Electronic Research Archive

ERA, 34(3): 1559–1584.
DOI: 10.3934/era.2026071
Received: 05 January 2026
Revised: 04 February 2026
Accepted: 12 February 2026
Published: 13 February 2026

https://www.aimspress.com/journal/ERA

Research article

A brain-inspired framework integrating Entorhinal-Hippocampal-Prefrontal circuits for adaptive navigation in mobile robots

Yishen Liao[1], Chenghua Wang[1], Naigong Yu[2,*], Hejie Yu[3] and Kun Xiao[1]

[1] Modern Industry School of Virtual Reality, Jiangxi University of Finance and Economics, Nanchang 330013, China
[2] School of Information Science and Technology, Beijing University of Technology, Beijing 100124, China
[3] Department of Precision Instrument, Tsinghua University, Beijing 100084, China

* **Correspondence:** Email: yunaigong@bjut.edu.cn.

Abstract: Autonomous navigation in complex, dynamic, and unstructured environments poses a major challenge for mobile robots due to limitations in traditional methods. Inspired by rodent entorhinal-hippocampal-prefrontal circuits, we proposed a brain-inspired navigation framework that leverages specialized spatial cells. The framework integrates entorhinal-hippocampal path integration from self-motion cues with prefrontal action selection via Sn-Plast modulated spiking neural networks (SNNs) to form initial goal-directed habits. Upon reaching the goal, CA1 place cells self-organize to optimize navigation routes, generating supervisory firing rate sequences that consolidate refined routes in the SNNs via feedback, enabling stable habits and rapid reshaping under task changes. Additionally, a chaining strategy decomposes complex long-distance tasks into sequential subtasks, enabling independent local habit formation and seamless integration into globally efficient routes. 3D robot simulation experiments showed superior performance over baseline algorithms in convergence speed and route efficiency. Moreover, the method exhibits rapid habit reshaping under dynamic task changes, reducing exploration episodes by 52.3%–62.2% and shortening route length by 5.6%–7.7% in multi-compartment environments via a hierarchical chaining strategy. Moreover, the framework can operate in real time at approximately 12 Hz, demonstrating its feasibility for robotic applications.

Keywords: brain-inspired navigation; spiking neural networks; Entorhinal-Hippocampal-Prefrontal; mobile robots; spatial cells

1. Introduction

Autonomous navigation in complex, dynamic, and unstructured environments represents one of the most fundamental and persistent challenges in mobile robotics. Advances in sensors, computing power, and algorithms have greatly improved robot navigation performance. However, challenges persist, including low path efficiency, localization inaccuracies, and poor adaptability to complex or dynamic environments [1]. Biological systems, evolved over millions of years, exhibit highly efficient, flexible, and adaptive navigation capabilities [2]. Through integrated mechanisms of perception, memory, learning, and decision-making, animals excel at navigating diverse and challenging settings. Studying these biological strategies not only deepens insights into neural principles but also inspires breakthroughs in robotic navigation [3]. Consequently, biomimetic approaches have gained traction as a promising avenue to overcome the limitations of traditional methods.

Rodents like rats demonstrate exceptional navigational prowess through specialized spatial cells and neural circuits in the entorhinal-hippocampal-prefrontal system [4]. Key spatial cells include grid cells and boundary vector cells (BVCs) in the entorhinal cortex [5,6], place cells in the hippocampus [7], and action cells in the prefrontal cortex [8]. These cells collectively enable precise path integration from self-motion cues in the entorhinal cortex, which converges in hippocampal CA3 for position representation [9]. Downstream, CA1 place cells dynamically remap their firing fields to refine and shorten routes based on experience and environmental structure [10]. When navigation tasks change, rats adapt through coordinated prefrontal-hippocampal interactions [11], enabling strategy shifts. Hippocampal remapping adjusts spatial representations to accommodate new obstacles or goals, facilitating rapid habit reshaping without requiring complete relearning [12]. In addition, in complex or extensive environments, rats further employ multi-segment chaining navigation, hierarchically decomposing long routes into sequential subtasks with localized habit formation in place cell sequences and prefrontal orchestration, enabling efficient modular adaptation and local reshaping without requiring full retraining of the global route [13].

Inspired by these neural mechanisms, we proposed a brain-inspired framework integrating entorhinal-hippocampal-prefrontal circuits for adaptive navigation in mobile robots. It enables adaptive navigation by integrating entorhinal-hippocampal path computation and prefrontal action mapping via Sn-Plast rule, followed by route optimization and feedback-driven SNN consolidation to support dynamic habit and multi-segment task learning. The major innovations are:

(i) A CA1 place cell-based route optimization module utilizing self-organization via global all-to-all and local sequential connectivity, which iteratively shortens trajectories from explorations, escapes local optima, integrates BVC inputs for obstacle avoidance, and produces globally efficient routes.

(ii) A memory consolidation feedback loop that derives theoretical firing rate sequences from optimized routes and uses them as supervisory signals to retrain the CA3-prefrontal SNN, establishing stable navigational habits for refined routes and enabling rapid habit reshaping under changing goals or obstacles.

(iii) A chaining navigation strategy that hierarchically decomposes complex long-distance tasks into sequential subtasks, enabling independent habit formation and optimization per segment, thereby reducing training complexity and enhancing scalability in large-scale environments.

The remainder of the manuscript is organized as follows. In Section 2, we review related works and highlight the limitations of other studies. In Section 3, we provide a detailed description of the proposed model. In Section 4, we present the experimental results and analysis. In Section 5, we

conclude the paper.

