Information-based Sampling for Spatiotemporal Field Estimation and Reconstruction in Environmental Monitoring

by

Jiahong Chen

B.Eng., Xiamen University, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Mechanical Engineering)

The University of British Columbia
(Vancouver)

March 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Information-based Sampling for Spatiotemporal Field Estimation and Reconstruction in Environmental Monitoring

submitted by **Jiahong Chen** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Mechanical Engineering**.

Examining Committee:

Prof. Clarence W. de Silva, Mechanical Engineering
Supervisor

Prof. Mu Chiao, Mechanical Engineering
Supervisory Committe Member

Prof. José Martí, Electrical and Computer Engineering
University Examiner

Prof. Ryozo Nagamune, Mechanical Engineering
University Examiner

Additional Supervisory Committee Members:

Prof. Hsi-Yung (Steve) Feng, Mechanical Engineering
Supervisory Committee Member

Prof. Dana Grecov, Mechanical Engineering
Supervisory Committee Member

Abstract

This dissertation addresses the near-optimal deployment problem of robot-sensory nodes in a spatiotemporal field. With limited resources, monitoring of a complex environment may face serious challenges in providing sufficient information for spatiotemporal signal estimation and reconstruction. It is therefore essential to retrieve most useful information from sampling locations while using a small number of sensor nodes. In this dissertation, three aspects are investigated to overcome the shortcomings of the existing information-based sampling methods.

First, a sensor node deployment method is designed to find the minimum number of sensor deployment locations while achieving near-optimal field estimation error. To this end, a sampling-based field exploration method is used to find near-optimal sampling locations over an infinite horizon. Moreover, spatiotemporal correlations of the sampling data are studied to find redundant signals. The corresponding sampling locations of the redundant signals are eliminated concerning the network connectivity.

Second, a deep reinforcement learning approach is proposed to accelerate the field exploration. Typically, field exploration methods are heavily dependent on random sampling, which has low efficiency. To avoid unnecessary or redundant sampling locations, observations from the sampling locations are utilized. Then a model-based information gain determination of the sampling locations is developed to evaluate the effectiveness of the approach. The proposed method can determine the informativeness of the spatiotemporal field by learning the information gain from the sampled area. The mobile sensory agents are then

encouraged to take more samples in the area of higher information gain. Consequently, the spatiotemporal field can be efficiently explored. Moreover, the selected sampling locations can near-optimally reconstruct the spatiotemporal field using statistical methods.

Third, a deep learning framework is designed to provide accurate reconstruction and prediction of the spatiotemporal field, using a limited number of observations. Nonlinear mapping from limited observations to the entire spatiotemporal field is needed in a sufficiently large spatiotemporal field. Hence, a deep learning method is proposed to extract sparse representations of the field and their nonlinear mappings. It is also proven that the proposed framework obeys Lipschitz continuity and that the observations collected by sparse representations are sufficient for spatiotemporal field reconstruction.

Lay Summary

The environmental pollution is a critical issue in todays world, and the ecological changes and pollution may happen anywhere. For example, aquatic pollutants may come from household waste, industrial effluent, and environmental contamination. Large scale monitoring systems can be established to determine the polluted areas and their dynamics. Then, corrective actions can be taken using that information. This dissertation seeks to develop information-based sampling approaches that can retrieve as much information as possible from a spatiotemporal (space-time) field. Redundant sampling locations are detected and eliminated to reduce the infrastructural cost. In this manner, the efficiency of the field exploration is improved while reducing the computational cost, by learning the sampling data. The collected data can be used to estimate the spatiotemporal field. Besides, the field can be reconstructed from the sampling data, and the environmental pollution and changes can be detected or predicted.

Preface

All the work presented in this dissertation was conducted by Jiahong Chen in the Industrial Automation Laboratory (IAL) at the University of British Columbia (UBC), Vancouver, under the direct supervision and guidance of Dr. Clarence W. de Silva, Professor in the Department of Mechanical Engineering at UBC. The technical content of each chapter, including literature survey, algorithm development and implementation, and experimentation, was all completed by Jiahong Chen under the guidance and advise of Dr. Clarence W. de Silva. In particular, Dr. de Silva proposed and supervised the overall research project, suggested the research topics, obtained funding, provided research facilities and field test opportunities, and revised and refined the dissertation and other publications.

Chapter 2 and Chapter 6 are based on the work published on:

- Jiahong Chen, Teng Li, Tongxin Shu, and Clarence W. de Silva, "Rapidly-exploring Tree with Linear Reduction: A Near-optimal approach for Spatiotemporal Sensor Deployment in Aquatic Fields using Minimal Sensor Nodes.", IEEE Sensors Journal, vol. 18, no. 24, pp. 10225-10239, 15 Dec.15, 2018.

Jiahong Chen was responsible for all major areas of problem formulation, algorithm preparation, and implementation, experimental validation, as well as manuscript composition. Teng Li and Tongxin Shu were involved in the early stages of concept formulation and contributed to manuscript editing. The prototype of the unmanned surface vehicle was developed collaboratively by Teng Li, Tongxin Shu and me, in the UBC Industrial Automation Laboratory, with the guidance of Dr. Clarence W. de Silva. Dr. Clarence W. de Silva

was the supervisory author on this work and was involved throughout the project in concept formation and manuscript composition, as indicated above.

A version of Chapter 3 and Chapter 4 has been submitted to a journal:

- Jiahong Chen, Tongxin Shu, Teng Li, and Clarence W. de Silva, "Deep Reinforced Learning Tree for Spatiotemporal Monitoring with Wireless Sensor Network.".

Jiahong Chen was responsible for all major areas of problem formulation, algorithm preparation and implementation, numerical experimentation, and manuscript composition. Tongxin Shu and Teng Li were involved in the early stages of concept formulation and contributed to manuscript editing. Dr. Clarence W. de Silva was the supervisory author on this work and was involved throughout the project in concept formation and manuscript composition, as indicated above.

A version of Chapter 5 is to be submitted to a journal:

- Jiahong Chen and Clarence W. de Silva, "Sparse Sensor Deployment for Spatiotemporal Field Reconstruction and Prediction.".

Jiahong Chen was responsible for all major areas of problem formulation, algorithm preparation and implementation, numerical simulation, and manuscript composition. Dr. Clarence W. de Silva was the supervisory author on this work and was involved throughout the project in concept formation and manuscript composition, as noted above.

Table of Contents

List of Tables

List of Figures

Nomenclature

$\mathcal{A}$	a two-dimensional environment of interest
$\mathbf{A}$	temporal information matrix
$\mathbb{A}$	action set for reinforcement learning
$\mathbf{b}$	bias vector
$\mathbf{C}$	measurement matrix
η	infinite horizon cost
$\mathbb{E}$	neighbors of a sampling location
E	edges of a space-filling tree
$\mathcal{G}$	deep learning framework
$\mathcal{G}_r$	reconstruction framework in $\mathcal{G}$
G	space-filling tree
$\mathcal{H}(\cdot)$	entropy function
$\mathbf{I}$	information matrix of Gaussian Elimination
$\mathcal{J}$	set of redundant sampling locations
$\mathbf{J}$	matrix of ones
$\mathcal{M}$	set of sensor readings
$\mathbf{M}$	sensor reading matrix
$\mathcal{MI}(\cdot)$	Mutual Information function
$\mathcal{N}$	deployment strategy
$\hat{\mathbb{N}}_i$	a set of deployment strategies at iteration i

$\mathbf{P}$	estimation error matrix
$\pi(\cdot)$	maximum eigenvalue of a matrix
ψ	a location in the two-dimensional environment
$\boldsymbol{\phi}$	high-dimensional spatiotemporal field
$\boldsymbol{\Psi}$	principal basis
$\mathbf{R}_{\cdot}$	covariance matrix of Gaussian white noise
ρ	step size for field exploration methods
$\mathbf{S}_{\psi}$	the spatial information matrix at location ψ
Σ	covariance matrix
$\mathcal{T}(\cdot)$	element-wise nonlinear rectifier function
V	vertices of a space-filling tree
$v_{\psi}[t]$	an observation in location ψ at time t
$\mathbf{W}$	weight matrix
$\mathbf{x}[t]$	time-varying coefficient vector

Glossary

CS Compressive sensing

DL Deep learning

DNN Deep neural network

DO Dissolved oxygen

DRL Deep Reinforcement Learning

DRLT Deep Reinforced Learning Tree

EOI Environment of Interest

GAN Generative adversarial networks

GB GigaByte

GPS Global Positioning System

LSTM Long short-term memory

MOO Multi-objective optimization

MSE Mean square error

MSN Mobile sensor network

NCEI National Centers for Environmental Information

NCOM Naval Oceanographic Office Regional Navy Coastal Ocean Model

NOAA National Oceanic and Atmospheric Administration

ORP Oxidation-reduction potential

PCA Principal component analysis

PREC PRECipitation dataset

ReLU Rectified Linear Units

RL Reinforcement learning

RMSE Root mean square error

ROS Robot operating systems

RRC Rapidly-exploring random cycles

RRCstar Rapidly-exploring random cycles star

RRLR Rapid Random exploring tree with Linear Reduction

RRT Rapidly-exploring random tree

SAFER Spatiotemporal sparse Auto FiEld Reconstructor

SLAM Simultaneous localization and mapping

SVD Singular value decomposition

USV Unmanned surface vehicle

VAE Variational auto-encoder

WSN Wireless sensor network

Acknowledgments

First and foremost, I wish to express my deepest gratitude to my research supervisor, Professor Clarence W. de Silva, who has continuously given me rigorous advice and encouragement throughout my entire research program at the University of British Columbia. His invaluable suggestions not only pushed my critical thinking to a higher level but also taught me to become an independent researcher. I truly appreciate all the support and advice that he provided throughout the years.

I wish to thank all my colleagues in the Industrial Automation Laboratory (IAL), Dr. Yunfei Zhang, Mr. Shan Xiao, Dr. Muhammad Tufail Khan, Dr. Lili Meng, Dr. Min Xia, Dr. Shujun Gao, Mr. Hani Balkhair, Dr. Teng Li, Mr. Zhuo Chen, Mr. Tongxin Shu, Mr. Sheikh Tanvir, Mr. Fan Yang, Mr. Lucas Falch, Ms. Swapna Premasiri, Mr. Hiroshan Gunawardane, Mr. Jing Wang, Mr. Kuangen Zhang for their friendship and help on both academic affairs and personal life.

Additionally, I would like to acknowledge the financial support for the research presented in this thesis, through the research grants from IC-IMPACTS: India-Canada Centre for Innovative Multidisciplinary Partnership to Accelerate Community Transformation and Sustainability, obtained by Dr. Clarence W. de Silva as the Principal Investigator, and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation held by Dr. Clarence W. de Silva.

In particular and closing, I want to thank my family for their unquestionable support throughout my upbringing.

Chapter 1

Introduction

1.1 Motivation

In recent decades, public awareness has been considerably raised with regard to various types of environmental pollution issues. Environment-related problems significantly affect public health, in both developing and developed countries. In this context, environmental monitoring systems can be designed to sample the key attributes that adversely affect the living environment and then inform the public and relevant authorities about the latest undesirable conditions and take corrective actions. Deployed sensors may comprise a wireless sensor network (WSN) that can carry out different types of tasks such as persistent monitoring [1], event detection [2], and spatiotemporal field estimation and reconstruction [3, 4].

Typically, an environmental monitoring system consists of a group of sensor nodes deployed in an environment of interest (EOI). In practice, only a limited number of sensor nodes are deployed in critical and fixed locations due to a limitation of resources, such as battery life and the cost of sensor node components. It is therefore essential to allocate sensor nodes to optimal locations and possible move them as appropriate when the optimal conditions change. In this manner, a deployed sensor network can receive the greatest amount of information while maintaining an acceptable environmental monitoring estima-

tion error.

Traditionally, dense deployment has been performed to achieve high monitoring resolution for a field. The deployed static sensors were assumed to detect events within a specific sensing range, which is regarded as the coverage area of sensor nodes. As the sensor network spreads over the field, the EOI can be fully covered. Originally, researchers tended to deploy sensor nodes randomly in the field; efficient data transmission protocols and the sleep-active scheduling method were then used to increase the network lifetime [5, 6]. However, randomly deployed sensor nodes in a field can result in high infrastructure costs and redundant area coverage. Hence, the sensor nodes should be deployed within an infinite horizon and dynamically according to suitable optimizing criteria.

In contrast to static sensor networks, mobile sensor networks provide mobility to sensor nodes, enabling them to move as appropriate within the EOI. In this manner, mobile sensors can optimize their locations to attain sufficient area coverage in an effective manner. The mobility of sensor nodes may also help the sensor network adapt to environmental changes. Researchers have made great efforts to maximize the coverage ratio and minimize the traveling distance of the mobile sensors [7, 8]. Typically, a small number of mobile sensors are randomly deployed or placed along a narrow area. A planning algorithm then instructs each mobile agent to move around to increase the coverage ratio while minimizing the traveling distance. In this manner, the maximum coverage area will be provided by the deployed sensors in an "optimal" manner. Redundant sensors can be removed or relocated to further optimize the system and reduce costs.

However, if the EOI is large, deploying sensors in a distributed manner may still require a large number of sensor nodes. Hence, the EOI can be determined as a spatiotemporal field, and sampling-based deployment strategies can be developed to use only a small number of sensors to estimate the entire spatiotemporal field optimally. The sensor deployment locations of the sampling-based strategies are determined by minimizing the estimation error over the EOI, or by maximizing the information gain from the field. Novel algorithms

have been developed to explore the spatiotemporal field efficiently and find the near-optimal sampling locations for the sampling data [3, 9]. These approaches first model the EOI as an analytical model. Mobile sensor nodes are then sent out to explore the EOI. Near-optimal sampling locations can subsequently be determined by minimizing the field estimation error.

Moreover, maximum information can be retrieved from the EOI by selecting the most informative subset from the sampling locations [10]. Information theory-based methods calculate the entropy or mutual information of the sampling data, after which the subset of the sampling data that provides maximum information is selected. Thus, the corresponding locations of the selected subset can be determined as the most informative sampling locations, and the maximum information gain can be achieved from the EOI.

In some cases, merely achieving minimal estimation error in a field might result in a waste of limited resources. Sensory data collected from the deployment locations may have some redundancy according to specific criteria, such as linear dependency or information theory. In such cases, the selected near-optimal sampling locations can be further reduced to minimize the number of deployed sensor nodes [11]. A redundancy-removing algorithm can first find the optimal sensor deployment location set, which contains a set of sensors that achieves the lowest estimation error in the EOI. The correlation of various sampling locations is then investigated. If the sampling data at one or more locations can be reconstructed from the remainder of the observations, they are regarded as redundant locations. By removing these redundant locations, the most informative subset can be selected from the optimal set to reduce the number of sensor nodes.

Currently, efficient optimization methods, such as RRT-based approaches, heavily depend on exploration metrics, which may result in extensive computing time. The rapidly-exploring random tree (RRT) algorithm, a common exploration method, can efficiently explore an infinite horizon [12]. Randomly selecting the subsequent sampling point in a field guarantees an optimal solution for deployment locations; however, continuously sampling random locations may result in a duplication of efforts. Hence, information acquired from

sampled locations can be utilized to accelerate the exploration [13]. By learning the quality of sampled locations, reinforcement learning (RL)-based algorithms can decide the best future exploration steps given the current sampling locations. Furthermore, geodesic distance, information gain, and estimation error can be considered as the sampling reward, which helps to avoid unnecessary exploration locations.

Given the observations from the optimal sampling locations, it is also essential to predict the full state of the EOI. Recently, researchers have attempted to reconstruct the EOI according to the spatiotemporal correlations of the sampling data [4]. Statistical methods, such as principal component analysis (PCA) and compressive sensing (CS), have been applied to perform signal compression and reconstruction. These approaches project the original signal onto a lower-dimensional space, which indicates essential parts of the original signal. Thus, the corresponding sampling location can be selected as the critical sampling location for reconstruction, and used to interpret how the signal is encoded into this low-dimensional space. Statistical methods can then decode the observations from the low-dimensional space to the original signal. Hence, given future observations from the selected sampling location, the entire spatiotemporal field can be predicted. However, traditional statistical approaches may have limitations in capturing both spatial and temporal correlations. Consequently, Deep Learning (DL) may be utilized to reconstruct signals at higher efficiency and precision.

1.2 Research Objectives

It is important to identify and evaluate the performance and the weaknesses of existing environmental monitoring systems, with the aim of conducting persistent monitoring, spatiotemporal reconstruction, and the information-based sampling of the systems. Optimization methods can then be implemented to generate more efficient deployment strategies and to precisely reconstruct the spatiotemporal field according to the sampled data. The primary problem in this context is to find, using limited observations, an efficient and near-optimal

sensor deployment strategy for estimating and reconstructing the spatiotemporal field.

The first goal of this dissertation is to design a near-optimal sensor node deployment algorithms with minimal sensor nodes for comprehensive environmental monitoring. The second goal is to develop efficient exploration methods to accelerate the optimization. The sampling-based near-optimal strategy that is developed for sensor node deployment should be capable of learning sampled data and taking intelligent sampling actions. The third goal is to reconstruct and predict the extensive spatiotemporal field, using limited data. Since only a limited number of sensor nodes are deployed in the spatiotemporal field, the observations taken from these sampling locations should be able to accurately reconstruct the full scenario.

1.3 Related Work

1.3.1 Field Coverage with a Static Sensor Network

A static WSN, which is the most widely used type of sensor network in real-world applications, provides high system redundancy and monitoring resolution over the EOI.

In the beginning, Vuran et al. worked on addressing challenges with regard to WSNs that were densely deployed at fixed locations and used a large number of sensor nodes to continuously observe physical phenomenon [5]. The spatial and temporal correlations within the WSNs were utilized for the development of efficient communication protocols to save network costs and energy consumption. Next, Yoon et al. focused on the spatiotemporal correlations of sensed data in WSNs and the inherent physics of the environment [6]. The researchers developed a clustering algorithm to cluster nodes according to similar sensing values and spatiotemporal correlation. Then, only one sensor reading is transmitted to one cluster, to reduce transmission costs.

However, the spatial correlation between sensors and cluster-heads may result in low accuracy and may not fully exploit spatial and temporal correlations. Moreover, the transmitted data cannot be used to reconstruct the entire network, and choosing an appropriate

threshold that suits the current environment would be extremely difficult.

Consequently, Gupta and his colleagues designed techniques that exploit spatiotemporal correlations in sensor data to minimize the communication and energy costs of data collection in a sensor network [1]. Their algorithm used linear regression methods to find highly correlated sensor nodes and cluster them into small subsets. The sensor network then needed to collect data from only one sensor node in each subset. Based on Guptas work, Alsheikh et al. utilized neural networks to determine redundant sensors [11]. The use of neural networks made the algorithm suitable for a variety of sensory data and improved the regression accuracy.

Villas et al. considered the case of large-scale, dense WSNs [2, 14]. With high-density deployment it is very likely to detect spatially and temporally correlated information in a small group of sensor nodes. Such information can be used to reduce the extent of communication and data exchange; by eliminating such redundancy, energy can be saved.

However, in these methods, since these sensor networks are static, and unnecessary sensor nodes are turned off but not removed, the deployment is not "optimal" and also there will be significant initial deployment cost. Thus, it is not desirable to deploy static sensor nodes arbitrarily and densely in large-scale field monitoring.

1.3.2 Field Coverage with Dynamic Sensor Network

In contrast to static sensor networks, mobile sensor networks empower sensor nodes with mobility, enabling them to provide dynamic coverage of the field of interest. In this manner, overlapping of the sensing areas by nodes can be minimized, and hence the number of sensor nodes can be reduced. The mobility of sensor nodes may also help the sensor network adapt to environmental changes in an optimal manner [15].

Heo and Varshney presented an autonomous deployment method for mobile WSNs [16, 17]. Their algorithm converts random deployment into uniform distribution with minimal sensor node movement. They proposed an algorithm to increase the ratio of region

coverage while using limited computational time and relatively short mean sensor node travel distance, which can extend the network lifetime over the field of interest.

Abo and his colleagues attempted to use robotic sensors to maintain dynamic coverage of a field for as long as possible [8, 18, 19]. The sensor node deployment strategy in the field is the essential factor related to the coverage; however, the traditional dense deployment method requires many static sensor nodes and can cause coverage gaps in the sensing field. Therefore, the researchers proposed a new deployment method based on multi-objective immune optimization to minimize the coverage gaps. The coverage area of the WSN is maximized by rescheduling the placement of the mobile sensors with limited mobility and communication connectivity. The sensors with mobility not only improve the coverage of the area but also reduce the cost of long-term deployment.

Although mobile WSNs succeeded in minimizing the number of sensors required in the field and the energy consumption of the network, they did not consider the sampling data correlation, which is widely viewed in static sensor network protocols. In particular, data redundancy can be minimized by modeling environmental dynamics, using information-based sampling strategies.

1.3.3 Spatiotemporal Monitoring with Wireless Robotic Sensor Networks

Typically, a wireless robotic sensor network consists of a fleet of mobile sensor nodes, which can be spatiotemporally deployed in an EOI. Information-based sampling methods utilize a limited number of mobile sensor nodes, which provide sufficient information from a field when deployed in relatively few critical locations. It is therefore essential to allocate sensor nodes to their optimal locations for receiving maximum possible information while minimizing the environmental monitoring estimation error. To achieve this goal, sensor nodes need to search dynamically within an infinite horizon with specific optimizing criteria [20], and strategies such as spatiotemporal sensor deployment can be used [21] [22].

Outfitting sensor nodes with mobility makes a sensor network more efficient and flex-

ible by reorganizing the sensor network and reducing the number of sensor nodes. These methods deploy a limited number of sensor nodes that can provide maximum useful information [23] or achieve the least estimation error for an environment [3]. Krause et al. proposed a mutual information approximation method that improves the sensor node placement quality [10]. They tested this criterion using real-world datasets and determined that mutual information can achieve lower root mean square error (RMSE) in the environmental estimation than entropy. Du et al. proposed a near-optimal sensor node deployment strategy for monitoring hydro-dynamics and water quality in urban areas. They used entropy [24] and mutual information [23]] to gather maximum information in selected locations [25] [26].

Lan et al. developed Rapidly-exploring random cycles (RRC) to estimate a spatiotemporal field in a dynamic environment by deciding the locations in the field where one or more robots should reach to acquire data [27–29]. RRC first models the environment using past data, based on which mobile sensor nodes are deployed to explore the EOI. Locations are then determined by minimizing the largest eigenvalue of the error covariance matrix of the field estimate error over an infinite horizon. By doing so, they seek to estimate the environment continuously with a minimum number of sensor nodes. Lan et al. proposed a new scheme called Rapidly-exploring Random Cycles star (RRCstar) that enhances RRC by escaping local minima [3]. RRCstar, a variant of RRC, seeks to find the lowest global estimation error over the field by rewiring those deployments with high estimation error.

1.3.4 Field Exploration Assisted by Reinforcement Learning

Outfitting sensor nodes with mobility to monitor infinite horizon problems has improved the performance and efficiency of WSNs significantly. Mobility approaches deploy a limited number of sensor nodes that can provide adequate amounts of useful information and achieve the minimum estimation error for an environment [30].

Field exploration algorithms, such as the RRT, have been further proposed to efficiently

search a non-convex field [20]. Karaman et al. evaluated the quality of solutions provided by RRTs. The proposed algorithm, RRT*, was proven to be asymptotically optimal as the number of samples increases [31]. Subsequently, sampling-based path planning algorithms have been used to study sampled data on-the-fly while exploring [3, 9, 32]. However, the performance of RRT-based algorithms is highly reliant on the distance measures that are applied, and the exploration strategy has low efficiency.

Sivamurugan et al. proposed a sampling-based policy iteration to explore an infinite horizon efficiently [13]. Specifically, reinforcement learning (RL) was used to study the relationship of geodesic distance among the sampled points on-the-fly, which significantly improved the computational efficiency of RRT-based methods. However, only geodesic distance was utilized as the cost function to calculate the rewards of the algorithm, which neglected spatiotemporal correlations of the sampling data. With the help of RL, algorithms can determine the best actions under the current state and converge to a near-optimal situation [33–38]. Nevertheless, more computation time is consumed by the algorithm to explore all possible states, as the environment and the states become more complex. Moreover, the applications of RL-based algorithms have been limited to environments with directly extractable features or to low-dimensional environments that are fully observable. Thus, DL has been introduced to approximate similar states, which enables automatic feature learning through the gradient descent process [39, 40].

Mnih et al. improved RL with the advantage of deep neural networks (DNNs) to address the issue about a complex environment [41, 42]. A DNN, which can generate a precise model of a complex nonlinear domain, is a useful tool in many fields, including image recognition [43], natural language processing [44], and bioinformatics [45]. Mnih et al. tested their Deep Reinforcement Learning (DRL) algorithms on the practical domain of Atari 2600 games [46]. In their algorithm, DRL uses only the pixels and the game score as inputs and achieves performance at the professional human level. Pfeiffer and his colleagues adopted a DRL model for path-planning problems with respect to sensor data and goal

information [47]. Robots can utilize such information to plan paths over an infinite horizon and efficiently reach most informative goal positions. Shiarlis et al. also proposed an RL-based scheme to avoid repeatedly calculating costly planning procedures [48]]; their method achieved better performance with increased computational efficiency. However, the above methods are suitable only for end-to-end navigation and not for searching over an EOI.

1.3.5 Spatiotemporal Field Reconstruction

Recently, data-driven approaches have been developed to find principal locations for optimal sensor deployment. After the sampled data from a spatiotemporal field are analyzed, and the principal locations are calculated, the entire spatiotemporal field can be reconstructed using a limited set of observations. Data-driven approaches are of low complexity and more amenable to implementation than model-based convex optimization approaches. These data-driven approaches are called sparse representations, which can define a relatively large number of regions, such as $O(2^N)$, with merely $O(N)$ examples [49].Numerous complex systems can be reconstructed and predicted using data-driven sparse representation approaches. Applications include image reconstruction [50], image denoising [51–53], image super-resolution [54, 55], and structured signal recovery [56].

Compressive Sensing (CS) is a widely used signal recovery method for unknown signal reconstruction that utilizes sparse representations [57, 58]. Brunton et al. proposed a CS-based sparse sensor placement method for characterizing or classifying a high-dimensional system [59]. Their algorithm solves the ℓ_1 minimization problem of finding the measurement with the fewest zero entities for reconstructing the original signal. Lu et al. then proposed a convolutional CS framework to avoid inefficiency and blocking artifacts in the traditional CS algorithms [60]. Their algorithm senses the input image using a set of convolutional filters and reconstructs the image using linear convolutional measurements. Although CS can recover a comprehensive class of signals, it has limitations in capturing spatiotemporal patterns and does not perform well for large signals. Moreover, CS-based

algorithms need to solve a linear program to recover low-dimensional signals, which is time-consuming and has low efficiency.

In contrast, principal component analysis (PCA) extracts low-dimensional patterns and features from high-dimensional systems. PCA transforms input data into their orthogonal coordinate system and projects the data onto coordinates according to the significance of the variance. The selected principal components then retain the most variance in the data and can easily reconstruct the system. Manohar et al. proposed an optimal sparse sensor deployment scheme based on QR factorization of the sampling data matrix and singular value decomposition (SVD) [4], which can outperform the CS-based approaches. Guo et al. developed a sampling technique based on sparse topological data analysis to reconstruct a high-dimensional signal from a limited set of observations [61]. The optimal sparse set of samples was selected and used to reconstruct the input signal at high efficiency and precision. Lu et al. used SVD to learn basis images from training data and perform sparse reconstruction of fluid flows from random samples [62]. They investigated the interplay of data sparsity in the underlying flow system and reconstructed the entire space using a limited set of random samples.

A variant of PCA, called Sparse PCA (SPCA), has been developed as well. The principal components of traditional PCA are usually linear combinations of all input data, which can lead to errors in interpreting the derived principal components [63–65]. Hence, to encode the high-order information of the input data, Jenatton et al. developed an extension of SPCA, which specializes in image recognition and reconstruction [66]. Hein et al. focused on nonlinear eigenproblems, which comprise the fundamental part of many machine learning and statistics problems [67]. They developed a generalized inverse power method that guarantees convergence to a nonlinear eigenvector. Drmac et al. proposed a novel framework to construct a discrete empirical interpolation method [68]. Their algorithm utilizes QR factorization with column pivoting to provide near-optimal sampling for the reconstruction of nonlinear systems. The algorithm, based on structured regularization in which

sparse patterns are structured and stored in different sets of predetermined shapes, achieves state-of-the-art performance when applied to clustering and SPCA problems. Erichson et al. modeled SPCA as a matrix factorization problem with orthogonality constraints [69]. Their algorithm minimizes the subset of projection variables by incorporating various sparsity regularizers. Janková et al. suggested a de-biased SPCA to estimate the first vector of loadings and the largest eigenvalue of the covariance matrix [70]. They identified the bias in the traditional Lasso-based estimator and guaranteed asymptotic normality of the proposed estimator. Mollenhauer et al. provided a solid analytical foundation for the eigenvalue decomposition of operators on reproducing kernel Hilbert spaces, and extended the method of using SVDs to solve the eigenvalue decomposition problem [71].

1.3.6 Optimal Sparse Representation Through Deep Learning

Recently, DL methods have been utilized to improve the accuracy of signal reconstruction. Zeiler et al. investigated the functionality of convolution networks and deconvolution networks in image classification [72, 73]. Convolution networks downscale an original image into a low-rank feature space, which can be regarded as compressed information. In contrast, the unpooling of deconvolution networks reconstructs the compressed feature back to the original signal. With the help of deep convolution/deconvolution networks, the performance of compression and reconstruction improves significantly. Mousavi et al. developed a DL framework for recovering structured signals [56], using DL frameworks to capture nonlinear relationships. The proposed stacked denoising autoencoder is an unsupervised feature learner that captures statistical dependencies between various observations and significantly improves signal recovery quality. Loh et al. built a DL framework to learn the event reconstruction problem by placing photon sensors in an antineutrino detector [74].

Furthermore, generative models have succeeded in modeling data distributions using neural networks. Kingma et al. proposed variational auto-encoders (VAEs) [75] to encode a complex signal into a low-dimensional space and decode it. Goodfellow et al. then de-

veloped generative adversarial networks (GANs), which either generate superficial images or automatically encode input signals into compressed signals, using two neural networks that contest with each other [76]. Bora et al. proposed a generative model based on the compressive sensing method to use fewer measurements to achieve high accuracy [77]. The generative model is built upon VAEs and GANs. The researchers also proved that given the Lipschitz constant of the proposed algorithm, recovery performance can be guaranteed within a certain probability and error rate. Calculation of the Lipschitz constant was investigated in [78], where Gouk et al. discussed the effects of regularizing neural networks with Lipschitz continuity. They provided a method for calculating the Lipschitz constant of a neural network and then used the calculated Lipschitz constant to formulate the training process of the neural network. The performance of neural networks has cumulatively improved through regularization using Lipschitz continuity.

However, DL-based approaches usually encode the original signal into a low-dimensional space with a complex model, which can exacerbate the difficulty of deciding the most critical observation locations.

1.4 Contributions and Organization of the Dissertation

The main contributions of the dissertation are listed below.

- A near-optimal sensor node deployment strategy for deploying a minimal number of sensor nodes in a spatiotemporal field is developed. This strategy relocates sensor nodes to their best sensing locations while reducing the required number of sensor nodes without losing information. The approach is based on an environmental model and the linear dependence of sensor readings. In particular, the spatiotemporal correlation of sensor node deployment in a large geographic EOI is minimized. It is also proven that the optimization objective function, which uses the prior estimation error, is submodular and results in a near-optimal solution. Compared with other sensor node deployment strategies and baseline algorithms, this new strategy requires much

fewer sensor nodes to achieve the same, or lower, estimation error. The proposed algorithm is applied to a real-world dataset, which is collected by the developed robotic sensor nodes. The in situ tests confirm that the proposed near-optimal sensor node deployment strategy can achieve lowest field estimation error with least sensor nodes.

- A DRL framework is proposed to improve the efficiency of searching for the most informative sampling locations. The parameterized sampling locations in an infinite horizon space are modeled according to their spatiotemporal correlations and subject to various constraints, including field estimation error and information gain. As the model-based information gain can be efficiently calculated over an infinite horizon, the effectiveness of the sampling locations is learned during exploration by the robotic sensors. The robotic sensors are then instructed to avoid unnecessary sampling locations in future iterations. Also, it is proven that the proposed algorithm is capable of effectively searching for near-optimal sampling locations and guaranteeing a minimum field estimation error. Simulation based on the Naval Oceanographic Office Regional Navy Coastal Ocean Model (NCOM) is presented, which demonstrates the significant enhancements made by the proposed algorithm. Compared with traditional approaches, such as information theory, which is based on a greedy approach, or random sampling, the proposed method shows superior performance with regard to both estimation error and planning efficiency. Besides, a model-based statistical approach is developed to reconstruct the entire spatiotemporal field with the observations collected from the most informative sampling locations. The results show that the most informative sampling locations generated by the DRL framework can near-optimally condition the spatiotemporal field reconstruction.

- A model-free approach is developed to reconstruct the spatiotemporal field from limited observations. The sampling locations in an infinite horizon are obtained via canonical basis compression according to their spatiotemporal correlations and sub-

ject to reconstruction performance. Then, a simulated annealing-based approach is utilized to optimize the canonical basis for the optimal sampling locations. The sparse representations of the spatiotemporal field given by the canonical basis indicate optimal sampling locations. Auto-encoders are then tuned to reconstruct the spatiotemporal field from the observations in the selected sampling locations. In this manner, reconstruction from the sparse representation to the spatiotemporal field is learned. It is also proven that the proposed model is Lipschitz continuous and that the sparse sampling strategy is sufficient for reconstruction.

The organization of the dissertation is as follows:

Chapter 2 introduces the developed near-optimal sensor node deployment strategy, which is termed Rapid Random exploring tree with Linear Reduction and Connectivity (RRLR Connectivity). Linear correlations are investigated among the sampling data to identify spatiotemporally redundant observations. The corresponding sensor nodes of the redundant observations are then removed to reduce the number of deployed sensor nodes, with regard to the network connectivity. Besides, details regarding the proof about submodularity and near-optimal guarantee are provided.

Chapter 3 presents a novel spatiotemporal field exploration method called Deep Reinforced Learning Tree (DRLT), which can learn the information gain at sampling locations during exploration. The area with a lower information gain will subsequently have decreased probability of exploration. In this manner, the speed of finding the near-optimal sampling locations will increase.

Chapter 4 presents a model-based sensor deployment strategy for spatio-temporal field reconstruction. It shows that the model-based informative sampling strategy that is proposed in Chapter 3 can near-optimally condition the reconstruction of the spatiotemporal field. Also, a statistical method is developed to learn the linear mapping from observations of the informative sampling locations to the entire spatiotemporal field.

Chapter 5 presents a model-free deep learning framework for spatiotemporal field recon-

struction, called Spatiotemporal sparse Auto FiEld Reconstructor (SAFER). The canonical basis is utilized to compress the spatiotemporal field into limited observations, and nonlinear mapping from the observations to the original spatiotemporal field is learned. Further, proofs regarding Lipschitz continuity of the deep learning framework and sufficiency for the reconstruction of sparse representation are provided.

Chapter 6 illustrates the hardware design, the implementation of the robotic sensor module, and the results of the in situ experimentation. An environmental model is established based on the sampling data from the conducted in situ tests, according to the metrics introduced in Chapter 2. Then, the proposed methods are applied to the established environment model to test its effectiveness.

Chapter 7 concludes the dissertation by summarizing the main contributions, and indicating possible future work.

Chapter 2

Spatiotemporal Sensor Deployment using Minimal Sensor Nodes

2.1 Introduction

In recent decades, different types of pollution issues have significantly raised the public awareness, where water pollution has become a key concern. Water-related problems severely affect peoples health not only in developing countries but also in developed countries. In this context, an environmental monitoring system has been designed in our laboratory to sample the critical water quality attributes, determine the quality of water, and inform the public and relevant authorities the obtained results with view to possible corrective actions.

The layout of the developed environmental monitoring system is shown in Figure 2.1. In the beginning, technologies such as satellite imaging, mobile WSNs, and static WSNs may be utilized for sampling the conditions at several locations in the field. Then, based on the time series data sourced from different locations, methods like kriging can be used to establish a spatiotemporal model for the existing environment [79]. Next, based on this environmental model, field exploration method and heuristic algorithms can be developed to find the optimal sensor node deployment locations that meet specific optimization criteria

(e.g., receiving most information, achieving least estimation error, minimizing the power usage). Given these deployment locations, the redundant sensor nodes may be removed to keep a minimal number of sensors in the field to reduce infrastructure costs. In this manner, the deployed sensor nodes can form a WSN that persistently monitors the most informative locations in the field for sensing.

