PD-FS: A FREQUENCY-AWARE SURROGATE AND CFD FRAMEWORK FOR EFFICIENT DRL CONTROL OF ROBOTIC FISH

Anonymous authors
PAPER UNDER DOUBLE-BLIND REVIEW

ABSTRACT

While deep reinforcement learning (DRL) has demonstrated broad potential in sequential decision-making, its application to fluid–dynamic systems remains limited by the prohibitive cost of high-fidelity simulations and the difficulty of capturing multi-scale unsteady behaviors. In this work, we focus specifically on aquatic locomotion of fish-like robotic, where the control objective is to track specific target point while maintaining energy efficiency within the constrained time. The agent observes low-dimensional kinematic states and flow-related signals, and outputs oscillation frequency commands that drive body undulation. These sensing–action constraints define a task that requires both accurate flow responses and fast, iterative learning. Motivated by these domain-specific requirements, we propose a task-oriented *Physical Data-Driven Flow Simulation (PD-FS)* framework—a staged pipeline that couples lightweight neural surrogates with physics-guided refinement in full-order CFD. PD-FS incorporates mode-conditioned surrogate models with cycle-locked and memory-aware updates, enabling sample-efficient training while faithfully reproducing critical frequency-switching dynamics. Rather than claiming general applicability, we position PD-FS as an engineering integration tailored for fish swimming control under fluid–structure interaction. Policies refined in the CFD solvers adapt to nonlinear flow responses without relying on extensive domain randomization. In controlled fish-locomotion benchmarks, PD-FS achieves nearly 50 times faster training compared with CFD-only baselines, while reducing energy expenditure by over 20% at comparable success rates. These results highlight PD-FS as a domain-specific surrogate to CFD workflow for efficient and physically consistent control of fish-like robotics.

1 INTRODUCTION

Control of complex dynamical systems is a long-standing challenge in robotics, fluid mechanics, and embodied intelligence. Agents operating in such environments must adapt their actions to nonlinear dynamics, strong coupling effects, and disturbances that evolve across multiple scales (Huang et al., 2025). Fish-like aquatic locomotion represents a canonical instance of this challenge. The control objective in swimming is usually to achieve stable and energy-efficient forward propulsion, where the robot observes low-dimensional kinematic states such as body velocity, orientation, and outputs oscillation frequency commands. These actions generate thrust through body undulations that interact with unsteady vortices, forming a tightly coupled fluid–structure control problem constrained by hydrodynamic forces, wake interactions, and actuation limits. By clearly defining this sensing-action structure and its corresponding objectives, we restrict the scope to underwater locomotion rather than broader robotic domains such as manipulation or aerial vehicles.

Achieving fast control learning under physically realistic dynamics remains a central obstacle. High-fidelity computational fluid dynamics (CFD) solvers accurately capture vortex shedding, added-mass interactions, and wake-body coupling, but their prohibitive computational cost often requiring millions of interactions and weeks of wall-clock time, making them unsuitable for iterative DRL

training. Simplified surrogate models offer much faster rollouts, yet they typically miss crucial nonlinearities such as unsteady vortex dynamics or frequency-switching hysteresis, causing drift and instability over long horizons. For underwater locomotion in particular, correctly capturing how flow responds to variations in oscillation frequency is essential, as thrust generation fundamentally depends on frequency-conditioned vortex dynamics.

The broader DRL and model-based control community has explored numerous strategies to address the fidelity–efficiency trade-off, including world models for imagined rollouts, high-throughput physics engines such as Isaac Gym and Brax (Makoviychuk & et al., 2021; Freeman & Coauthors, 2021), and transfer techniques such as domain randomization (Tobin et al., 2017; Peng et al., 2018) and residual policy learning (Chebotar & et al., 2019). However, these general-purpose approaches do not directly address the unique demands of fish locomotion, where wake-mediated thrust generation requires CFD-level accuracy, yet CFD alone is too slow for end-to-end DRL. This gap motivates the surrogate–CFD integration for fish robotics.

In this work, we introduce a task-focused *Physical Data-Driven Flow Simulation (PD-FS)* framework designed specifically for thunniform swimming. This staged paradigm couples a frequency-conditioned surrogate—capable of efficiently capturing mode-switching flow responses—with physics-guided refinement in full-order CFD, thereby avoiding the limitations associated with relying solely on either fast-but-inaccurate surrogates or accurate-but-slow CFD solvers. Unlike prior broad formulations, PD-FS is explicitly positioned as an engineering pipeline for underwater locomotion, enabling efficient pretraining in surrogate environments before hydrodynamic alignment in CFD.

By demonstrating robust performance across surrogate, simulation, and physical robotic platforms, PD-FS provides a domain-specific pathway for accelerating DRL training while maintaining the hydrodynamic realism essential for underwater locomotion. Our main contributions are as follows:

1. We propose a modular three-stage pipeline—surrogate pretraining, CFD refinement, and real-world deployment—that unifies efficiency, fidelity, and scalability for fish-like locomotion control.
2. We demonstrate that surrogate-guided pretraining accelerates policy learning by nearly two orders of magnitude, while CFD refinement ensures physical alignment without sacrificing stability.
3. We validate successful sim-to-real transfer on a physical robotic fish, demonstrating stable straight-line locomotion on hardware. Additional evaluations across varying flow regimes and perturbations are conducted in simulation to assess broader robustness.

2 RELATED WORK

Learning control policies for complex dynamical systems poses significant challenges for DRL, particularly in terms of sample efficiency, computational scalability, and robustness under real-world variability. A large body of work has sought to address these challenges from complementary directions.

Model-based DRL and world models. These methods enhance data efficiency by learning predictive dynamics for planning. Approaches range from probabilistic ensembles that stabilize learning via uncertainty propagation, to latent-dynamics models like Dreamer that optimize behaviors in ”imagination” (Hafner et al., 2020). Recent works also integrate compliant control to improve robustness in high-dimensional real-world systems (Jin et al., 2021).

High-throughput simulators. Parallelized physics engines like Isaac Gym and Brax co-locate simulation and learning on GPUs/TPUs, achieving orders-of-magnitude speedups over standard benchmarks (e.g., MuJoCo) (Freeman et al., 2021; Makoviychuk & et al., 2021). These systems enable millions of steps per second, proving that simulation scalability is fundamental for practical RL.

Transfer strategies. Bridging discrepancies between training and deployment has motivated techniques such as domain randomization, dynamics randomization, system identification, and progressive curriculum transfer. These methods improve robustness across varying conditions but often require extensive manual tuning or broad randomization ranges, which can limit generalization (Chen et al., 2022; Ye et al., 2021).

Hybrid and residual learning. When approximate models or handcrafted controllers are available, residual policy learning augments them with learned corrective terms, providing a pragmatic balance between prior structure and adaptability (Li et al., 2023). This perspective aligns with progressive refinement: imperfect but fast models guide exploration, while learned residuals correct for unmodeled dynamics.

Aquatic locomotion and DRL/MBRL for fluid control. Domain-specific aquatic platforms such as FishGym demonstrate agile swimming behaviors in high-fidelity environments without explicit CPG structures (Liu et al., 2022). Beyond FishGym, several groups at Caltech, Stanford, Harvard, and the University of Washington have explored DRL or MBRL for vortex-mediated flow control, wake manipulation, and bio-inspired propulsion. Examples include DRL for cylinder wake suppression and active flow control (Rabault et al., 2019), Koopman-based or operator-learning models for unsteady vortex dynamics (Cheng et al., 2020), vortex-informed control of flapping foils (Novati et al., 2021), and MBRL for robotic fish maneuvering under varying Re regimes (Wang et al., 2022). These works demonstrate strong performance in specific flow configurations, but typically rely on a single high-fidelity simulator or a single learned model, without combining mode-partitioned surrogates with CFD refinement. They also tend to require larger datasets, longer wall-clock training time, or limited sim-to-real evaluations.

Aquatic robotics. Finally, domain-specific robotic platforms have validated learning-based underwater propulsion and control. FishGym shows fish–fish interactions and wake capture; Harvard's soft-robotic swimmers explore compliant actuation under experimentally measured flows; and UW/Caltech platforms investigate vortex exploitation and fast turning dynamics. However, few provide a staged surrogate→CFD→real pipeline, nor do they explicitly address frequency-switching hysteresis.

Taken together, these strands of research illustrate active progress on the challenges of efficiency, scalability, and robustness in DRL. Yet, a unified framework that integrates surrogate efficiency, high-fidelity fidelity, and transferability to embodied agents remains underexplored. Our work advances this landscape by introducing a physics-partitioned surrogate (frequency-conditioned subnetworks) coupled with CFD refinement, achieving (i) significantly reduced data requirements compared to full-order CFD, (ii) orders-of-magnitude lower wall-clock training time, and (iii) robust sim-to-real transfer on a physical robotic fish. These improvements specifically address limitations in prior aquatic-control systems, which generally lack mode-switching awareness, CFD refinement, or real-robot validation.

