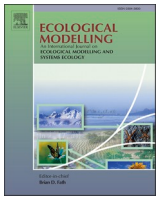
Ecological Modelling 477 (2023) 110259

Contents lists available at ScienceDirect

Ecological Modelling

journal homepage: www.elsevier.com/locate/ecolmodel



Modeling collective motion for fish schooling via multi-agent reinforcement learning

Xin Wang [a,b,c], Shuo Liu [a,b,c], Yifan Yu [a,b,c], Shengzhi Yue [a,b,c], Ying Liu [c,d], Fumin Zhang [e], Yuanshan Lin [a,b,c,*]

[a] College of Information Engineering, Dalian Ocean University, Dalian 116023, China
[b] Liaoning Key Laboratory for Marine Information Technology, Dalian 116023, China
[c] Key Laboratory of Environment Controlled Aquaculture (Dalian Ocean University) Ministry of Education, Dalian 116023, China
[d] College of Biosystems Engeering and Food Science, Zhejiang University, Hangzhou 310030, China
[e] Cheng Kar-Shun Robotics Institute, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China

ARTICLE INFO

Keywords:
Collective motion
Collective behavior
Animal behavior
Multi-agent reinforcement learning
Reinforcement learning

ABSTRACT

Complex collective motion patterns can emerge from very simple local interactions among individual agents. However, it is still unclear how and why the interactions among individuals lead to the emergence of collective motion. Modeling is an effective way to understand the mechanisms that govern collective animal motions. In this work, to avoid imposing fixed sets of rules on collective motion models a priori as classical approaches do, we propose a new method of modeling collective motion for fish schooling via multi-agent reinforcement learning. We model each fish individual as an artificial learning agent, whose policy is acquired by using mean field Q-learning (MFQ). The observation of each fish agent is represented as a multi-channel image, where each channel describes a different feature, such as an agent's position or an agent's orientation. The policy of an agent is approximated with a neural network trained with the MFQ algorithm, during which, agents are rewarded (or penalized) according to the number of neighbors and consecutive collisions between individuals. We study the dynamics of collective motion that emerge from the learned policy. The experimental results show that the learned policy can produce collective motion in groups of various sizes. In addition, three different collective motion patterns observed in nature emerged during the training process. The learned policy can help us gain new insight into how and why individual interactions lead to collective motion. This study also demonstrates that multi-agent reinforcement learning has great potential to be a new approach for analysis and modeling of collective motion.

1. Introduction

Collective motion is a common and interesting phenomenon in nature. It has received considerable attention in theoretical biology, ethology, cognition, and ecology. One of the most noticeable characteristics of animal groups is their ability to generate various complex swarming motion patterns. Extensive research recognizes that collective motion is self-organized and mainly results from local interactions between individuals who lack the ability to understand or directly control the collective (Schaerf et al., 2017). Modeling is a good way to understand the mechanisms that govern collective animal motions. Fish are a common model organism for the study of collective motion because they are swarming animals and extremely easy to obtain. In the past few decades, many collective motion models have been proposed, which can be mainly divided into two categories: group-based models and agent-based models. Group-based models (Kolpas et al., 2007; Yates et al., 2009; Bode et al., 2010; Jhawar et al., 2020; Jhawar and Guttal, 2020) are very successful at describing macroscopic properties such as phase transitions and metastable states, but they usually ignore the specific interactions among individuals. Agent-based models (Reynolds, 1987; Vicsek et al., 1995; Katz et al., 2011; Gautrais et al., 2012; Mwaffo et al., 2015) can integrate the detailed properties of an individual into the model when they describe the interactions among individuals. The main idea of agent-based models is that individuals are reduced to featureless particles. The linear velocity of each particle is usually assumed to be constant and its behavior can only be controlled by

* Corresponding author.
E-mail address: linyuanshan@dlou.edu.cn (Y. Lin).

https://doi.org/10.1016/j.ecolmodel.2022.110259
Received 2 May 2022; Received in revised form 16 December 2022; Accepted 22 December 2022
Available online 28 December 2022

changing its heading angle. The heading angle is affected by a fixed set of rules: alignment, cohesion, and separation. The three rules enable individuals in a group to maintain their orientation in the same direction, keeping the distance among each other relatively compact while avoiding collisions. These rules can be implemented by various models, which are mainly divided into two categories: self-propelled particles (SPP) models (Couzin et al., 2002; Ginelli and Chaté, 2010; Hemelrijk and Hildenbrandt, 2012; Vicsek and Zafeiris, 2012; Deutsch et al., 2012) and social forces (SF) models (Lukeman et al., 2010; Herbert-Read et al., 2011; Hinz and De Polavieja, 2017; Wright et al., 2019; Shaebani et al., 2020). In SPP models, fish individuals change their motion in response to others according to an alignment component, a cohesion component, and a collision avoidance component. Similarly, in the SF models, the fish are viewed as Newtonian particles subject to social and physical forces, ensuring group cohesion and reflecting the interaction (e.g. drag) with the environment (Collignon et al., 2016). Although the two models differ, both fundamentally rely on a fixed set of rules, and assume that the rules are known in advance. Obtaining such rules is very challenging, often requiring deep insight and rich domain knowledge. Usually, many complex factors will affect collective motion, so it is unrealistic to take all possible factors into account when creating a set of rules. Further, the rules are created by experts that likely have biases which may cause the collective motion patterns to be not properly explained. Instead, individuals in collective motion should be viewed as agents whose policy is learned from interacting with their environment and neighbors.

Therefore, to overcome those static rules, some researchers have begun to use reinforcement learning (RL) to model collective motion (Morihiro et al., 2008; Shimada et al., 2018; Ried et al., 2019; López-Incera et al., 2020; Costa et al., 2020). For example, an earlier RL-based collective motion model for the self-organized grouping of individuals was proposed in 2008 (Morihiro et al., 2008), in which the reward function was constructed based on the distances between individuals. Shimada et al. (2018) took into account not only the distances between individuals but also the directions of individuals when they designed the reward function. Specifically, they developed three metrics to measure the difference between the behavior of Reynolds' Boids model (Reynolds, 1987), and the behavior of the RL-based model. Conceptually similar work has also been carried out by Costa et al. (2020), in which they obtained a collective motion model based on RL by maximizing the group-level objective function (total reward during a simulated episode) representing the desired collective configuration. They showed that collective motion models learned through different reward functions can make fish schools form different motion patterns. These approaches obtained RL-based models by shaping reward functions with implicit assumptions, without directly providing information relating to the underlying forces of the three rules of alignment, cohesion, and separation. While these learned models have provided useful insights, they cannot guarantee that the assumed behavior is accurate at the individual level and moreover, they do not seem to address its origin (i.e. why individuals would respond in one way or another).

The increased likelihood of survival in the presence of predators is recognized as one of the benefits of collective motion for animals. Hence, some researchers proposed predator-prey models (Morihiro et al., 2008; Hahn et al., 2019; Sunehag et al., 2019; Wang et al., 2020) in which a predator and prey were placed in an environment where the predator attempted to catch the prey. At the end of each timestep, prey are rewarded to encourage policies that allow them to survive as long as possible. Simulation results showed that this simple reward led to emergent flocking behavior. In contrast, Durve et al. (2020), designed a reward function where agents that lose adjacent neighbors from within their perceptual field are penalized. Results showed that collective motion emerges spontaneously in a RL process from the minimization of the rate of neighbor loss. In these studies, however, the observation or state of an agent was usually represented with high-level variables, such as the angle between the current moving direction of the focal agent and the average direction of its neighbors, or the poses (position and orientation) of the k nearest neighbors. However, there is a large difference between such representation and the real sensory perception of an agent. Furthermore, most existing RL-based approaches obtain an agent's policy by using independent single-agent reinforcement learning algorithms in a multi-agent setting, in which other agents are considered as part of the environment. This straightforward approach has the problem of causing non-stationarity in the state transitions. As an agent tries to adapt its actions in certain states, other agents are doing the same, but they are considered as part of the environment for the first agent. Since the states of the agent are not independent, we cannot develop a policy that depends only on the observed states because the Markov property for each agent is lost.

To solve this environmental uncertainty, independent single-agent reinforcement learning algorithms were used in a multi-agent setting, leading to multi-agent reinforcement learning (MARL). In recent years, there have been many successful uses of multi-agent reinforcement learning (MARL) such as in poker games (Moravčík et al., 2017; Brown and Sandholm, 2018), DOTA 2 (Berner et al., 2019), and StarCraft II (Vinyals et al., 2019). As an animal group can be viewed as a multi-agent system where individuals can be modeled as agents, we hypothesize that group behaviors like fish schooling should result from the responses to very generic benefits in a spatial environment. So, multi-agent reinforcement learning algorithms should be a potentially effective method to model collective motion in fish schools. In this paper, we propose a new method to model the collective motion in fish schools, in which the fish individuals are modeled as artificial learning agents who can learn their movement policy through interacting with the environment. The proposed method uses Mean Field Theory (MFT) to simplify the interactions between agents.

In order to model fish sensory ability in a more detailed and realistic way, we use an image-like representation (multi-channel image) to describe the observation of a fish agent. The policy of a fish agent is approximated with a neural network trained by MFQ, which is good at handling RL tasks with large numbers of agents. In the training, fish agents are rewarded (or penalized) according to the number of neighbors and consecutive collisions among fish agents. In our model of collective motion, the interaction rules with neighbors are not fixed in advance, instead the fish agents develop them based on their past experiences. In contrast to other learning-based works, we avoid handcrafted shaping rewards, specific actions, or dynamics that would directly encourage coordination across agents. And we also analyze the properties of the collective motion patterns that emerge from the learned policy of fish agents. In a word, we aim to illustrate that multi-agent reinforcement learning has the potential to be a new approach for the analysis and modeling of collective motion and can generate new insight.

