Autonomous Goal Detection and Cessation in Reinforcement Learning: A Case Study on Source Term Estimation

Yiwei Shi[1] *, Muning Wen[2], Qi Zhang[3]†, Weinan Zhang[2]†, Cunjia Liu[4]†, Weiru Liu[1]†

[1] School of Engineering Mathematics and Technology, University of Bristol
[2] Department of Computer Science & Engineering, Shanghai Jiao Tong University
[3] School of Electronic and Information Engineering, Tongji University
[4] Department of Aeronautical and Automotive Engineering, Loughborough University
yiwei.shi@bristol.ac.uk

Abstract

Reinforcement Learning has revolutionized decision-making processes in dynamic environments, yet it often struggles with autonomously detecting and achieving goals without clear feedback signals. For example, in a Source Term Estimation problem, the lack of precise environmental information makes it challenging to provide clear feedback signals and to define and evaluate how the source's location is determined. To address this challenge, the Autonomous Goal Detection and Cessation (AGDC) module was developed, enhancing various RL algorithms by incorporating a self-feedback mechanism for autonomous goal detection and cessation upon task completion. Our method effectively identifies and ceases undefined goals by approximating the agent's belief, significantly enhancing the capabilities of RL algorithms in environments with limited feedback. To validate effectiveness of our approach, we integrated AGDC with deep Q-Network, proximal policy optimization, and deep deterministic policy gradient algorithms, and evaluated its performance on the Source Term Estimation problem. The experimental results showed that AGDC-enhanced RL algorithms significantly outperformed traditional statistical methods such as infotaxis, entrotaxis, and dual control for exploitation and exploration, as well as a non-statistical random action selection method. These improvements were evident in terms of success rate, mean traveled distance, and search time, highlighting AGDC's effectiveness and efficiency in complex, real-world scenarios.

Introduction

Reinforcement Learning (RL) optimizes decision-making and behavior in dynamic environments through trial and error and reward mechanisms. It is widely used in fields such as gaming (Mnih et al. 2015; Silver et al. 2017), robotic control (Levine et al. 2016), autonomous driving (Feng et al. 2023), industrial automation (Degrave et al. 2022) and Preference Alignment (Liu et al. 2022; Bai et al. 2023, 2024). RL involves agents interacting with the environment, learning through exploration and exploitation to achieve their goals.

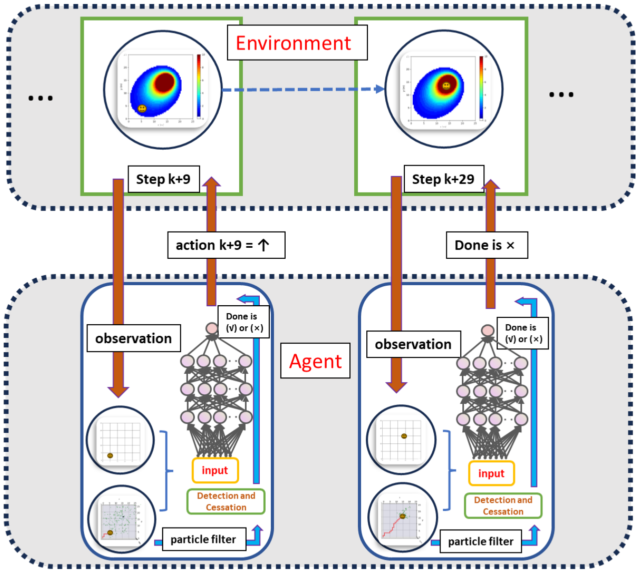


Figure 1: AGDC Structure Diagram for Solving the STE

However, in real-world applications like the Source Term Estimation (STE) problem (Steiner and Bushe 2001), the environment often lacks clear signals for the end of an episode or direct rewards, making it difficult for the agent to determine task completion.

Source Term Estimation involves identifying the location and characteristics (such as release rate) of hazardous gas emissions in the atmosphere, which is crucial for environmental monitoring and emergency response. However, this task presents several challenges. Firstly, the environment often does not provide clear cessation signals, and gas releases are typically invisible. Secondly, atmospheric turbulence results in sensor data that is sparse, intermittent, and time-varying. Additionally, direct measurements within the hazardous material dispersion area are often dangerous. Furthermore, sensor measurements are highly affected by noise and discontinuity. Consequently, agents must rapidly adjust their strategies in response to environmental changes and accurately estimate the source term despite unclear feedback.

*This work is sponsored by University of Bristol Scholarship.
†Corresponding author.

These challenges render source term estimation a complex task with significant uncertainty.

