Multi-Robot Information Gathering for Spatiotemporal Environment Modelling

Siva Kailas

CMU-RI-TR-23-64

August 15, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Prof. Katia Sycara, *chair*
Prof. George Kantor
Sha Yi

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

This material is based upon work supported by the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute: for Resilient Agriculture, Award No. 2021-67021-35329.

Abstract

Learning to predict or forecast spatiotemporal (ST) environmental processes from a sparse set of samples collected autonomously is a difficult task from both a sampling perspective (collecting the best sparse samples) and from a learning perspective (predicting unseen locations or forecasting the next timestep). We investigate two avenues of work concerning this problem. Firstly, we investigate coordinating a team of robots to adaptively sample a spatiotemporal environment to procure a dataset for learning a parametric neural model for forecasting. Recent work in spatiotemporal process learning focuses on using deep learning to forecast from dense samples. Moreover, collecting the best set of sparse samples is understudied within robotics. An example of this is robotic sampling for information gathering, such as using UAVs/UGVs for weather monitoring. Thus, we propose a methodology that leverages a neural methodology called Recurrent Neural Processes to learn spatiotemporal environmental dynamics for forecasting from selective samples gathered by a team of robots using a mixture of Gaussian Processes model in an online learning fashion. Thus, we combine two learning paradigms in that we use an active learning approach to adaptively gather informative samples and a supervised learning approach to capture and predict complex spatiotemporal environmental phenomena. Secondly, we investigate the multi-robot informative path planning problem by leveraging a multi-robot spatiotemporal adaptive sampling scheme and integrating informative path planning in a coordinated manner. Thus, we focus on investigating the sample collection process via multi-robot informative path planning. We present an approach for incorporating multi-robot informative path planning into a spatiotemporal adaptive sampling framework. We demonstrate this by modifying our previous methodology to consider path length constraints for sampling location selection. We also incorporate informative path planning to determine the best path to collect samples along while en route to collecting the desired sample. We achieve this in a decentralized manner by decoupling the process into two stages: the first stage uses our spatiotemporal mixture of Gaussian Processes (STMGP) model to determine the most informative sampling location via a mutual information lower bound heuristic and the second stage plans an informative path to collect the desired sample and other additional informative samples via submodular function optimization. Moreover, we effectively leverage peer-to-peer communication to enable coordination. Simulation results are provided to validate the effectiveness of our proposed approaches.

Funding

This material is based upon work supported by the AI Research Institutes program supported by NSF and USDA-NIFA under AI Institute: for Resilient Agriculture, Award No. 2021-67021-35329.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Spatiotemporal environmental phenomena are present in many domains. The ability to predict and forecast spatiotemporal processes would enable us to have a deeper understanding of numerous naturally-occurring environmental processes, including biological, chemical, and physical phenomena. It is often considered to be difficult to capture spatiotemporal phenomena from static or fixed sensors since informative sampling locations often change over time due to the spatiotemporal dynamics. As a result, a large number of static or fixed sensors are often needed to adequately sample a spatiotemporal environmental phenomena. Thus, mobile robots are typically considered to be much more capable for collecting samples in these types of environments. This is because mobile robots can be used in an autonomous manner to navigate spatially across an environment for multiple timesteps. Thus, they can adaptively adjust the sampling location at each timestep based on the spatiotemporal dynamics. Moreover, these mobile robots can be equipped with sensor payloads appropriate for different environments. This makes mobile robots an excellent candidate for applications such as climate monitoring or wildlife tracking. However, a persistent challenge in enabling mobile robots to be used for these domains is the lack of algorithmic approaches for coordinating mobile robot teams to leverage their capabilities to adaptively sample and plan trajectories in such environments.

There are a couple of difficulties associated with producing an coordinated multi-

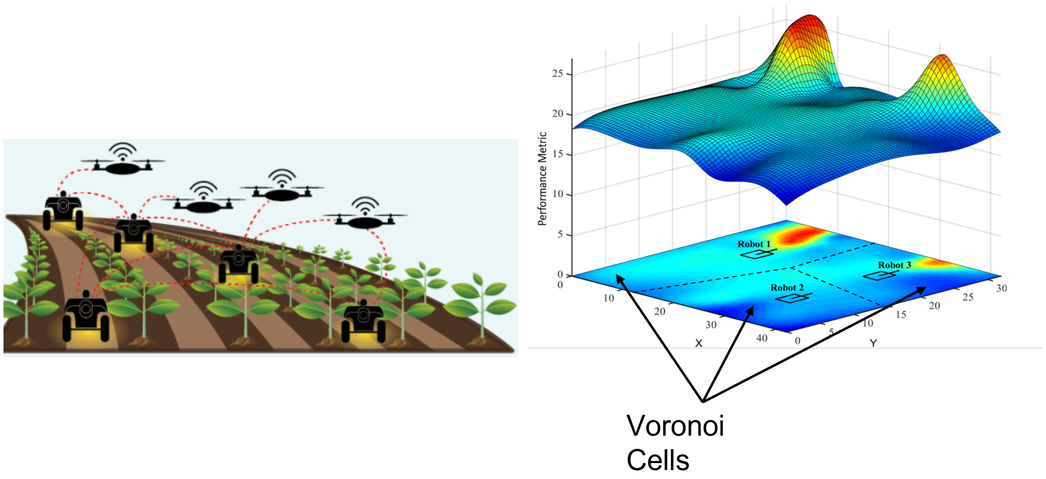

Figure 1.1: (Left) Visualization of multi-robot adaptive sampling in an agricultural scenario. (Right) Visualization of adaptive sampling where three robots are deployed into unknown environment and need to choose where to sample in the current timestep. Note that the environment density function changes at each timestep (i.e. the environment is spatiotemporally dynamic)

robot or multi-agent system to adaptively sample a spatiotemporal environment. The first difficulty that arises is learning a model that captures the complex, nonlinear dynamics associated with spatiotemporal phenomena. One category of approaches involves learning a model in an online fashion (i.e. learning the model while collecting the samples). These approaches tend to fall under the category of active learning or online learning. Another set of approaches involve learning a model from a dataset (i.e. learning the model after collecting the samples). These approaches tend to fall under the category of (strongly) supervised learning. The second difficulty that arises is how to collect informative samples. This entails waypoint selection or path planning for collecting samples or both. A plethora of prior studies have focused on achieving approximately optimal or suboptimal approaches for spatially-correlated time-invariant environments. In these studies, the robot or team of robots explores a static environment and collects more and more samples in this environment, and performance is measured by the efficacy of the sampling for learning an accurate model compared to exhaustive sampling.

In this thesis, we will investigate these issues for spatiotemporal environments. In chapter 3, we look into integrating multi-robot active learning and supervised

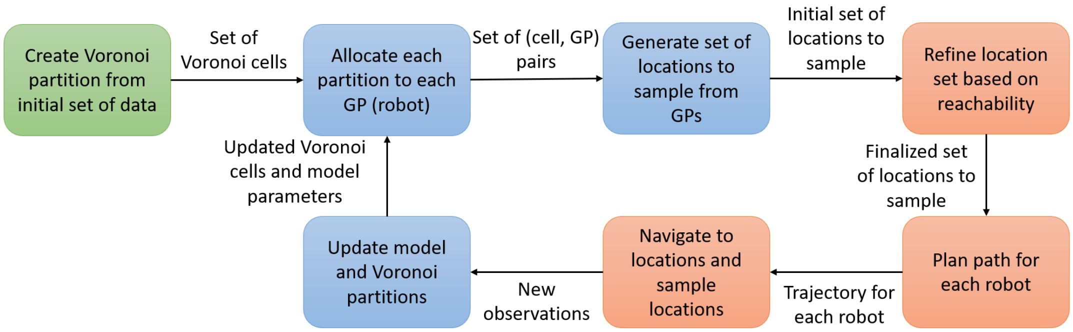
Create Voronoi partition from initial set of data
Set of Voronoi cells
Allocate each partition to each GP (robot)
Set of (cell, GP) pairs
Generate set of locations to sample from GPs
Initial set of locations to sample
Refine location set based on reachability
Finalized set of locations to sample
Plan path for each robot
Trajectory for each robot
Navigate to locations and sample locations
New observations
Update model and Voronoi partitions
Updated Voronoi cells and model parameters

Figure 1.2: General approach for multi-robot information gathering in this thesis. Although the work in both chapter 3 and chapter 4 include most or all of these components, the work in chapter 3 is roughly more focused on the components in blue, while the work in chapter 4 is roughly more focused on the components in red.

learning for forecasting a spatiotemporal environmental phenomena. In chapter 4, we look into a two-stage strategy for integrating spatiotemporal adaptive sampling with informative path planning for predicting unseen locations in a spatiotemporal environment.

1.2 Thesis Outline

We begin with a discussion of background and related works in chapter 2 to introduce some preliminary material and provide context regarding the field of robotic information gathering. The core of this thesis disucusses two different formulations and approaches for multi-robot information gathering in spatiotemporal environments. The first work presents an end-to-end methodology that uses effective and coordinated multi-robot information gathering to adaptively sample and procure a collection of samples as a dataset used to train a parametric spatiotemporal model. This is discussed in chapter 3. The second work deals with how to integrate spatiotemporal adaptive sampling and informative path planning to deal with path length constraints to predict unseen locations in a spatiotemporal environment. This is discussed in chapter 4. Finally, we offer some directions of work that are either actively being pursued currently or will be addressed in the future in chapter 5.

Chapter 2

Background and Related Works

2.1 Background

2.1.1 Active Learning, Online Learning, and Supervised Learning

Many learning problems in robotics have been addressed via (strongly) supervised learning. Supervised learning has provided incredible results both in the field of robotics as well as many other domains, especially through the use of neural networks. However, many robotic information gathering tasks are formulated within the scope of active learning. Active learning refers to a learner choosing which data to collect labels for such that it efficiently learns an accurate predictive model for unlabelled data. In this regard, active learning is a form of weakly supervised learning. Since robots have to sample the environment to collect environmental phenomena values (labels) for each location and timestep, the robotic information gathering problem often is addressed via active learning algorithms. Moreover, since the sampling process is temporal in nature as well, samples are often collected in a stream. Thus, the methodologies proposed for active robotic information gathering often follow an online learning paradigm as well, since online learning refers to learning techniques that update the model as data is collected sequentially.

2.1.2 Gaussian Processes and Neural Processes

Gaussian Processes are a popular methodology used to perform nonparametric regression. A Gaussian Process (GP) is a stochastic process that can be viewed as a distribution over functions. Gaussian Processes are often useful for active learning scenarios such as robotic information gathering since they are non-parametric and can be inferred directly from the data. Moreover, Gaussian Processes can provide both a point estimate (mean) and an uncertainty estimate (variance). This is especially useful for sampling strategies that incorporate model uncertainty to collect samples that improve the predictive performance of the model. Gaussian Processes are fully specified by a mean function m and covariance function k as shown in Equation (2.1).

$$f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot)) \tag{2.1}$$

For sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$, the covariance matrix $K_{A,B}$ can be defined as shown in Equation 2.2.

$$K_{A,B} = \begin{bmatrix} k(a_1, b_1) & \ldots & k(a_1, b_n) \\ \vdots & \ddots & \vdots \\ k(a_m, b_1) & \ldots & k(a_m, b_n) \end{bmatrix} \tag{2.2}$$

Given unlabelled test data x and a labelled training data X, Y, we can represent the probability $\mathcal{P}(f(x_i)|X, Y, x_i)$ as shown in Equation (2.3)-(2.5).

$$\mathcal{P}(f(x)|X, x) \sim \mathcal{N}(\mu(x), K(x)) \tag{2.3}$$

$$\mu(x) = K_{x,X} K_{X,X}^{-1} Y \tag{2.4}$$

$$K(x) = K_{x,x} - K_{x,X} K_{X,X}^{-1} K_{x,X}^{\top} \tag{2.5}$$

Thus, we can leverage Gaussian processes to provide point estimates (means) and an uncertainty estimates (variances) on unseen data. Moreover, the regression can be directly inferred from the data without the need of iterative or descent-based methods. However, it should be noted that regression and inference incur a $\mathcal{O}(n^3)$ complexity with respect to the dataset size n. Thus, GPs tend to scale poorly with increased addition of data from a computational complexity perspective. Moreover, although GPs are non-parameteric (i.e does not assume a fixed set of parameters a priori), GPs still have a limit in their representational capacity. In contrast, neural networks tend to be a rich class of models in terms of representational capacity despite being parametric, but often only provide point estimates without any uncertainty quantification.

Neural processes address the gap between Gaussian processes and neural networks. While the neural process is a parametric neural network model, it can estimate a distribution over a test set of points conditioned on a training set of points. Neural models often use an encoder module to compute a latent distribution $q(z|X,Y)$ from the training data. This latent distribution is often a factorized Gaussian distribution. Let $z_1, \dots, z_K \sim q(z|X,Y)$ be a set of samples drawn from the latent distribution. Then, the neural process model employs a decoder module to produce a set of predictions $y_1, \dots, y_K$ from $z_1, \dots, z_K$ and the test set. Then a point estimate (mean) and uncertainty estimate (variance) can be computed as shown in Equation (2.6)-(2.7).

$$\mu(x) = \frac{1}{K}\sum_k y_k \tag{2.6}$$

$$K(x) = \frac{1}{K-1}\sum_k (y_k - \mu(x))(y_k - \mu(x))^\top \tag{2.7}$$

However, while the above model does address some of the gaps regarding the rich representational capacity of neural networks and the uncertainty estimation capability of GPs, this model still requires iterative gradient descent-based approaches to learn the parameters from the data. Thus, the neural model and variants of the neural model cannot directly be used in robotic information gathering as is since there is a need to still procure the data due to the active learning and online learning nature of the problem.