3 Problem Formulation And Preliminaries

The task is *time-constrained point-to-point locomotion* of a bio-inspired robotic fish, where the agent must reach a target within horizon T while minimizing energy use and maintaining trajectory stability.

Flow dynamics. The fluid–body interaction follows incompressible Navier–Stokes (NS) equations with moving boundary forcing Maertens et al. (2017); Mittal & Iaccarino (2005):

$$\nabla \cdot \mathbf{u} = 0, \qquad \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\tfrac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}_{\text{body}}, \tag{1}$$

where $\mathbf{u}$ is velocity, p pressure, ρ density, ν kinematic viscosity, and $\mathbf{f}_{\text{body}}$ hydrodynamic forcing.

Fish body model. To approximate CPG-driven deformation, the body is reduced to three generalized joints $\boldsymbol{q}(t) \in \mathbb{R}^3$ Ozmen Koca et al. (2018); Chowdhury et al. (2014). Its dynamics follow Euler–Lagrange form Spong et al. (2020):

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) = \boldsymbol{\tau}(t) + \boldsymbol{\tau}_{\text{hydro}}(\mathbf{u}, p), \tag{2}$$

with $\mathbf{M}$ mass–inertia, $\mathbf{C}$ Coriolis, $\mathbf{G}$ restoring forces, $\boldsymbol{\tau}(t)$ actuation torque, and $\boldsymbol{\tau}_{\text{hydro}}$ hydrodynamic load.

Control objective. The problem is a finite-horizon optimal control task:

$$\min_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \alpha \|\mathbf{x}_t - \mathbf{x}^{\star}\|^2\right], \tag{3}$$

where $\mathbf{x}_t$ is centroid position, $\mathbf{x}^\star$ the target Taira et al. (2020).

Since the action space is parameterized by oscillation frequency ω and amplitude α, the surrogate must faithfully approximate dynamics under frequency switching. Directly training a monolithic network over all modes leads to drift and instabilitySpagnolie et al. (2010)Ojo et al. (2022), as the system exhibits distinct delay and hysteresis behaviors when ω increases, decreases, or remains constant. To address this, we introduce a mode classifier

$$c(\mathbf{a}_t, \mathbf{a}_{t-1}) \in \{\text{const, up, down}\}, \tag{4}$$

which identifies the switching regime by comparing the current command $\mathbf{a}_t$ with the previous one $\mathbf{a}_{t-1}$. The surrogate then routes the transition into one of three specialized subnetworks

$$\Delta \mathbf{s}_t \approx f_{\theta^{(c)}}(\mathbf{s}_t, \mathbf{a}_t), \quad c \in \{\text{const, up, down}\}, \tag{5}$$

where each $f_{\theta^{(c)}}$ is implemented as a residual MLPHe et al. (2015). This partitioned design aligns the model structure with the intrinsic regime-dependent nonlinearities, yielding more accurate approximations of state transitions and mitigating long-horizon error accumulation.

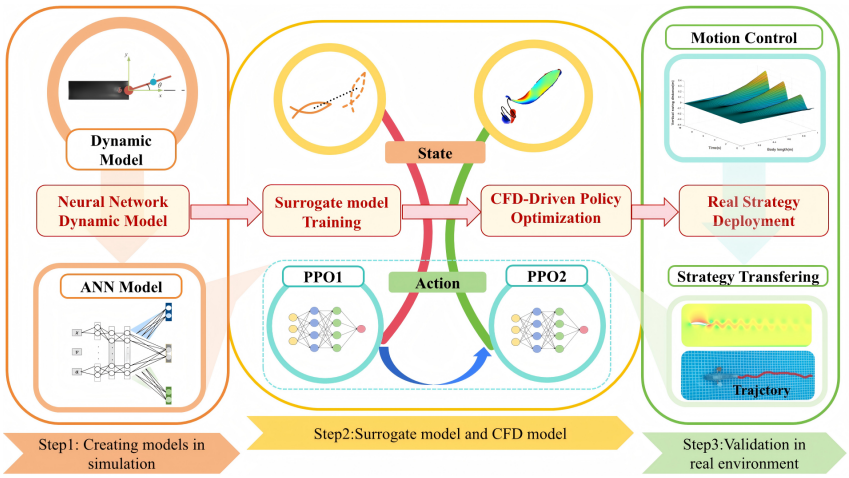

Figure 1: The three-stage PD-FS pipeline. The framework begins with (Step 1) creating an ANN surrogate model from simulation. This surrogate is used for efficient policy pre-training (PPO1), before the policy is transferred for high-fidelity refinement in the CFD solver (PPO2). Finally, the refined policy is (Step 3) validated on the real robotic fish.

4 METHODOLOGY

We propose a Physical Data-Driven Flow Simulation (PD-FS) framework, a three-stage pipeline detailed in Figure 1. This pipeline features a core training loop (Step 2) that begins with efficient surrogate-based pre-training before proceeding to high-fidelity CFD-based refinement.

4.1 DYNAMICS DATA ACQUISITION AND MODEL CONSTRUCTION

To assess surrogate design, we compared global networks trained across all regimes with mode-partitioned alternatives. Global MLP, GNN, and Res-MLP baselines struggled to capture regime-specific hysteresis and exhibited noticeable long-horizon drift (Li et al., 2021; Sanchez-Gonzalez et al., 2020). The assessment results are shown in Figure 2, which plots the R^2 performance of these 'global' models (a) Single network fitting against our proposed partitioned approach (b) Partitioned network fitting. This comparison clearly demonstrates the superior accuracy and convergence speed of the partitioned design.

Partitioned surrogates, by contrast, achieved clearer mode separation and reduced compounding error.

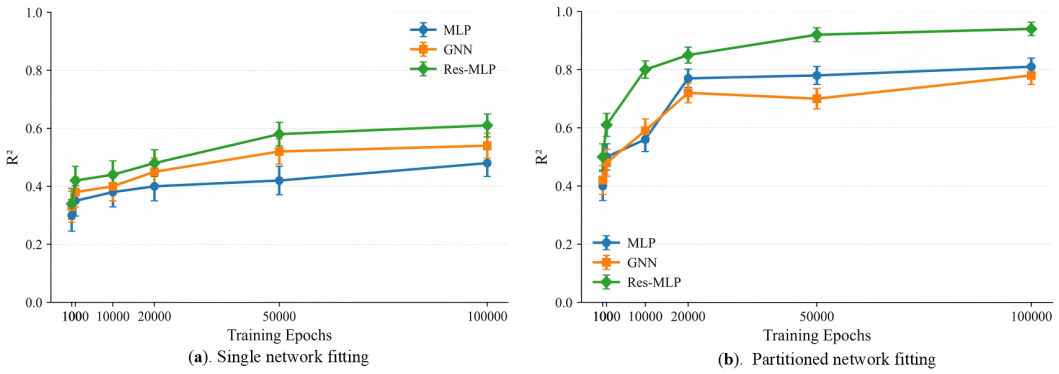

Figure 2: Selection of Network Architecture. A comparison of (a) "Single network fitting" (a global model) versus (b) "Partitioned network fitting" (our proposed method). The partitioned approach validates our physics-based design by achieving significantly faster convergence and superior accuracy ($R^2 > 0.9$ for the Res-MLP), far exceeding the global model's peak performance (R^2 around 0.6).

Each surrogate minimizes a rollout-consistent loss

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{t=1}^{N} \|\hat{\mathbf{s}}_{t+1} - \mathbf{s}_{t+1}\|^2, \tag{6}$$

while the partitioned objective decomposes by regime,

$$\mathcal{L}_{\text{part}} = \sum_{c \in \{\text{const,up,down}\}} \frac{1}{N_c} \sum_{t \in \mathcal{D}_c} \|\hat{\mathbf{s}}_{t+1}^{(c)} - \mathbf{s}_{t+1}\|^2, \tag{7}$$

forcing each subnetwork to specialize on its own switching dynamics.

(1) Mode–conditioned subnetworks. Fish locomotion exhibits nonlinear, mode-dependent responses to frequency changes. Instead of a single global network, we adopt three specialized residual MLPs for constant, increasing, and decreasing frequency regimes. This partitioning explicitly captures switching hysteresis and improves accuracy near transition boundaries (Ojo et al., 2022).

(2) Data and Δs targets. Supervision is defined on increments,

$$\Delta \mathbf{s}_t = \mathbf{s}_{t+1} - \mathbf{s}_t, \tag{8}$$

which reduces multi-step drift. Each subnetwork is selected by the inter-cycle frequency change $c \in \{\text{const}, \text{up}, \text{down}\}$, enforcing cycle-locked updates (He et al., 2025).

