Is FISHER All You Need in The Multi-AUV Underwater Target Tracking Task?

Guanwen Xie*, *Student Member, IEEE*, Jingzehua Xu*, *Student Member, IEEE*, Ziqi Zhang, Xiangwang Hou, *Student Member, IEEE*, Dongfang Ma, Shuai Zhang, Yong Ren, *Senior Member, IEEE* and Dusit Niyato, *Fellow, IEEE*

***Abstract*—It is significant to employ multiple autonomous underwater vehicles (AUVs) to execute the underwater target tracking task collaboratively. However, it's pretty challenging to meet various prerequisites utilizing traditional control methods. Therefore, we propose an effective two-stage learning from demonstrations training framework, FISHER, to highlight the adaptability of reinforcement learning (RL) methods in the multi-AUV underwater target tracking task, while addressing its limitations. The first stage utilizes imitation learning (IL) to realize policy improvement and generate offline datasets. To be specific, we introduce multi-agent discriminator-actor-critic based on improvements of the generative adversarial IL algorithm and multi-agent IL optimization objective derived from the Nash equilibrium condition. Then in the second stage, we develop multi-agent independent generalized decision transformer, which analyzes the latent representation to match the future states of high-quality samples rather than reward function, attaining further enhanced policies capable of handling various scenarios. Besides, we propose a simulation to simulation demonstration generation procedure to facilitate the generation of expert demonstrations in underwater environments, which capitalizes on traditional control methods and can easily accomplish the domain transfer to obtain demonstrations. Extensive simulation experiments from multiple scenarios showcase that FISHER possesses strong stability, multi-task performance and capability of generalization.**

***Index Terms*—Autonomous underwater vehicle, multi-agent reinforcement learning, learning from demonstrations, simulation to simulation**

I. Introduction

Autonomous underwater vehicle (AUV) [1]swarm has broad application prospects in underwater rescue, constructing seamless communication networks, and target tracking [2], with its broader detection range and strong maneuverability, compared to the single AUV scenario. Target tracking is a representative issue for swarm control, which places high demands on the performance of target proximity to the swarm, consistency between AUVs, avoidance of obstacles and collisions between AUVs, and more if necessary. Therefore, traditional approaches, such as methods based on Lyapunov vector fields, artificial potential field (APF), and model predictive control (MPC), typically have to make lots of mathematical simplification, making them lack generality and practicality.

Due to its strong ability to feature expression and meet demands, reinforcement learning (RL) provides an efficient solution to tackle these requisites and achieve effective tracking. For example, Yang *et al.* [3] took the original data of sensors as state and directly output control signals such as propeller thrust, which effectively overcomes the complex influence of the underwater environment. Besides, Wang *et al.* [4] took lots of demands into consideration, such as energy costs and information sharing. These researches have demonstrated the significant utility in target tracking problems. However, there are some challenges when applying RL. To be specific, the performance of agents strongly relies on the design of the reward function. Otherwise, detrimental outcomes, such as sub-optimal policies and reward hacking, may be produced [5]. A well-designed reward function must have tight monotonic correlations with the optimization objectives, which can not be satisfied as the number of objectives or agents increases. Besides, abundant interactions with the environment are required, which leads to high costs of time and computing resources, or even not feasible because of the high risk [6], [7].

The booming development of learning from demonstrations (LfD) in recent years, especially in imitation learning (IL) and offline reinforcement learning (ORL), provides feasible solutions to tackle the challenges of RL [8], [9]. IL requires the policy to learn to perform a task from limited demonstrations. The mainstream IL methods currently comply with the perspective of generative adversarial IL (GAIL) [10], which introduces a discriminator to align the policy with demonstrations. However, it confronts issues such as low sample efficiency and poor generalization performance. Furthermore, ORL is a widely-known variation of RL that aims to obtain optimal policy given a limited dataset with possibly sub-optimal trajectories without additional interactions [11]. ORL methods possess strong stability and comprehensive performance, but with drawbacks such as demands on the scale of the dataset, and deadly triangle (bootstrap, off-policy and approximation) in estimating the Q-function [12]. To make matters worse, it does not address the inherent dependency of RL on designing reward functions.

G. Xie and D. Ma are with the Ocean College, Zhejiang University, Zhoushan, 316000, China. E-mail: {3200101418, mdf2004}@zju.edu.cn.

J. Xu is with the Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, 518000, China. E-mail: xjzh23@mails.tsinghua.edu.cn.

Z. Zhang is with the School of Engineering, WestLake University, Zhejiang, 310030, China, Email: stevezhangz@163.com.

X. Hou and Y. Ren are with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China. E-mail: {xiangwanghou@163.com; reny@tsinghua.edu.cn}

S. Zhang is with the Department of Data Science, New Jersey Institute of Technology, State of New Jersey, 07450, USA. E-mail: sz457@njit.edu.

D. Niyato is with the College of Computing and Data Science, Nanyang Technological University, Singapore. E-mail: dniyato@ntu.edu.sg.

* These authors contributed equally to this work.

In this paper, we construct a sample-efficient LfD training framework, FISHER, which is dedicated in the multi-AUV target tracking task and complex underwater environment. FISHER exploits the predominant advantage of RL in multi-objective optimization while integrating the strengths of IL and ORL to obtain stable and optimal policies. Our main contributions lie in the following:

- To the best of our knowledge, we first employ an expert data-driven approach to execute underwater multi-AUV target tracking tasks effectively. We propose an end-to-end, easy-to-deploy framework with a simulation-to-simulation (sim2sim)-based procedure to generate expert demonstrations easily. It consists of IL as the first stage, and ORL as the second stage. The former enables efficient policy improvement with few-shot demonstrations, while the latter further enhances the policies' generalization and multi-task capabilities.
- In the first stage of FISHER, a sample efficient IL algorithm, discriminator-actor-critic (DAC), is introduced, which leverages the replay buffer, off-policy RL algorithm, and improvements for training discriminator to tackle challenges faced by GAIL-based algorithms. Then, we derive the optimization objective of the multi-agent IL algorithm, based on Nash equilibrium and solving the dual optimization problem, thus expanding DAC to multi-agent DAC (MADAC).
- In the second stage of FISHER, a reward function-irrelevant ORL algorithm, multi-agent independent generalized decision transformer (MAIGDT) is introduced. By learning features of state transition from the future, with the help of the hindsight information matcher (HIM), MAIGDT can replicate the demonstrations without prior knowledge. Comparative experiments and performance evaluation of target tracking tasks show that MAIGDT significantly outperforms RL and ORL methods, finally validating the effectiveness of the FISHER framework.

II. Related Work

A. Multi-agent Target Tracking

Numerous methods have been proposed for target tracking tasks. Muslimov *et al.* designed a decentralized Lyapunov vector field for the target following [13]. Shen *et al.* deployed a nonconvex programming algorithm into a federated learning framework to optimize performance metrics under communication and latency constraints jointly [14]. In [15], a neural network-based predictor was introduced to improve the stability of APF and guarantee the Lyapunov stability of obstacle avoidance and connectivity-preserving in target tracking tasks. [16] and [1] constructed a grid diagram-based topologically organized biological neurodynamics model to characterize dynamic environments, which guides AUVs to avoid obstacles and search the target.

Unfortunately, most of these works only apply to ideal and specifically designed environments, thus lacking general applicability due to the highly dynamic underwater environment and complex demands of the tasks [17], [18].

B. MARL-assisted Target Tracking

Wei *et al.* brought adversarial behaviors between followers and the target into the differential game framework and used MATD3 to optimize the policies, where the system can asymptotically approach Nash Equilibrium [19]. Xia *et al.* took spatial information entropy into account and utilized the MASAC algorithm, notably increasing the tracking success rate [20]. Yue *et al.* factorized the centralized critic network of MASAC to reduce the variance in policy updates and learn efficient credit assignments [21]. In [22], coronal bidirectionally coordinated prediction networks were deployed to MADDPG, aiming to imitate human thinking.

Compared to traditional methods, RL is capable of handling complex demands. However, due to the randomness and instability inherent in RL, the reward function needs to be intricately designed. Modifications to environmental parameters, such as the number of AUVs, usually require a reward function redesign, severely hindering its application.

C. RL-assisted Task With Demonstrations

In previous works, expert demonstrations usually enhance the policy training process. In [23] and [24], classical controller-generated trajectories were mixed into the replay buffer to stabilize the early training stage. Stevšić *et al.* utilized the MPC controller as an expert to pre-train the policy [25].

The most similar work to our own is TSDRL-EE from Wang *et al.* [26], which adopted TD3 algorithm with behavior cloning (TD3+BC) as first-stage imitation pre-training and self-evolving TD3 that screens excellent experience from replay buffer as the second stage. However, offline RL algorithms like TD3+BC have intense demands on the dataset scale, otherwise poor outcomes may be produced [27]. Besides, the expert-assisted RL training paradigm does not resolve the dependency on the reward function.

Different from these works, our proposed framework is reward function irrelevant, based on few-shot demonstrations. Besides, our sim2sim procedure does not require that the environment of demonstration and policy interaction be the same, notably facilitating expert trajectory generation.

III. System Model

In this section, we describe the AUV dynamic model, underwater detection model, action consistency, and Markov decision process (MDP). We consider the system model of the multi-AUV underwater target tracking task as shown in Fig.1, $N(N > 1)$ AUVs are responsible for tracking a target and moving on the same plane at d meters below the surface. The positions of the target and AUVs are denoted as $\boldsymbol{p}_T = [x_T(t), y_T(t)]$ and $\boldsymbol{p}_i = [x_i(t), y_i(t)]$, where $i \in \{1, 2, \cdots, N\}$. There are M obstacles $\{o_1, ..., o_M\}$ in the target tracking area, and AUVs should cooperatively track the target while guaranteeing these obstacles are away from the safe radius of AUVs.

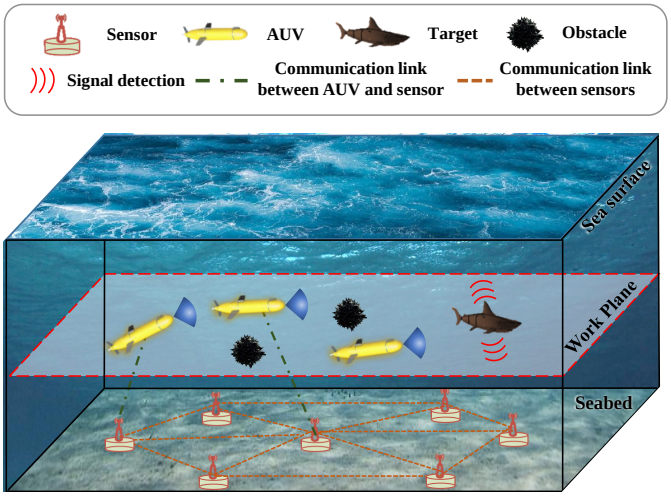


Fig. 1. Illustration of the multi-AUV underwater target tracking task.

