Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Design of Marine Oil Spill Emergency Resource Scheduling Framework Based on Improved A*-PSO-SA-BPPO

Jihao An
China University of Petroleum, East China
Peng Ren
China University of Petroleum, East China
Xinrong Lyu
lvxr@upc.edu.cn
China University of Petroleum, East China
Christos Grecos
University of Wisconsin–Parkside

Research Article

Keywords: Marine oil spill, Emergency resource scheduling, PSO, PPO, Path planning, Bayesian estimation

Posted Date: April 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9253567/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Design of Marine Oil Spill Emergency Resource Scheduling Framework Based on Improved A*-PSO-SA-BPPO

Jihao An[1], Peng Ren[1], Xinrong Lyu[1*], Christos Grecos[2]

[1]College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, 266580, China.
[2]Computer Science Department, University of Wisconsin Parkside, Jonesboro, AR, 72401, USA.

*Corresponding author(s). E-mail(s): lvxr@upc.edu.cn;

Abstract

Marine oil spills cause severe damage to ocean ecosystems and coastal economies. Timely scheduling of oil spill emergency resources is crucial for effective emergency response. Addressing the challenge of efficient resource scheduling under dynamically evolving oil spill conditions in marine environments, this paper proposes a hierarchical hybrid intelligent architecture. Firstly, it integrates an improved A*-PSO path planning module and an attention-based Bayesian Proximal Policy Optimization (SA-BPPO) reinforcement learning module to construct a 'path navigation-decision optimization' model for oil spill emergency resource scheduling. The improved A*-PSO path planning module generates initial paths and performs smoothing optimization, enabling rapid planning of efficient paths from start to end points, enhancing their feasibility and safety. Secondly, the self-attention Bayesian PPO (SA-BPPO) reinforcement learning module focuses on optimizing ship cleaning strategies. By incorporating an attention mechanism, the model's ability to focus on key information is enhanced. The Bayesian method estimates the uncertainty of strategy outputs, allowing ships to dynamically adjust cleaning strategies according to environmental changes. Experimental results demonstrate that the proposed architecture significantly outperforms algorithms like Deep Q-Learning (DQN), Soft Actor-Critic (SAC), Ant Colony Optimization (ACO), and Dijkstra in key indicators such as cleaning time, efficiency, and path optimization. Compared to DQN, the improved A*-PSO-SA-BPPO algorithm improves average cleaning efficiency by 36%. Compared to SAC, it improves average cleaning efficiency by 9.17%. Compared to ACO and Dijkstra algorithms, it improves average cleaning efficiency by 36.78% and 75%, respectively. Ablation experiments

further validate the effectiveness of each module that adding the improved A*-PSO module reduces cleaning time by 17.24% and increases cleaning efficiency by 39.45%.

Keywords: Marine oil spill, Emergency resource scheduling, PSO, PPO, Path planning, Bayesian estimation

1 Introduction

With the development of the global marine transportation industry, the frequency and scale of marine oil spill incidents have shown an upward trend, demanding higher capabilities in emergency response and resource scheduling.

For marine oil spill emergency resource scheduling, the first step is to quickly determine the location and extent of the spill. Subsequently, path planning is applied to form efficient and safe navigation routes for emergency resources like ships, guiding them swiftly to the spill area. Then, dynamic adjustment and decision optimization are performed during the cleanup operation. Based on the dynamic diffusion trend of the spill and real-time on-site conditions, the resource scheduling plan is adjusted in real-time to achieve multi-resource collaborative operations, enhancing cleanup efficiency and effectiveness. In the field of path planning, Particle Swarm Optimization (PSO) algorithms are widely used (Panda et al. 2017; Pu and Huang 2018; Ding et al. 2018), but they are prone to falling into local optima and have limited adaptability in dynamic environments. Addressing this, Ning et al. (2023) introduced a Gaussian position mutation strategy, effectively expanding the search space. Meanwhile, the A* algorithm has also gained attention in ship path planning. Vo et al. (2019) optimized path costs through a heuristic guidance strategy, improving planning efficiency in complex environments. Furthermore, research by Liu et al. (2019) on integrating ant colony algorithms with clustering algorithms, and Xin et al. (2019) on enhancing genetic algorithms for multi-domain inversion, have further pushed the boundaries of traditional optimization algorithms in dynamic environments.

With breakthroughs in Reinforcement Learning (RL) technology, path planning research has gradually shifted towards autonomous decision-making (Gong 2021; Zhai 2023; Guo 2021; Guan et al. 2023). However, the generalization ability of RL in dynamic environments still faces challenges, such as the overestimation problem of DQN identified by Li et al. (2023) and the convergence difficulty under sparse rewards identified by Deraj et al. (2023). This has prompted exploration of new methods, such as Bayesian optimization by Li et al. (2024a), and Zhu et al. (2025) proposed an adaptive generation algorithm for high-risk test scenarios for autonomous collision avoidance decision-making algorithms for ships. They established a reinforcement learning test framework based on Markov decision processes and constructed a collision risk index model using a fuzzy analytic hierarchy process. They addressed the issue of ship collision avoidance decision-making through deep reinforcement learning and the generation of high-risk test scenarios.

In the field of oil spill emergency resource scheduling, research often focuses on multi-objective optimization (Huang et al. 2019; Zhu et al. 2019; Chen 2020; Zhang et al. 2022). However, the limited adaptability of traditional optimization methods to uncertain environments has driven research towards integrating RL with historical data. Ou et al. (2024) proposed a dual-layer architecture combining DDPG for local collision avoidance and DQN for path regression. Their method leverages historical AIS data to extract characteristic trajectories via K-means clustering, providing a global static path reference. Unlike Ou et al. (2024)'s dual-layer DDPG-DQN framework, our method integrates an improved A*-PSO module for hierarchical path planning and a self-attention Bayesian PPO module for adaptive cleanup strategy optimization. This allows our model to handle large-scale oil spill diffusion and real-time uncertainty estimation, which are critical in open-sea emergency responses.

Recent advancements in oil spill detection and simulation further support these efforts, including SAR-based segmentation (Chai et al. 2025), forecast-corrected drift prediction (Ren et al. 2025; Li et al. 2024b); Shi et al. (2024) developed a few-shot segmentation network that employs adaptive feature enhancement to transfer from UAV optical to SAR imagery, achieving 75.88% accuracy in 5-shot experiments. Yu et al. (2025) conducted GNSS-R-based simulation studies for marine oil-spill detection that incorporate sea-surface asymmetry and wind-speed effects via KA-GO and Z-V models, validated with CYGNSS data. Ozair et al. (2024) applied SAR-based detection techniques to assess the impact on coastal and marine environments, reporting that spills can affect up to 118 km^2 of coastal areas under favorable wind conditions (3–9 m s^{-1}). Kuzmenko et al. (2025) integrated liquid neural networks with swarm robotics on MOOS-IvP platforms for autonomous multi-agent response, improving trajectory forecasting by 23% over LSTM in Deepwater Horizon simulations. These works reinforce the necessity of integrating real-time forecast fields with adaptive decision-making.

Nevertheless, existing research still lacks a task-coordinated scheduling mechanism integrating path planning and decision optimization specifically for oil spill emergency scenarios. Therefore, to address the multi-stage optimization problem of marine oil spill emergency resource scheduling, this paper adopts a solution integrating improved A*-PSO path planning and self-attention Bayesian PPO (SA-BPPO) reinforcement learning. It achieves a full-chain intelligent decision-making process from spill area detection to dynamic path planning and efficient cleanup, significantly enhancing the speed and effectiveness of oil spill emergency response. The model architecture designed in this paper is shown in Fig. 1.

2 Datasets

2.1 Experimental Area

The sea area near Qingdao, located in a temperate monsoon climate zone, has a complex marine environment. It is an important fishery and economic zone, home to large ports like Qingdao Port, with busy ship traffic and numerous offshore oil and gas extraction facilities and port oil storage facilities. This creates risks of oil spills from ship collisions, platform accidents, and storage facility leaks. The experimental

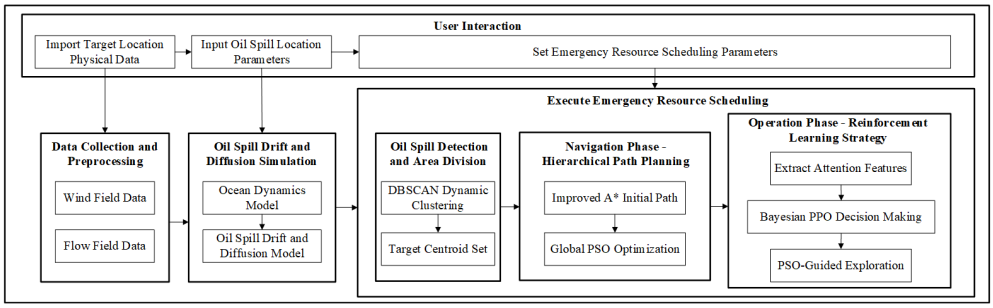


Fig. 1 Technical Architecture Diagram

area covers the sea near Qingdao with longitude approximately 120.0°E to 122.0°E and latitude 35.0°N to 37.0°N.