2. Related works

2.1. SLAM-based navigation methods

Traditional SLAM methods, including probabilistic approaches like EKF-SLAM, and graph optimization like ORB-SLAM, rely on building incremental maps using sensors to enable localization and path planning in unknown environments [14,15]. Bio-inspired variants of SLAM, such as RatSLAM [16] and its extensions (e.g., EM-SOL [17], RC-HMM [18], and Hybrid-NeuroSLAM [19]), draw inspiration from rodent hippocampal-entorhinal circuits, particularly place cells, grid cells, and head direction cells, to construct incremental experience maps or cognitive maps directly from sensory inputs. These methods replace traditional metric maps with topo-metric representations that mimic the brain's continuous attractor networks and loop-closure mechanisms, achieving robust localization and path planning in unknown or perceptually aliased environments.

Despite these advances, all bio-inspired SLAM approaches remain fundamentally dependent on pre-constructed or incrementally built spatial representations. This inherent reliance severely restricts their capability to handle goals located beyond the currently explored region or in rapidly changing environments, as classical planning algorithms require the target to lie within known traversable space. Consequently, they exhibit limited adaptability in truly unstructured, large-scale, or highly dynamic scenarios where obstacles, goals, or environmental layouts frequently change without prior exploration [20].

2.2. Traditional reinforcement learning navigation methods

To overcome the limitations of SLAM-based navigation, which relies on pre-built maps and cannot handle goals beyond mapped regions, reinforcement learning (RL) has driven research on navigation habit formation via exploration and reward-driven path learning. Pioneering approaches to intrinsic motivation include methods such as Intelligent Adaptive Curiosity (IAC) for autonomous exploration [21], which was later extended to Robust IAC (R-IAC) for path planning in unstructured environments, incorporating obstacle avoidance and goal adaptation [22]. Similarly, classic model-based reinforcement learning frameworks like Dyna, with its Q-learning instantiation known as Dyna-Q, have been effectively applied to domains such as robot navigation [23]. While traditional RL faces the curse of dimensionality, deep reinforcement learning (DRL) effectively mitigates this in robotic navigation. Notable advances include vector-based navigation using grid-like representations [24], deep Q-network (DQN) [25], and deep deterministic policy gradient (DDPG) [26]. However, DRL methods in navigation suffer from severe sample inefficiency, typically requiring massive environmental interactions and millions of training steps.

2.3. Bio-inspired navigation habit formation methods

To address the limitations of DRL in robotic navigation, approaches have adopted mapless navigation strategies inspired by the neural circuitry between the prefrontal cortex and hippocampus. Early implementations used simple feedforward networks to link place cells and action cells, with

weights updated through Q-learning rule during exploration [27]. Subsequent work introduced spiking neural networks (SNNs) with continuous-time differential learning and spike timing dependent plasticity (STDP) for reward prediction, markedly accelerating goal-directed behavior [28]. Following the development of the neuromodulatory model, the Sn-Plast framework incorporates acetylcholine and dopamine into reward-modulated STDP, enabling robust adaptation in dynamic environments [29]. Ang et al.'s subsequent work modeled the role of acetylcholine in Sn-Plast, demonstrating enhanced behavioral exploration and greater STDP flexibility, which outperformed other reward-based plasticity rules, strengthening its biological basis for flexible navigation habit formation [30].

In recent years, several researchers have further modeled hippocampal-prefrontal interactions by incorporating experience replay mechanism to refine navigation path and accelerate habit formation. For instance, Wang et al. constructed a brain-inspired memory consolidation model using a motivated developmental network, significantly enhancing memory plasticity by simulating neurotransmitter regulation and neocortex-dependent memory transitions [31]. This enables robots to achieve more efficient learning and robust decision-making during the navigation process. Similarly, Lv et al. further advanced the hippocampal-prefrontal inspired navigation method using meta-reinforcement learning (Meta-RL) [32]. Through closed-loop interaction, dynamic coupling between planning replay and strategy evaluation has been achieved, significantly improving the sample efficiency, convergence speed, and adaptability to uncertainty in robot navigation.

Despite these advances in modeling hippocampal-prefrontal interactions and experience replay, most methods lack biologically inspired mechanisms for experience-dependent place field remapping and geometric route optimization in the hippocampus. This deficiency results in generated navigation routes that remain tortuous without spontaneous adaptation to environmental structures for route shortening. Furthermore, these methods typically treat long-distance navigation as a monolithic process, lacking a hierarchical decomposition strategy inspired by rodent multi-segment chaining navigation [33], thereby limiting training efficiency and scalability in large-scale environments.

3. Materials and methods

3.1. Overall structure of the method

In this section, we detail the overall architecture and operational workflow of the proposed navigation method (Figure 1). During exploration, self-motion cues drive the entorhinal-hippocampal model [34] for path integration, generating activity in CA3 place cells that project to prefrontal action cells via an SNN for direction selection. Moreover, SNN synaptic weights are updated via the Sn-Plast rule. Upon goal arrival, hippocampal CA1 place cells self-organize using a hybrid energy function that combines global coordination, local sequencing, and BVCs inputs to optimize the navigation route, after which the theoretical firing rate sequences are calculated as the supervisory signals to retrain the SNN with eligibility-trace plasticity, thereby consolidating stable navigational habits. For dynamic tasks, detected changes prompt re-exploration from the disruption point, rapidly reshaping habits by reusing prior knowledge and re-engaging the "Sn-Plast–Optimization–Feedback" cycle. In particular, in complex or large-scale environments, a hierarchical chaining mechanism decomposes long-distance tasks into sequential subtasks based on salient transitions, enabling independent habit formation and optimization for each segment before seamless global integration, thereby reducing training complexity and enhancing scalability.

