2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

# Learning Recursive Bayesian Nonparametric Modeling of Moving Targets via Mobile Decentralized Sensors

Chang Liu, Yucheng Chen, Jake Gemerek, Hengye Yang and Silvia Ferrari

*Abstract*— **Bayesian nonparametric models, such as the Dirichlet Process Gaussian Process (DPGP), have been shown very effective at learning models of dynamic targets exclusively from data. Previous work on batch DPGP learning and inference, however, ceases to be efficient in multi-sensor applications that require decentralized measurements to be obtained sequentially over time. Batch processing, in this case, leads to redundant computations that may hinder online applicability. This paper develops a recursive approach for DPGP learning and inference in which a novel Dirichlet Process prior based on Wasserstein metric is used for measuring the similarity between multiple Gaussian Processes (GPs). Combined with the GP recursive fusion law, the proposed recursive DPGP fusion approach enables efficient online data fusion. The problem of active sensing for recursive DPGP learning and inference is also investigated by uncertainty reduction via expected mutual information. Simulation and experimental results show that the proposed approach successfully learns the models of moving targets and outperforms existing benchmark methods.**

## I. Introduction

Learning the behavior of moving targets using reconfigurable sensors, such as mobile cameras, is attracting increasing attention because of its wide variety of applications, ranging from environmental monitoring [1], [2], [3], to security and surveillance [4], [5], [6], and disaster response [7]. In these applications, the region of interest (ROI) greatly exceeds the size of the sensor's field of view (FOV) and, therefore, managing mobile sensor positions and orientations is crucial to collectively learn the behavior of many moving targets in the ROI. Unlike existing Kalman estimators and particle filters, the approach presented in this paper requires little or no prior information about target behavior, and is applicable when the number and dynamics of the targets of interest change over time, for example, as new targets enter and old targets leave the ROI [4], [6]. As a result, DPGP models provide a more flexible and systematic approach for modeling moving targets when compared to semi-Markov jump systems [8], linear stochastic models [9], and nonholonomic dynamics models [10], [11].

Bayesian nonparametric approaches, such as DPGPs, have been used for modeling a wide range of spatio-temporal phenomena thanks to their flexibility, expressiveness, and data-driven nature. GPs have been used to model ozone concentration [12], topology [13], and ocean salinity [14]. DPGPs have been used to model urban traffic [15], [16] and pedestrians [17] because of their ability to capture mixtures of behaviors. Previous GP and DPGP algorithms have relied primarily on batch learning and inference, whereby the model hyperparameters are learned from all of the available data at once. Because the computational complexity associated with GP inference is cubic with respect to the size of the training database [18], several approaches such as Sparse Gaussian Process regression [16], [19] and recursive fusion [13], [20] have been developed to improve its computational efficiency. The recursive fusion approach sequentially fuses new measurements as they are obtained. As such, it is suitable to learn GPs from multi-sensor measurements, because it can assimilate measurements that are obtained and transmitted to the fusion center sequentially and asynchronously. Also, the approach can leverage previous inference results so as to avoid performing redundant computations.

Methods for recursive GP fusion can be divided into data-level fusion and estimate-level fusion [21]. Data-level fusion [13] directly combines multiple sets of sensor measurements, while estimate-level fusion [20], [22] probabilistically fuses multiple prediction estimates that are first obtained by local GPs. Estimate-level fusion approaches, including Mixture of Experts [23] and Product of Experts [22] methods, are successful at distributing computational burden between sensors. However, they combine local sensor estimates in an ad-hoc approach, while data-level fusion builds GP estimators that fuse local measurements systematically, thereby allowing to quantitatively investigate the effects of communication delays, time asynchronicity, and bounded sensor FOV.

This paper contributes to existing data-level fusion methods by developing a recursive fusion approach for DPGP modeling of moving targets from data obtained by multiple and distributed sensor measurements. This is accomplished by the following technical contributions. First, a recursive data-level fusion approach for DPGP is proposed by developing an input dependent DP prior for measurement clustering and combining with a recursive GP method for online data fusion. Second, to quantitatively evaluate the similarity between new and existing measurements, this paper proposes to use the Wasserstein metric for modulating the DP prior. Third, the proposed DPGP fusion approach is applied to multi-target motion model estimation and the effects of the recursive fusion are quantitatively investigated. An analytical expression of information gain is obtained, which can be utilzied for sensor motion planning. Simulation and experimental results show the effectiveness of the proposed approach in learning multiple target motion models. The proposed recursive DPGP provides a computationally efficient way for online sensor fusion, which can be combined with

*This work is supported by ONR BRC grant N00014-17-1-2175.

Laboratory for Intelligent Systems and Controls (LISC), Cornell University, Itahca, NY, 14853.
Email:{cl775,yc2383,jrg362,hy469,ferrari}@cornell.edu

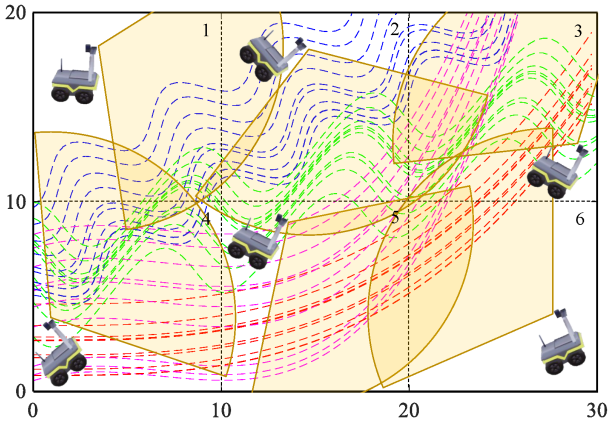


Fig. 1. Illustration of target kinematics learning using multiple mobile sensors. Yellow sectors represent sensor FOV. The dotted curves show the trajectories of four kinematic models, each represented by a different color. The six numbered blocks represent the moving area for each sensor.

the traditional batch DPGP learning by alternating between these two methods to achieve balance between computational efficiency and inference accuracy. To the best of authors' knowledge, this is the first work that proposes a recursive DPGP approach.

