

## Article

# Direct X-Rudder Path-Following Control for Underactuated AUVs via TIB-CSAC

Jiehui Tan <sup>1,2</sup>, Yushan Sun <sup>1,2,3,\*</sup>, Liwen Zhang <sup>1,2</sup> , Puxin Chai <sup>1,2</sup> and Zhan Liu <sup>1,2</sup><sup>1</sup> National Key Laboratory of Autonomous Marine Vehicle Technology, Harbin Engineering University, Harbin 150001, China<sup>2</sup> College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China<sup>3</sup> Shandong Future Robot Co., Ltd., Weihai 264209, China

\* Correspondence: sunyushan@hrbeu.edu.cn

## Abstract

To improve the path-following performance of an underactuated autonomous underwater vehicle (AUV) under varying path geometries and desired velocities, this study proposes a direct X-rudder control method based on Task-Informed Inductive-Bias Conservative Soft Actor–Critic (TIB-CSAC). The proposed method directly learns the X-rudder control policy from the path-following information of the current and subsequent path segments in a data-driven way, thereby avoiding the complex design and manual tuning of guidance laws and attitude controllers for rudder command generation. To support such two-segment policy learning, a task-informed inductive-bias encoder is proposed to construct structured and conditioned state representations, thereby improving sample efficiency and overall training quality. In addition, given the long-tail characteristics of task difficulty in agent training, a multi-head conservative value evaluation mechanism is incorporated to mitigate return drawdowns induced by challenging tasks in the tail stage of training and to enhance tail-stage convergence stability. The path-following performance is validated in three representative scenarios with different path pitch, path heading variations, and desired surge velocity conditions. The results show that, compared with the baseline soft actor–critic (SAC) method, TIB-CSAC improves multiple vertical and horizontal error metrics, including maximum absolute error, mean absolute error, tail error, and error threshold exceedance ratio. These results indicate that TIB-CSAC not only improves overall adherence to the reference path, but also more effectively suppresses extreme errors and tail errors, thereby demonstrating stronger path-following robustness and reliability.

**Keywords:** autonomous underwater vehicles; X-rudder control; path following; deep reinforcement learning; soft actor–critic; intelligent control



Academic Editor: Jinfen Zhang

Received: 25 April 2026

Revised: 4 June 2026

Accepted: 10 June 2026

Published: 14 June 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Owing to their simple structure and high maneuvering efficiency, X-rudder autonomous underwater vehicles (AUVs) have become important platforms for a wide range of underwater applications, including environmental monitoring, offshore infrastructure inspection, and marine resource exploration. As underwater missions become increasingly diverse, AUVs are expected to exhibit strong generalization capability and reliable performance under a broad range of operating conditions.

Path-following is a fundamental capability of AUVs. It enables an AUV to move along a reference path while minimizing its position error. In practice, the reference path

is commonly composed of multiple straight-line segments, and the path-following task can be characterized by parameters such as the path heading angle, path pitch angle, and desired surge velocity. However, achieving path-following performance that is both robust and strongly generalizable remains challenging. First, different task-parameter settings lead to substantial variations in task difficulty and impose different maneuvering demands. This challenge is particularly pronounced during path-segment transitions, where the AUV must rapidly adjust its motion and reduce tracking errors within a limited spatial range; these transitions are therefore more difficult than straight-segment following. Moreover, the underactuated nature of X-rudder AUVs, nonlinear hydrodynamic effects, and coupling among multiple motion variables further increase the complexity of unified controller design. As the task-generalization range expands, the workload and complexity associated with multi-parameter tuning also increase.

Due to the absence of direct lateral and vertical actuation in underactuated AUVs, conventional path-following methods typically use a guidance law to transform position errors into desired attitudes. These commands are then tracked by a controller, such as prescribed performance control [1], sliding-mode control [2,3], or model predictive control [4]. However, such model-based methods rely on accurate hydrodynamic models, and parameter identification is difficult and costly. Model-free controllers, such as proportional–integral–derivative (PID) controllers [5,6] and S-plane controllers [7–9], are easier to implement but usually rely on expert tuning. Moreover, under significant variations in path geometry and desired surge velocity, these hierarchical schemes often require re-tuning of the guidance law or controller [10], which limits their adaptability and generalization capability.

Deep reinforcement learning (DRL), with its self-learning capability and data-driven decision-making mechanism, provides a promising approach to path-following [5,11]. Sun et al. [12] proposed an improved deep deterministic policy gradient method (OSAM-DDPG) that learns a control policy conditioned on attitude and velocity errors. However, when a line-of-sight (LOS) guidance law with fixed parameters is used, its path-following performance under different operating conditions may be limited. Fang et al. [13] applied DDPG to achieve three-degree-of-freedom control at constant speed with variable pitch and yaw, while realizing path-following via waypoint tracking. Wang et al. [14] developed a simplified DDPG method that does not require prediction of future rewards and validated its performance on a helical path and an inclined straight path under a single constant desired speed. Wang et al. [15] proposed a SAC-based adaptive PID controller and verified its attitude-control performance in LOS-guided path-following, although its effectiveness in direct position-error regulation remains to be further investigated. Zhang et al. [16] proposed an improved DDPG-based path-following method integrated with a LOS guidance law and trained the algorithm on two fixed paths under a single constant desired speed. Zhang et al. [17] proposed a deep interactive reinforcement learning method, and Dong et al. [18] developed an improved SAC-based two-dimensional path-following method. In both studies, the agent was trained and evaluated using the same path geometries; therefore, its path-following performance across diverse path geometries remains an open issue.

Despite these advances, current DRL-based path-following studies still have limitations in state construction when strong generalization is required. On the one hand, many DRL methods rely on manually designed guidance laws with fixed parameters, making it difficult to adapt to path-following tasks with different parameter configurations. On the other hand, multiple types of original path-following information, including position errors, path heading/pitch variations, and progress relative to adjacent segments, are usually mapped by the guidance law into a small number of desired attitude commands [19]. If the agent mainly receives these attitude commands as inputs, task-level contextual in-

formation may not be explicitly represented in the agent state. Such local and indirect state representations may create a representation bottleneck, limiting the agent's conditional learning capability across operating conditions and weakening its generalization performance [20,21].

Existing path-following studies also have limitations in agent training. Some studies restrict agent training to basic control tasks involving attitude and velocity errors, and some methods are trained under only a single surge-speed setting or a single path-geometry condition. Such limited coverage of the training-task distribution can easily lead to policy overfitting, thereby degrading path-following performance in other tasks. In domain-randomized training, challenging operating conditions, in which task parameters such as the path heading angle, path pitch angle, and desired surge velocity simultaneously approach their boundary values, occur with relatively low probability. Meanwhile, compared with straight-segment following, path-segment transitions are more difficult and are associated with sparser experience samples. As a result, task difficulty spans a wide range under domain-randomized training and exhibits a pronounced long-tail distribution. However, existing methods such as DDPG and SAC are essentially risk-neutral expected-return optimization methods. Although they may achieve satisfactory average performance under most regular operating conditions, their path-following performance under more challenging tasks may remain limited.

To address the limited generalization capability and robustness of underactuated X-rudder AUVs under diverse path-following conditions, this study proposes a direct X-rudder control method based on Task-Informed Inductive-Bias Conservative Soft Actor-Critic (TIB-CSAC) for AUV path-following. The main contributions are summarized as follows:

- (1) A direct X-rudder control framework is proposed. In this framework, the X-rudder control policy is learned directly from two-segment path-following information, avoiding the complex hierarchical design of guidance laws and attitude controllers as well as the associated coupled parameter tuning.
- (2) A task-informed inductive-bias state encoder is developed to construct structured and task-conditioned representations of two-segment path-following states for direct policy learning. This design reduces the representation burden caused by direct concatenation of heterogeneous state variables and improves sample efficiency.
- (3) A multi-head conservative value-evaluation mechanism is introduced to improve value assessment for high-uncertainty state-action pairs, thereby reducing return drawdowns induced by challenging path-following tasks and improving tail-stage convergence stability.
- (4) The proposed method is validated in three representative scenarios with distinct path geometries and desired surge velocities. The results show improvements across multiple error metrics, including the maximum absolute error, mean absolute error, tail error, and threshold-exceedance proportion, demonstrating enhanced generalization capability and path-following performance under diverse operating conditions.

The remainder of this paper is organized as follows. Section 2 presents the problem statement. Section 3 describes the proposed method. Section 4 analyzes and discusses the experimental results, and Section 5 concludes the paper and outlines future work.

## 2. Problem Statement

### 2.1. Underactuated AUV Model

In this study, the AUV is equipped with a main thruster and an X-rudder configuration. The numbering and rotation directions of the X-rudders, as well as the Earth-fixed (O-XYZ)

and body-fixed ( $O_B$ - $X_B$  $Y_B$  $Z_B$ ) coordinate systems, are shown in Figure 1. The kinematic and dynamic model of the AUV is given by [22]

$$\dot{\eta} = J(\eta)v \quad (1)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau^a + \tau^d \quad (2)$$

where  $J$ ,  $M$ ,  $C$ ,  $D$ , and  $g$  represent the kinematic transformation matrix, inertia matrix, Coriolis-centripetal matrix, hydrodynamic damping matrix, and restoring-force vector, respectively.  $\eta = [x, y, z, \theta, \psi]^T$  and  $v = [u, v, w, q, r]^T$  denote the position-attitude vector in the Earth-fixed frame and the velocity vector in the body-fixed frame, respectively.

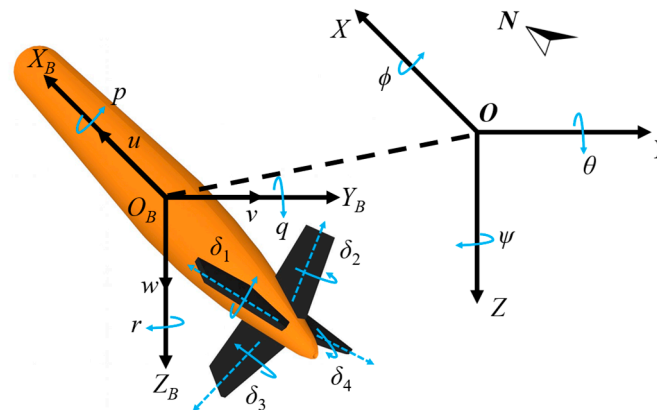


Figure 1. Coordinate system.

In practical AUV design, roll motion is usually expected to remain close to zero to ensure sensor and payload stability. This condition can be maintained passively by the hydrostatic restoring moment and hydrodynamic roll damping. Therefore, under the assumptions that gravity and buoyancy are in equilibrium, the center of buoyancy lies in the vertical plane, and sufficient hydrostatic restoring moment is available for near-zero roll motion, the kinematic and dynamic model can be written as [23,24]

$$\begin{cases} \dot{x} = u \cos(\psi) \cos(\theta) - v \sin(\psi) + w \cos(\psi) \sin(\theta) \\ \dot{y} = u \sin(\psi) \cos(\theta) + v \cos(\psi) + w \sin(\psi) \sin(\theta) \\ \dot{z} = -u \sin(\theta) + w \cos(\theta) \\ \dot{\theta} = q \\ \dot{\psi} = r / \cos(\theta) \end{cases} \quad (3)$$

$$\begin{cases} \dot{u} = (m_{22}vr - m_{33}wq - X_u u - X_{u|u}|u| + \tau_u^a + \tau_u^d) / m_{11} \\ \dot{v} = (-m_{11}ur - Y_v v - Y_{v|v}|v| + \tau_v^d) / m_{22} \\ \dot{w} = (m_{11}uq - Z_w w - Z_{w|w}|w| + \tau_w^d) / m_{33} \\ \dot{q} = ((m_{33} - m_{11})uw - M_q q - M_{q|q}|q| - \overline{BG}_z W \sin \theta + \tau_q^a + \tau_q^d) / m_{55} \\ \dot{r} = ((m_{11} - m_{22})uv - N_r r - N_{r|r}|r| + \tau_r^a + \tau_r^d) / m_{66} \end{cases} \quad (4)$$

where  $W$  is the AUV weight, and  $\overline{BG}_z$  is the vertical offset between the centers of mass and buoyancy.  $m_{(\cdot)}$  denotes the diagonal inertia terms of the mass matrix  $M$ , including added mass.  $(X_u, Y_v, Z_w, M_q, N_r)$  and  $(X_{u|u}, Y_{v|v}, Z_{w|w}, M_{q|q}, N_{r|r})$  represents linear and nonlinear damping coefficient, respectively.  $\tau^d = [\tau_u^d, \tau_v^d, \tau_w^d, \tau_q^d, \tau_r^d]^T$  is the environmental disturbance.  $\tau^a = [\tau_u^a, 0, 0, \tau_q^a, \tau_r^a]^T$  is the actuator input vector, where  $\tau_u^a$  is the thruster input, and  $\tau_q^a$  and  $\tau_r^a$  are the X-rudder inputs for pitch and yaw motion, respectively.

Unlike a conventional cross-rudder configuration, in which the horizontal and vertical rudders mainly generate pitch and yaw moments separately, the X-rudder contains four inclined control surfaces. As shown in Equation (5), each rudder contributes to both  $\tau_q^a$  and  $\tau_r^a$ ; therefore, the pitch and yaw moments are generated by the combined deflections of all four rudders. This configuration improves rudder-utilization efficiency but also introduces a more complex rudder-moment coupling relationship.

$$\begin{aligned}\tau_q^a &= u^2(M_{\delta_1}\delta_1 + M_{\delta_2}\delta_2 + M_{\delta_3}\delta_3 + M_{\delta_4}\delta_4) \\ \tau_r^a &= u^2(N_{\delta_1}\delta_1 + N_{\delta_2}\delta_2 + N_{\delta_3}\delta_3 + N_{\delta_4}\delta_4)\end{aligned}\quad (5)$$

where  $M_\delta$  and  $N_\delta$  are the hydrodynamic parameters of the rudder.

## 2.2. Path Following Objective

The desired path is represented as a polyline of  $n^p$  straight-line segments, where the  $i$ -th segment with endpoints  $P_{i-1}^p$  and  $P_i^p$  ( $i = 1, 2, \dots, n^p$ ) is defined by Equation (6).

$$P_i^p = \begin{bmatrix} x_i^p \\ y_i^p \\ z_i^p \end{bmatrix} = P_{i-1}^p + L_i^p \begin{bmatrix} \cos \psi_i^p \cos \theta_i^p \\ \sin \psi_i^p \cos \theta_i^p \\ -\sin \theta_i^p \end{bmatrix}, \quad \begin{cases} |\psi_i^p - \psi_{i-1}^p| \leq \Delta\psi_{\max} \\ |\theta_i^p| \leq \theta_{\max}^p \\ L_i^p \geq L_{\min} \end{cases} \quad (6)$$

where  $\Delta\psi_{\max}^p$  denotes the maximum variation between adjacent path headings  $\psi_i^p$ , and  $\psi_0^p$  is set to the initial yaw angle of the AUV.  $\theta_{\max}^p$  represents the maximum absolute value of the path pitch angle  $\theta_i^p$ , and  $L_{\min}$  is the minimum path-segment length  $L_i^p$ .

The vertical error is defined as the perpendicular distance from the AUV to the projection point on the path segment:

$$e_i^v = (z_i^p - z_{i-1}^p)\varepsilon_i + z_{i-1}^p - z \quad (7)$$

$$\varepsilon_i = (\overrightarrow{P_{i-1}^p P_{\text{AUV}}} \cdot \overrightarrow{P_{i-1}^p P_i^p}) / |\overrightarrow{P_{i-1}^p P_i^p}|^2 \quad (8)$$

where  $\varepsilon_i$  is an along-track parameter based on the AUV's position  $P_{\text{AUV}} = [x, y, z]^T$  relative to the path segment  $P_{i-1}^p P_i^p$ .

The horizontal error is defined as the horizontal offset with the minimum absolute value among the two adjacent path segments.

$$e_i^h = \begin{cases} d_k^h, k = \operatorname{argmin}_{j \in \{i, i+1\}} |d_j^h| & \text{if } i < n^p \\ d_{n^p}^h & \text{if } i = n^p \end{cases} \quad (9)$$

$$d_i^h = \overrightarrow{P_{i-1}^h P_{\text{AUV}}} \times \overrightarrow{P_{i-1}^h P_i^h} / |\overrightarrow{P_{i-1}^h P_i^h}| \quad (10)$$

where  $d_i^h$  denotes the horizontal distance from the AUV to the  $i$ -th path segment; and  $P_i^h = [x_i^p, y_i^p]^T$  and  $P_{\text{AUV}}^h$  denote the horizontal position.

In summary, this study aims to minimize both the horizontal and vertical position errors, as shown in Equation (11).

$$\min J = \sum_t (|e_{ie}^h(t)| + |e_{ie}^v(t)|) \quad (11)$$

$$i^e \leftarrow i^e + 1, \text{ if } \varepsilon_{ie} > 1 \quad (12)$$

where  $i^e$  denotes the path-segment index used to compute the position error, and  $\varepsilon_{i^e}$  is the along-track parameter in Equation (8). The index  $i^e$  is initialized to one and is incremented when  $\varepsilon_{i^e} > 1$ , indicating that the AUV has passed the endpoint of the current segment.

