Master's Thesis

Sound Event Localization and Detection for Mobile Robots

Sara Al-Rawi

Examiner: Prof. Dr. Wolfram Burgard
Prof. Dr. Abhinav Valada
Advisers: Jannik Zürn

University of Freiburg
Faculty of Engineering
Department of Computer Science
Autonomous Intelligent Systems

July 08th, 2021

Writing Period

17. 02. 2021 – 08. 07. 2021

Examiner

Prof. Dr. Wolfram Burgard

Second Examiner

Prof. Dr. Abhinav Valada

Advisers

Jannik Zürn

Declaration

I hereby declare that I am the sole author and composer of my thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Place, Date	Signature

Abstract

Mobile robots rely heavily on sensory information to understand the characteristics of the environment. Visual perception is widely used for robot perception; however, it can break down in various situations such as low visibility due to severe weather conditions or faulty visual sensors. To this extend, integrating auditory information becomes essential to increase perceptual precision and reduce ambiguity. To this end, we study sound event localization and detection for polyphonic sound (SELD) as a perceptual alternative in case of the absence of visual perception. We present two novel multi-tasking neural network architectures variants for SELD problem: visual attention-based parameter sharing and squeeze-and-excitation-based parameter sharing; for localizing and classifying sound events. The proposed networks learn the sound's features from multi-channel spectrograms of overlapping real-life sound events. Our experiments show that both networks achieve comparable performance to the current best multi-tasking network for the SELD problem. Furthermore, we demonstrate the networks' robustness against noise by adding real-life noise recorded in exterior environments; our experiments show that both networks' performance stay steady even when the noise ratio is relatively high.

Contents

List of Abbreviations

BCE Binary Cross Entropy 31

CNNs Convolutional Neural Networks 6, 11, 31

dB Decibel 56, 57

DBT Doubly Block-Toeplitz Matrix 14, 17

DCASE Detection and Classification of Acoustic Scenes and Events Workshop and Challenge 5, 8, 9, 50, 59

DoA Direction-of-Arrival 2, 5, 9, 27, 32, 43, 59

DSO Double Soft Orthogonality 14

FC Fully-connected layer 34

FOA First-order Ambisonics 9, 43, 44, 50, 60

GRU Gated Recurrent Neural Unit 7, 23

HMM Hidden Markov Models 6

Hz Herz 43, 54, 55

IR Impulse Response 41

MHSA Multi-head Self-attention 32, 34

MIC Tetrahedral Capsule Arrangement 42–44

MLS Maximum Length Sequence 41

MLT Multi-task Learning 22

MSE Mean Squared Error 31

MUSIC Multiple Emitter Location and Signal Parameter Estimation 6–8

NLP Natural Language Processing 17

OCNN Orthogonal Neural Networks 14, 17, 31

PIT Permutation Invariant Training 33

ReLU Rectified Linear Unit Activation Function 35

RNN Recurrent Neural Networks 7

SED Sound Event Detection 1, 2, 5, 9, 27, 32, 59

SELD Sound Events Localization and Detection iii, 1, 8, 9, 59

SNR Signal-to-Noise Ratio 44, 60

SO Soft Orthogonality 13

STFT Short Time Fourier Transform 29, 30

TCN Temporal Convolutional Network 8

List of Figures

List of Tables

1 Introduction

Humans rely on multiple sources of sensory information to understand their environments, such as vision and audio, which emphasises the importance of multimodality in building a comprehensive and enhanced perceptual system. To such an extent, mobile robots similarly require multimodal sensory information to perceive their surroundings robustly, enabling them to execute tasks in various environments.

Substantial research has been performed in the visual perception field, which plays an essential role in understanding the environment and directly impacting mobile robots tasks such as navigation. However, for the mobile robot to form an intensified perception and to carry out tasks in the environment, the system needs to consider additional modalities such as auditory modality to perceive the acoustic aspects of the environment. Despite the importance of visual perception for mobile robots, visual perception efficiency decreases significantly under severe weather conditions, low visibility, or breaks down entirely in the circumstances of faulty visual sensors. Providing mobile robots with an auditory modality suggests the object's location for stationary objects; moreover, it facilitates the trajectory of the moving objects in the case of their presence without direct visibility. One way to overcome the previously described shortcomings of the visual modality is to distinguish the type and position of sound-emitting objects, commonly referred to as SELD problem (Sound Event Localization and Detection). In literature, the SELD is a long-standing problem and is approached with several methods. The SELD problem includes two tasks that are solved simultaneously. The first task, sound event detection (SED), tries to identify an active sound event and assign it to one category out of several. The second

task is sound event localization realized by the sound direction-of-arrival (DoA) which tries to estimate the location of the sound source relative to the sensor.

This thesis proposes two novel multi-tasking neural network architectures for tackling the SELD problem for polyphonic sound. The networks learn the sound features from segments of a multichannel spectrogram and output the class of the active sound event in the segment and estimate the sound event's location in the Cartesian coordinates system. In the first architecture, we build a shared global feature space base on VGG style[46]; we further utilize a visual attention mechanism to form a features private space for each task; the private spaces contain Transformer layers which enable the network to split the polyphonic sound into distinct tracks; each track includes one sound event. In the second architecture, we build a private feature space for SED task and DoA task; the parameter sharing is maintained through adapted Squeeze and Excitation[21] blocks; the private spaces preserve Transformer layers similar to the first architecture. Moreover, we investigate the impact of imposing orthogonality constraints on the weights of the baseline EINV2[10]. We compared the performance of our architectures against the baseline EINV2 which achieved the best performance in 'DCASE Challenge and Workshop 2020'. Furthermore, we study the robustness of the architectures by corrupting the training audio recordings with real-life noises.

We evaluated our neural network architectures on the challenging **TAU-NIGENS Spatial Sound Events 2020**[34] dataset. The dataset contains audio recordings of different overlapping sound events from various directions and distances. The recordings recorded in various indoor spaces with different sizes, shapes and acoustical properties.

The thesis is structured as follows: in Chapter 2, we demonstrate the earlier methods that tackled the SELD problem. In Chapter 3, we explain the background knowledge which is essential to understand the building blocks of this work. In Chapter 4, we illustrate the technical approach of this work. In Chapter 5, we describe the audio datasets that we used for evaluating the neural network architectures.. Chapter 6 demonstrates the results

of the conducted experiments. Finally, Chapter 7 summaries this work and suggests further future works.

2 Related Work

For the past five years, the 'Detection and Classification of Acoustic Scenes and Events Workshop and Challenge' (DCASE) has attracted many researchers who work on the aspects of environmental sound classification and detection. To enrich the scientific research in this area, the DCASE community provides substantial datasets, including various sound events with different emitting locations; thus, these datasets are suitable for developing and evaluating methods that tackle sound event localization and detection problems. Since this work's main scope includes evaluating different learning methods for the problem mentioned above, this chapter reviews the following: the prominent parametric and deep learning-based approaches, additionally, the multi-task deep learning-based models from DCASE workshop and challenge.

2.1 Sound Event Localization and Detection

The sound event localization and detection (SELD) problem includes two synchronized sub-tasks. The first task (SED) attempts to distinguish an active sound event and classify it into one out of several categories. The second task is the direction of arrival (DoA) which attempts to estimate the location of an active sound event relative to the microphone. The DoA comes in the form of azimuth and elevation; however, deep learning-based approaches yield better results when (DoA) under the Cartesian coordinate system of three-dimensional space. In real-life scenarios, sound events overlap most of the time; a sound consists of more than one event simultaneously, called polyphony. In literature,

the task of detecting and classifying all the overlapping sound events is referred to as polyphonic SED. The SED task is modelled mostly as a multi-label classification problem and approached with different classification methods in a supervised fashion.

Mesaros et al. [28] proposed a sound event detection system based on networks of Hidden Markov Models (HMM) for classifying everyday-life sound events. The system is capable of detecting only the most salient event in a polyphonic sound which results in a non realistic outcome. The authors tested the robustness of the system by adding ambient background noise. The system achieved 24% overall accuracy in classifying 61 classes. Schmidt [36] proposed a multiple signal classification algorithm (MUSIC) for estimating the DoA. MUSIC belongs to the family of subspace methods. Therefore, the MUSIC algorithm depends on the eigenvectors of a covariance matrix computed from a multichannel sound and noise. The algorithm requires the number of overlapping sounds to determine the subspace's covariance matrix. This necessity considered a drawback since it is challenging to provide the number of overlapping sound events.

In recent years and due to deep learning advances, several deep learning-based approaches for tackling SED and DoA problems have emerged. An earlier solution considered a fully connected neural network for classifying polyphonic sound events, the proposed method by Cakir et al. [7] achieved 63.8% overall accuracy on a non public dataset. For tackling DoA problem, Takeda and Komatani [39] considered naïve deep neural networks for localizing the sound events, the authors further included directional information of the sub-bands and designed directional activators to handle complex numbers at each sub-band. Convolutional neural networks (CNNs) is widely considered for computer vision tasks and achieved remarkable success on many tasks such as image classification[17], due to that CNNs are considered for solving both SED and DoA problems. Hirvonen [20] investigated the usage of a CNN architecture that consists of four convolutional layers and three fully connected layers. The proposed method used for microphone array analysis for classifying the sound type either as speech or music. Each sound type further classified to one out of eight locations resulting in the DoA of the sound

event. The network evaluated on a dataset includes different music types and a speech set for speakers in various languages; the network achieved an overall accuracy 94%. Other solutions considered recurrent neural networks, Parascandolo et al. [32] presented a classification model that utilizes a bi-directional long short term memory recurrent neural network (RNN). The authors reported the model's performance on a dataset consists of real-life scenarios recordings 10 to 30 minutes long recorded in ten different contexts[19]. The model achieved an average F1-score of 65.5% on 1-second blocks and 64.7% on single frames. Deep learning-based models further employed for solving DoA problem. Adavanne et al. [4] presented a network called DOAnet; the network consists of CNNs and bi-directional gated recurrent neural units to capture the long term temporal information (GRU). In literature, the neural network architecture which combines CNNs and RNNs known as CRNN. DOAnet sequentially estimates two outputs a) spatial pseudo-spectrum (SPS) approached as a regression problem. SPS resembles the output of the MUSIC method and b) DoA determined by azimuth and elevation approached as a classification problem. The authors evaluated the DOAnet on synthesized static sounds datasets in two contexts anechoic and reverberant, for each context they considered three datasets with no overlapping, and with two and three overlapping sound events. DoAnet outperformed MUSIC by a large margin. Yalta et al. [44] replaced the MUSIC method with a residual network[16] on multichannel sound events, the authors used a dataset consists of recordings from 258 speakers. The authors further evaluated four residual networks with different residual blocks sizes. All the evaluated architectures outperformed the MUSIC algorithm.

2.2 Multi-task Learning for Sound Event Detection & Localization

Deep learning-based models for multi-task learning achieved great success in many machine learning applications such as computer vision[27], speech recognition[18] and

drug discovery[26]. This section reviews the most prominent approaches that tackled the SELD as a multi-task learning problem and presented at the DCASE workshop and challenge. All the reviewed approaches developed and evaluated on public datasets provided by the DCASE community.

Adavanne et al. [2] conducted an extensive study in which they designed a CRNN neural network named (SELDNet) which takes a spectrogram time-frame of a polyphonic sound as input and simultaneously detects and localizes the sound events. The authors approached the SED as a multi-label classification problem and the DoA as a regression problem. The CRNN network evaluated on different datasets which they consist of Ambisonic and Circular array audio formats. The method is further compared with several deep learning-based approaches and outperformed all the earlier approaches by a significant margin. However, the proposed method is incapable of detecting overlapping sounds that belong to the same class. Adavanne et al. [5] further considered the CRNN for detecting and tracking multiple moving sound events. The authors compared CRNN with a parametric method that combines both a) MUSIC[36] to estimate the number of active sound events in the input frame and b) particle filter which plays the role of a tracker. The reported results show the network's ability to detect and track multiple overlapping moving sound events; however, the results further show a higher localization error than the parametric method. As mentioned earlier, the CRNN utilizes RNNs as a building block. RNNs require a high memory buffer, and they are challenging to train them in parallel; thus, it is challenging to deploy these networks on embedded hardware. Coming from this idea Cao et al. [8] proposed a novel network architecture (SELD-TCN) that replaces the RNNs blocks of the CRNN with a temporal convolutional network (TCN). TCN introduced by Oord et al. [31], and it belongs to the CNNs family. TCN blocks rely on dilations to increase the size of convolution receptive fields; thus, the TCN block can capture the long-term dependencies and process the input in a parallel fashion, which gives it an advantage over the RNN. The authors evaluated the SELD-TCN on a sound events dataset 2018 from Tampere University of Technology[3] and compared their novel

architecture against the historic baseline SELDNet. They reported that SELD-TCN outperformed the baseline on all metrics.

Shimada et al. [37] introduced two deep learning-based systems for solving the SELD problem. The first system is a single-stage system that simultaneously estimates the SED and DoA; the authors considered an activity-decoupled Cartesian DoA vector as a target for both SED and DoA. The vector's magnitude represents the probability of an active sound event, while the vector's direction represents the DoA. The second system is a two-stage system that estimated the SED as a first step and employed a transfer learning approach to estimate the DoA. The authors showed that activity-decoupled Cartesian DoA vector improved the SELDNet and evaluated their systems on a challenging dataset consists of overlapping sound events, both stationary and moving[34]. The reported results show that both single-stage and two-stage systems outperform the SELDNet.

Cao et al. [9] presented a novel deep learning model which consists of three branches and outputs: sound event classification (SED), direction of arrival (DoA) and event activity detection (EAD) which attempts to combine the features from SED branch and DoA branch. The authors further introduced a track-wise output for each branch; each track predicts mostly one sound event; this approach enables the model to detect the overlapping sound events that belong to the same class and have different DoAs. Track-wise output arises a common permutation problem solved by adapting permutation invariant training[45]. The authors evaluated their model on[34] dataset and reported their results only on first-order Ambisonics (FOA) format. A detailed description about the dataset in chapter 5. Recently, Cao et al. [10] introduced a novel neural network named (EINV2) achieved the best performance in the DCASE challenge2020. The authors presented a neural network with two branches for SED and DoA in a VGG[38] style; moreover, they employed cross stitch[1] as a soft parameter sharing mechanism between the network branches. Permutation invariant training and a track-wise output from[9] considered for this work. The authors evaluated the EINV2 on a dataset introduced by Politis et al. [34]. The reported results establishes EINV2 as a state-of-the-art.

3 Background

In this chapter, we explain the background knowledge required to understand the building blocks of this work. In our work, we considered neural networks architectures based on Convolutional Neural Networks and Transformers, and we further studied the impact of imposing an orthogonality constraint on the networks' weights.

