Multi-Agent Generative Adversarial Interactive Self-Imitation Learning for AUV Formation Control and Obstacle Avoidance

Zheng Fang, Tianhao Chen, Tian Shen, Dong Jiang, Zheng Zhang, and Guangliang Li, *Member, IEEE*

***Abstract*—Multiple autonomous underwater vehicles (multi-AUVs) can cooperatively accomplish tasks that a single AUV cannot complete. Recently, multi-agent reinforcement learning has been introduced to control of multi-AUV. However, designing efficient reward functions for various tasks of multi-AUV control is difficult or even impractical. Multi-agent generative adversarial imitation learning (MAGAIL) allows multi-AUV to learn from expert demonstration instead of pre-defined reward functions, but suffers from the deficiency of requiring optimal demonstrations and not surpassing provided expert demonstrations. This letter builds upon the MAGAIL algorithm by proposing multi-agent generative adversarial interactive self-imitation learning (MAGAISIL), which can facilitate AUVs to learn policies by gradually replacing the provided sub-optimal demonstrations with self-generated good trajectories selected by a human trainer. Our experimental results in three multi-AUV formation control and obstacle avoidance tasks on the Gazebo platform with AUV simulator of our lab show that AUVs trained via MAGAISIL can surpass the provided sub-optimal expert demonstrations and reach a performance close to or even better than MAGAIL with optimal demonstrations. Further results indicate that AUVs' policies trained via MAGAISIL can adapt to complex and different tasks as well as MAGAIL learning from optimal demonstrations.**

***Index Terms*—Multi-agent reinforcement learning, imitation learning, AUV, formation control.**

I. Introduction

AUTONOMOUS underwater vehicle (AUV) plays an important role in underwater tasks of exploring marine resources and scientific research due to its flexibility [1], [2]. It can replace humans to perform dangerous underwater tasks such as survey of ocean topography and landforms, inspection, maintenance and repair of submarine oil pipelines. Reinforcement learning (RL) was introduced and applied to improve the autonomy and intelligence of AUV control [3], [4], [5], [6], [7], [8]. Through interactions with underwater environment, AUV with RL can learn a control policy adapting to the changes and uncertainty [9]. However, as the limited detection range and energy storage of a single AUV, it is necessary to use multi-AUV to cooperatively complete underwater tasks that single AUV cannot perform with the increasing complexity of underwater missions. Multiple AUVs can accomplish underwater detection, target search, object recognition etc., in a collaborative way, which can improve the efficiency of task execution and reduce the time and energy cost. There are many existing methods proposed for multi-agent path planning and obstacle avoidance, such as geometric optimization [10], game theory [11] etc. Combining RL and game theory has been shown to be able to create more intelligent agents capable of playing the game of GO at a higher level than human beings [12]. Multi-agent reinforcement learning (MARL) that is an interdisciplinary domain that includes game theory, machine learning, stochastic control, psychology, and optimisation, has been introduced to improve multi-AUV control in uncertain marine environments [13], [14], [15]. However, it is difficult to design efficient reward functions for various tasks, especially those complex and high-dimensional ones where most robots like AUVs will be operated in. Moreover, the difficulty of designing reward functions for MARL increases with the number of agents and complexity of their relationships [16], [17].

Imitation learning was proposed and successfully applied to robot control [18], [19] since it is much easier to provide demonstrations on performing a task than to design a reward function. There are mainly two kinds of imitation learning: one is behavior cloning (BC) and the other is inverse reinforcement learning (inverse RL). BC learns a mapping from an agent's states to optimal actions via supervised learning [20], but requires a large amount of data and cannot generalize to unseen situations and adapt to different tasks effectively. Inverse reinforcement learning agents learn control policies with extracted cost functions from expert demonstrations via reinforcement learning [21] and can effectively generalize to unseen states [22]. However, many inverse RL algorithms need a model to solve a sequence of planning or reinforcement learning problem in an inner loop and the performance might decrease if the planning or RL problem is not optimally solved [23], which prevents applying inverse RL for robot control to large and complex tasks. Ho et al. solved this problem by proposing a general model-free imitation learning method — generative adversarial imitation learning (GAIL) [23], which allows robots to directly learn policies from expert demonstrations in large and complex environments. Higaki et al. applied GAIL to realize ship's automatic collision avoidance by mimicking human expert performance [24]. Jiang et al. [25] implemented GAIL in AUV path following tasks and

Received 25 September 2024; accepted 11 February 2025. Date of publication 12 March 2025; date of current version 25 March 2025. This article was recommended for publication by Associate Editor H. Ravichandar and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported in part by the Natural Science Foundation of China under Grant 51809246, in part by Qingdao Natural Science Foundation under Grant 23-2-1-153-zyyd-jch, and in part by Young Taishan Scholars Program under Grant tsqn202408072. *(Zheng Fang, Tianhao Chen, and Tian Shen contributed equally to this work.) (Corresponding author: Guangliang Li.)*

The authors are with the College of Electronic Engineering, Ocean University of China, Qingdao 266100, China (e-mail: guangliangli@ouc.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3550743, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3550743

further proposed a generative adversarial interactive imitation learning (GA2IL) method combining GAIL with interactive RL [26], [27] to improve AUV's performance and stability in path following.

GAIL was extended to a multi-agent setting by proposing multi-agent generative adversarial imitation learning (MAGAIL) [28]. Fang et al. [13] successfully applied MAGAIL to a multi-AUV formation control task with a decentralized training and execution framework. However, MAGAIL shares the limitation with GAIL and other imitation learning methods that they assume the optimality of expert demonstrations and can seldom surpass the performance of demonstrations if the provided demonstrations are not optimal. On the other hand, Guo et al. assumed that optimal demonstrations are not available and agents should imitate "relatively better trajectories" generated by the agent. They proposed generative adversarial self-imitation learning (GASIL) [29] by imitating agent's past good trajectories measured via pre-defined reward functions, which violates the initial idea of the GAIL framework learning from solely demonstrations and avoiding pre-defined reward functions.

