

**Transfer Learning for Characterization of Small Unmanned  
Aircraft Communication**

by

**Neeti Wagle**

B.E., Mumbai University, 2006

M.S., Duke University, 2008

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
Department of Computer Science

2015

This thesis entitled:  
Transfer Learning for Characterization of Small Unmanned Aircraft Communication  
written by Neeti Wagle  
has been approved for the Department of Computer Science

---

Prof. Eric W. Frew

---

Prof. Nisar Ahmed

Date \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Wagle, Neeti (Ph.D., Computer Science)

Transfer Learning for Characterization of Small Unmanned Aircraft Communication

Thesis directed by Prof. Eric W. Frew

This dissertation develops a nonparametric, computationally efficient method for modeling the airborne communication environment for small unmanned aircraft systems (sUAS). Transfer learning for Gaussian process regression (GPR) allows the communication model to adapt to the spatial and temporal variations within an environment, and to the variations that arise across UAS hardware and missions.

Environment-specific radio frequency (RF) variations are learned by augmenting a parametric path loss model with a nonparametric Gaussian process (GP), which captures geospatial and time-varying characteristics of signal strength measurements. This dissertation assesses the performance of GP-based communication models through cross validation on 50 sets of real flight measurements collected using two different frequencies, three different airframes, and employing static and mobile transmitters. Measuring the performance using root mean squared error (RMSE) as well as mean standardized log loss (MSLL) evaluates both the predicted estimate and its uncertainty, and shows that the GP models improve prediction accuracy over the path loss model, with the spatio-temporal GPs improving over the spatial GPs.

The value of GP-based communication modeling is further demonstrated through integration with UAS data ferrying to opportunistically learn geospatial variations in RF measurements and use them in communication link scheduling. The iterative ferry-and-learn system is analyzed through a simulation study, showing ferry achieves 80% of optimal within 4 iterations, and 93% after 9 iterations, as the GP is able to converge quickly to the true radio frequency environment. Comparison with parametric least-squares learners in two extremes of RF scenarios shows that the GP better captures the stochasticity of the environment, especially in complicated cases. This demonstrates that learning RF variations using GPs provides a significant boost to the performance

of communication-aware UAS applications.

Because all communication in the environment is affected by the same factors, previously learned GPs contain information that is relevant while learning communication models for subsequent missions and on different UAS platforms. This dissertation proposes forward adaptive transfer learning for Gaussian process regression, FAT-GP, which allows previously learned GP models to be adapted forward as potential sources of knowledge for future learning tasks, which can be especially valuable when limited training data is available for the new task. FAT-GP combines the source task’s previously learned model, the source task’s training data, and target task’s training data to learn the target hyperparameters as well as the correlation between the two tasks. This extension to GPR not only generalizes transfers between GPs using different kernels, but also results in amortization of the training cost.

Such an adapt-and-update framework is in keeping with the philosophy of lifelong learning, and is valuable in UAS and other robotics missions, especially when operating in unstructured and unexplored environments.

## **Dedication**

To my parents, Aai and Baba, and my grandparents, Ama, Mummy and Daddy, who raised me in a world of possibilities, brought to life by hard work.

## Acknowledgements

I want to start by thanking my advisor, Dr. Eric Frew, from whom I have learned the science and the art of defining problems, constructing solutions, and presenting results. I want to thank Dr. Dirk Grunwald, Dr. Mike Mozer, Dr. Doug Sicker, and Dr. Nisar Ahmed for several great discussions that brought different perspectives to my research. Many grants and fellowships have made this interdisciplinary research possible, and I am grateful for their support. Finally, thanks to Glenda and the dissertation group for giving me the tools to set goals and accomplish them.

I have learned a lot from my peers during this process. Brainstorming sessions with Anthony Carfang, Holly Borowski, Craig Turansky, and Jason Durrie have helped me work through innumerable ideas. James Mack, Maciej Stachura, Kevin Rauhauser, Jason Roadman, Jack Elston, Doug Weibel, and Cory Dixon have been central to my education out in the field, where theory is put into practice. Thanks to Nick S., Will, Roger, Tevis, and Nick C. for being great lab-mates in the last couple of years. This has been a fine group of people to share ideas (and beers) with.

Many people have cheered from the sidelines. My parents have been my greatest champions, always patient, and reminding me that the process is as much about hard work as it is about character. Thanks to Seema, Anil, Meghana, and Komal Nerurkar for their constant love (and much needed comic relief). I have a lot of friends to thank for their undying support. Shweta Sanghvi, Nidhi Sehgal, Swati Yelgulwar, and Suyash Shringarpure for always being a phone call away; Aditya Kausik for encouraging me to grow beyond my work; Monish Sp, Aamir Shahpurwalla, and Omkar Pradhan for being family here in Boulder; Gavin Taylor, Mac Mason, and Susanna Ricco for teaching me a lot about robots, research, pie, and basketball: all valuable lessons for life.

I am truly thankful to all these people who have made this a rich journey.

## Contents

<b>Chapter</b>		
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Related Work . . . . .	3
1.2.1	Communication modeling . . . . .	3
1.2.2	Approximation and Adaptability in Gaussian processes . . . . .	6
1.2.3	Transfer Learning . . . . .	7
1.3	Problem Statement . . . . .	10
1.4	Solution Method . . . . .	12
1.5	Contributions . . . . .	14
1.6	Dissertation Outline . . . . .	15
<b>2</b>	<b>Experimental Platform and Evaluation Metrics</b>	<b>17</b>
2.1	Experimental Platform . . . . .	17
2.1.1	UAS System Description . . . . .	17
2.1.2	RF Signal Strength Datasets . . . . .	19
2.2	Empirical Path Loss Radio Model . . . . .	21
2.2.1	Flight Data Validation of the Empirical Models . . . . .	22
2.3	Evaluation Metrics . . . . .	24
2.3.1	Root Mean Squared Error (RMSE) . . . . .	25

2.3.2	Average Normalized Estimation Error (ANEES) . . . . .	25
2.3.3	Mean Standardized Log Loss (MSLL) . . . . .	26
2.4	Summary . . . . .	26
<b>3</b>	<b>Spatio-temporal Models of Airborne RF Communication</b>	<b>27</b>
3.1	Nonparametric Model of Receiver-Dependent Variation . . . . .	28
3.1.1	Spatial and Spatio-temporal GP . . . . .	29
3.1.2	GP based modeling of RF Communication . . . . .	31
3.2	Flight Data Validation Results . . . . .	33
3.2.1	Example Runs . . . . .	33
3.2.2	Gaussian Process Cross Validation . . . . .	37
3.2.3	Computation vs Accuracy Trade-off in GPs . . . . .	43
3.3	Summary . . . . .	45
<b>4</b>	<b>Improving Data Ferrying by Iteratively Learning the RF Environment</b>	<b>46</b>
4.1	Data Ferrying . . . . .	46
4.2	Data Ferry System . . . . .	49
4.3	RF Characterization for Data Ferrying . . . . .	51
4.3.1	Integrated System Overview . . . . .	51
4.3.2	Detailed Integration Considerations . . . . .	53
4.4	Case Study: Results and Discussion . . . . .	55
4.4.1	Initial Configuration . . . . .	55
4.4.2	Ferrying Performance . . . . .	57
4.4.3	GP Performance . . . . .	58
4.5	Learning Comparisons . . . . .	67
4.5.1	Radio Model Fidelity . . . . .	67
4.5.2	Evaluation Setup . . . . .	69
4.5.3	Comparison Results . . . . .	69

4.6	Summary . . . . .	76
<b>5</b>	<b>Forward Adaptive Transfer for Gaussian Process Regression</b>	<b>77</b>
5.1	FAT-GP: Forward Adaptive Transfer for Gaussian Processes . . . . .	79
5.1.1	Target Task Prediction . . . . .	79
5.1.2	Computational Complexity of Hyperparameter Learning . . . . .	84
5.1.3	FAT-GP Demonstration via 1-D problems . . . . .	86
5.2	Relationship between FAT-GP and Target GP . . . . .	90
5.2.1	Impact of Transfer on Mean and Variance Prediction . . . . .	90
5.2.2	Dissimilar source and target tasks . . . . .	95
5.3	Comparing the FAT-GP and Target GP . . . . .	96
5.3.1	RMSE and MSLR Comparisons . . . . .	96
5.3.2	Effect of Target Data Size on Transfer . . . . .	98
5.4	FAT-GP for Transfer Learning Communication Models . . . . .	101
5.5	Summary . . . . .	105
<b>6</b>	<b>Conclusion</b>	<b>106</b>
6.1	Spatiotemporal Characterization of Airborne RF Communication . . . . .	106
6.2	GP Modeling for Data Ferrying and Other UAS Applications . . . . .	106
6.3	Forward Adaptive Transfer Learning . . . . .	107
6.4	Future Work . . . . .	108
6.4.1	Explore-vs-Exploit for Spatio-temporal GPs . . . . .	108
6.4.2	Transfer Planning . . . . .	109
6.4.3	Heteroscedastic GPs . . . . .	109
6.4.4	Application of FAT-GP to other robotic missions . . . . .	109

<b>Bibliography</b>	111
---------------------	-----

## **Appendix**

<b>A</b> Gaussian Process Performance Results	119
<b>B</b> FAT-GP Prediction Analysis	126
B.1 FAT-GP Mean Components . . . . .	126
B.2 Impact of SNR on Variance Components . . . . .	128

## Tables

### Table

2.1	Tempest Flight Data for 433MHz Signal Strength Measurements . . . . .	20
2.2	NexSTAR Flight Data for 2.4GHz Signal Strength Measurements . . . . .	20
2.3	Skywalker X8 Flight Data for Signal Strength Measurements from Mobile Transmitters	21
3.1	Model comparison using cross validation Root Mean Squared Error (RMSE) averaged over datasets in each group . . . . .	38
3.2	Model-wise Predictive Distributions for $z_i$ . . . . .	41
3.3	Model comparison using cross validation Mean Standardized Log Loss (MSLL) av- eraged over the datasets in each group . . . . .	42
4.1	Model-wise Predictive Distributions for $z_i$ . . . . .	65
4.2	Radio Model Comparison Summary . . . . .	76
5.1	Computational cost comparison between AT-GP and FAT-GP (where $N_S > N_T$ ) . .	86
5.2	FAT-GP vs Other GP Configurations . . . . .	97
A.1	GP RMSE Performance for 433MHz Tempest Datasets . . . . .	119
A.2	GP RMSE Performance for 2.4GHz NexSTAR Datasets . . . . .	120
A.3	GP RMSE Performance for Skywalker X8 Mobile-Emitter Datasets . . . . .	120
A.4	GP MSLL for 433MHz Tempest Datasets . . . . .	121
A.5	GP MSLL for 2.4GHz NexSTAR Datasets . . . . .	122

A.6 GP MSLL for Skywalker X8 Mobile-Emitter Dataset . . . . .	122
A.7 Percentage-wise Iterative Training GPs for 433-Tempest Dataset . . . . .	123
A.8 Percentage-wise Iterative Training GPs for 2.4GHz NexSTAR Dataset . . . . .	123
A.9 Percentage-wise Iterative Training GPs for Skywalker X8 Mobile-Emitter Dataset . .	124
A.10 Fixed-Increment Iterative Training GPs for 433MHz Tempest Dataset . . . . .	124
A.11 Fixed-Increment Iterative Training GPs for 2.4GHz NexSTAR Dataset . . . . .	125
A.12 Fixed-Increment Iterative Training GPs for Skywalker X8 Mobile-Emitter Dataset .	125

## Figures

### Figure

1.1	Motivating communication-aware unmanned aircraft applications . . . . .	1
1.2	Problem Overview . . . . .	11
1.3	Solution Overview . . . . .	13
2.1	Unmanned Aerial System (UAS) Components . . . . .	18
2.2	Transmitters used to collect RF signal strength measurements. . . . .	19
2.3	Setup for mobile transmitter experiments . . . . .	21
2.4	Maps of the flight experiment setups with transmitter locations and flight trajectories	23
2.5	Estimation of radio model parameters from RSSI measurements using least squares .	24
2.6	Normplot of RF variations or residuals of the <i>a priori</i> empirical path loss model . .	24
3.1	Overview block diagram of nonparametric spatio-temporal characterization of air- borne RF environments . . . . .	27
3.2	Overview of Gaussian process learning and prediction. . . . .	29
3.3	RF variation modeling block diagram . . . . .	32
3.4	Mean prediction of the spatial isotropic GP of RF variations . . . . .	34
3.5	Variance prediction of the spatial isotropic GP of RF variations . . . . .	35
3.6	Comparison of path loss models and GP mean and variance predictions for 2.4GHz NexSTAR dataset 4 with and without outlier detection . . . . .	36
3.7	GP RMSE performance on groups of datasets collected using stationary transmitters	39

3.8	Error comparison between empirical radio model and spatial GPs of the RF variations for 2.4GHz NexSTAR dataset 10 . . . . .	40
3.9	Error comparison between empirical radio model and spatio-temporal GP of the RF variations for 2.4GHz NexSTAR dataset 10 . . . . .	40
3.10	GP MSLM on groups of datasets collected using stationary transmitters . . . . .	42
3.11	Training time vs prediction RMSE trade-off for increasing training set sizes obtained by iteratively training GPs percentage-wise (top row) and using fixed increment (bottom row). . . . .	44
4.1	Overview of the data ferrying problem. . . . .	46
4.2	Integrating RF characterization with data ferrying involves iterating through planning ferry paths based on a predicted environment, and improving the predictions based on signal strength samples gathered while ferrying. . . . .	53
4.3	True RF environments. . . . .	56
4.4	Initial estimates of RF environments. . . . .	56
4.5	The ferry path evolves as the RF model predictions are improved from (a) iteration 1 to (b) iteration 20. . . . .	57
4.6	Planned and actual ferry throughput performance, ferrying through the estimated RF models and the true RF environment. . . . .	58
4.7	Evolution of ferry paths and predicted environments from (a) a previous good trajectory through (b) the region north of node B with previously over-estimated signal strength, then learning to avoid that region in (c). . . . .	59
4.8	RF variations for the <i>a priori</i> models of (a) Node A and (b) Node B. . . . .	59
4.9	Mean prediction for GP learned on RF variations for Node B . . . . .	60
4.10	Variance for GP learned on RF variations for Node B . . . . .	60
4.11	Model error comparison for Node B . . . . .	62
4.12	GP validation error over 20 iterations. . . . .	62

4.13	GP performance consistency (ANEES) over 20 iterations. . . . .	64
4.14	Mean Standardized Log Loss (MSLL) for the accumulated path . . . . .	65
4.15	Mean Standardized Log Loss (MSLL) for the white boxed region. . . . .	66
4.16	Mean Standardized Log Loss (MSLL) for uniform grid over environment. . . . .	66
4.17	Case 1: a relatively clean omni-directional truth environment. . . . .	70
4.18	Case 2: a much more complicated truth environment, with dipole antennas, noise, and many interferers. . . . .	71
4.19	Initial estimates, i.e. <i>a priori</i> models of the RF environment, used in both cases. . .	72
4.20	Ferry’s expected (dashed) and actual (solid) throughput performance compared be- tween the three learners, averaged over 8 runs. . . . .	73
4.21	Path (solid) and full environment (dashed) validation RMS error comparison of the three learners, averaged over 8 runs of the near-omnidirectional environment . . . .	73
4.22	Path (solid) and full environment (dashed) validation RMS error comparison of the three learners, averaged over 8 runs of the complicated dipole environment . . . . .	74
4.23	Mean Standardized Log Loss (MSLL) for three learners over the entire accumulated path, averaged over 8 runs of the near-omni-directional environment . . . . .	75
4.24	Mean Standardized Log Loss (MSLL) for three learners over the entire accumulated path, averaged over 8 runs of the complicated dipole environment . . . . .	75
5.1	FAT-GP Overview . . . . .	79
5.2	Example where target task $T$ is a translation of source task $S$ i.e. $y_T = y_S + 2$ . . .	87
5.3	Example where target task $T$ is a scaling of source task $S$ i.e. $y_T = 2y_S$ . . . . .	88
5.4	Example $y_T = -2 \times y_S$ which demonstrates the value of $\lambda$ in learning negative correlations . . . . .	89
5.5	FAT-GP mean prediction components, and its comparison to Target GP . . . . .	91
5.6	FAT-GP variance prediction components and its comparison to Target GP . . . . .	93
5.7	Reduction of uncertainty in FAT-GP . . . . .	93

5.8	Coefficient of transfer $b$ as a function of SNR . . . . .	94
5.9	Effect of Target Data Size on MSLL for FAT-GP and other learning configurations .	98
5.10	Effect of Target Data Size on RMSE for FAT-GP and other learning configurations .	99
5.11	FAT-GP vs Target GP MSLL Comparison . . . . .	99
5.12	FAT-GP vs Target GP RMSE Comparison . . . . .	100
5.13	Evolution of FAT-GP's mean components from iteration 1 with 10 target samples (shown in top row) to iteration 56 with 65 samples (shown in bottom row) . . . . .	100
5.14	True environments in source and target tasks . . . . .	101
5.15	Source GP learned using 2500 samples of the source task . . . . .	102
5.16	Two scenarios with limited target training set of size 50 available. . . . .	102
5.17	Target GP and FAT-GP learned using 50 samples of the target task taken over the entire environment . . . . .	103
5.18	Target GP and FAT-GP learned using 50 samples of the target task taken over half the environment . . . . .	104

# Chapter 1

## Introduction

### 1.1 Motivation

A wide variety of unmanned aircraft system (UAS) missions critically depend on sensing and communication. Recent work in unmanned aircraft (UA) applications like cooperative path planning, target tracking, and data ferrying has recognized the role of communication in these tasks and designed control systems that specifically take bandwidth and sensing constraints into consideration [17, 80, 46, 94, 38, 19]. Fig. 1.1 shows a few examples of such communication-aware unmanned aircraft (UA) applications.

However, the performance of these tasks is limited by the communication models. In order to exploit the capacity of communication and sensing onboard these unmanned aircraft it is important to design refined models of the radio-frequency (RF) propagation.

The need for communication models is greater for UAS and other robotic missions operating

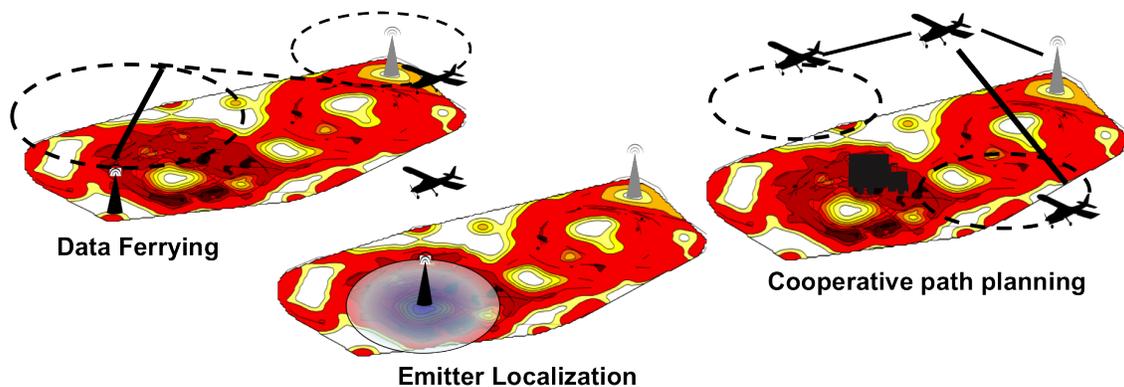


Figure 1.1: Motivating communication-aware unmanned aircraft applications

in unstructured and unexplored environments. However, the challenge in deriving or learning the model is also exacerbated by the inability to collect dense, extensive data in the environment. Additionally, models which make simplifying assumptions or generalizations fail to capture the environment specific artifacts and variations that arise in the field. To cope with the diverse operating conditions that such missions present, it is important for models to adapt as they observe the environment. This ability to learn and adapt is key in identifying sources of noise such as interferences, obstacles, and even hostile adversaries.

The communication for small UAS can be modeled using parametric methods which assume a functional form, and use the training data to estimate the parameters of the function. On the other hand, nonparametric models only assume that training data are correlated, and perform all inference using the training data directly. Modeling the communication using parametric and nonparametric approaches provides different advantages. While physics-based parametric models of radio propagation are complicated, estimating the parameters, i.e training the model, has a lower computational cost. On the other hand, nonparametric models, which have higher time and space complexities, provide tuned solutions entrenched in observed data. This direct trade-off between the cost and expressibility of the model sets up an inherent compromise in developing adaptable solutions in complex environments.

The training cost also impacts the ability to relearn, either when new data becomes available, or when the learning task evolves. A need for relearning may arise when a learned model is moved to different hardware platform, or when carrying out missions in dynamic environments which encounter changes due to several explicit and implicit factors. If these changes to the underlying learning task are drastic, they may render the existing model obsolete, requiring expensive and extensive data collection and training phases be repeated. However if the changes are smoother, it is desirable that, when possible, the model be updated by adapting to variations in the learning task itself.

Such an adapt-and-update framework is in keeping with the philosophy of lifelong learning [84], which would allow robots to accrue knowledge and exploit existing relationships between

entities in the world. This outlook examines the possibility that outdated models may still contain relevant and useful information, and considers how it can be reused to efficiently construct a new, updated model. Such a transfer of information can not only enrich the consequent model but also translate into significant savings in the training and learning process.

Transfer learning uses knowledge present in an one task and applies it to another, related task [58]. The objective is to identify the nature of these similarities, extract pertinent information from the source task, and use it effectively in the target task. In the case of RF propagation, we hypothesize that the environmental factors latent in the signal strength measurements are common to signal power reception in the same environment. Thus, the tasks of learning the RF model for different flights or using different radio hardware are inherently related through the common environment in which they operate.

Developing solutions that can seamlessly adapt to these various scenarios is central to allowing robots and UAS to perform long term missions maximizing knowledge and endurance, and minimizing cost. A formalized transfer learning framework for Gaussian processes has great value in several control systems. The ability to model and predict the sensing of wireless signals and the communication channels can enhance the performance of communication-aware path planning [80], data ferrying [10], and WiFi indoor robot localization [2, 20]. In addition, it can also improve the sensor model in estimation problems like target localization [86, 87], where the location of a radio is determined based on the signal strength observed in the environment. Finally, it would also be useful in other tasks using GPs to model time-varying fields such as temperature, winds, and other dynamic phenomena.

## 1.2 Related Work

### 1.2.1 Communication modeling

In cellular and communication networks, radio propagation models are used to assess the coverage provided by the selected transmitters or access points [61]. Increased pervasiveness of mobile

devices also spurred interest in radio-frequency modeling for location-aware computing [2]. These models have been further tuned for performing range estimation using wireless signals, required for high accuracy localization of mobile phones and robots [41, 45]. In this manner, several different platforms and their applications, have contributed to the advancement of communication modeling techniques.

#### 1.2.1.1 Physics based communication models

The most basic RF physics models assume that radio wave propagation behaves according to electromagnetic wave theory without disruption [66]. Many others add parameters or corrective terms to account for specific artifacts that affect signal strength measurements [61]. As such this class of models has a very restrictive view of RF behaviors, often based on small or localized set of observations.

In contrast, ray tracing approaches work with a large and dense set of measurements, using prior knowledge about the obstacles in the terrain, to map out every path that a signal transmission may take, and thus predict received signal strength. However, not only is this modeling technique expensive and time consuming, it makes the rigid assumptions that the environment is static.

The final class of physics based models capture the fading effects responsible for some of the variations in RF measurements. These fading effects for settings with line of sight (LOS) and non-LOS have been captured in the Rician and Rayleigh models, and generalized by the Nakagami distribution [2, 53, 61]. While empirical assessment has shown that these models are widely applicable, they cannot capture environment specific obstacles and interferences, and hence cannot provide a complete communication modeling solution.

A detailed survey and taxonomy of physics based communication models can be found in [61].

### 1.2.1.2 Gaussian process communication modeling

Since traditional parametric radio models use physics-based formulations [14, 47], they fail to account for RF variations and other artifacts that stem from interference and unknown environment-specific characteristics. This missing flexibility can be achieved by using nonparametric models which take a data driven approach, and can be tuned to the operating environment. Gaussian processes are nonparametric probabilistic generalizations of linear regression. This dissertation augments such a parametric, empirical path-loss radio model [66] with a nonparametric Gaussian process [67], and learns local, receiver-location dependent RF variations.

Gaussian processes provide a flexible, nonparametric framework which models the distribution over functions. The learning phase of a GP involves the learning of its hyperparameters which encode how correlation in the input space transforms to correlation in the output space. In order to make a prediction for an unseen input, the GP calculates the input's correlations with the inputs in the training set and uses the hyperparameters to compute a mean and variance prediction for the corresponding output. Because a GP is a continuous model it does not limit the input locations at which it can be used for prediction. If the model is asked for a prediction at a location far from all the training samples, the GP will indicate the lack of confidence in the prediction via a large value of variance. The details of the Gaussian process framework can be found in [74, 67, 68].

Because of their ability to capture input-dependent variations, GPs have been popular in modeling geospatial characteristics such as the coverage of wireless networks [61] and indoor RF communications for ground robots [21]. They have also been extended to a latent variable version to learn the location map simultaneously and consequently reduce calibration effort [20]. One perspective to RF modeling [54, 24, 22] used the GP for probabilistic multi-scale channel modeling, and captured the signal variations introduced by large scale and small scale fading. That work viewed received signal strength as a combination of distance dependent path loss power, large scale or shadowing component, and multipath small scale reflections [53]. Consequently, the spatial correlations and noise components of these fading effects were incorporated into a stationary spatial

GP [54, 24]. Model performance was validated by showing that improving the prediction for links between nodes and the base station led to an improvement in the performance of communication-aware motion planning for ground robots [26, 49, 48, 23].

Approaches that ignore temporal dependencies and limit their communication modeling to spatial, stationary designs are unable to capture the space-time interactions that complex obstacles and interference sources introduce into environments. Stationary models throw away some of the complexity by assuming that the entire space adheres to a single distance of decorrelation. That shortcoming was challenged and additional latent GPs were used to build non-stationary models of radio frequency propagation [31, 71, 62]. However, these required extensive and dense training data which was not only time consuming but also infeasible for applications that required modeling of fairly inaccessible or unexplored environments. Alternatively, the model’s expressive ability has also been increased by including temporal information into a stationary GP, which augments the input space with another dimension. That approach has been used to model dynamic wind fields [42], and this dissertation draws from it to capture the dynamic behavior of radio frequency environments. Empirical assessment of spatio-temporal models for environmental monitoring of lake temperatures and forest light intensities found the stationary and non-stationary versions of the covariance functions to have comparable performance, but spatio-temporal models outperformed spatial models [77].

### 1.2.2 Approximation and Adaptability in Gaussian processes

While flexibility and probabilistic design of GPs makes them attractive, their training cost and space requirements introduces challenges [78]. Depending on the constraints of the application, various approximations have been used to speed up their use.

Much attention has been given to performing online computation with large training sets, and solutions for fast sparsification have been developed [92, 72, 75, 43, 64, 78, 13]. These approaches use sufficient statistics to derive an approximation to the full GP, which is based on a selected set of training samples. Thus, by performing additional, expensive computation at training time, the

time and space complexities for prediction are reduced.

Because sparsification is expensive, resource constrained robotic applications with streaming data have taken a different outlook. For example, decentralized applications can afford to discard old data and only retain the most recent observations since the newest measurements made locally are of most interest [93]. Another approach has been to incorporate new data as it arrives by using stochastic gradient descent to learn hyperparameters in a cheap yet efficient manner [90, 27].

Unlike passive approaches which do not optimize their data collection, active learning uses mutual information to select locations whose measurements will be most informative about the remaining environment. Such planned data collection controls the amount of training data as well as the quality of information it brings [29, 39, 40]. This explore-and-exploit paradigm provides a systematic way to capture new changes in dynamic environments by switching from exploit to explore and gathering more informative measurements.

However, changes to the learning task itself still warrant restarting the data collection and learning effort. Instead of repeating these expensive and extensive phases, transfer learning can help reuse the relevant knowledge from the old model.

### 1.2.3 Transfer Learning

In traditional machine learning, it is assumed that the training samples and the future samples come from the same space and have the same distribution. However, this may not be true when the underlying distribution of the problem space is changing or if the model is learned on one corpus and intended to be used on a different but related corpus. Such learning problems where the original source of training samples and targeted testing samples come from different domains fall into the category of transfer learning problems [58].

Transfer learning refers to the problem of learning how information present in related tasks can be used to improve one or both of these tasks. Depending on what information is available and how the learning is done, such a transfer of knowledge between tasks is referred to by several names like learning to learn, knowledge transfer, multitask learning, lifelong learning, or meta learning [58].

Domain adaptation focuses on exploiting the synergy between similar tasks operating in different domains on different data distributions.

Out of the various types of transfer learning problems, our research is specifically focused on the subproblem of transductive transfer learning, wherein the original and goal tasks are the same but deal with data having different distributions [1]. In the field of natural language processing (NLP), this problem is more commonly referred to as domain adaptation [58], and has been solved using different models like ensemble classifiers [18] and logistic regression [65]. Unfortunately, majority of the transductive transfer learning work is focused on classification problems.

### 1.2.3.1 Hyperparameter Based Transfer Learning

Because most machine learning models learn parameters like weights, coefficients, or hyperparameters from the training data, transferring parameters has been one approach for transductive transfer learning. The methods in this category are, in effect, using these parameters as prior information for the goal task. For example, [44] used meta-features from the original task for feature selection in the goal task, inferring their different *a priori* relevance.

### 1.2.3.2 GP-based Transfer Learning

While parameter transfer methods have also been applied to Gaussian processes before ([51, 43, 73]), they all focused on multitask learning. All of these methods built on a perspective introduced in [51] whereby the regression function of each related task was viewed as a sample from a Gaussian process. This concept relates back to the GP being a distribution over functions. Under this perspective, training with a number of related tasks became equivalent to training on a set of function samples. Consequently, using this model on a new goal task was equivalent to testing a new function sample. Analogous to traditional regression, the more the number of related tasks in the training sample, better the performance on the new goal task. This perspective truly related the transfer learning paradigm to the traditional regression methods. However, it limited the relationship between tasks to a specific case whereby they all belonged to the same underlying

distribution.

More recently semi-parametric Gaussian process approaches have extended the GP framework in a natural manner for domain adaptation of a source or secondary task while learning a primary or target task [12]. Adaptive Transfer Learning (AT-GP) is a prominent example of this approach [7]. This research has its roots in indoor-localization transfer learning designed to reduce calibration costs ([97, 57, 98, 59]). This involved multiple schemes, each of which reused calibration data and was responsible for extending the model in one particular way: to new spaces, future times and different devices. However, because the work was based on a fingerprinting localization technique, it was not only restricted to indoor environments but also incapable of handling significant changes in the geometry or conditions of the environment. These shortcomings were overcome by AT-GP, which was GP-based and adaptive, and thus provided a more general modeling and transfer learning platform.

AT-GP draws inspiration from multitask learning and trains target and source tasks simultaneously. It learns the source and target tasks with the same kernel (function and hyperparameters). It learns the similarity between these tasks via a transfer kernel, which is a specially defined covariance matrix, and consequently tunes the impact that the source task has on predictions for the target task. However, samples from the target task are given higher importance while making predictions for a target sample. AT-GP is a prominent example of the semi-parametric transfer learning approaches which naturally extend the GP framework for domain adaptation.

Unfortunately, AT-GP constrains the two tasks to have an implicit transfer even when the tasks have zero or little correlation. In addition, the AT-GP formulation does not allow transfer from already learned source tasks to new target tasks, only providing a method for simultaneously learning the two tasks. When transfer learning is employed in robotic learning tasks with large training sets for the source and relatively small target training set, not much information can be gained by tuning the source’s hyperparameters. Consequently, the additional computational complexity is undesirable, and performing a one-way transfer from source to the target can result in significant amortization of training costs.

### 1.3 Problem Statement

Compared to indoor and mobile ground robots, fixed-wing UAS typically operate over large areas. These missions are time and resource constrained, and their data collection for learning their airborne communication is typically done over larger distances, at faster timescales, taking vehicle and environmental dynamics into consideration. While outdoor radio propagation is closer to the ideal free space behavior [66], in some cases where the unmanned aircraft is operating close to the ground or in urban canyons, interferers and obstacles cause local and correlated fading effects making the RF environment as complex as indoor environments. Prior information about factors that may influence the communication, such as interference sources and obstacles, may not be available, especially during missions in unexplored and unstructured environments. Finally, the use of commercial off the shelf (COTS) hardware in resource-constrained small UAS missions makes it imperative to adapt and tune the RF model in the field.

Using relatively low resolution measurements, to implicitly capture the latent factors affecting the communication, and learn high accuracy models of these complex radio frequency fields, is a challenging problem. No work has explored combining parametric and non-parametric (Gaussian processes) models for describing the airborne communication environment of small fixed wing unmanned aircraft. Because of the complexity of the problem, it is important that communication modeling solutions be assessed using real flight data collected under varying conditions. A key part of the assessment is the selection of appropriate metrics which allow comparison between parametric and nonparametric models. Such a comparison should not only include likelihood and accuracy of model estimates, but also factor in model uncertainty.

While nonparametric models provide high prediction accuracy, their tight coupling to training data make them susceptible to changes. These changes can be implicit such as modification in the radio field due to environmental factors, or explicit such as change in the hardware on which the model is employed. Such changes may significantly alter the learning task itself, rendering the model inaccurate and obsolete, and ultimately require that expensive and extensive data collection



Figure 1.2: Problem Overview

and training phases be repeated. In order to allow the learned model to be used across flights or missions, as well as across UAS platforms in heterogeneous fleets, it is desirable that, when possible, the model adapt to variations in the learning task itself.

Because all communication in the environment is affected by the same factors, each learned communication model contains information that is relevant to communications on different UAS or during subsequent missions. Hence, when learning communication models for these new targets, the previously learned model can act as a source of information. Measurements taken while learning this new target task help establish the relationship or overlap between the source and target. By transferring the common knowledge, the target model can be learned with minimal number of measurements and limited number of computations.

There are several challenges in transferring knowledge from an existing model to a related task. First and foremost, it is important to identify what nugget of information from the source should be transferred. Second, how this information is to be included into learning the target has

to be determined. Finally, the extent of transfer should be ascertained based on the relatedness of the two tasks. Addressing these design requirements is necessary for making the transfer efficient and beneficial to the learning of the target task, and the success of the transfer hinges on it.

This dissertation addresses how to design a data-driven, cost-effective method of modeling communication for small unmanned aircraft systems (sUAS), which takes all these challenges and desiderata into consideration. The emphasis is on developing a solution that not only adapts to the spatial and temporal variations within an environment, but also to the variations that arise across UAS hardware and missions.

## 1.4 Solution Method

This thesis augments the *a priori*, parametric path loss radio propagation model (obtained from the Friis transmission equation) [66] with a nonparametric Gaussian process model by learning the RF variations introduced due to the environment’s medium, obstacles and interference sources. Because these variations depend on the receiver’s location and other local effects, they are modeled using a spatio-temporal Gaussian process (GP). This combination of path loss model with GP of RF variations is shown by the orange block in Fig. 1.3.

GPs provide the flexibility to accommodate different RF propagation behaviors using a single methodology. In addition, because a GP is a continuous model it does not suffer from quantization errors during prediction. Spatio-temporal GPs use correlation between training samples to model the underlying process [67], and these relationships between the training samples are captured by learning the hyperparameters of the GP. Using these hyperparameters, the RF variations at an unseen location at another time can be predicted by computing its correlation with the training samples. This prediction is in the form of a probability distribution function because Gaussian processes are fully probabilistic in nature and are equipped to handle uncertainty. As shown in Fig. 1.3, communication-aware UAS missions can use the model’s predictions to assess their bandwidth and sensing limitations which operating the environment.

This dissertation assesses the performance of GPs in capturing diverse RF fields through cross

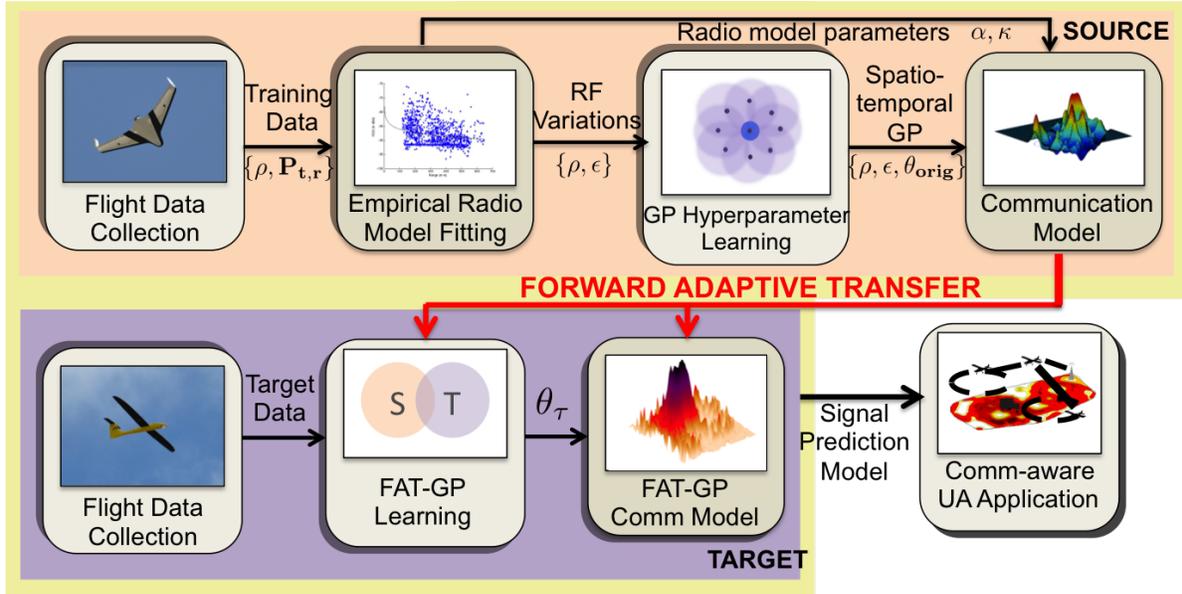


Figure 1.3: Solution Overview

validation on 50 sets of real flight measurements collected under many diverse conditions. It also demonstrates how the UAS mission of data ferrying benefits by learning the geospatial variations in the field and using them in communication link scheduling. Both these assessments are done using metrics that evaluate accuracy as well as uncertainty by means of likelihood.

Unfortunately, because the GP’s training is tightly coupled to its training setup, using this model in subsequent missions or on different UAS can significantly reduce the predictive power of the model. However, the existing GP model still contains valuable information about implicit factors that affect communication in this environment. Hence we employ transfer learning to allow the existing GP of variations to act as a source of information for updating or tuning the communication model to these other setups. This dissertation proposes a forward adaptive transfer learning for Gaussian process regression, or FAT-GP, which allows previously learned GP models to be adapted forward as potential sources of knowledge for future learning tasks, as shown in the purple block in Fig. 1.3.

FAT-GP combines the source task’s previously learned model, the source task’s training data, and target task’s training data to learn correlations within the target task, as well as the correlation

between the two tasks. A conceptual overview of the algorithm is presented in Figure 5.1. Finally, FAT-GP decouples the kernel and hyperparameter selection for the target task from those of the source task. This extension not only generalizes transfers between GPs using different kernels, but lets the source task’s large covariance matrix be precomputed, resulting in amortization of the training cost.

## 1.5 Contributions

Several subproblems have to be addressed before the proposed solution can be implemented. Solutions to these subproblems are the contributions of this dissertation, and are as follows:

- (1) **Experimental assessment of spatio-temporal models of small unmanned aircraft RF communication.** This dissertation assesses how spatial and spatio-temporal Gaussian processes improve airborne communication modeling over the *a priori* path loss model. This is done using a total of 50 datasets of signal strength measurements taken during flight experiments in the field. These datasets include 433MHz and 2.4GHz signal strength measurements from static or mobile transmitter collected using three different airframes. The GP is compared to the *a priori* model using root mean squared error (RMSE) and mean standardized log loss (MSLL) which evaluate the predicted estimate as well as its uncertainty. The prediction accuracy is also compared against the training cost to characterize the inherent trade-off between the training phase computation and prediction performance.
- (2) **Simulation based assessment of improvements in UA mission performance.** The GP based RF estimation is integrated with the UA mission of data ferry planning, factoring in considerations necessary for a smooth iterative ferry-and-learn process. The system is analyzed through simulation, and the contributions consist of two parts
  - With opportunistic GP learning, the ferry performance improves significantly, achieving 80% of optimal effective throughput within 4 iterations, and 93% after 9 iterations, as the GP is able to converge quickly to the true radio frequency environment.

- The iterative ferry-and-learn process is also implemented with parametric least squares modeling techniques, and the GP is compared to these alternative learning systems. This shows how the geospatial modeling and non-parametric adaptability of the GP allows it to capture environmental artifacts, and thus perform well regardless of the characteristics of the RF environment.

(3) **Theoretical and experimental assessment of forward adaptive transfer learning for Gaussian process regression (FAT-GP).** This dissertation develops FAT-GP, a forward adaptive transfer learning method for Gaussian process regression, which provides a framework for robotic learning tasks to leverage previously learned GP models, which can be especially valuable when limited training data is available for the new task. Theoretical analysis shows that FAT-GP amortizes cost by avoiding relearning the existing source GP. Experimental assessment demonstrates how including the knowledge from the source GP reduces the root mean squared error (RMSE) as well as the uncertainty in the predictions, even when very few training samples are available for the target task.

## 1.6 Dissertation Outline

Chapter 2 details the UAS platforms used in flight experiments for collecting signal strength measurements, introduces the traditional path-loss parametric radio model and outlines its shortcomings, and finally discusses the metrics that the later chapters use to assess and compare the nonparametric solutions to their parametric counterparts. Chapter 3 presents the design and experimental assessment of the spatial and spatio-temporal GP for capturing RF variations. This ability to learn and adapt to the environment using GP models of communication is integrated into the UAS mission of data ferrying in Chapter 4, and the performance is assessed and compared with parametric estimation technique via simulations. Chapter 5 extends the adaptability of the GP communication models to work across missions and hardware changes by using old models as sources of information in new learning tasks via forward adaptive transfer. Chapter 6 summarizes

the results of this dissertation, discusses shortcomings, and consequently identifies opportunities of future work.

