AUV Efficient Navigation Relying on Adaptive Proximal Policy Optimization

Jingzehua Xu[1,*], Yongming Zeng[2,*], Jintao Zhang[1], Xuanchen Li[1], Lingru Meng[2], Haocai Huang[2], Jingjing Wang[3(✉)], and Yong Ren[4]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2] Ocean College, Zhejiang University, Zhoushan, China
[3] School of Cyber Science and Technology, Beihang University, Beijing, China
drwangjj@buaa.edu.cn
[4] Department of Electronic Engineering, Tsinghua University, Beijing, China

Abstract. Safe and efficient navigation is crucial for autonomous underwater vehicles (AUVs) to perform various marine monitoring tasks. Considering the complex and unknown underwater environment and limited sensing ability of AUVs, the traditional methods based on models and relying on large amounts of input information are not practical enough, and reinforcement learning (RL) has been widely discussed as one of the most promising schemes. Among many RL algorithms, the proximal policy optimization (PPO) based on trust region optimization theory not only improves sampling efficiency but also reduces deployment complexity by constraining updates of new and old policies within an alternate trust region. However, the performance of PPO is easily influenced by fixed clipping bounds and lacks adaptability. In order to dynamically optimize clipping bounds, we propose the adaptive PPO (APPO) algorithm for AUV navigation tasks. APPO dynamically explores and exploits clipping bounds during online training using a bandit to maximize the value of the upper confidence bound of each candidate boundary, guiding PPO to use different clipping bounds at different stages of online training to improve training efficiency and stability. Extensive simulation experiments demonstrate that APPO is more suitable for AUV navigation tasks compared to other baseline algorithms, showing superior performance in terms of robustness, stability, and adaptability. To accelerate relevant research in this direction, the code for simulation will be released as open-source.

Keywords: Autonomous underwater vehicles · Efficient navigation · Adaptive proximal policy optimization · Multi-armed bandit.

1 Introduction

Autonomous underwater vehicles (AUVs) have greatly promoted the progress of marine science, playing a crucial role in the fields of seabed mapping, resource

* These authors contributed equally to this work.

survey, information collection and so on [8]. Efficient and robust navigation is the key to ensure that AUVs can complete tasks safely and efficiently in complex and unknown environments. However, conducting efficient and safe navigation, given constraints of limited prior environmental knowledge and the AUVs' restricted sensing capabilities, presents practical challenges and holds great value [2].

In the past few decades, the navigation methods used for various unmanned vehicles can be roughly divided into graph-based methods represented by Dijkstra algorithm and A-star Algorithm, sample-based methods represented by probabilistic roadmap and rapid exploration random tree algorithm, and artificial intelligence methods represented by ant colony optimization algorithm, genetic algorithm and neural networks [2]. Considering the complexity of the underwater environment, the above methods cannot be directly transferred to the AUVs. Therefore, the performance of the above traditional navigation methods in the ocean current environment is compared and analyzed in [12]. To solve the problem of slow convergence speed and poor effect of heuristic algorithms, Wen et al. proposed a fusion heuristic algorithm for AUV navigation under ocean current interference by integrating genetic algorithm, simulated annealing algorithm and ant colony optimization algorithm [10]. Gong et al. established a multi-trajectory planning model based on ant colony optimization and comprehensively considered constraints such as underwater environment and motion efficiency to deduce multiple alternative trajectories in order to select the best scheme [3]. Unfortunately, such model-based control methods not only rely on prior environmental information but also need to perform parameter tuning. In the face of unknown and dynamic environments, they suffer from high time complexity and computational complexity, lack generalization and learning ability, and have limited application scenarios.

Reinforcement learning (RL) is widely used in AUV navigation because it can optimize the decision-making of the agent by interacting with the environment, so as to cope with the dynamic changing environment without the need for an exact environment model [1]. However, with the increase of task complexity and environment state dimension, traditional RL algorithms will face dimension disaster due to insufficient memory. Deep reinforcement learning (DRL) combined with deep neural network is used to solve these problems and has achieved satisfactory results in various fields [4]. In [5], the author proposed an obstacle avoidance algorithm based on DRL based on the obstacles detected by sonar to ensure the safe navigation of AUVs in complex environments. In [11], the author designed an actor-critic structure optimal adaptive distributed controller based on DRL for end-to-end AUV motion planning and formation control. Other studies have applied DRL to AUV trajectory tracking, autonomous positioning, target hunting and other applications, all of which demonstrate the superior performance of DRL [9] [6] [7]. Although remarkable progress has been made in AUV navigation, DRL algorithms still face problems such as slow convergence, unstable training, and low learning efficiency [2]. Among DRL algorithms, trust region policy optimization (TRPO) improves stability and ensures monotony convergence by limiting the update of new policies to one trust region. Proximal

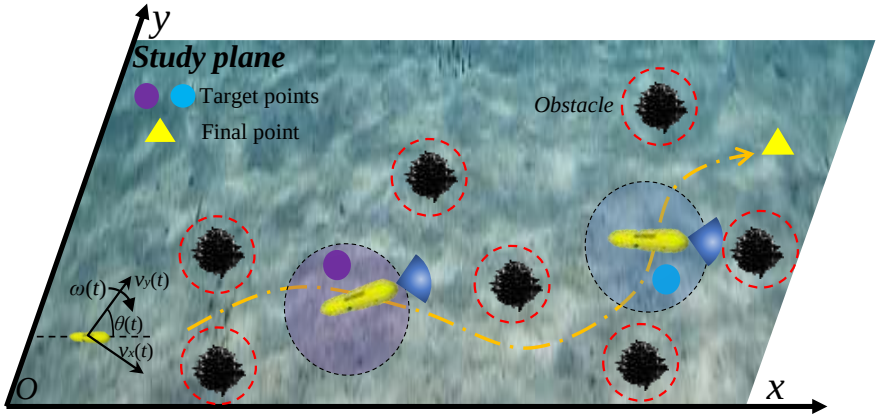


Fig. 1. Illustration of the AUV navigation system model in the obstacle environment, which consists of two main models, AUV dynamics model, and underwater detection model. The AUV is working to complete the navigation task while avoiding obstacles.