In this letter, we proposed multi-agent generative adversarial interactive self-imitation learning (MAGAISIL) by improving MAGAIL via replacing the provided expert sub-optimal demonstrations with agent generated good trajectories. However, different from GASIL, MAGAISIL allows a human trainer to evaluate whether the agent generated trajectories are better than the provided expert demonstrations instead of using pre-defined reward functions. Our results in three multi-AUV formation control and obstacle avoidance tasks on the Gazebo platform have shown that our MAGAISIL method with sub-optimal expert demonstrations can learn to reach a performance close to or even better than and generalize as well as those trained via MAGAIL with optimal demonstrations, and learned faster than traditional game-theoretic MARL approach — IPPO.

II. Background

A. Multi-Agent Reinforcement Learning

In multi-agent reinforcement learning (MARL), there are multiple agents interacting with the environment [30], [31]. Each agent i has its own policy π_i that can be used to select an action $a_{i,t}$ based on its observed state s_t at current time step t. Then the agent transitions to a next state and will receive a reward $r_{i,t}$. Similar to single-agent reinforcement learning, the goal for each agent is to learn a policy maximizing its discounted accumulated return. However, different from single-agent reinforcement learning, in MARL, the policy π_i of agent i is affected by other agents' policies. The most common concept to solve this problem is Nash Equilibrium (NE). In NE, agent i will not try to change its policy π_i if other agents do not change their policies, because its discounted accumulated return cannot continue to increase. That is to say, if all agents reach the equilibrium state, each learns a steady optimal policy.

In MARL, the relationship between agents can be divided into three settings based on the relationship between reward functions of agents: cooperative, competitive and a mixed setting [32]. In a fully cooperative setting, all agents perform the same task and share a same reward function. The relationship between reward functions of agents is zero-sum in a competitive setting. In other words, agents maximize their cumulative rewards by preventing each other from completing its task. In a mixed setting, each agent has its own task and reward function, which can be cooperative or competitive to other agents. In our experiments, the relationship between leader and follower AUVs is in a mixed setting since they perform different tasks.

B. Generative Adversarial Imitation Learning

Generative adversarial imitation learning (GAIL) [23] allows an agent to learn directly from expert demonstrations consisting of state-action pairs, avoiding to pre-define reward functions for various tasks. A GAIL agent trained a discriminator $D: S \times A \rightarrow (0, 1)$ to distinguish expert state-action pairs $(s, a) \sim \tau_E$ from agent state-action pairs $(s, a) \sim \tau_{agent}$, and a generator (i.e., policy π) to "fool" the discriminator by generating state-action pairs $(s, a) \sim \tau_{agent}$ as close as possible to expert state-action pairs $(s, a) \sim \tau_E$ by maximizing $\mathbb{E}_\pi[log(D(s, a))]$. The agent generates its trajectory τ_{agent} by interacting with the environment with its current policy π. That is to say, a GAIL agent learns a policy directly by generating a distribution of the agent's state-action pairs as close as possible to the distribution of state-action pairs from the expert demonstrations. In summary, the GAIL algorithm can be summarized as finding a saddle point (π, D):

$$-\lambda H(\pi) + \mathbb{E}_\pi[log(D(s, a))] + \mathbb{E}_{\pi_E}[log(1 - D(s, a))], \quad (1)$$

where $H(\pi) \triangleq \mathbb{E}_\pi[-log\pi(a \mid s)]$, is the γ-discounted causal entropy [33] of the policy π, λ is the weight of entropy $H(\pi)$.

III. Methodology

The MAGAIL method extends GAIL to multi-agent learning and allows multiple agents to learn from provided expert demonstrations [28]. However, MAGAIL shares the limitation with GAIL and other imitation learning methods that they can seldom surpass the performance of demonstrations. On the other hand, generative adversarial self-imitation learning (GASIL) [29] aims to imitate agent's past good trajectories by measuring them via pre-defined reward functions, but violates the initial idea of the GAIL framework to allow learning from demonstrations and avoid pre-defining reward functions. In this letter, we proposed multi-agent generative adversarial interactive self-imitation learning (MAGAISIL) by improving MAGAIL via replacing the expert sub-optimal demonstrations with agent generated good trajectories. However, different from GASIL, MAGAISIL allows a human trainer to evaluate whether the agent generated trajectories are better than the provided expert demonstrations instead of using pre-defined reward functions. Therefore, we expect and hypothesize that our MAGAISIL method allows agents to learn solely from and obtain much better performance than sub-optimal expert demonstrations, resolving the limitation of MAGAIL that it can seldom surpass the performance of demonstrations. Fig. 1 illustrates the mechanism of our proposed MAGAISIL method.

As shown in Fig. 1, MAGAISIL will take provided sub-optimal expert demonstrations as input. Then, each agent will learn an Actor (i.e. control policy) and a Critic (i.e. value function) via independent proximal policy optimization (IPPO) [34]. In addition, a discriminator will be trained to distinguish the state-action pairs of agent's generated trajectories from provided expert demonstrations. Specifically, during training, at time step t, Agent i will obtain local observation $o_{i,t}$, and select an action

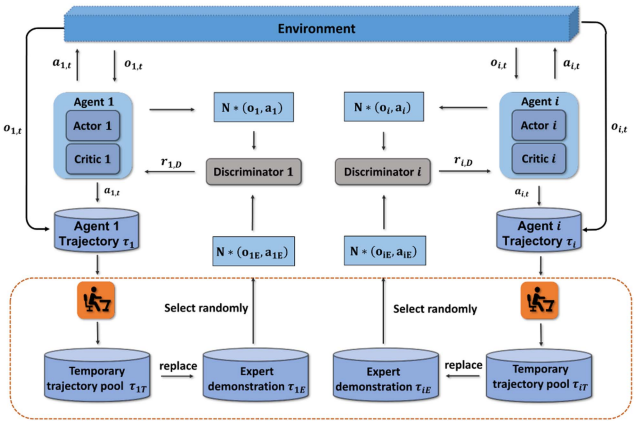

Fig. 1. Illustration of the mechanism for our multi-agent generative adversarial interactive self-imitation learning (MAGAISIL) method.