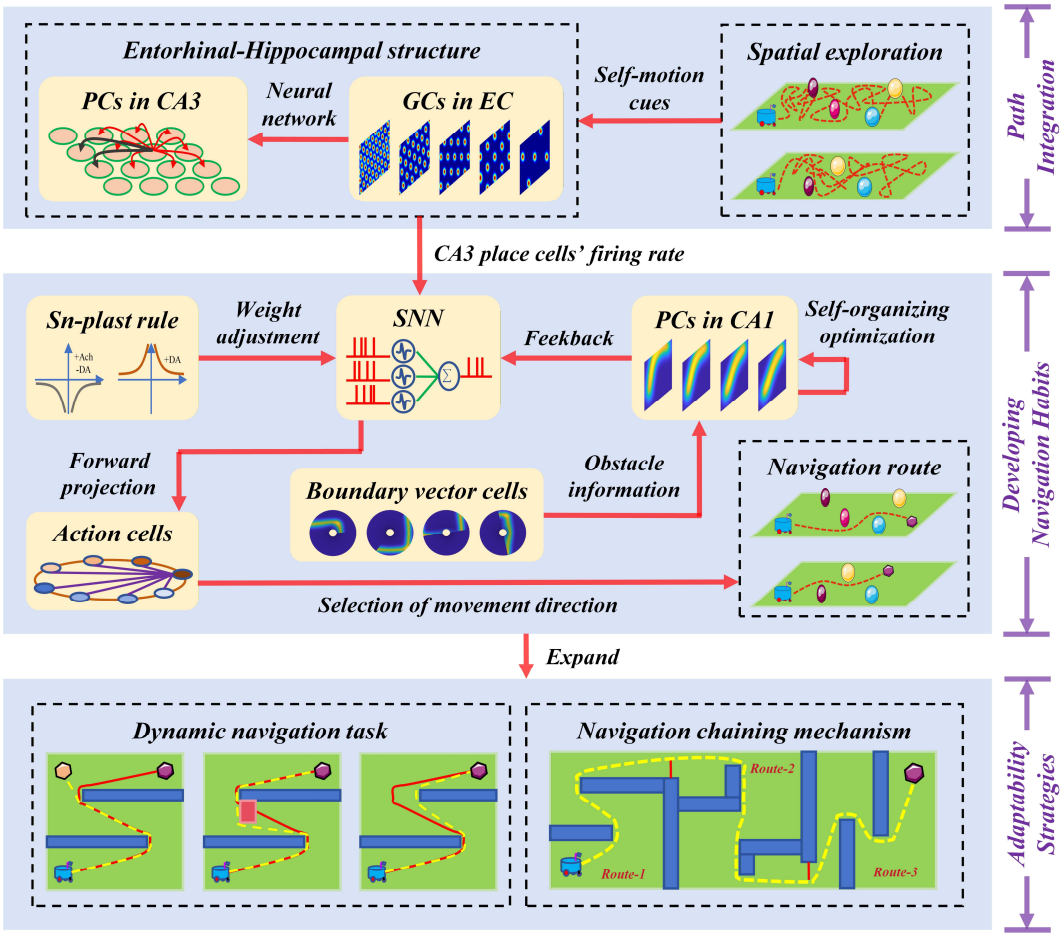


Figure 1. The comprehensive structure of the proposed method.

3.2. Hippocampal CA3-prefrontal navigation model

The dynamic interaction between the hippocampus and prefrontal cortex constitutes a critical determinant of future behavior. STDP-based SNNs are commonly employed to model information transfer between hippocampal CA3 place cells and prefrontal cortical action cells. In this study, the Sn-Plast model is used to simulate the learning process of rats in a goal-directed navigation task. The connection weight from CA3 place cells to action-cells SNNs is defined as w^{feed}, and the lateral connection weight between action cells is defined as w^{lat}. The membrane potential of the j-th action cell at time t is denoted $u_j(t)$, as described in Eq (1).

$$u_j(t) = Pcs(t) + Acs(t) + Spcs(t) \tag{1}$$

In Eq (1), $Pcs(t)$ denotes the spike train from CA3 place cells, $Acs(t)$ denotes the spike train from other action cells, and $Spcs(t)$ denotes the self-generated postsynaptic current, with explicit forms provided below:

$$Pcs(t) = \sum_i \sum_{\bar{t}_i \in F_i^{pc}, t > \hat{t}_j} w_{ji}^{\text{feed}} \cdot \epsilon(t - \bar{t}_i) \tag{2}$$

$$Acs(t) = \sum_{k, k \neq j} \bar{t}_k \in F_k^a, t > \hat{t}_j \sum_{jk}^{lat} \cdot \epsilon(t - \bar{t}_k) \tag{3}$$

$$Spcs(t) = \chi\Theta\left(t - \hat{t}_j\right) exp\left(-\frac{t-\hat{t}_j}{\tau_m}\right) \tag{4}$$

In Eqs (2) to (4), χ represents the refractory period, $\hat{t}_j$ is the last output spike time of the *i*-th place cell, $\bar{t}_i$ and $\bar{t}_k$ collect all spike arrival times from the *i*-th place cell and the *k*-th action cell, and $\Theta(t)$ is the Heaviside step. $\epsilon(t)$ is the postsynaptic potential, as described in Eq (5).