Statistical-based methods such as *Infotaxis*, *Entrotaxis* , and *Dual Control for Exploitation and Exploration* (DCEE) provide some solutions to these challenges. Infotaxis in (Vergassola, Villermaux, and Shraiman 2007; Hutchinson, Liu, and Chen 2018; Loisy and Eloy 2022) guides the search process by maximizing the rate of information acquisition, gradually reducing uncertainty to approximate the source of the gas. It is suitable for situations with high uncertainty regarding the gas source location but suffers from poor real-time performance and a tendency to get stuck in local optima. Entrotaxis in (Hutchinson, Oh, and Chen 2018; Zhao et al. 2020) minimizes the entropy of future measurements to guide path selection, optimizing information acquisition based on real-time measurement data. It performs robustly in noisy environments but has poor adaptability in dynamic settings or sudden changes, making rapid adjustments difficult. DCEE in (Chen, Rhodes, and Liu 2021; Rhodes, Liu, and Chen 2022) combines strategies of exploitation and exploration by dynamically adjusting the balance between the two, optimizing information acquisition and path selection. Suitable for highly variable environments, this strategy is complex to implement, requiring precise design of the trade-off mechanism between exploration and exploitation. Although these methods are effective in addressing the lack of environmental feedback and clear cessation signals, they share several common limitations: poor real-time performance, poor adaptability in rapidly changing environments, a tendency to get stuck in local optima, and complexity in implementation, particularly for the DCEE.

While Reinforcement Learning offers control capabilities, significantly improving efficiency and success rates, and avoiding local optima with good generalization capabilities, it effectively overcomes the limitations of statistical methods. However, RL struggles to effectively identify and autonomously cease goals in the absence of clear feedback. *Traditional statistical methods consist of two modules:* one for providing the agent with reward signals based on estimates, where rewards or information gain originate, and the other for determining the optimal action based on information gain. By replacing the control module of traditional statistical methods with RL while retaining the estimation module, a self-feedback mechanism can be introduced to RL. This helps RL algorithms check and cease goals at appropriate times. Although the rewards for the agent remain sparse in this context, the self-feedback mechanism provides signals from the agent itself rather than the environment, thus better supporting the training process.

Building on this foundation, we propose the concept of Autonomous Goal Detection and Cessation to address the STE problem using RL. AGDC introduces an autonomous goal detection and cessation mechanism within the RL framework, enabling the agent to automatically recognize and cease actions upon task completion. Specifically, the AGDC module employs Bayesian inference to estimate environmental dynamics and dynamically assess task progress. When the standard deviation of the estimated parameter values for environmental dynamics reaches a preset threshold, a cessation signal is automatically triggered. This approach not only addresses the issue of insufficient environmental feedback but also significantly enhances the adaptability and effectiveness of RL in complex and dynamic environments. By integrating the self-feedback mechanism with AGDC, RL algorithms can more efficiently accomplish STE tasks, thereby significantly improving capabilities in environmental monitoring and emergency response. The AGDC structure diagram is shown in Figure 1.

The main contributions of this paper are as follows: (1) We are the first to introduce, to the best of our knowledge, the AGDC module, which enables agents to autonomously recognize and evaluate goals, significantly increasing the capability of RL algorithms in feedback-limited environments. (2) We demonstrate the successful integration of AGDC with multiple RL algorithms in addressing the STE problem, achieving notable improvements over traditional methods. (3) Extensive experimental results confirm the superiority of AGDC-enhanced RL algorithms in terms of success rate, traveled distance, and search time, particularly in challenging environments with high uncertainty and limited feedback.

Problem Formulation and Peliminaries

In a two-dimensional search area denoted as $\Omega \subseteq \mathbb{R}^2$, where there is an expectation of encountering a hazardous release, a robot equipped with a gas sensor is tasked with traversing the area to calculate the release parameters, also referred to as the source term Θ_s. This data will serve as the requisite input for a convection-diffusion model (Vergassola, Villermaux, and Shraiman 2007), enabling the generation of hazard forecasts. Within this context, in an environment characterized by average wind speed $u_s \in \mathbb{R}^+$ in meters per second (m/s), wind direction ϕ_s in radians (rad), and diffusivity d_s in meters squared per second (m^2/s), we detect the concentration of a hazardous substance $x_k \in \mathbb{R}^+$ using a robot or a sensor located at position $p_k = [x_k, y_k]^T \in \Omega$ in meters (m). This hazardous material stems from a source term located at $p_s \in \Omega$, and it is released at a rate/strength $q_s \in \mathbb{R}^+$ in grams per second (g/s) with an average lifespan of $\tau_s \in \mathbb{R}^+$ in seconds (s).

Consequently, the parameters of the source term can be represented as follows:

$$\Theta_s = [p_s^T, q_s, u_s, \phi_s, d_s, \tau_s]^T \tag{1}$$

Note: It is assumed in this study that all other parameters, except for the location of the leak source, can be obtained through direct measurement. Therefore, the primary objective is to successfully obtain the source term p_s^T, as it is the most important information about the source term.

Gaseous Diffusion Model

The average gas concentration (Zhao et al. 2022), denoted as $m(p|\Theta_s)$, for a given location of the mobile sensor at position p_k over a time duration of τ_s, can be computed utilizing the parameters of the source term Θ_s as follows:

$$m(p_k|\Theta_s) = \frac{q_s}{4\pi d_s \|p_k - p_s\|} \exp\left[\frac{-\|p_k - p_s\|}{\lambda} + \psi\right],$$

where

$$\psi = -(x_k - x_s)u_s \cos\phi_s/2d_s - (y_k - y_s)u_s \sin\phi_s/2d_s$$
$$\lambda = \sqrt{d_s\tau_s/[1 + (u_s^2\tau_s/4d_s)]}.$$

Sensor Models

When mobile robots equipped with sensors conduct gas concentration detection in an environment, the measurement results are influenced by both internal and external factors. Internal factors include the sensitivity and calibration issues of the sensors, as well as the robot's internal dynamics during movement, such as vibrations and posture changes. These factors can significantly impact gas concentration measurements. For example, the sensitivity of metal–oxide gas sensors (Li et al. 2011; Neumann et al. 2013; Neumann and Bartholmai 2015) can vary with atmospheric contaminants, causing fluctuations in resistance and, consequently, the voltage readings. Furthermore, calibration issues, as discussed in the context of uncalibrated sensors, can lead to inaccuracies in the measured gas concentrations.