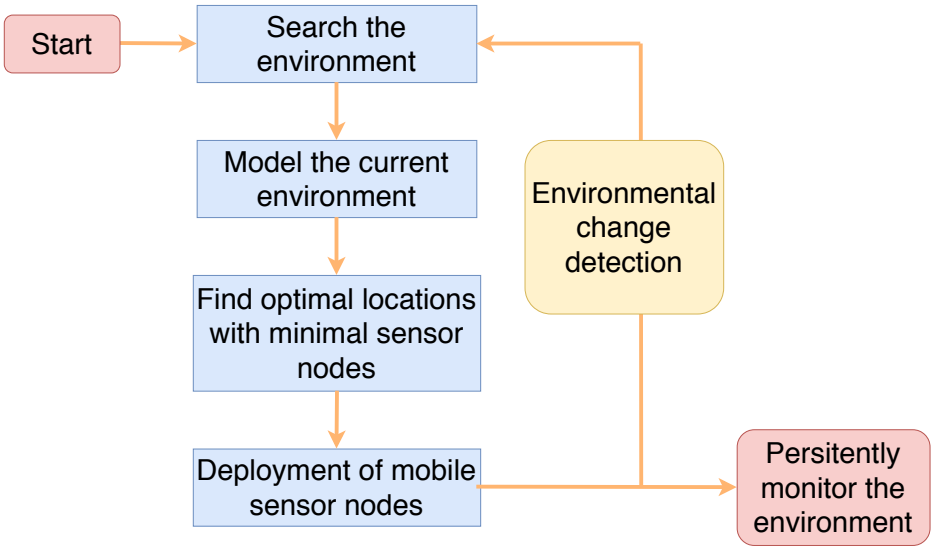


Figure 2.1: The framework of optimal sensor node deployment with minimal sensor nodes.

It is essential to preserve the network connectivity of a WSN while eliminating possible redundant sensor nodes. Traditionally, the selection of informative sampling locations does not consider the correlation of sampling data or the network connectivity. Hence, the spatiotemporal filed might be over-sampled or under-sampled. Over-sampled sensor deployment strategies do not consider the spatiotemporal correlation of the sampling data and will result in sensor node redundancy. Under-sampled sensor deployment strategies cannot provide sufficient sensor nodes to ensure network connectivity.

In this chapter, a method is proposed to identify proper nodes and remove linearly redundant sensor nodes efficiently while maintaining network connectivity. Through this process, the optimal sensor node deployment locations that achieve the lowest estimation error and share least linear correlation are determined. Based on these selected deployment locations,

mobile sensor nodes can be deployed to monitor the geographic EOI persistently. When a change in the geographic EOI is detected, the above process has to be repeated to generate a new model that satisfies the current environment.

The underlying method as proposed in this dissertation is called Rapid Random exploring tree with Linear Reduction and Connectivity (RRLR Connectivity). RRLR Connectivity is a near-optimal strategy of sensor node deployment, which is developed here to optimize sensor node deployment locations for field estimation. Initially, it is proved that using prior estimation error as the optimization function has a characteristic of submodularity, which will result in a near-optimal solution. Then, an RRT-based approach is developed to search for possible sensor node deployment locations by building space-filling trees. During the tree expansion, sampling locations with the lowest prior estimation error are found. Then, linearly correlated sampling data are investigated, and the redundant sampling locations are eliminated according to the network connectivity. In essence, the proposed method significantly reduces the number of sensor nodes required for persistent monitoring while achieving a lower or similar prior estimation error.

2.2 Preliminaries

2.2.1 Modeling of Environment

A common environmental model that is used for geological and environmental systems is chosen in this dissertation [79]. Consider a discrete time-variant scalar field $v_\psi[t] \subset \mathbb{R}$, where $\psi \in \mathcal{A}$ is an arbitrary point in the two-dimensional EOI ($\mathcal{A}$), which can be non-convex and filled with obstacles. Then, the environmental model can be represented as the output value of desired locations at a given timestamp. The model can be expressed as a function of the temporal information matrix ($\mathbf{A}$), the spatial information matrix ($\mathbf{S}_\psi$), location in the field ψ and time t:

$$v_\psi[t] = F(\mathbf{A}, \mathbf{S}_\psi, \psi, t) \tag{2.1}$$

The spatiotemporal environmental model is considered to be separable [80]. Thus, $v_\psi[t]$ may be decomposed as the dot product of the spatial information $\mathbf{S}_\psi$ and the time-varying coefficient vector $\mathbf{x}[t]$[3]:

$$v_\psi[t] = \mathbf{S}_\psi \mathbf{x}[t] + v[t] \tag{2.2}$$

where $v[t] \sim N(0, \mathbf{R}_v)$ is Gaussian white noise, and $\mathbf{R}_v$ is its covariance matrix, which is positive semidefinite. The noise is assumed to be time independent with $\mathbf{E}\{v[i]v[j]^T\} = 0, \forall i \neq j$. For simplicity, the variance of different noise is assumed to be the same [80].

The temporal information matrix $\mathbf{A}$, can be used for calculating the time-varying coefficient vector $\mathbf{x}[t] = [x_{t_1} \cdots x_{t_k}]^T$, which is regard as the state variable of a linear discrete time-invariant system:

$$\mathbf{x}[t+1] = \mathbf{A}\mathbf{x}[t] + w[t] \tag{2.3}$$

where $w[t] \sim N(0, \mathbf{R}_w)$, is Gaussian white noise and $\mathbf{R}_w$ is its covariance matrix, which is positive semidefinite. Also, $\mathbf{E}\{w[i]w[j]^T\} = 0, \forall i \neq j$. Since v and w are uncorrelated, $\mathbf{E}\{w[i]v[j]^T\} = 0, \forall i, j$. In this dissertation, temporal information matrix $\mathbf{A}$ and its covariance matrix $\mathbf{R}_w$ can be learnt from sampled data at the station using subspace identification methods [81, 82], with the raw observation data from National Centers for Environmental Information (NCEI) of U.S. National Oceanic and Atmospheric Administration (NOAA) [83–85].

Then, the spatial information matrix $\mathbf{S}_\psi$ is the row vector consisting of k spatial basis functions [79]:

$$\mathbf{S}_\psi = [s_1(\psi), \cdots, s_k(\psi)] \tag{2.4}$$

Each basis function is represented by a Gaussian basis function as $s_i(\psi) = Ke^{-\frac{||(\psi)-(\psi_i)||^2}{2\sigma^2}}$, where ψ_i represents the location of the ith basis center, and σ is its variance.

In this manner, Equations 2.2 and 2.3 jointly represent a linear discrete time-invariant system, and the spatiotemporal information is captured. Although due to its robustness the Gaussian radial basis function is selected to represent the basis, the proposed method does

not rely on the specific type of basis function.

2.2.2 Estimation Error of Deployment Location

It has been shown that the linear Kalman filter is suitable for estimating time-varying coefficients, and its prior error covariance matrix can be used as a quality measure for the criteria. It is an optimal estimator that minimizes the mean square error (MSE) of the system [80] [86]. The prior estimation of the linear Kalman filter is used for updating the error covariance matrix as:

$$\mathbf{P}[t+1|t] = \mathbf{A}\mathbf{P}[t|t]\mathbf{A}^T + \mathbf{R}_w \tag{2.5}$$

where:

$$\mathbf{P}[t|t] = \mathbf{P}[t|t-1] - \mathbf{P}[t|t-1]\mathbf{S}_{\psi_i}^T \times (\mathbf{S}_{\psi_i}\mathbf{P}[t|t-1]\mathbf{S}_{\psi_i}^T + \mathbf{R}_v)^{-1}\mathbf{S}_{\psi_i}\mathbf{P}[t|t-1] \tag{2.6}$$

Hence, the prior estimation error matrix of the linear Kalman filter can be represented as:

$$\begin{aligned}\mathbf{P}[t+1|t] &= \mathbf{A}\mathbf{P}[t|t-1]\mathbf{A}^T \\ &- \mathbf{A}\mathbf{P}[t|t-1]\mathbf{S}_{\psi_i}^T \times (\mathbf{S}_{\psi_i}\mathbf{P}[t|t-1]\mathbf{S}_{\psi_i}^T + \mathbf{R}_v)^{-1}\mathbf{S}_{\psi_i}\mathbf{P}[t|t-1]\mathbf{A}^T + \mathbf{R}_w\end{aligned} \tag{2.7}$$

The maximum eigenvalue of the prior covariance matrix, $\pi(\mathbf{P}[t+1|t])$, is the objective function for optimization because the largest eigenvalue represents the worst estimation accuracy at the current point. Furthermore, it has been shown that to remove the effect of transient high error of estimation in the beginning, the prior error (persistent monitoring error) should be evaluated as time goes to infinity. Also, when there are several deployment locations in each deployment strategy, estimation about a single sensor node in one deployment strategy will converge as time goes to infinity [3]. It can be represented as:

$$\mathbf{P}_{\infty}^{i} = \mathbf{A}\mathbf{P}_{\infty}^{i}\mathbf{A}^T - \mathbf{A}\mathbf{P}_{\infty}^{i}\mathbf{S}_{\psi_i}^T \times (\mathbf{S}_{\psi_i}\mathbf{P}_{\infty}^{i}\mathbf{S}_{\psi_i}^T + \mathbf{R}_v)^{-1}\mathbf{S}_{\psi_i}\mathbf{P}_{\infty}^{i}\mathbf{A}^T + \mathbf{R}_w \tag{2.8}$$

where $\mathbf{P}_{\infty}^{i}$ indicates the prior error covariance for the ith deployment location of strategy $\mathcal{N}$ as time goes to infinity.

Then, the minimum largest eigenvalue in each possible deployment strategy is the cost:

$$\Gamma(\mathcal{N}) = \pi(\mathbf{P}_{\infty}^{\mathcal{N}}) = \max_{i=1,2,\ldots} \pi(\mathbf{P}_{\infty}^{i}) \tag{2.9}$$

where $\mathcal{N}$ is the deployment strategy containing several deployment locations that are selected by the algorithm, $\mathbf{P}_{\infty}^{i}$ indicates the prior error covariance for the ith deployment location of strategy $\mathcal{N}$ as time goes to infinity, and $\pi(\mathbf{P}_{\infty}^{\mathcal{N}})$ denotes the largest eigenvalue of the prior error covariance matrix for strategy $\mathcal{N}$ as time goes to infinity.

2.2.3 Matrix Linear Redundancy

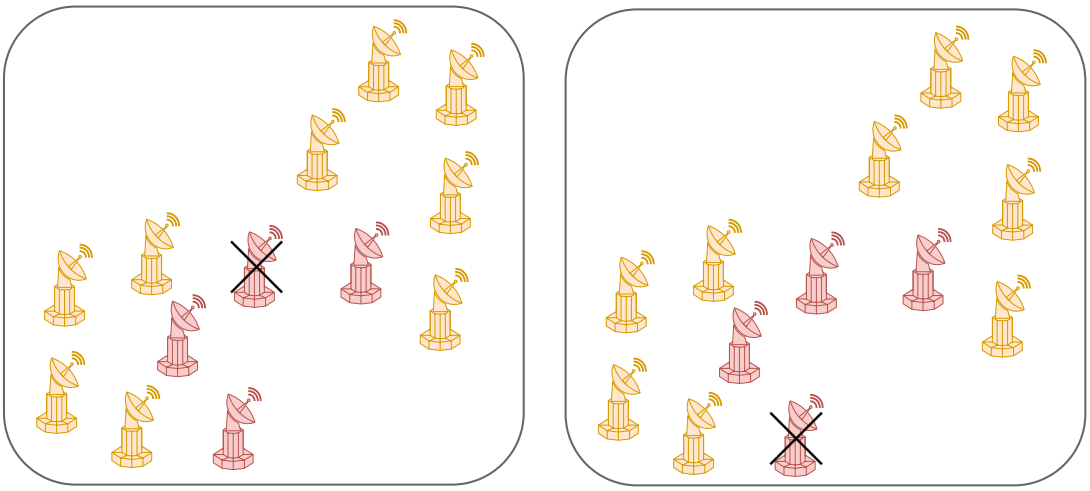
Assume that a set of sensor nodes, $\mathcal{N} = \{n_1, n_2, \cdots, n_k\}$, is persistently deployed in a geographic EOI. Denote sensor readings for M snapshots as $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k\}$, where $\mathbf{m} \in \mathbb{R}^{M\times 1}$. Hence, the readings of these k sensor nodes can be placed in a matrix $\mathbf{M} \in \mathbb{R}^{k\times M}$ as $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_k]^T$. The goal is to find if there are readings $\mathbf{m}_i$ that can be linearly represented by other data rows in matrix $\mathbf{M}$; i.e., $\mathbf{M}$ does not have the full rank, and these rows can be expressed as a linear combination of others:

$$\mathbf{m}_i = \sum k_j \cdot \mathbf{m}_j, (k_j \neq 0 \text{ and } \mathbf{m}_i, \mathbf{m}_j \in \mathcal{M}). \tag{2.10}$$

In this way, the redundancy of the node deployment can be indicated by the linear dependent relationship of sensor node readings when they are persistently deployed. Any nonzero $\mathbf{m}_j$ with the associated nonzero k_j is then defined as a correlated neighbor of $\mathbf{m}_i$. Then, $\mathbf{m}_i$ together with its correlated neighbors comprise a linearly correlated set Φ. If several nodes are linearly dependent in set Φ, information from some of these nodes (called redundant nodes) can be inferred from other nodes in this set.

2.2.4 WSN Connectivity of the Informative Sampling Locations

The deployed sensor nodes comprise a wireless sensor network. During the process of redundant sensor node elimination, the network connectivity might be broken if the remaining sensor nodes are far from each other. As shown in Figure 2.2, red sensor nodes denote the ones that have spatiotemporally correlated sampling data, and one of them can be removed. However, eliminating the sensor node as in Figure 2.2 (a) might break the connectivity of the WSN due to the increase of communication distance. Instead, removing the sensor node as in Figure 2.2 (b) can preserve the sensor network connectivity while eliminating the redundant sensor node.



(a) Network connectivity broken due to redundant sensor node removal.

(b) Network connectivity preserved after redundant sensor node removal.

Figure 2.2: Illustration of network connectivity. (Red sensor nodes have spatiotemporally correlated sampling data.)

In this dissertation, the network connectivity is assumed to be based on the communication distance of the robotic sensors and does not depend on the node size. Therefore, the network connectivity can be formulated as:

$$\forall n_i, n_j \in \mathcal{N}, \exists k\text{-hop communication path.} \tag{2.11}$$

For RRT-based informative sampling strategies, the distance between any 1-hop sensor

nodes will be less than its exploration step size. Hence, making the step size less than the communication distance can preserve the sensor networks connectivity. However, removing the redundant sensor nodes may break the sensor network connectivity, as shown in Figure 2.2. Thus, the constraint of network connectivity should be considered during the optimization of sampling locations.

2.2.5 Problem Formulation

Now, the informative sensor deployment problem can be formulated mathematically. Assume that the spatial sensing field is geographically infinite since every location in the field might be explored to obtain a better estimation of the environment. A small subset of locations, $\mathcal{N} \subseteq \mathcal{A}$, can then choose to best estimate the environment. The quality of the selected deployment strategy $\mathcal{N}$ can be evaluated by $\Gamma(\mathcal{N})$, which denotes the persistent estimation error and the infinite horizon cost. Each deployment strategy $\mathcal{N}$ may have several deployment locations, and every sampling location will persistently sample data from the environment and generate the data matrix $\mathbf{M}$. Moreover, redundant sensor nodes should be eliminated while obtaining sufficient information and preserving the network connectivity.

The problem can be formally defined as:

$$\min_{\mathcal{N} \subseteq \mathcal{A}} \Gamma(\mathcal{N}), \tag{2.12}$$

subject to:

$$\begin{cases} |\mathbf{M}| = \text{rank}(\mathbf{M}) \\ Equation\ 2.11 \end{cases}$$

2.3 Rapidly-exploring Random Tree with Connected Linear Reduction

2.3.1 Optimization Scheme for Near-optimal Sensor Node Deployment

Selecting a small subset of locations over the infinite horizon to minimize the prior estimation error has been shown to be NP-hard [10]. Therefore, it is unlikely to find the optimal subset $\mathcal{N} \subseteq \mathcal{A}$ of this problem. Instead, a near-optimal solution is provided. In this chapter, Rapid-exploring random tree (RRT) is used to search a non-convex geographic EOI and find near-optimal solutions. RRT can efficiently build a space-filling tree $G = (V, E)$ over the infinite horizon, where V represents the vertices and E represents the edges. Space-filling tree G contains many spanning trees, whose vertices indicate the possible deployment locations within the field. Hence, the prior estimation error of the corresponding sampling locations can be determined, and the spanning tree with the smallest estimation error can be computed.

The critical factor for near-optimal optimization is that more observation will result in less information gain and estimation error reduction. Moreover, it leads to convergence of the maximization or minimization of the cost function. This concept is formalized as submodularity [87]. A function Γ is said to be submodular if:

$$\forall \mathcal{A}^* \subseteq \mathcal{A}^{**} \subseteq \mathcal{A} \text{ and } n \in \mathcal{A} \setminus \mathcal{A}^{**}$$

$$\Gamma(\mathcal{A}^* \cup n) - \Gamma(\mathcal{A}^*) \geq \Gamma(\mathcal{A}^{**} \cup n) - \Gamma(\mathcal{A}^{**})$$

Proposition 2.1 show that the cost function of the proposed method is submodular.

Proposition 2.1. *Function $\Gamma(\mathcal{N})$ is submodular.*

Proof. According to Equation 2.9, let $\mathcal{N} \subseteq \mathcal{N}^* \subseteq \mathcal{A}$ and a location $n \in \mathcal{A} \setminus \mathcal{N}^*$. The goal

is to satisfy following equation:

$$\Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) \geq \Gamma(\mathcal{N}^* \cup n) - \Gamma(\mathcal{N}^*). \tag{2.13}$$

Then, let $\mathcal{N} \cup \mathcal{A} = \mathcal{N}^*$,

$$\begin{aligned} \Gamma(\mathcal{N}^*) = \pi(\mathbf{P}_\infty^{\mathcal{N}^*}) &= \pi(\mathbf{P}_\infty^{\mathcal{N} \cup \mathcal{A}}) \\ &= \max(\max_{i \in \mathcal{N}} \pi(\mathbf{P}^i{}_\infty), \max_{i \in \mathcal{A}} \pi(\mathbf{P}^i{}_\infty)) \\ &= \max(\alpha, \beta), \end{aligned} \tag{2.14}$$

where $\alpha = \max_{i \in \mathcal{N}} \pi(\mathbf{P}^i{}_\infty)$, $\beta = \max_{i \in \mathcal{A}} \pi(\mathbf{P}^i{}_\infty))$. Thus, $\Gamma(\mathcal{N}) = \max_{i \in \mathcal{N}} \pi(\mathbf{P}^i{}_\infty) = \alpha$.

Similarly,

$$\begin{aligned} \Gamma(\mathcal{N}^* \cup n) = \pi(\mathbf{P}_\infty^{\mathcal{N}^* \cup n}) &= \pi(\mathbf{P}_\infty^{\mathcal{N} \cup \mathcal{A}} \cup n) \\ &= \max(\max_{i \in \mathcal{N}} \pi(\mathbf{P}^i{}_\infty), \max_{i \in \mathcal{A}} \pi(\mathbf{P}^i{}_\infty), \pi(\mathbf{P}^n{}_\infty)) \\ &= \max(\alpha, \beta, \gamma), \end{aligned} \tag{2.15}$$

where $\pi(\mathbf{P}^n{}_\infty) = \gamma$. Then,

$$\begin{aligned} \Gamma(\mathcal{N} \cup n) &= \pi(\mathbf{P}_\infty^{\mathcal{N} \cup n}) \\ &= \max(\max_{i \in \mathcal{N}} \pi(\mathbf{P}^i{}_\infty), \pi(\mathbf{P}^n{}_\infty)) \\ &= \max(\alpha, \gamma). \end{aligned} \tag{2.16}$$

Next,

$$\begin{aligned} &\Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) - \Gamma(\mathcal{N}^* \cup n) + \Gamma(\mathcal{N}^*) \\ =& \max(\alpha, \gamma) - \alpha - \max(\alpha, \beta, \gamma) + \max(\alpha, \beta). \end{aligned} \tag{2.17}$$

If $\gamma \leq \alpha$, Equation 2.17 equals $\alpha - \alpha - \max(\alpha, \beta) + \max(\alpha, \beta) = 0$.

If $\gamma > \alpha$, when $\gamma < \beta$, Equation 2.17 equals:

$$\begin{aligned} & \gamma - \alpha - \max(\gamma, \beta) + max(\alpha, \beta) \\ \rightarrow & \gamma - \alpha - \beta + \beta \\ \rightarrow & \gamma - \alpha \\ & > 0. \end{aligned}$$

If $\gamma > \alpha$, when $\gamma > \beta$, Equation 2.17 equals:

$$\begin{aligned} & \gamma - \alpha - \gamma + max(\alpha, \beta) \\ \rightarrow & max(\alpha, \beta) - \alpha \\ & \geq 0. \end{aligned}$$

Thus,

$$\begin{aligned} & \max(\alpha, \gamma) - \alpha - \max(\alpha, \beta, \gamma) + \max(\alpha, \beta) > 0 \\ \rightarrow & \Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) - \Gamma(\mathcal{N}^* \cup n) + \Gamma(\mathcal{N}^*) > 0 \\ \rightarrow & \Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) > \Gamma(\mathcal{N}^* \cup n) - \Gamma(\mathcal{N}^*), \end{aligned}$$

which satisfies Equation 2.13.

□

Next, Proposition 2.2 proves that using RRT-based approach to find sensor deployment locations guarantees a near-optimal result.

Proposition 2.2. *RRT-based sampling algorithm finds near-minimal estimation error, as the error monotonically decreases with the algorithm iterates.*

Proof. Let $\mathcal{N}_i$ be the set of feasible deployment strategy produced in ith iteration, and the corresponding estimation error is σ_i. Similarly, the set of feasible deployment strategy in $i+1$th iteration is $\mathcal{N}_{i+1}$ and the estimation error is σ_{i+1}. Then, denote the new searching

location in $i+1$th iteration as n. According to Equation 2.12, the deployment strategy is greedily selected to be the one with minimal estimation error:

$$\begin{aligned}\sigma_{i+1} &= \min_{\mathcal{N}\subseteq\mathcal{N}_{i+1}} \Gamma(\mathcal{N}) \\ &= \min(\min_{\mathcal{N}\subseteq\mathcal{N}_i} \Gamma(\mathcal{N}), \min_{\mathcal{N}\subseteq\mathcal{N}_{i+1}\setminus\mathcal{N}_i} \Gamma(\mathcal{N})).\end{aligned} \tag{2.18}$$

Then,

$$\sigma_{i+1} = \min(\sigma_i, \min_{\mathcal{N}\subseteq\mathcal{N}_{i+1}\setminus\mathcal{N}_i} \Gamma(\mathcal{N})) \leq \sigma_i. \tag{2.19}$$

Hence, the estimation error will decrease monotonically as the algorithm iterates. Besides, according to the principle of submodularity, the greedily selected deployment strategy is near-optimal. Thus, the result of RRT-based sampling algorithm has the near-minimal estimation error as it converges according to the iteration. □

In this manner, the near-optimal sensor deployment locations can be found. Besides, the linear dependence of the sampling data can be captured and eliminated by detecting their linear correlation while expanding the spanning tree. The algorithm of RRLR Connectivity is presented as Algorithm 2.1.

The inputs of Algorithm 2.1 are the exploration step size ρ, matrices $\mathbf{A}$ and $\mathbf{S}$ that contain the spatiotemporal information, and the geographic EOI ($\mathcal{A}$). Line 1 initializes the algorithm by setting up the starting location of the space-filling tree, where G denotes the space-filling tree generated by the rapid-exploring random tree, and V and E stand for its vertices and edges, respectively. Then, line 2 initializes the estimation error and the sensor deployment set. Also, τ denotes the set of computed deployment locations, which should be empty in the beginning, and η records the infinite horizon cost calculated through prior estimation error on locations in set τ.

Next, the algorithm is executed iteratively to find near-optimal sensor deployment locations. In each iteration, a location v_{rand} will be randomly selected in $\mathcal{A}$. Then, the nearest vertex in the space-filling tree G will be calculated as $v_{nearest}$, and the new sampling loca-

Algorithm 2.1: Rapid-exploring Random tree with Connected Linear Reduction.

Input: ρ, $\mathbf{A}$, $\mathbf{S}$, $\mathcal{A}$
Output: η, τ

1 $V \leftarrow \phi_0$, $E \leftarrow \emptyset$, $G \leftarrow (V, E)$;
2 $\tau \leftarrow \emptyset$, $\eta \leftarrow \infty$;
3 **for** $i = 1$ ***to*** T ***do*** **do**
4 v_{rand} = RAND_NODE($\mathcal{A}$);
5 $v_{nearest}$ = NEARVERTEX(v_{rand}, G);
6 v_{new} = STEP($v_{nearest}$, v_{rand}, ρ);
7 $\hat{\mathbb{N}}$ = GenPath(G, v_{new}, $v_{nearest}$);
8 $\tilde{\eta} = \min_{\tilde{\mathcal{N}} \in \hat{\mathbb{N}}} \Gamma(\tilde{\mathcal{N}})$;
9 $\tilde{\tau} = \arg\min_{\tilde{\mathcal{N}} \in \hat{\mathbb{N}}} \Gamma(\tilde{\mathcal{N}})$;
10 **if** $\tilde{\eta} < \eta$ **then**
11 $\eta = \tilde{\eta}$;
12 $\mathbf{M}$ = genSample($\mathbf{A}, \mathbf{S}, \tilde{\tau}, i, M$);
13 $\mathbf{M}^*, \mathbf{I}^*$ = LinearDependentDetection($\mathbf{M}, \epsilon$);
14 G, τ = RedundancyElimination($\mathbf{M}^*, \mathbf{I}^*, G, \tilde{\tau}, d_{com}$);
15 **end**
16 $V \leftarrow V \cup v_{new}$;
17 $E \leftarrow E \cup edge(v_{nearest}, v_{new})$;
18 **end**

tion v_{new} is selected by moving from $v_{nearest}$ in the direction of v_{rand} at most ρ. Next, line 7 finds spanning trees $\hat{\mathbb{N}}$ of G that pass through the vertices v_{new} and $v_{nearest}$. $\hat{\mathbb{N}}$ indicates the possible sensor deployment strategies. The strategy with minimal estimation error within set $\hat{\mathbb{N}}$ is calculated in line 8 to line 9. Therefore, as shown in line 10 to line 15, if a sensor deployment strategy that has the lowest estimation error is found, it will be recorded and checked for sampling location redundancy.

Line 12 generates sampling data based on the environment model and the sampling location. Then, the data matrix $\mathbf{M}$ is passed to Algorithm 2.2 to detect linearly correlated neighbors, and the results of Algorithm 2.2 are used to eliminate redundant sampling locations with regard to network connectivity according to Algorithm 2.4.

Last, the new vertex and its edge are added to the space-filling tree for next iterations. RRLR Connectivity runs on a local or cloud server, which can communicate with each

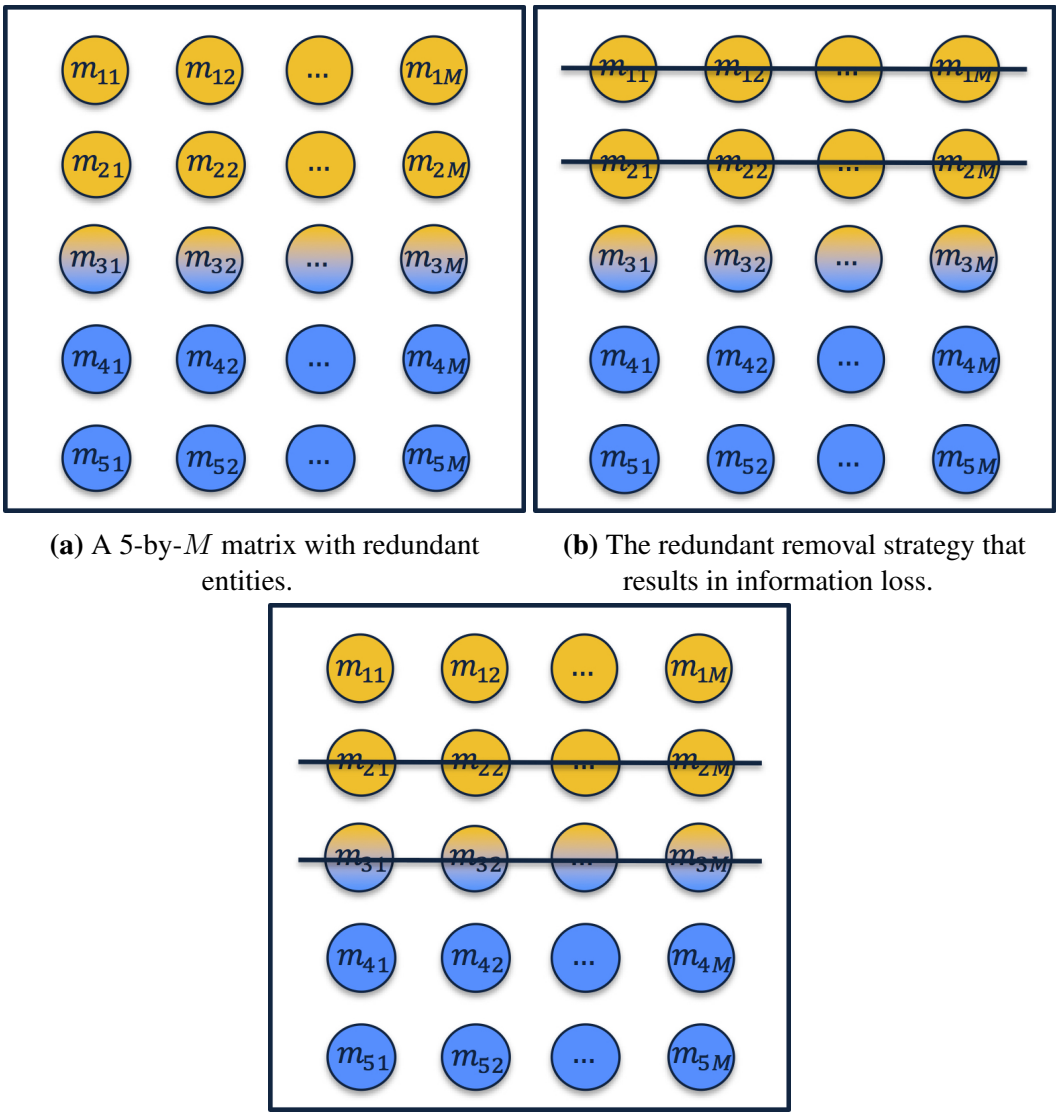
mobile sensor node. Currently, it is assumed that no communication is needed among the sensor nodes. As a result, information will be processed by the local server. Instructions on the sampling locations are sent to the mobile sensor nodes after computation.

2.3.2 Linearly Dependent Sampling Location Elimination with regard to Network Connectivity

Usually, multiple homogeneous sensor nodes will be deployed in the spatiotemporal field. Each of them will generate a time series of data for each type of sensor. Hence, sampling data collected in different locations should be checked to examine the redundancy of the deployed locations. If linearly correlated time series exist, the matrix of data will not be of full rank, and the dimension of the data matrix can be reduced to its rank by eliminating the redundant rows. Accordingly, it is not necessary to deploy sensor nodes in all required locations because the information provided by these redundant sensor nodes can be reconstructed [88, 89]. However, arbitrary removal of redundant sampling data according to the rank of the data matrix might result in information loss. Figure 2.3 illustrates a case for preserving information of the sampled data while removing redundant sampling locations.

Figure 2.3 (a) presents a 5-by-M matrix with redundant entities, whose rank is 3. Therefore, there exist two rows of the matrix that can be removed while not resulting in information loss. Assuming that the rows of the matrix that have the same color are linearly correlated, each row can be reconstructed by the rest of the rows with the same color according to Equation 2.10. Note that the nodes that have gradient color from yellow to blue are linearly correlated with rows with both colors. In this manner, the rows with gradient color contain the most duplicated information, and removing them may not result in information loss since they have a higher chance to be reconstructed.

Figure 2.3 (b) presents a case where randomly removing redundant rows of the data matrix may result in information loss. If the first row and the second row of the matrix are removed due to the rank of the matrix, the information will be lost as the removed rows



(a) A 5-by-M matrix with redundant entities.

(b) The redundant removal strategy that results in information loss.

(c) The removed entities can be reconstructed.

Figure 2.3: Illustration of redundant sensor elimination.

cannot be reconstructed from the remaining rows. Therefore, a possible redundant removal strategy is to remove the rows with the highest redundancy first. As shown in Figure 2.3 (c), removing the second row and the third row of the matrix does not result in information loss.

During the process of removing redundant rows in the data matrix, the sensor network connectivity should also be considered. Removing all the redundant sensor nodes might break the connectivity of the wireless sensor network. Thus, the constraint concerning the

wireless network connectivity is added:

$$\forall n_i \in \mathcal{N}, \exists n_j \in \mathcal{N} : \text{dist}(n_i, n_j) \leq d_{com}, \tag{2.20}$$

where n_i and n_j are two locations belonging to the deployment set $\mathcal{N}$, and d_{com} is the threshold for the largest communication distance. If there exists at least one location whose nearest neighbor is further than τ_{com}, the connectivity of the network is broken.

Hence, the redundant removal strategy of RRLR Connectivity is proposed to detect and eliminate redundant sensor nodes with regard to information loss and network connectivity. This method first identifies linearly correlated neighbors and then removes the redundant sampling locations. As a precursor to the algorithm, a way to detect linearly correlated sampling data for a given matrix $\mathbf{M}$ is presented as Algorithm 2.2.

Algorithm 2.2: Linearly Correlated Neighbor Detection.

Input: $\mathbf{M}, \epsilon$
Output: $\mathbf{M}^*, \mathbf{I}^*$
1 $M, k \leftarrow \text{size}(\mathbf{M})$;
2 $r \leftarrow \text{rank}(\mathbf{M})$;
3 $\mathbf{I} = \text{eye}(M)$;
4 $\mathbf{M}^*, \mathbf{I}^* = \text{GaussianElimination}(\mathbf{M}, \mathbf{I}, \epsilon, M, k, r)$;

As shown in Algorithm 2.2, the inputs are the data matrix $\mathbf{M}$ and the noise threshold ϵ. Line 1 calculates the size of the input data matrix, and line 2 calculates its rank. Then, line 3 builds an identity matrix at the size of sampling locations. Identity matrix records the row addition actions of the Gaussian elimination and contains the information of linearly correlated sets. The use of the identity matrix ensures that the ith row in $\mathbf{I}$ represents the same row in matrix $\mathbf{M}$. By carrying out the same row addition action (i.e., add the same row with the same coefficient) as in matrix $\mathbf{M}$, $\mathbf{I}$ can store the information about how row addition is done in the matrix $\mathbf{M}$. Then, data matrix, identity matrix, noise threshold, matrix size, and the rank of the data matrix are passed to Algorithm 2.3 to detect linearly correlated neighbors.

Algorithm 2.3: Gaussian Elimination.

Input: $\mathbf{M}, \mathbf{I}, \epsilon, M, k, r$
Output: $\mathbf{M}^*, \mathbf{I}^*$

```
1  i =1, j = 1;
2  while i ≤ M do
3  |   i_max = arg max_{i=j,···,M} abs(M[i, j]);
4  |   if M[i_max, j] ≤ ε then
5  |   |   j++;
6  |   else
7  |   |   pivot = M[i_max, j];
8  |   |   swap(M[i, ·], M[i_max, ·]);
9  |   |   swap(I[i, ·], I[i_max, ·]);
10 |   |   M[i : M, j : k] −= M[i : M, j] × J_{1,k}/pivot;
11 |   |   I[i : M, j : k] −= I[i : M, j] × J_{1,k}/pivot;
12 |   |   i++, j++;
13 end
14 M* = M, I* = I;
```

Line 1 of Algorithm 2.3 initializes the index of the data matrix. Then, line 3 finds the pivot of the ith row. This step improves the stability of the algorithm and avoids adding the same row multiple times. Then, the effect of noise is considered, and a threshold ϵ is utilized to remove such effect. Line 4 checks whether the value of the pivot is less than ϵ for the current iteration. If so, the current iteration should be skipped because it is considered as a zero row.

Line 7 to Line 12 carry out row addition for each nonzero row. Line 7 records the value of the pivot of this iteration. Then, line 10 carries out row addition for removing row redundancy, where $\mathbf{J}$ is a 1-by-k matrix of ones. At the same time, line 11 records the row addition action in identity matrix $\mathbf{I}$. When the iteration ends, there should be zero rows in the output matrix $\mathbf{M}^*$ if the input matrix $\mathbf{M}$ has row redundancy, and the number of zero rows plus the rank of the input matrix $\mathbf{M}$ should be equal to the row dimension M. Moreover, the input identity matrix $\mathbf{I}$ records the information of how row addition is operated in the matrix $\mathbf{M}$, and such information is stored in the output matrix $\mathbf{I}^*$.

Removing one redundant row found by Algorithm 2.3 will not result in information loss

since it can be linearly represented by other rows. However, randomly removing multiple rows in the data matrix may result in information loss, as shown in Figure 2.3.

