Adaptive Bio-Inspired Fish Simulation with Deep Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Our goal is to synthesize realistic underwater scenes with various fish species in different fish cages, which can be utilized to train computer vision models to automate fish counting and sizing tasks. It is a challenging problem to prepare a sufficiently diverse labeled dataset of images from aquatic environments. We solve this challenge by introducing an adaptive bio-inspired fish simulation. The behavior of caged fish changes based on the species, size and number of fish, and the size and shape of the cage, among other variables. In this paper, we propose a method for achieving schooling behavior for any given combination of variables, using multi-agent deep reinforcement learning (DRL) in various fish cages in arbitrary environments. Furthermore, to visually reproduce the underwater scene in different locations and seasons, we incorporate a physically-based underwater simulation.

1 Introduction

The sustainability of the food supply and the protection of marine resources are two of the most important global issues, and are part of the agenda set by the United Nations to be achieved by 2030 [1]. Fish farming is one of the keys to a sustainable seafood supply to support the global population, as seafood consumption is growing rapidly. One of the current major problems in the fish farming industry is managing feeding. Under-feeding slows down the growth of the fish due to malnutrition. Over-feeding is even worse as it can kill the fish and the residual food contaminates the marine environment. Therefore, we attempt to automate fish counting and sizing using inexpensive commercially available RGB cameras and computer vision to optimize the amount and timing of feeding.

Deep learning has achieved great success in the field of computer vision. The quality of the dataset determines the accuracy of deep learning models, but it is difficult to obtain data to train a fish counting model. Ishiwaka et al. [1] proposed a fish schooling simulation called Foids, and demonstrated the efficacy of using a CG synthetic dataset for training. Foids adds rules based on fish biology to the Boids [2] algorithm. However, both Boids and Foids require manually setting a number of parameters for a given fish species, which must be adjusted again should any conditions change. To solve this problem, we propose a method to autonomously generate schooling behavior in fish using multi-agent deep reinforcement learning. Fish behavior in a fish farming cage depends not only on natural factors like temperature and light intensity, but also the size and shape of the fish cage, and even the species, number, and size of the fish themselves. By taking into account biological data such as the preferred light intensity, temperature, and inter-fish distance of each fish, our method achieved several distinct patterns of collective behavior depending on population density through autonomous learning. Furthermore, we developed a physically-based underwater environment simulation. This

[1] https://sdgs.un.org/2030agenda

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

simulation is capable of accurately reproducing the conditions of underwater scenes of arbitrary locations and seasons. The bio-inspired fish simulation and physically-based environment simulation allow for the creation of a high-quality synthetic dataset, with which we successfully trained a deep learning model to count fish of various species in fish cages.

In this paper, we explore related work in Sec. 2. We then describe the implementation details of the simulation, data synthesis, and the computer vision models in Sec. 3 and 4. In Sec. 5, we show the experimental results and analysis followed by a conclusion in Sec. 6.

2 Related Work

It is well known that mammals use reinforcement learning [3]. Experiments in neuroscience have demonstrated that dopamine neurons play a role in rewards in reinforcement learning [4, 5, 6]. These dopamine neurons are present not only in mammals but in flies and fish as well [7]. It has been shown that dopamine neurons are used in cooperative behavior [8] and learning [9] in fish. Therefore, we adopted reinforcement learning for training cooperative schooling behavior.

In the field of fish swimming simulations, DRL has been used to produce efficient swimming and verify adaptation behavior [10, 11, 12, 13]. These studies focused on simulations of a few fish or soft-bodied animals, but they did not cover large numbers of fish or differences between species. Tu and Terzopoulos [14] proposed a fish simulation with perception, cognition and muscle based locomotion. Lindsey [15] defined 12 types of locomotion in fish, and Satoi et al.[16] incorporated this definition and proposed a controller which automatically selects a type of locomotion based on a desired target behavior. Although those methods worked well at simulating individual fish behaviors, they were not directly useful for our objective of simulating tens of thousands of fish, while forming schooling behavior organically through interactions among the fish.

DRL-based approaches have also been studied for flocking control tasks [17, 18, 19] and crowd simulation [20, 21]. These models do not take biological specifics and their influences on flocking behaviors of simulated objects into consideration, while the adaptation of these species-dependent characteristics constitute a crucial part of our fish control system.

There are relatively few publicly available computer vision datasets of underwater images. Three major such datasets are OzFish [22], Deepfish [23], and a dataset from Ditria et al. [24]. These datasets lack the species and environments seen in fish farms, making them unsuitable for our goals.

The biological properties we use in our simulation are aggregation, cohesion and separation rules[25, 26];fish's preferred temperature and light intensity [27, 28, 29]; personal space [30, 31]; sense of depth due to hydrostatic pressure [32, 33]; response to obstacles [34, 35]; and decision making interval [36], as well as the social ranking of fish. Fish often form dominance hierarchies, where the dominant fish display aggressive behavior by chasing the subordinate individuals from time to time [37, 38]. The details of the biological background are described in the Appendix. In this paper, we set rewards and penalties for DRL based on these properties.

3 Fish Simulation

In this section, we introduce a bio-inspired fish simulation trained with DRL. The bold letters $\mathbf{v}$, $\mathbf{d}$ and $\mathbf{u}$ are used to denote directional 3D vectors, and the uppercase calligraphic letter $\mathcal{I}$ denotes a higher-order tensor. We structure the behavioral model of fish as a multi-agent reinforcement learning problem, where fish agents need to navigate around a constrained 3D space that resembles a fish farm environment while adapting to multiple factors that simultaneously influence their behavior. During the course of interacting with the environment and other fish, each agent acts based on its own observations and is responsible for learning a general policy that maximizes a reward. We model the policies using neural networks and optimize them using Proximal Policy Optimization (PPO) [39] with Generalized Advantage Estimation (GAE) [40].

