DeepFoids: Adaptive Bio-Inspired Fish Simulation with Deep Reinforcement Learning

Yuko Ishiwaka[1*†], **Xiao S. Zeng**[2*], **Shun Ogawa**[1],
Donovan Michael Westwater[2], **Tadayuki Tone**[1], **Masaki Nakada**[2‡]
[1]SoftBank Corp., Japan, [2] NeuralX Inc., USA

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 task. 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 [§]. 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 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

[*] Yuko Ishiwaka and Xiao S. Zeng equally contributed to this work.
[†] yuko.ishiwaka@g.softbank.co.jp
[‡] masaki@neuralx.ai
[§] https://sdgs.un.org/2030agenda

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

learning. Furthermore, we developed a physically-based underwater environment simulation. This 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

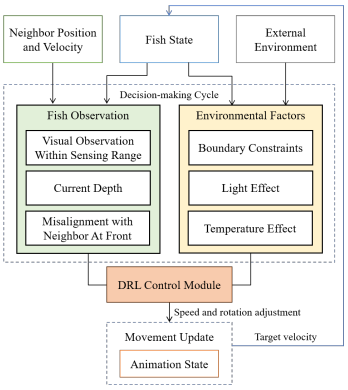


Figure 1: Simulation framework.

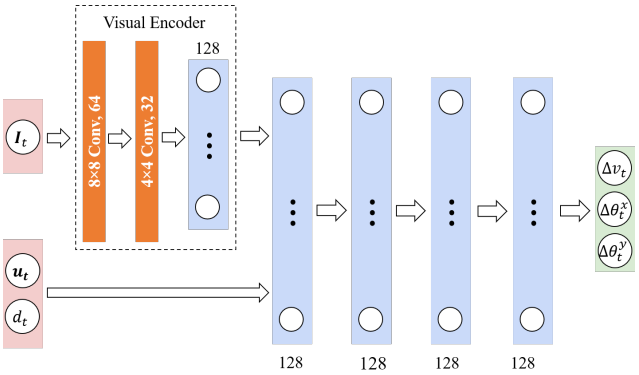


Figure 2: Architecture of policy network.

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.

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 width and height are defined by the grid resolution, which is set to be 34×20. There are 6 channels encoding a scalar value of the normalized distance from the closest detected object within the fish's sensing range d_{sense} to the agent and a one-hot encoding of the object type (i.e. fish, boundary or obstacle). 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 blueto 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:

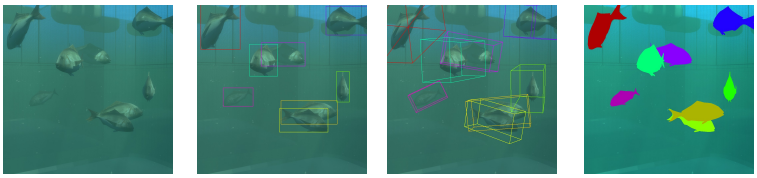
$$r_t^E = -w^r\ \Delta\theta_t - w^s\ \left|\Delta v_t^a\right|.$$

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:

$$r_t^M = \begin{cases} -10 & \text{if} \quad \theta^{rt} \leq \theta_t^x \leq (360 - \theta^{rt}). \\ 2 & \text{if} \quad -\theta^{rt} < \theta_t^x \leq \theta^{rt} \text{ and } v_t^a \geq v^{st}. \\ 0 & \text{otherwise.} \end{cases}$$

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 escape 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 an escape penalty weight w^{tar} for the target as follows:

$$r_t^C = \begin{cases} w^{\text{agg}}\,(\mathbf{d} \cdot \mathbf{v}_t^a) & \text{if fish is an aggressor in chase mode and chases after its target.} \\ 100 & \text{if fish is an aggressor in chase mode and collides with its target.} \\ -w^{\text{tar}}\,(\mathbf{d} \cdot \mathbf{v}_t^a) & \text{if fish is a target in escape mode.} \\ 0 & \text{otherwise.} \end{cases}$$



(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.

The chase and escape modes end for the two fish either 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) 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.

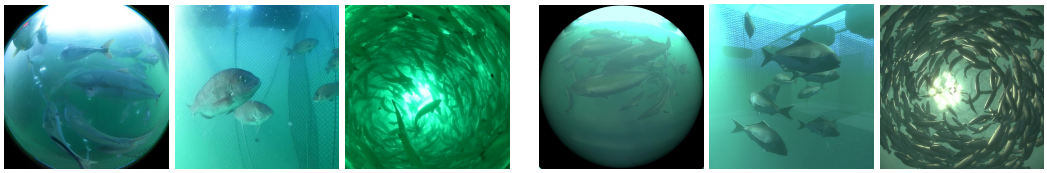
4 Synthetic Dataset and Fish Counting

Our simulation is developed on Unity Engine [¶] with ML-Agents toolkit [46], and Crest Ocean System HDRP asset [‖] 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 [47] 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.

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

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



(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.

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.

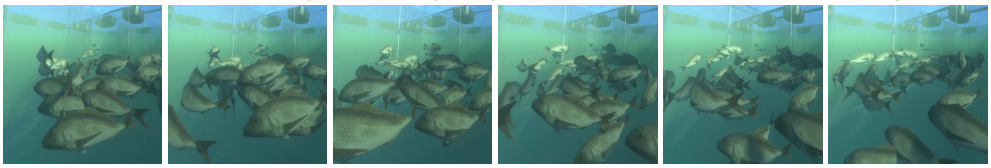
Simulation vs Reality: We focus on the relationship between fish density in a cage and the resulting fish school states, and compare simulation results with data obtained in the field. We use two kinds of schools in our simulations, sparse (0.59 fish$/\mathrm{m}^3$) and dense (14.7 fish$/\mathrm{m}^3$), 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 30000 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 [48, 49]. 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 [49] 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. Several trajectories of individual fish in real video in both sparse and dense schools are exhibited in Fig. 7, and they resemble the ones in simulation results drawn by the bold trajectories in the right panels.

Moreover, we show the influence of the dominance hierarchy on fish behavior in Fig. 8. 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.

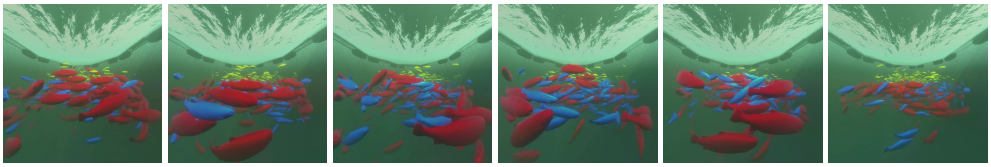
PPO vs SAC: We compare the results of simulations trained by PPO and SAC [50], in the default environment of coho salmon, in which 1000 coho salmon swim in a large octagonal cage with edges of 3 meters and a height of 4.6 meters. Both PPO and SAC eventually produced agents forming swimming patterns that 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 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. 9. 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.