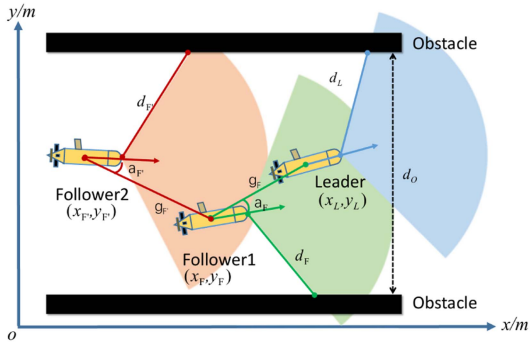

Fig. 2. State representation of the leader AUV and two follower AUVs in the tasks. (x_L, y_L) is the coordinate of the leader AUV's current position, (x_F, y_F) and $(x_{F'}, y_{F'})$ are the coordinates of the follower AUVs.

$a_{i,t}$ with its current policy π_{θ_i}: $a_{i,t} \sim \pi_{\theta_i}(a_i \mid o_{i,t})$. Then, it will transition to a new state upon performing the selected action. The Agent i will repeat the cycle of selecting action and obtaining observation until the end of an episode. The received state-action pairs during one episode by interacting with the environment compose the trajectory τ_i of Agent i. N state-action pairs (o_i, a_i) from the agent's trajectory τ_i will be selected and N state-action pairs (o_{iE}, a_{iE}) from the provided expert trajectory τ_{iE} will be selected and used to train the discriminator D_{ω_i} via ADAM [35] with the loss function as:

$$\mathbb{E}_{\tau_i}\left[\log\left(D_{\omega_i}(o,a)\right)\right] + \mathbb{E}_{\tau_{iE}}\left[\log\left(1 - D_{\omega_i}(o,a)\right)\right]. \quad (2)$$

The updated discriminator D_{ω_i} will be used to provide rewards $r_{i,D}$ for updating the Actor and Critic of Agent i:

$$r_{i,D} = -\log\left(1 - D_{\omega_i}(o,a)\right). \quad (3)$$

In addition, at the end of an episode, the trajectory τ_i of Agent i will be visualized in a window and shown to a human trainer, who can compare with the provided expert demonstrations according to her knowledge and experience. If the human trainer thinks the agent's generated trajectory is better than the expert demonstrations, the trajectory τ_i of Agent i will be stored in the temporary trajectory pool τ_{i_T}. Otherwise, the trajectory τ_i of Agent i will be disregarded. We set a limit to the number of trajectories in the pool τ_{i_T} and when it is full, all stored trajectories in τ_{i_T} are used to replace the current expert demonstrations τ_{i_E}. At the same time, τ_{i_T} will be cleared to store new trajectories in the following training process.

IV. Simulation Setup

We evaluated our method by conducting experiments in three formation control and obstacle avoidance of multi-AUV tasks on the Gazebo simulation platform. The simulator is extended from Unmanned Underwater Vehicle Simulator [36] with the model of Sailfish 210 developed in our lab.in the simulated underwater environment.

A. Simulation Tasks

We set up three formation control and obstacle avoidance tasks in our experiments. Fig. 2 shows the observed state information of a leader AUV and two follower AUVs in the tasks. In Task I, the objective of the task is to allow the leader AUV to go through a square pipe, with two follower AUVs following the leader AUV in a line at a distance of 18 meters, while keeping a safe distance from the walls on both sides of the pipe. The distance between walls of two sides in the pipe is 30 meters. The leader AUV, follower AUV1 and follower AUV2 will start from the position $(18, 15)$, $(9, 15)$ and $(1, 15)$, respectively, and the ending position is at $(240, 15)$. The task is terminated and a new episode will start in the following situations:

1) Leader AUV or follower AUV is too close to obstacles, i.e.,$|d_L| \leq 2$ or $|d_F| \leq 2$ or $|d_{F'}| \leq 2$
2) The distance between two AUVs is too close or far, i.e.,$|g_F| < 3$ or $|g_F| > 33$ or $|g_{F'}| < 3$ or $|g_{F'}| > 33$
3) The heading deviation of follower AUVs is too large, i.e.,$|a_F| \geq \frac{\pi}{3}$ or $|a_{F'}| \geq \frac{\pi}{3}$.

To test the adaptability of our method in complex tasks, we added dense spherical obstacles with a radius of 5 meters in various places (e.g. in the middle, close to the corner and the walls) of the pipe in Task II, and changed angles of walls and extended the pipe to be 300 meters long in Task III.

B. State Representation and Action Space

In the tasks, the leader-follower method was adopted and a leader AUV with two follower AUVs were considered for simplification, which can be easily extended to complex tasks with more AUVs. A decentralized training and execution framework was used as in [13]. Fig. 2 shows the state representation of AUVs in the task. As shown in Fig. 2, the black squares represent the walls of pipe or obstacles, d_o denotes the distance between walls of the pipe, which is set to be 30 meters. All AUVs need to detect and avoid collision with the wall and/or obstacles with sonar sensor and go through the pipe as soon as possible. The detection angle range of the sonar sensor is set to be $[-\frac{\pi}{3}, \frac{\pi}{3}]$ which will be divided into 6 sectors, as shown by the shaded areas in Fig. 2. The sonar sensor has 600 beams which are equally distributed in the 6 sectors. The detection distance is 33 meters at most. AUV will take the shortest distance detected by all the sectors as the distance to the obstacle. The detected shortest distances by the leader AUV, follower AUV1 and follower AUV2 are denoted as d_L, d_F and $d_{F'}$, respectively. The leader AUV's state is represented as $o_L = \{d_{L1}, d_{L2}, d_{L3}, d_{L4}, d_{L5}, d_{L6}\}$, and the two follower AUVs' states are represented in a similar format as $o_{F(')} = \{g_{F(')}, a_{F(')}, d_{F(')1}, d_{F(')2}, d_{F(')3}, d_{F(')4}, d_{F(')5}, d_{F(')6}\}$. Here, d_{Li} and $d_{F(')i}$ are the detected distance by each sector of the

TABLE I
ACTION SETUP FOR AUVS TO PERFORM BY SETTING THE ANGLES OF FOUR RUDDERS (UPPER, RIGHT, LOWER AND LEFT)

Action	Upper	Right	Lower	Left
turn left 1	$-14°$	0	$14°$	0
turn left 2	$-20°$	0	$20°$	0
go straight	0	0	0	0
turn right 1	$14°$	0	$-14°$	0
turn right 2	$20°$	0	$-20°$	0

sonar sensor on leader AUV and follower AUVs, respectively, $i = 1, 2, \ldots, 6$. $g_{F(')}$ is the directrix distance between AUVs, e.g., g_F is the distance between the leader AUV and follower AUV1 and is computed as $g_F = \sqrt{(y_F - y_L)^2 + (x_F - x_L)^2}$. $a_{F(')}$ is the heading deviation of follower AUVs, e.g., the heading deviation a_F of follower AUV1 is computed as $a_F = a_{FH} - \operatorname{atan}2(\frac{y_F - y_L}{x_F - x_L})$, where a_{FH} denotes the current heading of the follower AUV1.

