN}(\mathcal{G}(\mathbf{X})) = [\mathbf{M_1},\ldots,\mathbf{M_C}] \tag{2.4}$$

Each estimated mask $\mathbf{M}_i$ is multiplied with the input to obtain the estimated representation of source i:

$$\mathbf{Y}_i = \mathcal{G}(\mathbf{X}) \odot \mathbf{M}_i, \quad i \in \{1,\ldots,C\} \tag{2.5}$$

with $\odot$ denoting point-wise multiplication. The decoder maps each $\mathbf{Y}_i$ to the time domain by transposed convolution with a bank of N *synthesis* filters $\{v_n(t)\}_{n=0...N-1}$ of length L:

$$\hat{s}_i(t) = \sum_{k=0}^{K-1}\sum_{n=0}^{N-1} \mathbf{Y}_i(k,n)v_n(t-kH) \tag{2.6}$$

2.2.1 Free Filterbank

In the standard free filter approach, all filter weights $\{u_n(t)\}$ and $\{v_n(t)\}$ are jointly learned within the masking network. This provides maximum flexibility in filter design, allowing the network to discover optimal filter configurations through the training process. Free filterbank without any constraints is equivalent to a one dimension convolution.

2.2.2 Parametrized Filterbank

Parameterized filters belong to a predefined family of filters, with parameters learned concurrently with the network. We defined parameterized analytic analysis filters as:

$$u_n(t;\theta) = 2f_2 \operatorname{sinc}(2\pi f_2 n) - 2f_1 \operatorname{sinc}(2\pi f_1 n) = 2f_w \operatorname{sinc}(2\pi f_w n)\cos(2\pi f_c n), \tag{1}$$

where $\theta = \{f_1, f_2\}$, $f_w = f_2 - f_1$, and $f_c = \frac{f_1+f_2}{2}$.All filters drawn from this family are even functions, thus making it unsuitable for resynthesis.

2.2.3 Fixed Filterbank

Fixed filters represent handcrafted transforms such as the STFT, gammatone or Mel filters. In the case of the STFT:

$$u_n(t) = h_a(t)e^{-2j\pi\frac{n}{N}} \quad \text{and} \quad v_n(t) = h_s(t)e^{2j\pi\frac{n}{N}} \tag{2}$$

with h_a and h_s the analysis and synthesis windows.

A desirable property of time-frequency representations is shift invariance, i.e., invariance to small delays in the time domain. Analytic filters [17] have this property. Namely, the modulus of the convolution between a real-valued signal and an analytic filter is the envelope of that signal in the frequency band defined by the filter. The STFT filters (2) are examples of such analytic filters, and the magnitude of the STFT is the corresponding shift-invariant representation. Given any real-valued filter $u(t) \in \mathbb{R}^{1\times L}$, a corresponding analytic filter $u_{analytic}(t)$ can be obtained as

$$u_{analytic}(t) = u(t) + j\mathcal{H}[u(t)] \tag{3}$$

where $\mathcal{H}$ denotes the Hilbert transform which imparts a $-\pi/2$ phase shift to each positive frequency component. In the following, we detail the proposed analytic expansion of both parameterized and free filters

2.2.4 Analytic Filterbank

We define parameterized analytic analysis filters u_n as

$$u_n(t;\theta) = 2f_w \operatorname{sinc}(2\pi f_w t)\left(\cos(2\pi f_c t) - j\sin(2\pi f_c t)\right) = 2f_w \operatorname{sinc}(2\pi f_w t)e^{-2j\pi f_c t} \tag{4}$$

This complements the original family of even filters (1) with odd ones. The new family $\{u_n\}_{n=1..N}$ can form a complete basis of the signal space, and each filter is analytic so that

$$\Im(u_n(t;\theta)) = \mathcal{H}[\Re(u_n(t,\theta))] \tag{5}$$

The corresponding family of synthesis filters is defined as:

$$v_n(t;\phi) = 2g_n f_w \operatorname{sinc}(2\pi f_w t) e^{2j\pi f_c t} \tag{6}$$

where $\phi = \{f_1, f_2, g\}$, and g_n is a gain parameter learned to improve resynthesis
Similarly, in the case of free filters, we propose to ensure that the learned filters are analytic by parameterizing them by their real part and computing the corresponding analytic filter via (8) during the forward pass of the network. This is applied to both analysis and synthesis filters.