3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized type of neural networks, and they are designed to work with two and three dimensional inputs such as images and 3D volumina. CNNs utilizes convolution operation to extract the features of a given input signal (e.g. image). The convolution operation, which is the mathematical concept of CNNs, is given by equation 1:

$$(I * K)(i, j) = \sum_{m} \sum_{n} I(m, n) K(i - m, j - n) \tag{1}$$

where I is the input signal, and K is the kernel which contains a set of weights. Convolution is a linear operation, and it involves multiplying the set of weights in K, which is smaller in size than I, by I. The operation is performed systemically by sliding K over I and applying the dot product. A dot product is an element-wise multiplication between input K-sized patch of I and K, which is then summed, resulting in a single value, since K is smaller than I; this enables K to discover a distinct feature across the entire input I.

The result of multiplying K by I multiple times is a two dimensional matrix known as a *feature map*. Figure 1 illustrates the convolution operation. For a given input I multiple kernels K are applied to extract different features.

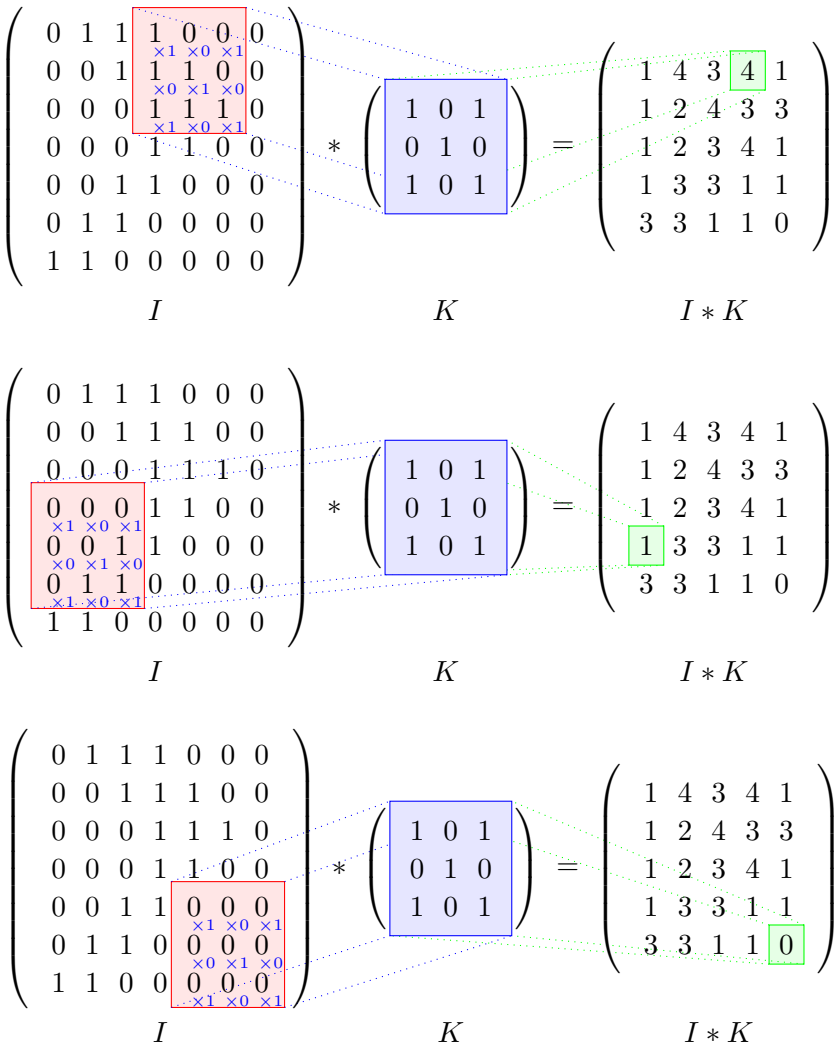


Figure 1: Illustration of Convolution Operation

The operation is applied in a sliding window fashion. The output of multiplying the weights in the kernel K by the input data I is a matrix known as a *feature map*. The convolution operation is applied to the input multiple times results in distinct feature maps.

CNNs proved to be very powerful and are employed broadly for building complex robotics

systems which involve solving different tasks such as object detection and semantic segmentation[47]. Furthermore, CNNs are considered in various application areas such as bioinformatics[35] and medical applications[23].

3.2 Orthogonal Constraints in Convolutional Neural Networks

It has been proved that training very deep convolutional neural networks involves some challenges, such as vanishing and exploding gradients. Furthermore, the underutilization of the network's capacity resulted from features redundancy[12, 15]. Due to that, several approaches have emerged to combat these challenges, including 'Orthogonal Constraints'. Orthogonal Constraints aims to decorrelate the kernels by reducing the feature redundancy, resulting in performance gain compared to non constrained CNNs. This section covers the mathematical concepts of two orthogonal constraints in CNNs by Bansal et al. [6] and Wang et al. [42].

3.2.1 Kernels Orthogonality

Xie et al. [43] proposed an orthogonality regularizer that enforces the Gramm Matrix of the weight matrix to be close to the identity under Frobenius norm F, this type of constraint known as soft orthogonality (SO). SO is given by equation 2 :

$$SO = \lambda \left\| W^T W - I \right\|_F^2 \tag{2}$$

where λ is the regularization coefficient, W is the weight matrix and I is the identity matrix, equation 2 can be seen as a weight decay term that limits the set of parameters close to the Stiefel manifold rather than inside the hypersphere. The gradient is computed straightforwardly by $4\lambda W(W^T W - I)$ and can be appended to the gradients w.r.t W.

Bansal et al. [6] reported in their paper the shortcoming of equation 2 as a minimization objective. The SO is valid for under-complete weight matrix $W(m \geq n)$ which its Gramm matrix W^TW can be close to identity; however, for over-complete weight matrix $W(m < n)$ this is not the case. An orthogonal W must fulfil $W^TW = WW^T = I$ to achieve this, the authors suggested a double soft orthogonality (DSO) which takes in consideration both under-complete weight matrix $W(m \geq n)$ and over-complete weight matrix $W(m < n)$. Equation 3 describes DSO regularizer:

$$DSO = \lambda(\left\|W^TW - I\right\|_F^2 + \left\|WW^T - I\right\|_F^2) \tag{3}$$

3.2.2 Orthogonal CNNs

In the literature, the orthogonal constraints is further investigated. Wang et al. [42] proposed a new method known as orthogonal CNNs (OCNN), which proved to achieve higher performance gain than the kernels orthogonality method. The authors analyzed the convolutional layers' spectrogram, which remains non-uniform even when the kernel matrix is orthogonal. OCNN formulates the kernels K as doubly block-Toeplitz matrix (DBT) $\mathcal{K}$ and connects the input X with the output of the convolution operation Y by a DBT $\mathcal{K}$. Then the orthogonality constraint imposed directly on $\mathcal{K}$ of the kernels K. Block-Toeplitz Matrix can be defined as a matrix in which each descending diagonal from left to right is constant. The convolution operation $Conv$ for a convolution layer can be performed based on DBT as follows:

$$Y = Conv(K, X) \Longleftrightarrow y = \mathcal{K}x \tag{4}$$

where the input $X \in \mathbb{R}^{C \times H \times W}$, C, H and W are the number of channels, hight and width of the input, respectively. Kernel $K \in \mathbb{R}^{M \times C \times k \times k}$ and the output $Y \in \mathbb{R}^{M \times H^{'} \times W^{'}}$. The flattened input X is the vector x. The DBT matrix $\mathcal{K}$ has non-zero entries correspond to a distinct kernel K_i of an input's spatial location. The DBT $\mathcal{K} \in \mathbb{R}^{(MH^{'}W^{'}) \times (CHW)}$

formed from $K \in \mathbb{R}^{M \times C \times k \times k}$. The output Y is obtained by reshaping the vector y to the size $M \times H^{'} \times W^{'}$ where M is a set of kernels. Figure 2 illustrates the convolution operation in which the kernel is formulated as a DBT matrix. The input X of a size $C \times 4 \times 4$, kernel $M \times 2 \times 2$ and stride 1. The main aim of orthogonality constraint is achieving a $\mathcal{K}$ with a uniform spectrum, which results in de-correlated kernels and less feature redundancy.

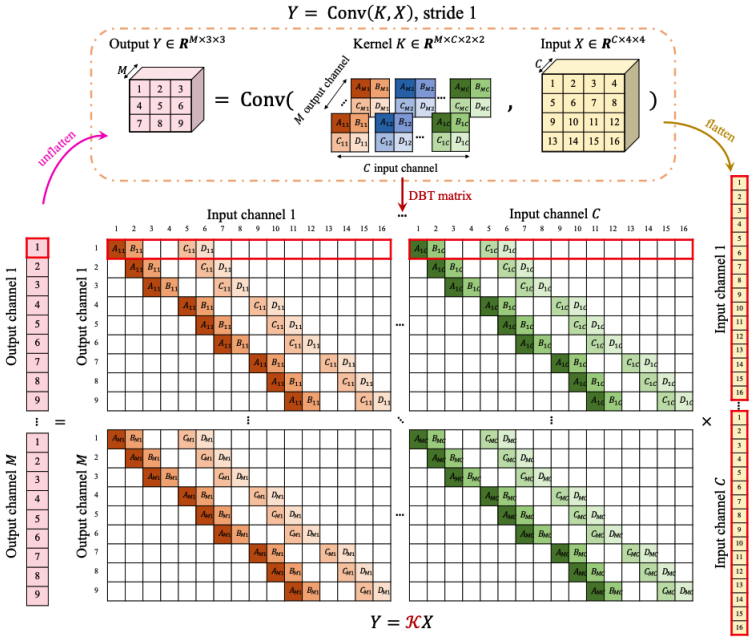


Figure 2: Illustration of Convolution Operation based on block-Toeplitz matrix (DBT)[42].

The input X is flattened to a vector x, the kernel $K \in \mathbb{R}^{M \times C \times k \times k}$ is formulated as DBT matrix $\mathcal{K} \in \mathbb{R}^{(MH^{'}W^{'}) \times (CHW)}$. The output of the convolution operation is $y = \mathcal{K}x$ where y is the vector form of $Y \in \mathbb{R}^{M \times H^{'} \times W^{'}}$. The input X of a size $C \times 4 \times 4$, kernel $M \times 2 \times 2$ and stride 1.

In practise, $\mathcal{K}$ can be a fat matrix $(MH^{'}W^{'}) \leq (CHW)$ or a tall matrix $(MH^{'}W^{'}) > (CHW)$ depending on the convolution layer configuration, thus both row and a column orthogonality are required. Since DBT is a sparse structured matrix, this results in an efficient and non-brute force algorithm for imposing the row and column orthogonality.

As mentioned earlier, each row of $\mathcal{K}$ represents a distinct kernel K_i of an input's spatial location (h', w') and flattened to a vector $\mathcal{K}_{ih',w',\cdot} \in \mathbb{R}^{CHW}$.

The row orthogonality condition is given by equation 5 describes the row orthogonality condition:

$$\langle \mathcal{K}_{ih'_1w'_1,\cdot}, \mathcal{K}_{ih'_2w'_2,\cdot} \rangle = \begin{cases} 1, (i, h'_1, w'_1) = (j, h'_2, w'_2) \\ 0, \text{otherwise} \end{cases} \tag{5}$$

Equation 5 can be realized by the following: for a convolution with a kernel size k and stride S, the region of the input to be examined for orthogonality can be computed by the original convolution with a padding value $P = \lfloor \frac{k-1}{S} \rfloor \cdot S$. To achieve a near-row orthogonal convolution in terms of DBT matrix $\mathcal{K}$, let $Z = Conv(K, K, padding = P, stride = S)$, the row orthogonality objective (roo) to be minimized is:

$$roo = I_{r0} - Z \tag{6}$$

where $I_{r0} \in \mathbb{R}^{M \times M \times (\frac{2P}{S+1}) \times (\frac{2P}{S+1})}$ is an identity matrix has zero entries except for the center $M \times M$ region. Column orthogonality can be achieved first by obtaining the column $\mathcal{K}_{\cdot,ihw}$ of $\mathcal{K}$

$$\mathcal{K}_{\cdot,ihw} = \mathcal{K}\mathbf{e}_{ihw} = Conv(K, E_{i,h,w}) \tag{7}$$

where $\mathbf{e}_{ihw}$ is a flattened vector of the input $E \in \mathbb{R}^{C \times H \times W}$ which has zero entries except for i^{th} channel of the location (h, w). Equation 8 describes the column orthogonality condition:

$$\langle \mathcal{K}_{\cdot,ih_1w_1}, \mathcal{K}_{\cdot,ih_2w_2} \rangle = \begin{cases} 1, (i, h_1, w_1) = (j, h_2, w_2) \\ 0, \text{otherwise} \end{cases} \tag{8}$$

Similar to row orthogonality, the region of the input to be examined for orthogonality can be computed by the original convolution with a kernel size k and padding value $P = k - 1$, to achieve near-column orthogonal convolution, let $Z = Conv(K^T, K^T, padding = k -$

$1, stride = 1$), the column orthogonality objective (coo) to be minimized is:

$$coo = I_{c0} - Z \tag{9}$$

where $K^T \in \mathbb{R}^{C\times C\times(2k-1)\times(2k-1)}$, and $I_{r0} \in \mathbb{R}^{C\times C\times(2k-1)\times(2k-1)}$ is an identity matrix has zero entries except for the center $C \times C$ region. In practise, the column or row orthogonality is imposed during training depending whether the DBT is tall or row matrix. The authors claimed that OCNN is enough to achieve the orthogonality in CNNs; however, we considered both orthogonality approaches in our experiments to create an additional baseline named EINV2-7C; EINV2-7C outperformed the baseline EINV2.

3.3 Attention Mechanism

Attention is a technique that enables the neural network to enhance essential parts of the input, such as words in natural language processing (NLP) tasks or parts of the image in computer vision tasks, by placing higher importance weights on a subset of the features. This section covers the employment of attention mechanism in both NLP tasks and computer vision tasks.

3.3.1 Transformer

Transformer architecture proposed by Vaswani et al. [41] achieved state-of-the-art results on machine translation tasks. The transformer's fundamental concept utilizes a self-attention mechanism, enabling the model to incorporate the previous input in the current representation; additionally, the authors refined the self-attention by adding so-called multi-head attention. The architecture consists of two components: encoder block and decoder block. The encoder is composed of a self-attention and feed-forward layer. In contrast, the decoder consists of an additional self-attention to apply to the encoder's output. The authors further added residual connections for both the encoder and decoder.