The paper is organized as follows. The problem formulation is presented in Section II. Section III provides background knowledge of the DPGP model. The proposed approach is presented in Section IV by describing (a) the recursive fusion rule for GP, (b) the input-dependent DP prior using Wasserstein metric for GPs, (c) recursive DPGP fusion approach, and (d) the expected uncertainty reduction using the recursive DPGP fusion for active sensing. Simulation and hardware results are presented in Section V. Finally, conclusions are drawn in Section VI.

## II. Problem Formulation

This paper considers the problem of learning the kinematics of mobile targets traversing a bounded workspace $\mathcal{W} \subset \mathbb{R}^2$ using multiple mobile robots (Fig. 1). Each robot is equipped with a vision sensor and is confined to a subset of $\mathcal{W}$, for example due to security or energy constraints. As a result, fusing measurements from all sensors in the network is necessary for accomplishing global modeling and estimation of target kinematics in $\mathcal{W}$. It is assumed that each target follows a time-invariant nonlinear ordinary differential equation (ODE),

$$\dot{\mathbf{x}}_j(t) = \mathbf{f}_i\left[\mathbf{x}_j(t)\right] \triangleq \mathbf{v}_j(t), \quad j = 1,\ldots,N(t), \tag{1}$$

where $\mathbf{x}_j(k) \in \mathcal{W}$ and $\mathbf{v}_j(t) \in \mathbb{R}^2$ represent the position and velocity of the $j$th target at time $t$. The total number of targets in the workspace, $N(t)$, is time varying because targets move in and out of the ROI. The vector-valued function $\mathbf{f}_i : \mathbb{R}^2 \to \mathbb{R}^2$ represents the $i$th *velocity field* (VF) that describes the velocity at each position of the workspace, and is assumed unknown *a priori*. It is also assumed that $\mathbf{f}_i$ is drawn from a family of continuously differentiable vector-valued functions, $\mathcal{F} = \{\mathbf{f}_1,\ldots,\mathbf{f}_M\}$, where $M$ is also unknown *a priori*.

Assuming a constant sampling interval $\Delta t$, at any discrete time $k$ a camera obtains both target position and velocity measurements from consecutive video frames obtained from the field of view (FOV) projection onto $\mathcal{W}$, denoted by $\mathcal{S}(k) \subset \mathcal{W}$. The camera obeys a nonlinear measurement model with additive Gaussian noise,

$$\mathbf{z}_j(k) = \begin{cases} \mathbf{h}\left[\mathbf{x}_j(k), \mathbf{v}_j(k)\right] + \mathbf{n}(k) & \text{if } \mathbf{x}_j(k) \in \mathcal{S}(k) \\ \varnothing & \text{if } \mathbf{x}_j(k) \notin \mathcal{S}(k) \end{cases}, \tag{2}$$

where $\mathbf{n} \in \mathbb{R}^4$ is a white, zero-mean Gaussian noise vector with a known and symmetric covariance matrix $\sigma^2\mathbf{I} \in \mathbb{R}^{4\times4}$, where $\mathbf{I}$ is the identity matrix. The nonlinear measurement model, described by the vector function $\mathbf{h}(\cdot) : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^4$, can be obtained from computer vision principles [**?**]. The objective is to collectively estimate $\mathbf{f}_i$ by using measurements from all sensors. Note that the realization of VF can only be observed, i.e., $\mathbf{z}_j(k)$ is nonempty, when the target is inside a sensor FOV, as shown in (2). This results in intermittent sensor measurements and thus a need for a recursive sensor fusion strategy.

## III. Background on Dirichlet Process Gaussian Process

Previous work by the PIs and other authors successfully applied DPGPs to learn the kinematics of an unknown number of moving targets from noisy sensor measurements in batch mode [15], [24]. Since the number of targets, $N(t)$, and the number of VFs, $M(t)$, are both unknown and not necessarily equal, the target-VF associations are modeled by a set of discrete random variables, $c_j \in \{1,\ldots,M\}$, for $j \in \{1,\ldots,N\}$, where the $j$th target is assigned to $c_j$th VF. The distribution of mixture weights, $\boldsymbol{\pi} = [\pi_1,\ldots,\pi_M]^T$, is modeled by the Dirichlet Process (DP).

The main benefit of using DP is its capability to automatically handle the birth of new classes. It has been shown that the assignment of $j$th data, $c_j$, given all other data assignment $\bar{\mathbf{c}}_j = \{c_i | i \neq j\}$ and the concentration parameter $\alpha > 0$, follows the following probability distribution [25], [26]:

$$p(c_j = i | \bar{\mathbf{c}}_j, \alpha) = \frac{\bar{n}_{j,i}}{n-1+\alpha}, \tag{3a}$$

$$p(c_j \neq c_{j'} \forall\, j' \neq j | \bar{\mathbf{c}}_j, \alpha) = \frac{\alpha}{n-1+\alpha}, \tag{3b}$$

where $\bar{n}_{j,i}$ is the number of data that are assigned to $i$th class excluding the $j$th data and $n$ is the total number of data points. These equations show that the conditional probability of assignment to an existing class is proportional to the number of data belonging to that class (3a), and the conditional probability of creating a new class is proportional to $\alpha$ (3b).

A multioutput GP defines a multivariate distribution over functions, $P(\mathbf{f}_i)$, for $\mathbf{f}_i$, $i = 1,\ldots,M$. Let $\{\mathbf{f}_i(\mathbf{x}_1),\ldots,\mathbf{f}_i(\mathbf{x}_n) | \mathbf{x}_l \in \mathcal{W}, j = 1,\ldots,n\}$ be a set of vector function values obtained at $n$ points in $\mathcal{W}$. Define $\boldsymbol{\theta}_i(\mathbf{x}_j) \triangleq \mathbb{E}[\mathbf{f}_i(\mathbf{x}_j)]$ as the GP mean, and $\boldsymbol{\Psi}_i(\mathbf{x}_j, \mathbf{x}'_j) \triangleq \mathbb{E}\left\{[\mathbf{f}_i(\mathbf{x}_j) - \boldsymbol{\theta}_i(\mathbf{x}_j)][\mathbf{f}_i(\mathbf{x}'_j) - \boldsymbol{\theta}_i(\mathbf{x}'_j)]^T\right\}$ as the GP covariance, where $\mathbb{E}[\cdot]$ is the expectation operator. Then, $\boldsymbol{\theta}_i$ and $\boldsymbol{\Psi}_i$ fully

specify the $i$th GP [18]. For simplicity, it is assumed that all GPs share the same known covariance function $\mathbf{\Psi}$.

