

Online Learning and Coverage of Unknown Fields Using Random-Feature Gaussian Processes

Ruijie Du

Ruoyu Lin

Yanning Shen*

Magnus Egerstedt

RUIJIED@UCI.EDU

RLIN10@UCI.EDU

YANNINGS@UCI.EDU

MAGNUS@UCI.EDU

*Department of Electrical Engineering and Computer Science,
University of California, Irvine*

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

This paper proposes a framework for multi-robot systems to perform simultaneous learning and coverage of a domain of interest characterized by an unknown and potentially time-varying density function. To overcome the limitations of Gaussian Process (GP) regression, we employ Random Feature GP (RFGP) and its online variant (O-RFGP) that enables online and incremental inference. By integrating these with Voronoi-based coverage control and Upper Confidence Bound (UCB) sampling strategy, a team of robots can adaptively focus on important regions while refining the learned spatial field for efficient coverage. The incremental update mechanism of O-RFGP naturally supports time-varying environments, allowing efficient adaptation without retaining historical measurement data. Furthermore, to the best of our knowledge, we provide the first theoretical analysis of online learning and coverage through a regret-based formulation, establishing asymptotic no-regret guarantees in the time-invariant setting. The effectiveness of the proposed framework is demonstrated through simulations with both time-invariant and time-varying density functions, along with a physical experiment with a time-varying density function.

Keywords: Multi-robot systems, online learning, coverage control, Gaussian processes

1. Introduction

Environmental exploration and monitoring play a vital role in applications ranging from disaster response to ecological preservation, where multi-robot systems provide significant advantages due to their resilience and effectiveness in data collection and coverage over large domains (Rizk et al., 2019). To guide the deployment of robots, a common approach to characterize the relative importance of the environment is through a density distribution function across the domain of interest, i.e., a spatial field. For example, in wildfire monitoring or fighting (Pham et al., 2017; Rickenbach et al., 2024; Lin et al., 2025), the density function can represent the heat distribution or fuel accumulation, while in sensor network deployment (Pimenta et al., 2010; Schwager et al., 2011; Belal et al., 2023; Xu and Tzoumas, 2024), it can encode the spatial concentration of target-relevant information.

However, in many real-world scenarios, such a density distribution is not known a priori. Hence, the robots must perform environmental exploration to learn the underlying **unknown** density. To this end, recent approaches leverage Bayesian inference to learn such unknown density functions while simultaneously deploying robots to efficiently cover the spatial field (Benevento et al., 2020; Santos et al., 2021; Nakamura et al., 2022; McDonald et al., 2021; Landgren et al., 2021; Xu and

* Corresponding author.

Zhe, 2024). This involves the joint process of exploration and exploitation, where cooperative robots gradually improve their knowledge of the environment through learning the unknown density while simultaneously covering relatively more important or informative areas.

Bayesian inference typically employs the **Gaussian Process (GP)** as a probabilistic surrogate that learns unknown functions with uncertainty quantification (Srinivas et al., 2012; Williams and Rasmussen, 2006; Calandriello et al., 2019; Bogunovic et al., 2016; Berkenkamp et al., 2019). Combined with the Upper Confidence Bound (UCB) sampling technique (Auer, 2000) and Voronoi-based coverage control (Lloyd, 1982; Cortes et al., 2004), the GP-based methods proposed in Benvenuto et al. (2020); Santos et al. (2021); Nakamura et al. (2022) enable simultaneous learning of an unknown density function and coverage over the learned spatial field. These previous works rely on the assumption that the underlying density function is time-invariant, which limits their applicability in dynamic scenarios such as wildfire monitoring, where the density function is time-varying. Pratisoli et al. (2025) empirically adapts the GP-based framework of Nakamura et al. (2022) to the time-varying setting. However, the results in Pratisoli et al. (2025) suggest that effective learning and coverage of time-varying densities remains challenging.

A key limitation of GP-based methods lies in their cubic computational complexity with respect to the number of training data (Williams and Rasmussen, 2006). To tackle this challenge, we leverage the **random feature-based Gaussian process (RFGP)** to alleviate the computational complexity. RFGP is an approximation method that enables scalable inference in kernel-based learning and Gaussian processes (Rahimi and Recht, 2007; Shen et al., 2019; Calandriello et al., 2019; Mutny and Krause, 2018; Wang et al., 2018; Lu et al., 2020). RFGP brings significant improvements in computational cost with acceptable or even negligible performance loss. However, RFGP still requires all historical measurements to be associated with outputs, which limits its applicability in learning time-varying density functions. To this end, we resort to **Online RFGP (O-RFGP)**, which updates incrementally once new training data becomes available, enabling real-time adaptation to changing environments. The lightweight computation and incremental update mechanism make O-RFGP particularly suitable for time-varying function learning. Based on the efficient and online updates obtained via O-RFGP, our proposed framework enables real-time learning and coverage of time-varying spatial fields. The main contributions of this paper are summarized as follows.

c1) We propose a simultaneous learning and coverage framework for exploring and monitoring unknown spatial fields. With its incremental and lightweight updates, the proposed framework is capable of real-time operation and well-suited for physical robotic deployment.

c2) To the best of our knowledge, our framework offers the first analytical results on online learning and coverage through a regret-based analysis in the time-invariant setting. Under mild assumptions, we prove that the proposed algorithm has asymptotic no-regret with respect to the coverage obtained with the corresponding ground truth density.

c3) The performance of the proposed framework is evaluated through simulations with both time-invariant and time-varying densities. Its effectiveness for real-time deployment is further demonstrated in a physical experiment with a time-varying density. The results of both simulations and the physical experiment provide encouraging evidence that the proposed framework can effectively maintain its performance and stability in time-varying settings.

2. Problem Formulation

Consider a group of n robots operating in a convex domain of interest $\mathcal{D} \subset \mathbb{R}^2$. The position of robot $i \in N \triangleq \{1, \dots, n\}$ is denoted by $x^i \in \mathcal{D}$, and the positions of all robots are grouped into a matrix $\mathbf{x} \triangleq [x^1, \dots, x^n]^\top \in \mathbb{R}^{n \times 2}$. The problem of coverage control concerns designing strategies for a group of robots to achieve optimal coverage of \mathcal{D} associated with a density distribution. Specifically, the density distribution function, $f(x) : \mathbb{R}^2 \rightarrow \mathbb{R}_{>0}$, encodes the relative importance of a location $x \in \mathcal{D}$ in the sense that the higher the value of $f(x)$, the more important the location x is. Typically, the subdomain (or region of dominance) that robot i is in charge of is given by the Voronoi cell, $\mathcal{V}^i \triangleq \{q \in \mathcal{D} : \|q - x^i\|_2 \leq \|q - x^j\|_2, \forall j \neq i \in N\}$. When the density function f is known, an optimal coverage can be asymptotically achieved by each robot i following the gradient flow of the locational cost function

$$\ell(\mathbf{x}, f) = \sum_{i=1}^n \int_{x \in \mathcal{V}^i} \|x - x^i\|_2^2 f(x) dx \quad (1)$$

with respect to x^i (Cortes et al. (2004)), resulting in a centroidal Voronoi tessellation (CVT), $\mathbf{c}^i = \frac{\int_{\mathcal{V}^i} x f(x) dx}{\int_{\mathcal{V}^i} f(x) dx}$, $\forall i \in N$, where \mathbf{c}^i is the centroid (i.e., center of mass) of \mathcal{V}^i with respect to the density function f . We denote a CVT by the matrix $\mathbf{c} \triangleq [\mathbf{c}^1, \dots, \mathbf{c}^n]^\top \in \mathbb{R}^{n \times 2}$.

However, in practice, as discussed in Section 1, the density function f can be unknown a priori. Then, the group of robots needs to achieve an optimal coverage of a surrogate density function learned from their measurements. Specifically, at position x^i , robot i takes the measurement $y^i = f(x^i) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. As a robot typically takes independent measurements at each time using the same sensor, the measurement noise ϵ is independent and identically distributed (i.i.d.) across time and space (Benevento et al., 2020).

Let all variables related to time $t \in \{1, \dots, T\}$ be affixed with a subscript $(\cdot)_t$. For example, $\mathbf{x}_t \triangleq [x_t^1, \dots, x_t^n]^\top$ denotes the locations of all the n robots at time t . As each robot takes a measurement y_t^i at time t , all measurements at time t are grouped in a vector $\mathbf{y}_t \triangleq [y_t^1, \dots, y_t^n]^\top \in \mathbb{R}^n$. We use the subscript $(\cdot)_t$ to indicate the variables with the range of time steps from the beginning to t . For example, the positions of all the n robots from $t = 1$ to $t = T$ are grouped into the matrix $\mathbf{x}_{:T} \in \mathbb{R}^{nT \times 2}$ and their corresponding measurements are denoted as the vector $\mathbf{y}_{:T} \in \mathbb{R}^{nT}$. Let $f_t(x)$ denote the time-varying density value at location x at time t , whereas $f(x)$ represents the time-invariant density function at location x whose value remains the same over time.

As the true density function is unknown, robots need to simultaneously learn and update the surrogate density function \hat{f}_t at each time step based on collected training data, i.e., measurements from all robots, and follow a control law to efficiently cover the domain associated with \hat{f}_t . The centroid of \mathcal{V}_t^i with respect to \hat{f}_t becomes $\mathbf{c}_t^i = \frac{\int_{\mathcal{V}_t^i} x \hat{f}_t(x) dx}{\int_{\mathcal{V}_t^i} \hat{f}_t(x) dx}$, and the control law for the team of robots is given by Cortes et al. (2004)

$$\dot{\mathbf{x}}_t = \kappa(\mathbf{c}_t - \mathbf{x}_t), \quad (2)$$

where $\mathbf{c}_t \triangleq [\mathbf{c}_t^1, \dots, \mathbf{c}_t^n]^\top$, and $\kappa > 0$ is a control gain.

3. Density Learning

In this section, we detail the density learning methods. The unknown density function $f(\cdot)$ is modeled as the realization of a GP, specified by a mean function, $\mu(\cdot) : \mathcal{D} \rightarrow \mathbb{R}$, and a covariance (kernel) function, $k(\cdot, \cdot) : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$. In particular, $\sigma^2(x) = k(x, x)$. We assume that the structures of μ and k are characterized by certain hyperparameters ρ^* and τ^* (Srinivas et al., 2009),

$$f(\cdot) \sim \mathcal{GP}(\mu(\cdot; \rho^*), k(\cdot, \cdot; \tau^*)). \quad (3)$$

For time-varying density, (3) transforms into $f_t(\cdot) \sim \mathcal{GP}(\mu(\cdot; \rho_t^*), k(\cdot, \cdot; \tau_t^*))$, where ρ_t^* and τ_t^* are hyperparameters at t .

To learn an unknown function f that has a GP prior, the noisy measurements $\mathbf{y}_{:t}$ at positions $\mathbf{x}_{:t}$ are leveraged. Then, the learned density value \hat{f}_t at location $x \in \mathcal{D}$ at time t can be represented by

$$\hat{f}_t(x) \sim \mathcal{GP}(\mu_t(x), k_t(x, x)), \quad (4)$$

where μ_t and k_t are mean and covariance functions calculated based on $\mathbf{y}_{:t}$ and $\mathbf{x}_{:t}$. In the following content, we will introduce i) the GP-based approach in Section 3.1, ii) the RFGP-based approach in Section 3.2, and iii) the O-RFGP-based approach in Section 3.3, to calculate μ_t and k_t .

Table 1: Computational complexity and memory cost of GP, RFGP, and O-RFGP.