2. MFQ-based method for modeling collective motion

Well-known complex collective motion can emerge from simple local interactions of agents who lack the ability to observe or directly control the collective. And the local interactions of agents can be viewed as the policies that agents use to respond to their observations. So, modeling collective motion can be simplified to finding the agents' policies that can generate complex collective motion patterns commonly observed in nature. We attempt to obtain the policies of agents that generate collective motion patterns via RL. In the RL framework, agents interact with the environment in a closed-loop manner and learn by maximizing the accumulative reward obtained from their interactions with the environment. To this end, we describe six components: 1) the Markov Decision Process model, 2) the kinematic model of the fish agent and the environment, 3) the observation model of the fish agent, 4) the representation of the fish agent's action, 5) the reward function and policy representation, and 6) the MFQ-based learning algorithm in the following subsections. Furthermore, we give three indicators to quantify

and analyze the properties of collective motion.

2.1. Problem formulation

In this paper, we try to address the problem of obtaining agents' policies that generate the desired collective patterns using a reinforcement learning framework. Therefore, we need to develop an MDP to describe the collective motion of fish schools.

In this paper, each individual in the swarm is modeled as an artificial learning agent, which has its own local sensory capability and decision-making capability on how to respond to the sensory input. In addition, we assume that 1) all agents in the swarm are identical, 2) they have the same dynamics, observation capabilities, and action capabilities, and 3) every agent can only partially observe the global swarm system state due to its limited sensory capability. Note that since all agents in the swarm are homogeneous, the state space, the action space, the transition model, the observation model, and the policy for an agent are assumed to be invariant to the permutations of the agents and define individual model $\mathbb{A}$. In the following, we describe the components of a multi-agent Markov decision process (including the individual model and the swarm model) below. The individual model describes the homogeneous agent common architecture and is defined as a tuple $\mathbb{A} = (S^{(i)}, O^{(i)}, A^{(i)}, P^{(i)}, \pi^{(i)})$,where i represents each individual agent.

$S^{(i)}$: the agent's local state space, a part of the global state of the swarm system;
$O^{(i)}$: the agent's local observation space;
$A^{(i)}$: the agent's action space;
$P^{(i)}$: a mapping $S^{(i)} \times A^{(i)} \times S^{(i+1)} \to [0, \infty)$, the local state transition model of the agent, such as the agent's kinematics model
$R^{(i)}$: a mapping $O^{(i)} \times \{A^{(i)}\}^N \to R^{(i)}$, the agent-level reward function which can only access the agent's local observation;
$\pi^{(i)}$: a mapping $O^{(i)} \times A^{(i)} \to [0, 1]$, the local policy of the agent, that is, its decision depends on its current neighborhood only and not on the whole swarm state.
Based on the individual model $\mathbb{A}$, the swarm model is defined as a tuple $W = (N, \mathbb{A}, S, P, O, R, \gamma)$, where:
N: the number of agents in the swarm;
$\mathbb{A}$: above individual models;
S: the swarm's global state space, a joint space $\{S^{(i)}\}^N$, swarm state $s = (s^1, \ldots, s^N) \in S$, $s^1 \in S^{(i)}$, $j = 1, \ldots, N$;
P: the mapping $S \times \mathbb{A} \times S \to [0, \infty)$, the swarm's global state transition model;
O: the mapping $S \times \{1, \ldots, N\} \to O^{(i)}$, agent's observation model which determines the local observation $o^j \in O^{(i)}$ for agent j at given swarm state $s \in S$;
R: the mapping $S \times \mathbb{A} \to R$, the swarm-level global reward function which can access the global state;
γ: discount factor, $\gamma \in (0, 1)$

Like SwarMDPs, the proposed MDP with the individual model exploits the system symmetries and provides a more compact system representation, especially when the number of agents in a swarm is large.

Based on the proposed MDP, we can use multi-agent deep reinforcement learning to find a local policy that can produce collective motion. Specifically, since collective motion is viewed as the result of cooperative tasks among individuals, finding a local policy can be formulated as an optimization problem which maximizes the total accumulative reward. It is defined as Eq. (1).

$$\max_{\pi^{(i)}} J(\pi^{(i)}) = \max_{\pi^{(i)}} E[\sum_{t=0}^{\infty} (R(s_t, a_t, s_{t+1}) + \sum_{j=1}^{N} R^{(i)}(o_t^j, a_t)) | \pi^{(i)}] \quad (1)$$

Where $R(.)$ and $R^{(i)}(.)$ are the swarm-level function and individual-level reward function respectively, $E[.]$ is expectation operation, a_t is joint action at time step t $a_t = (\pi^{(i)}(o_t^1), \ldots, \pi^{(i)}(o_t^N))$.. It should be pointed out that Eq. (1) may have a swarm-level reward function, an individual-level reward function or both in a task. In MDPs, a swarm-level reward function can be used to describe the system-level goal of a task such as the desired collective motion pattern.

2.2. A kinematic model of the fish agent and the environment

In real life, a fish can freely swim by beating its fins in certain three-dimensional spaces, such as ponds, rivers, lakes, or marine environments, and fish schooling usually consists of many fish of the same species with approximately the same body size and age (Hemelrijk and Kunz, 2005). For simplicity, we define fish schools as many homogenous fish agents moving together with movements in a two-dimensional box with periodic boundary conditions. That means that if an agent leaves the box to the right, it will immediately enter it again from the left (same for the other direction or around the top and bottom).

Let us define a global reference frame, X and Y in a horizontal plane with length expressed in an arbitrary unit, which we call body length (BL). In this reference frame, the state of an agent j is represented by its position (x^j, y^j) and orientation ϕ^j, that is,

$$s^j = (x^j, y^j, \phi^j) \in S^{(i)} = \{(x, y, \phi) \in L^3 : 0 \le x < x_{max}, 0 \le y < y_{max}, 0 \le \phi < 2\pi\}.$$

The linear and angular velocities of agent j are represented by v^j and ω^j respectively. Thus, a fish agent moving in the 2D plane can be modeled as a sample unicycle (see De Souza et al., 2021 for more detail). So, the kinematic model of a fish agent j can be given by Eq. (2).

$$\begin{cases} \dot{x}^j = v^j \cos\phi^j \\ \dot{y}^j = v^j \sin\phi^j \\ \dot{\phi}^j = \omega^j \end{cases} \quad (2)$$

It is assumed that the fish agent has first-order kinematics during movement. That is, the position and orientation of the fish agent are determined by its own speed and angular velocity, respectively. A fish agent's state is updated at discrete time steps, where the operator *mod* is used to implement periodic boundary conditions. It is defined as Eq. (3).

$$\begin{cases} x_{t+1}^j = (x_t^j + v^j \cos\phi_t^j) \bmod x_{\max} \\ y_{t+1}^j = (y_t^j + v^j \sin\phi_t^j) \bmod y_{\max} \\ \phi_{t+1}^j = (\phi_t^j + \omega^j) \bmod 2\pi \end{cases} \quad (3)$$

2.3. The fish agent's local observation model

It has been hypothesized that vision is a primary mode of perception for fish. While fish can achieve a field of vision close to 360° by beating their tail, their perceptual distance is limited (McComb and Kajiura, 2008). Thus, we can think of the perceptual area of a fish agent as a circle of radius L (see Fig. 1).

Here, we assume that the fish agent can perceive spatial information (position and orientation), which contains the location relationship information between the focal agents and its neighbors within its perceptual area. As grid-type information is more convenient to feed into the neural network, for simplicity, we approximate the perceptual field with a square of side L located at the focal agent's position. Here, we use k*k image-like representation with multiple channels to encode the spatial information as shown in Fig. 1 (right panel). The parameter k is usually set with an odd number to keep the focus agent in the middle of the grid. Every channel can represent a different feature, where the first channel is a binary matrix with 1 indicating the position of agents (the focal agent itself and its neighbors) and 0 otherwise. The value of

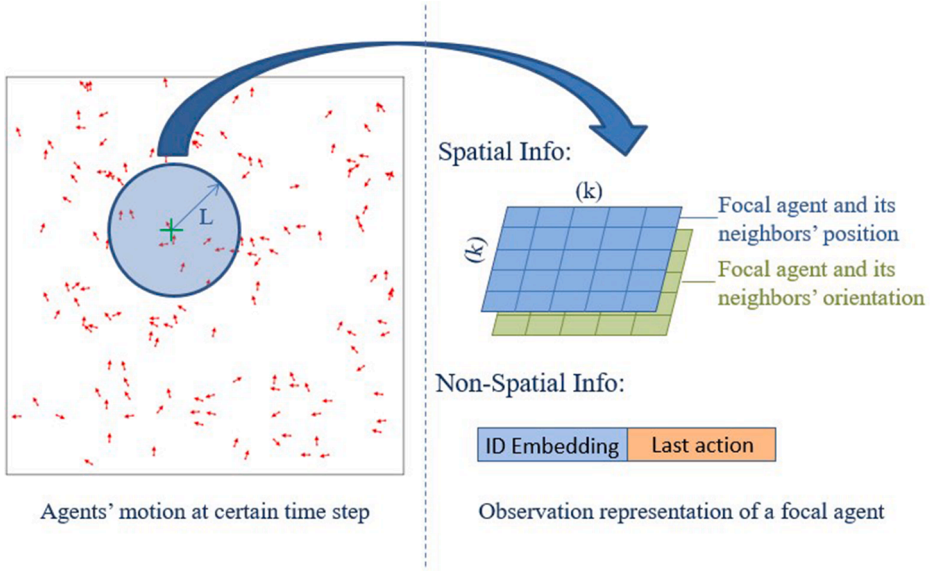


Fig. 1. The agent's observation representation.

each cell (pixel) in the first channel can be determined with Eq. (4).

$$o^j[0][r][c] = I\left\{0 \le r = \frac{(x^i - x^j + L) * (k)}{2L} \le k-1 \text{ and } c = \frac{(y^i - y^j + L) * (k)}{2L} \le k-1\right\},\ i = [1, \ldots, N] \tag{4}$$

where, $o^j[0][r][c]$ represents the value of the pixel (r, c) in the first channel for focal agent j; L is the focal agent perceptual range;(x^j, y^j) is the position of the focal agent j; similarly, (x^i, y^i) is the position of the neighbor i within the perceptual area of the focal agent j; $\lfloor . \rfloor$ is the floor round operator; $I\{\}$ is the indicator operator which takes on a value of 1 if its argument is true, and 0 otherwise.