Under proper assumptions [27], the following DPGP mixture model,

$$\begin{aligned} \{\boldsymbol{\theta}_i, \boldsymbol{\pi}\} &\sim \mathrm{DP}(\alpha, \mathrm{GP}(\mathbf{0}, \mathbf{\Psi})),\ i = 1, \ldots, \infty \\ c_j &\sim \mathrm{Cat}(\boldsymbol{\pi}),\ j \in \{1, \ldots, N\} \\ \hat{\mathbf{f}}_{c_j}(\mathbf{x}) &\sim \mathrm{GP}(\boldsymbol{\theta}_{c_j}, \mathbf{\Psi}),\ \mathbf{x} \in \mathcal{W},\ j \in \{1, \ldots, N\}, \end{aligned} \tag{4}$$

can be used to describe the target kinematics in (1) from the position and velocity measurements (2), where "Cat" denotes the categorical distribution and "$\sim$" denotes "is distributed as". Each VF is modeled as a Gaussian process, and the target-VF association is modeled by $c_j$. The hyperparameters of DPGP can be learned using Monte Carlo approaches [25].

To model the VF, the collocation-point method [28], [**?**] that represents the distribution of velocity in the workspace using the velocities at a set of fixed collocation points is adopted. Specifically, let $\mathbf{\Upsilon} \triangleq \{\boldsymbol{\xi}_i \in \mathcal{W} | i = 1, \ldots, V\}$ be a set of $V$ collocation points distributed on a uniform grid in $\mathcal{W}$. Then every VF approximation is represented by $\hat{\mathbf{f}}_i(\mathbf{\Upsilon}) \triangleq \left[\hat{\mathbf{f}}_i(\boldsymbol{\xi}_1)^T, \ldots, \hat{\mathbf{f}}_i(\boldsymbol{\xi}_V)^T\right]^T$. All $M$ VFs' approximation learned by DPGP is represented by $\hat{\mathbf{f}}(\mathbf{\Upsilon}) \triangleq \left[\hat{\mathbf{f}}_1(\mathbf{\Upsilon})^T, \ldots, \hat{\mathbf{f}}_M(\mathbf{\Upsilon})^T\right]^T$.

Let $\mathcal{V}_{\mathbf{k}} \triangleq \{\mathbf{v}(k) | k \in \mathbf{k}\}$ represent a set of target velocities measured at time steps $\mathbf{k}$ and $\mathcal{X}_{\mathbf{k}} \triangleq \{\mathbf{x}(k) | k \in \mathbf{k}\}$ denote the corresponding positions. Also define $\mathcal{V}_{i,\mathbf{k}} \subset \mathcal{V}_{\mathbf{k}}$ and $\mathcal{X}_{i,\mathbf{k}} \subset \mathcal{X}_{\mathbf{k}}$ as the set of velocities and positions associated with $i$th VF. Then the inference of $i$th VF, $\hat{\mathbf{f}}_i(\mathbf{\Upsilon})$, follows a Gaussian Model [18], [29]:

$$\hat{\mathbf{f}}_i(\mathbf{\Upsilon}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_i, \hat{\mathbf{\Sigma}}_i), \text{ where}$$

$$\hat{\boldsymbol{\mu}}_i = K(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}}) \mathbf{L}_i \mathcal{V}_{i,\mathbf{k}}, \tag{5a}$$

$$\hat{\mathbf{\Sigma}}_i = K(\mathbf{\Upsilon}, \mathbf{\Upsilon}) - K(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}}) \mathbf{L}_i K(\mathcal{X}_{i,\mathbf{k}}, \mathbf{\Upsilon}), \tag{5b}$$

$$\mathbf{L}_i = \left[K(\mathcal{X}_{i,\mathbf{k}}, \mathcal{X}_{i,\mathbf{k}}) + \sigma^2 I\right]^{-1},$$

where $K(\cdot, \cdot)$ is the kernel for GP.

## IV. Recursive Sensor Fusion for DPGP

The recursive DPGP sensor fusion approach presented in this paper consists of two key steps: sequential assignment of target trajectories to VF classes, followed by recursive fusion of trajectories in GP models. Classic recursive GP fusion for multi-target modeling is first reviewed in Section IV-A for completeness. Then, a new recursive DPGP fusion approach, using a Wasserstein metric-based DP prior for sequential assignment of trajectories, is presented in Section IV-B. Finally, the use of recursive DPGP fusion in uncertainty reduction prediction for active sensing applications is presented in Section IV-C.

### A. Recursive GP Fusion

Consider the GP inference where two sets of training data for $i$th VF, $(\mathcal{X}_{i,\mathbf{k}_1}, \mathcal{V}_{i,\mathbf{k}_1})$ and $(\mathcal{X}_{i,\mathbf{k}_2}, \mathcal{V}_{i,\mathbf{k}_2})$, are sequentially provided. This is a common situation in sensor fusion, especially when measurements are intermittently obtained. Different from (5) that computes $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,\mathbf{k}_1}, \mathcal{V}_{i,\mathbf{k}_2})$ in a batch manner, where $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V})$ means the estimated VF based on measurements $\mathcal{V}$, a recursive GP fusion first computes $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,\mathbf{k}_1})$ and then derives $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,\mathbf{k}_1}, \mathcal{V}_{i,\mathbf{k}_2})$ from $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,\mathbf{k}_1})$.

