

Dynamic Weight Adaptation in Spiking Neural Networks Inspired by Biological Homeostasis*

Yunduo Zhou¹, Bo Dong², Chang Li¹, Yuanchen Wang¹, Xuefeng Yin¹, Yang Wang¹, Xin Yang^{1†}

¹Key Laboratory of Social Computing and Cognitive Intelligence, Dalian University of Technology

²Cephia AI

zyd_dlut@163.com, bo.dong@cephia.ai, lchang@mail.dlut.edu.cn, ,
xfyin@mail.dlut.edu.cn, yangwang06@mail.dlut.edu.cn, xinyang@dlut.edu.cn

Abstract

Homeostatic mechanisms play a crucial role in maintaining optimal functionality within the neural circuits of the brain. By regulating physiological and biochemical processes, these mechanisms ensure the stability of an organism's internal environment, enabling it to better adapt to external changes. Among these mechanisms, the Bienenstock, Cooper, and Munro (BCM) theory has been extensively studied as a key principle for maintaining the balance of synaptic strengths in biological systems. Despite the extensive development of spiking neural networks (SNNs) as a model for bionic neural networks, no prior work in the machine learning community has integrated biologically plausible BCM formulations into SNNs to provide homeostasis. In this study, we propose a Dynamic Weight Adaptation Mechanism (DWAM) for SNNs, inspired by the BCM theory. DWAM can be integrated into the host SNN, dynamically adjusting network weights in real time to regulate neuronal activity, providing homeostasis to the host SNN without any fine-tuning. We validated our method through dynamic obstacle avoidance and continuous control tasks under both normal and specifically designed degraded conditions. Experimental results demonstrate that DWAM not only enhances the performance of SNNs without existing homeostatic mechanisms under various degraded conditions but also further improves the performance of SNNs that already incorporate homeostatic mechanisms.

1 Introduction

Homeostasis in biological organisms is a critical mechanism for maintaining survival and functionality by regulating physiological and biochemical processes to ensure internal environmental stability (Marder and Goaillard 2006; Desai 2003; Turrigiano and Nelson 2004; Marder and Prinz 2002; Mease et al. 2013). In the nervous system, homeostasis can be achieved through various mechanisms such as synaptic scaling, excitatory-inhibitory balance, Bienenstock, Cooper, and Munro (BCM) theory, and dynamic firing thresholds (Desai 2003; Keck, Hübener, and Bonhoeffer 2017; Turrigiano and Nelson 2004). These mechanisms

finely tune and balance neuronal activity, ensuring that the nervous system maintains a relatively stable activity level in response to external stimuli or internal disturbances, thereby preventing excessive excitation or inhibition of neurons. Several studies have shown that the absence or dysregulation of neuronal homeostasis can cause neuronal activity levels to escalate rapidly, resulting in network dysfunction (O'Leary et al. 2014; Marder and Prinz 2002).

Given the critical role of homeostasis in biological neural networks, many researchers have sought to incorporate this mechanism into spiking neural networks (SNNs) to enhance networks' performance (Meng, Jin, and Yin 2011; Hao et al. 2020; Kim et al. 2021; Xu, Liu, and Pei 2022; Sun and Du 2021). Due to the bio-inspired nature of SNNs, incorporating biological mechanisms into them is relatively straightforward (Roy, Jaiswal, and Panda 2019; Maass 1997), making SNNs an ideal platform for such endeavors. However, while previous attempts have introduced homeostatic mechanisms into SNNs, most of these approaches have been heuristic or empirical, lacking biological interpretability and failing to validate the effectiveness of neural homeostasis in artificial systems. Ding et al. (Ding et al. 2022) addressed this gap by proposing a biologically plausible method to implement homeostasis in SNNs through dynamic firing thresholds. Inspired by the adaptive thresholding observed in barn owl cochlear neurons, their method calculates the firing threshold of each neuron in real time based on membrane potential, dynamically modulating neuronal activity levels. This approach successfully introduces homeostasis into SNNs by regulating intrinsic neuronal properties.

Nonetheless, achieving homeostasis solely by adjusting intrinsic neuronal properties is incomplete. In biological systems, homeostasis depends not only on the regulation of individual neurons but also on the dynamic balance of interneuronal relationships (Desai 2003; Keck, Hübener, and Bonhoeffer 2017). The BCM theory (Bienenstock, Cooper, and Munro 1982), a classical model of synaptic plasticity, is a key mechanism for maintaining homeostasis by adapting synaptic weights based on neurons' historical activity levels to stabilize subsequent neuronal responses. Based on this theory, we developed the Dynamic Weight Adaptation Mechanism (DWAM), a mechanism that dynamically ad-

*Accepted at AAAI 2026. This is the extended version including technical appendices.

†Corresponding author.

justs network weights in real-time according to the firing relationships and activity levels of neurons. Specifically, DWAM provides two key contributions: 1) It assesses firing relationships between neurons to evaluate their correlation, enhancing weights between highly correlated neurons while weakening those between weakly correlated ones, thereby maintaining accurate inter-neuronal associations. 2) It dynamically determines the direction of weight modifications based on neuronal activity levels, preventing sustained enhancement of highly correlated neurons or persistent weakening of weakly correlated neurons, ensuring long-term balance and network stability.

While DWAM is inspired by the BCM theory, maintaining consistency with its principles and formulaic structure, the inherent differences between SNNs and biological neural systems make the direct application of the original biological BCM model ineffective for introducing homeostasis into SNNs. Therefore, we analyze the original BCM model and leverage the statistical firing patterns of SNN neurons to refine the strength of weight adjustments, tailoring it for application in SNNs.

We validated our method through dynamic obstacle avoidance and continuous control tasks under both normal and specifically designed degraded conditions. Across all scenarios, the developed BCM mechanism consistently delivered performance improvements. Notably, these gains were achieved by directly integrating DWAM into the host SNNs, without any fine-tuning.

In particular, we make the following contributions in this work:

- We propose a dynamic weight adaptation mechanism for SNNs, which represents the first implementation of biologically inspired homeostasis from the perspective of inter-neuronal relationships, based on the BCM theory.
- We design an approach that optimizes the original biological model’s weight adjustment strength using statistical firing cues from neurons, ensuring higher stability and feasibility in SNNs.
- We validate the effectiveness of the proposed method across multiple tasks and degraded conditions, demonstrating that DWAM achieves biologically inspired homeostasis and significantly enhances SNN generalization across diverse tasks and conditions. Notably, all results are obtained by directly integrating DWAM into host SNNs without any fine-tuning, highlighting its plug-and-play nature.

2 Background and Related Work

2.1 Homeostasis and BCM Theory

The BCM theory provides a framework for understanding synaptic plasticity by describing how synaptic weights change in response to neuronal activity. According to BCM theory, the synaptic modification is governed by the rule (Bienenstock, Cooper, and Munro 1982):

$$\frac{dw_{ij}}{dt} = \phi(x_i)x_j \quad (1)$$

$$\phi(x_i) = x_i(x_i - \theta_M) \quad (2)$$

where w_{ij} is the synaptic weight of the connection between neuron i and neuron j , x_j is the presynaptic activity, x_i is the postsynaptic activity, and $\phi(x_i)$ is a function of the postsynaptic activity. Here, θ_M is a sliding modification threshold that adapts based on the history of postsynaptic activity and ensures a balance between synaptic potentiation (when $x_i > \theta_M$) and depression (when $x_i < \theta_M$). The sliding modification threshold θ_M is typically given by:

$$\theta_M = \langle x_i^2 \rangle \quad (3)$$

where $\langle x_i^2 \rangle$ represents the long-term average of the squared postsynaptic activity.

2.2 Spiking Neural Networks (SNNs)

SNNs are a class of biologically inspired neural networks that aim to replicate the information processing mechanisms of biological neurons. Unlike traditional networks that process continuous signals, SNNs transmit information through discrete spikes, which occur when a neuron’s membrane potential exceeds a certain threshold. These spikes are generated by the integration of incoming signals over time. When the potential reaches the threshold, the neuron “fires”, sending a spike to neighboring neurons, and the potential resets.

Over the years, various neuron models have been developed to simulate the dynamics of spiking neurons, with two commonly used ones being the Spike Response Model (SRM) (Gerstner 1995) and the Leaky Integrate-and-Fire (LIF) model (Gerstner and Kistler 2002). More details about these two models are provided in Supplementary Note 2.

2.3 SNN with Homeostasis

Although homeostatic mechanisms are well studied in biology, their integration in machine learning is limited. Some works adopt BCM or STDP to guide SNN learning (Meng, Jin, and Yin 2011; Pan et al. 2023; Kheradpisheh et al. 2018; Gupta and Saurabh 2024; Wang et al. 2024), but they are task-specific or applied only during training, offering no functional homeostasis during inference and thus not directly comparable to our approach.

Other works have attempted to introduce homeostatic effects during the testing phase by adapting neuronal parameters such as thresholds or synaptic strengths (Hao et al. 2020; Kim et al. 2021; Xu, Liu, and Pei 2022; Zhang et al. 2025, 2022; Wang et al. 2023). However, these approaches are largely empirical or heuristic in nature, lacking biological interpretability and principled justification.

In contrast, Ding et al. (Ding et al. 2022) proposed a biologically inspired method for calculating the dynamic membrane potential threshold in SNNs. This threshold is related to the neuron’s membrane potential and the previous depolarization rate. They also demonstrated the advantages of their method compared to the aforementioned approaches that lack biological grounding. Therefore, the primary comparison method in this study is that of Ding et al.