A. AUV Dynamic Model

Given that each AUV tracks target in the horizontal plane, we can express the dynamic model using a three-degree-of freedom underdrive model, with a body-referenced frame $\boldsymbol{v}_i = [v_{i,x}(t), v_{i,y}(t), w_i]$ and a world-referenced frame $\boldsymbol{\eta}_i = [x_i(t), y_i(t), \theta_i]$, where $v_{i,x}(t), v_{i,y}(t), w_i$ and θ_i represent the surge velocity, sway velocity, angular velocity and yaw angle of the AUV i, respectively. Here we adopt the simplified Fossen's dynamic model [28], and the kinetic equation of AUV i can be expressed as

$$\boldsymbol{M}_i\dot{\boldsymbol{v}}_i + \boldsymbol{C}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{D}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{G}_i\boldsymbol{\eta}_i = \boldsymbol{\tau}_i, \tag{1}$$

where $\boldsymbol{M}_i$ represents the inertia matrix including the additional mass of AUV i, while $\boldsymbol{C}_i$ denotes the Coriros centripetal force matrix of AUV i, and $\boldsymbol{D}_i$ stands for the damping matrix caused by viscous hydrodynamic. Besides, $\boldsymbol{G}_i$ is the composite matrix of gravity and buoyancy, and $\boldsymbol{\tau}_i$ denotes the control input of AUV i. The relation between $\boldsymbol{v}_i$ and $\boldsymbol{\eta}_i$ can be expressed by the kinematic equation

$$\dot{\boldsymbol{\eta}}_i = \boldsymbol{J}(\boldsymbol{\eta}_i)\boldsymbol{v}_i, \tag{2}$$

where the transformation matrix $\boldsymbol{J}$ is given by

$$\boldsymbol{J}(\boldsymbol{\eta}_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

To be applied in simulation, the kinematic and kinetic equations above are discretized over time, and we can obtain

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

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

where $F(\boldsymbol{\eta}_t, \boldsymbol{v}_t) = \boldsymbol{\tau}_t - \boldsymbol{C}(\boldsymbol{v}_t)\boldsymbol{v}_t - \boldsymbol{D}(\boldsymbol{v}_t)\boldsymbol{v}_t - \boldsymbol{G}\boldsymbol{\eta}_t$, and ΔT is the time interval.

B. Underwater Detection Model

AUVs use sonar to detect the target and obstacles in the environment. The attenuation of underwater acoustic propagation can be specified by the active sonar equation

$$EM = SL - 2TL + TS - (NL - DI) - DT. \tag{6}$$

All parameters in Eq. (6) are in dB, where SL, TL, TS, NL, and DI represent the emission sound strength, transmission loss, target strength related to the target reflection area, environmental noise level and directionality index, respectively. DT and EM are the sonar's detection threshold and echo margin, respectively.

For AUV-to-AUV communication, it is only necessary for an AUV to receive the signal from another one, which can be modeled using the passive sonar equation

$$EM = SL - TL - NL + DI - DT. \tag{7}$$

The transmission loss TL is related to the AUV-target distance d and center acoustic frequency f, i.e.

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

$$\alpha(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\times10^{-4}f^2 + 0.003, \tag{9}$$

where $\alpha(f)$ is the empirical formula for the attenuation of sound waves in water. Since EM and d show a monotonically decreasing relationship, the maximum detection radius r_c of an AUV is

$$r_c = \underset{d}{\operatorname{argmax}}\{EM(d) \geq 0\}. \tag{10}$$

Given that the transmission loss of the passive sonar equation is only from the one-way propagation loss between AUVs rather than the two-way in the active equation, we consider that the communication range between AUVs significantly exceeds the tracking distance from the AUV to the target, namely, that AUVs' communication will be available.

C. Action Consistency

A high swarm consistency means that AUVs can track the target jointly. However, as the number of AUVs increases, it is hard to express the consistency directly from mutual distances between AUVs. Here, we use topology connectivity among AUVs to define the consistency. Similar to Eq. (7), we define the signal-to-noise ratio (SNR) between AUV i and AUV j in underwater communication

$$a_{ij} = SL - TL - NL + DI. \tag{11}$$

Then we formulate the AUV swarm as a graph and utilize the Laplace matrix $L \in \mathbb{R}^{N\times N}$ to describe the consistency of the AUV swarm. The element in the i-th row and j-th column is defined as follows:

$$l_{ij} = \begin{cases} -a_{ij}, & i \neq j, a_{ij} \geq DT, \\ \sum_{k=1,k\neq i}^{k=N} a_{ik}, & i = j, a_{ik} \geq DT, \\ 0, & i \neq j, a_{ij} < DT. \end{cases} \tag{12}$$

Thereby, the algebraic connectivity is the second smallest eigenvalue λ of matrix L. Larger λ predicates stronger consistency of the swarm.

D. Markov Decision Process

We model the interaction between AUVs and the target as an MDP, which assumes that the actions of AUVs only depend on the current state of the environment. This MDP can be expressed as a quintuple

$$\Omega = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}, \gamma), \tag{13}$$

where $\boldsymbol{S}$ represents the state space, and $\boldsymbol{A}$ denotes the action space of AUVs. Besides, $\boldsymbol{P}$ stands for the state transition probability function, and $\boldsymbol{R}$ represents the reward function, and γ is the discount factor. To be specific, the details of each element in the tuple can be listed as follows:

1) State space $\boldsymbol{S}$: The i-th AUV's state $\boldsymbol{s}_i(t) \in \mathbb{R}^{4N+2N_o}$ in the state space $\boldsymbol{S}_i$ is the concatenation of these parts

- **Target's position and velocity:** This part is signified by formula below:

$$\boldsymbol{s}_{i_1}(t) = \left\{x_{i,t}(t), y_{i,t}(t), v_{x_{i,t}}(t), v_{y_{i,t}}(t)\right\}, \tag{14}$$

  where $x_{i,t}(t)$ and $y_{i,t}(t)$ are the relative positions of the target to the AUV, while $v_{x_{i,t}}(t)$ and $v_{y_{i,t}}(t)$ are the relative velocities. These values are defined in the coordinate system of the polar axis in which the direction of the i-th AUV is facing, namely $x_{i,t}(t) = d_i(t)\cos(\theta_{i,t}(t))$, the same applies hereinafter.
- **Other AUVs' position and velocity:** This part is defined similarly. It is worth noting that this part includes the position and velocity information of all other AUVs

$$\begin{aligned}\boldsymbol{s}_{i_2}(t) = \big\{&x_{i,j}(t), y_{i,j}(t), v_{x_{i,j}}(t), v_{y_{i,j}}(t) \\ &\mid j \in \{1, ..., N\} \setminus \{i\}\big\}.\end{aligned} \tag{15}$$

- **Obstacles' position:** It is assumed that the AUV swarm can detect at most N_o obstacles, and this part is defined as

$$\begin{aligned}\boldsymbol{s}_{i_3}(t) = \big\{&EM_j\cos(\theta_{io_j}(t)), EM_j\sin(\theta_{io_j}(t)) \\ &\mid j \in \{1, ..., N_o\} \setminus \{i\}\big\},\end{aligned} \tag{16}$$

  where EM_j is the echo margin of obstacle o_j, while the angle between o_j's position relative to the AUV and AUV's orientation is defined as θ_{io_j}. When less than N_o obstacles are detected, the according EM is set to 0dB.

2) Action space $\boldsymbol{A}$: In MDP, each AUV makes action by its observing state. Here we define the action space $\boldsymbol{A}_i$ and corresponding action $\boldsymbol{a}_i(t)$ of the i-th AUV, according to the AUV motion model in Section III-A:

$$\boldsymbol{A}_i = [0, v_{\max}] \times [-\omega_{\max}, \omega_{\max}], \tag{17}$$

$$\boldsymbol{a}_i(t) = [v_i(t), w_i(t)], \tag{18}$$

where $||\boldsymbol{v}_i(t)|| = \sqrt{v_{i,x}(t)^2 + v_{i,y}(t)^2} \in [0, v_{\max}]$ and $||\boldsymbol{w}_i(t)|| \in [0, w_{\max}]$. Then, next state is obtained from interactions with environment, according to these actions and the state probability distributions: $\boldsymbol{s}_i(t) \times \boldsymbol{a}_1(t) \times \cdots \times \boldsymbol{a}_N(t) \longmapsto \boldsymbol{P}_{\boldsymbol{i}}(\boldsymbol{s}_i(t+1))$.

3) Reward function $\boldsymbol{R}$: As mentioned before, the reward function should be highly correlated with our requirements of the target tracking task. Still, it is not easy to achieve this, especially for complex scenarios. Therefore, we only utilize the reward function as an indicator to measure performance in simple scenarios, utilize a typical RL algorithm for training, and compare it with FISHER in the later experimental section. For the FISHER framework, the rewards of the MDP are derived from latent variables, which improves the policy to approximate the expert demonstrations. The details will be discussed in Section IV.

Here, the reward function $r_i(t) \in \boldsymbol{R}$ of i-th AUV consists of optimization objectives that correspond to our demands, which are listed as follows:

$$r_{ti_i}(t) = \begin{cases} d_i(t) - d^t_{\min}(t), & d_i(t) > d^t_{\min}, \\ 0, & d_i(t) < d^t_{\min}, \end{cases} \tag{19}$$

$$\begin{aligned}r_{o_i}(t) = &\sum_{j=1, j\neq i}^{N} (d_{\text{safe}} - d_{ij}(t)) + \sum_{k=1, k\neq i}^{M} (d_{\text{safe}} - d_{i,o_k}(t)), \\ &\text{for all } d_{ij}(t) < d_{\text{safe}},\ d_{i,o_k}(t) < d_{\text{safe}},\end{aligned} \tag{20}$$

$$r_{l_i}(t) = \begin{cases} \lambda_0 - \lambda_{\max}, & \lambda(t) \geq \lambda_{\max}, \\ \lambda_0 - \lambda(t), & \lambda(t) < \lambda_{\max}. \end{cases} \tag{21}$$

The meaning of each term in Eq. (19) $\sim$ (21) are elaborated as follows:

*1) **Target tracking reward:*** r_{ti_i} is determined by the distance between i-th AUV and the target, aiming at encouraging a single AUV to tracks the target independently, where $d^t_{\min}$ is the optimal distance from the target. We also introduce a term $r_{tc}(t) = \max_i \{r_{ti_i}(t)\}$ to represent overall tracking performance.

*2) **Collision avoidance penalty:*** r_{o_i} is used to avoid collision with all other AUVs and obstacles. This penalty is from all AUVs and obstacles that are closer than the safe distance d_{safe} from the current AUV, and all the penalties will be summed up.

*3) **Swarm consistency reward:*** This reward is directly from the algebraic connection λ. As the value of λ is usually much larger than 0, we offset λ with a constant λ_0. Excessive λ is truncated to avoid collisions.

To compare with the proposed FISHER framework, while accent the limitations of designing the reward function, we set two weight factors, a and b for r_{ti_i} and r_{tc}, to adjust the positivity of AUVs tracking target. Here, we propose three settings: **Cooperative:** $a = 1, b = 0$; **Mixed**: $a = 0.5, b = 0.5$; **Split**: $a = 0, b = 1$. The cooperative setting only requires that at least one AUV approach the target, while the split setting encourages each AUV to maintain proximity unilaterally for the consideration of robustness. Simulation results will be detailed in Section V.