Specifically, $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,\mathbf{k}_1}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{i,1}, \hat{\mathbf{\Sigma}}_{i,1})$ by directly using (5), where

$$\hat{\boldsymbol{\mu}}_{i,1} = K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_1}\right)\left[K\left(\mathcal{X}_{i,\mathbf{k}_1}, \mathcal{X}_{i,\mathbf{k}_1}\right) + \sigma^2 \mathbf{I}\right]^{-1} \mathcal{V}_{i,\mathbf{k}_1},$$

$$\hat{\mathbf{\Sigma}}_{i,1} = K(\mathbf{\Upsilon}, \mathbf{\Upsilon}) - K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_1}\right)\left[K\left(\mathcal{X}_{i,\mathbf{k}_1}, \mathcal{X}_{i,\mathbf{k}_1}\right) + \sigma^2 \mathbf{I}\right]^{-1} K\left(\mathcal{X}_{i,\mathbf{k}_1}, \mathbf{\Upsilon}\right).$$

Continuing to fuse $(\mathcal{X}_{i,\mathbf{k}_2}, \mathcal{V}_{i,\mathbf{k}_2})$ results in the new probability distribution $\hat{\mathbf{f}}_i(\mathbf{\Upsilon} | \mathcal{V}_{i,1}, \mathcal{V}_{i,2}) \sim \mathcal{N}(\hat{\boldsymbol{\mu}}_{i,2}, \hat{\mathbf{\Sigma}}_{i,2})$, where

$$\hat{\boldsymbol{\mu}}_{i,2} = \hat{\boldsymbol{\mu}}_{i,1} + \left[K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_1}\right), K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_2}\right)\right] \mathbf{L}_i \begin{bmatrix} \mathcal{V}_{i,\mathbf{k}_1} \\ \mathcal{V}_{i,\mathbf{k}_2} \end{bmatrix} \tag{6a}$$

$$\hat{\mathbf{\Sigma}}_{i,2} = \hat{\mathbf{\Sigma}}_{i,1} - \left[K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_1}\right), K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_2}\right)\right] \mathbf{L}_i \begin{bmatrix} K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_1}\right) \\ K\left(\mathbf{\Upsilon}, \mathcal{X}_{i,\mathbf{k}_2}\right) \end{bmatrix} \tag{6b}$$

where,

$$\mathbf{L}_i \triangleq \begin{bmatrix} K_{11}^{-1} K_{12} \tilde{K}^{-1} K_{21} K_{11}^{-1} & -K_{11}^{-1} K_{12} \tilde{K}^{-1} \\ -\tilde{K}^{-1} K_{21} K_{11}^{-1} & \tilde{K}^{-1} \end{bmatrix}$$

$$\tilde{K} \triangleq K_{22} - K_{21} K_{11}^{-1} K_{12}, \quad K_{11} \triangleq K\left(\mathcal{X}_{i,\mathbf{k}_1}, \mathcal{X}_{i,\mathbf{k}_1}\right) + \sigma^2 \mathbf{I},$$

$$K_{12} \triangleq K_{21}^T = K\left(\mathcal{X}_{i,\mathbf{k}_1}, \mathcal{X}_{i,\mathbf{k}_2}\right), \quad K_{22} \triangleq K\left(\mathcal{X}_{i,\mathbf{k}_2}, \mathcal{X}_{i,\mathbf{k}_2}\right) + \sigma^2 \mathbf{I}.$$

Equations (6) were presented in [13]. However, no derivation was provided. This paper provides the derivation in the Appendix.

The recursive formulation (6) allows the online fusion of GP process without incurring large computational overhead. To see this, let the dimension of $\mathcal{X}_1$ and $\mathcal{X}_2$ be $n_1 \times d$ and $n_2 \times d$ respectively, where $n_1, n_2$ represents the number of data points in $\mathcal{X}_1$ and $\mathcal{X}_2$ and $d$ is the dimension of a data point. Then when fusing $\mathcal{X}_2$, the batch approach (5) will incur computation complexity $O(n_1 + n_2)^3$ since $[\mathcal{X}_1, \mathcal{X}_2]$ are fused as an ensemble. On the contrary, the recursive update will incur $O\left(n_1^2 n_2\right)$ if $K_{11}^{-1}$ is computed before $\mathcal{X}_2$ is received (this is usually possible since $\mathcal{X}_1$ and $\mathcal{X}_2$ are sequentially obtained). Since in the recursive fusion, $n_1$ is the size of previously received data and $n_2$ is that of the newly obtained data, $n_1 \gg n_2$ in general. Therefore the recursive fusion approach (6) can significantly improve the online computational efficiency.

### B. Recursive DPGP fusion

The conditional probability of target assignment is used in determining the target-VF association. To be specific, let $(\mathcal{X}_{c_j}, \mathcal{V}_{c_j})$ be the set of measured positions and velocities of $j$th target. The conditional probability of target-VF association is represented as follows,

$$p(c_j = i | \bar{\mathbf{c}}_j, \mathcal{X}_{c_j}, \mathcal{V}_{c_j}) \propto p(c_j = i | \bar{\mathbf{c}}_j) p(\mathcal{V}_{c_j} | \mathcal{X}_{c_j}, c_j = i), \tag{7}$$
$$i = 1, \ldots, M,\ j = 1, \ldots, N,$$

where $p(c_j = i | \bar{\mathbf{c}}_j)$ is given by the DP prior (3) and $p(\mathcal{V}_{c_j} | \mathcal{X}_{c_j}, c_j = i)$ is the conditional probability that can be computed by using the GP model $\hat{\mathbf{f}}_i(\mathbf{\Upsilon})$ in (5). The value of $c_j$ is then decided according to $p(c_j = i | \bar{\mathbf{c}}_j, \mathcal{X}_{c_j}, \mathcal{V}_{c_j})$. Once the class assignment is determined, the new trajectory can be fused into the assigned VF class using the sequential fusion rule (6).

