Journal of *Marine Science and Engineering*

MDPI

Article

Reinforcement-Learning-Based Adaptive PID Depth Control for Underwater Vehicles Against Buoyancy Variations

Jian Wang [1,2,3], Shuxue Yan [1,3], Honghao Bao [1,3], Cong Chen [1,3], Deyong Yu [1,3], Jixu Li [1,2,3], Xi Chen [1,3], Rui Dou [1,3], Yuangui Tang [1,3,*] and Shuo Li [1,3,*]

[1] State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; wangjian3@sia.cn (J.W.); yansx@sia.cn (S.Y.); baohanghao@sia.cn (H.B.); chencong@sia.cn (C.C.); yudeyong@sia.cn (D.Y.); lijixu@sia.cn (J.L.); chenxi3@sia.cn (X.C.); dourui@sia.cn (R.D.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Key Laboratory of Marine Robotics of Liaoning Province, Shenyang 110169, China
* Correspondence: tyg@sia.cn (Y.T.); shuoli@sia.cn (S.L.)

Abstract

Underwater vehicles performing sampling tasks often encounter significant buoyancy variations due to payload adjustments and environmental changes, which severely challenge the stability and accuracy of controllers. To address this issue, this paper proposes a hybrid control framework that integrates Proximal Policy Optimization (PPO) with adaptive PID tuning. The framework employs PPO to dynamically adjust PID parameters online while incorporating output saturation, stepwise quantization, and dead zone filtering to ensure control safety and actuator longevity. A dual-error state representation—combining instantaneous error and its derivative—along with actuator command buffering is introduced to compensate for system lag and inertia. Comparative simulations and experimental tests demonstrate that the proposed method achieves faster convergence, lower steady-state error, and smoother control signals compared to both conventional PID and pure PPO-based control. The framework is validated through pool tests and field trials, confirming its robustness under realistic hydrodynamic disturbances. This work provides a practical and safe solution for adaptive depth control of sampling-capable AUVs operating in dynamic underwater environments.

Keywords: underwater vehicles; PPO-based adaptive PID; Autonomous and Remotely operated Vehicle (ARV)

Academic Editor: Weicheng Cui

Received: 8 January 2026
Revised: 5 February 2026
Accepted: 6 February 2026
Published: 7 February 2026

1. Introduction

Underwater vehicles equipped with sampling capabilities play a pivotal role in marine exploration, environmental monitoring, and resource extraction [1]. Haidou-1 is a hybrid autonomous and remotely operated vehicle (ARV) designed for multipurpose scientific investigations, as shown in Figure 1. It has achieved a maximum depth of 10,908 m in the western pool of the Challenger Deep and probed a depth of 10,909 m in the eastern pool [2]. Featuring a streamlined design and self-powered propulsion, the vehicle is equipped with a 6-degree-of-freedom (DOF) electric manipulator. Haidou-1 emphasizes integrated exploration and intervention capabilities, enabling it to carry both manipulative and mapping payloads simultaneously. As a result, it can perform wide-area

autonomous surveys as well as real-time close-range imaging, sampling, and manipulation tasks.

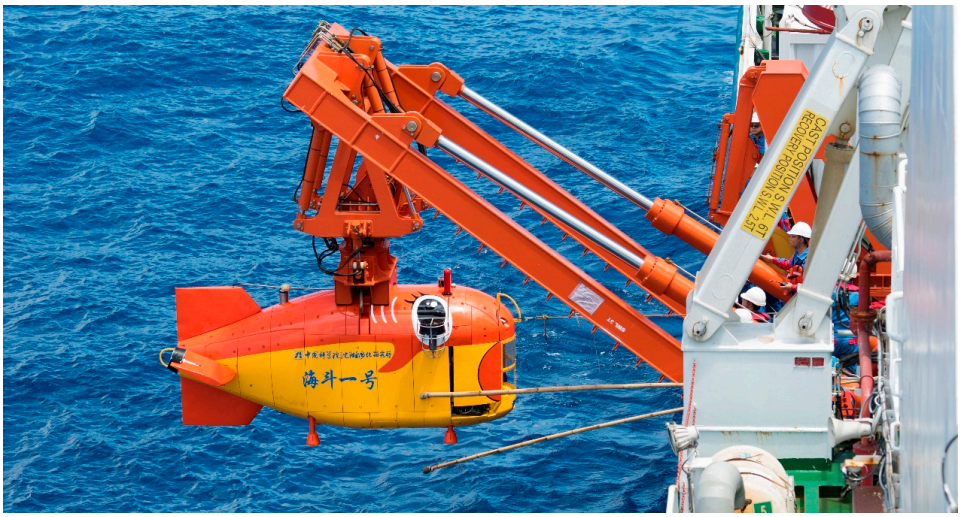

Figure 1. Haidou-1 featuring an integrated design for detection and operation tasks is capable of diving to the deepest known point in the Earth's oceans.

However, operational tasks such as sample collection or the deployment of seafloor instruments—typically carried out using the bow payload zone—can introduce significant buoyancy variations, as illustrated in Figure 2. Furthermore, buoyancy is also substantially influenced by changes in operating depth, primarily due to the increase in seawater density under extreme pressure. Taking Haidou-1 as an example, with a mass of 2640 kg and considering the ambient seawater density at the trench bottom (~1074 kg/m^3), the increased buoyancy amounts to approximately 1368 N. After accounting for a hull compression of 37.8 L at 11,000 m depth, the net increase in buoyancy reaches about 475 N. These combined variations directly affect the vehicle's buoyancy state, making precision and stability in depth control critical to mission success. Traditional control methods, including fixed-gain PID controllers, often struggle to perform satisfactorily under such dynamic conditions due to their limited adaptability to nonlinear system dynamics and unmodeled external disturbances.

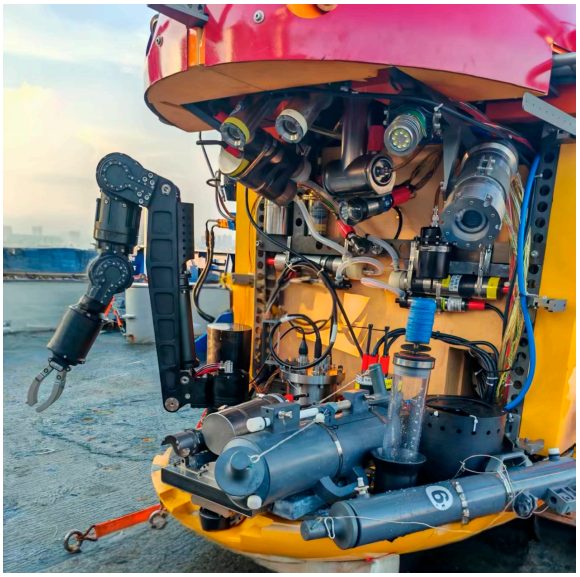

Figure 2. Haidou-1 is equipped with a manipulator for collecting samples (e.g., rocks, organisms, sediments, and water) and conducting tasks such as in situ sensor measurements and seafloor instrument deployment. These operations can induce considerable buoyancy variations.

Recent advances in deep reinforcement learning (DRL) offer promising alternatives for adaptive control. Among DRL algorithms, Proximal Policy Optimization (PPO) has gained widespread popularity for its stability, sample efficiency, and ease of implementation. However, direct application of PPO to underwater robotics faces several critical challenges: slow convergence during training, high computational demand, and—most notably—safety risks associated with random action exploration. For high-cost, high-risk AUV platforms, unsafe control actions can lead to hardware damage or mission failure, rendering pure DRL approaches impractical for real-world deployment.

To bridge this gap, this paper proposes a hybrid PPO-PID control framework that synergizes the adaptive learning capability of PPO with the structural safety and reliability of PID control. The key contributions of this work are threefold:

1. A safe and efficient PPO-based PID tuning mechanism that adapts PID gains online while constraining outputs within actuator operational limits.
2. A layered output processing module integrating saturation constraint, stepwise quantization, and dead zone filtering to enhance control smoothness and hardware safety.
3. A structured state representation incorporating dual-error characterization and actuator command buffering to compensate for system delays and improve learning efficiency.

The proposed framework is validated through comprehensive simulations, controlled pool tests, and real-world lake trials, demonstrating superior performance in convergence speed, tracking accuracy, and robustness under buoyancy variations and environmental disturbances.

The remainder of this paper is organized as follows: Section 2 reviews related work in underwater robot control and DRL applications. Section 3 details the proposed PPO-PID control framework. Section 4 presents experimental results and comparative analysis. Section 5 concludes the paper and suggests future research directions.

2. Related Work

The adaptive control of AUVs has long been a prominent research challenge, driven by the need for robust performance in complex, time-varying hydrodynamic environments. This section reviews the evolution of control strategies, focusing on the transition from classical methods to data-driven and hybrid approaches, and identifies the specific research gap addressed by this work.

2.1. Classical and Adaptive Control Strategies

Conventional control methodologies, such as proportional–integral–derivative (PID) control, sliding mode control (SMC), and model predictive control (MPC), form the foundation of many operational AUV systems [3]. PID controllers are particularly prevalent in industrial applications due to their structural simplicity, interpretability, and ease of implementation [4]. A scheme that optimizes PID control to achieve stable near-bottom fixed-height navigation using vertical thrusters was proposed [5]. However, their effectiveness heavily relies on precise parameter tuning and lacks adaptability to significant changes in plant dynamics, such as buoyancy variations or unmodeled external disturbances [6].

To address these limitations, classical adaptive control techniques like gain-scheduling and model-reference adaptive control have been explored [7,8]. A comparative study on robust adaptive control for underwater vehicles was conducted, while novel adaptive control laws for six-degrees-of-freedom (6-DOF) AUV control were proposed [9]. Despite

these advancements, most deployed AUVs still utilize fixed-gain PID controllers, as automatic tuning often requires high-fidelity models or persistent excitation conditions rarely met in practice.

2.2. Deep Reinforcement Learning for Underwater Robotics

Machine learning (ML) techniques have increasingly been applied in naval architecture and marine engineering, leveraging data from experiments and high-fidelity simulations to extract underlying physics and improve control strategies [10]. Reinforcement Learning (RL), especially Deep RL (DRL), has emerged as a promising model-free alternative for learning control policies through environmental interaction [11]. Early applications in underwater robotics included discretized state or action spaces, such as Q-learning for vision-based control [12] and neural networks in behavior-based architectures [13]. DRL has been successfully applied to underwater target tracking by autonomous surface vehicles, demonstrating robust path planning and adaptability in real-world ocean conditions [14]. The development of algorithms like deep deterministic policy gradient (DDPG) [15] and soft actor-critic (SAC) [16] enabled direct learning of continuous control actions, expanding DRL's applicability to low-level AUV control [17,18]. A DDPG-based adaptive low-level controller for AUVs was developed, validated through real experiments on the Nessie VII AUV [19]. However, pure DRL faces challenges including high sample complexity, safety concerns during exploration, and real-time stability issues [20]. For unmanned surface vehicles (USVs) which share similar marine environmental challenges with AUVs, a mapless navigation method based on DRL was proposed to address sim-to-real transfer and slow convergence issues [21]. Soft underwater robots also benefit from DRL frameworks that enable stable locomotion in disturbed water by learning from simulation and transferring policies to real robots [22]. Beyond locomotion and control, DRL combined with visual perception and attention mechanisms improves collaborative manipulation tasks such as pushing and grasping in underwater environments, enhancing success rates under realistic conditions [23]. These achieve strong performance but suffer from interpretability and safety concerns.

