EasyUUV: An LLM-Enhanced Universal and Lightweight Sim-to-Real Reinforcement Learning Framework for UUV Attitude Control

Anonymous Authors

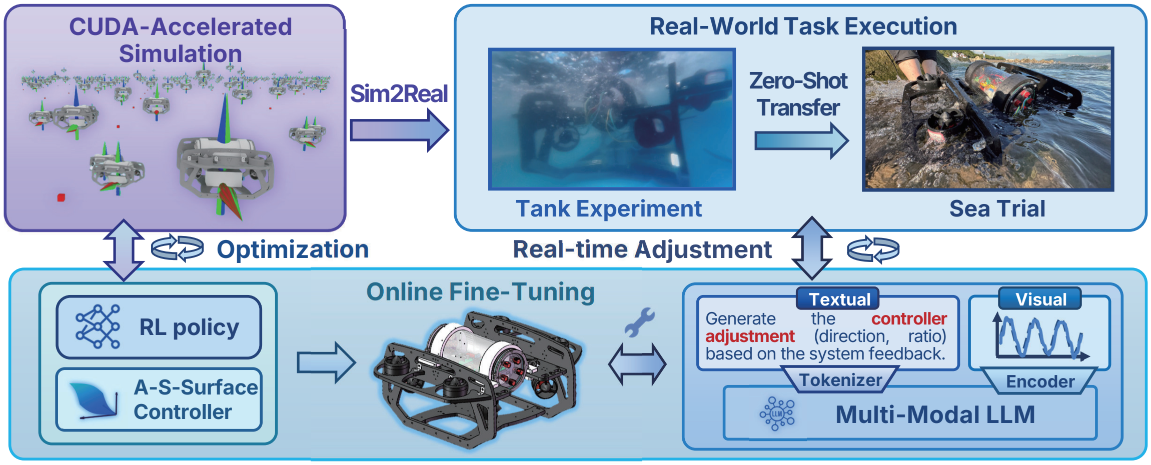


Fig. 1: **Illustration of our developed EasyUUV framework**. EasyUUV is an LLM-enhanced universal and lightweight Sim2Real RL framework for UUV attitude control, which trains the expert policy via RL in parallelized simulation, while transferring it to a real UUV platform. A multimodal LLM agent further adapts controller parameters using dynamics and sensor feedback for robust performance.

***Abstract*— Despite recent advances in Unmanned Underwater Vehicle (UUV) attitude control, existing methods still struggle with generalizability, robustness to real-world disturbances, and efficient deployment. To address the above challenges, this paper presents EasyUUV, a Large Language Model (LLM)-enhanced, universal, and lightweight simulation-to-reality reinforcement learning (RL) framework for robust attitude control of UUVs. EasyUUV combines parallelized RL training with a hybrid control architecture, where a learned policy outputs high-level attitude corrections executed by an adaptive S-Surface controller. A multimodal LLM is further integrated to adaptively tune controller parameters at runtime using visual and textual feedback, enabling training-free adaptation to unmodeled dynamics. Also, we have developed a low-cost 6-DoF UUV platform and applied an RL policy trained through efficient parallelized simulation. Extensive simulation and real-world experiments validate the effectiveness and outstanding performance of EasyUUV in achieving robust and adaptive UUV attitude control across diverse underwater conditions. The source code is available in the following repository: `https://anonymous.4open.science/r/easyuuv/`.**

I. INTRODUCTION

Unmanned Underwater Vehicles (UUVs) are transforming underwater operations, playing critical roles in marine research [1], environmental monitoring [2], and resource exploration [3]. However, achieving robust and intelligent autonomy for UUVs—particularly in attitude control—remains an open challenge. UUVs operate in complex, highly dynamic, and partially observable environments, where nonlinear hydrodynamics, ocean currents, and wave disturbances introduce significant uncertainty. These factors complicate the design of reliable attitude control systems, which are essential for high-stakes missions such as coral reef navigation [4], pipeline inspection [5], and sample retrieval [6].

Traditional and mainstream controllers—such as PID [7], Model Predictive Control (MPC) [8], Sliding Mode Control (SMC) [7], and Fuzzy Logic Control (FLC) [9]—offer partial solutions but are hindered by their reliance on accurate dynamics modeling or limited adaptivity. Their performance often degrades in uncertain conditions due to modeling inaccuracies, hysteresis, and control overshoots, raising risks in unstructured real-world deployments [10], [11].

Reinforcement Learning (RL), by contrast, has emerged as a promising data-driven alternative for autonomous agents to learn robust control policies through interaction [12]. In particular, unlike traditional controllers that have inadequate adaptivity or depend on accurate system identification, RL can learn directly from experience, thereby eliminating the need for precise hydrodynamic modeling and enabling end-to-end optimization toward task objectives [13]. As a result, RL is especially well-suited for underwater environments, which are nonlinear, partially observable, and difficult to model analytically. Building on this advantage, RL's adaptability to high-dimensional, nonlinear dynamics has motivated extensive UUV-related research [14], [15], [16]. Nevertheless, despite these strengths, three major challenges remain: the simulation-to-reality (Sim2Real) gap, limited generalizability, and deployment inefficiency. To address these issues, domain randomization can partially mitigate model mismatch by injecting variability into simulation [14]; however, it still cannot fully prevent instability under attitude perturbations or parameter shifts [17]. Moreover, the high computational cost of RL training and the lack of generalizable hydrodynamic/thruster models further constrain its practical application across diverse UUV platforms.

In addition to the challenges above, real-world deployment of RL-based controllers often requires extensive manual tuning to account for variations in vehicle dynamics, environmental conditions, and sensor noise—especially when transitioning across different UUV platforms or operational domains [18]. This lack of adaptability not only hinders scalability but also increases the risk of degraded performance or mission failure in unfamiliar conditions [19]. Fortunately, the introduction of the large language model (LLM) enables online, training-free adaptation of controller parameters [20]. By leveraging historical system trajectories, real-time sensory feedback, and task-specific context encoded in both visual and textual forms, the LLM can dynamically adjust key control parameters without interrupting operation [21]. This capability enhances the robustness and generalizability of the control system, allowing a single RL-trained policy to maintain stable performance across diverse and uncertain underwater environments [22].

