Journal of *Marine Science and Engineering*

MDPI

Article

Knowledge-Guided Reinforcement Learning with Artificial Potential Field-Based Demonstrations for Multi-Autonomous Underwater Vehicle Cooperative Hunting

Yang Wang [1,2], Weiye Wang [1,2,3] and Deshan Chen [1,2,*]

[1] State Key Laboratory of Maritime Technology and Safety, Wuhan University of Technology, Wuhan 430063, China; wangyang.itsc@whut.edu.cn (Y.W.); wywang@whut.edu.cn (W.W.)
[2] National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan 430063, China
[3] Hainan Institute, Wuhan University of Technology, Sanya 572000, China
* Correspondence: dschen@whut.edu.cn

Abstract: Multi-AUV cooperative hunting requires autonomous underwater vehicles (AUVs) to strategize the encirclement of evaders while navigating around obstacles and other AUVs. Despite the promise of multi-agent reinforcement learning (MARL) in continuous control problems, its low sample efficiency poses a challenge in unknown environments and complex control scenarios. To overcome these limitations, we present a Knowledge-Guided Reinforcement Learning (KG-RL) approach, which integrates an Artificial Potential Field (APF) to enhance sample efficiency and operational safety. Our methodology is bifurcated into pre-training and fine-tuning phases. During the pre-training phase, an APF is employed to generate a concise set of demonstration trajectories that provide agents with foundational knowledge. Subsequently, the fine-tuning phase leverages real-time APF knowledge to direct the learning process, encouraging agents to balance following demonstrated actions with seeking out more optimal solutions. We assess the efficacy of our method through extensive simulations across diverse tasks, demonstrating its ability to expedite the learning process and yield more strategic decision-making. Our approach achieves superior results compared to traditional MARL benchmarks, particularly in learning efficiency, decision quality, and overall performance.

Keywords: AUV; cooperative hunting; multi-agent reinforcement leaning; path planning

check for updates

Academic Editor: Giuseppe Giorgi

Received: 31 December 2024
Revised: 18 February 2025
Accepted: 22 February 2025
Published: 24 February 2025

Citation: Wang, Y.; Wang, W.; Chen, D. Knowledge-Guided Reinforcement Learning with Artificial Potential Field-Based Demonstrations for Multi-Autonomous Underwater Vehicle Cooperative Hunting. *J. Mar. Sci. Eng.* **2025**, *13*, 423. https://doi.org/10.3390/jmse13030423

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Autonomous underwater vehicles (AUVs) have become increasingly integral in a variety of sectors, including marine scientific research, resource exploration, underwater search and rescue, and military operations. The coordinated deployment of multiple AUVs, often referred to as a swarm, offers significant advantages over single-vehicle operations in terms of operational efficiency and stability when executing intricate tasks. At the heart of multi-AUV operations lies the cooperative hunting problem, where a collective of intelligent AUVs collaborates to pursue and capture evaders, presenting a complex challenge that necessitates sophisticated strategy and coordination [1].

The pursuit of effective strategies for multi-robot systems, particularly in the context of cooperative hunting, has been the focus of extensive research. This body of work has highlighted the importance of considering environmental dynamics, the spatial awareness of evaders and hunters, and the integration of various operational elements. For example, Grinton [2] investigated the generation of compatible commitments among hunters

through commitment and convention cooperation mechanisms. The study involved comparing different control strategies and communication schemes to understand their impact on cooperative hunting efficiency. Zhu et al. [3] proposed a novel approach combining self-organizing maps with velocity synthesis to tackle dynamic task allocation and path planning in multi-AUV systems. Similarly, Liu et al. [4] devised a collaborative path planning strategy for multi-AUV operations, incorporating the effects of time-varying ocean currents and optimizing routes using dynamic programming. However, these studies usually assume that the environment is known or static, which is often not the case in real-world scenarios where the environment can be dynamic and largely unpredictable.

Deep Reinforcement Learning (DRL) has emerged as a transformative approach across diverse fields, offering a solution to the challenges posed by complex and unknown environments. Unlike traditional learning methods, DRL is capable of learning optimal policies directly from interaction with the environment, without the need for prior knowledge. This self-learning capability [5] has led to promising outcomes in tasks involving both discrete and continuous control systems, with early applications focusing on environments that are stable and easily observable.

The application of DRL to underwater multi-AUV cooperative tasks has garnered significant attention, with recent studies investigating RL-based algorithms specifically tailored for multi-AUV hunting scenarios. These include dynamic trajectory prediction, negotiation allocation, and DRL-based path optimization. For instance, a distributed collaborative search system for multi-AUVs was proposed in [6], which facilitates the exchange and integration of information among AUVs. This system then employs an RL-based method to determine optimal search paths based on environmental conditions. However, the complexity of underwater environments, characterized by their dynamic and often unpredictable nature, presents significant challenges in the collection of effective training samples. The demand for vast quantities of data, even for seemingly simple behaviors, escalates the complexity and cost of training for more intricate, high-dimensional tasks. Consequently, achieving high sample efficiency is a critical challenge that DRL algorithms must surmount.

In this paper, we focus on the application of DRL to multi-AUV hunting missions, proposing an end-to-end path planning method grounded in a multi-agent reinforcement learning (MARL) framework. Our approach aims to tackle the problem of cooperative hunting in perceivable dynamic environments. Firstly, a hunt team is established and a simulation environment that reflects the attributes of the hunters and the requirements of the task is constructed. We then utilize an APF-based method to generate a set of demonstration trajectories prior to the reinforcement learning process. These demonstration trajectories serve as a form of prior knowledge, enabling the agents to develop behaviors akin to those of expert policies. Finally, during the RL process, agents are guided by the demonstration trajectories and their interactions with the environment, allowing them to select between expert policies or to identify superior alternatives based on the evaluation of states and actions.

The primary contributions of this work are threefold:

(1) The introduction of a MARL framework to tackle the multi-AUV cooperative hunting problem. We develop a three-dimensional seabed and ocean currents environment model to train and assess the effectiveness of our method. Additionally, we design novel reward functions that account for motion constraints, the impact of ocean currents, and the formation characteristics of the AUVs.

(2) The proposal of a Knowledge-Guided Reinforcement Learning (KG-RL) algorithm that accelerates the training process by employing an APF-based approach, significantly enhancing training speed and reducing computational costs. Our method stands out by

utilizing a prior policy to generate pre-training data and providing more direct and precise guidance to agents during the RL process.

(3) The conduct of comprehensive simulations that incorporate unknown multiple 3D obstacles and dynamic turbulence to evaluate the performance of the proposed KG-RL method in underwater hunting tasks. The results of these simulations demonstrate the superior performance of our method over traditional MARL algorithms, particularly in terms of learning speed, success rate, and energy efficiency.

The structure of this paper is as follows: Section 2 discusses related work in the field. Section 3 provides a detailed description of the cooperative hunting mission. Section 4 presents the MADDPG-based hunting algorithm. Section 5 showcases the simulation experiments conducted for various scenarios. Finally, Section 6 offers conclusions and potential directions for future research.

2. Related Works

2.1. Multi-AUV Cooperative Hunting

Cooperative hunting can be considered an extension of flocking control, where numerous conventional control strategies have been developed to address related issues. Generally, these can be divided into four categories: leader-following approach [7,8], Artificial Potential Field (APF) approach [9], virtual structure [10], and behavior-based approach [11]. Ma et al. [12] proposed a cooperative hunting strategy utilizing dynamic alliances to pursue moving targets effectively. Drawing inspiration from bio-inspired neural network models, Tan et al. [13] developed an advanced dynamic alliance algorithm for cooperative hunting, emphasizing the relationships between robots and their targets. To improve target pursuit, Ref. [14] incorporated metrics such as dispersion, homodromous, and district-difference degrees into an enhanced potential field method, facilitating dynamic obstacle avoidance and mitigating local minima issues. Wei et al. [15] introduced a strategic target allocation method for multi-AUV systems, aiming to improve both the operational efficiency and survival rates of the AUVs engaged in multiple-target missions. Ref. [16] utilized an HBSOM algorithm with a neural wave network structure to address operational challenges in complex underwater environments for multi-AUV systems, simulated using a 3D grid map. Reference [17] studied a method for hunting multiple targets in a 3D underwater environment, which includes three stages: task allocation, formation tracking, and target capture. Experimental results show that this method can effectively strike multiple underwater targets. To address multi-robot hunting in uncertain environments, Feng et al. [18] proposed a battlefield ambush scenario using UUVs, employing the Hybrid Quantum-behaved Particle Swarm Optimization (HQPSO) algorithm to identify paths with the highest probability of target detection.

These approaches have demonstrated strengths in fairness, task success rates, and the facilitation of task formation and completion. However, in unknown environments, traditional methods often struggle to balance safety and efficiency. When an AUV swarm encloses an escaping target, it must respond rapidly to the target's evasion tactics, necessitating high levels of timeliness and precision. Current state-of-the-art methods in cooperative hunting often fall short in modeling the intricate coupling relationships within multi-AUV systems and in characterizing the long-term impacts, such as cumulative costs, of specific actions within the control policy framework.

2.2. RL-Based Algorithms for Underwater Hunting

Reinforcement Learning (RL) has garnered significant attention for its effectiveness in sequential decision-making under uncertain and complex conditions. An increasing number of studies are leveraging RL for cooperative hunting tasks. Ref. [19] combined

traditional frontier exploration methods with DRL, enabling AUVs to perform tasks in unknown underwater environments. In Ref. [20], a DRL-based target tracking method for AUVs was introduced, mapping sensor inputs to control commands through an end-to-end structure, achieving autonomous capabilities in cooperative hunting and dynamic obstacle avoidance. Considering different mission scenarios of UUVs, He et al. [21] proposed the asynchronous multithreading proximal policy optimization-based motion planning methods, and the effectiveness of the methods was verified in several simulated environments. Wang et al. [22] presented an adaptive dynamic allocation method of round-up points based on adjacent UUVs. Then an intelligent cooperative target capture scheme was investigated using multi-agent deep reinforcement learning (MADDPG), which has better adaptability to UUV swarms in dynamic adversarial environments. Xia et al. [23] improved the MAPPO algorithm to solve the multi-target cooperative hunting problem, and validated the performance and effectiveness of this method in different scenarios through simulation experiments.

