Deep Hydro-Sim: Structured Residual Hydrodynamics for the RL-Based AUV Control

Ruihan Wu[1], Chunying Li[2,4*], Qirong Lei[1], Le Huang[1], Pengcheng Li[1] and Shuxiang Guo[1,2,3*]
[1]The Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China
[2]The Advanced Institute for Ocean Research, Southern University of Science and Technology, Shenzhen, Guangdong 518055, China
[3]The Aerospace Center Hospital, School of Life Science and the Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, Ministry of Industry and Information Technology, Beijing Institute of Technology, Beijing 100081, China
[4]State Key Laboratory of Ocean Sensing, ZJU-Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, Hangzhou, 311215, China
* Corresponding authors: licy@sustech.edu.cn, guo.shuxiang@sustech.edu.cn

***Abstract*—Reinforcement Learning (RL) for Autonomous Underwater Vehicle (AUV) control requires both hydrodynamic fidelity and training throughput, but existing simulators usually trade one for the other: dedicated underwater simulators offer high fidelity but limited RL throughput, whereas efficient engines such as MuJoCo lack native hydrodynamics. To address this issue, Deep Hydro-Sim is proposed as a hybrid Stonefish-MuJoCo framework that distills residual hydrodynamic forces from a high-fidelity simulator into a fast rigid-body simulator through a real-time physics plugin and a physics-informed residual model. In open-loop replay, the learned hydrodynamics reduces surge velocity RMSE from 1.784 m/s to $\sim$1.0 m/s. In SAC training across five seeds, the no-hydrodynamics setting converges quickly under unrealistic dynamics, the baseline MLP diverges in all runs, and the proposed PhysicsFeature-MLP trains stably. Together, these results indicate that cross-simulator hydrodynamics distillation is a practical path toward scalable RL-based AUV control with improved fidelity.**

***Index Terms*—Reinforcement learning (RL), autonomous underwater vehicles (AUVs), residual hydrodynamics, simulator.**

I. Introduction

Autonomous Underwater Vehicles (AUVs) are increasingly used for marine inspection, environmental monitoring, underwater intervention, and ocean exploration, driving demand for reliable underwater motion planning and control systems [1]–[3]. At the same time, learning-based control is becoming an important direction for underwater autonomy [4].

However, directly developing and validating AUV controllers on physical platforms is expensive, time-consuming, and difficult to repeat at scale. Such experiments require hardware preparation, safety supervision, maintenance, and stable test conditions, while Reinforcement Learning (RL) further amplifies this burden because it depends on many interactions. Simulation has therefore become an essential tool for developing underwater controllers and training RL policies.

Recent RL studies have reported promising results on path following, obstacle avoidance, and docking [9]–[11], but their effectiveness still depends strongly on the simulator used for training. Existing underwater simulators still fall into two broad groups. Dedicated underwater simulators such as Stonefish and modular UUV/AUV platforms provide richer marine-task realism, underwater sensing, and hydrodynamic support [5]–[7]. By contrast, general-purpose rigid-body engines such as MuJoCo and the NVIDIA Isaac series are attractive for large-scale RL because of their computational efficiency and mature toolchains [8]. This creates a practical trade-off: realism-oriented underwater simulators better match marine physics, whereas fast RL engines remain physically incomplete for underwater control. Consequently, RL for AUV control still faces a core simulator challenge: achieving sufficient physical fidelity without sacrificing the interaction throughput needed for stable and efficient learning. This mismatch is not a minor modeling detail. Underwater motion depends on strongly coupled six-degree-of-freedom dynamics, velocity-dependent damping, and added-mass effects; policies trained without these forces can exploit unrealistically easy accelerations and control regimes that are unlikely to survive in more faithful simulators or deployment.

To address this gap, this paper focuses on learning residual AUV hydrodynamics through a hybrid Stonefish-to-MuJoCo distillation architecture. Six-degree-of-freedom motion data are collected from Stonefish, residual hydrodynamic forces are computed by subtracting known physics terms, neural networks are trained to predict them, and the learned model is deployed inside MuJoCo as a real-time force plugin. The results show that high aggregate offline regression accuracy does not necessarily lead to high replay fidelity or training stability; preserving coupled and angular hydrodynamic behavior is more important for closed-loop performance. The main contributions of this paper are summarized as follows:

1) A hybrid Stonefish-to-MuJoCo simulator architecture for scalable RL-based AUV control with learned residual hydrodynamics.
2) A comparative residual-model study showing that the physics-informed PhysicsFeature-MLP best preserves

the coupled and angular dynamics that matter most for deployment.