Method	GP	RFGP	O-RFGP
Computational complexity	$\mathcal{O}(T^3)$	$\mathcal{O}(TD^2)$	$\mathcal{O}(D^2)$
Memory cost	$\mathcal{O}(T^2)$	$\mathcal{O}(TD)$	$\mathcal{O}(D^2)$

3.1. Gaussian Process (GP)

Assuming the mean function is a function parameterized by ρ_t and the kernel function parameterized by τ_t , the corresponding μ^{GP} and k^{GP} at time step t are defined as Williams and Rasmussen (2006)

$$\mu_t^{\text{GP}}(x) = \mu(x; \rho_t) + \mathbf{k}_t^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} (\mathbf{y}_{:t-1} - \mu_{:t-1}(\rho_t)), \quad (5)$$

$$k_t^{\text{GP}}(x, x; \tau_t) = k(x, x; \tau_t) - \mathbf{k}_t^\top (\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t, \quad (6)$$

where $\mu_{:t-1}(\rho_t) \triangleq [\mu(\mathbf{x}_1; \rho_t), \dots, \mu(\mathbf{x}_{t-1}; \rho_t)]^\top$, $\mathbf{k}_t \in \mathbb{R}^{nt}$ is the vector with the $((j-1)n+i)$ th entry as $k(x_j^i, x; \rho^t)$, and $\mathbf{K}_t \in \mathbb{R}^{nt \times nt}$ is the covariance matrix with the $((j-1)n+i, (j'-1)n+i')$ th entry as $k(x_j^i, x_{j'}^{i'}; \rho^t)$, $\forall i, i' \in \{1, \dots, n\}$, $\forall j, j' \in \{1, \dots, t\}$. Specifically, at time step t , the measurements $\mathbf{y}_{:t}$ at locations $\mathbf{x}_{:t}$ accumulated by the n robots are used to learn the density function. According to Williams and Rasmussen (2006), the parameters ρ_t and τ_t can be obtained by

$$\rho_t, \tau_t = \arg \min_{\rho, \tau} [(\mu_{:t}(\rho) - \mathbf{y}_{:t})^\top (K_t(\tau) + \sigma^2 \mathbf{I})^{-1} (\mu_{:t}(\rho) - \mathbf{y}_{:t})] + \log |K_t(\tau) + \sigma^2 \mathbf{I}|,$$

where $|\cdot|$ denotes the determinant of a matrix. Note that the choices of the mean and covariance functions are not unique (Williams and Rasmussen, 2006). For example, in the case where the linear model and Radial Basis Function (RBF) kernel are used, we have: $\mu_t(x; \rho_t) \triangleq \rho_t^\top x$, and $k(x, x'; \tau_t) \triangleq \exp\left(-\frac{\|x-x'\|_2^2}{2\tau_t^2}\right)$. Since GP needs the memory of saving the covariance matrix K_t and requires the computation of the inverse of $(\mathbf{K}_t + \sigma^2 \mathbf{I})$, at time T , the memory cost is $\mathcal{O}(T^2)$ and the computational complexity is $\mathcal{O}(T^3)$, as shown in Table 1. As the number of samples increases, updating the mean and kernel functions of the GP becomes computationally intensive, limiting the applicability of GP in real-time robotic applications.

3.2. Random Feature Gaussian Process (RFGP)

Random feature Gaussian Process (RFGP) provides an efficient way to scale GP by approximating the kernel using random features. It is more suitable for applications requiring GPs but constrained by computational resources, such as physical robotics implementations in real-time.

We define the real-valued $2D$ -dimensional random feature (RF) vector as

$$\phi_{\mathbf{v}}(x) \triangleq \frac{1}{\sqrt{D}} [\sin(\mathbf{v}_1^\top x), \cos(\mathbf{v}_1^\top x), \dots, \sin(\mathbf{v}_D^\top x), \cos(\mathbf{v}_D^\top x)]^\top, \quad (7)$$

where D denotes the number of random features and \mathbf{v}_i is a random vector sampled from the spectral distribution of the kernel (Shen et al., 2019). And we define $\phi_{\mathbf{v}}(\mathbf{x}) \triangleq [\phi_{\mathbf{v}}(x^1), \dots, \phi_{\mathbf{v}}(x^n)]$. A low-rank approximation of \mathbf{K}_t can be achieved by using the standardized shift-invariant \bar{k} , whose inverse Fourier transform can be approximated by Lu et al. (2020) $\bar{k}(x, x') \approx \frac{1}{D} \sum_{i=1}^D e^{j\mathbf{v}_i^\top (x-x')} = \phi_{\mathbf{v}}^\top(x) \phi_{\mathbf{v}}(x')$, and $k = \sigma_\theta^2 \bar{k}$, where σ_θ^2 is the magnitude of k . A low-rank ($2D$) approximation of \mathbf{K}^t is shown as $\bar{\mathbf{K}}_t = \sigma_\theta^2 \Phi_t \Phi_t^\top$, where $\Phi_t \triangleq [\phi_{\mathbf{v}}(\mathbf{x}_1), \dots, \phi_{\mathbf{v}}(\mathbf{x}_t)]^\top$. Then, the μ^{RF} and k^{RF} at time step t can be calculated by $\mu_t^{\text{RF}}(x) = \phi_{\mathbf{v}}^\top(x) \left(\Phi_t^\top \Phi_t + \frac{\tau_{\text{RF}} \mathbf{I}_{2D}}{\sigma_\theta^2} \right)^{-1} \Phi_t^\top \mathbf{y}_{:t}$, $k_t^{\text{RF}}(x, x) = \phi_{\mathbf{v}}^\top(x) \left(\frac{\Phi_t^\top \Phi_t}{\tau_{\text{RF}}} + \frac{\mathbf{I}_{2D}}{\sigma_\theta^2} \right)^{-1} \phi_{\mathbf{v}}(x)$. Let $\theta_t = \left(\Phi_t^\top \Phi_t + \frac{\tau_{\text{RF}} \mathbf{I}_{2D}}{\sigma_\theta^2} \right)^{-1} \Phi_t^\top \mathbf{y}_{:t}$, $\Sigma_t = \left(\frac{\Phi_t^\top \Phi_t}{\tau_{\text{RF}}} + \frac{\mathbf{I}_{2D}}{\sigma_\theta^2} \right)^{-1}$, μ_t^{RF} and k_t^{RF} at time step t can be rewritten as

$$\mu_t^{\text{RF}}(x) = \phi_{\mathbf{v}}^\top(x) \theta_t, \quad k_t^{\text{RF}}(x, x) = \phi_{\mathbf{v}}^\top(x) \Sigma_t \phi_{\mathbf{v}}(x). \quad (8)$$

Then, θ_t and Σ_t can be considered as the parameters of RFGP. For example, if $\mu = \rho^\top x$, ideally, $\phi_{\mathbf{v}}^\top(x) \theta^* = \rho^\top x$, where θ^* denotes the optimal model parameters. As such, RFGP hugely reduces computational complexity and memory cost by using a fixed dimension RF vector to approximate the kernel, i.e., at time T , the memory cost is $\mathcal{O}(TD)$ and the computational complexity is $\mathcal{O}(TD^2)$, where D is much smaller than T , as summarized in Table 1.

3.3. Online Random Feature Gaussian Process (O-RFGP)

The RF-based function approximant μ^{RF} and k^{RF} readily accommodates online operation (Lu et al., 2020), which is necessary for many time-critical applications, such as simultaneous learning and covering time-varying density functions. The online approach allows interactive learning, i.e., the prediction at time step t receives \mathbf{x}_t at the beginning of slot t and then updates after receiving measurements at the end of slot t . Specifically, the online learning procedure is shown as follows:

1. Predict mean and covariance at location \mathbf{x}_t as

$$\mu_t^{\text{O}}(\mathbf{x}_t) = \phi_{\mathbf{v}}^\top(\mathbf{x}_t) \theta_{t-1}, \quad \sigma_t^2(\mathbf{x}_t) = k_t^{\text{O}}(\mathbf{x}_t, \mathbf{x}_t) = \phi_{\mathbf{v}}^\top(\mathbf{x}_t) \Sigma_{t-1} \phi_{\mathbf{v}}(\mathbf{x}_t). \quad (9)$$

2. Upon receiving \mathbf{y}_t , update θ_t and Σ_t as

$$\theta_t = \theta_{t-1} + \sigma_t^{-2}(\mathbf{x}_t) \Sigma_t \phi_{\mathbf{v}}(\mathbf{x}_t) (\mathbf{y}_t - \mu_t^{\text{O}}(\mathbf{x}_t)), \quad (10)$$

$$\Sigma_t = \Sigma_{t-1} - \sigma_t^{-2}(\mathbf{x}_t) \Sigma_{t-1} \phi_{\mathbf{v}}(\mathbf{x}_t) \phi_{\mathbf{v}}^\top(\mathbf{x}_t) \Sigma_{t-1}. \quad (11)$$

The O-RFGP can significantly save data storage and further reduce the computational complexity in the sense that at each time step, the computational complexity and the memory cost are both $\mathcal{O}(D^2)$, as summarized in Table 1. This iterative update mechanism enables O-RFGP to adapt to time-varying function learning by relying solely on the current measurements. Meanwhile, the low computational complexity of O-RFGP allows real-time robotic implementations.

Algorithm 1 O-RFGP-Aided Coverage Algorithm (ORC)

-
- 1: **Initialization:**
 - 2: Non-decreasing sequence $\{\beta_t\}_{t=1}^T$.
 - 3: Random locations $\mathbf{x}_0 = [x_0^1, \dots, x_0^n]^\top$.
 - 4: Compute $\mu_0(\cdot)$ and $\sigma_0(\cdot)$.
 - 5: **for** $t = 1 \dots T$ **do**
 - 6: Compute the surrogate \hat{f}_t for all $x \in \mathcal{D}$ by $\hat{f}_t(x) = \mu_{t-1}(x) - \sqrt{\beta_t} \sigma_{t-1}(x)$.
 - 7: Run control law (2) for robots to reach \mathbf{x}_t , i.e., a discrete integration step of (2).
 - 8: Determine the location $\tilde{x}_t = \arg \max_{x \in \mathcal{D}} \sigma_{t-1}(x)$, where to take additional measurement.
 - 9: Obtain the measurements $y_t^i = f_t(x_t^i) + \epsilon$, $\forall i \in \{1, \dots, n\}$, $\tilde{y}_t = f_t(\tilde{x}_t) + \epsilon$.
 - 10: Update $\mu_t(\cdot)$ via (10) and $\sigma_t(\cdot)$ via (11)
 - 11: **end for**
-

4. Learning-Aided Coverage Algorithm

In this section, first, we propose an algorithm that enables robots to simultaneously learn an unknown spatial field and efficiently cover the most recently acquired estimate of it. The algorithm, which integrates GP-based learning and Voronoi-based coverage control, combined with the UCB sampling strategy, guides the robots’ sensing and movement, incrementally driving them toward a CVT with respect to the evolving learned density function. Specifically, this is achieved by alternating between a prediction step, which allows the multi-robot team to select new sampling points, and a correction step, which allows it to update the learned density to be covered. A detailed outline of the proposed algorithm is presented in Algorithm 1. Note that the estimation step is based on O-RFGP but can be directly replaced with GP or RFGP as presented in Section 3. We define the cumulative regret as the sum of the r_t at time step t as the difference between the locational cost with current locations \mathbf{x}_t and the optimal locational cost

$$R_T \triangleq \sum_{t=1}^T r_t, \text{ with } r_t = \ell(f, \mathbf{x}_t) - \min_{\mathbf{x}} \ell(f, \mathbf{x}) \geq 0. \quad (12)$$

A small cumulative regret R_T indicates that the achieved coverage cost $\ell(f, \mathbf{x}_t)$ remains close to the optimal cost, meaning that the robots attain coverage performance comparable to the ideal case with full knowledge of the density function. The theoretical guarantee of sub-linear regret in the time-invariant setting is formally established in Theorem 2, where the density function f , as in (3), does not vary over time. The detailed regret analysis is provided in Appendix A. Extending the analysis to time-varying density functions would naturally require regret bounds that depend on a temporal variation measure of the density, such as a “total variation” (Bogunovic et al., 2016). Since such variation budgets are typically unknown a priori and may grow with time, we leave the regret analysis for the time-varying case as an important direction for future work. The empirical results in Section 5 and Appendix G demonstrate that the cumulative regret remains sub-linear and is influenced by the same key parameters (D and β_t), which suggests that similar principles govern the performance of O-RFGP in the time-varying setting.

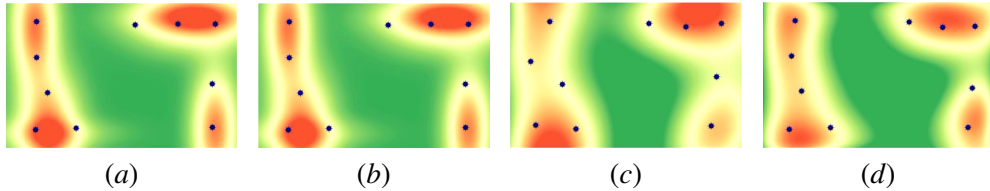


Figure 1: The final configuration (at time step 500) of ten robots covering the time-invariant density distribution that is: (a) the ground truth, (b) learned using GP, (c) learned using RFGP, and (d) learned using O-RFGP. Areas in red (green) indicate higher (lower) importance.

