2026 IEEE International Conference on Robotics and Automation (ICRA)
June 1-5, 2026. Vienna, Austria

Long-Term Mapping of the Douro River Plume with Multi-Agent Reinforcement Learning

Nicolò Dal Fabbro[1], Milad Mesbahi[1], Renato Mendes[2,3], João Borges de Sousa[2,3], George J. Pappas[1]

2026 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3315-8160-2/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICRA57385.2026.11696002

***Abstract*— We study the problem of long-term (multiple days) mapping of a river plume using multiple autonomous underwater vehicles (AUVs), focusing on the Douro river representative use-case. We propose an energy - and communication - efficient multi-agent reinforcement learning approach in which a central coordinator intermittently communicates with the AUVs, collecting measurements and issuing commands. Our approach integrates spatiotemporal Gaussian process regression (GPR) with a multi-head Q-network controller that regulates direction and speed for each AUV. Simulations using the Delft3D ocean model demonstrate that our method consistently outperforms both single- and multi-agent benchmarks, with scaling the number of agents both improving mean squared error (MSE) and operational endurance. In some instances, our algorithm demonstrates that doubling the number of AUVs can more than double endurance while maintaining or improving accuracy, underscoring the benefits of multi-agent coordination. Our learned policies generalize across unseen seasonal regimes over different months and years, demonstrating promise for future developments of data-driven long-term monitoring of dynamic plume environments.**

I. Introduction

Monitoring dynamic coastal environments in real-time is a persistent challenge in both environmental science and robotics [1], [2]. Coastal waters evolve under the interplay of currents, winds, tides, and river discharge, producing transient patterns that are difficult to capture with conventional methods. A prominent example of this dynamism is a river plume: a buoyant outflow of freshwater that extends into the ocean. River plumes, which are defined by their steep salinity gradients, are integral to the mixing processes that impact fisheries, water quality, and the spread of pollutants in the coastal areas [3], [4]. However, given their large extension in the ocean (hundreds of square kilometers) and rapid variability, tracking and mapping river plumes via fixed sensors or manned surveys is usually impractical [5].

Autonomous underwater vehicles (AUVs) offer a promising alternative. Indeed, AUVs can adapt their trajectories to the evolving conditions of aquatic environments, collect measurements across large spatial domains, and operate continuously over extended periods. Yet using AUVs for long-term plume monitoring introduces several challenges: (i) the salinity field that characterizes the plume, a scalar function of space and time, evolves on the same timescale as vehicle motion, meaning that the process shifts significantly while measurements are being collected; (ii) ocean currents in the plume area can dramatically hinder the AUVs mobility; (iii) endurance is constrained by onboard energy reserves, resulting in a trade-off between coverage and longevity; (iv) inter-agent communication in the ocean plume waters is severely constrained due to horizontal and vertical density variations, so AUVs usually need to dwell at the sea surface to transmit information [6]. Hence, coordinating multiple AUVs to map a river plume presents unique challenges, which we aim to address in this work.

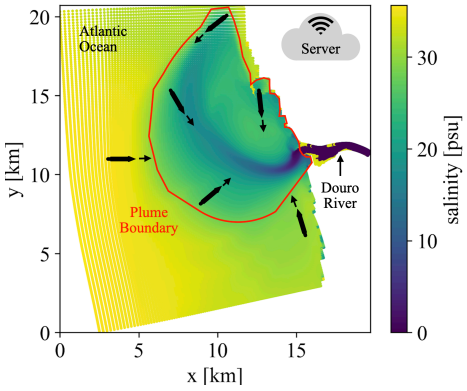


Fig. 1. Plume monitoring setting. The Douro River discharges into the Atlantic Ocean, generating a dynamic salinity plume (whose edge is represented by a red boundary). Multiple AUVs (black arrows) collect trajectory-constrained measurements and intermittently communicate with a central server to coordinate.

The Douro River plume, where freshwater from the river with the greatest discharge in Portugal's northwest coast enters the Atlantic, exemplifies these challenges and motivates this work. Its shape, extent, and orientation vary widely with river discharge, wind forcing, and tidal cycles. During high-flow periods, strong freshwater outflow drives the plume offshore, where it can be advected for tens of kilometers along the coast [7]. More than 500,000 residents of Porto and Vila Nova de Gaia live along the Douro banks and draw economic value from the estuary. Hence, timely and accurate salinity maps, which capture the state of the plume, are highly valuable for informed management [8], [9].

Contributions. In this paper, we propose and evaluate a cooperative data-driven multi-agent control framework for long-duration (multiple days), energy-aware mapping of the Douro river plume. In our proposed system architecture, a server remotely coordinates a fleet of multiple light AUVs

N. Dal Fabbro acknowledges the support from his AI x Science Postdoctoral Fellowship awarded through the University of Pennsylvania's IDEAS, DDDI, and Penn AI initiatives. R. Mendes and J. Borges de Sousa acknowledge the support of Fundação para a Ciência e a Tecnologia (FCT) for its financial support to LAETA via the project UID/50022/2025.

[1]Department of Electrical and Systems Engineering, University of Pennsylvania, USA. Correspondence to: ndf96@seas.upenn.edu.
[2]Laboratório de Sistemas e Tecnologia Subaquática (LSTS), Faculdade de Engenharia da Universidade do Porto, Portugal
[3] Laboratório Associado de Energia, Transportes e Aeronáutica (LAETA), INEGI, Porto, Portugal

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

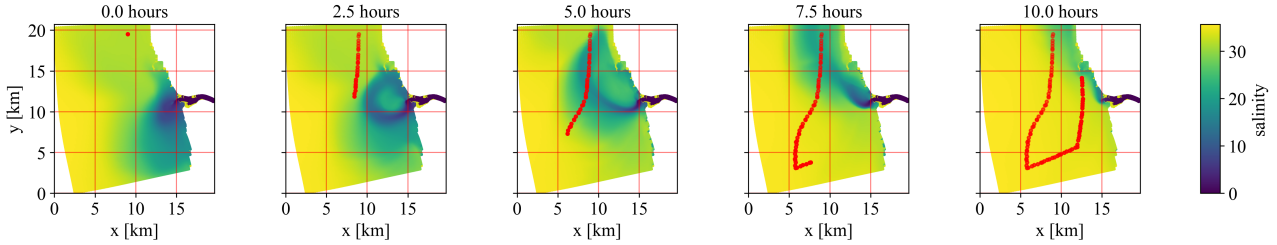


Fig. 2. Illustrative example of 10 hours of spatiotemporal evolution of the Douro river plume and AUV mobility (red trajectory), during March 2, 2018. The AUV uses the propulsion that provides the nominal speed of $1m/s$ (in absence of ocean flow). Note that the AUV mobility is impacted by the currents.

