VIDEO2SWIMFISH: AN AUTOMATED PIPELINE FOR RECONSTRUCTING CONTROLLABLE FISH MODELS AND BIOLOGICAL LOCOMOTION FROM REAL FISH VIDEOS

Anonymous authors
Paper under double-blind review

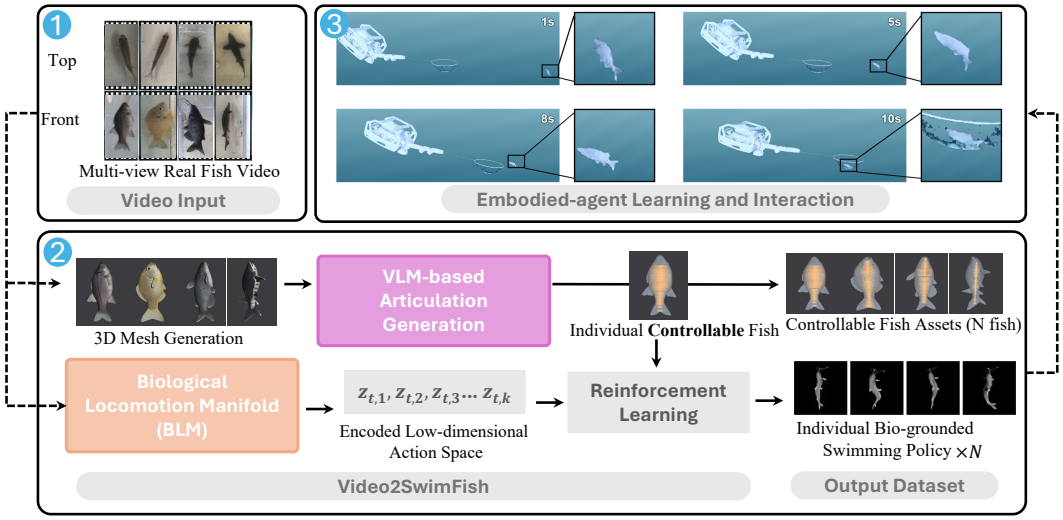

Figure 1: Overview of Video2SwimFish. Given multi-view real fish videos, Video2SwimFish reconstructs individual fish meshes with controllable articulations and derives a Biological Locomotion Manifold (BLM) from real-fish swimming motion. Reinforcement learning then uses the BLM to learn an individual swimming policy for each controllable fish, producing a dataset of biologically grounded, controllable fish assets for downstream embodied-agent learning and interaction.

ABSTRACT

We present **Video2SwimFish**, an automated pipeline and benchmark for building controllable fish assets from real-fish videos for underwater embodied AI. Given synchronized multi-view videos of an individual fish, the pipeline reconstructs a metrically scaled deformable mesh from a VLM-selected canonical frame, generates internal articulation adapted to that individual's morphology through a VLM actor–critic loop, and extracts a Biological Locomotion Manifold (BLM) from the fish's observed midline curvature. The BLM provides a low-dimensional action space bounded by real-fish motion, enabling an individual swimming policy to be learned for each reconstructed fish. We release two paired datasets: synchronized top- and front-view recordings of 120 individual fish across 6 species, and the controllable assets and individual swimming policies derived from them. Because every asset is tied to the animal it came from, the dataset supports a benchmark that evaluates locomotion learning not only on task success but on fidelity to that individual in trajectory shape, body curvature, and tail-beat frequency, across trajectory following, reward-free swimming behavior transfer from video, and a downstream case study in which a simulated BlueROV underwater robot captures one of the assets. We find that task success and locomotion fidelity do not necessarily improve together: the method achieving the highest task completion is not the method achieving the highest locomotion fidelity, and we identify faithful reproduction of individual animal locomotion as an open challenge for the community. Anonymous project website: `https://video2swimfish.vercel.app`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

AI and reinforcement learning are creating growing opportunities for embodied agents to assist with monitoring, inspection, and management in aquaculture environments (FAO, 2024; FAO et al., 2025; Sumana et al., 2026; Oliveira et al., 2021). However, training embodied agents in real-world aquaculture environments is costly, hard to scale (Amundsen et al., 2024), and involves potential risks to animal welfare (Pino et al., 2024). Simulation provides a safer and more scalable alternative. Existing simulation frameworks can already model many of the physical components needed for underwater robot learning, including fluid systems (Todorov et al., 2012), articulated robot dynamics (Makoviychuk et al., 2021), and deformable-body dynamics (Faure et al., 2012; Hu et al., 2019). A major gap, however, is *the lack of realistic, deformable, and controllable fish with a swimming policy that are both grounded in real-world observations, together with standardized tools for evaluating locomotion and biological fidelity in simulation*. Real-fish videos provide a low-cost, non-invasive, and scalable source of biological observations that can capture both fish morphology and swimming behavior(Cui et al., 2025). Thus, we present **Video2SwimFish**, an automated pipeline that reconstructs fish models and learns biological locomotion from real fish videos. The pipeline can be reused by users without specialized biological knowledge to generate bio-grounded, controllable fish from new real-fish videos, each equipped with an individual swimming policy.

Building such a fish from real-world videos presents two main challenges. Reconstructing an articulated 3D fish from video is the first challenge. NeRF-based methods are effective for reconstructing 3D shapes and deformable objects from images and videos (Mildenhall et al., 2020; Pumarola et al., 2021; Yang et al., 2022), but they do not directly produce the internal articulation needed to control a simulated fish and perform poorly when reconstructing fast-moving, highly deformable fish. Recent VLM-based methods such as URDFormer and Articulate-Anything can further turn visual inputs into articulated simulation assets (Chen et al., 2024; Le et al., 2025). However, these methods mainly target rigid articulated objects, whereas a fish is a deformable body whose internal articulation must be inferred and coupled with the reconstructed mesh.

The second challenge is to learn a biologically grounded swimming policy. In order to ensure that the learned policy is grounded in biological locomotion, we therefore train it to follow the swimming motion observed in the real-fish video. Existing approaches can learn locomotion using reinforcement learning (Schulman et al., 2017), imitation learning(Ross et al., 2011), or hand-designed locomotion models such as central pattern generators (CPGs) (Peng et al., 2021; Xie et al., 2019). However, CPGs impose a predefined locomotion pattern rather than being grounded in the motion of each individual fish, while imitation learning typically requires corresponding joint-level demonstrations. In our setting, the joint states underlying the motion in a real-fish video are not directly available, making direct joint-level imitation difficult. Reinforcement learning can learn task-directed locomotion, but task rewards alone do not specify how the fish should swim. For example, the fish could succeed in the task but behave very differently from a real fish.

To address these challenges and enable systematic evaluation, our contributions in Video2SwimFish are fourfold:

1. Video2SwimFish Pipeline: We introduce an automated pipeline that automatically transforms real fish videos into realistic, controllable, and deformable simulated fish assets, each equipped with an individual swimming policy.

2. Biological Locomotion Manifold: We introduce the BLM, a method that extracts and encodes an action space from observed animal motion: it reduces an individual fish's own midline curvature to a low-dimensional manifold bounded by that animal's observed range and rate of deformation, giving reinforcement learning a control space grounded in the individual rather than hand-designed.

3. Datasets: We contribute two tightly coupled datasets. The first is, to our knowledge[1], the largest synchronized orthogonal multi-view fish video dataset designed for individual 3D morphology reconstruction and free-swimming behavior capture, spanning 6 species with 20 individuals per species, serving as the input to our pipeline. The second is a processed simulation dataset of 120 controllable fish assets, each paired with a learned swimming policy, serving as the output. We also provide a detailed data collection protocol so that anyone can capture new real-fish videos and apply

[1]The closest prior work, 3D-ZeF (Pedersen et al., 2020), records several fish of one species per sequence and outputs trajectories rather than individual morphology. Ours contain exactly one identified fish each, across 120 individuals and 6 species.

our pipeline to generate new fish assets.

4. Benchmark: We evaluate state-of-the-art locomotion learning methods across standardized swimming tasks and a biological fidelity evaluation protocol. Our analysis indicates that while existing methods can achieve reasonable task success rates, simultaneously maintaining biological fidelity to the corresponding real fish remains a significant challenge — policies that succeed at the task often deviate substantially from the locomotion patterns of their real-fish counterparts. We thus identify biological fidelity as an open challenge for the community.

2 Related Work

From Video to Controllable 3D Assets. NeRF (Mildenhall et al., 2020) and its dynamic extensions model deformable objects with canonical representations and deformation fields (Pumarola et al., 2021; Park et al., 2021), and BANMo recovers animatable 3D models from casual videos (Yang et al., 2022); none provides the internal articulation needed for control, and all struggle with fast, highly deformable motion. Single-image reconstruction methods (Hong et al., 2024; Tochilkin et al., 2024; Long et al., 2024; Xu et al., 2024) generate 3D assets efficiently, while URDFormer and Articulate-Anything build articulated simulation assets from visual inputs (Chen et al., 2024; Le et al., 2025), but target static objects or rigid articulations. Video2SwimFish instead couples real-fish video reconstruction with articulation generation to produce deformable, controllable fish.

