FISHER: An Efficient Sim2sim Training Framework Dedicated in Multi-AUV Target Tracking via Learning from Demonstrations

Guanwen Xie[1,*], Jingzehua Xu[1,*], Yimian Ding[1], Xinqi Wang[2], Dongfang Ma[3], Jingjing Wang[4(✉)], and Yong Ren[5]

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

Abstract. Multiple autonomous underwater vehicles (AUVs) target tracking problem is a significant challenge for AUV swarm control, which is crucial to the growth of the marine industry. To emphasize the great adaptability while tackling the limitations of reinforcement learning (RL) methods in Multi-AUV target tracking tasks, we propose an efficient two-stage learning from demonstrations (LfD) training framework, FISHER, based on few-shot expert demonstration, featuring imitation learning (IL) and offline reinforcement learning (ORL). In the first stage, we develop a sample-efficient algorithm, multi-agent discriminator actor-critic (MADAC), to facilitate the imitation of expert policy and the generation of offline datasets. In the second stage, based on the decision transformer (DT), the reward function-independent algorithm, multi-agent independent generalized decision transformer (MAIGDT) is utilized for further policy improvement. Simultaneously, we propose a simulation to simulation (sim2sim) method to facilitate the generation of expert trajectories, which is compatible with traditional methods like artificial potential field (APF). Through comparative experiments, we verify the improvement of the proposed MADAC and MAIGDT algorithms. Finally, full target tracking simulation processes show that FISHER can achieve performance comparable to expert demonstrations, thereby further demonstrating the strong practicality of FISHER framework. To accelerate relevant research in this direction, the code for simulation will be released as open-source.

Keywords: Autonomous underwater vehicles · Target tracking · Reinforcement learning · Simulation to simulation · Learning from demonstrations.

[*] These authors contributed equally to this work.

1 Introduction

Due to their powerful maneuverability and wide sensing capabilities, multiple autonomous underwater vehicles (AUVs) have broad application prospects in the construction of the Internet of Underwater Things (IoUT) network, underwater rescue, target tracking etc. Particularly, target tracking is a representative issue, which requires AUVs to keep close to the moving target, while keeping excellent action consistencies and avoiding AUV-target or AUV-AUV collisions simultaneously. The numerous prerequisites make it challenging to use traditional control methods to achieve effective formation control. Fortunately, reinforcement learning (RL) provides an efficient way to solve this problem, due to its strong ability to feature expression and robustness to meet various demands. However, there still exists some challenges when applying RL: (1) The performance of agents considerably relies on the design of the reward function, especially for multiple objectives. A poorly designed reward function may lead to undesirable outcomes, such as sub-optimal policies and reward hacking. (2) RL methods need abundant interactions between agents and the environment, which leads to heavy costs of time and computing resources.

Thanks to the recent booming development of learning from demonstrations (LfD) in RL, these aforementioned issues can be effectively addressed. Imitation learning (IL) and offline reinforcement learning (ORL) are two primary topics in this field. On the one hand, the objective of IL is to learn a policy effectively from limited expert demonstrations. Most current methods are mainly based on generative adversarial imitation learning (GAIL) [5], which aligns the policy with expert demonstrations by training a discriminator. However, the original GAIL suffers from the instability of generative adversarial methods. Besides, original GAIL generally utilizes policy obtained via on-policy algorithms for training, such as proximal policy optimization (PPO) [11], which results in low sample efficiency and unsatisfactory performance. Furthermore, IL methods typically have various limitations, such as poor generalization and multitasking performance. On the other hand, ORL is proposed to obtain a generally applicable policy given a dataset with possibly sub-optimal trajectories, without additional online data collection. However traditional ORL methods still rely on the design of the reward function. Besides, ORL usually makes high demands on the scale of the offline dataset, otherwise, bad outcomes may be brought forth[9]. These factors mentioned before make it difficult for IL and ORL to be deployed independently in practical LfD scenarios.

To fully exploit the advantages of RL in dealing with complex demands while overcoming its main challenge, we propose a two-stage LfD training network named FISHER, and apply it for the underwater multi-AUV target tracking tasks. Our main contributions can be presented as follows:

- We introduce FISHER, an efficient and reward function irrelevant LfD training framework using few-shot expert demonstrations, which can be easily generated utilizing traditional tracking methods like APF, and transformed by proposed simulation to simulation (sim2sim) procedure. Then IL is used

for efficient policy improvement, and ORL is utilized to further enhance both generalization and multi-task performance. The framework is deployed on a high-precision simulation platform for marine target tracking tasks.
- To tackle problems in the GAIL-based IL algorithm, we introduce the discriminator actor-critic (DAC) algorithm and expand it into the multi-agent DAC (MADAC). Leveraging the replay buffer, off-policy RL algorithm, and improvements for generative adversarial networks (GAN) training, MADAC shows a significant boost in training efficiency, while reducing computation loss and demand for environment interaction.
- To tackle the challenges in ORL, we introduce the multi-agent independent generalized decision transformer (MAIGDT), without depending on a reward function. Then we demonstrate through comparative experiments and evaluation of Multi-AUV target tracking processes that MAIGDT significantly outperforms traditional methods, thereby validating the effectiveness of our training framework. To accelerate relevant research in this direction, the code for the simulation will be released as open-source.

