Direction of Arrival Estimation: A Comparative Analysis of Various Methods Under Practical Disruptive Conditions

Sven BARRAY, Tanguy LEGRAND, Nathan SANDJA

March 2024

Contents

1 Introduction

The task of Direction of Arrival (DOA) estimation is a critical challenge in the field of signal processing, underpinning technologies in diverse sectors such as radar systems, telecommunications, and audio processing. This challenge involves discerning the spatial origin of signals, a task complicated by the interplay of source characteristics and sensor configurations. With technological advancements and the increasing complexity of application environments, the quest for accurate and robust DOA estimation methods has become more pressing.

At the heart of our study is a well-defined problem geometry that serves as our ground for comparing different DOA estimation techniques. We consider a scenario where two distant sources emit signals captured by a uniform linear array (ULA) of sensors. This configuration is chosen for its simplicity and relevance, offering a clear framework for assessing the efficacy of various estimation methods. The assumption of infinite distance between the sources and the sensor array simplifies the signal model to a degree, allowing for a focus on the estimation methods themselves rather than the complexities of signal propagation.

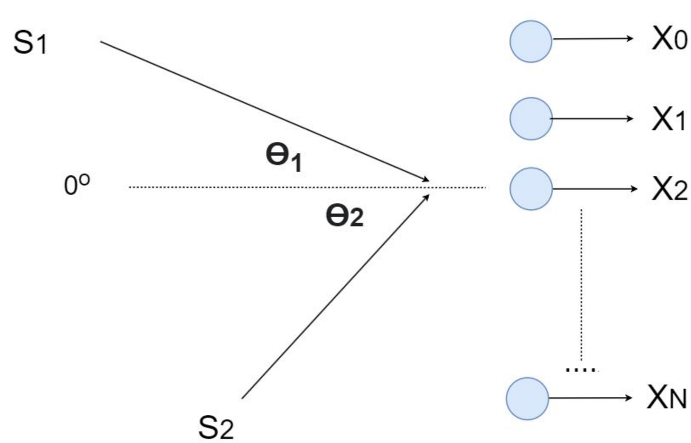

Figure 1: Two signals S1 and S2 are sent by two distinct sources. An array of Uniform linear sensors, represented by the blue dots, receive the combined signal as Xn, n being the index of the array. The objective is to determine the angles $\theta 1$ and $\theta 2$ from which the source signals originate from.

The conventional models for DOA estimation are capable of detecting signals within a angular range of -90° to 90°. However, their performance tends to degrade towards the extremities of this range, which often prompts researchers to focus on a narrower operational range to ensure higher accuracy and reliability in their findings. In alignment with this approach, our study will concentrate on an angular range from -60° to 60°.

Within this context, we delve into a comparative analysis of five distinct DOA estimation methods, each documented in its own scientific paper and selected for their application to this precise problem setting. These methods and their associated research articles are as follow:

- The Beamforming Method - *Two Decades of Array Signal Processing Research: The Parametric Approach* - by Krim, H., & Viberg, M. (1996)[1]
- The MUSIC Method - *MUSIC, Maximum Likelihood, and Cramer-Rao Bound* - by Stoica, P., & Nehorai, A. (1989)[2]
- The Maximum Likelihood Method - *Exact and Large Sample Maximum Likelihood Techniques for Parameter Estimation and Detection in Array Processing* - by Ottersten, B. (1993)[3]
- A Convolutional Neural Network (CNN) - *Deep networks for direction-of-arrival estimation in low SNR. IEEE Transactions on Signal Processing* - by Papageorgiou, G. K. et al. (2021)[4]
- A custom version of the CNN above, trained specifically to be robust to the disruption parameters we introduce below.

By focusing on methodologies that address the same geometric configuration, our study aims to isolate the performance of each method from the underlying problem setup, providing a clear comparison of their effectiveness.

In addition, theoretical methods and controlled environments often fall short of capturing the full range of challenges encountered in real-world applications. Acknowledging this gap, our analysis extends beyond the conventional signal-to-noise ratio (SNR) considerations to include three additional types of disruptions, which we will call "disruption parameters" throughout this paper. The three potential disruptions we added are well known in the field, but rarely documented when evaluating a method's quality.

These are the disruption parameters we will consider:

- Signal-to-noise ratio (SNR): establishes the ratio between the power (variance) of the power and its noise, exprimed in decibels (dB). A table to help understand how SNR scales is available in Appendix 10.
- Correlation between source signals: accounts for potential inter-signal interference as signals might be correlated, for instance through an echo.

- Variance ratio between source signals: addresses the real-world likelihood of disparate signal strengths, rather than all signals having the exact same power.
- Positional perturbation: reflects the practical reality of slight misalignment within the sensor array, which is not always perfectly aligned with uniform distances.

Employing a Python-based framework, our study rigorously evaluates how these five DOA estimation methods perform under conditions that simulate real-world situations. This comprehensive approach sheds light on the robustness and adaptability of traditional and deep learning techniques to practical disruptions.

2 Array signal model

2.1 Base model

Consider a scenario with T snapshots of a signal, indexed by t where $t = 1, 2, \ldots, T$. We have K sources, indexed by k where $k = 1, \ldots, K$, as well as a vector of N sensors, indexed by n where $n = 1, 2, \ldots, N$. The studies we base our research on either had $N = 16$ or a varying number, and various values for T. In our study, we set N to 16 and T to 500.

The angle of each source k relative to the array is denoted by θ_k. The signal emitted from source k at snapshot t is represented by a complex number $s_{\theta_k}(t)$. The matrix S, representing these signals, has dimensions $T \times K$, containing complex numbers that capture both amplitude and phase. Each value is random, following a Gaussian distribution of mean 0.

In DOA, a steering vector is a vector that simulates the delay of when a sensor receive the signal compared to the other sensors. The formula of the steering vector depends on the geometry of the problem. In the context of a Uniform Linear Array (ULA), the formula for the steering vector at angle θ_k is given by:

$$a_n(\theta_k) = \exp\left(-j\frac{2\pi d}{\lambda}(n-1)\sin(\theta_k)\right) \tag{1}$$

where n is the sensor index, and j is the imaginary unit. d represents the distance between each sensor and λ the wavelength. As these two parameters do not affect the methods' performance, we've set d and λ to 1 and 2 respectively.

Here, $a_n(\theta_k)$ is the n^{th} element of the steering vector for the source at θ_k.

The array steering matrix $A(\theta)$, with dimensions $N \times K$, comprises the steering vectors for all angles of interest.

The noise at each snapshot, $e(t)$ is assumed to be a Gaussian random variable with a mean of 0, accounting for random perturbations in the signal. Its standard variation is set to 1, as it simplifies formulas yet to be introduced.

Therefore, the received signal at snapshot t is modeled as:

$$x(t) = A(\theta)s(t) + e(t) \tag{2}$$

The matrix X, encapsulating all snapshots, has dimensions $N \times T$ and represents the complete signal received by the array.

2.2 Incorporating Disruption Parameters

Out of our four disruption parameters, three of them are the signal-to-noise ratio (SNR), correlation between source signals, and the variance ratio between source signals. These parameters are integrated into our model by manipulating the covariance matrix P of the signal matrix S.

Within this matrix, P_{kk} corresponds to the variance of the k^{th} source signal. To maintain our predefined conditions, P_{22} is set to be a multiple of P_{11}, consistent with the variance ratio parameter. Moreover, the SNR is calibrated by ensuring the variance ratio between the signal and the noise e (which is assigned a variance of 1) aligns with our specified SNR parameter. The correlation parameter is introduced by computing the off-diagonal terms P_{12} and P_{21} through the following formula:

$$\text{covariance} = \rho \times \sqrt{P_{11} \times P_{22}} \tag{3}$$

where ρ represents the correlation coefficient between the sources, assumed to be real.

To construct the signal matrix S that embodies these disruption parameters, we employ the Cholesky decomposition technique. This technique generates a matrix whose covariance matrix is P, thus embedding within S the three disruption parameters.

The fourth parameter under consideration is positional perturbation. For each sensor n, the positional perturbation value, denoted pp_n, follows a Gaussian distribution with a mean of 1 and a standard deviation equivalent to our positional perturbation parameter. Integrating pp_n into the steering vector equation:

$$a_n(\theta_k) = \exp\left(-j\frac{2\pi d}{\lambda}(n-1)\sin(\theta_k)\cdot pp_n\right) \tag{4}$$

offsets the sensor's phase offset by a factor of pp_n. Phase offset can be caused by a sensor's misalignment or error in tuning, which are analogous from a theorical standpoint. The degree of misalignment intensifies as pp_n diverges from 1. Consequently, the greater the standard deviation of pp_n, as defined by our positional perturbation parameter, the more pronounced the sensor misalignments will be.

2.3 Cramér-Rao Lower Bound

The Cramér-Rao Lower Bound (CRLB) sets a theoretical benchmark in estimation theory, representing the lowest variance achievable by an unbiased estimator. While it's important to acknowledge that the estimators employed in our research are not unbiased, the CRLB still serves as an efficient tool for assessing estimator performance. This lower bound can be used as a comparison against the Mean Squared Error (MSE) of our estimators to evaluate their efficiency. The MSE measures the average squared difference between the estimated values and the true values, providing a comprehensive metric of estimator accuracy and variability.

The MSE is pertinent to compare with the CRLB as it reflects both the variance and the bias of the estimators. A lower MSE indicates a more accurate and consistent estimator. Comparing the MSE of our estimators to the CRLB helps us discern how close their performance is to the theoretical optimum.

In our specific array signal model, the expression for the CRLB is as follows:[2]

$$\mathrm{CRLB}(\hat{\theta}_k) = \frac{\sigma^2}{2N}\left[\mathrm{RE}\left\{\left[\left(D^*\left[I - A(A^*A)^{-1}A^*\right]D\right)\circ P^T\right]^{-1}\right\}\right]_{kk} \tag{5}$$

where:
σ is the standard deviation of the noise e.
D is the derivative of A with respect to θ.
I is the Identity matrix.
$*$ denotes the conjugate transpose operation.
T is the transpose operation.
$\circ$ denotes the Hadamard product.
RE[] denotes the real part of a matrix.

3 Methods Overview

3.1 Beamforming

Beamforming, as a directional signal processing technique, leverages the phase and amplitude of the signal at each sensor to direct the sensitivity of the array towards specific directions while suppressing signals from other directions. The fundamental concept behind beamforming is to exploit the constructive and destructive interference of signals, enabling the array to focus on a signal from a particular direction. This directional focus enhances the signal-to-noise ratio (SNR) for the signal of interest, making beamforming particularly effective in environments with multiple interfering sources.[1]

Beamforming involves computing the output power for specific direction angles to form what is known as the beamforming spectrum. This spectrum is essentially a composite of the outputs across a range of angles, providing insight into the direction of the received signals. The method hinges on the concept of adjusting the array's sensitivity to favor signals from a targeted direction, thereby enhancing signal clarity and DOA estimation accuracy.