Once the linearly correlated rows that stand for the corresponding redundant nodes have been detected, they can be relocated to other uncovered places or completely removed, to reduce the cost of resources. The fundamental requirements in the removal of redundant rows from a correlating cluster are: 1. not breaking the correlation of any clusters, and 2. not resulting in information loss or broken network connectivity. Typically, different clusters might share the same redundant nodes, and there remains uncertainty about whether removing such nodes will break the linear correlation in other clusters. Besides, as there might be multiple redundant nodes in all sampling nodes, randomly removing all of them might result in information loss if the sampling data from the removed nodes cannot be reconstructed.

To this end, redundant sensor readings are found by utilizing the output matrices $\mathbf{M}^*, \mathbf{I}^*$ through Algorithm 2.2. The row addition information about the Gaussian elimination of data matrix $\mathbf{M}$ is stored in $\mathbf{I}^*$. Zero rows of the data matrix are the redundant ones, whose row addition information is stored in the corresponding rows of the matrix $\mathbf{I}^*$, and the column indices of nonzero elements of $\mathbf{I}^*$ represent the linearly correlated rows in matrix $\mathbf{M}$. Therefore, linearly correlated sampling data can be recovered from $\mathbf{I}^*$.

Algorithm 2.4 aims to eliminate redundant rows and the deployment of the corresponding sensor nodes. This algorithm reduces the size of the deployment set by discovering the linear redundancy among the collected sampling data in different sensor nodes, based on the outputs of Algorithm 2.2. Moreover, the network connectivity is preserved during the elimination of redundant deployment locations.

Algorithm 2.4 first finds the set that stores the redundant sampling locations from line 2 to line 4. All rows (sensor nodes) within the linearly correlated set are picked into the redundant sampling location $\mathcal{J}$. Then, only $|\mathbf{M}^*| - \text{rank}(\mathbf{M}^*)$ sensor nodes in the set $\mathcal{J}$ should be selected as shown in line 7. Besides, as there are multiple options for removing

Algorithm 2.4: Redundancy elimination w.r.t. connectivity.

Input: $\mathrm{M}^*, \mathbf{I}^*, G, \tau, d_{com}$
Output: G, τ

1 $\mathcal{J} \leftarrow \emptyset$;
2 **foreach** $\max(\mathbf{M}^*_{\cdot i}) \leq \epsilon$ *and* $\mathbf{I}^*_{ij} \geq \epsilon$ **do**
3 $\quad \mathcal{J} \leftarrow \mathcal{J} \cup j$;
4 **end**
5 $J = |\mathcal{J}|$;
6 **foreach** $j \in \mathcal{J}$ **do**
7 $\quad$ **while** $|\mathcal{J}| \geq J + r - M$ **do**
8 $\quad\quad$ **if** *min-dist*$(\tau(j), \tau \setminus \tau(j)) < d_{com}$ ***and*** $|\tau| < |\mathrm{M}^*| - $*rank*$(\mathrm{M}^*)$ **then**
9 $\quad\quad\quad \tau \setminus \tau(j)$;
10 $\quad\quad\quad G(\tau(j)) = \emptyset$;
11 $\quad\quad\quad \mathcal{J} \setminus j$;
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 **end**

the redundant sensor nodes to reduce the size of matrix M to its rank, network connectivity should be considered during this process. As line 8 shows, if the minimal distance between the redundant deployment location and the entire deployment strategy $\tau \setminus \tau(j)$, which is calculated by min-dist function, is less than the threshold, this redundant deployment can be removed. In this manner, the network connectivity will not break when removing the redundant sensor nodes.

2.4 Simulation Results

Numerical simulations are conducted using the dataset that was generated from a computational fluid dynamics (CFD) model to simulate a 2-dimensional spatiotemporal field based on COMSOL software [90]. Real-world experiments usually have restrictions on the resources, which calls for the deployment of a limited number of sensor nodes in the field. CFD simulations avoids this problem by generating an unlimited number of sensing locations in the field. The test data generated by COMSOL software contains a simulation of three heat sources in a squared aquatic field. It also considers the effects of diffusion,

horizontal heat transfers by small flows and the influence of obstacles. In the present work, a simulation generates a time series for 50 random sampling locations with noise in a 50-meter by 50-meter square field. Figure 2.4 illustrates the sensor deployments. Orange squares denote sensor node deployment locations, and the blue area indicates the aquatic environment.

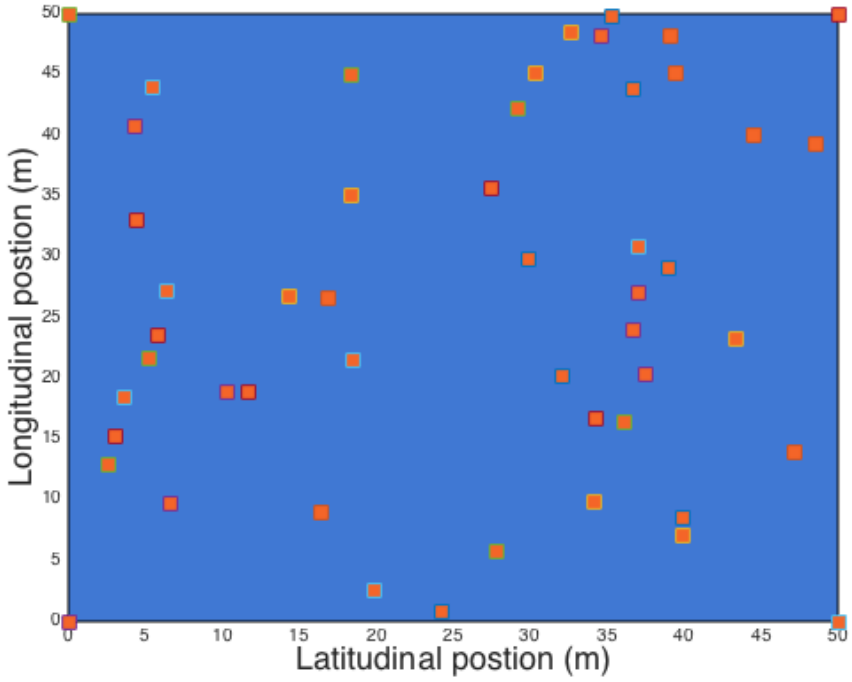


Figure 2.4: Deployment locations in CFD simulation.

The field estimation uncertainty is calculated to validate the effectiveness of the generated near-optimal deployment locations. Then, a color map is drawn to indicate the estimation uncertainties according to the given deployment locations. High uncertain area has bright color, and lower ones is shown in a dark color. The estimation uncertainty is calculated for each point ψ over the field $\mathcal{A}$, which is denoted by its variance $V(\psi), \psi \in \mathcal{A}$. The

variance at location ψ can be calculated as:

$$\begin{aligned} V(\psi) &= E[(\mathbf{S}_\psi(\mathbf{x}[t] - \hat{\mathbf{x}}[t])) \cdot (\mathbf{S}_\psi(\mathbf{x}[t] - \hat{\mathbf{x}}[t]))^T] \\ &= tr(\mathbf{P}[t]\mathbf{S}_\psi^T\mathbf{S}_\psi), \end{aligned} \tag{2.21}$$

where tr function calculates the trace of the matrix.

Before applying RRLR Connectivity to these datasets, the parameters should be tuned to satisfy the environment of the sampling area. Matrices $\mathbf{A}, \mathbf{S}, \mathbf{R}_w$ and $\mathbf{R}_v$, are studied from the data sampled from five sampling locations. In this case, the number of basis functions is set to five. The parameter values for $s_i(\psi) = Ke^{-\frac{||(\psi_t^i)-(\psi_j)||^2}{2\sigma^2}}$ are chosen as: σ^2 = square sum of the basis function variance, and K = average of the entire vector. The number of iterations is set to 2000 and the step length is 10 meters.

The comparison algorithms, RRC and RRCstar are executed using the same parameter values. Besides this, two algorithms, random search (RS) and greedy algorithm, are executed as benchmark problems with the same common parameters. RS moves around randomly from the initial point to different sampling locations. Then, the deployment with minimal estimation error is computed from all locations visited in the field. Greedy Algorithm is based on RS, but it does not take new observations randomly. It takes four (GD4) or eight (GD8) random attempts in the current location. Each attempt will generate a new observation location near the current location of the sensing agent. Then, together with the existing observations, the action leading to lower estimation error will be picked and executed as the next step. Also, a version of RRLR Connectivity that did not consider the network connectivity, which is termed as RRLR, is added for benchmark. All experiments were conducted on a PC with 4.0 GHz Quad-Core CPU and 16 G memory.

Figure 2.5 presents the simulation results generated from the CFD dataset using RRT-based algorithms. RRC requires 10 sensor nodes to monitor the selected environment persistently while RRC needs 12, as indicated in Figure 2.5 (a)-(b). In contrast, RRLR and RRLR Connectivity reduce 9 and 4 redundant deployment locations, respectively, with the

help of Linear Reduction. They only require 12 and 6 sensor nodes in the field, respectively. Comparing Figure 2.5 (a)-(d), RRLR-based algorithms remove several closely deployed sensor nodes according to their linear redundancy in the sampling data. This process not only reduces the number of sensors in the field but also helps the system to explore more area in the field with limited resources. Besides, by eliminating redundant deployments, RRLR and RRLR Connectivity explore a more deployable area (i.e., the bottom left area in Figure 2.5 (c)-(d)), which will reduce the field estimation uncertainty, as indicated in Figure 2.7.

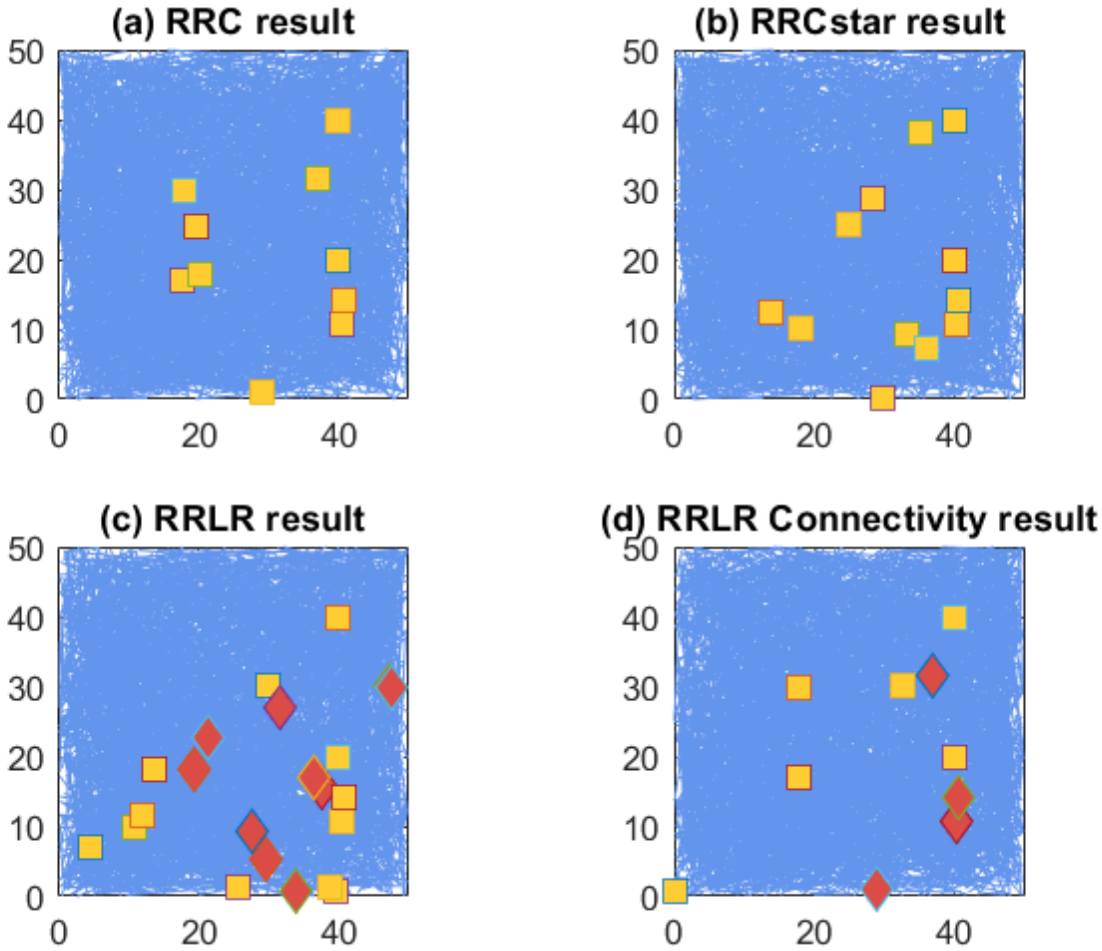


Figure 2.5: Deployment results generated by RRT-based algorithms.

Figure 2.6 (a) - (c) present the results generated by the random search and the greedy algorithm, where too many sensor nodes are stuck in a narrow area of local minima and cannot escape. Especially, RS is stuck around its initial search point as it picks the next sampling location randomly. Although the greedy-based approach has good coverage on exploring the EOI, the results easily entrap in local minima. As indicated in the figure, the sampling nodes are narrowly deployed in a small area, which will result in a considerable loss.

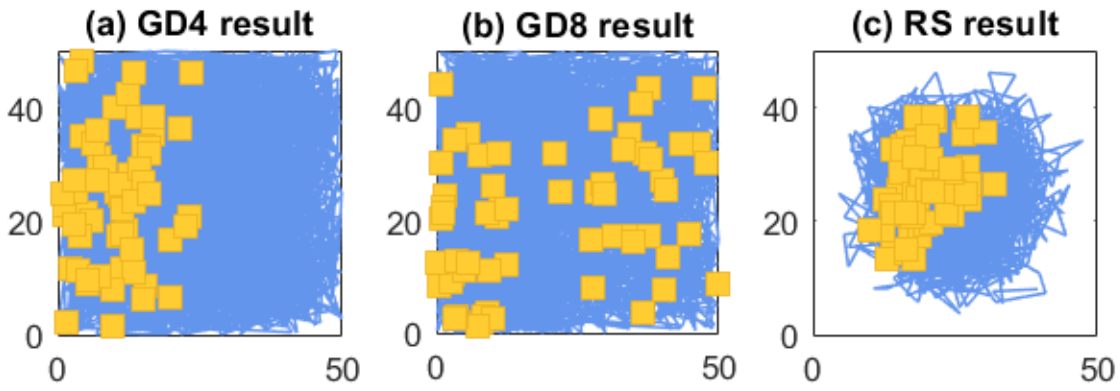


Figure 2.6: Deployment results generated by random search-based algorithms.

Figure 2.7 presents the field estimation uncertainty using the results generated through the CFD dataset. The results show that all four RRT-based algorithms generate similar uncertain regions in the EOI since all of them are near-optimal algorithms. RRLR-based algorithms generate better results by placing the sensors close to the bottom left area of the EOI, which results in lower estimation uncertainty in that area. The statistical measurements of the estimation uncertainty and the required number of sensor nodes are shown in Table 2.1. The results show that the RRLR-based algorithms achieve low uncertainty over the field with a small number of sensors. Especially, RRLR Connectivity finds the best deployment strategy with low uncertainty and variance while using minimal sensor nodes.

Figure 2.8 shows the estimation error from different algorithms over the iterations. The

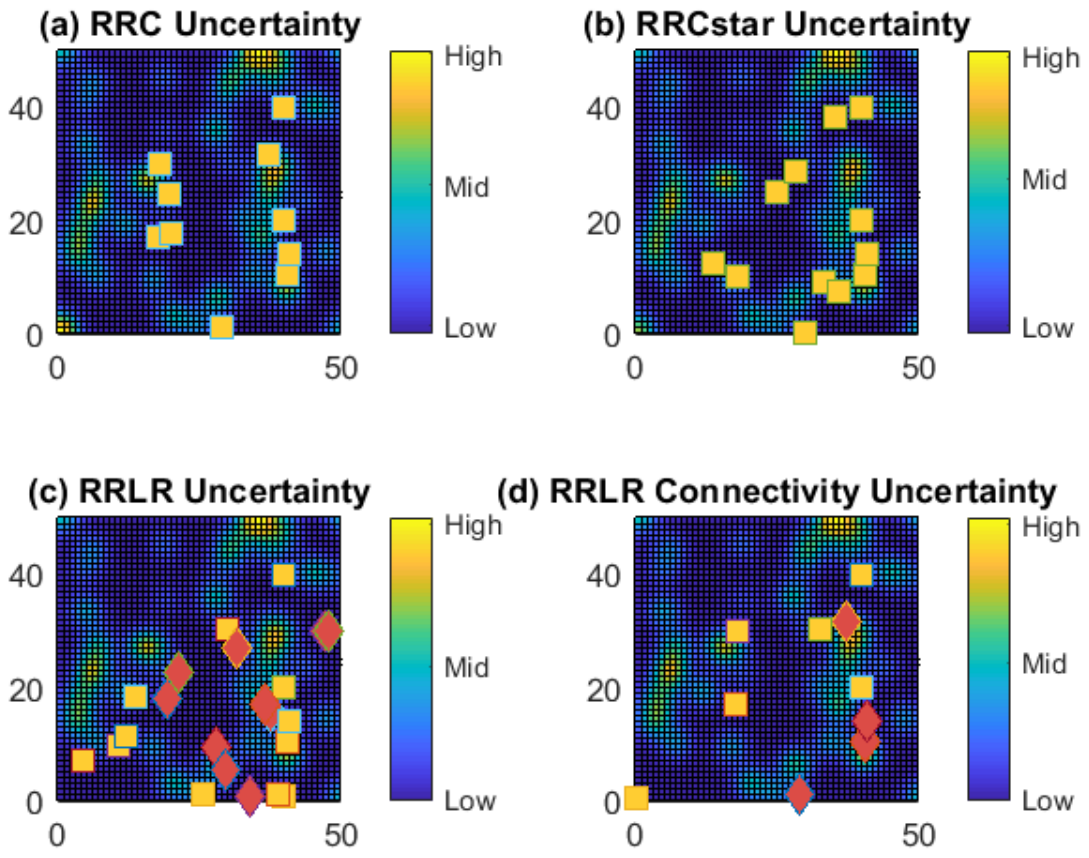


Figure 2.7: Field estimation uncertainty results of RRT-based algorithms.

Table 2.1: Statistical measurement of the field estimation uncertainty.

	Max	Min	Variance	# sensors
RRC	3.74	$3.05 * 10^{-5}$	0.14	10
RRCstar	3.37	$3.11 * 10^{-5}$	0.16	12
RRLR	3.37	$1.80 * 10^{-5}$	0.16	12
RRLR Connectivity	3.48	$1.82 * 10^{-5}$	0.17	6

randomized algorithm does not perform well and can hardly reduce the estimation error. However, all RRT-based algorithms reduce the estimation error significantly and also converge. Hence, according to Proposition 2.2, all the RRT-based algorithms achieve a near-optimal solution with the CFD dataset. Moreover, as indicated in Table 2.1, RRLR Connectivity uses least sensor nodes to perform the persistent monitoring with minimal field estimation uncertainty. RRLR Connectivity only requires six sensor nodes in the field, while all other benchmark algorithms need more than ten. Thus, it can be concluded that the proposed algorithm achieves near-optimal solution while significantly reducing the number of sensor nodes in the field.

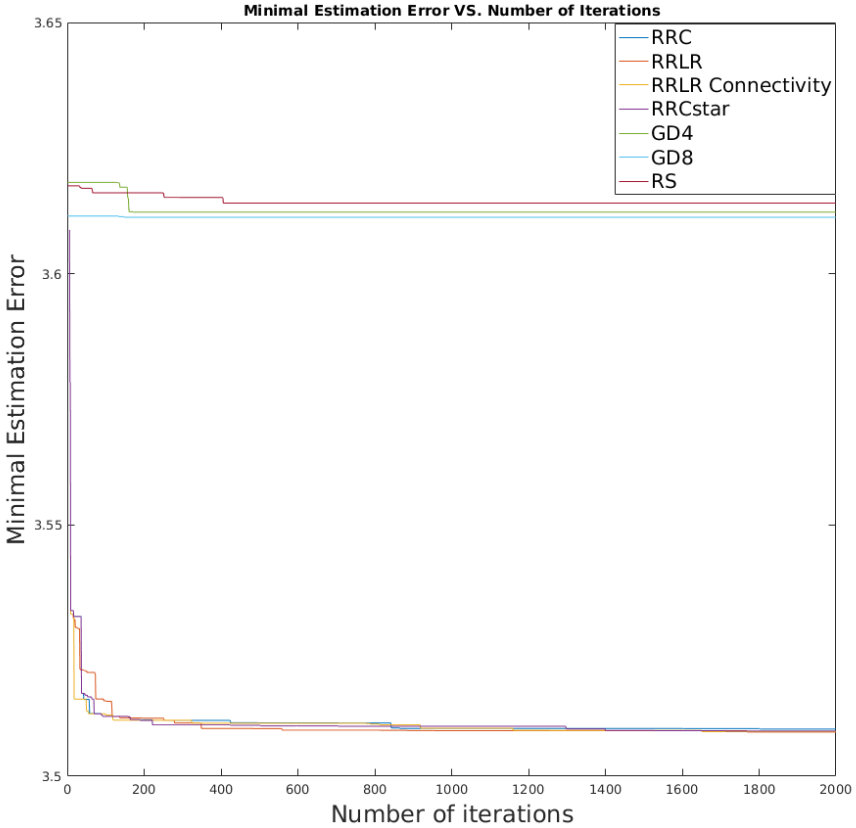


Figure 2.8: Change of minimum estimation error.

In conclusion, RRLR Connectivity requires fewer sensor nodes to persistently monitor the environment over the EOI while receiving the lowest estimation error and field estimation uncertainty. This significant improvement results from the elimination of redundant

sampling data during the tree expansion. RRLR Connectivity can also realize the near-optimal sensor deployment strategy while preserving the network connectivity.

Another test based on in situ results is provided in Chapter 6.

2.5 Summary

In this chapter, a novel strategy was proposed to find a near-optimal solution for determining the sensor node deployment locations in a spatiotemporal field. The proposed method aimed to reduce the redundancy in sampled data while optimizing the estimation over an infinite horizon and preserving the network connectivity. It utilized row redundancy in the sampling data matrix to eliminate unnecessary rows (sensor nodes) and the structure of the space-filling tree in the field to determine the suitable deployment locations. Also, the redundant sampling locations were eliminated with regard to the network connectivity. Simulation results showed that the number of sensor nodes needed in the field was significantly reduced while achieving maximum prior estimation error and minimal field estimation uncertainty.

Chapter 3

Deep Reinforced Learning Tree for Efficient Spatiotemporal Exploration

3.1 Introduction

As introduced in Chapter 2, sampling location optimization is an essential part of spatiotemporal monitoring, and the sampling redundancy can be detected and eliminated through that. However, RRT-based spatiotemporal field exploration schemes are inefficient due to random sampling. Therefore, this chapter seeks to improve the efficiency of spatiotemporal field exploration.

Considerable research effort has gone into the monitoring of environmental processes that occur in the ecosystems. Such work can facilitate the generation of early warnings about events that can adversely affect the stability of an Environment of Interest (EOI) [91]. It will also provide adequate time for responding and minimizing possible negative impacts on the environment, and taking proper corrective actions.

Typically, data sampling from a large EOI can be quite laborious. For example, when monitoring an aquatic environment, sensors are primarily used to sample real-time data of some critical parameters (e.g., temperature, conductivity) of the water [92]. For this, oceanographers and limnologists commonly rely on the manual sensor deployment at fixed

locations from boat [93]. Clearly, searching for suitable monitoring locations and conducting persistent monitoring using robotic sensors is a better approach.

Generally, the environmental features of an EOI are spatiotemporally correlated. Thus, persistent monitoring using sensor networks or robotic sensors is required to ensure the observation quality while employing a relatively small number of sensor nodes in the field [94]. Common strategies that can provide field coverage for persistent monitoring include geometry coverage using disk models, and by placing robotic sensors in the most informative locations of the EOI.

The traditional approach of field monitoring utilizes a static sensor network, which is not accurate and efficient. This approach first densely deploys sensor nodes in the EOI. Then, redundant sensor nodes are removed according to the spatiotemporal correlation of the sensor node location and readings [1, 5, 95–97]. Such methods typically require a dense deployment of sensor nodes in the initial stage and then turning off the deployed sensors selectively, which is costly not adequately accurate. For better performance, mobile sensor nodes should be employed.

Researchers considered as well taking critical observations from the field according to information theory rather than taking as many samples as possible. Guestrin et al., Krause et al. and Singh et al. proposed mutual information-based methods that produce a near-optimal informative sampling strategy [10, 23, 87]. They tested their methods using real-world datasets and in situ tests and determined that mutual information could acquire more information from the field while achieving a lower root mean square error (RMSE) for the environment estimation. Du et al. proposed an optimal sensor placement strategy for monitoring hydrodynamics and water quality in an urban district, using entropy and mutual information to gather most information in selected locations [24, 25]. Nguyen et al. proposed an efficient spatial sensor selection method using Mutual Information and Gaussian Markov random fields [98]. Ma et al. developed a data-driven learning method and planning metrics for environmental sampling [99]. However, approaches that are based on traditional

information theory only focus on choosing a subset that is most informative from the perspective of pre-determined observation locations. Although these approaches find sampling locations in a near-optimal way, they may not be suitable for infinite horizon problems.

Researchers have recognized that mobile sensor nodes can search over an infinite horizon with improved performance and efficiency in persistent field monitoring. Typically, field exploration methods like RRT are used to search over an infinite horizon. However, RRT-based approaches pick sampling locations randomly, and do not utilize the sampled information, which causes low efficiency. RL is utilized to accelerate the sampling scheme [13]. When the learning agents in the RL-based algorithms face complex real-world situations, it is hard for them to derive environmental features efficiently through high-dimensional sensory inputs. Deep learning may be utilized to help RL understand a complex environment [42].

In the present dissertation, the EOI is modeled as a Gaussian process (GP), and the coverage of the field can be provided by searching for the most informative sampling locations [10, 100]. The objective of the informative planning method that is developed in this dissertation is to determine a set of sampling locations that minimize the field estimation error over the EOI. This objective can be achieved by taking samples from the EOI and identifying the most informative subset of sampling locations, using field exploration methods like RRT. Then, as the sampling data from different locations are not equally important, not all sampling locations that are selected by RRT need to be visited. Therefore, a deep reinforcement learning (DRL) approach is utilized in the present work to study the effectiveness of the sampled locations and to avoid unnecessary sampling locations. The process of searching over the spatiotemporal EOI can be accelerated in this manner and the most informative sampling locations can found efficiently.

Also, a novel field exploration method is developed in the dissertation based on the developed DRL model. The developed algorithm is called Deep Reinforced exploring Learning Tree (DRLT), which can efficiently find near-optimal sampling locations for a set of

mobile robotic sensors in an EOI. The robotic sensors are assumed to be point sensors, which only capture information at their location, in the presence of Gaussian white noise. Moreover, the placement quality of the robotic sensors is determined by the field estimation error as calculated by a Kalman filter [3], and the field coverage is evaluated by estimating the monitoring uncertainty in the field. Usually, finding an optimal solution over an infinite horizon is intractable, and it has shown to be NP-hard. However, DRLT can produce a series of sampling locations with monotonically decreasing cost, which is proved to be near-optimal with minimum estimation error of the EOI.

3.2 Problem Definition and Mathematical Background

3.2.1 Modeling of Environment

In this chapter, the primary goal is to accelerate the field exploration scheme that was introduced in Chapter 2, by learning from the sampled data. The environment modeling approach that is used in the present chapter is the same as in Chapter 2, which is a common environmental model used for geological and environmental systems. A discrete time-variant scalar field $v_\psi[t] \subset \mathbb{R}$ is used to present the spatiotemporal field, where $\psi \in \mathcal{A}$ is an arbitrary point in a two-dimensional (2-D) EOI $\mathcal{A}$, which can be non-convex and filled with obstacles. The model can be expressed as a function of the temporal information matrix ($\mathbf{A}$), the spatial information matrix ($\mathbf{S}_\psi$), location in the field ψ and time t:

$$\begin{aligned} v_\psi[t] &= \mathbf{S}_\psi \mathbf{x}[t] + v[t], \\ \mathbf{x}[t+1] &= \mathbf{A}\mathbf{x}[t] + w[t]. \end{aligned} \tag{3.1}$$

The temporal information matrix $\mathbf{A}$ is used to calculate the time-varying coefficient vector $\mathbf{x}[t] = [x_{t_1} \cdots x_{t_k}]^T$. $\mathbf{x}[t]$, which is considered as the state vector of a linear discrete time-invariant system. The spatial information matrix $\mathbf{S}_\psi$ is the row vector consisting of k spatial basis functions $\mathbf{S}_\psi = [s_1(\psi), \cdots, s_k(\psi)]$.

3.2.2 Sampling Quality of Deployment Location

Estimation Error

In the present chapter, a linear Kalman filter is also used for estimating the system output $v_\psi[t]$, as in Chapter 2. Linear Kalman filter is an optimal estimator for a linear discrete time-invariant system[86, 101]. Thus, the prior estimation error calculated by the Kalman filter is used as a criterion to indicate the estimation quality over the field [28]. The prior estimation of the linear Kalman filter is used for updating the error covariance matrix [101].

As shown in Chapter 2, the largest eigenvalue in the estimation error matrix is regarded as the cost of the deployment strategy:

$$\Gamma(\mathcal{N}) = \pi(\mathbf{P}_\infty^{\mathcal{N}}) = \max_{i=1,2,\ldots} \pi(\mathbf{P}_\infty^{i}).$$

Here $\pi(\mathbf{P}_\infty^{\mathcal{N}})$ denotes the largest eigenvalue of the prior error covariance matrix for strategy $\mathcal{N}$ as time goes to infinity. The efficient computation of the prior estimation error follows the approach given in [3].

Accordingly, the near-optimal solution is achieved by finding the deployment strategies of minimal cost:

$$\begin{aligned}\mathcal{N}_{opt} &= \arg\min_{\mathcal{N}} \Gamma(\mathcal{N}) = \arg\min_{\mathcal{N}} \pi(\mathbf{P}_\infty^{\mathcal{N}}) \\ &= \arg\min_{\mathcal{N}} (\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^{i})).\end{aligned}$$

Information Gain of New Sampling Locations

In the present work, information theory provides the basis for learning tree expansion. It depends on a location with higher entropy or mutual information that will collect more information from the field. Then, the sampling location has higher information gain and gives a higher opportunity to reduce the field estimation error. Assume that $\mathcal{A}$ is a field to

be monitored and all possible observations compose the set $\mathcal{A}$. Let $\mathcal{N} \subset \mathcal{A}$ means a subset in this field, which contains information about the deployment locations.

In the set of selected locations in the EOI, $\mathcal{N}_i = \{n_1, n_2, \cdots n_i\} \subset \mathcal{A}$, the chain rule of entropy gives:

$$\mathcal{H}(\mathcal{N}_i) = \mathcal{H}(n_i|\mathcal{N}_{i-1}) + \cdots + \mathcal{H}(n_2|\mathcal{N}_1) + \mathcal{H}(n_1|\mathcal{N}_0). \tag{3.2}$$

Based on entropy, a subset is found that maximizes the mutual information (MI) between the observed locations ($\mathcal{N}$) and the rest of the space ($\mathcal{A} \setminus \mathcal{N}$). MI can be expressed as a function of entropy:

$$\begin{aligned} \mathcal{MI}(\mathcal{N}) = \mathcal{MI}(\mathcal{N}; \mathcal{A} \setminus \mathcal{N}) &= \mathcal{H}(\mathcal{N}) - \mathcal{H}(\mathcal{N}|\mathcal{A} \setminus \mathcal{N}) \\ &= \mathcal{H}(\mathcal{N}) - \mathcal{H}(\mathcal{A}) + \mathcal{H}(\mathcal{A} \setminus \mathcal{N}), \end{aligned} \tag{3.3}$$

where $\mathcal{H}(\mathcal{A} \setminus \mathcal{N})$ is the entropy of unobserved locations in the field, and $\mathcal{H}(\mathcal{A} \setminus \mathcal{N}|\mathcal{N})$ is the conditional entropy of unobserved location set $\mathcal{A} \setminus \mathcal{N}$ given observations at $\mathcal{N}$.

Deep Reinforcement Learning

Unlike supervised learning, reinforcement learning (RL) does not require a dataset to train; it learns through interaction with the environment [102, 103]. In this work, the learning process of DRL network takes place while analyzing the information gain of the new observations in the field. Higher information gain encourages the DRL to take more observations and not get trapped at locations, which helps in reducing the estimation error. Thus, the goal of the proposed method is to select actions that maximize cumulative future information gain.

To model a RL problem, five key attributes should be defined: states $n \in \mathcal{A}$, action set $a \in \mathbb{A}$, transition probability between states $T(n, a, n')$, policy $P(n)$, reward r and value function $Q^{P(\cdot)}(n, a)$. The action set contains information about the actions can be taken in

different states; e.g., move forward or make a turn. Thus, each state n corresponds to an action set $\mathbb{A}$ and actions will make the robot move from state n to n'. In the present work, the transition between states is deterministic; hence, $T(n, a, n') = 1$. To select suitable actions in the current state, policy $P(n)$ should be defined, which regulates the rule of searching over the field and determines all possible next states that can be reached by the given actions $a_t \in \mathbb{A}_{n_t}$ in its current state s_t. Then, the reward r of taking action a at state n can be calculated. Rewards provide a form of evaluation for possible actions from a particular state, and they are bounded. A value function is introduced to quantify the accumulated reward based on the selected action in the current state. It can be represented using Bellman equations for a given policy $P(\cdot)$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \tag{3.4}$$

Here γ is the discount factor for future rewards and α is the learning rate. Also, the maximization function ensures that actions with best future rewards will be selected in each iteration.

Based on the mentioned theory, the process of RL is shown as follows. First, initial state s_0 is selected. Second, an action a_0 is selected according to the rewards, which leads to some state s_1. Third, by executing these steps iteratively, RL will generate a sequence including states and action:

$$s_0, a_0, s_1, a_1, \cdots, s_t, a_t.$$

The accumulated rewards will converge to a maximum value and lead to a steady state [104] [105].

3.2.3 Field Coverage and Network Connectivity Analysis

Apart from achieving the lowest field estimation error, it is also essential to ensure that the deployment strategy has good coverage of the EOI. Good coverage of EOI avoids

densely sampling on a narrow area with high variance and provides better recognition of the field. In the present work, point sensors are carried by the robotic platforms to monitor a large field. Hence, the traditional disk model [106] for active sensors, like radar or sonar, cannot be used, and the coverage of the field is determined by taking into account estimation uncertainty. The field estimation uncertainty is determined by the variance of the observation in each point of the field, given that the robotic sensor is deployed according to the deployment strategy. The variance of a location in the EOI, $Var(v_\psi)$, according to Equation 2.2 and [3], is given by:

$$\begin{aligned} Var(v_\psi) &= E[(\mathbf{S}_\psi(\mathbf{x}[t] - \hat{\mathbf{x}}[t])) \cdot (\mathbf{S}_\psi(\mathbf{x}[t] - \hat{\mathbf{x}}[t]))^T] \\ &= tr(\mathbf{P}[t]\mathbf{S}_\psi^T\mathbf{S}_\psi). \end{aligned}$$

Then the pixel-wise uncertainty of the EOI is calculated based on the corresponding covariance matrix $\mathbf{P}[t]$ and the spatial correlation matrix $\mathbf{S}_\psi$.

Robotic sensors establish a WSN network, which requires them to communicate with each other. It is assumed that the network connectivity is based only on the positions of the robotic sensors and does not depend on the node size. Also, it is assumed that the locations are known, bandwidth is adequate, and all the robotic sensors are homogeneous. Constraints ensure that the deployed sensors and the sinks form a connected wireless sensor network is as:

$$\forall n_i \in \mathcal{N}, \exists n_j \in \mathcal{N} : ||n_i - n_j|| \leq \tau_{com}.$$

Since the informative deployment strategy is generated from the exploring tree, the parent node n_p of n_i will be no farther than the exploring step size τ_{step}. Hence, letting $\tau_{step} \leq \tau_{com}$, there exists at least one node n_p in $\mathcal{N}$ within maximum communication distance of each robotic sensor n_i. This ensures the network connectivity.

3.3 Deep Reinforced Learning Tree

In this section, a sampling-based exploring tree is presented to find the near-optimal solution for informative sampling over an infinite horizon. Unlike the traditional RRT-based algorithms, in the present approach, the sampling locations are not chosen randomly, and the information gain is utilized to assist a DRL model in efficiently finding areas with more significant information gain. Typically, RRT-based algorithms sample new observations from the field randomly, which has low efficiency. In the present work, DRLT uses a DRL model to study the information gain of new observations of each action. Higher information gain will give a higher reward to the DRL, and the DRL will instruct the exploring tree to sample areas with more information rather than sample randomly.