Learning Locomotion from Videos. Imitation-from-observation methods learn behaviors from state or video demonstrations without action labels (Liu et al., 2017; Torabi et al., 2018; Edwards et al., 2019; Raychaudhuri et al., 2021), but require aligned observation spaces and an inverse dynamics model trained in the target environment, whose pseudo-labels compound error when simulated and real dynamics diverge. Reference-motion methods such as DeepMimic and AMP (Peng et al., 2018; 2021) instead combine reference motions with reinforcement learning, but need joint-level references that real-fish videos do not provide. For fish-like robots, CPGs and RL over their parameters generate rhythmic swimming (Ijspeert, 2008; Wang et al., 2012; Zhang et al., 2020; Deng et al., 2024; Rodwell & Tallapragada, 2023), yet those parameters are shared across individuals rather than reflecting how a particular fish moves. These limitations motivate the systematic evaluation of task performance against biological fidelity that Video2SwimFish supports.

Simulation Assets and Embodied-Learning Benchmarks. Large-scale 3D asset datasets (Deitke et al., 2023; Wu et al., 2023) and articulated-object datasets (Mo et al., 2019) supply models for simulation, and platforms such as SAPIEN, MuJoCo, Isaac Gym and Isaac Sim simulate articulated and deformable bodies at scale (Xiang et al., 2020; Todorov et al., 2012; Makoviychuk et al., 2021; Mittal et al., 2023). On these, benchmarks such as AI2-THOR, Habitat, ManiSkill and BEHAVIOR-1K standardize embodied-agent training (Xia et al., 2020; Szot et al., 2021; Gu et al., 2023; Li et al., 2024), and RoboGen generates tasks and scenes with foundation models (Wang et al., 2024). For underwater robotics, FishGym provides articulated fish, fluid–structure interaction and RL (Liu et al., 2022), but its fish are hand-designed rather than grounded in biological observation, and no standardized evaluation of biological fidelity exists. Video2SwimFish fills this gap, linking simulated fish directly to their real-world counterparts.

3 Video2SwimFish

Problem Statement. Given a set of real-fish videos $\mathcal{V} = \{V_i\}_{i=1}^{N}$, each a synchronized top- and front-view recording of one individual, our goal is to produce for each a simulated fish $F_i = (M_i, K_i, \pi_i)$ that can autonomously swim in a physics-based environment: a deformable mesh M_i preserving that fish's morphology, an internal articulation $K_i = (\mathcal{B}_i, \mathcal{J}_i, \Psi_i)$ of rigid bones $\mathcal{B}_i$, joints $\mathcal{J}_i$ with their limits and drive parameters, and an attachment Ψ_i binding mesh to bones so that actuating $\mathcal{J}_i$ deforms M_i, and a swimming policy π_i consistent with the motion observed in V_i. Unlike rigid articulated objects, the bones are internal and the observable body is deformable. Ground-truth kinematics are not available from video, so K_i cannot be optimized directly; its quality is instead reflected in π_i, since a poorly constructed articulation limits the reachable motion space.

Our framework reconstructs M_i and K_i from the video (Figure 2), extracts the observed swimming motion into a low-dimensional Biological Locomotion Manifold (BLM), and uses that manifold as the action space for goal-conditioned reinforcement learning, which trains π_i. Applied across $\mathcal{V}$, this yields a collection of simulated fish, each with its own morphology, articulation, and swimming

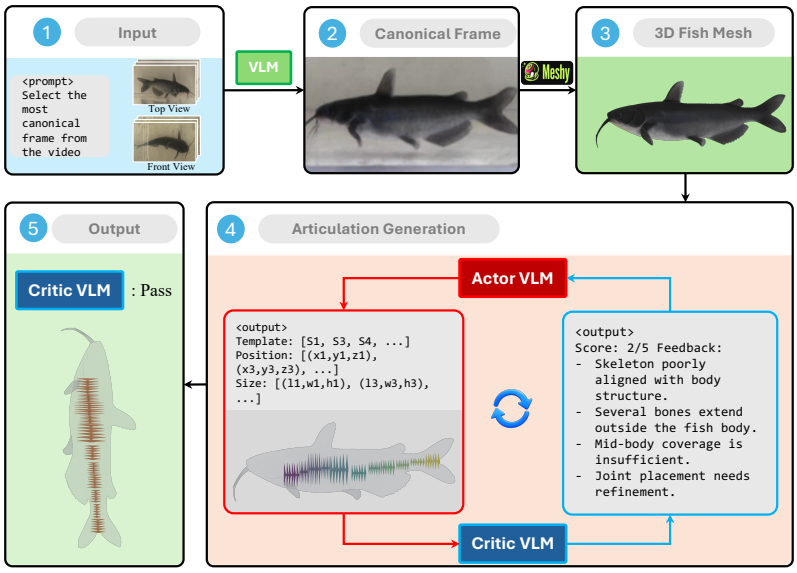

Figure 2: The Video2SwimFish articulation generation pipeline. A VLM selects the most canonical frame from the multi-view video, from which Meshy reconstructs a 3D fish mesh. An actor VLM then proposes a skeleton as a list of bone templates with positions and sizes, and a critic VLM scores it and returns feedback; the two iterate until the critic accepts.

policy.

3.1 Mesh Reconstruction

A fish video captures continuously changing body poses, and different poses can introduce transient body deformations that distort the morphology in a single frame. Since image-to-3D reconstruction operates on a single image, selecting an arbitrary frame may therefore produce a mesh that reflects a transient swimming pose rather than a stable morphology. We therefore first identify a canonical[2] body-pose frame from the video.

For this step, we use a pretrained Qwen3-VL model (Bai et al., 2025) to evaluate the body configuration of each candidate frame and select the frame that best represents the fish's canonical morphology. The selected frame is cropped to the fish region using background subtraction (Zivkovic, 2004) against the per-video median background, so that the image-to-3D system receives the fish rather than the tank environment. The selected frame is then used as the visual reference for subsequent 3D reconstruction.

The selected canonical frame is then given to Meshy, an off-the-shelf image-to-3D system, to reconstruct the three-dimensional fish mesh (Meshy, 2026)[3]. To scale the mesh to the fish's actual size, we estimate each fish's body length from video, calibrated using the known tank dimensions visible in both views. The reconstructed mesh is scaled to this estimated length before the subsequent articulation construction stage. Meshy takes the selected image as the visual reference and returns a 3D mesh. Thereby, we achieve a fish mesh that is consistent with the morphology of the corresponding real fish for the subsequent articulation construction stage. Comparing each mesh's lateral silhouette against the segmented canonical frame gives a mean affine-aligned IoU of 0.88 (Appendix C).

[2]Canonical here means the pose closest to the fish's resting morphology, a standard notion in deformable 3D reconstruction (Pumarola et al., 2021; Park et al., 2021; Yang et al., 2022).

[3]We explored a range of video- and image-based reconstruction methods, including deformable NeRF variants (Pumarola et al., 2021; Park et al., 2021; Yang et al., 2022), feed-forward image-to-3D models (Tochilkin et al., 2024; Long et al., 2024; Xu et al., 2024), and VLM-driven voxel generation. The NeRF-based methods degraded on fast, non-rigid fish motion, the feed-forward models produced fused or truncated fins, and voxel generation was limited by context length to resolutions too coarse for thin structures such as fins. Meshy gave the most usable reconstructions.

3.2 ACTOR–CRITIC AUTO-CONSTRUCTION OF FISH ARTICULATION

A reconstructed fish mesh does not specify how the fish should be controlled. In particular, it does not provide an internal skeleton, joint locations, joint configurations, or the physical parameters required to construct a controllable fish, including joint stiffness, damping, and actuator properties. Manually designing these quantities for every reconstructed fish would be expensive and difficult to scale.

Following recent VLM-based actor–critic approaches for articulated object modeling (Le et al., 2025), we employ a VLM actor–critic framework to automatically construct the internal articulation of each reconstructed fish. Note that this VLM-based actor–critic is distinct from the reinforcement-learning actor–critic used for policy training (Sutton et al., 1998). Here, the actor VLM proposes an articulation configuration, while the critic VLM evaluates its physical plausibility and compatibility with the fish morphology and provides feedback to the actor VLM. This actor–critic process is iterated until the critic VLM determines that the configuration reaches our empirically defined satisfactory score.

In our design, particularly, the actor VLM is a pretrained Qwen3-VL model fine-tuned on a small collection of manually constructed fish models. Our manually constructed dataset contains 74 training examples for fine-tuning the actor VLM. These examples provide the actor VLM with fish-specific knowledge about how an articulated structure should be placed inside a fish mesh.