It is worth noting that the original filters were specifically designed for speech separation tasks, implementing an encoder-decoder architecture. This architecture leverages analysis filters for signal decomposition and synthesis filters for signal reconstruction. However, for the purposes of our investigation, we focus exclusively on the encoder component, specifically the analysis filters. These filters are instrumental in analyzing the input signal and generating an abstract geometric representation that the network can exploit to learn salient signal features. The rationale behind this approach lies in the encoder's capability to transform the raw signal into a high-dimensional feature space where relevant patterns become more distinguishable. This transformation facilitates the network's ability to identify and extract crucial signal characteristics, which is fundamental for subsequent processing stages. The abstract representation generated by the analysis filters serves as a rich feature space that captures both local and global signal properties, enabling more effective pattern recognition and feature extraction.

2.2.5 Custom Convolutional Encoder

The design of the custom encoder was driven by empirical evidence that emerged during the training phase. Performance analysis revealed that "free" filters consistently produced superior results, although this trend was not uniform across all folds. A significant observation emerged from analyzing the correlation between the evaluation metric and loss function: at the conclusion of the predetermined training period, the model exhibited clear signs of active learning, suggesting unexplored potential in feature extraction capabilities.

Technical documentation indicates that a "free" filterbank encoder is fundamentally equivalent to a one-dimensional convolution operation (Conv1D). This understanding led to the development of a novel architectural approach: an encoder based on Conv1D operations with enhanced structural depth. This design choice was motivated by the necessity to enable the model to learn representations at a higher abstraction level, thus facilitating the capture of more complex and significant patterns in the input data.

The proposed encoder implements a sequential structure of operations for each filtering stage:

1. One-dimensional convolution (`Conv1D`) for feature extraction
2. Batch normalization (`BatchNorm1D`) to stabilize the learning process
3. Exponential Linear Unit (`ELU`) activation function to introduce non-linearity
4. MaxPooling (`MaxPool1d`) operation for dimensional reduction and feature selection

This architectural configuration was designed to optimize both feature extraction capabilities and training stability, while maintaining computational efficiency.

First Layer
`Conv1d()`, kernel_size = 5, stride = 1, padding = 2
`BatchNorm1d()`
`ELU()`
`MaxPooling()`, kernel_size = 5, stride = 1, padding = 2
Second Layer
`Conv1d()`, kernel_size = 5, stride = 1, padding = 2
`BatchNorm1d()`
`ELU()`
`MaxPooling()`, kernel_size = 5, stride = 1, padding = 2
Third Layer
`Conv1d()`, kernel_size = 5, stride = 1, padding = 2
`BatchNorm1d()`
`ELU()`
`MaxPooling()`, kernel_size = 5, stride = 1, padding = 2
Fourth Layer
`Conv1d()`, kernel_size = 3, stride = 1, padding = 1
`BatchNorm1d()`
`ELU()`
`MaxPooling()`, kernel_size = 3, stride = 1, padding = 2

Table 2.1: Structure of the custom convolutional encoder

Chapter 3

TRAINING SETUP

This section delineates the training methodology employed for our sound distance estimation model.