The second channel represents the orientation of agents (the agent itself and its neighbors). Similarly, the value of each cell (pixel) in the first channel can be determined with Eq. (5).

$$o^j[0][r][c] = \phi^i * I\left\{0 \le r = \frac{(x^i - x^j + L) * (k)}{2L} \le k-1 \text{ and } c = \frac{(y^i - y^j + L) * (k)}{2L} \le k-1\right\},\ i = [1, \ldots, N] \tag{5}$$

where, ϕ^i is the orientation of the neighbor j. In addition, we also assume that the fish agent can remember certain non-spatial information, such as its ID (representing its observation information), last action, last reward, and so on. So, an agent's observation includes spatial information and non-spatial information. For non-spatial information, we use a one-dimensional vector to represent it.

2.4. Representation of fish agent's action

As described in Section 2.2, a fish agent can control its own linear velocity and angular velocity. Like most existing collective behavior models, here the linear velocity of the fish agent is considered constant, thus the angular velocity becomes the only control input that can change its orientation. Since fish have a limited turning angle, we assume that the angular velocity lies within the range $[-\omega_{max}, \omega_{max}]$. We discretize the action space $[-\omega_{max}, \omega_{max}]$ into M equally-sized intervals, as shown in Fig. 2. The parameter M is usually set with an odd number to keep the current orientation in the middle of the action space. The blue arrow is the orientation of the focal fish agent at the current time step.

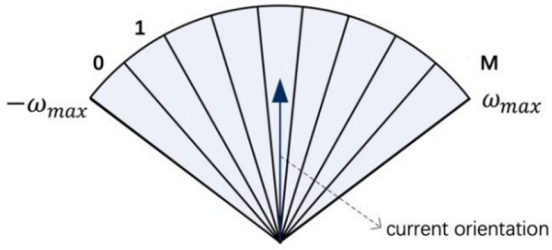


Fig. 2. Focal agent's action representation.

2.5. Reward function

The reward function plays an important role in RL that determines what the agent's motivations are (i.e. how the agent "ought" to behave). If the reward function is designed properly, the agents will learn better and faster. Collective motion in fish schools may have many benefits, such as access to information via local enhancement, detecting predators with the many-eyes effect, and group defense against predators via the confusion effect, which are recognized by most biologists. From this perspective, the more neighbors a fish has, the more benefits it gains. So, the number of neighbors should be included as a factor in the reward function. Further, as collisions are rare in real life, the number of collisions also should be included as another factor of the reward function. Thus, we can express the individual-level reward function for agent j as follows as Eq. (6).

$$R^{(i)}\left(o_t^j, a_t\right) = R^{(i)}\left(o_t^j, a_t^1, \ldots, a_t^j, \ldots, a_t^N\right) = K_n * N_n\left(o_t^j\right) + K_c * N_c\left(o_t^j\right) \tag{6}$$

where, $N_n(o_t^j)$ is a function that determines the number of neighbors in a given local observation o_t^j; similarly, $N_c(o_t^j)$ is a function that determines the continuous collision number of the agent j in given observation history; K_n and K_c are the coefficients of the number of neighbors and the number of collisions, respectively.

It should be noted that while a RL model may contain a swarm-level reward function, we do not use one in this paper as imposing direct rules to force collective motion is contrary to our original objective.

2.6. MFQ-based learning algorithm

As described in Section 2.2, fish schooling usually contains dozens or even hundreds of individual fish and each individual fish is modeled as a learning agent. Therefore, obtaining the fish's policy is an RL problem with many agents, which is very intractable due to the curse of dimensionality and the exponential growth of agent interactions. Fortunately, the Mean Field Q learning (MFQ) proposed by Yang et al. (2018), is very good at handling complex RL with many agents. Therefore, we here use the MFQ algorithm to find a shared local policy for agents that can generate collective motion. In the MFQ-based learning algorithm, the interactions within the population of fish agents are approximated by those between the focal fish agent and the average effect from the neighboring agents. That is, the influence of all neighboring fish agents on the focal fish agent is approximated to that of the virtual mean fish agent on the focal fish agent.

In most multi-agent reinforcement learning, the joint action grows proportionally with respect to the number of fish agents, N. To address this, MFQ uses the Q-function which uses the factorized pairwise local interactions as shown in Eq. (7).

$$Q^j(s,a) = \frac{1}{n^j}\sum_{k\in\eta(j)} Q^j\left(s,a^j,a^k\right) \tag{7}$$

where, n^j is the number of neighbors of fish agent j and $\eta(j)$ is the index set of neighboring fish agents.

Next, the pairwise local interactions can be further approximated under certain conditions by using the mean-field theory. It is defined as Eq. (8).

$$\frac{1}{n^j}\sum_{k\in\eta(j)} Q^j\left(s,a^j,a^k\right) \approx Q^j\left(s,a^j,\frac{1}{n^j}\sum_{k\in\eta(j)} a^k\right) = Q^j\left(s,a^j,\overline{a}^j\right) \tag{8}$$

where, $\overline{a}^j$ is the mean action over the neighborhood of the focal fish agent j. So, the mean action $\overline{a}^j$ can be viewed as the action of the virtual fish agent. Thus, many pairwise interactions can be simplified to those between the focal agent and the virtual fish agent. It is worth noting that a^k is a one-hot vector encoding and $\overline{a}^j$ is a vector of fractions corresponding to the probability that each action may be executed by a fish agent.

Once we obtain the mean field Q function Q^j, its corresponding policy can be obtained according to Eq. (9). So, with the mean-field approximation, the MARL problem is converted into finding the optimal mean field Q^j. If Q^j is parameterized with the weights ϕ^j, it is trained by minimizing the loss function with gradient-based optimizers where y^j is the target mean field value calculated with the target Q function $Q^j\_$ parameterized with the weights $\phi^j\_$ (Eq. (10)).

$$\pi^j\left(a^j\middle|s,\overline{a}^j\right) = \frac{\exp\left(-\beta Q^j\left(s,a^j,\overline{a}^j\right)\right)}{\sum_{a^{j'}\in A}\exp\left(-\beta Q^j\left(s,a^j,\overline{a}^j\right)\right)} \tag{9}$$

$$L\left(\phi^j\right) = \left(Q^j_{\phi^j}\left(s,a^j,\overline{a}^j\right) - y^j\right)^2 \tag{10}$$

The mean field Q-function Q^j, called the evaluation network, is approximated with a neural network. Similarly, the target Q function $Q^j\_$ is approximated with another neural network to reduce the correlation of data. The two neural networks have the same structure with three data streams in their first half, see Fig. 3. The first stream handles the fish agent's spatial observation using convolutional layers, the second stream deals with non-spatial features (containing the last action, the last reward) using a fully connected layer, and the third stream encodes the mean action information (coming from the virtual fish). In the second half, the three parts are concatenated together, then passed forward

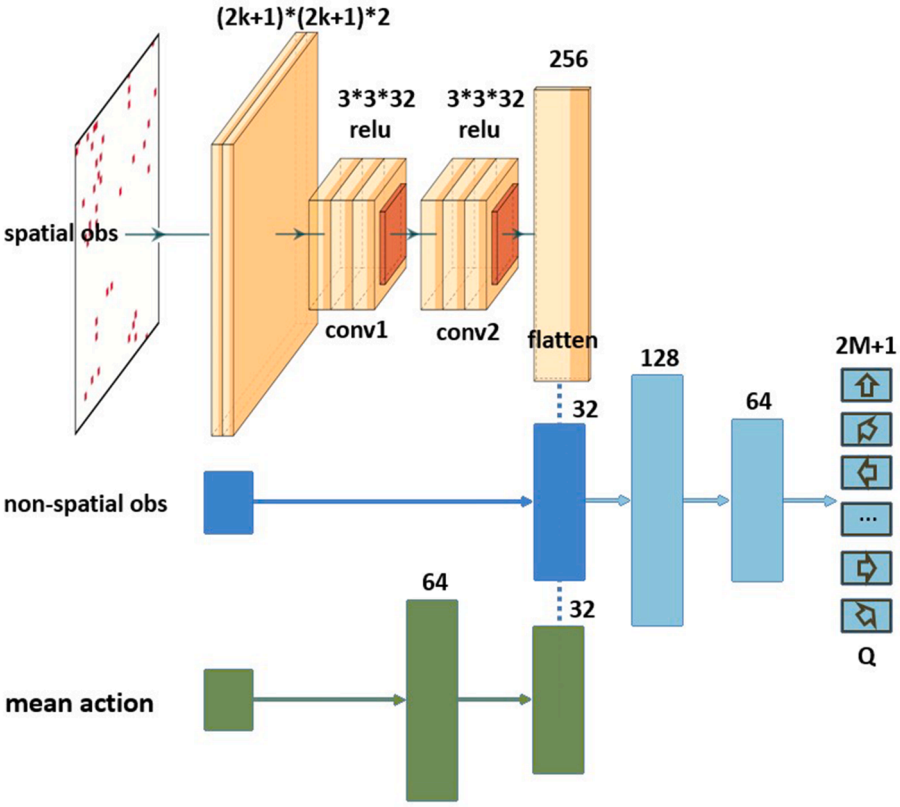


Fig. 3. Q neural network architecture.

to two fully-connected layers, and finally output a single Q value. It is worth noting that the mean action in the figure is the mean action of all neighboring fish agents within the perceptual range of the fish agent. And we assume that the neighbors within the perceptual range of the focal fish agent will not change greatly in two adjacent time steps. Therefore, the mean action of the neighbors at the previous moment is used as an input into the neural network for training.