## Chapter 2

### Experimental Platform and Evaluation Metrics

This dissertation aims to build improved models for airborne RF communication of small unmanned aircraft. These models are evaluated using data and measurements from flight experiments in the field. This chapter presents an overview of the different UAS and radios deployed, the configuration of the flight experiments used for data collections, and characteristics of the various measurement datasets. As a baseline, these datasets are modeled using the parametric, empirical path-loss model, and its performance shortcomings are quantified to motivate the need for nonparametric solutions.

Finally, a variety of metrics are introduced which evaluate different aspects of model performance. They not only assess the prediction estimates of these models, but also their uncertainties, and their ability to explain the observed data. Together these metrics provide a systematic way for comparing the competing parametric and nonparametric models of communication.

#### 2.1 Experimental Platform

##### 2.1.1 UAS System Description

Signal strength measurement datasets were collected using three different unmanned aerial systems (UAS). All three setups used different airframes and different radio frequencies in flight operations.

The three airframes used were the NexSTAR, Tempest and Skywalker X8 UAS. The NexSTAR aircraft, shown in Fig. 2.1a, was built from the Hobbico's hobby kit, consisted of a balsa wood



(a) NexSTAR UAS



(b) Tempest UAS



(c) Skywalker X8 UAS

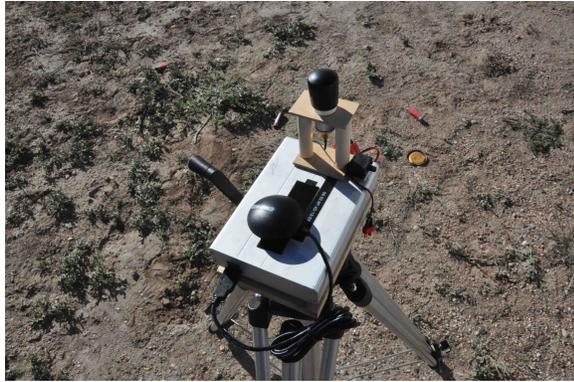


(d) RECUV Ground Station Van

Figure 2.1: Unmanned Aerial System (UAS) Components

airframe, with a wingspan of 1.75m, and was modified in order to outfit with the necessary sensors [69]. The Tempest aircraft, shown in Fig. 2.1b, is a sturdy fiberglass airframe with a wingspan of 3.2m, and can carry up to 5.8kgs during take-off [70]. The Skywalker X8 [33], shown in Fig. 2.1c, has a foam airframe and wingspan of 2.12m.

While Cloud Cap Technology's Piccolo Plus Autopilot [16] was used during the NexSTAR flights, the Tempest and Skywalker X8 flights used the SwiftPilot from Black Swift Technologies [5]. The autopilots communicate with the aircraft via the ground station (shown in Fig. 2.1d), which has a dedicated 900MHz link with the aircraft at all times. Further details of these UAS shown in Fig. 2.1 can be found in [79].



(a) 2.4GHz WiFi antenna



(b) Beacon emitting 433MHz signals

Figure 2.2: Transmitters used to collect RF signal strength measurements.

### 2.1.2 RF Signal Strength Datasets

In all the flight operations, signal strength measurements were logged onboard the aircraft. For example, during the NexSTAR flights, a WiFi omni-directional, spherical antenna on the aircraft measured and logged the signal strength of a 2.4GHz WiFi transmitter on the ground, as shown in Fig. 2.2a. In experiments conducted before June 2009, the antenna was a regular 2.4GHz dipole antenna, whereas afterwards it was switched to a WiFi-Plus omni antenna [91]. Similarly, during the Tempest flights, the aircraft logged 433MHz measurements from a mesh network radio (MNR) [6], shown in Fig. 2.2b. The Tempest and NexSTAR flights both used a static i.e. stationary transmitter, which was placed at a fixed location, whose GPS coordinates were recorded. Knowing the true location of the transmitter and the aircraft at all times, the range for each signal measurement could be calculated. Tables 2.1 and 2.2 each enumerate 20 and 18 signal measurement datasets, which were collected by Tempest and NexSTAR from 433MHz and 2.4GHz static-transmitters, respectively.

While collecting measurements for mobile transmitters, a 433MHz beacon and a 2.4GHz WiFi transmitter, along with a GPS sensor, were attached to the roof of the van, as shown in Fig. 2.3. This setup ensured that, once again, ground truth on the transmitter location was logged. Table 2.3 lists 12 datasets of measurements collected by a Skywalker X8 UAS from a mobile transmitter.

Table 2.1: Tempest Flight Data for 433MHz Signal Strength Measurements

No.	Dataset	Flight	Emitter	No. Measurements
1	2013-01-26	1	mnr0	1258
2	2013-02-08	1	mnr0	4593
3	2013-02-08	2	mnr0	2704
4	2013-03-05	1	mnr0	425
5	2013-03-05	2	mnr0	543
6	2013-03-07	1.1	mnr0	166
7	2013-03-07	1.2	mnr0	208
8	2013-03-07	1.3	mnr0	1574
9	2013-03-13	1.1	mnr0	427
10	2013-03-13	1.2	mnr0	904
11	2013-03-22	1	mnr0	993
12	2013-04-12	1	mnr0	549
13	2013-04-12	1	mnr1	338
14	2013-04-30	1	mnr0	399
15	2013-04-30	1	mnr1	133
16	2013-05-14	1	mnr0	420
17	2013-05-14	2	mnr0	376
18	2013-05-17	1.1	mnr0	283
19	2013-05-17	1.2	mnr0	205
20	2013-06-27	1	mnr0	504

Table 2.2: NexSTAR Flight Data for 2.4GHz Signal Strength Measurements

No.	Dataset	Flight	Emitter	No. Measurements
1	2008-10-14	1	mnr0	1628
2	2009-03-13	1	mnr0	3777
3	2009-03-13	1	mnr1	3785
4	2009-03-13	1	trailer	4529
5	2009-05-12	1	mnr0	750
6	2009-05-12	1	mnr1	745
7	2009-05-12	1	trailer	1993
8	2009-05-20	1	mnr0	2737
9	2009-05-20	1	mnr1	2697
10	2009-05-20	1	trailer	3514
11	2009-09-24	1	mnr0	1920
12	2010-09-02	1	mnr2	2484
13	2010-09-02	2	mnr2	2551
14	2010-09-02	2	laptop	1630
15	2010-09-13	1	mnr2	2502
16	2010-09-13	2	mnr2	1995
17	2010-09-13	2	laptop	1995
18	2011-07-12	1	mnr0	255



Figure 2.3: Setup for mobile transmitter experiments

Table 2.3: Skywalker X8 Flight Data for Signal Strength Measurements from Mobile Transmitters

No.	Dataset	Flight	Emitter	No. Measurements
1	2013-09-24	1	433 beacon	979
2	2013-09-24	2	433 beacon	2162
3	2013-10-17	1	433 beacon	1995
4	2013-10-17	1	2.4GHz WiFi antenna	1975
5	2013-10-25	1	433 beacon	1957
6	2013-10-25	1	2.4GHz WiFi antenna	1926
7	2013-10-25	2	433 beacon	1056
8	2013-10-25	2	2.4GHz WiFi antenna	1749
9	2013-11-01	1	433 beacon	1181
10	2013-11-01	1	2.4GHz WiFi antenna	2080
11	2013-11-01	3	433 beacon	967
12	2013-11-01	3	2.4GHz WiFi antenna	1899

These datasets in Tables 2.1-2.3 will be used in RF characterization experiments in this thesis.

## 2.2 Empirical Path Loss Radio Model

The signal power  $P_{t,r}$ , from a transmitter at location  $\mathbf{p}_t$ , received at a location  $\mathbf{p}_r$ , can be empirically modeled with the path loss model [66]

$$\begin{aligned}
 P_{t,r} &= P_0 \left( \frac{d_0}{\|\mathbf{p}_t - \mathbf{p}_r\|} \right)^\alpha G_{t,r} \\
 &= \frac{k_0}{\|\mathbf{p}_t - \mathbf{p}_r\|^\alpha} G_{t,r}
 \end{aligned} \tag{2.1}$$

where  $P_0$  and  $d_0$  are the reference power and reference distance,  $k_0 = P_0 d_0^\alpha$ , and  $G_{t,r}$  captures unmodeled variations in the path loss. Receivers measure the signal power<sup>1</sup> in decibels

$$\begin{aligned}
 P_{t,r}^{dB} &= 10 \log_{10} \left( \frac{k_0}{\|\mathbf{p}_t - \mathbf{p}_r\|^\alpha} G_{t,r} \right) \\
 &= 10 \log_{10} \left( \frac{k_0}{r^\alpha} \right) + 10 \log_{10}(G_{t,r}) \\
 &= 10 \log_{10}(k_0) - \alpha 10 \log_{10}(\|\mathbf{p}_t - \mathbf{p}_r\|) + \nu \\
 &= \kappa - \alpha 10 \log_{10}(\|\mathbf{p}_t - \mathbf{p}_r\|) + \nu
 \end{aligned} \tag{2.2}$$

where  $\kappa = 10 \log_{10}(k_0)$ , and  $\nu$  is additive white Gaussian noise (AWGN). The terms  $\kappa$  and  $\alpha$  are radio model parameters and depend on the operating environment. While  $\kappa$  depends on the radio source itself, the path loss exponent or the propagation decay exponent  $\alpha$  depends on environmental factors like atmospheric conditions and objects present in the environment.

Traditionally, simplistic empirical radio models  $\Xi_{t,r}$  have used only path loss to predict the power received at a given range from the transmitter.

$$\Xi_{t,r} = \Xi_t(\mathbf{p}_r) = \kappa - \alpha 10 \log_{10}(\|\mathbf{p}_t - \mathbf{p}_r\|) \tag{2.3}$$

These empirical models assume that the noise in the measurement  $z_{i,j}$ , for transmitter  $i$ , made at receiver location  $\mathbf{p}_j$ , is the additive white Gaussian noise (AWGN)  $\nu$ .

$$z_{i,j} = \Xi_i(\mathbf{p}_j) + \nu \tag{2.4}$$

Under this assumption, the radio model parameters,  $\kappa$  and  $\alpha$ , can be estimated from a dataset of measurements,  $\mathbf{z} = [z_{i1}, z_{i2}, \dots, z_{iN}]^T$ , collected by measuring a known transmitter  $i$ 's received power at  $N$  known locations  $\mathbf{p}_j$  for  $j = 1, \dots, N$ . The estimation is done by fitting the signal measurements using least squares.

### 2.2.1 Flight Data Validation of the Empirical Models

Fig. 2.4 shows flight experiment maps for 3 datasets with the unmanned aircraft's trajectory (shown as a blue solid line) as it collected signal strength measurements from the transmitter

---

<sup>1</sup> Often, this is derived from received signal strength indicator (RSSI) whose mapping to received signal power is unique to the manufacturer.

(shown as a red triangle).

Because the true locations of both the receiver and transmitter are known, we can compute the range at which each measurement was made. This range data can be used to generate the plots in Fig. 2.5, which shows the each measurement by a blue mark. Fitting these measurements using least squares estimates the radio model parameters. For example in Fig. 2.5a the radio model parameters are estimated to be  $\kappa = -51.99$  and  $\alpha = 0.5$ . These estimated parameters can then be used to predict the signal strength measurement at any range, which is shown by the red curve.

In Fig. 2.5a several measurements are normally distributed around the least squares fit. This is confirmed by the normplot shown in Fig. 2.6a, where the pink line is close to the black line but does deviate slightly, showing that the noise is not entirely AWGN. This is indicative of the relatively cleaner nature of the 433MHz communications. In contrast, in Example 2, although Fig. 2.5b shows that while a fair portion of the 2.4GHz measurements are clustered around the fitted curve, there are several outliers. These outliers are clearer in the normplot view shown in Fig. 2.6b, where both tails of the normplot are severely deviated from the straight black line.

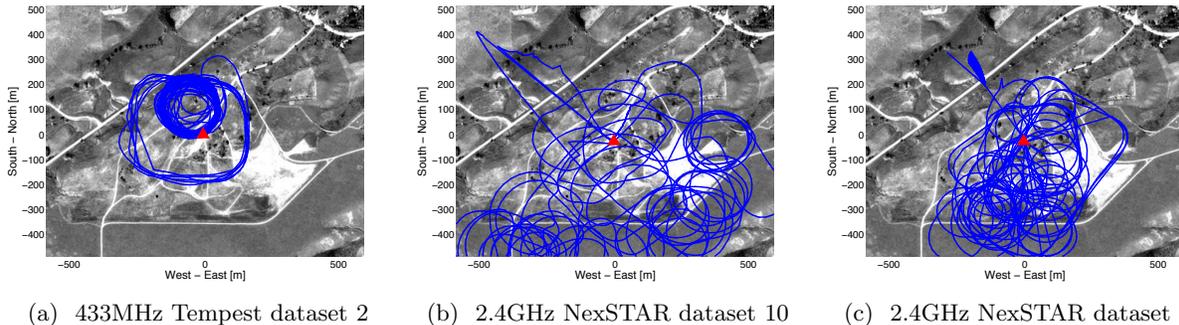


Figure 2.4: Maps of the flight experiment setups with transmitter locations and flight trajectories

The WiFi noise problem is exacerbated in Example 3, shown in Fig. 2.5c, where systematic large outliers between ranges of 300m and 500m cause the least squares solution to fail completely. This failure is further apparent in Fig. 2.6c where the pink line massively deviates from the solid black line. The presence of these outliers in a narrow band signals towards some local interference issues, which results in large variations not consistent with the path loss model. Together, examples

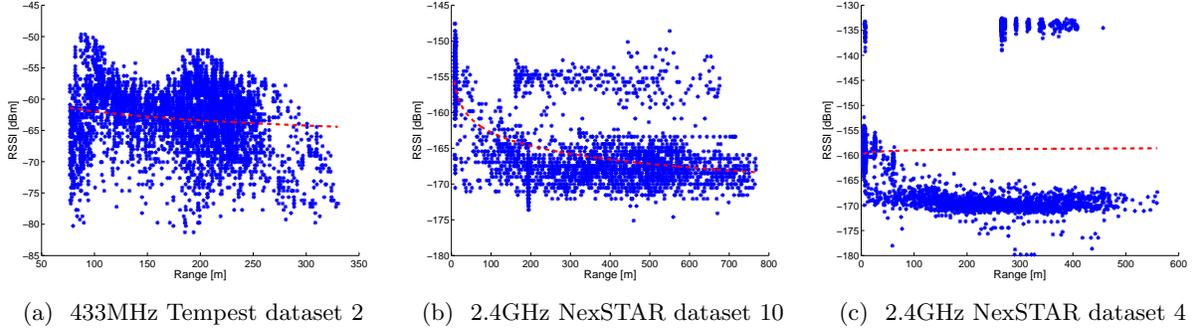


Figure 2.5: Estimation of radio model parameters from RSSI measurements using least squares

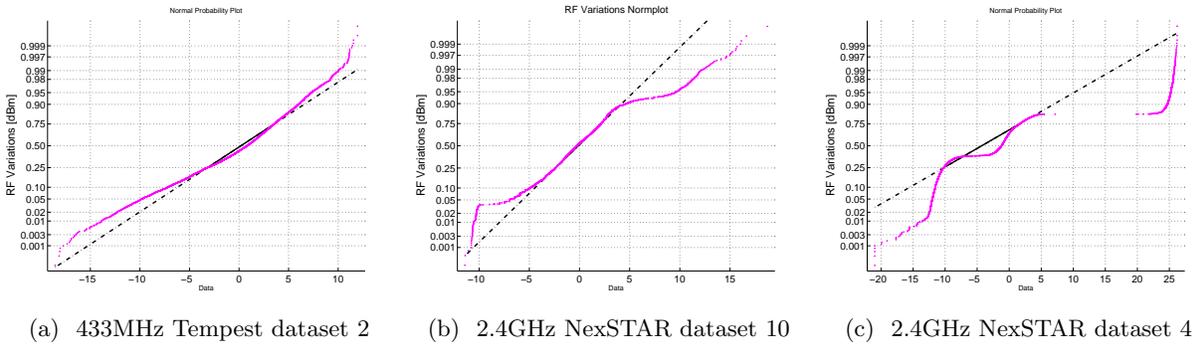


Figure 2.6: Normplot of RF variations or residuals of the *a priori* empirical path loss model

2 and 3 represent a widespread problem faced by communication-aware UAS missions using 2.4GHz communications. In the presence of such errors, UAS data ferrying, which plans UA trajectories through regions of good communication, will over or under estimate the actual environment, consequently achieving a worse throughput than planned. Similarly, these errors can result in filter divergence when performing radio source localization with the communication model as the sensing model, leading to large estimation errors. This motivates the need for refined, nonparametric radio models which learn the RF variations.

### 2.3 Evaluation Metrics

The rest of this thesis develops nonparametric models to improve over the *a priori*, empirical path loss model. Flight datasets are used to evaluate whether these models do in fact perform better than the *a priori* models. This quantitative evaluation compares the models by using metrics that

measure errors and the uncertainty. This section outlines the metrics that will be used in the rest of the thesis for presenting detailed performance comparison of models.

Each of these metrics assess a model's performance by comparing its prediction on a validation set against their known measurement. Each sample in the validation set consists of the input variable  $\rho_i$  and the corresponding response variable  $z_i$ . Thus, the validation set is given by  $\{(\rho_i, z_i)\}$  for  $i = 1, \dots, N_v$ . In this thesis,  $z$  represents the received signal strength, and  $\rho$  represents the receiver state, which is either its position  $\mathbf{p} = [x, y]^T$ , or its position and time  $[\mathbf{p}^T t_k]^T$ .

### 2.3.1 Root Mean Squared Error (RMSE)

The root mean squared error (RMSE) measures the accuracy of a model's predictions or estimates. RMSE of a model on the validation set is calculated as

$$\text{RMSE} = \sqrt{\frac{1}{N_v} \sum_{i=1}^{N_v} (z_i - \mu_i)^2} \quad (2.5)$$

where  $\mu_i$  is the model's estimate or mean prediction for the  $i^{\text{th}}$  sample. Note that the term inside the root is the sample variance of the errors, and hence RMSE is the sample standard deviation of the model's prediction errors.

### 2.3.2 Average Normalized Estimation Error (ANEES)

The RMS error does not capture the uncertainty predictions of fully probabilistic models because it ignores the variance. Average NEES (Normalized Estimation Error Squared) or the squared Mahalanobis Distance (MD) [4] helps assess the mean and variance of the prediction by comparing it with measured RF variation

$$\text{NEES} = \text{MD}^2 = \frac{(z_i - \mu_i)^2}{\sigma_i^2} \quad (2.6)$$

$$\text{ANEES} = \frac{1}{N_v} \sum_{i=1}^{N_v} \frac{(z_i - \mu_i)^2}{\sigma_i^2} \quad (2.7)$$

where  $\mu_i$  and  $\sigma_i^2$  are the model's mean and variance predictions for the  $i^{\text{th}}$  sample.

### 2.3.3 Mean Standardized Log Loss (MSLL)

A standard method for evaluating a model is to measure how it performs relative to a baseline model  $M_B$  in terms of mean standardized log loss (MSLL) [68]. It represents the disagreement between the model and measurements taken in the real world. Thus, it can be viewed as a loss function to be minimized, and hence is also called the log loss. The MSLL for model  $M$  is defined as the mean of the differences between negative log likelihoods of a given validation set.

$$MSLL_M = \frac{1}{N_v} \sum_{i=1}^{N_v} (-\log p(z_i|M, \rho_i) - [-\log p(z_i|M_B, \rho_i)]) \quad (2.8)$$

The negative log likelihood in this equation is defined as

$$\begin{aligned} -\log p(z_i|M, \rho_i) &= -\log \left[ \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(z_i - \mu_i)^2}{2\sigma_i^2}\right) \right] \\ &= \frac{1}{2} \log(2\pi\sigma_i^2) + \frac{(z_i - \mu_i)^2}{2\sigma_i^2} \end{aligned} \quad (2.9)$$

where  $\mu_i$  and  $\sigma_i^2$  denote the mean and variance of the predictive distribution for the  $i^{th}$  sample. Consequently, the MSLL will be approximately zero for models similar to the baseline, while better models have negative MSLL.

## 2.4 Summary

50 real flight datasets of signal strength measurements are used for evaluating airborne RF communication models. These datasets consist of 433MHz and 2.4GHz signal strength measurements from stationary and mobile transmitters, collected from three different small unmanned aircraft. Measuring the performance using root mean squared error (RMSE), Average normalized estimation error (ANEES), and mean standardized log loss (MSLL) evaluates both the predicted estimate and its uncertainty, and can be used to compare the accuracy of nonparametric solutions against that of the *a priori* empirical model.

## Chapter 3

### Spatio-temporal Models of Airborne RF Communication

Traditional radio models use physics-based parametric formulations [14, 47], which fail to account for RF variations and other artifacts that stem from unknown interference and environment-specific characteristics. Section 2.2 presented the path-loss radio model [66], which is an example of such a parametric communication model, and highlighted its shortcomings.

This dissertation augments such parametric models with a nonparametric spatio-temporal Gaussian process (GP) [67], and learns local, receiver-location dependent RF variations. Fig. 3.1 shows how the residuals of the empirical path-loss model are used for learning a GP of RF variations. In this manner the GP adapts an *a priori* model to learn the time-varying and geospatial characteristics specific to the environment in which the small unmanned aircraft is operating.

The experiments presented in this chapter use flight experiment data to show how spatio-temporal GPs reduce the error in communication models for small UAS. These techniques, however, can be used more broadly for learning other airborne communication networks.

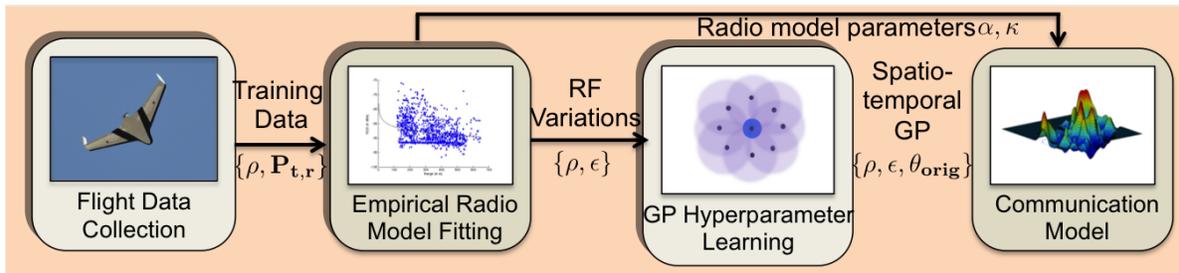


Figure 3.1: Overview block diagram of nonparametric spatio-temporal characterization of airborne RF environments

### 3.1 Nonparametric Model of Receiver-Dependent Variation

Several factors influence the noise in the signal measurement at a location. In general, radio noise comes from a variety of sources (e.g. fast fading, shadow fading, multipath, interference) that can be correlated with the transmitter state, receiver state, or their relative positions and orientations.

This work assumes that the noise has a component which is a function of the measurement location  $\mathbf{p}_r$  and time  $t_k$ . This dependence on receiver state  $\boldsymbol{\rho}_r = [\mathbf{p}_r^T, t_k]^T$  is expressed as

$$\begin{aligned} P_{t,r} &= \kappa - \alpha 10 \log_{10}(\|\mathbf{p}_t - \mathbf{p}_r\|) + v_r \\ &= \Xi_{t,r} + v_r \end{aligned} \quad (3.1)$$

where  $v_r \sim \mathcal{N}(\mu(\boldsymbol{\rho}_r), \sigma^2(\boldsymbol{\rho}_r))$  represents local RF variation at  $\mathbf{p}_r$  and  $t_k$ .

Specifically, this model for the RF variations can be learned using the same dataset of measurements,  $\mathbf{z} = [z_{i1}, z_{i2}, \dots, z_{iN}]^T$ , mentioned in Subsection 2.2. Because the variation is captured in a separate, standalone term, it can be isolated when measurements are taken from a friendly, known radio. The variation  $e_j$  at a receiver state  $\boldsymbol{\rho}_j$  is the error or residual of the empirical radio model, and is calculated as

$$e_j = z_{i,j} - \Xi_{i,j} = v_j \quad j = 1, \dots, N \quad (3.2)$$

i.e the difference between the signal measurement  $z_{i,j}$  and the empirical radio model estimate  $\Xi_i(\mathbf{p}_j)$  (Equation 2.3). This set of isolated, location- and time-tagged RF variations can now be used to learn  $v_j \forall j$  i.e. Gaussian distribution predictions for RF variations throughout the environment.

Our approach makes use of a nonparametric model to describe the environment variations [87]. Specifically, the  $N$  samples of RF variations computed from Equation (3.2), serve as the training set  $\{\boldsymbol{\rho}_j, e_j\}_{j=1:N}$  to train a Gaussian process.

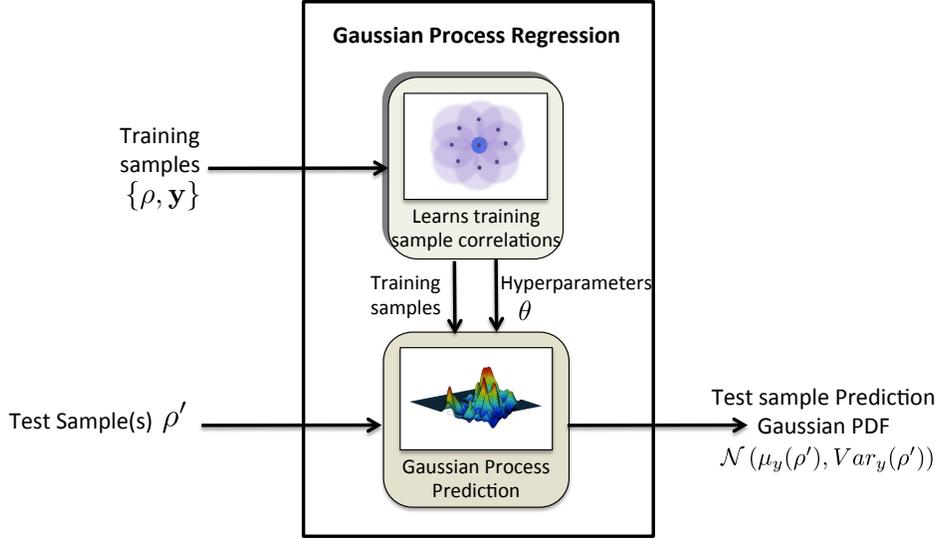


Figure 3.2: Overview of Gaussian process learning and prediction.

### 3.1.1 Spatial and Spatio-temporal GP

The details of the Gaussian process framework can be found in [68, 74, 4]. The overview of the GP framework is illustrated in a block diagram in Fig. 3.2, and the main points are summarized in this section.

Once the GP is trained, the learned model can be used to predict the RF variation  $e'$  at unseen state  $\rho'$ . This prediction is in the form of a probability density function (PDF), which comprises of the expected value (mean) and the variance,

$$(\mu_e(\rho'), \sigma_e^2(\rho')) = \mathcal{GP}(\rho' | \{\rho_{1:N}, \mathbf{e}_{1:N}, \theta\}) \quad (3.3)$$

where  $\theta$  are the hyperparameters of the Gaussian process. This PDF is obtained by calculating the joint distribution of the unseen state with the states of the training samples, as follows

$$\begin{aligned} \mu_e(\rho') &= k(\rho, \rho')^T (K + \sigma_n^2 I)^{-1} \mathbf{e} \\ \sigma_e^2(\rho') &= k(\rho', \rho') - k(\rho, \rho')^T (K + \sigma_n^2 I)^{-1} k(\rho, \rho') \end{aligned} \quad (3.4)$$

Here  $k(\rho, \rho')$  is the  $N \times 1$  vector of correlations of the new state with all the training points' states,  $\mathbf{e}$  is the  $N \times 1$  vector of measured variations at the training points,  $K$  is an  $N \times N$  kernel matrix with entries  $k_{ij} = k(\rho_i, \rho_j)$ , and  $\sigma_n^2$  is assumed to be the noise variance of the original process.

In the rest of the thesis, the short form  $\mathcal{GP}'$  is used to represent prediction based on the learned stochastic model, as shown in (3.3).

The GP computes correlations using a kernel function  $k(\boldsymbol{\rho}, \boldsymbol{\rho}')$ . The choice of the correlation function or kernel is a key design decision when using a GP. Due to its infinite differentiability, a squared exponential or Gaussian kernel is popularly used. It is defined as

$$k(\boldsymbol{\rho}, \boldsymbol{\rho}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\rho} - \boldsymbol{\rho}')^T L^{-1}(\boldsymbol{\rho} - \boldsymbol{\rho}')\right) \quad (3.5)$$

where  $\sigma_f^2$  is the signal variance, and  $L$  is a  $D \times D$  diagonal matrix when  $\boldsymbol{\rho} \in \mathbb{R}^D$ . The diagonal elements are given by  $L(d, d) = l_d^2$ , where  $l_d$  is the lengthscale of dimension  $d$ , where  $d = 1, \dots, D$ .

The nature of the correlations and the number of lengthscales differ depending on what dimensions are included in the state, and whether they are given equal importance or not. For example, in a spatial GP where  $\boldsymbol{\rho} = \mathbf{p} = [x, y]^T$ , if the X and Y dimensions of the position are given equal importance, it results in a *spatial isotropic* GP, with the following kernel

$$\begin{aligned} k_{sIso}(\boldsymbol{\rho}, \boldsymbol{\rho}') &= \sigma_f^2 \exp\left(-\frac{1}{2} \frac{(x - x')^2 + (y - y')^2}{l_s^2}\right) \\ &= \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\rho} - \boldsymbol{\rho}')^T L^{-1}(\boldsymbol{\rho} - \boldsymbol{\rho}')\right) \quad \text{where } L = l_s I_2 \end{aligned}$$

On the other hand, we can use a different lengthscale for each dimension, to determine their importance or relevance to the problem being modeled. This method when used with a spatial GP, is known as *spatial automatic relevance determination (ARD)*. The following kernel function allows for non-isotropy along the X and Y dimensions.

$$\begin{aligned} k_{sARD}(\boldsymbol{\rho}, \boldsymbol{\rho}') &= \sigma_f^2 \exp\left(-\frac{1}{2} \left[ \frac{(x - x')^2}{l_x^2} + \frac{(y - y')^2}{l_y^2} \right]\right) \\ &= \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\rho} - \boldsymbol{\rho}')^T L^{-1}(\boldsymbol{\rho} - \boldsymbol{\rho}')\right) \quad \text{where } L = \begin{bmatrix} l_x^2 & 0 \\ 0 & l_y^2 \end{bmatrix} \end{aligned}$$

Cross-diagonal terms can also be included to define a fully anisotropic kernel function which can directional effects present in the environment. Finally, if ARD is used with a spatio-temporal GP,

the kernel function has the following form

$$\begin{aligned}
 k_{stARD}(\boldsymbol{\rho}, \boldsymbol{\rho}') &= \sigma_f^2 \exp\left(-\frac{1}{2} \frac{(x-x')^2}{l_x^2} + \frac{(y-y')^2}{l_y^2} + \frac{(t-t')^2}{l_t^2}\right) \\
 &= \sigma_f^2 \exp\left(-\frac{1}{2} (\boldsymbol{\rho} - \boldsymbol{\rho}')^T L^{-1} (\boldsymbol{\rho} - \boldsymbol{\rho}')\right) \quad \text{where } L = \begin{bmatrix} l_x^2 & 0 & 0 \\ 0 & l_y^2 & 0 \\ 0 & 0 & l_t^2 \end{bmatrix}
 \end{aligned}$$

Since the GP uses the difference between timestamps to compute temporal correlation, the timestamp  $t$  can be a GPS timestamp, or times since beginning of flight.

Note that in the case of the mobile transmitter, because the transmitter location is also included, the spatial ARD GP will have 4 lengthscales:  $l_{xr}^2$ ,  $l_{yr}^2$ ,  $l_{xt}^2$ , and  $l_{yt}^2$ . Similarly, the spatio-temporal ARD GP for a mobile transmitter will have these 4 lengthscales as well as  $l_t^2$ .

Training a Gaussian process involves learning  $\sigma_n$ ,  $\sigma_f$ , and  $l_d$  which are the *hyperparameters*  $\theta$  of the model. The hyperparameters are derived by maximizing the log likelihood function of the Gaussian process for the  $n$  sample points in the training data set

$$\begin{aligned}
 \theta &= \arg \min_{\theta} \ln p(\mathbf{e}|\theta) \\
 &= \arg \min_{\theta} \left[ -\frac{1}{2} \mathbf{e}^T C_N^{-1} \mathbf{e} - \frac{1}{2} \ln |C_N| - \frac{N}{2} \ln 2\pi \right]
 \end{aligned} \tag{3.6}$$

where  $C_N(\theta) = K(\{\boldsymbol{\rho}\}_{1:n}; \sigma_f, L) + \sigma_n^2 I$ .

### 3.1.2 GP based modeling of RF Communication

Prior work has shown that shadow fading effects have a log normal distribution in the non-dB domain, which results in a Gaussian distribution in the dB domain [54]. Their correlations have been modeled in ground robots using an exponential autocorrelation function [54, 24, 22]. Since these fading effects are responsible for the geospatial RF variations, and the exponential autocorrelation function maps to a GP with a squared exponential or Gaussian kernel, we use a GP to model these receiver dependent variations.

Fig. 3.3 shows a block diagram of our approach for modeling communication by learning RF variations. As explained in 2.2, RF signal measurements are used to learn radio model parameters

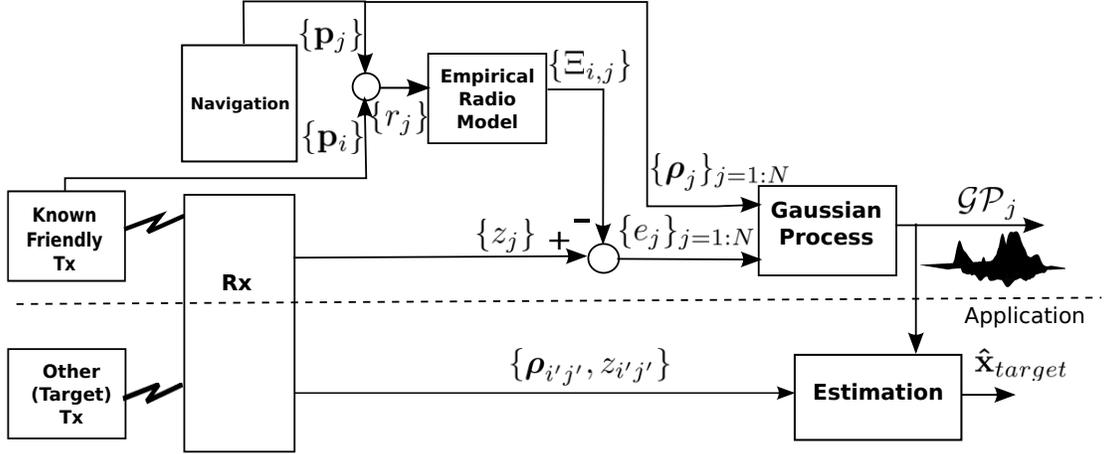


Figure 3.3: RF variation modeling block diagram

of an empirical path loss model. To make the system robust, in our experiments, the least squares solution is estimated after removing the outliers. Outliers are samples which lie outside the inner fences, given by  $Q1 - 1.5IQR$  and  $Q3 + 1.5IQR$ , where  $Q1$  and  $Q3$  are the 25<sup>th</sup> and 75<sup>th</sup> percentiles, and  $IQR = Q3 - Q1$  is the interquartile range [85]. This inexpensive pre-processing step results in improved path loss radio models, whose residuals are more tuned to their physical significance.

These residuals of the *a priori* path loss model, or receiver dependent RF variations  $e$ , and their corresponding receiver state  $\rho$  serve as the input and response variables of the Gaussian process. Correlations in the training data are learned using spatial isotropic, spatial ARD, or spatio-temporal ARD kernels presented in Section 3.1.1. All three of these kernels are Gaussian, learn hyperparameters using MLE, and provide a Gaussian PDF prediction. Yet, they help to capture different and increasingly complex environmental characteristics. In the case of stationary transmitters the input variable represents the state of only the receiver. When the GP is used to capture only spatial correlations,  $\rho_j = \mathbf{p}_j$ , whereas for a spatio-temporal GP,  $\rho_j = [\mathbf{p}_j^T, t_{kj}]^T$ . In the case of mobile transmitters, the states in the spatial and spatio-temporal GPs are given by  $[\mathbf{p}_j^T, \mathbf{p}_i^T]^T$  and  $[\mathbf{p}_j^T, \mathbf{p}_i^T, t_{kj}]^T$ , respectively. If other aspects of the receiver are considered relevant to the variations, those can also be included in the receiver state, and consequently in GP.

This ability to model different radio environments using a single methodology exhibits the

true modeling flexibility of a GP. In this manner the nonparametric nature of the GP can capture a wide range of diverse RF behaviors. There are a few more advantages to using a GP to model the RF variations. Because a GP is a continuous model, it does not limit the states at which it can be used for prediction, consequently eliminating issues that stem from quantizations. On the other hand, it does indicate lack of confidence i.e. uncertainty in the prediction via a large value of variance. Finally, because its prediction is in the form of a Gaussian PDF, a GP can provide a concise yet fully probabilistic framework.

Because the GP only learns the RF variations in the environment, to obtain a prediction for signal strength received at  $\mathbf{p}_j$  from transmitter  $\mathbf{p}_i$ , we have

$$\mathcal{M}_{i,j} = \Xi_{i,j} + \mathcal{GP}_j \quad (3.7)$$

where the GP's mean prediction is added to the prediction of the empirical path loss model.

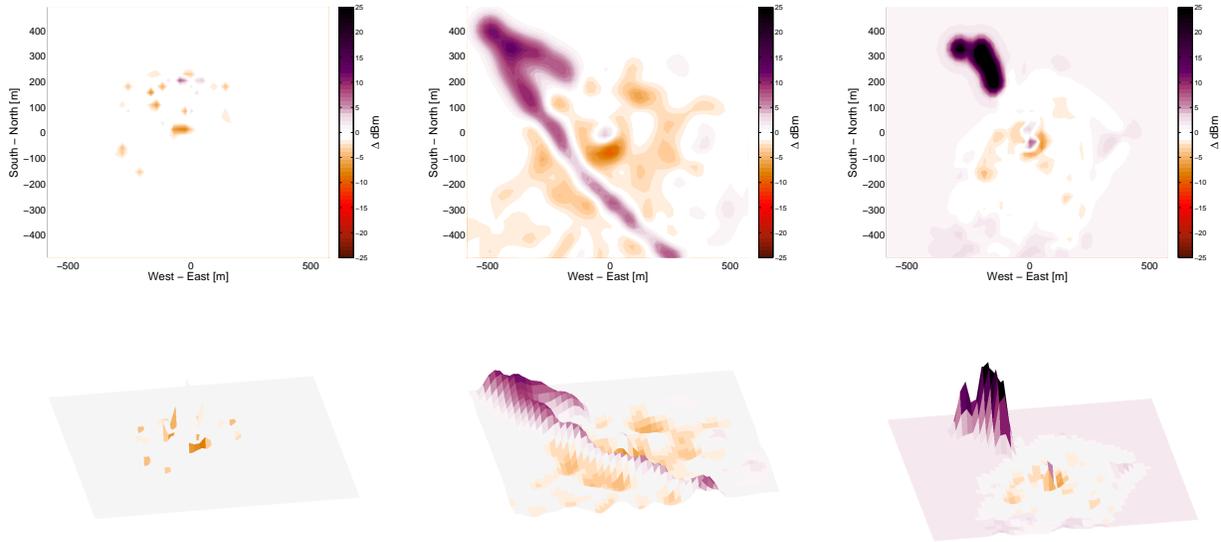
## 3.2 Flight Data Validation Results

This section evaluates the RF characterization obtained by modeling RF variations using a Gaussian process.

### 3.2.1 Example Runs

The same three example datasets presented in Section 2.2.1 are used here to explain how Gaussian processes learn RF variations. The first two are examples of 433MHz and 2.4GHz datasets, respectively, where the GP learns the RF variations and reduces the error in the radio model. The third example illustrates how the GP identifies a region of local interference, considerably improving the prediction accuracy compared to the *a priori*, empirical path loss model.

Once outlier detection is completed, and radio model parameters for the path loss model are estimated from location and time tagged measurement dataset, the residuals give the RF variations for that dataset. The Gaussian process models are trained on RF variation datasets, as explained in Section 3.1.1. The trained GP then provides mean and variance predictions for the RF variations



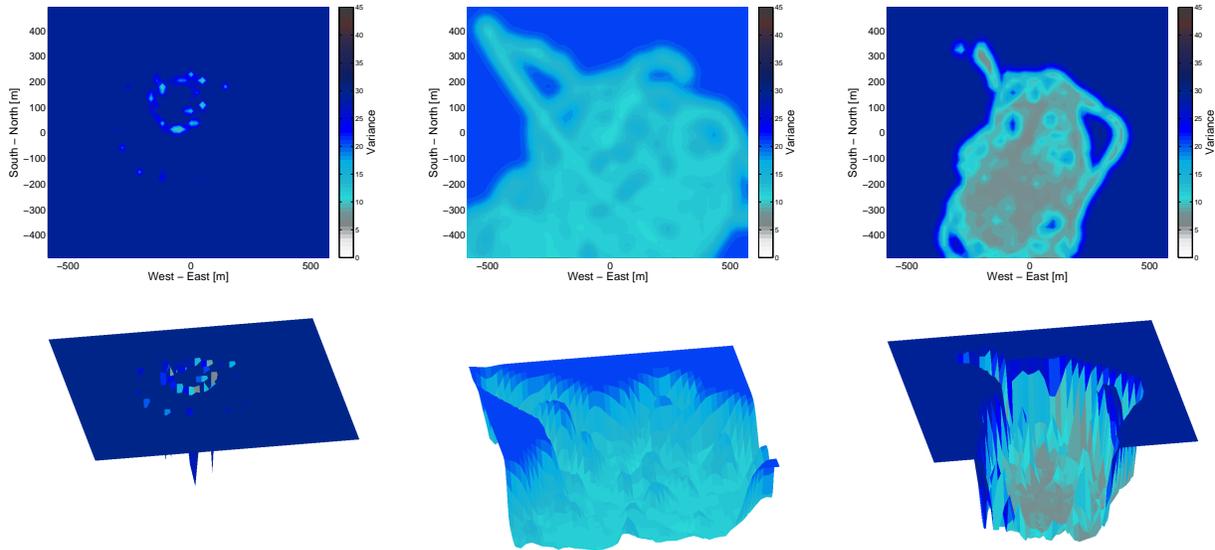
(a) 433MHz Tempest dataset 2      (b) 2.4GHz NexSTAR dataset 10      (c) 2.4GHz NexSTAR dataset 4

Figure 3.4: Mean prediction of the spatial isotropic GP of RF variations

throughout the field, as shown in Figs. 3.4 and 3.5 respectively. Both figures show 2D (top row) and 3D (bottom row) views for predictions obtained from a spatial isotropic GP. Note that the GP mean predictions in Fig. 3.4 are predictions of the RF variations. Thus, they are corrections to be applied to the empirical radio model estimates. Purple and red represent regions where the empirical radio model has underestimated and overestimated the signal strength, respectively. Regions where the GP predicts that the truth does not deviate from the path loss model are shown in white.