$$\epsilon(t) = \frac{\epsilon_0}{\tau_m - \tau_s}\left(e^{\frac{-t}{\tau_m}} - e^{\frac{-t}{\tau_s}}\right)\Theta(t) \tag{5}$$

In Eq (5), ϵ_0 is a fixed scaling factor, τ_s is the synaptic rise time, and τ_m stands for the membrane time constant. External pulse trains are delivered stochastically, with instantaneous firing rates governed by a time-varying Poisson process $\lambda_j\left(u_j(t)\right)$, as described in Eq (6).

$$\lambda_j\left(u_j(t)\right) = \lambda_0 \exp\left(\frac{u_j(t)-\theta_{me}}{\Delta u}\right) \tag{6}$$

In Eq (6), θ_{me} is the spiking threshold, Δu is the width of each injected current pulse, and λ_0 is the inverse of the mean inter-pulse interval when the membrane potential sits exactly at threshold. Distinct action cells encode different preferred movement directions. Define the preferred direction associated with the *j*-th action cell as a_j, as described in Eq (7).

$$a_j = a_0\left(sin\left(\frac{2j\pi}{N_{AC}}\right), cos\left(\frac{2j\pi}{N_{AC}}\right)\right) \tag{7}$$

In Eq (7), N_{AC} is the action cells quantity and a_0 is the motion speed. Then, define w_{jk}^{lat} as the connection weight between the *k*-th and *j*-th action cells, as described in Eq (8).

$$w_{jk}^{lat} = \frac{w_-}{N_{AC}} + w_+ \frac{f(j,k)}{N_{AC}} \tag{8}$$

In Eq (8), w_+ and w_- serve as the adjustable weights governing cyclic interactions, and $f(j,k)$ stands for the connected function, as described in Eq (9).

$$f(j,k) = \left(1 - \delta_{jk}\right)e^{\zeta\, cos\left(\theta_j - \theta_k\right)} \tag{9}$$

In Eq (9), ζ is the adjustment constant. It can be seen that action cells with similar preferred directions mutually excite each other, whereas those with dissimilar directions inhibit one another. This mechanism ensures that only a small subset of action cells is active at any time, thereby producing smoother trajectories for the robot. The navigation process requires continuous updating of the robot's position, a procedure that is driven by the activity of action cells, as described in Eq (10).

$$\Delta x(t) = \begin{cases} a(t) & \text{if } x(t+1) \text{ in the square} \\ d_{ft} \cdot u\left(x(t)\right) & \text{otherwise} \end{cases} \tag{10}$$

From Eq (10), it is evident that dynamic obstacle avoidance occurs continuously during navigation. If the robot's position $x(t+1)$ lies in free space, no special action is taken. However, once an obstacle is encountered at $x(t+1)$, the robot immediately moves a distance in the opposite direction. The mathematical expression for the action $a(t)$ is given by Eq (11).

$$a(t) = \frac{1}{N_{AC}} \sum_j (Y_j \circ \gamma)(t) a_j \tag{11}$$

In Eq (11), Y_j is the initial firing sequence of the j-th action cell, and γ stands for the filter; the mathematical expression are described in Eqs (12) and (13).

$$Y_j = \sum_{\bar{t}_j \in F_j^a} \delta(t - \bar{t}_j) \tag{12}$$

$$\gamma(t) = \frac{e^{\frac{-t}{\tau}\gamma} - e^{\frac{-t}{\nu\gamma}}}{\tau_\gamma - \nu_\gamma} \Theta(t) \tag{13}$$

Additionally, to mitigate excessive boundary effects, the connection weights between place cells corresponding to the encoding area and obstacle boundaries and the action cells are set to zero. The synaptic weights in the SNNs form the basis of the robot's navigation capability. Consequently, the synaptic plasticity is critical for the formation of navigational habits. The synaptic plasticity learning rule of the Sn-Plast is described in Eq (14).

$$\Delta w_{ji}(t) = \eta_{stdp} A \left(\sum_{\bar{t}_i \in F_i^{pc}} \sum_{\bar{t}_j \in F_j^a} W(\bar{t}_j - \bar{t}_i) \right)(t) \tag{14}$$

In Eq (14), η_{stdp} is the learning rate, A is the neuromodulatory signal, $\bar{t}_i$ and $\bar{t}_j$ are the all spike arrival times from the i-th place cell and the j-th action cell, F_i^{pc} and F_j^a are the time set, and $W(x) = e^{-|x|/\tau_{stdp}}$ stands for the STDP time window. The working principle of the Sn-Plast framework is illustrated in Figure 2.

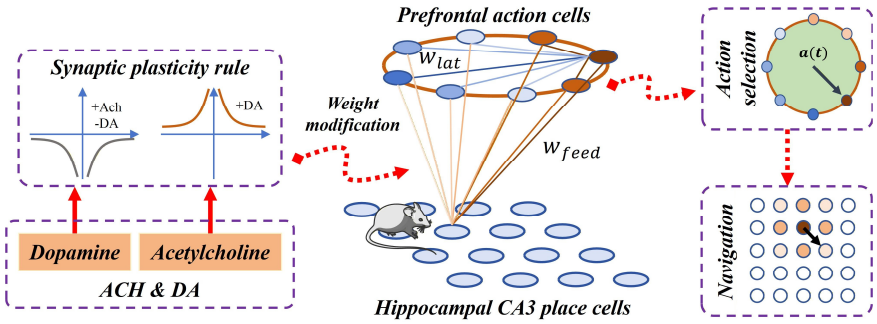


Figure 2. Working principle of the Sn-Plast framework.