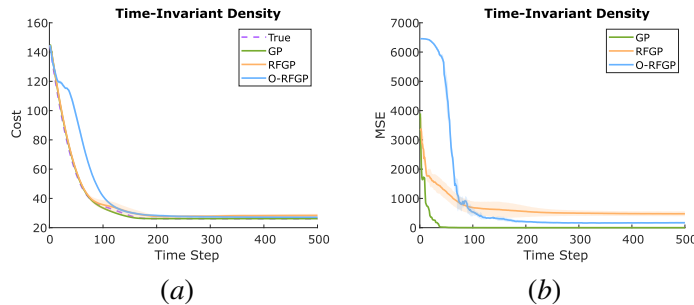


Figure 2: The evolution of the locational cost and the MSE between the learned density and the true density with respect to time steps.

5. Experiments

In this section, we demonstrate the effectiveness of the proposed framework ORC through both simulations and a physical experiment. The simulations are designed to evaluate performance under both time-invariant and time-varying spatial fields, thereby examining the framework’s capability to achieve accurate learning and efficient coverage in static as well as dynamic environments. We further conduct a physical experiment to demonstrate the computational efficiency and effectiveness of O-RFGP for deployment of physical multi-robot systems in real-time.

5.1. Simulations

In this subsection, we conduct simulations to evaluate and compare the performance of the proposed algorithm on time-invariant and time-varying density functions separately. The detailed experimental settings are presented in Appendix F. A summary of how to use each learning method to estimate the (time-invariant or time-varying) density is as follows: **GP**: compute $\mu_t(\cdot)$ and $\sigma_t(\cdot)$ as equations (5) and (6); **RFGP**: compute $\mu_t(\cdot)$ and $\sigma_t(\cdot)$ as equations (8); **O-RFGP**: compute $\mu_t(\cdot)$ and $\sigma_t(\cdot)$ as equations (9) by iteratively updating θ and Σ as equations (10) and (11). We use the mean squared error (MSE) between f_t and \hat{f}_t to evaluate the learning quality, and compute the locational cost using f_t as equation (1) to evaluate the coverage quality. Unless otherwise specified, we average the results over 3 independent runs for all experiments. More experimental results on the sensitivity analysis of hyperparameters D and β_t , and the computational efficiency comparison of GP, RFGP, and O-RFGP are presented in Appendix G.

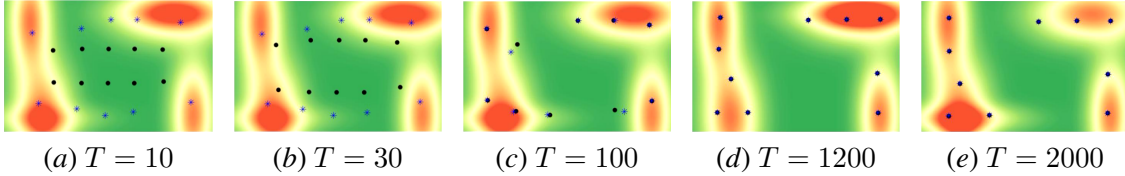


Figure 3: The heatmaps of ten robots covering the true time-varying density distribution at different time steps, where the black dots represent the positions of robots and the blue asterisks represent the centers of mass of their Voronoi cells.

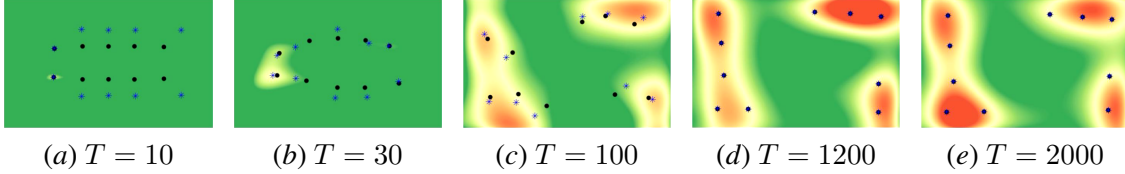


Figure 4: The heatmaps of 1 trial of ten robots covering the learned time-varying density distribution using O-RFGP.

5.1.1. TIME-INVARIANT DENSITY

The measurements collected only at the current time step are leveraged when using O-RFGP, while all measurements collected from the beginning up to the current time step are leveraged when using the GP and RFGP for learning the time-invariant density. The simulation results are shown in Fig.1 and Fig.2. The final configurations of one trail, at time step 500, of the robots covering the learned time-invariant density using the GP, RFGP and O-RFGP are shown in Fig.1(b), Fig.1(c) and Fig.1(d), respectively. Fig.2(a) shows that the coverage qualities achieved by the GP, RFGP, and O-RFGP are almost the same as the coverage quality over the ground truth density. Fig.2(b) and Fig.1 indicate that O-RFGP and RFGP have higher MSE compared with GP. These results are consistent with the regret analysis in Appendix A since RFGP and O-RFGP have larger γ_T . It is worth noting that although all three learning methods achieve similar coverage quality in the end, O-RFGP and RFGP do so with significantly lower computational cost than GP, as seen in Table 1 and Appendix G.2. Unlike the RFGP and O-RFGP, the GP has cubic computational complexity in the number of training samples. This implies the limitation in applying the GP in real-time robotic implementations as discussed in Section 1.

5.1.2. TIME-VARYING DENSITY

We conduct simulations using the measurements that robots collect only at the current time step to avoid stale data for time-varying density learning. The heatmaps of one trial of ten robots covering the learned density using O-RFGP, GP, and RFGP are shown in Fig.4, Fig.5, and Fig.6, respectively. Fig.7 shows that the coverage quality and MSE for density learning resulting from GP and RFGP are apparently higher than those from O-RFGP. In Fig.7(b), the MSE values of RFGP and GP are not only higher than that of O-RFGP but also exhibit high-frequency fluctuations. These fluctuations in MSE lead to higher and less stable coverage costs for GP and RFGP compared to O-RFGP, as shown in Fig.7(a). Compared with Fig.3, the heatmap results in Fig.5 and Fig.6 also indicate that both GP and RFGP do not provide accurate estimation results in the time-varying setting. In contrast, Fig.4 shows that O-RFGP is the only method capable of accurately estimating the true density over

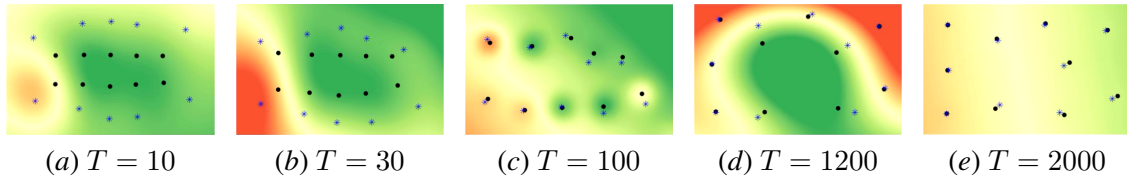


Figure 5: The heatmaps of 1 trial of ten robots covering the learned time-varying density distribution using GP based on the measurements collected at the current time step

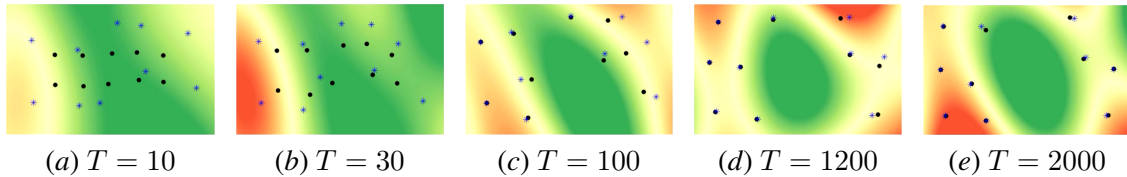


Figure 6: The heatmaps of 1 trial of ten robots covering the learned time-varying density distribution using RFGP based on the measurements collected at the current time step.

time. Since GP and RFGP are designed for learning the time-invariant density using all historical data, as discussed in Section 4, they are not supposed to adapt to the evolution of the time-varying density in a straightforward manner. In contrast, O-RFGP can adapt to the evolution of the time-varying density, achieving almost the same coverage quality as the ground truth density across time, along with accurate estimation. Since O-RFGP iteratively updates the learned density using current measurements as the density varies over time, it is not influenced by out-of-date measurements and able to capture the dynamic property of the time-varying density, as discussed in Section 3.3.

In summary, O-RFGP not only benefits from lightweight computation and memory, but also employs the incremental update structure that only relies on the most recent collected data. These two properties enable real-time learning and adaptation, and the simulation results further suggest the potential of O-RFGP in handling time-varying density for real-world robotic applications.

5.2. Physical Experiment

To highlight the performance and effectiveness of ORC in learning time-varying density on physical robotic systems in real-time, we implement it on ten differential-drive wheeled robots for simultaneous learning and covering an unknown time-varying density. The size of the physical

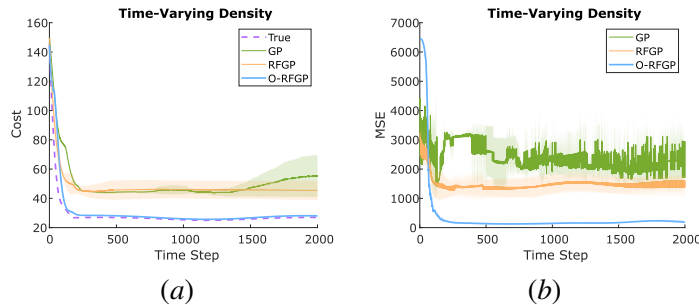


Figure 7: The evolution of the locational cost and the MSE between the learned density and the true density with respect to time step. The measurements collected by robots at the current time step are leveraged when using the GP, RFGP, and O-RFGP.

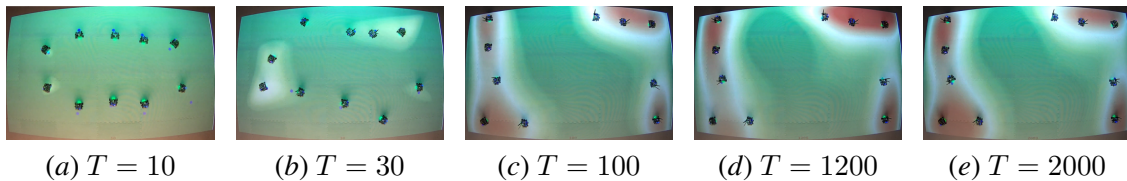


Figure 8: The screenshots of ten differential-drive wheeled robots covering the learned time-varying density distribution using O-RFGP. The blue dots represent the centers of mass of their corresponding Voronoi cells.

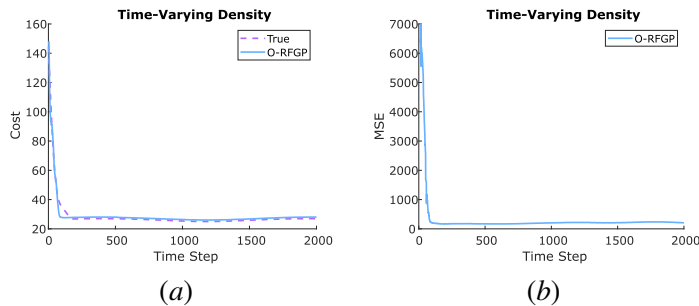


Figure 9: The evolution of the locational cost and the MSE between the learned density in the physical experiment (corresponding to Fig.8) and the true density in the simulation (corresponding to Fig.3). The measurements collected by robots at the current time step are leveraged to learn the time-varying density using O-RFGP.

testbed (Pickem et al., 2017) is the same as the domain of interest \mathcal{D} in numerical simulations. The location of Robot i , x^i , corresponds to the position of a point near the wheel axle of the robot. Single integrator dynamics of x^i is mapped by the near-identity diffeomorphism (Olfati-Saber, 2002) to the unicycle dynamics of the differential-drive wheeled robot. The experimental results are shown in Fig.8 and Fig.9. As shown in Fig.9(a), the locational cost for the time-varying density learned by O-RFGP closely matches that of the ground truth, indicating comparable coverage quality. In Fig.9(b), the MSE between the learned and true densities also converges, confirming the effectiveness of O-RFGP in density learning. Comparing Fig.3, Fig.4, Fig.7, Fig.8, and Fig.9, O-RFGP performs consistently across both simulation and physical experiments, which demonstrate that ORC enables multi-robot systems to efficiently learn and cover unknown, time-varying spatial fields in real time.