Rather than generating an arbitrary skeleton directly, we provide the actor VLM with a library of skeleton templates (Appendix G). A template represents a valid bone segment that can be placed inside the fish body. The actor VLM selects templates and determines their locations and sizes to construct the articulation. At each iteration, the actor VLM observes rendered views of the fish mesh together with the current articulation and proposes a modification to the current configuration. The modification may add, remove, reposition, or resize a bone. We adopt a fixed physically validated preset for joint DOFs, limits, and drive parameters shared across all fish. Since the simulated fish is not required to replicate the mechanical constraints of a real fish body, this design gives the reinforcement learning policy a sufficiently rich action space to discover effective swimming gaits. Per-bone mass is derived from the fish's estimated body volume and distributed in proportion to each bone's volume.

Let $K^{(k)}$ denote the articulation at iteration k. The actor VLM predicts

$$a^{(k)} = \mathcal{A}_\phi \left(M, K^{(k)}, f^{(k-1)} \right) \tag{1}$$

where M is the reconstructed fish mesh described in Section 3.1, and $f^{(k-1)}$ is the critic VLM feedback from the previous iteration (empty at $k = 0$). At each iteration, the actor VLM observes three rendered views of the fish mesh with the current articulation highlighted beneath a semi-transparent skin, and receives the measured offset of each bone from the body centerline as quantitative context alongside the critic VLM feedback. The proposed modification produces a new articulation

$$K^{(k+1)} = \mathcal{T} \left(K^{(k)}, a^{(k)} \right) \tag{2}$$

The proposed articulation is then evaluated by the critic VLM. The critic VLM receives the fish mesh and the proposed articulation and provides a score together with feedback describing the problems that should be corrected. The critic VLM feedback can identify issues such as inappropriate bone placement, insufficient coverage of the body, or implausible articulation structure. Formally,

$$\left(r^{(k)}, f^{(k)} \right) = \mathcal{C}_\psi \left(M, K^{(k+1)} \right) \tag{3}$$

where $r^{(k)}$ is the critic VLM score and $f^{(k)}$ is the corresponding feedback.

The actor VLM uses this feedback to update the articulation and proposes the next modification. The loop runs for a fixed budget of K iterations or until the critic VLM accepts. If the actor VLM oscillates between adding and removing the same bone, its sampling temperature is raised to escape the cycle. Upon termination, the highest-scoring articulation across all iterations is selected and passed through a deterministic structural completion step $\mathcal{R}$, which fills internal gaps and extends the bone chain toward the head and tail:

$$K^* = \mathcal{R} \left(K^{(k)} \right) \tag{4}$$

The final articulation K^* specifies the bones, joints, and physical parameters of the fish. The deformable fish mesh is then coupled to this internal articulated structure in the physics simulator. The resulting fish is therefore both deformable and controllable, with its individual morphology determined by the reconstructed mesh and its control structure determined by the articulation auto-construction process.

3.3 BIOLOGICAL LOCOMOTION MANIFOLD (BLM)

Real-fish videos capture swimming as continuous body deformation rather than explicit joint-level motion. In contrast, the reconstructed fish is actuated through its internal joints, making it difficult to directly learn the locomotion from real-fish videos. We therefore introduce a *Biological Locomotion Manifold (BLM)*, a low-dimensional representation of biological body deformation extracted from individual fish swimming videos that enables reinforcement learning to control the fish through biologically grounded motion.

We represent fish body deformation using the curvature of the fish body along its longitudinal axis. At timestep t, the curvature sampled at N body locations is represented as

$$\mathbf{c}_t = [\kappa(\xi_1, t), \ldots, \kappa(\xi_N, t)]^\top \tag{5}$$

where $\xi \in [0, 1]$ denotes the normalized position along the fish body.

We collect curvature profiles from real-fish videos and apply principal component analysis (PCA) (Pearson, 1901) to extract a low-dimensional biological action representation. Let $\boldsymbol{\mu}$ denote the mean curvature profile and $V \in \mathbb{R}^{N\times d}$ denote the first d principal components. A curvature profile can then be approximated as

$$\mathbf{c}_t \approx \boldsymbol{\mu} + V\mathbf{a}_t \tag{6}$$

where $\mathbf{a}_t \in \mathbb{R}^d$ contains the PCA coefficients.

We use the PCA coefficients as the biological action representation. In our main experiments, we retain the first 4 principal components, which capture 99.92% of the curvature variance in our recordings: the remaining modes add almost no expressive range but would contribute action noise in every exploration step. The range and temporal variation of this action representation are derived directly from the real-fish trajectories. Specifically, for each PCA coefficient, we compute its absolute per-step change

$$\delta a_{i,t} = |a_{i,t+1} - a_{i,t}| \tag{7}$$

and define the maximum per-step change as the 99th percentile over the real-fish data:

$$\Delta a_{i,\max} = P_{99}\left(\delta a_{i,t}\right) \tag{8}$$

The resulting vector

$$\Delta\mathbf{a}_{\max} = [\Delta a_{1,\max}, \ldots, \Delta a_{d,\max}]^\top \tag{9}$$

therefore defines a biologically grounded bound on how rapidly each PCA coefficient can change. The coefficient limits $\mathbf{a}_{\max}$ are likewise determined from the observed real-fish trajectories.

3.4 DATA COLLECTION

Multi-View Fish Videos. We collect synchronized multi-view videos of individual fish in a controlled aquaculture environment using two FLIR Blackfly S BFS-U3-200S6C-C cameras equipped with 12 mm C-mount lenses. One camera is mounted top-down 0.82 m above the water surface and one front-facing 0.78 m from the front wall, both hardware-synchronized via external trigger. Fish are placed one at a time in a transparent tank (76.2 cm × 29.9 cm × 44.5 cm, water depth 42 cm), so that each recording contains a single individual. Camera intrinsics and the inter-camera geometry are calibrated using a ChArUco board prior to each session. Each fish is recorded individually at 32 fps 2736 × 1824 resolution for two and a half minutes per trial. Our dataset comprises 6 species spanning 120 individual fish, per-species details are given in Appendix A.

Fish Skeleton Dataset. To fine-tune the actor VLM, we manually construct fourteen controllable fish models containing 85 bones in total; twelve of them yield 74 training examples, with two held out for validation. Drawing from publicly available biological references on fish skeletal anatomy and body mechanics (Lauder & Tangorra, 2015; Videler, 1993), we design bone mesh templates and place them inside each fish mesh, ensuring alignment with the longitudinal body midline, no

interpenetration with the mesh surface, and continuous skeletal coverage from head to caudal fin. Each model includes a complete joint configuration and physical parameters informed by biological studies of fish intervertebral joint mechanics and body stiffness (Long Jr & Nipper, 1996; Jimenez et al., 2023; Lauder & Tangorra, 2015), serving as fish-specific fine-tuning examples for the actor.

4 BENCHMARK

We evaluate Video2SwimFish from three perspectives. First, whether reconstructed fish can learn to reproduce the trajectories of their real counterparts (Task 1: Trajectory Following). Second, whether the swimming behavior observed in video can be transferred to the reconstructed fish without any additional reward (Task 2: Free Swimming from Video). Third, whether the resulting fish can serve as an interactive component of the environment for training downstream embodied agents (Case Study: BlueROV Underwater Robot Capture). Biological fidelity is measured alongside task performance in Task 1 and Task 2.

4.1 EXPERIMENTAL SETUP

Fish. All experiments use assets produced by the pipeline of Section 3 without any per-fish adjustment. For each species we select two individuals (twelve fish) for Task 1, one per species (six fish) for Task 2, and two fish for the Case Study. Every fish uses the same actuator preset (3-DoF joints, $\pm 45^\circ$, PD stiffness 120) and the same physics step ($1/120$ s, control at 30 Hz). Fluid forces come from a quasi-steady resistive panel model applied to the deformable skin, shared by every method and every fish; the formulation and its coefficients are given in Appendix G.1.

BLM control. The BLM policy acts in the manifold of Section 3.3 rather than in joint space. With $\hat{\mathbf{a}}_t \in [-1, 1]^d$ the normalized policy action, the coefficients are updated incrementally,

$$\mathbf{a}_{t+1} = \mathrm{clip}(\mathbf{a}_t + \hat{\mathbf{a}}_t \odot \Delta\mathbf{a}_{\max},\ -\mathbf{a}_{\max},\ \mathbf{a}_{\max}) \tag{10}$$

which keeps the rate of change of every coefficient inside the range observed in the fish's own video. The target curvature $\boldsymbol{\mu} + V\mathbf{a}_{t+1}$ is mapped to joint targets by a fish-specific curvature-to-joint controller (Appendix D), identified once per fish in simulation by exciting each lateral joint and regressing the resulting body curvature (damped least-squares inverse of the empirical Jacobian (Wampler, 1986; Nakamura & Hanafusa, 1986; Tikhonov, 1963)).

Training protocol. Every run is trained until convergence: a run stops when its completion rate (Task 1) or capture success (Case Study) stays above 0.90 (0.85) for three consecutive epochs, or when its best value has not improved by more than 0.02 over the last 20 epochs after at least 40 epochs.

