

Article

BDAT-Planner: Bioinspired Dynamic Adaptive Threshold Planner for Underwater Collision Avoidance of AUVs

Boyang Zhang, Zhicheng Zhang and Weixing Feng *

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; zby0624@hrbeu.edu.cn (B.Z.); zeechenzhang@hrbeu.edu.cn (Z.Z.)

* Correspondence: fengweixing@hrbeu.edu.cn

Abstract

Safe and intelligent collision avoidance technology is essential for the autonomous underwater vehicle (AUV) to navigate in underwater environments. Most existing spike methods are constrained by a fixed static threshold and are unable to dynamically adjust to threshold changes reasonably, leading to difficulties in robustly adapting to external dynamic interference and thus resulting in insufficient homeostasis and generalization. To address these limitations, inspired by the dynamic threshold changes in biological neural systems, a bioinspired dynamic adaptive threshold (BDAT) is proposed. Combining the spiking neural network with deep reinforcement learning, a novel bioinspired dynamic adaptive threshold planner (BDAT-Planner) framework is constructed for underwater dynamic collision avoidance tasks performed by AUVs in complex, unknown environments. The proposed BDAT-Planner consists of the spiking dynamic adaptive actor network (SDAAN) and the deep critic normal network (DCNN). The BDAT is deployed to each spiking neuron in the SDAAN, dynamically adjusting the spike firing rate through threshold changes and avoiding excessive excitation or inhibition, thus maintaining homeostasis. The spiking encoder and spiking decoder are designed to convert continuous information and spiking sequences. Experimental results from both the training process and evaluation process (ablation studies, comparison experiments, and homeostasis experiments) demonstrate that the proposed BDAT-Planner has achieved superior performance in dynamic collision avoidance and model homeostasis compared to static threshold methods and existing comparison methods. The novel idea of bioinspired dynamic adaptive threshold can maintain model homeostasis and effectively enhance its adaptability to external dynamic interference, which offers significant development potential for promoting the efficient and stable operation of AUVs in marine environments.

Keywords: autonomous underwater vehicle; dynamic collision avoidance; spiking neural network; deep reinforcement learning; dynamic adaptive threshold



Academic Editor: Jinfen Zhang

Received: 4 May 2026

Revised: 27 May 2026

Accepted: 29 May 2026

Published: 30 May 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

Autonomous underwater vehicles (AUVs) can perform autonomous operations in complex and unknown marine environments. With advantages such as high autonomy, maneuverability, and environmental adaptability, they have been widely applied in fields such as marine topographic mapping, marine environmental monitoring, underwater resource exploration, and seabed facility monitoring [1–3]. Underwater collision avoidance technology, as one of the core technologies for intelligent and safe underwater navigation of AUVs, is applied during the entire navigation process [4]. The underwater environment

is dynamic and complex, with most of its characteristics being unknown a priori. When the AUV operates in unfamiliar waters, it may encounter various obstacles, particularly dynamic obstacles. The speed and direction of dynamic obstacles are often more difficult to determine, requiring the AUV to make accurate decisions on how to respond. Even in the presence of external disturbances, the model can still maintain stability, ensuring its generalization and adaptability to achieve safe and efficient collision avoidance [5,6].

Traditional collision avoidance methods [7] typically operate under the premise of known maps, relying heavily on the current environment, with significantly insufficient autonomy and intelligence. The development of reinforcement learning (RL) [8] and deep reinforcement learning (DRL) [9,10] has enabled collision avoidance tasks to gradually get rid of map dependency. Through interactive trial-and-error learning with the environment, these methods have demonstrated certain advantages in solving collision avoidance decision problems. Nevertheless, such methods lack the capability for rapid processing and interference resistance, particularly in unknown dynamic underwater environments, thereby impacting the intelligent and safe navigation of AUVs.

The biological brain is a complicated organ that consists of a large number of functionally diverse neurons. These neurons are the fundamental information processing units in the brain, and are structurally interconnected in specific configurations by synapses [11]. As it has evolved, the brain has gradually established a series of intrinsic mechanisms to ensure its tremendous information-processing ability [12,13]. Consequently, the better we understand biologically plausible brain mechanisms, the more effectively we can design efficient and reliable networks. Spiking neural network (SNN) is the new generation of neural network that has received considerable attention in recent years [14]. In contrast to traditional artificial neural networks (ANNs), spiking neurons emit spike signals when their membrane potential crosses the threshold and propagate to the postsynaptic neuron. By mimicking brain-inspired computational policies, the SNN has demonstrated enormous potential [15,16]. The advantages of the SNN perfectly align with the requirements of AUV collision avoidance tasks, particularly in unknown and dynamic underwater environments. The integration of both will bring new opportunities for the intelligence and autonomy of underwater robots.

However, simple fusion struggles to explore the biological advantages of SNNs. Biological neurons utilize the spontaneous regulation mechanism to maintain model homeostasis, ensuring a stable spiking firing rate and excitability, to robustly adapt to various external environmental interferences and exhibit strong generalization [17,18]. Additionally, a dynamic threshold, one type of regulatory mechanism in biological neurons, plays an essential role in maintaining neuronal homeostasis by regulating the firing rate of neurons [19,20]. It is worth mentioning that most existing SNNs focus solely on synaptic plasticity between neurons, neglecting the homeostasis properties within neurons themselves. Specifically, spiking neurons are constrained by the fixed static threshold, which is hard to adapt to varying external conditions, thereby limiting the diversity of neurons and the expression of the model. It will cause the model to struggle with dynamic obstacles and external environmental interference, thereby compromising collision avoidance performance and potentially increasing collision risk. Consequently, spiking neurons need to dynamically regulate the threshold appropriately to maintain model homeostasis, to robustly adapt to various external conditions, which is crucial and significant for dynamic collision avoidance of AUVs.

Consequently, in this paper, combining an SNN and DRL, a bioinspired dynamic adaptive threshold (BDAT) is introduced, and a novel bioinspired dynamic adaptive threshold framework (BDAT-Planner) is proposed for dynamic collision avoidance tasks of AUVs in complex and unknown underwater environments. The proposed BDAT-Planner

consists of the spiking dynamic adaptive actor network (SDAAN) and the deep critic normal network (DCNN). The proposed BDAT combines an adaptive energy threshold (AET), dynamic depolarization threshold (DDT), and retrospective gradient threshold (RGT), which correlate the dynamic adaptive threshold with the membrane potential of spiking neurons, the preceding rate of depolarization, and the historical variation rate of the threshold, respectively. We deploy the BDAT into each spiking neuron, which mimics the behavior of threshold self-adaptive dynamic changes in biological neural systems by dynamically regulating the firing rate of neurons through threshold changes to maintain model homeostasis. Simultaneously, the spiking encoder and spiking decoder are designed to convert continuous information and spiking sequences. Experimental results from the training and evaluation process have demonstrated that, in comparison, under both normal conditions and external interference, the BDAT-Planner proposed in this paper has achieved the best results in both dynamic collision avoidance performance and spiking model homeostasis. The dynamic adaptive threshold method proposed in this paper can provide AUVs with a safe and efficient collision avoidance ability, which can perform strong adaptability and generalization when facing external dynamic interference.

In summary, the main contributions of this paper are as follows:

- (1) In this paper, combining an SNN and DRL, a novel bioinspired dynamic adaptive threshold framework, BDAT-Planner, is proposed for underwater dynamic collision avoidance tasks of AUVs in complex and unknown environments, ensuring both safety and efficiency.
- (2) The proposed BDAT combines an AET, DDT, and RGT to mimic the behavior of threshold-adaptive dynamic changes in biological neural systems. By dynamically regulating the firing rate through spiking threshold changes, excessive excitation or inhibition of neurons is avoided, thereby maintaining homeostasis.
- (3) Experimental results have demonstrated that our proposed dynamic adaptive threshold framework achieves biologically plausible homeostasis. It can robustly adapt to external dynamic interference and offer strong generalization, which significantly enhances the dynamic collision avoidance ability of AUVs in complex underwater environments.

The remainder of this paper is organized as follows: Section 2 reviews relevant work, which provides the theoretical background for the methods. Section 3 performs modeling of the AUV, including the coordinate reference system, the hydrodynamic model, and forward-looking sonar. Section 4 details the proposed BDAT-Planner framework, including the overall framework, the BDAT, the spiking encoding and spiking decoding, and the hybrid training algorithm. Section 5 analyzes the training and evaluation experiments, including experimental setup, training performance analysis, ablation experiments, comparative experiments, visualized trajectories, and homeostasis results. Finally, based on the relevant research, model framework, and experimental results, Section 6 summarizes the entire work and discusses future research prospects.

2. Related Work

2.1. DRL Method

RL and DRL optimize policy generation through continuous interaction between agents and environments, which gets rid of the dependency on scene maps in traditional methods. The algorithm's efficiency is further enhanced, and it has attracted widespread attention from researchers in robot collision avoidance tasks [21,22]. Zhang et al. [23] proposed a novel hybrid framework for AUVs operating in dynamically constrained underwater environments, which combined multithreaded energy-optimized batch-informed trees (MEO-BITs) and deep Q-networks (DQNs). This method facilitated dynamic obstacle

detection and adaptive local navigation, which validated the superiority of the proposed framework in simulation experiments. Xia et al. [24] proposed a DRL approach based on proximal policy optimization (PPO). By utilizing the relative position between the obstacles and USV and the closest approach point, high-dimensional state features were constructed, including the collision track layer and danger layer, and a convolutional neural network (CNN) structure that fused multiple features, and multiple scales were designed. The experiment verified that the method could safely avoid dynamic obstacles. For non-cognitive environments involving dynamic obstacles and external interference, Wang et al. [25] proposed a real-time collision avoidance strategy based on a flexible steering method under the framework of improved soft actor-critic (SAC), and designed a disturbance estimation and cancelation method based on dynamic compensation.

2.2. SNN Method

Compared to the collision avoidance mechanisms in nature, existing robotic collision avoidance methods still have many shortcomings. For instance, facing unknown and uncertain environments, they cannot dynamically adjust policies, and their environmental adaptability is poor. Consequently, drawing on the information processing mechanisms of biological collision avoidance systems in nature is crucial for the research on dynamic collision avoidance policies for AUVs. The SNN is regarded as the third generation of neural networks. It transmits spike signals to postsynaptic neurons, which has advantages such as event-driven and spatial-temporal information processing [26,27]. The advantages of an SNN make its integration with AUV dynamic collision avoidance more extensive and rational. Recently, researchers have been exploring the integration of SNNs into the field of DRL. Related studies have directly converted a DQN into an SNN [28,29]. However, compared to the ANN with the same structure, the models directly converted typically exhibit lower performance. To mitigate the issue of performance degradation, researchers have proposed several improved policies. Zhang et al. [30] proposed a novel neuromorphic planner with spiking memory (NSM-Planner) framework, introducing a spiking memory unit called soft spiking_GRU (SS_GRU) capable of storing updated historical information. Wang et al. [31] proposed a modular navigation controller based on an SNN, consisting of three sub-controllers: the obstacle-avoidance SNN controller, the wall-following SNN controller, and the goal-approaching controller. The method primarily relies on manually specified rules for collision avoidance and navigation. When environmental conditions change, the model lacks generalization. Liu et al. [32] proposed a brain-inspired navigation method based on an SNN, which introduced the asymptotic gradient method to optimize backpropagation during training to enhance robustness.

Additionally, researchers have further refined the SNN model, and have explored its effectiveness in complex decision-making tasks. Tang et al. [33] designed a robot-based collision avoidance navigation model, which applied the proposed method to mapless navigation research. Zhao et al. [34] proposed a reward-modulated spiking neural network (RSNN) that integrates biomimetic reward modulation. Compared to online learning methods based on ANNs, the method has exhibited better stability. Sanyal et al. [35] proposed an event-based physics-guided neuromorphic planner (EV-Planner) for collision avoidance tasks, which introduced a shallow spiking neural network and a lightweight energy-aware physics-guided neural network (PgNN). Yang et al. [36] proposed the multi-critic DDPG with a spiking memory (MC-DDPGSM) framework for collision avoidance navigation tasks. The proposed method has achieved better results in terms of success rate and navigation time. Zhang et al. [37] proposed an efficient collision avoidance strategy based on an SNN. The experimental results have demonstrated that the proposed method achieves a high navigation success rate and temporal efficiency. Although these improvements have

enhanced the performance of the SNN to some extent, spiking neurons focus solely on updating weights between synapses while neglecting the internal homeostasis of neurons themselves and lack adaptability to different external conditions, which limits the expressive ability of SNNs. Consequently, if the above method is applied to underwater dynamic collision avoidance tasks, it will also face challenges such as the difficulty in adaptively adjusting homeostasis.

2.3. Dynamic Threshold Method

Self-homeostasis behavior of biological neurons is extensively observed in various biological nervous systems in nature. It can regulate neuronal excitability, utilizing self-regulatory mechanisms to maintain internal homeostasis [38–40]. Recently, some researchers have applied the homeostasis of neurons to SNNs. Zhang et al. [41] proposed a soft-reset leaky integrate-and-fire (LIF) model that optimizes the exponential distribution of membrane potential states. By regulating the membrane potential as an auxiliary output signal to the desired distribution, it maximizes information entropy. Fang et al. [42] endowed each spiking neuron with a learnable membrane time constant, which automatically adapts to the temporal structure of the input signal to accommodate varying stimulus frequencies and task requirements. However, this method is difficult to adapt to dynamically changing tasks. Sun et al. [43] proposed a novel collaborative learning method by leveraging the principle that spiking thresholds are intrinsic neural characteristics. It exhibits rich dynamics on a millisecond timescale, with an appropriate firing rate providing stable signal transmission for the model.

The dynamic threshold, as one of the fundamental properties of biological neurons, serves as a regulatory mechanism for maintaining neuronal homeostasis, which has demonstrated the effectiveness of the dynamic threshold in multiple studies [44,45]. Previous studies have been mostly confined to the static constant threshold. The dynamic threshold more closely mimics the complex properties of biological neurons, which can effectively adaptively adjust the threshold magnitude to ensure the homeostasis of the overall firing rate. Wester et al. [46] showed through theoretical and electrophysiological studies that the threshold increases with rising membrane potential, limiting high firing rates and suppressing low firing rates. Higgs et al. [47] tested the hypothesis that low-voltage-activated Kv1 channels affect threshold dynamics in layer 2–3 PNs, which indicated that Na⁺ channel inactivation and K⁺ channel activation are the primary mechanisms for threshold regulation. The variation in the threshold is negatively correlated with the depolarization rate of neurons, which makes neurons better at filtering and modulates their reflective sensitivity. Hao et al. [48] combined the symmetric spike-timing-dependent plasticity (sym-STDP) rule with biologically plausible synaptic scaling and the intrinsic plasticity of the dynamic threshold. When the neuronal threshold is excessively high, neurons rarely fire action potentials. To mitigate the threshold increase, the dynamic threshold model DT1 was proposed. Kim et al. [49] added a device model with a non-linear weight update to evaluate the performance of SNNs, which introduced a self-designed dynamic threshold model DT2. Simulation experiments have demonstrated that this method yields better results, but it is still subject to the influence of constant parameters since the firing rate is maintained at a manually set constant solely through thresholds.

At present, existing dynamic threshold methods exhibit certain shortcomings, such as simplistic model structures, manual configuration of dynamic scaling factors, and the continued use of fixed constants during evaluation. Related applied research remains in the primary stages. The underwater environment is complicated and unknown, which makes dynamic collision avoidance tasks more demanding on the generalization and robustness of the model. Consequently, it is necessary to design a biologically plausible dynamic

adaptive threshold method that effectively adapts to external dynamic interference and maintains model homeostasis, which is critical for AUVs in the underwater dynamic collision avoidance tasks.