6. Conclusion

This paper proposes ORC (Algorithm 1), a framework that integrates O-RFGP and Voronoi-based coverage control to enable a team of robots to learn an unknown density function and achieve efficient spatial coverage. To the best of our knowledge, we provide, for the first time, a regret-based theoretical analysis that guarantees the performance of online learning-aided coverage control in time-invariant settings. The effectiveness of ORC is validated through simulations, and its strong potential in handling time-varying density fields is further demonstrated through additional simulations and a physical experiment, confirming its capability for real-time learning and coverage. Future work includes investigating extensions to the parameter update mechanism of O-RFGP, enabling effective learning of unknown densities with more complex temporal evolution. It also involves extending the regret-based analysis to time-varying settings, and generalizing the proposed framework to heterogeneous sensing modalities and noise characteristics across robots.

Acknowledgments

Work in the paper is supported by NSF ECCS 2207457, ECCS 2412484, and ECCS 2442964.

References

- Peter Auer. Using upper confidence bounds for online learning. In *Proceedings 41st annual symposium on foundations of computer science*, pages 270–279. IEEE, 2000.
- Mehdi Belal, Dario Albani, and Lorenzo Sabattini. Understanding the role of time-varying targets in adaptive distributed area coverage control. In *International Symposium on Experimental Robotics*, pages 239–249. Springer, 2023.
- Alessia Benevento, María Santos, Giuseppe Notarstefano, Kamran Paynabar, Matthieu Bloch, and Magnus Egerstedt. Multi-robot coordination for estimation and coverage of unknown spatial fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7740–7746. IEEE, 2020.
- Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24, 2019.
- Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying gaussian process bandit optimization. In *Artificial Intelligence and Statistics*, pages 314–323. PMLR, 2016.
- Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory*, pages 533–557. PMLR, 2019.
- Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on robotics and Automation*, 20(2):243–255, 2004.
- Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. Distributed cooperative decision making in multi-agent multi-armed bandits. *Automatica*, 125:109445, 2021.
- Ruoyu Lin, Soobum Kim, and Magnus Egerstedt. Heterogeneous collaborative pursuit via coverage control driven by Fokker-Planck equations. *IEEE Transactions on Robotics*, 2025.
- S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- Qin Lu, Georgios Karanikolas, Yanning Shen, and Georgios B Giannakis. Ensemble gaussian processes with spectral features for online interactive learning with scalability. In *International Conference on Artificial Intelligence and Statistics*, pages 1910–1920. PMLR, 2020.
- Wenhao Luo, Changjoo Nam, George Kantor, and Katia Sycara. Distributed environmental modeling and adaptive sampling for multi-robot sensor coverage. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1488–1496, 2019.
- Andrew McDonald, Lai Wei, and Vaibhav Srivastava. Online estimation and coverage control with heterogeneous sensing information. In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 144–149. IEEE, 2021.

- Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. *Advances in Neural Information Processing Systems*, 31, 2018.
- Kensuke Nakamura, María Santos, and Naomi Ehrich Leonard. Decentralized learning with limited communications for multi-robot coverage of unknown spatial fields. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9980–9986. IEEE, 2022.
- R Olfati-Saber. Near-identity diffeomorphisms and exponential tracking and e-stabilization of first-order nonholonomic se (2) vehicles. *Proceeding of the 2002 ACC*, 2002.
- Huy X Pham, Hung M La, David Feil-Seifer, and Matthew Deans. A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6648–6653. IEEE, 2017.
- Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt. The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1699–1706. IEEE, 2017.
- Luciano CA Pimenta, Mac Schwager, Quentin Lindsey, Vijay Kumar, Daniela Rus, Renato C Mesquita, and Guilherme AS Pereira. Simultaneous coverage and tracking (scat) of moving targets with robot networks. In *Algorithmic Foundation of Robotics VIII: Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics*, pages 85–99. Springer, 2010.
- Federico Pratisoli, Mattia Mantovani, Amanda Prorok, and Lorenzo Sabattini. Distributed coverage control for time-varying spatial processes. *IEEE Transactions on Robotics*, 2025.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.
- Rahel Rickenbach, Johannes Köhler, Anna Scampicchio, Melanie N Zeilinger, and Andrea Caron. Active learning-based model predictive coverage control. *IEEE Transactions on Automatic Control*, 2024.
- Yara Rizk, Mariette Awad, and Edward W Tunstel. Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–31, 2019.
- María Santos, Udari Madhushani, Alessia Benevento, and Naomi Ehrich Leonard. Multi-robot learning and coverage of unknown spatial fields. In *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 137–145. IEEE, 2021.
- Mac Schwager, Brian J Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.
- Yanning Shen, Tianyi Chen, and Georgios B Giannakis. Random feature-based online multi-kernel learning in environments with unknown dynamics. *Journal of Machine Learning Research*, 20(22):1–36, 2019.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE transactions on information theory*, 58(5):3250–3265, 2012.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Zhitong Xu and Shandian Zhe. Standard gaussian process is all you need for high-dimensional bayesian optimization. *arXiv e-prints*, pages arXiv–2402, 2024.

Zirui Xu and Vasileios Tzoumas. Performance-aware self-configurable multi-agent networks: A distributed submodular approach for simultaneous coordination and network design. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 5393–5400. IEEE, 2024.

Appendix A. Regret Analysis

In this section, we provide the regret analysis in the time-invariant setting, i.e. the density function f , as in (3), does not vary with respect to time. Remind that the regret, r_t , and the cumulative regret, R_T , are defined in equation (12):

$$r_t = \ell(f, \mathbf{x}_t) - \min_{\mathbf{x}} \ell(f, \mathbf{x}) \geq 0,$$

$$R_T \triangleq \sum_{t=1}^T r_t.$$

The *asymptotic no-regret* property, i.e., $\lim_{T \rightarrow \infty} \frac{R_T}{T} = 0$, is based on the sub-linear growth of R_T in T achieved by (13), which will be proved in the following content. A small cumulative regret R_T indicates that the achieved coverage cost $\ell(f, \mathbf{x}_t)$ remains close to the optimal cost, meaning that the robots attain coverage performance comparable to the ideal case with full knowledge of the true density function. In Algorithm 1, the non-decreasing sequence $\{\beta_t\}_{t=1}^T$ is required to achieve sub-linear regret, especially with the UCB sampling strategy, e.g., Calandriello et al. (2019); Srinivas et al. (2012); Benevento et al. (2020). Specifically we let

$$\beta_t \triangleq 2 \log \left(|\mathcal{D}| \frac{t^2 \pi^2}{6\delta} \right) \tag{13}$$

with $\delta \in (0, 1]$. The *asymptotic no-regret* property, i.e., $\lim_{T \rightarrow \infty} \frac{R_T}{T} = 0$, is based on the sub-linear growth of R_T in T achieved by (13). In Section 5, we set β_t as a constant value and observe that sublinear regret still remained. To proceed, we will need the following assumptions:

Assumption 1 *The optimal hyperparameters of $f(\cdot)$ exist, denoted as ρ^* and τ^* in equation (3).*

Assumption 2 *The domain \mathcal{D} is discretized.*

Assumption 3 *The control law solves the optimization problem $\mathbf{x}_t = \arg \min_{\mathbf{x}} \ell(\hat{f}_t, \mathbf{x})$.*

Assumption 1 simplifies the proof: it implies that the function is sampled from a known GP and has a low reproducing kernel Hilbert space (RKHS) norm (Srinivas et al., 2009). Note that $f(\cdot)$ represents the true density function, so ρ^* and τ^* are considered as hyperparameters of the true density and thus not accessible to the robots. Assumption 2 can also be further relaxed since the results can be generalized to a compact domain, and \mathcal{D} is discretized either in the simulations or the physical experiment of this paper. Assumption 3 is a theoretical idealization used for the regret analysis. However, the Voronoi-based coverage control law (2) typically achieves local optima without guaranteeing global optimality, e.g., Cortes et al. (2004); Luo et al. (2019). Nevertheless, extensive experiments in Section 5 empirically suggest that a local optimum (a CVT) reached by Algorithm 1 performs well in practice and may, in some cases, reach the global optimum. Importantly, our proposed framework provides the flexibility that alternative control laws can be directly incorporated without changing other components of Algorithm 1. Should future theoretical work prove that certain control laws attain global optimality, our analysis in this paper remains valid.

To establish the static regret bound using different density learning methods, we first introduce the following lemma.

Lemma 1 *Pick $\delta \in (0, 1]$, and set $\beta_t = 2 \log(|\mathcal{D}| \frac{\pi_t}{\delta})$, where $\sum_{t \geq 1} \pi_t^{-1} = 1, \pi_t > 0$. Then,*

$$|f(x) - \mu_{t-1}(x)| \leq \sqrt{\beta_t} \sigma_{t-1}(x)$$

holds for any $x \in \mathcal{D}$ and any $t \geq 1$ with probability at least $1 - \delta$.

Proof For GP, see Lemma 5.6 of Srinivas et al. (2012), while for RFGP and O-RFGP, see Appendices B and E. ■

We assume that the high probability event of Lemma 1 holds. Next, we will work towards the upper bound of accumulated regret through the following theorem.

Theorem 2 *Let $\delta \in (0, 1]$ and $\beta_t = 2 \log(|\mathcal{D}| \frac{t^2 \pi^2}{\delta})$. If the density function $f(\cdot)$ is the realization of a Gaussian Process with mean $\mu(\cdot)$ and covariance function $k(\cdot, \cdot) \leq 1$, then running Algorithm 1 ensures that*

$$\forall T \geq 1, R_t \leq \sqrt{c_1(n+1)T\beta_T\gamma_T}$$

holds with probability at least $1 - \delta$, where $c_1 \triangleq 8|D|^2 \max_{x, x'} \frac{\|x-x'\|_2^4}{\log(1+\sigma^{-2})}$. For kernels such that $\gamma_T = \mathcal{O}(T)$, Algorithm 1 has asymptotic no-regret.

Proof See Appendix C. ■

Note that the estimation step can also be implemented using either GP or RFGP. As shown in Srinivas et al. (2012), γ_T scales sub-linearly for GP with many kernels, e.g., $\gamma_T = \mathcal{O}(\log T)$ with

RBF kernel. However, γ_T for RFGP and O-RFGP usually cannot achieve information as well as GP since RFGP usually suffers from variance starvation (Calandriello et al., 2019; Wang et al., 2018), e.g., $\gamma_T = \mathcal{O}(\sqrt{T})$ with RBF kernel, details are shown in Appendix D. Nevertheless, RFGP and O-RFGP still ensure that γ_T scales sub-linearly, i.e., $\gamma_T = \mathcal{O}\left(2D \log\left(\frac{T}{D}\right)\right)$, with details in Appendix D.

As discussed in Lu et al. (2020); Shen et al. (2019), O-RFGP could potentially handle time-varying density, f_t . Extending this analysis to time-varying densities would require regret bounds that depend on a temporal variation measure of the density (Bogunovic et al., 2016). Since such variation budgets are typically unknown and may grow over time, we leave the regret analysis for the time-varying setting as future work. Nonetheless, Algorithm 1 is evaluated with a time-varying density function in Section 5.1.2 and Section 5.2, where the results are encouraging in the sense that O-RFGP learns a surrogate \hat{f}_t , and by running the control law (2) with \hat{f}_t , the team of robots can be guided to locations that efficiently cover the domain of interest, i.e., a time-varying CVT.

Appendix B. Proof of Lemma 1

In this section, we prove that Lemma 1 holds when the RFGP approach computes μ :

$$|f(x) - \mu_{t-1}(x)| \leq \sqrt{\beta_t} \sigma_{t-1}(x)$$

holds for any $x \in \mathcal{D}$ and any $t \geq 1$ with probability at least $1 - \delta$ while picking $\delta \in (0, 1]$, and set $\beta_t = 2 \log\left(|\mathcal{D}| \frac{\pi_t}{\delta}\right)$, where $\sum_{t \geq 1} \pi_t^{-1} = 1, \pi_t > 0$.