A schematic illustration of our fish simulation pipeline is provided in Fig.1. The parameters controlling biological and environmental factors in the fish simulation are set based on the literature presented in the Appendix and field data collected from fish farms at Onmaehama and Nishiki, Japan. Fish animation is generated by the finite state machine based on the velocity of the locomotion and the fish's behavior state, which is explained in more details in the Appendix.

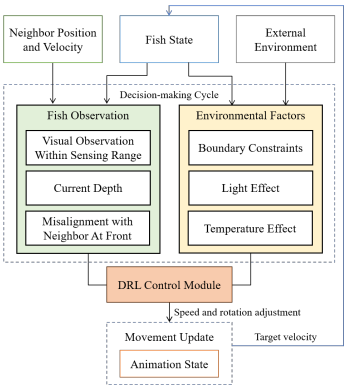


Figure 1: Simulation framework.

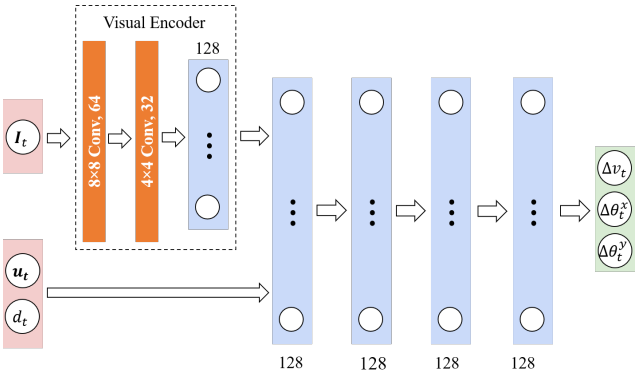


Figure 2: Architecture of policy network.

Biological and Environmental Factors: We incorporate important biological and environmental drivers of fish swimming patterns into the simulation framework. A fish stays within a comfortable distance of its neighboring fish and aligns its direction with that of its nearest neighbor in the forward direction [2, 41]. Its activity space is constrained by the cage boundaries and water surface, which reproduce the commercial cage environment and facilitate the formation of schooling. Additionally, caged fish are divided into dominant and subordinate groups, where the dominant individuals may initiate antagonistic acts by aggressively approaching the subordinate members. These factors are carefully integrated into the process of learning control policies that generate the delta velocity of fish $\Delta\mathbf{v}_t^f$ at each time step t through the use of the PPO algorithm.

There are effects of underwater light intensity and temperature on the fish's vertical distribution, which we represent as $\Delta\mathbf{v}_{\text{light}}$ and $\Delta\mathbf{v}_{\text{temp}}$. A fish has a preferred range of light intensity and water temperature, and it changes its vertical position to stay in the comfortable zone. The details of the computation are explained in the Appendix.

In addition to the above-mentioned components, we integrate the fish decision making interval into the framework to emulate the latency of fish's responses to environmental changes. The duration (in units of simulation steps) of the decision making interval, Δt_{res}, for simulated species is predefined in accordance with the literature studies described in Sec. 2. At the time of simulation, given the time interval of simulation steps (Δt_{sim}), a fish updates its observation of the environment every $\lfloor \Delta t_{\text{res}}/\Delta t_{\text{sim}} \rfloor$ steps and takes actions between updates. The delta accumulated velocity to apply at each simulation step ($\Delta\mathbf{v}_t^a$) can then be derived as follows:

$$\Delta\mathbf{v}_t^a = \Delta\mathbf{v}_t^f + \frac{\Delta\mathbf{v}_{\text{light}} + \Delta\mathbf{v}_{\text{temp}}}{\lfloor \Delta t_{\text{res}}/\Delta t_{\text{sim}} \rfloor}. \tag{1}$$

States and Actions: Each state $s_t \in$ state space $\mathcal{S}$ encodes the information a fish agent observes in the environment at time step t. It can be represented by a tuple $(\mathbf{u}_t, d_t, \mathcal{I}_t)$, where $\mathbf{u}_t$ is the difference between forward directions of the agent and its nearest neighboring fish in front of it at current step t, d_t is the depth of the agent with respect to the water surface, and $\mathcal{I}_t$ is a visual observation tensor. During the time of exploration, a fish agent collects visual observations with a spatial grid sensor that imitates the sensing area of real fish [42]. The visual observation is stored as a third-order tensor, whose dimensions are grid width, grid height, and number of channels. The channels contain the distance information and multi-hot values of detectable objects (e.g. other agents and cage walls) within the fish's sensing range d_{sense}. d_{sense} is valued at 2 body lengths (BL) for yellowtail amberjack (yellowtail) [43] and 3 BL for coho salmon and red seabream [42]. We stack three visual observations together to infer movement before passing them to the networks. All the state components are computed in the local coordinate system of the agent, with the origin located at the body center and the z-axis parallel to the fish's facing direction. Note that a simulated fish is not capable of precisely observing its speed and rotation since a real fish can only sense its relative rotation through the use of the lateral line system [44, 45].

The action $a \in$ action space $\mathcal{A}$ determined by the policy specifies the delta speed (Δv_t), as well as delta rotation about the x-axis ($\Delta\theta_t^x$) and the y-axis ($\Delta\theta_t^y$) in degrees. The rotation angle about the z-axis is clamped by a small angle θ^{zt} to avoid unnatural rolling behavior. Δv_t is also clamped by

the maximum delta speed Δv_{max} allowed in the cage environment. The three action components are then used to compute the delta velocity of fish $\Delta \mathbf{v}_t^f$ at the current time step and thus drive the motion of the fish agent as depicted in Eq. (1).

