Informative and Communication-Efficient Multi-Agent Path Planning for Pollution Plume Monitoring

Mohamed Sami Assenine, Walid Bechkit, Hervé Rivano

Informative and Communication-Efficient Multi-Agent Path Planning for Pollution Plume Monitoring

Mohamed Sami Assenine
INSA Lyon, Inria
CITI, UR3720
69621 Villeurbanne, France
mohamed-sami.assenine@insa-lyon.fr

Walid Bechkit
INSA Lyon, Inria
CITI, UR3720
69621 Villeurbanne, France
walid.bechkit@insa-lyon.fr

Hervé Rivano
INSA Lyon, Inria
CITI, UR3720
69621 Villeurbanne, France
herve.rivano@insa-lyon.fr

***Abstract*—In this paper, we propose an efficient framework for monitoring pollution plumes using sensor-equipped drones. Our approach leverages the power of Reinforcement Learning and Mutual Information to strategically plan drone paths in order to maximize the informativeness of the data collected while minimizing communication costs. We propose a multi-agent Independent Q-Learning scheme, where drones act independently but share a global team reward. The reward is calculated based on both the reduction in plume estimation uncertainty and the communication costs. The proposed framework is adaptable to various problem instances, making it suitable for monitoring diverse physical phenomena. We conduct extensive simulations showing the effectiveness of our approach in achieving high-quality plume monitoring, with an error in variance estimation ranging from 3% to 5% when compared with ground-truth value. Results also show that our solution offers good compromise between plume estimation and communication costs. This framework outperforms the random-walk approach up to 32.88% and genetic-based heuristics up to 4.2% in terms of total rewards under the proposed scenarios. The proposed framework is advantageous because it excels not only in providing a good solution but also in inferring it in a reasonable time especially compared to a solution provided by genetic-based heuristics.**

***Index Terms*—Pollution Plume Monitoring, Mutual Information, Communication Costs, Multi-agent Informative Path Planning, Reinforcement Learning, Gaussian Process.**

I. Introduction

Growing concerns about environmental pollution and its adverse effects on human health and ecosystems have spurred interest in advanced monitoring systems. Monitoring accidental pollution plumes, with their complex dispersion in the atmosphere, requires accurate and efficient methods. Traditional monitoring approaches, like ground-based sensors and satellite observations, often face limitations in coverage, mobility, and real-time data acquisition, especially for dynamic phenomena. In response, autonomous drones offer a promising solution to address these challenges and enhance pollution monitoring efficiency.

In pollution plume monitoring using limited battery autonomous drones, the primary challenge lies in optimizing fleet trajectories to efficiently gather useful data for plume monitoring. Our framework addresses this challenge by aiming to maximize data informativeness for accurate plume estimation while minimizing communication costs for operational efficiency. To achieve this, we integrate Reinforcement Learning (RL) [1], [2] and Mutual Information (MI) [3], leveraging the capabilities of environmental sensor-equipped drone fleets for effective pollution monitoring.

In the emerging context of autonomous drones, RL is increasingly used to enable them to learn and make decisions in real-time. This approach is particularly important for trajectory design in emergency situations, where human intervention can be perilous as in [4]. In this study, we adopt the Independent Q-Learning (IQL) [5] scheme, which allows each drone to act independently and learn its policy based on shared rewards reflecting each drone's contribution to reducing uncertainty in plume estimates. With IQL, a drone can effectively explore the environment and adapt its actions to maximize the informativeness of the collected data while minimizing communication costs with a base station.

To validate our approach, we assume the pollution plume follows a Gaussian Process (GP), inspired by studies in diverse domains such as seawater salinity [6], water temperature [7], and pollution [8], [9]. GPs provide reliable estimations when configured with an appropriate kernel [10]. This modeling choice is grounded in their ability to offer uncertainty information, which is crucial in uncertain environmental contexts with sparse measurements. It is essential to note that the GP assumption underlies our hypothesis, playing a fundamental role in both pollution plume and communication modeling.

We evaluated our pollution plume monitoring framework, comparing various RL models and justifying the selection of Rainbow DQN [11]. The performance of our solution was compared to a random-walk approach and a genetic-based heuristic, considering their relevance in path planning, the quality of pollutant estimates, and communication costs. Furthermore, we assessed computational efficiency and scalability with varying drone numbers. The results showed that our framework outperforms baseline approaches, achieving higher

This work has been funded and supported by the French National Research Agency (ANR) under the project ANR-21-CE25-0003 (DRON-MAP).

rewards, accurate pollutant estimation, and reduced communication costs. Moreover, our framework exhibits scalability and efficiency in terms of solution inference time.

In this paper, we make the following key contributions:

- We propose a novel RL-based framework for pollution plume monitoring using a DQN-Rainbow model to optimize data informativeness and communication costs.
- We propose a reward function based on adequate modeling of the data informativeness, along with GP-modeled communication costs.
- We conduct several tests on two simulated pollution plume scenarios, followed by a comparative analysis with two baseline approaches to demonstrate the effectiveness of our approach.

This paper is organized as follows: Section II covers related works on MI based path planning and communication costs modeling. Section III formulates the pollution plume monitoring problem, defining objectives and requirements. Section IV presents fundamental principles of RL and recent variants, as well as the use of MI in path planning and communication link quality modeling. In Section V, we detail our proposed solution. Section VI shows some implementation details and the experimental setup. In Section VII, we assess the performance of our framework through various tests on two plume scenarios and comparison with baseline approaches. Finally, Section VIII summarizes our contributions.

II. Related Work

In spatial phenomena monitoring using mobile vehicles carrying sensors with communications capabilities, selecting optimal paths is crucial. We focus on two aspects: the informativeness of sensor locations and their ability to communicate their data efficiently. We review some relevant works on path planning using Mutual Information (MI) criteria and communication cost modeling with Gaussian Processes (GPs).

A. Mutual Information Based Path Planning

The authors of [3] introduced the use of MI as a criterion for selecting informative points in wireless sensor placement. They proposed a heuristic and a greedy approach approximating the NP-hard informative sensor placement problem with constant approximation ratios. In [12], the authors addressed a sensor deployment problem to minimize maintenance costs while ensuring tolerable detection quality and full connectivity, considering thermal degradation and battery depletion. They used the sensing quality metric based on MI to evaluate spatial phenomena. The work [13] focused on optimal wind sensor placement over a large urban water reservoir. They determined the most informative locations for accurate wind prediction in real-time, using either entropy or MI as the sensor placement criterion. The entropy criterion found informative, spaced locations, while the MI criterion reduced uncertainty by maximizing MI between selected locations and the rest.