3.3. Route optimization based on hippocampal CA1 place cells

After navigational habits are established, the learned trajectories are stored in hippocampal CA1 place cells. Nevertheless, the trajectories generated by Sn-Plast are not optimal. In contrast, mammals

can progressively refine navigation routes based on environmental structure. Physiological studies have shown that hippocampal CA1 place cells possess the capacity for dynamic adjustment, which can render the route more suitable for the navigation task. To make the navigation route shorter, smoother, and obstacle-avoiding, we propose a route optimization method based on the self-organizing mechanism of CA1 place cells. The fundamental principle is to drive the iterative update of the place cell receptive field centers by minimizing the composite neural energy function. First, define the neural population composed of n CA1 place cells as $\mathcal{N}$, as described in Eq (15).

$$\mathcal{N} = \{r_1, r_2, \ldots, r_n\}, r_i \in \mathbb{R}^2 \tag{15}$$

In Eq (15), r_i represents the place field center position of the i-th place cell. Based on the route constraints, we fix the starting place cell as r_{start} (corresponding to the start point) and the ending as r_{end} (corresponding to the goal). These two place cells remain fixed throughout the optimization process, ensuring that the route always starts from the correct start point and reaches the goal. Then, define the total neural energy function as $E(\mathcal{N})$, as described in Eq (16).

$$E(\mathcal{N}) = E_{local}(\mathcal{N}) + \beta E_{global}(\mathcal{N}) \tag{16}$$

In Eq (16), β is a non-negative hyperparameter that balances the relative importance of the local sequential constraint and the global coordination term. Large values of β promote globally smooth and compact paths, while smaller values emphasize a strict sequential order between adjacent site cells. E_{local} represents the local connectivity energy, which reflects the sequential activation pattern of CA1 place cells along the rat's movement route, ensuring temporal and spatial continuity of the optimized path, as described in Eq (17).

$$E_{\text{local}}(\mathcal{N}) = \sum_{i=1}^{n-1} \|r_{i+1} - r_i\| \tag{17}$$

In Eq (16), E_{global} represents the global coordination energy, which represents the all-to-all connectivity between place cells and penalizes excessive spatial separation or irregular distribution among place cell centers, as described in Eq (18).

$$E_{\text{global}}(\mathcal{N}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} \|r_j - r_i\| \tag{18}$$

In Eq (18), w_{ij} represents the connection strength between the i-th and the j-th place cells, computed based on their functional distance, as described in Eq (19). This ensures that nearby place cells interact strongly, while those farther apart exhibit significantly weaker interactions.

$$w_{ij} = \frac{\alpha}{|i-j|+1}, \alpha > 0 \tag{19}$$

In Eq (19), α represents the decay coefficient that controls the strength attenuation of long-range synaptic connections, simulating the natural weakening of synaptic strength with distance in biological neural systems and contributing to a smooth spatial gradient in the global coordination energy. Then, the neural energy function $E(\mathcal{N})$ is optimized via gradient dynamics, as described in Eq (20).

$$\nabla E(\mathcal{N}) = \nabla E_{local}(\mathcal{N}) + \beta \nabla E_{global}(\mathcal{N}) \tag{20}$$

In Eq (20), $\nabla E_{global}(\mathcal{N})$ represents the gradient of the global coordination energy, as described in Eq (21). It guides each place field centers to move toward a denser, smoother distribution.

$$\nabla_{r_k} E_{\text{global}} = \sum_{j \neq k} sign(j,k) w_{|k-j|} \frac{r_j - r_k}{\|r_j - r_k\|} \tag{21}$$

In Eq (21), $sign$ represents the symbolic function, which describes the formation of excitatory or inhibitory connections, as shown in Eq (22). It effectively balances attraction and repulsion for a cell distribution that is both dense and avoids collapse.

$$sign(j,k) = \begin{cases} +1 & if\ j > k\ (Forward\ Connection) \\ -1 & if\ j > k\ (Reverse\ Connection) \end{cases} \tag{22}$$

In Eq (22), $\nabla E_{local}(\mathcal{N})$ represents the gradient of the local connectivity energy. It guides the movement of each place field center to prevent sequential discontinuities and disorder. Define the shift vector between adjacent place cells as $d_k = r_{k+1} - r_k$, with length $l_k = \|d_k\|$, denoting the distance between the field centers of neighboring cells. Then, compute the synaptic adjustment direction from the local connectivity gradient, as shown in Eq (23).

$$\frac{\partial l_k}{\partial r_i} = \begin{cases} -\frac{d_k}{l_k} & if\ i = k \\ +\frac{d_k}{l_k} & if\ i = k+1 \\ 0 & otherwise \end{cases} \tag{23}$$

Therefore, the local connectivity gradient $\nabla_{r_i} E_{local}$ for a single place cell shown in Eq (24). The core mechanism uses unit direction vectors to pull each cell toward its sequential neighbors, reducing inter-cell distances.

$$\nabla_{r_i} E_{local} = \begin{cases} -\frac{r_2 - r_1}{\|r_2 - r_1\|} & i = 1 \\ \frac{r_i - r_{i-1}}{\|r_i - r_{i-1}\|} - \frac{r_{i+1} - r_i}{\|r_{i+1} - r_i\|} & 1 < i < n \\ \frac{r_n - r_{n-1}}{\|r_n - r_{n-1}\|} & i = n \end{cases} \tag{24}$$