Underwater Simulation: The results of the experiments with each component of the environment simulation are illustrated in Fig. 10. Fig. 10(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. 10(b). Fig. 10(c) depicts a timelapse of the scene. Fig. 10(d) demonstrates the change in brightness with increasing depth, since more light is absorbed at greater depths. Fig. 10(e) shows the influence of changing

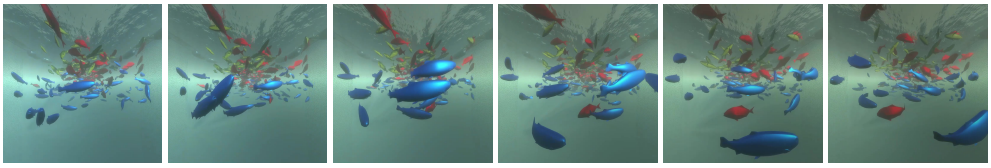
(a) 10 red seabream in a cubic cage with an edge length of 3 meters (left three) and 5 meters (right three).



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

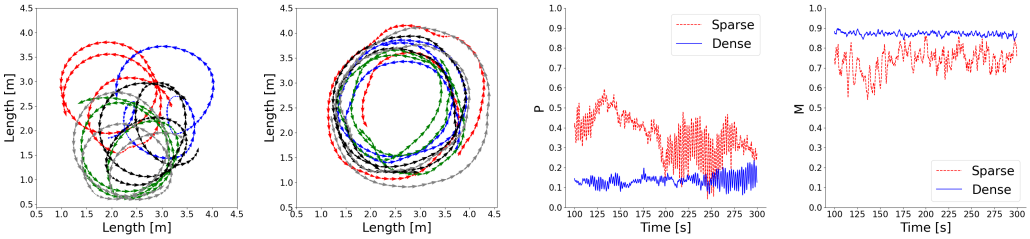


(c) 1000 coho salmon with body scales of 0.5 (yellow), 1.0 (blue) and 1.5 (red) in an octagon cage with an edge length of 3 meters. The simulated fish form two vertical layers based on their body sizes.



(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.

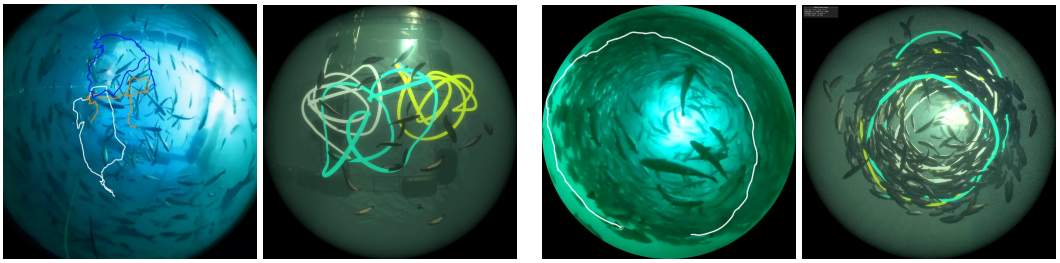


(a) Sampled trajectories

(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.

small particle concentration on the brightness of the scene. A higher concentration of small particles 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. 10(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.



(a) Sparse school (b) Dense school

Figure 7: Fish trajectories in schools in real video (left panels) of coho salmons and simulated ones (right panels). The number of fish and the size and shape of the fish cage in each simulation are described in the caption in Fig. 6. The parameters of real ones are as follows : (a) a sparse school (the fish count is 615) in a rectangular cage with 6.5m edge and 6m depth. Some fluctuations are observed on the orbits in real videos because the camera cannot be fixated completely in the ocean. (b) a dense school (the fish count is approximately 30000) in a octagon cage with 6.5m edge and 10m depth.

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

Figure 8: 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.

Fish Counting: We applied our modified YOLOv4 model [47][**] trained with the synthetic dataset to real footage. Fig. 11 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, size estimation and tracking tasks in order to make more useful tools for fish farms.

A limitation of the current pipeline is the manual setting of weights for reward functions. We will explore techniques that can dynamically adjust weights based on the performance of the DRL controller in the training process. Moreover, we chose not to apply any behavior cloning method because it was difficult to collect and annotate enough samples of fish trajectory data from real video, which was the main challenge we aimed to solve with the proposed method for the preparation of the dataset for computer vision models. However, the fine-tuning of the trained DRL model using behavior cloning with a small sample of the real fish behavioral data is an interesting method to enhance the reality of the simulation that can be worth working on.

[**] under Apache License 2.0

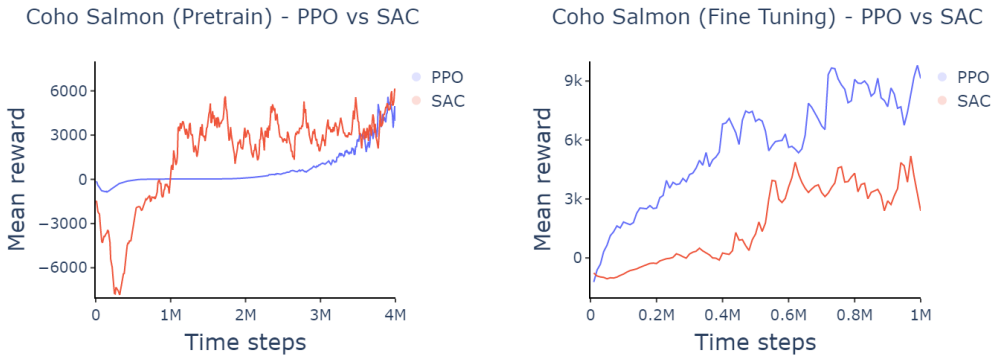
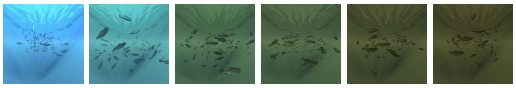
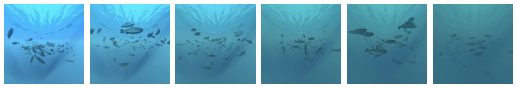


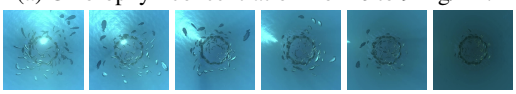
Figure 9: Comparison of per-episode mean cumulative rewards over all coho salmon agents trained using PPO (blue) and SAC (red) algorithms in pretraining (left) and fine tuning (right) phases.



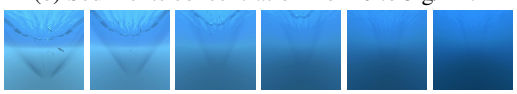
(a) Chlorophyll concentration from 0 to 9 mg/m^3.



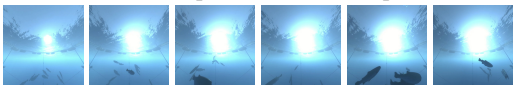
(b) Sediments concentration from 0 to 5 g/m^3.



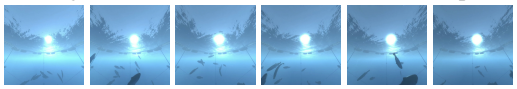
(c) Time-lapse from noon to 6 pm.



(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 10: Experiments with each component of the environment simulation.

In this paper, we excluded the effect of tidal currents and the influence of fish body temperature on movement dynamics from our simulations, and the proposed simulation method is not applicable to special occasions such as fish behavior in feeding and disease conditions yet. We plan to incorporate them to make our simulation more biologically accurate.