3) A downstream RL verification, with a shared signal-conditioning infrastructure held constant, showing that PhysicsFeature-MLP yields the most stable SAC convergence and supports curriculum training for increasingly complex AUV tasks.
4) To support reproducibility and future research, the project code and curated dataset will be organized and released as open-source resources.

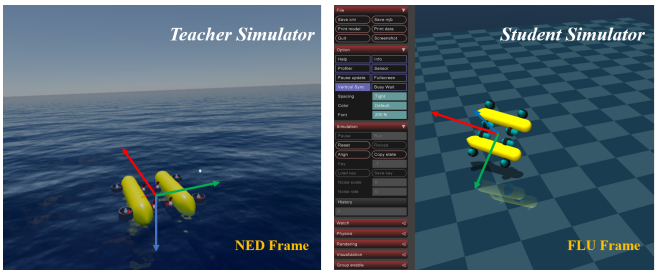
II. Related Work

A. RL for AUV Control. Yu et al. reviewed the growing use of RL in AUV motion systems [4]. Ma et al. proposed a neural-network model-based RL controller for AUV 3-D path following [9]. Tang et al. studied obstacle-aware path planning in unknown environments using improved deep RL [10]. Palomeras and Ridao addressed AUV docking under realistic assumptions with deep RL [11], while Marchel et al. introduced curriculum-based DRL for path planning on simplified electronic navigation charts [16]. These studies demonstrate the task-level potential of RL for underwater systems, but they generally treat the simulator as fixed and do not analyze how hydrodynamic fidelity affects replay behavior or training stability.

B. Underwater Simulators. Cieślak introduced Stonefish as an open-source simulator for marine robotics with dedicated underwater dynamics and sensor support [6]. Z. Zhang et al. presented a modular UUV simulation platform, and X. Zhang et al. developed a MOOS/Unreal-Engine-based AUV simulation system for richer visualization, system integration, and algorithm verification [5], [7]. These platforms improve marine-task realism and engineering usability, but they are not primarily designed for high-throughput RL training. Conversely, fast general-purpose engines such as MuJoCo and the NVIDIA Isaac series are effective back ends for scalable RL workflows, but underwater hydrodynamics must still be modeled externally [8].

C. Hybrid Learning and Transfer. Hybrid-learning work addresses model mismatch from another angle. Zhang et al. proposed a sim-to-real pipeline for underwater obstacle avoidance based on a high-fidelity model [12]. Gao et al. showed that learned residual physics can improve transfer performance for soft robots [13]. However, these studies do not examine cross-simulator hydrodynamics distillation from a high-fidelity underwater simulator to a faster RL engine for AUV control.

Gap. Taken together, existing work advances RL tasks, simulator development, and residual learning, but usually studies them separately. To the best of our knowledge, prior work has not jointly examined learned residual hydrodynamics distillation from a high-fidelity underwater simulator to a fast RL simulator and its consequences for both open-loop replay fidelity and closed-loop AUV RL stability.



(a) Stonefish (b) MuJoCo

Fig. 1. AUV scenes in the Stonefish simulator and the MuJoCo simulator.

III. Hybrid Simulator Architecture

In the overall pipeline, Stonefish serves as the teacher simulator and operates in a NED (North-East-Down) coordinate frame at 50 Hz. A hybrid excitation strategy generates 6-DOF motion data covering single-axis sweeps, coupled maneuvers, and stochastic exploration. Residual hydrodynamic forces are computed by subtracting known inertial terms from thruster forces. A neural network then learns this residual mapping and is embedded into a MuJoCo physics plugin, enabling Soft Actor-Critic (SAC) [15] training at 25 Hz control with 500 Hz physics. Fig. 1 shows the AUV in both simulators.