2.3. Hybrid Learning-Based Adaptive Control

To combine the stability of classical control with the adaptability of learning, hybrid approaches have gained significant traction [24]. These methods embed learning algorithms (e.g., neural networks, RL agents) within classical control structures for online parameter adjustment [25]. Hybrid learning-based adaptive control for AUVs often uses PID as a stable, interpretable backbone and lets DRL agents (DDPG, SAC, PPO) adapt the gains online. This combines guaranteed baseline performance with data-driven adaptation to strong disturbances, model uncertainties, and changing tasks. DRL schemes use actor–critic agents to tune several PID loops simultaneously, processing only low-level dynamic signals while compensating for environmental uncertainties in both simulation and real experiments [26]. Recent work couples PID with Soft Actor–Critic (SAC) to obtain adaptive, learning-based controllers that keep PID's structure while improving robustness and performance [27]. Similar SAC-PID designs for USV trajectory control include Lyapunov-style stability analysis and show faster convergence, stronger robustness, and better generalization than manual tuning, genetic algorithms, and DDPG-based tuning [28]. PID-DRL hybrids for AUV tracking under ocean currents or disturbances use DRL modules to provide optimal thrust or steering commands [29]. A meta-learning and self-adaptation hybrid approach uses deep neural networks to model hydrodynamics offline and adapts parameters online, significantly improving trajectory tracking under varying ocean currents [30]. Radial basis function neural networks combined with adaptive compensators provide model-free approximation of unknown dynamics, ensuring asymptotic error

convergence despite uncertainties [31]. However, these studies often lack comprehensive experimental validation under realistic disturbances (e.g., buoyancy variation) and overlook practical aspects like real-time computational feasibility.

In summary, while the hybrid RL-classical control represents a promising direction for adaptive AUV control, a significant gap persists in the literature. Specifically, few studies offer rigorous experimental validation under realistic and challenging conditions such as buoyancy variations, and even fewer are explicitly designed with integrated mechanisms for training safety and guaranteed real-time stability. This work directly addresses these shortcomings by introducing a novel PPO-based adaptive PID controller for AUV depth control. The proposed method strategically integrates the PPO algorithm within a PID framework, augmented with safety-aware output processing, to deliver a balanced solution that combines adaptability with operational safety. Our contributions are substantiated through comprehensive co-simulation and in-lake trials, confirming the controller's efficacy and robustness.

3. Proposed PPO-Based Adaptive Control Framework

The proposed framework consists of four core components: PPO policy optimization for PID parameter tuning, layered output processing (saturation, stepwise quantization, deadzone filtering), structured state representation (dual-error characterization and actuator command buffering), and a multi-objective reward function design. The overall architecture of the framework is illustrated in Figure 3.

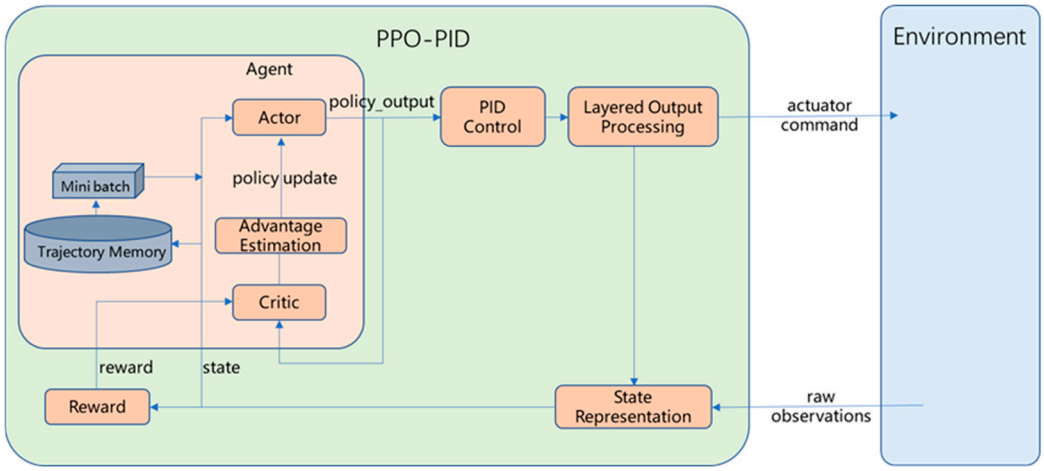

Figure 3. Schematic diagram of the proposed PPO-PID adaptive control framework.

3.1. PPO Algorithm for PID Tuning

This section details the core design of the PPO-PID hybrid tuning mechanism. The framework leverages the PID controller's structural reliability as the baseline control backbone, while integrating the PPO algorithm's adaptive learning capability to dynamically adjust PID parameters online. The output of the PPO-based PID controller is formulated as:

$$u_{PPO\text{-}PID}(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \quad (1)$$

where K_p, K_i, K_d are the proportional, integral, and derivative gains, respectively. Instead of manually tuning these gains, the PPO algorithm learns to adjust them adaptively based on the system's state.

For each gain dimension, the original output of the PPO actor network is constrained to the range $a_p, a_i, a_d \in [-1, 1]$. The actual PID gains are obtained by linear scaling with the engineering empirical gains of the ARV for depth control as the reference, and the specific mapping formulas are given by:

$$K_p = K_{pexp} \times 20 \times (a_p + 1)/2$$
$$K_i = K_{iexp} \times 20 \times (a_i + 1)/2 \quad (2)$$
$$K_d = K_{dexp} \times 20 \times (a_d + 1)/2$$

where K_{pexp}, K_{iexp}, K_{dexp} are the engineering empirical PID gains for the ARV depth control. Derived from the above scaling rules, the PID gains possess a sufficiently wide and engineering-reasonable adjustment interval, ranging from 0 to 20 times the empirical gains.

The PPO algorithm optimizes a stochastic policy $\pi_\theta(a|s)$, where θ is the policy network's parameters, $a = [a_p, a_i, a_d]$ is the action, and s is the state vector. The objective function for PPO is:

$$L(\theta) = \mathbb{E}_{\hat{\pi}}\left[\min\left(\frac{\pi_\theta(a|s)}{\hat{\pi}(a|s)}A(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\hat{\pi}(a|s)}, 1-\epsilon, 1+\epsilon\right)A(s,a)\right)\right] \quad (3)$$

where $\hat{\pi}$ is the old policy, $A(s,a)$ is the advantage function, and $\epsilon = 0.2$ (a typical value for PPO) ensures policy updates are constrained within a trust region, balancing exploration and exploitation.

3.2. Layered Output Processing

To ensure actuator safety, prolong hardware lifespan, and maintain control stability, a layered output processing module is designed to refine the raw control commands from the PPO-PID controller.

3.2.1. Smooth Saturation Constraint

To enforce actuator operational boundaries and prevent commands from exceeding hardware capabilities, a smooth saturation mechanism is implemented via the function $\Psi_{\max}$:

$$\Psi_{\max}(a_{\text{raw}}) = T_{\max} \cdot \tanh\left(\frac{a_{\text{raw}}}{T_{\max}}\right) \quad (4)$$

where a_{raw} denotes the raw control command output by the policy network, $T_{\max}$ is the manufacturer-specified maximum thrust magnitude of the actuator, and $\Psi_{\max}(a_{\text{raw}}) \in (-T_{\max}, T_{\max})$ represents the executable control action after smooth saturation processing.

Unlike a hard clamp, the tanh function provides a continuous and differentiable mapping, avoiding abrupt discontinuities that could confuse the policy gradient during training and induce instability in real-time execution. This formulation confines commands within physical hardware limits while avoiding abrupt discontinuities, ensuring control continuity and compliance with actuator constraints.

3.2.2. Stepwise Quantization

To mitigate actuator chattering and mechanical wear, a stepwise control strategy is applied. This method quantizes continuous action commands into discrete increments, ensuring adjustments occur only when the commanded change exceeds a minimum threshold $\delta_{\min}$. The quantized action update is computed via the function Ψ_{rate}:

$$\Psi_{\text{stepwise}}(a_i^{\text{cmd}}, a_i^{\text{prev}}) = a_i^{\text{prev}} + \Delta a_i^{\text{step}} \quad (5)$$

where

$$\Delta a_i^{\text{step}} = \begin{cases} 0 & \text{if } |a_i^{\text{cmd}} - a_i^{\text{prev}}| < \delta_{\min}, \\ \text{sign}(\Delta a_i^{\text{raw}}) \cdot \delta_{\min} \cdot \left\lfloor \frac{|\Delta a_i^{\text{raw}}|}{\delta_{\min}} \right\rfloor & \text{otherwise,} \end{cases} \quad (6)$$

with $\Delta a_i^{\text{raw}} = a_i^{\text{cmd}} - a_i^{\text{prev}}$.

The stepwise control strategy enforces a minimum dwell time between consecutive actuator adjustments. It significantly reduces mechanical wear by limiting high-frequency operational stress on critical components such as bearings and transmission systems. By bounding stepwise fluctuations, the method ensures control stability during transient disturbances, suppressing erratic oscillations and maintaining persistent command signals even under dynamic operating conditions. Furthermore, the quantization of continuous commands into discrete increments reduces cumulative action changes, directly lowering energy consumption and operational costs. This holistic approach harmonizes hardware longevity, dynamic stability, and energy efficiency, making it indispensable for applications demanding both precision and sustainability.

3.2.3. Dead Zone Filtering

Thrusters exhibit deadband characteristics during low-speed operation due to actuator drives. To address this limitation, a threshold-based filtering mechanism is implemented through the function Ψ_{dead}, which suppresses subthreshold control signals:

$$\Psi_{\text{deadzone}}(a_i^{\text{cmd}}) = \begin{cases} 0 & \text{if } |a_i^{\text{cmd}}| < a_{\text{dead}}, \\ a_i^{\text{cmd}} & \text{otherwise,} \end{cases} \tag{7}$$

where a_{dead} denotes the deadzone threshold. This mechanism systematically rejects ineffective low-magnitude commands ($|a_i^{\text{cmd}}| < a_{\text{dead}}$) to eliminate energy dissipation caused by futile actuator micro-adjustments. Additionally, it attenuates residual oscillations in low-speed regimes, ensuring energy-efficient operation.

The physical constraint layer integrates all functions hierarchically via Ψ_{physics}:

$$\Psi_{\text{physics}}(a) = \Psi_{\text{dead}}(\Psi_{\text{rate}}(\Psi_{\text{max}}(a))) \tag{8}$$

This cascaded structure ensures actuator limits are enforced while maintaining smooth and stable control signals.

3.3. Structured State Representation

Effective state representation is critical for RL performance, as it directly influences the agent's ability to learn meaningful control policies. In this work, we introduce an Augmented Error and Lag-Compensated (AELC) state representation. The proposed state vector $s(t)$ integrates dual-error characterization and actuator command buffering to capture system dynamics and compensate for lag effects.

