

**LEARNING-BASED DIRECTION-OF-ARRIVAL ESTIMATION WITH
DISTRIBUTED SPARSE ARRAYS IN UNDERWATER
ENVIRONMENTS**

A Thesis
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree
MASTER OF SCIENCE

by
Shelley Su
August 2025

Thesis Approvals:

Fauzia Ahmad, Thesis Advisor, Electrical and Computer Engineering
Yimin D. Zhang, Electrical and Computer Engineering
Li Bai, Electrical and Computer Engineering

©
Copyright
2025

by

Shelley Su

All Rights Reserved

ABSTRACT

Source direction-of-arrival (DOA) estimation is an important objective in ocean surveillance. Vertical line arrays with uniform sensor spacing are commonly employed for this purpose. However, for a given number of sensors, these arrays provide limited aperture and lower degrees-of-freedom for DOA estimation as compared to sparse arrays that employ non-uniform sensor spacings. When there are size, weight, power, or cost constraints, distributed configurations of sparse arrays can be the right choice, especially when sonobuoys or unmanned underwater platforms are used for their deployment. Under a centralized processing framework, noncoherent processing relaxes the stringent calibration requirements imposed by coherent algorithms on distributed arrays. As the spatial covariance matrix corresponding to distributed sparse arrays is typically sparse with missing elements, high-resolution subspace-based techniques, such as Multiple Signal Classification (MUSIC), provide degraded performance, especially in the presence of array imperfections due to ocean currents and environmental gradients. In contrast, a pre-trained machine learning (ML) method that can infer the number of sources and estimate their DOAs accurately under array imperfections would be computationally better suited toward real-time operation.

In recent years, research has been conducted using ML methods for DOA estimation with uniform and sparse line array geometries, including distributed configurations. However, the focus of the works on distributed configurations has been primarily on additive white Gaussian noise assumptions. In the context of underwater surveillance, ocean ambient noise often deviates from this assumption as it arises from multiple sources, including ocean turbulence, marine life, geological processes, and commercial or industrial shipping activities. As such, its statistical characteristics vary based on location and time. In addition, multiple sensors in the array may pick up similar noise sources, resulting in spatial correlations. Leveraging existing ML-based schemes designed for white Gaussian noise assumptions, this thesis aims to devise ML-based solutions to estimate DOAs of multiple sources in the pres-

ence of underwater ambient noise with distributed sparse arrays. We focus on quiet deep-ocean waters, far from the shipping lanes, where ambient noise can be approximated as spatially-correlated Gaussian noise. Using simulated covariance matrices from a distributed system of three sparse subarrays with sensor imperfections as input, we demonstrate how a white Gaussian noise based ML model can be successfully adapted for spatially-correlated Gaussian noise using warm start or few shot learning. The proposed solutions yield capable pre-trained ML models that offer flexible and lightweight solutions for real-time operation in the field.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABBREVIATIONS	ix
CHAPTER	
1 INTRODUCTION	1
1.1 Ocean Ambient Noise	3
1.2 Array Imperfections	4
1.3 Proposed Method	6
2 MODELS AND BACKGROUND	9
2.1 Notation	9
2.2 Signal Model	10
2.3 Multiple Signal Classification (MUSIC)	12
2.4 Correlated Noise Model	13
2.5 Multi-layer Perceptron (MLP)	14
2.6 Warm Start Learning	16
2.7 Few-Shot Meta-Learning	16
3 DOA ESTIMATION IN WHITE GAUSSIAN NOISE	20
3.1 Simulation Setup	20
3.1.1 Distributed Array Configuration	20
3.1.2 Dataset Generation	21
3.2 Preliminary Investigations	22
3.2.1 Uniformly-spaced Sources	22
3.2.2 Testing With Unseen Data	23
3.2.3 Model and Dataset Adjustments	25

	Page
3.3 Model Architecture	26
3.3.1 Fixed Number of Sources	27
3.3.2 Varying Number of Sources	28
3.3.3 Detection Tolerance	29
3.4 DOA Estimation Results	30
3.5 Conclusion	33
4 DOA ESTIMATION IN SPATIALLY-CORRELATED GAUSSIAN NOISE	34
4.1 WGN-Trained Models	34
4.1.1 Experimental Setup	34
4.1.2 Classification Results	35
4.1.3 Remarks	37
4.2 Warm Start Learning	37
4.2.1 Experimental Setup	38
4.2.2 Classification Results	38
4.2.3 Remarks	42
4.3 Few-Shot Meta-Learning	42
4.3.1 Experimental Setup	42
4.3.2 Classification Results	45
4.3.3 Conclusion	48
5 CONCLUSION AND FUTURE WORK	50
REFERENCES	52

LIST OF TABLES

Table	Page
3.1 Results of Baseline Models in WGN.	31
3.2 Results of Baseline Ensemble Models on Test Data in WGN.	33
4.1 AR(1) Noise & WGN-Trained Models, Varying α -Values.	36
4.2 AIC False Recovery for AR(1) Noise.	37
4.3 AR(2) Noise & WGN-Trained Models, Correlated α -Values.	41
4.4 Warm Start Learning: AR(1) Noise $\alpha_1 = 0.9$	43
4.5 Warm Start Learning: AR(2) Noise $\alpha_1 = 0.50, \alpha_2 = 0.49$	44
4.6 Few Shot Learning: AR(1) Noise $\alpha = 0.9$	47
4.7 Few Shot Learning: AR(2) Noise $\alpha_1 = 0.50, \alpha_2 = 0.49$	48

LIST OF FIGURES

Figure	Page
1.1 Vertical line array sonobuoy.	2
2.1 An example of an MLP with a single hidden layer.	15
2.2 Composition of few-shot meta-learning task.	17
3.1 Distributed sparse array configuration.	21
3.2 Loss versus epoch for five sources using hold-out testing.	23
3.3 Loss versus epoch for five sources using out-of-sample testing.	24
3.4 Example of incorrect source prediction with unseen data.	25
3.5 MLP architecture.	27
3.6 Example of prediction within detection tolerance of 1°	30
3.7 Distribution of AIC model selection.	32
4.1 Test accuracy (0° tolerance) for warm start learning & AR(1) data. . .	39
4.2 Test accuracy (0° tolerance) for warm start learning & AR(2) data. . .	40

ABBREVIATIONS

AIC	Akaike Information Criterion
AR	Autoregressive Process
AR(1)	First-order Autoregressive Process
AR(2)	Second-order Autoregressive Process
BCE	Binary Cross Entropy
dB	Decibel
CNN	Convolutional Neural Network
DMLP	Deep Multi-layer Perceptron
DOA	Direction-of-Arrival
EVD	Eigenvalue Decomposition
FNN	Feed-forward Neural Network
GARCH	Generalized Auto-regressive Conditional Heteroskedasticity
ML	Machine Learning
MAML	Model-Agnostic, Meta-Learning
MLP	Multi-layer Perceptron
MUSIC	Multiple Signal Classification
NOAA	National Oceanic and Atmospheric Administration
ONC	Ocean Networks Canada
SNR	Signal-to-Noise Ratio
ULA	Uniform Line Array
USU	Underwater Search Units
UUV	Unmanned Underwater Vehicles
VLA	Vertical Line Array
WGN	White Gaussian Noise

CHAPTER 1

INTRODUCTION

Direction-of-arrival (DOA) estimation of acoustic sources is crucial for effective ocean surveillance and monitoring [1, 2]. Typically, an array of acoustic sensors, or hydrophones, is deployed for DOA estimation which quite often assumes a vertical linear structure in underwater applications. Traditionally, the vertical line array (VLA) is a uniform linear array (ULA), having a uniform inter-element spacing of one-half wavelength. Sparse line arrays, on the other hand, employ non-uniform sensor placements [3–7], thereby offering advantages over ULAs when applications impose cost, size, or weight constraints. Available methods for processing VLA measurements to estimate source DOAs are dictated by the employed array configuration [1, 8–15].

Underwater platforms can often benefit from array configurations with fewer sensors. A towed array, which is a system of hydrophones towed behind a submarine, can be kilometers long [16] and often multiple arrays are towed [17]. Minimizing its footprint can translate to reducing infrastructure requirements for storage and the towing platform. Similarly, unmanned underwater vehicles (UUV) or underwater search units (USU) are other mobile platforms that are limited by the surface area of the vehicle in case of body-mounted sensors or the weight the vehicle can tow if using a towed array.

Another type of underwater platform used for short-term monitoring is the sonobuoy [18, 19]. Sonobuoys can be deployed in the ocean from aircrafts or ships and can be equipped with sensor arrays, signal processing capabilities, and communication systems to transmit data to remote operators. These devices are also often constrained by size and weight. For instance, the contents of a sonobuoy itself must be contained within its deployment canister, which must house a float, cable housing, drogue, battery, electronics, terminal weight, and the hydrophone array, along with the array support structure. The canister of an A-size sonobuoy, for example, is approximately 36 inches in length and 4.88 inches in diameter, which severely limits the

number of hydrophones that can be used. Figure 1.1 shows a rendering of a deployed sonobuoy [18, adapted from] containing uniform upper and lower arrays. The drogue and terminal weight provide structural stability.

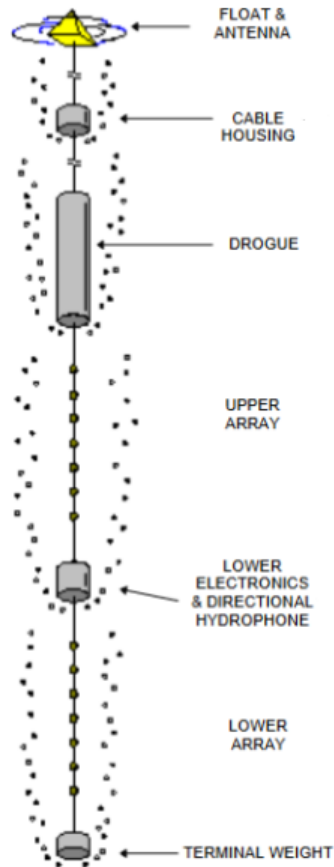


Figure 1.1. Vertical line array sonobuoy.

Sparse arrays require fewer sensors to achieve a given aperture as compared to a ULA. When a fixed number of sensors is considered, sparse arrays can provide higher resolution in beamforming due to larger apertures, and can estimate more sources than sensors due to higher degrees-of-freedom offered by the non-uniformity of its sensor placements. The severity of issues, such as mutual coupling between sensors which affect closely-spaced sensors in ULAs, may also be reduced when the array is sparse. However, the trade-off of these benefits is an increased signal processing complexity.

Distributed hydrophone array systems are also gaining traction as an effective means of ocean monitoring and surveillance [20, 21]. Efficient deployment of such systems is accomplished by smaller autonomous platforms or sonobuoys, which again imposes size, weight, and cost constraints on the subarrays themselves comprising the distributed configuration [22–24]. Coherent processing of the resulting measurements under a centralized framework, however, imposes strict calibration constraints on the distributed system. While coherent processing can result in higher resolution and array sensitivity, it also comes at the price of high computational overhead. Non-coherent processing may ease such constraints and permit reliable DOA estimation even under partly calibrated conditions [25–27].

Additionally, there are inherent aspects of the ocean environment that can render DOA estimation, whether using a single array or a distributed array, quite challenging. One such dominant factor is the ocean ambient noise, as discussed below.

1.1 Ocean Ambient Noise

A common challenge of signal processing for in-air and underwater applications is detecting a source signal embedded in noise [1, 28, 29]. However, characterizing ocean ambient noise can be a difficult task. Ref. [30, p.215] offers an apt summary of ocean ambient noise, saying that “[it] is eminently characterized by variability.” This is because many factors contribute to ocean ambient noise. These include tides and the hydrostatic effect of waves, seismic disturbances, ocean turbulence, nonlinear wave interactions, ship traffic, surface waves, thermal noise, and intermittent biological sounds [30–32].

In quiet deep-sea waters, far from the shipping lanes, surface waves or ocean turbulence may be the major contributor to noise. On the other hand, in shallow harbor waters, ship traffic, tides, and marine life would be dominant components. Circling back to the observation offered by Ref. [30, p.215], ocean ambient noise varies from location to location and is different at different depths. Noise arising from sources, such as turbulence and distant shipping, is typically approximated as

colored Gaussian, while noise arising from site-specific sources, e.g., marine life in warmer waters and harbor ship traffic, generally have non-Gaussian characteristics.

Various statistical models have been proposed in the literature to capture the non-Gaussian characteristics of ocean ambient noise. Gaussian mixture model and Middleton Class-A models were used in [33]. A more recent work examines the use of alpha stable distribution to model ambient noise specific to the northern South China Seas [34]. However, the aforementioned models correspond to a single sensor. Interestingly, several accessible repositories also exist with underwater recordings from organizations, such as Ocean Networks Canada (ONC) and the National Oceanic and Atmospheric Administration (NOAA). Exploration of these repositories revealed that ONC and NOAA primarily provide single sensor data from a geographical network of hydrophones [35, 36], although ONC also provides limited recordings using geometric hydrophone arrays. Ref. [37] presents a MATLAB-based simulator which accepts shipping, wind, rain, turbulence, and thermal noise sources as inputs and outputs noise characteristics in the frequency domain. Generalized autoregressive conditional heteroskedasticity (GARCH) model was employed in [38, 39] to fit real array measurements in sea environments.

From an array processing perspective, an important characteristic of ocean ambient noise is its spatially correlated nature across the array. Refs. [40, 41] developed models for spatial correlation in the ocean caused by surface generated noise, while Ref. [42] analyzed experimental data to examine the spatial correlation arising from turbulence caused by towed arrays. Spatially correlated noise is modeled as a first-order autoregressive (AR(1)) process in [43] to analyze the detection performance and in [44] for performance evaluation of sparse reconstruction based DOA estimation, both using co-prime sparse arrays.