The sample complexity of model-free deep reinforcement learning (DRL) poses a significant challenge for its application in real-world robotics. This issue stems from several factors, including the inherent limitations of on-policy learning methods. On-policy approaches, such as trust region policy optimization [24] and asynchronous advantage actor-critic [25], restrict agents to learning solely from experiences generated by the current policy. Consequently, task complexity exacerbates the problem, as it leads to an exponential increase in the number of gradient steps and the volume of samples required per step to develop an effective policy. To address these limitations, off-policy learning methods have emerged as a promising alternative. These methods improve sampling efficiency by allowing agents to learn from experiences generated by any policy within the same environment. Examples include deep Q-networks [26] and deep deterministic policy gradients [27]. Despite their advantages, off-policy learning faces significant hurdles when used with nonlinear approximators like deep neural networks, as it is highly sensitive to hyperparameters and poses considerable implementation challenges [28].

Recent advancements have sought to overcome these obstacles by leveraging the maximum entropy framework. Notable contributions by Haarnoja et al. and Nachum et al. have led to the development of innovative algorithms such as soft Q-learning [29], soft actor-critic [30], and path consistency learning [31]. These methods have demonstrated substantial improvements in data efficiency and training stability, paving the way for more practical and reliable DRL applications in complex environments.

2.3. Expert Demonstrations for DRL

Incorporating expert demonstrations offers a promising approach to guiding agent learning in reinforcement learning. These demonstrations can be utilized in various ways, such as integrating them into a replay buffer [32,33] or employing them for pre-training the policy. Pre-training transforms the optimization task into a supervised learning task, allowing the agent to gain a foundational understanding of the task [34]. A prominent approach to leveraging expert demonstrations is imitation learning (IL), which bypasses the need for explicit reward signals. Instead, IL teaches agents to perform tasks by learning a direct mapping from observations to actions [35,36]. IL methods are typically categorized into behavior cloning (BC) [37], inverse reinforcement learning (IRL) [38], and generative adversarial imitation learning (GAIL) [39]. Despite its potential, IL often requires a substantial dataset of expert demonstrations to achieve effective results. While GAIL addresses this limitation by enabling robust policy learning from fewer demonstrations, it assumes that the provided expert demonstrations are optimal. This assumption may restrict the

effectiveness of the learned policy, particularly when expert demonstrations are imperfect or suboptimal.

In the context of underwater hunting tasks, collecting extensive amounts of expert demonstrations while ensuring they are derived from the optimal policy is impractical. Moreover, it is often feasible to construct a sparse yet informative reward signal that provides specific feedback on task performance. Therefore, a significant challenge lies in developing highly sample-efficient learning algorithms that can effectively utilize both the sparse reward signal and a limited set of non-optimal demonstrations, ultimately achieving the performance that exceeds those of the expert.

3. Problem Formulation and Environment Modeling

In this section, we introduce a three-dimensional ocean currents model to simulate the complex underwater environment. We detail the environment and objectives of the hunting task, design the state and action spaces for the MARL framework based on the kinematic model of the AUV, and present a value function specifically tailored for the cooperative hunting task.

3.1. Problem Statement

The multi-AUV cooperative hunting mission described in this paper is as follows: Several AUVs, equipped with forward-looking sonar, navigate underwater at random. Upon detecting an unknown intruder, an AUV disseminates a hunting command to neighboring AUVs. The receiving AUVs promptly converge on the target and collaborate to encircle the evader. While the boundaries of the operational area and the presence of an evader are known to the AUVs, the comprehensive environmental details of the area remain unknown to both the AUVs and the evader. The aim of this paper is to devise an effective cooperative hunting strategy for AUVs, focusing on the strategic aspect rather than the control intricacies of the AUV system itself.

To delineate the application scenarios and prerequisites for the proposed multi-AUV cooperative hunting strategy, the following clarifications are provided:

(1) The cooperative hunting strategy outlined in this paper is intended for non-combat applications, such as the encirclement and expulsion or the accompanying surveillance of illegal invasive submerged vehicles. It is assumed that our AUVs have a superior speed to that of the target.
(2) The location of the invasive target is detectable by the sonar equipped on our AUVs and can be shared through real-time transmission [19]. The sonar is capable of detecting a 120° horizontal sector with a detection range of up to 120 m.
(3) The depth variations of the AUV are relatively small compared to the variations on the horizontal plane, so we simplify the three-dimensional problem to an enclosing task on a two-dimensional plane.
(4) At least one AUV is capable of detecting the relative position information with respect to the target within the local coordinate system of the AUV. This information includes the target's relative position and orientation.

Existing studies consider that when the encircling AUV surrounds the target AUV and forms a closed Apollonius circle around it, it is considered a successful encirclement. Based on this, the encirclement target of this study is to form an encirclement circle under limited time or energy consumption, and the distance between any pursuer and evader is less than the preset threshold.

3.2. Environment Modeling

The multi-AUV cooperative hunting problem can be effectively modeled as a Markov Decision Process (MDP), with the objective of identifying an optimal policy that maximizes the cumulative discounted rewards for all participating agents. An MDP is formally characterized as a tuple $(S, A, O, R, P, \gamma, N)$, where

- $S = (s_1, \ldots, s_N)$ is a set of all states of the agent. In the context of our AUVs, these states may include the positions, velocities, and orientations of the AUVs and the evader, along with other relevant environmental factors.
- $A = (a_1, \ldots, a_N)$ is a set of all actions. For the AUVs, these actions could involve movement commands, such as adjusting speed, changing direction, or executing specific maneuvers to encircle the evader.
- $P : S \times A \times \widetilde{S} \to [0, 1]$ denotes the transition probability from state $\boldsymbol{s}$ to state $\boldsymbol{s}'$ when action $\boldsymbol{a}$ is executed. This function encapsulates the dynamics of the underwater environment and the effects of the AUVs' actions.
- R is the global reward function, which assigns an instantaneous reward based on the current state $\boldsymbol{s}$ and the action $\boldsymbol{a}$ taken. The reward function is designed to encourage the agent to take actions that are helpful in completing the task.
- $\gamma \in [0, 1]$ is the discount factor, which determines the influence of future rewards when calculating long-term returns.

3.3. Spaces of Observations and Actions

In order to facilitate the description of the motion and attitude information of the AUV, it is necessary to establish both a geographic inertial coordinate system and a carrier coordinate system, as shown in Figure 1. In this study, we establish the NED coordinate system with any point in the sea as the origin to serve as the inertial coordinate system, used to describe the AUV's position and attitude underwater. We establish a body coordinate system with the centroid of the AUV as the origin, used to describe the linear and angular velocities of the AUV. The motion parameters of the AUV in two coordinate systems are defined as shown in Table 1. By using rotation matrices and transformation matrices, we can precisely describe how the linear velocity and angular velocity of the AUV affect its global position and orientation. This study focuses on quickly planning the AUV's capture path; therefore, the control inputs for the AUV are $\dot{u}$, q, and r, from which the kinematic equations of the AUV can be derived:

$$\begin{cases} \dot{x} = u \cos\psi \cos\theta \\ \dot{y} = u \sin\psi \cos\theta \\ \dot{z} = -u \sin\theta \\ \dot{\theta} = q \\ \dot{\psi} = r / \cos\theta, \theta \neq \pm\pi/2 \end{cases} \tag{1}$$

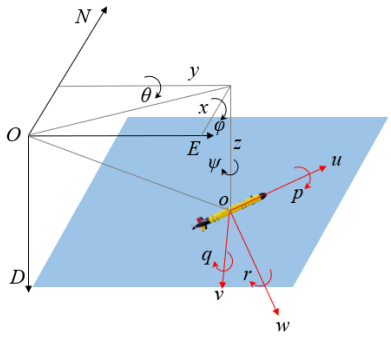


Figure 1. Coordinate reference systems of AUV.

Table 1. Parameters and symbols of AUV motion.

Motion	Velocity	Pose	Range
Surge	u	x	/
Sway	v	y	/
Heave	w	z	/
Roll	p	φ	$[-\pi/2, \pi/2]$
Pitch	q	θ	$[-\pi/2, \pi/2]$
Yaw	r	ψ	$[-\pi, \pi]$

There are numerous constraints that the AUV takes into consideration during task execution. In order to plan a rational path, the AUV's constraints are depicted as follows:

$$\begin{cases} -35^\circ \leq \theta \leq 35^\circ \\ 0\ \mathrm{m/s} < u < 3\ \mathrm{m/s} \\ -2^\circ/\mathrm{s} \leq q \leq 2^\circ/\mathrm{s} \\ -10^\circ/\mathrm{s} \leq r \leq 10^\circ/\mathrm{s} \\ -0.2\ \mathrm{m/s^2} \leq \dot{u} \leq 0.2\ \mathrm{m/s^2} \end{cases} \tag{2}$$

The state of the AUV during the encirclement of a target is illustrated in Figure 2, where the AUVs are indicated in blue, and the target is marked in red. The observations available to each AUV encompass

- The relative positions of the target and other AUVs, as detected by the sonar system.
- The distances to nearby obstacles, also identified through sonar detection.
- The forward speed, pitch angular velocity, and yaw rate of the AUV.
- The difference in hunting angles between adjacent AUVs, denoted as $\Delta\alpha_i = |\alpha_{i,k}| - |\alpha_{i,k}|$.

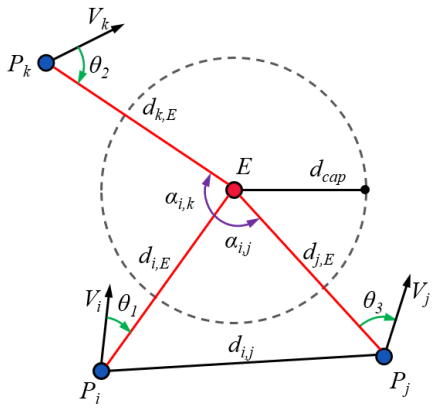


Figure 2. The state space of multi-AUV system.

3.4. Goal and Reward Functions

In the cooperative hunting scenario, the AUVs must work together to swiftly encircle an escaping target, while the target endeavors to evade and maintain distance from the AUV system. The AUV also needs to navigate around obstacles and avoid collisions with other AUVs. Considering the operational constraints of AUVs, including their range and energy consumption, the task is optimized for fewer steps and minimal speed variation. An episode of the hunting task concludes under three conditions: (1) The mission is successful when the distance between any pursuer and the escapee is less than a predefined threshold and each AUV is evenly distributed around the escapee. (2) If a collision occurs, the episode is immediately terminated. (3) When the number of time steps exceeds the maximum step limit, the episode is terminated.