### 3. X-Rudder Path Following Control Based on TIB-CSAC

#### 3.1. Markov Decision Process Design

##### 3.1.1. States

Let  $i^o$  denote the index used for state observation. As shown in Equations (13) and (14),  $i^o$  is initialized to one and is incremented when the horizontal error relative to the next target segment remains within a prescribed threshold for a duration  $t_{\min}^o$ , thereby preserving state continuity during path transitions. To avoid excessive lag behind  $i^e$ ,  $i^o$  is also forcibly incremented when  $i^e - i^o > 2$ . After each update of  $i^o$ ,  $t^o$  is reset to zero.

$$i^o \leftarrow i^o + 1, \text{ if } t^o \geq t_{\min}^o \text{ or } i^e - i^o > 2 \quad (13)$$

$$t^o = \begin{cases} t^o + \Delta t & \text{if } |d_{i^o+1}^h| \leq d_{\min}^h \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

The overall observation state vector consists of four components:

$$\mathbf{s} = [\mathbf{s}_{\text{dyn}}^T, \mathbf{s}_{\text{task}}^T, \mathbf{s}_{i^o}^T, \mathbf{s}_{i^o+1}^T]^T \quad (15)$$

where  $\mathbf{s}_{\text{dyn}}$  denotes the general dynamic state describing the current motion and actuator conditions of the AUV.

$$\mathbf{s}_{\text{dyn}} = [\mathbf{v}^T, \dot{\psi}, \delta^T]^T \quad (16)$$

where  $\mathbf{v} = [u, v, w, q, r]^T$  denotes the velocity vector, and  $\delta = [\delta_1, \delta_2, \delta_3, \delta_4]^T$  denotes the X-rudder-angle vector.  $\mathbf{s}_{\text{task}}$  is the task-level state that provides information on the desired surge velocity, path-geometry variations, and path-following progress:

$$\mathbf{s}_{\text{task}} = [u, \Delta\psi_{i^o}^p, \Delta\theta_{i^o}^p, d_{i^o}^p, d_{i^o}^{p'}]^T \quad (17)$$

where  $\Delta\theta_{i^o}^p = \theta_{i^o+1}^p - \theta_{i^o}^p$  and  $\Delta\psi_{i^o}^p = \psi_{i^o+1}^p - \psi_{i^o}^p$  denote the changes in path pitch and path heading, respectively.  $d_{i^o}^p$  and  $d_{i^o}^{p'}$  are the distances from the AUV projection points on segments  $\mathbf{P}_{i^o-1}^p \mathbf{P}_{i^o}^p$  and  $\mathbf{P}_{i^o}^p \mathbf{P}_{i^o+1}^p$  to waypoint  $\mathbf{P}_{i^o}^p$ , encoding the progress of the AUV along the two adjacent segments.

$$d_{i^o}^p = (\overrightarrow{\mathbf{P}_{\text{AUV}} \mathbf{P}_{i^o}^p} \cdot \overrightarrow{\mathbf{P}_{i^o-1}^p \mathbf{P}_{i^o}^p}) / |\mathbf{P}_{i^o-1}^p \mathbf{P}_{i^o}^p| \quad (18)$$

$$d_{i^o}^{p'} = (\overrightarrow{\mathbf{P}_{i^o}^p \mathbf{P}_{\text{AUV}}} \cdot \overrightarrow{\mathbf{P}_{i^o}^p \mathbf{P}_{i^o+1}^p}) / |\mathbf{P}_{i^o}^p \mathbf{P}_{i^o+1}^p| \quad (19)$$

$\mathbf{s}_i$  ( $i = i^o, i^o + 1$ ) denotes the segment-specific state that describes the relative path-following error for the current or subsequent path segment. It provides preview information for the upcoming transition and reduces abrupt changes in the policy input.

$$\mathbf{s}_i = [\Delta\theta_i^R, e_i^v, \dot{e}_i^v, \Delta\psi_i^R, d_i^h, \dot{d}_i^h] \quad (20)$$

where  $\Delta\theta_i^R = \theta_i^p - \theta$  and  $\Delta\psi_i^R = \psi_i^p - \psi$  represent the relative pitch and yaw angles with respect to segment  $\mathbf{P}_{i-1}^p \mathbf{P}_i^p$ , respectively.  $e_i^v$  and  $\dot{e}_i^v$  are the vertical error defined in Equation (7) and its rate, respectively; and  $d_i^h$  and  $\dot{d}_i^h$  are the horizontal distance defined in Equation (10) and its rate, respectively.



To avoid index overflow when the AUV reaches the final segment, a virtual path point  $P_{n^p+1}^p$  is appended so that the extended segment  $P_{n^p}^p P_{n^p+1}^p$  maintains the same geometry as  $P_{n^p-1}^p P_{n^p}^p$ , with  $L_{n^p+1}^p = L_{n^p}^p$ ,  $\psi_{n^p+1}^p = \psi_{n^p}^p$ , and  $\theta_{n^p+1}^p = \theta_{n^p}^p$ .

### 3.1.2. Actions

The action vector  $\mathbf{a}_t = [\Delta\delta_t^\theta, \Delta\delta_t^\psi] \in [-1, 1]^2$  is defined as the normalized pitch and yaw control increments at each time step. These increments are mapped to X-rudder angles according to Equation (21). The X-rudder design enables all four rudders to jointly generate pitch and yaw moments, reducing the risk of single-rudder saturation and increasing the available equivalent moment under the rudder-angle limit.

$$\delta_t = \begin{bmatrix} \delta_{1,t} \\ \delta_{2,t} \\ \delta_{3,t} \\ \delta_{4,t} \end{bmatrix} = \Delta\delta_{\max} \begin{bmatrix} +1 & -1 \\ -1 & -1 \\ -1 & +1 \\ +1 & +1 \end{bmatrix} \begin{bmatrix} \Delta\delta_t^\theta \\ \Delta\delta_t^\psi \end{bmatrix} + \delta_{t-1}, |\delta| \leq \delta_{\max} \quad (21)$$

where  $\Delta\delta_{\max}$  and  $\delta_{\max}$  denote the maximum rudder-angle change per time step and the maximum rudder angle, respectively.

### 3.1.3. Reward

The reward  $r_{RL}$  consists of two position-error penalties and a velocity penalty that prevents the AUV from moving opposite to the tangent vector  $\mathbf{t}^p$  of the desired path. In addition, the reward terms are normalized to reduce the influence of scale differences among components, preventing the learning process from being dominated by a single error term.

$$r_{RL} = -k_1^r \text{sat}(|e_{ie}^h/e_{\max}^h|) - k_2^r \text{sat}(|e_{ie}^v/e_{\max}^v|) + k_3^r \text{sat}(r_{ie}^v/r_{\max}^v) \quad (22)$$

$$r_{ie}^v = \min(0, [\dot{x}, \dot{y}, \dot{z}] \cdot \mathbf{t}^p) \quad (23)$$

$$\mathbf{t}^p = [\cos(\psi_{ie}^p) \cos(\theta_{ie}^p), \sin(\psi_{ie}^p) \cos(\theta_{ie}^p), -\sin(\theta_{ie}^p)] \quad (24)$$

where  $k_i^r$  ( $i = 1, 2, 3$ ) denotes the weighting coefficients of the reward terms, and  $e_{\max}^h$ ,  $e_{\max}^v$ , and  $r_{\max}^v$  denote the corresponding normalization factors.  $\text{sat}(x) = \min(1, \max(-1, x))$ .

### 3.1.4. Termination

The episode terminates when  $i^e = n^p + 1$ , indicating that the AUV has passed the endpoint of the final path segment, or when the time exceeds the maximum limit  $t_{\max}^t$ .

$$d_t = \begin{cases} \text{True}, i^e = n^p + 1 \\ \text{True}, t > t_{\max}^t \end{cases} \quad (25)$$

## 3.2. Task-Informed Inductive-Bias Conservative SAC Algorithm

### 3.2.1. Task-Informed Inductive-Bias Encoder

In this study, the input state vector consists of dynamic states, segment-specific path-following states, and task-level contextual states, which play distinct functional roles in decision making. Most DRL methods typically flatten and concatenate the state variables into a single unstructured vector and process them using a shared multilayer perceptron (MLP). This approach relies on implicit representation learning to recover the latent structure during training. When the state variables are high-dimensional and heterogeneous, interference among irrelevant or weakly relevant features may increase the burden of representation learning. As a result, sample efficiency and overall training quality may

be limited. To mitigate this limitation, a Task-Informed Inductive-Bias (TIB) encoder is proposed to provide structured and task-conditioned state representations.

The overall framework is illustrated in Figure 2. Specifically, the dynamic state vector is first embedded into a compact latent feature  $h_{\text{dyn}}$ . Two segment-specific branches with identical architectures but independent parameters then encode the path-following states associated with the current and next path segments, respectively. Conditioned on  $h_{\text{dyn}}$ , each branch extracts segment-specific features, which are fused into a single representation  $h_{\text{fuse}}$ .

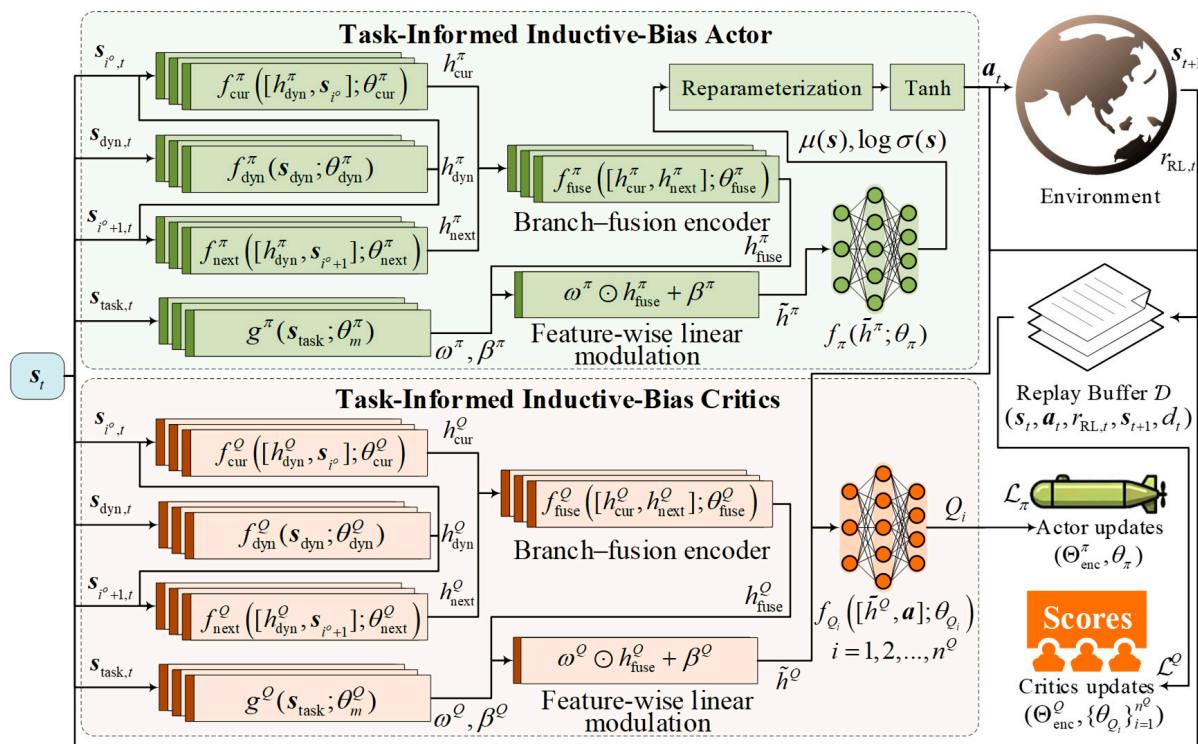
$$h_{\text{dyn}} = f_{\text{dyn}}(s_{\text{dyn}}; \theta_{\text{dyn}}) \quad (26)$$

$$h_{\text{cur}} = f_{\text{cur}}([h_{\text{dyn}}, s_{i^0}]; \theta_{\text{cur}}) \quad (27)$$

$$h_{\text{next}} = f_{\text{next}}([h_{\text{dyn}}, s_{i^0+1}]; \theta_{\text{next}}) \quad (28)$$

$$h_{\text{fuse}} = f_{\text{fuse}}([h_{\text{cur}}, h_{\text{next}}]; \theta_{\text{fuse}}) \quad (29)$$

where  $f_{\text{dyn}}(\cdot)$ ,  $f_{\text{cur}}(\cdot)$ ,  $f_{\text{next}}(\cdot)$ , and  $f_{\text{fuse}}(\cdot)$  are implemented as MLPs, and  $\theta_{\text{dyn}}$ ,  $\theta_{\text{cur}}$ ,  $\theta_{\text{next}}$ , and  $\theta_{\text{fuse}}$  denote their parameters.



**Figure 2.** The overall architecture of the TIB-CSAC algorithm.

To adapt encoder features to segment-specific path-following states, a feature-wise linear modulation (FiLM) network  $g(\cdot)$  generates scaling and shifting vectors  $(\omega, \beta)$  from the task-level state  $s_{\text{task}}$ . The fused feature  $h_{\text{fuse}}$  is then modulated element-wise.

$$[\omega, \beta] = g(s_{\text{task}}; \theta_m) \quad (30)$$

$$\tilde{h} = \omega \odot h_{\text{fuse}} + \beta \quad (31)$$

where  $\omega, \beta \in \mathbb{R}^{|h_{\text{fuse}}|}$ .  $g(\cdot)$  is implemented as an MLP and  $\theta_m$  denotes its parameters.  $\odot$  denotes element-wise multiplication.



In general, the overall TIB mapping can be written compactly as

$$\tilde{h} = F_{\text{enc}}(\mathbf{s}; \Theta_{\text{enc}}) \quad (32)$$

where  $\Theta_{\text{enc}} = \{\theta_{\text{dyn}}, \theta_{\text{cur}}, \theta_{\text{next}}, \theta_{\text{fuse}}, \theta_m\}$  denotes all encoder parameters.

Accordingly, the actor encoder produces a latent feature, which is then fed into the policy network to parameterize a squashed Gaussian policy. The action is subsequently sampled using the reparameterization trick.

$$\tilde{h}^{\pi} = F_{\text{enc}}(\mathbf{s}; \Theta_{\text{enc}}^{\pi}) \quad (33)$$

$$[\mu(\mathbf{s}), \log \sigma(\mathbf{s})] = f_{\pi}(\tilde{h}^{\pi}; \theta_{\pi}) \quad (34)$$

$$\mathbf{a} = \tanh(\mu(\mathbf{s}) + \sigma(\mathbf{s}) \odot \varepsilon) \quad (35)$$

where  $\Theta_{\text{enc}}^{\pi}$  and  $\theta_{\pi}$  denote the parameters of the actor encoder and policy network, respectively;  $\varepsilon \sim \mathcal{N}(0, I)$ , and  $\mu(\mathbf{s})$  and  $\sigma(\mathbf{s})$  represent the mean vector and standard deviation vector of the diagonal Gaussian distribution, respectively.

Similarly, the critics adopt the same TIB encoder architecture as the actor but use different parameters to avoid gradient interference between policy and critic updates.

$$\tilde{h}^Q = F_{\text{enc}}(\mathbf{s}; \Theta_{\text{enc}}^Q) \quad (36)$$

$$Q_i(\mathbf{s}, \mathbf{a}) = f_{Q_i}(\tilde{h}^Q, \mathbf{a}; \theta_{Q_i}) \quad (37)$$

where  $i = 1, 2, \dots, n^Q$ , and  $n^Q$  denotes the number of critic heads.  $\Theta_{\text{enc}}^Q$  and  $\theta_{Q_i}$  denote the parameters of the critic encoder and the critic heads, respectively.

### 3.2.2. Conservative Soft Actor–Critic Algorithm

During path-following training, task difficulty spans a broad range and exhibits a long-tail distribution. Conventional SAC is a risk-neutral optimization method that primarily maximizes the overall expected return and lacks an explicit mechanism for handling rare but challenging operating conditions. When sample coverage is insufficient in these difficult regimes, critic value learning for the corresponding state–action pairs relies more heavily on function approximation with sparse data support, resulting in higher estimation uncertainty and lower reliability. Combined with the actor’s tendency to favor actions with high Q-values, this can produce locally optimistic value estimates and thus limit path-following performance in challenging regions.

To address this issue, this paper introduces a multi-head conservative value-evaluation mechanism. Specifically,  $n^Q$  ( $n^Q \geq 2$ ) critic heads with independent parameters are built on the shared critic TIB encoder. For any state–action pair  $(\mathbf{s}, \mathbf{a})$ , the mean Q-value  $\mu_Q$  and the critic disagreement  $\sigma_Q$  are given by

$$\mu_Q(\mathbf{s}, \mathbf{a}) = \frac{1}{n^Q} \sum_{i=1}^{n^Q} Q_i(\mathbf{s}, \mathbf{a}) \quad (38)$$

$$\sigma_Q(\mathbf{s}, \mathbf{a}) = \sqrt{\frac{1}{n^Q} \sum_{i=1}^{n^Q} (Q_i(\mathbf{s}, \mathbf{a}) - \mu_Q(\mathbf{s}, \mathbf{a}))^2} \quad (39)$$

Policy improvement is guided by  $\mu_Q$ , which represents the overall Q-value level, and  $\sigma_Q$ , which characterizes estimation uncertainty. This design discourages the policy from favoring actions with high predicted Q-values but large critic disagreement.

$$Q_C(s, a) = \mu_Q(s, a) - \sqrt{\ln n^Q} \sigma_Q(s, a) \quad (40)$$

$$\mathcal{L}_\pi = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi(\cdot|s)} [\alpha \log \pi(a|s) - Q_C(s, a)] \quad (41)$$

where  $\sqrt{\ln n^Q}$  denotes the uncertainty-penalty coefficient. As  $n^Q$  increases,  $\sigma_Q$  can characterize value-estimation uncertainty more stably, and the penalty strength is therefore moderately increased. This growth is kept sublinear to avoid introducing an overly conservative bias that would impair policy-optimization efficiency.

The loss function for the entropy temperature coefficient  $\alpha$  is defined as

$$\mathcal{L}_\alpha = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\phi(\cdot|s)} [-\alpha (\log \pi(a|s) + \mathcal{H}_{\text{tar}})] \quad (42)$$

where  $\mathcal{D}$  denotes the replay buffer, and  $\mathcal{H}_{\text{tar}}$  is the target entropy.