2.2 Marine Dynamic Data Acquisition

The drift and diffusion of marine oil spills are primarily driven by the combined action of wind fields and flow fields. The sources of both data used in this paper are shown in Table 1.

Table 1 Marine Environmental Parameter Datasets

Data	Source	Content
Wind Field	NCEP-NCAR Reanalysis 1 Product	Time coverage: 4 daily values from June 1 to June 10, 2021. Spatial coverage: 2.5°×2.5° global grid (30.0°E to 210.0°E, 30.0°N to 39.0°N).
Flow Field	Global Ocean Physics Reanalysis GLORYS12V1	Time coverage: 1 daily value from June 1 to June 10, 2021. Spatial coverage: 118.0°E to 216.0°E, 33.33°N to 41.25°N.

The wind field and flow field data (Table 1) used in the current experiment are only used as the power input to drive the drift of oil particles and have not yet further distinguished the limitations of sea conditions such as "big waves" or "quiet sea" on the operational capabilities of ships. Therefore, the three speed modes (27 km/h, 18 km/h, 5.4 km/h) in Table 7 are only related to the relative distance from the center of the oil spill, and the forced deceleration rules when the wind speed is greater than 15 m/s or the wave height is greater than 2 m have not been superimposed for the time being.

2.3 Marine dynamic data preprocessing

No additional wind/flow preprocessing was performed in this study. The wind and current fields listed in Table 1 are ingested directly by the open-source OpenOil module (OpenDrift 3.6).

CMEMS.nc files (1/12° daily) and NCEP-NCAR uwnd/vwnd files (1.875° 6-hourly) are read natively via reader_netCDF_CF_generic without interpolation, re-gridding, or gap-filling.

Spatial domain (118.08–126.00 °E, 33.33–41.25 °N) and temporal range (2021-06-01 00:00–2021-06-10 00:00) are automatically aligned by OpenDrift's internal time-matching mechanism.

Variables (x_sea_water_velocity, y_sea_water_velocity, x_wind, y_wind) are used as-is, ensuring full reproducibility of the drift-diffusion simulation with the public repository at https://github.com/OpenDrift/opendrift.

2.4 Oil Spill Drift and Diffusion Trajectory

This paper employs the Lagrangian-based OpenDrift oil spill model to simulate the drift and diffusion of marine oil spills. The results are shown in Fig. 2, where 800 simulated oil particles were used, initial leakage radius is 2km, and input physical data included wind and current fields from June 1 to June 10, 2021. The green area represents the initial position of oil particles, the blue area represents their final position, and the red area represents their stranded position.

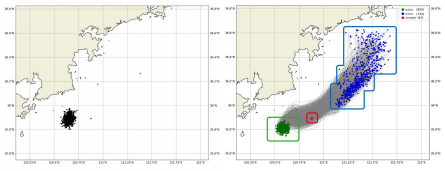


Fig. 2 Oil Spill Drift and Diffusion

3 Hierarchical Path Planning Navigation Based on Improved A*-PSO

This paper proposes an "initial path-global optimization-dynamic correction" hierarchical path planning architecture, achieving efficient collaboration through algorithm and information transfer mechanisms.

3.1 Path Navigation Based on Improved A* Algorithm

In the navigation phase, this paper first improves the A* algorithm, using the centroid of the oil spill area as the target point to quickly generate an initial shortest path from the start point to the end point.

3.1.1 Hierarchical Grid Mapping

This paper divides the sea area into 0.05° grids to plan the shortest path for ships from their current position to the target oil spill area. Table 2 shows the grid search space strategy.

Table 2 Grid Search Space Strategy

Gridded Search Space	Strategy
Coarse-grained layer (0.1°)	Global search, quickly generates initial paths, reduces computational overhead.
Fine-grained layer (0.05°)	Local obstacle avoidance, handles dynamic ships and temporary obstacles.

3.1.2 Improved Cost Function

To avoid obstacles, an obstacle distance penalty term is introduced, extending the traditional A* cost function $f(n) = g(n) + h(n)$ to:

$$f(n) = g(n) + \lambda \cdot h(n) + \mu \cdot d(n) \tag{1}$$

Where, $\lambda = 1.2$, $\mu = 0.5$ represent the weight coefficients for the heuristic function and the obstacle penalty term, respectively. These coefficients were determined through grid search within the ranges of $\lambda \in \{1.0, 1.1, 1.2, 1.3\}$ and $\mu \in \{0.3, 0.4, 0.5, 0.6\}$. This combination improved the average path efficiency by 6.52% without increasing collision risk. $g(n)$ is the actual moving cost from the start point to node n considering ship steering energy consumption:

$$g(n) = \sum_{k=1}^{n} \left(||p_k - p_{k-1}|| + \eta \cdot |\theta_k - \theta_{k-1}| \right) \tag{2}$$

Where, p_k is the first k a path to the point of latitude and longitude coordinates, (unit: °). $p_k - p_{k-1}$ is the distance cost from node $k-1$ to node k, reflecting the direct movement expense of the path; θ_k is the heading Angle of the ship at the kth path point (unit: rad). $\eta \cdot |\theta_k - \theta_{k-1}|$ is the turning penalty term, with $\eta = 0.1$ is the turning penalty coefficient and $|\theta_k - \theta_{k-1}|$ is the absolute difference in turning angles, accounting for the additional cost of vessel maneuvers.

In Eq. (1), $h(n)$ is the heuristic function:

$$h(n) = (|x_n - x_{\text{goal}}| + |y_n - y_{\text{goal}}|) \cdot \left(1 + \frac{t}{T_{\max}}\right) \tag{3}$$

Where, t as the current time step and $T_{\max}$ as the maximum time step, $|x_n - x_{\text{goal}}| + |y_n - y_{\text{goal}}|$ is the Manhattan distance, estimating the distance from the current node n to the target node; $\left(1 + \frac{t}{T_{\max}}\right)$ is the time dynamic weighting term, designed to encourage faster convergence to the target as time progresses.

In Eq. (1), $d(n)$ is the obstacle distance penalty term:

$$d(n) = \sum_{o_j \in \mathcal{O}_t} \max\left(0, 1 - \frac{||p_n - o_j||}{d_{\text{safe}}}\right) \tag{4}$$

Where, o_j as the obstacle and $d_{\text{safe}} = 0.02° \approx 2.2$km as the safe distance threshold, ensuring the path avoids obstacles.

3.1.3 Dynamic Re-planning Mechanism

Local path updates are triggered when any of the following conditions are detected:

Ship deviation from current path exceeds threshold: $\delta_{\text{dev}} = 0.1°$

New obstacle enters safety radius:

$\mathcal{O}_t = \{(x_j, y_j) | \exists \text{r}_k, ||(x_j, y_j) - (x_k, y_k)|| < 0.02\}$

Oil spill centroid offset: $||c_{\text{new}} - c_{\text{old}}|| > 0.02°$

3.2 Path Smoothing Optimization Based on PSO

Although the improved A* algorithm can find a feasible path from start point to end point, this path may contain many sharp turns, affecting ship navigation efficiency and safety. Therefore, this paper introduces the Particle Swarm Optimization (PSO) algorithm to smooth the path generated by the improved A*.

3.2.1 Particle Encoding Design

Each particle represents a candidate path, encoded as a sequence of intermediate waypoints:

$$X_i = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\} \tag{5}$$

Where, (x_k, y_k) are intermediate points uniformly sampled on the initial path generated by the improved A* algorithm, ensuring path traversability.

3.2.2 Dynamic Fitness Function

Considering path length, smoothness, and safety margin, the dynamic fitness function proposed in this paper is shown below:

$$F(X_i) = \alpha \cdot L(X_i) + \beta \cdot S(X_i) + \gamma \cdot M(X_i) \tag{6}$$

Experimental results indicate that the weight coefficients in the formula are $\alpha = 0.7$, $\beta = 0.3$, $\gamma = 0.2$.