Acknowledgment

We are grateful to K. Suda, T. Yoshida, G. Yasui and others from SoftBank Corp., R. Mizutani, T. Tsurumi, S. Sone, and others from Nosan Corporation for field works. We also thank M. L. Eastman and S. Kakazu from SoftBank Corp. for their supports on computer graphics pipeline and M. A. N. Kemas from NeuralX Inc. for computer vision models.

Figure 11: 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.

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] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, et al. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

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

[48] 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.

[49] 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.

[50] 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]

A Biological Background

Fish species prefer different temperatures and will change their swimming patterns or vertical alignment based on the surrounding water temperature [1]. Generally, adult salmonids prefer water temperature within the 8-9 degrees Celsius range [2], whereas yellowtail amberjack and red seabream like higher temperatures [3, 4]. Caged fish also show distinct swimming patterns at daytime and night. They move very little and stay closer to water surface after sunset, but gradually become more active and swim deeper after sunrise as the amount of sunlight increases [5, 6]. Additionally, when fish in the cage reach a certain crowding density threshold, they attempt to maintain their optimal personal space while still being able to move around [7] by forming a circular schooling pattern, consistent with the Boids model [8]. Fish normally react to their neighbors within a distance of 2 to 3 body lengths (BLs) [9, 10] and can learn to avoid cage walls [11, 12]. The reaction appears regulated by visual stimuli [7], as well as the lateral line system, which allows the fish to feel the vibrations of their neighbors [13]. Fish align their orientation primarily to the fish in front of them, and adjust speed as necessary to avoid collision with neighbors [10]. Caged fish like salmonids often settle into two or three vertical layers [14] since they typically group with similarly sized fish [15], where larger fish tend to swim at lower depths and smaller fish stay at higher depths [16, 17]. Moreover, the decision making interval of 95% of fish species is controlled by the Mauthner cells (M-cells) that trigger a rapid escaping movement (C-start) for risk avoidance and command neurons active in decision-making [18, 19]. Miller et al. [20] studied zebrafish social groups and found that action potentials for swimming, which is generated by the neural circuit, lasted for a duration of 50-200 ms. Since salmonids have lower body temperature whereas yellowtail amberjack and red seabream have similar body temperatures as zebrafish, we set 200ms as the the decision-making interval for coho salmon and 100ms as that for yellowtail amberjack and red seabream.

B Environmental Background

Colors underwater are determined by the intrinsic colors of objects and the scattering and absorption of light. The main factors that determine the absorption and scattering are the properties of the water itself and the particulates in the water, usually chlorophyll and sediments [21]. Water absorbs light of different wavelengths at different rates, characterized by attenuation coefficients, which can be changed by the presence of particulates [22, 23]. The exact relationship between transmitted light and the light absorbed is expressed by the Beer Lambert law [21]. Chlorophyll absorbs blue light the most, followed by red, then green. Thus a higher concentration of chlorophyll shifts the coloring in a reddish green direction. Sediments in the water scatter the light, altering the amount of light absorbed as well as changing the attenuation coefficients. Sediments and suspended solids also create turbidity and occlusion underwater. In our model, sediment concentration is the main driving force behind turbidity.

C Simulation Setting

We created eleven simulation environments to assess the performance of proposed fish simulation framework and generate synthetic dataset for computer vision tasks. The configuration of each environment is available in Table 1. The first three environments reproduce the underwater scenes in fish farms we collected field data from and are used as default scenes to pretrain each of the three species. The other eight environments are used to examine the adaptiveness of the framework and transfer learning is applied. Values in the cage size column are in units of meter. The biological and environmental parameters used in these scenes are summarized in in Table 2.

D Effects of Light Intensity and Temperature on Fish Behavior

A fish prefers to stay within a certain area of the tank where its light intensity preference matches the local light intensity computed by the Beer-Lambert law:

$$I = I_0 \times e^{-ad} \tag{1}$$

where I and I_0 denote the light intensity at the local depth and on water surface, a is an attenuation coefficient and d is the depth of fish. I_0 and α are valued based on the field data collected from

Table 1: Simulation environment configurations

Species	Fish Number	Body Scale	Cage Size	Cage Shape
Coho salmon	1000	[0.9, 1.1]	Edge of 3, Height of 4.6	Octagon
Yellowtail amberjack	45	[0.9, 1.1]	3×3×3	Cube
Red seabream	10	[0.9, 1.1]	3×3×3	Cube
Red seabream	10	[0.9, 1.1]	5×5×5	Cube
Red seabream	50	[0.9, 1.1]	3×3×3	Cube
Coho salmon	300	0.5	Edge of 3, Height of 4.6	Octagon
Coho salmon	300	1.0	Edge of 3, Height of 4.6	Octagon
Coho salmon	300	1.5	Edge of 3, Height of 4.6	Octagon
Coho salmon	1000	0.5, 1.0 or 1.5	Edge of 1.8, Height of 3	Octagon
Coho salmon	1000	0.5, 1.0 or 1.5	Edge of 3, Height of 4.6	Octagon
All three species	300	[0.9, 1.1]	Edge of 3, Height of 4.6	Octagon

Table 2: Biological and environmental parameters in fish simulation

Description	Parameters	Unit	Value
Body length	BL	m	[0.34, 0.52]
Speed while schooling (coho salmon)	v_t^a	BL/second	[0.2, 1.9]
Speed while schooling (yellowtail amberjack)	v_t^a	BL/second	[0.2, 2.1]
Speed while schooling (red seabream)	v_t^a	BL/second	[0.2, 2.7]
Initial speed	v_0	BL/second	0.2
Maximum delta speed in cage environment	Δv_{max}	BL/second	4.1
Light intensity at water surface (Onmaehama)	I_0	PAR	[0, 519.69]
Light intensity at water surface (Nishiki)	I_0	PAR	[0, 15.13]
Light attenuation coefficient (Onmaehama)	a	N/A	1.5
Light attenuation coefficient (Nishiki)	a	N/A	0.26
Lower bound of preferred light intensity (coho salmon, regular size)	I_{lpref}	PAR	[1.1, 4.5]
Lower bound of preferred light intensity (coho salmon, small size)	I_{lpref}	PAR	[1.1, 4.5]
Lower bound of preferred light intensity (coho salmon, large size)	I_{lpref}	PAR	[0.1, 0.5]
Lower bound of preferred light intensity (yellowtail amberjack)	I_{lpref}	PAR	[0.01, 0.02]
Lower bound of preferred light intensity (red seabream)	I_{lpref}	PAR	5.75
Upper bound of preferred light intensity (coho salmon, regular size)	I_{hpref}	PAR	[3.5, 7]
Upper bound of preferred light intensity (coho salmon, small size)	I_{hpref}	PAR	[17.5, 35]
Upper bound of preferred light intensity (coho salmon, large size)	I_{hpref}	PAR	[2.5, 4]
Upper bound of preferred light intensity (yellowtail amberjack)	I_{hpref}	PAR	[341.55, 805]
Upper bound of preferred light intensity (red seabream)	I_{hpref}	PAR	368
Steepness parameter of light reaction	I_{lsteep}	PAR	-20
Steepness parameter of light reaction	I_{hsteep}	PAR	1000
Lower bound of preferred temperature (coho salmon)	T_{lpref}	°C	8
Lower bound of preferred temperature (yellowtail amberjack)	T_{lpref}	°C	18
Lower bound of preferred temperature (red seabream)	T_{lpref}	°C	20
Upper bound of preferred temperature (coho salmon)	T_{hpref}	°C	9
Upper bound of preferred temperature (yellowtail amberjack)	T_{hpref}	°C	28
Upper bound of preferred temperature (red seabream)	T_{hpref}	°C	28
Steepness parameter of temperature reaction	T_{lsteep}	°C	-60
Steepness parameter of temperature reaction	T_{hsteep}	°C	80
Temperature at water surface at noon	T_{surf}	°C	8.3