We set five discrete actions for all AUVs, which can be performed by setting the thruster speed and angles of four rudders, including two actions for turning left, two for turning right and one for going straight, as shown in Table I. The upper and lower rudders are set to control the horizontal direction of AUV. The left and right rudders are used to control AUV to float and dive in the underwater environment, which are set to 0 as the tasks are in a 2D space. The thruster speed of AUV is set to 300 r/s.

C. Evaluation Metrics

Due to the subjectivity of the rewards of the learned discriminator with expert demonstrations, we defined reward functions for all AUVs to evaluate our proposed method. The defined reward functions for all AUVs are never used for learning, but only used for testing the learned policies from demonstrated trajectories with MAGAIL and our method MGAISIL. The reward function for the leader AUV is defined as:

$$r_L = 1 - \frac{|d_L - 17.3|}{8.65}, \tag{4}$$

where 17.3 meters is a safe distance for AUV derived based on the detection angle range of the sonar sensor and the distance between walls of the pipe, which can keep AUV in the middle of the pipe. The reward functions for the follower AUVs are defined as:

$$r_{F(')} = 0.5 * r^c_{F(')} + 0.5 * r^a_{F(')}, \tag{5}$$

where $r^c_{F(')}$ represents the rewards for tracking the leader AUV or previous follower AUV1, and is defined as $r^c_{F(')} = -\frac{|a_{F(')}|*3}{\pi} + |1 - \frac{|g_{F(')} - 18|}{15}|$ based on the distance $g_{F(')}$ between AUVs and the heading deviation $a_{F(')}$. Similar to the leader AUV, $r^a_{F(')} = 1 - \frac{|d_F - 17.3|}{8.65}$ are defined based on the distance from the wall of pipe $d_{F(')}$ to keep the follower AUVs in the middle of the pipe. Even we could consider all these aspects to set an appropriate reward function in the leader-follower task, it is very time-consuming and requires much more expertise than providing sub-optimal demonstrations.

In our experiments, we trained both leader and follower AUVs with our MAGAISIL method with sub-optimal expert demonstrations. In addition, MAGAIL trained with both optimal and sub-optimal expert demonstrations and traditional game-theoretic MARL method IPPO learning from the above defined reward functions were also used as comparisons. Each method has a policy network and a critic network, which were represented with fully connected neural networks (FCNs). The structure of FCN is: FCN (actor) = [64,128,64,5,5] and FCN (critic) = [64,128,64,1], the activate functions of the hidden and last layer are Tanh and Softmax respectively. No activate function for the last layer of the critic network. We set $N = 256$, i.e., each time 256 state-action pairs are randomly selected from trajectory generated by each AUV and provided expert demonstrations respectively to update the discriminator. During one episode, the discriminator is updated 3 times and the generator is updated 9 times for each AUV to reduce the variation of the policy caused by the discriminator's changes. The discount factor is set to be $\gamma = 0.99$, $\lambda = 1.0$ and the clipping factor $\epsilon = 0.09$ in the IPPO algorithm. The maximum number of trajectories in the temporary trajectory pool is set to be 10.

During training, at the end of each episode, the AUV's current trajectory and its four indices including average distance to the wall from both leader AUV and follower AUV, distance between leader AUV and follower AUV, angle between leader AUV and follower AUV, are shown in a pop-up window for evaluation by the human trainer. The human trainer needs to be aware of the following metrics (i.e. the task requirements) to evaluate whether the AUV's current trajectory is good or not:

1) The average distance from leader AUV and follower AUV to the wall is close to 17.3 meters, respectively.
2) The heading deviation between leader AUV and follower AUV is close to 0.
3) The distance between two AUVs is around 18 meters.
4) The smoothness of the trajectory.

The closer are the four indices of an AUV's trajectory to these metrics and the more natural is the AUV's trajectory, the more likely is it to be selected by the human trainer and added to the temporary trajectory pool. That is to say, the human trainer makes decisions not only based on these four objective metrics but also her subjective judgments such as the naturalness and smoothness of the trajectory, since two AUV trajectories might have the same or similar performance in terms of the four objective metrics but one might be smoother than the other. Since the human trainer only needs to evaluate at the end of one episode during training, it generally took only a few seconds for each evaluation as long as the human trainer is familiar with the task requirements.

V. RESULTS AND DISCUSSION

This section presents and analyzes our experimental results by comparing the policy performance trained with our MAGAISIL learning from sub-optimal expert demonstrations to MAGAIL learning from both optimal and sub-optimal expert demonstrations and IPPO from defined reward functions in Task I as described in Section IV-A. In addition, to evaluate the adaptability of our MAGAISIL method, we tested the trained policies of three AUVs in Task II and III. Note that the results of follower AUV2 are similar to those of follower AUV1 since they both taking the same role by tracking the follower AUV1 and leader AUV respectively, and not shown in the letter due to limited space.