Once the total synaptic gradient $\nabla E(\mathcal{N})$ is computed, it is used to update the receptive field centers of all CA1 place cells. The update rule is given by Eq (25).

$$\mathcal{N}^{(t+1)} = \mathcal{N}^{(t)} - \eta_{CA1} \nabla E\left(\mathcal{N}^{(t)}\right) \tag{25}$$

In Eq (25), η_{CA1} denotes the learning rate, which controls the step size of each update to prevent overly abrupt and non-plastic changes. To address the common presence of obstacles in environments, it is essential to integrate their influence into the route optimization. Therefore, the proposed method segments the optimization procedure based on the firing mechanism of BVCs. In the course of robotic exploration, LiDAR sensors collect obstacle information and discretized into a manageable format, which is described as $OBS = \{obs^1, obs^2, \ldots, obs^k\}$. Then, the obstacle information is supplied as input to the receptive field model of BVCs. Define the activation level of the *j*-th unit within the *i*-th

layer as Bu_streng_{ij}, as detailed in Eq (26).

$$Bu\_streng_{ij} = \sum_{k=1}^{Num_{obs}} e^{-\|obs^k - Bu\_pos_{ij}\|^2/\sigma_{Bu}} \tag{26}$$

In Eq (26), Num_{obs} denotes the size of the OBS, Bu_pos_{ij} denotes the position corresponding to the sensing unit, and σ_{Bu} denotes the sensitivity constant. This ensures that BVCs' firing depends on the robot's distance and direction relative to obstacles. During route optimization, the distance from each place field center to the nearest obstacle is computed in real time to ensure obstacle avoidance. Define the shortest distance from the field center of the *i*-th place cell to the nearest obstacle at time t as $l_{min}^i(t)$, as detailed in Eq (27).

$$l_{min}^i(t) = \min(\|p^i(t) - obs^k\|) \tag{27}$$

When a place cell's $l_{min}^i(t)$ falls below threshold l_{th}, it is anchored and ceases to update. Define $P_{fix}(t)$ as the set of place cells whose firing field centers have been anchored at time t. Equations (28) and (29) outline the update protocol for the set $P_{fix}(t)$. This prevents the route from approaching the obstacles too closely, ensuring a safe navigation margin.

$$P_{fix}(t+1) = sort(P_{fix}(t) \cup \{p^{i \in i_{fix}}(t+1)\}) \tag{28}$$

$$i_{fix} = \{l_{min}^i(t+1) \le l_{th} | l_{min}^i(t+1) - l_{min}^i(t) < 0\} \tag{29}$$

In Eqs (28) and (29), the "sort" function ranks elements in ascending order. For place cells with indices from i_{start} to i_{end}, the route optimization function is designated as $Cor(i, i_{start}, i_{end})$. The route optimization hot task can be represented by Eq (30).

$$Cor\left(i, p_{fix}^i(t), p_{fix}^{i+1}(t)\right) \ for \ p_{fix}^i(t), p_{fix}^{i+1}(t) \in P_{fix}(t) \tag{30}$$

By employing the above-described mechanism, the navigation route is gradually refined and shortened within obstacle-laden environments, reaching optimal performance. The operational procedure of the navigation route segmentation adjustment is shown in Figure 3.

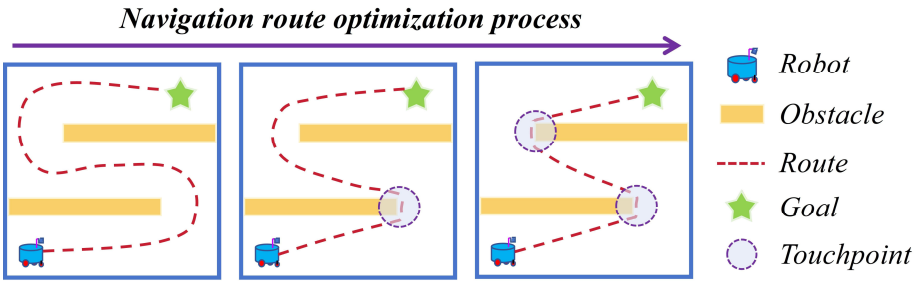


Figure 3. Working principle of the navigation framework.

3.4. Feedback of the navigation route

Upon completion of route optimization, the refined navigation route needs to be reinjected into the SNN to create a persistent memory. Define p^i and p^{i+1} as two adjacent place cells, $line^i$ as their connecting segment, and θ_{te} as the angle between $line^i$ and the x-axis, calculated in Eq (31).

$$\theta_{te} = arctan(\frac{p_y^i - p_y^{i+1}}{p_x^i - p_x^{i+1}}) \tag{31}$$

Then, the theoretical spiking sequences for action cells and CA3 place cells are derived based on the optimized route, as given in Eqs (32) and (33).

$$R_{ac}^j = \begin{cases} 1 & e^{-\vartheta_{AC} \cdot Delt(\theta_{te}, \frac{2j\pi}{N_{AC}})} \geq R_{AC}^{TH} \\ 0 & e^{-\vartheta_{AC} \cdot Delt(\theta_{te}, \frac{2j\pi}{N_{AC}})} < R_{AC}^{TH} \end{cases} \tag{32}$$

$$R_{pc}^k = \begin{cases} 1 & e^{-\frac{\left\| p^i - r_{pc}^k \right\|^2}{\sigma_{CA3}}} \geq rand \\ 0 & e^{-\frac{\left\| p^i - r_{pc}^k \right\|^2}{\sigma_{CA3}}} < rand \end{cases} \tag{33}$$