3.3.1. Dual-Error Characterization

To explicitly characterize control inaccuracies and system dynamic trends, the error $e(t)$ and its temporal derivative $\Delta e(t)$ are constructed:

$$\begin{cases} e(t) = depth_{\text{ref}}(t) - depth_{\text{actual}}(t) \\ \Delta e(t) = \dfrac{e(t) - e(t - \Delta t)}{\Delta t} \end{cases} \tag{9}$$

The augmented error state:

$$e_{state}(t) = \begin{bmatrix} e(t) \\ \Delta e(t) \end{bmatrix} \tag{10}$$

This dual-error formulation explicitly quantifies both the magnitude and temporal trend of control deviations, enhancing sensitivity to steady-state errors and transient dynamics. By integrating these features, the framework enables adaptive compensation for nonlinear control regimes and transient disturbances, thereby improving tracking accuracy and stability in dynamic environments.

3.3.2. Actuator Command Buffering for Lag Compensation

To address the combined effects of actuator response delays and inherent vehicle inertia in underwater robots, a temporal buffering model is integrated into the state space. This model explicitly captures historical actuator commands within a fixed time window ($N = 2$) to compensate for lag and inertia:

$$a_{state}(t) = \begin{bmatrix} a(t) \\ a(t-1) \\ a(t-2) \end{bmatrix} \tag{11}$$

where $a(t-k)$ represents the actuator command issued at time $t-k$ ($k = 0,1,2$), and $N = 2$ denotes the temporal window size, retaining the current and past two commands.

The model embeds sequential actuator commands matrix a_{state}into the state representation, enabling the policy network to leverage historical actions for compensating actuator delay and vehicle inertia. This lightweight design eliminates dependency on explicit weighted summation or derivative-based terms, significantly reducing computational complexity for real-time applications. By retaining actuator commands within a finite temporal window ($N = 2$), the framework ensures sufficient temporal context to address transient dynamics while avoiding computational overhead from excessive historical data. The tunable window size (N) allows for adaptation to varying hydrodynamic conditions.

3.4. Multi-Objective Reward Function Design

The reward function is a critical component of the RL framework, as it guides the policy network to learn optimal control strategies that align with the ARV's depth control objectives (tracking accuracy, smoothness, and energy efficiency). Based on the system constraints and practical operational requirements, a weighted multi-term reward function is proposed, integrating error minimization, stability enhancement, overshoot suppression, and acceleration constraint. The design balances conflicting objectives (e.g., fast convergence vs. smooth control) through heuristic weight tuning validated by offline simulations.

The total reward $r(t)$ at time step t is defined as a weighted sum of five components:

$$r(t) = \omega_e \cdot r_e(t) + \omega_s \cdot r_s(t) + \omega_o \cdot r_o(t) + r_{sparse}(t) \tag{12}$$

where $r_e(t)$, $r_s(t)$, $r_o(t)$, $r_{sparse}(t)$ denote the tracking error penalty, stability penalty, overshoot penalty, and sparse positive reward terms, respectively. $\omega_e, \omega_s, \omega_o$ are the weights of each reward term. Given the varying operational scenarios of ARVs, requirements for tracking accuracy, motion stability, and overshoot suppression differ significantly. Corresponding weights are thus customized to prioritize core performance metrics in alignment with specific mission objectives.

The definitions and physical meanings of each reward term are detailed below:

1. Tracking Error Penalty ($r_e(t)$)

This term prioritizes depth tracking accuracy by penalizing deviations between the actual depth $z(t)$ and the reference depth $z_{\text{ref}}(t)$. The instantaneous depth error is defined as:

$$e(t) = |z(t) - z_{\text{ref}}(t)| \tag{13}$$

The error penalty is designed as a negative reward to minimize deviations:

$$r_e(t) = -e(t) \tag{14}$$

2. Stability Penalty ($r_s(t)$)

To suppress abrupt changes in control signals and ensure smooth depth adjustment, a stability penalty is introduced based on the error derivative (rate of error change), which reflects the transient dynamics of the system. The error derivative is calculated as:

$$\dot{e}(t) = \frac{e(t) - e(t - \Delta t)}{\Delta t} \tag{15}$$

The stability penalty is:

$$r_s(t) = -10 \cdot |\dot{e}(t)| \tag{16}$$

The coefficient 10 amplifies the penalty for rapid error fluctuations.

3. Overshoot Penalty ($r_o(t)$)

Overshoot poses risks to sampling equipment. This term detects overshoot by monitoring the sign change in the depth error and applies an additional penalty:

$$r_o(t) = \begin{cases} -5.0 \cdot e(t) & \text{if } e(t) \cdot e(t - \Delta t) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $e(t - \Delta t)$ is the depth error at the previous time step.

4. Sparse Positive Reward ($r_{sparse}(t)$)

To encourage the policy to converge to high-precision tracking, a sparse positive reward is added when the depth error falls below a threshold ($\epsilon = 0.1$ m, the required accuracy for sampling missions):

$$r_{sparse}(t) = \begin{cases} 1.0 & \text{if } e(t) < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

This term reinforces the policy's preference for steady-state accuracy, accelerating convergence to the target error range.

A baseline weight configuration is formulated based on the core requirements of typical ARV survey missions, where motion stability and depth tracking accuracy are prioritized over overshoot suppression. A weight of $\omega_e = 0.8$ is assigned, as tracking accuracy constitutes a pivotal control objective for ARVs; a weight of $\omega_s = 1.0$ is set to emphasize control smoothness, which is critical for enhancing motion stability and guaranteeing consistent sensor performance. A weight $\omega_o = 0.5$ is set for scenarios with severe buoyancy variations. This moderate weight balances overshoot suppression with the primary objectives.

4. Experiments and Results

To validate the proposed PPO-PID control framework, comprehensive simulations, controlled pool tests, and real-world sea trials were conducted. The performance of the proposed method was compared with conventional PID and pure PPO control under varying buoyancy conditions and environmental disturbances.

4.1. Simulation Setup

The simulation is based on the dynamic model of the Haidou-1 ARV. As illustrated in Figure 4, Haidou-1 utilizes six thrusters and two rudder motors for actuation, enabling motion control in both the horizontal and vertical planes. Twin fixed vertical thrusters are located at the front. Two main thrusters are fixed on the rotating elevators at the stern. Two vectored thrusters cooperate differentially to be capable of heading control.

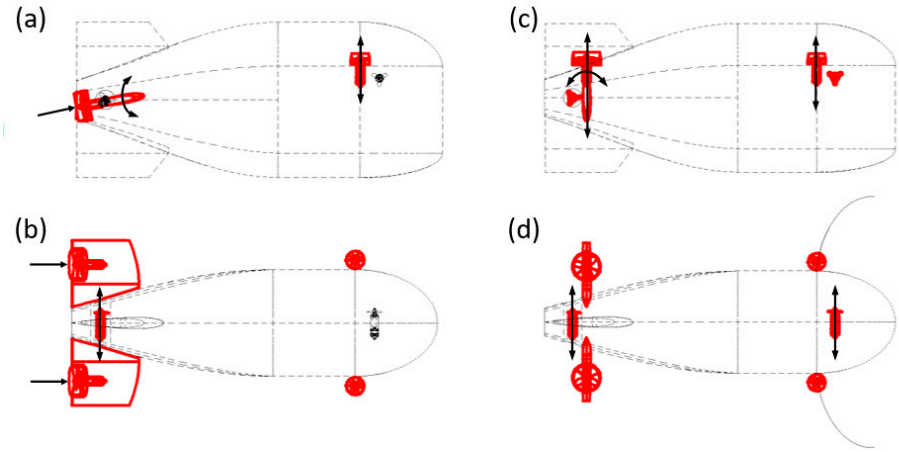

Figure 4. The thrusters and elevators ensure maneuverability and stability in the horizontal and vertical planes: (**a**,**b**) are the thrust allocation diagrams of flight motion control; (**c**,**d**) are the thrust allocation diagrams of vertical motion control.

In this study, we focus on the hover control scenario depicted in Figure 4c. When the stern elevators are rotated to a vertical position, they combine with the fixed bow vertical thrusters to form a total of four vertically aligned thrusters. This configuration grants the vehicle excellent maneuverability and stability in the vertical plane, allowing it to maintain a stable hover even under varying buoyancy conditions.

To accurately simulate the motion characteristics of the underwater vehicle in dynamic marine environments, a six-degree-of-freedom hydrodynamic simulation model is constructed based on Newton–Euler rigid body dynamics theory, viscous hydrodynamic coefficient model, and propeller thrust theory. The core idea is to focus on capturing the dynamic effects of buoyancy–gravity variations and their impact on force-moment balance, integrating buoyancy–gravity resultant force, propeller thrust, viscous hydrodynamic resistance, and ocean current disturbance with Newton–Euler equations as the core to describe the vehicle's translation and rotation motion, and adopting the variable-step fourth-order Runge–Kutta method to solve dynamic differential equations for balancing computational efficiency and solution accuracy.

The core mathematical formulas of the model are as follows: the Newton–Euler dynamic equation serves as the core motion equation, expressed as:

$$M\dot{V} + C(V)V + D(V)V + g(\eta) = \tau_{thrust} + \tau_{disturbance} \quad (19)$$

where M is the combined mass matrix, $\dot{V}$ is the linear and angular acceleration vector, $C(V)$ is the Coriolis–Centrifugal force matrix, $D(V)$ is the hydrodynamic damping matrix, $g(\eta)$ is the buoyancy–gravity combined force/moment vector (a core term reflecting buoyancy variations), τ_{thrust} is the propeller thrust vector, and $\tau_{disturbance}$ is the environmental disturbance vector.

The buoyancy–gravity combined force, a key factor for depth control, is given by:

$$F_x^{gb} = -(Mg - B)\sin\theta$$

$$F_y^{gb} = (Mg - B)\cos\ \theta\sin\ \phi \tag{20}$$

$$F_z^{gb} = (Mg - B)\cos\ \theta\cos\ \phi$$

where M is the vehicle mass, B is the buoyancy (varying with payload, depth, and seawater density), g is the gravitational acceleration, and ϕ (roll) and θ (pitch) are the vehicle attitude angles.

The propeller thrust model providing adjustable force to counteract buoyancy variations is:

$$T = (1 - \eta)\rho n^2 D^4 K_T(J) \tag{21}$$

$$J = \frac{U_a}{nD} \tag{22}$$

where J is the advance ratio, K_T is the thrust coefficient, ρ is the seawater density, U_a is propeller inflow velocity, n is the propeller speed and D is the propeller diameter.

The viscous hydrodynamic resistance reflecting the resistance effect of seawater is:

$$X = \rho X_{uu} u^2$$

$$Y = \rho Y_v uv \tag{23}$$

$$N = \rho N_r ur$$

where X_{uu}, Y_v, N_r are hydrodynamic coefficients and u, v, r are motion parameters.

The model is implemented in Python, with modules for buoyancy–gravity calculation, thrust generation, hydrodynamic resistance solving, and dynamic equation integration working collaboratively. It can real-time simulate the vehicle's motion states under varying buoyancy conditions, laying a foundation for verifying the adaptive performance of the close-loop controller.