2.1.3 Mutual Information

For robotic information gathering, an acquisition function or sampling heuristic is often necessary to optimize which samples to choose. Information-theoretic criterion are often useful in these scenarios since choosing informative samples can help improve the model performance. While many such information-theoretic criterion exist, such as entropy-based or Fischer information-based metrics, we rely on mutual information. Mutual information measures the mutual dependence between two variables, say X and Y. Formally, mutual information can be defined as shown in Equation (2.8).

$$\begin{aligned} I(X,Y) &= D_{\mathrm{KL}}(P_{X,Y} \| P_X \otimes P_Y) = \mathbb{E}_{P_{X,Y}}\left[\log\left(\frac{P_{X,Y}}{P_X P_Y}\right)\right] \\ &= \mathbb{E}_{P_{X,Y}}\left[\log\left(\frac{P_{X|Y}}{P_X}\right)\right] = \mathbb{E}_{P_{X,Y}}\left[\log\left(\frac{P_{Y|X}}{P_Y}\right)\right] \end{aligned} \tag{2.8}$$

While Equation (2.8) shows the standard formulation of mutual information, we utilize a different formulation of mutual information that involves entropy $H(\cdot)$ as shown in Equation (2.9).

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{2.9}$$

In general, computing mutual information can be difficult and certain optimization problems encountered in this thesis involving mutual information calculation can yield combinatorial, intractable problems. However, using the formulation of mutual information shown in Equation (2.9) will yield a submodular optimization function in one of the optimization problems encountered in this thesis, which is particularly useful for computing the mutual information.

2.2 Related Works

Existing methods for the adaptive sampling and informative path planning problems for a single robot include recursive-greedy approaches [35], differential entropy-based approaches [15], and generation of an informative path [6]. Other work leverages a Gaussian process motivated by an information-theoretic hierarchical structure [20], or exploration of branch and bound techniques in conjunction with a Gaussian process for environment modelling and informative path planning [1]. The use of more computationally efficient approximates to Gaussian processes, such as sparse Gaussian processes, have been explored within the context of single agent informative planning

[21]. Most of these works have been limited to single robot and/or single Gaussian Process. For example, [27] proposed a decentralized active sensing approach using a mixture of Gaussian processes that affect the same process component. This allows for robots to coordinate their actions when they are affecting the same Gaussian process component [27]. Other work investigated a decentralized multi-robot informative adaptive sampling approach for uni-model GP by using dynamic Voronoi partitions to coordinate robot actions [11].

Many approaches leverage myopic horizons to allow for replanning, in that they maximize the information-criterion over a short lookahead or horizon. One approach is to use receding horizons. The work in [8] proposed a receding horizon approach that satisfies temporal logic specification for domains where safety and reliability are critical. The work in [39] proposed a receding horizon approach that modifies the lookahead step size to avoid local optimas in favor of better optimas. Receding horizon methods plan optimal paths within the myopic horizon reactively and do not maximally utilize prior environment information. Nonmyopic approaches typically attempt to achieve long-term horizon optimization, which often means they do not replan as often. The work in [38] proposed a nonmyopic approach by reformulating the informative path planning problem as a Global Kriging Variance Minimization problem. The work in [26] used a Gaussian mixture model to identify information clusters, which are each allocated search time, and then use a modified version of Monte Carlo Tree Search to generate long plans without incurring exhaustive search. An informed sampling-based approach was shown to be effective in the work by [24], especially for large and higher dimensional spaces.

A recent line of work has emerged that utilizes reinforcement learning to address the adaptive sampling and informative path planning problem into a reinforcement learning context. The work in [31] cast the static informative path planning problem into a reinforcement learning context and leverage reinforcement learning and Monte Carlo Tree Search. Both [28] and [29] addressed the multi-robot adaptive sampling problem with multi-agent reinforcement learning.

Among the aforementioned work, only a few address the multi-agent or multi-robot adaptive sampling and informative path planning problem. In addition to this, none of the aforementioned work addresses spatiotemporal environments. Some preliminary studies have been investigated for adaptive sampling in spatiotemporal environments.

For example, [7] use a combination of historical and predicted data from a model to address the spatiotemporal environment, while [25] leverage an already trained neural network in conjunction with an extended Kalman filter-based adaptive sampling regime. The work in [32] collect samples over an entire region or partition at each timestep. Essentially, these works makes some assumptions that are divergent from the standard adaptive sampling and informative path planning problem formulations to reduce the difficulty of dealing with spatiotemporal environments. In this thesis, we assume samples can only be collected at the current location of each robot and that we do not have access to entire forecasted knowledge of future timesteps or already trained neural network models. This is more akin to the standard adaptive sampling and informative path planning problem formulations that have been investigated in prior work. The combination of leveraging a multi-robot system for learning and modelling a spatiotemporal environment is relatively understudied. Thus, we investigate this broad, difficult, and understudied robotics problem in the remainder of this thesis.

Chapter 3

Multi-Robot Adaptive Sampling for Supervised Spatiotemporal Environment Forecasting

3.1 Introduction

Multi-robot systems can be used in a variety of complex tasks, which enable them to be used in applications such as coverage, sampling, and exploration in unknown environments [4, 16, 17, 19, 22] A notable challenge in multi-robot systems is the multi-robot information gathering problem, which encompasses a variety of formulations including multi-robot adaptive sampling [28, 40], multi-robot sensor coverage [16, 19], and multi-robot informative path planning [5]. However, learning to forecast spatiotemporal environmental processes is relatively understudied within the context of robotics, let alone multi-robot systems. While there is prior work in deep learning for spatiotemporal process learning in domains such as high-frequency trading and video surveillance, these works often investigate learning from a set of timesteps where each timestep contains the entire spatial context [23, 33, 34, 36]. Such approaches may not be useful in cases where we can only gather sparse amounts of data at each timestep.

Thus, in this work, we are particularly interested in such a variant of the multi-robot adaptive sampling problem, in which a group of robots are deployed in an environment given random starting configurations and then seek to gather the best samples in the environment for learning a parametric (i.e. deep learning-based) spatiotemporal forecasting model. This problem formulation is different from prior work in a couple of ways. Firstly, we address the problem of multi-robot adaptive sampling, which involves coordinating a team of robots effectively to sample an environmental phenomena. Most of the prior work in adaptive sampling has been limited to single robot and/or single Gaussian Process (GP) [1, 6, 20, 21, 35]. Secondly, we consider a environment with an arbitrary spatiotemporal process. Spatiotemporal processes depend on both the spatial features of the underlying phenomena and time, but a large amount of prior work in multi-robot adaptive sampling only addresses spatially correlated time-invariant processes [11, 16, 19, 27]. Thirdly, we consider determining samples for learning a parametric spatiotemporal model for forecasting. Recent work in adaptive sampling often rely solely on non-parametric learning methods for modelling the environment phenomena, which is often restrictive and can only handle limited spatiotemporal evolution rates [1, 11, 16, 19, 20, 21, 27]. Moreover, parametric learning methods can often scale better with data and have powerful representational capacity, making them an interesting modelling approach for arbitrary spatiotemporal environment phenomena.

There are many real-world motivations for considering this multi-robot adaptive sampling problem formulation. Multiple UAVs/UGVs can be used to monitor climate phenomena for better weather forecasts or to track animals for wildlife monitoring. This multi-robot adaptive sampling problem formulation is also applicable to wildfire monitoring or agricultural yield forecasting. The use of parametric or deep learning models could be introduced in all the aforementioned domains if an approach was able to successfully integrate coordinated spatiotemporal multi-robot sampling to gather information to train or generate these learned spatiotemporal models. Towards realizing this goal, we propose an integrated methodology for learning a forecasting model that leverages 2 components: (1) a neural methodology called Recurrent Neural Processes (RNP); and (2) a coordinated multi-agent spatiotemporal adaptive sampling methodology via a mixture of Gaussian Processes (MGPs) to collect the best samples for training the RNP for accurate forecasting. In essence, we propose to leverage an active learning-based methodology (such as MGPs) to procure an informative set of samples that provides the data distribution for enabling supervised learning-based methods (such as RNPs) to be used for spatiotemporal environment learning, modelling, and forecasting.

Our **main contribution** in this work is an end-to-end methodology that starts with effective and coordinated multi-robot information gathering to adaptively sample and procure a collection of samples as a dataset used to train a parametric spatiotemporal model. Moreover, our approach is able to select highly informative samples that improve the predictive performance of the spatiotemporal model. To the best of our knowledge, this is the first work to address integrating multi-robot information gathering with deep learning for spatiotemporal model learning, as prior work often chooses to focus on one or the other.

3.2 Problem Formulation

Consider a set of n robots moving in a bounded environment $Q \subset \mathbb{R}^2$ and assume the environment can be discretized into a set of points $q \in Q$. Moreover, let $T \subset \mathbb{R}$ be the time range of interest and assume that this has been discretized into a set of timesteps $t \in T$. With this, the position of each robot $i \in \{1, \dots, n\}$ at timestep t can be denoted by $x_i^t \in Q$. We assume the environment is free of obstacles and can be partitioned into n Voronoi cells at any timestep t where each Voronoi cell V_i^t is defined in (3.1).

$$V_i^t = \{q \in Q : ||q - x_i^t||_2 \leq ||q - x_j^t||_2 \forall j \neq i\} \tag{3.1}$$

This is a common assumption made in robotic information gathering that is reasonable for most situations in robot exploration [11, 16, 19]. Each Voronoi cell V_i^t corresponds to robot i, meaning robot i is allocated that space at timestep t. This will be leveraged in the proposed active learning methodology to avoid collisions and improve initial environmental modelling. The current location of the robot x_i is sampled and added to its dataset to be used to adaptively choose the next location.

Regarding the distribution of environmental phenomenon at each point of interest q at time t, there exists an unknown density function $\phi(q, t) : Q \times T \rightarrow \mathbb{R}$ that maps the location q and time t to the scalar value of the phenomena $\phi(q, t)$. That is, the unknown environmental phenomena is both spatial and time varying since it maps each spatial location q and timestep t to a real-valued density value, such as air temperature, animal presence, etc.

Now, let $X_t = \Big[(q_1, t), \dots (q_m, t)\Big]$ be the vector of candidate sampling locations recommended at timestep t and $X_{\{t_p:t_{p+k}\}}$, which represents the concatenation of the vectors of m candidate sampling locations chosen at each of the k timesteps from timestep t_p and t_{p+k}, be defined as shown in Equation (3.2) where $\bigoplus$ represent vector concatenation.

$$X_{\{t_p:t_{p+k}\}} = \bigoplus_{t \in \{t_p:t_{p+k}\}} X_t \tag{3.2}$$

Furthermore, let $F_X(t_p : t_{p+k}, q) : X_{\{t_p:t_{p+k}\}} \to \mathbb{R}$ be a parameterized model inferred or trained from data X that estimates or approximates $\phi(q, p + k + 1)$. Then, we propose to address the optimization problem shown in Equation (3.3).

$$\underset{X}{\operatorname{argmin}} \sum_{q \in Q} \sum_{p \in T} (F_X(t_p : t_{p+k}, q) - \phi(q, p + k + 1))^2 \tag{3.3}$$

Note that the robots or agents must procure the data X in a single-shot manner, which is akin to actual robotic information gathering. That is, given a window of timesteps in the spatiotemporal environment, each robot or agent only gets to collect m samples at each timestep in the environment, and the agents or robots must proceed sequentially in time (i.e. they cannot move revisit a previous timestep). Thus, it is imperative that the robots or agents coordinate with one another in order to choose the best samples given a minuscule amount of initial data.

We choose to parameterize F as a recurrent neural process, an extension of neural processes for handling spatial and temporal dependencies in data [13]. However, any parametric model designed for spatiotemporal modelling can be chosen if desired. However, the question to address is how to coordinate a team of agents or robots to choose a good set of samples at each timestep to collect into dataset X that will likely improve forecasting performance for a parametric spatiotemporal model.

To address this, we propose to use a spatiotemporal mixture of Gaussian Processes (STMGP) where each Gaussian Process component uses a squared exponential kernel with automatic relevance determination to gather data in an adaptive sampling manner to approximately solve the optimization problem shown in Equation (3.3).

At each timestep, the robots use the STMGP model to inform the next timestep sampling locations and move towards those locations. The STMGP model then updates itself with the collected samples in a distributed manner and produces a set of samples to the recurrent neural process to learn a forecasting model.

Thus, we start with an active learning methodology (robotic information gathering) and then leverage a supervised learning approach (use of gathered data for model learning) to capture the spatiotemporal dynamics of the environment for forecasting. This ultimately provides a step towards bridging the gap between robotic information gathering and powerful parametric models for spatiotemporal environment modelling.