A. Dynamic Modeling

Following standard marine-craft notation [14], the 6-DOF rigid-body dynamics of an AUV in body frame are:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau}_{\text{prop}} \tag{1}$$

where $\mathbf{M}$ is the system inertia matrix (rigid body + added mass), $\mathbf{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $\mathbf{D}(\boldsymbol{\nu})$ is the hydrodynamic damping matrix, $\mathbf{g}(\boldsymbol{\eta})$ captures gravitational and buoyancy restoring forces, $\boldsymbol{\tau}_{\text{prop}}$ is the propulsion force, and $\boldsymbol{\nu} = [u, v, w, p, q, r]^T$ is the body-frame velocity vector. For implementation clarity, we write the Coriolis term as

$$\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{C}_{\text{RB}}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{C}_{\text{A}}(\boldsymbol{\nu})\boldsymbol{\nu} \tag{2}$$

where $\mathbf{C}_{\text{RB}}$ denotes rigid-body Coriolis/centripetal effects and $\mathbf{C}_{\text{A}}$ denotes the added-mass contribution. In teacher-side label construction, $\mathbf{M}$ is treated as a constant; any remaining state-dependent added-mass mismatch is absorbed into the learned residual. Acceleration is included as an input because it is informative for these omitted inertial effects.

We rearrange (1) to isolate the residual hydrodynamic term:

$$\boldsymbol{\tau}_{\text{res}} = \boldsymbol{\tau}_{\text{prop}} - \mathbf{M}\dot{\boldsymbol{\nu}} - \mathbf{g}(\boldsymbol{\eta}) \approx \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} \tag{3}$$

Thus, $\boldsymbol{\tau}_{\text{res}}$ is the total non-restoring residual force represented by the teacher data; it is not a pure damping term.

A neural network f_θ learns the mapping from the current kinematic state to residual forces:

$$f_\theta : [\boldsymbol{\nu},\, \dot{\boldsymbol{\nu}}] \in \mathbb{R}^{12} \rightarrow \boldsymbol{\tau}_{\text{res}} \in \mathbb{R}^6 \tag{4}$$

with the 12-dimensional input $[\,u, v, w, p, q, r, \dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}\,]$, 6-dimensional output $[F_x, F_y, F_z, \tau_\phi, \tau_\theta, \tau_\psi]$, and training data

from the Stonefish dataset described in Section IV. The model is trained by minimizing $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{\tau}_{\text{res}}^{(i)} - f_\theta(\mathbf{x}^{(i)})\|^2$.

B. MuJoCo Hydro Plugin Integration

The learned residual model is deployed through a MuJoCo physics plugin that applies a fixed signal-conditioning pipeline at each control step (25 Hz). This infrastructure is shared across all RL conditions, including the no-hydrodynamics baseline with zero network output, so that it does not confound the comparison of residual models. The pipeline consists of:

1) A Kalman filter estimates body-frame acceleration $\dot{\boldsymbol{\nu}}$ from velocity measurements, reducing the noise amplification that would arise from direct finite differencing.
2) Coordinate conversion applies a FRD↔FLU body-frame mask $[1, -1, -1, 1, -1, -1]$ to align Stonefish-trained models with MuJoCo's frame convention.
3) The neural network predicts the residual force $\boldsymbol{\tau}_{\text{ML}} = f_\theta(\boldsymbol{\nu}, \dot{\boldsymbol{\nu}})$.

Restoring forces $\mathbf{g}(\boldsymbol{\eta})$ remain native MuJoCo terms based on the Center of Mass (CoM) and Center of Buoyancy (CoB). The Coriolis treatment follows the decomposition in (2): MuJoCo already contributes the rigid-body term $\boldsymbol{\tau}_{C,\text{RB}}^{\text{mj}}$, whereas the learned target in (3) represents the full $\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu}+\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$. To avoid double counting, the external force injected into MuJoCo is

$$\boldsymbol{\tau}_{\text{ext}} = \boldsymbol{\tau}_{\text{ML}} - \boldsymbol{\tau}_{C,\text{RB}}^{\text{mj}}, \tag{5}$$

so that the simulator sees

$$\boldsymbol{\tau}_{C,\text{RB}}^{\text{mj}} + \boldsymbol{\tau}_{\text{ext}} \approx \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}. \tag{6}$$

This formulation makes the network role explicit: it replaces the missing underwater nonlinear force with the teacher-consistent residual, while MuJoCo retains native rigid-body and restoring mechanics. A low-pass filter ($\beta = 0.9$), per-axis force clipping, and a velocity deadband that smoothly attenuates forces near zero velocity are applied as standard signal conditioning. These components mitigate common numerical artifacts associated with injecting external forces into a rigid-body engine, including high-frequency oscillations, outlier spikes, and chattering near rest.

Remark on acceleration estimation. The filtered acceleration plugin remains an explicit delayed feedback system because $\dot{\boldsymbol{\nu}}$ is influenced by previously applied residual forces. The filter removes the same-step algebraic dependence that would arise if raw MuJoCo acceleration were fed directly into $f_\theta(\boldsymbol{\nu}, \dot{\boldsymbol{\nu}})$. Consequently, the closed loop is delayed and bounded rather than instantaneous.

