

1D source localisation for hearing aids with distributed, low data-rate models

Jonas Van Damme

Student number: 01900486

Supervisor: Prof. dr. ir. Nilesh Madhu

Counsellors: Stijn Kindt, Siyuan Song, Jasper Maes

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Electrical Engineering Technology - Automation

Academic year 2023-2024

Admission to loan

The author gives permission to make this master's dissertation available for consultation and to copy parts of this master dissertation for personal use.

In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this master dissertation.

Jonas Van Damme, June 13, 2024

Acknowledgements

With my growing interest during my master year I am grateful to make my thesis about deep learning in sound localisation. Hereby I want to thank everyone contributing to my work. Starting with my promotor prof. dr. ir. Nilesch Madhu who provided me this thesis and shared and showed interests in this work. Next, I want to thank my supervisors, ir. Stijn Kindt, ing. Jasper Maes and ir. Siyuan Song. They offered me guidance and shared their experiences to speed up the process while sharing thoughts of new ideas. I am extra grateful to Stijn who was actively thinking through the process of my work. Further I want to thank everyone else in the team of ASPIRE for making me feel part of the group during my thesis. I enjoyed the activities and transparent environment where you can be yourself. Different cultures or languages were seen as added value.

The Imec team that provided excellent maintenance for the servers I was using. Even unpredictable shut downs were always resolved quickly without further long term problems. Next, I would like to thank the personnel of the AA-tower to always greet me with a laugh when arriving and leaving next to making the office a nice place to work in.

I want to thank my friends and family, especially my mom, brother and sister to always support me when I was having hard times. My friends helped me to improve my work life balance to not feel overworked. At last I want to thank my dad who was a wonderful father and left beautiful memories behind. These memories are one of the strongest driving forces I can have.

Jonas Van Damme, June 13, 2024

Explanation in connection with the master's dissertation

This master's dissertation is part of an exam. Any comments formulated by the assessment committee during the oral presentation of the master's dissertation are not included in this text.

Abstract

Hearing aids have undergone significant evolution, transitioning into devices capable of significant compute on-ear. Next to these computations, high demand processing in the hearing-aid system is done in a centralised device, where more computational power is available. This however requires (wireless) data transmission between these devices, which can adversely impact battery life. Different strategies for bandwidth efficient audio source localisation for binaural hearing devices are compared. A common strategy is to use audio codecs, like the LC3+ codec to minimise the bandwidth. Here, a bandwidth-compressed version of a state-of-the-art cooperative system is also investigated. The co-operative model first processes the microphone signals locally, before transmitting features to the central processor for further analysis. Lowering bandwidth of the system implies a suitable compression of these features. The investigation encompasses various optimisations, including the quantisation of the embedded features, adjustments to the feature dimensions, as well as modifications to time and frequency resolutions. In general the cumulative adjustments led towards an improved performance which was slightly below, but well comparable to, the centralised (baseline) model. Additionally, proof-of-concept experiments on knowledge distillation and potential expansions of the cooperative model have been successfully tested.

1D source localisation for hearing aids with distributed, low data-rate models

Jonas Van Damme

Supervisor: Prof. dr. ir. Nilesh Madhu

Counsellors: Stijn Kindt, Siyuan Song, Jasper Maes

Abstract—Hearing aids have undergone significant evolution, transitioning into devices capable of significant compute on-ear. Next to these computations, high demand processing in the hearing-aid system is done in a centralised device, where more computational power is available. This however requires (wireless) data transmission between these devices, which can adversely impact battery life. Different strategies for bandwidth efficient audio source localisation for binaural hearing devices are compared. A common strategy is to use audio codecs, like the LC3+ codec to minimise the bandwidth. Here, a bandwidth-compressed version of a state-of-the-art cooperative system is also investigated. The co-operative model first processes the microphone signals locally, before transmitting features to the central processor for further analysis. Lowering bandwidth of the system implies a suitable compression of these features. The investigation encompasses various optimisations, including the quantisation of the embedded features, adjustments to the feature dimensions, as well as modifications to time and frequency resolutions. In general the cumulative adjustments led towards an improved performance which was slightly below, but well comparable to, the centralised (baseline) model. Additionally, proof-of-concept experiments on knowledge distillation and potential expansions of the cooperative model have been successfully tested.

Index Terms—hearing-aid, DNN-based source localisation, bandwidth optimisation, speech codec, knowledge distillation

I. INTRODUCTION

Hearing is one of the major senses, essential for communication, creativity and even survival. As human lifespans continue to increase, age-related hearing loss will likely lead to a corresponding rise in the demand for hearing augmentation – beyond what current hearing aids can offer. The binaural nature of human hearing helps with the perception of auditory cues such as directionality and distance estimation, which is important for source localisation and separation. To replicate this natural auditory experience using binaural hearing aids, source localisation is, thus, an important first step. Typically, the direction of arrival (DOA) is estimated in the azimuth plane and used to inform subsequent sound separation stages [1] and for binaural cue preservation/recreation. Existing, free field methods for estimating DOA rely on the utilisation of multiple microphones and/or arrays [2]–[4] to spatially sample the sound field and, thereby, derive the DOA of the various sources. State-of-the-art (SotA) solutions for DOA estimation exploit deep neural networks (DNN). Typically the models make use of convolutional neural networks (CNN) trained on simulated audio data [5]. Temporal context was added in the

network for better tracking of moving sources [6], [7]. Temporal convolutional network (TCN) and long short-term memory (LSTM) made temporal context in the CNN [5] possible. As the use of free-field arrays in on-ear hearing aids is phased out, it is essential to include the recorded hearing-device and head-related transfer functions (HD-HRTFs) [8] of the specific microphones in the training methodology. Processing within the configuration of behind-the-ear (BTE) devices can be executed either within the on-ear devices themselves or at a centralised processing unit. The communication between these devices can be facilitated wirelessly using protocols such as Bluetooth low-energy (BLE). This brings up another constraint on top of accuracy or model size: the limited bandwidth of the communication platforms. The lossy BLE LC3+ codec is used in the baseline model from previous work [9], [10], where the audio data from each BTE unit is transmitted to the central processing point, where DOA estimation is performed. The results demonstrate that when the codec is included in the training loop, the localisation accuracy is *on par with* (or even slightly *better*) than the DNN trained and tested with the full-bandwidth (uncompressed) audio. Here, an alternative is explored where, instead of sending audio signals directly to the central processor, each hearing device first processes the captured audio independently. The devices then send embedded features to the central processor for the final DOA estimate [11]. To optimise the bandwidth in this model, several techniques are investigated such as: quantising the feature embeddings, reducing the number of embeddings sent per second, different input normalisation and lowering the embedding dimension. In addition, knowledge distillation and model expansion have been explored. The paper is organised as follows: Section 2 first details the two localisation paradigms and the bandwidth optimisation considerations. Section 3 discusses the optimisations in more detail. In Section 4, the knowledge distillation techniques that were tested are explained. Section 5 provides an overview of the methods used for model expansion. The experimental evaluation and results are discussed in Section 6, followed by the conclusions in Section 7.

II. MODELS

A. Signal model

The m th microphone signal at ear e is denoted as $y_{e,m}(n)$. This is a combination of the contributions of J sources

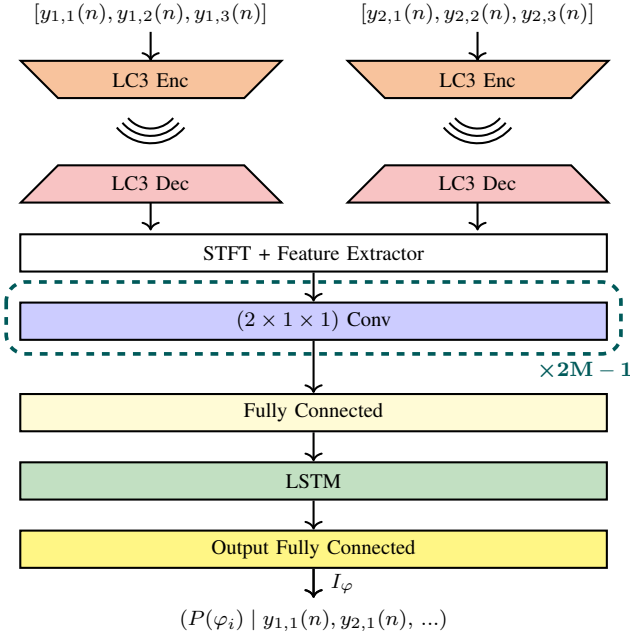


Fig. 1: The centralised localisation algorithm. Initially, the microphone signals are acquired from the LC3+ codec, followed by the estimation of the direction of arrival (DOA) for the incoming sound sources.

$x_{e,m,j}(n)$ and additive noise $v_{e,m}(n)$:

$$y_{e,m}(n) = \sum_{j=0}^{J-1} x_{e,m,j}(n) + v_{e,m}(n) \quad (1)$$

where n is the sample index for a signal sampled at f_s Hz.

All networks operate on the short-time Fourier transform (STFT) representation of the microphone signals, which will be represented as $Y(k, l)$, with k the frequency bin index and l the time frame index. This is computed with a square root Von Hann window, frame size K and hop length L . This representation is often chosen for speaker localisation and separation because of the sparsity and disjointness properties of speech signals [12] in the STFT domain.

B. Centralised localisation model

Fig.1 depicts the model architecture of the centralised model from [9]. For this model, the microphone signals are first sent to the central processing unit. This could either be done by transmitting the full bandwidth signals or by using an audio codec. As was shown in [10], the choice of codec is important. Using the lossy LC3+ encoded signal during training resulted in the same (or sometimes even slightly higher) localisation accuracy than using the full bandwidth signals. This is why the LC3+ codec, a standard codec for Bluetooth low energy, is chosen for the study here. Input features, extracted from the full bandwidth or lossy encoded data, consist of the sine and cosine of the short-time Fourier transform and the normalised STFT amplitude. The normalisation of the

amplitude is performed across all microphones (channels) and will subsequently be referred to as N_c :

$$F_{e,m}(k, l) = \begin{bmatrix} \sin(\angle Y_{e,m}(k, l)) \\ \cos(\angle Y_{e,m}(k, l)) \\ |\overline{Y_{e,m}}(k, l)| \end{bmatrix} \quad (2)$$

with $\angle Y_{e,m}$ the phase of $Y_{e,m}$ and $\overline{Y_{e,m}} = |Y_{e,m}| / \left(\frac{1}{M} \sum_{\nu=1}^M |Y_{e,\nu}| \right)$. Here, $M_e = 3$ is the number of microphones in each BTE device, and $M = 2M_e$ is the total number of microphones in the binaural setup.

At the central unit, the features are stacked along a microphone-channel dimension in such a way that the microphones at different ears are interleaved: $\mathbf{F}(k, l) = [F_{1,1}(k, l), F_{2,1}(k, l), F_{1,2}(k, l), \dots]$. Preliminary evaluations showed this was optimal configuration for the subsequent convolution blocks as this leads to successive convolutions with microphones on opposite ears. The $M - 1$ convolutional blocks (**Conv**) each have a kernel size of 2 along the microphone channel dimension, and iteratively combine the microphones of the binaural array. Note that at this stage each time-frequency point is still processed independently.

The fully connected (**FC**) layer combines the narrowband frequency content independently for each time frame. The **LSTM** layer then incorporates temporal context in a causal manner. A final **Output FC** is used to go from the latent features towards an classifier, where each class represents a DOA sector. The DOA resolution is decided by the number of DOA classes I_ϕ – more classes make for a finer estimate, but will require more training. For the multi-source case ($J > 1$), the choice of a classification-based localisation was shown to outperform regression based outputs [13].

C. Co-operative system

The centralised localisation model works well but relies on audio codecs to reduce the bandwidth. The LC3+ codec (a waveform codec) could lower the data throughput to 16kbps per microphone channel and, while lossy, still did not impact localisation performance when it was included in the training [9]. Going to even lower bitrates often requires neural network based generative models. But these are optimised for single-channel data transmission and do not retain spatial information effectively [10]. Another improvement could be provided by extracting latent features at each ear independently and sending these to the central processor for a final DOA estimate. Then the challenge of low bandwidth is translated into compacting the latent feature vector in an appropriate manner.

Such a feature sharing model is depicted in fig. 2, and is based on the broadband-mixing, co-operative localisation architecture (BM-CLA) of [11]. In the original paper, the goal was to exchange information between two distributed microphone arrays and improve the localisation accuracy by sharing latent features. However, in this context, the architecture will be repurposed to transmit low-bitrate latent features to the central processor. This architecture shares the same underlying

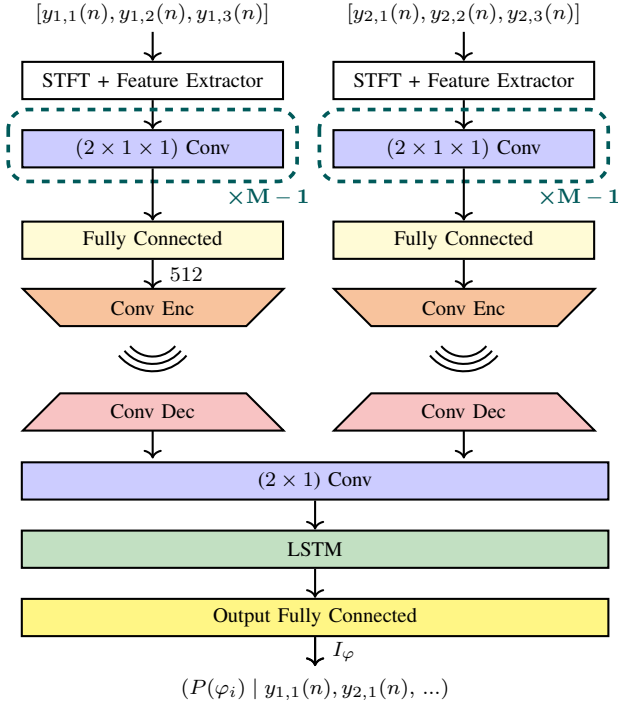


Fig. 2: The broadband mixing co-operative localisation architecture (BM-CLA). Here, the mixing of information occurs after the FC layer, and is also performed by an inter-array convolution.

architecture as the centralised model, which also makes them suitable for a fair comparison.

Instead of sending the data directly to the central processor, the same features as in eq. (2) are independently computed in each earpiece individually. These are again combined through $M_e - 1$ a convolutional layers (**Conv**), following which the narrowband features are combined to yield broadband information using the fully connected layer (**FC**). This yields the latent features intended for transmission to the central processor, as it was already shown in [11] that sharing broadband information was more beneficial than narrowband information.

D. Bandwidth reduction in the co-operative model

A convolutional encoder (**Conv Enc**) and decoder (**Conv Dec**) along the time axis, are added in order to reduce the bandwidth requirements to send these latent features. At the central processor, after decoding, a final convolutional layer combines the information from the two earpieces together. An LSTM and final, fully connected output layer is added, to predict source activity in each DOA sector.

The convolutional encoder can reduce the bandwidth with three main methods. Firstly, it can reduce the feature dimension $\mathcal{F}$, by having a smaller output dimension than the input dimension, effectively compressing the information. Secondly, increasing the temporal-stride S increases the time compression of the model, reducing the frequency of feature transmission to the central processor. Lastly, the features can

also be quantised to further reduce the number of bits to represent the features. The convolutional decoder is actually a deconvolutional layer, that reverses the operation done in the encoder. The feature dimension gets expanded again, temporal interpolation is performed and, where feature-quantisation is performed, the bit depth is increased to the normal 32 bit float numbers.

E. Bandwidth considerations

The centralised system with uncompressed audio requires a bandwidth (in bit per second (bps)) of:

$$BW_{\text{uncompressed}} = 2 \times M_e \times f_s \times b, \quad (3)$$

where the bit depth (b) is typically 32 bits per sample.

For the codec-compressed data transmission, the bandwidth can be calculated as:

$$BW_{\text{coded}} = 2 \times M_e \times \text{bit rate}, \quad (4)$$

where the lowest bit rate of the LC3+ codec is 16 kbits per second.

For the co-operative networks, the bandwidth is calculated right after the convolution encoder layer. The formula is:

$$BW_{\text{cooperative}}(\mathcal{F}, S, Q) = 2 \times \mathcal{F} \times \frac{f_s}{L \times S} \times \frac{b}{Q}, \quad (5)$$

which makes a clear split in the three components where the bandwidth can be reduced. Decreasing the number of features $\mathcal{F}$ per time frame, increasing either the hop length L or the stride S , or decreasing the bits per feature with the quantisation factor Q . Note that this BW is *independent* of the number of microphones per ear, since this gets compressed with the convolutional layers.

Note that for the centralised systems, the availability of the audio signals at the central processor might also be useful for other application like speech enhancement, scene classification, *etc.*. Nevertheless, this is very dependant on the application and while here is looked at the localisation accuracy, these other applications should also be considered in holistic implementations.

III. TESTED BANDWIDTH REDUCTION CONFIGURATIONS

In this section, the tested methods for improving the data transfer in the BM-CLA model are described.

A. Experimental setup

The image-source model [14] is used to generate the room impulse responses (RIRs) for simulating the reverberation in the room. For the binaural hearing-aid, the head and behind-the-ear hearing devices are simulated with the hearing-device and head-related transfer function (HRTF). The output of the networks $P(\varphi_i) | F(m), F(m-1), \dots$ is a multi-hot vector over the DOA range for the chosen resolution, consisting of 0 (inactive sectors) and 1 (active sectors). A fixed resolution of 5° is created with $I = 72$ classes in the 0° to 360° azimuth angle range. The baseline network has 32.25×10^6 multiply-accumulate operations (MACs) and 10.59×10^6 parameters.

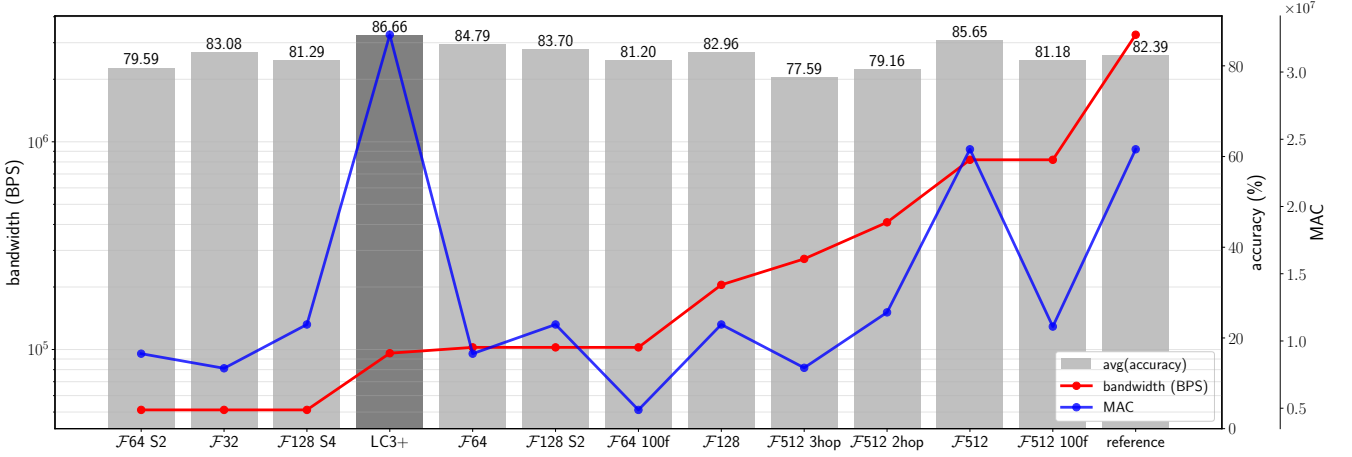


Fig. 3: The bar plot shows average DOA **accuracies** (%) for models equipped with PTSQ and channel, time, frequency amplitude normalisation. $\mathcal{F}$ denotes the number of features, S stands for encoder-decoder with given stride, and f stands for the number of input frequencies. Parameters not explicitly mentioned are consistent with those of the reference model. The reference model is the BM-CLA model with parameters as in [11], with normalisation as the centralised model and no quantisation. The LC3+ is the baseline equipped with the LC3+ codec. Y-axis for bandwidth is logarithmic.

B. Quantisation

Post-training static quantisation (PTSQ) is a method used to exchange value accuracy for value compression in neural network models. This technique involves converting the floating-point data type, typically represented with 32 bits per feature, to an unsigned integer format with 8 bits. Utilising an unsigned integer is viable owing to the prior rectified linear unit (ReLU) activation function, which ensures non-negativity of the values. Subsequently, once the values are compressed to an integer format, they undergo decompression back to float32, thereby affording greater flexibility during the latter stages of network processing.

C. Number of embedded features

Modifying the number of embedded features serves to constrict the bottleneck within network models. This adjustment facilitates a refined selection of broadband embedded features, thereby ensuring the preservation of accuracy.

D. Time resolution

The latent localisation features are computed per STFT frame and transmitted to the central processing module. The frequency of feature transmission can be calculated using the formula:

$$f_f = \frac{f_s}{L \times S} \quad (6)$$

When time resolution is reduced by increasing the parameter L , only specific time sequences are considered. Alternatively, temporal compression can be achieved by employing a convolutional encoder followed by a convolutional decoder within the network bottleneck, as illustrated in fig. 2. Unlike simple sequence skipping, this method allows for the learning of compression patterns while retaining all relevant information.

E. Frequency resolution

Numerous studies have demonstrated that low frequencies primarily influence interaural time differences (ITD), while high frequencies predominantly affect interaural level differences (ILD) [15] [16]. In this context, the initial 100 frequencies are extracted from the original set of 257 frequencies. Consequently, the upper frequency limit extends only to 3125 Hz, instead of 8000 Hz. While the exclusion of the highest frequencies simplifies the problem in terms of calculations, it also entails a loss of detailed information. However, such simplification may yield favorable outcomes for the methodologies proposed in preceding sections.

F. Amplitude normalisation

The amplitude normalisation from section II-B as for the centralised model would be ineffective for the co-operative model due to the extra bandwidth of $2 \times K \times \frac{f_s}{L} \times b$. To solve this, normalisation over channel, frequency and time is done. This results in only $2 \times b$ bits per second.

IV. KNOWLEDGE DISTILLATION

For enabling smaller models in operation, knowledge distillation (KD) comes in mind. A better and often bigger teacher helps the student, often smaller model, during training for better generalisation. Three different training schedules with offline, online and alternate training were tested. Additionally different teacher-student combinations were trained to test teacher dependency. The loss function can be written as follows:

$$Loss = (1 - \alpha)BCE(\hat{y}^S, y) + \alpha(KL(\hat{y}^S, \hat{y}^T)) \quad (7)$$

The parameter α is a weight for the loss from the difference between student ($\hat{y}^S$) and teacher ($\hat{y}^T$) logits which undergo

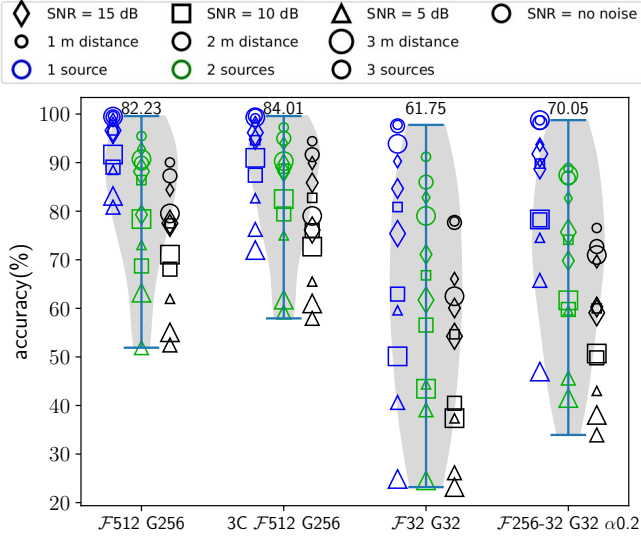


Fig. 4: The violin plot depicts the test accuracy for on the left, two models trained on the dataset as in [9] with 512 features $\mathcal{F}$ and GRU (G) 256. The 3C denotes the one extra convolution. With on the right, the models are trained on less data. The most right one is trained with a teacher with 256 features.

logsoftmax and softmax. This loss is calculated with the Kullback-Leibler (KL) loss function, a function often used in KD [17]. This loss will be added to the original loss without teacher, calculated with the binary cross entropy (BCE) loss function between student logits and true labels.

A. Attention network

To make the weight α dynamic, an attempt to learn α was elaborated with a single fully connected layer with the logits of teacher and student as input. Here the attention network was learned on the loss function of the student to predict the KD weight.

B. Ensemble learning

Multiple teacher models could be deployed together [18], this with the idea in mind for better generalisation. For implementing the loss of these multiple teachers, the choices between average loss or best loss were tested.

V. MODEL EXPANSION

As knowledge distillation relies on having a better teacher, the BM-CLA models were modified to achieve the maximum accuracy from the given data. The training data was not increased to focus on model architecture only.

A. Circular padding

The limit of convolutions is set by the number of mics per ear. This gives $M_e - 1$ convolutions on every ear. By using circular padding, this number can be incrementally increased by one after each convolution. Consequently, additional information could now be gathered from the intra-ear microphones.

B. Skip connection

Inter-ear information is only exchanged after the formation of broadband features, which may result in the loss of critical information. To minimise this potential loss, amplitude information from one microphone on each ear is shared. This information is processed in the same manner as the other inputs before being shared in the fully connected layer.

TABLE I: This table contains average accuracies in %. The first half shows student accuracies ($\mathcal{F}$ 32 L32) with different teachers, trained for 60 and 120 epochs. The second part compares the skip variation with the reference model. The last part compares fully trained models with new normalisation (Ncft) versus old normalisation (Nc).

model	accuracy (%)	epochs
(KD) base- $\mathcal{F}$ 32 L32	70.80	60
(KD) base $\mathcal{F}$ 256-32 L32 α 0.2 best	68.78	60
(KD) base $\mathcal{F}$ 256-32 L32 α 0.2 avg	67.96	60
(KD) $\mathcal{F}$ 256-32 L32 α 0.2	72.15	60
(KD) $\mathcal{F}$ 256-32 L32 α 0.2 online	65.47	60
(KD) $\mathcal{F}$ 256-32 L32 Att alternate	68.92	60
(KD) $\mathcal{F}$ 256-32 L32 Att	69.28	60
$\mathcal{F}$ 32 L32	65.45	60
(KD) $\mathcal{F}$ 256-32 L32 Att	73.99	120
$\mathcal{F}$ 32 L32	73.41	120
skip	83.75	120
reference	82.39	120
baseline Ncft	87.64	60
baseline	85.65	60

VI. RESULTS & DISCUSSION

As depicted in fig. 3, lowering the features, seemed to be most beneficial for lowering bandwidth and remaining accuracy. Additionally, the MAC number reduced as well. For lowering the time resolution, encoder-decoders with stride (S) larger than one, are better than increasing hop length. For amplitude normalisation (Ncft) with mean over all dimensions, lowering the number of input frequencies is harmful for accuracy. This might be caused due to the increased importance of low frequencies, where as in lower frequencies, the interaural level differences become useless due to diffraction.

The $\mathcal{F}$ 64 model approaches the performance of the LC3+ codec but still falls short. This discrepancy likely arises because the co-operative (BM-CLA) models incorporate cross-array/cross-ear information only at the broadband latent features level at the central processor, leading to some information loss. This suggests that an earlier integration of information might be beneficial.

Knowledge distillation did not yield any improvement in performance after full training with better teachers, trained on the same dataset as the student. On the other hand, training is accelerated. When teacher models are trained on larger datasets compared to the student, knowledge distillation significantly enhances accuracy. Potentially superior teacher models could be developed using larger architectures trained on more extensive datasets and incorporating an additional convolution layer. The inter-ear amplitude information appears to be efficiently processed within the 1D convolution layer

alongside the features. At first sight, adding an extra branch with inter-ear information did not seem to confer a lot of advantage.

VII. CONCLUSION

This master's dissertation presents an improvement of the BM-CLA model by presenting it as co-operative system aimed at bandwidth reduction when transmitting data. Accuracy and bandwidth were benchmarked against the centralised system baseline, which operated on audio encoded with the lossy LC3+ codec at 16 kbps. Models with comparable bandwidth for data transmission exhibited comparable performance compared to the LC3+ model. The BM-CLA model appeared to reach the threshold of its learning capacity when lowering the features and temporal resolution simultaneously. Interestingly, the quantisation of transmitted features did not adversely impact performance. Using amplitude normalisation over channel, frequency and time dimension, improved overall performance for both the co-operative and centralised model. A proof-of-concept for knowledge distillation in sound localisation tasks is successfully demonstrated. For gaining extra performance in the BM-CLA model with no data adjustments, adding one extra intra-ear convolution is proposed. Further investigations could explore methods to enhance the baseline performance of the centralised approach, employed with the LC3+ or newer audio codec. Lastly, knowledge distillation could be expanded by utilising larger and more advanced teachers with exploring new training schemes such as those employed in generative adversarial networks.

REFERENCES

- [1] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Spatially selective speaker separation using a dnn with a location dependent feature extraction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 930–945, 2024.
- [2] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [3] N. Madhu, R. Martin, U. Heute, and C. Antweiler, "Acoustic source localization with microphone arrays," *Advances in Digital Speech Transmission*, pp. 135–170, 2008.
- [4] S. Jiang, L. Wu, P. Yuan, Y. Sun, and H. Liu, "Deep and cnn fusion method for binaural sound source localisation," *The Journal of Engineering*, vol. 2020, no. 13, pp. 511–516, 2020.
- [5] S. Chakrabarty and E. A. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 136–140.
- [6] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Exploiting temporal context in cnn based multisource doa estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1594–1608, 2021.
- [7] A. Bohlender, L. Roelens, and N. Madhu, "Improved deep speaker localization and tracking: Revised training paradigm and controlled latency," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [8] F. Denk, S. M. Ernst, S. D. Ewert, and B. Kollmeier, "Adapting hearing devices to the individual ear acoustics: Database and target response correction functions for various device styles," *Trends in hearing*, vol. 22, p. 2331216518779313, 2018.
- [9] S. Song, S. Kindt, J. Maes, A. Bohlender, and N. Madhu, "Influence of lossy speech codecs on hearing-aid, binaural sound source localisation using DNNs," in *Forum Acusticum2023, the 10th Convention of the European Acoustics Association*, 2023.
- [10] —, "Comparative study of LC3plus and Lyra codec on DNN-based source localisation for hearing aids," in *Speech Communication; 15th ITG Conference*. VDE, 2023, pp. 71–75.
- [11] S. Kindt, A. Bohlender, and N. Madhu, "2d acoustic source localisation using decentralised deep neural networks on distributed microphone arrays," in *Speech Communication; 14th ITG Conference*. VDE, 2021, pp. 1–5.
- [12] S. Rickard and O. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–529.
- [13] P. Cooreman, A. Bohlender, and N. Madhu, "CRNN-based multi-DOA estimator: Comparing classification and regression," in *Speech Communication; 15th ITG Conference*. VDE, 2023, pp. 156–160.
- [14] E. Accolti and F. Miyara, "Fast and controllable box-shaped room impulse response algorithm," *XIII Reunión de Trabajo en Procesamiento de la Información y Control, RPIC*, 2009.
- [15] M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent, "Sound source localization," *European Annals of Otorhinolaryngology, Head and Neck Diseases*, vol. 135, no. 4, pp. 259–264, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187972961830067X>
- [16] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 451–455.
- [17] F. Ruffy and K. Chahal, "The state of knowledge distillation for classification," *arXiv preprint arXiv:1912.10850*, 2019.
- [18] H.-H. Chou, C.-T. Chiu, and Y.-P. Liao, "Cross-layer knowledge distillation with kl divergence and offline ensemble for compressing deep neural network," *AP-SIPA Transactions on Signal and Information Processing*, vol. 10, p. e18, 2021.

1D geluidslokalisatie voor gehoorapparaten met lage data transfer voor een gedistribueerd model

Jonas Van Damme

Promotor: Prof. dr. ir. Nilesh Madhu

Begeleiders: Stijn Kindt, Siyuan Song, Jasper Maes

Samenvatting—Gehoorapparaten hebben een aanzienlijke evolutie doorgemaakt en zijn getransformeerd tot apparaten die in staat zijn tot aanzienlijke berekeningen op het oor. Naast deze berekeningen wordt verwerking met hoge eisen in het gehoorapparaat systeem uitgevoerd in een gecentraliseerd apparaat, waar meer rekenkracht beschikbaar is. Dit vereist echter (draadloze) gegevensoverdracht tussen deze apparaten, wat de batterijduur nadelig kan beïnvloeden. Verschillende strategieën voor bandbreedte-efficiënte audio bronlokalisatie voor binaurale gehoorapparaten worden vergeleken. Een veelgebruikte strategie is het gebruik van audiocodexs, zoals de LC3+ codec, om de bandbreedte te minimaliseren. Hier wordt ook een bandbreedte-gecomprimeerde versie van een state-of-the-art gedistribueerd systeem onderzocht. Het coöperatieve model verwerkt eerst de microfoonsignalen lokaal voordat de functies naar de centrale processor worden verzonden voor verdere analyse. Het verlagen van de bandbreedte van het systeem impliceert een geschikte compressie van deze functies. Het onderzoek omvat verschillende optimalisaties, waaronder de kwantisatie van de ingebedde functies, aanpassingen aan de functiedimensies, evenals wijzigingen in tijd- en frequentieresoluties. In het algemeen leidden de cumulatieve aanpassingen tot een verbeterde prestatie die iets lager, maar goed vergelijkbaar was met het gecentraliseerde basismodel. Bovendien zijn proof-of-concept experimenten met *knowledge distillation* en potentiële uitbreidingen van het coöperatieve model succesvol getest.

Index Terms—gehoorapparaten, DNN-gebaseerde geluidsbron detectie, bandbreedte optimalisatie, spraak codec, *knowledge distillation*

I. INLEIDING

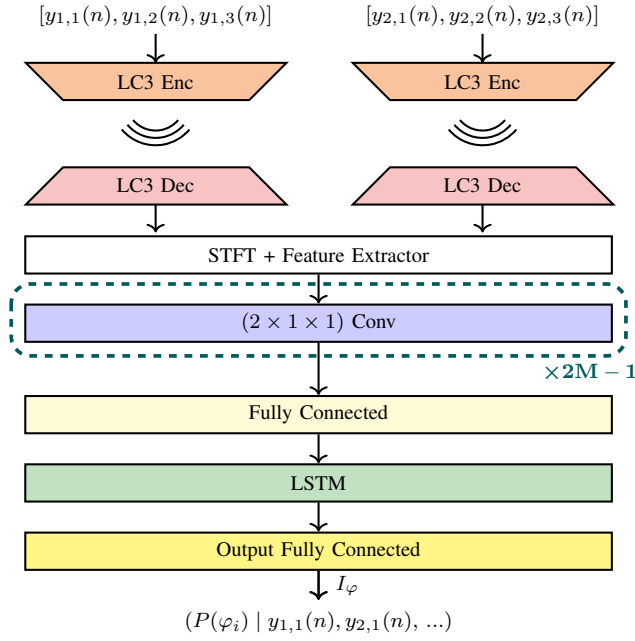
Horen is een van de belangrijkste zintuigen, essentieel voor communicatie, creativiteit en zelfs overleven. Met een toenemende levensverwachting zal ook het aantal mensen met leeftijdsgebonden gehoorverlies toenemen, wat de vraag naar geavanceerdere gehoorapparaten zal doen stijgen. De binaurale aard van menselijk gehoor helpt bij het waarnemen van auditieve signalen zoals richting en afstandsschatting, wat belangrijk is voor bronlokalisatie en -scheiding. Om deze natuurlijke auditieve ervaring na te bootsen met behulp van binaurale gehoorapparaten, is bronlokalisatie dan ook een belangrijke eerste stap.

Typisch wordt de richting van aankomst (DOA) geschat in het azimuthale vlak en gebruikt om de daaropvolgende geluidsscheidingssfasen [1] te informeren voor behoud/creatie van binaurale signalen. Bestaande methoden voor het schatten van DOA in vrije velden maken gebruik van meerdere microfoons en/of arrays [2]–[4] om het geluidsveld ruimtelijk

te bemonsteren en zo de DOA van de verschillende bronnen af te leiden. *State-of-the-art* (SotA) oplossingen voor DOA-schatting maken gebruik van diepe neurale netwerken (DNN). Gewoonlijk maken de modellen gebruik van convolutionele neurale netwerken (CNN) getraind op gesimuleerde audiodata [5]. Temporele context werd toegevoegd in het netwerk voor betere tracking van bewegende bronnen [6], [7]. Temporele convolutionele netwerken (TCN) en *long short-term memory* (LSTM) maakten temporele context in de CNN [5] mogelijk. Aangezien er afstapt wordt van het gebruik van vrijstaande arrays, waarbij die op het oor worden gedragen in gehoorapparaten, is het essentieel om de opgenomen hoorapparaat- en hoofdgerelateerde overdrachtsfuncties (HD-HRTFs) [8] van de specifieke microfoons in de trainingsmethodologie op te nemen.

Verwerking binnen de configuratie van de achter-het-oor (BTE) apparaten kan worden uitgevoerd binnen de apparaten zelf of bij een gecentraliseerde verwerkingsunit. De communicatie tussen deze apparaten kan draadloos worden gefaciliteerd met behulp van protocollen zoals Bluetooth *low-energy* (BLE). Dit brengt een andere beperking met zich mee naast nauwkeurigheid of modelgrootte: de beperkte bandbreedte van de communicatieplatformen. De ruwere BLE LC3+ codec wordt gebruikt in het basismodel van vorig werk [9], [10], waarbij de audiodata van elk BTE-apparaat wordt verzonden naar het centrale verwerkingspunt, waar DOA-schatting wordt uitgevoerd. De resultaten tonen aan dat wanneer de codec wordt opgenomen in de trainingslus, de lokaliserings-nauwkeurigheid vergelijkbaar is met (of zelfs iets beter dan) de DNN getraind en getest met de volledige bandbreedte (niet-gecomprimeerde) audio.

Hier wordt een alternatief onderzocht waarbij, in plaats van audiosignalen rechtstreeks naar de centrale processor te sturen, elk hoorapparaat eerst de opgenomen audio onafhankelijk verwerkt en ingesloten functies naar de centrale processor stuurt voor de definitieve DOA-schatting [11]. Om de bandbreedte in dit model te optimaliseren, worden verschillende technieken onderzocht, zoals: het kwantiseren van de functie-embeddings, het verminderen van het aantal embeddings dat per seconde wordt verzonden, een andere invoernormalisatie en het verlagen van de embeddingdimensie. Bovendien zijn *knowledge distillation* en modelexpansie onderzocht. Het artikel is als volgt georganiseerd: Sectie 2 beschrijft eerst de twee lokaliseringsparadigma's en de overwegingen voor bandbreedte-



Figuur 1: Het gecentraliseerde lokalisatie-algoritme. Aanvankelijk worden de microfoonsignalen verkregen van de LC3+ codec, gevolgd door de schatting van de aankomsthoek (DOA) voor de inkomende geluidsbronnen.

optimalisatie. Sectie 3 bespreekt de optimalisaties in meer detail. In Sectie 4 worden de geteste knowledge distillatie-ontechieken uitgelegd. Sectie 5 geeft een overzicht van de gebruikte methoden voor modelexpansie. De experimentele evaluatie en resultaten worden besproken in Sectie 6, gevolgd door de conclusies in Sectie 7.

II. MODELLEN

A. Signaal model

Het m de microfoonsignaal bij oor e wordt aangeduid als $y_{e,m}(n)$. Dit is een combinatie van de bijdragen van J bronnen $x_{e,m,j}(n)$ en additieve ruis $v_{e,m}(n)$:

$$y_{e,m}(n) = \sum_{j=0}^{J-1} x_{e,m,j}(n) + v_{e,m}(n) \quad (1)$$

waarbij n de sample index is voor een signaal gesampled aan f_s hertz.

Alle netwerken werken met de *short-term Fourier transform* (STFT) van de microfoonsignalen, die worden weergegeven als $Y(k, l)$, waarbij k de frequentie-index is en l de tijdsindex. Dit wordt berekend met een over de wortel genomen Van-Hann venster, framegrootte K en hopgrootte L . Deze representatie wordt vaak gekozen voor sprekerlokalisatie en -scheiding vanwege de spaarzaamheid en de gescheidenheidseigenschappen van spraaksignalen [12] in het STFT-domein.

B. Gecentraliseerd lokalisatie model

Fig.1 geeft de modelarchitectuur weer van het gecentraliseerde model uit [9]. Voor dit model worden de microfoonsignalen eerst naar de centrale verwerkingsunit gestuurd. Dit kan worden gedaan door de volledige bandbreedtesignalen te verzenden of door gebruik te maken van een audiocodec. Zoals aangetoond in [10], is de keuze van de codec belangrijk. Het gebruik van het ruwe LC3+ gecodeerde signaal tijdens de training resulteerde in dezelfde (of soms zelfs iets hogere) lokalisatienauwkeurigheid dan het gebruik van de volledige bandbreedtesignalen. Daarom is de LC3+ codec, een standaard codec voor Bluetooth low energy, gekozen voor de studie hier. Invoerkenmerken, geëxtraheerd uit de volledige bandbreedte of ruw gecodeerde data, bestaan uit de sinus en cosinus van de short-term Fourier transformatie en de genormaliseerde STFT-amplitude. De normalisatie van de amplitude wordt uitgevoerd over alle microfoons (kanalen) en zal verder worden aangeduid als N_c :

$$F_{e,m}(k, l) = \begin{bmatrix} \sin(\angle Y_{e,m}(k, l)) \\ \cos(\angle Y_{e,m}(k, l)) \\ |\overline{Y_{e,m}}(k, l)| \end{bmatrix} \quad (2)$$

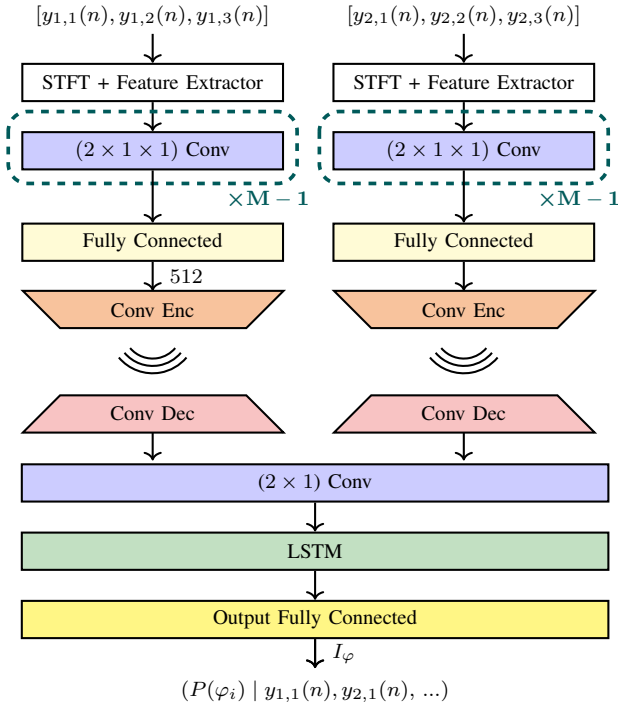
met $\angle Y_{e,m}$ de fase van $Y_{e,m}$ and $\overline{Y_{e,m}} = |Y_{e,m}| / \left(\frac{1}{M} \sum_{\nu=1}^M |Y_{e,\nu}| \right)$. Hier is, $M_e = 3$ het aantal microfoons in elk BTE-apparaat, en $M = 2M_e$ is het totaal aantal microfonen in de binaurale opstelling.