policy optimization (PPO) based on TRPO reduces deployment complexity by limiting the updating of old and new policies to an alternative trust region. However, the fixed clipping bound limits the performance of PPO, and PPO cannot adjust the conservative degree of its strategy update according to the current learning situation. Therefore, it is very beneficial to explore and study dynamic clipping bounds to improve PPO performance.

Based on the above analysis, this paper proposes a novel DRL method named Adaptive PPO (APPO) for AUV navigation. APPO features an adaptive clipped trust region mechanism that dynamically adjusts the clipping bounds via bandit during online training. By maximizing the upper confidence bound (UCB) value of each candidate bound, PPO is guided to use different cut bounds in different stages of online training, so as to select the best cut bounds in each stage to improve the algorithm performance. Simulation results show that our proposed APPO can not only adapt to navigation tasks in different scenarios, but also has higher learning rate and stability compared with other baseline algorithms. To accelerate relevant research in this direction, the code for simulation will be released as open-source.

The remainder of this paper is structured as follows: Section 2 introduces the system model of the AUV navigation task. Section 3 introduces the problem formulation, while Section 4 introduces the methodology, mainly including the APPO algorithm design and its principle modules. Section 5 presents experiments and comparisons, followed by the conclusions drawn in Section 6.

2 System Model

The AUV navigation model we considered is shown in Fig. 1, and the AUV conducts the navigation task in a two-dimensional plane with a fixed depth.

During the process, the AUV can obtain the location of the target points through various underwater sensors, and detect the environment through sonar to avoid obstacles in order to reach the target points safely. The task ends when the AUV reaches the final point. The AUV motion model and the underwater detection model are described in detail in Section 2.1 and Section 2.2, respectively.

2.1 AUV Motion Model

Without loss of generality, we consider a three-degree-of-freedom motion model for AUV navigation tasks. At time t, the AUV's body frame reference is represented by $\boldsymbol{v} = [v_x(t), v_y(t), \omega(t)]^{\mathrm{T}}$, where $v_x(t)$, $v_y(t)$, $\omega(t)$ are surge velocity, sway velocity and yaw angular velocity, respectively. And the world reference frame at time t can be represented by $\boldsymbol{\eta} = [x(t), y(t), \theta(t)]^{\mathrm{T}}$, where $x(t)$ and $y(t)$ indicate the position of the AUV, while $\theta(t)$ denotes the yaw angle. According to Fossen's motion [5], the motion model of AUV considering hydrodynamics and hydrostatic forces is

$$\dot{\boldsymbol{\eta}}(t) = \boldsymbol{J}\left(\boldsymbol{\eta}(t)\right) \cdot \boldsymbol{v}(t), \tag{1}$$

$$\boldsymbol{M}_A \dot{\boldsymbol{v}}(t) + \boldsymbol{C}_A\left(\boldsymbol{v}(t)\right) \cdot \boldsymbol{v}(t) + \boldsymbol{D}_A\left(\boldsymbol{v}(t)\right) \cdot \boldsymbol{v}(t) + \boldsymbol{G}_A\left(\boldsymbol{\eta}(t)\right) = \boldsymbol{\tau}(t), \tag{2}$$

where $\boldsymbol{M}_A$, $\boldsymbol{C}_A$, $\boldsymbol{D}_A$ and $\boldsymbol{G}_A$ represent the inertia matrix with the added mass of the AUV, the Coriois centripetal force matrix, the damping matrix for the viscous fluid force and the composite matrix of gravity and buoyancy, respectively. Moreover, $\boldsymbol{\tau}(t)$ is the control input, while $\boldsymbol{J}\left(\boldsymbol{\eta}(t)\right)$ stands for the transformation matrix, which can be defined as

$$J\left(\boldsymbol{\eta}(t)\right) = \begin{bmatrix} \cos\theta(t) & -\sin\theta(t) & 0 \\ \sin\theta(t) & \cos\theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

According to practical application, above equations need to be discretized as

$$\boldsymbol{\eta}(t+1) = \boldsymbol{\eta}(t) + \Delta T \cdot \boldsymbol{J}\left(\boldsymbol{\eta}(t+1)\right) \cdot \boldsymbol{v}(t), \tag{4}$$

$$\boldsymbol{v}(t+1) = \boldsymbol{v}(t) + \Delta T \cdot \boldsymbol{M}_A^{-1} F\left(\boldsymbol{\eta}(t), \boldsymbol{v}(t)\right), \tag{5}$$

where ΔT is the time interval, and $F\left(\boldsymbol{\eta}(t), \boldsymbol{v}(t)\right)$ can be calculated by

$$F\left(\boldsymbol{\eta}(t), \boldsymbol{v}(t)\right) = \boldsymbol{\tau}(t) - \boldsymbol{C}_A\left(\boldsymbol{v}(t)\right) \cdot \boldsymbol{v}(t) - \boldsymbol{D}_A\left(\boldsymbol{v}(t)\right) \cdot \boldsymbol{v}(t) - \boldsymbol{G}_A\left(\boldsymbol{\eta}(t)\right). \tag{6}$$