Previous papers aimed to optimally place sensors to maximize data informativeness. From this, the Informative Path Planning (IPP) problem emerged, which involves determining waypoints and efficient paths within a budget constraint. In this paragraph, we discuss two proposed solutions for IPP. In [6], the authors presented a path planning method for autonomous underwater vehicles to maximize sea salinity information using GP modeling and the MI criterion. They employed a recursive search algorithm with coarse discretization for glider waypoints, which might be limited for complex configurations. In [14], the authors tackled the problem of planning efficient paths for multiple mobile robots to collect Wi-Fi signal strength data indoors using a RL-DQN [15] framework and the MI criterion. They proposed reward sharing methods to enable independent learning among cooperative agents.

While offline spatial mapping worked well in previous works due to the static nature of the phenomena being monitored, dynamic phenomena such as pollution plumes, as in our case, require online data collection. In this work, we introduce a new multi-agent heuristic based on independent learners using Rainbow's DQN approach [11] with reward sharing. In parallel, we update the GP regression model governing the MI criterion in a continuous online fashion to capture in real-time our pollution plumes.

B. Communication Costs Modeling with GP

Krause et al. [16] proposed an approach for sensor placement that combines sensing quality using MI [3] and communication costs. They used a parametric model for link reception rate based on the probabilistic framework of GPs, assuming no acknowledgement and no temporal correlation of lossy links. Based on the expected number of transmissions [17] between two sensors as a cost metric, the GP is used to model and predict communication costs.

In our study, we use a GP regression model to learn and predict the expected number of packets to be transmitted between two nodes to ensure the arrival of one of them. We continually refine this GP model with new packet sending experiences. This flexible approach will enable better adaptation to changing communication conditions than fixed communication models.

III. Problem Statement

In this work, we consider a fleet of drones equipped with environmental sensors, computing capabilities, and wireless communication systems. When alerted of an accidental pollution plume, the drones are deployed to collaboratively monitor the plume in real-time and generate accurate maps. We consider rotary-wing drones capable of hovering, which considerably improves measurement quality. Our approach involves iterative improvements of plume estimation thanks to an informative path planning solution based on the informativeness and the communication costs.

We assume that the monitored phenomenon follows a GP, and our goal is to quickly estimate the parameters of the GP distribution while minimizing communication costs between drones. Given the sparsity of data measured by drones, two issues arise. The first concerns how to better estimate the parameters of the overall distribution using these

sparse measurements, and the second involves finding the optimal positions of the drones at each iteration to achieve our objectives.

In our scenario, we consider a centralized system through a ground base station, guiding drones to their new positions at each iteration. Before relocation, the station must receive pollution measurements from the drones to update the GP regression and determine the next most informative positions. Using a multi-hop approach, potentially passing through other nearby drones, these drones and the base station communicate. The communication cost between two nodes is the expected number of transmissions, including the number of retransmissions. We thus define the cost of a communication path from a drone to the base station as the number of end-to-end transmissions. We seek to identify the optimal packet-sending paths for each drone with the base station at each step. Optimizing communications enhances system reliability and accelerates packet exchanges.

The spatial domain in which the plume is studied is assumed to be discretized into a regular grid / lattice $G = (V, E)$ where V is the set of vertices representing the points of interest and E is the set of edges between these points. For each subset $\mathcal{A} \subseteq V$, we note $f(\mathcal{A})$ the measurement's quality or the informativeness of the measurements of pollution taken at the $\mathcal{A}$ locations defined by MI as shown in IV-A.

A drone i traverses a path $\mathcal{P}_i = (\mathbf{v}_1^i, \mathbf{v}_2^i, \cdots, \mathbf{v}_l^i)$ means that $\mathcal{P}_i$ is a sequence of l locations starting with node $\mathbf{v}_1^i$ and arriving at $\mathbf{v}_l^i$. The energy cost $C(\mathcal{P}_i)$ of taking the path $\mathcal{P}_i$ is the sum of the moving expenses between each two successive locations of l and the sensing costs there.

For a set of N paths $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_N\}$, one for each drone, $f(\mathcal{P}) = f(\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_N)$ gives the quality of the measurements gathered across all this paths. Finding a set $\mathcal{P}^*$ of N paths, with known starting locations $\mathbf{v}_s = (\mathbf{v}_s^1, \cdots, \mathbf{v}_s^N)$, that maximized the function f is the solution to our problem. This is done while considering these two requirements:

- **Budget constraints.** The cost of a path taken by the i-th drone $C(\mathcal{P}_i)$ should not exceed a bounded cost $\mathbf{b}^i$ which represents the maximum budget of the i-th drone.
- **Communication costs.** Minimize at each step the number of data packets a drone should send in order for one of them to reach the base station (communication cost).

In other terms, solving our problem involves finding $\mathcal{P}^*$ with consideration of budget constraints and communication costs aspect.

IV. Background

In view of the issues raised in the previous section, we propose examining MI-based path planning and wireless communication link modeling. Additionally, we introduce RL concepts, including DQN and its variant, Rainbow.

A. GP-based Mutual Information for Path Planning

IPP is a subdomain of path planning problems where path relevance depends on data informativeness. To quantify this, we establish uncertainty by assuming that the phenomenon monitored can be modeled by a GP [18]. This means that the data $\mathbf{y}_V$ at all locations V of a grid follow a multivariate joint Gaussian distribution, whether note:

$$\mathbf{y}_V \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_{\mathbf{v}_1}) \\ \vdots \\ m(\mathbf{x}_{\mathbf{v}_n}) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_{\mathbf{v}_1}, \mathbf{x}_{\mathbf{v}_1}) & \cdots & k(\mathbf{x}_{\mathbf{v}_1}, \mathbf{x}_{\mathbf{v}_n}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{\mathbf{v}_n}, \mathbf{x}_{\mathbf{v}_1}) & \cdots & k(\mathbf{x}_{\mathbf{v}_n}, \mathbf{x}_{\mathbf{v}_n}) \end{bmatrix}\right), \quad (1)$$

where for each node $\mathbf{v} \in V$ of coordinates $\mathbf{x}_\mathbf{v}$, the corresponding data (pollutant concentrations in our case) is denoted by $\mathbf{y}_\mathbf{v} \in \mathbf{y}_V$. $m(\cdot)$ denotes a mean function, $k(\cdot, \cdot)$ is a kernel and $n = |V|$. We denote the multivariate Gaussian distribution by $\mathcal{N}(m(X_V), \sum_V)$, where X_V is a $n \times 2$ 2D coordinate matrix of V vertices and $\sum_V$ is a $n \times n$ covariance matrix defined by the kernel function [14]. In the literature, a widely used kernel is the Radial Basis Function (RBF) kernel, where:

$$k(\mathbf{x}_{\mathbf{v}_p}, \mathbf{x}_{\mathbf{v}_q}) = \sigma_f^2 \cdot exp\left(-\frac{||\mathbf{x}_{\mathbf{v}_p} - \mathbf{x}_{\mathbf{v}_q}||^2}{2l^2}\right), \quad (2)$$

such that σ_f^2 is the common variance of $\mathbf{y}_V$ and l is the length scale. The variance defines the upper limit of covariance between variables, while the length scale attenuates the correlation between points $\mathbf{v}_p$ and $\mathbf{v}_q$ based on the Euclidean distance.