The core of the beamforming technique lies in the computation of the weight vector $\mathbf{w}$ for a range of many directions of interest, noted doi_i, with i the index of the angle, $i \in I$. In our study, this range includes every angle from -60° to 60° with a uniform step of 0.1, hence $I = 1201$ angles in total. The weight vector $\mathbf{w}$ manipulates the array's received signals to reinforce those coming from the desired direction.

Given an angle doi_i, the weight vector is calculated as follows:

$$\mathbf{w}(doi_i) = \frac{a(doi_i)}{\sqrt{a^H(doi_i)a(doi_i)}} \tag{6}$$

where $\hat{\mathbf{R}}^{-1}$ is the inverse of the estimated signal covariance matrix of X, $\mathbf{a}(doi_i)$ is the steering vector for the direction doi_i, and H represents the Hermitian transpose. This formulation ensures that the beamforming weights adjust the phase and amplitude of the received signal to maximize the output for signals from the direction doi_i.

The output of the beamformer for a given direction doi_i is then calculated as:

$$\text{Output}(doi_i) = \left|\mathbf{w}^H \hat{\mathbf{R}} \mathbf{w}\right| \tag{7}$$

where $|\cdot|$ denotes the absolute value. This output represents the power received from the direction doi_i after the beamforming process. By computing this output across the whole range of angles, we construct the beamforming spectrum, which is a comprehensive representation of the power distribution as a function of direction. The peaks in this spectrum indicate the estimated directions of arrival for the signals of interest.

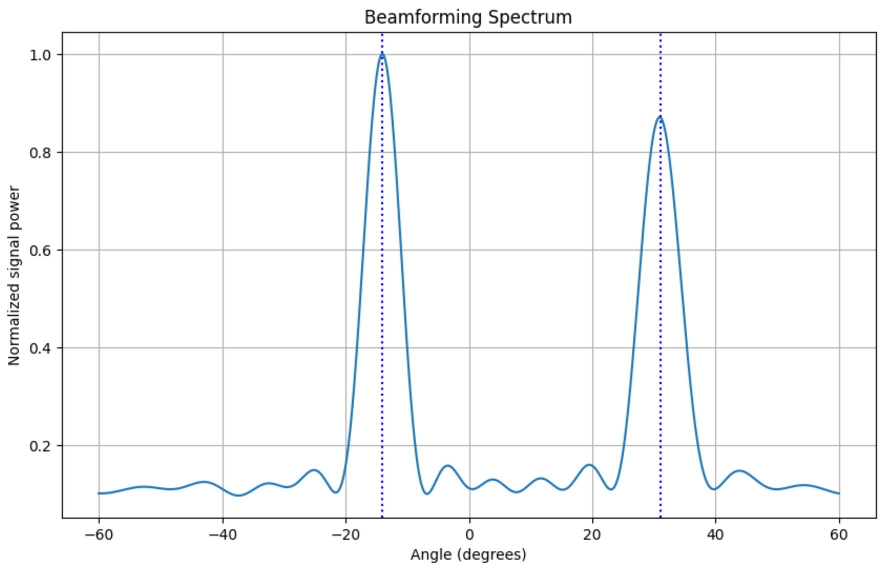

Figure 2: A visualized Beamforming spectrum. The peaks of the spectrum correspond to the estimation. The blue dotted lines correspond to the target angles.

3.2 MUSIC

The Multiple Signal Classification (MUSIC) algorithm is a state-of-the-art technique for Direction of Arrival (DOA) estimation that exploits the eigenstructure of the received signal's covariance matrix to discern the spatial origins of multiple independant signals.[2] Unlike beamforming, which directly manipulates the received signal through a weighted sum, MUSIC classifies signals based on the orthogonality between the signal subspace and the noise subspace. This distinction allows MUSIC to achieve high resolution and accuracy in estimating DOAs, especially in scenarios with closely spaced sources or in the presence of substantial noise.

The calculation of the MUSIC spectrum is rooted in the eigenvalue decomposition of the estimated signal covariance matrix, $\mathbf{R}$:. From this decomposition, we extract the eigenvectors corresponding to the noise subspace, denoted as **NSP**. This selection is based on the premise that the dimensionality of the noise subspace exceeds that of the signal subspace, given the greater number of noise sources relative to signal sources.

Similarly to the beamforming method, we calculate the MUSIC spectrum for each direction of interest doi_i. This calculation involves the steering vector $\mathbf{a}(doi_i)$ for each angle doi_i, and the spectrum for this angle is determined by:

$$\text{MUSIC Spectrum}(doi_i) = \frac{1}{\mathbf{a}^H(doi_i)\mathbf{NSPNSP}^H\mathbf{a}(doi_i)} \tag{8}$$

This expression inversely relates the projection magnitude of the steering vector onto the noise subspace to the strength of the signal originating from direction doi_i, with peaks in the spectrum signifying the DOAs.

To construct the MUSIC spectrum, the procedure iterates over the entire set of directions of interest (I), calculating MUSIC Spectrum(doi_i) for each. The resulting spectrum forms a detailed map of power distribution across the examined angular range, with its peaks directly indicating the estimated DOAs of incoming signals.

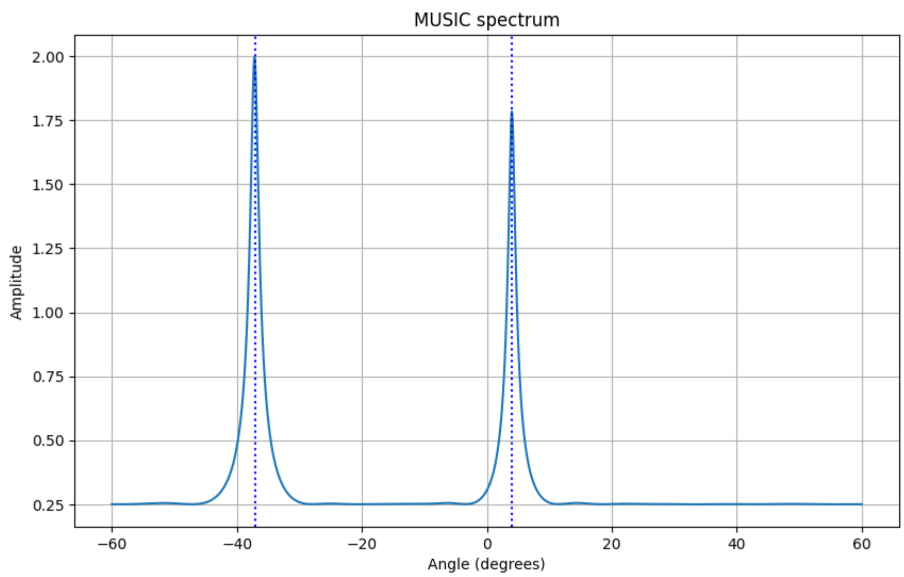

Figure 3: A visualized MUSIC spectrum. The peaks of the spectrum correspond to the estimation. The blue dotted lines correspond to the target angles.

3.3 Deterministic Maximum Likelihood

Maximum Likelihood Estimation is a fundamental statistical approach used across various domains to estimate the parameters of a model in a way that maximizes the likelihood of observing the given data. Its versatility and robustness have made it an efficient choice in many fields, including DOA problems.

The principle behind maximum likelihood estimation is to find the parameter values that make the observed data most probable under a specified statistical model.

In the context of DOA estimation, the Deterministic Maximum Likelihood (DML) method leverages this principle by iteratively refining angle estimates to maximize the likelihood function, defined in terms of the difference between observed and model-predicted signals. This process relies on minimizing a cost function, typically the mean square error, to quantitatively assess the accuracy of the estimates at each iteration.[3]

The iterative process allows progressive enhancing in the precision of DOA estimates. It begins with an initial set of guessed angle estimates, which are then updated in each iteration based on the optimization of the cost function. The algorithm proceeds until reaching a stopping criterion, such as a predefined threshold for the cost function's change, indicating that further iterations are unlikely to result in significant improvements.

The versatility of the DML method in DOA estimation is particularly notable when initial estimates of signal origins are informed rather than arbitrary. This foundation allows for a more directed and potentially efficient search for the true directions of arrival. Moreover, the method's performance can be enhanced through a variety of strategies, the implementation of which may vary depending on the specific requirements and characteristics of the scenario at hand, such as the expected proximity of source signals. Although our study did not adopt these strategies—owing to their highly situational nature—below is are two illustrative examples of potential tactics for optimizing the DML method's effectiveness:

- **Multiple initial guesses:** Implementing the algorithm with different starting points helps avoid local minima, enhancing the likelihood of uncovering the true DOAs.
- **Adaptative parameter updates:** Employing adaptive strategies to update parameters based on the optimization's progress can enhance the efficiency and reliability of convergence towards optimal estimates. This could involve adjusting the magnitude and direction of parameter adjustments dynamically, based on the curvature of the likelihood function or the reduction in the objective function from one iteration to the next.
- **Regularization techniques:** Applying regularization mitigates the risk of overfitting, particularly in high-noise situations, by penalizing large adjustments in the angle estimates.

3.4 Convolutional Neural Network

Next, we employ a deep learning model, more specifically a Convolutional Neural Network (CNN). We used the exact same structure and settings to train this model than the original research paper did[4]. In the next section, we will present a variation of this model where you adjusted the model's settings to make it robust specifically to our disruption parameters.

This model takes as input the true array manifold matrix, which is a matrix $X$of size $3 \times N \times N$ where N is the number of sensors. The third dimension represents the three different channels, the first two are the real and imaginary parts of the covariance matrix, and the third is the phase.

For the output the model considers an on-grid approach, i.e. the model considers that the angles can only be found in a grid G of a certain interval between α and -α with α the maximum angle, and with a certain step μ through the interval.

$$\mathbf{G} = \begin{pmatrix} -\alpha & \dots & -\mu & 0 & \mu & \dots & \alpha \end{pmatrix}$$

The model performs a multi-label classification task on this grid based on the input data, producing an output estimate of the angles on the grid. In other words, a Z vector of size equal to the number of points on the grid with a probability of presence on each grid point which represents the angle corresponding to its index -α.

$$\mathbf{Z} = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 & 1 & 0\dots & 0 \end{pmatrix}$$

3.4.1 Model architecture

The model is made up of 24 different layers.

It is first composed of 2D convolutional layers to extract features from the input data. To do this, there are 4 2D convolutional layers each, followed by a normalisation layer and then a Relu activation layer. All the convolutional layers have 256 filters, the first of which has a kernel of size 3×3 and a step of 2, then the subsequent ones have a kernel of size 2×2 and a step of 1.

The next layer, the 13th, is a flatten layer which transforms the features extracted in the convolutional layers into a vector which will be used as input for the second part of the model : the fully connected layers which estimate the angles from these features.