4.2. Simulation Results

4.2.1. Initial Training Performance

The pure PPO agent, operating without any pre-existing control knowledge or policy priors, initially engages in extensive random exploration. This often results in suboptimal and erratic control behavior, as the agent must learn the dynamics of the depth-control task entirely through trial and error. In marked contrast, the proposed PPO-PID controller is strategically initialized within a proven control framework. It builds directly upon the stable, deterministic logic of a conventional PID controller, which provides immediate, regulatory performance. From this baseline, the reinforcement learning component then fine-tunes and optimizes the policy.

This fundamental difference in initial conditions is clearly illustrated in Figure 5, which presents a comparative visualization of the depth-tracking performance during the agents' initial learning phases. The test was conducted under a demanding scenario: maintaining a constant depth of 20 m in a near-natural buoyancy condition. As the figure demonstrates, the PPO-PID controller achieves immediate stability and superior reference tracking from the very beginning, with minimal overshoot and steady oscillation. Meanwhile, the pure PPO agent exhibits significant oscillatory excursions and unstable behavior, a direct consequence of its initial random exploration phase. This comparison underscores a critical advantage of the hybrid approach: it mitigates the high-risk, low-performance initial exploration typical of pure RL by bootstrapping learning from a safe and competent prior policy, thereby accelerating convergence and enhancing operational safety during training.

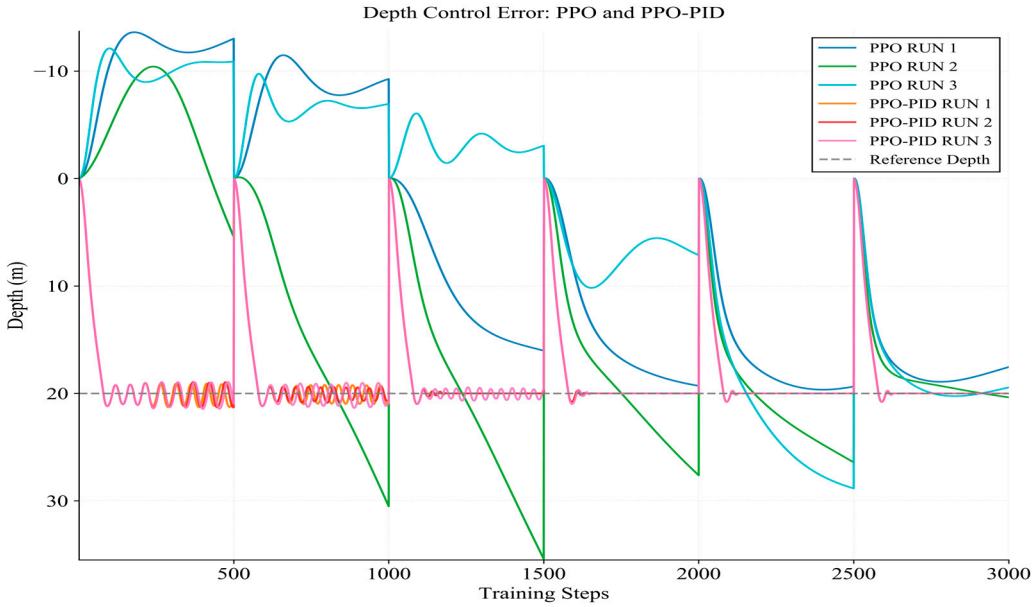

Figure 5. Depth control performance of PPO and PPO-PID during the initial phase.

4.2.2. Reward Convergence Analysis

A comparative visualization of the learning progression is presented in Figure 6, which illustrates the mean reward trajectory along with the variability (shaded region) across five independent runs for each method. Key contrasts are summarized as follows:

1. Initial Performance & Stability

The PPO-PID controller demonstrates immediately stable and superior performance from the onset of training. Its reward curves cluster within a high-value range (approximately −800 to −1100) from the first recorded step, indicating that the underlying PID logic provides a robust and safe baseline. In stark contrast, the pure PPO agent's initial rewards are orders of magnitude worse (as low as −15,541) and exhibit extreme variance between runs, reflecting its dependence on random exploration from a suboptimal policy.

2. Learning Efficiency & Convergence

The PPO-PID method shows rapid convergence to a near-optimal performance band. The variance between runs remains consistently low throughout the training process, indicating reliable and repeatable learning. Conversely, the pure PPO requires substantially more experience (tens of thousands of steps) to gradually and noisily improve, with some runs still showing poor convergence even at later stages. This underscores the sample efficiency gained by warm-starting the learning process with a functional controller.

3. Final Performance & Robustness

The steady-state performance of PPO-PID is both higher and more consistent than that of pure PPO. The final reward distribution for PPO-PID is tight and centered around a high value, while the pure PPO results are scattered across a much wider and lower range. This highlights the PPO-PID's robustness to different random initializations and its ability to reliably find a high-performing policy.

These results empirically validate the core thesis of the hybrid design. The pure PPO agent, lacking prior control knowledge, undergoes a prolonged, unstable, and potentially unsafe exploration phase. The proposed PPO-PID architecture mitigates this by leveraging the prior stability of the PID controller. The learning agent is tasked with fine-tuning an already functional policy rather than discovering one from scratch. This not only accelerates learning and improves final performance but also, critically, ensures safe and

acceptable performance throughout the entire training process—a vital consideration for real-world robotic systems.

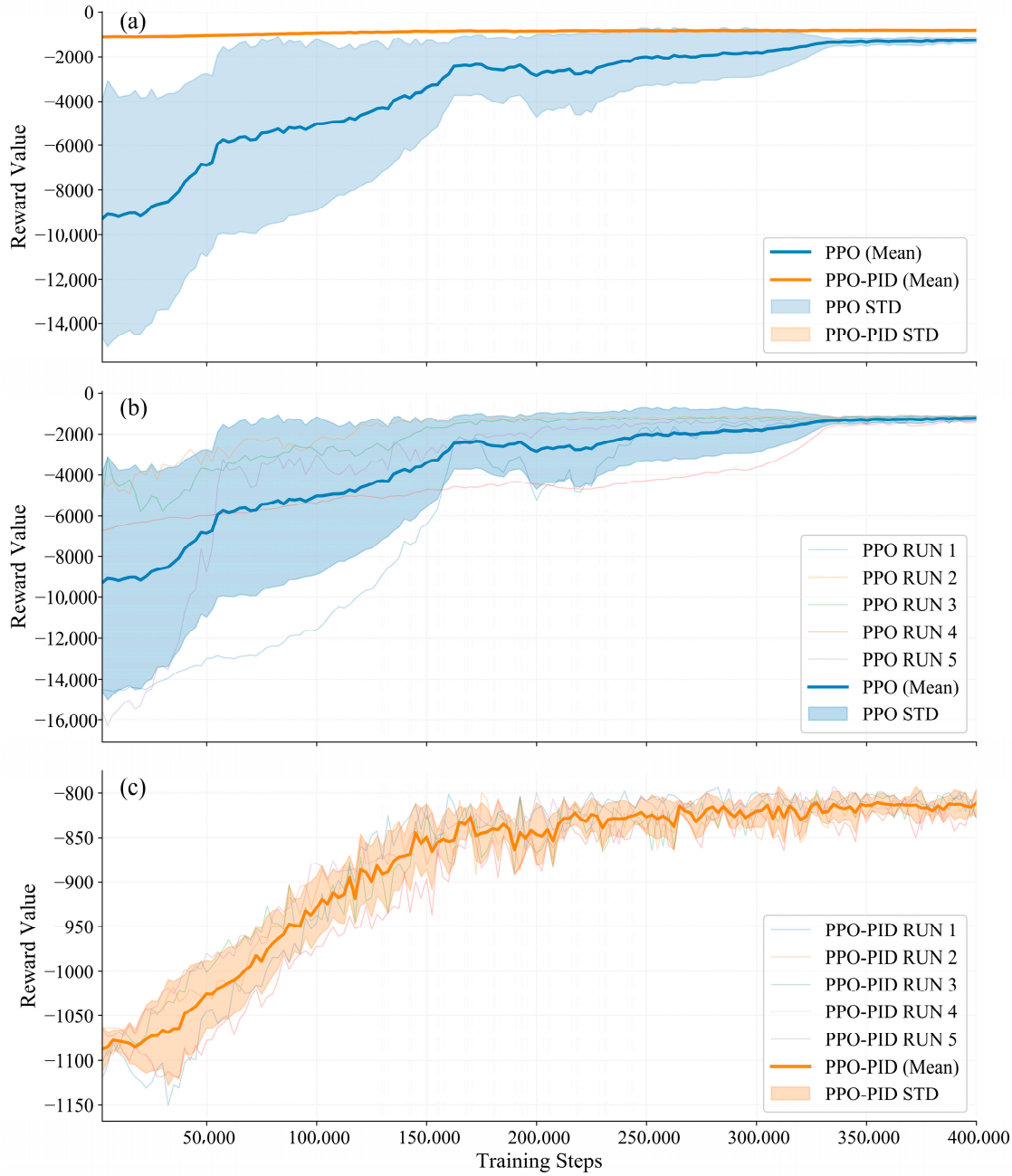

Figure 6. Reward values of PPO and PPO-PID during depth control training. (**a**) Direct comparison of the average reward (solid lines) and standard deviation bands (shaded areas) across five independent runs for both methods. (**b**) Detailed learning trajectories of the pure PPO agent. Individual runs (thin lines) show high initial variance, inconsistent convergence and slow learning progress. (**c**) Detailed learning trajectories of the proposed PPO-PID controller. All runs start from a high-performance baseline and converge rapidly to a narrow, high-reward region, illustrating robust, repeatable, and safe learning.

4.2.3. State Representation Effectiveness

The synthesized AELC state vector simultaneously captures the evolving dynamic relationship between the agent's actions and the system's output (error), while explicitly accounting for temporal delays. This structured representation reduces the learning burden on the RL algorithm by providing relevant, pre-processed temporal features. As

evidenced by the comparative results in Figure 7, the use of the AELC representation leads to higher initial and final rewards, faster convergence, and lower variance across training runs compared to a baseline state representation. This demonstrates its critical role in stabilizing training, improving sample efficiency, and ultimately yielding a more robust and high-performing control policy for adaptive depth control under buoyancy variations.

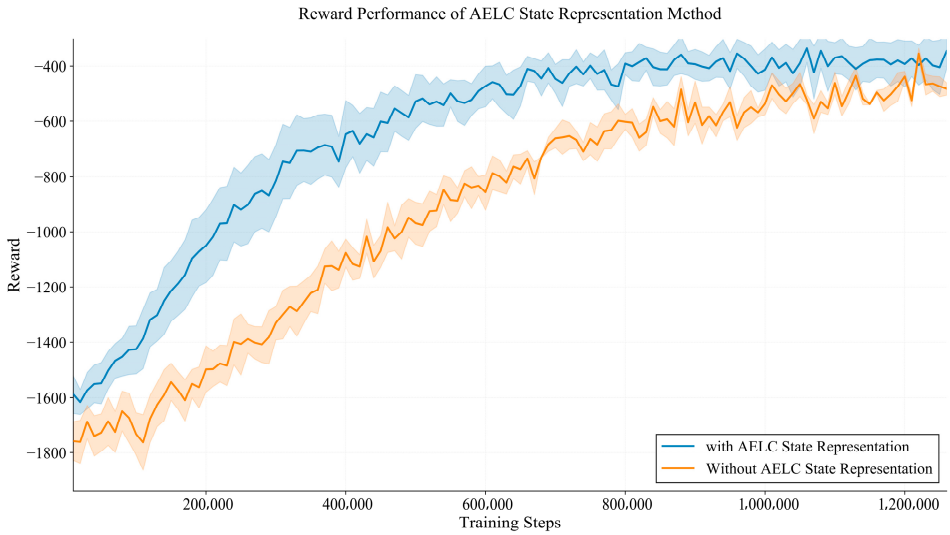

Figure 7. Reward of PPO-PID controllers with and without the AELC state representation. The controller utilizing the proposed AELC state representation exhibits a steeper learning curve, demonstrating the critical role of structured state information in improving sample efficiency for depth control.