Bij de centrale computereenheid worden de features gerangschikt langs een microfoon-kanaaldimensie op zo'n manier dat de microfoons bij verschillende oren worden afgewisseld: $\mathbf{F}(k, l) = [F_{1,1}(k, l), F_{2,1}(k, l), F_{1,2}(k, l), \dots]$. Voorlopige evaluaties toonden aan dat dit de optimale configuratie was voor de daaropvolgende convolutieblokken, omdat dit leidt tot opeenvolgende convoluties met microfoons aan tegenovergestelde oren. De $M - 1$ convolutionele blokken (**Conv**) hebben elk een kernelgrootte van 2 langs de microfoon-kanaaldimensie, en combineren iteratief de microfoons van de binaurale array. Merk op dat op dit punt elk tijd-frequentiepunt nog steeds onafhankelijk wordt verwerkt.

De volledig verbonden (**FC**) laag combineert de smalle bandfrequentie-inhoud onafhankelijk voor elk tijdframe. De **LSTM** laag integreert vervolgens temporele context op een causale manier. Een laatste **Output FC** wordt gebruikt om van de embedde features naar een *classifier* te gaan, waarbij elke klasse een DOA-sector vertegenwoordigt. De DOA-resolutie wordt bepaald door het aantal DOA-classes I_φ – meer classes zorgen voor een nauwkeurigere schatting, maar vereisen meer training. Voor het meerdere bronnen geval ($J > 1$), bleek dat de keuze voor classificatiegebaseerde lokalisatie beter presteerde dan regressiegebaseerde outputs [13].

C. Gedistribueerd systeem

Het gecentraliseerde lokalisatiemodel werkt goed, maar vertrouwt op audiocoders om de bandbreedte te verminderen. De LC3+ codec (een golfvormcodec) kon de gegevensoverdracht verlagen tot 16 kbps per microfoonkanaal en, hoewel verliesgevend, had het nog steeds geen invloed op



Figuur 2: De architectuur voor breedbandmixende gedistribueerde lokalisatie (BM-CLA). Hier vindt de mix van informatie plaats na de FC-laag, en wordt nadien ook uitgevoerd door een inter-arrayconvolutie.

de lokalisatieprestaties wanneer deze in de training werd opgenomen [9]. Het verlagen van de *bitrates* vereist vaak neurale netwerkgebaseerde generatieve modellen. Maar deze zijn geoptimaliseerd voor gegevensoverdracht via één kanaal en behouden ruimtelijke informatie niet effectief [10]. Een andere verbetering zou kunnen worden geboden door embedde features onafhankelijk op elk oor te extraheren en deze naar de centrale processor te sturen voor een definitieve DOA-schatting. Vervolgens wordt de uitdaging van een lage bandbreedte vertaald naar het compact maken van de embedde featurevector op een geschikte manier.

Een dergelijk model voor het delen en verzenden van features wordt afgebeeld in fig. 2, en is gebaseerd op de breedbandmixende, gedistribueerde lokalisatiearchitectuur (BM-CLA) van [11]. In het oorspronkelijke artikel was het doel om informatie uit te wisselen tussen twee gedistribueerde microfoonarrays en de lokalisatienauwkeurigheid te verbeteren door embedde features te delen. Echter, in deze context zal de architectuur worden aangepast om embedde features met een lage bitrate naar de centrale processor te sturen. Deze architectuur deelt dezelfde onderliggende structuur als het gecentraliseerde model, wat ze ook geschikt maakt voor een eerlijke vergelijking.

In plaats van de gegevens rechtstreeks naar de centrale processor te sturen, worden dezelfde features als in vergelijking 2 afzonderlijk berekend in elk oorapparaat. Deze

worden opnieuw gecombineerd door middel van $M_e - 1$ convolutielaag (**Conv**), waarna de dunne-bandige features worden gecombineerd om breedbandinformatie te produceren met behulp van de volledig verbonden laag (**FC**). Dit levert de embedde features op die men naar de centrale processor wenst te verzenden, omdat al is aangetoond in [11] dat het delen van breedbandinformatie voordeliger was dan dunne-bandinformatie.

D. Bandbreedte vermindering in het gedistribueerde model

Een convolutionele encoder (**Conv Enc**) en decoder (**Conv Dec**) langs de tijdsas worden toegevoegd om de bandbreedte-vereisten te verminderen om deze embedde features te verzenden. Bij de centrale processor, na het decoderen, combineert een laatste convolutionele laag de informatie van de twee oorapparaten samen. Een LSTM en een laatste, volledig verbonden uitvoerlaag worden toegevoegd om de activiteit van de bron in elke DOA-sector te voorspellen.

De convolutionele encoder kan de bandbreedte op drie hoofdmethoden verminderen. Ten eerste kan de featuredimensie $\mathcal{F}$ verminderd worden door een kleinere uitvoerdimensie te hebben dan de invoerdimensie, waardoor de informatie effectief wordt gecomprimeerd. Ten tweede verhoogt het vergroten van de tijdsprong S de tijdscompressie van het model, waardoor de frequentie van de functieoverdracht naar de centrale processor wordt verlaagd. Ten slotte kunnen de features ook worden gekwantiseerd om het aantal bits om de features voor te stellen verder te verminderen. De convolutionele decoder is eigenlijk een deconvolutionele laag, die de bewerking die in de encoder zijn uitgevoerd, omkeert. De featuredimensie wordt opnieuw uitgebreid, temporale interpolatie wordt uitgevoerd en, waar feature-kwantisering wordt uitgevoerd, wordt de bitdiepte nadien opnieuw verhoogd tot de normale 32-bit floating point-nummers.

E. Bandbreedte berekeningen

Het gecentraliseerde systeem met niet-gecomprimeerd geluid heeft een bandbreedte (bits per seconde(bps)) van:

$$BW_{\text{uncompressed}} = 2 \times M_e \times f_s \times b, \quad (3)$$

Waarbij de bitdiepte (b) meestal gelijk is aan 32 bits per sample.

De bandbreedte voor data transmissie met codec kan als volgt berekend worden:

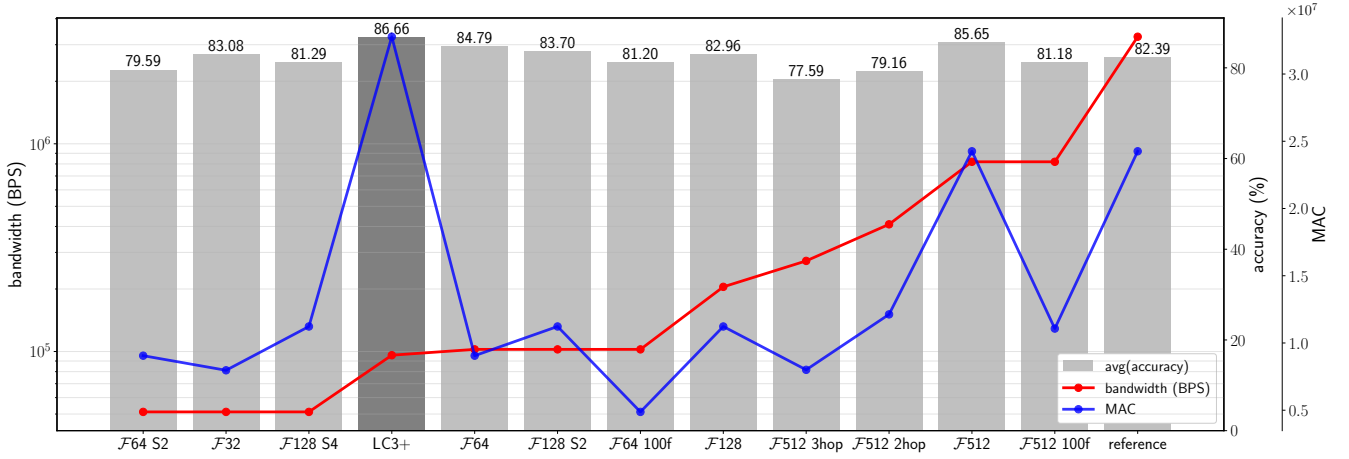
$$BW_{\text{coded}} = 2 \times M_e \times \text{bit rate}, \quad (4)$$

Met de laagste *bit rate* van 16 kbits per seconde voor de LC3+ codec.

Voor gedistribueerde netwerken wordt de bandbreedte berekend net achter de convolutionele encoder laag. De formule gaat als volgt:

$$BW_{\text{coöperatief}}(\mathcal{F}, S, Q) = 2 \times \mathcal{F} \times \frac{f_s}{L \times S} \times \frac{b}{Q}, \quad (5)$$

Dit maakt een duidelijke splitsing in de drie componenten waar de bandbreedte kan worden verminderd. Verminderen



Figuur 3: De staafdiagram toont gemiddelde DOA-nauwkeurigheden (%) voor modellen uitgerust met PTSQ en normalisatie van kanaal-, tijd- en frequentieamplitude. $\mathcal{F}$ duidt op het aantal features, S staat voor encoder-decoder met gegeven stapgrootte, en f staat voor het aantal invoerfrequenties. Parameters die niet expliciet worden genoemd, zijn consistent met die van het referentiemodel. Het referentiemodel is het BM-CLA-model met parameters zoals in [11], zonder kwantisatie en normalisatie zoals het gecentraliseerde model. De LC3+ is het basismodel uitgerust met de LC3-codec. De y-as voor bandbreedte is logaritmisch.

van het aantal features $\mathcal{F}$ per tijdsframe, het vergroten van de hop-grootte L of de stapgrootte S , of het verminderen van het aantal bits per feature met de kwantisatiefactor Q . Merk op dat deze bandbreedte onafhankelijk is van het aantal microfoons per oor, aangezien dit wordt gecomprimeerd met de convolutionele lagen.

Merk op dat voor het gecentraliseerd systeem de beschikbaarheid van de audiosignalen bij de centrale processor ook nuttig kan zijn voor andere toepassingen zoals spraakverbetering, scèneclassificatie, etc. Dit is echter zeer afhankelijk van de toepassing en hoewel hier alleen gekeken wordt naar de localisatie nauwkeurigheid, moeten deze andere toepassingen ook worden overwogen in volledig uitgewerkte implementaties.

III. BANDBREEDTE REDUCTIETECHNIKEN

Deze sectie beschrijft de geteste methode voor het verbeteren van de data wisselwerking in het BM-CLA model.

A. Experimentele configuratie

Het *image-source model* [14] wordt gebruikt om de kamer impulsantwoorden (RIRs) te genereren voor het simuleren van de nagalm in de kamer. Voor het binaurale hoortoestel worden het hoofd en de achter-het-oor hoortoestellen gesimuleerd met de hoortoestel- en hoofd-gerelateerde overdrachtsfunctie (HRTF).

De output van de netwerken $P(\varphi_i) | F(m), F(m-1), \dots$ is een *multi-hot* vector met een DOA range hetzelfde als de gekozen resolutie, bestaande uit 0 (inactieve sectoren) en 1 (actieve sectoren). Een vaste resolutie van 5° met $I = 72$ klassen is gekozen voor een 0° tot 360° azimuth breedte. Het basismodel bestaat uit 32.75×10^6 multiply-accumulate operaties (MAC) en 10.59×10^6 parameters.

B. Kwantisatie

Post-training static quantisation (PTSQ) is een methode die gebruikt wordt voor compressie van de waarden in neurale netwerk modellen. Deze techniek houdt in dat het floating-point datatype, dat typisch wordt weergegeven met 32 bits per feature, wordt omgezet naar een *unsigned integer* formaat met 8 bits. Het gebruik van een unsigned integer is haalbaar dankzij de voorafgaande gelijkrichter (ReLU) activatiefunctie, die zorgt voor niet-negativiteit van de waarden. Vervolgens, zodra de waarden zijn gecomprimeerd naar een integer formaat, worden ze weer gedecomprimeerd naar float32. Dit biedt meer flexibiliteit tijdens de latere stadia van de netwerkverwerking.

C. Aantal van embedde features

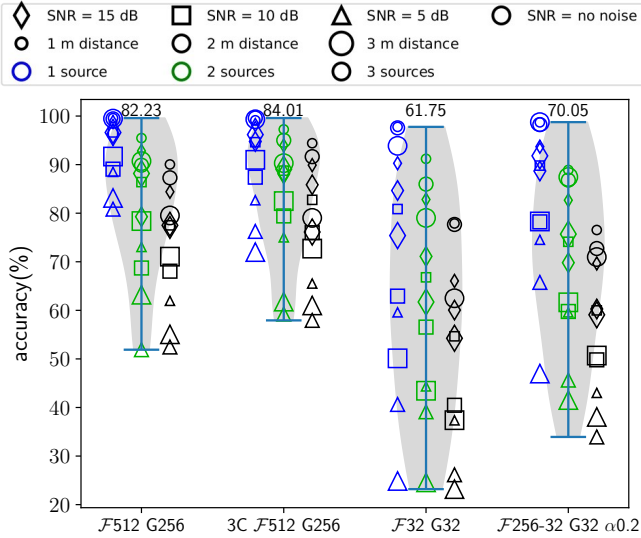
Het aanpassen van het aantal embedde features zorgt voor een vernauwing van het knelpunt in de modellen. Deze veranderingen verzorgen een verfijnde selectie van breedband-embedde features. Hierbij wordt de nauwkeurigheid, voor het mogelijke, gewaarborgd.

D. Resolutie in tijd

De ingebedde localisatie features zijn berekend per STFT frame en dan verzonden naar de centrale berekeningseenheid. De frequentie van deze handeling kan worden berekend als volgt:

$$f_f = \frac{f_s}{L \times S} \quad (6)$$

Wanneer de tijdsresolutie wordt verminderd door de parameter L te verhogen, worden alleen specifieke tijdreeksen in beschouwing genomen. Als alternatief kan temporele compressie worden bereikt door het gebruik van een convolutionele encoder, gevolgd door een convolutionele decoder binnen het netwerkknelpunt, zoals geïllustreerd in Fig.2. In tegenstelling



Figuur 4: De violplot toont de testnauwkeurigheid. Aan de linkerkant zijn twee modellen te zien die getraind zijn op de dataset zoals in [9] met 512 features $\mathcal{F}$ en GRU (G) 256. De 3C duidt op één extra convolutie. Aan de rechterkant zijn de modellen te zien die met minder data zijn getraind. Het meest rechtse model is getraind met een teacher met 256 features.

tot eenvoudig de reeksen over te slaan, maakt deze methode het mogelijk om compressiepatronen te leren terwijl alle relevante informatie behouden blijft.

E. Resolutie in frequentie

Talrijke studies hebben aangetoond dat lage frequenties voornamelijk invloed hebben op interaurale tijdverschillen (ITD), terwijl hoge frequenties voornamelijk invloed hebben op interaurale niveauverschillen (ILD) [15] [16]. In dit verband worden de eerste 100 frequenties uit de oorspronkelijke set van 257 frequenties gehaald. Daardoor strekt de bovengrens van de frequentie zich slechts uit tot 3125 Hz, in plaats van 8000 Hz. Hoewel het uitsluiten van de hoogste frequenties het probleem vereenvoudigt op mathematisch niveau, gaat dit ook gepaard met een verlies aan gedetailleerde informatie. Echter, een dergelijke vereenvoudiging kan gunstige resultaten opleveren voor de methoden die in voorgaande secties zijn voorgesteld.

F. Normalisatie van de amplitude

De amplitude-normalisatie uit sectie II-B zoals voor het gecentraliseerde model, zou ineffectief zijn voor het coöperatieve model vanwege de extra bandbreedte van $2 \times K \times \frac{f_s}{L} \times b$. Om dit op te lossen, wordt normalisatie over het kanaal, frequentie en tijd uitgevoerd. Dit resulteert in slechts $2 \times b$ bits per seconde.

IV. KNOWLEDGE DISTILLATION

Om kleinere modellen operationeel te maken, komt knowledge distillation (KD) naar voor. Een betere en vaak grotere teacher helpt de student, vaak een kleiner model, tijdens de

training voor betere generalisatie. Drie verschillende trainings-schema's met offline, online en afwisselende training werden getest. Daarnaast werden verschillende leraar-student combinaties getraind om de afhankelijkheid van de leraar te testen. De verliesfunctie (loss) kan als volgt worden geschreven:

$$Loss = (1 - \alpha)BCE(\hat{y}^S, y) + \alpha(KL(\hat{y}^S, \hat{y}^T)) \quad (7)$$

De parameter α is een gewicht voor het verlies uit het verschil tussen de waarden van de voorspellingen van de student ($\hat{y}^S$) en de leraar ($\hat{y}^T$), die worden onderworpen aan logsoftmax en softmax. Dit verlies wordt berekend met de Kullback-Leibler (KL) verliesfunctie, die vaak wordt gebruikt in KD. Dit verlies wordt toegevoegd aan het originele verlies zonder leraar, dat wordt berekend met de binomiale kruisentropie (BCE) verliesfunctie tussen de voorspellingen van de student en de ware labels.

A. Attentie netwerk

Voor een dynamisch gewicht α , werd er geprobeerd om het gewicht te leren met een enkele volledig verbonden laag met de predictied van de leraar en de student als invoer. Hierbij werd het attentienetwerk geleerd op de verliesfunctie van de student om het KD-gewicht te voorspellen.

B. Ensemble leren

Meerdere leraarmodellen kunnen samen worden ingezet [17], met het idee om een betere generalisatie te verkrijgen. Voor het implementeren van het verlies van deze meerdere leraren werden keuzes tussen gemiddeld verlies of best verlies getest.

V. MODEL EXPANSIE

Aangezien knowledge distillation afhankelijk is van het hebben van een betere leraar, werden de BM-CLA-modellen aangepast om de maximale nauwkeurigheid uit de gegeven data te halen. De trainingsdata werd niet vergroot om zich alleen te kunnen richten op de modelarchitectuur.

A. Circulaire opvulling

De limiet van convoluties wordt bepaald door het aantal microfoons per oor. Dit resulteert in $M_e - 1$ convoluties op elk oor. Door circulaire padding te gebruiken, kan dit aantal stapsgewijs met één worden verhoogd na elke convolutie. Hierdoor kan aanvullende informatie worden verzameld van de intra-oor microfoons.

B. Overbruggende connectie

Inter-oor informatie wordt pas uitgewisseld na de vorming van breedbandfeatures, wat kan leiden tot het verlies van cruciale informatie. Om dit mogelijke verlies te minimaliseren, wordt amplitude-informatie van één microfoon aan elk oor gedeeld. Deze informatie wordt op dezelfde manier verwerkt als de andere invoergegevens voordat ze worden gedeeld in de volledig verbonden laag.

Tabel I: Gemiddelde nauwkeurigheden in procenten worden weergegeven in deze tabel. Het eerste deel toont de nauwkeurigheid van studenten ($\mathcal{F}32L32$) met verschillende leraren, getraind gedurende 60 en 120 epochs. Het tweede deel vergelijkt de skip-variant met het referentiemodel. Het laatste deel vergelijkt volledig getrainde modellen met nieuwe normalisatie (Ncft) versus oude normalisatie (Nc).

model	accuracy (%)	epochs
(KD) base- $\mathcal{F}32 L32$	70.80	60
(KD) base $\mathcal{F}256-32 L32 \alpha 0.2$ best	68.78	60
(KD) base $\mathcal{F}256-32 L32 \alpha 0.2$ avg	67.96	60
(KD) $\mathcal{F}256-32 L32 \alpha 0.2$	72.15	60
(KD) $\mathcal{F}256-32 L32 \alpha 0.2$ online	65.47	60
(KD) $\mathcal{F}256-32 L32$ Att alternate	68.92	60
(KD) $\mathcal{F}256-32 L32$ Att	69.28	60
$\mathcal{F}32 L32$	65.45	60
(KD) $\mathcal{F}256-32 L32$ Att	73.99	120
$\mathcal{F}32 L32$	73.41	120
skip	83.75	120
reference	82.39	120
baseline Ncft	87.64	60
baseline	85.65	60

VI. RESULTATEN & DISCUSSIE

Zoals weergegeven in fig. 3, leek het verlagen van de features het meest gunstig om de bandbreedte te verlagen en de nauwkeurigheid te behouden. Bovendien verminderde ook het aantal MAC's. Voor het verlagen van de tijdsresolutie zijn encoder-decoders met een stride (S) groter dan één beter dan het vergroten van de hoplength. Voor amplitude-normalisatie (Ncft) met het gemiddelde over alle dimensies is het verlagen van het aantal invoerfrequenties schadelijk voor de nauwkeurigheid. Dit kan worden veroorzaakt door het grotere belang van lage frequenties, waarbij bij lagere frequenties de interaurale amplitudeverschillen door diffractie nutteloos worden.

Het model $\mathcal{F}64$ benadert de prestaties van de LC3+ codec, maar komt toch tekort. Deze discrepantie ontstaat waarschijnlijk doordat de coöperatieve (BM-CLA) modellen alleen op het niveau van de breedband embedde features op de centrale processor informatie tussen arrays/oren integreren, wat leidt tot enig informatieverlies. Dit suggereert dat een eerdere integratie van informatie gunstig zou kunnen zijn.

Knowledge distillation leverde geen verbetering op in prestaties na volledige training met betere leraren, getraind op dezelfde dataset als de student. Aan de andere kant wordt de training versneld. Wanneer leraarmodellen worden getraind op grotere datasets in vergelijking met de student, verbetert knowledge distillation de nauwkeurigheid aanzienlijk. Mogelijk kunnen superieure lerarenmodellen worden ontwikkeld met behulp van grotere architecturen die zijn getraind op uitgebreidere datasets en het opnemen van een extra convolutielaag. De inter-oor amplitude-informatie lijkt efficiënt te worden verwerkt binnen de 1D-convolutielaag in de vorm van features. Op het eerste zicht leek het toevoegen van een extra tak met inter-oor informatie niet veel voordeel te bieden.

VII. CONCLUSIE

Een verbetering van het BM-CLA-model werd uitgewerkt door het te presenteren als een gedistribueerd systeem gericht op het verlagen van de bandbreedte bij het verzenden van gegevens. Nauwkeurigheid en bandbreedte werden vergeleken met de basismodel van het gecentraliseerde systeem, dat werkte met audio gecodeerd met de verliesgevende LC3+ codec op 16 kbps. Modellen met vergelijkbare bandbreedte voor gegevensoverdracht vertoonden vergelijkbare prestaties in vergelijking met het LC3+ model. Het leek erop dat het BM-CLA-model de drempel van zijn leercapaciteit bereikte toen de features en de temporale resolutie tegelijkertijd werden verlaagd. Interessant genoeg had de kwantisatie van verzonden features geen nadelige invloed op de prestaties. Het gebruik van amplitude-normalisatie over de kanaal-, frequentie- en tijddimensies verbeterde over het algemeen de prestaties van zowel het coöperatieve als het gecentraliseerde model. Een proof-of-concept voor knowledge distillation bij geluidslokalisatietaken wordt succesvol gedemonstreerd. Om extra prestaties te behalen in het BM-CLA-model zonder gegevensaanpassingen voor te stellen, wordt het toevoegen van één extra intra-oor-convolutie voorgesteld. Verdere onderzoeken zouden methoden kunnen verkennen om de al indrukwekkende prestaties van het gecentraliseerde model te verbeteren, toegepast met de LC3+ of een nieuwere audiocodec. Ten slotte zou knowledge distillation kunnen worden uitgebreid door gebruik te maken van grotere en geavanceerdere leraren en het verkennen van nieuwe trainingsmethoden zoals die worden gebruikt in generatieve tegenstandernetwerken (GAN).

REFERENTIES

- [1] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Spatially selective speaker separation using a dnn with a location dependent feature extraction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 930–945, 2024.
- [2] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [3] N. Madhu, R. Martin, U. Heute, and C. Antweiler, "Acoustic source localization with microphone arrays," *Advances in Digital Speech Transmission*, pp. 135–170, 2008.
- [4] S. Jiang, L. Wu, P. Yuan, Y. Sun, and H. Liu, "Deep and cnn fusion method for binaural sound source localization," *The Journal of Engineering*, vol. 2020, no. 13, pp. 511–516, 2020.
- [5] S. Chakrabarty and E. A. Habets, "Broadband doa estimation using convolutional neural networks trained with noise signals," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 136–140.
- [6] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Exploiting temporal context in cnn based multisource doa estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1594–1608, 2021.

- [7] A. Bohlender, L. Roelens, and N. Madhu, "Improved deep speaker localization and tracking: Revised training paradigm and controlled latency," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [8] F. Denk, S. M. Ernst, S. D. Ewert, and B. Kollmeier, "Adapting hearing devices to the individual ear acoustics: Database and target response correction functions for various device styles," *Trends in hearing*, vol. 22, p. 2331216518779313, 2018.
- [9] S. Song, S. Kindt, J. Maes, A. Bohlender, and N. Madhu, "Influence of lossy speech codecs on hearing-aid, binaural sound source localisation using DNNs," in *Forum Acusticum2023, the 10th Convention of the European Acoustics Association*, 2023.
- [10] —, "Comparative study of LC3plus and Lyra codec on DNN-based source localisation for hearing aids," in *Speech Communication; 15th ITG Conference*. VDE, 2023, pp. 71–75.
- [11] S. Kindt, A. Bohlender, and N. Madhu, "2d acoustic source localisation using decentralised deep neural networks on distributed microphone arrays," in *Speech Communication; 14th ITG Conference*. VDE, 2021, pp. 1–5.
- [12] S. Rickard and O. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–529.
- [13] P. Cooreman, A. Bohlender, and N. Madhu, "CRNN-based multi-DOA estimator: Comparing classification and regression," in *Speech Communication; 15th ITG Conference*. VDE, 2023, pp. 156–160.
- [14] E. Accolti and F. Miyara, "Fast and controllable box-shaped room impulse response algorithm," *XIII Reunión de Trabajo en Procesamiento de la Información y Control, RPIC*, 2009.
- [15] M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent, "Sound source localization," *European Annals of Otorhinolaryngology, Head and Neck Diseases*, vol. 135, no. 4, pp. 259–264, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187972961830067X>
- [16] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 451–455.
- [17] H.-H. Chou, C.-T. Chiu, and Y.-P. Liao, "Cross-layer knowledge distillation with kl divergence and offline ensemble for compressing deep neural network," *APSIPA Transactions on Signal and Information Processing*, vol. 10, p. e18, 2021.

Contents

Abstract	v
List of figures	xxii
List of tables	xxvii
List of abbreviations	xxix
List of symbols	xxxi
1 Introduction	1
1.1 Sound localisation for hearing aids	1
1.1.1 Diffuse field	2
1.1.2 Hardware layout	2
1.2 Behind-the-ear hearing aids	2
1.3 Compared models	3
1.4 Research questions and objectives	3
2 Literature review	4
2.1 Human hearing limits	4
2.2 Sound localisation	4
2.2.1 Physical characteristics of sound	5
2.2.2 Short-term Fourier transform	5
2.2.3 Room acoustics and room impulse responses	5
2.2.4 Binaural hearing	7
2.2.5 Beamforming	9
2.2.6 Steered response-power phase transform	10
2.2.7 Multiple signal classification	10
2.3 Machine learning	11
2.3.1 Neural networks	11
2.3.2 Convolutional layers	13
2.3.3 Recurrent layers	14
2.3.4 Codec	18

2.4	Network compression	20
2.4.1	Pruning	20
2.4.2	Knowledge distillation	20
2.4.3	Generative adversarial network	22
3	Materials	24
3.1	Software	24
3.2	Hardware	24
3.3	Data	24
4	Foundational model structures	27
4.1	Signal model	27
4.1.1	Amplitude normalisation	28
4.2	Centralised model	28
4.3	Co-operative model	29
4.3.1	Network layer equations	30
4.4	Model name conventions	32
4.5	Test method	33
5	Model compression	34
5.1	Methods	34
5.1.1	Metrics	34
5.1.2	Quantisation	36
5.1.3	Reducing layer size	37
5.1.4	Time resolution	37
5.1.5	Frequency resolution	37
5.1.6	Amplitude normalisation	38
5.1.7	Knowledge distillation	38
5.1.8	Attention network	40
5.1.9	Ensemble learning	41
5.1.10	Training schemes	41
5.1.11	Small dataset	41
5.2	Results and discussion	41
5.2.1	Training behaviour	42
5.2.2	Quantisation	44
5.2.3	Reducing layer size	45
5.2.4	Time - frequency resolution	47
5.2.5	Amplitude normalisation	49
5.2.6	Knowledge distillation	51
5.2.7	GRU	63

5.3	Conclusion	64
6	Model expansion	65
6.1	Methods	65
6.1.1	Input features	65
6.1.2	Hyper-parameters	65
6.1.3	Circular padding	66
6.1.4	Skip connections	66
6.2	Results and discussion	67
6.2.1	Input features	67
6.2.2	Circular padding	74
6.2.3	Skip connection	77
6.2.4	Dropout	78
6.2.5	Overfitting	79
6.3	Conclusion	81
	Conclusion	82
	Ethical and social reflection	82
	References	84
	Appendices	88
	Appendix A: accuracy overview	89

List of figures

1.1	BTE hearing aid configuration (Signia, 2023)	2
2.1	Room impulse response of gunshot (ProSoundWeb · EH Media LLC , 2022)	5
2.2	"Geometrical representation of direct sound wave (solid red line), 1st-order reflection paths (solid blue lines) and the 2nd-order reflection paths (solid black line). The letter S and R refers to a sound source and the receiver " [1] (Anjana Rathnayake, 2019)	6
2.3	Sinc filter	8
2.4	Hanning window	8
2.5	Popular activation functions used in neural networks to learn complex patterns. (Tashmit, 2020)	12
2.6	Neural network with fully connected layers. Input, hidden and output layers with activations shown. X is the input, W is the weight, b is the bias, A is the activation and Y is the predicted output. [2] (Vuckovic, 2015)	12
2.7	CNN architecture sound localisation [3] (Soumitro Chakrabarty , 2019)	14
2.8	"Temporal Convolutional Network comprising L = 4 convolution layers (from bottom to top) with exponentially increasing dilations 2 ⁰ , 2 ¹ ,..., 2 ^{L-1} , all filters have length K = 3 " [4] (Alexander Bohlender , Ann Spriet , Senior Member, IEE, Wouter Tirry, and Nilesh Madhu , 2021)	14
2.9	LSTM block with 4 gates and the GRU block with 3 gates [5] (Phi Michael, 2020)	15
2.10	"The Markov chain with parameter p is used to model the activity of the source with index j: A _j (λ)=1 represents activity whereas A _j (λ)=0 represents inactivity"[4]. (Alexander Bohlender, 2021	16
2.11	CRNN architecture sound localisation [4] (Alexander Bohlender , Ann Spriet, Wouter Tirry, Nilesh Madhu 2021)	16
2.12	The narrowband mixing co-operative localisation architecture (NM-CLA). [6] (Stijn Kindt, 2021)	17
2.13	the broadband mixing co-operative localisation architecture (BM-CLA). H . [6] (Alexander Bohlender , 2021)	18
2.14	The localisation accuracies(%) of the base model on the x-axis and the encoded model on the y-axis. The encoded models are trained with the corresponding codec bit rate of 16 kbps.[7] (Siyuan Song , 2023)	19
2.15	"Localisation accuracies (%) over different evaluation conditions. The left subplot presents results for 3-channel audio transfer and the right subplot for 6-channel audio transfer. Each subplot contains the baseline result, the result obtained when the baseline CRNN is fed codec-processed audio and, lastly, the results of the CRNN, re-trained with the codec-processed audio and tested in matched conditions" [8] (S. Song, S.Kindt, J. Maes, A. Bohlender, N. Madhu , 2021)	19
2.16	Unstructured and structured pruning for deep neural networks and convolutional neural networks. [9] (Chen Qi, 2021)	20
2.17	Local pruning on the left and global pruning on the right for deep neural networks. [10] (Hugo Tessier, 2021)	21

2.18	Knowledge distillation with inheriting knowledge from teacher to student, only updating the student network.	21
2.19	GAN with trainable generator and discriminator, configuration used to train image generators. (Google, 2022)	23
3.1	The positioning of two BTE arrays at each ear and channel index of each microphone (Siyuan Song, 2023)[7], the geometry is the same as in [11].	25
3.2	Speaker directions for the Oldenburg test setup.	25
4.1	The centralised localisation algorithm. Initially, the microphone signals are acquired from the LC3+ codec, followed by the estimation of the direction of arrival (DOA) for the incoming sound sources.	28
4.2	The broadband mixing co-operative localisation architecture (BM-CLA). Here, the mixing of information occurs after the FC layer, and is also performed by an inter-array convolution.	29
4.3	Legend for tested accuracy conditions.	33
5.1	Amplitude normalisations depicted per channel per microphone.	38
5.2	BM-CLA model with $\mathcal{F}=F1$ as teacher and compressed student BM-CLA model with $\mathcal{F}=F2$. Knowledge distillation with 2 additional losses.	40
5.3	Depicts the attention layer to optimise α for knowledge distillation. The fully connected layer takes the logits from both the student and teacher models as input. This input is processed, and the sigmoid function outputs an α value between 0 and 1 for every time index. The mean value per time index is then computed to obtain a single value.	40
5.4	Training BCE loss for 120 epochs starting at epoch 5 for models with different parameter size.	42
5.5	Validation BCE loss for 120 epochs starting at epoch 5 for models with different parameter size.	42
5.6	Comparison of test accuracy for training 60 and 120 epochs.	43
5.7	Boxplot illustrating the difference in accuracy between the averages of 14 models with and without PTSQ.	44
5.8	Test accuracy for adjusted BM-CLA models with varying quantities of broadband embedded features at the fully connected layer was evaluated. All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	45
5.9	Test accuracy for adjusted BM-CLA models with varying quantities of broadband embedded features at the Fully Connected layer, LSTM and Conv1D were evaluated. The models with adjusted temporal layer have an equally adjusted Conv1D layer. All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	46
5.10	Test accuracy for BM-CLA models with focus on time and frequency resolution. The models were trained for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	47
5.11	Test accuracy for BM-CLA models with amplitude normalisation over the channels (N_c) and normalisation over the channels, frequencies and time (N_{cft}) over 120 epochs.	49

5.12	Test accuracy plotted in bars ordered in rising bandwidth from left to right. Bandwidth is illustrated on the red curve and MAC number is illustrated with the blue curve. All models have undergone PTSQ and amplitude normalisation (Ncft) with one normalisation factor. The reference model is the $\mathcal{F}^{512}$ model with Nc normalisation without PTSQ. The darker color represents the centralised architecture. Bandwidth due to normalisation is not included.	50
5.13	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	51
5.14	Test accuracy for BM-CLA models with the following name conventions, teacher-student. Where the teachers have slightly different architecture parameters. All models underwent training for 60 epochs.	52
5.15	Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}^{32}$ L32 trained online and offline over 60 epochs.	53
5.16	Validation BCE loss for 120 epochs starting at epoch 5 for KD online training.	54
5.17	Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}^{32}$ L32 trained with alternation of the attention layer and offline training over 60 epochs.	55
5.18	Validation BCE loss for 120 epochs starting at epoch 5 for KD alternate training.	56
5.19	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and average hidden weight factor (Avg). All models underwent training for 60 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	57
5.20	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs.	58
5.21	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs. The terms "best," "average," "alternate," and "online" refer to distinct training methodologies.	59
5.22	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and learned hidden weight factor (Att). All models underwent training for 60 epochs except for the teacher $\mathcal{F}^{256}$ L256. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	60
5.23	Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and learned hidden weight factor (Att). All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.	60
5.24	Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}^{32}$ G32 trained on smaller dataset over 120 epochs.	62

5.25	Average test accuracy for BM-CLA models with the adjusted temporal layer, across varying numbers of sources. Models utilising the GRU (G) are represented with light colors, while those employing the LSTM (L) are depicted with darker (grey) colors. The centralised baseline is depicted with amplitude normalisation N_c and N_{cft}	63
6.1	Intuitive drawing for circular padding with each color referring to the information from a microphone on the hearing device.	66
6.2	Comparison of test accuracy under various conditions for the baseline model using phase (P) input versus the baseline model using amplitude (A) input over 120 epochs.	67
6.3	Comparison of test accuracy under various conditions for the baseline model using phase (P) input versus the baseline model using both amplitude (A) and phase (P) inputs over 120 epochs.	67
6.4	Difference in test accuracy between the baseline model using phase (P) and amplitude (A) inputs (improvement represented on the positive axis) and the baseline model using phase (P) inputs alone (improvement represented on the negative axis) over 120 epochs.	68
6.5	Comparison of test accuracy under various conditions for the BM-CLA model using phase (P) input versus the BM-CLA model using amplitude (A) input over 120 epochs.	71
6.6	Comparison of test accuracy under various conditions for the BM-CLA model using phase (P) input versus the BM-CLA model using both amplitude (A) and phase (P) inputs over 120 epochs.	71
6.7	Difference in test accuracy between the BM-CLA model using phase (P) and amplitude (A) inputs (improvement represented on the positive axis) and the BM-CLA model using phase (P) inputs alone (improvement represented on the negative axis) over 120 epochs.	73
6.8	Test accuracy for the models utilising a GRU of 256 (G_{256}), $\mathcal{F} = 512$, with circular padding and C convolutions at the ear-device is depicted. The boxplot in grey illustrates the accuracy distribution for each model. The test conditions are detailed in the legend above the plot.	74
6.9	Test accuracy was evaluated for models with varying configurations of convolutional layers, fully connected (FC) layers, and temporal layers. For example, 3C128 represents a model with 3 convolutional layers, each containing 128 filters. The number of nodes in the FC layer is identical to that in the temporal layer and Conv1D layer. The notation 'G' denotes the use of a GRU instead of a LSTM. All models were trained until minimal validation loss, except for the reference models BM-CLA $\mathcal{F}_{512}$ and the baseline model, which were trained for 120 epochs. The boxplot in grey illustrates the accuracy distribution for each model. The test conditions are detailed in the legend above the plot.	75
6.10	Comparison of test accuracy for the BM-CLA model equipped with a GRU and a skip connection.	77
6.11	Comparison of test accuracy for the baseline model with dropout rates $D=0$ trained over 60 epochs and $D=0.5$ trained over 120 epochs.	78
6.12	Comparison of test accuracy for the BM-CLA model with dropout rates $D=0$ trained over 60 epochs and $D=0.5$ trained over 120 epochs.	78
6.13	Binary cross entropy loss curves during training on the training data are displayed above the validation loss curves for models exhibiting signs of overfitting. The nomenclature follows a convention where, for instance, '3C128' indicates three convolutions with 128 filters per convolution. Similarly, the symbol $\mathcal{F}$ denotes the number of nodes in the embedded features, matching the size of the temporal layer. The letter 'G' signifies the utilisation of GRU as the temporal layer, instead of a LSTM.	80

14	Accuracy - bandwidth overview of most interesting compressed and expanded models with the bandwidth axis logarithmic.	89
15	Accuracy - MAC overview of most interesting compressed and expanded models with the MAC axis logarithmic.	90
16	Accuracy - parameter overview of most interesting compressed and expanded models with the parameter axis logarithmic.	91

List of tables

2.1	Used octave bands for calculating RIRs.	9
3.1	"Data generation: training, validation and test set" (Siyuan Song , 2023)	26
4.1	Centralised model hyperparameters.	32
4.2	BM-CLA reference model hyperparameters.	32
5.1	Given names for network layer parameter variables.	34
5.2	Average accuracies in % for all tested conditions for the models depicted in fig. 5.6.	43
5.3	Average accuracies in % for all tested conditions for the models depicted in Figure 5.8.	45
5.4	Average accuracies in % for all tested conditions for the models depicted in Figure 5.9.	46
5.5	Average accuracies in % for all tested conditions for the models depicted in Figure 5.10.	47
5.6	Test accuracy, bandwidth and MAC from models in Figure 5.6.	50
5.7	Average accuracies in % for all tested conditions for the models depicted in Figure 5.13.	52
5.8	Average accuracies in % for all tested conditions for the models depicted in Figure 5.19.	57
5.9	Average accuracies in % for all tested conditions for the models depicted in Figures 5.22 and 5.23.	61
5.10	Average accuracies in % for all tested conditions for the models depicted in Figure 5.25. Bold numbers indicate better with LSTM.	63
6.1	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline A. The averages are taken for all the conditions with the specified constraint.	69
6.2	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline P. The averages are taken for all the conditions with the specified constraint.	69
6.3	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline AP. The averages are taken for all the conditions with the specified constraint.	70
6.4	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ A. The averages are taken for all the conditions with the specified constraint.	71

6.5	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ P. The averages are taken for all the conditions with the specified constraint.	72
6.6	Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ AP. The averages are taken for all the conditions with the specified constraint.	72
6.7	Average accuracies in % for all tested conditions for the models depicted in Figure 6.8.	74
6.8	Average accuracies in % for all tested conditions for the models depicted in Figure 6.9.	76
6.9	Average accuracies in % for changes in dropout rates from Figures 6.12 6.11.	78

List of abbreviations

BCE Binary Cross-Entropy.

BLE Bluetooth Low Energy.

BM-CLA Broadband Mixing Co-operative Localisation Architecture.

BRIR Binaural Room Impulse Response.

BTE Behind-the-Ear.

BW Band width.

CE Cross Entropy.

CNN Convolutional Neural Network.

cosEL Cosine Embedding Loss.

CPU Central Processing Unit.

CRNN Convolutional Recurrent Neural Network.

CUDA Compute Unified Device Architecture.

DNN Deep Neural Network.

DOA Direction of Arrival.

FC Fully Connected.

FLOP Floating-Point Operation.

GCC Generalised Cross Correlation.

GRU Gated Recurrent Unit.

HD Hearing Device.

HRTF Head-Related Transfer Function.

ILD Interaural Level Difference.

ITD Interaural Time Difference.

KD Knowledge Distillation.

KL Kullback-Leibler.

LC-BLSTM Latency Controlled-Bidirectional Long-Short Term Memory.

LSTM Long-Short Term Memory.

MAC Multiply-Accumulate.

MAF Minimal Audible Field.

MAP Minimal Audible Pressure.

MUSIC Multiple Signal Classification.

Nc Normalisation over channel.

Ncft Normalisation over channel, frequency and time.

PTSQ Post-Training Static Quantisation.

ReLU Rectified Linear Unit.

RIR Room Impulse Response.

RNN Recurrent Neural Networks.

RT60 Reverberation Time.

SNR Signal-to-Noise Ratio.

SotA State-of-the-Art.

SPL Sound Pressure Level.

SRP-PHAT Steered Response Power Phase Transform.

STFT Short-Term Fourier Transform.