The variances, shown in Fig. 3.5, represent the uncertainty in the mean predictions. Higher variance corresponds to higher uncertainty and is depicted by the darker colors. The lighter colors represent lower variances i.e. lower uncertainty. The variance at a location is dependent on the location's proximity to samples in the training set, the overall signal variability seen in the training set (represented by the hyperparameter  $\sigma_f^2$ ), as well as the noise variance  $\sigma_n^2$ .

The mean and variance plots for all examples have the same colorbar, for a side-by-side visual comparison. Once again, the cleaner 433MHz example contains very sparse deviations, as seen from



(a) 433MHz Tempest dataset 2      (b) 2.4GHz NexSTAR dataset 10      (c) 2.4GHz NexSTAR dataset 4

Figure 3.5: Variance prediction of the spatial isotropic GP of RF variations

Fig. 3.4a. Because the UA flies the same two paths repeatedly throughout this mission (Fig. 2.4a), most of the field is unexplored. Consequently, the variance in most of these unexplored regions, as seen in Fig. 3.5a, is relatively high to signify the possibility that other small pockets of high signal variability may exist. Adding exploration into the trajectory can reduce this high uncertainty, as seen in Fig. 3.5b for example 2, where the variance in most of the region has reduced, with only the corner regions at a higher value. The mean prediction for Example 2 in Fig. 3.4b identifies regions where moderate RF variations exist. However, for the most part, these corrections are locally clustered and small in magnitude.

In contrast to the first 2 examples, in Fig. 3.4c, Example 3 identifies a region (shown in purple) in the north-west corner of the field where the empirical radio model has grossly underestimated the signal strength. This exemplifies the GP’s ability to start for an **a priori** model, and learn and adapt to significant environmental artifacts like localized interference. Also, the fully probabilistic nature of the GP’s predictions warn of the presence of such artifacts with high variances outside the explored regions, as seen in Fig. 3.5c.

Fig. 3.6 highlights how the least squares solution used to learn the path loss model can fail in the presence of outliers arising from local interference, as in Example 3. Since RF variations are residuals of the empirical radio model, these failures impact the Gaussian process regression as well. This is seen in the GP's mean prediction (shown in second row of Fig. 3.6a), which requires large negative correction for most of the region.

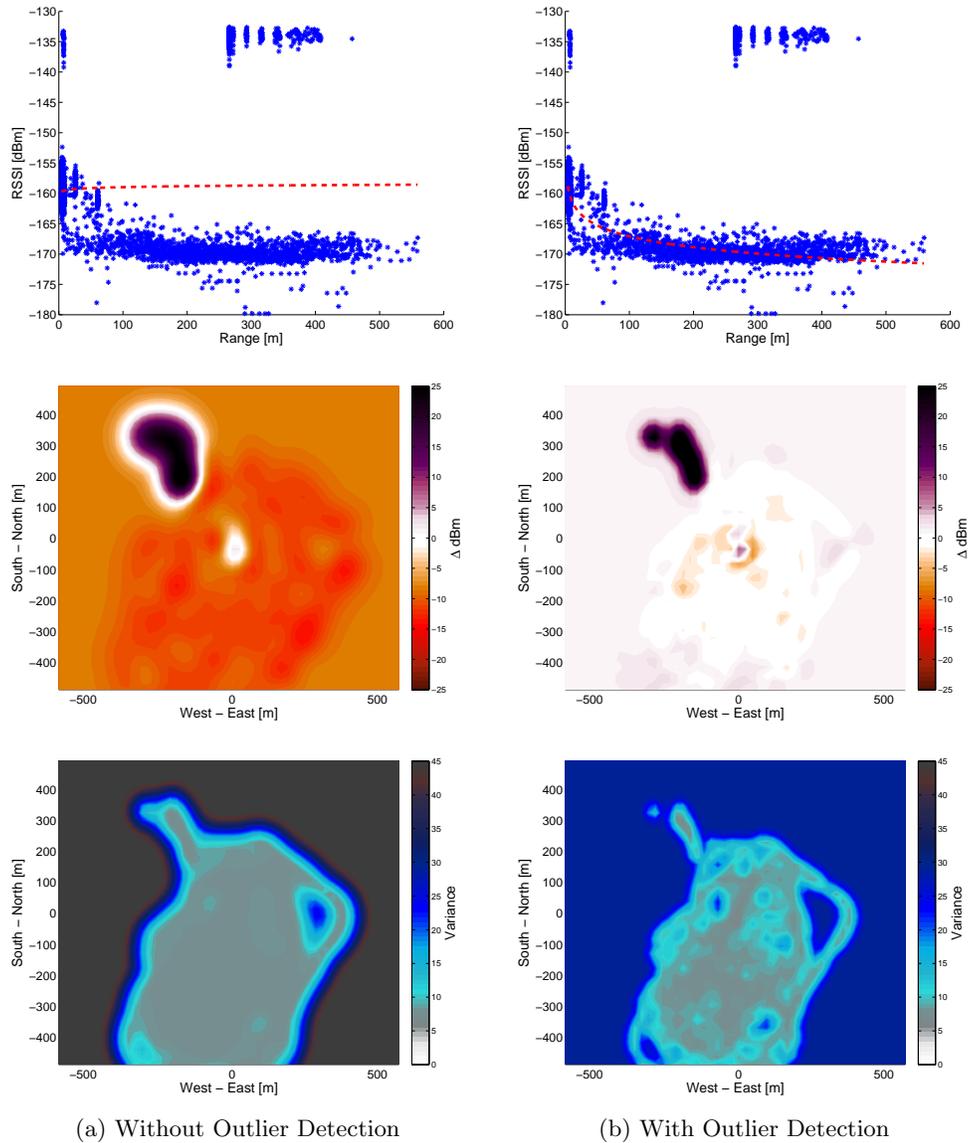


Figure 3.6: Comparison of path loss models and GP mean and variance predictions for 2.4GHz NexSTAR dataset 4 with and without outlier detection

Fig. 3.6b shows that removing the outliers results in low RF variations in most of the region other than top left corner. This makes it easier to identify local artifacts of the region. In addition, smaller residuals reduce the variability in the GP and lead to higher confidence (i.e. lower variance predictions) for the entire region. In contrast to the 2.4GHz dataset 4, the other two examples remain unchanged with outlier detection. Consequently, outlier detection is an inexpensive pre-processing step which can improve the baseline empirical communication models in the presence of some local variability.

### 3.2.2 Gaussian Process Cross Validation

The three examples presented in Section 3.2.1 differ in their communication frequencies, UA trajectory, and magnitude and nature of RF variations. The GP provides an elegant methodology for adapting the empirical radio model to their specific environments and challenges. This section quantitatively examines how the GP models i.e  $\mathcal{M} = \Xi + \mathcal{GP}$  improve over the empirical radio model  $\Xi$ .

The performance of the GP radio models is evaluated using 5-fold cross validation, with root mean squared error (RMSE) and mean standardized log loss (MSLL) as the metrics. 5-fold cross validation divides the dataset into 5 equal-sized, randomly selected groups. In each run, one group is used as validation data for testing the performance of a GP trained on all the data from the 4 other groups. This process is repeated 5 times, leaving out a different group as the validation set each time. The final reported metric is the average of the validation RMSE and MSLL over the 5 runs.

The K-fold cross validation methodology provides a generalized assessment of model performance, and we use it as a platform for comparing different types of GPs. Specifically, we train and assess spatial isotropic, spatial ARD, and spatio-temporal ARD Gaussian processes for each of the datasets in Tables 2.1, 2.2, and 2.3. The RMSE and MSLL of these GPs are also compared against the empirical radio model from Section 2.2. Finally, each of these models (i.e. empirical or GP radio models) are learned with and without outlier detection. These detailed, per-dataset,

comparative results are shown in the Appendix A. Note that spatio-temporal ARD models could not be constructed for the 433MHz datasets because accurate timestamps were unavailable.

### 3.2.2.1 Root Mean Squared Error (RMSE)

The RMSE results for each dataset in the 3 groups, presented in Tables A.1-A.3 in the appendix, are summarized here in Table 3.1 by averaging over all datasets within each group. In all 3 groups, the GPs improve over the empirical radio model. Also, where available, the spatio-temporal GP does better (i.e. has lower RMS error) than its spatial isotropic and ARD counterparts.

Table 3.1: Model comparison using cross validation Root Mean Squared Error (RMSE) averaged over datasets in each group

Outlier detection	Model	433 MHz Tempest	2.4 GHz NexSTAR	Mobile Skywalker X8
Without	Empirical radio model	4.07	2.95	3.63
	Spatial Isotropic GP	3.45	1.97	3.04
	Spatial ARD GP	3.44	1.98	2.98
	Spatio-temporal ARD GP	–	1.58	2.78
With	Empirical radio model	4.05	3.20	3.63
	Spatial Isotropic GP	3.46	1.98	3.04
	Spatial ARD GP	3.44	1.99	2.98
	Spatio-temporal ARD GP	–	1.59	2.78

However, these averages presented in Table 3.1 do not reflect the large variability seen in the RMSE across datasets of any group. These are better illustrated using violin plots [32] shown in Fig. 3.7. Each violin represents a different type of radio model, marked on the Y-axis. For comparison, the RMSE of radio models constructed without and with outlier detection are shown side by side in orange and yellow respectively. The median and mean are shown by black diamond and black circle within, respectively. The shape of a violin captures the distribution of the RMS errors of datasets in a group. Said differently, the mass of the violin is concentrated around the RMSE values obtained in that group.

In all three groups of datasets, we see that the RMSE decreases from left to right i.e. both

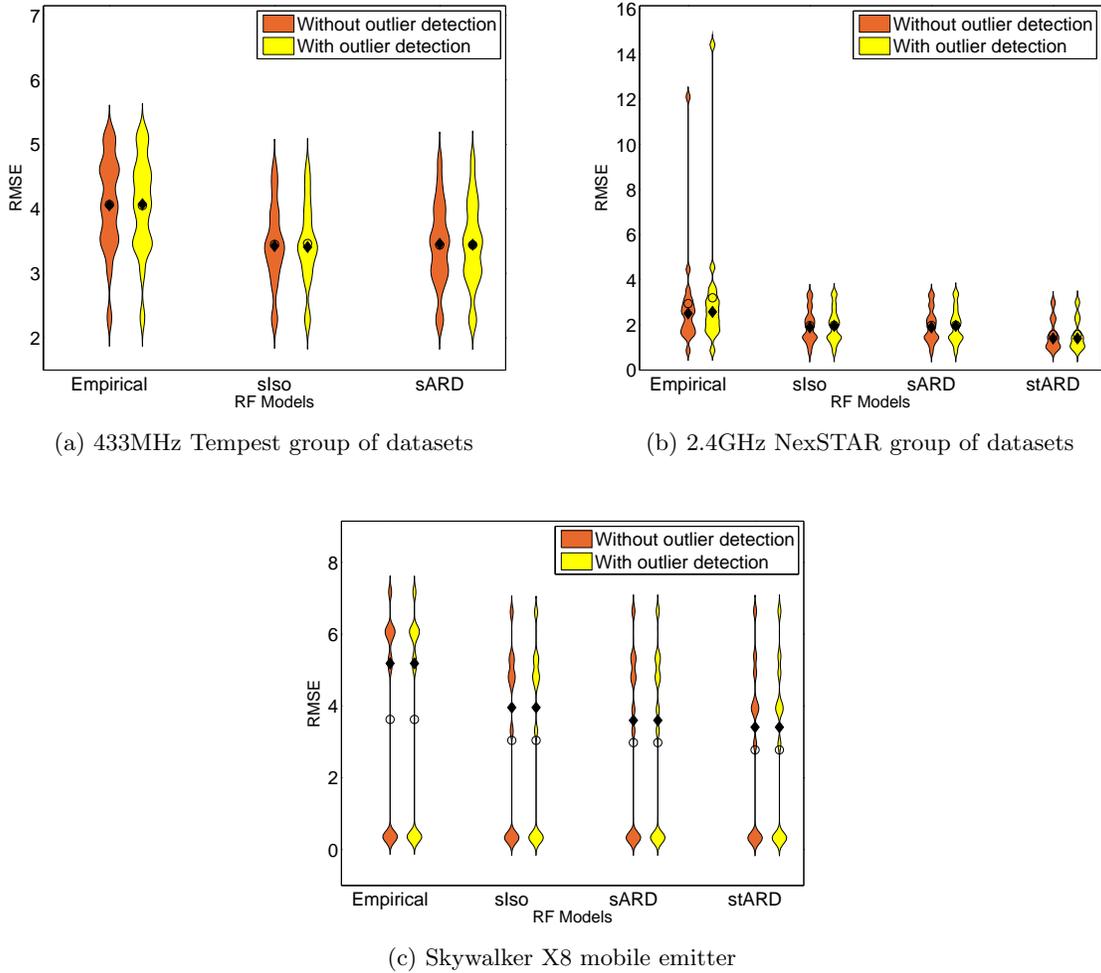


Figure 3.7: GP RMSE performance on groups of datasets collected using stationary transmitters

the median and mean of the RMSE for the GP models are lower than the empirical radio model. The improvement is especially prominent in Fig. 3.7b where the empirical radio model does much worse on some datasets than the corresponding GPs. Even when the extremes of two violin plots are similar like in the case of the GP models, the distribution of the mass shows that the spatio-temporal GP model improves over the spatial GPs.

Also, the addition of outlier detection in preprocessing does not change performance of any radio models. However, as explained in Section 3.2.1, the resulting models have better physical interpretability, and in the case of the GP have higher confidence in their mean predictions.

Because the 433MHz measurements are fairly clean, with mostly AWGN, the improvement

in this flight group is muted. However, in the case of 2.4GHz data, where the measurements are very noisy, the Gaussian process models, especially the spatio-temporal ARD GP, help prevent large errors seen due to environmental artifacts. This is clearly seen by comparing Figs. 3.8 and 3.9, which shows the error histograms for Example 2 from Section 3.2.1. Each of the histograms

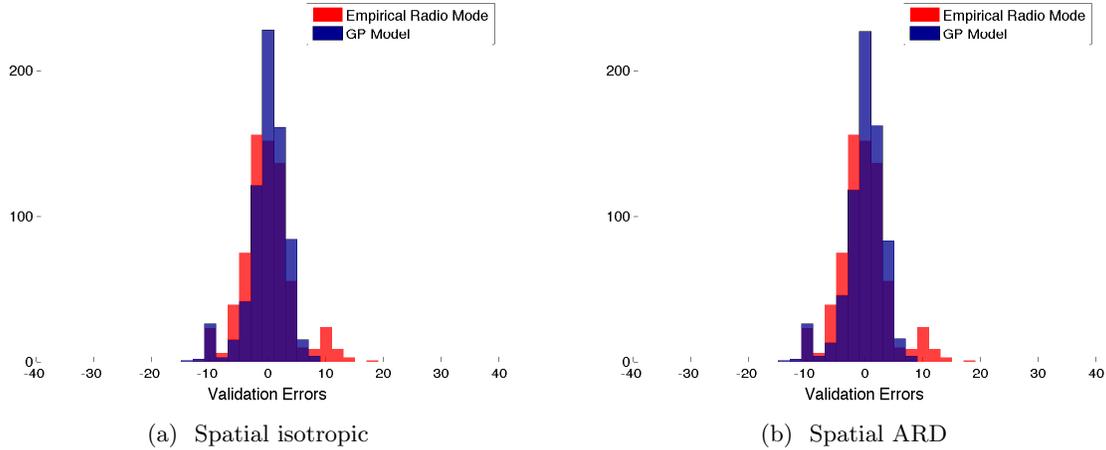


Figure 3.8: Error comparison between empirical radio model and spatial GPs of the RF variations for 2.4GHz NexSTAR dataset 10

compare the validation set errors of a GP model (shown in blue) with those of the empirical radio model (shown in red), with the purple region representing their overlap. While the errors from the spatial isotropic and ARD GPs collect around zero, they are still have a fair amount of overlap

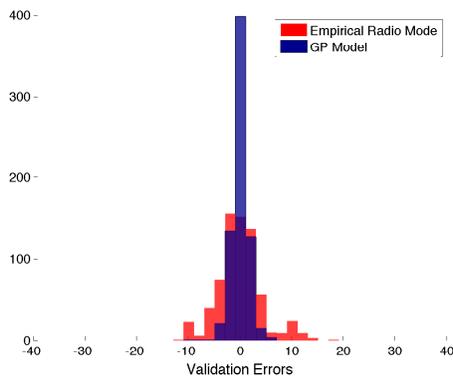


Figure 3.9: Error comparison between empirical radio model and spatio-temporal GP of the RF variations for 2.4GHz NexSTAR dataset 10

with the errors of the empirical model. In contrast, Fig. 3.9 shows that the spatio-temporal ARD GP reduces most of the errors to be below 5, with over 400 of them having a magnitude of less than or equal to 1.

It is important to note that the performance of the spatial ARD GP is limited because it is restricted to a diagonal lengthscale matrix  $L$ . A richer model can be obtained by learning an  $L$  matrix with off-diagonal terms to capture the cross correlations between the  $x$  and  $y$  dimensions, allowing for a completely anisotropic model. However, these additional terms will increase the number of hyperparameters, consequently increasing the training time.

### 3.2.2.2 Mean Standardized Log Loss (MSLL)

As explained in Section 2.3.3, MSLL provides a method for comparing models using their predictive distributions given by  $\mathcal{N}(\mu_i, \sigma_i^2)$  for response variable  $z_i$ . This is done by evaluating the log loss which measures the disagreement between the probability density function predicted by the model and the measurements. For the RF characterization problem, the MSLL is calculated for a dataset of receiver locations  $\rho_i$  and the corresponding signal measurements  $z_i$  for  $i = 1, \dots, N$ . The baseline model is the empirical radio model  $\Xi$ , which is assumed to have the training points made available to the GP models i.e  $\mathcal{M} = \Xi + \mathcal{GP}$ . The baseline’s predictive variance is equal to the sample variance of its residuals, given by  $\hat{\sigma}^2$ . Table 3.2 summarizes the mean and variance of the predictive distributions of these models.

Table 3.2: Model-wise Predictive Distributions for  $z_i$

Model	$\mu_i$	$\sigma_i^2$
Baseline	$\Xi_i$	$\hat{\sigma}^2$
$\mathcal{M}$ using Spatial Isotropic GP	$\Xi_i + \mu_{e,sIso}(\rho_i)$	$\sigma_{e,sIso}(\rho_i)^2$
$\mathcal{M}$ using Spatial ARD GP	$\Xi_i + \mu_{e,sARD}(\rho_i)$	$\sigma_{e,sARD}(\rho_i)^2$
$\mathcal{M}$ using Spatio-temporal ARD GP	$\Xi_i + \mu_{e,stARD}(\rho_i)$	$\sigma_{e,stARD}(\rho_i)^2$

Tables A.4-A.6 in the appendix show the MSLL for each of the datasets in the 3 groups as a mean and standard deviation across 5 cross validation runs. The MSLL of the 3 GP models are summarized in Table 3.3, and visually displayed using violin plots in Fig. 3.10.

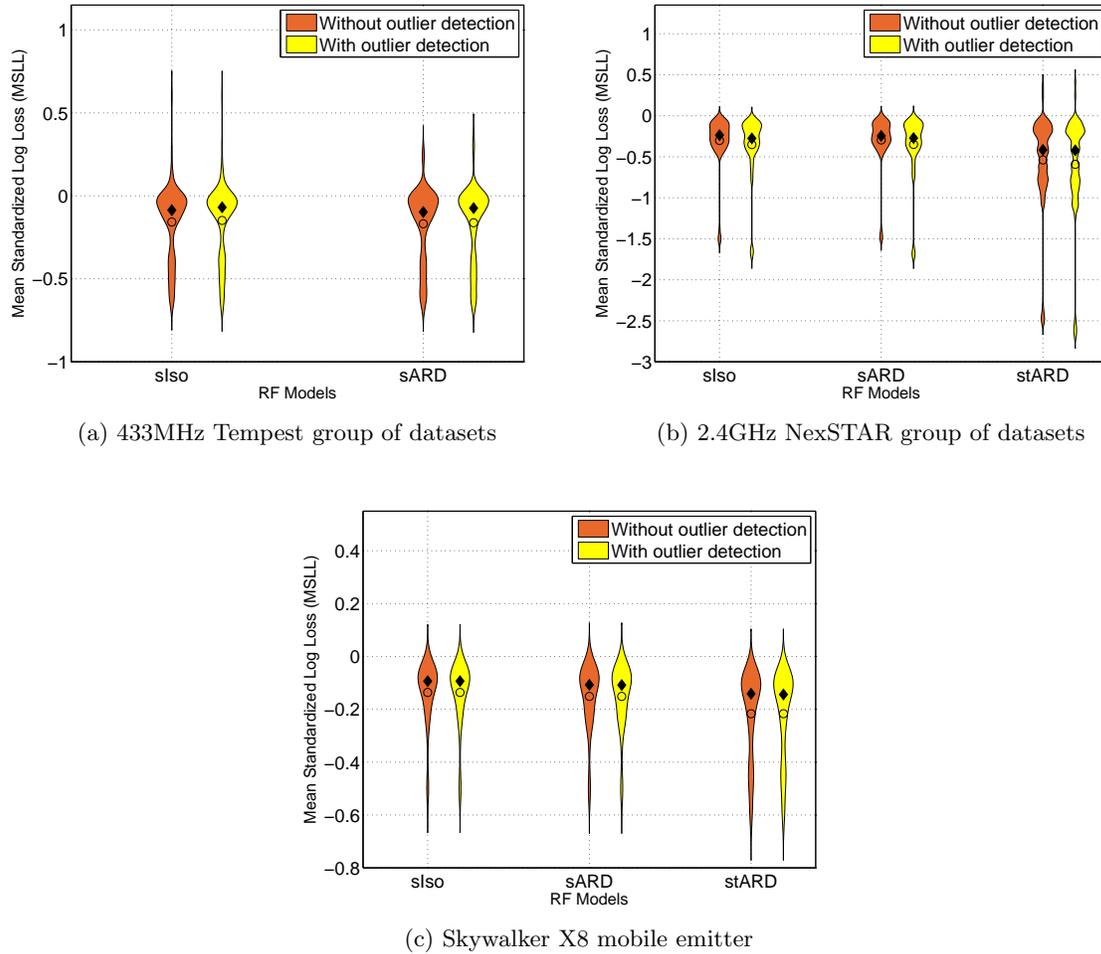


Figure 3.10: GP MSLL on groups of datasets collected using stationary transmitters

Table 3.3: Model comparison using cross validation Mean Standardized Log Loss (MSLL) averaged over the datasets in each group

Outlier detection	Model	433 MHz Tempest	2.4 GHz NexSTAR	Mobile Skywalker X8
Without	Spatial Isotropic GP	-0.16	-0.30	-0.14
	Spatial ARD GP	-0.17	-0.30	-0.15
	Spatio-temporal ARD GP	–	-0.54	-0.22
With	Spatial Isotropic GP	-0.15	-0.35	-0.14
	Spatial ARD GP	-0.16	-0.35	-0.15
	Spatio-temporal ARD GP	–	-0.59	-0.22

More negative MSLL indicates better models while close to zero indicates that the model is comparable to the baseline. As noted before, because 433MHz is a cleaner frequency, the im-

provement seen by using a GP in this group is small. This is confirmed by the MSLL values in Fig. 3.10a, which are concentrated just below zero with means between  $-0.15$  and  $-0.16$  for the different models. However, in the case of 2.4GHz, shown in Fig. 3.10b, the spatial models are at  $-0.30$  and  $-0.35$  without and with outlier detection. This number is nearly doubled in the case of the corresponding spatio-temporal models. Thus, once again, we see that the spatio-temporal models (where available) perform better than the spatial counterparts.

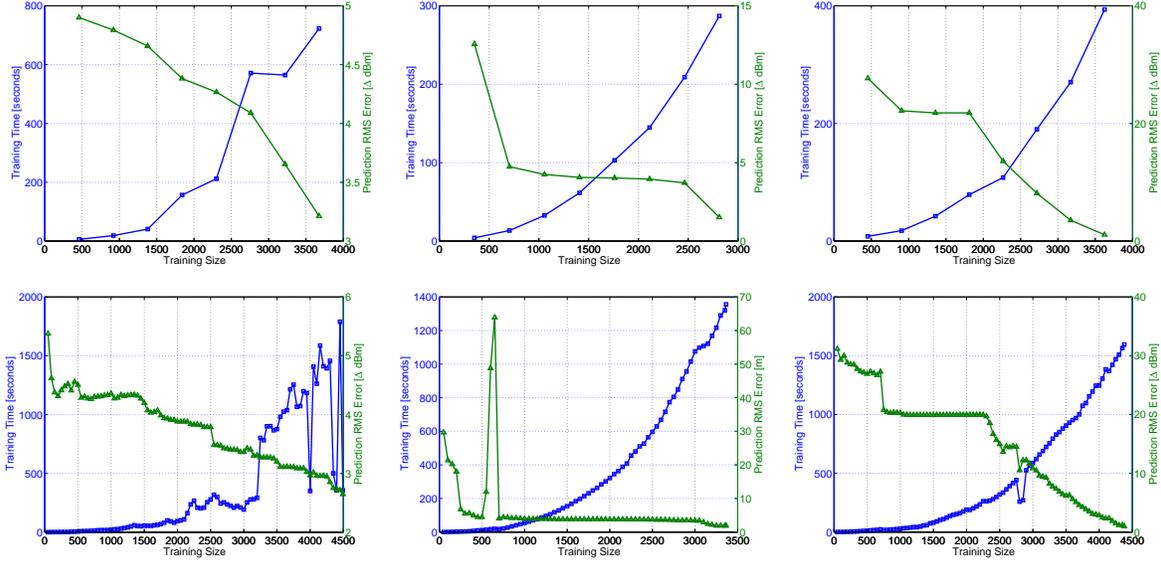
Also of note are the MSLL values for 2.4GHz NexSTAR dataset 4, where even the spatial GP models have a much more negative MSLL of around  $-1.47$ . Performing outlier detection and including the temporal dimension in the GP further reduces this to around  $-2.60$ . These values indicate that learning a GP in this example drastically improves the performance because it identifies and accounts for an environment specific artifact missed by the *a priori* empirical path loss model.

### 3.2.3 Computation vs Accuracy Trade-off in GPs

A GP's prediction accuracy is critically dependent on the size of the training set. As the size of the training set grows, so does the diversity, improving the GP's ability to predict across the field. Unfortunately, an increase in the training set also drastically increases the training time, primarily due to increased cost of inverting a larger covariance matrix during every iteration of the MLE optimization. These conflicting characteristics of a GP introduce a trade-off between accuracy and training cost.

To examine this, the GP was trained iteratively, with more training samples in each iteration. The training time and the corresponding prediction RMSE were compared for increasing sizes of the training set to illustrate the nature of this trade-off. Two versions of this experiment were carried out. The first one performed percentage-wise iteration, where 20% of the data was set aside as validation set, and the training set was grown from 10% to 80%. In the second version of the experiment 150 random samples made up the validation set, while the remaining samples were added to the training set in fixed increments of 50.

The detailed results of these two experiments are presented in Appendix A. Fig. 3.11 shows the results for the three examples from Section 3.2.1, which can be used for understanding the implications of this trade-off. In both cases, percentage-wise and fixed-increment iteration, the



(a) Spatial ARD for 433MHz Tempest dataset 2 (b) Spatio-temporal ARD for 2.4GHz NexSTAR dataset 10 (c) Spatio-temporal ARD for 2.4GHz NexSTAR dataset 4

Figure 3.11: Training time vs prediction RMSE trade-off for increasing training set sizes obtained by iteratively training GPs percentage-wise (top row) and using fixed increment (bottom row).

training time or cost rises almost exponentially. In contrast, the RMSE drops linearly, plateauing for some iterations in both experiments. These plateaus correspond to iterations where newly added training samples are similar to the existing training data, and do not contribute any new information to reduce the prediction error. However, these additional samples do add to the training costs. This illustrates not only the need for exploration of unseen regions, but also the diminishing returns property of localized samples.

Note that these experiments quantify training expenditure for offline GP training, where the training cost comprises mostly of training time. When a GP is trained online during a UA mission, several factors such as fuel, computation time, endurance, etc. factor into and contribute to the training cost. However, the general nature of the findings in this Section will extend to the online

training process as well.

Fortunately, once the GP is trained and the hyperparameters are learned, the covariance matrix does not change and inverting it is a one time operation. This inversion can be precomputed and used for all future predictions, making predictions much cheaper than the training phase.

### 3.3 Summary

Airborne radio frequency (RF) communication for small unmanned aircraft can be characterized by capturing local, receiver-dependent RF variations from baseline models. By learning these spatial or spatio-temporal variations using the nonparametric, data-driven approach of a Gaussian process (GP), an **a priori** model like the empirical path loss model can be adapted to a specific environment. Diverse RF behaviors can be captured by this fully probabilistic methodology using a variety of kernel functions. This chapter evaluates and compares the performance of spatial and spatio-temporal GP models against the *a priori* empirical model by performing cross validation on 50 flight measurement datasets. These datasets consist of 433MHz and 2.4GHz signal strength measurements from stationary and mobile transmitters, collected from three different small unmanned aircraft. Measuring the performance using root mean squared error (RMSE) as well as mean standardized log loss (MSLL) evaluates both the predicted estimate and its uncertainty, and shows that the GP models improve prediction accuracy over the *a priori* empirical model, with the spatio-temporal GPs improving over the spatial GPs. Finally, examining the effect of training set size on the performance illustrates the inherent trade-off that exists between training cost and accuracy of the GP model.

## Chapter 4

### Improving Data Ferrying by Iteratively Learning the RF Environment

Learning RF variations using a Gaussian process can help adapt *a priori* communication models to specific environments. This improved knowledge of the communication model can provide a significant boost to the performance of communication-aware UAS applications which are designed to take bandwidth and sensing constraints into consideration. This chapter assesses how the ability to learn and adapt the communication model to the environment impacts performance of the communication critical UAS application of data ferrying.

#### 4.1 Data Ferrying

An unmanned aircraft can benefit communication-challenged sensor networks, acting as a data ferry for the network. A data ferry is more than just a wireless relay; the ferry physically carries data as it moves through the environment in order to communicate with the sensor nodes [76, 3]. The unmanned aircraft's mobility extends communication range, enabling communication

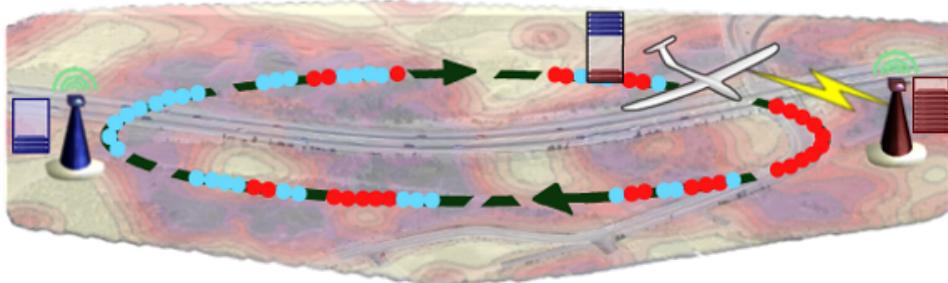


Figure 4.1: Overview of the data ferrying problem.

between distant and otherwise disconnected nodes [95]. A data ferrying system controls two distinct sets of dynamics: the communication link scheduling between nodes and aircraft, and the motion of the aircraft through the environment [96, 83, 25]; and doing this well requires knowledge of the radio-frequency (RF) environment throughout large areas.

RF environments can be modeled theoretically based on signal propagation and decay [66]. Probabilistic shadowing and fast fading [26], as well as terrain features [47] add more detail. Though these models can provide a reasonable starting point for planning, additional variations in the RF environment where the sensors and ferry are deployed will yield modeling errors [55], leading to significant differences between planned and actual data ferry performance [10]. Thus for accurate ferry planning, some form of on-line learning and model refinement is necessary.

Reinforcement learning has been examined for data ferrying [60], avoiding RF models altogether. Here, the ferry can fly many paths with varied communication schedules to determine the optimal solution, in a process similar to stochastic optimization. However, this suffers from the amount of candidate trajectories that the ferry must fly, requiring exponentially more time and energy to reach the optimal solution [28]. Further, RF environments typically exhibit spatial correlations that can be learned and modeled [66, 87, 54, 26]. Instead of the slow model-free approach of reinforcement learning, a model-based approach has the advantage of learning and refining an RF model with fewer passes, resulting in better ferry paths faster.

This idea has been tested with an unscented Kalman filter [81] to fit a physics-based model, showing RF model refinement rapidly after the ferry takes flight. However, [81] limits the communication model to a few set of parameters to represent an omni-directional antenna in the RF environment, which restricts the accuracy of the refined model. Resolution can be improved by incorporating more parameters to capture additional propagation effects, such as fading [52] and directionality [8]. The improved resolution of these models not only requires a larger training set, but will still fall short of the complexity of the RF environment. The parametric nature of physics-based models limits the accuracy in modeling stochastic environments.

Gaussian processes are a non-parametric alternative for measurement prediction in stochastic

environments. While the predictions of the GP improve as more training data from the environment become available, its data requirements are modest compared to the extensive in-flight data collection required for model-free reinforcement learning [60]. The work presented in this chapter uses the GP framework described in Chapter 3 to capture RF variations, i.e. deviations from an *a priori* physics-based model.

Most motion control work for learning these RF variations has focused on environment learning as the sole objective, eg. random motion [87], signal extremum seeking [22, 80], or full environment characterization [22, 35]. Here our objective is data ferrying. Rather than taking time away from ferrying to explore the environment, this work learns the RF environment only *opportunistically* while ferrying. The unmanned aircraft ferries data between the nodes, along a path planned using some initial RF model. While ferrying, the aircraft is able to sense the received signal strength from each node, and feed these measurements into the GP. The GP uses these measurements to refine the RF model, enabling the ferry to replan a smarter ferrying route. Section 4.2 describes the aircraft’s path and communication planner, which requires a model of the RF environment.

This chapter details the integration of the ferry planner with the GP’s RF estimation (Section 4.3). Important considerations are described to smooth this iterative ferry-and-learn process, and the general behavior of the integrated system is discussed. The system is analyzed through simulation in Section 4.4, showing significant ferry performance improvement with opportunistic GP learning. The ferry-and-learn process is then implemented with alternative estimation approaches in Section 4.5. The second contribution is the evaluation of the Gaussian process against learning systems using model-based parameters through least-squares fitting, similar to [81] and [8]. This shows how the geospatial modeling and non-parametric adaptability of the GP allows it to capture environmental artifacts, and thus perform well regardless of the characteristics of the RF environment.

## 4.2 Data Ferry System

Data ferrying refers to an unmanned aircraft transferring data between stationary ground nodes by augmenting wireless communication with physically moving with the data for some portion of its delivery. This work focuses on two nodes,  $A$  and  $B$ , where source node  $B$  needs data delivered to destination node  $A$  through the data ferrying aircraft. This specific scenario simplifies the discussion of the ferrying process; the process can easily be extended to more nodes with more communication flows. To implement on-line, the fast near-optimal discrete-time ferry planner from [11, 9] is used<sup>1</sup>, and described here.

The ferrying system is optimized by controlling the ferry's motion  $\mathbf{u}_m$  and communication  $\mathbf{u}_c$  to form a closed path within a fixed time horizon. This closed path is then discretized into  $N$  steps. Let the sequence of state trajectories be  $\mathbf{p} = \{p_i\}$  for vehicle poses,  $\mathbf{t} = \{t_i\}$  for the associated times, and  $\mathbf{B} = \{\mathbf{b}_i\} = \{b_i^a, b_i^b, b_i^f\}$  for the nodes and ferry, for  $i$  from 0 to  $N$ . The vehicle dynamics can be described as

$$p_{i+1} = f_m(p_i, u_{m,i}) \quad (4.1)$$

$$t_{i+1} = f_t(t_i, u_{m,i}). \quad (4.2)$$

while the buffer dynamics are described by

$$\mathbf{b}_{i+1} = f_c(\mathbf{b}_i, \mathcal{M}_A, \mathcal{M}_B, u_{c,i}) \quad (4.3)$$

where the stochastic radio frequency (RF) environments are represented by  $\mathcal{M}_A, \mathcal{M}_B$ . These map locations to probability density functions (e.g. the mean received signal strength for node  $A$  at location  $i$ ,  $\bar{s}_{a,i}$  and its associated variance  $\sigma_{a,i}^2$ ). The ferry planner uses the mean signal strengths to predict throughput through Shannon-Hartley Channel Capacity  $c = \beta \log_2(1 + \bar{s})$ , where  $\beta$  is the channel's bandwidth. Further, channel coding and MAC-Layer protocols (such as those defined by 802.11 [36]) determine specific communication rates in a stair-stepped fashion based on the channel's capacity. Buffer limits on the nodes and ferry complicate this system even more. For

---

<sup>1</sup> These algorithms were designed and implemented by Anthony Carfang

example, when the ferry's buffer is full, it can no longer collect more data regardless of the RF channel's capacity. These two factors lead to modal nonlinear system dynamics for Equation 4.3, which are described fully in [11].

In addition to the free initial conditions, the entire input is  $\mathbf{u} = [\mathbf{u}_m, \mathbf{u}_c, p_0, \mathbf{b}_0, N]$ , and the periodic ferrying problem is

$$\max_{\mathbf{u}} J(\mathbf{p}(\mathbf{u}_m), \mathbf{t}(\mathbf{u}_m), \mathbf{B}(\mathbf{u}_c)) \quad (4.4a)$$

$$s.t. \quad \text{Eqs. (4.1)-(4.3)} \quad (4.4b)$$

$$p_0 = p_N \quad (4.4c)$$

$$t_0 = 0 \quad (4.4d)$$

$$p_0, \mathbf{b}_0, N \text{ free.} \quad (4.4e)$$

The objective  $J$  is the effective throughput evaluated over the closed ferry path, specifically here the average rate of delivering data to node  $A$ :

$$J(\mathbf{p}, \mathbf{t}, \mathbf{B}) = \frac{b_N^a - b_0^a}{t_N} \quad (4.5)$$

Even with just two sensor nodes, optimizing the ferry's performance is a challenging nonlinear problem with modal dynamics. However, the dynamics exhibit a naturally cascaded structure that can be exploited [11]. The motion and time dynamics in (4.1) and (4.2) depend only on  $\mathbf{u}_m$ . Since channel signal strength generally depends on position and time [87], the buffer dynamics depend on both  $\mathbf{u}_m$  and  $\mathbf{u}_c$ . The full problem objective (4.4a) can be transformed to

$$\max_{\mathbf{u}_m} J(\mathbf{p}, \mathbf{t}, \pi(\mathbf{p}, \mathbf{t})) \quad (4.6)$$

where the optimal bandwidth control policy is dependent on the vehicle's motion:

$$\pi(\mathbf{p}, \mathbf{t}) = \arg \max_{\mathbf{u}_c} J^c(\mathbf{p}, \mathbf{t}, \mathbf{B}(\mathbf{u}_c)) \quad (4.7)$$

At the top level, the ferry planning algorithm optimizes over motion control; then at the lower level, an optimal bandwidth policy is determined for the given motion trajectory.

To solve the higher level nonlinear problem, this work uses a genetic algorithm [89] that controls the ferry’s motion. Each candidate chromosome is fed to the lower level policy in (4.7). To reduce computational burden, this policy uses the fast near-optimal implementations described in [9], obtaining solutions that are on average 99.8 % of the optimal. Significantly, the policy (and the ferry problem) assumes accurate knowledge of the RF environment. Inaccurate knowledge results in discrepancies between the performance that the ferry planner expects and the performance actually achieved. Specifically, over estimating the RF environment can result in the ferry not meeting its data delivery guarantees. Hence, it is critical to learn the RF environment to reduce these discrepancies and improve the ferrying solution. This chapter assesses the impact that learning GP based communication models has on the ferrying performance.

### 4.3 RF Characterization for Data Ferrying

This section describes how the GP-based approach is combined with the ferrying process, using data collected while ferrying, to improve the RF model. The general combined process flow is first described, followed by several key integration details to help ensure smooth operation.

#### 4.3.1 Integrated System Overview

In the first iteration, the ferry plans its trajectory based on *a priori* RF models that cover the full environment for both nodes, setting  $\mathcal{M}_{A,0} = \Xi_A$  and  $\mathcal{M}_{B,0} = \Xi_B$ . These models can be generated using basic radio propagation theory or modeling tools such as SPLAT! [47]. The initial models do not need to be very accurate, though a bound on *a priori* error is still to be determined. The simulation studies in Sections 4.5 and 4.4 will show how over-estimating the RF environment generally leads to better performance.

The genetic algorithm optimizes the ferry’s first path with these initial models. The unmanned aircraft then flies along this path in the real environment, collecting and delivering data to nodes A and B. During this process, it simultaneously measures and logs the strength of the signal  $z_{i,j}$  it is receiving from each of the two nodes. Each of these datasets can be compared to the

*a priori* models' estimates for signal strengths at each location along the path. These differences, given by

$$e_{A,i} = z_{A,i} - \Xi_{A,i} \quad i = 1, \dots, N_A \quad (4.8)$$

are node A's RF variations, as in Equation (3.2). Node B's variations,  $e_{B,i}$  for  $i = 1, \dots, N_B$ , can be similarly computed from the flight measurements.