This second part of the model is made up of 3 dense layers each followed by a Relu activation layer to add non-linearity and a dropout layer to make the model more robust. The 3 dense layers have 4096, 2048 and 1024 neurons respectively. Then there is a final dense layer of $2G + 1$ neurons with G being the maximum angle of the grid and considering a step of 1 in this same grid. This layer is followed by a sigmoid activation layer to return a value between 0 and 1 for each point on the grid.

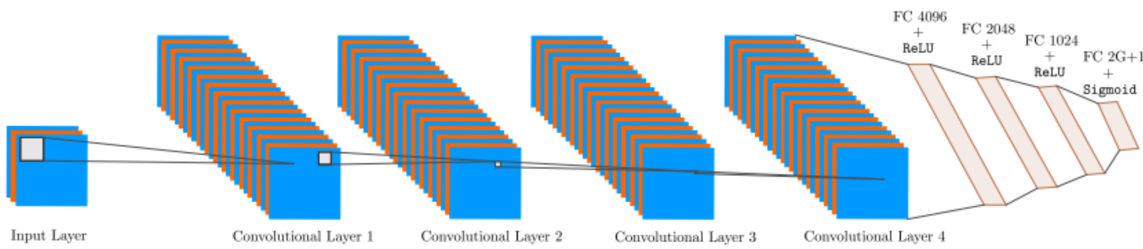

Figure 4: Model Architecture [4]

3.4.2 Model training

The grid is set between angles -60 and 60 degrees with a resolution of 1 degree, so there are 121 points in the grid.

The number of sources is set to 2, and for each SNR level, we consider all the possible combinations of angles in the grid, i.e. 7260 data for each SNR level in our training set. As we found that training the model only on low SNRs also performed well on high SNRs, we train the model on SNRs ranging from -20dB to 0dB with a step size of 5. Our training set is therefore made up of $7260 \times 5 = 36300$ example trainings. This training set is randomly divided between the training set (90%) and the validation set (10%). The batch size is set to 32 and the model is trained over 200 epochs.

The model is trained using binary cross-entropy loss, the Adam optimizer with $\beta1 = 0.9$ and $\beta2 = 0.999$ and an initial learning rate of 0.001. The learning rate is reduced by 50% every 10 epochs to ensure convergence.

Our implementation of the exact same model yielded slightly poorer performance in out testing. In particular, the model has a strong tendency to estimate adjacent angles in low SNR situations. It is still unclear to us if the reason it didn't perform as well was because of an inaccurate implementation, a difference in training environment, or because the performance analysis presented in the original research paper was done in conditions advantageous to the model.

3.5 Custom Convolutional Neural Network

In order to improve performance in the specific context of our problem, in particular to make it more robust to variations of our disruption parameters, some of the parameters of the previous model have been modified.

Now, we train the model on different values for the variance and correlation between the two sources, the SNR and the positional perturbation parameter. For each set of parameters, we consider all the possible combinations of angles in the grid, i.e. 7260 data for each set of parameters in our training set. The model is trained with the SNR values of -15 to 10 with a step size of 5. We didn't feel the necessity to also train it at -20 SNR, since the original model performed poorly at that SNR level anyway. We expanded the range of positive SNR by adding 5 and 10. The reason the original study didn't train with these SNR levels was because it wasn't necessary since the model performed well at high SNR anyway. However, we are adding disruptions, meaning the model isn't guaranteed to perform well at high SNR. It is also trained with: variance ratios with values of 1, 10 and 50; coefficient correlation with values of 0, 0.95 and 0.99; and the positional perturbation parameter with values of 0, 0.01 and 0.1. This resulted in a much larger training set, as it had a size of 7260*6*3*3*3=1,176,120 example trainings.

The other modifications made to the model are the addition of 4 pixels of padding to the first convolutional layer, the batch size is increased from 32 to 64, the model is trained over 30 epochs and the learning rate still has an initial value of 0.001 but is now divided by 2 every 5 epochs.

Working on such a large dataset grew the training time by a substantial amount. As a result, we weren't able to experiment a lot with our training parameters, and it is likely better training configurations exist.

4 Known strengths and weaknesses of the methods

4.1 Traditional methods

The traditional methods for Direction of Arrival (DOA) estimation, including Beamforming, MUSIC, and the Deterministic Maximum Likelihood (DML) method, each exhibit unique strengths and weaknesses that influence their performance under various conditions. While all methods demonstrate high efficiency in ideal scenarios—characterized by minimal disturbances, low noise levels, and sufficiently separated source angles—their performance diverges when faced with more challenging conditions.

- **Beamforming:** Beamforming is known for its simplicity and speed, making it a popular choice for real-time applications. However, its effectiveness diminishes when source angles are closely spaced. This is due to the method's relatively wide lobes, which merge into a single indistinguishable lobe when source angles approach within (approximately) 12° of each other with our implementation in the cases we tested, thereby impeding accurate DOA estimation for closely spaced sources.

- **MUSIC:** In contrast, the MUSIC method benefits from significantly thinner lobes, allowing it to accurately resolve closely spaced sources down to about a 5° separation in our simulated situations. This precision, however, comes at the cost of a slightly increased computational complexity due to the requirement for eigenvector decomposition. MUSIC also assumes signal independence and can experience degraded performance when dealing with highly correlated signals.

- **Deterministic Maximum Likelihood (DML):** The DML method stands out for its very substantial computational demands if you want your estimations to be of high quality. Additionally, obtaining high-quality results requires enough testing to determine the optimal parameters for the specific context. However, given enough time to go through this process, DML tends to yield the best results out of the three traditional methods. Additionally, unlike the other two traditional methods, DML still works when estimating signals very close to each other, and yields better results as the number of sources goes up.

In could be noted that in the case of a single source signal, Beamforming becomes an equivalent of DML, making it the standard method in such cases.

4.2 Deep Learning Methods

Deep learning approaches, such as Convolutional Neural Networks (CNNs), have been explored as alternative methods for Direction of Arrival (DOA) estimation. These methods utilize complex neural network architectures to process the input data and estimate the DOAs. While offering new capabilities, deep learning methods also present unique aspects compared to traditional approaches.

4.2.1 Strengths

- **High Adaptability:** Deep learning models are capable of learning complex patterns in the data, making them highly adaptable to a variety of signal environments. This adaptability enables them to effectively handle scenarios with low signal-to-noise ratios (SNRs), high levels of signal correlation, and closely spaced sources.

- **Customization for Specific Conditions:** A notable advantage of deep learning models is their ability to be customized for specific operational conditions. For instance, our custom CNN designed to be resilient against predefined disruption parameters is such an example. This level of customization allows for targeted improvements in model robustness that are difficult to achieve with traditional methods.

- **Flexibility:** Deep learning models are not bound by some specific constraints, while traditional models often are. For example, none of traditional methods we presented work if the number of sensors does not exceed the number of sources, and the first two require the X matrix to be invertible, which could not be the case. This offers deep learning models flexibility that is not available to other methods.

4.2.2 Weaknesses

- **Complex Training Process:** The training of deep learning models is computationally intensive and time-consuming. Determining the optimal model architecture and hyperparameters can be a meticulous process that requires substantial computational resources.

- **Model Opacity:** The decision-making process within deep learning models is often difficult to interpret. The complex nature of these models can obscure the understanding of how specific decisions or estimations are made, limiting the opportunity for manual adjustments based on intuitive understanding.

5 Determining Robustness to Disruptions through Monte Carlo Approximation

To rigorously assess the robustness of various DOA estimation methods in the presence of disruptions, our study employs Monte Carlo Approximation. This statistical method involves performing a large number of simulations to approximate the distribution of an estimator's performance across different scenarios. By systematically varying the conditions under which each DOA estimation method is tested —by changing our disruption parameters— we can gather extensive data on how each method performs under a wide range of conditions.

Monte Carlo approximation is particularly suited for this task due to its ability to model complex systems where analytical solutions are impractical or impossible. By simulating thousands of scenarios, each representing a unique combination of disruption parameters, we can obtain a robust statistical understanding of each method's performance. This approach allows us to quantify the impact of each disruption parameter on the accuracy and reliability of DOA estimates, providing a comprehensive comparison of the methods' effectiveness.

In implementing Monte Carlo approximation, we simulate each DOA estimation method's performance over a predefined set of conditions, repeatedly sampling from distributions that represent realistic variations in the disruption parameters. This process to compute the Mean Square Error (MSE) of the methods in various conditions, providing a detailed measure of each method's accuracy and consistency in the face of the examined disruption.

5.1 Outlier Management Strategy

In the field of DOA estimation, the nature of errors is binary: estimations typically are either close to the true angle or significantly misplaced. This scenario implies that errors are not incrementally proportional to the distance from the target, making the handling of outliers crucial for accurate analysis. An estimate considerably off target, whether by 20° or 40°, is equally erroneous, but larger errors disproportionately impact the calculated variance, distorting an assessment of a method's accuracy.

For our comparative analysis of DOA estimation methods, adopting a consistent criterion for outlier removal is essential to maintain fairness and methodological integrity. To this end, we have chosen to apply the same outlier interval across all methods based on the performance characteristics of the MUSIC method. The MUSIC algorithm is regarded as a standard benchmark in DOA estimation for scenarios involving two sources, due to its balance between accuracy, resolution, and computational efficiency.

Outliers are identified and removed based on their deviation from the mean estimation angle, with the threshold set at three standard deviations from the target. This standard deviation is calculated using the Monte Carlo estimations of the MUSIC method with 0 SNR and no other interference as a benchmark. The removal applies to both estimated angles: if either deviates enough from their respective target, the estimate is classified as an outlier and excluded from further analysis. This criterion reflects a conservative yet statistically principled approach to outlier management, allowing for the exclusion of extreme misestimations while retaining the bulk of the data for analysis. By applying this threshold uniformly across all evaluated DOA estimation methods, we ensure that the comparative analysis is conducted on a level playing field, with each method's performance assessed under equivalent conditions of outlier management.

5.2 Minimum and Maximum Theoretical MSE

In our study, the specific conditions under which we operate allow us to define theoretical minimum and maximum Mean Square Error (MSE) values that our methods should not surpass. This determination provides a benchmark against which we can compare the performance of each DOA estimation method. These theoretical limits will be included in our future comparison graphics as a reference, with the minimum MSE represented by a red dotted line and the maximum MSE by a blue dotted line.

For an easier-to-read comparison in our figures that include both traditional methods and CNN-based approaches, we standardize the grid resolution across the methods. Specifically, we adjust the grid used for traditional methods, originally set with a point every 0.1 degrees, to match the grid resolution of our CNN models, which is every 1 degree. The grid remains the initial one in figures that don't include the CNN models.