Proof: Since we can have $f(x) \sim \mathcal{N}(\mu_{t-1}(x), \sigma_{t-1}^2(x))$. Let $r \sim \mathcal{N}(0, 1)$ and $c_5 > 0$, we can have

$$P(r > c) = e^{-\frac{c^2}{2}} \frac{1}{\sqrt{2\pi}} \int e^{-\frac{(r-c_5)^2}{2-(r-c_5)}} \leq e^{-\frac{c^2}{2}} P(r > 0) = \frac{1}{2} e^{-\frac{c^2}{2}}.$$

Let $r = \frac{f(x) - \mu_{t-1}}{\sigma_{t-1}(x)}$ and $c_5 = \sqrt{\beta_t}$, then

$$P(|f(x) - \mu_{t-1}| > \sqrt{\beta_t} \sigma_{t-1}(x)) \leq e^{-\frac{\beta_t}{2}}.$$

Since the domain \mathcal{D} is discrete per Assumption 2, applying the union bound,

$$P\left(|f(x) - \mu_{t-1}| > \sqrt{\beta_t} \sigma_{t-1}(x), \forall x \in \mathcal{D}\right) \leq |\mathcal{D}| e^{-\frac{\beta_t}{2}},$$

which can be rewritten as

$$P\left(|f(x) - \mu_{t-1}| \leq \sqrt{\beta_t} \sigma_{t-1}(x), \forall x \in \mathcal{D}\right) \geq 1 - |\mathcal{D}| e^{-\frac{\beta_t}{2}}. \quad (14)$$

Choosing $|\mathcal{D}| e^{-\frac{\beta_t}{2}} = \frac{\delta}{\pi_t}$, we can have $\beta_t = 2 \log\left(|\mathcal{D}| \frac{\pi_t}{\delta}\right)$:

$$P\left(|f(x) - \mu_{t-1}| \leq \sqrt{\beta_t} \sigma_{t-1}(x), \forall x \in \mathcal{D}\right) \geq 1 - \delta.$$

Proof of Lemma 1 is completed.

Appendix C. Proof of Theorem 2

In the proof of Theorem 2, we assume the high-probability event of Lemma 1 holds. Firstly, we define

$$\begin{aligned}\ell^* &\triangleq \min_{\mathbf{x}} \ell(f, \mathbf{x}) \\ \mathbf{x}^* &\triangleq \arg \min_{\mathbf{x}} \ell(f, \mathbf{x})\end{aligned}$$

Based on Assumption 3, $\mathbf{x}_t = \arg \min_{\mathbf{x}} \ell(\hat{f}_t, \mathbf{x})$, we have

$$\ell(\hat{f}_t, \mathbf{x}_t) \leq \ell(\hat{f}_t, \mathbf{x}^*) \leq \ell^* \quad (15)$$

where the last inequality follows since the event in Lemma 1 holds:

$$\begin{aligned}\mu_{t-1}(x) - \sqrt{\beta_t} \sigma_{t-1}(x) &\leq f(x) \leq \mu_{t-1}(x) + \sqrt{\beta_t}, \\ \hat{f}_t(x) &= \mu_{t-1}(x) - \sqrt{\beta_t} \sigma_{t-1}(x), \\ \hat{f}_t(x) &\leq f(x) \quad \forall x \in \mathcal{D}, \\ \ell(\hat{f}_t, \mathbf{x}^*) &\leq \ell^*.\end{aligned}$$

Then, the regret at step t is

$$\begin{aligned}r_t &\triangleq \ell(f, \mathbf{x}_t) - \ell^* \stackrel{(a)}{\leq} \ell(f, \mathbf{x}_t) - \ell(\hat{f}_t, \mathbf{x}_t) \\ &\stackrel{(b)}{=} \sum_i \sum_{q \in \mathcal{V}_t^i} \|q - x_t^i\|_2^2 (f(q) - \hat{f}_t(q)) \\ &\stackrel{(c)}{\leq} 2\sqrt{\beta_t} \left(\sum_i \sum_{q \in \mathcal{V}_t^i} \|q - x_t^i\|_2^2 \right) \max_{q \in \mathcal{D}} \sigma_{t-1}(q) \\ &\leq 2c_0 \sqrt{\beta_t} \max_{q \in \mathcal{D}} \sigma_{t-1}(q) \triangleq 2c_0 \sqrt{\beta_t} \sigma_{t-1}(\tilde{x}_t),\end{aligned} \quad (16)$$

where (a) follows from (15), (b) follows by the definition of locational cost, (c) holds because $f(q) \leq \mu_{t-1}(q) + \sqrt{\beta_t} \sigma_{t-1}(q)$ and $\hat{f}_t(q) = \mu_{t-1}(q) - \sqrt{\beta_t} \sigma_{t-1}(q)$, and c_0 is defined as

$$c_0 = |\mathcal{D}| \max_{x, x' \in \mathcal{D}} \|x - x'\|_2^2.$$

Note that c_0 only depends on the domain \mathcal{D} and not on the number of robots n .

Since β_t is non-decreasing with respect to t , we have

$$(r_t)^2 \leq 4c_0^2 \beta_t (\sigma_{t-1}(\tilde{x}_t))^2 \quad (17)$$

$$\leq c_1 \beta_T \frac{1}{2} \log \left(1 + \left(\frac{\sigma_{t-1}(\tilde{x}_t)}{\sigma} \right)^2 \right), \quad (18)$$

by setting $c_1 = \frac{8c_0^2}{\log(1+\sigma^{-2})}$. Then, we have

$$\sum_{t=1}^T (r_t)^2 \leq c_1 \beta_T \sum_{t=1}^T \log \left(1 + \left(\frac{\sigma_{t-1}(\tilde{x}_t)}{\sigma} \right)^2 \right). \quad (19)$$

According to (19), we will derive the information gain which upper bounds the last term, i.e.,

$$\sum_{t=1}^T \log \left(1 + \left(\frac{\sigma_{t-1}(\tilde{\mathbf{x}}_t)}{\sigma} \right)^2 \right)$$

. For $t \in \{1, \dots, T\}$, we define $f(\mathbf{x}_{:t}) \in \mathbb{R}_{nt}$ as

$$f(\mathbf{x}_{:t}) \triangleq [f(\mathbf{x}_1^1) \cdots f(\mathbf{x}_1^n) \cdots f(\mathbf{x}_t^1) \cdots f(\mathbf{x}_t^n)]^\top.$$

Recall that for every robot $i \in \{1, \dots, n\}$ at each t , we have

$$y_t^i = f(x_t^i) + \epsilon,$$

where all noises are $\mathcal{N}(0, \sigma^2)$ and i.i.d. Then,

$$\begin{aligned} I(f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T}); \mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) &= h(\mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) - h(\mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T} | f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T})) \\ &\stackrel{(a)}{=} h(\mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) - \frac{(n+1)T}{2} \log(2\pi e \sigma^2), \end{aligned} \quad (20)$$

where (a) follows by

$$\begin{aligned} h(\mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T} | f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T})) &= - \sum_{y \in \mathbf{y}_{:T}, \tilde{y} \in \tilde{\mathbf{y}}_{:T}} \log p(y) \\ &= - \sum_{y \in \mathbf{y}_{:T}, \tilde{y} \in \tilde{\mathbf{y}}_{:T}} \log \left[\frac{1}{\sqrt{2\pi\sigma^2} e^{\frac{(y-f(x))^2}{2\sigma^2}}} \right] \\ &= - \sum_{y \in \mathbf{y}_{:T}, \tilde{y} \in \tilde{\mathbf{y}}_{:T}} \left(-\frac{1}{2} \log(2\pi\sigma^2) - 0 \right) \\ &= \frac{(n+1)T}{2} \log(2\pi\sigma^2). \end{aligned}$$

By the chain rule of differential entropy, we have

$$\begin{aligned} h(\mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) &= \sum_{t=1}^T h(\mathbf{y}_t, \tilde{y}_t; \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) \\ &= \sum_{t=1}^T h(\mathbf{y}_t; \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) + h(\tilde{y}_t; \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) \end{aligned} \quad (21)$$

According to ORC algorithm, the sequence of sensing points $\mathbf{x}_{:t}$ and $\tilde{\mathbf{x}}_{:t}$ is a *deterministic* function of $\mathbf{y}_{:t-1}$ and $\tilde{\mathbf{y}}_{:t-1}$. Since \tilde{y}_t and \mathbf{y}_t conditioned on $\mathbf{x}_{:t}, \tilde{\mathbf{x}}_{:t}, \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}$ that are distributed according to $\mathcal{N}(\mu_{t-1}(\tilde{x}_t), \sigma^2 + \sigma_{t-1}(\tilde{x}_t)^2)$ and $\mathcal{N}(\tilde{\mu}_t, K_t(\mathbf{x}_t))$ with $\tilde{\mu}_t \in \mathbb{R}^n$ and $K_t(\mathbf{x}_t) \in \mathbb{R}^{n \times n}$, we have

$$\begin{aligned} h(\tilde{y}_t; \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) &= h(\tilde{y}_t; \mathbf{x}_{:t}, \tilde{\mathbf{x}}_{:t}, \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) \\ &= \frac{1}{2} \log(2\pi e(\sigma^2 + \sigma_{t-1}(\tilde{x}_t)^2)), \end{aligned} \quad (22)$$

$$\begin{aligned} h(\mathbf{y}_t; \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) &= h(\mathbf{y}_t; \mathbf{x}_{:t}, \tilde{\mathbf{x}}_{:t}, \mathbf{y}_{:t-1}, \tilde{\mathbf{y}}_{:t-1}) \\ &= \frac{1}{2} \log |2\pi e(\sigma^2 \mathbf{I} + K_t(\mathbf{x}_t))|. \end{aligned} \quad (23)$$

Combining (22) and (23) together, we obtain

$$\begin{aligned}
 I(f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T}); \mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) &= \sum_{t=1}^T \left[\frac{1}{2} \log |2\pi e(\sigma^2 \mathbf{I} + \boldsymbol{\Sigma}_t)| + \frac{1}{2} \log (2\pi e(\sigma^2 + \sigma_{t-1}^2(\tilde{x}_t))) \right] \\
 &\quad - \frac{(n+1)T}{2} \log(2\pi e\sigma^2), \\
 &= \sum_{t=1}^T \left[\frac{1}{2} \log \left| \mathbf{I} + \frac{1}{\sigma^2} \mathbf{K}_t \right| + \frac{1}{2} \log \left(1 + \frac{\sigma_{t-1}^2(\tilde{\mathbf{x}}_t)}{\sigma^2} \right) \right] \\
 &\geq \sum_{t=1}^T \frac{1}{2} \log \left(1 + \left(\frac{\sigma_{t-1}(\tilde{x}_t)}{\sigma} \right)^2 \right). \tag{24}
 \end{aligned}$$

Since the exact behavior of $I(f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T}); \mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T})$ is difficult to analyze, we use the universal upper bound as [Srinivas et al. \(2012\)](#)

$$\begin{aligned}
 I(f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T}); \mathbf{y}_{:T}, \tilde{\mathbf{y}}_{:T}) &= I(\tilde{\mathbf{y}}_{:T}; f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T})) \\
 &\quad + \sum_{i=1}^n I(\mathbf{y}_{:T}^i; f(\mathbf{x}_{:T}), f(\tilde{\mathbf{x}}_{:T}) | \mathbf{y}_{:T}^1, \dots, \mathbf{y}_{:T}^{i-1}, \tilde{\mathbf{y}}_{:T}) \\
 &\leq I(\tilde{\mathbf{y}}_{:T}; f(\tilde{\mathbf{x}}_{:T})) + \sum_{i=1}^n I(\mathbf{y}_{:T}^i; f(\mathbf{x}_{:T}^i)) \\
 &\leq (n+a) \max_{\mathcal{A} \subset \mathcal{D}; |\mathcal{A}|=T} I(\mathbf{y}_{\mathcal{A}}; f_{\mathcal{A}}). \tag{25}
 \end{aligned}$$

Setting $\gamma_T \triangleq \max_{\mathcal{A} \subset \mathcal{D}; |\mathcal{A}|=T} I(\mathbf{y}_{\mathcal{A}}; f_{\mathcal{A}})$,

$$\gamma_T = I(\mathbf{y}_{\mathcal{A}}; f_{\mathcal{A}}) = \frac{1}{2} \log \left| \mathbf{I} + \frac{1}{\sigma^2} \mathbf{K}_{\mathcal{A}} \right|, \tag{26}$$

and combining (19), (24) and (25), we can obtain

$$\sum_{t=1}^T (r_t)^2 \leq c_1(n+1)\beta_T\gamma_T. \tag{27}$$

Per Cauchy-Schwarz inequality,

$$R_t = \sum_{t=1}^T r_t \leq \sqrt{c_1(n+1)T\beta_T\gamma_T}. \tag{28}$$

Appendix D. Information Gain γ_T for RFGP

Since $\mathbf{K}_{\mathcal{A}}$ is approximated as $\sigma_{\theta}^2 \boldsymbol{\Phi}_T \boldsymbol{\Phi}_T^{\top}$ for RFGP, the γ_T is

$$\gamma_T = \log \left| \mathbf{I} + \frac{\sigma_{\theta}^2}{\sigma^2} \boldsymbol{\Phi}_T \boldsymbol{\Phi}_T^{\top} \right|. \tag{29}$$