To mitigate value overestimation and error propagation in critic bootstrapping, the temporal-difference (TD) target for the  $i$ -th critic head is calculated using the minimum of two randomly selected and distinct target critic heads. Let  $a_{t+1} \sim \pi(\cdot|s_{t+1})$ ; the TD target is

$$y_{t,i} = r_{RL,t} + \gamma(1 - d_t) \left( \min_{j=n_1, n_2} \bar{Q}_j(s_{t+1}, a_{t+1}) - \alpha \log \pi(a_{t+1} | s_{t+1}) \right) \quad (43)$$

where  $\bar{Q}_j$  denotes the Q-value estimated by the  $j$ -th target critic;  $n_1, n_2 \in \{1, 2, \dots, n^Q\}$  and  $n_1 \neq n_2$ ;  $\gamma$  is the discount factor; and  $d_t$  is the episode termination indicator.

To preserve critic diversity for uncertainty estimation, each critic head is updated using its own TD regression loss on an independently sampled mini-batch:

$$\mathcal{L}_i^Q = \mathbb{E}_{(s_t, a_t, r_{RL,t}, s_{t+1}, d_t) \sim \mathcal{B}_i, \mathcal{B}_i \sim \mathcal{D}} [(Q_i(s_t, a_t) - y_{t,i})^2] \quad (44)$$

where  $\mathcal{B}_i$  denotes the mini-batch independently sampled for the  $i$ -th value head from the replay buffer  $\mathcal{D}$ .

To ensure consistent state representations, the parameters of the TIB critic encoder are updated using the gradient of the mean TD loss:

$$\mathcal{L}_{\text{enc}}^Q = \frac{1}{n^Q} \sum_{i=1}^{n^Q} \mathcal{L}_i^Q \quad (45)$$

The target critic encoder and its heads are updated through soft updates:

$$\bar{\Theta}_{\text{enc}}^Q \leftarrow \tau \Theta_{\text{enc}}^Q + (1 - \tau) \bar{\Theta}_{\text{enc}}^Q \quad (46)$$

$$\bar{\theta}_{Q_i} \leftarrow \tau \theta_{Q_i} + (1 - \tau) \bar{\theta}_{Q_i} \quad (47)$$

The overall algorithm is summarized in Algorithm 1.

**Algorithm 1** Training procedure

---

Input: Critic size  $n^Q$ ; time interval  $\Delta t$ ; soft-update rate  $\tau$ ;  
discount factor  $\gamma$ ; batch size  $n^B$ ; termination flag  $d_t$ ;  
Initialize: actor  $\pi(a|s)$ ; critics  $\{Q_i\}_{i=1}^{n^Q}$ ; temperature  $\alpha$ ;  
target critics  $\{\bar{Q}_i\}_{i=1}^{n^Q}$ ; Replay buffer  $\mathcal{D}$ ;

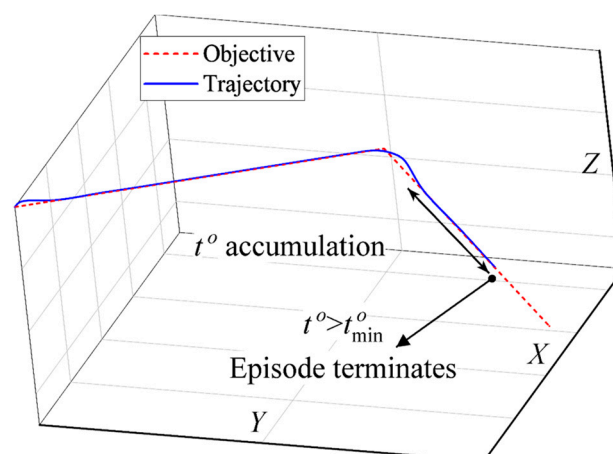
- 1: // Interaction
- 2: for each environment step do
- 3:   Get current observation  $s_t$ ;
- 4:   Sample action  $a_t \sim \pi_\phi(\cdot|s_t)$ ;
- 5:   Execute  $a_t$ , obtain  $r_{RL,t}$ ,  $s_{t+1}$ ,  $d_t$ ;
- 6:   If  $d_t = \text{true}$ , reset the environment;
- 7:   Store  $\{(s_t, a_t, r_{RL,t}, s_{t+1}, d_t)\}$  into  $\mathcal{D}$ ;
- 8: // Parameter updates
- 9:   if  $|\mathcal{D}| > n^{\text{bat}}$  then
- 10:     Sample independent mini-batch  $\{\mathcal{B}_i\}_{i=1}^{n^Q}$ ;
- 11:     Sample next action  $a_{t+1} \sim \pi_\phi(\cdot|s_{t+1})$  for  $(s_{t+1} \text{ in } \{\mathcal{B}_i\}_i^{n^Q})$ ;
- 12:     Compute TD target using Equation (43);
- 13:     Update critic using Equations (44) and (45);
- 14:     Update actor using Equation (41);
- 15:     Update  $\alpha$  using Equation (42);
- 16:     Soft update target critics using Equations (46) and (47);
- 17:   end if
- 18: end for

---

## 4. Experiments

### 4.1. Basic Settings

To facilitate path-following learning, each training episode consists of a two-segment reference path randomly initialized under the constraints in Equation (6). At the beginning of each episode, the initial AUV velocity is set to zero, and the desired surge speed  $u_d$  is randomly sampled from  $[0.5, 2.0]$  m/s. This setting covers the acceleration process from zero speed to the desired surge speed, thereby including the initial low-authority phase of the X-rudder in the training distribution. To preserve state continuity and ensure stable policy updates, the observation index  $i^o$  is kept fixed at its initial value for most of the episode. Therefore, an episode terminates when either the maximum time limit  $t_{\max}^t$  is exceeded or the stabilization-time criterion in Equation (13) is satisfied. The observation index  $i^o$  is updated only in the latter case, as shown in Figure 3.



**Figure 3.** Schematic diagram of the training process.

The hydrodynamic parameters are adopted from Ref. [25] and are provided in Table 1; the training task settings are summarized in Table 2. The bounded time-varying external disturbances acting on the AUV hydrodynamics are given in Equation (48). The sinusoidal and cosine terms approximate low-frequency force/moment disturbances with wave-like periodic characteristics.

$$\begin{cases} \tau_u^d(t) = u_d/u_{d,\max}(5\sin(k_1^d t + \varphi_1^d) + 5\cos(k_1^d t + \varphi_2^d) + 5\cos(k_1^d t + \varphi_3^d)) \\ \tau_v^d(t) = u_d/u_{d,\max}(0.5\sin(k_1^d t + \varphi_1^d) + \cos(k_1^d t + \varphi_2^d) + 0.1\cos(k_1^d t + \varphi_3^d)) \\ \tau_w^d(t) = u_d/u_{d,\max}(0.5\sin(k_1^d t + \varphi_1^d) + \cos(k_1^d t + \varphi_2^d) + 0.1\cos(k_1^d t + \varphi_3^d)) \\ \tau_q^d(t) = u_d/u_{d,\max}(\sin(k_2^d t + \varphi_1^d) + 2\cos(k_2^d t + \varphi_2^d) + 2\cos(k_1^d t + \varphi_3^d)) \\ \tau_r^d(t) = u_d/u_{d,\max}(\sin(k_2^d t + \varphi_1^d) + \cos(k_2^d t + \varphi_2^d) + 2\cos(k_1^d t + \varphi_3^d)) \end{cases} \quad (48)$$

where  $k_1^d$  and  $k_2^d$  denote the disturbance angular frequencies, while  $\varphi_1^d$ ,  $\varphi_2^d$ , and  $\varphi_3^d$  denote the disturbance phase shifts, which are randomly sampled at the beginning of each episode within the predefined ranges in Table 2. In the subsequent comparative tests, these parameters are fixed to  $k_1^d = 0.1$ ,  $k_2^d = 0.2$ ,  $\varphi_1^d = 0$ ,  $\varphi_2^d = \pi/4$ , and  $\varphi_3^d = \pi/6$ .

**Table 1.** Hydrodynamic parameters.

| Parameter           | Value                                                                                                                          | Parameter       | Value                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------|-----------------|----------------------------------------------|
| Inertia terms       | $(m_{11}, m_{22}, m_{33}) = (215, 265, 265) \text{ kg}$<br>$(m_{55}, m_{66}) = 80 \text{ kg} \cdot \text{m}^2$                 | Rudder rate     | $\Delta\delta_{\max} = 0.3491 \text{ rad/s}$ |
| Damping coefficient | $(X_u, Y_v, Z_w) = (70, 100, 100) \text{ kg} \cdot \text{m}^2$<br>$(M_q, N_r) = 50 \text{ kg} \cdot \text{m}^2/\text{s}$       | Rudder limit    | $\delta_{\max} = 0.4189 \text{ rad}$         |
| Damping coefficient | $(X_{u u }, Y_{v v }, Z_{w w }) = 100 \cdot (1, 2, 2) \text{ kg/m}$<br>$(M_{q q }, N_{r r }) = 50 \text{ kg} \cdot \text{m}^2$ | AUV weight      | $W = 1813 \text{ N}$                         |
| Rudder coefficients | $M_\delta = 0.1595 \cdot [+1, -1, -1, +1]$<br>$N_\delta = 0.1595 \cdot [+1, -1, -1, +1]$                                       | Vertical offset | $\overline{BG_z} = 0.02 \text{ m}$           |

**Table 2.** Training task settings.

| Parameter                         | Value                                            | Parameter                | Value                                      |
|-----------------------------------|--------------------------------------------------|--------------------------|--------------------------------------------|
| Initial AUV states                | $\eta_0 = 0, v_0 = 0$                            | Path segments            | $n^p = 2$                                  |
| Maximum adjacent heading change   | $\Delta\psi_{\max}^p = 1.5707 \text{ rad}$       | Time step                | $\Delta t = 0.2 \text{ s}$                 |
| Maximum absolute path pitch angle | $\Delta\theta_{\max}^p = 0.1222 \text{ rad}$     | Time threshold           | $t_{\max}^o = 10 \text{ s}$                |
| Minimum segment length            | $L_{\min} = 100 \text{ m}$                       | Error threshold          | $d_{\min}^h = 0.1 \text{ m}$               |
| Normalization factors             | $(e_{\max}^h, e_{\max}^v, r_{\max}) = (4, 2, 4)$ | Reward weights           | $k_i^r = 5 \text{ (} i = 1, 2, 3 \text{)}$ |
| Maximum task time limit           | $t_{\max}^t = 1.3\text{sum}(L^p/\hat{u})$        | Disturbance phase shifts | $ \varphi_1^d  < \pi/8$                    |
| Disturbance frequencies           | $0.08 < k_1^d < 0.12$                            | Disturbance phase shifts | $\pi/8 <  \varphi_2^d  < 3\pi/8$           |
| Disturbance frequencies           | $0.18 < k_2^d < 0.22$                            | Disturbance phase shifts | $\pi/12 <  \varphi_3^d  < \pi/4$           |

In this study, the AUV speed is regulated by an incremental S-plane controller, with the thruster parameters and control law provided in Appendix A. X-rudder path-following control is realized using three off-policy DRL methods: SAC [26], DDPG [27], and the proposed method. SAC is selected as the main baseline to directly evaluate the contributions of the proposed modules within the SAC framework. DDPG is included as another representative off-policy continuous-control algorithm, providing a deterministic-policy comparison. A conventional hierarchical model-free benchmark is also included, combining

LOS guidance [18], an S-plane controller, and pseudoinverse-based control allocation. The algorithm settings are summarized in Table 3. Shared or comparable hyperparameters and network scales are kept consistent wherever applicable so that the comparison is conducted under unified algorithmic configurations and comparable model capacities.

**Table 3.** Algorithm settings.

| Parameter               | Value                            | Parameter           | Value                           |
|-------------------------|----------------------------------|---------------------|---------------------------------|
| Learning rate           | $\text{lr} = 2 \times 10^{-4}$   | Dynamic encoder     | MLP( $2 \times 32$ ), ReLU      |
| Discount factor         | $\gamma = 0.99$                  | Segment branches    | MLP( $2 \times 64$ ), ReLU      |
| Critic number           | $n^Q = 6$                        | Fusion layer        | MLP( $1 \times 64$ ), ReLU      |
| Soft update coefficient | $\tau = 1 \times 10^{-3}$        | FiLM modulation     | g:MLP( $1 \times 64$ ), ReLU    |
| Training episodes       | $n^{\text{eps}} = 1200$          | Policy network      | MLP( $2 \times 128$ ), ReLU     |
| Replay Buffer size      | $n^{\text{buf}} = 1 \times 10^6$ | Critic head network | MLP( $2 \times 128$ ), ReLU     |
| Batch size              | $n^{\text{bat}} = 256$           | Target entropy      | $\mathcal{H}_{\text{tar}} = -2$ |

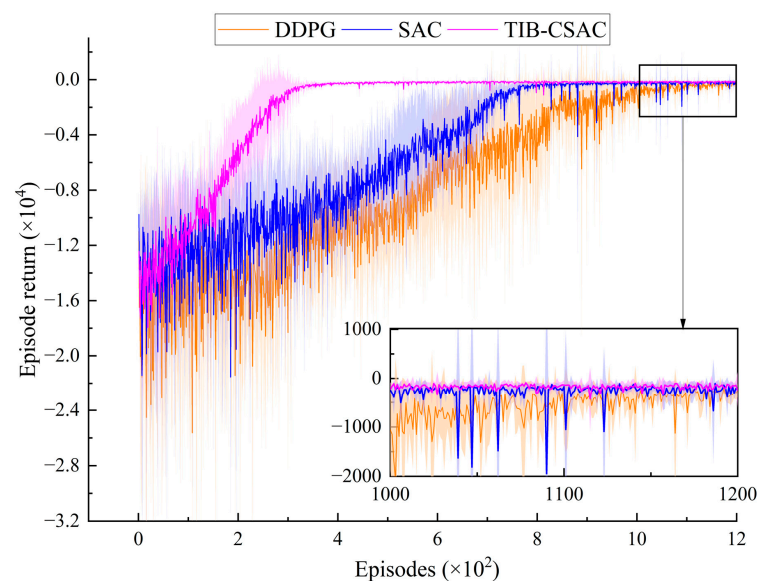
#### 4.2. Training Results

To mitigate stochastic effects and ensure reliable comparisons, each method was trained independently for 1200 episodes using five different random seeds. The training results are shown in Figure 4, where the solid lines represent the episode-wise mean return  $\bar{R}_j$  across five runs, and the shaded regions indicate the corresponding episode-wise standard deviation  $\sigma_j$ . As illustrated in Figure 4, TIB-CSAC converges faster during training; the magnified view also shows superior late-stage stability, with no sharp declines in episode rewards as observed for DDPG and SAC.

$$\bar{R}_j = \frac{1}{K} \sum_{k=1}^K R_{j,k} \quad (49)$$

$$\sigma_j = \sqrt{\frac{1}{K-1} \sum_{k=1}^K (R_{j,k} - \bar{R}_j)^2} \quad (50)$$

where  $R_{j,k}$  denotes the episode return at episode  $j$  in the  $k$ -th run,  $k = 1, 2, \dots, K$ ,  $K = 5$ .



**Figure 4.** Episode return curves in the comparative study.

To quantify tail-stage performance, as defined in Equations (51)–(53), let  $\bar{R}_{\text{tail}}$  denote the mean episode return over the tail stage of training averaged across runs, and let  $\sigma_{\text{tail}}$  denote the average standard deviation of episode returns over the tail stage within each run. As detailed in Table 4, compared with baseline SAC and DDPG, the proposed method increases  $\bar{R}_{\text{tail}}$  from  $-2.7927 \times 10^2$  and  $-5.5372 \times 10^2$  to  $-1.7244 \times 10^2$  and reduces  $\sigma_{\text{tail}}$  from  $5.1210 \times 10^2$  and  $4.6465 \times 10^2$  to  $1.2368 \times 10^2$ , indicating improved tail-stage training performance and enhanced tail-stage stability.

$$\bar{R}_{\text{tail},k} = \frac{1}{N_{\text{tail}}} \sum_{j=n^{\text{eps}}-N_{\text{tail}}+1}^{n^{\text{eps}}} R_{j,k} \quad (51)$$

$$\bar{R}_{\text{tail}} = \frac{1}{K} \sum_{k=1}^K \bar{R}_{\text{tail},k} \quad (52)$$

$$\sigma_{\text{tail}} = \frac{1}{K} \sum_{k=1}^K \sqrt{\frac{1}{N_{\text{tail}}-1} \sum_{j=n^{\text{eps}}-N_{\text{tail}}+1}^{n^{\text{eps}}} (R_{j,k} - \bar{R}_{\text{tail},k})^2} \quad (53)$$

where  $N_{\text{tail}}$  denotes the number of episodes in the tail-stage training and is set to 200.