However, naively computing the information gain from the infinite horizon is computationally intensive. Therefore, the present method also provides an efficient computation method for the information gain. In this chapter, the EOI is modeled as a linear-Gaussian dynamic system, which captures spatiotemporal correlations. Hence, the sampling value at each point of the EOI can be modeled as a dependent variable of other sampling locations. Additionally, a model-based mutual information gain is developed. Given a specific environmental model, the information gain can be computed efficiently from the spatiotemporal correlation. Thus, by encouraging the exploration and learning of areas of higher information gain, the locations for achieving a lower estimation error can be determined efficiently.

3.3.1 Guaranteed Near-optimality for Informative Sampling Using Exploring Tree

The sampling-based exploring tree can find the deployment strategy with minimal estimation error over the field, and overall, the solution is near-optimal. A pertinent proposition is presented now.

Proposition 3.1. *DRLT algorithm gives the minimal feasible estimation error, which monotonically decreases as the exploring tree expands.*

Proof of Proposition 3.1. Let G_i is the graph produced by the exploring tree iteration i, which contains a set of paths $\hat{\mathbb{N}}_i$. Then, $\mathcal{N}_i \in \hat{\mathbb{N}}_i$ gives the sampling locations computed by DRLT in the ith iteration, where $\mathcal{N}_i = \arg\min_{\mathcal{N}\in\hat{\mathbb{N}}_i} \Gamma(\mathcal{N})$, and the estimation error is $\epsilon_i = \min_{\mathcal{N}\in\hat{\mathbb{N}}_i} \Gamma(\mathcal{N})$. Thus, $\mathcal{N}_i$ gives the deployment set with minimal feasible estimation error ϵ_i in its iteration.

Next, in the iteration $i+1$, a new sampling location n is added to graph G_i, and the graph become G_{i+1}, and the set of paths become $\hat{\mathbb{N}}_{i+1}$. Denoting the added paths as $\hat{\mathcal{N}}$, where $\hat{\mathcal{N}} \cup \hat{\mathbb{N}}_i = \hat{\mathbb{N}}_{i+1}$. The minimal error in iteration $i+1$ becomes:

$$\begin{aligned}\epsilon_{i+1} &= \min_{\mathcal{N}\in\hat{\mathbb{N}}_{i+1}} \Gamma(\mathcal{N}) = \min_{\mathcal{N}\in\{\hat{\mathcal{N}}\cup\hat{\mathbb{N}}_i\}} \Gamma(\mathcal{N}) \\ &= \min(\epsilon_i, \min_{\mathcal{N}\in\hat{\mathcal{N}}} \Gamma(\mathcal{N})) \leq \epsilon_i.\end{aligned}$$

Thus, DRLT produces a monotonically decreasing estimation error as the algorithm iterates. □

Given the deployment strategy with minimal estimation error over the field, it can be shown that this result is near-optimal. Generally, the selection of locations over an EOI to minimize the cost is NP-hard [10], and finding the optimal solution is computationally expensive. Thus, a near-optimal solution is provided.

Normally, a near-optimal optimization can be guaranteed by greedy maximization or minimization of the submodular function [107]. The diminishing return property ensures convergence of the submodular function. A function $f(\cdot)$ is submodular if:

$$f(A \cup n) - f(A) \geq f(B \cup n) - f(B), \tag{3.5}$$

where $A \subset B \subset C$, and $n \in C \setminus B$.

Proposition 3.2. *The sampling locations with minimum estimation error returned by DRLT are near-optimal.*

Proof of Proposition 3.2. First, the function to calculate the estimation error should be submodular. Let $\mathcal{N}$ and $\mathcal{N}^*$ be two sets that contain sampling locations, where $\mathcal{N} \subset \mathcal{N}^* \subset \mathcal{A}$. Denote the new observation location as $n \in \mathcal{A} \setminus \mathcal{N}^*$, and the subtraction of sets is $\hat{\mathbb{N}} = \mathcal{N}^* \setminus \mathcal{N}$. Thus, the estimation error of set $\mathcal{N}^*$ is:

$$\begin{aligned}\Gamma(\mathcal{N}^*) = \pi(\mathbf{P}_\infty^{\mathcal{N}^*}) =& \pi(\mathbf{P}_\infty^{\mathcal{N}\cup\hat{\mathbb{N}}}) \\ &= \max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \max_{i\in\hat{\mathbb{N}}} \pi(\mathbf{P}_\infty^i)).\end{aligned}$$

Similarly,

$$\begin{aligned}\Gamma(\mathcal{N}^* \cup n) =& \pi(\mathbf{P}_\infty^{\mathcal{N}\cup\hat{\mathbb{N}}\cup n}) \\ &= \max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \max_{i\in\hat{\mathbb{N}}} \pi(\mathbf{P}_\infty^i), \pi(\mathbf{P}_\infty^n)) \\ \Gamma(\mathcal{N} \cup n) =& \pi(\mathbf{P}_\infty^{\mathcal{N}\cup n}) \\ &= \max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \pi(\mathbf{P}_\infty^n)).\end{aligned}$$

Then,

$$\begin{aligned}&\Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) - \Gamma(\mathcal{N}^* \cup n) + \Gamma(\mathcal{N}^*) \\ =& \max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \pi(\mathbf{P}_\infty^n)) - \max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i) - \\ &\max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \max_{i\in\hat{\mathbb{N}}} \pi(\mathbf{P}_\infty^i), \pi(\mathbf{P}_\infty^n)) + \\ &\max(\max_{i\in\mathcal{N}} \pi(\mathbf{P}_\infty^i), \max_{i\in\hat{\mathbb{N}}} \pi(\mathbf{P}_\infty^i)) \\ \geq& 0.\end{aligned} \tag{3.6}$$

Equation 3.6 can be reorganized as:

$$\Gamma(\mathcal{N} \cup n) - \Gamma(\mathcal{N}) \geq \Gamma(\mathcal{N}^* \cup n) - \Gamma(\mathcal{N}^*). \tag{3.7}$$

This is identical to Equation 3.5. Thus, the set function $\Gamma(\cdot)$ is submodular. Then,

using the greedily selected deployment set as shown in Proposition 3.1, we can achieve near-optimal solution. □

3.3.2 Model-based Spatiotemporal Information Gain

Now, a model-based spatiotemporal mutual information calculation is developed to efficiently capture the information gain of new observation locations. The result can be used as the action rewards of the DRL model, which accelerate the process of finding the deployment strategy with near-optimal estimation error on the EOI.

First, the covariance of two random variables, which represents the modeled value in two locations of EOI, can be computed from the environmental model according to:

$$cov(X, Y) = E[XY^T] - E[X]E[Y]^T, \tag{3.8}$$

where $X = v_{\psi_i}[t+1]$, $Y = v_{\psi_j}[t+1]$. Given the environmental model in Equation 2.2, where $w[t]$ and $v[t]$ are both Gaussian white noise with a zero mean, we can represent the X and Y in Equation 3.8 as:

$$\begin{aligned} v_{\psi_i}[t] =& \mathbf{S}_{\psi_i}\mathbf{x}[t] + v[t] \\ =& \mathbf{S}_{\psi_i}(\mathbf{A}\mathbf{x}[t-1] + w[t-1]) + v[t] \\ \Rightarrow X = v_{\psi_i}[t+1] =& \mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_i}w[t] + v[t+1]. \end{aligned} \tag{3.9}$$

and

$$Y = v_{\psi_j}[t+1] = \mathbf{S}_{\psi_j}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_j}w[t] + v[t+1]. \tag{3.10}$$

Substitute, Equation 3.9 and 3.10 into $E[XY^T]$:

$$\begin{aligned}
E[XY^T] =& E[(\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_i}w[t] + v[t+1]) \\
& \cdot (\mathbf{S}_{\psi_j}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_j}w[t] + v[t+1])^T] \\
=& E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]\mathbf{x}[t]^T\mathbf{A}^T\mathbf{S}_{\psi_j}^T + \mathbf{S}_{\psi_i}w[t]\mathbf{x}[t]^T\mathbf{A}^T\mathbf{S}_{\psi_j}^T \\
& + v[t+1]\mathbf{x}[t]^T\mathbf{A}^T\mathbf{S}_{\psi_j}^T + \mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]w[t]^T\mathbf{S}_{\psi_j}^T \\
& + \mathbf{S}_{\psi_i}w[t]w[t]^T\mathbf{S}_{\psi_j}^T + v[t+1]w[t]^T\mathbf{S}_{\psi_j}^T \\
& + \mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]v[t+1]^T + \mathbf{S}_{\psi_i}w[t]v[t+1]^T \\
& + v[t+1]v[t+1]^T].
\end{aligned}$$

Since $v[t]$ does not depend on $\mathbf{x}[t]$, we have $E[v[t]\mathbf{x}[t]^T] = 0$. Also, $\mathbf{E}\{w[i]v[j]^T\} = 0, \forall i, j$, $\mathbf{E}\{w[t]v[t+1]^T\} = 0$. Hence,

$$\begin{aligned}
E[XY^T] =& E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]\mathbf{x}[t]^T\mathbf{A}^T\mathbf{S}_{\psi_j}^T] \\
& + E[\mathbf{S}_{\psi_i}w[t]\mathbf{x}[t]^T\mathbf{A}^T\mathbf{S}_{\psi_j}^T] \\
& + E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]w[t]^T\mathbf{S}_{\psi_j}^T] \\
& + E[\mathbf{S}_{\psi_i}w[t]w[t]^T\mathbf{S}_{\psi_j}^T] + E[v[t+1]v[t+1]^T].
\end{aligned} \tag{3.11}$$

Furthermore, $E[\mathbf{x}[t]w[t]^T]$ can be simplified as:

$$\begin{aligned}
E[\mathbf{x}[t]w[t]^T] &= E[(\mathbf{A}\mathbf{x}[t-1] + w[t-1])w[t]^T] \\
&= E[\mathbf{A}\mathbf{x}[t-1]w[t]^T + w[t-1]w[t]^T] \\
&= E[\mathbf{A}\mathbf{x}[t-1]w[t]^T] \\
&= \cdots \\
&= E[\mathbf{A}^{t-1}\mathbf{x}[0]w[t]^T] = \mathbf{A}^{t-1}\mathbf{x}[0]E[w[t]^T] \\
&= \mathbf{A}^{t-1}\mathbf{x}[0] \cdot 0 = 0.
\end{aligned} \tag{3.12}$$

Next, substitute $E[w[t]w[t]^T] = \mathbf{R}_w$, $E[v[t+1]v[t+1]^T] = \mathbf{R}_v$ and Equation 3.12 into

Equation 3.11:

$$E[XY^T] = \mathbf{S}_{\psi_i}\mathbf{A}E[\mathbf{x}[t]\mathbf{x}[t]^T]\mathbf{A}^T\mathbf{S}_{\psi_j}^T + \mathbf{S}_{\psi_i}\mathbf{R}_w\mathbf{S}_{\psi_j}^T + \mathbf{R}_v. \tag{3.13}$$

Then, the $E[X]E[Y]^T$ part in Equation 3.8 can be expressed as :

$$\begin{aligned} E[X]E[Y]^T =& E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_i}w[t] + v[t]] \\ & \cdot E[\mathbf{S}_{\psi_j}\mathbf{A}\mathbf{x}[t] + \mathbf{S}_{\psi_j}w[t] + v[t]]^T \\ =& E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]] \cdot E[\mathbf{S}_{\psi_j}\mathbf{A}\mathbf{x}[t]] \\ =& E[\mathbf{S}_{\psi_i}\mathbf{A}\mathbf{x}[t]\mathbf{x}[t]^T\mathbf{A}\mathbf{S}_{\psi_j}] \\ =& \mathbf{S}_{\psi_i}\mathbf{A}E[\mathbf{x}[t]\mathbf{x}[t]^T]\mathbf{A}\mathbf{S}_{\psi_j}. \end{aligned} \tag{3.14}$$

Substitute equation 3.13 and 3.14 into function 3.8:

$$cov(v_{\psi_i}, v_{\psi_j}) = \mathbf{S}_{\psi_i}\mathbf{R}_w\mathbf{S}_{\psi_j}^T + \mathbf{R}_v$$

Some Gaussian noise is added to avoid the singularity.

As the values of all feasible sampling points, v_{ψ_i}, are random variables, they can be formulated as a random vector $\mathbf{V} = [v_{\psi_1}, v_{\psi_2}, \cdots, v_{\psi_n}]^T$. The matrix Σ is the covariance matrix of $\mathbf{V}$, where $\Sigma_{ij} = cov(v_{\psi_i}, v_{\psi_j}) = \mathbf{S}_{\psi_i}\mathbf{R}_w\mathbf{S}_{\psi_j}^T + \mathbf{R}_v$. Then,

$$\Sigma = \begin{bmatrix} \mathbf{S}_{\psi_1}\mathbf{R}_w\mathbf{S}_{\psi_1}^T & \mathbf{S}_{\psi_1}\mathbf{R}_w\mathbf{S}_{\psi_2}^T & \cdots & \mathbf{S}_{\psi_1}\mathbf{R}_w\mathbf{S}_{\psi_n}^T \\ \mathbf{S}_{\psi_2}\mathbf{R}_w\mathbf{S}_{\psi_1}^T & \mathbf{S}_{\psi_2}\mathbf{R}_w\mathbf{S}_{\psi_2}^T & \cdots & \mathbf{S}_{\psi_2}\mathbf{R}_w\mathbf{S}_{\psi_n}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{\psi_n}\mathbf{R}_w\mathbf{S}_{\psi_1}^T & \mathbf{S}_{\psi_n}\mathbf{R}_w\mathbf{S}_{\psi_2}^T & \cdots & \mathbf{S}_{\psi_n}\mathbf{R}_w\mathbf{S}_{\psi_n}^T \end{bmatrix} + \mathbf{R}_v \cdot \mathbf{J}, \tag{3.15}$$

where **J** is the matrix of ones.

It has been shown that entropy can be represented as a function of covariance [10]. Hence, the conditional entropy function in the right-hand side of Equation 3.2 can be represented as the monotonic function of its variance:

$$\begin{aligned}\mathcal{H}(n_i|\mathcal{N}_{i-1}) &= \frac{1}{2}\log(2\pi e\sigma^2_{n_i|\mathcal{N}_{i-1}}) \\ &= \frac{1}{2}\log(\sigma^2_{n_i|\mathcal{N}_{i-1}}) + \frac{1}{2}(\log(2\pi)+1).\end{aligned} \tag{3.16}$$

Denote Σ^*, the sub-matrix of Σ, as the covariance matrix of n_i and $\mathcal{N}_{i-1}$, which computes the conditional variance $\sigma_{n_i|\mathcal{N}_{i-1}}$ [108, 109]. The covariance matrix Σ^* is denoted as:

$$\Sigma^* = \begin{bmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{bmatrix},$$

where Σ_{11} is the variance of n_i, and Σ_{22} is the covariance of set $\mathcal{N}_{i-1}$. Then, Σ_{12} and Σ_{21} give the covariances between the random variable n_i and the set $\mathcal{N}_{i-1}$. The conditional variance $\sigma_{n_i|\mathcal{N}_{i-1}}$ can be expressed as:

$$\sigma_{n_i|\mathcal{N}_{i-1}} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \tag{3.17}$$

As the present goal is to maximize the increased MI while retrieving new observations n from the field in each iteration, the increased MI (information gain) can be represented as:

$$\delta_n = \mathcal{MI}(\mathcal{N} \cup n) - \mathcal{MI}(\mathcal{N}). \tag{3.18}$$

According to [10] and Equations 3.3, 3.16 and 3.17, Equation 3.18 can be simplified as:

$$\begin{aligned}\delta_n &= \mathcal{H}(n|\mathcal{N}) - \mathcal{H}(n|\bar{\mathcal{N}}) \\ &= \frac{1}{2}\log(\sigma^2_{n|\mathcal{N}}) - \frac{1}{2}\log(\sigma^2_{n|\bar{\mathcal{N}}}) \\ &= \log(|\frac{\Sigma_n - \Sigma_{n\mathcal{N}}\Sigma^{-1}_{\mathcal{N}\mathcal{N}}\Sigma_{\mathcal{N}n}}{\Sigma_n - \Sigma_{n\bar{\mathcal{N}}}\Sigma^{-1}_{\bar{\mathcal{N}}\bar{\mathcal{N}}}\Sigma_{\bar{\mathcal{N}}n}}|),\end{aligned} \tag{3.19}$$

where $\bar{\mathcal{N}} = \mathcal{A} \setminus (\mathcal{N} \cup n)$.

In this manner, the model-based spatiotemporal information gain can be calculated efficiently using Equation 3.15 and Equation 3.19. Besides, the calculation for information gain on any new sampling location over the infinite horizon is feasible.

3.3.3 Deep Reinforcement Learning Model for Maximum Information Retrieval over a Spatiotemporal Field

In this section, a DRL model is developed for learning the information gain of sampling locations and gives instructions on sampling actions. The DRL model does not depend on prior training knowledge to accelerate the exploration, and it facilitates the exploring tree to expand by monitoring the quality of the new sampling locations online. The model architecture is shown in Figure 3.1.

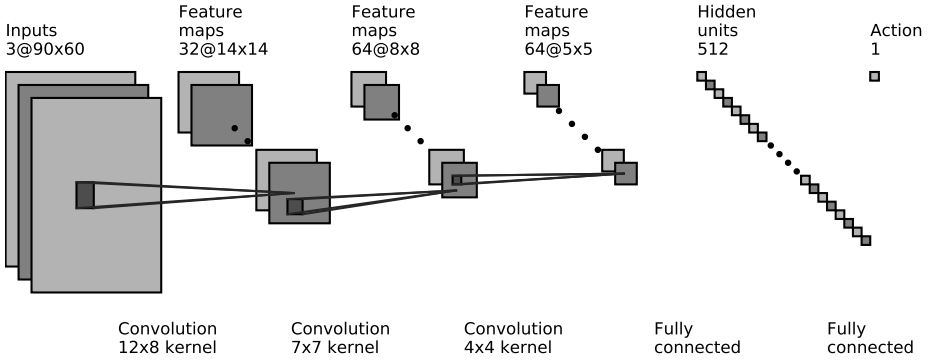


Figure 3.1: Schematic architecture for DRL model[1]

[1]This figure is generated by adapting the code from https://github.com/gwding/draw_convnet

The input for the DRL model is the frame that indicates the current sampling locations on the EOI. Then the input frame is downscaled to a $90 \times 60 \times 3$ image (from $1200 \times 800 \times 3$) to reduce the computational expense and the memory usage. The input frame to the DNN has size $90 \times 60 \times 3$, which is processed by a convolution layer with 32 filters of 12×8 with stride 6×4, followed by rectifier linearity (ReLu). The second layer has 64 filters of 7×7 with stride 1, is also followed by a ReLu. The third convolutional layer convolves 64 filters of size 4×4 with stride 1 followed by ReLu. The final hidden layer is fully-connected and consists of 512 rectifier units. The output layer is fully-connected with a single output for each valid action.

In the case of random exploring trees, the location of the newly added observation on the EOI is infinite. So, in the proposed model, the selection of the location is narrowed down towards the direction of its closest vertex in the spanning tree. Then, their relative direction is divided into right angles, and the selection of new locations is decomposed into four actions. In Figure 3.2, the blue circle represents the current sampling locations, and diamonds stand for the possible sampling locations for the next step. The relative angle is divided into four right angles as indicated by the background color. Note that the actions pass to the DRL model includes only the corresponding right angle of the new sampling location.

The reward that is passed to the DRL network is given by the information gain, as obtained in Section 3.3.3. In each iteration, the information gain of the new sampling location is calculated and encourages the mobile agents to move towards the area of higher information gain. Besides, as the variable with the small value in the logarithmic function (Equation 3.19) gives a sizable negative information gain, all non-positive gains are clipped to zero. Clipping the information gain in this manner makes the model to merely focus on searching the area with a more significant positive information gain and makes it easier to learn.

The DRL model takes on the learning process as the tree expands. In the beginning, the DRL model produces random actions for the exploring tree. Then, as more rewards from

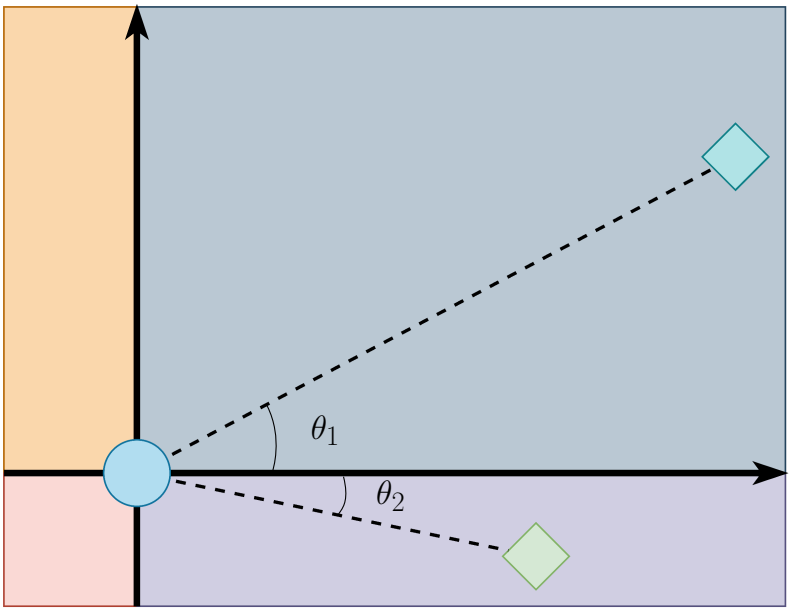


Figure 3.2: Illustration of the action structure. Blue circle: current sampling locations; diamonds: possible sampling locations for next step.

Algorithm 3.1: extendDRL()

Input: $G, \mathcal{A}, \Sigma$
Output: n

Initialization: set replay buffer B with size N, set value function Q with randomized weight θ;
With probability p select a random action $a_t \in \mathbb{A}$, otherwise select $a_t = \arg\max_a Q(s_t, a_t; \theta)$;
Compute new sampling location n based on a_t and $\mathcal{A}$.;
Generate deployment strategy $\mathcal{N}$ based on n and G;
Limit selected action a_t to step size;
Execute action a_t to calculate reward $r_t = \delta_{n|\mathcal{N}}$ from Σ and observe new state s_{t+1};
Save state transition (s_t, a_t, s_{t+1}, r_t) to B;
Perform a Adam optimization to update the model.

the sampling location are given, more precise instructions on the tree expanding will be obtained. The deep Q-learning algorithm for tree expansion is given in Algorithm 3.1.

The input for the Algorithm 3.1 is the exploring tree, which provides the current state and then outputs the generated sampling location. In the beginning, the agent selects an action with the p-greedy policy based on Q, which selects a random action from the action set $\mathbb{A}$ with probability p and uses the Q-based greedy policy with probability $1 - p$. Then,

the action is executed, and the deployment strategy is generated to calculate the reward. The deployment strategy is the new sampling location n combined with its closest path in the tree G. Last, the model is updated with experience replay [42] and a double Q-network to [110] to avoid divergence. In this manner, the utilization of the deep Q-network can study the information gain during the tree exploration period. Then, as areas with higher information gain are sampled at a higher priority, the overall estimation error will reduce quickly.

3.3.4 Architecture for Deep Reinforced Learning Tree

Based on the results in Sections 3.3.1, 3.3.2 and 3.3.3, DRLT is proposed here to search for near-optimal sampling locations and efficiently estimate the spatiotemporal EOI.

The deep reinforced exploring learning tree is built based on the proposed DRL model to accelerate the process of finding near-optimal sampling locations. The DRLT builds a reinforced space-filling tree, which efficiently searches the non-convex EOI. Unlike traditional RRT-based algorithms, DRLT does not randomly expand the space-filling tree. It studies the collected information from the field and focuses more on areas of higher interest, which helps to achieve lower estimation error faster. The DRLT is described in Algorithm 3.2. It begins with the initialization of the exploring tree and the value of estimation error. The initial estimation error is set to positive infinity. Then, the algorithm iterates T times. In each iteration, the sampling location is estimated by the DRL model using the method presented in Algorithm 3.1. Then, line 4 finds the nearest vertex of n in G. And line 5 finds a set of vertices in tree G, which is within the radius r of the sampling point n. It is followed by the function that generates all possible sampling strategies in line 6. The details of line 6 are in Algorithm 3.3. Next, line 7 finds the sampling strategies with the lowest estimation error given the new sampling point n. Last, the algorithm checks if a lower overall estimation error is reached and correspondingly expands the exploring tree. In this manner, the DRLT algorithm finds the near-optimal solution for the informative sampling problem, which is

executed efficiently through the help of DRL model.

Algorithm 3.2: Deep Reinforced exploring Learning Tree

Input: $\mathcal{A}, \mathbf{A}, \mathbf{R}_w, \mathbf{R}_v, \mathbf{S}_\psi, \Sigma$
Output: $\mathcal{N}$

Initialization: set exploring tree V with starting point ψ_0, edge E as empty set $\emptyset$ and tree G as (V, E). The minimum estimation error is set to infinity in the beginning, $\omega = \infty$;
for *1 to* T **do**
  $n = extendDRL(G, \mathcal{A}, \Sigma)$;
  $n_{near} = Nearest(n, G)$;
  $\mathbb{E} = Cluster(G, n, r)$;
  $\hat{\mathbb{N}}$ = GenDEPL(n, $\mathbb{E}$, G);
  $\tilde{\mathcal{N}} = \arg\min_{\tilde{\mathcal{N}} \in \hat{\mathbb{N}}} \Gamma(\tilde{\mathcal{N}})$;
  if $\Gamma(\tilde{\mathcal{N}}) < \omega$ **then**
    $\omega = \Gamma(\tilde{\mathcal{N}})$;
    $\mathcal{N} = \tilde{\mathcal{N}}$;
  end
  $V \leftarrow V \cup n$;
  $E \leftarrow E \cup edge(n, n_{near})$;
  $G \leftarrow (V, E)$;
end

Algorithm 3.3 generates possible sampling strategies given the new sampling location n and its neighbors $\mathbb{E}$. It outputs a set of subtrees of G as sampling plans including node n and each of its neighbors. Then, the sampling plan with lowest estimation error is selected from these available sampling plans. Next, the deployment strategy is compared with the overall estimation error to find the global minimal. The input set for Algorithm 3.3 is the new sampling point n, the set of its neighbors $\mathbb{E}$, and the exploration tree G. Each deployment strategy is generated based on the basic RRT-based planning principle, which traces back from the locations in the set $\mathbb{E}$ to its root. Hence, the algorithm generates sampling strategies for each neighbor of n.

In this manner, the DRLT algorithm can find a near-optimal deployment strategy with minimal estimation error from the field. By encouraging itself to explore areas with higher information gain using DRL, the DRLT algorithm also completes the goal efficiently.

Algorithm 3.3: GenDEPL()

Input: n, $\mathbb{E}$, G
Output: $\hat{\mathbb{N}}$

$\hat{\mathbb{N}} = \emptyset$;
foreach $n_e \in \mathbb{E}$ **do**
  $n' = n_e$;
  $\mathcal{N} = \{n, n'\}$;
  while *hasparent(n')* **do**
    n'=parent(n', G);
    $\mathcal{N} \cup n'$
  end
  $\hat{\mathbb{N}} \cup \mathcal{N}$
end

3.4 Simulation Results

This section presents numerical simulations using the proposed algorithm and data from NOAA. Specifically, the DRLT algorithm and benchmark algorithms are used to plan the most informative sampling locations to estimate the SST over the Caribbean Sea.

3.4.1 NOAA Sea Surface Temperature Dataset

The test dataset for the proposed algorithm and the benchmark algorithms is NOAA coastal water temperature dataset, which contains the sea surface temperature of a rectangular region of size 1200 units $\times$ 800 units at 1/30 degree ($\sim$3km) resolution in the Caribbean Sea [83, 85]. In this dataset, the locations of the obstacles and the sampling data of the basis center are given. The preprocessing of the dataset fills the missing measurements of the stationary data, and the missing measurements are merely recorded as the same value as the previous one. As the values of such attributes do not change significantly in a day when compared to seasonal changes, the dataset is down-sampled to simplify the modeling process by setting a time window to 6 hours. After that, the subspace identification algorithm [82] is utilized to learn temporal information of the environmental model.

3.4.2 Benchmark Algorithms

The proposed algorithm are compared with three state-of-art algorithms: Information driven planner (INFO) [99], Rapidly exploring Random Cycles (RRC) [3], and Policy iteration on continuous domains using RRTs (RRTPI) [13]. Also, a native RRT is implemented to find informative sampling locations for the benchmark algorithm. Only the informativeness maximization procedure of INFO is adapted to produce sampling locations; the exploration method remains the same for the proposed algorithm and other benchmark algorithms. The planner of INFO samples the next point with the largest mutual information gain in the field. By building an exploring tree with such points, it will gradually find the most informative strategy with the lowest estimation error. Besides, RRC seeks to build cycles from the exploring tree with the lowest estimation error. The deployment strategy is established by finding suitable cycles with the lowest estimation error among the paths of an exploring tree. Similar to RRC, the RRT finds the paths but not cycles of the exploring tree with the lowest estimation error. Also, both RRC and RRT explore the tree by random sampling from the field, which does not utilize the feedback of information gain of the sampling points. RRTPI is an approach that is intended for solving geodesic-based problems. Therefore, top-10 locations are picked in the area with the highest mutual information. The exploring tree will have more substantial rewards when approaching those top-10 locations. In this manner, the exploring tree will tend to sample more points in the areas of higher information gain. The simulation was conducted on a PC with 4.0 GHz Quad-Core CPU and 16 G memory. The DRL model was implemented using the TensorFlow framework [111] and run without GPUs.

The change of SST is modeled using the spatial basis function and the temporal correlation. As given in Equation 2.4, the Gaussian radian basis function is selected as the basis function. Hence, the ith element of the temporal matrix is $s_i(\psi) = Ke^{-||(\psi_t^i)-(\psi_j)||^2/2\sigma^2}$, where K is the scale factor, ψ_t^i is the location of the ith robotic sensor in the field at time t, ψ_j is the location of the ith basis center, and σ is the Gaussian variance. In the present work, the

scale factors are selected as 10 and the variance is 100 units. The Gaussian radial basis centers are selected using the metrics described in [3], and the number of basis centers is nine. The temporal information is learned from the NOAA dataset. The starting location of the exploring tree for the proposed and benchmark algorithms is $\psi_0 = \{500units, 200units\}$ and the step size for the exploring tree is 300 units. The number of iteration is limited to 500 times due to the fast convergence of the proposed algorithm. It is found that the performance of the proposed work does not depend on the selected environmental model and its parameters.

3.4.3 Performance Comparison

Figure 3.3 (a) - (e) presents the information-based sensor deployment strategies as computed using INFO, RRC, RRT, RRTPI, and DRLT, respectively, where the black stars indicate the location of informative sampling locations, the grey area represents the obstacles, the white area is the EOI, and the blue lines indicate the exploring tree generated from different algorithms. In Figure 3.3 (d), yellow squares represent the pre-calculated top-10 locations with the highest information gain in the EOI. For all five figures, the proposed algorithm and the benchmark algorithms, both searched over most of the EOI, which fulfill the goal of minimizing the estimation error over the EOI by making observations. Besides, the proposed algorithm tends to search some specific areas that have higher information gain and leads to a faster realization of the minimum estimation error.

Figure 3.4 shows the field estimation uncertainty over the EOI when deploying robotic sensors using the information-based deployment strategy as generated by the proposed and the benchmark algorithms. Then, the color map is used to indicate the estimation uncertainties when deploying robotic sensors in the locations of red dots. A more significant number in the color bar represents a higher variance in the EOI, and indicates that the deployment strategy will have higher uncertainty in this area. Also, as all five algorithms achieve near-optimal solutions, the estimation uncertainty over the field of the resulting sampling

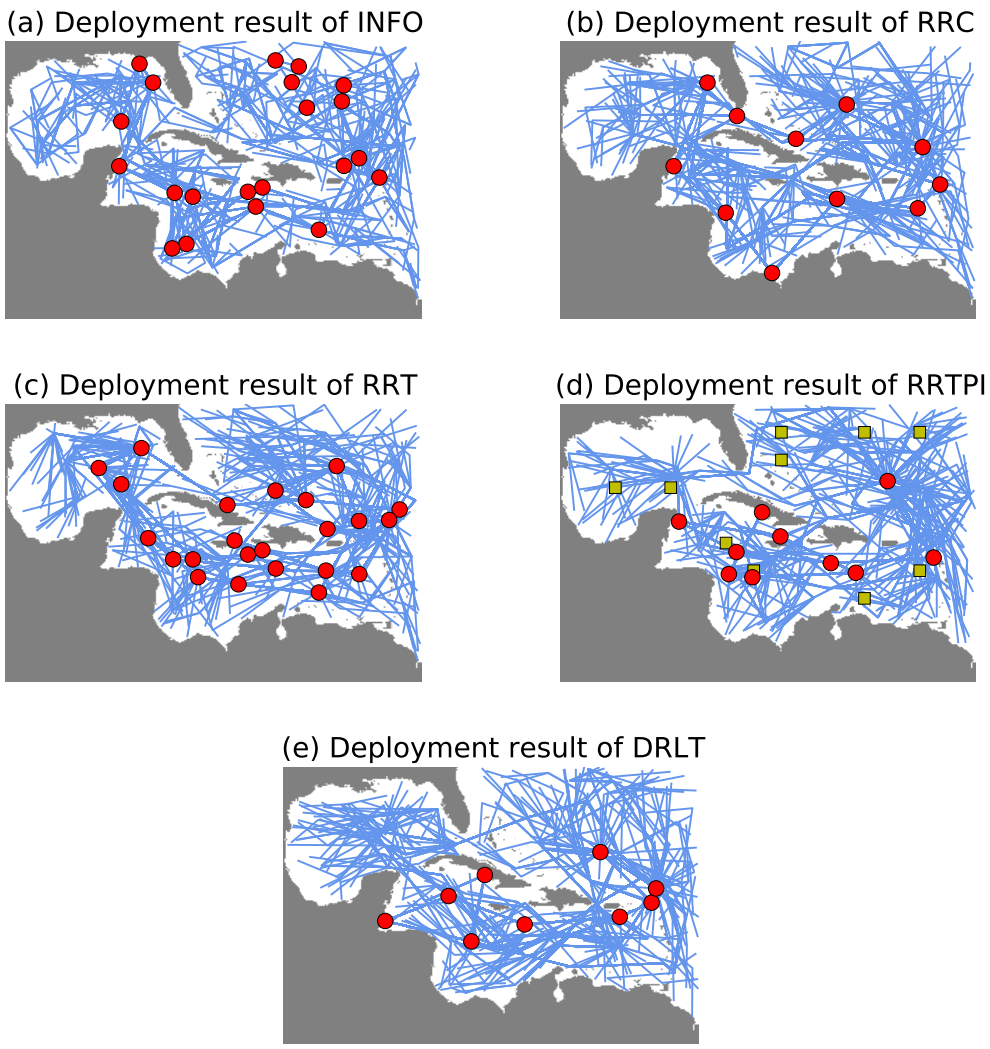


Figure 3.3: Deployment strategy for information-based sampling. Red dots: computed information-based sampling locations; blue lines: exploring tree; yellow squares: informative centers for RRTPI.

strategies is at a low level. Specifically, when the robotic sensors are deployed using the strategy provided by DRLT, the maximum uncertainty over the field is 4.75, and the uncertainty does not exceed 2 in most areas. Also, the average field uncertainty is as low as 0.73, which means that the proposed information-based deployment strategy has fast convergence in the EOI. INFO has performed well in minimizing the field estimation uncertainty with the computed deployment strategy. As it is an algorithm that is based on information gain, it attempts to find the sampling location with the highest information gain from the field.