3.1 Dataset

The dataset used for experiments follows the same setup as in [27]. Briefly, anechoic speech recordings obtained from the TIMIT dataset [28] are convolved with the simulated omnidirectional RIRs from an image-source room simulator for shoebox geometries [29]. This simulator allows for frequency-dependent wall absorption and directional encoding of image sources in 5th order Ambisonics format. The elevation range between the source and the receiver spanned from −35° to 35°. To compile a list of materials and their respective absorption coefficients for each surface type (ceiling, floor, and wall), we refer to widely used acoustical engineering tables [30]. For each unique simulated room with its room-source-distance configuration, a random material is assigned to each surface, resulting in 2912 possible material combinations. Compared to randomizing directly the target RT60 for each simulated room, this randomization approach allows us to avoid matching unnatural reverberation times to specific room volumes (e.g., a very long RT60 for a small room) and ensure a more natural distribution of reverberation times. The final distribution of reverberation times exhibits a median, 10th percentile, and 90th percentile of 0.83 s, 0.42 s, and 2.38 s, respectively. Furthermore, the positions of the sound sources are uniformly distributed in terms of the azimuth angle relative to the receiver. The experiments include 2500 audio files of 10 s

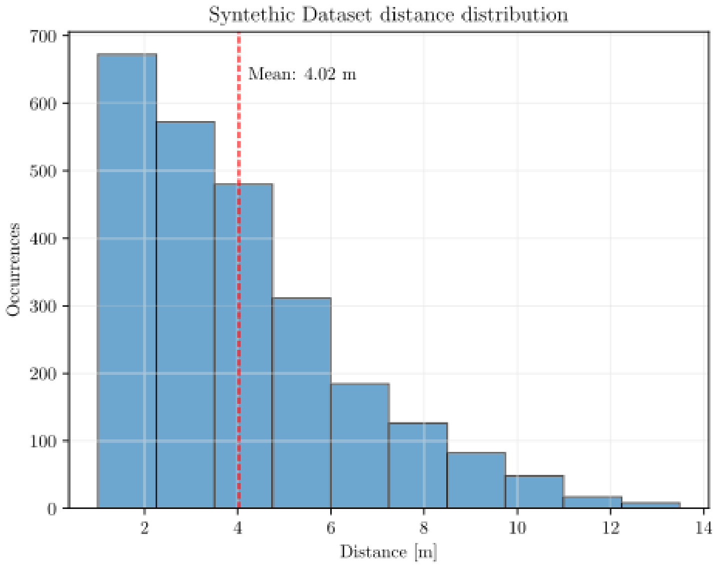


Figure 3.1: Distribution of distances in synthetic dataset

duration at 16 kHz in compliance with the speech dataset. In the evaluation, 5-fold cross validation is used where 1500, 500, and 500 files are assigned to training, validation, and testing in each fold.

3.2 Loss Function

The mean squared error (MSE) loss is used to train the DNN system. Let $y \in \mathbb{R}$ be the true distance of a static sound source. In addition, let $\mathbf{y} \in \mathbb{R}^{T \times 1}$ be the vector consisting of frame-wise ground truth distances. Then, the loss used in the training phase for a single sample is

$$\mathcal{L}(y, \hat{y}, \mathbf{y_t}, \hat{\mathbf{y}}_\mathbf{t}) = (y - \hat{y})^2 + \|\mathbf{y_t} - \hat{\mathbf{y}}_\mathbf{t}\|^2 \tag{3.1}$$

where the loss is averaged across the batch dimension to be exploited by the backpropagation algorithm. Thanks to the imposition of the loss, the model predicts a distance for each time bin and, from this information, a single-valued distance

3.3 Metrics

The performance evaluation of our approach utilizes the mean absolute error (**MAE**) ($\mathcal{L}_1$) as the performance measure for the entire test dataset

$$\mathcal{L}_1(y, \hat{y}) = |y - \hat{y}| \tag{3.2}$$

where the ground truth $y \in \mathbb{R}$ and the prediction $\hat{y} \in \mathbb{R}$ are considered. Additionally, the performance is assessed by calculating the **MAE** within different distance ranges. This analysis allows us to quantify the relative error of our model concerning source distance. We define the relative **MAE** ($r\mathcal{L}_1$), which includes the real speaker distance in the evaluation, as follows:

$$r\mathcal{L}_1(y, \hat{y}) = \frac{\mathcal{L}_1(y, \hat{y})}{y} = \frac{|y - \hat{y}|}{y} \tag{3.3}$$

MSE has not been considered in the performance evaluation.