4.2 TASK 1: TRAJECTORY FOLLOWING

In Task 1, each episode samples a 5 s window of the fish's *own* real trajectory (position and heading from the top view). Only the shape of the path matters, not where in the tank it happened, so the window is moved and turned until its first frame sits at the simulated fish's starting pose, and the fish swims along it from there. Progress is measured by a phase variable s_t, the arc length of the path point nearest the fish, so that the fish is scored on how far along the path it has got rather than on where it should be at time t; a fish that swims the right shape slowly is not penalized for being slow. The reward tracks a look-ahead point 0.3 BL further along the path, together with heading alignment and the per-step increase in s_t. An episode is completed when s_t reaches the end of the path.

We score each roll-out on how well it follows the path and on how closely its body motion matches the real fish. *Completion rate*: fraction of episodes whose phase reaches the path end. *Fréchet distance*: discrete Fréchet distance (Eiter et al., 1994) between the executed and reference paths, measured in the body frame of the episode start and expressed in body lengths.[4] *Phase monotonicity*: fraction of steps in which s_t does not decrease, which detects policies that drift backwards or circle rather than advancing. Biological fidelity is measured on the same roll-outs: the 1-Wasserstein distance between the distributions of simulated and real body curvature $\kappa \cdot \mathrm{BL}$ over 20 body stations,

[4]Body length (BL) is the fish's snout-to-tail-base length; distances and speeds are conventionally normalized by it so that fish of different sizes can be compared (Videler, 1993).

the dominant body-wave frequency and its deviation from the real fish, and the swimming speed (BL/s) and its deviation.

Table 1: Task 1 trajectory following on twelve fish (two per species): completion rate, Fréchet distance (BL), curvature Wasserstein distance $W_1(\kappa)$, dominant frequency error Δf (Hz) and speed error Δv (BL/s); mean over fish. Best per column in bold.

Method	Completion ↑	Fréchet ↓	$W_1(\kappa)$ ↓	Δf ↓	Δv ↓
Joint RL	0.70	0.48	1.43	1.61	0.52
Joint RL + AMP	0.51	**0.34**	**0.78**	2.11	0.64
CPG + RL	0.67	0.58	1.16	0.70	0.38
BCO + RL	0.47	0.56	1.89	0.81	0.44
BLM + RL (Ours)	**0.89**	0.65	1.01	**0.56**	**0.33**

Table 1 shows no method dominating, and the two groups of metrics pull in opposite directions; Figure 3 shows the five methods on the same reference trajectory. BLM+RL completes 0.89 of episodes against 0.70 for the next best method, and has the lowest frequency and speed errors, but the largest Fréchet distance in the table. Joint RL+AMP is its mirror image: the most accurate path (0.34 BL) and the closest curvature distribution ($W_1(\kappa)$ 0.78), but it finishes only half its episodes and has the worst tail-beat frequency of all five methods. A policy can therefore track a path closely while beating its tail at twice the animal's rate, and a benchmark that reported only completion, or

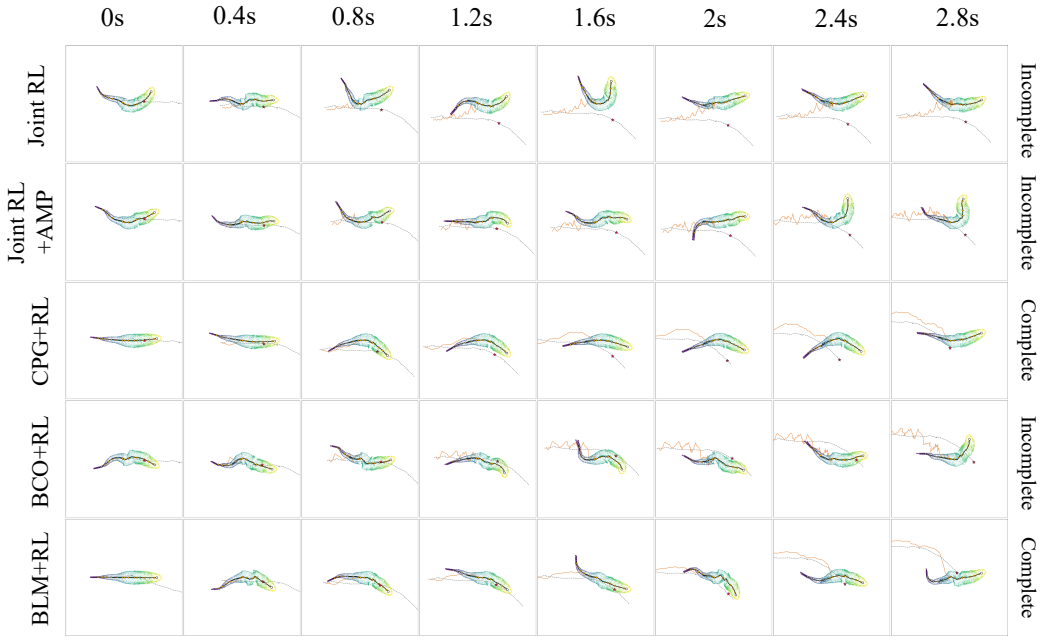

Figure 3: Qualitative comparison of five methods following the same reference trajectory in Task 1. Snapshots are shown at 0.4-s intervals. The dashed black curve denotes the reference trajectory, the orange trail shows the simulated fish's executed path, and the star marks the real fish's position on the reference trajectory at each time step. Completion status is indicated on the right.

4.3 Task 2: Free Swimming from Video

Task 2 benchmarks imitation-from-observation methods. We therefore evaluate biological fidelity only, asking whether the motion a policy produces is characteristic of that fish rather than whether it retraces one particular recording.

We compare two methods, BLM+IL and BCO (Torabi et al., 2018), which differ only in the space they clone in. BLM+IL applies behavior cloning directly in the manifold. Its demonstrations are midline curvature profiles, extracted from the top-view video by segmentation alone (Appendix B.7); since the BLM coefficients are the action space, this sequence is already an action sequence and no inverse dynamics model is needed. BCO clones in joint space instead, and must first train an inverse

dynamics model on the fish's own random interaction in simulation to pseudo-label the video. Both optimize the same behavior cloning objective on the same video, with early stopping on a held-out validation split.

Table 2: Task 2 free swimming from video on six fish (one per species): 20 real initial states, 5 s roll-outs, no reward. Best per column in bold.

Method	Fréchet ↓	Speed err. ↓	Heading stab. ↑	$W_1(\kappa)$ ↓	Δf ↓
BCO	2.79	0.39	0.56	1.63	**0.012**
BLM + IL	**2.08**	**0.11**	0.56	**0.74**	0.038

Table 2 shows BLM+IL ahead on every fidelity metric except tail-beat frequency: it halves the curvature distance (0.74 vs. 1.63) and cuts the speed error by more than a factor of three (0.11 vs. 0.39), while BCO matches the real frequency more closely (0.012 vs. 0.038). Cloning in the manifold avoids the inverse dynamics model whose pseudo-labels degrade when simulated and real dynamics diverge, and the gap is largest on exactly the quantity the manifold encodes, body curvature.

4.4 Case Study: BlueROV Capture

To show that the produced fish are usable as interactive assets, a BlueROV2 with a 6-DoF manipulator and a rigid basket is trained with PPO to capture a free-swimming fish whose motion is generated by its own Task-2 BLM+IL policy. Success requires the fish inside the basket with the opening facing up. We train capture policies for two fish (a channel catfish and a lake sturgeon) and report the capture success rate and epochs to convergence (Table 3).

Table 3: Case Study: ROV capture of video-driven fish.

Prey	Capture success rate↑	Epochs to convergence
Channel catfish 002	0.988	44
Lake sturgeon 016	0.992	49

5 Discussion, Limitations, and Future Work

Video2SwimFish provides a reusable pipeline for transforming real-fish videos into controllable simulated fish, with benchmarks for both task performance and locomotion fidelity. Our results show that task success and locomotion fidelity do not necessarily improve together: the method with the highest task completion is not necessarily the most faithful to the corresponding real fish. The observation-derived Biological Locomotion Manifold (BLM) improves several fidelity measures while retaining strong task performance, although no method consistently dominates across all metrics. These results motivate evaluating simulated animal agents by both downstream task performance and correspondence to the locomotion of the individual animal from which the asset was derived.

Limitations. Our fish were recorded in laboratory tanks, whose limited size may constrain natural swimming behavior. Simulations use an approximate quasi-steady hydrodynamic model rather than full fluid–structure interaction; thus, agreement in trajectory, curvature, speed, and tail-beat frequency does not establish comprehensive biomechanical or physiological validity. Locomotion experiments use selected reconstructed fish with sufficient clean video and assets passing the stated validity checks; performance across all 120 assets therefore remains unestablished. The BlueROV capture study is a two-fish simulation case study rather than a comprehensive real-world evaluation. Our approved animal-use protocol covered observational video collection and did not authorize robot–fish interaction; such experiments would require separate approval, an appropriate test site, reliable underwater hardware, and additional animal-welfare considerations. Finally, BLM does not consistently outperform all baselines, indicating that constraining learning by observed individual locomotion alone does not fully resolve the tradeoff between task competence and biological fidelity.