4.2.4. Hardware-Friendly Actuation for Dynamic Adaptation During Buoyancy Variations

The closed-loop depth control performance of the proposed PPO-PID controller under buoyancy variations is shown in Figure 8. In the depth-tracking result, the actual depth closely follows the target depth even when random, sustained buoyancy disturbances are applied (Figure 8b). The corresponding actuator command (Figure 8c) reflects the effect of the integrated physical-constraint layer, which enforces smooth saturation, stepwise quantization, and dead-band filtering to produce a stable, hardware-friendly signal.

This constraint layer translates the neural network's policy output into hardware-executable commands that respect physical limits while maintaining the smoothness required for stable closed-loop dynamics. As a result, high-frequency command oscillations are directly suppressed, and the control output exhibits clear stepwise variations instead of continuous noisy fluctuations (Figure 8c). Moreover, the approach improves energy efficiency by eliminating the cumulative waste associated with rapid, minor adjustments. The stable and quantized output also reduces the workload on the power electronics, leading to cooler operation and extended hardware lifespan.

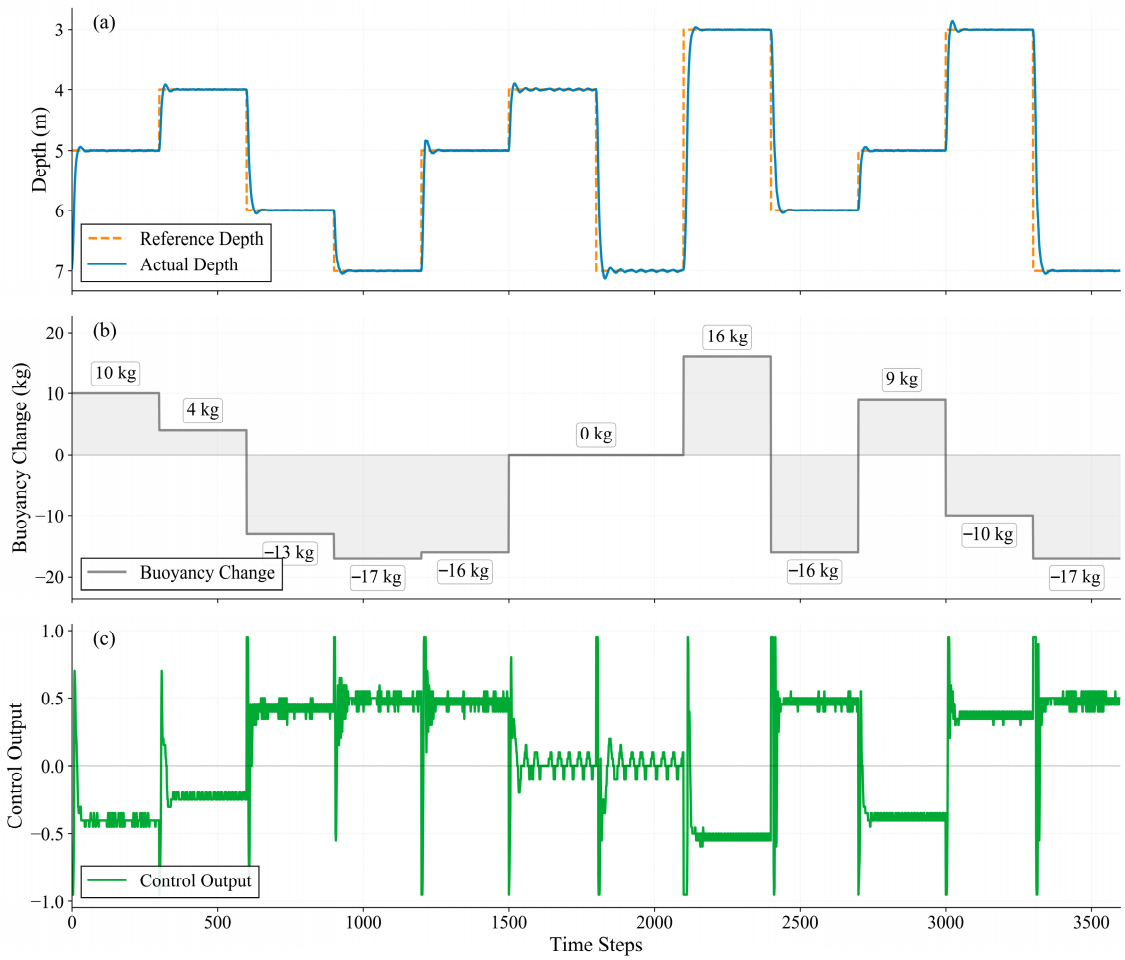

Figure 8. Closed-loop depth control performance of the proposed PPO-PID controller under buoyancy variations. (**a**) Depth tracking (**b**) Buoyancy disturbance: The controller is tested under random, sustained buoyancy change (**c**) Constrained control output: The final actuator command demonstrates the effect of the integrated physical constraint layer resulting in a stable, hardware-friendly signal.

4.2.5. Comparative Performance Analysis

The simulation dataset compares the depth-tracking performance of three control strategies—PID, PPO, and PPO-PID—under variable buoyancy conditions. The simulation scenario was designed to evaluate depth-tracking performance in reference depth (3 m → 8 m → 5 m → 10 m → 6 m) and abrupt variations in buoyancy (0 kg → +5 kg → −20 kg → +5 kg → 20 kg). Here, 0 kg corresponds to neutral buoyancy; +5 kg simulates increased buoyancy due to higher seawater density at greater depth; −20 kg represents the collection of heavy samples such as rocks; and +20 kg mimics the deployment of a payload, such as a seafloor detection device.

All controllers respond to buoyancy variations; however, the PPO-PID hybrid controller delivers the most stable and rapid compensation. In contrast, the PID controller exhibited pronounced transient deviations following abrupt buoyancy changes, while the pure PPO strategy struggled to fully counteract the buoyancy shift, leading to a persistent depth error. As summarized in Figure 9, the hybrid PPO-PID controller surpasses both the PID and PPO methods across all evaluated metrics, effectively minimizing steady-state error, reducing overshoot, and enhancing overall stability and robustness.

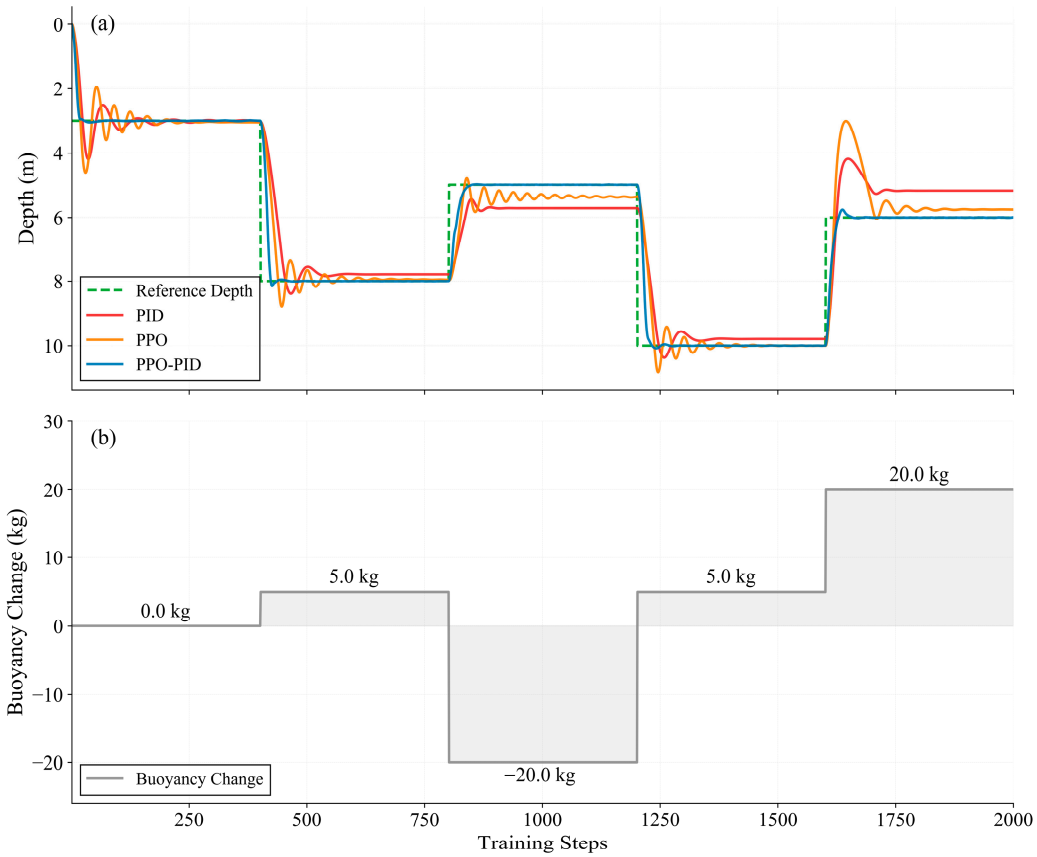

Figure 9. Comparative control performance of PID, PPO and PPO-PID: (**a**) Depth tracking; (**b**) Buoyancy disturbance.

The effectiveness of the PPO-PID controller stems from its two-layer structure: the PPO module provides adaptive, learning-driven optimization, while the PID framework ensures reliable and stable control performance. By integrating reinforcement learning with classical control theory, this hybrid approach achieves a robust and efficient control strategy suitable for dynamic and uncertain operational environments. It successfully balances the robustness of classical control with the adaptability of reinforcement learning, offering a promising methodology for underwater vehicles operating in dynamically changing conditions.

4.3. Pool Test Validation

To validate the feasibility and safety of the proposed algorithm in a real-world environment, physical experiments were conducted in a water tank facility. Owing to the substantial dimensions of the test setup, an AUV equipped with three vertical thrusters was adopted for the experiments (Figure 10). Its detailed technical specifications are presented in Table 1.

Table 1. Specifications of the AUV used in the pool and lake trials.