C. Residual Model Architectures

Five architectures share the same 12→6 input-output contract (Table I):

The **PhysicsFeature-MLP** augments the raw kinematic state with physics-informed features such as quadratic drag terms $v_i|v_i|$ and cross-axis products before a standard MLP. This design injects known hydrodynamic structure into the input space, allowing the network to focus on residual corrections instead of rediscovering basic drag relationships from data. All models use the same data split, normalization, optimizer (Adam), and training budget.

TABLE I
CANDIDATE RESIDUAL MODEL ARCHITECTURES

Model	Architecture	Params	Key Idea
Baseline MLP	12→128→256→128→6	42K	Feedforward
ResMLP-12×6	Proj→4 res. blocks→6	130K	Skip + LayerNorm
MultiHead	Trunk + 6 heads	114K	Per-axis heads
MoE-HydroNet	4 experts + gate	77K	Regime gating
PhysFeature	Engineered→MLP	73K	$v\|v\|$ features

D. RL Task Design and Curriculum

The RL environment uses MuJoCo at 500 Hz physics with 25 Hz control (frame skip 20), with episodes of up to 500 steps. Policies are trained with SAC, an off-policy maximum-entropy algorithm well suited to continuous control. The agent commands 6-DOF normalized actions $\mathbf{a} \in [-1, 1]^6$ mapped to 8 thrusters via a linear mixing matrix.

The 36-dimensional observation vector is:

$$\mathbf{o} = [\mathbf{R}^T(\mathbf{p}_g - \mathbf{p}),\ \mathbf{v},\ \boldsymbol{\omega},\ \mathbf{q},\ \mathbf{R}^T\mathbf{g},\ d,\ \mathbf{s}_{1:15},\ h,\ \mathbf{a}_{\text{imu}}] \tag{7}$$

including body-frame target direction (3), Doppler Velocity Log (DVL) velocity (3), gyroscope (3), quaternion (4), gravity-in-body (3), depth (1), 15-beam sonar (15), altitude (1), and accelerometer (3). Sensor noise is injected for robustness: DVL ± 0.05 m/s, depth ± 0.02 m, sonar ± 0.05 m, and gyro ± 0.01 rad/s.

The reward function uses potential-based reward shaping (PBRS) with distance potential $\varphi(s) = -(\|\mathbf{p} - \mathbf{p}_g\|/d_{\max}) \cdot \varphi_s$:

$$R_t = \underbrace{[\varphi(s_t) - \varphi(s_{t-1})]}_{\text{shaping}} + w_a \tfrac{\cos\theta_a + 1}{2} + w_u \tfrac{\cos\theta_u + 1}{2} + R_s - C \tag{8}$$

where θ_a is the heading alignment angle, θ_u is the upright angle, R_s is the success bonus, and C aggregates costs for obstacle proximity, energy, action magnitude, and smoothness.

A three-stage curriculum, motivated by progressively increasing task difficulty [16], advances when the agent achieves 97% success over 150 episodes:

Stage 1 is a 3D stochastic point-to-point navigation task designed to enable the AUV to master attitude stabilization and basic motion control maneuvers—such as forward movement, turning, and hovering.

Stage 2 adds an exponential obstacle penalty: $C_{\text{obs}} = w_d \exp(-1.5(d_{\min} - d_c))$ for sonar distances in $[d_c, d_w]$ with $d_c = 0.4$ m and $d_w = 4.0$ m, and a hard penalty w_{col} for $d_{\min} < d_c$. The obstacle curriculum itself progresses from an empty scene to an 8-pillar slalom.

Stage 3 introduces an A* path planner (0.05 m grid, 1.2 m safety margin, 8-directional connectivity) that generates a reference trajectory. The agent tracks a lookahead point 3.5 m ahead using exponential moving average (EMA) smoothing

($\alpha = 0.15$). A Huber-like cross-track error (CTE) penalty prevents both small deviations and training instability:

$$C_{\text{CTE}} = w_c \cdot \begin{cases} e^2 & \text{if } |e| < 1\,\text{m} \\ 2|e| - 1 & \text{otherwise} \end{cases} \tag{9}$$

IV. Data and Experimental Protocol

A. Stonefish Data Collection

The AUV is an 8-thruster underactuated platform (mass 151 kg, 4 horizontal + 4 vertical thrusters, max thrust 155 N each). Data were collected in Stonefish at 50 Hz using a hybrid excitation strategy: (1) single-axis sinusoidal sweeps on surge, sway, heave, and yaw; (2) coupled maneuvers including circles, corkscrew, and drift turns; (3) stochastic exploration via the Ornstein-Uhlenbeck random walk. Each session recorded 20 channels: 6-DOF pose, 6-DOF body velocity, and 8 motor commands. 857,542 samples were collected.