For each of the nodes, a GP is trained on these location-tagged datasets of RF variations for all measurements taken by the ferry up until that point. These variations are spatial in nature, and are used by the GP to learn hyperparameters  $\theta_{\cdot,n}$  for each radio for every interval, which include the lengthscale or the distance of correlation  $l_s$ , signal variance  $\sigma_f^2$ , and noise variances  $\sigma_n^2$ . These hyperparameters and the training set can then predict, through (3.4), the RF variation for each node at any location in the entire environment

$$\begin{aligned} (\mu_A(p'), \sigma_A^2(p')) &= \mathcal{GP}(p' | \{\mathbf{p}_{1:n}, \mathbf{e}_{A,1:n}, \theta_{A,n}\}) \\ (\mu_B(p'), \sigma_B^2(p')) &= \mathcal{GP}(p' | \{\mathbf{p}_{1:n}, \mathbf{e}_{B,1:n}, \theta_{B,n}\}). \end{aligned} \quad (4.9)$$

Thus, once the training is complete, the learned model can be used to predict the RF variation at any given location  $p'$ . In the rest of the chapter, we use the short forms  $\mathcal{GP}_{A,n}$  and  $\mathcal{GP}_{B,n}$  to represent these learned stochastic models, which predict the signal correction values for some or all of the locations in an environment.

The GP predictions are then combined with the *a priori* model to improve the RF estimates. At the end of iteration  $n$ , the updated models of the RF environments are

$$\begin{aligned} \mathcal{M}_{A,n} &= \Xi_A + \mathcal{GP}_{A,n} \\ \mathcal{M}_{B,n} &= \Xi_B + \mathcal{GP}_{B,n} \end{aligned} \quad (4.10)$$

and are used by the ferry planner in the next iteration to select path  $\mathbf{p}_{n+1}$ . The entire process is summarized in the block diagram shown in Fig. 4.2.

In this manner, improvements learned from the previous iteration's data are incorporated in ferry planning and decision making during the subsequent iteration. The subsequent improvements

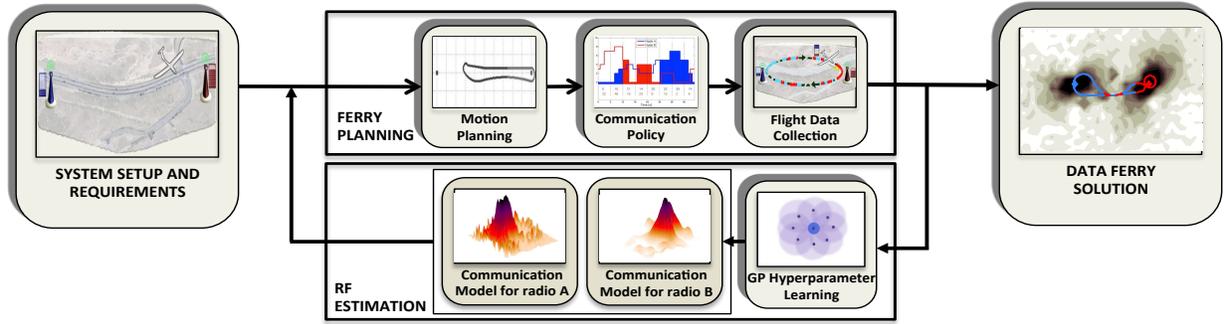


Figure 4.2: Integrating RF characterization with data ferrying involves iterating through planning ferry paths based on a predicted environment, and improving the predictions based on signal strength samples gathered while ferrying.

are a result of accumulating more variation data, which increases the size of the training set, and consequently improves the GP predictions. With better accuracy in the RF models, the ferry can plan a better trajectory.

It is possible that the paths generated between sequential models  $\mathbf{p}_n$  and  $\mathbf{p}_{n+1}$  have no intersections if the predicted environments drastically change between  $\mathcal{M}_{A,n-1}$  and  $\mathcal{M}_{A,n}$ . For this work, the transition between paths  $\mathbf{p}_n$  and  $\mathbf{p}_{n+1}$  is ignored. Future work will investigate adding further constraints to  $p_0$  in (4.4e), as well as transitioning toward a receding horizon framework.

### 4.3.2 Detailed Integration Considerations

#### 4.3.2.1 Initializing Subsequent Iterations

The genetic algorithm to optimize the ferry path uses a population of 20 chromosomes over 200 generations. The initial 20 chromosomes are generated randomly, and evolve according to various crossover and mutation rates that are tuned to work reasonably well for data ferrying (similar to those in [11]). After 200 generations, the best path is then flown to sample the environments. With these measurements known, this trajectory is then used to seed one chromosome of the next iteration, with random chromosomes for the remaining 19. If the RF environment is perfectly known and unchanging between iterations, then seeding the genetic algorithm in this way ensures the ferry's performance will not decrease through iterations. Note this is not guaranteed through

changing environments, i.e. as the RF model changes through learning.

Optimizing the GP hyperparameters has a similar initialization process. The first iteration initializes the hyperparameters to a set of values expected to perform decently for most Signal-Strength applications. In subsequent iterations, the GP is optimized over all accumulated data, rather than just the most recent sampling flight data. Since the data set accumulates, the hyperparameters for each optimization begin with the previous iteration's values. This tends to result in faster GP optimization time.

#### **4.3.2.2 Limited measurement**

The data for each iteration's estimates are sampled from optimized ferry paths, which will tend to be flight patterns near and between both nodes. Over the full environments, this leads to a heavy sampling from a limited region. For a ferry that wants to fly mostly between the nodes, such a sampling region is useful. However, this means the predicted values far from this sampling region will have high variances (low confidence), and high potential for error. The result is that the ferry may miss out on trajectories through unexplored pockets of strong communication. Future work will examine integrating an exploration method in the ferry's trajectory optimization to alleviate this issue.

#### **4.3.2.3 Preventing GP Minimization Failure**

Another issue from heavily sampling a small region is the risk of the GP over-fitting data. In extreme cases, the optimization of the GP may fail, resulting in divergent signal strength estimates. This type of failure can be detected by examining the log marginal likelihood values returned by the conjugate gradient optimization function: if the negative log likelihood values are higher than a preset threshold, the learning is considered to have failed. Empirically, successful maximum likelihood estimations return values of negative log likelihood below  $10^4$ ; negative log likelihood values above  $10^5$  indicate that the optimization failed after getting stuck in a local minimum. Such failures are mostly seen in the early iterations, when limited training data is available. When this

happens, the resulting GP is thrown out, and the ferry returns to the previous model (e.g. set  $\mathcal{M}_{A,n+1} = \mathcal{M}_{A,n}$  instead of (4.10)). This allows the ferry to plan based on the last-known good environment estimate.

## 4.4 Case Study: Results and Discussion

This section presents a simulation case study to illustrate the process of integrating Data Ferrying with Gaussian process RF Estimation, and then to further analyze the system's behavior and performance.

### 4.4.1 Initial Configuration

Two sensor nodes are set in an obstacle-free environment, 800 meters apart. The aircraft flies at a constant speed of 25 m/s as it ferries data between the nodes. Its goal is to maximize the amount of data transferred over a closed trajectory with maximum duration of 2 minutes. The true RF environments for the nodes are based on dipole antennas, several interferers and additional RF noise. Though simulated, this type of environment is similar to those commonly seen during flight experiments [81]. Figure 4.3 shows the mean fields of the RF propagation models for each node. The inherent stochastic nature of the RF environments is reflected by adding a zero-mean normally-distributed variation term  $\nu = \mathcal{N}(0, 0.5)$  to this mean field as the ferry flies through these truth environments. Here, the variance is chosen to reflect the stochasticity of RF signal sampling, based on experiments between ground nodes and unmanned aircraft across the UHF communication band; a value of 0.5 balances the low variance seen at 433MHz [82] and with the higher variance seen at 2.4GHz [81].

Note that though the figures are separated, the two environments do overlap; they are displayed separately for clarity of the environments. Significantly, the signal propagation within these RF environments is such that the spacing of the nodes is far enough to make direct communication a challenge; thus the nodes need the aircraft to ferry data for them.

The *a priori* environment estimates  $\Xi_{(\cdot)}$  shown in Fig. 4.4 have the correct node locations,

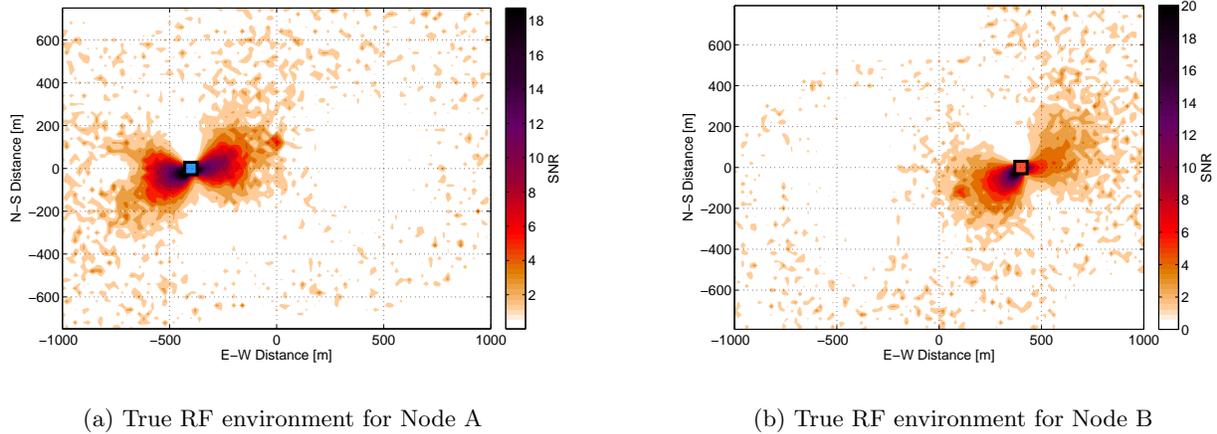


Figure 4.3: True RF environments.

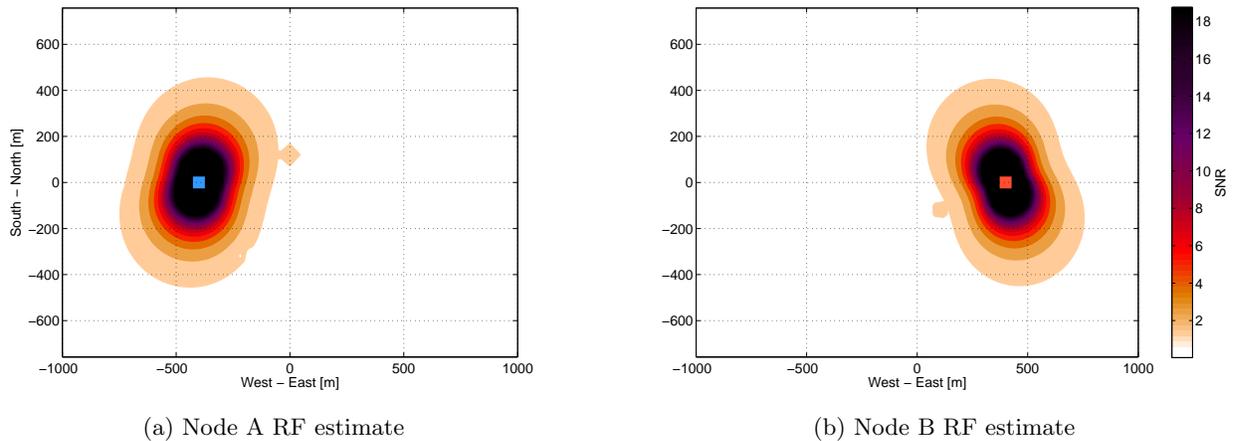


Figure 4.4: Initial estimates of RF environments.

but higher power levels, less interferers, and incorrect antenna angles than the true environment.

In practice, the combined ferry-and-learn system would continue through the duration of the ferrying aircraft's flight. For this evaluation, the system is run for 20 iterations. The performance of our combined ferry-and-learning system is bounded by what the ferry would achieve if optimized with the true RF environment known; for the environments in Figs. 4.3, the optimal path achieves an average throughput of 22.4 Mbps.

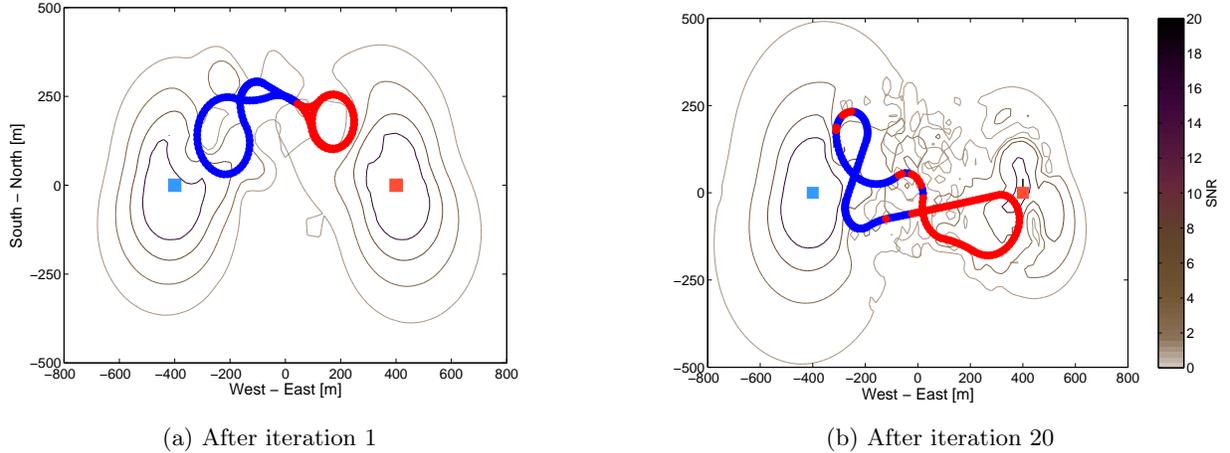


Figure 4.5: The ferry path evolves as the RF model predictions are improved from (a) iteration 1 to (b) iteration 20.

#### 4.4.2 Ferrying Performance

Figure 4.5 shows how the optimized paths vary from the first iteration to the 20th. The red sections of the paths indicate the ferry listening to  $B$ , and the blue for delivering to  $A$ . The first iteration generates a trajectory based on  $\Xi_{(\cdot)}$ , with an expected throughput of 18.2 Mbps, but actually achieving 6.9 Mbps because of the drastic model error. The contour lines behind the ferry paths represent the predicted RF environments after the GP is trained. Over 20 iterations, the predictions are very different in Fig. 4.5b in comparison to Fig. 4.5a, with much more detail in the contour lines between the two nodes where the ferry has sampled the most data. With a much better prediction here, the ferry can plan to take advantage of rapid variations in signal strength by switching links frequently between  $A$  and  $B$ . As a result, the ferry's actual performance converges to its predicted performance, ultimately improving to an effective throughput of 21 Mbps.

The ferry's actual performance improves rapidly while also converging to its predicted performance as the GP predictions are refined (Fig. 4.6). The first 4 iterations show quick convergence between the ferry's expected performance based on  $\mathcal{M}_{(\cdot),i}$  and what is actually obtained while flying through the true environment; the trend also shows rapid improvement for both planned and actual performance. The same trend is evident from iterations 5 to 16. After 16 iterations, the

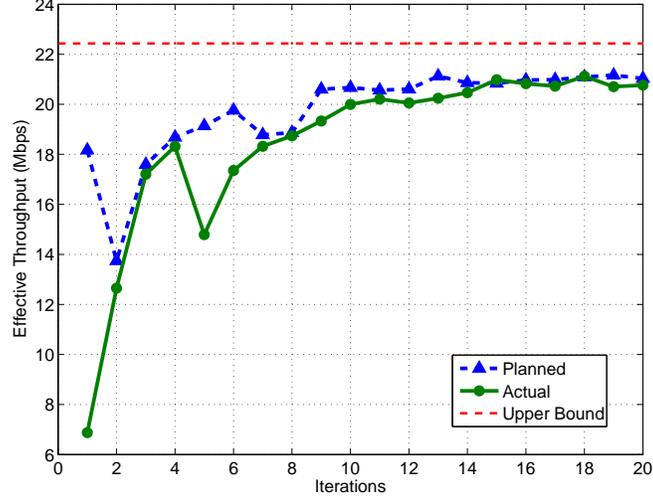


Figure 4.6: Planned and actual ferry throughput performance, ferrying through the estimated RF models and the true RF environment.

solutions have converged to within the signal strength sensing noise.

The dip in performance at iteration 5 results from a significant change in planned path (Fig. 4.7). The ferry path at iteration 4 passes through a region just southwest (down and left) of node  $B$ . In sampling this region, the GP determined the signal strength was generally weaker in that region than the *a priori* estimate had predicted. An unsampled region above and right of node  $B$  was predicted to have good signal strength (shown by the contour lines in Fig. 4.7a), which would improve the ferry’s performance. When flying that path, performance drops as it learns after Iteration 5 that the region has poor signal strength as well. Then with the improved GP predictions of this region, following iterations are able to avoid that region, shown immediately with Iteration 6 (Fig. 4.7c).

### 4.4.3 GP Performance

#### 4.4.3.1 General Behavior

Figure 4.8 shows the RF variations  $\mathbf{e}_{(\cdot)}$  from the true environment that are not captured in the *a priori* model, as explained in Eqs. (3.2) and (4.8). These variations are large and negative around the nodes as the *a priori* model’s incorrect antenna angles and higher power levels overestimate

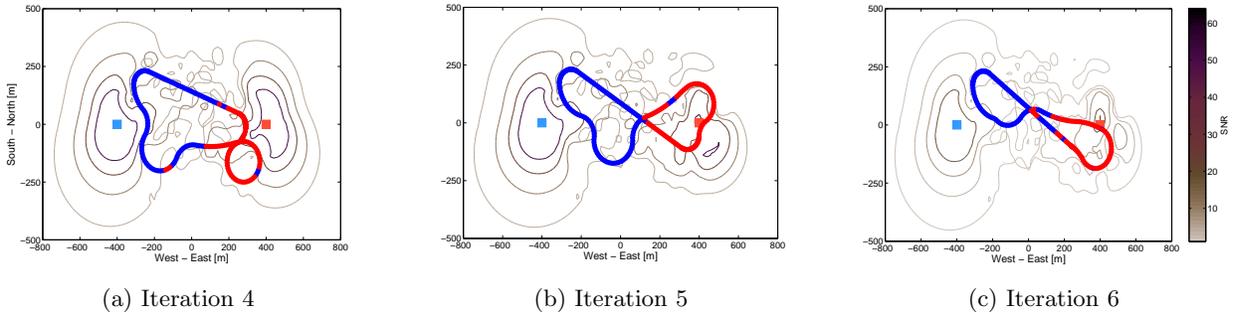


Figure 4.7: Evolution of ferry paths and predicted environments from (a) a previous good trajectory through (b) the region north of node B with previously over-estimated signal strength, then learning to avoid that region in (c).

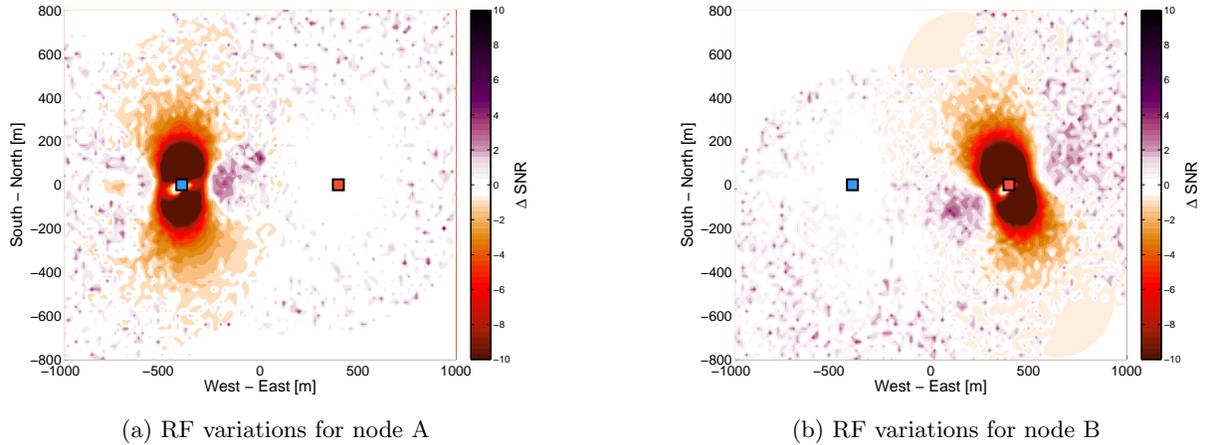


Figure 4.8: RF variations for the *a priori* models of (a) Node A and (b) Node B.

signal strength. Away from the nodes, the *a priori* model assumes a much cleaner decay to 0 than the true environment, resulting in small positive variations. These RF variations are not Gaussian in nature, and thus cannot be modeled by a single additive white Gaussian noise (AWGN) term as in Eq. (2.4). They do however exhibit spatial similarity in the case of both nodes; it is this location-dependent noise (Eq. (3.1)) that can be learned.

The goal of the GP is to learn these RF variations from the *a priori* model by capturing the spatial correlations via its hyperparameters. Figures 4.9a and 4.9b show the mean field of the  $\mathcal{GP}_{B,1}$  and  $\mathcal{GP}_{B,20}$  predictions, respectively. As the training dataset grows with each iteration, the GP learns the RF variations for node B in greater detail by further tuning its hyperparameters.

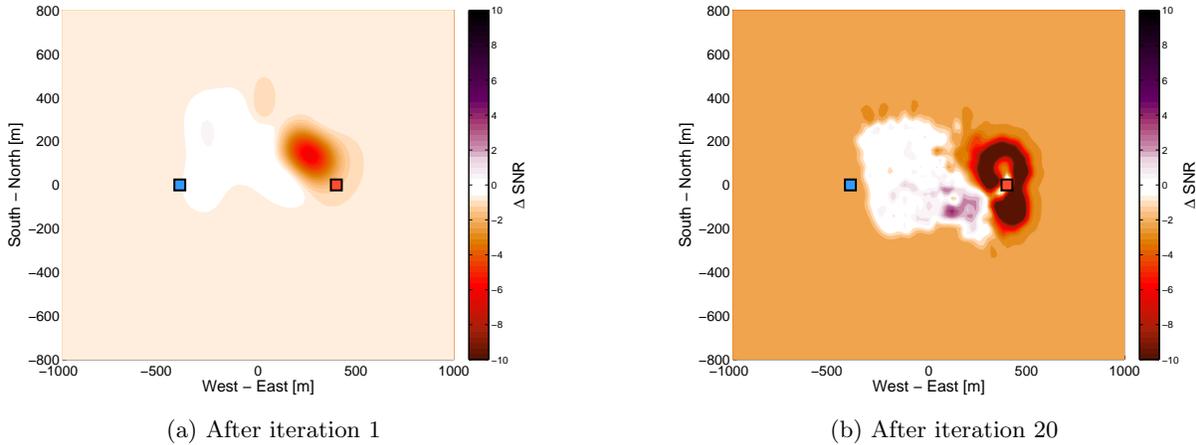


Figure 4.9: Mean prediction for GP learned on RF variations for Node B

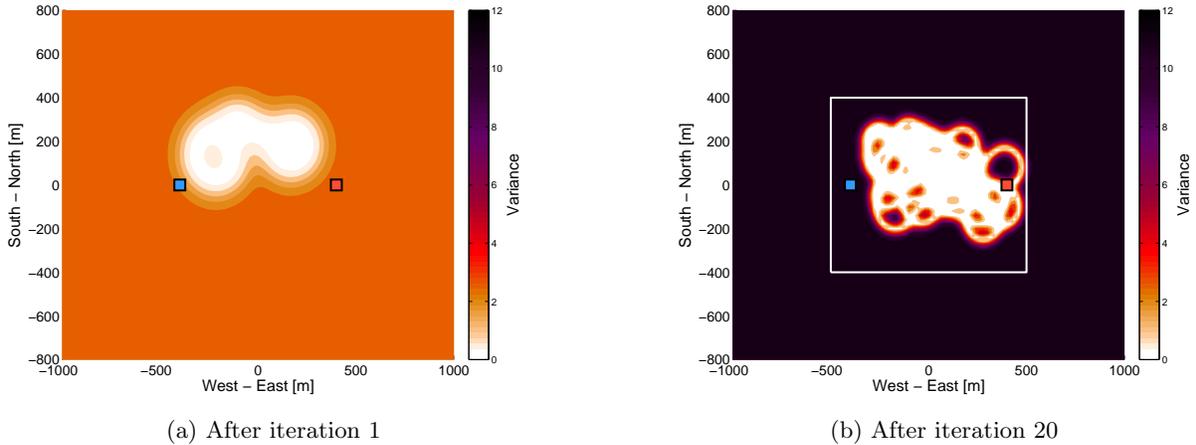


Figure 4.10: Variance for GP learned on RF variations for Node B

For example, after iteration 1, the GP mean prediction (Fig. 4.9a) has very little detail and has not captured any of the key features of Fig. 4.8b. However, by iteration 20 the mean prediction of GP (Fig. 4.9b) has learned most of the dark red region surrounding node B, and closely matches the initial error (i.e the RF variations of the *a priori* model, Fig. 4.8b).

Besides the mean field, GP predictions also provide a variance, which represents the uncertainty in the mean predictions. Figures 4.10a and 4.10b show the GP variance throughout the environment, at iterations 1 and 20, respectively. The white color marks the regions of very low variance, i.e. high certainty, which correspond to the sampled region where the ferry has flown.

Because the GP’s ability to accurately predict decreases further away from the sampled regions, the corresponding variance increases, represented by darker colors. As the ferry samples more of the environment with each iteration, the regions with low variance grow from Fig. 4.10a to Fig. 4.10b. Similar behavior is seen in the GP progression for node A. The white box in Fig. 4.10b represents a symmetric area around the nodes that represents a region intuitively of most interest to a data ferrying unmanned aircraft; and importantly, the variance within this region is significantly reduced.

The largest values of variance in Figs. 4.10a and 4.10b, however, differ greatly. This discrepancy stems from the different hyperparameters learned from different size training sets at these two iterations. At iteration 1, when the training set and sampled region are small, the training samples don’t exhibit much variability. Hence, the GP learns hyperparameters with small values for  $\sigma_f^2$  and  $\sigma_n^2$ , and calculates the maximum variance of 2.81, assuming the rest of the environment has small variance as well. As the training set grows, the GP learns that the signal deviations are in fact highly variable, and it increases the signal variance  $\sigma_f^2$ . Now the maximum variance at iteration 20 is at 11.72 in Fig. 4.10b, which in fact reflects the increase in the uncertainty of the unexplored far away regions, which may also prove to be highly variable.

The value of the GP’s learning can be seen by comparing the model error for the *a priori* model  $\Xi_{(\cdot)}$  (Fig. 4.11a) with the model error for  $\mathcal{M}_{(\cdot),20}$ , close to the region of ferrying interest (shown by the white box in Fig. 4.10b). Figure 4.11 zooms into this box, and shows that the high error regions of Fig. 4.11a (represented by the darker colors) have shrunk considerably after 20 iterations. The exception to this is around the left side of Fig. 4.11b with a non-zero error, an area not flown and sampled by the ferry during the 20 iterations.

#### 4.4.3.2 Root Mean Squared Error (RMSE)

The GP and the overall model’s prediction performance at any iteration  $i$  can be measured by calculating the mean prediction’s error on unseen measurement data, i.e a validation dataset. The errors over the entire validation dataset can be aggregated using root mean squared (RMS)

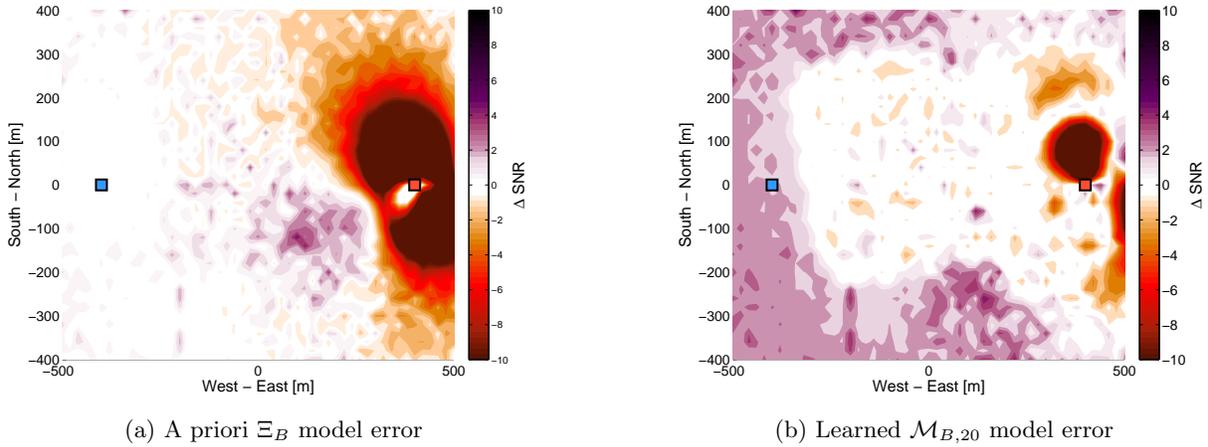


Figure 4.11: Model error comparison for Node B

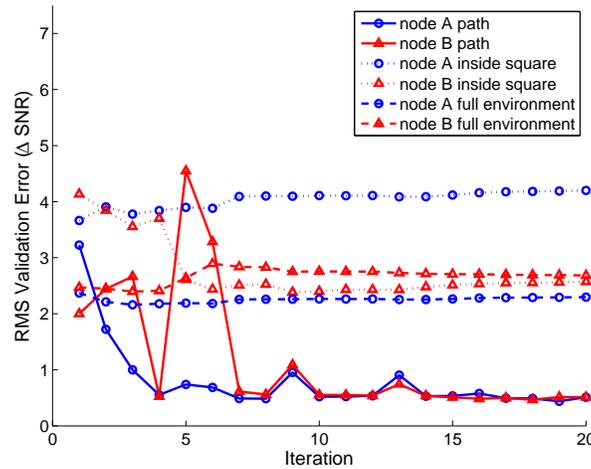


Figure 4.12: GP validation error over 20 iterations.

error (Fig. 4.12). The bold lines represent the RMS error for the path-based validation set, which at iteration  $i$  comprises of points sampled by the path flown during iteration  $i + 1$ , and are used to evaluate  $\mathcal{GP}_i$ . This path is optimized given the predictions of  $\mathcal{GP}_i$ , and being a new path means the new sampled points are not part of  $\mathcal{GP}_i$ 's training set. The path-based RMS error starts out higher than  $2dB$ , but converges to  $0.5dB$ , approaching the variance seen in the true environment. In fact, by the  $7^{th}$  iteration, the RMSE of A and B reduce by 84% and 69%, respectively. The thick dashed lines in Fig. 4.12 represent the RMS error of mean predictions at locations on a uniform

grid over the entire environment, which includes boundary points far outside the sampled regions. Since the ferry paths naturally remain within a small region, the unsampled points dominate this error and prevent it from reducing significantly.

The thin dotted lines in Fig. 4.12 represent the RMSE over the focused ferrying region (defined by the white box in Fig. 4.10b). Note that from Fig. 4.4a, the most significant **a priori** error occurs near the nodes, which bias the RSME values within this focused region. As the ferry flies through this region, the RSME for node B in this region reduces below the full environment error, as expected. However, after 20 iterations, the model’s remaining error is biased by the left side of the region (Fig. 4.11b), which has been left unsampled; hence the region error of 2.57 is only marginally better than the full environment error at 2.68. This is even more drastic for node A, where the unsampled left side of the region includes those high **a priori** errors, which mostly remain through the 20 iterations. This biases the regional RSME of 4.2 higher than the full environment error of 2.3. Significantly though, the ferrying system is able to perform well despite these errors. Because the true environment for node A includes a strong region to the right of the node (Fig. 4.3a), the ferry planner’s erroneously high expectation of throughput around and left of the node was not enough to draw the ferry to that area, leaving that area unsampled. The different behaviors of these regional RMS errors, and the full environment errors, make a compelling case that wider exploration throughout the environment can produce significant reduction in model error, which may help further close the gap between good ferry performance after 20 iterations, and the upper bound in Fig. 4.6.

#### 4.4.3.3 Average Normalized Estimation Error (ANEES)

The RMS error does not capture the fully probabilistic prediction of the GP, because it ignores the variance. To capture this aspect of the GP we evaluate the model using Average NEES (Normalized Estimation Error Squared) or the squared Mahalanobis Distance (MD) (presented in Section 2.3.2. Fig. 4.13 presents the ANEES of the GPs for nodes A and B over 20 iterations.

A high value of NEES indicates optimistic predictions with a low GP variance in spite of

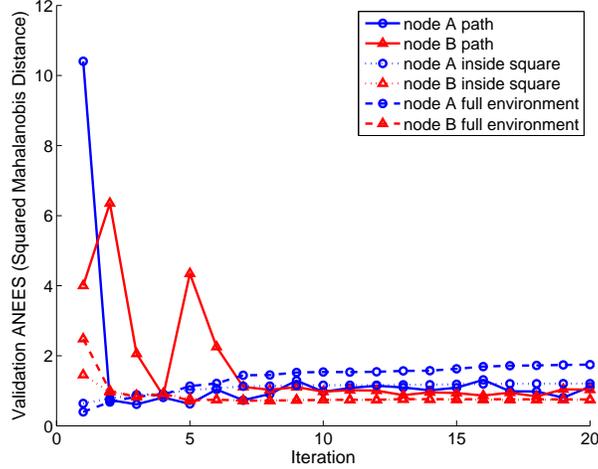


Figure 4.13: GP performance consistency (ANEES) over 20 iterations.

the error being high. For example,  $\mathcal{GP}_{A,1}$  has a RMSE of 3.22 over the path validation set, with variances  $\leq 0.42$ , resulting in ANEES=10.40. In contrast, the same GP,  $\mathcal{GP}_{A,1}$ , predicts pessimistically on the full environment dataset, setting the variance of the far away boundary regions to 16.40 when the RMSE=2.36, leading to a low ANEES=0.39. However, in subsequent iterations, with more data, the ANEES values for both nodes along the path, in the white box region, as well as the full environment quickly converge, indicating that the GP predictions become more consistent and tuned to the inherent characteristics of the field being learned.

#### 4.4.3.4 Mean Standardized Log Loss (MSLL)

As in Section 3.2.2.2 for flight validation of the GP models, the MSLL for the ferry's GP is calculated for a dataset of receiver locations  $\rho_i$  and the corresponding signal measurements  $z_i$  for  $i = 1, \dots, N$ . This is the dataset of the accumulated path, i.e. the training set at the end of 20 iterations. The baseline model at each iteration is the *a priori* model, and its predictive variance  $\sigma_{\Xi}^2$  is equal to the sample variance of its residuals across the entire environment. The GP learned at each iteration is compared to this fixed, *a priori* baseline. For comparison, we also compute the MSLL of the true distribution and a trivial model. The trivial model is defined in the GP literature [68] as the Gaussian with sample mean and sample variance of the training data, denoted by  $\hat{\mu}_n$  and

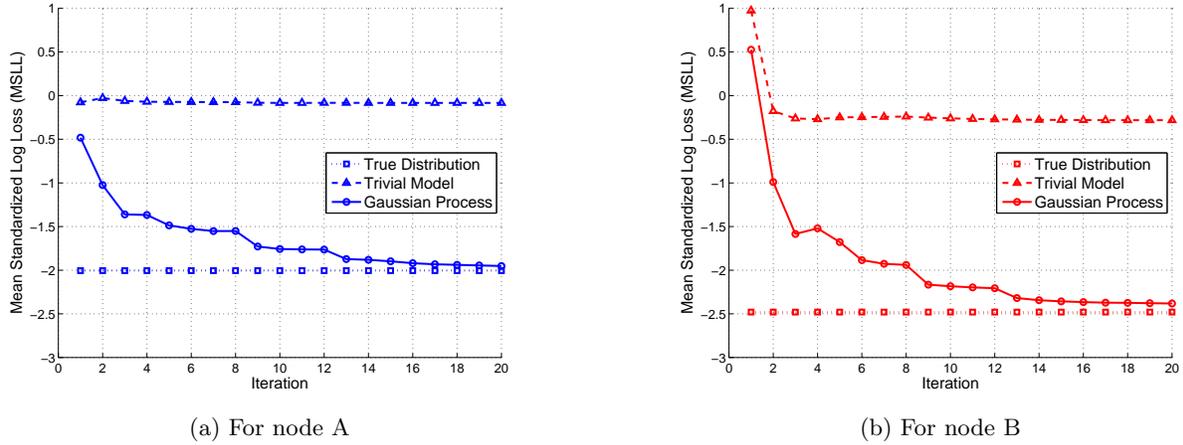


Figure 4.14: Mean Standardized Log Loss (MSLL) for the accumulated path

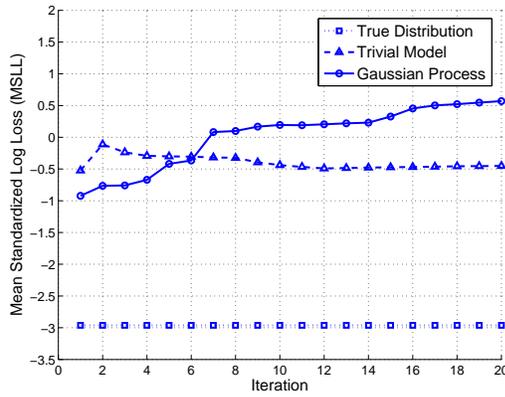
$\hat{\sigma}_n^2$  respectively, for the  $n^{\text{th}}$  iteration. Table 4.1 summarizes the mean and variance of the predictive distributions of these models. Note that  $y_i$  is the true mean field at  $\rho_i$ , and the measurement  $z_i$  is

Table 4.1: Model-wise Predictive Distributions for  $z_i$ .

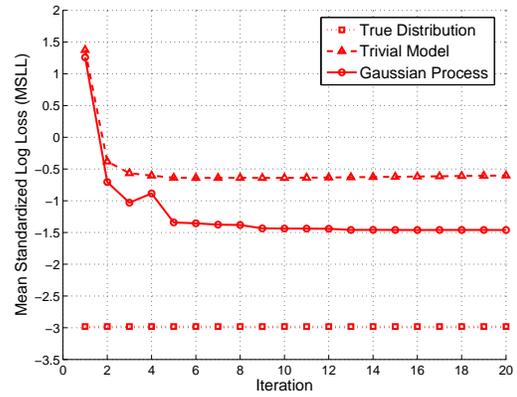
Model	$\mu_i$	$\sigma_i^2$
Baseline	$\Xi_i$	$\hat{\sigma}_{\Xi}^2$
Trivial	$\Xi_i + \hat{\mu}_n$	$\hat{\sigma}_n^2$
Gaussian Process	$\Xi_i + \mu_e(\rho_i)$	$\sigma_e(\rho_i)^2$
True Distribution	$y_i$	0.5

drawn from true distribution i.e  $z_i \sim \mathcal{N}(y_i, 0.5)$ .

Fig. 4.14 shows the MSLL comparison of the models in Table 4.1 for nodes A and B. The predictive distributions are calculated along the accumulated path, i.e. the training set at the end of 20 iterations. For both the nodes, the trivial and *a priori* models perform almost identically, resulting in an MSLL close to zero throughout. However, the red and blue solid lines show that the GP MSLL gets more negative over 20 iterations, improving over the baseline. This is because as the GP receives more training data, i.e. more information about the environment, what the model learns agrees more with what is observed during ferrying. Through this iterative learning the GP not only outperforms the *a priori* and trivial models, but also begins to approach the MSLL of the true distribution.

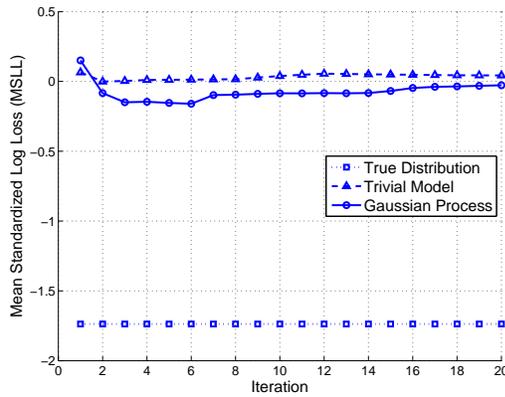


(a) For node A

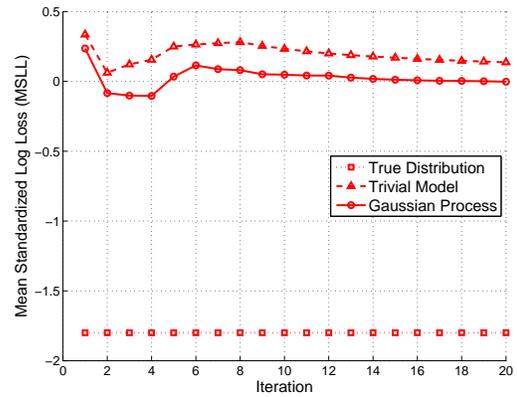


(b) For node B

Figure 4.15: Mean Standardized Log Loss (MSLL) for the white boxed region.



(a) For node A



(b) For node B

Figure 4.16: Mean Standardized Log Loss (MSLL) for uniform grid over environment.

Fig. 4.15 shows the MSLL for the white box region for the two nodes. For node B, the trend and relative behavior of the three models matches those in Fig. 4.14. However, because the ferry does not explore and correct the large errors around node A as explained in Section 4.4.3.2, its MSLL does not reduce over the 20 iterations. Fig. 4.16 shows MSLL for uniform grid points over the entire environment, which is higher than in Fig. 4.14. However, in the absence of training samples from the boundary regions, the disagreement between the GP and unexplored areas cannot be fully removed, resulting in higher log loss and higher MSLL.

As the GP explores the spatially varying regions, its observations reveal more of the complex-

ities of the environment. This increases the GP’s uncertainty for unexplored regions, and causes a rise in the GP’s log loss. This can be seen for node B from iteration 4 to 6, and corresponds to the situation explained in Fig. 4.7 where the ferry discovers two over-estimated regions around node B. To explain such newfound variability and noise in the training data, the GP learns new hyperparameters, which get better tuned as more training data is received in subsequent iterations, resulting in the log loss and MSLR decreasing once again. In this manner, in a complex and spatially varying environment the MSLR cannot be expected to monotonically decrease.

However, the GP’s MSLR at each iteration is negative and lower than the corresponding trivial model. This shows that the GP is building improved models through opportunistic learning.

## 4.5 Learning Comparisons

The ferry-and-learn methodology can use any method for learning RF environments. This section analyzes the choice of Gaussian process learning in comparison to two other common forms of modeling, highlighting the benefits of the GP over the complex RF environments.