(LAUVs) [5], handling computationally heavy decisions and requiring only minimal, intermittent radio communication with the vehicles, which devote energy to low-level navigation and sensing. Within this setting, we propose an algorithm that combines a simple and reliable non-parametric Gaussian process regression (GPR) estimator with a reinforcement learning (RL) decision-making module based on a deep Q-network (DQN). Since most of the energy consumption of the considered LAUVs comes from the vehicle propulsion, we introduce a multi-head DQN architecture that decouples per-agent direction and speed decisions, paired with a reward function that allows us to control the tradeoff between estimation accuracy and energy efficiency.

We evaluate our approach and conduct a simulation study using a numerical model implementation of the Douro estuary with the open-source Delft3D ocean model [10], [11], which combines real coastal geometry and historical environmental data of the study area to generate high fidelity, realistic salinity and flow fields under distinct wind, tidal, and discharge regimes [11]. Our results show that the proposed approach (i) effectively maps the plume and generalizes to unseen conditions across different months and years; (ii) consistently outperforms baselines and benchmarks in both single- and multi-agent settings; (iii) effectively leverages multi-agent collaboration, both in terms of estimation accuracy and fleet endurance, for example scaling from 3 to 6 vehicles more than doubling mission lifetime.

Related work. Early coastal-robotics surveys rely on pre-planned coverage (lawn-mower transects, yo-yo depth cycling) [12], approaches still common operationally [13] but prone to oversampling and poor adaptability. Classic adaptive approaches frame the problem as informative path-planning (IPP) with Gaussian process surrogates [14], selecting waypoints that maximize posterior variance reduction or mutual information under travel constraints. Longer horizons have been pursued via rapidly exploring random trees embeddings [15] and non-myopic combinatorial methods such as GP-aware MILP and submodular branch-and-bound [16], yet these approaches generally rely on fixed graph discretizations, offer limited online re-planning, and scale poorly. Within plume monitoring, recent works focus on classification-oriented objectives such as expected integrated Bernoulli variance (EIBV) minimization, which prioritize sampling along uncertain river-ocean interface zones [17], [18]. While relevant, these methods are single-vehicle, single-step sighted, energy-unaware, and limited to short missions (∼4h). Multi-AUV studies, on the other hand, remain largely heuristic and oriented toward short-term front following rather than long-term full mapping [5]. Model-free RL offers an alternative to traditional IPP and plume monitoring methods. Convolutional neural network (CNN)-based DQN and multi-head DQN surpassed lawn-mower baselines for multi-agent lake monitoring [19], though only on static processes and coarse grids. Similarly, the work in [20] also explored the idea of combining GPR and RL on static, synthetic datasets. Aerial domains show similar patterns in static pollution plumes, wildfire tracking, and radio maps [21]–[23].

II. Problem Formulation

We model the salinity field of the Douro river plume as a spatiotemporal scalar function

$$f : \Omega \times \mathbb{R}_+ \to \mathbb{R} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ denotes a planar domain and $\mathbb{R}_+$ is time. In this work, we consider 2-dimensional mapping of the plume, which is usually considered the region in Ω where salinity is lower than the open-ocean background level f_{ocn} (≈ 35 psu). For a chosen threshold $\zeta > 0$, we can define the plume at time t as

$$\mathcal{P}(t) = \{\, x \in \Omega : f_{\text{ocn}} - f(x,t) \geq \zeta \,\}. \tag{2}$$

Such $\mathcal{P}(t)$ evolves dynamically and, depending on environmental conditions (such as wind, ocean flow, discharge level, and tidal cycle), at a speed comparable to AUVs mobility (1m/s), as we illustrate in Fig. 2.

Our objective in this work is to construct estimates $\hat{f}$ of the salinity field f over time, based on measurements collected by a fleet of N AUVs deployed over Ω. A vehicle n follows the following mobility model [24]:

$$\dot{x}^{(n)}(t) \;=\; u^{(n)}(t) + c\Big(x^{(n)}(t),\, t\Big)\,, \tag{3}$$

where $x^{(n)}(t) \in \Omega$ is the AUV position, $u^{(n)}(t) \in \mathbb{R}^2$ is its commanded velocity, and $c(x,t)$ is the ocean flow, described by a time-varying vector field

$$c : \Omega \times \mathbb{R}_+ \to \mathbb{R}^2, \quad c(x,t) = (c_{\text{lon}}(x,t), c_{\text{lat}}(x,t)), \tag{4}$$

where $c_{\text{lon}}(x,t)$ and $c_{\text{lat}}(x,t)$ are the longitude and latitude velocity components of the current, respectively.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

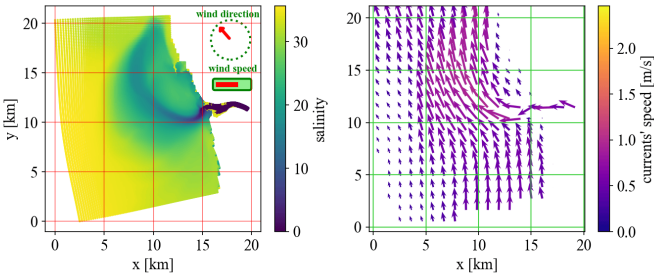


Fig. 3. Salinity map of the Douro river plume (on the left) and a visualization of the ocean flow (on the right). Note the correlation between the salinity map and the currents' distribution, and that the speed of the currents gets above $1m/s$, namely the nominal speed of the AUVs.

To construct the estimate $\hat{f}$ of the unknown function f over space and time, we want to collect measurements of the function f with the AUVs. Given that the AUVs move according to (3), the measurements that we can use for the estimation are *constrained* in space and time by their N trajectories. To formally state our problem formulation, we need to introduce some notation. Let us first define a vector of discrete time slots $\sigma = [0, \Delta, 2\Delta, ..., T\Delta]$ which we index with $\mathcal{K} = \{0, 1, ..., T\}$. In the rest of the paper, we are interested in updating the estimate $\hat{f}$ over the slots $\sigma[1], ..., \sigma[T]$ with a slot granularity of $\Delta = 30$ minutes.

