

# Decentralized Gaussian Process Classification and an Application in Subsea Robotics

Yifei Gao  
*Virginia Tech*  
yifeig22@vt.edu

Hans J. He  
*Virginia Tech*  
hjh2bs@vt.edu

Daniel J. Stilwell  
*Virginia Tech*  
stilwell@vt.edu

James McMahon  
*US Naval Research Laboratory*  
Acoustics Division, Code 7130  
james.mcmahon@nrl.navy.mil

**Abstract**—Teams of cooperating autonomous underwater vehicles (AUVs) rely on acoustic communication for coordination, yet this communication medium is constrained by limited range, multi-path effects, and low bandwidth. One way to address the uncertainty associated with acoustic communication is to learn the communication environment in real-time. We address the challenge of a team of robots building a map of the probability of communication success from one location to another in real-time. This is a decentralized classification problem – communication events are either successful or unsuccessful – where AUVs share a subset of their communication measurements to build the map. The main contribution of this work is a rigorously derived data sharing policy that selects measurements to be shared among AUVs. We experimentally validate our proposed sharing policy using real acoustic communication data collected from teams of Virginia Tech 690 AUVs, demonstrating its effectiveness in underwater environments.

## I. INTRODUCTION

We address challenges imposed by teams of cooperating autonomous underwater vehicles that seek to coordinate their actions underwater. Communication between AUVs is necessary to support coordination, but the acoustic communication channel available to AUVs is extremely low-bandwidth often unreliable [1] [2]. To maximize the ability of AUVs to collaborate in real-time, we seek a method for the AUVs to cooperatively build a map that can be used to predict the probability of communication success. We address this goal as a Gaussian process classification problem that seeks to estimate the probability of successful or unsuccessful communication from a transmitting AUV at a specific location to a receiving AUV at another location. Implementing Gaussian process classification across a team of agents that have only very low bandwidth communication channels between agents requires novel advances in decentralized Gaussian process classification, which is the principal contribution of our work.

In practice, a team of AUVs can be pre-programmed with the maximum range at which they can successfully communicate. This range is necessarily conservative. Instead, we seek an approach to a team of AUVs developing a map of

This work was supported by the Seale Coastal Zone Observatory and the Office of Naval Research via grants N00014-23-1-2345, N00014-24-1-2267, and N00014-25-1-2224.

Yifei Gao, Hans J. He, and Daniel Stilwell are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, James McMahon is with the Acoustics Division, Code 7130, US Naval Research Laboratory, Washington, D.C.



Fig. 1: Virginia Tech 690 AUVs

the probability of communication success in real-time based on their current experience attempting to communicate in the environment. A conservative range can be used as a prior, and the possibility of communicating at longer ranges can be learned in real-time. In past work, we have investigated Gaussian process classification in a centralized manner for this application [3]. In this work, we explicitly address the challenge of decentralized Gaussian process classification (GP) so that the team of AUVs can learn communication success maps in real-time. Toward this goal, we develop fundamental advances in GP classification that can be applied to other application.

In the literature, most of the work including our prior work [2], [4]–[9] use Gaussian process regression to predict received signal strength (RSS) or use a manifold learning algorithm to predict signal to noise ratio (SNR) [10]. In [3], we address the application using GP classification, which allows us to explicitly learn and predict the variable of interest, which is the probability of communication success, rather than a proxy variable, such as SNR. Our prior work [3] employs Gaussian process classification to generate a communication map among two AUVs in a centralized manner suitable for post-processing. We also take advantage of the properties of Pólya-Gamma random variables in [11] to make prior conjugate, which is not otherwise the case for GP classification.

We use sparse Gaussian process classification to represent the communication map. Due to the low-bandwidth communication channel, the map should be represented with as few measurements of virtual measurements as possible.

Gaussian process classification is used in a variety of applications that range from image classification [12] to collision prediction for robot motion planning [13]. GP classification is widely used in active learning for label prediction [14]. A computation challenge of GP classification arises from the prior being Bernoulli and therefore non-conjugate. Recent work that addresses this challenge with clever use of Pólya-Gamma random variables [15] [16] [17], which we adopt in our work. Our approach to sparse GP classification, which builds upon the seminal works [15] [16] [18]. The use of sparse GP classification is because we seek a minimal representation of the environment due to the low bandwidth communication channel available underwater.

The contributions of this work are

- We provide a rigorously derived policy selecting data to share among agents for decentralized sparse GP classification under limited communication bandwidth.
- For an application in underwater communication, We show empirically the efficacy of our data sharing policy. We conduct numerical experiments using real data from field trials with teams of small AUVs that communicate acoustically.

The remainder of the paper is as follows. In Section II we review sparse GP classification with Pólya-Gamma random variables. In Section III, we derive an upper bound for the KL divergence between the estimated posterior and true posterior which we use to establish a data sharing policy. In Section IV we illustrate and evaluate our results on real communication data set acquired by a team of AUVs.

## II. PRELIMINARIES

In this section, we briefly review sparse Gaussian process binary classification with Pólya-Gamma random variables. For further details are found in [19], [15], [11], [20], [21].

### A. Gaussian Process Binary Classification

In standard Gaussian process (GP) classification, we consider learning a latent function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  from measurements  $\{\mathbf{x}, y\}$  where  $\mathbf{x} \in \mathbb{R}^d$  and binary label  $y \in \{0, 1\}$ . We assume this latent function is sampled from a Gaussian process  $\mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$  where  $\mu$  denotes the latent mean function and  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a kernel function. The posterior closed-form formula for Gaussian process regression (i.e., learning  $f$ ) is covered in [19].

For GP binary classification, we want to learn the probability distribution

$$p(y_* = 1 | \mathbf{y}, \mathbf{x}_*) = \int p(y_* = 1 | f_*) p(f_* | \mathbf{y}, \mathbf{x}_*) df_* \quad (1)$$

where  $\mathbf{y}$  is the set of all training measurement labels and  $\mathbf{x}_*$  is the test input. The conditional likelihood  $p(y_n | f_n)$  is defined as a logistic function

$$p(y_n | f_n) = \left( \frac{e^{f_n}}{1 + e^{f_n}} \right)^{y_n} \left( 1 - \frac{e^{f_n}}{1 + e^{f_n}} \right)^{(1-y_n)}$$

which is non-Gaussian. The logistic function maps the range of  $f(\mathbb{R})$  to the interval  $[0, 1]$ , inducing a valid probabilistic interpretation. We want to predict the probability of

a successful communication event given any test location  $\mathbf{x}_*$ . The latent posterior  $p(f_* | \mathbf{y}, \mathbf{x}_*)$  on (1) is non-Gaussian since the likelihood is non-conjugate. However, the logistic likelihood can be made conjugate by introducing a Pólya-Gamma random variable [11].