3.3 Proposed Approach

We now present a high-level representation of our approach described in Algorithm 1. Given 1% of the data as initial data randomly (Line 1) and random starting configurations (Line 2 - 3), the robots first sample their current location (Line 8), collect that sample into their local datasets (Line 9), update their local GP components (Line 10), and compute a mixture of their local GP parameters (Line 12). The robots use the spatiotemporal mixture of GPs to infer the mean and variance of each location in the environment at the current timestep (Line 13), and use this inference to determine where to sample in the next timestep (Line 14). The new sampling locations are used to adjust the Voronoi partitions and the local datasets of each robot (Lines 16 - 17) via communication. The robots also use a uni-model GP to compute an approximate covariance matrix across all locations at the current timestep (Line 18), which is used to inform which samples to procure as part of the dataset for training the RNP (Line 19). This is repeated for the allotted number of timesteps in the environment, where the environment changes after each timestep in accordance to ϕ. Finally, the procured dataset is used to train the RNP to accurately forecast (Line 21). The RNP is then evaluated on unseen timesteps in the environment to determine how well the RNP captured the spatiotemporal phenomena for forecasting (Line 22).

```
Algorithm 1 STMGP + RNP
Require: $k$ {$k$ is time window size}
Require: $m$ {$m$ is number of samples per timestep to provide to RNP}
$\mathbf{X}_i, \mathbf{Y}_i \leftarrow$ GetRandomInitialData($1\%$)
$q_t =$ GetRandomStartingLocations()
$V_t =$ UpdateVoronoiPartitions($q_t$)
$X_C, Y_C \leftarrow \emptyset, \emptyset$
$\hat{\gamma}_t \leftarrow \mathbf{0}$ {$\hat{\gamma}_t(i)$ is the mutual information lower bound for robot $i$}
for $t = t_p : t_{p+k}$ do
    for each robot $i$ do
        $x_i, y_i \leftarrow$ SampleLocation($q_t(i)$) {returns $\phi(q_t(i), t) + \epsilon$}
        $\mathbf{X}_i, Y_i \leftarrow \mathbf{X}_i \cup x_i, \mathbf{Y}_i \cup y_i$
        $GP_i \leftarrow$ FitGP($\mathbf{X}_i, \mathbf{Y}_i$)
    end for
    $MGP_t \leftarrow$ MixGPs($GP_i$) {section 3.4.2}
    $\mu^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i}, \sigma^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i} \leftarrow MGP_t.\mu, MGP_t.\sigma$ {section 3.4.2}
    $q_t \leftarrow \text{argmax}_{q_i^*} \sqrt{\sigma^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}} - \sqrt{\hat{\gamma}_{t-1}}$ {section 3.4.3}
    $\hat{\gamma}_t \leftarrow \hat{\gamma}_{t-1} + \sigma^*_{q_t|\mathbf{X}_i,\mathbf{Y}_i}$ {section 3.4.3}
    $V_t =$ UpdateVoronoiPartitions($q_t$)
    $\mathbf{X}_i, \mathbf{Y}_i \leftarrow$ ExchangeSamplesBasedOnVoronoiPartitions($\mathbf{X}_i, \mathbf{Y}_i, V_t$)
    $\tilde{K} \leftarrow$ FitGP($\bigcup_i \mathbf{X}_i, \bigcup_i \mathbf{Y}_i$).$K$
    $X_C(t), Y_C(t) \leftarrow$ StreamSubmodularSecretary($\tilde{K}, m$) {section 3.4.3}
end for
$F_{X_C,Y_C} \leftarrow$ AddSamplesToDatasetAndTrainRNP($\{X_C, Y_C\}$)
EvaluateOnUnseenData($F_{X_C,Y_C}$)
```

This approach modifies and integrates various components, and we describe each component, its modifications, and its role in the integrated system in the subsections below. Algorithm 1 also details where each subsection corresponds to in the integrated system.

3.4 Spatiotemporal Mixture of Gaussian Processes

We now outline the active learning methodology used to procure a set of samples used for learning a parametric spatiotemporal environment model for forecasting. We first present a simple spatiotempoal Gaussian Process formulation in subsection 3.4.1 and then present how we produce a mixture of these spatiotemporal GPs in subsection 3.4.2

3.4.1 Gaussian Processes

Gaussian Processes are a widely used probabilistic model that defines distributions over functions. They have been a common approach for modeling spatial phenomena since they allow for modeling the hidden mapping from training data to the target phenomenon with consideration of uncertainty.

To use GP regression, we first make the assumption that the target phenomena follows a multivariate joint Gaussian distribution. Given this assumption and a collection of observations from the target phenomena $\phi(q,t)$, a learned GP model produces an estimated Gaussian probability distribution of $\phi(q,t)$, denoted by the function $\mu(q,t) = \mathbb{E}[\phi(q,t)]$. The learned GP model also produces a covariance function $k((q,t),(q',t')) = \mathbb{E}[(\phi(q,t) - \mu(q,t))^\top(\phi(q',t') - \mu(q',t'))]$.

Recall from section 3.2 that $X_t = \Big[(q_1,t),\ldots(q_m,t)\Big]$ is the vector of candidate sampling locations recommended at timestep t and $X_{\{t_p:t_{p+k}\}} = \bigoplus_{t\in\{t_p:t_{p+k}\}} X_t$ is the concatenation of the vectors of m candidate sampling locations chosen at each of the k timesteps from timestep t_p and t_{p+k} where $\bigoplus$ represent vector concatenation. Furthermore, let $Y_t = \Big[y_1,\ldots,y_m\Big] \in \mathbb{R}$ be the vector such that $y_i = \phi(q_i,t) + \epsilon$ where $\epsilon \sim \mathcal{N}(0,\sigma_n^2)$ and σ_n^2 is the variance of the noise. This denotes that the observed values sampled by each agent or robot from the environment is noisy. We can then define $Y_{\{t_p:t_{p+k}\}}$ as shown in Equation (3.4)

$$Y_{\{t_p:t_{p+k}\}} = \bigoplus_{t\in\{t_p:t_{p+k}\}} Y_t \tag{3.4}$$

For conciseness, we shall denote $\mathbf{X} = X_{\{t_p:t_{p+k}\}}$ and $\mathbf{Y} = Y_{\{t_p:t_{p+k}\}}$. Using this, we can represent the mean vector $\mu_{\mathbf{X}}$ and covariance matrix $K_{\mathbf{X},\mathbf{X}}$ as shown in Equation (3.5) and Equation (3.6) respectively.

$$\mu_{\mathbf{X}} = \Big[\mu(q_1, t_p), \ldots, \mu(q'_m, t_{p+k})\Big] \tag{3.5}$$

$$K_{\mathbf{X},\mathbf{X}} = \begin{bmatrix} k((q_1,t_p),(q_1,t_p)) & \ldots & k((q_1,t_p),(q'_m,t_{p+k})) \\ \vdots & \ddots & \vdots \\ k((q'_m,t_{p+k}),(q_1,t_p)) & \ldots & k((q'_m,t_{p+k}),(q'_m,t_{p+k})) \end{bmatrix} \tag{3.6}$$

Thus, the joint probability distribution over the output value of a vector of sampling points can be written as shown in Equation (3.7).

$$\mathbf{Y} \sim \mathcal{N}(\mu_{\mathbf{X}}, K_{\mathbf{X},\mathbf{X}}) \tag{3.7}$$

Now, let $X_{t_{p+k+1}} = \Big[(q''_1, t_{p+k+1}), \ldots (q''_M, t_{p+k+1})\Big]$ denote the set of all locations Q at timestep t_{p+k+1} where $\{q''_1, \ldots, q''_M\} = \{q : \forall q \in Q\}$. Furthermore, let $Y_{t_{p+k+1}}$ be the unknown values of the environment at timestep t_{p+k+1} that we wish to estimate. For conciseness, we shall denote $\mathbf{X}^* = X_{t_{p+k+1}}$ and $\mathbf{Y}^* = Y_{t_{p+k+1}}$. The posterior mean vector $\mu_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}}$ and covariance matrix $K_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}}$ can be generated as shown in Equation (3.8) and Equation (3.9) respectively.

$$\mu_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}} = K_{\mathbf{X}^*,\mathbf{X}} K^{-1}_{\mathbf{X},\mathbf{X}} \mathbf{Y} \tag{3.8}$$

$$K_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}} = K_{\mathbf{X}^*,\mathbf{X}^*} - K_{\mathbf{X}^*,\mathbf{X}} K^{-1}_{\mathbf{X},\mathbf{X}} K^{\top}_{\mathbf{X}^*,\mathbf{X}} \tag{3.9}$$

Thus, the posterior target distribution of target set $\mathbf{X}^*$ conditioned on $\mathbf{X}$ is given in Equation (3.10).

$$\mathbf{Y}^* \sim \mathcal{N}(\mu_{\mathbf{X}^*|\mathbf{X}}, K_{\mathbf{X}^*|\mathbf{X}}) \tag{3.10}$$

The aforementioned structure defines a uni-model Gaussian Process and provides good generalization for learning density functions. However, there are representational limits to a uni-model Gaussian Process. We can improve the representational capacity by leveraging a set of local Gaussian Processes. Moreover, if we can compute this mixture of Gaussian Processes in a decentralized fashion, we can incorporate this into a multi-robot or multi-agent system. This motivates subsection 3.4.2.

3.4.2 Mixture of Gaussian Processes

Our approach leverages a Mixture of Gaussian Processes (MGPs) for adaptive sampling in an environment Q. Here, each agent i has a learned local Gaussian Process component GP_i, which yields a set of m GP components $\{GP_1, \ldots GP_m\}$ and an associated probability function $P(q, i_g) = P(q \in Q \text{ is best described by } GP_{i_g})$. Suppose the agent i has sampled locations $\mathbf{X}_i$ and suppose $\mathbf{Y}_i$ are the associated ground-truth values sampled from locations $\mathbf{X}_i$. Moreover, suppose agent i has a mean $\mu_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ and covariance matrix $K_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ as well. Now, let $x^* \in \mathbf{X^*}_i$. Furthermore, let $\mu_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i} \in \mu_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ and $\sigma_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i} \in diag(K_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i})$. We can represent the local conditional posterior mixture mean $\mu^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i}$ as shown in Equation (3.11).

$$\mu^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i} = \sum_{i=1}^{m} P(q, i_g) * \mu_{x^*_i|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} \tag{3.11}$$

We can compute an intermediate term for determining the local conditional posterior variance as shown in Equation (3.12).

$$diff = (\mu_{x^*_i|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} - \mu^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i})^2 \tag{3.12}$$

Using the intermediate term in Equation (3.12), we can represent the local conditional posterior mixture variance $\sigma^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i}$ as shown in Equation (3.13).

$$\sigma^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i} = \sum_{i=1}^{m} P(q, i_g) * (\sigma_{x^*_i|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} + diff) \tag{3.13}$$

Here, $\mathbf{X}_{i_g} \subset \mathbf{X}_i$ and $\mathbf{Y}_{i_g} \subset \mathbf{Y}_i$, and each represents the set of samples and ground-truth values that can be best described by GP_{i_g} (i.e. $\mathbf{X}_{i_g} = \{q \in \mathbf{X}_i : \arg\max_{i'_g} P(q, i'_g) = i_g\}$. The above mixture procedure can be approximately carried out in a decentralized manner, which we do based on [19].

Depending on the goal of the adaptive sampling, we may either wish to maximize the mean ($x^* = \text{argmax}_{x^*_i} \mu^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i}$), variance ($x^* = \text{argmax}_{x^*_i} \sigma^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i}$), or a combination of the two similar to an upper confidence bound heuristic ($x^* = \text{argmax}_{x^*_i} \mu^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i} + \beta\sigma^*_{x^*_i|\mathbf{X}_i,\mathbf{Y}_i}$). Since we wish to address reducing the mean squared error as shown in Equation (3.3), we choose to leverage mutual information as our sampling heuristic. We describe this in subsection 3.4.3.

3.4.3 Mutual Information for Sampling

There are two sampling aspects that are occuring. The first aspect is each agent choosing which sample to collect for updating the mixture of GPs model. The second aspect is choosing which samples to recommend for collection to provide to the parametric spatiotemporal model for training. We inform both sampling aspects via a mutual information-based criterion.

Mutual Information for Updating Mixture of GPs

Calculating mutual information is known to be difficult, but in order to have the most accurate representation of the environment, each agent should choose the most informative sample. Thus, for optimizing each local Gaussian Process component, we choose a sample that optimizes for a lower bound on the mutual information between the current local Gaussian Process GP_i and the noisy observations to be collected $\mathbf{Y}_i$ from the sampling locations $\mathbf{X}_i$ at timestep t. Let $\gamma_t = \max_{\mathbf{X}_i} I_t(GP_i, \mathbf{X}_i)$ where $I(\cdot)$ is the mutual information. For Gaussian Processes, the lower bound in Equation (3.14) holds.

$$\hat{\gamma}_t = \sum_{t'=1}^{t} \sigma^2_{t'}(x_{t'}) \leq \frac{2}{\log(1 + \sigma_n^{-2})} \gamma_t \tag{3.14}$$

The lower bounded quantity $\hat{\gamma}_t$ can be computed as shown in Equation (3.15), and optimizing for the lower bounded quantity $\hat{\gamma}_t$ yields the optimization strategy shown in Equation (3.16) for each agent.

$$\hat{\gamma}_t = \hat{\gamma}_{t-1} + \sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} \tag{3.15}$$

$$x^* = \operatorname*{argmax}_{x_i^*} \sqrt{\sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}} - \sqrt{\hat{\gamma}_{t-1}} \tag{3.16}$$

Mutual Information for Recommending Samples

At each timestep t, the agents must determine which sampling locations to sample and collect as data for training the parametric spatiotemporal model. We once again rely on utilizing mutual information to maximize the number of informative samples chosen. Once again, computation of mutual information is difficult. Moreover, instead of determining the mutual information with respect to a function (which was the case in subsubsection 3.4.3), we have to find the subset of locations that provides the maximal informativeness with respect to one another. Let $\Phi_t = \{\phi(q,t) : \forall q \in Q\}$ be the set of environment values at all the locations in the environment Q at timestep t. Moreover, let R_t be a set of c sampling points to procure as part of the dataset for learning the parametric spatiotemporal model and their associated environment density values. We wish to solve the optimization problem posed in Equation (3.17).

$$R_t^* = \underset{R_t}{\operatorname{argmax}} \, \mathcal{I}(R_t, \Phi_t \setminus R_t) = \underset{R_t}{\operatorname{argmax}} \, \mathcal{H}(\Phi_t \setminus R_t) - \mathcal{H}(\Phi_t \setminus R_t | R_t) \tag{3.17}$$

The problem in Equation (3.17) is a combinatorial, intractable problem [41]. However, mutual information gain is a monotone and submodular function [18, 41]. As a result, we leverage submodular function optimization to find a selection of sampling locations that is a provably good approximation to the optimal choice. Similar to [18, 41], we employ a greedy-based approach to avoid the combinatorial complexity of this optimization problem. In particular, we employ a stream-based secretary algorithm to approximately solve for the optimal R_t^* at each timestep t based on the algorithm in [18] shown in Algorithm 2. This ensures that a set of candidate samples can be chosen quickly and in a tractable manner prior to the next timestep.