Let $\gamma_k^{(n)} : [(k-1)\Delta, k\Delta] \to \Omega$ denote the trajectory navigated by agent n during time slot k and let $\Gamma_k^{(n)} = \left\{ \left(\gamma_k^{(n)}(t), t \right) : t \in [(k-1)\Delta, k\Delta] \right\}$. We denote the sampled set of space-time coordinates along $\Gamma_k^{(n)}$ by

$$Z_k^{(n)} = \left\{ \left(x^{(n)} \left(t_{k,1}^{(n)} \right), t_{k,1}^{(n)} \right), ..., \left(x^{(n)} \left(t_{k,z_k^n}^{(n)} \right), t_{k,z_k^n}^{(n)} \right) \right\},$$

with $z_k^n = |Z_k^{(n)}|$. Note that $(k-1)\Delta = t_{k,1}^{(n)} \leq \cdots \leq t_{k,z_k^n}^{(n)} = k\Delta$. Let the set of space-time locations visited up to time slot k by agent n be denoted by $D_k^{(n)} = \{Z_1^{(n)}, ..., Z_k^{(n)}\}$ and let $D_k^N = \{D_k^{(n)}\}_{n=1}^N$ be the overall set of coordinates visited by the fleet up to time k. We denote the noisy fleet-measured salinity values up to slot k by

$$f_k^N = \{f(x,t) + \epsilon_{x,t} : (x,t) \in D_k^N\}, \tag{5}$$

where $\epsilon_{x,t}$ is a generic noise term. Given space-time coordinates A, let $f(A) = \{f(x,t) + \epsilon_{x,t} : (x,t) \in A\}$ be the corresponding measurements, and let $\mathcal{M}_k^N = \{D_k^N, f_k^N\}$.

Given a measurement-based predictor $\hat{f}(\cdot | \mathcal{M}_k)$ for the salinity field $f(\cdot)$, and an evaluation grid $G \subset \Omega$, we formulate the long-term mapping problem as follows:

$$\textbf{Long-Term Mapping Problem} \begin{cases} \min\limits_{\{\mathcal{M}_k^N\}_{k=1}^T} & \dfrac{1}{|G|\,T} \displaystyle\sum_{k=1}^{T} \sum_{x \in G} e_k^2 \\ \text{s.t.} & Z_k^{(n)} \subset \Gamma_k^{(n)} \subset \Omega, \quad \forall k, n, \end{cases} \tag{6}$$

where note that in the case of the AUVs sampling the constraints are satisfied by construction, and

$$e_k = f\left(x, \sigma[k]\right) - \hat{f}\left(x, \sigma[k] \mid \mathcal{M}_k^N\right), \tag{7}$$



Fig. 4. Light autonomous underwater vehicles.

recalling that $\mathcal{M}_k^N = \{D_k^N, f_k^N\}$, and $D_k^N = \{D_k^{(n)}\}_{n=1}^N = \{\{Z_l^{(n)}\}_{l=1}^k\}_{n=1}^N$.

Discussion and challenges. Given that the function $f(\cdot)$ is unknown and measurements become available only in real time, problem (6) must be solved online, sequentially building the set $\mathcal{M}_k^N$, for $k = 1, ..., T$. The first main challenge in doing this is due to the physical constraints in building the trajectories D_k^N, *combined with the fact that the field $f(\cdot, t)$ changes in time at a speed comparable to the speed with which AUVs can move in the water*. Mathematically, the new locations $\{Z_k^{(n)}\}$ that we can add to the set D_{k-1}^N can only cover a limited space which is comparable to the space that the plume $\mathcal{P}(t)$ in (2) covers in the same amount of time. This is also referred to as the *lack of synopticity* in the trajectory-constrained measurements [25]. We illustrate this in the sequence of frames in Fig. 2, where we plot the evolution of the field $f(\cdot, t)$ over 10 hours and the corresponding mobility of one AUV across the area of interest. A second relevant challenge is that the ocean flow (illustrated in Fig. 3) can significantly impact or even nullify the mobility of an AUV. Note that the ocean flow can push at the same speed as the vehicle's nominal speed of 1m/s (mobility model in (3)), effectively stalling progress. At the same time, given that the current vector field $c(\cdot)$ is correlated with the salinity field and wind forcing, it may also be exploited for navigation if learned appropriately. These two aforementioned challenges strongly motivate the study of a data-driven, long-horizon sequential decision-making solution (e.g., RL), one which we provide in this work. Beyond these constraints, we also address two fundamental engineering aspects. First, multi-agent coordination depends on *communication*, which is intermittent and bandwidth-limited in ocean plume environments. Can we effectively orchestrate multiple AUVs in solving problem (6) with minimal and sporadic communication? Second, endurance hinges on energy efficiency. Can we take sequential decision-making actions that not only aim to keep the mean squared error (MSE) in (6) low, but which also smartly regulate the propulsion used to generate the trajectories $\Gamma_k^{(n)}$? Because hydrodynamic drag scales cubically with speed, halving velocity reduces energy consumption by a factor of eight [26], thus providing substantial gains in vehicle endurance. In the remainder of the paper, we provide an answer to these questions in the affirmative, illustrating our solution and the results of our simulation study.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

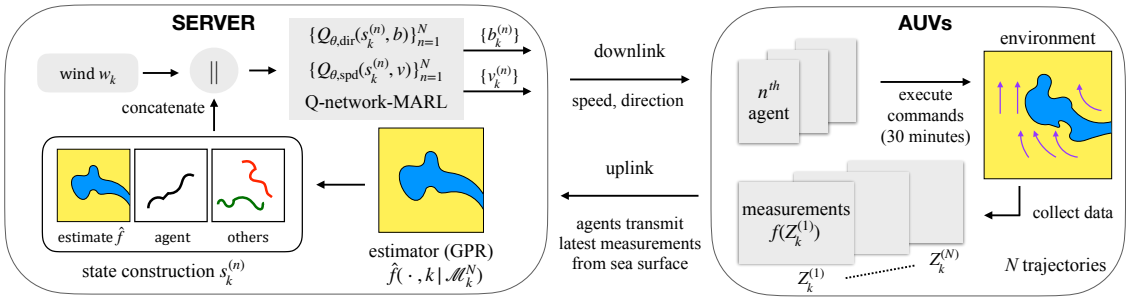


Fig. 5. System architecture.

LAUVs. The sensing agents considered in this work are light autonomous underwater vehicles (LAUVs) [5]. Each LAUV is equipped with a CTD probe (conductivity, temperature, depth) to measure salinity, navigation sensors, and communication modems (Wi-Fi, GSM, satellite) available when surfacing. The nominal survey speed of the LAUVs is 1.0 m/s, yielding an operational endurance of approximately 72 hours. As mentioned in the previous sections, smartly regulating the speed can provide dramatic energy savings, boosting the fleet's endurance. To do this, we introduce a second cruise speed, which we set to 0.4m/s. Traveling at this speed, the vehicle's endurance can be approximately 8 times longer. However, note that the rapid variability of plume dynamics and the strength of coastal currents prevent exclusive reliance on the secondary speed, highlighting the need for adaptive velocity control. On the computational side, each vehicle runs lightweight processors sufficient for low-level navigation, sensing and radio communication. We show a picture of LAUVs in Fig. 4.