A. Learning Curves

Fig. 3(a) and (b) shows the cumulative rewards received by the leader AUV and follower AUV1 during training via IPPO, MAGAISIL with sub-optimal demonstrations, MAGAIL with

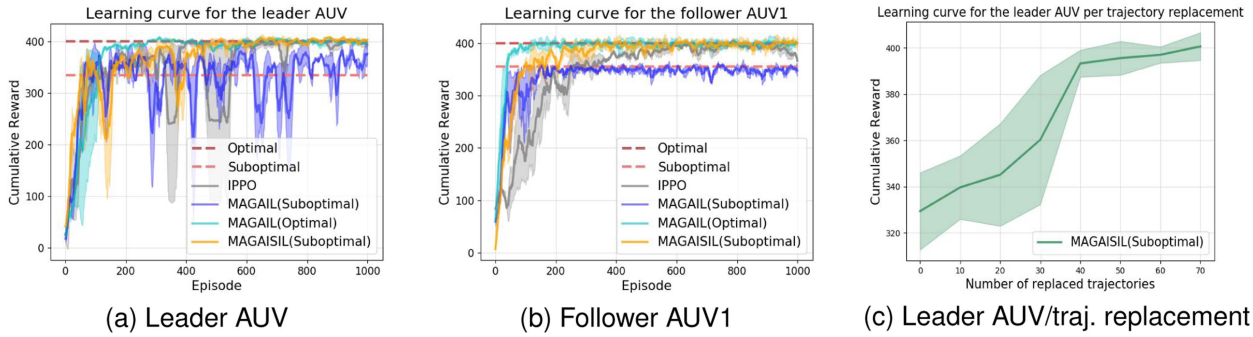

Fig. 3. Cumulative rewards received by the leader AUV (a) and follower AUV1 (b) trained via MAGAISIL with suboptimal demonstrations (MAGAISIL Suboptimal), MAGAIL with sub-optimal (MAGAIL Suboptimal) and optimal (MAGAIL Optimal) demonstrations, and IPPO learning from predefined reward functions in Task I. (c) shows the leader AUV's learning curve along the number of trajectories being replaced. The shaded area is the 0.95 confidence interval and the bold line is the mean performance over three experimental trials. Two red lines show the performance of expert optimal and sub-optimal demonstrations.

sub-optimal and optimal demonstrations in Task I, measured by the predefined reward functions for leader and follower AUVs in Section IV-C, respectively. From Fig. 3 we can see that, for both leader and follower AUVs, at the beginning process, the speed of our MAGAISIL agent learning from sub-optimal demonstrations is similar to the MAGAIL agent learning from sub-optimal demonstrations, which is slower than the MAGAIL agent learning from optimal demonstrations but faster than the IPPO agent. This might be because the good agent-generated trajectories selected by the human trainer in our MAGAISIL method did not fully replace the sub-optimal expert demonstrations yet. However, after about 400 episodes' training, our MAGAISIL agent learning from sub-optimal demonstrations reached a performance close to optimal expert demonstrations together with the MAGAIL agent learning from optimal demonstrations and stabilized afterwards. In contrary, the performance of the MAGAIL agent learning from sub-optimal demonstrations and IPPO agent still fluctuated around/above the sub-optimal demonstrations throughout the training process.

We also studied the impact of replacing original trajectories with self-generated ones on the performance of AUVs, as shown in Fig. 3(c). From Fig. 3(c) we can see that the performance of the leader AUV gradually improves as self-generated trajectories replace the original sub-optimal demonstrations, and after replacing 70 trajectories, the performance of the leader AUV tends to converge to optimal.

In summary, while the performance of MAGAIL agent learning from sub-optimal demonstrations is limited by sub-optimal expert demonstrations, our MAGAISIL agent can learn to reach a performance close to optimal demonstrations faster than traditional game-theoretic MARL IPPO method via gradually replacing the sub-optimal demonstrations with self-generated good trajectories selected by a human trainer.

B. Performance

We also tested and compared the final control policies of the leader AUV and follower AUVs trained in Task I via our MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations for 10 times from the perspectives of trajectory, distance to the walls of pipe or obstacles, distance between leader AUV and follower AUV1, heading deviation of follower AUV1, as shown in Table II, Figs. 4(a), 5(a), 6(a), 7(a), and 8(a). From Fig. 4(a) we can see that, the leader AUV and follower AUV1 trained via our MAGAISIL with sub-optimal demonstrations and MAGAIL with optimal demonstrations can successfully complete Task I by generally following the middle of the pipe. The leader AUV and follower AUV1 trained via our MAGAISIL method even performed a bit better than the one via MAGAIL at the turnings in the pipe. While further examining the observations of both leader AUV and follower AUV1 in the testing process, we found that although the leader AUV in the provided sub-optimal expert demonstrations fluctuated dramatically around the middle of the pipe after turning, the leader AUV trained with our MAGAISIL method can get a performance close to MAGAIL learning from optimal demonstrations, which can immediately get back to the safe distance around 17.3 meters (derived based on the detection angle range of the sonar sensor and the distance between walls of the pipe, refer to Section IV-C), and keep itself in the middle of the pipe (Fig. 5(a)).

For the follower AUV1 in the provided sub-optimal expert demonstrations, its distance to the leader AUV fluctuated dramatically between 12 and 25 meters and heading deviation also fluctuated dramatically (Figs. 6(a) and 7(a)). After training with our MAGAISIL method, the follower AUV1 can obtain a performance close to MAGAIL learning from optimal demonstrations, and keep a distance to the leader AUV at around 18 meters with a heading deviation significantly lower than that of MAGAIL (Table II, $p < 0.01$ via student t-test). Moreover, the follower AUV1 trained with our MAGAISIL method can keep a safe distance to the walls of pipe even though it fluctuated largely during and after turnings in the provided sub-optimal expert demonstrations, and even performed better than the MAGAIL agent learning from optimal demonstrations (Fig. 8(a)).

In summary, our results show the leader AUV, follower AUV1 and follower AUV2 trained with our MAGAISIL method can surpass the provided sub-optimal expert demonstrations and get a performance close to or even better than those trained via MAGAIL with optimal demonstrations.

C. Adaptability to Complex and Different Tasks

We also tested and compared the adaptability of our MAGAISIL method to complex and different tasks by running the saved final control policies of leader AUV, follower

TABLE II
TESTED MEAN PERFORMANCE OF THE LEADER AND FOLLOWER AUV1 OVER 10 TRIALS IN THE THREE TASKS.