Future Work. We will investigate more realistic fluid–structure interaction, broader swimming conditions, and more systematic evaluation across individuals, species, and behaviors, including reconstruction and simulation stability. We will compare global, species-specific, and individual-specific locomotion representations and develop objectives that better balance task performance

with multiple dimensions of biological fidelity. Subject to appropriate animal-use approval and safety protocols, we will also investigate sim-to-real transfer and robot–fish interaction in controlled environments. The released datasets, simulation assets, and benchmark metrics provide a foundation for these directions and for studying individual locomotion as an explicit constraint on embodied-agent learning.

AI Use Statement. In this work, we used generative AI tools to generate synthetic datasets: a pretrained vision–language model (Qwen3-VL) and an off-the-shelf image-to-3D system (Meshy) are components of the proposed pipeline, and the released fish assets are their output, as described in Section 3. We have not used generative AI tools for any other task requiring disclosure under the ICLR 2027 AI policy. We have reviewed all AI-assisted work: every generated asset was inspected by the authors before release. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

Ethics statement

Animal subjects. This work involves video recording of live fish. All procedures were reviewed and approved by our institution's Institutional Animal Care and Use Committee (IACUC) prior to data collection, under an approved protocol covering observational video-based study of fish for aquaculture research [protocol number withheld for anonymity; will be provided in the camera-ready version]. Our study is purely observational: fish were recorded while swimming freely in a holding tank, and no invasive procedure, restraint, tagging, or experimental manipulation was performed at any point. No animals were euthanized for this study. Recording sessions were kept short, water quality and lighting were controlled throughout, and animals were returned to their holding conditions immediately after each session. We note that reducing reliance on live-animal experimentation is itself part of the motivation for this work: the resulting simulated fish are intended to let embodied agents be trained without repeated interaction with real animals.

Dataset release. We release both the multi-view fish video dataset and the derived simulation assets at `https://huggingface.co/datasets/video2swimfish/video2swimfish-dataset`. The recordings contain no human subjects and no personally identifiable information. The datasets are released under the CC BY-NC 4.0 license for non-commercial research use. The data collection protocol is documented in Section 3.4 so that the conditions under which the data were obtained are transparent to users; anyone reproducing our collection procedure with live animals should obtain their own institutional animal-ethics approval.

Potential impact. The intended application is training embodied agents for monitoring and management tasks in aquaculture. We note that agents trained to interact with or capture fish could, if deployed without care, be used in ways that adversely affect animal welfare, and we encourage downstream users to evaluate welfare implications before real-world deployment.

Reproducibility statement

We release the synchronized multi-view videos of all 120 fish and the 120 derived simulation assets, each with its mesh, articulation and trained swimming policy, at `https://huggingface.co/datasets/video2swimfish/video2swimfish-dataset`. Section 3.4 describes the recording setup, including cameras, tank dimensions and calibration, and Appendix A gives per-species details, so that new videos can be collected under the same conditions. Section 3 describes each stage of the pipeline, and Appendix G describes the skeleton templates and the hand-built fish models used to fine-tune the actor VLM. The pipeline relies on an open-weight VLM, Qwen3-VL (Bai et al., 2025), and a commercial image-to-3D service, Meshy (Meshy, 2026), whose model may change over time. Because of this, and because the actor–critic loop samples stochastically, re-running the pipeline may not reproduce identical assets; all benchmark results are therefore computed on, and reproducible from, the released assets. Section 4.1 gives the simulation settings (actuator preset, physics and control rates, hydrodynamic model and coefficients), the BLM controller and the training and stopping criteria, with the curvature-to-joint controller detailed in Appendix D. Sections 4.2–4.4 define each task and metric, and Appendix B.7 describes how midline curvature is extracted from video. Anonymous source code for the pipeline, simulation environment, benchmark tasks and baselines can be found at `https://anonymous.4open.science/r/Anonymous-Video2SwimFish-538D/README.md`.

REFERENCES

Herman B Amundsen, Marios Xanthidis, Martin Føre, Sveinung J Ohrem, and Eleni Kelasidi. Aquaculture field robotics: Applications, lessons learned and future prospects. *arXiv preprint arXiv:2404.12995*, 2024.

Shuai Bai, Yuxuan Cai, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

Meng Cui, Xubo Liu, Haohe Liu, Jinzheng Zhao, Daoliang Li, and Wenwu Wang. Fish tracking, counting, and behaviour analysis in digital aquaculture: a comprehensive survey. *Reviews in Aquaculture*, 17(1):e13001, 2025.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13142–13153, 2023.

Hankun Deng, Donghao Li, Colin Nitroy, Andrew Wertz, Shashank Priya, and Bo Cheng. Robot motor learning shows emergence of frequency-modulated, robust swimming with an invariant strouhal number. *Journal of The Royal Society Interface*, 21(212):20240036, 2024.

Ashley D. Edwards, Himanshu Sahni, Yannick Schroecker, and Charles L. Isbell. Imitating latent policies from observation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Thomas Eiter, Heikki Mannila, et al. Computing discrete fréchet distance. 1994.

FAO. The state of world fisheries and aquaculture 2024 – blue transformation in action. Technical report, Food and Agriculture Organization of the United Nations, Rome, 2024.

Food FAO et al. Guidelines for sustainable aquaculture, 2025.

François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, et al. Sofa: A multi-model framework for interactive physical simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer, 2012.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2024.

Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B. Tenenbaum, William T. Freeman, Jiajun Wu, Wojciech Matusik, and Eftychios Sifakis. Chainqueen: A real-time differentiable physical simulator for soft robotics. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8876–8882. IEEE, 2019. doi: 10.1109/ICRA.2019.8794193.

Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

Yordano E Jimenez, Kelsey N Lucas, John H Long Jr, and Eric D Tytell. Flexibility is a hidden axis of biomechanical diversity in fishes. *Journal of Experimental Biology*, 226(Suppl_1):jeb245308, 2023.

George V Lauder and James L Tangorra. Fish locomotion: biology and robotics of body and fin-based movements. In *Robot fish: bio-inspired fishlike underwater robots*, pp. 25–49. Springer, 2015.

Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic modeling of articulated objects via a vision-language foundation model. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=s3FTX4Ay55.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. BEHAVIOR-1K: A human-centered, embodied AI benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024.

Wenji Liu, Kai Bai, Xuming He, Shuran Song, Changxi Zheng, and Xiaopei Liu. Fishgym: A high-performance physics-based simulation framework for underwater robot learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6268–6275. IEEE, 2022.

YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *arXiv preprint arXiv:1707.03374*, 2017.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, and Wenping Wang. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9970–9980, 2024.

John H Long Jr and Karen S Nipper. The importance of body stiffness in undulatory propulsion. *American Zoologist*, 36(6):678–694, 1996.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Meshy. Image to 3D: Convert photos into 3D models. https://docs.meshy.ai/en/webapp/image-to-3d, 2026. Accessed: 2026-08-16.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, pp. 405–421, 2020.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.

Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 909–918, 2019.

Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. 1986.

Luiz FP Oliveira, António P Moreira, and Manuel F Silva. Advances in agriculture robotics: A state-of-the-art review and challenges ahead. *Robotics*, 10(2):52, 2021.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, and Thomas B Moeslund. 3d-zef: A 3d zebrafish tracking benchmark dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2423–2433. IEEE, 2020.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics*, 37(4):143:1–143:14, 2018. doi: 10.1145/3197517.3201311.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics*, 40 (4):1–20, 2021. doi: 10.1145/3450626.3459670.

Andrea Pino, Rosario Vidal, Elisabeth Tormos, José Miguel Cerdà-Reverter, Raúl Marín Prades, and Pedro J Sanz. Towards fish welfare in the presence of robots: Zebrafish case. *Journal of Marine Science and Engineering*, 12(6):932, 2024.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10318–10327, 2021.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020.

Dripta S. Raychaudhuri, Sujoy Paul, Jeroen van Baar, and Amit K. Roy-Chowdhury. Cross-domain imitation from observations. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Colin Rodwell and Phanindra Tallapragada. Physics-informed reinforcement learning for motion control of a fish-like swimming robot. *Scientific reports*, 13(1):10754, 2023.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

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

Sahr Lamin Sumana, Xiaojun Jing, Honghui Hu, Chengfeng Zhang, Mukhtar Muhammad Abdulla-teef, Eric Komba Foyoh Mani, Xuefeng Wu, Abubakar Shuaibu, and Shengyan Su. Embodied ai for aquaculture monitoring: From stationary sensing to mobile robotic observation. *Aquacultural Engineering*, pp. 102766, 2026.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021.