III. Plume Monitoring with MARL

In this section, we describe the system architecture and the algorithm we design to solve problem (6). In Fig. 5 we provide a graphical illustration of the system architecture, and Algorithm 1 outlines the main control pipeline. The key rationales behind the system architecture are (i) offloading computation as much as possible from the AUVs to the server, prioritizing their endurance and (ii) relying only on intermittent and extremely low overhead communication. The pipeline at a given time slot k is as follows: first, all agents dwell on the sea surface and establish a connection with the cloud server via Wi-Fi or GSM. During this communication, each AUV $n = 1, ..., N$ uplinks its latest measurements $\{Z_k^{(n)}, f(Z_k^{(n)})\}$ collected over the past 30 minutes. In our implementation, the number of per-agent per-slot measurements z_k^n depends on speed and ocean currents, averaging 5 and limited to a maximum of 10. Note that transmitting 5 space-time coordinates and the corresponding measurements only involves transmitting 20 float numbers, which - in the worst case - only requires 160 bytes (always < 500 bytes even with wireless transmission overhead). Upon receiving the new measurements $\{Z_k^{(n)}, f(Z_k^{(n)})\}_{n=1}^N$, the server aggregates them with the previous set, resulting in $\mathcal{M}_k^N = \{D_k^N, f_k^N\} = \mathcal{M}_{k-1}^N \cup \{Z_k^{(n)}, f(Z_k^{(n)})\}_{n=1}^N$ and

Algorithm 1 Plume-DQN-GP

Input: AUVs $n = 1, ..., N$, action space ($\mathcal{H}$, $\mathcal{V}$), hyper-parameters
Initialization: deploy N AUVs in predetermined initial locations, directions b_0^n and speeds v_0^n for $n = 1, ..., N$
for $k = 1, \ldots, T$ **do**
  for $n = 1, \ldots, N$ in parallel **do**
    The n-th agent receives $(b_{k-1}^{(n)}, v_{k-1}^{(n)})$ commands from server and executes, visiting trajectory locations $Z_k^{(n)}$ and collecting measurements $f(Z_k^{(n)})$ for 30 minutes, and then transmits sets $\{Z_k^{(n)}, f(Z_k^{(n)})\}$ to server from sea surface.
  end for
  Server receives $\{Z_k^{(n)}, f(Z_k^{(n)})\}_{n=1}^N$, aggregates $\mathcal{M}_k^N = \{D_k^N, f_k^N\} = \mathcal{M}_{k-1}^N \cup \{Z_k^{(n)}, f(Z_k^{(n)})\}_{n=1}^N$, and uses it to update $\hat{f}(\cdot, \sigma[k] | \mathcal{M}_k^N)$.
  Server constructs per-agent state embeddings $s_k^{(n)}$ from $\hat{f}(\cdot, \sigma[k] | \mathcal{M}_k)$, trajectories D_k and wind vector.
  Server computes directions and speeds $\{b_k^{(n)}, v_k^{(n)}\}$ for $n = 1, ..., N$ using the Q-network, see (15).
  Server transmits $\{b_k^{(n)}, v_k^{(n)}\}$ in downlink to agents.
end for

updates the salinity map estimate $\hat{f}(\cdot, \sigma[k] | \mathcal{M}_k^N)$. Based on $\hat{f}$, the current agent trajectories D_k^N, and a vector containing the wind speed and direction w_k, the server determines the control policy $\pi(\hat{f}, \mathcal{M}_k^N, w_k)$ for all AUVs $n = 1, ..., N$ for the next 30 minutes time slot, which results in direction $\{b_k^{(n)}\}_{n=1}^N$ and speed levels $\{v_k^{(n)}\}_{n=1}^N$. At that point, the server downlinks the commands to the AUVs, which resubmerge and resume sampling, starting to build $\{Z_{k+1}^{(n)}, f(Z_{k+1}^{(n)})\}_{n=1}^N$. The estimator $\hat{f}(\cdot, \sigma[k] | \mathcal{M}_k^N)$ and the policy $\pi(\hat{f}, \mathcal{M}_k^N, w_k)$ form the core of our framework. We detail these two components in the following subsections, respectively.

A. Estimation Module

To produce measurement-based estimates $\hat{f}(\cdot, \sigma[k] | \mathcal{M}_k^N)$ we resort to non-parametric Gaussian process regression (GPR). GPR represents a very appealing choice in this case because (i) it encodes spatial and temporal correlations through a simple, interpretable kernel function; (ii) it is highly flexible and reliable when in need of interpolating the sparse and highly irregular samples $\mathcal{M}_k^N$ produced by

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

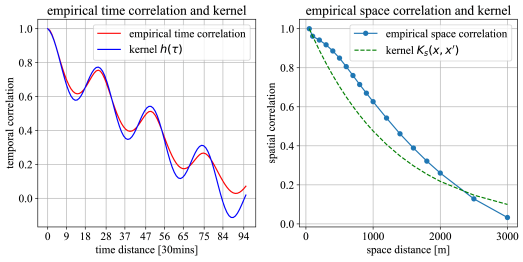


Fig. 6. Temporal and spatial kernels $h(\tau)$ and $K_s(x, x')$ fitting the corresponding empirical correlations.

the trajectory-constrained AUVs. Our GPR estimator places a spatiotemporal Gaussian prior on the salinity field of interest

$$f(x,t) \;\sim\; \mathcal{GP}\big(m(x,t),\, K\big((x,t),(x',t')\big)\big), \tag{8}$$

with mean function $m(\cdot)$ and covariance kernel $K(\cdot,\cdot)$. In this work, we fix the mean $m(x,t) = f_{\text{ocn}}$. Given the set $\mathcal{M}_k^N$, the posterior mean at (x,t) can be computed in closed form as:

$$\hat{f}(x,t \mid \mathcal{M}_k^N) = f_{\text{ocn}} + k_* \bar{K}^{-1}\big(f(D_k^N) - f_{\text{ocn}}\big) \tag{9}$$

where $k_* = K\big((x,t), D_k^{(N)}\big)$, $\bar{K} = K\big(D_k^{(N)}, D_k^{(N)}\big) + \sigma^2 I$ are the (x,t)-kernel section and the kernel data matrix, respectively, with σ^2 capturing the measurement noise. To keep computations tractable over long horizons, we retain only the most recent M slots in $\mathcal{M}_k^N$, discarding older samples.