The path length term is:

$$L(X_i) = \sum_{k=1}^{n-1} ||p_{k+1} - p_k||_2 \tag{7}$$

The smoothness term is measured by waypoint curvature variance:

$$S(X_i) = \frac{1}{n-2} \sum_{k=2}^{n-1} \left(\frac{||(p_{k+1} - p_k) - (p_k - p_{k-1})||}{||p_k - p_{k-1}||} \right)^2 \tag{8}$$

The safety margin term avoids proximity to obstacles:

$$M(X_i) = \sum_{k=1}^{n} \sum_{o_j \in \mathcal{O}} \exp\left(-\frac{||p_k - o_j||^2}{2\sigma^2} \right) \tag{9}$$

Where, $\sigma = 0.02°$ which is the safety distance threshold, $\mathcal{O}$ is the set of static obstacles.

3.2.3 Velocity Update Strategy

Particle velocity and position updates follow the standard PSO mechanism:

$$v_i^{t+1} = \omega \cdot v_i^t + c_1 r_1 (pbest_i - X_i^t) + c_2 r_2 (gbest - X_i^t) \tag{10}$$

$$X_i^{t+1} = X_i^t + v_i^{t+1} \tag{11}$$

Where, the inertia weight ω =0.729, the acceleration coefficients $c_1 = c_2 = 1.494$, $r_1, r_2 \sim U(0,1)$. To prevent oscillation, velocity clamping is introduced:

$$v_{\max} = 0.1 \cdot (X_{\max} - X_{\min}) \tag{12}$$

3.3 Hierarchical Path Planning Algorithm Based on Improved A*-PSO

The improved A-PSO hierarchical path planning algorithm quickly generates an initial path using the improved A* algorithm and enhances path smoothness and safety by introducing the Particle Swarm Optimization (PSO) algorithm. This mechanism balances planning efficiency and path quality by reducing global search space and local computational complexity. Hierarchical optimization ensures smooth transitions in the global path, avoiding local oscillations. Improved A* ensures the path length is near-optimal, while PSO ensures path navigability and safety. Fig. 3 shows the flowchart of the improved A*-PSO hierarchical path planning algorithm.

4 Cleanup Strategy Based on Bayesian PPO

4.1 PPO Algorithm

The cleanup phase of marine oil spills requires ships to efficiently traverse the spill area, maximizing oil removal efficiency. Traditional methods struggle to adapt to dynamic changes in oil distribution and environmental uncertainty. Reinforcement Learning (RL), by learning through interaction with the environment, provides an adaptive decision-making framework.

PPO is a policy gradient-based RL algorithm that addresses issues in traditional RL such as unstable policy updates, policy collapse, and the difficulty of balancing optimization efficiency and stability.

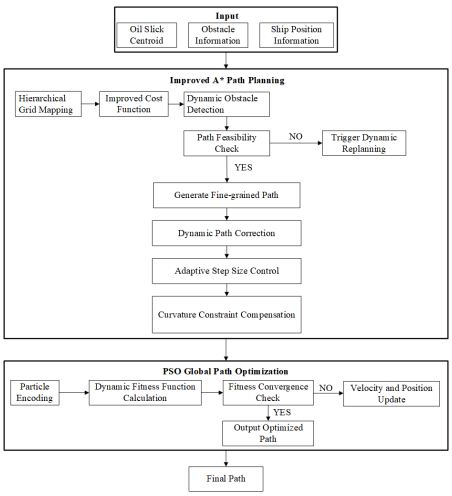


Fig. 3 Flowchart of Improved A*-PSO Hierarchical Path Planning Algorithm

4.1.1 PPO Objective

PPO limits the policy update magnitude by clipping the objective function.

$$L^{\text{CLIP}}(\theta) = \widehat{\mathrm{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t \right) \right] \tag{13}$$

Where, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, ε controls the update magnitude. $\hat{A}_t$ is the GAE (Generalized Advantage Estimate), calculated by:

$$\hat{A}_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \tag{14}$$

Where, $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ is the value function, γ is the discount factor, adaptively adjusted through the principle of policy entropy maximization.

4.1.2 Value Function and Entropy Regularization

The PPO model value function loss is designed as follows:

$$L^{VF}(\theta) = \widehat{\mathrm{E}}_t \left[\left(V_\theta(s_t) - V_t^{\text{target}} \right)^2 \right] \tag{15}$$

Where, $V_t^{\text{target}} = r_t + \gamma V(s_{t+1})$. To encourage exploration, entropy regularization is introduced:

$$S[\pi_\theta](s_t) = -\sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t) \tag{16}$$

The total loss is shown below:

$$L(\theta) = L^{\text{CLIP}}(\theta) - c_1 L^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \tag{17}$$

4.2 Uncertainty Estimation — Bayesian Proximal Policy Optimization (BPPO) Framework

Traditional PPO relies on deterministic policy updates, making it difficult to effectively quantify decision risk and being prone to collapse during abrupt environmental changes. To address these issues, this paper proposes a Bayesian Proximal Policy Optimization (BPPO) framework. By integrating Monte Carlo Dropout and an uncertainty-aware reward mechanism, it achieves risk-controllability and environmental adaptability in policy optimization.

4.2.1 Monte Carlo Dropout Mechanism

This mechanism introduces Dropout layers into the policy network $\pi_\theta(a|s)$, treating them as approximate variational inference for a Bayesian neural network. The network randomly drops some neurons during each forward pass, producing probabilistically meaningful policy outputs, dropout rate p=0.3 is chosen following Gal and Ghahramani (2016). Our ablation study also showed that a higher rate (p=0.5) increased cleanup time by approximately 3 steps, while a lower rate (p=0.1) underestimated variance and caused policy oscillation. For a given state s_t ,the uncertainty estimate of the action distribution is:

$$\mathrm{E}[a_t] = \frac{1}{M}\sum_{m=1}^{M} \pi_\theta(a_t|s_t, \varepsilon_m) \tag{18}$$

$$\mathrm{Var}(a_t) = \frac{1}{M}\sum_{m=1}^{M} \left(\pi_\theta(a_t|s_t, \varepsilon_m) - \mathrm{E}[a_t]\right)^2 \tag{19}$$

In Eqs. (18) and (19), $\varepsilon_m \sim \mathrm{Dropout}(p)$ is the m-th Monte Carlo sample, and M=10 is the number of samples , which is selected based on experience. This is a commonly used compromise between estimation accuracy and computational efficiency. This design allows the agent to perceive epistemic uncertainty arising from environmental dynamics, automatically enhancing exploratory behavior during sudden oil spill diffusion changes.

4.2.2 Uncertainty-Aware Policy Optimization

This experiment introduces an uncertainty penalty term into the PPO objective function, constructing a risk-sensitive policy update mechanism. The Bayesian PPO loss function is designed as follows:

$$\begin{gathered}\mathcal{L}^{BPPO}(\theta) = \\ \mathrm{E}_t\left[\min\left(r_t(\theta)A_t, \mathrm{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)A_t\right)\right] - \lambda \mathrm{Var}(a_t)\end{gathered} \tag{20}$$

Where, $\lambda = 0.5$ is the uncertainty penalty coefficient. This formula ensures policy update stability while suppressing the selection probability of high-variance actions, exhibiting smoother control characteristics in navigation path planning.

4.2.3 Policy Update Constraint

To avoid overly conservative policy updates, a dynamic threshold constraint is applied to the uncertainty penalty term, shown in Eq. (21):

$$\mathrm{E}_s[U(s)] \leq U_{\max} = \frac{1}{\sqrt{K}} \cdot \mathrm{E}_s\left[\max_a \pi_{\theta_{\text{old}}}(a|s)\right] \tag{21}$$

Where, $\mathrm{E}_s[U(s)]$ denotes the expected value of uncertainty $U(s)$ over states , representing the average level of decision uncertainty across the state space, with $U(s)$ estimated via Bayesian methods.$\mathrm{E}_s\left[\max_a \pi_{\theta_{\text{old}}}(a \mid s)\right]$ represents the expected value of the old policy $\pi_{\theta_{\text{old}}}$'s confidence in the optimal action in state s , where $\max_a \pi_{\theta_{\text{old}}}(a|s)$ measures the old policy's certainty in the best action. When the actual uncertainty exceeds $U_{\max}$, it triggers re-initialization of the policy network parameters, preventing the model from getting stuck in local minima.