### B. Gaussian process classification with Pólya-Gamma (PG) random variable

As proposed in [11], Pólya-Gamma random variable  $w \sim PG(b, c)$  with probability density function

$$p(w | b, c) = \frac{1}{2\pi^2} \sum_{k=1}^{\infty} \frac{g_k}{(k - \frac{1}{2})^2 + (\frac{c}{2\pi})^2}$$

where  $b > 0$ ,  $c \in \mathbb{R}$  and  $g_k \sim Ga(b, 1)$  are independent Gamma random variables. We assume an independent observation model conditioned with PG random variables  $\mathbf{w}$ ,

$$p(\mathbf{y} | \mathbf{w}, \mathbf{f}) = \prod_{n=1}^N p(y_n | w_n, f_n)$$

with

$$p(y_n | w_n, f_n) = \frac{1}{2} \exp \left\{ -\frac{w_n}{2} \left( f_n^2 - 2f_n \frac{\kappa_n}{w_n} \right) \right\}$$

where  $w_n \sim PG(1, c_n)$  and  $\kappa_n = y_n - 1/2$ . Based on the Laplace transform property of Pólya-Gamma random variable [11], the joint conditional likelihood with PG variables is proportional to a Gaussian distribution

$$p(\mathbf{y} | \mathbf{w}, \mathbf{f}) \propto \mathcal{N}(\boldsymbol{\Omega}^{-1} \boldsymbol{\kappa} | \mathbf{f}, \boldsymbol{\Omega}^{-1})$$

where  $\boldsymbol{\Omega} = diag(\mathbf{w})$  is a diagonal matrix of Pólya-Gamma variables and  $\boldsymbol{\kappa} = [\kappa_1, \dots, \kappa_N]^T \in \mathbb{R}^N$ . The prior of latent function is assumed  $p(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \mathbf{0}, \mathbf{K}_{NN})$  where  $\mathbf{K}_{NN}$  is the Gram matrix taking total measurements  $\mathbf{X}$  as input. Conditioned on  $\mathbf{y}$  and the PG random variables  $\mathbf{w}$ , the posterior over the latent function values is also Gaussian

$$p(\mathbf{f} | \mathbf{y}, \mathbf{w}) = \mathcal{N}(\mathbf{f} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \boldsymbol{\Sigma} \boldsymbol{\kappa}, \quad \boldsymbol{\Sigma} = (\mathbf{K}_{NN}^{-1} + \boldsymbol{\Omega})^{-1}$$

### C. Sparse Gaussian process with Pólya-Gamma Variables

Computing the posterior distribution  $p(\mathbf{f} | \mathbf{y}, \mathbf{w})$  requires the  $\mathcal{O}(N^3)$  procedure of inverting the Gram matrix  $\mathbf{K}_{NN}$ . The author of [20] proposes sparse Gaussian process to reduce computational complexity to  $\mathcal{O}(M^3)$ , introducing  $M$  inducing points to summarize total data set with size  $N$  where  $M \ll N$ . Denote the inducing locations as  $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_M]^T$ ,  $\mathbf{z}_i \in \mathbb{R}^d$ , and let  $\mathbf{f}_M = [f_1(\mathbf{z}_1), \dots, f_M(\mathbf{z}_M)]^T$  be the inducing output vector. The conditional prior on latent function values  $\mathbf{f}$  given  $\mathbf{f}_M$  is

$$p(\mathbf{f} | \mathbf{f}_M) = \mathcal{N}(\mathbf{f} | \mathbf{K}_{\mathbf{X}, \mathbf{Z}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \mathbf{f}_M, \mathbf{K}_{\mathbf{X}, \mathbf{X}} - \mathbf{K}_{\mathbf{X}, \mathbf{Z}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}, \mathbf{X}}^T)$$

where kernel matrix  $\mathbf{K}_{\mathbf{X}, \mathbf{Z}} \in \mathbb{R}^{N \times M}$  and  $\mathbf{K}_{\mathbf{Z}, \mathbf{Z}} \in \mathbb{R}^{M \times M}$ . Note that now we only need to invert kernel matrix  $\mathbf{K}_{\mathbf{Z}, \mathbf{Z}}$ . In [20], the author shows that we can estimate the latent posterior as  $p(\mathbf{f} | \mathbf{y}) \approx \int p(\mathbf{f} | \mathbf{f}_M) q(\mathbf{f}_M) d\mathbf{f}_M$  using the variational

distribution  $q(\mathbf{f}_M)$ . However, in standard GP classification, the variational distribution for inducing outputs does not have a closed-form expression. In next section, we show that while conditioning on Pólya-Gamma random variables  $\boldsymbol{\omega}$ , we can compute a closed-form expression for  $q(\mathbf{f}_M|\mathbf{w}) = \mathcal{N}(\mathbf{f}_M|\boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Sigma}_{\mathbf{w}})$  where  $\boldsymbol{\mu}_{\mathbf{w}}$  and  $\boldsymbol{\Sigma}_{\mathbf{w}}$  are the mean vector and covariance matrix conditioned on  $\mathbf{w}$ . This yields a closed-form formula for the latent posterior conditioned on Pólya-Gamma variables  $p(\mathbf{f}|\mathbf{y}, \mathbf{w})$ ,

$$p(\mathbf{f}|\mathbf{y}, \mathbf{w}) \approx \int p(\mathbf{f}|\mathbf{f}_M)q(\mathbf{f}_M|\mathbf{w})d\mathbf{f}_M = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{S})$$

where

$$\begin{aligned}\mathbf{m} &= \mathbf{K}_{\mathbf{X}, \mathbf{Z}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \boldsymbol{\mu}_{\mathbf{w}} \\ \mathbf{S} &= \mathbf{K}_{\mathbf{X}, \mathbf{X}} - \mathbf{K}_{\mathbf{X}, \mathbf{Z}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{Z}, \mathbf{X}}^T + \mathbf{K}_{\mathbf{X}, \mathbf{Z}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \boldsymbol{\Sigma}_{\mathbf{w}} \mathbf{K}_{\mathbf{Z}, \mathbf{Z}}^{-1} \mathbf{K}_{\mathbf{X}, \mathbf{Z}}^T\end{aligned}$$

### III. SELECTION OF INDUCING POINTS

In this section, we first derive the upper bound for KL divergence between the estimated and true posterior inspired from [18]. Then we show this upper bound can be used as to select inducing points to share. To see the detailed derivation of the Kullback-Leibler(KL) divergence upper bound in Gaussian process regression, please refer to [18].