In order to guide the AUVs to complete their tasks, we designed a composite reward function for reinforcement learning. Specifically, it includes a final reward r_{end}:

$$r_{end} = \begin{cases} -10, & \text{If the task failed} \\ 20, & \text{If the task succeeded} \end{cases} \tag{3}$$

A step reward r_{step} is used to urge AUVs to complete the task of cooperative hunting with as few time steps as possible, which is expressed as

$$r_{step} = -1 \tag{4}$$

A transition reward is used to guide the AUV to move towards the target. For a pursuer i at time step t, the transition reward r_{trans} is represented as

$$r_{trans} = d_{i,E}(o_{i,t-1}) - d_{i,E}(o_{i,t}) \tag{5}$$

where $d_{i,E}$ represents the distance between a pursuer and the evader.

A capture reward r_{cap} to guide the AUV in approaching the capture circle as closely as possible, which is represented as

$$r_{cap} = -\left|d_{i,E}(o_{i,t}) - d_{cap}\right| \tag{6}$$

A collision reward r_{col} is used to encourage pursuers to avoid obstacles, and it is represented as

$$r_{col} = \begin{cases} d_{i,obs}(o_{i,t}) - th_{col}, \text{ if } d_{i,obs}(o_{i,t}) < th_{col} \\ 0, \text{ else} \end{cases} \tag{7}$$

where $d_{i,obs}(o_{i,t})$ represents the distance between pursuer i and the nearest obstacle or other pursuers at time t, which can be detected by sonar. th_{col} is the threshold distance for AUV collision.

Considering the influence of ocean currents, which can be both a challenge and an opportunity due to the AUVs' low inertia [40], a current reward r_{cost} is defined to reflect the energy expenditure associated with navigating paths under the influence of currents:

$$r_{cost} = -\left|\sum_{i=1}^{N-1} k|V_{i,t-1}|^3 \cdot \frac{p_{t-1}p_t}{V_{i,t-1}}\right| \tag{8}$$

where $V_{i,t-1}$ denotes the still water speed of AUVs in the $p_{t-1}p_t$ path.

Considering the above reward functions, the composite reward function for pursuer i can be represented as

$$r = \rho_{end} \cdot r_{end} + \rho_{step} \cdot r_{step} + \rho_{trans} \cdot r_{trans} + \rho_{cap} \cdot r_{cap} + \rho_{col} \cdot r_{col} + \rho_{cost} \cdot r_{cost} \tag{9}$$

where ρ_{end}, ρ_{step}, ρ_{trans}, ρ_{cap}, ρ_{col}, and ρ_{cost} are weight coefficients of corresponding rewards.

3.5. Three-Dimensional Ocean Currents Model

Due to the horizontal motion scale of ocean currents being much larger than that of vertical motion, ocean currents are often approximated as multi-layer flows on a two-dimensional plane. This simplification is widely adopted in oceanographic modeling to capture the main dynamical features of ocean currents. In this study, an ocean current

model was constructed by superimposing several viscous laminar vortices outlined in [41], and the resulting numerical model is expressed in mathematical terms as follows:

$$\begin{cases} V_x\left(\vec{r}\right) = -\Gamma \frac{y-y_0}{2\pi\left(\vec{r}-\vec{r_0}\right)}\left(1-e^{-\left(\frac{\left(\vec{r}-\vec{r_0}\right)^2}{\sigma^2}\right)}\right) \\ V_y\left(\vec{r}\right) = \Gamma \frac{x-x_0}{2\pi\left(\vec{r}-\vec{r_0}\right)}\left(1-e^{-\left(\frac{\left(\vec{r}-\vec{r_0}\right)^2}{\sigma^2}\right)}\right) \\ \omega\left(\vec{r}\right) = \frac{\Gamma}{\pi\sigma}e^{-\left(\frac{\left(\vec{r}-\vec{r_0}\right)^2}{\sigma^2}\right)} \end{cases} \tag{10}$$

In this formula, $V_x\left(\vec{r}\right)$, $V_y\left(\vec{r}\right)$, and $\omega\left(\vec{r}\right)$ represent the vortex current velocity components in the horizontal, longitudinal, and vertical directions, respectively. Γ represents the intensity of the vortex current, $\vec{r_0}$ and σ represent the center vector and radius of the vortex current, respectively. It is important to note that this study focuses solely on spatial variations of the current, thereby assuming that the unit current magnitude remains constant within the operational region of the AUV throughout its mission.

4. Knowledge-Guided Reinforcement Learning Methodology

4.1. MARL Framework

MARL serves as a sophisticated model framework designed to optimize the coordinated control among multiple AUVs. Within this framework, the policy output for each agent is governed by a distinct neural network, allowing for individualized decision-making while promoting collective objectives.

The MADDPG algorithm extends the actor-critic methodology of the Deep Deterministic Policy Gradient (DDPG) to cater to multi-agent environments. MADDPG operates on the principle of centralized training with decentralized execution (CT-DE). During centralized training, each agent's critic network assimilates the state and action information from all agents, fostering a comprehensive understanding of the multi-agent system. In contrast, during decentralized execution, each agent's actor network generates decisions predicated on local information, ensuring agility and autonomy.

A significant advantage of MADDPG is its streamlined requirements; it does not necessitate an intricate model of the environment's dynamics nor does it rely on a differential communication channel among agents. This characteristic renders MADDPG particularly well-suited for complex and dynamic underwater environments where such detailed models and communication infrastructure may be impractical.

The basic framework of the MADDPG algorithm is shown in Figure 3. During the training process, each agent has its own distributed policy: $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_N\}$, $\boldsymbol{\theta_\pi} = \{\theta_{\pi_1}, \ldots, \theta_{\pi_N}\}$, $\boldsymbol{\pi}$ represents the policy, and $\boldsymbol{\theta_\pi}$ is a parameterized representation of the policy. Each has a centralized Q-function: $\boldsymbol{Q} = \{Q_1, \ldots, Q_N\}$, $\boldsymbol{\theta_Q} = \{\theta_{Q_1}, \ldots, \theta_{Q_N}\}$.$\boldsymbol{Q}$ represents the Q-Function, and $\boldsymbol{\theta_\pi}$ is a parameterized representation of the Q-Function. Considering the deterministic policies, the gradient of agent i can be calculated as

$$\nabla_{\theta_{\mu_i}} J_i = \mathbb{E}_{(\boldsymbol{o},\mathbf{a})\sim\mathcal{D}}\left[\nabla_{\theta_{\mu_i}}\mu_i\left(o_i \mid \theta_{\mu_i}\right)\nabla_{a_i}Q_i\left(\boldsymbol{o}, a_1, \ldots, a_N \mid \theta_{Q_i}\right)\Big|_{a_i=\mu_i(o_i)}\right] \tag{11}$$

where $\mathcal{D}$ is the experience replay buffer that records many experience tuples, and the distributed policy is refined by minimizing the loss function of the gradient:

$$L_{actor}(\theta_{\pi_i}) = \mathbb{E}_{(\boldsymbol{o},a,r,\boldsymbol{o}')\sim D}\left[-Q_i^{\boldsymbol{\mu}}\left(\boldsymbol{o}, a_1, \ldots, a_N \middle| \theta_{\pi_i}\right)\right] \tag{12}$$

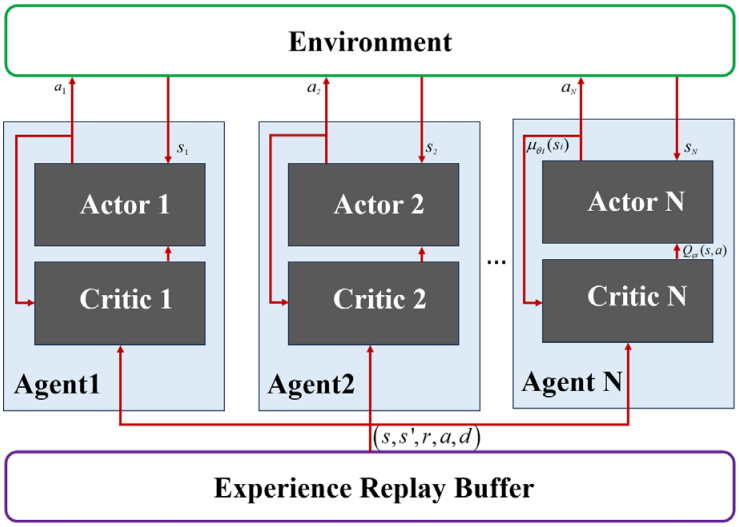


Figure 3. The framework of MADDPG.

The centralized Q-function is updated accordingly:

$$L_{actor}(\theta_{\pi_i}) = \mathbb{E}_{(\boldsymbol{o},a,r,\boldsymbol{o}')\sim D}\left[-Q_i^{\boldsymbol{\mu}}(\boldsymbol{o}, a_1, \ldots, a_N \middle| \theta_{\pi_i})\right] \tag{13}$$

where γ_i is the optimization target of the critic:

$$y_i = r_i + \gamma Q_i'(\boldsymbol{o}', a_1', \ldots, a_N')\big|_{a_j'=\boldsymbol{\mu}_j'(o_j')} \tag{14}$$

where $\boldsymbol{\mu}'$ is the set of target policies with delayed parameters $\theta^{Q1'}$, which introduces stability to the training process of the critic.

4.2. Knowledge-Guided MADDPG

In the MADDPG algorithm, the random initialization of network parameters often necessitates extensive interactions with the environment to learn effective policies. Improper initialization can lead to the algorithm becoming ensnared in local optima or facing difficulties in converging. Moreover, the exploration–exploitation trade-off inherent in reinforcement learning can also result in low sample efficiency. Behavior cloning (BC) can enhance efficiency by emulating optimal policies or demonstrations. While traditional BC is efficient due to its minimal connection to the MDP, it may encounter challenges when agents encounter states not seen during training.

This paper introduces an extension to the MADDPG framework that amalgamates the benefits of RL and IL while mitigating their respective drawbacks. The computation flowchart of this method is shown in Figure 4. The KG-RL algorithm encompasses two phases: a pre-training phase and a fine-tuning phase, designed to achieve global generalization and maintain sample efficiency. During the pre-training phase, an expert demonstration policy guides the learning process. This policy, despite not being optimal, significantly outperforms randomly initialized policies by imparting relevant task-solving knowledge. In this stage, agents leverage the expert's knowledge without direct interaction with the environment, which substantially accelerates the learning process. The existing literature [36,39] has tested the training speed of offline learning on benchmark problems, demonstrating that learning from offline data can effectively accelerate training speed.

In the fine-tuning phase, agents interact with the environment, storing their experiences in a replay buffer, which also includes expert demonstrations. This combined buffer is used to gradually refine the agent's policy network weights. Deviating from traditional

IL, the agent receives action recommendations from the expert policy during interactions with the environment, guiding its learning process. The advantage of this approach is that when the agent encounters an unfamiliar state, it can refer to the expert policy. If the expert's policy yields higher returns, the agent will adopt the expert's action; otherwise, the agent will rely on its own policy to decide on an action. This method allows the agent to leverage expert advice when necessary while still exploring and refining its own policy over time.