TCN Temporal Convolutional Network.

TDOA Time Difference of Arrival.

List of symbols

$\mathcal{F}$	Latent features
b	Bit depth
f_f	Frequency of feature transmission
f_s	Sample frequency
I_φ	Number of azimuth angles
J	Number of sources
j	Source index
K	Frame size (frequencies)
k	Frequency bin index
K_s	Kernel size
L	Hop length (samples)
l	Time frame index
M	Number of microphones
m	Microphone index
M_e	Number of microphones per ear
N	Number of frames
N_f	Number of feature maps
N_G	Number of gates in temporal layer type
N_T	Number of temporal layer nodes
N_{Gr}	Number of groups in convolutional layer
N_{in}	Number of input features
N_{out}	Number of output features
PA	Parameter count
S	Stride

1

Introduction

Hearing aids are improving in a fast pace. This is important for making them more accessible for everyone and having a better user experience. For aiming to these goals, simpler hardware should be used. This can make the device more durable and sustainable. In order to not lower the quality of the hearing aid, better software should be used.

The quality of hearing aids can be assessed on different topics. Some examples are, source separation, sound localisation, noise cancelling, sound clarity, occlusion, battery life, comfort, durability, etc.

The focus in this thesis lays on sound localisation and battery life. The sound localisation is important for humans in order to pay better attention and awareness for sounds.

Software modifications can help improving computational efficiency. This can be done by adjusting the used software algorithm in many ways. In this thesis, a deep learning model is taken under the loop in order to reduce the amount of data that are transferred in the used algorithm. The data in the used algorithm will be transferred with Bluetooth low energy. The results of this model will be compared with a baseline from another approach using a codec.

The scope for hearing aids will lay on behind-the-ear (BTE) hearing aids. No real life tests were further done. All tests were done in a virtual setup with simulated and recorded data. Currently existing machine learning models are good performing. These models will be taken as a reference measurement during this research.

1.1 Sound localisation for hearing aids

One of the primary challenges for hearing aid users is understanding speech in noisy environments. Source localisation enables hearing aids to distinguish between the sound source of interest and background noise. This selective amplification improves speech intelligibility and comprehension. Sound localisation is vital for spatial awareness. It allows users to identify the direction of sounds, such as approaching vehicles, alarms, or someone calling out. This is crucial for navigation and personal safety in everyday environments. Furthermore, good localisation restores a more natural listening experience. It helps in recreating a sense of space and distance, allowing users to feel more connected to their environment and enhancing social interactions. By providing clearer sound quality and reducing background noise, source localisation can significantly reduce the cognitive load and listening effort for hearing aid users. This leads to less fatigue and a more comfortable listening experience. In hearing aids binaural hearing is supported, where both ears work together to process sounds. This results

in better sound quality, balance, and clarity.

1.1.1 Diffuse field

Indoor sound localisation faces significant challenges due to the diffuse field effect. In such environments, sound waves reflect off multiple surfaces while creating overlapping reflections and reverberations. This leads to multipath interference, making it difficult to determine the direct sound path. Reverberation further complicates the situation by mixing direct sound with prolonged reflections, obscuring the sound source. The diffuse field causes ambiguity in direction detection and reduces the signal-to-noise ratio, complicating localisation. Additionally, frequency-dependent effects distort the sound signal, as different frequencies are absorbed or reflected differently.

1.1.2 Hardware layout

No hardware will be tested but the software should be used on the end user its devices. Therefore the software should be optimised for its user utilisation. The hearing aid technologies will consist out of multiple wireless connected devices. The ear devices themselves and a central unit.

1.2 Behind-the-ear hearing aids



Figure 1.1: BTE hearing aid configuration (Signia, 2023)

One of the main challenges in inter-ear communication is balancing data transmission rates with power consumption. BTE devices must transmit data effectively while conserving battery life. Ensuring real-time communication between devices is crucial for natural sound perception. High latency can disrupt the synchronisation between the two ears, leading to a disjointed auditory experience. Transmitting complex auditory data at low rates without significant loss of quality is challenging. Effective compression algorithms that do not compromise sound quality are essential. The algorithms, often artificial intelligence models with deep neural networks (DNN), must adapt to varying acoustic environments in real-time. This with posing a computational challenge at low data rates.

Despite remarkable advancements, there are still areas for improvement. For instance, in complex auditory scenes with multiple sound sources, distinguishing individual sources remains a challenge. Additionally, the adaptability of DNNs in

rapidly changing environments needs further enhancement.

Future research could focus on optimising DNN architectures for more efficient low-rate data transmission, improving battery technology for longer device life and developing more advanced algorithms for real-time environmental adaptability and sound source separation.

BTE hearing aids with DNN-based low data rate inter-ear communication represents a significant leap in auditory technology, offering users an improved listening experience. However, the quest for energy-efficient, real-time processing with minimal data loss remains ongoing, with ample scope for future research and development in this field.

1.3 Compared models

Methods for sound localisation exist with being excellent for indoor sound localisation. unfortunately, for the two methods in the scope of this master's dissertation, only one of the models is already optimised for use in BTE hearing aids. The optimised model will be called the centralised baseline and is explained in Section 4.2. The second model using a broadband mixing co-operative localisation architecture, lacks in efficient use of bandwidth during operation, this will be explained in 4.3.

1.4 Research questions and objectives

The primary aim of this master's dissertation is to explore methods to reduce energy use for the broadband mixing co-operative localisation architecture, with primary scope on reducing the bandwidth. The research questions that guide this study are as following:

1. **Research question:** What methods to reduce bandwidth, have lowest impact on accuracy performance?
Objective: Explore dimensions and parameters that can be adjusted to reduce bandwidth during data transmission. Testing variations of architecture, input data, transmission data size.
2. **Research question:** Is it possible to reduce computational complexity without loss of performance?
Objective: Investigating common knowledge distillation techniques from the field of image classification and see if knowledge distillation is possible for sound localisation with BTE hearing aids. Testing different teacher-student methods and training schedules.
3. **Research question:** What model expansion techniques improve model accuracy the most?
Objective: Testing extra layer configurations and data streams through the model to improve accuracy on the same training dataset.

2

Literature review

The study field of source localisation is a research topic for many decades in many fields. In the audio field everything started with algorithms based on signal processing. Later on, machine learning models had it's breakthrough in exceeding the performance of the signal processing approaches, especially in various realistic environments. Hereby the used machine learning techniques and previous works will be discussed.

2.1 Human hearing limits

Every person has different boundaries in hearing. From sound source localisation to audible frequency limits. The frequency range of audibility is from 20 Hz to 20 kHz. These are the lower and upper bounds in the available measurements for most young, otologically normal listeners. For most people the highest audible frequencies are much lower. For children upper bounds of 16-18 kHz and for adults 14 kHz are more common. Humans tend to be most sensitive for frequencies in range of 1000 Hz to 2000 Hz. When certain loud sounds are present, the sound immediately after might be undetectable. This is called fatigue or masking phenomenon [12].

In the work of [13] the (absolute) minimal audible field (MAF) and minimal audible pressure (MAP) thresholds are studied. They observed that the hearing sensitivity thresholds below 1000 Hz and above 4000 Hz show a loss of 6 dB/octave and 24 dB/octave in sound pressure level (SPL) respectively.

Humans can localise sound sources very well. Although, localisation failures occur when two sources' direct path come in short after each other. This phenomenon is called the precedence effect [14].

2.2 Sound localisation

First, the main aspects in sound localisation will be reviewed, starting from the characteristics of the sound waves. These sound waves have interaction with their environment which results in new situations for sound localisation. Approaches to interact with this effect will be discussed here.

2.2.1 Physical characteristics of sound

Sound can be described as longitudinal pressure waves, whom travel through air or other media. Basic characteristics are wave length or frequency, amplitude, phase and propagation velocity. These characteristics determine the perceived pitch, loudness and direction of the sound. The regular way of working with sound signals is usual in the time domain. For further exploration of the sound signal, the frequency domain is used.

2.2.2 Short-term Fourier transform

Short-term Fourier transform (STFT) is a useful technique for getting frequency information over time[15]. The result of a STFT can be plotted in a three dimensional way. This is called a spectrogram when the amplitude is squared. Starting with a discrete sampled signal $x[n]$ with a fixed sampling rate f_s and a discrete window function $w[n]$ with length N . Here the length N of the window function determines the duration of the STFT frames, which amounts to N/f_s seconds. A trade off in resolution is made in time or frequency depending on the window size. Windows are shifted with a sample jump of L samples. The hop size L should be smaller than the window size N to have overlap. This will preserve information in the window borders depending to it's previous and following frames. The mathematical representation is given as

$$STFT : x[n] \equiv X(k, l) = \sum_{n=0}^{N-1} x[n + mL] w[n] e^{-j2\pi k/N} \quad (2.1)$$

with l is the time frame index and k the frequency bin index.

2.2.3 Room acoustics and room impulse responses

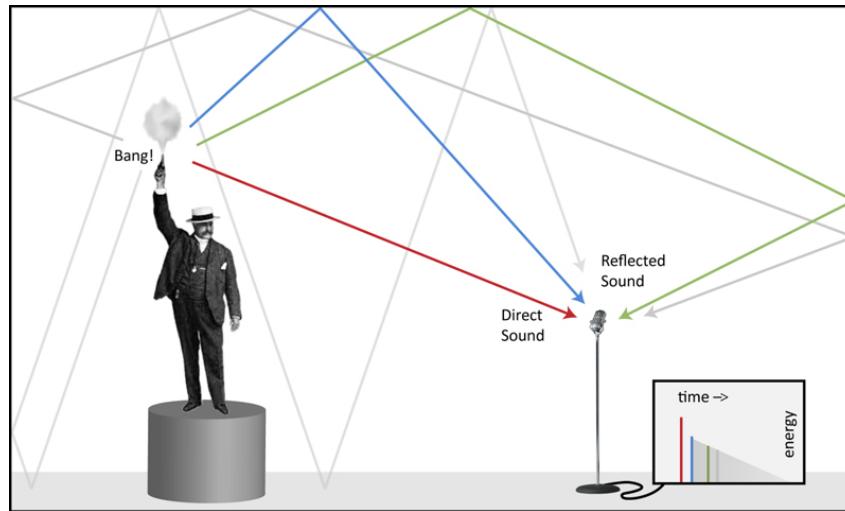


Figure 2.1: Room impulse response of gunshot (ProSoundWeb · EH Media LLC , 2022)

Room acoustic refers to the way sound behaves in an enclosed space. The nature of this behavior is influenced by the size and shape of the room, materials used in its construction and the objects present in the room. In any room, sound waves are reflected, absorbed, and diffused by walls, ceilings, floors, and objects. Hard surfaces tend to reflect sound, while

2 Literature review

softer materials like curtains or carpets absorb sound. The balance of these elements determines the room's acoustical characteristics.

The reverberation time (RT60) is the time to decay by 60dB in a closed space. The RT60 will depend on various factors like the room properties but also the conditions the room is in, like humidity and temperature.

A room impulse response (RIR) is an equation in audio signal processing to measure the acoustic characteristics of a room in response to an impulse-like sound source. Here room is seen as a system and the behaviour in time is measured. It captures how sound emitted from a source is altered by the room environment before being received at a specific point such as a microphone or a listener's ear.

A RIR consists out of three main parts. In the direct path, the sound travels directly from the source to the receiver without any reflection. Next, the early reflections come from surfaces like walls, ceiling and floor that reach shortly after the direct sound. These waves provide spatial and textural cues about the room. Lastly, there is a reverberation tail which is the decaying sound caused by numerous reflections within the room.

Image source model

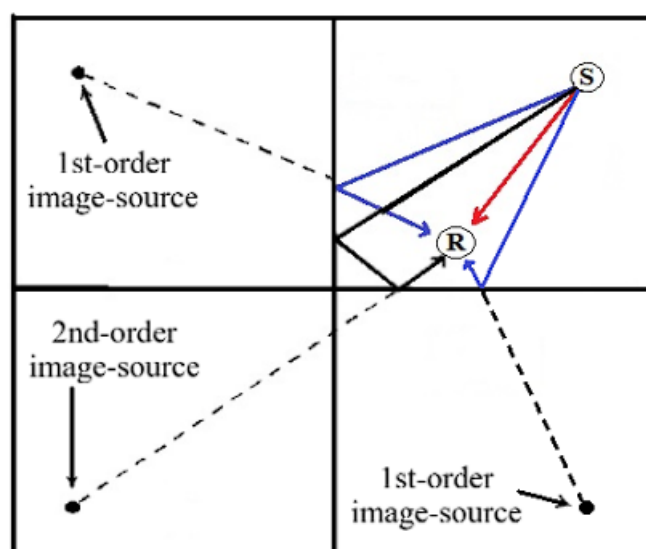


Figure 2.2: "Geometrical representation of direct sound wave (solid red line), 1st-order reflection paths (solid blue lines) and the 2nd-order reflection paths (solid black line). The letter S and R refers to a sound source and the receiver " [1] (Anjana Rathnayake, 2019)

The Image Source Model is a mathematical method used to simulate the effect of room acoustics on sound propagation [16]. It is particularly effective for predicting early reflections in a room. In order to do this, the model creates virtual sound sources, called "image sources", outside the actual room boundaries. These image sources represent reflections of the real sound source of the room surfaces. Due to computational complexity, the late dense reverberation is done using other methods like ray tracing.

2.2.4 Binaural hearing

Good hearing is crucial for everyday tasks such as understanding speech in noisy environments, identifying the location of potential dangers, speakers, and navigating through space. Binaural hearing helps to locate these sources and especially when multiple sources are present [17]. The use in hearing aids creates a more natural and immersive listening experience.

Each person has a unique way in which sounds are filtered by their head, torso, and external ears (pinnae) before reaching the eardrum. The head creates the shadow effect for the incoming sound [18]. These physiological filters add subtle cues to the sound that help in localising sources in both the horizontal and vertical planes. The head-related transfer function (HRTF) varies from person to person and is a complex function of the sound with different sound source directions.

Binaural features

The use of multiple microphones is necessary in sound source localisation for the reason that binaural features are extracted from the received signal differences like time differences of arrival. Multiple variations of array architectures can exist to maximise these differences in all sorts of sound arrival directions. The latter will not be further discussed.

Time difference of arrival (TDOA) or interaural time difference (ITD) is a crucial concept in signal processing and localisation, often employed in various fields, including acoustics and wireless communications. TDOA represents the time delay between the arrivals of a signal at multiple receivers. This feature is frequency-dependent. This is because when the frequency increases, the wavelength decreases. Following this, the time difference between the receivers will become smaller and independent from the sound source localisation. Human and animals use this feature as well[19]. Mathematically, TDOA for one frequency can be expressed as,

$$TDOA = \frac{D}{c} \quad (2.2)$$

with D the Euclidean distance between two receivers and c the propagation speed of sound. Interaural level difference (ILD) refers to the difference in intensity or level between two channels or sensors. Just like TDOA, human and animals make use of this as well. In [20] human hearing behaviour is being compared with neural network behaviour using these features and different types of dataset processing to create a natural environment.

In the works [21] [22], ITD and ILD are performing good as input features for neural networks built for binaural sound source localisation in 2D and 3D with both noisy and reverberant environments.

According to the research in [23], sound localisation depends on two binaural (ILD and ITD) and one monaural (HRTF) types of cue. A change in used cues takes place around 1500 Hz, with ITD below and ILD used above. Due to this change, the binaural cues give a 'gray area' between 1000 and 3000 Hz. Here the binaural cues are inefficient. With frequencies below 1000 Hz or wave length 0.343 m, the waves can bend around the head due to diffraction. This results in minimal ILD while higher frequencies are better covered by the shadow effect of the head and less prone to diffraction. The ITD on the other hand works best with low frequencies with waves larger than the distance between the ears. However, at higher frequencies, where wavelengths are shorter than the head, multiple phase cycles can occur between the ears. This leads to phase ambiguity, making ITD hard to impossible.

Head-related transfer function

Because hearing aids are placed on the ear, the head can block the sound and have an impact on the sound waves. In order to simulate the effects of the head, ears and torso, a head-related transfer function is used. This function is obtained by measuring the signals in an anechoic chamber with person or dummy heads wearing hearing aids. Because everyone's ears and heads properties are different, a head-related transfer function is personal [24][25].

Binaural room impulse response

A binaural room impulse response (BRIR) is a measure of how a room transforms sound from a point in space to the receivers interpretation. It captures not only the room's acoustical properties but also the spatial characteristics of sound, including direction and distance. One way to capture BRIR's, is using a mannequin with microphones placed in the ears or using specialised microphone setups. The impulse response is recorded by playing a sound (often a short burst or sweep) from a speaker in the room and recording how it arrives at the listener's position. Another method is to convolve a HRTF with a RIR.

For calculating the BRIR for every azimuth angle the following is done in the time domain.

The h_{srir}^i is calculated for every image source i with

$$h_{srir}^i(t) = \sum_{s=0}^{fdl} \left(\frac{1}{d}\right)^i \text{Hann}(s) \text{sinc}(a(t)^s) \quad (2.3)$$

d = Euclidean distance (mic, img source) fdl = length of the fractional delay filter

The distance travelled by sound is directly proportional with the propagation time. Due to possible smaller propagation times than the sample frequency supports, the fractions of samples could be taken in account as well. For reducing computing complexity a look up table is used to find a together with linear interpolation for achieving the higher accuracy of the sinc filter. A Hanning window is used to get rid of discontinuities at the filter ends which could lead towards spectral leakage.

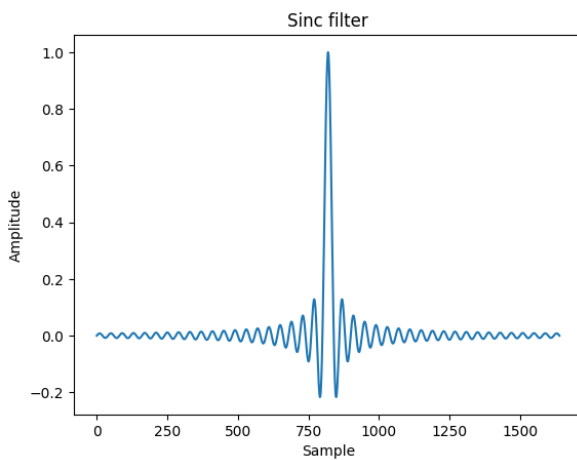


Figure 2.3: Sinc filter

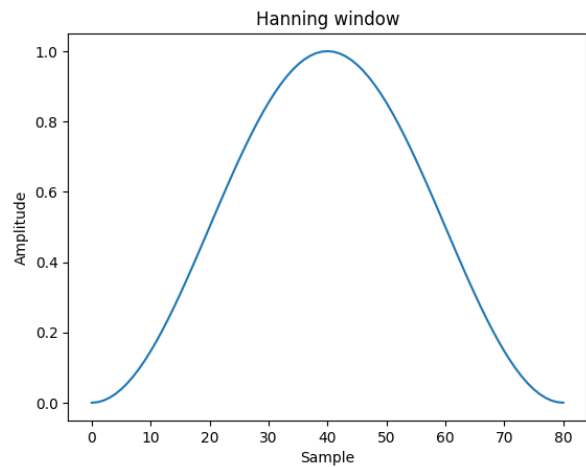


Figure 2.4: Hanning window

2 Literature review

$$\text{sinc}(n) = \frac{\sin(\pi n)}{\pi n} \quad (2.4)$$

$$w_{Hann}(n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \quad N = \text{window size} \quad (2.5)$$

The air absorption is taken into consideration using this formula:

$$a_d = e^{-\frac{1}{2}\gamma d_t} \quad d_t = t_{travelled} \cdot 343 \quad \gamma = \text{absorption factor for that frequency band} \quad (2.6)$$

The absorption factor is frequency dependent, therefore the air decay (a_d) is calculated for every octave band (table 2.1). This is more accurate than only using one absorption constant because the RIRs will be used with vocal sounds which contain multiple frequencies. After this is done, all the decays are summed per channel to achieve the total impact of the air. The process is repeated for every image source.

$$h_{brir} = \sum_{i=0}^{order} h_{sir}^i * h_{hrtf}^i \quad (2.7)$$

For every reverberated wave, the convolution with the corresponding room impulse response h_{sir} is taken with impulse response of the head-related transfer function for sound coming from that direction h_{hrtf} .

Table 2.1: Used octave bands for calculating RIRs.

octave bands	125	250	500	1000	2000	4000	8000
--------------	-----	-----	-----	------	------	------	------

2.2.5 Beamforming

Beamforming is a signal processing technique used in sound source localisation that involves manipulating the phase and amplitude of signals received by an array of microphones to create a directional focus. Multiple microphone arrays are necessary. The spacing and pattern of this array are crucial for its directional sensitivity. The TDOA is key for beamforming.

Beamforming enhances the signal from a specific direction while suppressing noise and signals from other directions. By aligning the adjusted signals, using the phase and amplitude, a zone of sensitivity towards the source direction is received.

Limitations are set when the microphones are placed closely together. The real world factors like noise and reverberation form a problem as well. For hearing devices, the physical presence of the device can create an occlusion effect, altering the natural perception of sound. Beamforming can sometimes alter or suppress spatial cues that are important for natural hearing and sound localisation. This can lead to a less natural listening experience or difficulties in perceiving the auditory scene. Lastly multiple sources are sometimes difficult to distinguish.

2.2.6 Steered response-power phase transform

The steered response-power phase transform (SRP-PHAT) algorithm is a popular one-stage method used for sound source localisation in environments with multiple microphones[26] [27] [28]. By Estimating the TDOA using generalised cross correlation (GCC), the source location can be found. For two microphones, receiving signals x_1 and x_2 with time difference τ and impulse response functions h_1 and h_2 the GCC can be written as:

$$\varphi_{x_1 x_2}^g(\tau) = E\{x_1(t) * h_1(t)(x_2(t + \tau) * h_2(t))\} \quad (2.8)$$

For identifying the location of the sound source, the point where this spectrum reaches its maximum is used.

$$\hat{\tau} = \arg \max \{\varphi_{x_1 x_2}^g(\tau)\} \quad (2.9)$$

This method can be extended to the frequency domain by using Parseval's theorem where the cross power spectral density plays a significant role to normalise the amplitude.

$$\varphi_{x_1 x_2}(\tau) = \int_{-\infty}^{\infty} H_1(f) H_2^*(f) \phi_{x_1 x_2}(f) e^{j2\pi f \tau} df \quad (2.10)$$

The algorithm combines the generalised cross-correlation method with phase transform weighting ($\phi_{x_1 x_2}$), which enhances its robustness against noise and reverberation. The PHAT weight function for normalisation can be written as follows:

$$\phi_{x_1 x_2}(f) = \frac{1}{|H_1(f) H_2^*(f)|} \quad (2.11)$$

This technique is well-regarded for its accuracy particularly in complex acoustic environments. The major drawback is the presence of multiple local maximum for the direction of arrival.

2.2.7 Multiple signal classification

The multiple signal classification (MUSIC) algorithm is a popular method for sound source localisation in array signal processing[29] [30] [28]. It estimates the direction of arrival (DOA) of multiple sources by exploiting the eigenstructure of the covariance matrix S of sensor array data. The M incoming signals from multiple sources D can be written in complex form as matrices. The MUSIC algorithm is limited to the constraint of $M > D$. X is the signals matrix with A the matrix of directional mode vectors, F the complex values corresponding to the received signals phase and amplitude and W the sensed noise.

$$X = AF + W \quad (2.12)$$

$$S \triangleq \overline{XX^*} = \overline{AFF^*A^*} + \overline{WW^*} \quad (2.13)$$

When assuming the incident signal and noise are uncorrelated, the signal can be written as following with P singular.

$$S = APA^* + \lambda_{\min} S_0 \quad \lambda_{\min} \geq 0 \quad \lambda_{\min} = \min \text{ noise variance} \quad (2.14)$$

2 Literature review

When ordered eigenvalues of S in a decreasing order, the corresponding eigenvectors of the p highest values span the signal subspace. The noise subspace consists of the remaining vectors and is perpendicular to the signal subspace.

A vector from the signal space $a(\theta)$ should be perpendicular to all the vectors of the noise subspace $\{v_i\}_{i=p+1}^M$. In order to measure the orthogonality, the Euclidean distance is taken.

$$d^2 = \|E_N^* a(\theta)\|^2 = a^*(\theta) E_N E_N^* a(\theta) = \sum_{i=p+1}^M |a^*(\theta) v_i|^2 \quad (2.15)$$

The peaks of the DOA can be found with the reciprocal of the orthogonality distance.

$$P_{MUSIC}(\theta) = \frac{1}{d^2} \quad (2.16)$$

This approach results in high-resolution DOA estimation, especially effective in scenarios with multiple sources and in the presence of noise.

2.3 Machine learning

Machine learning and especially neural networks form today's state-of-the-art algorithms for many prediction and recognition tasks. For audio, machine learning has proven many good results with different types of network architectures. When deep neural networks (DNN) are used for sound source localisation, they typically process spatial and temporal information from microphone arrays to identify the location of sound sources. However, unlike beamforming, DNNs do not explicitly compute directional gains or apply phase delays to steer the array's sensitivity. Instead, they learn to infer source locations from complex patterns in the data, which can include spatial cues like time differences of arrival, level differences between microphones but also more abstract features that are not directly related to the known physical principles of sound propagation.

2.3.1 Neural networks

Neural networks are a group of state-of-the-art machine learning architectures. The amount of nodes and layers will give a trade off between generalisation and complexity of the network. A network with fully connected layers and activation functions can be seen in fig. 2.6 Due to the activation functions (fig. 2.5) the network has the possibility to learn non linear complex problems. Training the network on data exists of two main sequences, forward and backward propagation. During the forward propagation, the predictions with the current state of the network are done. Then a loss difference is calculated between the predicted value and true value. The loss function will decide how the loss is calculated and depends on the task and requirements of the network[31]. With this loss the weights are updated for better functionality.

The mean squared error or L2 loss always gives a positive loss. As result of using the squared loss, big mistakes have more affect than small mistakes. So the model is encouraged to make more small mistakes over less big mistakes.

$$loss(x, y) = (x - y)^2 \quad (2.17)$$

2 Literature review

Cross entropy (CE) loss reflects the difference between the predicted distribution and the true distribution. This loss is mostly used in classification tasks. In eq. (2.18) N is the number of classes.

$$loss(x, y) = - \sum_{i=1}^N y \log(\hat{y}) \quad (2.18)$$

Just like the CE loss, the Kullback-Leibler (KL) divergence loss compares the distribution of the output and true values. The difference is, KL loss does not only take prediction confidence in consideration to calculate the loss.

$$loss(x, y) = y \log\left(\frac{y}{\hat{y}}\right) \quad (2.19)$$

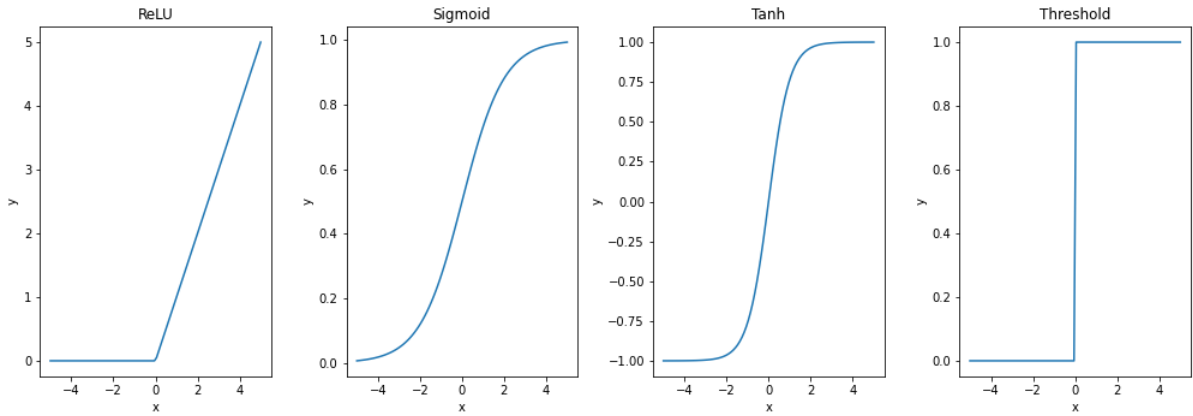


Figure 2.5: Popular activation functions used in neural networks to learn complex patterns. (Tashmit, 2020)

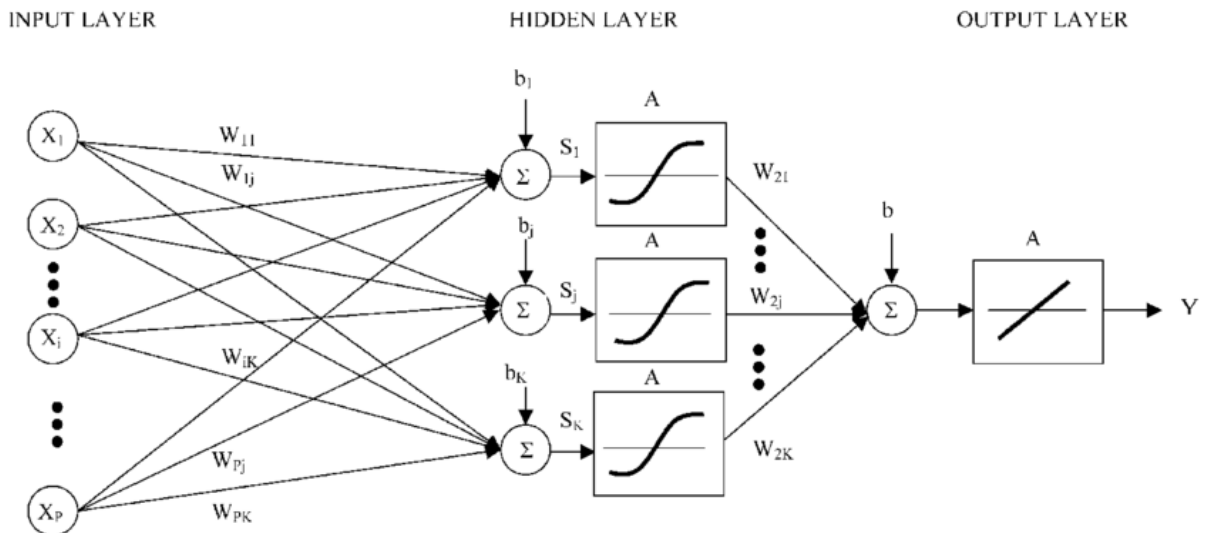


Figure 2.6: Neural network with fully connected layers. Input, hidden and output layers with activations shown. X is the input, W is the weight, b is the bias, A is the activation and Y is the predicted output. [2] (Vuckovic, 2015)

Batches

When models are trained, the entire data set is fed into the model for every epoch. This task is computationally expensive. Batches can divide this task over multiple resources. Next to more efficient resource use, batches can train the model faster by having gradient descent after every batch. By having gradient descent on different batches, the model is trained by more different data. This can give better generalisability.

2.3.2 Convolutional layers

Sound classification has undergone a transformative evolution with the introduction of convolutional neural networks (CNNs), and at the core of this revolution lies the pivotal element of convolutional layers. In this section, we embark on a journey to unravel the profound impact of convolutional layers in the domain of sound classification.

Convolutional layers are the cornerstone of CNNs, introducing a systematic approach to discern complex patterns within audio data. The convolution operation involves the application of kernels across input spectrograms, waveforms or features. These filters act as discerning agents, capturing local features and enabling the model to perceive intricate patterns at varying levels of abstraction.

As filters traverse the input data, they extract specific features, giving rise to feature maps. Each feature map represents the activation of a particular filter across the input, contributing collectively to the model's comprehension of the sound data.

Post-convolution, pooling layers come into play for downsampling the feature maps, retaining essential information while reducing computational complexity. Max pooling, a widely used technique, captures the most prominent features, further enhancing the model's efficiency.

Injecting non-linearity into the model is achieved through rectified linear units (ReLU), applied element-wise. ReLU enhances the model's capacity to learn intricate relationships within the sound data, contributing to its overall discriminative power.

In the diverse landscape of sound data, which often comprises multiple channels (e.g., stereo audio), convolutional layers are designed to process multi-channel inputs concurrently. This adaptability adds a layer of versatility, making CNNs robust in handling a wide array of datasets.

For scenarios involving spectrograms as input, 2D convolutions come into play. These convolutions capture both temporal dependencies and frequency patterns, proving especially effective in sound classification tasks where a holistic understanding of the audio data is crucial.

To tackle overfitting challenges, dropout layers are strategically introduced after convolutional layers. Dropout, a regularisation technique, randomly deactivates a fraction of activations during training, enhancing the model's generalisation capabilities.

Convolutional layers, with their innate ability to autonomously learn hierarchical representations, emerge as the linchpin in sound classification tasks. This exploration not only sheds light on the inner workings of CNNs in audio processing but also sets the stage for future advancements in the dynamic field of sound classification.

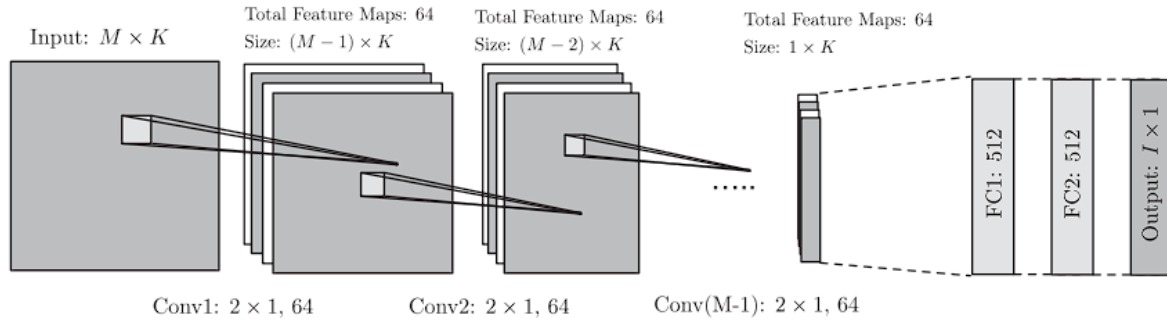
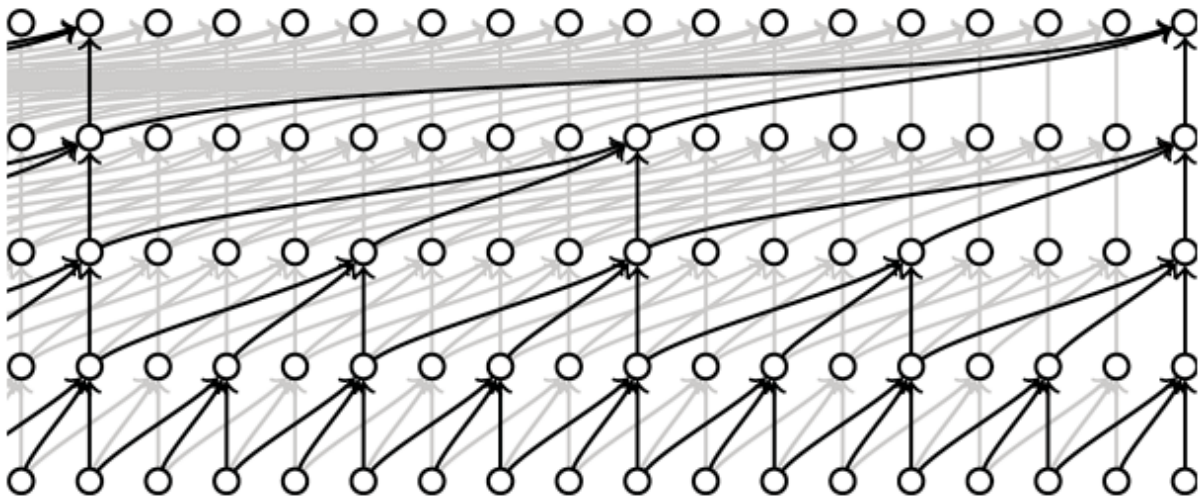


Figure 2.7: CNN architecture sound localisation [3] (Soumitro Chakrabarty , 2019)

In a supervised learning convolutional neural network for DOA estimation for multiple speakers, empirical studies [3] have shown that the feature extraction could be successful with $M-1$ convolutional layers. The given input is the phase map of the STFT for each time frame. The provided network in fig. 2.7 was able to adapt in changing environments with different acoustic setups like reverberation and noise. The network is trained on synthetically generated data and tested on both synthetic and real data. These data contained multiple active sources at the time. After comparing these results to conventional algorithms like SRP-PHAT and MUSIC, it was clear that the CNN outperformed the former methods.

2.3.3 Recurrent layers

Temporal convolutional network

Figure 2.8: "Temporal Convolutional Network comprising $L = 4$ convolution layers (from bottom to top) with exponentially increasing dilations $2^0, 2^1, \dots, 2^{L-1}$, all filters have length $K = 3$ " [4] (Alexander Bohlender , Ann Spriet , Senior Member, IEE, Wouter Tirry, and Nilesch Madhu , 2021)

A Temporal convolutional network (TCN) is a type of neural network designed for sequence modeling tasks. It uses causal convolutions to ensure that the output at any time step is influenced only by current and past inputs, maintaining the temporal order. TCNs employ dilated convolutions, which introduce gaps between filter elements, allowing the network to capture long-range dependencies without a significant increase in computational cost. Unlike recurrent neural networks (RNNs) like long-short term memories (LSTM), TCNs process the entire sequence in parallel. This results in fixed-length inputs and outputs and more efficient computation. TCNs also use residual connections, similar to those in ResNets, which help train deeper networks and reduce the vanishing gradient problem. This architecture enables TCNs to handle long-range dependencies more effectively than LSTMs and provides more stable gradients. TCNs are well-suited for tasks such as time series forecasting, natural language processing, audio signal processing, and anomaly detection in sequential data [32]. A drawback of using a TCN is the increased memory usage due to the large number of parameters required to capture long-range dependencies. This can lead to higher computational demands and complexity in designing the optimal network architecture.

Long-short term memory and gated recurrent unit

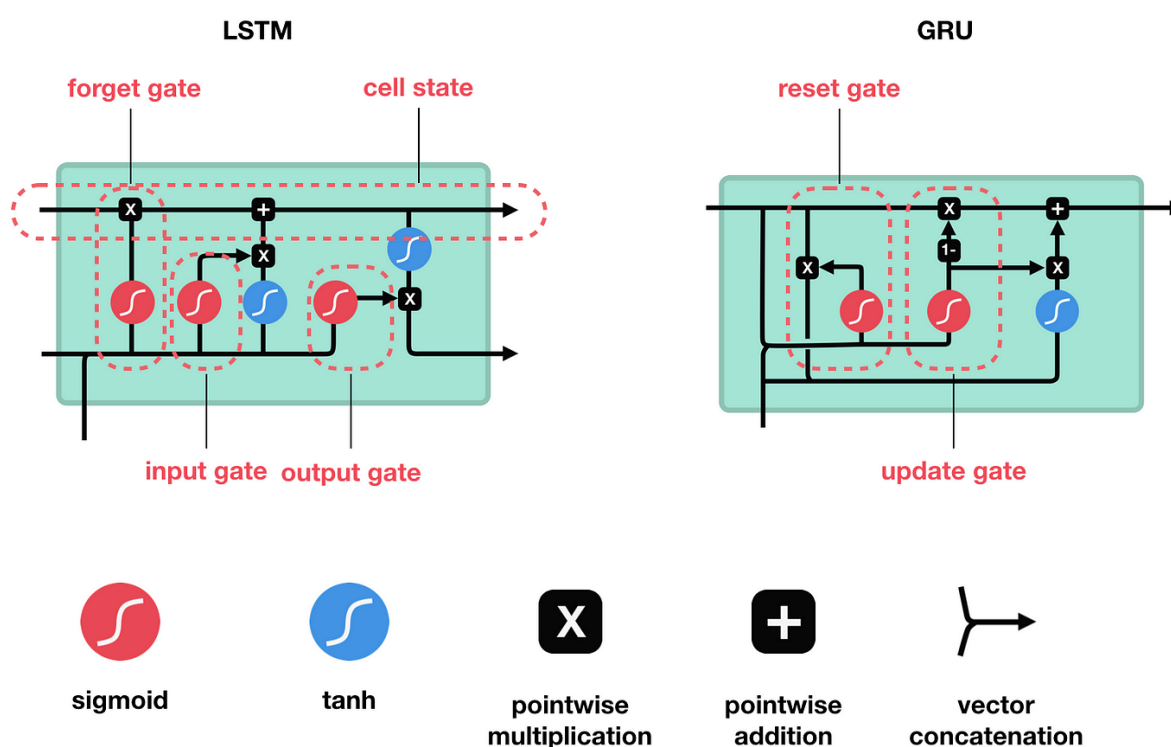


Figure 2.9: LSTM block with 4 gates and the GRU block with 3 gates [5] (Phi Michael, 2020)

The long short-term memory is used in RNN and affiliates. The network block has three gates an one memory cell. The Forget gate makes it possible to decide if previous information can be removed from the cell state. The Input gate can decide what information will be stored in the cell state. The cell state, updates the cell state with the new candidate values, after filtering through the forget gate and the input gate. The Output gate will then decide what the output is based on the cell state.

2 Literature review

The Gated Recurrent Unit (GRU) is a simplified variant of the LSTM block, differing primarily in its exclusion of a cell state and reliance solely on a hidden state. The GRU employs an update gate, which serves a function analogous to the combined roles of the forget and input gates in an LSTM. This gate determines which information from the past should be discarded and what new information should be incorporated into the hidden state. Additionally, the GRU features a reset gate that controls the extent to which previous information is forgotten. Due to its reduced number of gates, the GRU can be a more computationally efficient option compared to the LSTM. However, both the LSTM and the GRU are capable of delivering strong performance across various applications [33][34]. The choice between them may depend on the specific requirements and characteristics of the task at hand.

Markov model

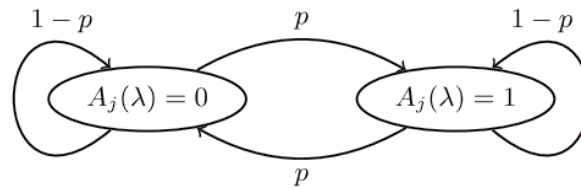


Figure 2.10: "The Markov chain with parameter p is used to model the activity of the source with index j : $A_j(\lambda)=1$ represents activity whereas $A_j(\lambda)=0$ represents inactivity"[4]. (Alexander Bohlender, 2021

During training, multiple speakers can be located up to two at the time. The decision is made by a simple Markov model that decides whether a source is active. These signals are added together in the frequency domain.