3. Modeling of AUV Collision Avoidance

In this work, the structural modeling of the AUV is completed with SolidWorks 2021 SP5.1 software. To achieve the dynamic collision avoidance tasks of AUVs in complex and unknown underwater scenarios at a fixed depth [50], it is simplified to a two-dimensional horizontal collision avoidance problem, where the effects of heave, roll, and pitch of ship dynamics are ignored. The reference coordinate frame of the AUV is shown in Figure 1. The earth-fixed frame $O - X_0Y_0Z_0$ is located on the earth, and the body-fixed frame $\Omega - XYZ$ is placed at the center of its gravity. We define the position and orientation $\eta = [x, y, \psi]^T$ in the earth-fixed frame $O - X_0Y_0Z_0$ and define the velocity vector corresponding to the surge, sway, and yaw $V = [u, v, r]^T$ in the body-fixed frame $\Omega - XYZ$.

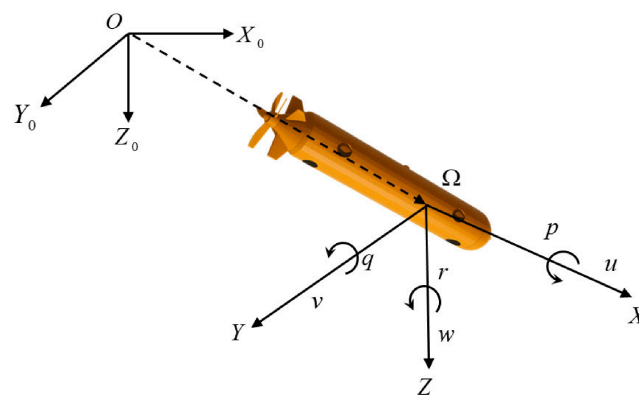


Figure 1. Reference coordinate frames of AUV.

The kinematic equations of the AUV are as follows:

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

where the transformation matrix $J(\psi)$ is expressed as

$$J(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The AUV can be approximated as a uniformly massed rigid body with vertical and horizontal symmetry. The Newton–Euler equations are utilized to establish the dynamic model as follows:

$$M\dot{V} + C(V)V + D(V)V = \tau \quad (3)$$

where M denotes the mass and inertia matrix, $C(V)$ represents the Coriolis centripetal matrix, and $D(V)$ represents the hydrodynamic damping matrix. τ denotes the expression entered by the actuator.

$$\begin{aligned} M &= M_{RB} + M_A \\ &= \begin{bmatrix} m - X_{\ddot{u}} & 0 & 0 \\ 0 & m - Y_{\ddot{v}} & mx_G - Y_{\ddot{r}} \\ 0 & mx_G - Y_{\ddot{r}} & I_z - N_{\ddot{r}} \end{bmatrix} \end{aligned} \quad (4)$$

$$C(V) = C_{RB}(V) + C_A(V)$$

$$= \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r - m(x_G r + v) \\ 0 & 0 & mu - X_{\dot{u}}u \\ m(x_G r + v) - Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u - mu & 0 \end{bmatrix} \quad (5)$$

$$D(V) = \begin{bmatrix} X_{|u|u}|u| + X_u & 0 & 0 \\ 0 & Y_{|v|v}|v| + Y_v & -Y_{|v|r}|v| - Y_r \\ 0 & -N_{|r|v}|r| - N_v & N_{|r|r}|r| + N_r \end{bmatrix} \quad (6)$$

where M_{RB} and M_A represent the inertia matrices of the rigid body and the added mass, respectively. m represents the mass of the AUV, and x_G refers to the horizontal coordinate of its center of gravity. $C_{RB}(V)$ and $C_A(V)$ denote the Coriolis centripetal matrices of the rigid body and the added mass, respectively. Other variables denote the hydrodynamic coefficients of the AUV, such as $X_{\dot{u}}$, $N_{\dot{r}}$, and $Y_{v|v}|v|$, etc.

The complexity of the unknown underwater environment imposes certain limitations on AUVs in terms of environmental perception. Optical sensors are constrained by limitations such as low underwater visibility and poor contrast, while electromagnetic waves are hindered by issues like rapid attenuation underwater and short propagation distances. Thus, an acoustic device, i.e., forward-looking sonar, is employed as the primary sensor for obstacle detection in the AUV. Its advantages mainly lie in the superiority of sound wave propagation underwater, reliable obstacle detection data, and adaptability to different aquatic environments. The forward-looking sonar selected can detect a 120° horizontal sector, comprising 18 beams numbered 0 to 17 from left to right, which provide 18 ranging values as obstacle detection information returned by the sensor.

This study aims to propose a bioinspired framework that can robustly adapt to various external conditions while maintaining homeostasis of the model. It is applied to underwater dynamic collision avoidance tasks for AUVs in complex and unknown environments. More specifically, when the AUV encounters a dynamic underwater obstacle, the membrane potential of the spiking neuron undergoes a rapidly rising phase. At this time, the overall threshold should be lowered to encourage spike firing, to promptly adapt to the external dynamic interference. Nevertheless, when underwater noise induces slight fluctuations in the membrane potential of spiking neurons, the overall threshold should be increased to suppress noise interference. Consequently, by dynamic adaptive learning of the threshold of spiking neurons, which avoids excessive excitation or inhibition of neurons, the adaptability and generalization of the model are enhanced. Even in complex and unknown underwater environments with external interference, it can still avoid obstacles stably and efficiently, successfully reaching the target point, as shown in Figure 2.

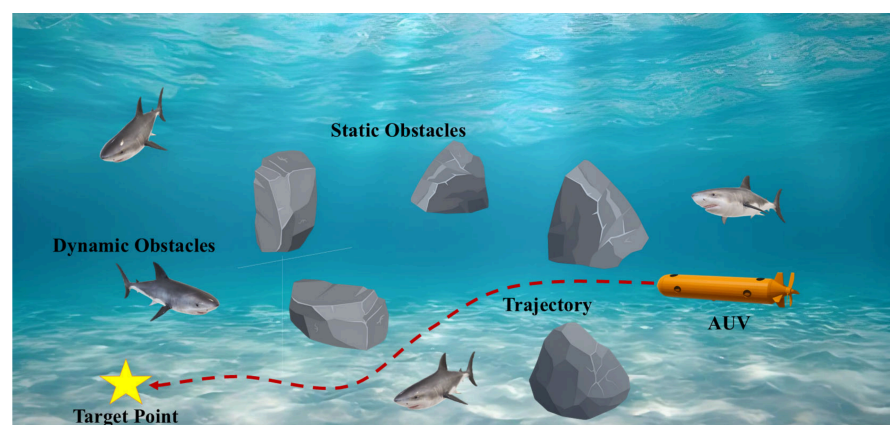


Figure 2. Dynamic collision avoidance of AUV in the underwater environment.

encoder and the spiking decoder are required for conversion. We deploy the proposed BDAT into each spiking neuron and construct the SDAAN. S_i is processed by the spiking encoder and then input into the SDAAN. A_i is decoded by the spiking decoder. The AUV interacts with the environment to obtain R_i . We store $\{S_i, A_i, R_i, S_{i+1}\}$ in the experience replay buffer and obtain $\{S_k, A_k, R_k, S_{k+1}\}$ through random sampling. Additionally, the target spiking dynamic adaptive actor network (TSDAAN) and target deep critic normal network (TDCNN) are constructed. We update the DCNN by minimizing $L_{TD}(\theta)$, and update the SDAAN by maximizing $L_Q(\zeta)$. The soft update method is employed to update the parameters of the target networks by weighting the parameters of the SDAAN and DCNN with certain percentages to the TSDAAN and TDCNN, respectively.

4.2. Bioinspired Dynamic Adaptive Threshold

In the natural biological nervous system, the threshold of neurons is not a fixed constant but dynamically changes. It changes in response to membrane potential variations and adaptively adjusts the threshold to ensure stable firing rates. It outputs stable and uniform spikes and maintains the model in homeostasis. If the static threshold is adopted, when external conditions cause interference, neurons may exhibit excessive excitement or negativity due to abnormally high or low firing rates over a certain period, thereby compromising the overall model performance. Figure 4 shows the firing mode of a spiking neuron inspired by the biological brain.

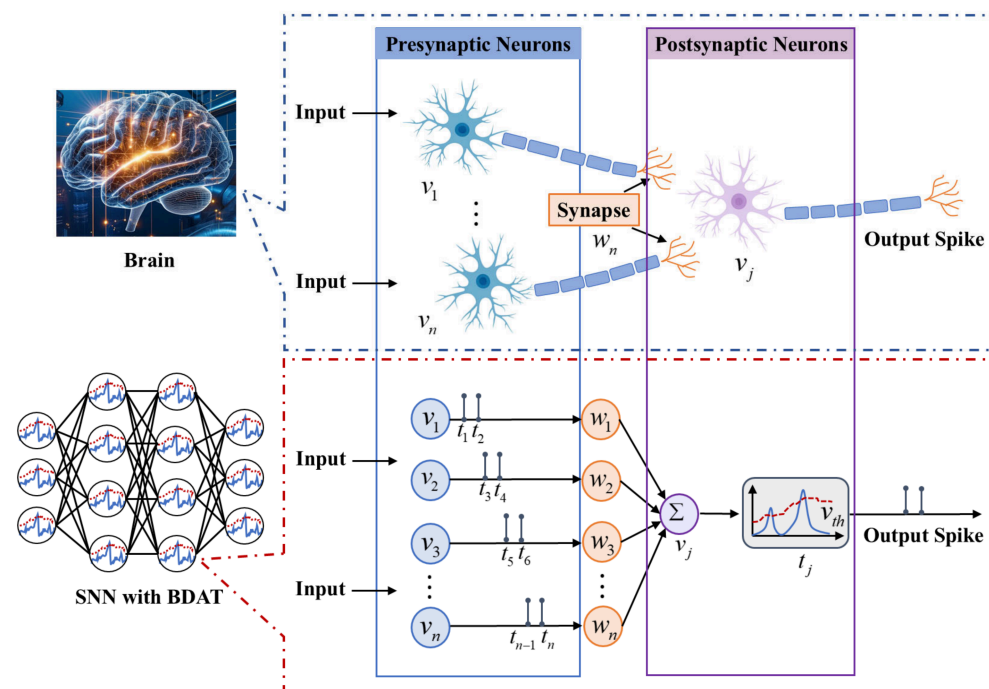


Figure 4. Bioinspired fire mode of spiking neurons.

The BDAT is a bioinspired dynamic adaptive threshold model, which can predict the current spiking firing threshold based on the interlayer membrane potential, the preceding rate of depolarization, and the historical rate of the threshold. Among these, the AET utilizes the principle that the dynamic adaptive threshold is positively correlated with the average membrane potential value to estimate the threshold. The DDT exhibits a biological observation that the dynamic adaptive threshold is negatively correlated with the preceding rate of depolarization. Simply focusing on the prior depolarization rate is insufficient, lacking memory and failing to consider the transient changes in longer-term neuronal activity patterns. The RGT utilizes the historical rate of change in the dynamic

adaptive threshold to provide neurons with memory of past states. The three components work in collaboration to obtain the optimal threshold, which ensures the model outputs stable spikes. As shown in Figure 5, the BDAT is deployed for each layer of spiking neurons in the SDAAN. For the j -th neuron in the n -th layer, the BDAT is expressed as follows:

$$\theta_{DA,j}^n(t+1) = \lambda \cdot \theta_{E,j}^n(t) + \beta \cdot \theta_{D,j}^n(t+1) + \gamma \cdot \theta_{G,j}^n(t+1) \quad (7)$$

where $\theta_{E,j}^n(t)$ denotes the AET, $\theta_{D,j}^n(t+1)$ represents the DDT, and $\theta_{G,j}^n(t+1)$ denotes the RGT. λ , β and γ refer to the weighting parameters. We deploy the BDAT to each spiking neuron in the SDAAN, and update the membrane potential and dynamic threshold with the above equation.

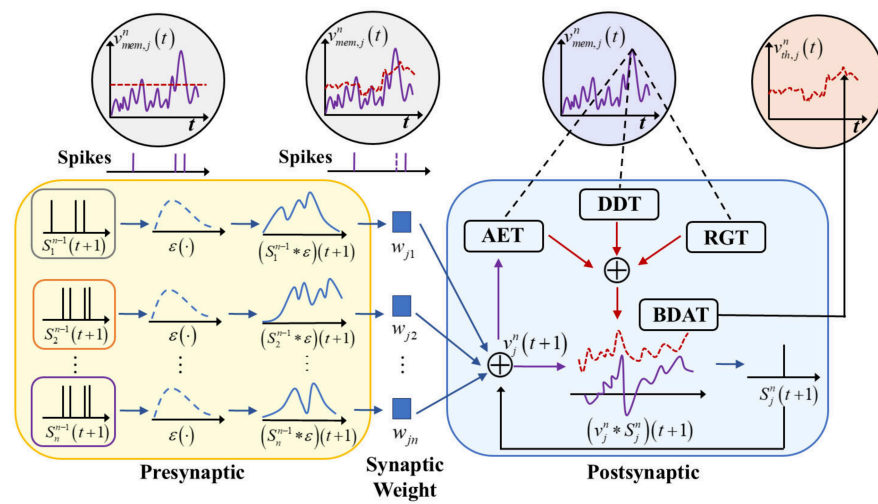


Figure 5. The BDAT processing flow.

The proposed BDAT is deployed to each spiking neuron, as shown in the following equations:

$$C_n^t = \eta_c \cdot C_n^{t-1} + W_n \cdot O_{n-1}^t + b_n \quad (8)$$

$$U_n^t = \eta_v \cdot U_n^{t-1} \cdot \chi(O_n^{t-1}) + C_n^t \quad (9)$$

$$\chi(O_n^{t-1}) = 1 - O_n^{t-1} \quad (10)$$

$$O_n^t = h(U_n^t) = H(U_n^t - \theta_{DA,j}^n) \quad (11)$$

where C_n^t represents the membrane current and U_n^t devotes the membrane voltage. W_n and b_n represent the synaptic weight matrix and synaptic bias vector, respectively. $\chi(O_n^{t-1}) = 1 - O_n^{t-1}$ represents the reset gate. η_c and η_v are decay factors. $H(U_n^t - \theta_{DA,j}^n)$ refers to the fire gate. $\theta_{DA,j}^n$ represents the dynamic adaptive threshold.

When the membrane potential of spiking neurons undergoes a rapid rise, such as when the AUV encounters underwater dynamic obstacles, the overall threshold $\theta_{DA,j}^n$ should be decreased to encourage spike firing, to robustly adapt to external dynamic obstacle interference. In this case, even though the AET increases its threshold $\theta_{E,j}^n$ as the membrane potential rises, the DDT decreases its threshold $\theta_{D,j}^n$ more rapidly. The RGT employs the average of the dynamic threshold historical change rate $\theta_{G,j}^n$ to avoid sensitive threshold changes. Additionally, when the membrane potential slightly changes, the overall threshold $\theta_{DA,j}^n$ should be increased to suppress spike firing, thereby mitigating underwater noise interference. In this case, the AET increases its threshold $\theta_{E,j}^n$ due to the rise in membrane potential caused by noise, but due to minor membrane potential fluctuations caused by underwater noise, the depolarization rate between the two moments is close to 0, so $\theta_{D,j}^n$

remains a relatively constant value. The RGT continues to consider the impact of past threshold change rates and avoids excessive threshold fluctuations to maintain homeostasis. Consequently, the whole model can enhance the adaptation to varying external conditions, with homeostasis and strong generalization.