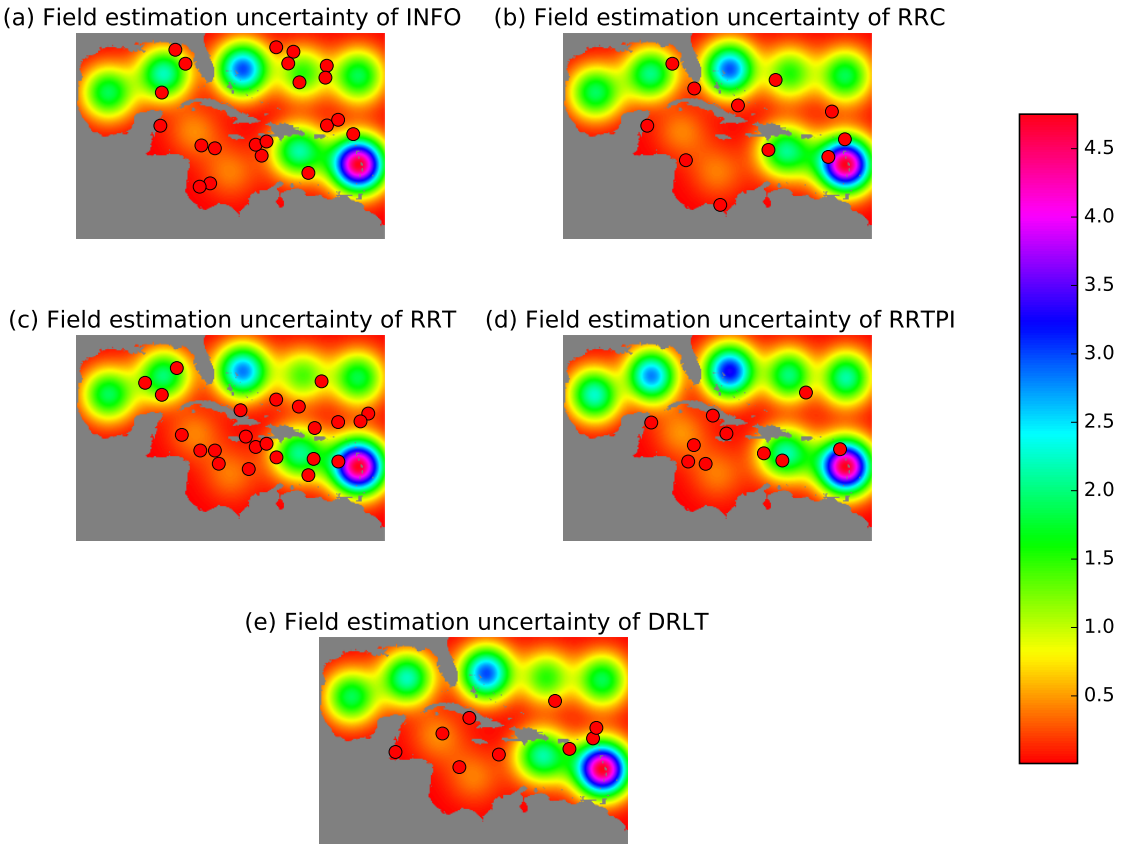


Figure 3.4: Field estimation uncertainty for information-based sampling.

Hence, it achieves a low field estimation uncertainty. Other benchmark algorithms also achieve low estimation uncertainty as all of them can find near-optimal results.

Figure 3.5 shows how the estimation error changes as the number of iterations increases. The upper bound for visualizing the lowest estimation error is limited to 400 due to the high estimation error in the beginning. The cost of the computed informative sampling strategies decreases monotonically as the algorithms execute. The proposed algorithm, DRLT, uses the least number of iterations to reduce the overall estimation error from infinity to 400, which shows an exceptional ability in finding better sampling locations with limited information. The DRLT also finds a deployment strategy with lower estimation error faster, compared to the benchmark algorithms in the subsequent iterations. Moreover, among all five algorithms, DRLT achieves the lowest estimation error in the field using a limited num-

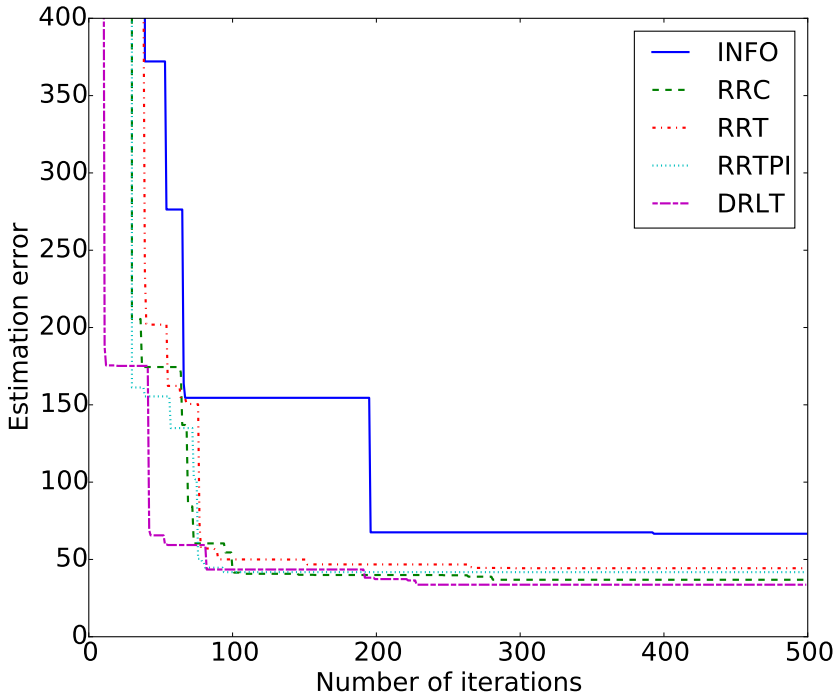


Figure 3.5: Reduction of the estimation error with iteration (Blue solid line: result of INFO; green dash line: RRC; red dash-dot line: RRT; light blue dot line: RRTPI; purple dash-dash line: DRLT).

ber of iterations, which also shows that it can find a near-optimal solution efficiently and using fewer iterations of computation.

Figure 3.6 presents the minimal estimation found by the proposed and the benchmark algorithms with respect to the computational time. The RRC algorithm takes the longest time to complete 500 iterations of tree expansion, followed by the INFO algorithm. Both algorithms take more than two thousand seconds, which is not computationally efficient when compared to the speed of other three algorithms. Since RRT, RRTPI, and DRLT use less than eight hundred seconds to complete the computation, Figure 3.6 is zoomed in as Figure 3.7 to compare their performance.

Figure 3.7 indicates that DRLT uses the least computing time to find the deployment strategy with the lowest estimation error in the field. RRT and RRTPI also find a good deployment strategy efficiently, but they seem to get trapped locally, and the estimation

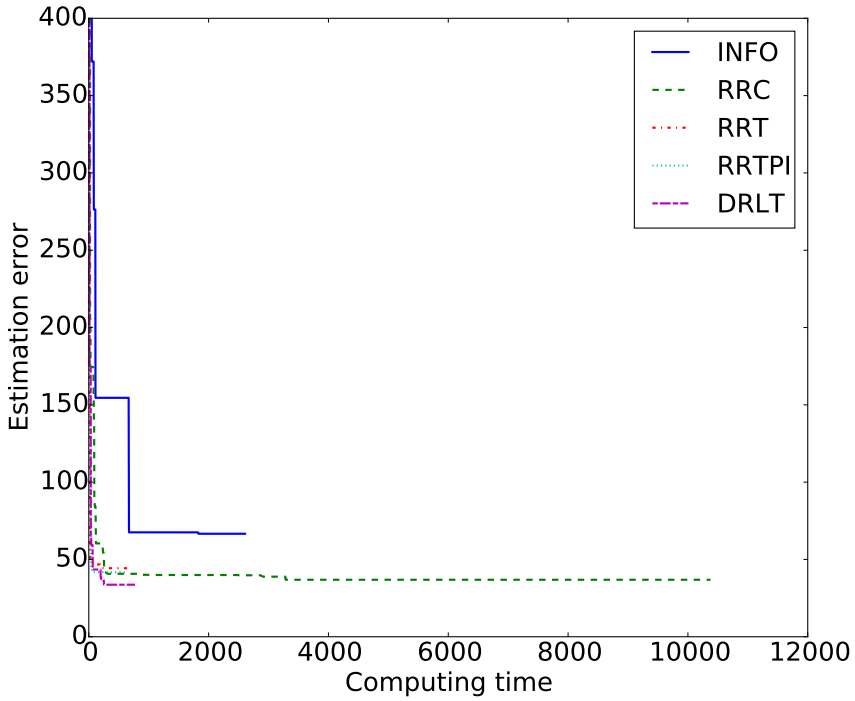


Figure 3.6: Estimation error of the information-based deployment strategy determined by the proposed and the benchmark algorithms decreases monotonically with time.

error stops decreasing quite early. INFO performs worst with the highest estimation error and long computation time. Moreover, although RRC finds the second best solution within three hundred seconds, it takes more than ten thousand seconds to finish all five hundred iterations, which might be caused by the high computational cost of finding cycles among the paths in the exploring tree. The cycles are determined by finding two paths that share the same parent from the newly added observation. Hence, as the size of the exploring tree grows, the computation for finding the cycles with a lower estimation error increases exponentially.

Figure 3.8 presents the change of the cumulative reward of DRLT during the exploration and the corresponding estimation error in each iteration. In the present work, the cumulative reward is set to the sum of the information gain when sampling in a new location, and higher cumulative rewards can lead to a better learning process. Notably, DRLT maximizes the

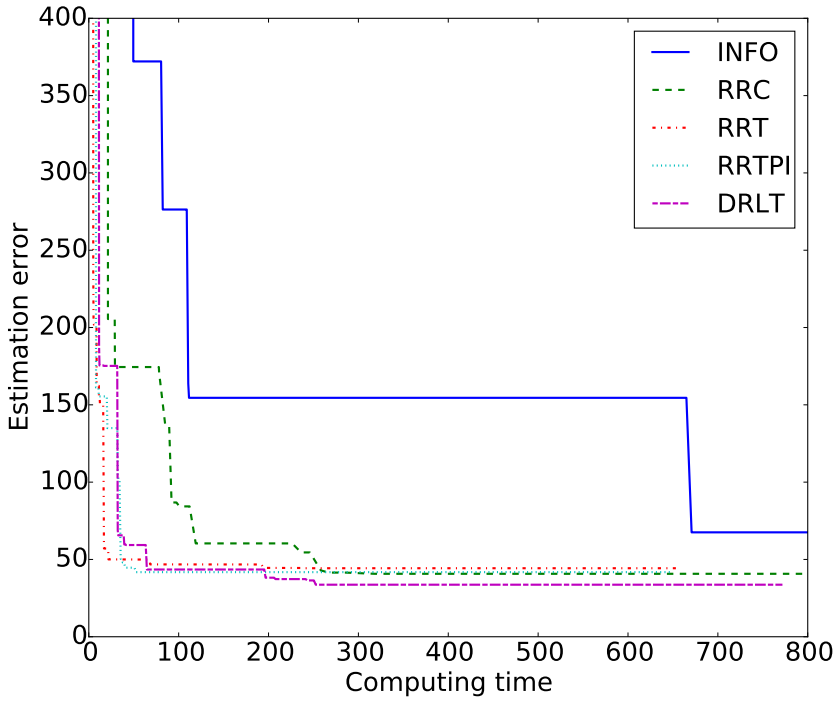


Figure 3.7: Estimation error of the information-based deployment strategy determined by the proposed and the benchmark algorithms decreases monotonically with time. (Zoomed in)

cumulative reward in a stable manner as the tree explores, which lead to a steady state with the near-optimal result. By comparing the two curves in Figure 3.8, it is seen that DRLT can retrieve information from the field continuously to efficiently reduce the estimation error.

Table 3.1 summaries the numerical results of the simulations. It shows that the DRLT achieves the lowest estimation error at 33.66, which is 9% lower than the second-best result (RRC) and about half the worst result. Also, DRLT does not require a high computing time to obtain this near-optimal result. It only takes 771.85 seconds to complete 500 iterations of tree exploration, which only takes 7% of the computing time to outperform the second-best result. The small estimation error together with the low computing time indicate that DRLT is the most efficient algorithm out of those considered, for obtaining a near-optimal deployment strategy. Besides, by adopting sampling strategies for robotic sensor deployment from both the proposed and benchmark algorithms, the field estimation uncertainty also becomes

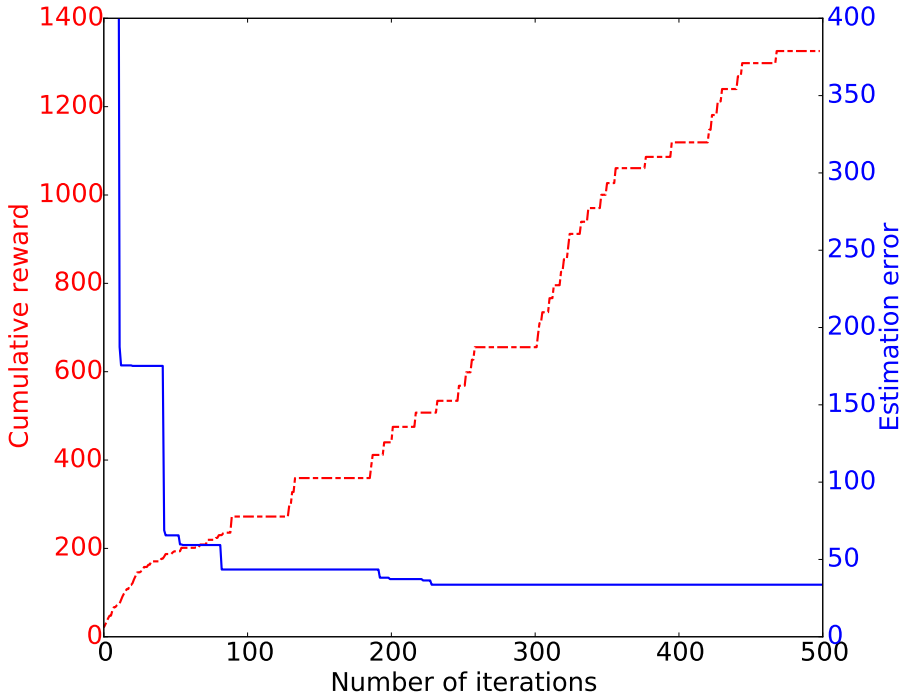


Figure 3.8: Cumulative reward and estimation error of DRLT. (Red dashed line: cumulative reward of the DRLT; solid blue line: corresponding estimation Error).

small. In particular, the maximum uncertainty over the field does not exceed 5 for all algorithms, and the average uncertainty is about 0.7. The speed reduction of the estimation error convergence is also examined. Setting the threshold at 400, DRLT only takes 11 iterations and 11.44 seconds to reduce the estimation error from infinity to the threshold value. All other algorithms require more than 30 iterations to achieve this reduction.

Table 3.1: Numerical results for DRLT and benchmarks.

Algorithm	INFO	RRC	RRT	RRTPI	DRLT
Estimation Error	66.62	36.86	44.31	41.78	33.66
Computing Time	2608.61	10375.94	653.27	649.10	771.85
Max Uncertainty	4.60	4.61	4.61	4.86	4.75
Min Uncertainty	0.0056	0.0056	0.0056	0.0060	0.0058
Mean Uncertainty	0.70	0.71	0.69	0.81	0.73
Estimation Error Reaches 400 (iteration)	39	30	39	30	11
Estimation Error Reaches 400 (time)	49.49	21.30	4.96	8.017	11.44

3.5 Summary

This chapter focused on developing a WSN deployment strategy that could efficiently achieve minimal estimation error. A novel algorithm, DRLT, was proposed to search over a sizable spatiotemporal field efficiently with the assistance of information gain. The developed deployment strategy was proved to be near-optimal. The simulation results were obtained for the NOAA dataset, and they confirmed that the proposed algorithm could outperform the benchmark algorithms in both performance and efficiency.

Chapter 4

Model-based Sensor Deployment for Spatiotemporal Field Reconstruction

4.1 Introduction

Besides spatiotemporal field estimation, it is essential to reconstruct the spatiotemporal field using the sampling data. The pollution and environmental changes can be detected and predicted from the reconstructed field. Subsequently, corrective actions may be taken to minimize the hazards. This chapter seeks to reconstruct the entire spatiotemporal field from the observations sampled at near-optimal sampling locations.

The spatiotemporal field should be constructed using observations sampled in the model-based near-optimal sampling locations. With the help of sparse learning, statistical methods are widely used to recover signals from limited observations [4, 58, 112]. Sparse learning investigates linear mappings from low-dimensional sparse observations to the original high-dimensional signal. To this end, a method based on principal component analysis (PCA) can be utilized to learn a projection basis from the training data. Then, a tailored singular value decomposition (SVD) approach can be used to reconstruct the entire spatiotemporal field, given future observations in the field.

In this chapter, a model-based approach for sensor node deployment and spatiotemporal

field reconstruction is proposed. The proposed approach deploys sensor nodes that can represent the spatiotemporal field near-optimally given the environmental model. Observations in the deployed locations can then be used to reconstruct the spatiotemporal field by learning the sparse representation mapping. Moreover, it is proven that the sensor deployment method, DRLT, generates near-optimal sensor deployment locations for spatiotemporal field reconstruction.

4.2 Mathematical Background and Problem Formulation

4.2.1 Sparse Sensor Deployment Analysis

The PCA is optimal for recovering high-dimensional signals from unknown contents. It converts correlated observations into principal components that have mere linear redundancy. In this manner, high-dimensional signals can be represented by a few sparse representations in the field. Then, sparse learning methods can be used to study the relationship between the compressed signal and the original one, to reconstruct a full signal from a small subset of measurements.

Denote the EOI as a high dimensional state $\boldsymbol{\phi} \in \mathbb{R}^m$, and the spatiotemporal dynamics of $\boldsymbol{\phi}$ can be captured by the low dimensional observations. The selected observations ($\mathbf{y}$) can be represented by a measurement matrix $\mathbf{C} \in \mathbb{R}^{r \times m}$, where $r \ll m$ is the number of measurements:

$$\mathbf{y} = \mathbf{C} \cdot \boldsymbol{\phi}. \tag{4.1}$$

Note that $\boldsymbol{\phi}$ can be sparsely represented by a principal basis:

$$\boldsymbol{\phi} = \boldsymbol{\Psi}_r \cdot \mathbf{a}, \tag{4.2}$$

where $\boldsymbol{\Psi}_r \in \mathbb{R}^{m \times r}$ is the principal basis learned from training data, and $\mathbf{a} \in \mathbb{R}^r$.

4.2.2 Problem Formulation

Mathematically, the sparse signal reconstruction and be expressed as the projection from a low-dimensional observation space to a high-dimensional signal:

$$\hat{\boldsymbol{\phi}} \leftarrow \xi(\mathbf{y}) : \mathbb{R}^m \leftarrow \mathbb{R}^r.$$

As $r \ll m$, this system is underdetermined. Hence, there are infinite solutions. In this manner, sampling locations should be optimized, and the mean error is utilized to evaluate the reconstructed spatiotemporal field:

$$\arg\min_{\xi} \|\boldsymbol{\phi} - \xi(\mathbf{y})\|.$$

4.3 Proposed Method

4.3.1 Sparse Learning for Spatiotemporal Reconstruction

High-dimensional states $\boldsymbol{\phi} \in \mathbb{R}^m$ can be represented as linear combinations of r orthogonal eigenmodes ψ. This low-dimensional state can be learned from training data by pruning the PCA basis. $\boldsymbol{\phi}$ may be decomposed to produce temporal and spatial coefficient [4]. In this manner, the linear combination of $\boldsymbol{\phi}$ shown in Equation 4.2 can be represented as:

$$\boldsymbol{\phi}_i \approx \sum_{k=1}^{r} a_k(t_i)\psi_k(\phi), \tag{4.3}$$

where $\boldsymbol{\phi}_i$ is the spatiotemporal field at time i and $\psi_k(\phi)$ is the time independent spatial coefficient. The temporal coefficient $a_k(t_i)$ varies with time t_i.

Next, ψ_k and a_k can be computed by Singular Value Decomposition (SVD). Training data $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2 \dots \boldsymbol{\phi}_M]$ are given for M snapshots. The tailored SVD basis consists of orthonormal left singular vectors $\boldsymbol{\Psi}$, right singular vectors $\mathbf{V}$ and the diagonal matrix $\mathbf{S}$.

Hence, the training data can be represented as:

$$\mathbf{\Phi} = \mathbf{\Psi} \cdot \mathbf{S} \cdot \mathbf{V}. \tag{4.4}$$

Then, the dimension of the right-hand part of Equation 4.4 can be reduced to r according to the Eckart-Young theorem [113]:

$$\begin{aligned} \mathbf{\Phi} \approx \mathbf{\Phi}^{*} = \arg\min_{\tilde{\mathbf{\Phi}}} \left\| \mathbf{\Phi} - \tilde{\mathbf{\Phi}} \right\|_F \\ s.t. \quad rank(\tilde{\mathbf{\Phi}}) = r, \end{aligned} \tag{4.5}$$

where $\mathbf{\Phi}^{*} = \mathbf{\Psi}_r \cdot \mathbf{S}_r \cdot \mathbf{V}_r$ and $\|\cdot\|_F$ is the Frobenius norm. In this manner, PCA can reduce the dimension of the high-dimensional system by using orthogonal projection. In the present work, rank r indicates the number of observations in the EOI. Hence, information in the corresponding r rows of $\mathbf{\Psi}$, $\mathbf{S}$, and $\mathbf{V}$ will be extracted as $\mathbf{\Psi}_r$, $\mathbf{S}_r$, and $\mathbf{V}_r$ respectively.

Then, a canonical measurement matrix $\mathbf{C}$ is used to select critical sampling locations sparsely, according to Equation 4.1:

$$\mathbf{C} = [\boldsymbol{e}_{\gamma_1}, \boldsymbol{e}_{\gamma_2}, \cdots, \boldsymbol{e}_{\gamma_r}]^T, \tag{4.6}$$

where $\boldsymbol{e}_j \in \mathbb{R}^m$ is the canonical basis vectors. Since matrix $\mathbf{C}$ is canonical, the observation $\mathbf{y}$ can be simplified as:

$$\mathbf{y} = \mathbf{C}\boldsymbol{\phi} = [\phi_{\gamma_1}, \phi_{\gamma_2}, \cdots, \phi_{\gamma_r}]^T, \tag{4.7}$$

where $\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_r\} \subset \{1, 2, \cdots, m\}$ is the set of indexes for selected sensors. In this manner, Equation 4.1 can be used to derive sampling locations from $\boldsymbol{\phi}$ efficiently.

Next, the observations in the field, $\phi_j \in \boldsymbol{\phi}$, can be represented as a linear combination of the basis and the coefficients according to Equation 4.3 and Equation 4.5:

$$\phi_j = \sum_{k=1}^{r} \mathbf{\Psi}_{jk} \mathbf{a}_k. \tag{4.8}$$

Observations in the sensor deployment locations can then be expressed as the linear combination of the canonical basis and the states according to Equation 4.7 and Equation 4.8:

$$y_i = \sum_{j=1}^{n} \mathbf{C}_{ij}\phi_j = \sum_{j=1}^{n} \mathbf{C}_{ij} \cdot \sum_{k=1}^{r} \mathbf{\Psi}_{jk}\mathbf{a}_k. \tag{4.9}$$

Equation 4.9 can be simplified as:

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{\Psi}_r \cdot \mathbf{a}. \tag{4.10}$$

According to Equation 4.3, $\boldsymbol{\phi}$ can be reconstructed by the basis coefficients $\hat{\mathbf{a}}$:

$$\hat{\boldsymbol{\phi}} = \mathbf{\Psi}_r\hat{\mathbf{a}}. \tag{4.11}$$

Usually, only r sensors are deployed in the field, and only the corresponding readings are obtained, which is denoted as $\mathbf{y} \in \mathbb{R}^r$. Hence, according to Equation 4.10 and Equation 4.11:

$$\begin{aligned} \hat{\boldsymbol{\phi}} =& \mathbf{\Psi}_r\hat{\mathbf{a}} \\ =& \mathbf{\Psi}_r(\mathbf{C}\mathbf{\Psi}_r)^{-1}\mathbf{y}, \end{aligned} \tag{4.12}$$

where $\mathbf{\Psi}_r \in \mathbb{R}^{n\times r}$ can be learned through SVD. $\mathbf{C} \in \mathbb{R}^{r\times n}$ can be obtained by finding the most informative locations in the field, such as near-optimal sensor deployment locations developed in Chapter 3. Thus, the environment $\hat{\boldsymbol{\phi}}$ can be reconstructed given r observations $\mathbf{y}$, where $m \gg r$.

4.3.2 Model-based Sparse Sensor Deployment with Informative Sampling

Optimal sensor deployment locations can guarantee the most accurate reconstruction of $\hat{\boldsymbol{\phi}}$. For this reason, the locations of the sensor nodes should be optimized to acquire the most

information from point measurements in sparsely selected sampling locations. Observations from the sparse sampling locations can then be used to reconstruct high-dimensional states, given the tailored SVD basis. Recall that γ represents the structure of the canonical matrix $\mathbf{C}$, which gives the sampling locations in the field. Therefore, γ affects the reconstruction performance. Thus, the optimal γ and its corresponding sampling locations should provide the most information about the field.

Choosing a suitable number of sensors in the field while maintaining a low noise level in the data is a challenge. Gavish et al. proved an optimal threshold, singular value hard thresholding (SVHT), that guarantees minimal signal reconstruction error [114]. While ensuring least reconstruction error, SVHT finds only the optimal value for reducing the size of the tailored SVD basis and cannot find the optimal sensor deployment locations. Hence, the result provided by SVHT is regarded as the reconstruction target. In this manner, the number of sampling locations, r, can be calculated, and r rows of $\mathbf{\Psi}$, corresponding to the sensor deployment locations should be obtained with maximum information in the spatiotemporal field.

DRLT is a model-based informative sampling technique that finds near-optimal sampling locations for spatiotemporal field estimation. Proposition 4.1 shows that the sampling locations selected by DRLT can near-optimally represent the field.

Proposition 4.1. *DRLT generates sampling locations that near-optimally condition the representation of the spatiotemporal field for spatiotemporal reconstruction.*

Proof. Proposition 3.2 shows that the objective function of DRLT is submodular. The diminishing returns property of the submodular function guarantees a near-optimal solution in the case of greedy maximization/minimization [107].

Moreover, as shown in algorithm 3.2, DRLT greedily selects target sensor deployment locations. Hence, as the learning tree expands, sufficient samples are observed in the field, and the greedy selection guarantees that the resulting sampling set is near-optimal. The objective function finds the sensor deployment set that seeks the lowest estimation error

during the greedy selection process. At the same time, DRLT encourages a sensor to take more samples in the area with higher information gain. Therefore, the mutual information between the sampling locations and the rest of the field is maximized. Thus, DRLT can generate sampling locations that near-optimally represent and reconstruct the spatiotemporal field by collecting the most information from the spatiotemporal field.

□

In this manner, r sampling locations selected by DRLT can be converted into γ to construct the canonical measurement matrix $\mathbf{C}$. Given the observations at these locations, the spatiotemporal field can be reconstructed.

The proposed sparse learning method for spatiotemporal field reconstruction is given in Algorithm 4.1. Line 1 initializes the algorithm by pre-processing the training data, where the training data can be obtained from the environmental model. Mean normalization is carried out to facilitate faster learning. Line 2 learns the training data by decomposing it into orthonormal matrices using SVD. Line 3 to line 7 project the calculated near-optimal sampling set ($\mathcal{N}$) into the corresponding locations (γ) in the ϕ space, where $\mathcal{N}$ is generated by DRLT using Algorithm 3.2. Next, γ is organized as the canonical matrix $\mathbf{C}$, and the low-dimensional representation of the orthonormal singular vector $\mathbf{\Psi}$ is extracted to finish the training process. Last, line 10 performs the spatiotemporal reconstruction from limited observation $\mathbf{y}$ to $\hat{\phi}$.

In this way, the spatiotemporal field can be reconstructed by using the observations from the calculated sparse sampling locations.

4.4 Simulation

This section presents the simulation results of the proposed model-based sensor deployment approach for spatiotemporal field reconstruction, using a National Oceanic and Atmospheric Administration (NOAA) dataset. The proposed algorithm is compared with state-of-the-art informative sensor deployment methods, and the reconstruction performance is

Algorithm 4.1: Sparse Learning from Training Data.

Input: $\mathcal{N}, \mathbf{\Phi}_{train}, \mathbf{y}$
Output: $\hat{\phi}$

Init.;
$\mathbf{\Psi}, \mathbf{S}, \mathbf{V}$ = svd($\mathbf{\Phi}_{train}$);
$\gamma = \emptyset$;
for $i = 1, 2, \cdots, r$ **do**
    $\gamma_i \overset{\Psi}{\leftarrow} \mathcal{N}_i$;
    $\gamma = [\gamma, \gamma_i]$;
end
$\mathbf{C} \leftarrow$ canonical(γ);
$\mathbf{\Psi}_r \leftarrow \mathbf{C} \cdot \mathbf{\Psi}$;
$\hat{\phi} \leftarrow \mathbf{\Psi}_r(\mathbf{C}\mathbf{\Psi}_r)^{-1}\mathbf{y}$;

analyzed based on the mean square error (MSE) between the reconstructed field and the ground truth.

4.4.1 Experimental Setup

The NOAA dataset covers the southeast Americas Seas region, which includes the Gulf of Mexico and the Caribbean Sea [83]. The daily sea surface temperature in 2017 was extracted from this dataset to model the environment.

The performance target and the benchmark algorithms for informative sensor deployment are as follows:

- **SVHT**. Singular value hard thresholding (SVHT) finds optimal thresholds for the dimensions of the tailored SVD basis that guarantee minimal signal reconstruction error [114]. The SVHT compression and reconstruction method reduces only the size of the SVD basis, and the resulting tailored SVD basis is used to represent a higher-dimensional space. Therefore, SVHT cannot find the sampling locations because it requires the state to be fully observable and encodes the original signal to another space with a lower dimension. The result generated by SVHT is regarded as the performance target.

- **RRC**: Rapidly-exploring random cycles (RRC) is an RRT-based method that generates sampling locations with the lowest estimation error along with a cycled path within the space-filling tree [3]. The cycled sampling locations can be used to perform periodic sampling using a single robot.

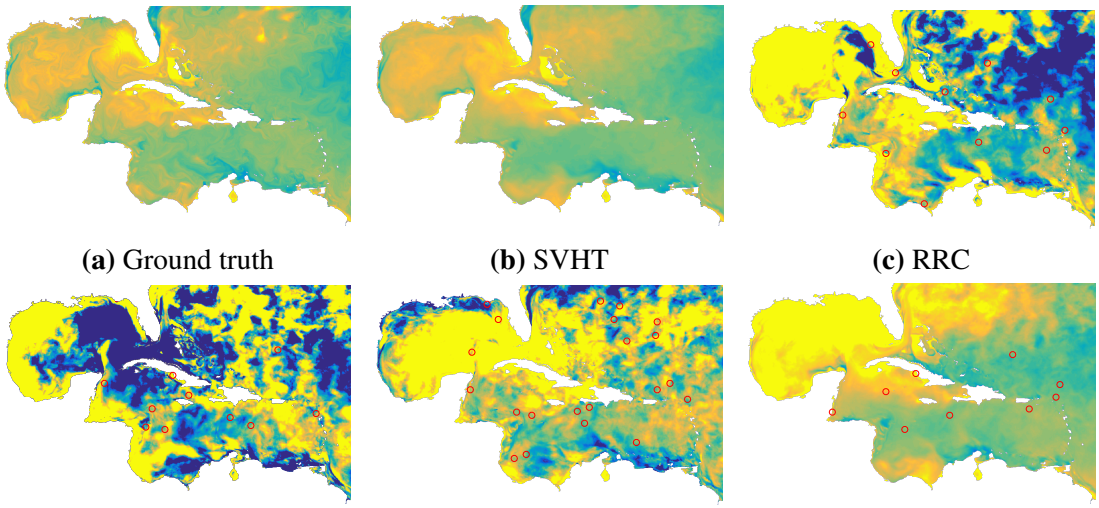
- **INFO**: The approach of informativeness maximization is utilized to find sampling locations with maximum mutual information in the field [99]. As more samples are taken in the spatiotemporal field, a near-optimal sampling set that maximizes the mutual information is generated.

- **RRTPI**: RRTPI is a reinforcement learning-based RRT method that efficiently solves geodesic-based exploration [13]. Ten locations with the highest mutual information are selected in the spatiotemporal field, and the exploring tree has a higher probability of exploring areas with higher information gain and efficiently finding the sampling locations with the lowest estimation error.

4.4.2 Simulation Results

Figure 4.1 presents the ground truth of a spatiotemporal field for testing as well as the reconstruction results. The red circles in the figure indicate the informative sampling locations calculated by the corresponding planning algorithms. SVHT generates the best reconstruction result as it tailors only the information that is less significant in the SVD basis. The input signal is compressed and reconstructed via an orthogonal projection with a tailored SVD basis.

RRC and RRTPI collect only limited information and have the worst reconstruction performance as they focus merely on minimizing the estimation error of the spatiotemporal field. The MSEs for the field reconstruction of RRC and RRTPI are 39.37 and 64.17, respectively. As these two methods do not optimize the information gain in the field, sensor nodes may be placed in locations that provide less effective observations. Some spatial locations may reduce the estimation error by finding abrupt changes in the field but not help



(a) Ground truth **(b)** SVHT **(c)** RRC

(d) RRTPI **(e)** INFO **(f)** DRLT

Figure 4.1: Ground-truth and reconstruction results from the proposed and benchmark algorithms. (Red circle: sampling locations generated from different information-based planning algorithms.)

to represent the general information in the field.

Note that both INFO and DRLT achieve lower reconstruction error as they maximize the mutual information in the field, which is critical for representing the spatiotemporal field. The MSEs for the field reconstruction of INFO and DRLT are 9.08 and 2.59, respectively. INFO captured more information in the southern Caribbean Sea and performed better in reconstruction than RRC and RRTPI. However, the reconstruction in the northern Caribbean Sea and the Gulf of Mexico is rather poor due to the inadequateness of information is captured. DRLT leverages the minimization of the estimation error and the maximization of information gain to outperform the benchmarks in reconstruction. As shown in Figure 4.1 (f), the reconstruction in both southern and northern Caribbean Sea is better than in the benchmark algorithms. The Gulf of Mexico is the only location where information is missing as no sampling locations exist.

Table 4.1 indicates the MSE for the reconstruction using the proposed method and the benchmark algorithms. The target reconstruction performance, generated by SVHT, is 0.12. DRLT produces the sampling locations with lower MSE when compared to the benchmark

Table 4.1: MSE of the proposed method and the benchmark algorithms.

SVHT*	RRC	RRTPI	INFO	DRLT
0.12	39.37	64.17	9.08	2.59

algorithms.

4.5 Summary

This chapter proposed a model-based sensor deployment strategy for spatiotemporal field reconstruction. The proposed model-based sensor deployment strategy significantly outperformed the benchmark algorithms in reconstructing a spatiotemporal field with limited observations. However, the reconstruction accuracy of the proposed method was high compared with the target performance generated by SVHT. Therefore, the reconstruction of a spatiotemporal field has room for improvement. In general, model-based sampling methods may result in inaccuracy when the environmental model cannot adequately express a highly dynamic field of interest. Hence, model-free approaches, such as DL approaches, should be utilized to improve the spatiotemporal reconstruction accuracy.

Chapter 5

Model-free Sensor Deployment for Spatiotemporal Field Reconstruction and Prediction

5.1 Introduction

Chapter 4 introduced the linear reconstruction of the spatiotemporal field from the observations. Linear reconstruction may cause some inaccuracies in capturing spatiotemporal mappings between the observations and the overall spatiotemporal field. In this chapter, nonlinear spatiotemporal mappings are investigated to optimize the sampling location and improve the spatiotemporal reconstruction accuracy.

Optimal spatiotemporal field reconstruction usually leverages a latent low-dimensional representation, which empowers sparse sampling. In this context, it is clearly advantageous to deploy a small number of sensors optimally when estimating and predicting a high-dimensional systems. The optimal sensor deployment is intractable using brute-force approaches, and it is shown to be NP-hard.

Typically, sensor locations are chosen using convex optimization. Model-based sampling approaches have been utilized for deploying sensors near-optimally to estimate moderate-

size spaces [3, 9, 115]. Also, robotic sensors can be deployed to sample and estimate an environment [116–119]. Moreover, probabilistic approaches have been proposed to handle the trade-off between exploration and exploitation in an increasingly generic manner [120, 121]. Their objective is formulated based on information theory, which enables a robotic sensor to sample unexplored areas in the field at higher information gain. However, these approaches mainly focus on simultaneous localization and mapping (SLAM) tasks, which are limited to geometric relationships. In general, model-based convex optimization methods highly rely on the environmental model and use a greedy approach only to navigate robots to the most informative sampling locations. Hence, model-based near-optimal sensor deployment methods may entrap at local minima and result in less accurate field reconstruction and prediction.

Recently, data-driven approaches, such as principal component analysis (PCA) and compressive sensing (CS), have been developed to determine the primary locations for optimal or near-optimal sensor deployment. Data-driven approaches are of low complexity and are easier to implement than model-based convex optimization approaches. Observations obtained from the sampling locations can be used to reconstruct the entire spatiotemporal field. In contrast to CS and PCA systems, which employ sparse linear representations for signal reconstruction, a deep learning (DL) framework can support both linear and nonlinear measurement reconstruction and provide better reconstruction performance. Besides, the nonlinear activation function of a DL framework reduces the likelihood of gradient vanishing.

In this chapter, a novel DL framework called Spatiotemporal sparse Auto FiEld Reconstructor (SAFER) is developed to optimize the sensor deployment locations for spatiotemporal field reconstruction and prediction. The spatiotemporal field is encoded into sparse sampling locations, where key deployment locations are selected as the low-rank representation of the entire spatiotemporal field. Then, an auto-reconstructor is used to capture nonlinear spatiotemporal correlations between the sparse sampling locations and the rest of

the field. The observations in the selected locations can then be decoded to the entire spatiotemporal field. The proposed DL framework is proven to be Lipschitz continuous, and it is adequate for reconstructing and predicting the spatiotemporal field via observations in a minimal number of sampling locations.