3 Method

The core concept of the biological BCM theory can be divided into two main components, which describe the mech-

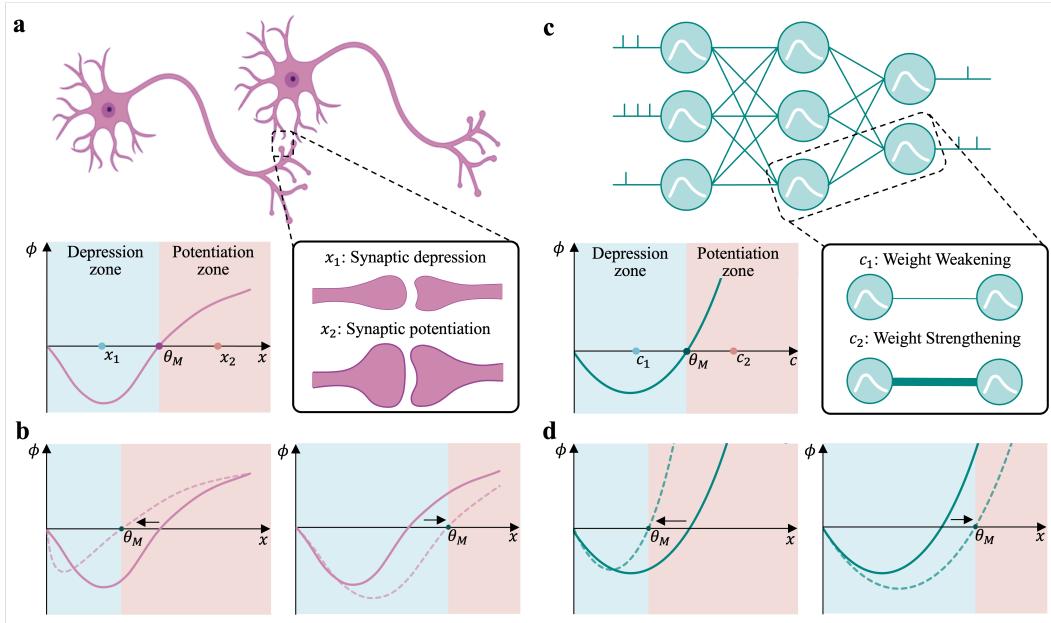


Figure 1: Comparison between biological BCM theory and the proposed DWAM in SNNs. a. Illustration of the BCM theory in biological neural systems. The synaptic modification function (ϕ) is governed by the postsynaptic neuron’s activity (x) and a modification threshold (θ_M). The depression zone ($x < \theta_M$) leads to synaptic weakening (e.g., x_1), while the potentiation zone ($x > \theta_M$) results in synaptic strengthening (e.g., x_2). b. The dynamic adjustment of the modification threshold (θ_M) based on the historical activity level of the postsynaptic neuron. A higher historical activity level shifts θ_M to the right, reducing potentiation likelihood, whereas lower historical activity shifts θ_M to the left, increasing potentiation likelihood. (illustrated as a shift from the solid line to the dashed line in the figure) c. Proposed DWAM for SNNs, inspired by the BCM theory. The synaptic modification function (ϕ) in DWAM operates on a similar principle, where the postsynaptic neuron’s firing rate (c) interacts with a modification threshold (θ_M) to determine weight weakening (c_1) or strengthening (c_2). d. As with BCM, a higher historical firing rate shifts θ_M to the right, whereas a lower firing rate shifts it to the left, enabling balanced and stable weight updates in the SNNs.

anism of synaptic strength adjustment and the mechanism for maintaining stability.

The first component describes how synaptic strength is adjusted based on the postsynaptic neuron’s activity relative to a modification threshold. As shown in Figure 1a, when the presynaptic neuron fires and the postsynaptic activity (e.g., x_1) is below the threshold (θ_M), the synapse is considered weakly correlated and is suppressed. Conversely, when the activity (e.g., x_2) exceeds θ_M , the synapse is strengthened. This mechanism dynamically modulates synaptic strength according to neuronal correlation, promoting balanced synaptic regulation.

The second component governs the dynamic adaptation of the modification threshold itself based on the postsynaptic neuron’s activity history (Figure 1b). Low historical activity lowers θ_M , increasing the chance of synaptic strengthening, while high activity raises θ_M , promoting synaptic weakening. This prevents unbounded potentiation or depression, maintaining synaptic stability over time.

In this study, we incorporate the BCM theory into SNNs to achieve a biologically inspired homeostatic mechanism. While prior work such as BDETT (Ding et al. 2022) intro-

duces homeostasis by regulating intrinsic neuronal properties during training, our approach focuses on inter-neuronal relationships, providing a complementary and orthogonal mechanism that can coexist with intrinsic regulation. Importantly, unlike BDETT which requires retraining to exploit noise resilience, our method is plug-and-play and can be applied directly to pre-trained SNNs during inference, enabling homeostasis without extra computational cost. This combination of biological plausibility, orthogonal homeostatic strategy, and deployment efficiency makes our approach both practically and conceptually significant.

Section 3.1 elaborates on how to implement the synaptic strength adjustment mechanism in SNNs, corresponding to the first component of the BCM theory. Section 3.2 introduces the realization of the stability maintenance mechanism in SNNs, addressing related challenges and presenting our solutions. Together, these two components form the proposed DWAM.

3.1 The Synaptic Adjustment Mechanism

The synaptic strength adjustment mechanism in SNNs can be implemented by substituting the firing rate c of spik-

ing neurons for the activity level x of biological neurons in Eqs. 1 and 2. At a given timestamp t , the modification of the weight from the j -th neuron in the $l-1$ -th layer to the i -th neuron in the l -th layer is defined as follows:

$$w_{ij}(t) = w_{ij}(t-1) + \phi_{ij}(t)c_j^{l-1}(t)|w_{ij}(t-1)|\psi, \quad (4)$$

$$\phi_{ij}(t) = c_i^l(t)(c_i^l(t) - \theta_i^l(t)) \quad (5)$$

where ϕ is the modification function for the weights; c is the firing rate of the neuron; θ is the sliding modification threshold; w_{ij} is the weight from the j -th neuron in the $l-1$ -th layer to the i -th neuron in the l -th layer, and ψ is a constant. In this implementation, we employ $|w|\psi$ as a scaling factor to dynamically adjust the magnitude of weight changes based on the current weight value. This design ensures that the adjustment for smaller weights does not become excessively large, preventing undue disruption to their stability. Similarly, it allows larger weights to receive sufficient adjustments, avoiding under-scaling that could reduce the effectiveness of weight modulation.

According to these formulas, the weight change is proportional to the product of the modification function and the presynaptic neuron firing rate. Additionally, the postsynaptic neuron firing rate is governed by a dynamic threshold level (θ). When the current firing rate exceeds this threshold, the weight is strengthened; conversely, when the firing rate falls below the threshold, the weight is weakened. As shown in Figure 1c, this process aligns with the first component of the biological BCM theory.

3.2 The Stability Maintenance Mechanism

According to the BCM theory, the sliding modification threshold θ_M in Eq.3 is a super-linear function of the history of postsynaptic activity. Benuskova et al. (Benuskova, Diamond, and Ebner 1994) developed a computational model of a single representative barrel cell based on the BCM theory, where θ_M is defined as:

$$\theta_M(t) = \left[\frac{\langle c^2(t) \rangle_\tau}{\eta} \right], \quad (6)$$

$$\langle c^2(t) \rangle_\tau = \frac{1}{\tau} \int_{-\infty}^t c^2(t') e^{-\frac{(t-t')}{\tau}} dt' \quad (7)$$

where η and τ are positive scaling constants. From these relations, it follows that $\theta_M(t)$ is positively correlated with the past firing rate, and thus it can be used to measure the averaged neuron activity in the recent past. To make it applicable to SNNs, we discretize Eqs. 6 and 7 as:

$$\theta_{M,i}^l(t) = (1 - \alpha)\theta_{M,i}^l(t-1) + \alpha(c_i^l(t))^2 \quad (8)$$

This is a commonly used recursive form of exponential moving average, where α controls the update rate and corresponds to the decay constant of the sliding modification threshold.

Although Eqs. 4, 5, and 8 successfully reproduce the BCM theory in biological neural systems, when θ_M defined in Eq. 8 is used as $\theta_i^l(t)$ in Eq. 5, the resulting SNN applying Eqs. 4, 5, and 8 fails to achieve the expected performance. To distinguish this approach from the final method,

we refer to the method based on these three equations as Biological-Originated DWAM (BioDWAM). This discrepancy arises primarily due to differences in neuronal firing rates. In biological systems, neurons typically maintain relatively low firing rates to ensure sparse coding and energy efficiency (Olshausen and Field 1996; Stringer et al. 2019). However, in SNNs, some neurons may need to sustain high firing rates for extended periods to meet task requirements and maintain network performance. When a neuron exhibits prolonged high firing rates, the sliding modification threshold gradually increases and stabilizes at a high level. According to Eqs. 4 and 5, this leads to continuous weakening of the weights connected to the neuron, preventing the network from reaching a stable state.

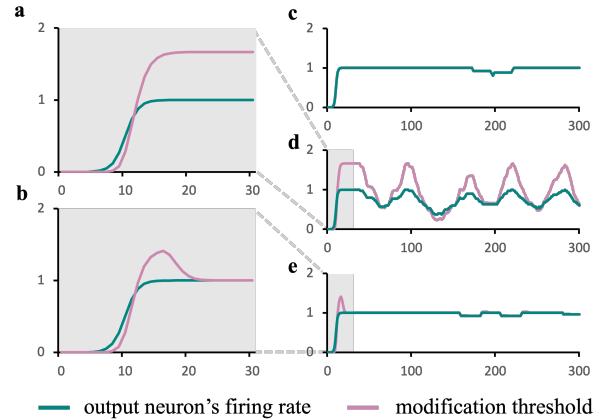


Figure 2: Behavioral comparison between BioDWAM and DWAM in the toy example. All subplots show simulation steps (x-axis) versus value (y-axis), with green and pink curves indicating firing rate and modification threshold, respectively. a & b. Zoomed-in views of the initial phase in d and e, showing early threshold adaptation. c. Firing rate stabilizes at 1 without synaptic adjustment. d. BioDWAM induces oscillatory instability at high firing rates. e. DWAM maintains stability by modulating threshold according to firing variability.

To illustrate this phenomenon, we designed a toy example. In this setup, the output layer of the network contains only a single neuron, trained to maintain a high firing rate, ideally close to 1. We applied BioDWAM directly to the output layer of this network and recorded the changes in the output neuron's firing rate and modification threshold over time. See Supplementary Note 3 for further training details.

Figure 2c shows the firing rate of the output neuron without the application of BioDWAM. The firing rate gradually increases and stabilizes at 1, representing the ideal behavior of the network. In contrast, Figure 2d illustrates the behavior when BioDWAM is applied. Initially, when the output neuron's firing rate is low, the sliding modification threshold remains within a reasonable range. However, as the firing rate increases to a high level, the threshold rises significantly. According to Eqs. 4 and 5, this results in a weakening of the

weights connected to the neuron. As a consequence, the firing rate begins to decrease. Once the firing rate falls below the sliding modification threshold, the weights are strengthened again, causing the firing rate to rise once more. This cycle continues, leading to an unstable oscillatory output characterized by wave-like patterns.

This observation highlights that directly applying the original biological synaptic adjustment model to SNNs can introduce stability issues, particularly when dealing with neurons with high firing rates.