#### A. Lower and Upper bound for Log Marginal likelihood

We assume that we are given a set of known measurements of communication success acquired by an agent (an AUV, in our case) that consist of locations  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$  and a corresponding labels  $y_n \in \{0, 1\}$ . By assuming independent observations, the conditional likelihood can be written

$$p(\mathbf{y}|\mathbf{w}, \mathbf{f}) = \prod_{n=1}^N p(y_n|w_n, f_n) = C\mathcal{N}(\boldsymbol{\Omega}^{-1}\boldsymbol{\kappa}|\mathbf{f}, \boldsymbol{\Omega}^{-1})$$

where  $C$  is a unknown constant that arises from the joint conditional likelihood being proportional to a Gaussian distribution. The marginal likelihood conditioned on Pólya-Gamma random variables  $\mathbf{w}$  is

$$p(\mathbf{y}|\mathbf{w}) = \int p(\mathbf{y}|\mathbf{w}, \mathbf{f})p(\mathbf{f})d\mathbf{f} = C\mathcal{N}(\boldsymbol{\Omega}^{-1}\boldsymbol{\kappa}|\mathbf{0}, \boldsymbol{\Omega}^{-1} + \mathbf{K}_{NN}) \quad (2)$$

Similarly, we follow the derivation in [20] and [21] to derive a lower and upper bound for  $\log p(\mathbf{y}|\mathbf{w})$  which is used to compute upper bound of KL divergence between estimated and true posterior. We compute directly

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{w}) &\geq \log C - \frac{N}{2} \log(2\pi) \\ &\quad - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} (\boldsymbol{\Omega}^{-1}\boldsymbol{\kappa})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\Omega}^{-1}\boldsymbol{\kappa}) \\ &\quad - \frac{1}{2} \text{tr}(\boldsymbol{\Omega}\tilde{\mathbf{K}}) = \mathcal{L}_{lower}\end{aligned} \quad (3)$$

where  $\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1} + \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}$  and  $\tilde{\mathbf{K}} = \mathbf{K}_{NN} - \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}$ . Note that  $\mathcal{L}_{lower}$  lower bounds the log marginal likelihood, and if  $\mathbf{X} = \mathbf{Z}$ , then  $\text{tr}(\boldsymbol{\Omega}\tilde{\mathbf{K}}) = 0$  and  $\mathcal{L}_{lower} = \log p(\mathbf{y}|\mathbf{w})$ . We can show that while conditioning

on Pólya-Gamma variables  $\mathbf{w}$ , the optimal closed-form distribution for inducing variables  $q(\mathbf{f}_M|\mathbf{w})$  can be written as

$$q(\mathbf{f}_M|\mathbf{w}) = \mathcal{N}(\mathbf{f}_M|\mathbf{K}_{MM}\tilde{\boldsymbol{\Sigma}}^{-1}\mathbf{K}_{MN}\boldsymbol{\kappa}, \mathbf{K}_{MM}\tilde{\boldsymbol{\Sigma}}^{-1}\mathbf{K}_{MM}) \quad (4)$$

where  $\tilde{\boldsymbol{\Sigma}} = \mathbf{K}_{MM} + \mathbf{K}_{MN}\boldsymbol{\Omega}\mathbf{K}_{NM}$  and  $\boldsymbol{\kappa}_n = y_n - \frac{1}{2}$  is one of the entries in vector  $\boldsymbol{\kappa}$  with  $y_n \in \{0, 1\}$ .

Following the derivation in [21], we can write an upper bound for log marginal likelihood conditioned on Pólya-Gamma variable,

$$\begin{aligned}\log p(\mathbf{y}|\mathbf{w}, \mathbf{X}) &\leq \log C - \frac{N}{2} \log(2\pi) \\ &\quad - \frac{1}{2} \log |\boldsymbol{\Omega}^{-1} + \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}| \\ &\quad - \frac{1}{2} \mathbf{c}^T (\boldsymbol{\Omega}^{-1} + p\mathbf{I} + \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN})^{-1} \mathbf{c} \\ &= \mathcal{L}_{upper}\end{aligned} \quad (5)$$

where  $\mathbf{c} = \boldsymbol{\Omega}^{-1}\boldsymbol{\kappa}$  and  $p = \text{tr}(\tilde{\mathbf{K}})$ . If  $\mathbf{X} = \mathbf{Z}$ , then we again arrive at  $\mathcal{L}_{upper} = \log p(\mathbf{y}|\mathbf{w})$ .

#### B. KL Divergence Upper bound

Following the analysis in [18], an upper bound is derived for KL divergence between the estimated posterior  $q(\mathbf{f}, \mathbf{f}_M) = p(\mathbf{f}|\mathbf{f}_M)q(\mathbf{f}_M|\mathbf{w})$  and true posterior  $p(\mathbf{f}, \mathbf{f}_M|\mathbf{w}, \mathbf{y}) = p(\mathbf{f}|\mathbf{f}_M)p(\mathbf{f}_M|\mathbf{w}, \mathbf{y})$  which is denoted as  $KL[q(\mathbf{f}, \mathbf{f}_M)||p(\mathbf{f}, \mathbf{f}_M|\mathbf{w}, \mathbf{y})] \triangleq KL[Q_w||P_w]$ . We don't know the true inducing posterior conditioned on Pólya-Gamma variables  $\mathbf{w}$  denoted as  $p(\mathbf{f}_M|\mathbf{w}, \mathbf{y})$  but we can use  $q(\mathbf{f}_M|\mathbf{w})$  to approximate it. This KL divergence quantifies how close we are to the true posterior conditioned on  $\mathbf{w}$ . Let log marginal likelihood conditioned on  $\mathbf{w}$  denoted as  $\mathcal{L} = \log p(\mathbf{y}|\mathbf{w})$ . From (3) and (5), we can have  $\mathcal{L}_{lower} \leq \mathcal{L} \leq \mathcal{L}_{upper}$  and we can also directly show that  $\log p(\mathbf{y}|\mathbf{w}) = KL[Q_w||P_w] + \mathcal{L}_{lower}$  from which we write the upper bound  $KL[Q_w||P_w] \leq \mathcal{L}_{upper} - \mathcal{L}_{lower}$ . Note that the constant  $C$  in the joint conditional likelihood  $p(\mathbf{y}|\mathbf{w}, \mathbf{f})$  is eliminated in the difference between upper and lower bound of the log marginal likelihood  $\log p(\mathbf{y}|\mathbf{w})$ . Therefore, we can write