### 4.5.1 Radio Model Fidelity

Improving the RF model with a Gaussian process of RF variations takes a data-driven and nonparametric approach to communication modeling. Because a GP defines a distributions over functions, this single methodology can learn the RF variations stemming from diverse RF behaviors and environmental artifacts. This does not require prior knowledge of the hardware of the nodes, or the presence of interferers. Instead, the GP implicitly captures these factors from the training data via the hyperparameters and the correlations within the measurements. While this approach can be computationally more intensive, it provides a single, flexible method for modeling all kinds of radio propagation, as well as for correcting any inaccurate *a priori* model.

In contrast to a GP, physics-based models have parameterized functional forms, which assume specific antenna patterns and how RF transmission would propagate in the environment. These models, then, capture the RF behavior from the training data by parameter estimation. The

following are the two examples of physics models, which differ in the type of antenna they model, and the number of parameters they use.

#### 4.5.1.1 Omni-directional Antenna Model

The omni-directional antenna model [80] assumes that the signal power received at all locations at a range  $r$  away from the transmitter can be modeled as

$$P_{Rx,dBm} = 10 \log_{10} \left( \frac{k_0}{r^\gamma} \right) + \nu. \quad (4.11)$$

This model is an example of the traditional empirical models represented in Equation (2.2), and thus,  $\nu$  represents the additive white Gaussian noise in the measurements. The term  $k_0$  represents the transmitter's power density, and  $\gamma$  represents the path-loss exponent, and is dependent on environmental factors like atmospheric conditions, and obstacles in the environment.

#### 4.5.1.2 Dipole Antenna Model

The dipole antenna [8] model represents the signal power received at a range  $r$  from the transmitter as

$$P_{Rx,dBm} = 10 \log_{10} \left( \frac{k_1 \sin^2(\xi) + k_2 \cos^2(\xi)}{r^\gamma} \right) + \nu \quad (4.12)$$

where, once again,  $\nu$  and  $\gamma$  are the AWGN and the path-loss exponents. In contrast to the omni-directional antenna model, the dipole antenna model combines transmit power, gain, and fading effects via directional power density terms  $k_1$  and  $k_2$ .  $\xi$  is the angle of the aircraft in the frame of the node's antenna pattern:

$$\xi = \psi - \phi \quad (4.13)$$

where  $\psi$  is the nadir angle on the ground node's antenna, and  $\phi$  is the relative angle (from the axis pointing East) between the aircraft and the ground node. While this model is designed to capture directional effects, it can also model omni-directional antennas by setting  $k_1 = k_2$ .

## 4.5.2 Evaluation Setup

To evaluate how the nonparametric approach of a GP overcomes the limitation of physics-based radio modeling approaches, we compare it to estimation systems based on both the omnidirectional and dipole antenna models. Specifically, the ferry-and-learning system is implemented with an estimator that uses either the omnidirectional or the dipole antenna model in place of the GP. In the case of the physics-based models, at each iteration, a least squares solver uses the accumulated ferry measurements for each node, and calculates the radio parameters that best fit the ferry measurements. For the omnidirectional model,  $[k_0, \gamma]$  are estimated with a linear least squares fit, while the dipole model estimates  $[k_1, k_2, \psi, \gamma]$  with a nonlinear least squares solver [8]. These parameters are then used for predicting RF signal strengths within the ferry planner in the following iteration. This process is continued for 15 iterations.

The comparison was performed on RF measurements from two specific truth environments:

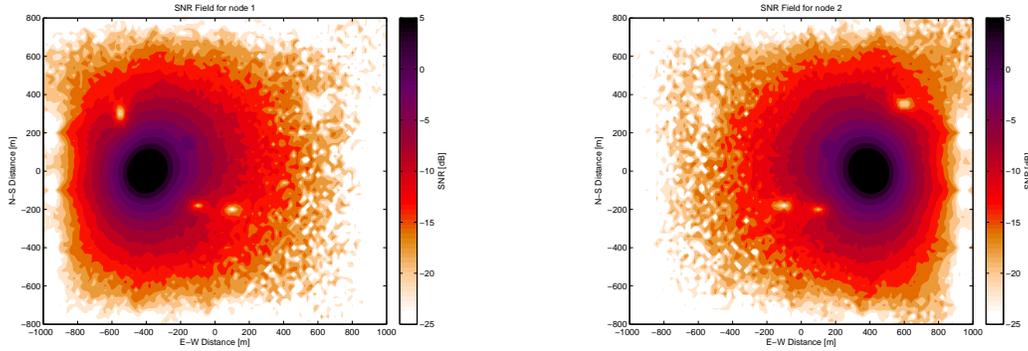
1. nodes with near-omnidirectional antennas, and little interference and thermal noise (Fig. 4.17),
- and 2. nodes with highly directional dipole antennas, in an environment with interferers and significant noise (Fig. 4.18).

These cases validate the GP's benefits over the two extremes of RF environments. For a fair comparison between the solvers, and limited by the few parameters of the omnidirectional model, the estimators begin with an *a priori* model (Fig. 4.19) that reflects an omnidirectional antenna with erroneously high power densities, resulting in RMS errors of 13 dB for the near-omni case, and 18 dB for the dipole case.

## 4.5.3 Comparison Results

### 4.5.3.1 Ferrying Performance

The ferry's performance is bounded by what could actually be achieved if the true environment was known accurately. In both cases, optimizing the data ferry with perfect knowledge results in an effective throughput of 16 Mbps - plus or minus 0.5 Mbps depending on the stochastic fluctuations of the RF environments. Because of this randomness, the following results are averaged



(a) Mean field of truth SNR for A

(b) Mean field of truth SNR for B

Figure 4.17: Case 1: a relatively clean omni-directional truth environment.

from each simulation run 8 times.

Figure 4.20 shows how the effective throughput that the ferry achieves grows over the iterations, as the learned models of the environment improve. In Case 1, the ferry’s actual throughput is 14.62 Mbps for the linear least squares estimator with the omni-directional antenna model, 14.30 Mbps for nonlinear least squares with the dipole antenna model, and 14.50 Mbps for the Gaussian process. Because the true environment varies only slightly from what Eq. 4.11 can capture, the actual performance for each learning system is fairly close over the 15 iterations.

In contrast, the ferry’s throughput for Case 2 shows a much larger variation as the least squares models struggle to capture the nuances of the true RF environments. Not surprisingly, the linear least squares fit of the omni-directional antenna model has the lowest throughput after 15 iterations at 9.31 Mbps; the nonlinear least squares fit of the dipole antenna model captures the true environments a little better to yield a throughput of 13.36 Mbps; the GP captures the true environment best, enabling a ferry performance of 14.61 Mbps.

The differences between the expected performance (dashed lines) and the actual performance (solid lines) in Fig. 4.20 further compare the learner performances. In the first case, although the actual performances are close among the three learners, the model error remaining after 15 iterations using the linear least squares model results in a performance error of 0.69 Mbps; the nonlinear least squares model is a little better at 0.46 Mbps; the GP captures the truth model

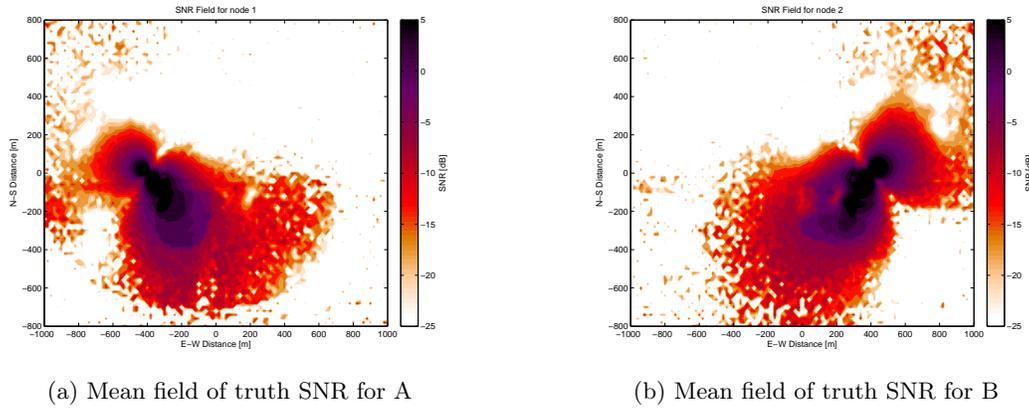


Figure 4.18: Case 2: a much more complicated truth environment, with dipole antennas, noise, and many interferers.

the best, with a performance error of only 0.21 Mbps. In the more complex environments of Case 2, the errors have a larger range, where the performance error for the linear least squares model is 1.1 Mbps; the nonlinear least squares has an error of only 0.28 Mbps; and the GP has a ferry performance error of only 0.19 Mbps.

Though the least squares errors are larger than those of the GP, an error of 1.1 Mbps is not enormous - that is, even the linear least squares model is able to somewhat predict the actual performances of its resulting ferry path; and the nonlinear least squares model predicts performance extremely well. This initially indicates that the least squares models are estimating the RF world reasonably well, despite achieving a lower ferrying throughput. However, this is actually because of model over-fitting. In the iterative ferry-and-learn system, a path is determined based on a limited model; the models are then fit to the signal strength data sampled along these paths. The paths and model predictions eventually stabilize, with the path changing very little, and the models then being fit to very similar data.

This behavior is seen after Iteration 10 in Fig. 4.20b for the two Least Squares models, where the ferry's performance flattens, and the predicted performance converges to it. The same condition can occur for the ferry-and-learn setup with the Gaussian process. However, because of the limitations of the least squares models, the stabilized models and resulting paths are stuck at lower local maxima than that of the GP. To reduce the probability of any models and paths getting

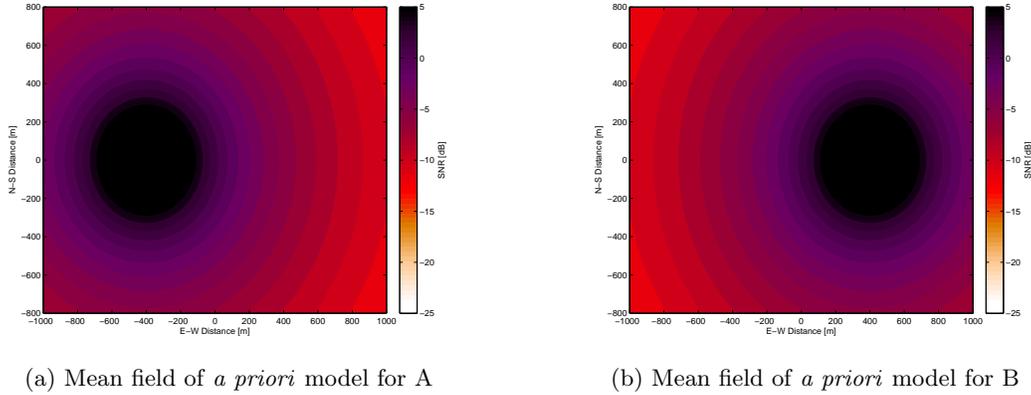


Figure 4.19: Initial estimates, i.e. *a priori* models of the RF environment, used in both cases.

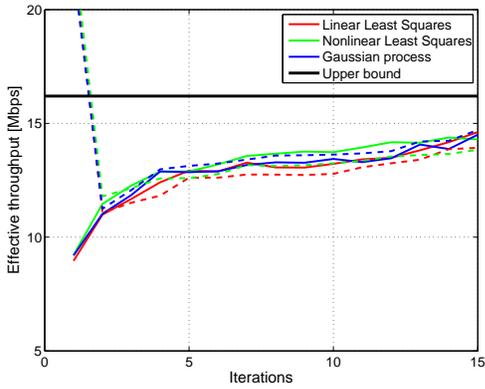
stuck in local maxima, future work will investigate including an exploration component into path planning.

#### 4.5.3.2 Estimator Analysis

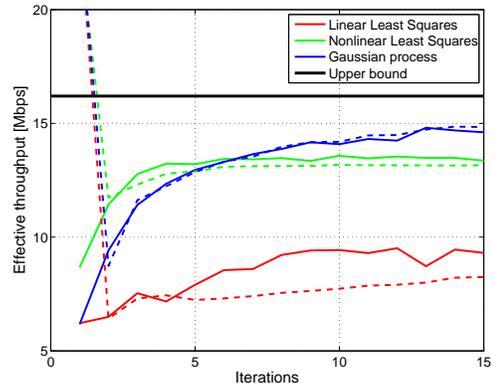
The estimation errors on the path and full environment validation datasets loosely represent the learners' prediction performances in the region between the nodes and throughout the environment, respectively. Consequently, the root mean squared error (RMSE) for the path validation datasets indicates how well the RF environment is known while the ferry is being planned. On the other hand, the RMSE for the full environment validation dataset indicates a learner's ability to extrapolate to unseen regions of the environment.

When the environment is fairly clean and near-omni-directional, all three learners perform similarly, reducing the RMSE from the *a priori* 13 dB to 2 or below on the path validation set, as shown in Fig. 4.21. In fact, with all three learners, the error reduces immediately and converges 3 iterations. This is consistent with the ferry performance of all learners exceeding 12 Mbps only after 4 iterations, as shown in Fig. 4.20a.

In contrast, the path validation RMSE of the three learners after 15 iterations differ greatly in the case of the complicated dipole environment, shown in Fig. 4.22. For both the nodes, the RMSE of the linear least squares estimator is above 5.5 dB, and performs the worst of all 3 learners. While the nonlinear least squares estimator performs better than its linear counterpart, with an

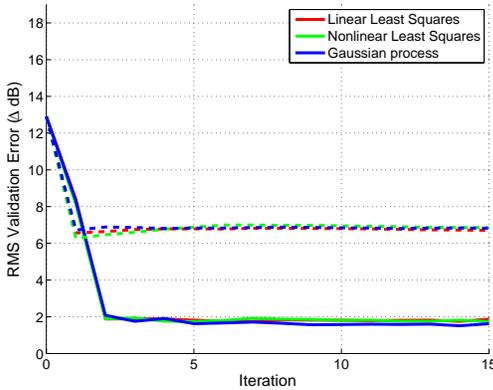


(a) Case 1: near-omni-directional environment

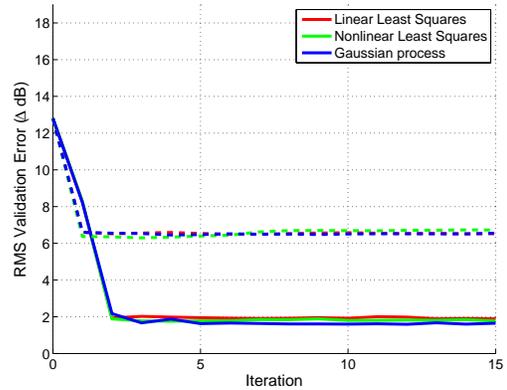


(b) Case 2: dipole environment

Figure 4.20: Ferry’s expected (dashed) and actual (solid) throughput performance compared between the three learners, averaged over 8 runs.



(a) For node A



(b) For node B

Figure 4.21: Path (solid) and full environment (dashed) validation RMS error comparison of the three learners, averaged over 8 runs of the near-omnidirectional environment

RMSE higher than 2.6 dB. Only the Gaussian process learner achieves performance comparable to case 1 with a final RMSE below 2 dB for both nodes. However, faced with a more challenging learning task, the convergence in this case is slower, with the errors falling below 2 dB only after 7 iterations once the training set grows sufficiently large.

When the true environment closely matches a clean omni-directional antenna, all the learned models extrapolate well to most of the unexplored parts of the environment, as seen by the dashed lines in Fig. 4.21. This is because in case 1 the signal largely decays with distance and the

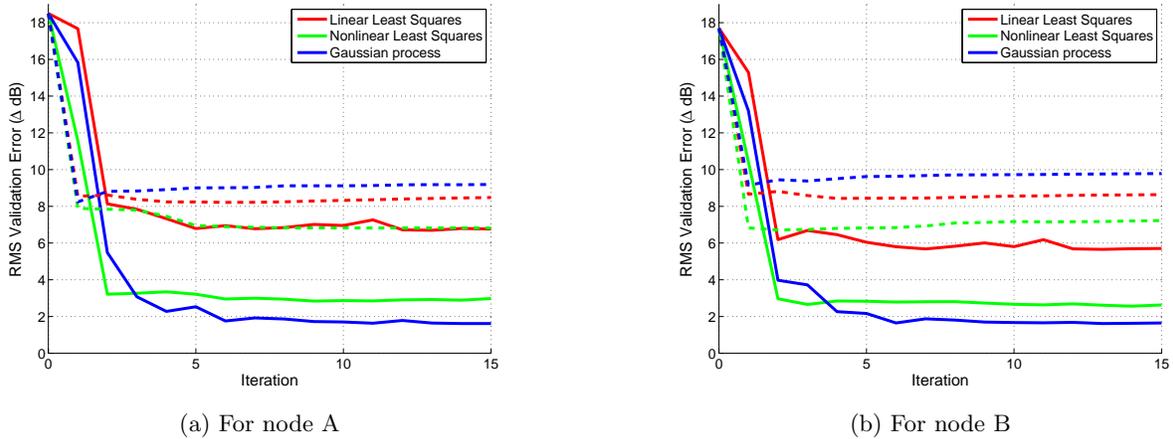
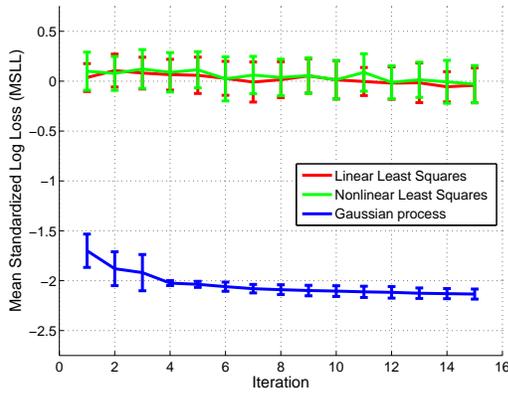


Figure 4.22: Path (solid) and full environment (dashed) validation RMS error comparison of the three learners, averaged over 8 runs of the complicated dipole environment

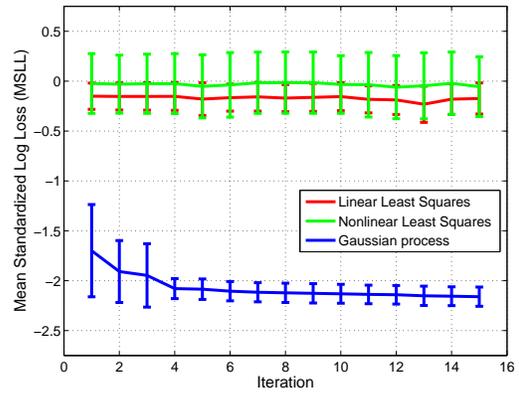
correlations are stationary; and the parameter-based models can capture this trend. This clean trend and stationarity is not present in case 2 with the presence of interferers, making it hard to extrapolate to unexplored regions, consequently increasing the full environment validation RMSE, as shown in Fig. 4.22. In addition, because the GP predicts based on correlations with training data, it especially fails on the boundaries. While the GP does warn against the predicted means in those regions with high variance there, the RMSE cannot capture this indication of uncertainty.

Comparing the models using Mean Standardized Log Loss or MSLL (defined in Section 4.4.3.4) provides an alternate view of how the estimators perform in these stochastic environments. Fig. 4.23 and 4.24 show the MSLL for the three learners on the entire accumulated path, i.e. the training data available to the learners in the last iteration. The line represents the average MSLL over 8 runs, and the error bars show the standard deviation. These MSLL values are computed with *a priori* model as the baseline, with more negative values being better. Once again, the predictive variance of the *a priori* is taken to be the variance in its residuals across the environment.

In the omni-directional scenario, the least squares models are comparable to the *a priori* model, and their MSLL remains close to zero through the 16 iterations. In the dipole case, the linear and nonlinear least squares model achieve average MSLL values of  $-0.7$  and  $0.4$ , respectively. The GP, on the other hand, outperforms the *a priori* and least-squares models in both omni-directional

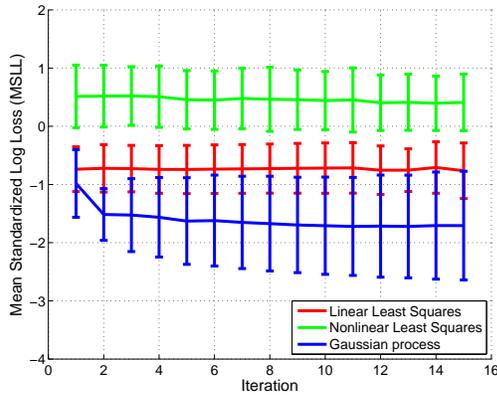


(a) For node A

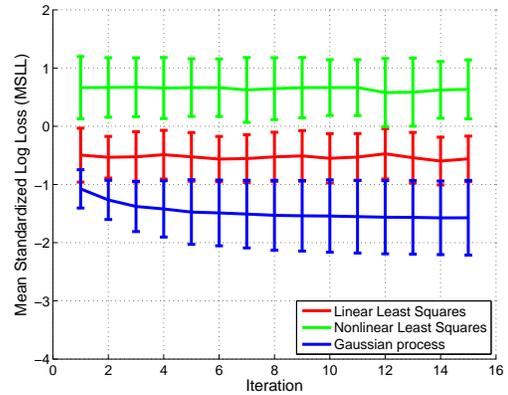


(b) For node B

Figure 4.23: Mean Standardized Log Loss (MSLL) for three learners over the entire accumulated path, averaged over 8 runs of the near-omni-directional environment



(a) For node A



(b) For node B

Figure 4.24: Mean Standardized Log Loss (MSLL) for three learners over the entire accumulated path, averaged over 8 runs of the complicated dipole environment

and dipole scenarios, achieving average MSLL values of  $-2.1$  and  $-1.7$  respectively. Note that the variations of MSLL over the 8 runs is larger in the dipole case than the omni-directional, reflective of the dipole case being inherently more complex.

Table 4.2 summarizes the effectiveness of each learner, and its impact on ferrying performance. The Gaussian process does not predict the RF environment over the full area as well as the nonlinear least squares parameter-based model. However, learning how complicated the RF environment is from the limited data it has seen, the GP better captures the stochastic characteristics of the

environment. Comparing the full probabilistic predictions of the three learners shows how the GP outperforms the least squares models. Thus, the GP is able to adapt to especially complex environments over observed areas; and with the best RSME in the general ferrying region, the GP enables the best and most accurate ferrying performance.

Table 4.2: Radio Model Comparison Summary

Metric	Node	Near-omni-directional			Complicated dipole		
		environment			environment		
		Linear	Nonlinear	GP	Linear	Nonlinear	GP
Ferry Throughput [Mbps]	-	14.62	14.30	14.50	9.31	13.36	14.61
Ferry Throughput Error [Mbps]	-	0.69	0.46	0.21	1.1	0.28	0.19
Path Validation RMSE [dB]	A	1.86	1.74	1.62	6.77	2.98	1.62
	B	1.89	1.77	1.65	5.70	2.63	1.62
Environment Validation RMSE [dB]	A	6.70	6.85	6.82	8.48	6.82	9.19
	B	6.52	6.73	6.55	8.63	7.22	9.78
MSLL over accumulated path	A	-3.89	-3.77	-4.86	-0.98	-3.75	-2.74
	B	-3.55	-3.02	-4.87	-1.16	-2.71	-2.60

## 4.6 Summary

In a data ferrying unmanned aircraft system, ferrying performance requires knowledge of the communication environment through which the aircraft moves. This work integrates ferry optimization with opportunistically learning the radio environment through the use of a Gaussian process (GP). The unmanned aircraft’s trajectory is initially optimized with an **a priori** model. After flying one circuit of the closed trajectory, RF variations observed by the ferry are used to train a GP and improve the model of the environment. This iterative ferry-and-learn system is analyzed through a simulation study, showing ferry performance improves rapidly. The ferry achieves 80% of optimal within 4 iterations, and 93% after 9 iterations, as the GP is able to converge quickly to the true radio frequency environment. This work further compares the GP to common parameter-based estimation methods through two extremes of RF environments. The non-parametric nature of a Gaussian process allows for a higher resolution model, resulting in the ferry’s performance converging to a significantly higher upper bound than parameter-based methods.

## Chapter 5

### Forward Adaptive Transfer for Gaussian Process Regression

Nonparametric learning techniques provide a data-driven approach that is useful for robotic missions in unstructured, diverse, and unexplored environments. Gaussian process regression (GPR) [68, 74] has been used in this manner to capture various environmental phenomena. As demonstrated in the previous chapters, GPs accommodate diverse spatial and spatio-temporal behaviors using a single methodology, learn inherent characteristics of the environment via hyperparameters, and provide fully probabilistic predictions at any location in the environment.

We can take greater advantage of these learned GP models by making them adaptable to subsequent robotic missions in these dynamic environments. Unfortunately, the GP's training is tightly coupled to specific environmental conditions, and changes to the effective environment in subsequent missions can significantly reduce the predictive power of the model. Relearning the model, and achieving performance comparable to the previous model, incurs a large cost. While the computational limitations can be overcome by moving the training optimization off the UA, relearning may still be infeasible given time, endurance, and other resource constraints of the mission. Even if a relearning effort is undertaken through exploration of the environment, the problem remains that the UA does not have a model for reasoning about the world in the interim.

Instead of repeating the expensive and extensive data collection and training phases, transfer learning can help reuse the relevant knowledge from the old model. In this way, the existing GP model can act as a source of information for updating the new target task GP model [88]. This flavor of transfer learning, wherein the source and target tasks are the same, but deal with different

data distributions, is known as transductive transfer learning or domain adaptation [1, 58].

Adaptive Transfer Learning (AT-GP) [12, 7] is the most prominent example of domain adaptation for Gaussian processes. Naturally extending the GP framework, AT-GP uses a transfer kernel to learn task similarity, and uses it to determine the extent to which the source must contribute in target task predictions. Unfortunately, this transfer kernel is designed as a single joint covariance matrix, where both the source and target datasets use the same kernel function, and consequently the same hyperparameters. This design choice imposes an unnecessary constraint on the target task, resulting in an implicit transfer from the source, even when the tasks have zero or little correlation. Because this formulation learns two tasks simultaneously using their combined training corpi, it cannot address how to transfer knowledge from previously learned models, which is a common scenario in robotic missions. Additionally, robotic missions are often resource and time constrained, and the additional computational complexity of learning from the combined datasets is especially undesirable.

In this chapter we describe *forward adaptive transfer learning for Gaussian process regression*, *FAT-GP*, which allows previously learned GP models to be adapted forward as potential sources of knowledge for future learning tasks. FAT-GP combines the source task’s previously learned model, the source task’s training data, and target task’s training data to learn the target hyperparameters as well as the correlation between the two tasks. A conceptual overview of the algorithm is presented in Fig. 5.1.

FAT-GP decouples the kernel and hyperparameter selection for the target task from those of the source task. Because the source task reuses already learned hyperparameters, its large covariance matrix can be precomputed, reducing the computational complexity of the training. Finally, in order to ensure that the joint covariance matrix is positive semidefinite, the cross covariance kernel function is defined as a product of the task similarity and the convolution of the source and target task kernels.

The main contribution of this chapter is providing a framework for robotic learning tasks to leverage previously learned GP models, which can be especially valuable when limited training

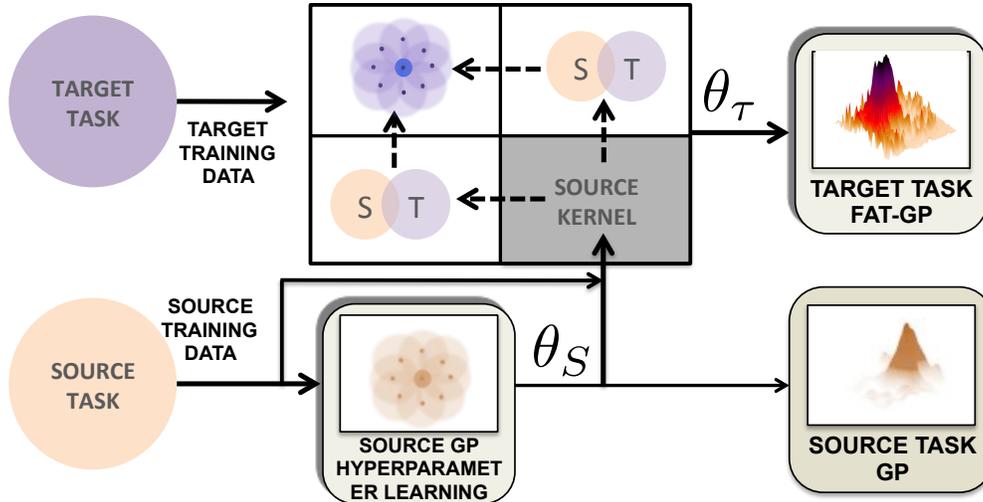


Figure 5.1: FAT-GP Overview

data is available for the new task. Reusing previously learned models allows past experience, while staying intact, to factor into new decisions, as and when relevant. This feed-forward structure exploits the synergy between old and new learning tasks in keeping with the idea of lifelong learning. At the same time, the FAT-GP algorithm seamlessly handles situations where the environment undergoes a drastic change, and relearning from scratch is inevitable. FAT-GP provides robots a way to reason about the world, as it is transitioning. Such a transfer learning algorithm would be useful while modeling time-varying fields like temperature, winds, and other dynamic phenomena.

## 5.1 FAT-GP: Forward Adaptive Transfer for Gaussian Processes

### 5.1.1 Target Task Prediction

Consider two regression tasks  $S$  and  $T$ , which operate on input and response variables  $\mathbf{x}$  and  $y$  of the same dimensionality, and may be related. Additionally, we assume that  $S$  has already been completed, and a Gaussian process model  $\mathcal{GP}_S$  with hyperparameters  $\theta_S$  has been learned. The goal of FAT-GP is to transfer relevant knowledge from source task  $S$  to incomplete target task  $T$ .

Gaussian processes make the smoothness assumption, whereby the response variables  $y_i$  have

a Gaussian joint distribution.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{bmatrix}, \begin{bmatrix} c_{11} & k_{12} & \cdots & k_{1N} \\ k_{21} & c_{22} & \cdots & k_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \cdots & c_{NN} \end{bmatrix} \right) \quad (5.1)$$

In keeping with this, FAT-GP assumes that the source and target task labels are jointly distributed as

$$\begin{bmatrix} \mathbf{y}_T \\ \mathbf{y}_S \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{m}_T \\ \mathbf{m}_S \end{bmatrix}, \begin{bmatrix} C_{TT} & K_{TS} \\ K_{ST} & C_{SS} \end{bmatrix} \right) \quad (5.2)$$

where the mean and covariance matrices are denoted by  $\mathbf{m}_{joint}$  and  $C_{joint}$ . Subscripts  $S$  and  $T$  represent components that are specific to source and target tasks, respectively, and superscript  $T$  is the transpose of the matrix. The on-diagonal block matrices are given by  $C_{(\cdot)} = K_{(\cdot)} + \sigma_{n(\cdot)}^2 I$ , as defined in Section 3.1.1. Thus,  $C_{TT}$  and  $C_{SS}$  are square matrices with dimensions  $N_T$  and  $N_S$  respectively, and are defined over the task-specific training samples  $X_S$  and  $X_T$ .  $K_{ST} = K_{TS}^T$  are the cross covariance matrices, where  $K_{TS}$  is a  $N_T \times N_S$  rectangular matrix. Thus, matrix  $C_{joint}$  is a symmetric, square matrix with dimensionality  $N_T + N_S$ . Similarly,  $\mathbf{m}_{joint}$  is a column vector of length  $N_T + N_S$ .

Since  $C_{joint}$  is a kernel covariance matrix, it must be positive semi-definite (PSD). One way to ensure that  $C_{joint}$  is PSD is to ignore task separation between the two datasets and define  $C_{TT}$ ,  $C_{SS}$ , and  $K_{TS}$  using the same kernel function and hyperparameters. Let this formulation of  $C_{joint}$  be denoted by  $C_D$ , such that

$$C_D(x_i, x_j) = k(x_i, x_j; \theta) + \sigma_n^2 \delta_{ij} \quad \text{for } i, j = 1, \dots, N_D \quad (5.3)$$

where  $k$  is any valid kernel function with hyperparameters  $\theta$ ,  $N_D = N_T + N_S$ , and  $x_i$  and  $x_j$  belong to  $X_D = [X_T^T, X_S^T]^T$ . This is equivalent to learning a single standard GP (as defined in Section 3.1.1) by combining both tasks' datasets. However, this formulation gets rid of valuable context that the data comes from two different tasks with different distributions.

AT-GP, which learns source and target simultaneously, capitalizes on this contextual information so that prediction for a certain task is based on all training data for that task, and the influence of data from the other potentially-correlated task is controlled by a scaling factor  $\lambda \in [-1, +1]$ . Hence, this formulation of  $C_{joint}$  is defined as

$$C_{ATGP}(x_i, x_j) = \begin{cases} k(x_i, x_j; \theta) + \sigma_{n(\cdot)}^2 \delta_{ij} & \text{when } x_i \text{ and } x_j \text{ belong to the same task} \\ \lambda k(x_i, x_j; \theta) & \text{when } x_i \text{ and } x_j \text{ belong to different tasks} \end{cases} \quad (5.4)$$

where  $\sigma_{n(\cdot)}^2$  is the noise variance. Note that, in contrast to Equation (5.3),  $i, j = 1, \dots, N_{(\cdot)}$  when  $x_i$  and  $x_j$  belong to the same task, whereas  $i = 1, \dots, N_T$  and  $j = 1, \dots, N_S$  when they belong to different tasks. While this formulation still imposes the same kernel function on both tasks, its use of  $\lambda$  distinguishes the  $C_{TT}$ ,  $C_{SS}$ , and  $K_{TS}$ , maintaining task boundaries.

FAT-GP also maintains task boundaries, but takes a modular approach with the design of the block diagonal matrices of the covariance matrix. It assumes that the source and target tasks have different and task-specific kernels and hyperparameters. Thus, the joint covariance matrix is formulated as

$$C_\tau(x_i, x_j) = \begin{cases} k_{\tau(T)}(x_i, x_j; \theta_{\tau(T)}) + \sigma_{n\tau(T)}^2 \delta_{ij} & \text{when } x_i, x_j \in T \text{ and } i, j = 1, \dots, N_T \\ k_S(x_i, x_j; \theta_S) + \sigma_{nS}^2 \delta_{ij} & \text{when } x_i, x_j \in S \text{ and } i, j = 1, \dots, N_S \\ \lambda k_{cross}(x_i, x_j; \theta_S, \theta_{\tau(T)}) & \text{when } x_i \in T \text{ and } x_j \in S \text{ i.e. different tasks} \end{cases} \quad (5.5)$$

where  $\theta_{\tau(T)}$  denotes target task hyperparameters learned during the FAT-GP training. Because the kernel functions corresponding to the block diagonal matrices have different hyperparameters, the selection of the kernel function for the cross covariance block matrices has to be made carefully to ensure that  $C_\tau$  is positive semidefinite. Results presented in [50] define the kernel function for cross covariance matrices as the convolution of the kernels used in the block diagonal matrices.

$$k_{cross} = k_S * k_{\tau(T)} \quad (5.6)$$

Similar to AT-GP, we multiply the cross covariance matrices by a scalar  $\lambda \in [-1, +1]$ , which is a measure of the similarity of the two tasks.

$$K_{TS} = \lambda K_{cross} = K_{ST}^T \quad (5.7)$$

Making use of Theorem 1 from [7], it can be shown that the  $C_\tau$  is PSD when the cross covariance matrices  $K_{TS}$  and  $K_{ST}$  are defined as in Equation (5.7). The similarity measure,  $|\lambda| \leq 1$ , is an additional hyperparameter, which captures the correlation between the source and target task, and is learned along with  $\theta_{\tau(T)}$ . When the source GP is a previously learned model, a value of  $\lambda$  close to zero signifies that past experience is obsolete, and the new model must be learned from scratch, possibly after extensive data collection. Thus, the joint covariance matrix for FAT-GP is given by

$$C_\tau = \begin{bmatrix} C_{TT} & K_{TS} \\ K_{ST} & C_{SS} \end{bmatrix} = \begin{bmatrix} C_{\tau(T)}(X_T, X_T) & \lambda K_{cross}(X_T, X_S) \\ \lambda K_{cross}^T(X_T, X_S) & C_S(X_S, X_S) \end{bmatrix} \quad (5.8)$$

where all the block matrices are functions of (source-specific, task-specific, and task-similarity based) hyperparameters. Of these, the source hyperparameters are known *a priori*. The remaining unknown hyperparameters, denoted by  $\theta_\tau = \{\theta_{\tau(T)}, \lambda\}$ , are learned during training.

The similarity between the two tasks can also be captured by making  $\lambda$  a function of inputs to the kernel function i.e. making  $\lambda$  a function of  $x$ . While this nonstationary formulation would provide a higher resolution understanding of the correlations across tasks, the number of hyperparameters would increase by  $N_S \times N_T$ , making the transfer cost prohibitive.

Consider, for example, a FAT-GP where both tasks use a Gaussian kernel with  $\theta_S = \{\sigma_{f_S}^2, \sigma_{n_S}^2, L_S\}$  and  $\theta_{\tau(T)} = \{\sigma_{f_T}^2, \sigma_{n_T}^2, L_T\}$ , respectively. In both tasks,  $\sigma_{f(\cdot)}$  represents the signal variance, while  $L_{(\cdot)}$  is the diagonal matrix of lengthscales. Thus, the kernel functions for the on-diagonal matrices is given by

$$k_{(\cdot)}(x_i, x_j) = \sigma_{f(\cdot)}^2 \exp \left[ -\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T L_{(\cdot)}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \right] \quad (5.9)$$

and the cross covariance kernel is given by

$$k_{cross}(\mathbf{x}_{T,i}, \mathbf{x}_{S,j}) = 2^{D/2} \sqrt{\sigma_{f_T}^2 \sigma_{f_S}^2} \frac{|L_{\tau(T)}|^{\frac{1}{4}} |L_S|^{\frac{1}{4}}}{|L_{\tau(T)} + L_S|^{\frac{1}{2}}} \exp \left[ -\frac{1}{2} (\mathbf{x}_{T,i} - \mathbf{x}_{S,j})^T \left( \frac{L_{\tau(T)} + L_S}{2} \right)^{-1} (\mathbf{x}_{T,i} - \mathbf{x}_{S,j}) \right] \quad (5.10)$$

where  $D$  is the number of anisotropic input dimensions. If, however, the GP is isotropic i.e. the

same lengthscale is used for all input dimensions, the cross covariance kernel would be given by

$$k_{cross}(\mathbf{x}_{T,i}, \mathbf{x}_{S,j}) = \sqrt{\sigma_{f\tau(T)}^2 \sigma_{fS}^2} \sqrt{\frac{2l_{\tau(T)}l_S}{l_{\tau(T)}^2 + l_S^2}} \exp \left[ -\frac{\|\mathbf{x}_{T,i} - \mathbf{x}_{S,j}\|^2}{l_{\tau(T)}^2 + l_S^2} \right] \quad (5.11)$$

An important and notable characteristic of the cross covariance kernel function in Equations (5.10) and (5.11) is that they are non-stationary [56, 62] in the task domain i.e. kernel function averages over the lengthscales local to each of the tasks. As a result, this kernel function implicitly captures the covariance between two  $\mathbf{x}$  inputs as a combination of the characteristics of both tasks.

Once the FAT-GP is trained and the hyperparameters are known (as explained in Section 5.1.2) it can be used to predict the target task response  $y'_T$  for any unseen input variable  $\mathbf{x}'$ . The joint distribution of  $y'$  with  $\mathbf{y}_T$  and  $\mathbf{y}_S$  can be written by calculating  $\mathbf{x}'$ 's correlation with target as well as source training samples.

$$\begin{bmatrix} y' \\ \mathbf{y}_T \\ \mathbf{y}_S \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} m_{\tau(T)}(\mathbf{x}') \\ \mathbf{m}_{\tau(T)}(X_T) \\ \mathbf{m}_S(X_S) \end{bmatrix}, \begin{bmatrix} k_{\tau}(\mathbf{x}', \mathbf{x}') & k_{\tau}(\mathbf{x}', X_T) & k_{\tau}(\mathbf{x}', X_S) \\ k_{\tau}(\mathbf{x}', X_T)^T & C_{TT} & K_{TS} \\ k_{\tau}(\mathbf{x}', X_S)^T & K_{ST} & C_{SS} \end{bmatrix} \right) \quad (5.12)$$

The kernel functions  $k_{\tau}$  are interpreted based on the task assignment of their two inputs. Hence, since  $\mathbf{x}'$  belongs to the target task, based on Equation (5.5)

$$\begin{aligned} k_{\tau}(\mathbf{x}', \mathbf{x}') &= k_{\tau(T)}(\mathbf{x}', \mathbf{x}') \\ k_{\tau}(\mathbf{x}', X_T) &= k_{\tau(T)}(\mathbf{x}', X_T) \\ k_{\tau}(\mathbf{x}', X_S) &= \lambda k_{cross}(\mathbf{x}', X_S) \end{aligned} \quad (5.13)$$

Consequently, the prediction for  $y'$  is given by its conditional distribution

$$p(y' | \mathbf{x}', \mathbf{y}_T, X_T, \mathbf{y}_S, X_S, \theta_{\tau}, \theta_S) = \mathcal{N}(\mu'_{\tau}, C'_{\tau})$$

where

$$\begin{aligned} \mu'_{\tau} &= \mathbf{m}_{\tau(T)}(\mathbf{x}') + \begin{bmatrix} k_{\tau}(\mathbf{x}', X_T) & k_{\tau}(\mathbf{x}', X_S) \end{bmatrix} C_{\tau}^{-1} \left( \begin{bmatrix} y_T \\ y_S \end{bmatrix} - \begin{bmatrix} \mathbf{m}_{\tau(T)}(X_T) \\ \mathbf{m}_S(X_S) \end{bmatrix} \right) \\ C'_{\tau} &= k_{\tau}(\mathbf{x}', \mathbf{x}') + \sigma_{n\tau(T)}^2 - \begin{bmatrix} k_{\tau}(\mathbf{x}', X_T) & k_{\tau}(\mathbf{x}', X_S) \end{bmatrix} C_{\tau}^{-1} \begin{bmatrix} k_{\tau}(\mathbf{x}', X_T) \\ k_{\tau}(\mathbf{x}', X_S) \end{bmatrix} \end{aligned} \quad (5.14)$$

Equations (5.12), (5.14) and (5.13) easily scale from predictions for individual target inputs  $\mathbf{x}'$  to individual or joint prediction for multiple target inputs  $X'$ .