2.2 Underwater Detection Model

During the navigation process, the AUV uses the sonar to detect the environment, including obstacles and target points, which can be modeled using the active sonar equation [9]

$$EM = SL - 2TL(f, d) + TS - NL(f) + DI - DT, \tag{7}$$

where SL, TL, TS, NL and DI are the emission sound strength, transmission loss, target strength, environmental noise level and directionality index of target, respectively [2]. Additionally, DT and EM represent the detection threshold and echo margin, respectively. Furthermore, TL is related to the detection distance d and the center acoustic frequency f, which can be expressed as

$$TL = 20\log(d) + d \times a(f) \times 10^{-3}, \tag{8a}$$

$$a(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75 \times 10^{-4} f^2 + 0.003, \tag{8b}$$

where $\alpha(f)$ is the attenuation coefficient of sound wave in water, and the maximum detection radius r_m of the AUV can be determined by considering the monotonically decreasing relationship between EM and the detection distance d, and we have

$$r_m = \arg\max_d \{EM(d) \geq 0\}. \tag{9}$$

3 Problem Formulation

We describe the AUV navigation problem as a Markov decision process (MDP), which can be defined by a quintuple, i.e.

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}(\cdot \mid s(t), a(t)), \mathcal{R}, \gamma), \tag{10}$$

where $\mathcal{S}$ and $\mathcal{A}$ denotes the state and action space of the AUV, respectively. Moreover, γ is the discount factor, while $\mathcal{P}(\cdot \mid s(t), a(t))$ represents the state transition probability function. To be intuitive, at time t, AUV selects the action $a(t) \in \mathcal{A}$ according to its policy π_θ by observing the current state $s(t) \in \mathcal{S}$, and transitions to the next state $s(t+1) \sim \mathcal{P}(\cdot \mid s(t), a(t))$ and gets the reward $r(t) \in \mathcal{R}$. The details are as follows:

State space: In the navigation task, the observation space of AUV at time t is $s(t)$, which can be defined as

$$s(t) = [\boldsymbol{l}(t), l_{A\leftrightarrow T}(t), \theta(t), \phi_{A\leftrightarrow T}(t), \chi(t)], \tag{11}$$

where $\boldsymbol{l}(t)$ contains the distances detected by sonar between the AUV and various obstacles, while $l_{A\leftrightarrow T}(t)$ represents the distance between the AUV and the target point. $\theta(t)$ and $\phi_{A\leftrightarrow T}(t)$ respectively indicate the orientation angle (yaw angle) of the AUV and the angle between the AUV and the target point. Furthermore, $\chi(t) \in \{0, 1\}$, and $\chi(t) = 1$ indicates the current training episode has concluded, while vice versa.

Action space: In the process of navigation task, the AUV makes action $a(t)$ at time t by observing the state $s(t)$ and action $a(t)$, which can be given by

$$a(t) = [v(t), \omega(t)], \tag{12}$$

where $\|v(t)\| = \sqrt{v_x(t)^2 + v_y(t)^2}$ and $\|\omega(t)\|$ indicate the linear and angular velocity of the AUV, respectively. And the AUV can adjust its own motion state by changing its linear and angular velocity.

Reward function: We need to design the corresponding reward function to guide the AUV to make reasonable decisions in the complex environment to optimize the navigation trajectory to safely complete the navigation task. The rewards received by the AUV at time t consist of the following parts

$$r_c(t) = -500\,\mathrm{ceil}\left(l_{\mathrm{safe}}/\min\left(\boldsymbol{l}(t)\right)\right), \tag{13}$$

$$r_g(t) = 1000\,\mathrm{ceil}\left(l^{\max}_{A\leftrightarrow T}/l_{A\leftrightarrow T}(t)\right), \tag{14}$$

$$r_e(t) = -0.2 + 5(l_{A\leftrightarrow T}(t-1) - l_{A\leftrightarrow T}(t)) + 2(\phi_{A\leftrightarrow T}(t-1) - \phi_{A\leftrightarrow T}(t)), \tag{15}$$

where $r_c(t)$ is a penalty term used to prevent the AUV from colliding with the obstacles, while l_{safe} is the safe distance between the AUV and the obstacles, and ceil$(\cdot)$ is the integer up function. Additionally, when the AUV detects the target point for the first time, it receives a reward $r_g(t)$. In addition, we use the reward item $r_e(t)$ to encourage the AUV to get closer to the target point. Therefore, the total reward available for AUV at time t can be weighted by

$$r(t) = \delta_c r_c(t) + \delta_g r_g(t) + \delta_e r_e(t), \tag{16}$$

where δ_c, δ_g and δ_e are the weights of each reward or penalty item, respectively, which can be adjusted according to the application needs.

Based on the above analysis, we summarize several engineering constraints that need to be considered during the actual navigation process, and formulate a constraint optimization problem whose goal is to optimize the policy of the AUV to maximize the total expected return. The constrained optimization problem can be expressed as

$$\max_{\pi_\theta} J(\theta) = \max_{\pi_\theta} E\left[\sum_{t'=t}^{T=\infty} \gamma^{t'-t} r_{t'}\left(s(t), \pi_\theta\left(a(t) \mid s(t)\right)\right)\right], \tag{17a}$$

$$s.t.\ \ \min\left(\boldsymbol{l}(t)\right) \geq l_{\mathrm{safe}}, \tag{17b}$$

$$s.t.\ l_{A\leftrightarrow T}(t) \leq l^{\max}_{A\leftrightarrow T}, \tag{17c}$$

$$v_{\min} \leq \|v(t)\| \leq v_{\max}, \omega_{\min} \leq \|\omega(t)\| \leq \omega_{\max}, \tag{17d}$$

where Eq. (17a) denotes the optimization objective, and Ineq. (17b) represents the constraint that prevents the AUV from colliding with obstacles. Moreover, Ineq. (17c) stands for the constraint that ensures the AUV to get to the target point, while Ineq. (17d) restricts the velocity and angular velocity range of the AUV.