Geoffrey Taylor. Analysis of the swimming of long and narrow animals. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pp. 158–183, 1952.

Andrei N Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Sov Dok*, 4:1035–1038, 1963.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4950–4957, 2018. doi: 10.24963/ijcai.2018/687.

John J Videler. *Fish swimming*, volume 10. Springer Science & Business Media, 1993.

Charles W Wampler. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1): 93–101, 1986.

Ming Wang, JunZhi Yu, Min Tan, and JianWei Zhang. Multimodal swimming control of a robotic fish with pectoral fins using a cpg network. *Chinese science bulletin*, 57(10):1209–1216, 2012.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 51936–51983, 2024.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–814, 2023.

Fei Xia, William B. Shen, Chengshu Li, Priya Kasimbeg, Micael Edmond Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2): 713–720, 2020.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11097–11107, 2020.

Fengran Xie, Yong Zhong, Ruxu Du, and Zheng Li. Central pattern generator (cpg) control of a biomimetic robot fish for multimodal swimming. *Journal of Bionic Engineering*, 16(2):222–234, 2019.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

Gengshan Yang, Minh Vo, Natalia Neverova, Deva Ramanan, Andrea Vedaldi, and Hanbyul Joo. Banmo: Building animatable 3d neural models from many casual videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2863–2873, 2022.

KS Yeo, SJ Ang, and C Shu. Simulation of fish swimming and manoeuvring by an svd-gfd method on a hybrid meshfree-cartesian grid. *Computers & Fluids*, 39(3):403–430, 2010.

Tianhao Zhang, Runyu Tian, Chen Wang, and Guangming Xie. Path-following control of fish-like robots: A deep reinforcement learning approach. *IFAC-PapersOnLine*, 53(2):8163–8168, 2020.

Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 2, pp. 28–31. IEEE, 2004.

A Dataset Details

Source videos. Each fish is recorded in a glass tank from two hardware-synchronized views (front and top) at 32 fps; all behavior signals are resampled to 30 Hz, the control rate of the simulated fish. Every video contains exactly one fish; a second silhouette that occasionally appears is the reflection in the tank wall and is removed by the mesh critic (Appendix B.2). The dataset contains 6 species $\times$ 20 individuals $=$ 120 fish (Table 4).

Metric scale. Pixel-to-meter scale is calibrated once per species from the known 30-inch front face of the tank (88.5, 89.8, 87.8, 89.4, 88.7 and 89.0 px/in for channel catfish, lake sturgeon, bluegill, white bass, brook trout and brown trout; 104 px/in across the top view). Body length is the median silhouette length over ≥ 150 valid frames; frames outside 0.4–$2.5\times$ the running median are rejected. When a fish has too few valid frames, the species median length is used and flagged (`size_source`).

Table 4: Dataset summary. Coverage is the fraction of body length spanned by the bone chain; alignment is the mean angle between each bone axis and the local body centerline.

Species	n	length [cm]	bones (med.)	coverage	alignment	own PCA basis
Channel catfish	20	12.6–31.1 (med. 20.4)	14.5	74%	10.9°	20/20
Lake sturgeon	20	4.6–18.9 (med. 10.0)	15	72%	8.7°	18/20
Bluegill	20	1.6–9.5 (med. 6.8)	15	74%	11.1°	18/20
White bass	20	4.3–12.9 (med. 6.3)	15	74%	10.6°	17/20
Brook Trout	20	5.3–16.7 (med. 10.2)	14	73%	10.0°	18/20
Brown Trout	20	4.7–12.1 (med. 9.3)	15	75%	11.3°	17/20

B Reconstruction Pipeline

All 120 assets were produced by the same automated pipeline (`scripts/run_dataset_v2sf.py`); no per-fish manual editing is involved. The stages are run in dependency order and each stage is resumable.

B.1 Canonical frame and crop

A VLM (Qwen3-VL-32B-Instruct) selects, from the front-view video, one frame in which the fish is fully visible and laterally presented. A background-subtraction crop isolates the fish.

B.2 Image-to-3D and mesh critic

The crop is sent to the Meshy-6 image-to-3D API. The returned mesh is validated by a VLM mesh critic (same model) on four rendered views, rejecting meshes that are not a fish, are truncated, contain more than one fish, or show the tank rather than the animal. A deterministic component cleaner then welds coincident vertices, splits connected components, and removes any secondary component whose oriented bounding box is disjoint from, larger than 10% outside of, or smaller than 15% of the area of the main body (this removes the reflection fish that the image model sometimes hallucinates as a second body). A mesh that fails the critic is resubmitted once with a corrected crop.

B.3 Canonicalization

The mesh is leveled by aligning its first principal axis to $+x$ (head at $+x$), scaled to the measured body length, and re-centered so the bounding-box center sits at the origin.

B.4 Skeleton construction: VLM actor–critic with geometric regularization

Articulation is produced by an actor–critic loop over rigid box bones:

- **Actor.** Qwen3-VL-32B-Instruct fine-tuned with LoRA (rank 8, α as in `train_actor_lora.py`, 4 epochs, learning rate 10^{-4}) on 14 expert-annotated fish skeletons. Given three orthographic renders of the current skeleton inside the semi-transparent skin, it emits one edit per step (`add`, `remove`, `reposition`, `resize`) with a position and size in meters.
- **Geometric verifier.** Every proposed bone is shrunk anisotropically (per axis) until all its vertices lie inside the skin, and snapped to the local body centerline.
- **Critic.** The base VLM scores the skeleton from 1 (very poor) to 5 (excellent) from the same three views plus measured midline facts (offset of each bone from the centerline, coverage, gaps); a skeleton is accepted at score ≥ 4. The loop runs for at most 6 iterations. Deterministic decoding (temperature 0) can fall into add/remove 2-cycles; after 4 alternating iterations the actor temperature is raised to 0.7 to escape.
- **Structural completion $\mathcal{R}$.** The accepted (or best-scoring) skeleton is passed through the deterministic structural completion step of Sec. 3.2: bones are snapped to a thickness-weighted body centerline, overlapping bones are merged, inner gaps wider than 2% of body length are filled with box bones of the same section, and the chain is extended toward the skull and the caudal-fin base (the points where the cross-section falls below 15% / 3% of its maximum), with a cap of 22 bones. Every new or moved bone goes through the same containment shrink, and the critic re-scores the result.

The actor typically places 5–6 bones; after regularization the exported chains have 12–19 bones (median 15), 1755 bones in total. The final critic score is 5 for 114/120 fish.

B.5 Export and Stage-1 verification

Bones become rigid cylinders/boxes connected by 3-DoF D6 joints ($\pm 45^\circ$ per axis, PD position drives, stiffness $k_p = 120$, damping 6); the skin becomes a PhysX FEM deformable body attached to the bones. Each asset is checked for (i) containment: no more than 5 bone vertices outside the skin and penetration < 2 mm; (ii) coverage; and (iii) alignment: mean bone–centerline angle $< 15^\circ$, max $< 30^\circ$. 87/120 assets pass the strict containment test; the remainder have 1–5 vertices marginally outside (one has 27). Coverage is bounded by design: the expert skeletons used for training span only 38–64% of body length (no bones in the skull or the caudal fin), and the regularized chains reach 64–83%.

B.6 Hydrodynamics

A panel model replaces the CFD: the FEM skin is partitioned into per-bone slices; each slice receives quadratic normal drag ($C_{d,\perp} = 1.0$, the source of tail thrust) and skin friction ($C_{d,\parallel} = 0.01$), applied as an external wrench every physics step. A per-bone impulse limiter (fraction 0.5 of the bone momentum) and a body-speed cap (1.5 BL/s) prevent the explicit-integration runaway of quadratic drag.

B.7 Behavior extraction

The top-view video is segmented (ROI-masked background subtraction, area threshold 25 px, morphological opening 3 px, Gaussian $\sigma = 7$), the midline is extracted and resampled to 20 arc-length stations, and the signed curvature $\kappa \cdot \mathrm{BL}$ is computed per frame. Frames with $\kappa \cdot \mathrm{BL} > 4$ at any station are rejected as segmentation noise. A per-fish PCA basis (4 modes) is fitted on the clean frames; fish with fewer than 100 clean frames (7/120) use a species-pooled basis. The same frames provide the reference trajectories (5 s windows, re-rooted at the window start) used by the trajectory-following task and the biofidelity metrics.

C Mesh Fidelity: Silhouette IoU Against the Canonical Frame

To quantify how faithfully the image-to-3D stage (Sec. B.2) reproduces the animal it was given, we compare the silhouette of each generated mesh with the silhouette of the canonical input frame it was generated from.

Table 5: Silhouette IoU between the generated mesh (orthographic lateral render) and its canonical input frame for the two most cleanly segmented fish per species. Similarity: 4-DOF alignment with the head end matched; affine: 6-DOF refinement; body only: both silhouettes with fins morphologically stripped before the affine IoU.