Based on the above analysis, we develop EasyUUV, a lightweight and universal Sim2Real RL framework enhanced with LLM. By combining parallelized RL training with LLM-driven adaptation, it helps bridge the Sim2Real gap for scalable deployment across platforms. The hybrid architecture employs an RL policy for high-level corrections executed by a nonlinear adaptive S-Surface (A-S-Surface) controller, ensuring robust control under 6-DoF coupling. EasyUUV leverages an Isaac Lab [23]-based simulation with hydrodynamic models for efficient parallel training and fast GPU-based convergence. At runtime, a multimodal LLM agent adjusts controller parameters using visual and textual feedback, maintaining performance under unmodeled dynamics, noise, and actuator drift, thus providing a generalizable solution for real-world UUV attitude control.

Our main contributions are summarized as follows:

- **A universal and lightweight RL-based control framework:** EasyUUV supports scalable Sim2Real attitude control through platform-agnostic modeling and a CUDA-accelerated parallelized RL training architecture. To enhance control performance under noise and disturbances, we further develop an A-S-Surface controller that integrates nonlinear control and adaptive compensation for improved robustness.
- **LLM-driven adaptive controller tuning:** EasyUUV integrates a multimodal LLM-based module that adaptively adjusts controller parameters at runtime based on historical dynamic responses and real-time sensory feedback, enabling robust adaptation without retraining.
- **Zero-shot transfer and extensive experiments:** We develop a low-cost UUV platform integrated with our LLM-enhanced Sim2Real RL framework, enabling zero-shot transfer of expert policies from simulation to reality. Validated through tank experiments and sea trials, EasyUUV showcases outperforming robustness and adaptivity across diverse conditions in attitude control.

II. ARCHITECTURE AND MODULES

In this section, we introduce the EasyUUV framework in detail, including both simulation and hardware platforms.

A. Architecture Overview

As shown in Fig. 2, the proposed EasyUUV framework integrates three components: a composite controller combining an RL policy, a nonlinear A-S-Surface module, and LLM-based adaptation; a parallelized RL training environment with hydrodynamic and thruster modeling; and a Sim2Real deployment pipeline for real-world adaptation. The observation vector includes the target attitude, current attitude, and depth offset, while the RL policy outputs deviation commands that the A-S-Surface controller converts into control inputs and PWM signals to drive the thrusters.

The RL policy is trained in a high-fidelity simulation built on NVIDIA Isaac Lab with MuJoCo-based hydrodynamics [24], where domain randomization (DR) over parameters such as COB–COM offsets [14] and thruster nonlinearities improves generalization, and GPU acceleration ensures rapid convergence. After training, the policy is deployed directly to real UUVs without fine-tuning, while at runtime a multimodal LLM agent enhances adaptability by adjusting controller parameters in real time based on visual dynamics and textual sensor feedback, ensuring robust zero-shot transfer and reliable performance in diverse underwater conditions.

B. Simulation Platform and Controller Design

A carefully designed simulation platform is therefore essential for achieving the above capabilities. In the following, we develop a platform that enables efficient RL training and zero-shot policy transfer to real UUVs through simplified, hardware-agnostic modeling integrated with NVIDIA Isaac Lab, while also supporting our LLM-enhanced RL-based control method for robust and adaptive UUV attitude control.

For **hydrodynamic modeling**, we adopt MuJoCo-based phenomenological models [24] to simulate rigid-body interactions in fluid. Each object is approximated by an equivalent inertia box computed from its mass m and inertia tensor $\mathcal{I}$, with half-dimensions r_i that can be computed as follows:

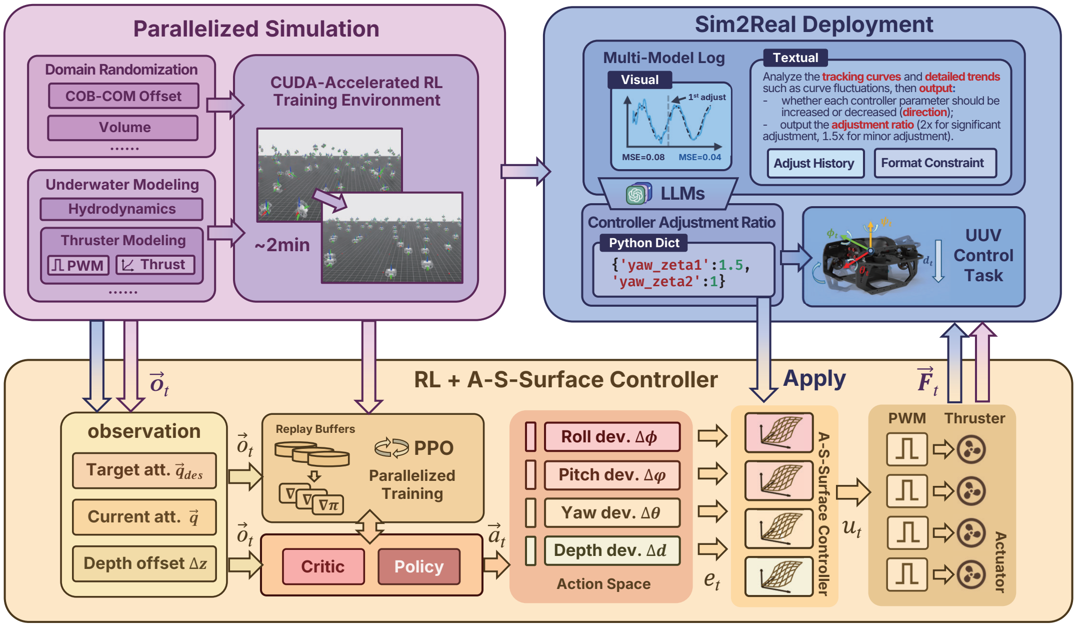
$$r_i = \sqrt{\frac{3}{2m}(\mathcal{I}_{jj} + \mathcal{I}_{kk} - \mathcal{I}_{ii})}, \tag{1}$$

which enables the calculation of total fluid forces $\mathbf{f}_{\text{inertia}} = \mathbf{f}_D + \mathbf{f}_V$ and torques $\mathbf{g}_{\text{inertia}} = \mathbf{g}_D + \mathbf{g}_V$, incorporating both drag and viscous effects. Drag forces and torques are modeled as $f_{D,i} = -2\rho r_j r_k |v_i| v_i$ and $g_{D,i} = -\frac{1}{2}\rho r_i (r_j^4 + r_k^4)|\omega_i|\omega_i$, while viscous terms are $f_{V,i} = -6\beta\pi r_{\text{eq}} v_i$ and $g_{V,i} = -8\beta\pi r_{\text{eq}}^3 \omega_i$, with $r_{\text{eq}} = (r_x + r_y + r_z)/3$ and β denoting the fluid viscosity.