Spatiotemporal kernel. We adopt a space-time separable spatiotemporal kernel of the following form:

$$K\big((x,t),(x',t')\big) = K_s(x,x')\, h\,(\tau)\,, \tag{10}$$

where $\tau = |t - t'|$. We choose the functions $K_s(x,x')$ and $h\,(|t-t'|)$ by analyzing and fitting the empirical correlations from the historical data. In particular, we set

$$K_s(x,x') = \lambda^2 \exp\!\left(-\frac{\|x - x'\|}{\ell}\right), \tag{11}$$

with hyperparameters $\lambda, \ell > 0$ and

$$h(\tau) = \beta_0 - \beta_1\,\tau + \beta_2\Big(\cos\big(\pi\tau/T_0\big) - 1\Big), \tag{12}$$

which combines a linearly decaying term and a periodic oscillation, whose period T_0, not surprisingly, is the same as the tidal period of 12.5 hours. All hyperparameters are fit to historical data by matching empirical correlations, as we illustrate in Figure 6. In the implementation, we made sure that the hyperparameters and memory window M ensure positive definiteness of the kernel.

B. Decision Making Module

We cast the construction of trajectories D_k^N over time slots $k = 1, ..., T$ to solve (6) as a (centralized) multi-agent sequential decision-making problem. In particular, we want to design a decision making policy $\pi(\cdot)$ that, at each time slot k, given measurements and trajectory history in $\mathcal{M}_k^N$, and exogenous inputs (e.g., wind), selects a direction and a speed level for each AUV for the next time slot (30 minutes). Given the large amount of historical environmental data and the availability of the advanced Delft3D simulator, a natural choice to learn $\pi(\cdot)$ is model-free multi-agent reinforcement learning (MARL) [27]. Due to the well-known problem of *centralized* MARL lacking scalability [28], we adopt the typical solution of treating the problem as if we were in a decentralized MARL setting with full communication [29]. To do so, we need to define a per-agent state space $\mathcal{S}$ and action space $\mathcal{A}$, and a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We seek a policy map $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative discounted reward $\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k)\right]$. At each time step k, the server builds the state $s_k^{(n)}$ for each agent using the current estimate $\hat{f}$ and the set D_k, and then selects an action $a_k^{(n)}$, for $n = 1, ..., N$. To learn policy $\pi(\cdot)$, we use Q-learning [27], which, for a given discount factor $\gamma > 0$, targets the optimal action–value function

$$Q^\star(s,a) = \mathbb{E}\Big[\, r + \gamma \max_{a'} Q^\star(s',a') \,\Big|\, s, a\Big], \tag{13}$$

and uses it to select actions. Since the state space is high-dimensional, we approximate $Q(s,a)$ with a neural network with parameters θ and denote it by $Q_\theta(s,a)$.

We factor the action space $\mathcal{A}$ into discrete direction and speed components

$$\begin{aligned} &a = (b, v) \;\in\; \mathcal{H} \times \mathcal{V} = \mathcal{A}, \;\text{where} \\ &\mathcal{H} = \{0, 45, ..., 315\}\;\text{deg},\; \mathcal{V} = \{0.4,\, 1.0\}\;\text{m/s}. \end{aligned} \tag{14}$$

The two velocity levels enable us to control the trade-off between endurance and agility: the lower cruise speed (0.4 m/s) extends mission duration by conserving propulsion energy, while the higher speed (1.0 m/s) enables more accurate plume monitoring at the expense of battery life.

To improve scalability, we decouple the choice of speed and direction, decomposing Q_θ into two components, each sharing a dueling architecture [30] with state-value function $V_{\bar{\theta},\theta_1}(s)$ and advantage functions $A^{\text{dir}}_{\bar{\theta},\theta_2}(s,b)$: $A^{\text{spd}}_{\bar{\theta},\theta_3}(s,v)$,

$$\begin{aligned} Q_{\theta,\text{dir}}(s,b) &= V_{\bar{\theta},\theta_1}(s) + A^{\text{dir}}_{\bar{\theta},\theta_2}(s,b) - \frac{1}{|\mathcal{H}|}\sum_{b' \in \mathcal{H}} A^{\text{dir}}_{\bar{\theta},\theta_2}(s,b') \\ Q_{\theta,\text{spd}}(s,v) &= V_{\bar{\theta},\theta_1}(s) + A^{\text{spd}}_{\bar{\theta},\theta_3}(s,v) - \frac{1}{|\mathcal{V}|}\sum_{v' \in \mathcal{V}} A^{\text{spd}}_{\bar{\theta},\theta_3}(s,v') \end{aligned},$$

where note that $\theta = \{\bar{\theta}, \theta_1, \theta_2, \theta_3\}$. For agent $n \in \{1, ..., N\}$ in state $s_k^{(n)}$, action selection is then performed per head:

$$\begin{aligned} b_k^{(n)} &= \arg\max_{b \in \mathcal{H}} Q_{\theta,\text{dir}}(s_k^{(n)}, b), \\ v_k^{(n)} &= \arg\max_{v \in \mathcal{V}} Q_{\theta,\text{spd}}(s_k^{(n)}, v), \end{aligned} \tag{15}$$

which establishes direction and speed.

State construction. The state representation $s_k^{(n)}$ comprises three components. Following prior work on image-based policy inputs [31], we render both spatial estimates and trajectory traces into a fixed-resolution 3-channels image, compressed by a convolutional neural network (CNN) into a compact embedding. Specifically, the three channels for the n-th agent are (i) a compressed image of the GP estimate of the salinity field $\hat{f}(\cdot, \sigma[k])$, (ii) one image marking in white

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

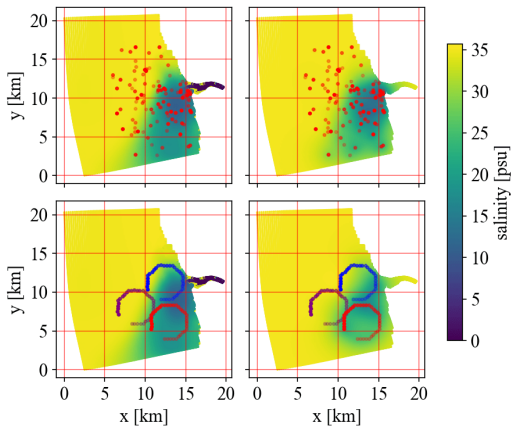


Fig. 7. Left column and right column: ground-truth and estimated salinity field, respectively. On top: uniform random sampling on the non-homogeneous spatial grid, with measurement locations denoted as red points. At the bottom, example of trajectory-constrained measurements: fixed rotations' trajectories of three agents (in different colors).

over a black background recent measurement locations of the agent, $Z_k^{(n)}, Z_{k-1}^{(n)}, ...$ (iii) one image marking in white over a black background recent measurement locations for agent n teammates, $\{Z_k^{(l)}, Z_{k-1}^{(l)}, ...\}_{l \neq n}$. The trajectory marks intensities are log-scale weighted by recency. In addition, a wind vector $w_k = [\text{angle}, \text{speed}]$, processed by a NN to produce an embedding, is concatenated with the rest of the network. See Fig. 5 for an illustration of the state construction. Including the GP variance in the state did not provide improvements in our experiments, so we do not include it.