Then, the continuous differential entropy of $\mathbf{y}_V$ is given by:

$$H(\mathbf{y}_V) = \frac{1}{2} \ln |\sum\nolimits_V| + \frac{n}{2}(1 + \ln(2\pi)). \quad (3)$$

Given the collected measurements $\mathbf{y}_S$ at points X_S, the posterior distribution of $\mathbf{y}_V$ given $\mathbf{y}_S$ is $\mathcal{N}(\mu', \sum')$ where [18]:

$$\begin{aligned} \mu' = m(X_V) + K(X_V, X_S) \cdot (K(X_S, X_S) + \sigma_n^2 I)^{-1} \\ \cdot (\mathbf{y}_S - m(X_S)), \end{aligned} \quad (4)$$

$$\begin{aligned} \sum{}' = K(X_V, X_V) - K(X_V, X_S)[K(X_S, X_S) + \sigma_n^2 \cdot I]^{-1} \\ \cdot K(X_S, X_V), \end{aligned} \quad (5)$$

here σ_n^2 is the variance of the Gaussian noise, and $K(A, B)$ is the kernel matrix from $k(\cdot, \cdot)$ with pairwise entries in sets A and B. The conditional differential entropy of $\mathbf{y}_V$ given $\mathbf{y}_S$ is then expressed as:

$$H(\mathbf{y}_V|\mathbf{y}_S) = \frac{1}{2} \ln |\sum{}'| + \frac{n}{2}(1 + \ln(2\pi)). \quad (6)$$

The latter conditional entropy is used to assess the uncertainty in prediction of $\mathbf{y}_V$ given $\mathbf{y}_S$. To quantify the informativeness of a set of paths $\mathcal{P}$, we use the criterion of MI [3]:

$$\begin{aligned} f(\mathcal{P}) &= MI(\mathbf{y}_V; \mathbf{y}_S) \\ &= H(\mathbf{y}_V) - H(\mathbf{y}_V|\mathbf{y}_S). \end{aligned} \quad (7)$$

The idea behind MI lies in rewarding locations that notably decrease uncertainty, prioritizing them over potential locations where no measurements have been taken [19], [20]. In addition to this, Krause et al. [16] show that MI leads to intuitive placements with prediction accuracy superior to alternative approaches.

B. Wireless Communication Links Quality Modeling

We need to ensure reliable communication links and minimize unnecessary retransmissions, which have a twofold effect: on drone energy consumption and on expediting message delivery. Roughly, if the probability for a successful transmission between two drone locations s and t is $\theta_{s,t}$, then the expected number of transmissions is $1/\theta_{s,t}$. But often, we have to estimate and predict these probabilities of successful transmissions, so in general, we only obtain a distribution $P(\theta_{s,t})$ with density $p(\theta_{s,t})$ instead of one fixed value for $\theta_{s,t}$. Analytically, the expected number of transmissions is given as [16]:

$$c_{s,t} = \int_{\theta} \frac{1}{\theta_{s,t}} p(\theta_{s,t})\, d\theta_{s,t}. \tag{8}$$

Based on this formula, one can compute the expected number of transmissions, including the retransmissions, for each pair of locations $(s,t) \in V^2$. What we need to do at present is estimate $\theta_{s,t}$ and find its density function $p(\theta_{s,t})$. In [16], the authors propose to use a GP to model the transmission success probabilities $\theta_{s,t}$ for each pair $(s,t) \in V^2$. The 3-dimensional GP regression model is fed with inputs of type: $\{(s, ||\mathbf{x}_s - \mathbf{x}_t||, \theta^e_{s,t}) \mid (s,t) \in V^2\}$. Denoted as $\theta^e_{s,t}$, this represents the success rate of packet transmission from the drone at point s to the drone at point t during epoch e.

Once the GP model is sufficiently fed with data, we can run a regression on it, which will allow to generate an estimate of the probability of successful transmission $\mu_{\theta_{s,t}}$ and its variance $\sigma^2_{\theta_{s,t}}$ for each pair $(s,t) \in V^2$. Assuming the predictive distribution for $p(\theta_{s,t})$ is normal with mean $\mu_{\theta_{s,t}}$ and variance $\sigma^2_{\theta_{s,t}}$, we can now compute $c_{s,t}$ based on equation 8 [16].

We assume the theoretical drone's transmission range is infinite, so the expected number of transmissions exponentially increases as a drone moves farther away.

C. Deep Q-Networks and Rainbow Variant

RL is a machine learning approach where an agent learns to perform tasks by interacting with an uncertain environment. The environment evaluates the actions allowing the agent to discover the most rewarding ones [1]. RL problems are typically modeled using Markovian Decision Processes (MDP) for decision making in sequential, stochastic environments. The agent observes states, takes actions, and aims to maximize rewards, considering their future accumulation [1].

RL algorithms are categorized as policy-based and value-based approaches [2]. One of the most widely used algorithms is Q-Learning [21], which uses an iterative approach to estimate Q-values. These values estimate future rewards for choosing specific actions and following the same policy in the future. However, conventional Q-Learning becomes impractical for complex problems due to the exponential increase in storing Q-values for each state-action pair in the tabular method.

A recent implementation of Q-Learning utilizes Deep Neural Networks (DNNs) to approximate Q-values, known as Deep Q-Network (DQN) [15]. Mnih et al. proposed a set of effective techniques for efficient DQN implementation in their article [22]. These techniques include:

- **Trade-off exploration-exploitation.** DQN uses the ϵ-greedy policy, where it selects the action with the highest Q-value with a probability of $1-\epsilon$ and chooses randomly with a probability of ϵ. During training, the value of ϵ decreases progressively.
- **Double DQN.** This involves the use of two DNNs, the classic Q-Network and a new one called "Target". This technique ensures greater stability and significantly better performance.
- **Experience replay.** This corresponds to building a data set of episodic experiments on which the learning agent is then trained by sampling mini-batches of experiments.

This latest paper heralded the arrival of highly effective methods such as the Rainbow variant of DQN [11]. This article examined several extensions to the DQN algorithm (Double DQN [23], Prioritized DQN [24], Dueling DQN [25], Distributional DQN [26], Noisy DQN [27]) and empirically studied their combination. Their experiments have shown that the combination (Rainbow) outperforms state-of-the-art performance, both in terms of data efficiency and final performance.