**Table 4.** Training metric results in the comparative study.

| Method   | $\bar{R}_{\text{tail}}$ | $\sigma_{\text{tail}}$ | $\overline{\text{TRE}}$ | $\overline{\text{AUC}}$ |
|----------|-------------------------|------------------------|-------------------------|-------------------------|
| TIB-CSAC | $-1.7244 \times 10^2$   | $1.2368 \times 10^2$   | $2.9560 \times 10^2$    | $-2.6398 \times 10^6$   |
| SAC      | $-2.7927 \times 10^2$   | $5.1210 \times 10^2$   | $7.3320 \times 10^2$    | $-6.5231 \times 10^6$   |
| DDPG     | $-5.5372 \times 10^2$   | $4.6465 \times 10^2$   | $9.3920 \times 10^2$    | $-9.2165 \times 10^6$   |

To quantify training efficiency, episode returns are first smoothed using a  $W$ -episode moving average. For the  $k$ -th run,  $\text{TRE}_k$  is defined as the earliest episode  $t$  such that the smoothed return  $\tilde{R}_{t,k} \geq R_{\text{th}}$  satisfies Equation (55), and the overall TRE is averaged over  $K$  runs according to Equation (56). Here,  $W = 50$  and  $R_{\text{th}} = -1 \times 10^3$ . Moreover, the average area under the training curve is used to quantify overall training quality, as defined in Equation (57).

$$\tilde{R}_{t,k} = \frac{1}{W} \sum_{j=1}^W R_{t-j,k}, t \geq W \quad (54)$$

$$\text{TRE}_k = \min \left\{ t \in \{W, \dots, n^{\text{eps}}\} \mid \tilde{R}_{t,k} \geq R_{\text{th}} \right\} \quad (55)$$

$$\overline{\text{TRE}} = \frac{1}{K} \sum_{k=1}^K \text{TRE}_k \quad (56)$$

$$\overline{\text{AUC}} = \frac{1}{K} \sum_{k=1}^K \int_1^{n^{\text{eps}}} R_{j,k} dj \quad (57)$$

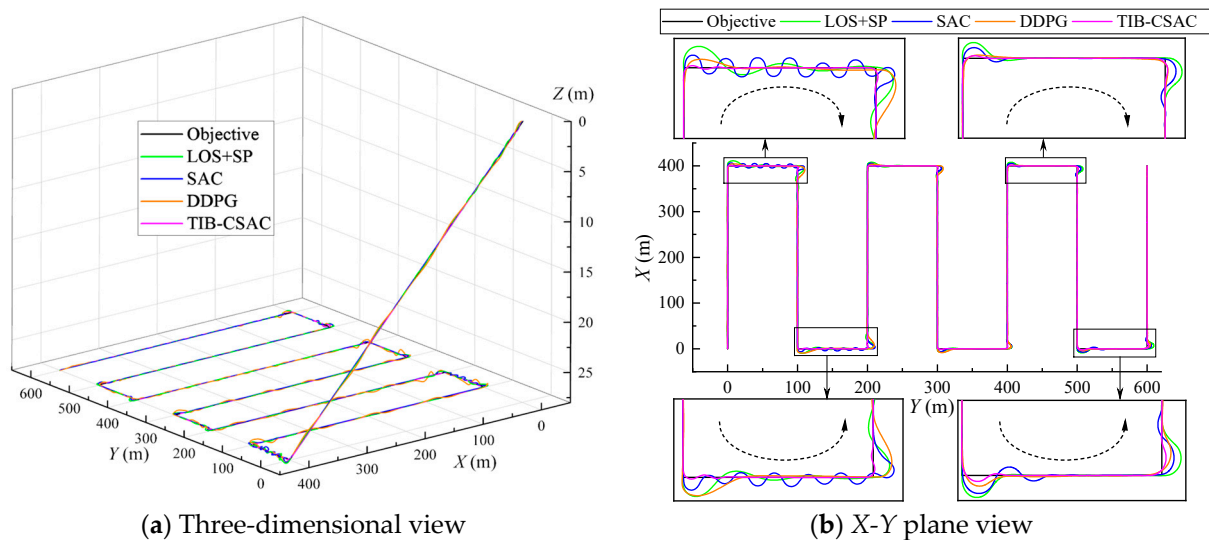
As detailed in Table 4, TIB-CSAC reaches the near-final performance threshold with fewer training episodes than SAC and DDPG:  $\overline{\text{TRE}}$  equal to  $2.9560 \times 10^2$  compared with  $7.3320 \times 10^2$  and  $9.3920 \times 10^2$ . It also attains a less negative cumulative return in  $\overline{\text{AUC}}$ , namely  $-2.6398 \times 10^6$  versus  $-6.5231 \times 10^6$  and  $-9.2165 \times 10^6$ , indicating improved sample efficiency and overall training quality.



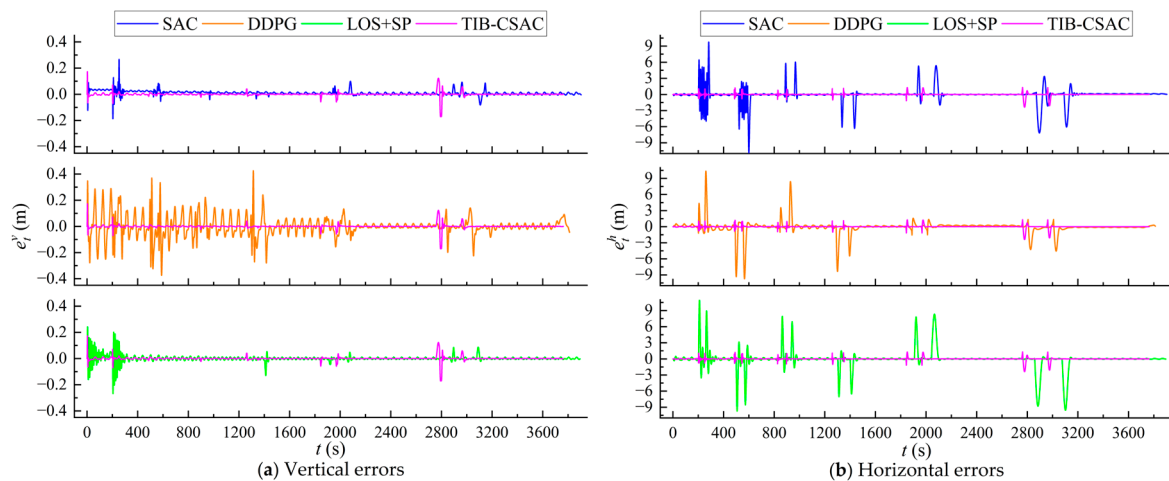
### 4.3. Path Following Results

#### 4.3.1. Case 1 Boustrophedon Path

A boustrophedon path is used to assess horizontal path-following performance during aggressive orthogonal turns under varying desired speeds  $u_d$ . In the horizontal plane, the path consists of 400 m survey lines connected by 100 m short transects. Moreover,  $u_d$  is updated after the AUV completes two consecutive turns and then travels 100 m along the path, with  $u_d$  taking values of 2.0, 1.7, 1.4, 1.1, 0.8, and 0.5 m/s. The detailed results are shown in Figures 5–8.



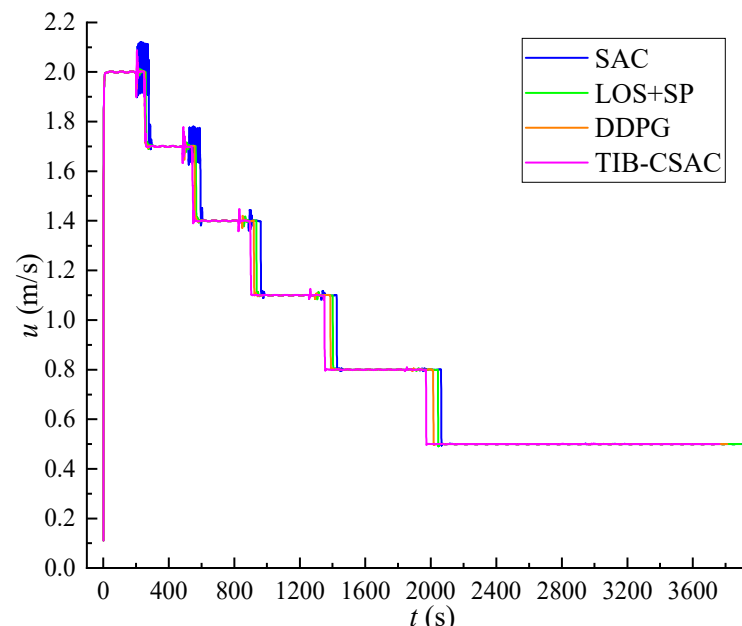
**Figure 5.** Trajectories in Case 1 (the dashed line with arrows indicates the path direction).



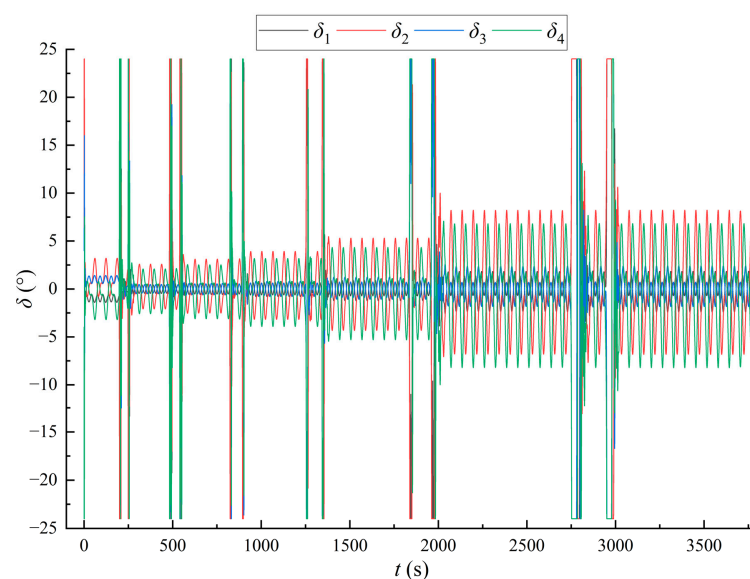
**Figure 6.** Path-following errors in Case 1.

As shown by the overall trajectories in Figure 5, all methods accomplish the prescribed boustrophedon path-following task. The zoomed-in views in Figure 5b further indicate that the main differences among these methods are concentrated in a few challenging regions, particularly around orthogonal turns and the subsequent transition phases. As shown in Figures 6 and 7, under different velocity conditions, TIB-CSAC achieves tighter adherence to the reference path, with overall position errors remaining within a relatively small range. During orthogonal-turn transitions, despite variations in rudder hydrodynamic effectiveness across velocities, TIB-CSAC consistently maintains lower horizontal overshoot and faster error decay. By contrast, although the baseline methods keep the horizontal error close to zero under most conditions, they tend to exhibit larger overshoot peaks

or oscillations and slower error decay around orthogonal turns. Overall, the results of Case 1 show that TIB-CSAC provides more robust horizontal path-following performance across a range of surge velocities, especially in the small subset of challenging orthogonal-turn conditions.



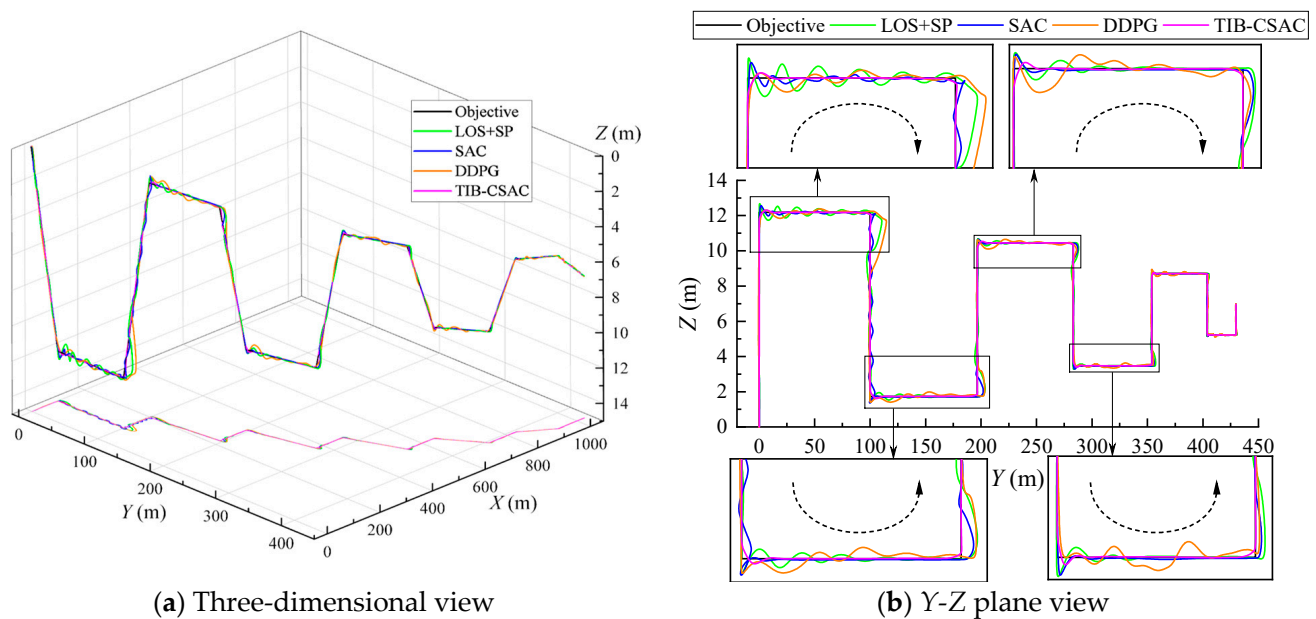
**Figure 7.** Surge velocities in Case 1.



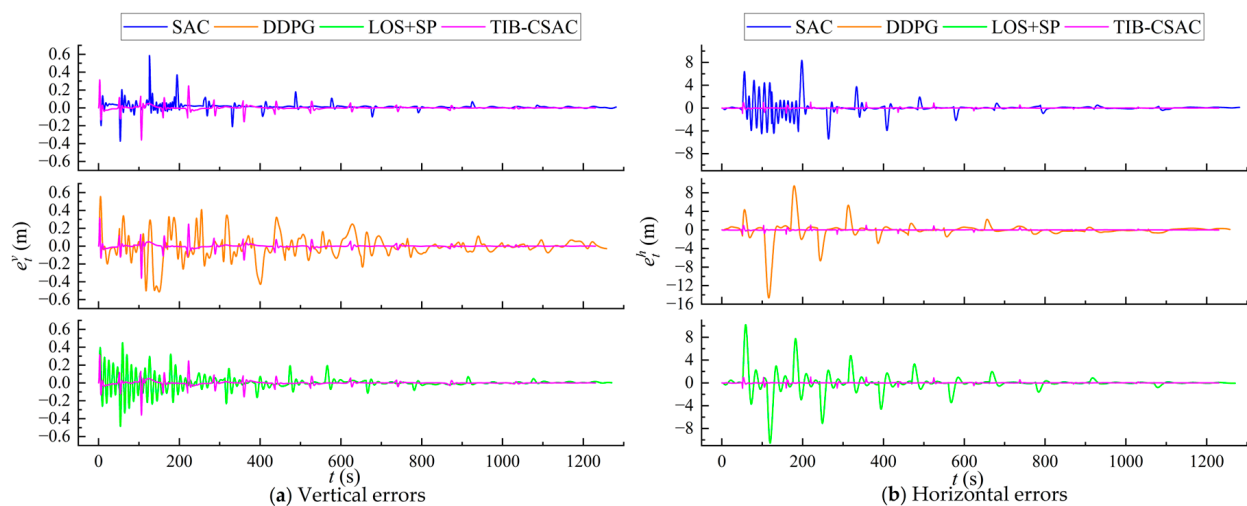
**Figure 8.** TIB-CSAC rudder angles in Case 1.

#### 4.3.2. Case 2 Trapezoidal Path

A 3D trapezoidal path is used to validate the transient path-following response under segment-wise simultaneous variations in the desired path heading, path pitch, and surge velocity. The desired surge velocity  $u_d$  is set sequentially to [2, 1.875, 1.75, 1.625, 1.5, 1.375, 1.25, 1.125, 1, 0.875, 0.75, 0.625, 0.5] m/s; the path pitch is set to [−0.1222, 0, 0.1047, 0, −0.0873, 0, 0.0698, 0, −0.0524, 0, 0.0349, 0, −0.0175] rad; and the path heading is set to [0, 1.5708, 0, 1.309, 0, 1.0472, 0, 0.7854, 0, 0.5236, 0, 0.2618, 0] rad. The length of each segment is 100 m. The detailed results are shown in Figures 9–12.

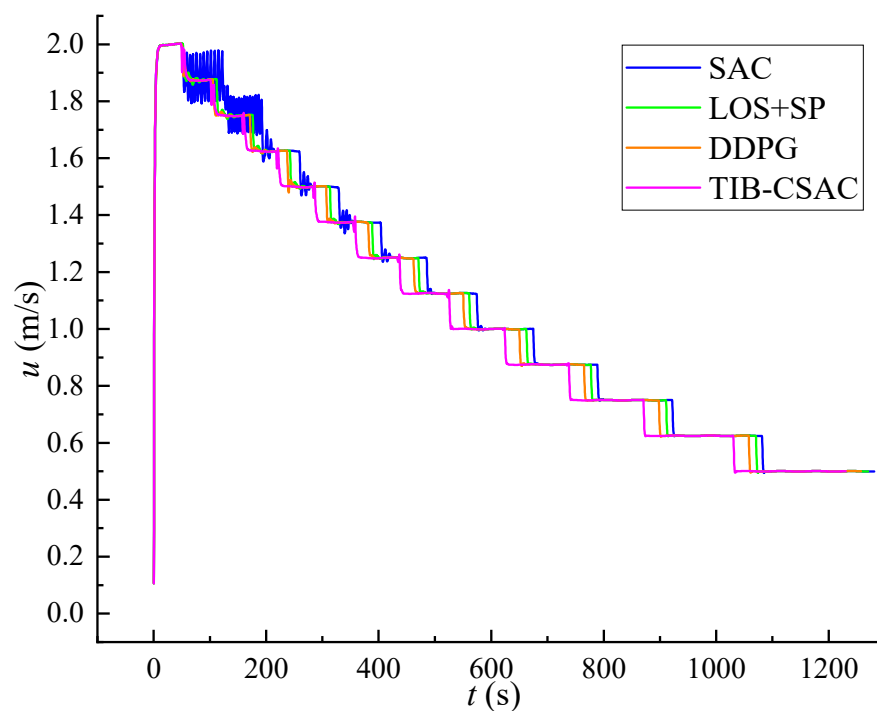


**Figure 9.** Trajectories in Case 2 (the dashed line with arrows indicates the path direction).

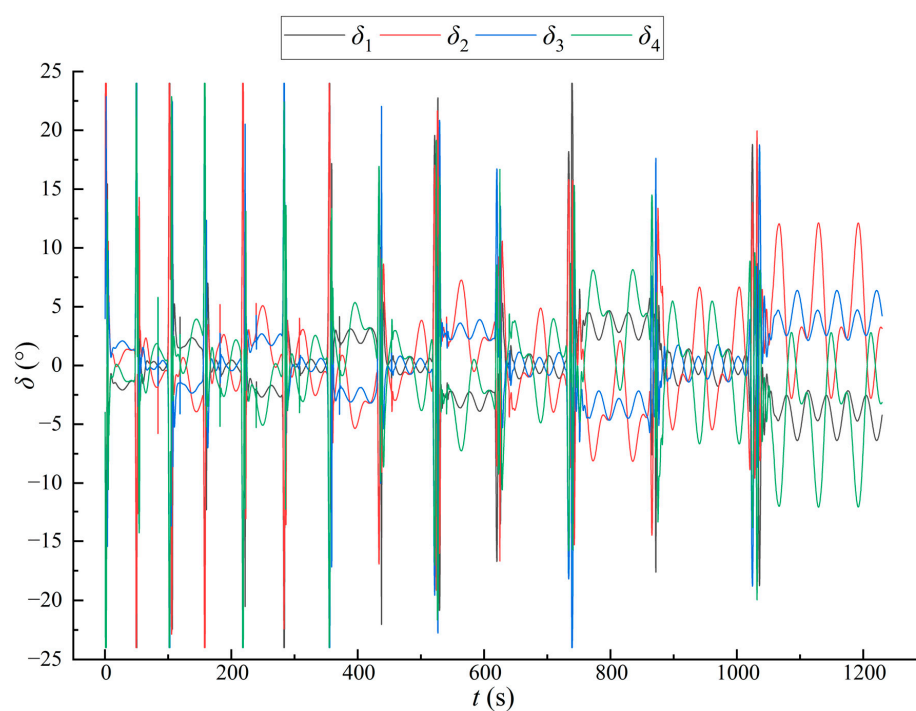


**Figure 10.** Path-following errors in Case 2.

As shown in Figure 9, all methods complete the 3D trapezoidal path-following task, and the main discrepancies arise around path-segment transitions. As indicated by Figures 10 and 11, TIB-CSAC yields smaller overall position errors and more stable transitions between path segments than the baseline methods under simultaneous variations in surge velocity and path geometry. In particular, under higher-velocity conditions, TIB-CSAC more effectively suppresses horizontal error peaks caused by large path-heading changes and mitigates persistent vertical-error oscillations associated with large path-pitch variations. As shown in Figure 12, the X-rudder angles of TIB-CSAC remain smooth and continuous except for brief pulse-like adjustments during path-segment transitions. Overall, Case 2 demonstrates that TIB-CSAC provides a more stable transient response under simultaneous variations in surge velocity and path geometry.