In Eq (32), N_{AC} indicates the number of action cells, $Delt$ is the angular difference computation function, R_{ac}^j denotes the firing rate of the *j*-th action cell, R_{AC}^{TH} represents the threshold, and ϑ_{AC} is the adjustment coefficient. In Eq (33), R_{pc}^k refers to the firing rate of the *k*-th place cell, r_{pc} denotes the population firing center of place cells, σ_{CA3} functions as the tuning parameter for place field radius, and $rand$ is a random value between 0 and 1. Next, the action $a(t)$ is derived from the action cells' firing rate via Eqs (11)–(13). Upon reaching the goal, the synaptic weights are updated by applying a plasticity rule that integrates STDP with eligibility traces, as shown in Eq (34).

$$\Delta w_{ji}(t) = \eta A \left(\sum_{\bar{t}_i \in F_i^{pc}} \sum_{\bar{t}_j \in F_j^a} W(\bar{t}_j - \bar{t}_i) \right) \circ \psi \Big)(t) \tag{34}$$

In Eq (34), ψ is the eligibility trace, and the remaining parameters are consistent with Eq (14). This enables weight adjustment to trace back to key transition points, filtering out short-term noise and ensuring stable long-term path encoding. It supports rapid adaptation to task or environmental changes without relearning, forming robust navigation habits.

3.5. Dynamic and multi-segment navigation tasks

To enhance adaptability and robustness, the proposed method extends support for dynamic navigation tasks and a chaining navigation mechanism. Navigation tasks often vary due to environmental changes or task adjustments. When the robot encounters a blockage along the original route (indicating a new obstacle), fails to locate the target at the original endpoint (signifying a goal change), or detects that an obstacle on the route has disappeared (indicating a reduction in environmental obstacles), it will re-explore the environment from the blockage point, the previous

endpoint, or the current detection point, thereby forming new navigational habits adapted to the updated task. Additionally, a chaining strategy is introduced to handle navigation tasks in highly complex environments. First, the complete task is adaptively decomposed into multiple independent segments by detecting salient environmental features (such as narrow passages detected via LiDAR or visual landmarks). Specifically, when the robot arrives at a location with narrow passages and visual landmarks, it is confirmed as a subtask boundary. This mechanism ensures that the end point of the subtask becomes the starting point of the next subtask, yielding $sub\_G_i = sub\_S_{i+1}$. The expression of task decomposition is shown in Eq (35).

$$sub\_tasks \leftarrow [(sub\_S_1, sub\_G_1), \dots, (sub\_S_i, sub\_G_i), \dots, (sub\_S_K, sub\_G_K)] \tag{35}$$

In Eq (35), sub_tasks denotes the complete task, and (sub_S_i, sub_G_i) represent the start and end points of the i-th subtask. sub_S_1 and sub_G_K represent the start and end points of the complete task. Then, habitual behavior patterns are progressively developed for each segment through training, enabling independent optimization of the SNN weights. Finally, these segments are concatenated for seamless continuity. This strategy reduces training complexity for long routes while enhancing modular scalability, suiting large-scale environments.

By applying all the computational methods presented in Section 3, the complete operation of the navigation framework can be constructed. The pseudocode of the navigation method is shown in Algorithm 1. Its core idea is to transform complex long-distance navigation into sequential processing of multiple independent subtasks through task decomposition (Line 2). Within each subtask (Lines 3–16), the robot sequentially performs spatial exploration (Line 4), path integration (Line 5), Sn-Plast calculation (Line 6), CA1-based route optimization, and feedback consolidation (Lines 5–10), and concatenates all optimized segments to form a seamless global path (Line 17).

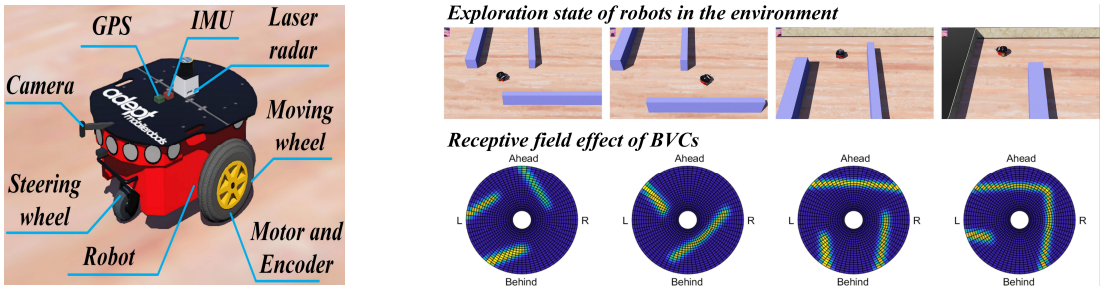
Algorithm 1 Pseudocode of the navigation method

1: Initialize population and parameters
2: Using Eq (35) to decompose the navigation task into K segments
3: **for** each segment i = 1 to K **do**
4: Start exploring from the starting point sub_S_i
5: Using Eqs (1)–(13) to calculate the motion command for the next moment
6: Using the entorhinal-hippocampal CA3 model for path integration
7: Using Eq (14) to adjust the synaptic weights of the SNN
8: **if** find the sub_G_i of the i-th subtask **do**
9: Using Eqs (28) and (29) to optimize the navigation route
10: Using Eqs (31)–(34) to feedback the optimized route
11: Continue with the next subtask
12: **end if**
13: **if** environmental changes or task adjustments **do**
14: Re-exploring from the obstruction intersection or prior endpoint
15: **end if**
16: **end for**
17: Concatenate all segments to achieve the complete task
18: **Return** the complete navigation route