V. Our Solution

Due to the NP-hardness of Multi-agent IPP (MIPP) problems [14], heuristic solutions are essential for efficient computation. Traditional heuristic approaches like genetic algorithms need a new execution whenever a problem parameter changes. In contrast, RL-based solutions excel in adapting to various problem instances with different parameters within a single training round. In what follows, we outline the step-by-step execution of our framework, showing the blocks that constitute the final solution.

A. Framework Overview

Our iterative framework (Fig. 1) starts with environmental measurements taken while drones hover. A GP regression uses the new and past measurements to update the plume's state $\mu = (\mu_1, \mu_2, \cdots, \mu_n)$ and variance estimates σ^2_f. Using this information and drone positions, the RL model proposes new neighboring positions to maximize plume knowledge. The procedure is repeated until the drones' battery capacity is exhausted. To initialize the parameters of the GP model, a random exploratory flight of a few steps is needed to collect measurements before operational deployment.

B. Global RL Model Scheme

Our RL solution adopts a multi-agent Independent Q-Learning (IQL) scheme [5], [28]. This allows each agent to learn and choose actions independently while, in our case, partially sharing input states. Furthermore, the environment assigns a common reward to the entire team, which considers the communication costs and the reduction in plume uncertainty thanks to the new measurements taken by the drones.

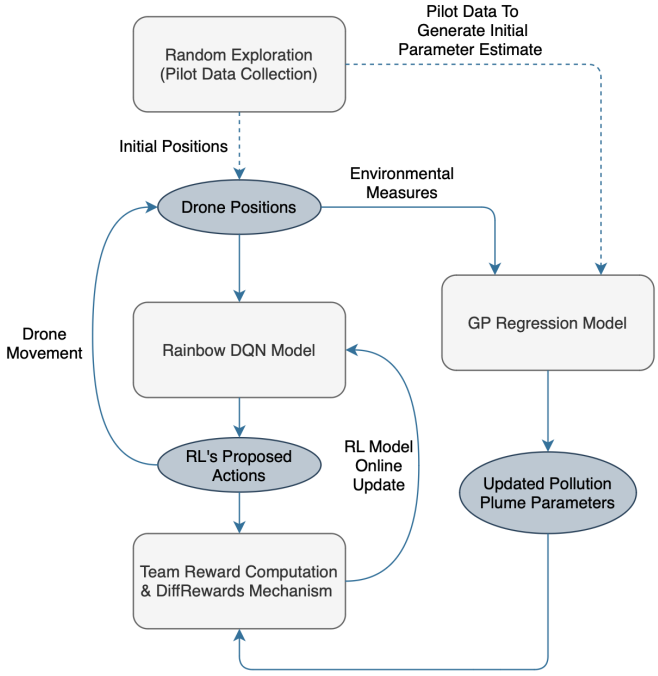


Fig. 1: Overall diagram of our proposed framework

Our IQL scheme comprises a number of RL models equal to the number of agents involved. An RL model, in our case, is made up of several fully-connected neural networks of the DQN Rainbow type (see subsection IV-C). We present in Table I the configuration parameters of our RL-based solution. The last layer of the target Q-Network produces distribution diagrams of possible actions. To determine the best action (next jump node), we rely on the ϵ-greedy policy.

C. States Representation and Action Selection

In this subsection, we define the input state of a learning agent (a drone) and the set of possible actions at each step:

- **Input state of an agent.** The agent's state at a given step is a quadruple composed of the actual estimated map, the drone's current 2D coordinates, its remaining budget and the locations of the other drones.
- **Set of actions.** The agent can choose between jumping to one of the neighboring vertices or staying in its current position. To address drone battery heterogeneity, a dummy action is included. A flight episode ends when all agents have exhausted their budget.

D. Defining and Sharing Rewards

After the execution of all agents' actions, the environment generates a collective team reward. This reward $R_T(\mathbf{v}_t)$ at time t is the aggregation of MI reward $R_{\text{MI}}(\mathbf{v}_t)$ and the network communication cost reward $R_{\text{CC}}(\mathbf{v}_t)$ of the topology formed by the drones locations $\mathbf{v}_t$ with the base station.

The MI reward $R_{\text{MI}}(\mathbf{v}_t)$ is corresponding to the difference between the informative function (see IV-A) of the path set at actual step ($\mathcal{P}_t$) and this set at the previous step ($\mathcal{P}_{t-1}$) as shown by the formula 9. This reward can be expressed as the decrease in plume uncertainty (entropy) after the new measurements have been collected at step t (see formula 10).

$$R_{\text{MI}}(\mathbf{v}_t) = f(\mathcal{P}_t) - f(\mathcal{P}_{t-1}), \tag{9}$$

$$R_{\text{MI}}(\mathbf{v}_t) = H(\mathbf{y}_V|\mathbf{y}_{\mathcal{P}_{t-1}}) - H(\mathbf{y}_V|\mathbf{y}_{\mathcal{P}_t}). \tag{10}$$

The MI reward is distributed among all agents using a reward sharing mechanism, also known as credit assignment [29]. We adopt the Difference Rewards approach [30], which shares the reward based on each agent's marginal contribution to the team reward. The partial MI reward of an agent i at time t can be given as follows:

$$R_{\text{MI}}(\mathbf{v}_t^i) = H(\mathbf{y}_V|\mathbf{y}_{\mathcal{P}_t^{-i}}) - H(\mathbf{y}_V|\mathbf{y}_{\mathcal{P}_t}), \tag{11}$$

where $\mathcal{P}_t^{-i}$ denotes the collection of paths taken by all agents excluding the path of agent i in order to evaluate its contribution. Given that the sum of these individual rewards may not necessarily be equal to the total MI reward ($\sum_{i=1}^{N} R_{\text{MI}}(\mathbf{v}_t^i) \neq R_{\text{MI}}(\mathbf{v}_t)$), normalization is required:

$$R_{\text{MI}}(\mathbf{v}_t^i) \leftarrow R_{\text{MI}}(\mathbf{v}_t^i) \times \frac{R_{\text{MI}}(\mathbf{v}_t)}{\sum_{j=1}^{N} R_{\text{MI}}(\mathbf{v}_t^j)}. \tag{12}$$

To compute the reward relative to the network communication costs of a drone i with the base station at time t ($R_{\text{CC}}(\mathbf{v}_t^i)$), we first need to find the best path between these two. This may be a direct link, or a multi-hop link passing through one or more other drones to transfer packets. The optimum paths between the drones and the base station are found using Dijkstra's algorithm by considering the inverse of the logarithms of the probabilities of successful packet transmission $-\log(1/c_{v_1,v_2})$ between each two points v_1 and v_2. So, if the most interesting path between the drone i and the base station ($\mathbf{v}_{bs}$) is the following ($\mathbf{v}_t^i = v_1, v_2, \cdots, v_k = \mathbf{v}_{bs}$) then the communication cost of this drone is written as follows:

$$\text{CC}(\mathbf{v}_t^i) = \sum_{j=1}^{j=k-1} \log(c_{v_j,v_{j+1}}). \tag{13}$$