Based on the determinant of a square matrix,

$$\begin{aligned} \left| \mathbf{I} + \frac{1}{\sigma^2} \Phi_T \Phi_T^\top \right| &\leq \left(\frac{\text{trace}(\mathbf{I} + c_3 \Phi_T \Phi_T^\top)}{2D} \right)^{2D} \\ &= \left(\frac{\text{trace}(\mathbf{I}) + c_3 \text{trace}(\Phi_T \Phi_T^\top)}{2D} \right)^{2D}, \end{aligned} \quad (30)$$

where $c_3 \triangleq \frac{\sigma_\theta^2}{\sigma^2}$, $\text{trace}(\mathbf{I}) = 2D$ and

$$\begin{aligned} \text{trace}(\Phi_t \Phi_t^\top) &= \sum_{i=1}^{2D} k_{ii} = \sum_{i=1}^{2D} \frac{1}{D} (\phi_{\mathbf{v}}^\top(\mathbf{x}_1))_i (\phi_{\mathbf{v}}(\mathbf{x}_1))_i + \dots + \frac{1}{D} (\phi_{\mathbf{v}}^\top(\mathbf{x}_1))_T (\phi_{\mathbf{v}}(\mathbf{x}_1))_T \\ &\leq \frac{1}{D} \sum_{i=1}^{2D} NT = \mathcal{O}(2T). \end{aligned}$$

By combining them, we can get

$$\gamma_T \leq \log \left(\left(\frac{D + c_3 T}{D} \right)^{2D} \right) = \mathcal{O} \left(2D \log \left(\frac{T}{D} \right) \right). \quad (31)$$

Thus, the information gain, γ_T , scales sub-linearly for RFGP and O-RFGP.

Appendix E. Convergence of O-RFGP

Without noisy measurements, $|f(x) - \mu_t(x)| = \mathcal{O}(\frac{1}{\sqrt{T}})$ for GP (Srinivas et al., 2009) and RFGP (Rahimi and Recht, 2007). As for O-RFGP, we will additionally show that $|f(x) - \mu_t(x)| = \mathcal{O}(\frac{\log T}{T})$ in this section. For simplicity and clarity of proof, we set $n = 1$ then \mathbf{x}_t can be rewritten as x_t .

Lemma 3 *For O-RFGP,*

$$|f(x) - \mu_t(x)| = \mathcal{O}\left(\frac{\log T}{T}\right). \quad (32)$$

To prove Lemma 3, we define the negative log-likelihood loss and regret for prediction

$$\mathcal{L}(\mu(x; \theta); f(x)) \triangleq -\log p(f(x) | \mu(x; \theta)) \quad (33)$$

$$R_{\text{pred}}(T) \triangleq \sum_{t=1}^T \mathcal{L}(\mu(x_t; \theta); f(x_t)) - \sum_{t=1}^T \mathcal{L}(\mu(x; \theta^*); f(x_t)) \quad (34)$$

where θ^* denotes the optimal solution and $\mu(x; \theta)$ is computed as (9). We omitted the superscript ‘O’ here for clarity and readability. Then, we need to introduce additional assumptions of Lu et al. (2020):

Assumption 4 *The loss, \mathcal{L} , is continuously twice differentiable with $|\frac{d^2}{dz_t^2} \mathcal{L}(z_t; f(x_t))| \leq c_4$.*

Assumption 5 *The loss, \mathcal{L} , is convex and has bounded derivative $|\frac{d}{dz_t} \mathcal{L}(z_t; f(x_t))| \leq L$*

Assumption 6 The kernel, \bar{k} , is shift-invariant, standardized and bounded as $\bar{k}(x, x') \leq 1, \forall x, x'$

To establish the regret bound for O-RFGP, we also need the following lemma.

Lemma 4 Under Assumptions 4, for a fixed $\epsilon_2 > 0$, the following bound holds with probability at least $1 - 2^8 \left(\frac{\sigma_*}{\epsilon_2}\right)^2 \exp\left(\frac{-D\epsilon_2^2}{4d+8}\right)$

$$\sum_{t=1}^T \mathcal{L} \leq \sum_{t=1}^T \mathcal{L}(\phi_{\mathbf{v}}(x_t)\theta^*; f(x_t)) + \frac{\|\theta^*\|^2}{2\sigma_\theta} + D \log\left(1 + \frac{T\sigma_\theta^2}{2D}\right). \quad (35)$$

Proof Defining the cumulative loss over T slots with a fixed θ as

$$\mathcal{L}_\theta \triangleq -\log(f(\mathbf{x}_{:T})|\theta; \mathbf{x}_{:T}) = \sum_{t=1}^T \mathcal{L}(\phi_{\mathbf{v}}^\top(x_t)\theta; f(x_t)), \quad (36)$$

the expected cumulative loss over $q(\theta)$, a pdf of the fixed strategy θ , can be defined as

$$\bar{\mathcal{L}}_{q_\theta} \triangleq \mathbb{E}_q[\mathcal{L}_\theta] = \int_{\theta} q(\theta) \mathcal{L}_\theta d\theta. \quad (37)$$

Based on Bayes rule, we have

$$\begin{aligned} \sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) &= \sum_{t=1}^T -\log(f(x_t)|f(x_{t-1}); \mathbf{x}_{:t}) \\ &= -\log p(f(\mathbf{x}_{:t}); \mathbf{x}_{:t}). \end{aligned} \quad (38)$$

Employing the definition of the Kullback-Leibler (KL) divergence, we have

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) - \bar{\mathcal{L}}_{q_\theta} = \int_{\theta} q(\theta) \log \frac{p(f(\mathbf{x}_{:t})|\theta; \mathbf{x}_{:t})}{p(f(\mathbf{x}_{:t}); \mathbf{x}_{:t})} d\theta. \quad (39)$$

Since $p(f(\mathbf{x}_{:t}), \theta; \mathbf{x}_{:t}) = p(f(\mathbf{x}_{:t})|\theta; \mathbf{x}_{:t})p(\theta) = p(f(\mathbf{x}_{:t}); \mathbf{x}_{:t})p(\theta|f(\mathbf{x}_{:t}); \mathbf{x}_{:t})$, the RHS of (39) can be rewritten as

$$\begin{aligned} \int_{\theta} q(\theta) \log \frac{p(f(\mathbf{x}_{:t})|\theta; \mathbf{x}_{:t})}{p(f(\mathbf{x}_{:t}); \mathbf{x}_{:t})} d\theta &= \int_{\theta} q(\theta) \log \frac{p(\theta|f(\mathbf{x}_{:t}); \mathbf{x}_{:t})}{p(\theta)} d\theta \\ &= \int_{\theta} q(\theta) \log \frac{q(\theta)}{p(\theta)} d\theta - \int_{\theta} q(\theta) \log \frac{q(\theta)}{p(\theta|f(\mathbf{x}_{:t}); \mathbf{x}_{:t})} d\theta \\ &\leq KL(q(\theta)||p(\theta)). \end{aligned}$$

Hence, it holds for any $q(\theta)$ (Lemma 2 of Lu et al. (2020)) that

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) \leq \bar{\mathcal{L}}_{q_\theta} + KL(q(\theta)||p(\theta)). \quad (40)$$

Let $q(\theta) = \mathcal{N}(\theta; \theta^*, \xi^2 \mathbf{I}_{2D})$ and $p(\theta) = \mathcal{N}(\theta; \mathbf{0}, \sigma_\theta^2 \mathbf{I}_{2D})$, combined with (40), we can have

$$\begin{aligned} \sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) - \bar{\mathcal{L}}_{q_\theta} &\leq KL(q(\theta) \| p(\theta)) \\ &= 2D \log \sigma_\theta + \frac{1}{2\sigma_\theta} (\|\theta^*\|^2 + 2D\xi^2) - 2D - 2D \log \xi. \end{aligned} \quad (41)$$

Let $z_t = \phi_{\mathbf{v}}^\top(x_t)\theta$ and $z_t^* = \phi_{\mathbf{v}}^\top(x_t)\theta^*$. Taking Taylor's expansion of $\mathcal{L}(z_t; f(x))$ around z_t^* results in

$$\mathcal{L}(z_t; f(x)) = \mathcal{L}(z_t^*; f(x)) + \frac{d\mathcal{L}(z_t^*; f(x))}{dz_t} (z_t - z_t^*) + \frac{d^2}{dz_t^2} \mathcal{L}(h(z_t); f(x)) \frac{(z_t - z_t^*)^2}{2}, \quad (42)$$

where $h(z_t)$ is some function lying between z_t and z_t^* . Taking the expectation of (42) with respect to $q(\theta)$ leads to

$$\begin{aligned} \mathbb{E}_q &= \mathcal{L}(z_t^*; f(x)) + \frac{d\mathcal{L}(z_t^*; f(x))}{dz_t} \times 0 + \mathbb{E}_q \left[\frac{d^2}{dz_t^2} \mathcal{L}(h(z_t); f(x)) \frac{(z_t - z_t^*)^2}{2} \right] \\ &\stackrel{(a)}{\leq} \mathcal{L}(z_t^*; f(x)) + c_4 \mathbb{E} \left[\frac{(z_t - z_t^*)^2}{2} \right] \\ &\stackrel{(b)}{\leq} \mathcal{L}(\phi_{\mathbf{v}}^\top(x_t)\theta; f(x)) + \frac{c_4 \xi^2}{2}, \end{aligned} \quad (43)$$

where (a) makes use of Assumption 4, and (b) relies on the bound $\|\phi_{\mathbf{v}}(x_t)\|^2 \leq 1$. Summing (43) from $t = 1$ to T , we have

$$\bar{\mathcal{L}}_{q_\theta} \leq \mathcal{L}_{\theta^*} + \frac{Tc_4\xi^2}{2}. \quad (44)$$

Combining with (41), the following inequality holds

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) \leq \mathcal{L}_{\theta^*} + \frac{Tc_4\xi^2}{2} + 2D \log \sigma_\theta + \frac{1}{2\sigma_\theta} (\|\theta^*\|^2 + 2D\xi^2) - 2D - 2D \log \xi. \quad (45)$$

Since the RHS is a convex function of ξ with minimal value at $\xi^2 = \frac{2D\sigma_\theta^2}{2D+Tc_4\sigma_\theta^2}$, replacing ξ with the minimal value in (45) leads to

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) \leq \sum_{t=1}^T \mathcal{L}(\phi_{\mathbf{v}}(x_t)\theta^*; f(x_t)) + \frac{1}{2\sigma_\theta} \|\theta^*\|^2 + D \log \left(1 + \frac{Tc_4\sigma_\theta^2}{2D} \right).$$

Proof of Lemma 4 is completed. ■

Lemma 4 bounds the cumulative Bayesian loss. Next, we will further bound the loss of O-RFGP estimator relative to the best function estimator in the original RKHS.

Theorem 5 Under Assumptions 4, 5 and 6, and with optimal function estimator (\hat{f}^*) belonging to the RHKS \mathcal{H}^* induced by k^* , for a fixed $\epsilon_2 > 0$, the following bound holds with probability at least $1 - 2^8(\frac{\sigma_*}{\epsilon_2})^2 \exp(\frac{-D\epsilon_2^2}{4d+8})$

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) - \sum_{t=1}^T \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \leq \frac{(1 + \epsilon_2)^2}{2\sigma_{\theta^2}} + D \log \left(1 + \frac{Tc_4\sigma_{\theta}^2}{2D} \right) + \epsilon_2 LTC, \quad (46)$$

where C is a constant, $d = 2$ in our case. Let $\epsilon_2 = \mathcal{O}(\frac{\log T}{T})$, $R_{pred}(T) = \mathcal{O}(\log T)$.