The pseudo-code of the MFQ-based learning algorithm for fish agents is shown as Algorithm 1. In the beginning, the algorithm initializes the evaluation network Q^j and target network $Q^j\_$ with random weights. At each time step, the algorithm does the following things. First, the fish agents receive their observations. Next, the fish agents sample actions from the evaluation network Q^j according to their observations and previous mean action. Then, the environment performs the actions of agents simultaneously and generates interaction data (current observations, current actions, new observations, and rewards) which is saved into the replay buffer. Finally, the algorithm optimizes the evaluation network Q with minibatch SGD or Adam, and uses a soft-update to update network $Q^j\_$.

2.7. Indicators for properties of collective motion

The order parameter, average number of neighbors, and average nearest neighbor ratio are indicators often used to analyze the properties of collective motion each of which we describe below.

(1) Order parameter (alignment)

The order parameter is an index to quantify the degree of order in the collective motion. It is used to evaluate the alignment of each fish agent in the school. The order parameter is defined as Eq. (11).

$$\varphi(t) = \frac{1}{N}\left|\sum_{i=1}^{N}\frac{v_i}{\| v_i\|}\right| \tag{11}$$

where N is the number of agents; v_i represents the velocity vector of agent i, $\| v_i\|$ represents the Euclidean norm of the velocity vector of agent i, t represents the t-th time step. The order parameter goes from 0 to 1, where $\varphi \to 0$ means that the orientations of the agents are disordered and $\varphi \to 1$ means that the agents are aligned.

Algorithm 1
MFQ-based learning algorithm for fish agents.

```
Initialize evaluation network Q^j and target network Q^j_ with random weights
  for i = 1,..., n epochs do
  # add agents with random poses to environment
  env.reset()
  for t = 1,..., k do
  # get observations of all agents from environment
  obs_t = env.get_observation()
  mean_act_t = env.get_neighbors_mean_act()
  # a_t = (π^(t)(o_t^1), ..., π^(t)(o_t^N)), using Eq. (9)
  act_t = models.act(obs_t, mean_act_t)
  # all fish agents move one step
  next_act_t = env.step(act_t)
  # all fish agents get a reward from environment by using Eq.(6)
  reward_t = env.get_reward()
  # save interaction data into replay buffer
  add_to_buffer(obs_t, act_t, next_act_t, reward_t)
  # optimizing with minibatch SGD or Adam
  data = get data from buffer
  divide data into M minibatches
  for k = 1, 2, ..., K do
  for i = 1, 2, ..., M do
  calculate loss function and its gradient e by using Eq. (10)
  Update the evaluation network Q^j by minimizing the loss function
  end for
  use the soft-update to update network Q^j_
  end for
  end for
  end for
```

(1) Average number of neighbors (cohesion)

The average number of neighbors is used to evaluate the cohesion of the fish agent school. It is defined as Eq. (12).

$$M = \frac{1}{N}\sum_{i=1}^{N} m_i \tag{12}$$

where, m_i is the number of neighbors of the ith agent, N is the number of agents.

(1) Average nearest-neighbor ratio (density)

The Nearest Neighbor Distance (NND) is used as a measure of the spatial relationship to explore the distribution pattern of plant or animal populations (Clark and Evans, 1954). Inspired by the above, we apply the average nearest-neighbor ratio (*ANNR*) between fish agents to evaluate the density and distribution of its neighbors defined in Eq. (13).

$$ANNR = \frac{D_{actual}}{D_{expect}} \tag{13}$$

$$D_{actual} = \frac{\sum_{i=1}^{N} d_i}{N} \tag{14}$$

$$D_{expect} = \frac{0.5}{\sqrt{\frac{N}{A}}} \tag{15}$$

where, D_{actual} is the actual average nearest neighbor distance as defined in Eq. (14). $D_{expected}$ is the average distance when neighbors surrounding the agent are evenly distributed as defined in Eq. (15). d_i is the distance between the i th agent and its nearest neighboring agent, N is the number of neighboring agents, and A is the area of the perceptual range of the agent. If $ANNR < 1$, it means that the density and distribution of its neighbors is higher than would be expected by random chance.

3. Simulation results

3.1. Experiments setting

In our experiments, fish agents move in a 60×60BL (Body Length) 2D

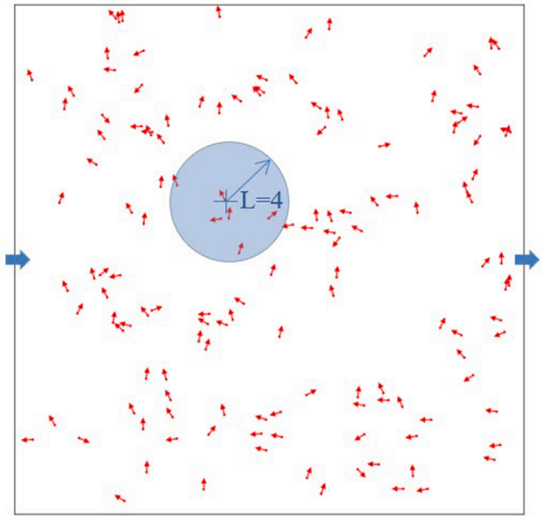


Fig. 4. The environment used in experiments.

box with periodic boundary conditions and no obstacles as shown in Fig. 4. For simplicity, we assume the living environment of fish has uniform physiochemical properties. The linear velocity of each fish agent is fixed to 3BL/s. The angular velocity is bounded between [-π/3, π/3] and discretized into 361 equal parts with each part representing an action (i.e. 361 actions). Thus, we have 361 discretized actions, and the observation radius of each fish agent was set to 4BL with no noise in the agent's observations. And at each time step, each agent chooses one of the 361 actions based on its observations.

Particularly common at the beginning of training, some fish agents may encounter consecutive collisions with their neighbors, termed an agent jam. In an agent jam, many fish agents cannot move forward causing the training process to halt. When this occurs, the simulation is restarted. If the fish agent continues to collide with fish agents up to 10 times in a row, then the fish agent's position is reset to a new random collision-free position.

3.2. Training process

The overview of training can be defined as follows, in the beginning, all neural networks of the MFQ are initialized with random values; then, fish agents interact with their neighbors and the environment to produce interaction experiences; when the interaction experiences in the buffer reach a defined amount, the training process is activated. Here, we set the learning rate to 0.0001, $\gamma = 0.95$, batch size of 64, and the size of the replay buffer is fixed to 1024. For the reward function (Eq. (6)), we set the coefficient $K_n = 0.1$ and $K_c = -0.1$.

During training, the complete process of interaction between fish agents and the environment has the structure displayed in Fig. 5, where:

- With the number of $N = 140$ fish agents, we execute training of 120 episodes during which the fish agents develop new behaviors to get the reward.
- At the beginning of each episode, all fish agents are placed at random positions and orientations. The termination condition of each episode is when the number of specified time steps (i.e., 400 steps) is reached.
- At each time step, every fish agent first evaluates its surroundings, then receives its corresponding observation. The observation of itself and the mean action of its neighbors serve as inputs for the current evaluation network which outputs an action for each fish agent. Once all agents' actions are obtained, the environment performs them simultaneously. Finally, all agents' interaction experiences are placed into the replay buffer.

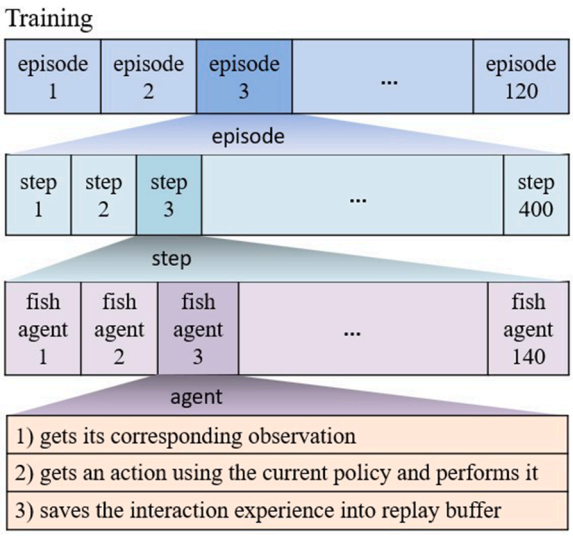


Fig. 5. The process of interaction between fish agents and the environment.

During the training session, we utilize the average reward that fish agents receive at each time step to evaluate the success of the learning process. As shown in Fig. 6, the average reward at each time step increases gradually as the number of training episodes grows. Specifically, the reward goes from 0.04 to 0.82, which demonstrates that the performance of the fish agent's policy gradually improves. The insets in Fig. 6 are some snapshots of fish agents' movement in the training process, occurring in the last few moments of the corresponding episode. From Fig. 6, we can observe that the fish agents move randomly and obtain a lower reward in the early phase of the training, while the fish agents eventually learn to form groups and receive a higher reward in the later phase of the training. As described in 2.6, since the number of fish agents is too large and every fish agent learns simultaneously, the learning may encounter non-stationarity of environment and high computational complexity. This may lead to the non-convergence of the model. However, in Fig. 6 we see that the fish agents can overcome learning difficulties by using MFQ, so that each fish agent is maximizing its own expected reward, and eventually learns to school.