The communication cost of the entire topology and the corresponding reward are given by:

$$\text{CC}(\mathbf{v}_t) = \frac{1}{N} \cdot \sum_{i=1}^{i=N} \text{CC}(\mathbf{v}_t^i), \tag{14}$$

$$R_{\text{CC}}(\mathbf{v}_t) = \frac{1}{\text{CC}(\mathbf{v}_t)}. \tag{15}$$

We can now give the aggregated partial reward of a drone i at step t with a weighting $(\alpha, 1-\alpha)$ between MI and communication cost rewards, such that $\alpha \leq 1$:

$$R_T(\mathbf{v}_t^i) = \alpha \cdot \tanh(R_{\text{MI}}(\mathbf{v}_t^i)) + (1-\alpha) \cdot R_{\text{CC}}(\mathbf{v}_t^i). \tag{16}$$

With $R_{\text{CC}}(\mathbf{v}_t^i) = \frac{1}{\text{CC}(\mathbf{v}_t^i)}$. The $\tanh$ function was chosen to normalize the MI reward values so that they are in the same interval as those of the communication costs. Finally, we define the total team reward as follows:

$$R_T(\mathbf{v}_t) = \sum_{i=1}^{i=N} R_T(\mathbf{v}_t^i). \tag{17}$$

From formulas 15, 16 and 17 we can easily deduce that $R_T(\mathbf{v}_t) \leq N$ because $R_T(\mathbf{v}_t^i) \leq 1$. This confirms that the total reward of an episode is less than or equal to δN. Where δ is the maximum number of steps per episode. Therefore, through the construction of the reward function, we establish an upper bound on the maximum total reward, and consequently maximize the performance of an optimal solution.

E. Proposed Framework Algorithm

Algorithm 1 presents the IQL scheme using DQN Rainbow models, along with reward calculation that combines MI and communication costs. The Difference Rewards mechanism is used to share team rewards among the agents of the framework.

Algorithm 1: Multi-agent IQL scheme based on DQN Rainbow models

Data: $\langle G, N, \mathbf{v}_s, \mathbf{b}, R_T(\cdot), K \rangle$
Result: Neural Network parameters θ.
initialization of the prioritized replay buffers $\mathcal{M} = \{\mathcal{M}^i \,|\, i \in \{1, \cdots, N\}\}$ (see reference [24])
initialize $\theta = \{\theta^i \,|\, i \in \{1, \cdots, N\}\}$ randomly and set $\theta^- = \theta$
initialization of the maximum size of an episode δ
for *episode* $e = 1, 2, ...$ **do**
  get initial global state s_0
  for *step* $t = 1, 2, ..., \delta$ **do**
    for *agent* $i = 1, 2, ..., N$ **do**
      if the agent's budget is not exhausted **then**
        - get actual position of i-th agent $\mathbf{v}_t^i$
        - with probability ϵ, choose a random action else choose the action with the biggest Q-value returned by the Rainbow model
    execute the joint action $\mathbf{a}_t = (a_t^1, a_t^2, ..., a_t^N)$
    calculate the reward of each agent $[R_T(\mathbf{v}_t^1), ..., R_T(\mathbf{v}_t^N)]$ (formula 16)
    store each transition tuple $(s_t, a_t^i, R_T(\mathbf{v}_t^i), s_{t+1})$ in $\mathcal{M}^i$
    for each agent i sample a batch of $(s_j, a_j^i, R_T(\mathbf{v}_t^i), s_{j+1})$ from $\mathcal{M}^i$ and update θ^i by minimizing the TD Loss
  update $\theta^- = \theta$ with some period K
return θ

VI. Validation and Experimental Setup

To validate the proposed solution, we develop a simulator with three main components. One simulates pollution plumes to model the unknown ground truth, another simulates packet transmission, and a third implements our RL-based solution simulating drone movement. In the following paragraphs, we detail the implementation of this simulator and the experimental setup.

A. Pollution Plume Simulation

In our tests, we use a 15×9 regular grid to represent a rectangular geographical area. The pollutant plume dispersion is simulated using a basic Gaussian fluid mechanics model [31]. Assuming a wind to be along the x-axis and the pollutant source is located at coordinates $(0, 0, h_s)$, this model gives the pollutant concentration at any point (x, y, z) in free space using the following formulas [32]:

$$C(x,y,z) = \frac{Q}{2\pi u \sigma_y \sigma_z} e^{\left(\frac{-y^2}{2\sigma_y^2}\right)} \left[e^{\left(\frac{-(z-H_e)^2}{2\sigma_z^2}\right)} + e^{\left(\frac{-(z+H_e)^2}{2\sigma_z^2}\right)} \right]. \tag{18}$$

Here $H_e = h_s + \Delta h(\cdot)$ is the effective release height where $\Delta h(\cdot)$ is the plume elevation calculated using the Briggs formulas [33], which depend on wind speed, volumetric flow rate, gravity constant, ambient temperature, pollutant temperature, and the position on the x-axis.

Table I summarizes the parameters of the formula 18 and the values we used to simulate pollution plumes. We relied on the realistic values provided in [32], [33]. Fig. 2 shows an example of a simulated plume with three identical pollutant sources. To simulate a real plume over time, several instances are generated where the average concentrations are computed based on the pollution map derived from formula 18. These instances are generated using a random function based on a multivariate Gaussian distribution, incorporating a defined variance and a length scale that establishes the covariance matrix.

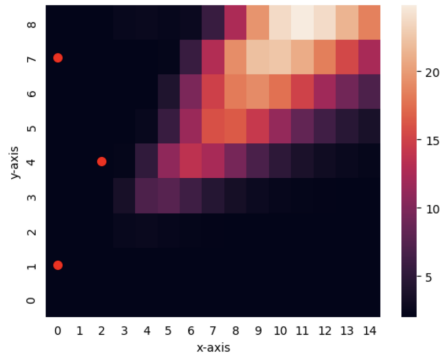


Fig. 2: Heat map of pollutant concentrations (in mg/m^3) at a height of $30\,m$ from the sources represented by the red dots with a wind at 28° from the x-axis.

B. Data Packet Transmission Simulation

In subsection IV-B, we explained how to model communication link qualities between drones over an area using a GP. To achieve this, we used a GP regression fed with transmission success rates in each epoch. This led us to implement a data packet transmission simulator. Based on the distance between transmitter and receiver, it generates a scenario for sending several packets and calculates the Packet Error Rate (PER). The PER is mainly determined by the signal strength at the receiver. The received power P_r depends on the transmit power (P_t) and its attenuation throughout the communication channel.