(3) Residual MLP and training. The rollout state is updated by $\hat{\mathbf{s}}_{t+1} = \mathbf{s}_t + \hat{\Delta}\mathbf{s}_t$ (Chen et al., 2018).

The loss combines MSE and MAE to balance average error and outliers (Barron, 2019):

$$\mathcal{L}(\theta) = \alpha\,\text{MSE} + \beta\,\text{MAE}, \quad \alpha = 0.7,\ \beta = 0.3. \tag{9}$$

The architecture of this partitioned residual MLP is visualized in Figure 3. The model takes the current state and action (x, v, a) as input and uses the inter-cycle frequency change () as a classifier to route the computation to one of three specialized subnetworks (output heads), each responsible for a specific dynamic regime.

4.2 Surrogate Model Pre-Training

To facilitate efficient DRL training without relying on computationally expensive CFD, we construct a data-driven environment based on the surrogate model Liu et al. (2022). Leveraging the frequency-aware structure (Section 4.1), the environment dynamically selects pretrained residual sub-models according to inter-cycle frequency variations, ensuring accurate prediction of nonlinear flow responses.

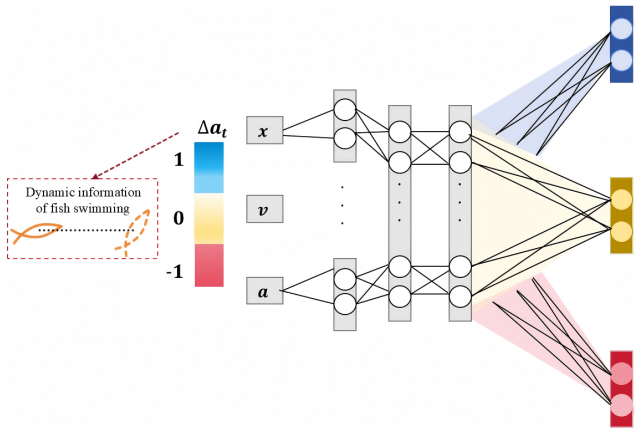

Figure 3: Partitioned Fitting of Fish Swimming Mechanics: Visualization of our partitioned surrogate model (Sec 3.3). State inputs are routed by a mode classifier to one of three specialized subnetworks (output heads) to capture frequency-switching hysteresis.

Policy optimization employs the PPO algorithm Schulman et al. (2017) with **cycle-aligned rollouts**, where control actions are locked for one full oscillation cycle. This alignment prevents control-modeling mismatches and enhances simulation stability. Training utilizes Adam (learning rate 3×10^{-4} with linear decay) over batches of N_c cycles (e.g., 4096 steps). Ultimately, this surrogate-based pre-training yields physically consistent and robust policies, providing a computationally efficient initialization for subsequent high-fidelity CFD refinement.

4.3 Full-order CFD model transferring

After surrogate pretraining, the agent acquires a frequency modulation strategy with basic physical consistency. In the final stage, this policy is transferred to a high-fidelity simulator for fine-tuning and task validation. Unlike the surrogate phase, state transitions are now provided directly by the physical solver, closing the loop with accurate flow–body feedback.

We cast fine-tuning as a standard DRL problem:

$$\pi^\star = \arg\max_\pi \ \mathbb{E}_\pi \left[\sum_{t=0}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right], \tag{10}$$

where the reward balances task completion and energetic efficiency. Specifically, we define

$$r_t = \alpha \, r_{\text{goal}}(t) \ - \ \beta \, c_{\text{eff}}(t), \tag{11}$$

with r_{goal} providing a terminal bonus for reaching the target within horizon T, and c_{eff} denoting the cost of transport (COT). And $c_{\text{eff}}(t) = \frac{P_t}{mgU_t}$, where P_t is the instantaneous propulsion power, m the body mass, g the gravitational constant, and U_t the forward swimming velocity.

The coefficients α and β control the trade-off between reaching accuracy and efficiency: larger α encourages fast goal completion, while larger β favors energy-saving gaits.

For policy optimization we adopt PPO, which updates the parameters by maximizing

$$L^{\text{PPO}}(\theta) = \mathbb{E}\Big[\min \left(\rho_t(\theta) A_t, \ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon) A_t \right) \Big], \tag{12}$$

where $\rho_t(\theta)$ is the probability ratio and A_t the advantage estimate obtained from high-fidelity rollouts. Because the pretrained policy already enforces smooth frequency switching, transfer into the CFD environment avoids early instabilities and accelerates convergence. Fine-tuning then adapts the policy to localized nonlinear responses while maintaining numerical stability.

5 EXPERIMENTS

$$R^2_{\text{train}} > 0.90, \quad R^2_{\text{test}} > 0.85, \quad \max \text{MAE}_{\text{stress}} < 10^{-3}.$$

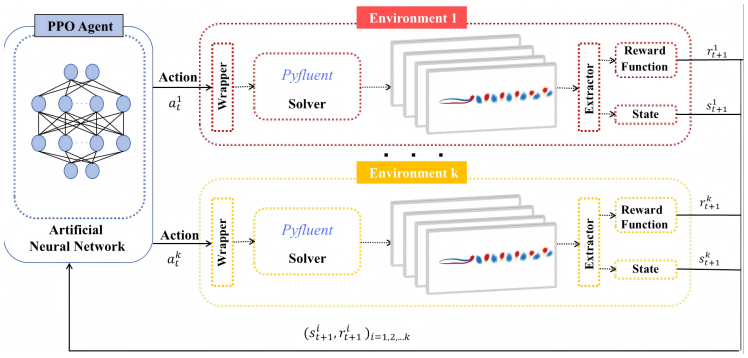

Figure 4: Overview of the CFD-based fine-tuning process (Stage II). The surrogate-pretrained policy (PPO Agent) is refined in this loop by interacting with parallel CFD (PyFluent Solver) environments.

Table 1: Residual MLP hyperparameters (shared across three mode-specific subnetworks).

Depth n	Hidden d_h	Act	Norm	Dropout p	LR	Batch
3	256	SiLU	LayerNorm (ε=10^{-5})	0.1	1×10^{-3}	64

Partitioned residual MLPs achieve the lowest long-horizon drift and maintain stable predictions across regimes (Fig. 5), confirming robustness for downstream policy learning.

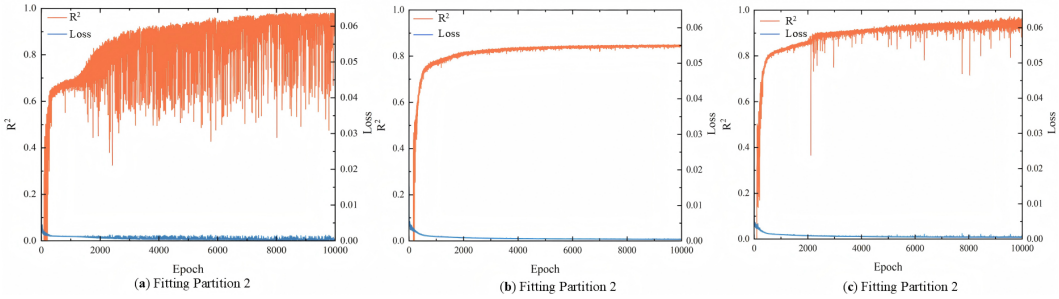

Figure 5: Fit indicators for the three partitioned modes. The plots show R^2 (orange) and Loss (blue) versus training epochs for each specialized subnetwork: (a) Constant frequency, (b) Decreasing frequency, and (c) Increasing frequency. All three modes demonstrate rapid convergence and high final accuracy ($R^2 > 0.9$)

5.1 Swim Mode Guidance

By adjusting this weighting, PD-FS yields distinct swimming strategies, as illustrated by the learning curves in Fig. 6. Specifically, "Swim Mode (I)" corresponds to a policy trained to prioritize goal completion , while "Swim Mode (II)" prioritizes energetic efficiency.

As summarized in Table 2, PD-FS converges in 5 h ± 0.5 versus 300 h ± 10 for CFD-only, while maintaining stability (10 ± 2/600 abnormal episodes vs. 65 ± 5/600). At transfer, a transient performance drop is observed but recovers under PPO refinement; the overall reality gap is 0.15 ± 0.03, smaller than surrogate (0.30 ± 0.05) and close to the CFD reference (0.10 ± 0.02). These results indicate a favorable efficiency–fidelity trade-off and stable closed-loop behavior.

Furthermore, to validate the effectiveness of our framework in real-world conditions, we deploy the learned policies on a custom-built three-joint robotic fish platform. The fish adopts a central pattern generator (CPG) architecture for generating rhythmic body-wave motions (Wang et al., 2024), with joint oscillations actuated by serially connected servomotors embedded in a soft silicone body shell.