4 Methodology

In this section, we mainly introduce the principals for APPO, which consists of three main modules, trust region optimization, multi-armed bandit and upper confidence bound, and sampling clipping bound with alternate uncertainty term. Based on the modules, we finally present the pseudo-code of the APPO algorithm in detail.

4.1 Trust Region Optimization

Importance Sampling. Algorithms such as TRPO and PPO employ the approach of importance sampling to transform on-policy algorithms into approximations of off-policy algorithms. This adaptation allows for the utilization of collected data in the training of the current policy through the application of the policy gradient method, i.e.

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}}[\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}}], \tag{18}$$

where τ is the collected data, π_{old} represents the old policy, while π_{cur} denotes the current policy. In addition, $A^{\pi_{\text{old}}}$ stands for the advantage function, whose value is determined by state, action and π_{old} of the AUV.

KL Divergence and Trust Region Optimization. The expression in Eq. (18) could result in the deviation of the new policy from old policy, complicating the attainment of the optimal solution. Consequently, it becomes essential to reduce the Kullback-Leibler (KL) divergence between the new and old policies, denoted as $D_{\text{KL}}(\pi_{\text{cur}}||\pi_{\text{old}})$, thereby limiting the updates to the policy to remain within a designated trust region

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}}[\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}} - D_{\text{KL}}(\pi_{\text{cur}}||\pi_{\text{old}})]. \tag{19}$$

Furthermore, we can also directly constrain the updates between the new and old policies within a fixed trust region, thus Eq. (19) can be translated into

$$\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau \sim \pi_{\text{old}}}[\min(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}}, \text{clip}(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}}, 1-\epsilon, 1+\epsilon) A^{\pi_{\text{old}}})], \tag{20}$$

where ϵ is the clipping bound controling the range for policy updating.

4.2 Multi-Armed Bandit and Upper Confidence Bound

Considering n independent variables $\mathcal{T} = \{\epsilon_0, \epsilon_1, \cdots, \epsilon_i, \cdots, \epsilon_n\}$ that are identically distributed, we model the process of sampling from these clipping bounds as a multi-armed bandit game. Upon sampling the i-th clipping bound ϵ_i, an immediate reward $r_{t=N_{\epsilon_i}}$ is obtained, where N_{ϵ_i} represents the cumulative number of times ϵ_i has been accessed, Subsequently, we can compute the expected return as $\mathbb{E}[R_i|\epsilon_i]$, and we have

$$U(\epsilon_i) = \mathbb{E}[R_i|\epsilon_i] = \sum_{t=0}^{t=N_{\epsilon_i}} \gamma^t r_t(\epsilon_i). \tag{21}$$

In the procedure of sampling from $\mathcal{T}$ and updating $\mathbb{E}[R_i|\epsilon_i]$, opting for the most promising clipping bound based on the highest expected return during

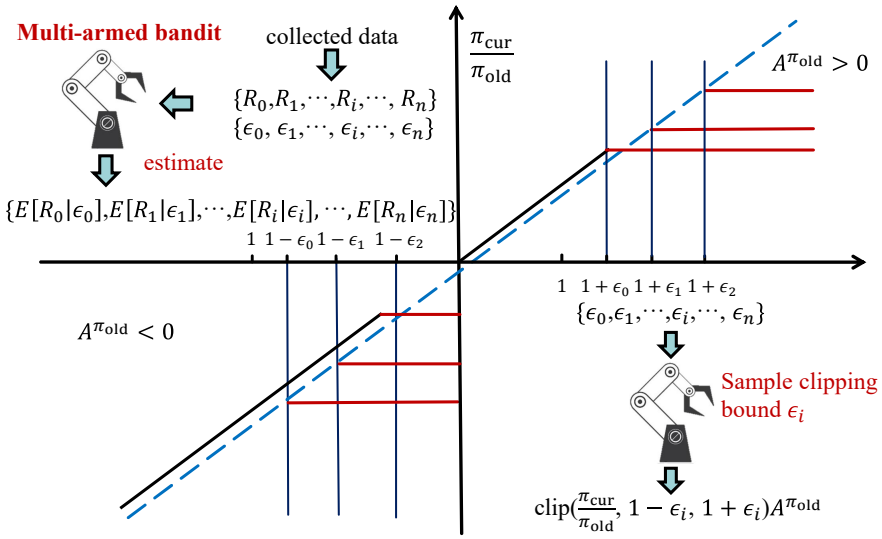


Fig. 2. The schematic diagram of proposed APPO algorithm.

sampling is known as exploitation. In contrast, it is known as exploration. Notably, solely relying on exploitation for updating the expected return estimations without incorporating exploration could hinder our ability to determine the optimal clipping bound, which might result in overestimating the confidence of sub-optimal clipping bounds. To mitigate the risks of balance between exploitation and exploration, necessitates the integration of uncertainty estimation into the process.