1.2 Array Imperfections

Subspace-based techniques, such as Multiple Signal Classification (MUSIC) [45], are widely used for DOA estimation due to their high-resolution capabilities. However,

MUSIC's performance relies on precise array calibration and the assumption of white Gaussian noise (WGN). As a result, imperfections in the array and the presence of colored noise can lead to degraded DOA estimation performance [44, 46, 47].

Regardless of uniform or sparse configurations, array imperfections can be a common occurrence. These can arise from various sources, including manufacturing imperfections, environmental changes, and sensor position perturbations in the array. In the underwater environment, array imperfections can be exacerbated by ocean currents, temperature variations, and pressure changes. The impact of these imperfections is typically modeled as gain and phase errors in the signal measurement. Moreover, as previously stated, ocean ambient noise can exhibit colored Gaussian characteristics at certain depths and geographical locations, while also following non-Gaussian distributions at other depths and locations. Consequently, the reliability of subspace techniques, e.g., MUSIC, can diminish under such conditions, necessitating the consideration of alternative approaches capable of easily adapting to array imperfections and different noise environments. Therefore, a signal processing solution that accounts for ambient noise characteristics and array imperfections for a distributed system can be particularly useful for underwater applications.

Most approaches proposed to address array imperfections or colored noise in DOA estimation with sparse arrays do not consider distributed configurations [48–53]. Existing works focusing on DOA estimation with distributed sparse arrays under partly calibrated conditions assume the presence of WGN [25, 26]. In these works, only knowledge of the relative sensor positions within each subarray is assumed. Raw measurements at each subarray are processed locally to produce subarray covariance matrices. The covariance matrix from each subarray is sent to the central processing station, where these are fused to generate the overall covariance matrix, followed by DOA estimations. In Ref. [26], the authors show that more sources can be detected with such non-coherent processing of subarrays than with each individual subarray for an imperfectly calibrated array. Building on this premise, Ref. [25] demonstrates how machine learning (ML) can be used for a non-coherent processing scheme to make accurate DOA estimations with partly-calibrated distributed sparse system. As both

works assume the presence of additive WGN, a signal processing solution that can generate reliable DOA estimations with a distributed sparse array and adapt to array imperfections and the presence of spatially correlated noise would be a valuable asset for underwater applications.

1.3 Proposed Method

Inspired by the work in [25], we explore the potential of ML as a promising non-coherent processing solution to this challenge. Unlike conventional covariance-based signal processing algorithms, ML algorithms can be trained to recognize patterns and make accurate predictions even in the presence of distributed array imperfections. By leveraging large datasets of simulated or real-world signals, ML models can potentially learn to account for array imperfections in a distributed system and provide accurate DOA estimations. This flexibility and adaptability make ML an attractive option for underwater signal processing applications.

In this thesis, we develop an ML framework that can overcome the limitations of subspace-based techniques, such as MUSIC, by effectively handling array imperfections and spatially correlated noise for a distributed system of sparse subarrays under the centralized non-coherent processing framework. More specifically, our approach focuses on training ML models to accurately estimate the DOAs of multiple sources with partly-calibrated subarrays with imperfections in the presence of both white and spatially correlated Gaussian noise. We assume knowledge of the relative sensor positions within each subarray, but lack knowledge of the distance between the subarrays. Due to the absence of open-source data for VLAs and ocean ambient noise, we train an ML model on simulated autoregressive (AR) noise. A multi-layer perceptron (MLP) architecture is considered for DOA estimation of both a fixed number and varying number of sources. A summary of the contributions is provided below.

- *An ML solution for DOA estimation of varying number of sources under WGN and array imperfections:* We begin by training an MLP in WGN. The initial architecture considered is an adapted 2-hidden layer MLP proposed by [25].

General observations are made on data generation and ML techniques that were implemented in order to train suitable, generalized, baseline models for unseen data. In the end, separate models are trained for fixed number of sources (one to five sources) as well as a single model which accounts for a varying number of sources. These models represent a DOA estimation solution when the noise characteristics can be approximated as white Gaussian and the array is not perfectly calibrated.

- *An ML solution via warm start learning for DOA estimation with partly-calibrated arrays in spatially correlated Gaussian noise:* Measurements containing AR(1) and second-order autoregressive (AR(2)) noise are directly applied to pre-trained models for WGN. This step assesses the model’s ability to generalize and maintain accuracy without further training in new noise conditions. Additionally, model selection using Akaike information criterion (AIC) is considered as a technique for source detection when varying number of sources are present. To improve the model’s adaptability to AR(1) and AR(2) noise, we employ warm start learning. This technique initializes the model with weights from the pre-trained MLP in WGN and fine-tunes it using varying amounts of data from the new spatially correlated noise environment. We leverage the pre-trained model’s knowledge and accelerate convergence to achieve robust DOA estimations in spatially correlated noise scenarios.
- *An ML solution with updated pre-trained WGN models via few-shot learning:* A few-shot meta-learning algorithm is implemented to enable the pre-trained model to adapt to new noise environments with minimal additional data. Few-shot meta-learning is highly suitable for underwater acoustic applications where model updates might need to be made quickly and accurately with a small amount of data. Meta-learning empowers the model to generalize across various noise types and offers another form of model adaptability.

To date, the work presented in this thesis has resulted in the following publications:

- Shelley Su and Fauzia Ahmad, "Data-driven direction finding with distributed sparse arrays under calibration errors," in *Proc. SPIE Defense + Commercial Sensing: Machine Learning from Challenging Data Conference*, vol. 13460, Article ID 13460-6, Orlando, FL, April 2025.
- Shelley Su and Fauzia Ahmad, "Few-shot learning for direction-of-arrival estimation in underwater environments using distributed sparse arrays," to be submitted to the *IEEE Underwater Acoustic Signal Processing Workshop*, Narragansett, RI, October 2025.

The remainder of the thesis is organized as follows. Chapter 2 presents an overview of foundational topics covered in this thesis, namely, the distributed signal model, MUSIC, spatially correlated autoregressive Gaussian noise models, MLP, warm start learning, and few-shot learning. MUSIC is summarized as a popular subspace approach to high-resolution DOA estimation, prior to discussing the ML concepts, i.e., MLP architecture, warm start training, and few-shot meta-learning. In Chapter 3, we outline the simulation approach for generating training, evaluation, and testing datasets and train/validate the MLP-based multilabel classifier for DOA estimation in WGN. Performance of the proposed model is evaluated using testing data in terms of classification accuracy. Chapter 4 examines the impact of autoregressive noise on MLP models trained under WGN assumption and also provide results of warm start training and few-shot learning. Lastly, Chapter 5 offers topics for future work.

CHAPTER 2

MODELS AND BACKGROUND

In this chapter, we present the signal model for distributed sparse arrays and the ocean ambient noise model considered in this thesis. The MUSIC algorithm and the basic structure of a multi-layer perceptron network are also described. We also provide an overview of warm start and few-shot learning.

2.1 Notation

Throughout this thesis, the following notations are used:

- A lowercase (uppercase) bold letter represents a vector (matrix).
- $(\cdot)^*$, $(\cdot)^T$, and $(\cdot)^H$ denote complex conjugation, matrix transpose, and matrix conjugate transpose (Hermitian) operations, respectively.
- \mathbf{I}_M is the $M \times M$ identity matrix.
- $\text{vec}(\cdot)$ returns the column-wise vectorization of its matrix argument.
- $\text{diag}(\cdot)$ returns a diagonal matrix with the elements of its vector argument on the main diagonal.
- $\text{triu}(\cdot)$ returns the elements constituting the upper triangular part of its matrix argument.
- $\mathbb{E}[\cdot]$ denotes the expectation operator.
- ‘ \circ ’ denotes the Hadamard product.
- $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ denote the real and imaginary parts of their complex-valued argument, respectively.

2.2 Signal Model

We consider a distributed configuration of K sparse subarrays, each comprising M hydrophones. The hydrophones in each subarray are assumed to be located at integer multiples of $\lambda/2$, with λ being the wavelength of operation. For P narrowband, uncorrelated sources at directions θ_p , $p = 1, 2, \dots, P$, the received signal vector at time t , corresponding to the k th subarray, can be expressed as

$$\mathbf{x}_k(t) = \sum_{p=1}^P \rho_k \mathbf{a}_k(\theta_p) s_p(t) + \mathbf{n}_k(t), \quad k = 1, 2, \dots, K. \quad (2.1)$$

Here, ρ_k denotes the phase shift at the reference sensor of the k th subarray due to location displacement, $s_p(t)$ is the signal waveform of the p th source, $\mathbf{n}_k(t)$ is the noise vector which is assumed to be uncorrelated with the source signals, and $\mathbf{a}_k(\theta_p)$ is the steering vector of the k th subarray for direction θ_p , given as

$$\mathbf{a}_k(\theta_p) = \left[1, e^{-\frac{j2\pi}{\lambda} d_{k2} \sin \theta_p}, \dots, e^{-\frac{j2\pi}{\lambda} d_{kM} \sin \theta_p} \right]^T, \quad (2.2)$$

with d_{km} representing the distance between the first and the m th hydrophone in the k th subarray. Defining the source signal vector as

$$\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_P(t)]^T, \quad (2.3)$$

and the k th subarray manifold matrix as

$$\bar{\mathbf{A}}_k = [a_k(\theta_1), a_k(\theta_2), \dots, a_k(\theta_P)], \quad k = 1, 2, \dots, K, \quad (2.4)$$

we can express the signal model in matrix-vector form as

$$\mathbf{x}_k(t) = \rho_k \bar{\mathbf{A}}_k \mathbf{s}(t) + \mathbf{n}_k(t), \quad k = 1, 2, \dots, K. \quad (2.5)$$

The signal model in (2.5) corresponds to the ideal case. In the presence of array imperfections, such as calibration errors, imperfect knowledge in sensor gain, and hydrophone position uncertainties, the array signal model in (2.5) can be modified to incorporate the resulting gain and phase errors as

$$\mathbf{x}_k(t) = \rho_k \mathbf{A}_k \mathbf{s}(t) + \mathbf{n}_k(t), \quad \mathbf{A}_k = \text{diag}(\mathbf{g}_k) \bar{\mathbf{A}}_k, \quad k = 1, 2, \dots, K, \quad (2.6)$$

where $\mathbf{g}_k = [\alpha_{k,1}e^{j\beta_{k,1}}, \dots, \alpha_{k,M}e^{j\beta_{k,M}}]^\top$ and $\alpha_{k,m}$ and $\beta_{k,m}$ are the respective gain and phase errors corresponding to the m th hydrophone of the k th subarray. The covariance matrix of the received signal vector for the k th subarray is given by

$$\mathbf{R}_k = \mathbb{E} [\mathbf{x}_k(t)\mathbf{x}_k^\text{H}(t)] = \mathbf{A}_k\mathbf{R}_s\mathbf{A}_k^\text{H} + \mathbf{R}_{n,k}, \quad (2.7)$$

where $\mathbf{R}_{n,k}$ is the noise covariance matrix corresponding to the k th subarray and $\mathbf{R}_s = \text{diag}([\sigma_1^2, \sigma_2^2, \dots, \sigma_P^2]^\top)$ is the source covariance matrix with σ_p^2 being the power of the p th source. We note that under the assumption of independent and identically distributed WGN, $\mathbf{R}_{n,k} = \sigma_{n,k}^2\mathbf{I}_M$ with $\sigma_{n,k}^2$ denoting the noise variance corresponding to the k th subarray. Also, in practice, \mathbf{R}_k is estimated as an average over T snapshots, that is,

$$\hat{\mathbf{R}}_k = \frac{1}{T} \sum_{t=1}^T \mathbf{x}_k(t)\mathbf{x}_k^\text{H}(t). \quad (2.8)$$

We assume a partially distributed processing framework, wherein each subarray transmits its locally computed covariance matrix to a central fusion center. Further processing to obtain the overall covariance matrix, corresponding to the distributed system of K subarrays, and subsequent DOA estimations are carried out at the fusion center. Since the sparse configuration and thus the extent of each subarray are not required to be identical, the subarray covariance matrices can be of different dimensions. We assume the covariance matrix of the largest subarray to be of size $L_0 \times L_0$, which also denotes the size of the overall covariance matrix at the fusion center when opting for noncoherent combining. We note that, unlike coherent combining, noncoherent combining only necessitates intra-subarray hydrophone position calibration and does not require inter-subarray distance calibration. Further, owing to the sparse configurations of the subarrays, the locally computed covariance matrices are sparse, which generally yield a sparse fused covariance matrix with missing elements. With these stipulations, the overall covariance matrix corresponding to the system of K sparse subarrays can be expressed as

$$\mathbf{R} = \left(\sum_{k=1}^K \hat{\mathbf{R}}_k \circ \mathbf{b}_k \mathbf{b}_k^\text{T} \right) \circ \mathbf{D}, \quad (2.9)$$

where the binary vector \mathbf{b}_k , defined as

$$[\mathbf{b}_k]_l = \begin{cases} 1, & \frac{l\lambda}{2} \in \mathbb{S}_k \\ 0, & \text{otherwise} \end{cases}, \quad (2.10)$$

indicates the presence or absence of a hydrophone at position $\frac{l\lambda}{2}$ in the k th subarray for $l \in [0, 1, \dots, L_0 - 1]$, and the matrix \mathbf{D} averages the redundant lag entries in $\hat{\mathbf{R}}_k$. The (u, v) th element of \mathbf{D} can be expressed as

$$[\mathbf{D}]_{u,v} = \frac{1}{\sum_{k=1}^K [\mathbf{b}_k \mathbf{b}_k^T]_{u,v}} + \epsilon, \quad (2.11)$$

where ϵ is a regularization constant applied to provide numerical stability.