The attenuation $\mathcal{A}(d)$ over a distance d is defined as the ratio between P_t and $P_r(d)$. We denote by $P_r(d)$ the signal

power received at a distance d from the transmitter. Here's the attenuation formula (P_t and $P_r(d)$ are expressed in dBm, attenuation is in dB) [34]:

$$\mathcal{A}(d) = P_t - P_r(d). \tag{19}$$

The log-normal path loss model [35] is commonly used for attenuation, expressing it as a function of three factors:

$$\mathcal{A}(d) = \mathcal{A}_0 + \mathcal{A}_1(d) + \mathcal{A}_2, \tag{20}$$

here, $\mathcal{A}_0$ represents the path loss at the reference distance d_0. $\mathcal{A}_1(d)$ is the log-distance path loss as a function of distance d and reference distance d_0, while $\mathcal{A}_2$ is a random variable reflecting the attenuation induced by fading. Here is the development of each of the three factors [34]:

$$\begin{cases} \mathcal{A}_0 = 10 \cdot \log(d_0) + 20 \cdot \log(f_0) - 27.55, \\ \mathcal{A}_1(d) = \alpha_0 \cdot 10 \cdot \log(\frac{d}{d_0}), \\ \mathcal{A}_2 \sim \mathcal{N}(0, \sigma^2), \end{cases} \tag{21}$$

where f_0 is the frequency (in MHz), α_0 denotes the path loss exponent, and $\mathcal{N}(0, \sigma^2)$ represents a normal random variable.

Since the power of the transmitters is known, we can use formulas 19, 20 and 21 to calculate the power received by a receiver $P_r(d)$. Then using the following formula, one can derive the PER:

$$PER_r(d) = \frac{1}{2} - \frac{1}{\sqrt{(}2\pi)} \int_0^{\frac{P_r(d)-a}{b}} e^{-\frac{x^2}{2}} dx, \tag{22}$$

where a is the sensitivity level and b is the quarter of the transition interval length. The latter is the power attenuation interval that passes the PER from $\sim 1\%$ to $\sim 100\%$ and vice versa [35].

Calculating the PER between a sender s and a receiver r enables to use a Bernoulli random variable with a probability of $(1 - PER_r(||\mathbf{x}_s - \mathbf{x}_t||))$ to simulate multiple packet transmissions between s and r, determining whether the packets will be received or not. Table I summarizes all the parameter values used for the packet transmission simulations.

General parameters	
Grid size of the target area	15×9
Distance between two adjacent nodes	$50\,m$
Reward aggregation parameter (α)	0.7
Pollution plume simulation parameters	
Pollutant source height (h_s)	$30\,m$
Emission rate at source (Q)	$1.59\,g/s$
Wind speed (u)	$5\,m/s$
Pollution plume elevation ($\Delta h(x)$)	$2.126 \times 10^{-4} \cdot x^{2/3}$
Horizontal dispersion coefficient (σ_y)	$1.36 \cdot \|x\|^{0.82}$
Vertical dispersion coefficient (σ_z)	$0.275 \cdot \|x\|^{0.69}$
Packet transmission simulation parameters	
Transmit power (P_t)	14 dBm
Reference distance (d_0)	$1\,m$
Frequency (f_0)	868 MHz
Path loss exponent (α_0)	3
Variance of the random variable $\mathcal{A}_2$ (σ^2)	4
Sensitivity level (a)	-90 dBm
Transition interval length (b)	6 dB
RL parameters	
Size of an episode (δ)	50 steps
Learning rate	5×10^{-4}
Discount factor (γ)	0.97
Train batch size	32
N-step for Q-learning	2 steps
Atom number of the distributional DQN	51
Target network update period (K)	10 episodes
Neural structure of the hidden layers	3 × 128-unit layer
Hidden layers of the dueling architecture	2 × 128-unit layer

TABLE I: Summary of some general parameters

C. Experimental Setup

The proposed framework is implemented on an Apple M1 chip equipped with 64 GB of memory, utilizing Python 3.8, the TensorFlow platform, and the libraries: GPy[1] and RLlib[2].

VII. Performance Evaluation

We assess the overall performance of our framework focusing on the RL model's ability to strike a balance between exploration for plume coverage and staying close to the base station to minimize communication costs.

In the following subsections, we validate our solution. Firstly, we justify using the DQN Rainbow model over other options, and provide its training curve and parameters. Secondly, we assess its performance against two well-known approaches based on uncertainty reduction metrics, communication cost, and plume estimation errors. Finally, we demonstrate computational efficiency for different agent numbers, confirming scalability.

A. Model Selection and Training Process

In what follows, we will consider the environment as we have already defined it in the subsection VI-A. Before starting on the DQN Rainbow, we evaluate several RL models: Classical DQN [15], [22], Recurrent DQN [36], Categorical DQN (DQN-C51) [26], Double DQN [23] and Dueling DQN [25]. We trained each of them on 150000 steps with 3 drones and obtained the results summarized in table II. Note that this comparison is made with an average evaluation of the total rewards obtained from inference over 30 different episodes.

We chose Rainbow and trained it for 250000 steps to ensure convergence. The training curve is depicted in Fig. 3. The wind direction, relative to the x-axis, was regularly and randomly

[1]https://github.com/SheffieldML/GPy.

[2]https://docs.ray.io/en/latest/rllib/index.html.

RL models	Performance ratio of some RL models to Rainbow
Classical DQN	85.4%
DQN C51	94.3%
Double DQN	96.4%
Recurrent DQN	97.0%
Dueling DQN	98.8%

TABLE II: Comparison of some RL models with the Rainbow

changed during the training process. This deliberate variability served the purpose of allowing agents to adeptly track the plume and learn on several scenarios. Three agents were deployed, all starting at coordinate point $(0, 4)$ (base station location) at the beginning of each episode. For additional Rainbow configuration parameters, refer to table I.

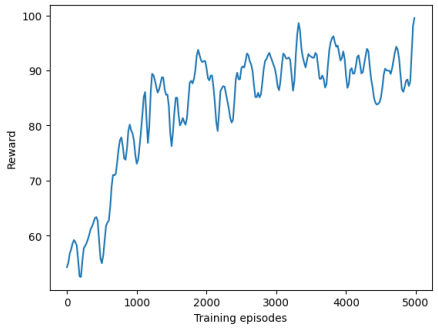
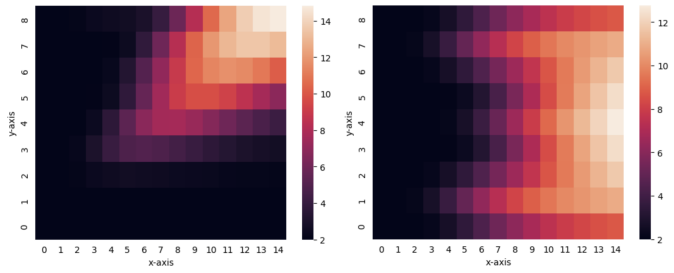


Fig. 3: Rainbow's average reward per episode during training

B. Path Planning Informativeness

After training our Rainbow DQN-based RL model (abbreviated by Rainbow in the following), we compared it with two other approaches: a random-walk (Random) and a genetic algorithm-based heuristic (GA). The random-walk selects adjacent nodes randomly for each agent at each step. The genetic algorithm involves individuals representing potential solutions, where each individual comprises a sequence of genes representing agent actions during an episode step. The initial population consists of 50 individuals, and a sub-population of the 25 best individuals is selected. Two-point crossover is performed with a probability of $p_c = 0.6$, and minimal mutations at the gene level are applied with a probability of $p_m = 0.1$. This process is iterated until convergence is achieved. These parameters offer a good compromise between run time and efficiency.