For **thruster dynamics**, we implement a realistic actuation pipeline that modulates thrust via PWM signals to electronic speed controllers. Based on empirical data from Blue Robotics T200 thrusters at 16V [25], the thrust output τ_Ω (in N) is modeled as a function of normalized input $a \in [-1, 1]$, corresponding to 1100–1900 µs PWM, using a piecewise quadratic fit:

$$\tau_\Omega = \begin{cases} 29.54a^2 + 26.10a - 2.44, & a \in (0.08, 1], \\ 0, & a \in [-0.08, 0.08], \\ -21.75a^2 + 21.75a + 2.07, & a \in [-1, -0.08). \end{cases} \tag{2}$$

To improve policy generalization and real-world adaptability, we apply **domain randomization** within a high-efficiency, parallelized simulation environment [26]. During training, key parameters such as the COB–COM offset,



Analyze the tracking curves and detailed trends such as curve fluctuations, then output:
- whether each controller parameter should be increased or decreased (direction);
- output the adjustment ratio (2x for significant adjustment, 1.5x for minor adjustment).

```
{'yaw_zeta1':1.5,
'yaw_zeta2':1}
```

Fig. 2: Architecture of the EasyUUV framework, which comprise three parts: (a) RL and A-S-Surface-based composite controller module; (b) Parallelized RL training simulation environment developed on Isaac Lab; and (c) Sim2Real deployment module for real-world adaption.

TABLE I: Domain Randomization Configuration Details

Parameters	Distributions	Values (Low, High)
COB-COM offset (m)	Uniform Sphere	(0.075, 0.15)
Volume (L)	Uniform	(1.5, 3)
Controller Gain	Uniform	(15, 30)% of the relative value

volume, and controller gains are randomly perturbed to account for structural and dynamic variations. For instance, the COB–COM offset affects torques from gravity and buoyancy—critical for attitude control. All simulation models are implemented in Isaac Lab and trained with GPU acceleration. The full set of randomized parameters is listed in Table I.

Building on this simulation environment, we implement an **RL policy** using the RSL-RL library [27] with Proximal Policy Optimization (PPO) for training [28]. The UUV observes a 9-dimensional state vector $\vec{o}_t = \{\vec{q}, \vec{q}_{\text{des}}, \Delta z\}$, where Δz represents the depth error, while $\vec{q}$ and $\vec{q}_{\text{des}}$ denote the current and desired attitude quaternions, which ensure singularity-free orientation tracking. The policy then outputs a 4-dimensional action vector $\vec{a}_t = \{\Delta\phi, \Delta\varphi, \Delta\theta, \Delta d\}$, representing deviations in roll, pitch, yaw, and depth, which are passed to the low-level controller, while a reward function composed of three terms guides the policy toward stable behavior. The terms are listed as follows:

- $r_q = \exp(-|\vec{q}\vec{q}^{\star}_{\text{des}}|)$ encourages orientation alignment,
- $r_p = \exp(-||\vec{a}||^b)$ penalizes excessive control actions (with $b = 1$),
- $r_z = \exp(-||\Delta z||^2)$ promotes accurate depth tracking.

These terms are then linearly weighted to guide the policy toward stable and efficient behavior.

At the low level, we employ an **A-S-Surface controller** [29] to ensure a fast and robust response under underwater disturbances. Based on the system state $\mathbf{x}(t) = [\delta(t), \dot{\delta}(t)]^\top$, the angle error and its derivative are defined as $e(t) = \delta_{\text{des}} - \delta(t)$ and $\dot{e}(t) = -\dot{\delta}(t)$. The control output is computed as:

$$u_t = \frac{2}{1 + \exp(-\zeta_1 e(t) - \zeta_2 \dot{e}(t))} - 1 + \Delta u(t), \quad (3)$$

with the adaptive compensation term updated by:

$$\Delta u(t+1) = \Delta u(t) + \alpha e(t) \text{sign}(u_t), \quad (4)$$

where α is a tunable learning rate. This formulation offers both high gain for large deviations and smooth convergence near the setpoint.

To enable zero-shot transfer in Sim2Real deployment and reduce manual tuning under runtime variations, we incorporate a **multimodal LLM** that adaptively adjusts controller parameters (e.g., ζ_1, ζ_2) without the need for retraining. Specifically, the LLM processes two types of input:

- Visual logs, providing a compact representation of control trends and tuning history, conveying complex information concisely while avoiding redundant decisions;
- Textual data, including sensor readings and user instructions that offer fine-grained, non-visual information.

These inputs are processed via a lightweight API, with the LLM prompted by context-rich histories and restricted to output adjustment values. Rather than generating new parameters directly, these outputs indicate the direction and scaling factor for modifications. To ensure stability and precision, several fuzzy rules are predefined for scaling factors (e.g., 2× / 0.5× for major changes, 1.5× / 0.67× for finer refinements). This approach enhances numerical stability when handling quantitative inputs [30] and enables timely tuning decisions, thereby improving control robustness in dynamic and various underwater environments.

C. Hardware Platform

Our EasyUUV hardware platform (Fig. 3) is a compact, low-cost, and modular testbed built to support the LLM-

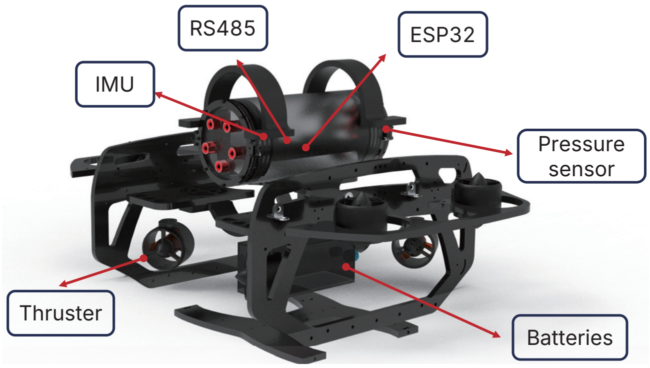


Fig. 3: Exploded view of our EasyUUV hardware platform.

enhanced RL framework and enable Sim2Real zero-shot transfer. The hull combines 3D-printed ABS with aluminum, balancing durability and ease of manufacturing. Costing about $1000 USD, it is far more affordable than conventional UUVs, while its modular design supports diverse payloads. The propulsion system uses eight custom-built thrusters with thrust characteristics similar to Blue Robotics T200, arranged in a fully actuated 6-DOF configuration with vibration isolation to reduce sensor noise.