Due to the Hermitian property of the covariance matrix, we only use the upper triangular elements of \mathbf{R} for the learning-based DOA estimation. That is, the data vector, \mathbf{r} , used as input to the learning model is given by

$$\mathbf{r} = [(\mathcal{R}(\tilde{\mathbf{r}}))^T (\mathcal{I}(\tilde{\mathbf{r}}))^T]^T, \quad \tilde{\mathbf{r}} = \text{vec}(\text{triu}(\mathbf{R})). \quad (2.12)$$

2.3 Multiple Signal Classification (MUSIC)

Once the fused covariance matrix, \mathbf{R} , has been computed, we can proceed with high-resolution DOA estimation by applying techniques such as subspace-based methods [45, 54, 55]. The MUSIC estimator of source DOAs is a popular subspace-based approach [45], with its pseudo-spectrum at angle θ is given by

$$P(\theta) = \frac{1}{\mathbf{a}_k(\theta)^H \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}_k(\theta)}, \quad (2.13)$$

where \mathbf{U}_n is the noise subspace $[\mathbf{u}_D, \mathbf{u}_{D+1}, \dots, \mathbf{u}_{L_0-1}]$ with $\mathbf{u}_D, \mathbf{u}_{D+1}, \dots, \mathbf{u}_{L_0-1}$ being the ordered eigenvectors of \mathbf{R} arranged from the largest corresponding eigenvalues to the smallest, and D denotes the signal subspace dimension. The MUSIC pseudo-spectrum in (2.13) is computed on a grid of DOAs covering the angular region of interest, and the locations of the peaks in the spectrum determine the source DOA estimates. Prior knowledge of the number of sources, i.e., D , is required for applying

MUSIC which can be determined using information theoretic criteria, such as AIC [56].

The performance of the MUSIC algorithm, however, degrades in the presence of correlated noise, which is typical of underwater environments. The MUSIC algorithm is also sensitive to calibration and positioning errors, as in the considered case of a distributed system of subarrays with imperfections. Also, in case of sparse covariance matrices with missing elements, matrix completion method is required prior to application of the MUSIC algorithm [57, 58], but these methods also suffer from performance degradation under array imperfections. The computational complexity associated with eigenvalue decomposition, together with that of matrix interpolation methods, also renders real-time source localization with subspace-based methods challenging. In contrast to MUSIC, deep learning-based approaches to DOA estimation can shift the computational complexity to offline training of the architecture and, at the same time, deal with array imperfections [25].

2.4 Correlated Noise Model

Spatial correlation is a salient feature of the ocean ambient noise. This type of noise, which is correlated with itself across the array, can negatively impact the performance of DOA estimation techniques, which are designed under the assumption of spatially white noise. Similar to [43], we assume the correlated noise to be wide-sense stationary Gaussian, and modeled as an auto-regressive process, expressed by

$$c[m] = \sqrt{\frac{\sigma_W^2}{\sigma_C^2}} w[m] + \sum_{i=1}^q \alpha_i c[m-i]. \quad (2.14)$$

Here, $w[m]$ is the input driving noise that is assumed to be white Gaussian with zero mean and variance σ_W^2 , α_i is the sensor correlation factor between sensors separated by $i\lambda/2$, σ_C^2 is the variance of the AR process, and q is the model order. The scale factor $\sqrt{\frac{\sigma_W^2}{\sigma_C^2}}$ in (2.14) is applied to ensure the correlated noise power equals σ_W^2 at a single sensor.

In this work, we focus on first-order and second-order AR noise processes, i.e., AR(1) and AR(2), for which the respective variances are given by [59]

$$\sigma_C^2 = \frac{1}{1 - \alpha_1^2} \sigma_W^2, \quad (\text{AR(1) model}) \quad (2.15)$$

$$\sigma_C^2 = \frac{(1 - \alpha_2)}{(1 + \alpha_2)(1 - \alpha_1 - \alpha_2)(1 + \alpha_1 - \alpha_2)} \sigma_W^2. \quad (\text{AR(2) model}) \quad (2.16)$$

Certain conditions for α_1 and α_2 must be met for the AR noise to be both stationary and stable. For an AR(1) model, the constraint is $0 \leq \alpha_1 < 1$, while the conditions for AR(2) model are

$$|\alpha_2| < 1, \quad \alpha_1 + \alpha_2 < 1, \quad \alpha_2 - \alpha_1 < 1.$$

2.5 Multi-layer Perceptron (MLP)

A multi-layer perceptron is an ML model that consists of an input layer, one or more hidden layers, and an output layer. The input layer, consisting of neurons or nodes with each representing a feature or input variable, receives the incoming data samples as their input. The output layer produces the final output of the network. It often consists of one or more neurons depending on the problem (e.g., classification or regression). Hidden layers lie between the input and output layers and perform complex computations through weighted neuron inter-connections, progressively transforming the input layer neuron activations into the output samples. All neurons in adjacent layers are fully connected, meaning each neuron in one layer is connected to every neuron in the next layer. Figure 2.1 depicts a simple example of the architecture of an MLP with one hidden layer. MLPs with more than three layers are often referred to as deep MLPs (DMLPs).

In an MLP, each inter-connection has a weight associated with it, which determines the strength of the connection. Each neuron implements an intermediate processing step by linearly combining its inputs using its inter-connection weights and adding a bias term. An activation function is applied to the intermediate result to create the output of the neuron. This function introduces non-linearity into the network, allowing it to learn complex patterns in the incoming data.

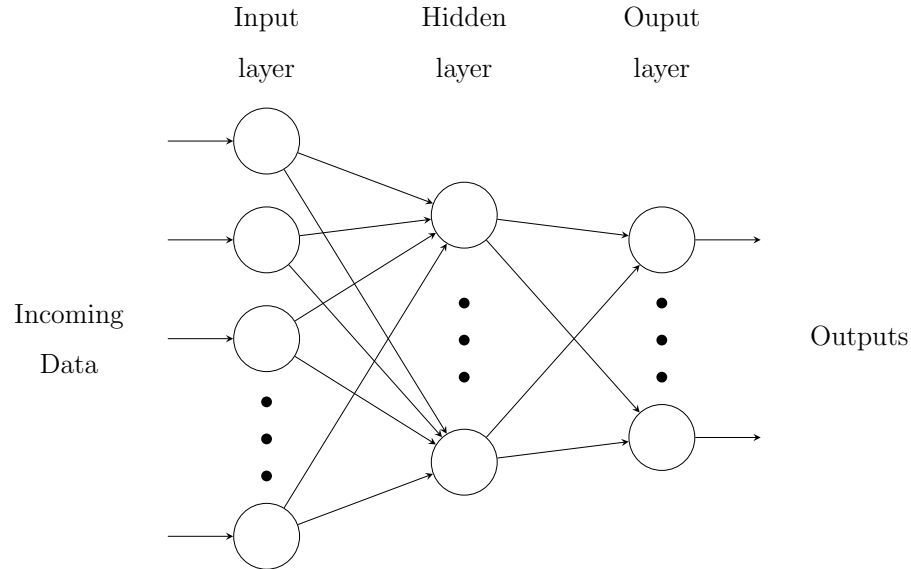


Figure 2.1. An example of an MLP with a single hidden layer.

During training, the network adjusts weights and biases of the inter-connections using optimization, such as gradient descent, to minimize a loss function. This process is known as backpropagation. The loss function is used to evaluate how well the MLP parameters are being learned. It can assume the form of a distance between the current output of the MLP and the expected output.

Denoting the input samples of the n th layer of an MLP by \mathbf{y}^n and the output samples by \mathbf{z}^n , the input-output relationship for the n th layer of the MLP can be expressed as

$$\mathbf{z}^n = g(\mathbf{W}^n \mathbf{y}^n + \mathbf{b}^n), \quad (2.17)$$

where \mathbf{W}^n denotes the weight matrix for all inter-connections between the n th and the $(n - 1)$ th layers, \mathbf{b}^n represents the bias vector, and $g(\cdot)$ is the activation function. This equation describes the forward pass of the network [60, 61]. In the backward pass, the loss function is optimized, thereby enabling adjustment or “learning” of the weights and biases of the MLP.

2.6 Warm Start Learning

Warm start learning refers to an ML technique where pre-trained weights and biases are utilized instead of starting from a random initialization. This approach is a type of transfer learning, which utilizes the knowledge gained from a previous training session, allowing the model to potentially achieve faster convergence and better perform on new data. Warm start learning is commonly used in transfer learning, where a model trained on a large dataset is fine-tuned on a smaller, domain-specific dataset. Warm start models have the potential to perform better on new data, particularly when the new task is similar to the original task. The pre-trained model parameters have extracted relevant features and help to reduce the risk of overfitting on new, smaller datasets.

Instances where warm start learning is not suitable occur when there are domain mismatches, i.e. when the original task and the new task are vastly different. In such cases, the benefit of pre-training is negated and the model may require substantial retraining. Another drawback of warm start training is catastrophic forgetting [62]. In other words, warm start training will update to the newer data but will be unable to accurately predict the original data it was trained on. Catastrophic forgetting is more prone to occur with domain mismatch. However, when the original and new domains are similar, warm start learning offers a clear benefit and serves as a useful tool when resources and/or new data are limited.

2.7 Few-Shot Meta-Learning

A more recent development in machine learning is few-shot learning [63–65]. In contrast to warm start training, which may require many additional training samples for the new task, the samples in few-shot training are in the range of one to five per class. Therefore, few-shot learning is particularly useful in cases where acquisition of large amounts of data is challenging or when we want to update a model on new tasks with limited data. Several types of few-shot learning methods exist in the

literature [66, 67]. In this work, we employ an optimization-based few-shot meta-learning approach, namely, model-agnostic meta-learning (MAML) [64].

Few-shot learning is an ML paradigm where a model is trained to learn and generalize on new tasks with a small number of examples. The terms K -shot and N -way are used to describe few-shot learning, where K is the number of samples for each class and N is the number of classes in a support set. The support set is used to adjust model parameters. A query set contains new examples not contained in the support set and is used to evaluate the model’s generalization ability. When combined, the query and support set form the basis for a task. Figure 2.2 shows an example of a 2-shot, 3-way multilabel classification task, where the multilabel contains any combination of labels for cat, dog, or rabbit, and the query set contains three unseen samples.

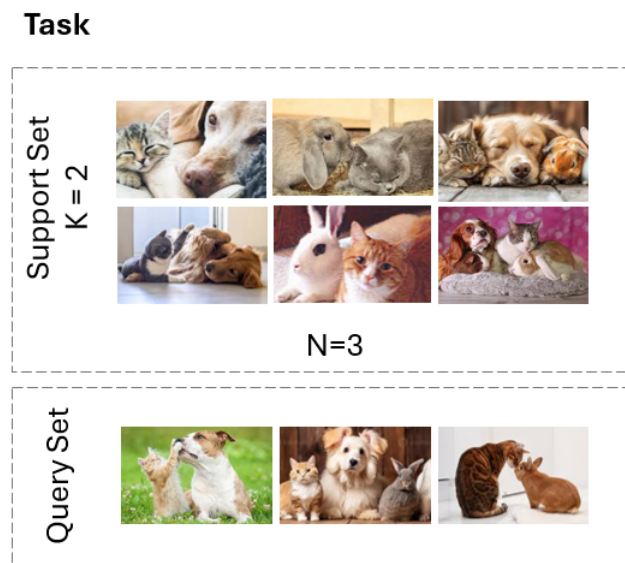


Figure 2.2. Composition of few-shot meta-learning task.

Meta-learning enables quick adaptation in few-shot learning scenerios. The concept of meta-learning is often described as “learning to learn”, or imparting ML models with the ability to notice similarities and differences between classes as opposed to learning only its features. For MAML, as the term model-agnostic implies, any type

of ML model that uses gradient-based learning can be used. MAML works by training the model on a variety of tasks sampled from a distribution. Training consists of an adaptation and optimization process for each task. In the adaptation process, model parameters are fine-tuned using gradients calculated on the task’s support set. Optimization occurs after the adaptation process by minimizing the evaluation loss on the task’s query set.

Few-shot meta-learning begins with a meta-learner, which we denote as a parameterized function f_φ . The meta-learner can be any user-defined neural network architecture initialized with either random or pre-defined weights. The MAML algorithm is a two-tiered approach, consisting of an inner loop for adaptation and an outer loop for optimization [64]. In the inner loop, support set samples, \mathcal{S}_i , undergo a forward pass, where the meta-learner makes predictions. A loss function, $\mathcal{L}(\cdot)$, is used to quantify how well the model predictions match the true labels. Gradient descent is performed on the backward pass and optimized such that the updated parameters after gradient descent are

$$\varphi'_i = \varphi - \mu \nabla \mathcal{L}_{\mathcal{S}_i}(f_\varphi), \quad (2.18)$$

where μ is the learning rate of the inner loop. The inner loop represents local optimizations using the support sets.

The adapted model parameters, φ'_i , from the inner loop are passed to the outer loop. In the outer loop, the query set, \mathcal{Q}_i , is used as input and the loss on predictions of the forward pass is optimized through gradient descent. That is, the objective is to minimize the loss on the query set,

$$\min_{\varphi} \sum_{\mathcal{Q}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{Q}_i}(f_{\varphi'_i}), \quad (2.19)$$

where $p(\mathcal{T})$ is the task distribution. The global parameters, φ , are updated by the outer loop as

$$\varphi \leftarrow \varphi - \nu \nabla_{\varphi} \sum \mathcal{L}_{\mathcal{Q}_i}(f_{\varphi'_i}), \quad (2.20)$$

where ν is the learning rate parameter of the outer loop. This process iterates over all tasks $\mathcal{T}_i \sim p(\mathcal{T})$. Algorithm 2.1 outlines the process for MAML.