2 System Model and Problem Formulation

In this section, we briefly present the AUV dynamic model and underwater detection model for modeling and better simulating the target tracking task. Then, the Markov decision process (MDP) is introduced to lay a foundation for proposed FISHER framework.

Considering that a moving target T, a group of $N(N>1)$ AUVs are responsible for tracking the target, and both the target and AUVs move on the same plane with a fixed depth d. Target's position vector is denoted as $\boldsymbol{p}_T = [x_T(t), y_T(t)]$. Similarly, the position vectors of tracker AUVs are denoted as $\boldsymbol{p}_i = [x_i(t), y_i(t)], i \in \boldsymbol{N}, \boldsymbol{N} = \{1, ..., N\}$. Besides, there are also M obstacles $\{o_1, ..., o_M\}$ in the environment, and each AUV needs to track the target while avoiding these obstacles as much as possible.

2.1 AUV Dynamics Model

Since AUVs track the target in the horizontal plane, without loss of generality, their dynamic models can be expressed by the three-degree of freedom underdrive model. We denote that AUV i has the body reference frame $\boldsymbol{v}_i = [v_{i,x}(t), v_{i,y}(t), w_i]$, and the world reference 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 are surge velocity, sway velocity, angular velocity and yaw angle, respectively. The basic kinematic equation of an AUV is given by

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

Then, the kinetic equation of AUV can be expressed as

$$\boldsymbol{M}_i\dot{\boldsymbol{v}}_i + \boldsymbol{C}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{D}_i(\boldsymbol{v}_i)\boldsymbol{v}_i + \boldsymbol{G}_i(\boldsymbol{\eta}_i) = \boldsymbol{\tau}_i, \tag{2}$$

where $\boldsymbol{M}_i$ represents the inertia matrix including the additional mass of AUV i, $\boldsymbol{C}_i$ denotes the Coriolis centripetal force matrix, while $\boldsymbol{D}_i$ is the damping matrix caused by viscous hydrodynamic. Besides, $\boldsymbol{G}_i$ represents the composite matrix of gravity and buoyancy, and $\boldsymbol{\tau}_i$ is the control input of AUV i. Additionally, we discretize the kinematic and kinetic equations above over time, and we obtain

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\eta}_t + \mathit{\Delta} T \cdot \boldsymbol{J}\left(\boldsymbol{\eta}_t\right) \boldsymbol{v}_t, \tag{3}$$

$$\boldsymbol{v}_{t+1} = \boldsymbol{v}_t + \mathit{\Delta} T \cdot \boldsymbol{M}^{-1} F\left(\boldsymbol{\eta}_t, \boldsymbol{v}_t\right), \tag{4}$$

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

2.2 Underwater Detection Model

We use the active sonar equation of the underwater environment to model the detection process between the AUV and target, i.e.

$$EM = SL - 2TL + TS - (NL - DI) - DT, \tag{5}$$

where the unit of all parameters is dB, and SL, TL, TS, NL, 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 detection threshold and the echo margin of sonar, respectively.

Similarly, we model the communication between AUVs using the passive sonar equation, and we have

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

Furthermore, TL is related to AUV-target distance d and center acoustic frequency f, i.e.

$$TL = 20 \lg(d) + d \times \alpha(f) \times 10^{-3}, \tag{7}$$

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

where $\alpha\left(f\right)$ is an 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 can be expressed as

$$r_c = \underset{d}{\operatorname{argmax}} \{EM(d) \geq 0\}. \tag{9}$$

2.3 Markov Decision Process

Given the assumption that AUV's behavior only depends on the current state, the target tracking process can be modeled as a Markov decision process (MDP), which includes state space $\mathcal{S}_i$, action space $\mathcal{A}_i$, and reward function $\mathcal{R}_i$.

State space $\mathcal{S}_i$: In MDP, the state of each AUV is observable, and the ith AUV's state $\boldsymbol{s}_i\left(t\right) \in \mathbb{R}^{4N+4}$ in the state space $\mathcal{S}_i$ can be expressed as

$$\begin{aligned}\boldsymbol{s}_i(t) = \big\{ & x_{i,t}(t), y_{i,t}(t), v_{x_{i,t}}(t), v_{y_{i,t}}(t), x_{i,j}(t), y_{i,j}(t), v_{x_{i,j}}(t), v_{y_{i,j}}(t), \\ & EM_k \cos\left(\theta_{ok_i}\right), EM_k \sin\left(\theta_{ok_i}\right)\big\}_{j\in \boldsymbol{N}, j\neq i, k\in\{1,\ldots,N_o\}}\,,\end{aligned} \tag{10}$$

which consists of three parts: 1) The initial 4 terms denote the target's position and velocity, whose values are defined in the coordinate system of the polar axis in which the direction of the AUV i is facing, namely $x_{i,t}\left(t\right) = d_i\left(t\right) \cos\left(\theta_{i,t}\left(t\right)\right)$. The same applies hereinafter. 2) The intermediate $4N-4$ terms are other AUVs' positions and velocities. 3) The final $2N_o$ terms represent the obstacles' position. We assume that an AUV can detect at most N_o of the nearest obstacles, and the echo margin of the obstacle k is EM_k. When less than N_o obstacles are detected, corresponding EM is set to 0dB.