Proof By Rahimi and Recht (2007), for a given shift-invariant standardized kernel \bar{k} , the maximum point-wise error of the RF kernel approximate is uniformly bounded with probability at least $1 - 2^8(\frac{\sigma_*}{\epsilon_2})^2 \exp(\frac{-D\epsilon_2^2}{4d+8})$,

$$\sup_{x, x'} \left| \phi_{\mathbf{v}}^{\top}(x) \phi_{\mathbf{v}}(x') - \bar{k}(x, x') \right| \leq \epsilon_2, \quad (47)$$

where ϵ_2 is a given constant, D is the number of the feature vectors, d is dimension of x , and $\sigma_*^2 \triangleq \mathbb{E}[\|\mathbf{v}\|^2]$ is the second-order moment of the RF vector \mathbf{v} .

The optimal function estimator in \mathcal{H}^* incurred by k^* is expressed as $\hat{f}(x) \triangleq \sum_{t=1}^T \hat{\alpha}_t k^*(x, x_t) = \sigma_{\theta}^2 \sum_{t=1}^T \hat{\alpha}_t \bar{k}(x, x_t)$; and its RF-based approximate is $\check{f}^*(x) \triangleq \phi_{\mathbf{v}}^{\top}(x) \theta^*$ with $\theta^* \triangleq \sigma_{\theta}^2 \sum_{t=1}^T \hat{\alpha}_t \phi_{\mathbf{v}}(x)$. Then, we can have

$$\begin{aligned} & \left| \sum_{t=1}^T \mathcal{L}(\check{f}^*(x_t); f(x_t)) - \sum_{t=1}^T \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \right| \\ & \stackrel{(a)}{\leq} \sum_{t=1}^T \left| \mathcal{L}(\check{f}^*(x_t); f(x_t)) - \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \right| \\ & \stackrel{(b)}{\leq} \sum_{t=1}^T L \sigma_{\theta}^2 \left| \sum_{t'=1}^T \hat{\alpha}_{t'} \phi_{\mathbf{v}}^{\top}(x_t) \phi_{\mathbf{v}}(x_{t'}) - \sum_{t'=1}^T \hat{\alpha}_{t'} \bar{k}(x_t, x_{t'}) \right| \\ & \stackrel{(c)}{\leq} \sum_{t=1}^T L \sigma_{\theta}^2 \sum_{t'=1}^T |\hat{\alpha}_{t'}| \left| \phi_{\mathbf{v}}^{\top}(x_t) \phi_{\mathbf{v}}(x_{t'}) - \bar{k}(x_t, x_{t'}) \right|, \end{aligned} \quad (48)$$

where (a) follows the triangle inequality, (b) makes use of Assumption 5, and (c) results from the Cauchy-Schwarz inequality. Combining (48) with (47), we find that

$$\left| \sum_{t=1}^T \mathcal{L}(\check{f}^*(x_t); f(x_t)) - \sum_{t=1}^T \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \right| \leq \sum_{t=1}^T L \sigma_{\theta}^2 \epsilon_2 \sum_{t'=1}^T |\hat{\alpha}_{t'}| \leq \epsilon_2 LTC, \quad (49)$$

where $C \triangleq \sum_{t=1}^T \sigma_{\theta}^2 |\hat{\alpha}_t|$. Thus, we can have

$$\left| \sum_{t=1}^T \mathcal{L}(\phi_{\mathbf{v}}^{\top}(x_t) \theta^*; f(x_t)) - \sum_{t=1}^T \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \right| \leq \epsilon_2 LTC. \quad (50)$$

The uniform convergence bound (47) and Assumption 6 imply that

$$\sup_{x, x'} \phi_{\mathbf{v}}^{\top}(x) \phi_{\mathbf{v}}(x') \leq \epsilon_2 + 1, \quad (51)$$

which leads to

$$\begin{aligned} \|\theta^*\|^2 &\triangleq \left\| \sigma_{\theta}^2 \sum_{t=1}^T \hat{\alpha}_t \phi_{\mathbf{v}}(x_t) \right\|^2 \\ &= \sigma_{\theta}^4 \sum_{t=1}^T \sum_{t'=1}^T \hat{\alpha}_t \hat{\alpha}_{t'} \phi_{\mathbf{v}}^{\top}(x_t) \phi_{\mathbf{v}}(x_{t'}) \\ &\leq (1 + \epsilon_2) C^2. \end{aligned} \quad (52)$$

Combing (50), (52) and Lemma 4, we can have

$$\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) - \sum_{t=1}^T \mathcal{L}(\hat{f}^*(x_t); f(x_t)) \leq \frac{(1 + \epsilon_2) C^2}{2\sigma_{\theta}^2} + D \log \left(1 + \frac{T c_4 \sigma_{\theta}^2}{2D} \right) + \epsilon_2 LTC. \quad (53)$$

Proof of Theorem 5 is completed. ■

Theorem 5 provides the upper bound of prediction for O-RFGP: $R_{\text{pred}}(T) = \mathcal{O}(\log T)$ while \mathcal{L} is Bayesian loss. Assuming the loss of optimal estimator \hat{f}^* is relatively negligible, $\mathcal{L}(\hat{f}^*(x_t); f(x_t)) \approx 0$, we can have $\sum_{t=1}^T \mathcal{L}(\mu(x; \theta); f(x)) = \mathcal{O}(\log T)$. Then, $\mathcal{L}(\mu_t(x); f(x))$ can be rewritten as

$$\begin{aligned} \mathcal{L}(\mu_t(x_t); f(x_t)) &= -\log p(f(x_t; \theta) | \mu_t(x_t)) \\ &= \frac{(f(x_t) - \mu_t(x_t))^2}{2\sigma_t^2} + \frac{1}{2} \log(2\pi\sigma_t^2). \end{aligned}$$

Since σ_t^2 is bounded due to the bounded kernel, the cumulative squared error also scales as $\mathcal{O}(\log T)$: by Cauchy-Schwarz inequality, we can have

$$\begin{aligned} \sum_{t=1}^T |f(x_t) - \mu_t(x_t)| &\leq \sqrt{T \sum_{t=1}^T (f(x_t) - \mu_t(x_t))^2} \\ &= \mathcal{O}(\sqrt{T \log T}). \end{aligned}$$

Since the cumulative absolute error is $\mathcal{O}(\sqrt{T \log T})$, the per-step error is

$$|f(x) - \mu_t(x)| = \mathcal{O}\left(\sqrt{\frac{\log T}{T}}\right).$$

Although it converges slower than $\mathcal{O}(\frac{1}{\sqrt{T}})$, it is still fast enough for the requirement in Lemma 1.

Appendix F. Experimental Settings and Implementation Details

The domain of interest \mathcal{D} is a rectangular domain with the origin located at the center of it, x -axis ranging from -1.6 to 1.6 meters, and y -axis ranging from -1 to 1 meter. The common parameters for the simulations are set as follows. The initial positions of the team of robots are $x_0^1 = [-0.8, -0.2]^\top$, $x_0^2 = [-0.8, 0.2]^\top$, $x_0^3 = [-0.4, -0.2]^\top$, $x_0^4 = [-0.4, 0.2]^\top$, $x_0^5 = [0, -0.2]^\top$, $x_0^6 = [0, 0.2]^\top$, $x_0^7 = [0.4, -0.2]^\top$, $x_0^8 = [0.4, 0.2]^\top$, $x_0^9 = [0.8, -0.2]^\top$, and $x_0^{10} = [0.8, 0.2]^\top$. The initialization of the learned density is a uniform distribution across \mathcal{D} . Each robot takes a noisy measurement of the density value at its current position at each time step. The parameters of RFGP for learning the mean and variance of the density function are $\tau_{\text{RF}} = 0.1$, $\sigma_\theta^2 = 5$. The non-decreasing sequence $\sqrt{\beta_t}$ is set as 10^{-3} for RFGP and 10^5 for O-RFGP. The noise ϵ is sampled from a zero-mean Gaussian distribution with $\sigma = 0.01$. The number of random features are $D = 20$ and $D = 40$ for RFGP and O-RFGP, respectively.

The time-invariant density function is defined as $f(x) = \sum_{i=1}^6 A_i \exp(-(x - \mathbf{m}_i)^\top S_i (x - \mathbf{m}_i))$, where the amplitudes are $A_1 = A_2 = A_3 = A_4 = 150$, $A_5 = A_6 = 50$, the means are $\mathbf{m}_1 = [1.3, -0.75]^\top$, $\mathbf{m}_2 = [-1.2, 0.7]^\top$, $\mathbf{m}_3 = \mathbf{m}_6 = [1, 0.8]^\top$, $\mathbf{m}_4 = \mathbf{m}_5 = [-1, -0.8]^\top$, and the covariance matrices are $S_1 = S_2 = \text{diag}(0.3^2, 0.8^2)$, $S_3 = \text{diag}(0.8^2, 0.4^2)$, $S_4 = \text{diag}(0.4^2, 0.8^2)$, $S_5 = S_6 = \text{diag}(0.85^2, 0.2^2)$.

The time-varying density function is defined as $f_t(x) = \sum_{i=1}^6 A_i(t) \cdot \exp(-(x - \mathbf{m}_i)^\top S_i (x - \mathbf{m}_i))$, where the amplitudes are $A_1 = A_2 = A_3 = A_4 = 150$, $A_5(t) = 50(1 + \sin(0.12t))$, $A_6(t) = 50(1 - \sin(0.12t))$, the means are $\mathbf{m}_1 = [1.3, -0.75]^\top$, $\mathbf{m}_2 = [-1.2, 0.7]^\top$, $\mathbf{m}_3 = \mathbf{m}_6 = [1, 0.8]^\top$, $\mathbf{m}_4 = \mathbf{m}_5 = [-1, -0.8]^\top$, and the covariance matrices are $S_1 = S_2 = \text{diag}(0.3^2, 0.8^2)$, $S_3 = \text{diag}(0.8^2, 0.4^2)$, $S_4 = \text{diag}(0.4^2, 0.8^2)$, $S_5 = S_6 = \text{diag}(0.85^2, 0.2^2)$.

Appendix G. Additional Experiments

Apart from Section 5.1, we perform several additional simulations for: 1. sensitivity study of the number of random features D and non-decreasing sequence β_t , and 2. comparison of the computational efficiency among GP, RFGP, and O-RFGP.

G.1. Sensitivity Study

G.1.1. DIMENSION OF RANDOM FEATURES D

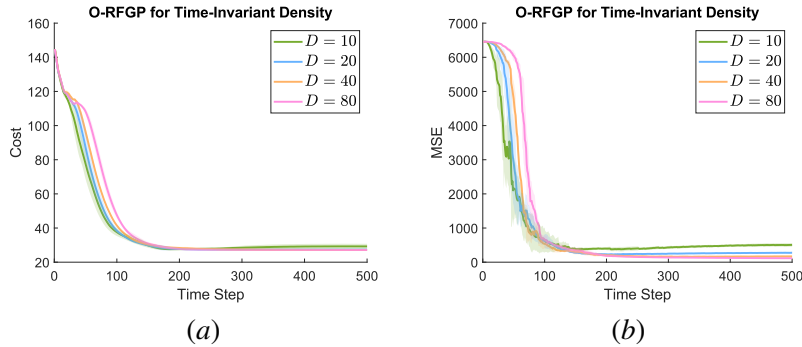


Figure 10: Sensitivity study of D when using O-RFGP to estimate the time-invariant density.

Table 2: Cost when using O-RFGP with different values of D to estimate the time-invariant density. Associated with Fig. 10(a).

D	$T = 50$	$T = 100$	$T = 200$	$T = 500$
10	77.61 ± 8.92	37.57 ± 2.25	27.63 ± 0.39	29.25 ± 1.52
20	87.28 ± 8.77	38.43 ± 1.46	27.72 ± 0.62	27.34 ± 0.02
40	96.35 ± 2.46	41.47 ± 0.53	28.32 ± 0.68	27.15 ± 0.22
80	108.18 ± 1.7754	47.57 ± 1.88	27.74 ± 0.25	27.14 ± 0.04

Table 3: MSE ($\times 10^3$) when using O-RFGP with different values of D to estimate the time-invariant density.. Associated with Fig. 10(b).