Metrics	Task I		Task II		Task III	
	MAGAISIL	MAGAIL	MAGAISIL	MAGAIL	MAGAISIL	MAGAIL
Leader AUV-Distance to Obstacle	18.07 ± 0.13	17.94 ± 0.12	14.81 ± 0.11	14.95 ± 0.08	16.29 ± 0.10	16.21 ± 0.11
Follower AUV1-Distance to Obstacle	14.55 ± 0.24	13.89 ± 0.12	13.42 ± 0.17	13.26 ± 0.16	12.97 ± 0.24	13.76 ± 0.13
Follower AUV1-Distance to Leader AUV	17.73 ± 0.27	18.38 ± 0.20	14.23 ± 0.54	16.84 ± 0.42	21.08 ± 0.30	18.95 ± 0.19
Follower AUV1-Heading Deviation	0.010 ± 0.004	0.026 ± 0.006	0.008 ± 0.006	0.018 ± 0.007	0.016 ± 0.004	0.028 ± 0.007

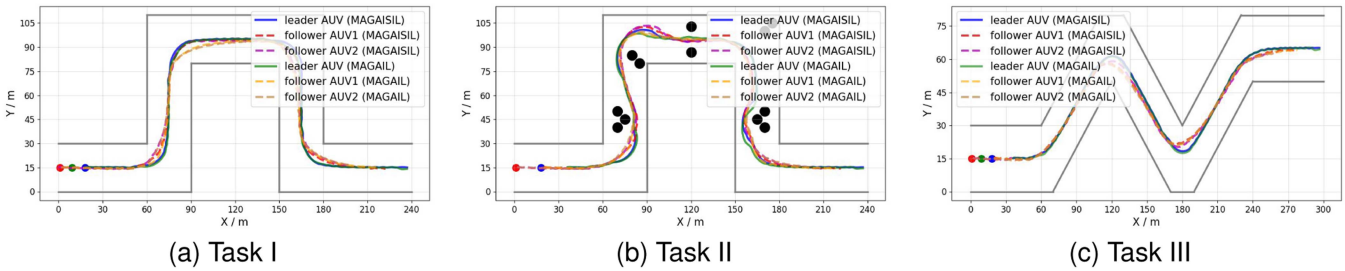

Fig. 4. Tested trajectories of the leader AUV, follower AUV1 and follower AUV2 in Task I, II and III using final control policies trained in Task I via MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations.

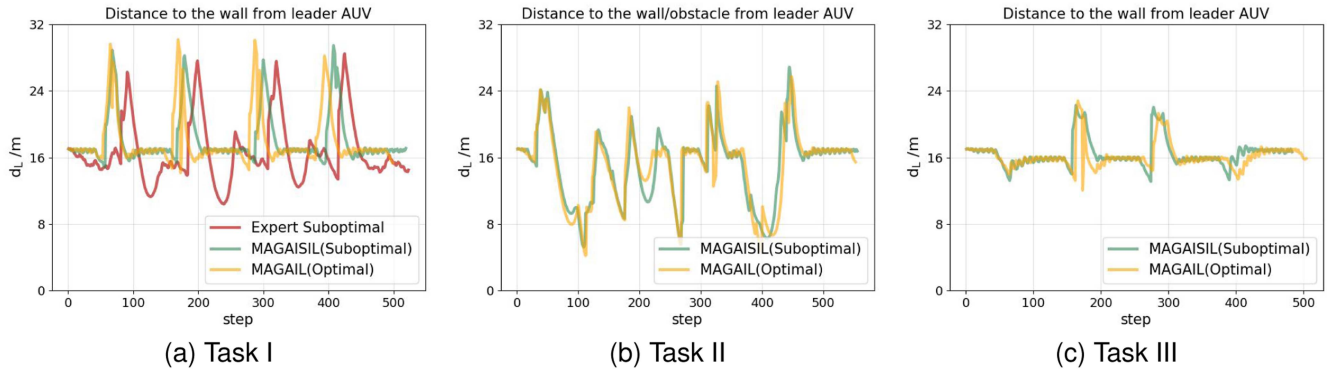

Fig. 5. Distance to the wall of pipe or obstacles from the leader AUV in Task I, II and III tested using final control policies trained in Task I via MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations. The red line in (a) shows the distance to the wall of pipe from the leader AUV in the sub-optimal expert demonstration.

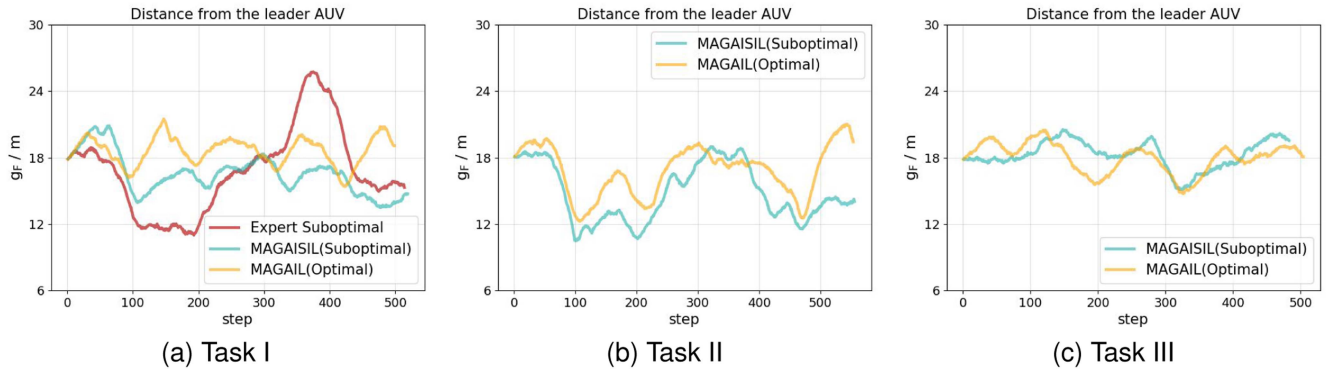

Fig. 6. Distance between the leader AUV and follower AUV1 in Task I, II, and III tested using final control policies trained in Task I via MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations. The red line in (a) shows distance between the leader AUV and follower AUV1 in the sub-optimal expert demonstration.