Parameter	Specification
Operating Depth	200 m
Overall Dimensions	1.7 m (Length) × 0.7 m (Width) × 0.4 m (Height)
Mass in Air	150 Kg
Horizontal Control	2 horizontal thrusters, 1 tunnel thruster
Vertical Control	3 tunnel thrusters

Navigation and Positioning	Ultra-short baseline (USBL) integrated with inertial navigation
Scientific Payload	Cameras, upward-looking sonar, Conductivity, Temperature, Depth (CTD) and other sensors

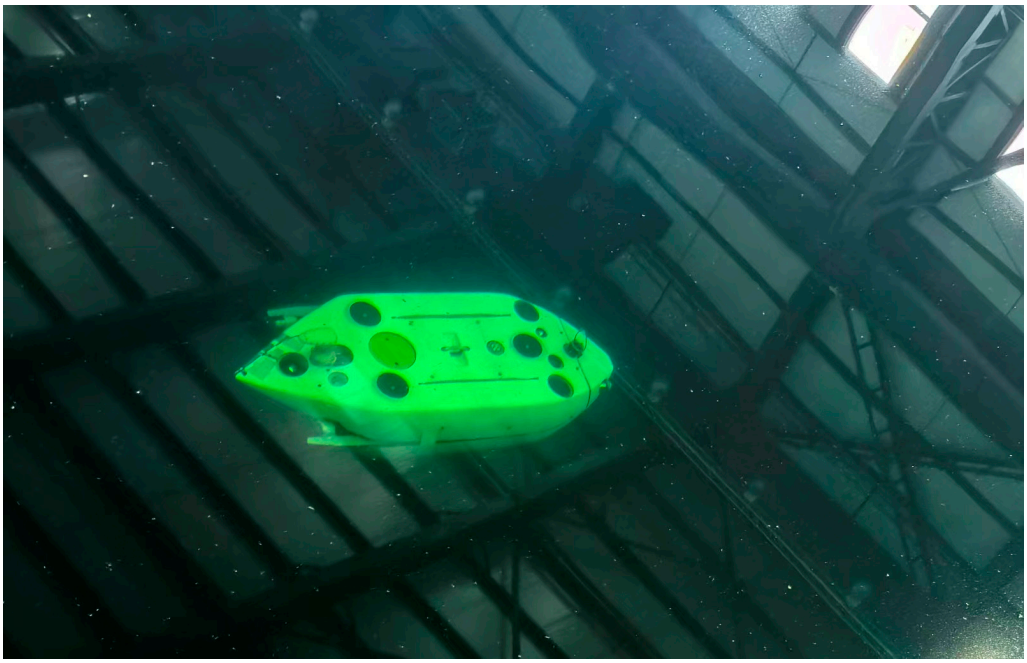

Figure 10. AUV undergoing training in the pool.

The proposed controller is deployed on an Intel Celeron N5105 processor—a quad-core, four-thread low-power chip with a base frequency of 2.0 GHz and a thermal design power (TDP) of only 10 W. This low-power hardware configuration is fully consistent with the practical engineering constraints of underwater autonomous equipment. The PPO-PID control system runs at a base frequency of 2 Hz, matching the frequency of the PID controller.

Initial trials with a pure PPO policy—which inherently generates exploratory random actions—were discontinued after only a few runs, as the approach posed unacceptable safety risks, including potential collisions with the tank walls.

4.3.1. Training from Scratch

In the absence of a hydrodynamic model, the PPO-PID controller was trained from scratch in the real tank. Figure 11 shows the overall profile with two setpoints at 3 m and 6 m. The controller achieved stable depth regulation after approximately 2100 s of training. However, when the target depth was changed, an additional 1500 s of training were required before stable tracking. The rectangles highlight the steady-state segments, which are magnified in panels (b) and (c). (b) During the steady phase at 3 m depth, the controller maintains the setpoint with an RMSE of 0.032 m. (c) At the 6 m setpoint, the depth is maintained with an RMSE of 0.019 m in the steady phase.

4.3.2. Pre-Trained Model Deployment

For the second test, the pre-trained model from the first experiment was loaded and directly deployed. As shown in Figure 12, the controller achieved stable depth regulation within approximately 180 s. After the target depth was subsequently changed, the system required only about 100 s to re-stabilize. The steady-state intervals, marked by rectangles,

are detailed in panels (b) and (c): during the 3 m steady-state phase, depth is maintained with an RMSE of 0.042 m, while at 6 m, the RMSE is further reduced to 0.014 m.

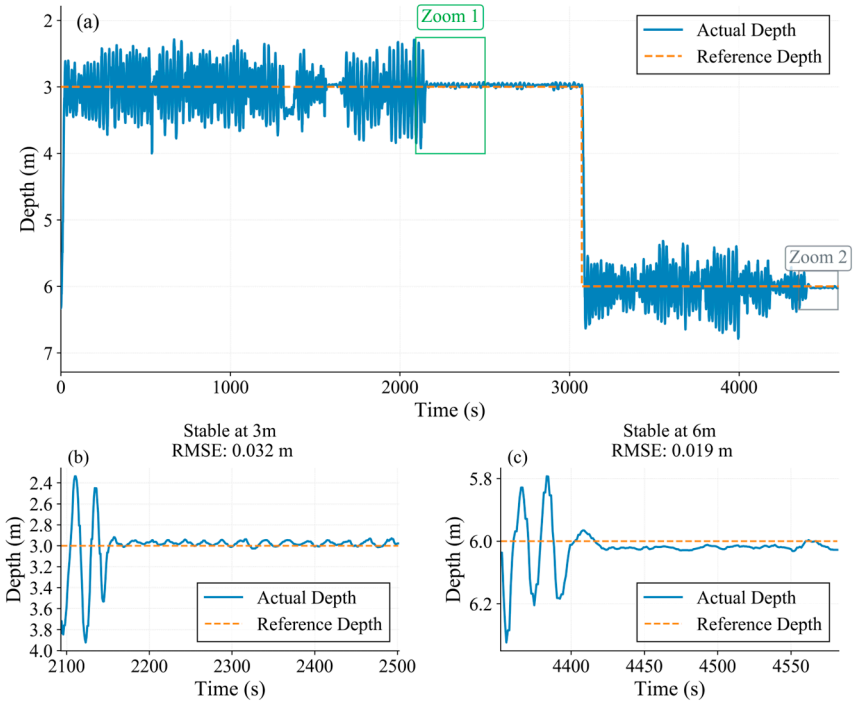

Figure 11. Time-domain depth control performance of the PPO-PID controller. (**a**) Depth tracking training process; the rectangles mark the steady-state segments that are zoomed in panels (**b**,**c**).

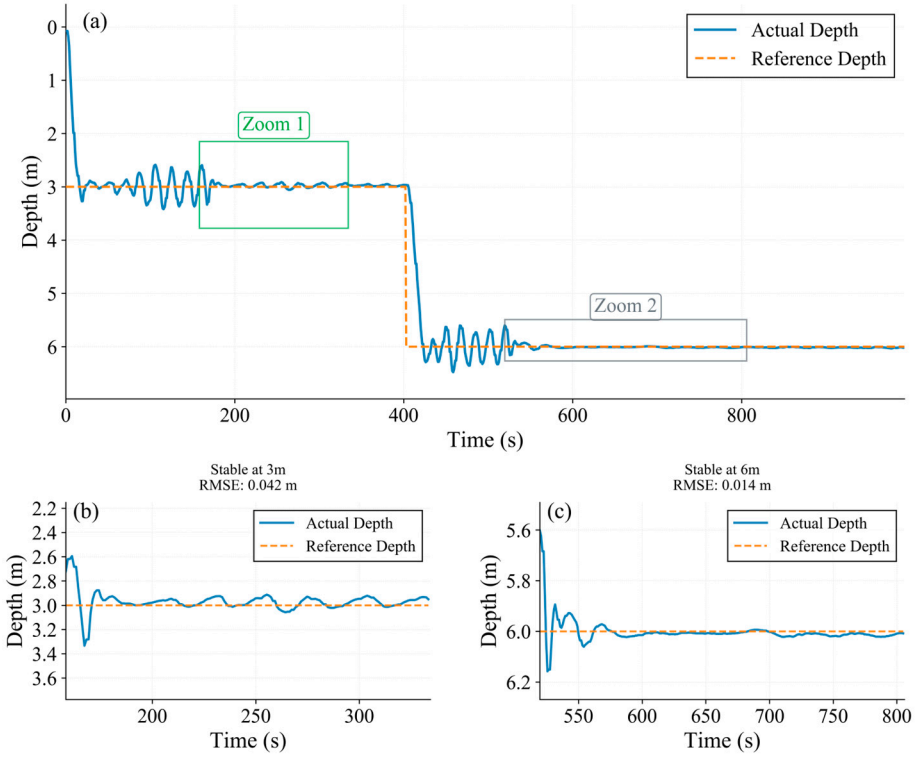

Figure 12. Depth control performance of the trained PPO-PID controller with pre-trained model loaded. (**a**) Depth profile with two setpoint plateaus at 3 m and 6 m. The controller stabilizes at each depth within <200 s. The rectangles highlight the steady-state intervals detailed in panels (**b**,**c**).

By employing a pre-trained model, the controller achieved stabilization at each depth within 200 s, representing a dramatic improvement in convergence speed and operational readiness over training from scratch, underscoring the practical advantages of transfer learning for real-world underwater robots deployment.

4.3.3. Comparative Test with PID

The experiment evaluates the performance of two depth control strategies for an underwater vehicle under neutral buoyancy conditions (Figure 13). The system was tasked with tracking depth setpoints of 3 m and 5 m. The transition between controllers occurred at approximately t = 1490 s, allowing for a direct comparison under identical conditions. The PPO-PID controller (blue region) shows smoother transitions and lower steady-state error than the PID controller (green region). RMSE values are annotated for each steady-state segment. The PPO-PID controller significantly outperforms the PID controller, with lower RMSE values at both setpoints.

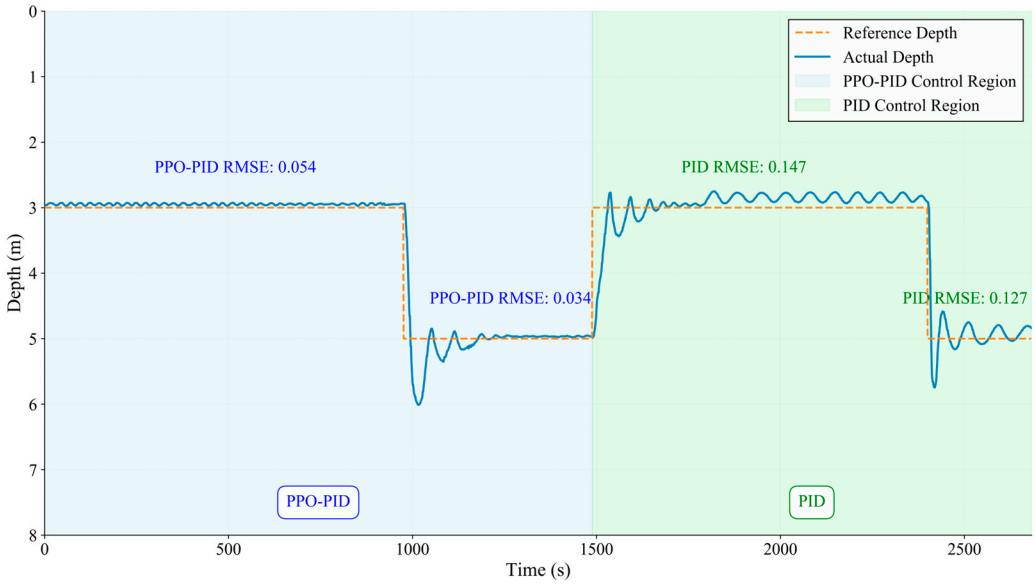

Figure 13. Depth tracking performance comparison between PPO-PID and PID controllers under neutral buoyancy.