Species	Fish	IoU (similarity)	IoU (affine)	IoU (body only)
bluegill	013	0.85	0.88	0.88
bluegill	019	0.80	0.82	0.81
catfish	002	0.86	0.90	0.89
catfish	013	0.77	0.84	0.78
lake sturgeon	002	0.88	0.91	0.93
lake sturgeon	006	0.78	0.85	0.84
white bass	004	0.87	0.89	0.89
white bass	008	0.89	0.93	0.93
mean		0.84	0.88	0.87

Protocol. The mesh is rendered with an orthographic camera on the lateral axis (Blender Workbench, 1000^2 px, uniform background), and its silhouette is the set of pixels that differ from the background, with interior holes filled. The photo silhouette is obtained with U^2-Net (Qin et al., 2020) on the canonical crop; the alpha matte is thresholded at $30/255$ (a threshold of 128 removes the translucent caudal and pectoral fins), closed with a 7 px kernel, reduced to its largest connected component, and hole-filled. The two silhouettes are brought into a common frame by a similarity transform computed from image moments (centroid, principal axis, extent along the axis, i.e. 4 DOF), with the head end matched explicitly: every render has the head at $-x$ by construction (Sec. B), and the head side of each photo was labelled by hand, since automatic width-profile rules confused the head with the caudal fan on catfish and sturgeon. The dorsal side is chosen as the better of the two vertical flips. On top of the similarity alignment we report a 6-DOF affine refinement (anisotropic scale, rotation, shear, translation; Powell search maximising IoU), which absorbs the mild perspective of the photograph. All alignments are rigid or affine in 2-D; no non-rigid warping is used, so the numbers are an upper bound only under a global transform.

Segmentation quality. The left column of Fig. 4 shows that a visible share of the disagreement originates in the photo silhouette rather than in the mesh. Thin and translucent structures are the main casualty: the dorsal and anal fins of both bluegill are largely absorbed into the background, the catfish barbels survive only as a broken filament, and in lake sturgeon #006 the tank floor is close enough in intensity to the animal that part of the substrate is retained along the ventral edge. These are properties of a single video frame shot through a tank wall under diffuse lighting, not of the reconstruction: the mesh renders in the middle column carry the fins that the segmentation drops, so those pixels are counted as mesh-only (blue) even where the mesh is correct. The reported IoU is therefore a lower bound on reconstruction accuracy.

Clean subset. The canonical crops are single video frames taken through the tank wall, so their segmentation quality varies; some crops include part of the tank floor, some fins are lost to motion blur. Table 5 and Fig. 4 therefore report the two most cleanly segmented fish per species, chosen by inspecting the U^2-Net outlines before any IoU was computed. The affine IoU is 0.88 on average and reaches 0.93 (white bass #008) and 0.91 (lake sturgeon #002). Stripping the fins from both silhouettes with a morphological opening (kernel 6% of the body length) changes the mean by less than 0.01, i.e. the residual disagreement is not concentrated in the fins alone; the overlays in Fig. 4 show that it comes from the pose of the pectoral and pelvic fins, an erect versus folded dorsal fin, and a slight body bend in the photograph, none of which a global 2-D transform can remove. The affine parameters found are small (rotation $< 2^\circ$, scales 0.93–1.10, shear < 0.12), so the refinement is a genuine alignment rather than a distortion of the mesh.

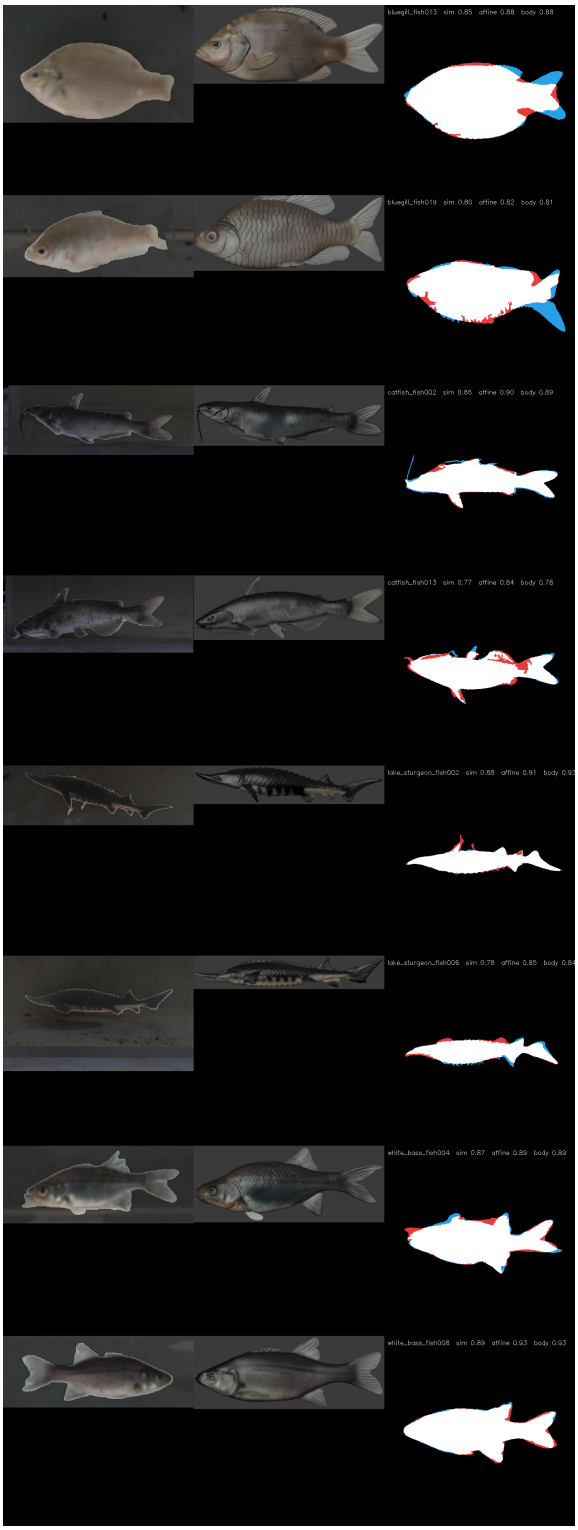

Figure 4: Mesh-versus-photo silhouettes for the eight fish of Table 5. Each panel shows the canonical crop with its U^2-Net segmentation (left), the lateral render of the generated mesh (middle), and the two silhouettes after head-matched affine alignment (right; white: both, red: photo only, blue: mesh only). The remaining disagreement sits in the fin pose and a slight body bend of the photographed animal.

D SIMULATION AND CONTROL

Isaac Sim 5.1 / Isaac Lab (PhysX, GPU). Physics step $1/120$ s, control at 30 Hz (decimation 4), zero gravity (neutral buoyancy), FEM material Young's modulus 10^5 Pa, Poisson ratio 0.45, elasticity damping 0.05. 256 parallel environments per run. A blow-up guard terminates an episode when any joint velocity exceeds 200 rad/s or the FEM nodal velocity limit is exceeded; such episodes count toward the blow-up rate and never as a completion.

Action spaces. *Joint*: PD position targets for all $3(n_b - 1)$ joint DoF, scaled to $\pm 45°$. ***BLM***: coefficients of the per-fish curvature PCA basis, decoded to joint targets through the in-simulation calibration matrix Φ (20 stations $\times$ DoF), identified once per fish by exciting each lateral DoF at amplitude 0.5 and regressing the resulting curvature. *CPG*: a traveling-wave generator with amplitude envelope, frequency and phase offsets fitted to the fish's own curvature (parameters in `cpg_params.npz`); the policy outputs $[\Delta A, \Delta f, \Delta b]$.

E BENCHMARK TASKS

E.1 TASK 1: TRAJECTORY FOLLOWING

Each episode samples a 5 s window of the fish's own real trajectory (position and heading), re-rooted at the fish's reset pose. The phase s_t is the arc-length of the nearest path point to the fish; the look-ahead target sits 0.3 BL ahead. The episode ends when the phase reaches the end of the path (completion), on blow-up, or after 60 s. Reward terms are distance to the look-ahead point, heading alignment and a progress bonus. Metrics (logged every epoch on 16 tracked environments):

- `completion_rate`: fraction of episodes whose phase reached the path end *without a blow-up*;
- `frechet_dist_bl`: discrete Fréchet distance between the executed and reference paths in body lengths (blow-up episodes excluded);
- `wasserstein_curvature`: 1-Wasserstein distance between the distributions of simulated and real $\kappa \cdot \mathrm{BL}$ over all stations, plus per-station `wasserstein_station_i`;
- `dominant_freq_hz` and its error to the real fish, `swim_speed_bl_s` and its error, `phase_monotonicity` (fraction of steps with non-decreasing phase);
- `train/blowup_rate`: fraction of the last 256 episodes ended by the guard.