Our Solution To retain the original stabilizing effect of the BCM theory in neural networks and to avoid the instability caused by high firing-rate neurons, we introduce a balancing term in Eq. 8:

$$\theta_i^l(t) = \theta_{M,i}^l(t) \zeta \frac{\sigma(c_i^l, n)}{\mu(c_i^l, n)} + c_i^l(t) \left(1 - \zeta \frac{\sigma(c_i^l, n)}{\mu(c_i^l, n)}\right) \quad (9)$$

Here, ζ is a positive scaling constant, and σ and μ denote the standard deviation and mean of the postsynaptic neuron's firing rates over the past n time steps. Their ratio, $\frac{\sigma}{\mu}$, is the coefficient of variation (CV), which reflects the stability of neuronal activity.

As shown in Eq. 9, the CV modulates the influence of the biologically originated sliding threshold $\theta_{M,i}^l(t)$ on $\theta_i^l(t)$. A high CV (i.e., unstable firing) increases the influence of $\theta_{M,i}^l(t)$, pulling $\theta_i^l(t)$ toward it. Conversely, a low CV (i.e., stable firing) causes $\theta_i^l(t)$ to align more closely with the current firing rate. As indicated in Eqs. 4 and 5, when $\theta_i^l(t)$ matches the firing rate, weight updates diminish, preventing persistent high-frequency firing from continuously weakening synaptic weights.

We applied DWAM to the same toy example as before. As shown in Figure 2e, DWAM effectively mitigates the instability observed in BioDWAM at high firing rates. To better illustrate the behavioral differences, the early phases of Figure 2d and e are enlarged in Figure 2a and b. Initially, DWAM behaves similarly to BioDWAM, but once the neuron stabilizes at a high firing rate, DWAM's correction threshold gradually converges to that rate. This allows DWAM to retain the biologically consistent behavior of the BCM theory at low firing rates while avoiding the instability seen when directly applying the biological model to SNNs.

By integrating Eqs. 4, 5, and 9, we successfully incorporate the BCM theory into SNNs, providing biologically-plausible homeostasis for the network.

4 Experiments and Results

We validated the effectiveness of DWAM in two types of tasks: a robot obstacle avoidance task and continuous control tasks. To further evaluate its generalization across different neuron models, we conducted experiments using both LIF and SRM neurons.

Robotic Obstacle Avoidance Task: In this task, the robot navigates from a random start to a target point in an obstacle-filled environment. A trial is successful if the robot reaches the target without collisions. Each round includes 200 trials, with the success rate (SR) defined as: $SR = \frac{N_{\text{success}}}{N_{\text{total}}}$, where N_{success} is the number of successful trials and $N_{\text{total}} = 200$.

The task is repeated over five rounds, and the final evaluation metric is the average success rate across all rounds.

Continuous Control Tasks: We used two standard OpenAI Gym environments (Brockman 2016): HalfCheetah-v3 and Ant-v3. The goal is to control a half-mechanical cheetah or a quadruped robot to move forward over 1000 steps while maximizing forward velocity. Each round consists of 10 trials, and the highest cumulative reward is recorded per round. The task is repeated for 10 rounds, and the average score across rounds is used for evaluation.

In addition to performance metrics (SR and reward), we evaluated the homeostasis of host SNNs using two firing-rate-based measures: HM_m and HM_{std} . These measures quantify the mean and variance of absolute changes in neuron firing rates across P degraded-condition trials, relative to the standard condition, reflecting the stability of network activity. Details on these measures can be found in Supplementary Note 4.

Experimental Setup **Robotic Obstacle Avoidance Task:** We selected two types of host SNNs: one without a homeostasis mechanism, the Spiking Action Network (SAN) (Tang, Kumar, and Michmizos 2020), and another with the best current homeostasis mechanism, the Bioinspired Dynamic Energy-Temporal Threshold (BDETT) (Ding et al. 2022). SAN and BDETT were first trained using the spiking deep deterministic policy gradient (SDDPG) framework (Tang, Kumar, and Michmizos 2020). After training, we directly integrated DWAM into the trained models without any further retraining, resulting in SAN-DWAM and BDETT-DWAM. We then conducted comparisons within each model group (i.e., SAN vs. SAN-DWAM, and BDETT vs. BDETT-DWAM) to evaluate the effectiveness of DWAM. See Supplementary Note 5 for further training details.

Continuous Control Task: we selected two types of host SNNs: one without a homeostasis mechanism, the population-coded SAN (PopSAN) (Tang et al. 2020), and the other with the best current homeostasis mechanism, BDETT (Ding et al. 2022). PopSAN is an improved version of SAN, featuring specifically designed encoders and decoders to handle high-dimensional control tasks. We construct PopSAN-DWAM and BDETT-DWAM using the same method as the obstacle avoidance task. Both PopSAN and BDETT were trained using the twin-delayed deep deterministic policy gradient (TD3) algorithm (Fujimoto, Hoof, and Meger 2018). See Supplementary Note 6 for further training details.

To verify the necessity of our modifications to the original biological formulation, we further conducted controlled comparisons by integrating BioDWAM into SAN, PopSAN, and BDETT, resulting in SAN-BioDWAM, PopSAN-BioDWAM, and BDETT-BioDWAM. These variants were included in the experiments as supplementary baselines. Their results and implementation details are provided in Supplementary Notes 5 and 6.

4.1 Robotic Obstacle Avoidance with DWAM

In this task, a mobile robot navigates from a random start to a random target within a given scene, using a Robot Peak light

Table 1: Quantitative performance of the robotic obstacle avoidance task under degraded conditions, where higher SR indicates better performance, and lower HM_m and HM_{std} values reflect improved homeostasis.

Name	0.2				6.0				GN			
	LIF		SRM		LIF		SRM		LIF		SRM	
	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}
SAN	88.2%	0.109/0.115	86.0%	0.130/0.117	71.7%	0.018/0.022	75.6%	0.027/0.028	87.6%	0.033/0.031	81.6%	0.039/ 0.034
SAN-DWAM	90.6%	0.091/0.113	87.3%	0.121/0.112	84.3%	0.018/0.022	76.1%	0.026/0.027	88.4%	0.031/0.030	83.3%	0.037/0.034
BDETT	70.7%	0.104/0.109	64.0%	0.073/0.097	79.6%	0.016/0.022	71.6%	0.021/ 0.022	87.5%	0.025/ 0.026	73.4%	0.019/0.023
BDETT-DWAM	82.6%	0.103/0.103	75.0%	0.070/0.096	82.8%	0.015/0.021	80.0%	0.017/0.022	89.3%	0.024/0.026	76.6%	0.019/0.022
8-bit Loihi weight												
Name	GN weight				30% Zero weight							
	LIF		SRM		LIF		SRM		LIF		SRM	
	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}	SR	HM_m/HM_{std}
SAN	85.4%	0.005/0.005	85.5%	0.004/0.003	55.2%	0.144/0.168	0.0%	0.166/0.158	66.3%	0.102/0.104	5.7%	0.108/0.105
SAN-DWAM	90.1%	0.004/0.003	84.7%	0.004/0.004	70.4%	0.106/0.127	0.5%	0.162/0.148	77.4%	0.093/0.102	14.0%	0.105/0.100
BDETT	91.1%	0.003/0.003	83.3%	0.003/0.002	79.6%	0.062/0.081	79.0%	0.043/0.051	56.7%	0.057/0.070	56.0%	0.056/0.066
BDETT-DWAM	93.8%	0.003/0.003	86.9%	0.002/0.002	89.8%	0.052/0.073	84.4%	0.035/0.042	78.3%	0.046/0.057	81.6%	0.054/0.059

detection and ranging (RPLIDAR) sensor for obstacle detection. To evaluate DWAM’s homeostatic capability, we tested it under several degraded conditions based on a static environment: (1) dynamic obstacles, by adding moving objects; (2) degraded input, by disturbing LIDAR data using “0.2”, “6.0” and “GN” methods; and (3) weight uncertainty, by perturbing trained weights via “8-bit Loihi”, “GN weight” and “30% zero weight.” Detailed explanations of these disturbance methods can be found in Supplementary Note 5.

Success Rate Table 1 presents the success rate comparison between the host SNN and the host SNN integrated with DWAM under degraded conditions, evaluated on both LIF and SRM neuron models. The integration of DWAM consistently improves the success rates of the host SNN to varying degrees, except for the SAN model using SRM neurons under the 8-bit Loihi weight condition. Notably, SAN-DWAM demonstrates clear performance gains over SAN, particularly under severe input perturbations such as GN noise with LIF neurons. Similarly, BDETT-DWAM shows a consistent advantage over BDETT, especially under conditions involving significant weight degradation. These results show that DWAM effectively introduces a stability mechanism into SNNs lacking intrinsic homeostasis, enabling them to overcome challenging conditions. The observed improvements also suggest that multiple stability mechanisms can coexist within SNNs, supporting the viability of our approach.

Homeostasis Metrics In biological systems, each neuron has a unique firing rate set point. When a network is in homeostasis, the firing rates of individual neurons under different conditions should remain close to their respective set points. In other words, as the network transitions from standard conditions to degraded conditions, an SNN with stronger homeostasis is expected to exhibit smaller changes in firing rates, reflected by lower values of the HM_m and HM_{std} metrics.

Table 1 compares firing rate changes in the host SNN with and without DWAM across various degraded conditions. DWAM consistently reduces both metrics, except in

the SAN-SRM model under 8-bit Loihi weights. These results support DWAM’s effectiveness in embedding biologically inspired homeostatic regulation into SNNs.

4.2 Continuous Control with DWAM

The continuous control tasks include HalfCheetah-v3 and Ant-v3, which vary in complexity: HalfCheetah has a 17D observation and 6D action space, while Ant uses 110D and 8D, respectively. This diversity enables a comprehensive evaluation of DWAM in continuous control. In both tasks, the objective is to control a six-joint half-mechanical cheetah (HalfCheetah) or a quadruped robot (Ant) to move forward for 1000 steps while maximizing velocity. As in the obstacle avoidance task, we tested under degraded conditions based on the baseline environment, including degraded input (disturbing observations via “MIN”, “MAX”, and “GN”) and weight uncertainty (same as in obstacle avoidance). Detailed explanations of these disturbance methods can be found in Supplementary Note 6.

Reward The reward comparison between the host SNN and the host SNN integrated with DWAM under degraded conditions for the HalfCheetah-v3 and Ant-v3 tasks is shown in Table 2 and 3. The comparison was conducted using both the LIF and SRM neuron models. Across all degraded conditions, the cumulative rewards of the host SNN show clear improvements after integrating DWAM. Specifically, SAN-DWAM consistently outperforms SAN, with the most noticeable gains observed under challenging input perturbations based on the LIF model. Similarly, BDETT-DWAM achieves higher rewards than BDETT, particularly under severe weight degradation using the SRM model. These findings, combined with the earlier results from the obstacle avoidance task, further validate the generalization capability of DWAM. It not only enhances performance in specific tasks but also demonstrates robustness across diverse task domains.