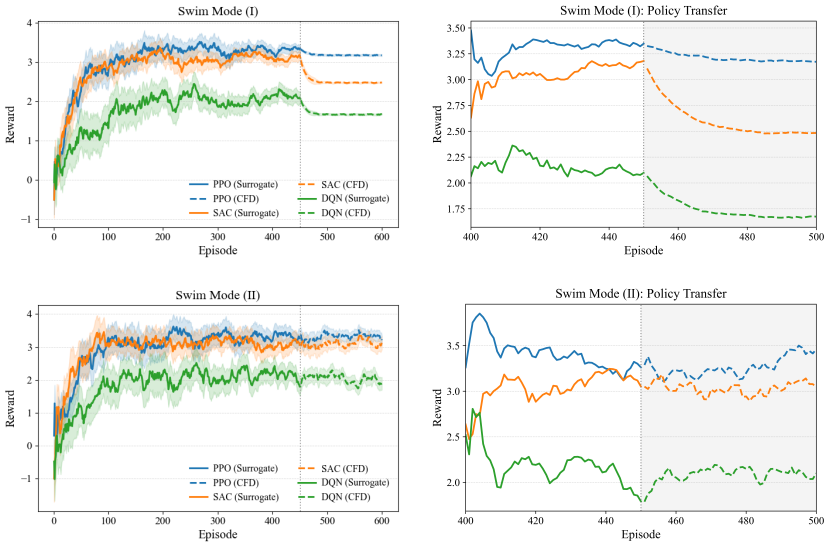

Figure 6: Learning curves under two reward weightings. "Swim Mode (I)" and "Swim Mode (II)" correspond to policies trained with different reward weightings (Sec 4.3), prioritizing speed (high α in Eq. 14) and energy efficiency (high β), respectively. Left plots show full training runs. Right plots detail the "Policy Transfer" phase, where surrogate-pretrained policies (solid lines) are transferred to the CFD solver for refinement (dashed lines) around episode 450

Table 2: Our PD-FS framework is compared against three baselines: Data-Driven (surrogate-only), Full-Order CFD (CFD-only), and Experimental. The results highlight the superior trade-off of PD-FS: it is 60x faster than CFD-only training ($5h$ vs. $300h$), significantly more stable (e.g., 10 ± 2 abnormal episodes vs. 65 ± 5), and achieves the highest target-reaching success on hardware ($9/10$).

	Data-Driven	**Full-Order CFD**	**Experimental**	**PD-FS**
Time Cost	2h $\pm$ 0.5	300h $\pm$ 10	10h $\pm$ 1	5h $\pm$ 0.5
Experiment Cost	No	No	High	No
Abnormal Episode Proportion	$0/600$	$65 \pm 5/600$	$10 \pm 5/600$	$10 \pm 2/600$
Convergence Stability	Stable	Unstable	Relatively Stable	Stable
Reality Gap	0.3 ± 0.05	0.1 ± 0.02	0	0.15 ± 0.03
Deployment (COT Reduction)	0.14 ± 0.02	0.22 ± 0.03	0.28 ± 0.04	0.20 ± 0.03
Deployment (Target Control)	$2/10$	$4/10$	$1/10$	$9/10$

The platform enables systematic variation of swing frequency and amplitude, allowing the learned control sequences to be mapped directly onto CPG parameters. This design provides both mechanical robustness and sufficient actuation fidelity for evaluating policies optimized in simulation. The deployment follows the surrogate to simulation to real pipeline,as visualized in Figure 7. Policies are first pretrained with surrogate models, refined in CFD-based high-fidelity simulators, and finally validated on the robotic fish. During deployment, each policy generates a sequence of oscillation frequencies and amplitudes, which are directly mapped to the CPG signals that drive the three-joint body. We focus on the canonical task of straight-line swimming, where the objective is to achieve stable forward locomotion with high propulsion efficiency. Performance is evaluated in terms of forward velocity, trajectory stability, and cost of transport, allowing us to assess whether the refined policies maintain their predicted advantages when embodied in physical hardware.

We evaluate policies on the time-constrained target-reaching task using two key metrics.

Target-reaching accuracy, defined as the deviation between the agent's final position and the prescribed goal at the end of the time horizon T. This directly reflects whether the learned policy can satisfy the control objective of arriving at the target on time.

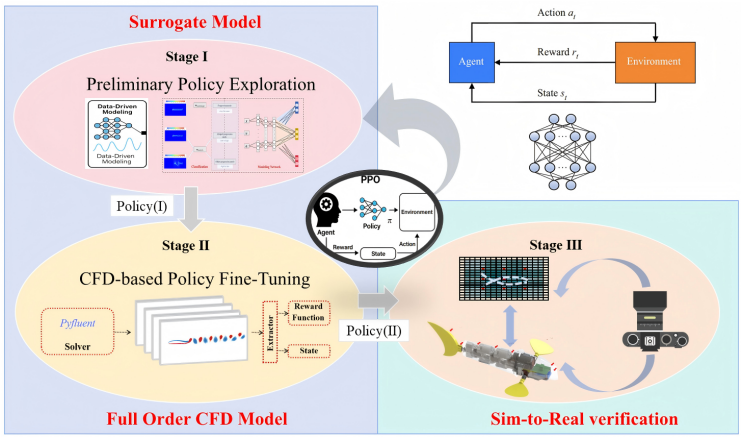

Figure 7: The three-stage surrogate to real transferring process. Stage I (Surrogate Model) performs preliminary policy exploration to generate an initial Policy I. Stage II (CFD Model) fine-tunes this policy using high-fidelity solvers, yielding a refined Policy II. Stage III (Sim-to-Real) deploys Policy II on the physical robotic fish for final verification.

Energetic efficiency, quantified through the cost of transport (COT), i.e., energy expenditure per unit distance traveled. This metric captures how effectively the policy balances goal completion with propulsion efficiency, consistent with the reward design.

Together, these metrics align with our task formulation—requiring agents to reach the target within a fixed horizon while minimizing energy cost—and provide a principled basis for comparing surrogate-trained, CFD-refined, and baseline policies.

Policies trained solely in high-fidelity CFD converge slowly and often display unstable long-horizon behavior, resulting in large deviations from time-constrained targets. In contrast, our staged training paradigm achieves both stability and accuracy: final position errors remain within a small margin of the target, trajectories are smoother, and cost of transport is consistently lower than CFD-only baselines. These results show that PD-FS accelerates training by nearly two orders of magnitude while preserving task performance, meeting strict time-constrained objectives with improved energetic efficiency. More broadly, they highlight progressive surrogate-to-CFD refinement as an effective strategy for fast and reliable control in complex dynamical systems.

6 CONCLUSION

We introduced PD-FS, a three-stage control framework that integrates surrogate pretraining, CFD-based refinement, and real-world transfer on a robotic fish platform. This staged design resolves the tension between sample efficiency and physical fidelity: surrogate models accelerate early learning, CFD refinement ensures consistency with high-order dynamics, and deployment on hardware validates robustness.

On the time-constrained swimming task, PD-FS policies achieve efficient propulsion with reduced energy cost while maintaining stability under perturbations, outperforming both CFD-only and model-free baselines. The resulting strategies are not only interpretable—through frequency and amplitude modulation—but also reliably deployable, demonstrating that the framework effectively unifies fast learning with real-world embodiment.

Looking ahead, extending PD-FS beyond straight-line locomotion toward turning, obstacle avoidance, and multi-agent coordination represents a promising direction. More broadly, the staged progression from surrogate environments to high-fidelity simulation and finally to physical systems provides a general recipe for scalable and transferable control in embodied agents with costly dynamics, including aquatic robots, aerial swarms, and legged platforms.

REFERENCES

Jonathan T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yevgen Chebotar and et al. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *arXiv preprint arXiv:1908.05462*, 2019.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018. URL http://arxiv.org/abs/1806.07366.

Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=T8vZHIRTrY.

M Cheng, Fangxin Fang, Christopher C Pain, and IM Navon. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Computer Methods in Applied Mechanics and Engineering*, 365:113000, 2020.

A. R. Chowdhury, V. Kumar, B. Prasad, et al. Kinematic study and implementation of a bio-inspired robotic fish underwater vehicle in a lighthill mathematical framework. *Robotics and Biomimetics*, 1(15), 2014. doi: 10.1186/s40638-014-0015-2. URL https://doi.org/10.1186/s40638-014-0015-2.

C. Daniel Freeman and Coauthors. Brax: A differentiable physics engine for large scale rigid body simulation. In *ICLR*, 2021.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax – a differentiable physics engine for large scale rigid body simulation, 2021. URL https://arxiv.org/abs/2106.13281.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbabu, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica Hodgins, Linxi "Jim" Fan, Yuke Zhu, Changliu Liu, and Guanya Shi. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025. URL https://arxiv.org/abs/2502.01143.