5.2 Preliminary and Problem Formulation of Sensor Selection

There are $\binom{m}{r} = \frac{m(m-1)...(m-r+1)}{r(r-1)...1}$ possible selections of r sensors in an m-dimensional environment. It has been shown that recovering the corresponding r-sparse signal from $r \ll m$ observations is an NP-hard problem [122]. The PCA and CS methods have shown their capability to recover high-dimensional signals from an unknown content. However, they are limited to the analysis of linear correlations between the low-dimensional representations and the high-dimensional space. Besides, the spatiotemporal pattern of the signal can be obtained by extracting dominating features from the training set. Hence, a higher reconstruction performance can be achieved by learning the nonlinear mappings in the specific domain of interest.

Typically, sparse sampling can reconstruct a full signal from a small subset of measurements. Denote the physical phenomenon of an Environment of Interest (EOI) as the high dimensional state $\boldsymbol{\phi} \in \mathbb{R}^m$. The nonlinear spatiotemporal dynamics of $\boldsymbol{\phi}$ can be captured by the low dimensional observations for prediction and control. Hence, $\tilde{\boldsymbol{\phi}}$ can be reconstructed from its low-rank representations based on sparse sampling that optimizes the spatiotemporal field reconstruction:

$$\tilde{\boldsymbol{\phi}} = \mathcal{G}_r(\mathbf{y}), \tag{5.1}$$

where $\mathcal{G}_r : \mathbb{R}^r \rightarrow \mathbb{R}^m$ is the reconstruction function, and $\mathbf{y} \in \mathbb{R}^r$ is the sparsely sampled data from $\boldsymbol{\phi}$. Therefore, the principal task of the proposed work is to design a measurement

matrix $\mathbf{C} \in \mathbb{R}^{r \times m}$ to sparsely sample critical observations from $\boldsymbol{\phi}$:

$$\mathbf{y} = \mathbf{C} \cdot \boldsymbol{\phi}, \tag{5.2}$$

where $r \ll m$ is the number of measurements.

Substituting Equation 5.1 into Equation 5.2 yields:

$$\tilde{\boldsymbol{\phi}} = \mathcal{G}_r(\mathbf{C} \cdot \boldsymbol{\phi}) = \mathcal{G}(\boldsymbol{\phi}), \tag{5.3}$$

where $\mathcal{G}$ is the DL framework used for reconstructing the spatiotemporal field with sparsely sampled measurements. The goal in this work is to optimize sensor placement locations within $\boldsymbol{\phi}$, which correspond to the measurement matrix $\mathbf{C}$.

5.2.1 Sparse Sensor Deployment Analysis

In sparse sensor deployment, the spatiotemporal field is only observable at the locations that have deployed sensors. Then, a model is trained to reconstruct the entire spatiotemporal field with limited observations by investigating their nonlinear mappings.

A high-dimensional system $\boldsymbol{\phi} \in \mathbb{R}^m$ may be expressed as the linear/nonlinear combinations of r orthogonal eigenmodes. Hence, high-dimensional systems can be projected into the lower-dimensional subspace. Then, high-dimensional system at time i, $\boldsymbol{\phi}_i \in \mathbb{R}^m$, can be represented according to Equation 5.1:

$$\boldsymbol{\phi}_i \approx \tilde{\boldsymbol{\phi}}_i = \mathcal{G}_r(\mathbf{y}_i), \tag{5.4}$$

where $\mathbf{y}_i$ are the observations from the deployment locations and $\mathcal{G}_r$ is the time independent auto-reconstructor. The temporal coefficient $\mathbf{y}_i$ varies with time i. Therefore, $\mathcal{G}$ can be can be trained from the sampled data, given the measurements $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots \boldsymbol{\phi}_M]$ for M

snapshots:

$$\tilde{\mathbf{\Phi}} = \mathcal{G}(\mathbf{\Phi}, \hat{\theta}), \tag{5.5}$$

where $\hat{\theta}$ stands for the hyper-parameters.

The model is trained to minimize the difference between the training data and the model output:

$$\mathbf{\Phi}^* = \arg\min_{\tilde{\mathbf{\Phi}}} \left\| \mathbf{\Phi} - \tilde{\mathbf{\Phi}} \right\|_2, \tag{5.6}$$

where $\mathbf{\Phi}^*$ is the reconstructed signal and $\|\cdot\|_2$ is the ℓ_2 norm. The model $\mathcal{G}$ that generates the reconstructed signal can be used for predicting the future spatiotemporal field, given the sampling data at that time.

In this manner, the problem of sparse sensor deployment can be formulated as:

$$\begin{aligned} &\min \sum_{\boldsymbol{\phi} \in \mathbf{\Phi}} \left\| \boldsymbol{\phi} - \mathcal{G}(\boldsymbol{\phi}, \hat{\theta}) \right\|_2, \\ &s.t. \ |\mathbf{C} \cdot \boldsymbol{\phi}| = r, \end{aligned} \tag{5.7}$$

where $|\cdot|$ is the cardinality of the selected sampling locations.

5.3 Sparse Sensor Deployment using Auto Field Reconstructor

The proposed approach incorporates a DL framework to learn spatiotemporal correlations in the training data, and determines the critical sampling locations that optimize the spatiotemporal field reconstruction. It is a model-free approach, which does not need to model the environment with prior knowledge.

5.3.1 Sparse Sampling for Reconstruction

In this chapter, a high-dimensional signal is assumed to be reconstructed from a small subset of point observations. Therefore, the sensor deployment locations are optimized to achieve better reconstruction performance. Denote the spatiotemporal field as an m-

dimensional space:

$$\boldsymbol{\phi} = [\phi_1 \phi_2 \cdots \phi_m]^T, \tag{5.8}$$

where $\phi.$ is a random variable that represents the sampling value at the corresponding location.

Then, the measurement matrix $\mathbf{C}$ can be used to select the sampling locations:

$$\mathbf{C} = [\mathbf{e}_{\gamma_1} \mathbf{e}_{\gamma_2} \cdots \mathbf{e}_{\gamma_r}]^T, \tag{5.9}$$

where $\mathbf{e}_j \in \mathbb{R}^m$ are the canonical basis vectors.

According to Equation 5.2, observations in the sensor placement locations can be expressed as a linear combination of the canonical basis and the high dimensional input data:

$$y_i = \mathbf{e}_{\gamma_i} \boldsymbol{\phi}$$

Since matrix $\mathbf{C}$ is canonical, the observation $\mathbf{y}$ can be simplified as:

$$\mathbf{y} = [\phi_{\gamma_1} \phi_{\gamma_2} \cdots \phi_{\gamma_r}]^T, \tag{5.10}$$

where $\gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_r\} \subset \{1, 2, \cdots, m\}$ is the set of indices for the selected sampling locations. Hence, the observations in $\mathbf{y}$ are downsampled from the m-dimensional spatiotemporal field. Then, as $\mathbf{y}$ can be directly observed from the spatiotemporal field, the spatiotemporal field $\tilde{\boldsymbol{\phi}}$ can be reconstructed given the observations at the sampling locations and the nonlinear mapping $\mathcal{G}_r$.

A schematic diagram for sparse sensor deployment is illustrated in Figure 5.1. The observations at the sampling locations $\mathbf{y}$ are selected by the measurement matrix $\mathbf{C}$ to guarantee the best feasible reconstruction $\tilde{\boldsymbol{\phi}}$.

The sensor deployment locations in the spatiotemporal field correspond to the sensor readings in the input matrix $\boldsymbol{\phi}$. Hence, the sparse sensor deployment strategy is to compute

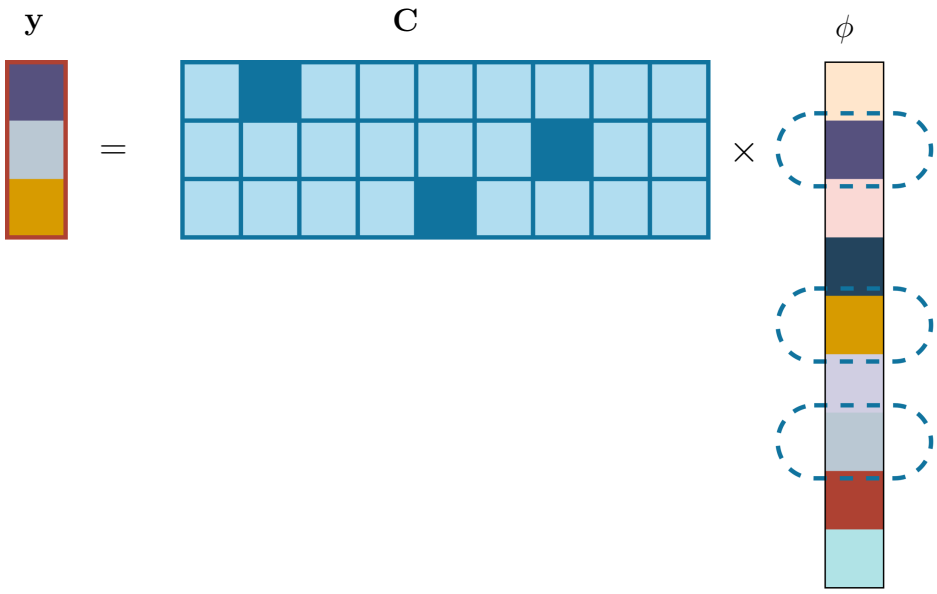


Figure 5.1: Illustration of generating sparse sampling locations using a canonical basis. Dark blue region in matrix $\mathbf{C}$ stands for 1, rest of the matrix contains 0; second, seventh and fifth row of $\boldsymbol{\phi}$ is extracted according to $\mathbf{C}$ and compose $\mathbf{y}$.

the rows of $\boldsymbol{\phi}$ that optimally represent the spatiotemporal correlation in the field. $\mathcal{G}_r$ should be trained to learn spatiotemporal correlations between the observation $\mathbf{y}$ and $\boldsymbol{\phi}$.

5.3.2 Deep Learning based Spatiotemporal Field Reconstruction

According to Equation 5.1, input signal $\boldsymbol{\phi}$ can be compressed as $\mathbf{y} = \mathbf{C}\boldsymbol{\phi}$. Hence, the input training dataset can be expressed as $\boldsymbol{\Phi}$ and the corresponding compressed signal is $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2 \dots \mathbf{y}_M]$. Given a suitable canonical matrix $\mathbf{C} \in \mathbb{R}^{r \times m}$, the input data matrix can be compressed to key observations:

$$\mathbf{Y} = \mathbf{C}\boldsymbol{\Phi} \tag{5.11}$$

Then, the training set for the proposed deep learning framework has M pairs of the input signal and the corresponding observations in the selected locations:

$$\mathcal{S}_{train} = \{(\mathbf{y}_1, \boldsymbol{\phi}_1), (\mathbf{y}_2, \boldsymbol{\phi}_2), \cdots, (\mathbf{y}_M, \boldsymbol{\phi}_M)\}.$$

The nonlinear signal reconstruction mapping is learned from this training set. Similarly,

the test set $\mathcal{S}_{test}$ contains several pairs of signals and observations.

Traditional sparse reconstruction methods show that using greedy and iterative approaches provide better performance by using the matrix-vector multiplication with the computational cost of $O(nr)$ [56]. Besides, fully connected layers suffer from the curse of dimensionality and consume vast system memory [123]. For example, the spatiotemporal field in the training data is a 360×180 image, and there are 44219 non-obstacle pixels in each image. Then, the weights between two fully connected layers will be stored in a 44219-by-44219 matrix using float 32, which consumes a massive amount of memory and it is hard to train. Specifically,

$$\begin{aligned} \text{memory consumption} &= 44219^2 * 32 \text{ bits} \\ &= 62570238752 \text{ bits} \\ &= 7.82 \text{ GB}. \end{aligned} \tag{5.12}$$

Therefore, a stacked structure is utilized to save memory, where each layer of the DL framework has m inputs and r outputs or vice versa.

Then, a multi-layer DNN is used to learn the signal reconstruction from limited sampling points. In the beginning, the input signal is compressed to r-dimension using a logical matrix with the canonical basis:

$$\mathbf{y} = \mathbf{C} \cdot \boldsymbol{\phi}. \tag{5.13}$$

Then, each hidden layer can be represented as

$$\mathbf{x}_{h_{i+1}} = \mathcal{T}(\mathbf{W}_{h_i}\mathbf{x}_{h_i} + \mathbf{b}_{h_i}), \tag{5.14}$$

where $\mathbf{W}_{h_i} \in \mathbb{R}^{m \times r}, \mathbf{b}_{h_i} \in \mathbb{R}^m$ or $\mathbf{W}_{h_i} \in \mathbb{R}^{r \times m}, \mathbf{b}_{h_i} \in \mathbb{R}^r$, depending on the size of the input data.

Besides, $\mathbf{x}$ denotes the hidden neurons and $\mathcal{T}(\cdot)$ is the element-wise nonlinear rectifier function. Note that $\mathbf{x}_{h_0}$ is the input of the hidden layers, $\mathbf{y}$. Then, the calculation for the

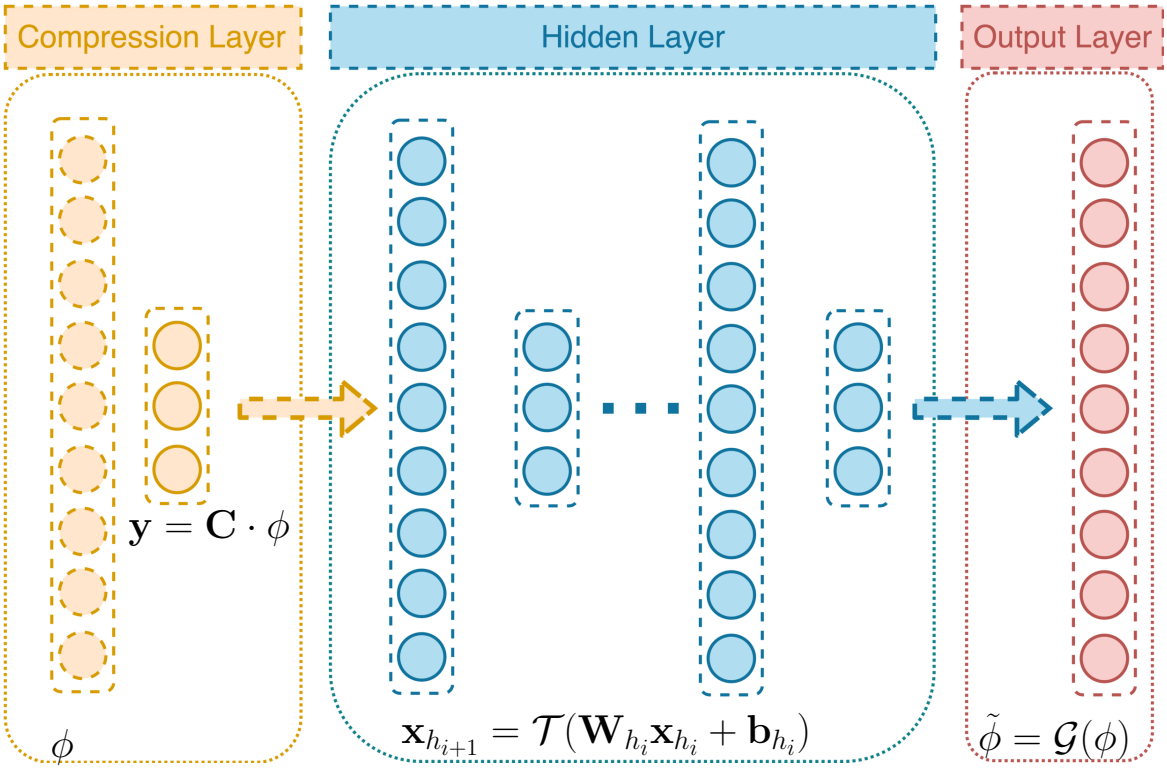


Figure 5.2: The framework of the auto field reconstructor. Yellow nodes represent the compression layers which optimize sparse sampling locations; blue nodes are neurons in the hidden layer for model fitting; red layer is the output layer.

output layer is given by:

$$\tilde{\boldsymbol{\phi}} = \mathcal{T}(\mathbf{W}_o \mathbf{x}_h + \mathbf{b}_o), \tag{5.15}$$

where $\mathbf{W}_o \in \mathbb{R}^{r \times m}$ and $\mathbf{b}_o \in \mathbb{R}^m$ are the weighting matrix and the bias of last hidden layer, respectively. As the size of the output layer must match the dimension of the environment, the number of hidden layers must be even.

Denote hyper-parameters as $\hat{\theta} = \{\mathbf{C}, \mathbf{W}_{h_1}, \mathbf{b}_{h_1}, \mathbf{W}_{h_2}, \mathbf{b}_{h_2}, \cdots, \mathbf{W}_o, \mathbf{b}_o\}$. The nonlinear mapping from the input training data $\boldsymbol{\phi}$ to the output data $\tilde{\boldsymbol{\phi}}$ can be represented as $\tilde{\boldsymbol{\phi}} = \mathcal{G}(\boldsymbol{\phi}, \hat{\theta})$. The the mean squared error (MSE) is used as the loss function:

$$\mathcal{L}(\mathcal{G}, \mathbf{C}) = \frac{1}{M} \sum_{i=1}^{M} \left\| \mathcal{G}(\boldsymbol{\phi}_i, \hat{\theta}) - \boldsymbol{\phi}_i \right\|_2^2 \tag{5.16}$$

The framework of the proposed model is shown in Figure 5.2. Two-dimensional images

of the spatiotemporal field are reshaped into a one-dimensional vector at size m and fed into the input layer. Then, r key sampling locations, $\mathbf{y}$, are selected via the canonical measurement matrix $\mathbf{C}$. Next, $\mathbf{y}$ is passed through the multi-layer stacked auto-reconstructor to reconstruct the spatiotemporal field ($\tilde{\phi}$). All hidden layers are applied with nonlinear Rectified Linear Units (ReLU). Last, Adam optimization is utilized to compare the difference between $\tilde{\phi}$ and ϕ, and minimize the loss.

Once the training process minimizes the difference between the model output and the input data, the auto field reconstructor learns the spatiotemporal correlation between the selected deployment locations and the entire spatiotemporal field. Hence, once the sensor samples new data after the deployment, the entire spatiotemporal field can be reconstructed.

Next, it is essential to optimize $\mathbf{C}$ matrix for sparse sampling. Finding the best sampling locations helps to train optimal decoders for the best reconstruction performance. Algorithm 5.1 explains how the sparse sampling locations are optimized during the model training process for reconstruction.

Algorithm 5.1: SAFER

Input: $\mathbf{\Phi}$
Output: $\mathcal{G}(\cdot)$, $\mathbf{C}$
Initialization: set measurement matrix $\mathbf{C}$ as a random canonical matrix, and randomly draw hyper-parameters $\hat{\theta}$ from uniform distributions with a zero mean.;
for *1 to* T **do**
    Shuffle m input data;
    Use Adam optimization to minimize $\mathcal{L}(\mathcal{G}, \mathbf{C})$;
    $\hat{\mathbf{C}}$ = swap($\mathbf{C}$);
    if $\mathcal{L}(\hat{\mathcal{G}}, \hat{\mathbf{C}}) < \mathcal{L}(\mathcal{G}, \mathbf{C}) || rand() < \rho$ **then**
        $\mathbf{C} = \hat{\mathbf{C}}$;
        $\mathcal{G} = \hat{\mathcal{G}}$;
    end
end

Initially, a random measurement matrix $\mathbf{C}$ is generated to randomly select the sensor deployment locations. Also, the hyper-parameters are initialized. Then, the training process iterates for T times to optimize the deployment strategy for the spatiotemporal field

reconstruction. In each iteration, m input data are shuffled to reduce the variance and ensure the model to be general and has less over-fit. Line 4 minimize the loss according to the model indicated in Figure 5.2. Line 5 to line 10 seek to find a better sensor deployment strategy in each iteration. Line 5 swaps one non-zero column in the matrix $\mathbf{C}$ with one zero column to search for better sensor deployment strategies by introducing new observations for the model. Line 6 checks if the updated measurement matrix and the model will result in a lower loss. If new observations can achieve better performance, the updated model will be able to provide overall better performance. This scheme not only achieves a better model but also adopts a worse model according to a slowly decreasing probability ρ. Allowing worse solutions provides the capability to avoid local minima in simulated annealing approaches [124].

5.3.3 Lipschitz Continuity and Reconstruction Capability for the Proposed Framework

Lipschitz continuity is a strong property of function continuity in mathematical analysis. It limits the change rate of the function and bounds the slope of the function within the Lipschitz constant. Lipschitz continuity also guarantees that the function obeys the Cauchy-Lipschitz Theorem, which ensures the existence and uniqueness of solutions to certain first-order equations [125]. Definition 5.1 and Proposition 5.1 show that the proposed model $\mathcal{G}$ is L-Lipschitz. Besides, Definition 5.2 and Lemma 5.1 show that r samples are sufficient for spatiotemporal field reconstruction.

The definition of Lipschitz continuity can be formulated as follows:

Definition 5.1. A function, $\mathcal{G} : \mathbb{R}^r \rightarrow \mathbb{R}^m$, is Lipschitz continuous if there exists a positive real constant L such that:

$$\left\|\mathcal{G}(\mathbf{x}_1) - \mathcal{G}(\mathbf{x}_2)\right\|_2 \leq L \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|_2, \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^r, \tag{5.17}$$

where the value of L is the Lipschitz constant, and the function $\mathcal{G}$ is L-Lipschitz.

Then, as shown in Proposition 5.1, the Lipschitz continuity of the proposed network can be proved layer-by-layer and the corresponding Lipschitz constant can be calculated.

Proposition 5.1. *$\mathcal{G}$ is L-Lipschitz.*

Proof. The Lipschitz constant of the proposed framework can be computed as a series of function compositions:

$$\mathcal{G}(\boldsymbol{\phi}) = (\mathcal{G}_1 \circ \mathcal{G}_2 \circ \cdots)(\mathbf{x}), \tag{5.18}$$

where $\mathcal{G}_i$ can be the linear operation between layers and activation function. Denote the Lipschitz constant of the function as $L(\cdot)$. Then, Lipschitz constant of the functional compositions can be expressed as:

$$L(\mathcal{G}) = \prod L(\mathcal{G}_i). \tag{5.19}$$

In this manner, the Lipschitz constant of $\mathcal{G}$ can be calculated layer by layer. Denote a hidden layer with r inputs as

$$\mathcal{G}_h(\boldsymbol{\phi}) = \mathbf{W}\mathbf{x} + \mathbf{b}. \tag{5.20}$$

The Lipschitz constant is given by the spectral norm of the weighting matrix under the ℓ_2 norm [126, 127]. Substitute Equation 5.20 in to Equation 5.17:

$$\begin{aligned} \|(\mathbf{W}\mathbf{x}_1 + \mathbf{b}) - (\mathbf{W}\mathbf{x}_2 + \mathbf{b})\|_2 &\leq L \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \\ \|\mathbf{W}\boldsymbol{\alpha}\|_2 &\leq L \|\boldsymbol{\alpha}\|_2 \\ \frac{\|\mathbf{W}\boldsymbol{\alpha}\|_2}{\|\boldsymbol{\alpha}\|_2} &\leq L, \end{aligned} \tag{5.21}$$

where $\boldsymbol{\alpha} = \mathbf{x}_1 - \mathbf{x}_2$. Hence, the smallest Lipschitz constant can be derived as the operator norm:

$$L(\mathcal{G}_h) = \sup_{\boldsymbol{\alpha} \neq 0} \frac{\|\mathbf{W}\boldsymbol{\alpha}\|_2}{\|\boldsymbol{\alpha}\|_2}. \tag{5.22}$$

In the case of ℓ_2 norm, the operator norm can be computed by using the largest singular

value of the weighting matrix $\mathbf{W}$ [78, 126].

The $\mathbf{W}$ is initialized with Gaussian random variables. Then $\mathbf{W}^T\mathbf{W}$ is its Wishart ensemble. The expectation of the largest eigenvalue of $\mathbf{W}^T\mathbf{W}$ is $4r$ according to [128–130]. Hence, the largest singular value of $\mathbf{W}$ is the square root of $\mathbf{W}^T\mathbf{W}$, which is $2\sqrt{r}$. Thus, $\mathcal{G}_h(\cdot)$ is Lipschitz continuous and its Lipschitz constant is $2\sqrt{r}$. Similarly, the hidden layer with m inputs has a Lipschitz constant of $2\sqrt{m}$.

The Lipschitz constant of the compression layer can also be calculated similarly. The smallest Lipschitz constant of the compression layer is the largest singular value of the logical matrix with the canonical basis:

$$L(\mathcal{G}_c) = \sup_{\boldsymbol{\alpha}\neq 0} \frac{\|\mathbf{C}\boldsymbol{\alpha}\|_2}{\|\boldsymbol{\alpha}\|_2}. \tag{5.23}$$

The singular values of the matrix $\mathbf{C}$ are the square roots of the eigenvalues of $\mathbf{C}^T\mathbf{C}$. Denote λ as the eigenvalue of $\mathbf{C}^T\mathbf{C}$ and $\mathbf{v} = [v_1 v_2 \cdots v_m]$. Then:

$$\begin{aligned} \lambda\mathbf{v} &= \mathbf{C}^T\mathbf{C}\mathbf{v} \\ &= \mathbf{E}\mathbf{v}, \end{aligned} \tag{5.24}$$

where $\mathbf{E} = diag(e_1, e_2, \cdots, e_m), e_i \in \{0, 1\}$ is a diagonal identity matrix.

Therefore,

$$(\lambda - e_i)v_i = 0. \tag{5.25}$$

The eigenvalue of $\mathbf{C}^T\mathbf{C}$ is either 0 or 1, thus the Lipschitz constant of compress layer (largest singular value of matrix $\mathbf{C}$) is $L(\mathcal{G}_c) = 1$.

The Lipschitz continuity of the activation function can be analyzed in a similar manner. In this chapter, ReLU is used as the activation function:

$$ReLU(x) = \max(0, x), \tag{5.26}$$

whose derivative is:

$$ReLU'(x) = \begin{cases} 0, & \text{if } x > 0 \\ 1, & \text{otherwise.} \end{cases} \tag{5.27}$$

Hence, the maximum gradient of the ReLU function is 1, which indicates that ReLU is Lipschitz continuous and the Lipschitz constant of ReLU is $L(\mathcal{G}_{a_i}) = 1$.

In this manner, the Lipschitz constant for the whole network can be derived according to Equation 5.19:

$$\begin{aligned} L(\mathcal{G}) &= \prod L(\mathcal{G}_{h_i}) \times \prod L(\mathcal{G}_{a_i}) \times L(\mathcal{G}_c) \\ &= \prod L(\mathcal{G}_{h_i}) \\ &= (4\sqrt{mr})^{d/2}, \end{aligned} \tag{5.28}$$

where d is the number of hidden layers. Thus, the proposed model $\mathcal{G}$ follows Lipschitz continuity. □

Next, it is essential to show that the sparsely sampled observations are sufficient to reconstruct a spatiotemporal field. In sparse sampling and reconstruction, Restricted Eigenvalue Condition (REC) is a sufficient condition for robust reconstruction. REC requires all sparse vectors $\boldsymbol{\phi}$ to be far from the null-space of matrix $\mathbf{C}$:

$$\|\mathbf{C}\boldsymbol{\phi}\| \geq \gamma \|\boldsymbol{\phi}\|, \tag{5.29}$$

where γ is a constant greater than 0. It has been proved that the functions satisfying the REC is sufficient for reconstruct sparse signals, such as Lasso [131, 132] and Elastic Net [133].

In this chapter, a restricted version of REC is proposed to fit the sparse sample reconstruction problem, called Canonical Restricted Eigenvalue Condition (CREC).

Definition 5.2. Denoting $\mathcal{G} \in \mathbb{R}^r$, a canonical matrix $\mathbf{C}$ is said to satisfy CREC$(\mathcal{G}, \gamma, \delta)$, if

$\forall x_1, x_2 \in \mathcal{S}$ and $\exists \gamma > 0, \delta \geq 0$,

$$\|\mathbf{C}(x_1 - x_2)\|_2 \geq \gamma \|x_1 - x_2\|_2 - \delta.$$

The CREC definition restricts that for any two signals in $\mathcal{G}$, the corresponding compressed observations should have at least the same significance as the original signal. Therefore, there is a possibility to recover the unknown vector from these observations. A canonical matrix $\mathbf{C}$ is used to perform sparse sampling for signal reconstruction rather than the iid random matrix used for signal compression in [77].

Proposition 5.1 has shown that the proposed framework for signal reconstruction is L-Lipschitz. Then, following lemmas show that the canonical matrix sampling r observations will satisfy the CREC for sparse signal reconstruction given $\mathcal{G}_r : \mathbb{R}^r \to \mathbb{R}^m$.

Lemma 5.1. *Denote $B(r) = \{\mathbf{y} | \mathbf{y} \in \mathbb{R}^r\}$ as the ℓ_2-norm ball in $\mathbb{R}^r$. A random canonical matrix, $\mathbf{C}$, satisfies CREC$(\mathcal{G}_r(B(r)), \gamma, \delta)$, where $\gamma > 0$, $\delta > 0$.*

Proof. Denote $\mathbf{y} \in \mathbb{R}^r$, where $\mathbf{x} = \mathcal{G}_r(\mathbf{y})$, for any $\mathbf{y}, \mathbf{y}' \in B(r)$, the sparse signal reconstruction model is L-Lipschitz according to Definition 5.1. Hence,

$$\|\mathcal{G}_r(\mathbf{y}) - \mathcal{G}_r(\mathbf{y}')\|_2 \leq L \|\mathbf{y} - \mathbf{y}'\|_2 .$$

Denote a net ζ to be the $\frac{\delta}{L}$-net on $B(r)$. Then, $\mathcal{G}_r(\zeta)$ is the δ-cover of $\mathcal{G}_r(B(r))$ according to the L-Lipschitz property.

Then, $\forall \mathbf{y}, \mathbf{y}' \in B, \exists \mathbf{y}_1, \mathbf{y}_2 \in \zeta$, such that $\mathcal{G}_r(\mathbf{y}_1)$ and $\mathcal{G}_r(\mathbf{y}_2)$ are δ-close to $\mathcal{G}_r(\mathbf{y})$ and $\mathcal{G}_r(\mathbf{y}')$, respectively. Therefore, according to triangle inequality,

$$\begin{aligned} \|\mathcal{G}_r(\mathbf{y}) - \mathcal{G}_r(\mathbf{y}')\|_2 \leq & \|\mathcal{G}_r(\mathbf{y}) - \mathcal{G}_r(\mathbf{y}_1)\|_2 + \\ & \|\mathcal{G}_r(\mathbf{y}_1) - \mathcal{G}_r(\mathbf{y}_2)\|_2 + \|\mathcal{G}_r(\mathbf{y}_2) - \mathcal{G}_r(\mathbf{y}')\|_2 \\ \leq & \|\mathcal{G}_r(\mathbf{y}_1) - \mathcal{G}_r(\mathbf{y}_2)\|_2 + 2\delta \end{aligned} \tag{5.30}$$

Similarly,

$$\begin{aligned}\|\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\|_2 \leq &\|\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}\mathcal{G}_r(\mathbf{y})\|_2 + \\ &\|\mathbf{C}\mathcal{G}_r(\mathbf{y}) - \mathbf{C}\mathcal{G}_r(\mathbf{y}')\|_2 + \\ &\|\mathbf{C}\mathcal{G}_r(\mathbf{y}') - \mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\|_2 \\ \leq &\|\mathbf{C}\mathcal{G}_r(\mathbf{y}) - \mathbf{C}\mathcal{G}_r(\mathbf{y}')\|_2 + \mathcal{O}(\delta).\end{aligned} \tag{5.31}$$

Then, for samples in the δ-cover of $\mathcal{G}_r(B(r))$,

$$\begin{aligned}\|\mathcal{G}_r(\mathbf{y}_1) - \mathcal{G}_r(\mathbf{y}_2)\|_2 &= \left\|\mathbf{C}^T\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}^T\mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\right\|_2 \\ &\leq \left\|\mathbf{C}^T\right\|_2 \|\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\|_2 .\end{aligned} \tag{5.32}$$

The ℓ_2 norm, $\left\|\mathbf{C}^T\right\|_2$, is the largest singular value of $\mathbf{C}^T$ according to Lemma 5.2. Hence,

$$\begin{aligned}\frac{\|\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\|_2}{\|\mathcal{G}_r(\mathbf{y}_1) - \mathcal{G}_r(\mathbf{y}_2)\|_2} &\geq \frac{1}{\|\mathbf{C}^T\|_2} \geq 1 \\ \|\mathbf{C}\mathcal{G}_r(\mathbf{y}_1) - \mathbf{C}\mathcal{G}_r(\mathbf{y}_2)\|_2 &\geq \frac{1}{\|\mathbf{C}^T\|_2} \|\mathcal{G}_r(\mathbf{y}_1) - \mathcal{G}_r(\mathbf{y}_2)\|_2 .\end{aligned} \tag{5.33}$$

Next, substitute Equation 5.30 and Equation 5.31 into Equation 5.33:

$$\frac{1}{\|\mathbf{C}^T\|} \|\mathcal{G}_r(\mathbf{y}) - \mathcal{G}_r(\mathbf{y}')\| \leq \|\mathbf{C}\mathcal{G}_r(\mathbf{y}) - \mathbf{C}\mathcal{G}_r(\mathbf{y}')\| + O(\delta). \tag{5.34}$$

Thus, $\mathbf{C}$ satisfies CREC$(\mathcal{G}_r(B(r)), \gamma, \delta)$, where $\gamma = \frac{1}{\|\mathbf{C}^T\|}$. □

Lemma 5.2. *ℓ_2 norm of the canonical matrix $\mathbf{C}$ equals the largest singular value of $\mathbf{C}$.*

Proof. ℓ_2 norm of a matrix can be expressed as the square root of the inner product of the matrix itself and its conjugate transpose:

$$\|\mathbf{C}\|_2 = \sqrt{\mathbf{C}^\dagger\mathbf{C}}. \tag{5.35}$$

Besides, the spectral norm of matrix $\mathbf{C}$, $\|\mathbf{C}\|_2$ can be represented as:

$$\|\mathbf{C}\|_2 = \sup_{\|x\|_2=1} \|\mathbf{C}x\|_2 , \tag{5.36}$$

where x is a temporary variable that assists the proof in this lemma.

Denote $\mathbf{H} = \mathbf{C}^\dagger\mathbf{C}$, which is a Hermitian matrix. Then, iff (if and only if) a linear transformation of Euclidean space $\mathbf{E}$ is Hermite, there exists an orthonormal basis of $\mathbf{E}$ that comprises all the eigenvectors of $\mathbf{H}$. Let $\lambda_1, \lambda_2, \cdots, \lambda_m$ be the eigenvalues of Hermitian matrix $\mathbf{H}$, and $e_1, e_2, \cdots, e_m$ be the orthonormal basis of $\mathbf{E}$.

Also, let $x = a_1e_1 + a_2e_2 + \cdots + a_me_m$. The norm of x is :

$$\|x\| = \langle \sum_{i=1}^{m} a_ie_i, \sum_{i=1}^{m} a_ie_i \rangle^{\frac{1}{2}} = \sqrt{\sum\nolimits_{i=1}^{m} a_i^2} = 1. \tag{5.37}$$

Hence,

$$\begin{aligned} \|\mathbf{C}x\|_2 &= \langle \mathbf{C}x, \mathbf{C}x \rangle = \langle x, \mathbf{C}^\dagger\mathbf{C}x \rangle \\ &= \langle x, \mathbf{H}x \rangle \\ &= \sum\nolimits_{i=1}^{m} \sqrt{\lambda_i} a_i \\ &\leq \sqrt{\lambda_*}, \end{aligned} \tag{5.38}$$

where $\sqrt{\lambda_*}$ as the largest eigenvalue of matrix $\mathbf{H}$. Therefore, $\|\mathbf{C}\|_2 \leq \sqrt{\lambda_*}$.

Besides,

$$\|\mathbf{C}\|_2 \geq \langle x, \mathbf{H}x \rangle = \langle e_*, \lambda_* e_* \rangle = \sqrt{\lambda_*}. \tag{5.39}$$

Thus, $\|\mathbf{C}\|_2 = \sqrt{\lambda_*}$. Also, as the largest singular value of $\mathbf{C}$ is equal to the square root of $\mathbf{H}$, the ℓ_2 norm of the canonical matrix $\mathbf{C}$ equals the largest singular value of $\mathbf{C}$.

□

Therefore, Proposition 5.1 and Lemma 5.1 are proved, and they jointly confirm that the

proposed model $\mathcal{G}$ is L-Lipschitz and is sufficient to reconstruct a spatiotemporal field with r observations.

5.4 Simulation Results

This section presents the conducted numerical simulations of SAFER using data from NOAA. The proposed algorithm is compared with some state-of-the-art algorithms for reconstructing a global spatiotemporal field.