D	$T = 50$	$T = 100$	$T = 200$	$T = 500$
10	2.15 ± 0.91	0.65 ± 0.37	0.40 ± 0.04	0.51 ± 0.05
20	2.90 ± 1.24	0.62 ± 0.19	0.23 ± 0.01	0.28 ± 0.26
40	4.78 ± 0.53	0.54 ± 0.13	0.20 ± 0.04	0.17 ± 0.03
80	6.05 ± 0.15	0.66 ± 0.09	0.19 ± 0.02	0.12 ± 0.01

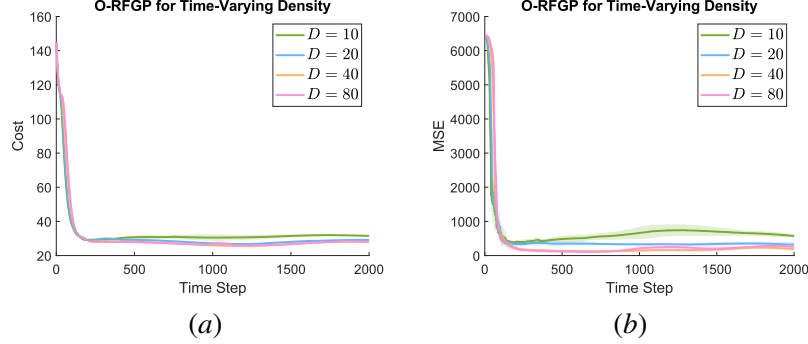


Figure 11: Sensitivity study of D when using O-RFGP to estimate the time-varying density.

Table 4: Cost when using O-RFGP with different values of D to estimate the time-varying density. Associated with Fig. 11(a).

D	$T = 50$	$T = 100$	$T = 500$	$T = 1000$	$T = 1500$	$T = 2000$
10	77.71 ± 5.06	37.95 ± 1.50	30.72 ± 1.06	30.49 ± 1.81	31.48 ± 0.88	31.49 ± 0.63
20	82.42 ± 1.35	38.45 ± 1.56	29.18 ± 0.80	27.18 ± 0.44	28.02 ± 0.61	29.15 ± 0.64
40	97.14 ± 5.16	42.79 ± 1.22	28.09 ± 0.13	26.00 ± 0.21	26.68 ± 0.21	28.04 ± 0.27
80	105.52 ± 2.26	45.11 ± 1.09	27.81 ± 0.23	26.71 ± 1.15	26.70 ± 0.24	28.12 ± 0.07

Table 5: MSE ($\times 10^3$) when using O-RFGP with different values of D to estimate the time-varying density. Associated with Fig. 11(b).

D	$T = 50$	$T = 100$	$T = 500$	$T = 1000$	$T = 1500$	$T = 2000$
10	1.60 ± 0.68	0.64 ± 0.20	0.49 ± 0.09	0.66 ± 0.18	0.70 ± 0.13	0.57 ± 0.05
20	2.06 ± 0.09	0.71 ± 0.31	0.35 ± 0.07	0.33 ± 0.04	0.34 ± 0.05	0.32 ± 0.07
40	5.22 ± 0.88	0.57 ± 0.07	0.14 ± 0.02	0.14 ± 0.03	0.17 ± 0.01	0.19 ± 0.04
80	5.93 ± 0.17	0.56 ± 0.12	0.19 ± 0.03	0.21 ± 0.06	0.20 ± 0.07	0.25 ± 0.06

To analyze the influence of the number of random features D , the results in Fig. 10, Table 2, Table 3, Fig. 11, Table 4 and Table 5 show that large values of D lead to accurate estimation of both time-invariant and time-varying densities, which in turn improves the overall coverage quality. Although the theoretical analysis holds only in the time-invariant setting, it suggests that similar principles also govern the performance of O-RFGP in the time-varying setting. In particular, as D increases, the MSE decreases, consistent with the analysis presented in Appendix E.

G.1.2. NON-DECREASING SEQUENCE β_t

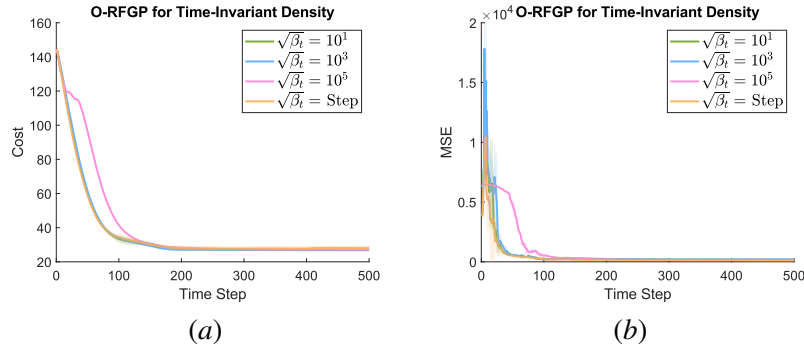


Figure 12: Sensitivity study of $\sqrt{\beta_t}$ when using O-RFGP to estimate the time-invariant density. “ $\sqrt{\beta_t} = \text{Step}$ ” denotes a piecewise constant schedule: $\sqrt{\beta_t} = 10^1$ for time steps 0-100, $\sqrt{\beta_t} = 10^2$ for time steps 100-200, $\sqrt{\beta_t} = 10^3$ for time steps 200-300, $\sqrt{\beta_t} = 10^4$ for time steps 300-400, and $\sqrt{\beta_t} = 10^5$ for time steps 400-500.

 Table 6: Cost when using O-RFGP with different values of D to estimate the time-invariant density. Associated with Fig. 12(a).

$\sqrt{\beta_t}$	$T = 50$	$T = 100$	$T = 200$	$T = 500$
10^1	60.10 ± 1.10	33.41 ± 2.00	27.83 ± 0.36	26.92 ± 0.21
10^3	61.53 ± 0.53	33.88 ± 0.96	27.09 ± 0.53	26.78 ± 0.28
10^5	96.35 ± 2.46	41.47 ± 0.53	28.32 ± 0.68	27.15 ± 0.22
Step	58.81 ± 0.11	34.40 ± 2.07	28.07 ± 1.42	28.05 ± 1.43

Table 7: MSE ($\times 10^3$) when using O-RFGP with different values of D to estimate the time-invariant density. Associated with Fig. 12(b).

$\sqrt{\beta_t}$	$T = 50$	$T = 100$	$T = 200$	$T = 500$
10^1	0.56 ± 0.06	0.24 ± 0.08	0.21 ± 0.07	0.19 ± 0.05
10^3	0.59 ± 0.08	0.21 ± 0.04	0.21 ± 0.01	0.20 ± 0.01
10^5	4.76 ± 0.53	0.54 ± 0.13	0.20 ± 0.04	0.17 ± 0.03
Step	0.50 ± 0.04	0.28 ± 0.19	0.17 ± 0.07	0.13 ± 0.03

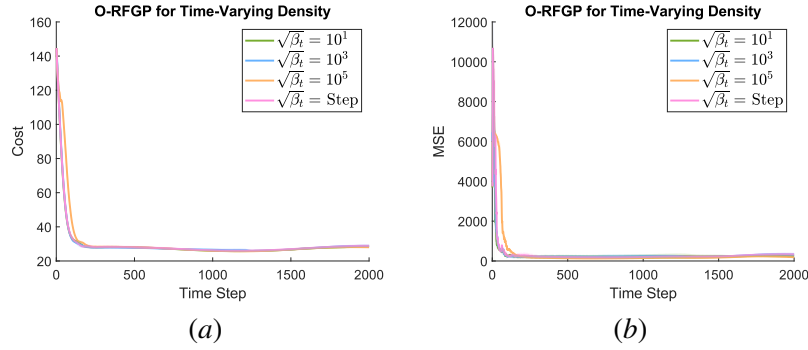


Figure 13: Sensitivity study of $\sqrt{\beta_t}$ when using O-RFGP to estimate the time-varying density. “ $\sqrt{\beta_t} = \text{Step}$ ” denotes a piecewise constant schedule: $\sqrt{\beta_t} = 10^1$ for time steps 0-500, $\sqrt{\beta_t} = 10^2$ for time steps 500-1000, $\sqrt{\beta_t} = 10^3$ for time steps 1000-1500, and $\sqrt{\beta_t} = 10^4$ for time steps 1500-2000.

Table 8: Cost when using O-RFGP with different values of D to estimate the time-varying density. Associated with Fig. 13(a).

$\sqrt{\beta_t}$	$T = 50$	$T = 100$	$T = 500$	$T = 1000$	$T = 1500$	$T = 2000$
10^1	59.13 ± 0.45	33.45 ± 1.28	28.02 ± 0.59	25.99 ± 0.35	26.68 ± 0.41	28.56 ± 0.52
10^3	60.60 ± 0.49	32.76 ± 0.79	27.68 ± 0.22	26.58 ± 0.78	26.77 ± 0.52	28.84 ± 0.71
10^5	97.14 ± 5.16	42.79 ± 1.22	28.09 ± 0.13	26.01 ± 0.21	26.68 ± 0.21	28.04 ± 0.27
Step	60.19 ± 1.37	33.71 ± 1.62	28.02 ± 0.83	26.22 ± 0.51	26.94 ± 0.53	28.86 ± 0.78

Table 9: MSE ($\times 10^3$) when using O-RFGP with different values of D to estimate the time-varying density. Associated with Fig. 13(b).

$\sqrt{\beta_t}$	$T = 50$	$T = 100$	$T = 500$	$T = 1000$	$T = 1500$	$T = 2000$
10^1	0.59 ± 0.12	0.26 ± 0.10	0.22 ± 0.09	0.22 ± 0.11	0.23 ± 0.08	0.25 ± 0.07
10^3	0.58 ± 0.09	0.21 ± 0.06	0.22 ± 0.08	0.24 ± 0.09	0.22 ± 0.004	0.32 ± 0.09
10^5	5.22 ± 0.88	0.57 ± 0.07	0.14 ± 0.02	0.14 ± 0.03	0.17 ± 0.01	0.19 ± 0.04
Step	0.54 ± 0.05	0.27 ± 0.05	0.20 ± 0.07	0.20 ± 0.05	0.23 ± 0.04	0.33 ± 0.08

To analyze the influence of the non-decreasing sequence β_t , the results in Fig. 12, Table 6, Table 7, Fig. 13, Table 8 and Table 9 show that ORC maintains sub-linear regret across a wide range of β_t values. Although a smaller β_t value yields slightly lower MSE, we select $\sqrt{\beta_t} = 10^5$ in most simulations as a larger value provides more precise boundary visualization in heatmaps.

G.2. Computational complexity

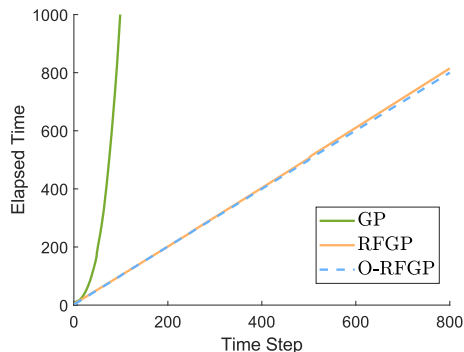


Figure 14: Comparison of the computational efficiency among GP, RFGP, and O-RFGP. The unit of the total elapsed time is the average value of the per-iteration running time of O-RFGP.

Fig. 14 shows that the running time of GP is significantly longer than those of RFGP and O-RFGP. At around time step 200, both RFGP and O-RFGP achieve roughly a thousandfold improvement in computational efficiency over GP. The results verify the theoretical results presented in Table 1. Notably, due to its lightweight computation, unlike the standard GP, O-RFGP is well-suited for real-time implementation on physical robotic platforms, as highlighted in Section 1 and demonstrated in Section 5.2. In addition, we conduct a case study using GP, where the robots are

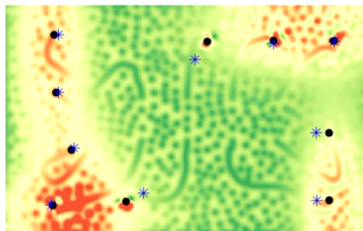


Figure 15: The learned time-varying density using GP (with all measurements collected from the beginning up to the current time step leveraged) at time step 760, where the black dots represent the positions of robots and the blue asterisks represent the centers of mass of their Voronoi cells.

ignorant of the time-dependency of the density function. Thus, all measurements collected from the beginning of the experiment up to the current time step are leveraged for density learning. In addition to the computational burden of GP, we find in a simulation that the learning result starts to diverge at around time step 760, as shown in Fig. 15. Although the density learning error of the GP decreases as the number of training samples increases, as analyzed in Section A, the number of mismatched data points increases due to the time-dependency of the density. This result indicates the limitations of the methods proposed in Srinivas et al. (2012); Williams and Rasmussen (2006);

[Calandriello et al. \(2019\)](#) in time-varying scenarios. Overall, as discussed in Section 1, GP is not suitable for real-time applications due to its high computational complexity, consistent with Table 1 and Section G.2.