We compute a near-optimal approximation of the mutual information $\tilde{\mathcal{I}}(\cdot)$ using submodular function optimization. To compute the approximation of mutual information via submodular function optimization in Algorithm 2, the covariance matrix $K = \Big[k(q_i, q_j)\Big]_{\forall q_i, q_j \in Q}$ is needed. We approximate this covariance matrix via a uni-model Gaussian Process fitted using the data collected by the agents and inferred on $\{q_i \in Q\}$.

Algorithm 2 Stream-based Submodular Secretary Algorithm [18]

```
Require: $\tilde{\mathcal{I}}(\cdot\,; \{q_i \in Q\}, K(\{q_i \in Q\}, \{q_j \in Q\}))$
$R_t \leftarrow \emptyset$
$r \leftarrow 0$
for each segment $S_l = \{q_i \in Q : \frac{n(l-1)}{k} < i \leq \frac{nl}{k}\}$ do
    for each $q_i \in S_l$ do
        if $\frac{n(l-1)}{k} < i < \frac{n(l-1)}{k} + \frac{n}{k\exp(1)}$ then
            if $\tilde{\mathcal{I}}(R_t \cup q_i) > r$ then
                $r \leftarrow \tilde{\mathcal{I}}(R_t \cup q_i)$
            end if
        else if $\tilde{\mathcal{I}}(R_t \cup q_i) > r$ or $i == \frac{nl}{k}$ then
            $R_t \leftarrow R_t \cup q_i$
            $r \leftarrow 0$
            break
        end if
    end for
end for
return $R_t$
```

3.4.4 Training Local ST Gaussian Process Components

Note that we need to specify a kernel function for each GP model. For this, we choose to use a squared exponential kernel, which we augment with automatic relevance determination. Let $x_a = \begin{bmatrix} q_a & t_a \end{bmatrix}$ and $x_b = \begin{bmatrix} q_b & t_b \end{bmatrix}$. Furthermore, let $d = dim(x_a) = dim(x_b)$ and let $x(i)$ denote the i-th element in vector x. Then, we can represent the squared exponential kernel with automatic relevance determination as shown in Equation (3.18).

$$k(x_a, x_b|\theta = \begin{bmatrix} \sigma & \sigma_f & \sigma_n \end{bmatrix}) = \sigma_f^2 \exp\left[-\frac{1}{2} \sum_{m=1}^{d} \frac{(x_a(m) - x_b(m))^2}{\sigma(m)^2} \right] \tag{3.18}$$

We can optimize for the parameters in θ in a Bayesian framework by maximizing the natural logarithm of the marginal likelihood. The optimization problem can be posed as $\theta^* = \text{argmax}_\theta \log \mathbb{P}(\mathbf{Y}|\mathbf{X}, \theta)$ where $\log \mathbb{P}(\mathbf{Y}|\mathbf{X}, \theta)$ is given in Equation (3.19).

$$\log \mathbb{P}(\mathbf{Y}|\mathbf{X}, \theta) = -\frac{N}{2} \log(2\pi) - \frac{1}{2} \log |K_{\mathbf{X},\mathbf{X}}| - \frac{1}{2} \mathbf{Y}^\top K_{\mathbf{X},\mathbf{X}}^{-1} \mathbf{Y} \tag{3.19}$$

3.5 Recurrent Neural Process

We choose to model the spatiotemporal parametric model as a Recurrent Neural Process (RNP) [13]. This model extends the Neural Process model, which is designed for static spatial processes. The Neural Process model summarizes all the contextual information in a fixed size embedding and learns an approximate latent distribution $\tilde{\mathbb{P}}(z|x_C, y_C)$ where (x_C, y_C) is the context set, which is composed of the sample locations x_C and their corresponding environment density values y_C. The Recurrent Neural Process extends this idea by considering a spatiotemporal process as a related sequence of static spatial processeses. Thus, instead of first computing a single encoding s of (x_C, y_C) directly as a Neural Process would have, the Recurrent Neural Process computes an encoding s_t of (x_C, y_C) for each timestep and then uses $S = [s_t]_{\forall t}$ to forecast $s_{t'}$. The Recurrent Neural Process then uses the forecasted state representation $s_{t'}$ to parameterize the approximate latent distribution $\tilde{\mathbb{P}}(z_{t'}|x_C, y_C)$. Finally, the RNP draws from the distribution $z'_{t'} \sim \tilde{\mathbb{P}}(z_{t+1}|x_C, y_C)$ to produce estimates for $y_{t'}$ given $x_{t'}$ using a decoder module. We summarize this in Algorithm 3.

3.6 Empirical Evaluation

We evaluated our approach on a spatiotemporal dataset to demonstrate the efficacy of our methodology. We describe the empirical evaluation below.

3.6.1 Experimental Setup

We test our methodology on a spatiotemporal air temperature dataset [10]. This dataset contains monthly air temperature readings from 1948 to near present in 2.5 degree latitude × 2.5 degree longitude discretization. We test on a 45 × 21 size grid with the same discretization, yielding an area of 112.5 degree latitude by 50 degree longitude area. Figure 3.1 shows a visualization of the dataset for a

Algorithm 3 Recurrent Neural Process [13]

```
Require: ${\{(x_C, y_C)\}_{t=p}, \dots \{(x_C, y_C)\}_{t=p+k}}$
Require: $x_{t'}$
$S \leftarrow \emptyset$
for each $\{(x_C, y_C)\}_t \in \{\{(x_C, y_C)\}_{t=p}, \dots \{(x_C, y_C)\}_{t=p+k}\}$ do
    $s_t \leftarrow \texttt{Encoder}(\{(x_C, y_C)\}_t)$
    $S \leftarrow S \cup s_t$
end for
$s_{t'} \leftarrow \texttt{Forecaster}(S)$
$\mu_{t'}, \sigma_{t'} \leftarrow \texttt{LatentDistributionEncoder}(s_{t'})$
$y_{t'} \leftarrow \emptyset$
for $i = 1, \dots, N$ do
    $z'_{t'} \sim \tilde{\mathbb{P}}(z_{t+1} | x_C, y_C, \mu_{t'}, \sigma_{t'})$
    $y_{t'} \leftarrow y_{t'} \cup \texttt{Decoder}(x_{t'}, z'_{t'})$
end for
$\bar{y}_{t'} \leftarrow \frac{1}{N} \sum_i y_{t'}(i)$
return $\bar{y}_{t'}$
```

couple of consecutive timesteps to demonstrate qualitatively the spatial and temporal correlation present within the data.

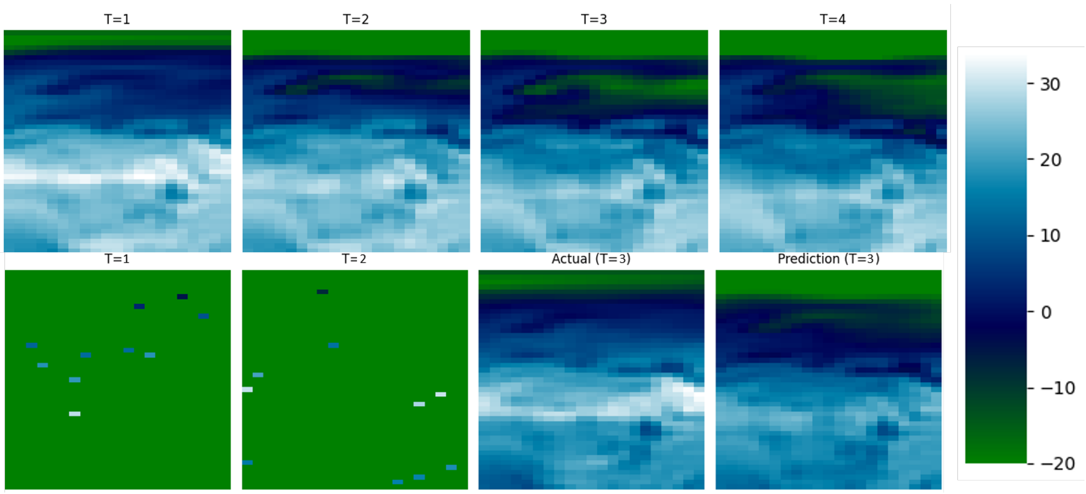

Figure 3.1: (Top) Visualization of the ground truth air temperature data across 4 consecutive timesteps. (Bottom) Visualization of an example RNP inference for a time window of two timesteps. The first two frames with 10 samples selected are provided as input to the RNP, third frame is the ground truth, and the fourth frame is the RNP prediction

We test three different sampling strategies for recommending samples to the RNP for training: (1) Pure random sampling, which is the common sampling strategy used in most prior work that use deep learning for spatiotemporal modelling; (2) Choosing the samples with the highest uncertainty or variance based on the mixture of Gaussian Processes model, which is the common sampling strategy used in active learning for static spatiotemporal modelling; and (3) Choosing samples based on the largest mutual information gain, which is our proposed approach. Our primary metric for analysis is the root mean square error (RMSE), since this is the most common metric used in adaptive sampling and reconstruction literature. We set $k = 6$ and $m = 10$ (section 3.2, Algorithm 1).

3.6.2 Results

We first evaluate the performance of the spatiotemporal mixture of GPs model on forecasting the next timestep after collecting samples and updating the STMGP model in the current timestep. Figure 3.2 shows that with less than 1% of the spatiotemporal dataset provided as initial data per robot, the STMGP model with mutual information as the sampling heuristic (our approach) is able to converge within 5 timesteps to a model that can accurately forecast the next timestep with an RMSE between 4 and 6. Note that the RMSE fluctuates due to the environment changing at each timestep and the STMGP forecasting the next timestep given only the data it has seen until that timestep. The maximum and minimum air temperature in this dataset are -45.76 and 42.15, so this RMSE is quite low (normalized RMSE = 0.0683). This demonstrates that the STMGP is accurately predicting the air temperature distribution for the next timestep, so the STMGP model is clearly choosing appropriate samples to ensure that it maintains a representative model. This validates the use of mutual information as the sampling heuristic for updating the STMGP model.

However, there is clearly a limit to the achievable RMSE using a nonparametric model such as the STMGP in a pure active learning scheme. From Figure 3.3 (right), we can see that the recurrent neural process (RNP) is able to achieve a better RMSE regardless of the sampling strategy used. This demonstrates the gap between the use of nonparametric models and parametric models, especially powerful function

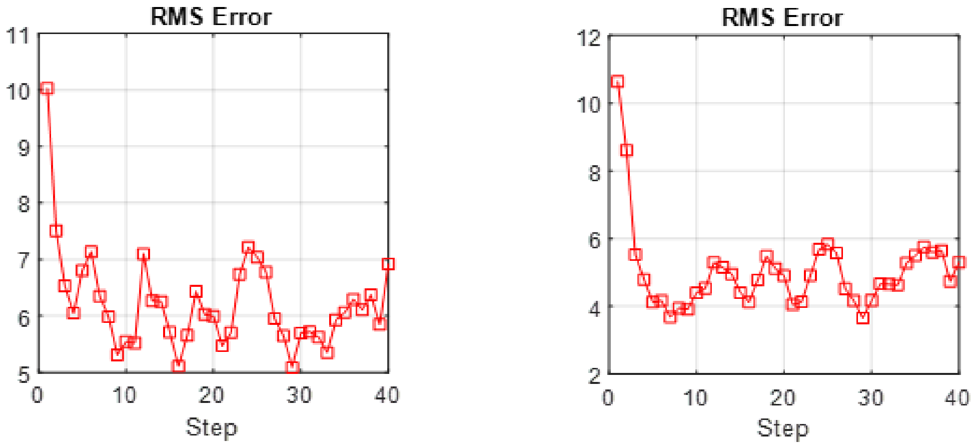

Figure 3.2: RMSE of forecasting the next timestep at each timestep in the environment using only the STMGPs using Maximum Variance Sampling (Left) versus using Maximum Mutual Information Sampling (Right). The use of mutual information as the sampling strategy obtains a better (lower) RMSE.