An ESP32-WROOM microcontroller executes the A-S-Surface controller at 100 Hz using gains pre-calibrated in simulation testing. A low-latency RS-485 tether relays real-time commands from a surface laptop and runs an expert-level RL policy. A 9-DOF MPU9250 IMU with complementary filtering handles sensor fusion. The entire platform fits within a 30 L waterproof case, weighs under 20 kg, and is operable by one person—making it portable and cost-effective for research-grade attitude control. By mirroring simulation dynamics and using LLM for online fine-tuning, EasyUUV enables robust Sim2Real zero-shot transfer.

III. EXPERIMENTS

In this section, we describe the experimental setup used for both simulation and real-world testing. As shown in Fig. 4, the EasyUUV testbed consists of UUV hardware connected to a host computer, which performs RL training, policy deployment, and real-time sensor data collection. To mimic realistic underwater dynamics, we also apply two dedicated perturbation generators in a confined indoor tank.

A. Simulation Setup

The simulation training was conducted on a computer equipped with a Ryzen 9 7945HX CPU and an RTX 4060 GPU. A total of 460 episodes ($\sim 3 \times 10^7$ steps) were completed in approximately 130 seconds, demonstrating high computational efficiency and rapid policy iteration.

Toward the end of training, we introduce two evaluation tasks to assess the Mean Square Error (MSE) performance of different control strategies. **Task 1** involves tracking a smooth sinusoidal signal, while **Task 2** requires following a more complex trajectory constructed by summing multiple sine waves with distinct frequencies:

$$s(t) = A \cdot \sum_{f \in \mathcal{F}} \sin(2\pi f t), \tag{5}$$

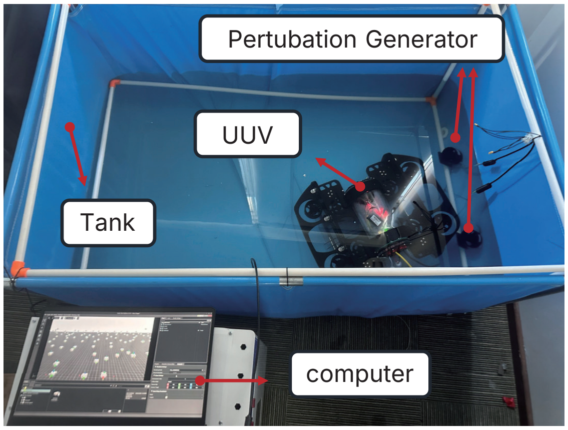


Fig. 4: Experimental testbed for real-world validation of EasyUUV.

where A is the amplitude (in radians), $\mathcal{F}$ is the set of frequencies (in Hz), and t denotes time (in seconds). The specific parameters for each attitude angle are:

- **Yaw:** A = 1.35, $\mathcal{F}$ = {-0.1, 0.2, 0.4, 0.8, 1.6, -3.2},
- **Pitch:** A = 1.10, $\mathcal{F}$ = {-0.1, 0.2, 0.5, -1.0, 2.0, 3.5},
- **Roll:** A = 0.95, $\mathcal{F}$ = {0.15, 0.3, 0.5, -0.9, 1.8, -3.0}.

To further evaluate tracking accuracy, we define the **compound error** at time t as the sum of absolute differences between actual and desired yaw, pitch, and roll angles extracted from the corresponding quaternions:

$$\text{CompoundError}_t = \sum_{i \in \{\phi, \varphi, \theta\}} |i_t - i_{\text{des},t}|. \tag{6}$$

B. Simulation Results

We first conduct the simulation RL training, as shown in Fig. 5. The curves compare three controllers—RL with A-S-Surface, S-Surface, and PID—in terms of cumulative reward and MSE. Here the controller parameters are primarily adopted from [11] to ensure a fair baseline for comparison. In Fig. 5(Left), A-S-Surface converges the fastest and achieves the highest final reward, indicating superior learning efficiency. S-Surface shows slower convergence and lower reward, while PID performs worst with minimal improvement. In Fig. 5(Right), A-S-Surface also maintains the lowest MSE throughout training, followed by S-Surface with moderate error, and PID with consistently the highest MSE. These results highlight clear advantages of the A-S-Surface controller in adaptive control, particularly in improving learning and tracking performance.

Building on these observations, Fig. 6 provides a complementary comparison using bar charts of MSE values for the same three controllers across two representative tasks under RL-enabled (w/ RL) and non-RL (w/o RL) settings. In the RL case (Fig. 6(Left)), A-S-Surface consistently achieves the lowest MSE, demonstrating superior tracking accuracy and robustness, while S-Surface shows moderate performance and PID the highest MSE. In the non-RL case (Fig. 6(Right)), all controllers experience a performance drop with higher MSE, yet A-S-Surface still performs best, indicating its adaptive structure provides baseline robustness. Overall, these results underscore the combined benefits of RL and adaptive control: RL effectively reduces tracking error,

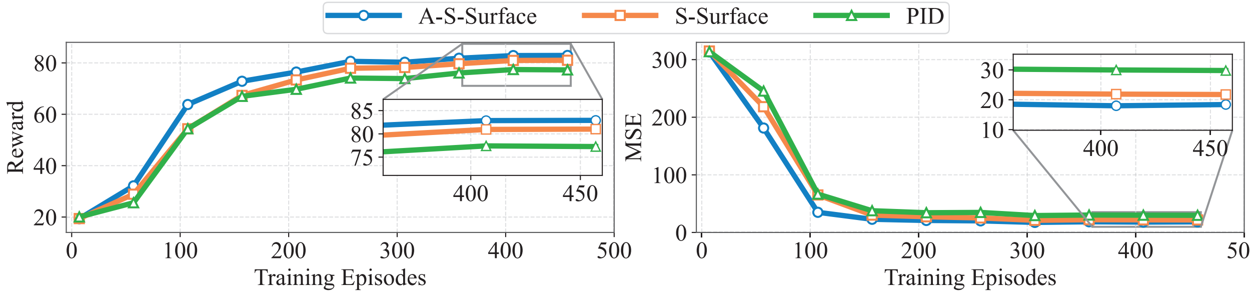


Fig. 5: Training curves of RL with different controllers in terms of reward and MSE. (Left) Reward curves. (Right) MSE curves.