B. Residual Force Construction

Velocities were smoothed with a Kalman filter, and accelerations were estimated from the filtered velocity signal via finite differences. Propulsion forces were then reconstructed from the recorded 8-channel motor commands using the same thruster allocation employed in Stonefish,

$$\boldsymbol{\tau}_{\text{prop}} = \mathbf{B}\mathbf{f}_{\text{thr}}, \qquad \mathbf{f}_{\text{thr}} = [f_1, \ldots, f_8]^T. \tag{10}$$

To reduce label mismatch around zero crossings and thrust reversals, the reconstruction uses the same first-order actuator dynamics and small command deadband as the simulator-side thruster layer. Incorporating these actuator details ensures consistency between the teacher and student labels.

The residual target was finally computed as

$$\boldsymbol{\tau}_{\text{res}} = \boldsymbol{\tau}_{\text{prop}} - \mathbf{M}\dot{\boldsymbol{\nu}} - \mathbf{g}(\boldsymbol{\eta}), \tag{11}$$

where $\mathbf{M}$ includes the rigid-body and added-mass terms used in the label construction. The MuJoCo student uses the same 8-thruster actuation interface rather than instantaneous body-force injection, preserving comparable thrust transients during replay and RL training.

C. Training Protocol

An 80/20 split was used for training and testing. To prevent temporal leakage, the dataset was split by contiguous segments rather than individual samples. Specifically, it was partitioned into blocks of at least 150 consecutive timesteps, and whole blocks were assigned to either the training or test set. This strategy prevents temporally correlated samples from being shared across sets and avoids the overly optimistic metrics produced by naive per-sample random splitting of time-series data. Inputs and outputs were independently standardized (zero mean, unit variance). All five models were trained with Adam (lr $= 10^{-3}$, weight decay 10^{-5}), gradient clipping at 5.0, early stopping with patience of 30, and a ReduceLROnPlateau scheduler.

For SAC training, we used 8 parallel environments, batch size 512, replay buffer 10^6, and learning rate 3×10^{-4}.

TABLE II
Offline Residual Model Comparison

Model	Test RMSE	Test MAE	Mean R^2
Baseline MLP	11.81	5.94	0.9824
PhysicsFeature-MLP	**8.46**	**4.14**	**0.9902**
MultiHead-HydroNet	11.94	6.11	0.9875
MoE-HydroNet	12.39	5.96	0.9716
ResMLP-12×6	13.46	6.71	0.9896

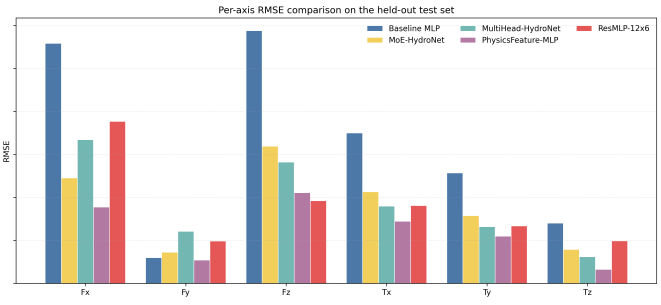


Fig. 2. Per-axis RMSE of the five residual models.

Each of the three physics configurations was trained with five independent random seeds; all seeds were retained for analysis to assess result consistency. This setting was chosen to balance sample efficiency with the higher per-step cost introduced by hydro-enhanced simulation.

V. Results and Discussion

A. Offline Residual Model Comparison

Table II reports test-set metrics. PhysicsFeature-MLP achieves the best aggregate one-step metrics, indicating that high-capacity architectures can fit the teacher data well offline. However, deployment decisions cannot rely on scalar RMSE or mean R^2 alone: the Baseline MLP still reports a high mean $R^2 = 0.9824$, yet later destabilizes RL. This mismatch indicates that closed-loop training stability is more sensitive to structured per-axis errors and extrapolation behavior than to a single aggregate regression score.