Action space $\mathcal{A}_i$: The action $\boldsymbol{a}_i\left(t\right)$ in the action space $\mathcal{A}_i$ can be expressed as two high-level control parameters

$$\boldsymbol{a}_i(t) = \left[\boldsymbol{v}_i(t), \boldsymbol{w}_i(t)\right], \tag{11}$$

where $||\boldsymbol{v}_i(t)|| = \sqrt{v_{i,x}(t)^2 + v_{i,y}(t)^2} \in [0, v_{\max}]$ and $||\boldsymbol{w}_i(t)|| \in [0, w_{\max}]$.

Reward function $\mathcal{R}_i$: To some degree, the reward function can reflect the tracking performance of the AUV swarm. It is utilized for traditional RL algorithms to train agents for comparison. **The reward function is not utilized for training the FISHER framework.** There are three parts that are important for the target tracking task

$$r_{ti_i}(t) = \begin{cases} d_i(t) - d^t_{\min}(t), & d_i(t) > d^t_{\min}, \\ 0, & d_i(t) < d^t_{\min}, \end{cases} \tag{12}$$

$$r_{o_i}(t) = \sum_{j=1, j\neq i}^{N} \left(d_{\text{safe}} - d_{ij}(t)\right) + \sum_{k=1, k\neq i}^{M} \left(d_{\text{safe}} - d_{i,o_k}(t)\right), d_{ij}(t) < d_{\text{safe}} \text{ and } d_{i,o_k}(t) < d_{\text{safe}}, \tag{13}$$

$$r_{l_i}(t) = \begin{cases} d^l_i(t) - d^l_{\min}(t), & d^l_i(t) > d^l_{\min}(t), \\ 0, & d^l_i(t) < d^l_{\min}(t). \end{cases} \tag{14}$$

The definition and meaning of each term in Eq. (12)∼(14) are elaborated as follows: 1) The reward term r_{ti_i} is used to encourage a single AUV to track the target, which can be determined by the distance between AUV i and the target. $d^t_{\min}$ denotes the optimal distance from the target. We also introduce the term $r_{tc}(t) = \max_i \left\{r_{ti_i}(t)\right\}$ to reflect overall tracking performance. 2) The penalty term r_{o_i} is used to avoid collision with all other AUVs and obstacles. For each AUV or obstacle that is less than the safe distance d_{safe} from the current AUV, a corresponding penalty will be applied and all the penalties will be summed up. 3) The reward term r_{l_i} is utilized to encourage each AUV to keep good swarm consistency. To be intuitive, we use a simplified form here, namely an AUV cannot be too far from the nearest AUV in the swarm. where $d_i^l(t) = \min_j \{d_{ij}(t)\}$. Similarly, $d_{\min}^l$ is the optimal distance from other AUVs.

Furthermore, to adjust the positivity of the AUVs tracking target by adjusting the term r_{ti_i} and r_{tc}, we set two weight factors, w_1 and w_2 for r_{ti_i} and r_{tc}, respectively, and we put three settings for signifying them: **Cooperative**: $w_1 = 1$, $w_2 = 0$; **Mixed**: $w_1 = 0.5$, $w_2 = 0.5$; **Split**: $w_1 = 0$, $w_2 = 1$. The cooperative setting only requires that at least one AUV approach the target, while the split setting encourages each AUV to maintain proximity to the target individually. Finally, the overall reward function can be calculated as follows

$$r_i(t) = a\left(w_1 r_{tc}(t) + w_2 r_{ti_i}(t)\right) + w_3 r_{o_i}(t) + w_4 r_{l_i}(t) + r_b, \tag{15}$$

where $W = [aw_1, aw_2, w_3, w_4]$ is the weight vector and r_b is a bias constant.

3 Methodology

In this section, we introduce the training framework FISHER for the multi-AUV target tracking task based on few-shot expert demonstrations. We first introduce our sim2sim method in detail, which can easily generate expert trajectories. Then we present two stages of FISHER: MADAC for sample-efficient imitation learning and MAIGDT for training generalizable policy to complete the target tracking task. The schematic diagram of our proposed training framework is depicted in Fig. 1.

3.1 Sim2sim Expert Demonstration Generation

It is of great difficulty to directly generate expert trajectories through traditional RL methods when the designed reward function is sub-optimal. Therefore, it's necessary to simplify the generation process with the proposed sim2sim method.

To be specific, our sim2sim method consists of the following components: 1) we first simplify the tracking environment, ignoring underwater and other environmental effects, and considering AUVs and the target as particles. This allows us to take advantage of traditional target tracking methods, such as artificial potential field (APF) [6], to obtain AUVs' trajectories. 2) Then we train a simple policy for a single AUV to reach a specific point in the underwater simulation environment, without any obstacle. The state space is composed of positions of the AUV and target point, with the action space being consistent with that adopted by FISHER. The reward function is the negative value of the Euclidean distance to the target point. It's quite simple to optimize the training objective, and the tracking error can be quickly reduced to less than 0.2m. 3) Finally, we deploy the aforementioned model to each AUV to complete the target tracking tasks in the simulation environment, under the guidance of the AUVs' optimal position obtained previously. We can add some disturbance parameters and repeat this procedure to enhance the diversity of expert trajectories.