Reward: The reward r_t at each time step is defined to encourage schooling behavior while avoiding boundary collisions and to be consistent with the biological studies described in Sec. 2:

$$r_t = r_t^{BC} + r_t^{NC} + r_t^{BD} + r_t^{ND} + r_t^{E} + r_t^{M} + r_t^{C}. \tag{2}$$

The reward r_t^{BC} represents the penalty of colliding with the spatial boundaries, which include the cage walls and water surface. It has a fixed value of -300 if a boundary collision occurs or 0 otherwise. r_t^{NC} penalizes the collision with neighboring fish detected by box colliders using an associated weight w^{NC} and accumulates with the number of colliding agents N_{hit}:

$$r_t^{NC} = -w^{NC}\ N_{\text{hit}}$$

The boundary avoidance reward r_t^{BD} encourages the fish to keep a distance from a detected spatial boundary. Its value depends on the agent's sensing range d_{sense}, the number of detected boundaries N_{bnd}, the distance d_{i} to the boundary i and the boundary avoidance weight w^{BD}:

$$r_t^{BD} = -w^{BD} \sum_{i=1}^{N_{\text{bnd}}} \left(\frac{d_{\text{sense}} - d_{\text{i}}}{d_{\text{sense}}} \right).$$

The neighbor interaction reward r_t^{ND} encourages the fish to stay close to its neighbors within the sensing range and to align its direction with those of its neighbors. The angle $\Delta\theta_i^{\text{mov}}$ in degrees between the directions of the agent and each of its N_{nei} neighbors is calculated and a neighbor interaction weight w^{ND} is used for the computation:

$$r_t^{ND} = w^{ND} \sum_{i=1}^{N_{\text{nei}}} \left(\frac{90 - \Delta\theta_i^{\text{mov}}}{90} \right).$$

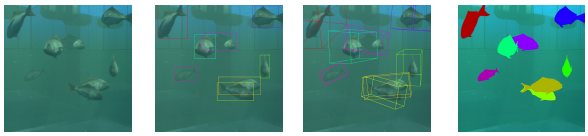
On the other hand, r_t^E penalizes energy consumption of the fish while rotating its body or adjusting speed. It is computed from a rotation penalty weight w^r, a speed penalty weight w^s together with the delta angle $\Delta\theta_t$ of the body rotation and the delta accumulated speed Δv_t^a at the current time step:

$$r_t^E = -w^r\ \Delta\theta_t - w^s\ \left|\Delta v_t^a\right|.$$

The movement reward r_t^M encourages the fish to swim faster than a minimum speed and penalizes sudden changes in depth caused by aggressive pitch motion (around the local x-axis). In the expression below, the variable θ^{rt} denotes the pitch angle threshold, v^{st} is the speed threshold, θ_t^x is the current pitch angle, and v_t^a represents the current accumulated speed. The reward r_t^M has a value of -10 when $\theta^{rt} \leq \theta_t^x \leq (360 - \theta^{rt})$, a value of 2 when $-\theta^{rt} < \theta_t^x \leq \theta^{rt}$ and $v_t^a \geq v^{st}$, and a value of 0 otherwise.

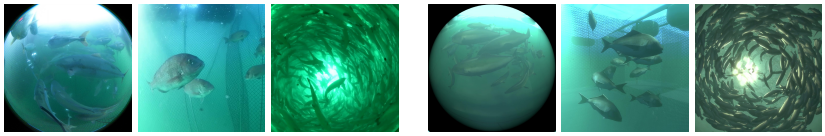
Lastly, r_t^C denotes the chase reward which encourages aggression or escape behavior based on the social rank of the fish. Specifically, a dominant fish (aggressor) randomly starts a chase mode with a small probability p_a and initiates an attack on its nearest subordinate neighbor (target). This triggers the subordinate being chased to start its chase mode and swim away from the aggressor. We reward the aggressor by a fixed large value if it collides with its target. This process can be expressed using the aggressor's accumulated velocity $\mathbf{v}_t^a$, a normalized vector $\mathbf{d}$ spanning from the aggressor to the target, a chase reward weight w^{agg} for the aggressor, and a chase penalty weight w^{tar} for the target as follows: the reward r_t^C has a value of $w^{\text{agg}}\left(\mathbf{d} \cdot \mathbf{v}_t^a\right)$ if a fish is an aggressor in chase mode and chases after its target, a value of 100 if a fish is an aggressor in chase mode and collides with its target, a value of $-w^{\text{tar}}\left(\mathbf{d} \cdot \mathbf{v}_t^a\right)$ if a fish is a target in chase mode, and a value of 0 otherwise. The chase mode ends for both fish when they collide, the target escapes to the back of the aggressor, or after a short period of time as the dominants prefer not wasting energy on a prolonged chase. A detailed explanation of parameters used in action definition and reward function is available in Appendix.

Network: The simulation of realistic fish schooling behaviors can be broken down into a series of tasks where fish agents need to choose optimal actions from continuous action space. In the Actor-Critic framework used by PPO, a policy (actor) π is modeled as a neural network (Fig. 2)



(a) Scene (b) 2D and 3D bounding boxes (c) Silhouette

Figure 3: Synthetic dataset example. (a) scene rendering. (b) 2D and 3D bounding box annotations. (c) silhouette annotation.



(a) Frames: real video (b) Frames: simulation.