Convolutional recurrent neural network

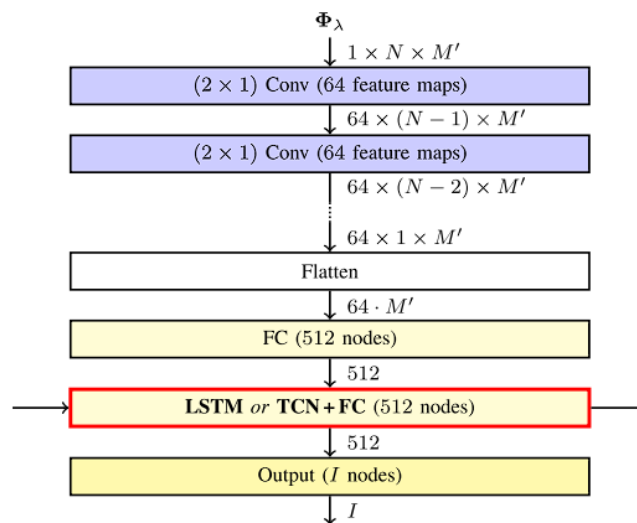


Figure 2.11: CRNN architecture sound localisation [4] (Alexander Bohlender , Ann Spriet, Wouter Tirry, Nilesh Madhu 2021)

2 Literature review

Further improvements on [3] were done by using the temporal context of the signals[4]. The training data was generated with an extension of a Markov model to simulate moving sources. Instead of putting in the input by frame, a set of frames are given to have the effect of moving sources while training the network. The movement of the sources is instant and random. The long-term temporal context, longer than previous mili-seconds, was created by adding a LSTM after the last Fully Connected (FC) layer or by adding a temporal convolutional network between the two last FC layers. Both methods were superior to older approaches.

Tracking gradually moving sources is explored in [35]. Adding gradually moving speakers into the training has improved the tracking of sources in the tests. Additionally, a latency controlled-bidirectional LSTM (LC-BLSTM) was implemented. This could improve the error in the transition between speakers onsets and offsets. A drawback of LC-BLSTM is a higher chance of tracking noise as a speaker.

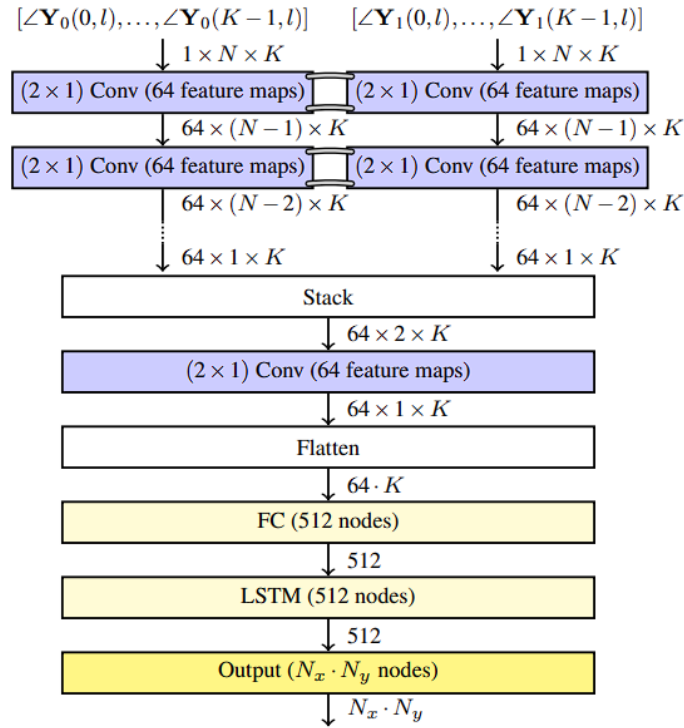


Figure 2.12: The narrowband mixing co-operative localisation architecture (NM-CLA). [6] (Stijn Kindt, 2021)

Sharing data from multiple arrays to a central device is used in order to improve the sound localisation. For 2D localisation, narrow- and broadband sharing networks were introduced to overcome asynchronous comparisons which gave problems in triangulations of fig. 2.11. The narrowband model in fig. 2.12 shares its data to the main device for every frequency band separately after the intra-array convolutions. On the other hand the broadband model in fig. 2.14 shares the data after a fully connected layer where the frequencies merge together to 512 features. So in the latter, only 512 features have to be sent instead of $64 \cdot K$ with K the amount of frequency bins. Therefor the broadband model requires less bandwidth to send data. By sharing the data in a later stage of the network, the problem of loading asynchronous had been overcome.

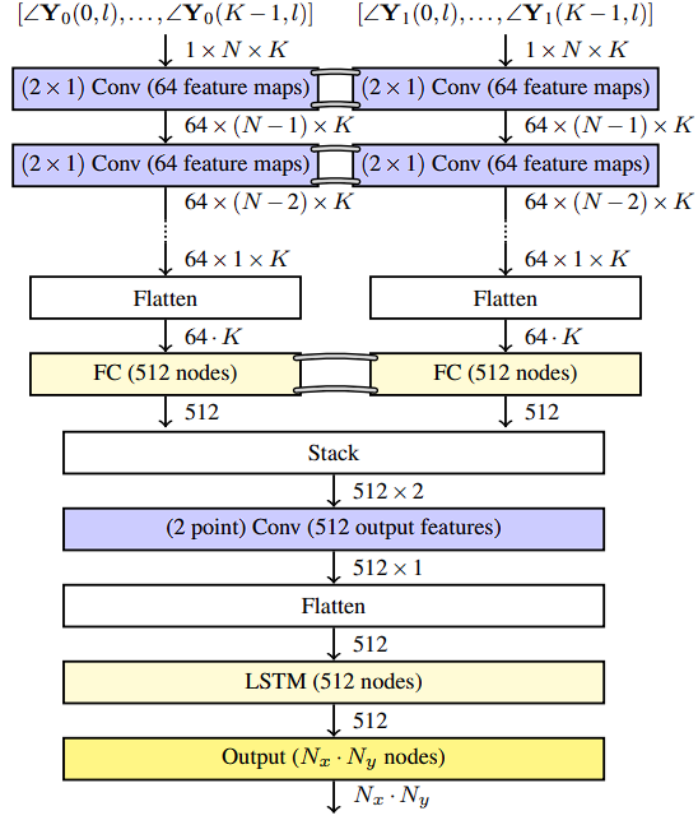


Figure 2.13: the broadband mixing co-operative localisation architecture (BM-CLA). H . [6] (Alexander Bohlender , 2021)

2.3.4 Codec

For systems on end user devices, the network should be as simple as possible. This is done in order to ease the calculations and makes it possible to use less computing power. This will result in cheaper more sustainable hardware. To further improve and adapt the model for its end user, a codec could be used when data have to be transferred. Lossy compression is used mostly in the audio field. This is an irreversible degradation of the signal to minimise the signal size in terms of bits.

The LC3+ codec is implemented in [7]. The model in fig. 2.11 is used as baseline. Adjustments are made for training on binaural data with six channels. In this model, the microphone data from the arrays is shared before going through the convolutional recurrent neural network (CRNN). The data will contain sampled sequential audio data. In the application of this model for BTE hearing aids, the data are sent through Bluetooth to a centralised processing unit. Sharing the audio to one of the on-ear devices is tested as well. Now only half of the microphone data should be sent through. To reduce power consumption for the data transfer the LC3+ codec is tested for 16, 32, and 64 kbps bit rate.

When applying the codec on a model trained with encoded signals, no degradation in accuracy was obtained. Even a better generalisation in noisy conditions was achieved, resulting in a performance increase. The models trained on encoded data are robust for all the provided bit rates.

The Lyra codec is a machine learning based audio codec designed for real-time speech. The signal gets simplified with

2 Literature review

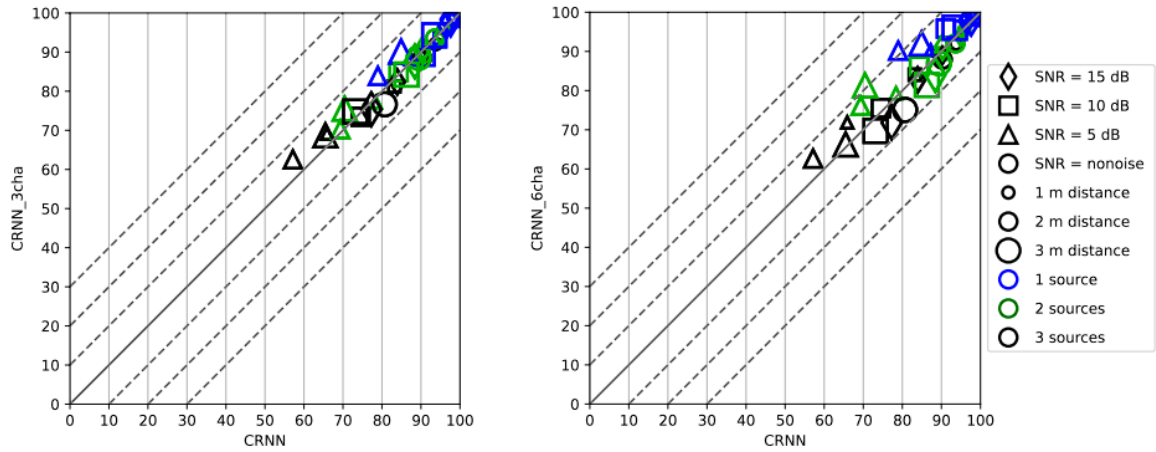


Figure 2.14: The localisation accuracies(%) of the base model on the x-axis and the encoded model on the y-axis. The encoded models are trained with the corresponding codec bit rate of 16 kbps.[7] (Siyuan Song , 2023)

extracted features and is reconstructed with a generative network. A simple model with parallel computing of different frequency ranges gives similar performance like traditional codecs in terms of computations. The codec model is trained on thousands of hours of speech over 70 languages with several speakers. This codec can run on the bitrates 3.2, 6 and 9 kbps. In a comparative study[8] between LC3+ and Lyra, the generalisation performance of the baseline (fig. 4.1) is tested. Networks were tested with encoded input on models trained with normal training data and with encoded training data. The codec appears to adversely affect performance when training is not conducted on it. On the situation with only encoding three channels, the model seems to have problems on challenging scenarios with multiple speakers and low signal-to-noise ratios (SNR). The mismatch of information transfer might be found in the convolutions with the inter-ear information.

The accuracy performance of the LC3+ in the tested scenarios is overall better than the Lyra.

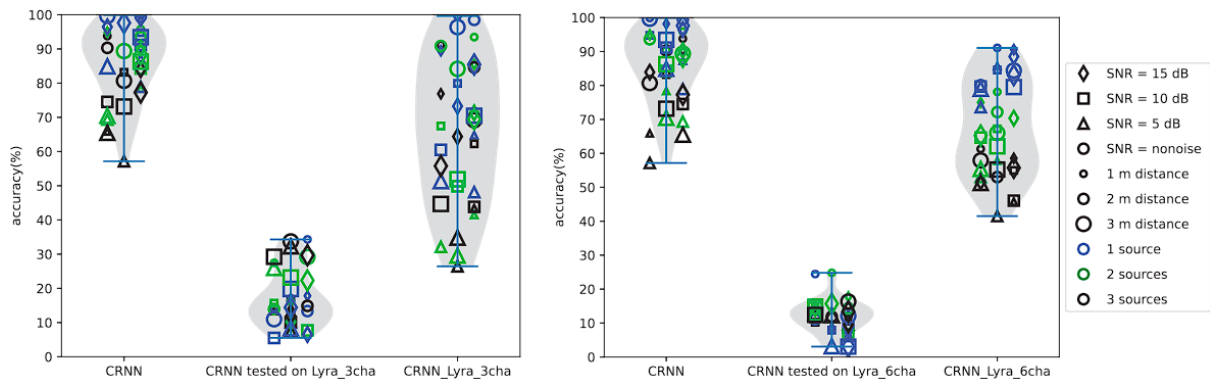


Figure 2.15: "Localisation accuracies (%) over different evaluation conditions. The left subplot presents results for 3-channel audio transfer and the right subplot for 6-channel audio transfer. Each subplot contains the baseline result, the result obtained when the baseline CRNN is fed codec-processed audio and, lastly, the results of the CRNN, re-trained with the codec-processed audio and tested in matched conditions" [8] (S. Song, S.Kindt, J. Maes, A. Bohlender, N. Madhu , 2021)

2.4 Network compression

2.4.1 Pruning

Network pruning is used to make the network faster and smaller. Weights or connections in the network are set to zero to achieve simplification. This technique can be divided into two main branches. Structured pruning and unstructured pruning. For both techniques a weight parameter is used to target parameters that can be ignored [10].

Unstructured pruning

Individual connections are the target of unstructured pruning. This has the advantage that it can effect on small parts of the network whereby it can be used in a greater scale. The drawback of this is that most hardware is not affected by the pruning.

Structured pruning

For structured pruning, entire, elements, channels or filters can be ignored. This results in lower computations. Therefore the needed bandwidth and latency will decrease.

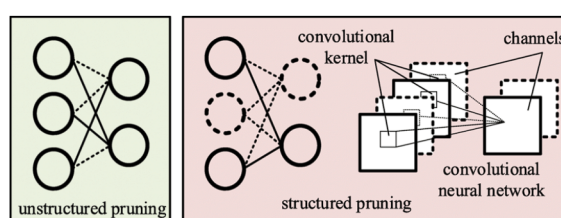


Figure 2.16: Unstructured and structured pruning for deep neural networks and convolutional neural networks. [9] (Chen Qi, 2021)

Selection criteria

In the first criterion, the absolute magnitude will decide whether the link has to be pruned. The idea roots from the idea that lower weights have less contribution and will eventually lower in magnitude in further training.

The magnitude from the gradient can be seen as another criterion [36]. During backpropagation the loss can give a cue for the importance of the weights.

Pruning can be classified in two other groups. In global pruning the amount of pruning is spread over the entire network for the given factor. While in local pruning, each part of the network has a separate pruning. Global pruning has shown to perform better. A drawback of global pruning can be that the layer dimensions won't fit anymore.

2.4.2 Knowledge distillation

Knowledge distillation in deep learning is used for many purposes. A first use is for compressing a DNN model with maintaining the accuracy as high as possible. A second use example is when a given model have to be adjusted but the data are

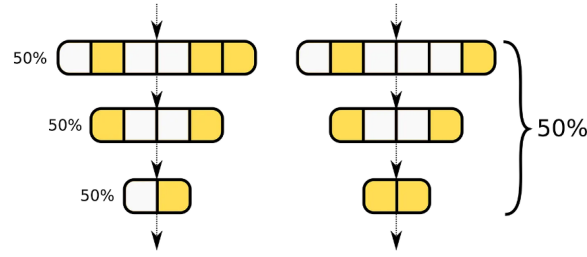


Figure 2.17: Local pruning on the left and global pruning on the right for deep neural networks. [10] (Hugo Tessier, 2021)

not available, this due to privacy or costs. In this thesis only the former will be discussed. The technique is known in various types of problems. Most known tasks are visual [37][38], LLM and speech problems. In these topics, predictions like: image classification, face recognition, neural machine translation, text generation, question answering, speech recognition, audio classification, acoustic event detection and many more have models that have obtained great knowledge distillation results.

The basic idea of knowledge distillation says that a student, the smaller model, will learn from the teacher, the bigger model.

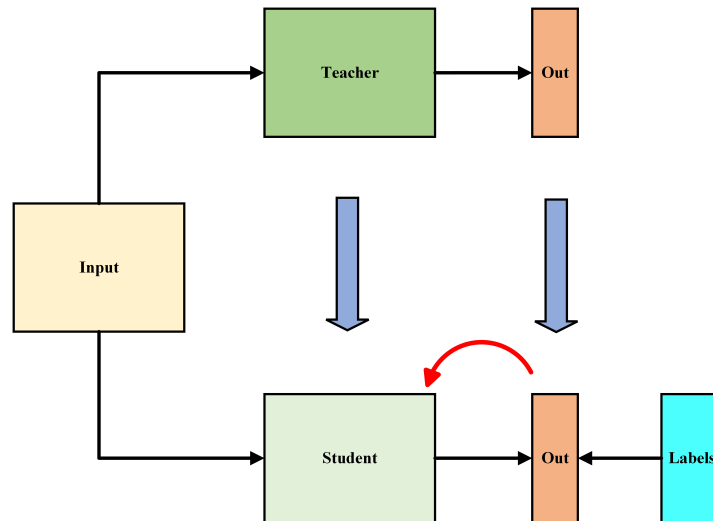


Figure 2.18: Knowledge distillation with inheriting knowledge from teacher to student, only updating the student network.

Many tested methods for knowledge distillation are making use of a custom loss function. In [39] a summary of the most used ones is given.

Response-based knowledge

A first often used variant is the response-based knowledge, this method assumes that the student can better understand the problem when he is corrected not only by the true label but also by the predicted outcome from the teacher. This would lead the student in the right direction.

The Kullback-Leibler (KL) divergence loss works good for adding predictions comparison to the loss function[40][41], [42] [39].

Feature-based knowledge

A variant of the response-based knowledge is the feature-based knowledge distillation. Here the same principle is used in order to learn how and what features should be extracted from the given data. For the response-based KD, the classes are the same and so are the dimensions. In case of feature-base KD, the dimensions of the student model might be different from the teachers. The first solution to overcome this problem is average pooling after the convolutions on the flatten layer. More complex solutions involve using a fully connected or convolutional layer to align the dimensions.

Relation-based knowledge

The goal of combining the previous two techniques is to further imitate the teacher. A normal approach of training a network is by using the loss between label and prediction in order to adjust the network weights. By imitating the teacher, the student tries to learn how the teacher deals with the problem. Here the correlation between the feature map differences and states are taken into consideration.

Attention-based knowledge

As for most papers in KD, this [43] paper for image classification shows a novel approach for online knowledge distillation, leveraging peer-teaching student networks. By utilising softened outputs from a teacher network to guide smaller student models, they demonstrate significant performance enhancements. Their introduction of the temperature parameter T to soften the softmax function adds an innovative dimension to the distillation process, improving its efficacy. Unlike traditional KD methods that rely solely on cross-entropy loss for aligning student and teacher logits, this method incorporates mutual learning and ensemble techniques to further optimise knowledge transfer. Moreover, distilling and transferring attentive maps between peers at multiple layers showcases new approach of feature extraction and representation. Overall, this approach presents a promising avenue for improving the efficiency and efficacy of online knowledge distillation in image classification tasks.

2.4.3 Generative adversarial network

Certain KD layouts can be seen as generative adversarial network layouts (GAN). When GAN is used for image generation, an example configuration can be seen in fig. 2.19. Here the generator will be the network that is learned to make the images. The discriminator will be learned to say if the generator did his job well with making the expected image. For training, generator and discriminator are altered for training after an ahead specified amount of epochs. So the training happens almost totally separate. Except for the generator where the forward pass goes through the discriminator.

In GAN-based knowledge distillation, the student and teacher networks act as generators. The discriminator classifies feature maps based on their originating network. By adding the discriminator's inverse loss to the generators' training, the generators aim to fool the discriminator by mimicking each other's feature map distribution. This adversarial approach enables a network to learn the overall feature map distribution of its peer, rather than just aligning features directly. Training

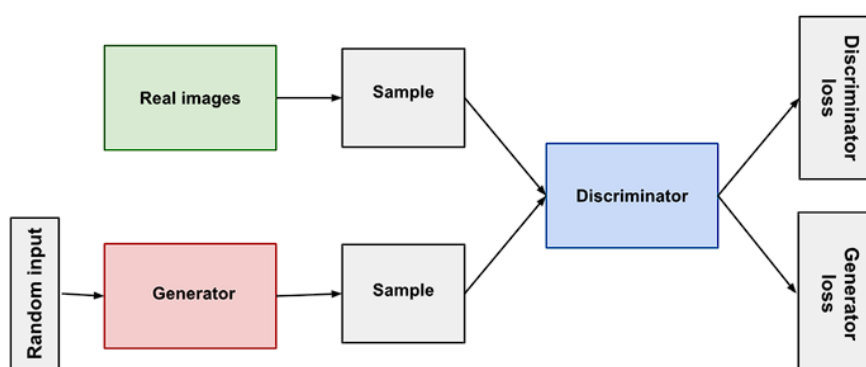


Figure 2.19: GAN with trainable generator and discriminator, configuration used to train image generators. (Google, 2022)

a discriminator to distinguish between feature map distributions, and having the networks learn to fool it, enhances each network's ability to capture the shared feature map distribution. This exchange of feature map distributions helps the networks converge to a better feature map manifold, which improves generalisation and accuracy. Additionally, this method is well-suited for secure federated learning environments, as it abstracts the specific origins of feature maps, thus protecting the privacy of individual data sources. [44]

3

Materials

3.1 Software

The proposed solutions are being developed in the Python programming language. For implementing the machine learning methods, the framework provided by Pytorch is used. For the purpose of optimising model adjustment and training to achieve optimal compression, the code bases from [7] [6] were available.

Compute unified device architecture (CUDA) enables parallel processing by breaking down tasks into smaller threads that can be executed concurrently on the GPU. This is particularly beneficial for computationally intensive tasks. CUDA is integrated in the Pytorch framework, therefore training the deep neural networks speed up significantly.

Docker containers are used to provide access to Imec's servers. The containers provide separation from the underlying system to overcome compatibility issues. Computing and memory resources can be selected by the start of the runtime. In these containers, Pytorch images provide the necessary dependencies and packages to encapsulate the entire run environment. The containers makes training possible at any time up to 10 containers at the time.

3.2 Hardware

The used hardware for training, generating and developing the models and data, exist of CPU and GPU from cloud servers provided by Imec. The tested methods were trained on one or two GPUs of the type NVIDIA GeForce GTX 1080 Ti or NVIDIA A40. The CPU memory required to load the data was 26 GB on average. On low demand of the cluster 8, a NVIDIA A40 was available to speed up training.

3.3 Data

The data for training and testing the proposed solutions is the same as in [7]. Speaker voices from the TIMIT and PTDB-TUG dataset are being used for the training and validation set. TSP speech is used for the test set. All the speech contains different sentences read by different native English speakers with different dialects. A broad group of English voice sounds are covered by using this training data. This is important for making the model able to generalise independent on who the speaker is.

3 Materials

The Oldenburg dataset is used for the HRTF [45]. The resolution of the HRTF is dependent on the measurement setup. The used HRTF for the BTE device with a microphone configuration is similar as shown in fig. 3.1.

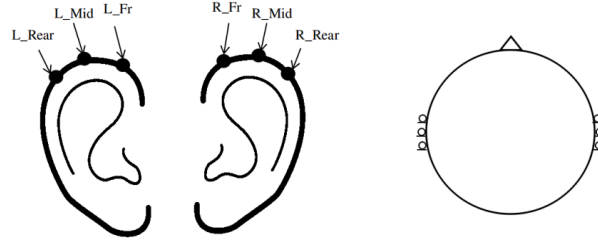


Figure 3.1: The positioning of two BTE arrays at each ear and channel index of each microphone (Siyuan Song, 2023)[7], the geometry is the same as in [11].

The equations have been measured for the following speaker configuration.

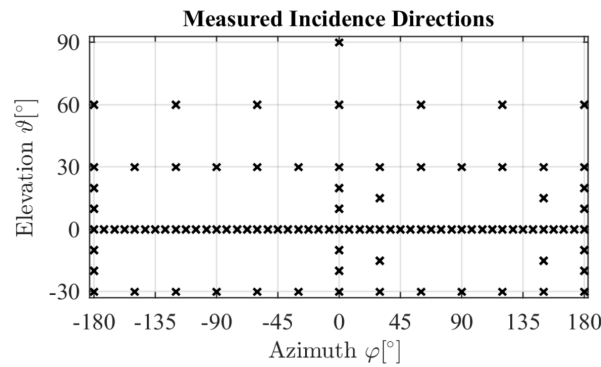


Figure 3.2: Speaker directions for the Oldenburg test setup.

In the azimuth plane, every five degrees a measuring is done. This results in 72 positions in this plane. Also different positions are measured in the vertical planes but these are not important for our azimuth estimation.

For the RT60 value a range from 0.2 to 8 seconds have been chosen for the audio absorption factor in room acoustics. These are values of common rooms of the sizes in the training set. For the validation and test set, values in between the values of the training set are chose.

Additionally, several instances of the receivers' location are used. By changing the position parameters in the training data, the model will learn to generalise the problem over all the positions. New instances in unseen data should be estimated in a good way because of the generalisation. In every room, seven random azimuth positions are taken. For all of these azimuth positions, four different distances between the array and the wall are taken. The relative distance between the array and wall is chosen to overcome impossible situations of the array with standing out of the room.

Lastly, the SNR will be randomly chosen between 0 and 30 dB. This gives the ability to adjust the model parameters to ignore the noise in the incoming signals at different SNRs.

Table 3.1: "Data generation: training, validation and test set" (Siyuan Song , 2023)

Training set	Speech signals	7438 utterances from TIMIT and PTDB-TUG dataset
	Room dimensions (m)	R1: $(6 \times 6 \times 2.5)$, R2: $(5 \times 4 \times 2.8)$, R3: $(10 \times 6 \times 2.4)$, R4: $(8 \times 3 \times 3.1)$, R5: $(8 \times 5 \times 2.9)$, R6: $(4 \times 9 \times 3.3)$, R7: $(7 \times 7 \times 2.3)$, R8: $(5 \times 6 \times 3.6)$, R9: $(9 \times 6 \times 3.2)$, R10: $(11 \times 7 \times 3)$
	T60	R1: 0.3 s, R2: 0.2 s, R3: 0.8 s, R4: 0.4 s, R5: 0.6 s, R6: 0.5 s, R7: 0.7 s, R8: 0.45 s, R9: 0.55 s, R10: 0.75 s
	Array positions	7 different positions in each of the rooms
	Source-array distances	20%, 40%, 60% and 80% of the distance between array and wall
Validation set	Additive noise	Simulated diffuse noises with SNRs from 0 to 30 dB
	Speech signals	2280 utterances TIMIT and PTDB-TUG dataset
	Room dimensions(m)	R11: $(5.5 \times 7.5 \times 2.7)$, R12: $(8.5 \times 4.5 \times 3.5)$), R13: $(6.5 \times 6.5 \times 2.3)$
	T60	R11: 0.525 s, R12: 0.625 s, R13: 0.475 s
	Array positions	4 different positions in each of the rooms
Test set	Source-array distances	30%, 50%, 70% and 85% of the distance between array and wall
	Additive noise	Simulated diffuse noises with SNRs from 0 to 30 dB
	Speech signals	1444 utterances from TSP speech database
	Room dimensions & T60	R14: $(5 \times 4 \times 2.5)$ m, 0.2 s; R15: $(7 \times 6 \times 3.5)$, 0.6 s
	Array positions	4 different positions in each room
Test set	Source-array distances	1m, 2m, 3m (Only with the angles inside the room)
	Additive noise	Simulated diffuse noises with SNRs of [5, 10 ,15] dB or no noise

4

Foundational model structures

4.1 Signal model

In this section incoming signals and input for the sound localisation model are described. Unless otherwise specified, these conventions will apply to all the models tested in the thesis. Starting with the incoming sound signals on all the M microphones where the m th microphone signal at ear e is denoted as $y_{e,m}(n)$. This is a combination of the contributions of J sources $x_{e,m,j}(n)$ and additive noise $v_{e,m}(n)$:

$$y_{e,m}(n) = \sum_{j=0}^{J-1} x_{e,m,j}(n) + v_{e,m}(n) \quad (4.1)$$

where n is the sample index for a signal sampled at f_s Hz. All networks operate on the short-time Fourier transform representation of the microphone signals, which will be represented as $Y(k, l)$, with k the frequency bin index and l the time frame index. This is computed with a square root Von Hann window, frame size K , hop length L and number of frames N . This representation is often chosen for speaker localisation and separation because of the sparsity and disjointness properties of speech [15] in the STFT domain. The input features, extracted from the full bandwidth sequential data, consist of the sine and cosine of the STFT, and the normalised STFT amplitude:

$$F_{e,m}(k, l) = \begin{bmatrix} \sin(\angle Y_{e,m}(k, l)) \\ \cos(\angle Y_{e,m}(k, l)) \\ |\overline{Y_{e,m}}(k, l)| \end{bmatrix} \quad (4.2)$$

with $\angle Y_{e,m}$ the phase of $Y_{e,m}$ and $\overline{Y_{e,m}}$ the normalised amplitude.

Training

During training, the number of frames N to process will be equal to 200. This decision is rooted from the fact that the model is trained on two-second time sequences. The size of the number of frames is important to give the temporal layer enough time to effectively learn temporal behaviour during the forward pass before the backpropagation. Secondly, it optimises GPU utilisation by facilitating parallel computation during training, thereby maximising computational throughput.

4.1.1 Amplitude normalisation

For the centralised model, all the signals are available and averages can be taken over any dimension. Therefore normalisation is taken with an average over the channels (N_c) with normalisation factors for every frequency and time bin.

$$\overline{Y_{e,m}} = |Y_{e,m}| / \left(\frac{1}{M} \sum_{\nu=1}^M |Y_{e,\nu}| \right) \quad (4.3)$$

This differs for the BM-CLA model, where inter-array signals are not accessible. For the concept of proof for the adjustments in the BM-CLA model, with focus on broadband embedded feature sharing, the same normalisation as in the centralised model from eq. (4.3) is used. Later in chapter 5 a method for lower bandwidth requirements is suggested.

4.2 Centralised model

For this model, the microphone signals are first sent to the central processing unit. This could either be done by transmitting the full bandwidth signals or by using an audio codec. As was shown in [8], the choice of codec is important. Using the lossy LC3+ encoded signal during training resulted in the same (or sometimes even slightly higher) localisation accuracy than using the full bandwidth signals. This is why the LC3+ codec, a standard codec for Bluetooth low energy, is chosen for comparison.

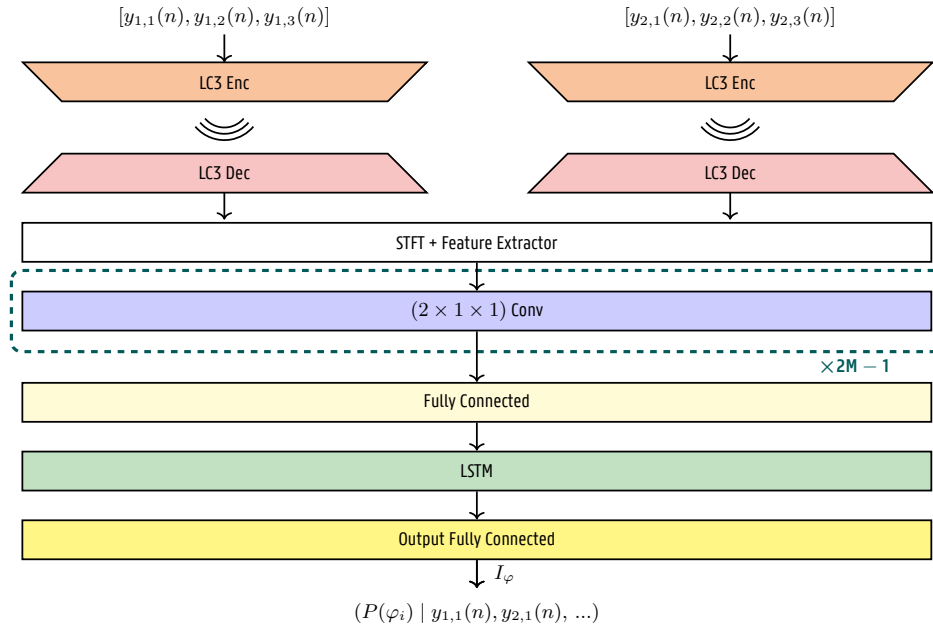


Figure 4.1: The centralised localisation algorithm. Initially, the microphone signals are acquired from the LC3+ codec, followed by the estimation of the direction of arrival (DOA) for the incoming sound sources.

At the central unit, the features are stacked along a microphone-channel dimension in such a way that the microphones at different ears are interleaved: $\mathbf{F}(k, l) = [F_{1,1}(k, l), F_{2,1}(k, l), F_{1,2}(k, l), \dots]$. Preliminary evaluations showed this was optimal configuration for the subsequent convolution blocks as this leads to successive convolutions with microphones

4 Foundational model structures

on opposite ears. The $M - 1$ convolutional blocks (**Conv**) each have a kernel size of two along the microphone channel dimension, and iteratively combine the microphones of the binaural array. Note that at this stage each time-frequency point is still processed independently.

The fully connected (**FC**) layer combines the narrowband frequency content independently for each time frame. The **LSTM** layer then incorporates temporal context in a causal manner. A final **Output FC** is used to go from the latent features towards an classifier, where each class represents a DOA sector. The DOA resolution is decided by the number of DOA classes I_φ – more classes make for a finer estimate, but will require more training. For the multi-source case ($J > 1$), the choice of a classification-based localisation was shown to outperform regression based outputs [46]. Henceforth, the Baseline model will be identified as the model shown in fig. 4.1, with parameters specified in table 4.1, omitting the LC3+ codec.

4.3 Co-operative model

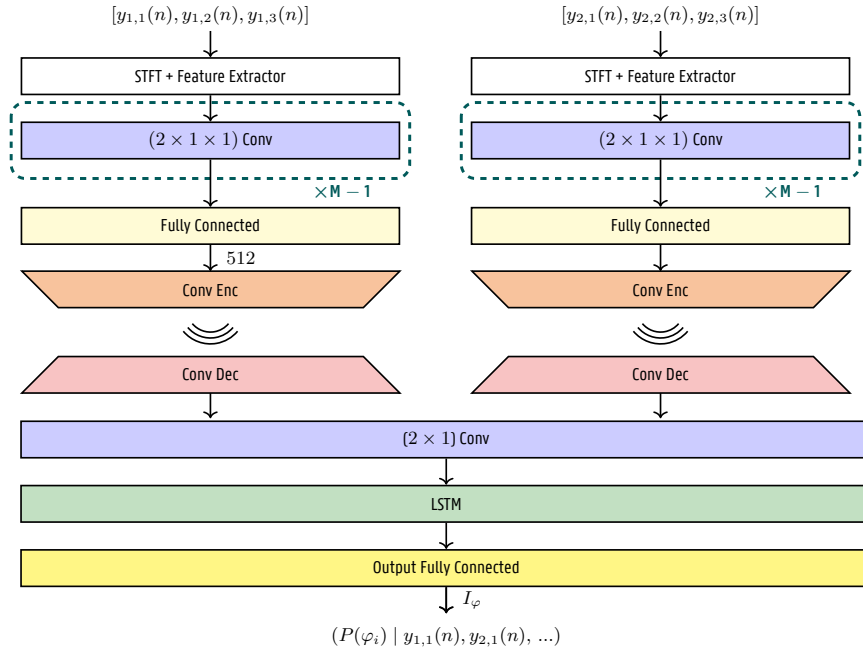


Figure 4.2: The broadband mixing co-operative localisation architecture (BM-CLA). Here, the mixing of information occurs after the FC layer, and is also performed by an inter-array convolution.

The centralised localisation model works well but relies on audio codecs to reduce the bandwidth. The LC3+ codec (a waveform codec) could lower the data throughput to 16kbps per microphone channel and, while lossy, still did not impact localisation performance when it was included in the training [7]. Going to even lower bitrates often requires neural network based generative models. But these are optimised for single-channel data transmission and do not retain spatial information effectively [8]. Another improvement could be provided by extracting latent features at each ear independently and sending these to the central processor for a final DOA estimate. Then the challenge of low bandwidth is translated into compacting the latent feature vector in an appropriate manner.

4 Foundational model structures

Such a feature sharing model is depicted in fig. 4.2, and is based on the BM-CLA [6]. In the original paper, the goal was to exchange information between two distributed microphone arrays and improve the localisation accuracy by sharing latent features. However, in this context, the architecture is repurposed to send low bitrate latent features to the central processor. This architecture shares the same underlying architecture as the centralised model, which also makes them suitable for a fair comparison.

Instead of sending the data directly to the central processor, the same features as in eq. (4.2) are independently computed in each earpiece individually. These are again combined through $M_e - 1$ a convolutional layers (**Conv**), following which the narrowband features are combined to yield broadband information using the fully connected layer (**FC**). This yields the latent features ($\mathcal{F}$) to be transmit to the central processor, as it was already shown in [6] that sharing broadband information was more beneficial than narrowband information. From this point forward, the BM-CLA reference model $\mathcal{F}512$ will be referred to as the model depicted in fig. 4.2, with parameters detailed in table 4.2, excluding the **Conv Enc** and **Conv Dec** components. If stride S is mentioned, the encoder-decoder is used with the corresponding stride. These convolutions use groups with equal size of the input to only interchange information per channel, and so reduce complexity (eq. (5.3)).

4.3.1 Network layer equations

In order to do compression, a good understanding of the used formulas is required. The structure of the formulas is essential for later calculations of the parameter count and MACs. In the formulas used, batch normalisation and additional activation functions are not taken into account because of lower impact on the previous mentioned metrics. Batch size will be chosen equal to one.

Convolutional layer (Conv)

$$Y(C_{out}) = B(C_{out}) + \sum_{k=0}^{C_{in}-1} A(C_{out_j}, k) * X(k) \quad (4.4)$$

$$H_{out} = \frac{H_{in} + 2K_s[0] - 1}{S[0]} + 1 \quad W_{out} = \frac{H_{in} + 2K_s[1] - 1}{S[1]} + 1 \quad (4.5)$$

In this equation, padding and dilation are set to 1. Here the input matrix is $X_{C_{in} \times H_{in} \times W_{in}}$, the output matrix is $Y_{C_{out} \times H_{out} \times W_{out}}$ and the weight matrix A has the same dimensions as K_s . The C denotes the number of frames (N) in the used model. H stands for the size of the first dimension, this the number of mics M_e per ear in the intra-ear convolutions. W stands for the width in the second dimension and is the amount of frequencies (K) per frame in the BM-CLA model. The formula computes a weighted sum of each input elements across all input channels using the convolutional kernels. This is done for each corresponding element in the output feature map. The bias term is then added to this sum to produce the final output value. This operation is repeated for each element in the output feature map to generate the entire layer output [47].

Fully connected layer (FC)

$$Y = AX + B \quad (4.6)$$

The number of input and output nodes are seen as N_{in} and N_{out} . In the equation, $Y_{N_{out} \times 1}$ represents the output matrix, $A_{N_{out} \times N_{in}}$ denotes the weight matrix, $X_{N_{in} \times 1}$ stands for the input matrix, and $B_{N_{out} \times 1}$ signifies the bias matrix. For

4 Foundational model structures

the case of the used models, the number of frames (N) dimension should be reshaped to the batch dimension. In this equation, the weight matrix is multiplied by the input matrix. The resulting product is then augmented by the bias term B . The output matrix represents the result of this linear transformation, where each element of Y is computed as a weighted sum of the elements in X , adjusted by the bias term B [48].

Long-short term memory (LSTM)

$$I_t = \sigma(W_{ii}X_t + B_{ii} + W_{Hi}H_{t-1} + B_{Hi}) \quad (4.7)$$

$$F_t = \sigma(W_{if}X_t + B_{if} + W_{Hf}H_{t-1} + B_{Hf}) \quad (4.8)$$

$$G_t = \tanh(W_{ig}X_t + B_{ig} + W_{Hg}H_{t-1} + B_{Hg}) \quad (4.9)$$

$$O_t = \sigma(W_{io}X_t + B_{io} + W_{Ho}H_{t-1} + B_{Ho}) \quad (4.10)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot G_t \quad (4.11)$$

$$H_t = O_t \odot \tanh(C_t) \quad (4.12)$$

Similar as for the fully connected layer, the number of input and output nodes are seen as N_{in} and N_{out} . In the equations, $I_{t \times N_{out} \times 1}$ is the output for the input gate and $X_{t \times N_{in} \times 1}$ the input at step t . Here, $W_{ii \times N_{out} \times N_{in}}$ and $B_{ii \times N_{out} \times 1}$ are the matrix weight and bias associated with the input gate. The parameters $W_{hi \times N_{out} \times N_{out}}$ and $B_{hi \times N_{out} \times 1}$ are the matrix weight and bias associated with the previous hidden state H_{t-1} . The σ denotes the sigmoid activation function and $\tanh$, the hyperbolic tangent function. For the forget gate $F_{t \times N_{out} \times 1}$, the cell state gate $G_{t \times N_{out} \times 1}$ and the output gate $O_{t \times N_{out} \times 1}$, the same name conventions for the weights and biases are used. In the usecase of the proposed models, the number of frames N dimension should be reshaped to the batch dimension. The input gate, determines the extent to which new information should be stored in the cell state. Next, the forget gate computes the degree to which each component of the cell state should be forgotten. Then the candidate cell state or cell gate, computes the new candidate values for the cell state. The update cell $C_{t \times N_{out} \times 1}$, updates the cell state by combining the forget gate, input gate and candidate cell state. Near the end, the output gate determines which parts of the cell state should be the output. Lastly the output in state t ($H_{t \times N_{out} \times 1}$) is calculated. A visualisation of the workflow can be seen in fig. 2.9.

Gated recurrent unit (GRU)

$$R_t = \sigma(W_{ir}X_t + B_{ir} + W_{hr}H_{t-1} + B_{hr}) \quad (4.13)$$

$$Z_t = \sigma(W_{iz}X_t + B_{iz} + W_{hz}H_{t-1} + B_{hz}) \quad (4.14)$$

$$N_t = \tanh(W_{in}X_t + B_{in} + R_t \odot (W_{hn}H_{t-1} + B_{hn})) \quad (4.15)$$

$$H_t = (1 - Z_t) \odot N_t + Z_t \odot H_{t-1} \quad (4.16)$$

Here, the $R_{t \times N_{out} \times 1}$, $Z_{t \times N_{out} \times 1}$ and $N_{t \times N_{out} \times 1}$ are the reset, update and new gates respectively. The other name conventions are analogue to these of the LSTM. The update gate controls the balance between keeping the previous hidden state and incorporating the new input. It acts similarly to the combination of the forget gate and input gate in the LSTM. The reset gate allows the model to decide how much past information to ignore, enabling the model to reset the hidden state

4 Foundational model structures

if needed. The candidate activation provides new information to be added to the hidden state, modulated by the reset gate. Lastly, the final hidden state is produced for the current time step.

4.4 Model name conventions

During the dissertation, the following name conventions will be used to describe the used architecture parameters.

Table 4.1: Centralised model hyperparameters.

baseline	
Conv2D layers	5
Fully Connected nodes	512
LSTM nodes	512
Fully Connected nodes	72

Table 4.2: BM-CLA reference model hyperparameters.

BM-CLA reference ($\mathcal{F}512$)		
Conv2D layers	2	2
Fully Connected nodes	512	512
Conv1D filters	512	
LSTM nodes	512	
Fully Connected nodes	72	

Unless otherwise specified, all models are trained on the data from table 3.1 with amplitude normalisation (N_c) from section 4.1.1. If parameters are not mentioned, they are the same as those in the reference model. For the Conv1D layer, the number of nodes always equals the size as in the reference model, unless if the temporal layer is adjusted, it equals the temporal layer size. The temporal layer is similar to the reference model with LSTM unless different mentioned. Naming conventions are clarified through examples. For instance, 3C128 $\mathcal{F}1024$ G refers to a BM-CLA model with 3 Conv2D layers (instead of 2) and 128 filters per convolution (instead of 64 in the reference model). The size of embedded feature layers, Conv1D, and temporal layers (a GRU in this case) are set to 1024.