Algorithm 2.1 Model-Agnostic Meta-Learning [64]

Input: $p(\mathcal{T})$: distribution over tasks

Input: μ, ν : step size parameters

- 1: randomly initialize φ
 - 2: **while** not done **do**
 - 3: Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
 - 4: **for** all \mathcal{T}_i **do**
 - 5: Evaluate $\nabla_{\varphi} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi})$ with respect to K examples
 - 6: Compute adapted parameters with gradient descent: $\varphi'_i = \varphi - \mu \nabla_{\varphi} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi})$
 - 7: **end for**
 - 8: Update $\varphi \leftarrow \varphi - \nu \nabla_{\varphi} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\varphi'_i})$
 - 9: **end while**
-

CHAPTER 3

DOA ESTIMATION IN WHITE GAUSSIAN NOISE

In this chapter, we consider source DOA estimation in WGN. We extend the learning method proposed in [25], wherein the study considered a fixed number of uniformly-spaced sources, to incorporate a varying number of sources with both uniform and non-uniform separations. The model extension enables estimation of the number of sources as well as their respective DOAs. The former capability eliminates the reliance on information theoretic criteria, such as AIC, for determining the number of sources prior to DOA estimation.

3.1 Simulation Setup

Distributed Array Configuration

We consider the distributed array configuration depicted in Figure 3.1, which is composed of three collinear subarrays with four hydrophones per subarray. This configuration is the same as in [25]. Specifically, the hydrophone locations in the three subarrays are given as

$$\begin{aligned}
 \mathbb{S}_1 &= \{0, 2, 5, 9\} \lambda/2, \\
 \mathbb{S}_2 &= \{0, 10, 11, 23\} \lambda/2, \\
 \mathbb{S}_3 &= \{0, 6, 14, 30\} \lambda/2.
 \end{aligned} \tag{3.1}$$

Clearly, the hydrophones in the subarrays are sparsely located and, as a result, the fused correlation matrix is sparse with missing elements. Also, the largest subarray aperture equals $30 \lambda/2$, which implies that $L_0 = 31$. That is, the fused covariance matrix is of dimension 31×31 .

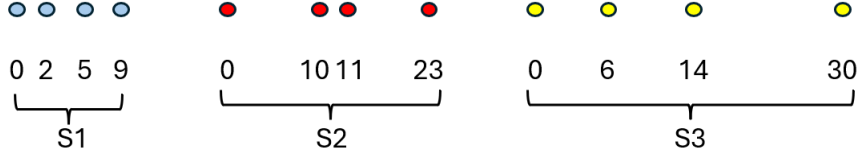


Figure 3.1. Distributed sparse array configuration.

Dataset Generation

MATLAB simulation of the signal model in Section 2.2 is used to generate the datasets for experimentation. The number of sources, P , is varied from one to five and the sources are assumed to be located within the angular sector $[-60^\circ, 60^\circ]$. A 1° grid is utilized which yields a total of 121 distinct DOAs. Varying source separations from 2° to 10° with 1° increment are considered and both uniform and non-uniform source DOAs are utilized. For each value of P , we determine different sets of unique uniform and non-uniform source DOAs. All sources are assumed to have the same power and the SNR is set as 0 dB. The antenna gains are independently generated from a uniform distribution between 0.9 and 1.1, whereas the phase errors are independently selected from a uniform distribution between -9° and 9° . $T = 200$ snapshots are utilized for each scenario to obtain the corresponding fused sample correlation matrix.

For the training dataset, we only use seven out of the nine source separations, omitting source separations of 4° and 8° . The validation dataset uses only the unseen separations of 4° and 8° . The testing dataset, on the other hand, utilizes all nine source separations. However, the sets of unique DOAs used for testing are different from those employed for training and validation. The data vectors containing the real and imaginary parts of the upper triangular elements of the fused correlation matrices and true, one-hot encoded, DOA classification labels are used as inputs to the learning model.

3.2 Preliminary Investigations

Uniformly-spaced Sources

To begin, 4,850 data samples are generated for five uniformly-spaced sources with 2° , 6° , or 10° separation. We split the dataset into 90% training and 10% test sets. Following the MLP architecture in [25], we define a model with an input layer, two hidden layers, and an output layer. The input layer corresponds to the dimensions of the vectorized covariance matrix, which is a vector of 930 elements. The output layer has 121 neurons, corresponding to the DOAs $\in [-60^\circ, 60^\circ]$ with 1° step. Lastly, the first and second hidden layers have 512 and 256 neurons, respectively.

Rectified linear unit (ReLU) activation is used on all hidden layers. Because the MLP model outputs a 121-element multilabel classification corresponding to the angular sector of interest, a sigmoid activation is used in the output layer to generate classification probabilities for each label. The sigmoid activation is coupled with binary cross entropy (BCE) loss, as BCE treats each label as a separate binary classification problem. The loss is minimized through gradient descent and is defined as

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \left[\mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i) \right], \quad (3.2)$$

where N is the number of training samples, $\hat{\mathbf{y}}$ is the predicted output and \mathbf{y} is the true value of the output. The first term, $(\mathbf{y}_i \log \hat{\mathbf{y}}_i)$ measures the loss for the positive class which represents source presence in the angular grid space. This term penalizes the model if the true label is 1 and the predicted probability is low. Similarly, the second term, $((1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i))$, provides a penalty if the true label is 0 and the predicted probability is high.

The model is trained using a 0.001 learning rate, *Adam* optimizer, and a batch size of 64. No regularization techniques, such as dropout or batch normalization, are used. Exact match accuracy is employed as our performance metric, which deems a multilabel classification as accurate only when all five DOAs are correctly identified.

We are able to obtain 100% accuracy on the training set and 99.3% accuracy on the test set. Figure 3.2 shows how both training and test losses decrease by epoch during training and about 10 epochs are needed for model convergence. In this scenario, the model is tested on data it encounters during training, i.e., the data always corresponds to five uniformly-spaced sources. For the MLP model to be useful, it needs to perform well on unseen data, which forms the premise for the next experiments.

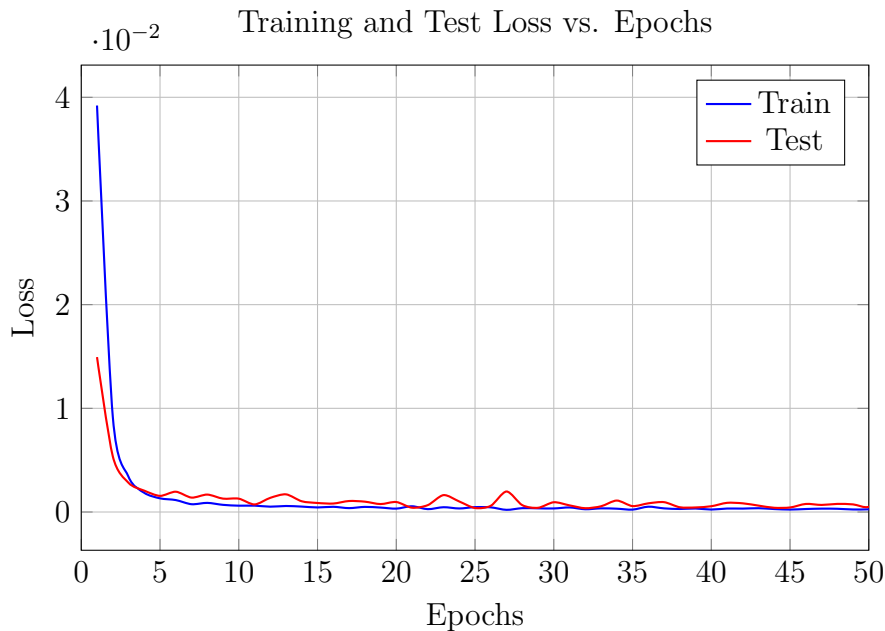


Figure 3.2. Loss versus epoch for five sources using hold-out testing.

Testing With Unseen Data

The MLP architecture in Section 3.2.1 is used for out-of-sample testing, where the model is tested with data withheld during training. Keeping with the 90% training and 10% test set ratio used previously, a 485-sample dataset of five sources with 4° and 8° uniform separations is generated for testing; these are angular separations not seen during model training. With no changes made to activation functions, loss, or learning rate, the result is 99.79% accuracy on the training set and 0.46% accuracy on the test set. To observe what happens on unseen data when a model is trained

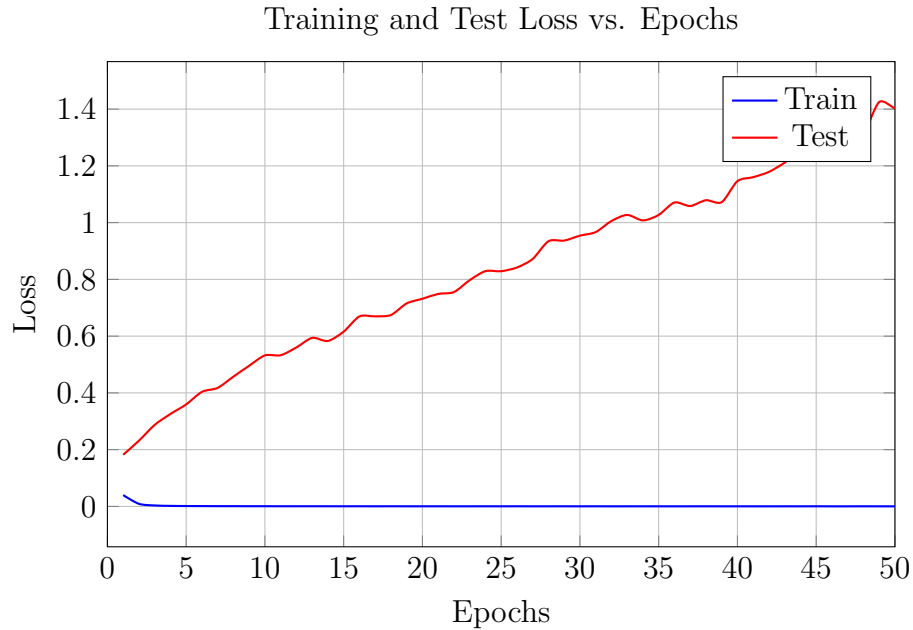


Figure 3.3. Loss versus epoch for five sources using out-of-sample testing.

on 2° , 6° , and 10° separations, we plot the BCE loss for training and test data in Figure 3.3. While the loss decreases quickly for the training set, the loss on unseen test data continually increases with each epoch. This is a characteristic indication that the model has overfit to training data. The low training loss, coupled with high test loss, indicates the model parameters are not generalized enough to make accurate predictions on unseen data. Despite the fact that the unseen dataset contains DOAs for five sources, the large majority of errors occur because the model underestimates the number of sources. Specifically, the trained model predicts an incorrect number of sources on the unseen data 89.49% of the time. An example of incorrect source number and DOA predictions is provided in Figure 3.4, where only two out of five sources are estimated with biased DOAs.

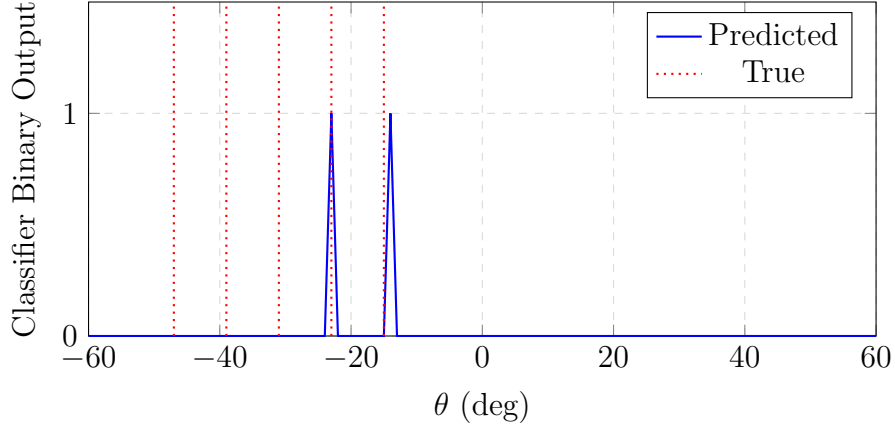


Figure 3.4. Example of incorrect source prediction with unseen data.

Model and Dataset Adjustments

Next, two changes are utilized to avoid overfitting. First, the composition of the training samples is changed from uniform spacing to non-uniform spacing to allow for more sample diversity. Models trained on uniform spacing yielded low accuracy on unseen data, and non-uniform spacing is expected to permit further generalization of the model. The change also better aligns with real-life scenarios, where sources are often non-uniformly spaced. Training sets contain data for DOAs separated by 2° to 10° separations, with 4° and 8° separations withheld. Validation sets only contain 4° and 8° separations, while test sets contain the full span of 2° and 10° separations.

Second, both batch normalization and dropout are used as regularization techniques in our model to prevent overfitting and enable further generalization of the model. Dropout prevents overfitting by randomly deactivating neurons in a layer during each training iteration. Batch normalization, on the other hand, adjusts the input to each layer by subtracting the mean and dividing by the standard deviation of the inputs within a mini-batch. Mathematically, batch normalization can be represented as [68]

$$\hat{\mathbf{y}}^n = \frac{(\mathbf{W}^n \mathbf{y}^n + \mathbf{b}^n) - \xi_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (3.3)$$

$$\mathbf{z}^n = g(\gamma^n \hat{\mathbf{y}}^n + \delta^n).$$

Recall (2.17), which describes the input-output relationship for MLP neurons. The input to batch normalization uses the output of the neuron, before the activation function is applied. For batch normalization, the mean and variance of the minibatch are calculated, denoted by ξ_B and σ_B^2 , respectively. A small term ϵ is applied for stability, and $\hat{\mathbf{y}}^n$ is the normalized output. \mathbf{z}^n is the final scaled and shifted output after applying the activation function and learnable parameters, γ^n and δ^n , from batch normalization. The learnable parameters ensure the network can still discern patterns in the data, even after the data has been normalized.