Figure 14 compares the depth-tracking performance of the PID controller and the proposed PPO-PID controller under a persistent +5 kg buoyant disturbance. The depth reference is 5 m throughout the test. After the initial transient, the PID controller exhibits an RMSE of 0.414 m. The subsequent PPO-PID controller is divided into eight intervals, with the RMSE decreasing monotonically from 0.191 m in the first interval to 0.133 m in the last. This demonstrates the agent's ability to fine-tune the PID gains online. The shaded backgrounds (light-green for PID, light-blue for PPO-PID) emphasizes the performance contrast: the blue region shows a visibly tighter envelope around the reference line. These results indicate that reinforcement learning is able to compensate for the constant positive buoyancy more effectively than fixed-gain PID, without any prior knowledge of the disturbance magnitude.

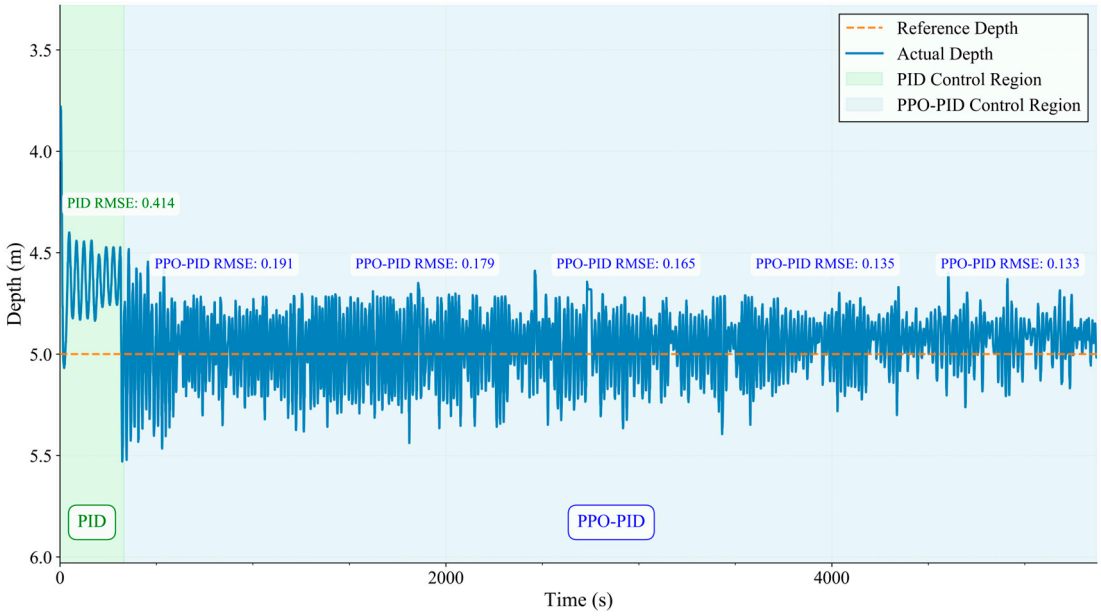

Figure 14. Depth-tracking experiment under a constant +5 kg buoyant load. PPO-PID shows a visibly tighter envelope around the reference line and is able to compensate for the constant positive buoyancy more effectively than fixed-gain PID.

4.4. Field Tests

To further validate the framework's robustness in real-world conditions, lake trials were conducted under wave and current disturbances (Figure 15). The test scenarios included depth control under near-neutral buoyancy and positive buoyancy, with reference depth changes.

Figure 15. AUV deployed for lake trials.

4.4.1. Near-Neutral Buoyancy Performance

Figure 16 compares the depth-tracking performance of the PID and PPO-PID controllers under near-neutral buoyancy conditions. The reference depth profile comprises four sequential setpoints: 9 m, 6 m, 3 m, and 1 m. The top panel displays the response of the PID controller, and the bottom panel corresponds to the PPO-PID controller. Both controllers successfully track the step changes in depth; however, the PPO-PID controller exhibits reduced overshoot during transitions. The root mean square error (RMSE) for each steady depth-holding segment is annotated in the figure, showing that the PPO-PID controller consistently achieves lower steady-state error, which indicates improved steady-state precision.

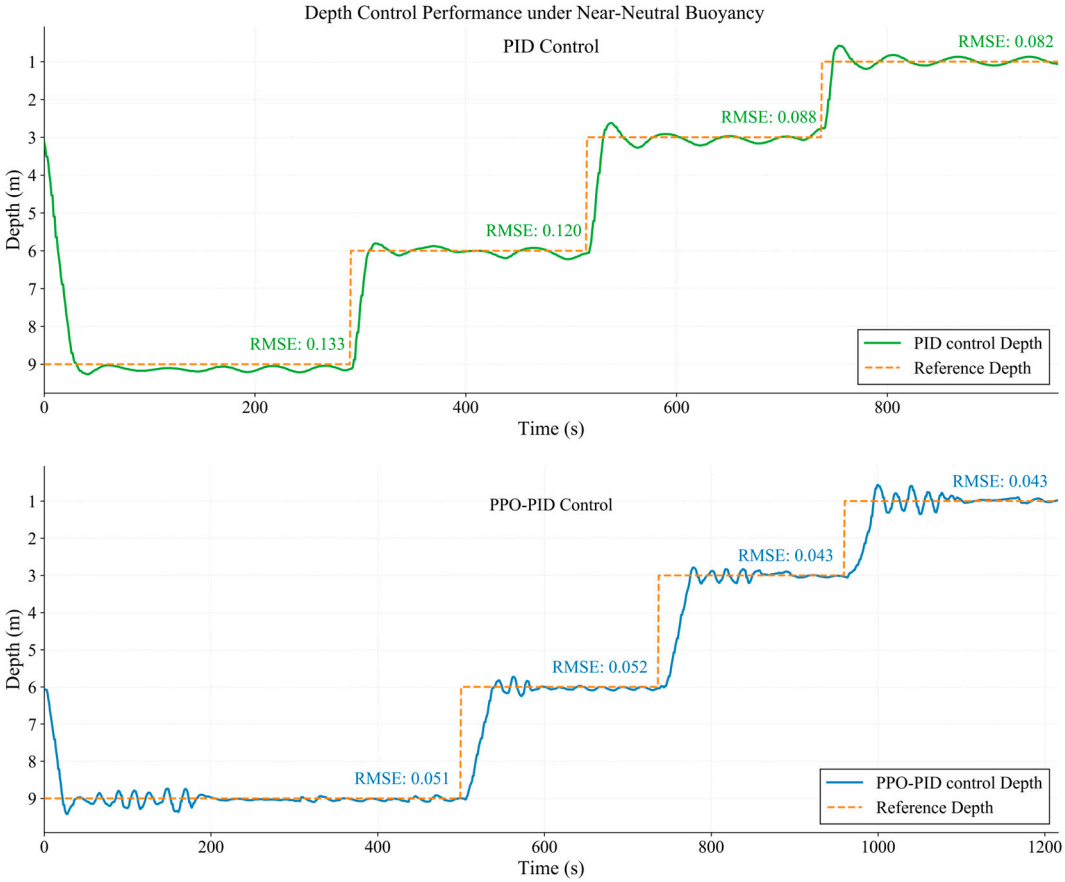

Figure 16. Comparative depth control performance under near-neutral buoyancy in lake.

4.4.2. Buoyancy Disturbance Performance

The depth-tracking performance of the PID and PPO-PID controllers under +3 kg disturbed buoyancy is presented in Figure 17. The top panel shows the response of the PID controller, while the bottom panel corresponds to the PPO-PID controller. Under positive buoyancy, the vehicle is subjected to a constant upward buoyant force, which requires sustained control effort to maintain the target depth. The root mean square error (RMSE) for each stable depth-holding segment is annotated in the figure. The PPO-PID controller demonstrates improved steady-state accuracy, indicating its enhanced ability to adapt to buoyancy-induced disturbances.

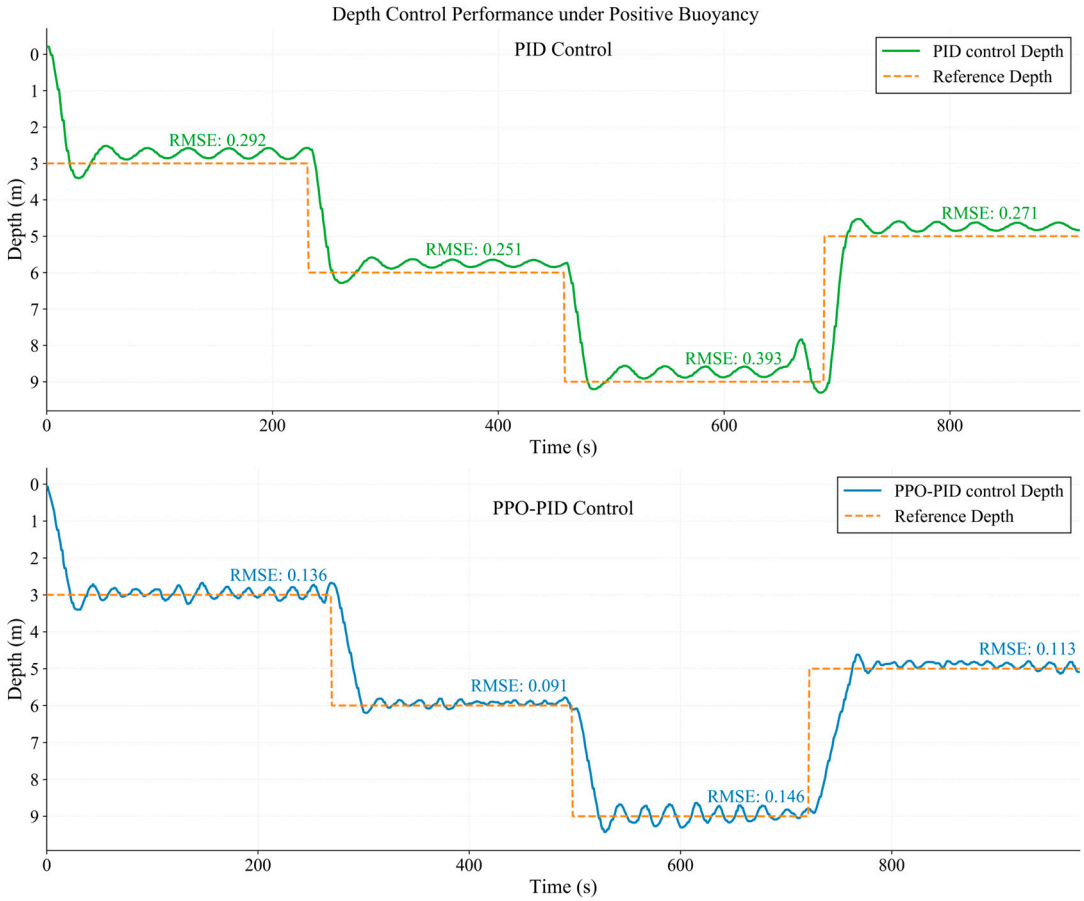

Figure 17. Comparative depth control performance under positive buoyancy.