Then, the reward function r_i of i-th AUV can be given by

$$r_i(t) = w_1(ar_{tc}(t) + br_{ti_i}(t)) + w_2 r_{o_i}(t) + w_3 r_{l_i}(t), \tag{22}$$

where $W = [aw_1, bw_1, w_2, w_3]$ is the weight vector.

IV. Methodology

In this section, we detail the expert demonstration-based training framework FISHER for target tracking. First, we introduce our sim2sim method, which aims to simplify the generation of expert trajectories. Then, we demonstrate the

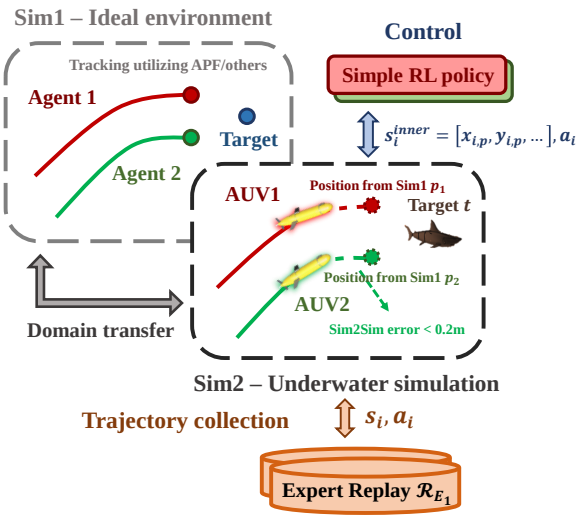


Fig. 2. A simple illustration of sim2sim procedure.

two stages of FISHER in order, followed by the detailed description of the overall architecture of the proposed FISHER framework.

A. Sim2sim Expert Demonstration Generation

It is intractable to directly generate demonstrations in underwater (simulation) environment due to the significant complexity, while RL methods suffer from the drawbacks of designing reward functions. Therefore, we propose a sim2sim procedure to simplify this process. The overall procedure is illustrated in Fig.2.

To be specifical, our sim2sim method consists of these parts:

1) ***Demonstration in simplified environment:*** We first simplify the environment, ignoring underwater and other environmental effects, and regarding AUVs and the target as particles. Then, we can utilize traditional target tracking methods, such as APF [29], to directly control AUVs' position and velocity without considering the structure of actuators.

2) ***Sim2sim:*** To do this, we train a one-by-one tracking RL policy, whose sole function is to allow AUV to reach a specific point with a specific velocity in the simulated environment with disturbance. The state space consists of the AUV's position and orientation, as well as the target's position and velocity, while its action space is identified as described in Section III. The reward function consists of the negative Euclidean distance to the target point and the negative MSE error of target velocity. Due to the simplicity of the training objective, the tracking error can quickly converge to < 0.2m.

3) ***Expert buffer collection:*** The RL policy derived from 2) is deployed to each AUV to execute the target tracking task in simulation under the guidance of demonstrations generated in 1). Disturbance parameters may be applied to enhance the complexity of the environment and the diversity of the demonstrations.

B. Improved Imitation Learning

GAIL is an important IL method that guides the policy in approximating the demonstrations by training a discriminator to distinguish between expert and policy-generated trajectories. To achieve this, GAIL employs the maximum entropy inverse RL (IRL) framework to obtain the reward function, and then the expert policy can be derived via RL procedure[1]

$$\mathrm{RL}(r) = \max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi}[r(s,a)], \tag{23}$$

$$\begin{aligned} \mathrm{IRL}_{\psi}(\pi_E) = &\operatorname*{argmax}_{r \in \mathbb{R}^{\boldsymbol{S}\times\boldsymbol{A}}} -\psi(r) + \mathbb{E}_{\pi_E}[r(s,a)] \\ &- \left(\max_{\pi \in \Pi} H(\pi) + \mathbb{E}_{\pi}[r(s,a)] \right), \end{aligned} \tag{24}$$

where $H(\pi) = \mathbb{E}_{s_t, a_t \sim \pi}\left[-\sum_{t=0}^{\infty} \gamma^t \log \pi(a_t|s_t)\right]$ denotes the γ-discounted casual entropy, π_E is the expert policy, and ψ is a reward function regulizer [8]. According to Ho. *et al.* [10], we can obtain the dual optimum

$$\mathrm{RL} \circ \mathrm{IRL}_{\psi}(\pi_E) = \operatorname*{argmin}_{\pi \in \Pi} -H(\pi) + \psi^{\star}(\rho_{\pi} - \rho_{\pi_E}), \tag{25}$$

where $\rho_{\pi}(s,a) = \pi(a|s)\sum_{t=0}^{\infty} \gamma^t \boldsymbol{P}(s_t = s|\pi)$ stands for policy's occupancy measure. GAIL utilizes a well-designed regulizer ψ_{GA}, and final objective can be expressed as

$$\begin{aligned} \psi^*_{GA}(\rho_{\pi} - \rho_{\pi_E}) = &\max_{D} \mathbb{E}_{\pi_E}[\log(D(s,a))] \\ &+ \mathbb{E}_{\pi}[\log(1 - D(s,a))], \end{aligned} \tag{26}$$

where ψ^*_{GA} is the convex conjugate of ψ, and $D : \boldsymbol{S} \times \boldsymbol{A} \to (0,1)$ is the discriminator. Finally, the policy can be improved via on-policy RL algorithms such as TRPO [30] and PPO [31]. However, it is intractable that GAIL demands extensive interactions with the environment. To address this, Kostrikov *et al.* [32] introduces the replay buffer to store previously generated trajectories. Then, the training objective of the discriminator can be expressed as

$$\mathcal{L}_D = \mathbb{E}_{\mathcal{R}}\left[\log\left(D(s,a)\right)\right] + \mathbb{E}_{\pi_E}\left[\log\left(1 - D(s,a)\right)\right], \tag{27}$$

where $\mathcal{R}$ denotes the replay buffer of the i-th AUV. Then, we can employ off-policy actor-critic algorithms for policy training, such as SAC [33] and TD3 [34], and this training paradigm is named as discriminator actor-critic (DAC).

Besides, We utilize some common improvements for discriminator training, including gradient penalty (GP) [35] and spectral normalization(SN) [36], which can notably enhance training stability and efficiency. Also, absorbing state s_a [37] is introduced to avoid a policy reaching the episode termination positively. Specifically, the absorbing state is entered when an episode is abnormally terminated, and every action transits the state to itself, namely: $r(s_a, \cdot) = 0$, and $\boldsymbol{P}(s_{t+1} = s_a | s_t = s_a, \cdot) = 1$. For consistency and unbiased reward function for absorbing state, internal reward function for training RL algorithm is signified by

$$r(s,a) \leftarrow \log D(s,a) - \log(1 - D(s,a)). \tag{28}$$

C. Multi-Agent Discriminator Actor-Critic

We turn our attention to extending DAC to multi-AUV scenarios. The optimum policies for MARL can be derived

[1] [10] uses the cost function $c : \boldsymbol{S} \times \boldsymbol{A} \to \mathbb{R}$ to notate these optimization objectives. The definition of the cost function is the opposite of the reward function.

Algorithm 1: MADAC training & Offline dataset generation (the first stage of the FISHER framework)

Initialize: Replay buffer $\boldsymbol{\mathcal{R}} = [\mathcal{R}_1, \ldots, \mathcal{R}_N]$, expert trajectory buffer $\boldsymbol{\mathcal{R}}_E = [\mathcal{R}_{E_1}, \ldots, \mathcal{R}_{E_N}]$, discriminator network D, policy network π_{θ_i} with corresponding critic network, offline dataset $\mathcal{R}_{o_i}$ of AUV i.
for *each episode* k **do**
  Reset the training environment.
  for *each environment timestep* t **do**
    Sample action $a_{t_i} \sim \pi_{\theta_i}(\cdot \mid s_{t_i})$.
    Collect the next state $s_{t+1_i} \sim \boldsymbol{P}_i(\cdot|\boldsymbol{s}, \boldsymbol{\pi})$.
    Store transition $\mathcal{R}_i \leftarrow \mathcal{R}_i \cup \{(s_{t_i}, a_{t_i}, \cdot, s_{t+1_i})\}$.
  end
  for *each IL gradient step* **do**
    Sample transitions from replay $\{(\boldsymbol{s}_t, \boldsymbol{a}_t, \cdot, \cdot)\}_{t=1}^{B} \sim \boldsymbol{\mathcal{R}}$, $\{(\boldsymbol{s}'_t, \boldsymbol{a}'_t, \cdot, \cdot)\}_{t=1}^{B} \sim \boldsymbol{\mathcal{R}}_E$.
    Calculate loss $\mathcal{L}_D = \sum_{b=1}^{B} \log D(\boldsymbol{s}_b, \boldsymbol{a}_b) - \log(1 - D(\boldsymbol{s}'_b, \boldsymbol{a}'_b))$.
    Update D with Adam+GP+SN.
  end
  for *each RL gradient step* **do**
    Sample $\{(s_{t_i}, a_{t_i}, \cdot, s_{t+1_i})\}_{t=1}^{B} \sim \mathcal{R}_i$.
    for $b = 1, \ldots, B$ **do**
      $r_i \leftarrow \log D(s_{b_i}, a_{b_i}) - \log(1 - D(s_{b_i}, a_{b_i}))$.
      $(s_{b_i}, a_{b_i}, \cdot, s_{b+1_i}) \leftarrow (s_{b_i}, a_{b_i}, r_i, s_{b+1_i})$.
    end
    Update π_{θ_i} with SAC [33].
  end
end
Collect trajectories τ_i using optimal policy $\pi^*_{\theta_i}$.
Store transition $\mathcal{R}_{o_i} \leftarrow \mathcal{R}_{o_i} \cup \tau_i$.

from a Nash equilibrium, namely, an agent cannot improve its own policy to achieve higher rewards if other agents keep their policies fixed. It can be expressed as a constrained optimization problem [38]

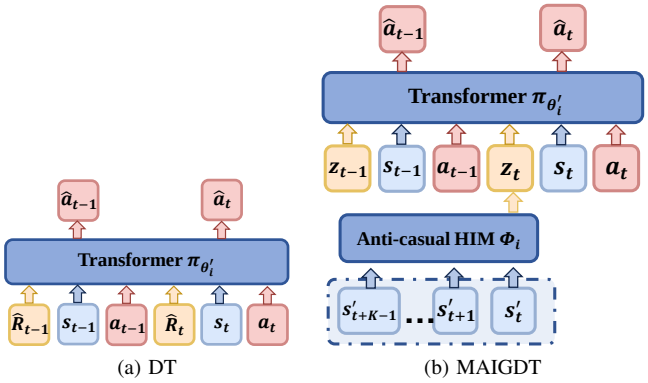
$$\begin{aligned} \text{MARL}(\boldsymbol{R}) &= \underset{\boldsymbol{\pi} \in \Pi, \boldsymbol{v}}{\operatorname{argmin}} f_{\boldsymbol{r}}(\boldsymbol{\pi}, \boldsymbol{v}) - H(\boldsymbol{R}), \\ s.t.\ v_i(\boldsymbol{s}) &\geq q_i(\boldsymbol{s}, a_i),\ \forall i \in \{1, \cdots, N\}, \end{aligned} \tag{29}$$

where $f_{\boldsymbol{r}}(\boldsymbol{\pi}, \boldsymbol{v}) = \sum_{i=1}^{N} \left(\sum_{\boldsymbol{s} \in \boldsymbol{S}} v_i(\boldsymbol{s}) - \mathbb{E}_{a_i \sim \pi_i(\cdot|\boldsymbol{s})} q_i(\boldsymbol{s}, a_i) \right)$, $\boldsymbol{s} = [s_1, \cdots, s_N]$ is the global state, $\boldsymbol{v} \triangleq [v_1, ..., v_N]$ denotes the value functions of policies, while $\boldsymbol{q}$ is the corresponding Q-function. If the Nash equilibrium is satisfied, the objective has a minimal value of zero, which is the only solution to the Nash equilibrium [39].