3.3. Agents can learn schooling-like behaviors

In order to verify whether the learned policy can generate collective motion, we equipped 140 fish agents with the learned policy and placed them in the environment. Initial conditions, i.e. location and directions, were randomly generated for all fish agents. Then the fish agents moved according to their policy. Fig. 7 is a collection of snapshots of agents' configuration at various time steps t within an episode. In Fig. 7, we can see that fish agents autonomously formed a highly parallel group from the initially disordered configuration. That is, the learned policy can produce collective motion patterns observed in nature verifying the effectiveness of MFQ-based methods for modeling collective motion.

3.4. Collective motion dynamics

We analyze the properties of the collective motion that emerged from the learned policy of fish agents in this section. We focus on three main properties of the school, namely alignment, cohesion, and density. We quantify and analyze these properties with the order parameter, the average number of neighbors, and the average nearest-neighbor ratio.

3.4.1. Analysis of collective motion during training

This section shows the order parameter, the average number of neighbors, and the average nearest neighbor ratio vary for fish agents as training progresses. In Fig. 8(a), we show the evolution of the order parameter as training is advancing. The order parameter rapidly increases in the early phases of training, and then it fluctuates between 0.8 and 0.95 after about the 8th episode. This implies that the fish agent school eventually achieves a highly ordered configuration where fish agents can follow and align with the neighboring fish agents for collective motion. What is more noteworthy is that we did not explicitly specify the fish agents with information on how to align, thus the alignment was spontaneously induced by the collective motion of the fish agents.

Fig. 8(b) shows the evolution of the average number of neighbors as the training processes. We also observe that the average number of neighbors gradually increases and finally stabilizes in the range of 6–10. This reflects that the fish agents have learned a policy of collective motion that forms a cohesive school.

As shown in Fig. 8(c), the average nearest-neighbor ratio gradually decreases with respect to the number of training episodes, from about 1.3 to around 0.47. The value of the ANNR is higher in the beginning because the fish agents are just starting to learn to increase the number of neighbors around themselves. As the training progresses, the fish agents start to form a cohesive school—the average number of neighbors stabilizes, which leads to a decrease in ANNR.

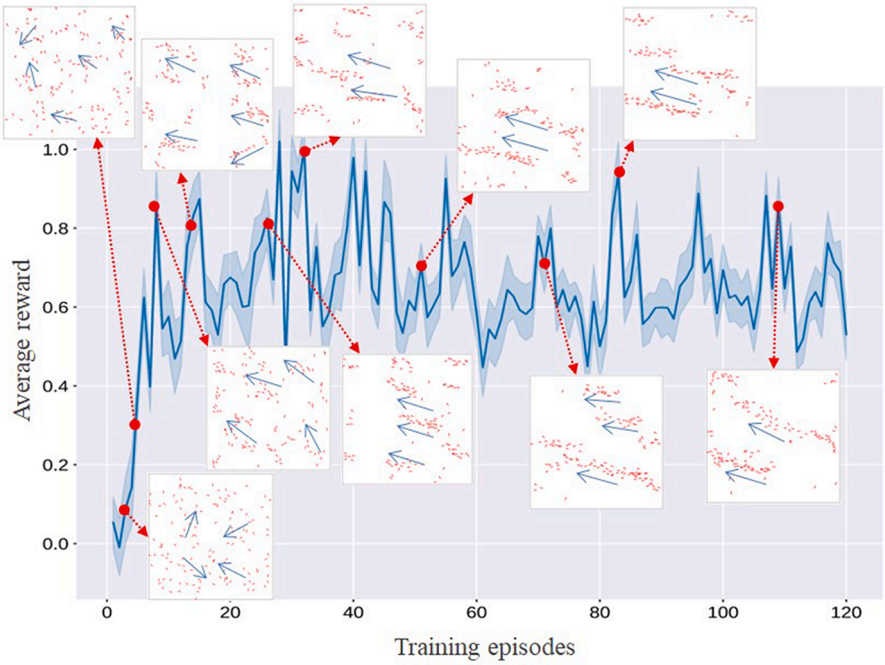


Fig. 6. The average reward of fish agents at each time step with respect to training episodes.

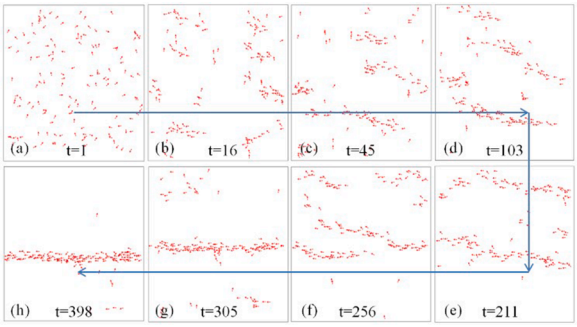


Fig. 7. The movement snapshots of fish agents during test.

3.4.2. Analysis on collective motion produced by the learned policy

This section verifies whether the collective motion generated by the learned policy is consistent with the collective motion characteristics of natural fish schools. Since three indicators (the order parameter, the average number of neighbors, and the average nearest neighbor ratio) can measure the key features of collective motion, here we use them to analyze the collective motion produced by the learned policy. Specifically, we conducted an experiment in which 140 fish agents were created and equipped with the learned policy. At each time step, every fish agent obtains information about its surrounding environment and receives an action (e.g. a change in the fish agent's turning angle), according to the learned policy. During the experiment, we monitor the order parameter, the average number of neighbors, and the average nearest neighbor ratio, shown as Fig. 9.

We can see the order parameter of the fish agent school gradually increases as the number of time steps grows in Fig. 9(a). At the beginning of the test, the fish agents are randomly initialized in the environment, which leads to order parameters of only 0.25. After about 20 steps, the order parameters stabilize above 0.9 indicating that the fish learn to select actions that follow their neighbors and form a highly ordered school. The evolution of the average number of neighbors throughout the test is given in Fig. 9(b). The learned policy is such that the average number of neighbors increases until the school stabilizes. This implies that a cohesive school can emerge as a consequence of the interactions between the fish agents using the learned policy. As shown in Fig. 9(c), the average nearest neighbor ratio gradually decreases with respect to the number of time steps, from about 0.97 to around 0.62. This reflects that the fish agent school formed by the fish agent through learned policy not only exhibits alignment and cohesion but also has a higher concentration of neighbors. In terms of the three indicators, this verifies that the collective motion derived from the learned policy is similar to the collective motion patterns observed in real fish.

3.5. Learned policy can be deployed in groups of different sizes

To evaluate how our method changes with group size, we repeated our methods with 20, 50, 80, 110, 140, and 170 fish agents. The order parameters, average number of neighbors, and average nearest neighbor ratio of different groups are shown in Fig. 10. We can see that for all group sizes, the order parameter increases as time progresses, and eventually plateaus at a high degree of order close to a value of 0.95. Similarly, for all group sizes, the average number of neighbors increases as the time step grows, and the average nearest-neighbor ratio decrease as the time step grows. This demonstrates that the learned policy can produce collective motion for different group sizes.

In Fig. 10(a), we can see that smaller groups have a lower average order parameter but higher variance. For example, the order parameter of groups with 20 agents fluctuates around 0.83, and has a very high variance. Smaller groups have higher average nearest-neighbor ratio and variance, which may be because fish agents rarely encounter each other when the group size is small leading to unstable group configurations. For larger groups with $n = 110$, 140, and 170, the order parameters rise quickly and finally stabilize at about 0.9 with small variances. Because the average nearest-neighbor ratio decreased quickly, stabilizing at about 0.62, this indicates that larger groups more easily achieve collective motion and stay in stable group configurations.

3.6. Agents acquire different collective motion patterns

As we all know, the collective pattern of fish agent schools is related to the collective motion policy of fish agents. One of the most common collective patterns is called highly parallel, and can easily be generated