Homeostasis Metrics The impact of various degraded conditions on the firing rate changes of the host SNN and

Table 2: Quantitative performance of the Half Cheetah-v3 task under degraded conditions, where higher reward indicates better performance, and lower HM_m and HM_{std} values reflect improved homeostasis.

Name	MIN				MAX				GN			
	LIF		SRM		LIF		SRM		LIF		SRM	
	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}
PopSAN	9993	0.035/0.036	10354	0.029/0.027	10251	0.028/0.027	10511	0.030/0.027	4050	0.085/0.073	3641	0.095/0.084
PopSAN-DWAM	10155	0.032/0.034	10413	0.026/0.025	10306	0.027/0.026	10639	0.027/0.025	4139	0.083/0.072	3705	0.093/0.083
BDETT	9906	0.020/ 0.027	11050	0.017/0.018	9756	0.023/0.028	10723	0.018/0.018	3425	0.057/0.066	3701	0.086/0.071
BDETT-DWAM	10003	0.019/0.027	11149	0.016/0.017	9864	0.021/0.027	10868	0.017/0.017	3498	0.057/0.065	4037	0.076/0.061
8-bit Loihi weight												
Name	LIF				SRM				LIF			
	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}	Reward	HM_m/HM_{std}
	PopSAN	10576	0.003/0.003	10793	0.003/0.003	7158	0.055/0.046	9671	0.023/0.021	4467	0.105/0.088	7255
PopSAN-DWAM	10651	0.003/0.003	10807	0.003/0.002	7194	0.055/0.045	9882	0.021/0.019	4845	0.103/0.087	7522	0.063/0.053
BDETT	10253	0.002/0.002	11278	0.005/0.006	8885	0.017/0.017	10691	0.015/0.013	1584	0.091/0.097	8167	0.049/ 0.038
BDETT-DWAM	10327	0.002/0.002	11414	0.004/0.006	8962	0.016/0.017	11048	0.014/0.010	1759	0.088/0.094	8219	0.047/0.038

Table 3: Quantitative performance of the Ant-v3 task under degraded conditions, where higher reward indicates better performance, and lower HM_m and HM_{std} values reflect improved homeostasis.

Name	MIN				MAX				GN			
	LIF		SRM		LIF		SRM		LIF		SRM	
	Reward	HM_m/HM_{std}										
PopSAN	3521	0.092/0.066	5687	0.057/0.060	4795	0.079/0.077	5430	0.051/0.049	2160	0.125/0.091	2109	0.093/0.080
PopSAN-DWAM	3794	0.066/0.060	5709	0.055/0.059	4900	0.079/0.76	5581	0.048/0.047	2541	0.105/0.087	2153	0.092/0.080
BDETT	4186	0.039/0.061	4275	0.058/0.052	3638	0.049/0.071	4510	0.064/0.063	2182	0.070/0.093	2275	0.090/0.073
BDETT-DWAM	4214	0.039/0.061	4388	0.056/0.050	3691	0.049/0.071	4516	0.063/0.062	2230	0.069/0.092	2517	0.080/0.062
8-bit Loihi weight												
Name	LIF				SRM				LIF			
	Reward	HM_m/HM_{std}										
	PopSAN	5904	0.008/0.011	6044	0.007/0.007	1324	0.185/0.113	5580	0.025/0.021	1580	0.124/0.108	4091
PopSAN-DWAM	5948	0.007/0.010	6050	0.006/0.005	1519	0.145/0.097	5627	0.021/0.019	1873	0.122/0.102	4219	0.077/0.062
BDETT	4732	0.003/0.004	5732	0.008/0.010	3626	0.023/0.028	4939	0.040/0.035	1937	0.055/0.072	4949	0.040/0.035
BDETT-DWAM	4774	0.003/0.003	5822	0.007/0.009	3637	0.022/0.028	5558	0.017/0.015	2157	0.053/0.071	5030	0.037/0.033

the host SNN integrated with DWAM in the HalfCheetah-v3 and Ant-v3 tasks are shown in Table 2 and 3, respectively. In all degraded conditions, the homeostasis metrics of the host SNN showed consistent reductions following the incorporation of DWAM.

These results further confirm that the proposed DWAM is an effective method for introducing biologically-inspired homeostatic mechanisms into SNNs. By minimizing deviations in neuronal firing rates, it stabilizes network performance, even under challenging conditions.

5 Conclusion

In this paper, we proposed the DWAM as a biologically-inspired approach to integrating homeostatic mechanisms into SNNs. Drawing on the principles of the BCM theory, DWAM enables SNNs to dynamically regulate synaptic strength, maintaining network stability across varying levels of neuronal activity. We evaluated the proposed method across multiple tasks and under various degradation conditions. The experimental results demonstrated that DWAM

not only enhances the performance of SNNs lacking homeostatic mechanisms in challenging scenarios but also seamlessly integrates with existing homeostatic methods, improving the performance of host SNNs already equipped with such features. This method broadens the scope of SNN design, offering a novel perspective for enhancing network robustness and adaptability in a biologically plausible manner. Nevertheless, the interplay and joint effects of multiple homeostatic mechanisms within SNNs remain poorly understood. If such mechanisms could not only coexist but also complement each other as they do in biological systems, more effective and resilient forms of homeostasis might be achieved — a promising direction for future research.

References

- Benuskova, L.; Diamond, M. E.; and Ebner, F. F. 1994. Dynamic synaptic modification threshold: computational model of experience-dependent plasticity in adult rat barrel cortex. *Proceedings of the National Academy of Sciences*, 91(11): 4791–4795.

- Bienenstock, E. L.; Cooper, L. N.; and Munro, P. W. 1982. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2(1): 32–48.
- Brockman, G. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540*.
- Choi, J.; Park, K.; Kim, M.; and Seok, S. 2019. Deep reinforcement learning of navigation in a complex and crowded environment with a limited field of view. In *2019 International Conference on Robotics and Automation (ICRA)*, 5993–6000. IEEE.
- Desai, N. S. 2003. Homeostatic plasticity in the CNS: synaptic and intrinsic forms. *Journal of Physiology-Paris*, 97(4-6): 391–402.
- Ding, J.; Dong, B.; Heide, F.; Ding, Y.; Zhou, Y.; Yin, B.; and Yang, X. 2022. Biologically inspired dynamic thresholds for spiking neural networks. *Advances in Neural Information Processing Systems*, 35: 6090–6103.
- Driscoll, H. E.; Muraro, N. I.; He, M.; and Baines, R. A. 2013. Pumilio-2 regulates translation of Nav1. 6 to mediate homeostasis of membrane excitability. *Journal of Neuroscience*, 33(23): 9644–9654.
- Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 1587–1596. PMLR.
- Gerstner, W. 1995. Time structure of the activity in neural network models. *Physical Review E*, 51(1): 738–758.
- Gerstner, W.; and Kistler, W. M. 2002. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.
- Giachello, C. N.; and Baines, R. A. 2017. Regulation of motoneuron excitability and the setting of homeostatic limits. *Current opinion in neurobiology*, 43: 1–6.
- Gupta, A.; and Saurabh, S. 2024. Unsupervised learning in a ternary SNN using STDP. *IEEE Journal of the Electron Devices Society*, 12: 211–220.
- Hao, Y.; Huang, X.; Dong, M.; and Xu, B. 2020. A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule. *Neural Networks*, 121: 387–395.
- Keck, T.; Hübener, M.; and Bonhoeffer, T. 2017. Interactions between synaptic homeostatic mechanisms: an attempt to reconcile BCM theory, synaptic scaling, and changing excitation/inhibition balance. *Current opinion in neurobiology*, 43: 87–93.
- Kheradpisheh, S. R.; Ganjtabesh, M.; Thorpe, S. J.; and Masquelier, T. 2018. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99: 56–67.
- Kim, T.; Hu, S.; Kim, J.; Kwak, J. Y.; Park, J.; Lee, S.; Kim, I.; Park, J.-K.; and Jeong, Y. 2021. Spiking neural network (snn) with memristor synapses having non-linear weight update. *Frontiers in computational neuroscience*, 15: 646125.
- Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9): 1659–1671.
- Marder, E.; and Goaillard, J.-M. 2006. Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7): 563–574.
- Marder, E.; and Prinz, A. A. 2002. Modeling stability in neuron and network function: the role of activity in homeostasis. *Bioessays*, 24(12): 1145–1154.
- Mease, R. A.; Famulare, M.; Gjorgjieva, J.; Moody, W. J.; and Fairhall, A. L. 2013. Emergence of adaptive computation by single neurons in the developing cortex. *Journal of Neuroscience*, 33(30): 12154–12170.
- Meng, Y.; Jin, Y.; and Yin, J. 2011. Modeling activity-dependent plasticity in BCM spiking neural networks with application to human behavior recognition. *IEEE transactions on neural networks*, 22(12): 1952–1966.
- Olshausen, B. A.; and Field, D. J. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583): 607–609.
- O’Leary, T.; Williams, A. H.; Franci, A.; and Marder, E. 2014. Cell types, network homeostasis, and pathological compensation from a biologically plausible ion channel expression model. *Neuron*, 82(4): 809–821.
- Pan, W.; Zhao, F.; Zeng, Y.; and Han, B. 2023. Adaptive structure evolution and biologically plausible synaptic plasticity for recurrent spiking neural networks. *Scientific Reports*, 13(1): 16924.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.
- Roy, K.; Jaiswal, A.; and Panda, P. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784): 607–617.
- Shrestha, S. B.; and Orchard, G. 2018. SLAYER: Spike Layer Error Reassignment in Time. In *Advances in Neural Information Processing Systems*, volume 31, 1412–1421.
- Stringer, C.; Pachitariu, M.; Steinmetz, N.; Carandini, M.; and Harris, K. D. 2019. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765): 361–365.
- Sun, J.; and Du, P.-F. 2021. Predicting protein subchloroplast locations: the 10th anniversary. *Frontiers of Computer Science*, 15(2): 152901.
- Tang, G.; Kumar, N.; and Michmizos, K. P. 2020. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware. *arXiv preprint arXiv:2003.01157*.
- Tang, G.; Kumar, N.; Yoo, R.; and Michmizos, K. P. 2020. Deep Reinforcement Learning with Population-Coded Spiking Neural Network for Continuous Control. *arXiv preprint arXiv:2010.09635*.
- Turrigiano, G. G.; and Nelson, S. B. 2004. Homeostatic plasticity in the developing nervous system. *Nature reviews neuroscience*, 5(2): 97–107.

Wang, Y.; Dong, B.; Zhang, Y.; Zhou, Y.; Mei, H.; Wei, Z.; and Yang, X. 2023. Event-enhanced multi-modal spiking neural network for dynamic obstacle avoidance. In *Proceedings of the 31st ACM International Conference on Multimedia*, 3138–3148.