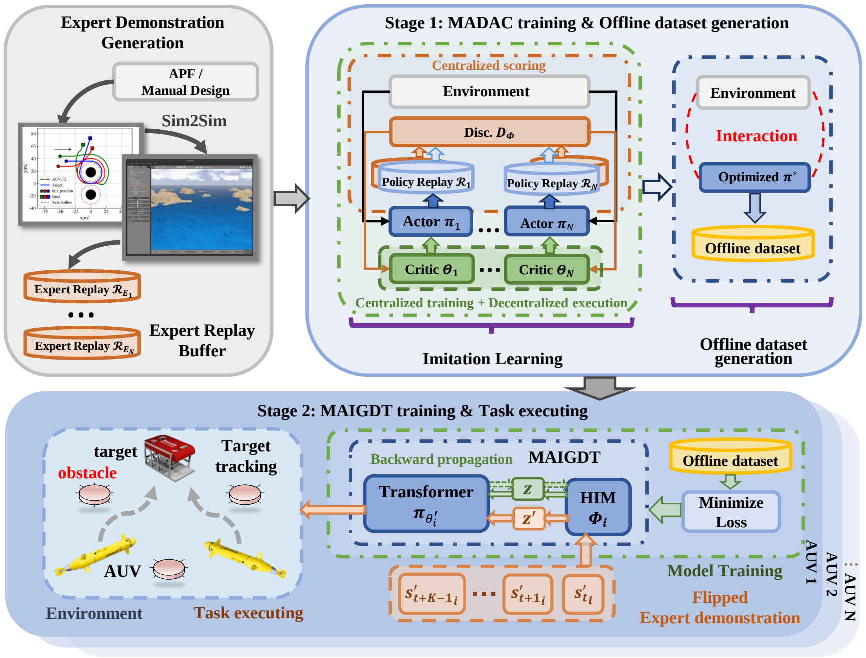


Fig. 1: The schematic diagram of our proposed training framework FISHER.

3.2 Multi-Agent Discriminator Actor-Critic

We achieve policy improvement by employing IL using a small number of expert trajectories. Existing methods primarily based on GAIL, which trains a discriminator to distinguish between expert trajectories and policy-generated trajectories, thereby guiding policy improvement and making the generated trajectories approximate the expert trajectories. The primary issue of original GAIL is the need for extensive environmental interaction. To address this, Kostrikov *et al.* [7] introduced the DAC algorithm, which utilized a replay buffer to store previously generated trajectories. Then, similar to Song *et al.* [12] of expanding GAIL to the multi-agent scenario, we can optimize the discriminator network D_i of AUV i as

$$\mathcal{L}_{\boldsymbol{D}} = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\boldsymbol{\mathcal{R}}}\left[\sum_{i=1}^{N}\log\left(D_i(\boldsymbol{s},\boldsymbol{a})\right)\right] + \sum_{i=1}^{N}\mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim\pi_{E_i}}\left[\log\left(1-D_i(\boldsymbol{s},\boldsymbol{a})\right)\right], \quad (16)$$

where $\boldsymbol{\mathcal{R}}$ denotes the replay buffer, $\boldsymbol{s} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$, $\boldsymbol{a} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$ and π_{E_i} represents global state and action, and the expert policy of AUV i. The output score D_i of the discriminator can guide policy improvement utilizing off-policy RL algorithms. In addition, our proposed DAC algorithm makes some refinements, such as introducing the absorbing state s_a [13] for termination of episodes, and introducing some improvements for GAN for stabilizing training, including

Algorithm 1 FISHER Algorithm

```
Initialize the training environment, including 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 $\pi_{\theta_i}$ with corresponding critic network, DT model parameters $\theta'_i$ with its anti-casual transformer $\mathit{\Phi}_i$ of AUV $i$.
for each episode $k$ do                                        ▷ Stage 1 : IL with MADAC
    Reset the training environment.
    for each environment timestep $t$ do                       ▷ Collect trajectories
        Sample action $\boldsymbol{a}_{t_i} \sim \pi_{\theta_i}(\cdot \mid \boldsymbol{s}_{t_i})$
        Collect the next state $s_{t+1_i}$ from environment
        Update replay buffer $\mathcal{R}_i \leftarrow \mathcal{R}_i \cup \{(\boldsymbol{s}_{t_i}, \boldsymbol{a}_{t_i}, \cdot, \boldsymbol{s}_{t+1_i})\}$
    end for
    for each IL gradient step do                               ▷ Update discriminator
        Sample transitions from replay $\{(\boldsymbol{s}_t, \boldsymbol{a}_t, \cdot, \cdot)\}_{t=1}^{B} \sim \boldsymbol{\mathcal{R}}$, $\{(\boldsymbol{s}'_t, \boldsymbol{a}'_t, \cdot, \cdot)\}_{t=1}^{B} \sim \boldsymbol{\mathcal{R}}_E$.
        $\mathcal{L}_D = \sum_{b=1}^{B} \log D(\boldsymbol{s}_b, \boldsymbol{a}_b) - \log(1 - D(\boldsymbol{s}'_b, \boldsymbol{a}'_b))$ .
        Update $D$ with Adam+GP+Spectral Normalization
    end for
    for each RL gradient step do                               ▷ Update policy
        Sample $\{(\boldsymbol{s}_{t_i}, \boldsymbol{a}_{t_i}, \cdot, \boldsymbol{s}_{t+1_i})\}_{t=1}^{B} \sim \mathcal{R}_i$
        for $b = 1, \ldots, B$ do
            $r_i \leftarrow \log D(\boldsymbol{s}_{b_i}, \boldsymbol{a}_{b_i}) - \log(1 - D(\boldsymbol{s}_{b_i}, \boldsymbol{a}_{b_i}))$
            $(\boldsymbol{s}_{b_i}, \boldsymbol{a}_{b_i}, \cdot, \boldsymbol{s}_{b+1_i}) \leftarrow (\boldsymbol{s}_{b_i}, \boldsymbol{a}_{b_i}, r_i, \boldsymbol{s}_{b+1_i})$
        end for
        Update $\pi_{\theta_i}$ with SAC[4]+CTDE
    end for
end for
Collect trajectories $\tau_i$ using optimal policy $\pi^*_{\theta_i}$.        ▷ Make offline datasets
Sample $n$ batches of sequence with length $K$ from the offline dataset $\tau_i$.
for each GDT gradient step do                                  ▷ Stage 2 : ORL with MAIGDT
    Flip the state of sequences and get $\boldsymbol{z}_i$ vectors from anti-casual transformer $\mathit{\Phi}_i$.
    Update models of GDT by Adam updating on $\mathit{\Phi}_i$ and $\theta'_i$ by $L_{\text{MSE}}(\theta'_i)$ of equation (Eq. (18)).
end for
Get expert demonstration $\tau'_{E_i}$ for imitation
while target tracking task timestep $t$ do                     ▷ FISHER evaluation loop
    Get sequence from timestep $t$ to $t + K - 1$ of $\tau'_{E_i}$, flip the state of sequence and get $\boldsymbol{z}_{t_i}$ vector from anti-casual transformer $\mathit{\Phi}_i$
    Predict action based on vector $\boldsymbol{z}_i$, state $\boldsymbol{s}_i$ and $\boldsymbol{a}_i$ of previous $K$ timesteps
end while
```