In the biological nervous system, the owl can predict spike firing based on the membrane potential in the inferior colliculus at the previous moment [51]. Inspired by this biological observation, the AET correlates the dynamic adaptive threshold with average membrane potential, which enhances information transmission among neurons. For the j -th neuron in the n -th layer, the AET is detailed as follows:

$$\theta_{E,j}^n(t) = \alpha \cdot (V_m^n(t) - v_j^n(t)) + V_T^n(t) + \log \left(1 + e^{\frac{V_m^n(t) - v_j^n(t)}{k_j}} \right) \quad (12)$$

$$V_m^n(t) = \mu(v_j^n(t)) - e_1 \cdot \gamma_1 \quad (13)$$

$$\gamma_1 = \max(v_j^n(t)) - \min(v_j^n(t)) \quad \text{for } j = 1, 2, \dots, k^n \quad (14)$$

$$V_T^n(t) = \mu(\theta_j^n(t)) - e_2 \cdot \gamma_2 \quad (15)$$

$$\gamma_2 = \max(\theta_j^n(t)) - \min(\theta_j^n(t)) \quad \text{for } j = 1, 2, \dots, k^n \quad (16)$$

where $v_j^n(t)$ denotes the postsynaptic membrane potential of the spiking neuron, k^n and represents the total number of neurons in the n -th layer. α , e_1 , e_2 , and k_j are the hyper-parameters. Equations (13)–(16) are utilized to adjust $V_m^n(t)$ and $V_T^n(t)$ in Equation (12). Specifically, we model $V_m^n(t)$ as the mean of the membrane potentials of the neurons in the layer n , which is shifted by the bias term $e_1\gamma_1$, that is $e_1(\max(v_j^n(t)) - \min(v_j^n(t)))$. This term is based on the range of membrane potentials of all neurons in this layer, as shown in Equations (13) and (14). Through this approach, the dynamic threshold model can be associated with the membrane potentials of all other neurons in the same layer, thereby enhancing information transmission between neurons. Additionally, the bias term is employed to adjust its sensitivity to the layer-wise potential range. $V_T^n(t)$ and $V_m^n(t)$ adopt a similar approach. We use the threshold potential data $\theta_j^n(t)$ instead of the membrane potential data, thereby enhancing the intrinsic connections between neurons within the same layer, as shown in Equations (15) and (16).

Biological observations reveal that the magnitude of the firing threshold in spiking neurons is negatively correlated with the preceding rate of depolarization [52]. Hence, we propose the DDT, which for the j -th neuron in the n -th layer, is defined as follows:

$$\theta_{D,j}^n(t+1) = \tau_d \cdot e^{-\frac{(v_j^n(t+1) - v_j^n(t))}{\tau_c}} \quad (17)$$

where $v_j^n(t+1) - v_j^n(t)$ denotes the depolarization. Higher depolarization, $v_j^n(t+1) - v_j^n(t) > 0$, leads to a lower DDT, while lower depolarization, $v_j^n(t+1) - v_j^n(t) < 0$, results in a higher DDT. τ_d and τ_c represent the decay factor and the decay constant for sensitivity to membrane potential changes, respectively. We connect the DDT to the voltage of the layer-wise firing threshold, which facilitates enhanced information exchange between layers in the SNN.

If the instantaneous variations in long-term neuronal activity patterns are not considered in a biological system, it may lead to excessive threshold fluctuations and potentially amplify noise, which affects stability and reliability. Consequently, the RGT is introduced to evaluate the current dynamic threshold by utilizing the historical rate of change in the dynamic threshold, which ensures that neurons not only rely on the current state but also

incorporate memory of past states. For the j -th neuron in the n -th layer, the RGT is detailed as follows:

$$\theta_{G,j}^n(t+1) = \frac{1}{N} \sum_{i=0}^{N-1} \alpha^i G_j^n(t-i) \quad (18)$$

$$G_j^n(t) = \frac{\theta_j^n(t+1) - \theta_j^n(t)}{\Delta t} \quad (19)$$

where $\theta_j^n(t)$ refers to the current threshold of the j -th neuron. α^i indicates the decay factor, and i indicates the temporal index offset. $G_j^n(t)$ denotes the instantaneous rate of change in the threshold of the j -th neuron. N denotes the window size.

As shown in Figure 6, the firing patterns of spiking neurons with a static threshold and dynamic adaptive threshold (BDAT) are exhibited, respectively.

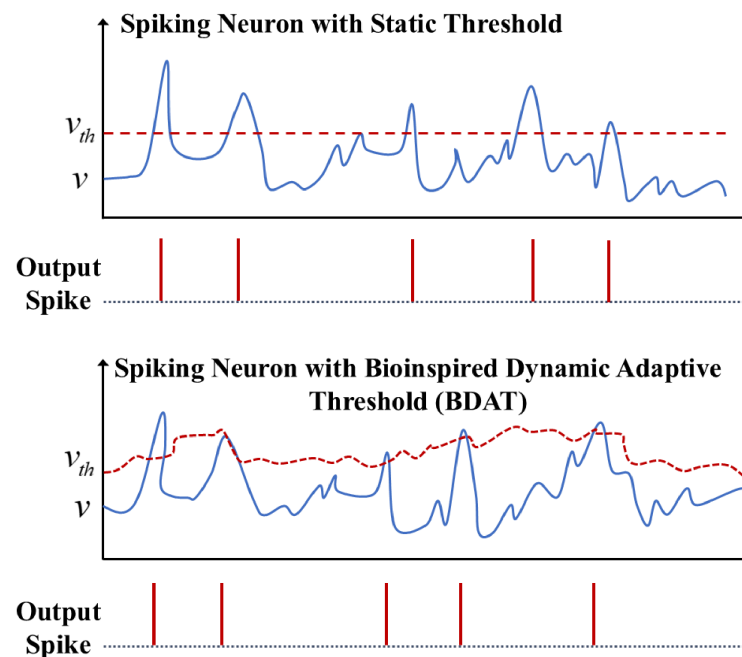


Figure 6. Spiking neuron with static threshold and BDAT.

4.3. Spiking Encoder and Spiking Decoder

The original observation state S_i obtained through the interaction between the AUV and the underwater environment is expressed in continuous real values, which cannot be directly input into the spiking neural network, SDAAN, for training. Hence, the spiking encoder is designed to convert S_i into state spike sequence S_{in} .

$$S_{in} = \text{Spiking_Encoder}(S_i) \quad (20)$$

The spiking encoder performs normalization and Poisson distribution operations on S_i . The normalization operation encodes each data point in $S_i = \{S_{dis}, S_{dir}, S_v, S_w, S_{so}\}$ within the range $[0, 1]$. Since the spike sequence cannot represent negative values, the settings of the angular spike probability S_{dir} and angular velocity spike probability S_w are similar, and need to be split into two parts, i.e., the left direction and the right direction. The original observation state $S_i = \{S_{dis}, S_{dir}, S_v, S_w, S_{so}\}$ is processed into the spike state probability $S_{-}S_i = \{S_{-}S_{dis}, S_{-}S_{dir}, S_{-}S_v, S_{-}S_w, S_{-}S_{so}\}$, as shown below:

$$S_{-}S_{dis} = \text{clip}\left(\frac{S_{dis_min}}{S_{dis}}, 0, 1\right) \quad (21)$$

$$S_{dir} = \begin{cases} S^-_{dir} = -\frac{S_{dir}}{\pi}, S^+_{dir} = 0 & \text{if } S_{dir} < 0 \\ S^+_{dir} = \frac{S_{dir}}{\pi}, S^-_{dir} = 0 & \text{if } S_{dir} \geq 0 \end{cases} \quad (22)$$

$$S_{S_v} = \frac{S_v}{S_{v_max}} \quad (23)$$

$$S_{S_w} = \begin{cases} S^-_{S_w} = -\frac{S_w}{\pi}, S^+_{S_w} = 0 & \text{if } S_w < 0 \\ S^+_{S_w} = \frac{S_w}{\pi}, S^-_{S_w} = 0 & \text{if } S_w \geq 0 \end{cases} \quad (24)$$

$$S_{S_{so}} = \frac{S_{so_min}}{S_{so}} \quad (25)$$

where S_{dis} and S_{dir} represent the Euclidean distance and angular direction from the AUV to the target point, respectively. S_v denotes the linear velocity and S_w represents the angular velocity. S_{so} represents the distance information returned by the sonar detecting the obstacle. S_{dis_min} denotes the minimum distance between the AUV and the target point. S^-_{dir} and S^+_{dir} represent the spike probability for the left angular direction and the spike probability for the right angular direction, respectively. S_{v_max} denotes the maximum velocity. $S^-_{S_w}$ and $S^+_{S_w}$ represent the spike probability for the leftward angular velocity and the spike probability for the rightward angular velocity, respectively. S_{so_min} denotes the minimum distance returned by the sonar for detecting obstacles.

Then, S_{S_i} is sampled by the Poisson distribution. After T iterations of the method, it is encoded as state spike sequence S_{in} with time step T , which is input into the SDAAN. After processing by the network, the corresponding action result A_{out} is output. The spiking decoder is employed for data conversion. The action spike sequence is averaged in the time dimension to obtain the action spike probability, which is then converted into the output action A_i suitable for the AUV, as shown below:

$$A_{out} = F_{SDAAN}(S_{in}|\zeta) \quad (26)$$

$$A_i = Spiking_Decoder(A_{out}) \quad (27)$$

To ensure the AUV performs underwater operations in safe and reasonable ways, the reward function R_{AUV} is set as follows:

$$R_{AUV} = 0.5(R_{dis} + R_{vel}) \quad (28)$$

$$R_{dis} = \begin{cases} R_{goal} & \text{if } D_{AUV_goal} < S_{goal_th} \\ R_{obs} & \text{if } D_{AUV_obs} < S_{obs_th} \\ d \cdot (D_{AUV_goal}(t-1) - D_{AUV_goal}(t)) & \text{otherwise} \end{cases} \quad (29)$$

$$R_{vel} = \lambda_1 \cdot V_{AUV}^2 \cdot \cos(\lambda_2 \cdot V_{AUV} \cdot W_{AUV}) - \lambda_3 \quad (30)$$

where R_{goal} represents the reward value when the AUV reaches the target. R_{obs} represents the penalty value for colliding with obstacles. S_{goal_th} and S_{obs_th} represent the reaching threshold and collision threshold, respectively. V_{AUV} and W_{AUV} represent linear velocity and angular velocity, respectively. d is a scaling factor. λ_1 , λ_2 and λ_3 represent constants. D_{AUV_obs} is defined as the collection of distances from the AUV center to all obstacles, from which the minimum distance is taken for comparison with S_{obs_th} . $D_{AUV_goal}(t-1) - D_{AUV_goal}(t)$ is defined as the distance variation between the AUV and the target during the past two moments. Positive values denote approach toward the target, while negative values indicate movement away. The forward propagation of the SDAAN is shown in Algorithm 1.

Algorithm 1 Forward Propagation of SDAAN $F_{SDAAN}(\zeta)$

Input: Original observation state $S_i = \{S_{dis}, S_{dir}, S_v, S_w, S_{so}\}$; maximum timestep T ; network layers N ; number of neurons per layer J ; weight matrix W_n ; bias vector b_n ; normal networks parameters ζ and θ ; target network parameters ζ' and θ' .

Output: Actual conversion action A_i .

- 1: Interaction between the AUV and the underwater environment to acquire the original observation state $S_i = \{S_{dis}, S_{dir}, S_v, S_w, S_{so}\}$.
- 2: The original observation state S_i is converted into spike state sequence S_{in} through normalization and Poisson sampling within the spiking encoder, serving as input to $F_{SDAAN}(\zeta)$:
 $S_{in} = \text{Spiking_Encoder}(S_i)$
- 3: Propagate $F_{SDAAN}(\zeta)$ with time t and layer n recurrently:
- 4: **for** $t = 1, \dots, T$ **do**
- 5: $S_1^t = S_{in,n}^t$
- 6: **for** $n = 1, \dots, N$ **do**
- 7: **for** $j = 1, \dots, J$ to neuron number of layer n **do**
- 8: Compute AET $\theta_{E,j}^n$, DDT $\theta_{D,j}^n$, and RGT $\theta_{G,j}^n$.
- 9: Deploy the BDAT $\theta_{DA,j}^n$ to each spiking neuron.
- 10: $\theta_{DA,j}^n(t+1) = \lambda \cdot \theta_{E,j}^n(t) + \beta \cdot \theta_{D,j}^n(t+1) + \gamma \cdot \theta_{G,j}^n(t+1)$
- 11: $C_n^t = \eta_c \cdot C_n^{t-1} + W_n \cdot O_{n-1}^t + b_n$
- 12: $U_n^t = \eta_v \cdot U_n^{t-1} \cdot \chi(O_n^{t-1}) + C_n^t$
- 13: $\chi(O_n^{t-1}) = 1 - O_n^{t-1}$
- 14: $O_n^t = h(U_n^t) = H(U_n^t - \theta_{DA,j}^n)$
- 15: **end for**
- 16: **end for**
- 17: $A_{out} = F_{SDAAN}(S_{in}|\zeta)$
- 18: **end for**
- 19: The output A_{out} from the SDAAN $F_{SDAAN}(\zeta)$ is processed and mapped by the spiking decoder to obtain A_i :
 $A_i = \text{Spiking_Decoder}(A_{out})$

4.4. Training of the Proposed BDAT-Planner

The training process of BDAT-Planner is shown in Algorithm 2. Among these, $F_{SDAAN}(\zeta)$ and $F_{DCNN}(\theta)$ are trained by randomly selecting minibatch transitions $\{S_k, A_k, R_k, S_{k+1}\}$ from the experience replay buffer. The corresponding target networks $F_{TSDAAN}(\zeta')$ and $F_{TDCNN}(\theta')$ are set up based on $F_{SDAAN}(\zeta)$ and $F_{DCNN}(\theta)$, respectively. $L_{TD}(\theta)$ and $L_Q(\zeta)$ are utilized to train the model, where γ represents a discount factor, as shown below:

$$L_{TD}(\theta) = (R_k + \gamma Q_{k+1} - Q_k)^2 \quad (31)$$

$$\begin{aligned} Q_{k+1} &= F_{TDCNN}(S_{k+1}, A_{k+1}|\theta') \\ &= F_{TDCNN}(S_{k+1}, \text{Spiking_Decoder}(F_{TSDAAN}(\text{Spiking_Encoder}(S_{k+1}|\zeta')))|\theta') \end{aligned} \quad (32)$$

$$\begin{aligned} Q_k &= F_{DCNN}(S_k, A_k|\theta) \\ &= F_{DCNN}(S_k, \text{Spiking_Decoder}(F_{SDAAN}(\text{Spiking_Encoder}(S_k|\zeta))))|\theta \end{aligned} \quad (33)$$

$$\begin{aligned} L_Q(\zeta) &= -Q'_k \\ &= -F_{DCNN}(S_k, \text{Spiking_Decoder}(F_{SDAAN}(\text{Spiking_Encoder}(S_k|\zeta))))|\theta \end{aligned} \quad (34)$$

The corresponding target networks $F_{TSDAAN}(\zeta')$ and $F_{TDCNN}(\vartheta')$ are updated by the soft update method, where η is a constant, as shown below:

$$\begin{aligned}\zeta' &= \eta\zeta + (1 - \eta)\zeta' \\ \vartheta' &= \eta\vartheta + (1 - \eta)\vartheta'\end{aligned}\quad (35)$$

Algorithm 2 Training Process of BDAT-Planner

Input: Spiking dynamic adaptive actor network (SDAAN) $F_{SDAAN}(\zeta)$; deep critic normal network (DCNN) $F_{DCNN}(\vartheta)$; target spiking dynamic adaptive actor network (TSDAAN) $F_{TSDAAN}(\zeta')$; target deep critic normal network (TDCNN) $F_{TDCNN}(\vartheta')$.

Output: TD loss $L_{TD}(\vartheta)$; Q loss $L_Q(\zeta)$.