**Figure 11.** Surge velocity in Case 2.



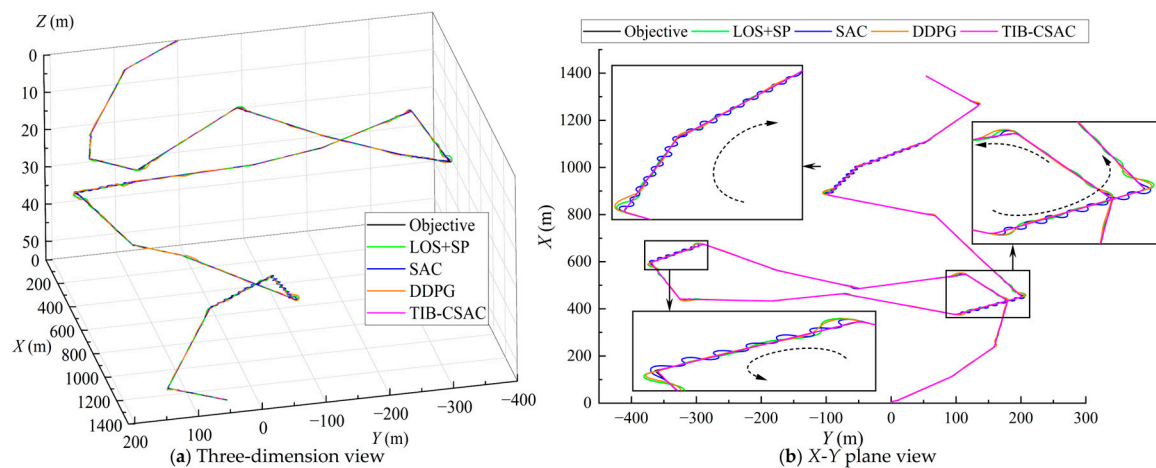
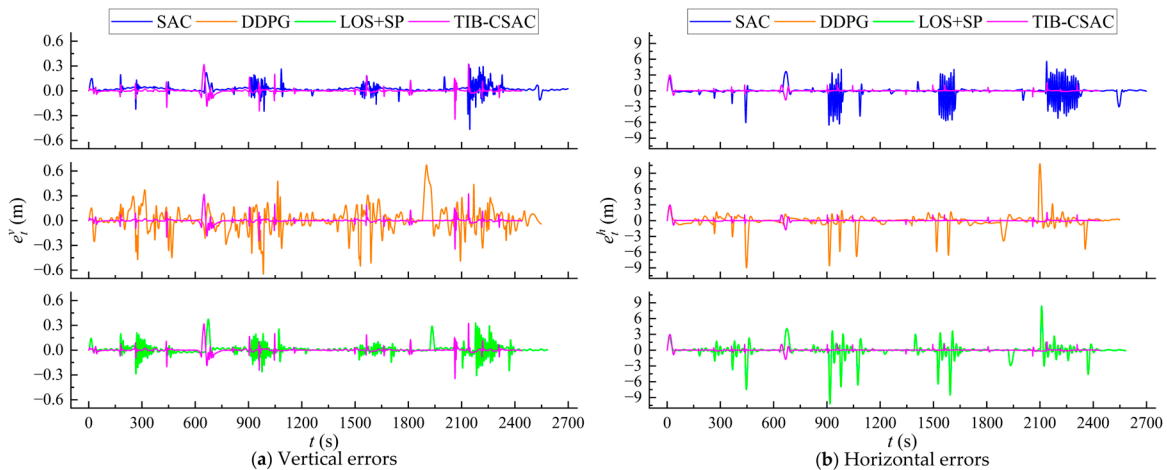
**Figure 12.** TIB-CSAC rudder angles in Case 2.

#### 4.3.3. Case 3 Random Path

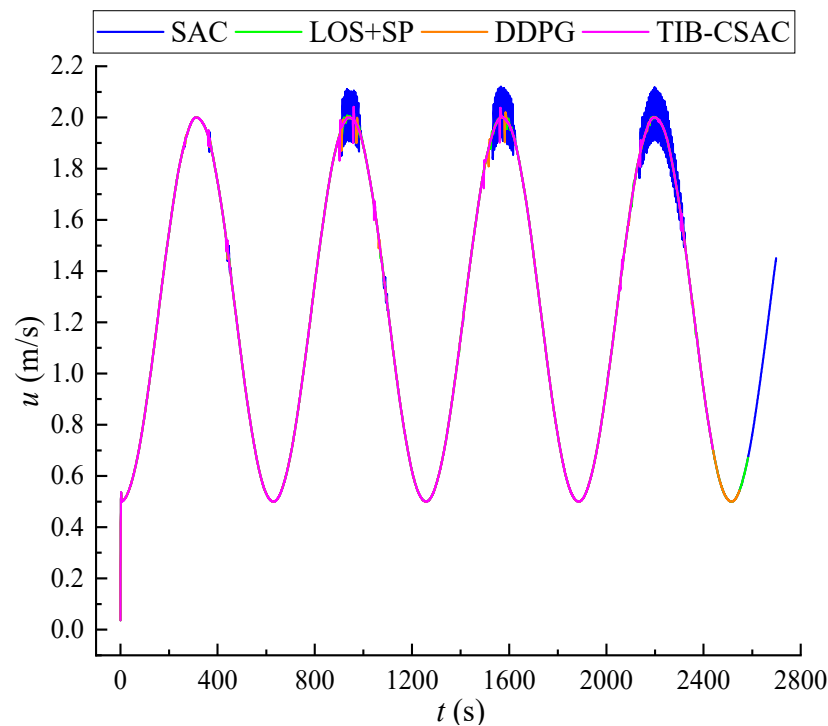
This section evaluates path-following performance on a 3D random path under time-varying surge-velocity conditions. Table 5 summarizes the waypoint configuration of the reference path, which consists of 20 waypoints in total, with the coordinate origin defined as the path-starting point. The velocity is set to  $u_d(t) = -0.75 \cos(0.01t) + 1.25$ . The results are shown in Figures 13–16.

**Table 5.** Waypoint Coordinates of Random Paths.

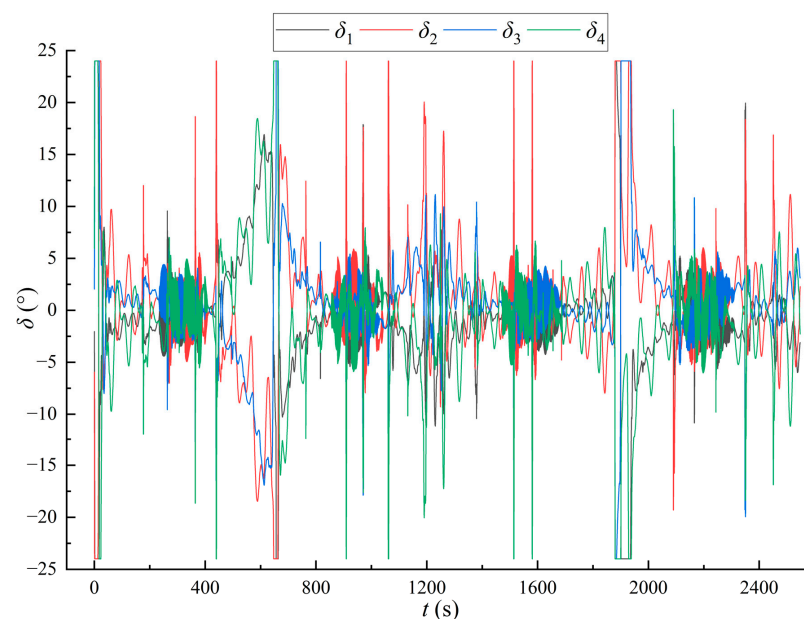
| Index       | 1      | 2      | 3      | 4      | 5      | 6       | 7       | 8       | 9       | 10      |
|-------------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|
| $x_i^p$ (m) | 111.79 | 238.38 | 434.02 | 546.22 | 486.21 | 562.53  | 674.69  | 596.73  | 441.31  | 433.10  |
| $y_i^p$ (m) | 92.60  | 158.75 | 179.64 | 115.61 | −47.99 | −174.84 | −288.56 | −370.22 | −325.50 | −185.18 |
| $z_i^p$ (m) | 2.56   | 14.46  | 14.68  | 15.56  | 3.68   | 11.19   | 15.01   | 20.85   | 11.28   | 18.72   |
| Index       | 11     | 12     | 13     | 14     | 15     | 16      | 17      | 18      | 19      | 20      |
| $x_i^p$ (m) | 464.11 | 375.32 | 450.57 | 628.69 | 794.98 | 883.33  | 1001.32 | 1110.58 | 1264.07 | 1387.24 |
| $y_i^p$ (m) | −74.54 | 99.49  | 201.61 | 131.19 | 68.23  | −97.93  | −52.65  | 54.26   | 135.41  | 54.28   |
| $z_i^p$ (m) | 20.23  | 24.20  | 22.87  | 32.63  | 31.45  | 43.51   | 32.77   | 35.95   | 51.05   | 51.89   |

**Figure 13.** Trajectories in Case 3 (the dashed line with arrows indicates the path direction).**Figure 14.** Path-following errors in Case 3.

As shown in Figures 13–15, the main differences in the 3D random path-following results are concentrated in challenging regions associated with higher surge velocities and large path-geometry variations. The baseline methods exhibit different error patterns: DDPG and LOS+SP show larger horizontal overshoot and more pronounced vertical-error oscillations in several regions, whereas SAC exhibits horizontal and vertical error fluctuations in certain intervals, accompanied by sustained oscillations in surge velocity. By contrast, under the combined effects of time-varying speed and random path-geometry variations, TIB-CSAC more effectively mitigates horizontal overshoot and vertical oscillations, demonstrating more robust path-following performance under diverse operating conditions.



**Figure 15.** Surge velocity in Case 3.



**Figure 16.** TIB-CSAC rudder angles in Case 3.

#### 4.3.4. Metric Analyses

Four metrics are used to quantitatively evaluate path-following performance: the maximum absolute error (MXAE), mean absolute error (MAE), conditional value at risk at 95% (CVaR<sub>95</sub>), and threshold-exceedance time ratio  $V_\varepsilon$ . CVaR<sub>95</sub> and  $V_\varepsilon$  are defined in Equations (58)–(60). Specifically, CVaR<sub>95</sub> denotes the conditional mean of the position-error magnitude beyond the 95th percentile of  $|e(t)|$ , while  $V_\varepsilon$  denotes the fraction of time for which  $|e(t)|$  exceeds the threshold  $\varepsilon = 0.1$  m. Detailed quantitative results are summarized in Tables 6 and 7.

$$\text{CVaR}_{95} = \mathbb{E} \left[ |e(t)| \mid |e(t)| \geq q_{95} \right] \quad (58)$$

$$q_{95} = \inf\{q : \Pr(|e(t)| \leq q) \geq 0.95\} \quad (59)$$

$$V_\varepsilon = \frac{1}{T} \int_0^T \chi(|e(t)| > \varepsilon) dt \quad (60)$$

where  $e(t)$  denotes the position error,  $q_{95}$  is the 95th percentile of  $|e(t)|$ ,  $\Pr(\cdot)$  denotes probability,  $\inf\{\cdot\}$  denotes the infimum,  $\chi(\cdot)$  is the indicator function, and  $T$  denotes the task duration.

**Table 6.** Vertical error metrics.

| Case | Metric                 | TIB-CSAC                | LOS+SP                  | SAC                     | DDPG                    |
|------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| C1   | MXAE (m)               | $1.7299 \times 10^{-1}$ | $2.6719 \times 10^{-1}$ | $2.6679 \times 10^{-1}$ | $4.2489 \times 10^{-1}$ |
|      | MAE (m)                | $4.1841 \times 10^{-3}$ | $1.0881 \times 10^{-2}$ | $1.3561 \times 10^{-2}$ | $4.7823 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m) | $4.7757 \times 10^{-2}$ | $8.6580 \times 10^{-2}$ | $6.3900 \times 10^{-2}$ | $2.4309 \times 10^{-1}$ |
|      | $V_\varepsilon$        | $7.4956 \times 10^{-3}$ | $1.4532 \times 10^{-2}$ | $3.0235 \times 10^{-3}$ | $1.4487 \times 10^{-1}$ |
| C2   | MXAE (m)               | $3.6035 \times 10^{-1}$ | $4.8607 \times 10^{-1}$ | $5.8540 \times 10^{-1}$ | $5.5639 \times 10^{-1}$ |
|      | MAE (m)                | $1.0904 \times 10^{-2}$ | $3.8505 \times 10^{-2}$ | $2.1844 \times 10^{-2}$ | $8.1225 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m) | $1.0198 \times 10^{-1}$ | $2.5271 \times 10^{-1}$ | $1.5847 \times 10^{-1}$ | $4.0743 \times 10^{-1}$ |
|      | $V_\varepsilon$        | $1.7398 \times 10^{-2}$ | $1.1855 \times 10^{-1}$ | $3.2938 \times 10^{-2}$ | $2.5016 \times 10^{-1}$ |
| C3   | MXAE (m)               | $3.4549 \times 10^{-1}$ | $3.7410 \times 10^{-1}$ | $4.6598 \times 10^{-1}$ | $6.7275 \times 10^{-1}$ |
|      | MAE (m)                | $1.2928 \times 10^{-2}$ | $3.3489 \times 10^{-2}$ | $3.1507 \times 10^{-2}$ | $9.6232 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m) | $1.2817 \times 10^{-1}$ | $2.1243 \times 10^{-1}$ | $1.6925 \times 10^{-1}$ | $4.4115 \times 10^{-1}$ |
|      | $V_\varepsilon$        | $2.5180 \times 10^{-2}$ | $8.2075 \times 10^{-2}$ | $6.1138 \times 10^{-2}$ | $3.3564 \times 10^{-1}$ |

**Table 7.** Horizontal error metrics.

| Case | Metric                 | TIB-CSAC                | LOS+SP                  | SAC                     | DDPG                    |
|------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| C1   | MXAE (m)               | $2.3813 \times 10^0$    | $1.0942 \times 10^1$    | $1.0892 \times 10^1$    | $1.0348 \times 10^1$    |
|      | MAE (m)                | $6.1981 \times 10^{-2}$ | $5.9013 \times 10^{-1}$ | $4.7040 \times 10^{-1}$ | $4.8171 \times 10^{-1}$ |
|      | CVaR <sub>95</sub> (m) | $8.4313 \times 10^{-1}$ | $7.1942 \times 10^0$    | $5.3551 \times 10^0$    | $5.1160 \times 10^0$    |
|      | $V_\varepsilon$        | $6.8683 \times 10^{-2}$ | $3.7126 \times 10^{-1}$ | $3.8413 \times 10^{-1}$ | $7.3974 \times 10^{-1}$ |
| C2   | MXAE (m)               | $1.0324 \times 10^0$    | $1.0575 \times 10^1$    | $8.3592 \times 10^0$    | $1.4654 \times 10^1$    |
|      | MAE (m)                | $3.6557 \times 10^{-2}$ | $7.0307 \times 10^{-1}$ | $4.6165 \times 10^{-1}$ | $7.6099 \times 10^{-1}$ |
|      | CVaR <sub>95</sub> (m) | $3.8266 \times 10^{-1}$ | $6.1709 \times 10^0$    | $4.2428 \times 10^0$    | $7.2433 \times 10^0$    |
|      | $V_\varepsilon$        | $6.6016 \times 10^{-2}$ | $5.7557 \times 10^{-1}$ | $4.9173 \times 10^{-1}$ | $7.9612 \times 10^{-1}$ |
| C3   | MXAE (m)               | $2.9708 \times 10^0$    | $1.0196 \times 10^1$    | $6.5206 \times 10^0$    | $1.0790 \times 10^1$    |
|      | MAE (m)                | $8.7358 \times 10^{-2}$ | $5.9247 \times 10^{-1}$ | $5.1517 \times 10^{-1}$ | $7.0218 \times 10^{-1}$ |
|      | CVaR <sub>95</sub> (m) | $1.0650 \times 10^0$    | $4.9855 \times 10^0$    | $4.1604 \times 10^0$    | $5.4241 \times 10^0$    |
|      | $V_\varepsilon$        | $1.4977 \times 10^{-1}$ | $5.8885 \times 10^{-1}$ | $4.6221 \times 10^{-1}$ | $8.4074 \times 10^{-1}$ |

Averaged over the three cases, TIB-CSAC yields lower overall error metrics than baseline SAC. In terms of MXAE, the average vertical and horizontal MXAE values of TIB-CSAC are reduced by  $1.4645 \times 10^{-1}$  m and 6.4623 m, respectively, indicating a stronger ability to suppress extreme errors. For MAE, the corresponding reductions are  $1.2965 \times 10^{-2}$  m and  $4.2044 \times 10^{-1}$  m, indicating closer overall adherence to the reference path. For CVaR<sub>95</sub>, the corresponding reductions are  $3.7903 \times 10^{-2}$  m and 3.8225 m, indicating more effective suppression of tail errors under difficult operating conditions. For  $V_\varepsilon$ , the corresponding reductions are  $1.5675 \times 10^{-2}$  and  $3.5120 \times 10^{-1}$ , indicating a lower proportion of time with threshold-exceeding errors and better maintenance of small-error path-following.