Figure 3 illustrates the default transformer's architecture; the original architecture is composed of six blocks of encoders and decoders (N = 6).

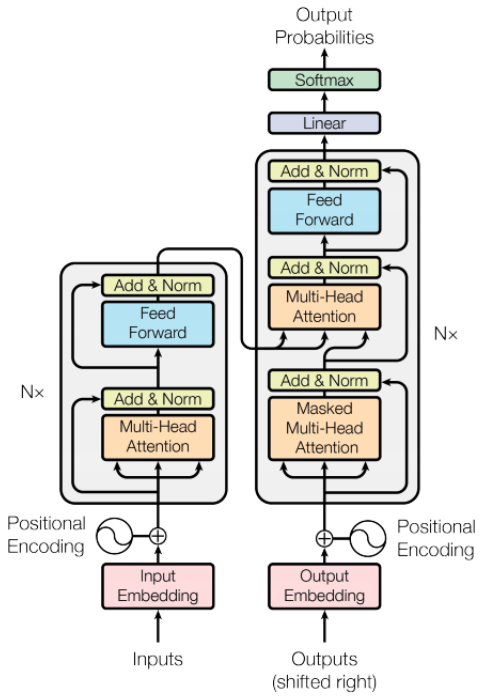


Figure 3: Illustration of the Transformer's Architecture[41].

The left side of the transformer represents the encoder. The encoder is composed of a self-attention and feed-forward layer, besides two residual connections. The right side represents the decoder identical to the encoder except having encoder-decoder attention and three residual connections. N corresponds to the number of encoders and decoders blocks. The original architecture consists of 6 encoders and decoders blocks.

As mention earlier the transformer implements the attention technique as a self-attention operation and further improves it with a multi-head approach. Self-attention determined by first generating for each input (e.g. word) three vectors: query, key and value. The query and key vectors of dimension d_k; the value vector of dimension d_v. These vectors are the output of projecting an input sequence $X \in \mathbb{R}^{n \times d}$, where n represents the tokens with d dimensions, using three trainable matrices. The original implementation by Vaswani

et al. [41] considered vectors of dimension 64, which make them eight times smaller than the encoder's/decoder's input/output. As a next step, the self-attention computes each element's score in the input sequence (e.g. word) against the rest of the sequence. The score is the dot product between the query q and key k vectors. The score is further divided by the square roots of key vectors' dimension $\sqrt{d_k}$; the scores are normalized with a softmax function. The softmax output used to obtain a weighted value vector. Equation 11 describes what so-called *scaled dot-product attention*. The authors contributed to (multiplicative attention) by adding the scaling part $\frac{1}{\sqrt{d_k}}$, which stabilizes the gradients. In practise, the self-attention is computed simultaneously on matrix form, which means the queries, keys and values are packed into matrices Q, K and V; respectively, and they are obtained using three trainable weight matrices $W_Q \in \mathbb{R}^{d \times d_k}$, $W_K \in \mathbb{R}^{d \times d_k}$ and $W_V \in \mathbb{R}^{d \times d_v}$; as described in equation 10

$$Q = XW_Q, K = XW_K, V = XW_V \tag{10}$$

The Scaled Dot-Product Attention is computed as follows:

$$SelfAttention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{11}$$

As we have mentioned earlier, self-attention is supported with a multi-head attention mechanism. Multi-head attention expands the representation subspace of the self-attention by creating multiple query Q, key K and value V matrices. The multiple matrices are linearly projected numerous times, based on a hyper-parameter h, with different learned projections to d_q, d_k and d_v dimensions, respectively. On the projected matrices, the scaled-Dot-Product is performed in parallel, resulting in outputs of dimension d_v, which is concatenated and further linearly projected, producing the final output. Figure 4 illustrates the multi-head attention mechanism.

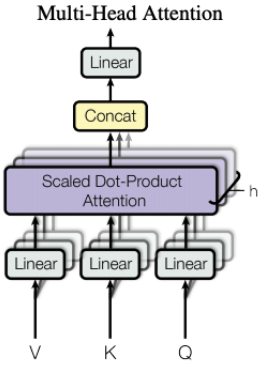


Figure 4: Illustration of Multi-headed Attention Mechanism[41].

The Q, K and V are query, key and value matrices, respectively. The Scaled Dot-Product Attention is performed h times in parallel, the output is concatenated and further projected, producing the final output.

3.3.2 Visual Attention

Visual attention can be categorised into soft and hard attention. Soft attention can be learned using gradient descent and has a value between 0 and 1, unlike hard attention, which is either 0 or 1; therefore, it can not be learned with gradient descent since it has no derivative.

Jetley et al. [22] presented the approach of soft visual attention and applied it to a multi-class classification task. The authors showed that soft visual attention improved the performance of VGG[46] by 7% on CIFAR dataset[25] on a multi-class classification task. The approach can be summarized as follows: on a specific convolution layer, an attention mask will be applied to suppress the irrelevant features. The attention mask has a value between 0 and 1 and generated by passing the convolution layer's output through an attention estimator. To generate attention aware features, element-wise multiplication applies between the attention mask and the convolution layer's original output. Formally, let $x \in \mathbb{R}^d$ be an input vector (e.g. image), $l_i \in \mathbb{R}^k$ local features

generated by a convolution layer for the spatial position i, $f_\theta(x)$ is the convolution layer with the parameter θ, $a \in [0,1]^k$ is the attention mask, which is also known as attention weight, $g_a \in \mathbb{R}^k$ is the final output of the attention mechanism. Typically, the attention mechanism computed as described in equation 12:

$$\begin{aligned} l_i &= f_\theta(x), \\ a_i &= f_\varphi(l_i), \\ g_a &= a_i \odot l_i \end{aligned} \tag{12}$$

where f_φ is the attention mask estimator with the parameter φ and $\odot$ denotes the element-wise multiplication. Soft attention is further progressed by considering the neural network's global features in attention mask estimation. For this purpose, a few approaches are examined, including but not limited to performing element-wise multiplication between l_i and the network's global features. Liu et al. [27] presented a sophisticated neural network architecture known as Multi-Task Attention Network (MTAN), which combines soft attention mechanism and multi-task learning approach; MTAN employed for solving semantic segmentation and depth estimation tasks. The network consists of a single global feature network and K task-specific private networks; the task-specific networks based entirely on a set of attention modules.

MTAN utilizes attention masks as feature selectors and applies them to the features learned in the global network, which enables the task-specific networks to learn task-specific features. The first attention module in a task-specific network described in equation 13, where p^j is j^{th} block in the global feature network, a_i^j is the attention mask for the task i and applies on the global block j. $\hat{a_i^j}$ is the task-specific features computed by element-wise multiplication of the attention mask with the global feature.

$$\hat{a_i^{(j)}} = a_i^{(j)} \odot p^{(j)} \tag{13}$$

where $\odot$ denotes the element-wise multiplication. The attention module's input is the

features learned in the global network. However, for the subsequent modules the input is generated by concatenating the global features u^j and the previously computed task-specific features $a_i^{\hat{j-1}}$; for $j \geq 2$ the attention mask a_i^j is computed as described in equation 14:

$$a_i^{(j)} = h_i^{(j)} \left(g_i^{(j)} \left([u^{(i)}; f^{(j)}(a_i^{\hat{j-1}})] \right) \right), j \geq 2 \tag{14}$$

where $g_i^{(j)}$ and $h_i^{(j)}$ combined are the attention mask estimator. The $h_i^{(j)}$ includes the non-linear function sigmoid to ensure that the learned attention mask $a_i^{(j)} \in [0, 1]$. As a result the learnt soft attention masks partially depend on the features learned in the global network.

3.4 Multi-task Learning for SELD

Multi-task Learning (MLT) led to successes in many deep-models based applications, including but not limited to robotics, computer vision and drug discovery. MLT aims to solve multiple tasks by benefiting from a shared representation. MLT can be motivated from a biological perspective; animals can learn a new task by integrating the knowledge obtained by learning tasks from different domains. Furthermore, we can motivate MLT from a machine learning point of view; due to a shared representation, MLT introduces inductive bias, which alleviates overfitting in deep models. Learning multiple tasks is a non-trivial problem since the tasks conflict most of the time, which means a shared representation will not be beneficial for all tasks and improving one task's performance might harm the others. This phenomenon is known by negative transfer[13]. Such challenges led the researchers to explore answering the following questions: *what to share? Furthermore, how to share?* A few popular approaches have emerged trying to balance the amount of sharing between the tasks to enable the deep model to leverage a performance improvement. The following sections cover the fundamental approaches upon which this work built.

3.4.1 Hard Parameter Sharing

Hard parameter sharing is the most used approach in MLT. In general, this approach consists of a set of shared hidden layers between the tasks, the shared layers split later to task specific layers. Figure 5 illustrates the hard parameter sharing approach in deep-based models.

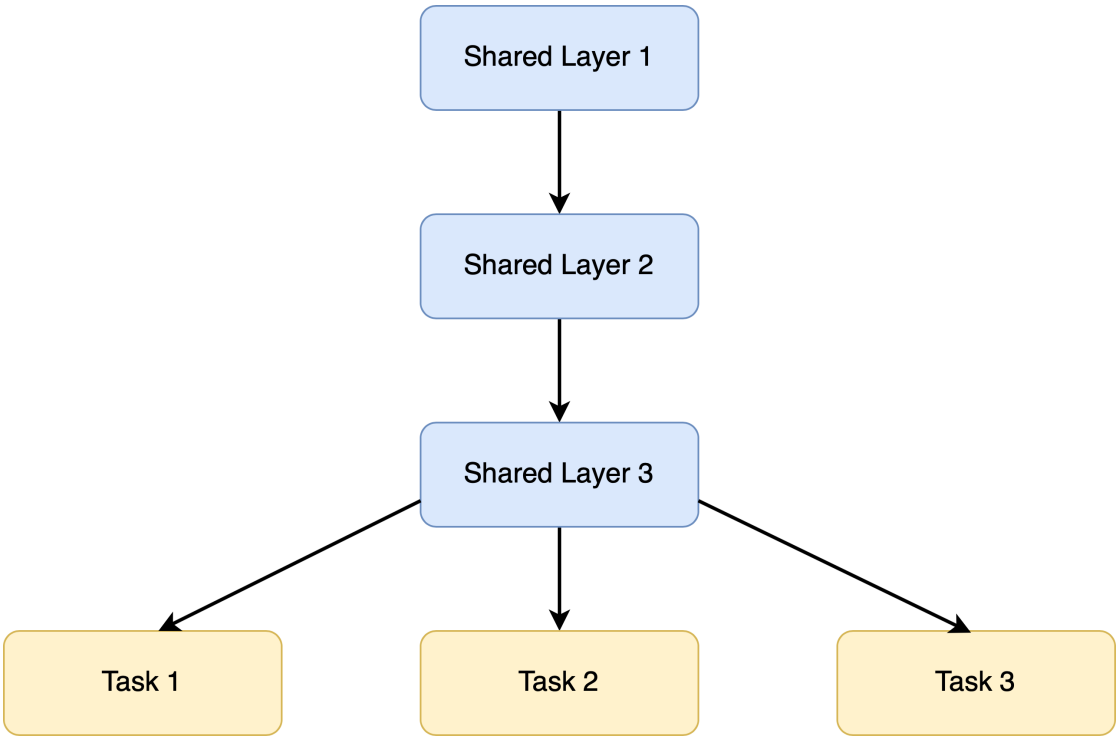


Figure 5: Illustration of Hard Parameter Sharing in Deep-based Models.

The figure illustrates a hard parameter sharing approach for multi-task learning in deep-based models. The shared layers are illustrated in light blue and task-specific layers are illustrated in light yellow.

In the context of SELD problem, most of the proposed solutions are based on hard parameter sharing approach with different architectural variation. Figure 6 illustrates SELDNet on the left and SELD-TCN on the right. The SELDNet proposed by Adavanne et al. [2] composed of three shared convolutional blocks, two GRU bi-directional and two

task specific full-connected layers. SELD-TCN proposed by Guirguis et al. [14] composed of three shared convolutional blocks, one TCN layer and two task specific full-connected layers.

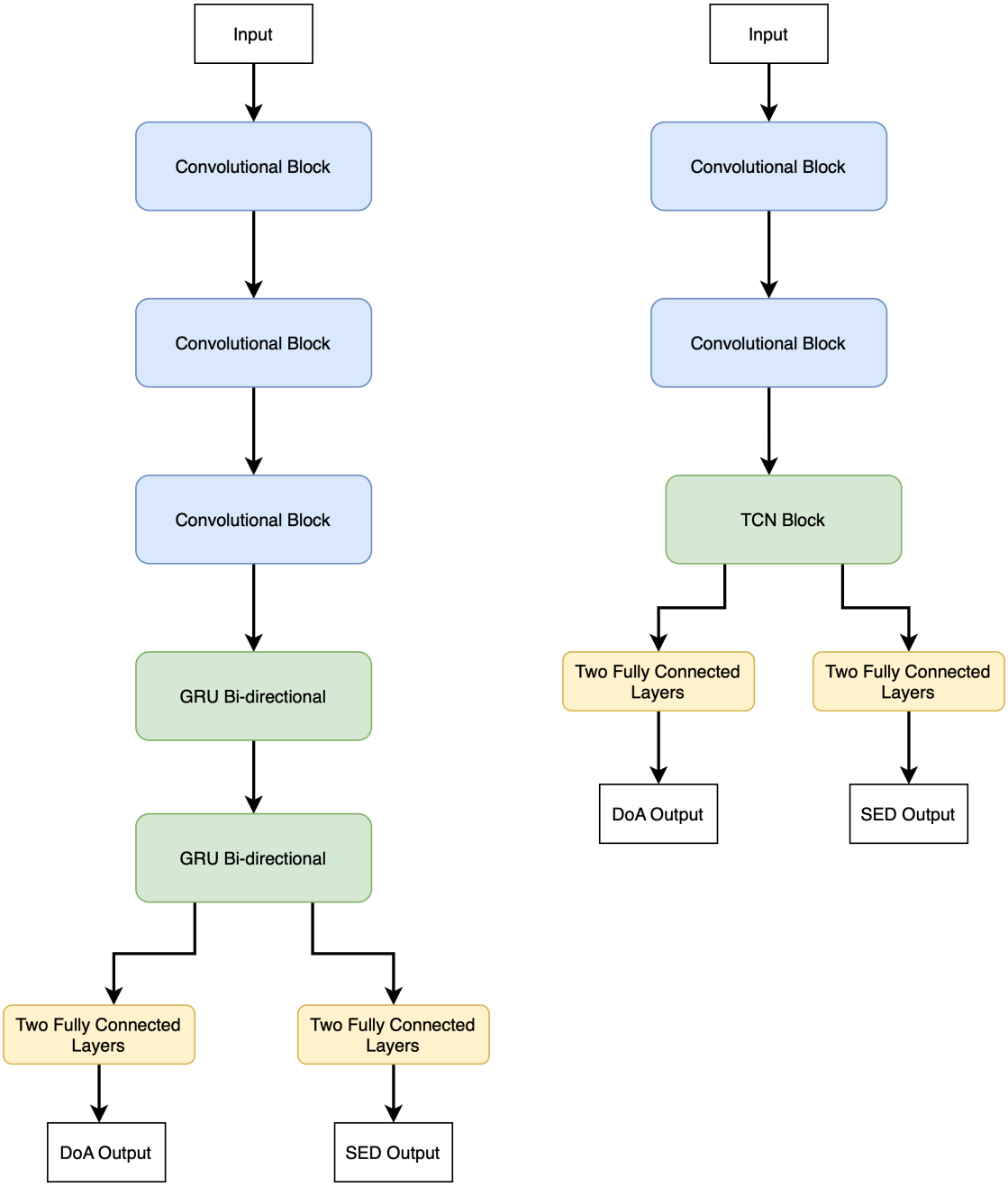


Figure 6: Illustration of SELDNet (left) and SELD-TCN (right).