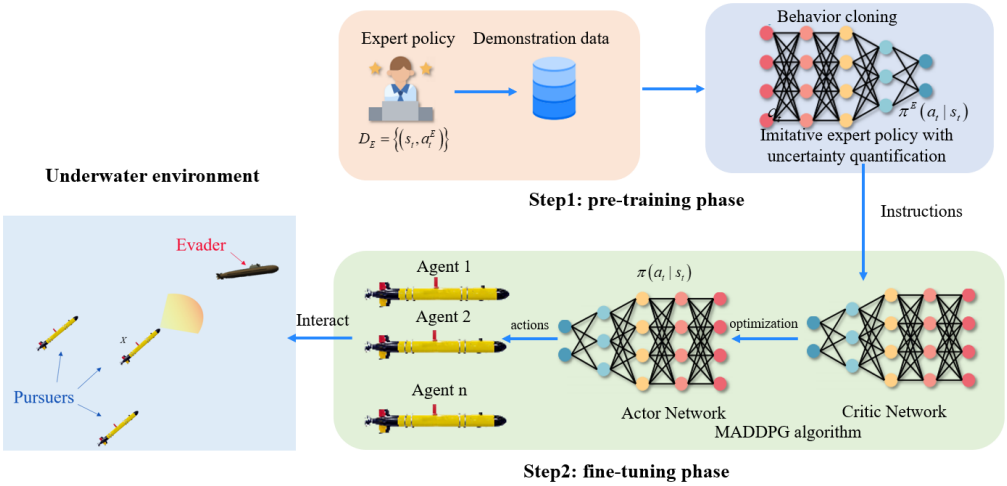


Figure 4. Computation flowchart of Knowledge-Guided MADDPG.

During the pre-training phase, continuous trajectories are obtained from expert policy (such as Artificial Potential Field or others). These expert demonstrations are archived in a dedicated expert replay buffer. Subsequently, the agent samples a batch of trajectories from this buffer for updating the actor and critic networks.

For actors, the composite loss function is articulated as

$$L_{pre}(\theta_{\pi_i}) = \lambda_{BC} L_{BC}(\theta_{\pi_i}) + \lambda_{actor} L_{actor}(\theta_{\pi_i}) + \lambda_{L_2\pi} L_{L_2}(\theta_{\pi_i}) \tag{15}$$

where L_{L_2} is a regularization term to prevent agents from overfitting demonstration trajectories. L_{BC} is the expert behavior cloning loss, which is designed to minimize the discrepancy between the actions predicted by the actor network and the actions taken by the expert under a given state:

$$L_{BC}(\theta_{\pi_i}) = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a})\sim\mathcal{D}_E}\left[\pi_i(o_i) - a_i^E\right]^2 \tag{16}$$

Q-learning loss of actor, $L_{actor}(\theta_{\pi_i})$, akin to Equation (12), is expressed as

$$L_{actor}(\theta_{\pi_i}) = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a})\sim\mathcal{D}_E}\left[-Q_i\left(\boldsymbol{o}, (o, a_1, \ldots, a_i, \ldots, a_N)|_{a_i=\pi_i(o_i)}\right)\right] \tag{17}$$

For the critics, the loss function is a synthesis of the Q-function and a regularization term:

$$L_{pre}(\theta_{Q_i}) = \lambda_{critic} L_{critic}(\theta_{Q_i}) + \lambda_{L_2 Q} L_{L_2}(\theta_{Q_i}) \tag{18}$$

The Q-function, resembling Equation (13), is adapted to utilize sampled trajectories from the expert replay buffer $\mathcal{D}_E$:

$$L_{critic}(\theta_{Q_i}) = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a})\sim\mathcal{D}_E}[Q_i(\boldsymbol{o}, a_1, \ldots, a_N) - y_i]^2 \tag{19}$$

Through pre-training, the initial distributions of the actors and critics are aligned closer to the expert trajectory, smoothing the transition from learning from expert demonstrations

to reinforcement learning. It is important to note that the policy's performance during the pre-training phase need not be optimal; it merely requires providing a robust starting point for the subsequent phase, thereby necessitating a modest amount of demonstration data.

During the fine-tuning phase, the agent engages with the environment to explore the optimal strategy, with the expert policy π_E continuing to guide the agent's behavior. The behavioral cloning loss is thus represented as

$$L_{imitation}(\theta_{\pi i}) = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{a})\sim\mathcal{D}}[\pi_i(o_i) - \pi_E(o_i)]^2 \quad \mathbf{1}Q_i(\boldsymbol{o}, \pi_E(o_i)) > Q_i(\boldsymbol{o}, \pi_i(o_i)) \tag{20}$$

The agent selectively imitates the expert's trajectories when the Q-function values of the expert policy's actions surpass those of the agent's current policy.

It is important to highlight that, unlike learning from offline data in the pre-training phase, the loss function in Equation (20) involves direct online interaction between the agent and the environment. While other methods might rely on pre-collected data from expert demonstrations or artificial agents to compute action differences, our approach directly imitates the actions recommended by the expert policy during each state encountered by the agent. This approach provides more immediate and precise guidance during learning, as the agent's updates are based on real-time interactions rather than historical data. The agent continuously updates its policy in response to expert feedback, allowing for adaptive learning that is closely aligned with the current state of the environment. In contrast, offline methods use static demonstrations, which can limit the agent's ability to adapt to new or dynamic environments. By directly incorporating expert recommendations during each interaction, our method provides more timely and context-aware feedback, improving the agent's ability to make decisions in complex environments.

For actors, the comprehensive loss function is designed as

$$L_{RL}(\theta_{\pi i}) = \lambda'_{actor} L_{actor}(\theta_{\pi i}) + \lambda_{imitatioon} L_{imitaion}(\theta_{\pi i}) \tag{21}$$

Regarding the critics, the target y_i for critic optimization in Equation (14) is also revised to refine Q-functions with the expert policy's input.

The critics, in turn, are adjusted to enhance the target y_i for optimization in Equation (14), aiming to refine the Q-functions based on the input from the expert policy.

$$\begin{aligned} y_i = r_i + \gamma max(Q'_i(\boldsymbol{o}', a'_1, \ldots, a'_n) \Big| a'_j = \pi'_j(o'_j), \\ Q'_i(\boldsymbol{o}', a'_1, \ldots, a'_i, \ldots, a'_n) | a'_i = \pi'_E(o'_i), a'_{j\neq i} = \pi'_j(o'_j)) \end{aligned} \tag{22}$$

This method compares the Q function generated by expert policy and the Q function obtained through iterative learning, with the goal of evaluating the expected Q-values of the next action under both policies. If the Q-function of the expert policy yields a higher reward, it suggests that the expert policy's action at that state is more optimal than that of the learned policy. As a result, the refined Q-functions lead to a more accurate estimation of expected returns, facilitating faster convergence and improving the efficiency of the training process. The proposed method is formally represented in the pseudocode provided as Algorithm 1.

Algorithm 1: Knowledge-Guided MADDPG

Initialize the cooperative hunting environment and MADDPG parameters
Initialize expert policy π_E such as APF
 # pre-training phase
for each *epoch* **do**
  Receive initial observation $\boldsymbol{o}$
  for time step $\in [1, 2, \ldots, \text{Max}]$ **do**
    Select action $a_i = \pi_E(o_i)$ for o_i using π_E
    Execute actions $\boldsymbol{a}$, received reward r and new observation $\boldsymbol{o}'$
    Store transition $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{o}')$ in pre-training replay buffer $\mathcal{D}_E$
    $\boldsymbol{o} \leftarrow \boldsymbol{o}'$
    for agent $i \in [1, 2, \ldots, N]$ **do**
      Sampling a random minibatch from $\mathcal{D}_E$
      Update actor by minimizing the loss defined as Equation (15)
      Update critic by minimizing the loss defined as Equation (18) with Equation (14) as target
    end for
    Update target networks softly with τ:
      $\theta'_{\pi_i} \leftarrow \tau\theta_{\pi_i} + (1-\tau)\theta'_{\pi_i}$
      $\theta'_{Q_i} \leftarrow \tau\theta_{Q_i} + (1-\tau)\theta'_{Q_i}$
  end for
end for
 # fine-tuning phase
for each *epoch* **do**
  Initialize a random process $\mathcal{N}$ for action exploration
Receive initial observation o
for time step $\in [1, 2, \ldots, \text{Max}]$ **do**
Select exploratory action $a_i = \pi_{\theta_i}(o_i) + \mathcal{N}_t$ using the current policy
Execute actions $\boldsymbol{a}$, received reward r and new observation $\boldsymbol{o}'$
Store $(\boldsymbol{o}, \boldsymbol{a}, \boldsymbol{r}, \boldsymbol{o}')$ in replay buffer $\mathcal{D}_{RL}$
    for agent $i \in [1, 2, \ldots, N]$ **do**
    Sampling a random minibatch from $\mathcal{D}_{RL}$
Update actor by minimizing the loss defined as Equation (21)
Update critic by minimizing the loss defined as Equation (13) with Equation (22) as target
    end for
    Update target networks softly with τ:
      $\theta'_{\pi_i} \leftarrow \tau\theta_{\pi_i} + (1-\tau)\theta'_{\pi_i}$
      $\theta'_{Q_i} \leftarrow \tau\theta_{Q_i} + (1-\tau)\theta'_{Q_i}$
  end for
end for

4.3. Expert Policy Based on Artificial Potential Field

The APF method is an algorithm widely used in robot path planning and navigation. Its basic principle is to construct a virtual potential field, enabling the robot to avoid obstacles and move towards the target point. The main advantages of this method are its simple calculations, strong real-time performance, and ease of implementation. The potential field $U(q)$ at position q is the superposition of attractive and repulsive fields, which can be defined as

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{23}$$

where $U_{att}(q)$ represents the attractive potential field generated by the target, and $U_{rep}(q)$ denotes the repulsive potential field generated by the obstacle.

The attractive potential field is articulated as

$$U_{att}(q) = \frac{1}{2} k_{att} (d_{i,obs}(o_{i,t}))^2 \tag{24}$$

where k_{att} denotes the gravitational gain parameter.

The repulsive potential field is expressed as

$$U_{rep}(q) = \begin{cases} \frac{1}{2} k_{rep} \left(\frac{1}{d_{i,obs}(o_{i,t})} - \frac{1}{th_{col}} \right), \text{ if } d_{i,obs}(o_{i,t}) < th_{col} \\ 0, \text{ else} \end{cases} \tag{25}$$

where k_{rep} denotes the gravitational gain parameter.