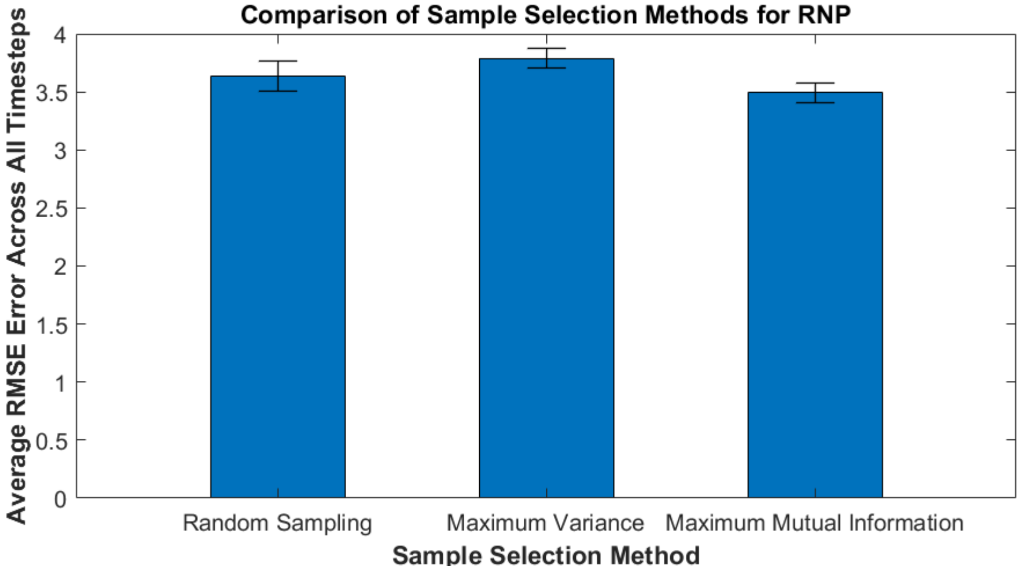

Figure 3.3: Average RMSE of forecasting on an evaluation dataset of unseen timesteps in the environment using the RNP with various sampling strategies

approximators that are common in deep learning. This further motivates finding an end-to-end methodology for integrating multi-robot information gathering and adaptive sampling with powerful parametric or deep learning methodologies for spatiotemporal forecasting and model learning.

We compare the performance of the recurrent neural process on three different sampling strategies: (1) Random Sampling, (2) Maximum Variance, and (3) Maximum Mutual Information **(our approach)**. We first train an RNP using the samples procured by each of the sampling strategies until convergence (50 epochs), and then evaluate the RNP on unseen time windows in the environment for forecasting. Table 3.1 shows the results of the achieved RMSE of the RNP on 11 different seeds. The random seeds randomize the initial configurations of the robots, the initial data given to the mixture of GPs (only applicable for (2) and (3)), the random noise added to the observation, and the train-evaluation dataset generation. The results demonstrate that using maximum mutual information produces the lowest RMSE error along with the lowest variance in the RMSE error in comparison to the other two sampling strategies. Using maximum variance produces consistent, low variance RMSE results in comparison to random sampling, which has high variance in the average RMSE. This is likely due to the larger volatility present in sampling randomly. However, using maximum uncertainty or variance as the sampling strategy produces a higher average RMSE than simple random sampling. However, using the combination of using mutual information for updating the MGP model as well as using mutual information for selecting candidate samples for the RNP yields both low average and variance in the RMSE, outperforming the other two strategies. We validate this difference is statistically significant with a 2-sample t-test to determine if the results are statistically significant. We find that using the maximum mutual information gain sampling strategy reports a p-value of 0.00394 when compared to using the random sampling strategy and reports a p-value of 3.9363e-08 when compared to using maximum variance sampling strategy, which indicates that our approach is statistically significantly better than alternative approaches based on prior work under a significance value of 0.01.

Finally, we analyze the convergence properties of the RNP under the various sampling strategies. In particular, we investigate how many epochs it takes on average for the RNP to outperform the RMSE achieved by Mixture of GPs for forecasting. Table 3.1 demonstrates that the RNP is able to achieve an RMSE less than 4 in the least number of epochs when using maximum mutual information, followed by random sampling and maximum variance. Thus, not only does our methodology provide an end-to-end approach that achieves more accurate spatiotemporal forecasting, but

RMSE and Convergence Comparison for RNP under Each Sampling Method		
Sampling Method	Mean RMSE ± SD	Mean Epoch ± SD
Random Sampling	3.635091 ± 0.130838	31.363636 ± 5.31550
Max Variance (STMGP)	3.788727 ± 0.082061	36.454545 ± 6.743346
Max Mutual Info (STMGP)	3.492363 ± 0.087282	27.636363 ± 3.170890

Table 3.1: (Left Col.) Sampling method used to procure data for RNP. (Center Col.) Average RMSE across 11 different seeds for training the RNP. The use of the MGPs with mutual information provided statistically significantly better RMSE error for forecasting. (Right Col.) Average number of epochs needed to achieve RMSE < 4.

does so in an efficient manner such that model training is effective and converges faster compared to other strategies and approaches. We once again validate that this difference is statistically significant with a 2-sample t-test. We find that using the maximum mutual information gain sampling strategy reports a p-value of 0.03154 when compared to using the random sampling strategy and reports a p-value of 0.00076 when compared to using maximum variance sampling strategy, which indicates that our approach is statistically significantly better than alternative sampling approaches based on prior work under a significance value of 0.05.

3.7 Conclusion

In this work, we present an end-to-end methodology that starts with effective and coordinated multi-robot information gathering to adaptively sample and procure a collection of samples as a dataset used to train a parametric spatiotemporal model. To the best of our knowledge, this is the first work to address integrating multi-robot information gathering with deep learning for spatiotemporal model learning, as prior work often chooses to focus on one or the other. By using a parametric model as the basis of modelling the spatiotemporal phenomena, we achieve more accurate results in forecasting as compared to nonparametric methodologies. Moreover, we show that we can enable these powerful parametric or deep learning methods for spatiotemporal modelling and forecasting to be used in robotic environment monitoring contexts by integrating multi-robot adaptive sampling with supervised spatiotemporal environment learning. We will now investigate incorporating constrained informative path planning to the STMGP methodology in chapter 4.

Chapter 4

Integrating Multi-Robot Adaptive Sampling and Informative Path Planning for Spatiotemporal Environment Prediction

4.1 Introduction

A notable challenge in multi-robot systems is the multi-robot information gathering problem, which encompasses a variety of formulations including multi-robot adaptive sampling [40], multi-robot sensor coverage [16, 19], and multi-robot informative path planning [5]. In this work, we are particularly interested in addressing the multi-robot adaptive sampling and informative path planning problem, in which a group of robots are deployed in an environment from random initial configurations and then seek to gather the best samples in the environment for predicting a spatiotemporal environment phenomena, such as temperature distributions (e.g. air or ocean), salinity, wind speeds, etc. There are many real-world motivations for considering this multi-robot adaptive sampling problem formulation. One such application is the use of multiple UAVs or UGVs to monitor climate phenomena for better weather forecasts or for monitoring agricultural and ecosystem well-being. However, this multi-robot adaptive sampling and informative path planning framework is also applicable to wildfire monitoring, agricultural yield forecasting, or wildlife tracking.

This problem formulation possesses a few differences compared to the prior literature. Firstly, we address the problem of multi-robot adaptive sampling, which involves coordinating a team of robots effectively to sample an environmental phenomena. Prior studies in adaptive sampling has produced many works for single robot scenarios [1, 6, 20, 21, 35]. Secondly, we consider an environment with an arbitrary spatiotemporal process. Spatiotemporal processes depend on both the spatial features and temporal evolution of the underlying phenomena, but a large amount of prior work in multi-robot adaptive sampling only addresses spatially correlated time-invariant processes [11, 16, 19, 27]. Since the environment changes at every timestep, choosing sub-informative samples will result in a lagging model, which is unlikely to recover and will produce erroneous predictions and estimates. Thirdly, we consider constraints regarding actuating mobile robots within a timestep by incorporating the constraints into both sample selection (via the spatiotemporal mixture of GPs) and informative path planning for choosing a path to collect samples along while en route to the desired sampling location. Thus, we also address the multi-robot informative path planning problem. While prior work leverages approaches such as recursive-greedy [35] or differential entropy [14, 15], we do so by taking a two-stage approach of first selecting a destination or goal sampling location of high mutual informativeness, followed by planning a path between the current location and the desired sampling location that maximizes cumulative mutual informativeness when collecting samples en route.

Our **main contributions** in this work are: (1) a two-stage strategy for integrating informative path planning into a spatiotemporal adaptive sampling framework; (2) an augmentation to our spatiotemporal adaptive sampling methodology that incorporates robotic actuation constraints to address multi-robot informative path planning while still producing effective and coordinated multi-robot information gathering to adaptively sample a spatiotemporal environment; and (3) empirical evaluation of multiple variants of our proposed framework. We find that our approach is able to select highly informative samples and paths that improve the predictive performance of the learned spatiotemporal model.

4.2 Problem Formulation

Consider a set of n robots moving in a bounded environment $Q \subset \mathbb{R}^2$ and assume the environment can be discretized into a set of points $q \in Q$. Moreover, let $T \subset \mathbb{R}$ be the time range of interest and assume that this has been discretized into a set of timesteps $t \in T$. With this, the position of each robot $i \in \{1, \ldots, n\}$ at timestep t can be denoted by $x_i^t \in Q$. We assume the environment is free of obstacles and can be partitioned into n Voronoi cells at any timestep t where each Voronoi cell V_i^t is defined in (4.1).

$$V_i^t = \{q \in Q : ||q - x_i^t||_2 \leq ||q - x_j^t||_2 \forall j \neq i\} \tag{4.1}$$

This is a common assumption made in robotic information gathering that is reasonable for most situations in robot exploration [11, 16, 19]. Each Voronoi cell V_i^t corresponds to robot i, meaning robot i is allocated that space at timestep t. This will be leveraged in the proposed active learning methodology (section 4.3.1) to avoid collisions and improve initial environmental modelling. Each location along the path of the robot x_i is sampled and added to its local dataset, which is then used to adaptively choose the next informative sampling location and corresponding paths to traverse.

Regarding the distribution of environmental phenomenon at each point of interest q at time t, there exists an unknown density function $\phi(q, t) : Q \times T \rightarrow \mathbb{R}$ that maps the location q and time t to the scalar value of the phenomena $\phi(q, t)$. That is, the unknown environmental phenomena is both spatial and time varying since it maps each spatial location q and timestep t to a real-valued density value, such as air temperature, animal presence, etc.

Let X_t be the set of locations that have been sampled (but not necessarily shared with one another) and their associated sample values. Moreover, let $F_{X_t}(q) : Q \times T \rightarrow \mathbb{R}$ be a model generated from data X_t that estimates $\phi(q, t)$. We propose to address the optimization problem shown in (4.2).

$$\underset{X_t}{\operatorname{argmin}} \sum_{q \in Q} \sum_{t \in T} (F_{X_t}(q) - \phi(q, t))^2 \tag{4.2}$$

Note that the robots can only collect a sparse or limited number of samples in a single-shot manner, which is akin to actual robotic information gathering. In other words, the robots must proceed sequentially forward in time, and cannot revisit a previous timestep once the timestep has elapsed. Thus, it is imperative that the robots coordinate with one another in order to choose the best sampling locations and paths given a miniscule amount of initial data.

The main question to address is *how to coordinate a team of robots to choose informative paths to collect a set of samples at each timestep into X that will likely improve forecasting performance of a working model of the environment*. In prior work , we proposed a spatiotemporal mixture of Gaussian Processes (STMGP) model, which we showed was able to determine informative samples to be used in updating a parametric (i.e. deep learning-based) model, which led to good forecasting performance [9]. However, this work did not consider constraints regarding robot actuation within a dynamic environment (i.e. path length constraints due to temporal dynamics). In other words, our prior work focused solely on the adaptive sampling problem formulation.

Thus, in this work, we now augment our adaptive sampling framework for spatiotemporal environments to address informative path planning as well and attempt to achieve this in a purely active learning fashion. We approach this in two stages. At each timestep, the robots use the STMGP model to inform the sampling locations for the next timestep such that they can be reached given robot actuation constraints within each timestep. Then, we determine the most informative path to take from the current location of the robot to the desired sampling location informed by the STMGP model. The robots use Voronoi tesselation to partition the environment amongst themselves and exchange relevant information via peer-to-peer communication for adjusting their local datasets, partitions, and for providing robust estimations. This leads to a decentralized approach that enables coordinated sampling and path planning.

4.3 Proposed Approach

Given 5% of the data as initial data to each robot (Line 1) and random starting configurations (Line 2 - 3), the robots first fit their local GP components (Line 7).

The robots use the spatiotemporal mixture of GPs to infer the mean and variance of each location in the environment at the current timestep (Line 10). The mean is used to evaluate the predictive accuracy of the model. The mean and variance are both used in conjunction with the maximum allowable path length in a given timestep to determine where to sample in the next timestep (Line 11). The robots then use communication to fit a uni-model GP to compute an approximate covariance matrix across all locations at the current timestep (Line 13). An approximate informative path planning subroutine, which uses the aforementioned covariance matrix, is then used to determine which path for each robot to take given the desired sampling location (Line 14). The robots then traverse along the planned paths and collect the samples while en route, and then eventually collect the sample at the desired location as well (Line 16 - 17). The new sampling locations (which is now the current locations of the robot after following the planned trajectories) are used to adjust the Voronoi partitions and the local datasets of each robot (Lines 19 - 20) via communication. This is repeated for the allotted number of timesteps in the environment, where the environment changes after each timestep in accordance to ϕ. Algorithm 4 also details where each subsection corresponds to in the system.