- 1: Randomly initialize spiking dynamic adaptive actor network $F_{SDAAN}(\zeta)$ and deep critic normal network $F_{DCNN}(\vartheta)$ with ζ and ϑ respectively.
 - 2: Initialize the target spiking dynamic adaptive actor network $F_{TSDAAN}(\zeta')$ and the target deep critic normal network $F_{TDCNN}(\vartheta')$ with $\zeta' \leftarrow \zeta$ and $\vartheta' \leftarrow \vartheta$.
 - 3: Initialize experience replay buffer.
 - 4: **For** $o = 1, \dots, O$ **do**
 - 5: Acquire original observation state S_i .
 - 6: **For** $i = 1, \dots, I$ **do**
 - 7: Obtain action A_i according to the current policy, spiking encoder, and spiking decoder:
 $A_i = \text{Spiking_Decoder}(F_{SDAAN}(\text{Spiking_Encoder}(S_i)))$
 - 8: Execute action A_i , get the reward R_i , and the new state S_{i+1} .
 - 9: Store $\{S_i, A_i, R_i, S_{i+1}\}$ into the experience replay buffer.
 - 10: Sample random minibatch transitions $\{S_k, A_k, R_k, S_{k+1}\}$ from the experience replay buffer.
 - 11: Update the critic network by minimizing $L_{TD}(\vartheta)$:
 $L_{TD}(\vartheta) = (R_k + \gamma Q_{k+1} - Q_k)^2$
 - 12: Update the actor network by maximizing $L_Q(\zeta)$:
 $L_Q(\zeta) = -Q'_k$
 - 13: Update the corresponding target network $F_{TSDAAN}(\zeta')$ and $F_{TDCNN}(\vartheta')$:
 $\zeta' = \eta\zeta + (1 - \eta)\zeta'$
 $\vartheta' = \eta\vartheta + (1 - \eta)\vartheta'$
 - 14: **end for**
 - 15: **end for**
-

Policy gradient backpropagation for the BDAT-Planner is executed by the spatial-temporal backpropagation algorithm (STBP) [53]. To address the issue of the spiking non-differentiable, the gradient is approximated by the pseudo-gradient function, where ρ is a non-zero constant, as shown below:

$$\dot{h}(U_n^t) = \frac{1}{\rho} H\left(\left|U_n^t - \theta_{DA,j}^n\right| - \frac{\rho}{2}\right) \quad (36)$$

To describe the gradient descent process, the backpropagation process is divided into four cases. a^{n+1} denotes the output of the $(n+1)$ -th layer of the critic network before the non-linear activation function, and W_c^{n+1} denotes its weights in the $(n+1)$ -th layer, as shown in the following equation:

(1) At the last time $t = T$ and the last layer $n = N$, $SC(T)$ represents the sum of output spikes by the actor network at time T. This is conducted for the output layer N, as shown below:

$$\nabla_{SC(T)} Loss = \frac{1}{T} \cdot \nabla_{Action} Loss = \frac{1}{T} \cdot W_c^{n+1'} \cdot \nabla_{a^{n+1}} Loss \quad (37)$$

$$\nabla_{s^N(T)} Loss = \nabla_{SC(T)} Loss \quad (38)$$

(2) At the last time $t = T$ and the previous layers $n < N$. For each layer $n \in [1, N)$, as shown in the following equation:

$$\nabla_{v^n(T)} Loss = \dot{h}(v^n(t)) \cdot \nabla_{s^n(T)} Loss \quad (39)$$

$$\nabla_{s^{n-1}(T)} Loss = W_a^{n'} \cdot \nabla_{v^n(T)} Loss \quad (40)$$

(3) At the previous times, $t < T$ and the last layer $n = N$. The output layer N , for each time $t \in [1, T)$, is shown below:

$$\nabla_{SC(t)} Loss = \nabla_{SC(t+1)} Loss = \frac{1}{T} \cdot \nabla_{Action} Loss = \frac{1}{T} \cdot W_c^{n+1'} \cdot \nabla_{a^{n+1}} Loss \quad (41)$$

$$\nabla_{s^N(t)} Loss = \nabla_{SC(t)} Loss \quad (42)$$

(4) At the previous times $t < T$ and the previous layers $n < N$. For each layer $n \in [1, N)$ and each time $t \in [1, T)$, η_v denotes the membrane potential decay factor, as shown in the following equation:

$$\nabla_{v^n(t)} Loss = \dot{h}(v^n(t)) \cdot \nabla_{s^n(t)} Loss + \eta_v \cdot (1 - s^n(t)) \cdot \nabla_{v^n(t+1)} Loss \quad (43)$$

$$\nabla_{s^{n-1}(t)} Loss = W_a^{n'} \cdot \nabla_{v^n(t+1)} Loss \quad (44)$$

Network parameters are updated every T time steps. The gradient of the loss function with respect to the parameters $\nabla_{W^n} Loss$ and $\nabla_{b^n} Loss$ of the SNN is expressed as follows:

$$\nabla_{W^n} Loss = \sum_{t=1}^T s^{n-1}(t) \cdot \nabla_{v^n(t)} Loss \quad (45)$$

$$\nabla_{b^n} Loss = \sum_{t=1}^T \nabla_{v^n(t)} Loss \quad (46)$$

5. Experimental Discussion and Analysis

5.1. Experiment Setup

The simulations in this paper are carried out under the Ubuntu 18.04 system with ROS–Gazebo serving as the simulator, and are executed on a system equipped with NVIDIA GeForce RTX 4080 and AMD Ryzen 7 5700X 8-core CPU. The experimental configuration is detailed in Table 1. The proposed BDAT-Planner is trained using curriculum learning [54]. We have set up four training environments in total, as shown in Figure 7. The gray cylinder represents the dynamic obstacle, and the yellow arrow indicates its direction of movement. Each cylinder moves repeatedly between two fixed points, and the distance and speed of its motion are set to provide sufficient space and time for the AUV to pass through. We sequentially set 100, 200, 300, and 400 training episodes in four training environments, with progressively increasing complexity, and randomly generated start points and target points.

Table 1. Specific Experimental configuration.

Name of Software and Hardware	Specific Model and Parameters
Operation System	Ubuntu 18.04
CPU	AMD Ryzen 7 5700X 8-Core
GPU	NVIDIA GeForce RTX 4080
Simulation System	ROS Melodic
Simulator	Gazebo
Development Languages	Python 3.6 and C++ 11
Training Framework	Pytorch-GPU 1.2

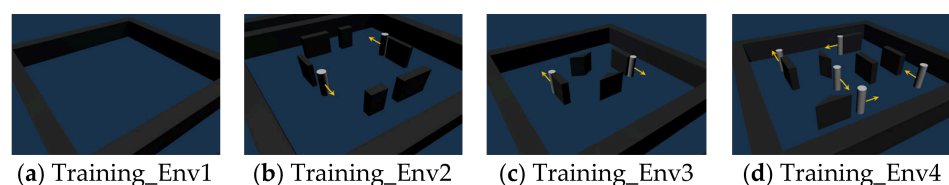


Figure 7. Four training environments with static and dynamic obstacles: (a) Training_Env1; (b) Training_Env2; (c) Training_Env3; (d) Training_Env4.

The hyperparameters for the BDAT-Planner are shown in Table 2. After initializing the collision avoidance model, due to the small number of training steps at the beginning, it is difficult to rely on randomly initialized parameters for action decisions. It is also difficult to avoid the model getting stuck on a local optimum early on. Consequently, the random exploration strategy κ is introduced, whereby only 10% of the initial decision actions are derived from the output of the actor network, while the remaining 90% are random values, thereby enhancing the initial exploration capabilities of the model. With continuous network updates, decision actions increasingly rely on data from the actor network. The probability of random values is continuously reduced by a discount factor of 0.999, so that ultimately 99% of decision actions come from the output of the actor network, while only 1% are random values. The training environment constructed in this paper consists of four underwater environments with progressively increasing difficulty. After training in simple environments, the model has developed basic collision avoidance capabilities and no longer requires 90% of the intensive random exploration. In subsequent training environments, the initial exploration rate is reduced to 0.6, so that balances strategy utilization and environmental exploration capabilities in complex scenarios. Consequently, during the training stage, the κ range for the four training environments is set to (0.9, 0.01), (0.6, 0.01), (0.6, 0.01), and (0.6, 0.01), respectively. The discount factor for the four environments is set to 0.999.

Table 2. Hyperparameters for the proposed BDAT-Planner.

Hyperparameters	Values
Neurons in each hidden layer of SDAAN	512
Neurons in each hidden layer of DCNN	512
Hidden layers of SDAAN, DCNN	3, 3
Learning rate of SDAAN, DCNN	0.00001, 0.0001
Size of experience replay buffer, batch size	100,000, 256
Reward thresholds (goal_th, obs_th)	0.8, 0.12
Range κ of Training_Env1 to Training_Env4	(0.9, 0.01), (0.6, 0.01), (0.6, 0.01), (0.6, 0.01)
Decay factor of the four training environments	0.999
Weighting parameters λ, β, γ of BDAT	1/3, 1/3, 1/3
Parameters α, k_j, e_1, e_2 of AET	0.01, 3.0, 0.2, 0.2
Parameters τ_c, τ_d of DDT	3.0, 1.0
Parameters α^i, N of RGT	0.95, 10
Time steps T	5

We set up two evaluation environments with only dynamic obstacles and two evaluation environments with mixed static–dynamic obstacles, as shown in Figure 8. The complexity of obstacle distribution varies in all underwater evaluation environments. In the only dynamic obstacle environments, DyEnv1 and DyEnv2, dynamically moving obstacles moving in the same direction and dynamically moving obstacles moving in mixed directions are added, respectively. In DyEnv1, the radius of the dynamic obstacle is 0.25 m, and the movement speed is 0.1 m/s. In DyEnv2, two types of dynamic obstacles with different radii and movement speeds are set. The radius is 0.25 m and 0.4 m, respectively, and the movement speed is 0.1 m/s and 0.2 m/s, respectively. In the mixed static–dynamic obstacle environments, Sta_DyEnv1 and Sta_DyEnv2, the number, shape, and distribution of static obstacles gradually increase in complexity, and the number and movement directions of dynamic obstacles also progressively become more complex. Multiple evaluation environments of varying complexity are established, which facilitates further verification of the adaptability and generalization of the BDAT-Planner in unseen or dynamic situations. Additionally, to avoid introducing random factors during the experiment, we set up four underwater evaluation environments of varying complexity and conducted five rounds of collision avoidance tests in each environment, with each round consisting of 200 trial paths. The start points and target points of each trial path are generated randomly. It helps enhance the reliability and rationality of the experimental results.

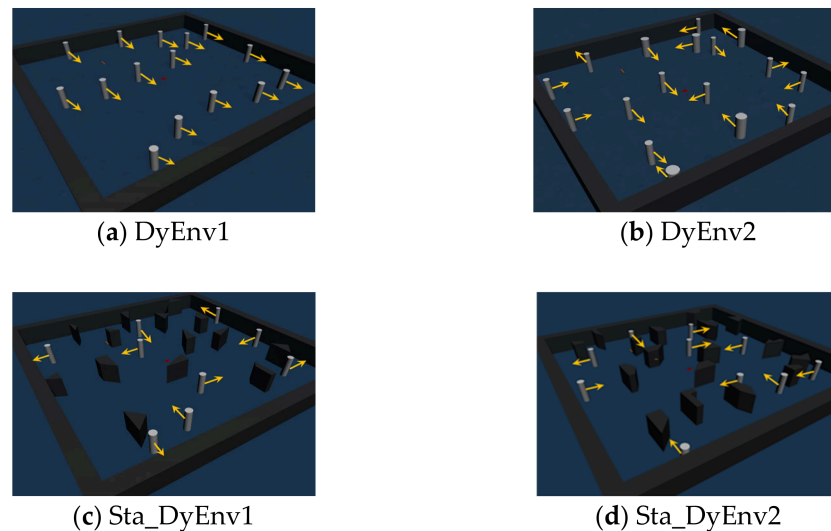


Figure 8. Different evaluation environments in the Gazebo simulator: (a) DyEnv1; (b) DyEnv2; (c) Sta_DyEnv1; (d) Sta_DyEnv2.

5.2. Performance Analysis During the Training Process

To validate the rationality of the chosen weighting parameters (λ , β , γ) in the BDAT mechanism, we have conducted the following experiments to perform sensitivity analysis on these key parameters, comparing the impact of changes in these weights on model training and collision avoidance success rates, as shown in Figure 9 and Table 3. As can be observed, if the weighting parameters are biased toward a particular mechanism, spiking neurons may exhibit unstable threshold firing and dynamic over-regulation, making it difficult to achieve stable spiking output and compromising network stability and reliability. It causes lower collision avoidance success rates and poor performance. When the value of a certain parameter is too large, the dynamic threshold fluctuates too abruptly, which causes training oscillation and results in poor collision avoidance performance. When the value of a certain parameter is too small, the ability to dynamically adjust the threshold is reduced, which leads to slow training convergence and inflexible collision avoidance strategies. In contrast, when λ , β and γ are set to $1/3$, the three are harmonized with each other, and the dynamic threshold adjustment is neither too aggressive nor too sluggish, thus ensuring that the spike firing is stable and reliable. The model exhibits quick convergence during training and yields stable results. Among the four underwater evaluation environments set up in this paper, this parameter combination yields the highest collision avoidance success rate. It has achieved balance between training convergence speed and collision avoidance success rate. Consequently, through sensitivity analysis experiments, it has been validated that the selected weighting parameters are rational and effective.

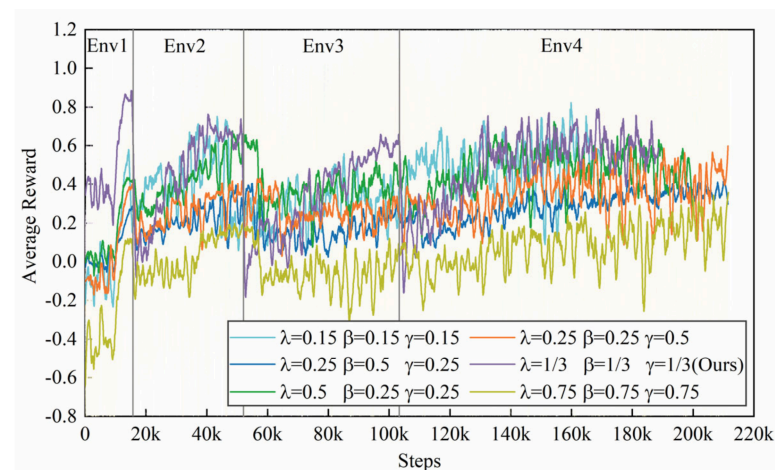


Figure 9. Training results of different weighting parameters of BDAT.

Table 3. Success rates of different weighting parameters in different evaluation environments.

Parameters			Success Rate			
λ	β	γ	DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
0.15	0.15	0.15	82.5%	81%	85%	83.5%
0.25	0.5	0.25	84.5%	82.5%	87%	85%
0.5	0.25	0.25	85%	83%	88.5%	86.5%
1/3	1/3	1/3	88%	86.5%	91.5%	89%
0.25	0.25	0.5	86%	84%	89.5%	87%
0.75	0.75	0.75	81.5%	82%	84.5%	82.5%

Additionally, we have conducted the following experiments to perform sensitivity analysis on multiple parameters, investigating their impact on collision avoidance performance under different values, as shown in Table 4. Among these, α , e_1 , e_2 and k_j are parameters in the AET. α and k_j are two hyperparameters. When α and k_j are set to 0.01 and 3.0, respectively, the method proposed in this paper achieves the highest collision avoidance success rate in different evaluation environments of varying complexity. e_2 and e_2 are employed to balance the contributions of specially designed bias terms. It can be observed that when e_1 and e_2 are set to 0.2 and 0.2, respectively, the proposed method yields the highest collision avoidance success rate, which exhibits optimal effectiveness. Consequently, when α and k_j are set to 0.01 and 3.0, respectively, e_1 and e_2 are selected as 0.2 and 0.2; the most effective and robust collision avoidance performance has been achieved under underwater evaluation environments of different complexities.

Table 4. Success rates of different key parameters in different evaluation environments.