In the remaining tests, we consider two scenarios not necessarily seen during training. The first scenario involves a plume with a wind direction of 20° relative to the x-axis (Fig. 4a), and the drones initially take off from the point $(0, 4)$, the same position as the base station. In the second scenario, the plume has a wind direction along the x-axis (Fig. 4b), and the drones take off from the center of the monitored area, specifically point $(7, 4)$, which was not the case during training. This second scenario tests their adaptability to a

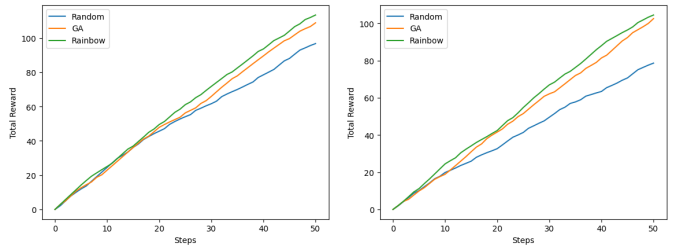


(a) Scenario 1: wind direction 20° (b) Scenario 2: wind direction 0°

Fig. 4: Two pollutant concentration heat maps (in mg/m^3) showing scenarios with different wind orientations.

different starting point and the extent to which they yield satisfactory results. We highlight that RL offers advantages such as transfer learning and the ability to quickly adapt to environmental changes. We note that the results presented in the following are averaged from path inference on 30 different episodes to have comparisons that are statistically significant.

Fig. 5 displays the total reward evolution during an episode for the three implemented methods. Initially, rewards are similar for all approaches, but as the episode progresses, the gap widens, especially between Random and the other two. GA and Rainbow perform closely, with Rainbow slightly outperforming GA by 4.2% and 1.88% in the first and second scenarios, respectively. Comparing Rainbow with Random, the performance difference ranges from 17.24% to 32.88%.



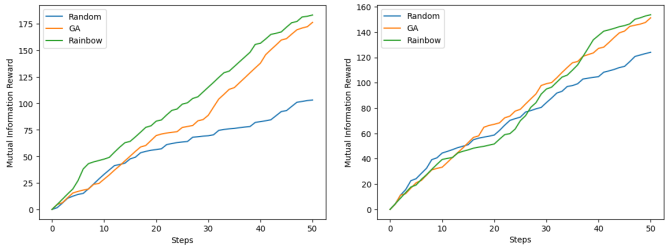
(a) Scenario 1 (b) Scenario 2

Fig. 5: Comparison of total rewards for the three approaches studied during a drone flight episode.

Considering the total reward as the aggregation of MI reward (Fig. 6) and communication cost (Fig. 7), we examine their evolution curves to understand the achieved rewards.

Looking at the first scenario (Fig. 6a), we can see that Random is far behind GA and Rainbow, who are always close together. This still allows Random to give total rewards not far behind the other two, thanks to the low average number of packets retransmitted (Fig. 7a). Since the drones start from the same position as the base station, the communication cost remains very low in the case of the random approach and rarely reaches 3 packets per step, simply because these drones remain moving randomly around the station.

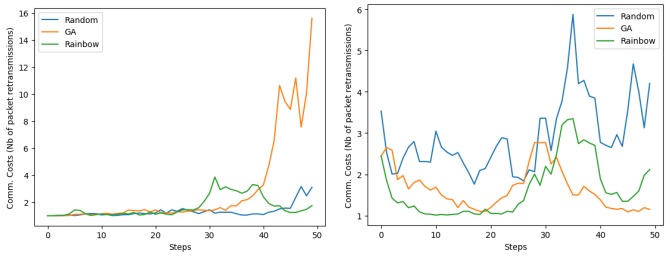
On the other hand, we have high MI reward for the GA and Rainbow (Fig. 6a) while communication costs (Fig. 7a) are a



(a) Scenario 1 (b) Scenario 2

Fig. 6: Comparison of MI rewards for the three approaches studied during a drone flight episode.

little worse than with Random. For the Rainbow, between the 30th and 40th steps, it sends the drones to explore a little far from the base station, which increases the number of packets sent to 3-4. The GA, which was fine at the start, explodes towards the end, reaching an average of 16 packets sent.



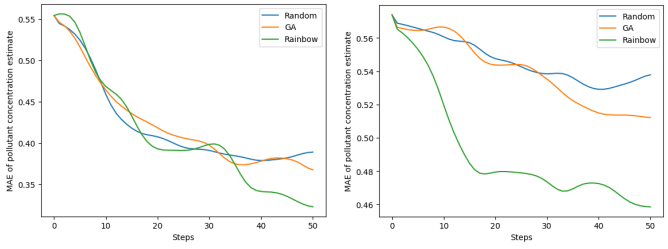
(a) Scenario 1 (b) Scenario 2

Fig. 7: Comparison of communication costs for the three approaches studied during a drone flight episode.

In the second scenario, communication costs present different outcomes (Fig. 7b). Random performs poorly as the drones start from the center of the area. Conversely, the other two methods succeed in identifying the right region where each drone's average number of sent packets doesn't exceed 4. Overall, these results indicate that Rainbow performs best in striking a balance between exploration and maintaining an optimal communication distance with the base station.

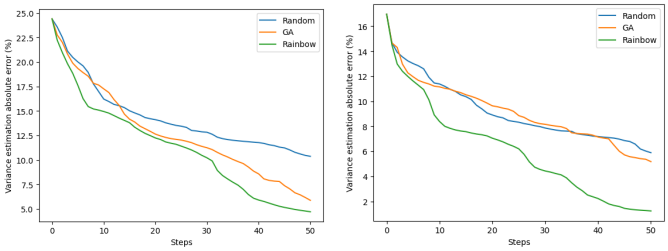
To validate Rainbow's performance, we analyze pollutant concentration and variance estimation for the two scenarios. In terms of map estimation (Fig. 8), Rainbow shows a clear advantage over baseline models, outperforming GA by $0.05mg/m^3$ in both scenarios. In scenario one, Rainbow achieves a mean absolute difference of $0.32mg/m^3$ compared to ground-truth, while other methods remain above $0.37mg/m^3$. In scenario two, Rainbow's MAE is $0.46mg/m^3$ while the GA is at $0.51mg/m^3$.