5.2.1 Minimum Theoretical MSE

Given that our analysis employs grid-based versions of all DOA estimation models, it's important to recognize that the MSE could never reach zero for the estimation of randomly generated angles not aligned with the grid. In the best case, the closest grid point is chosen as the estimate, which introduces a small but unavoidable error. This scenario leads to the minimum possible MSE, which occurs when every estimation lands on the grid point nearest to the actual angle.

This scenario represents the best-case performance for grid-based models, where the estimation error is solely due to the granularity of the grid itself. The value of this minimum MSE is determined by the square of the distance between the target angle and its nearest grid point. This distance essentially reflects the smallest possible deviation given the grid's resolution.

5.2.2 Maximum Theoretical MSE

When assessing the maximum theoretical Mean Square Error (MSE) in our DOA estimation study, we consider scenarios where disruption parameters are so severe that all useful signal information is essentially lost. Under these extreme conditions, the estimation process degenerates to random guessing within the operational angular range, devoid of any reliable information regarding the actual direction of arrival.

This random estimation behavior, governed by the disruptions, leads to a scenario where the predictions are uniformly distributed across the angular range considered reliable—specifically, the range not excluded by our outlier management criteria. In such a situation, the maximum MSE reflects the variance of a uniform distribution over this interval.

The MSE in this case is calculated based on the properties of a uniform distribution over an interval $[a, b]$, where a and b denote the lower and upper bounds of the interval, respectively, within which we expect our estimates to fall after excluding outliers. The formula for the variance of a uniform distribution, which in this context represents our maximum theoretical MSE, is given by:

$$\text{Variance} = \frac{(b-a)^2}{12}$$

5.3 Overview

Unless specifically mentioned, we assume a variance ratio of 1, a correlation coefficient of 0, and a positional perturbation of 0 in our comparisons. These settings represent a scenario with no disruptions other than noise.

Moreover, it's important to clarify that the Deterministic Maximum Likelihood (DML) method is not included in our multi-method comparisons due to the intensive computational demands required for high-quality estimations. To facilitate Monte Carlo approximations within a feasible timeframe, we resorted to a simplified version of the DML algorithm. This streamlined variant does not fully capture the method's potential effectiveness and, as such, does not allow for a direct, fair comparison with other models. However, to still gauge how disruptions impact the DML method, we conducted separate Monte Carlo approximations on this simplified version, providing insight into its performance under various disruptive conditions, albeit with the understanding that these results reflect a constrained iteration of the DML algorithm. Also, be aware that our method comparison is conducted in scenarios where all methods are operational, meaning the source signals are separated by at least 15°. Analyzing situations where the angles are closer would not be meaningful, as Beamforming and MUSIC methods lose functionality in such tight proximities, and the CNN models we use are inherently inclined to predict adjacent angles.

The standard approach for comparing various methods in signal processing, particularly in the context of DOA estimation, involves examining the evolution of the Mean Square Error (MSE) at different Signal-to-Noise Ratio (SNR) levels. This method of comparison is demonstrated in the Figure 5, which plots the MSE against SNR for each of the estimation methods under consideration.

As depicted in the figure, all four methods under review exhibit poor performance at very low SNR levels, demonstrating behavior akin to random guessing at an SNR of -25dB. Beyond this point, there is a noticeable improvement in performance as SNR increases, with all methods converging towards the minimum theoretical MSE in the range of -5 to 0 SNR. Notably, the CNN models achieve this convergence slower compared to traditional methods. This observation suggests that in scenarios where SNR is within a reasonable range and devoid of other disruptive factors, the adoption of deep learning-based methods may not offer a significant advantage over traditional approaches. It is worth noting that our custom CNN model, trained for specific scenarios, is on par with the original CNN model even in a context with no disruptions.

5.4 The Methods in the face of Disruptions

5.4.1 Variance ratio

Figure 6 shows how the ratio between the ratio of both source signals affects the MSE of the methods. The non-custom CNN model had many outliers past a variance ratio of 5 (more than 90%), leading to its exclusion from the graphic.

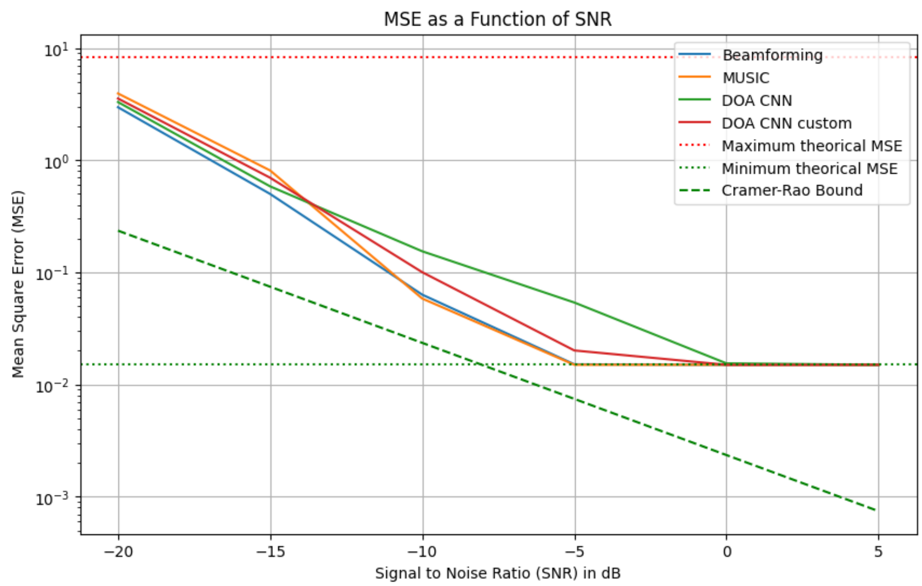

Figure 5: Comparing MSE as a function of SNR for four methods

This shows the strong limitations on deep learning models that weren't trained for the right purpose.

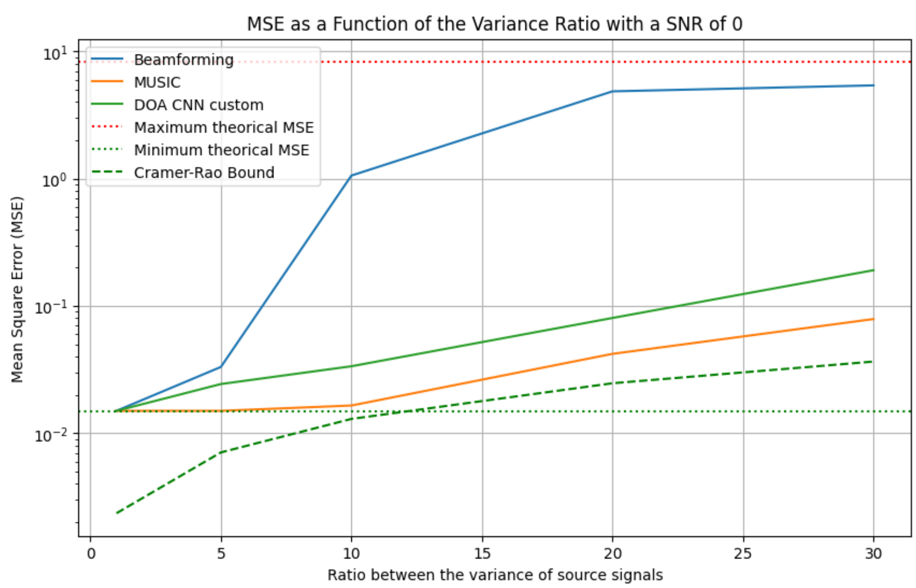

Figure 6: Comparing MSE for various variance ratio and SNR values

The Beamforming method exhibits a notable deterioration in performance as the variance ratio increases, with a significant spike in MSE observed just beyond a ratio of 5. Conversely, the MUSIC method demonstrates exceptional robustness against this type of disruption, maintaining a performance closely aligned with the Cramer-Rao Bound even at substantially higher variance ratios. This resilience highlights MUSIC's capacity to effectively navigate the challenges posed by uneven signal strengths.

The custom CNN model, designed with a focus on disruption robustness, delivers a competent MSE performance. However, its efficacy is somewhat overshadowed when compared to the superior resilience of the MUSIC method. Similarly, the Maximum Likelihood method, detailed further in Appendix 13 with multiple SNR values, parallels the MUSIC method in its robust performance under varied variance ratios, reinforcing the value of traditional estimation techniques in scenarios characterized by disparate signal strengths.

Figures displaying the individual methods' resilience to a high variance ratio are available in Appendix 11 to 14.

5.4.2 Correlation Coefficient

The analysis of how different DOA estimation methods respond to variations in the correlation coefficient rho, as shown in Figure 7, unveils intriguing patterns of resilience and vulnerability. Significantly, the custom CNN model demonstrates an exceptional level of robustness, with its performance remaining consistently unaffected across the entire range of rho values. Similarly, the Maximum Likelihood method, as detailed in Appendix 17, exhibits a comparable indifference to changes in correlation, maintaining its accuracy irrespective of the degree of signal correlation.

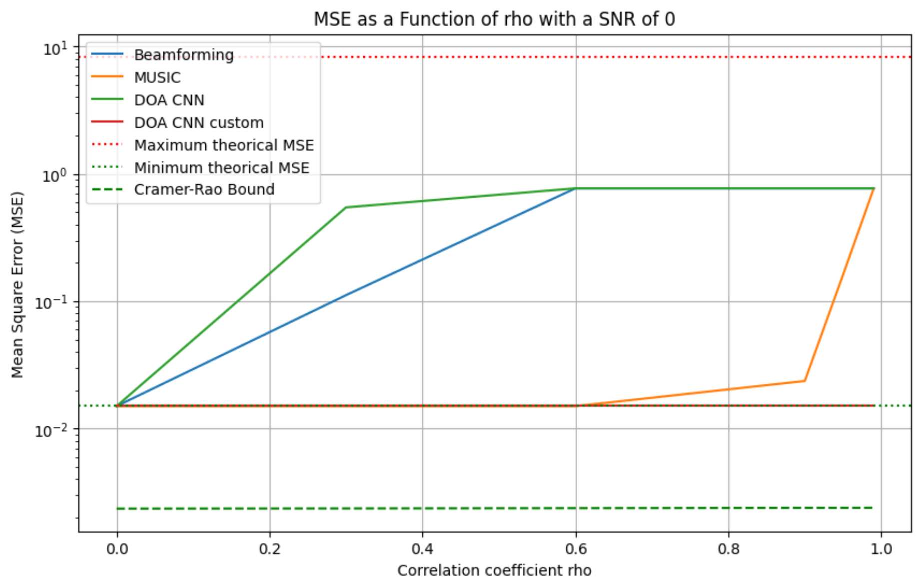

Figure 7: Comparing MSE for various correlation and SNR values

In contrast, the Beamforming method, the non-custom CNN model, and the MUSIC method show varied responses to increasing rho. All three approaches ultimately converge to a similar MSE value that does not correspond to the theoretical maximum MSE. Further investigation reveals that at high rho values, these methods tend to consistently estimate the same angles that deviate from the true target. This behavior suggests a systematic error or bias introduced by the correlation, leading these methods to become low variance but high bias estimators. The estimated values remain relatively stable across a broad spectrum of SNR levels, as documented in Appendix 15 through 18. It appears that heightened correlation effectively distorts the signal in a manner that these methods misinterpret, resulting in a persistent bias in their estimates.