3.4 Training Procedure and Experimental Setup

The model was trained using a comprehensive cross-validation approach to ensure robust performance and generalizability. The training methodology employed the Adam optimizer, a popular adaptive learning rate optimization algorithm known for its effectiveness in deep learning tasks.

Configuration	**Value**
Optimizer	Adam
Initial Learning Rate (λ)	0.0001
Batch Size	16 samples per iteration
Total Training Epochs	50

Table 3.1: Optimization Configuration

To mitigate overfitting and improve model generalization, we implemented a learning rate scheduling strategy. Specifically, the learning rate was reduced by 80% every 5 epochs when no improvement was observed in the Mean Squared Error (MSE) of the validation set. This adaptive learning rate approach helps the model converge more effectively by dynamically adjusting the step size during training. Cross - Validation Methodology: The training process was conducted using 5-fold cross-validation. This approach involves randomly partitioning the dataset into 5 equally sized subsets or 'folds'. In each iteration, 3 folds were used for training, 1 fold for validation and 1 fold for test, with this process repeated such that each fold serves as the validation and test set exactly once. This methodology ensures a comprehensive evaluation of the model's performance across different data subsets, providing a more robust estimate of the model's generalization capability.

Chapter 4

RESULTS

In this study, we evaluated the performance of various encoders for manipulating audio inputs in a source distance estimation task using a Convolutional Recurrent Neural Network (CRNN). The encoders tested were the Short-Time Fourier Transform (STFT), Analytic Free, Free, Parametric Sinc (Param Sinc), and a custom encoder composed of four convolutional blocks described in table 2.1. All encoders were sourced from the Asteroid library, which provides implementations of advanced audio processing techniques.

The dataset comprised 2,500 audio samples at a sampling frequency of 16 kHz, chosen because the speech spectrum is predominantly contained within the 0–8 kHz range. The average source distance in the dataset was approximately 4 meters, as seen in figure 3.1. Each of the five folds used three folds for training, one for validation, and one for testing. The Mean Absolute Error (MAE) was calculated for each fold to assess performance. The results are summarized in Table 4.1.