### 5.1.2 Computational Complexity of Hyperparameter Learning

The unknown hyperparameters in Equations (5.16) and (5.7) are  $\theta_\tau = \{\theta_{\tau(T)}, \lambda\}$ , where  $\tau$  represents a forward transfer inference, and  $\theta_{\tau(T)}$  denotes target task hyperparameters learned by FAT-GP.

Unlike the conventional GP, FAT-GP learns the hyperparameters by maximizing the log marginal likelihood of only the target response variables given the source data.

$$\theta_\tau^* = \arg \max_{\theta_\tau} \ln p(\mathbf{y}_T | \mathbf{y}_S, X_T, X_S, \theta_S, \theta_\tau) \quad (5.15)$$

Using Baye's rule on Equation (5.2), the marginal distribution of the response variables of the target task,  $\mathbf{y}_T$ , is conditionally inferred from source task as follows

$$p(\mathbf{y}_T | \mathbf{y}_S, X_T, X_S, \theta_T, \theta_S) = \mathcal{N}(\mu_{T|S}, C_{T|S}) \quad (5.16)$$

$$\text{where } \mu_{T|S} = \mathbf{m}_T + K_{TS} C_{SS}^{-1} (\mathbf{y}_S - \mathbf{m}_S)$$

$$C_{T|S} = C_{TT} - K_{TS} C_{SS}^{-1} K_{ST}$$

This approach mitigates two problems encountered when maximizing log likelihood of the joint distribution of source and target (Equation 5.2). First, it avoids calculation and inversion of an  $N_D = N_S + N_T$  size covariance matrix at each iteration of the maximization, instead calculating and inverting  $C_{T|S}$  which  $N_T \times N_T$ . Second, and more importantly, because the source has already been learned, it focuses the learning on the target.

Using the multivariate normal distribution from Equation (5.16) in Equation 5.15, the MLE is written as

$$\begin{aligned} \theta_\tau^* &= \arg \max_{\theta_\tau} \ln \left[ \frac{1}{2\pi^{N_T/2} |C_{T|S}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{y}_T - \mu_{T|S})^T C_{T|S}^{-1} (\mathbf{y}_T - \mu_{T|S}) \right\} \right] \\ &= \arg \max_{\theta_T} \left[ -\frac{1}{2} (\mathbf{y}_T - \mu_{T|S})^T C_{T|S}^{-1} (\mathbf{y}_T - \mu_{T|S}) - \frac{1}{2} \ln |C_{T|S}| - \frac{N_T}{2} \ln 2\pi \right] \end{aligned} \quad (5.17)$$

The first term in Equation (5.17) is the Mahalanobis distance between the observed target response variables  $\mathbf{y}_T$  and the FAT-GP predictive distribution. It quantifies the empirical risk of the learned FAT-GP. The second term is a regularization term which prevents overfitting. The ML estimator finds  $\theta_\tau^*$  by trading off between these two components.

MLE can be performed using a gradient descent optimizer. This requires the computation of the derivative of the log marginal likelihood with respect to each of the hyperparameters in  $\theta_\tau$  [68, 4]. Note that the covariance matrix is dependent on the hyperparameters, but the response variables  $\mathbf{y}_T$  are not.

$$\begin{aligned} \frac{\partial}{\partial \theta_\tau} \ln p(\mathbf{y}_T | \theta_\tau) = & -2 \left[ \frac{1}{2} (\mathbf{y}_T - \mu_{T|S})^T C_{T|S}^{-1} \frac{\partial (\mathbf{y}_T - \mu_{T|S})}{\partial \theta_\tau} \right] \\ & - \frac{1}{2} (\mathbf{y}_T - \mu_{T|S})^T \frac{\partial C_{T|S}^{-1}}{\partial \theta_\tau} (\mathbf{y}_T - \mu_{T|S}) - \frac{1}{2} \frac{\partial \ln |C_{T|S}|}{\partial \theta_\tau} \end{aligned}$$

which can be reduced to

$$\begin{aligned} \frac{\partial}{\partial \theta_\tau} \ln p(\mathbf{y}_T | \theta_\tau) = & (\mathbf{y}_T - \mu_{T|S})^T C_{T|S}^{-1} \frac{\partial \mu_{T|S}}{\partial \theta_\tau} \\ & + \frac{1}{2} (\mathbf{y}_T - \mu_{T|S})^T C_{T|S}^{-1} \frac{\partial C_{T|S}}{\partial \theta_\tau} C_{T|S}^{-1} (\mathbf{y}_T - \mu_{T|S}) - \frac{1}{2} \text{Tr} \left( C_{T|S}^{-1} \frac{\partial C_{T|S}}{\partial \theta_\tau} \right) \end{aligned} \quad (5.18)$$

The cost for each iteration of the training optimization comprises of calculating  $C_{T|S}$  and then inverting it. Equation (5.16) shows that the calculation of  $C_{T|S}$  is dominated by the inversion of  $C_{SS}$ . If both  $\theta_S$  and  $\theta_T$  were unknown (and equal as in AT-GP), the cost of evaluating Equation (5.17) is  $O(N_S^3 + N_T^3)$ , which is dominated by  $O(N_S^3)$  cost of inverting the square matrix  $C_{SS}$  of dimension  $N_S (> N_T)$ . However, in the FAT-GP formulation,  $\theta_S$  is known *a priori*. Hence  $C_{SS}^{-1}$  can be precomputed. This reduces the computational complexity to  $O(N_T^3 + N_S N_T)$ . The final cost savings will depend on the relative sizes of the source and task training sets. Thus, rehashing transfer learning in the context of robot based lifelong learning changes the computational requirements, as summarized in Table 5.1.

The key difference between FAT-GP and AT-GP is that FAT-GP allows  $\theta_S$  and  $\theta_T$  to be different. Note that adding the constraint  $\theta_S = \theta_T$ , and learning them simultaneously transforms FAT-GP to AT-GP as defined in [7]. Thus, AT-GP a special case of FAT-GP.

Table 5.1: Computational cost comparison between AT-GP and FAT-GP (where  $N_S > N_T$ )

	Source Task Learning	Per iteration cost of computing $C_{SS}^{-1}$	Training cost
AT-GP	Simultaneously with target task	$O(N_S^3)$	$O(N_S^3 + N_T^3)$
FAT-GP	Completed previously	Precomputed	$O(N_T^3 + N_S N_T)$

### 5.1.3 FAT-GP Demonstration via 1-D problems

This section uses 1-D problems to illustrate how FAT-GP can harness previously learned models for learning new models efficiently. More importantly, it highlights the improvement in performance that FAT-GP provides over the target GP i.e. GP learned with the limited target training samples. By selecting two 1-D signals which are related through transformation and where the similarity can be visually verified, it is clear how forward adaptive transfer for Gaussian process regression reduces the target task’s validation error in spite of the limited amount of training data.

In the first example, shown in Fig. 5.2, the (purple) target task is a translation of the (orange) source task. Visual inspection of these 1D signals in Fig. 5.2a easily shows they are identical in shape even though  $y_T = y_S + 2$ . The (orange) source signal is learned using a Gaussian process regression on a training set of 65 samples. The output is  $\mathcal{GP}_S$ , which is a nonparametric model for the source task. The mean and variance of the model are represented by the solid black line and the shaded gray area in Fig. 5.2b, while the training samples are denoted by orange dots. Finally, the points in the validation set are denoted by smaller black dots. Testing the prediction performance of  $\mathcal{GP}_S$  on this validation set of 30 samples results in a root mean squared error (RMSE) of 0.43.

The target learning task, on the other hand, has access to only 10 training samples, resulting in a high validation RMS error of 0.82. As shown in Fig. 5.2c, the  $\mathcal{GP}_T$  learns the peak and the trough near  $x = 0$  and  $x = 1$ , respectively. However, due to the limited number of training samples it fails to capture any of the features for  $x \leq -1$  and  $x \geq 1$ . In fact, the high variance at these  $x$  values signifies the GP’s lack of confidence in these predictions.

FAT-GP algorithm learns the hyperparameters by combining the 10 target training samples,

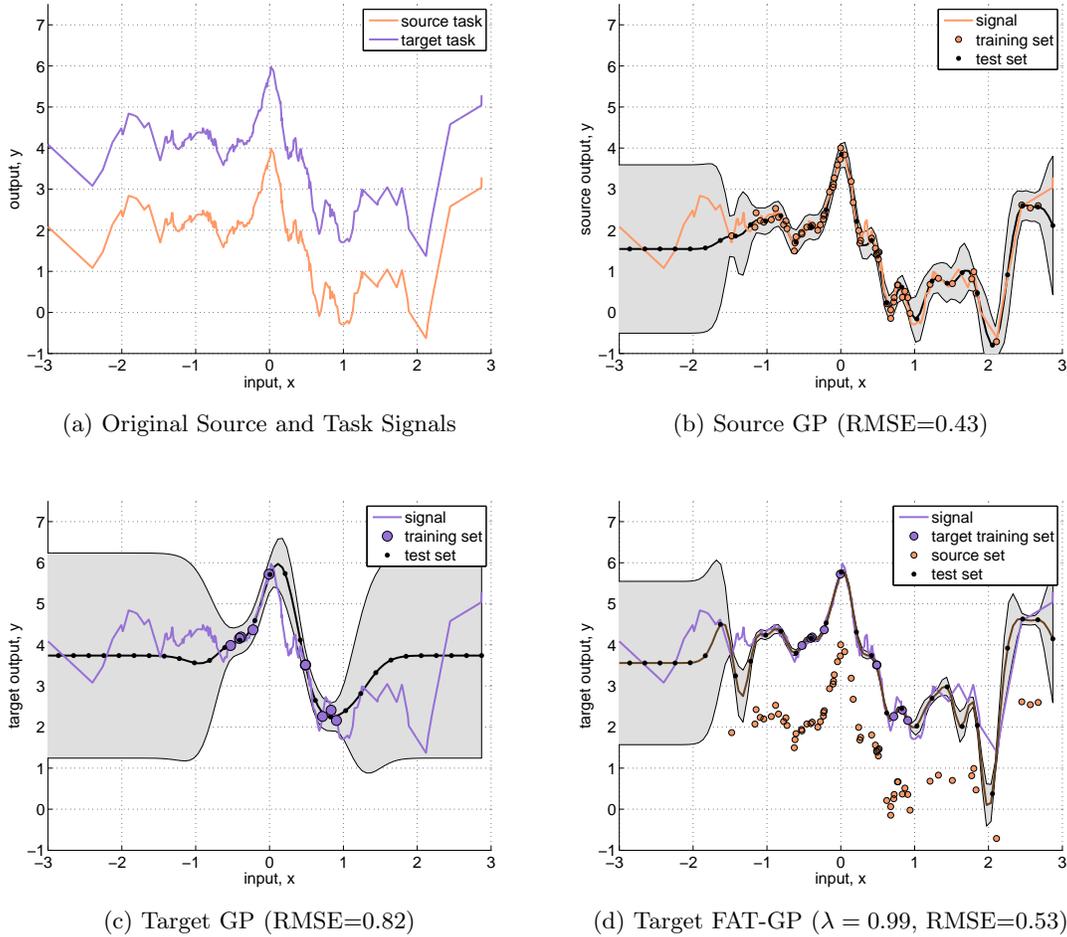


Figure 5.2: Example where target task  $T$  is a translation of source task  $S$  i.e.  $y_T = y_S + 2$

65 source training samples, and source model  $\mathcal{GP}_S$ . During this training process, it learns that the similarity between tasks  $T$  and  $S$  is  $\lambda = 0.99$ . Consequently, the source task heavily contributes to target task, and the resulting FAT-GP is shown in Fig. 5.2d. This FAT-GP's prediction achieves a much higher fidelity with the original (purple) target signal in Fig. 5.2a, and this is reflected in the low validation RMSE of 0.53. In addition, the FAT-GP is also more confident in its estimates having based its inference on a larger dataset of 75 samples.

Similar to the first example, in the second example in Fig. 5.3 the target and source tasks are related through an affine transformation. Fig. 5.3a shows that the target signal  $y_T$  is obtained by scaling the source task  $y_S$  by a factor of 2. This is the same source signal which was used in the

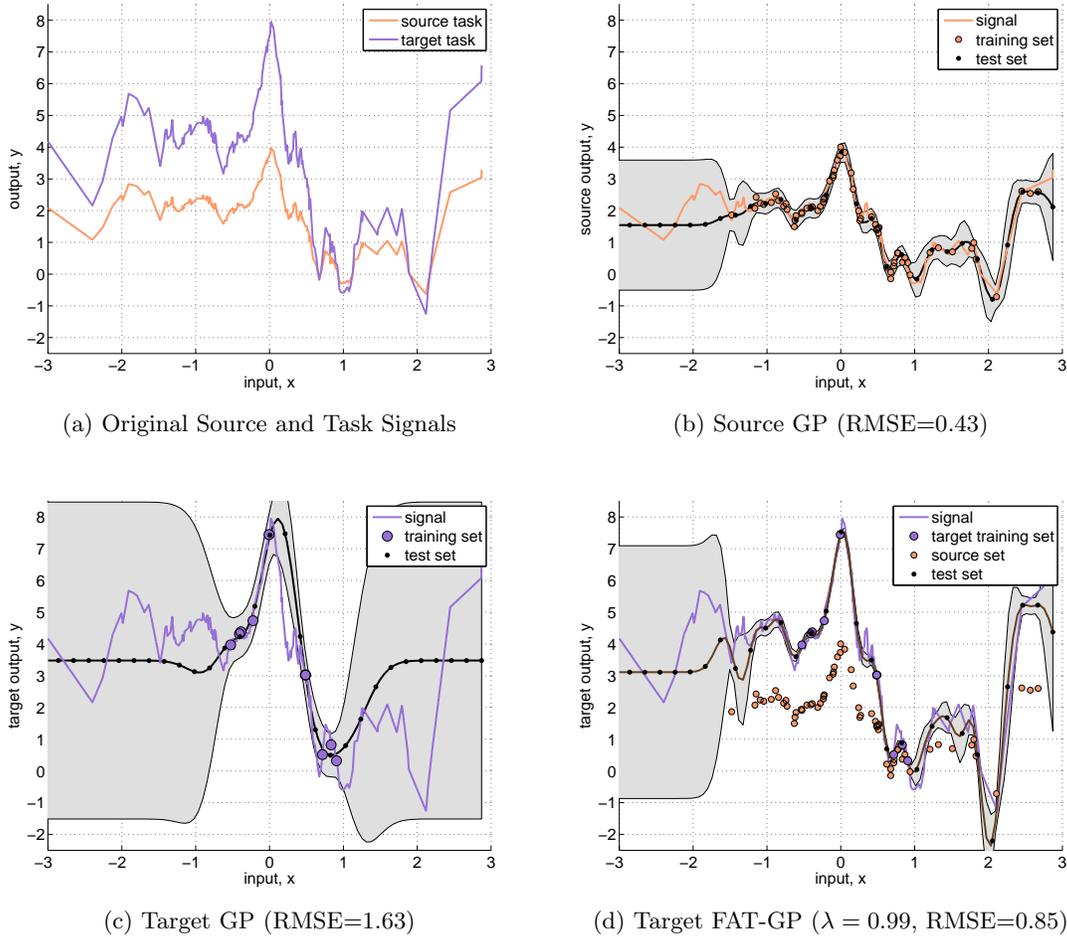


Figure 5.3: Example where target task  $T$  is a scaling of source task  $S$  i.e.  $y_T = 2y_S$

first example, and hence the validation RMSE is 0.43. Once again, the target prediction in Fig. 5.3c suffers from high variance i.e. high uncertainty for  $x \leq -1$  and  $x \geq 1$ , where no target samples are available, as well as between  $x = 0$  and  $x = 0.5$ , where there is a large gap between samples. In this case, the RMSE is 1.63. Combining these 10 target samples with the 65 source samples and  $\mathcal{GP}_S$  results in the FAT-GP shown in Fig. 5.3d, which has a much lower RMSE of 0.85.

In both examples, including data from a similar task helps the GP regression confidently predict for  $x$  values not captured in its own dataset. On one hand, this makes up for the low density of training data in regions of the target task which have been poorly sampled, for example between  $-1$  and  $1$  on the X-axis, and reduces the uncertainty in the prediction. On the other hand,

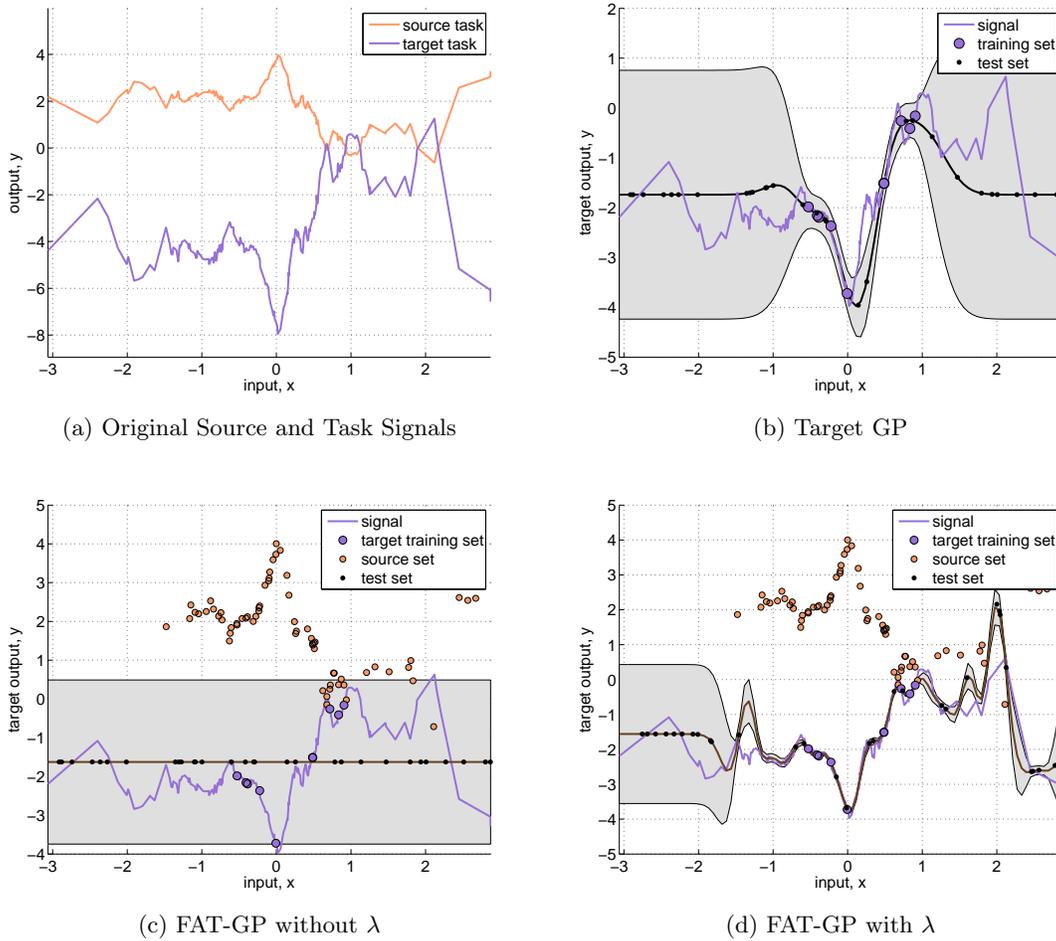


Figure 5.4: Example  $y_T = -2 \times y_S$  which demonstrates the value of  $\lambda$  in learning negative correlations

perhaps more importantly, the transfer provides the target task information about completely unexplored regions of its task space, such as the regions between 1 and 2, in both examples. As expected, the improvement does not extend to the regions where neither task has sampled. This can be seen by observing that both FAT-GPs continue to have high prediction variance between  $-3$  and  $-2$  even after the forward adaptive transfer.

The design choice of  $\lambda$  ranging from  $-1$  to  $1$  is deliberate, and meant to capture negative correlation. Fig. 5.4a shows source and target tasks where  $y_T = -2 \times y_S$ . Once again, the target GP in Fig 5.4b is learned using only 10 samples, and is a very crude model of the true signal. Limiting  $\lambda$  to positive values, results in a failed transfer as shown in Fig. 5.4c. On the other

hand, when  $\lambda$  is allowed to be in the range  $[-1, 1]$ , Fig. 5.4d learns  $\lambda = -0.9984$ . This shows that FAT-GP, as detailed in Section 5.1.1, is capable of capturing negative correlation between source and target tasks.

## 5.2 Relationship between FAT-GP and Target GP

The FAT-GP combines information from the target with transferred information from the source to learn a model of the target task. In order to understand what is transferred, how the target and transfer components interact, and when the transfer boosts the target performance, it is important to understand the relationship between FAT-GP with the conventional Target GP.

The target GP learns the target task using only the limited target task training data, contains the standard GP hyperparameters, and is trained using maximum likelihood estimation on the target data as follows

$$\begin{aligned} \text{Hyperparameters: } \theta_T &= \{\sigma_{fT}^2, \sigma_{nT}^2, L_T, \mathbf{m}_T\} \\ \text{Training: } \theta_T^* &= \arg \max_{\theta_T} \ln p(\mathbf{y}_T | X_T, \theta_T) \end{aligned} \quad (5.19)$$

Similarly, the prediction for target test sample  $\mathbf{x}'$  is given by the standard GP equations

$$\begin{aligned} \mu_T(\mathbf{x}') &= \mathbf{m}_T(\mathbf{x}') + k_T(\mathbf{x}', X_T) C_T(X_T, X_T)^{-1} (\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ \sigma_T^2(\mathbf{x}') &= k_T(\mathbf{x}', \mathbf{x}') + \sigma_{nT}^2 - k_T(\mathbf{x}', X_T) C_T(X_T, X_T)^{-1} k_T(X_T, \mathbf{x}') \end{aligned}$$

This GP provides a baseline for the performance that can be obtained without transfer. On the other hand, Forward Adaptive Transfer Learning i.e. FAT-GP ( $\tau$ ) uses *a priori* source hyperparameters, and then learns  $\theta_{\tau(T)}$  and  $\lambda$  using source and target data.

### 5.2.1 Impact of Transfer on Mean and Variance Prediction

To investigate the benefits of transfer, we analyse how the mean predictions from both these GPs compare. As shown in Appendix B.1, using properties of positive semidefinite matrices, and equations defined in Section 5.1, we find that the FAT-GP mean prediction comprises of two

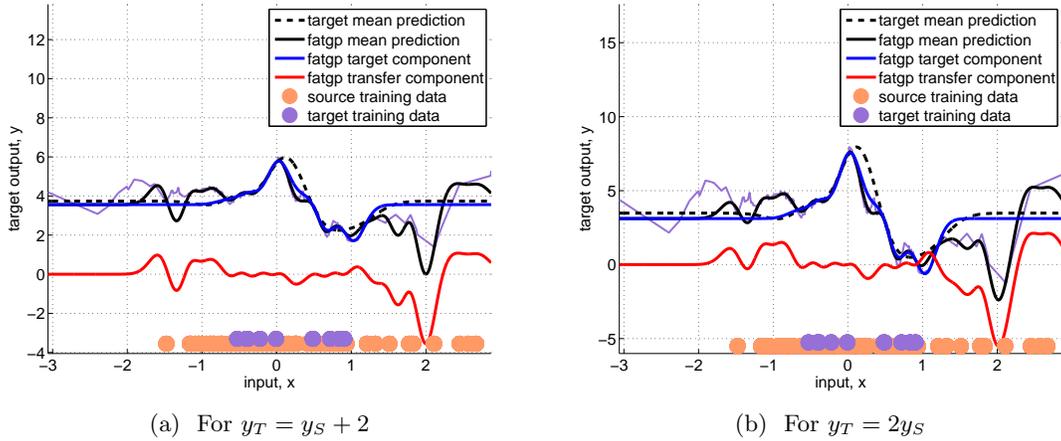


Figure 5.5: FAT-GP mean prediction components, and its comparison to Target GP

components.

$$\mu'_\tau = \underbrace{\begin{pmatrix} \mathbf{m}_\tau(\mathbf{x}') + \\ k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_\tau(X_T)) \end{pmatrix}}_{\text{Target component}} + \underbrace{\begin{pmatrix} [k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS} - k_\tau(\mathbf{x}', X_S)] \times \\ [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1} \times \\ [K_{ST}C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_\tau(X_T)) - (\mathbf{y}_S - \mathbf{m}_S(X_S))] \end{pmatrix}}_{\text{Transfer component}} \quad (5.20)$$

Fig. 5.5 shows these target and transfer components as blue and red solid lines, respectively. The mean predictions for FAT-GP and target GP are also shown using solid and dashed black lines. The blue target component is very close to the dashed black target GP prediction, since they both represents the information that is available in the target training set, shown by purple dots at the bottom of the figure. The red transfer component is primarily responsible for the differences between the FAT-GP and the Target GP. Note that its major contributions are in regions where no target samples are available; the gaps are filled by transferring from the source GP. Finally, for  $x \leq -2$ , where neither source nor target data is available, the FAT-GP cannot provide meaningful predictions.

Analogous to Equation 5.20, the variance prediction of the FAT-GP also comprises of two

components,

$$\sigma_\tau^2(\mathbf{x}') = \underbrace{\begin{pmatrix} k_\tau(\mathbf{x}', \mathbf{x}') + \sigma_{n\tau(T)}^2 \\ k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}k_\tau(X_T, \mathbf{x}') \end{pmatrix}}_{\text{Target Component}} - \underbrace{\begin{matrix} [k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS} - k_\tau(\mathbf{x}', X_S)] \times \\ [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1} \times \\ [K_{ST}C_{TT}^{-1}k_\tau(X_T, \mathbf{x}') - k_\tau(X_S, \mathbf{x}')] \end{matrix}}_{\text{Transfer Component}} \quad (5.21)$$

which can also be viewed as

$$\begin{aligned} \sigma_\tau^2(\mathbf{x}') = & \underbrace{\sigma_{f\tau(T)}^2 + \sigma_{n\tau(T)}^2}_{\text{prior variance}} - \underbrace{k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}k_\tau(X_T, \mathbf{x}')}_{\text{Reduction in uncertainty due to target data}} \\ & - \underbrace{[k_\tau(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS} - k_\tau(\mathbf{x}', X_S)] [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1} [K_{ST}C_{TT}^{-1}k_\tau(X_T, \mathbf{x}') - k_\tau(X_S, \mathbf{x}')] }_{\text{Reduction in uncertainty due to transfer}} \end{aligned} \quad (5.22)$$

The perspectives of Equations (5.21) and (5.22) are illustrated in Figures 5.6 and 5.7, respectively. Note that in Fig. 5.6, the FAT-GP variance prediction represented by the black solid line is obtained by subtracting the red solid line from the blue solid line. Since the blue line represents the target component of the variance prediction or uncertainty, it is low or zero where target samples are available, and rises to  $\sigma_{f\tau(T)}^2 + \sigma_{n\tau(T)}^2$  elsewhere. The transfer component shown by the red line represents a reduction in uncertainty due to transferring, and analogously provides zero correction where no source data is available.

More interestingly, however, it also provides zero correction where target samples are present. This alternating behavior is more clearly apparent in Fig. 5.7 in which the components from Equation (5.22) are represented by flat dashed blue, solid blue, and solid red lines, respectively. Here, the red and blue lines are both subtracted from the dashed blue line, reducing the uncertainty, to give the solid black line. Thus, the reduction in uncertainty from the transfer (shown in red) counters only the uncertainty not corrected by the target data itself.

Analyzing how transfer behaves based on the relative positions of the target and source data samples reveals a link to the signal-to-noise ratio (SNR) of the target task. As shown in Appendix B.2, how a source sample  $\mathbf{x}_S$  and a target sample  $\mathbf{x}_T$  reduce the uncertainty at an unseen sample

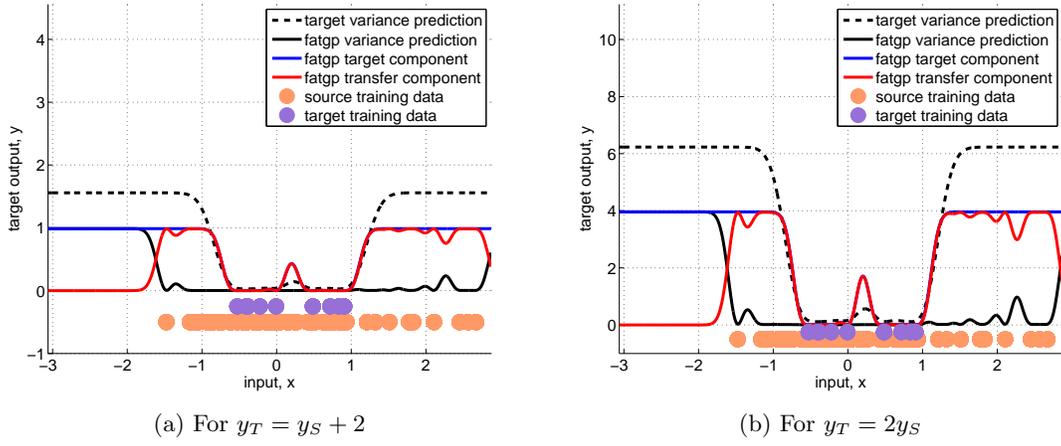


Figure 5.6: FAT-GP variance prediction components and its comparison to Target GP

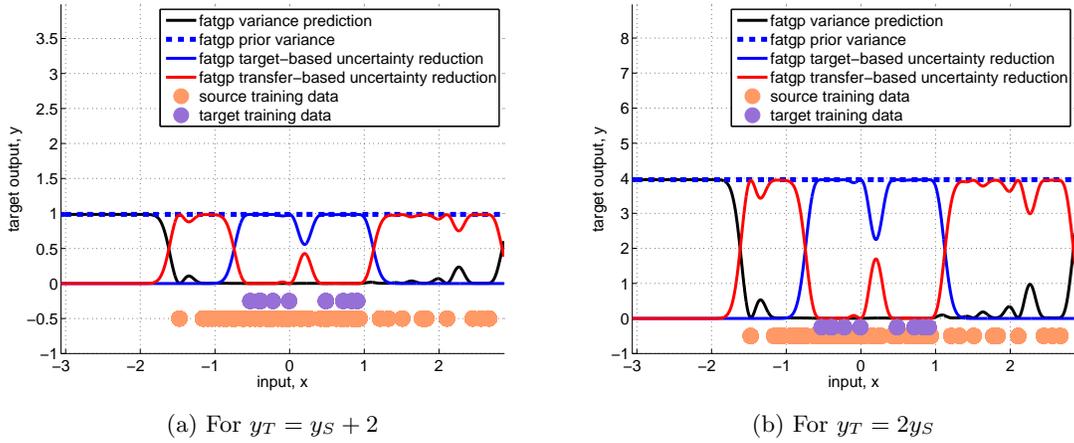


Figure 5.7: Reduction of uncertainty in FAT-GP

$\mathbf{x}'$  depends on their relative positions, and is captured in the coefficient of transfer  $b$  is given by

$$b = \begin{cases} \frac{\sigma_{n\tau(T)}^2}{\sigma_{n\tau(T)}^2 + \sigma_{f\tau(T)}^2} & \text{when source and target are colocated at } \mathbf{x}' \\ \frac{c_{TS}c_\tau(T)\sigma_{f\tau(T)}^2}{\sigma_{n\tau(T)}^2 + \sigma_{f\tau(T)}^2} & \text{when target is close to } \mathbf{x}' \\ -c_S & \text{when only a source is close to } \mathbf{x}' \end{cases} \quad (5.23)$$

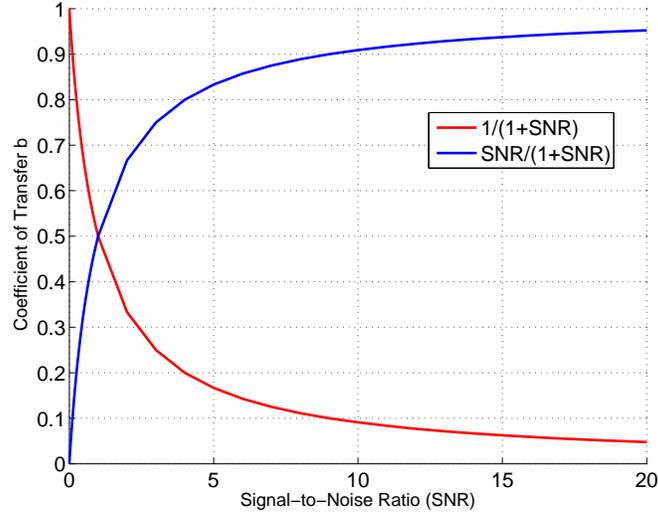


Figure 5.8: Coefficient of transfer  $b$  as a function of SNR

where

$$\begin{aligned}
 c_{TS} &= \exp \left[ -\frac{\|\mathbf{x}_T - \mathbf{x}_S\|^2}{l_{\tau(T)}^2 + l_S^2} \right] \\
 c_S &= \exp \left[ -\frac{\|\mathbf{x}' - \mathbf{x}_S\|^2}{l_{\tau(T)}^2 + l_S^2} \right] \\
 c_{\tau(T)} &= \exp \left[ -\frac{\|\mathbf{x}_T - \mathbf{x}'\|^2}{2l_{\tau(T)}^2} \right]
 \end{aligned}$$

Note that here *close to* implies being within the distance of correlation.

Because  $\sigma_{f_{\tau(T)}}^2$  and  $\sigma_{n_{\tau(T)}}^2$  represent the signal and noise variance respectively,  $\frac{\sigma_{f_{\tau(T)}}^2}{\sigma_{n_{\tau(T)}}^2}$  represents the signal-to-noise ratio (SNR) of the target portion of the FAT-GP. Hence, the first two cases in Equation (5.23) can also be viewed as

$$b = \begin{cases} \frac{1}{1+SNR} & \text{when source and target are colocated at } \mathbf{x}' \\ \frac{c_{TS}c_{\tau(T)}SNR}{1+SNR} & \text{when target is close to } \mathbf{x}' \end{cases} \quad (5.24)$$

Under this perspective, as shown in Fig. 5.8, when  $SNR < 1$ , the source will contribute heavily to predictions even when target data is available in the same region. On the other hand, when  $SNR > 1$ , for co-located source and target datapoints, the contribution is muted with growing SNR. As the distance between the source datapoint and  $\mathbf{x}'$  grows, the influence of the transfer is

routed via the target datapoints close to  $\mathbf{x}'$ , and dictated by a combination of the SNR and the correlation between the tasks.

### 5.2.2 Dissimilar source and target tasks

When the target and source are dissimilar i.e.  $\lambda = 0$ , all cross covariance terms go to zero. Then the difference between the two GPs' mean predictions is given by

$$\mu'_\tau = \mathbf{m}_{\tau(T)}(\mathbf{x}') + k_{\tau(T)}(\mathbf{x}', X_T) C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_{\tau(T)}(X_T)) \quad (5.25)$$

Comparing this against (5.20), we see that Equation (5.25) has the same form as the mean prediction for the target GP. These predictions would be equal if  $\theta_{\tau(T)} = \theta_T$  i.e. target hyperparameters learned in both GPs are identical. To investigate this we revisit the log marginal likelihood that the FAT-GP maximizes.

$$\theta_\tau^* = \arg \max_{\theta_T} -\frac{1}{2}(\mathbf{y}_T - \mu_{T|S}) C_{T|S}^{-1}(\mathbf{y}_T - \mu_{T|S}) - \frac{1}{2} \ln |C_{T|S}| - \frac{N_T}{2} \ln 2\pi \quad (5.26)$$

When  $\lambda$  is set to zero, from Equation (5.16) we see that  $K_{TS} = \mathbf{0} = K_{ST}$ , and this maximization becomes

$$\theta_\tau^* = \arg \max_{\theta_T} -\frac{1}{2}(\mathbf{y}_T - \mathbf{m}_T)^T C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T) - \frac{1}{2} \ln |C_{TT}| - \frac{N_T}{2} \ln 2\pi \quad (5.27)$$

which is the maximization used by the target GP to learn the hyperparameters. Therefore, the remaining hyperparameters  $\theta_{\tau(T)}$  that are learned are the same as  $\theta_T$  learned by the target GP. Hence, the difference between their mean predictions is given by

$$\mu'_\tau - \mu'_T = 0 \quad (5.28)$$

Thus, when the target and source task have no similarity i.e.  $\lambda = 0$ , the learned FAT-GP is in fact the same as the target GP.

### 5.3 Comparing the FAT-GP and Target GP

#### 5.3.1 RMSE and MSLL Comparisons

Using an obsolete or incorrect model can have adverse effects on performance of the mission. This section uses root mean squared error (RMSE), and mean standardized log loss (MSLL) to quantify how different learning schemes involving source and target task compare to the FAT-GP. We enumerate these different learning configurations here

- (1) Source GP (S): Only the source data  $\{X_S, \mathbf{y}_S\}$  is used for training and prediction. This configuration maps to the scenario where a previously learned GP model is considered plausible for the current mission and used as is. Comparing this configuration to FAT-GP will help illustrate the drawbacks of using outdated models.

$$\begin{aligned}
 \text{Hyperparameters: } \theta_S &= \{\sigma_{f_S}^2, \sigma_{n_S}^2, L_S, \mathbf{m}_S\} \\
 \text{Training: } \theta_S^* &= \arg \max_{\theta_S} \ln p(\mathbf{y}_S | X_S, \theta_S) \\
 \text{Prediction: } \mu'_S &= \mathbf{m}_S(\mathbf{x}') + k_S(\mathbf{x}', X_S) C_S(X_S, X_S)^{-1} (\mathbf{y}_S - \mathbf{m}_S(X_S)) \quad (5.29) \\
 C'_S &= k_S(\mathbf{x}', \mathbf{x}') + \sigma_{n_S}^2 - k_S(\mathbf{x}', X_S) C_S(X_S, X_S)^{-1} k_S(X_S, \mathbf{x}')
 \end{aligned}$$

- (2) No retraining (N): While only the source data is used for learning the hyperparameters during training, the prediction combines these learned hyperparameters with source and target data  $X_D = \{X_T, X_S\}$ . This configuration represents the scenario where data collected after training is added to the training set under the assumption that the underlying distribution is unchanged. Comparing this configuration with FAT-GP will help quantify the value of knowing the task assignment for the datasets.

$$\begin{aligned}
 \text{Hyperparameters: } \theta_S &= \{\sigma_{f_S}^2, \sigma_{n_S}^2, L_S, \mathbf{m}_S\} \\
 \text{Training: } \theta_S^* &= \arg \max_{\theta_S} \ln p(\mathbf{y}_S | X_S, \theta_S) \\
 \text{Prediction: } \mu'_N &= \mathbf{m}_S(\mathbf{x}') + k_S(\mathbf{x}', X_D) C_S(X_D, X_D)^{-1} (\mathbf{y}_D - \mathbf{m}_S(X_D)) \quad (5.30) \\
 C'_N &= k_S(\mathbf{x}', \mathbf{x}') + \sigma_{n_S}^2 - k_S(\mathbf{x}', X_D) C_S(X_D, X_D)^{-1} k_S(X_D, \mathbf{x}')
 \end{aligned}$$

- (3) All data training (D): As mentioned in Section 5.1.1, this configuration ignores separation between the  $X_T$  and  $X_S$ , and uses  $X_D = [X_T^T, X_S^T]^T$  for learning hyperparameters. Like the no-retraining (R) configuration, this also helps quantify the value of knowing the task assignment, but this incurs an additional training overhead.

$$\text{Hyperparameters: } \theta_D = \{\sigma_f^2, \sigma_n^2, L, \mathbf{m}_D\} \quad (5.31)$$

$$\text{Training: } \theta_D^* = \arg \max_{\theta_D} \ln p(\mathbf{y}_D | X_D, \theta_D)$$

$$\text{Prediction: } \mu'_D = \mathbf{m}_D(\mathbf{x}') + k_D(\mathbf{x}', X_D) C_D(X_D, X_D)^{-1} (\mathbf{y}_D - \mathbf{m}_D(X_D))$$

$$C'_D = k_D(\mathbf{x}', \mathbf{x}') + \sigma_{nD}^2 - k_D(\mathbf{x}', X_D) C_D(X_D, X_D)^{-1} k_D(X_D, \mathbf{x}')$$

The learning configurations were compared by learning models for the 1D examples presented in Figures 5.2 and 5.3. Each learner was provided with the same data: source GP, 65 source training samples, and 10 target training samples. The experiment was repeated 20 times. Table 5.2 presents RMSE and MSLL for each of the learning configurations, averaged over 20 runs. Since the source GP makes no use of target data, the MSLL values were calculated with the source GP as the baseline. All other configurations were expected to improve over this baseline i.e. have negative MSLL values.

Table 5.2: FAT-GP vs Other GP Configurations

Learning configuration	$y_T = y_S + 2$		$y_T = 2y_S$	
	RMSE	MSLL	RMSE	MSLL
Source GP (S)	2.06	0	1.83	0
No retraining (R)	1.99	-4.62	1.75	-6.41
Target GP (T)	0.75	-43.32	1.42	-38.18
FAT-GP ( $\tau$ )	0.59	-40.16	0.57	-38.79
All data training (D)	1.95	-41.62	1.78	-36.97

Examining the MSLL values in the table shows that ignoring target data, as in the source GP case, results in the worst performance. While the no-retraining (N) case improved over the source GP, models that learned hyperparameters from the target data did significantly better than the S and N configurations. While configurations T,  $\tau$ , and D had comparable MSLL values, FAT-GP reduced the RMSE over the other learners.

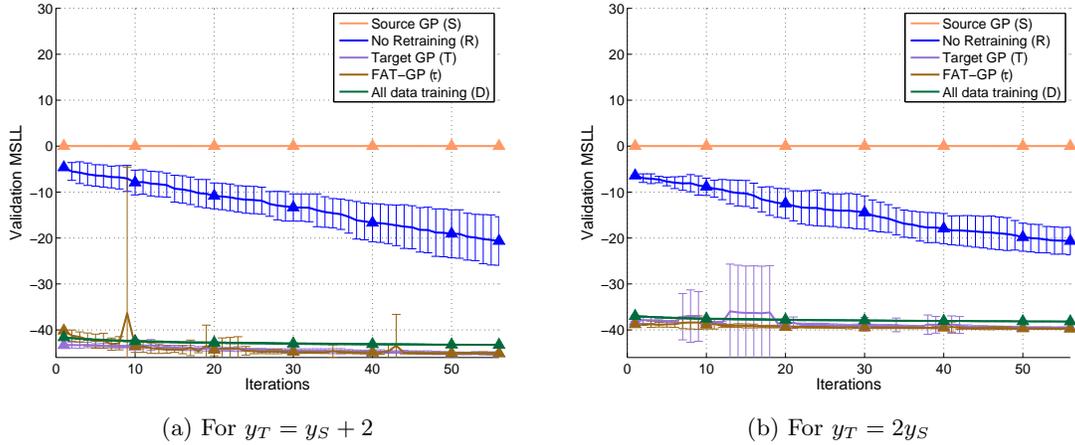


Figure 5.9: Effect of Target Data Size on MSLL for FAT-GP and other learning configurations

### 5.3.2 Effect of Target Data Size on Transfer

To examine the effect of increasing target training data, the FAT-GP and the other learners was trained iteratively. Starting with 10 training samples from the target task at iteration 1, a new random target training sample was added at each iteration. This was repeated for 56 iterations till both the source and target task had 65 measurements. The performance of the learners at each iteration was evaluated against a fixed validation set.