4.2.4 Uncertainty-Guided Exploration Strategy

In the action selection phase, cognitive uncertainty $U(s) = \mathrm{Var}(a_t)$ used to dynamically adjust the Boltzmann explore temperature $\tau(s)$ Eq. (22). When uncertainty is high, the temperature rises, the distribution of action probabilities becomes flatter, and exploration is encouraged. When uncertainty is low, the temperature drops, and the distribution of action probabilities becomes sharper, tending to utilize the current optimal strategy:

$$\tau(s) = \tau_{\text{base}} \cdot (1 + \beta U(s)), \quad \beta = 0.5 \tag{22}$$

The adjusted action probability is shown below:

$$\pi_{\text{exp}}(a|s) = \frac{\exp(\pi_\theta(a|s)/\tau(s))}{\sum_{a'} \exp(\pi_\theta(a'|s)/\tau(s))} \tag{23}$$

The optimized model encourages exploration in high-uncertainty states and favors exploitation in low-uncertainty states, balancing the exploration-exploitation trade-off.

4.3 Limitations of Uncertainty Estimation

While Monte-Carlo Dropout provides an efficient approximation of epistemic uncertainty, three limitations remain:

4.3.1 Computational overhead

M = 10 forward passes increase inference latency by $\approx 1.8\times$ relative to a single deterministic pass;

4.3.2 Sampling error

the variance estimate scales as $1/\sqrt{M}$, yielding a $\pm 6\%$ error bound under GPU memory constraints;

4.3.3 Rare-event coverage

extreme wind/current conditions (probability ¡ 1 %) remain under-represented, potentially leading to over-confident actions.

Algorithm 1 Uncertainty-Guided Boltzmann Exploration Strategy

Require: Current state s, policy network π_θ, base temperature $\tau_{\text{base}} = 0.1$, uncertainty scaling factor $\beta = 0.5$, number of Monte Carlo samples $M = 10$, dropout rate $p = 0.3$

Ensure: Selected action a

Stage 1: Policy Distribution and Uncertainty Estimation via Monte Carlo Dropout

for $m := 1$ **to** M **do**

  $\pi_m(a|s) \leftarrow \pi_\theta(a|s, \varepsilon_m)$ ▷ $\varepsilon_m \sim \text{Dropout}(p)$

end for

$E[a|s] \leftarrow \frac{1}{M} \sum_{m=1}^{M} \pi_m(a|s)$ ▷ Expected action probability distribution

$\text{Var}[a|s] \leftarrow \frac{1}{M} \sum_{m=1}^{M} (\pi_m(a|s) - E[a|s])^2$ ▷ Epistemic uncertainty quantification $U(s)$

Stage 2: Adaptive Temperature Adjustment Based on Uncertainty

$\tau(s) \leftarrow \tau_{\text{base}} \cdot (1 + \beta \cdot \text{Var}[a|s])$ ▷ Dynamic temperature scaling

Stage 3: Action Selection via Boltzmann Exploration

Initialize action scores vector: $\mathbf{q} \leftarrow \emptyset$

for each action $a_i \in \mathcal{A}$ **do**

  $q_i \leftarrow \exp\left(E[a_i|s]/\tau(s)\right)$ ▷ Boltzmann exploration score

  $\mathbf{q} \leftarrow \mathbf{q} \cup \{q_i\}$

end for

$\mathbf{p} \leftarrow \mathbf{q} / \sum_{j=1}^{|\mathcal{A}|} q_j$ ▷ Normalize to probability distribution

$a \sim \text{Categorical}(\mathbf{p})$ ▷ Sample action from probability distribution

return a

Future work will explore variational inference and importance sampling to reduce M without loss of accuracy.

4.4 Self-Attention Mechanism PPO

A self-attention mechanism (Attention) is introduced into the PPO network to enhance the model's focus on key information. It dynamically adjusts the importance weights of state features through learnable weight matrices.

4.4.1 Attention scores obtained after feature transformation

$$e_t = \tanh(\mathbf{X}\mathbf{W} + \mathbf{b}) \tag{24}$$

Where, $\mathbf{X}$ is the input feature matrix with dimensions $B \times D$, B is the batch size and D is the feature dimension. $\mathbf{W}$ is the trainable weight matrix with dimensions $D \times D$. is the trainable bias vector with dimensions D. And $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is the activation function, constraining the output to the [-1,1] interval.

4.4.2 Attention Weight Normalization

Normalization along the feature dimension according to Eq. (25):

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{D} \exp(e_{ik})} \tag{25}$$

and $\forall i \sum_{j=1}^{D} a_{ij} = 1$.

4.4.3 Weighted Feature Output

$$x'_{ij} = x_{ij} \times a_{ij} \tag{26}$$

Where i denotes the sample index, j denotes the feature index.

4.4.4 Input Feature Space

The feature vector processed by the attention layer originates from a 14-dimensional raw state space. The feature composition is shown in Table 3.

Table 3 State Space Design

Dimension (14)		Processing
0-1	Ship Longitude, Latitude	Scaled to [-1,1] interval
2-5	Oil Spill Distance, Direction	Z-score Standardization
6-9	Other Ship Centroid Coordinates, Nearest Distance, Count	Min-max Normalization
10-11	Oil Spill Centroid Coordinates	Coordinate System
12	Operation Status Flag	Boolean (0/1)
13	Distance to Oil Spill Boundary	Logarithmic Scaling

Through feature transformation and weight normalization, the importance distribution of state features is dynamically adjusted. The weighted features are processed through fully connected layers, Dropout, etc., finally outputting action probabilities and state values. Compared to treating all features uniformly, dynamic weighting allows the model to focus more on information most relevant to the decision, thereby enhancing policy effectiveness and improving the model's robustness and adaptability in non-stationary environments.

4.5 Exploration Strategy Dynamic Optimization Mechanism Based on PSO

This paper proposes a PSO-based exploration mechanism, forming a dual-layer optimization architecture with the policy network. The fitness function is defined as:

$$f(\theta_i) = \mathrm{E}_{\tau \sim \pi_{\theta_i}}[\sum_{t=0}^{T} \gamma^t r_t] + \alpha \cdot \mathrm{Entropy}(\pi_{\theta_i}) \tag{27}$$

Where, $\alpha = 0.01$ is the entropy regularization coefficient. The particle swarm velocity update equations are:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (pbest_i - \theta_i^k) + c_2 r_2 (gbest - \theta_i^k) \tag{28}$$

$$\theta_i^{k+1} = \theta_i^k + v_i^{k+1} \tag{29}$$

Experimental results show that in Eq. (28) ,$\omega = 0.729, c_1 = c_2 = 1.494$, which guides vessels to continuously explore high-reward strategy areas and synchronizes the optimal particle parameters to the main network. This hybrid exploration strategy ensures global search capability while avoiding the inefficient random perturbations of traditional ε-greedy methods in continuous action spaces.

4.6 Reward Function and Training Parameters

This paper guides ships to quickly cover high-value areas through direction-distance-coverage multi-scale incentives, combining rewards and penalties to prevent local optima. Table 4 shows the reward-penalty mechanism.

Table 4 Reward-Penalty Mechanism

Reward/Penalty		**Weight**	**Effect**
Direction Reward	Move towards the center of mass of the oil spill	×2	Encourages ship movement towards high-density oil spill areas, reducing approach time.
Coverage Reward	Cleaning ratio	×8	Quantifies task progress, drives strategy to prioritize covering high-density areas.
Path Efficiency Reward	Straight-line distance / Actual path distance.	×5	Encourages training for the shortest path.
Distance Reward	Near the oil spill index attenuation	×5	Exponential decay as ship gets closer to oil spill core. Avoids local oscillation.
Collaboration Reward	Distance to other ships.	×3	Prevents ship clustering, promotes distributed cleanup. Balances individual and global goals.
Time Penalty	Active steps penalty.	×0.3	Avoids ineffective wandering.
Uncertainty Penalty	Sum of uncertainties	×0.5	Based on Bayesian PPO estimated policy variance. Reduces uncertainty.
Boundary Penalty	Cross the boundary	−15	Avoids ships leaving the oil spill area.

The weights in Table 4 were empirically determined based on the relative importance of each factor in the oil spill cleanup process. For instance, positive rewards for distance and coverage (e.g., +Coverage Reward×8) prioritize efficient movement and

area coverage, as these directly contribute to cleanup speed and effectiveness, while negative penalties for time or uncertainty (e.g., - Time Penalty $\times 0.3$) are set lower to discourage unsafe behaviors without overly restricting exploration. These values were tuned through iterative experiments to achieve a balance between rapid convergence and stable policy learning, as observed in the training results where higher positive weights accelerated efficiency gains.

Table 5 shows the model training parameters. The model learning rate is set to 0.0001. A low learning rate combined with a clipping mechanism ensures PPO stability, guaranteeing stable model parameter updates and gradual convergence to the optimal solution. A large discount factor adapts to the long-term nature of cleanup tasks. The number of training epochs is set to 1000, with a maximum of 200-time steps per epoch, ensuring sufficient training opportunities while avoiding excessive training time. The experience replay buffer capacity is 10,000. Mini-batches are randomly sampled for training to break data correlation and improve model generalization.