3.3 Model Architecture

In order to appropriately capture the features of more diverse datasets, more dense layers are required. The network architecture is, therefore, expanded from two to six hidden layers. Regularization in the form of dropout and batch normalization are added, as they support better prediction outcomes. ReLU remains the activation function for the hidden layers along with sigmoid activation for the output layer. The learning rate is adjusted down from 0.001 to 0.0001 for smoother convergence. *Adam* optimizer remains the same. The updated model architecture is depicted in Figure 3.5.

The number of neurons for input and output layers remain the same. However, the number of neurons in the hidden layers are first increased and then decreased between layers. Specifically, we use 1024 neurons in the first hidden layer, 2048 neurons in the second hidden layer, 4096 neurons each in the third and fourth hidden layers, 2048 neurons in the fifth hidden layer, and 1024 neurons in the sixth hidden layer. Architectures using the same number of neurons in all layers were tested, but could not achieve similar or better accuracy than when the number of neurons was varied as in the chosen model.

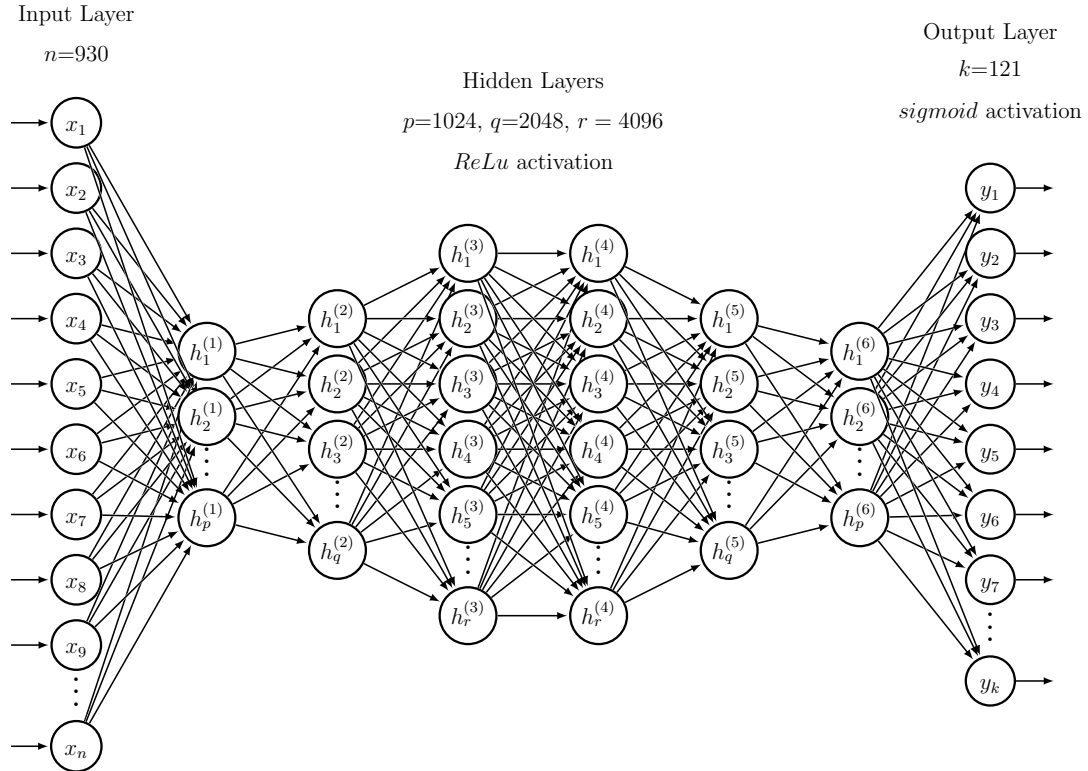


Figure 3.5. MLP architecture.

By increasing the number of neurons initially, we can capture and extract a range of features from the input data, allowing the network to learn complex patterns and representations. By decreasing the number of neurons, the network is allowed to focus on the most relevant features. Batch normalization, a dropout of 0.2, and a batch size of 32 (decreased from a size of 64) is used to further prevent overfitting. This architecture forms the basis of all DMLP models going forward in this thesis.

Fixed Number of Sources

DMLPs were trained for fixed number of sources, which can be used with the assumption of prior knowledge of source number. One to five sources are considered, resulting in five separate models, one for each considered source number. Since the single-source scenario has the least complexity, reliable results are achieved with fewer

training samples. For the single-source model, 100,000 training samples are generated by repeating selected sets of unique DOAs and pairing them with WGN realizations. For two to five source models, 500,000 samples are generated for training in a similar manner, utilizing both uniform and non-uniform spacings for three or more sources. The test set consists of 100,000 samples of unseen data, generated similar to the training sets.

Varying Number of Sources

For DOA estimation of a varying number of sources, we propose three different methods. The first two methods utilize the pre-trained five models for fixed number of sources, whereas the third method trains a single DMLP model to accommodate any number of sources from one to five.

In the first method, the five DMLP models are used for predicting DOAs when the number of sources in the test sample is not known *a priori*. Subspace-based techniques, such as MUSIC, require knowledge of the source number for DOA estimation. The source number can be estimated via information theoretic criteria prior to applying the MUSIC algorithm. We also employ the AIC to estimate the number of sources in the test sample, and then use the estimated source number to choose the most suitable pre-trained DMLP model. In this way, we enable the separately trained models to be used for predictions on a dataset consisting of a varying number of sources. The AIC is defined as

$$AIC(P) = 2P - 2\log L(P), \quad (3.4)$$

where P represents the number of sources, and $\log L(P)$ is the log-likelihood of the model with P sources. The AIC is computed for different presumed values of P and the value of P corresponding to the lowest AIC value is chosen as the best fit for the data. In order to use this method, we must loop through all presumed P values for every test sample, generate the log-likelihood for each case, determine the best fit using the lowest AIC score, then use the DMLP model selected to make the prediction. The efficacy of the AIC method can be measured in terms of the false

recovery rate, which represents the percentage of incorrect model selections across all selections.

The second technique for utilizing the pre-trained models is an ensemble method. The output layer of each pre-trained DMLP model uses a sigmoid activation, which generates a probability of presence of a source at each considered DOA in the $[-60^\circ, 60^\circ]$ angular sector. The probabilities for each of the five pre-trained models are averaged to predict the multilabel classification on a test sample. A threshold on the average probability can be applied and adjusted to provide improved performance. Unlike the first technique, this technique does not assume prior knowledge of the number of sources and leverages only the output probabilities of all five fixed source number models to make a prediction.

Lastly, a single DMLP model trained on data for a varying number of sources is executed. Similar to DMLPs for fixed number of sources, the model is trained on a dataset of 2° to 10° separations with 4° and 8° withheld and validated on a dataset of 4° and 8° separations. Similar to the ensemble method, we do not require prior knowledge of the source number for DOA estimation. For this model, the training dataset comprises 500,000 samples with each source number represented equally.

All three aforementioned methods are tested on a dataset of 100,000 samples containing all separations from 2° to 10° with source number varying from one to five and having uniform and non-uniform source separations. The validation set contains 100,000 samples.

Detection Tolerance

Based on the intended use for underwater surveillance, 1° granularity is not altogether necessary. We consider tolerances of $\pm 1^\circ$ and $\pm 2^\circ$, corresponding to respective spans of 3° and 5° centered at the true DOA. The tolerance is applied by finding the indices of the 121-element multilabel where the one-hot encoded output is a one. If the predicted and true source numbers are equal and the separation between the predicted and true indices is within the specified tolerance, we consider the prediction

to be correct within the applied tolerance. Figure 3.6 provides a simple example with five non-uniformly spaced sources, where the predicted DOAs of the third and fourth sources exhibit slight biases. Under zero tolerance, this example would qualify as an incorrect prediction. However, as the biases fall within $\pm 1^\circ$ of the respective true DOAs, this test case would qualify as a successful prediction under the $\pm 1^\circ$ tolerance.

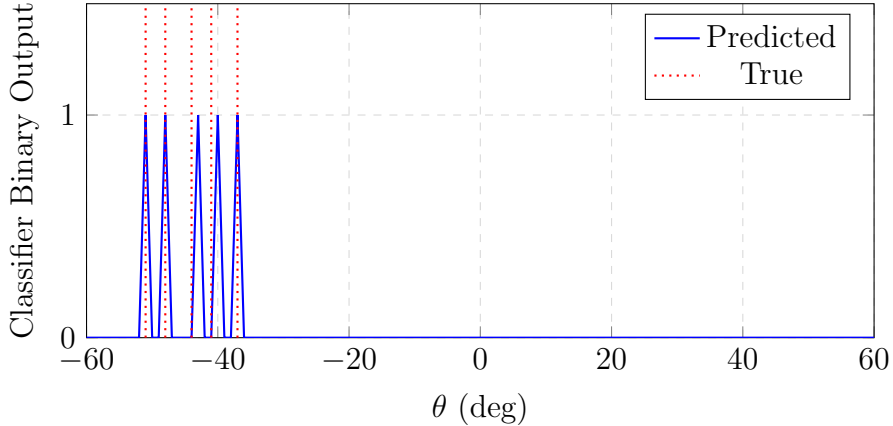


Figure 3.6. Example of prediction within detection tolerance of 1° .

3.4 DOA Estimation Results

Table 3.1 summarizes the accuracy of the models trained on WGN. The first five rows correspond to the fixed source number experiments described in Section 3.3.1. The last three rows correspond to the three methods for varying number of sources, discussed in Section 3.3.1. Specifically, the sixth, seventh, and eighth rows correspond to the single DMLP model, the ensemble method, and the method using AIC for model selection, respectively. We observe a large drop in the validation accuracy between one and five sources, about a 40% decrease. However, results improve considerably when detection tolerance is applied. Validation accuracy for the five source model improves from 59.50% to 80.24% when 2° tolerance is applied. We observe that the DMLPs for fixed source number show high test accuracy ($> 91\%$) even when no detection tolerance is applied. However, as the number of sources

increases, the validation and test accuracies exhibit a decreasing trend, indicating the growing complexity of the DOA estimation problem with more sources. We observe roughly a 5% decrease between one source and five source test accuracies with 0° tolerance. However, overall performance improves and the performance gaps between one source and five sources are reduced when tolerances are applied. The AIC-based model selection method provides the lowest accuracy both with and without detection tolerance. This is attributed to the fact that the false recovery rate of AIC is 42.20%. It is well-known that the AIC tends to under-estimate the number of sources at low SNR [69]. Finally, the ensemble method outperforms the AIC-based model selection, but it underperforms the single DMLP model trained for varying number of sources.

Table 3.1. Results of Baseline Models in WGN.

Sources	Training	Accuracy (%)					
		Validation ($\pm 0^\circ$ tol.)	Validation ($\pm 1^\circ$ tol.)	Validation ($\pm 2^\circ$ tol.)	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
1	99.86	96.50	96.71	96.71	96.74	96.74	96.74
2	97.69	94.05	94.93	94.93	96.86	97.05	97.05
3	99.19	87.17	93.08	93.08	96.22	97.39	97.39
4	99.25	78.67	88.85	88.87	93.95	96.62	96.62
5	99.00	59.50	80.15	80.24	91.18	95.60	95.60
1-5 ^s	95.79	76.52	86.54	86.63	91.76	94.48	94.49
1-5 ^e					71.23	73.74	73.74
1-5 ^a					55.04	55.71	55.71

^s Single model

^e Ensemble

^a Model selection using AIC

Delving further into the AIC-selection method, we observe a tendency towards selecting models for one, two, or three sources. Figure 3.7 shows the distribution of model selections. The testing dataset consists of each source number represented equally with 20,000 samples per source number. Recall in (3.4), AIC is determined

by the number of sources and the log-likelihood. If log-likelihoods of signals models are similar, AIC will bias towards lower source number and underestimate higher-valued source models. Additionally, a higher log-likelihood results in a lower AIC value. With further investigation, we find that the log-likelihood between models is distinguishable, as the log-likelihood between two models is equal in only 32 samples out of 42,202 incorrect model selections. More often, the lower source number models generate higher log-likelihoods, resulting in lower overall AIC values.

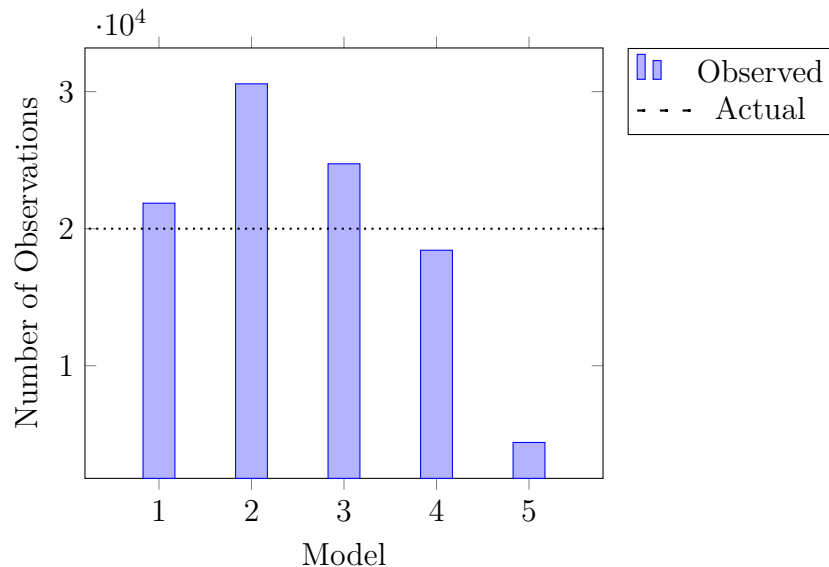


Figure 3.7. Distribution of AIC model selection.