Figure 4: Comparison of the scenes captured from different views at a fish farm and in the simulation. Left-to-right: yellowtail, red seabream and coho salmon. Note that a fish-eye lens is used for yellowtail while a regular lens is used for coho salmon and red seabream.

that maps a state s_t to a Gaussian distribution over action $\pi(a_t|s_t)$. The agent's visual observation $\mathcal{I}_t$, which is formed as a 3D tensor, is processed by a visual encoder comprising two convolutional layers and a dense layer before being passed to a fully-connected network with other observation components. The first and the second convolutional layers contain 16 8×8 filters and 32 4×4 filters respectively and use Leaky ReLU activation. The fully-connected layer in the visual encoder contains 128 units and also uses Leaky ReLU as the activation function. Each of the other four fully-connected layers consists of 128 units and uses the Swish activation function. The output layer is just a linear layer with the output size equal to the size of action space. We train the policies using PPO with a clipped surrogate objective [39]. An in-depth review of this algorithm can be found in the Appendix. The advantage for the policy gradient is calculated by GAE [40]. The value function (critic) $V(s_t)$ is represented as another deep neural network with similar architecture except the output layer is a single unit. The networks are built on the open source library PyTorch.

4 Synthetic Dataset and Fish Counting

Our simulation is developed on Unity Engine [2], and Crest Ocean System HDRP asset [3] is utilized to create the effect of sun and ocean waves. The simulation is controlled by parameters including, but not limited to, fish biological details such as fish species, count, size, and speed; temperature and light preferences; latitude, longitude, date and time; wave size, sediment and chlorophyll concentrations; sunlight color and intensity; and light attenuation and scattering. A major drawback of prior fish simulation systems [31, 1] is the need for manual parameter tuning when applying the simulation to new species or new environments. However, in this work we leverage the power of DRL to train the fish to adapt to changing environmental conditions given their biological data, and apply the species-specific trained models to emulate the corresponding swimming patterns. We also use the physically-based environment simulation described in the Appendix to control the scene with a small number of hyperparameters instead of extensive adjustment with programmable shaders. These improvements significantly automate the simulation process and allow us to generate an arbitrarily large and varied dataset by randomly setting the above mentioned parameters within a physically valid range. Fig. 3 displays an example of a synthesized dataset.

The fish counting system comprises three modules: an image pre-processing module, which converts the input video to a sequence of images and applies denoising; a fish detection module, which is a trained network based on YOLOv4 [46] with the synthetic dataset; and a fish counting module. The system takes as input a fish cage video, and outputs an estimate of the total number of fish shown in the video. The details of each component can be found in the Appendix.

[2] available at `https://unity3d.com/`

[3] available at `https://assetstore.unity.com/packages/tools/particles-effects/crest-ocean-system-hdrp-164158` under Standard Unity Asset Store EULA license

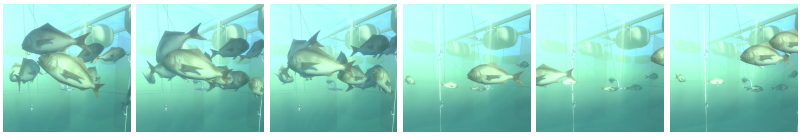
5 Experiments and Results

We describe the training process, experimental results of the fish behavioral model with diverse biological and cage settings, underwater simulation, and fish counting.

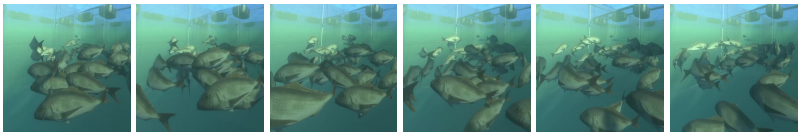
Training: We trained the behavioral models of coho salmon, yellowtail and red seabream. A two-step training scheme was employed to facilitate the adaption for agents in eleven environments with the same action space, observation space and types of detectable objects, but different fish species, fish sizes, fish numbers, cage sizes and cage shapes. We pretrained the three species in their corresponding default scenes, which contain 1000 coho salmon, 45 yellowtail and 10 red seabream, respectively, with the cage configurations set to be the same as the fish farm environments we collected field data from. The pretraining results were usually sufficient to enable the agents to navigate and avoid cage boundaries to some extent, but collisions between fish and with boundaries still occur and unnatural movements persist. We then performed transfer learning based on the pretrained policies for each species in its corresponding environment-specific experiments, including the default scene that is used to reproduce the real scene. We found the use of such a two-step training scheme led to faster convergence and overall performance improvements compared to training from scratch for each environment. We enabled early termination of an episode if the agent collides with either a cage wall or the water surface. The agent would subsequently restart a new episode from a random position within the cage with a random valid rotation and an initial speed v_0. This strategy discourages boundary collision and prevents the agents from collecting data that is irrelevant to boundary avoidance. The values of hyperparameters in the pretraining and transfer learning phases are similar if not identical. For yellowtail and red seabream, we ran a total number of two million time steps and used a linearly decaying learning rate of 0.001 in both phases. For coho salmon, we adopted the same settings, but increased the pretraining and transfer learning steps to four million and one million time steps respectively, because the 1000 coho salmon in the default scene is much more than the other two species. The total wall-clock training times for four million and five million simulation steps were around 5500 seconds and 6800 seconds, respectively. A detailed explanation of the eleven simulation environments, hyperparameter settings, and the learning curves of two training stages are available in the Appendix. We also demonstrate the training progress of yellowtail in a recorded demo that is part of our supplementary video. The training took place on an NVIDIA GeForce RTX 3080 Laptop GPU installed on a Windows 10 machine with a 3.3 GHz AMD Ryzen 9 5900HX CPU.

Fish Behavior: We simulated swimming scenes of the above-mentioned three species in order to provide a variety of synthetic data to train the computer vision system that detects and tracks fish. Fish agents were initialized at random positions in the cage with random valid rotations, and would start to form circling patterns by following the trained policies. A qualitative comparison between the image sequences from the simulation and the videos captured underwater is displayed in Fig. 4. Note that the swimming patterns of the three fish species are distinct in the captured video. Our simulation successfully reproduced such behavioral differences as well as the environmental variation such as lighting, water color and turbidity.