5.4.1 Experimental Setup

Model training

Back-propagation is used to fine-tune the layer weights and biases for SAFER. The proposed network is trained with ADAM optimizer [134] by using the default settings: $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^8$. The learning rate is set to 0.001, and the mini-batch size is set to 20. The input sequence is shuffled to avoid model overfitting and reduce the variance. The dataset is divided into a training set, and a test set according to their sequence in the time series. The beginning part of the time series is regarded as the training set for learning the nonlinear map from the observations to the entire spatiotemporal field. The test set then composes the remaining snapshots of the spatiotemporal field in the time series. Hence, the training error indicates the reconstruction performance of the conducted algorithm, while the test error indicates the prediction performance. In this manner, given the unknown future observations in the deployed location, the ability to predict the future spatiotemporal field is examined.

The simulation was conducted on a PC with 4.0 GHz Quad-Core CPU and 16 G memory. The DL framework was implemented using the TensorFlow framework [111] and run without GPUs.

Dataset

Simulations were conducted using two global climate datasets. The first one is the National Oceanic and Atmospheric Administration (NOAA) global sea surface temperature dataset (SST) spanning from 1990 to 2018, which is publicly available online at [135]. The SST dataset provides weekly global sea surface temperature means in 1.0-degree latitude $\times$ 1.0-degree longitude global grid (180 $\times$ 360) [136]. The second dataset is the NOAA's PRECipitation REConstruction Dataset (PREC) [137]. This dataset provides monthly global precipitation constructed on a 2.5-degree latitude $\times$ 2.5-degree longitude grid over the global (72 $\times$ 144) for the period from Jan. 1948 to Aug. 2018 [138].

Following [4], the data for the first 16 years are selected as the training set, and the remaining data are used as the test set, for the SST dataset. As for PREC dataset, the first 70% of spatiotemporal field scenarios are chosen as the training set, and the remaining 30% scenarios are used as the test set for performance evaluation.

Evaluation Metrics

The proposed framework is evaluated by using two metrics, MSE@N, and VAR@N, where N is the number of deployed sensor nodes. MSE@N is the Mean Square Error that measures the average of the squares of the errors between predictions and the ground truth. It is calculated as:

$$\text{MSE@}N = \frac{1}{m \cdot M} \sum_{i=1}^{m} \sum_{j=1}^{M} (\mathbf{\Phi}_{ij} - \tilde{\mathbf{\Phi}}_{ij})^2. \tag{5.40}$$

VAR@N calculates the mean variance of MSE for all snapshots of the spatiotemporal field. Then, a low VAR@N indicates that the MSE of the predicting test snapshots will not change significantly in most locations. Hence, the model can perform a more generalized reconstruction and have a higher prediction ability.

$$\text{VAR@}N = \frac{1}{m} \sum_{i=1}^{m} (\frac{1}{M} \sum_{j=1}^{M} (\mathbf{\Phi}_{ij} - \tilde{\mathbf{\Phi}}_{ij})^2 - \text{MSE@}N)^2. \tag{5.41}$$

Benchmark Algorithm

The proposed algorithm, SAFER, is compared with the following benchmark algorithms:

- **SVHT**. Singular value hard thresholding (SVHT) is a PCA-based approach that finds optimal thresholds for the dimensions of the tailored SVD basis, guaranteeing minimal signal reconstruction error [114]. The SVHT compression and reconstruction method reduces only the size of the SVD basis, and the resulting tailored SVD basis can be used to represent a higher-dimensional space as a low-dimensional one. The reconstruction is based on the signal that is compressed via the tailored SVD basis, using the left singular vectors and the corresponding orthogonal projections [139, 140]. Therefore, SVHT cannot find the sampling locations because it requires the state to be fully observable and encodes the original signal to another space with a lower dimension. Nevertheless, SVHT-based reconstruction is widely used in statistical signal processing to reduce the size of an original signal. Hence, the result of SVHT is regarded as the target performance.

- **Q-DEIM**. QR factorization with the discrete empirical interpolation method (Q-DEIM) utilizes a greedy approximation solution provided by the matrix QR factorization with column pivoting. The sensor deployment scheme of Q-DEIM seeks rows of $\mathbf{\Psi}_r$, corresponding to the point sensor locations in the spatiotemporal field that will optimally condition the inversion of the measurement matrix [4, 68]. The canonical measurement matrix $\mathbf{C}$ is then extracted from the corresponding sampling locations of $\mathbf{\Psi}_r$.

- **CS**. Compressive sensing (CS) can efficiently compress and reconstruct a signal by solving suitable underdetermined linear systems [57, 58, 141]. Similar to SAFER and Q-DEIM, the signal is compressed via a canonical sparse sensing matrix: $\mathbf{y} = \mathbf{C} \cdot \boldsymbol{\phi}$.

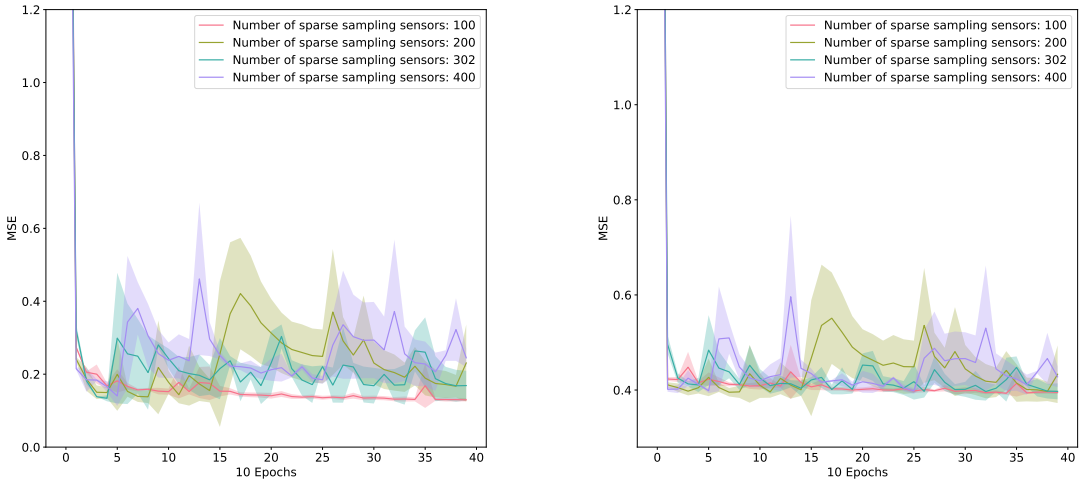
- ℓ_2 **Random**. ℓ_2 Random is a baseline algorithm in which the sensor nodes are randomly deployed in the field. The reconstruction method of ℓ_2 Random is the same

as that of Q-DEIM, and the random seed for generating the sampling locations is set same as the one used in CS. Hence, the sensors are deployed in the same locations as in CS. The spatiotemporal field is reconstructed using the observations from these random locations and the information is learned from the training data.

5.4.2 Training Details

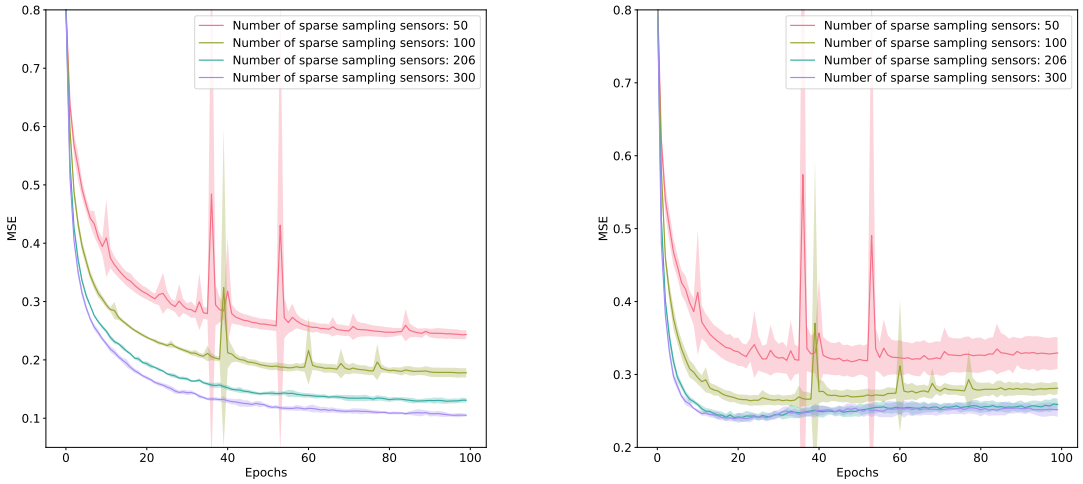
Figure 5.3 presents the reconstruction error in the SST dataset for both the training set and the test set during the model training. Four thresholds for sparse reconstruction are selected, including the optimal threshold for signal reconstruction, calculated by singular value hard thresholding (SVHT) [114]. The optimal hard threshold for singular value reconstruction is 302, and the other thresholds are chosen as 100, 200, and 400. All selected thresholds result in a rapid MSE decrease within a few iterations. All the final training errors are less than 0.3, and the test errors are less than 0.45. The MSE variance during the training process increases as more sensor nodes are deployed in the field. Note that 100 sparse sampling sensors are sufficient to achieve the lowest training error and the second-lowest test error. Moreover, since increasing the number of sampling locations would not result in a significant reconstruction improvement in the test set, 100 sparse sampling locations are sufficient for sparse field reconstruction in this dataset.

Figure 5.4 presents the training and test errors for the PREC dataset. The optimal hard threshold, SVHT, is 206, and the other thresholds are selected as 50, 100, and 300. Figure 5.4 (a) shows that the MSE of the training decreases steadily for all threshold values. Thresholds 206 and 300 achieve the lowest training error, which is close to 0.1, and the corresponding test error, which is approximately 0.25, is significantly lower than the test error for the rest of the threshold values. Furthermore, the training variance is small for thresholds 206 and 300. In these cases, deploying 206 sensors in the field can provide a highly accurate reconstruction for global precipitation reconstruction while using the smallest number of sensor nodes.



(a) Training error. **(b)** Test error.

Figure 5.3: Training using SST dataset with 100, 200, 302, 400 sparse sampling sensors.



(a) Training error. **(b)** Test error.

Figure 5.4: Training using PREC dataset with 50, 100, 206, 300 sparse sampling sensors.

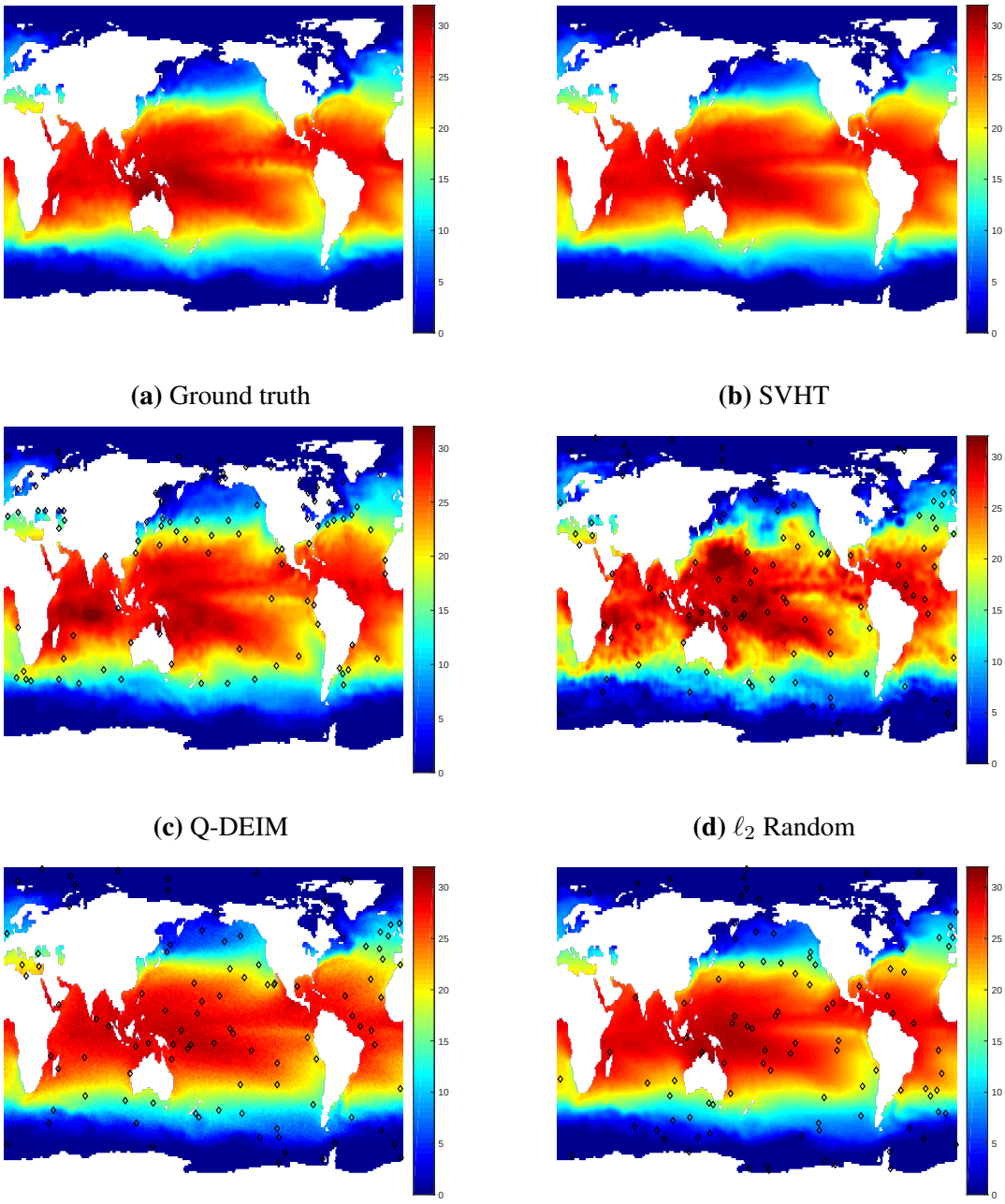
5.4.3 Performance Comparison

SST Dataset

Figure 5.5 presents the reconstruction results for the SST dataset with 100 sensor nodes, using the proposed and the benchmark algorithms. The black diamonds in this figure indicate the sensor deployment locations. The first snapshot of the SVHT and SAFER produces the visibly best reconstruction when compared with the ground truth. Details of the ground truth are fully reconstructed, and critical features of global SST, such as ocean current and El Niño, are captured. As shown in Figure 5.5 (f), SAFER tends to deploy sensor nodes near the boundaries of different temperature levels.

As sea surface temperature is profoundly affected by variations in solar radiation and the natural properties of water, global SST is usually similar at the same latitude or near the same ocean current. Hence, the discovered temperature boundary can be utilized to provide more information as the environment changes. In this manner, the SST distribution patterns and the sampling locations identified in the training data can provide sufficient information for nonlinear mapping, from limited observations to the entire spatiotemporal field. Moreover, the features and sampling locations can better fit the test data that are generated in the future. Consequently, more information can be collected, and the reconstruction model can be generalized to estimate the entire spatiotemporal field, given limited observations.

Q-DEIM produces slightly worse reconstruction results than SAFER. The reconstructed SST increases significantly over the Indian Ocean. Nevertheless, the SST reconstruction for other oceans is successful. Note that the sensor deployment locations selected by Q-DEIM are close to the borders of continents, which may occur due to higher dynamics in coastal areas. However, although the information collected in coastal regions has a higher variance than in blue water, it is difficult to generalize local information to a global situation. Instead, more observations in the oceanic area will help to produce a better reconstruction model. In comparison, the proposed method, SAFER, learns more information by deploying sensor



(a) Ground truth

(b) SVHT

(c) Q-DEIM

(d) ℓ_2 Random

(e) CS

(f) SAFER

Figure 5.5: Ground truth of the testing spatiotemporal field and reconstruction from the proposed and the benchmark algorithms in SST dataset. (Black diamonds: sampling locations.)

nodes in a more distributed manner over the spatiotemporal field.

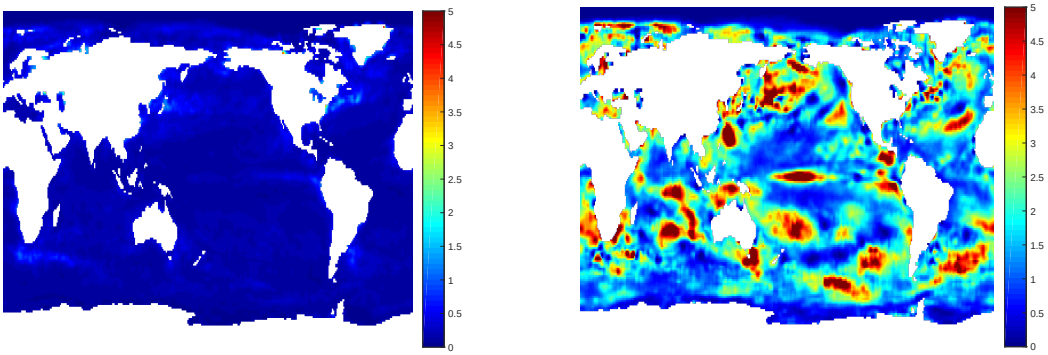
CS generates less accurate reconstruction results than Q-DEIM. The temperature distributions along the latitude are captured; however, the details of the ground truth are lost, which blurs the reconstructed field image. Information about inland bodies of water, such as the Mediterranean Sea, the Black Sea, and the Great Lakes, is also lost. Moreover, as CS-based approaches do not provide a scheme to optimize the sensor deployment locations, the sparse sensing matrix is randomly chosen. Therefore, the reconstruction process may have higher variance if the selected sampling locations are ineffective.

ℓ_2 Random produces the worst reconstruction results. The reconstructed SST is significantly higher than the ground truth in several oceans. As in CS, the sampling locations are randomly selected; therefore, less valid observations are obtained. As a result, ℓ_2 Random cannot provide satisfactory results in reconstruction and estimation.

Figure 5.6 presents the prediction variance for all test sets, which examine the reconstruction variance for long-term observation and evaluate the ability to predict the future spatiotemporal field. The presented variance is calculated pixel-wise and evaluates if the reconstruction MSE changes significantly for the prediction of all future time series generations. Hence, a higher variance shown in this figure indicates a lower confidence for future prediction.

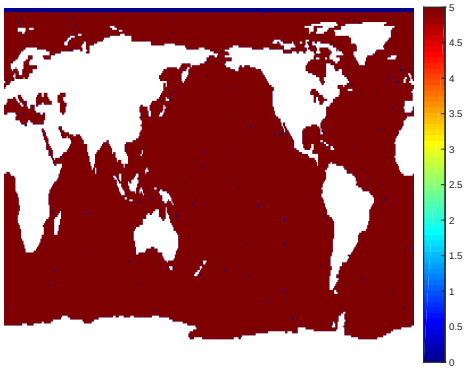
As shown in Figure 5.6 (a), SVHT achieves the lowest variance across the entire field. SAFER has the second-lowest estimation variance for the test data. Higher uncertainty in prediction occurs in some offshore areas. For example, the variance in the northern Pacific Ocean is high, which may be affected by the Aleutian Islands for the SST disturbance introduced by the archipelago. CS and Q-DEIM have higher variance in predicting future spatiotemporal fields than SAFER. Their reconstruction and prediction models are not generalized and are not robust for estimating environmental changes.

Moreover, the selected sensor deployment locations cannot provide sufficient information for the reconstruction model. As a result, when the environment changes, the sensor

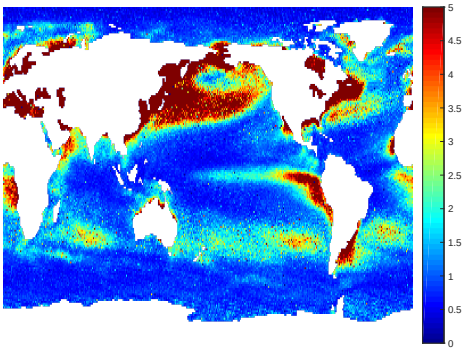


(a) SVHT

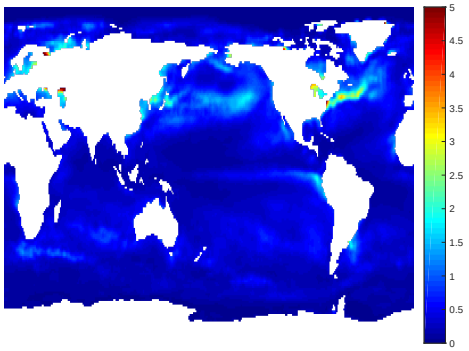
(b) Q-DEIM



(c) ℓ_2 Random



(d) CS



(e) SAFER

Figure 5.6: Reconstruction variance for all test snapshots in the SST dataset.

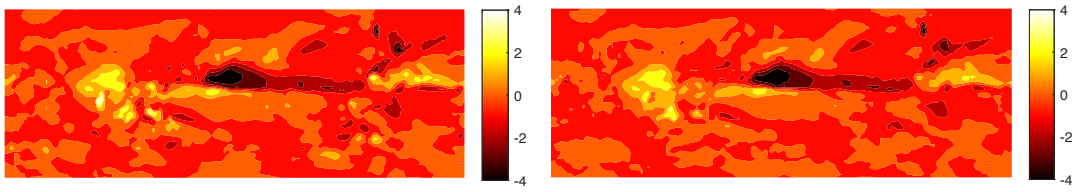
deployment strategies generated by these two approaches cannot perform well. ℓ_2 Random has the highest variance due to randomly deployed sensor nodes. Hence, CS, Q-DEIM and ℓ_2 Random are not suitable for sensor deployment or spatiotemporal field reconstruction.

PREC Dataset

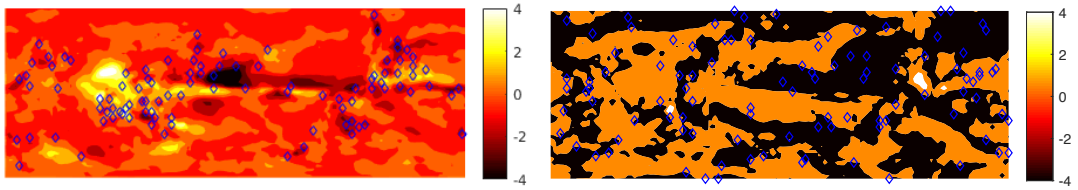
Figure 5.7 presents the reconstruction results for the PREC dataset. Generally, as global precipitation is more complicated than sea surface temperature, the reconstruction results are less accurate for the PREC dataset than for the SST dataset. Similar to the results for the SST dataset, SVHT and SAFER produce the best reconstruction results for the PREC dataset, while Q-DEIM produces the second-best reconstruction result. SAFER generates a more generalized model than Q-DEIM. Although some features are lost, the spatiotemporal field is accurately reconstructed. In contrast, Q-DEIM tends to over-fit the training data. Hence, slight increases/decreases in precipitation may have a significant impact on the reconstruction, causing a higher reconstruction MSE.

As shown in Figure 5.7 (d)-(e), CS and ℓ_2 Random cannot produce effective reconstruction for the PREC dataset. Significant disturbance and variance are introduced, and the original spatiotemporal field can neither be reconstructed nor predicted.

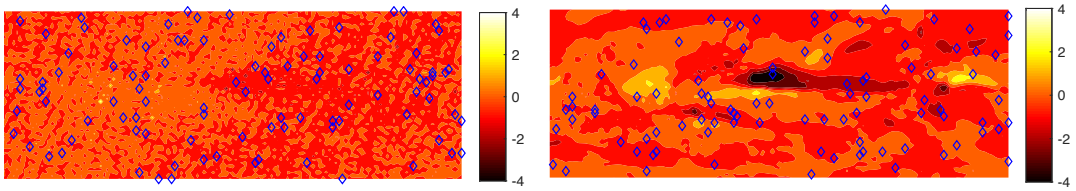
The variances in predicting future spatiotemporal fields for the PREC dataset are presented in Figure 5.8. The results for ℓ_2 Random are excluded from this figure due to the high variance of this approach in all the sampling locations. SAFER achieves the lowest overall variance throughout the spatiotemporal field, which means that this approach has high robustness for predicting a complex spatiotemporal field. SVHT and Q-DEIM have elevated variance for all test sets, and the variance in test snapshots changes dramatically. Moreover, future spatiotemporal field reconstruction and prediction will be significantly less accurate. Hence, although the reconstruction snapshots shown in Figure 5.7 (b) - (c) are satisfactory, SVHT and Q-DEIM are not suitable for long-term spatiotemporal field reconstruction and prediction. Similarly, CS performs poorly in future spatiotemporal field reconstruction and



(a) Ground truth **(b)** SVHT

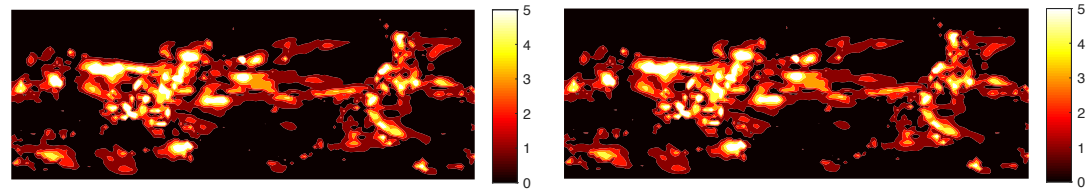


(c) Q-DEIM **(d)** ℓ_2 Random

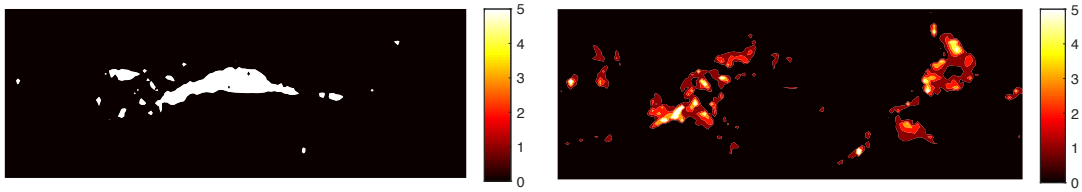


(e) CS **(f)** SAFER

Figure 5.7: Ground truth of the testing spatiotemporal field and reconstruction from the proposed and benchmark algorithms in the PREC dataset. (Blue diamonds: sampling locations.)



(a) SVHT **(b)** Q-DEIM



(c) CS **(d)** SAFER

Figure 5.8: Reconstruction variance for all test snapshots in the PREC dataset.

prediction.

Table 5.1 summarizes the performance of the SST and the PREC datasets based on the evaluation metrics described in Section 5.4.1. In general, SAFER achieves the best reconstruction accuracy for long-term spatiotemporal field reconstruction and prediction in both

datasets, given limited observations. SAFER provides a performance superior to that of the benchmark algorithms, concerning both reconstruction accuracy and model robustness. The minimum reconstruction errors of SAFER in the SST and the PREC datasets are 0.393 and 0.252, respectively. Also, these values that are close to the optimal reconstruction bound. The reconstruction accuracy is at least 61.49% higher than in the benchmark algorithms, and the model robustness in predicting long-term spatiotemporal fields is increased by at least 33.85%. The lowest variance in predicting long-term precipitation data is obtained in SAFER. Hence, in reconstructing and predicting complex long-term spatiotemporal data, SAFER can produce a more generalized and robust model.

Q-DEIM and CS generate the second-best results in reconstructing and predicting the spatiotemporal data, given few observations. Q-DEIM can achieve better performance if the number of sensor nodes is small since its mechanism attempts to extract the essential information from the SVD basis. However, as the number of sensor nodes in the field increases and more information is observed, CS gradually achieves better results.

ℓ_2 Random produces the worst results with both datasets, as it generates the highest reconstruction error and variance.

5.5 Summary

In this chapter, a model-free sensor deployment strategy for spatiotemporal field reconstruction and prediction was developed. A novel DL framework was established to optimize sensor deployment locations for spatiotemporal field reconstruction and prediction. It was proven in this chapter that the proposed model is Lipschitz continuous and that the sparse sampling strategy is sufficient for reconstruction. Simulation results were conducted using two NOAA environmental datasets. The results showed that the proposed method could outperform the benchmark algorithms in both reconstruction accuracy and model robustness.

[1]SVHT provides the optimal reconstruction MSE bound and is regarded as the target algorithm for reconstructing an encoded signal. Note that SVHT cannot reconstruct the signal given few observations.

Table 5.1: Comparison of performance using the two datasets.

Dataset	Metric	SVHT[1]	Q-DEIM	ℓ_2 Random	CS	SAFER	Improv.
SST	MSE@100	0.212	1.783	25.520	2.083	0.396	77.79%
	MSE@200	0.161	2.329	27.477	1.564	0.408	73.91%
	MSE@302	0.133	2.534	26.748	1.400	0.393	71.93%
	MSE@400	0.114	2.53	27.904	1.253	0.407	67.52%
	VAR@100	0.002	0.527	163.12	1.174	0.130	75.33%
	VAR@200	0.001	0.654	195.78	0.426	0.129	70.41%
	VAR@302	8.8e-04	1.049	155.66	0.250	0.106	57.60%
	VAR@400	6.6e-04	1.076	124.73	0.192	0.127	33.85%
PREC	MSE@50	0.174	0.857	19.213	1.216	0.330	61.49%
	MSE@100	0.119	0.981	23.400	1.013	0.281	71.36%
	MSE@206	0.076	1.147	30.260	1.019	0.258	74.68%
	MSE@300	0.059	1.424	33.822	0.900	0.252	72.00%
	VAR@50	0.002	0.193	185.36	0.550	4.6e-04	99.76%
	VAR@100	0.001	0.406	137.07	0.338	6.96e-05	99.98%
	VAR@206	4.3e-04	0.44	106.64	0.342	9.49e-05	99.97%
	VAR@300	2.7e-04	0.528	81.895	0.227	6.47e-05	99.97%

Chapter 6

Hardware Implementation and Experimentation

6.1 Introduction

This chapter presents the hardware design and implementation of the environmental monitoring system, as well as the in situ experimentation carried out at the Yosef Wosk Reflecting Pool of the University of British Columbia in Vancouver, Canada. A prototype unmanned surface vehicle (USV) is developed to take water data samples in the aquatic field, which can model the environment for validating the proposed algorithms.

Researchers have developed robotic sensors and sensor networks to monitor aquatic environments. However, the existing work has some drawbacks such as not using self-propelled monitoring platforms [87, 142], or has difficulties in long-term deployment in extensive areas [143, 144]. In this chapter, a fully automated and cost-effective USV is developed for environmental monitoring.

As shown in Figure 6.1, the USV consists of three groups of components: water quality sensors, onboard controllers, and the mobile base of the USV. Figure 6.1 (a) shows the top view of the USV; for signal transmission, the onboard controller and a GPS receiver are placed on top of the USV. Figures 6.1 (b)-(c) show the arrangement of the sensors and

(a) Top view. **(b)** Back view.

(c) Front-right view. **(d)** Side view.

Figure 6.1: The prototype of the developed USV.

propellers; to take water data samples, the water quality sensors are placed around the USV without interfering with one another. Also, to power the sensors, a solar panel is used, which is placed on top of the USV.

The USVs can explore the field of interest and sample data in a distributed manner. The collected data is then used to build the environmental model for information-based sensor deployment.

6.2 Hardware Design

6.2.1 Water Quality Sensors

The water quality sensors used in the hardware implementation are Libelium Plug & Sense devices. Five types of water quality sensors are adopted in the developed USV: pH

Table 6.1: Specifications for the water quality sensors.

Sensor	Sensing range	Accuracy
pH value sensor	0 - 14 pH	± 0.5 mV
DO sensor	0 - 20 mg/L	$\pm 2\%$
Temperature sensor	0 - 100°C	± 0.60
Conductivity sensor	N/A	N/A
ORP sensor	0 - $\pm$1999 mV	N/A

value, dissolved oxygen (DO), temperature, conductivity, and oxidation-reduction potential (ORP). These sensors can adequately carry out water quality monitoring. Sensor data is read via serial ports, using Arduino-based software [1]. The specifications for the selected sensors are given in Table 6.1 [2]

6.2.2 Onboard Controllers

TThe onboard controller and the design of the autonomous navigation system of the USV are all based on Pixhawk, an autopilot controller for small unmanned vehicles [3]. The USV can be automatically navigated to selected locations, based on the GPS coordinates and gyroscope data. In the beginning of the navigation process, a map of the field of interest is loaded, and obstacles are detected. The GPS receiver provides the current location of the USV, and the pose of the USV is obtained from the gyroscope. The USV can then adjust itself during the navigation process according to the relative angle between the USV and the target location. The feedback control loop of the USV navigation system is shown in Figure 6.2. As shown there, the USV can continuously adjust its movements according to the localization information and through feedback of its relative position from the target point.

[1] https://www.arduino.cc/
[2] http://www.libelium.com/downloads/documentation/smart_water_sensor_board.pdf
[3] http://pixhawk.org

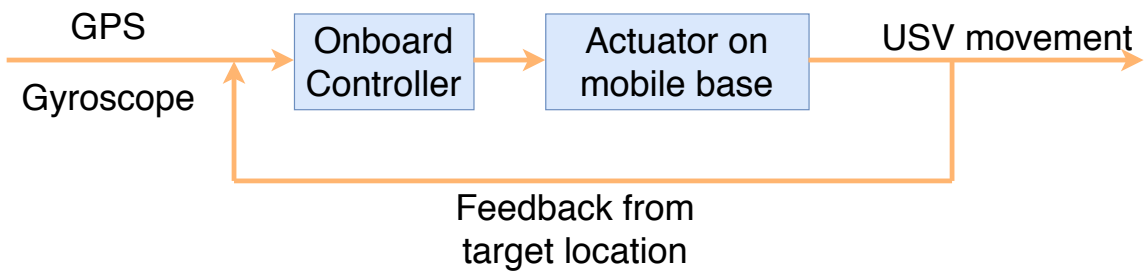


Figure 6.2: Feedback control of the USV.

6.2.3 Mobile Base of USV

The mobile base is developed from the Blue Robotics T100 Thruster [4]. To extend the USV life, fewer propellers are used, and the water sealing and buoyancy of the USV are redesigned. A buoy is added to provide sufficient buoyancy for onboard equipment, such as water quality sensors and the onboard controller. To prevent water damage to the electronic equipment, the onboard equipment is placed in a sealed enclosure, while the antennas are kept on top of the USV to transmit data.

As the in situ experiments are carried out in an aquatic environment with a weak current, the USV is equipped with only two propellers. More propellers can be added to the USV to provide sufficient thrust, depending on the environment of a field test. The specifications for the propellers are given in Table 6.2 [5]

Table 6.2: Specifications for the propeller.

Item	Specification
Maximum Forward Thrust	2.36 kg_f
Maximum Reverse Thrust	1.85 kg_f
Minimum Thrust	0.01 kg_f
Rotational Speed	300-4200 rev/min

[4] https://www.bluerobotics.com/
[5] https://docs.bluerobotics.com/thrusters/

6.3 In Situ Tests and Experimental Results

6.3.1 Experiment Setup for In Situ Tests

The developed USV is used for in situ experiments carried out at the Yosef Wosk Reflecting Pool. The framework for the conducted tests is shown in Figure 6.3. In the beginning of the experimental process, the USV explores the field of interest for a generalized understanding of the environment. Then, based on the sampled data, the environment is spatiotemporally modeled according to the metrics introduced in Chapter 2. Next, information-based sampling strategies are executed to find near-optimal sampling locations, such as field estimation, reconstruction, and prediction. Finally, the sensor nodes (USVs) are deployed in the computed locations for long-term monitoring. The above process will be periodically executed to update the model and obtain better performance of the algorithm.