Parameters		Success Rate				
Parameters of AET	α	k_j	DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
	0.005	1.0	84%	82%	86.5%	85%
	0.01	3.0	88%	86.5%	91.5%	89%
	0.03	5.0	87%	84.5%	90%	88%
	0.05	6.0	86.5%	85%	89%	88.5%
	0.1	10.0	83.5%	83%	87%	84%
	e_1	e_2	DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
	0.1	0.1	87%	85%	89.5%	87.5%
	0.2	0.2	88%	86.5%	91.5%	89%
	0.3	0.3	86%	85.5%	89%	86.5%
	0.4	0.4	85%	82.5%	87.5%	85.5%
	0.5	0.5	84.5%	82%	87%	84%
	1.0	1.0	82%	80.5%	83.5%	82.5%
Parameters of DDT	τ_c	τ_d	DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
	1.0	0.2	83.5%	82%	86%	83.5%
	1.5	0.5	85%	83%	88.5%	87.5%
	3.0	1.0	88%	86.5%	91.5%	89%
	4.0	2.0	84%	82.5%	87%	85%
Parameters of RGT	α^i	N	DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
	0.8	5	85%	83.5%	86%	84.5%
	0.95	10	88%	86.5%	91.5%	89%
	0.9	20	86%	84%	87%	85.5%
	0.98	50	87.5%	85%	88%	86%

τ_c and τ_d denote the decay factor and decay constant in the DDT, respectively. τ_c adjusts the sensitivity of the DDT to the continuous temporal changes in the membrane potential of spiking

neurons. It can be seen that when τ_c is set to 3.0 and τ_d is chosen as 1.0, the proposed method achieves the highest collision avoidance success rate in different evaluation environments of varying complexity. Additionally, α^i and N denote parameters in the RGT, and represent the window size and decay factor, respectively. It can be seen that values that are either too large or too small make it difficult to achieve stable and efficient dynamic collision avoidance performance. When α^i is chosen as 0.95 and N is set to 10, the model shows excellent dynamic collision avoidance performance, achieving the highest collision avoidance success rate across various evaluation environments. It can avoid unstable and sensitive dynamic thresholds while maintaining stability and robustness. Through the above experimental setup, we have conducted sensitivity analyses on multiple key parameters in the BDAT, which have validated the efficiency and rationality of the selected key parameters.

We have plotted the average reward curves in four training environments to evaluate the training convergence performance of different methods, as shown in Figure 10. Specifically, DWA [55] is based on the static window assumption, which assumes the environment remains unchanged throughout the entire motion planning process. It fails to adapt to changes in dynamic obstacles, resulting in poor training performance. DDPG performs well in the environment with the simple obstacle distribution, but exhibits unstable performance in the environment with complex dynamic obstacles, with no overall convergence. Compared to DDPG, the other methods perform well in the four training environments, which demonstrates that introducing an SNN into the AUV for collision avoidance is reasonable. Among them, SDDPG [33] and SPPO [56] only combine an SNN and DRL, employing the static threshold for spiking neurons. In the underwater dynamic obstacle environment, significant fluctuations emerged during the later stages of training, which resulted in suboptimal training performance. Although DT1 [48] and DT2 [49] consider the importance of dynamic thresholds in spiking neurons, the obtained reward values are relatively lower. With the increase in dynamic obstacles in the new training environments, particularly in Training_Env3 and Training_Env4, the reward curve shows significant fluctuations, leading to unstable performance.

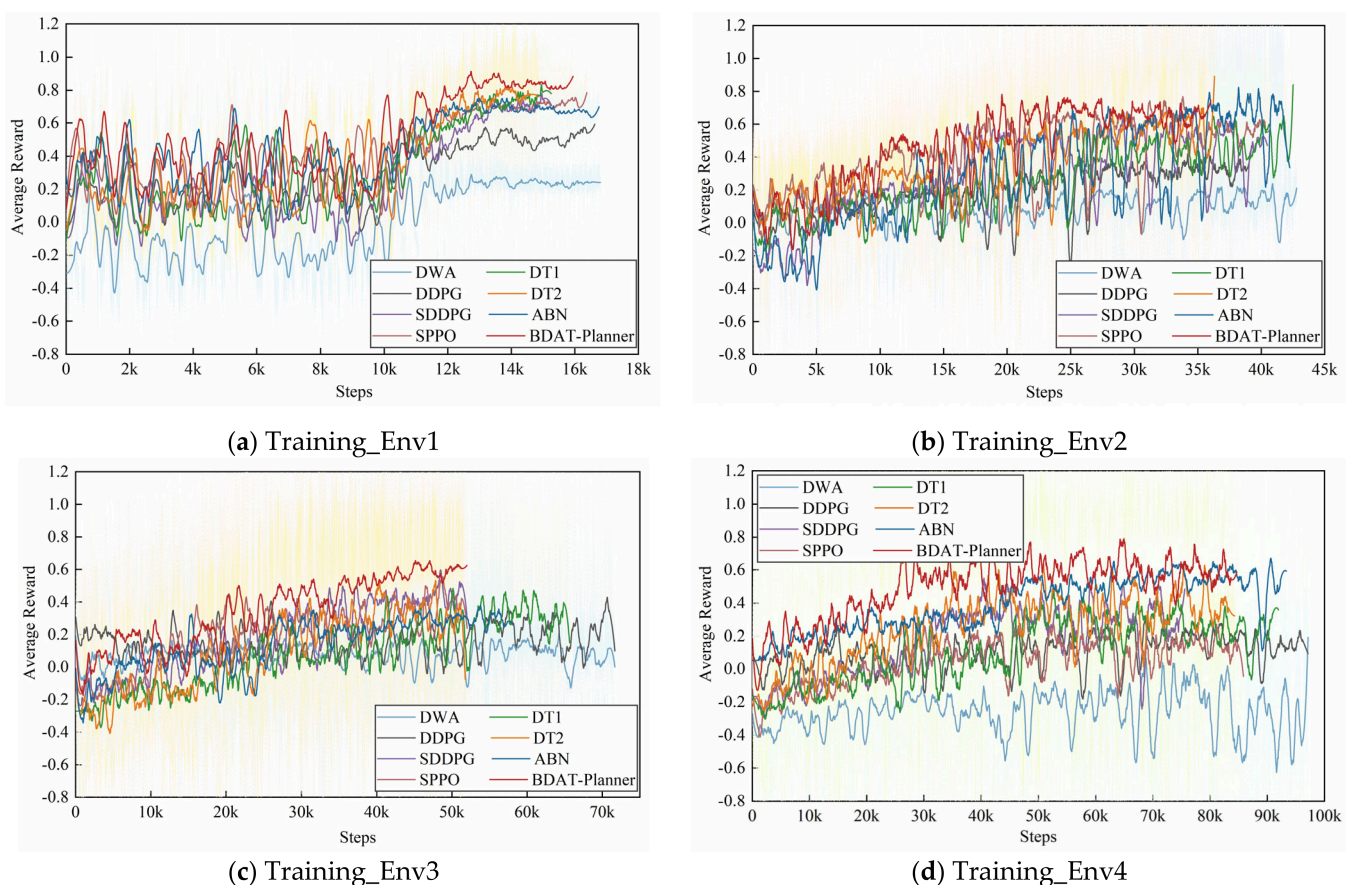


Figure 10. Average reward of different methods during the training process: (a) Training_Env1; (b) Training_Env2; (c) Training_Env3; (d) Training_Env4.

Although ABN [57] outperforms the above comparison methods in four training environments, it requires more iterative steps to converge, which results in slower training convergence. In the latter three training environments, reward value fluctuations still occur, which cannot fully adapt to changes in complex dynamic environments, resulting in poor generalization performance. In contrast, the proposed BDAT-Planner can rapidly learn a suitable dynamic collision avoidance policy, converge quickly, and achieve stable training results. After learning collision obstacle policy in this environment, the method remains adaptable to complex dynamic changes in new environments, which demonstrates strong robustness and generalization in various underwater dynamic environments.

To validate the reasonableness of the time step T , we have conducted a comparative analysis of the training time for the proposed BDAT-Planner method under three different time steps ($T = 5, 10, 15$), as shown in Figure 11a, and the success rates under three different time steps in different evaluation environments, as shown in Figure 11b. The temporal channel number of the spiking model is T , meaning the model has T time steps, i.e., each decision made by the AUV consists of time windows of size T . It can be observed that, during the training stage, as the time step T increases, the training time also gradually increases. In different evaluation environments, the collision avoidance success rate also improves to some extent as the time step T increases. Nevertheless, when T is large, although the collision avoidance success rate increases, it comes at the cost of higher computational costs and longer training times. Consequently, after taking into account the computational cost, training time, and collision avoidance performance of the task, we finally select the time step size T as five. Even with smaller time step, the model still achieves efficient and stable dynamic collision avoidance performance.

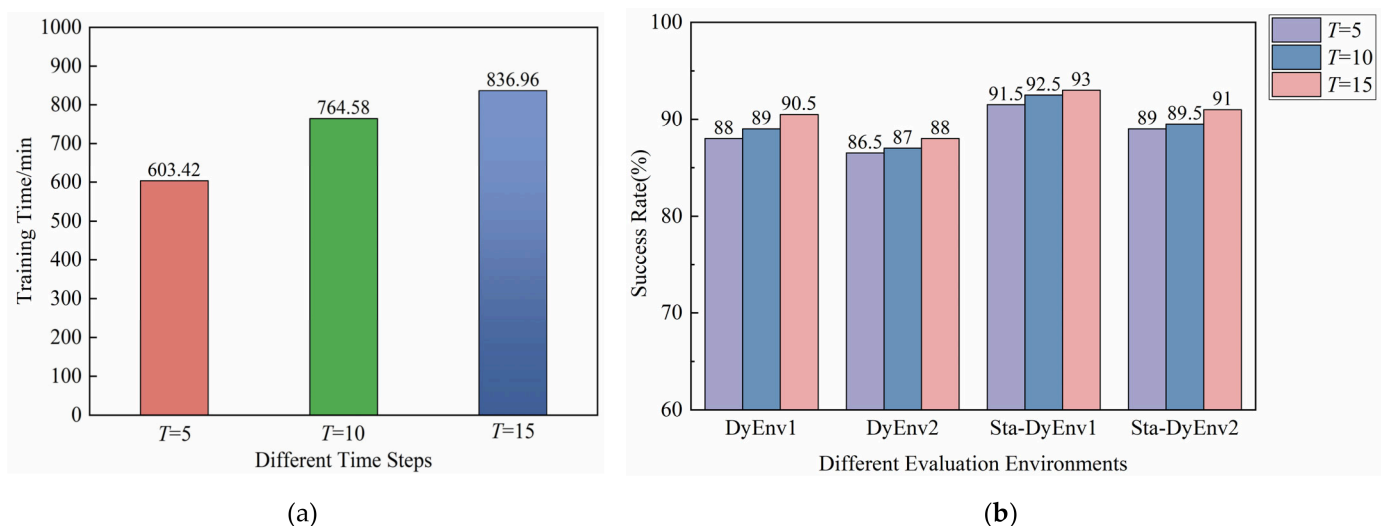


Figure 11. Training time for BDAT-Planner at different time steps and success rates in different evaluation environments at different time steps. (a) Training time. (b) Success rate.

5.3. Performance Analysis During the Evaluation Process

Success rate, collision rate, and overtime percentage are adopted as the primary evaluation metrics to evaluate the underwater dynamic collision avoidance abilities of all methods. Success rate refers to the percentage of times the AUV reaches the target point without colliding with any static or dynamic obstacles out of the total number. Collision rate refers to the percentage of collisions between the AUV and static or dynamic obstacles out of the total number of collisions. Overtime percentage denotes the percentage of times within specified steps where no collision occurred, and the target point is not reached, relative to the total number.

The ablation studies are conducted to validate the effectiveness of each component and pairwise component combinations of the BDAT-Planner proposed in this paper. Table 5 shows the ablation experiment results in different underwater evaluation environments. Among them, the baseline refers to the method that employs neither the AET, DDT, nor the RGT, which means that the spiking neuron adopts the static threshold. Only_AET, Only_DDT, and Only_RGT represent methods employing only a single component. AET+DDT, AET+RGT, and DDT+RGT represent methods of pairwise

component combinations. As can be seen, the results from the different evaluation environments in the table have validated that the proposed BDAT-Planner achieves better underwater dynamic collision avoidance performance than any single component or pairwise component combination. Methods based on a single component have improved collision avoidance performance to some extent, but their adaptability to complex dynamic environments and generalizability remain limited. Although the performance of the method using pairwise component combinations has improved, it still fails to achieve optimal collision avoidance performance. In DyEnv1 and DyEnv2, the success rate of BDAT-Planner increased by 10.5% and 10%, compared to the baseline, respectively. In Sta_DyEnv1 and Sta_DyEnv2, the proposed method achieved success rates 10% and 8.5% higher than the baseline, respectively. Consequently, the BDAT-Planner that combines the AET, DDT, and RGT can rationally adjust the threshold of spiking neurons, exhibits strong adaptability to various external dynamic obstacles, and provides optimal dynamic collision avoidance capabilities for AUVs in complex and unknown underwater environments.

Table 5. Results of ablation studies in different underwater evaluation environments.

Evaluation Environments	Metrics	Baseline	Only_AET	Only_DDT	Only_RGT	AET+ DDT	AET+ RGT	DDT+ RGT	BDAT_Planner
DyEnv1	Success Rate	77.5%	82.5%	83.5%	81.5%	86%	84.5%	85%	88%
	Collision Rate	22.5%	17.5%	16.5%	18.5%	14%	15.5%	15%	12%
DyEnv2	Success Rate	76.5%	82%	81.5%	81%	83%	84%	84.5%	86.5%
	Collision Rate	23.5%	18%	18.5%	19%	17%	16%	15.5%	13.5%
Sta_DyEnv1	Success Rate	81.5%	86.5%	85.5%	87%	88%	89.5%	89%	91.5%
	Collision Rate	15.5%	11%	12%	11%	8.5%	8%	8.5%	6.5%
	Overtime Percentage	3%	2.5%	2.5%	2%	3.5%	2.5%	2.5%	2%
Sta_DyEnv2	Success Rate	80.5%	82%	84%	84.5%	85.5%	86%	87%	89%
	Collision Rate	12%	11.5%	12%	12%	9%	9.5%	8.5%	7.5%
	Overtime Percentage	7.5%	6.5%	4%	3.5%	5.5%	4.5%	4.5%	3.5%

Table 6 presents the experimental results of different contrast methods in two dynamic obstacle environments and two mixed static–dynamic obstacle environments. DWA is a non-learning traditional method. DDGP is an ANN-based method, while the others are SNN-based methods. Among them, SDDPG and SPPO employ the static threshold in spiking neurons, while DT1, DT2, ABN, and the proposed BDAT-Planner all utilize the dynamic threshold. The experimental results reveal that DWA has the poorest collision avoidance performance and is prone to falling into local optima in complex dynamic obstacle environments. When new obstacles occur, the trajectory planning may not adjust in time, potentially causing the AUV to collide with the obstacle. In different evaluation environments, the ANN-based DDGP method achieves relatively low success rates, with higher average path distance and average path time. In contrast, the SNN leverages properties such as spiking-addition data flow and spatial–temporal information processing, which yield better results when applied to dynamic collision avoidance tasks, exhibiting both rationality and reliability. SDDPG and SPPO both adopt the static threshold, where the firing threshold of spiking neurons is a fixed constant. Although SPPO employs higher-performance modern actor–critic variants, during continuous learning and prolonged operation, neurons may exhibit excessive excitement or negativity due to abnormally high or low firing rates over a certain period, which makes it difficult to maintain stable and efficient performance, resulting in poor collision avoidance capabilities.