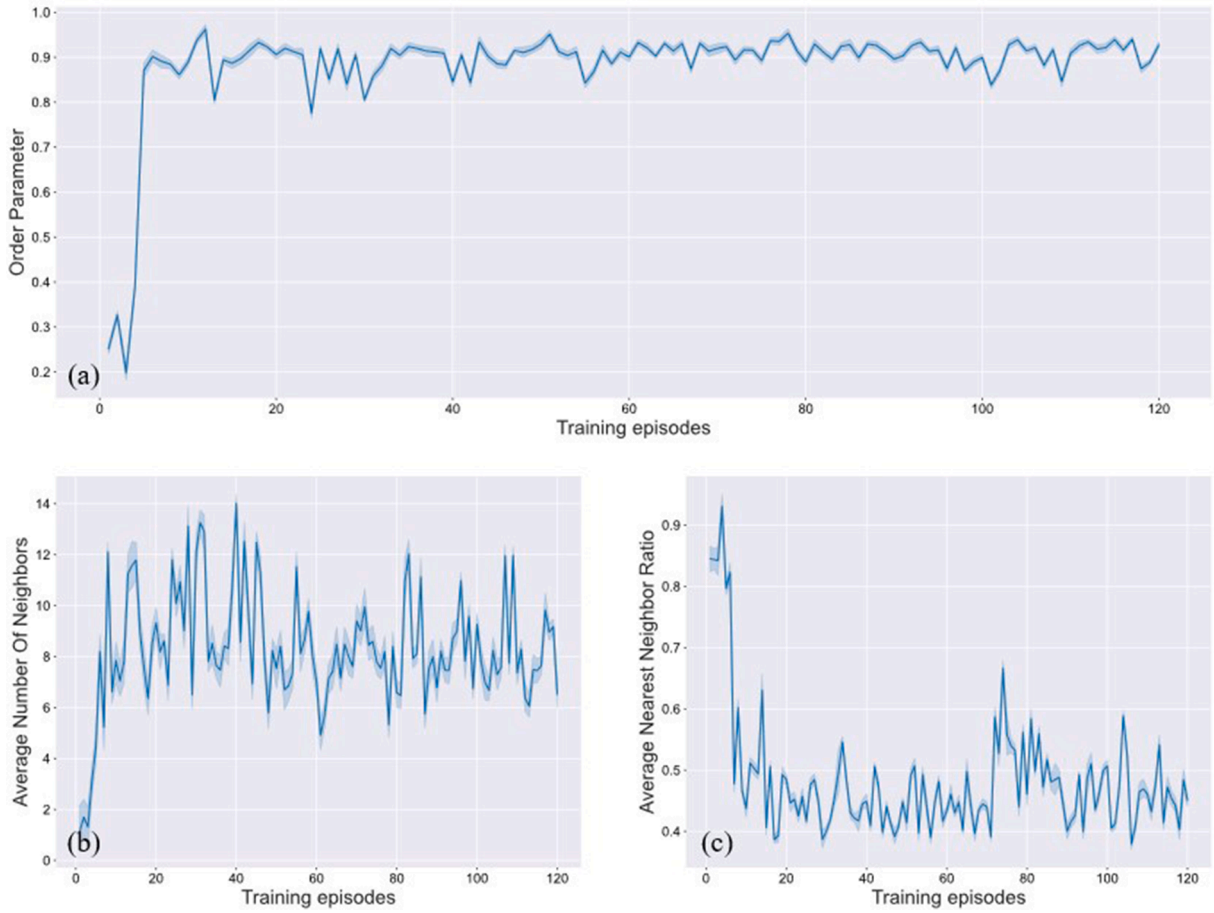


Fig. 8. The properties of collective motion in the training.

by the learned policy. However, we also observed two other collective patterns during the training process. As shown in Fig. 11, in addition to the highly parallel pattern mentioned above, we also observed rotating full-core milling and rotating hollow-core milling patterns which can also be observed in real fish schools. These patterns typically occur when fish schools are foraging, avoiding predators, etc., which may be an evolutionary behavior of animals in nature. The two collective patterns of rotating full-core milling and rotating hollow-core milling are acquired at episode 64 and 84 in our training, respectively. This phenomenon was spontaneously induced by the collective motion of the fish agent school without any additional intervention. This indicates that our proposed method has the potential to acquire policies generating various patterns of fish collective motion observed in real life.

4. Discussion

In this study, we have introduced a MARL-based approach to modeling the collective motion of fish schools and demonstrate its effectiveness.

First, the choice of the Markov decision process (MDP) used to describe collective motion is not trivial. While there exists several kinds of MDPs for multi-agent settings, we propose a new multi-agent Markov decision process (MDP) inspired by SwarMDPs (Huttenrauch et al., 2017). Both are represented with an individual model and a swarm model (the agent in the system is homogeneous), which have a more compact system representation than distributed and centralized MDP. While the two models are similar, they differ in that the SwarMDPs 1) do not have an individual-level reward function and 2) only provide a transition density function for the global state system. Our proposed MDP provides individual-level and swarm-level reward functions. The individual-level reward function can be used to describe the private or local benefits for an agent; while the swarm-level reward function provides us an option to describe the swarm's task in a bird's eye view. Furthermore, our proposed MDP provides individual-level and swarm-level state transition models. The individual-level state transition model describes the agent's local state transition without regard to other agents or environmental constraints, while the swarm-level state transition model describes the state transition of the whole swarm system, which may include agents' interactions, constraints, or other factors (i.e. noise).

Since the policy of fish agents essentially maps the agent's observation into a probability distribution over an agent's potential actions, the agent's observation representation is a key component in obtaining the policy by learning. In this paper, the observation of the fish agents is represented with a multi-channel image, where the image channels can contain different features (agents' position and orientation) based on a local view of an agent. But most existing studies describe an agent's observation with high-level variables, such as the angle between its current direction and the average direction of its neighbors, or the poses of the k nearest neighbors. While this makes policy mapping simpler and easier to learn, it is also unrealistic. In contrast, our image-like observation for one fish agent is closer to the sensory perception of real fish but is more difficult to implement particularly as the number of fish agents increase. Fortunately, our proposed MFQ-based method can easily overcome this and our results show that the fish agents can still learn a policy that produces collective motion.

In a RL setting, agents not only interact with the environment but also with each other. Thus, large number of agents worsen the

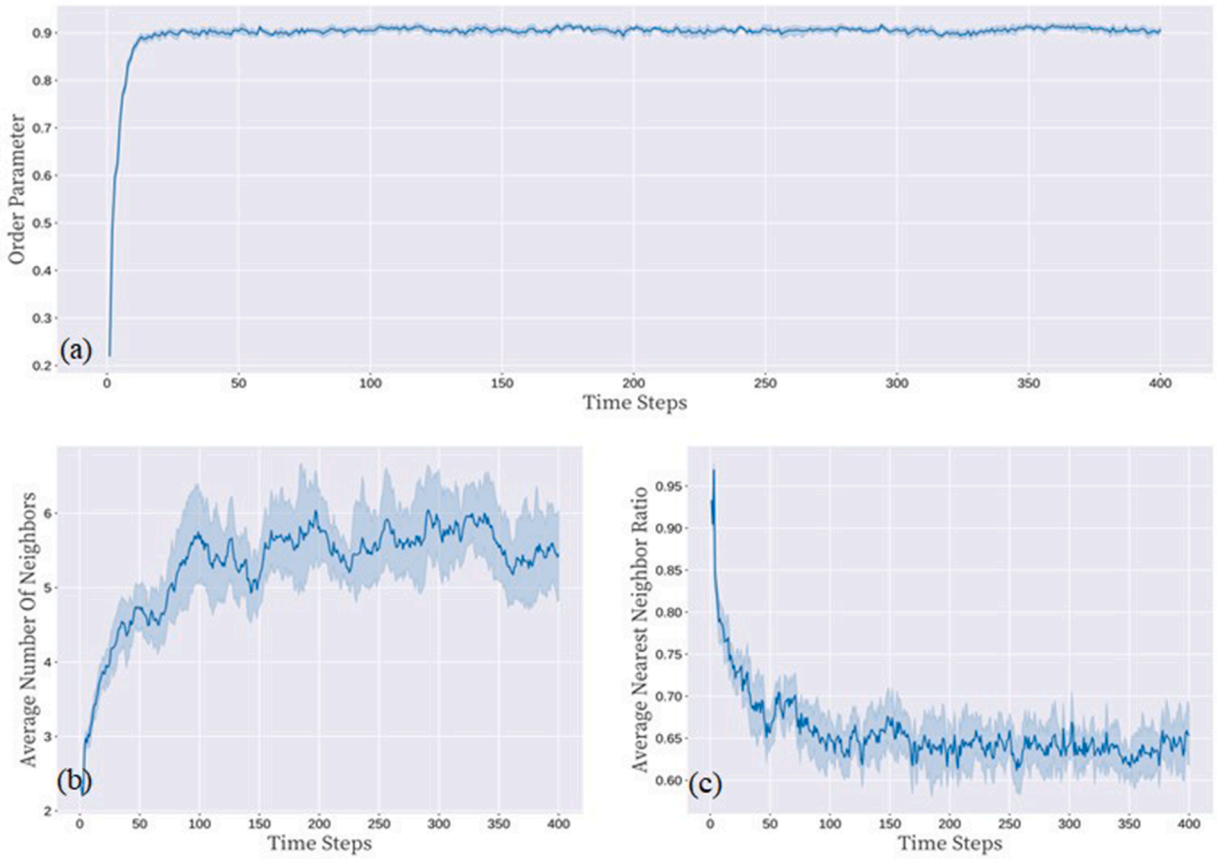


Fig. 9. The properties of collective motion produced by the learned policy.

computational complexity of learning resulting in models that are unstable and fail to converge (Matignon et al., 2012). In this paper, we use MFQ to train collective motion models, so that the interactions within the population of fish agents are approximated by those between the focal fish agent and the average effect from the neighboring agents. As the MFQ iteratively learns each agent's best response to the mean effect from its neighbors, effectively transforming the many-entity problem into a two-entity problem. Thus, with MFQ the computational complexity of learning becomes independent of the number of agents. It should be noted that we approximate an inherently continuous time process of fish schooling movement into discrete time intervals for ease of computation. At each time step, each individual fish perceives the spatial location (orientation and position) of its neighbors (it's observation), determines its action, and performs it assuming the action does not result in a with another individual. To simplify the mathematical treatment, we assume that all fish agents universally undergo synchronous updating meaning that all fish agents perceive their spatial information simultaneously and undertake their actions at the same time.