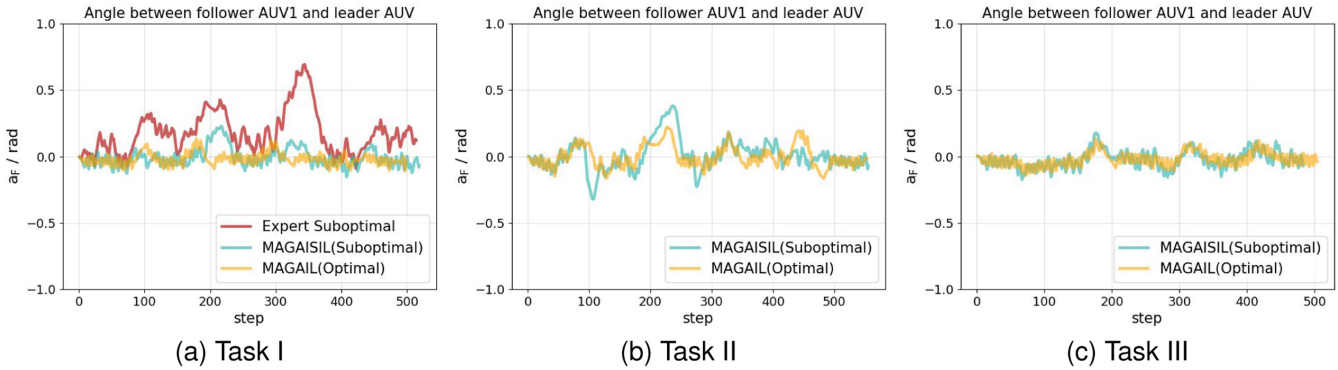

Fig. 7. The heading deviation of the follower AUV1 in Task I, II, and III tested using final control policies trained in Task I via MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations. The red line in (a) shows heading deviation of the follower AUV1 in the sub-optimal expert demonstration.

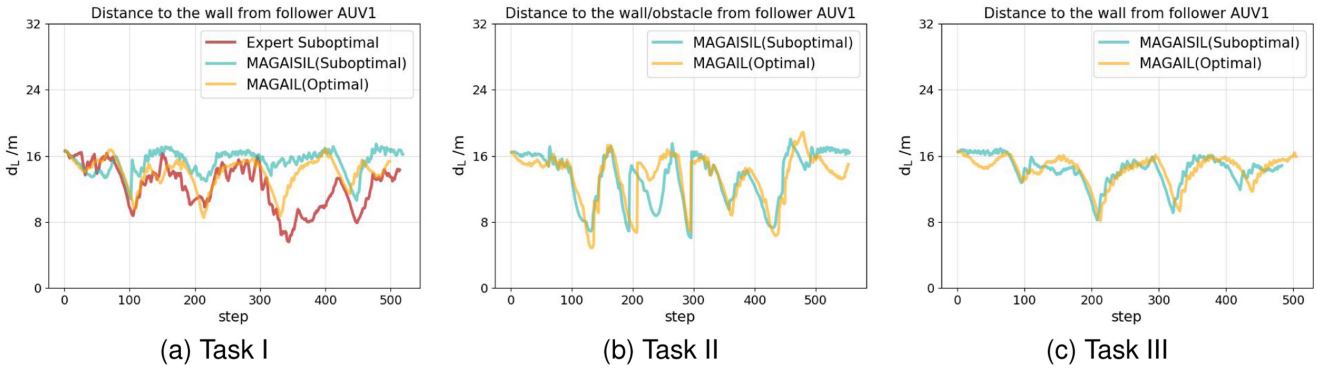

Fig. 8. The distance to the wall of pipe or obstacles from the follower AUV1 in Task I, II, and III tested using final control policies trained in Task I via MAGAISIL with sub-optimal expert demonstrations and MAGAIL with optimal expert demonstrations. The red line in (a) shows the distance to the wall of pipe from the follower AUV2 in the sub-optimal expert demonstration.

AUV1 and AUV2 trained in Task I via MAGAISIL with sub-optimal demonstrations and via MAGAIL with optimal demonstrations for 10 times in Task II with extra obstacles and in Task III with changed angles of the walls and extended length of the pipe. Other settings are the same as Task I.

Table II shows the mean performance and Figs. 4(b), (c), 5(b), (c), 6(b), (c), 7(b), (c), and 8(b), (c) show the example performances of the leader AUV and the follower AUV1 trained via MAGAISIL and MAGAIL in terms of trajectory, distance to the walls of pipe or obstacles, distance between leader AUV and follower AUV1, heading deviation of follower AUV1, respectively. From these results in Task II and III we can see that, the control policies of leader AUV and follower AUVs trained via MAGAISIL and MAGAIL can adapt well to complex and different tasks even with added obstacles or changed angles of wall. Moreover, the leader and follower AUVs trained via MAGAISIL with sub-optimal demonstrations can obtain a similar performance to those trained via MAGAIL with optimal demonstrations. The leader AUV and follower AUV1 trained via MAGAISIL even performed a bit better than those via MAGAIL at the turnings in the pipe (Fig. 4(b) and (c)). However, the distance between leader AUV and follower AUV1 trained via MAGAISIL decreased largely after 100 steps compared to those via MAGAIL, but gradually increased after that and is similar to those trained via MAGAIL after about 300 steps (Fig. 6(b)). This might be because of the effect of added obstacles, which were first met after the first turning at about 100 steps. This is consistent with the a bit larger fluctuation of the heading deviation of the follower AUV1 trained via MAGAISIL compared to MAGAIL, which also starts from about 100 steps (Fig. 7(b)).

VI. Conclusion

In this letter, we builds upon the MAGAIL algorithm by proposing multi-agent generative adversarial interactive self-imitation learning (MAGAISIL), which can facilitate agents to learn policies by gradually replacing the provided sub-optimal demonstrations with self-generated good trajectories selected by a human trainer. Results in three multi-AUV formation control and obstacle avoidance tasks on the Gazebo platform show that AUVs trained via MAGAISIL can surpass the provided sub-optimal expert demonstrations and learn to reach a performance close to or even better than those trained via MAGAIL with optimal demonstration. Further analysis indicates MAGAISIL adapts to complex and different tasks as well as MAGAIL learning from optimal demonstrations.