In the context of knowledge distillation, the teacher is mentioned first, followed by the student and learning method. If not specified, offline training is used. For example, in base $\mathcal{F}256$ -32L32 $\alpha 0.2$ best, the teachers are the baseline model and the $\mathcal{F}256$ L256 BM-CLA model, with 256 features, Conv1D, and LSTM layer sizes. The student with $\mathcal{F}32$ L32 has 32 features, Conv1D, and LSTM layer size. In this instance, the best teacher is used in the knowledge distillation with a weight factor α of 0.2. Only for the $\mathcal{F}256$ L256 model as teacher in KD, the L256 is removed to save space in the graphs. In all other cases,

4 Foundational model structures

apart from knowledge distillation the temporal layer should be mentioned if it differs from the reference model.

4.5 Test method

The centralised and co-operative models both give predictions for the multi-hot vector over the DOA ranges for every time index. So the output has a dimension of $[\text{batch}, N, I_\varphi]$. The azimuth angles are the relative positions to the listener. The accuracy calculation will be based on the prediction error. For calculating the error in the case of one source sound, only the highest probabilities are used. For multiple sources, only the largest prediction peaks are used for every source, to have vectors with fewer estimated angles for every time frame. Dimensions of predicted and true angles per time frame are equalised to the largest dimension. Then, $\frac{I_\varphi!}{(I_\varphi - I_{true})!}$ possible permutations of all sources between the true and predicted angles are made. If the estimated amount of predicted sources is bigger than the true sources, only the DOA's within the true label dimension remain. Now the absolute angle error of all sources is calculated of all sources. The lowest error with corresponding angle estimation will be taken. Lastly, with a buffer of five degrees for all sources together, the accuracy can be calculated with the total true predicted frames divided by the total amount of frames.



Figure 4.3: Legend for tested accuracy conditions.

The legend presents the average test accuracies for the given scenarios. These tests were conducted using the data outlined in table 3.1. A total of 900 samples were tested, with 25 samples allocated to each condition. During testing, the positions of the listener and speaker are held stationary, thereby eliminating the Doppler effect.

5

Model compression

With model compression, the model must have an accuracy as high as possible with focus on complexity and so energy consumption. Not only the amount of parameters but also the floating-point operations and the bandwidth will be considered. The latter will have the main focus. Starting with the BM-CLA model that is proven to work good in highly noisy and reverberant environments with multiple speakers (fig. 4.2). Reduction from layer sizes, to quantisation and knowledge distillation. All models have conventions as described in chapter 4.

5.1 Methods

5.1.1 Metrics

Parameters

Reducing the size of the model will decrease the amount of trainable parameters and so, close to the amount of calculations. This will contribute to the reduction of energy use. This will also lower the inference time which can improve the comfort of the end user. Lastly, training will go faster with lower amount of parameters. The number of parameters can be calculated when looking at the equations 4.4, 4.6, 4.7-4.12. For the GRU, only the number of gates will be different from the LSTM. In the equations provided, the weights of the layers are denoted on the left-hand side, while the biases are enclosed within brackets and also situated on the left side. Adjacent to the formula for each layer type, the parameter count pertaining to the $\mathcal{F}512$ reference model is explicitly indicated.

Table 5.1: Given names for network layer parameter variables.

PA	= parameter count
N_{in}	= number of input features
N_{out}	= number of output features
N_f	= number of feature maps
N_G	= number of gates (LSTM, GRU)
N_T	= number of temporal layer nodes
N_{Gr}	= number of groups
K_s	= kernel size

5 Model compression

$$PA_{Conv2D} = N_{in}K_sN_{f2d} + (N_{f2d}) \quad (8,576 + 128) \quad (5.1)$$

$$PA_{FC} = N_{f2d}K\mathcal{F} + (\mathcal{F}) \quad (8,421,376 + 512) \quad (5.2)$$

$$PA_{Conv1D} = \frac{\mathcal{F}}{N_{Gr}}K_s\frac{N_{f1d}}{S} + (\frac{N_{f1d}}{S}) \quad (524,288 + 512) \quad (5.3)$$

$$PA_{LSTM(GRU)} = N_TN_G(\frac{N_{f1d}}{S} + N_T) + (2N_TN_G) \quad (2,097,152 + 4096) \quad (5.4)$$

$$PA_{Out} = N_TI_\varphi + (I_\varphi) \quad (36,864 + 72) \quad (5.5)$$

$$PA_{BatchnormConv2D} = 2N_{f2d} \quad (128) \quad (5.6)$$

$$PA_{BatchnormConv1D} = 2N_{f1d} \quad (1024) \quad (5.7)$$

$$PA_{BatchnormFC} = 2\mathcal{F} \quad (1024) \quad (5.8)$$

$$PA_{BatchnormLSTM(GRU)} = 2N_T \quad (1024) \quad (5.9)$$

The total parameter count in the models can be calculated by summing up the outcomes of the formulas corresponding to each layer. The formulas illustrates that reducing the layer size significantly impacts the total parameter count. Specifically, halving the layer size results in a proportional reduction in parameters across two layers, with the exception of the output layer. Furthermore, employing a larger stride (S) in convolutional layers has the similar compounded effect for amplifying the reduction in the parameter count. This dual influence, underscores the critical role of both layer size and stride in optimising the computational efficiency of the models. Note, that the bias and batch normalisation have lower impact than the weight parameters. Also convolutions have much lower parameters for most cases. For example, the parameter count for a FC with 512 input and 512 output will have 262,144 parameters while a conv2D with 64 input, kernel 2 and 64 filters, will have only 8,192 parameters. Even more parameters are trained for the LSTM layer, a 512 input and 512 output would give 2,097,152 parameter count without bias. The advantage of the GRU is that only 3 gates are used an so the parameter count is only 75% of the parameter count from the LSTM.

MAC

Multiply-accumulate (MAC) is referred to as a multiplication followed by an addition. This can be seen as two floating-point operations(FLOPs). Both numbers give an indication for the amount of calculations for going through the network once. The amount of computations is inversely proportional to the amount of energy used. Therefor this parameter will be the first measure next to the parameter count, to compare models by energy consumption and complexity. The following formulas give approximations for the MACs, here the bias is left out for the parameter count (PA). The number of frames in real-time applications is one ($N = 1$).

$$MAC_{Conv2D} \approx PA \times NK \quad (5.10)$$

$$MAC_{FC,LSTM,GRU} \approx PA \times N \quad (5.11)$$

5 Model compression

The formulas indicate that the MAC operations align with the parameter count. Note, that for the 2D convolutions, the matrix calculations are across the entire input, and so multiplied by the number of frames. This results in much higher MAC in comparison to its parameter count. It is important to mention that for the on-ear devices, calculations are performed twice. However, this duplication is not counted in the parameter tally because of the parameter sharing.

Data transmission rate

A third metric of the energy consumption will be the data transmission. The transmission rate is dependent of what transmission channel will be used for the end user. Sending more data remotely will always use more energy. The most prominent use of transmission channel is Bluetooth Low Energy at this moment. Because energy consumption for data transmission is protocol and hardware dependent, therefore the bandwidth will be used as measurement.

The centralised system with uncompressed audio requires a bandwidth (in bit per second (bps)) of:

$$BW_{\text{uncompressed}} = 2 \times M_e \times f_s \times b, \quad (5.12)$$

where the bit depth b is typically 32 bits per sample.

For the codec-compressed data transmission the bandwidth can be calculated as:

$$BW_{\text{coded}} = 2 \times M_e \times \text{bit rate}, \quad (5.13)$$

where the lowest bit rate of the LC3+ codec is 16 kbits per second.

For the co-operative networks, the bandwidth is calculated right after the Convolution Encoder layer. The formula is:

$$BW_{\text{cooperative}}(\mathcal{F}, S, Q) = 2 \times \mathcal{F} \times \frac{f_s}{L \times S} \times \frac{b}{Q} + (2 \times b), \quad (5.14)$$

which makes a clear split in the three components where the bandwidth can be reduced. Decreasing the number of features $\mathcal{F}$ per time frame, increasing either the hop length L or the stride S , or decreasing the bits per feature with the quantisation factor Q . Note that this BW is *independent* of the number of microphones per ear, since this gets compressed with the convolutional layers. The added number in brackets is for the amplitude normalisation as described in section 5.1.6. Given its negligible size, it will not be taken into consideration.

It is notable that for the centralised systems, the availability of the audio signals at the central processor might also be useful for other application like speech enhancement, scene classification, *etc.* Nevertheless, this is very dependant on the application and while here is only looked at the localisation accuracy, these other applications should also be considered in holistic implementations.

5.1.2 Quantisation

Post-training static quantisation (PTSQ) is a method used to exchange value accuracy for value compression in neural network models. This technique involves converting the floating-point data type, typically represented with 32 bits per feature, to an unsigned integer format with 8 bits. The unsigned integers get a value in range from 0 to 0.256 with a scale resolution

5 Model compression

of 0.01. Utilising an unsigned integer is viable owing to the prior ReLU activation function, which ensures non-negativity of the values. Subsequently, once the values are compressed to an integer format, they undergo decompression back to float32, thereby affording greater flexibility during the latter stages of network processing. Using the quantisation on the transmitted broadband features, results in four times lower bandwidth.

5.1.3 Reducing layer size

The reference BM-CLA has 512 nodes in the FC layer and 512 in the LSTM layer. The amount of nodes in the FC layer can be reduced to force the model to make a better selection of broadband embedded features, and so reduce the shared data between the devices. This might lead to decreased performance. On the other hand, this could make the model generalise better because of more abstraction. When lowering the amount of features, the LSTM might be set smaller as well, because of more compact information flow. Analysis of the parameter equations delineated in equations 5.2 and 5.4, representing the fully connected and temporal layers respectively, underscores the substantial influence exerted by these layers on the overall parameter count. This pronounced impact emanates primarily from the dimensions of the input tensor corresponding to the FC layer, which encompasses a spatial extent of $64 \times K$. Next to contributing to the parameter count, the reduction of the FC layer is proportional to the used bandwidth in the feature transmission. It is noteworthy that opportunities for parameter reduction exist across the FC, Conv1D, and LSTM layers, all of which collectively contribute significantly to the parameter count. Specifically, these layers collectively account for a substantial proportion of the parameter count, representing 11,042,816 parameters out of the aggregate 11,096,776 parameters, constituting an overwhelming majority at 99.51%.

5.1.4 Time resolution

The latent localisation features are computed per STFT frame and transmitted to the central processing module. The frequency of feature transmission can be calculated using the formula:

$$f_f = \frac{f_s}{L \times S} \quad (5.15)$$

When time resolution is reduced by increasing the parameter L , only specific time sequences are considered. Alternatively, temporal compression can be achieved by employing a convolutional encoder followed by a convolutional decoder within the network bottleneck, as illustrated in fig. 4.2. Unlike simple sequence skipping, this method allows for the learning of compression patterns while retaining all relevant information. The compression is done by using a stride (S) larger than one. Both compression techniques divide the bandwidth size by adjusting S or L . For the Decoder Encoder solution, the drawback are the extra layers which bring $2 \times PA_{Conv1D}$ more parameters which is negligible due to the high number of groups.

5.1.5 Frequency resolution

Numerous studies have demonstrated that low frequencies primarily influence ITD, while high frequencies predominantly affect ILD [49] [50]. In this context, the initial 100 frequencies are extracted from the original set of 257 frequencies. Consequently, the upper frequency limit extends only to 3125 Hz, instead of 8000 Hz. While the exclusion of the highest frequencies

5 Model compression

might simplify the problem in terms of calculations, it also entails a loss of detailed information. However, such simplification may yield favorable outcomes for the methodologies proposed in preceding sections. Lowering the frequencies from 257 to 100 would also reduce the parameters for the fully connected layer as described in eq. (4.6) with a factor 2.57.

5.1.6 Amplitude normalisation

Most of the tested models employ normalisation as described in section 4.1.1. This approach is used to maintain consistency with the baseline model, thereby mitigating issues related to bandwidth.

For real use case scenarios, this would add $2 \times K \times \frac{f_s}{L} \times b$ bits per second to the bandwidth for sharing the averages (b = bit depth) for normalisation. To overcome this problem, normalisation with the amplitude with the channel-frequency-time average (Ncft) can be used. Next to improving bandwidth for the co-operative approach, this normalisation might also improve overall accuracy. The frequencies with lower importance will now stay low on impact due to the normalisation factor over all frequencies. Additionally, this will be averaged over time what again, ignores unused frequency bins in a longer time span. This abstraction is visualised in fig. 5.1 for one time instance and only three microphones.

$$\overline{Y_{e,m}} = |Y_{e,m}| / \left(\frac{1}{2MKL} \sum_{\varepsilon=1}^2 \sum_{\nu=1}^{M_e} \sum_{\kappa=1}^K \sum_{\eta=1}^{\frac{f_s}{L}} |Y_{\varepsilon,\nu,\kappa,\eta}| \right) \quad (5.16)$$

Only of mentioned, this normalisation is used. Now the bandwidth needed for normalisation is only $2 \times b$.

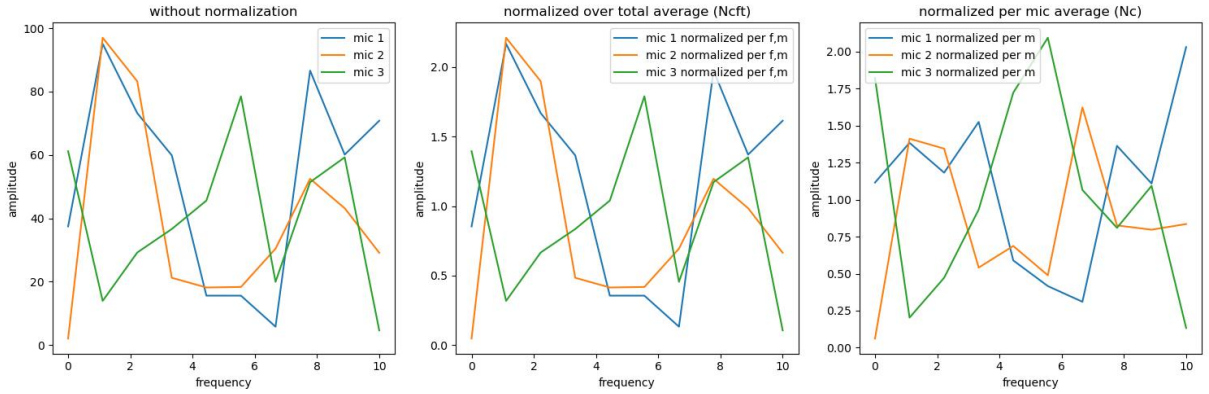


Figure 5.1: Amplitude normalisations depicted per channel per microphone.

5.1.7 Knowledge distillation

As described in section 2.4.2, with knowledge distillation (KD) a smaller model, the student, inherits knowledge from the better and often bigger model, the teacher. Student training methods will be explained in this section.

Students

The student architecture and model will be the model that is used in the real application. Therefore the feature and parameter size have to be chosen with care. The architecture of the BM-CLA model is taken under the loop to test different hyper

5 Model compression

parameter combinations. Variations of fig. 4.2 will serve as the student models. Additionally, training on a smaller dataset than that used for the teacher model will be tested. This approach is employed to amplify the performance disparity between the student and teacher models.

Teachers

For the teacher, the size is not important for the application itself. The model will only be used during training the student. This gives freedom in choosing what the teacher architecture will look like. Bigger versions of the baseline model (fig. 4.1) and the BM-CLA models itself will be used. With in mind, the better the teacher the better the student will be.

Loss

The goal of the KD is that the student mimics the methods that were learned in the teacher model. This is done by adding a loss made up from comparisons in the hidden layers between teacher and student, instead of only the output layer and the true labels. The following loss functions were tested:

$$Loss = (1 - \alpha)BCE(\hat{y}^S, y) + \alpha(CE(\hat{y}^S, \hat{y}^T)) \quad (5.17)$$

$$Loss = (1 - \alpha)BCE(\hat{y}^S, y) + \alpha((1 - \beta)CE(\hat{y}^S, \hat{y}^T) + (\beta cosEL(\hat{f}^S, \hat{f}^T)) \quad (5.18)$$

$$Loss = (1 - \alpha)BCE(\hat{y}^S, y) + \alpha(KL(log\sigma(\hat{y}^S), \sigma(\hat{y}^T))) \quad (5.19)$$

The loss consist of two or three combined losses with α and β the weights for each loss. The first loss is between the student its predicted output and the target labels. Loss will increase in line with distance from the wanted output. This loss is calculated with a binary cross-entropy (BCE) which is used widely for binary classification, even for multi-class classification. Only using this, would form the loss for the baseline and reference models. A second loss is calculated based on the predictions of the student and teacher networks. To have a better overview of the predictions, Softmax is taken for the teacher outputs and $\log(\text{Softmax})$ is used for the student outputs for the Kullback-Leibler (KL) loss. Hereby not only the correct output will be taken in to account but also the entire distribution for how the teacher sees the given situation. This could lead the student better in a rough direction for better generalisation. The used loss functions are KL divergence loss and cross entropy (CE). A comparison between the FC hidden layers of the networks define the third loss. This is calculated with a cosine embedding loss (cosEL). The loss functions are defined as following.

$$Loss_{KL}(y_{pred}, y_{true}) = y_{true} \log \frac{y_{true}}{y_{pred}} \quad (5.20)$$

$$Loss_{CE}(x, y) = \sum_{n=1}^N \frac{l_n}{\sum_{n=1}^N y_n} \quad l_n = \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})} \quad (x = y_{pred}, y = y_{true}) \quad (5.21)$$

$$Loss_{cosE}(y_{pred}, y_{true}) = 1 - \cos(y_{pred}, y_{true}) \quad (5.22)$$

For the hidden feature loss with cosine loss, average pooling is done to match dimensions between student and teacher.

5 Model compression

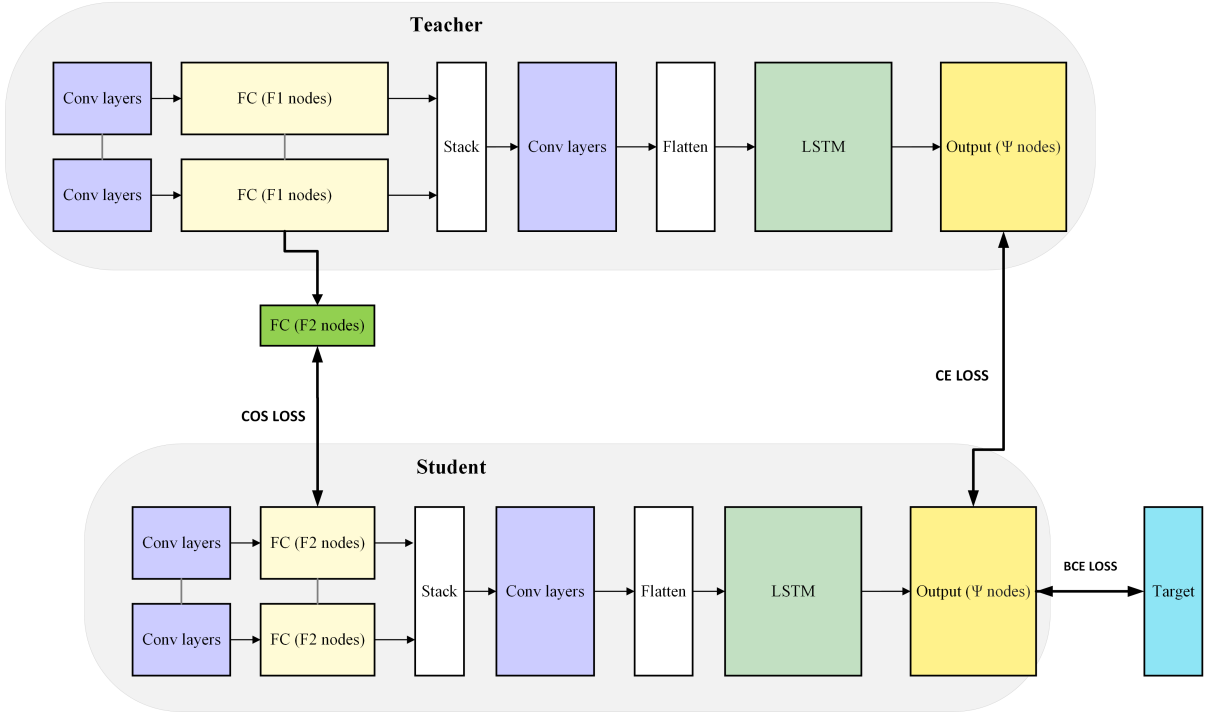


Figure 5.2: BM-CLA model with $\mathcal{F}=\mathcal{F}_1$ as teacher and compressed student BM-CLA model with $\mathcal{F}=\mathcal{F}_2$. Knowledge distillation with 2 additional losses.

5.1.8 Attention network

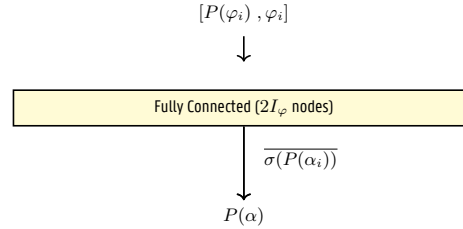


Figure 5.3: Depicts the attention layer to optimise α for knowledge distillation. The fully connected layer takes the logits from both the student and teacher models as input. This input is processed, and the sigmoid function outputs an α value between 0 and 1 for every time index. The mean value per time index is then computed to obtain a single value.

An attention network has been tested to learn a hyper parameter α during training. This approach is based on the notion that the assistance provided by the teacher network may vary across different localisation scenarios. Consequently, the networks can autonomously determine the optimal, now dynamic, hyperparameter α . After getting the right weight α , backpropagation can be done for updating the weights. In this attention model, the outputs from both the teacher and student networks serve as input. The attention layer leverages the student's loss, such that when the student's loss is higher, the attention layer is adjusted accordingly. This dynamic adjustment ensures that an improved attention mechanism leads to enhanced performance of the student network, and vice versa.

5 Model compression

5.1.9 Ensemble learning

Utilising multiple teachers may provide greater opportunities for the student model to learn the best features from each teacher. In this study, both the reference model and baseline model were engaged concurrently as teachers. Initially, the average loss between both student-teacher pairs was calculated. An alternative approach involved selecting the optimal teacher for each training batch based on the binary cross-entropy loss of each teacher's output relative to the labels. The teacher exhibiting the smallest loss was then chosen to guide the student model for that particular input.

5.1.10 Training schemes

For the knowledge distillation without attention network, two training schemes were tested. The first one is offline learning. Here the teacher is pretrained and no backpropagation is done in the teacher network. The second training scheme is online learning. As the term says, in online learning, both the teacher and student learn together at the same time. The student might have a better understanding of the learning path the teacher will take during training.

Two training schemes were implemented for the attention model. The first method will be online training at the same epoch as the student. In this approach, the attention model is trained online with the student network. This allows the attention model for gradual learning. An alternative method involves alternating the training phases of the student and attention networks. Specifically, after every 20 epochs of student network training, the attention network is trained for 20 epochs. This alternating training strategy provides the attention model with dedicated periods to adjust and refine its parameters. This potentially leads to better alignment with the student at different learning stages.

5.1.11 Small dataset

To enhance the performance difference between the student and teacher models, the student will be trained on a relatively smaller dataset. This subset is derived from the dataset detailed in table 3.1. Specifically, it includes only the first five rooms with four distinct positions in each room, 60 out of the 72 noise instances, and 6318 out of 7438 utterances from TIMIT and PTDB-TUG.

5.2 Results and discussion

In this section, the adjusted models will be compared first in terms of accuracy, followed by comparisons that include bandwidth and size. During training, validation curves were used as indicators of potential test performance. However, this correlation was not always consistent, suggesting that early stopping may not be beneficial for all models. Consequently, it is important to note that models trained for 60 epochs may yield different outcomes if trained for 120 epochs. An overview of the compressed and expanded models can be found in the appendices fig. 14, fig. 15, fig. 16. Models can be chosen according to constraints and accuracy expectations.

5.2.1 Training behaviour

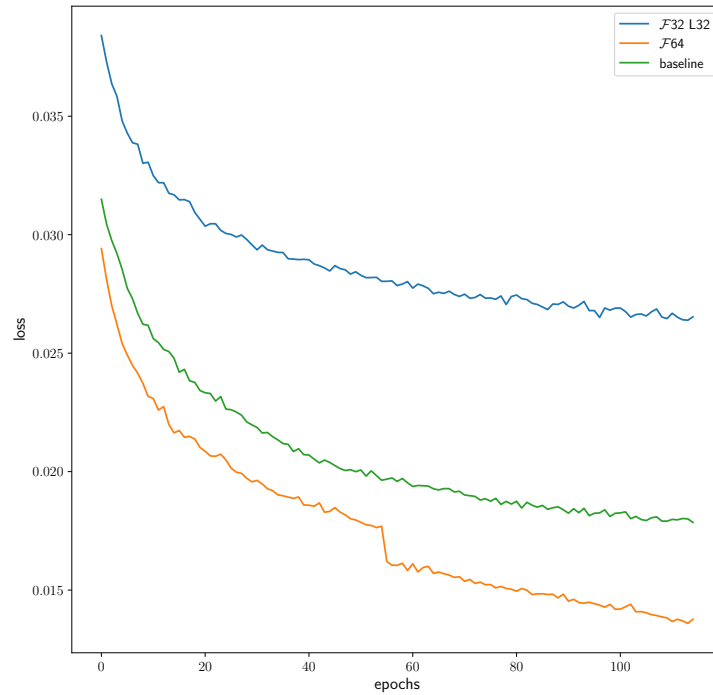


Figure 5.4: Training BCE loss for 120 epochs starting at epoch 5 for models with different parameter size.

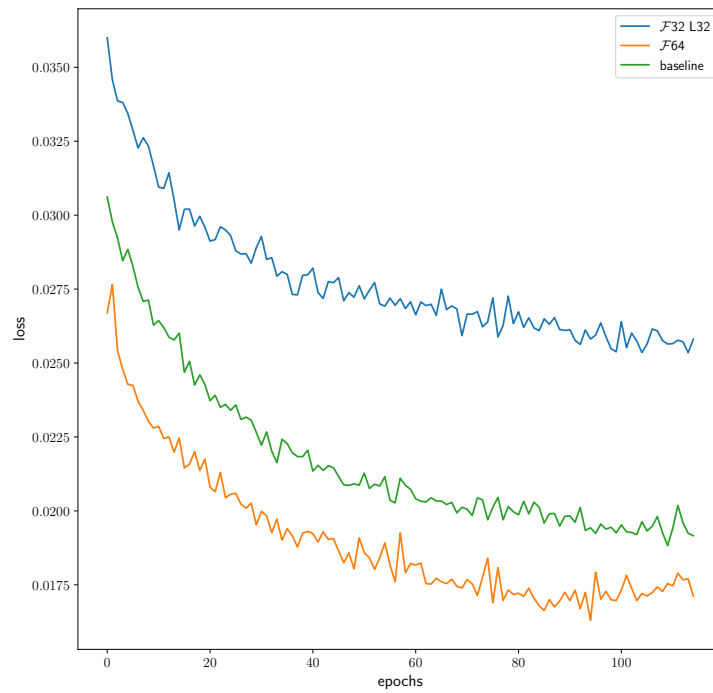


Figure 5.5: Validation BCE loss for 120 epochs starting at epoch 5 for models with different parameter size.

5 Model compression

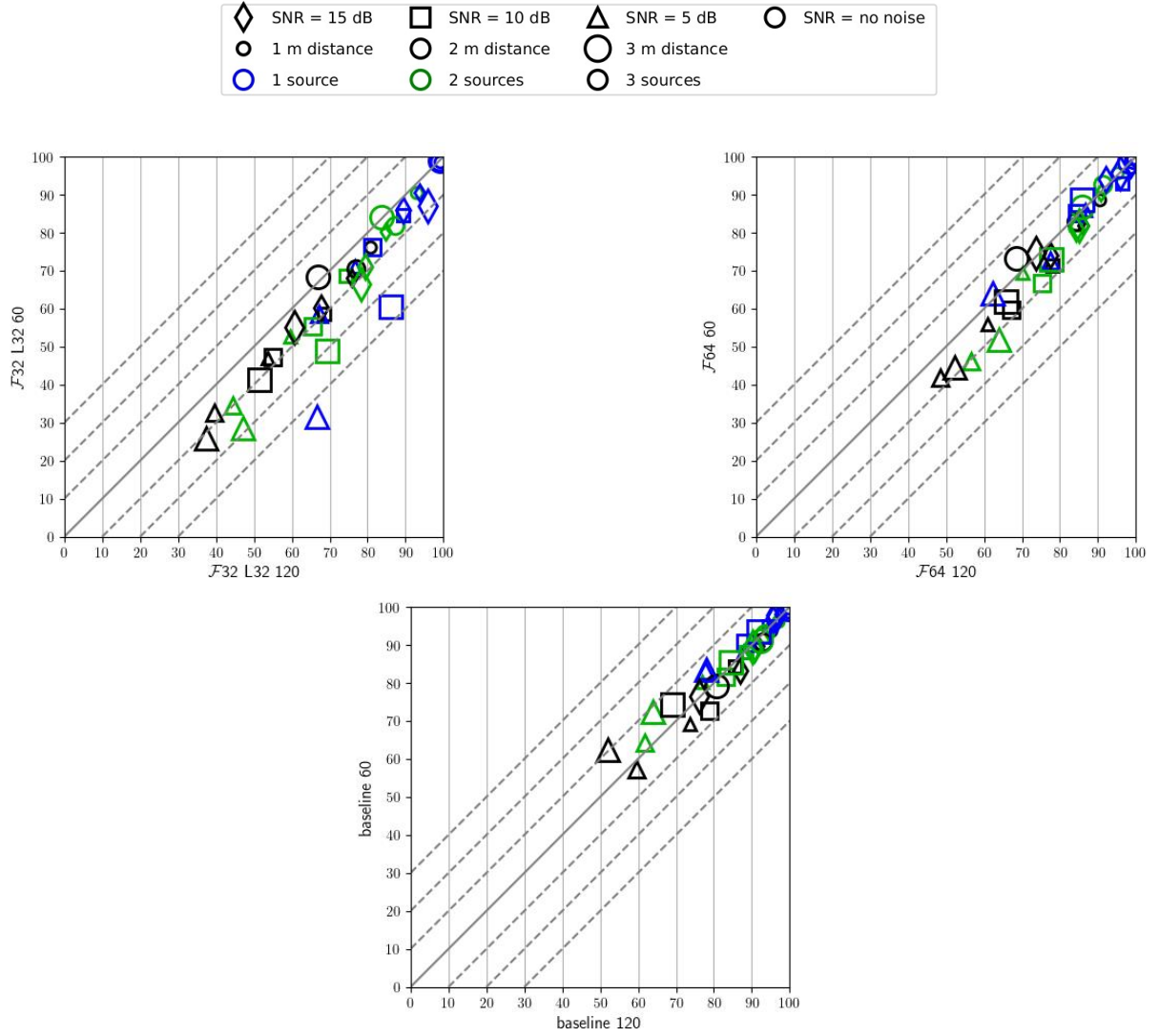


Figure 5.6: Comparison of test accuracy for training 60 and 120 epochs.

Table 5.2: Average accuracies in % for all tested conditions for the models depicted in fig. 5.6.

conditions		
epochs	60	120
model	accuracy (%)	
$\mathcal{F}_{32} \text{ L}_{32}$	65.45	73.41
$\mathcal{F}_{64}$	77.82	82.89
baseline	85.65	85.13

Examining fig. 5.6, the test performance shows significant differences depending on whether the models were trained for 60 or 120 epochs. Notably, the baseline model exhibited a decrease in accuracy, primarily due to weaker performance under

5 Model compression

5 dB SNR conditions. Smaller models displayed even larger discrepancies between the 60- and 120-epoch training regimens. The learning curves in Figures 5.4 and 5.5 reveal similar behaviors in the loss curves during training for all three models. Before knowing this occurrence in validation and test differences, some models during the research were only trained for 60 epochs before actual overfitting was seen in the training curves. This decision was made with efficiency in mind, aiming to mitigate time wastage during the training process. Training the models typically spanned durations ranging from 3 to 12 days depending on model size and what hardware was used. Due to examples as the $\mathcal{F}32$ L32, where the models performance changes enormous when training to 120 epochs, conclusions could be different at 120 epochs for these models. For this reason, certain tested combinations are not implemented in this dissertation.

5.2.2 Quantisation

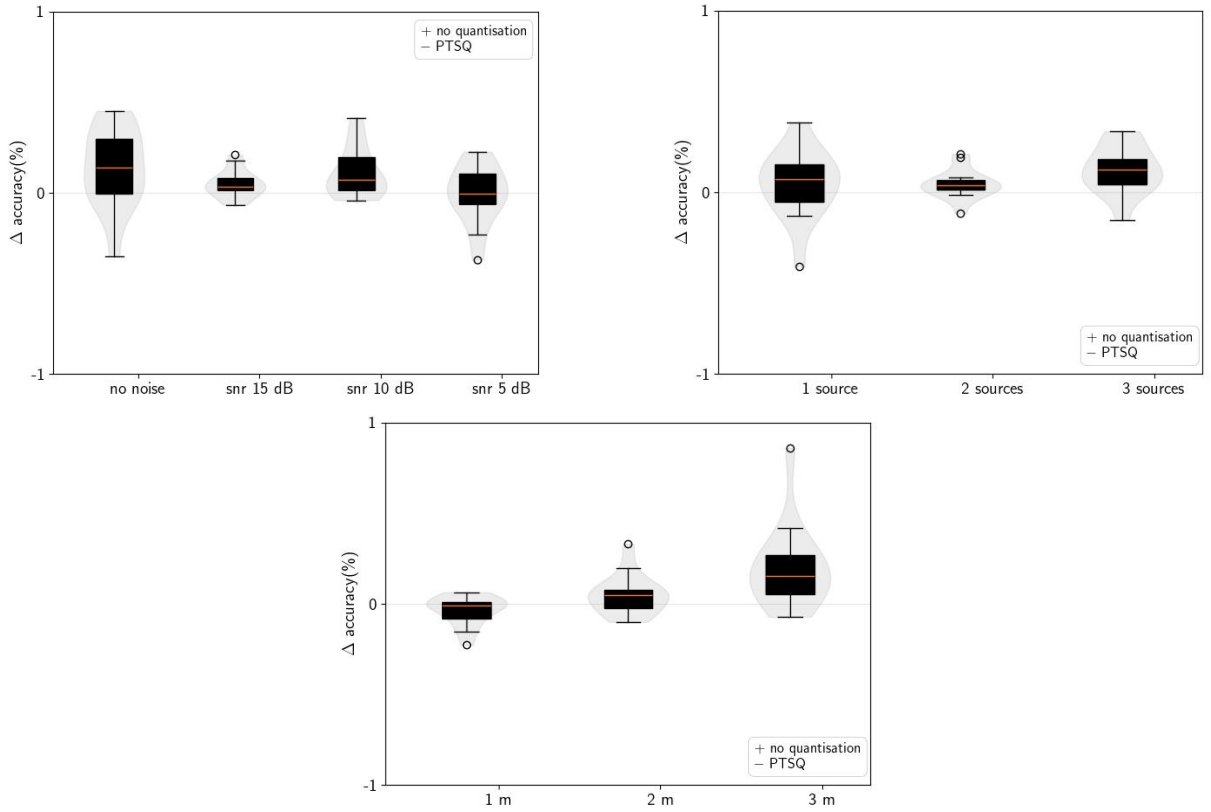


Figure 5.7: Boxplot illustrating the difference in accuracy between the averages of 14 models with and without PTSQ.

The accuracy impact of post-training static quantisation on the embedded broadband features is minimal. This is visualised in fig. 5.7, which presents the accuracy differences averaged over testing conditions for 14 models with and without quantisation. The differences in accuracy observed are less than 1% between the models with and without PTSQ.

5.2.3 Reducing layer size

Number of embedded features

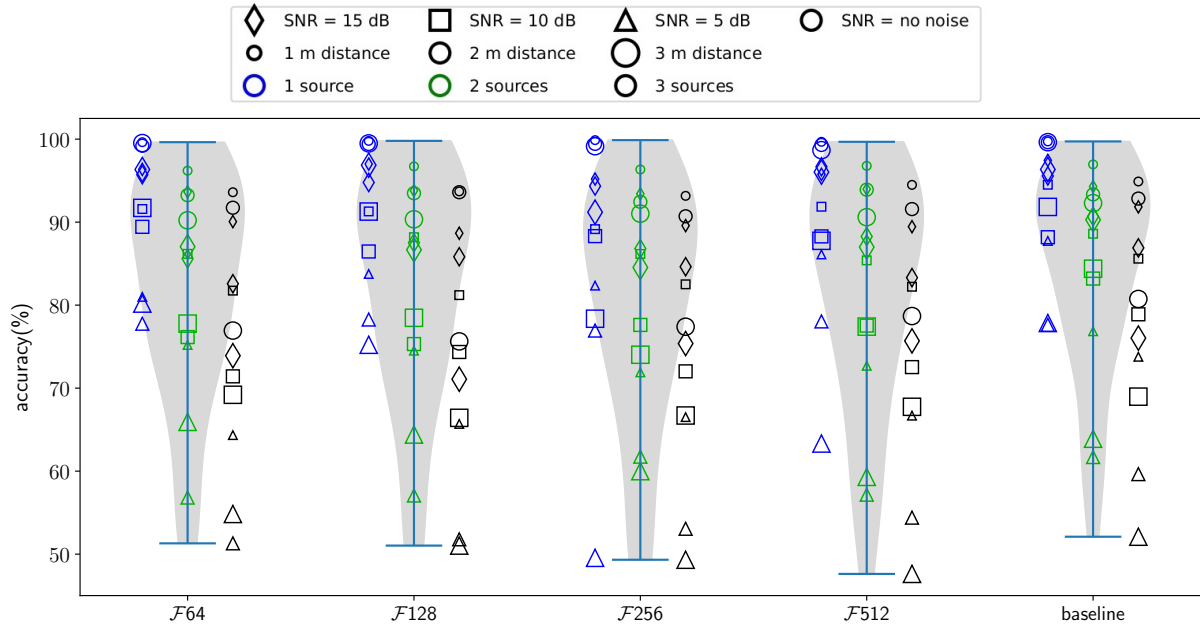


Figure 5.8: Test accuracy for adjusted BM-CLA models with varying quantities of broadband embedded features at the fully connected layer was evaluated. All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

Table 5.3: Average accuracies in % for all tested conditions for the models depicted in Figure 5.8.

model	accuracy (%)	parameters (millions)	MAC (millions)
$\mathcal{F}_{64}$	82.89	3.27	9.05
$\mathcal{F}_{128}$	82.75	4.39	11.25
$\mathcal{F}_{256}$	81.14	6.62	15.55
$\mathcal{F}_{512}$	82.39	11.10	24.25
baseline	85.13	10.60	32.75

Reducing the number of embedded features positively impacts accuracy. In the tested models, performance remained stable with down to 64 features. Notably, models with 128 and 64 features outperformed the reference model by approximately 0.5%. Conversely, the model with 256 features demonstrated weaker performance in single-source conditions compared to the reference model. When comparing the $\mathcal{F}_{64}$ model to the baseline, the baseline exhibited an overall better performance, with a difference of approximately 2% across nearly all conditions. When analysing the parameters and MACs, the BM-CLA achieves comparable to better performance with 2.68 times fewer computations and 3.39 times fewer parameters as in the reference model. Considering the formula for bandwidth 5.14, the bandwidth decreases proportionally with the reduction in

the number of features.

Number of embedded features, temporal layer and conv1D

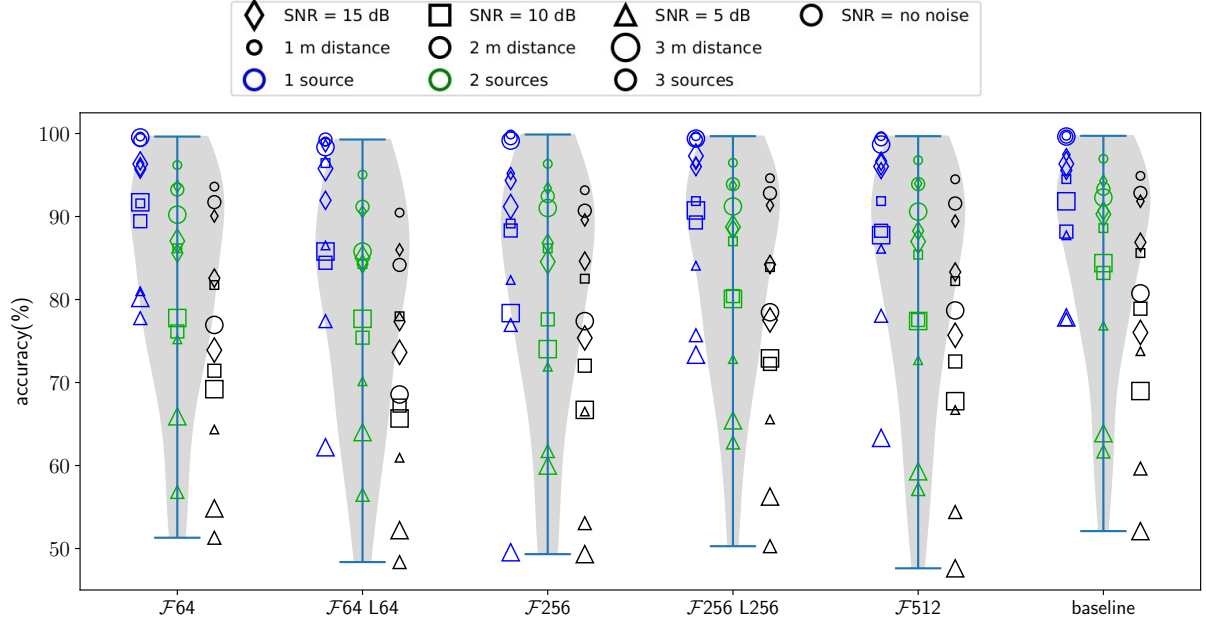


Figure 5.9: Test accuracy for adjusted BM-CLA models with varying quantities of broadband embedded features at the Fully Connected layer, LSTM and Conv1D were evaluated. The models with adjusted temporal layer have an equally adjusted Conv1D layer. All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

Table 5.4: Average accuracies in % for all tested conditions for the models depicted in Figure 5.9.

model	accuracy (%)	parameters (millions)	MAC (millions)
$\mathcal{F}_{64}$	82.89	3.27	9.05
$\mathcal{F}_{64}$ L64	80.22	1.10	6.90
$\mathcal{F}_{256}$	81.14	6.62	15.55
$\mathcal{F}_{256}$ L256	83.73	4.90	13.85
$\mathcal{F}_{512}$	82.39	11.10	24.25
baseline	85.13	10.60	32.75

Modifying the reference model to $\mathcal{F}_{64}$ L64, where the number of embedded features equals the number of nodes in the LSTM (64) and filters in the Conv1D, demonstrates a generally negative impact on model performance. No specific anomalies are observed in the differences. Conversely, for the model with 256 features, aligning the LSTM and Conv1D layers size with the number of broadband features, results in improved accuracy. Notably, the 3-meter, 1-source, and high-noise conditions exhibit the most significant improvement, averaging a 5% increase in accuracy for these scenarios. Examining the model

5 Model compression

sizes, it is evident that reducing the LSTM and Conv1D layers significantly improves the model's compactness, resulting in a parameter reduction of up to 2.97 times for the $\mathcal{F}64$ L64 compared to the $\mathcal{F}64$ model.