The SELDNet (left) composed of three shared convolutional blocks, two GRU Bi-directional and two task specific fully connected layers[2]. SELD-TCN composed of three shared convolutional blocks, one TCN layer and two task specific full-connected layers[14]. The shared layers are illustrated in light blue, special shared block (GRU Bi-directional and TCN Block) illustrated in light green; the task-specific layers illustrated in light yellow.

3.4.2 Soft Parameter Sharing

Soft parameter sharing approach is composed of individual layers for each task and information flow between the task-specific branches in parallel to the task-specific input. Cross-stitch proposed by Mis [1] is an example for this approach. The cross-stitch approach suggests maintaining a network for each task and parameter sharing units between the networks, and the cross-stitch units combine the activations from the networks of multiple tasks, such that the input to each layer is a linear combination of the output of the previous layers from every task network. The linear combinations have a learnable task-specific weight α, such that the influence of the information coming from different tasks is controlled. Figure 7 illustrates the Cross-stitch approach as a soft sharing mechanism. The figure shows two networks; Network A and B for each task and five cross-stitch units that enable the information flow between the networks.

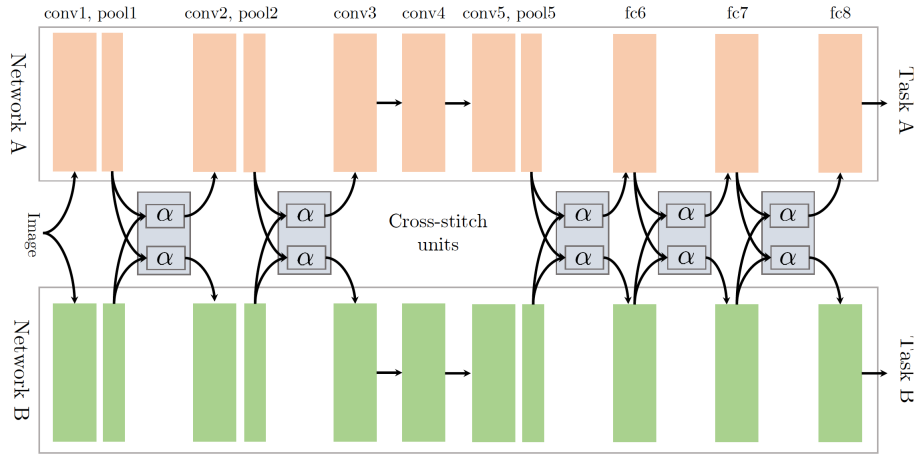


Figure 7: Illustration of the Cross-Stitch Units with Two AlexNet Networks[1].

The network consists of two branches (networks) for task A and B and cross-stitch units. The branches share the information through cross-stitch units; α denotes the learnable task-specific weight.

4 Approach

In this chapter, we describe our proposed approaches for tackling the SELD problem. We start by describing the auditory data preprocessing and feature extraction; we describe two proposed multi-task architectures: visual attention-based parameter sharing and squeeze-and-excitation-based parameter sharing.

4.1 Problem Definition

Audio is described by the attributes amplitude, frequency and timbre, which give each audio a fingerprint. We opted for sound features extracted from a spectrogram representation since studies have shown that spectrogram representation is more suitable for classification problems than other features[24]. We formulated the SELD as a multi-task learning problem; our aim is neural network architectures that enable efficient parameter sharing between the SED and DoA tasks such that a performance gain can be observed for both tasks. We considered spectrogram segments as the input and we trained both architectures in an end-to-end fashion. Our approach can be summarized as follows:

1. The multi-tasking nature of the SELD problem and the design of the baseline EINV2 that comprises two identical branches tend to introduce feature redundancy, resulting in the network's capacity underutilization. To overcome this challenge, we imposed orthogonality constraints on the weights of the EINV2 network, which resulted in a performance gain.

2. Our first variant neural network is visual attention-based parameter sharing inspired by Liu et al. [27]. The network is comprised of a shared global feature space based on VGG[46] style and two identical private feature spaces. The private spaces are built by stacking four attention modules to learn task-specific features. They share features with the global feature space; we further preserved the Transformer layers similar to the baseline EINV2[10]; the Transformer layers separate the sound events into distinct tracks such that the network can predict overlapped sound events with different direction-of-arrivals. The output format of the network follows the same approach considered by EINV2, which explained in the following sections.

3. Our second variant neural network is squeeze-and-excitation-based parameter sharing. We adapted Squeeze-and-Excitation blocks (S&E)[21] as a sharing mechanism, which replace the cross-stitch units considered in EINV2.

4. Since this work intended to be used in outdoor environments, we conducted extensive experiments to study the impact of real-life noise on the networks' performance. We added noise recorded in exterior environments with different signal-to-noise ratio settings.

4.2 Audio Preprocessing and Feature Extraction

We segmented each audio signal into 15 equal length segments. For each segment, we computed a *Hann* window function with a length of 1024 and hop length 600. Hann window is defined as follows:

$$w(n) = 0.5\left(1 - cos\left(2\pi\frac{n}{M}\right)\right), 0 \leq n \leq M-1 \tag{15}$$

where M is the number of samples in the segment. Figure 8 illustrates the waveform of an audio segment of four channels, where the x-axis represents the audio's samples and

the y-axis represents the audio's amplitude.

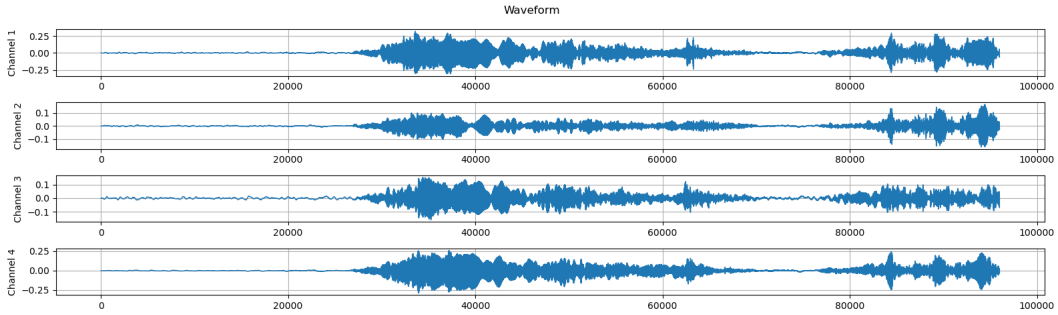


Figure 8: Illustration of an audio segment of four channels. The audio segment is taken from the validation fold of FOA format dataset.

The x-axis represents the samples in the audio and the y-axis represents the amplitude.

As a next step, we applied Short Time Fourier Transform (STFT) to generate the audio's spectrogram representation. Discrete Short Time Fourier Transform $\mathcal{X}$ is defined in equation 16:

$$\mathcal{X}(w) = \sum_{n=-\infty}^{+\infty} x(n) \exp^{-jwn} \tag{16}$$

where x real-valued discrete-time audio signal obtained by equidistant sampling with respect to a fixed sampling rate, in our case sampling rate is 24000, $w \in (-\infty, +\infty)$ is a Hann window function. The spectrogram is a two-dimensional representation of the squared magnitude of the STFT is realized by equation 17:

$$\mathcal{Y}(w) = |\mathcal{X}(w)|^2 \tag{17}$$

Finally, we converted the spectrogram to Mel scale in a decibel (dB) unit. Moreover, we computed a *3D Sound Intensity Vector* which can be defined as the power per unit time through a unit area that is perpendicular to the direction in which the sound wave is travelling. As mentioned earlier, we are dealing with audio consists of four channels w, x, y and z which they receive the signal equally from all directions; a *Sound Intensity*

Vector can be defined as follows:

$$\mathbf{I} = \rho \mathbf{v} \tag{18}$$

where ρ is the sound pressure and $\mathbf{v}$ is the particle velocity which can be defined as $w, \mathbf{v} = (v_x, v_y, v_z)^T$. We computed the intensity vector from STFT and on the Mel scale as follows:

$$\begin{aligned} \mathbf{I}(f,t) &= \frac{1}{c\rho} \Re \left\{ W^*(f,t) \cdot \begin{bmatrix} X(f,t) \\ Y(f,t) \\ Z(f,t) \end{bmatrix} \right\}, \\ \mathbf{I}^{norm}(k,t) &= -\mathbf{H}(k,t) \frac{\mathbf{I}(f,t)}{\|\mathbf{I}(f,t)\|_2} \end{aligned} \tag{19}$$

where c and ρ are the density and velocity of the sound, X, Y, Z are the STFT of x,y, and z, respectively, $\Re$ denotes the real part, $*$ indicates the conjugate, $\|.\|_2$ is the l_2 norm of the vector. k indicates the index of the bin, and $\mathbf{H}$ is the Mel filter bank. Adding a *3D Sound Intensity Vector* makes the networks' input size equal to seven. The *3D Sound Intensity Vector* is indicated in the figures 10 and 12 as 3D intensity vector.

The spectrogram can be visualized through a two-dimensional image, where the x-axis represents time, and the y-axis represents frequency. The parameters of the audio preprocessing have been chosen empirically based on earlier various approaches[9, 2]. Figure 9 illustrates the spectrogram of an audio segment. The x-axis represents the frame over time and y-axis represents the number of frequency bins=256.

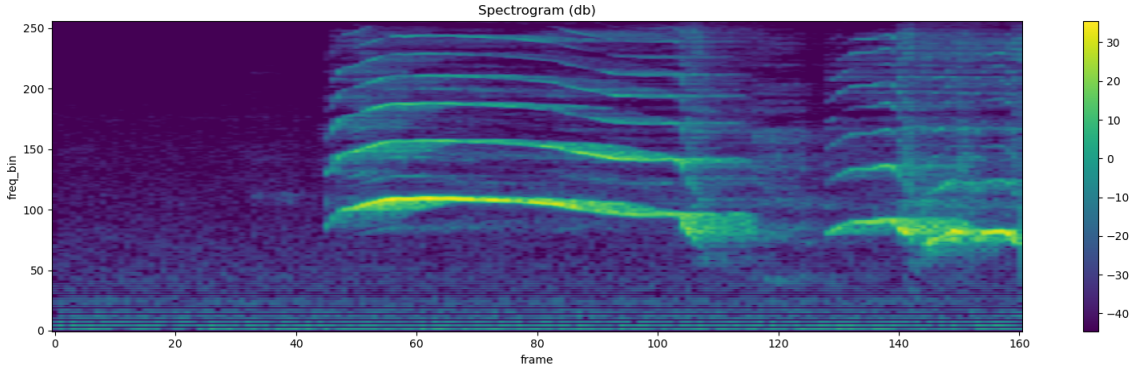


Figure 9: Illustration of the spectrogram in Mel scale for the audio's first channel; taken from the validation set of FOA format dataset.

The x-axis represents the frame over time the y-axis represents the number of frequency bins; in our case frequency bins=256.

4.3 Constrained Baseline

In Chapter 3, we explained the challenges in training deep CNNs. We are mainly concerned with the underutilization of the neural network's capacity resulting from feature redundancy in CNNs. To combat this challenge, we utilized combined orthogonality constraints OCNN by Wang et al. [42] and orthogonal kernels by Bansal et al. [6]. The constraints implemented as follows: for each convolution layer in EINV2, we computed a combined orthogonal distance of OCNN and kernel orthogonality. Summing up the layers' combined orthogonal distances results in an orthogonal loss $\mathcal{L}_{orth}$. We added the weighted loss $\alpha\mathcal{L}_{orth}$ to the total loss given by equation 20.

$$\mathcal{L}_{total} = \mathcal{L}_{SED} + \lambda\mathcal{L}_{DoA} + \alpha\mathcal{L}_{orth} \tag{20}$$

where $\mathcal{L}_{SED}$ is a binary cross-entropy loss (BCE) for sound event detection task (SED) and $\mathcal{L}_{DoA}$ is the mean squared error (MSE) for sound event localization task (DoA); λ, α

are weight coefficients. Equation 21 and 22 are BCE and MSE, respectively.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \tag{21}$$

where N is the number of training samples, y is the ground truth, which is either 0 or 1 and $\hat{y}$ is the predicted probability of an active sound event.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{22}$$

where N is the number of training samples, y is the ground truth and $\hat{y}$ is the predicted location of an active sound event. $\mathcal{L}_{orth}$ is DSO realized from equation 3 or is row or column orthogonality objective realized from equation 8 and 9, respectively.

4.4 Visual Attention-based Parameter Sharing

We are dealing with a multi-channel spectrogram which is fundamentally a multi-channel two-dimensional image. Consequently, visual attention-based parameter sharing is a promising mechanism for the SELD problem. Our visual attention-based parameter sharing network for SELD (VASELD) is comprised of a global shared feature space based on VGG convolutional encoder style[46] and two private spaces. The private spaces are identical, and each is comprised of four attention modules.The global shared feature space aims to learn general features shared between SED and DoA, while the attention modules in the private spaces filter out irrelevant features for each task. Additionally, we preserved the multi-head self-attention layers (MHSA). MHSA separate the output of the attention modules into tracks according to the maximum number of overlaps in the dataset; in our case, the maximum number of overlaps is two.