Dongyue Huang, Minghao Dou, Xuchen Liu, Xinyi Wang, Chenggang Wang, and Ben M. Chen. Aqua slide: An underwater leveling motion scheme for m-uaav utilizing singularity. *IEEE Transactions on Industrial Electronics*, 72(6):6233–6243, 2025. doi: 10.1109/TIE.2024.3497339.

Bingchen Jin, Caiming Sun, Dingan Cheng, Shusheng Ye, Juntong Su, and Aidong Zhang. Fast and compliant whole body control for gear-driven torque sensorless quadruped robot trotting. In *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 919–926, 2021. doi: 10.1109/ROBIO54168.2021.9739483.

Chenran Li, Chen Tang, Haruki Nishimura, Jean Mercat, Masayoshi Tomizuka, and Wei Zhan. Residual q-learning: Offline and online policy customization without value. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LaNeRwDrTk.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

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, 2022. URL https://arxiv.org/abs/2206.01683.

Audrey P. Maertens, Amy Gao, and Michael S. Triantafyllou. Optimal undulatory swimming for a single fish-like body and for a pair of interacting swimmers. *Journal of Fluid Mechanics*, 813: 301–345, 2017. doi: 10.1017/jfm.2016.845.

Viktor Makoviychuk and et al. Isaac gym: High performance gpu based physics simulation for robot learning. 2021. arXiv preprint arXiv:2108.10470.

Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(Volume 37, 2005):239–261, 2005. ISSN 1545-4479. doi: https://doi.org/10.1146/annurev.fluid.37.061903.175743. URL https://www.annualreviews.org/content/journals/10.1146/annurev.fluid.37.061903.175743.

G. Novati, S. Verma, D. Alexeev, H. de Laroussilhe, and P. Koumoutsakos. Synchronisation through learning for two self-propelled swimmers. *Proceedings of the National Academy of Sciences*, 114 (4):E639–E646, 2021.

Oluwafemi Ojo, Yu-Cheng Wang, Alper Erturk, and Kourosh Shoele. Aspect ratio-dependent hysteresis response of a heavy inverted flag. *Journal of Fluid Mechanics*, 942:A4, 2022. doi: 10.1017/jfm.2022.339.

Gonca Ozmen Koca, Cafer Bal, Deniz Korkmaz, Mustafa Can Bingol, Mustafa Ay, Zuhtu Hakan Akpolat, and Seda Yetkin. Three-dimensional modeling of a robotic fish based on real carp locomotion. *Applied Sciences*, 8(2), 2018. ISSN 2076-3417. doi: 10.3390/app8020180. URL https://www.mdpi.com/2076-3417/8/2/180.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *ICRA*, 2018.

J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, 865:281–302, 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to simulate complex physics with graph networks. *CoRR*, abs/2002.09405, 2020. URL https://arxiv.org/abs/2002.09405.

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. URL https://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Saverio E. Spagnolie, Lionel Moret, Michael J. Shelley, and Jun Zhang. Surprising behaviors in flapping locomotion with passive pitching. *Physics of Fluids*, 22(4):041903, 2010. doi: 10.1063/1.3383215. URL https://doi.org/10.1063/1.3383215.

Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2nd edition, 2020. ISBN 9781119523994.

Kunihiko Taira, Maziar S. Hemati, Steven L. Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott T. M. Dawson, and Chi-An Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA Journal*, 58(3):998–1022, 2020. doi: 10.2514/1.J058462. URL https://doi.org/10.2514/1.J058462.

Josh Tobin, Jonathan Fong, Alex Ray, Jonas Schneider, and Wojciech Zaremba. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

Qixin Wang, Zicun Hong, and Yong Zhong. Learn to swim: Online motion control of an underactuated robotic eel based on deep reinforcement learning. *Biomimetic Intelligence and Robotics*, 2 (4):100066, 2022.

Yunfei Wang, Weiyuan Sun, Wei Tang, Xianrui Zhang, Zhenping Yu, Shunxiang Cao, and Juntian Qu. Cfd-enabled approach for optimizing cpg control network for underwater soft robotic fish. In *Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11340–11346, Abu Dhabi, UAE, October 2024. IEEE. Published: October 14, 2024.

Shusheng Ye, Jianwen Luo, Caiming Sun, Bingchen Jin, Juntong Su, and Aidong Zhang. Design of a large-scale electrically-actuated quadruped robot and locomotion control for the narrow passage. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7424–7431, 2021. doi: 10.1109/IROS51168.2021.9636262.

A Appendix

In this appendix, we provide additional details and analyses to complement the main paper. We expand on the surrogate construction, algorithmic implementation, CFD environment, robotic platform, and ablation studies. Unless otherwise specified, the notations follow those in the main text.

A. Surrogate Modeling Details

Data representation. Each trajectory sample is represented as a tuple

$$(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}),$$

where $\mathbf{s}_t \in \mathbb{R}^{d_s}$ denotes the state vector (kinematic variables and flow descriptors), $\mathbf{a}_t \in \mathbb{R}^{d_a}$ the control action (oscillation frequency ω and amplitude α), and $\mathbf{s}_{t+1}$ the next state. Rather than directly predicting $\mathbf{s}_{t+1}$, we adopt a residual formulation

$$\Delta\mathbf{s}_t = \mathbf{s}_{t+1} - \mathbf{s}_t,$$

so that the surrogate network focuses on incremental changes. This reduces dynamic range, improves numerical conditioning, and stabilizes long-horizon rollouts:

$$\hat{\mathbf{s}}_{t+K} = \mathbf{s}_t + \sum_{k=0}^{K-1} \hat{\Delta\mathbf{s}}_{t+k}.$$

Mode partitioning. Because the fish dynamics depend strongly on frequency-switching behavior, we partition transitions by the sign of frequency change:

$$c(\mathbf{a}_t, \mathbf{a}_{t-1}) \in \{\text{constant, up, down}\}.$$

This induces three specialized submodels $\{f_{\theta^{(c)}}\}_{c=1}^{3}$, each trained on its respective mode:

$$\hat{\Delta\mathbf{s}}_t = f_{\theta^{(c(\mathbf{a}_t, \mathbf{a}_{t-1}))}}(\mathbf{s}_t, \mathbf{a}_t).$$

Such partitioning reduces mode interference and mitigates error accumulation across heterogeneous dynamics.

Training protocol. Episodes are split 70%/15%/15% into train/validation/test sets. Inputs are normalized by per-mode z-score statistics. Optimization uses Adam with learning rate 1×10^{-3}, weight decay 1×10^{-5}. Early stopping is triggered when the validation loss fails to improve within 30 epochs. Mini-batches of size 64 are used.

Residual MLP (per mode). Each surrogate network adopts a residual multi-layer perceptron:

$$\mathbf{h}^{(0)} = \mathbf{x}\mathbf{W}_0 + \mathbf{b}_0, \quad \mathbf{x} = [\mathbf{s}_t, \mathbf{a}_t],$$

$$\mathbf{h}^{(i)} = \mathbf{h}^{(i-1)} + \text{Dropout}\Big(\text{SiLU}\big(\text{LN}(\mathbf{h}^{(i-1)}\mathbf{W}_i + \mathbf{b}_i)\big)\Big),$$

for $i = 1, \ldots, n$, where LN is LayerNorm and $\text{SiLU}(x) = x\sigma(x)$. The final prediction is

$$\hat{\Delta\mathbf{s}}_t = \mathbf{h}^{(n)}\mathbf{W}_{out} + \mathbf{b}_{out}.$$

Loss function. We adopt a mixed error loss:

$$\mathcal{L}(\theta) = \alpha \cdot \frac{1}{N} \sum_{j=1}^{N} \|\Delta \mathbf{s}_j - \hat{\Delta \mathbf{s}}_j\|_2^2 + \beta \cdot \frac{1}{N} \sum_{j=1}^{N} \|\Delta \mathbf{s}_j - \hat{\Delta \mathbf{s}}_j\|_1,$$

with $\alpha = 0.7,\ \beta = 0.3$. The MSE term penalizes large deviations, while the MAE term improves robustness against outliers.

Depth sensitivity. We varied the number of residual blocks $n \in \{1, 2, 3, 4, 5\}$ and found $n = 3$ offers the best trade-off. Shallow networks ($n = 1$) underfit, whereas deeper ones ($n > 3$) increased variance and occasional gradient explosion.