Fig. 5.9 shows the MSLL and RMSE results of the 5 learners over 56 iterations. Increasing the target data has no impact on the source GP, which shows up as a flat (orange) line in all 4 plots. In the case of configuration N, as the proportion of the target data goes up, it begins to capture the statistics of the target task, gradually improving its MSLL. However, as in Section 5.3.1, configurations T,  $\tau$ , and D outperform S and N in terms of MSLL. Even so, Fig. 5.10 shows that the RMSE of learner D is close to that of N, with the Target GP and FAT-GP achieving a much lower RMSE by the last iteration.

Figures 5.11 and 5.12 take a closer look at how the Target GP and FAT-GP compare. While the FAT-GP's MSLL is marginally better than the Target GP, Fig. 5.12 shows that the FAT-GP has a lower RMSE compared to Target GP. In fact, in both examples, the Target GP RMSE at the last iteration is reached by FAT-GP around iteration 20. This highlights how FAT-GP can provide

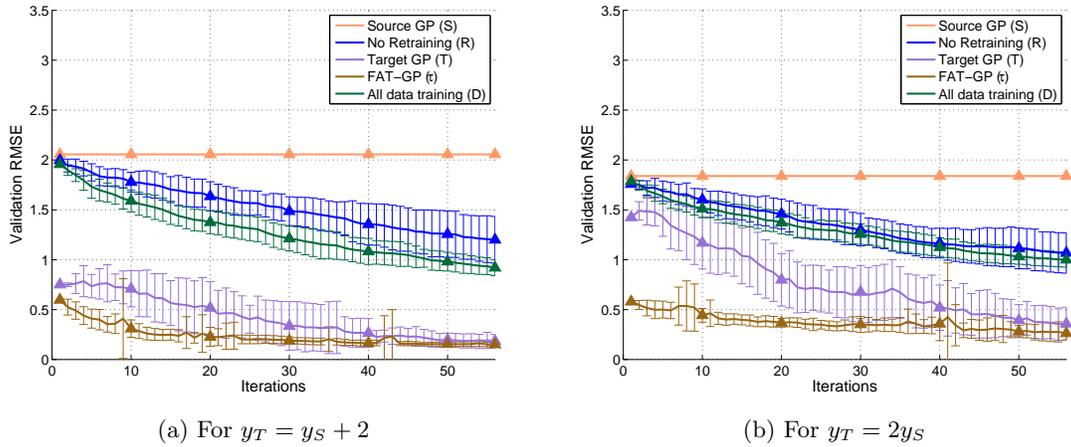


Figure 5.10: Effect of Target Data Size on RMSE for FAT-GP and other learning configurations

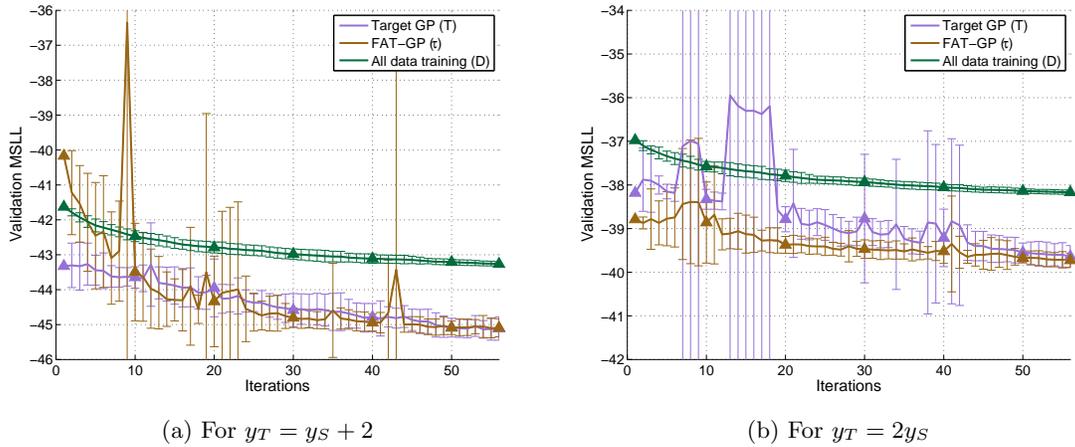


Figure 5.11: FAT-GP vs Target GP MSLL Comparison

an efficient interim model while more target task data is being collected.

As the target task's training set grows beyond 40, the big divide between the RMSE performance of the Target and FAT GP begins to narrow. As the information in the target training set increases, the need to transfer knowledge decreases. This is reflected in Fig. 5.13, which shows change in the mean components (discussed in Section 5.2.1) between the first and last iteration. Top row reproduces the plots from Fig. 5.5 for comparison with those in the bottom row, which are from the last iteration. Compared to iteration 1, the target GP (black dashed), FAT-GP (black solid), and FAT-GP's target component (blue solid) all closely match the (purple) true target, while

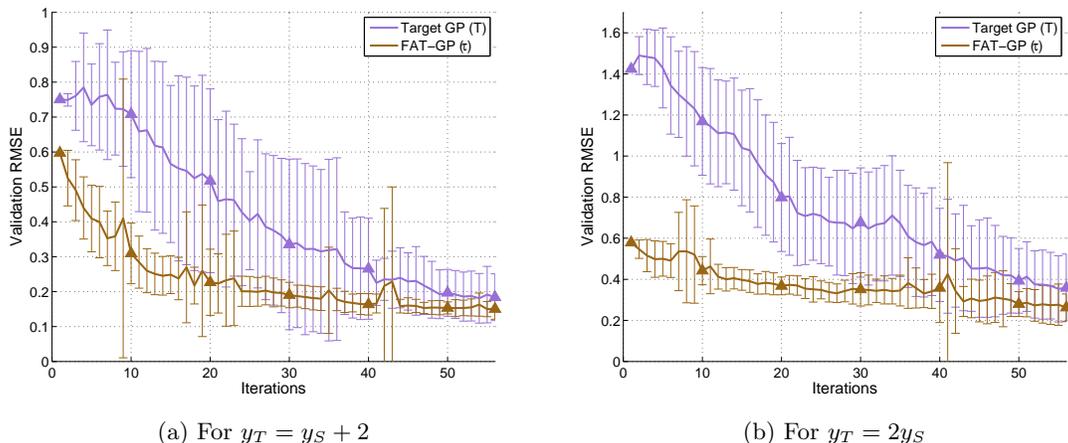


Figure 5.12: FAT-GP vs Target GP RMSE Comparison

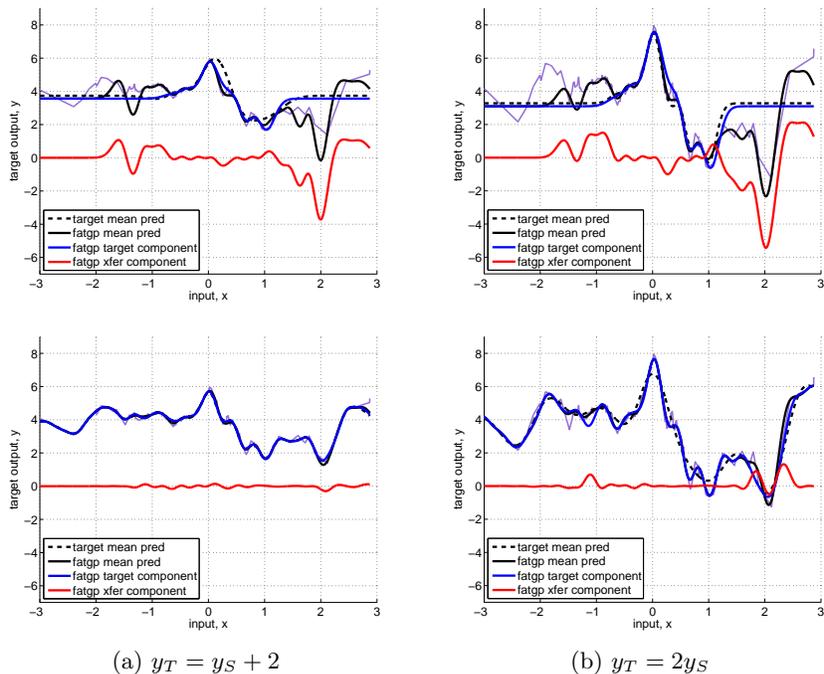


Figure 5.13: Evolution of FAT-GP’s mean components from iteration 1 with 10 target samples (shown in top row) to iteration 56 with 65 samples (shown in bottom row)

the (red) transfer component is mostly a flat line. On the lines of the SNR discussion in Section 5.2.1, as the noise in the data goes down and SNR grows, the contribution of transfer greatly diminishes.

## 5.4 FAT-GP for Transfer Learning Communication Models

This section uses a simulation study to illustrate how FAT-GP would be useful in learning communication models in the presence of little training data, and limited exploration.

The source and target task are setup such that

$$\mathbf{y}_S = f(X_S) + v_S \quad (5.32)$$

$$\mathbf{y}_T = g(f(X_T)) + v_T \quad (5.33)$$

where the first term represents the mean field, and the second term represents the geospatial variations. The variations  $v_S$  and  $v_T$  are sampled from the same distribution  $\mathcal{N}(0, K_{variation})$ . In Figure 5.14, which show the source and target tasks, their mean fields, and variations,  $g(\cdot) = 2 \times (\cdot)$ . The source GP is learned using a dense set of 2500 training samples as shown in Figure 5.15a. The

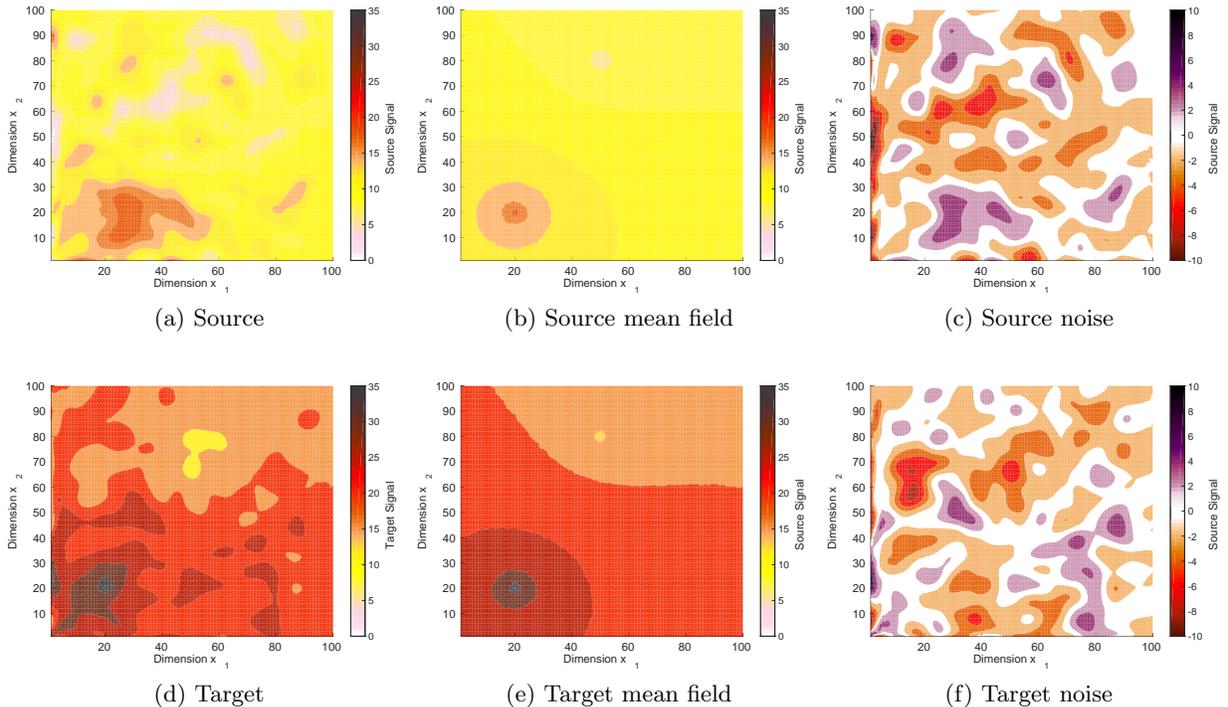


Figure 5.14: True environments in source and target tasks

mean prediction of this source GP achieves a high fidelity with the original source task, which can be seen by comparing Fig 5.15b with Fig. 5.14a.

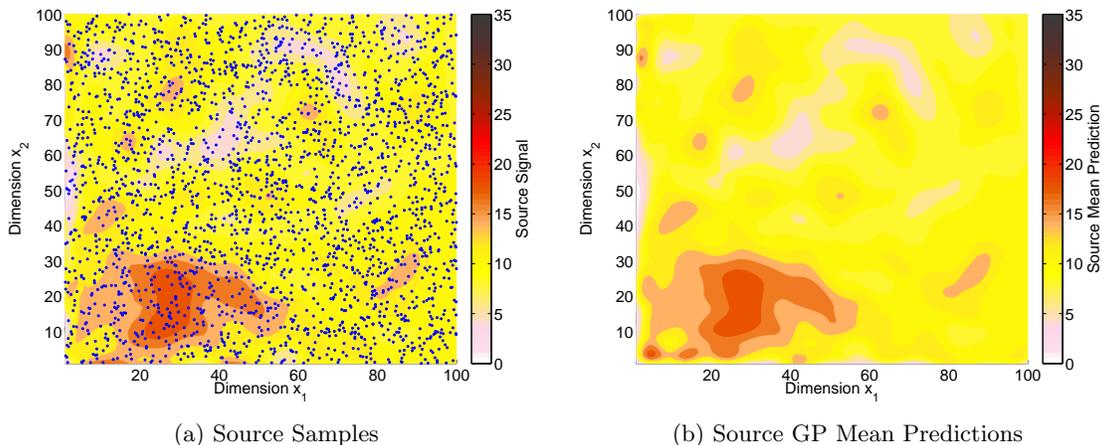


Figure 5.15: Source GP learned using 2500 samples of the source task

Figure 5.16 presents two scenarios for learning the target task with limited training set of size 50. In Fig. 5.16a the samples are spread over the entire environment, representing a situation

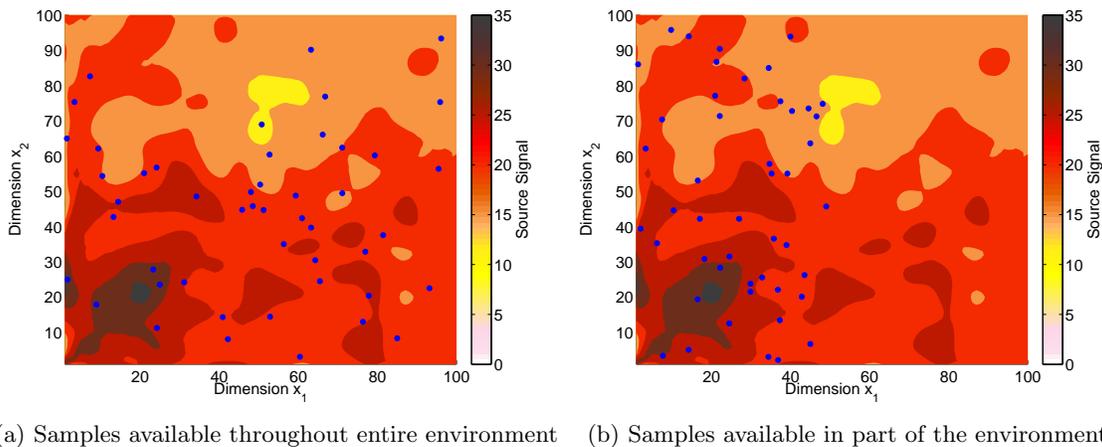


Figure 5.16: Two scenarios with limited target training set of size 50 available.

where a low density of data is available for the target task. Fig. 5.16b, on the other hand, represents the situation where due to the large scale of environments in robotic missions, target data is only available for part of the environment. In both these cases, FAT-GP can leverage the high fidelity source GP, which has captured inherent characteristics of the environment, and learn better models until more target data can be collected.

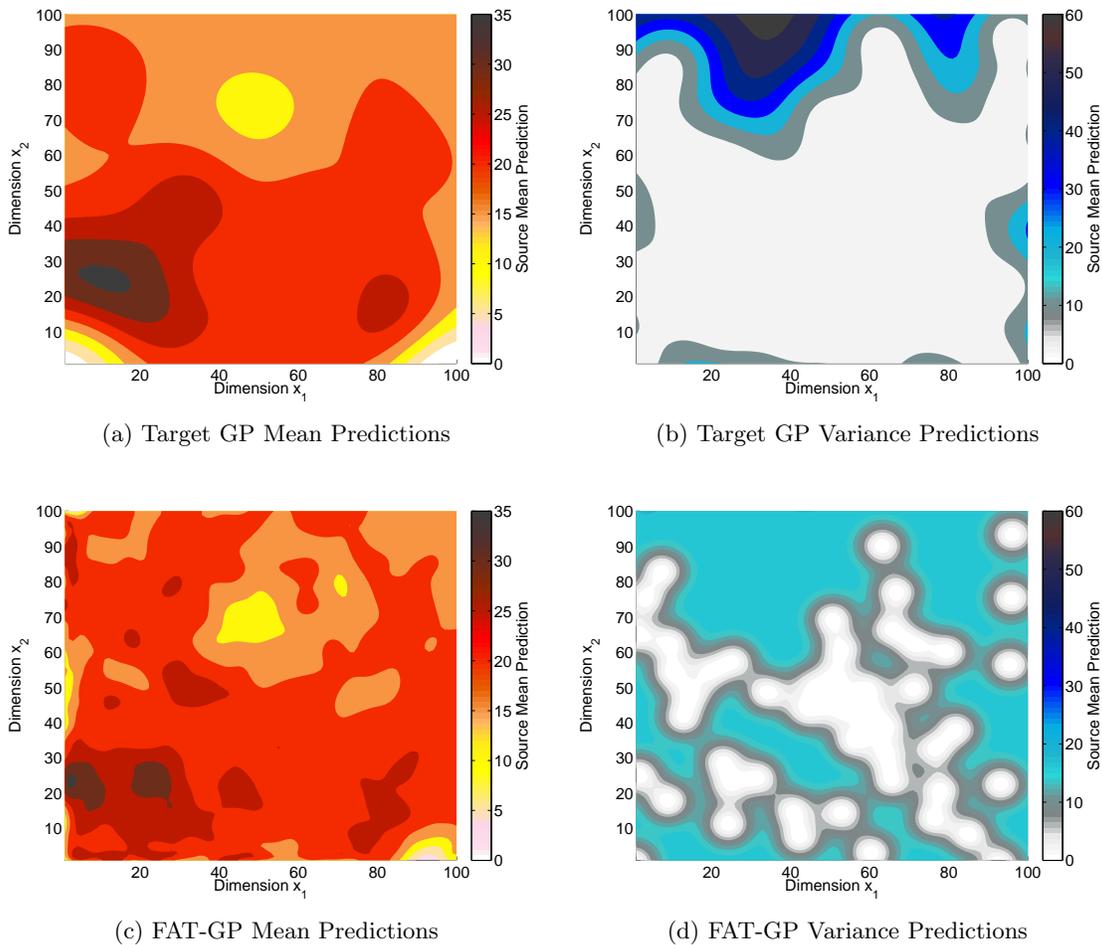


Figure 5.17: Target GP and FAT-GP learned using 50 samples of the target task taken over the entire environment

Figures 5.17 and 5.18 show a comparison of the Target GPs and FAT-GPs learned using the two datasets shown in Fig. 5.16. In the first example, where samples are available throughout the entire environment, the Target GP shows an RMSE of 3.58. In spite of the high error, its confidence is high in most of the region. Due to similarity between the tasks, FAT-GP learns a value of  $\lambda = 0.94$  during the training. Using the source GP and source data in its predictions, the FAT-GP reduces the RMSE to 2.97. Seeing the variability of the larger source dataset also causes the FAT-GP to adapt its uncertainty, so that it has very high confidence only locally around the target samples. The resulting predictive distribution outperforms the over-confident Target GP. Consequently, MSL of the FAT-GP with the target GP as the baseline is  $-0.2276$ .

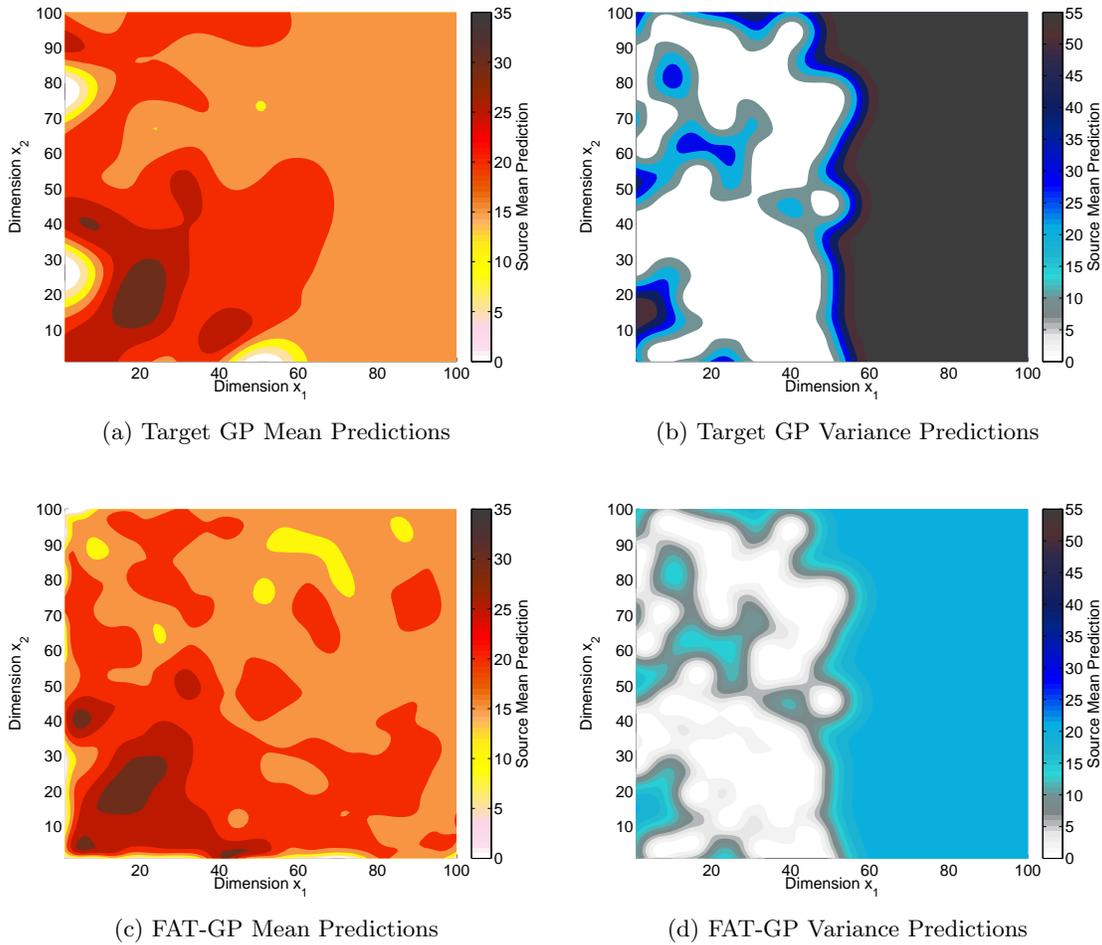


Figure 5.18: Target GP and FAT-GP learned using 50 samples of the target task taken over half the environment

In the second example, because the environment is observed only partially, the RMSE and variance predictions are higher than in the first example. The higher RMSE of 5.75 is mostly due to the right half of the environment, where the target GP predicts a flat field with value equal to mean of the observed  $y_T$ . As this region is outside the distance of correlation for all the training data, it also has the highest variance. The FAT-GP, which learns  $\lambda = 0.96$  and transfers source data proportionally, reduces the uncertainty of the unobserved half of the region, and achieves an RMSE of 4.46. Once again, because it outperforms the over confidence target GP in terms of accuracy and uncertainty, the MSL of the FAT-GP with respect to the target GP is  $-0.11$ .

In both these case, the FAT-GP makes use of the correlation between the source and target

task to make up for the limited information in the small target dataset. In this manner, previous observations and models of the environment can help inform model updates, especially as the environment is being explored for updated observations. However, efficient and beneficial transfer largely depends on the locations at which target samples are taken. The choice of target locations not only affects the information they provide about the task itself, but also helps learn the correlations between the tasks effectively. Hence, future work must focus on *transfer planning*, which is active learning based on the combined objectives of transfer and target learning.

## 5.5 Summary

This chapter proposes a forward adaptive transfer learning method, FAT-GP, which allows robots to leverage previously learned Gaussian process regression models and use them as sources of information in new learning tasks. This is especially valuable when limited training data is available for the new target task. FAT-GP decouples the kernel and hyperparameter selection for the target task from those of the source task, providing an inference framework that is desirable when dealing with real world, dynamic environments. Additionally, because the source task’s large covariance matrix is precomputed, FAT-GP amortizes cost and is computationally cheaper than AT-GP and other GP approaches using transfer kernels. Simulations studies on 1D and 2D examples show that similar source tasks can considerably improve the target’s performance. More importantly, the FAT-GP exploits the correlations between the source and target to achieve a low error with much less target data, and can thus serve as an efficient model in the interim as more target data is collected through exploration.

## Chapter 6

### Conclusion

#### 6.1 Spatiotemporal Characterization of Airborne RF Communication

Chapter 3 presented a nonparametric Gaussian process based approach for learning the environment specific residuals or RF variations for an *a priori* model such as the empirical path loss model. In addition, an inexpensive pre-processing step involving outlier detection was used to obtain models with better physical interpretability.

The improvements in performance obtained by augmenting spatial isotropic, spatial ARD, and spatio-temporal ARD GPs to the empirical model were evaluated by performing cross validation experiments on 50 datasets of signal strength measurements, collected using two different radio frequencies, three different airframes, and employing static or mobile transmitters. These comparisons showed that GPs not only learn receiver dependent noise but also help identify environmental artifacts like localized interference.

The assessment of prediction error as well as the model uncertainty using metrics RMSE and MSLI illustrated that the GPs build improved models by learning time-varying and geospatial characteristics of the environment. Spatio-temporal GPs further reduced the error and uncertainty compared to their spatial counterparts.

#### 6.2 GP Modeling for Data Ferrying and Other UAS Applications

Chapter 4 addressed the issue of uncertain RF communication environments when planning ferry routes. Errors between an **a priori** model and the true environment were sampled as the air-

craft ferries data. These errors were fed to a Gaussian process to better predict the RF environment elsewhere, leading to more accurate plans for the ferry’s path.

The simulation studies showed the adaptability of the GP in comparison to parameter-based estimators. Through relatively clean RF environments where the parameter-based methods perform well, the GP performed just as well and converged just as quickly. In more complicated environments, the non-parametric nature of the GP allowed it to more accurately capture complex RF behaviors in the ferrying region between the sensor nodes. Further, the quick convergence of the ferry-and-learn process to improve RF models enabled the ferry to plan better paths and increase effective throughput with the nodes.

### 6.3 Forward Adaptive Transfer Learning

Chapter 5.3.2 presented a new inference framework which allows GP for new target learning tasks to use previously learned GP models for transfer of knowledge. By using convolution of kernels and task correlations for cross covariance terms, FAT-GP decouples the two tasks and allows them to have different kernel functions and hyperparameters. 1D examples demonstrated how forward adaptive transfer allowed FAT-GP to reduce both the prediction error and the uncertainty of the target task by transferring from similar, correlated source task.

Investigation of transfer components of the mean and variance predictions of the FAT-GP showed that the influence of the source on the predictions was directly linked to the signal-to-noise ratio (SNR) of the target components. This was further demonstrated in experiments that iteratively increased the target training set, where the increase in target task knowledge i.e. an increase in SNR corresponded with diminishing contribution from transfer.

The FAT-GP was compared to other learning configurations including the target GP using RMSE and MSLL and found to provide a better model with reduced prediction error. In the iterative experiments, the low error shown by the target GP after 56 iterations was attained by the FAT-GP in 20 iterations. This highlighted how FAT-GP can provide an efficient interim model while more target task data is being collected. 2D results further emphasized how FAT-GP learn

models for the entire environment using limited, sparse, or localized measurements.

## 6.4 Future Work

### 6.4.1 Explore-vs-Exploit for Spatio-temporal GPs

To reduce the uncertainty in environments, it is important to explore and take measurements throughout to capture the diversity in the environment. Since localized sampling provides diminishing returns, yet drives up the training time, it is important to choose these locations carefully. Previous work has demonstrated how active learning can be used to set up an explore-and-exploit paradigm for GP learning in sensor networks, ground robots, and underwater vehicles. However, exploration policies for spatio-temporal GP models onboard small unmanned aircraft will have to additionally take vehicle dynamics and temporal variations into consideration.

This thesis also presented results for mobile transmitters, which increases dimensionality of the GP's input variable. The solution to this variation of the communication modeling problem can be greatly improved by planning the UA trajectory to increase observability of the measurements as well as reduce the uncertainty in the model.

The planning objective may be further complicated by competing or even conflicting objectives of the UAS mission. For example, opportunistic communication learning for data ferrying, as presented in Chapter 4, used the trajectories planned for the ferrying objective. Under the mission's planning objective, regions of overestimated signal strength would inherently get explored and corrected, but the opposite case of finding a strong signal in an expected weak region would not occur in this framework. Thus, lack of exploration could be a significant hindrance. A challenging open question lies in how to combine UAS mission objectives with the active learning objective for improving the model.

### 6.4.2 Transfer Planning

The success of transfer learning hinges to a great extent of the relative locations at which source and task measurements are available, which help ascertain the degree to which the tasks are correlated. Active learning i.e. selecting which locations to sample for the target task can result in a quick convergence of the hyperparameter  $\lambda$ , and help further reduce the uncertainty in the model. Planning how the target task is sampled will also provide a controlled framework for analyzing the PAC (probably approximately correct) guarantees of FAT-GP. Future work should investigate how to design an objective function that balances the learning and transfer objectives.

### 6.4.3 Heteroscedastic GPs

While Gaussian processes presented in this thesis provide great way to capture geospatial variations, they assume homoscedasticity i.e. the noise variance  $\sigma_n^2$  is the same everywhere in the environment. UKF-based radio source localization using such homoscedastic GPs showed good results for cleaner frequencies like  $433MHz$ , but was plagued by filter divergence for noisier RF communications using  $2.4GHz$ . Relaxing this assumption and learning heteroscedastic GPs may be critical to learning models for such missions with higher precision requirements. Future work should assess how heteroscedastic models compare to their homoscedastic counterparts.

### 6.4.4 Application of FAT-GP to other robotic missions

Due to their flexibility and ability to handle uncertainty, GPs are becoming increasingly ubiquitous in robotic learning tasks. They have been used for monitoring environmental phenomena [29, 39], mapping the terrain in which the robot navigates [62, 34], modeling motion control and dynamics [37, 15], as well as tracking dynamic obstacles [30]. They have also been employed for communication modeling and energy harvesting [42, 63] for robots in terrestrial, aerial, and aquatic environments. In each of these applications, changes in the underlying environment or setup can require that a new, updated GP be learned. In the cases where the new learning task has overlap with the previously learned model, FAT-GP can be used to adapt to the changes with a relatively

smaller set of measurements. Future work should assess how FAT-GP can impact the performance of other robotic missions which encounter dynamic environments and setups. The result of applying FAT-GP to these diverse robotic missions, will also highlight FAT-GP's limitations, which can be used for further improving its transfer learning framework.

## Bibliography

- [1] Andrew Arnold, Ramesh Nallapati, and William W. Cohen. A Comparative Study of Methods for Transductive Transfer Learning. In Seventh IEEE International Conference on Data Mining Workshops (ICDMW), pages 77–82, October 2007.
- [2] Paramvir Bahl and Venkata N Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies., volume 2, pages 775–784, 2000.
- [3] Deepak Bhaduria and Volkan Isler. Data gathering tours for mobile robots. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3868–3873, October 2009.
- [4] Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer New York, 2009.
- [5] Black Swift Technologies. SwiftPilot Avionics System. <http://www.blackswifttech.com/products.html>.
- [6] Timothy X Brown, Brian M Argrow, Eric W Frew, Cory Dixon, Daniel Henkel, Jack Elston, and Harvey Gates. Experiments Using Small Unmanned Aircraft to Augment a Mobile Ad Hoc Network, chapter 28, pages 123–145. Number ISBN-13: 9780521895842. 2007.
- [7] Bin Cao, Sinno Jialin Pan, Yu Zhang, Dit-Yan Yeung, and Qiang Yang. Adaptive Transfer Learning. In Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10), pages 407–412, 2010.
- [8] Anthony J Carfang and Eric W Frew. Real-time Estimation of Wireless Ground-to-Air Communication Parameters. In International Conference on Computing, Networking and Communications, Wireless Ad Hoc and Sensor Networks Symposium, pages 975–979, 2012.
- [9] Anthony J. Carfang and Eric W. Frew. Fast Bandwidth Allocation Policies for Persistent Data Ferrying. In IEEE Globecom: International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), pages 1355–1360, December 2013.
- [10] Anthony J Carfang, Eric W Frew, and Timothy X Brown. Improved Delay-Tolerant Communication by Considering Radio Propagation in Planning Data Ferry Navigation. In AIAA Guidance, Navigation and Control Conference, pages 1–14, Toronto, Canada, 2010.
- [11] Anthony J. Carfang, Eric W. Frew, and Derek B. Kingston. Cascaded Optimization of Aircraft Trajectories for Persistent Data Ferrying. Journal of Aerospace Information Systems (JAIS), 11(12):807–820, December 2014.

- [12] Kian Ming A. Chai. Generalization Errors and Learning Curves for Regression with Multi-task Gaussian Processes. In Advances in Neural Information Processing Systems (NIPS), pages 1–9, 2009.
- [13] Krzysztof Chalupka, Christopher K I Williams, and Iain Murray. A Framework for Evaluating Approximation Methods for Gaussian Process Regression. The Journal of Machine Learning Research, 14(1):333—350, 2013.
- [14] Yongguang Chen and Hisashi Kobayashi. Signal Strength Based Indoor Geolocation. In 2002 IEEE International Conference on Communications. Conference (ICC), volume 1, pages 436–439, 2002.
- [15] Girish Chowdhary, Hassan Kingravi, Robert C Grande, Jonathan P How, and Patricio Vela. Bayesian Nonparametric Model Reference Adaptive Control using Gaussian Processes. In AIAA Guidance, Navigation and Control (GNC) Conference, pages 1–17, 2013.
- [16] Cloud Cap Technology. Piccolo Plus. [http://www.cloudcaptech.com/piccolo\\_plus.shtm](http://www.cloudcaptech.com/piccolo_plus.shtm).
- [17] Kai Daniel, Bjoern Dusza, Andreas Lewandowski, and Christian Wietfeld. Airshield: A System-of-Systems MUAV Remote Sensing Architecture for Disaster Response. In IEEE International Systems Conference (SysCon), pages 196–200, 2009.
- [18] Hal III Daume. Frustratingly Easy Domain Adaptation. Conference of the Association for Computational Linguistics, pages 256–263, 2007.
- [19] Sergio Ferreira, Guilherme Carvalho, Filipe Ferreira, and Joao Sousa. Assessing the capacity of man-portable UAVs for network access point localization, using RSSI link data. In International Conference on Unmanned Aircraft Systems (ICUAS), pages 355–364, Orlando, FL, 2014.
- [20] Brian Ferris, Dieter Fox, and Neil D Lawrence. WiFi-SLAM Using Gaussian Process Latent Variable Models. In Proceedings of the 20th international joint conference on Artificial intelligence, pages 2480–2485, 2007.
- [21] Brian Ferris, Dirk Hähnel, and Dieter Fox. Gaussian Processes for Signal Strength-Based Location Estimation. In Proceedings of Robotics: Science and Systems, 2006.
- [22] Jonathan Fink and Vijay Kumar. Online Methods for Radio Signal Mapping with Mobile Robots. In 2010 IEEE International Conference on Robotics and Automation, pages 1940–1945, May 2010.
- [23] Jonathan Fink, Alejandro Ribeiro, and Vijay Kumar. Robust Control for Mobility and Wireless Communication in Cyber-Physical Systems With Application to Robot Teams. Proceedings of the IEEE, 100(1):164–178, January 2012.
- [24] Alireza Ghaffarkhah and Yasamin Mostofi. Channel Learning and Communication-Aware Motion Planning in Mobile Networks. In American Control Conference (ACC), pages 5413–5420. IEEE, 2010.
- [25] Alireza Ghaffarkhah and Yasamin Mostofi. Optimal Motion and Communication for Persistent Information Collection using a Mobile Robot. In IEEE Globecom: International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), pages 1532–1537, December 2012.

- [26] Alejandro Gonzalez-Ruiz, Alireza Ghaffarkhah, and Yasamin Mostofi. A Comprehensive Overview and Characterization of Wireless Channels for Networked Robotic and Control Systems. Journal of Robotics, 2011:1–19, 2011.
- [27] Robert C. Grande, Girish Chowdhary, and Jonathan P. How. Nonparametric Adaptive Control using Gaussian Processes with Online Hyperparameter Estimation. In 52nd IEEE Conference on Decision and Control (CDC), pages 861–867, December 2013.
- [28] Robert C. Grande, Thomas J. Walsh, and Jonathan P. How. Sample Efficient Reinforcement Learning with Gaussian Processes. In Proceedings of the 31st International Conference on Machine Learning (ICML), volume 32, 2014.
- [29] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-Optimal Sensor Placements in Gaussian Processes. In Proceedings of the 22nd international conference on Machine learning - ICML '05, pages 265–272, New York, USA, August 2005. ACM Press.
- [30] Jason Hardy, Frank Havlak, and Mark Campbell. Multiple-Step Prediction Using a Two Stage Gaussian Process Model. In American Control Conference (ACC), pages 3443–3449, June 2014.
- [31] D. Higdon, J. Swall, and J. Kern. Non-Stationary Spatial Modeling. Bayesian statistics, 6:761–768, 1999.
- [32] Jerry L. Hintze and Ray D. Nelson. Violin Plots: A Box Plot-Density Trace Synergism. The American Statistician, 52(2):181–184, May 1998.
- [33] Hobby King. Skywalker X8 UAV. [http://www.hobbyking.com/hobbyking/store/\\_\\_27132\\_\\_skywalker\\_x\\_8\\_fpv\\_uav\\_flying\\_wing\\_2120mm.html](http://www.hobbyking.com/hobbyking/store/__27132__skywalker_x_8_fpv_uav_flying_wing_2120mm.html).
- [34] Geoffrey A. Hollinger, Arvind A. Pereira, and Gaurav S. Sukhatme. Learning Uncertainty Models for Reliable Operation of Autonomous Underwater Vehicles. In International Conference on Robotics and Automation, pages 5593–5599, 2013.
- [35] Geoffrey A. Hollinger and Gaurav S. Sukhatme. Sampling-based robotic information gathering algorithms. The International Journal of Robotics Research, 33(9):1271–1287, June 2014.
- [36] IEEE Computer Society. IEEE Std 802.11-2007: IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN MAC and PHY Specifications. Technical report, LAN/MAN Committee, IEEE Computer Society., 2007.
- [37] Jonathan Ko and Dieter Fox. GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models. In International Conference on Intelligent Robots and Systems, volume 27, pages 75–90, May 2008.
- [38] Andrew Kopeikin, Sameera S. Ponda, Luke B. Johnson, and Jonathan P. How. Multi-UAV Network Control through Dynamic Task Allocation: Ensuring Data-Rate and Bit-Error-Rate Support. In IEEE Globecom: International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), pages 1579–1584, December 2012.

- [39] Andreas Krause and Carlos Guestrin. Nonmyopic Active Learning of Gaussian Processes : An Exploration-Exploitation Approach. In Proceedings of the 24th international conference on Machine learning (ICML), pages 449–456. ACM, 2007.
- [40] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. The Journal of Machine Learning Research, 9:235—284, 2008.
- [41] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Lydia E. Kavraki, and Dan S. Wallach. Robotics-Based Location Sensing Using Wireless Ethernet. Wireless Networks, 11(1-2):189–204, January 2005.
- [42] Nicholas R J Lawrance and Salah Sukkarieh. Path Planning for Autonomous Soaring Flight in Dynamic Wind Fields. In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 2499–2505. IEEE, 2011.
- [43] Neil D Lawrence and John C Platt. Learning to Learn with the Informative Vector Machine. In Proceedings of the 21st International Conference on Machine Learning, page 65. ACM, 2004.
- [44] Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks. In Proceedings of the 24th International Conference on Machine Learning, pages 489–496, 2007.
- [45] Julia Letchner, Dieter Fox, and A. LaMarca. Large-scale localization from wireless signal strength. In Proceedings of the national conference on artificial intelligence (AAAI), volume 20, page 15. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.
- [46] Chi Harold Liu, Ting He, Kang-won Lee, Kin K. Leung, and Ananthram Swami. Dynamic Control of Data Ferries under Partial Observations. In IEEE Wireless Communications and Networking Conference (WCNC), 2010.
- [47] John A Magliacane. SPLAT! A Terrestrial RF Path Analysis Application for Linux/Unix, 2008.
- [48] Mehrzad Malmirchegini, Alireza Ghaffarkhah, and Yasamin Mostofi. Impact of Motion and Channel Parameters on the Estimation of Transmitter Position in Robotic Networks. In IEEE Globecom: International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), 2012.
- [49] Mehrzad Malmirchegini and Yasamin Mostofi. On the Spatial Predictability of Communication Channels. IEEE Transactions on Wireless Communications, 11(3):964–978, 2012.
- [50] Arman Melkumyan and Fabio Ramos. Multi-Kernel Gaussian Processes. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI), pages 1408–1413, 2011.
- [51] Thomas P Minka and Rosalind W. Picard. Learning How to Learn is Learning with Point Sets. Technical report, 1997.
- [52] Yasamin Mostofi. Decentralized Communication-Aware Motion Planning in Mobile Networks: An Information-Gain Approach. Journal of Intelligent and Robotic Systems, 56(Special Issue: Unmanned Autonomous Vehicles):233–256, 2009.