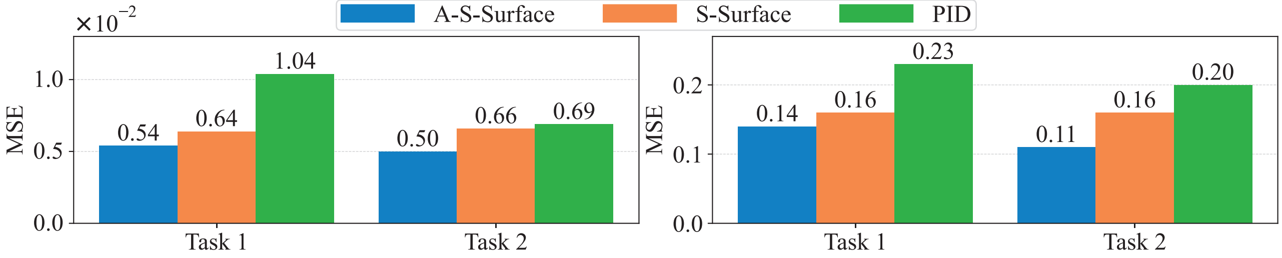


Fig. 6: Comparison of MSE across two tasks for different controllers, under both RL and non-RL settings. (Left) w/ RL. (Right) w/o RL.

TABLE II: The MSE Results under Varying Domain Randomization Levels in Task1 and Task2.

Settings		**NDR**	**SDR**	**LDR**
Task 1	In domain	0.0054	**0.0051**	0.0061
	Pos. buoy	0.0344	**0.0087**	0.0110
	Neg. buoy	0.0339	**0.0091**	0.0092
Task 2	In domain	**0.0050**	0.0057	0.0061
	Pos. buoy	0.0388	**0.0066**	0.0160
	Neg. buoy	0.0320	**0.0079**	0.0132

and A-S-Surface remains the most reliable across different conditions.

Building on this foundation, we next examine the effect of DR for RL training. Specifically, we present MSE results under different DR levels—None (NDR), Small-scale (SDR), and Large-scale (LDR)—to analyze the impact of physical variability on policy generalization. To evaluate out-of-domain performance, we fixed the UUV mass while varying its volume to create two conditions: density at 0.95× and 1.05× the default value ($\approx$ water density), denoted as Pos. buoy and Neg. buoy, respectively. The policies were trained with RL using the A-S-Surface controller. As shown in TABLE II, policies without DR suffer significant MSE degradation under buoyancy shifts, whereas SDR and LDR reduce performance loss, with SDR achieving better generalization and LDR showing less stability. Thus, these findings suggest that exposing policies to broader physical uncertainties during training improves robustness and cross-domain performance.

Having validated robustness against environmental variability, we then turn to attitude tracking performance. To further evaluate attitude tracking in simulation, Fig. 7 compares yaw, pitch, and roll responses under Task 2. In Fig. 7(Left), RL+A-S-Surface achieves the closest tracking with fast convergence and minimal steady-state error, RL+S-Surface shows larger deviations and mild oscillations, while RL+PID responds more slowly with significant errors, especially in pitch and roll. Fig. 7(Right) further shows that A-S-Surface with RL attains higher accuracy and responsiveness than its non-RL counterpart, which exhibits delays and larger errors. Moreover, Fig. 8(Left) illustrates compound error evolution: RL+A-S-Surface maintains the lowest and most stable error, RL+S-Surface shows moderate fluctuations, and RL+PID suffers larger deviations, particularly around 10–12s and near the end. Finally, Fig. 8(Right) confirms that RL reduces both average error (from μ=0.452 to μ=0.103) and variability. Overall, the results highlight the advantage of integrating RL with adaptive control for robust multi-axis attitude regulation.

C. Real-World Deployment

Building on the simulation results, we first conduct the tank experiment under disturbance-free conditions to evaluate EasyUUV's Sim2Real zero-shot transfer capability using the expert-level RL policy directly taken from simulation with the A-S-Surface controller. As shown in Fig. 9, the RL-enabled controller tracks desired commands more closely, keeping roll and pitch near the origin with reduced drift and phase lag, while the non-RL case shows larger deviations. In addition, Fig. 10 further compares compound error curves, where the RL-enabled controller achieves lower average error (μ=0.2356vs. 0.3836, and μ=0.2421 vs. 0.2876) and reduced variability (σ=0.080 vs. 0.150, and σ=0.0896 vs. 0.0970), indicating improved robustness. Taken together, these findings preliminarily confirm EasyUUV can achieve effective zero-shot transfer from simulation to real-world deployment, significantly enhancing multi-axis tracking performance.

After the initial disturbance-free tests, we further activate the perturbation generators to validate the effectiveness of LLM's online fine-tuning capacity. As shown in Fig. 11, EasyUUV rapidly suppresses disturbances and restores the

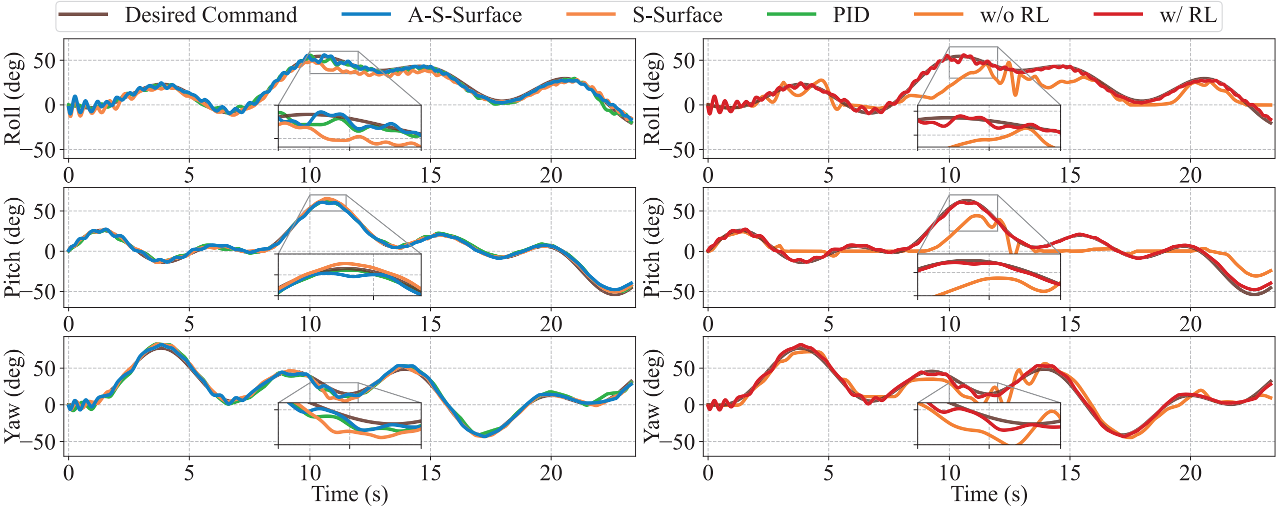


Fig. 7: Comparison of UUV attitude tracking response curves for different control strategies in simulation experiments. (Left) RL with different controllers. (Right) w/o RL and w/ RL settings.