Accoring to Song *et al.* [40], we can finally obtain the objective of training discriminator(s), similar to Eq. (26)

$$\max_{\boldsymbol{D}} \mathbb{E}_{\boldsymbol{\mathcal{R}}}[\sum_{i=1}^{N} \log D_i(\boldsymbol{s}, \boldsymbol{a})] + \mathbb{E}_{\pi_E}[\sum_{i=1}^{N} \log(1 - D_i(\boldsymbol{s}, \boldsymbol{a}))], \tag{30}$$

where the proof of Eq. (30) is defered to the Appendix.



(a) DT (b) MAIGDT

Fig. 3. The architecutres of DT [41] and MAIGDT.

Next, we discuss the discriminator's training paradigms. Similar to MARL, we can train the discriminator in a centralized or decentralized setting. Specifically, the centralized setting utilizes a single discriminator that takes trajectories concatenated with all AUVs and assigns the same score to each AUV, namely $D = D_1 = \cdots = D_N$. In contrast, the decentralized setting equips a discriminator for each AUV, i.e. $D_i(\boldsymbol{s}, \boldsymbol{a}) = D_i(s_i, a_i)$. As the training stability is crucial to RL training procedure, we employ the centralized setting, which will exhibit more significant benefits as the number of AUVs N grows. The performance difference of the two settings is given in the subsequent section.

D. Multi-Agent Independent Generalized Decision Transformer

Offline RL helps policy improvement without interacting with the environment, among these algorithms, decision transformer (DT) [41] is an important application of generative models in ORL. It abstracts ORL problems into the seq2seq problems, thus eliminating the wrong estimation of Q-function in TD-based ORL algorithms.

To be specific, DT utilizes autoregressive predict-based language models, like GPT-2, to predict action. The GPT-like models utilize a stack of multiple decoders, namely self-attention layers with layer norm (LN) and residual connections. The self-attention layer takes n input tokens as embeddings $\{x_i\}_{i=1}^{n}$, and outputs embeddings with same dimension $\{z_i\}_{i=1}^{n}$. Specifically, input tokens are mapped to the key (k_i), query (q_i) and value (v_i) via linear transformations. Output tokens are the weighted average of values, based on the dot product between query and key

$$z_i = \sum_{j=1}^{n} \text{softmax}(\{< q_i, k_{j'} >\}_{j'=1}^{n})_j \cdot v_j. \tag{31}$$

Then, DT takes a batch of segments of trajectories with K timesteps, and the modified trajectories as the token sequence. A typical snippet of trajectory from timestep t can be denoted as

$$\tau'_{t_i} = \left(\hat{r}_i^{(t)}, s_i^{(t)}, a_i^{(t)}, \ldots, \hat{r}_i^{(t+K-1)}, s_i^{(t+K-1)}, a_i^{(t+K-1)} \right). \tag{32}$$

Here, the original DT utilizes the expected return $\hat{r}_{t_i} = \sum_{t'=t}^{T} r_{t'_i}$ of the i-th AUV, as an indicator of features of the

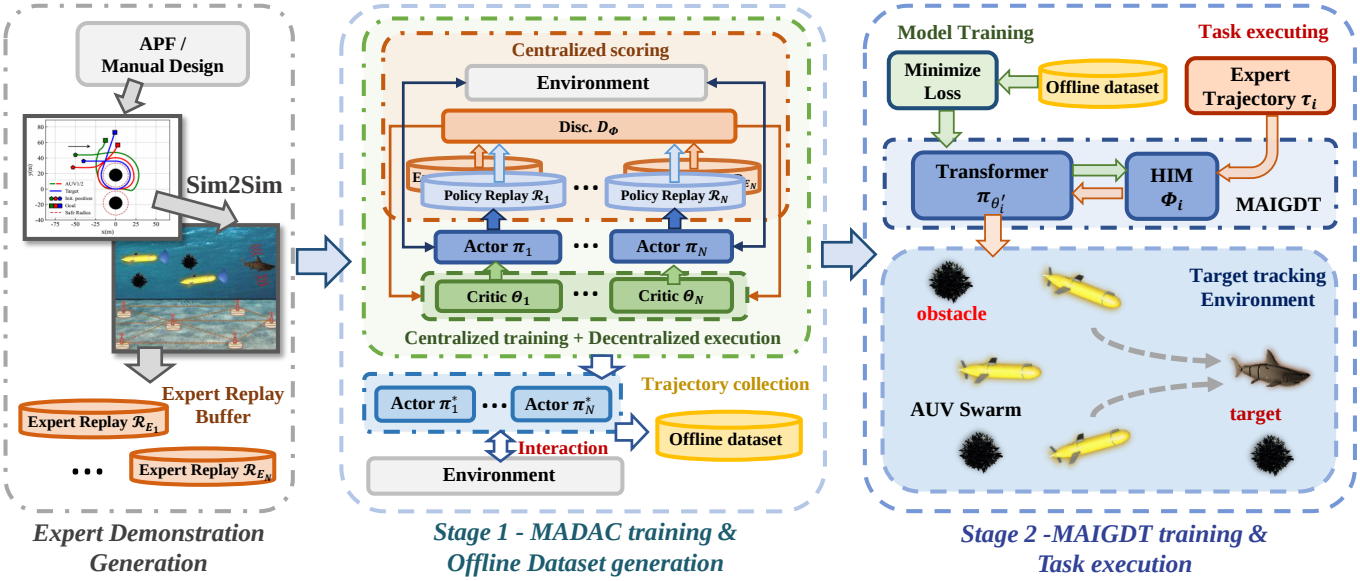


Fig. 4. The overall architecture of the FISHER framework in the multi-AUV underwater target tracking task.

trajectory. Then, the DT model predicts the next token, based on the input token sequence. Therefore, the prediction head corresponding to the input token $s_i(t)$ is trained to predict $\hat{a}_i(t)$. The training loss of the DT model for each timestep is averaged, namely

$$\max_{\pi_{\theta_i'}} J'(\theta_i') = \min_{\pi_{\theta_i'}} \mathcal{L}_{\text{MSE}}(\theta_i') = \min_{\pi_{\theta_i'}} [-\frac{1}{B}\sum_{j=1}^{B}(a_j - \hat{a}_j)^2]. \quad (33)$$

With the utilization of transformer and autoregressive training, DT is able to match trajectories with high returns based on credits assigned by self-attention layers. However, this training paradigm still relies on the reward function. Besides, as expected return is the only indicator of expected trajectories, multitasking performance can be poor or not feasible.

Furuta *et al.* [42] have proposed the generalized decision transformer (GDT) and demonstrated that we can use other information, rather than expected return, to find positive examples with certain contextual parameter values as a hindsight information matcher (HIM). Thus, we can improve DT to match the state transition of selected demonstrations to predict actions. In particular, we can use a second transformer Φ, which utilizes the anticasual design, namely takes a reverse-order state sequence as the input. The output of transformer Φ is the vector z that contains the information of future state transitions. Given that Φ is differentiable to DT's action-prediction loss, Φ can learn sufficient features of states by optimizing the Eq. (33), and DT is proficient in matching any distribution to an arbitrary precision. Then, when executing target tracking tasks, we can specify an expert trajectory τ_E' and use Φ to get features, which guides DT to replicate the one-shot demonstration efficiently. To be intuitive, the architectures of DT and MAIGDT are illustrated in Fig. 3.

Based on GDT and different from MADAC, we extend GDT to MAIGDT by implementing a decentralized training setting here, thanks to the powerful generalization capabilities of transformer-based models, and minor perturbations do not significantly affect the performance. We further demonstrate the stability of MAIGDT in Section V. Also, the decentralized setting contributes more to the deployment and scalability to policies.

Algorithm 2: MAIGDT training & Task executing (the second stage of the FISHER framework)

Initialize: Offline dataset $\mathcal{R}_{o_i}$, DT model parameters θ_i' with ialgorithmts anti-casual transformer Φ_i of AUV i.
Sample n batches of sequence with length K from the offline dataset τ_i.
for *each GDT gradient step* **do**
  Flip the state of sequences and get z_i vectors from anti-casual transformer Φ_i.
  Update models of GDT by Adam updating on Φ_i and θ_i' by $L_{\text{MSE}}(\theta_i')$ of Eq. (33).
end
Get expert demonstration τ_{E_i}' for imitation.
while *target tracking task timestep* t **do**
  Get flipped state sequence from timestep $t+K-1$ to t of τ_{E_i}', and get z_{t_i} vector from anti-casual transformer Φ_i.
  Predict action based on vector z_i, state s_i and a_i of previous K timesteps.
end

E. The Overall Architecture of The FISHER Framework

As traditional control methods and reward function-based RL methods may not viable in multi-objective tasks, we propose the efficient training framework FISHER, which utilizes expert demonstrations. The overall architecture of FISHER is depicted in Fig. 4, and the pseudo-code refers to Algorithm 1 and Algorithm 2. Above all, the sim2sim

TABLE I
PARAMETERS OF THE ENVIRONMENT AND ALGORITHM.