Using the above method, the AUV can perform path planning and navigation under the influence of the total potential field. However, the attractor field rapidly diminishes when the AUV approaches the target point, leading to insufficient attraction to stabilize the robot at the ideal position, which in turn causes oscillations. To address this issue, we propose a novel approach known as the hunting preference strategy, aimed at optimizing the coordination and efficiency of the pursuers. The attraction of the target to the pursuer is expressed as

$$F_{att} = \left[d_{cap} - \left| d_{i,obs}(o_{i,t}) \right| \right] \cdot \frac{1}{\left| d_{i,obs}(o_{i,t}) \right|} \cdot d_{i,obs}(o_{i,t}) \tag{26}$$

The force function for an agent i, generated by all neighboring friendly AUVs, can be expressed as

$$F_{cap,i} = \begin{cases} \sum_{j=1}^{n} \left[\left(\frac{1}{d_{i,j}(o_{i,t})} - \frac{2 \sin\left(\frac{\pi}{n}\right)}{d_{\text{detec}}} \right) \cdot \frac{1}{\left\| d_{i,j}(o_{i,t}) \right\|^2} \cdot d_{i,j}(o_{i,t}) \right], \text{if } d_{i,j}(o_{i,t}) \le d\text{detec} \\ 0, \text{ else} \end{cases} \tag{27}$$

where d_{detec} denotes the perceived range of AUVs, and n is the number of all pursuers.

5. Simulation Results and Analysis

This section presents a comprehensive suite of simulations designed to evaluate the efficacy of our proposed method within the context of multi-AUV cooperative hunting scenarios. We detail the environmental model configurations, assessment metrics, parameter settings, statistical analyses, and outcomes pertinent to the problem at hand. The simulations were conducted utilizing Python 3.7 on a system outfitted with an i9-10900 CPU, NVIDIA RTX 3080 GPU (Manufacturer: NVIDIA Corporation, Santa Clara, CA, USA), and a Win10 64-bit operating system.

5.1. Environment Model Settings

Across all simulation experiments, a 3D underwater workspace measuring 100 × 100 × 30 units is established with five pursuers and one evader. To replicate the cooperative hunting dynamics, an initial distance of greater than 15 m between the pursuers and the evader is meticulously maintained. The maximum velocities for the pursuers and evaders are capped at 1.5 and 1.2 units per time step, respectively, with a time step resolution of 0.1. The locations of obstacles, pursuers, and the evader, as well as the obstacles' sizes, are randomized at the beginning of each episode. In addition, the intensity and radius of the vortex are also randomly initialized. It is noteworthy that the evaders are endowed with a higher maximum speed to account for the pursuers' quantitative

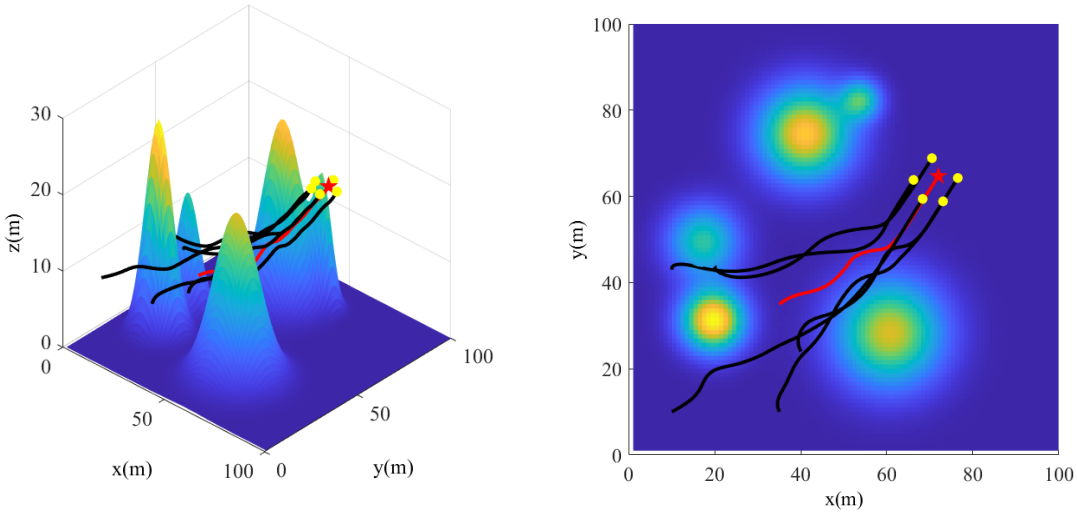
advantage, as it is nearly impossible for the evaders to elude all pursuers consecutively without this speed differential. The simulation concludes when there are no pursuers remaining or upon reaching the predefined maximum number of interactions.

The architecture of both actor and critic networks includes two hidden layers with 32 nodes each. ReLU activation functions are used in the initial layers, while hyperbolic tangent functions are applied to the output layer, ensuring actions and messages fall within the range of $[-1, 1]$. The global critic network utilizes an expanded structure, also featuring two hidden layers with 32 nodes each. For function approximation, a linear activation function is applied to the output layer. The training parameters include a discount factor of 0.95, a soft update factor of 0.01, a replay buffer capacity of 30,000, a maximum length of 500, and a batch size of 512. Simulations are conducted for up to 3,000,000 training steps.

5.2. Simulation Results in Calm Waters

To substantiate the efficacy of our approach, we juxtapose the paths orchestrated by our method against those of MADDPG and the APF-based method within a calm water scenario. The AUVs are constrained by a maximum speed of 3 m/s, with a forward-looking sonar detection range d = 30 m. The initial positions of the pursuers are strategically set to (10, 10, 9), (40, 24, 11), (35, 10, 9), (10, 43, 10), and (20, 44, 9), while the evader commences at (35, 35, 8).

As depicted in Figure 5a, at the simulation's inception, the pursuers and the target are dispersed across the environment. Upon detection of the target by a pursuer, the information is promptly disseminated among adjacent pursuers, prompting them to assemble a formation and initiate pursuit. Conversely, upon detecting a pursuer, the evader swiftly executes evasive maneuvers. Over time, the pursuers manage to overtake the target, culminating in an effective encirclement formation. Figure 5b presents a 2D projection of the entire hunting sequence. The simulation outcomes demonstrate that our method is adept at accomplishing the hunting task within 3D underwater settings, with the pursuers adeptly navigating around obstacles.

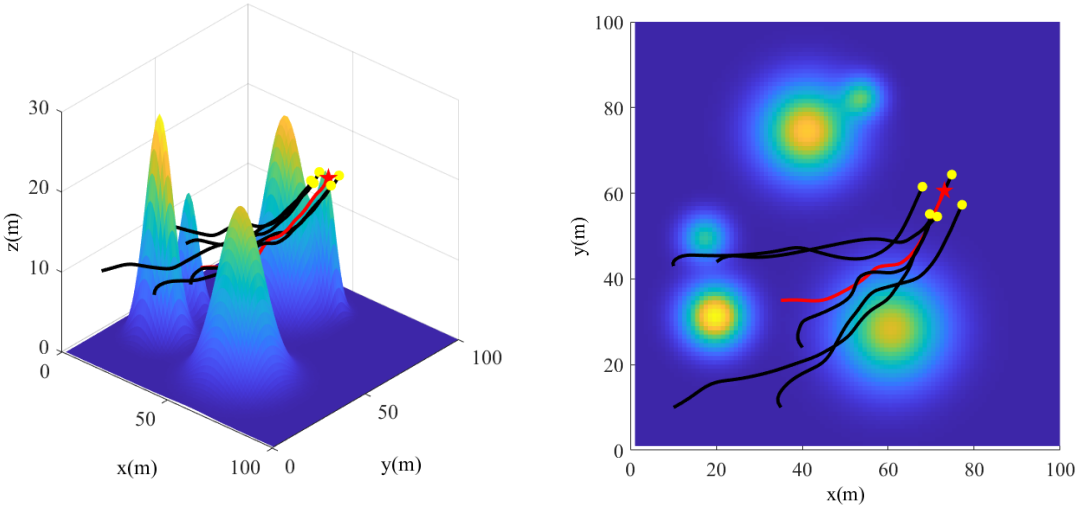


(**a**) (**b**)

Figure 5. Path planned by the proposed method. (**a**) 3D path planned by the proposed method; (**b**) 2D projection of the proposed method. The red lines represent the target's path, while the black lines represent the hunting team's path.

Figure 6a illustrates the trajectory planned by the MADDPG method. When juxtaposed with the path in Figure 6a, it is observed that the path lengths devised by both methods are

comparable, predominantly due to our method's foundational MADDPG framework and a shared set of training parameters. Consequently, the maximum cumulative rewards for both methods exhibit minimal variance post-convergence, indicating a parity in overall performance. Nevertheless, Figure 6b reveals certain imperfections in the MADDPG-planned paths, such as superfluous turns and acceleration/deceleration cycles, which also escalate energy expenditure.

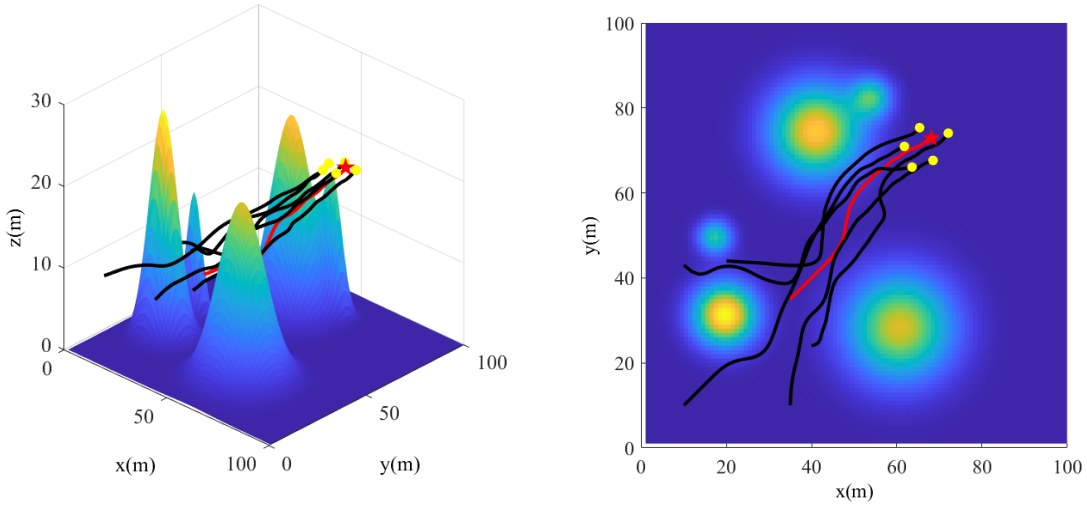


(**a**) (**b**)

Figure 6. Path planned by MADDPG. (**a**) 3D path planned by MADDPG; (**b**) 2D projection of MADDPG. The red lines represent the target's path, while the black lines represent the hunting team's path.