We further investigate the performance of the ensemble model. As an experiment to observe how the ensemble method performs as the complexity of the DOA problem changes, other datasets were generated to test its performance on up to two, three, four, or five sources. In this experiment, the output of the models are weighted equally, but the probability threshold is adjusted to improve and achieve the results listed in Table 3.2. The following threshold values are used: 0.45 for the 1-2 source ensemble, 0.50 for 1-3 sources, 0.40 for 1-4 sources, and 0.40 also for 1-5 sources. From Table 3.2, we see decreasing accuracy as the diversity of sources increases.

Table 3.2. Results of Baseline Ensemble Models on Test Data in WGN.

Sources	Accuracy (%)		
	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
1-2	88.11	88.17	88.17
1-3	87.13	88.59	88.59
1-4	76.51	78.46	78.46
1-5	71.23	73.74	73.74

3.5 Conclusion

Despite array imperfections and partly calibrated distributed array, the results indicate that DMLP models can be successfully trained to detect sources in WGN, both in fixed and varying source number scenarios. For fixed source number DMLPs, the accuracy is $>91\%$ without tolerance, and the numbers improve when tolerance is applied. For varying source models and methods, the single model for varying sources outperforms both the AIC model selection and the ensemble method. The AIC method, in particular, tends to under-estimate the number of sources for the considered low SNR, thereby leading to lowest accuracy values across the board. The performance of the ensemble method was shown to degrade as the complexity of the detection problem increased. Its accuracy is 20% lower compared to the single model for varying source number. Since the ensemble is composed of models trained and optimized on a fixed number of sources, the ensemble method struggled in a varying source number scenario, especially because the models are given equal weight. Out of all the methods used to estimate DOAs for varying source number, the single model performs best and has the additional benefit of needing no prior knowledge of the number of sources.

CHAPTER 4

DOA ESTIMATION IN SPATIALLY-CORRELATED GAUSSIAN NOISE

In Chapter 3, DMLPs were trained for fixed and varying number of sources embedded in WGN. However, underwater acoustic arrays often encounter spatially correlated Gaussian noise. Based on this characterization, we consider AR(1) and AR(2) processes to represent spatially correlated noise. We first use signals corrupted by AR(1) and AR(2) noise to compute the vectorized upper triangular part of the covariance matrices, which are then used as input to the pre-trained WGN models of Chapter 3 to observe the impact of spatially correlated noise on DOA estimation performance. That is, the saved parameters from WGN-trained models are initialized and predictions on unseen correlated noise data are made without any additional training. Next, we employ warm start training and few-shot meta-learning to fine-tune the WGN-trained models to cater to spatially correlated noise.

4.1 WGN-Trained Models

Experimental Setup

Correlation coefficients are assumed to be the same across the subarrays. Test sets for individual models with a fixed number of sources consist of 100,000 samples, while data for varying sources are composed of 20,000 samples for each source number from one to five, totaling 100,000 samples. For AR(1), α_1 -values are initialized to 0.1, 0.5, or 0.9, where 0.1 represents a weaker correlation between hydrophones as compared to 0.9. For AR(2), several sets of values (α_1, α_2) are applied, namely, (0.50, 0.20), (0.50, 0.49), (0.90, 0.09), and (0.85, 0.14), respectively.

Classification Results

Table 4.1 summarizes the performance of the WGN-trained models with AR(1) noise based input. For each value of α_1 , the first five rows correspond to the fixed source number models. The last three rows correspond to the three methods for varying number of sources, namely, the single DMLP model, the ensemble method, and the method using AIC for model selection. We observe that $\alpha_1 = 0.1$ and $\alpha_1 = 0.5$ do not greatly affect DOA estimation accuracy of the WGN-trained models, except for the case of the ensemble model, which exhibits performance degradation across all considered correlation coefficient values. The overall trend is largely in line with baseline model accuracies with WGN noise based input, with decreasing accuracy values as the number of sources increases. On the other hand, there is a 9.75% drop in test accuracy without detection tolerance when the strength of the correlation increases from $\alpha = 0.1$ to $\alpha = 0.9$ for the single model trained on varying number of sources. Similar to the case when testing with WGN based datasets, the AIC model determination method yields lower accuracy than the single model for varying source number. The false recovery values for the AIC corresponding to different values of α_1 are provided in Table 4.2. For $\alpha_1 = 0.9$, the false recovery probability drops a bit further, as the AIC is known to provide incorrect source number estimates in correlated noise [70].

The results for AR(2) noise are presented in Table 4.3. The AR(2) datasets exhibit similar trends as AR(1) noise, with $\alpha_1 = 0.50$ and $\alpha_2 = 0.49$ having the greatest impact on performance with increasing number of sources. The AIC based model selection method, again, suffers due to the negative impact of correlated noise and low SNR conditions on source number estimates. Likewise, the ensemble method provides the worst performance for all considered values of α_1 and α_2 .

Table 4.1. AR(1) Noise & WGN-Trained Models, Varying α -Values.

Corr Coeff (α_1)	Sources	Accuracy (%)		
		Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
(0.1)	1	96.77	96.77	96.77
	2	96.72	96.92	96.72
	3	96.03	97.27	97.27
	4	93.77	96.30	96.30
	5	91.16	95.71	95.71
	1-5 ^s	91.56	94.36	94.37
	1-5 ^e	25.94	25.94	25.94
	1-5 ^a	54.90	55.55	55.55
(0.5)	1	96.82	96.82	96.82
	2	96.92	97.08	97.08
	3	96.20	97.42	97.42
	4	93.53	96.38	96.39
	5	91.14	95.67	95.67
	1-5 ^s	91.50	94.25	94.26
	1-5 ^e	25.56	25.56	25.56
	1-5 ^a	55.11	55.76	55.77
(0.9)	1	96.36	96.36	96.36
	2	96.39	96.48	96.48
	3	93.19	94.92	94.92
	4	88.55	92.01	92.01
	5	81.81	87.38	87.39
	1-5 ^s	77.95	80.98	81.00
	1-5 ^e	24.94	24.94	24.94
	1-5 ^a	55.91	56.69	56.69

^s Single model

^e Ensemble

^a Model selection using AIC

Table 4.2. AIC False Recovery for AR(1) Noise.

Corr Coeff (α_1)	False Recovery (%)
(0.1)	42.39
(0.5)	42.20
(0.9)	40.45

Remarks

The results of the AR(1) and AR(2) noise datasets, when input to pre-trained WGN models, point to an underlying need for model adaptability to different noise scenarios. DMLP models see the most degradation in DOA estimation accuracy in AR(1) noise when $\alpha_1 = 0.9$ and in AR(2) noise when $\alpha_1 = 0.50$ and $\alpha_2 = 0.49$. The single model for varying number of sources, in particular, struggles in AR(1) noise when $\alpha_1 = 0.9$, but is less affected by AR(2) noise, though its performance is degraded nonetheless. When considering the sparse geometry of the subarrays, there are many $l\lambda/2$ locations in the subarray that do not contain a hydrophone. Referring to Figure 3.1, subarray S2 is the only subarray with sensors positions in adjacent locations separated by $\lambda/2$, while the closest hydrophones in subarray S1 are separated by a distance of λ . The sparse geometry appears to benefit DOA estimation in AR(2) noise.

In order to address model adaptability, we next consider warm start learning and few-shot meta-learning. For the fixed source number models, we only investigate adaptability for three, four, and five sources cases, as the performance of the WGN trained models was impacted more so as the number of sources increased.

4.2 Warm Start Learning

The DMLP models are updated and trained with datasets containing AR(1) or AR(2) noise via warm start learning. The models are initialized with pre-trained

parameters and fine-tuned using training samples with correlation coefficients that have the strongest impact on the distributed array. A varying number of samples are used to observe the effect of training data size on model performance.

Experimental Setup

As DOA estimation performance of WGN trained models was impacted by datasets with strong noise correlations, warm start learning is considered to fulfill the demand for model adaptability. We use training, validation, and test datasets with AR(1) noise, where $\alpha = 0.9$, and AR(2) noise, where $\alpha_1 = 0.50$ and $\alpha_2 = 0.49$. Early stopping is implemented, where the model is stopped if the validation loss does not improve after 10 epochs and the best weights and biases prior to stopping are saved. The epoch number listed in Tables 4.4 and 4.5 corresponds to the last epoch of loss improvement.

For the single model under varying number of sources, each source number is represented equally in the training samples. For all models, input training samples of 5, 10, 50, 100, 500, 1000, 5000, and 10,000 are considered. The validation set consists of 20,000 samples, and the test set contains 100,000 samples. No spacings are withheld in the datasets in order to optimize DMLP models for DOA estimations in AR(1) and AR(2) noise.

Classification Results

Classification accuracies for DOA estimation using warm start learning are collected in Tables 4.4 and 4.5 for AR(1) and AR(2) noise, respectively. The AR(1) noise results show that few samples are needed to fine-tune fixed source number models. In particular, five-source model exhibits DOA estimation improvement after only five samples. Recall that when the WGN-trained model was asked to make predictions on $\alpha = 0.9$ AR(1) data without retraining, the accuracy was 81.81% without tolerance. When the model is trained with five samples of data, an improvement of 4.49% is recorded. Likewise, for the varying source model, there is a 4.92% improvement in

DOA estimation accuracy after five samples with no tolerance. As expected, DOA estimation accuracy increases as training samples increase.

Accuracy versus number of training samples for the different models is plotted for AR(1) noise in Figure 4.2. We observe a large jump in accuracy for the single varying-source model after 10 samples. Similarly, fixed source models again see improvements with as few as five samples. The single model for varying sources displays larger improvements. Accuracy versus samples is plotted for AR(2) in Figure 4.2, showing the largest improvement for the varying source model between 5 and 10 samples. We also observe that the number of epochs needed to achieve these improvements is modest, often < 50 epochs, with the exception of the single model for varying number of sources.

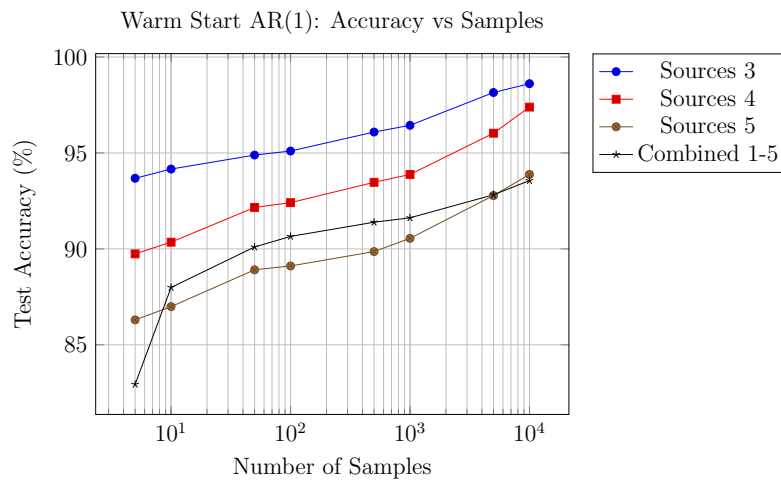


Figure 4.1. Test accuracy (0° tolerance) for warm start learning & AR(1) data.

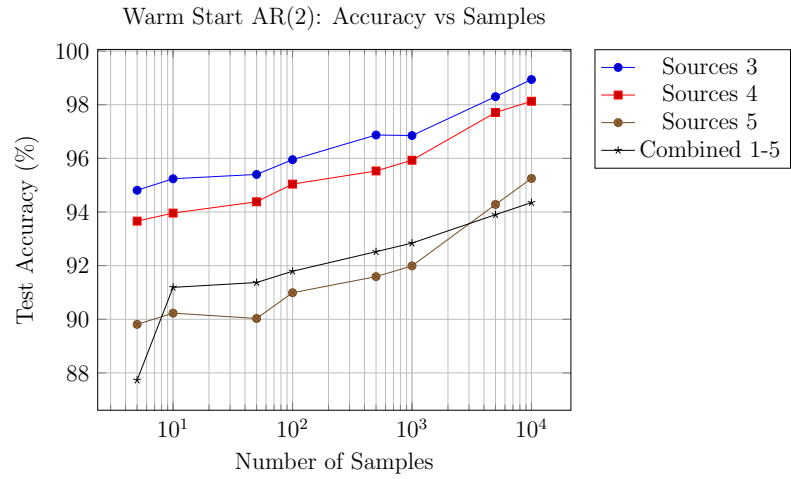


Figure 4.2. Test accuracy (0° tolerance) for warm start learning & AR(2) data.

Table 4.3. AR(2) Noise & WGN-Trained Models, Correlated α -Values.