4.3.1 Spatiotemporal Adaptive Sampling Methodology

Spatiotemporal Mixture of GPs

We first briefly describe the spatiotemporal mixture of GPs (STMGP) model from our prior work [9]. Let $\mathbf{X}$ be the locally collected data consisting of the sampling location and timestep for a given robot and $\mathbf{Y}$ be the corresponding environment values for the (location, time) in $\mathbf{X}$. We can represent the mean vector $\mu_{\mathbf{X}} = [\mu(q,t)]_{(q,t)\in\mathbf{X}}$ and positive definite symmetric covariance matrix $K_{\mathbf{X},\mathbf{X}} = [\mathbf{k}((q,t),(q',t'))]_{(q,t),(q',t')\in\mathbf{X}}$ where $\mathbf{k}(\cdot)$ is the kernel function. The posterior mean vector is given as $\mu_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}} = K_{\mathbf{X}^*,\mathbf{X}}K^{-1}_{\mathbf{X},\mathbf{X}}\mathbf{Y}$ and the posterior covariance matrix is given as $K_{\mathbf{X}^*|\mathbf{X},\mathbf{Y}} = K_{\mathbf{X}^*,\mathbf{X}^*} - K_{\mathbf{X}^*,\mathbf{X}}K^{-1}_{\mathbf{X},\mathbf{X}}K^{\top}_{\mathbf{X}^*,\mathbf{X}}$ where $\mathbf{X}^*$ is all possible locations $q \in Q$ for the next timestep. This serves as the local GPs in the methodology described below.

The mixture of Gaussian Processes was first introduced by [37], and has been used in sensor coverage of static environments by [16, 19] due to its improved

Algorithm 4 STMGP + IPP

```
Require: approxMI {boolean flag determining how to find most informative path}
Require: useRatio {boolean flag determining sampling goal location selection}
Require: $B$ {maximum allowed path length per timestep}
$\mathbf{X}_i, \mathbf{Y}_i \leftarrow$ GetRandomInitialData($5\%$)
$q_t \leftarrow$ GetRandomStartingLocations()
$V_t \leftarrow$ UpdateVoronoiPartitions($q_t$)
$\hat{\gamma}_t \leftarrow \mathbf{0}$ {$\hat{\gamma}_t(i)$ is the MI lower bound for robot $i$}
for $t \in T$ do
    for each robot $i$ do
        $GP_i \leftarrow$ FitGP($\mathbf{X}_i, \mathbf{Y}_i$)
    end for
    $MGP_t \leftarrow$ MixGPs($GP_i$)  {section 4.3.1}
    $\mu^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i}, \sigma^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i} \leftarrow MGP_t.\mu, MGP_t.\sigma$  {section 4.3.1}
    $q_{t+1} \leftarrow$ SampLocSelect($\sigma^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i}, \hat{\gamma}_t, V_t, q_t, B,$ useRatio)  {section 4.3.2}
    $\hat{\gamma}_t \leftarrow \hat{\gamma}_t + \sigma^*_{q_t|\mathbf{X}_i,\mathbf{Y}_i}$ {section 4.3.2}
    $\tilde{K} \leftarrow$ FitGP($\bigcup_i \mathbf{X}_i, \bigcup_i \mathbf{Y}_i$).$K$  {via P2P comm}
    $\Psi_t \leftarrow$ IPPSubroutine($\tilde{K}, q_t, q_{t+1},$ approxMI)  {section 4.3.2}
    for each robot $i$ do
        $x_i, y_i \leftarrow$ MoveAndCollectSamples($\Psi_t(i)$)
        $\mathbf{X}_i, Y_i \leftarrow \mathbf{X}_i \cup x_i, \mathbf{Y}_i \cup y_i$
    end for
    $V_t \leftarrow$ UpdateVoronoiPartitions($q_{t+1}$)
    $\mathbf{X}_i, \mathbf{Y}_i \leftarrow$ CommDataUsingVoronoi($\mathbf{X}_i, \mathbf{Y}_i, V_t$)
    EvaluateAndCalculateRMSE($\mu^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i}$)
end for
```

representational capacity. In our prior work, we modified the mixture of Gaussian Processes approach for spatiotemporal environments by utilizing the aforementioned spatiotemporal GP components, which were used for procuring samples for training a neural process model for spatiotemporal forecasting [9]. We briefly summarize the mixing process and the sampling strategy.

Our distributed approach leverages a Spatiotemporal Mixture of Gaussian Processes (STMGP) for adaptive sampling and prediction in a spatiotemporal environment Q. Here, each robot i has a learned local Gaussian Process component GP_i, which yields a set of n GP components $\{GP_1, \dots GP_n\}$ and an associated probability function $P(q, i_g) = P(q \in Q \text{ is best described by } GP_{i_g})$. Suppose the robot i has sampled locations $\mathbf{X}_i$ and suppose $\mathbf{Y}_i$ are the associated ground-truth values sampled from locations $\mathbf{X}_i$. Moreover, suppose robot i has a mean $\mu_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ and covariance matrix $K_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ as well. Now, let $x^* \in \mathbf{X^*}_i$. Furthermore, let $\mu_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} \in \mu_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i}$ and $\sigma_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} \in \operatorname{diag}(K_{\mathbf{X^*}_i|\mathbf{X}_i,\mathbf{Y}_i})$. We can represent the local conditional posterior mixture mean $\mu^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i}$ and local conditional posterior mixture variance $\sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i}$ as shown in Equation (4.3) and Equation (4.4) respectively where $\mathit{diff}(i) = (\mu_{x_i^*|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} - \mu^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i})^2$.

$$\mu^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} = \sum_{i=1}^{m} P(q, i_g) * \mu_{x_i^*|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} \tag{4.3}$$

$$\sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} = \sum_{i=1}^{m} P(q, i_g) * (\sigma_{x_i^*|\mathbf{X}_{i_g},\mathbf{Y}_{i_g}} + \mathit{diff}(i)) \tag{4.4}$$

Here, $\mathbf{X}_{i_g} \subset \mathbf{X}_i$ and $\mathbf{Y}_{i_g} \subset \mathbf{Y}_i$, and each represents the set of samples and ground-truth values that can be best described by GP_{i_g} (i.e. $\mathbf{X}_{i_g} = \{q \in \mathbf{X}_i : \arg\max_{i'_g} P(q, i'_g) = i_g\}$. The mixing probabilities are determined via a distributed EM procedure, and we modify $\mathbf{X_i}$ and $\mathbf{Y_i}$ based on the Voronoi tesselation induced by the current location of each robot. The above procedure is carried out in a decentralized manner using the method in [9, 19].

Training Local ST Gaussian Processes

We now specify the kernel function for each local GP model. We use a squared exponential kernel, which we augment with automatic relevence determination structure to compute a length hyperparameter for each input dimension. Let $x_a = \begin{bmatrix} q_a & t_a \end{bmatrix}$ and $x_b = \begin{bmatrix} q_b & t_b \end{bmatrix}$. Let $d = \dim(x_a) = \dim(x_b)$ and let $x(i)$ denote the i-th element in vector x. Then, we can represent the squared exponential kernel with automatic relevence determination as shown in Equation (4.5).

$$\mathbf{k}(x_a, x_b | \theta = \begin{bmatrix} \sigma & \sigma_f & \sigma_n \end{bmatrix}) = \sigma_f^2 \exp\left[-\frac{1}{2} \sum_{m=1}^{d} \frac{(x_a(m) - x_b(m))^2}{\sigma(m)^2} \right] \tag{4.5}$$

We can optimize for the parameters in θ in a Bayesian framework by maximizing the natural logarithm of the marginal likelihood [30]. The optimization problem to compute the optimal θ^* can be posed as $\theta^* = \operatorname{argmax}_\theta \log \mathbb{P}(\mathbf{Y}|\mathbf{X}, \theta)$, where $\log \mathbb{P}(\mathbf{Y}|\mathbf{X}, \theta) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|K_{\mathbf{X},\mathbf{X}}| - \frac{1}{2}\mathbf{Y}^\top K_{\mathbf{X},\mathbf{X}}^{-1}\mathbf{Y}$.

Since we seek to address reducing the mean squared error as shown in Equation (4.2), we choose to leverage mutual information as our heuristic for goal sampling location selection (adaptive sampling) and path selection (informative path planning). We describe this in section 4.3.2.

4.3.2 Informative Path Planning

We now address the problem of informative path planning, as it is necessary to determine what path the robots take for collecting samples when transitioning between one timestep to the next. Since the time elapsed contained in a given timestep is limited, the robots can only move a certain distance in the environment at each timestep (i.e. the robots cannot move to any arbitrary location at each timestep). Thus, it is crucial that each robot plans the most informative path for collecting samples. We choose to leverage mutual information gain as our information criterion, which we demonstrate in prior work is a better criterion in comparison to alternative ones such as entropy [14].

We approach this problem in a two-stage fashion. First, given the maximum path length that can be traversed within the timestep, we determine the best sampling location that is reachable using the STMGP model and a mutual information-based criterion. We present two approaches to utilize the STMGP model for determining the optimal destination sampling location (i.e. goal location).

Once this sampling location (i.e. goal location) has been determined, we explore the set of possible paths to reach that location and determine which location provides the most informativeness via mutual information approximated using submodular function approximation. We present two approaches for this as well.

Formulation for Informative Path Planning

Let ψ_i be the trajectory or path of robot i and let B be the budget or maximum allowable path length in a timestep. Furthermore, let $I(\psi_i)$ and $C(\psi_i)$ be the information gain (i.e. mutual information gain) and path cost (i.e. path length) associated with ψ_i. We address the following optimization problem shown in Equation (4.6) for each robot, which is known as the multi-robot informative path planning problem.

$$\psi_i^* = \underset{\psi_i}{\operatorname{argmax}}\, I(\psi_i) \text{ s.t. } C(\psi_i) \leq B \tag{4.6}$$

Some prior work have also proposed a variant of the informative path planning problem as shown in Equation (4.7).

$$\psi_i^* = \underset{\psi_i}{\operatorname{argmax}}\, \frac{I(\psi_i)}{C(\psi_i)} \text{ s.t. } C(\psi_i) \leq B \tag{4.7}$$

In section 4.3.2, we outline a mutual information-based heuristic to select which sampling location will provide the most informative sample with respect to the current model, and we set that as a sampling goal location to reach. We propose two approaches for doing this based on the two styles of multi-robot informative path planning optimization schemes shown in Equation (4.6) and Equation (4.7). In section 4.3.2, we then outline choosing which path to take to reach that sampling goal location such that the cumulative information gained from collecting samples along the path is maximized. In this section, we also investigate two approaches, one being (on average) a faster but more approximate version.

Mutual Information for Selecting Goal Sampling Location

Calculating mutual information is known to be difficult but in order to have the most accurate representation of the environment, each robot should choose the most informative sample. Thus, for optimizing each local Gaussian Process component, we choose a sample that optimizes for a lower bound on the mutual information between the current local Gaussian Process GP_i and the noisy observations to be collected $\mathbf{Y}_i$ from the sampling locations $\mathbf{X}_i$ at timestep t. Let $\gamma_t = \max_{\mathbf{X}_i} I_t(\mathbf{X}_i, \phi | GP_i)$ where $I(\cdot)$ is the mutual information. For Gaussian Processes, the lower bound $\hat{\gamma}_t = \sum_{t'=1}^{t} \sigma_{t'}^2(x_{t'}) \leq \frac{2}{\log(1+\sigma_n^{-2})}\gamma_t$ holds [3]. The lower bounded quantity $\hat{\gamma}_t$ can be computed as shown in Equation (4.8), and optimizing for the lower bounded quantity $\hat{\gamma}_t$ yields the optimization strategy shown in Equation (4.9) for each robot [3].

$$\hat{\gamma}_t = \hat{\gamma}_{t-1} + \sigma^*_{x^*|\mathbf{X}_i,\mathbf{Y}_i} \tag{4.8}$$

$$x^* = \underset{x_i^* \in V_t(i)}{\operatorname{argmax}} \sqrt{\sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}} - \sqrt{\hat{\gamma}_{t-1}} \text{ s.t. } \operatorname{dist}(x_i^*, q_t) \leq B \tag{4.9}$$

From this we can simply find the sampling location that satisfies Equation (4.9) for each robot, which would address Equation (4.6). Alternatively, we could consider a strategy based on solving Equation (4.7) as shown in Equation (4.10).

$$x^* = \underset{x_i^* \in V_t(i)}{\operatorname{argmax}} \frac{\sqrt{\sigma^*_{x_i^*|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}} - \sqrt{\hat{\gamma}_{t-1}}}{C(x_i^*, q_t)} \text{ s.t. } \operatorname{dist}(x_i^*, q_t) \leq B \tag{4.10}$$

Generally, $C(x_i^*, q_t) = \operatorname{dist}(x_i^*, q_t)$, which often leads to myopic waypoints. Thus, most methods that address Equation (4.7) often generate waypoints that are close to the current location of the robot. This tends to work well in greedy structures [35] or RL-based schemes [28, 29, 31]. However, since we are leveraging a decoupled strategy, we often do not want a myopic waypoint to be chosen as our sampling goal location at each timestep. Thus, we propose $C(x_i^*, q_t) = \frac{(B - \operatorname{dist}(x_i^*, q_t))^2}{B} + 1$. This is a shifted and scaled quadratic term centered around the budget the produces a value of 1 if the path chosen exhausts the budget B and produces a larger value the shorter the path length. We find that this empirically works well in our empirical evaluation in section 4.4. We show both approaches in Algorithm 5.