Reward design. We design a reward function which combines cooperative and individual components. This incentivizes agents to cooperate by minimizing the global MSE while also seeking individual credit for reducing MSE along their own trajectory. Indeed, experimenting with a purely cooperative global reward, we incurred in the well-known *credit assignment* problem [32]. Let us introduce a salinity contrast score

$$F_k = \left| f_{\text{ocn}} - \frac{1}{|G|} \sum_{x \in G} f(x, \sigma[k]) \right|, \quad f_{\text{ocn}} \approx 35 \text{ psu}, \quad (16)$$

which characterizes how much freshwater is present in the ocean at time slot k. Our reward function for agent n is

$$\begin{aligned} r_k^{(n)} &= r_{k,g} + r_{k,n}, \text{ where} \\ r_{k,g} &= -\eta_0 e_k^G + \eta_1 \frac{F_k}{1 + e_k^G} - \eta_2 \sum_{n=1}^{N} v_k^{(n)}, \\ r_{k,n} &= \eta_3 \sum_{(x,t) \in Z_k^{(n)}} \left(\hat{f}_{k-1}(x) - f(x, \sigma[k]) \right)^2, \end{aligned} \quad (17)$$

where e_k^G is the global MSE in the evaluation grid G, $\hat{f}_{k-1}(x) = \hat{f}(x, \sigma[k-1] | \mathcal{M}_{k-1}^N)$ and $\eta_0, ..., \eta_3 > 0$ are hyperparameters. The term $r_{k,n}$ is the key individual credit that the agent earns for visiting locations x along its trajectory $Z_k^{(n)}$, where the previous estimate $\hat{f}_{k-1}(x)$ was significantly inaccurate relative to the actual field $f(x, \sigma[k])$. The term $\sum_{n=1}^{N} v_k^{(n)}$ penalizes higher speed at a fleet level.

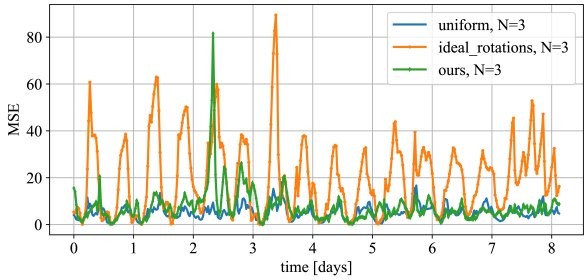


Fig. 8. Comparison of MSE over time for the two baselines (uniform and rotations) and our algorithm, interval of 8 days beginning of February 2018

The salinity contrast score F_k incentivizes a reduction in the MSE when there is more freshwater in the plume.

IV. Simulation Study and Results

In this section, we illustrate the details of our simulation study and of our numerical results.

Dataset. We train our algorithm over a selection of 4 months data from the year 2018 (February, April, October and December) and evaluate over different months from 2016-2018 (see Table I). Overall, we use a total of $\approx 6k$ time frames for the training, with a 30 minutes time resolution. Each frame spans $\approx 5 \times 10^4$ grid locations and measurement noise is i.i.d. $\mathcal{N}(0,\ 0.01)$.

Training setup. We minimize the Bellman loss for the Q-network, computed over mini-batches of size 64 sampled from a replay buffer. We use Adam optimizer, and we perform soft updates using a target network. Each policy used in this section has been trained for 6500 episodes of length 150 ($\approx$ 3 days) and a discount factor $\gamma = 0.9$, with the first 1500 episodes of pure exploration. The Q-network uses a 3-layer CNN followed by two fully connected layers.

Two baselines. In Fig. 7, we illustrate two different sampling schemes on ground-truth salinity fields f (left) and their GP estimates $\hat{f}$ (right) for October 2018. Both methods use the same budget of 15 measurements every 30 minutes with a memory window of 24 frames, and all estimates employ the GPR model of Sec. III-A. The top panels show an unconstrained uniform strategy: measurements are placed uniformly at random in the spatial non-homogeneous grid of the Delft3D numerical model, which has a higher density of grid points in higher variance areas. In practice, this effectively biases sampling toward high-variance locations closer to the river mouth. Note that since uniform placement ignores the trajectory constraint $Z_k^{(n)} \subset \Gamma_k^{(n)}$, $\forall k, n$, it is not physically realizable with the AUVs, but does provide a fundamental baseline. If one could freely deploy measurements without vehicle constraints, this uniform scheme would represent a natural choice for estimating f. In the bottom panels of Figure 7, we show an example of a predetermined strategy which is compliant with the constraints in (6). This particular strategy is based on rotations around some strategic core points in the plume, and, for the sake of this illustration, we assume that the robots are able to perform the rotations without being impacted by the ocean currents. Although the trajectories are strategically placed in key areas, one

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

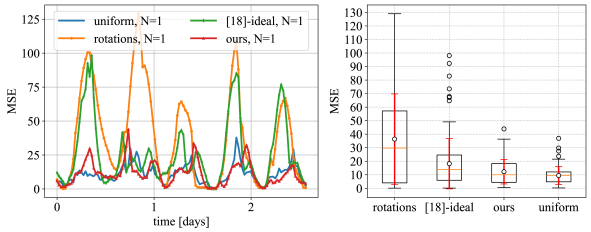


Fig. 9. Example of mapping performance when using a *single* ($N = 1$) AUV. Comparison between our algorithm, baselines and the benchmark [18]. On the left, an example of the MSE evolution over time for a small time interval ($\approx$ 3 days). On the right, box plots of the MSE over time obtained by the algorithms executed over the whole month of February 2018.