The traditional prior of the Dirichlet Process (3) is input independent. This, however, loses useful information on similarity between trajectories, which is critical for the assignment of trajectories to VF classes. Therefore, inspired by [29], this paper uses an input dependent estimate for $\bar{n}_{i,j}$. Specifically, define

$$\bar{n}_{j,i} = (n-1)\frac{\sum_{j'\neq j}\bar{K}_\beta(\hat{\mathbf{f}}_j(\Upsilon),\hat{\mathbf{f}}_{j'}(\Upsilon))\delta(c_{j'},i)}{\sum_{j'\neq j}\bar{K}_\beta(\hat{\mathbf{f}}_j(\Upsilon),\hat{\mathbf{f}}_{j'}(\Upsilon))}, \tag{8}$$

where $\delta(c_{j'},i)$ is the delta function that takes value one if and only if $c_{j'}=i$. Kernel function $\bar{K}_\beta$ encodes the relation between trajectories. This paper uses the Wasserstein metric [30] for measuring the similarity between two GPs. The Wasserstein metric on probability measures derives from the optimal transport theory and is used to describe the optimal cost of transporting a unit of mass from one probability measure to another [30]. The Wasserstein metric for two GPs can be represented as follows [31],

$$W_2^2\left(\hat{\mathbf{f}}_1(\Upsilon),\hat{\mathbf{f}}_2(\Upsilon)\right) = l^2(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2) + Tr\left(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2-2\left(\boldsymbol{\Sigma}_1^{\frac{1}{2}}\boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_1^{\frac{1}{2}}\right)^{\frac{1}{2}}\right), \tag{9}$$

where $Tr$ is the trace operator for matrices and $\mu_i$ and $\Sigma_i$ are mean and covariance matrix of $\hat{\mathbf{f}}_i$ $(i=1,2)$. The $l$ denotes the metric between two vectors, and this paper uses the Frobenius norm. The kernel function $\bar{K}_\beta(x_i,x_{i'})$ is then defined as

$$\bar{K}_\beta\left(\hat{\mathbf{f}}_1(\Upsilon),\hat{\mathbf{f}}_2(\Upsilon)\right) = e^{-\beta W_2^2\left(\hat{\mathbf{f}}_1(\Upsilon),\hat{\mathbf{f}}_2(\Upsilon)\right)}, \tag{10}$$

where $\beta$ is a tuning parameter. The computational complexity of computing $p(c_j=i|\bar{\mathbf{c}}_j)$ is $O(MV^3)$, where $V$ is the number of collocation points and $M$ is the number of existing VFs. This is due to executing matrix square root and multiplication operations, of complexity $O(V^3)$, in (9) for $M(t)$ times in (8).

### C. Expected Uncertainty Reduction

The recursive DPGP fusion approach discussed so far has been focused on sequentially fusing measurements from multiple sensors. It can be easily applied to multi-target modeling in active sensing settings, where sensors strategically select their positions to make informative measurements. The key for active sensing is choosing positions that will maximally reduce estimation uncertainty. The KL divergence is used to evaluate the effects of measurements in reducing the uncertainty of VF estimation. By utilizing the recursive GP in (6), this part presents, for the first time in literature, an analytic expression of new measurement's uncertainty reduction.

Specifically, define the information gain of velocity and position measurements, $\mathcal{V}^*$ and $\mathcal{X}^*$, as

$$D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}^*\right) = D_{\mathrm{KL}}\left(P(\hat{\mathbf{f}}(\Upsilon)|\mathcal{V}^*)\|P(\hat{\mathbf{f}}(\Upsilon))\right), \tag{11}$$

where $D_{\mathrm{KL}}(r\|q) = -\int r(\xi)\log\frac{r(\xi)}{s(\xi)}d\xi$ denotes the KL divergence between two probability measures $r$ and $q$. Then the expected KL divergence for expected measurement $\mathcal{V}^*$ can be represented as

$$D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}^*\right) = \mathbb{E}_{\mathcal{V}^*}\mathbb{E}_{c^*}D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}^*\right), \tag{12}$$

where the expectations are taken over the stochastic measurement outcomes $\mathcal{V}^*$ and the association between the measurement and VF classes $c^*$. According to [**?**], $D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}^*\right)$ can be decomposed into the sum of mutual informations from each VF, i.e.,

$$D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}^*\right) = \sum_{i=1}^{M}\pi_i I\left(\hat{\mathbf{f}}_i(\Upsilon);\mathcal{V}^*\right), \tag{13}$$

where $I\left(\hat{\mathbf{f}}_i(\Upsilon);\mathcal{V}^*\right)$ represents the mutual information of obtaining measurements $\mathcal{V}^*$. Define $\hat{\boldsymbol{\Sigma}}_i^*$ as the GP covariance of $i$th VF after fusing $(\mathcal{X}^*,\mathcal{V}^*)$. By utilizing the recursive GP fusion in (6), it can be derived that

$$\begin{aligned} I\left(\hat{\mathbf{f}}_i(\Upsilon);\mathcal{V}^*\right) &= H\left(\hat{\mathbf{f}}_i(\Upsilon)\right) - H\left(\hat{\mathbf{f}}_i(\Upsilon)|\mathcal{V}^*\right) = \frac{1}{2}\log\frac{\left|\hat{\boldsymbol{\Sigma}}_i\right|}{\left|\hat{\boldsymbol{\Sigma}}_i^*\right|} \\ &= \frac{1}{2}\log\frac{\left|\hat{\boldsymbol{\Sigma}}_i\right|}{\left|\hat{\boldsymbol{\Sigma}}_i - [K(\Upsilon,\mathcal{X}_i),K(\Upsilon,\mathcal{X}^*)]\mathbf{L}_i\begin{bmatrix}K(\mathcal{X}_i,\Upsilon)\\K(\mathcal{X}^*,\Upsilon)\end{bmatrix}\right|} \\ &= -\frac{1}{2}\log\left|I - \hat{\boldsymbol{\Sigma}}_i^{-1}[K(\Upsilon,\mathcal{X}_i),K(\Upsilon,\mathcal{X}^*)]\mathbf{L}_i\begin{bmatrix}K(\mathcal{X}_i,\Upsilon)\\K(\mathcal{X}^*,\Upsilon)\end{bmatrix}\right|, \end{aligned} \tag{14}$$

where $H$ represents the information entropy. The first equality comes from the relationship between mutual information and entropy, and the second equality follows from the entropy of multivariate Gaussian distributions. By plugging in (6b) and using the property that $\det AB = \det A\det B$ if both $A$ and $B$ are square matrices, the last equality is obtained.