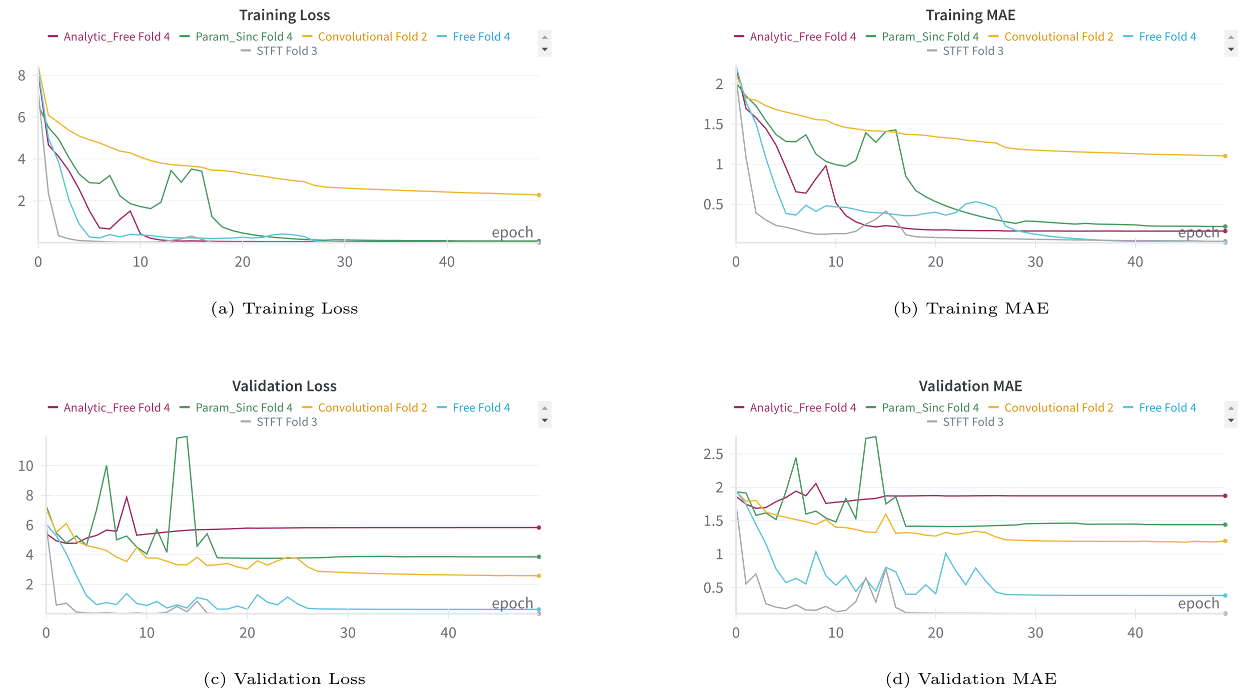
Encoder Type	**Fold1 Test**	**Fold2 Test**	**Fold3 Test**	**Fold4 Test**	**Fold5 Test**	**CV Mean**
STFT	0.136	0.197	0.113	0.127	0.117	0.138
Free	0.435	1.826	1.857	0.328	1.558	1.200
Analytic Free	1.855	1.901	1.845	1.810	1.892	1.861
Param Sinc	1.772	1.762	1.959	1.846	1.741	1.816
Custom Encoder	1.178	1.139	1.161	1.157	1.215	1.170

Table 4.1: MAE performance comparison of different encoder types across folds and mean value across folds.

4.1 Analysis of Results

The STFT encoder achieved the lowest average MAE of 0.138, significantly outperforming the other encoders. The custom encoder yielded an average MAE of 1.170, which is better than the analytic free, free, and parameterized sinc filterbanks but still notably higher than the STFT. The other filterbanks exhibited average MAEs ranging from 1.273 to 1.861.

Given that the average source distance in the dataset is approximately 4 meters, an MAE in the range of 1 to 1.5 meters constitutes a substantial error, rendering the performance of the analytic free, free, and parameterized sinc filterbanks suboptimal for precise distance estimation.



(a) Training Loss

(b) Training MAE

(c) Validation Loss

(d) Validation MAE

Figure 4.1: Training and validation loss trend for the test fold with the best MAE value.

Encoder Type	Analytic Free	Param Sinc	Convolutional	Free	STFT
MAE	1.810	1.392	1.139	0.328	0.113

Table 4.2: MAE value of the best-performing test fold.

4.2 Correlation Between Filter Properties and Performance

The superior performance of the STFT encoder can be attributed to its inherent properties. The STFT utilizes fixed, handcrafted filters defined as:

$$u_n(t) = h_a(t)e^{-2j\pi\frac{n}{N}} \quad \text{and} \quad v_n(t) = h_s(t)e^{2j\pi\frac{n}{N}} \tag{4.1}$$

where $h_a(t)$ and $h_s(t)$ are the analysis and synthesis windows, respectively. The STFT filters are analytic and provide a shift-invariant time-frequency representation, capturing both amplitude and phase information effectively. This allows the subsequent CRNN to learn more discriminative features pertinent to distance estimation.

In contrast, the analytic free and free filterbanks involve filters whose weights are entirely learned during training:

$$u_n(t) = \text{learned weights} \tag{4.2}$$

without any initial structure or prior knowledge. While this offers flexibility, it also poses challenges in learning meaningful representations from limited data, potentially leading to overfitting or suboptimal feature extraction. The high MAE values for these encoders suggest that the learned filters failed to capture the essential characteristics of the audio signals related to source distance.