Each track in the SED space outputs one active class out of 14, and each track in the DoA branch outputs one location in the Cartesian coordinates (x, y, z) for an active classified

sound event. Track-wise output format introduced by Cao et al. [11] and described in equation 23:

$$Y_{track} = \{(y_{SED}, y_{DoA}) \mid y_{SED} \in \mathbb{1}_S^{M \times K}, y_{DoA} \in \mathbb{R}^{M \times 3}\} \tag{23}$$

where y_{SED} and y_{DoA} are track-wise prediction for classification and localization tasks, respectively. $\mathbb{1}_S$ is one-hot encoding for K classes, M is the number of tracks where $M \ll K$. As mentioned in chapter 2, the track-wise output format leads to a permutation problem, since the tracks are not reserved per class. Therefore, the network is trained with frame-level permutation invariant loss (tPIT). PIT is the minimum loss for all possible predictions at the frame t. The tPIT, $\mathcal{L}^{PIT}(t)$ is defined in equation 24

$$\mathcal{L}^{PIT}(t) = \min_{\alpha \in P(t)} \sum_{M} \{\mathcal{L}_{SED}(t) + \mathcal{L}_{DoA}(t)\} \tag{24}$$

where M is number of tracks, in our case $M = 2$ and $\alpha \in P(t)$ is a permutation pair. We further decorrelated the SED and DoA private spaces by encouraging the orthogonality and we compute what so-called loss difference denoted by $\mathcal{L}_d$. Equation 25 and 26 define $\mathcal{L}_d$ and VASELD total loss, respectively.

$$\mathcal{L}_d = \left\| G_{SED}^T G_{DoA} - I \right\|_F^2 + \left\| G_{SED} G_{DoA}^T - I \right\|_F^2 \tag{25}$$

where G_{SED} and G_{DoA} are the weights learned in the SED and DoA private spaces, respectively. I is the identity matrix and F is Frobenius norm. VASELD total loss:

$$\mathcal{L}_{total} = \mathcal{L}_{SED} + \lambda \mathcal{L}_{DoA} + \beta \mathcal{L}_d \tag{26}$$

where $\mathcal{L}_{SED}$ and $\mathcal{L}_{DoA}$ is the tPIT, λ, β are the weights coefficients, and $\mathcal{L}_d$ is loss difference. Figure 10 illustrates the VASELD architecture, where the input is a four-channel spectrogram in Mel scale with a 3D intensity vector. The figure illustrates a global shared feature space in blue, identical private spaces for the SED task in red and

DoA task in green. Each private space consists of four attention modules with a track-wise output format, consisting of a multi-head self-attention (MHSA) and a fully-connected layer (FC) for each output track. The network trained with permutation invariant loss on the frame-level (tPIT), which is shown in dotted red boxes, and loss difference denoted by $\mathcal{L}_d$; shown in a dotted red box.

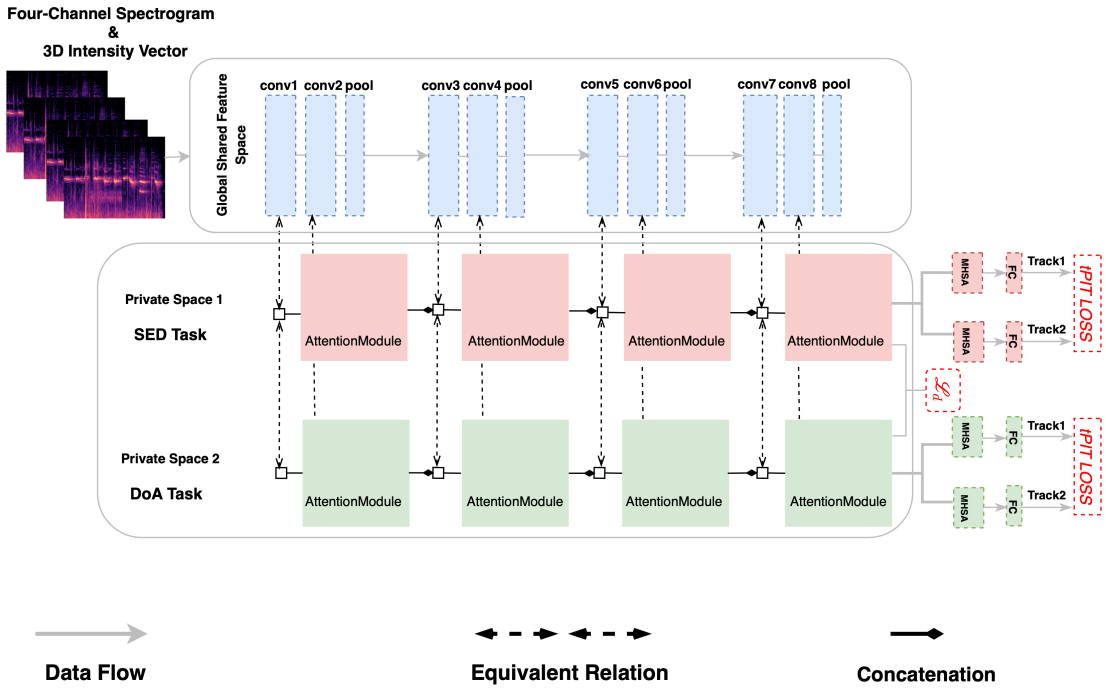


Figure 10: Illustration of the Visual-based parameter Sharing Architecture for SELD (VASELD).

The input is a four-channel spectrogram in Mel scale with a 3D intensity vector. The figure illustrates a global shared feature space in blue, identical private spaces for the SED task in red and DoA task in green. Each private space consists of four attention modules with a track-wise output format, consisting of a multi-head self-attention (MHSA) and a fully-connected layer (FC) for each output track. The network trained with permutation invariant loss on the frame-level (tPIT), which is shown in dotted red boxes, and loss difference denoted by $\mathcal{L}_d$; shown in a dotted red box.

Figure 11 illustrates the attention module architecture bounded in black. The module's

input is formed by merging the output from the previous module with a convolution layer output from the global feature space. Each attention module comprises of attention mask estimator, which is g and h. The attention mask estimator consists of two convolution layer with kernel size 1×1, two batch normalization layer, an activation function ReLU, and a non-linear Sigmoid function to ensure that the learned mask a has a value between 0 and 1. An element-wise multiplication is performed between a and the learned features in the global feature space p to generate the task-specific feature $\hat{a}$. The f function consists of a convolution layer with kernel size 3×3, batch normalization layer, an activation function ReLU, and average pooling is applied to adjust the output's size. The only difference between the modules is the size of the average pooling layers.

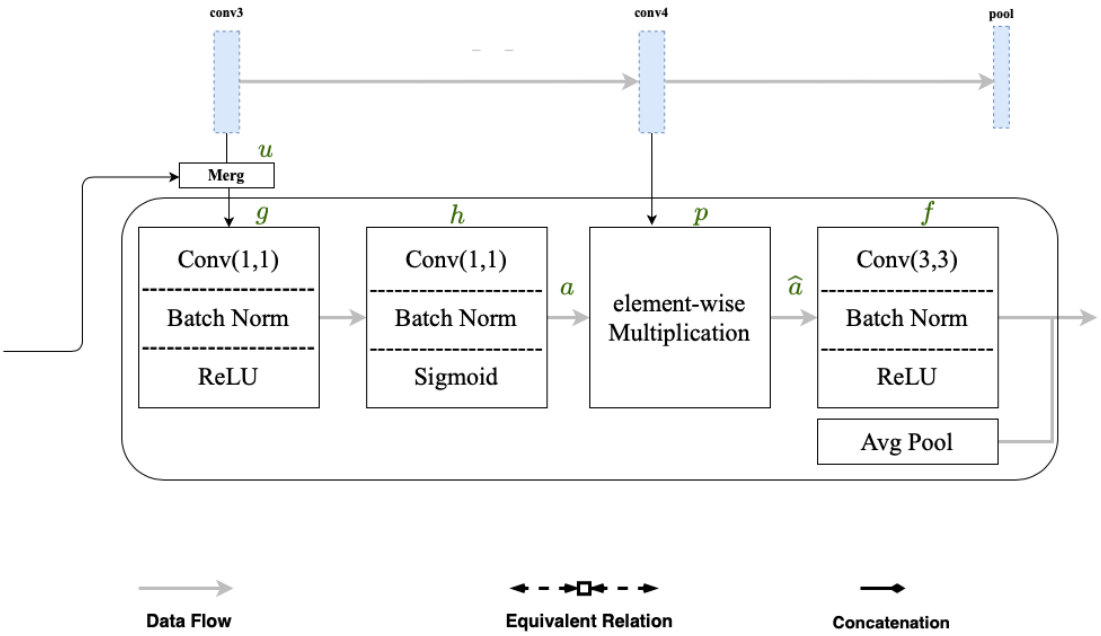


Figure 11: Illustration of the attention module of the VASELD.

Conv denotes convolution layers, Batch Norm denotes batch normalization layer and Avg Pool denotes average pooling. g and h combined are the attention mask estimator. h includes the non-linear function Sigmoid to ensure the learned attention mask a is between 0 and 1 and p is the features learned in the global feature space. Element-wise multiplication is performed between a and the learned features in the global feature space p to generate the task-specific feature $\hat{a}$. Finally, the f function consists of a convolution layer with kernel size 3×3, batch normalization layer, an activation function ReLU, and average pooling is applied to adjust the output's size.

4.5 Squeeze-and-Excitation-based Parameter Sharing

In multi-task learning, the performance of the deep-based model is highly affected by the parameter sharing mechanisms. To this extent, we opted for replacing the cross-stitch units in EINV2 with S&E blocks.

S&E block can be considered a type of attention mechanism for feature recalibration, and it is concerned with modelling the relationship between the channels of the convolution

layer, which allows the neural network to learn to focus on task-relevant features and neglect the task-irrelevant features . Since deep-based models are data-hungry, S&E block has very little overhead regarding increasing the network's capacity. In our work, we adapted the original block S&E as illustrated in figure 13. Moreover, the novel network S&ESELD is illustrated in figure 12. The S&ESELD is comprised of two identical branches, one for classification task (SED) and one for localization task (DoA). Each branch based on VGG style and includes two MHSA layers for tracks separation as mentioned earlier.

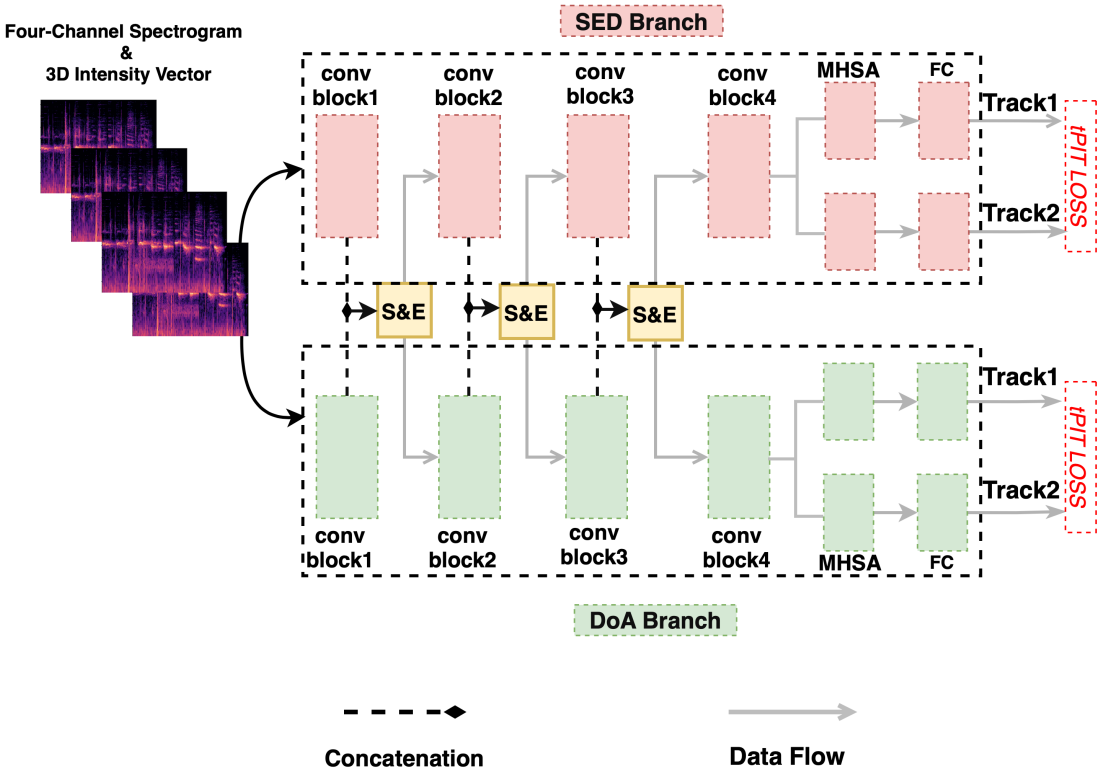


Figure 12: Illustration of the Squeeze-and-Excitation-based Parameter Sharing Architecture for SELD (S&ESELD).)

The input is a four-channel spectrogram in Mel scale with a 3D intensity vector. The figure illustrates two identical branches SED in red and DoA in green. Each branch consists of four convolutional blocks (conv block) with a track-wise output format, further, consisting of a multi-head self-attention (MHSA) and a fully-connected layer (FC) for each output track. Each convolutional block consists of two convolution layers, batch normalization and ReLU activation function. The Squeeze and Excitation blocks (S&E) illustrated in yellow boxes as they connect the branches. The network trained with permutation invariant loss on the frame-level (tPIT), which is shown in dotted red boxes.

The S&ESELD branches share their parameters through S&E blocks, which operate as follows: the input to each block is the channel-level concatenation of SED and DoA outputs; we denote the set of feature maps by $U \in \mathbb{R}^{H \times W \times C}$. U passes through a *squeeze operation*, which aggregates the feature maps across the spatial dimensions ($H \times W$)

producing a channel descriptor of size $(1 \times 1 \times C)$. *Squeeze operation* considered a cheap operation, since it does not increase the number of parameters. Then the channel descriptor passes through an *excitation operation*, which generates per-channel weights through a simple gating mechanism; in our case, a *Sigmoid* function. The weights are applied to the input U to generate the output $\hat{U}$ of the S&E block. The output is a channel-wise weighted U. We further used a convolution layer with a kernel size one to adjust the size of $\hat{U}$ and further feed $\hat{U}$ to the next convolution blocks in both SED and DoA branches. Similar to VASELD, the S&ESELD network trained with tPIT loss and orthogonality constraint has been imposed similar to constrained EINV2. Figure 13 illustrates the S&E block.