aquaculture sites as described in Sec. I. Any deviation of I from the light intensity preference range results in a vertical delta velocity ($\Delta\mathbf{v}_{\text{light}}$) for the fish as shown in Eq. (2):

$$\Delta\mathbf{v}_{\text{light}} = \begin{cases} \frac{I_{\text{lpref}} - I}{I_{\text{lpref}} - I_{\text{lsteep}}}\,[0,1,0] & \text{if} \quad I \leq I_{\text{lpref}} \\ \frac{I - I_{\text{hpref}}}{I_{\text{hsteep}} - I_{\text{hpref}}}\,[0,-1,0] & \text{if} \quad I \geq I_{\text{hpref}} \end{cases}, \tag{2}$$

where parameters I_{lsteep} and I_{hsteep} control the steepness of the reaction to light and are set in an identical manner to [9]. I_{lpref} and I_{hpref} denote the lower and upper bounds of the light preference interval and are valued based on the biological studies of each simulated species [24, 3, 4, 5]. Similarly, a fish likes to stay in the area where the local temperature satisfies its temperature preference. We

calculate the temperature induced vertical delta velocity ($\Delta \mathbf{v}_{\text{temp}}$)

$$\Delta \mathbf{v}_{\text{temp}} = \begin{cases} \frac{T_{\text{lpref}} - T}{T_{\text{lpref}} - T_{\text{lsteep}}} [0, 1, 0] & \text{if} \quad T \leq T_{\text{lpref}} \\ \frac{T - T_{\text{hpref}}}{T_{\text{hsteep}} - T_{\text{hpref}}} [0, -1, 0] & \text{if} \quad T \geq T_{\text{hpref}} \end{cases}, \tag{3}$$

where the under-water ambient temperature (T) is obtained using a two-layer method based on the studies by [25] and [26]. The temperature response steepness parameters (T_{lsteep} and T_{hsteep}) are set to be the same as those in [9], and the bounds of the temperature preference interval (T_{lpref} and T_{hpref}) are determined according to [3], [4] and [2].

E Deep Reinforcement Learning

Proximal Policy Optimization Proximal Policy Optimization (PPO) algorithm [27] improves upon vanilla policy gradient method by providing more stability and reliability during learning. It is a variant of Trust Region Policy Optimization (TRPO) algorithm [28] with significantly simpler implementation by optimizing a surrogate loss instead of KL divergence constraints. In our work, the clipped surrogate loss $L^{\text{CLIP}}(\theta)$ with respect to the current policy parameters θ is defined as follows:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_{s_t, a_t} \left[\min(l_t(\theta)\hat{A}_t, \text{clip}(l_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right]$$

$$l_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

π_θ and $\pi_{\theta_{\text{old}}}$ are current and old policies. $l_t(\theta)$ is the likelihood ratio of the action probability under current policy over that of old policy. As the new policy deviates from the old policy, $l_t(\theta)$ will deviate from 1 and be constrained by the interval $[1-\epsilon, 1+\epsilon]$. This means whenever excessive policy update happens the gradient will be set to zero and therefore prevents the new policy from being too different from previous one. $\hat{A}_t$ represents the generalized advantage estimation [29]. A minimum value is then chosen to obtain a lower bound of loss.

Since the parameters between the policy and value function are shared in networks, we further integrate the squared-error loss of value function ($L_t^{\text{VF}}(\theta)$) and an entropy term ($S[\pi_\theta](s_t)$) ensuring sufficient exploration to obtain the final objective loss:

$$L_t^{\text{CLIP+VF+}S}(\theta) = \mathbb{E}_{s_t, a_t} \left[L_t^{\text{CLIP}}(\theta) - cL_t^{\text{VF}}(\theta) + \beta S[\pi_\theta](s_t) \right]$$

where value function loss coefficient c is valued at 0.5 and entropy coefficient β is valued at 0.0005.

Parameters in Action and Reward Definition The values of parameters used in action definition and reward function for PPO are listed in Table 3. The weight of each reward term is set according to the objective of each training. For example, we set the boundary avoidance weight to be the highest when the DRL controller was trained for the very first time because the primary goal of the training for the fish was to avoid the collision to the fish cage. When it is learnt, we increased the weights for the neighbor collision penalty and energy consumption penalty so that fish learn the smooth schooling in an environment as well as energy saving, which is an important benefit of being in a school [30].

PPO Training Configuration We present the training hyperparameters of PPO in Table 4. Similar configurations were used for pretraining and transfer learning phases except the total number of steps when training coho salmon. This is primarily due to the variation in environment configurations, which makes retraining necessary. We also include an illustration of the learning curves of the three species in Fig. 1 to showcase the performance improvements over the entire training process.

SAC Training Configuration We report the training hyperparameters of Soft Actor-Critic (SAC) algorithm [31] that we employed to train the fish agents aside from PPO in Table 5. Since we only compare the results of SAC and PPO in the default environment of coho salmon, the total numbers of time steps for SAC agents in pretraining and fine tuning phases are also four and one million.

Table 3: Parameters in action definition and reward function

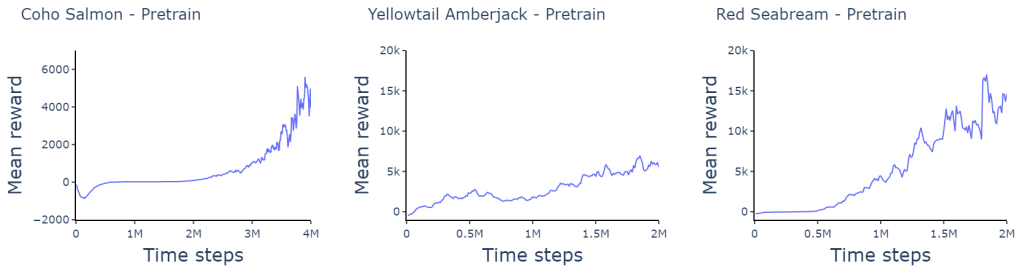
Description	Parameter	Unit	Value
Sensing range (coho salmon and red seabream)	d_{sense}	BL	3
Sensing range (yellowtail amberjack)	d_{sense}	BL	2
Clamping angle of rotation about z-axis	θ^{zt}	Degree	10
Pitch angle threshold	θ^{rt}	Degree	53
Speed threshold	v^{st}	BL/second	0.3
Probability to start chase mode at each time step	p_a	N/A	0.005
Neighbor collision weight (pretraining)	w^{NC}	N/A	0.5
Neighbor collision weight (transfer learning)	w^{NC}	N/A	4
Boundary avoidance weight	w^{BD}	N/A	2
Neighbor interaction weight	w^{ND}	N/A	1.5
Rotation penalty weight (pretraining)	w^{r}	N/A	0.001
Rotation penalty weight (transfer learning)	w^{r}	N/A	0.5
Speed penalty weight (pretraining)	w^{s}	N/A	5
Speed penalty weight (transfer learning)	w^{s}	N/A	10
Chase reward weight for aggressors	w^{agg}	N/A	8
Escape penalty weight for targets	w^{tar}	N/A	1