	Parameters	Values
Environment Parameters	Hydroacoustic parameters	100dB,3dB,3dB,
	SL, TS, DI, DT, NL	20dB,30dB
	Transmit frequency f	1.0rad/s
	Maximum speed $v_{\max}$	2.4m/s
	Maximum angular speed $w_{\max}$	1.0rad/s
Algorithm Parameters	Reward weight factor w_1, w_2, w_3	-0.25,-0.4, $-0.2/N$
	Distance parameters $d^t_{\min}, d_{\text{safe}}$	12m,8m
	Consistency parameters $\lambda_{\max}, \lambda_0$	$52N$,$50N$
	Hidden layer size	256
	batch size	256
	Discount factor γ	0.99
	Learning rate of SAC/MADAC	3×10^{-4}
	Learning rate of MAIGDT	1×10^{-4}
	MADAC gradient penalty factor	1.0
	MAIGDT context length K	20

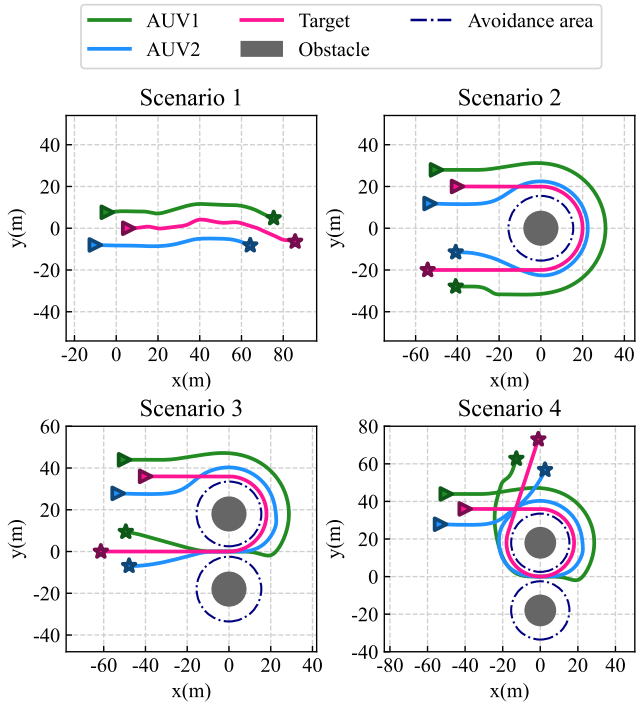
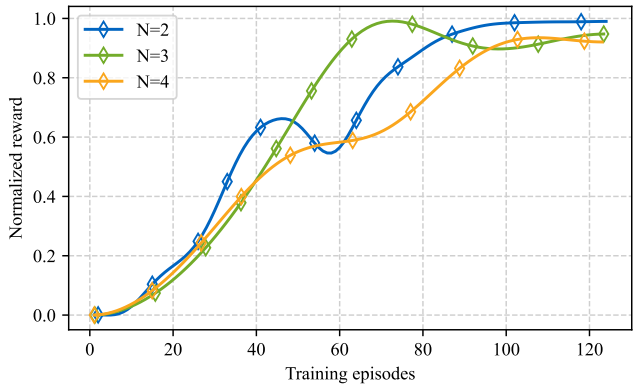
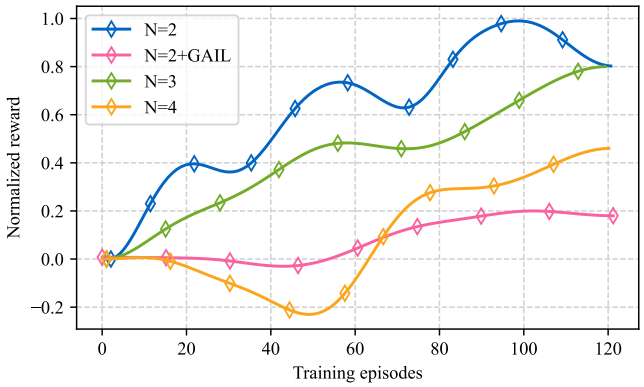


Fig. 5. Trajectories of the target and AUVs of the expert demonstrations, and obstacle distributions in different featured scenarios.



(a) MADAC ($N = 2, 3, 4$)



(b) MAIDAC ($N = 2, 3, 4$) and GAIL+PPO ($N = 2$)

Fig. 6. The training curves of (a) MADAC ($N = 2, 3, 4$). (b) MAIDAC ($N = 2, 3, 4$) and GAIL+PPO ($N = 2$).

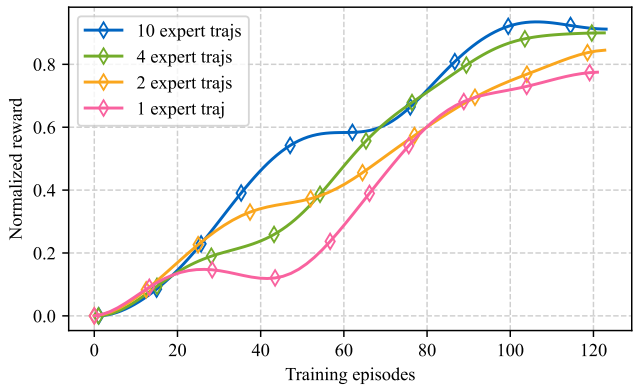


Fig. 7. The training curves of MADAC given 10(default), 4, 2, 1 expert trajectories.

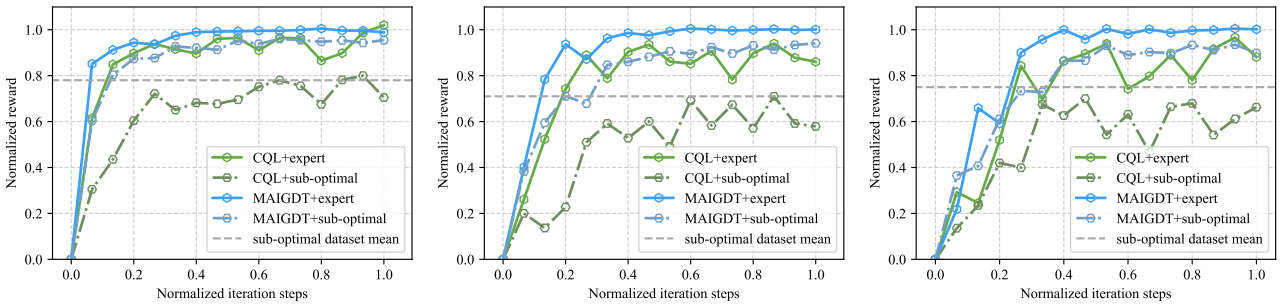
procedure is first executed to accomplish the domain transfer of demonstrations and collect the expert replay buffer. Then in the first stage, MADAC is utilized for independent training and trajectory collecting in its corresponding demonstration scenario, which can be executed in parallel, and all trajectories are consolidated to generate a large-scale offline dataset that guarantees the capability of ORL. Subsequently, in the second stage, MAIGDT leverages the HIM transformer Φ to learn to approximate trajectories in the offline dataset sufficiently. Finally, policies applicable across various scenarios with one-shot demonstration can be acquired.

V. SIMULATION RESULTS AND DISCUSSIONS

In this section, we demonstrate the performance of the FISHER framework through simulation experiments. First, we introduce the settings of the simulation experiments. Then, we detail the experiment scenario and demonstration design. Subsequently, we present the design of performance metrics, simulation results, and detailed discussions.

A. Experiment Settings

We verify the effectiveness of FISHER through comprehensive experiments in the simulation environment. Initially,



(a) $N = 2$ (b) $N = 3$ (c) $N = 4$

Fig. 8. The training curves of CQL and MAIGDT utilizing expert and sub-optimal dataset with (a) $N = 2$ AUVs. (b) $N = 3$ AUVs. (c) $N = 4$ AUVs.

AUVs and the target orientate towards the positive x-axis. Since the state space designed before has good rotation invariance, the orientation does not affect the experimental results. Besides, it is guaranteed that the target is in the detection range of each AUV in the initial state. The AUVs actuate at a frequency of 12.5Hz during experiments to track the target with a speed of 1.2m/s.

For algorithm parameters, since the MADAC utilize SAC [33] as its policy, the related parameters are also mainly referenced from SAC. Similarly, the parameters setting of MAIGDT mainly refer to DT [41]. Additionally, for the baseline algorithms for comparison, the parameters are set according to the original paper. The other parameters of the environment and algorithm are mainly listed in Table I for summary.

B. Experiment Scenarios and Demonstrations

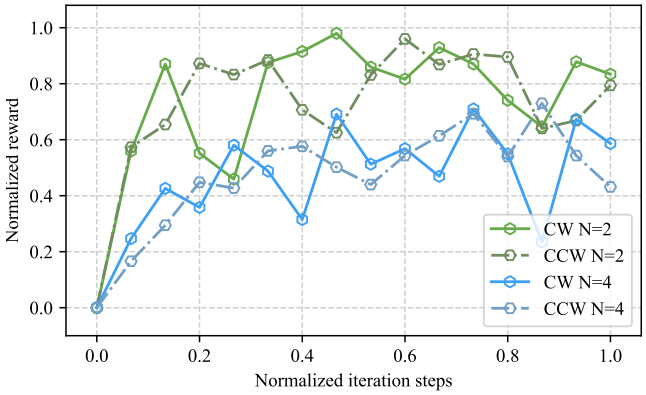
Several scenarios are designed to evaluate the performance of proposed MADAC and MAIGDT algorithms employed in the overall FISHER framework. These scenarios feature different target moving trajectories and obstacle distributions, and we design the corresponding demonstrations for them. We divide these scenarios into two parts as shown in Fig. 5:

The first part features sparse obstacle(s). As all the objectives are relatively uncomplicated to be optimized, RL methods usually demonstrate passable performance in this part. Here, we design two scenarios, and we label them as scenario 1 and scenario 2.

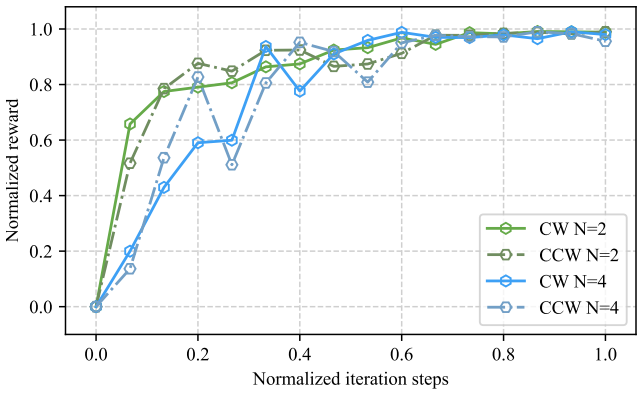
In contrast, the second part, which features dense obstacles, presents a challenge to optimization given the reward function, as AUVs must weigh the objectives dynamically. For example, AUVs must reorganize their formation while passing through obstacles. Accordingly, we also design two scenarios, which are labeled as scenario 3 and scenario 4. The position of the target and obstacle(s) and expert trajectories of these mentioned scenarios are shown in Fig. 5 in detail.

C. Experiment Results and Analysis

Various experiments are conducted based on these scenarios mentioned before. Firstly, we evaluate the performance of MADAC and MAIGDT through comparative experiments in scenarios with sparse obstacle(s) quantitatively using the reward function, which is relatively capable of aligning with our demands in simple situations.