Corr Coeff (α_1, α_2)	Sources	Accuracy (%)		
		Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
(0.50, 0.20)	1	96.66	96.66	96.66
	2	96.84	97.03	97.03
	3	95.98	97.19	97.19
	4	94.04	96.55	96.56
	5	90.71	95.39	95.40
	1-5 ^s	91.29	94.03	94.04
	1-5 ^e	25.35	25.35	25.35
	1-5 ^a	55.14	55.78	55.78
(0.50, 0.49)	1	96.82	96.82	96.82
	2	96.25	96.50	96.50
	3	94.69	96.38	96.38
	4	93.55	96.13	96.13
	5	89.64	94.73	94.73
	1-5 ^s	86.15	89.20	89.21
	1-5 ^e	24.96	24.96	24.96
	1-5 ^a	55.76	56.47	56.47
(0.90, 0.09)	1	97.12	97.12	97.12
	2	96.93	97.07	97.07
	3	96.24	97.38	97.38
	4	95.06	97.19	97.20
	5	91.92	96.01	96.01
	1-5 ^s	92.01	94.57	94.58
	1-5 ^e	25.95	25.95	25.95
	1-5 ^a	54.85	55.41	55.41
(0.85, 0.14)	1	97.17	97.17	97.17
	2	96.84	97.01	97.01
	3	96.19	97.33	97.33
	4	94.95	97.11	97.11
	5	91.73	95.86	95.86
	1-5 ^s	91.91	94.55	94.56
	1-5 ^e	25.85	25.85	25.85
	1-5 ^a	54.63	55.25	55.25

^s Single model^e Ensemble^a Model selection using AIC

Remarks

Under both AR(1) and AR(2) noise, accuracy improves with only five samples and continues to improve with more test samples for all models, indicating a robust pre-trained model. The results suggest that the DMLP models are well-suited for warm start updates. Based on improvements in DOA accuracy, warm start learning offers adaptability to new noise environments and retains the ability to make accurate DOA estimations with array imperfections in the presence of spatially correlated noise. In the next section, few-shot meta-learning is evaluated as an alternative method for model update.

4.3 Few-Shot Meta-Learning

Few-shot meta-learning is explored as another way to update models on various types of noise the subarrays might encounter. The pre-trained WGN model is updated using a meta-learning paradigm, specifically, MAML, where it is exposed to datasets with AR(1) and AR(2) noise. We investigate whether this technique will allow pre-trained models to adapt quickly and generalize to different noise patterns.

Experimental Setup

For few-shot MAML, the pre-trained WGN model weights are initialized for the meta-learner, transferring knowledge from pre-trained models at the start in an effort to support fast adaptation. This series of training examines several K -shot N -way combinations. The considered K values are $K = 1$ and $K = 5$, which represent the number of samples per class, while the considered N values are $N = 5$, $N = 10$, and $N = 20$, each representing the number of classes in a task. Resampling is performed for each task to allow the model to learn on a diverse set of tasks.

Table 4.4. Warm Start Learning: AR(1) Noise $\alpha_1 = 0.9$.

Sources	Samples	Epochs	Accuracy (%)				
			Train	Val	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
3	5	7	100.00	94.31	93.68	95.38	95.38
	10	17	100.00	94.89	94.16	95.86	95.86
	50	26	100.00	95.17	94.89	96.45	96.45
	100	15	99.00	95.37	95.10	96.67	96.67
	500	9	100.00	96.19	96.09	97.21	97.21
	1000	6	99.50	96.48	96.44	97.27	97.27
	5000	12	99.86	98.08	98.15	98.50	98.50
	10000	8	99.64	98.61	98.61	98.78	98.78
4	5	14	100.00	89.48	89.74	93.45	93.45
	10	12	100.00	90.37	90.35	94.06	94.08
	50	26	100.00	92.29	92.16	95.42	95.42
	100	23	98.00	92.96	92.41	95.52	95.53
	500	15	99.80	93.83	93.47	96.25	96.25
	1000	7	99.40	93.99	93.88	96.45	96.45
	5000	23	99.88	96.47	96.03	97.41	97.41
	10000	22	99.89	97.20	97.39	98.33	98.33
5	5	21	100.00	86.02	86.30	92.21	92.22
	10	39	100.00	87.18	86.99	93.19	93.22
	50	51	100.00	89.18	88.91	94.62	94.62
	100	21	100.00	89.33	89.11	94.16	94.16
	500	15	100.00	90.31	89.86	95.17	95.17
	1000	16	90.80	89.30	90.55	95.23	95.23
	5000	17	99.98	92.70	92.78	95.75	95.75
	10000	9	99.71	93.78	93.88	96.60	96.60
1-5 ^s	5	17	100.00	82.87	82.94	86.78	86.79
	10	67	100.00	88.04	87.99	91.62	91.64
	50	64	100.00	90.16	90.10	93.32	93.33
	100	29	100.00	90.73	90.65	93.71	93.72
	500	8	99.80	91.43	91.39	94.27	94.27
	1000	5	98.40	91.79	91.62	94.33	94.34
	5000	5	98.24	92.96	92.82	95.00	95.01
	10000	5	98.32	93.56	93.56	95.44	95.44

^s Single model

Table 4.5. Warm Start Learning: AR(2) Noise $\alpha_1 = 0.50$, $\alpha_2 = 0.49$.

Sources	Samples	Epochs	Accuracy (%)				
			Train	Val	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
3	5	8	100.00	94.71	94.81	96.46	96.46
	10	28	100.00	95.46	95.24	96.78	96.78
	50	34	100.00	95.37	95.40	96.91	96.91
	100	13	100.00	96.10	95.95	97.22	97.22
	500	9	100.00	96.66	96.87	97.73	97.73
	1000	7	99.70	96.90	96.85	97.71	97.71
	5000	7	99.84	98.28	98.30	98.67	98.67
	10000	12	99.68	99.00	98.94	99.09	99.09
4	5	5	100.00	93.64	93.66	96.23	96.23
	10	21	100.00	93.73	93.96	96.48	96.48
	50	33	100.00	94.36	94.38	96.68	96.68
	100	18	100.00	94.79	95.04	97.06	97.06
	500	7	100.00	95.56	95.53	97.39	97.39
	1000	16	100.00	95.71	95.93	97.59	97.59
	5000	23	100.00	97.78	97.71	98.58	98.58
	10000	13	99.90	98.12	98.13	98.76	98.76
5	5	4	100.00	90.41	89.81	94.83	94.83
	10	9	100.00	90.33	90.23	95.07	95.07
	50	6	98.00	90.36	90.03	94.95	94.95
	100	17	100.00	91.27	90.99	95.44	95.44
	500	7	99.00	91.69	91.59	95.78	95.79
	1000	8	99.90	91.99	91.99	95.99	95.99
	5000	7	99.92	94.59	94.28	96.83	96.83
	10000	11	99.87	95.34	95.26	97.29	97.29
1-5 ^s	5	31	100.00	87.65	87.73	91.01	91.03
	10	29	100.00	90.93	91.19	93.97	93.99
	50	39	100.00	91.28	91.37	94.14	94.16
	100	23	98.00	91.56	91.79	94.39	94.40
	500	7	99.20	92.35	92.52	94.89	94.89
	1000	7	99.20	92.74	92.84	95.11	95.12
	5000	8	98.90	94.01	93.90	95.66	95.66
	10000	7	98.49	94.86	94.35	95.82	95.82

^s Single model

To create a task for multilabel classification, we adopt from the method described in [71]. We consider each individual label in the multilabel classification as a class. For our application, an individual label corresponds to a DOA. In task sampling, N DOAs are randomly selected and K samples that contain the DOAs are gathered to form the support set. As a hypothetical example, for 1-shot, 3-way few-shot learning, we randomly select -24° , 1° , and 9° for DOA and select one covariance sample containing a source at -24° , one sample corresponding to a source at 1° , and one sample with a source at 9° for the support set. If our query set consists of 10 samples, then we randomly select 10 samples that contain a source at either -24° , 1° , or 9° .

The MAML algorithm consists of an inner loop, which produces adapted parameters using the support set for a given task, and an outer loop, which evaluates the adapted parameters on the query set for each task. The outer loop is iterated over the number of tasks in the query set. However, the inner loop optimizes adapted parameters using the support set, S_i , and can be iterated multiple times. For this implementation, the number of steps for the inner loop is set to 20. The number of samples in the query set, Q_i , is set to 15, and the number of tasks \mathcal{T}_i is set to 10. The total number of samples needed to train the model can be calculated as

$$\text{Total samples} = \bar{T} \times (N \times K + N \times Q), \quad (4.1)$$

where \bar{T} is the number of tasks and Q denotes the number of samples in the query set. While the training samples vary in our investigations, the test dataset has a fixed value of 100,000 samples.

Classification Results

Tables 4.6 and Table 4.7 summarize the findings of few-shot meta-learning for AR(1) and AR(2) noise cases, respectively, for different combinations of K and N . For signals embedded in AR(1) and AR(2) noise, few-shot learning can be used to update models with as little as one example for five classes, i.e., 1-shot 5-way. In this case, 800 training samples were used. Without tolerance, the five-source model in AR(1) noise

was able to achieve 90.65% accuracy and the varying source number model was able to achieve 91.22%. To compare few-shot MAML to warm start learning, we consider 5-shot 5-way training, which uses 1,000 training samples. Without tolerance, warm start learning generates 91.62% accuracy, while few-shot learning is able to achieve 91.99% accuracy. The results are comparable.

For the case of AR(2) noise, the accuracy for the five-source model when few-shot MAML is implemented produces similar results to warm start learning as well; 5-shot 5-way accuracy without tolerance is 92.52% and warm start learning is 91.99%. For the varying source model, accuracy without tolerance is 92.79% for few-shot MAML, compared to 92.84% for warm start learning.

Table 4.6. Few Shot Learning: AR(1) Noise $\alpha = 0.9$.

Sources	K -shot	N -way	Accuracy (%)				
			Train Avg Loss ($\times 10^{-3}$)	Train Avg	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
3	1	5	2.79	91.20	96.02	97.26	97.26
	5	5	1.98	93.07	97.40	98.25	98.25
	1	10	2.28	93.73	96.51	97.49	97.49
	5	10	1.99	93.60	97.84	98.47	98.47
	1	20	2.03	93.97	97.51	98.13	98.13
	5	20	1.62	94.57	98.19	98.68	98.68
4	1	5	8.15	78.27	93.54	96.32	96.33
	5	5	8.20	78.40	94.66	97.32	97.32
	1	10	7.16	80.20	94.37	96.88	96.88
	5	10	6.68	82.00	94.84	97.21	97.22
	1	20	6.99	80.63	94.43	96.98	96.98
	5	20	6.58	81.30	95.07	97.09	97.09
5	1	5	8.61	80.93	90.65	95.52	95.53
	5	5	8.50	82.40	91.17	95.72	95.72
	1	10	8.73	80.07	90.73	95.49	95.49
	5	10	7.02	83.67	91.75	95.99	95.99
	1	20	8.18	81.47	91.75	95.90	95.90
	5	20	7.28	82.50	92.71	96.53	96.53
1-5 ^s	1	5	6.07	84.40	91.22	94.00	94.00
	5	5	5.07	85.87	91.99	94.49	94.49
	1	10	5.39	86.80	91.75	94.34	94.34
	5	10	4.75	86.47	92.51	94.90	94.91
	1	20	5.67	86.27	92.04	94.57	94.57
	5	20	4.88	87.53	92.72	95.03	95.05

^s Single model

Table 4.7. Few Shot Learning: AR(2) Noise $\alpha_1 = 0.50$, $\alpha_2 = 0.49$.

Sources	K -shot	N -way	Accuracy (%)				
			Train Avg Loss ($\times 10^{-3}$)	Train Avg	Test ($\pm 0^\circ$ tol.)	Test ($\pm 1^\circ$ tol.)	Test ($\pm 2^\circ$ tol.)
3	1	5	2.88	91.47	96.35	97.40	97.40
	5	5	2.14	93.47	97.60	98.35	98.35
	1	10	2.24	93.07	97.12	97.94	97.94
	5	10	1.18	96.33	98.08	98.62	98.62
	1	20	1.73	95.10	97.60	98.32	98.32
	5	20	1.15	96.63	98.42	98.85	98.85
4	1	5	4.53	85.87	94.95	96.99	96.99
	5	5	4.07	87.47	96.06	97.73	97.73
	1	10	3.76	89.33	95.61	97.48	97.49
	5	10	2.99	91.80	96.36	97.90	97.90
	1	20	3.21	90.13	95.97	97.68	97.68
	5	20	2.59	92.47	96.71	98.10	98.10
5	1	5	8.60	80.53	91.78	95.86	95.86
	5	5	7.65	82.53	92.52	96.25	96.25
	1	10	7.54	80.80	92.01	96.05	96.06
	5	10	6.10	84.67	93.01	96.56	96.57
	1	20	5.71	86.20	92.34	96.18	96.18
	5	20	4.58	88.40	93.48	96.75	96.75
1-5 ^s	1	5	4.58	88.13	92.14	94.58	94.58
	5	5	5.37	85.87	92.79	94.97	94.97
	1	10	4.68	87.27	92.55	94.83	94.84
	5	10	4.60	86.67	93.27	95.34	95.34
	1	20	4.18	88.50	92.85	95.03	95.04
	5	20	4.22	88.77	93.55	95.56	95.57

^s Single model

Conclusion

Few-shot MAML learning is the obvious choice when data is extremely limited, as it is specifically designed to enable rapid adaptation to new tasks with minimal data. The results indicate similar performance to warm start training when the same

number of input samples are considered, suggesting that both methods are capable of achieving robust performance under the considered conditions. While AR(1) and AR(2) spatially-correlated noise is employed in our examinations, the spatial noise remains Gaussian in nature, allowing warm start learning models, pre-trained on WGN, to generate accurate DOA estimations with a small number of samples.