Examining the three different collective patterns founding during the training process, we found that the highly parallel pattern is more likely to appear when the number of fish agents is small, and the rotating full-core milling pattern and the rotating hollow-core milling pattern are more likely to appear when the number of fish agents is large. Therefore, we presume that the rotating full-core milling pattern and the rotating hollow-core milling pattern in fish agent schools may be evolutionary behaviors to reduce collisions with their neighbors. Fish schools could form the rotating hollow-core milling pattern when encountering predators in the real life, and could form the rotating full-core milling pattern when foraging in the nature. Furthermore, it is noteworthy that the policies that produced the three collective patterns are obtained with the same neural network in the same environment but in different training phases which indicates that our proposed method has the potential to acquire the policies generating various patterns of fish schooling observed in nature. However, due to the nature of neural network modeling, we only obtain a policy at the end of training that consistently produces the collective pattern of highly parallel. We suspect that the other two policies are lost due to the phenomenon known as "catastrophic forgetting" of neural networks (French, 1999). Catastrophic forgetting is when a connectionist networks forget all previously learned information in the process of learning new information. Neural networks are particularly prone to catastrophic forgetting, a known weakness of such models as they are unable to learn multiple tasks sequentially. If we had a mechanism to record the policies of all collective patterns utilized in the course of training, then the learned policy needed to generate the different collective patterns would be obtained.

In summary, reinforcement learning (RL) has made significant progress in recent years, and it has achieved great success in solving various sequential decision-making problems (Mnih et al., 2015; Silver et al., 2016). Multi-Agent Reinforcement Learning (MARL) is the integration of multi-agent systems and reinforcement learning that only enables the agent to learn by itself by interacting with the environment in a "trial and error" way but also solves the problems of convergence and the curse of dimensionality. Swarming animals can be viewed as a multi-agent system, and using MARL allows individuals to be modeled as agents with behavioral learning processes realistic to real animals without the need for pre-specified interaction rules. So, it is a natural way to use MARL to study the swarming behavior of such a system and also a behavioral learning simulation approach that is closer to real animals. Furthermore, it is a promising way to allow the agent to autonomously discover previously unknown rules. Thus, multi-agent reinforcement learning can be a potentially powerful tool to model,

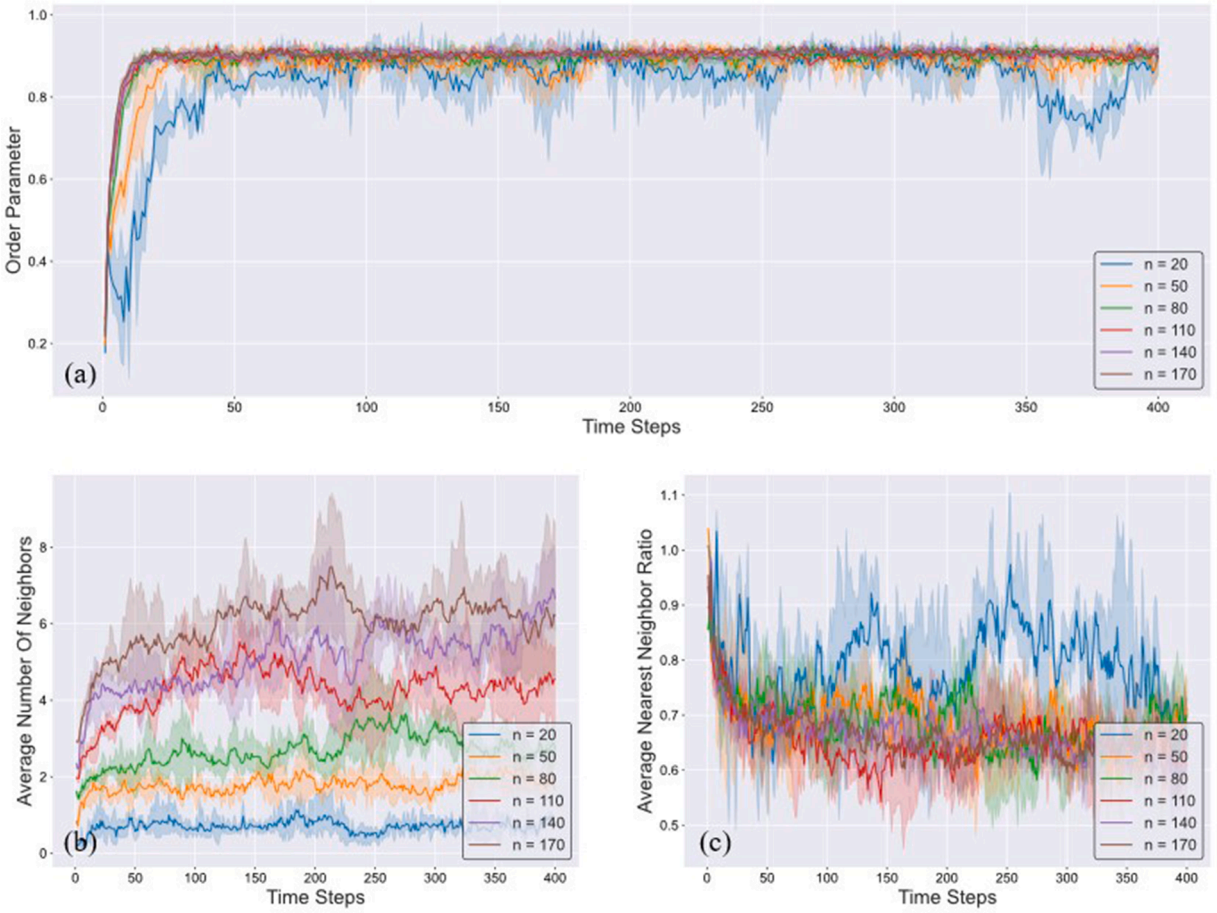


Fig. 10. The properties of collective motion for various groups in the test.

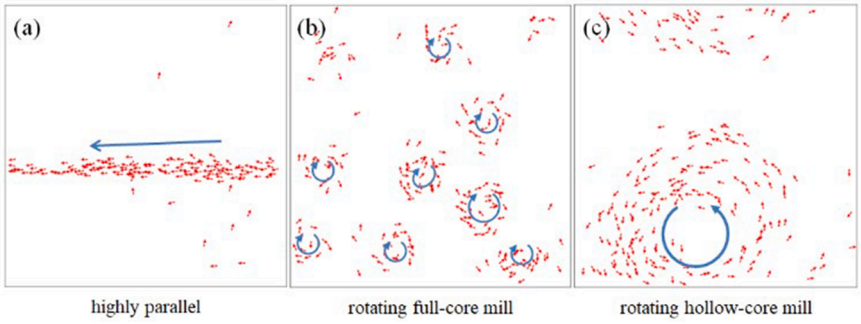


Fig. 11. Three collective motion patterns emerged during training.

analyze and understand the behavior of complicated fish schooling.

5. Conclusion

In this paper, we developed an MFQ-based method to model collective motion for fish schooling. Specifically, we modeled a fish as a learning agent, whose policy was employed as an effective "brain" to each agent in the collective motion model. The policy can be obtained via the MFQ algorithm even though the observation was represented with a multi-channel image (containing more low-level details) and the reward function was designed only with the number of neighbors and consecutive collisions between individuals. The results showed that the learned policy can produce collective motion even in groups of different sizes. The properties of collective motion that emerged from the learned policy were similar to those of classical collective motion. To the best of our knowledge, there are no previous studies that model the collective motion of fish schools using multi-agent reinforcement learning algorithms. Improving upon classic models, our MARL-based model does not directly provide or impose interaction rules between individuals in advance, but allows them to emerge as the result of learning in a given task environment. Thus, such a MARL-based model allows us to directly test whether the benefits are a possible causal explanation for the observed behavior.

Interestingly, three different collective motion patterns which can be seen in nature were observed during the training process. However, the concrete reasons for these phenomena are still unclear as the interpretability of neural networks, considered a black-box approach, is not straightforward and is a hot topic and an active area of research in the

field of machine learning. Therefore, the mechanistic reasons behind the emergence of these three collective patterns is the subject of future research.

CRediT authorship contribution statement

Xin Wang: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Shuo Liu:** Conceptualization, Software, Investigation, Writing – review & editing, Visualization. **Yifan Yu:** Conceptualization, Writing – review & editing, Investigation. **Shengzhi Yue:** Conceptualization, Investigation, Writing – review & editing. **Ying Liu:** Conceptualization, Supervision, Writing – review & editing, Resources, Supervision. **Fumin Zhang:** Conceptualization, Funding acquisition, Resources, Writing – review & editing, Supervision. **Yuanshan Lin:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This study was partly supported by Liaoning Province Natural Science Foundation (2020-KF-12-09), Foundation of Liaoning Educational Committee (LJKZ0730, QL202016), Open Foundation of Key Laboratory of Environment Controlled Aquaculture (Dalian Ocean University) Ministry of Education (202219), Liaoning key research and development program (2020JH2/10100043).

References

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., Józefowicz, R., Gray, S., Olsson, C., Pachocki, J., Petro v, M., Pinto, H.P.d.O., Raiman, J., Salimans, T., Schlatt er, J., Schneider, J., Sidor, S., Sutskever, I., Tang, J., Wols ki, F., Zhang, S., 2019. Dota 2 with large scale deep reinforcement learning. https://arxiv.org/abs/191.2.06680.

Bode, N.W.F., Frank, D.W., Wood, A.J., 2010. Making noise: emergent stochasticity in collective motion. J. Theor. Biol. 267, 292–299. https://doi.org/10.1016/j.jtbi.2010.08.034.

Brown, N., Sandholm, T., 2018. Superhuman AI for heads-up no-limit poker: libratus beats top professionals. Science 359, 418–424. https://doi.org/10.1126/science.aao.1733.

Clark, P.J., Evans, F.C., 1954. Distance to nearest neighbor as a measure of spatial relationships in populations. Ecology 35, 445–453.

Collignon, B., Séguret, A., Halloy, J.A., 2016. A stochastic vision-based model inspired by zebrafish collective behaviour in heterogeneous environments. R. Soc. Open Sci. 3, 150473 https://doi.org/10.1098/rsos.150473.