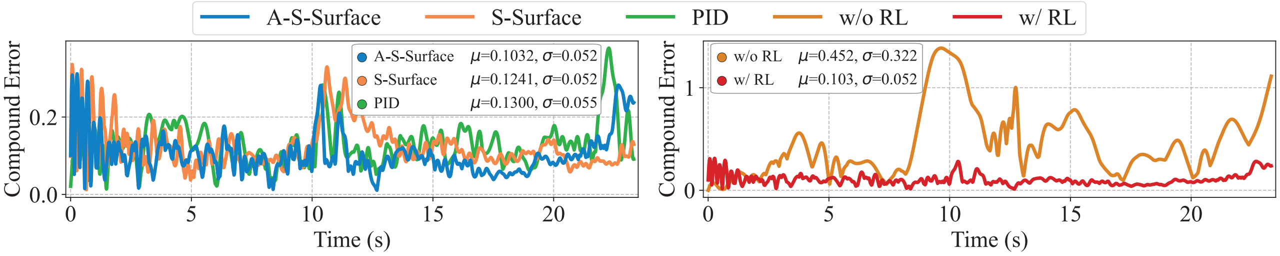


Fig. 8: Comparison of UUV attitude compound error curves for different control strategies in simulation experiments. (Left) RL with different controllers. (Right) w/o RL and w/ RL settings.

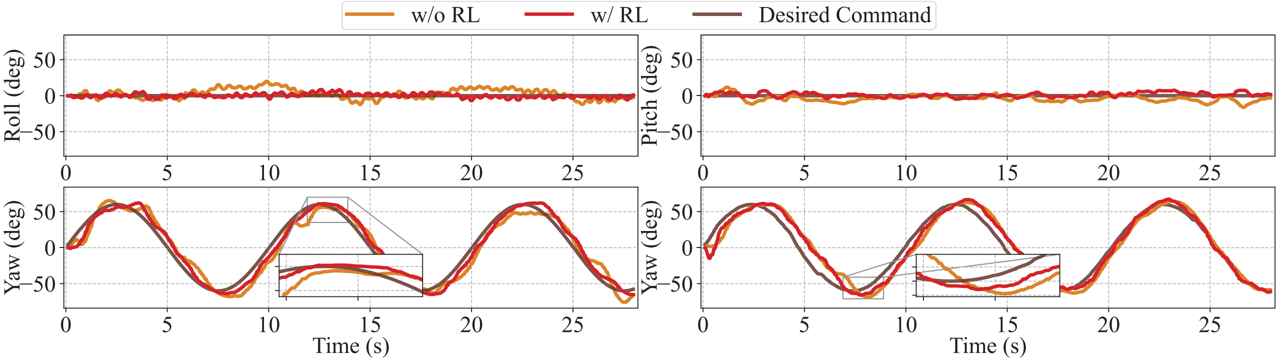


Fig. 9: Comparison of UUV attitude tracking response curves for different attitude angles combination, under both RL and non-RL settings in real-world experiments. (Left) Yaw and Roll. (Right) Yaw and Pitch.

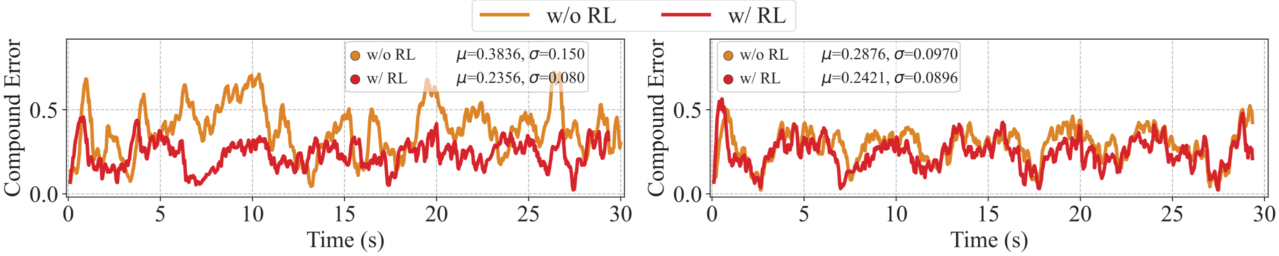


Fig. 10: Comparison of UUV attitude compound error curves for different attitude angles combination, under both RL and non-RL setting in real-world experiments. (Left) Yaw and Roll. (Right) Yaw and Pitch.

vehicle to the desired trajectory, demonstrating strong robustness in real-world conditions. Fig. 12 further evaluates yaw tracking under turbulence, where LLM-based online parameter tuning progressively reduces the mean squared error from $0.0812\ \mathrm{rad}^2$ to $0.0179\ \mathrm{rad}^2$ after two adjustments, significantly enhancing tracking accuracy and stability. These

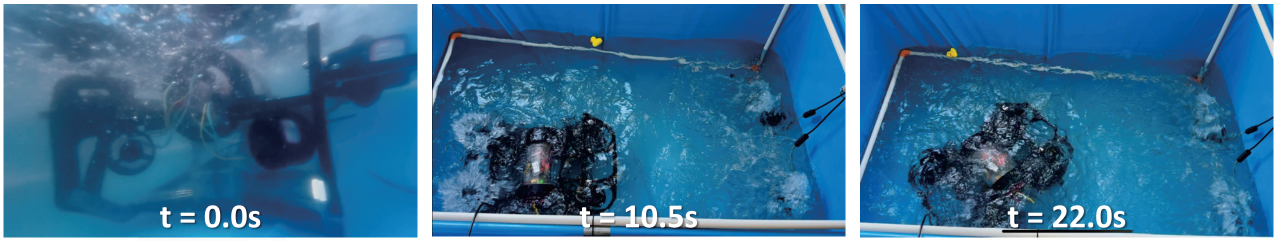


Fig. 11: Snapshots of EasyUUV operating in a indoor tank at $t = 0.0$s, 12.5s, and 25.0s in real-world experiments.