Baselines. Joint RL (PPO on joint targets), Joint RL+AMP (adversarial motion prior on a 46-dim feature: 40 bending values at 20 stations, left/right and up/down, plus 6 motion features; discriminator 2×128, learning rate 1.5×10^{-5}, update every 128 steps, batch 512, input noise 0.3, replay 40k, style weight 0.2), CPG+RL, BCO+RL (PPO initialized from a behavior-cloned policy whose pseudo-labels come from an inverse dynamics model trained on the fish's own random-interaction data) and BLM+RL (PPO in the PCA space).

PPO. rl_games; actor–critic MLP $[128, 64]$ with ELU, state-independent log-std initialized at -0.5, $\gamma = 0.99$, GAE $\lambda = 0.95$, learning rate 3×10^{-4} with KL-adaptive schedule (target 0.008), clip 0.2, entropy 0.003, horizon 256, 5 mini-epochs, minibatch 32 per environment, observation and value normalization. All baselines share these settings and the same fish, actuator gains, joint limits and reference windows.

E.2 TASK 2: FREE SWIMMING FROM VIDEO

The policy sees only quantities a top-view video can supply: planar position and velocity and heading, re-rooted at reset, plus its own BLM coefficients (4-dim). ***BLM+IL*** is pure behavior cloning of the real 30 Hz coefficient sequences; ***BCO*** adds an inverse dynamics model to pseudo-label the video with joint actions. BC uses early stopping on the validation loss (patience 10). Evaluation: 20 real initial states (random frames with at least 5 s of clean data ahead), 5 s roll-outs, no target. Metrics: Fréchet distance to the real continuation, forward speed (BL/s), heading stability $1/(1 + \mathrm{std}(\dot{\psi}))$, and the curvature/frequency metrics of Task 1.

E.3 Case Study: ROV capture

A BlueROV2 with a 6-DoF manipulator and a rigid basket is trained (PPO, 256 environments, same hyper-parameters) to capture a free-swimming fish. The prey is driven by the fish's own Task-2 BLM+IL policy; success requires the fish inside the basket with the opening facing up. Capture policies are trained for two fish (channel catfish 002 and lake sturgeon 016). The logged `success_rate` is the mean of the last outcome of every environment.

E.4 Training protocol and stopping rules

A job scheduler runs two jobs per GPU and stops a run only on convergence: completion (or capture success for Case Study) at or above 0.90 (0.85) for three consecutive epochs; or a plateau (no improvement above 0.02 in the best completion over 20 epochs after at least 40 epochs). A run whose blow-up rate exceeds 0.30 for three consecutive epochs after a 10-epoch warm-up is marked failed and retried once with a different seed. There is no wall-clock cap; a process that produces no epoch for 40 minutes (PhysX device hang) is restarted from its latest checkpoint. Runs of the same species are interleaved so that every species advances at the same rate. All metrics are logged to Weights & Biases every epoch.

Fish selection. Two fish per species are used for Task 1, one per species for Task 2 and two fish for Case Study, chosen among assets passing Stage-1 with the most clean video frames. One selected sturgeon (6.8 cm, 10 g) was replaced by a larger conspecific (15.2 cm, 200 g) because its FEM body was numerically unstable at the shared physics step regardless of actuator gain (> 200 guard triggers in the calibration probe at $k_p \in \{3, 10, 30, 120\}$).

F Compute

All experiments run on one workstation with two NVIDIA RTX PRO 6000 (96 GB) GPUs. A single trajectory-following run at 256 environments takes about 10 min per PPO epoch when two runs share a GPU; the whole box saturates at ≈ 0.4 epochs/min regardless of the number of concurrent runs, because the FEM soft-body solver dominates. Mesh generation costs about 30 Meshy credits per fish; the actor–critic loop takes 13–40 min per fish on one GPU (shared VLM).

G Manually Constructed Skeleton Templates

The actor is fine-tuned on, and the template library is derived from, fourteen fish models whose skeletons were placed by hand. This section documents how they were made.

Species and meshes. The set covers fourteen body plans spanning the shapes in our video dataset and beyond: carp, pumpkinseed, red snapper, roach, ruby, rudd, salmon, sea bass, spotted seatrout, tench, tiger barb, tilapia, trevally and zander. Each starts from a closed, textured fish mesh in Blender (metric units, length axis $+x$, head at $+x$, dorsoventral axis y), canonicalized exactly as the pipeline meshes are (Sec. B). Body length-to-height ratios range from 1.8 (tiger barb, pumpkinseed) to 3.8 (salmon).

Bone placement. Bones are axis-aligned box segments, placed one at a time along the longitudinal body midline from the skull toward the caudal peduncle, following published descriptions of teleost axial skeletons and body mechanics (Lauder & Tangorra, 2015; Videler, 1993). Three rules were enforced while placing them: (i) the box center lies on the body midline in both the lateral and the dorsal view; (ii) no box vertex may leave the skin (checked in Blender by intersecting each box with the mesh); (iii) consecutive boxes abut so the chain is continuous from the first post-cranial segment to the caudal peduncle. Segment count follows the visible flexibility of the species: 4 (pumpkinseed) to 8 (zander), 85 bones in total, median 6. Boxes are not rotated (all rotations are identity), so a bone is fully described by its center and size; the head and the caudal fin themselves are left unarticulated, which is why the expert chains span only 38–64% of body length (Sec. B.4). No armature or skinning is authored: the mesh–bone coupling is produced later by the FEM attachment step, identically for these models and for the pipeline assets.

Joints and physical parameters. The hand-placed models carry no per-bone joint parameters. Joint type (3-DoF D6), limits ($\pm 45^\circ$), PD gains ($k_p = 120$, $k_d = 6$) and the per-bone mass rule (body volume distributed in proportion to bone volume) are a single physics preset applied by the export step to every fish, so the annotator's decisions are limited to *where* bones go and *how large* they are. The preset values were chosen from the ranges reported for fish intervertebral joint mechanics and body stiffness (Jimenez et al., 2023) and validated in simulation on the salmon model before being frozen.

From models to a template library and training data. Each model is exported to a `skeleton.json` (bone centers, sizes, world transforms and the mesh bounding box). Two artifacts are built from these files:

- *Template library.* The actor prompt lists every template by name with its bone count and body-shape ratios (length/height, length/thickness); the actor selects the template whose body plan best matches the target fish and edits its bones (add / remove / reposition / resize) inside the new mesh.
- *Fine-tuning set.* For every model the bones are replayed one at a time in head-to-tail order; at step i the partial articulation (bones $0 \dots i{-}1$) is rendered in the three views used at inference, and the pair (renders, current articulation) $\rightarrow$ "add bone i with this center and size" is one supervised example. Roach and trevally are held out for validation, leaving 74 training examples from the other twelve models, which is the set the LoRA actor is trained on (Sec. B.4).

G.1 Hydrodynamics

Resolving the flow around a deformable body at every physics step is far too slow for the scale of training this benchmark requires, so we replace the fluid solver with a resistive panel model: the force on each body element is taken to be the force on a corresponding element of a straight cylinder moving at the same speed and inclination (Taylor, 1952). The deformable skin is partitioned into per-bone slices, and each surface panel p of a slice (area A_p, outward normal $\hat{\mathbf{n}}_p$) receives a quadratic drag force with normal (form-drag) and tangential (skin-friction) components,

$$\mathbf{f}_p = -\tfrac{1}{2}\rho A_p \left(C_n \left| v_{p,n} \right| v_{p,n}\, \hat{\mathbf{n}}_p + C_t \left\| \mathbf{v}_{p,t} \right\| \mathbf{v}_{p,t} \right), \qquad v_{p,n} = \mathbf{v}_p \cdot \hat{\mathbf{n}}_p, \;\; \mathbf{v}_{p,t} = \mathbf{v}_p - v_{p,n}\hat{\mathbf{n}}_p, \tag{11}$$

where $\mathbf{v}_p$ is the panel velocity induced by the motion of its bone, ρ is the water density, and $C_n = 1.0$, $C_t = 0.01$ are the flat-plate form-drag and skin-friction coefficients. Panel forces and their moments about the bone origin are summed into one wrench per bone and applied at every physics step. The normal term on the tail slices is what produces thrust, consistent with the elongated-body account of undulatory propulsion.

Quadratic drag integrated explicitly can reverse a bone's velocity within a single step, which sends the FEM body into the blow-up guard of Appendix D. Two limits prevent this: the force on each bone is capped at half its momentum per step, and the body speed is capped at 1.5 BL/s. Neither is active at normal swimming speeds, so they do not shape the gaits the policies learn.

The model is quasi-steady and therefore neglects added mass, vortex shedding and wake history, all of which a full fluid solver resolves (Liu et al., 2022; Yeo et al., 2010). Absolute swimming speeds should be read as specific to this simulator rather than as predictions about the animal. What the benchmark requires is weaker: that the same approximation apply to every method and every fish, so that differences between methods are attributable to the controller rather than to the fluid model.