The parameterized sinc filters, defined as:

$$u_n(t;\theta) = 2f_w \operatorname{sinc}(2\pi f_w t)e^{-2j\pi f_c t} \tag{4.3}$$

introduce a parameterization that restricts the filters to a specific family governed by cutoff frequencies f_1 e f_2. Although this adds some structure compared to free filters, the parameterization may still be insufficient to capture the complex features necessary for accurate distance estimation, as evidenced by the relatively high MAE.

The custom encoder, comprising four convolutional blocks described in table 2.1, performed better than the parameterized and free filterbanks but not as well as the STFT. The convolutional layers can learn hierarchical feature representations, which may be more effective than the learned filterbanks in capturing spatial cues from the audio signals. However, without the explicit time-frequency representation provided by the STFT, the custom encoder may lack the resolution needed for precise distance estimation.

Chapter 5

CONCLUSION

In signal processing and machine learning, the encoder represents a critical preprocessing stage that transforms raw input data into an abstract representation, enabling subsequent neural network architectures to effectively analyze and interpret complex acoustic signals. The encoder's role extends beyond mere signal transformation; it serves as a fundamental mechanism for feature extraction, directly influencing the model's ability to capture and represent intrinsic signal characteristics. In the domain of Source Distance Estimation (SDE), the encoder's design becomes particularly pivotal. By creating an abstracted representation of the acoustic signal, the encoder fundamentally mediates the network's capacity to extract and interpret distance-related information. Subtle modifications to the encoder's architecture can dramatically alter the model's performance, underscoring the encoder's significance as a critical determinant of overall system effectiveness.

The empirical results reveal a marked performance disparity among the evaluated encoder architectures. The Short-Time Fourier Transform (STFT) consistently demonstrated superior performance across all experimental configurations, achieving substantially lower Mean Absolute Error (MAE) values ranging from 0.113 to 0.197. In stark contrast, alternative encoder approaches exhibited significantly higher error rates. Specifically, the Analytic Free and Parametric Sinc encoders showed comparatively poor performance, with MAE values consistently exceeding 1.7, indicating substantial deviation from ground truth distance estimations. The Custom Encoder, while marginally improving upon the aforementioned approaches, still failed to approach the precision of the STFT, with MAE values around 1.1 to 1.2. The STFT's exceptional performance underscores its effectiveness in capturing and representing the intricate spectral characteristics essential for accurate source distance estimation. This outcome suggests that the STFT's well-established mathematical properties and its ability to provide a time-frequency representation remain unmatched by more novel or flexible encoder architectures in this specific domain.

The STFT encoder significantly outperformed the other encoders, achieving a mean MAE of 0.138 meters. In contrast, the other encoders yielded mean MAE values ranging from 1.170 to 1.861 meters. Given that the average source distance is approximately 4 meters, an MAE of 1 to 1.5 meters corresponds to a substantial relative error of 25

The superior performance of the STFT encoder can be attributed to its comprehensive frequency-domain representation of the audio signal. STFT captures both magnitude and phase information across time and frequency, enabling the model to exploit detailed spectral cues associated with distance-related acoustic phenomena such as reverberation, attenuation, and frequency-dependent absorption.

In contrast, the Analytic Free and Free encoders employ learnable filter banks without fixed filters, relying on the model to discover optimal filters during training. While this approach offers flexibility, it may not effectively capture essential spectral features unless the training data is sufficiently large and diverse, which may not be the case with a dataset of 2,500 samples. The high MAE values for these encoders suggest that they failed to learn the necessary spectral representations for accurate distance estimation.

The Param Sinc encoder uses parameterized sinc functions to create a set of learnable band-pass filters. Although it provides a more structured approach than the Free encoders, it may still lack the ability to capture complex spectral patterns unless carefully tuned. The performance of the Param Sinc encoder was marginally better than the Analytic Free encoder but still inadequate compared to the STFT encoder.