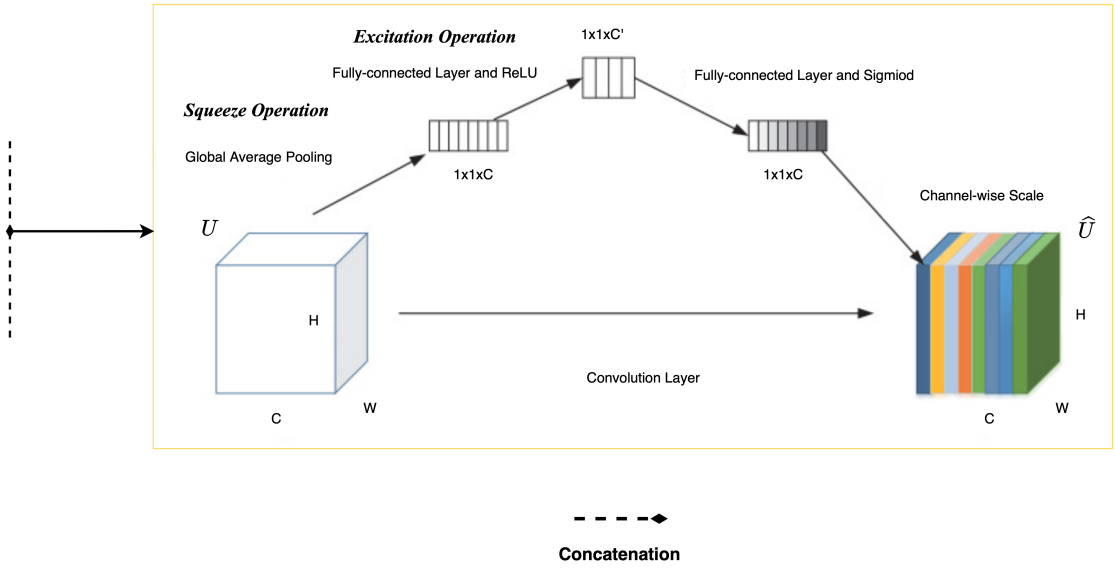


Figure 13: Illustration of the Squeeze-and-Excitation Block.

The Squeeze-and-Excitation Block bounded by the yellow line. The block's input is $U \in \mathbb{R}^{H \times W \times C}$. The *squeeze operation* which is (global average pooling) generates a channel descriptor of size $(1 \times 1 \times C)$ and *excitation operation* generates per-channel weights through a sigmoid function. Finally, the output $\hat{U} \in \mathbb{R}^{H \times W \times C}$ generated by applying channel-wise scaling on U.

5 Datasets

In this chapter, we describe the audio datasets that we considered for this work which are the **TAU-NIGENS Spatial Sound Events 2020**[34] for training the neural networks to tackle the SELD problem; an additional dataset named **ESC-50**[33] is considered for generating urban ambient noises.

5.1 TAU-NIGENS Spatial Sound Events 2020 Dataset

The **TAU-NIGENS Spatial Sound Events 2020** is a labelled dataset that contains sound recordings, consisting of sound events of various categories from various acoustical spaces, source directions and distances as observed from the microphone position. The sounds were recorded at fifteen different indoor locations with different shapes, sizes and sound absorption properties. The main purpose of this dataset is mimicking real-life sounds; therefore, the impulses response (IR) acquired using a stationary Eigenmike spherical microphone array. A Genelec G Three loudspeaker, mounted on a wheeled platform, was used to playback a maximum length sequence (MLS) around the Eigenmike microphone. The dataset contains both stationary and moving sound sources; the moving sources have three angular speeds of 10, 20 or 40 degree/second. The measured impulse response directions and distances differ with the spaces with azimuths range between $\phi \in [-180, 180)$ and the elevations range between $\theta \in [-45, 45]$ degrees. As encoding of the spatial information differs with the spatial sound format, Politis et al. [34] provided the sound recordings in two different 4-channel spatial sound formats, extracted from the

32-channel Eigenmike format. The first format is MIC extracted by selecting a subset of the Eigenmike channels, corresponding to a tetrahedral capsule arrangement. The second format FOA, extracted through a matrix of 4×32 conversion filters. The labels of the sound recordings are provided on the frame-level for both the category of the active sound event and the direction represented by azimuth and elevation.

5.1.1 Tetrahedral Capsule Arrangement

The tetrahedral capsule arrangement (MIC) format has the microphones arranged in spherical coordinates given by azimuth angle ϕ , elevation angle θ and radius r (ϕ, θ, r): (45°, 35°, 4.2 cm), (-45°, -35°, 4.2 cm), (135°, -35°, 4.2 cm) and (-135°, 35°, 4.2 cm), taken from channel 6, 10, 26 and 22 of the Eigenmike. The channels encode the sound direction-of-arrival (DoA) with both time differences and level differences. Time differences result from the spacing and level differences following the acoustic shadowing, the area in which the sound reflects or partially fails to propagate, and the hard spherical baffle in between.

The directional response is a synonym to microphone directionality which is the microphone's sensitivity relative to the direction defined by the azimuth and the elevation from which the sound arrives. There are various directional patterns known as polar patterns. Polar patterns represent 360°sensitivity variation around the microphone such that the microphone is in the centre, and 0°represents the front.

Politis et al. [34] describe the directional responses $H_m(\phi_m, \theta_m, \phi, \theta, \omega)$ of the MIC format given by an analytical expression as follows:

$$H_m(\phi_m, \theta_m, \phi, \theta, \omega) = \frac{1}{(\omega R/c)^2} \sum_{n=0}^{30} \frac{i^{n-1}}{h'^{(2)}(\omega R/c)} (2n+1) P_n(\cos \gamma_m) \tag{27}$$

where m is the channel number, (ϕ_m, θ_m) are the microphone's azimuth and elevation position, $\omega = 2\pi f$ is the angular frequency computed from the frequency f. $R = 0.042m$

is the array radius, $c = 343m/s$ is the speed of the sound, $\cos(\gamma_m)$ is the cosine angle between the microphone position and the DoA. P_n is the Legendre polynomial of degree n, and $h'^{(2)}$ is the derivative with respect to the argument of a spherical Hankel function of the second kind. The expansion is limited to 30 terms which provide a negligible modelling error up to 20 kHz.

5.1.2 First-Order Ambisonics

First-Order Ambisonics format (FOA) also know as Ambisonic B-format is 4-channel spatial sound that covers 360°. The channels are (W, X, Y, Z), where W is an omnidirectional polar pattern, containing all sounds in the sphere, coming from all directions at equal gain and phase, X is a polar pattern pointing forward, Y is a polar pattern pointing to the left and Z is a polar pattern pointing up. Figure 14 illustrates First-Order Ambisonics Format, polar patterns of an omnidirectional microphone W(top), and three directional microphones in each one of the directions X, Y, Z(bottom).

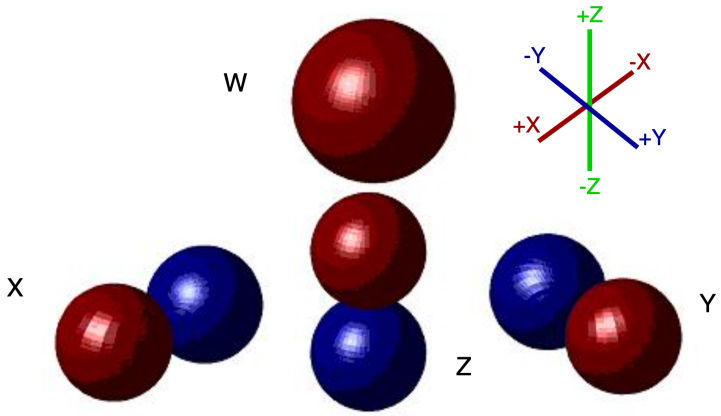


Figure 14: Illustration of First-Order Ambisonics Format.

Polar patterns of an omnidirectional microphone W(top), and three directional microphones in each direction X, Y, Z(bottom)

Unlike the MIC format, the FOA format provides only level differences with no time

differences. According to Politis et al. [34], the FOA format directional responses of the m channel $H_m(\phi, \theta, f)$ to a sound event from DoA defined by azimuth angle ϕ and elevation angle θ,at frequency f is defined as follows:

$$\begin{aligned} H_1(\phi, \theta, f) &= 1 \\ H_2(\phi, \theta, f) &= \sin(\phi) * \cos(\theta) \\ H_3(\phi, \theta, f) &= \sin(\theta) \\ H_4(\phi, \theta, f) &= \cos(\phi) * \cos(\theta) \end{aligned} \tag{28}$$

where $H_1(\phi, \theta, f)$, $H_2(\phi, \theta, f)$, $H_3(\phi, \theta, f)$, $H_3(\phi, \theta, f)$ and $H_4(\phi, \theta, f)$ are the directional responses of the channels W, Y, Z and X, respectively.

5.1.3 Dataset Specification

The dataset contains 14 distinct sound event classes and the DoA for each sound event in both formats MIC and FOA. The sound events categories are given by the class labels; the azimuth and elevation give the DoA labels in degree. The class and DoA labels are provided at the frame level of the recordings. The dataset specifications can be summarized as follows: the training dataset contains 600 one-minute long sound scene recordings; the evaluation dataset contains 200 one-minute long sound scene recordings; the total sound event samples is 700 distributed over the 14 classes; the dataset includes two overlapping sound events at maximum; an ambient noise collected from all the indoor locations in which the sounds recorded and added to the sound events with various signal-to-noise-ratio (SNR).

5.2 Ambient Noise Dataset

The **ESC-50**[33] is a labelled dataset collected to assists the research of environmental sound events classification. The dataset contains 2000 environmental sound recordings,

each 5-second-long for 50 distinct classes distributed over five major categories. The dataset is suitable for benchmarking methods of environmental sound events classification. As our work intended for outdoor environments, we generated ambient noise only from *urban noises* category. The *urban noises* consists of 10 sound classes such as car horn, train, church bell, siren and helicopter.

6 Experiments

6.1 Evaluation Metrics

To assess the performance of our novel architectures, we use two sets of metrics that measure the sound events detection and localization performance in addition to the joint performance $SELD$ score.

1. The first set of metrics considered in the DCASE challenge and workshop 2019 and earlier. The detection performance for the SED task introduced by Mesaros et al. [29] is evaluated by encountering the sound event's presence or absence compared to the labels of the same class. Typical SED metrics include precision (P), recall (R), F-score and error rate (ER); the metrics are computed on a segment of one second with no overlap. The segment-wise results are obtained from the frame level; a sound event is considered active in the whole segment if it appeared in one frame. The F-score and ER are defined as follows:

$$\begin{aligned} P &= \frac{TP}{TP+FP}, \\ R &= \frac{TP}{TP+FN}, \\ F &= \frac{2PR}{P+R} \\ ER &= \frac{D+I+S}{N} \end{aligned} \tag{29}$$

where TP is true positive if the prediction and the ground truth of the same class, FP and I are false positive or insertion if the prediction is active and ground truth is inactive, FN and D are false negative and deletion if the prediction is inactive and ground truth is

active; S is a substitution error if one true positive and one true negative appearing at the same time. N is the total number of ground truth events. An ideal neural network will have the F-score of the SED task equal to one and an error rate equal to zero. The S, I and D, respectively, computed for a one-second segment as follows:

$$\begin{aligned} S &= \min(FN, FP) \\ I &= \max(0, FP - FN) \\ D &= \max(0, FN - FP) \end{aligned} \tag{30}$$

Similarly, the localization error is measured by determining the closest detected class event. The localization error is the angular distance $\theta \in [0, 180]$ between the predicted and ground truth DoA. In case the output in Cartesian coordinate, the DoA prediction is $(x_{pred}, y_{pred}, z_{pred})$ and DoA ground truth is (x_{gt}, y_{gt}, z_{gt}), θ is computed as follows:

$$\theta = 2 \cdot \arcsin\Big(\frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{2}\Big) \cdot \frac{180}{\pi} \tag{31}$$

where $\Delta x^2 = x_{gt} - x_{pred}$, $\Delta y^2 = y_{gt} - y_{pred}$, $\Delta z^2 = z_{gt} - z_{pred}$. We further compute the DoA frame recall (LR) as follows:

$$LR = \frac{TP}{TP + FN} \tag{32}$$

where TP is the total true positive in which the predicted DoAs are equal to the ground truths, FN false negative in which the predicted DoAs is unequal to the ground truth. An ideal neural network will have DoA frame recall equal to one (frame recall reported in percentage) and an error rate equal to zero. Additionally, we report a joint evaluation metric $SELD$ score, which is computed as follows:

$$SELD = \frac{(SED_{score} + DoA_{score})}{2} \tag{33}$$

where

$$\begin{aligned} SED_{score} &= \frac{(ER + (1 - F))}{2}, \\ DoA_{score} &= \frac{\frac{DoA_{error}}{180} + (1 - LR)}{2} \end{aligned} \tag{34}$$

2. The second set of metrics introduced by Mesaros et al. [30] and illustrated in figure 15. The new metrics are considered in the DCASE challenge and workshop 2020 and later. These metrics are *Location-sensitive detection* and *Class-sensitive localization*.

   Location-sensitive detection: for the SED task, a threshold θ is defined. A predicted active sound class is considered TP if its location does not exceed θ, if the threshold exceed the prediction is considered false positive, and the undetected event is a false negative. According to this, the F-score (F) and error rate (ER) from 29 redefined with a subscript θ, in our case $\theta \leq 20°$ and the SED metrics reported as $F_{\leq 20°}$ and $ER_{\leq 20°}$.

   Class-sensitive localization: the localization error is only calculated between sounds with the same label in each frame. The localization frame recall, and localization error reported as $LR_{\leq 20°}$ and $LE_{\leq 20°}$, respectively. The joint evaluation metric $SELD$ score is computed accordingly and reported as $SELD_{\leq 20°}$. Figure 15 illustrates the SED and DoA and their joint metrics of sound events from two different classes, where the ground truths is labelled as rectangles and predictions as circles. Figure 15.a illustrates the detection of an active or inactive sound event class, figure 15.b illustrates computing the localization error by comparing the prediction with the closest ground truth and neglects the detected active sound event class, figure 15.c illustrates the join class-sensitive localization and location-sensitive detection metric.

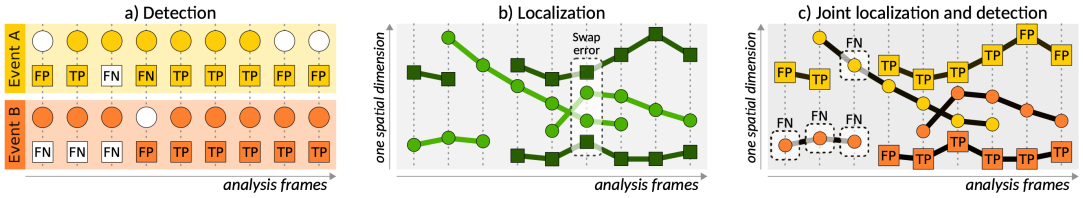


Figure 15: Illustration of the evaluation metrics of detection, localization and joint metric[30].