Table 5 Model Training Parameters

Parameter	Value
Learning Rate	0.0001
Discount Factor (γ)	0.99
GAE Parameter (λ)	0.95
Clipping Threshold (ε)	0.2
Training Epochs	1000
Batch Size	64
Experience Replay Capacity	10000

4.7 Advantage Analysis

The self-attention Bayesian PPO (SA-BPPO) algorithm shows computational efficiency compared to standard PPO. The introduced uncertainty penalty term effectively constrains the policy update magnitude, enabling the algorithm to exhibit superior performance and efficiency in complex environments. Table 6 shows the theoretical advantage analysis of this algorithm.

Table 6 Theoretical Advantage Analysis

Evaluation Metric	Analysis
Risk-Sensitive Decision	Quantifies policy uncertainty, avoids executing high-risk actions near oil spill boundaries.
Dynamic Adaptability	Monte Carlo Dropout provides uncertainty estimates, enabling strategies to respond quickly to abrupt environmental changes.
Computational Efficiency	Only adds K forward passes during policy inference. Training complexity remains the same order of magnitude as standard PPO.

4.8 Cleanup Strategy Algorithm Flow Based on self-attention Bayesian PPO (SA-BPPO)

The proposed self-attention Bayesian PPO (SA-BPPO) network achieves intelligent decision optimization through a three-layer architecture of attention mechanism-Bayesian PPO-PSO exploration. First, the self-attention mechanism performs weighted feature calculation on the input 14-dimensional state features, dynamically adjusting attention to input features. Second, the Bayesian PPO network performs Monte Carlo Dropout sampling to quantify epistemic uncertainty in policy output. Third, a PSO-guided exploration mechanism is introduced to break through local optima. These three components dynamically collaborate, significantly improving adaptability to the randomness of oil spill diffusion while ensuring path planning efficiency. Fig. 4 shows the flowchart of the cleanup strategy algorithm based on Bayesian PPO.

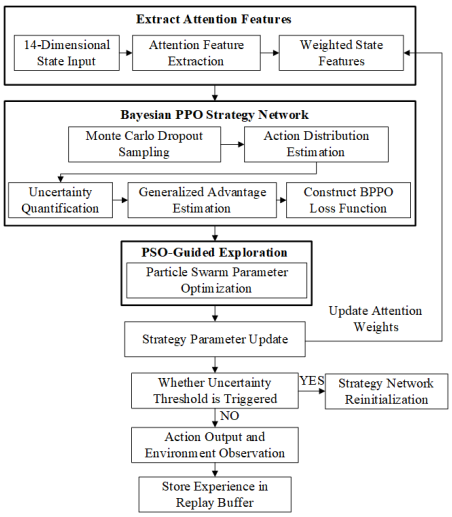


Fig. 4 Flowchart of Cleanup Strategy Algorithm Based on SA-BPPO

5 Collaborative Mechanism of Hybrid Architecture

5.1 Multi-Algorithm Collaboration Mechanism

5.1.1 Navigation Phase

Target Selection: Use DBSCAN(Density-Based Spatial Clustering of Applications with Noise) to identify oil spill clusters. Calculate the centroid as the target point. Ship speed varies dynamically based on distance. If no valid cluster exists, return to the nearest oil point or the current ship position.

Path Planning: Use the improved A* algorithm to generate an initial path, ensuring the shortest distance. Use PSO to optimize the global path, reducing tortuosity and collision risk.

5.1.2 Operation Phase

Exploration Strategy: Self-attention Bayesian PPO(SA-BPPO), based on a Boltzmann exploration strategy, efficiently cleans within the oil spill area, maximizing coverage and collaboration efficiency.

Boundary Constraint: Calculate the convex hull for each cluster identified by DBSCAN to define the precise boundary of the oil spill area, restricting ship activity. If a ship goes out of bounds, project it to the nearest boundary point.

Cleanup Logic: Ships clean oil within a dynamic radius of 0.02° (approx. 2 km). Maximum cleanup per timestep is 50 oil particles.

5.1.3 Multi-Algorithm Collaborative Process.

Fig. 5 shows the multi-algorithm collaboration flowchart.

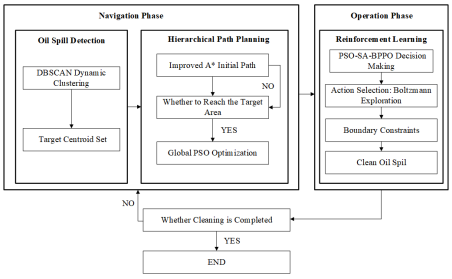


Fig. 5 Multi-Algorithm Collaboration Flowchart

5.2 Ship Segmented Speed Strategy

This paper employs an exponentially weighted average speed strategy based on oil spill distance: cruise speed when far from the spill area, medium speed for operation when moderately close, and fine cleanup speed when very close. Table 7 shows the ship dynamic speed strategy.

Table 7 Ship Dynamic Speed Strategy

Speed Mode	Oil Distance (km)	Speed (km/h)	Strategy
Long-Range Cruise	> 5	27	Large-scale rapid coverage
Medium-Range Operation	1–5	18	Medium-precision cleanup
Fine Operation	< 1	5.4	High-precision recovery

6 Model Evaluation

6.1 Evaluation Metrics

To comprehensively evaluate algorithm performance in oil spill cleanup tasks, the following key metrics are adopted:

6.1.1 Cleaning Time

The number of time steps consumed from the start of cleanup until all oil particles are cleaned. This directly reflects algorithm execution efficiency. Fewer time steps indicate faster task completion.

6.1.2 Cleaning Efficiency

Defined as the proportion of oil particles cleaned per unit timestep. This metric comprehensively considers cleanup speed and effectiveness, reflecting the algorithm's ability to balance speed and quality, defined in Eq. (30).

$$\eta_{clean} = \frac{I - N}{I \times (T + 1 \times 10^{-8})} \tag{30}$$

Where, I is the initial number of oil particles, N is the current number of oil particles, and T is the timestep length.

6.1.3 Path Efficiency

The ratio of the total length of the shortest path from the starting position to each oil point to the actual sailing path length of the ship, averaged over all ships. This measures the reasonableness of path planning. Higher path efficiency indicates better ship routes.

Path efficiency is defined as follows:

$$d_{\text{straight}} = \text{haversine}(\phi_s, \lambda_s, \phi_e, \lambda_e) \tag{31}$$

$$\bar{\eta}_{path} = \frac{d_{\text{straight}}}{d_{\text{actual}}} \quad (d_{\text{actual}} > 0) \tag{32}$$

$$\bar{\eta}_{path} = \frac{1}{n} \sum_{i=1}^{n} \eta_i \tag{33}$$

Where, (ϕ_s, ϕ_e) is latitude (in radians), (λ_s, λ_e) is longitude (in radians), n is the number of ships.

6.1.4 Remaining Particles

This indicates the number of uncleaned oil particles at the end of each training epoch. The metric reflects the thoroughness of cleanup. Fewer remaining particles indicate better cleanup effectiveness.

6.1.5 Load Balancing Index (LBI)

This metric evaluates the scalability of multi-vessel collaboration by measuring the evenness of workload distribution among vessels. It is defined as:

$$LBI = 1 - \frac{\sigma(C)}{\mu(C)} \tag{34}$$

where C = [c1, c2, ..., cn] is the vector of particles cleaned by each of the n vessels, $\sigma(C)$ is the standard deviation of C, and $\mu(C)$ is the mean of C. LBI ranges from 0 to 1, with values closer to 1 indicating better load balancing and stronger scalability in multi-agent scenarios, ensuring efficient collaboration without idle or overloaded vessels.

6.2 Experimental Platform

All experiments ran on a PC configured with an Intel Core i5-8300H processor and 16GB RAM, with a operating system of Windows 11. Experiments were conducted in a Python 3.11 environment, primarily relying on libraries such as TensorFlow 2.16.1, pandas 1.5.3, Keras 3.2.1, and scikit-learn 1.2.2 to build the algorithm model and implement the training and evaluation processes.

6.3 Experimental Results

The following experiments were conducted with 3 vessels unless otherwise specified. As shown in Fig. 6(a) (Cleaning time vs. training epochs), ship cleaning time fluctuates but gradually decreases, stabilizing at 21 timesteps, indicating steadily rising cleaning efficiency. As shown in Fig. 6(b) (Loss value vs. training epochs), the loss value gradually decreases with increasing epochs, stabilizing at 16. As shown in Fig. 6(c) (Exploration rate vs. training epochs), the exploration rate decreases from 1 to 0.1 as training epochs increase.