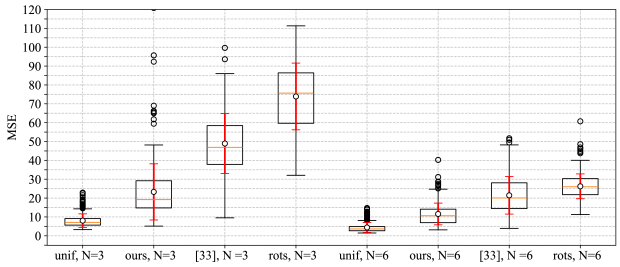


Fig. 10. Multi-agent performance for the test month of March 2018. "unif" and "rots" stand for uniform and rotations baselines, respectively.

can already see from Fig. 7 how the constrained sampling estimate struggle to capture the state of the plume, while the uniform sampling strategy provides a better estimation. To illustrate this numerically, in Figure 8 we show the evolution of the MSE over time for these two baselines ("uniform" and "ideal rotations"). One can see the notable difference in terms of MSE for the two different strategies: the MSE of the rotations is roughly 6 times higher than the unconstrained uniform baseline under the same sampling budget.

Single-agent performance. First, we analyze the performance of our algorithm when deploying only one AUV $N = 1$ to map the river plume. This allows us to compare our solution directly with the one proposed in [18], which only considered a single-agent setting, and which is based on evaluating the expected integrated Bernoulli variance (EIBV) for each potential direction at every step. For this benchmark, we implement an *ideal* version in which the decision maker has access to the actual IBV for each candidate direction, and we term this algorithm "ideal [18]". This allows us to compare against the best possible version of [18] and to obtain results which do not depend on the specific numerical way of computing the EIBV. A fundamental parameter of [18] is a salinity threshold that defines the plume front. While the original paper fixed this at 24 psu, we found this too low for the Douro plume and selected the best threshold from $\{28, 30, 32, 34\}$, choosing 32 psu. We show an example of simulation outcome in Figure 9, where on the left, we plot MSE evolution over three days for four algorithms, and on the right, we report monthly MSE box plots for February 2018. The two plots reveal how, despite [18] performing well at certain parts of the process, it tends to lose track of the plume over the long run, while our RL policy, in contrast, consistently maintains low MSE. This result is not surprising,

TABLE I

POLICY GENERALIZATION ACROSS SEASONAL REGIMES

	$N = 3$			$N = 6$		
Month	**Ours** MSE	End. (d)	[33] MSE	**Ours** MSE	End. (d)	[33] MSE
Mar '18	**23.14**	3.5	47.87	**11.46**	3.1	21.07
Sep '18	**10.36**	4.6	18.48	**5.44**	5.0	6.27
Nov '18	**13.96**	4.2	20.99	**7.88**	4.3	8.24
Jan '16	**27.03**	3.2	49.19	**12.84**	3.3	19.53
Feb '16	**24.26**	3.3	49.74	**14.07**	3.4	21.32
Oct '17	**3.69**	13.0	4.04	**2.79**	15.6	2.04

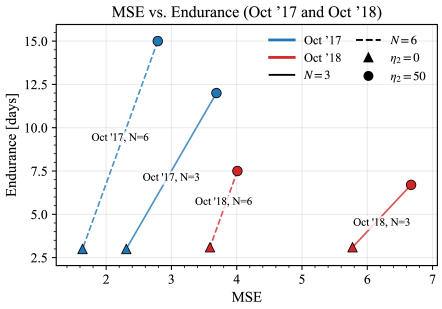


Fig. 11. Endurance-MSE tradeoff for the representative month of October.

as the quick temporal evolution of the plume can make the decisions taken by [18] based on the current prior Gaussian field myopic with respect to the spatiotemporal evolution of the plume, whereas our learned policy explicitly optimizes long-horizon mapping performance.

Multi-agent performance. We now evaluate performance in the multi-agent setting, focusing on: (i) MSE gains provided by the multi-agent coordination when increasing N, and on the comparison with existing literature benchmarks; (ii) the energy efficiency improvements induced by varying the η_2 reward hyperparameter in (17) and the corresponding MSE-energy tradeoff. As a multi-agent benchmark, we implement the recent work in [33] that proposed an algorithm based on adaptive Voronoi partitioning of the space of interest to track a time-varying process. We empirically tune the exploration-exploitation hyperparameters in [33] and show the best results in terms of MSE. In Fig. 10, we compare the different schemes for $N = 3$ and $N = 6$ for March 2018. We first note how the adaptive Voronoi partition solution [33] significantly outperforms the strategic rotations solution. We also note that our algorithm provides maps with half of the MSE compared to [33], for both $N = 3$ and $N = 6$. In Fig. 12, we show a sequence of frames to illustrate the multi-agent control performance of the agents with respect to the salinity field f evolution over a 15 hour time interval. In Fig. 11 we show the endurance–MSE trade-off obtained by varying the reward weight η_2 in (17), using October 2017 and October 2018 as exemplars. Increasing η_2 biases the policy toward lower propulsion use, thus extending endurance at a modest cost in accuracy, while scaling from $N = 3$ to $N = 6$ both reduces MSE and more than doubles endurance. Table I summarizes results obtained with $\eta_2 = 50$, and further confirms that our algorithm generalizes across unseen (during training) years and seasonal regimes.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.

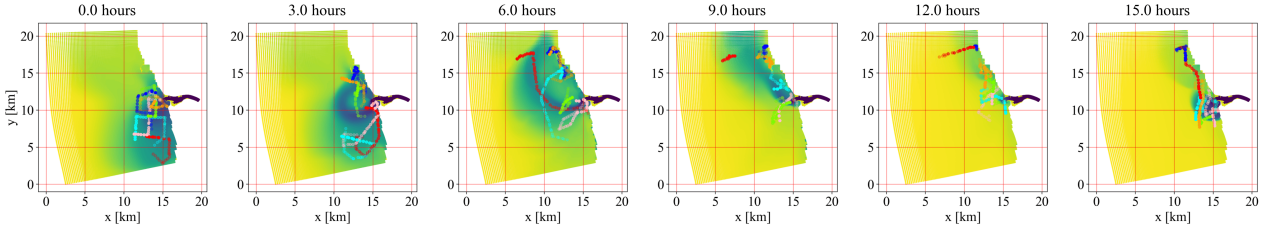


Fig. 12. Example of the multi-agent control performance of our solution with $N = 6$ agents over a 15 hours time window, in the testing month of March 2018. The colored points show the different agents' trajectories and corresponding measurements' locations.

V. Conclusion and Future Work

We conducted a simulation study showing promising performance in mapping the Douro river plume with multiple underwater vehicles over long time horizons. Future works include considering 3D mapping and real-world deployment of the LAUVs robots in the river plume.

References

[1] S. Raine and T. Fischer, "Ai-driven marine robotics: Emerging trends in underwater perception and ecosystem monitoring," *arXiv preprint arXiv:2509.01878*, 2025.