- [53] Yasamin Mostofi, Alejandro Gonzalez-ruiz, Alireza Gaffarkhah, and Ding Li. Characterization and Modeling of Wireless Channels for Networked Robotic and Control Systems - A Comprehensive Overview. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4849 – 4854, 2009.
- [54] Yasamin Mostofi, Mehrzad Malmirchegini, and Alireza Ghaffarkhah. Estimation of Communication Signal Strength in Robotic Networks. In International Conference on Robotics and Automation, pages 1946—1951, Anchorage, Alaska, 2010. IEEE.
- [55] Calvin Newport, David Kotz, Yougu Yuan, Robert S. Gray, Jason Liu, and Chip Elliott. Experimental Evaluation of Wireless Simulation Assumptions. Simulation, 83(9):643–661, September 2007.
- [56] Christopher J Paciorek and Mark J Schervish. Nonstationary Covariance Functions for Gaussian Process Regression. Advances in Neural Information Processing Systems, 16:273–280, 2004.
- [57] Sinno Jialin Pan, Dou Shen, Qiang Yang, and James T Kwok. Transferring Localization Models Across Space. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pages 1383—1388, 2008.
- [58] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345—1359, 2010.
- [59] Sinno Jialin Pan, Vincent Wenchen Zheng, Qiang Yang, and Derek Hao Hu. Transfer learning for WiFi-based Indoor Localization. In Association for the Advancement of Artificial Intelligence (AAAI) workshop, pages 43–48, 2008.
- [60] Ben Pearre and Timothy X. Brown. Model-Free Trajectory Optimisation for Unmanned Aircraft Serving as Data Ferries for Widespread Sensors. Remote Sensing, 4(12):2971–3005, October 2012.
- [61] Caleb Timothy Phillips. Geostatistical Techniques for Practical Wireless Network Coverage Mapping, PhD thesis, University of Colorado, 2012.
- [62] Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary Gaussian Process Regression using Point Estimates of Local Smoothness. Machine Learning and Knowledge Discovery in Databases, pages 204–219, 2008.
- [63] Patrick A Plonski, Pratap Tokekar, and Volkan Isler. Energy-Efficient Path Planning for Solar-Powered Mobile Robots. Journal of Field Robotics, 30(4):583–601, July 2013.
- [64] Joaquin Quinonera-Candela and Carl Edward Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. The Journal of Machine Learning Research, 6:1939–1959, 2005.
- [65] Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing Informative Priors using Transfer Learning. In Proceedings of the 23rd International Conference on Machine Learning (ICML), pages 713–720, 2006.
- [66] Theodore S Rappaport. Wireless Communications: Principles and Practice. 2002.

- [67] Carl Edward Rasmussen. Gaussian Processes for Machine Learning. In Advanced Lectures on Machine Learning, pages 63–71. Springer, 2004.
- [68] Carl Edward Rasmussen and Christopher K I Williams. Gaussian Processes for Machine Learning. MIT press, 2006.
- [69] Research and Engineering Center for Unmanned Vehicles. NexSTAR UAS. <http://recuv.colorado.edu/nexstar/>.
- [70] Research and Engineering Center for Unmanned Vehicles. Tempest UAS. <http://recuv.colorado.edu/tempest/>.
- [71] A.M. Schmidt and A. O’Hagan. Bayesian Inference for Non-Stationary Spatial Covariance Structure via Spatial Deformations. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 65(3):743–758, 2003.
- [72] Anton Schwaighofer and Volker Tresp. Transductive and Inductive Methods for Approximate Gaussian Process Regression. In Advances in Neural Information Processing Systems, number November, pages 953–960, 2002.
- [73] Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning Gaussian Process Kernels via Hierarchical Bayes. In Advances in Neural Information Processing Systems, volume 17, pages 1209–1216, 2005.
- [74] Matthias Seeger. Gaussian Processes for Machine Learning. International Journal of Neural Systems, 14(2):69–106, 2004.
- [75] Matthias Seeger, Christopher K I Williams, and Neil D Lawrence. Fast Forward Selection to Speed Up Sparse Gaussian Process Regression. In Workshop on AI and Statistics 9, 2003.
- [76] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003., pages 30–41. Ieee.
- [77] Amarjeet Singh, Fabio Ramos, Hugh Durrant-Whyte, and William J Kaiser. Modeling and Decision Making in Spatio-Temporal Processes for Environmental Surveillance. In 2010 IEEE International Conference on Robotics and Automation, pages 5490–5497, May 2010.
- [78] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In Advances in Neural Information Processing Systems, volume 18, pages 1257–1264, 2006.
- [79] Maciej Stachura. Cooperative Planning for UAS-based Sensor Networks In Realistic Communication Environments, PhD thesis, University of Colorado, 2014.
- [80] Maciej Stachura and Eric W. Frew. Cooperative Target Localization with a Communication-Aware Unmanned Aircraft System. In AIAA Guidance, Navigation, and Control Conference, page 15, Toronto, Canada, 2010. AIAA.
- [81] Maciej Stachura and Eric W. Frew. Cooperative Target Localization with a Communication-Aware Unmanned Aircraft System. Journal of Guidance, Control, and Dynamics, 34(5):1352–1362, September 2011.

- [82] Maciej Stachura, Neeti Wagle, and Eric W Frew. A Comparison of Filters for UAS-Based Localization of Stationary RF Sources. In Aviation Forum, 2015.
- [83] Ryo Sugihara and Rajesh K Gupta. Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility. Distributed Computing in Sensor Systems, pages 386–399, 2008.
- [84] Sebastian Thrun. Learning To Learn: Introduction. In Learning To Learn. Kluwer Academic Publishers, 1996.
- [85] John W. Tukey. Exploratory Data Analysis. Reading, Mass., 1977.
- [86] Neeti Wagle and Eric W Frew. A Particle Filter Approach to WiFi Target Localization. In AIAA Guidance, Navigation, and Control Conference, Toronto, Canada, August 2010.
- [87] Neeti Wagle and Eric W. Frew. Spatio-temporal Characterization of Airborne Radio Frequency Environments. In IEEE Globecom: International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (WiUAV), pages 1269–1273, Houston, TX, December 2011.
- [88] Neeti Wagle and Eric W Frew. Transfer Learning for Dynamic RF Environments. In American Control Conference (ACC), Montreal, Canada, 2012.
- [89] Matthew Wall. GALib: A C++ Library of Genetic Algorithm Components. Technical Report August, 1996.
- [90] Yali Wang and Brahim Chaib-draa. A Marginalized Particle Gaussian Process Regression. In Advances in Neural Information Processing Systems, pages 1187—1195, 2012.
- [91] WiFi-Plus. WiFi 2.4GHz Omni Antennas. <http://www.wifi-plus.com/antennas/24ghzantennaspeccs.html>.
- [92] Christopher K I Williams and Matthias Seeger. Using the Nyström Method to Speed Up Kernel Machines. In Advances in Neural Information Processing Systems, 2001.
- [93] Yunfei Xu, Jongeun Choi, and Songhwai Oh. Mobile Sensor Network Navigation Using Gaussian Processes With Truncated Observations. IEEE Transactions on Robotics, 27(6):1118–1131, December 2011.
- [94] Pengcheng Zhan, Kai Yu, and A. Lee Swindlehurst. Wireless Relay Communications with Unmanned Aerial Vehicles: Performance and Optimization. IEEE Transactions on Aerospace and Electronic Systems, 47(3):2068–2085, 2011.
- [95] Zhensheng Zhang. Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. IEEE Communications Surveys and Tutorials, 8(1):24–37, 2006.
- [96] Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In Proceedings of the 5th ACM international Symposium on Mobile Ad Hoc Networking and Computing, pages 187–198, 2004.
- [97] Vincent Wenchen Zheng, Sinno Jialin Pan, Qiang Yang, and Jeffrey Junfeng Pan. Transferring Multi-device Localization Models using Latent Multi-task Learning. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pages 1427–1432, 2008.

- [98] Vincent Wenchen Zheng, Evan Wei Xiang, Qiang Yang, and Dou Shen. Transferring Localization Models Over Time. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pages 1421–1426, 2008.

## Appendix A

### Gaussian Process Performance Results

Table A.1: GP RMSE Performance for 433MHz Tempest Datasets

No.	Without outlier detection				With outlier detection			
	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio temporal ARD GP
1	4.56	3.40	3.41	–	4.54	3.39	3.40	–
2	5.16	3.21	2.98	–	5.18	3.37	2.98	–
3	4.99	3.09	3.09	–	5.00	3.09	3.09	–
4	4.65	4.29	4.18	–	4.39	4.28	4.20	–
5	3.57	3.14	3.15	–	3.42	3.14	3.14	–
6	3.66	3.44	3.38	–	3.64	3.42	3.37	–
7	5.10	4.63	4.74	–	5.11	4.64	4.76	–
8	4.01	2.28	2.27	–	4.05	2.27	2.27	–
9	3.33	3.30	3.05	–	3.37	3.28	3.04	–
10	4.73	2.90	2.91	–	4.75	2.91	2.92	–
11	4.01	3.65	3.66	–	4.02	3.65	3.66	–
12	4.62	4.45	4.41	–	4.62	4.45	4.41	–
13	4.47	3.52	3.55	–	4.47	3.53	3.56	–
14	4.09	3.93	4.02	–	4.10	3.93	4.01	–
15	3.49	3.45	3.55	–	3.46	3.43	3.52	–
16	4.24	4.00	4.00	–	4.23	4.07	4.10	–
17	3.84	3.83	3.85	–	3.84	3.83	3.85	–
18	3.49	3.41	3.51	–	3.48	3.40	3.49	–
19	3.03	2.78	2.76	–	3.03	2.90	2.76	–
20	2.32	2.30	2.29	–	2.32	2.30	2.29	–
Avg	4.07	3.45	3.44	–	4.05	3.46	3.44	–

Table A.2: GP RMSE Performance for 2.4GHz NexSTAR Datasets

No.	Without outlier detection				With outlier detection			
	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP
1	3.55	3.35	3.35	3.04	3.68	3.40	3.41	3.06
2	1.90	1.43	1.40	0.99	1.92	1.40	1.40	1.00
3	1.68	1.52	1.54	1.03	1.71	1.52	1.54	1.03
4	12.10	2.75	2.80	1.05	14.42	2.71	2.70	1.09
5	1.55	1.23	1.24	1.02	1.52	1.22	1.23	1.03
6	1.73	1.47	1.47	1.53	1.74	1.47	1.47	1.54
7	2.49	1.89	1.88	1.10	3.47	2.02	2.02	1.14
8	2.62	1.90	1.90	1.27	2.63	1.90	1.90	1.27
9	3.06	2.08	2.08	1.37	3.04	2.07	2.08	1.37
10	4.46	3.28	3.29	1.57	4.55	3.31	3.32	1.57
11	1.49	1.09	1.11	0.95	2.30	1.09	1.10	0.94
12	1.97	1.42	1.42	1.42	2.06	1.43	1.43	1.43
13	2.19	1.52	1.52	1.52	2.22	1.49	1.49	1.49
14	0.86	0.80	0.80	0.80	0.87	0.81	0.81	0.81
15	2.86	2.35	2.36	2.36	2.99	2.43	2.43	2.43
16	3.21	2.92	2.92	2.92	3.23	2.94	2.94	2.94
17	2.81	2.28	2.30	2.30	2.81	2.28	2.30	2.30
18	2.53	2.15	2.17	2.12	2.53	2.15	2.17	2.13
Avg	2.95	1.97	1.98	1.58	3.20	1.98	1.99	1.59

Table A.3: GP RMSE Performance for Skywalker X8 Mobile-Emitter Datasets

No.	Without outlier detection				With outlier detection			
	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Empirical Radio Model	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP
1	5.05	4.61	3.89	3.89	5.05	4.61	3.89	3.89
2	7.17	6.61	6.64	6.64	7.17	6.61	6.64	6.64
3	6.09	5.23	5.26	4.92	6.09	5.23	5.26	4.92
4	0.37	0.34	0.34	0.35	0.37	0.34	0.34	0.35
5	5.89	5.38	5.40	5.39	5.89	5.38	5.40	5.39
6	0.35	0.33	0.33	0.32	0.35	0.33	0.33	0.32
7	6.14	4.86	4.86	4.07	6.14	4.86	4.86	4.07
8	0.38	0.35	0.35	0.33	0.38	0.35	0.35	0.33
9	5.32	3.31	3.30	2.94	5.32	3.31	3.30	2.94
10	0.37	0.35	0.35	0.34	0.37	0.35	0.35	0.34
11	6.07	4.83	4.72	3.88	6.07	4.83	4.72	3.88
12	0.31	0.28	0.28	0.27	0.31	0.28	0.28	0.27
Avg	3.63	3.04	2.98	2.78	3.63	3.04	2.98	2.78

Table A.4: GP MSLL for 433MHz Tempest Datasets

No.	Without outlier detection			With outlier detection		
	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP
1	$-0.38 \pm 0.03$	$-0.37 \pm 0.03$	–	$-0.38 \pm 0.03$	$-0.37 \pm 0.03$	–
2	$-0.53 \pm 0.14$	$-0.63 \pm 0.04$	–	$-0.47 \pm 0.15$	$-0.64 \pm 0.04$	–
3	$-0.47 \pm 0.04$	$-0.47 \pm 0.04$	–	$-0.47 \pm 0.04$	$-0.47 \pm 0.04$	–
4	$-0.05 \pm 0.13$	$-0.10 \pm 0.06$	–	$-0.00 \pm 0.10$	$-0.04 \pm 0.05$	–
5	$-0.13 \pm 0.01$	$-0.12 \pm 0.02$	–	$-0.09 \pm 0.03$	$-0.08 \pm 0.03$	–
6	$0.12 \pm 0.32$	$0.03 \pm 0.20$	–	$0.12 \pm 0.32$	$0.05 \pm 0.22$	–
7	$-0.09 \pm 0.06$	$-0.06 \pm 0.09$	–	$-0.09 \pm 0.06$	$-0.06 \pm 0.09$	–
8	$-0.59 \pm 0.05$	$-0.58 \pm 0.06$	–	$-0.60 \pm 0.05$	$-0.59 \pm 0.06$	–
9	$0.02 \pm 0.05$	$-0.08 \pm 0.06$	–	$-0.00 \pm 0.05$	$-0.10 \pm 0.08$	–
10	$-0.49 \pm 0.06$	$-0.50 \pm 0.08$	–	$-0.49 \pm 0.07$	$-0.49 \pm 0.07$	–
11	$-0.09 \pm 0.02$	$-0.09 \pm 0.03$	–	$-0.10 \pm 0.02$	$-0.09 \pm 0.03$	–
12	$-0.04 \pm 0.03$	$-0.04 \pm 0.03$	–	$-0.04 \pm 0.03$	$-0.04 \pm 0.03$	–
13	$-0.23 \pm 0.07$	$-0.22 \pm 0.08$	–	$-0.23 \pm 0.08$	$-0.22 \pm 0.08$	–
14	$-0.04 \pm 0.04$	$-0.02 \pm 0.03$	–	$-0.04 \pm 0.04$	$-0.02 \pm 0.03$	–
15	$0.01 \pm 0.05$	$0.06 \pm 0.12$	–	$0.01 \pm 0.08$	$0.07 \pm 0.16$	–
16	$-0.06 \pm 0.05$	$-0.06 \pm 0.05$	–	$-0.04 \pm 0.04$	$-0.03 \pm 0.04$	–
17	$-0.00 \pm 0.00$	$0.00 \pm 0.01$	–	$-0.00 \pm 0.00$	$0.00 \pm 0.01$	–
18	$-0.02 \pm 0.03$	$0.00 \pm 0.01$	–	$-0.03 \pm 0.03$	$-0.00 \pm 0.02$	–
19	$-0.10 \pm 0.08$	$-0.10 \pm 0.08$	–	$-0.03 \pm 0.10$	$-0.10 \pm 0.08$	–
20	$0.01 \pm 0.04$	$-0.01 \pm 0.01$	–	$0.01 \pm 0.04$	$-0.01 \pm 0.01$	–

Table A.5: GP MSLL for 2.4GHz NexSTAR Datasets

No.	Without outlier detection			With outlier detection		
	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP
1	-0.05 ± 0.02	-0.05 ± 0.02	-0.15 ± 0.03	-0.08 ± 0.02	-0.07 ± 0.02	-0.18 ± 0.02
2	-0.38 ± 0.08	-0.35 ± 0.07	-0.77 ± 0.07	-0.33 ± 0.04	-0.33 ± 0.03	-0.77 ± 0.08
3	-0.10 ± 0.01	-0.09 ± 0.03	-0.60 ± 0.02	-0.12 ± 0.02	-0.11 ± 0.04	-0.61 ± 0.02
4	-1.47 ± 0.06	-1.46 ± 0.05	-2.45 ± 0.07	-1.66 ± 0.04	-1.67 ± 0.05	-2.60 ± 0.08
5	-0.23 ± 0.13	-0.22 ± 0.14	-0.42 ± 0.15	-0.22 ± 0.13	-0.21 ± 0.13	-0.39 ± 0.14
6	-0.16 ± 0.05	-0.16 ± 0.05	-0.00 ± 0.27	-0.16 ± 0.05	-0.16 ± 0.05	0.01 ± 0.29
7	-0.28 ± 0.06	-0.28 ± 0.06	-0.82 ± 0.09	-0.54 ± 0.06	-0.54 ± 0.05	-1.11 ± 0.09
8	-0.32 ± 0.03	-0.32 ± 0.04	-0.71 ± 0.06	-0.32 ± 0.04	-0.32 ± 0.04	-0.72 ± 0.06
9	-0.39 ± 0.04	-0.39 ± 0.04	-0.84 ± 0.06	-0.39 ± 0.04	-0.39 ± 0.04	-0.83 ± 0.06
10	-0.29 ± 0.01	-0.29 ± 0.01	-1.03 ± 0.05	-0.30 ± 0.01	-0.30 ± 0.01	-1.06 ± 0.04
11	-0.29 ± 0.06	-0.28 ± 0.04	-0.45 ± 0.03	-0.72 ± 0.07	-0.72 ± 0.05	-0.89 ± 0.05
12	-0.35 ± 0.05	-0.34 ± 0.05	-0.34 ± 0.05	-0.38 ± 0.05	-0.38 ± 0.05	-0.38 ± 0.05
13	-0.42 ± 0.05	-0.42 ± 0.05	-0.42 ± 0.05	-0.41 ± 0.04	-0.41 ± 0.04	-0.41 ± 0.04
14	-0.06 ± 0.02	-0.06 ± 0.02	-0.06 ± 0.02	-0.07 ± 0.02	-0.07 ± 0.02	-0.07 ± 0.02
15	-0.19 ± 0.04	-0.19 ± 0.04	-0.19 ± 0.04	-0.20 ± 0.04	-0.20 ± 0.04	-0.20 ± 0.04
16	-0.09 ± 0.02	-0.09 ± 0.02	-0.09 ± 0.02	-0.08 ± 0.03	-0.08 ± 0.03	-0.08 ± 0.03
17	-0.20 ± 0.02	-0.19 ± 0.01	-0.19 ± 0.01	-0.20 ± 0.02	-0.19 ± 0.01	-0.19 ± 0.01
18	-0.16 ± 0.09	-0.15 ± 0.09	-0.17 ± 0.11	-0.16 ± 0.09	-0.15 ± 0.09	-0.17 ± 0.11

Table A.6: GP MSLL for Skywalker X8 Mobile-Emitter Dataset

No.	Without outlier detection			With outlier detection		
	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP	Spatial Isotropic GP	Spatial ARD GP	Spatio- temporal ARD GP
1	-0.10 ± 0.04	-0.23 ± 0.05	-0.23 ± 0.05	-0.10 ± 0.04	-0.23 ± 0.05	-0.23 ± 0.05
2	-0.07 ± 0.01	-0.06 ± 0.01	-0.06 ± 0.01	-0.07 ± 0.01	-0.06 ± 0.01	-0.06 ± 0.01
3	-0.15 ± 0.03	-0.14 ± 0.03	-0.21 ± 0.05	-0.15 ± 0.03	-0.14 ± 0.03	-0.21 ± 0.05
4	-0.09 ± 0.01	-0.10 ± 0.02	-0.11 ± 0.02	-0.09 ± 0.01	-0.10 ± 0.02	-0.11 ± 0.02
5	-0.09 ± 0.03	-0.08 ± 0.03	-0.08 ± 0.03	-0.09 ± 0.03	-0.08 ± 0.03	-0.08 ± 0.03
6	-0.05 ± 0.02	-0.04 ± 0.02	-0.10 ± 0.04	-0.05 ± 0.02	-0.04 ± 0.03	-0.10 ± 0.04
7	-0.21 ± 0.03	-0.22 ± 0.03	-0.43 ± 0.06	-0.21 ± 0.03	-0.22 ± 0.03	-0.43 ± 0.06
8	-0.06 ± 0.03	-0.09 ± 0.04	-0.13 ± 0.03	-0.06 ± 0.03	-0.09 ± 0.04	-0.13 ± 0.03
9	-0.46 ± 0.07	-0.46 ± 0.07	-0.57 ± 0.05	-0.46 ± 0.07	-0.46 ± 0.07	-0.57 ± 0.05
10	-0.07 ± 0.02	-0.07 ± 0.02	-0.10 ± 0.01	-0.07 ± 0.02	-0.07 ± 0.02	-0.10 ± 0.01
11	-0.21 ± 0.05	-0.22 ± 0.02	-0.43 ± 0.05	-0.21 ± 0.05	-0.22 ± 0.02	-0.43 ± 0.05
12	-0.08 ± 0.02	-0.09 ± 0.02	-0.15 ± 0.05	-0.08 ± 0.02	-0.09 ± 0.02	-0.15 ± 0.05

Table A.7: Percentage-wise Iterative Training GPs for 433-Tempest Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSLL	Training Set Size	Training Time	Prediction RMSE	Prediction MSLL
1	126.00	1.20	5.32	0.36	1007.00	31.60	3.29	-0.41
2	460.00	5.33	4.90	-0.06	3675.00	719.65	3.21	-0.56
3	271.00	1.83	4.83	0.27	2164.00	68.72	3.37	-0.41
4	43.00	0.35	4.49	0.01	340.00	5.54	3.65	-0.00
5	55.00	0.33	3.60	0.04	435.00	4.16	3.06	-0.12
6	17.00	0.30	5.29	0.35	133.00	0.72	3.85	0.02
7	21.00	0.28	8.64	4.14	167.00	0.95	4.40	-0.11
8	158.00	1.16	4.14	-0.04	1260.00	29.18	2.02	-0.63
9	43.00	0.37	3.56	0.03	342.00	3.40	3.29	-0.04
10	91.00	0.57	4.74	-0.20	724.00	10.30	2.73	-0.52
11	100.00	0.54	4.05	0.06	795.00	13.35	3.77	-0.05
12	55.00	0.45	5.03	0.08	440.00	6.44	4.41	-0.07
13	34.00	0.32	4.40	0.29	271.00	1.79	3.70	-0.23
14	40.00	0.34	4.27	0.09	320.00	2.40	3.65	-0.06
15	14.00	0.20	5.94	2.50	107.00	1.03	4.37	-0.04
16	42.00	0.34	4.16	0.25	336.00	2.62	3.41	-0.07
17	38.00	0.32	4.72	0.12	301.00	2.81	4.32	-0.00
18	29.00	0.31	3.82	0.44	227.00	1.64	3.22	-0.07
19	21.00	0.28	2.90	0.02	164.00	0.69	2.52	-0.12
20	51.00	0.42	2.62	0.19	404.00	3.59	2.43	0.05
Average	85.45	0.76	4.57	0.45	680.60	45.53	3.43	-0.17

Table A.8: Percentage-wise Iterative Training GPs for 2.4GHz NexSTAR Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSLL	Training Set Size	Training Time	Prediction RMSE	Prediction MSLL
1	163.00	1.00	12.34	0.03	1303.00	40.24	3.35	-0.15
2	378.00	4.48	19.46	26.42	3022.00	349.78	1.06	-0.73
3	379.00	5.57	23.94	35.86	3028.00	496.04	1.05	-0.57
4	453.00	8.21	27.67	1.27	3624.00	399.41	1.08	-2.45
5	75.00	0.55	1.56	-0.00	600.00	7.35	0.86	-0.50
6	75.00	0.54	2.36	-0.01	596.00	4.90	1.44	5.11
7	200.00	1.63	97.59	56.27	1595.00	76.73	1.17	-1.04
8	274.00	2.20	2.93	-0.02	2190.00	160.02	1.32	-0.69
9	270.00	2.48	3.14	-0.08	2158.00	169.80	1.56	-0.72
10	352.00	3.15	12.56	-1.46	2812.00	288.12	1.52	-1.07
11	192.00	1.30	2.18	-0.30	1536.00	77.93	0.97	-0.84
12	249.00	1.82	2.40	0.08	1988.00	75.56	1.42	-0.37
13	256.00	1.57	2.80	-0.27	2041.00	84.10	1.53	-0.38
14	163.00	0.96	0.89	0.18	1304.00	36.91	0.75	-0.07
15	251.00	1.75	86.45	-322.98	2002.00	107.21	2.41	-0.19
16	200.00	1.82	16.98	1.01	1596.00	51.87	3.18	-0.07
17	200.00	1.21	2.87	0.00	1596.00	52.07	2.17	-0.23
18	26.00	0.33	2.66	-0.22	204.00	1.44	2.01	-0.17
Average	230.89	2.25	17.82	-11.35	1844.17	137.75	1.60	-0.29

Table A.9: Percentage-wise Iterative Training GPs for Skywalker X8 Mobile-Emitter Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSL	Training Set Size	Training Time	Prediction RMSE	Prediction MSL
1	98.00	1.44	14.30	-1.63	784.00	20.97	3.92	-0.19
2	217.00	3.03	23.85	0.03	1730.00	85.20	6.97	-0.05
3	200.00	1.74	10.95	0.30	1596.00	98.22	5.00	-0.26
4	198.00	1.66	0.53	-0.09	1580.00	113.78	0.33	-0.06
5	196.00	1.65	6.39	-0.03	1566.00	152.31	5.23	-0.06
6	193.00	2.08	0.30	-0.04	1541.00	112.56	0.26	-0.13
7	106.00	0.85	6.47	-0.19	845.00	34.85	3.74	-0.52
8	175.00	1.63	0.35	-0.02	1400.00	77.89	0.31	-0.12
9	119.00	0.89	9.58	-0.62	945.00	26.12	3.09	-0.54
10	208.00	1.70	0.40	-0.05	1664.00	103.84	0.35	-0.14
11	97.00	0.80	57.30	-7.51	774.00	25.35	3.81	-0.39
12	190.00	1.49	0.35	-0.09	1520.00	121.53	0.27	-0.14
Average	166.42	1.58	10.90	-0.83	1328.75	81.05	2.77	-0.22

Table A.10: Fixed-Increment Iterative Training GPs for 433MHz Tempest Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSL	Training Set Size	Training Time	Prediction RMSE	Prediction MSL
1	50.00	0.45	4.64	-0.03	1158.00	55.01	2.65	-0.50
2	50.00	0.42	5.37	0.03	4493.00	341.88	2.65	-0.68
3	50.00	0.39	14.88	-1.63	2604.00	100.44	3.13	-0.49
4	50.00	0.46	5.67	0.04	325.00	2.49	4.27	-0.05
5	50.00	0.42	3.56	0.05	443.00	5.01	3.34	-0.05
6	50.00	0.41	3.24	-0.04	66.00	0.48	3.02	-0.07
7	50.00	0.37	6.79	0.15	108.00	0.59	4.98	-0.04
8	50.00	0.38	18.49	0.34	1474.00	40.07	2.06	-0.67
9	50.00	0.74	3.82	0.04	327.00	4.22	3.28	-0.06
10	50.00	0.66	5.16	0.21	804.00	16.42	2.82	-0.58
11	50.00	0.54	5.19	0.02	893.00	21.32	3.43	-0.15
12	50.00	0.64	5.28	0.01	449.00	6.67	4.38	-0.04
13	50.00	0.63	3.36	-0.21	238.00	2.33	3.28	-0.23
14	50.00	0.55	4.22	-0.01	299.00	3.27	4.03	-0.08
15	—	—	—	—	—	—	—	—
16	50.00	0.58	4.04	0.01	320.00	7.81	4.23	0.08
17	50.00	0.60	5.07	0.69	276.00	2.76	4.25	-0.02
18	50.00	0.62	3.80	0.11	183.00	2.18	3.76	0.06
19	50.00	0.61	2.99	0.00	105.00	0.84	2.78	-0.10
20	50.00	0.63	2.38	-0.03	404.00	5.80	2.50	0.04
Average	47.50	0.51	5.40	-0.01	748.45	30.98	3.24	-0.18

Table A.11: Fixed-Increment Iterative Training GPs for 2.4GHz NexSTAR Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSL	Training Set Size	Training Time	Prediction RMSE	Prediction MSL
1	50.00	0.52	6.10	0.08	1478.00	48.41	3.89	-0.12
2	50.00	0.54	23.74	321.79	3627.00	565.80	1.13	-0.79
3	50.00	0.49	3.35	-1.39	3635.00	834.17	0.82	-0.66
4	50.00	0.47	30.46	-87.92	4379.00	629.77	0.98	-2.49
5	50.00	0.48	1.70	-0.02	600.00	6.54	0.98	-0.39
6	50.00	0.55	1.80	0.02	595.00	9.28	1.62	-0.06
7	50.00	0.50	265.47	5868.80	1843.00	98.56	1.11	-1.04
8	50.00	0.53	2.80	0.09	2587.00	230.27	1.22	-0.75
9	50.00	0.52	2.82	0.10	2547.00	230.39	1.04	-0.98
10	50.00	0.49	187.93	-165.97	3364.00	434.73	1.91	-0.79
11	50.00	0.49	86.14	-384.35	1770.00	107.36	0.89	-0.71
12	50.00	0.45	2.21	-1.43	2334.00	87.78	1.40	-0.33
13	50.00	0.50	4.21	0.23	2401.00	96.56	1.38	-0.35
14	50.00	0.51	0.88	39.62	1480.00	41.83	0.72	-0.11
15	50.00	0.48	18.60	8.89	2352.00	139.27	2.30	-0.28
16	50.00	0.47	24.34	0.79	1845.00	67.57	2.97	-0.05
17	50.00	0.46	16.94	0.24	1845.00	67.87	2.33	-0.21
18	50.00	0.47	2.25	-0.12	105.00	0.69	2.13	-0.19
Average	50.00	0.50	37.87	311.08	2154.83	205.38	1.60	-0.57

Table A.12: Fixed-Increment Iterative Training GPs for Skywalker X8 Mobile-Emitter Dataset

No.	First Iteration				Last Iteration			
	Training Set Size	Training Time	Prediction RMSE	Prediction MSL	Training Set Size	Training Time	Prediction RMSE	Prediction MSL
1	50.00	0.69	10.79	0.11	829.00	22.04	4.43	-0.12
2	50.00	0.68	74.52	3.38	2012.00	102.19	7.08	-0.04
3	50.00	0.66	19.45	0.10	1845.00	97.34	4.48	-0.20
4	50.00	0.75	7.13	15.12	1825.00	164.82	0.29	-0.20
5	50.00	0.58	42.48	0.37	1807.00	164.04	5.41	-0.10
6	50.00	0.68	1.05	0.04	1776.00	150.19	0.37	-0.07
7	50.00	0.66	6.61	0.06	906.00	43.77	4.20	-0.43
8	50.00	0.69	7.05	-55.91	1599.00	113.68	0.31	-0.17
9	50.00	0.66	139.77	27.18	1031.00	34.49	2.82	-0.59
10	50.00	0.70	2.16	0.97	1930.00	166.19	0.33	-0.13
11	50.00	0.73	88.95	-81.16	817.00	28.40	4.32	-0.26
12	50.00	0.64	11.24	23.38	1749.00	190.91	0.23	-0.17
Average	50.00	0.68	34.27	-5.53	1510.50	106.50	2.86	-0.21

## Appendix B

### FAT-GP Prediction Analysis

#### B.1 FAT-GP Mean Components

The mean prediction equation for the FAT-GP is given by

$$\begin{aligned}\mu'_\tau &= \mathbf{m}_T(\mathbf{x}') + k_\tau(\mathbf{x}', X_D)C_\tau(X_D, X_D)^{-1}(\mathbf{y}_D - \mathbf{m}_\tau(X_D)) \\ &= \mathbf{m}_T(\mathbf{x}') + \begin{bmatrix} k_\tau(\mathbf{x}', X_T) & k_\tau(\mathbf{x}', X_S) \end{bmatrix} \begin{bmatrix} C_{TT} & K_{TS} \\ K_{ST} & K_{SS} \end{bmatrix}^{-1} \left( \begin{bmatrix} \mathbf{y}_T \\ \mathbf{y}_S \end{bmatrix} - \begin{bmatrix} \mathbf{m}_T(X_T) \\ \mathbf{m}_S(X_S) \end{bmatrix} \right)\end{aligned}$$

Using the formula for inverses of block matrices we get

$$C_\tau(X_D, X_D)^{-1} = \begin{bmatrix} C_{TT} & K_{TS} \\ K_{ST} & K_{SS} \end{bmatrix}^{-1} = \begin{bmatrix} C_{TT}^{-1} + C_{TT}^{-1}K_{TS}MK_{ST}C_{TT}^{-1} & -C_{TT}^{-1}K_{TS}M \\ -K_{ST}C_{TT}^{-1} & M \end{bmatrix}$$

where  $M = [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1}$ . Notice that

$$M = C_{S|T}^{-1} = [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1}$$

which is given by

$$p(\mathbf{y}_S \mid \mathbf{y}_T, X_T, X_S, \theta_T, \theta_S) = \mathcal{N}(\mu_{S|T}, C_{S|T})$$

$$\text{where } \mu_{S|T} = \mathbf{m}_S + K_{ST}C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T)$$

$$C_{S|T} = C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}$$

Plugging the inverse back into the mean prediction expression we get

$$\mu'_\tau = \mathbf{m}_T(\mathbf{x}') + \begin{bmatrix} k_\tau(\mathbf{x}', X_T) & k_\tau(\mathbf{x}', X_S) \end{bmatrix} \begin{bmatrix} C_{TT}^{-1} + C_{TT}^{-1}K_{TS}C_{S|T}^{-1}K_{ST}C_{TT}^{-1} & -C_{TT}^{-1}K_{TS}C_{S|T}^{-1} \\ -C_{S|T}^{-1}K_{ST}C_{TT}^{-1} & C_{S|T}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{y}_T - \mathbf{m}_T(X_T) \\ \mathbf{y}_S - \mathbf{m}_S(X_S) \end{bmatrix}$$

and expanding the expression we get

$$\begin{aligned} \mu'_\tau &= \mathbf{m}_T(\mathbf{x}') + k_T(\mathbf{x}', X_T)C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ &\quad + k_T(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS}C_{S|T}^{-1}K_{ST}C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ &\quad - k_T(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS}C_{S|T}^{-1}(\mathbf{y}_S - \mathbf{m}_S(X_S)) \\ &\quad - k_S(\mathbf{x}', X_S)C_{S|T}^{-1}K_{ST}C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ &\quad + k_S(\mathbf{x}', X_S)C_{S|T}^{-1}(\mathbf{y}_S - \mathbf{m}_S(X_S)) \end{aligned}$$

Grouping the terms with common factors,

$$\begin{aligned} \mu'_\tau &= \mathbf{m}_T(\mathbf{x}') + k_T(\mathbf{x}', X_T)C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ &\quad + [k_T(\mathbf{x}', X_T)C_{TT}^{-1}K_{TS} - k_S(\mathbf{x}', X_S)] \times \\ &\quad [C_{SS} - K_{ST}C_{TT}^{-1}K_{TS}]^{-1} \times \\ &\quad [K_{ST}C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) - (\mathbf{y}_S - \mathbf{m}_S(X_S))] \end{aligned}$$

This equation can also be written in terms of cross covariance matrices and similarity measure as

$$\begin{aligned} \mu'_\tau &= \mathbf{m}_T(\mathbf{x}') + k_T(\mathbf{x}', X_T)C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) \\ &\quad + [\lambda k_T(\mathbf{x}', X_T)C_{TT}^{-1}K_{cross}(X_T, X_S) - k_S(\mathbf{x}', X_S)] \times \\ &\quad [C_{SS} - \lambda^2 K_{cross}(X_S, X_T)C_{TT}^{-1}K_{cross}(X_T, X_S)]^{-1} \times \\ &\quad [\lambda K_{cross}(X_S, X_T)C_{TT}^{-1}(\mathbf{y}_T - \mathbf{m}_T(X_T)) - (\mathbf{y}_S - \mathbf{m}_S(X_S))] \end{aligned}$$

## B.2 Impact of SNR on Variance Components

Equation (5.22) (stated here again) shows that the variance prediction of a FAT-GP contains two terms that reduce the uncertainty

$$\begin{aligned} \sigma_\tau^2(\mathbf{x}') &= \underbrace{\sigma_{f_\tau(T)}^2 + \sigma_{n_\tau(T)}^2}_{\text{prior variance}} - \underbrace{k_\tau(\mathbf{x}', X_T) C_{TT}^{-1} k_\tau(X_T, \mathbf{x}')}_{\text{Reduction in uncertainty due to target data}} \\ &\quad - \underbrace{[k_\tau(\mathbf{x}', X_T) C_{TT}^{-1} K_{TS} - k_\tau(\mathbf{x}', X_S)] [C_{SS} - K_{ST} C_{TT}^{-1} K_{TS}]^{-1} [K_{ST} C_{TT}^{-1} k_\tau(X_T, \mathbf{x}') - k_\tau(X_S, \mathbf{x}')] }_{\text{Reduction in uncertainty due to transfer}} \end{aligned}$$

Fig. 5.7 shows that the reduction due transfer goes to zero when a target sample is co-located with a source sample. This section analyzes the root of this interesting behavior.

To understand how training samples affect variance predictions, we examine the interactions between individual samples with an unseen input sample  $\mathbf{x}'$ . The correlations of  $\mathbf{x}'$  with a source task sample  $\mathbf{x}_S$  and a target task sample  $\mathbf{x}_T$ , for a given set of source and FAT-GP hyperparameters, are given by

$$\begin{aligned} k_\tau(\mathbf{x}', \mathbf{x}_T) &= c_{\tau(T)} \sigma_{f_\tau(T)}^2 \\ k_\tau(\mathbf{x}', \mathbf{x}_S) &= \lambda c_S \sqrt{\sigma_{f_\tau(T)}^2 \sigma_{f_S}^2} \sqrt{\frac{2l_{\tau(T)} l_S}{l_{\tau(T)}^2 + l_S^2}} \end{aligned}$$

where

$$\begin{aligned} c_S &= \exp \left[ -\frac{\|\mathbf{x}' - \mathbf{x}_S\|^2}{l_{\tau(T)}^2 + l_S^2} \right] \\ c_{\tau(T)} &= \exp \left[ -\frac{\|\mathbf{x}_T - \mathbf{x}'\|^2}{2l_{\tau(T)}^2} \right] \end{aligned}$$

Their correlation of the source and target samples can also be calculated as

$$\begin{aligned} k(\mathbf{x}_T, \mathbf{x}_S) &= \lambda \sqrt{\sigma_{f_\tau(T)}^2 \sigma_{f_S}^2} \sqrt{\frac{2l_{\tau(T)} l_S}{l_{\tau(T)}^2 + l_S^2}} \exp \left[ -\frac{\|\mathbf{x}_T - \mathbf{x}_S\|^2}{l_{\tau(T)}^2 + l_S^2} \right] \\ &= \lambda c_{TS} \sqrt{\sigma_{f_\tau(T)}^2 \sigma_{f_S}^2} \sqrt{\frac{2l_{\tau(T)} l_S}{l_{\tau(T)}^2 + l_S^2}} \end{aligned}$$

where

$$c_{TS} = \exp \left[ -\frac{\|\mathbf{x}_T - \mathbf{x}_S\|^2}{l_{\tau(T)}^2 + l_S^2} \right]$$

Thus, the reduction due to transfer (last term in Equation (5.22)) is be written as

$$K_{ST}C_{TT}^{-1}k_{\tau}(X_T, \mathbf{x}') - k_{\tau}(X_S, \mathbf{x}') = \lambda b \sqrt{\sigma_{f\tau(T)}^2 \sigma_{fS}^2} \sqrt{\frac{2l_{\tau(T)}l_S}{l_{\tau(T)}^2 + l_S^2}}$$

where  $b$  is a coefficient of transfer and is given by

$$b = \frac{c_{TS}c_{\tau(T)}\sigma_{f\tau(T)}^2 - c_S\sigma_{f\tau(T)}^2 - c_S\sigma_{n\tau(T)}^2}{\sigma_{f\tau(T)}^2 + \sigma_{n\tau(T)}^2}$$

Depending on the relative positions of  $\mathbf{x}_S$ ,  $\mathbf{x}_T$ , and  $\mathbf{x}'$ , the coefficient of transfer will differ. When the source and the target sample are co-located with  $\mathbf{x}'$ ,  $c_{TS} = c_S = c_T = 1$ . Then the coefficient of transfer is given by

$$b = -\frac{\sigma_{n\tau(T)}^2}{\sigma_{f\tau(T)}^2 + \sigma_{n\tau(T)}^2}$$

When  $\mathbf{x}'$  is outside the source's distance of correlation, i.e.  $c_S = 0$ , it influences  $\mathbf{x}'$  via the target samples close to it. Here the coefficient of transfer is given by

$$b = \frac{c_{TS}c_{\tau(T)}\sigma_{f\tau(T)}^2}{\sigma_{n\tau(T)}^2 + \sigma_{f\tau(T)}^2}$$

When  $\mathbf{x}'$  is outside the distance of correlation of all target samples i.e.  $c_{\tau(T)} = 0$ , but is close to a source sample, the source will impact  $\mathbf{x}'$  proportional to

$$b = -c_S$$

Thus, in the absence of target samples in the region of  $\mathbf{x}'$ , the variance prediction depends directly on correlation between  $\mathbf{x}_S$  and  $\mathbf{x}'$ .