$$KL[Q_w||P_w] \leq \frac{1}{2} \text{tr}(\tilde{\mathbf{K}}) \lambda_{\max}(\boldsymbol{\Omega}) + \frac{1}{2\lambda_{\min}(\boldsymbol{\Omega})} \|\boldsymbol{\kappa}\|_2^2$$

where  $\tilde{\mathbf{K}} = \mathbf{K}_{NN} - \mathbf{K}_{NM}\mathbf{K}_{MM}^{-1}\mathbf{K}_{MN}$ .  $\lambda_{\max}(\boldsymbol{\Omega})$  and  $\lambda_{\min}(\boldsymbol{\Omega})$  are maximum and minimum eigenvalue of diagonal matrix  $\boldsymbol{\Omega}$  hence largest and smallest corresponding entries in the diagonal. Since we assume that we are given Pólya-Gamma random variables  $\mathbf{w}$  to form  $\boldsymbol{\Omega}$ , only  $\text{tr}(\tilde{\mathbf{K}})$  and  $\|\boldsymbol{\kappa}\|_2^2$  affect the KL divergence upper bound. We cannot control  $\|\boldsymbol{\kappa}\|_2^2$  since it scales with size  $N$  of the current data set.

Selecting inducing points which minimize  $\text{tr}(\tilde{\mathbf{K}})$  is non-trivial but [18] [21] [22] [23] propose determinantal point process and ridge leverage scores to select  $m \leq M$  number of inducing points in an acceptable amount of computation time. However, in this paper, we are only interested in selecting small number of inducing points where  $m \leq 2$  due to limited communication bandwidth.

### C. Communication Among Multiple agents

In this section we examine precisely what information should be shared among agents. We assume that each agent is within its own locality region and denote local measurements obtained by agent  $i$  be  $\{\mathbf{X}_i, \mathbf{y}_i\}$ . We assume each agent's locality region is independent, and that each agent selects inducing points directly from their measurement locations  $\mathbf{X}_i$ . Each agent's set of inducing points  $\mathbf{Z}_i$  are therefore independent with respect to those of other agents. Suppose agent  $i$  chooses  $m_i$  inducing points from its current inducing set  $M_i$  in its locality region which minimizes  $tr(\tilde{\mathbf{K}}_i)$ ,

$$tr(\tilde{\mathbf{K}}_i) = tr(\mathbf{K}_{N_i N_i} - \mathbf{K}_{N_i m_i} \mathbf{K}_{m_i m_i}^{-1} \mathbf{K}_{m_i N_i})$$

where  $N_i$  denotes the total number of local data set size obtained from agent  $i$  in the locality region. After selecting  $m_i$  number of inducing points, agent  $i$  computes the variational distribution  $q(\mathbf{f}_{m_i} | \mathbf{w}_i) = \mathcal{N}(\mathbf{f}_{m_i} | \boldsymbol{\mu}_{m_i}, \boldsymbol{\Sigma}_{m_i})$  using (4) with  $\mathbf{K}_{MM}$  and  $\mathbf{K}_{NM}$  replaced by  $\mathbf{K}_{m_i m_i}$  and  $\mathbf{K}_{N_i m_i}$ . The information package which each agent needs to share is  $\{\mathbf{Z}_{m_i}, q(\mathbf{f}_{m_i} | \mathbf{w}_i)\}$ , where  $\mathbf{Z}_{m_i}$  denotes  $m_i$  inducing locations selected to minimize  $tr(\tilde{\mathbf{K}}_i)$ . Based on the upper limit of the KL divergence derived Section III-B, agent  $i$  selects  $m_i$  inducing points and computes  $q(\mathbf{f}_{m_i} | \mathbf{w}_i)$ , which minimizes the upper bound of KL divergence in its locality region.

### D. Prediction

For the case of  $n$  agents, each agent summarizes its locality region with  $q(\mathbf{f}_{m_i} | \mathbf{w}_i)$ , which it shares with other agents. We denote the complete set of inducing variables as  $\mathbf{f}_M = [\mathbf{f}_{m_1}, \dots, \mathbf{f}_{m_n}]$  where  $\mathbf{f}_{m_i} = f(\mathbf{Z}_{m_i})$  are the latent function vectors taking inducing locations  $\mathbf{Z}_{m_i}$  as input. Then we can write the joint variational distribution after each agent  $i$  receives information packages  $\{\mathbf{Z}_{m_j}, q(\mathbf{f}_{m_j} | \mathbf{w}_j)\}$  where  $i \neq j$  as  $q(\mathbf{f}_M | \mathbf{w}) = \mathcal{N}(\mathbf{f}_M | \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M)$  where

$$\boldsymbol{\mu}_M = \begin{bmatrix} \boldsymbol{\mu}_{m_1} \\ \vdots \\ \boldsymbol{\mu}_{m_i} \\ \vdots \\ \boldsymbol{\mu}_{m_n} \end{bmatrix}, \boldsymbol{\Sigma}_M = \begin{bmatrix} \boldsymbol{\Sigma}_{m_1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\Sigma}_{m_i} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \boldsymbol{\Sigma}_{m_n} \end{bmatrix} \quad (6)$$

where  $\boldsymbol{\mu}_M \in \mathbb{R}^{M \times 1}$  and  $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times M}$  where  $M = m_1 + m_2 + \dots + m_n$ . Note this decentralized formulation (block-diagonal structure) is due to assuming independence between the locality region for each AUV, which we arrive at by noting that the effect of measurements outside of a locality region are vanishingly small. Therefore, the distribution for the value of the latent function at test location  $\mathbf{x}_*$  is

$$\begin{aligned} p(f_* | \mathbf{y}, \mathbf{w}) &\approx \int p(f_* | \mathbf{f}_M) q(\mathbf{f}_M | \mathbf{w}) d\mathbf{f}_M \\ &= \mathcal{N}(f_* | \mathbf{k}_{n* M} \mathbf{K}_{MM}^{-1} \mathbf{f}_M, k_{n* n*} - \mathbf{k}_{n* M} \mathbf{K}_{MM}^{-1} \mathbf{k}_{M n*}) \end{aligned}$$

where  $p(f_* | \mathbf{f}_M)$  is the Gaussian conditional prior over  $f_*$  which has a closed-form expression and  $\mathbf{k}_{n* M}$  is a vector in

$\mathbb{R}^{1 \times M}$ . Then the probability of success on test location  $\mathbf{x}_*$  given training data  $\mathbf{y}$  is approximated as

$$p(y_* = 1 | \mathbf{y}, \mathbf{w}) \approx \int p(y_* | f_*) p(f_* | \mathbf{y}, \mathbf{w}) df_*$$

which can be computed by 1-D Gaussian quadrature.