Fig. 2 shows the per-axis breakdown. The most decision-relevant difference appears on the coupled angular channels, where Coriolis effects and cross-axis interactions are strongest. PhysicsFeature-MLP reduces these errors more consistently than the plain Baseline MLP, helping explain why it remains stable after deployment even though aggregate offline metrics alone would not have selected it.

B. Open-Loop Fidelity Replay

To test whether offline force accuracy translates to trajectory-level fidelity, a 90 s nine-segment command sequence was replayed in Stonefish and MuJoCo using five plugin variants: MJ0 (no learned hydro), MJ1 (Baseline MLP), MJ2 (ResMLP), MJ3 (MultiHead), and MJ4 (PhysicsFeature-MLP). The sequence includes surge/heave/yaw sinusoids and coupled circular and drift maneuvers.

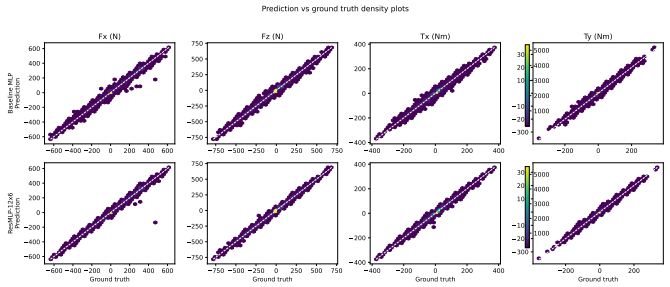


Fig. 3. Prediction-ground-truth density plots on selected axes. Top: Baseline MLP. Bottom: ResMLP-12×6.

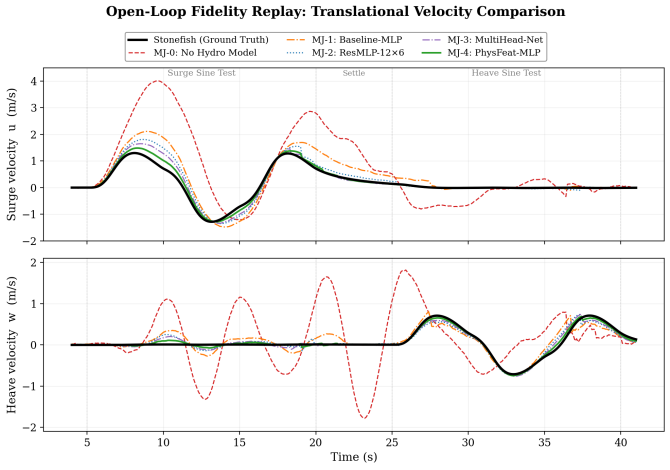


Fig. 4. Surge and heave velocity overlay: Stonefish vs. MuJoCo variants. Learned-hydro variants track Stonefish more closely than MJ0.

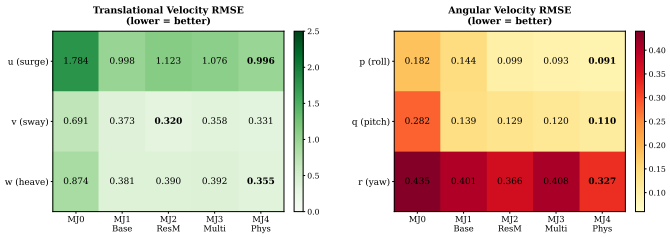


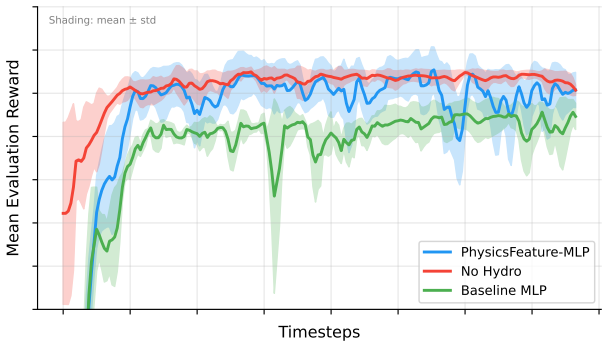
Fig. 5. Per-axis velocity RMSE heatmap for translational and angular axes. Lower is better.

Fig. 4 shows a representative result: all learned-hydro variants substantially outperform MJ0. Surge RMSE drops from 1.784 to 0.996 in the best setting, a 44% reduction that demonstrates the value of distillation.