gradient penalty (GP) [3] and spectral normalization (SN) [10]. Next, we turn our attention to extending DAC to multi-AUV scenarios. We tested two representative architectures for this extension. The centralized setting sets a discriminator for all AUVs, namely $D_1 = \ldots = D_N = D$, while the policies are trained in the centralized setting. In contrast, The decentralized setting sets a discriminator for each AUV, namely $D_i(\boldsymbol{s}, \boldsymbol{a}) = D_i(\boldsymbol{s}_i, \boldsymbol{a}_i)$. In this paper, we adopt the centralized setting due to its stability in training. We will also compare the performance of the two settings in the subsequent sections.

3.3 Multi-Agent Independent Generalized Decision Transformer

We utilize ORL for further policy improvement, effectively enhancing the generalization and multitasking performance. Traditional ORL methods optimize the Bellman objective, therefore the estimation accuracy of the policy gradient is seriously affected by the sufficiency of the dataset. Thus, DT [1] is introduced to abstract ORL problems into seq2seq problems and use sequences to model targets.

DT employs a transformer-based GPT-2 model for autoregressive training and action prediction. Original DT reshapes the trajectory in the offline dataset. A modified trajectory can be denoted as

$$\tau_i' = \left(\hat{r}_{1_i}, \boldsymbol{s}_{1_i}, \boldsymbol{a}_{1_i}, \hat{r}_{2_i}, \boldsymbol{s}_{2_i}, \boldsymbol{a}_{2_i}, \ldots, \hat{r}_{T_i}, \boldsymbol{s}_{T_i}, \boldsymbol{a}_{T_i}\right), \tag{17}$$

where $\hat{r}_{t_i} = \sum_{t'=t}^{T} r_{t_i'}$ denotes the expected total reward of AUV i. When training the model, we sample n batches of sequence with length K from the offline dataset. The prediction head corresponding to the input token $s_i\left(t\right)$ is trained to predict $\hat{a}_i\left(t\right)$, and the losses for each timestep are averaged. The training objective of the DT model $\pi_{\theta_i'}$ is illustrated as

$$\max_{\pi_{\theta_i'}} J'(\theta_i') = \min_{\pi_{\theta_i'}} \mathcal{L}_{\mathrm{MSE}}(\theta_i') = \min_{\pi_{\theta_i'}} [-\frac{1}{N}\sum_{j=1}^{N}(a_j - \hat{a}_j)^2]. \tag{18}$$

However, the original DT still relies on the design of the reward function. Furuta *et al.* [2] have demonstrated that DT is doing hindsight information matching, namely using future information to find positive examples with certain contextual parameter values (e.g. returns-to-go for DT). Therefore, we can make DT to match the state transition of expert demonstrations, rather than predicting action using return-to-go. This can be achieved by replacing the return-to-go of the original DT with the information statistics of sequences.

Specifically, we use a second transformer $\varPhi$, which takes a reverse-order state sequence as input. The output of transformer $\varPhi$ is a vector $\boldsymbol{z}$ that contains the information of future states. Since $\varPhi$ is differentiable to DT's action-prediction loss, $\varPhi$ can learn sufficient features of states by optimizing equation Eq. (18), and DT is enough to match any distribution to an arbitrary precision. When executing target tracking tasks, we specify an expert trajectory τ_E' and use $\varPhi$ to get features of it, which guides DT to efficiently imitate the demonstration. Fig. 1 also shows this process, where $\boldsymbol{z}$ substitutes the return-to-go to facilitate the transformer predicting the action.