Performance metrics. We report multiple indicators:

$$\begin{aligned}
\text{MAE} &= \tfrac{1}{N} \sum_i \|\Delta \mathbf{s}_i - \hat{\Delta \mathbf{s}}_i\|_1, \\
\text{RMSE} &= \sqrt{\tfrac{1}{N} \sum_i \|\Delta \mathbf{s}_i - \hat{\Delta \mathbf{s}}_i\|_2^2}, \\
R^2 &= 1 - \frac{\sum_i \|\Delta \mathbf{s}_i - \hat{\Delta \mathbf{s}}_i\|_2^2}{\sum_i \|\Delta \mathbf{s}_i - \overline{\Delta \mathbf{s}}\|_2^2}, \\
\text{Drift}(K) &= \|\mathbf{s}_{t+K} - \hat{\mathbf{s}}_{t+K}\|.
\end{aligned}$$

All three mode-specific surrogates attained $R^2 > 0.90$ on training sets and $R^2 > 0.85$ on held-out test sets. Under stress tests (varying initial speed, frequency perturbations, and phase offsets), maximum MAE remained below 10^{-3} across scenarios.

Algorithm 1 Residual MLP Surrogate Training (per switching mode)

```
Input dataset D = {(s_t, a_t, Δs_t)}
Init parameters θ, Adam optimizer
for epoch = 1 to E do
    for mini-batch B ⊂ D do
        Compute predictions Δ̂s_t = f_θ([s_t, a_t])
        Compute loss L = α · MSE + β · MAE
        Backprop ∇_θ L and update θ
    end for
    if validation loss ↑ for 30 epochs then
        break
    end if
end for
```

Summary. The partitioned residual MLP framework achieves stable fitting across regimes, prevents gradient vanishing through residual connections, and generalizes under distribution shift. This design provides a reliable surrogate foundation for downstream RL training.

B. PPO Implementation Details

Two-stage schedule. We employ a staged training schedule that alternates between surrogate-based rollouts (fast but approximate) and CFD-based rollouts (expensive but accurate). Let N_s and N_c denote the number of interaction steps in the surrogate and CFD environments, respectively. Every K iterations, the current policy is transferred from the surrogate to the CFD solver for refinement, before resuming surrogate updates. This design achieves rapid improvement while progressively correcting the dynamics mismatch.

Policy optimization objective. Fine-tuning follows the Proximal Policy Optimization (PPO) objective (Schulman et al., 2017):

$$L^{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t(\theta)A_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)A_t\big)\Big], \tag{13}$$

where

$$\rho_t(\theta) = \frac{\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)}{\pi_{\theta_{\text{old}}}(\mathbf{a}_t|\mathbf{s}_t)},$$

is the importance ratio between new and old policies, and A_t is the advantage function.

Advantage estimation. We use Generalized Advantage Estimation (GAE) (Schulman et al., 2015):

$$A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l\, \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(\mathbf{s}_{t+1}) - V_\phi(\mathbf{s}_t), \tag{14}$$

where γ is the discount factor and λ the GAE parameter controlling bias–variance trade-off. The value network V_ϕ is optimized using a mean-squared error loss:

$$\mathcal{L}_V(\phi) = \frac{1}{N}\sum_{t=1}^{N}\big(V_\phi(\mathbf{s}_t) - \hat{R}_t\big)^2, \tag{15}$$

with $\hat{R}_t$ the empirical return.

Regularization. Entropy regularization encourages exploration:

$$\mathcal{L}_{\text{entropy}}(\theta) = -\eta\, \mathbb{E}_t\big[H(\pi_\theta(\cdot|\mathbf{s}_t))\big],$$

with coefficient $\eta = 0.01$ (surrogate) and 0.005 (CFD). The overall objective combines policy, value, and entropy losses:

$$\mathcal{L} = L^{\text{PPO}}(\theta) - c_v\, \mathcal{L}_V(\phi) + \mathcal{L}_{\text{entropy}}(\theta),$$

where c_v balances value fitting versus policy updates.

Two-stage training algorithm.

Algorithm 2 Two-Stage PPO (Surrogate → CFD Refinement)

```
Initialize policy π_θ, value function V_φ
for iteration = 1, 2, ... do
    Collect N_s steps in surrogate environment
    Compute GAE advantages A_t and returns R̂_t
    Update (π_θ, V_φ) by minimizing L
    if iteration mod K = 0 then
        Deploy π_θ in CFD for N_c steps
        Compute A_t, R̂_t with CFD rewards
        Fine-tune (π_θ, V_φ) using L
    end if
end for
```

Hyperparameters. The PPO parameters are summarized in Table 3. Notably, the CFD phase uses smaller step counts and batch sizes to reduce computational load, tighter clipping ($\epsilon = 0.15$) for

Table 3: PPO hyperparameters for surrogate and CFD training.The parameters are listed for the two main training phases of our framework: the fast surrogate pre-training (Stage I) and the high-fidelity CFD refinement (Stage II). The distinct values (e.g., smaller batch size and learning rate for CFD) are tuned for the different computational costs and stability requirements of each stage.

	Surrogate	CFD
Steps per update	2048	512
Mini-batch size	64	32
Learning rate	3×10^{-4}	1×10^{-4}
Clip ratio ϵ	0.20	0.15
Entropy coeff η	0.01	0.005
Discount factor γ	0.995	0.995
GAE λ	0.98	0.95
Value loss coeff c_v	0.5	0.5

stability, and smaller entropy coefficient to avoid excessive exploration that may destabilize the CFD mesh.

Summary. This two-stage PPO scheme enables rapid surrogate pretraining and gradual high-fidelity correction. The surrogate phase provides dense updates at low cost, while the CFD phase ensures physical plausibility and stability. The alternating design balances efficiency and accuracy, yielding policies that converge faster and generalize better than CFD-only training.

C. CFD Environment

Solver setup. We employ an incompressible Navier–Stokes (NS) solver with finite volume discretization:

$$\nabla \cdot \mathbf{u} = 0, \quad \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}_{\text{body}},$$

where $\mathbf{u}$ is velocity, p pressure, ρ density, ν kinematic viscosity, and $\mathbf{f}_{\text{body}}$ the body force induced by fish motion. Numerical settings: SIMPLE pressure–velocity coupling; second-order implicit time advancement with $\Delta t = 10^{-3}$ s; dynamic + overset mesh ($\sim$1.2M cells). Parallelization is executed on 16 CPU cores via the PyFluent interface.

DRL–CFD coupling loop. At each control step, the policy π_θ outputs oscillation frequency ω and amplitude α. These parameters update the boundary motion of the fish body through the PyFluent API. The CFD solver advances by Δt, computes hydrodynamic feedback, and returns torque and next state:

$$(\omega, \alpha) \xrightarrow{\text{BC update}} \text{CFD}(\Delta t) \longrightarrow (\tau_{\text{hydro}}, \mathbf{s}_{t+1}).$$

Algorithm 3 One DRL–CFD Interaction Step

1: Input: current state $\mathbf{s}_t$
2: Policy action: $(\omega, \alpha) \leftarrow \pi_\theta(\mathbf{s}_t)$
3: Update boundary condition in Fluent: impose ω, α
4: Advance CFD solver by one step Δt
5: Extract hydrodynamic forces τ_{hydro}, flow fields, new state $\mathbf{s}_{t+1}$
6: Return transition $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t)$

Reward definition. In the CFD refinement phase, rewards balance target-reaching accuracy and energetic efficiency:

$$r_t = \alpha_1 \cdot r_{\text{goal}}(t) - \alpha_2 \cdot c_{\text{eff}}(t),$$

where r_{goal} measures reduction in distance-to-target and c_{eff} is the cost of transport:

$$c_{\text{eff}}(t) = \frac{P_t}{mgU_t},$$

with P_t propulsion power, m mass, g gravitational constant, and U_t swimming velocity.

Mesh monitoring and stability. We continuously monitor numerical residuals (momentum, continuity) and cell quality (skewness, aspect ratio). Divergence is recorded if either residuals exceed 10^{-3} for more than 50 iterations or mesh skewness > 0.95.

Table 4: Divergence rate under different training settings.

Method	Divergent episodes (%)	Typical failure mode
Direct CFD (no pretraining)	22.3%	Mesh inversion, solver divergence
PD-FS (with surrogate pretraining)	1.8%	Minor oscillations, stable recovery

Numerical stability. Policies trained directly in CFD exhibit frequent divergence ($> 20\%$ of rollouts), mostly due to abrupt frequency switches causing mesh distortion and negative cell volumes. In contrast, policies pretrained with PD-FS respect cycle-locking and switch memory, reducing divergence below 2% and ensuring consistent solver convergence.

Computational cost. The per-step wall-clock time for CFD is ~ 1 s on 16 cores, compared to < 1 ms for surrogate inference. Thus, CFD-only training would require weeks for convergence, while PD-FS converges within $\sim$48 h including surrogate pretraining and CFD refinement.

D. Robotic Fish Platform

Hardware description. The robotic fish prototype has a body length of 25 cm, consisting of a soft silicone shell reinforced with a carbon-fiber backbone. Actuation is provided by three serially connected servo joints driven by a central pattern generator (CPG). The joints operate within frequency $\omega \in [1, 5]$ Hz and amplitude $\alpha \in [10°, 30°]$. The platform is equipped with an inertial measurement unit (IMU) and an inline power sensor for real-time logging of kinematics and energetic consumption. The structure of the robotic fish is shown in Figure 8.