The UCB is a decision-making algorithm, which serves to maintain an equilibrium between the exploration and exploitation of options by integrating an uncertainty estimation $\hat{U}(\epsilon_i)$

$$U^{UCB}(\epsilon_i) = U(\epsilon_i) + \hat{U}(\epsilon_i). \tag{22}$$

Besides, the uncertainty estimation of i-th clipping bound ϵ_i can be formulated as

$$\hat{U}(\epsilon_i) = \lambda \sqrt{\frac{N^{\text{bandit}}}{N_{\epsilon_i}^{\text{bandit}} + \text{eps}}}, \tag{23}$$

where λ is a coefficient to control the value of uncertainty estimation, while eps is a very small float number set to prevent value overflow.

Specifically, given the sampling times $N_{\epsilon_i}^{\text{bandit}}$ of ϵ_i and total sampling times $N^{\text{bandit}} = \sum_{\epsilon_i \in \mathcal{T}} N_{\epsilon_i}^{\text{bandit}}$, if a certain clipping bound is sampled infrequently, resulting in a lower N_{ϵ_i}, it will correspondingly yield a higher $\frac{N^{\text{bandit}}}{N_{\epsilon_i}^{\text{bandit}}}$, leading to a larger U^{UCB}. This encourages the exploitation of such clipping bounds.

Algorithm 1 APPO Algorithm

Initialize the parameters, including the multi-armed bandit with clipping bounds $\mathcal{T} = \{\epsilon_0, \epsilon_1, \cdots, \epsilon_i, \cdots, \epsilon_n\}$, the total sampling times N, the sampling times of each bandit $\{N_{\epsilon_0}^{\text{bandit}}, N_{\epsilon_1}^{\text{bandit}}, \cdots, N_{\epsilon_i}^{\text{bandit}}, \cdots, N_{\epsilon_n}^{\text{bandit}}\}$, online replay buffer $\mathcal{D}_{\text{online}}$, the critic network of PPO, old and new policies π_{old}, π_{cur} of the AUV.
for each episode k **do**
  Reset the training environment, total reward, and visitation counters.
  for each time step t **do**
    Sample an action according to the policy:
      $a_t \sim \pi_{\text{old}}\left(a_t \mid s_t\right)$;
    Collect the next state from environment:
      $\boldsymbol{s}_{t+1} \sim \mathcal{P}\left(s_{t+1} \mid s_t, a_t\right)$;
    Calculate reward r_t by Eq. (13) $\sim$ Eq. (16);
    Store sampling tuple (s_t, a_t, r_t, s_{t+1}) into $\mathcal{D}_{\text{online}}$.
    Computing UCB values $U^{UCB}(\epsilon_i)$ via Eq. (21)$\sim$(23);
    Sample a clipping bound according to the maximum of UCB values:
      $\epsilon^* \leftarrow \arg\max_{\epsilon_i}\{U^{UCB}(\epsilon_i)|\epsilon_i \in \mathcal{T}\}$
  end for
  Update the policy π_{cur} with Eq. (20)
    $\mathcal{J}_{\pi_{\text{cur}}} = \max \mathbb{E}_{\tau\sim\pi_{\text{old}}}[\min(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}} A^{\pi_{\text{old}}}, \text{clip}(\frac{\pi_{\text{cur}}}{\pi_{\text{old}}}, 1-\epsilon^*, 1+\epsilon^*) A^{\pi_{\text{old}}})]$.
  Update the evaluated return of the bandit:
    $R^{\text{bandit}} \leftarrow R^{\text{bandit}} + R_{\epsilon^*}^{\text{bandit}}$
  Update the total visitation counter and bandit visitation counter by
    $N^{\text{bandit}} \leftarrow N^{\text{bandit}} + 1$
    $N_{\epsilon^*}^{\text{bandit}} \leftarrow N_{\epsilon^*}^{\text{bandit}} + 1$.
  Update the old policy with the new policy:
    $\pi_{\text{old}} \leftarrow \pi_{\text{cur}}$
  Update the critic network in PPO via mean-squared error L_{MSE}.
end for

4.3 Sampling Clipping bound with Alternate Uncertainty Term

Therefore, we can sample the optimal clipping bound with highest UCB value to efficiently balance exploration with exploitation of candidate clipping bounds

$$\epsilon^* \leftarrow \arg\max_{\epsilon_i}\{U^{UCB}(\epsilon_i)|\epsilon_i \in \mathcal{T}\}. \tag{24}$$

Next, we detail the updating process for the expected return of clipping bound ϵ_i, namely, updating $\mathbb{E}[R_i|\epsilon_i]$. For each sampled clipping bound ϵ_i, we first update the policy π_{old}. Subsequently, we evaluate this updated policy π_{cur}, obtaining the average evaluated return $R_{\epsilon_i}^{\text{bandit}}$ as the reward $r_{N_{\epsilon_i}}$ for arm ϵ_i. Consequently, we can compute the expected return of sampling ϵ_i as $\mathbb{E}[R_i|\epsilon_i]$.

Based on the above modules, we can integrate them together with PPO to compose the design of APPO algorithm, whose pseudo-code is detailed in Algorithm 1.

Table 1. Parameters of Simulation Experiment

Parameters	Values
Max velocity $v_{\max}$	1.0 m/s
Max angular velocity $\omega_{\max}$	1.6 rad/s
Experimental site size	40m × 40m
Safe distance $l_{\min}^{i \leftrightarrow j}$	1.6 m
Target distance $l_{\max}^{i \leftrightarrow T}$	2.5 m
discount factor γ	0.99
Maximum steps per episode T	2000
Time step per episode Δt	0.25
Training episodes ε	1000
Hidden layer size	256

5 Experiments

In this section, we aim to validate the proposed APPO through simulation experiments of training a single AUV for the navigation task. First we present the experimental setup, followed by a detailed description of the entire experiment process. Subsequently, we analyze and discuss the results of the experiments, focusing on the performance of APPO.