Fig. 9 shows a comparable correlation and trend between variance estimates and pollution map estimates. The Rainbow model notably reduces the initial variance error estimate from 16%-25% to 3%-5%, whereas the other two approaches yield final errors within the range of 6%-12%.



(a) Scenario 1 (b) Scenario 2

Fig. 8: Evolution of the mean absolute error in pollutant concentration estimation during a drone flight episode.



(a) Scenario 1 (b) Scenario 2

Fig. 9: Evolution of the variance error estimation during a drone flight episode.

C. Computation Efficiency

Fig. 10 shows the average execution times for inferring actions with varying numbers of drones for different approaches. The RL-based solution is highly efficient, inferring in a few hundred milliseconds at worst, while the GA solution takes over thirty minutes per instance.

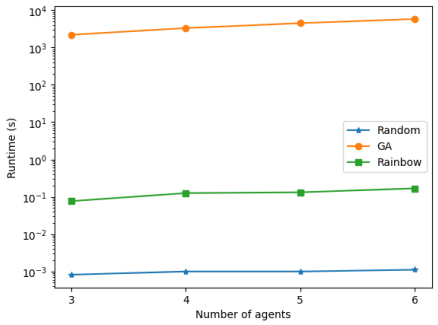


Fig. 10: Approximate path inference run times

These results confirm both the scalability of our solution, with respect to the number of learning agents, and its efficiency in terms of inference time, while outperforming baseline approaches. The time complexity of path inference is given by $O(\delta \cdot N \cdot (|\theta| + a))$, where δ is the number of steps in an episode, $|\theta|$ is the number of neural network weights, and a denotes the number of possible actions.

VIII. Conclusion

This paper presents an effective framework for pollution plume monitoring using a drone fleet equipped with environmental sensors. Leveraging RL with GP modeling proves to be a promising approach to tackle environmental monitoring challenges through autonomous systems. Our solution aims to plan drone paths which maximize data informativeness and minimize communication costs. The modeling of the plume and the probabilities of successful transmission between points are carried out using GP regression models. These models help calculate informativeness with MI and deduce drone deployment communication costs. Our simulation experiments demonstrate the proposed approach's effectiveness in high-quality plume monitoring. Compared to baseline methods, our framework outperforms random-walk and genetic-based heuristics in various scenarios, proving its efficiency in path planning. We believe that our framework, thanks to RL, is well-suited for adapting to various problem instances monitoring diverse physical phenomena modeled by GPs.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[3] C. Guestrin, A. Krause, and A. P. Singh, "Near-optimal sensor placements in gaussian processes," in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 265–272.

[4] Y. Guan, S. Zou, K. Li, W. Ni, and B. Wu, "Mappo-based cooperative uav trajectory design with long-range emergency communications in disaster areas," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2023, pp. 376–381.

[5] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth International Conference on Machine Learning (ICML)*, 1993, pp. 330–337.

[6] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 4791–4796.

[7] R. Grbić, D. Kurtagić, and D. Slišković, "Stream water temperature prediction based on gaussian process regression," *Expert Systems with Applications*, vol. 40, no. 18, pp. 7407–7414, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413004764

[8] O. Hamelijnck, T. Damoulas, K. Wang, and M. Girolami, "Multi-resolution multi-task gaussian processes," *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[9] H. Liu, C. Yang, M. Huang, D. Wang, and C. Yoo, "Modeling of subway indoor air quality using gaussian process regression," *Journal of hazardous materials*, vol. 359, pp. 266–273, 2018.

[10] K.-C. Ma, L. Liu, and G. S. Sukhatme, "Informative planning and online learning with sparse gaussian processes," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4292–4298.

[11] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[12] X. Yu, K. Ergun, L. Cherkasova, and T. Š. Rosing, "Optimizing sensor deployment and maintenance costs for large-scale environmental monitoring," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, no. 11, pp. 3918–3930, 2020.

[13] W. Du, Z. Xing, M. Li, B. He, L. H. C. Chua, and H. Miao, "Sensor placement and measurement of wind for water quality studies in urban reservoirs," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 3, pp. 1–27, 2015.

[14] Y. Wei and R. Zheng, "Multi-robot path planning for mobile sensing through deep reinforcement learning," in *IEEE INFOCOM 2021-IEEE International Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[16] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Robust sensor placements at informative and communication-efficient locations," *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, no. 4, pp. 1–33, 2011.

[17] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proceedings of the 9th annual International Conference on Mobile Computing and Networking*, 2003, pp. 134–146.

[18] C. E. Rasmussen, *Gaussian Processes in Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_4

[19] Y. Wei and R. Zheng, "Informative path planning for mobile sensing with reinforcement learning," in *IEEE INFOCOM 2020-IEEE International Conference on Computer Communications*. IEEE, 2020, pp. 864–873.

[20] Y. Wei, C. Frincu, and R. Zheng, "Informative path planning for location fingerprint collection," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1633–1644, 2019.

[21] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[23] H. Hasselt, "Double q-learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 23, 2010.

[24] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[25] 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 (ICML)*. PMLR, 2016, pp. 1995–2003.

[26] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 449–458.

[27] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin *et al.*, "Noisy networks for exploration," *arXiv preprint arXiv:1706.10295*, 2017.

[28] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PloS one*, vol. 12, no. 4, 2017.

[29] D. H. Wolpert and K. Tumer, "Optimal payoff functions for members of collectives," *Advances in Complex Systems*, vol. 4, pp. 265–279, 2001.

[30] D. T. Nguyen, A. Kumar, and H. C. Lau, "Credit assignment for collective multiagent rl with global rewards," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[31] A. Daly and P. Zannetti, "Air pollution modeling-an overview," *Ambient air pollution*, pp. 15–28, 2007.

[32] J. Weil and R. Brower, "An updated gaussian plume model for tall stacks," *Journal of the Air Pollution Control Association*, vol. 34, no. 8, pp. 818–827, 1984.

[33] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal wsn deployment models for air pollution monitoring," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2723–2735, 2017.

[34] A. Bachir, W. Bechkit, Y. Challal, and A. Bouabdallah, "Joint connectivity-coverage temperature-aware algorithms for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1923–1936, 2014.

[35] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. USA: Prentice Hall PTR, 2001.

[36] M. Hausknecht and P. Stone, "Deep recurrent q-learning for partially observable mdps," in *2015 AAAI fall symposium series*, 2015.