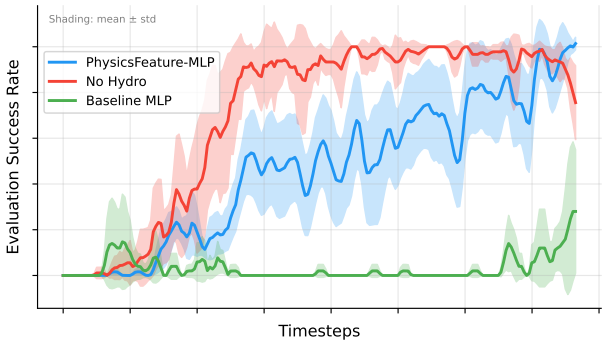
Table III and Fig. 5 provide the full breakdown. Overall, MJ4 is selected for the subsequent RL experiments because it provides the most balanced trajectory-level fidelity across the full 6-DOF motion space, with clear gains on sway and angular channels. This is the more relevant criterion for downstream control because attitude regulation, turning, and coupled maneuvering depend on these dynamics. The largest remaining mismatches appear in coupled segments, the most demanding regime for a memoryless residual model.

TABLE III
OPEN-LOOP REPLAY RMSE (VELOCITY, LOWER IS BETTER)

Axis	MJ0	MJ1 (Base)	MJ4 (PhysFeat)	Best
u (surge)	1.784	**0.996**	0.998	MJ1
v (sway)	0.691	0.373	0.331	MJ2
w (heave)	0.874	0.381	**0.355**	MJ4
p (roll)	0.182	0.144	**0.093**	MJ4
q (pitch)	0.282	0.139	**0.110**	MJ4
r (yaw)	0.435	0.401	**0.327**	MJ4



(a) Mean evaluation reward



(b) Evaluation success rate

Fig. 6. SAC training curves under three MuJoCo physics configurations (5 seeds). Solid lines show means; shaded bands show ±1 standard deviation.

C. RL Training Study

To assess the proposed hybrid simulator at the system level, SAC was trained under three MuJoCo physics configurations sharing the same signal-conditioning infrastructure. Each configuration used five independent random seeds, and the residual model was the only varying component. R_0 sets the network output to zero, R_{MLP} uses the Baseline MLP, and R_{PhysFeat} uses the replay-selected PhysicsFeature-MLP.

Fig. 6 shows the training curves across five seeds for each setting. R_0 converges fastest under unrealistically permissive dynamics. R_{MLP} is unstable and diverges across all seeds, indicating that high aggregate offline scores alone do not guarantee stable deployment. The Baseline MLP appears to accumulate structured errors in attitude and turning dynamics, shifting the state distribution and producing increasingly aggressive residual outputs. By contrast, R_{PhysFeat} remains stable and achieves the best task performance. This result shows

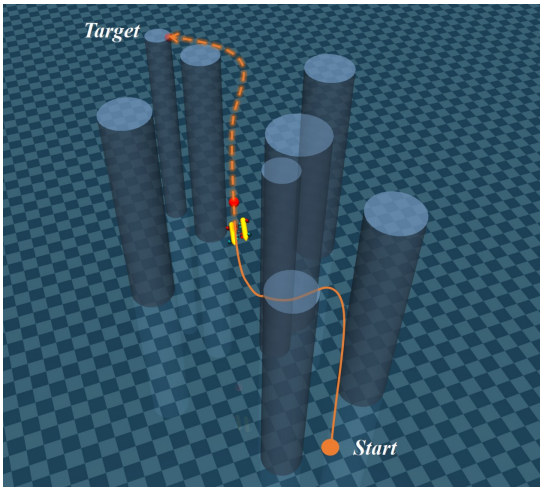


Fig. 7. Stage 3: A*-guided navigation in a complex environment.

that accurate angular and coupled residual forces matter more than scalar regression metrics. Its physics-informed features better preserve the lateral and angular dynamics identified in the replay study, improving extrapolation and closed-loop damping during aggressive maneuvers. Because the signal-conditioning pipeline is fixed across all conditions, the performance differences can be attributed to residual-model quality.

Fig. 7 illustrates the final curriculum stage for A*-guided navigation. The scene contains randomly generated columnar obstacles. The RL controller handles attitude control and real-time tracking based on a forward-facing red sphere.

Discussion and Limitations. No real-world sim-to-real validation was performed, so the present claims remain limited to simulator-to-simulator fidelity and RL trainability. The RL comparison uses five seeds per condition, which establishes consistent qualitative trends but remains modest for statistical inference. The curriculum results should be interpreted as a feasibility demonstration of stage progression rather than a full stage-wise ablation.