Using (13) and (14) quantifies the expected uncertainty reduction for measurement $\mathcal{V}^*$. This closed-form information gain function can be used in conjunction with optimal control approaches for deciding the most informative sensor positions. The interested readers can refer to [6], [32], [33] for more details.

**Remark** It should be noted that the proposed recursive DPGP approach is not to replace the batch approach of DPGP learning. Instead, it provides a computationally efficient way for online sensor fusion. In fact, the recursive fusion approach can be combined with the batch DPGP learning by alternating between these two methods: between two consecutive batch learning, multiple rounds of recursive fusion can be conducted to efficiently fuse newly obtained sensor measurements. The batch learning can then be conducted to optimize hyperparameters and refine clustering of data. The combination of recursive and batch DPGP fusion provides a desirable balance between computational efficiency and inference accuracy.

## V. Simulations and Experiments

### A. Numerical Simulation

In this section, the proposed recursive DPGP fusion approach is verified for a variety of target kinematics. The simulations are performed in the Unreal Engine® (Fig. 2), which is a commercial physical engine that provides high-fidelity images and allows computer vision algorithms to be tested on. Six moving cameras are used to collectively monitor a workspace with size $30m\times 20m$. Each camera moves within a subset of the workspace and the cameras'

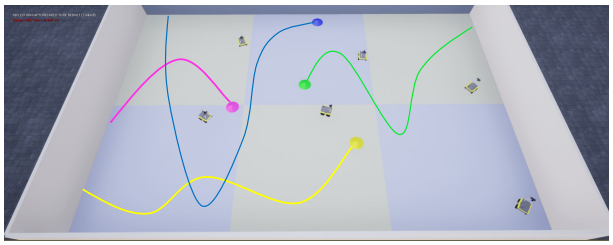

Fig. 2. The simulated workspace in UnrealEngine is of size $30m \times 20m$. Multiple moving targets traverse the workspace and their positions and velocities are measured by six mobile sensors. The example trajectories of moving targets are showing as colored curves.

moving areas are disjoint. All cameras have the same sensing range of $10m$ with horizontal angle of view being $110°$. Image frames are obtained with around $3.3Hz$ by each camera. Targets of four different kinematics continuously traverse the workspace, as shown in Fig. 2. The radial basis function kernel [18] is used for GP inference (5). The effectiveness of the recursive DPGP fusion is evaluated by comparing with three benchmark methods, including no fusion, random fusion, and ground-truth association. The first method does not update the DPGP model, which is the baseline approach. The second is a random association algorithm that randomly assigns target trajectories to VFs. The third uses the ground-truth knowledge of target-VF association for assigning trajectories. All fusion happens after a target moves outside of the workspace. Note that no hyperparameter learning is conducted in this process. Learning hyperparameters will be conducted in the phase of batch fusion.

All four fusion approaches are evaluated in two tasks – target motion prediction and expected uncertainty reduction – to compare the performance of the target kinematics learned by each fusion method. In the first task, the predicted target positions in $\mathcal{W}$ at each time step are compared with true positions. A prediction horizon $h = 6$ steps (i.e., $2s$) is used and the prediction is evaluated by using the root-mean-square error (RMSE) as follows. Let $K_j$ denote the number of measurements obtained from the $j$th target. Also define the position of $j$th target estimated from a DPGP model, $\hat{\mathbf{x}}_j(k)$, as the weighted sum of predicted positions from all VF classes. Then the RMSE is defined as

$$\epsilon_{\text{pos}} = \frac{1}{N_T}\sum_{j=1}^{N_T}\sqrt{\frac{1}{K_j}\sum_{k=1}^{K_j}\sum_{t=k+1}^{k+h}\|\mathbf{x}_j(t|k) - \hat{\mathbf{x}}_j(t|k)\|^2}, \tag{15}$$

where $N_T$ is the total number of targets appearing in the workspace, $\hat{\mathbf{x}}_j(t|k)$ is the predicted target position for time $t$ from time $k$.

In the second task, the expected uncertainty reduction, calculated using (13) and (14), are computed along the target trajectories. The differences of expected uncertainty reduction between the ground-truth association and recursive DPGP fusion, random fusion, and no-fusion approaches, respectively are computed and compared. Specifically, the

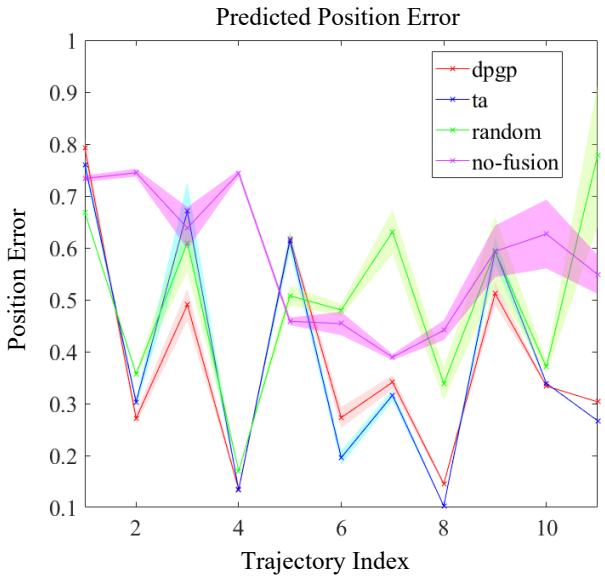


Fig. 3. The mean and standard deviation of the RMSE of predicted position error over ten simulation sets are shown. The x-axis shows the number of trajectories that have been measured by sensors in each simulation. Lines show the mean error over ten sets and the shaded areas show the corresponding standard deviation.

RMSE of differences are computed, defined as

$$\epsilon_{\text{unc}} = \frac{1}{N_T}\sum_{j=1}^{N_T}\sqrt{\frac{1}{K_j}\sum_{k=1}^{K_j}\|D\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}_{\mathbf{k}}^*\right) - D_{\text{gt}}\left(\hat{\mathbf{f}}(\Upsilon);\mathcal{V}_{\mathbf{k}}^*\right)\|^2}, \tag{16}$$

where $D_{\text{gt}}$ is the expected uncertainty reduction for the ground-truth association. $\mathcal{V}_{\mathbf{k}}^*$ denotes the set of predicted measurement in the prediction horizon $h$.