External factors encompass environmental variations in gas temperature, humidity, and pressure, all of which can affect measurement results. Wind turbulence, in particular, is a major contributor to noise in sensor readings. This is modeled using Gaussian distributions to describe the uncertainty in gas concentration measurements more accurately. For instance, the noise caused by wind turbulence is represented as $\nu_{\text{wind}} \sim \mathcal{N}(0, \sigma_{\text{wind}}^2)$, where σ_{wind} is a parameter that characterizes the instability of the wind conditions.

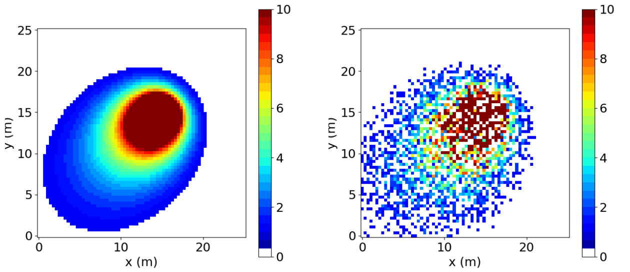
Specifically, the noise due to internal factors from the sensor is represented as v_{inter}. Therefore, the value of the gas concentration $c(p_k|\Theta)$ measured by the robot at position p_k is $c(p_k|\Theta) = m(p_k|\Theta) + v_{\text{inter}}$, where $v_{\text{inter}} \sim \mathcal{N}(u, \sigma)$ and v_{inter} follows a normal distribution $\mathcal{N}$ with mean u and variance σ. This Gaussian noise model is widely used to emulate sensor measurements in simulations, accounting for both wind turbulence and sensor errors.

In a 25-meter by 25-meter area, as an illustrative example, the gaseous diffusion model can provide the gas concentration (GC) at each location. The robot samples the value obtained from the gas concentration model, as shown in Figure (2a), while also considering the presence of noise in the measured values at each location, depicted in Figure (2b). The source term parameters in the gas model are $q_s = 5\,\text{g/s}, u_s = 2\,\text{m/s}, \phi_s = 45°, d_s = 2\,\text{m}^2/\text{s}, \tau_s = 10\,\text{s}$.

In the STE problem, the robot can only obtain (or sample) the gas concentration measurements at its current location during the exploration process. Concentrations at other locations are not accessible, highlighting the importance of accurate sensor modeling and noise representation in ensuring reliable gas concentration measurements

Partially Observable Markov Decision Process

In the search area, the robot's task is to estimate the location of a gas leak source with minimal movement steps. The robot lacks prior knowledge of the leak source's location and gathers data through measurements at each step, making decisions based on this information. To address this challenge, we use the Partially Observable Markov Decision Process (POMDP) Reinforcement Learning method. This approach enables the robot to make effective decisions in an environment that is not fully observable, allowing for accurate localization of the gas leak source. A POMDP is a tuple $(S, O, A, T, Pr, R, \gamma)$, where S is a set of states, O is a set of observations, A is a set of actions, $T(s'|s, a) : S \times A \rightarrow S$ is transition probabilities, $Pr(o|s, a) : S \rightarrow O$ is observation probabilities, $R(s, a) : S \times A \rightarrow \mathbb{R}$ is a reward function, and γ is a discount factor. The objective in POMDP is to obtain an optimal policy to maximize expected cumulative rewards $\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_k]$.



(a) Map of GC model (b) Map of GC with sensor noise

Figure 2: The example of map of the Gas-Diffusion model

Methodology

Bayesian Approximation for Belief Distribution

The STE challenge is characterized as a Partially Observable Markov Decision Process, which implies that optimal decision-making cannot rely solely on the present observation, as complete state information in environment is not available. To address this issue, we have adopted this approach combining RL and Bayesian inference to effectively estimate environmental dynamics (belief) distribution and provide more unobservable information to assist decision-making.

Variational inference (Jordan et al. 1999) and particle filter (Gordon, Salmond, and Smith 1993) are common Bayesian inference methods. The former approximates the target distribution by optimizing a differentiable lower bound but typically requires simplifying assumptions that may not suit complex environments. The latter, on the other hand, handles nonlinear and non-Gaussian distributions more effectively and adapts to dynamic conditions without needing such assumptions. Particle filter approximate the true distribution through sufficient sampling and offer flexibility in managing multimodal distributions, tracking potential source locations and release rates, and providing real-time updates. Therefore, particle filter outperform variational inference in STE problems.