Wang, Y.; Mei, H.; Bao, Q.; Wei, Z.; Shou, M. Z.; Li, H.; Dong, B.; and Yang, X. 2024. Apprenticeship-inspired elegance: synergistic knowledge distillation empowers spiking neural networks for efficient single-eye emotion recognition. *arXiv preprint arXiv:2407.09521*.

Xu, M.; Liu, F.; and Pei, J. 2022. Endowing spiking neural networks with homeostatic adaptivity for APS-DVS bi-modal scenarios. In *Companion Publication of the 2022 International Conference on Multimodal Interaction*, 12–17.

Zhang, J.; Dong, B.; Zhang, H.; Ding, J.; Heide, F.; Yin, B.; and Yang, X. 2022. Spiking transformers for event-based single object tracking. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 8801–8810.

Zhang, J.; Zhang, M.; Wang, Y.; Liu, Q.; Yin, B.; Li, H.; and Yang, X. 2025. Spiking neural networks with adaptive membrane time constant for event-based tracking. *IEEE Transactions on Image Processing*.

Supplementary Note 1: Bioinspired Dynamic Energy-Temporal Threshold (BDETT)

The BDETT provides a dynamic adjustment of neuron spiking thresholds in Spiking Neural Networks (SNNs) to emulate biological properties. This method draws inspiration from two biologically plausible observations:

Positive Correlation with Average Membrane Potential: The threshold increases as the neuron's average membrane potential grows, stabilizing the spiking activity in the presence of sustained input.

Negative Correlation with Preceding Depolarization Rate: The threshold decreases when the membrane potential rises rapidly, enhancing the neuron's sensitivity to sudden input changes.

To mathematically define the dynamic spiking threshold, the BDETT uses the following formulation (Ding et al. 2022):

$$\Theta_i^l(t+1) = \frac{1}{2}(E_i^l(t) + T_i^l(t+1)), \quad (10)$$

where $E_i^l(t)$ is the dynamic energy threshold (DET) of the neuron for ensuring a positive correlation, and $T_i^l(t+1)$ is the dynamic temporal threshold (DTT), which ensures a negative correlation.

DET: For the i -th neuron in the l -th layer at timestamp t , BDETT defines DET as (Ding et al. 2022):

$$E_i^l(t) = \eta(v_i^l(t) - V_m^l(t)) + V_\theta^l(t) + \ln(1 + e^{\frac{v_i^l(t) - V_m^l(t)}{\psi}}), \quad (11)$$

$$V_m^l(t) = \mu(v_i^l(t)) - 0.2(\max(v_i^l(t)) - \min(v_i^l(t))) \quad \text{for } i = 1, 2, \dots, n^l, \quad (12)$$

$$V_\theta^l(t) = \mu(\Theta_i^l(t)) - 0.2(\max(\Theta_i^l(t)) - \min(\Theta_i^l(t))) \quad \text{for } i = 1, 2, \dots, n^l, \quad (13)$$

where $v_i^l(t)$ is the neuron postsynaptic membrane potential at timestamp t ; μ is the mean operator; n^l is the total number of neurons in the l -th layer; and η and ψ are two hyperparameters, which are set empirically.

DTT: For the i -th neuron in the l -th layer, the DTT at timestamp $t+1$ is defined as (Ding et al. 2022):

$$T_i^l(t+1) = a + e^{\frac{-(v_i^l(t+1) - v_i^l(t))}{C}}, \quad (14)$$

$$a = -e^{-|\mu(\Theta_i^l(t))|} \quad \text{for } i = 1, 2, \dots, n^l. \quad (15)$$

This dynamic threshold design ensures that neurons achieve homeostasis by adapting their firing thresholds according to input patterns.

Supplementary Note 2: Spiking Neural Networks

Spiking Neural Networks (SNNs)

SNNs are a class of biologically inspired neural networks that aim to replicate the information processing mechanisms of biological neurons. Unlike traditional artificial neural networks, SNNs communicate through discrete spike events rather than continuous signals. These spikes are generated when the membrane potential of a neuron crosses a certain threshold, resulting in an action potential, a process known as firing. The state of a neuron in SNNs is typically categorized into resting, depolarization, and hyperpolarization. In the resting state, the neuron's membrane potential remains stable at a baseline level. Depolarization refers to an increase in membrane potential, making the neuron more likely to fire, while hyperpolarization describes a decrease, reducing the likelihood of firing.

A sequence of spikes is called a spike train and is defined as $s(t) = \sum_{t^{(f)} \in \mathcal{F}} \delta(t - t^{(f)})$, where \mathcal{F} represents the set of times at which the individual spikes occur (Shrestha and Orchard 2018). When an input spike train reaches a neuron, it influences the membrane potential through a combination of integration and decay processes. If the membrane potential exceeds a predefined threshold, the neuron emits an output spike, which in turn propagates to downstream neurons. This mechanism is the foundation of SNN computation. Over the years, various neuron models have been developed to simulate the dynamics of spiking neurons, with two commonly used ones being the Spike Response Model (SRM) (Gerstner 1995) and the Leaky Integrate-and-Fire (LIF) model (Gerstner and Kistler 2002).

Spike Response Model (SRM) The SRM describes a neuron's membrane potential as a dynamic response to incoming spikes. When an input spike train $s_i(t)$ arrives at a neuron, each spike triggers a response characterized by a kernel function $\epsilon(t)$, representing the temporal evolution of the depolarization process. The cumulative depolarization of the neuron is given by: $\sum_i w_i (\epsilon * s_i)(t)$, where w_i is the synaptic weight of the i -th input.

The SRM also incorporates the process of hyperpolarization by introducing a refractory kernel $\zeta(t)$. This kernel accounts for the temporary suppression of the neuron's ability to fire after a spike, effectively modeling the hyperpolarization phase: $(\zeta * s)(t)$, where the second term reduces the membrane potential after a spike, preventing immediate firing.

With an SRM, a feedforward SNN architecture with n_l layers can be defined. Given N^l incoming spike trains at layer l , $s_i^l(t)$, the forward propagation process of the network is mathematically defined as follows (Gerstner 1995; Shrestha and Orchard 2018):

$$v_i^{l+1}(t) = \sum_{j=1}^{N^l} w_{ij}(\varepsilon * s_j^l)(t) + (\zeta * s_i^{l+1})(t-1), \quad (16)$$

$$s_i^{l+1}(t) = f_s(v_i^{l+1}(t)), \quad (17)$$

$$f_s(v) : v \rightarrow s, s(t) := s(t) + \delta(t - t^{(f+1)}), \quad (18)$$

$$t^{f+1} = \min\{t : v(t) = \Theta, t > t^{(f)}\}, \quad (19)$$

where $f_s(\cdot)$ is a spike function and Θ is the membrane potential threshold, which is static and the same for all neurons in the network. This static threshold is the one that we replace with the proposed biologically plausible dynamic threshold.

Leaky Integrate-and-Fire (LIF) The LIF model is a simplified variant of the SRM. Its depolarization process is similar to that of the SRM, but it directly processes incoming spike trains while ignoring the spike response kernel. After the neuron fires, the membrane potential is immediately reset to a lower value, typically modeled as 0 or the resting potential, marking the onset of the hyperpolarization phase.

The forward propagation process of the network can be defined as:

$$v_i^{l+1}(t) = \sum_{j=1}^{N^l} w_{ij} s_j^l(t) + v_i^{l+1}(t-1) f_d(s_i^{l+1}(t-1)) + b_i^{l+1}, \quad (20)$$

$$s_i^{l+1}(t) = f_s(v_i^{l+1}(t)), \quad (21)$$

$$f_d(s(t)) = \begin{cases} D & s(t) = 0 \\ 0 & s(t) = 1, \end{cases} \quad (22)$$

where b_i^{l+1} is an adjustable bias that is learned to mimic a dynamic threshold behavior. However, the biases of this model are static during forwarding propagation. In contrast, the proposed dynamic threshold is dynamic and automatically adapts to membrane potentials.

Supplementary Note 3: Details of the toy example

The network structure and training process used in the toy example are identical to those in the robotic obstacle avoidance experiments (details can be found in Supplementary Note 5). However, DWAM is applied only to the neurons controlling linear velocity and their associated weights, with the firing rate and modification threshold tracked over time. The experimental scenario is a simplified version of the obstacle avoidance task, where all static and dynamic obstacles have been removed. In the absence of obstacles, the firing frequency of the linear velocity neurons approaches 1 to provide higher linear velocity, allowing the robot to reach the target point more quickly.

Supplementary Note 4: Homeostasis Metrics

We optimized the calculation of the homeostasis metrics. Existing work defines the homeostasis metrics as (Ding et al. 2022):

$$\begin{aligned} \text{FR}_m &= \mu(\text{FR}_m^p) \quad \text{for } p = 1, 2, \dots, P, \\ \text{FR}_{std}^m &= \mu(\text{FR}_{std}^p) \quad \text{for } p = 1, 2, \dots, P, \\ \text{FR}_{std}^s &= \sigma(\text{FR}_{std}^p) \quad \text{for } p = 1, 2, \dots, P, \\ \text{FR}_m^p &= \mu(f_i^{l,p}) \quad \text{for } i = 1, 2, \dots, N^l \quad l = 1, 2, \dots, L, \\ \text{FR}_{std}^p &= \sigma(f_i^{l,p}) \quad \text{for } i = 1, 2, \dots, N^l \quad l = 1, 2, \dots, L, \\ f_i^{l,p} &= \frac{\sum_{t^p=1}^{T^p} s_i^l(t^p)}{T^p} \end{aligned} \quad (23)$$

where, T^p is the time taken for the p -th trial and $f_i^{l,p}$ is the firing rate of the i -th neuron in the l -th layer during the p -th trial; FR_m^p denotes the mean firing rate of all neurons of an SNN during the p -th trial, and FR_{std}^p is the standard deviation of all neuron firing rates for an SNN during the p -th trial; FR_m is the mean neuron firing rate of an SNN across all P trials; FR_{std}^m is the average of P standard deviations, and each of them is the standard deviation of the neuron firing rates of an SNN during a single trial; FR_{std}^s denotes the standard deviation of the P standard deviations; FR_{std}^s represents the standard deviation across all P trials, while FR_{std}^m denotes the mean of these standard deviations. And the homeostasis of the network is measured by the difference between the homeostasis metrics in the base condition and the homeostasis metrics in the degraded condition. This

method measures the homeostasis of the network as a whole by taking the mean (standard deviation) and then the difference, but it ignores the homeostasis of individual neurons.