Our method can easily generalize to other multi-agent domains beyond AUVs, such as unmanned aerial vehicles (UAVs), robotic swarms for search and rescue operations, and autonomous ground vehicles for logistics and transportation etc.

However, the workload and time needed for the human trainer to train all agents will increase dramatically as the number of agents increases in the task. Future work will focus on studying methods using large language models to evaluate the agent's trajectory, which has proven to be able to provide evaluative feedback to train RL agents [37]. In addition, safe reinforcement learning from human feedback [38] with formal interpretability [39], stability [40] to guarantee low-bound performance, and Responsibility Sensitive Safety (RSS) model [41] could be considered to integrate with our method together to monitor and safeguard the risk of AUVs or other robots during operation using learned model with our method.

REFERENCES

[1] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 131–149, Jan. 2014.

[2] C. Cheng, Q. Sha, B. He, and G. Li, "Path planning and obstacle avoidance for AUV: A review," *Ocean Eng.*, vol. 235, 2021, Art. no. 109355.

[3] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[4] A. El-Fakdi and M. Carreras, "Two-step gradient-based reinforcement learning for underwater robotics behavior learning," *Robot. Auton. Syst.*, vol. 61, no. 3, pp. 271–282, 2013.

[5] Q. Zhang, J. Lin, Q. Sha, B. He, and G. Li, "Deep interactive reinforcement learning for path following of autonomous underwater vehicle," *IEEE Access*, vol. 8, pp. 24258–24268, 2020.

[6] R. B. Grando et al., "Deep reinforcement learning for mapless navigation of a hybrid aerial underwater vehicle with medium transition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1088–1094.

[7] B. Hadi, A. Khosravi, and P. Sarhadi, "Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle," *Appl. Ocean Res.*, vol. 129, 2022, Art. no. 103326.

[8] C. Zhang, P. Cheng, B. Du, B. Dong, and W. Zhang, "AUV path tracking with real-time obstacle avoidance via reinforcement learning under adaptive constraints," *Ocean Eng.*, vol. 256, 2022, Art. no. 111453.

[9] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, and G. G. Acosta, "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning," *Robot. Auton. Syst.*, vol. 107, pp. 71–86, 2018.

[10] S. J. Guy, J. Van Den Berg, M. C. Lin, and D. Manocha, "Geometric methods for multi-agent collision avoidance," in *Proc. 26th Annu. Symp. Comput. Geometry*, 2010, pp. 115–116.

[11] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205–2213, 2009.

[12] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[13] Z. Fang et al., "Autonomous underwater vehicle formation control and obstacle avoidance using multi-agent generative adversarial imitation learning," *Ocean Eng.*, vol. 262, 2022, Art. no. 112182.

[14] G. Wang, F. Wei, Y. Jiang, M. Zhao, K. Wang, and H. Qi, "A multi-auv maritime target search method for moving and invisible objects based on multi-agent deep reinforcement learning," *Sensors*, vol. 22, no. 21, 2022, Art. no. 8562.

[15] C. Lin, G. Han, T. Zhang, S. B. H. Shah, and Y. Peng, "Smart underwater pollution detection based on graph-based multi-agent reinforcement learning towards AUV-based network its," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7494–7505, Jul. 2023.

[16] D. Gu and E. Yang, "Multiagent reinforcement learning for multi-robot systems: A survey," Technical Report of the Department of Computer Science, 2004.

[17] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," *Innovations Multi-agent Syst. Appl.-1*, pp. 183–221, 2010.

[18] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[19] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annu. Rev. Control Robot., Auton. Syst.*, vol. 3, pp. 297–330, 2020.

[20] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proc. 13th Int. Conf. Artif. Intell. Statist. Workshop Conf. Proc.*, 2010, pp. 661–668.

[21] A. Y. Ng et al., "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2000, Art. no. 2.

[22] J. Ho, J. Gupta, and S. Ermon, "Model-free imitation learning with policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2760–2769.

[23] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4565–4573.

[24] T. Higaki and H. Hashimoto, "Human-like route planning for automatic collision avoidance using generative adversarial imitation learning," *Appl. Ocean Res.*, vol. 138, 2023, Art. no. 103620.

[25] D. Jiang et al., "Generative adversarial interactive imitation learning for path following of autonomous underwater vehicle," *Ocean Eng.*, vol. 260, 2022, Art. no. 111971.

[26] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 4, pp. 337–349, Aug. 2019.

[27] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4300–4308.

[28] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7472–7483.

[29] Y. Guo, J. Oh, S. Singh, and H. Lee, "Generative adversarial self-imitation learning," 2018, *arXiv:1812.00950*.

[30] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[31] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 146264–146272, 2019.

[32] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*. Berlin, Germany: Springer, pp. 321–384, 2021.

[33] M. Bloem and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 4911–4916.

[34] C. Schroeder De Witt et al., "Is independent learning all you need in the starcraft multi-agent challenge?," 2020, *arXiv:2011.09533*.

[35] Z. Zhang, "Improved Adam optimizer for deep neural networks," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Serv.*, 2018, pp. 1–2.

[36] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV simulator: A Gazebo-based package for underwater intervention and multi-robot simulation," in *Proc. MTS/IEEE Monterey*, 2016, pp. 1–8.

[37] K. Chu, X. Zhao, C. Weber, M. Li, and S. Wermter, "Accelerating reinforcement learning of robotic manipulations via feedback from large language models," in *Proc. Conf. Robot Learn. Workshop*, 2023.

[38] J. Dai et al., "Safe RLHF: Safe reinforcement learning from human feedback," in *Proc. 12th Int. Conf. Learn. Representations*, 2024.

[39] Y. Wang, Q. Qian, and D. Boyle, "Probabilistic constrained reinforcement learning with formal interpretability," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024, pp. 51303–51327.

[40] T. Xu, Y. Liang, and G. Lan, "Crpo: A new approach for safe reinforcement learning with convergence guarantee," in *Proc. 38 th Int. Conf. Mach. Learn.*, 2021, pp. 11480–11491.

[41] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," 2018. *arXiv:1708.06374*.