The Particle filter is employed for iterative calculations of the environmental dynamics distribution. The belief distribution at time step k is approximated using a set of N particles, which are random samples $\{\Theta_k^i, w_k^i\}_{i=1:N}$, where Θ_k^i represents the i_{th} point estimation of source parameters (i.e., the belief), and w_k^i is the associated weight, with

$\sum_{i=1}^{N} w_k^i = 1$. The approximated belief distribution $b(\Theta_k)$ is expressed using samples and associated weights as follows:

$$b(\Theta_k) = \sum_{i=1}^{N} w_k^i \delta(\Theta_k - \Theta_k^i), \tag{2}$$

where $\delta(\cdot)$ represents the Dirac delta function, which places all its weight at the value Θ_k^i.

The sampling weights $\{w_k^i\}_{i=1:N}$ are updated through iterative sequential importance sampling (Tokdar and Kass 2010). At each step, a new sample of source parameter estimates $\{\Theta_k^i\}_{i=1:N}$ is obtained from the proposal distribution $q(\Theta_k^i)$. Then, the unnormalized weight $\bar{w}_{k+1}^{(i)}$ corresponding to the source term vector Θ_k of the particle filter is updated as follows:

$$\bar{w}_{k+1}^i \propto w_k^i \cdot \frac{Pr(o_{k+1}|\Theta_{k+1}^i)\, T(\Theta_{k+1}^i|\Theta_k^i)}{q(\Theta_{k+1}^i|\Theta_k^i, \mathbf{o}_{1:k+1})}. \tag{3}$$

Obviously, although the dynamic environment and observation results may appear to change due to measurement errors and noise, the parameters constituting the environment (source term) remain fixed, leading to $\Theta_{k+1} = \Theta_k$ for N particles. Assuming that the proposal distribution is consistent with the posterior distribution, the formula can be simplified as follows, (see additional materials for details):

$$\bar{w}_{k+1}^i = w_k^i \cdot Pr(o_{k+1}|\Theta_{k+1}^i). \tag{4}$$

Normalization of the sampling weights w_k^i can be used to approximate the posterior distribution:

$$w_{k+1}^i = \frac{w_k^i \cdot Pr(o_{k+1}|\Theta_{k+1}^i)}{\sum_{i=1}^{N} w_k^i \cdot Pr(o_{k+1}|\Theta_{k+1}^i)} = \frac{\bar{w}_{k+1}^i}{\sum_{i=1}^{N} \bar{w}_{k+1}^i}. \tag{5}$$

As the number of learning iterations increases, the weights of most particles tend to zero, leading to the degeneracy problem. To address this issue, resampling is employed, and this method is based on the Markov Chain Monte Carlo move step (Ristic, Arulampalam, and Gordon 2004).

We set a resampling threshold ϵ, and when the number of effective samples falls below this threshold, the resampling process is triggered. The effective samples N_{eff} are calculated by:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2}. \tag{6}$$

Autonomous Goal Detection and Cessation

Particle filter not only estimates the distribution of unknown environments but also combines with observational data to provide a more accurate representation of the true state. Additionally, it can generate self-cessation evaluation signals, indicating when computation can be ceased. When the goal is achieved, the particles forming the point estimate converge to a smaller range. By calculating the standard deviation of the particle parameters, the degree of parameter convergence can be observed, facilitating the judgment of goal completion progress.

When the standard deviation (STD) of the belief Θ_k by particle filter is lower than the given Cessation Threshold ζ, the search ceases, and we consider the source term to be successfully estimated, meaning the goal is achieved. As a result, the agent rewards itself rather than relying on the environment. STD is denoted as:

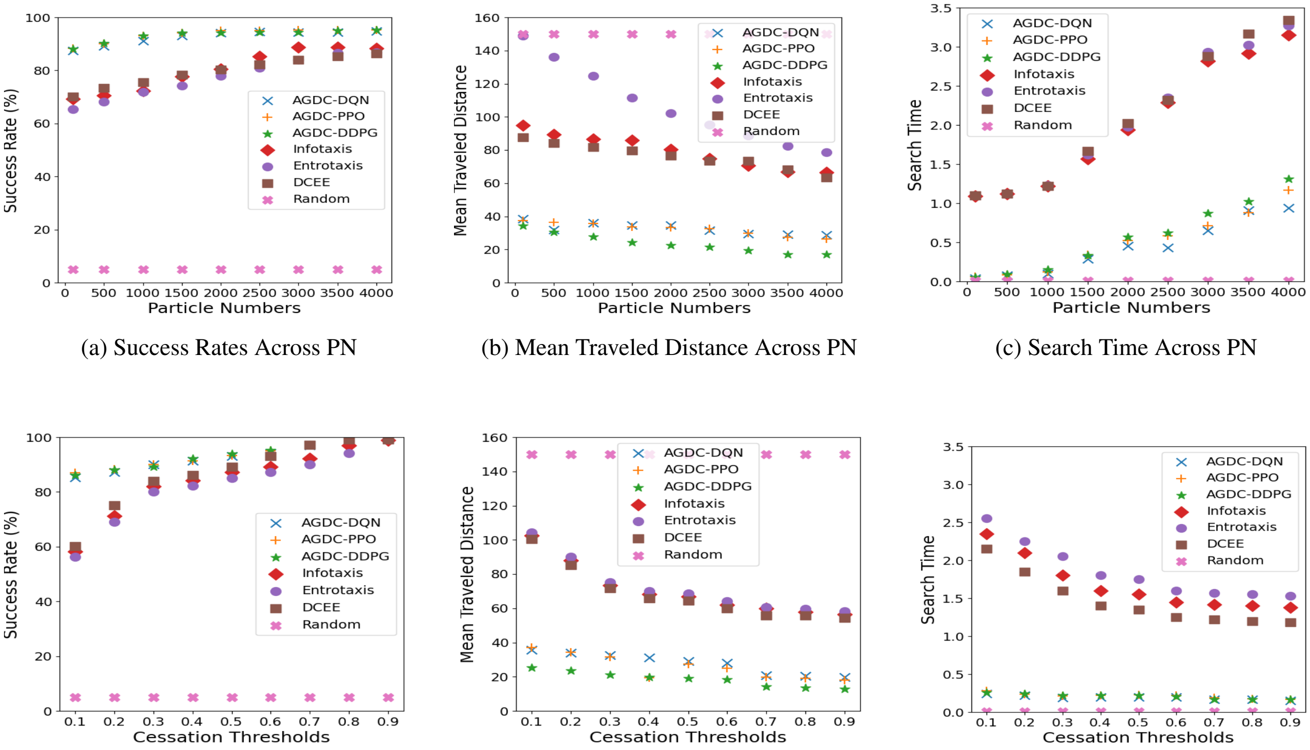
$$STD = \sqrt{diag(Cov(\Theta))}, \tag{7}$$