Illustration of the SED and DoA and their joint metrics of sound events from two different classes, where the ground truths is labelled as rectangles and predictions as circles. a) illustrates the detection of an active or inactive sound event class, b) illustrates computing the localization error by comparing the prediction with the closest ground truth and neglects the detected active sound event class, c) illustrates the join class-sensitive localization and location-sensitive detection metric.

6.2 Baseline Methods

We compare our approaches with the baseline EINV2. EINV2 reported the best performance on dataset FOA format[34] in the DCASE challenge and workshop 2020. EINV2's input is a four-channel spectrogram for the SED branch of the network; a 3-dimensional intensity vector is further included for the DoA branch; thus, the DoA branch of the network has a seven-channel input. Our experiments show that adding a 3-dimensional intensity vector for the SED task enhances the performance, at least for the dataset format FOA. Thus, we report the EINV2 with the input of four and seven channels as EINV2-4 and EINV2-7, respectively. We further created a *constrained EINV2* as an additional baseline, which is reported as EINV-7C. We report the results for both datasets formats FOA and MIC. All the reported results are on the validation fold demonstrated in table 1 for a maximum overlapping sound event of two. In figure 16, we visualize the predictions of VASELD architecture on an audio wave taken from the evaluation dataset mentioned in section 5.1.3.

6.3 Training Details

The input for both architectures (VASELD and S&ESELD) is a spectrogram on Mel scale and a 3-dimensional intensity vector, making the total number of the input channels for SED and DoA tasks equal seven. In table 3 and 4, we demonstrate the results of our experiments for both FOA and MIC formats, respectively. Table 1 demonstrates the training and validations folds that we used in all experiments.

Table 1: Training and Validation Folds for FOA and MIC Formats

Training Folds	Validation Folds
2,3,4,5,6	1

Table 2 demonstrates audio preprocessing and feature extraction configuration for both S&ESELD and VASELD architectures. S&ESELD and VASELD trained end-to-end. We trained S&ESELD with batch size = 32, for 100 epochs and $\mathcal{L}_{orth}$ with a $\alpha = 1e-5$. We considered Adam optimizer with a learning rate = 5e-4, a step scheduler with a step size = 80 and multiplicative factor = 0.3. VASELD is trained with slightly different parameters as follows: batch size = 8, learning rate = 1e-4; we further considered $\mathcal{L}_d$ with a weight coefficient $\beta = 1$.

Table 2: Audio Preprocessing and Feature Extraction Configuration

Parameter	Value
Sampling Rate	24000
Window Type	Hann
Hop Length	600
Number of FFT	1024
Number of Mel bins	256

6.4 Experimental Results

In table 3, we demonstrate the results of our experiments for the dataset format FOA. The results show that including a 3D intensity vector for both SED and DoA tasks enhances the performance as EINV2-7 shows better performance than EINV2-4, concluding that the SED and DoA tasks benefit from each other. EINV2-7C represents a *Constrained Baseline*, the results reveal that forcing orthogonality constraints alleviate the capacity underutilization problem, which results in a performance gain. Moreover, the results show that the visual attention-based parameter sharing architecture (VASELD) outperforms the baseline EINV2-4 on all metrics; however, EINV2-7C achieves a better error rate and

localization error for both sets of metrics. Squeeze-and-Excitation-based parameter sharing architecture (S&ESELD) outperforms the EINV2-4 in all metrics except for localization error LE since it introduces a loss in performance. S&ESELD architecture does not outperform VASELD either EINV2-7C, suggesting that cross-stitch-based parameter sharing supplemented with the orthogonality constraints outperforms the squeeze-and-excitation-based parameter sharing.

We further report the joint evaluation metric $SELD$ score for each architecture computed according to equation 33. The score is reported as $SELD_{\leq 20^\circ}$ for *Class-sensitive localization* and *Location-sensitive detection* metrics, otherwise $SELD$.

Table 3: Performance comparison between the neural network architectures evaluated on the FOA validation fold

An ideal network have $F_{\leq 20^\circ}$, $LR_{\leq 20^\circ}$, F and LR equal to one; $ER_{\leq 20^\circ}$, $LE_{\leq 20^\circ}$, ER and LE equal to zero. An ideal $SELD$ score is zero. The up arrows indicate higher is better and down arrows indicate lower is better. The best result is in bold; the second best is underlined.

Networks	$F_{\leq 20^\circ} \uparrow$	$ER_{\leq 20^\circ} \downarrow$	$LR_{\leq 20^\circ} \uparrow$	$LE_{\leq 20^\circ}(\%) \downarrow$	$SELD_{\leq 20^\circ} \downarrow$	$F \uparrow$	$ER \downarrow$	$LR \uparrow$	$LE(\%) \downarrow$	$SELD \downarrow$
EINV2-4	0.6841	0.4124	0.753	10.422	0.2583	0.7534	0.3522	0.7693	8.906	0.2198
EINV2-7	**0.7040**	0.3942	0.765	9.565	0.2446	0.7686	0.3346	0.7818	<u>8.91</u>	0.2084
EINV2-7C	<u>0.7239</u>	**0.3691**	**0.7814**	**9.395**	**0.2290**	**0.7828**	**0.3171**	0.**7889**	**8.652**	**0.1984**
VASELD	**0.7240**	<u>0.3732</u>	<u>0.7806</u>	<u>10.093</u>	<u>0.2312</u>	<u>0.7823</u>	<u>0.3205</u>	0.7721	9.568	<u>0.2048</u>
S&ESELD	0.7048	0.3878	0.7647	10.200	0.2437	0.7661	0.3324	<u>0.7871</u>	9.371	0.2078

In table 4, we demonstrate the results of our experiments for dataset format MIC. In general, all the evaluated architectures present lower performance on MIC format than on FOA format. The results show that including a 3D intensity vector for both SED and DoA tasks for the baseline EINV2 does not introduce a performance gain. On the contrary, there is a relatively significant decline in performance, especially in $F_{\leq 20^\circ}$, suggesting that the benefit of the 3D intensity vector is highly correlated with the microphone positions used to receive the sound. Nevertheless, forcing an orthogonality constraint compensates for the performance's decline. VASELD performance stays steady, outperforms the

EINV2-4, and achieves comparable results to EINV2-7C except for the localization error in which a performance decline can be observed. Unlike S&ESELD, which performs poorly compared to VASELD and EINV2-7C; however, S&ESELD achieves a better $SELD$ score than EINV2-4. According to that, we presume that visual attention-based parameter sharing architecture maintains higher robustness than cross-stitch-based and squeeze-and-excitation based parameter sharing architectures.

Table 4: Performance comparison between the neural network architectures evaluated on the MIC validation fold

An ideal network have $F_{\leq 20^\circ}$, $LR_{\leq 20^\circ}$, F and LR equal to one; $ER_{\leq 20^\circ}$, $LE_{\leq 20^\circ}$, ER and LE equal to zero. An ideal $SELD$ score is zero. The up arrows indicate higher is better and down arrows indicate lower is better. The best result in bold; the second best is underlined.

Networks	$F_{\leq 20^\circ} \uparrow$	$ER_{\leq 20^\circ} \downarrow$	$LR_{\leq 20^\circ} \uparrow$	$LE_{\leq 20^\circ}(\%) \downarrow$	$SELD_{\leq 20^\circ} \downarrow$	$F \uparrow$	$ER \downarrow$	$LR \uparrow$	$LE(\%) \downarrow$	$SELD \downarrow$
EINV2-4	0.6644	0.4278	0.7519	12.353	0.2583	0.7523	0.352	0.7662	10.755	0.2198
EINV2-7	0.3975	0.6612	0.7542	29.675	0.2446	0.7552	0.3508	0.7648	27.859	0.2084
EINV2-7C	<u>0.679</u>	<u>0.4140</u>	**0.7708**	<u>11.787</u>	**0.2290**	**0.7713**	**0.3307**	**0.7775**	**10.7320**	**0.1984**
VASELD	**0.6778**	**0.4138**	<u>0.7601</u>	**11.503**	<u>0.2312</u>	<u>0.7623</u>	0.3394	0.7662	<u>10.960</u>	<u>0.2048</u>
S&ESELD	0.4178	0.6425	0.7585	27.680	0.2437	0.7596	<u>0.3389</u>	<u>0.7703</u>	23.938	0.2078

Figure 16 illustrates the spectrogram and VASELD network's predictions (right) compared to the ground truth (left). The predictions are generated on an audio recording of 60 seconds long taken from the evaluation split of the FOA dataset.

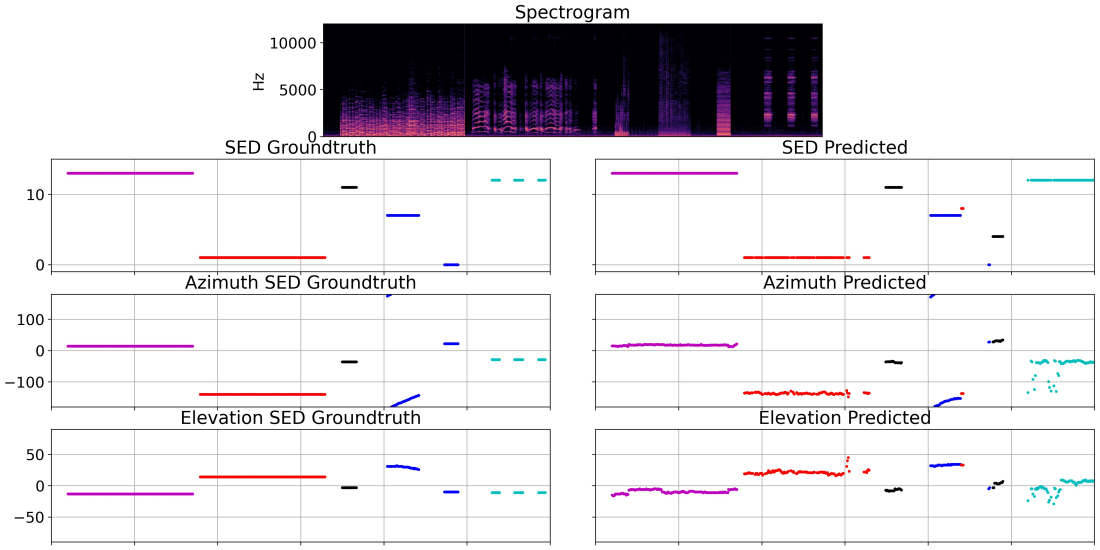


Figure 16: Illustration of the VASELD predictions compared to the ground truths on dataset format FOA.

Illustration of the spectrogram and the SED and DoA ground truths and predictions, the colours indicate different active sound event classes in the audio where the y-axis represents the class indices. The DoA represented in terms of azimuth $\theta \in [-180, 180]$ and elevation $\phi \in [-90, 90]$. The spectrogram's x-axis represents the time measured in seconds, and the y-axis represents the frequency measured in hertz (Hz).

Furthermore, figure 17 illustrates the failure cases of VASELD network. The figure shows that the network failed in classifying and localizing sound events.

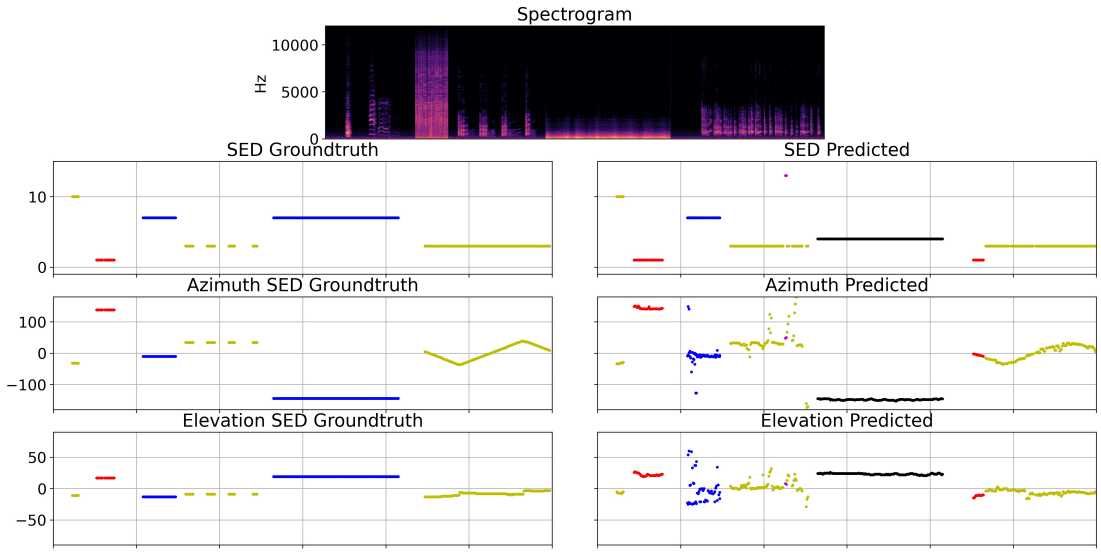


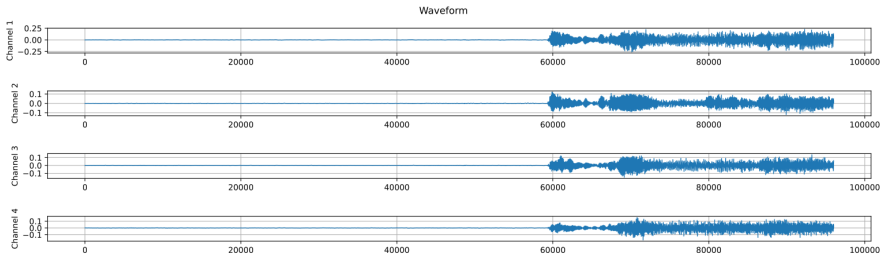
Figure 17: Illustration of the VASELD Fail Case on dataset format FOA.

Illustration of the spectrogram and the SED and DoA ground truths and predictions, the colours indicate different active sound event classes in the audio where the y-axis represents the class indices. The DoA represented in terms of azimuth $\theta \in [-180, 180]$ and elevation $\phi \in [-90, 90]$. The spectrogram's x-axis represents the time measured in seconds, and the y-axis represents the frequency measured in hertz (Hz).