F Physically Based Environment Simulation

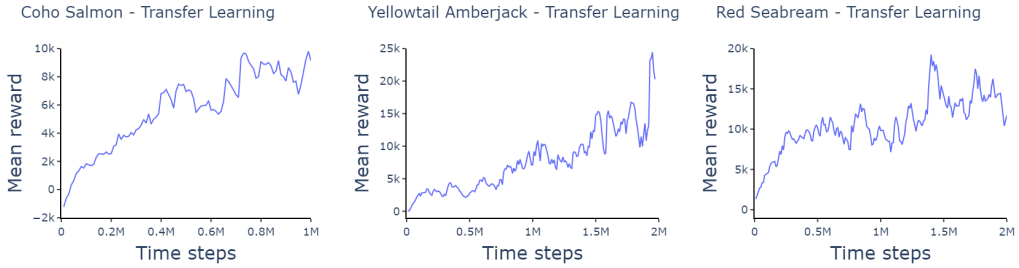
Light Attenuation The simulation colors the water by first marching rays in fixed distance steps for every pixel on screen. The ray's direction is determined based on the pixel's location when transformed into world space. The current color of the pixel is then treated as incoming light, and becomes the input for the attenuation calculation using the Beer-Lambert Law shown in Eq. (1). I_0 represents the incoming light (or in this case the current associated pixel color) of the ray, a (or in this case $a_{\text{r/g/b}}$) is the attenuation coefficient of red, green or blue color in the vector $\mathbf{ac}$ which is determined by three components that we explain next, and d is the distance the ray of light has traveled through the water [32]. Then the simulation calculates the ambient light at the current point the ray is at. This is calculated by first summing up all the light that is heading towards the current point, and multiplying it by the dot product of the sunlight direction and ray direction. Next, the output of the ambient calculation is multiplied by the output of the attenuation calculation and is then added to a running total. Then the ray marches for a step in its set direction and the process repeats until it reaches its maximum distance. Once that is done, the running total is returned as the output color for the pixel associated with the given ray.

Table 4: Hyperparameters used in training fish agents with PPO.

Parameter	Value
Value function loss coefficient for PPO c	0.5
Entropy coefficient for PPO β	0.0005
PPO clip threshold ϵ	0.2
Regularization parameter for GAE λ	0.95
Batch size	1024
Buffer size	10240
Learning rate	0.001
Num. of epochs	3
Discount factor γ	0.99
Time horizon	64
Total number of steps (pretraining, coho salmon)	4 million
Total number of steps (transfer learning, coho salmon)	1 million
Total number of steps (pretraining, yellowtail amberjack and red seabream)	2 million
Total number of steps (transfer learning, yellowtail amberjack and red seabream)	2 million



(a) Mean cumulative reward over all agents at each episode in pretraining phase. Left-to-right: coho salmon, yellowtail amberjack and red seabream



(b) Mean cumulative reward over all agents at each episode in transfer learning phase. Left-to-right: coho salmon, yellowtail amberjack and red seabream

Figure 1: Learning curves from PPO of the three fish species in pretraining and transfer learning phases in their corresponding default environments. Performance is measured by mean cumulative reward per episode smoothed by exponential moving average. Notably, coho salmon took four million time steps during pretraining followed by one million during transfer learning because of its larger quantity. Nevertheless the other two species were trained for two million steps in both phases.

The attenuation coefficient vector consists of three parts: the attenuation of the water itself, the attenuation of chlorophyll, and the attenuation of the suspended sediments. Each part is a 3 dimensional vector representing the attenuation coefficients for red, green, and blue. The values of these coefficient vectors are determined based on data found in [32, 33, 34]. The coefficients of each are combined together, with the chlorophyll and sediments both multiplied by the concentration of their respective parameters using the model described in [32] and [35]. The formula is as follows:

$$\mathbf{ac} = \mathbf{ac}_{\text{bw}} + co_{\text{chl}}\mathbf{ac}_{\text{chl}} + co_{\text{sed}}\mathbf{ac}_{\text{sed}} \tag{4}$$

where $\mathbf{ac}_{\text{bw}}$ is the attenuation coefficient vector of the base water, co_{chl} is the concentration of chlorophyll, $\mathbf{ac}_{\text{chl}}$ is the attenuation coefficient vector of chlorophyll, co_{sed} is the concentration of the sediments, and $\mathbf{ac}_{\text{sed}}$ is the attenuation coefficient vector of sediments.

Table 5: Hyperparameters used in training coho salmon agents with SAC.

Parameter	Value
Target smoothing coefficient τ	0.005
Initial entropy coefficient α (pretraining)	0.3
Initial entropy coefficient α (fine tuning)	0
Average steps per update of policy	1000
Batch size	1024
Replay buffer size	102400
Learning rate (pretraining)	0.001
Learning rate (fine tuning)	0.0005
Discount factor γ	0.99
Time horizon	64
Total number of steps (pretraining)	4 million
Total number of steps (fine tuning)	1 million

The factors such as the concentrations of sediments and chlorophyll are set by the simulation to random values from within user defined ranges. These ranges are based off values found in real world, like those laid out in [33] and [35]. The scene uses these factors to determine everything else about the environment. For example, the turbidity of the water is determined by the concentration of suspended sediments.

Light Scattering A light scattering system computes the light refraction based on the medium's properties such as the particles inside a fluid [36]. It uses a phase function, which can be obtained from a volume scattering function (VSF), to calculate the amount of light traveling through a medium and being scattered towards the viewer. We use the Kopelevich model [37] as the VSF to determine how different wavelengths of light are scattered underwater using large and small suspended particles. It can be formulated as the following equation [38]:

$$\text{VSF} = \beta_w + v_s \beta_s \psi \left(\frac{500}{\lambda}\right)^{1.7} + v_l \beta_l \psi \left(\frac{500}{\lambda}\right)^{0.3} \tag{5}$$

where β_w, β_s and β_f denote the VSF of pure sea water, small and large particles respectively and their values can be selected from tabular values in [37] by performing visual inspection. v_s and v_l are the concentrations of small and large particles (in units of parts per million). ψ is the angle between the light and view directions, and λ is the wavelengths of red, green and blue. The resulting VSF is stored in a vector that is used to calculate the scattering coefficients for the three colors via numerical integration using the trapezoidal rule. Finally, the VSF vector is divided by the scattering coefficients to obtain the phase function for each of the colors, which is then used to multiply the output of the ray marching system to scatter the color based on the RGB value.

G Characteristics of fish school states

Data provided by the simulation are body length l_{BL}, position $\mathbf{x} = (x, y, z)$ of fish body center, speed v and its direction vector $\mathbf{d} = (d_x, d_y, d_z)$ ($|\mathbf{d}| = 1$) for each individual fish and each time slice. The velocity vector is $\mathbf{v} = v\mathbf{d}$, and the deviation from the center of the school $\mathbf{x}_{\mathrm{c}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$ is $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}_{\mathrm{c}}$, where N denotes the number of fish in the school.