5.1 Experimental Setup

During the simulation, we employ two distinct sets of parameters: the simulation environment and algorithm parameters. These sets of parameters are considered comprehensively to ensure an effective evaluation.

Simulation Environment Parameters. The simulation is carried out on a 40m × 40m area with a water depth of −200m, on which the obstacles are randomly distributed. At the beginning, the position of the AUV is randomly distributed, and the AUV knows its own position. The area boundaries act as obstacles to restrict the AUVs in the specified area.

Algorithm Parameters. The implementation of APPO incorporates various parameters and settings. The discount factor γ is assigned a value of 0.99. During each episode, a maximum of 2,000 steps T are allowed, with a simulation time step Δt of 0.25s. a hidden layer size of 256 is utilized. All the parameters are detailed in Table 1 for a summary.

5.2 Experimental Results and Analysis

Algorithm Comparison Experiments. We first conducted simulation experiments on a 40m×40m square, employing the APPO algorithm to train the AUV for navigation and obstacle avoidance. Each training episode commenced

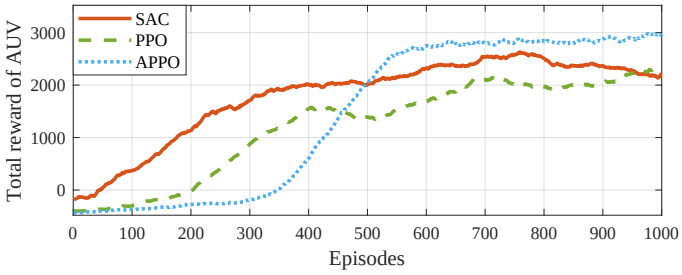
with the AUV positioned at a specified coordinate (x, y, z), with a training reset condition, denoted as $\chi(t)$, triggered upon colliding obstacle or upon reaching the maximum step count in an episode, thereby initializing the AUV's position to $(x^{'}, y^{'}, z^{'})$ for the subsequent episode. At each episode, the AUV constantly interacted with the surrounding environment, making real-time decision and obtaining corresponding reward based on the AUV's current state. However, the preliminary training phase, characterized by AUV's suboptimal policy, resulted in frequent collisions with obstacles. While the collected interaction experience served as the valuable data for training the policy and networks after each episode. It is also notable that each bandit arm had very high uncertainty estimation at the beginning of the training, which necessitated the APPO algorithm to engage in exploring each arm.

As training progressed, the APPO adeptly balanced exploration and exploitation of clipping bounds across different arms, leading to select the optimal arm with corresponding clipping bound for AUV to obtain highest expected reward. And throughout the training process, the AUV simultaneously assimilated valuable insights from its interactions, facilitating policy improvement. This iterative learning process culminated in the AUV's proficiency to navigate towards the target point while adeptly avoiding obstacles, thereby marking a transition from initial suboptimal policy to final expert policy. After 1000 training episodes, the AUV has mastered an expert policy, thereby fulfilling the navigation task. The evolution of the AUV's policy, as evidenced by the total reward curve during RL training, is depicted in Fig. 3(a).

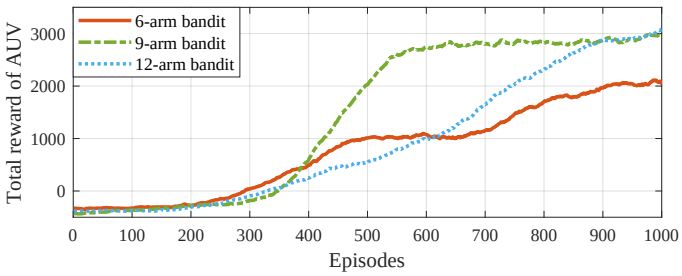
Furthermore, in an endeavor to evaluate the superiority of APPO, comparative experiments were conducted utilizing both the Proximal Policy Optimization (PPO) and the Soft Actor-Critic (SAC) algorithms under identical experimental setup. The experiment results are delineated in Fig. 3(a). The curves in Fig. 3(a) revealed that, in preliminary stages, both PPO and SAC exhibited significant policy improvement, achieving reward of 1400 and 1800 after 500 episodes' training, surpassing the Adaptive PPO's reward of 1000.

Nevertheless, the Adaptive PPO demonstrates a superior advantage in policy improvement in the last 500 episodes, evidenced by a consistent increment in reward, ultimately converging to a value of 2941. In contrast, PPO and SAC's reward oscillated around the 2000 and 2400, respectively. These observations underscore APPO's superiority in facilitating navigation and obstacle avoidance for AUV, while showcasing its enhanced training stability.

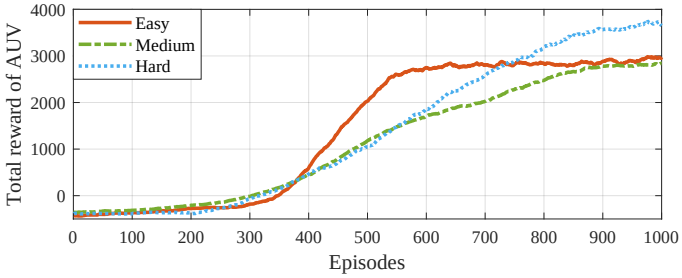
Ablation Experiments. To assess the impact of different bandit arm numbers on the performance of the APPO algorithm, ablation experiments were conducted with the arm numbers varying from 6 and 12, respectively. Corresponding clipping bounds were defined as [0.005, 0.05, 0.12, 0.16, 0.20, 0.24] for the 6-arm bandit, while [0.005, 0.01, 0.03, 0.05, 0.08, 0.12, 0.16, 0.17, 0.20, 0.24, 0.28, 0.32] for the 12-arm bandit. These experiments were performed under identical other parameter settings, with the outcome of the reward curves illustrated in Fig. 3(b). By observing Fig. 3(b), it can be found that in the preliminary



(a) Reward curves of different algorithms.