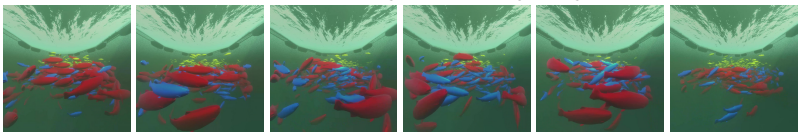
We also showcase the simulation results of five example environments with different species, fish size, fish number, cage size and shape settings in Fig. 5. A body scale of 1.0 refers to the default body lengths (BL) of coho salmon, yellowtail and red seabream, which were set to be 0.49 meters, 0.52 meters, and 0.34 meters respectively according to the collected field data. We additionally applied a mild repulsive force acting as an object-to-object colliding force to push any penetrating fish away from each other to further facilitate collision avoidance among agents. The fish swimming patterns vary with the scene configuration since they result from a combination of factors discussed in Sec 3. Notably, the 10 red seabream in Fig. 5(a: left 3) slowly circle around the cage center given the small cage size, whereas the same 10 fish in Fig. 5(a: right 3) learn to swim rapidly and away from the center since the cage size is much larger. On the other hand, the red seabream in Fig. 5(b) form a more compact school and slowly circle around the entire small cage because of their larger quantity. In addition, we simulated two fictional scenes where fish with large body size disparity (Fig. 5(c)) and multiple species (Fig. 5(d)) are mixed together. With associated biological measurements and trained policies, fish in each of the size and species groups exhibit distinct swimming patterns while influencing those of other groups in the same fish cage.



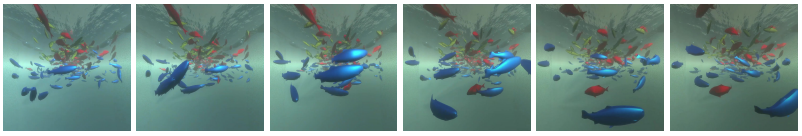
(a) 10 red seabream in a cubic cage with an edge length of 3 meters (left three) and 5 meters (right three).



(b) 45 red seabream in a cubic cage with an edge length of 3 meters.



(c) 1000 coho salmon with body scales of 0.5 (yellow), 1.0 (blue) and 1.5 (red) in an octagon cage with an edge length of 3 meters. The simulated fish form two vertical layers based on their body sizes.



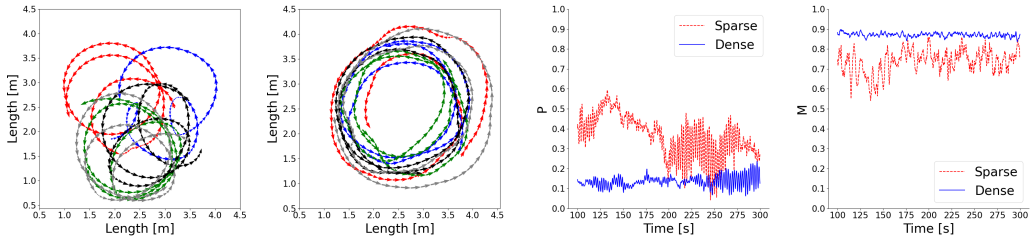
(d) 100 coho salmon (blue), 100 yellowtail (yellow) and 100 red seabream (red) in an octagon cage with an edge length of 3 meters. The swimming pattern vary with their distinct biological factors.

Figure 5: Sequence of frames from example environments with diverse fish simulation configurations. All fish have random body scales within the interval [0.9, 1.1] by default unless specified.

Simulation vs Reality: We focus on the relationship between fish density in a cage and the resulting fish school states, and compare simulation results with data obtained in the field. We use two kinds of schools in our simulations, sparse (0.59 fish/m^3) and dense (14.7 fish/m^3), where the densities are set to be consistent with the field work. In the real fish farms, the sparse cage is $6.5\mathrm{m} \times 6.5\mathrm{m}$ square with a depth of $6\mathrm{m}$ and contains 272 fish. The dense cage is an octagon with $6.5\mathrm{m}$ edges and a depth of $10\mathrm{m}$ and contains about 30 000 fish. We quantify states of schools, and focus on milling, characterized by regular circulation around an axis, and swarming, in which fish aggregate with a weak polarization [47, 48]. Fig.6 shows sampled trajectories, a polar order parameter P given by the norm of the average of direction vectors for each fish in a school, and a normalized angular momentum M representing how regular the circulation is [48] for two schools with different densities. Details can be found in the Appendix. The simulation result demonstrates that the collective behavior of the school transitions from swarming to milling as density increases, and this tendency is in good agreement with the real data.

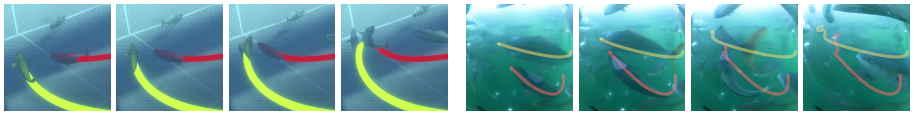
Moreover, we show the influence of the dominance hierarchy on fish behavior in Fig. 7. We initialized the scene with 20% of fish being dominant and the rest being subordinate based on the reported distribution of social hierarchies among fish in [38]. The dominant individuals would occasionally chase their subordinate neighbor until colliding with or losing sight of the latter, which matches our observation of the fish behavior in underwater videos captured in aquaculture sites. A demonstration of the simulation scenes is also included in our supplementary video.