Overall, TIB-CSAC exhibits reliable path-following performance under diverse surge-velocity and path-geometry conditions, with clear advantages in overall path adherence, error-peak suppression, tail-error reduction, and persistence within the prescribed error threshold.

Several supplementary metrics are also introduced to quantitatively evaluate X-rudder control behavior and task-level effort. These metrics include the rudder saturation ratio



$R_{\delta}^{sat}$ , second-order rudder-angle difference index  $J_{\Delta^2\delta}$ , rudder-rate smoothness index  $J_{\dot{\delta}}$ , speed-weighted rudder-effort proxy  $J_{\delta,u}$ , and AUV energy-consumption proxy  $J_{AUV}$ , as defined in Equations (61)–(65). Here,  $N$  is the number of time steps. The quantitative results are summarized in Table 8.

$$R_{\delta}^{sat} = \frac{1}{4N} \sum_{i=1}^4 \sum_{k=1}^N \chi(|\delta_i(k)| \geq 0.95\delta_{\max}) \quad (61)$$

$$J_{\Delta^2\delta} = \frac{1}{4(N-2)} \sum_{i=1}^4 \sum_{k=3}^N (\delta_i(k) - 2\delta_i(k-1) + \delta_i(k-2))^2 \quad (62)$$

$$J_{\dot{\delta}} = \frac{1}{4(N-1)} \sum_{i=1}^4 \sum_{k=2}^N ((\delta_i(k) - \delta_i(k-1)) / \Delta t)^2 \quad (63)$$

$$J_{\delta,u} = \frac{1}{4N} \sum_{i=1}^4 \sum_{k=1}^N (u(k)\delta_i(k))^2 \quad (64)$$

$$J_{AUV} = \Delta t \sum_{k=1}^N u^3(k) \quad (65)$$

**Table 8.** Control-related metrics.

| Case | Metric               | TIB-CSAC                | LOS+SP                  | SAC                     | DDPG                    |
|------|----------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| C1   | $R_{\delta}^{sat}$   | $3.6415 \times 10^{-2}$ | $3.5432 \times 10^{-2}$ | $7.1948 \times 10^{-2}$ | $1.9965 \times 10^{-2}$ |
|      | $J_{\Delta^2\delta}$ | $5.9789 \times 10^{-2}$ | $7.9926 \times 10^{-1}$ | $2.0554 \times 10^{-2}$ | $6.3354 \times 10^0$    |
|      | $J_{\dot{\delta}}$   | $7.5725 \times 10^0$    | $1.0895 \times 10^1$    | $7.7465 \times 10^0$    | $4.2052 \times 10^1$    |
|      | $J_{\delta,u}$       | $2.7156 \times 10^1$    | $1.8364 \times 10^1$    | $7.7713 \times 10^1$    | $1.4899 \times 10^1$    |
|      | $J_{AUV}$            | $5.5553 \times 10^3$    | $5.7613 \times 10^3$    | $5.9402 \times 10^3$    | $5.6670 \times 10^3$    |
|      | $T$                  | $3.7622 \times 10^3$    | $3.8948 \times 10^3$    | $3.9028 \times 10^3$    | $3.8116 \times 10^3$    |
| C2   | $R_{\delta}^{sat}$   | $6.0163 \times 10^{-3}$ | $1.0941 \times 10^{-2}$ | $5.5885 \times 10^{-2}$ | $2.7831 \times 10^{-3}$ |
|      | $J_{\Delta^2\delta}$ | $3.1153 \times 10^{-1}$ | $2.3985 \times 10^0$    | $1.0248 \times 10^{-1}$ | $6.8936 \times 10^0$    |
|      | $J_{\dot{\delta}}$   | $1.8446 \times 10^1$    | $3.1841 \times 10^1$    | $2.1538 \times 10^1$    | $4.7438 \times 10^1$    |
|      | $J_{\delta,u}$       | $3.2003 \times 10^1$    | $2.6966 \times 10^1$    | $1.7262 \times 10^2$    | $1.9909 \times 10^1$    |
|      | $J_{AUV}$            | $2.2942 \times 10^3$    | $2.4531 \times 10^3$    | $2.5507 \times 10^3$    | $2.4031 \times 10^3$    |
|      | $T$                  | $1.2300 \times 10^3$    | $1.2704 \times 10^3$    | $1.2812 \times 10^3$    | $1.2576 \times 10^3$    |
| C3   | $R_{\delta}^{sat}$   | $4.1298 \times 10^{-2}$ | $2.0751 \times 10^{-2}$ | $9.3597 \times 10^{-2}$ | $2.6845 \times 10^{-2}$ |
|      | $J_{\Delta^2\delta}$ | $6.1293 \times 10^{-1}$ | $1.5771 \times 10^0$    | $1.1750 \times 10^{-1}$ | $1.3719 \times 10^1$    |
|      | $J_{\dot{\delta}}$   | $1.6996 \times 10^1$    | $1.9188 \times 10^1$    | $2.1262 \times 10^1$    | $8.9930 \times 10^1$    |
|      | $J_{\delta,u}$       | $3.4639 \times 10^1$    | $2.3822 \times 10^1$    | $2.0899 \times 10^2$    | $2.6849 \times 10^1$    |
|      | $J_{AUV}$            | $7.5449 \times 10^3$    | $7.5713 \times 10^3$    | $7.7279 \times 10^3$    | $7.5638 \times 10^3$    |
|      | $T$                  | $2.4384 \times 10^3$    | $2.5830 \times 10^3$    | $2.6988 \times 10^3$    | $2.5480 \times 10^3$    |

As shown in Table 8, TIB-CSAC maintains a relatively low rudder saturation ratio in all three cases, although LOS+SP or DDPG achieves lower  $R_{\delta}^{sat}$  values in some cases. For the second-order rudder-angle difference  $J_{\Delta^2\delta}$ , SAC generally yields smaller values, while TIB-CSAC remains lower than LOS+SP and DDPG. This indicates that the proposed method introduces necessary corrective rudder actions during path transitions without causing severe rudder-angle fluctuations. In terms of the rudder-rate smoothness index  $J_{\dot{\delta}}$ , TIB-CSAC achieves the lowest values in all three cases, suggesting smoother rudder-rate behavior. For the speed-weighted rudder-effort proxy  $J_{\delta,u}$ , TIB-CSAC is not always the lowest, but it is consistently lower than SAC. Moreover, TIB-CSAC achieves the lowest AUV energy-consumption proxy  $J_{AUV}$  and the shortest task-completion time  $T$  in all three cases. These results indicate that the improved path-following performance of TIB-CSAC is

not obtained through persistent rudder saturation or excessive task-level effort but through generally reasonable actuator behavior.

#### 4.4. Ablation Study

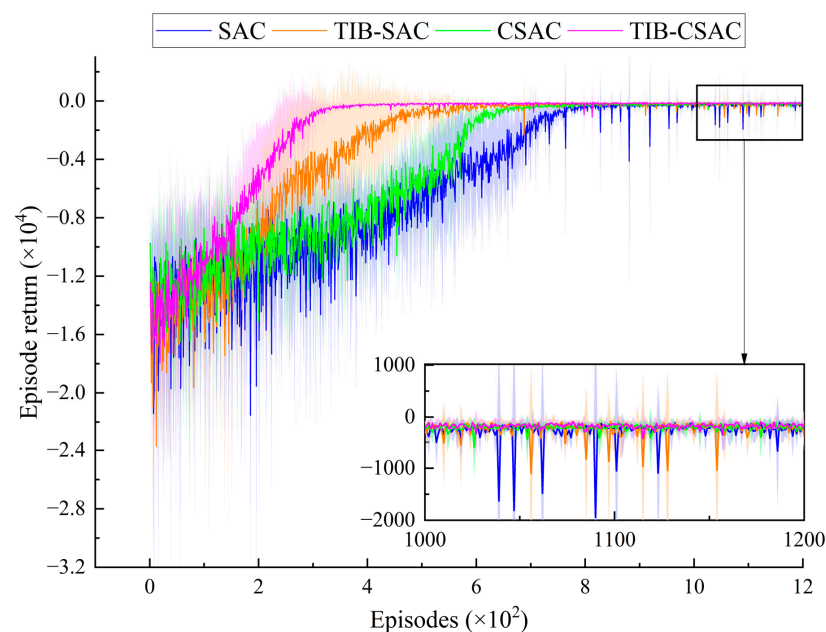
To further analyze the contribution of each module, this section conducts an ablation study on the full TIB-CSAC method and three comparison variants: the baseline SAC; CSAC, in which the TIB encoder is replaced by fully connected layers of comparable size; and TIB-SAC, which combines the TIB encoder with the baseline SAC. Each method is trained for 1200 episodes using five random seeds.

##### 4.4.1. Quantitative Analysis of the TIB Encoder

Table 9 and Figure 17 quantify the contribution of the TIB encoder through two controlled comparisons: CSAC versus TIB-CSAC and SAC versus TIB-SAC. Compared with TIB-CSAC, the  $\overline{\text{TRE}}$  of CSAC increases from  $2.9560 \times 10^2$  to  $6.4180 \times 10^2$ , corresponding to 2.17 times that of TIB-CSAC. Meanwhile, the  $\overline{\text{AUC}}$  decreases from  $-2.6398 \times 10^6$  to  $-5.4341 \times 10^6$ , corresponding to 2.06 times the absolute magnitude of TIB-CSAC. A consistent trend is also observed in the comparison between TIB-SAC and SAC. Specifically, the  $\overline{\text{TRE}}$  of TIB-SAC decreases from  $7.3320 \times 10^2$  to  $4.3060 \times 10^2$ , corresponding to 0.59 times that of SAC, while the  $\overline{\text{AUC}}$  increases from  $-6.5231 \times 10^6$  to  $-3.9544 \times 10^6$ , corresponding to 0.61 times the absolute magnitude of SAC. In addition, the tail-stage metrics show consistent but secondary improvements after introducing the TIB encoder, with  $\sigma_{\text{tail}}$  decreasing in both SAC versus TIB-SAC and CSAC versus TIB-CSAC. Overall, these results indicate that the structured and conditioned representations induced by the TIB encoder primarily accelerate policy formation and improve overall training quality.

**Table 9.** Training metric results in the ablation study.

| Method   | $\overline{\text{TRE}}$ | $\overline{\text{AUC}}$ | $\sigma_{\text{tail}}$ | $\overline{R}_{\text{tail}}$ |
|----------|-------------------------|-------------------------|------------------------|------------------------------|
| SAC      | $7.3320 \times 10^2$    | $-6.5231 \times 10^6$   | $5.1210 \times 10^2$   | $-2.7927 \times 10^2$        |
| CSAC     | $6.4180 \times 10^2$    | $-5.4341 \times 10^6$   | $1.3339 \times 10^2$   | $-2.0564 \times 10^2$        |
| TIB-SAC  | $4.3060 \times 10^2$    | $-3.9544 \times 10^6$   | $3.5529 \times 10^2$   | $-2.3032 \times 10^2$        |
| TIB-CSAC | $2.9560 \times 10^2$    | $-2.6398 \times 10^6$   | $1.2368 \times 10^2$   | $-1.7244 \times 10^2$        |



**Figure 17.** Episode return curves in the ablation study.

#### 4.4.2. Quantitative Analysis of the Conservative Value Evaluation Mechanism

The contribution of the conservative value-evaluation mechanism is quantified through two controlled comparisons: TIB-SAC versus TIB-CSAC and SAC versus CSAC. Relative to TIB-CSAC, the  $\sigma_{\text{tail}}$  of TIB-SAC increases from  $1.2368 \times 10^2$  to  $3.5529 \times 10^2$ , corresponding to 2.87 times that of TIB-CSAC. A similar effect is observed in the comparison between CSAC and SAC, where the  $\sigma_{\text{tail}}$  of CSAC decreases from  $5.1210 \times 10^2$  to  $1.3339 \times 10^2$ , corresponding to 0.26 times that of SAC. Collectively, these results indicate that the multi-head conservative value-evaluation mechanism effectively mitigates tail-stage return fluctuations and reduces return drawdowns induced by challenging tasks, thereby improving tail-stage training stability.

#### 4.4.3. Overall Ablation Discussion

Compared with baseline SAC, TIB-CSAC reduces  $\overline{\text{TRE}}$  and  $\sigma_{\text{tail}}$  to 0.40 and 0.24 times those of SAC, respectively, while yielding less negative  $\overline{\text{AUC}}$  and  $\overline{R}_{\text{tail}}$  values whose absolute magnitudes are reduced to 0.40 and 0.62 times those of SAC, respectively. These results indicate that combining the TIB encoder with the conservative value-evaluation mechanism enables TIB-CSAC to achieve faster convergence, higher overall training quality, and better tail-stage performance.

#### 4.5. Sensitivity Study

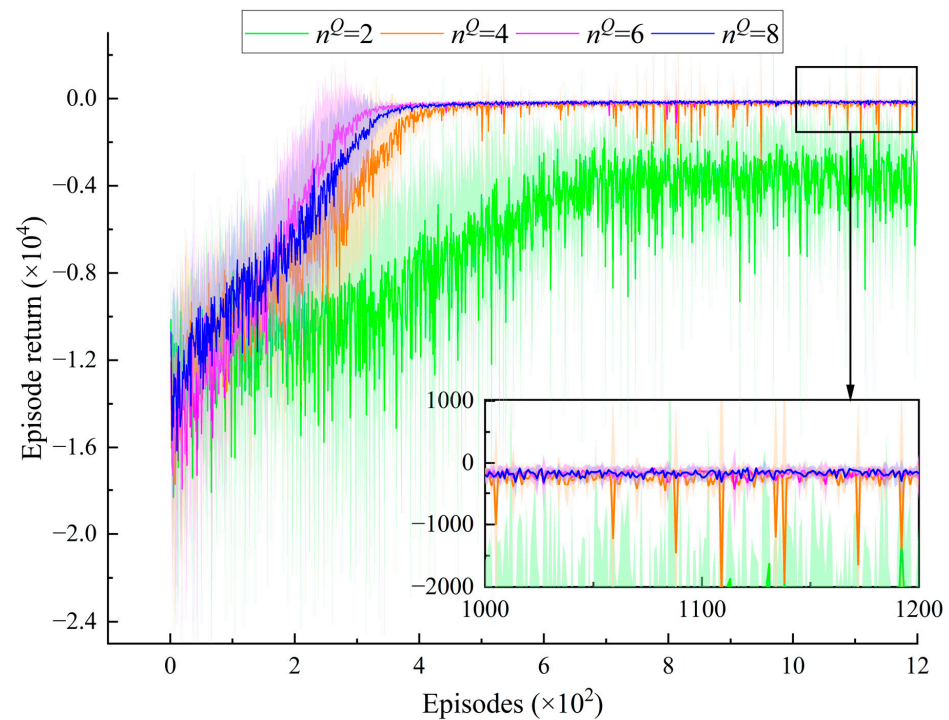
##### 4.5.1. Sensitivity Analysis on Critic Size

To further investigate the influence of a key hyperparameter on the performance of the proposed algorithm, this section conducts a sensitivity study on the critic size  $n^Q$  used in the multi-head conservative value evaluation mechanism. For each parameter setting, training is performed for 1200 episodes using five random seeds.

As shown in Figure 18 and Table 10, when  $n^Q = 2$ , the return curves remain in a low-return region with severe fluctuations, indicating that an overly small  $n^Q$  is insufficient to provide reliable conservative value evaluation for effective policy learning. With  $n^Q = 4$ , the return curves begin to enter the high-return region, but noticeable return drawdowns still persist after convergence, suggesting that value-evaluation robustness remains inadequate. Increasing  $n^Q$  from 4 to 6 effectively improves tail-stage stability, with  $\sigma_{\text{tail}}$  decreasing from  $5.3876 \times 10^2$  to  $1.2368 \times 10^2$ , corresponding to 0.23 times that of the  $n^Q = 4$  setting. By contrast, further increasing  $n^Q$  from 6 to 8 yields only a modest additional reduction in  $\sigma_{\text{tail}}$ , from  $1.2368 \times 10^2$  to  $1.0508 \times 10^2$ , corresponding to 0.85 times that of the  $n^Q = 6$  setting, while convergence becomes slower, with  $\overline{\text{TRE}}$  increasing from  $2.9560 \times 10^2$  to  $3.3480 \times 10^2$ , corresponding to 1.13 times that of the  $n^Q = 6$  setting. This trend suggests that an overly large  $n^Q$  may further suppress tail-stage volatility, but the associated increase in conservatism may also make policy improvement more cautious, thereby reducing sample efficiency. Therefore, under the algorithmic setting adopted in this study,  $n^Q = 6$  provides a favorable trade-off between convergence speed and tail-stage performance.