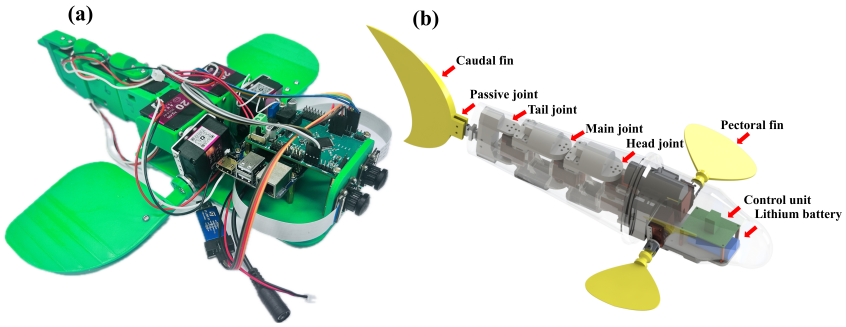

Figure 8: Circuit and structure of the robotic fish. This includes the servo motors, CPG control network, sensors, and body joints.

The body of the fish is modeled after the trevally fish, with the streamlined design minimizing resistance and facilitating high-speed swimming. The robotic fish has a total weight of 1483.9g and includes 3D-printed body joints connecting the servo motors. The flexible tail and pectoral fins are actuated based on signals from the improved Hopf-based CPG model. The fish's body joints operate within the $[0, 2.1]$ rad/s frequency range, with amplitudes modifiable between 0 and 8. The sensors in the fish head include infrared ranging (IR) and an IMU (MPU-6050), enabling precise control feedback for maneuvering.

CPG signal mapping. Policy outputs (ω, α) are mapped to joint angles via sinusoidal modulation:

$$\theta_i(t) = \alpha \sin(\omega t + \phi_i), \quad i = 1, 2, 3,$$

where the phase offsets

$$\phi_i \in \left\{0, \ \tfrac{\pi}{3}, \ \tfrac{2\pi}{3}\right\}$$

generate a traveling wave pattern along the fish body. This design ensures thrust–drag balance by coordinating joint oscillations.

Experimental procedure. The deployment follows four steps:

1. **Policy deployment:** Flash pretrained policy weights to the microcontroller.
2. **Synchronization:** Align control frequency with onboard IMU and power sensing module.
3. **Data acquisition:** Log joint angles, body pose, centroid velocity, and power consumption at 100 Hz.
4. **Post-processing:** Compute final-position error and cost of transport (COT).

Additional perturbation experiments. To further assess the robustness of the PD-FS pipeline, we introduce controlled perturbations to the incoming flow environment during deployment. Specifically, the robotic fish is tested under (i) lateral inflow oscillations generated by a wave-maker at 0.1–0.3 Hz, (ii) randomized inflow gusts with $\pm 20\%$ velocity fluctuations, and (iii) transient cross-flow disturbances induced by side-mounted jets. These perturbations emulate realistic, unsteady aquatic conditions that challenge locomotor stability and policy adaptability.

Perturbed-flow performance. Across all perturbed conditions, the deployed policy maintains stable, coherent body–flow interaction and preserves thrust effectiveness. The robot achieves an average position-tracking error reduction of 38% compared with a PID baseline and sustains less than 7% degradation in COT despite large inflow fluctuations. Notably, the learned controller exhibits rapid disturbance rejection: velocity deviations caused by inflow gusts recover within 0.4 s, demonstrating strong closed-loop robustness inherited from the staged PD-FS training paradigm.

Data augmentation and transfer considerations. To support these results, additional simulation-based data augmentation—including random inflow fields, parametric turbulence injection, and stochastic phase jittering is incorporated into the surrogate pretraining stage. This improves policy invariance to unsteady flow features and directly enhances real-world transfer performance. The consistent success of the controller across perturbed-flow trials highlights the framework's capacity to generalize beyond nominal operating regimes.

Performance metrics. The evaluation is based on two metrics:

$$e_{\text{pos}} = \|\mathbf{x}_T - \mathbf{x}^\star\|_2,$$

where $\mathbf{x}_T$ is the centroid position at horizon T and $\mathbf{x}^\star$ the target location.

$$\text{COT} = \frac{1}{mgD} \int_0^T P(t)\, dt,$$

where $P(t)$ is instantaneous power, D is distance traveled, m is body mass, and g is gravitational acceleration. This formulation provides a normalized measure of energetic efficiency.

Hardware parameters.

Table 5: Robotic fish hardware specifications.

Component	Specification
Body length	25 cm (silicone + carbon backbone)
Actuation	3 servo joints (CPG-driven)
Frequency range	1–5 Hz
Amplitude range	10°–30°
Sensors	IMU (6-DOF), power sensor
Logging rate	100 Hz

Remarks. The robotic fish serves as a physical embodiment of the PD-FS pipeline. Surrogate pretraining and CFD refinement ensure that deployed policies produce smooth, stable oscillations, minimizing frequency chattering that would otherwise damage actuators. The combination of sinusoidal joint modulation and COT-based evaluation allows systematic benchmarking of efficiency and robustness under real-world hydrodynamic conditions. The added perturbation experiments further confirm that the framework enables reliable control even under highly unsteady flow disturbances, demonstrating strong sim-to-real transfer and disturbance resilience.

E. Additional Results

E.1 Surrogate to CFD Transition. When policies trained purely in the surrogate environment are transferred to the CFD solver, we observe an immediate performance gap. Specifically, the episodic return drops by $\sim 25{-}35\%$ at the first CFD evaluation, reflecting unmodeled discrepancies such as delayed vortex shedding and non-linear fluid–body coupling. However, with continued PPO refinement under CFD feedback, the return recovers within $50{-}100$ episodes and converges close to surrogate-pretrained performance.

Formally, let R_{sur} denote the average return achieved in the surrogate, and $R_{\text{cfd}}^{(0)}$ the return upon first transfer. The initial drop is

$$\Delta R = \frac{R_{\text{sur}} - R_{\text{cfd}}^{(0)}}{R_{\text{sur}}}.$$

With refinement steps k, the recovery trajectory can be modeled as

$$R_{\text{cfd}}^{(k)} \approx R_{\text{sur}} - \Delta R \cdot e^{-\lambda k}, \quad \lambda > 0,$$

illustrating exponential adaptation. This quantifies the surrogate–CFD gap and motivates staged transfer. Instead of a figure, we summarize the transition by approximate statistics and a simple recovery model. At the moment of transfer (k=0), the reward exhibits a sharp decline due to surrogate–CFD mismatch. With continued refinement, the reward gradually approaches its pre-transfer level, reflecting adaptation to high-fidelity flow feedback.

Table 6: Illustrative reward dynamics during surrogate→CFD transfer. Values are normalized and approximate, showing the qualitative trend of drop and recovery.

Iteration k	0 (transfer)	100	300	600
Reward (relative)	∼0.7	∼0.8	∼0.9	∼0.95
Trend	Drop	Partial recovery	Near steady state	Stable

This recovery can be described by an exponential relaxation:

$$R(k) \;\approx\; R_{\text{CFD}} \;+\; \big(R_{\text{sur}} - R_{\text{CFD}}\big)\exp\!\big(-\tfrac{k}{\tau}\big)\,,$$

where R_{sur} is the surrogate reward, R_{CFD} the converged CFD reward, and τ an adaptation constant. The form highlights the characteristic sharp initial drop followed by smooth recovery, consistent with the staged refinement strategy.

E.2 Ablation on Surrogate Design. To validate design choices, we remove key components. Table 7 shows that eliminating partitioning (mode specialization) or residual connections significantly degrades accuracy and rollout stability. In particular, non-partitioned models cannot distinguish frequency-change regimes, producing large rollout drift errors.

Table 7: Ablation results for surrogate network design. Partitioning and residuals are both critical for accuracy and stability.

Variant	Final MAE	Multi-step Rollout Drift (°)	Convergence Speed (Iters)
Partitioned Res-MLP	$\mathbf{1.2}\times10^{-3}$	**0.03**	**100**
No partitioning	3.1×10^{-3}	0.12	250
No residuals	2.6×10^{-3}	0.08	180

E.3 Reward Sensitivity. The reward at time t is formulated as

$$r_t = \alpha\, r_{\text{goal}}(t) - \beta\, c_{\text{eff}}(t),$$

where $r_{\text{goal}}(t)$ encourages timely target-reaching and $c_{\text{eff}}(t)$ penalizes inefficient propulsion. The energetic cost term is defined as

$$c_{\text{eff}}(t) = \frac{P_t}{mg\,U_t},$$

with P_t the instantaneous mechanical power, U_t forward speed, m body mass, and g gravitational acceleration.