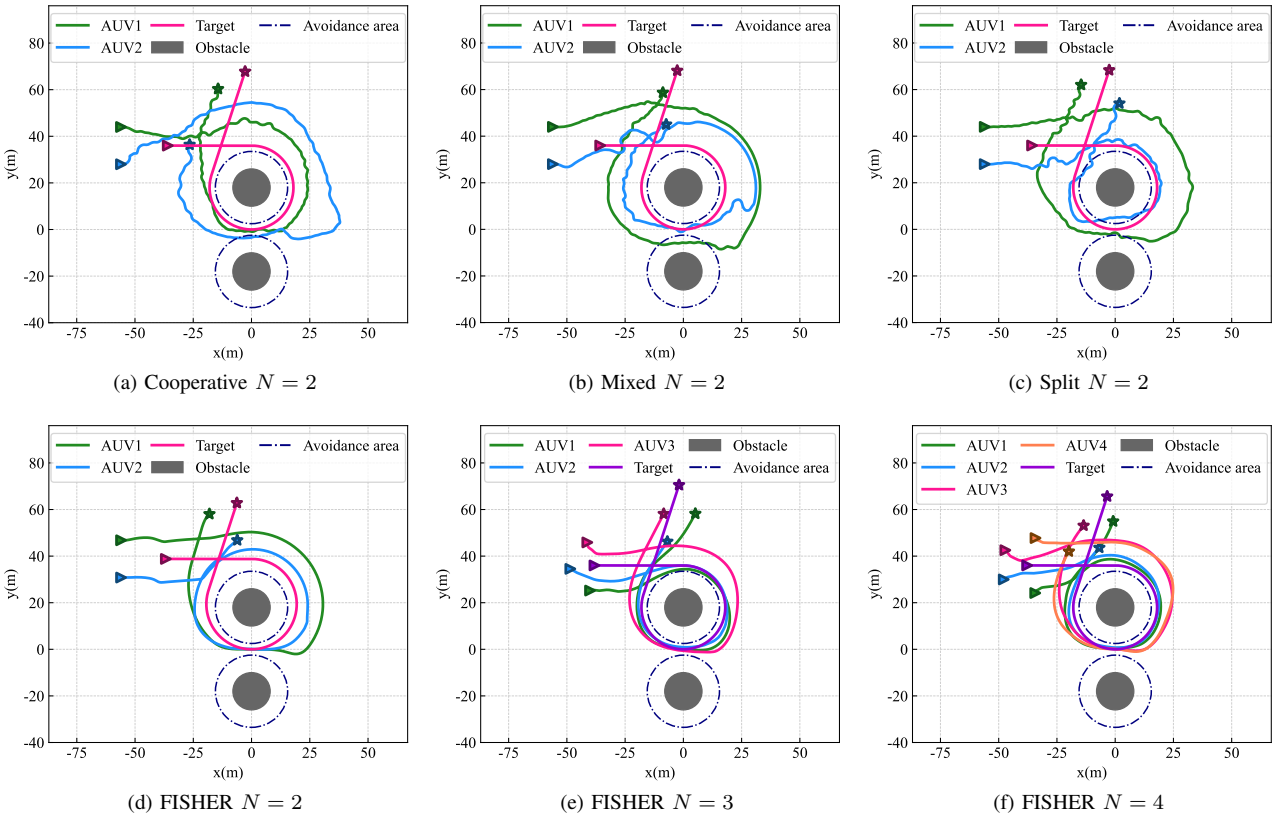
(a) CQL ($N = 2, 4$)



(b) MAIGDT ($N = 2, 4$)

Fig. 9. The training curves of CW and CCW tasks taken from scenario 2. (a) Utilizing CQL ($N = 2, 4$) for training. (b) Utilizing MAIGDT ($N = 2, 4$) for training.

Fig. 6 displays the training results of MADAC, multi-agent DAC with a decentralized setting (MAIDAC), and a mainstream implementation of GAIL (GAIL+PPO), in scenario 1 with the number of AUVs ranging from $N = 2$ to $N = 4$. It should be noted that the reward between different N cannot be compared directly. Consequently, we normalize the reward, such that the average reward obtained by randomly initialized policies is recorded as 0, while the reward from the expert trajectory is set to 1. Observations from Fig. 6(b) demonstrate



(a) Cooperative $N = 2$ (b) Mixed $N = 2$ (c) Split $N = 2$

(d) FISHER $N = 2$ (e) FISHER $N = 3$ (f) FISHER $N = 4$

Fig. 10. Representative tracking trajectories of AUVs via FISHER and SAC+CTDE under different reward settings and AUV number, respectively. (a) SAC+CTDE+Cooperative. (b) SAC+CTDE+Mixed. (c) SAC+CTDE+Split. (d) FISHER ($N = 2$). (e) FISHER ($N = 3$). (f) FISHER ($N = 4$).

that the GAIL method, due to its low sample efficiency and unsatisfactory training stability, shows virtually no policy improvement even in the simplest case of $N = 2$. Although MAIDAC, which adopts a decentralized setting, shows little difference from MADAC at $N = 2$, training becomes unstable from $N = 3$ onwards, encountering a significant bottleneck, which indicates that AUVs can hardly explore advantageous states. In contrast, the training curve of MADAC for each number N is stable, and finally, MADAC achieves a reward close to that of experts.

Moveover, Fig. 7 illustrates the training curves of MADAC given a different number of demonstrations when $N = 4$. As illustrated in the results, with the increase of the demonstrations (expert trajectories), the normalized reward shows an upward trend at the same training epoch, which showcases the acceleration and improvement effects on the training process. Besides, despite the presence of a reduction of performance for fewer demonstrations, MADAC does not necessitate an excessive number of expert trajectories, thereby ensuring sufficient training stability.

On the other hand, to verify the superior performance of MAIGDT, we conduct the comparative experiments in scenario 1 utilizing MAIGDT and classical TD-based ORL algorithm consevative Q-learning (CQL) [43], employing the expert dataset from MADAC (expert) and the dataset with sub-optimal trajectories from SAC (sub-optimal), respectively. The training results of MAIGDT and CQL are depicted in Fig. 8. The mean and standard deviation of the sub-optimal dataset trajectory rewards are represented in the figure with dashed lines and semi-transparent fill. Although the training curves of CQL also exhibit an upward trend, capable of enabling policy improvement. Nevertheless, significant fluctuations persist in the training process even after convergence. This is unacceptable, due to the unpredictable performance in the policy deployment. Conversely, MAIGDT's regression-based training approach ensures training stability. Furthermore, in situations of dataset degradation and an increase in the AUV number N, CQL's performance drastically deteriorates, while GDT, by learning the state transition rather than explicit rewards, can still adeptly replicate the performance of expert demonstrations.

Furthermore, we conduct extensive experiments to assess the multi-task performance of MAIGDT. To accomplish this, we formulate two tasks, both originating from scenario 2, but with forward directions rotating clockwise (CW) and counterclockwise (CCW). The results of the experiment are illustrated in Fig. 9. For $N = 2$, the training outcomes of CQL are quite unstable, with the reward of the two tasks exhibiting significant fluctuation. For $N = 4$, CQL is incapable of reaching convergence, as the reward function fails to align with the optimization objective of the target tracking task, which is further demonstrated later in this section. Still, MAIGDT can achieve the best performance in both tasks.

Subsequently, we further evaluate the overall performance of the FISHER framework in scenarios with dense obstacles. As the environment becomes more complicated, the reward

TABLE II
PERFORMANCE METRICS OF AUVS TRACKING TARGET UTILIZING SAC+CTDE AND PROPOSED FISHER FRAMEWORK IN SCENARIO 4.

Experiments	Cooperative $N=2$	Mixed $N=2$	Split $N=2$	FISHER $N=2$	FISHER $N=3$	FISHER $N=4$
$\mathbb{E}$(min-distance) /m	16.38 ± 0.30	15.14 ± 0.89	14.16 ± 0.79	13.02 ± 0.89	12.50 ± 1.13	12.24 ± 1.37
Std(min-distance) /m	2.95 ± 0.55	2.61 ± 0.41	2.38 ± 0.59	2.02 ± 0.35	2.30 ± 0.56	2.57 ± 0.64
$\mathbb{E}$(consistency)	87.69 ± 4.44	96.11 ± 1.94	97.88 ± 0.91	97.53 ± 0.69	140.99 ± 3.55	191.12 ± 5.09
Std(consistency)	5.35 ± 1.42	4.88 ± 1.58	2.99 ± 0.64	1.42 ± 0.37	4.97 ± 2.60	8.20 ± 3.98
Min(obs-distance) /m	7.92 ± 0.76	6.16 ± 0.66	3.91 ± 0.79	10.28 ± 0.15	9.19 ± 0.87	9.01 ± 0.74
Danger time /s	8.44 ± 2.13	12.23 ± 3.17	16.64 ± 4.65	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

function is unsuitable for evaluating performance. For quantitative evaluation, we introduce some performance indicators similar to Yang *et al.* [3]: minimum distance mean, minimum distance standard deviation, consistency mean, consistency standard derivation, minimum obstacle distance and danger time. Here, the minimum distance represents the distance between the target and the nearest AUV to the target, the minimum obstacle distance refers to the shortest distance between the obstacle and the AUVs throughout the entire process, while the duration in danger is defined as the period which at least one AUV that is less than 8m away from an obstacle, and consistency is represented by the algebraic connectivity λ. Due to the randomness of RL training process, we train the policies until convergence from scratch 3 times to measure the training stability, and all results are presented as $a \pm b$, where b is the standard deviation of metrics between these policies.

To reveal the limitations of designing the reward function, we also train a classical RL algorithm for continuous action space - SAC [33] following a centralized training with distributed execution (CTDE) manner (SAC+CTDE), with the three settings of the reward function mentioned in Section III, namely cooperative, mixed and split, respectively. Performance metrics of SAC+CTDE with three reward settings in $N = 2$ and MAIGDT in $N = 2, 3, 4$ of scenario 4 are shown in Table II, and the corresponding trajectories are shown in Fig. 10. Additionally, the minimum distance mean and minimal obstacle distance of expert demonstrations are 12m and 10.8m, respectively, while the consistency are 100.1 for $N = 2$, 150.2 for $N = 3$, and 200.2 for $N = 4$.

For $N = 2$, SAC+CTDE performs passable capability only in certain performance metrics, such as better tracking performance in the split setting and better obstacle avoidance in the cooperative setting. However, all reward functions fail to achieve a balance under multiple objectives. The trajectories of AUVs are not smooth and exhibit significant jitters. Moreover, SAC+CTDE fails to track the target and AUVs turn back when encountering obstacles in $N = 3$ and $N = 4$ with any reward function of the three, due to the increasing deviation of the reward function from expected optimization goal. The representative example of failed tracking processes of SAC+CTDE is shown in Fig. 11. Conversely, FISHER successfully replicates demonstrations, showcasing superior performance comparable to those of experts, while ensuring stability as the number of AUVs increases.

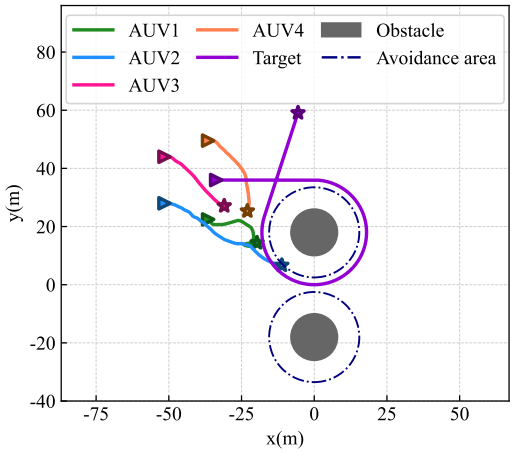


Fig. 11. An example of tracking failure utilizing SAC+CTDE for training with $N = 4$ AUVs.