Consider a simple case: two trials were conducted on a network with four neurons n_1, n_2, n_3, n_4 under both base and degraded conditions. In the base condition, the firing rates of the four neurons in the two trials are $f_{i=1,2,3,4}^{p=1} = (0.3, 0.5, 0.5, 0.7)$ and $f_{i=1,2,3,4}^{p=2} = (0.2, 0.5, 0.5, 0.8)$; In the degraded condition, the firing rates of the four neurons in the two trials are $f_{i=1,2,3,4}^{p=1} = (0.7, 0.5, 0.5, 0.3)$ and $f_{i=1,2,3,4}^{p=2} = (0.8, 0.5, 0.5, 0.2)$. According to Eq. 23, ΔFR_m , ΔFR_{std}^m and ΔFR_{std}^s are equal to 0, suggesting the network has an optimal homeostasis. However, in reality, the firing rates of n_1 and n_4 fluctuate significantly.

In biology, each neuron has its unique firing rate set-point and homeostasis (Giachello and Baines 2017). In the biological experiment (Driscoll et al. 2013), the homeostasis is calculated by first taking the difference and then averaging (or finding the standard deviation), which can adequately measure the steady state of each neuron. Therefore, we optimized the calculation of the homeostasis metrics, defined as follows:

$$\begin{aligned} \text{HM}_m &= \mu(\text{abs}(f_i^{l,p_{base}} - f_i^{l,p_{degraded}})) \quad \text{for } p = 1, 2, \dots, P, \\ \text{HM}_{std} &= \sigma(\text{abs}(f_i^{l,p_{base}} - f_i^{l,p_{degraded}})) \quad \text{for } p = 1, 2, \dots, P, \\ f_i^{l,p} &= \frac{\sum_{t=1}^{T^p} s_i^l(t)}{T^p} \end{aligned} \quad (24)$$

where $f_i^{l,p}$, μ , and σ are defined in the same way as $f_i^{l,p}$ in Eq. 23; $\text{abs}()$ is taken as an absolute value; $f_i^{l,p_{base}}$ and $f_i^{l,p_{degraded}}$ are the firing rates of the i -th neuron in the l -th layer during the p -th trial in the base and degraded conditions, respectively.

Supplementary Note 5: Additional Details on Obstacle Avoidance Experiments

In this task, we control a mobile robot to navigate from a random starting point to a random target point within a given scene. The robot is equipped with a Robot Peak light detection and ranging (RPLIDAR) system, which serves as the obstacle detection sensor. It provides a 180-degree field of view composed of 18 range measurements. To evaluate DWAM’s ability to provide homeostasis to the network, we conducted experiments under various degraded conditions based on the baseline static environment. These conditions include the dynamic obstacles condition, degraded input conditions, and weight uncertainty conditions.

Dynamic Obstacles: 11 dynamic obstacles were added to the baseline static environment. These obstacles continuously move between two points. Since these obstacles were absent during training, this condition evaluates DWAM’s ability to maintain homeostasis in a constantly changing environment.

Degraded Input: Building on the dynamic obstacles condition, the robot’s LIDAR range data was perturbed using three methods: “0.2”: The range measurements of the 3rd, 9th, and 15th beams were set to 0.2 meters, reporting obstacles regardless of their actual presence. “6.0”: The range measurements of the same beams were set to 6.0 meters (the average visible range in the test environment), making them unable to detect any obstacles. “GN”: Gaussian noise (Choi et al. 2019), $\mathcal{N}(0, 1.0)$, was added to all 18 range measurements. These conditions test DWAM’s ability to maintain homeostasis in the presence of unstable input signals.

Weight Uncertainty: Building on the dynamic obstacles condition, The robot’s learned weights from the training phase were perturbed using three methods: “8-bit Loihi weight”: Weights were converted to 8-bit integers to simulate precision loss when deploying SNNs on neuromorphic chips. “GN weight”: Gaussian noise $\mathcal{N}(0, 0.05)$ was added to each weight. “30% zero weight”: Randomly setting 30% of the weights to zero in each layer. These scenarios evaluate DWAM’s ability to maintain homeostasis in the presence of unstable network parameters.

We selected two types of host SNNs: one without a homeostasis mechanism, the Spiking Action Network (SAN) (Tang, Kumar, and Michmizos 2020), and another with the best current homeostasis mechanism, the Bioinspired Dynamic Energy-Temporal Threshold (BDETT) (Ding et al. 2022). Due to the similarity of the tasks, we used almost the same experimental and training setup as Ding.

Experimental Setup

In the obstacle avoidance experiments, our evaluation baseline model and test environment are the SAN and its original simulated test environment, respectively. The SAN is a part of the spiking deep deterministic policy gradient (SDDPG) framework (Tang, Kumar, and Michmizos 2020), which is a fully connected four-layer SNN (i.e., three 256-neuron hidden layers and one two-neuron output layer). This network maps a state s of a robot to a control action a . Specifically, a state $s = \{G_{dis}, G_{dir}, \nu, \omega, L\}$ is encoded into 24 Poisson spike trains as inputs of the SAN, and each spike train has T timesteps. G_{dis} and G_{dir} are the relative 1-D distances from the robot to the goal and a 2-D direction (i.e., right and left directions), respectively; ν and ω are the robot’s 1-D linear and 2-D angular velocities (i.e., rightward and leftward angular velocities); L denotes the distance measurements obtained from a Robo Peak light detection and ranging (RPLIDAR) laser range scanner (range: 0.2-40 m), which has a field of view of 180 degrees with 18 range measurements, each with a 10-degree resolution.

The two output spike trains represent the robot's linear and angular velocities, respectively, and are decoded to control the robot via an action $a = \{\nu_L, \nu_R\}$, where ν_L and ν_R are the left and right wheel speeds of the differential-drive mobile robot, respectively (Tang, Kumar, and Michmizos 2020).

BDETT builds upon the implementation of SAN by replacing its firing threshold with the threshold calculated using the BDETT method (Ding et al. 2022). The specific calculation details can be found in Supplementary Note 1.

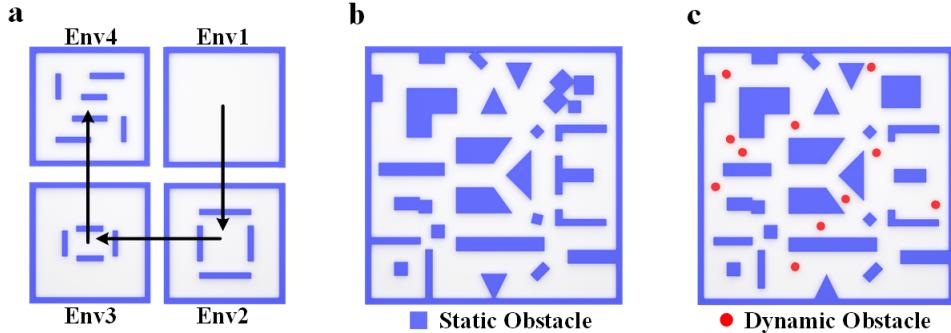


Figure 3: Illustrations of the training, static testing and dynamic testing environments. a. The training environments of the obstacle avoidance tasks. The training processes of all competing SNNs start from Env1 and end with Env4. b. Static testing environment. c. Dynamic testing environment. In addition to static obstacles, 11 dynamic obstacles are inserted.

Training Setup

The SAN and BDETT are trained with the original SDDPG framework. The training environments consist of four different maps, as shown in Figure 3a. In particular, during the training process, we set 100, 200, 300, and 400 start-goal pairs in the Env1, Env2, Env3, and Env4, respectively. The training starts from Env1 and ends with Env4. Following the training protocol described by Tang, Kumar, and Michmizos (2020), the hyperparameters related to training are set as follows: $D = 0.75$ for the LIF; $\eta = 0.01$ and $\psi = 4.0$ for the dynamic energy threshold (DET); $C = 3.0$ for the dynamic temporal threshold (DTT); $\tau_s = \tau_r = 1.0$ for the SRM; collision reward = -20 ; goal reward = 30 ; step reward = 15 ; goal l_2 distance threshold = 0.5 m; obstacle l_2 distance threshold = 0.35 m; ϵ ranges for Env1 to Env4 of $(0.9, 0.1)$, $(0.6, 0.1)$, $(0.6, 0.1)$, and $(0.6, 0.1)$, respectively; and corresponding ϵ -decays of 0.999 for the four environments. During the training procedure, we set the batch size to 256 and the learning rates to 0.00001 for both the actor and critic networks.

After training SAN and BDETT, we directly integrated BioDWAM and DWAM into the trained models without any further retraining, resulting in SAN-BioDWAM, SAN-DWAM, BDETT-BioDWAM, and BDETT-DWAM. We then conducted comparisons within each model group (i.e., SAN vs. SAN-DWAM vs. SAN-BioDWAM, and BDETT vs. BDETT-DWAM vs. BDETT-BioDWAM) to evaluate the effectiveness of DWAM. The hyperparameters of BioDWAM and DWAM were set as follows: $\psi = 0.00005$, $\alpha = 0.5$, $\zeta = 2.0$, and $n = 5.0$.

Notably, Ding et al. only provided the source code for the BDETT method based on the LIF model, including the network architecture and testing phase. As a result, we reproduced the BDETT implementation based on the SRM model by referring to the available LIF-based code. Moreover, due to the lack of training code, we reimplemented the training procedure following the protocol described by Ding et al. (Ding et al. 2022). In certain experiments, this reproduction process may have led to lower BDETT performance compared to that reported in the original paper. However, since BDETT-DWAM is built upon BDETT and our experimental setup is designed to compare the two under identical conditions, any performance degradation would affect both methods equally. Therefore, such degradation is offset and does not affect the validity of our evaluation of DWAM.

We use PyTorch (Paszke et al. 2019) to train and test all competing SNNs with an i7-14700K CPU and an NVIDIA GTX 4090 GPU. We direct the readers to the SAN algorithm (Tang, Kumar, and Michmizos 2020) and BDETT algorithm (Ding et al. 2022) for details.

Experimental Results

Table 4 presents the success rates and homeostasis metrics of various methods in both static and dynamic scenarios. σ represents the variance. Table 5 presents the success rates and homeostasis metrics of various methods under different degraded conditions.

Understanding the Functional Role of DWAM

To investigate how DWAM contributes to network robustness, we randomly selected one trajectory from the obstacle avoidance task as a toy example. Both SAN and SAN-DWAM models were tested along the same start-to-goal path under noise-free and noisy(GNW) conditions. During each run, we recorded the firing rates of all hidden-layer neurons at each time step. These

Table 4: Quantitative performance of obstacle avoidance in static and dynamic obstacle scenarios.