Compared to the static threshold, the dynamic threshold is more bioinspired and better aligns with the properties of actual biological neurons. We dynamically adjust the firing rate based on threshold changes in neurons to maintain the homeostasis of the model. Among these, although the DT1 method employs the dynamic scaling coefficient, it remains constrained by fixed constants, which makes it difficult to achieve satisfactory results in dynamic obstacle environments. The success rates in the four dynamic evaluation environments are only 81%, 80.5%, 85%, and 83.5%, respectively. The DT2 method adopts the dynamic threshold to maintain the neuronal firing rate at a desired

constant target frequency, which limits the number of neurons expected to fire in each layer and restricts the adaptability of the model. The success rates in the four dynamic evaluation environments are only 82%, 80%, 86%, and 82.5%, respectively. The ABN method automatically adjusts the spiking firing mechanism by utilizing the membrane gradient, retrospective gradient, and spike efficiency. Compared to the DT1 method and the DT2 method, collision avoidance accuracy has improved slightly. Nevertheless, in underwater environments with dense dynamic obstacles, it still struggles to adapt promptly to changes in moving obstacles, resulting in higher average path distance and average path time. In contrast, the BDAT-Planner proposed in this paper has achieved success rates of 88% and 86.5% in DyEnv1 and DyEnv2, respectively, and 91.5% and 89% in Sta_DyEnv1 and Sta_DyEnv2, respectively, attaining the highest success rates. Simultaneously, under the premise of safety and non-collision, underwater route planning is more efficient, without unnecessary detours, resulting in shorter path distance and reduced path time consumption.

Table 6. Comparison results of different methods in different evaluation environments.

Evaluation Environments	Metrics	DWA	DDPG	SDDPG	SPPO	DT1	DT2	ABN	BDAT-Planner
DyEnv1	Success Rate	74.5%	76%	77.5%	79%	81%	82%	85.5%	88%
	Collision Rate	25.5%	24%	22.5%	21%	19%	18%	14.5%	12%
	Average Path Distance (m)	15.31	15.17	14.96	14.99	14.68	14.61	14.23	13.96
	Average Path Time (s)	38.39	38.13	37.97	37.89	37.68	37.04	36.94	36.58
DyEnv2	Success Rate	71%	73.5%	76.5%	78%	80.5%	80%	83.5%	86.5%
	Collision Rate	29%	26.5%	23.5%	22%	19.5%	20%	16.5%	13.5%
	Average Path Distance (m)	15.81	15.62	15.21	15.13	14.99	15.03	14.90	14.57
	Average Path Time (s)	42.24	41.89	39.57	39.06	38.65	39.11	38.91	38.16
Sta_DyEnv1	Success Rate	77%	80%	81.5%	82%	85%	86%	88.5%	91.5%
	Collision Rate	19.5%	17.5%	15.5%	17.5%	11%	11%	9.5%	6.5%
	Overtime Percentage	3.5%	2.5%	3%	0.5%	4%	3%	2%	2%
	Average Path Distance (m)	16.74	16.32	16.17	15.81	15.68	15.47	14.89	14.61
	Average Path Time (s)	47.28	46.32	45.15	44.76	44.41	43.02	42.23	42.58
Sta_DyEnv2	Success Rate	74%	78%	80.5%	81.5%	83.5%	82.5%	85%	89%
	Collision Rate	20%	12.5%	12%	12.5%	12%	12%	8.5%	7.5%
	Overtime Percentage	6%	9.5%	7.5%	6%	4.5%	5.5%	6.5%	3.5%
	Average Path Distance (m)	17.02	16.48	15.92	15.67	15.09	15.23	14.58	14.78
	Average Path Time (s)	54.03	53.62	53.11	52.84	52.37	52.67	52.09	51.87

The dynamic collision avoidance performance of the proposed BDAT-Planner and other methods is evaluated in four different underwater evaluation environments. The visualized trajectories are shown in Figures 12–19. The successful trajectory is shown by the blue solid line, the collision trajectory by the red solid line, and the timeout trajectory by the purple solid line. The start point and target point are represented by the green triangle and red dot, respectively. The yellow arrow indicates the direction of movement for the dynamic obstacle.

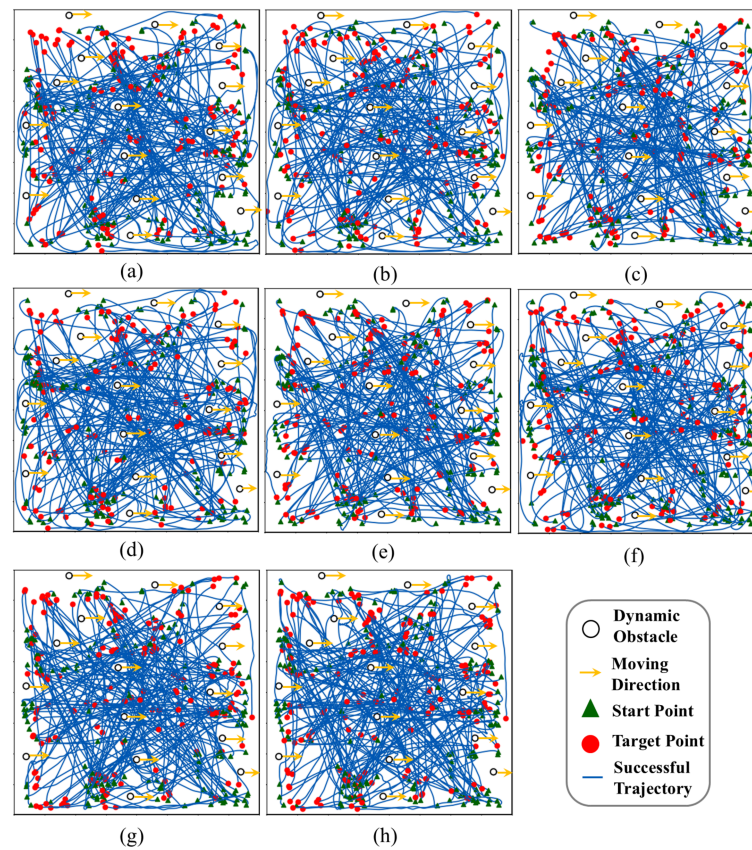


Figure 12. Success trajectories in DyEnv1: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

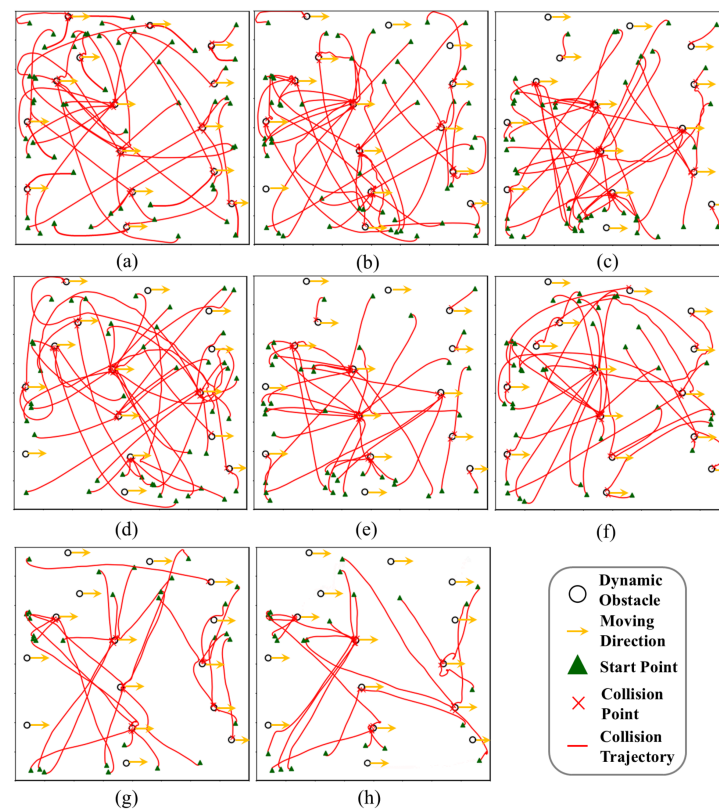


Figure 13. Failure trajectories in DyEnv1: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

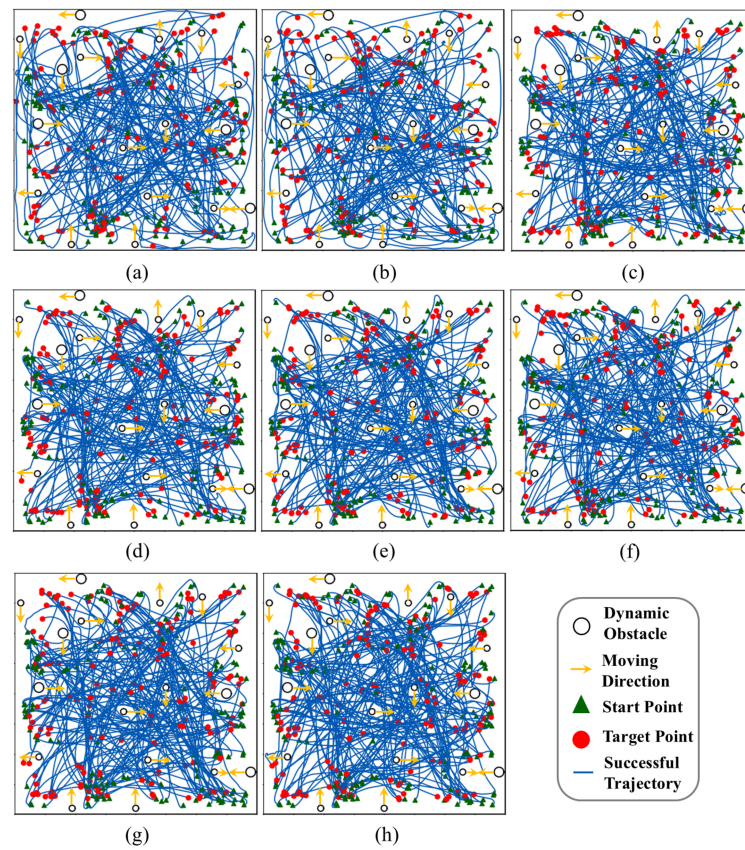


Figure 14. Success trajectories in DyEnv2: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

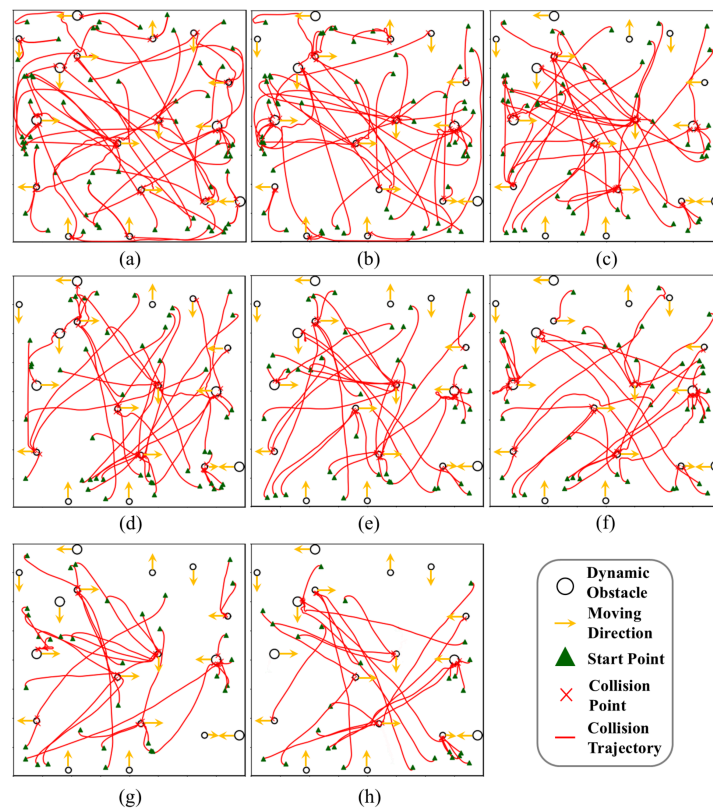


Figure 15. Failure trajectories in DyEnv2: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

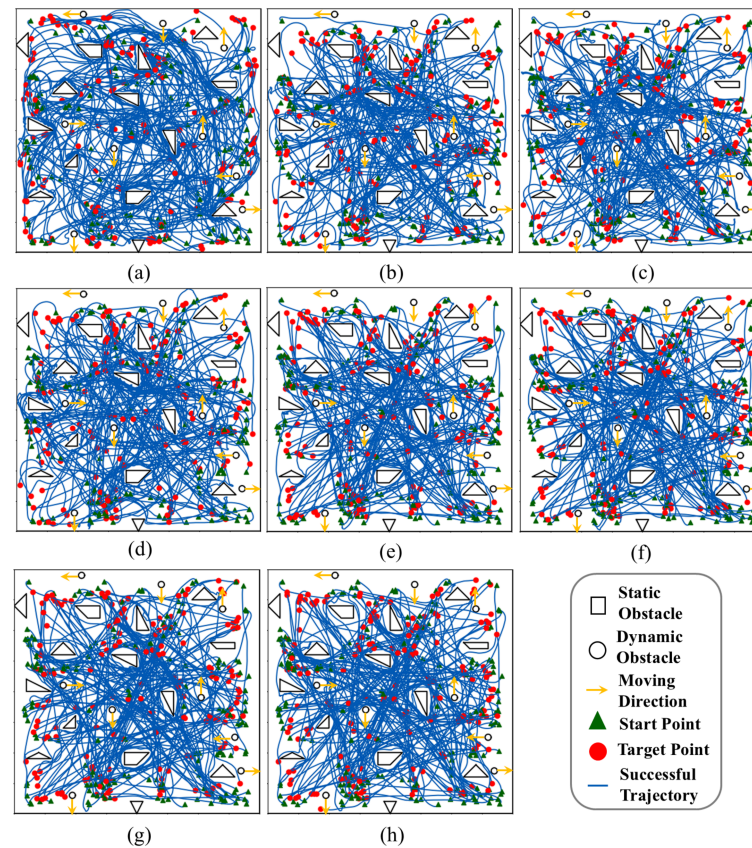


Figure 16. Success trajectories in Sta_DyEnv1: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

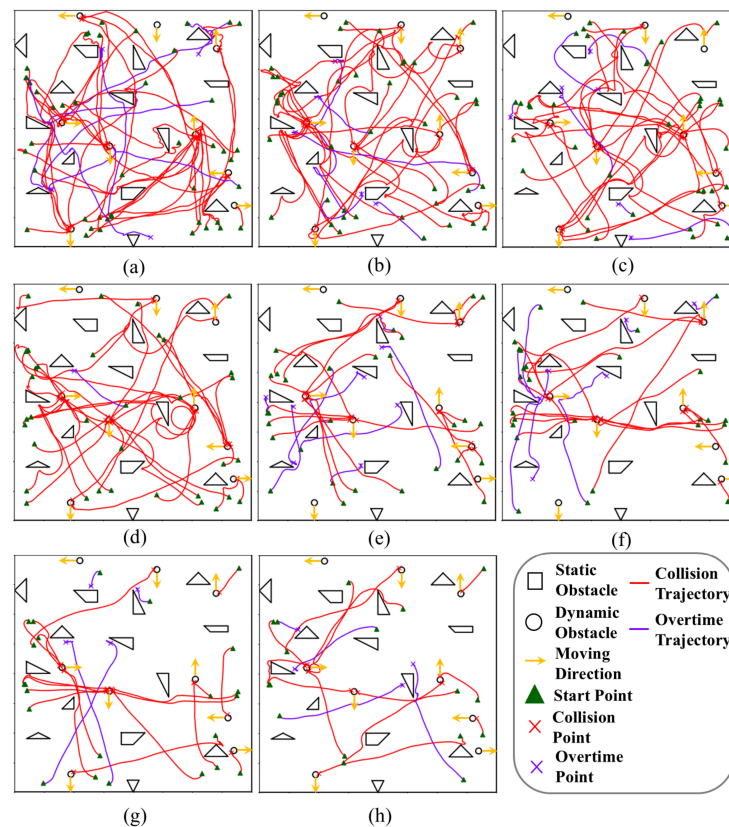


Figure 17. Failure trajectories in Sta_DyEnv1: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