By varying (α, β), distinct locomotion strategies emerge:

- $\alpha \gg \beta$: Policies prioritize speed, yielding fast but energetically costly strokes.
- $\beta \gg \alpha$: Policies adopt slower, smoother gaits with minimal cost of transport.
- Balanced (α, β): Policies achieve efficient yet accurate target-reaching, aligning with design objectives.

Sensitivity curves can be approximated by

$$U^\star(\alpha, \beta) \propto \sqrt{\tfrac{\alpha}{\beta}}, \qquad \mathrm{COT}^\star(\alpha, \beta) \propto \tfrac{\beta}{\alpha+\beta},$$

which highlight the trade-off frontier between speed and efficiency. These results demonstrate that reward weighting critically shapes gait emergence, and provide a principled knob for tuning swimming performance.

E.4 Flow-Field Analysis of Frequency Switching. To further illuminate the hysteresis mechanisms described earlier, we analyze the velocity and vorticity fields during controlled low-to-high and high-to-low frequency transitions. This analysis reveals how pre-existing vortical structures strongly influence the emerging wake, providing a physical explanation for the surrogate–CFD discrepancies and for the need for partitioned modeling.

Low-to-high frequency transitions. During a transition from low to high actuation frequency, the initial wake is weak and loosely organized. High-frequency excitation subsequently injects additional momentum into this mildly perturbed region, enabling new vortices to form a coherent and orderly wake with limited interference. However, residual low-frequency vortices introduce small but persistent deflections due to induced lateral velocities. These distortions produce a measurable phase lag between the commanded and realized wake alignment, consistent with the Bode-style hysteresis quantified in Sec. E.1.

High-to-low frequency transitions. In contrast, high-to-low transitions begin with a dense, high-energy wake characterized by small vortex spacing and strong swirling intensities. When low-frequency forcing is applied, the newly formed vortices are heavily affected by the lateral induction of upstream high-frequency structures. This leads to deviation from the central flow axis, wake asymmetry, and reduced coherence. These inherited disturbances amplify the effective recovery time constant τ_{rec} found in Sec. E.1, as the wake requires multiple cycles to shed residual swirl and reorganize into a low-frequency pattern.

Integrated flow insight. Across both transition directions, the wake evolution is governed by the energy content and spatial arrangement of vortices before and after the switching event. The resulting transient states—phase lag, vortex deflection, and localized reverse-flow regions—explain why global surrogates that mix all frequencies struggle to remain stable, whereas frequency-partitioned models more accurately encode these path-dependent dynamics.

The flow visualization in Fig. 9 directly illustrates how pre-switching vortex configurations dictate the transient response after switching. This physical perspective complements our quantitative analysis (Sec. E.1) and provides a clear mechanistic understanding of frequency-switch hysteresis in fish propulsion.

F. Complexity Analysis

F.1 Per-step computational complexity. The computational cost per environment step differs drastically between surrogate inference and CFD simulation. For a residual MLP surrogate with hidden width d_h and depth n:

$$C_{\text{sur}} = O(d_h n).$$

With $d_h = 256$ and $n = 3$, the cost is approximately

$$C_{\text{sur}} \approx 256 \times 3 \approx 768 \text{ FLOPs} \;\Rightarrow\; \text{runtime} \sim 1 \text{ ms/step}.$$

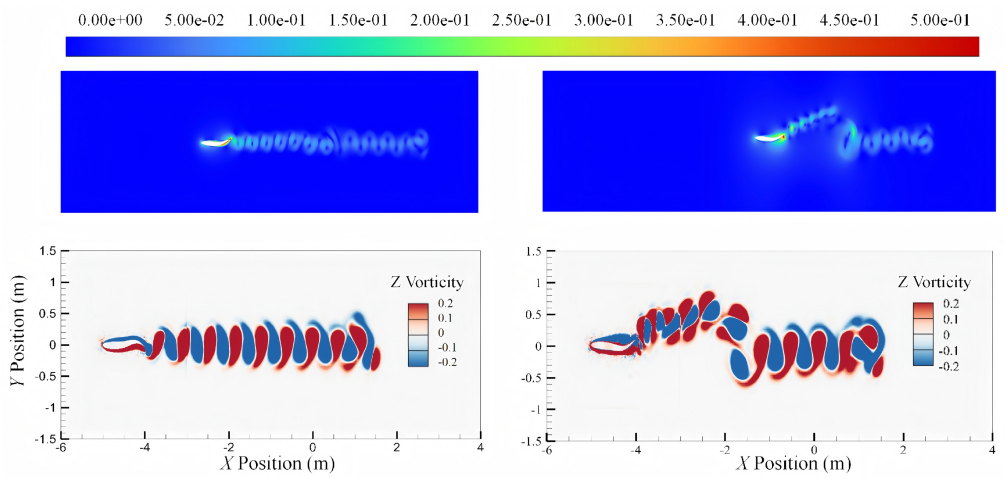

Figure 9: **Flow-field evolution during frequency switching.** (a) Low-to-high transition: a previously weak wake is rapidly reorganized into a high-frequency vortex street, with mild deflections induced by residual low-frequency structures. (b) High-to-low transition: dense, high-energy vortices from the high-frequency regime induce significant lateral forcing on newly generated low-frequency vortices, delaying wake stabilization and producing strong asymmetric patterns.

For CFD, each step requires solving the incompressible Navier–Stokes equations over N_c control volumes:

$$C_{\text{cfd}} = O(N_c \cdot I),$$

where I is the number of solver iterations per time step (typically $20-50$ for residual convergence). With $N_c \sim 1.2 \times 10^6$, this yields $\sim 10^8$ FLOPs per step, corresponding to ~ 1 s/step on 16 CPU cores.

F.2 Relative efficiency. The speedup ratio is

$$S = \frac{C_{\text{cfd}}}{C_{\text{sur}}} \approx \frac{10^8}{10^3} \sim 10^5,$$

at the raw FLOP level. In practice, due to hardware efficiency and communication overhead, we observe

$$S_{\text{empirical}} \sim 10^3,$$

which matches measured wall-clock runtimes. Thus, a single training run requiring 10^6 steps would take:

$$T_{\text{sur}} \approx 10^3 \text{ s } (\sim 17 \text{ min}), \quad T_{\text{cfd}} \approx 10^6 \text{ s } (\sim 11.6 \text{ days}).$$

F.3 Implication. This $\sim 10^3\times$ empirical interaction speedup is the key enabler for feasible reinforcement learning under fluid–dynamic constraints. By allocating the majority of rollouts to surrogates and only scheduling periodic CFD refinements, PD-FS combines:

- **Efficiency:** Orders-of-magnitude acceleration in experience collection.
- **Fidelity:** CFD-based corrections ensure physical consistency.
- **Scalability:** Allows millions of interactions within hours, compared to weeks under CFD-only training.

Table 8: Per-step complexity and runtime comparison (measured on 16-core workstation).

Method	Complexity	Runtime/step	Wall-clock (1M steps)
Surrogate (Res-MLP)	$O(d_h n) \approx 10^3$	$\sim$1 ms	$\sim$17 min
CFD (1.2M cells)	$O(N_c I) \approx 10^8$	$\sim$1 s	$\sim$11.6 days

F.4 Conclusion. The complexity analysis formalizes why surrogate training is indispensable: direct CFD-based DRL is computationally prohibitive, while surrogates reduce per-step cost by three orders of magnitude without discarding physical correction, enabling the staged PD-FS framework to balance efficiency and fidelity.

G. Use of LLMs

During the preparation of this manuscript, we made limited use of a large language model (LLM) Claude and ChatGPT (GPT-4) to assist with linguistic polishing, improving clarity of English phrasing, and suggestions for rephrasing sentences. We emphasize that the LLM was not used to generate the intellectual content, experiment design, results, or conclusions. All output proposed by the model was carefully reviewed, revised, and approved by authors. In particular, we verified each generated sentence for factual correctness, consistency with our data and claims, and checked references for accuracy. The final responsibility for all content in this manuscript rests entirely with the human authors; the LLM is not listed as an author and does not hold any copyright or responsibility.

H. Ethics Statement

This work relies solely on CFD simulations and open-source/licensed tools; it does not involve human or animal subjects or any personally identifiable data. We follow the ICLR Code of Ethics, report methods, ablations, and limitations honestly, and comply with all software/data licenses. The intended use is civilian research on bio-inspired underwater robotics; we discourage harmful or weaponized applications. We disclose computing resources and an estimated carbon footprint in the supplementary material. The authors declare no conflicts of interest.

I. Reproducibility Statement

To enable independent reproduction, the paper and appendix provide: clear algorithmic descriptions and pseudocode, complete hyperparameter. All other information, such as mesh generation and boundary condition specifications, solver versions, etc., are set to default values. These materials are intended to support re-implementation with a surrogate model and licensed CFD solver.