Mutual Information for Path Planning

When each robot selects the most informative sampling location (that is reachable) to navigate to, it induces a closed set of possible paths to consider. Thus, once each

Algorithm 5 SamplingGoalLocationSelection

```
Require: $\sigma^*_{q_i^*|\mathbf{X}_i,\mathbf{Y}_i}$ {STMGP uncertainty or variance}
Require: $\hat{\gamma}_{t-1}$ {Mutual information lower bound}
Require: $V_t$ {Voronoi partitions}
Require: $q_t$ {Current locations of robots}
Require: $B$ {Maximum allowable path length or budget}
Require: useRatio {Flag for goal samp select method}
$q_{t+1} \leftarrow \emptyset$
if useRatio == True then {use Ratio method}
    for each robot $i$ do
        $\Theta(i) \leftarrow \sqrt{\sigma^*_{q_i^* \in V_t(i)|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}(i)} - \sqrt{\hat{\gamma}_{t-1}(i)}$
        $\delta(i) \leftarrow \texttt{GetAllShortestDists}(q_t(i), V_t(i))$
        $C(i) \leftarrow \frac{(B-\delta(i))^2}{B} + 1$
        $\Omega(i) \leftarrow \frac{\Theta(i)}{C(i)}$
        $\Omega(i)[\delta(i) > B] = -1$
        $q_{t+1} \leftarrow q_{t+1} \cup \operatorname{argmax}_q \Omega(i)$
    end for
else {simple search for maxima}
    for each robot $i$ do
        $\Theta(i) \leftarrow \sqrt{\sigma^*_{q_i^* \in V_t(i)|\mathbf{X}_i,\mathbf{Y}_i} + \hat{\gamma}_{t-1}(i)} - \sqrt{\hat{\gamma}_{t-1}(i)}$
        $\delta(i) \leftarrow \texttt{GetAllShortestDists}(q_t(i), V_t(i))$
        $\Theta(i)[\delta(i) > B] = -1$
        $q_{t+1} \leftarrow q_{t+1} \cup \operatorname{argmax}_q \Theta(i)$
    end for
end if
return $q_{t+1}$
```

robot chooses which sampling goal location to navigate to, the robot must then choose which path to choose such that the samples collected at each location along the path maximizes the cumulative information gained. We rely on utilizing mutual information to maximize the informativeness of the path chosen.

Once again, computation of mutual information is difficult. Moreover, instead of determining the mutual information with respect to a function (which was the case in section 4.3.2), we have to evaluate the cumulative return of collecting a set of locations which provides the maximal informativeness with respect to one another. Let $\Psi^i_{(q_t,x^*)}$ be the set of samples along the ith candidate path from the current location q_t to the goal location x^* determined by the optimization strategy shown in Equation (4.9). Furthermore, let $\Phi_t = \{\phi(q,t) : \forall q \in Q\}$ be the set of environment values at all the locations in the environment Q at timestep t. We wish to solve the optimization problem posed in Equation (4.11) for each robot.

$$\begin{aligned}\Psi^* &= \underset{\Psi^i_{(q_t,x^*)}}{\operatorname{argmax}}\, \mathcal{I}(\Psi^i_{(q_t,x^*)}, \Phi_t \setminus \Psi^i_{(q_t,x^*)}) \\ &= \underset{\Psi^i_{(q_t,x^*)}}{\operatorname{argmax}}\, \mathcal{H}(\Phi_t \setminus \Psi^i_{(q_t,x^*)}) - \mathcal{H}(\Phi_t \setminus \Psi^i_{(q_t,x^*)} | \Psi^i_{(q_t,x^*)})\end{aligned} \tag{4.11}$$

Since mutual information gain is a monotone and submodular function [12, 18, 41], we leverage submodular function optimization to find a provably good approximate path (i.e. set of sampling locations) to the optimally informative path. We present two approaches to utilizing submodular function optimization to select the provably near-optimal informative path. One approach is a simply to find the path with the maximum information gain after computing the approximate mutual information gain for each path using submodular function optimization, which requires analyzing each candidate path. Alternatively, we can employ a greedy-based approach similar to [18, 41] to find the approximately optimal informative path by treating the set of paths as a stream of batched samples and employing a stream-based secretary algorithm to approximately solve for the optimal Ψ^* at each timestep t based on the methodology in [18]. We show both approaches in Algorithm 6.

Algorithm 6 IPPSubroutine

```
Require: $\tilde{\mathcal{I}}(\cdot|\tilde{K})$ {$\tilde{K}$ is the covariance matrix used to approximate MI via
    submodular function optimization}
Require: $q_t$ {Current locations of robots}
Require: $q_{t+1}$ {Goal locations of robots}
Require: approxMI {Flag to determine IPP method}
$\Psi_t \leftarrow \emptyset$
if approxMI == True then {stream secretary [18]}
  for each robot $i$ do
    $\Psi^i_{(q_t,q_{t+1})} \leftarrow$ ComputeAllPaths$(q_t, q_{t+1})$
    $r \leftarrow 0$
    $n \leftarrow |\Psi^i_{(q_t,q_{t+1})}|$
    for each path $\psi^i \in \Psi^i_{(q_t,q_{t+1})}$ do
      if $0 < i < \frac{n}{\exp(1)}$ and $\tilde{\mathcal{I}}(\psi^i) > r$ then
        $r \leftarrow \tilde{\mathcal{I}}(\psi^i)$
      else if $\tilde{\mathcal{I}}(\psi^i) > r$ or $i == n$ then
        $\Psi_t(i) \leftarrow \psi^i$
        break
      end if
    end for
  end for
else {simple search for maxima}
  for each robot $i$ do
    $\Psi^i_{(q_t,q_{t+1})} \leftarrow$ ComputeAllPaths$(q_t, q_{t+1})$
    $r \leftarrow 0$
    for each path $\psi^i \in \Psi^i_{(q_t,q_{t+1})}$ do
      if $\tilde{\mathcal{I}}(\psi^i) > r$ then
        $r \leftarrow \tilde{\mathcal{I}}(\psi^i)$
        $\Psi_t(i) \leftarrow \psi^i$
      end if
    end for
  end for
end if
return $\Psi_t$
```

4.4 Empirical Evaluation

We evaluated our approach on a spatiotemporal dataset to demonstrate the efficacy of our methodology. We describe the empirical evaluation below.

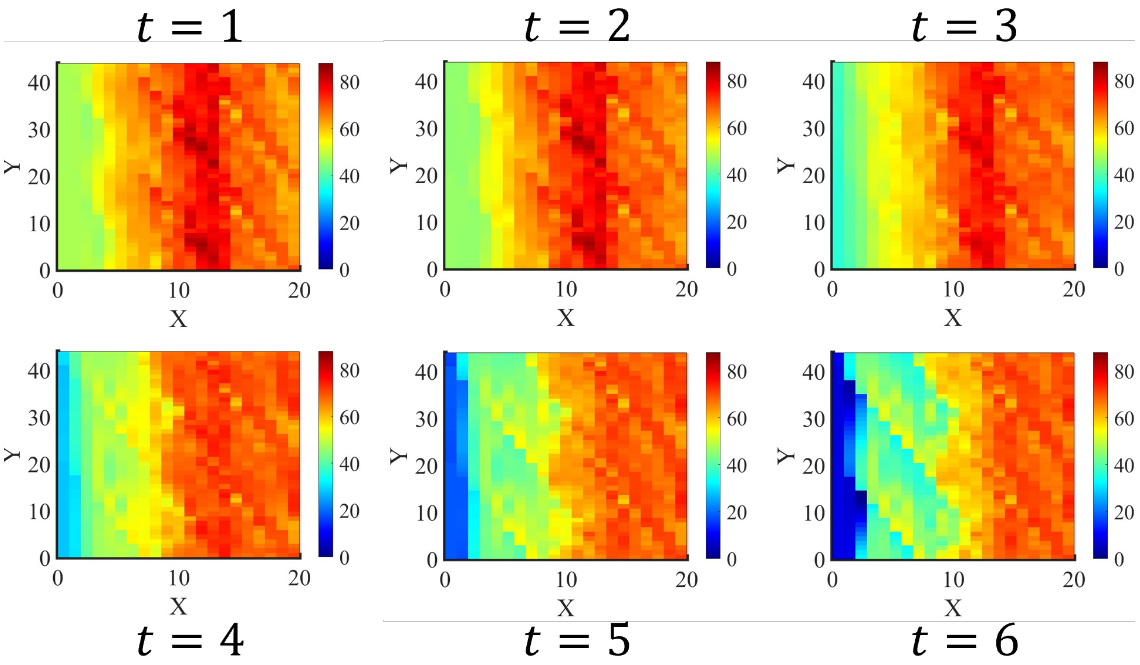

Figure 4.1: Visualization of the ground truth spatiotemporal air temperature data across 6 consecutive timesteps

4.4.1 Experimental Setup

We test our methodology on a spatiotemporal air temperature dataset [10]. This dataset contains monthly air temperature readings from 1948 to near present in 2.5 degree latitude × 2.5 degree longitude discretization. We test on a 45 × 21 size grid with the same discretization, yielding an area of 112.5 degree latitude by 50 degree longitude area. Figure 4.1 shows a visualization of the dataset for a couple of consecutive timesteps to demonstrate qualitatively the spatial and temporal correlation present within the data. We analyse root mean square error (RMSE), since this metric is used in adaptive sampling and reconstruction literature.

4.4.2 Results

We evaluate the performance of the spatiotemporal mixture of GPs (STMGP) model augmented with informative path planning (IPP) on predicting all locations in the timestep after collecting samples and updating the STMGP model. Figure 4.2 and Table 4.1 shows that with less than 5% of the spatiotemporal dataset provided as initial data per robot, the STMGP model is able to stably converge within to a model that can accurately predict the spatiotemporal environment with an average RMSE of around 5. Moreover, with the exception of the first couple of timesteps (due to the need to explore and collect data), the STMGP model achieves an RMSE between 4 to 6. Note that the RMSE fluctuates due to the environment changing at each timestep and the STMGP predicting the spatiotemporal environmental phenomena given only the data it has seen until that timestep. The maximum and minimum air temperature in this dataset are -45.76 and 42.15, so this RMSE is quite low (normalized RMSE = 0.0569).

IPP Method Sampling Goal Selection	Stream Secratary	Maxima Search
Ratio	5.10287, 5.0663	5.03722, 5.12957
Direct	4.98927, 5.13966	4.99739, 5.1613

RMSE using shortest path: 5.33637

Table 4.1: Comparison of RMSE for each proposed approach where each cell corresponds to ($B = 5$, $B = 10$)

This demonstrates that the STMGP is accurately predicting the air temperature distribution of the environment despite having very little knowledge of the underlying spatiotemporal dynamics, so the STMGP model is clearly choosing appropriate sampling goal locations and informative paths to reach those sampling goal locations to ensure that it maintains a representative model.

In addition to this, we test the performance of the STMGP with informative path planning integrated against the STMGP with a simple shortest path planning algorithm and no budget constraint or constraints on sampling goal location. This serves

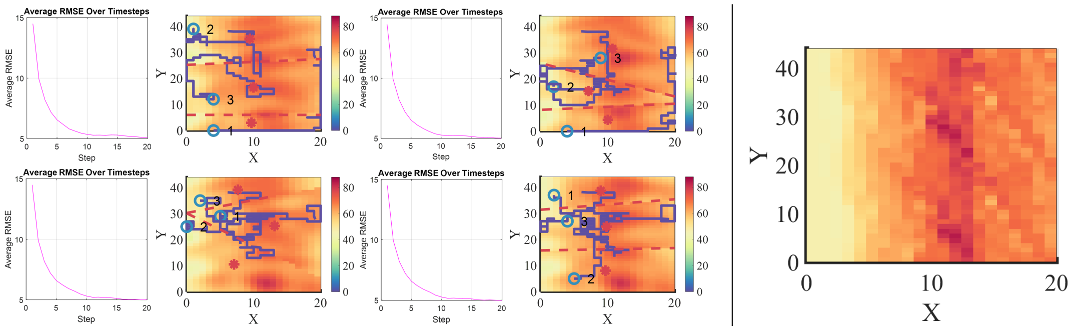

Figure 4.2: (Left) Average RMSE of predicting all locations in the next timestep for each timestep using the STMGPs with $B = 5$ and visualization of robot trajectories with estimation of final timestep. Each quadrant corresponds spatially to the cell in Table 4.1. (Right) Ground truth of final timestep.

as a representative baseline of our prior work to compare our proposed approaches for augmenting our STMGP adaptive sampling framework with informative path planning. From Table 4.1, we see that all the proposed approaches outperform our unconstrained STMGP model with shortest path planning. Thus, we can see that even with constraints, our STMGP adaptive sampling framework augmented with the informative path planning handles the constraints such that it performs better than our baseline prior work.