## IV. NUMERICAL RESULTS

We evaluate and illustrate our approach to decentralized sparse GP classification using data acquired from previous field trials with teams of Virginia Tech 690 autonomous underwater vehicles (see Figure 1). The 690 AUV displaces approximately 43Kg and is 2.23 meters long. It communicates acoustically using the WHOI Micromodem 2, which operates at 25KHz. Experiments were performed with two AUVs in Claytor Lake, near Dublin, Virginia, USA, and with two and three AUVs in Massachusetts Bay near Boston, Massachusetts, USA. The AUVs communicate using time-division multiple access communication; each AUV broadcasts to all other AUVs sequentially, with each broadcasting AUV assigned a sequence of known time-intervals. Therefore if a receiving AUV does not receive a data packet during a time when it is expected, the occurrence of an unsuccessful communication event can be noted and recorded.

The location of a communication event is represented by  $\mathbf{x} \in \mathbb{R}^4$ , where the first two dimensions indicate the easting and northing coordinates of the broadcasting vehicle in the horizontal plane, and the last two dimensions contain the easting and northing coordinates of the receiving vehicle. For the data presented herein, all AUVs operated within a few meters of the same depth, so depth variations are not considered. Distance between AUVs is not explicitly included as a parameter since the relative positions of transmitting and receiving AUVs implicitly encode this information. To reduce computational complexity and enable straightforward visualization of communication events on 2D plots, environmental factors such as multipath effects and water current speed are not incorporated in this model. These factors will be addressed in future work.

We note that communication success measurements are obtained by all AUVs that receive or fail to receive a communication packet from a broadcasting AUV. The broadcasting AUV cannot assess success or failure. The broadcasting AUV can be informed of a receiving AUV's communication measurements, but only via direct communication of the measurement from the receiving AUV. Of course, the acoustic communication channel available to the AUVs is very low bandwidth, and we presume that all communication measurements cannot be shared among the team of AUVs. This motivates our development of a data-sharing policy (see Section III) for decentralized sparse Gaussian process classification.

We evaluate our decentralized sparse GP classification approach using three datasets as detailed in TABLE I. The column **Agent** indicates the specific AUV involved in each dataset. For each AUV, the *Unsuccessful Events* and *Successful Events* columns represent the total number of failed

and successful communication attempts when that particular AUV acted as the receiving vehicle. These measurements were obtained by the AUV, and are therefore stored locally onboard the AUV. For instance, in **Dataset 3**, AUV 1 successfully received 119 data packets from AUV 2 or AUV 3, and it failed to receive 132 data packets from AUV 2 and AUV 3.

Using the data acquired from the field, we implement decentralized GP classification by sharing inducing points (i.e. virtual measurements in each broadcast data packet), and we choose which measurements to share using three different strategies. We select inducing points that minimize  $\text{tr}(\tilde{\mathbf{K}})$  as described on Section III, and we compare this approach to randomly selecting inducing points, and to selecting inducing points that maximize  $\text{tr}(\mathbf{K})$ , which our analysis suggests should be the least useful inducing points. We address two cases: in the first we share one inducing point, and in the second we share two inducing points. While an acoustic modem data packet may have room for more inducing points, the data packet is usually needed for communicating other missions-specific data. Thus we believe our experiments with one and two inducing points is representative of real-world constraints.

We evaluate the classification prediction accuracy with the ration  $\text{ACC} = \frac{N_{\text{correct}}}{N_{\text{total}}}$  where  $N_{\text{correct}}$  is the number of correct prediction.  $N_{\text{total}}$  is the total number of test points  $\mathbf{x}_*$ . If the true label is  $y = 1$ , and prediction gives a probability larger than 0.5, we treat it as a correct prediction and  $y = 0$  is addressed similarly. We assess the certainty of the prediction using

$$\text{NLL} = -\frac{1}{N_{\text{total}}} \sum_{i=1}^{N_{\text{total}}} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

For all three data sets, we use the squared exponential kernel function with unit signal variance, for any two inputs  $\mathbf{x}$  and  $\mathbf{x}'$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2l^2}\right)$$

We initially standardize all communication data sets to make them have zero mean and unit variance. Then we use GPflow [24] to train on all standardized data sets and determine an optimal length-scale parameter  $l = 0.289$ , which we then fix for all experiments across all agents. Note that real-time hyper-parameter learning is not explicitly addressed in this work. For each decentralized communication data set, we establish a reference radius  $r$  based on the squared exponential kernel function with the fixed length-scale. This radius is defined:

$$r = l \cdot \sqrt{-2 \cdot \log(\epsilon)}$$

with  $\epsilon$  tolerance where we have  $k(\mathbf{x}, \mathbf{x}') < \epsilon$  if  $\|\mathbf{x} - \mathbf{x}'\| > r$ . We use this radius to define a *locality region* within each agent's local data set. Each region is centered at one of the local measurements stored in the agent's data set and encompasses all points within distance  $r$  of this center. Note

that locality region  $R(r, \mathbf{c})$  with center  $\mathbf{c} \in \mathbb{R}^4$  which we defined is a ball containing all locations  $\mathbf{l} \in \mathbb{R}^4$  which is

$$R(r, \mathbf{c}) = \{\mathbf{l} \in \mathbb{R}^4 : \|\mathbf{l} - \mathbf{c}\|_2 \leq r\}$$

and  $\mathbf{c}$  is one of the measurement point obtained by AUV. Only data points inside this locality region are considered when training the sparse Gaussian process classification model.