Type	Name	LIF		SRM	
		SR (σ)	SR (σ)	SR (σ)	SR (σ)
Static obstacle	SAN	97.5% (1.0)	97.3% (1.2)		
	SAN-BioDWAM	91.9% (1.7)	83.3% (3.4)		
	SAN-DWAM	98.4% (0.9)	98.5% (1.3)		
	BDETT	97.7% (0.5)	98.3% (0.9)		
	BDETT-BioDWAM	78.0% (1.9)	38.3% (3.2)		
	BDETT-DWAM	98.4% (1.2)	98.4% (0.9)		
Dynamic obstacle	SAN	85.9% (2.4)	87.3% (2.5)		
	SAN-BioDWAM	81.5% (3.6)	67.2% (2.9)		
	SAN-DWAM	90.1% (1.3)	88.9% (1.9)		
	BDETT	91.1% (1.5)	80.1% (2.9)		
	BDETT-BioDWAM	70.8% (7.0)	42.7% (1.9)		
	BDETT-DWAM	92.0% (1.8)	87.3% (1.6)		

recordings yielded four groups of neuronal activity: (1) SAN (no noise), (2) SAN (with noise), (3) SAN-DWAM (no noise), and (4) SAN-DWAM (with noise), which were then analyzed using PCA.

As shown in Figure 4a, the PCA embeddings of SAN under noisy conditions (blue) deviate noticeably from those under normal conditions (green), indicating that noise substantially alters neuronal dynamics. In contrast, Figure 4b shows that for SAN-DWAM, the noisy and noise-free data are closely aligned in the PCA space, suggesting that DWAM helps preserve stable neuronal activation despite noise. These findings demonstrate that DWAM enhances the stability and noise robustness of the network by maintaining more consistent firing patterns across conditions.

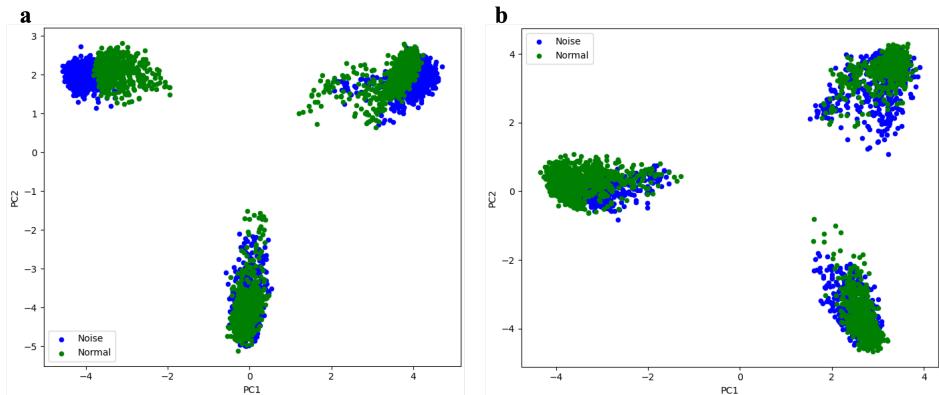


Figure 4: **PCA comparison of SAN and SAN-DWAM under noise.** Each plot shows PCA projections of hidden-layer neuronal activity during a randomly selected trajectory. Green and blue points denote normal and noisy conditions, respectively.

Supplementary Note 6: Additional Details on Continuous Control Experiments

The continuous control tasks include two subtasks of varying difficulty. HalfCheetah-v3 features a 17-dimensional observation space and a 6-dimensional action space, while Ant-v3 has a 110-dimensional observation space and an 8-dimensional action space. The inclusion of tasks with different levels of complexity allows for a more comprehensive evaluation of DWAM’s performance in continuous control scenarios. In these two subtasks, the objective is to control a six-joint half-mechanical cheetah (a quadruped robot) to move forward on a plane for 1000 steps while maximizing its forward velocity. Similar to the obstacle avoidance task, we conducted experiments under various degraded conditions based on the baseline environment. These experiments include degraded input conditions, and weight uncertainty conditions.

Degraded Input: We introduced three types of disturbances to the observation data collected by the robot: “MIN”: In each trial, a random dimension of the observation space (which could represent a joint position or velocity) is replaced with its observable minimum value (-inf). “MAX”: Similar to the “MIN” condition, except that the selected dimension is replaced with its observable maximum value (inf). “GN”: In each trial, Gaussian noise sampled from $\mathcal{N}(0, 0.1)$ is added to every dimension

of the observation space.

Weight Uncertainty: The conditions here are the same as those used in the obstacle avoidance task.

we selected two types of host SNNs: one without a homeostasis mechanism, the population-coded SAN (PopSAN) (Tang et al. 2020), and another with the best current homeostasis mechanism, BDETT (Ding et al. 2022). Due to the similarity of the tasks, we used almost the same training setup as Ding et al. (Ding et al. 2022).

Training Setup

The adopted population-coded SAN (PopSAN) is trained by using the twin-delayed deep deterministic policy gradient (TD3) off-policy algorithm (Fujimoto, Hoof, and Meger 2018) and the following hyperparameter settings: $D = 0.75$ for the LIF; $\eta = 0.01$ and $\psi = 6.0$ for the DET; $C = 3.0$ for the DTT; and $\tau_s = \tau_r = 1.0$ for the SRM. Compared to the settings of the obstacle avoidance tasks, the only different setting is the value of ψ for the DET. Following the training protocol of the PopSAN, we set the batch size to 100 and the learning rates to 0.0001 for both the actor and critic networks. The reward discount factor is set to 0.99, and the maximum length of the replay buffer is set to 1 million.

We construct PopSAN-BioDWAM, PopSAN-DWAM, BDETT-BioDWAM, and BDETT-DWAM using the same method as the obstacle avoidance task. The BioDWAM and DWAM hyperparameters were set as follows: for the Ant-v3 environment, $\psi = 0.0001$; for the Half Cheetah-v3 environment, $\psi = 0.001$, $\alpha = 0.5$, $\zeta = 2.0$, and $n = 5.0$.

Notably, Ding et al. only provided the source code for the BDETT method based on the LIF model, including the network architecture and testing phase. As a result, we reproduced the BDETT implementation based on the SRM model by referring to the available LIF-based code. Moreover, due to the lack of training code, we reimplemented the training procedure following the protocol described by Ding et al. (Ding et al. 2022). In certain experiments, this reproduction process may have led to lower BDETT performance compared to that reported in the original paper. However, since BDETT-DWAM is built upon BDETT and our experimental setup is designed to compare the two under identical conditions, any performance degradation would affect both methods equally. Therefore, such degradation is offset and does not affect the validity of our evaluation of DWAM.

We use PyTorch (Paszke et al. 2019) to train and test all competing SNNs with an i7-14700K CPU and an NVIDIA GTX 4090 GPU.

Reward

Table 6 and 8 present the success rates and homeostasis metrics of various methods in the Mujoco HalfCheetah-v3 Task and Ant-v3 Task tasks under standard scenarios, respectively. Table 7 and 9 summarize the success rates and homeostasis metrics of various methods in the Mujoco HalfCheetah-v3 Task and Ant-v3 Task under different degraded conditions, respectively.

Table 5: Quantitative performance and homeostasis metrics of obstacle avoidance under various degraded conditions.

Type	Name	LIF			SRM		
		SR (σ)	HM _m	HM _{std}	SR (σ)	HM _m	HM _{std}
0.2	SAN	88.2% (1.0)	0.10895	0.11510	86.0% (0.6)	0.12981	0.11743
	SAN-BioDWAM	83.9% (1.9)	0.11736	0.11677	66.4% (3.0)	0.13217	0.11770
	SAN-DWAM	90.6% (1.0)	0.09109	0.11282	87.3% (3.2)	0.12123	0.11233
	BDETT	70.7% (2.4)	0.10358	0.10895	64.0% (3.1)	0.07301	0.09742
	BDETT-BioDWAM	58.1% (1.2)	0.10476	0.10410	27.6% (2.8)	0.07530	0.09635
	BDETT-DWAM	82.6% (3.2)	0.10331	0.10331	75.0% (1.5)	0.07028	0.09562
6.0	SAN	71.7% (2.4)	0.01844	0.02180	75.6% (2.9)	0.02707	0.02807
	SAN-BioDWAM	75.3% (7.0)	0.01888	0.02219	57.3% (2.5)	0.02719	0.02799
	SAN-DWAM	84.3% (2.8)	0.01827	0.02162	76.1% (2.4)	0.02629	0.02740
	BDETT	79.6% (2.0)	0.01593	0.02198	71.6% (3.7)	0.02079	0.02239
	BDETT-BioDWAM	51.2% (2.4)	0.01775	0.02180	29.3% (2.5)	0.02198	0.02234
	BDETT-DWAM	82.8% (4.7)	0.01521	0.02147	80.0% (2.8)	0.01744	0.02205
GN	SAN	87.6% (4.1)	0.03307	0.03137	81.6% (3.6)	0.03944	0.03419
	SAN-BioDWAM	82.8% (1.4)	0.03400	0.03149	68.8% (1.5)	0.03792	0.03420
	SAN-DWAM	88.4% (3.9)	0.03057	0.03049	83.3% (2.5)	0.03728	0.03381
	BDETT	87.5% (2.0)	0.02464	0.02598	73.4% (3.0)	0.01941	0.02268
	BDETT-BioDWAM	63.2% (2.9)	0.02729	0.02622	47.1% (8.5)	0.02357	0.02339
	BDETT-DWAM	89.3% (1.4)	0.02374	0.02572	76.6% (2.2)	0.01917	0.02199
8-bit Loihi weight	SAN	85.4% (2.0)	0.00525	0.00499	85.5% (3.7)	0.00363	0.00346
	SAN-BioDWAM	81.3% (1.5)	0.00468	0.00440	82.9% (4.0)	0.03535	0.03891
	SAN-DWAM	90.1% (1.2)	0.00378	0.00344	84.7% (5.1)	0.00392	0.00362
	BDETT	91.1% (0.5)	0.00279	0.00284	83.3% (1.4)	0.00270	0.00238
	BDETT-BioDWAM	65.8% (1.0)	0.00319	0.00277	34.2% (1.7)	0.01475	0.01062
	BDETT-DWAM	93.8% (0.2)	0.00275	0.00265	86.9% (1.9)	0.00215	0.00223
GN weight	SAN	55.2% (6.4)	0.14374	0.16831	0.0% (0.0)	0.16573	0.15850
	SAN-BioDWAM	0.7% (1.2)	0.15385	0.16528	0.0% (0.0)	0.16647	0.16080
	SAN-DWAM	70.4% (0.7)	0.10598	0.12725	0.5% (1.0)	0.16165	0.14818
	BDETT	79.6% (1.6)	0.06210	0.08107	79.0% (1.9)	0.04273	0.05052
	BDETT-BioDWAM	53.5% (4.3)	0.05816	0.07334	30.3% (2.9)	0.04603	0.04489
	BDETT-DWAM	89.8% (1.7)	0.05189	0.07328	84.4% (2.7)	0.03500	0.04207
30% Zero weight	SAN	66.3% (13.2)	0.10150	0.10449	5.7% (3.9)	0.10763	0.10460
	SAN-BioDWAM	44.3% (13.5)	0.09764	0.10746	7.5% (5.7)	0.10776	0.10445
	SAN-DWAM	77.4% (2.0)	0.09283	0.10234	14.0% (14.7)	0.10545	0.10042
	BDETT	56.7% (9.9)	0.05747	0.07007	56.0% (4.8)	0.05638	0.06649
	BDETT-BioDWAM	37.6% (6.5)	0.05318	0.06247	25.7% (3.3)	0.05862	0.06159
	BDETT-DWAM	78.3% (2.9)	0.04580	0.05699	81.6% (2.5)	0.05351	0.05937

Table 6: Quantitative Performance of Mujoco HalfCheetah-v3 Task under standard testing condition.