**Table 10.** Training metric results in the sensitivity study.

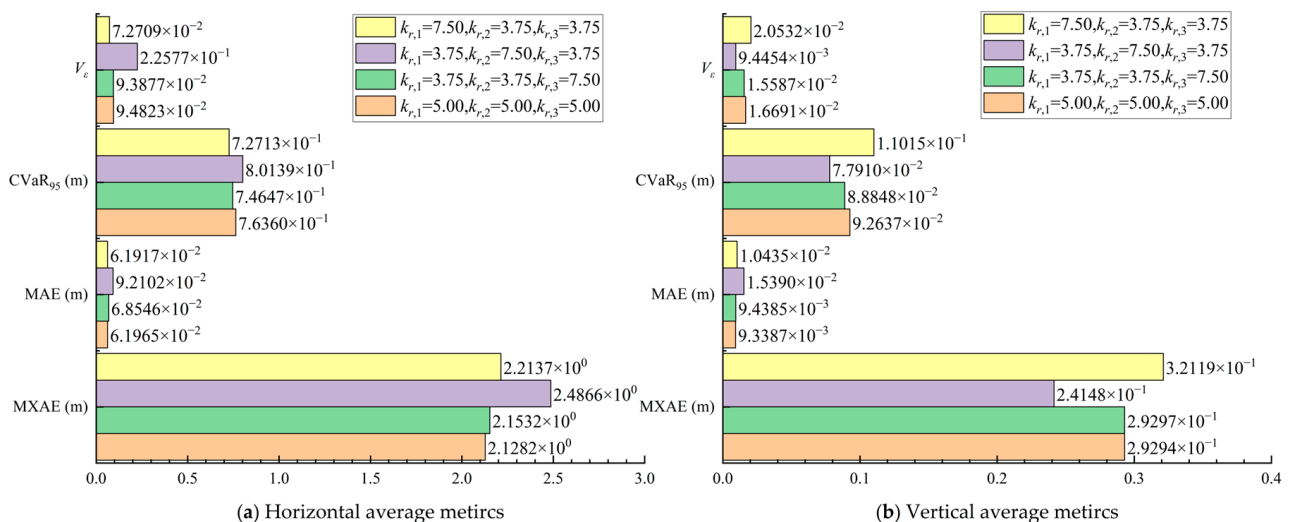
| $n^Q$ | $\sigma_{\text{tail}}$ | $\overline{\text{TRE}}$ | $\overline{\text{AUC}}$ | $\overline{R}_{\text{tail}}$ |
|-------|------------------------|-------------------------|-------------------------|------------------------------|
| 2     | $2.4269 \times 10^3$   | $1.2000 \times 10^3$    | $-7.9829 \times 10^6$   | $-3.7323 \times 10^3$        |
| 4     | $5.3876 \times 10^2$   | $3.8780 \times 10^2$    | $-3.2756 \times 10^6$   | $-2.7346 \times 10^2$        |
| 6     | $1.2368 \times 10^2$   | $2.9560 \times 10^2$    | $-2.6398 \times 10^6$   | $-1.7244 \times 10^2$        |
| 8     | $1.0508 \times 10^2$   | $3.3480 \times 10^2$    | $-2.6063 \times 10^6$   | $-1.6741 \times 10^2$        |



**Figure 18.** Episode return curves in the sensitivity study.

#### 4.5.2. Sensitivity Analysis on Reward Weights

To further evaluate the influence of reward weights on path-following performance, a sensitivity analysis was conducted for  $k_{r,1}$ ,  $k_{r,2}$ , and  $k_{r,3}$ . In addition to the default setting  $k_{r,1} = k_{r,2} = k_{r,3} = 5$ , three reward-weight combinations were tested in Cases 1–3:  $[k_{r,1}, k_{r,2}, k_{r,3}] = [7.50, 3.75, 3.75]$ ,  $[3.75, 7.50, 3.75]$ , and  $[3.75, 3.75, 7.50]$ . These settings increase the relative weights of the horizontal-error penalty, vertical-error penalty, and reverse-motion penalty, respectively. The averaged metric results over Cases 1–3 are shown in Figure 19, and the detailed metrics for each case are provided in Tables 11 and 12.



**Figure 19.** Average path-following metrics of Case 1–3.

**Table 11.** Horizontal error metrics for sensitivity analysis of reward weights.

| Case | $[k_{r,1}, k_{r,2}, k_{r,3}]$ | [7.50, 3.75, 3.75]      | [3.75, 7.50, 3.75]      | [3.75, 3.75, 7.50]      | [5.0, 5.0, 5.0]         |
|------|-------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| C1   | MXAE (m)                      | $2.8895 \times 10^0$    | $3.2273 \times 10^0$    | $2.7073 \times 10^0$    | $2.3813 \times 10^0$    |
|      | MAE (m)                       | $9.3567 \times 10^{-2}$ | $1.2752 \times 10^{-1}$ | $8.1307 \times 10^{-2}$ | $6.1981 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m)        | $1.2507 \times 10^0$    | $1.1357 \times 10^0$    | $1.0947 \times 10^0$    | $8.4313 \times 10^{-1}$ |
|      | $V_\varepsilon$               | $8.7584 \times 10^{-2}$ | $3.7600 \times 10^{-1}$ | $7.3903 \times 10^{-2}$ | $6.8683 \times 10^{-2}$ |
| C2   | MXAE (m)                      | $1.1418 \times 10^0$    | $1.6357 \times 10^0$    | $1.1187 \times 10^0$    | $1.0324 \times 10^0$    |
|      | MAE (m)                       | $4.0079 \times 10^{-2}$ | $8.7472 \times 10^{-2}$ | $5.9850 \times 10^{-2}$ | $3.6557 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m)        | $4.0264 \times 10^{-1}$ | $7.0743 \times 10^{-1}$ | $5.4121 \times 10^{-1}$ | $3.8266 \times 10^{-1}$ |
|      | $V_\varepsilon$               | $6.0212 \times 10^{-2}$ | $2.0276 \times 10^{-1}$ | $1.1075 \times 10^{-1}$ | $6.6016 \times 10^{-2}$ |
| C3   | MXAE (m)                      | $2.6098 \times 10^0$    | $2.5967 \times 10^0$    | $2.6334 \times 10^0$    | $2.9708 \times 10^0$    |
|      | MAE (m)                       | $5.2106 \times 10^{-2}$ | $6.1310 \times 10^{-2}$ | $6.4482 \times 10^{-2}$ | $8.7358 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m)        | $5.2805 \times 10^{-1}$ | $5.6101 \times 10^{-1}$ | $6.0352 \times 10^{-1}$ | $1.0650 \times 10^0$    |
|      | $V_\varepsilon$               | $7.0330 \times 10^{-2}$ | $9.8560 \times 10^{-2}$ | $9.6979 \times 10^{-2}$ | $1.4977 \times 10^{-1}$ |

**Table 12.** Vertical error metrics for sensitivity analysis of reward weights.

| Case | $[k_{r,1}, k_{r,2}, k_{r,3}]$ | [7.50, 3.75, 3.75]      | [3.75, 7.50, 3.75]      | [3.75, 3.75, 7.50]      | [5.0, 5.0, 5.0]         |
|------|-------------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| C1   | MXAE (m)                      | $2.5174 \times 10^{-1}$ | $1.5607 \times 10^{-1}$ | $1.8283 \times 10^{-1}$ | $1.7299 \times 10^{-1}$ |
|      | MAE (m)                       | $1.0644 \times 10^{-2}$ | $1.0499 \times 10^{-2}$ | $6.4236 \times 10^{-3}$ | $4.1841 \times 10^{-3}$ |
|      | CVaR <sub>95</sub> (m)        | $1.1995 \times 10^{-1}$ | $5.1650 \times 10^{-2}$ | $7.6575 \times 10^{-2}$ | $4.7757 \times 10^{-2}$ |
|      | $V_\varepsilon$               | $2.4152 \times 10^{-2}$ | $4.2438 \times 10^{-4}$ | $1.6045 \times 10^{-2}$ | $7.4956 \times 10^{-3}$ |
| C2   | MXAE (m)                      | $3.8565 \times 10^{-1}$ | $3.0402 \times 10^{-1}$ | $3.7452 \times 10^{-1}$ | $3.6035 \times 10^{-1}$ |
|      | MAE (m)                       | $1.0350 \times 10^{-2}$ | $1.6619 \times 10^{-2}$ | $1.0264 \times 10^{-2}$ | $1.0904 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m)        | $1.1253 \times 10^{-1}$ | $9.9581 \times 10^{-2}$ | $9.9590 \times 10^{-2}$ | $1.0198 \times 10^{-1}$ |
|      | $V_\varepsilon$               | $2.0667 \times 10^{-2}$ | $1.5981 \times 10^{-2}$ | $1.5450 \times 10^{-2}$ | $1.7398 \times 10^{-2}$ |
| C3   | MXAE (m)                      | $3.2619 \times 10^{-1}$ | $2.6435 \times 10^{-1}$ | $3.2154 \times 10^{-1}$ | $3.4549 \times 10^{-1}$ |
|      | MAE (m)                       | $1.0312 \times 10^{-2}$ | $1.9053 \times 10^{-2}$ | $1.1628 \times 10^{-2}$ | $1.2928 \times 10^{-2}$ |
|      | CVaR <sub>95</sub> (m)        | $9.7986 \times 10^{-2}$ | $8.2500 \times 10^{-2}$ | $9.0378 \times 10^{-2}$ | $1.2817 \times 10^{-1}$ |
|      | $V_\varepsilon$               | $1.6776 \times 10^{-2}$ | $1.1930 \times 10^{-2}$ | $1.5267 \times 10^{-2}$ | $2.5180 \times 10^{-2}$ |

As shown in Figure 19, the relative allocation of reward weights affects the emphasis of path-following performance. Relative to the default setting, increasing the relative weight of  $k_{r,1}$  improves most horizontal error metrics, especially by reducing the horizontal threshold-exceedance ratio  $V_\varepsilon$  and horizontal tail error CVaR<sub>95</sub>. Increasing the relative weight of  $k_{r,2}$  reduces the vertical  $V_\varepsilon$ , CVaR<sub>95</sub>, and MXAE compared with the default setting, but it also increases the horizontal error metrics, indicating that emphasizing vertical-error suppression may weaken horizontal path-following performance. After C is increased, the horizontal and vertical error metrics remain generally close to those of the default setting. This occurs because the reverse-motion penalty mainly serves as an auxiliary constraint. When the AUV rarely tends to move opposite to the path direction, increasing  $k_{r,3}$  has a relatively limited effect on the final path-following error metrics.

Overall, although the setting  $k_{r,1} = k_{r,2} = k_{r,3} = 5$  does not achieve the best value for every individual metric, it provides a more balanced performance between horizontal and vertical path-following errors and avoids significant degradation in either error category. Therefore, this weight setting is adopted as the default reward-function configuration.

#### 4.6. Stochastic Disturbance

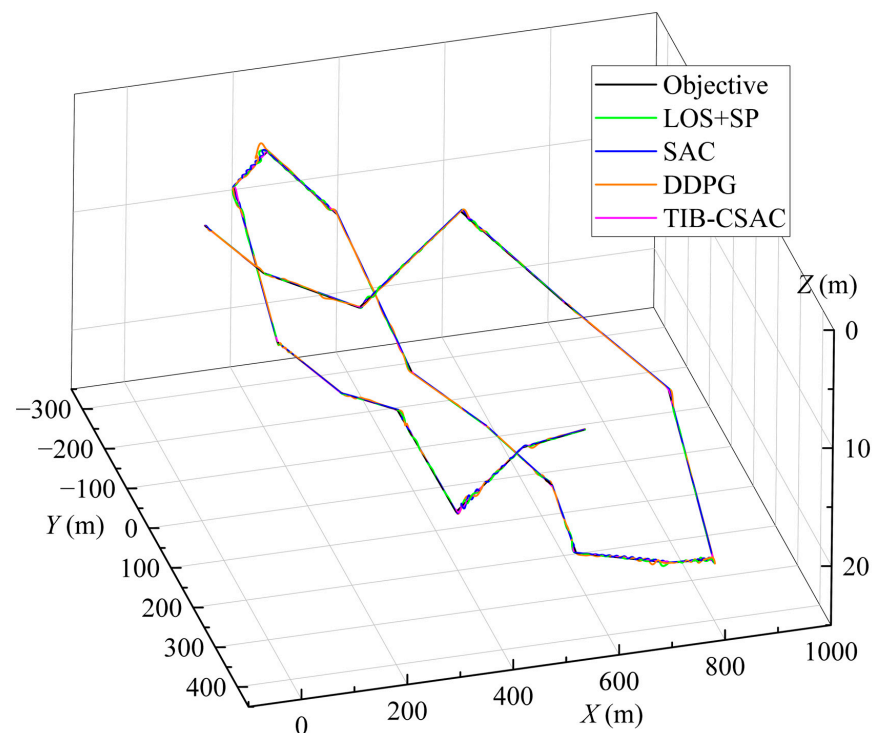
To further evaluate the robustness of the proposed controller under stochastic environmental disturbances, an additional test is conducted using a first-order Gauss–Markov disturbance model [28]. The test is performed on a 3D random path with 20 waypoints, where the coordinate origin is defined as the path-starting point. The desired surge velocity is set to  $u_d(t) = -0.75 \cos(0.01t) + 1.25$ . The waypoint configuration is summarized in Table 13, and the results are shown in Figures 20–22 and Tables 14 and 15.

$$\begin{aligned}\tau_k^d &= \text{clip}(\tau_k^d, -A, A) \\ \tau_{k+1}^d &= \rho \tau_k^d + \sigma_d \sqrt{1 - \rho^2} \varepsilon_k \\ \rho &= \exp(-\Delta t / T_c)\end{aligned}\quad (66)$$

where  $\tau^d = [\tau_u^d, \tau_v^d, \tau_w^d, \tau_q^d, \tau_r^d]^T$ ;  $A = [15.0, 1.6, 1.6, 5.0, 4.0]^T$  denotes the theoretical upper bound of each disturbance channel in n Equation (48);  $\varepsilon_k \sim \mathcal{N}(0, I_5)$ ;  $\sigma_d = A/3$ ; and  $T_c = 30$  s.

**Table 13.** Coordinates of the reference-path waypoints.

| Index       | 1      | 2       | 3       | 4       | 5      | 6      | 7      | 8      | 9      | 10     |
|-------------|--------|---------|---------|---------|--------|--------|--------|--------|--------|--------|
| $x_i^p$ (m) | 112.13 | 307.52  | 497.67  | 657.61  | 813.02 | 835.02 | 731.12 | 619.40 | 621.43 | 525.10 |
| $y_i^p$ (m) | 0.00   | −32.65  | −33.77  | 52.53   | 155.41 | 311.17 | 365.13 | 209.29 | 96.76  | 30.63  |
| $z_i^p$ (m) | 4.72   | 9.98    | 3.04    | 8.81    | 13.64  | 23.01  | 20.88  | 24.59  | 22.71  | 19.30  |
| Index       | 11     | 12      | 13      | 14      | 15     | 16     | 17     | 18     | 19     | 20     |
| $x_i^p$ (m) | 408.72 | 315.79  | 229.22  | 138.34  | 161.95 | 218.32 | 283.15 | 423.69 | 509.14 | 595.53 |
| $y_i^p$ (m) | −33.07 | −152.37 | −264.84 | −196.47 | −52.90 | 98.86  | 197.75 | 133.86 | 224.75 | 297.61 |
| $z_i^p$ (m) | 16.14  | 6.18    | 4.09    | 4.13    | 12.66  | 12.21  | 10.74  | 22.41  | 14.44  | 11.01  |



**Figure 20.** 3D trajectories.



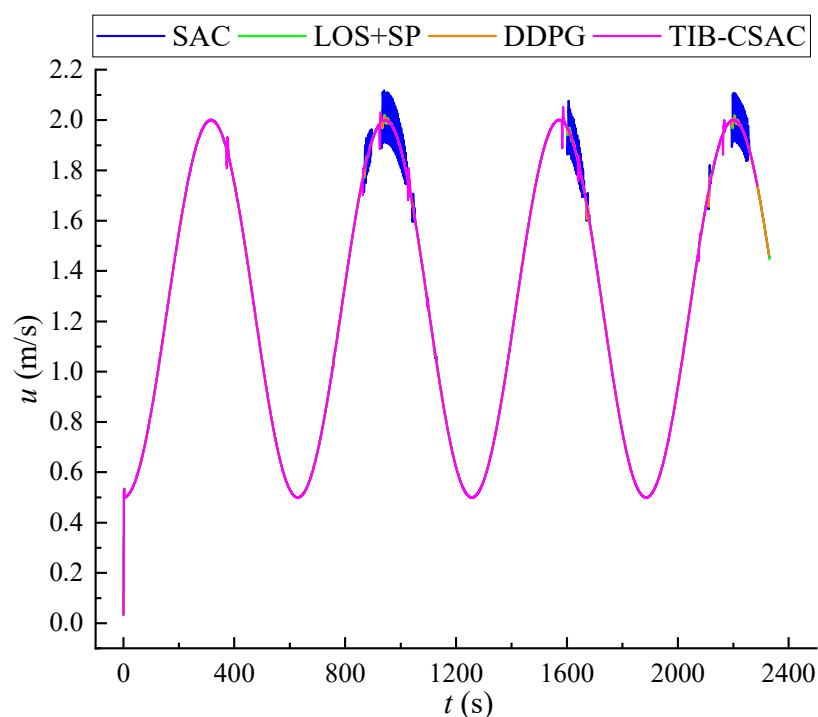


Figure 21. Surge velocity.