Order parameters According to [39, 40] two parameters, a polar order parameter P and a normalized angular momentum M of the school, are used to characterize state of school. The parameters P and M are given respectively as

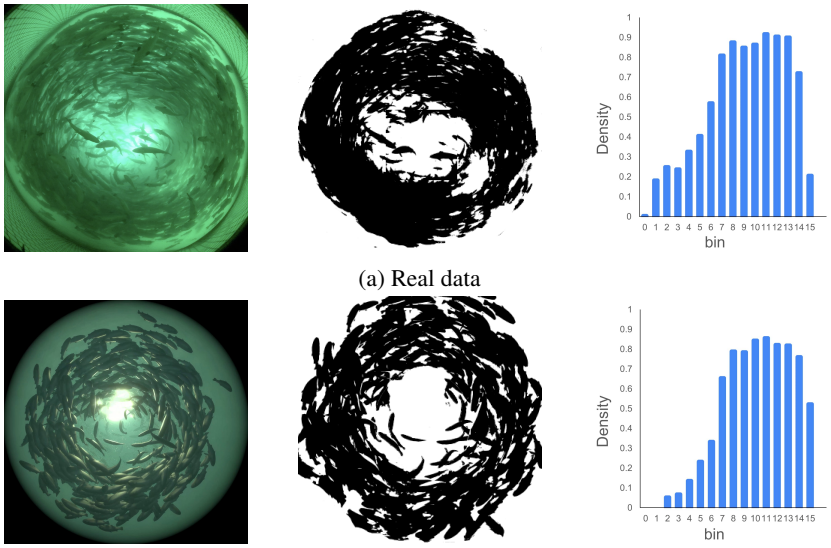
$$P = \left| \frac{1}{N} \sum_{i=1}^{N} \mathbf{d}_i \right|, \quad M = \left| \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{r}_i \times \mathbf{v}_i}{|\mathbf{r}_i||\mathbf{v}_i|} \right| = \left| \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{r}_i \times \mathbf{d}_i}{|\mathbf{r}_i|} \right|, \tag{6}$$

in the same manner as in Ref.[40]. All fish move in the same direction in parallel when $P \simeq 1$ [39, 40], which cannot be observed in a fish cage. A large $M \simeq 1$ and small $P \simeq 0$ brings about the milling state, in which fish in a school rotate around an axis regularly [39, 40]. The smaller M and P indicate that the state is swarming state rather than milling state [39].

Similarity between pictures of real and the simulation To see how the simulation results are consistent with the real data obtained from the field work, we compare the images obtained by both of them. These images are fish-eye lens view, and black and white images are made with some threshold from them. We compare the density of black pixels for each bin obtained by dividing each image into 16 annuli. As a result exhibited in Fig. 2, the simulation image seems to be consistent with the real one.

H Fish Counting

Image Pre-Processing The first module converts the input video to a sequence of images and applies denoising to produce the data format required by the fish detection module. The denoising process applies median blur to the images to reduce the effect of noise, resulting from caustics and sediment and other particles, often seen in underwater photography.



(a) Real data

(b) Simulation

Figure 2: Comparison between (a) real and (b) simulated obtained images. For both (a) and (b), left, middle, and right panels are respectively the fish-eye view picture, black and white images made from the left ones with some threshold, and density of black pixels for each bin obtained from the fish-eye images by dividing them into 16 annuli whose centers are the center of each picture.

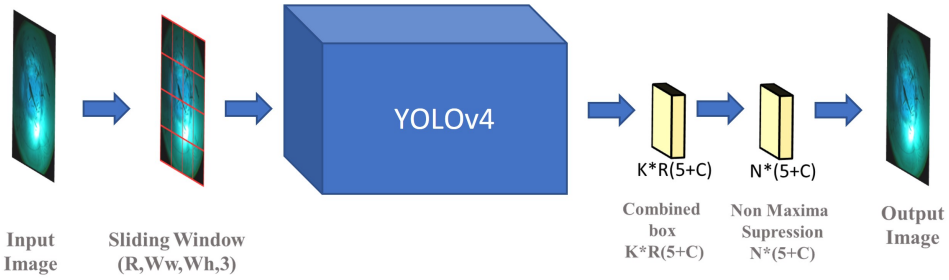


Figure 3: Fish detection model network architecture. This module is a region proposal network based on the YOLOv4 model. A sliding window is added as a first layer to crop sub-windows to magnify smaller fish in original images, which effectively avoids false negatives during smaller fish detection.

Fish Detection Fig. 3 shows the network architecture of the fish detection model, based on YOLOv4 [41]. The input to the model is a sequence of images, and the output is a list of proposed bounding boxes (x, y, w, h), where (x, y) is the coordinate of the top left corner of the bounding box, and w and h are the width and height of the bounding box, respectively, with an associated class label and confidence score. We set the confidence score threshold to be 0.5 and any proposed regions below the threshold are discarded. In order to solve the issue of false negatives in the case of smaller fish, a sliding window is added as the first layer of the network. Each window crops a sub-image SI_i with a size of 416×416 from the original image I. The sliding window is shifted repeatedly horizontally and vertically by an amount equal to half the window size, producing a sub-image SI_i at each position that is fed to the YOLOv4 network. The produced number of sub-images, R, is computed using Eq. (7).

$$R = \left(\left| \frac{I_{\text{width}} - SI_{\text{width}}}{s} \right| + 1 \right) \times \left(\left| \frac{I_{\text{height}} - SI_{\text{height}}}{s} \right| + 1 \right), \tag{7}$$

where SI_{width} and SI_{height} are the width and height of the sliding window, and I_{width} and I_{height} are the width and height of the original image, respectively. s is a step-size term and is set to be half of the sliding window size. This technique magnifies small fish in the original images. The computation time increases by a factor of R when processed serially; however, it helps to reduce the number of false negatives which is a major problem with the original YOLOv4. Note that we concatenate all the results from the cropped images into a single array, and apply Non-Maximum Suppression (NMS) to produce a single output, unlike the standard YOLO where the NMS is applied just after a single detection block.

Fish Counting After the completion of the detection module, the estimation of fish count is tabulated for each frame, and the maximum value is taken as the final estimate of the number of fish in the cage. Despite its simplicity, we found the MAX method to give us the highest counting accuracy.

Training We trained the model for 2000 epochs on a single Nvidia GeForce GTX 1080 Ti with an initial learning rate of 0.001. The training process completed in 5 hours. We then manually annotated 5 frames from 5 videos recorded at a real fish farm–a total of 25 images–and used them to fine-tune the model.

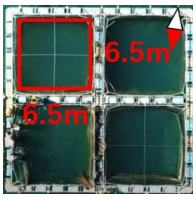
I Field Data collection

We collected data on four species (two salmonids, yellowtail, red seabream) at marine fish cages. The salmonids cages are in the north, and the yellowtail and red seabream are in the south. Our goal was to measure the physical properties and dimensions of the environment and fish, to set the parameters of our simulation, and to record video of caged fish schooling behavior. In this appendix, we will discuss that data collection process and show our results.

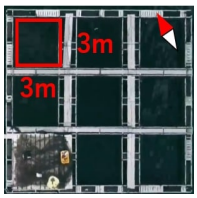
Location In March, May and July of 2021, we collected data from coho salmon and trout salmon cages on a northern fish farm (latitude 38.465, longitude 141.480). Both fish cages were cubes with a side length of 6.5m (Fig. 4(a)). Then, in June and November of 2021, we collected data from yellowtail and red seabream cages on a southern fish farm (latitude 34.215 and longitude 136.386). Both cages were cubes with side lengths of 3m (Fig. 4(b)).