Costa, T., Laan, A., Heras, F.J., de Polavieja, G.G., 2020. Automated discovery of local rules for desired collec- tive-level behavior through reinforcement learning. Front. Phys. 8, 1. https://doi.org/10.3389/fphy.2020.00200.

Couzin, I.D., Krause, J., James, R., Ruxton, G.D., Franks, N.R., 2002. Collective memory and spatial sorting in animal groups. J. Theor. Biol. 218, 1. https://doi.org/10.1006/jtbi.2002.3065.

De Souza, C., Newbury, R., Cosgun, A., Castillo, P., Vidolov, B., Kulić, D., 2021. Decentralized multi-agent pursuit using deep reinforcement learning. IEEE Robot. Autom. Lett. 6, 4552–4559. https://doi.org/10.1109/LRA.2021.3068952.

Deutsch, A., Theraulaz, G., Vicsek, T., 2012. Collective motion in biological systems. Interface Focus 2, 689. https://doi.org/10.1098/rsfs.2012.0048.

Durve, M., Peruani, F., Celani, A., 2020. Learning to flock through reinforcement. Phys. Rev. E. 102, 012601 https://doi.org/10.1103/PhysRevE.102.012601.

French, R.M., 1999. Catastrophic forgetting in connectionist networks. Trends Cognit. Sci. 3, 128–135. https://doi.org/10.1016/S1364-6613(99)01294-2.

Gautrais, J., Ginelli, F., Fournier, R., Blanco, S., Soria, M., Chaté, H., Theraulaz, G., 2012. Deciphering interactions in moving animal groups. PLoS Comput. Biol. 8, e1002678 https://doi.org/10.1371/journal.pcbi.1.002678.

Ginelli, F., Chaté, H., 2010. Relevance of metric-free interactions in flocking phenomena. Phys. Rev. Lett. 105, 168103 https://doi.org/10.1103/PhysRevLett.105.1681.03.

Hahn, C., Phan, T., Gabor, T., Belzner, L., Linnhoff-Popien, C., 2019. Emergent escape-based flocking behavior using multi-agent reinforcement learning. https://arxiv.org/abs/1905.04077.

Hemelrijk, C.K., Hildenbrandt, H., 2012. Schools of fish and flocks of birds: their shape and internal structure by self-organization. Interface Focus 2, 726. https://doi.org/10.1098/rsfs.2012.0025.

Hemelrijk, C.K., Kunz, H., 2005. Density distribution and size sorting in fish schools: an individual-based model. Behav. Ecol. 16, 178–187. https://doi.org/10.1093/beheco/arh149.

Herbert-Read, J., Perna, A., Mann, R., Schaerf, T., Sumpter, D., Ward, A., 2011. Inferring the rules of interaction of shoaling fish. Proc. Natl. Acad. Sci. USA 108, 18726–18731. https://doi.org/10.1073/pnas.11093.55108.

Hinz, R.C., De Polavieja, G.G., 2017. Ontogeny of collective behavior reveals a simple attraction rule. Proc. Natl. Acad. Sci. USA 114, 2295–2300.

Hüttenrauch, M., šošić, A., Neumann, G., 2017. Guided deep reinforcement learning for swarm systems. https://doi.org/10.48550/arXiv.1709.06011.

Jhawar, J., Guttal, V., 2020. Noise-induced effects in collective dynamics and inferring local interactions from data. Philos. Trans. R. Soc. Lond., B, Biol. Sci. 375, 20190381 https://doi.org/10.1098/rstb.2019.0381.

Jhawar, J., Morris, R.G., Amith-Kumar, U.R., Raj, M.D., Rogers, T., Rajendran, H., Guttal, V., 2020. Noise-ind-uced schooling of fish. Nat. Phys. 16, 488–493. https://doi.org/10.1038/s41567-020-0787-y.

Katz, Y., Tunstrøm, K., Ioannou, C., Huepe, C., Couzin, I., 2011. Inferring the structure and dynamics of interactions in schooling fish. Proc. Natl. Acad. Sci. USA 108, 18720–18725. https://doi.org/10.1073/pnas.110758.3108.

Kolpas, A., Moehlis, J., Kevrekidis, I.G., 2007. Coarse-g0rained analysis of stochasticity-induced switching between collective motion states. Proc. Natl. Acad. Sci. (USA) 104, 5931–5935. https://doi.org/10.1073/pnas.0608 270104.

López-Incera, A., Ried, K., Müller, T., Briegel, H.J., 2020. Development of swarm behavior in artificial learning agents that adapt to different foraging environments. PLoS ONE 15, e0243628. https://doi.org/10.1371/journal.pone.0243628.

Lukeman, R., Li, Y.X., Edelstein-Keshet, L., 2010. Inferring individual rules from collective behavior. Proc. Natl. Acad. Sci. USA 107, 12576–12580. https://doi.org/10.1073/pnas.100176.3107.

Matignon, L., Laurent, G.J., Le Fort-Piat, N., 2012. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. Knowl. Eng. Rev. 27, 1–31. https://doi.org/10.1017/S0269.8889120.00057.

McComb, D.M., Kajiura, S.M., 2008. Visual fields of four batoid fishes: a comparative study. J. Exp. Biol. 211, 482–490. https://doi.org/10.1242/jeb.014506.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Venes s, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjelan d, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Leg g, D., Hassabis, D., 2015. Human-level control through dee p reinforcement learning. Nature 518, 529–533. https://doi.org/10.1038/nature14236.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., Bowling, M., 2017. DeepStack: expert-level artificial intelligence in heads-up no-limit poker. Science 356, 508–513. https://doi.org/10.1126/science.aam6960.

Morihiro, K., Nishimura, H., Isokawa, T., Matsui, N., 2008. Learning grouping and anti-predator behaviors for multi-agent systems. In: Proceedings of the International Conference on Knowledge-Based and Intelligent Infor mation and Engineering Systems. Springer, pp. 426–433. https://doi.org/10.1007/978-3-540-85565-1_53.

Mwaffo, V., Anderson, R.P., Porfiri, M., 2015. Collective dynamics in the vicsek and vectorial network models beyond uniform additive noise. J. Nonlinear Sci. 25, 1053–1076. https://doi.org/10.1007/s00332-015-926.0-y.

Reynolds, C.W., 1987. Flocks, herds and schools: a distributed behavioral model. SIGGRAPH Comput. Graph. 21, 25–34. https://doi.org/10.1145/37402.37406.

Ried, K., Müller, T., Briegel, H.J., 2019. Modelling collective motion based on the principle of agency: general framework and the case of marching locusts. PLoS ONE 14, e0212044. https://doi.org/10.1371/journal.pone.0212044.

Schaerf, T.M., Dillingham, P.W., Ward, A.J.W., 2017. The effects of external cues on individual and collective behavior of shoaling fish. Sci. Adv. 3, 1–16. https://doi.org/10.1126/sciadv.1603201.

Shaebani, M.R., Wysocki, A., Winkler, R.G., Gompper, G., Rieger, H., 2020. Computational models for active matter. Nat. Rev. Phys. 2, 181–199. https://doi.org/10.1038/s42254-020-0152-1.

Shimada, K., Bentley, P., 2018. Learning how to flock: deriving individual behaviour from collective behaviour with multi-agent reinforcement learning and natural evolution strategies. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. ACM, pp. 169–170. https://doi.org/10.1145/3205651.3205770.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Pa nneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., N ham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leac h, M., Kavukcuoglu, K., Graepel, T., Hassabis, D., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489. https://doi.org/10.1038/nature16961.

Sunehag, P., Lever, G., Liu, S., Merel, J., Heess, N., Leibo, J.Z., Hughes, E., Eccles, T., Graepel, T., 2019. Reinfor cement learning agents acquire flocking and symbiotic behaviour in simulated ecosystems. In: Proceedings of the 2019 Conference on

Artificial Life: How Can Artificial Life Help Solve Societal Challenges. MIT, pp. 103–110. https://doi.org/10.1162/isal_a_00148.
Vicsek, T., Czirok, A., Ben-Jacob, E., Cohen, I., Shochet, O., 1995. Novel type of phase transition in a system of self-driven particles. Phys. Rev. Lett. 75, 1226. https://doi.org/10.1103/PhysRevLett.75.1226.
Vicsek, T., Zafeiris, A., 2012. Collective motion. Phys. Rep. 517, 71–140. https://doi.org/10.1016/j.physrep.2012.03.004.
Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J.P., Jaderberg, M., Vezhnevets, A.S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T.L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., Silver, D., 2019. Grandmaster level in starcraft II using multi-agent reinforcement learning. Nature 575, 1–5. https://doi.org/10.1038/s41586-019-1724-z.
Wang, X., Cheng, J., Wang, L., 2020. A reinforcement learning-based predator-prey model. Ecol. Complex. 42, 100815 https://doi.org/10.1016/j.ecocom.2020.100815.
Wright, C.M., Lichtenstein, J.L., Doering, G.N., Pretorius, J., Meunier, J., Pruitt, J.N., 2019. Collective personalities: present knowledge and new frontiers. Behav. Ecol. Sociobiol. 73, 1. https://doi.org/10.1007/s00265-019-2639-2.
Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J., 2018. Mean field multi-agent reinforcement learning. In: Proceedings of the 35th International Conference on Machi ne Learning. PMLR, pp. 5571–5580.
Yates, C.A., Erban, R., Escudero, C., Couzin, I.D., Buhl, J., Kevrekidis, I.G., Maini, P.K., Sumpter, D.J.T., 2009. Inherent noise can facilitate coherence in collective swarm motion. Proc. Natl. Acad. Sci. U.S.A. 106, 5464–5469. https://doi.org/10.1073/pnas.0811195106.