Table 1: Simulation Parameters

Parameters	Values
Hydroacoustic parameters SL,TS,DI,DT,NL	100dB, 3dB, 3dB, 20dB, 30dB
Hydroacoustic transmit frequency f	10kHz
Maximum speed parameters $v_{\max}$, $\omega_{\max}$	2.4m/s, 1.0rad/s
Reward weight factor (a, w_3, w_4, r_b)	-0.125, -0.2, -0.1, 3
Distance parameters $\left(d^t_{\min}, d_{\text{safe}}, d^l_{\min}\right)$	12m, 8m, 16m
Learning rate	0.0003(MADAC),0.0001(MAIGDT)
MADAC gradient penalty factor	1.0
MAIGDT context length K	20

4 Simulation Experiments

In this section, we first introduce the utilized experiment settings. Then we detail the design of experiment scenarios and corresponding expert trajectories, followed by the discussion of experiment results and analysis in Section 4.3.

4.1 Experiment Settings

We verify the effectiveness of the proposed FISHER by simulating the whole process of a two-AUV swarm tracking target. In the beginning, the positions of AUVs are (-20m, 8m) and (-20m, -8m) relative to the target, which is oriented at the x-axis initially. Then, the policies control AUVs at a frequency of 12.5Hz. Other representative parameters of the simulation are provided in Table 1 for a summary.

4.2 Design of Scenarios and Expert Trajectories

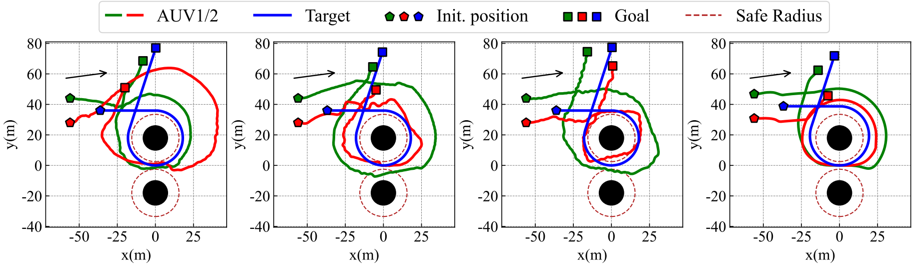
We design several scenarios that feature different target moving trajectories and obstacle distributions, and all of them have corresponding expert trajectories of two AUVs. These scenarios are divided into the two parts as follows:

The first part possesses sparse obstacle(s). As the scenarios are not complex, reward function-based RL methods can achieve acceptable performance. The obstacle distribution and expert trajectories are shown in Fig. 2, and we label these scenarios as Scenario 1, and Scenario 2.

However, the second part with dense obstacles makes it hard to correspond with the reward function, for AUVs must reorganize the formation while passing through obstacles. Therefore, we introduce some performance indicators to evaluate these scenarios, which will be detailed in Section 4.3. Similarly, we introduce two scenarios and label them as Scenario 3 and Scenario 4.

4.3 Experiment Results and Analysis

We first evaluate the effectiveness of the two stages of FISHER, MADAC and MAIGDT, by comparative experiments in the scenario(1/2) of sparse obstacle(s) based on the accumulated reward obtained by AUVs. Then, we perform



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Fig. 2: Trajectories of the target, expert demonstrations of AUVs and obstacle distribution of different scenarios. (a) Scenario 1 (sparse obstacle). (b) Scenario 2 (sparse obstacle). (c) Scenario 3 (dense obstacles). (d) Scenario 4 (dense obstacles).

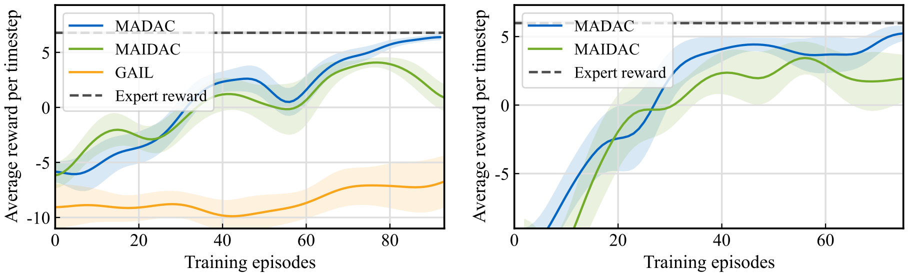
the target tracking tasks in the scenarios(3/4) of dense obstacles, using some performance indicators to evaluate FISHER and a representative MARL baseline, SAC following the centralized training and decentralized execution(CTDE) manner(denoted as MASAC). Thereby, we can verify the effectiveness and advantages of the proposed FISHER framework.

To start with, we conduct experiments to compare MADAC with the original GAIL implementation (GAIL + PPO)[1] with the centralized multi-agent setting, and the decentralized settings of multi-agent DAC (named MAIDAC), given 10 expert trajectories. And the experiment results are shown in Fig. 3.

Observations from Fig. 3(a) illustrate that MAIDAC converges more rapidly and stably than the original GAIL, due to the introduction of replay buffer and off-policy SAC algorithm. And MAIDAC finally achieves expert-level reward after 90 training episodes in Fig. 3(a). Besides, both MADAC and MAIDAC can converge rapidly, but only MADAC can achieve expert-level reward, and MADAC possesses stronger stability. As the number of AUVs increases, MAIDAC shows more distinct disadvantages compared to MADAC, and discussions are deferred to future work.