6.4 Comparative Experiments

6.4.1 Comparison with Reinforcement Learning Algorithms

A comparison of the improved A*-PSO-SA-BPPO algorithm DQN and SAC algorithm on various metrics is shown in Table 8.

As shown in Fig. 7(a) and Fig. 8(a), the best cleaning times for both algorithms are the same. However, because the improved A*-PSO-SA-BPPO algorithm introduces the improved A*-PSO navigation in the initial stage to guide ships directly to the oil spill area, it reduces aimless initial PPO exploration and its convergence speed to the best cleaning time is significantly faster than DQN and SAC.

As shown in Fig. 7(b) and Fig. 8(b), the cleaning efficiency of the improved A*-PSO-SA-BPPO algorithm increases with training epochs and reaches a high stable value earlier. The cleaning efficiency of DQN and SAC increases more slowly. Under the same training epochs, the average cleaning efficiency of the improved A-PSO-BPPO algorithm is 36% higher than DQN , and about 9.17% higher than SAC.

As shown in Fig. 7(c)and Fig. 8(c), although the final loss value of the improved A*-PSO-SA-BPPO algorithm is 16.10, the difference in loss values is not significant

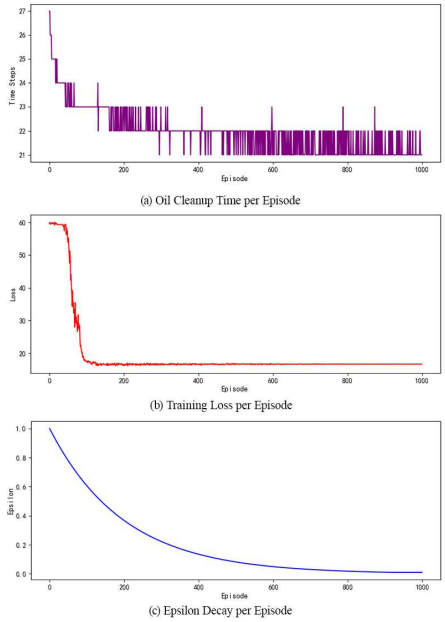


Fig. 6 Training Results with Training Epochs

Table 8 Comparison of results between improved A*-PSO-SA-BPPO, DQN and SAC

Algorithm	Min Cleaning Time	Convergence Time to Best Time	Avg Cleaning Efficiency	Min Loss Value	Loss Convergence Epochs	Avg Path Efficiency	Average LBI	Min Remaining Particles
Improved A*-PSO-SA-BPPO	21 ± 1.0	310 ± 12.0	0.0476 ± 0.0021	16.16 ± 0.85	149 ± 9.0	0.5741 ± 0.0089	0.92 ± 0.03	0
DQN	21 ± 2.0	432 ± 13.0	0.0350 ± 0.0023	10.10 ± 1.15	560 ± 15.0	0.5588 ± 0.0113	0.81 ± 0.05	0
SAC	21 ± 1.0	352 ± 11.0	0.0436 ± 0.0019	12.15 ± 0.93	258 ± 13.0	0.5632 ± 0.0074	0.87 ± 0.04	0

and slightly higher than the final loss value of 10.10 of the DQN algorithm and 12.15 of the SAC algorithm, the improved A*-PSO-SA-BPPO algorithm constrains the update amplitude of the strategy through the uncertainty penalty term (Eq. (20)). It avoids the violent oscillations that may occur in the early stage of training (see the shorter Loss Convergence Epochs in Table 8), thereby bringing a more stable and robust training process. Therefore, the Loss value of the improved A*-PSO-SA-BPPO algorithm drops rapidly in the early stage of training and reaches a stable low value in

an earlier training phase, while the Loss value of the DQN and SAC algorithm drops more slowly and has a longer convergence time.

As shown in Table 8, the average LBI of the improved A*-PSO-SA-BPPO across all rounds is 0.92±0.03, whereas the average LBI of DQN is 0.81±0.05 and that of SAC is 0.87±0.04, indicating that its multi-ship scalability has improved by 13.58% compared to DQN and by 5.74% compared to SAC.

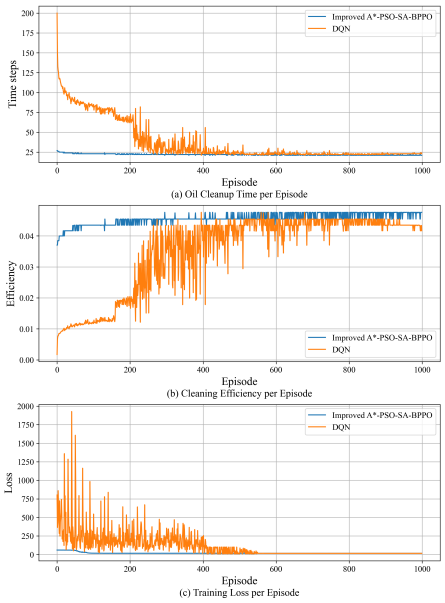


Fig. 7 Comparison of training curves between Improved A*-PSO-SA-BPPO and DQN

Comprehensive analysis of all indicators shows that the improved A*-PSO-SA-BPPO algorithm can plan ship action paths more efficiently and reasonably.

6.4.2 Comparison with Traditional Algorithms

A comparison of the improved A*-PSO-SA-BPPO algorithm with ACO and Dijkstra algorithms on various metrics is shown in Table 9. Results show that improved A*-PSO-SA-BPPO is about 8.69% faster than ACO and about 22.22% faster than Dijkstra in best cleaning time. Its average cleaning efficiency is about 36.78% higher than ACO and about 75% higher than Dijkstra, indicating it can cover and clean oil spill areas more quickly. Its average path efficiency is about 28.64% higher than ACO and about 31.22% higher than Dijkstra, indicating the paths generated are shorter and more optimal, reducing unnecessary detours and improving overall cleanup efficiency.The improved A*-PSO-SA-BPPO yields an average LBI of 0.92, outperforming ACO (0.78) and Dijkstra (0.72) by 17.95% and 27.78%, respectively, indicating superior multi-vessel load balancing.

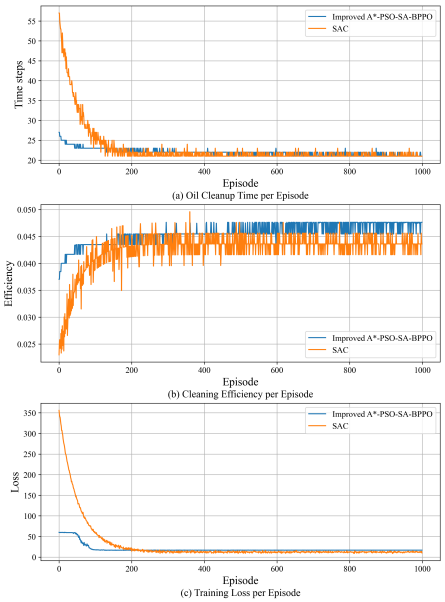


Fig. 8 Comparison of training curves between Improved A*-PSO-SA-BPPO and SAC

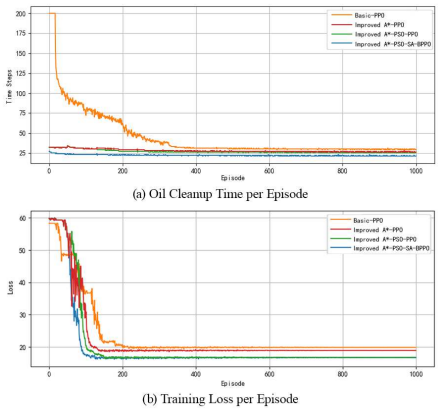
Table 9 Comparison of Improved A*-PSO-SA-BPPO with Traditional Algorithms

Algorithm	Best Cleaning Time	Avg Cleaning Efficiency	Avg Path Efficiency	Average LBI	Min Remaining Particles
Improved A*-PSO-SA-BPPO	21 ± 1.0	0.0476 ± 0.0021	0.5741 ± 0.0089	0.92 ± 0.03	0
ACO	23 ± 3.0	0.0348 ± 0.0042	0.4463 ± 0.0156	0.78 ± 0.03	0
Dijkstra	27 ± 2.0	0.0272 ± 0.0031	0.4375 ± 0.0134	0.72 ± 0.05	0

Comprehensive analysis of all indicators demonstrates the significant advantage of the improved A*-PSO-SA-BPPO algorithm in oil spill cleanup tasks.