Equipment Fig. 5 shows the equipment used. Fig. 5a is circuit board used to measure light intensity. A Raspberry Pi is used to collect data from the TSL2571 light intensity sensor. To help it withstand the water pressure, the circuit board was placed in a clear smartphone case with the sensor positioned to avoid being obstructed. Additionally, the case was placed in a waterproof housing. After waterproofing, the sensor was attached alongside cameras to PVC pipes and used to measure light intensity at various depths underwater. Fig. 5b shows the device used to measure water temperature, the SK SATO SK1260. The sensor must be in direct contact with sea water, so it was lowered into the ocean. To film video of the fish schooling behaviors, we used KODAK 4KVR360 cameras. A camera placed at the center of the floor of the cage pointing up (Fig.5c) was fixed in place with four ropes. The cameras placed at the sides of the cage facing the center (Fig.5d) were fixed to PVC pipes and lowered to the target depth. When filming from multiple depths simultaneously, the cameras were fixed to long PVC pipes at 1m intervals, as in Fig.5f. The PVC pipe fixtures are shown in Fig.5e. By fixing the T-shaped part to the iron pipes of the cage, long video shoots are possible. The cameras are placed at R, G, B, and Y as shown in Fig. 4c.

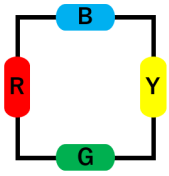
Environment Data Fig. 6a shows an example of the results of our light intensity measurements. The measurements were taken at the southern fish farm in November 2021, starting at 11:10 AM. The x-axis shows the number of minutes since the start, while the y-axis is the light intensity (Lux). Light blue shows the readings at a depth of 5m (center bottom), red shows readings at 3m (side), yellow shows readings at 2m (side), and black shows readings at 1m (side). It can be seen that light has lower intensity at greater depths. Sudden noisy drops in brightness are likely the result of fish blocking light from reaching the sensor. Fig. 6b shows the temperature at each depth. The top green line is data from the southern farm in June. The rest are data from the north farm, recorded in (from bottom to top) March, May, and July. Although there are seasons without any temperature variation, at other times the temperature can differ by 3-4 degrees. During the period from May to July, when



(a) In north



(b) In south



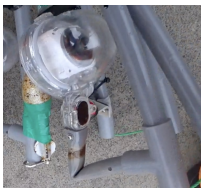
(c) Camera locations

Figure 4: Fish firming cages and camera locations. (a) Salmonoids fish cages in north (coho salmon and trout salmon), (b) yellowtail and red sea bream fish cages in south, (c) camera locations in a cage.

(a)

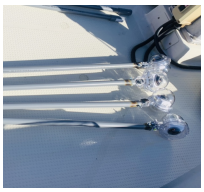
(b)

(c)



(d)

(e)



(f)

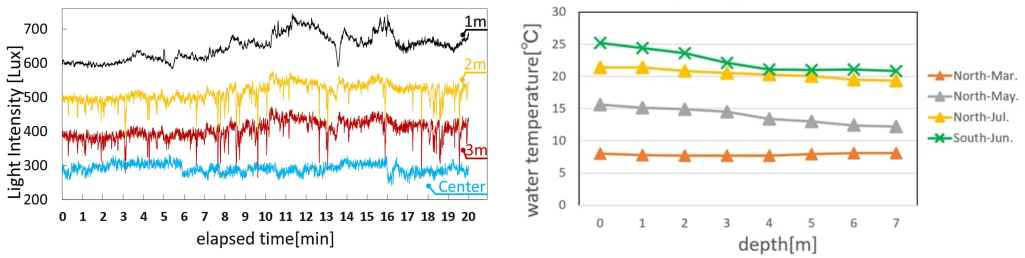
Figure 5: Equipment used at the fish cage. (a) light intensity sensor, (b) water temperature sensor, (c) center bottom camera, (d) side camera. A light intensity sensor is set behind the camera, (e) fixing PVC pipe, with cameras attached, to the cage, (f) cameras attached to a PVC pipe.

water temperature exceeds 10 degrees, the chlorophyll content of the water increases, lowering the clarity of the water.

Body Measurements and Fish Count The number of fish in a fish farm is unclear. Fish are purchased by weight as juveniles. This weight is divided by an average weight to give an approximate count, but this can result in up to 50% error. A fish's value on the market is determined by weight. Therefore, fish farmers try to monitor the state of their fish (length, height, width, and weight) in order to optimize their feeding, but the process is dependent on farmers' intuition. The only way to accurately measure a fish is by hand after removing it from the water[1]. To do so, fish are scooped up by a net a few fish at a time; when the fish are released again, they are counted by eye. When populations are small this process can be performed for all fish, but it is not realistic for more than 1000 fish. For such cases, the process is performed for approximately 10% of the fish and the result is multiplied by 10, giving an inexact value. Fig. 7c shows the current fish counting process.

Fish body measurements are even more difficult. To measure a fish, the fish is removed from the water and anesthetized to prevent injury, and measured by hand with a tape measure (see Fig. 7). Fig. 8 shows the body lengths of 10 randomly selected fish. Fig. 8a shows the results for salmon (coho salmon and trout salmon), and Fig. 8b shows the results for yellowtail and red seabream. The x-axis shows the month the measurement was performed. The lines show the range of body lengths, and the symbol is the average length (coho salmon: gray, trout salmon: orange, yellowtail: yellow, red seabream: red). Salmon farming begins in December, and they are shipped to market in July. The graph shows that there exists large variation in salmon body length, which gets smaller as July approaches. The shipping date for yellowtail and red seabream is not fixed; they are shipped once they reach a set size.

[1]These measurement methods, while designed to avoid injuring the fish, are neither completely safe nor particularly accurate. There is a demand for an alternative method to obtain these data from video data



(a) Light intensity (b) Water temperature

Figure 6: Environment data: (a) November 2021, Northern farm light intensity. From bottom to top: center bottom (5m), 3m, 2m, 1m. (b) Water temperature for each depth.

(a) Scooping fish (b) Fish body measurement (c) Fish counting

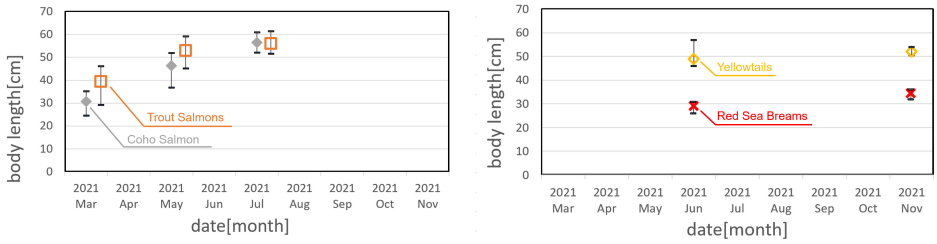
Figure 7: Fish body measurement and counting process

Video Recording Videos at the fish farm were filmed with KODAK 4KVR360 cameras. One camera was placed at the bottom center of the cage, facing the ocean surface. The others were attached to PVC pipes placed at the center of each side at 1m intervals. The footage captured in March, May, and July 2021 on the northern farm and June 2021 on the southern farm was filmed in the KODAK 4KVR360 235 degree dome mode. The November 2021 footage on the southern farm was filmed in the KODAK 4KVR360 155 degree front mode. The coho salmon, yellowtail, and red seabream footage is show in Fig. 9. The cameras are placed at R, G, B, Y in Fig.4c. The salmon and yellowtail footage was taken at a depth of 3m, while the red seabream footage was taken at a depth of 2m. The footage at these depths contained the largest number of fish for each species, showing that the preferred depth differs per species.