Moreover, we evaluate the demand of the proposed MADAC algorithm for the number of expert demonstrations, and we conduct comparative experiments in Scenario 1. As Fig. 4 shows, more expert demonstrations can accelerate the training speed and stability. However, generally speaking, our algorithm does not require an extensive number of trajectories, and satisfactory results can be obtained with a limited number of demonstrations. Next, we turn our attention to comparing our proposed MAIGDT with conservative Q-learning (CQL) [8][2], a typical ORL baseline. The trajectories of the offline dataset adopted here are generally sub-optimal, and there is a significant imbalance in the rewards obtained

[1] https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail
[2] https://github.com/aviralkumar2907/CQL



(a) Scenario 1 (sparse obstacle)

(b) Scenario 2 (sparse obstacle)

Fig. 3: Average total reward curves of all AUVs relying on MADAC, MAIDAC and GAIL for training in different scenarios. (a) Scenario 1 (sparse obstacle). (b) Scenario 2 (sparse obstacle).

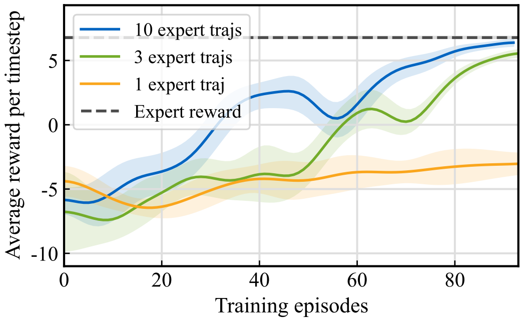
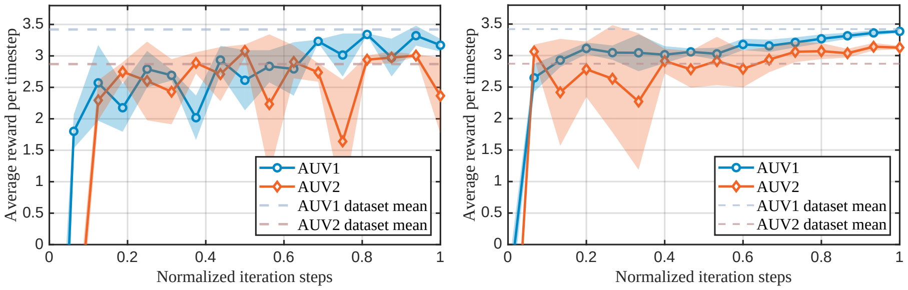


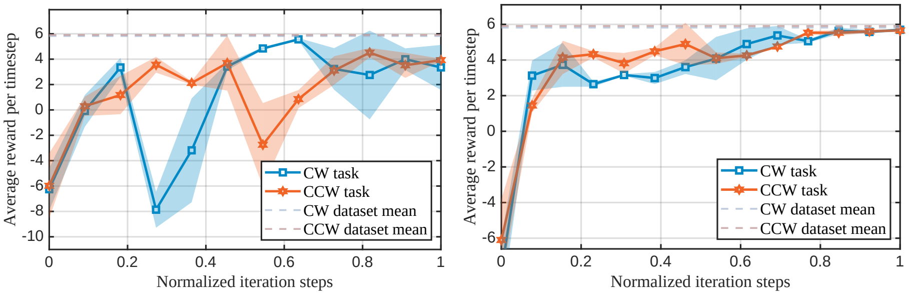
Fig. 4: Average total reward curves of all AUVs utilizing MADAC with different numbers of trajectories for training in Scenario 1 (sparse obstacle).



(a) Utilize CQL for training

(b) Utilize MAIGDT for training

Fig. 5: Average total reward curves of each AUV utilizing different algorithms for training in Scenario 1. (a) Utilize CQL for training. (b) Utilize MAIGDT for training.



(a) Utilize CQL for training

(b) Utilize MAIGDT for training

Fig. 6: Average total reward curves of all AUVs utilizing different algorithms in CW and CCW tasks taken from Scenario 2. (a) Utilize CQL for training. (b) Utilize MAIGDT for training.

by two AUVs, making it challenging to obtain satisfactory outcomes. The outcomes of experiments are depicted in Fig. 5. As Fig. 5 shows, MAIGDT outperforms CQL in terms of training stability and final performance, with MAIGDT's final performance exceeding the dataset's average.

Then we evaluate the multi-task capability of the proposed MAIGDT. To realize this, we design two tasks, both derived from Scenario 2, but with forward directions being clockwise (CW) and counterclockwise (CCW), and conduct experiments to compare the performance between CQL and MAIGDT. As illustrated in Fig. 6, the training of CQL is quite unstable, with the rewards of the two AUVs fluctuating drastically. In contrast, our proposed MAIGDT demonstrates commendable performance and robust stability across both tasks.

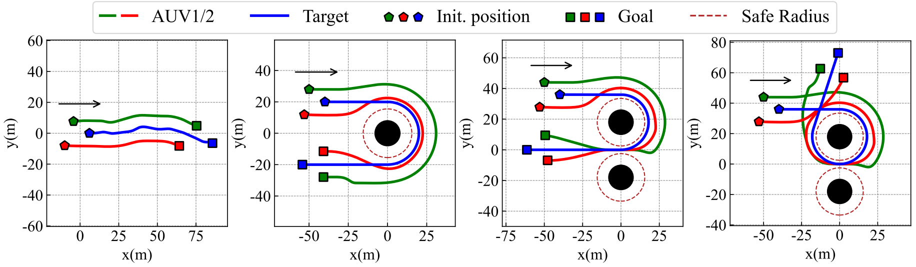
Finally, we perform the target tracking tasks in the scenario of dense obstacles (Scenario 4). For comparison, we utilize the MASAC baseline trained with three reward settings respectively, as demonstrated in Section 2.3, to reveal the limitations of the reward function design.