The divergence in performance between these methods becomes particularly evident in their sensitivity to the correlation coefficient. While the Beamforming and non-custom CNN methods demonstrate a susceptibility to even low levels of correlation, manifesting in a rapid ascent to their bias-induced MSE plateau, the MUSIC method displays remarkable resilience, maintaining its precision up to a rho threshold of approximately 0.8. Beyond this point, its performance degrades abruptly and significantly, mirroring the rapid deterioration observed in the other methods.

5.4.3 Positional Perturbation

In our examination of the methods' resilience to positional perturbation, shown in Figure 8, while the MSE of each method raises at a similar pace while the disruption grow, a clear hierarchy of performance still emerges.

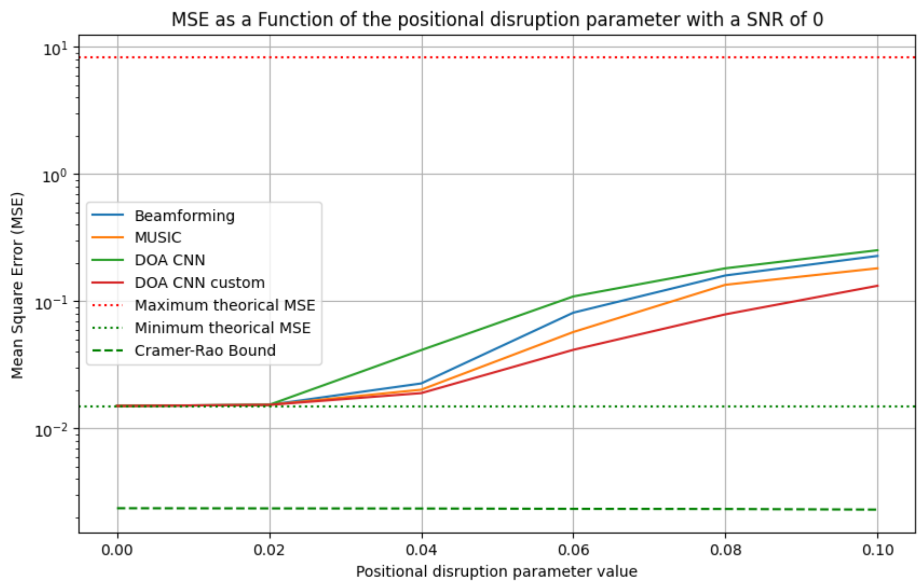

Figure 8: Comparing MSE for various positional disruption and SNR values

The custom CNN model outshines its counterparts by demonstrating the most robust behavior against sensor misalignments, showcasing its effectiveness in environments with positional disruptions. On the opposite end, the non-custom CNN model exhibits the highest MSE, underscoring the significance of specialized training for enhancing resilience. Among the traditional approaches, MUSIC maintains better performance over Beamforming, showcasing its relative advantage in handling sensor array irregularities.

The individual Figures displaying the individual methods' performance at various levels of SNR are available in Appendix 19 to 22

5.4.4 Many moderate disruptions at once

In our comprehensive evaluation under a scenario combining moderate disruptions—specifically, a variance ratio of 5, a correlation coefficient of 0.5, and a positional perturbation parameter of 0.05—our findings are depicted in Figure 9. The non-custom CNN model was excluded due to a predominance of outlier estimates once again.

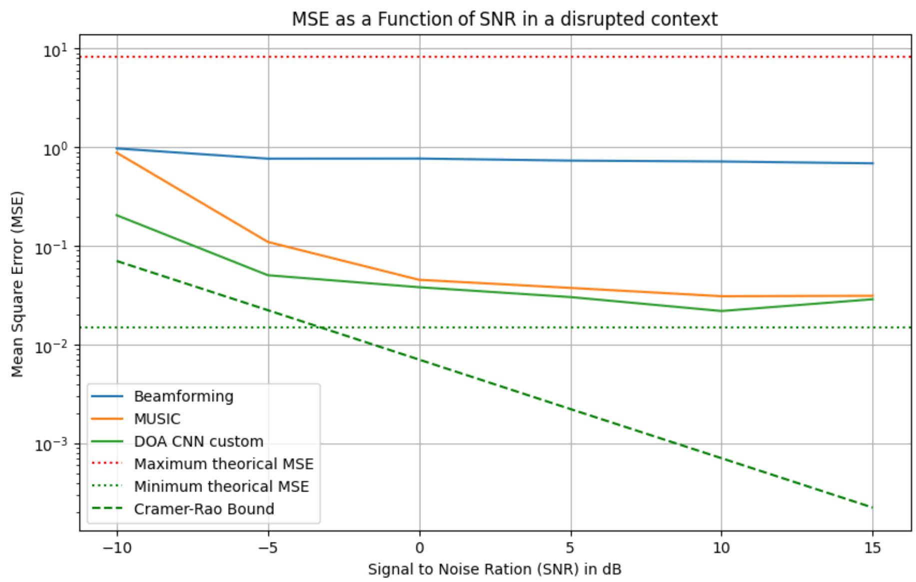

Figure 9: Comparing MSE for various SNR values with a variance ratio of 5, rho of 0.5, and a positional perturbation parameter of 0.05

In this multifaceted disruption environment, Beamforming struggles significantly, manifesting elevated MSE across the SNR spectrum. Conversely, MUSIC initially falters at lower SNR levels but notably improves as SNR increases, aligning more closely with optimal performance. The custom CNN, however, consistently exhibits strong performance, demonstrating resilience even at lower SNR levels, underscoring its effectiveness in handling complex, real-world signal disruption scenarios.

5.4.5 Comparison Highlights

Throughout our comparative analysis, several key observations have emerged that illuminate the strengths and weaknesses of the evaluated Direction of Arrival (DOA) estimation methods under various disruptive conditions.

Firstly, the custom Convolutional Neural Network (CNN) model has consistently showcased remarkable resilience across all types of disruptions. Its superior performance underscores the effectiveness of deep learning models that are meticulously tailored to anticipate and counteract specific disruptive influences. This result highlights the potential of specialized deep learning approaches in signal processing.

In contrast, the non-custom CNN model's performance was notably subpar in most scenarios lacking targeted training against specific disruptions. This discrepancy between the custom and non-custom models starkly demonstrates the critical importance of model training that closely aligns with anticipated real-world conditions. It also serves as a cautionary note: even custom specialized models might underperform when encountering unforeseen types of disruptions not covered during their training phase.

Although the Deterministic Maximum Likelihood (DML) method's full potential was not directly comparable due to computational constraints, the simplified version we analyzed indicated robust performance across varied disruptions. This suggests that, with sufficient computational resources, the DML method offers a highly effective approach to DOA estimation in the presence of complex disturbances.

The MUSIC method displayed moderate resilience, maintaining decent performance across a spectrum of disruptions, except under extreme conditions where its efficacy noticeably diminished. This behavior reflects MUSIC's utility in moderately challenging environments, with limitations becoming apparent as disruptions intensify.

Finally, Beamforming demonstrated significant vulnerability to disruptions, consistently exhibiting the weakest performance among the methods evaluated. This susceptibility highlights the method's limitations in complex, real-world signal processing scenarios and suggests a preference for alternative techniques under such conditions.

In summary, our analysis underscores the importance of method selection and model training tailored to the specific challenges of the operational environment in DOA estimation tasks. The findings advocate for a careful consideration of each method's strengths and vulnerabilities in the face of disruptions, guiding the selection towards the most resilient and effective approach for any given scenario.

6 Areas for Improvement

The main regret we had through this study was the high computational demands of our custom CNN model that didn't allow us to experiment a lot with the parameters. In particular, we believe a custom loss function that penalizes estimations of very close angles on the grid could be highly beneficial, since the model often predicts adjacent angles in low SNR, which alters the quality of the model. We also wish we trained the model for SNR values over 10, as its performance stagnates with higher SNR instead of improving. Finally, we were also unable to explore a version of the model that would produce random angle combinations on a fine-grained grid, instead of enumerating every possible combination on a grid with a wider step size.

In a more general matter, in the pursuit of refining the efficacy of DOA estimation methods, it is essential to investigate alternative avenues that hold potential for overcoming current constraints. In this section, we delve into unexplored methodologies that remain to be rigorously tested but present intriguing prospects. By investigating these new approaches, we aim to expand the scope of potential solutions for optimizing the performance of DOA estimation techniques.

6.1 Deep MUSIC

In addition to CNN, others deep learning methods have been explored in the realm of DOA problems. For example, a documented alternative is DeepMUSIC (Multiple Signal Classification via Deep Learning)[5], which consists of using deep learning to predict a MUSIC spectrum from a covariance matrix. This method is specified by training Q deep learning models on Q partitioned regions of the MUSIC spectrum, then reconstructing the spectrum and making the final prediction. The deep neural networks takes as input a matrix X of size $3 \times N \times N$, and outputs a MUSIC spectrum. After partitioning each spectrum into Q subspectra, deep neural networks are created.

6.1.1 Model Architechture

This model has 17 different layers, including the input and output layers. There are 4 2D convolutional layers to extract features from data with 3 channels and 256 filters. The first two convolutional layers have a kernel size of 5 x 5, and the next two have a kernel size of 3 x 3. Each of these layers is followed by a batch normalization layer and a ReLU activation layer. The 14th layer is a fully connected layer that adds non-linearity. The 15th layer is a softmax layer used for facilitating training and interpretation of the model for classification tasks, and a dropout layer is included to prevent overfitting by encouraging the network to learn more robust and generalizable features, thus improving the model's performance on unseen data. Finally the output of each model is a music spectrum for each subregion. The final prediction is built by combining every spectra to get the full spectrum, then find the peaks of the spectra to estimate the DOA angles. A figure representing the Neural architechture of this model is available in Appendix 23

6.2 Hybrid model-based Deep Learning

Another type of approach that bridges the gap between model-based methods and deep learning methods has been developing over the last few years. These new methods, known as hybrid model-based deep learning, use neural networks to optimise traditional model-based methods.[6]

The techniques discussed in this report fall into two categories: model-based methods and data-driven methods.

The classic model-based approach, like for example beamforming, uses domain knowledge of the problem as physical understanding of it to set his decisions rules. It creates mathematicals models of the problem and the solver based on it, so they are task-specific. Once the problem has been modelled and solved, it is guaranteed that a solution will be found, which will also be interpretable, and its reliability can be measured. However, it is often not possible to access sufficient knowledge about the problem, so the mathematical model becomes a simplified description of reality. The various approximations made can greatly reduce the performance of the method and lead to the model not being respectful of the real data. This category of method therefore quickly becomes limited when it is not possible to represent the problem using a mathematical model, or simply when the model is too complex. The presence of certain hyperparameters in models that are tuned manually adds a certain lack of rigour to the method and can sometimes be painful. In addition, decision-making can be fairly slow for this type of method, which often uses iterative solvers.