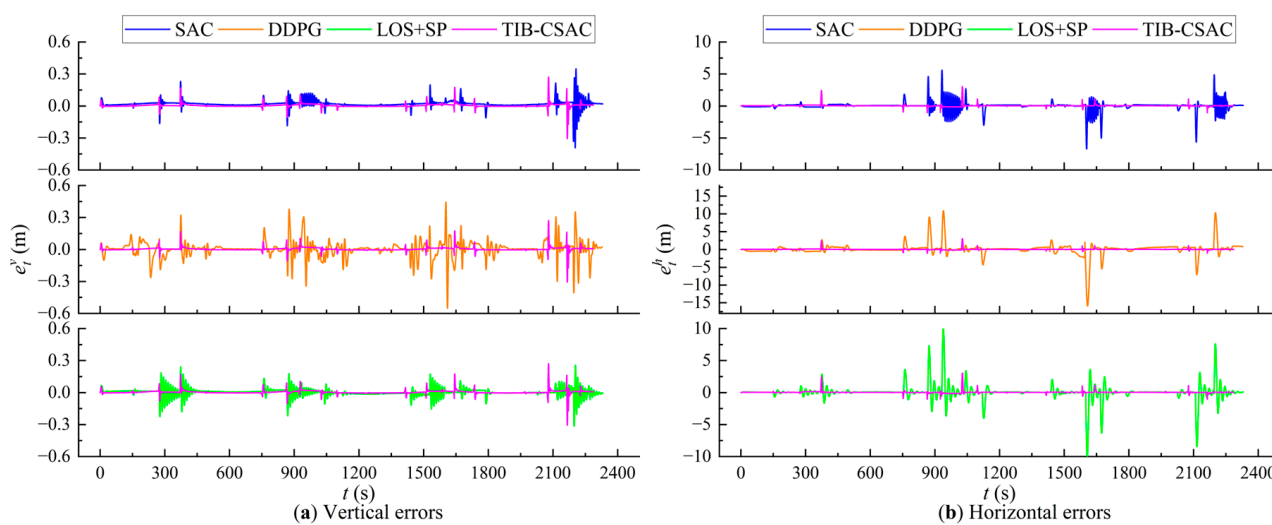


Figure 22. Path-following error.

Table 14. Path following error metrics.

|            | Metric                 | TIB-CSAC                | LOS+SP                  | SAC                     | DDPG                    |
|------------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Vertical   | MXAE (m)               | $3.0580 \times 10^{-1}$ | $3.1465 \times 10^{-1}$ | $3.9217 \times 10^{-1}$ | $5.5085 \times 10^{-1}$ |
|            | MAE (m)                | $8.9867 \times 10^{-3}$ | $2.3031 \times 10^{-2}$ | $2.5653 \times 10^{-2}$ | $4.6731 \times 10^{-2}$ |
|            | CVaR <sub>95</sub> (m) | $6.6192 \times 10^{-2}$ | $1.3515 \times 10^{-1}$ | $1.3048 \times 10^{-1}$ | $2.6511 \times 10^{-1}$ |
|            | $V_\varepsilon$        | $7.9594 \times 10^{-3}$ | $4.0323 \times 10^{-2}$ | $3.1242 \times 10^{-2}$ | $1.2729 \times 10^{-1}$ |
| Horizontal | MXAE (m)               | $3.0449 \times 10^0$    | $1.0180 \times 10^1$    | $6.7056 \times 10^0$    | $1.5886 \times 10^1$    |
|            | MAE (m)                | $6.2952 \times 10^{-2}$ | $5.0292 \times 10^{-1}$ | $3.0592 \times 10^{-1}$ | $7.7647 \times 10^{-1}$ |
|            | CVaR <sub>95</sub> (m) | $4.6887 \times 10^{-1}$ | $4.9808 \times 10^0$    | $2.6753 \times 10^0$    | $6.6688 \times 10^0$    |
|            | $V_\varepsilon$        | $1.2621 \times 10^{-1}$ | $4.2244 \times 10^{-1}$ | $5.8750 \times 10^{-1}$ | $9.3249 \times 10^{-1}$ |



**Table 15.** Rudder metrics.

| Metric               | TIB-CSAC                | LOS+SP                  | SAC                     | DDPG                    |
|----------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| $R_{\delta}^{sat}$   | $7.3909 \times 10^{-3}$ | $4.1181 \times 10^{-3}$ | $5.3944 \times 10^{-2}$ | $4.9386 \times 10^{-3}$ |
| $J_{\Delta^2\delta}$ | $5.0755 \times 10^{-1}$ | $2.2047 \times 10^0$    | $2.1324 \times 10^{-1}$ | $1.8090 \times 10^1$    |
| $J_{\dot{\delta}}$   | $1.3140 \times 10^1$    | $2.5033 \times 10^1$    | $3.1093 \times 10^1$    | $1.1789 \times 10^2$    |
| $J_{\delta,u}$       | $2.5234 \times 10^1$    | $1.4796 \times 10^1$    | $1.7047 \times 10^2$    | $1.6914 \times 10^1$    |
| $J_{AUV}$            | $7.2190 \times 10^3$    | $7.4023 \times 10^3$    | $7.4040 \times 10^3$    | $7.3943 \times 10^3$    |
| $T$                  | $2.2866 \times 10^3$    | $2.3312 \times 10^3$    | $2.3302 \times 10^3$    | $2.3286 \times 10^3$    |

As shown in Figure 20, all methods complete the 3D path-following task under stochastic disturbances. The error curves in Figure 22 show that the main error fluctuations are concentrated around path-segment transitions and high-curvature regions. Under stochastic disturbances, TIB-CSAC suppresses both vertical and horizontal error fluctuations more effectively than the baseline methods. As summarized in Table 14, TIB-CSAC achieves the lowest MXAE, MAE, CVaR95, and  $V_e$ , indicating better suppression of peak errors, average errors, tail errors, and threshold-exceeding errors under stochastic environmental disturbances. The supplementary control-related results in Table 15 further show that this improvement is not accompanied by excessive rudder saturation or unreasonable task-level effort. Therefore, the Gauss–Markov disturbance test provides further evidence of the stochastic-disturbance robustness of the proposed method.

## 5. Conclusions

This study proposes a TIB-CSAC method to improve the X-rudder path-following performance of an underactuated AUV under varying path geometries and desired surge velocities. The proposed method learns the X-rudder control policy directly from two-segment path-following information, thereby avoiding the hierarchical guidance-control architecture and the associated coupled parameter tuning. In addition, a task-informed inductive-bias encoder is introduced to support segment-wise policy learning, improving sample efficiency and overall training quality. A multi-head conservative value-evaluation mechanism is also incorporated to mitigate return drawdowns induced by challenging path-following tasks and to improve tail-stage convergence stability. Overall, compared with baseline SAC, TIB-CSAC reduces  $\overline{TRE}$  and  $\sigma_{tail}$  to 0.40 and 0.24 times those of SAC, respectively, while yielding less negative  $\overline{AUC}$  and  $\overline{R}_{tail}$  values whose absolute magnitudes are reduced to 0.40 and 0.62 times those of SAC, respectively.

The path-following performance of the proposed method is validated in three representative scenarios with distinct path geometries and desired surge velocities. The results show that TIB-CSAC improves multiple position-error metrics compared with baseline SAC. Averaged over the three cases, the vertical and horizontal MXAE values are reduced by  $1.4645 \times 10^{-1}$  m and 6.4623 m, respectively. The corresponding reductions in MAE are  $1.2965 \times 10^{-2}$  m and  $4.2044 \times 10^{-1}$  m, while those in CVaR<sub>95</sub> are  $3.7903 \times 10^{-2}$  m and 3.8225 m. In addition, the vertical and horizontal  $V_e$  values are reduced by  $1.5675 \times 10^{-2}$  and  $3.5120 \times 10^{-1}$ , respectively. These results indicate that TIB-CSAC not only improves overall adherence to the reference path but also more effectively suppresses extreme position errors, reduces tail errors, and maintains path-following within the prescribed error threshold, thereby demonstrating improved robustness and reliability.

The current results are mainly obtained through simulation-based evaluation. To reduce the mismatch between the simulator and the real environment, sim-to-real transfer methods provide a promising direction for adapting DRL policies trained in simulation to real AUV systems with limited real-world data [29,30]. For example, Wu et al. improved the real-world deployment of a simulation-trained navigation policy using human-guided

interventions and demonstrations [31], while Hsu et al. fine-tuned a simulation-trained policy in a safety-constrained laboratory environment before deploying it on a quadrupedal robot for real-world visual navigation [32]. Therefore, future work will focus on software-system integration, real-world data collection, sim-to-real policy transfer learning, and open-water experiments to further evaluate the proposed method. In addition, the adopted 5-DOF model may be insufficient when rudder faults or asymmetric rudder effectiveness occur. Future work will also investigate fault-tolerant X-rudder control in a full 6-DOF framework by further considering roll dynamics and roll control.

**Author Contributions:** Conceptualization, J.T.; Methodology, J.T.; Software, L.Z.; Formal analysis, P.C.; Investigation, Z.L.; Resources, Y.S.; Writing—original draft, J.T.; Writing—review & editing, J.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study is supported by the Natural Science Foundation of Heilongjiang Province of China under Grant No. ZD2020E005. Additional support was provided by the Special Funds of the Taishan Industrial Leading Talents Project.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author due to privacy restrictions.

**Conflicts of Interest:** Author Yushan Sun was employed by the company Shandong Future Robot Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Appendix A

The actuator input of the thruster is modeled as

$$\tau_u^a(n) = \rho n^2 D^4 (K_{Tp1} \cdot u / nD + K_{Tp2}) \quad (A1)$$

where the seawater density is set to  $\rho = 1025 \text{ kg/m}^3$ , the propeller diameter is  $D = 0.25 \text{ m}$ , and the linear coefficients are  $K_{Tp1} = -0.3077$  and  $K_{Tp2} = 0.220$ .

The rotational speed of thruster  $n_t$  follows an incremental S-plane controller:

$$n_t = \bar{n}_t \cdot n_{\max} \quad (A2)$$

$$\bar{n}_t = \text{sat}(\bar{n}_{t-1} + \eta_n \cdot \mathcal{S}(e_u, \dot{e}_u)) \quad (A3)$$

$$e_u = \text{sat}\left(\frac{u_d - u}{\Delta u_{\max}}\right), \dot{e}_u = \text{sat}\left(\frac{\dot{u}_d - \dot{u}}{\Delta \dot{u}_{\max}}\right) \quad (A4)$$

$$\mathcal{S}(e, \dot{e}) = \frac{2}{1 + \exp(-k_1 e - k_2 \dot{e})} - 1 \quad (A5)$$

where  $\bar{n}_t$  is the normalized control output. The maximum rotational speed is set to  $n_{\max} = 1800 \text{ rpm}$ , the update rate is  $\eta_n = 0.08$ , normalization factors for error are  $\Delta u_{\max} = 2.0$  and  $\Delta \dot{u}_{\max} = 4.0$ , and the S-plane controller factors are  $k_1^u = 2.0$  and  $k_2^u = 1.8$ .

## References

- Shen, Z.; Wang, Q.; Dong, S.; Yu, H. Prescribed performance dynamic surface control for trajectory-tracking of unmanned surface vessel with input saturation. *Appl. Ocean Res.* **2021**, *113*, 102736. [CrossRef]
- He, L.; Zhang, Y.; Li, S.; Li, B.; Yuan, Z. Three-Dimensional Path Following Control for Underactuated AUV Based on Ocean Current Observer. *Drones* **2024**, *8*, 672. [CrossRef]
- Zheng, J.; Song, L.; Liu, L.; Yu, W.; Wang, Y.; Chen, C. Fixed-time sliding mode tracking control for autonomous underwater vehicles. *Appl. Ocean Res.* **2021**, *117*, 102928. [CrossRef]
- Rout, R.; Subudhi, B. Design of Line-of-Sight Guidance Law and a Constrained Optimal Controller for an Autonomous Underwater Vehicle. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 416–420. [CrossRef]

5. Wang, L.; Li, S.; Liu, J.; Wu, Q. Data-driven path-following control of underactuated ships based on antenna mutation beetle swarm predictive reinforcement learning. *Appl. Ocean Res.* **2022**, *124*, 103207. [\[CrossRef\]](#)
6. S., P.; Rajendran, S. A unified seakeeping and manoeuvring model with a PID controller for path following of a KVLCC2 tanker in regular waves. *Appl. Ocean Res.* **2021**, *116*, 102860. [\[CrossRef\]](#)
7. He, Y.; Xie, Y.; Pan, G.; Cao, Y.; Huang, Q.; Ma, S.; Zhang, D.; Cao, Y. Depth and Heading Control of a Manta Robot Based on S-Plane Control. *J. Mar. Sci. Eng.* **2022**, *10*, 1698. [\[CrossRef\]](#)
8. Xu, F.; Zhang, L.; Zhong, J. Three-Dimensional Path Tracking of Over-Actuated AUVs Based on MPC and Variable Universe S-Plane Algorithms. *J. Mar. Sci. Eng.* **2024**, *12*, 418. [\[CrossRef\]](#)
9. Jiang, C.; Lv, J.; Wan, L.; Wang, J.; He, B.; Wu, G. An Improved S-Plane Controller for High-Speed Multi-Purpose AUVs with Situational Static Loads. *J. Mar. Sci. Eng.* **2023**, *11*, 646. [\[CrossRef\]](#)
10. He, L.; Xie, M.; Zhang, Y. A Review of Path Following, Trajectory Tracking, and Formation Control for Autonomous Underwater Vehicles. *Drones* **2025**, *9*, 286. [\[CrossRef\]](#)
11. Wang, D.; Shen, Y.; Wan, J.; Sha, Q.; Li, G.; Chen, G.; He, B. Sliding mode heading control for AUV based on continuous hybrid model-free and model-based reinforcement learning. *Appl. Ocean Res.* **2022**, *118*, 102960. [\[CrossRef\]](#)
12. Sun, Y.; Ran, X.; Zhang, G.; Wang, X.; Xu, H. AUV path following controlled by modified Deep Deterministic Policy Gradient. *Ocean Eng.* **2020**, *210*, 107360. [\[CrossRef\]](#)
13. 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. [\[CrossRef\]](#)
14. Wang, Z.; Li, Y.; Ma, C.; Yan, X.; Jiang, D. Path-following optimal control of autonomous underwater vehicle based on deep reinforcement learning. *Ocean Eng.* **2023**, *268*, 113407. [\[CrossRef\]](#)
15. 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. [\[CrossRef\]](#)
16. Zhang, C.; Cheng, P.; Du, B.; Dong, B.; Zhang, W. AUV path tracking with real-time obstacle avoidance via reinforcement learning under adaptive constraints. *Ocean Eng.* **2022**, *256*, 111453. [\[CrossRef\]](#)
17. Zhang, Q.; Lin, J.; Sha, Q.; He, B.; Li, G. Deep Interactive Reinforcement Learning for Path Following of Autonomous Underwater Vehicle. *IEEE Access* **2020**, *8*, 24258–24268. [\[CrossRef\]](#)
18. Dong, N.; Liu, S.; Ip, A.W.H.; Yung, K.L.; Gao, Z.; Juan, R.; Wang, Y. End-to-end autonomous underwater vehicle path following control method based on improved soft actor–critic for deep space exploration. *J. Ind. Inf. Integr.* **2025**, *45*, 100792. [\[CrossRef\]](#)
19. Gu, N.; Wang, D.; Peng, Z.; Wang, J.; Han, Q.-L. Advances in line-of-sight guidance for path following of autonomous marine vehicles: An overview. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *53*, 12–28. [\[CrossRef\]](#)
20. Fan, J.; Li, W. DRIBO: Robust Deep Reinforcement Learning via Multi-View Information Bottleneck. In Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research, Baltimore, MD, USA, 17–23 July 2022; pp. 6074–6102.
21. Schaul, T.; Horgan, D.; Gregor, K.; Silver, D. Universal Value Function Approximators. In Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, Lille, France, 6–11 July 2015; pp. 1312–1320.
22. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; Wiley: Hoboken, NJ, USA, 2011.
23. Chai, P.; Sun, Y.; Wu, W.; Zhou, H.; Gao, F. AUV preset performance approximate dynamic programming path following control based on fixed time guidance law. *Ocean Eng.* **2025**, *321*, 120425. [\[CrossRef\]](#)
24. Zhang, L.; Sun, Y.; Chai, P.; Tan, J.; Zheng, H. Prescribed-performance time-delay compensation control for UUV trajectory tracking in main-branch water conveyance tunnel transitions under unknown input delays. *Ocean Eng.* **2025**, *342*, 122941. [\[CrossRef\]](#)
25. Pettersen, K.Y.; Egeland, O. Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle. *IEEE Trans. Autom. Control*. **1999**, *44*, 112–115. [\[CrossRef\]](#)
26. Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P. Soft actor-critic algorithms and applications. *arXiv* **2018**, arXiv:1812.05905. [\[CrossRef\]](#)
27. 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* **2015**, arXiv:1509.02971. [\[CrossRef\]](#)
28. An, X.; Xing, H.; Li, H.; Shi, H.; Gu, Y. Construction and verification of the autonomous underwater helicopter’s digital twin system for missions simulation. *Ocean Eng.* **2025**, *341*, 122641. [\[CrossRef\]](#)
29. Ju, H.; Juan, R.; Gomez, R.; Nakamura, K.; Li, G. Transferring policy of deep reinforcement learning from simulation to reality for robotics. *Nat. Mach. Intell.* **2022**, *4*, 1077–1087. [\[CrossRef\]](#)
30. Zhu, W.; Guo, X.; Owaki, D.; Kutsuzawa, K.; Hayashibe, M. A Survey of Sim-to-Real Transfer Techniques Applied to Reinforcement Learning for Bioinspired Robots. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3444–3459. [\[CrossRef\]](#) [\[PubMed\]](#)

31. Wu, J.; Zhou, Y.; Yang, H.; Huang, Z.; Lv, C. Human-Guided Reinforcement Learning With Sim-to-Real Transfer for Autonomous Navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14745–14759. [[CrossRef](#)]
32. Hsu, K.-C.; Ren, A.Z.; Nguyen, D.P.; Majumdar, A.; Fisac, J.F. Sim-to-Lab-to-Real: Safe reinforcement learning with shielding and generalization guarantees. *Artif. Intell.* **2023**, *314*, 103811. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.