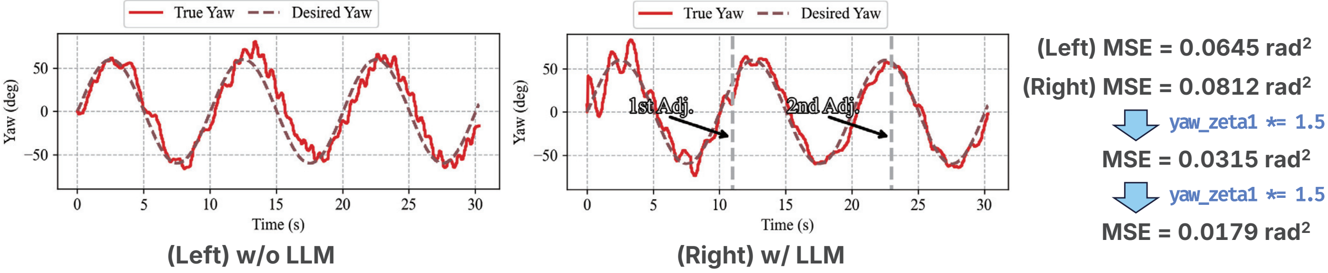


Fig. 12: Tracking response curves under turbulent perturbations with and without LLM-based online fine-tuning of controller parameters.

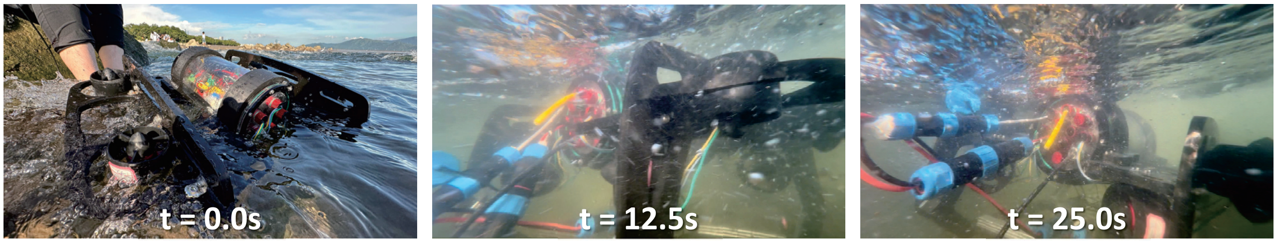


Fig. 14: Snapshots of EasyUUV operating in real ocean conditions at $t = 0.0$s, 12.5s, and 25.0s in sea trials.

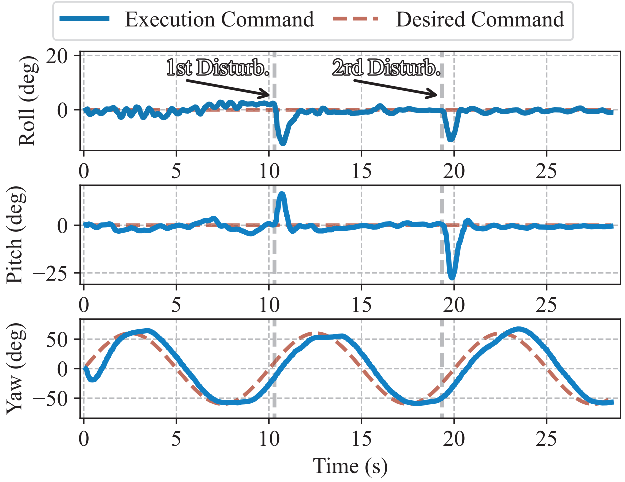


Fig. 13: Tracking response curves along the roll, pitch, and yaw axes in tank experiments under turbulent and transient perturbation.

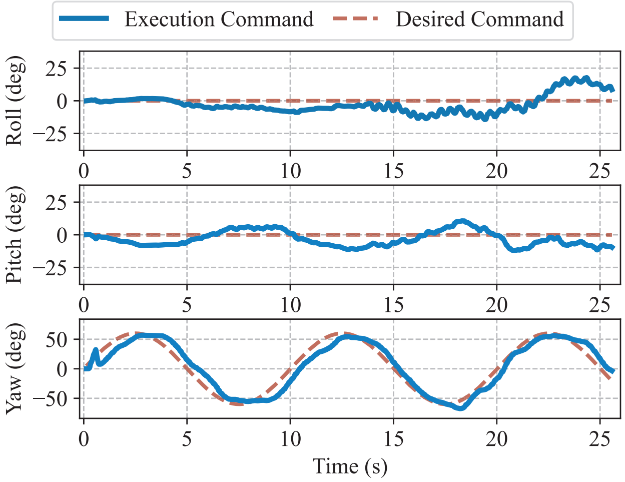


Fig. 15: Tracking response curves along the roll, pitch, and yaw axes in sea trials under wave-induced turbulence.

results confirm that EasyUUV not only withstands perturbations but also leverages LLM-based adjustments for adaptive, high-precision control in dynamic underwater environments.

To more rigorously assess robustness, two strong transient perturbations are manually introduced at 10.3s and 19.4s. As illustrated in Fig. 13, under turbulent disturbances, the EasyUUV still tracks the desired trajectory closely on all three axes with steady-state errors near zero. Although roll and pitch briefly deviate when the manual perturbations occur, the framework quickly suppresses the errors and restores the trajectory, while yaw tracking remains accurate throughout. These results verify the robustness and stability of the proposed framework, enabling EasyUUV to maintain high-precision control under turbulent and sudden disturbances while demonstrating strong Sim2Real transfer capability.

Based on the above tests, we finally extend the evaluation to sea trials. Figs. 14 and 15 present the results, complementing the earlier tank experiments and highlighting the framework's zero-shot domain transfer capability. The tracking curves indicate that EasyUUV closely follows desired commands in roll, pitch, and yaw under wave-induced turbulence, with steady-state errors near zero. In addition, snapshots at $t = 0.0$s, 12.5s, and 25.0s clearly illustrate operation in real ocean conditions with waves and strong flows. Together with the tank experiments, these results confirm that the framework transfers directly from controlled environments to open-sea settings without retraining, achieving robust disturbance rejection and stable high-precision control.