Figure 7a showcases the trajectory plotted by the APF-based method. In contrast to the path in Figure 6a, the path engineered by our method exhibits enhanced smoothness. Moreover, our approach efficaciously diminishes the pursuit distance, bolstering operational efficiency and curtailing energy consumption. As illustrated in Figure 7b, the APF-based method frequently engineers paths rife with abrupt turns, which are less than ideal for AUVs given their constrained maneuverability. In contrast, the sharp turns in Figure 5b are significantly reduced, primarily due to the introduction of energy consumption factors in the design of the reward function, which encourages the autonomous underwater vehicle to select low-energy and smooth paths, thereby reducing abrupt accelerations, decelerations, and turns in the trajectory, avoiding unnecessary energy waste.

A comparative analysis of the time and energy expended by each method to complete the hunting task is encapsulated in Table 2. As indicated in Table 2, our method consummates the task in a mere 35.6 s, outperforming alternative methods (39.2 s, 43.7 s) in terms of speed. Moreover, the average curvature of the paths planned with our method is 0.063, which is smaller than the other methods (0.069, 0.078). Smaller average curvature indicates that the paths are smoother and easier to pass through. Additionally, our method consumes the least energy (323.79 kJ), undercutting alternative methods (397.72 kJ, 463.65 kJ). The average encirclement distance of our method is the smallest (5.47 m), which is less than the average enclosure distances of other methods (6.29 m and 8.33 m), indicating a better encirclement performance. Furthermore, our method achieves the highest average speed (2.01 m/s) and cumulative reward (−92.88), surpassing the performance of other methods.



(**a**) (**b**)

Figure 7. Path planned by the APF-based method. (**a**) 3D path planned by the APF-based method; (**b**) 2D projection of the APF-based method. The red lines represent the target's path, while the black lines represent the hunting team's path.

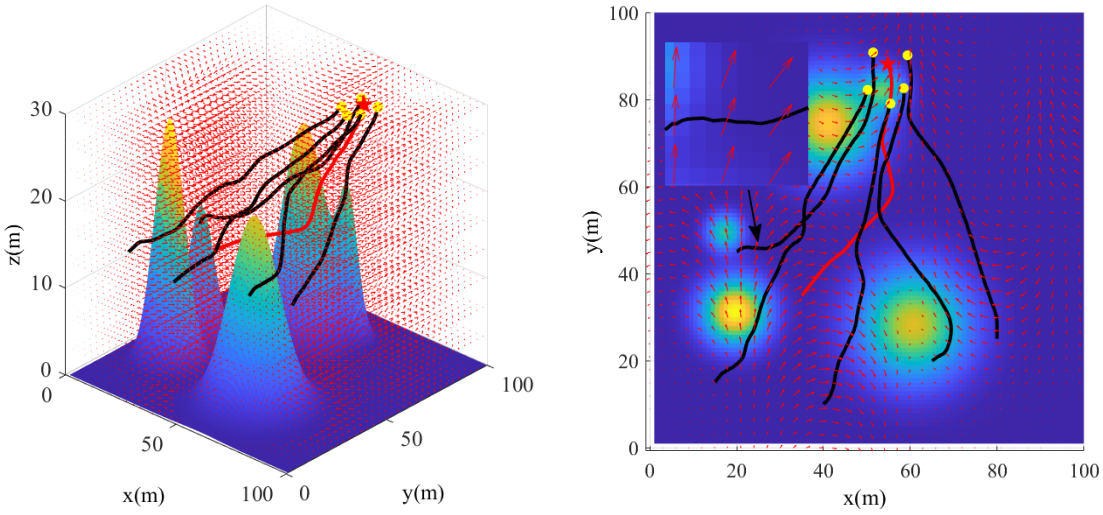
Table 2. Results by three methods in scenario 1.

Algorithm	Consumed Energy (KJ)	Mission Time (S)	Average Speed (M/S)	Average Curvature	Cumulative Reward	Average Encirclement Distance
Our Method	**323.79**	**35.6**	**2.01**	**0.063**	**−92.88**	**5.47**
MADDPG	397.72	39.2	1.85	0.069	−109.63	6.29
APF	463.65	43.7	1.67	0.078	−124.15	8.33

5.3. Simulation Results with Influence of Ocean Currents

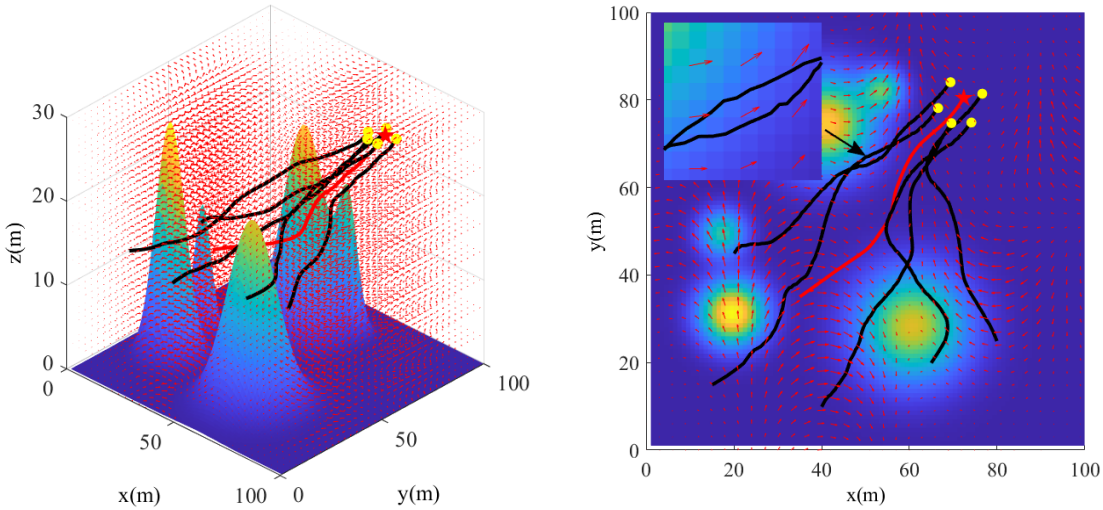
As an important dynamic factor in the marine environment, ocean currents have a significant impact on the path planning of AUVs. Utilizing ocean currents effectively can not only improve the energy efficiency and navigation accuracy of AUVs, but also enhance their mission execution capability and safety. In this section, the AUV's maximum speed in still water is set to 3 m/s, with a steady flow speed of the current at 0.2 m/s. The vortex current parameters are specified as $\sigma = 1.0$, $\Gamma = 20$, and the detection range of the side-scan sonar is d = 30 m. The initial positions of the hunters are coordinated as (15, 15, 14), (80, 60, 15), (40, 10, 15), (65, 20, 14), and (20, 45, 16), with the evader's starting position at (30, 30, 15).

As illustrated in Figure 8a, Figure 9a, Figure 10a, all three methodologies successfully navigate around obstacles with safety, yet the trajectories exhibit distinct characteristics and performance indices. The path of the hunters in Figure 10a has more abrupt turns and large velocity variations in the z-axis direction, which is clearly a challenge for the maneuverability of the AUV. The APF-based method relies on predefined potential fields that are fixed once generated. These fields may not adapt in real-time to the evader's changing strategies. Notably, the paths orchestrated by our method are, to a certain extent, aligned with the direction of the ocean current, a strategic choice aimed at conserving energy or reducing time expenditure.



(**a**) (**b**)

Figure 8. Path planned by the proposed method in ocean current environment. (**a**) 3D path planned by the proposed method; (**b**) 2D projection of the proposed method. The red lines represent the target's path, while the black lines represent the hunting team's path.



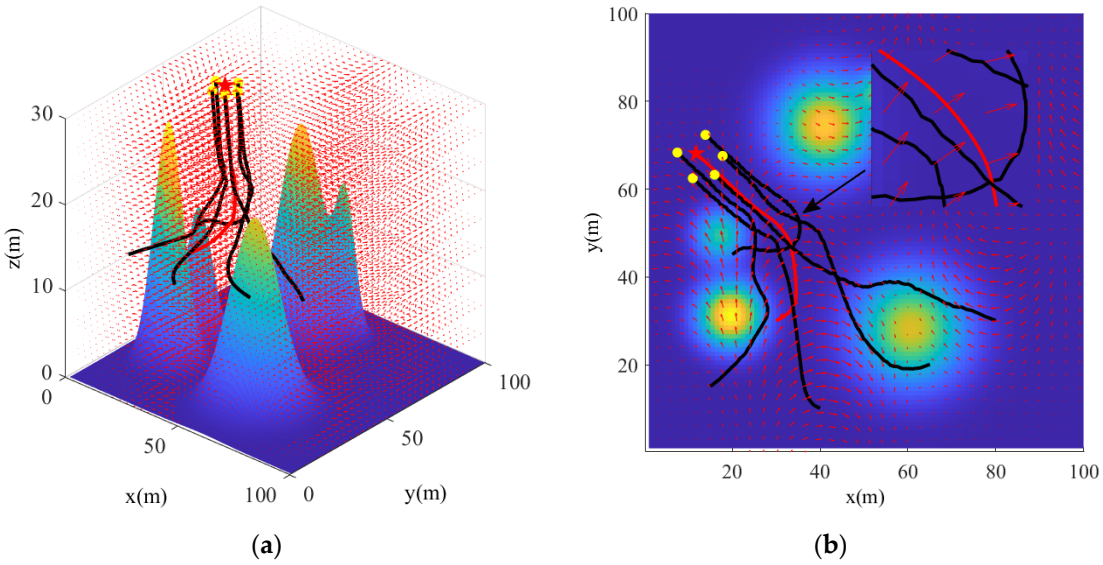
(**a**) (**b**)

Figure 9. Path planned by MADDPG in ocean current environment. (**a**) 3D path planned by MADDPG; (**b**) 2D projection of MADDPG. The red lines represent the target's path, while the black lines represent the hunting team's path.

Figure 10b depicts a phase in which the hunters commence movement against the current, resulting in diminished velocity and a protracted final path. In contrast, hunters in both Figures 8b and 9b have learned to utilize ocean currents to reduce energy consumption. In addition, the number of turns of the AUV in Figure 8b is significantly reduced, indicating better performance of our method. From the local enlarged view, it is obvious that the paths in Figure 8b are smoother, indicating that our method has better performance in dynamic environments.