Our approach assesses how effectively one or two inducing points can summarize the information provided by all measurements within a locality region. Before evaluating how our data sharing policy enables an AUV to make communication predictions in locality regions containing data points it has not yet encountered, we first conduct two local simulations by selecting inducing points within locality regions in each agent's local data set. If the selected good inducing points yield better predictions on test points within the locality, we can be confident that when an agent shares these inducing points with other agents, those received agents can make better predictions on this locality region even if they have not yet visited that region themselves.

For **Dataset 1** (see Table I) there are two AUVs. We partition the data corresponding to which agent received or failed to receive a communication event and label partitioned data **Agent 1 local data set** and **Agent 2 local data set**. We then perform 100 random permutations for train-test splits on each agent's local dataset. In each permutation, 65 percent of the data is allocated for training and 35 percent for testing. For every permutation, we identify a small locality region defined by a radius  $r = 0.4$  with a fixed center  $\mathbf{c}$  and we select inducing points from this fixed locality region. It's important to note that since the communication data was previously standardized, this radius  $r$  has no unit. We observe that increasing  $r$  would necessitate more than one or two inducing points to adequately summarize the information within the locality region. In subsequent visualizations, we revert to the raw communication events data, where coordinates of each AUV are expressed in meters with northing and easting positions.

Since it is a random permutation, each locality region encompasses distinct training data points. Within each locality region, each agent selects  $m = 1$  and  $m = 2$  inducing points that effectively summarize the data within that locality. We classify inducing point selection strategies as follows: *good* inducing point locations are selected by minimizing  $\text{tr}(\tilde{\mathbf{K}})$ , where we constrain selection to existing data point locations within the locality; *bad* inducing point locations maximize  $\text{tr}(\tilde{\mathbf{K}})$ ; and *random* inducing point locations are randomly selected from data points  $\mathbf{x}$  within the region.

After selecting the inducing locations  $\mathbf{Z}_{m_i}$ , each agent computes  $q(\mathbf{f}_{m_i} | \mathbf{w}_i)$  based on equation (4) and uses this to predict the unseen 35 percent test data points within each identified locality region. We assume that  $\mathbf{w}_i$  is available for each agent, we compute it via the Gibbs sampling technique described in [16], which involves sampling from the Pólya-Gamma posterior  $p(\mathbf{w}_i | \mathbf{f}_i)$  and  $p(\mathbf{f}_i | \mathbf{y}_i, \mathbf{w}_i)$  for a specified number of iterations to produce a sampled  $\mathbf{w}_i$ .

TABLE I: AUV Communication Events Across Different Datasets

| Dataset   | Location          | Agent | Unsuccessful Events | Successful Events |
|-----------|-------------------|-------|---------------------|-------------------|
| Dataset 1 | Massachusetts Bay | 1     | 248                 | 198               |
|           |                   | 2     | 219                 | 227               |
| Dataset 2 | Claytor Lake      | 1     | 366                 | 197               |
|           |                   | 2     | 336                 | 227               |
| Dataset 3 | Massachusetts Bay | 1     | 132                 | 119               |
|           |                   | 2     | 105                 | 130               |
|           |                   | 3     | 103                 | 109               |

Results for each agent selecting inducing points and evaluate on its test points inside locality region before sharing those inducing points are presented in Tables II and III. Note the values shown on the table are averaged over 100 since we have 100 random permutations. We observe that for **Agent 2**, selecting one or two good inducing points offers substantial prediction advantages over random or bad inducing points. This advantage is more pronounced compared to the **Agent 1**, as prediction performance is also strongly influenced by the distribution of local data within each locality region.

TABLE II: (Dataset 1: Agent 1 local data set)

| SGPC-PG                  | ACC                    | NLL                    |
|--------------------------|------------------------|------------------------|
| Good: select one / two   | <b>0.6498 / 0.6552</b> | 0.6685 / <b>0.6280</b> |
| Random: select one / two | 0.6458 / 0.6489        | 0.6546 / 0.6444        |
| Bad: select one / two    | 0.6427 / 0.6455        | 0.6558 / 0.6499        |

TABLE III: (Dataset 1: Agent 2 local data set)

| SGPC-PG                  | ACC                    | NLL                    |
|--------------------------|------------------------|------------------------|
| Good: select one / two   | <b>0.7152 / 0.7272</b> | <b>0.5902 / 0.5588</b> |
| Random: select one / two | 0.5747 / 0.6249        | 0.6706 / 0.6353        |
| Bad: select one / two    | 0.4838 / 0.4835        | 0.7004 / 0.7029        |

We visualize all training measurements on one locality region (see Figure 3) from **Agent 2 local data set** where

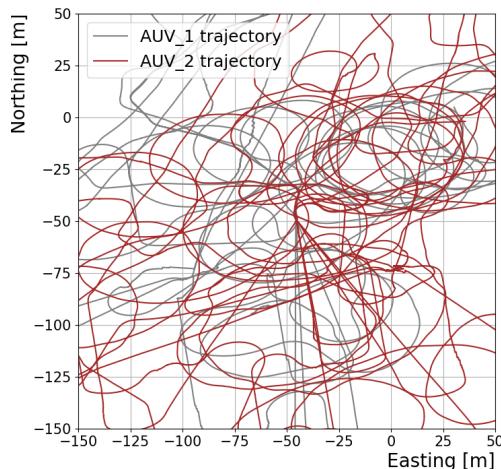


Fig. 2: Partial trajectories of two AUVs.

AUV 2 acts as the receiving vehicle. Note the locality region  $R(r, \mathbf{c})$  is not shown in the Figure 2 since it is a ball in  $\mathbb{R}^4$ . The arrows in the figure represent communication events where AUV 1 is broadcasting to AUV 2 so all arrows are pointing from a point along AUV 1 trajectory to a point along the trajectory of AUV 2. Since AUV 2 is the receiving AUV, all of these communication events are only stored onboard AUV 2. Green arrows indicate successful communication events perceived by the agent, while black arrows represent unsuccessful communication events. The agent selects two good/random/bad inducing locations based on the data inside this locality region. We observe that the selection of two good inducing points that minimizes  $tr(\tilde{\mathbf{K}})$  tends to identify locations near data cluster areas, enabling better representation of all data within this region. In contrast, bad inducing locations typically fall in areas that poorly represent the local data set within this region.