The unique strength of few-shot MAML lies in its ability to adapt to new and diverse conditions. This adaptability makes it particularly well-suited for scenarios involving domain disparities that deviate from the training distribution. In such cases, few-shot MAML’s framework can provide an advantage by generalizing across tasks and building a model that is adaptable to unseen variations. On the other hand, warm start training leverages pre-trained knowledge effectively, excelling in scenarios where the testing conditions closely align with noise that is Gaussian distributed.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, we considered the problem of source DOA estimation in underwater environments using a distributed sparse array with imperfections. We explored various ML models and techniques for handling spatially-correlated noise and array imperfections under the noncoherent processing paradigm. We trained models for a fixed number of sources, which can be used when prior knowledge of the number of sources is available. The resulting models were used for DOA estimation for a varying number of sources by exploiting the AIC model selection method and an ensemble method. The performance of these two methods is compared against that of a single model trained on varying number of sources. The single model for varying source number outperformed the AIC and ensemble methods by a large margin. The results also demonstrated the efficacy of warm start learning and few-shot MAML in updating pre-trained models to new data when the data is limited. Since training is carried out entirely offline, we have demonstrated how DMLP models can offer a robust alternative to subspace-based methods. In case of the latter, real-time DOA estimation can be computationally expensive and the performance can degrade in spatially-correlated noise and under array imperfections.

The scope of this study was limited to spatially-correlated Gaussian noise. Incorporating other statistical models for non-Gaussian noise, applicable in shallow waters, could provide a more comprehensive understanding of noise impacts and exercise the flexibility of our methods. Few-shot MAML, in particular, is often used when there are domain disparities with few samples. Testing with other noise characteristics naturally follows as a further course of action. Fine-tuning the models to account for other ocean ambient noise characteristics is essential for enhancing their performance and viability in real-world applications. In addition, optimizing deep learning models by reducing the number of training samples and using smaller model architectures may yield more efficient and faster models without compromising accuracy. Future

work should include adjusting the DMLP architecture for fewer sources to be more compact, as the stored parameters for a pre-trained DMLP of this size can be quite large. Smaller architectures are especially necessary for consideration in embedded signal processing systems. Future work can focus on optimizations to strike a balance between computational efficiency and performance.

REFERENCES

- [1] H. L. Van Trees, *Optimum Array Processing*. Wiley, 2004.
- [2] X. Li, H. Lu, and H. Chen, *Direction of Arrival Estimation Methods and Applications*. Scholars' Press, 2024.
- [3] A. Moffet, "Minimum-redundancy linear arrays," *IEEE Transactions on Antennas and Propagation*, vol. 16, no. 2, pp. 172–175, 1968.
- [4] G. Bloom and S. Golomb, "Applications of numbered undirected graphs," *Proceedings of the IEEE*, vol. 65, no. 4, pp. 562–570, 1977.
- [5] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010.
- [6] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, Feb 2011.
- [7] S. Qin, Y. D. Zhang, and M. G. Amin, "Generalized coprime array configurations for direction-of-arrival estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, 2015.
- [8] K. H. Kheirollahpour, A. Mahmoudi, and B. Dumitrescu, "DOA estimation in the autocorrelation domain for coprime array," *Digital Signal Processing*, vol. 110, pp. 1–9, 2021, Art. ID 102940.
- [9] D. G. Chachlakis, T. Zhou, F. Ahmad, and P. P. Markopoulos, "Minimum mean-squared-error autocorrelation processing in coprime arrays," *Digital Signal Processing*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1051200421000737>
- [10] P. Pal and P. P. Vaidyanathan, "Pushing the limits of sparse support recovery using correlation information," *IEEE Transactions on Signal Processing*, vol. 63, no. 3, pp. 711–726, 2015.
- [11] Z. Zheng, Y. Huang, W.-Q. Wang, and H. C. So, "Direction-of-arrival estimation of coherent signals via coprime array interpolation," *IEEE Signal Processing Letters*, vol. 27, pp. 585–589, 2020.
- [12] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang, "Direction-of-arrival estimation for coprime array via virtual array interpolation," *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 5956–5971, 2018.
- [13] C. Zhou, Y. Gu, Y. D. Zhang, Z. Shi, T. Jin, and X. Wu, "Compressive sensing-based coprime array direction-of-arrival estimation," *IET Communications*, vol. 11, no. 11, pp. 1719–1724, 2017.
- [14] Y. Liu and J. R. Buck, "Gaussian source detection and spatial spectral estimation using a coprime sensor array with the min processor," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 186–199, 2018.

- [15] C. Liu and P. P. Vaidyanathan, "Remarks on the spatial smoothing step in coarray MUSIC," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1438–1442, 2015.
- [16] C. Kopp, "Identification underwater with towed array sonar," *Defence Today*, pp. 32–33, December 2009.
- [17] S. G. Lemon, "Towed-array history, 1917–2003," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 365–373, 2004.
- [18] R. Holler, A. Horbach, and J. McEachern, *The Ears of Air ASW: A History of U.S. Navy Sonobuoys*. Navmar Applied Sciences Corporation, 2008.
- [19] R. Holler, "The evolution of the sonobuoy from World War II to the Cold War," Navmar Applied Sciences Corporation, DTIC Technical Report ADC08281, 2012. [Online]. Available: <https://apps.dtic.mil/sti/citations/tr/ADA597432>
- [20] A. A. Sheikh, E. Felemban, M. Felemban, and S. B. Qaisar, "Challenges and opportunities for underwater sensor networks," in *12th International Conference on Innovations in Information Technology*, 2016, pp. 1–6.
- [21] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 154–165. [Online]. Available: <https://doi.org/10.1145/1098918.1098936>
- [22] Y. Li, B. Ouyang, T. Zhou, J. Thomas, S. Bavar, E. Weber, L. Michieletto, T.-C. Su, and F. Ahmad, "Laboratory and field experimental study of underwater inflatable co-prime sonar array (UICSA)," *Journal of Civil Engineering and Construction*, vol. 11, no. 1, pp. 41–55, 2022.
- [23] B. Ouyang, Y. Li, T. Zhou, T.-C. Su, F. Dalglish, A. Dalglish, and F. Ahmad, "Compressing two ways: The initial study of an underwater inflatable co-prime sonar array (UICSA)," in *Proceedings of the SPIE Compressive Sensing VII: From Diverse Modalities to Big Data Analytics*, vol. 10658, 04 2018, Art. ID 106580H.
- [24] Y. Li, J. Thomas, B. Ouyang, T. Su, and F. Ahmad, "Numerical study of underwater inflatable co-prime sonar array (UICSA)," in *Proceedings of the ASME 39th International Conference on Ocean, Offshore and Arctic Engineering*, vol. 4: Pipelines, Risers, and Subsea Systems, 08 2020, pp. 1–6, Art. ID OMAE2020-18393.
- [25] S. R. Pavel, M. W. T. S. Chowdhury, Y. D. Zhang, D. Shen, and G. Chen, "Machine learning-based direction-of-arrival estimation exploiting distributed sparse arrays," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 241–245.
- [26] W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Non-coherent direction-of-arrival estimation using partly calibrated arrays," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5776–5788, 2018.
- [27] S. R. Pavel and Y. D. Zhang, "Neural network-based compression framework for DOA estimation exploiting distributed array," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 4943–4947.

- [28] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing*. Prentice Hall Signal Processing Series, 1993.
- [29] C. S. Horton, *Signal Processing of Underwater Acoustic Waves*. The University of Texas at Austin, 1969.
- [30] R. Urick, *Principles of Underwater Sound*. McGraw-Hill, 1983.
- [31] G. Wenz, "Acoustic ambient noise in the ocean: Spectra and sources," *The Journal of the Acoustical Society of America*, vol. 34, no. 12, pp. 1936–1956, 1962.
- [32] P. C. Etter, *Underwater Acoustic Modeling and Simulation*. Spon Press, 2003.
- [33] M. Bouvet and S. Schwartz, "Detection in underwater noises modeled as a gaussian-gaussian mixture," in *ICASSP '86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, 1986, pp. 2795–2798.
- [34] G. Song, X. Guo, and L. Ma, "The Alpha stable distribution in ocean ambient noise modelling," *MATEC Web of Conferences*, vol. 283, Jan 2019.
- [35] NOAA, "Passive acoustic data archive," 2025, accessed: 2025-03-17. [Online]. Available: <https://www.ncei.noaa.gov/products/passive-acoustic-data>
- [36] ONC, "Hydrophone arrays - platforms," 2025, accessed: 2025-03-17. [Online]. Available: <https://wiki.oceannetworks.ca/display/PLATFORM/Hydrophone+Arrays>
- [37] C. D. S. Tollefsen and S. Pecknold, "A simple yet practical noise model," Defence Research and Development Canada - Atlantic Research Center, Tech. Rep., 2022.
- [38] X.-Y. Guo, F. Li, G.-P. Tie, and L. Ma, "Overview of ocean ambient noise and application prospects," *Physics*, vol. 43, no. 11, pp. 723–731, 2014. [Online]. Available: <https://wuli.iphy.ac.cn/en/article/doi/10.7693/wl20141103>
- [39] H. Amiri, H. Amindavar, and M. Kamarei, "Underwater noise modeling and direction-finding based on conditional heteroscedastic time series," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4, 2006, pp. IV–IV.
- [40] W. Kuperman and F. Ingenito, "Spatial correlation of surface generated noise in a stratified ocean," *J. Acoustical Soc. Amer.*, vol. 67, no. 6, pp. 1988–1996, 1980.
- [41] H. Cox, "Spatial correlation in arbitrary noise fields with application to ambient sea noise," *J. Acoustical Soc. Amer.*, vol. 54, no. 5, pp. 1289–1301, 1973.
- [42] K. Cipolla and W. L. Keith, "Measurements of the wall pressure spectra on a full-scale experimental towed array," *Ocean Eng.*, vol. 35, no. 10, pp. 1052–1059, 2008.
- [43] R. Bautista and J. R. Buck, "Detecting gaussian signals using coprime sensor arrays in spatially correlated gaussian noise," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1296–1306, 2019.
- [44] F. Ahmad, P. Pal, and T. Zhou, "Reliable DOA estimation in spatially correlated noise with nonuniform arrays," in *55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 1646–1650.

- [45] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [46] A. Paulraj and T. Kailath, "Eigenstructure methods for direction of arrival estimation in the presence of unknown noise fields," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 13–20, 1986.
- [47] A. Ferreol, P. Larzabal, and M. Viberg, "On the asymptotic performance analysis of subspace DOA estimation in the presence of modeling errors: Case of MUSIC," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 907–920, 2006.
- [48] S. Vorobyov, A. Gershman, and K. Wong, "Maximum likelihood direction-of-arrival estimation in unknown noise fields using sparse sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 34–43, 2005.
- [49] T. Li and A. Nehorai, "Maximum likelihood direction finding in spatially colored noise fields using sparse sensor arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1048–1062, 2011.
- [50] M. Malek-Mohammadi, M. Jansson, A. Owrang, A. Koochakzadeh, and M. Babaie-Zadeh, "DOA estimation in partially correlated noise using low-rank/sparse matrix decomposition," in *Proceedings of the IEEE Workshop Sensor Array and Multichannel Signal Processing*, 2014.
- [51] A. Koochakzadeh and P. Pal, "Sparse source localization using perturbed arrays via bi-affine modeling," *Digital Signal Processing*, vol. 61, pp. 15–25, 2017.
- [52] Z.-M. Liu and Y.-Y. Zhou, "A unified framework and sparse bayesian perspective for direction-of-arrival estimation in the presence of array imperfections," *IEEE Transactions on Signal Processing*, vol. 61, no. 15, pp. 3786–3798, 2013.
- [53] F. Ahmad, "Robust co-prime sensing with underwater inflatable passive sonar arrays," Office of Naval Research, Tech. Rep., 2022.
- [54] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [55] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, 1983, pp. 336–339.
- [56] M. Wax and T. Kailath, "Determining the number of signals by information theoretic criteria," in *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 9, 1984, pp. 232–235.
- [57] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2117–2130, 2012.
- [58] Q. Liu, Z. Lai, Z. Zhou, F. Kuang, and Z. Jin, "A truncated nuclear norm regularization method based on weighted residual error for matrix completion," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 316–330, 2015.
- [59] A. Nosedal. (2019) The autocorrelation function and ar(1), ar(2) models. Lecture slides. [Online]. Available: <https://mcs.utm.utoronto.ca/~nosedal/sta457/ar1-and-ar2.pdf>

- [60] A. Lindholm, N. Wahlström, F. Lindsten, and T. Schön, *Machine Learning: A First Course for Engineers and Scientists*. Cambridge University Press, 2022.
- [61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [62] J. T. Ash and R. P. Adams, “On warm-starting neural network training,” 2020. [Online]. Available: <https://arxiv.org/abs/1910.08475>
- [63] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” 2017. [Online]. Available: <https://arxiv.org/abs/1606.04080>
- [64] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135. [Online]. Available: <https://proceedings.mlr.press/v70/finn17a.html>
- [65] M. Cheng, H. Wang, and Y. Long, “Meta-learning-based incremental few-shot object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2158–2169, 2022.
- [66] H. Gharoun, F. Momenifar, F. Chen, and A. H. Gandomi, “Meta-learning approaches for few-shot learning: A survey of recent advances,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.07502>
- [67] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” 2020. [Online]. Available: <https://arxiv.org/abs/1904.05046>
- [68] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [69] Y. Li, N. Wang, and R. J. Carroll, “Selecting the number of principal components in functional data,” *J. Am. Stat. Assoc.*, vol. 103, pp. 1–21, 2013.
- [70] A. P. Liavas and P. A. Regalia, “On the behavior of information theoretic criteria for model order selection,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1689–1695, 2001.
- [71] C. Simon, P. Koniusz, and M. Harandi, “Meta-Learning for Multi-Label Few-Shot Classification,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2022, pp. 346–355. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1109/WACV51458.2022.00042>