The custom encoder, consisting of four convolutional blocks, showed improved performance over the Free and Analytic Free encoders but did not match the STFT encoder. Convolutional layers are effective at capturing local patterns in the time domain but may not extract the detailed frequency-domain information essential for this task. The relatively high MAE indicates that time-domain convolutions alone are insufficient for precise distance estimation from audio signals.

Interestingly, the Free encoder exhibited significant variability across folds, with folds 3 and 5 achieving lower MAE values of 0.455 and 0.343 meters, respectively. This inconsistency may be due to the encoder's sensitivity to initialization parameters or overfitting to specific subsets of the data. Such instability undermines the reliability of the Free encoder for practical applications.

In summary, the STFT encoder's ability to provide a rich and stable frequency-domain representation makes it the most suitable choice for source distance estimation in audio signals. The other encoders, while theoretically flexible, did not capture the necessary spectral features to achieve low MAE values. Future research should focus on enhancing encoder architectures to combine the detailed spectral analysis of STFT with the adaptability of learnable filters, potentially through hybrid models or advanced regularization techniques to prevent overfitting and improve generalization.

Justification of the STFT's Superior Performance

The effectiveness of the STFT encoder can be further justified by its alignment with the characteristics of the dataset. Since the speech spectrum is predominantly contained within the 0-8 kHz range, the STFT's fixed filters are well-suited to capture the relevant frequency components. Additionally, the shift invariance property of analytic filters ensures robustness to small temporal delays, which is crucial in distance estimation tasks where timing cues play a significant role.

Limitations of Other Encoders

The learned filterbanks (analytic free and free) and parameterized sinc filters may not have been able to generalize well from the dataset due to their reliance on learning filter coefficients from scratch or within a constrained parameter space. This could lead to insufficient representation of the critical spectral and temporal features necessary for accurate distance estimation.

Moreover, the higher MAE values suggest that these encoders might have struggled with capturing phase information, which is essential for determining the time delays associated with different source distances. The inability to adequately model these aspects likely contributed to their poorer performance.

Conclusion

The experimental results demonstrate that the choice of encoder significantly impacts the performance of the CRNN model in source distance estimation tasks. The STFT encoder outperforms other encoders due to its fixed, analytic filters that provide a robust time-frequency representation, capturing essential amplitude and phase information. The custom convolutional encoder, while better than the learned filterbanks, does not match the performance of the STFT, highlighting the importance of utilizing appropriate signal representations that align with the underlying physics of the problem.

These findings suggest that for tasks involving precise acoustic measurements, such as source distance estimation, encoders that leverage fixed, well-understood transformations like the STFT are preferable over learned filterbanks, which may not capture the necessary signal characteristics without substantially more data or additional architectural considerations.

References

1. M. Wölfel and J. W. McDonough, *Distant Speech Recognition.* Hoboken, NJ, USA: Wiley, 2009.

2. J. H. DiBiase, “A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays,” Ph.D. dissertation, Division Eng., Brown Univ., Providence, RI, USA, 2000.

3. T. Rodemann, “A study on distance estimation in binaural sound localization,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 425–430.

4. M. Dietz, S. D. Ewert, and V. Hohmann, “Auditory model based direction estimation of concurrent speakers from binaural signals,” *Speech Communication*, vol. 53, no. 5, pp. 592–605, 2011.

5. P. Zahorik, D. S. Brungart, and A. W. Bronkhorst, “Auditory distance perception in humans: A summary of past and present research,” *ACTA Acustica united with Acustica*, vol. 91, no. 3, pp. 409–420, 2005.

6. C. Mendonça, P. Mandelli, and V. Pulkki, “Modeling the perception of audiovisual distance: Bayesian causal inference and other models,” *PloS one*, vol. 11, no. 12, p. e0165391, 2016.