PPO vs SAC: We compare the results of simulations trained by PPO and SAC [49], in the default environment of coho salmon, in which 1000 coho salmon swim in a large octagonal cage with edges of 3 meters and a height of 4.6 meters. Both PPO and SAC eventually produced agents forming swimming patterns that resembled those of real coho salmon in reference videos, while the former yielded more stable outcomes. The same reward parameter setting used for PPO was initially applied to SAC in the pretraining stage. Although the SAC agents collected a higher average reward over



(a) Sampled trajectories (b) Characteristics of school

Figure 6: Relation between density and states (swarming or milling) of fish schools and their characteristics: (a) 5 trajectories in sparse (0.59 fish/m^3, left) and dense (14.7 fish/m^3, right) schools. The left and right panels show swarming and milling state respectively. The fish cages are square with 4.0m edges in the left panel and octagon with 1.66m edges in the right panel. (b) Time series of P (left) and M (right) for dense (blue line) and sparse (red dashed line) schools.



(a) Simulation results. (b) Real recordings.

Figure 7: Sequence of frames from an aggressive behavior instance where a dominant fish (with red trajectory) chases after a subordinate neighbor (with yellow trajectory) until hitting the latter.

episodes from an early stage of training than their PPO counterparts, they became over-trained in the later phase of training and exhibited a circling pattern where all salmon tended to cruise in the same direction and align their body rotations, which would be unnatural in a real-world farming cage setting. This result was jointly caused by a higher sample efficiency and entropy maximization strategy of the SAC algorithm that encourages exploration. After adjusting the reward parameters accordingly, SAC agents were able to swim in equally natural patterns as PPO agents after the pretraining and fine tuning stages. However, no significant improvement in the cumulative average reward was observed in the fine tuning stage of SAC and its resulting value was lower than that of PPO as shown in Fig. 8. A possible explanation is that the aggregated penalties from constraining factors such as boundary and neighbor collisions overwhelmed SAC agents as their off-policy nature required them to collect experience from previous episodes into the replay buffer for future updates.

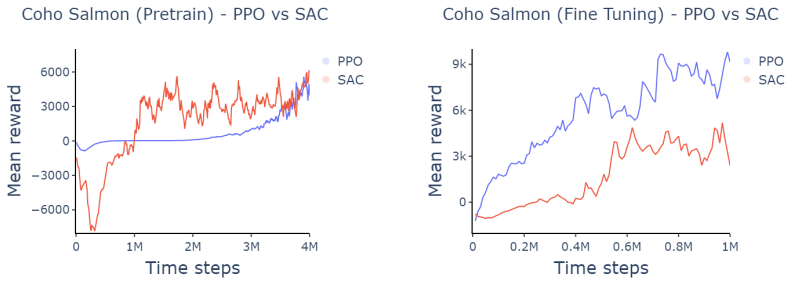
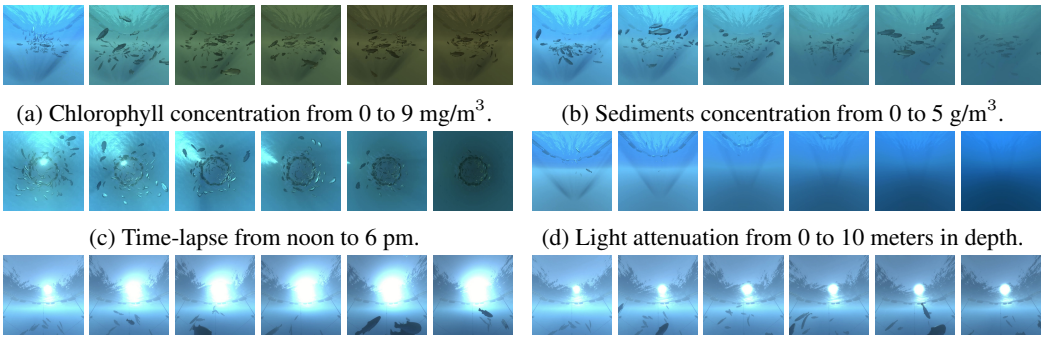


Figure 8: Comparison of per-episode mean cumulative rewards over all coho salmon agents trained using PPO (blue) and SAC (red) algorithms in pretraining (left) and fine tuning (right) phases.

Underwater Simulation: The results of the experiments with each component of the environment simulation are illustrated in Fig. 9. Fig. 9(a) shows how the underwater scene becomes more reddish brown with higher concentrations of chlorophyll resulting in an increasing attenuation of blue. A higher value of sediment concentration makes the water cloudier as shown in Fig. 9(b). Fig. 9(c) depicts a timelapse of the scene. Fig. 9(d) demonstrates the change in brightness with increasing depth, since more light is absorbed at greater depths. Fig. 9(e) shows the influence of changing small particle concentration on the brightness of the scene. A higher concentration of small particles



(a) Chlorophyll concentration from 0 to 9 mg/m^3.

(b) Sediments concentration from 0 to 5 g/m^3.

(c) Time-lapse from noon to 6 pm.

(d) Light attenuation from 0 to 10 meters in depth.

(e) Light scattering with small particles (<1 micron in diameter) concentrations from 0 ppm to 4 ppm.

(f) Light scattering with large particles (>1 micron in diameter) concentrations from 0 ppm to 4 ppm.

Figure 9: Experiments with each component of the environment simulation.

increases the amount of light scattered, both towards and away from the camera, causing the light around the sun to brighten and everything away from the light source to darken. Fig. 9(f) shows the influence of large particle concentration, which works the same way as small particle concentration but has a smaller effect on light scattering.

Figure 10: Example images of a fish counting system using YOLOv4 trained model. The captured video data with (left) coho salmon (middle) yellowtail (right) red seabream were fed to our fish counting pipeline, and the total number of fish in each cage was estimated automatically.