References

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

[2] 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.

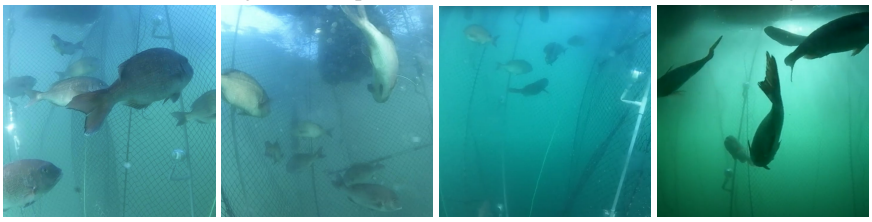


(a) (b)

Figure 8: Body length: (a) Salmon body length. Orange: trout salmon, gray: coho salmon. (b) Yellowtail and red seabream body length. Red marks: Red seabream, yellow marks: yellowtails.

(a) Coho salmons at a depth of 3m in April, 2022 at north fish cage.

(b) Yellowtail amberjack at a depth of 3m in November, 2021 at south fish cage.



(c) Red seabreams at a depth of 2m in November, 2021 at south fish cage.

Figure 9: Examples of recorded video shots of (a) coho salmon, (b) yellowtail amberjack, and (c) red seabreams. For each fish species, from left to right panels show camera views located at R, G, B, and Y respectively.

[3] J Kohbara, I Hidaka, F Matsuoka, T Osada, K Furukawa, M Yamashita, and M Tabata. Self-feeding behavior of yellowtail, seriola quinqueradiata, in net cages: diel and seasonal patterns and influences of environmental factors. *Aquaculture*, 220(1-4):581–594, 2003.

[4] Tomoki Honryo, Michio Kurata, Dario Sandval, Saki Yamao, Amado Cano, and Yoshifumi Sawada. Effect of water temperature and light intensity on swim bladder inflation and growth of red sea bream pagrus major larvae. *Fisheries science*, 84(3):553–562, 2018.

[5] 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.

[6] 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.

[7] 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.

[8] Frode Oppedal, Tim Dempster, and Lars H Stien. Environmental drivers of atlantic salmon behaviour in sea-cages: a review. *Aquaculture*, 311(1-4):1–18, 2011.

[9] 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.

[10] 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.

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

[12] 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.

[13] Horst Bleckmann. 3-d-orientation with the octavolateralis system. *Journal of Physiology-Paris*, 98(1-3):53–65, 2004.

[14] KF Cubitt, S Churchill, D Rowsell, DA Scruton, RS McKinley, et al. 3-dimensional positioning of salmon in commercial sea cages: assessment of a tool for monitoring behaviour. pages 25–33, 2003.

[15] Jens Krause, Jean-Guy J Godin, and David Brown. Phenotypic variability within and between fish shoals. *Ecology*, 77(5):1586–1591, 1996.

[16] Malthe Hvas, Ole Folkedal, and Frode Oppedal. Fish welfare in offshore salmon aquaculture. *Reviews in Aquaculture*, 13(2):836–852, 2021.

[17] Ole Folkedal, Lars Helge Stien, Jonatan Nilsson, Thomas Torgersen, Jan Erik Fosseidengen, and Frode Oppedal. Sea caged atlantic salmon display size-dependent swimming depth. *Aquatic Living Resources*, 25(2):143–149, 2012.

[18] C.A.G. Wiersma and K. Ikeda. Interneurons commanding swimmeret movements in the crayfish, procambarus clarki (girard). *Comparative Biochemistry and Physiology*, 12(4):509–525, 1964.

[19] Henri Korn and Donald S Faber. The mauthner cell half a century later: a neurobiological model for decision-making? *Neuron*, 47(1):13—28, July 2005.

[20] 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.

[21] Charles L. Gallegos and Moore. Factors contributing to water-column light attenuation. *Chesapeake Bay submerged aquatic vegetation water quality and habitat-based requirements and restoration targets: a second technical synthesis*, pages 35 – 55, 2000.

[22] Annick Bricaud, André Morel, Marcel Babin, Karima Allali, and Hervé. Claustre. Variations of light absorption by suspended particles with chlorophyll a concentration in oceanic (case 1) waters: Analysis and implications for bio-optical models. *Journal of Geophysical Research*, 1998.

[23] Michael Solonenko and Curtis D. Mobley. Inherent optical properties of jerlov water types. *Applied optics*, 54 17:5392–401, 2015.

[24] Kevin R Stuart and Mark Drawbridge. The effect of light intensity and green water on survival and growth of cultured larval california yellowtail (seriola lalandi). *Aquaculture*, 321(1-2):152–156, 2011.

[25] John P Abraham, M Baringer, NL Bindoff, T Boyer, LJ Cheng, JA Church, JL Conroy, CM Domingues, JT Fasullo, J Gilson, et al. A review of global ocean temperature observations: Implications for ocean heat content estimates and climate change. *Reviews of Geophysics*, 51(3):450–483, 2013.

[26] Yoshimi Kawai and Akiyoshi Wada. Diurnal sea surface temperature variation and its impact on the atmosphere and ocean: A review. *Journal of oceanography*, 63(5):721–744, 2007.

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

[28] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[29] 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.

[30] Stefano Marras, Shaun S Killen, Jan Lindström, David J McKenzie, John F Steffensen, and Paolo Domenici. Fish swimming in schools save energy regardless of their spatial position. *Behavioral ecology and sociobiology*, 69(2):219–226, 2015.

[31] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[32] Charles L Gallegos and KA Moore. Factors contributing to water-column light attenuation. *Chesapeake bay submerged aquatic vegetation water quality and habitat-based requirements and restoration targets: A second technical synthesis*, 2000.

[33] Annick Bricaud, André Morel, Marcel Babin, Karima Allali, and Hervé Claustre. Variations of light absorption by suspended particles with chlorophyll a concentration in oceanic (case 1) waters: Analysis and implications for bio-optical models. *Journal of Geophysical Research: Oceans*, 103(C13):31033–31044, 1998.

[34] Derya Akkaynak, Tali Treibitz, Tom Shlesinger, Yossi Loya, Raz Tamir, and David Iluz. What is the space of attenuation coefficients in underwater computer vision? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4931–4940, 2017.

[35] Michael G Solonenko and Curtis D Mobley. Inherent optical properties of jerlov water types. *Applied optics*, 54(17):5392–5401, 2015.

[36] Nathan Hoobler. Fast, flexible, physically-based volumetric light scattering. In *Game Developers Conference*, page 608, 2016.

[37] OV Kopelevich. Small-parameter model of optical properties of sea water. *Ocean optics*, 1:208–234, 1983.

[38] Curtis D Mobley. In *Light and Water: Radiative Transfer in Natural Waters*. Academic, 1994.

[39] 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.

[40] 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.

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