[2] A. Pastra, T. Johansson, J. Soares, and F. E. Muller-Karger, "The use of emerging autonomous technologies for ocean monitoring: insights and legal challenges," *Frontiers in Marine Science*, vol. 12, p. 1561737, 2025.

[3] R. Mendes, N. Vaz, D. Fernández-Nóvoa, J. Da Silva, M. Decastro, M. Gómez-Gesteira, and J. Dias, "Observation of a turbid plume using modis imagery: The case of douro estuary (portugal)," *Remote sensing of environment*, vol. 154, pp. 127–138, 2014.

[4] A. R. Horner-Devine, R. D. Hetland, and D. G. MacDonald, "Mixing and transport in coastal river plumes," *Annual Review of Fluid Mechanics*, vol. 47, no. 1, pp. 569–594, 2015.

[5] D. Teixeira, J. B. d. Sousa, R. Mendes, and J. Fonseca, "3D Tracking of a River Plume Front with an AUV," in *OCEANS 2021: San Diego – Porto*, 2021.

[6] T. Yan, Z. Xu, S. X. Yang, and S. A. Gadsden, "Formation control of multiple autonomous underwater vehicles: a review," *Intelligence & Robotics*, vol. 3, no. 1, 2023.

[7] M. E. Vieira and A. A. Bordalo, "The douro estuary (portugal): a mesotidal salt wedge," *Oceanologica Acta*, vol. 23, no. 5, pp. 585–594, 2000.

[8] E. Estévez, T. Rodriguez-Castillo, A. M. González-Ferreras, M. Cañedo-Argüelles, and J. Barquin, "Drivers of spatio-temporal patterns of salinity in spanish rivers: a nationwide assessment," *Philosophical Transactions of the Royal Society B*, vol. 374, no. 1764, p. 20180022, 2019.

[9] R. Friedland, T. Neumann, S. Piehl, H. Radtke, and G. Schernewski, "Characterization of river plume dynamics for a better water quality management," *Frontiers in Marine Science*, vol. 12, p. 1617660, 2025.

[10] H. Gerritsen, E. De Goede, F. Platzek, M. Genseberger, J. Van Kester, and R. Uittenbogaard, "Validation Document Delft3D-FLOW; a software system for 3D flow simulations," *The Netherlands: Delft Hydraulics, Report X*, vol. 356, p. M3470, 2007.

[11] M. C. Sousa, A. S. Ribeiro, M. Des, R. Mendes, I. Alvarez, M. Gomez-Gesteira, and J. M. Dias, "Integrated high-resolution numerical model for the nw iberian peninsula coast and main estuarine systems," *Journal of Coastal Research*, no. 85, pp. 66–70, 2018.

[12] Y. Zhang, J. G. Bellingham, J. P. Ryan, B. Kieft, and M. J. Stanway, "Two-dimensional mapping and tracking of a coastal upwelling front by an autonomous underwater vehicle," in *2013 OCEANS - San Diego*, 2013.

[13] J. Hwang, N. Bose, and S. Fan, "AUV Adaptive Sampling Methods: A Review," *Applied Sciences*, vol. 9, no. 15, 2019.

[14] J. Das, J. Harvey, F. Py, H. Vathsangam, R. Graham, K. Rajan, and G. S. Sukhatme, "Hierarchical probabilistic regression for auv-based adaptive sampling of marine phenomena," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5571–5578.

[15] G. Hollinger and G. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, pp. 1271–1287, 08 2014.

[16] S. Dutta, N. Wilde, and S. L. Smith, "Informative path planning in random fields via mixed integer programming," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022.

[17] Y. Ge, J. Eidsvik, and T. Mo-Bjørkelund, "3-D adaptive AUV sampling for classification of water masses," *IEEE Journal of Oceanic Engineering*, vol. 48, no. 3, pp. 626–639, 2023.

[18] M. O. Berild, Y. Ge, J. Eidsvik, G.-A. Fuglstad, and I. Ellingsen, "Efficient 3d real-time adaptive auv sampling of a river plume front," *Frontiers in Marine Science*, vol. Volume 10 - 2023, 2024.

[19] S. Y. Luis, D. G. Reina, and S. L. T. Marín, "A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: The Ypacaraí lake patrolling case," *IEEE access*, vol. 9, pp. 17 084–17 099, 2021.

[20] L. Zhao, "Adaptive path planning using Gaussian process regression: a reinforcement learning approach," in *Fourth International Conference on Signal Processing and Computer Science (SPCS 2023)*, 2023.

[21] M. S. Assenine, W. Bechkit, I. Mokhtari, H. Rivano, and K. Benatchba, "Cooperative deep reinforcement learning for dynamic pollution plume monitoring using a drone fleet," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 7325–7338, 2023.

[22] E. Krijestorac and D. Cabric, "Deep learning based active spatial channel gain prediction using a swarm of unmanned aerial vehicles," *arXiv preprint arXiv:2310.04547*, 2023.

[23] A. Viseras, M. Meißner, and J. Marchal, "Wildfire front monitoring with multiple uavs using deep q-learning," *IEEE Access*, 2021.

[24] M. Aguiar, J. B. de Sousa, J. M. Dias, J. E. da Silva, R. Mendes, and A. S. Ribeiro, "Trajectory optimization for underwater vehicles in time-varying ocean flows," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, 2018.

[25] D. Gomis, A. Pascual, and M. A. Pedder, "Errors in dynamical fields inferred from oceanographic cruise data: Part II. The impact of the lack of synopticity," *Journal of Marine Systems*, vol. 56, no. 3-4, pp. 334–351, 2005.

[26] J. Carlton, *Marine propellers and propulsion*. Butterworth-Heinemann, 2018.

[27] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[28] S. Li, J. K. Gupta, P. Morales, R. Allen, and M. J. Kochenderfer, "Deep implicit coordination graphs for multi-agent reinforcement learning," *arXiv preprint arXiv:2006.11438*, 2020.

[29] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[30] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*, 2016.

[31] J. Kim, D. Jang, and H. J. Kim, "Distributed multi-agent target search and tracking with gaussian process and reinforcement learning," *International Journal of Control, Automation and Systems*, vol. 21, no. 9, p. 3057–3067, 2023.

[32] D. T. Nguyen, A. Kumar, and H. C. Lau, "Credit assignment for collective multiagent RL with global rewards," *Advances in neural information processing systems*, 2018.

[33] F. Pratissoli, M. Mantovani, A. Prorok, and L. Sabattini, "Distributed coverage control for time-varying spatial processes," *IEEE Transactions on Robotics*, 2025.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:00:10 UTC from IEEE Xplore. Restrictions apply.