5. Conclusions

This paper proposed a hybrid PPO-PID control framework for adaptive depth control of AUVs under significant buoyancy variations. By integrating the adaptive learning capability of PPO with the structural safety and reliability of PID control, the framework achieves online tuning of PID parameters while ensuring actuator safety through a layered output processing module. A dual-error state representation combined with actuator command buffering effectively compensates for system lag and inertia, improving learning efficiency and control responsiveness.

Comprehensive simulations, controlled pool tests, and real-world lake trials validate the proposed method. Comparative results show that the PPO-PID framework outperforms both conventional PID and pure PPO control. It excels in convergence speed, steady-state accuracy, control smoothness, and robustness under dynamic buoyancy changes and environmental disturbances. The framework provides a practical, safe, and efficient solution for depth control of sampling-capable underwater vehicles operating in uncertain underwater environments.

While this study focuses on depth control as a foundational validation of the proposed method, the framework is inherently generalizable to the design of closed-loop controllers for multi-degree-of-freedom (DOF) motion regulation. Future research will extend the framework to multi-degree-of-freedom (DOF) control, including heading, pitch and other motions. This expansion will enable full-pose stabilization of underwater vehicles in complex underwater missions.

Additionally, we plan to integrate transfer learning techniques to improve the framework's generalization capability across different AUV platforms, hydrodynamic conditions, and mission profiles. By pre-training policies in simulation and fine-tuning them

with limited real-world data, the learning process can be accelerated and made more sample-efficient.

Author Contributions: Conceptualization, J.W., Y.T. and S.L.; methodology, J.W. and H.B.; software, J.W. and H.B.; validation, H.B., C.C., D.Y. and J.L.; formal analysis, S.Y., C.C., D.Y. and J.L.; investigation, S.Y., J.L., X.C. and R.D.; resources, D.Y. and R.D.; data curation, J.W., S.Y. and X.C.; writing—original draft preparation, J.W.; writing—review and editing, C.C., X.C., R.D., Y.T. and S.L.; visualization, J.W. and H.B.; supervision, S.L.; project administration, Y.T. and S.L.; funding acquisition, Y.T. Y.T. and S.L. are co-corresponding authors with equal contribution. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (2023YFC2813000), the National Natural Science Foundation of China (62173320), and the Youth Innovation Promotion Association of Chinese Academy of Sciences (2022198).

Data Availability Statement: The original contributions presented in this study are included in the article material. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors thank Fuli Wang, Yu Cheng, Yang Lu, Xingya Yan, and Yongqiang Yu for their assistance with the experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AELC	Augmented Error and Lag-Compensated
AUV	Autonomous Underwater Vehicle
ARV	Autonomous and Remotely operated Vehicle
CTD	Conductivity, Temperature, Depth
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
DOF	Degree of Freedom
ML	Machine Learning
MPC	Model Predictive Control
PID	Proportional–Integral–Derivative
PPO	Proximal Policy Optimization
PPO-PID	Proximal Policy Optimization–Proportional–Integral–Derivative
RL	Reinforcement Learning
SAC	Soft Actor–Critic
SMC	Sliding Mode Control
USBL	Ultra-short Baseline
USV	Unmanned Surface Vehicle

References

1. Wang, J.; Chen, C.; Yu, D.; Zhou, J.; Tang, Y.; Cheng, Y.; Chen, X.; Dou, R.; Wang, F.; Liu, H.; et al. Development and Sea Trials of the Wenhai-1 Hybrid Underwater Vehicle for Comprehensive Survey of Gravity, Magnetism, Topography, and Sampling. In *OCEANS 2024—Singapore*; IEEE: New York, NY, USA, 2024; pp. 1–7. https://doi.org/10.1109/OCEANS51537.2024.10682151.
2. Wang, J.; Tang, Y.; Li, S.; Lu, Y.; Li, J.; Liu, T.; Jiang, Z.; Chen, C.; Cheng, Y.; Yu, D.; et al. The Haidou-1 hybrid underwater vehicle for the Mariana Trench science exploration to 10,908 m depth. *J. Field Robot.* **2024**, *41*, 1054–1079. https://doi.org/10.1002/rob.22307.
3. Bingul, Z.; Gul, K. Intelligent-PID with PD Feedforward Trajectory Tracking Control of an Autonomous Underwater Vehicle. *Machines* **2023**, *11*, 300. https://doi.org/10.3390/machines11020300.

4. Sahoo, A.; Dwivedy, S.; Robi, P. Development of a PID Control Strategy for a Compact Autonomous Underwater Vehicle. In Proceedings of the ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering. Volume 7B: Ocean Engineering. Glasgow, Scotland, UK, 2019. https://doi.org/10.1115/omae2019-95345.
5. Zheng, R.; Ma, Y.; Zang, B.; Han, X.; An, J. Stable Control for AUV's Near-bottom and Low-speed Sailing Based on Vertical Thruster. *Robot* **2016**, *38*, 588–592. https://doi.org/10.13973/j.cnki.robot.2016.0588.
6. Rout, R.; Subudhi, B. Inverse optimal self-tuning PID control design for an autonomous underwater vehicle. *Int. J. Syst. Sci.* **2017**, *48*, 367–375. https://doi.org/10.1080/00207721.2016.1186238.
7. Sarhadi, P.; Noei, A.; Khosravi, A. Model reference adaptive PID control with anti-windup compensator for an autonomous underwater vehicle. *Robot. Auton. Syst.* **2016**, *83*, 87–93. https://doi.org/10.1016/j.robot.2016.05.016.
8. Santhakumar, M.; Asokan, T. A Self-Tuning Proportional-Integral-Derivative Controller for an Autonomous Underwater Vehicle, Based on Taguchi Method. *J. Comput. Sci.* **2010**, *6*, 862–871. https://doi.org/10.3844/jcssp.2010.862.871.
9. Yan, Z.; Wang, M.; Xu, J. Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics. *Ocean Eng.* **2019**, *173*, 802–809. https://doi.org/10.1016/j.oceaneng.2019.01.008.
10. Panda, J. Machine learning for naval architecture, ocean and marine engineering. *J. Mar. Sci. Technol.* **2021**, *28*, 1–26. https://doi.org/10.1007/s00773-022-00914-5.
11. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-Level Control Through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533.
12. Gaskett, C.; Wettergreen, D.; Zelinsky, A. Reinforcement Learning Applied to the Control of an Autonomous Underwater Vehicle. In Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Monterey, CA, USA, 24–28 July 1999; pp. 125–131.
13. Carreras, M.; Yuh, J.; Batlle, J.; Ridao, P. A Behavior-Based Scheme Using Reinforcement Learning for Autonomous Underwater Vehicles. *IEEE J. Ocean. Eng.* **2005**, *30*, 416–427.
14. Masmitja, I.; Martín, M.; O'Reilly, T.; Kieft, B.; Palomeras, N.; Navarro, J.; Katija, K. Dynamic robotic tracking of underwater targets using reinforcement learning. *Sci. Robot.* **2023**, *8*, eade7811. https://doi.org/10.1126/scirobotics.ade7811.
15. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2016**, arXiv:1509.02971.
16. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. Soft Actor-Critic Algorithms and Applications. *arXiv* **2018**, arXiv:1812.05905.
17. Yu, R.; Shi, Z.; Huang, C.; Li, T.; Ma, Q. Deep Reinforcement Learning Based Optimal Trajectory Tracking Control of Autonomous Underwater Vehicle. In Proceedings of the 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 4958–4965.
18. Fang, Y.; Huang, Z.; Pu, J.; Zhang, J. AUV Position Tracking and Trajectory Control Based on Fast-Deployed Deep Reinforcement Learning Method. *Ocean Eng.* **2022**, *245*, 110452.
19. Carlucho, I.; De Paula, M.; Wang, S.; Petillot, Y.; Acosta, G.G. Adaptive Low-Level Control of Autonomous Underwater Vehicles Using Deep Reinforcement Learning. *Robot. Auton. Syst.* **2020**, *129*, 103565.
20. Jiang, P.; Song, S.; Huang, G. Attention-Based Meta-Reinforcement Learning for Tracking Control of AUV with Time-Varying Dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6388–6401.
21. Wang, N.; Wang, Y.; Zhao, Y.; Wang, Y.; Li, Z. Sim-to-Real: Mapless Navigation for USVs Using Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 895.
22. Li, G.; Shintake, J.; Hayashibe, M. Deep Reinforcement Learning Framework for Underwater Locomotion of Soft Robot. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2021 https://doi.org/10.1109/icra48506.2021.9561145.
23. Gao, J.; Li, Y.; Chen, Y.; He, Y.; Guo, J. An Improved SAC-Based Deep Reinforcement Learning Framework for Collaborative Pushing and Grasping in Underwater Environments. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 2512814. https://doi.org/10.1109/tim.2024.3379048.
24. Guan, Z.; Yamamoto, T. Design of a Reinforcement Learning PID Controller. *IEEJ Trans. Electr. Electron. Eng.* **2021**, *16*, 1354–1360.
25. Yu, X.; Fan, Y.; Xu, S.; Ou, L. A Self-Adaptive SAC-PID Control Approach Based on Reinforcement Learning for Mobile Robots. *Int. J. Robust Nonlinear Control* **2022**, *32*, 9625–9643.
26. Liu, R.; Cui, Z.; Lian, Y.; Li, K.; Liao, C.; Su, X. AUV Adaptive PID Control Method Based on Deep Reinforcement Learning. In Proceedings of the 2023 China Automation Congress (CAC), Chongqing, China, 25–27 November 2023; pp. 2098–2103.

27. Wang, Y.; Hou, Y.; Lai, Z.; Cao, L.; Hong, W.; Wu, D. An Adaptive PID Controller for Path Following of Autonomous Underwater Vehicle Based on Soft Actor–Critic. *Ocean Eng.* **2024**, *307*, 118171.
28. Guan, W.; Xi, Z.; Cui, Z.; Zhang, X. Adaptive trajectory controller design for unmanned surface vehicles based on SAC-PID. *Brodogradnja* **2025**, *76*, 76206. https://doi.org/10.21278/brod76206.
29. Wang, C.; Du, J.; Wang, J.; Ren, Y. AUV Path Following Control using Deep Reinforcement Learning Under the Influence of Ocean Currents. In Proceedings of the 2021 5th International Conference on Digital Signal Processing, Chengdu, China, 2021. https://doi.org/10.1145/3458380.3459041.
30. Zhang, Y.; Che, J.; Hu, Y.; Cui, J.; Cui, J. Real-Time Ocean Current Compensation for AUV Trajectory Tracking Control Using a Meta-Learning and Self-Adaptation Hybrid Approach. *Sensors* **2023**, *23*, 6417. https://doi.org/10.3390/s23146417.
31. Kumar, N.; Rani, M. An efficient hybrid approach for trajectory tracking control of autonomous underwater vehicles. *Appl. Ocean. Res.* **2020**, *95*, 102053. https://doi.org/10.1016/j.apor.2020.102053.