(b) Reward curves of APPO with different bandit arms.

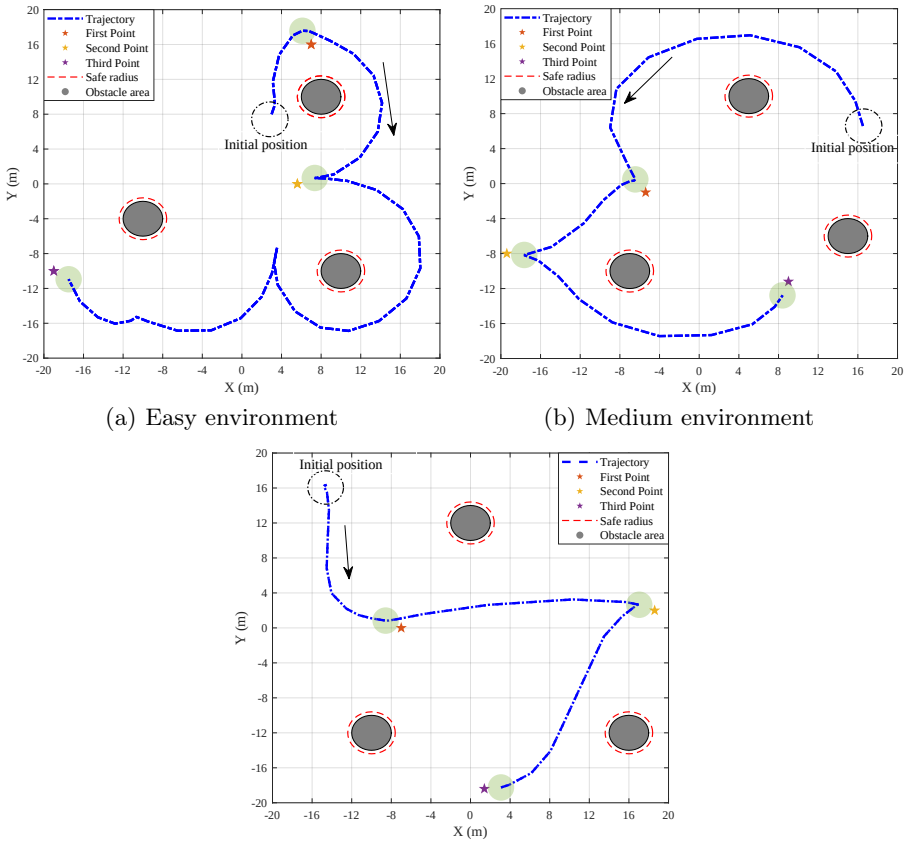


(c) Reward curves of APPO in environments with different difficulty.

Fig. 3. (a) Reward curves of different algorithms (SAC, PPO and APPO). (b) Reward curves of APPO with different bandit arms (6-arm, 9-arm and 12-arm). (c) Reward curves of APPO in various environments with different difficulty (Easy, Medium, Hard).

training stage, the increase speed of reward decreased first and then increased as the number of arms rose. While in the last 500 training episodes, the final reward value varied from 1986, to 2941 and to 2763 when arm number ranged from 6 to 12, respectively, demonstrating an initial increase and subsequent decrease in reward values with rising arm numbers.

This phenomenon suggests that the number of arms influences the balance between exploration and exploitation of the clipping bound in the APPO algorithm. With fewer arms, the algorithm tends to expedite the exploration of clipping bounds, thereby accelerating preliminary-stage policy improvement but



(a) Easy environment

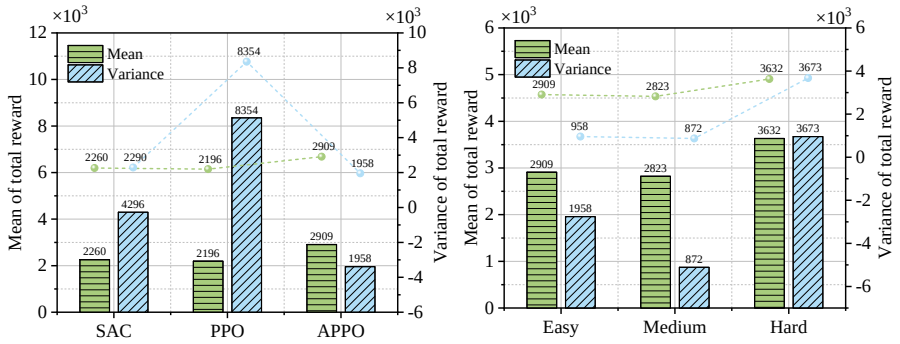
(b) Medium environment

(c) Hard environment

Fig. 4. Trajectories of the AUV in the environment with different difficulty. (a) Trajectories of the AUV in the easy environment. (b) Trajectories of the AUV in the medium environment. (c) Trajectories of the AUV in the hard environment. (The green circles denote the target distance $l_{\max}^{i \leftrightarrow T}$.)

potentially leading to premature convergence to local optima, as reflected by lower reward values in the end. Conversely, a bandit with more arm extends the exploration phase, decelerating initial policy improvement but mitigating the risk of suboptimal convergence, hence achieving higher reward values in the final episode. This analysis underscores the significance of selecting optimal number of arms with corresponding clipping bounds to maximize algorithm performance.