5.2.4 Time - frequency resolution

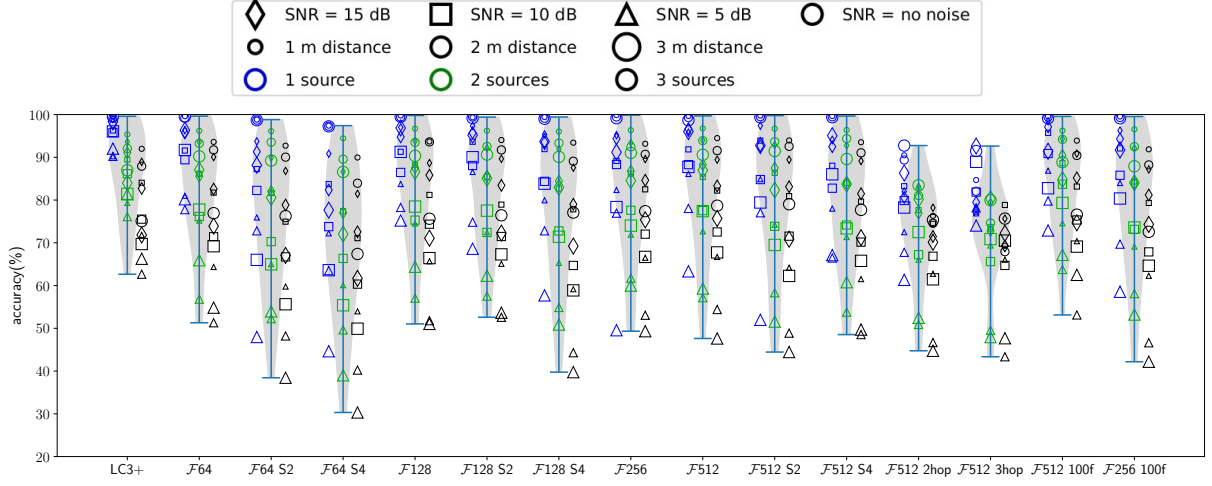


Figure 5.10: Test accuracy for BM-CLA models with focus on time and frequency resolution. The models were trained for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

Table 5.5: Average accuracies in % for all tested conditions for the models depicted in Figure 5.10.

model	accuracy (%)	parameters (millions)	MAC (millions)
LC3+	86.66	10.07	32.75
$\mathcal{F}64$	82.89	3.27	9.05
$\mathcal{F}64$ S2	76.57	3.27	9.05
$\mathcal{F}64$ S4	71.14	3.27	9.05
$\mathcal{F}128$	82.75	4.39	11.23
$\mathcal{F}128$ S2	82.07	4.39	11.23
$\mathcal{F}128$ S4	78.54	4.39	11.23
$\mathcal{F}256$	81.14	6.62	15.55
$\mathcal{F}512$	82.39	11.10	24.25
$\mathcal{F}512$ S2	80.36	11.10	24.25
$\mathcal{F}512$ S4	80.17	11.10	24.25
$\mathcal{F}512$ 2hop	73.38	11.10	12.13
$\mathcal{F}512$ 3hop	72.89	11.10	8.00
$\mathcal{F}512$ 100f	84.04	5.95	11.07
$\mathcal{F}256$ 100f	79.38	4.05	7.55

5 Model compression

In Figure 5.10, a comparison per model is made for the accuracy under tested conditions. Model names correspond to the architecture configurations detailed in Figure 4.2. If the stride S is not mentioned, the Conv Enc and Conv Dec are absent, and the features $\mathcal{F}$ are transmitted directly. In cases where the hop is not specified, the hop length defaults to one times L . First, the impact of lowering the time dimension is discussed. By increasing the hop length, an overall drop in performance is clearly visible. Specifically, for the zero and 15dB SNR conditions, performance collapsed the most. Respectively 16.09% and 11.82% accuracy differences compared to the $\mathcal{F}512$ reference model. This decline in performance highlights the critical role of the missed temporal information. The difference between doubling the hop length and tripling it is marginal despite the MAC difference. The use of a Conv Enc and Conv Dec is significantly more beneficial for lowering the time resolution in terms of accuracy. The most significant differences are observed in combination with the FC layer. When the number of embedded broadband features is reduced, information details are lost when the time dimension is decreased. This is clearly visible for the $\mathcal{F}64$ model. Conversely, when enough features are available, lowering the time dimension has a low impact on accuracy.

The number of features also influences behavior with respect to the frequency resolution. Lowering the number of frequencies per time frame has a positive impact on the $\mathcal{F}512$ model accuracy and MAC wise. Conversely, reducing the frequency resolution has a negative effect on the model with fewer features (256) on the accuracy. This might indicate that sound localisation becomes more challenging with lower frequency resolution. Here the low frequencies that lead to wrong ILD and so wrong information might have more influence.

5.2.5 Amplitude normalisation

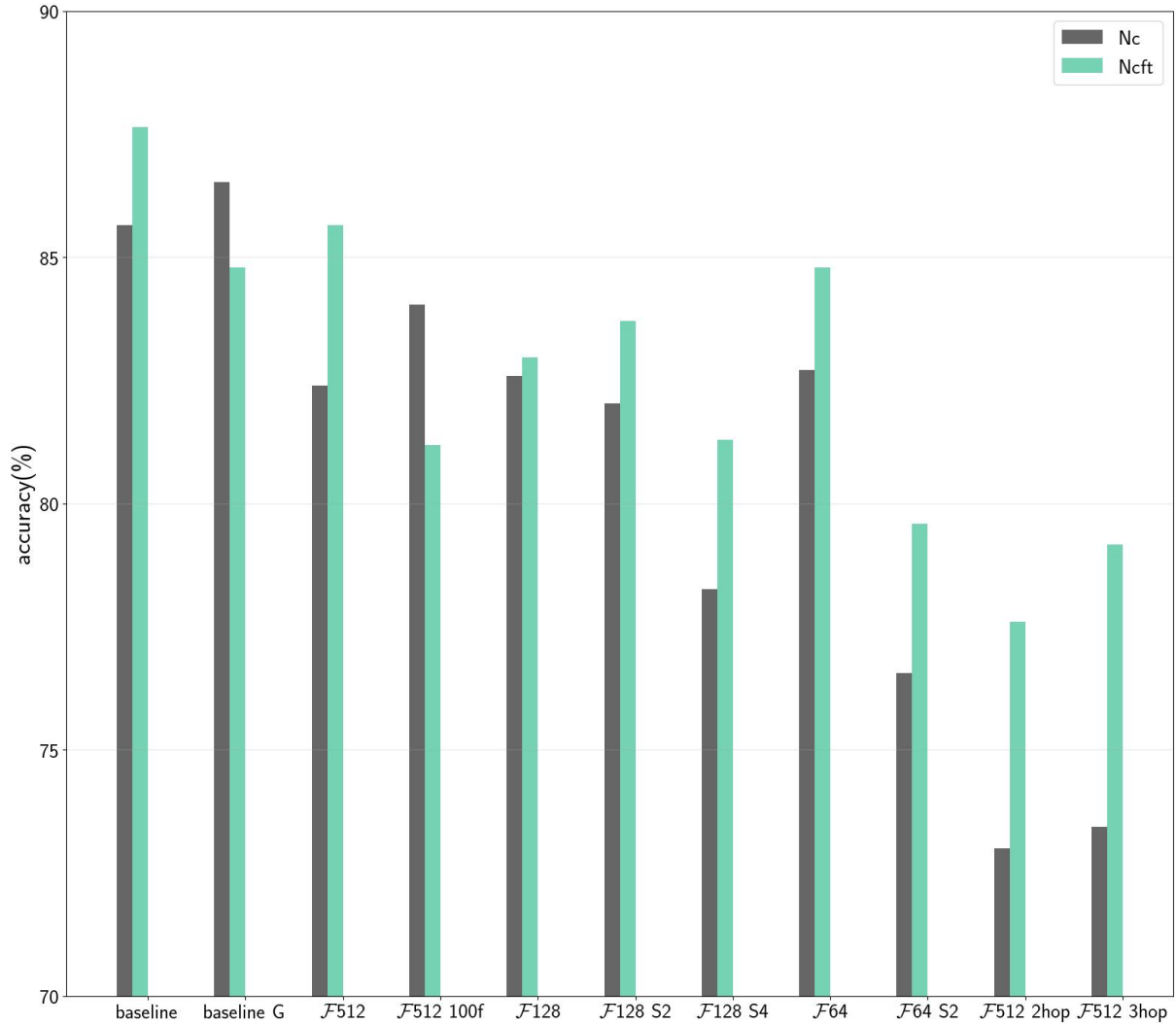


Figure 5.11: Test accuracy for BM-CLA models with amplitude normalisation over the channels (Nc) and normalisation over the channels, frequencies and time (Ncft) over 120 epochs.

Using the mean of the amplitude over all channels, time frames and frequencies, brings down the used bandwidth to 64 bps for normalisation. This is a factor 25,700 times smaller than using average over channels only. Amplitude normalisation across all dimensions leads to enhanced accuracy for most models, likely because it allows for generalisation over all frequencies. This makes noise in unused speaker frequencies less significant. Additionally, normalising over time decreases the impact of silent time frames. Conversely, models with fewer frequencies show reduced performance with comprehensive normalisation. In such cases, normalisation over only the channels is more effective. For only the 100 lowest frequencies, the phase has a greater impact than the amplitude due to the lower frequency spectrum (section 2.2.4). Since ITD is more crucial than ILD, the influence of amplitude on lowest frequencies, can be detrimental. Particularly because low frequency signals perform poorly for ILD. Taking these wrong ILD into account may lead to lower resolution in the output and thus

5 Model compression

lower overall accuracy. For the baseline equipped with a GRU, the temporal structure of the amplitudes might become too complex to be processed by a GRU.

All combinations

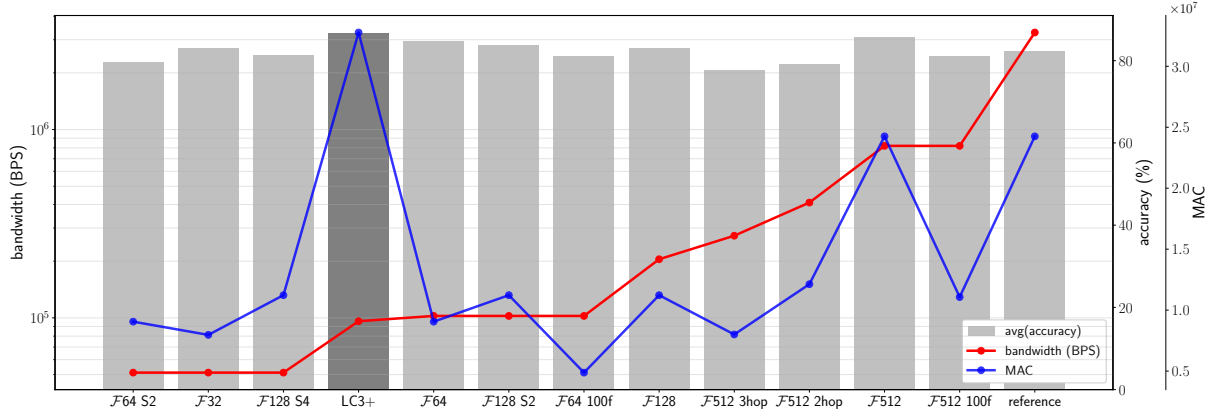


Figure 5.12: Test accuracy plotted in bars ordered in rising bandwidth from left to right. Bandwidth is illustrated on the red curve and MAC number is illustrated with the blue curve. All models have undergone PTSQ and amplitude normalisation (Ncft) with one normalisation factor. The reference model is the $\mathcal{F}512$ model with Nc normalisation without PTSQ. The darker color represents the centralised architecture. Bandwidth due to normalisation is not included.

Table 5.6: Test accuracy, bandwidth and MAC from models in Figure 5.6.

model	accuracy (%)	bandwidth (thousands)	MAC (millions)
$\mathcal{F}64$ S2	79.59	51.2	9.06
$\mathcal{F}32$	83.08	51.2	7.97
$\mathcal{F}128$ S4	81.29	51.2	11.23
LC3+	86.66	96.0	32.25
$\mathcal{F}64$	84.79	102.4	9.05
$\mathcal{F}128$ S2	83.70	102.4	11.23
$\mathcal{F}64$ 100f	81.20	102.4	4.87
$\mathcal{F}128$	82.96	204.8	11.23
$\mathcal{F}512$ 3 hop	77.59	273.1	8.00
$\mathcal{F}512$ 2hop	79.16	409.6	12.13
$\mathcal{F}512$	85.65	819.2	24.25
$\mathcal{F}512$ 100f	81.18	819.2	11.07
reference	82.39	3276.8	24.25

In the graphs presented in Figure 5.6, it is observable that the accuracy demonstrates a remarkable level of stability across a considerable spectrum of bandwidth variations. Interestingly, the bandwidth exhibits a range discrepancy spanning a factor

5 Model compression

of 32, whereas the MAC exhibits a comparatively narrower range, with a factor of 6.6 between its maximum and minimum values. No clear correlation in number of features or MAC size is present. Particularly noteworthy is the absence of any BM-CLA models outperforming the centralised baseline in terms of accuracy, which is equipped with the LC3+ codec. The MAC number on the other hand is better for the co-operative model.

5.2.6 Knowledge distillation

In this section of the thesis, knowledge distillation will be evaluated using various training methods and teacher-student combinations. By employing KD, complex and larger models can be leveraged to facilitate the training of smaller models used in hearing aids. This approach combines the advantages of both worlds, achieving high accuracy while maintaining low energy consumption. Next to usage in sound localisation, these ideas are applicable in many fields as earlier described in section 2.4.2. The use of loss functions other than the one specified in equation 5.20, exhibited significantly poor performance from the beginning. Therefore, no further testing was conducted with these alternative loss functions.

Different teacher layer architecture

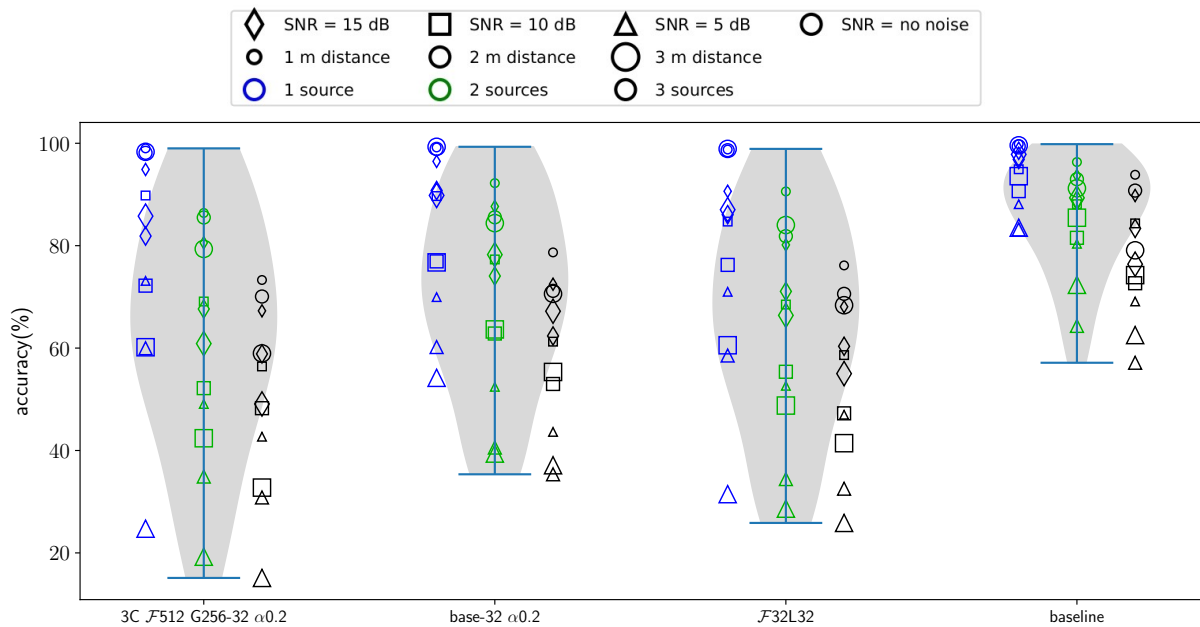


Figure 5.13: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

5 Model compression

Table 5.7: Average accuracies in % for all tested conditions for the models depicted in Figure 5.13.

model	accuracy (%)
3C $\mathcal{F}512$ G256-32 $\alpha 0.2$	63.02
base-32 $\alpha 0.2$	70.80
$\mathcal{F}32$ L32	65.45
baseline	85.65

Different teacher architectures have several outcome (for training 60 epochs), with the baseline model as the teacher providing better accuracy compared to the BM-CLA model utilising three convolutions. Since only the logits are used for the tested knowledge distillation, the structural complexity of the model should theoretically have no direct impact. The hidden weight factor α on 0.2 yielded the best results and will be utilised in subsequent experiments.

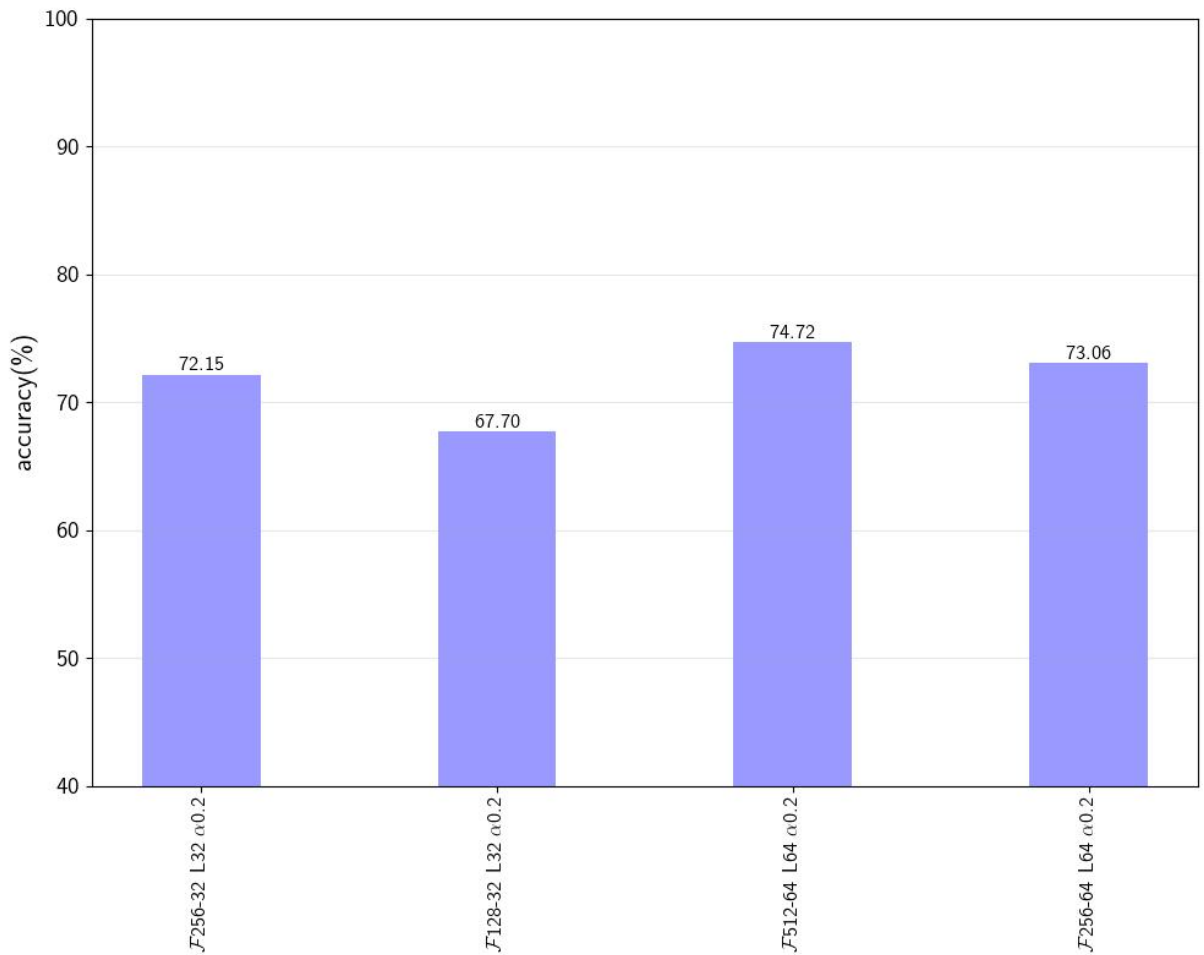


Figure 5.14: Test accuracy for BM-CLA models with the following name conventions, teacher-student. Where the teachers have slightly different architecture parameters. All models underwent training for 60 epochs.

5 Model compression

Only changing the network size with parameter adjustments, as expected, indicated that the larger teachers performed better. However, as shown later in Figure 5.20, for the $\mathcal{F}_{64}$ L64 model, a better-performing teacher does not necessarily translate to effective knowledge transfer.

Online KD training

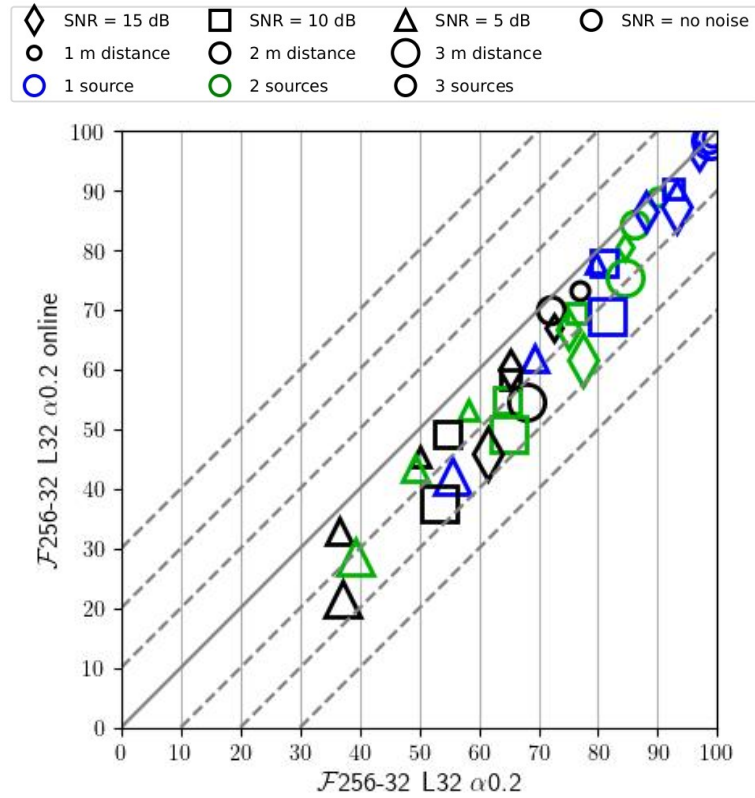


Figure 5.15: Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}_{32}$ L32 trained online and offline over 60 epochs.

5 Model compression

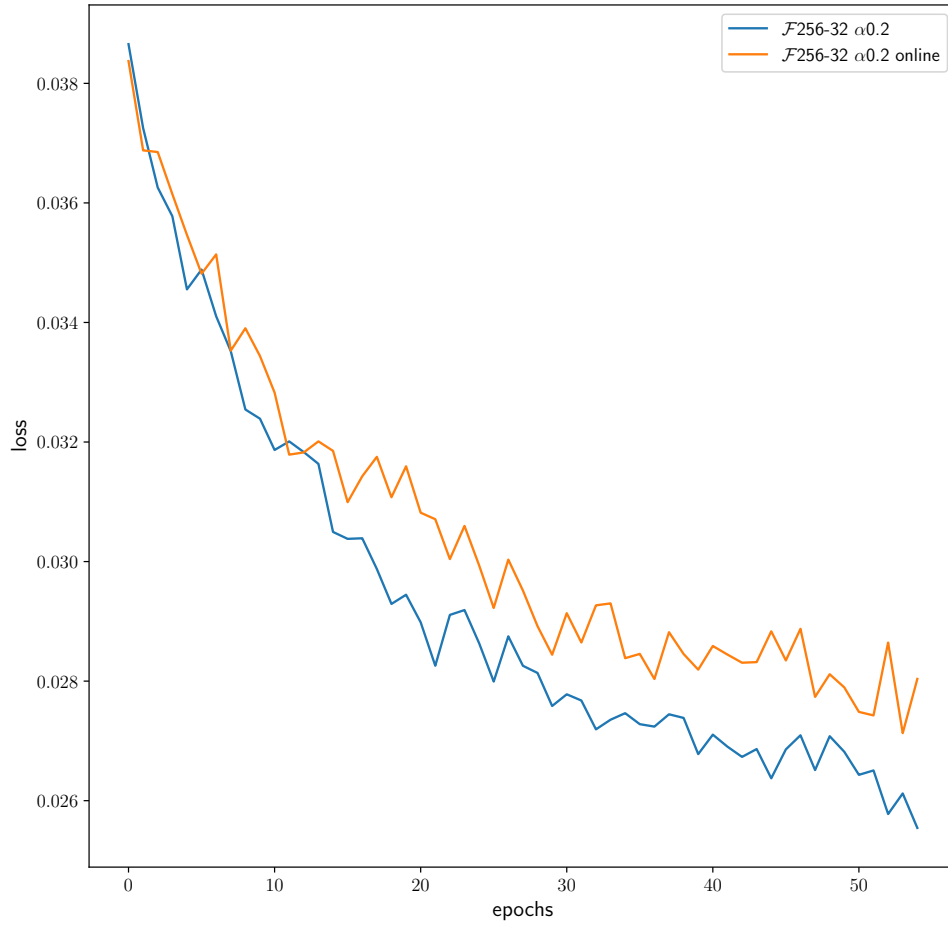


Figure 5.16: Validation BCE loss for 120 epochs starting at epoch 5 for KD online training.

During online training, the loss curve (5.18) exhibited poorer performance compared to offline training. Consequently, the training process was halted at 60 epochs. Next to expensive computing, the test results obtained from online training were not promising. This led to the decision to cease further testing in online training.

Alternate training

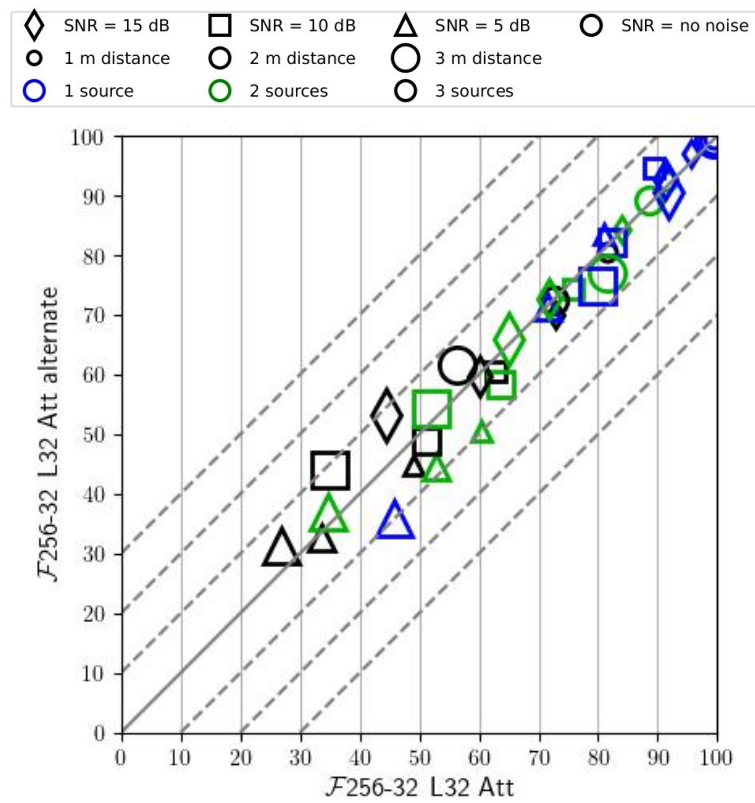


Figure 5.17: Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}^{32}$ L32 trained with alternation of the attention layer and offline training over 60 epochs.

5 Model compression

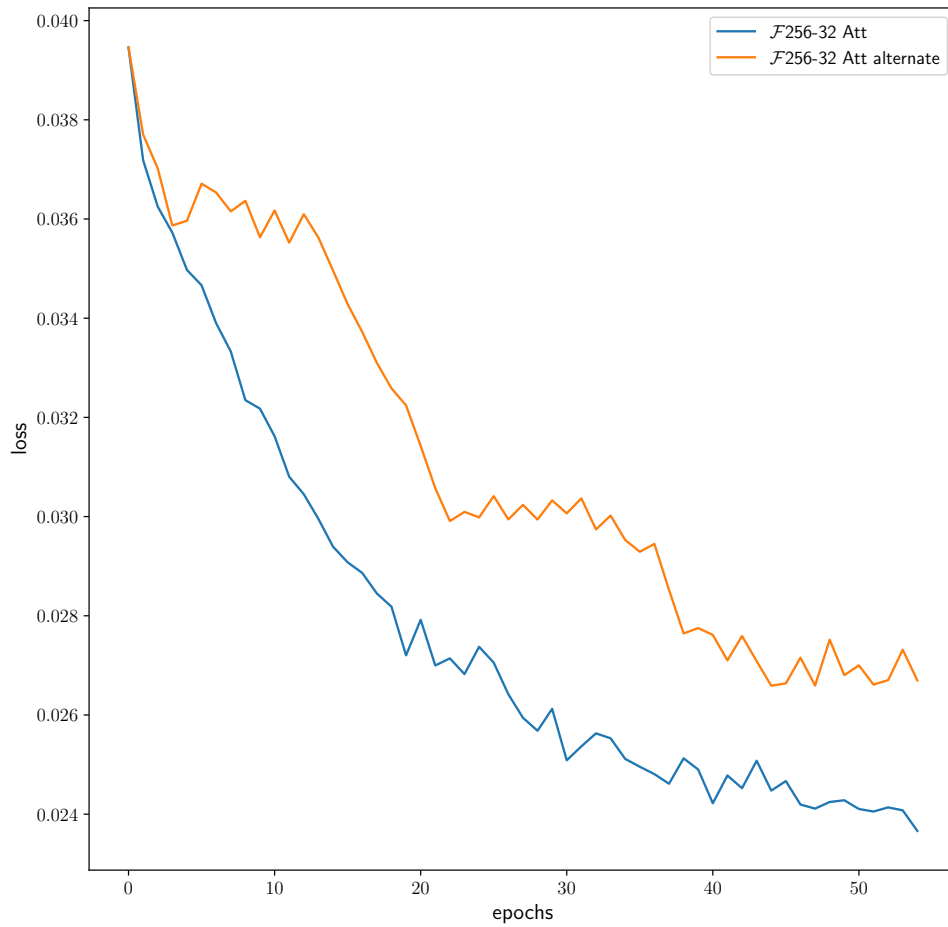


Figure 5.18: Validation BCE loss for 120 epochs starting at epoch 5 for KD alternate training.

Similarly to the case of online training, the validation loss curve (5.18) was not very promising. Therefore, the training was stopped at 60 epochs. Evaluating the test results at 60 epochs, as indicated by the validation loss, revealed no significant gains. Consequently, an additional 30 more epochs were conducted, yielding no improvements in results.

Ensemble KD training

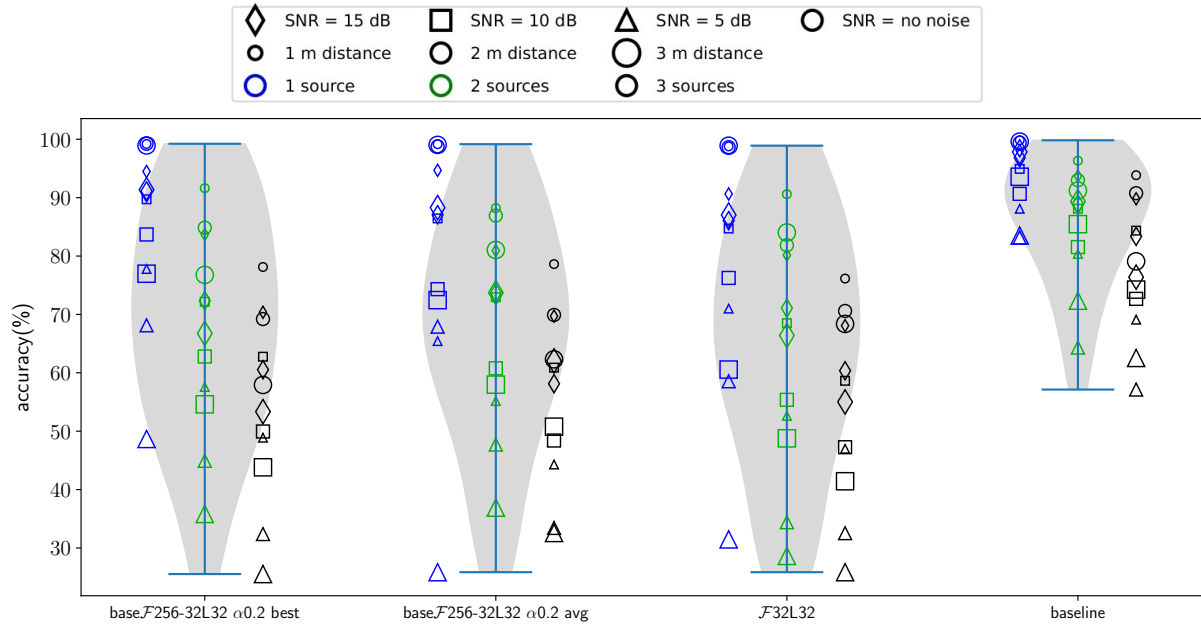


Figure 5.19: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and average hidden weight factor (Avg). All models underwent training for 60 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

Table 5.8: Average accuracies in % for all tested conditions for the models depicted in Figure 5.19.

model	accuracy (%)
base.F256-32 α 0.2 best	68.78
base.F256-32 α 0.2 avg	67.96
F32 L32	65.45
baseline	85.65

Examining the violin plot in Figure 5.8, it is evident that the ensemble teachers only slightly improved performance under high noise conditions. Within the given number of epochs, the models showed marginal improvement. It will be demonstrated later that results might differ significantly at 120 epochs.

KD - 60 epochs overview

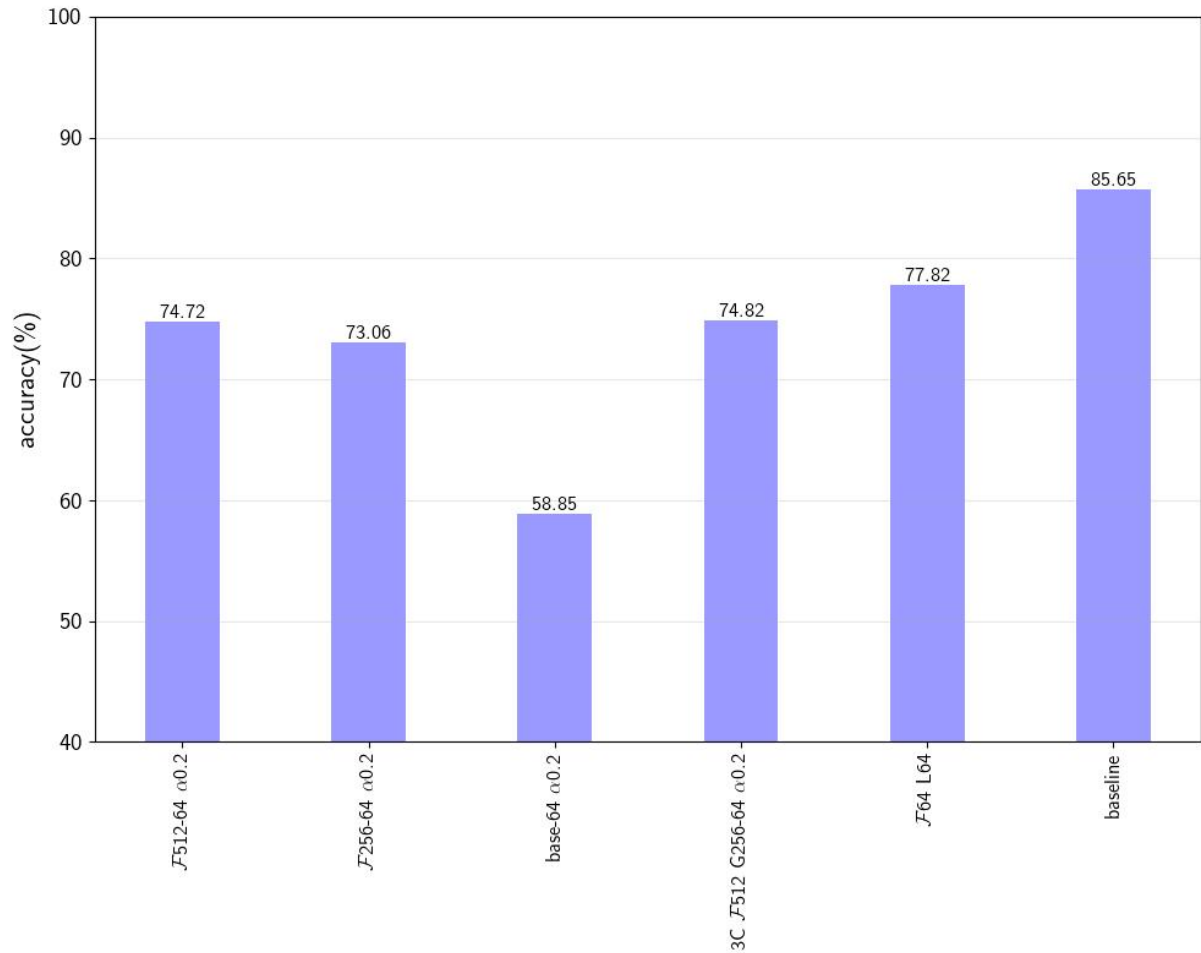


Figure 5.20: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs.

Knowledge distillation was also tested on a larger student model, $\mathcal{F}_{64} \text{ L64}$. The outcomes for the teacher-student combinations are differed from those observed with the $\mathcal{F}_{32} \text{ L32}$ student. For the larger student model, KD using a better teacher model trained on the same dataset, did not show any advantage at 60 epochs. In fact, a drop in performance can be observed.

5 Model compression

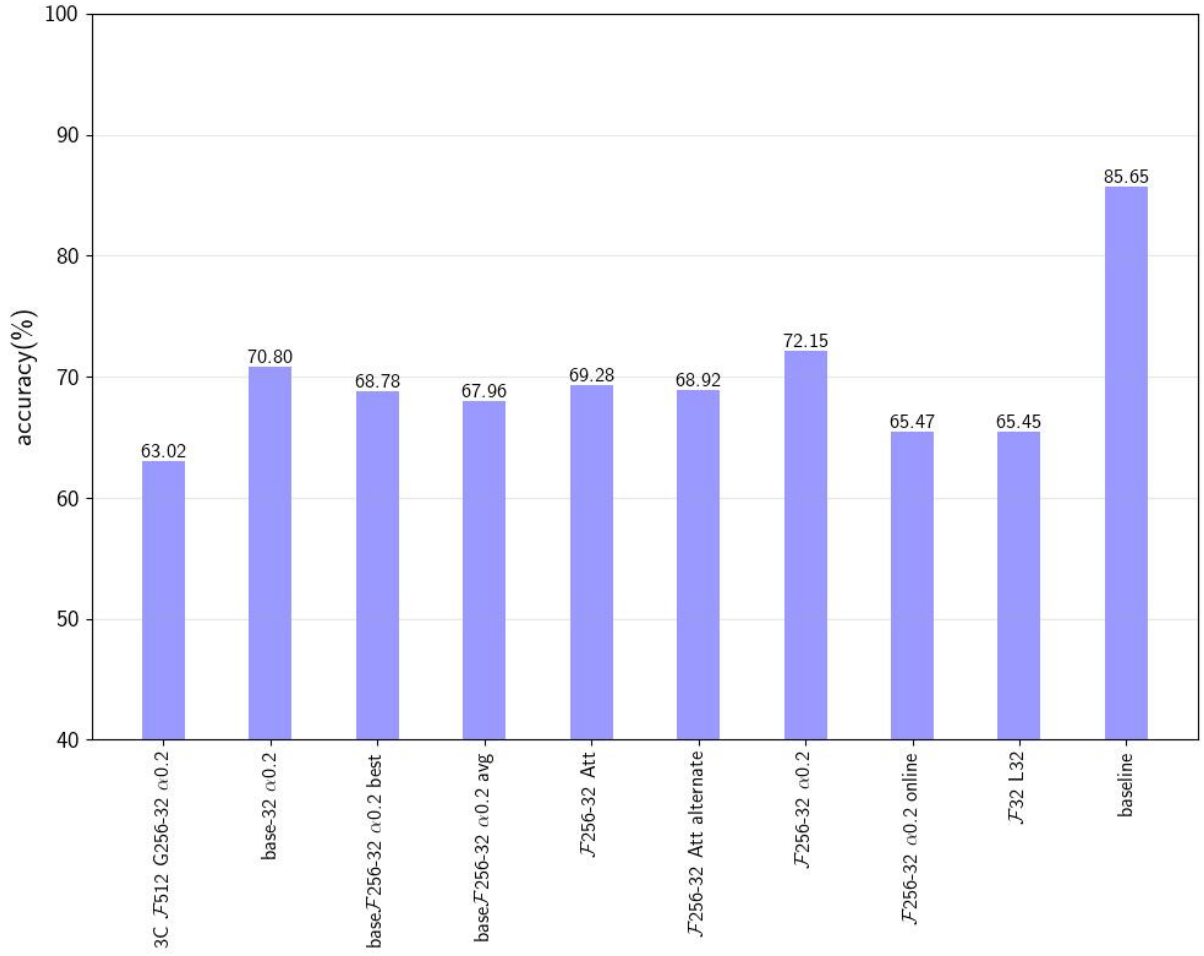


Figure 5.21: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the teachers have a slightly different architecture. All models underwent training for 60 epochs. The terms "best," "average," "alternate," and "online" refer to distinct training methodologies.

When evaluating all tested combinations over 60 epochs without data modifications, several conclusions can be drawn. Firstly, knowledge distillation can facilitate faster training. Secondly, alternate and online training methods did not show any advantage over offline training in the tested scenarios. Furthermore, the impact of better teachers trained on the same dataset was evident. However, the smaller the performance difference between student and teacher accuracy, the smaller the advantage gained, or it might even result in a disadvantage. It is also observed that the best model is not always the best teacher.