6.4.3 Ablation Experiment

A comparison of the improved A*-PSO-SA-BPPO algorithm variants in the ablation experiment on various metrics is shown in Table 10. When adding the improved A* navigation module relative to basic PPO, cleaning time decreased by about 10.34%, average cleaning efficiency increased by about 33.43%, average path efficiency increased by 6.73%, and loss convergence time decreased by 35 epochs. Similarly, adding PSO



(a) Oil Cleanup Time per Episode

(b) Training Loss per Episode

Fig. 9 Comparison of Ablation Experiment Results vs. Training Epochs

parameter optimization to the improved A*-PPO reduced cleaning time by about 7.69%, increased average cleaning efficiency by about 4.51%, increased average path efficiency by 3.55%, and reduced loss convergence time by 8 epochs. Adding the attention Bayesian module relative to the improved A*-PSO-PPO reduced cleaning time by about 12.5%, increased average cleaning efficiency by about 2.81%, increased average path efficiency by 1.04%, and reduced loss convergence time by 13 epochs.

Table 10 Ablation Experiment Results for Improved A*-PSO-BPPO

Algorithm	Best Cleaning Time	Avg Cleaning Efficiency	Min Loss Value	Loss Convergence Epochs	Avg Path Efficiency	Average LBI	Min Remaining Particles
Basic PPO	29 ± 4.0	0.0332 ± 0.0031	19.46 ± 2.12	205 ± 18.0	0.5141 ± 0.0121	0.75 ± 0.05	2 ± 1
Improved A*-PPO	26 ± 2.0	0.0443 ± 0.0032	18.48 ± 1.95	170 ± 11.0	0.5487 ± 0.0112	0.84 ± 0.05	0 ± 0
Improved A*-PSO-PPO	24 ± 2.0	0.0463 ± 0.0025	16.34 ± 1.42	162 ± 11.0	0.5682 ± 0.0092	0.88 ± 0.03	0 ± 0
Improved A*-PSO-SA-BPPO	21 ± 1.0	0.0476 ± 0.0021	16.15 ± 0.85	149 ± 9.0	0.5741 ± 0.0089	0.92 ± 0.03	0 ± 0

As shown in Figs. 9(a) and 9(b), basic PPO requires the model to freely explore to find the oil spill area before starting cleanup, resulting in a longer convergence time. Adding the improved A* navigation algorithm avoids this aimless free exploration from the start point to the oil area, significantly reducing initial cleaning time. As the PSO module is responsible for path smoothing, adding it improves path efficiency and slightly reduces cleaning time. The attention module helps the model focus on important features, so adding it improves the loss value, cleaning time, and convergence time. Adding the improved A*-PSO module increases LBI from 0.75 (basic PPO)

to 0.88 (+17.33%); further adding SA-BPPO raises it to 0.92 (+4.55%), confirming enhanced multi-agent scalability.

6.5 Multi-Scenario Oil Spill Emergency Scheduling Robustness Test

To verify the generalization capability of SA-BPPO under varying spill conditions, we conducted supplementary experiments with three additional particle counts—400, 1000 and 1500—while simultaneously adjusting diffusion radius (1.0–3.0 km) and current velocity (0.8 to 2.0 wind speed).

Table 11 summarizes the results averaged over five random seeds for each setting. Across all scenarios, SA-BPPO maintains stable performance:

Cleanup time ranges from 18 ± 0.0 (400 particles, low diffusion) to 29 ± 2.0 (1500 particles, high diffusion), demonstrating linear scalability with particle count.Average cleaning efficiency η_{clean} remains above 0.045.Path efficiency η_{path} exceeds 0.55 even under the highest current velocity, indicating robust trajectory planning. LBI remains above 0.89 across scenarios, validating robust multi-vessel scalability under varying spill sizes.

Table 11 Robustness Test Results

Particles	Initial Leakage Radius (km)	Multiple of wind speed	Avg Cleanup Time	Min Loss Value	Avg Cleaning Efficiency	Avg Path Efficiency	Average LBI	Min Remaining Particles
400	1.0	0.8 wind speed	18 ± 0.0	16.13 ± 0.85	0.0495 ± 0.0019	0.5782 ± 0.0073	0.91 ± 0.02	0
1000	2.5	1.5 wind speed	24 ± 1.0	16.25 ± 0.72	0.0479 ± 0.0022	0.5742 ± 0.0065	0.90 ± 0.03	0
1500	3.0	2.0 wind speed	29 ± 2.0	16.26 ± 0.96	0.0482 ± 0.0017	0.5756 ± 0.0068	0.89 ± 0.04	0

6.6 Simulation Based on Forecasted Oceanographic Data

To address the latency inherent in re-analysis fields, we repeated the experiment using forecast data from 1 June 2025 00:00 UTC to 10 June 2025 00:00 UTC: Wind field data: The forecast wind field data provided by the Global Forecast System (GFS) of the National Center for Environmental Prediction (NCEP) were used with a temporal resolution of 6 hours and a spatial resolution of 0.25°×0.25°, lead time is 12 h.

Flow field data: The Global Ocean Physics Analysis and Forecast provided by Copernicus Marine Environment Monitoring Service (CMEMS) of the European Union was used to forecast the flow field data, with a temporal resolution of 1 day and a spatial resolution of 0.083°×0.083°.

The same SA-BPPO agent (weights frozen from Section 2.2) was deployed without retraining. After initializing 800 oil droplets at 35.9° N, 120.6° E, the cleaning time

of the agent reaches 22 ± 1 step. The average clearance efficiency (η_{clean}) reached 0.048±0.002, indicating that the proposed algorithm could effectively adapt to the uncertainty of prediction data. The path efficiency (η_{path}) is 0.56±0.01. Ships can dynamically adjust the path to cope with the predicted drift trend of oil pollution. The multi-ship cooperation mechanism remains efficient in the forecast data scenario, and there is no path conflict or repeated cleaning phenomenon.

The experimental results verify the effectiveness and robustness of the proposed method under the support of real-time ocean forecast data, which provides a basis for the application of the algorithm in the actual emergency system.

6.7 Emergency Resource Scheduling Validation in Real Oil Spill Accidents: Sanchi Collision (6 Jan 2018, 30°51.1' N, 124°57.6' E, East China Sea)

On 6 January 2018 the tanker Sanchi collided with CF Crystal and released ≈136,000 t (≈ 160,000 m^3) of condensate and bunker fuel into the East China Sea. To demonstrate the practical applicability of SA-BPPO, we simulated the emergency response to the Sanchi collision using the same re-analysis wind (NCEP-NCAR) and current (Global Ocean Physics Reanalysis GLORYS12V1) datasets as in Section 2.2.

Initialisation: 2000 oil particles were seeded at 30°51.1' N, 124°57.6' E at 20:00 UTC 6 Jan 2018, matching the reported spill centroid.

Resources: three emergency vessels dispatched from Zhoushan (30.0° N, 122.1° E), Shanghai (31.2° N, 121.5° E) and Ningbo (29.9° N, 121.6° E).

Within the 72-hour forecast window, SA-BPPO achieved:

cleanup time = 42 ± 2 steps;

average cleaning efficiency = 0.0463 ± 0.0012;

These results confirm that the proposed framework can handle real-scale incidents under actual meteorological and oceanographic conditions without retraining.

7 Conclusion

In the era of global industrialization, the risks and complexity of marine oil spill accidents are increasing. The improved A*-PSO-SA-BPPO algorithm proposed in this paper provides a new solution for marine oil spill emergency resource scheduling. In the model design, by combining improved A-PSO path planning with self-attention Bayesian PPO (SA-BPPO) reinforcement learning, a full-chain oil spill cleanup framework from dynamic path planning to efficient cleanup is achieved. Experimental results show that this model outperforms traditional algorithms in multiple key metrics, validating its effectiveness in collaborative scheduling of marine oil spill emergency resources. Compared to the DQN algorithm, this algorithm achieves the same cleaning time but significantly faster convergence, with a 36% improvement in average cleaning efficiency. This is mainly attributed to the effectiveness of the improved A*-PSO navigation module in reducing initial exploration, accelerating the learning process. Compared to SAC, our algorithm improves average cleaning efficiency by 9.17% , demonstrating better adaptability to dynamic oil spill diffusion through the integration of path planning and uncertainty-aware optimization. Compared to traditional algorithms

like ACO and Dijkstra, this algorithm is 8.69% and 22.22% faster in cleaning time, improves average cleaning efficiency by 36.78% and 75%, and improves path efficiency by 28.63% and 31.22%, respectively. Experiments adding modules individually show that adding the improved A* module reduces cleaning time by 10.34%, adding the PSO module reduces it by 7.69%, and adding the attention Bayesian module reduces it by 12.5%. Additionally, the high LBI (0.92 on average) confirms the framework's scalability in multi-vessel deployments. These results indicate the distinct benefits of the hybrid architecture.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China under Project No. 61971444; in part by the Innovative Research Team Program for Young Scholars at Universities in Shandong Province under Project 2020KJN010 and in part by China University of Petroleum (East China) Graduate Education and Teaching Reform Project YJG2023041.