where $Cov(\cdot)$ represents the covariance, and $diag(\cdot)$ denotes the trace of the matrix.

Calculating the STD of the particle filter belief and comparing it with a threshold as a signal to cease the search process is effective because the standard deviation reflects the concentration level of the belief distribution. A smaller standard deviation indicates that most particles are clustered around a certain estimate, suggesting that the belief state has converged. This convergence implies that the robot's estimation of the source term parameters (such as the leak location) has become more accurate and stable. When the STD is less than the preset threshold ζ, it indicates that the belief state estimation is sufficiently precise, making further algorithm execution yield diminishing returns, thus allowing the search process to cease, saving computational resources and time. Additionally, using the standard deviation as a cessation condition is relatively simple and intuitive, making it easier to implement and understand, facilitating easier decision-making and control in practical applications.

Algorithm 1: Reinforcement Learning with AGDC

```
Initialize environment $\mathcal{E}$, policy $\pi$ and value function $V$,
particle filter with $N$ particles
for episode = 1 to $M$ do
    Initialize observation $o_0$ from $\mathcal{E}$, particles $\{\Theta\}_{i=1}^N$,
    state estimate $\hat{s}_0$ from particles $\{\Theta_k^i\}_{i=1}^N$ and observation $o_0$
    for time step = 1 to $k$ do
        Sample $a_k \sim \pi_\theta(s_k)$, $o_k \sim \mathcal{E}$
        Update particles $\{\Theta_k^i\}_{i=1}^N$ using particle filter
        for each particle $i$ do
            Sample new particle $\Theta_{k+1}^i \sim T(s_{k+1}|s_k, a_k)$
            Compute weight $w_{k+1}^i = w_k^i \cdot Pr(s_{k+1}|\Theta_{k+1}^i)$
            if $N_{eff} < \epsilon$ then
                Normalize weights $w_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N_p} w_t^{(j)}}$
                Resample $\{\Theta_{k+1}^i\}_{i=1}^N$ based on $\{w_{k+1}^i\}_{i=1}^N$
            end if
        end for
        $r_k \leftarrow 0$, $done \leftarrow$ False
        if $STD < \zeta$ then
            $r_k \leftarrow$ value$> 0$, $done \leftarrow$ True
        end if
        Estimate state $\hat{s}_{k+1}$ from $\{\Theta_k^i\}_{i=1}^N$ and $o_k$
        Update policy $\pi$ and $V$ using $\hat{s}_k$, $a_k$, $r_k$, and $\hat{s}_{k+1}$
        Set $o_k \leftarrow o_{k+1}$, $\hat{s}_k \leftarrow \hat{s}_{k+1}$
    end for
end for
```



(a) Success Rates Across PN

(b) Mean Traveled Distance Across PN

(c) Search Time Across PN

(d) Success Rates Across CT

(e) Mean Traveled Distance Across CT

(f) Search Time Across CT

Figure 3: Comparative Analysis of Particle Numbers (PN) and Cessation Thresholds(CT)

For example, suppose the covariance matrix of the particle filter belief state is $Cov(\Theta) = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$. Using *diag*($\cdot$), the standard deviations for the x and y coordinates can be calculated as $STD_{x/y} = \sqrt{\sigma_{x/y}^2}$. This approach allows for an intuitive assessment of the uncertainty for each parameter. In contrast, directly calculating the standard deviation of the entire covariance matrix would require complex matrix operations, making the results less intuitive to understand. Therefore, using *diag*($\cdot$) for standard deviation calculation is a more effective and intuitive method. The details of the training algorithm are shown in Algorithm 1.

Experiment

In this paper, we utilized the STE Environment (STEenv) as the experimental platform to investigate the application of AGDC in RL. The STE Environment, described in Section 2, comprises a Gaussian diffusion model and a sensor model. Instead of providing rewards to the Agent, it transmits the current position and the test concentration at that position as observations. Any RL algorithm can be integrated with the AGDC module to address the challenges posed by the STE.