Having shown that selected *good* inducing points improve local prediction performance, we extend our analysis to multi-agent contexts. Results demonstrate that sharing *good* inducing points among decentralized agents consistently outperforms sharing *random* or *bad* inducing points. We evaluate performance when agents concatenate received inducing locations and variational distributions to form the decentralized posterior  $q(\mathbf{f}_M | w)$  Section (III-D), then predict on test points across all locality regions.

Our results clearly demonstrate that in the decentralized scenario, agents benefit from sharing good inducing points with each other under limited communication bandwidth, achieving increased accuracy and decreased negative test likelihood (NLL), see tables IV V and VI. We visualize test communication event predictions covered by two locality regions visited by both agents in the **Dataset 1** (see Figure 4). In these figures, green solid triangles indicate correct predictions (predictive probability  $p > 0.5$  if  $y = 1$  and  $p \leq 0.5$  if  $y = 0$ , while red solid triangles denote incorrect predictions. Although the two locality regions in Figure 4 are actually partially overlap, our prediction results still demonstrate that sharing good inducing points provides significant advantages over alternative approaches.

TABLE IV: (Dataset 1)

| SGPC-PG                 | ACC                    | NLL                    |
|-------------------------|------------------------|------------------------|
| Good: share one / two   | <b>0.6844 / 0.6941</b> | <b>0.6315 / 0.5815</b> |
| Random: share one / two | 0.5972 / 0.6504        | 0.6674 / 0.6296        |
| Bad: share one / two    | 0.5606 / 0.5620        | 0.6765 / 0.6765        |

TABLE V: (Dataset 2)

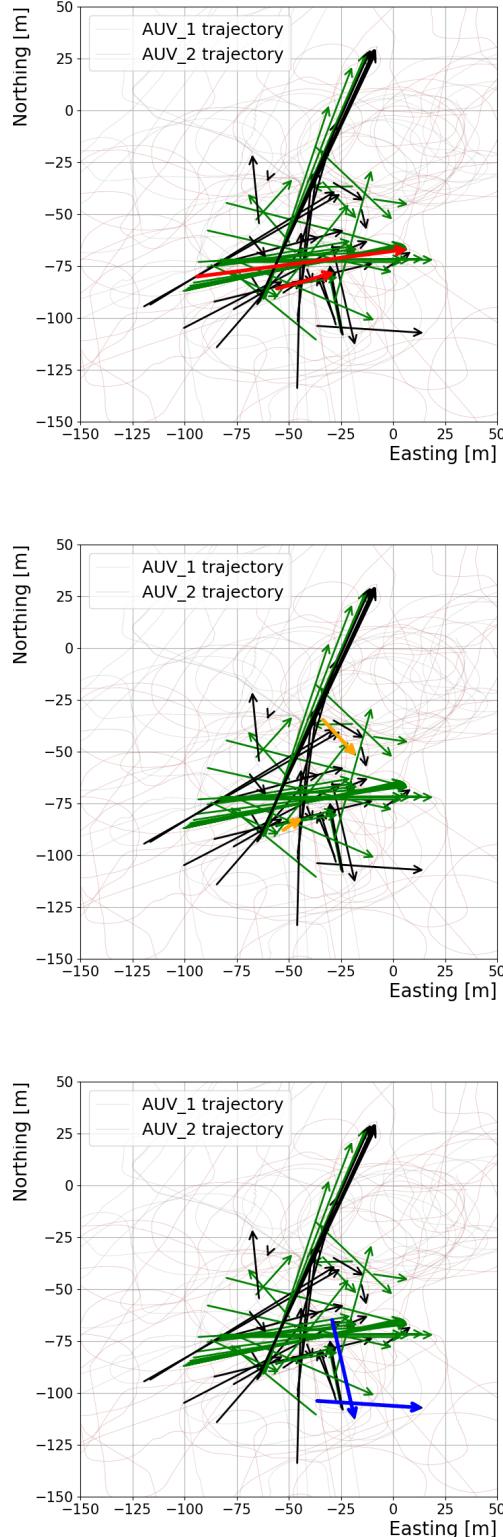


Fig. 3: Two AUVs trajectories (upper figure), Good(red arrows), random (yellow arrows)and bad (blue arrows )inducing points chosen by local agent inside a locality region.

TABLE VI: (Dataset 3)

| SGPC-PG                 | ACC             | NLL                    |
|-------------------------|-----------------|------------------------|
| Good: share one / two   | 0.7442 / 0.7442 | <b>0.5020 / 0.4918</b> |
| Random: share one / two | 0.7485 / 0.7493 | 0.5756 / 0.5327        |
| Bad: share one / two    | 0.7433 / 0.7437 | 0.6320 / 0.6111        |

TABLE VI: (Dataset 3)

| SGPC-PG                 | ACC                    | NLL                    |
|-------------------------|------------------------|------------------------|
| Good: share one / two   | <b>0.7101 / 0.7270</b> | <b>0.6267 / 0.5976</b> |
| Random: share one / two | 0.6979 / 0.6967        | 0.6433 / 0.6273        |
| Bad: share one / two    | 0.6947 / 0.6981        | 0.6550 / 0.6532        |

## V. CONCLUDING REMARKS

We present a decentralized data sharing policy that leverages the theoretical advantages of sparse Gaussian process classification with Pólya-Gamma random variables. Our results demonstrate that inducing points selected through this policy offer significant advantages over random and bad inducing points when constructing communication maps under bandwidth limitations. Future research directions include developing more sophisticated sharing policies for scenarios with overlapping locality regions and optimizing the utilization of previously received inducing points to enhance prediction accuracy. These advancements will further improve coordination capabilities among autonomous underwater vehicles operating in challenging communication environments.