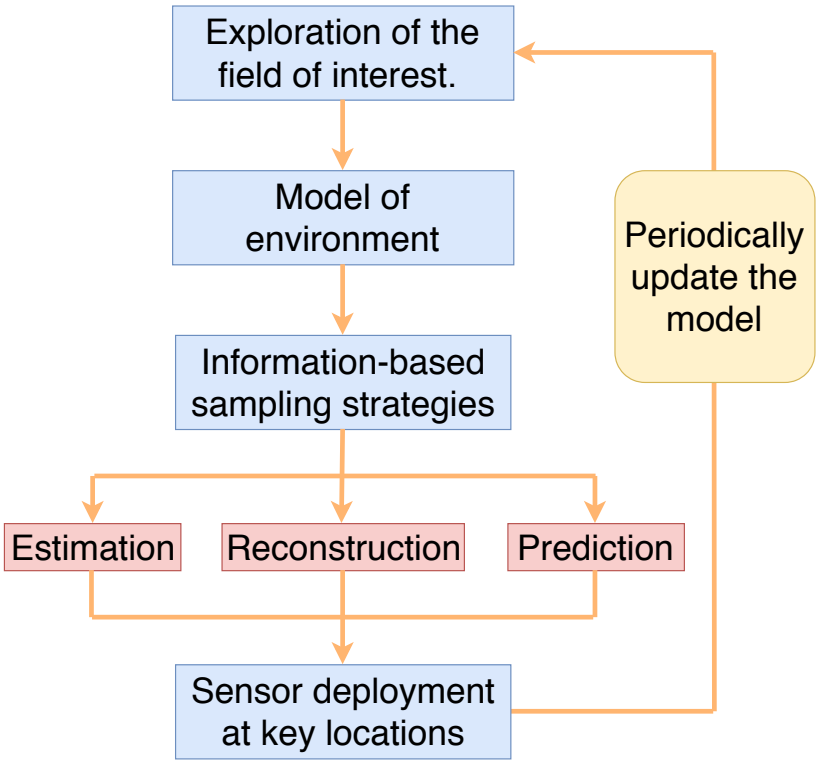
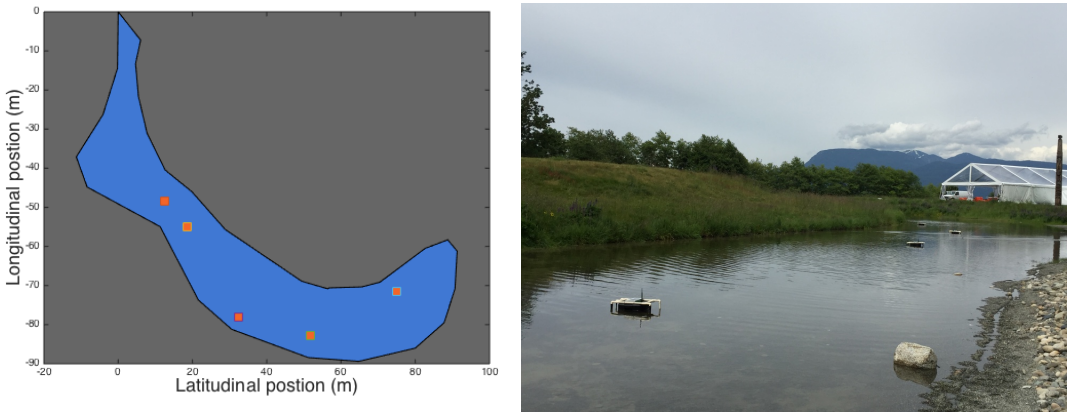


Figure 6.3: The framework for in situ experiments.

The in situ test collects five key water quality attributes (temperature, pH value, con-

ductivity, ORP, and DO) from five locations in the pool. Sampling locations are shown in Figure 6.4 (a), where an orange square denotes a sensor node location, blue area indicates the aquatic environment, and the gray area is the region (land or obstacle) that the mobile sensor nodes cannot reach. The area shown in Figure 6.4 (a) is 120 meters wide and 90 meters long. Boundaries are drawn according to the rough GPS coordinates and turned into the scale of one meter by selecting the top-left point as the origin. Figure 6.4 (b) presents the real-world experiment scenario. The proposed information-based sampling method assumes that each sensor node collects only one water quality attribute, and the spatiotemporal model is established based on the temperature sensor readings.



(a) Sensor deployment locations. **(b)** Real-world deployment scenario.

Figure 6.4: Sensor deployment for the in situ test.

6.3.2 Experimental Results for Spatiotemporal Sensor Deployment using Minimal Sensor Nodes

In this section, methods introduced in Chapter 2 are carried out based on the in situ test results. Figure 6.5 shows the results generated from four RRT-based algorithms and the environmental model generated from the real-world sampled data. Figure 6.5 (a) and Figure 6.5 (b) are generated from two benchmark algorithms, and Figure 6.5 (c) and Figure 6.5 (d) show the results generated by the proposed algorithm RRLR and RRLR Connectivity, respectively. The yellow square denotes the suitable sensor node deployment location; red

diamond blocks represent the redundant deployment locations, and the blue lines indicate graph G with several spanning trees.

As shown in Figure 6.5 (a) and Figure 6.5 (b), both RRC and RRCstar require 12 sensor node deployment locations to achieve the minimum prior estimation error. However, some sensor nodes are too close, which can result in redundancy. They can be eliminated by using the RRLR algorithm, as shown in Figure 6.5 (c) and Figure 6.5 (d). Only eight sensor nodes are needed to be deployed in the geographic EOI when using RRLR to achieve minimum prior estimation error. Moreover, RRLR Connectivity not only reduces the redundant sampling locations but also maintains the network connectivity. If we remove the results of all redundant sensor nodes from RRLR as shown in Figure 6.5 (c), the wireless sensor network will be split into two parts, and the connectivity will be broken. RRLR Connectivity only removes two redundant sampling locations to maintain both network connectivity and the minimal deployment location.

Figures 6.6 (a) - (c) present the results generated by RS, GD4, and GD8, respectively. RS only searches a small part of the environment as it uses a randomized approach, which may not generate a near-optimal result. In contrast, GD4 and GD8 search the field sufficiently and achieves a lower estimation error as indicated in Figure 6.8. However, these benchmark methods require many deployment locations and may not be able to explore a sufficient geographic EOI. Hence, the randomized approach cannot collect sufficient information from the field, and the sampling data redundancy is not investigated. Hence, they cannot guarantee a near-optimal spatiotemporal sensor deployment while using minimal sensor nodes.

Figure 6.7 presents the field estimation uncertainty results when using the deployment strategies generated by the RRT-based algorithms. The field estimation uncertainty is calculated based on what is presented in Chapter 2. Estimation uncertainties of random search methods are ignored due to their poor performance in generating deployment strategies. As Figure 6.7 shows, all four methods will generate reasonable estimation of the EOI since

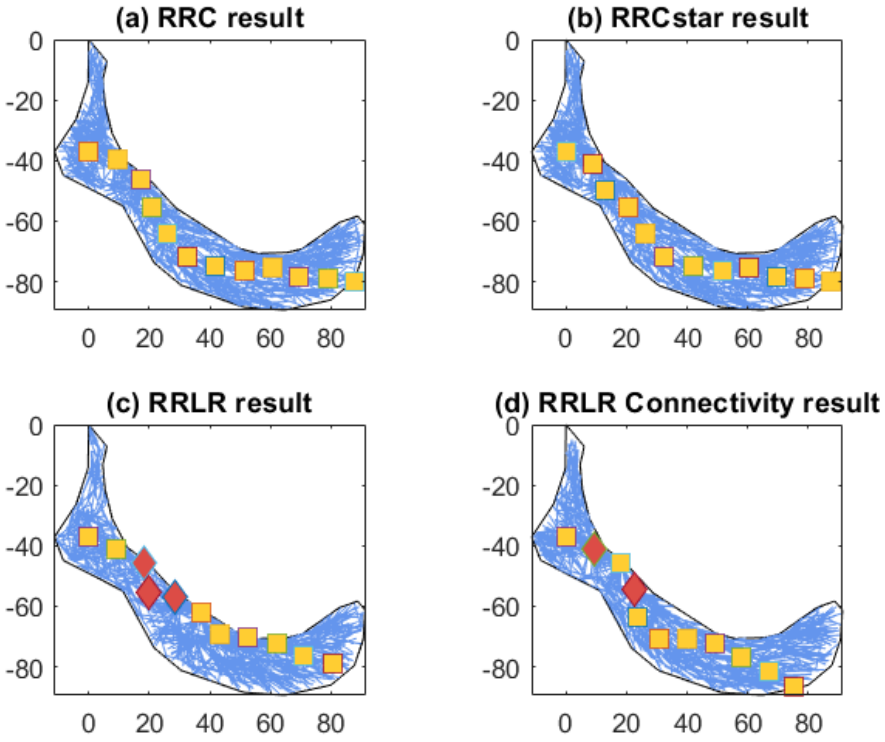


Figure 6.5: Deployment results generated by RRT-based algorithms.

most regions of the estimated field have low estimation uncertainty.

Moreover, as Table 6.3 shows, the overall estimation uncertainty is very low over the EOI for all algorithms. Thus, both the proposed algorithm and the benchmark RRT-based algorithm will generate good deployment locations with low estimation uncertainty. Besides, the proposed algorithms use 33% and 25% fewer sensor nodes to achieve similar performance.

Table 6.3: Statistical measurement of the field estimation uncertainty.

Method	Max	Min	Variance	# sensors
RRC	$1.31 * 10^{-118}$	0	$2.55 * 10^{-120}$	12
RRCstar	$1.02 * 10^{-118}$	0	$1.99 * 10^{-120}$	12
RRLR	$8.39 * 10^{-118}$	0	$1.63 * 10^{-119}$	8
RRLR Connectivity	$2.09 * 10^{-118}$	0	$4.08 * 10^{-120}$	9

Figure 6.8 presents the minimal prior estimation error found by the algorithms at each

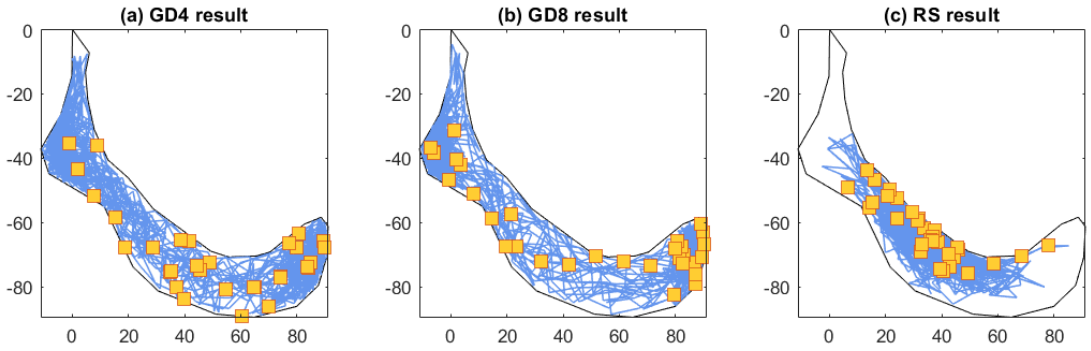


Figure 6.6: Deployment results generated by random search-based algorithms.

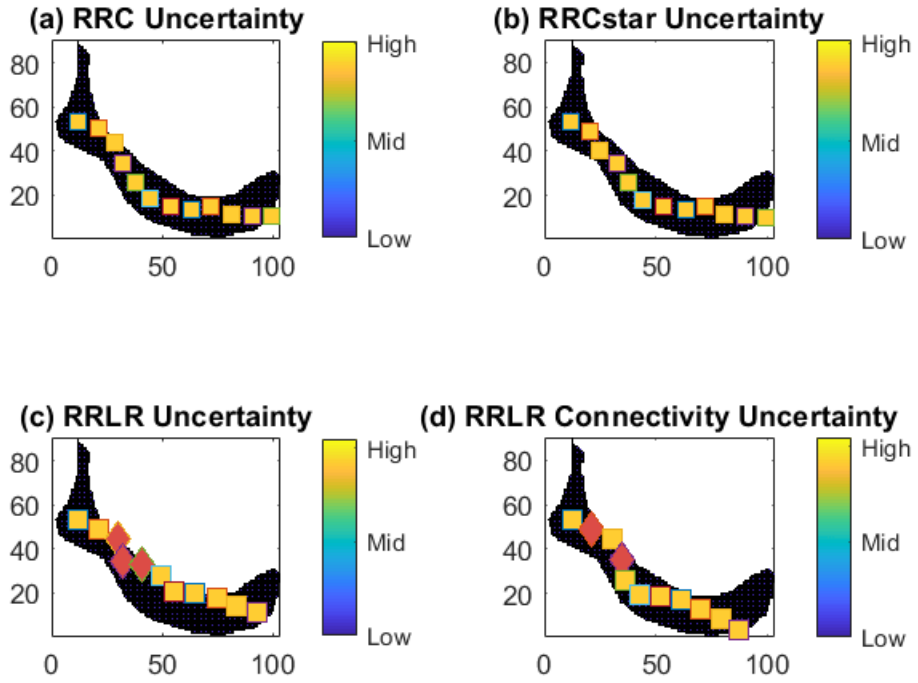


Figure 6.7: Field estimation uncertainty results of RRT-based algorithms.

iteration. All results are monotonically decreasing as the minimum estimation error is recorded in each iteration. The prior estimation error generated by the randomized methods stays at a high level, while the one provided by the RRT-based algorithms decrease significantly. RS algorithm produces the worst result since it is only able to search in a limited area. The greedy search-based approach improves the performance slightly by exploring

more parts of EOI. However, all these three algorithms can easily entrap at local minima without achieving a near-optimal solution.

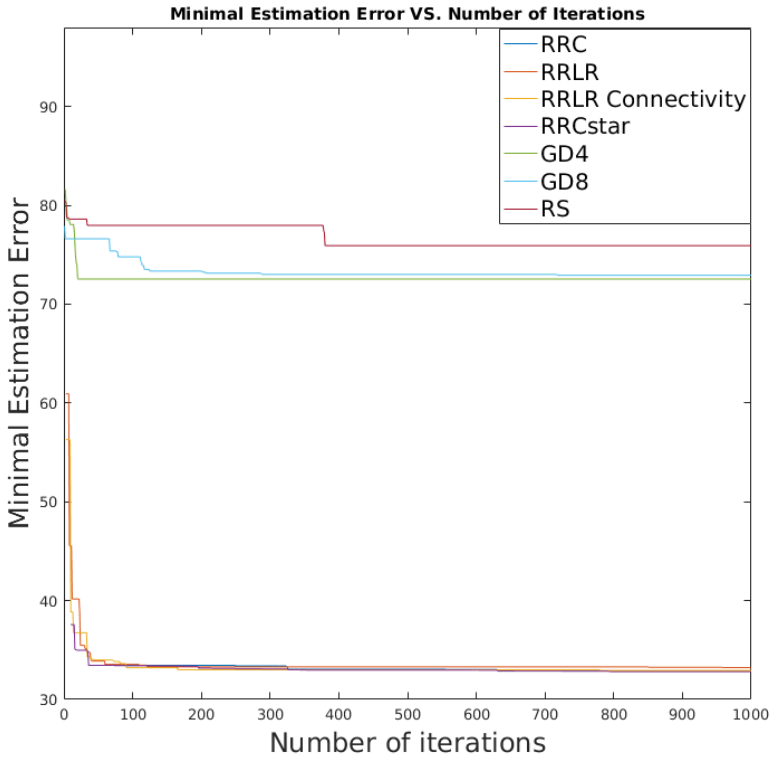


Figure 6.8: Change of minimum estimation error.

According to Proposition 2.2, all four RRT-based algorithms provides near-optimal results after 1000 iterations as the estimation error converges. Hence, the estimation error of the RRT-based algorithms converges with the number of iteration. In this manner, the in situ test confirms that the proposed method achieves the lowest estimation error while deploying minimal sensor nodes.

6.4 Summary

In this chapter, the developed prototype of the autonomous surface vehicle was described and used for the in situ experiments. An environmental model was built based on the conducted in situ tests and used for evaluating the effectiveness of the proposed information-based sampling strategies. The results confirmed that the proposed methods

outperform the benchmark algorithms in both simulation tests and real-world tests.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this dissertation, several main challenges of informative sampling for spatiotemporal field estimation, reconstruction, and prediction were investigated. Sampled data from the field was analyzed so as to preserve the most informative sampling locations and minimize infrastructure costs. In this manner, the selected sampling locations were able to provide sufficient information for various monitoring tasks such as field estimation, reconstruction, and prediction. Learning methods were also used to improve the model efficiency and performance. In the present work, the actual sensor navigation cost is not considered, which simplifies the computations in the optimization algorithms.

The first goal of this dissertation was to find the most informative sampling locations in a field in order to minimize the infrastructure costs while maintaining a good estimation accuracy of the field and preserving the network connectivity. To this end, a near-optimal sensor node deployment strategy was developed. The proposed method investigated the linear dependency of the sampled data, given the environment model. Then, spatiotemporally correlated sampling locations could be eliminated while giving due consideration to network connectivity. Moreover, it was proven in the dissertation that the proposed method had submodularity and generated a near-optimal sampling plan. The results demonstrated

that the proposed method required fewer sensor nodes than the benchmark algorithms to achieve a sufficiently low estimation error.

Second, the computational complexity in finding the near-optimal sampling locations was reduced. A DRL-based field exploration method was proposed to accelerate the process of finding the most informative sampling locations. Feedback from each sampling location in the field was studied to avoid unnecessary sampling locations, and a model-based information gain was calculated to evaluate the effectiveness of each sampling location during the exploration. As a result, the proposed method, DRLT, was able to learn from the sampled locations and guide the robotic sensor to avoid unnecessary sampling locations. The sampling locations found by the DRL model were proven to be near-optimal. Simulations were conducted using an NCOM dataset. The proposed algorithm showed significant improvement in both field estimation performance and planning efficiency.

Then, the spatiotemporal field was reconstructed based on the observations from the selected near-optimal sampling locations. A model-based sensor deployment strategy was developed to reconstruct the spatiotemporal field with limited observations. Statistical methods were utilized to find linear mappings from the observations to the entire spatiotemporal field. The sampling locations generated by the proposed method were proven to be near-optimal for spatiotemporal reconstruction. Moreover, the simulation results demonstrated that the proposed information-based sampling methods provided the best reconstruction performance when compared with the benchmark algorithms.

However, model-based reconstruction methods find only the linear mappings from the sampling data to the overall spatiotemporal field. Some features of the original signal were lost when the spatiotemporal field became more complicated. Therefore, a model-free DL-based framework was developed to capture the nonlinear relationships between the observations in the critical sampling locations and the spatiotemporal field. The sensor deployment locations were then optimized by a simulated annealing-based approach during the training. Hence, the new observations in the selected sampling locations could be used to reconstruct

and predict the entire spatiotemporal field, given the trained DL framework.

It was proven as well that the proposed DL framework had Lipschitz continuity, and that the generated sampling locations were sufficient for spatiotemporal reconstruction and prediction.

Several USVs were developed for environmental monitoring, and in situ tests were conducted. The collected data samples were used to build an environmental model for an information-based sampling algorithm. The experimental results showed that the proposed algorithm had superior performance in real-world scenarios when compared to the benchmark algorithms.

7.2 Future Work

In possible future developments, multichannel sensor data analysis may be carried out to perform multi-object information sampling. In the present investigation, the proposed methods and the state-of-the-art approaches focused only on detecting the most informative sampling locations for a single channel of data sampling. In future work, sensor fusion and multichannel signal processing may be utilized to analyze various types of sensor data and to decide the optimal sampling locations for spatiotemporal field monitoring.

In this dissertation, DL methods were utilized to investigate the optimal nonlinear spatial mappings from the observations to the entire spatiotemporal field. However, the analysis of the spatial and temporal information was separate. Time series data analysis methods may be incorporated to more accurately predict the spatiotemporal field. Techniques such as recurrent neural networks (RNNs) and long short-term memory (LSTM) may also be used to learn temporal changes while learning the spatial mappings between the observations and the overall spatiotemporal field [145–147]. In this manner, nonlinear spatial mappings can be captured based on the temporal predictions. Consequently, the reconstruction precision of the field will improve.

The USV developed in this dissertation has limitations in carrying out experiments in

quite complex environments. Currently, the environment of interest is assumed to be known, and the obstacles are assumed to be static. Hence, fully autonomous exploration in an unknown environment with dynamic obstacles would not be feasible with the current USVs. Simultaneous localization and mapping (SLAM) techniques may be utilized to build a map of an unknown environment while simultaneously enabling the USV to localize [148–150]. RGB cameras and laser range finders may be employed to meet the requirements of SLAM, and to simultaneously enable the USV to avoid dynamic obstacles. Moreover, open-source robotic middleware, such as robot operating systems (ROS) may be used for easier development and control of a USV [151].

Bibliography

[1] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 1, p. 4, 2008. → pages 1, 6, 44

[2] L. A. Villas, A. Boukerche, H. A. De Oliveira, R. B. De Araujo, and A. A. Loureiro, "A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks," *Ad Hoc Networks*, vol. 12, pp. 69–85, 2014. → pages 1, 6

[3] X. Lan and M. Schwager, "Rapidly exploring random cycles: Persistent estimation of spatiotemporal fields with multiple sensing robots," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1230–1244, 2016. → pages 1, 3, 8, 9, 20, 21, 46, 47, 50, 64, 65, 81, 85

[4] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Systems*, vol. 38, no. 3, pp. 63–86, 2018. → pages 1, 4, 11, 73, 75, 102, 103

[5] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004. → pages 2, 5, 44

[6] S. Yoon and C. Shahabi, "The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 3, no. 1, p. 3, 2007. → pages 2, 5

[7] F. Li, J. Luo, W. Wang, and Y. He, "Autonomous deployment for load balancing k-surface coverage in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 279–293, 2015. → page 2

[8] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Utilisation of multi-objective immune deployment algorithm for coverage area maximisation with limit mobility in wireless sensors networks," *IET Wireless Sensor Systems*, vol. 5, no. 5, pp. 250–261, 2015. → pages 2, 7

[9] J. Chen, T. Li, T. Shu, and C. W. de Silva, "Rapidly-exploring tree with linear reduction: A near-optimal approach for spatiotemporal sensor deployment in aquatic fields using minimal sensor nodes," *IEEE Sensors Journal*, 2018. → pages 3, 9, 85

[10] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008. → pages 3, 8, 25, 44, 45, 52, 57, 58

[11] M. A. Alsheikh, S. Lin, H.-P. Tan, and D. Niyato, "Area coverage under low sensor density," in *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, pp. 173–175, IEEE, 2014. → pages 3, 6

[12] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998. → page 3

[13] M. S. Sivamurugan and B. Ravindran, "Rrtpi: Policy iteration on continuous domains using rapidly-exploring random trees," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 4362–4367, IEEE, 2014. → pages 4, 9, 45, 64, 81

[14] L. A. Villas, A. Boukerche, D. L. Guidoni, H. A. De Oliveira, R. B. De Araujo, and A. A. Loureiro, "An energy-aware spatio-temporal correlation mechanism to perform efficient data collection in wireless sensor networks," *Computer Communications*, vol. 36, no. 9, pp. 1054–1066, 2013. → page 6

[15] N. Ganganath, C.-T. Cheng, and K. T. Chi, "Distributed antiflocking algorithms for dynamic coverage of mobile sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1795–1805, 2016. → page 6

[16] N. Heo and P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1597–1602, IEEE, 2003. → page 6

[17] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, 2005. → page 6

[18] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm," *Computers & Electrical Engineering*, vol. 43, pp. 76–89, 2015. → page 7

[19] M. Abo-Zahhad, N. Sabor, S. Sasaki, and S. M. Ahmed, "A centralized immune-voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks," *Information Fusion*, vol. 30, pp. 36–51, 2016. → page 7

[20] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006. → pages 7, 9

[21] L. Zhao, W. Zhang, J. Hu, A. Abate, and C. J. Tomlin, "On the optimal solutions of the infinite-horizon linear sensor scheduling problem.," *IEEE Trans. Automat. Contr.*, vol. 59, no. 10, pp. 2825–2830, 2014. → page 7

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning with deterministic μ-calculus specifications," in *American Control Conference (ACC), 2012*, pp. 735–742, IEEE, 2012. → page 7

[23] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, pp. 265–272, ACM, 2005. → pages 8, 44

[24] N. Cressie, *Statistics for spatial data*. John Wiley & Sons, 2015. → pages 8, 44

[25] W. Du, Z. Xing, M. Li, B. He, L. H. C. Chua, and H. Miao, "Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs," in *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, pp. 167–178, IEEE, 2014. → pages 8, 44

[26] W. Du, Z. Xing, M. Li, B. He, L. H. C. Chua, and H. Miao, "Sensor placement and measurement of wind for water quality studies in urban reservoirs," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 3, p. 41, 2015. → page 8

[27] X. Lan and M. Schwager, "Planning periodic persistent monitoring trajectories for sensing robots in gaussian random fields," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 2415–2420, IEEE, 2013. → page 8

[28] X. Lan and M. Schwager, "A variational approach to trajectory planning for persistent monitoring of spatiotemporal fields," in *American Control Conference (ACC), 2014*, pp. 5627–5632, IEEE, 2014. → page 47

[29] X. Lan and S. Di Cairano, "Continuous curvature path planning for semi-autonomous vehicle maneuvers using rrt," in *Control Conference (ECC), 2015 European*, pp. 2360–2365, IEEE, 2015. → page 8

[30] L. V. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Mobile robotic wireless sensor networks for efficient spatial prediction," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 1176–1181, IEEE, 2014. → page 8

[31] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011. → page 9

[32] R. Cui, Y. Li, and W. Yan, "Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016. → page 9

[33] C. Diuk, A. Cohen, and M. L. Littman, "An object-oriented representation for efficient reinforcement learning," in *Proceedings of the 25th international conference on Machine learning*, pp. 240–247, ACM, 2008. → page 9

[34] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[35] C.-T. Law and J. I. Gold, "Reinforcement learning can account for associative and perceptual learning on a visual-decision task," *Nature neuroscience*, vol. 12, no. 5, pp. 655–663, 2009.

[36] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8, IEEE, 2010.

[37] M. Riedmiller, "Neural fitted q iteration-first experiences with a data efficient neural reinforcement learning method," in *ECML*, vol. 3720, pp. 317–328, Springer, 2005.

[38] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, 2015. → page 9

[39] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015. → page 9

[40] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017. → page 9

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013. → page 9

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. → pages 9, 45, 61

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012. → page 9

[44] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, vol. 2, p. 3, 2010. → page 9

[45] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 533–540, ACM, 2014. → page 9

[46] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents.," *J. Artif. Intell. Res.(JAIR)*, vol. 47, pp. 253–279, 2013. → page 9

[47] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1527–1533, IEEE, 2017. → page 10

[48] K. Shiarlis, J. Messias, and S. Whiteson, "Rapidly exploring learning trees," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1541–1548, IEEE, 2017. → page 10

[49] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. → page 10

[50] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007. → page 10

[51] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010. → page 10

[52] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2392–2399, IEEE, 2012.

[53] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, pp. 341–349, 2012. → page 10

[54] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, pp. 184–199, Springer, 2014. → page 10

[55] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016. → page 10

[56] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pp. 1336–1343, IEEE, 2015. → pages 10, 12, 91

[57] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001. → pages 10, 103

[58] Z. Zhang, T.-P. Jung, S. Makeig, Z. Pi, and B. D. Rao, "Spatiotemporal sparse bayesian learning with applications to compressed sensing of multichannel physiological signals," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 22, no. 6, pp. 1186–1197, 2014. → pages 10, 73, 103

[59] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Sparse sensor placement optimization for classification," *SIAM Journal on Applied Mathematics*, vol. 76, no. 5, pp. 2099–2122, 2016. → page 10

[60] X. Lu, W. Dong, P. Wang, G. Shi, and X. Xie, "Convcsnet: A convolutional compressive sensing framework based on deep learning," *arXiv preprint arXiv:1801.10342*, 2018. → page 10

[61] W. Guo, K. Manohar, S. L. Brunton, and A. G. Banerjee, "Sparse-tda: Sparse realization of topological data analysis for multi-way classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1403–1408, 2018. → page 11

[62] C. Lu and B. Jayaraman, "Interplay of sensor quantity, placement and system dimensionality on energy sparse reconstruction of fluid flows," *arXiv preprint arXiv:1806.08428*, 2018. → page 11

[63] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000. → page 11

[64] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[65] K. Sjöstrand, L. H. Clemmensen, R. Larsen, G. Einarsson, and B. K. Ersbøll, "Spasm: A matlab toolbox for sparse statistical modeling," *Journal of Statistical Software*, vol. 84, no. 10, 2018. → page 11

[66] R. Jenatton, G. Obozinski, and F. Bach, "Structured sparse principal component analysis," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 366–373, 2010. → page 11

[67] M. Hein and T. Bühler, "An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca," in *Advances in Neural Information Processing Systems*, pp. 847–855, 2010. → page 11

[68] Z. Drmac and S. Gugercin, "A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions," *SIAM Journal on Scientific Computing*, vol. 38, no. 2, pp. A631–A648, 2016. → pages 11, 103

[69] N. B. Erichson, P. Zeng, K. Manohar, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, "Sparse principal component analysis via variable projection," *arXiv preprint arXiv:1804.00341*, 2018. → page 12

[70] J. Janková and S. van de Geer, "De-biased sparse pca: Inference and testing for eigenstructure of large covariance matrices," *arXiv preprint arXiv:1801.10567*, 2018. → page 12

[71] M. Mollenhauer, I. Schuster, S. Klus, and C. Schütte, "Singular value decomposition of operators on reproducing kernel hilbert spaces," *arXiv preprint arXiv:1807.09331*, 2018. → page 12

[72] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," 2010. → page 12

[73] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014. → page 12

[74] C.-W. Loh, Z.-Q. Qian, R. Zhang, Y.-H. Liu, D.-W. Cao, W. Wang, H.-B. Yang, and M. Qi, "Deep learning the effects of photon sensors on the event reconstruction performance in an antineutrino detector," *Advances in High Energy Physics*, vol. 2018, 2018. → page 12

[75] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. → page 12

[76] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014. → page 13

[77] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," *arXiv preprint arXiv:1703.03208*, 2017. → pages 13, 98

[78] H. Gouk, E. Frank, B. Pfahringer, and M. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *arXiv preprint arXiv:1804.04368*, 2018. → pages 13, 96

[79] M. L. Stein, *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012. → pages 17, 19, 20

[80] J. Cortés, "Distributed kriged kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2816–2827, 2009. → pages 20, 21

[81] L. Ljung, *System Identification Toolbox 7: Getting Started Guide*. The MathWorks, 2008. → page 20

[82] "Estimate state-space model using subspace method - matlab n4sid." https://www.mathworks.com/help/ident/ref/n4sid.html. Accessed: 2017-08-24. → pages 20, 63

[83] "Naval oceanographic office regional navy coastal ocean model (ncom)." https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/navoceano-ncom-reg. Accessed: 2017-08-25. → pages 20, 63, 80

[84] U. D. of Commerce and N. N. C. for Environmental Information, "Coastal water temperature guide." http://www.nodc.noaa.gov/cwtg/all_tmap.html, Jun 2018.

[85] G. K. Rutledge, J. Alpert, and W. Ebisuzaki, "Nomads: A climate and weather model archive at the national oceanic and atmospheric administration," *Bulletin of the American Meteorological Society*, vol. 87, no. 3, pp. 327–341, 2006. → pages 20, 63

[86] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. → pages 21, 47

[87] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009. → pages 25, 44, 114

[88] B. Fateh and M. Govindarasu, "Energy minimization by exploiting data redundancy in real-time wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1715–1731, 2013. → page 30

[89] D. Ganesan, D. Estrin, and J. Heidemann, "Dimensions: Why do we need a new data handling architecture for sensor networks?," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 143–148, 2003. → page 30

[90] R. Jedermann, H. Paul, and W. Lang, "Compressed radio transmission of spatial field measurements by virtual sensors," in *Wireless for Space and Extreme Environments (WiSEE), 2016 IEEE International Conference on*, pp. 184–189, IEEE, 2016. → page 35

[91] H. M. La, W. Sheng, and J. Chen, "Cooperative and active sensing in mobile sensor networks for scalar field mapping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 1–12, 2015. → page 43

[92] T. Li, M. Xia, J. Chen, Y. Zhao, and C. de Silva, "Automated water quality survey and evaluation using an iot platform with mobile sensor nodes," *Sensors*, vol. 17, no. 8, p. 1735, 2017. → page 43

[93] G. Hitz, E. Galceran, M.-È. Garneau, F. Pomerleau, and R. Siegwart, "Adaptive continuous-space informative path planning for online environmental monitoring," *Journal of Field Robotics*, vol. 34, no. 8, pp. 1427–1449, 2017. → page 44

[94] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal wsn deployment models for air pollution monitoring," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2723–2735, 2017. → page 44

[95] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE transactions on parallel and distributed systems*, vol. 18, no. 7, 2007. → page 44

[96] E. Karasabun, I. Korpeoglu, and C. Aykanat, "Active node determination for correlated data gathering in wireless sensor networks," *Computer Networks*, vol. 57, no. 5, pp. 1124–1138, 2013.

[97] S. Temel, N. Unaldi, and O. Kaynak, "On deployment of wireless sensors on 3-d terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 1, pp. 111–120, 2014. → page 44

[98] L. V. Nguyen, S. Kodagoda, and R. Ranasinghe, "Spatial sensor selection via gaussian markov random fields," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1226–1239, 2016. → page 44

[99] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *Journal of Field Robotics*, 2017. → pages 44, 64, 81

[100] J. Binney, A. Krause, and G. S. Sukhatme, "Optimizing waypoints for monitoring spatiotemporal phenomena," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 873–888, 2013. → page 45

[101] C. W. De Silva, *Sensor systems: Fundamentals and applications*. Crc Press, 2016. → page 47

[102] R. S. Sutton, A. G. Barto, F. Bach, *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998. → page 48

[103] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011. → page 48

[104] T. Jaksch, R. Ortner, and P. Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1563–1600, 2010. → page 49

[105] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*, vol. 39. CRC press, 2010. → page 49

[106] M. Abo-Zahhad, N. Sabor, S. Sasaki, and S. M. Ahmed, "A centralized immune-voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks," *Information Fusion*, vol. 30, pp. 36–51, 2016. → page 50

[107] A. Krause and D. Golovin, "Submodular function maximization.," 2014. → pages 52, 78

[108] R. A. Johnson and D. Wichern, *Multivariate analysis*. Wiley Online Library, 2002. → page 57

[109] P. Ding, "On the conditional distribution of the multivariate t distribution," *The American Statistician*, vol. 70, no. 3, pp. 293–295, 2016. → page 57

[110] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning.," in *AAAI*, vol. 16, pp. 2094–2100, 2016. → page 61

[111] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org. → pages 64, 101

[112] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 912–926, 2011. → page 73

[113] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936. → page 76

[114] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 5040–5053, 2014. → pages 78, 80, 103, 104

[115] T. Li, M. Xia, J. Chen, S. Gao, and C. W. de Silva, "A hexagonal grid-based sampling planner for aquatic environmental monitoring using unmanned surface vehicles," in *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pp. 3683–3688, IEEE, 2017. → page 85

[116] L. V. Nguyen, S. Kodagoda, R. Ranasinghe, and G. Dissanayake, "Adaptive placement for mobile sensors in spatial prediction under locational errors," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 794–802, 2017. → page 85

[117] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 4292–4298, IEEE, 2017.

[118] M. Dunbabin and A. Grinham, "Quantifying spatiotemporal greenhouse gas emissions using autonomous surface vehicles," *Journal of Field Robotics*, vol. 34, no. 1, pp. 151–169, 2017.

[119] S. Garg and N. Ayanian, "Persistent monitoring of stochastic spatio-temporal phenomena with a small team of robots.," in *Robotics: Science and Systems*, 2014. → page 85

[120] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5156–5163, Nov 2013. → page 85

[121] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1375–1392, 2014. → page 85

[122] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *arXiv preprint arXiv:0901.3403*, 2009. → page 86

[123] R. Bellman, *Dynamic programming*. Courier Corporation, 2013. → page 91

[124] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983. → page 94

[125] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations*. Tata McGraw-Hill Education, 1955. → page 94

[126] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018. → pages 95, 96

[127] B. Neyshabur, "Implicit regularization in deep learning," *arXiv preprint arXiv:1709.01953*, 2017. → page 95

[128] A. Soshnikov and Y. V. Fyodorov, "On the largest singular values of random matrices with independent cauchy entries," *Journal of mathematical physics*, vol. 46, no. 3, p. 033302, 2005. → page 96

[129] P. Vivo, S. N. Majumdar, and O. Bohigas, "Large deviations of the maximum eigenvalue in wishart random matrices," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 16, p. 4317, 2007.

[130] S. Mendelson, G. Paouris, *et al.*, "On the singular values of random matrices," *Preprint*, 2014. → page 96

[131] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 4, pp. 1307–1330, 1986. → page 97

[132] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996. → page 97

[133] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005. → page 97

[134] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. → page 101

[135] NOAA, "Noaa optimum interpolation sea surface temperature v2." https://www.esrl.noaa.gov/psd/data/gridded/data.noaa.oisst.v2.html. [Accessed August 3, 2018]. → page 102

[136] R. W. Reynolds, N. A. Rayner, T. M. Smith, D. C. Stokes, and W. Wang, "An improved in situ and satellite sst analysis for climate," *Journal of climate*, vol. 15, no. 13, pp. 1609–1625, 2002. → page 102

[137] NOAA, "Noaa's precipitation reconstruction dataset (prec)." https://www.esrl.noaa.gov/psd/data/gridded/data.prec.html. [Accessed Sep 29, 2018]. → page 102

[138] M. Chen, P. Xie, J. E. Janowiak, and P. A. Arkin, "Global land precipitation: A 50-yr monthly analysis based on gauge observations," *Journal of Hydrometeorology*, vol. 3, no. 3, pp. 249–266, 2002. → page 102

[139] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993. → page 103

[140] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012. → page 103

[141] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008. → page 103

[142] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45–55, 2011. → page 114

[143] "Bluerov2 by blue robotics." http://docs.bluerobotics.com/brov2/. Accessed: 2016. → page 114

[144] "Heron by clearpath robotics." http://docs.bluerobotics.com/brov2/. Accessed: 2016. → page 114

[145] S. Haykin, *Neural networks*, vol. 2. Prentice hall New York, 1994. → page 127

[146] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[147] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015. → page 127

[148] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006. → page 128

[149] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[150] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai*, vol. 593598, 2002. → page 128

[151] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009. → page 128