KD - 120 epochs

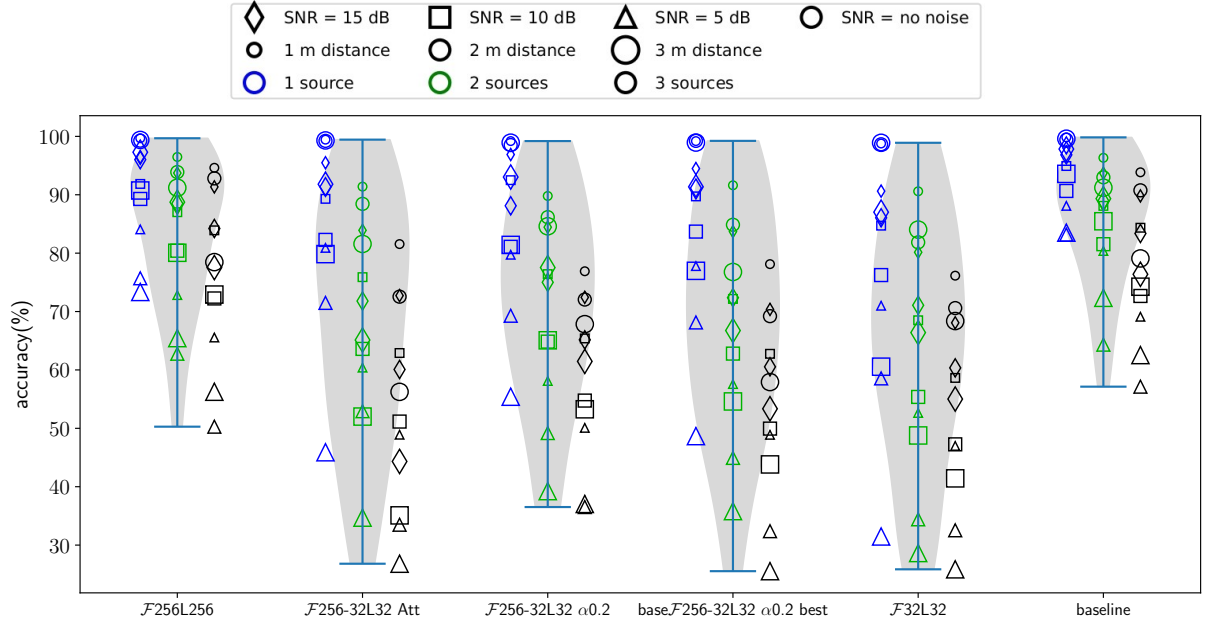


Figure 5.22: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and learned hidden weight factor (Att). All models underwent training for 60 epochs except for the teacher $F_{256}L_{256}$. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

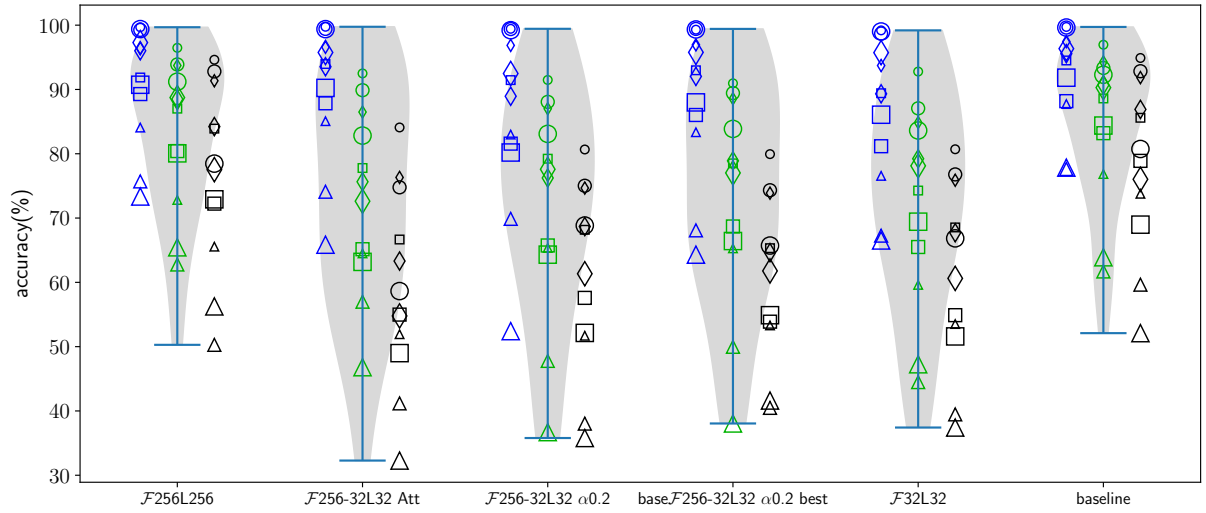


Figure 5.23: Test accuracy for BM-CLA models with the following name conventions, teacher-student learning method. Where the learning methods are best teacher (best), hidden weight factor (α) and learned hidden weight factor (Att). All models underwent training for 120 epochs. The grey boxplot illustrates the accuracy distribution for each model under test. The specific test conditions are detailed in the legend above the plot.

5 Model compression

Table 5.9: Average accuracies in % for all tested conditions for the models depicted in Figures 5.22 and 5.23.

conditions		
epochs	60	120
model	accuracy (%)	
$\mathcal{F}256L256$	79.40	83.73
$\mathcal{F}256\text{-}32L32$ Att	69.28	73.99
$\mathcal{F}256\text{-}32L32$ $\alpha 0.2$	72.15	73.06
base $\mathcal{F}256\text{-}32L32$ $\alpha 0.2$ Best	68.78	74.19
$\mathcal{F}32$ L32	65.45	73.41
baseline	85.65	85.13

For training the models in Table 5.9 until 60 epochs, knowledge distillation (KD) demonstrated an improvement over the student across all examples presented. Specifically, the model $\mathcal{F}256\text{-}32L32$ $\alpha 0.2$ exhibited a 6.7% higher accuracy compared to the model with the same architecture parameters, $\mathcal{F}32L32$, at this stage. However, when training continued to 120 epochs, the performance of $\mathcal{F}32L32$ improved more than expected, as detailed in Section 5.2.1. This further training resulted in the accuracy of the KD-enhanced models not being significantly better than those without KD. All three KD variants with different approaches yielded similar outcomes at 120 epochs. The student models derived from KD did not achieve performance levels close to those of their teacher models. Currently, the primary benefit of KD appears to be in accelerating the training process.

Smaller student dataset

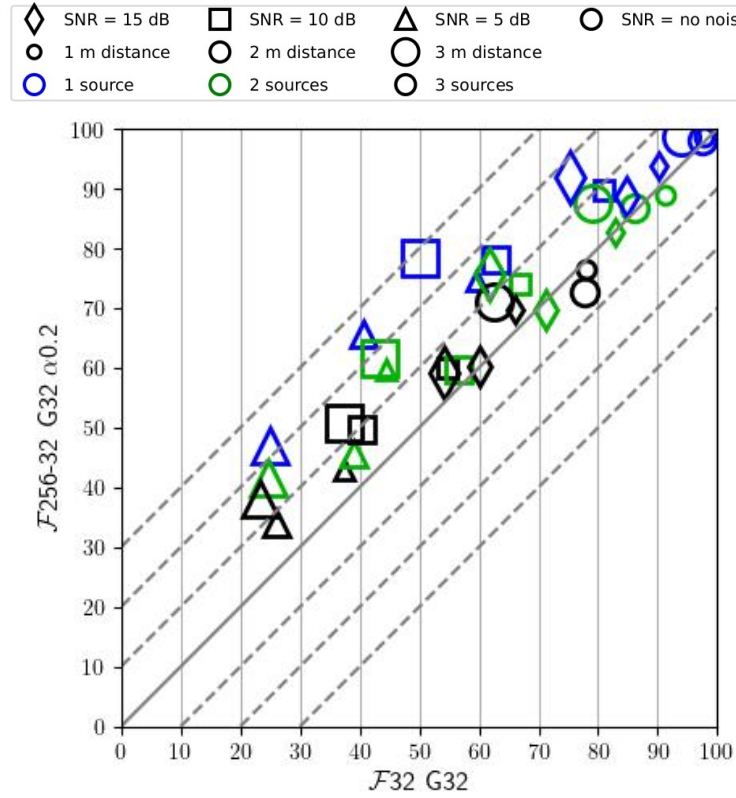


Figure 5.24: Comparison of test accuracy for the BM-CLA model using knowledge distillation with student $\mathcal{F}32G32$ trained on smaller dataset over 120 epochs.

The results presented in Table 5.9 were derived from models where both the teacher and student were trained on the same dataset. When knowledge distillation is performed using a more capable teacher model, trained on a relative larger dataset, the student model benefits significantly from this enhanced learning process. Specifically, a student model without KD achieved an average accuracy of 61.75%, whereas the same model with KD trained on the same dataset, reached an average accuracy of 70.05%. These findings indicate that leveraging a larger dataset for training the teacher model can provide substantial advantages to the student model, rather than restricting both to a smaller dataset (section 5.1.11). This approach holds promise for improving model performance through more extensive training of the teacher model.

5 Model compression

5.2.7 GRU

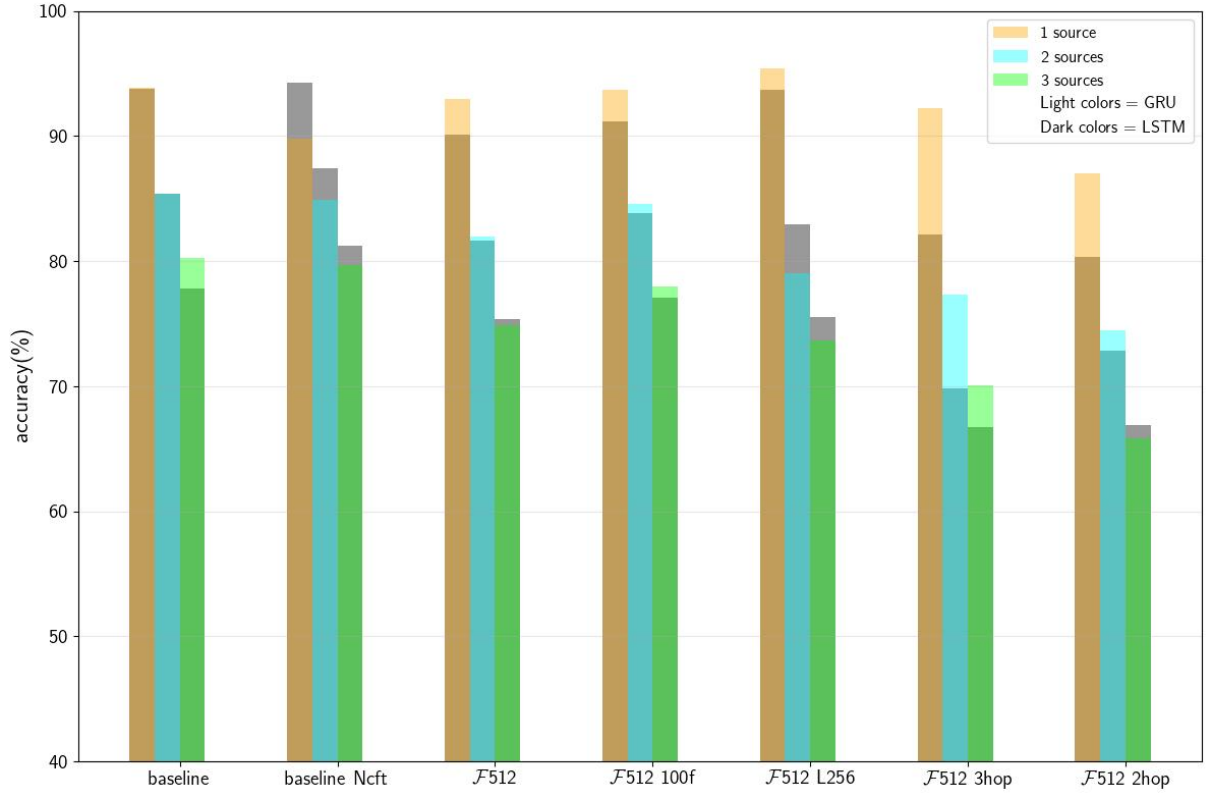


Figure 5.25: Average test accuracy for BM-CLA models with the adjusted temporal layer, across varying numbers of sources. Models utilising the GRU (G) are represented with light colors, while those employing the LSTM (L) are depicted with darker (grey) colors. The centralised baseline is depicted with amplitude normalisation Nc and Ncft.

Table 5.10: Average accuracies in % for all tested conditions for the models depicted in Figure 5.25. Bold numbers indicate better with LSTM.

conditions										
number of sources	1 source		2 sources		3 sources		1,2,3 sources			
temporal layer	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU	LSTM	GRU
model	accuracy (%)									MAC (millions)
baseline	93.79	93.90	85.39	85.41	77.78	80.28	85.65	86.53	32.75	32.25
baseline Ncft	94.28	89.77	87.45	84.87	81.20	79.73	87.64	84.79	32.75	32.25
F512	90.12	92.96	81.67	81.99	75.37	74.85	82.39	83.27	24.25	23.72
F512 100f	91.17	93.72	83.84	84.54	77.11	77.95	84.04	85.40	11.07	10.50
F512 (L/G)256	93.72	95.39	82.95	79.02	75.56	73.68	84.08	82.69	22.87	22.71
F512 3hop	82.11	92.20	69.79	77.35	66.77	70.10	72.89	79.88	8.00	7.81
F512 2hop	80.38	87.05	72.85	74.46	66.91	65.82	73.38	75.78	12.13	11.86

5 Model compression

Looking at Table 5.10, utilising a GRU instead of an LSTM appears to yield beneficial outcomes across most source scenarios. Reducing the number of nodes to 256 negatively impacts the GRU layer, whereas an improvement is observed in the LSTM. As depicted in Figure 5.25, all co-operative models exhibit improved performance under one-source conditions. Employing a GRU not only enhances accuracy but also reduces the network's size, thereby increasing computational efficiency. Looking at reducing the temporal layer by half, the LSTM seemed to improve while the GRU suffered in accuracy. In this configuration the LSTM wins in terms of accuracy and MACs. For the baseline with amplitude normalisation over channel, frequency and time, the LSTM is better in all source conditions.

5.3 Conclusion

The BM-CLA model could withstand a lot of compression without much negative effect on accuracy. Especially lowering the number of features had the most effect on lowering both MACs and bandwidth without dropping in accuracy. Once the number of features was lowered, other reductions seemed fatal for accuracy. The quantisation brought down the bandwidth by a factor of four. The quantisation effect was so small it could be ignored. Using an amplitude normalisation over channel, frequency, and time improved the accuracy and brought the impact of normalisation down to 64 bps. When looking at the MAC number, the BM-CLA seemed to be more efficient than the centralised model. Offline knowledge distillation using KL loss could be used for models with a significant difference in performance. Here, the important thing is the difference in training data and not just the difference in performance. Utilising a GRU had small improvement in performance for the same number of nodes as for the LSTM. Despite this occurrence, halving the number of nodes for LSTM seemed to be still better in some configurations.

6

Model expansion

In this chapter the focus will lay on DOA sound localisation accuracy performance with network size increase. The BM-CLA model will have small adjustments while the data remains unchanged during all the tested methods. Hereby the performance adjustments can be measured on the learning capabilities from the fixed dataset. The proposed models will be compared to the centralised baseline model without LC3+ codec. In the previous chapter 5, some of the expansion models are used as teachers for knowledge distillation. All models have conventions as described in chapter 4.

6.1 Methods

This section explains the methods tested for improving DOA performance, focusing solely on accuracy. First, changing the model's input with amplitude included, which is new for the BM-CLA model from previous work [6]. Next, tweaking hyper parameters of the network layers. Then adding intra-array convolutions. Lastly, adding inter-array amplitude information after the intra-array convolutions.

6.1.1 Input features

The input of the model within machine learning models, is one of the most important aspects, because it only can learn from what is going through. Many input features are used for binaural sound source localisation, from raw sequential audio signals, hand-crafted features (ILD, ITD) [51] to amplitude and phase of the STFT. Even gamma tone filters can be used to filter wanted frequencies from the incoming audio signal [52]. As seen in chapter 2 the STFT is often used in binaural hearing. The ILD and ITD, which are the binaural cues for sound source localisation, can be found in the amplitude and phase information of the STFT. To have a better understanding for how the model interacts with these features, the amplitude and phase were tested separately and together as input. Normalising the input enhances generalisation and accelerate the gradient descent process. The mean(m) and variance(v) over the first 1000 values in the middle of the time dimension of the trainloader batches are calculated. This was done for all tests.

$$input_{norm} = \frac{input - m}{\sqrt{v}} \quad (6.1)$$

6.1.2 Hyper-parameters

In the convolutional layers, when improving the amount of learnable kernels, the network can filter out more extinct features from the data. After the conv layers, the FC layer selects and processes the extracted features to make latent features. Having

6 Model expansion

more latent features can make learning more accurate. The temporal layer can adjust the receptive field to get temporal context during the classification. Note, that for enlarging the model size for the centralised system, the model size has no impact on the bandwidth, only the number of used microphones have impact. Next, by lowering dropout, the network can learn more distinct patterns and make full use of the connections. A drawback of lowering dropout is faster overfitting.

6.1.3 Circular padding

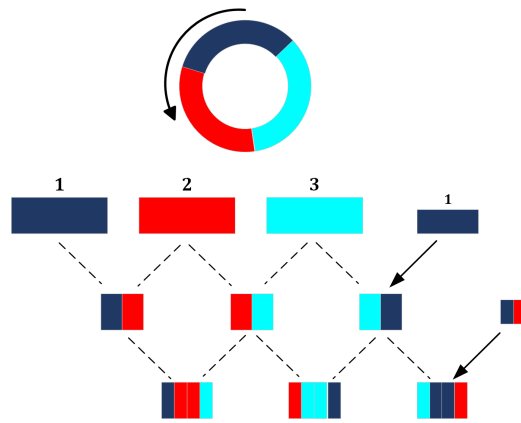


Figure 6.1: Intuitive drawing for circular padding with each color referring to the information from a microphone on the hearing device.

In the both the baseline (fig. 4.1) and the BM-CLA (fig. 4.2) make use of convolutions to combine the mic information. As the baseline has $2M_e - 1$ convolutions and the BM-CLA only has $M_e - 1$, this might cause the difference in performance. When adding more convolutions to the BM-CLA, more connections in between the mic info might occur. To keep in mind is that the inter-aural information is not shared through the convolutions in the BM-CLA model. With the use of circular padding, more convolutions can be taken between the intra-ear microphones. In figure 6.1, a visualisation shows that not only information from mics $m_{1,2}$ and $m_{2,3}$ is combined but also the combination $m_{1,3}$. By every additional comparison of the first and last mic in the input, an extra convolution is created. With the extra convolution more intra-mic information might be collected to form the embedded features.

6.1.4 Skip connections

In the BM-CLA model the embedded features consist out of intra-mic information only. The baseline model on the other hand has access to information from both intra- and inter-mic relations. This is done by the $M - 1$ convolutional layers. To mimic this in the feature model, a convolution between the amplitude of the first mic from each ear is taken. This provides extra ILD information for combining the inter-mic information. These results are then going through a fully connected layer of 512 nodes. Later these features are added to the ear-features and going through a 512 FC instead of a 1D convolution. As

temporal layer, a GRU is chosen. The rest of the network is similar to the $\mathcal{F}512$ reference model.

6.2 Results and discussion

In this section of the thesis, the obtained results and insights will be discussed. The tested models may have different hyper parameters. These will be given in the subscription of the images. The focus will lay on model accuracy performance with enlarging the model size. An overview of the compressed and expanded models can be found in the appendices fig. 14, fig. 15, fig. 16. Models can be chosen according to constraints and accuracy expectations.

6.2.1 Input features

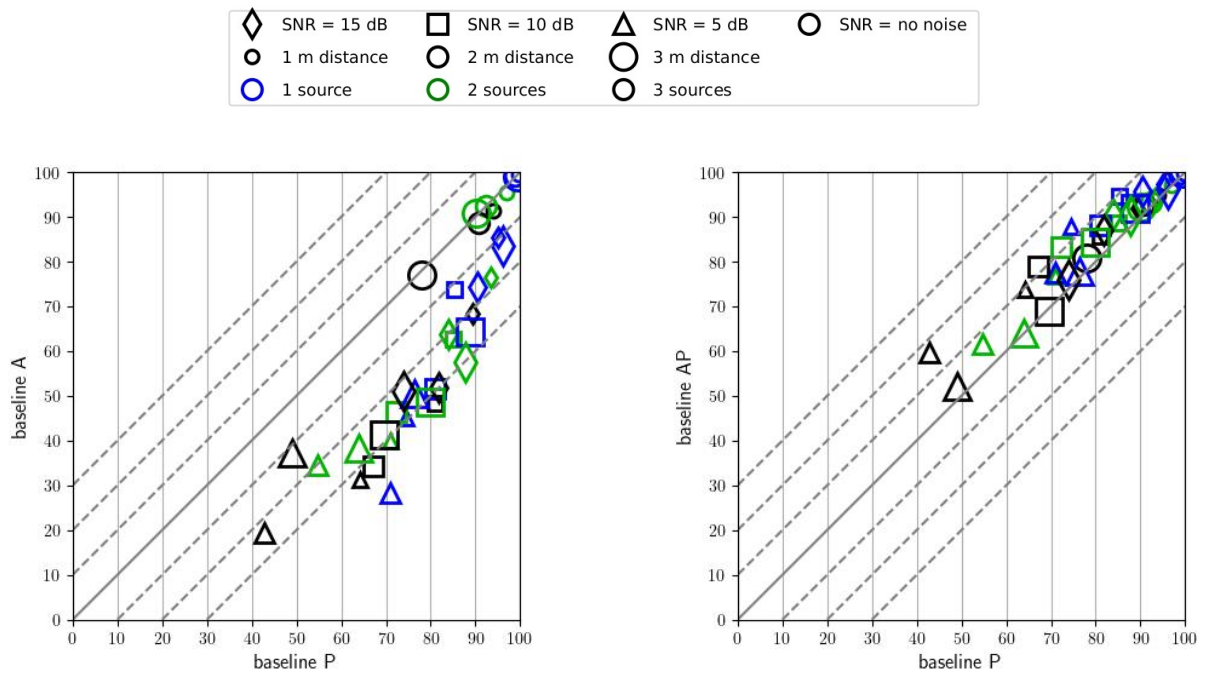


Figure 6.2: Comparison of test accuracy under various conditions for the baseline model using phase (P) input versus the baseline model using amplitude (A) input over 120 epochs.

Figure 6.3: Comparison of test accuracy under various conditions for the baseline model using phase (P) input versus the baseline model using both amplitude (A) and phase (P) inputs over 120 epochs.

6 Model expansion

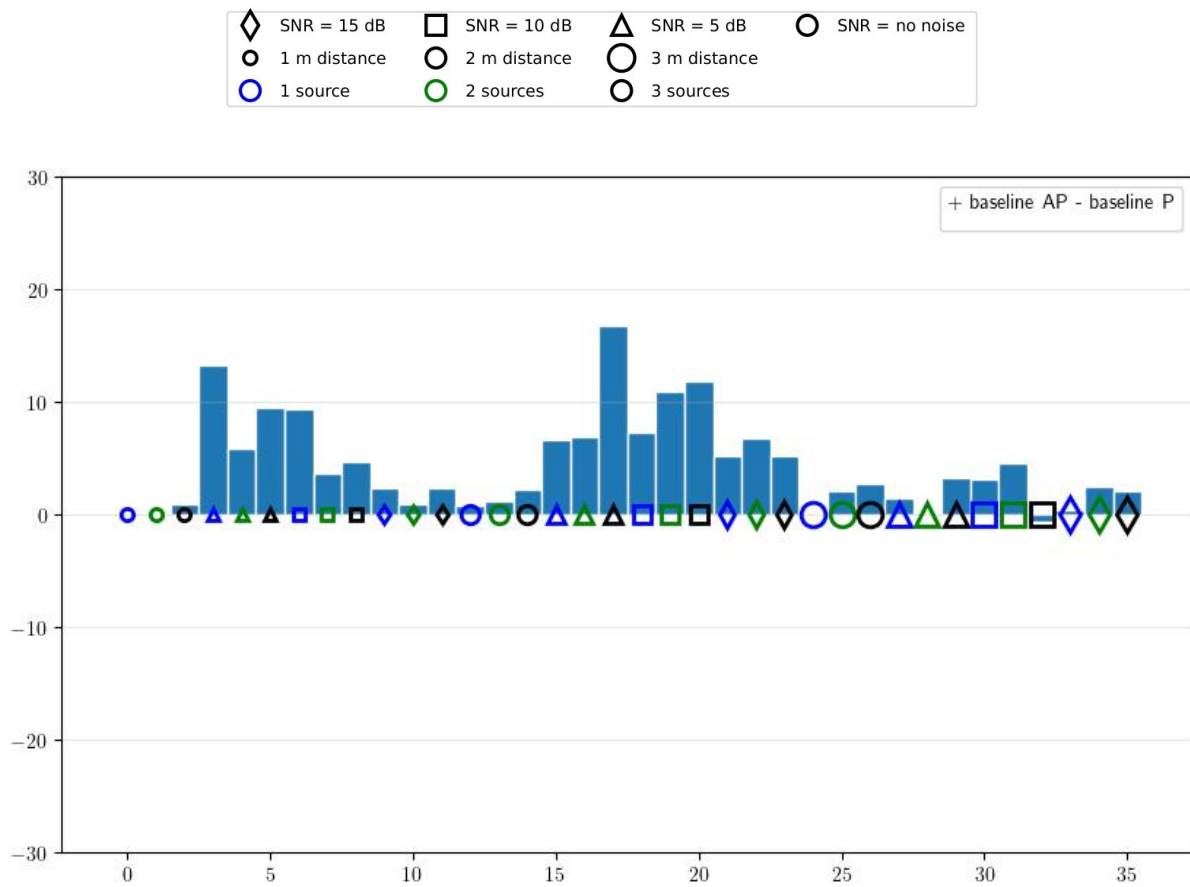


Figure 6.4: Difference in test accuracy between the baseline model using phase (P) and amplitude (A) inputs (improvement represented on the positive axis) and the baseline model using phase (P) inputs alone (improvement represented on the negative axis) over 120 epochs.

6 Model expansion

Table 6.1: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline A. The averages are taken for all the conditions with the specified constraint.

baseline A		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	92.61	20.85
SNR (15 dB)	68.06	
SNR (10 dB)	52.37	
SNR (5 dB)	36.07	
sources (1 source)	71.21	7.31
sources (2 sources)	62.31	
sources (3 sources)	53.31	
distance (1 m)	68.16	4.55
distance (2 m)	57.06	
distance (3 m)	61.60	

Table 6.2: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline P. The averages are taken for all the conditions with the specified constraint.

baseline P		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	93.30	11.48
SNR (15 dB)	87.97	
SNR (10 dB)	78.83	
SNR (5 dB)	63.05	
sources (1 source)	87.99	5.94
sources (2 sources)	80.93	
sources (3 sources)	73.45	
distance (1 m)	85.76	3.63
distance (2 m)	77.21	
distance (3 m)	79.40	

6 Model expansion

Table 6.3: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for baseline AP. The averages are taken for all the conditions with the specified constraint.

baseline AP		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	94.45	9.31
SNR (15 dB)	91.04	
SNR (10 dB)	84.91	
SNR (5 dB)	70.13	
sources (1 source)	92.17	5.58
sources (2 sources)	84.70	
sources (3 sources)	78.52	
distance (1 m)	90.18	3.75
distance (2 m)	84.01	
distance (3 m)	81.20	

The general trend observed for accuracy, when using amplitude as the sole input, indicates that localisation performance remains excellent under zero noise conditions across all scenarios. However, under varying sound conditions, the accuracy distribution decreases as the complexity of the sound environment increases. Factors such as a greater number of sources, lower SNR, and increased distance contribute to this decline. The average accuracy is 62.27%, with the standard deviation (σ) being most pronounced for SNR at 20.85, demonstrating a clear degradation in performance with increasing noise, as detailed in Table 6.1. Similarly, variations in the number of sources result in a standard deviation of 7.3. In contrast, distance has a small impact on accuracy, with a standard deviation of 4.55.

The performance of the baseline model using only phase as input demonstrates an overall average accuracy of 80.79%. Similar patterns observed with the amplitude-only model are evident here, with a notable difference in behavior under higher SNR conditions, where the phase model significantly outperforms the amplitude model. The SNR continues to have the most substantial impact on accuracy, with a standard variation of 11.48. For the number of sources and distances, the standard deviations are 5.94 and 3.63, respectively. The tested average values per condition for the baseline phase (P) model are presented in Table 6.2. As illustrated in Figure 6.2, it is clear that phase provides more relevant information than amplitude for binaural sound localisation in the baseline model.

Figure 6.4 illustrates the differences in accuracy across all test conditions for the models with P and AP. The inclusion of amplitude in the input data notably enhances performance in scenarios with SNRs of 15, 10, and 5 dB. Interestingly, conditions with a distance of 3 meters do not show as significant an improvement as those with a distance of 2 meters. Conditions with no noise show minimal improvement however, the accuracy was already relatively high in both the amplitude-only (A) and phase-only (P) models. No decline in accuracy is observed when amplitude is added to the phase as input for the baseline model and the overall accuracy is set to 85.13%.

6 Model expansion

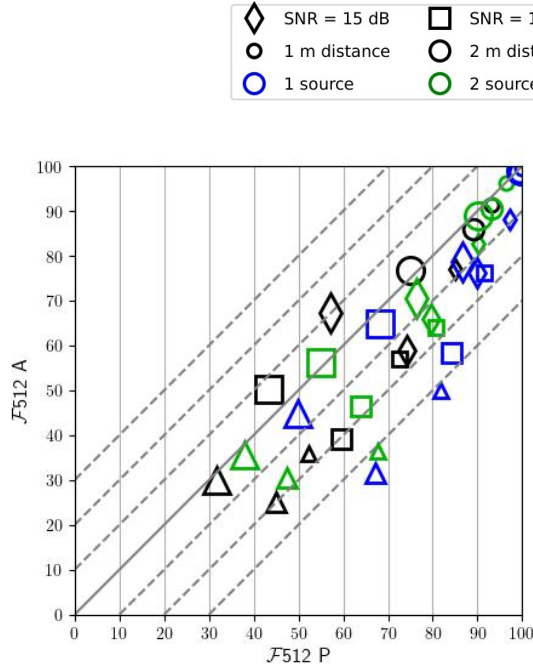


Figure 6.5: Comparison of test accuracy under various conditions for the BM-CLA model using phase (P) input versus the BM-CLA model using amplitude (A) input over 120 epochs.

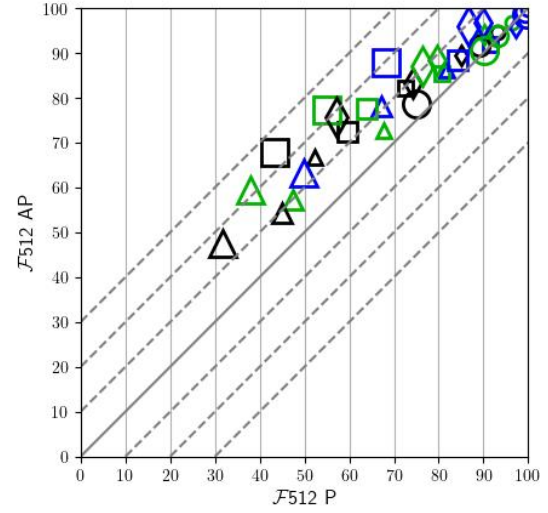


Figure 6.6: Comparison of test accuracy under various conditions for the BM-CLA model using phase (P) input versus the BM-CLA model using both amplitude (A) and phase (P) inputs over 120 epochs.

Table 6.4: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ A. The averages are taken for all the conditions with the specified constraint.

$\mathcal{F}512$ A		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	91.84	20.82
SNR (15 dB)	73.92	
SNR (10 dB)	56.95	
SNR (5 dB)	35.49	
sources (1 source)	72.13	5.85
sources (2 sources)	63.65	
sources (3 sources)	57.87	
distance (1 m)	71.19	5.05
distance (2 m)	58.93	
distance (3 m)	63.53	

6 Model expansion

Table 6.5: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ P. The averages are taken for all the conditions with the specified constraint.

$\mathcal{F}512$ P		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	92.83	14.70
SNR (15 dB)	81.78	
SNR (10 dB)	68.78	
SNR (5 dB)	53.46	
sources (1 source)	84.59	8.10
sources (2 sources)	73.21	
sources (3 sources)	64.83	
distance (1 m)	84.03	8.08
distance (2 m)	74.36	
distance (3 m)	64.24	

Table 6.6: Average accuracies in % for the given conditions with the overall standard deviation σ for the corresponding changes in test parameters for $\mathcal{F}512$ AP. The averages are taken for all the conditions with the specified constraint.

$\mathcal{F}512$ AP		
condition	accuracy (%)	standard deviation (σ)
SNR (no noise)	93.74	10.99
SNR (15 dB)	89.57	
SNR (10 dB)	81.19	
SNR (5 dB)	65.04	
sources (1 source)	90.12	6.04
sources (2 sources)	81.67	
sources (3 sources)	75.37	
distance (1 m)	87.92	4.28
distance (2 m)	81.75	
distance (3 m)	77.49	

The BM-CLA model with amplitude-only input exhibits an overall accuracy of 65.55%, showing a trend similar to the baseline model. The most significant difference is observed in the 15 dB conditions, where the BM-CLA model performs 5.86% better than the baseline (Table 6.1, Table 6.4). A more notable difference is observed with phase-only input for the BM-CLA model. As illustrated in Figure 6.5, not all noisy conditions show as much improvement as seen in the baseline model. Specifically, conditions with a 3-meter distance combined with noise show similar performance to the amplitude-only input. However, for 1-meter and 2-meter distances, improvements are substantial, with increases of 12.84% and 15.43%, respectively. The overall accuracy for the phase-only input is 74.21%.

6 Model expansion

Combining amplitude with phase input yields a positive impact, achieving an average accuracy of 82.39%. Unlike the baseline P model, which already exhibits strong performance, the BM-CLA model demonstrates significant potential for improvement. As depicted in Figure 6.7, the most notable improvements occur under more challenging acoustic conditions, such as a 3-meter distance, SNR levels of 5 and 10 dB, and scenarios with three sources. These conditions show respective improvements of 13.25%, 11.58%, 12.41%, and 13.25%. When these challenging conditions are combined, improvements exceeding 20% are achieved.

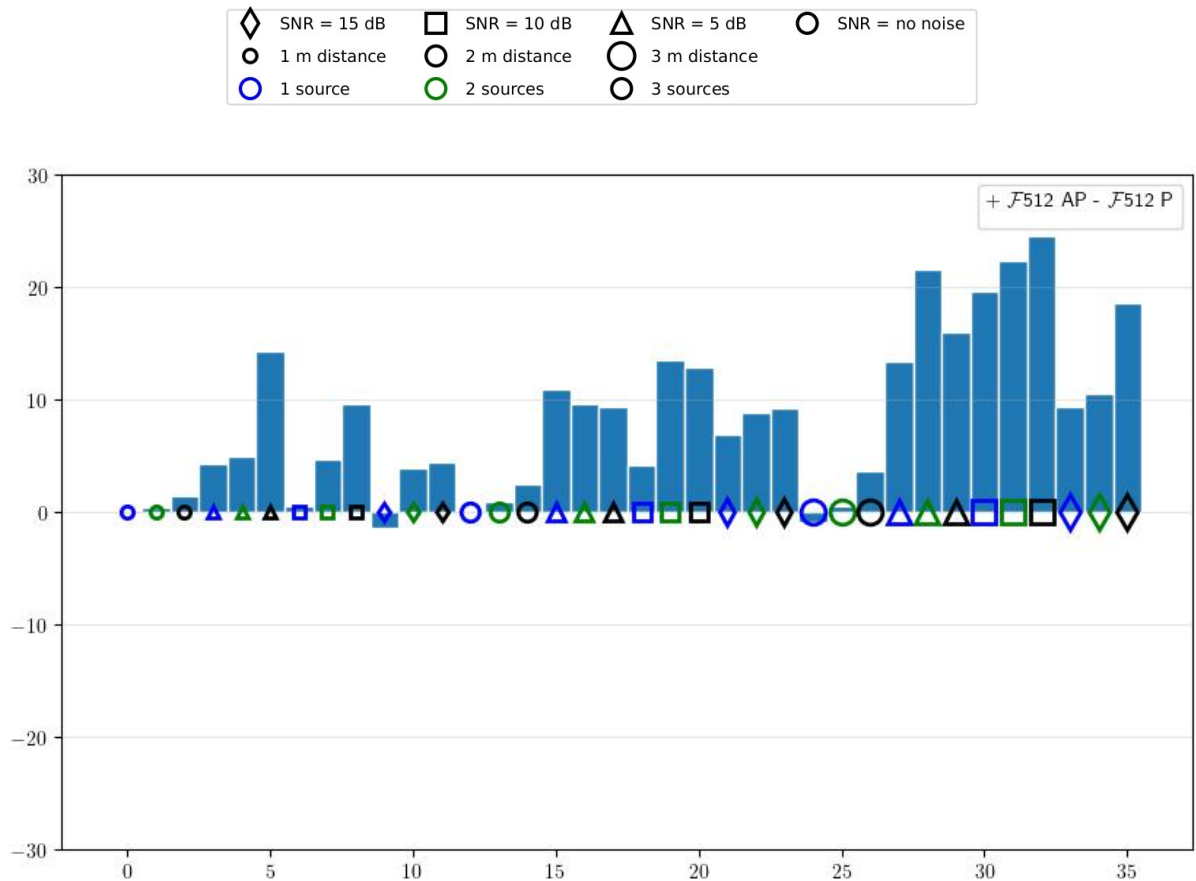


Figure 6.7: Difference in test accuracy between the BM-CLA model using phase (P) and amplitude (A) inputs (improvement represented on the positive axis) and the BM-CLA model using phase (P) inputs alone (improvement represented on the negative axis) over 120 epochs.

6.2.2 Circular padding

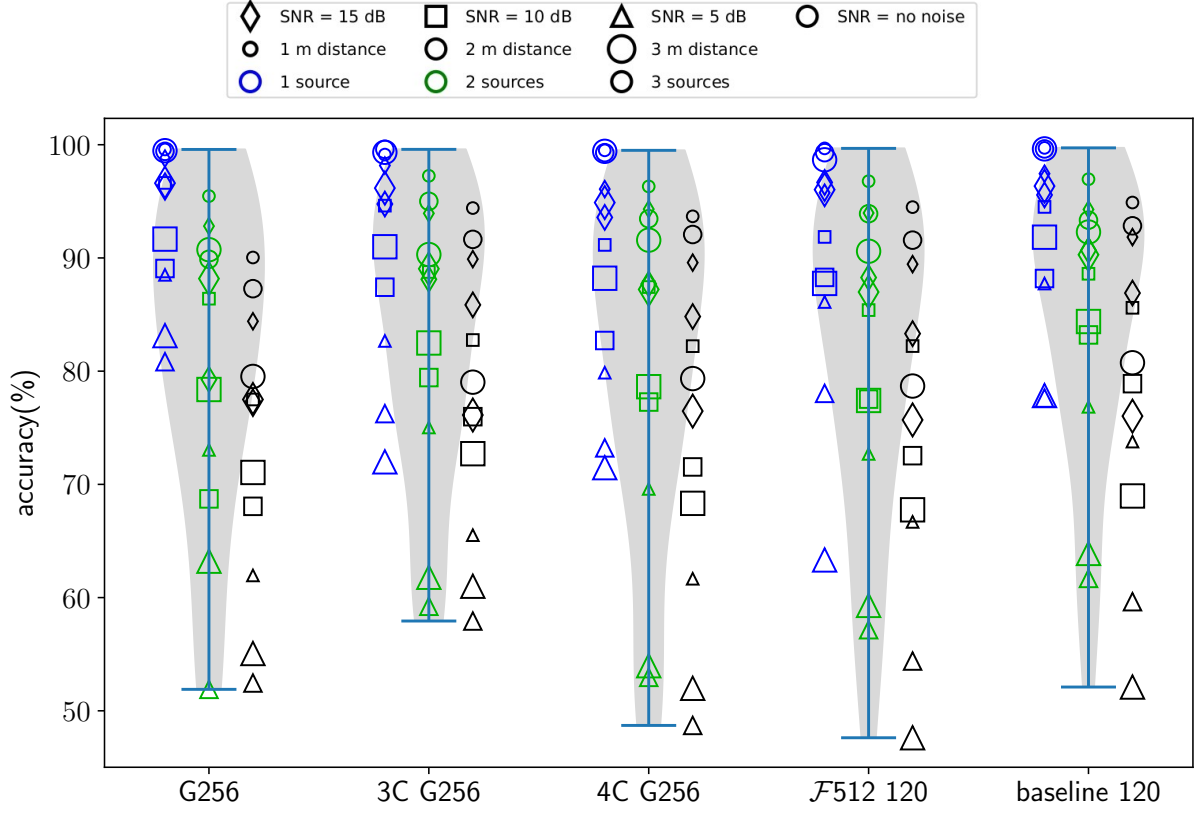


Figure 6.8: Test accuracy for the models utilising a GRU of 256 (G256), $\mathcal{F} = 512$, with circular padding and C convolutions at the ear-device is depicted. The boxplot in grey illustrates the accuracy distribution for each model. The test conditions are detailed in the legend above the plot.

Table 6.7: Average accuracies in % for all tested conditions for the models depicted in Figure 6.8.

model	accuracy (%)
G256	82.23
3C G256	84.01
4C G256	81.67
$\mathcal{F}512$	82.39
baseline	85.13

Circular padding was tested for models with GRU256, incorporating one and two additional convolutions. The G256 models were trained for only 60 epochs. During training, the minimal difference observed in the validation loss curve was not interpreted as a significant improvement. As discussed in Section 5.2.1, it is shown that the validation loss curve provides only blurred indications towards the end of the training process. The performance results are depicted in the violin plot

6 Model expansion

in Figure 6.8, comparing these models against the baseline model and the BM-CLA $\mathcal{F}512$. For the first two models, the one-source conditions undergo a performance decrease with the addition of an extra convolution, whereas the multi-source conditions showed improved accuracy. However, the overall accuracy increased by only 1.68%. Adding another convolution, resulting in four convolutions before extracting broadband latent features, did not provide any advantage. Nearly all tested conditions experienced a slight decrease in performance, culminating in a 2.24% drop in accuracy. When comparing the GRU256 models to the reference models BM-CLA $\mathcal{F}512$ and the baseline, the baseline model still performed better overall. Notably, the model with three convolutions marginally outperformed the $\mathcal{F}512$ reference model.