IV. CONCLUSIONS

In this paper, we introduce EasyUUV, an LLM-enhanced universal and lightweight Sim2Real RL framework for robust UUV attitude control. The framework integrates domain-randomized RL training with a hybrid control architecture that incorporates an A-S-Surface controller, while a multi-modal LLM agent provides runtime parameter fine-tuning without additional retraining. Built on a cost-effective 6-DoF UUV platform, EasyUUV enables efficient simulation-based policy learning and achieves zero-shot transfer to real-world deployment. Extensive simulation and field experiments demonstrate that EasyUUV offers stable, generalizable control, with superior robustness and consistent Sim2Real performance under diverse underwater conditions.

In the future, we plan to extend EasyUUV with visual–language models and image enhancement methods to enable higher-level goal interpretation and resilient navigation in unstructured, visually degraded environments.

REFERENCES

[1] L. Hawkes, O. Exeter, S. Henderson, C. Kerry, A. Kukulya, J. Rudd, S. Whelan, N. Yoder, and M. Witt, "Autonomous underwater videography and tracking of basking sharks," *Animal Biotelemetry*, vol. 8, August 2020.

[2] J. Rutledge, W. Yuan, J. Wu, S. Freed, A. Lewis, Z. Wood, T. Gambin, and C. Clark, "Intelligent shipwreck search using autonomous underwater vehicles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6175–6182.

[3] F. L. Peña, F. Orjales, and A. Deibe, "Development of a collaborative host-guest unmanned underwater vehicle docking system for inspection and maintenance of offshore structures," in *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, 2023, pp. 1–5.

[4] X. Lin, N. Karapetyan, K. Joshi, T. Liu, N. Chopra, M. Yu, P. Tokekar, and Y. Aloimonos, "Uivnav: Underwater information-driven vision-based navigation via imitation learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5250–5256.

[5] J. WEI, "Auv optical vision submarine pipeline inspection," in *2024 7th International Symposium on Autonomous Systems (ISAS)*, 2024, pp. 1–5.

[6] I. Salman, N. Karapetyan, A. Venkatachari, A. Q. Li, A. Bourbonnais, and I. Rekleitis, "Multi-modal lake sampling for detecting harmful algal blooms," in *OCEANS 2022, Hampton Roads*, 2022, pp. 1–9.

[7] A. Mitchell, E. McGookin, and D. Murray-Smith, "Comparison of control methods for autonomous underwater vehicles," *IFAC Proceedings Volumes*, vol. 36, no. 4, pp. 37–42, 2003, iFAC Workshop on Guidance and Control of Underwater Vehicles 2003, Newport, South Wales, UK, 9-11 April 2003.

[8] Z. Yan, P. Gong, W. Zhang, and W. Wu, "Model predictive control of autonomous underwater vehicles for trajectory tracking with external disturbances," *Ocean Engineering*, vol. 217, p. 107884, 2020.

[9] B. Hu, H. Tian, J. Qian, G. Xie, L. Mo, and S. Zhang, "A fuzzy-pid method to improve the depth control of auv," in *2013 IEEE International Conference on Mechatronics and Automation*, 2013, pp. 1528–1533.

[10] C.-W. Chen, N.-M. Yan, J.-X. Leng, and Y. Chen, "Numerical analysis of second-order wave forces acting on an autonomous underwater helicopter using panel method," in *OCEANS 2017 - Anchorage*, 2017, pp. 1–6.

[11] G. Xie, J. Xu, Y. Ding, Z. Zhang, S. Zhang, and Y. Li, "Never too prim to swim: An llm-enhanced rl-based adaptive s-surface controller for auvs under extreme sea conditions," 2025.

[12] W. Wei, J. Wang, J. Du, Z. Fang, Y. Ren, and C. L. P. Chen, "Differential game-based deep reinforcement learning in underwater target hunting task," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 462–474, 2025.

[13] I. Masmitja, M. Martin, T. O'Reilly, B. Kieft, N. Palomeras, J. Navarro, and K. Katija, "Dynamic robotic tracking of underwater targets using reinforcement learning," *Science Robotics*, vol. 8, no. 80, p. eade7811, 2023.

[14] L. Cai, K. Chang, and Y. Girdhar, "Learning to swim: Reinforcement learning for 6-dof control of thruster-driven autonomous underwater vehicles," 2025.

[15] D. Meger, J. C. G. Higuera, A. Xu, P. Giguère, and G. Dudek, "Learning legged swimming gaits from experience," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2332–2338.

[16] B. Hadi, A. Khosravi, and P. Sarhadi, "Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle," *Applied Ocean Research*, vol. 129, p. 103326, 2022.

[17] D. Xue, Z. Gengshi, X. Jian, and C. Tao, "Position and attitude control of uuv in the process of operation tasks," in *2018 37th Chinese Control Conference (CCC)*, 2018, pp. 2893–2898.

[18] V. Sufán and G. Troni, "Swim4real: Deep reinforcement learning-based energy-efficient and agile 6-dof control for underwater vehicles," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 7326–7333, 2025.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[20] R. Zahedifar, M. Soleymani Baghshah, and A. Taheri, "Llm-controller: Dynamic robot control adaptation using large language models," *Robotics and Autonomous Systems*, vol. 186, p. 104913, 2025.

[21] J. Xu, Z. Zheng, and Z. Wang, "Lac: Using llm-based agents as the controller to realize embodied robot," in *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2024, pp. 1894–1899.

[22] Q. Guo, X. Liu, J. Hui, Z. Liu, and P. Huang, "Utilizing large language models for robot skill reward shaping in reinforcement learning," in *Intelligent Robotics and Applications*, Singapore, 2025, pp. 3–17.

[23] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[24] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[25] BlueRobotics, "Bluerobotics t200 performace charts," 2019. [Online]. Available: https://cad.bluerobotics.com/T200-Public-Performance-Data-10-20V-September-2019.xlsx

[26] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, "Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 59–66.

[27] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[28] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[29] B. Li, Y. Xu, C. Liu, and W. Xu, "Simulation and preliminary experimental results on s-surface control of an autonomous underwater vehicle based on moos-ivp," in *2014 Oceans - St. John's*, 2014, pp. 1–6.

[30] Y. Yan, Y. Lu, R. Xu, and Z. Lan, "Do phd-level llms truly grasp elementary addition? probing rule learning vs. memorization in large language models," *arXiv preprint arXiv:2504.05262*, 2025.