Fish Counting: We applied our modified YOLOv4 model [46][4] trained with the synthetic dataset to real footage. Fig. 10 shows some results with the videos of coho salmon, yellowtail and red seabream. The trained fish counting algorithm estimated the number of fish in each frame, and the total number of fish was computed by taking the max of the estimated values while the network processes all the video frames. The counting results for coho salmons, yellowtails and red seabreams are 69:61, 22:32, and 7:7 (left:our system, right: manually counting) respectively. When the density of the fish is high and many occlusions are observed, the accuracy of the counting system drops.

6 Conclusions and Future Work

We presented a bio-inspired fish simulation system with DRL which adapts to various conditions autonomously by applying transfer learning, and which enabled us to simulate a broad range of fish behaviors in many scenarios. In addition, we introduced a physically-based underwater simulation which enabled us to simulate various undersea conditions. Using these, we were able to synthesize a large dataset with accurate annotations which we could use for computer vision tasks. We presented one application, fish counting, in this paper, but the potential of this synthetic dataset created with our novel simulations goes beyond fish counting. We will extend the application to 3D detection, pixel-wise segmentation and tracking tasks in order to make more useful tools for fish farms. In this paper, we excluded the effect of tidal currents from our simulations, and the proposed simulation method is not applicable to the fish behavior in feeding situation yet. We plan to incorporate them to make our simulation more biologically accurate.

[4] under Apache License 2.0

References

[1] Yuko Ishiwaka, Xiao S Zeng, Michael Lee Eastman, Sho Kakazu, Sarah Gross, Ryosuke Mizutani, and Masaki Nakada. Foids: bio-inspired fish simulation for generating synthetic datasets. *ACM Transactions on Graphics (TOG)*, 40(6):1–15, 2021.

[2] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. pages 25–34, 1987.

[3] Burrhus Frederic Skinner. A case history in scientific method. *American psychologist*, 11(5):221, 1956.

[4] Wolfram Schultz, Peter Dayan, and P. Read Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, 1997.

[5] Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(Supplement 3):15647–15654, 2011.

[6] P Read Montague and Gregory S Berns. Neural economics and the biological substrates of valuation. *Neuron*, 36(2):265–284, 2002.

[7] Fangmiao Sun, Jianzhi Zeng, Miao Jing, Jingheng Zhou, Jiesi Feng, Scott F Owen, Yichen Luo, Funing Li, Huan Wang, Takashi Yamaguchi, et al. A genetically encoded fluorescent sensor enables rapid and specific detection of dopamine in flies, fish, and mice. *Cell*, 174(2):481–496, 2018.

[8] João PM Messias, José R Paula, Alexandra S Grutter, Redouan Bshary, and Marta C Soares. Dopamine disruption increases negotiation for cooperative interactions in a fish. *Scientific reports*, 6(1):1–8, 2016.

[9] Joao PM Messias, Teresa P Santos, Maria Pinto, and Marta C Soares. Stimulation of dopamine d1 receptor improves learning capacity in cooperating cleaner fish. *Proceedings of the Royal Society B: Biological Sciences*, 283(1823):20152272, 2016.

[10] Siddhartha Verma, Guido Novati, and Petros Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854, 2018.

[11] Yi Zhu, Fang-Bao Tian, John Young, James C Liao, and Joseph CS Lai. A numerical study of fish adaption behaviors in complex environments with a deep reinforcement learning and immersed boundary–lattice boltzmann method. *Scientific Reports*, 11(1):1–20, 2021.

[12] Misha B Ahrens. Zebrafish neuroscience: Using artificial neural networks to help understand brains. *Current Biology*, 29(21):R1138–R1140, 2019.

[13] Sehee Min, Jungdam Won, Seunghwan Lee, Jungnam Park, and Jehee Lee. Softcon: Simulation and control of soft-bodied animals with biomimetic actuators. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019.

[14] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. page 43–50, 1994.

[15] Casimir Charles Lindsey. 1 - form, function, and locomotory habits in fish. *Fish Physiology*, 7:1–100, 1978.

[16] Daiki Satoi, Mikihiro Hagiwara, Akira Uemoto, Hisanao Nakadai, and Junichi Hoshino. Unified motion planner for fishes with various swimming styles. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 35(4), 2016.

[17] Peter Sunehag, Guy Lever, Siqi Liu, Josh Merel, Nicolas Heess, Joel Z Leibo, Edward Hughes, Tom Eccles, and Thore Graepel. Reinforcement learning agents acquire flocking and symbiotic behaviour in simulated ecosystems. In *ALIFE 2019: The 2019 Conference on Artificial Life*, pages 103–110. MIT Press, 2019.

[18] Pengming Zhu, Wei Dai, Weijia Yao, Junchong Ma, Zhiwen Zeng, and Huimin Lu. Multi-robot flocking control based on deep reinforcement learning. *IEEE Access*, 8:150397–150406, 2020.

[19] Huiyang Yu, Bo Liu, Chengyun Wang, Xuechao Liu, Xi-Yun Lu, and Haibo Huang. Deep-reinforcement-learning-based self-organization of freely undulatory swimmers. *Phys. Rev. E*, 105:045105, Apr 2022.

[20] Jaedong Lee, Jungdam Won, and Jehee Lee. Crowd simulation by deep reinforcement learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games*, MIG '18, New York, NY, USA, 2018. Association for Computing Machinery.

[21] Qingqing Wang, Hong Liu, Kaizhou Gao, and Le Zhang. Improved multi-agent reinforcement learning for path planning-based crowd simulation. *IEEE Access*, 7:73841–73855, 2019.