On the other hand, data-driven methods such as CNNs use the information contained in the data to set their decision rules. As they are not based on domain knowledge, they can be applied to scenarios where the analytical models are unknown or too complex. In fact, these methods are very general, and generally not very specific to the problem. However, despite their performance, these methods have a high level of abstraction, and are often treated as "black boxes" because they are very difficult to interpret and understand. They are therefore not considered reliable either, because even if they work, we don't fully understand how, and we are unable to rigorously prove that they work. These methods are also highly parameterised, so they require a massive dataset for their training, which is not always available. Therefore, training them requires a great deal of computational effort and energy, which, in the current global context, is a major negative point to bear in mind.

These two categories of methods are often seen as distinct, but in reality their main differences are the specificity of the method and its number of parameters. We can therefore represent these two categories at the extremes of a continuous spectrum of specificity and number of parameters, and between these two extremes, there are different mid-way methods which aim to recover the advantages of each: these are hybrid model-based deep learning methods.

These methods often involve using information about the domain via a mathematical model like the model-based methods, but using deep learning to optimise part of the method. There are several frameworks for this, such as the learned optimizer and deep unfolding.

The learned optimizer consists of using a deep learning algorithm to automatically tune the hyperparameters of model-based optimization that are usually tuned by hand. This process improves the performance of the method compared with manual tuning and reduces convergence time when using iterative solvers.

Deep Unfolding consists of converting an iterative algorithm into a Deep Neural-Network. This involves unfolding an iterative algorithm into a sequential procedure with a fixed number of iterations. Each iteration is then transformed into a layer of the neural network with the hyperparameters and any objective rules parameters for the iteration as layer parameters. This makes it possible to have different parameters for each iteration, which optimises the algorithm and increases its parameterisation and abstraction.

These methods make it possible to use domain knowledge, resulting in a more task-specific method that is above all interpretable and reliable, while at the same time improving performance. The use of deep learning eliminates the lack of rigour caused by manual tuning of hyperparameters, and in the case of mismatches and approximation errors in the mathematical model, improves performance thanks to the information contained in the data. In deep unfolding, the use of deep learning also improves the speed of the model, as the number of iterations is fixed and there are often fewer than in a classic model-based approach. The use of domain knowledge greatly reduces the number of method parameters, because in deep unfolding, for example, the neural network parameters are only those of the iterative algorithm. Training them therefore requires a much smaller database and reduced computational effort, so energy consumption is also greatly reduced.

In the context of estimating the direction of arrival of signals, it would have for example been possible to use the deep unfolding method on the maximum likelihood method to optimise the abscissa search for the source of the maximum likelihood in the gradient descent. Seeing how popular such methods have been in recent times, we can expect rich documentation on the subject in the incoming years. Once that happens, a new door will open to perform similar research as we did in this article once again.

7 Conclusion

In this study, we embarked on a comprehensive exploration of various Direction of Arrival (DOA) estimation methods under practical disruptive conditions, including signal-to-noise ratio (SNR), correlation between source signals, variance ratio between source signals, and positional perturbation. Our investigation spanned traditional methods such as Beamforming, MUSIC, and Maximum Likelihood, alongside deep learning approaches represented by a Convolutional Neural Network (CNN) and its custom variant designed to enhance robustness against the identified disruptions.

Our findings highlight the nuanced performances of these methods across different scenarios, underscoring the intricate balance between accuracy, computational complexity, and resilience to external disturbances. Notably, the MUSIC method demonstrated remarkable robustness and precision in environments with minimal disruptions, asserting its superiority in handling closely spaced sources and moderate levels of noise. Conversely, the Beamforming technique, while beneficial for its simplicity and computational efficiency, showed significant susceptibility to increased variance ratios and positional perturbations, limiting its applicability in more challenging conditions.

The deep learning models, particularly the custom-trained CNN, presented a compelling alternative, showcasing an impressive ability to adapt to complex disruptive parameters. This model not only withstood high variance ratios and correlation coefficients but also excelled in scenarios with positional perturbations, marking a significant step forward in leveraging artificial intelligence for signal processing applications. However, the reliance on extensive training and the need for large datasets underscore the practical considerations of implementing such models in real-world scenarios.

Through our Monte Carlo approximation approach, we provided a rigorous statistical foundation for evaluating the robustness of each method, offering insights that are critical for both theoretical advancements and practical applications in signal processing. Our analysis revealed the importance of method selection, tailored to the specific challenges of the operational environment, to achieve optimal DOA estimation performance.

Looking ahead, the integration of model-based approaches with deep learning, as hinted at in our discussion of hybrid model-based deep learning methods, represents an exciting frontier for future research. Such methodologies promise to harness the strengths of both domains, potentially leading to breakthroughs in signal processing that are both highly accurate and efficiently adaptable to varied conditions.

In conclusion, this study not only sheds light on the relative performances of various DOA estimation methods under disruptive conditions but also sets the stage for future innovations in the field. As we continue to push the boundaries of what is possible with signal processing technologies, the insights garnered from this research will undoubtedly contribute to the development of more robust, accurate, and efficient DOA estimation methods, capable of meeting the demands of increasingly complex application environments.

Glossary

Convolutional Neural Network (CNN): A class of deep neural networks, commonly applied to analyzing visual imagery, which employs a mathematical operation known as convolution. In the context of DOA estimation, CNNs are utilized for their ability to learn spatial hierarchies of features from signal data.

Cramer Rao Lower Bound: A theoretical lower bound for the variance of estimators of a parameter, offering a measure of the best possible accuracy any unbiased estimator can achieve. It serves as a benchmark to evaluate the efficiency of DOA estimation methods.

Deep Learning: A subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. It is especially useful in handling large volumes of data, as seen in DOA problem solutions using CNNs.

Direction Of Arrival (DOA) problem: The challenge of determining the direction from which a received signal was transmitted, which is crucial in many signal processing applications such as radar, sonar, and wireless communications.

Eigenvalue, Eigenvector: In linear algebra, an eigenvalue is a scalar that indicates how much the direction of an eigenvector is stretched during a linear transformation. Eigenvectors and eigenvalues are fundamental in the analysis of linear transformations that are key in separating signal and noise subspaces in methods like MUSIC for DOA estimation.

Layer (in deep learning): A building block of neural networks, consisting of a set of neurons, where each layer's output is sequentially fed as input into the next layer. Layers are crucial in deep learning models for extracting features from input data.

Learning Rate: A hyperparameter that controls the amount by which the weights of a neural network are updated during training. It is a critical factor in training deep learning models, affecting both the speed and quality of learning.

Loss function: A function that measures the difference between the actual and predicted outputs in training machine learning models. Minimizing this function is central to the process of learning from data.

Mean Square Error (MSE): A measure of the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. MSE is widely used as a loss function in regression problems and to evaluate the performance of DOA estimation methods.

Monte Carlo Approximation: A computational algorithm that relies on repeated random sampling to obtain numerical results, typically used to simulate the behavior of complex systems and processes. In the context of DOA estimation, it is used to approximate the distribution of estimator performances under various conditions.

Optimizer: An algorithm or method used to change the attributes of the neural network such as weights and learning rate in order to reduce the losses. Optimizers are essential in the training phase of deep learning models.

Signal-Noise Ratio (SNR): A measure used in science and engineering to quantify the level of a desired signal to the level of background noise. SNR is a critical factor in evaluating the performance of DOA estimation methods under varying conditions.

Snapshot: In signal processing, a snapshot is a single instance of data or a signal vector collected by an array of sensors at a specific point in time. The analysis of snapshots across time is pivotal in DOA estimation techniques.

References

[1] Krim, H., & Viberg, M. (1996). Two decades of array signal processing research: the parametric approach. IEEE signal processing magazine, 13(4), 67-94.

[2] Stoica, P., & Nehorai, A. (1989). MUSIC, maximum likelihood, and Cramer-Rao bound. IEEE Transactions on Acoustics, speech, and signal processing, 37(5), 720-741.

[3] Ottersten, B., Viberg, M., Stoica, P., & Nehorai, A. (1993). Exact and large sample maximum likelihood techniques for parameter estimation and detection. Radar Array Processing, 99-151.

[4] Papageorgiou, G. K., Sellathurai, M., & Eldar, Y. C. (2021). Deep networks for direction-of-arrival estimation in low SNR. IEEE Transactions on Signal Processing, 69, 3714-3729.

[5] Elbir, A. M. (2020). DeepMUSIC: Multiple signal classification via deep learning. IEEE Sensors Letters, 4(4), 1-4.

[6] Shlezinger, N., Eldar, Y. C., & Boyd, S. P. (2022). Model-based deep learning: On the intersection of deep learning and optimization. IEEE Access, 10, 115384-115398.

[7] Stoica, P., & Nehorai, A. (1990). Performance study of conditional and unconditional direction-of-arrival estimation. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(10), 1783-1795.

A Appendix

A.1 Signal-to-Noise Ratio

-20	-15	-10	-5	0	5	10	15	20
The signal's power is 100 times weaker than the noise's power	The signal's power is 31.6 times weaker than the noise's power	The signal's power is 10 times weaker than the noise's power	The signal's power is 3.16 times weaker than the noise's power	The signal and noise have the same power	The signal's power is 3.16 times stronger than the noise's power	The signal's power is 10 times stronger than the noise's power	The signal's power is 31.6 times stronger than the noise's power	The signal's power is 100 times stronger than the noise's power

deciBel as a unit followe a logaritmic scale. The operation to get the linear signal-to-noise ratio from the deciBel one is: 10^(SNR(dB)/10)

Figure 10:

A.2 Additional Figures

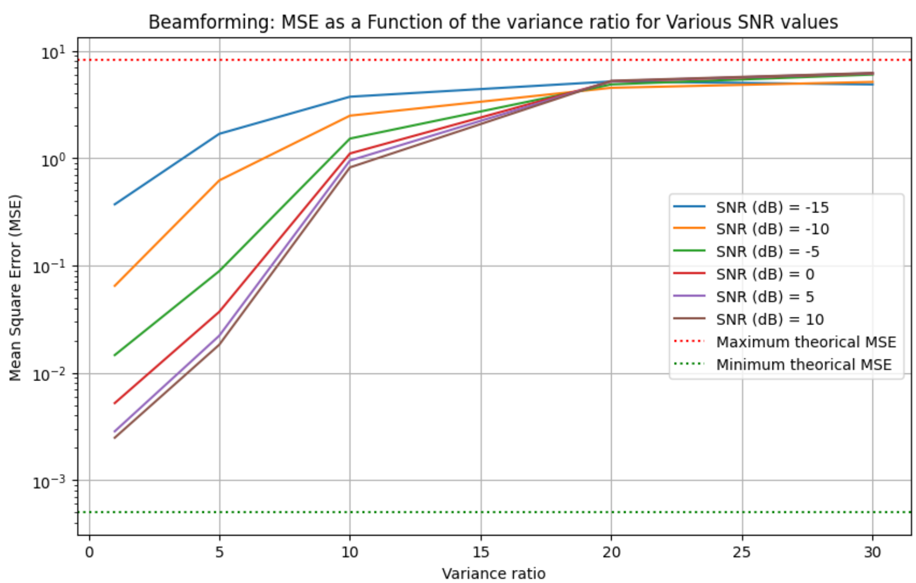

Figure 11: Beamforming: Comparing MSE for various variance ratio and SNR values

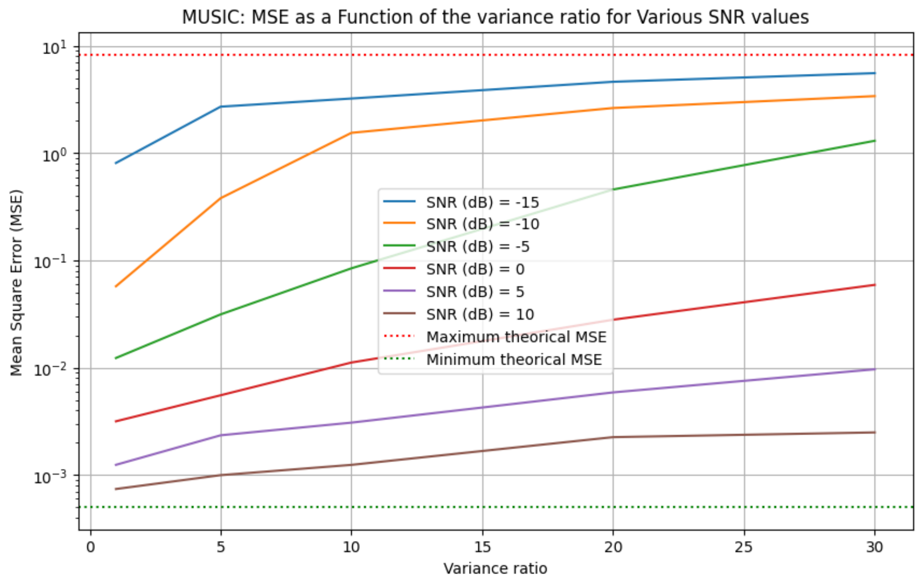

Figure 12: MUSIC: Comparing MSE for various variance ratio and SNR values

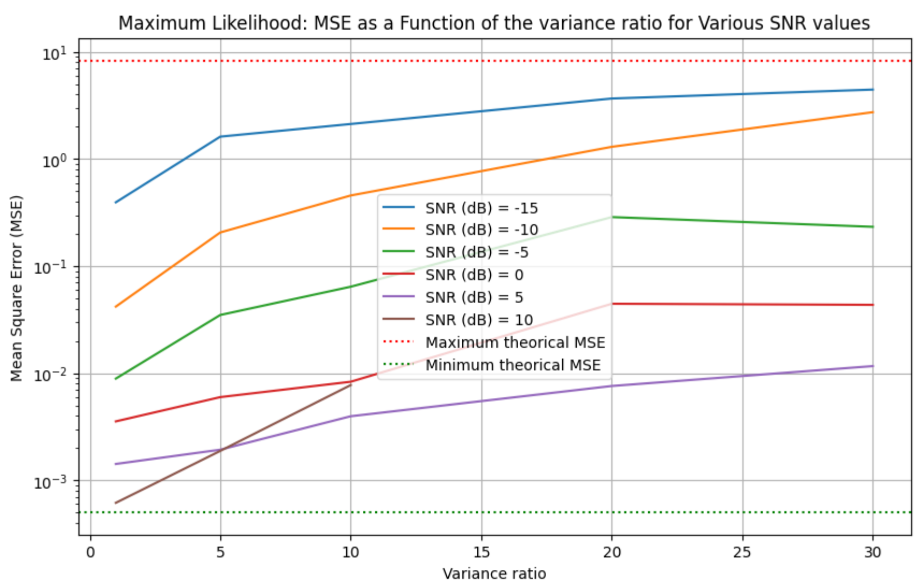

Figure 13: Maximum Likelihood: Comparing MSE for various variance ratio and SNR values