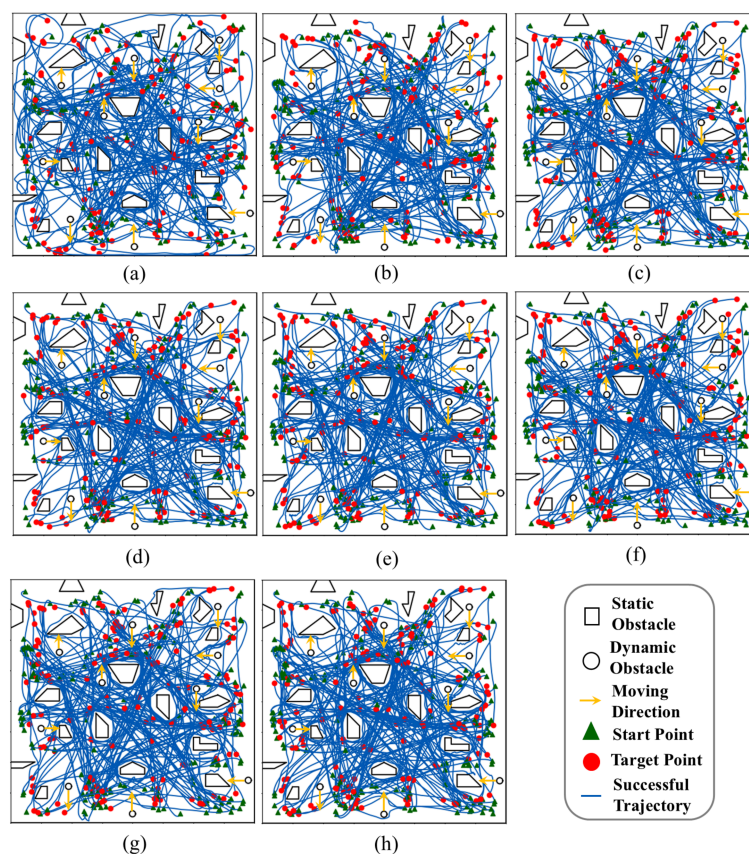


Figure 18. Success trajectories in Sta_DyEnv2. (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

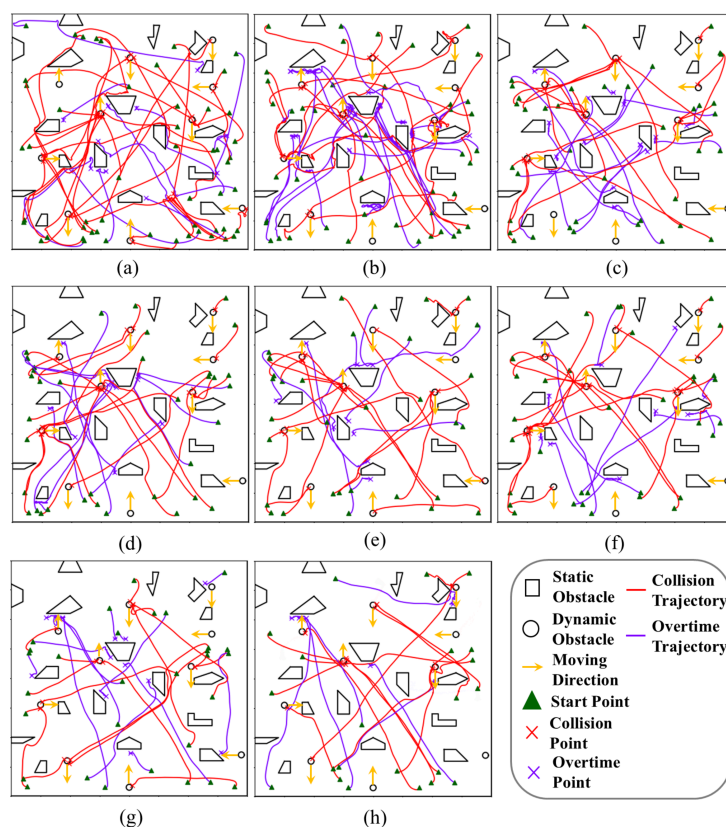


Figure 19. Failure trajectories in Sta_DyEnv2: (a) DWA; (b) DDPG; (c) SDDPG; (d) SPPO; (e) DT1; (f) DT2; (g) ABN; (h) the proposed BDAT-Planner.

The visualized trajectories have revealed that compared to the results observed in mixed static–dynamic obstacle environments, the failure paths of different methods in only dynamic obstacle environments have significantly increased. More specifically, regardless of the underwater evaluation environment, DWA and DDPG both exhibit the highest failure rates and show limited adaptability to dynamic environmental changes. In SDDPG and SPPO, the spiking neurons employ a fixed static threshold. Although the comparison method employs stronger modern actor–critic variants, it remains unable to effectively adjust itself based on changes in external dynamic interference, which results in suboptimal collision avoidance performance. Compared to the static threshold, the three heuristic dynamic thresholds (DT1, DT2, and ABN) exhibit slightly improved performance. Nevertheless, these methods rely on fixed hyperparameters during evaluation, which limits their ability to address challenging dynamic environments and makes it difficult to maintain model homeostasis. Comparatively, the BDAT-Planner enables the AUV to efficiently navigate away from both static and dynamic obstacles in different evaluation environments, achieving the fewest failed and timeout trajectories. The method facilitates dynamically and rationally adjusting the threshold of spiking neurons, which better captures dynamic changes in the underwater environment, robustly adapts to varying external conditions, maintains homeostasis, and offers strong generalization.

Additionally, Gaussian noise interference is added to the detection data acquired by the forward-looking sonar, which makes the simulation more closely resemble actual application scenarios, thereby demonstrating the robustness and adaptability of the proposed method to external conditions. As shown in Table 7, the experimental results of different methods under Gaussian noise interference in different evaluation environments are presented. The results indicate that the addition of Gaussian noise leads to a reduction in success rates for all comparison methods. Nevertheless, the BDAT-Planner proposed in this paper still achieves the highest success rate in different evaluation environments, with the least impact. It is reflected that even in the presence of dynamic obstacles and Gaussian noise interference, the proposed dynamic adaptive threshold method can still offer strong generalization and interference resistance abilities, and robustly adapt to the external environment while minimizing the impact of interference. It is also able to maintain model homeostasis while ensuring highly dynamic collision avoidance success rates.

Table 7. Performance of different methods under Gaussian noise in different environments.

External Interference	Methods	Success Rate			
		DyEnv1	DyEnv2	Sta_DyEnv1	Sta_DyEnv2
Gaussian Noise	DWA	65%	60.5%	68%	65.5%
	DDPG	67%	63.5%	72.5%	69.5%
	SDDPG	69.5%	67.5%	75%	73%
	SPPO	71.5%	69.5%	78%	75%
	DT1	74.5%	73.5%	82%	76.5%
	DT2	75%	72%	80.5%	77%
	ABN	80%	77%	84%	79%
	BDAT-Planner	84%	81.5%	88%	84.5%

Homeostasis represents the inherent characteristic of biological spiking neurons, which control the performance capabilities of the entire model through self-regulation. More specifically, when the AUV encounters dynamic obstacles, the membrane potential of the spiking neuron undergoes a rapid rise. Adaptive adjustment (dynamically decreasing the threshold) should be employed to increase the overall firing rate, encourage spike firing, prevent excessive suppression, and maintain certain levels of excitability. It enables it to respond to environmental changes and adapt to external dynamic disturbances, stably and efficiently avoiding dynamic obstacles and successfully reaching the target point. When underwater noise or other environmental disturbances cause slight fluctuations in the membrane potential of spiking neurons, the overall firing rate should be reduced through adaptive adjustment (dynamically raising the threshold) to suppress excessive excitation and maintain the overall homeostasis. In other words, if the threshold of spiking neurons lacks dynamic adaptive learning, the neurons become prone to overexcitation or excessive inhibition, leading to overall homeostasis imbalance and triggering unstable collision avoidance behavior. It

also leads to poor adaptability and robustness when confronted with various external disturbances. It makes it difficult to effectively respond to changes in dynamic obstacles and increases the risk of collisions. Consequently, maintaining homeostasis is crucial for safe and reliable collision avoidance in underwater navigation.

Based on the firing rate of spiking neurons, we employ the following statistical metrics, $R_s(n)$ and $R_{s,std}^{sta}(n)$, to evaluate the homeostasis performance of different spiking methods. These statistical metrics are motivated by existing work investigating homeostasis in biological neural networks [58,59], aiming to quantify the homeostasis performance of spiking models. $R_s(n)$ represents the mean of $R_{s,m}^p(n)$ during p trials, and $R_{s,std}^{sta}(n)$ refers to the standard deviation of $R_{s,m}^p(n)$ during p trials. Additionally, p denotes the number of collision avoidance trials for the AUV. During the evaluation, we randomly sample 200 starting points and target points for evaluation, consistent with the experimental evaluation setting, as shown in the following equations:

$$R_s(n) = \frac{1}{P} \sum_{p=1}^P R_{s,m}^p(n) \quad \text{for } p = 1, 2, \dots, P \quad (47)$$

$$R_{s,std}^{sta}(n) = \sigma(R_{s,m}^p(n)) \quad \text{for } p = 1, 2, \dots, P \quad (48)$$

where N denotes the number of network layers, and P denotes the number of trials. In this paper, it is set to 200. We define the spike firing rate of each layer $R_s^p(n) = r_n^s / N_n$, i.e., the number of spikes r_n^s sent by neurons in the n -th layer divided by the number of neurons N_n in the n -th layer. $R_{s,m}^p(n) = \frac{1}{N} \sum_{n=1}^N R_s^p(n)$ for $n = 1, 2, \dots, N$ represents the mean firing rate of all neurons during the p -th trial. $R_{s,std}^p(n) = \sigma(R_s^p(n))$ for $n = 1, 2, \dots, N$ denotes the standard deviation of the firing rate for all neurons during the p -th trial.

The fundamental objective of the BDAT-Planner in this paper for maintaining homeostasis is to enhance the performance of collision avoidance, which ensures it can safely and efficiently navigate around obstacles and successfully reach the target point. Consequently, spiking models with stronger homeostasis performance (smaller $\Delta R_s(n)$ and $\Delta R_{s,std}^{sta}(n)$ values) can achieve higher collision avoidance success rates in different evaluation environments compared to spiking models with weaker homeostasis. According to the results of the experimental success rates and homeostasis metrics, it can be observed that when transitioning from the baseline condition to environments with Gaussian noise interference, the BDAT-Planner proposed in this paper shows the smallest changes in homeostasis metrics, as shown in Table 8. For example, under Gaussian noise interference, in DyEnv2, the proposed method shows the smallest change in $\Delta R_s(n)$, only 0.008. Accordingly, the proposed BDAT-Planner has attained the highest success rate and optimal collision avoidance performance among the evaluation environments.

However, in static threshold methods (such as SDDPG and SPPO), the threshold of spiking neurons is fixed, which cannot dynamically adjust the threshold to regulate the firing rate. The homeostasis metrics of these two methods exhibit significant variation under different external disturbances, and the results indicate that these methods have a lower success rate and poor collision avoidance performance. These experimental results have demonstrated that the dynamic adaptive threshold method proposed in this paper achieves biologically plausible homeostasis behavior. Models with robust homeostasis performance can exhibit outstanding performance in complex environment collision avoidance tests, attaining higher collision avoidance success rates. Through dynamic adaptive learning of the threshold of spiking neurons, it avoids excessive excitation or inhibition of neurons, maintaining its own homeostasis. The BDAT-Planner proposed in this paper can robustly adapt to external disturbances and deliver strong generalization capabilities. It enables safe and efficient collision avoidance in underwater evaluation environments of various complexities, which significantly enhances the adaptability and stability of spiking models in dynamic underwater environments.

Table 8. Comparison of homeostasis performance with different methods.

Type	Method	Metrics			
		$R_s(n)$	$\Delta R_s(n)$	$R_{s,std}^{sta}(n)$	$\Delta R_{s,std}^{sta}(n)$
Baseline Condition (DyEnv2)	SDDPG	0.566		0.000963	
	SPPO	0.554		0.00114	
	DT1	0.516		0.00129	
	DT2	0.499		0.00182	
	ABN	0.502		0.00112	
	BDAT-Planner	0.485		0.000986	
Gaussian Noise (DyEnv2)	SDDPG	0.585	0.019	0.001846	0.000883
	SPPO	0.572	0.018	0.002036	0.000896
	DT1	0.531	0.015	0.002224	0.000934
	DT2	0.511	0.012	0.002632	0.000812
	ABN	0.513	0.011	0.001813	0.000693
	BDAT-Planner	0.493	0.008	0.001085	0.000099
Baseline Condition (Sta_DyEnv2)	SDDPG	0.553		0.000915	
	SPPO	0.542		0.00106	
	DT1	0.511		0.00121	
	DT2	0.489		0.00168	
	ABN	0.493		0.00107	
	BDAT-Planner	0.477		0.000942	
Gaussian Noise (Sta_DyEnv2)	SDDPG	0.574	0.021	0.001607	0.000692
	SPPO	0.562	0.020	0.001792	0.000732
	DT1	0.527	0.016	0.001966	0.000756
	DT2	0.501	0.012	0.002268	0.000588
	ABN	0.507	0.014	0.001583	0.000513
	BDAT-Planner	0.483	0.006	0.001023	0.000081

6. Conclusions

We introduce the dynamic adaptive threshold, BDAT, combining an SNN and DRL, to propose a novel biologically plausible BDAT-Planner framework for underwater dynamic collision avoidance of AUVs in complex, unknown environments. The BDAT-Planner consists of an SDAAN and DCNN. The proposed BDAT integrates an AET, which is positively correlated with membrane potential, a DDT, which is negatively correlated with the preceding rate of depolarization, and an RGT, which represents the rate of change in the historical threshold. These components interact and provide coordinated regulation. The mechanism is deployed in each spiking neuron of the SDAAN, where the spiking firing rate is adaptively adjusted through reasonable changes to the threshold to maintain homeostasis. We analyze training convergence across different methods using the reward curve during the training process. During the evaluation process, two dynamic obstacle environments (DyEnv1 and DyEnv2) and two mixed static–dynamic obstacle environments (Sta_DyEnv1 and Sta_DyEnv2) are constructed. The results from ablation studies and comparative experiments have demonstrated that, compared to static threshold methods and existing comparison methods, the proposed BDAT-Planner exhibits highly efficient dynamic collision avoidance abilities. The success rate is highest in the four evaluation environments, at 88%, 86.5%, 91.5%, and 89%, respectively. The homeostasis experimental results have demonstrated that, even under external environmental interference, the proposed BDAT-Planner exhibits minimal changes in homeostasis metrics, which indicates its strong adaptability and generalization. Furthermore, it can robustly adapt to external dynamic interference while maintaining homeostasis of the model.

Nevertheless, our method still has some unresolved issues. On the one hand, although the behaviors of certain biological neurons are effectively mimicked by the proposed model, it fails to fully capture all neuronal properties. In future work, advanced dynamic threshold models, incorporating elements such as dynamic refractory periods and network activity, will be explored for application in more complex spiking neurons. On the other hand, although the current research in this paper

demonstrates the effectiveness of the proposed BDAT-Planner, its deployment in real-world AUV scenarios may introduce some Sim2Real gaps. We have constructed multiple underwater evaluation environments of varying complexity and performed ablation studies, comparative experiments, and interference experiments to enhance the robustness of the method. The experimental results have also validated the efficiency and reliability of the proposed method. Nonetheless, there exists a certain disparity between simulation environments and real-world conditions. To obtain a well-trained collision avoidance model suitable for practical deployment, it is necessary to construct underwater environments that more closely approximate real-world conditions. In the future, it may be considered to incorporate elements such as real ocean topography data. Additionally, this paper focuses primarily on the innovative design and experimental validation of the dynamic collision avoidance method, with emphasis on the safety, effectiveness, and stability of the underwater dynamic collision avoidance method. In future research, we will further prove the assertions to enhance the completeness and rigor of our research.