Ten sets of simulations are conducted, with each set containing eleven randomly generated trajectories drawn from four kinematic models. To demonstrate the effects of fusion on improving prediction accuracy, the RMSE in predicted target position and expected uncertainty reduction over the ten simulation sets are summarized in Figs. 3 and 4 to show the trend of RMSE as more trajectories are observed. It can be observed that (1) fusing newly observed trajectories help reduce prediction errors as the no-fusion approaches results in largest prediction errors; (2) The proposed DPGP fusion approach effectively learns kinematics models and achieves close performance as that using the ground-truth association, while significantly exceeding the performances of both no-fusion and random fusion methods. It is worth noting that, the ground-truth association gives nonzero prediction error. This is because only the assignment of trajectories to VF classes uses the ground-true value. However, there is still stochasticity in position prediction. It can also be noticed that sometimes DPGP gives better performance than ground-truth association. This happens because the hyperparameters of DPGP is not trained online in the proposed approach.

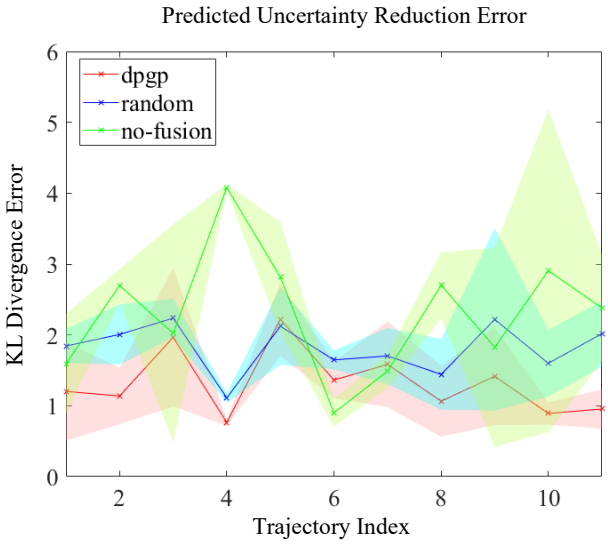


Fig. 4. The mean and standard deviation of the RMSE of prediction error over ten simulation sets are shown. The errors are computed by comparing with the prediction value of the ground-truth association approach.

### B. Hardware Experiment

The recursive DPGP fusion approach is also evaluated in a physical experiment, the setup of which is shown in Fig. 5. Two ZED® stereo cameras are used for measuring the position of flying targets (Crazyflie® drones) in a $4.5m \times 1.5m$ field. One camera is mounted on a moving platform (Parrot® ARDrone) and the other one is fixed to the ground. The targets are identified from the RGB camera frames. The target position in the world frame is computed by post-processing camera measurements and the state estimation from Crazyflie® drones at the $10Hz$ rate. The target follows one of three different kinematic models, as shown on the ground in Fig. 5. Seven trajectories with different initial states (target position and orientation) are generated for each model, totaling 21 trajectories. The target position prediction with 10-step prediction horizon (i.e., $1s$) is conducted. One trajectory for each model is used as the initial trajectory set of learning the DPGP model. The remaining 18 trajectories are fused and tested sequentially. An average RMSE of $0.3505m$ prediction error is achieved by using the recursive DPGP fusion, which shows very accurate prediction capability of the learned target models.

## VI. Conclusion

This paper develops a recursive fusion approach for DPGP modeling an unknown number of moving targets from data obtained by multiple and distributed sensor measurements. A recursive DPGP fusion approach is proposed that updates the learned target model by sequentially fusing sensor measurements. To incorporate similarity between GPs into DP clustering, an input-dependent DP prior that depends on Wasserstein metric is developed. A computationally efficient GP fusion method is then utilized to update the DPGP model. The application of the proposed approach in multi-target motion modeling is investigated. Numerical simulations and hardware experiments demonstrated a decreasing trend in DPGP model error using the recursive DPGP fusion, and an advantage over benchmark methods.

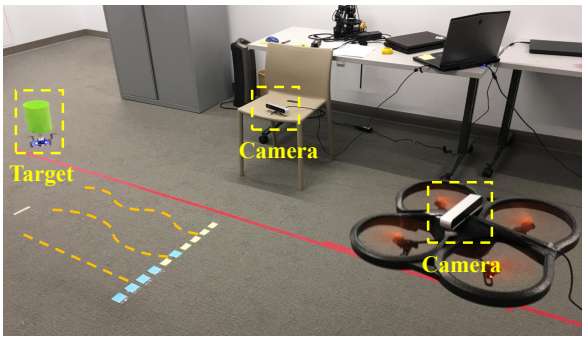


Fig. 5. In the experiment, two stereo cameras are used to measure moving targets. The target follows one of three kinematic models (orange dotted lines and curves) with different starting positions (blue and yellow stickers on the ground) in the workspace.

Future work will deal with certain drawbacks of DPGP, such as slow response to changes in motion [34]. Online hyperparameter updating for the decentralized recursive DPGP fusion also needs investigation. Besides, incorporating the DPGP recursive fusion with optimal sensor control for active sensing using multi-robot networks is another focus of future work.