VI. CONCLUSION

In this paper, we developed FISHER, an efficient training framework that leverages expert demonstrations generated from sim2sim for multi-AUV underwater target tracking task. We first introduced DAC to enhance the sample efficiency and training stability of the GAIL-based algorithm, and we expanded it to MADAC by optimizing the dual problem with the Nash equilibrium constraint. Then, MAIGDT was introduced to attain multi-task applicable policies with the help of latent variables of demonstrations from the anti-casual information extractor rather than designing reward functions like DT and other ORL methods. The MADAC and MAIGDT together constitute the two stages of FISHER framework. Simulation results in multiple scenarios reveal that FISHER excellently learns from demonstrations and achieves superior performance levels comparable to expert trajectories. Future work can focus on validating suitability in complex underwater conditions, such as vortex and dynamic environments, and combine the tasks executed in the real environment to settle challenges from the sim2real application.

APPENDIX
PROOF TO THE TRAINING OBJECIVE OF MADAC

Hereinafter, $\hat{v}_i(\boldsymbol{s})$, $\hat{q}_i(\boldsymbol{s}, a_i)$ represent $\hat{v}_i(\boldsymbol{s}; \boldsymbol{\pi}, \boldsymbol{r})$ and $\hat{q}_i(\boldsymbol{s}, a_i; \boldsymbol{\pi}, \boldsymbol{r})$, respectively.

Lemma 1: For the value function $\hat{v}_i(\boldsymbol{s})$ that meets the condition of the Bellman equation

$$\hat{v}_i(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{\pi}}[r_i(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in\boldsymbol{S}} \boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\hat{v}_i(\boldsymbol{s}')]. \tag{34}$$

Then $\hat{q}_i(\boldsymbol{s}, a_i) = \mathbb{E}_{\pi_{-i}}[r_i(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in\boldsymbol{S}} \boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\hat{v}_i(\boldsymbol{s}')]$ is defined similarly, where π_{-i} denotes all policies except the policy of i-th AUV. Then we can obtain

- 1° For any $\boldsymbol{\pi}$, $f_{\boldsymbol{r}}(\boldsymbol{\pi}, \boldsymbol{v}) = 0$.
- 2° $\boldsymbol{\pi}$ is the Nash equilibrium under $\boldsymbol{r}$ if and only if $\hat{v}_i(\boldsymbol{s}) \geq \hat{q}_i(\boldsymbol{s}, a_i), \forall i \in \{1, \cdots, N\}$.

Proof 1: By the definition of $\hat{v}_i(\boldsymbol{s})$, as actions are mutually independent conditioned on $\boldsymbol{s}$, we can obtain

$$\begin{aligned}\hat{v}_i(\boldsymbol{s}) &= \mathbb{E}_{\boldsymbol{\pi}}[r_i(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in\boldsymbol{S}} \boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\hat{v}_i(\boldsymbol{s}')] \\ &= \mathbb{E}_{\pi_i}[\mathbb{E}_{\pi_{-i}}[r_i(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in\boldsymbol{S}} \boldsymbol{P}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})\hat{v}_i(\boldsymbol{s}')]] \\ &= \mathbb{E}_{\pi_i}[\hat{q}_i(\boldsymbol{s}, a_i)].\end{aligned} \tag{35}$$

Therefore 1° can be proved. For 2°, clearly the Nash equilibrium conditions are violated if there exists $i \in \{1, \cdots, N\}$, $\boldsymbol{s}$ and a_i that $\hat{v}_i(\boldsymbol{s}) < \hat{q}_i(\boldsymbol{s}, a_i)$, namely the i-th AUV can choose a_i, when the corrsponding state is $\boldsymbol{s}$ and the policy follow π_i subsequently, to achieve higher expected return. If the constraint is met, then

$$\hat{v}_i(\boldsymbol{s}) \geq \mathbb{E}_{\pi_i}[\hat{q}_i(\boldsymbol{s}, a_i)] = \hat{v}_i(\boldsymbol{s}). \tag{36}$$

The Eq. (36) signifies that when the constraint is met, there must only one solution of $\hat{v}_i(\boldsymbol{s})$. Hence, 2° is proved. ■

Then we attempt to obtain the multiple-timestep TD equivalent of constraint:

Theorem 2: Assume that the AUVs' trajectory from timestep 0 to $t-1$ is denotes as $\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}$, then the state for timestep t is $\boldsymbol{s}^{(t)}$, and the action of i-th AUV is $a_i^{(t)}$, we denote the discounted expected return as

$$\begin{aligned}&\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)}) \\ =&\sum_{j=0}^{t-1} \gamma^j r_i(\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}) \\ +&\gamma^t \mathbb{E}[r_i(\boldsymbol{s}^{(t)}, \boldsymbol{a}^{(t)}) + \gamma \sum_{\boldsymbol{s}'\in\boldsymbol{S}} P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}^{(t)})\hat{v}_i(\boldsymbol{s}')].\end{aligned} \tag{37}$$

Then $\boldsymbol{\pi}$ reaches a Nash equilibrium if and only if

$$\begin{aligned}&\hat{v}_i(\boldsymbol{s}^{(0)}) \geq \mathbb{E}_{\pi_{-i}}\left[\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)})\right] \\ &\triangleq Q_i^{(t)}(\{\boldsymbol{s}^{(j)}, a_i^{(j)}\}_{j=0}^{t}), \forall t \in \mathbb{N}^+, i \in \{1, \cdots, N\}.\end{aligned} \tag{38}$$

Proof 2: Similarly, we consider that the constraint does not comply, namely exists $i \in \{1, \cdots, N\}$, $\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}$ that

$$\hat{v}_i(\boldsymbol{s}^{(0)}) < \mathbb{E}_{\pi_{-i}}[\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)})], \tag{39}$$

then i-th AUV can achieve a higher expected return by choosing $a_i^{(j)}$ when correspond state is $\boldsymbol{s}^{(j)}$ and following π_i subsequently. This contradicts the Nash equilibrium. If the constraint is met, then for all i and trajectory $\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}$,

$$\hat{v}_i(\boldsymbol{s}^{(0)}) \geq \mathbb{E}_{\pi_{-i}}[\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)})]. \tag{40}$$

As we can construct any $\hat{q}_i(\boldsymbol{s}^{(0)}, a_i^{(0)})$, which has the equivalent

$$\begin{aligned}&\hat{q}_i(\boldsymbol{s}^{(0)}, a_i^{(0)}) \\ &= \mathbb{E}_{\boldsymbol{\pi}}[\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)})] \\ &= \mathbb{E}_{\pi_i}[\mathbb{E}_{\pi_{-i}}[\hat{q}_i^{(t)}(\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}, \boldsymbol{s}^{(t)}, a_i^{(t)})]],\end{aligned} \tag{41}$$

which is direct, as taking expectation over π_i and π_{-i} simultaneously takes it over states(π_{-i}) and actions(π_i). As the $\boldsymbol{s}^{(0)}$ and $a_i^{(0)}$ can be arbitrary, we can extend it to

$$\hat{v}_i(\boldsymbol{s}) \geq \hat{q}_i(\boldsymbol{s}, a_i). \tag{42}$$

Then Theorem 1 can be proved according to Lemma 1. ■

According to Theorem 1, the optimizing objective of Nash equilibrium is always zero for the final solution. Therefore we can solve the dual problem of MARL and MAIRL by constructing the Lagrange multiplier

$$\begin{aligned}\max_{\lambda\geq 0}\min_{\boldsymbol{\pi}} &\sum_{i=1}^{N}\sum_{\tau_i\in\mathcal{T}_i^t} \lambda(\tau_i)\left(Q_i^{(t)}(\tau_i) - \hat{v}_i(\boldsymbol{s}^{(0)})\right) \\ &\triangleq L_{\mathbf{r}}^{(t+1)}(\pi, \lambda),\end{aligned} \tag{43}$$

where $\mathcal{T}_i^t$ is the set of all possible t-timestep length sequence $\{\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}\}_{j=0}^{t-1}$, $\boldsymbol{s}^{(0)} \sim \boldsymbol{P}_0(\boldsymbol{s})$ is the initial state. λ is the vector of $N \cdot |\mathcal{T}_i^t|$ Lagrange multipliers, where $|\mathcal{T}_i^t|$ is the number of sequences in $\mathcal{T}_i^t$.

Theorem 2: For any two sets of policies $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, we define that the probability of generating the sequence τ_i with policy π_i and π'_{-i}, namely

$$\begin{aligned}&\lambda'_{\boldsymbol{\pi}}(\tau_i) = \boldsymbol{P}_0(\boldsymbol{s})\pi_i(a_i^{(0)}|\boldsymbol{s}^{(0)}) \\ &\prod_{j=1}^{t} \pi_i(a_i^{(j)}|\boldsymbol{s}^{(j)}) \sum_{a_{-i}^{(j-1)}} \boldsymbol{P}(\boldsymbol{s}^{(j)}|\boldsymbol{s}^{(j-1)}, \boldsymbol{a}^{(j-1)})\pi'_{-i}(a_{-i}^{(j)}|\boldsymbol{s}^{(j)}).\end{aligned} \tag{44}$$

Then if the multipliers are the probability of Eq. (44) of corresponding sequences, the dual function can be expressed as

$$\begin{aligned}\lim_{t\to\infty} L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}', \lambda'_{\boldsymbol{\pi}}) = &\sum_{i=1}^{N} \mathbb{E}_{\pi_i, \pi'_{-i}}[r_i(\boldsymbol{s}, \boldsymbol{a})] \\ &- \sum_{i=1}^{N} \mathbb{E}_{\pi'_i \pi'_{-i}}[r_i(\boldsymbol{s}, \boldsymbol{a})].\end{aligned} \tag{45}$$

Proof 3: Here we denote that $Q'_i(\tau_i) = Q_i(\tau_i; \boldsymbol{\pi}', \boldsymbol{r})$, $\hat{q}'$ and $\hat{v}'$ are defined similarly. According to Eq. (43)

$$L_{\boldsymbol{r}}^{(t+1)}(\boldsymbol{\pi}', \lambda'_{\boldsymbol{\pi}}) = \sum_{i=1}^{N}\sum_{\tau_i\in\mathcal{T}_i} \lambda'(\tau_i)(Q'_i(\tau_i) - \hat{v}'_i(\boldsymbol{s}^{(0)})). \tag{46}$$

We expand the Eq. (46) by the definition of Q'_i and $\hat{v}'_i$, it can be noticed that

$$\begin{aligned}&\sum_{\tau_i\in\mathcal{T}_i} \lambda'(\tau_i) Q'_i(\tau_i) \\ &= \mathbb{E}_{\pi_i}[\mathbb{E}_{\pi'_{-i}}[\sum_{j=0}^{t-1} \gamma^j r_i(\boldsymbol{s}^{(j)}, \boldsymbol{a}^{(j)}) + \gamma^t \hat{q}'_i(\boldsymbol{s}^{(t)}, a_i^{(t)})]],\end{aligned} \tag{47}$$

which means that π_i is used for executing the first t timesteps, while π'_i is used subsequently, with other agents complying π'_{-i} all along. When $t \to \infty$, $\gamma^t \to 0$ and $\hat{q}'_i(\boldsymbol{s}^{(t)}, a_i^{(t)})$ is bounded, the Eq. (47) converges to $\mathbb{E}_{\pi_i,\pi'_{-i}}[r_i]$. Then, for the term $\hat{v}'_i(\boldsymbol{s}^{(0)})$, it can be observed that

$$\sum_{\tau_i \in \mathcal{T}_i} \lambda'(\tau_i)\hat{v}'_i(\boldsymbol{s}^{(0)}) = \mathbb{E}_{\boldsymbol{s}^{(0)}\sim \boldsymbol{P}^{(0)}(\boldsymbol{s})}[\hat{v}'_i(\boldsymbol{s}^{(0)})] = \mathbb{E}_{\boldsymbol{\pi}'}[r_i]. \tag{48}$$

Combining Eq. (47) and Eq. (48), the Theorem 2 can be proved. ■

Based on analysis before, we define the MAIRL procedure similar to Eq. (24)

$$\begin{aligned}\mathrm{MAIRL}_{\psi}(\boldsymbol{\pi}_E) = \operatorname*{argmin}_{\mathbf{r}} -\psi(\mathbf{r}) + \sum_{i=1}^{N}(\mathbb{E}_{\boldsymbol{\pi}_E}[r_i]) \\ - \max_{\boldsymbol{\pi}} \sum_{i=1}^{N}(\beta H_i(\pi_i) + \mathbb{E}_{\pi_i,\pi_{E_{-i}}}[r_i]),\end{aligned} \tag{49}$$

where $H_i(\pi_i)$ is the casual entropy of π_i and β is the hyper parameter represents the strength of regularization. If $N = 1$ and $\beta = 1$, the Eq. (49) is equivalent to the Eq. (24).