7. E. Georganti, T. May, S. van de Par, A. Harma, and J. Mourjopoulos, “Speaker distance detection using a single microphone,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 1949–1961, 2011.

8. E. C. Cherry, “Some experiments on the recognition of speech, with one and with two ears,” *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

9. A. Brendel and W. Kellermann, “Distance estimation of acoustic sources using the coherent-to-diffuse power ratio based on distributed training,” in *Proc. IEEE 16th Int. Workshop Acoustic Signal Enhancement*, 2018, pp. 1–5.

10. S. Vesa, “Sound source distance learning based on binaural signals,” in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2007, pp. 271–274.

11. S. Vesa, “Binaural sound source distance learning in rooms,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 8, pp. 1498–1507, Nov. 2009.

12. K. Zhagyparova, R. Zhagypar, A. Zollanvari, and M. T. Akhtar, “Supervised learning-based sound source distance estimation using multivariate features,” in *Proc. IEEE Region 10 Symp.*, 2021, pp. 1–5.

13. E. Georganti, T. May, S. van de Par, and J. Mourjopoulos, “Sound source distance estimation in rooms based on statistical properties of binaural signals,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 8, pp. 1727–1741, Aug. 2013.

14. M. Yiwere and E. J. Rhee, “Sound source distance estimation using deep learning: An image classification approach,” *Sensors*, vol. 20, no. 1, 2019, Art. no. 172.

15. A. Sobhdel, R. Razavi-Far, and S. Shahrivari, “Few-shot sound source distance estimation using relation networks,” 2021, arXiv:2109.10561.

16. R. Venkatesan and A. B. Ganesh, “Analysis of monaural and binaural statistical properties for the estimation of distance of a target speaker,” *Circuits, Syst., Signal Process.*, vol. 39, pp. 3626–3651, 2020.

17. S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, “Sound event localization and detection of overlapping sources using convolutional recurrent neural networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 34–48, Mar. 2019.

18. S. Adavanne, A. Politis, and T. Virtanen, “Localization, detection and tracking of

multiple moving sound sources with a convolutional recurrent neural network," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2019, pp. 20–24.

19. D. Byrne, "The speech spectrum-some aspects of its significance for hearing aid selection and evaluation," *Brit. J. Audiol.*, vol. 11, no. 2, pp. 40–46, 1977.

20. Y. Sudo, K. Itoyama, K. Nishida, and K. Nakadai, "Multi-channel environmental sound segmentation utilizing sound source localization and separation U-net," in *Proc. IEEE/SICE Int. Symp. Syst. Integration*, 2021, pp. 382–387.

21. W. Manamperi, T. D. Abhayapala, J. Zhang, and P. N. Samarasinghe, "Drone audition: Sound source localization using on-board microphones," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 508–519, 2022.

22. Z. Wang, P. Wang, and D. Wang, "Complex spectral mapping for single- and multi-channel speech enhancement and robust ASR," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1778–1787, 2020.

23. A. Pandey and D. Wang, "Exploring deep complex networks for complex spectrogram enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6885–6889.

24. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 448–456.

25. D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.

26. M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018.

27. G. García-Barrios, D. A. Krause, A. Politis, A. Mesaros, J. M. Gutiérrez-Arriola, and R. Fraile, "Binaural source localization using deep learning and head rotation information," in *Proc. 30th Eur. Signal Process. Conf.*, 2022, pp. 36–40.

28. J. Garofolo, "TIMIT acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 1993.

29. A. Politis, "Microphone array processing for parametric spatial audio techniques," Ph.D. dissertation, *Sch. Elect. Eng., Aalto Univ., Espoo, Finland*, 2016.

30. "Sound absorption coefficient chart: JCW acoustic supplies," Accessed: Jun. 17, 2023. *[Online]. Available: https://www.acoustic-supplies.com/absorption-coefficient-chart*