Environment Generalization Experiments. Furthermore, to verify the generalization capabilities of the APPO algorithm, we changed the environmental parameters of obstacles within the simulation. Specifically, we designed three environments of different difficulty varying from easy to hard. In the easy envi-



(a) Different algorithms. (b) Different environments.

Fig. 5. (a) The mean and variance of total reward using three different algorithms for training (SAC, PPO and APPO). (b) The mean and variance of total reward using APPO for training in environments with different difficulty (Easy, Medium, Hard).

ronment, the location of each obstacle is fixed, which can be randomly initialized at set intervals in the medium environment. While in the hard environment, the frequency of random initialization increases. Relying on these environments, we conducted simulation experiments employing APPO to train the AUV for navigation and obstacles avoidance, and utilized the trained model to complete the navigation task. The reward curves of training and visualization of trajectories in three different environment are depicted in Fig. 3(c) and Fig. 4, respectively.

Observations from these results revealed that AUV can obtain the expert policy via APPO for training in three different environments. Based on the trained model via APPO, AUV can adeptly navigate through various environments, consistently achieving high reward outcomes. This performance is indicative of APPO's robust generalization capability, demonstrating effectiveness in adapting to diverse challenges presented by fluctuating environmental conditions.

Finally, we calculated the mean and variance of the total reward from the last 100 episodes of algorithm comparison and environment generalization experiments, respectively, and visualized the results in Fig. 5. The APPO algorithm achieved higher mean while maintained a lower variance of total reward in Fig. 5(a). On the other hand, the AUV trained by APPO all got satisfactory outcomes in three different environments in Fig. 5(b). Through analyzing above observations, we can conclude that the proposed APPO outperforms the baseline algorithms, showcasing superior adaptability, robustness, and generalization. Future work will consider further improving the realism of the simulation, and conduct both simulation and real-world experiments in even more complex tasks.

6 Conclusion

This study investigates the AUV robust navigation problem in complex environments, modeling it as a Markov decision process and solving it using the

proposed APPO algorithm. APPO optimizes the selection of clipping bounds in PPO by dynamically exploring and exploiting clipping bounds during online training using a bandit, guiding PPO to make optimal choices at different stages of online training by maximizing the value of the upper confidence bound of each candidate bound. Compared to methods using fixed trust regions, APPO can dynamically respond to the requirements of training tasks with lower computational complexity. In simulation experiments, by gradually changing the environment of AUV navigation tasks from easy to hard, conducting ablation experiments, and comparing with different algorithms, it is found that APPO can efficiently complete tasks in different navigation scenarios, demonstrating superior adaptability, robustness, and generalization.

References

1. Chen, C., Chen, X.Q., Ma, F., Zeng, X.J., Wang, J.: A knowledge-free path planning approach for smart ships based on reinforcement learning. Ocean Engineering **189**, 106299 (2019)
2. Chu, Z., Wang, F., Lei, T., Luo, C.: Path planning based on deep reinforcement learning for autonomous underwater vehicles under ocean current disturbance. IEEE Transactions on Intelligent Vehicles **8**(1), 108–120 (2023)
3. Gong, Y.J., Huang, T., Ma, Y.N., Jeon, S.W., Zhang, J.: Mtrajplanner: A multiple-trajectory planning algorithm for autonomous underwater vehicles. IEEE Transactions on Intelligent Transportation Systems **24**(4), 3714–3727 (2023)
4. Hadi, B., Khosravi, A., Sarhadi, P.: Adaptive formation motion planning and control of autonomous underwater vehicles using deep reinforcement learning. IEEE Journal of Oceanic Engineering **49**(1), 311–328 (2024)
5. Jiang, P., Song, S., Huang, G.: Attention-based meta-reinforcement learning for tracking control of auv with time-varying dynamics. IEEE Transactions on Neural Networks and Learning Systems **33**(11), 6388–6401 (2022)
6. Schulman, J., Levine, S., Moritz, P., Jordan, M., Abbeel, P.: Trust region policy optimization. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. pp. 1889–1897 (2015)
7. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
8. Wang, Z., Zhang, Z., Wang, J., Jiang, C., Wei, W., Ren, Y.: Auv-assisted node repair for iout relying on multiagent reinforcement learning. IEEE Internet of Things Journal **11**(3), 4139–4151 (2024)
9. Wei, W., Wang, J., Du, J., Fang, Z., Ren, Y., Chen, C.L.P.: Differential game-based deep reinforcement learning in underwater target hunting task. IEEE Transactions on Neural Networks and Learning Systems **early access**, 1–13 (2023)
10. Wen, J., Yang, J., Wang, T.: Path planning for autonomous underwater vehicles under the influence of ocean currents based on a fusion heuristic algorithm. IEEE Transactions on Vehicular Technology **70**(9), 8529–8544 (2021)
11. Yan, J., Gong, Y., Chen, C., Luo, X., Guan, X.: Auv-aided localization for internet of underwater things: A reinforcement-learning-based method. IEEE Internet of Things Journal **7**(10), 9728–9746 (2020)
12. Zeng, Z., Sammut, K., Lian, L., He, F., Lammas, A., Tang, Y.: A comparison of optimization techniques for auv path planning in environments with ocean currents. Robot. Auton. Syst. **82**(C), 61–72 (2016)