VI. Conclusion

This paper has presented Deep Hydro-Sim, a hybrid Stonefish-to-MuJoCo framework that learns residual AUV hydrodynamics and verifies them through the RL-based AUV control. Three main findings emerge: (1) aggregate offline accuracy alone is not a sufficient model-selection criterion, because the physics-informed PhysicsFeature-MLP more faithfully captures the coupled and angular dynamics that matter for deployment; (2) open-loop replay shows that learned hydrodynamics reduces surge velocity RMSE by 44% relative to the no-hydro baseline, with PhysicsFeature-MLP providing the strongest overall fidelity on angular channels; (3) a controlled RL comparison across five independent seeds shows that the Baseline MLP diverges because of compounding prediction errors in the delayed feedback loop, whereas PhysicsFeature-MLP maintains stable SAC convergence and supports curriculum training for increasingly complex AUV tasks.

Future Work. Promising next steps include temporal models for history-dependent hydrodynamics, broader multi-platform evaluation, and eventual hardware validation.

Acknowledgment

This work was supported in part by the GuangDong Basic and Applied Basic Research Foundation General Program under Grant 2025A1515011007, in part by Shenzhen Science and Technology Program under Grant RCBS20231211090725048, in part by the High level of special funds from Southern University of Science and Technology under Grant G03034K003, and in part by the Open Project of the State Key Laboratory of Ocean Sensing under Grant OSKF-2025Z04.

References

[1] C. Li, S. Guo, and J. Guo, "Tracking control in presence of obstacles and uncertainties for bioinspired spherical underwater robots," *J. Bionic Eng.*, vol. 20, no. 1, pp. 323–337, 2023.

[2] C. Li, S. Guo, and J. Guo, "Study on obstacle avoidance strategy using multiple ultrasonic sensors for spherical underwater robots," *IEEE Sensors J.*, vol. 22, no. 24, pp. 24458–24470, 2022.

[3] A. Li, S. Guo, and C. Li, "An improved motion strategy with uncertainty perception for the underwater robot based on thrust allocation model," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 64–71, 2025.

[4] T. Yu, Q. Zhang, and T. Liu, "Reinforcement learning approaches in the motion systems of autonomous underwater vehicles," *Appl. Ocean Res.*, vol. 161, p. 104682, 2025.

[5] Z. Zhang, W. Mi, J. Du, Z. Wang, W. Wei, Y. Zhang, Y. Yang, and Y. Ren, "Design and implementation of a modular UUV simulation platform," *Sensors*, vol. 22, no. 20, p. 8043, 2022.

[6] P. Cieślak, "Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface," in *Proc. OCEANS 2019-Marseille*, Marseille, France, 2019, pp. 1–8.

[7] X. Zhang, Y. Fan, H. Liu, Y. Zhang, and Q. Sha, "Design and implementation of autonomous underwater vehicle simulation system based on MOOS and Unreal Engine," *Electronics*, vol. 12, no. 14, p. 3107, 2023.

[8] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vilamoura, Portugal, 2012, pp. 5026–5033.

[9] D. Ma, X. Chen, W. Ma, H. Zheng, and F. Qu, "Neural network model-based reinforcement learning control for AUV 3-D path following," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 893–904, 2023.

[10] Z. Tang, X. Cao, Z. Zhou, Z. Zhang, C. Xu, and J. Dou, "Path planning of autonomous underwater vehicle in unknown environment based on improved deep reinforcement learning," *Ocean Eng.*, vol. 301, p. 117547, 2024.

[11] N. Palomeras and P. Ridao, "Autonomous underwater vehicle docking under realistic assumptions using deep reinforcement learning," *Drones*, vol. 8, no. 11, p. 673, 2024.

[12] S. Zhang, L. Zhang, and Y. Chen, "Sim-to-real pipeline for training autonomous obstacle avoidance of underwater robots based on high-fidelity model," *Robotica*, vol. 43, no. 8, pp. 2952–2974, 2025.

[13] J. Gao, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real of soft robots with learned residual physics," *IEEE Robot. Autom. Lett.*, vol. 9, no. 10, pp. 8523–8530, 2024.

[14] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[15] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 1861–1870.

[16] Ł. Marchel, R. Kot, P. Szymak, and P. Piskur, "Model-based AUV path planning using curriculum learning and deep reinforcement learning on a simplified electronic navigation chart," *Appl. Sci.*, vol. 15, no. 11, p. 6081, 2025.