Baseline Algorithms and Evaluation Metrics

There are three evaluation metrics in this paper: *Success Rate* (SR) to measure the number of successful source term estimations in the trajectories, *Mean Traveled Distance* (MTD) to represent the average distance traveled before a successful estimation occurs, and *Search Time* (ST) to measure the duration from the initiation of a search to the successful estimation of the target information. In this context, achieving a shorter traveled distance while successfully estimating the source's location more frequently, along with a shorter search time indicating a quicker and more effective search process, signifies a more efficient approach.

We integrate the AGDC module into the DQN, PPO, and DDPG algorithms to address the STE problem. Additionally, we use four baseline approaches: Infotaxis, Entrotaxis, Dual Control for Exploitation and Exploration (DCEE), and a method that randomly selects actions for control. The first three are statistical methods, while the last one is a non-statistical method based on random action selection.

Infotaxis (Vergassola, Villermaux, and Shraiman 2007) aims to minimize the predicted posterior variance of the source location. It treats the search process as an information-gathering problem, where the objective is to reduce uncertainty about the source location.

Entrotaxis (Hutchinson, Oh, and Chen 2018) enforces the maximum entropy sampling principle, which directs the agent to move towards positions of maximum uncertainty to gather more informative data about the environment and the source location.

DCEE (Chen, Rhodes, and Liu 2021) integrates both exploitation and exploration by incorporating the uncertainty into the control decisions, thus allowing the robot to not only move towards the estimated target but also probe the environment to reduce uncertainty.

Random involves selecting actions randomly, without any strategic guidance or optimization. It serves as a non-statistical baseline to compare against more sophisticated methods, highlighting the benefits of informed and strategic action selection.

Source Parameter	Distribution
Source Location x_s	Uniform $\mathcal{U}(10, 25)$
Source Location y_s	Uniform $\mathcal{U}(10, 25)$
Release Strength q_s	Uniform $\mathcal{U}(100, 500)$
Wind Speed u_s	Uniform $\mathcal{U}(1, 4)$
Wind Direction ϕ_s	Uniform $\mathcal{U}(0, 360)$
Diffusivity d	Uniform $\mathcal{U}(1, 8)$
Sensor Noise α	Fixed at 0.3
Environmental Noise β	Fixed at 0.2
Effective Samples N_{eff} β	Fixed at 0.5

Table 1: Parameter Distributions for the Training Scenarios

Scenario Parameterization and Evaluation

The parameters for the training scenarios, set within a 30 × 30 area, are constructed by randomly initializing the source and environmental properties at the beginning of each training episode, including parameters such as the gas source location, wind speed, and wind direction, all of which are sampled from the probability distributions presented in Table **??**. The agent starts its search from a random location within the $(0, 5) \times (0, 5)$ area, with a speed of 1 meter per step, ensuring that it encounters a diverse set of scenarios during training, thereby promoting robust learning. The parameters for the testing scenarios consist of 1,000 randomly generated conditions that are not used during training. Although these scenarios are generated from the same parameter ranges as the training data, they present different specific conditions to ensure that the model is evaluated on unseen data.

Particle Numbers and Cessation Thresholds

We will discuss the effectiveness of integrating the AGDC module with RL by comparing three AGDC-enhanced RL algorithms against four baseline methods. Before delving into these comparisons, it is important to introduce two critical hyperparameters: Particle Numbers (PN) and Cessation Thresholds (CT) ζ. The Particle Numbers are pivotal for AGDC as they determine the number of samples used to approximate the point distribution, which is essential for accurately estimating the belief necessary for goal detection. The Cessation Thresholds are directly linked to the cessation aspect of AGDC, as they define the criteria for deciding when to stop the task, ensuring that the process concludes at the most appropriate moment. Two groups of experiments were conducted to investigate the impact of two key hyperparameters on the performance of different algorithms. The first group analyzed the variations in success rate, mean traveled distance, and search time as the number of particles increased from 100 to 4000. The second group examined algorithm performance under different cessation thresholds, ranging from 0.1 to 0.9. The results are presented in Figure 3. The trajectories of six methods (excluding the random method) are shown in Figure 4.