A comparative analysis presented in Table 3 reveals that our method achieves the smallest average curvature (0.067), which is 0.079 and 0.092 for the MADDPG and APF methods, respectively. Moreover, our method and MADDPG achieve higher average speeds (1.97 m/s and 1.82 m/s, respectively) and greater cumulative rewards (−94.47 and −109.63, respectively), in contrast to the APF-based method, which records an average

speed of 1.67 m/s and a cumulative reward of −112.35. The average encirclement distance of the proposed method is the smallest (6.32 m), which is less than the average enclosure distances of other methods (7.47 m and 9.14 m). Furthermore, the APF-based method incurs a significantly higher energy consumption (571.75 kJ) and requires a longer duration to complete the task (52.7 s) compared to the other methods. The data in Table 3 validate that the MARL approach can, after adequate training, learn to leverage the direction of ocean currents, thereby enhancing energy efficiency.



(**a**) (**b**)

Figure 10. Path planned by the APF-based method in ocean current environment. (**a**) 3D path planned by the APF-based method; (**b**) 2D projection of the APF-based method. The red lines represent the target's path, while the black lines represent the hunting team's path.

Table 3. Results by three methods in scenario 2.

Algorithm	Consumed Energy (KJ)	Mission Time (S)	Average Speed (M/S)	Average Curvature	Cumulative Reward	Average Encirclement Distance
Our Method	**302.68**	**37.5**	**1.97**	**0.067**	**−112.35**	**6.32**
MADDPG	432.22	44.9	1.82	0.079	−137.58	7.47
APF	571.75	52.7	1.58	0.092	−175.14	9.14

The convergence of reward curves is delineated in Figure 11. The curves indicate that our method not only learns at a faster pace but also attains a superior reward compared to MADDPG. Our method begins to converge after about 750,000 steps, whereas MADDPG only begins to converge after 2,000,000 steps, indicating superior sample efficiency. Moreover, our method significantly outperforms the APF-based method, demonstrating its capacity to exceed the performance of both the MADDPG policy and the APF-based strategy.

To ascertain the efficacy of our method in cooperative hunting endeavors, a comparative analysis was conducted with the APF-based method, BC, and MADDPG. The APF method's parameters are configured equivalently to our KG-RL approach. Given that KG-RL is an extension of the traditional MADDPG, both share identical hyperparameters and reward functions throughout the experiments. BC, a quintessential method for learning from demonstration in reinforcement learning, also utilizes the APF method to source demonstrations, matching the quantity of demonstrations provided to KG-RL. For the evaders, two escape strategies are implemented: one leveraging the APF method, and the other employing the DDPG algorithm.

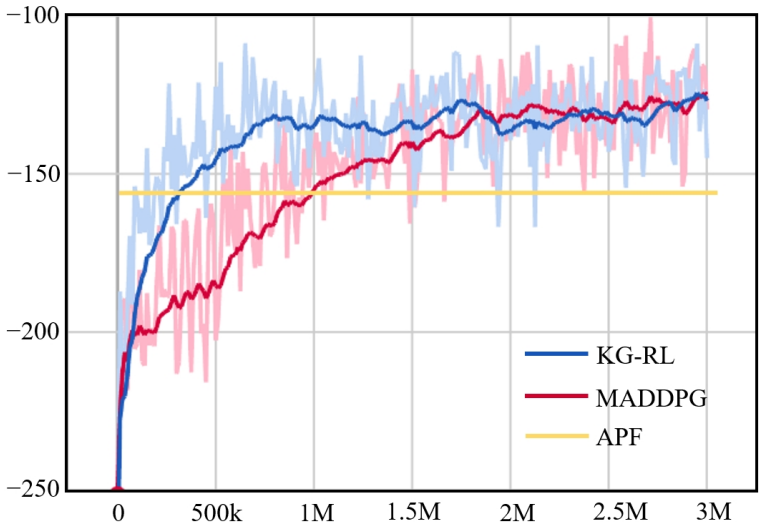


Figure 11. Convergence curves of reward of main experiments.

After the training phase, the policies learned by various algorithms are evaluated. We employ two escape methodologies to create evaluation sets, each consisting of 100 episodes with randomly determined initial states. Each algorithm is assessed over 100 episodes using both escape tactics. The performance of the pursuit is measured by the success rate and the consumed energy. An episode is considered successful if the target is captured within 500 steps. The success rate, calculated as the ratio of successful episodes to total evaluation episodes, primarily indicates the effectiveness of the pursuit strategy. Energy consumption is measured as the total energy depletion of all pursuers within an episode, with lower consumption reflecting a more efficient pursuit policy. After conducting 100 experiments, we recorded the number of successful attempts and energy consumption for each algorithm, and calculated the average energy consumption. Figures 12 and 13 illustrate the average success rates and average energy consumption of the four algorithms across the 100 evaluation episodes, respectively.

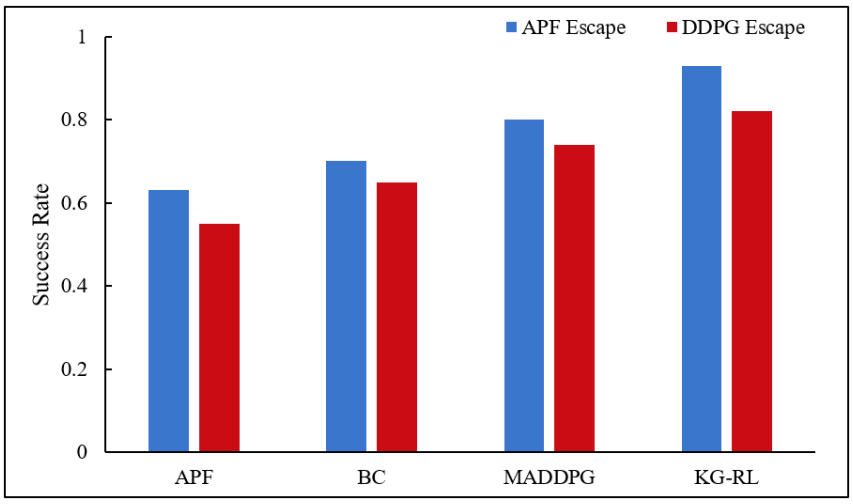


Figure 12. Success rate of 100 experiments.

Analysis of the data presented in Figure 12 indicates that the APF-based method is less effective, yielding the lowest success rate among the evaluated methods. BC, despite its shortcomings in learning policies capable of capturing the evader—owing to the suboptimal policy of the APF demonstrations from which it was derived—achieves a slightly higher success rate than APF. MADDPG, while exhibiting a slower learning curve, attains

a commendable success rate. This is attributable to the global critic's capacity for holistic environmental observation and the communication protocol that facilitates information exchange among pursuers. KG-RL notably surpasses both APF and BC in the evaluation, suggesting that the fine-tuning phase enhancements can elevate policy performance, enabling the final policy to transcend the confines of suboptimal antecedents. Notably, the APF method encounters difficulties when contending with the DDPG-based evasive strategy, a challenge exacerbated by the fact that the other three methods were trained against this DDPG strategy, thereby achieving superior outcomes. Figure 13 illustrates that as the policy gradients align with optimal policies, the system's energy consumption diminishes. The aforementioned evaluation outcomes collectively substantiate that the KG-RL method can markedly augment the efficiency and efficacy of the pursuit policy.

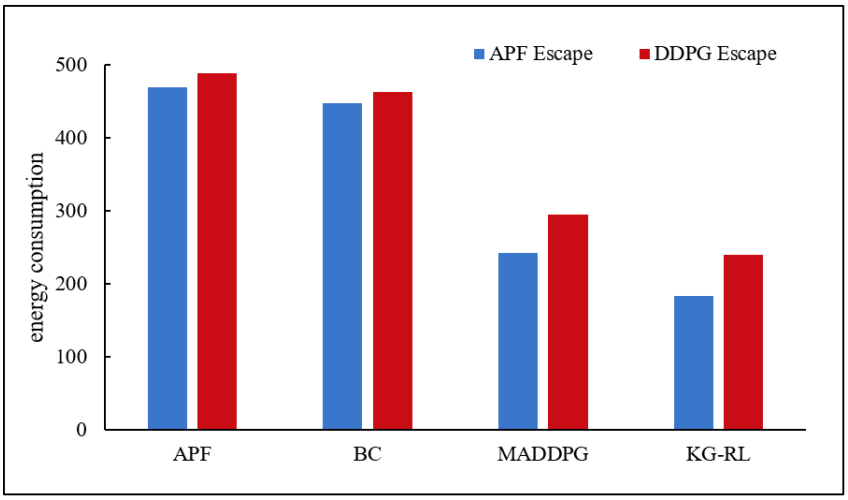


Figure 13. Average energy consumption of 100 experiments.

6. Conclusions

This paper introduces a knowledge-guided reinforcement learning approach, tailored to enhance the sample efficiency and safety of reinforcement learning algorithms in the context of multi-AUV cooperative hunting tasks. Our method integrates the APF with reinforcement learning to expedite the learning process for agents and mitigate the risks inherent in their interactions. Benefiting from the advantages of the generality of DRL, the policies obtained by training with our method are able to adapt to different environments and tasks, achieving superior performance in both static and dynamic underwater environments. In addition, compared to MADDPG, our method utilizes IL in the pre-training phase, which provides a quick start for the subsequent RL process by generating only a small number of expert demonstrations using the APF method, greatly accelerating the learning process. Furthermore, in the fine-tuning phase, by continuously improving the existing policies through online RL, the agents are able to judge whether to mimic the behavior of the APF to make appropriate decisions. Finally, the results of 100 experiments show that compared with the MADDPG algorithm, the proposed method improves the learning speed by 62.5%, the success rate by 16.3%, and the average energy consumption by 24.3%.

Our future work will consider scenarios with a larger number of pursuers and evaders to study the scalability of the proposed method. Experimental validation will be conducted to prove the effectiveness of the algorithm in the real world. Additionally, we plan to explore different expert strategies or MARL algorithms and test them on more benchmarks to improve the effectiveness and generality of the proposed methods. Furthermore, we

will investigate aspects such as inter-agent communication, coordinated cooperation, and conflict resolution among agents to address challenges present in real-world tasks.