Finally, we can derive the solution of the dual problem:

Theorem 3: Assume that the reward regularizer is additively separable for each AUV, namely $\phi(\boldsymbol{r}) = \sum_{i=1}^{N}\phi_i(r_i)$, and for all feasible $r \in \mathrm{MAIRL}_{\phi}(\pi_E)$ there is a unique solution for $\mathrm{MARL}(r)$. Then, the dual optimum can be expressed as

$$\begin{aligned}&\mathrm{MARL} \circ \mathrm{MAIRL}_{\psi}(\boldsymbol{\pi}_E) \\ &= \operatorname*{argmin}_{\pi\in\Pi} \sum_{i=1}^{N} -\beta H_i(\pi_i) + \psi_i^{\star}(\rho_{\pi_i,\pi_{E_{-i}}} - \rho_{\boldsymbol{\pi}_E}).\end{aligned} \tag{50}$$

Proof 4: We attempt to use Eq. (25) to solve the dual problem by decomposing the problem to the single-agent scenario. The RL objective for i-th AUV, where other AUVs complies policy $\pi_{E_{-i}}$, can be expressed as

$$\mathrm{RL}_i(r_i) = \max_{\pi_i\in\Pi} H_i(\pi_i) + \mathbb{E}_{\pi_i,\pi_{E_{-i}}}[r_i]. \tag{51}$$

Then the IRL objective of the same condition can be expressed as

$$\begin{aligned}\mathrm{IRL}_{i,\psi}(\boldsymbol{\pi}_E) = \operatorname*{argmin}_{r_i\in\mathbb{R}^{\boldsymbol{S}_i\times\boldsymbol{A}_i}} -\psi_i(r_i) + \mathbb{E}_{\pi_{E_i}}[r_i] \\ - \left(\max_{\pi_i\in\Pi} H_i(\pi_i) + \mathbb{E}_{\pi_i,\pi_{E_{-i}}}[r_i]\right).\end{aligned} \tag{52}$$

As we have assumed that the reward regularizer is additively separable, the solution to MAIRL can be expressed as a group of solutions of IRL

$$\mathrm{MAIRL}_{\psi} = [\mathrm{IRL}_{1,\psi}, \ldots, \mathrm{IRL}_{N,\psi}]. \tag{53}$$

Similarly,

$$\mathrm{MARL}_{\boldsymbol{r}} = [\mathrm{RL}_1(r_1), \ldots, \mathrm{RL}_N(r_N)]. \tag{54}$$

Then, we can solve the dual problem for each AUV analogous to Eq. (25), and Theorem 3 can be proved. ■

Analogous to Eq. (26), we can design a regularizer like ψ_{GA} in single-agent scenario. As the optimum solution for $\psi^{\star}_{\mathrm{GA}}(\rho_{\pi_i,\pi_{E_{-i}}}, \rho_{\boldsymbol{\pi}_E})$ is same to $\psi^{\star}_{\mathrm{GA}}(\rho_{\boldsymbol{\pi}}, \rho_{\boldsymbol{\pi}_E})$, namely $\boldsymbol{\pi}_E$, we can substitute the former of in Eq. (50) with the latter, and eventually the Eq. (30) can be derived. Specifically, the decentralized setting of the discriminator utilizes the regularizer of $\psi_i(r_i) = \psi_{\mathrm{GA}}(r_i)$, and the centralized setting utilizes the regularizer as follows:

$$\psi(\boldsymbol{r}) = \begin{cases}\psi_{\mathrm{GA}}(\boldsymbol{r}), & \text{if} \quad r_1 = \cdots = r_N, \\ \infty, & \text{otherwise.}\end{cases} \tag{55}$$

REFERENCES

[1] D. Zhu, B. Zhou, and S. X. Yang, "A novel algorithm of multi-auvs task assignment and path planning based on biologically inspired neural network map," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 333–342, 2020.

[2] X. Hou, J. Wang, T. Bai, Y. Deng, Y. Ren, and L. Hanzo, "Environment-aware auv trajectory design and resource management for multi-tier underwater computing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 2, pp. 474–490, 2023.

[3] Z. Yang, J. Du, Z. Xia, C. Jiang, A. Benslimane, and Y. Ren, "Secure and cooperative target tracking via auv swarm: A reinforcement learning approach," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[4] Z. Wang, J. Du, C. Jiang, Z. Xia, Y. Ren, and Z. Han, "Task scheduling for distributed auv network target hunting and searching: An energy-efficient aoi-aware dmappo approach," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 8271–8285, 2022.

[5] A. Pan, K. Bhatia, and J. Steinhardt, "The effects of reward misspecification: Mapping and mitigating misaligned models," in *International Conference on Learning Representations*, 2022.

[6] Y. Qiu, Y. Jin, L. Yu, J. Wang, Y. Wang, and X. Zhang, "Improving sample efficiency of multi-agent reinforcement learning with non-expert policy for flocking control," *IEEE Internet of Things Journal*, 2023.

[7] X. Hou, J. Wang, C. Jiang, Z. Meng, J. Chen, and Y. Ren, "Efficient federated learning for metaverse via dynamic user selection, gradient quantization and resource allocation," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 850–866, 2024.

[8] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell, "Bridging offline reinforcement learning and imitation learning: A tale of pessimism," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 702–11 716, 2021.

[9] J. Wang, H. Du, D. Niyato, Z. Xiong, J. Kang, B. Ai, Z. Han, and D. I. Kim, "Generative artificial intelligence assisted wireless sensing: Human flow detection in practical communication environments," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2024.

[10] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, pp. 4572–4580, 2016.

[11] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, vol. 5, 2020.

[12] R. Yang, C. Bai, X. Ma, Z. Wang, C. Zhang, and L. Han, "Rorl: Robust offline reinforcement learning via conservative smoothing," *Advances in neural information processing systems*, vol. 35, pp. 23 851–23 866, 2022.

[13] T. Z. Muslimov and R. A. Munasypov, "Coordinated uav standoff tracking of moving target based on lyapunov vector fields," in *2020 International Conference Nonlinearity, Information and Robotics (NIR)*. IEEE, 2020, pp. 1–5.

[14] Y. Shen, Y. Qu, C. Dong, F. Zhou, and Q. Wu, "Joint training and resource allocation optimization for federated learning in uav swarm," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2272–2284, 2022.

[15] Y. Shou, B. Xu, A. Zhang, and T. Mei, "Virtual guidance-based coordinated tracking control of multi-autonomous underwater vehicles using composite neural learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5565–5574, 2021.

[16] X. Cao, D. Zhu, and S. X. Yang, "Multi-auv target search based on bioinspired neurodynamics model in 3-d underwater environments," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2364–2374, 2015.

[17] Y. Shou, B. Xu, A. Zhang, and T. Mei, "Virtual guidance-based coordinated tracking control of multi-autonomous underwater vehicles using composite neural learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5565–5574, 2021.

[18] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. Rajan, S. Mao, and X. Shen, "A unified framework for guiding generative ai with wireless perception in resource constrained mobile edge networks," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2024.

[19] W. Wei, J. Wang, J. Du, Z. Fang, Y. Ren, and C. P. Chen, "Differential game-based deep reinforcement learning in underwater target hunting task," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[20] Z. Xia, J. Du, J. Wang, C. Jiang, Y. Ren, G. Li, and Z. Han, "Multi-agent reinforcement learning aided intelligent uav swarm for target tracking," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 931–945, 2021.

[21] L. Yue, M. Lv, M. Yan, X. Zhao, A. Wu, L. Li, and J. Zuo, "Improving cooperative multi-target tracking control for uav swarm using multi-agent reinforcement learning," in *2023 9th International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2023, pp. 179–186.

[22] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of drones: Multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] L. Xie, S. Wang, S. Rosa, A. Markham, and N. Trigoni, "Learning with training wheels: speeding up training with a simple controller for deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6276–6283.

[24] K. Wan, D. Wu, B. Li, X. Gao, Z. Hu, and D. Chen, "Me-maddpg: An efficient learning-based motion planning method for multiple agents in complex environments," *International Journal of Intelligent Systems*, vol. 37, no. 3, pp. 2393–2427, 2022.

[25] S. Stevšić, T. Nägeli, J. Alonso-Mora, and O. Hilliges, "Sample efficient learning of path following and obstacle avoidance behavior for quadrotors," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3852–3859, 2018.

[26] J. Wang, P. Zhang, and Y. Wang, "Autonomous target tracking of multi-uav: A two-stage deep reinforcement learning approach with expert experience," *Applied Soft Computing*, vol. 145, p. 110604, 2023.

[27] G. Macaluso, A. Sestini, and A. Bagdanov, "Small dataset, big gains: Enhancing reinforcement learning by offline pre-training with model-based augmentation," in *The 2nd AAAI Workshop on Artificial Intelligence with Biased or Scarce Data (AIBSD)*, 02 2024, p. 4.

[28] T. I. Fossen, *Nonlinear modelling and control of underwater vehicles*. Universitetet i Trondheim (Norway), 1991.

[29] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[30] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[32] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, "Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning," in *International Conference on Learning Representations*, 2019.

[33] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[34] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[36] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[37] R. S. Sutton and A. G. Barto, "Reinforcement learning: an introduction MIT press," *Cambridge, MA*, vol. 22447, 2018.

[38] J. Filar and K. Vrieze, *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[39] H. Prasad and S. Bhatnagar, "A study of gradient descent schemes for general-sum stochastic games," *arXiv preprint arXiv:1507.00093*, 2015.

[40] J. Song, H. Ren, D. Sadigh, and S. Ermon, "Multi-agent generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 31, 2018.

[41] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[42] H. Furuta, Y. Matsuo, and S. S. Gu, "Generalized decision transformer for offline hindsight information matching," in *International Conference on Learning Representations*, 2022.

[43] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.