Declarations.

- **Funding:** This work was partially supported by National Natural Science Foundation of China under Project No. 61971444; in part by the Innovative Research Team Program for Young Scholars at Universities in Shandong Province under Project 2020KJN010 and in part by China University of Petroleum (East China) Graduate Education and Teaching Reform Project YJG2023041.
- **Conflict of interest/Competing interests:** The authors declare that there are no financial interests, commercial affiliations, or other potential conflicts of interest that could have influenced the objectivity of this research or the writing of this paper.
- **Ethics approval and consent to participate:** Not applicable.
- **Consent for publication:** Not applicable.
- **Data availability:** The data that support the findings of this study are available from the corresponding author upon reasonable request.
- **Materials availability:** Not applicable.
- **Code availability:** The code that supports the findings of this study is available from the corresponding author upon reasonable request.
- **Author contribution:** Not applicable.

References

Chai Y, Han X, Wang Y, et al (2025) Transoilseg: A novel SAR oil spill detection method addressing data limitations and look-alike confusions. IEEE Transactions on Geoscience and Remote Sensing 63:1–16. https://doi.org/10.1109/TGRS.2025.3542810

Chen Z (2020) Research on the optimization of emergency materials scheduling in maritime oil spill accident. PhD thesis, Dalian Maritime University

Deraj R, Kumar RS, Alam MS, et al (2023) Deep reinforcement learning based controller for ship navigation. Ocean Engineering 273:113937. https://doi.org/10.

1016/j.oceaneng.2023.113937

Ding F, Zhang Z, Fu M, et al (2018) Energy-efficient path planning and control approach of USV based on particle swarm optimization. In: OCEANS 2018 MTS/IEEE Charleston, pp 1–6, https://doi.org/10.1109/OCEANS.2018.8604920

Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. arXiv preprint arXiv:150602142 https://doi.org/arXiv:1506.02142

Gong M (2021) Research on path planning for ship intelligent navigation. PhD thesis, Wuhan University of Technology

Guan W, Cui Z, Luo W (2023) Ship autonomous collision avoidance decision based on improved PPO algorithm. Journal of Dalian Maritime University 49(4):28–36

Guo S (2021) Research on coastal ship path planning based on deep reinforcement learning. PhD thesis, Dalian Maritime University

Huang L, Peng X, Zhang F, et al (2019) A cloud-based platform for marine oil spill emergency response capability visualization. In: 2019 5th International Conference on Transportation Information and Safety (ICTIS), pp 810–815, https://doi.org/10.1109/ICTIS.2019.8883807

Kuzmenko HC, Ehevich D, Gal O (2025) Autonomous oil spill response through liquid neural trajectory modeling and coordinated marine robotics. arXiv preprint arXiv:250812456 https://doi.org/arxiv-2508.12456

Li H, Li X, Zhang H, et al (2024a) End-to-end autonomous navigation method for drones based on AR-PPO. Journal of Beijing University of Aeronautics and Astronautics https://doi.org/10.13700/j.bh.1001-5965.2024.0582

Li J, Xiao D, Wen X, et al (2023) Coordinated optimization method for feeder container ship route planning and stowage based on DQN algorithm. Journal of Transport Information and Safety 41(6):132–141

Li Y, Lyu X, Ren P (2024b) Oil spill timely backtracking oriented by wind field correction with self-attention temporal convolutional networks. IEEE Journal of Oceanic Engineering 49(1):114–132. https://doi.org/10.1109/JOE.2023.3314101

Liu X, Li Y, Zhang J, et al (2019) Self-adaptive dynamic obstacle avoidance and path planning for USV under complex maritime environment. IEEE Access 7:114945–114954. https://doi.org/10.1109/ACCESS.2019.2935964

Ning J, Huang Y, You Y, et al (2023) Ship collision avoidance decision based on hybrid particle swarm algorithm. Journal of Dalian Maritime University 49(1):10

Ou C, Xie L, Cha T, et al (2024) Ship path planning based on deep reinforcement learning and historical trajectories. Navigation of China 47(1):36–44

Ozair M, Iqbal MF, Mahmood I, et al (2024) SAR-based oil spill detection and impact assessment on coastal and marine environments. Acta Oceanologica Sinica 43:123–140. https://doi.org/10.1007/s13131-024-2386-8

Panda A, Mallipeddi R, Das S (2017) Particle swarm optimization with a modified learning strategy and blending crossover. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp 1–8, https://doi.org/10.1109/SSCI.2017.8285235

Pu H, Huang H (2018) Global path planning algorithm for unmanned surface vehicle. Marine Sciences 42(1):95–107

Ren P, Jia Q, Xu Q, et al (2025) Oil spill drift prediction enhanced by correcting numerically forecasted sea surface dynamic fields with adversarial temporal convolutional networks. IEEE Transactions on Geoscience and Remote Sensing 63:1–18. https://doi.org/10.1109/TGRS.2025.3528631

Shi L, Wei X, Yang K, et al (2024) A few-shot oil spill segmentation network guided by multi-scale feature similarity modeling. Frontiers in Marine Science 11. https://doi.org/10.3389/fmars.2024.1481028

Vo AH, Yoon HK, Ryu J, et al (2019) Modified A* algorithm for obstacle avoidance for unmanned surface vehicle. Journal of Ocean Engineering and Technology 33(6):510–517. https://doi.org/10.26748/KSOE.2019.067

Xin J, Zhong J, Yang F, et al (2019) An improved genetic algorithm for path-planning of unmanned surface vehicle. Sensors 19(11):2640. https://doi.org/10.3390/s19112640

Yu J, Wang B, Song D (2025) Simulation study of GNSS-R-based marine oil spill detection considering sea surface asymmetry. Journal of Computer and Communications 13:234–250. https://doi.org/10.4236/jcc.2025.136016

Zhai P (2023) Research on intelligent ship collision avoidance algorithm in complex waters based on improved deep Q network. Master's thesis, Dalian Maritime University

Zhang H, Tao Nr, Yang N (2022) Emergencies materials dispatching of offshore oil spill with time-varying properties. Journal of Transportation Systems Engineering and Information Technology 22(6):12

Zhu F, Niu Y, Wei M, et al (2025) A high-risk test scenario adaptive generation algorithm for ship autonomous collision avoidance decision-making based on reinforcement learning. Ocean Engineering 320:120344. https://doi.org/10.1016/j.oceaneng.2025.120344

Zhu G, Yang K, Zhao Q, et al (2019) Bi-objective optimal planning for emergency resource allocation in the maritime oil spill accident response phase under uncertainty. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. Association for Computing Machinery, New York, NY, USA, pp 93–94, https://doi.org/10.1145/3319619.3326780

Biographies. **Jihao An** received the B.Eng. degree in Electronic Information Engineering from China University of Petroleum (East China), Qingdao, China, in 2023. He is currently pursuing the M.S. degree in New-Generation Electronic Information Technology, China University of Petroleum (East China). His research interests include machine learning and deep learning, with applications to remote sensing.

Peng Ren received the B.Eng. and M.Eng. degrees in electronic engineering from the Harbin Institute of Technology, Harbin, China, in 2003 and 2005, respectively, and the Ph.D. degree in computer science from the University of York, York, U.K., in 2010. He is currently a Professor with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China. His research interests include remote sensing and machine learning. Dr. Ren was a recipient of the K. M. Scott Prize from the University of York in 2011 and the Eduardo Caianiello Best Student Paper Award at the 18th International Conference on Image Analysis and Processing in 2015 (as the coauthor).

Xinrong Lyu received the B.Eng. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2004 and 2009, respectively. He is currently an Associate Professor with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao. His research interests include signal detection and data processing in ocean remote sensing field.

Christos Grecos received his B.Sc. degree in Coomputer Science from Concordia University in Montreal in 1994 and his Msc degree in HCI from Hariot Watt University in 1995. In June 2001, he completed his PhD degree at the School of Computing Glamorgan University. He is currently the Professor and the Chair of Computer Science Department at Arkansas State University and a Senior Technical Consultant at Strathclyde Forensics Ltd. His research interests include distributed computing, computer graphics and computer communications (networks).