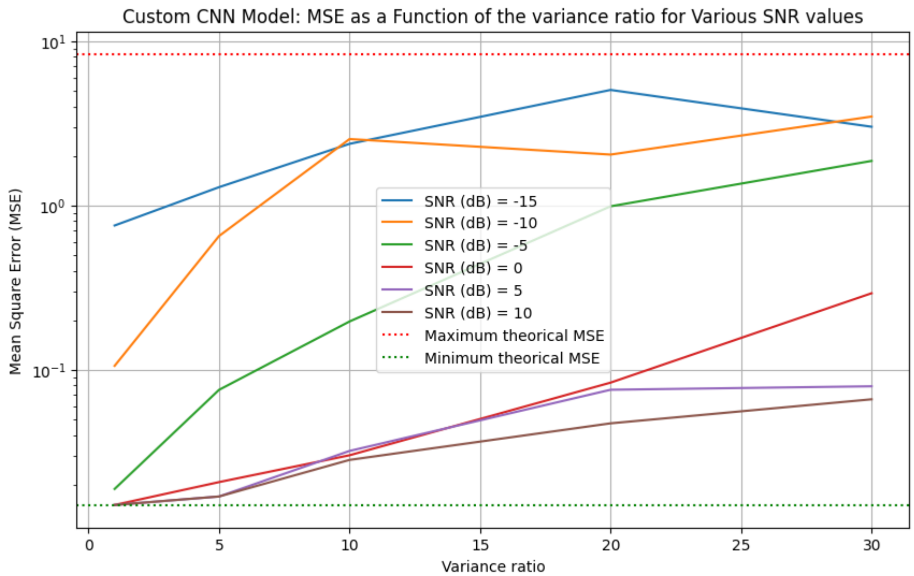

Figure 14: Custom CNN: Comparing MSE for various variance ratio and SNR values

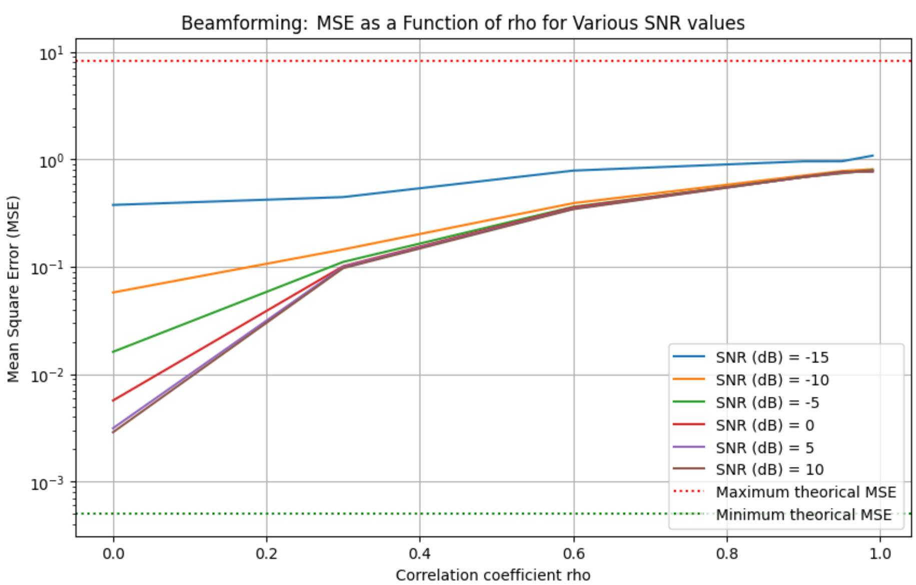

Figure 15: Beamforming: Comparing MSE for various rho and SNR values

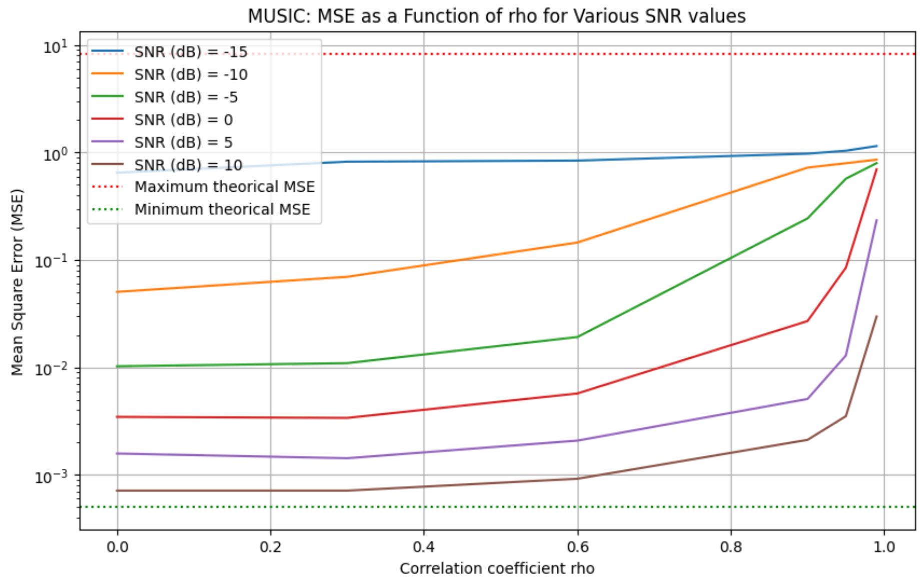

Figure 16: MUSIC: Comparing MSE for various rho and SNR values

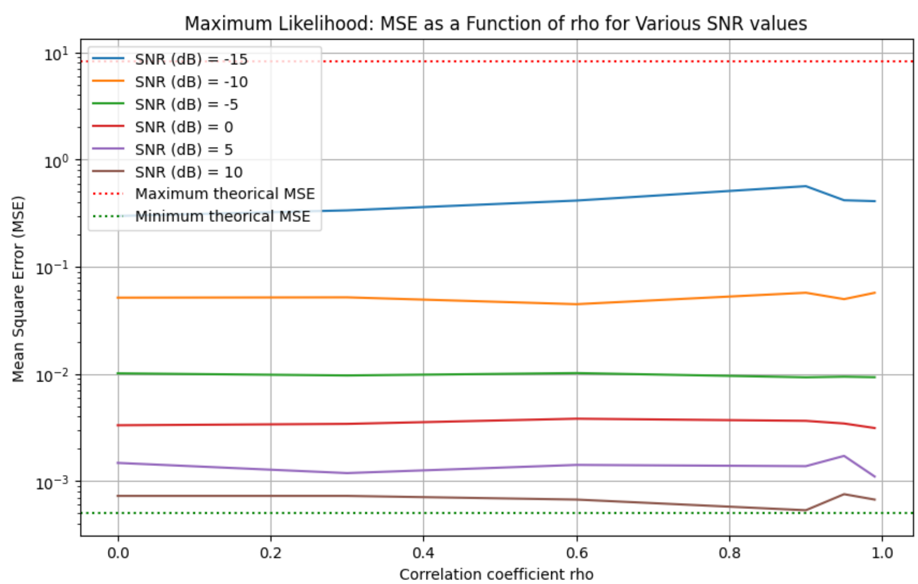

Figure 17: Maximum Likelihood: Comparing MSE for various rho and SNR values

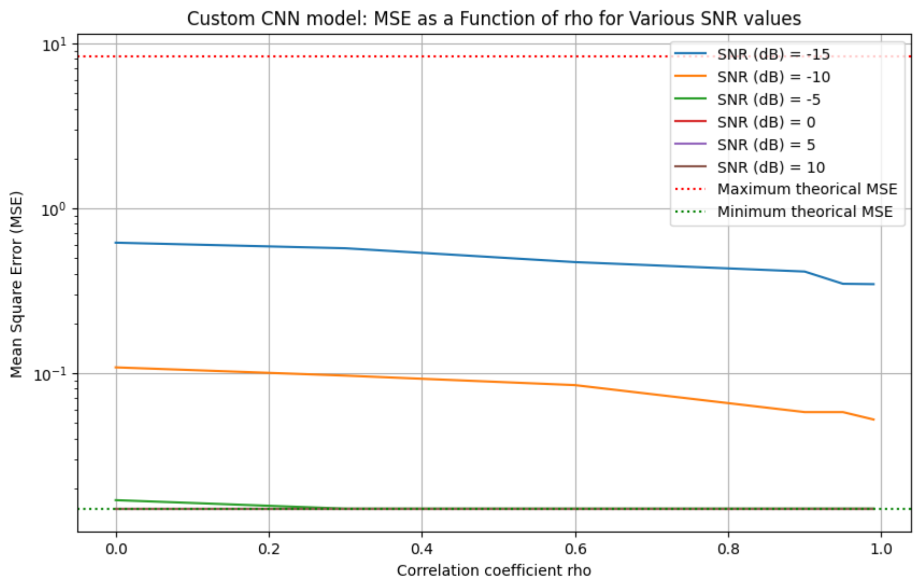

Figure 18: Custom CNN: Comparing MSE for various rho and SNR values

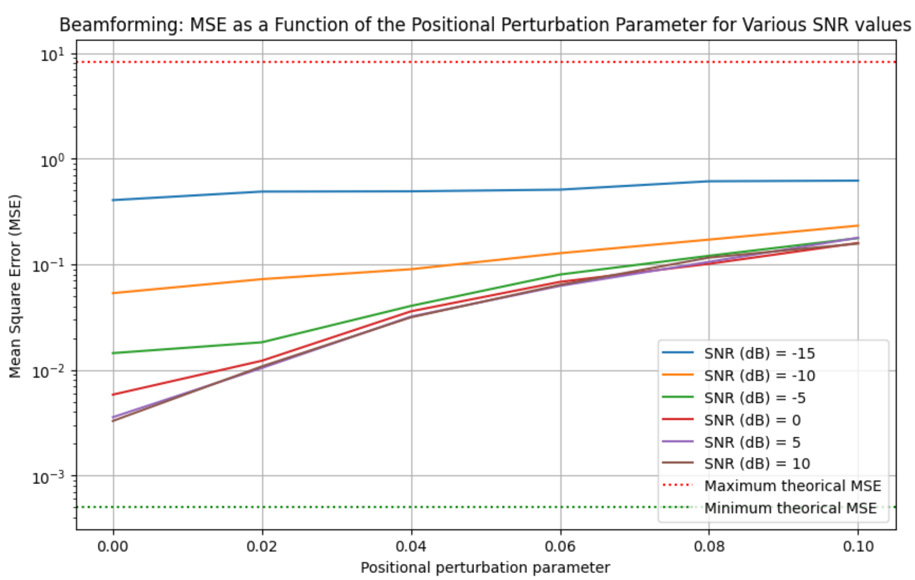

Figure 19: Beamforming: Comparing MSE for various positional disruption and SNR values

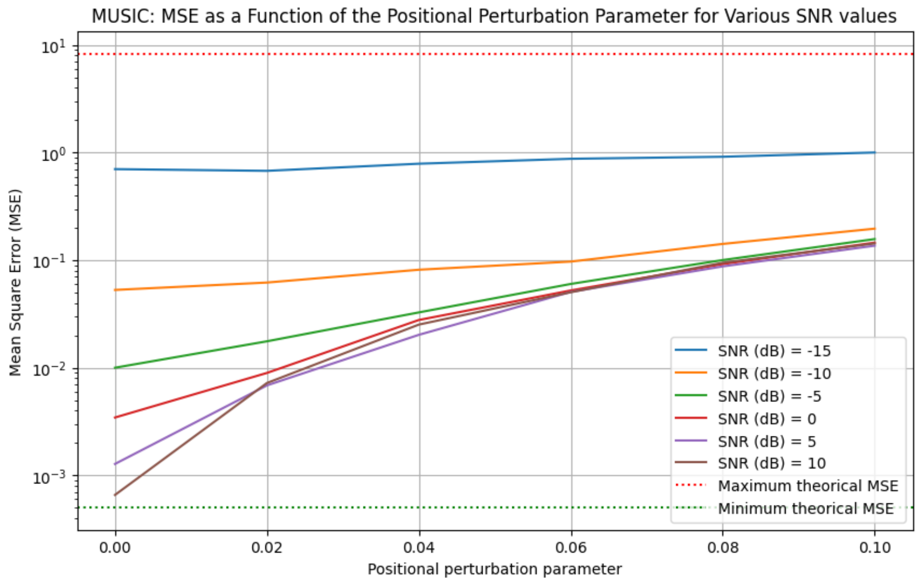

Figure 20: MUSIC: Comparing MSE for various positional disruption and SNR values

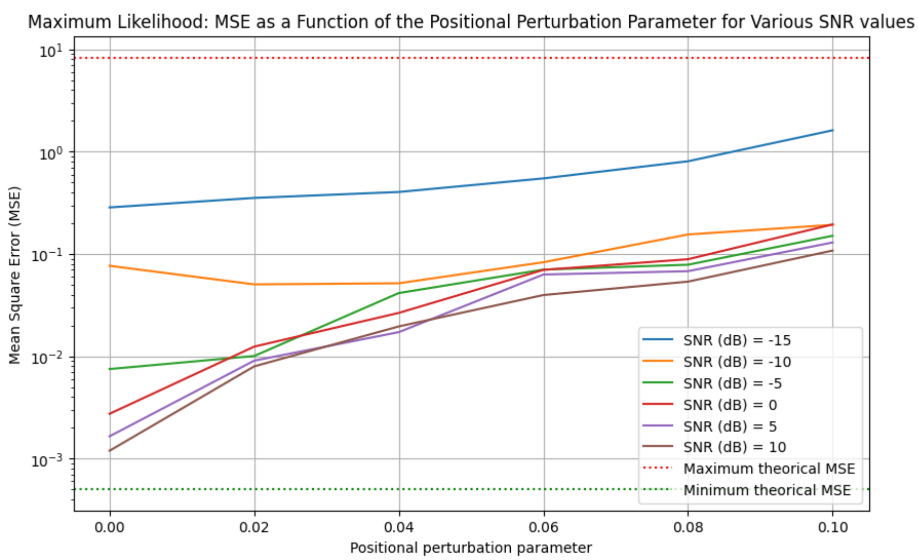

Figure 21: Maximum Likelihood: Comparing MSE for various positional disruption and SNR values

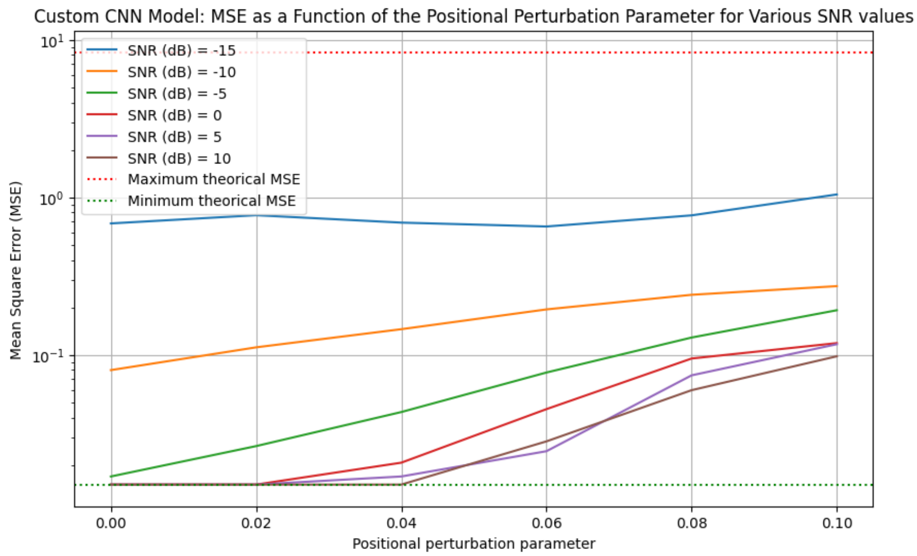

Figure 22: Custom CNN: Comparing MSE for various positional disruption and SNR values

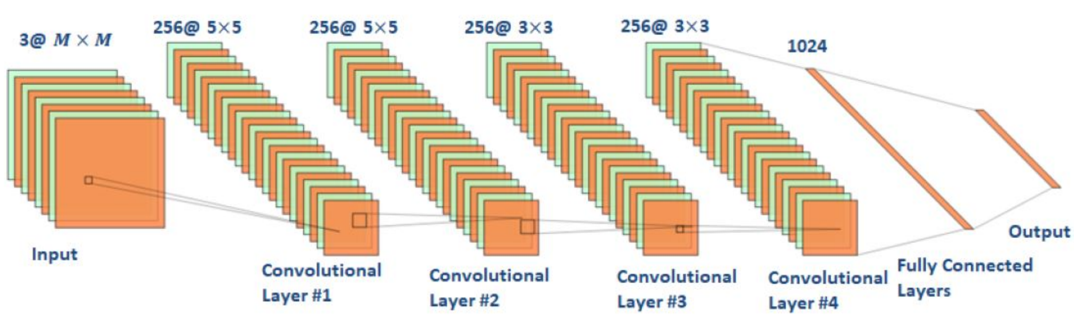

Figure 23: Deep Neural Network architecture for one subregion [5]