## APPENDIX

The recursive GP fusion (6) is proved in this part. Utilizing the conditional Gaussian property (5), the conditional mean of $\Upsilon$ given $\mathcal{X}_1$ and $\mathcal{X}_2$ is

$$\hat{\mu}_2 = K(\Upsilon,[X_1,X_2])\,K([X_1,X_2],[X_1,X_2])^{-1}\begin{bmatrix}\mathcal{V}_1\\ \mathcal{V}_2\end{bmatrix},$$

where $K([X_1,X_2],[X_1,X_2]) = \begin{bmatrix}K_{11} & K_{12}\\ K_{21} & K_{22}\end{bmatrix}$, with $K_{ij}, i,j \in \{1,2\}$ defined in (6). Using the Schur complement and Woodbury matrix identity [35] gives

$$K([X_1,X_2],[X_1,X_2])^{-1} = \begin{bmatrix}K_{11}^{-1}+K_{11}^{-1}K_{12}\tilde{K}^{-1}K_{21}K_{11}^{-1} & -K_{11}^{-1}K_{12}\tilde{K}^{-1}\\ -\tilde{K}^{-1}K_{21}K_{11}^{-1} & \tilde{K}^{-1}\end{bmatrix}.$$

Therefore $\hat{\mu}_2$ can be expanded as follows:

$$\begin{aligned}\hat{\mu}_2 &= K(\Upsilon,X_1)K_{11}^{-1}Z_1 + \\ &\quad [K(\Upsilon,X_1),K(\Upsilon,X_2)]\cdot \\ &\quad \begin{bmatrix}K_{11}^{-1}K_{12}\tilde{K}^{-1}K_{21}K_{11}^{-1} & -K_{11}^{-1}K_{12}\tilde{K}^{-1}\\ -\tilde{K}^{-1}K_{21}K_{11}^{-1} & \tilde{K}^{-1}\end{bmatrix}\begin{bmatrix}\mathcal{V}_1\\ \mathcal{V}_2\end{bmatrix} \\ &= \hat{\mu}_1 + [K(\Upsilon,X_1),K(\Upsilon,X_2)]\mathbf{L}\begin{bmatrix}\mathcal{V}_1\\ \mathcal{V}_2\end{bmatrix}.\end{aligned}$$

This proves (6a). Equation (6b) can be similarly proved.

## References

[1] M. Schwager, B. J. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward-facing cameras." Institute of Electrical and Electronics Engineers, 2009.

[2] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser, "Modeling and decision making in spatio-temporal processes for environmental surveillance," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5490–5497.

[3] J. R. Gemerek, S. Ferrari, and J. D. Albertson, "Fugitive gas emission rate estimation using multiple heterogeneous mobile sensors," in *Olfaction and Electronic Nose (ISOEN), 2017 ISOCS/IEEE International Symposium on*. IEEE, 2017, pp. 1–3.

[4] H. Wei and S. Ferrari, "A geometric transversals approach to sensor motion planning for tracking maneuvering targets," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2773–2778, 2015.

[5] C. Liu and J. K. Hedrick, "Model predictive control-based target search and tracking using autonomous mobile robot with limited sensing domain," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 2937–2942.

[6] G. Foderaro, P. Zhu, H. Wei, T. A. Wettergren, and S. Ferrari, "Distributed optimal control of sensor networks for dynamic target tracking," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 142–153, 2018.

[7] K. Mohta, M. Turpin, A. Kushleyev, D. Mellinger, N. Michael, and V. Kumar, "Quadcloud: a rapid response force with quadrotor teams," in *Experimental Robotics*. Springer, 2016, pp. 577–590.

[8] W. Lu, G. Zhang, S. Ferrari, R. Fierro, and I. Palunko, "An information potential approach for tracking and surveilling multiple moving targets using mobile sensor agents," in *Unmanned Systems Technology XIII*, vol. 8045. International Society for Optics and Photonics, 2011, p. 80450T.

[9] S. MartíNez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[10] S. Yi, Z. He, X. You, and Y.-M. Cheung, "Single object tracking via robust combination of particle filter and sparse representation," *Signal Processing*, vol. 110, pp. 178–187, 2015.

[11] D. Lee, C. Liu, Y.-W. Liao, and J. K. Hedrick, "Parallel interacting multiple model-based human motion prediction for motion planning of companion robots," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 52–61, 2017.

[12] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6136–6143.

[13] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Heteroscedastic gaussian processes for data fusion in large scale terrain modeling," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3452–3459.

[14] K.-C. Ma, L. Liu, and G. S. Sukhatme, "An information-driven and disturbance-aware planning method for long-term ocean monitoring," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 2102–2108.

[15] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A bayesian nonparametric approach to modeling motion patterns," *Autonomous Robots*, vol. 31, no. 4, p. 383, 2011.

[16] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[17] H. Wei, P. Zhu, M. Liu, J. P. How, and S. Ferrari, "Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 159–173, 2019.

[18] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 1 2006.

[19] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[20] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.

[21] K. Tiwari, S. Jeong, and N. Y. Chong, "Point-wise fusion of distributed gaussian process experts (fudge) using a fully decentralized robot team operating in communication-devoid environment," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 820–828, 2018.

[22] M. P. Deisenroth and J. W. Ng, "Distributed gaussian processes," *arXiv preprint arXiv:1502.02843*, 2015.

[23] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[24] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, no. 1, pp. 51–76, 2013.

[25] C. E. Rasmussen, "The infinite gaussian mixture model," in *Advances in neural information processing systems*, 2000, pp. 554–560.

[26] Y. W. Teh, "Dirichlet process," in *Encyclopedia of machine learning*. Springer, 2011, pp. 280–287.

[27] H. Wei, W. Lu, P. Zhu, S. Ferrari, M. Liu, R. H. Klein, S. Omidshafiei, and J. P. How, "Information value in nonparametric dirichlet-process gaussian-process (dpgp) mixture models," *Automatica*, vol. 74, pp. 360–368, 2016.

[28] F. Nobile, R. Tempone, and C. G. Webster, "A sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2309–2345, 2008.

[29] C. E. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," in *Advances in neural information processing systems*, 2002, pp. 881–888.

[30] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[31] A. Mallasto and A. Feragen, "Learning from uncertain curves: The 2-wasserstein metric for gaussian processes," in *Advances in Neural Information Processing Systems*, 2017, pp. 5660–5670.

[32] H. Wei, W. Lu, P. Zhu, G. Huang, J. Leonard, and S. Ferrari, "Optimized visibility motion planning for target tracking and localization," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 76–82.

[33] R. Marchant and F. Ramos, "Bayesian optimisation for informative continuous path planning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6136–6143.

[34] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, "Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 161–177.

[35] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *Siam Review*, vol. 23, no. 1, pp. 53–60, 1981.