Hardware in-the-loop experiments and real-world scenario experiments are crucial for further validating the practical application value of the proposed method. In the future, based on previous research and considering the mentioned situations, we will adopt the approach of simulation training and deployment fine-tuning to address practical deployment issues. We first intend to perform hardware-in-the-loop experiments and experimental pool validation to evaluate its safety performance and collision avoidance efficiency, and then conduct large-scale real-world ocean environment experiments. Simultaneously, to conduct practical engineering verification, it is also necessary to actively seek permissions from neuromorphic hardware companies and attempt to use their neuromorphic hardware chips for validation in real-world application scenarios. Once the hardware is available, we will consider potential constraints that may occur during practical application and make further improvements for deploying underwater robots in practical scenarios, thereby fully leveraging the advantages of the spiking model. Simultaneously, dynamic threshold models require neuromorphic chips with higher communication speeds. We are also actively exploring hardware chips that adapt to threshold changes and optimize the spiking homeostasis model to suit more spiking neurons. In future research, it is both necessary and feasible to deploy the proposed method onto the actual AUV, which will significantly advance the future development of neuromorphic approaches in the field of intelligent underwater unmanned systems.

Author Contributions: B.Z.: Writing—Original Draft, Writing—Review and Editing, Conceptualization, Methodology, Software, and Investigation. Z.Z.: Data Curation, Formal Analysis, and Validation. W.F.: Supervision and Funding Acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62572142, 62172121) and the Fundamental Research Funds for the Central Universities at Harbin Engineering University (3072025ZX2601).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Zhu, W.; Cui, W. From machinery to biology: A review on mapless autonomous underwater navigation. *J. Mar. Sci. Eng.* **2025**, *13*, 2202. [\[CrossRef\]](#)
2. Kot, R. Review of collision avoidance and path planning algorithms used in autonomous underwater vehicles. *Electronics* **2022**, *11*, 2301. [\[CrossRef\]](#)
3. Zhang, B.; Ji, D.; Liu, S.; Zhu, X.; Xu, W. Autonomous underwater vehicle navigation: A review. *Ocean Eng.* **2023**, *273*, 113861. [\[CrossRef\]](#)
4. Zhang, B.; Xing, H.; Zhang, Z.; Feng, W. Autonomous obstacle avoidance decision method for spherical underwater robot based on brain-inspired spiking neural network. *Expert. Syst. Appl.* **2025**, *274*, 127021. [\[CrossRef\]](#)
5. Zhao, J.; Liu, T.; Huang, J. Reinforcement learning-based model predictive path integral control for obstacle avoidance of autonomous underwater vehicles. *J. Field Robot.* **2025**, *42*, 4337–4352. [\[CrossRef\]](#)

6. Guo, X.; Zhao, D.; Fan, T.; Long, F.; Fang, C.; Long, Y. Autonomous underwater vehicle path planning based on improved salp swarm algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 1446. [\[CrossRef\]](#)
7. Li, Y.; Ma, Y.; Cao, J.; Yin, C.; Ma, X. An obstacle avoidance strategy for AUV based on state-tracking collision detection and improved artificial potential field. *J. Mar. Sci. Eng.* **2024**, *12*, 695. [\[CrossRef\]](#)
8. Liu, T.; Huang, J.; Zhao, J. Research on obstacle avoidance of underactuated autonomous underwater vehicle based on offline reinforcement learning. *Robotica* **2024**, *43*, 194–218. [\[CrossRef\]](#)
9. Huang, F.; Cui, Y.; Zhang, W.; Xu, J. Deep-reinforcement-learning-based reactive collision avoidance controller for an underactuated AUV using multibeam forward-looking sonar. *IEEE Internet Things* **2025**, *12*, 36604–36616. [\[CrossRef\]](#)
10. Yuan, J.; Han, M.; Wang, H.; Zhong, B.; Gao, W.; Yu, D. AUV collision avoidance planning method based on deep deterministic policy gradient. *J. Mar. Sci. Eng.* **2023**, *11*, 2258. [\[CrossRef\]](#)
11. Daoudal, G.; Debanne, D. Long-term plasticity of intrinsic excitability: Learning rules and mechanisms. *Learn Mem.* **2003**, *10*, 456–465. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Hashemi, P.; Nazarimehr, F.; Parastesh, F.; Towhidkhal, F.; Jafari, S. Inter-layer coupling effects on synchronization in multiplex neuronal networks with different topologies. *Chaos Solitons Fractals* **2025**, *201*, 117205. [\[CrossRef\]](#)
13. Shen, X.; Cui, X.; Catal, Y.; Li, J.; Northoff, G. Intrinsic neural timescales shape memory encoding and retrieval. *Neuroimage* **2026**, *325*, 121634. [\[CrossRef\]](#)
14. Roy, K.; Jaiswal, A.; Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature* **2019**, *575*, 607–617. [\[CrossRef\]](#)
15. Zhang, A.; Han, Y.; Niu, Y.; Gao, Y.; Chen, Z.; Zhao, K. Self-evolutionary neuron model for fast-response spiking neural networks. *IEEE Trans. Cogn. Dev. Syst.* **2022**, *14*, 1766–1777. [\[CrossRef\]](#)
16. Wang, X.; Lin, X.; Dang, X. Supervised learning in spiking neural networks: A review of algorithms and evaluations. *Neural Netw.* **2020**, *125*, 258–280. [\[CrossRef\]](#)
17. Zhang, W.; Linden, D.J. The other side of the engram: Experience driven changes in neuronal intrinsic excitability. *Nat. Rev. Neurosci.* **2023**, *4*, 885–900. [\[CrossRef\]](#)
18. Schrauwen, B.; Wardermann, M.; Verstraeten, D.; Steil, J.J.; Stroobandt, D. Improving reservoirs using intrinsic plasticity. *Neurocomputing* **2008**, *71*, 1159–1171. [\[CrossRef\]](#)
19. Cooper, L.N.; Bear, M.F. The BCM theory of synapse modification at 30: Interaction of theory with experiment. *Nat. Rev. Neurosci.* **2012**, *13*, 798–810. [\[CrossRef\]](#)
20. Ai, Q.; Yang, Y.; Cai, M.; Chen, K.; Liu, Q.; Ma, L. A cross-layer residual spiking neural network with adaptive threshold leaky integrate-and-fire neuron and learnable surrogate gradient. *Knowl.-Based Syst.* **2025**, *319*, 113575. [\[CrossRef\]](#)
21. Yuan, J.; Wang, H.; Zhang, H.; Lin, C.; Yu, D.; Li, C. AUV obstacle avoidance planning based on deep reinforcement learning. *J. Mar. Sci. Eng.* **2021**, *9*, 1166. [\[CrossRef\]](#)
22. Hadi, B.; Khosravi, A.; Sarhadi, P. Deep reinforcement learning for adaptive path planning and control of an autonomous underwater vehicle. *Appl. Ocean Res.* **2022**, *129*, 103326. [\[CrossRef\]](#)
23. Zhang, K.; Ye, Y.; Chen, K.; Li, Z.; Li, K. Enhanced AUV autonomy through fused energy-optimized path planning and deep reinforcement learning for integrated navigation and dynamic obstacle detection. *J. Mar. Sci. Eng.* **2025**, *13*, 1294. [\[CrossRef\]](#)
24. Xia, J.; Zhu, X.; Liu, Z.; Luo, Y.; Wu, Z.; Wu, Q. Research on collision avoidance algorithm of unmanned surface vehicle based on deep reinforcement learning. *IEEE Sens. J.* **2023**, *23*, 11262–11273. [\[CrossRef\]](#)
25. Wang, Z.; Zhang, X.; Li, W.; Feng, Z. Real-time UUV obstacle avoidance through flexible steering technology based on improved soft actor-critic framework. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 3003514. [\[CrossRef\]](#)
26. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw.* **1997**, *10*, 1659–1671. [\[CrossRef\]](#)
27. Taherkhani, A.; Belatreche, A.; Li, Y.; Cosma, G.; Maguire, L.P.; McGinnity, T.M. A review of learning in biologically plausible spiking neural networks. *Neural Netw.* **2020**, *122*, 253–272. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Tan, W.; Patel, D.; Kozma, R. Strategy and benchmark for converting deep q-networks to event-driven spiking neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Virtual, 2–9 February 2021; pp. 9816–9824. [\[CrossRef\]](#)
29. Patel, D.; Hazan, H.; Saunders, D.J.; Siegelmann, H.T.; Kozma, R. Improved robustness of reinforcement learning policies upon conversion to spiking neuronal network platforms applied to Atari Breakout game. *Neural Netw.* **2019**, *120*, 108–115. [\[CrossRef\]](#)
30. Zhang, B.; Zhang, Z.; Feng, W. NSM-planner: Neuromorphic planner with spiking memory for underwater autonomous obstacle avoidance decision of AUV. *Adv. Eng. Inform.* **2025**, *68*, 103686. [\[CrossRef\]](#)
31. Wang, X.; Hou, Z.; Lv, F.; Tan, M.; Wang, Y. Mobile robots' modular navigation controller using spiking neural networks. *Neurocomputing* **2014**, *134*, 230–238. [\[CrossRef\]](#)
32. Liu, X.; Guo, Y.; Chen, Y.; Zhou, J.; Zhang, Y.; Peng, W.; Huang, X.; Ma, Z. Enhancing navigation performance in unknown environments using spiking neural networks and reinforcement learning with asymptotic gradient method. *Complex. Intell. Syst.* **2025**, *11*, 156. [\[CrossRef\]](#)

33. Tang, G.; Kumar, N.; Michmizos, K.P. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 6090–6097. [\[CrossRef\]](#)
34. Zhao, F.; Zeng, Y.; Han, B.; Fang, H.; Zhao, Z. Nature-inspired self-organizing collision avoidance for drone swarm based on reward-modulated spiking neural network. *Patterns* **2022**, *3*, 100611. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Sanyal, S.; Manna, R.K.; Roy, K. EV-Planner: Energy-efficient robot navigation via event-based physics-guided neuromorphic planner. *IEEE Robot. Autom. Lett.* **2024**, *9*, 2080–2087. [\[CrossRef\]](#)
36. Yang, B.; Yuan, M.; Zhang, C.; Hong, C.; Pan, G.; Tang, H. Spiking reinforcement learning with memory ability for mapless navigation. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 6681–6688. [\[CrossRef\]](#)
37. Zhang, C.; Yip, K.W.; Yang, B.; Zhang, Z.; Yuan, M.; Yan, R.; Tang, H. CASRL: Collision avoidance with spiking reinforcement learning among dynamic, decision-making agents. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 8031–8038. [\[CrossRef\]](#)
38. Shim, H.G.; Lee, Y.S.; Kim, S.J. The emerging concept of intrinsic plasticity: Activity-dependent modulation of intrinsic excitability in cerebellar purkinje cells and motor learning. *Exp. Neurobiol.* **2018**, *27*, 139–154. [\[CrossRef\]](#)
39. Marder, E.; Goaillard, J.M. Variability, compensation and homeostasis in neuron and network function. *Nat. Rev. Neurosci.* **2006**, *7*, 563–574. [\[CrossRef\]](#)
40. Hansel, C.; Yuste, R. Neural ensembles: Role of intrinsic excitability and its plasticity. *Front. Cell Neurosci.* **2024**, *18*, 1440588. [\[CrossRef\]](#)
41. Zhang, A.; Gao, Y.; Niu, Y.; Li, X.; Chen, Q. Intrinsic plasticity for online unsupervised learning based on soft-reset spiking neuron model. *IEEE Trans. Cogn. Dev. Syst.* **2023**, *15*, 337–347. [\[CrossRef\]](#)
42. Fang, W.; Yu, Z.; Chen, Y.; Masquelier, T.; Huang, T.; Tian, Y. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 2661–2671. [\[CrossRef\]](#)
43. Sun, H.; Cai, W.; Yang, B.; Cui, Y.; Xia, Y.; Yao, D.; Guo, D. A synapse-threshold synergistic learning approach for spiking neural networks. *IEEE Trans. Cogn. Dev. Syst.* **2024**, *16*, 544–558. [\[CrossRef\]](#)
44. He, X.; Li, Y.; Zhao, D.; Kong, Q.; Zeng, Y. MSAT: Biologically inspired multistage adaptive threshold for conversion of spiking neural networks. *Neural Comput. Appl.* **2024**, *36*, 8531–8547. [\[CrossRef\]](#)
45. Ding, J.; Dong, B.; Heide, F.; Ding, Y.; Zhou, Y.; Yin, B.; Yang, X. Biologically inspired dynamic thresholds for spiking neural networks. In Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS), Virtual, 28 November–9 December 2022; pp. 6090–6103. [\[CrossRef\]](#)
46. Wester, J.C.; Contreras, D. Biophysical mechanism of spike threshold dependence on the rate of rise of the membrane potential by sodium channel inactivation or subthreshold axonal potassium current. *J. Comput. Neurosci.* **2013**, *35*, 1–17. [\[CrossRef\]](#)
47. Higgs, M.H.; Spain, W.J. Kv1 channels control spike threshold dynamics and spike timing in cortical pyramidal neurons. *J. Physiol.* **2011**, *589*, 5125–5142. [\[CrossRef\]](#)
48. Hao, Y.; Huang, X.; Dong, M.; Xu, B. A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule. *Neural Netw.* **2020**, *121*, 387–395. [\[CrossRef\]](#)
49. Kim, T.; Hu, S.; Kim, J.; Kwak, J.Y.; Park, J.; Lee, S.; Kim, I.; Park, J.K.; Jeong, Y. Spiking neural network (SNN) with memristor synapses having non-linear weight update. *Front. Comput. Neurosci.* **2021**, *15*, 646125. [\[CrossRef\]](#)
50. Hu, D.; Wu, Y.; Zhou, Y.; Zeng, Q. Actuator fault detection for automation underwater vehicle via extended state observer and adaptive threshold. In Proceedings of the 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 4106–4111. [\[CrossRef\]](#)
51. Fontaine, B.; Pena, J.L.; Brette, R. Spike-threshold adaptation predicted by membrane potential dynamics in vivo. *PLoS Comput. Biol.* **2014**, *10*, e1003560. [\[CrossRef\]](#) [\[PubMed\]](#)
52. Azouz, R.; Gray, C.M. Adaptive coincidence detection and dynamic gain control in visual cortical neurons in vivo. *Neuron* **2003**, *37*, 513–523. [\[CrossRef\]](#)
53. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Shi, L. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* **2018**, *12*, 331. [\[CrossRef\]](#)
54. Wang, Z.; Xuan, J.; Shi, T. Multi-label fault recognition framework using deep reinforcement learning and curriculum learning mechanism. *Adv. Eng. Inform.* **2022**, *54*, 101773. [\[CrossRef\]](#)
55. Li, J.; Lu, H.; Zhang, H.; Zhang, Z. Dynamic target hunting under autonomous underwater vehicle (AUV) motion planning based on improved dynamic window approach (DWA). *J. Mar. Sci. Eng.* **2025**, *13*, 221. [\[CrossRef\]](#)
56. Zanatta, L.; Barchi, F.; Manoni, S.; Tolu, S.; Bartolini, A.; Acquaviva, A. Exploring spiking neural networks for deep reinforcement learning in robotic tasks. *Sci. Rep.* **2024**, *14*, 30648. [\[CrossRef\]](#)

57. Kachole, S.; Sajwani, H.; Naeini, F.B.; Makris, D.; Zweiri, Y. Asynchronous bioplausible neuron for spiking neural networks for event-based vision. In Proceedings of the 2024 European Conference on Computer Vision (ECCV), Milan, Italy, 29 September–4 October 2024; pp. 399–415. [\[CrossRef\]](#)
58. Turrigiano, G.G.; Nelson, S.B. Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.* **2004**, *5*, 97–107. [\[CrossRef\]](#) [\[PubMed\]](#)
59. Zenke, F.; Hennequin, G.; Gerstner, W. Synaptic plasticity in neural networks needs homeostasis with a fast rate detector. *PLoS Comput. Biol.* **2013**, *9*, e1003330. [\[CrossRef\]](#) [\[PubMed\]](#)

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.