[22] University of Western Australia (UWA) Australian Institute of Marine Science (AIMS) and Curtin University. Ozfish dataset - machine learning dataset for baited remote underwater video stations. 2019.

[23] Alzayat Saleh, Issam H. Laradji, Dmitry A. Konovalov, Michael Bradley, David Vazquez, and Marcus Sheaves. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific Reports*, 10(1), Sep 2020.

[24] Ellen M Ditria, Rod M Connolly, Eric L Jinks, and Sebastian Lopez-Marcano. Annotated video footage for automated identification and counting of fish in unconstrained marine environments, 2021.

[25] Charles M. Breder and Florence Halpern. Innate and acquired behavior affecting the aggregation of fishes. *Physiological Zoology*, 19(2):154–190, 1946.

[26] William Wallace Reynolds and Martha Elizabeth Casterlin. Behavioral thermoregulation and the "final preferendum" paradigm. *American zoologist*, 19(1):211–224, 1979.

[27] Arne J Jensen, Bjørn O Johnsen, and Laila Saksgård. Temperature requirements in atlantic salmon (salmo salar), brown trout (salmo trutta), and arctic char (salvelinus alpinus) from hatching to initial feeding compared with geographic distribution. *Canadian Journal of Fisheries and Aquatic Sciences*, 46(5):786–789, 1989.

[28] I Huse and JC Holm. Vertical distribution of atlantic salmon (salmo salar) as a function of illumination. *Journal of Fish Biology*, 43:147–156, 1993.

[29] Anders Fernö, Ingvar Huse, Jon-Erik Juell, and Åsmund Bjordal. Vertical distribution of atlantic salmon (salmo solar l.) in net pens: trade-off between surface light avoidance and food attraction. *Aquaculture*, 132(3-4):285–296, 1995.

[30] Jon-Erik Juell. The behaviour of atlantic salmon in relation to efficient cage-rearing. *Reviews in fish biology and fisheries*, 5(3):320–335, 1995.

[31] Martin Føre, Tim Dempster, Jo Arve Alfredsen, Vegar Johansen, and David Johansson. Modelling of atlantic salmon (salmo salar l.) behaviour in sea-cages: A lagrangian approach. *Aquaculture*, 288(3-4):196–204, 2009.

[32] G. Macaulay, D. Wright, F. Oppedal, and T. Dempster. Buoyancy matters: Establishing the maximum neutral buoyancy depth of atlantic salmon. *Aquaculture*, 519:734925, 2020.

[33] Victoria A Davis, Robert I Holbrook, and Theresa Burt de Perera. Fish can use hydrostatic pressure to determine their absolute depth. *Communications biology*, 4(1):1–5, 2021.

[34] Lucy Odling-Smee and Victoria A Braithwaite. The role of learning in fish orientation. *Fish and Fisheries*, 4(3):235–246, 2003.

[35] Alisha A Brown, Marcia L Spetch, and Peter L Hurd. Growing in circles: Rearing environment alters spatial navigation in fish. *Psychological Science*, 18(7):569–573, 2007.

[36] Thomas H. Miller, Katie Clements, Sungwoo Ahn, Choongseok Park, Eoon Hye Ji, and Fadi A. Issa. Social status–dependent shift in neural circuit activation affects decision making. *Journal of Neuroscience*, 37(8):2137–2148, 2017.

[37] Chiweyite Ejike and Carl B Schreck. Stress and social hierarchy rank in coho salmon. *Transactions of the American Fisheries Society*, 109(4):423–426, 1980.

[38] Yoshitaka Sakakura and Katsumi Tsukamoto. Social rank in schools of juvenile yellowtail, seriola quinqueradiata. *Journal of Applied Ichthyology*, 14(1-2):69–73, 1998.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[41] James E Herbert-Read, Andrea Perna, Richard P Mann, Timothy M Schaerf, David JT Sumpter, and Ashley JW Ward. Inferring the rules of interaction of shoaling fish. *Proceedings of the National Academy of Sciences*, 108(46):18726–18731, 2011.

[42] Andreas Huth and Christian Wissel. The simulation of the movement of fish schools. *Journal of theoretical biology*, 156(3):365–385, 1992.

[43] Yoshitaka Sakakura and Katsumi Tsukamoto. Effects of density, starvation and size difference on aggressive behaviour in juvenile yellowtails (seriola quinquevadiata). *Journal of Applied Ichthyology*, 14(1-2):9–13, 1998.

[44] Yonggang Jiang, Zhiqiang Ma, and Deyuan Zhang. Flow field perception based on the fish lateral line system. *Bioinspiration & biomimetics*, 14(4):041001, 2019.

[45] Leif Ristroph, James C Liao, and Jun Zhang. Lateral line layout correlates with the differential hydrodynamic pressure on swimming fish. *Physical Review Letters*, 114(1):018102, 2015.

[46] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[47] Johann Delcourt and Pascal Poncin. Shoals and schools: back to the heuristic definitions and quantitative references. *Reviews in Fish Biology and Fisheries*, 22(3):595–619, 2012.

[48] Daniel S Calovi, Ugo Lopez, Sandrine Ngo, Clément Sire, Hugues Chaté, and Guy Theraulaz. Swarming, schooling, milling: phase diagram of a data-driven fish school model. *New Journal of Physics*, 16(1):015026, jan 2014.

[49] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section the fish counting part of 5 and 6.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We could not find anything.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary. The instructions of logic to implement the code is included.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Our research target is to generate fish schools, so we don't need to report error bars.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the training part of Section 5.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4 and 5.

   (b) Did you mention the license of the assets? [Yes] See Section 4 and 5.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Section 4 and 5.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We collected and generated the data by ourselves, so it is proprietary.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] All of our data is fish. These measurement methods, while designed to avoid injuring the fish.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]