Author Contributions: Conceptualization, Y.W. and W.W.; methodology, W.W.; software, W.W.; validation, W.W.; formal analysis, W.W. and D.C.; investigation, W.W.; resources, Y.W. and W.W.; data processing, W.W.; writing—original draft preparation, W.W.; writing—review and editing, W.W. and D.C.; visualization, W.W.; supervision, Y.W. and D.C.; project administration, Y.W. and D.C.; funding acquisition, Y.W. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (2023YFC3010803), the National Science Foundation of China (52372320, 52272424), the Key R&D Program of Hubei Province of China (2023BCB123).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

Abbreviations

APF	Artificial Potential Field
AUV	Autonomous Underwater Vehicle
BC	Behavior Cloning
DRL	Deep Reinforcement Learning
GAIL	Generative Adversarial Imitation Learning
HQPSO	Hybrid Quantum-behaved Particle Swarm Optimization
IL	Imitation Learning
IRL	Inverse Reinforcement Learning
KG-RL	Knowledge-Guided Reinforcement Learning
MADDPG	Multi-agent Deep Reinforcement Learning
MARL	Multi-agent Reinforcement Learning
MDP	Markov Decision Process
NED	North-East-Down

Symbols

$\boldsymbol{u}$	Surge
$\boldsymbol{v}$	Sway
$\boldsymbol{w}$	Heave
$\boldsymbol{p}$	Roll
$\boldsymbol{q}$	Pitch
$\boldsymbol{r}$	Yaw
$\boldsymbol{\pi}$	Policy
$\boldsymbol{\gamma}$	Discount factor
$\boldsymbol{\tau}$	Soft update factor
D	Replay Buffer
Γ	Intensity of the vortex
σ	Radius of the vortex current

References

1. Huang, Z.; Zhu, D.; Sun, B. A Multi-AUV Cooperative Hunting Method in 3-D Underwater Environment with Obstacle. *Eng. Appl. Artif. Intell.* **2016**, *50*, 192–200. [CrossRef]
2. Grinton, C. A Testbed for Investigating Agent E Ectiveness in a Multi-Agent Pursuit Game. Bachelor's Thesis, The University of Melbourne, Melbourne, Australia, 1996.

3. Zhu, D.; Huang, H.; Yang, S.X. Dynamic Task Assignment and Path Planning of Multi-AUV System Based on an Improved Self-Organizing Map and Velocity Synthesis Method in Three-Dimensional Underwater Workspace. *IEEE Trans. Cybern.* **2013**, *43*, 504–514. [PubMed]
4. Liu, M.; Xu, B.; Peng, X. Cooperative Path Planning for Multi-AUV in Time-Varying Ocean Flows. *J. Syst. Eng. Electron.* **2016**, *27*, 612–618. [CrossRef]
5. Xiang, G.; Su, J. Task-Oriented Deep Reinforcement Learning for Robotic Skill Acquisition and Control. *IEEE Trans. Cybern.* **2021**, *51*, 1056–1069. [CrossRef] [PubMed]
6. Liu, Y.; Wang, M.; Su, Z.; Luo, J.; Xie, S.; Peng, Y.; Pu, H.; Xie, J.; Zhou, R. Multi-AUVs Cooperative Target Search Based on Autonomous Cooperative Search Learning Algorithm. *J. Mar. Sci. Eng.* **2020**, *8*, 843. [CrossRef]
7. Liang, H.; Fu, Y.; Gao, J.; Cao, H. Finite-Time Velocity-Observed Based Adaptive Output-Feedback Trajectory Tracking Formation Control for Underactuated Unmanned Underwater Vehicles with Prescribed Transient Performance. *Ocean Eng.* **2021**, *233*, 109071. [CrossRef]
8. Li, Y.; Wu, Y.; He, S. Network-Based Leader-Following Formation Control of Second-Order Autonomous Unmanned Systems. *J. Frankl. Inst.* **2021**, *358*, 757–775. [CrossRef]
9. Sahu, B.K.; Subudhi, B. Flocking Control of Multiple AUVs Based on Fuzzy Potential Functions. *IEEE Trans. Fuzzy Syst.* **2017**, *26*, 2539–2551. [CrossRef]
10. Makavita, C.D.; Jayasinghe, S.G.; Nguyen, H.D.; Ranmuthugala, D. Experimental Study of Command Governor Adaptive Control for Unmanned Underwater Vehicles. *IEEE Trans. Control Syst. Technol.* **2017**, *27*, 332–345. [CrossRef]
11. Lyons, D.M.; Arkin, R.C.; Jiang, S.; Liu, T.-M.; Nirmal, P. Performance Verification for Behavior-Based Robot Missions. *IEEE Trans. Robot.* **2015**, *31*, 619–636. [CrossRef]
12. Ma, Y.; Cao, Z.; Dong, X.; Zhou, C.; Tan, M. A Multi-Robot Coordinated Hunting Strategy with Dynamic Alliance. In Proceedings of the 2009 Chinese Control and Decision Conference, CCDC 2009, Guilin, China, 17–19 June 2009; pp. 2338–2342.
13. Tan, X.; Ni, J.; Shao, L.; Wang, C. An Improved Dynamic Alliance Algorithm for Cooperative Hunting by Multi-Robots. *Sci. Technol. Eng.* **2013**, *13*, 1671–1815.
14. Ge, H.; Chen, G.; Xu, G. Multi-AUV Cooperative Target Hunting Based on Improved Potential Field in a Surface-Water Environment. *Appl. Sci.* **2018**, *8*, 973. [CrossRef]
15. Wei, N.; Liu, M.; Cheng, W. Decision-Making of Underwater Cooperative Confrontation Based on Modpso. *Sensors* **2019**, *19*, 2211. [CrossRef] [PubMed]
16. Ma, X.; Chen, Y.; Bai, G.; Sha, Y.; Zhu, X. Path Planning and Task Assignment of the Multi-AUVs System Based on the Hybrid Bio-Inspired SOM Algorithm with Neural Wave Structure. *J. Braz. Soc. Mech. Sci. Eng.* **2021**, *43*, 28. [CrossRef]
17. Cao, X.; Guo, L. A Leader–Follower Formation Control Approach for Target Hunting by Multiple Autonomous Underwater Vehicle in Three-Dimensional Underwater Environments. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141987066. [CrossRef]
18. Feng, W.; Ma, Y.; Li, H.; Liu, H.; Meng, X.; Zhou, M. Optimal Search Path Planning of UUV in Battlefeld Ambush Scene. *Def. Technol.* **2024**, *32*, 541–552. [CrossRef]
19. Cao, X.; Sun, C.; Yan, M. Target Search Control of AUV in Underwater Environment With Deep Reinforcement Learning. *IEEE Access* **2019**, *7*, 96549–96559. [CrossRef]
20. Cao, X.; Xu, X. Hunting Algorithm for Multi-AUV Based on Dynamic Prediction of Target Trajectory in 3D Underwater Environment. *IEEE Access* **2020**, *8*, 138529–138538. [CrossRef]
21. He, Z.C.; Dong, L.; Sun, C.Y.; Wang, J.W. Asynchronous Multithreading Reinforcement-Learning-Based Path Planning and Tracking for Unmanned Underwater Vehicle. *IEEE Trans. Syst. Man Cybern.* **2022**, *52*, 2757–2769. [CrossRef]
22. Wang, Z.; Wen, Z.W.; Xia, Q.X.; Cai, W.J. Deep Reinforcement Learning Based Multi-UUV Cooperative Control for Target Capturing. In Proceedings of the 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (Dasc/Picom/Cbdcom/Cyberscitech), Falerna, Italy, 12–15 September 2022; pp. 986–991.
23. Xia, J.; Luo, Y.; Liu, Z.; Zhang, Y.; Shi, H.; Liu, Z. Cooperative Multi-Target Hunting by Unmanned Surface Vehicles Based on Multi-Agent Reinforcement Learning. *Def. Technol.* **2023**, *29*, 80–94. [CrossRef]
24. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.; Abbeel, P. Trust Region Policy Optimization. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015; Volume 3, pp. 1889–1897.
25. Mnih, V.; Badia, A.P.; Mirza, L.; Graves, A.; Harley, T.; Lillicrap, T.P.; Silver, D.; Kavukcuoglu, K. Asynchronous Methods for Deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; Volume 4, pp. 2850–2869.
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
27. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2016**. [CrossRef]

28. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep Reinforcement Learning That Matters. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 3207–3214.
29. Haarnoja, T.; Tang, H.; Abbeel, P.; Levine, S. Reinforcement Learning with Deep Energy-Based Policies. In Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, Australia, 6–11 August 2017; Volume 3, pp. 2171–2186.
30. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, 10–15 July 2018; Volume 5, pp. 2976–2989.
31. Nachum, O.; Norouzi, M.; Xu, K.; Schuurmans, D. Bridging the Gap between Value and Policy Based Reinforcement Learning. *arXiv* **2017**. [CrossRef]
32. Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Overcoming Exploration in Reinforcement Learning with Demonstrations. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, QLD, Australia, 21–25 May 2018; pp. 6292–6299.
33. Hester, T.; Schaul, T.; Sendonaris, A.; Vecerik, M.; Piot, B.; Osband, I.; Pietquin, O.; Horgan, D.; Dulac-Arnold, G.; Lanctot, M.; et al. Deep Q-Learning from Demonstrations. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 3223–3230.
34. Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; Abbeel, P. Benchmarking Deep Reinforcement Learning for Continuous Control. In Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York, NY, USA, 19–24 June 2016; Volume 3, pp. 2001–2014.
35. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation Learning: A Survey of Learning Methods. *ACM Comput. Surv.* **2017**, *50*, 1–35. [CrossRef]
36. Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J.A.; Abbeel, P.; Peters, J. An Algorithmic Perspective on Imitation Learning. *Found. Trends® Robot.* **2018**, *7*, 1–179. [CrossRef]
37. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A Survey of Robot Learning from Demonstration. *Rob. Auton. Syst.* **2009**, *57*, 469–483. [CrossRef]
38. Ng, A.; Russell, S. Algorithms for Inverse Reinforcement Learning. In Proceedings of the Seventeenth International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; Volume 1, pp. 663–670.
39. Hausman, K.; Chebotar, Y.; Schaal, S.; Sukhatme, G.; Lim, J.J. Multi-Modal Imitation Learning from Unstructured Demonstrations Using Generative Adversarial Nets. *arXiv* **2017**. [CrossRef]
40. Jones, D.; Hollinger, G.A. Planning Energy-Efficient Trajectories in Strong Disturbances. *IEEE Robot. Autom. Lett.* **2017**, *2*, 2080–2087. [CrossRef]
41. Garau, B.; Alvarez, A.; Oliver, G. AUV Navigation through Turbulent Ocean Environments Supported by Onboard H-ADCP. In Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; Volume 2006, pp. 3556–3561.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.