For convenience, we introduce six performance indicators similar to Yang *et al.*[14], i.e., minimum distance mean, minimum distance standard deviation, consistency mean, consistency standard deviation, minimum distance, and danger duration. Minimum distance is the distance between the target and the AUV closest to the target. Minimum obstacle distance represents the minimum distance between the obstacle and the AUVs during the whole process. Consistency refers to the distance between AUVs. While danger duration denotes the time duration during which there is at least one AUV that is less than $d_{\text{safe}} = 8\text{m}$ away from an obstacle. To ensure the validity of the results, we train the policy of AUVs until convergence from scratch 3 times to test the training stability.

As shown in Table 1, the optimal values for minimum distance mean and consistency mean are 12m and 16m, respectively. The corresponding results in Scenario 4 are shown in Table 2, while the trajectories of AUVs recorded from the physical simulation environment are shown in Fig. 7, similar to Fig. 2.

Table 2: Performance of AUVs tracking target in three settings and proposed FISHER framework in Scenario 4. The result is shown in the format of $a \pm b$, where b signifies the standard deviation between policies from multiple training sessions.

Experiments	Cooperative	Mixed	Split	FISHER
$\mathbb{E}$(min-distance)	14.64m±0.27m	14.96m±1.04m	**13.88m±0.17m**	14.48m±0.14m
Std(min-distance)	2.60m±0.39m	3.11m±0.43m	1.82m±0.09m	**1.30m±0.02m**
$\mathbb{E}$(consistency)	26.50m±1.82m	17.56m±0.70m	**16.20m±0.66m**	16.64m±0.36m
Std(consistency)	8.19m±1.60m	3.36m±1.08m	2.79m±0.43m	**1.37m±0.04m**
Min(obs distance)	6.87m±1.48m	5.57m±1.65m	5.48m±1.23m	**10.41m±0.09m**
Danger time	9.29s±6.05s	11.59s±6.16s	16.11s±3.53s	**0.00s±0.00s**



(a) Cooperative (b) Mixed (c) Split (d) FISHER

Fig. 7: Representative tracking trajectories of AUVs utilizing MASAC with three reward settings and FISHER. (a) Cooperative setting. (b) Mixed setting. (c) Split setting. (d) FISHER.

It's evident that the benefits of a multi-AUV swarm are scarcely exhibited under the cooperative setting, while AUVs under the split setting tend to disregard the risks of crashing while tracking targets. In addition, the results of MASAC are notably unstable, with severe jiggling while AUVs track the target, reflecting the intrinsic shortcomings of traditional RL methods dependent on reward functions. In contrast, the proposed FISHER effectively acquires knowledge from expert policies, achieving performance that is close to the expert policy, and demonstrating strong stability in both the training process and task execution.

5 Conclusion

In this paper, we propose an efficient training framework FISHER and apply it to train multiple AUVs to complete target tracking tasks via LfD, while under the guidance of expert demonstration transformed by sim2sim. There are two stages in the FISHER: the first stage employs the MADAC algorithm to imitate the expert policy with high sample efficiency and then generates offline datasets. The second stage utilizes MAIGDT, enabling AUVs to make further policy improvements without designing a reward function. Comparative experiments are conducted to compare the performance of MADAC and GAIL, as well as MAIGDT and CQL, to show the superiority of our proposed algorithms. Finally, the target tracking task is evaluated in detail to demonstrate our proposed FISHER framework's remarkable performance and practicality. As a part of future work, we plan to further improve the realism of the simulation, and conduct both simulation and real-world experiments in even more complex tasks.

References

1. Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., Mordatch, I.: Decision transformer: Reinforcement learning via sequence modeling. In: Advances in Neural Information Processing Systems. pp. 15084–15097 (2021)
2. Furuta, H., Matsuo, Y., Gu, S.S.: Generalized decision transformer for offline hindsight information matching. In: International Conference on Learning Representations. pp. 1–28 (2022)
3. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5769–5779 (2017)
4. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning. pp. 1861–1870 (2018)
5. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. pp. 4572–4580 (2016)
6. Khatib, O.: Real-time obstacle avoidance system for manipulators and mobile robots. The International Journal of Robotics Research **5**(1), 90–98 (1986)
7. Kostrikov, I., Agrawal, K.K., Dwibedi, D., Levine, S.: Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In: International Conference on Learning Representations. pp. 1–15 (2019)
8. Kumar, A., Zhou, A., Tucker, G., Levine, S.: Conservative q-learning for offline reinforcement learning. In: Advances in Neural Information Processing Systems. pp. 1179–1191 (2020)
9. Macaluso, G., Sestini, A., Bagdanov, A.: 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). p. 4 (2024)
10. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
11. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
12. Song, J., Ren, H., Sadigh, D., Ermon, S.: Multi-agent generative adversarial imitation learning. In: Advances in Neural Information Processing Systems. pp. 1–12 (2018)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. A Bradford Book (2018)
14. Yang, Z., Du, J., Xia, Z., Jiang, C., Benslimane, A., Ren, Y.: Secure and cooperative target tracking via auv swarm: A reinforcement learning approach. In: IEEE Global Communications Conference. pp. 1–6 (2021)