## REFERENCES

- [1] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” *IEEE communications magazine*, vol. 47, no. 1, pp. 84–89, 2009.
- [2] D. Horner, *A data-driven framework for rapid modeling of wireless communication channels*. PhD thesis, Monterey, California: Naval Postgraduate School, 2013.
- [3] Y. Gao, H. Yetkin, M. James, and D. J. Stilwell, “Prediction of acoustic communication performance for auvs using gaussian process classification,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1244–1251, IEEE, 2024.
- [4] J. Zhao, X. Gao, X. Wang, C. Li, M. Song, and Q. Sun, “An efficient radio map updating algorithm based on k-means and gaussian process regression,” *The Journal of Navigation*, vol. 71, no. 5, pp. 1055–1068, 2018.
- [5] A. Quattrini Li, P. K. Penumarthy, J. Banfi, N. Basilico, J. M. O’Kane, I. Rekleitis, S. Nelakuditi, and F. Amigoni, “Multi-robot online sensing strategies for the construction of communication maps,” *Autonomous Robots*, vol. 44, pp. 299–319, 2020.
- [6] X. Wang, X. Wang, S. Mao, J. Zhang, S. C. G. Periaswamy, and J. Patton, “Deepmap: Deep gaussian process for indoor radio map construction and location estimation,” in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2018.
- [7] L. Clark, J. A. Edlund, M. S. Net, T. S. Vaquero, and A. akbar Aghamohammadi, “Propem-l: Radio propagation environment modeling and learning for communication-aware multi-robot exploration,” 2022.
- [8] G. P. Kontoudis, S. Krauss, and D. J. Stilwell, “Model-based learning of underwater acoustic communication performance for marine robots,” *Robotics and Autonomous Systems*, vol. 142, p. 103811, 2021.
- [9] G. P. Kontoudis and D. J. Stilwell, “A comparison of kriging and cokriging for estimation of underwater acoustic communication performance,” in *Proceedings of the 14th International Conference on Underwater Networks & Systems*, pp. 1–8, 2019.

- [10] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, "Snr prediction in cellular systems based on channel charting," in *2020 IEEE eighth international conference on communications and networking (ComNet)*, pp. 1–8, IEEE, 2020.
- [11] N. G. Polson, J. G. Scott, and J. Windle, "Bayesian inference for logistic models using polya-gamma latent variables," 2013.
- [12] Y. Bazi and F. Melgani, "Gaussian process approach to remote sensing image classification," *IEEE transactions on geoscience and remote sensing*, vol. 48, no. 1, pp. 186–197, 2009.
- [13] J. Muñoz, P. Lehner, L. E. Moreno, A. Albu-Schäffer, and M. A. Roa, "Collisiongp: Gaussian process-based collision checking for robot motion planning," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4036–4043, 2023.
- [14] G. Zhao, E. Dougherty, B.-J. Yoon, F. Alexander, and X. Qian, "Efficient active learning for gaussian process classification by error reduction," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9734–9746, 2021.
- [15] F. Wenzel, T. Galy-Fajou, C. Donner, M. Kloft, and M. Opper, "Efficient gaussian process classification using polya-gamma data augmentation," 2018.
- [16] I. Achituvé, A. Shamsian, A. Navon, G. Chechik, and E. Fetaya, "Personalized federated learning with gaussian processes," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8392–8406, 2021.
- [17] J. Snell and R. Zemel, "Bayesian few-shot classification with one-vs-each poly-augmented gaussian processes," *arXiv preprint arXiv:2007.10417*, 2020.
- [18] D. R. Burt, C. E. Rasmussen, and M. Van Der Wilk, "Convergence of sparse variational inference in gaussian processes regression," *Journal of Machine Learning Research*, vol. 21, no. 131, pp. 1–63, 2020.
- [19] C. E. Rasmussen, C. K. Williams, et al., *Gaussian processes for machine learning*, vol. 1. Springer, 2006.
- [20] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Artificial intelligence and statistics*, pp. 567–574, PMLR, 2009.
- [21] M. K. Titsias, "Variational inference for gaussian and determinantal point processes," in *Workshop on Advances in Variational Inference (NIPS)*, p. 14, 2014.
- [22] M.-A. Belabbas and P. J. Wolfe, "Spectral methods in machine learning and new strategies for very large datasets," *Proceedings of the National Academy of Sciences*, vol. 106, no. 2, pp. 369–374, 2009.
- [23] N. Anari, S. O. Gharan, and A. Rezaei, "Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes," in *Conference on Learning Theory*, pp. 103–115, PMLR, 2016.
- [24] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, "GPflow: A Gaussian process library using TensorFlow," *Journal of Machine Learning Research*, vol. 18, pp. 1–6, apr 2017.

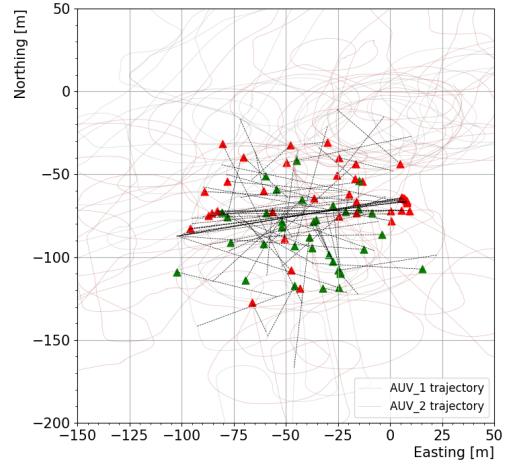
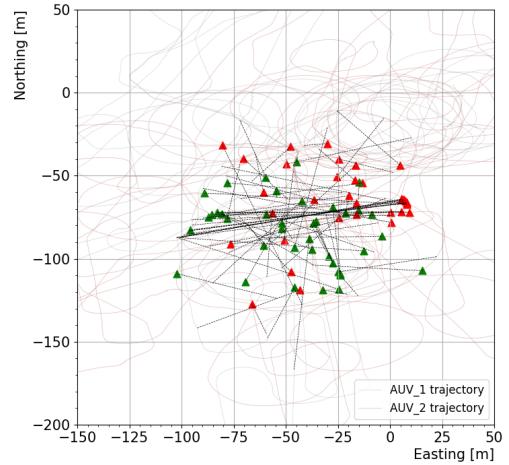
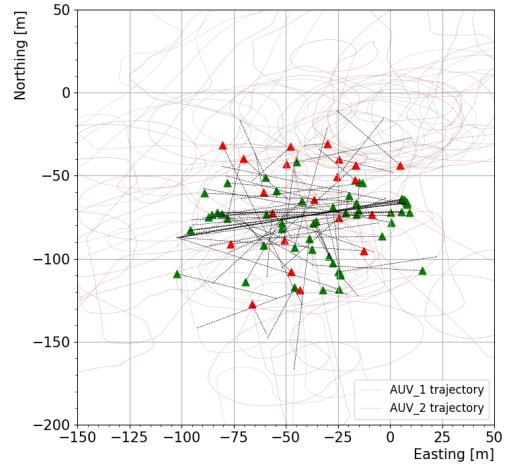


Fig. 4: Two agents decentralized prediction on test points from two different locality regions using good (upper figure)/random (middle figure)/bad (bottom figure) inducing points shared among each agent