Type	Name	LIF		SRM	
		Reward (σ)			
Standard	PopSAN	10618 (74)	10824 (70)		
	PopSAN-BioDWAM	10372 (68)	10827 (64)		
	PopSAN-DWAM	10643 (55)	10833 (61)		
	BDETT	10237 (89)	11312 (73)		
	BDETT-BioDWAM	10255 (81)	11306 (64)		
	BDETT-DWAM	10280 (65)	11336 (90)		

Table 7: Quantitative performance and homeostasis metrics of Mujoco HalfCheetah-v3 Tasks under various degraded conditions.

Type	Name	LIF			SRM		
		Reward (σ)	HM_m	HM_{std}	Reward(σ)	HM_m	HM_{std}
MIN	PopSAN	9993 (97)	0.03491	0.03565	10354 (83)	0.02872	0.02731
	PopSAN-BioDWAM	10002 (128)	0.03646	0.03742	10338 (104)	0.02809	0.02670
	PopSAN-DWAM	10155 (81)	0.03205	0.03372	10413 (55)	0.02569	0.02506
	BDETT	9906 (44)	0.01992	0.02684	11050 (110)	0.01701	0.01774
	BDETT-BioDWAM	9926 (76)	0.02010	0.02775	11018 (70)	0.01689	0.01768
	BDETT-DWAM	10003 (100)	0.01919	0.02678	11149 (87)	0.01633	0.01726
MAX	PopSAN	10251 (86)	0.02823	0.02706	10511 (57)	0.03009	0.02675
	PopSAN-BioDWAM	10186 (86)	0.02832	0.02675	10562 (80)	0.02824	0.02568
	PopSAN-DWAM	10306 (129)	0.02721	0.02614	10639 (76)	0.02744	0.02537
	BDETT	9756 (76)	0.02273	0.02836	10723 (91)	0.01783	0.01763
	BDETT-BioDWAM	9796 (114)	0.02229	0.02806	10766 (73)	0.01770	0.01746
	BDETT-DWAM	9864 (110)	0.02142	0.02721	10868 (92)	0.01745	0.01731
GN	PopSAN	4050 (80)	0.08455	0.07282	3641 (128)	0.09484	0.08403
	PopSAN-BioDWAM	4006 (148)	0.09100	0.07274	3693 (130)	0.09409	0.08418
	PopSAN-DWAM	4139 (157)	0.08301	0.07180	3705 (147)	0.09314	0.08277
	BDETT	3425 (175)	0.05746	0.06562	3701 (207)	0.08597	0.07145
	BDETT-BioDWAM	3403 (168)	0.05756	0.06618	3757 (224)	0.08662	0.07184
	BDETT-DWAM	3498 (182)	0.05711	0.06547	4037 (177)	0.07580	0.06112
8-bit Loihi weight	PopSAN	10576 (52)	0.00336	0.00279	10793 (56)	0.00309	0.00281
	PopSAN-BioDWAM	10388 (89)	0.00332	0.00301	10801 (97)	0.00354	0.00302
	PopSAN-DWAM	10651 (63)	0.00330	0.00270	10807 (27)	0.00278	0.00245
	BDETT	10253 (59)	0.00217	0.00241	11278 (128)	0.00501	0.00619
	BDETT-BioDWAM	10288 (102)	0.00328	0.00354	11247 (71)	0.00539	0.00589
	BDETT-DWAM	10327 (71)	0.00208	0.00231	11414 (83)	0.00444	0.00586
GN weight	PopSAN	7158 (182)	0.05511	0.04622	9671 (188)	0.02335	0.02089
	PopSAN-BioDWAM	7111 (351)	0.05669	0.04637	9622 (178)	0.02415	0.02130
	PopSAN-DWAM	7194 (342)	0.05511	0.04534	9882 (161)	0.02104	0.01938
	BDETT	8885 (116)	0.01662	0.01723	10691 (187)	0.01477	0.01324
	BDETT-BioDWAM	8952 (203)	0.01717	0.01710	10740 (124)	0.01500	0.01433
	BDETT-DWAM	8962 (150)	0.01630	0.01684	11048 (109)	0.01387	0.01013
30% Zero weight	PopSAN	4467 (999)	0.10544	0.08767	7255 (218)	0.06626	0.05548
	PopSAN-BioDWAM	4327 (597)	0.10757	0.08891	7225 (521)	0.06646	0.05623
	PopSAN-DWAM	4845 (698)	0.10305	0.08743	7522 (168)	0.06255	0.05270
	BDETT	1584 (555)	0.09059	0.09682	8167 (171)	0.04862	0.03843
	BDETT-BioDWAM	1425 (301)	0.09311	0.09820	8117 (310)	0.04753	0.04052
	BDETT-DWAM	1759 (347)	0.08837	0.09410	8219 (185)	0.04658	0.03798

Table 8: Quantitative Performance of Mujoco Ant-v3 Tasks under standard testing condition.

Type	Name	LIF		SRM	
		Reward (σ)			
Standard	PopSAN	5861 (70)	5997 (76)		
	PopSAN-BioDWAM	5889 (66)	5435 (128)		
	PopSAN-DWAM	5918 (61)	6002 (63)		
	BDETT	4736 (96)	5763 (58)		
	BDETT-BioDWAM	4735 (54)	5802 (65)		
	BDETT-DWAM	4761 (63)	5885 (84)		

Table 9: Quantitative performance and homeostasis metrics of Mujoco Ant-v3 Tasks under various degraded conditions.

Type	Name	LIF			SRM		
		Reward (σ)	HM_m	HM_{std}	Reward(σ)	HM_m	HM_{std}
MIN	PopSAN	3521 (550)	0.09214	0.06623	5687 (82)	0.05702	0.06007
	PopSAN-BioDWAM	3539 (216)	0.08624	0.07033	4967 (57)	0.06404	0.08363
	PopSAN-DWAM	3794 (141)	0.06646	0.06037	5709 (60)	0.05534	0.05897
	BDETT	4186 (97)	0.03949	0.06111	4275 (198)	0.05800	0.05183
	BDETT-BioDWAM	4033 (113)	0.04830	0.06803	4272 (166)	0.05973	0.06091
	BDETT-DWAM	4214 (113)	0.03926	0.06099	4388 (162)	0.05622	0.05048
MAX	PopSAN	4795 (71)	0.07907	0.07715	5430 (101)	0.05057	0.04851
	PopSAN-BioDWAM	4808 (104)	0.07942	0.07685	4369 (176)	0.06989	0.07343
	PopSAN-DWAM	4900 (95)	0.07875	0.07641	5581 (118)	0.04825	0.04737
	BDETT	3638 (70)	0.04927	0.07121	4510 (79)	0.06381	0.06269
	BDETT-BioDWAM	2779 (91)	0.08293	0.07792	4465 (90)	0.06342	0.06264
	BDETT-DWAM	3691 (92)	0.04878	0.07057	4516 (69)	0.06299	0.06224
GN	PopSAN	2160 (336)	0.12501	0.09122	2109 (96)	0.09279	0.08004
	PopSAN-BioDWAM	2216 (255)	0.11825	0.09035	2088 (107)	0.10151	0.10768
	PopSAN-DWAM	2541 (214)	0.10536	0.08702	2153 (111)	0.09187	0.07959
	BDETT	2182 (129)	0.07004	0.09268	2275 (101)	0.09007	0.07314
	BDETT-BioDWAM	1714 (83)	0.09131	0.09800	2045 (90)	0.09975	0.08309
	BDETT-DWAM	2230 (130)	0.06949	0.09205	2517 (176)	0.08012	0.06231
8-bit Loihi weight	PopSAN	5904 (43)	0.00827	0.01115	6044 (74)	0.00720	0.00672
	PopSAN-BioDWAM	5936 (51)	0.00799	0.01023	5452 (100)	0.00682	0.00613
	PopSAN-DWAM	5948 (53)	0.00724	0.00953	6050 (56)	0.00574	0.00501
	BDETT	4732 (113)	0.00290	0.00401	5732 (66)	0.00786	0.00958
	BDETT-BioDWAM	4699 (69)	0.00470	0.00466	5727 (64)	0.00806	0.00930
	BDETT-DWAM	4774 (60)	0.00257	0.00337	5822 (23)	0.00715	0.00908
GN weight	PopSAN	1324 (862)	0.18511	0.11284	5580 (92)	0.02456	0.02102
	PopSAN-BioDWAM	1420 (538)	0.16646	0.11603	5017 (97)	0.02663	0.02395
	PopSAN-DWAM	1519 (700)	0.14485	0.09669	5627 (104)	0.02150	0.01941
	BDETT	3626 (251)	0.02328	0.02773	4939 (109)	0.03999	0.03502
	BDETT-BioDWAM	2905 (87)	0.03265	0.03145	5550 (73)	0.02810	0.02518
	BDETT-DWAM	3637 (280)	0.02216	0.02762	5558 (96)	0.01728	0.01520
30% Zero weight	PopSAN	1580 (653)	0.12429	0.10770	4091 (258)	0.07828	0.06308
	PopSAN-BioDWAM	1703 (775)	0.13470	0.10780	3884 (312)	0.08358	0.07115
	PopSAN-DWAM	1873 (764)	0.12242	0.10198	4219 (435)	0.07659	0.06250
	BDETT	1937 (566)	0.05455	0.07207	4949 (106)	0.03999	0.03502
	BDETT-BioDWAM	1313 (490)	0.06653	0.08142	4890 (163)	0.04159	0.03685
	BDETT-DWAM	2157 (504)	0.05336	0.07075	5030 (150)	0.03711	0.03292