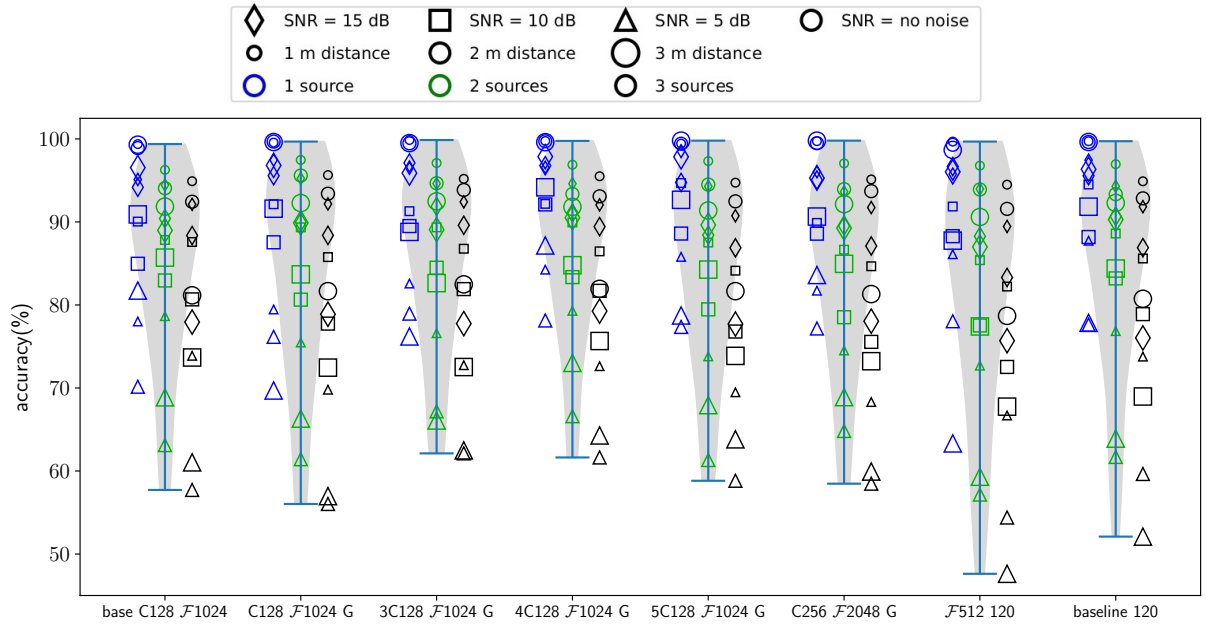


Figure 6.9: Test accuracy was evaluated for models with varying configurations of convolutional layers, fully connected (FC) layers, and temporal layers. For example, 3C128 represents a model with 3 convolutional layers, each containing 128 filters. The number of nodes in the FC layer is identical to that in the temporal layer and Conv1D layer. The notation 'G' denotes the use of a GRU instead of a LSTM. All models were trained until minimal validation loss, except for the reference models BM-CLA $\mathcal{F}512$ and the baseline model, which were trained for 120 epochs. The boxplot in grey illustrates the accuracy distribution for each model. The test conditions are detailed in the legend above the plot.

6 Model expansion

Table 6.8: Average accuracies in % for all tested conditions for the models depicted in Figure 6.9.

model	accuracy (%)
base C128 $\mathcal{F}1024$ L	85.11
C128 $\mathcal{F}1024$ G	84.76
3C128 $\mathcal{F}1024$ G	85.88
4C128 $\mathcal{F}1024$ G	86.86
5C128 $\mathcal{F}1024$ G	85.24
C256 $\mathcal{F}2048$ G	84.95
$\mathcal{F}512$	82.39
baseline	85.13

When examining the expanded baseline, base C128 $\mathcal{F}1024$ L, alongside the baseline itself, negligible disparities in accuracy are discernible. For this dataset, augmenting the baseline network for this dataset appears unwarranted. Conversely, the BM-CLA model demonstrates slight enhancements, averaging around 2% across all conditions. Particularly noteworthy is the observed improvement of 4.16% in the context of 3-meter distances. Introducing an additional convolutional layer, as in the case of 3C128 $\mathcal{F}1024$ G, yields marginal enhancements averaging approximately 1% across most conditions. Notably, the improvement is more pronounced in the context of 5 dB SNR, with an increase of 3.74%. The 4C128 is almost 3% better than the 3C128 model in 3-meter, 1-source and 5dB SNR conditions, resulting in a total average accuracy of 86.86%. With this accuracy, it even outperformed the baseline without LC3+ codec. Especially for the lower SNR and 3-meter sound conditions.

To explore the impact of further convolutions, the examination extends to models incorporating 5 convolutions for intra-array information processing. However, the introduction of two additional convolutions leads to a decline in average performance across nearly all conditions. It is worth noting that while scaling up the network size, outcomes may vary for larger datasets, albeit this aspect remains unexplored in this study. The largest network examined, denoted as C256 $\mathcal{F}2048$ G, exhibits performance similar to the $\mathcal{F}1024$ networks. This could potentially stem from overfitting (section 6.2.5) as later evidenced in the training curves. Nevertheless, the performance of the larger model still surpasses that of the reference BM-CLA model and closely aligns with the baseline.

6.2.3 Skip connection

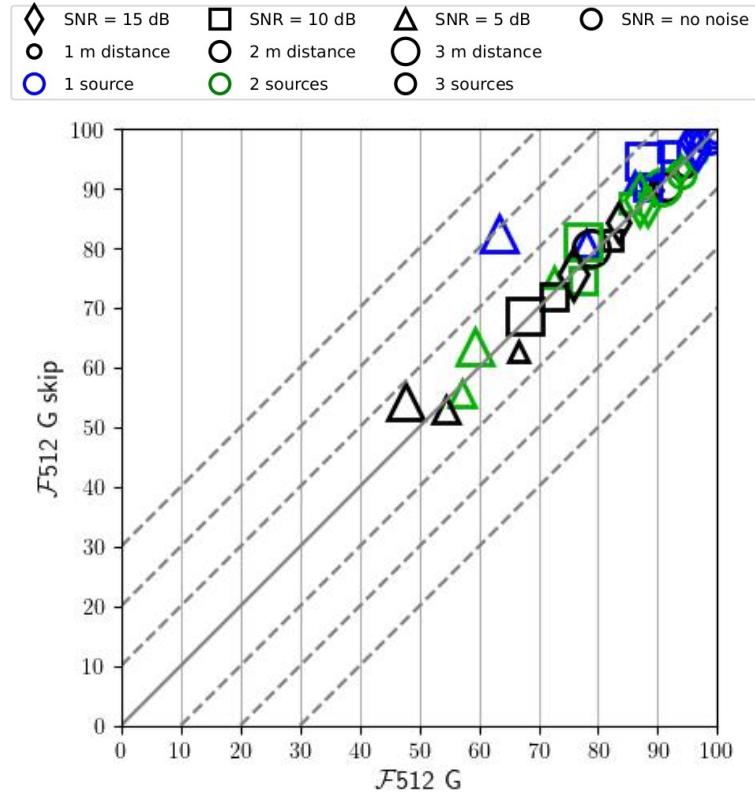


Figure 6.10: Comparison of test accuracy for the BM-CLA model equipped with a GRU and a skip connection.

When examining the test results at 120 epochs, no significant differences were observed. The only difference is better performance in noisy conditions for the one-source condition. Consequently, further testing was discontinued. This minimal difference in performance is likely due to the effective sharing of inter-aural amplitude information in the $\mathcal{F}_{512} G$, achieved through the use of broadband features.

6.2.4 Dropout

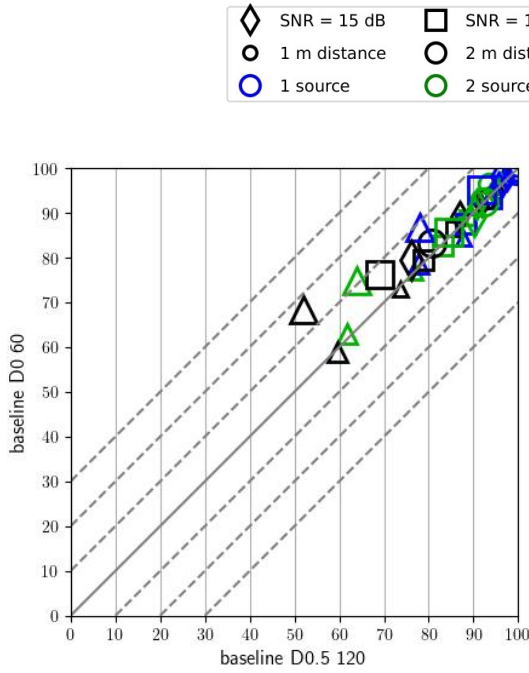


Figure 6.11: Comparison of test accuracy for the baseline model with dropout rates $D=0$ trained over 60 epochs and $D=0.5$ trained over 120 epochs.

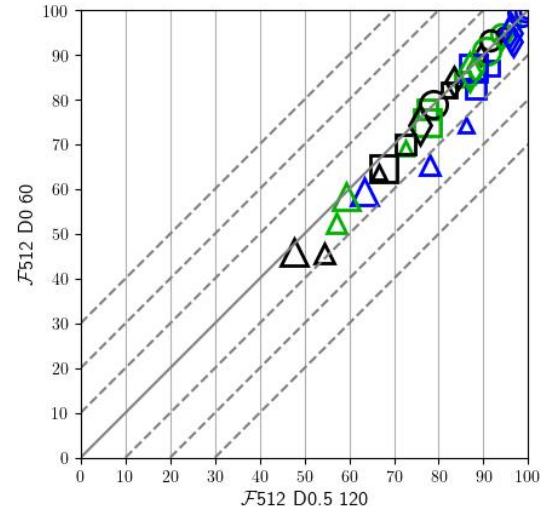


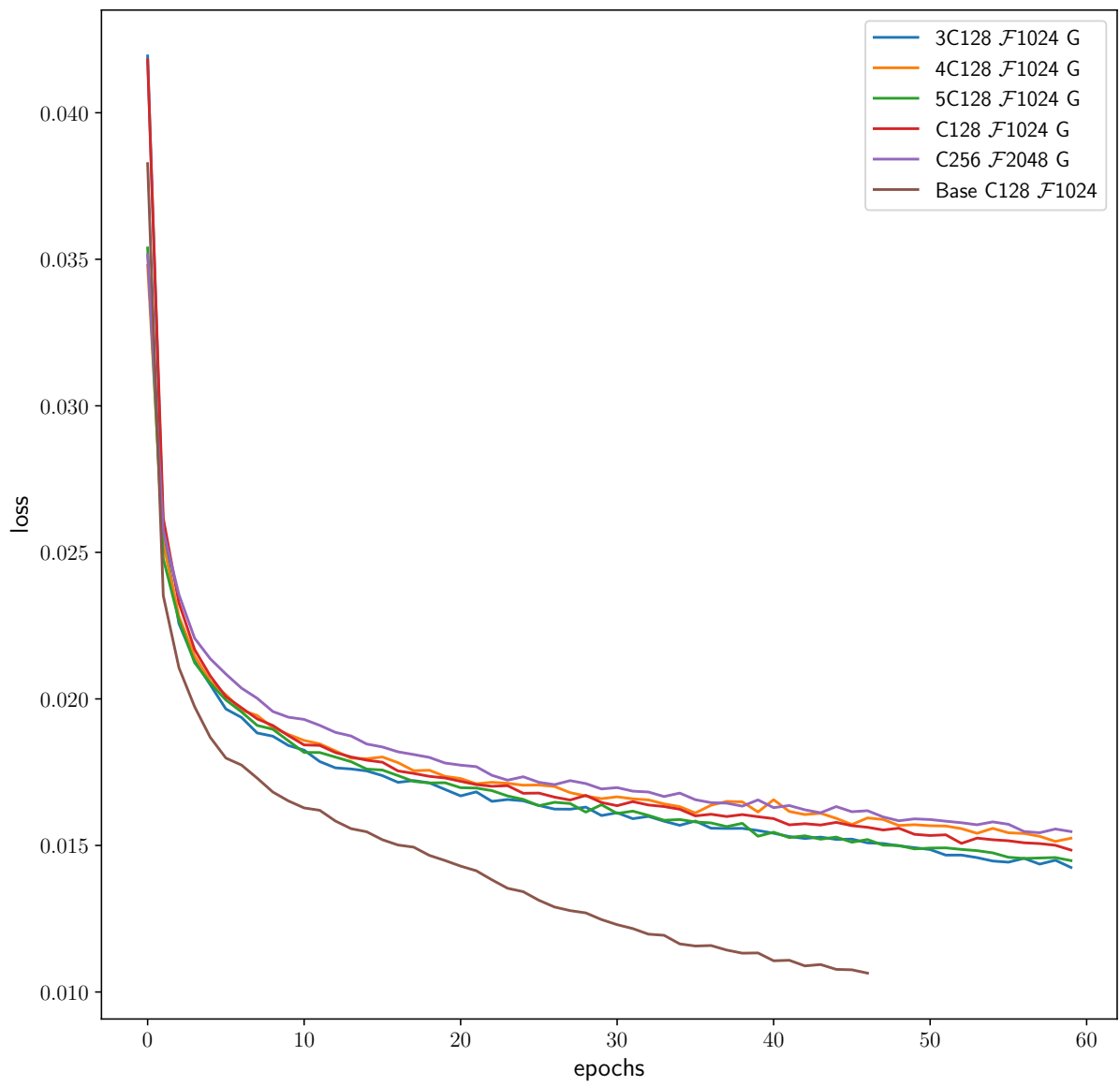
Figure 6.12: Comparison of test accuracy for the BM-CLA model with dropout rates $D=0$ trained over 60 epochs and $D=0.5$ trained over 120 epochs..

Table 6.9: Average accuracies in % for changes in dropout rates from Figures 6.12 6.11.

condition		
dropout	0	0.5
model	accuracy (%)	
$\mathcal{F}512$	80.27	82.39
baseline	86.77	85.13

As shown in Table 6.9, training the baseline model with a dropout rate of 0 over 60 epochs yields better performance than training it for 120 epochs with a dropout rate of 0.5. Conversely, the BM-CLA reference model exhibits the opposite trend. Consequently, the baseline model requires less computation time and energy to achieve full training. Furthermore, Figures 6.11, 6.12 indicate that dropout has the most significant impact under low SNR conditions, which are the most challenging for generalisation. This is expected, as noise can either simplify or complicate the problem of source localisation.

6.2.5 Overfitting



6 Model expansion

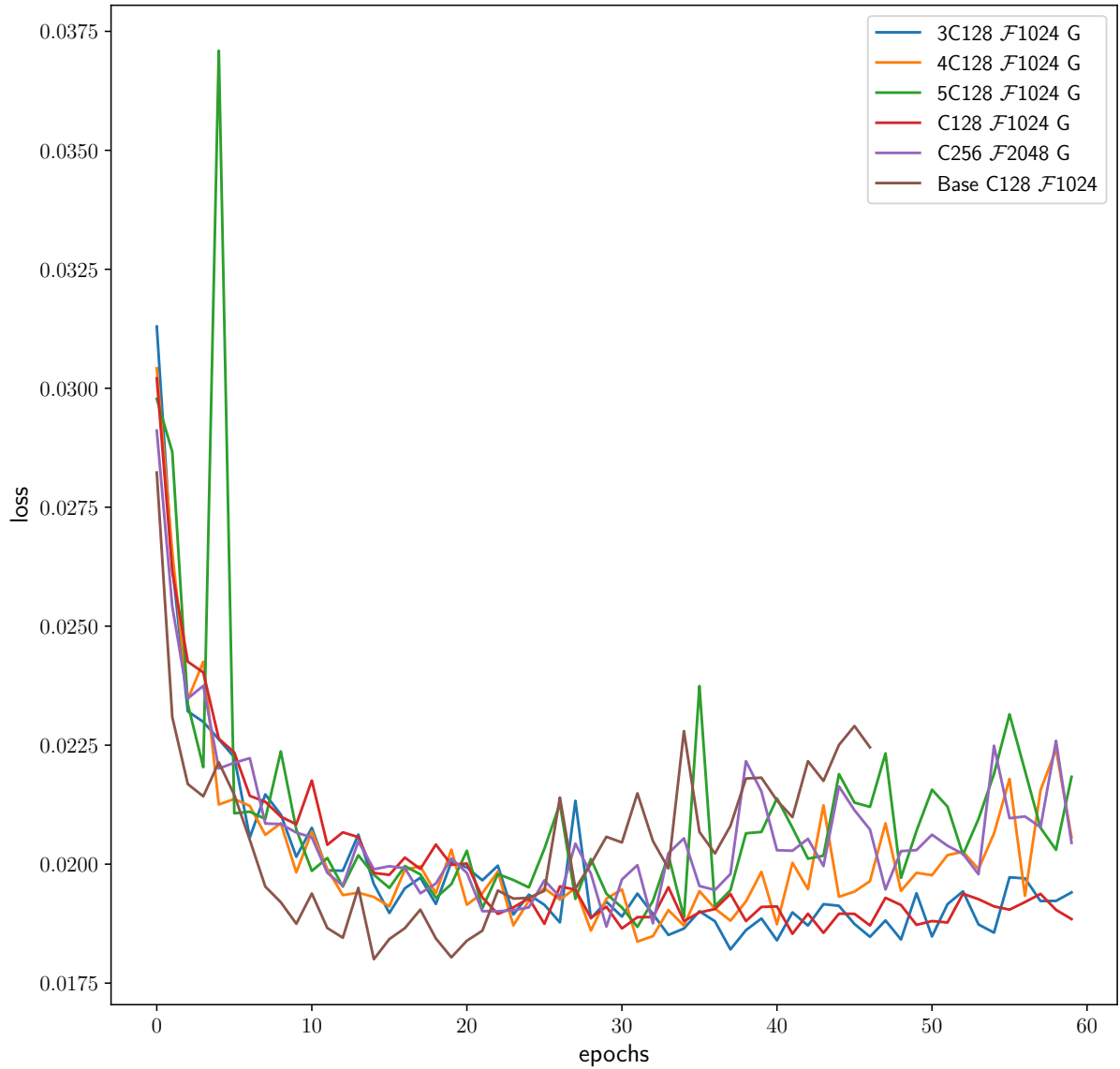


Figure 6.13: Binary cross entropy loss curves during training on the training data are displayed above the validation loss curves for models exhibiting signs of overfitting. The nomenclature follows a convention where, for instance, '3C128' indicates three convolutions with 128 filters per convolution. Similarly, the symbol $\mathcal{F}$ denotes the number of nodes in the embedded features, matching the size of the temporal layer. The letter 'G' signifies the utilisation of GRU as the temporal layer, instead of a LSTM.

The models with enlarged layer sizes exhibit indications of overfitting. The training loss curves, represented by binary cross-entropy (BCE) loss, display a decline characterised by an elbow-shaped curve. In contrast, the validation curves begin to rise in loss after a certain point. This upward trend indicates that the models are becoming overfitted, as the performance on unseen data deteriorates while the training loss continues to decrease. Consequently, these models were not trained

6 Model expansion

for the full duration of 120 epochs, unlike their smaller counterparts. The resolution of this overfitting phenomenon could potentially be attained through training on a more extensive dataset. However, this remedy necessitates a longer duration of training. Lastly, adding more dropout might make longer training possible resulting in better accuracy.

6.3 Conclusion

The combined use of amplitude and phase information results in significantly better accuracy performance compared to their separate utilisation. The amplitude information from the inter-ear microphones integrates effectively with the broadband features. Furthermore, the combination of intra-ear information is successfully enhanced by incorporating an additional convolutional layer. Additional skip connections, however, only has increased performance for one-source conditions. For the baseline model, lower dropout rates and shorter training durations appear beneficial. In contrast, the BM-CLA model performs better with extended training periods.

Conclusion

The primary aim of this master's dissertation was to explore methods to reduce energy for the sound broadband mixing co-operative sound localisation algorithm. Throughout the research, the five objectives were successfully achieved, leading to significant reduction and better variants of the reference model.

To begin with, the study explored a range of methods to reduce the bandwidth during operations for doing sound localisation on BTE hearing aids. Starting from good to better solutions, reducing the time resolution by increasing hop length or stride successfully reduced bandwidth. Lowering the number of transmitted features, quantising the data and normalising the amplitude over all dimensions became the best solution. These modifications minimally affected or even improved the model's performance. The normalisation also improved the centralised approach.

Next, to improve models minimised in size, multiple training schemes and teacher-student combinations for knowledge distillation were tested. These techniques showed promising results for applying a smaller model during inference deployment. The use of knowledge distillation could be introduced in the broader field of source localisation.

New approaches for improving model performance on the same dataset were experimented with success. By using extra convolutions, enhanced information sharing on the intra-ear level could be realised. Hereby, opening solutions for better models trained on bigger datasets. These models could then be used as teachers in KD.

It is important to mention that the centralised model equipped with an LC3+ codec still outperformed the aforementioned solutions. Particularly under more challenging listening conditions, with low SNR, the BM-CLA model appeared to reach the threshold of its learning capacity when lowering the features and temporal resolution simultaneously.

Further investigations could explore methods to enhance the baseline performance of the centralised approach equipped with a codec. With the fast evolutions of codecs other codecs like residual vector quantisation could be tested. Training with frequency selection by using a gamma tone filter might filter out the most used frequencies. Outstanding models with no limit in size and data can be explored for better knowledge distillation. Additionally, implementing generative adversarial training schedules could yield promising results, further improving the efficacy of the models.

Ethical and social reflection

Improving sound localisation software, and thus enhancing energy efficiency of the hearing aids device can be linked to several Sustainable Development Goals (SDG).

By enhancing hearing aid technology, individuals with hearing impairments can experience improved auditory perception, leading to better overall health, well-being, and quality of life. Enhanced sound localisation improves communication, safety, and social interaction for hearing aid users. This aligns with the SDG 3 which stands for good health and well-being.

The study contributes to SDG 9 for industry, innovation. When developing more energy-efficient hearing aids, the research supports sustainable consumption and production patterns. This reduces the environmental footprint of hearing aid usage, promoting sustainability in the production and use of medical devices. This fosters industrial development in medical technology.

6 Conclusion

By improving the efficiency of the software for sound localisation, lower energy consumption is achieved. Thereby smaller batteries and so lower amount of natural resources are exploited. Therefore this dissertation supports the guidelines of SDG 12 for sustainable production patterns.

Finally, this research contributes to the accessibility of hearing aids, helping to reduce the inequalities faced by individuals with hearing impairments. Affordable and efficient hearing aids can make assistive technology more accessible to diverse populations, including those in low-income settings. These developments reflect to the SDG 10 for reduced inequality.

In conclusion, the master's dissertation not only contributes to scientific and technological advancements but also holds positive ethical and social implications. Better hearing aid software helps to achieve multiple SDGs, particularly those related to the human well-being and hearing aid sustainability.

References

- [1] A. Rathnayake and W. K. Wanniarachchi, "Image source method based acoustic simulation for 3-d room environment," *International Journal of Scientific & Technology Research*, vol. 8, pp. 222–228, 11 2019.
- [2] A. Vuckovic, V. Radivojevic, A. Chen, and D. Popović, "Eeg drowsiness 2002 medengphy," 03 2015.
- [3] S. Chakrabarty and E. A. P. Habets, "Multi-speaker doa estimation using deep convolutional networks trained with noise signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 8–21, 2019.
- [4] A. Bohlender, A. Spriet, W. Tirry, and N. Madhu, "Exploiting temporal context in cnn based multisource doa estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1594–1608, 2021.
- [5] M. Phi, "Illustrated guide to lstm's and gru's: A step by step explanation," Jun 2020. [Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- [6] S. Kindt, A. Bohlender, and N. Madhu, "2d acoustic source localisation using decentralised deep neural networks on distributed microphone arrays," in *Speech Communication; 14th ITG Conference*, 2021, pp. 1–5.
- [7] S. Song, S. Kindt, J. Maes, A. B. Madhu *et al.*, "Influence of lossy speech codecs on hearing-aid, binaural sound source localisation using dnns," *arXiv preprint arXiv:2306.02344*, 2023.
- [8] S. Song, S. Kindt, J. Maes, A. Bohlender, and N. Madhu, "Comparative study of lc3plus and lyra codec on dnn-based source localisation for hearing aids," in *Speech Communication; 15th ITG Conference*, 2023, pp. 71–75.
- [9] C. Qi, S. Shen, R. Li, Z. Zhifeng, Q. Liu, J. Liang, and H. Zhang, "An efficient pruning scheme of deep neural networks for internet of things applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, 06 2021.
- [10] H. Tessier, "Neural network pruning 101, all you need to know not to get lost," in *towardsdatascience*, 2021.
- [11] H. Kayser, S. D. Ewert, J. Anemüller, T. Rohdenburg, V. Hohmann, and B. Kollmeier, "Database of multichannel in-ear and behind-the-ear head-related and binaural room impulse responses," *EURASIP Journal on advances in signal processing*, vol. 2009, pp. 1–10, 2009.
- [12] L. P. Feigen, "Physical characteristics of sound and hearing," *The American journal of cardiology*, vol. 28, no. 2, pp. 130–133, 1971.
- [13] W. A. Yost and M. C. Killion, "Hearing thresholds," *Encyclopedia of acoustics*, vol. 3, pp. 1545–1554, 1997.
- [14] H. Wallach, E. B. Newman, and M. R. Rosenzweig, "A precedence effect in sound localization," *The Journal of the Acoustical Society of America*, vol. 21, no. 4_Supplement, pp. 468–468, 1949.
- [15] S. Rickard and O. Yilmaz, "On the approximate w-disjoint orthogonality of speech," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. 1–529.
- [16] E. Accolti and F. Miyara, "Fast and controllable box-shaped room impulse response algorithm," *XIII Reunión de Trabajo en Procesamiento de la Información y Control, RPIC*, 2009.

6 References

- [17] M. L. Hawley, R. Y. Litovsky, and J. F. Culling, "The benefit of binaural hearing in a cocktail party: Effect of location and type of interferer," *The Journal of the Acoustical Society of America*, vol. 115, no. 2, pp. 833–843, 2004.
- [18] P. Avan, F. Giraudet, and B. Büki, "Importance of binaural hearing," *Audiology and Neurotology*, vol. 20, no. Suppl. 1, pp. 3–6, 2015.
- [19] E. A. Lopez-Poveda, "Chapter 10 - development of fundamental aspects of human auditory perception," in *Development of Auditory and Vestibular Systems*, R. Romand and I. Varela-Nieto, Eds. San Diego: Academic Press, 2014, pp. 287–314. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780124080881000105>
- [20] A. Francl and J. H. McDermott, "Deep neural network models of sound localization reveal how perception is adapted to real-world environments," *Nature human behaviour*, vol. 6, no. 1, pp. 111–133, 2022.
- [21] R. Roden, N. Moritz, S. Gerlach, S. Weinzierl, and S. Goetze, *On sound source localization of speech signals using deep neural networks*. Technische Universität Berlin, 2019.
- [22] S. Jiang, L. Wu, P. Yuan, Y. Sun, and H. Liu, "Deep and cnn fusion method for binaural sound source localisation," *The Journal of Engineering*, vol. 2020, no. 13, pp. 511–516, 2020.
- [23] M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent, "Sound source localization," *European annals of otorhinolaryngology, head and neck diseases*, vol. 135, no. 4, pp. 259–264, 2018.
- [24] F. Denk, S. M. Ernst, S. D. Ewert, and B. Kollmeier, "Adapting hearing devices to the individual ear acoustics: Database and target response correction functions for various device styles," *Trends in hearing*, vol. 22, p. 2331216518779313, 2018.
- [25] Z. Yang and R. R. Choudhury, "Personalizing head related transfer functions for earables," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 137–150. [Online]. Available: <https://doi.org/10.1145/3452296.3472907>
- [26] H. Do, H. F. Silverman, and Y. Yu, "A real-time srp-phat source location implementation using stochastic region contraction(src) on a large-aperture microphone array," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, 2007, pp. I–121–I–124.
- [27] M. Cobos, A. Marti, and J. J. Lopez, "A modified srp-phat functional for robust real-time sound source localization with scalable spatial sampling," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 71–74, 2011.
- [28] N. Madhu, R. Martin, U. Heute, and C. Antweiler, "Acoustic source localization with microphone arrays," *Advances in Digital Speech Transmission*, pp. 135–170, 2008.
- [29] Z. Khan, M. M. Kamal, N. Hamzah, K. Othman, and N. Khan, "Analysis of performance for multiple signal classification (music) in estimating direction of arrival," in *2008 IEEE International RF and Microwave Conference*. IEEE, 2008, pp. 524–529.
- [30] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

6 References

- [31] A. Opidi, "Pytorch loss functions: The ultimate guide," Aug 2023. [Online]. Available: <https://neptune.ai/blog/pytorch-loss-functions>
- [32] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.
- [33] M. Yasuda, Y. Koizumi, S. Saito, H. Uematsu, and K. Imoto, "Sound event localization based on sound intensity vector re-fined by dnn-based denoising and source separation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 651–655.
- [34] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2019.
- [35] A. Bohlender, L. Roelens, and N. Madhu, "Improved deep speaker localization and tracking: Revised training paradigm and controlled latency," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [36] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assessment," in *Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1988/file/07e1cd7dca89a1678042477183b7ac3f-Paper.pdf
- [37] J. Yang, B. Martinez, A. Bulat, G. Tzimiropoulos *et al*, "Knowledge distillation via softmax regression representation learning." International Conference on Learning Representations (ICLR), 2021.
- [38] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [39] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *CoRR*, vol. abs/2006.05525, 2020. [Online]. Available: <https://arxiv.org/abs/2006.05525>
- [40] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.
- [41] H.-H. Chou, C.-T. Chiu, and Y.-P. Liao, "Cross-layer knowledge distillation with kl divergence and offline ensemble for compressing deep neural network," *APSIPA Transactions on Signal and Information Processing*, vol. 10, p. e18, 2021.
- [42] F. Ruffy and K. Chahal, "The state of knowledge distillation for classification," *arXiv preprint arXiv:1912.10850*, 2019.
- [43] T. Su, Q. Liang, J. Zhang, Z. Yu, G. Wang, and X. Liu, "Attention-based feature interaction for efficient online knowledge distillation," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 579–588.
- [44] I. Chung, S. Park, J. Kim, and N. Kwak, "Feature-map-level online adversarial knowledge distillation," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2006–2015.

6 References

- [45] F. Denk, S. M. A. Ernst, S. D. Ewert, and B. Kollmeier, "Adapting hearing devices to the individual ear acoustics: Database and target response correction functions for various device styles," *Trends Hear.*, vol. 22, p. 2331216518779313, Jan. 2018.
- [46] P. Cooreman, A. Bohlender, and N. Madhu, "Crnn-based multi-doa estimator: Comparing classification and regression," in *Speech Communication; 15th ITG Conference*, 2023, pp. 156–160.
- [47] P. Contributors, "Conv2d¶," 2023. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Conv2d.html>
- [48] P. contributors, "Linear¶," 2023. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.Linear.html>
- [49] M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent, "Sound source localization," *European Annals of Otorhinolaryngology, Head and Neck Diseases*, vol. 135, no. 4, pp. 259–264, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187972961830067X>
- [50] P. Vecchiotti, N. Ma, S. Squartini, and G. J. Brown, "End-to-end binaural sound localisation from the raw waveform," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 451–455.
- [51] K. Youssef, S. Argentieri, and J.-L. Zarader, "A learning-based approach to robust binaural sound localization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2927–2932.
- [52] Q. Yang and Y. Zheng, "Deepear: Sound localization with binaural microphones," *IEEE Transactions on Mobile Computing*, 2022.

Appendices

Appendix A: accuracy overview

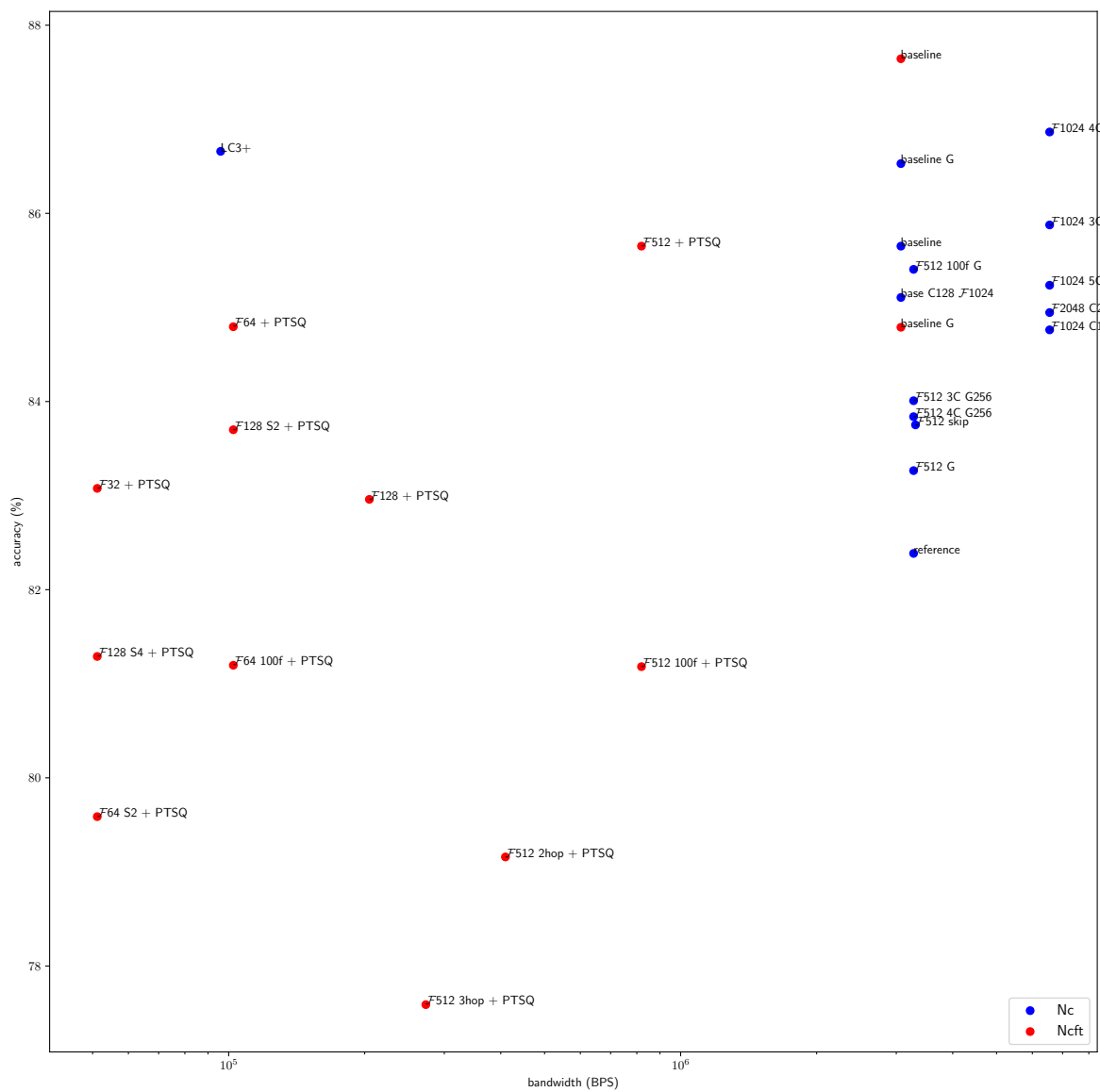


Figure 14: Accuracy - bandwidth overview of most interesting compressed and expanded models with the bandwidth axis logarithmic.

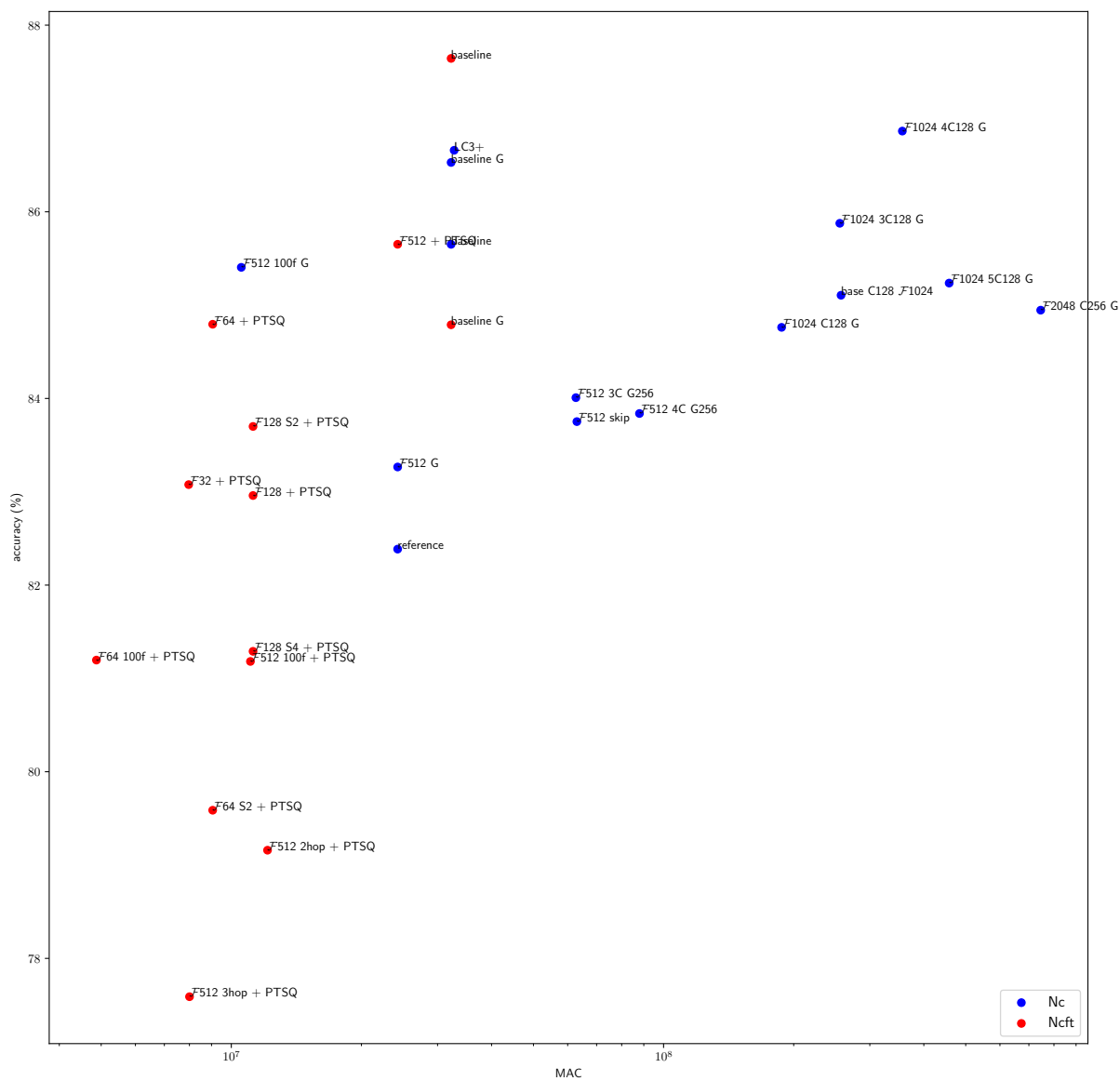


Figure 15: Accuracy - MAC overview of most interesting compressed and expanded models with the MAC axis logarithmic.

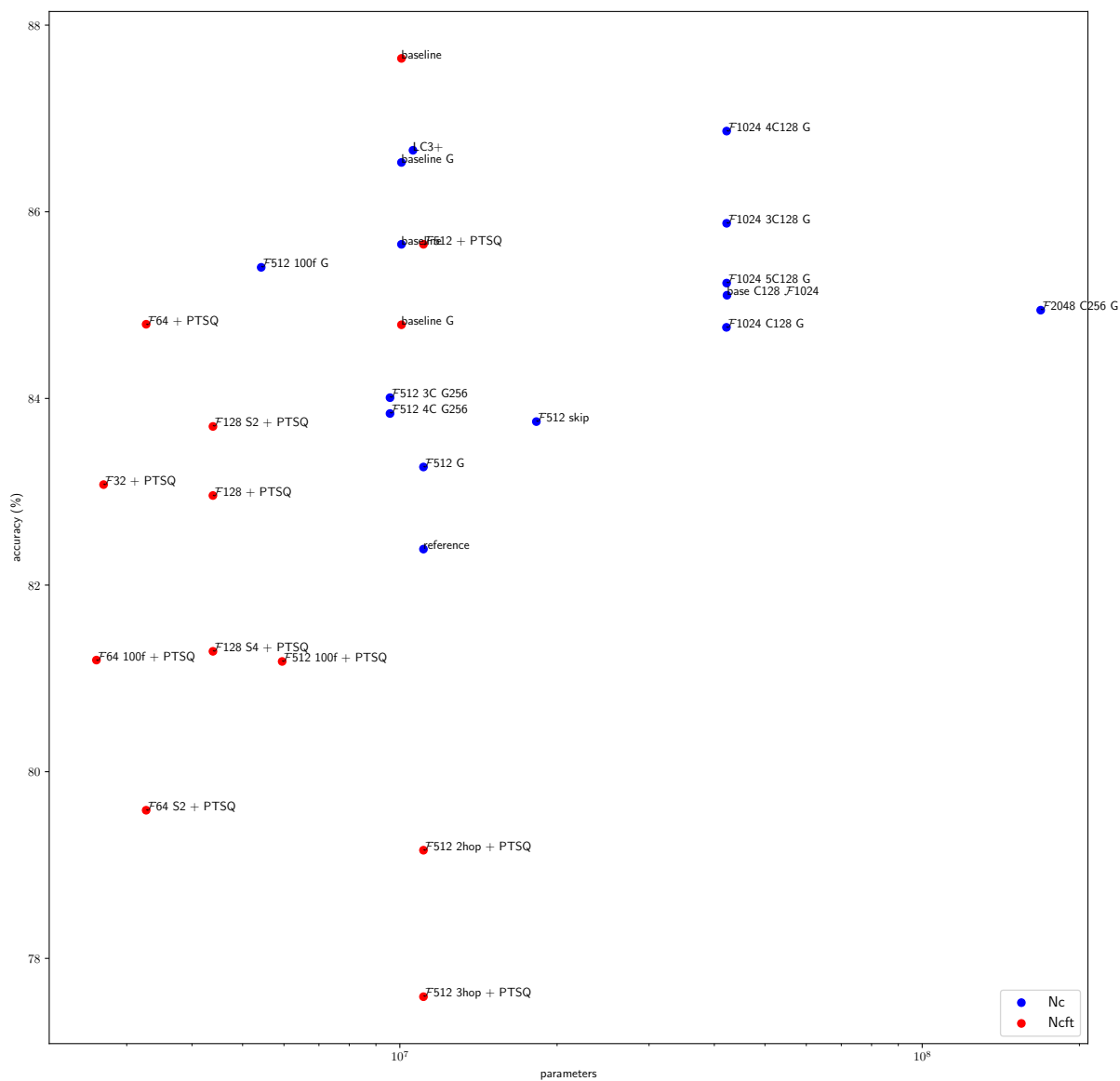


Figure 16: Accuracy - parameter overview of most interesting compressed and expanded models with the parameter axis logarithmic.