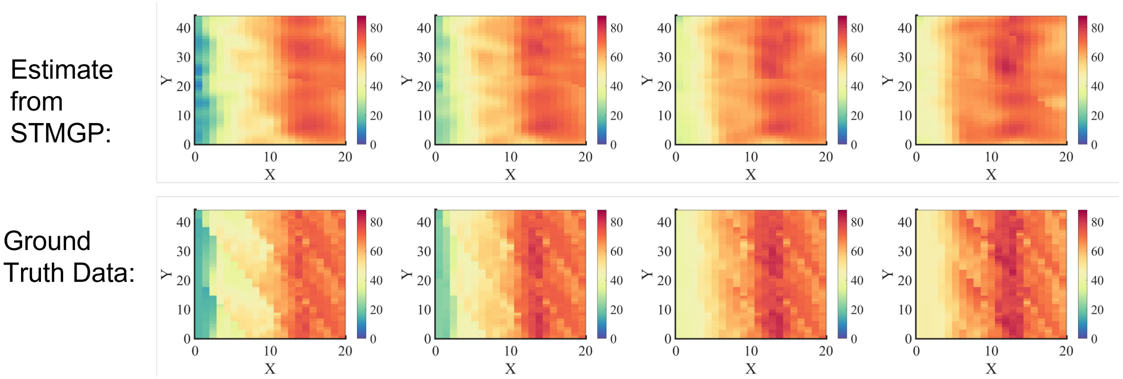

Figure 4.3: Visualization of the ground truth spatiotemporal density function versus the estimate from the STMGP+IPP model across 6 timesteps under $B = 5$. We showcase the method that achieved the best average RMSE based on Table 4.1 (Direct, Stream Secretary).

This demonstrates both the utility in using mutual information as the information criterion for informative path planning as well as the two-staged approach that we employ to achieve effective integration of adaptive sampling and informative path planning for spatiotemporal prediction.

4.5 Conclusion

In this work, we present an end-to-end methodology for effective and coordinated multi-robot information gathering to adaptive determine informative sampling goal locations and then plan informative paths to those locations to collect samples along the way. By leveraging a decoupled process that starts with adaptive sampling for sampling goal location selection followed by informative path planning, we are able to learn an effective active learning model of the spatiotemporal environment for accurate prediction.

Chapter 5

Conclusion

5.1 Summary of Contributions

In this thesis, we investigate the multi-robot adaptive sampling and informative path planning problem for spatiotemporal environments. Within this broad topic, we analyze two particular problem instances:

1. How to deploy a team of robots to adaptively sample a spatiotemporal environment to procure a sparse dataset for learning a parametric forecasting neural model?
2. How to integrate a spatiotemporal adaptive sampling framework with informative path planning under path length constraints in each timestep with minimal effect on predictive error?

We present two methodologies for addressing both of these formulations. Both are centered around the use of the spatiotemporal mixture of GPs (STMGP), which we compose of spatiotemporal unimodel GPs and use a squared exponential kernel with automatic relevence determination for the kernel structure. Moreover, we optimize for the hyperparameters in the automatic relevence determination kernel using a Bayesian optimization scheme. The use of the common squared exponential (RBF) kernel with automatic relevence determination indicates the lack of assumptions made regarding the spatiotemporal environment. Within the two methodologies, we also do not integrate specific information regarding the environment, ensuring that our approaches can be used for any arbitrary spatiotemporal environment.

The first methodology uses the STMGP to determine samples both for updating itself as well as for recommending a set of samples for training a neural model for spatiotemporal forecasting. We demonstrate that the use of the STMGP for recommending samples to the recurrent neural process (RNP) performs better than random sampling, which is what a majority of prior work in deep learning for spatiotemporal forecasting uses. Moreover, we show that the common heuristic used in prior work in adaptive sampling for environment reconstruction, the maximum uncertainty sampling stategy, is not appropriate for spatiotemporal environments, and that mutual information is a more appropriate heuristic for handling spatiotemporal reconstruction. The second methodology uses the STMGP to search for informative waypoints that are reachable under a maximum path length at each timestep. Using this waypoint, we search of the set of paths to find an maximally or approximately maximally informative path.

For both works, we demonstrate accurate predictions and forecasting, with low normalized RMSE. This performance demonstates that our approaches can coordinate a team of robots effectively to address these problem formulations in a distributed and decentralized manner. Moreover, we address significant knowledge gaps present in multi-robot spatiotemporal adaptive sampling and informative path planning. We show how to design a robust non-parametric approach using n GP components (represented by each of the n robots) with specific but necessary structures to achieve a low RMSE model across each timestep in the spatiotemporal environment. We also demonstrate how to integrate this into either learning a model for spatiotemporal forecasting (via strongly supervised learning) or integrating an informative path planning scheme that respects path length constraints between each timestep (accounting for robot actuation constraints).

5.2 Future Work

There are many avenues for future work that are currently being investigated or that we plan to investigate. One such avenue involves improving the STMGP model itself with better kernel structures. While we avoided using domain-specific kernels in this thesis to allow for generalizability to diverse spatiotemporal environments, the squared exponential (RBF) kernel is a commonly used kernel. Newer kernel structures, such as the Attentive Kernel [2], can be potentially used in the STMGP model for better modelling and prediction. Another avenue involves use of Log Gaussian Cox Processes as opposed to Gaussian Processes for modelling, which could be more useful in sparser spatiotemporal environments such as target tracking and prediction (as opposed to spatiotemporal fields such as air temperature or ocean salinity). Another avenue of research is addressing alternative and potentially better optimization strategies for adaptive sampling and informative path planning for the multi-robot scenario. For example, the use of multi-agent reinforcement learning could be investigated for learning a communication-action policy. The use of ergodic trajectories could also be useful potentially if integrated with the STMGP framework. These are all research directions we actively are investigating or plan to look into in the near future.

Bibliography

[1] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *2012 IEEE international conference on robotics and automation*, pages 2147–2154. IEEE, 2012. 2.2, 3.1, 4.1

[2] Weizhe Chen, Roni Khardon, and Lantao Liu. AK: Attentive Kernel for Information Gathering. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.047. 5.2

[3] Emile Contal, Vianney Perchet, and Nicolas Vayatis. Gaussian process optimization with mutual information. In *International Conference on Machine Learning*, pages 253–261. PMLR, 2014. 4.3.2

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004. 3.1

[5] Ayan Dutta, Anirban Ghosh, and O. Patrick Kreidl. Multi-robot informative path planning with continuous connectivity constraints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3245–3251, 2019. 3.1, 4.1

[6] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33 (9):1271–1287, 2014. 2.2, 3.1, 4.1

[7] Geoffrey A. Hollinger, Arvind A. Pereira, Jonathan Binney, Thane Somers, and Gaurav S. Sukhatme. Learning uncertainty in ocean current predictions for safe and reliable navigation of underwater vehicles. *Journal of Field Robotics*, 33(1):47–66, 2016. doi: https://doi.org/10.1002/rob.21613. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21613`. 2.2

[8] Austin Jones, Mac Schwager, and Calin Belta. A receding horizon algorithm for informative path planning with temporal logic constraints. In *2013 IEEE International Conference on Robotics and Automation*, pages 5019–5024. IEEE, 2013. 2.2

[9] Siva Kailas, Wenhao Luo, and Katia Sycara. Multi-robot adaptive sampling

for supervised spatiotemporal forecasting. In *Progress in Artificial Intelligence: 22nd EPIA Conference on Artificial Intelligence, EPIA 2023*. Springer, 2023. 4.2, 4.3.1, 4.3.1

[10] Eugenia Kalnay, Masao Kanamitsu, Robert Kistler, William Collins, Dennis Deaven, Lev Gandin, Mark Iredell, Suranjana Saha, Glenn White, John Woollen, et al. The ncep/ncar 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–472, 1996. 3.6.1, 4.4.1

[11] Stephanie Kemna, John G Rogers, Carlos Nieto-Granda, Stuart Young, and Gaurav S Sukhatme. Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2124–2130. IEEE, 2017. 2.2, 3.1, 3.2, 4.1, 4.2

[12] Andreas Krause. Sfo: A toolbox for submodular function optimization. *Journal of Machine Learning Research*, 11(38):1141–1144, 2010. 4.3.2

[13] Sumit Kumar. Spatiotemporal modeling using recurrent neural processes, 2019. 3.2, 3.5, 3

[14] Sumit Kumar, Wenhao Luo, George Kantor, and Katia Sycara. Active learning with gaussian processes for high throughput phenotyping, 2019. 4.1, 4.3.2

[15] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. Adaptive multi-robot wide-area exploration and mapping. In *Proceedings of the 7th International Joint Conference on Autonomous agents and Multiagent systems-Volume 1*, pages 23–30, 2008. 2.2, 4.1

[16] Wenhao Luo and Katia Sycara. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6359–6364, 2018. 3.1, 3.2, 4.1, 4.2, 4.3.1

[17] Wenhao Luo, Shehzaman S. Khatib, Sasanka Nagavalli, Nilanjan Chakraborty, and Katia Sycara. Distributed knowledge leader selection for multi-robot environmental sampling under bandwidth constraints. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5751–5757, 2016. 3.1

[18] Wenhao Luo, Changjoo Nam, and Katia Sycara. Online decision making for stream-based robotic sampling via submodular optimization. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 118–123, 2017. doi: 10.1109/MFI.2017.8170416. 3.4.3, 2, 4.3.2, 2

[19] Wenhao Luo, Changjoo Nam, George Kantor, and Katia Sycara. Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage.

In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1488–1496, 2019. 3.1, 3.2, 3.4.2, 4.1, 4.2, 4.3.1, 4.3.1

[20] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. An information-driven and disturbance-aware planning method for long-term ocean monitoring. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2102–2108. IEEE, 2016. 2.2, 3.1, 4.1

[21] Kai-Chieh Ma, Lantao Liu, Hordur K Heidarsson, and Gaurav S Sukhatme. Data-driven learning and planning for environmental sampling. *Journal of Field Robotics*, 35(5):643–661, 2018. 2.2, 3.1, 4.1

[22] Travis Manderson, Sandeep Manjanna, and Gregory Dudek. Heterogeneous robot teams for informative sampling. *arXiv preprint arXiv:1906.07208*, 2019. 3.1

[23] Patrick L. McDermott and Christopher K. Wikle. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *Entropy*, 21(2), 2019. ISSN 1099-4300. 3.1

[24] Brady Moon, Satrajit Chatterjee, and Sebastian Scherer. Tigris: An informed sampling-based algorithm for informative path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022. doi: 10.48550/ARXIV.2203.12830. URL `https://arxiv.org/abs/2203.12830.pdf`. 2.2

[25] Muhammad F Mysorewala, Dan O Popa, and Frank L Lewis. Multi-scale adaptive sampling with mobile agents for mapping of forest fires. *Journal of Intelligent and Robotic Systems*, 54:535–565, 2009. 2.2

[26] Joseph L Nguyen, Nicholas RJ Lawrance, and Salah Sukkarieh. Nonmyopic planning for long-term information gathering with an aerial glider. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6573–6578. IEEE, 2014. 2.2

[27] Ruofei Ouyang, Kian Hsiang Low, Jie Chen, and Patrick Jaillet. Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, page 573–580, Richland, SC, 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450327381. 2.2, 3.1, 4.1

[28] Lishuo Pan, Sandeep Manjanna, and M. Ani Hsieh. Marlas: Multi agent reinforcement learning for cooperated adaptive sampling, 2023. 2.2, 3.1, 4.3.2

[29] Mehdi Rahmati, Mohammad Nadeem, Vidyasagar Sadhu, and Dario Pompili. Uw-marl: Multi-agent reinforcement learning for underwater adaptive sampling using autonomous vehicles. In *Proceedings of the 14th International Conference*

on Underwater Networks & Systems, pages 1–5, 2019. 2.2, 4.3.2

[30] Carl Edward Rasmussen and Hannes Nickisch. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010. 4.3.1

[31] Julius Rückin, Liren Jin, and Marija Popović. Adaptive Informative Path Planning Using Deep Reinforcement Learning for UAV-based Active Sensing. In *IEEE International Conference on Robotics and Automation*, 2022. 2.2, 4.3.2

[32] Tahiya Salam and M. Ani Hsieh. Adaptive sampling and reduced-order modeling of dynamic processes by robot teams. *IEEE Robotics and Automation Letters*, 4 (2):477–484, 2019. doi: 10.1109/LRA.2019.2891475. 2.2

[33] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 802–810, Cambridge, MA, USA, 2015. MIT Press. 3.1

[34] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5622–5632, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. 3.1

[35] Amarjeet Singh, Andreas Krause, and William J Kaiser. Nonmyopic adaptive informative path planning for multiple robots. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1843–1850, 2009. 2.2, 3.1, 4.1, 4.3.2

[36] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 843–852. JMLR.org, 2015. 3.1

[37] Volker Tresp. Mixtures of gaussian processes. *Advances in neural information processing systems*, 13, 2000. 4.3.1

[38] Chenxi Xiao and Juan Wachs. Nonmyopic informative path planning based on global kriging variance minimization. *IEEE Robotics and Automation Letters*, 7 (2):1768–1775, 2022. 2.2

[39] Soo-Hyun Yoo, Andrew Stuntz, Yawei Zhang, Robert Rothschild, Geoffrey A Hollinger, and Ryan N Smith. Experimental analysis of receding horizon planning algorithms for marine monitoring. In *Field and service robotics*, pages 31–44. Springer, 2016. 2.2

[40] Bin Zhang and Gaurav S. Sukhatme. Adaptive sampling for field reconstruction with multiple moblie robots. In Gaurav Sukhatme, editor, *The Path to Autonomous Robots*, pages 1–13, Boston, MA, 2009. Springer US. ISBN 978-0-387-85774-9. 3.1, 4.1

[41] Howard Zhong, Guha Balakrishnan, Richard Strong Bowen, Ramin Zabih, and William T Freeman. Finding maximally informative patches in images. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3.4.3, 4.3.2