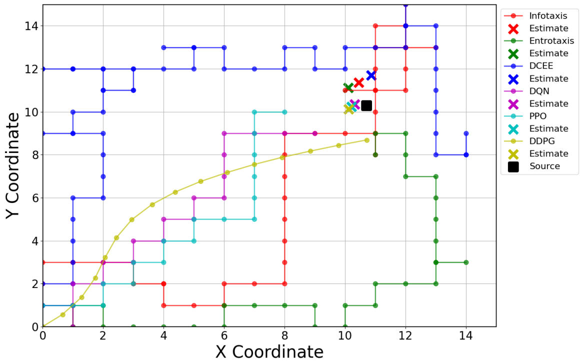


Figure 4: Trajectories of Various Methods

Analysis Across Various Particle Numbers: The success rates of AGDC-enhanced RL algorithms, such as AGDC-DQN, AGDC-PPO, and AGDC-DDPG, improve steadily with increasing particle numbers, reaching 95.27% for AGDC-PPO at 4000 particles. These algorithms outperform baseline methods like Infotaxis, Entrotaxis, and DCEE, which also benefit from more particles but remain less effective. The Random method remains largely ineffective, with success rates below 5%, regardless of particle count. Notably, DDPG's continuous action space allows for more flexible movement compared to the discrete steps of other methods. AGDC-enhanced algorithms also show a consistent reduction in mean traveled distance as particle numbers increase. For instance, AGDC-DDPG reduces its mean distance from 34.33 at 100 particles to 16.83 at 4000, demonstrating greater efficiency. In contrast, baseline methods like Entrotaxis exhibit significantly higher distances, highlighting their inefficiency. The Random method, unaffected by particle count, continues to exhibit very high traveled distances. Regarding search time, AGDC-enhanced algorithms consistently achieve quicker convergence compared to baseline methods. AGDC-DQN, for example, maintains a search time of 0.94 units at 4000 particles. In contrast, methods like Infotaxis take significantly longer, reaching 3.15 units, indicating slower and less effective strategies. The Random method, despite its speed, fails to produce meaningful results, underscoring its overall inefficiency.

Analysis Across Various Cessation Thresholds: The success rates of AGDC-enhanced algorithms, including AGDC-DQN, AGDC-PPO, and AGDC-DDPG, improve with higher cessation thresholds, with AGDC-PPO achieving 99.13% at a threshold of 0.9. This is due to the algorithms accumulating more confidence before stopping, leading to greater accuracy. Although Infotaxis and Entrotaxis also improve with higher thresholds, they require significantly more time and distance to reach similar success levels. As cessation thresholds rise, AGDC-enhanced algorithms also show reduced mean traveled distances, reflecting more efficient search cessation. AGDC-DDPG, for instance, reduces its distance to 12.8 units at a threshold of 0.9, while Infotaxis, even at its best, remains at 56.4 units, much higher than AGDC-DDPG. Search times for AGDC-enhanced algorithms decrease with higher thresholds, with AGDC-PPO reducing its time to 0.17 units at a threshold of 0.9. This indicates that these algorithms can efficiently conclude the search once confident in their estimation. **However, excessively high thresholds may artificially inflate success rates and reduce traveled distances, creating a misleading appearance of improved performance by lowering the task's rigor. Therefore, we advocate for setting an appropriate threshold that balances accuracy and task difficulty.**

Further Research Experiments

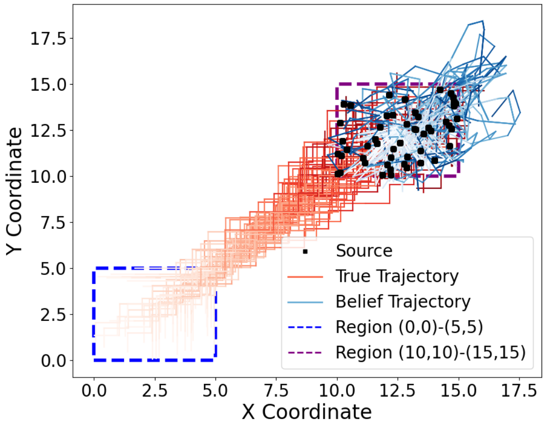


Figure 5: Multiple Trajectories from True Trajectory and Estimated (Belief) Trajectory with Color Gradient

An additional experiment was conducted to explore the AGDC module through a visualization approach. This experiment uses only the AGDC-based QDN method, with the environment set within a 20×20 area, and the source randomly located within the x, y from $\mathcal{U}(15, 20)$. The experimental results are shown in Figure 5 and 6. In Figure 5, 100 trajectories are displayed, with the red lines representing the agent's actual paths. The color intensity correlates with the time step count, with darker lines indicating more steps. The blue lines show the changes in the estimated source position. As the agent gets closer to the true source location (the goal), the estimated position (derived from Belief) also becomes more accurate, converging towards the goal.

Similarly, in Figure 6, the red line represents the distance between the agent and the true goal at each step, while the blue line shows the distance between the agent and the current estimated position. The red line in Figure 6 is relatively smooth, indicating that the agent consistently moves towards the true goal, even without knowing its exact location. This can explain why AGDC-based RL methods are efficient, as they avoid taking redundant steps. During this process, the estimated position (Belief) fluctuates with the incoming information but eventually converges near the true goal. We observed that towards the end, the blue line closely aligns with the red line, indicating high estimation accuracy. By setting different CT ζ, represented by the horizontal blue lines, we can determine varying success rates based on the number of red line points below the threshold. This explains why a larger threshold increases the success rate; however, when set too high, the prediction becomes distorted. Combining insights from both figures, we conclude that the agent can effectively achieve Autonomous Goal Detection and Cessation in a gradual manner. Even though the red line may not be available in real-time, the trend of the blue line can still be used to roughly estimate the current progress of recognition, which serves as a form of explainability for ADGC.