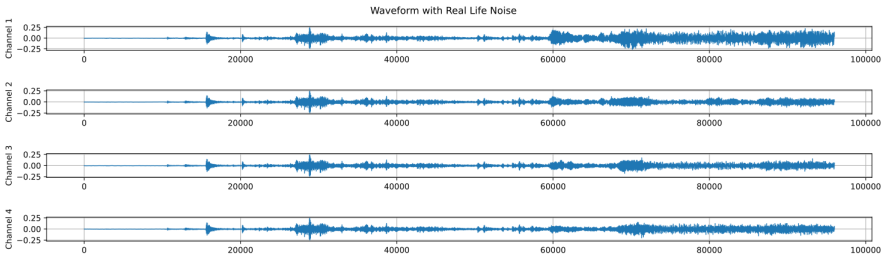
6.5 Robustness Against Noise

It is common to evaluate the networks' robustness against noise by adding noise sampled from a Gaussian distribution[47, 40]. In this work, the evaluated networks are meant to be used for a mobile robot in outdoor environments; hence, we opted for adding noises recorded in real-life exterior environments. We randomly sampled recordings from the *Noise Dataset* mentioned in section 5.2 with various signal-to-noise ratios, ranging from 0 to 40 with a step size 5. We evaluated the networks' robustness in terms of $SELD_{\leq 20^\circ}$ score in two training strategies and we considered only dataset format FOA : 1. we train our networks on noisy training folds and validate on a noisy validation fold and 2. we train our networks on the original training folds and validate on a noisy validation fold.

Figure 18 illustrates an audio segment and its noisy form with a signal-to-noise-ration (SNR) = 5dB. The audio segment is taken from the validation fold and includes two overlapping sound events.



(a) Original Audio Segment



(b) Noisy Audio Segment with SNR = 5dB

Figure 18: An audio segment taken from the FOA validation set and its noisy with 5dB of noise added.

Figure 19(left) shows a comparison between the VASELD and S&ESELD networks' performance measured in terms of $SELD_{\leq 20^\circ}$ score when the networks are trained and validated on noisy audio segments, the figure shows that the $SELD_{\leq 20^\circ}$ decreases as the SNR increases. Furthermore, the figure states clearly that the overall increase in $SELD_{\leq 20^\circ}$ remains relatively low for both networks compared to un-noisy results in table 3; even when the noise ratio is high such as when SNR = 0 and 5dB. Base on that, we conclude that the VASELD model can be used for conducting experiments on a robot in an outdoor environment.

Figure 19(right) shows a comparison between the VASELD and S&ESELD networks' performance measured in terms of $SELD_{\leq 20^\circ}$ score when the networks are trained on the original un-noisy data, whereas validated on noisy audio segments. The figure shows clearly that the $SELD_{\leq 20^\circ}$ decreases as the SNR increases reaching its lowest value when SNR = 40dB. In general, S&ESELD shows less robustness against noise compared to VASELD.

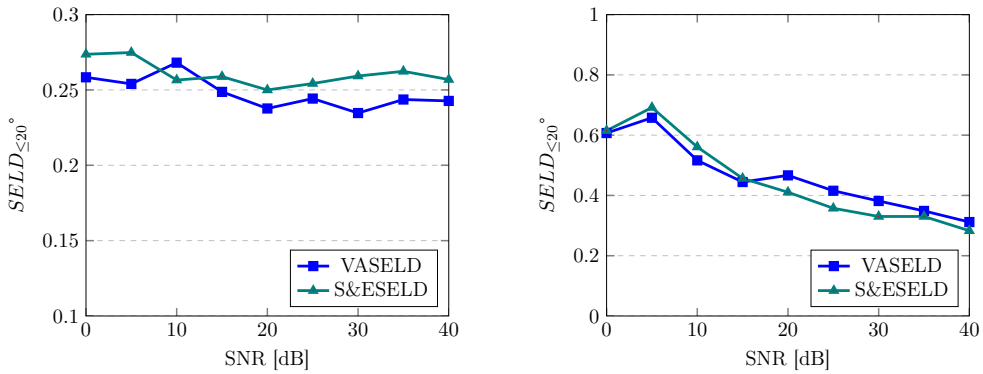


Figure 19: Robustness of (VASELD and S&ESELD) neural networks against noise measured in the joint $SELD_{\leq 20^\circ}$ metric following two training strategies.

The plot on the left shows a performance comparison in terms of $SELD_{\leq 20^\circ}$ between VASELD and S&ESELD when the networks are trained and validated on noisy folds. While the plot on the right shows a performance comparison in terms $SELD_{\leq 20^\circ}$ when the networks are trained on the original training folds and validated on a noisy validation fold.

7 Conclusion

This thesis proposes multi-tasking neural network architectures that can perform comparably to the current best architecture for tackling the sound event localization and detection problem (SELD). The neural network architecture EINV2[10] that achieved the best results in the DCASE workshop and challenge202 compromises two identical branches for SED and DoA task and relies on a Cross-Stitch parameter sharing mechanism to maintain the information sharing. The multi-tasking nature of the SELD problem and the EINV2's design lead to feature redundancy, resulting in the network's capacity underutilization. To this extend, we created a new baseline *constrained EINV2* that outperforms EINV2, the new baseline is the result of imposing orthogonality constraints on the weights of EINV2; imposing orthogonality constraints resulted in a performance gain as demonstrated in section 6.4.

Our proposed multi-tasking neural network architectures variants that tackle the SELD problem for polyphonic sound utilize different parameter sharing mechanisms. In the first architecture named **VASELD**, we utilized a visual attention mechanism for parameter sharing between the SED and DoA tasks. The architecture comprises a shared feature space in which global features are learned from the sound's spectrogram and two private spaces in which task-specific features are learned. The private spaces rely on stacked visual attention modules to learn task-specific features; moreover, we maintained the Transformer layers and the track-wise output format considered by EINV2. Furthermore, we utilize an additional loss that keeps the private spaces decorrelated by encouraging orthogonal private spaces during the training. Our architecture outperforms the EINV2 and achieves

comparable results to constrained EINV2 on the SED and DoA task metrics and the joint evaluation metric $SELD$ score. We further studied the network's robustness by adding ambient noise at different signal-to-noise ratios generated in outdoor environments. We evaluated the network using two different training strategies. The performance of **VASELD** measured in terms of the joint metric $SELD_{\leq 20^\circ}$ stays steady even when the SNR is low; suggesting the possibility of deploying the neural network on a real robot to conduct experiments in an outdoor environment.

A successful multi-tasking neural network is strongly affected by the parameter sharing mechanism. Thus, we considered Squeeze-and-Excitation blocks[21] as a parameter sharing mechanism replacing the Cross-Stitch units that are considered in the baseline EINV2. Our architecture named **S&ESELD** achieves comparable results to the baseline EINV2. However, the network shows higher localization error than constrained EINV2 and less robustness against noise at both training strategies compared to the **VASELD** architecture. We evaluated the **VASELD** and **S&ESELD** architectures on the challenging dataset **TAU-NIGENS Spatial Sound Events 2020**[34] dataset, the dataset contains sound recordings, consisting of overlapping sound events of different categories and recorded in various acoustical spaces, source directions and distances. The sound recordings provided in two different spatial sound formats MIC and FOA.

Although the **VASELD** architecture based on visual attention parameter sharing showed comparable performance to the baseline, there is still room for improvement as possible future works. Deep-based models are data-hungry; therefore, increasing the training samples and considering audio data augmentation techniques could further improve the network performance.

8 Acknowledgments

Firstly, I would like to thank Prof. Dr. Wolfram Burgard for giving me a valuable opportunity to work on my master's thesis in his well-respected lab. Secondly, I would like to thank my advisor Jannik Zürn, for his support, patience, enthusiasm and positive attitude throughout the thesis. This work would not be possible without him.

Further, I want to thank Juliane for her friendship; without our long weekends' calls and scientific discussions my time would be unbearable. I sincerely would like to thank Lorraine for her support and genuineness. Further, I sincerely thank Abdelrahman Younes for being a good listener and for the very long conversations and walks. Lastly, I must express my deep gratitude to my family for their love and for supporting my decisions.

Bibliography

[1] *Cross-Stitch Networks for Multi-task Learning*, 2016.

[2] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. Technical Report 1, 2019.

[3] Sharath Adavanne, Joonas Nikunen, Archontis Politis, and Tuomas Virtanen. TUT Sound Events 2018 - Ambisonic, Reverberant and Real-life Impulse Response Dataset, April 2018. URL `https://doi.org/10.5281/zenodo.1237793`.

[4] Sharath Adavanne, A. Politis, and T. Virtanen. Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network. *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1462–1466, 2018.

[5] Sharath Adavanne, A. Politis, and T. Virtanen. Localization, detection and tracking of multiple moving sound sources with a convolutional recurrent neural network. *ArXiv*, abs/1904.12769, 2019.

[6] N. Bansal, X. Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep cnns?, 2018.

[7] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2015. doi: 10.1109/IJCNN.2015.7280624.

[8] Yin Cao, Qiuqiang Kong, T. Iqbal, Fengyan An, W. Wang, and Mark D. Plumbley. Polyphonic sound event detection and localization using a two-stage strategy. *ArXiv*, abs/1905.00268, 2019.

[9] Yin Cao, T. Iqbal, Qiuqiang Kong, Yue Zhong, Wenwu Wang, and Mark D. Plumbley. Event-independent network for polyphonic sound event localization and detection. *ArXiv*, abs/2010.13092, 2020.

[10] Yin Cao, Turab Iqbal, Qiuqiang Kong, An Fengyan, Wenwu Wang, and Mark D Plumbley. An improved event-independent network for polyphonic sound event localization and detection. *arXiv preprint arXiv:2010.13092*, 2020.

[11] Yin Cao, Turab Iqbal, Qiuqiang Kong, Yue Zhong, Wenwu Wang, and Mark D. Plumbley. Event-independent network for polyphonic sound event localization and detection, 2020.

[12] Brian Cheung, Alex Terekhov, Yubei Chen, Pulkit Agrawal, and Bruno Olshausen. Superposition of many models into one, 2019.

[13] M. Crawshaw. Multi-task learning with deep neural networks: A survey. *ArXiv*, abs/2009.09796, 2020.

[14] Karim Guirguis, C. Schorn, A. Guntoro, Sherif Abdulatif, and Bin Yang. Seld-tcn: Sound event localization & detection via temporal convolutional networks. pages 16–20, 2021.

[15] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2016.

[16] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.

[18] W. He, Petr Motlícek, and J. Odobez. Deep neural networks for multiple speaker detection and localization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 74–79, 2018.

[19] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen. Audio context recognition using audio event histograms. In *2010 18th European Signal Processing Conference*, pages 1272–1276, 2010.

[20] T. Hirvonen. Classification of spatial audio location and content using convolutional neural networks. *Journal of The Audio Engineering Society*, 2015.

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[22] Saumya Jetley, Nicholas A. Lord, Namhoon Lee, and Philip H. S. Torr. Learn to pay attention, 2018.

[23] J. Ker, L. Wang, J. Rao, and T. Lim. *IEEE Access*.

[24] Peerapol Khunarsal, Chidchanok Lursinsap, and Thanapant Raicharoen. Very short time environmental sound classification based on spectrogram pattern matching. *Information Sciences*, 243:57–74, 2013. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2013.04.014. URL `https://www.sciencedirect.com/science/article/pii/S0020025513003113`.

[25] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[26] Celio F. Lipinski, Vinicius G. Maltarollo, Patricia R. Oliveira, Alberico B. F. da Silva, and Kathia Maria Honorio. Advances and perspectives in applying deep learning for drug design and discovery. *Frontiers in Robotics and AI*, 6:108, 2019. ISSN 2296-9144. doi: 10.3389/frobt.2019.00108. URL `https://www.frontiersin.org/article/10.3389/frobt.2019.00108`.

[27] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019. doi: 10.1109/CVPR.2019.00197.

[28] Annamaria Mesaros, Toni Heittola, Antti Eronen, and Tuomas Virtanen. Acoustic event detection in real life recordings. In *in Proc EUSIPCO*, 2010.

[29] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016. ISSN 2076-3417. doi: 10.3390/app6060162. URL `http://www.mdpi.com/2076-3417/6/6/162`.

[30] Annamaria Mesaros, Sharath Adavanne, Archontis Politis, Toni Heittola, and Tuomas Virtanen. Joint measurement of localization and detection of sound events. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 333–337, 2019. doi: 10.1109/WASPAA.2019.8937220.

[31] A. Oord, S. Dieleman, H. Zen, K. Simonyan, Oriol Vinyals, A. Graves, Nal Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, 2016.

[32] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Recurrent neural networks for polyphonic sound event detection in real life recordings. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar 2016. doi: 10.1109/icassp.2016.7472917. URL `http://dx.doi.org/10.1109/ICASSP.2016.7472917`.

[33] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL `http://dl.acm.org/citation.cfm?doid=2733373.2806390`.

[34] Archontis Politis, Sharath Adavanne, and Tuomas Virtanen. A dataset of reverberant spatial sound scenes with moving sources for sound event localization and detection.

In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE2020)*, November 2020. URL https://arxiv.org/abs/2006.01919.

[35] Limeng Pu, Rajiv Gandhi Govindaraj, Jeffrey Mitchell Lemoine, Hsiao-Chun Wu, and Michal Brylinski. Deepdrug3d: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS computational biology*, 15(2): e1006718–e1006718, Feb 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006718. URL https://pubmed.ncbi.nlm.nih.gov/30716081. 30716081[pmid].

[36] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986. doi: 10.1109/TAP.1986.1143830.

[37] Kazuki Shimada, Naoya Takahashi, S. Takahashi, and Yuki Mitsufuji. Sound event localization and detection using activity-coupled cartesian doa vector and rd3net. *ArXiv*, abs/2006.12014, 2020.

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

[39] R. Takeda and K. Komatani. Sound source localization based on deep neural networks with directional activate function exploiting phase information. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 405–409, 2016. doi: 10.1109/ICASSP.2016.7471706.

[40] Abhinav Valada, Luciano Spinello, and Wolfram Burgard. Deep feature learning for acoustics-based terrain classification, 2015.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[42] J. Wang, Y. Chen, R. Chakraborty, and Stella X. Yu. Orthogonal convolutional

neural networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11502–11512, 2020.

[43] Di Xie, Jiang Xiong, and Shiliang Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation, 2017.

[44] Nelson Yalta, Kazuhiro Nakadai, and Tetsuya Ogata. Sound source localization using deep learning models. *Journal of Robotics and Mechatronics*, 29(1):37–48, February 2017. ISSN 0915-3942. doi: 10.20965/jrm.2017.p0037.

[45] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and J. Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245, 2017.

[46] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection, 2015.

[47] Jannik Zürn, Wolfram Burgard, and Abhinav Valada. Self-supervised visual terrain¨ classification from unsupervised acoustic feature learning. *IEEE Transactions on Robotics*, 2020.