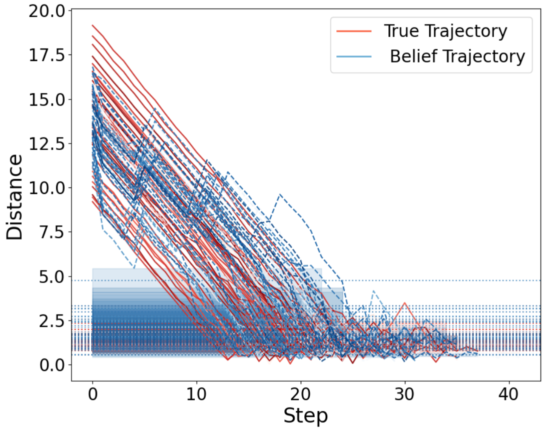


Figure 6: Distance from current position to goal and estimated belief at each step

Conclusion

In this study, we introduced the concept of Autonomous Goal Detection and Cessation within the Reinforcement Learning framework to address the challenges of STE. By integrating Bayesian inference with Reinforcement Learning, AGDC enhances the agent's ability to autonomously detect and cease actions upon achieving the goal, significantly improving adaptability and efficiency in complex and dynamic environments. Our experiments demonstrate that AGDC-based methods consistently outperform traditional statistical approaches in terms of success rate, traveled distance, and search time across varying scenarios. This indicates that AGDC provides a robust mechanism for agents to effectively navigate uncertain environments, making it a valuable tool for applications in environmental monitoring and emergency response. Future work could explore the scalability of AGDC to more complex multi-agent systems and real-world deployment scenarios.

References

Bai, F.; Zhang, H.; Tao, T.; Wu, Z.; Wang, Y.; and Xu, B. 2023. PiCor: Multi-Task Deep Reinforcement Learning with Policy Correction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6): 6728–6736.

Bai, F.; Zhao, R.; Zhang, H.; Cui, S.; Wen, Y.; Yang, Y.; Xu, B.; and Han, L. 2024. Efficient Preference-based Reinforcement Learning via Aligned Experience Estimation. *arXiv preprint arXiv:2405.18688*.

Chen, W.-H.; Rhodes, C.; and Liu, C. 2021. Dual control for exploitation and exploration (DCEE) in autonomous search. *Automatica*, 133: 109851.

Degrave, J.; Felici, F.; Buchli, J.; Neunert, M.; Tracey, B.; Carpanese, F.; Ewalds, T.; Hafner, R.; Abdolmaleki, A.; de Las Casas, D.; et al. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419.

Feng, S.; Sun, H.; Yan, X.; Zhu, H.; Zou, Z.; Shen, S.; and Liu, H. X. 2023. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953): 620–627.

Gordon, N. J.; Salmond, D. J.; and Smith, A. F. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, 107–113. IET.

Hutchinson, M.; Liu, C.; and Chen, W.-H. 2018. Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments. *IEEE Transactions on Control Systems Technology*, 27(6): 2388–2402.

Hutchinson, M.; Oh, H.; and Chen, W.-H. 2018. Entrotaxis as a strategy for autonomous search and source reconstruction in turbulent conditions. *Information Fusion*, 42: 179–189.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S.; and Saul, L. K. 1999. An introduction to variational methods for graphical models. *Machine learning*, 37: 183–233.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39): 1–40.

Li, J.-G.; Meng, Q.-H.; Wang, Y.; and Zeng, M. 2011. Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm. *Autonomous Robots*, 30: 281–292.

Liu, R.; Bai, F.; Du, Y.; and Yang, Y. 2022. Meta-Reward-Net: Implicitly Differentiable Reward Learning for Preference-based Reinforcement Learning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 22270–22284. Curran Associates, Inc.

Loisy, A.; and Eloy, C. 2022. Searching for a source without gradients: how good is infotaxis and how to beat it. *Proceedings of the Royal Society A*, 478(2262): 20220118.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Neumann, P. P.; and Bartholmai, M. 2015. Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit. *Sensors and Actuators A: Physical*, 235: 300–310.

Neumann, P. P.; Hernandez Bennetts, V.; Lilienthal, A. J.; Bartholmai, M.; and Schiller, J. H. 2013. Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms. *Advanced Robotics*, 27(9): 725–738.

Rhodes, C.; Liu, C.; and Chen, W.-H. 2022. Autonomous source term estimation in unknown environments: From a dual control concept to UAV deployment. *IEEE Robotics and Automation Letters*, 7(2): 2274–2281.

Ristic, B.; Arulampalam, S.; and Gordon, N. J. 2004. Beyond the Kalman Filter: Particle Filters for Tracking Applications.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Steiner, H.; and Bushe, W. 2001. Large eddy simulation of a turbulent reacting jet with conditional source-term estimation. *Physics of Fluids*, 13(3): 754–769.

Tokdar, S. T.; and Kass, R. E. 2010. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1): 54–60.

Vergassola, M.; Villermaux, E.; and Shraiman, B. I. 2007. 'Infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126): 406–409.

Zhao, Y.; Chen, B.; Wang, X.; Zhu, Z.; Wang, Y.; Cheng, G.; Wang, R.; Wang, R.; He, M.; and Liu, Y. 2022. A deep reinforcement learning based searching method for source localization. *Information Sciences*, 588: 67–81.

Zhao, Y.; Chen, B.; Zhu, Z.; Chen, F.; Wang, Y.; and Ma, D. 2020. Entrotaxis-Jump as a hybrid search algorithm for seeking an unknown emission source in a large-scale area with road network constraint. *Expert Systems with Applications*, 157: 113484.