4. Results and evaluations

4.1. Experimental procedure

To evaluate the performance of the proposed navigation method, we design robot platform simulation experiments in Webots. In the simulation environment, a Pioneer 3-DX mobile robot serves as the experimental platform. It is equipped with a camera for detecting navigation goals, a laser rangefinder for obstacle and boundary detection, an inertial measurement unit (IMU) for orientation, wheel encoders for linear and angular velocity measurements, and a GPS sensor for ground-truth positioning. The physical structure of the robot and its obstacle detection performance during environmental exploration are illustrated in Figure 4. It can be observed that the BVCs can accurately perceive obstacle information in the environment.



(a) Robot physical structure (Pioneer 3-DX robot's sensor configuration)

(b) Obstacle detection during environmental exploration (the ring represents the BVCs' receptive field. Higher brightness indicates the presence of the obstacle at that location)

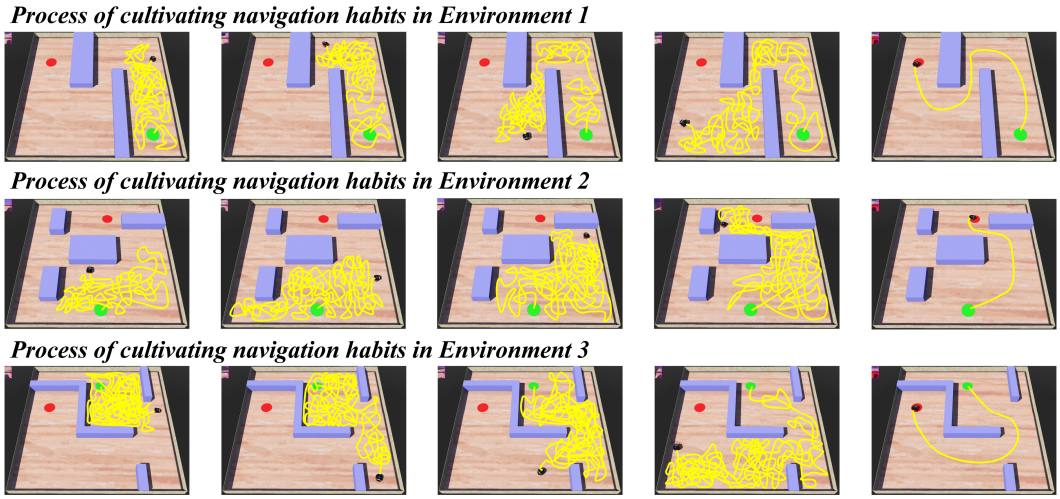
Figure 4. Pioneer 3-DX robot structure and obstacle detection in exploration.

Leveraging Webots' MATLAB interface, sensor data are processed in MATLAB to compute control commands, which are then sent back to the Webots simulation. Model execution and testing are conducted on a HP Victus 7 Laptop equipped with an NVIDIA RTX 3060 GPU, an Intel Core i7-11800k CPU, 16 GB of RAM, and running on the Windows 11 operating system. To accelerate computation, all matrix operations in the experiments are performed on the GPU. Parameter settings involve three key components: A hippocampal–prefrontal SNN model (Sn-Plast), route optimization method, and route feedback method. The parameter settings of Sn-Plast are configured following Ref. [28], and the remaining settings are as follows: Non negative hyperparameter β is set to 1, learning rate η_{CA1} is set to 0.1, decay coefficient α is set to 0.02, distance threshold l_{th} is set to 0.2m, the number of layers in BVCs' receptive network Bu_layer is set to 20, and the number of receptive units in each layer Su_ring is set to 72.

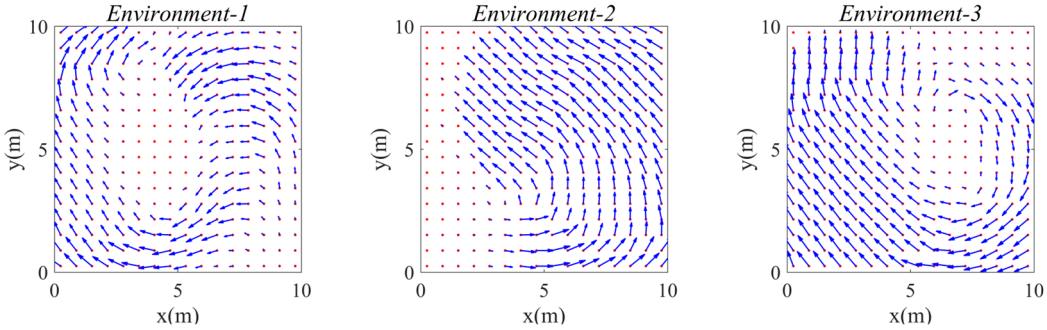
4.2. Navigation habit formation and route optimization experiments

Three distinct spatial arenas, each with an area of 10m × 10m, are constructed for navigation experiments. Within each arena, obstacles are placed, along with designated start and goal points for navigation. The Sn-plast model is employed to guide the robot in exploring the three spatial arenas to develop corresponding navigation habits. For each habit formation process, the number of exploration

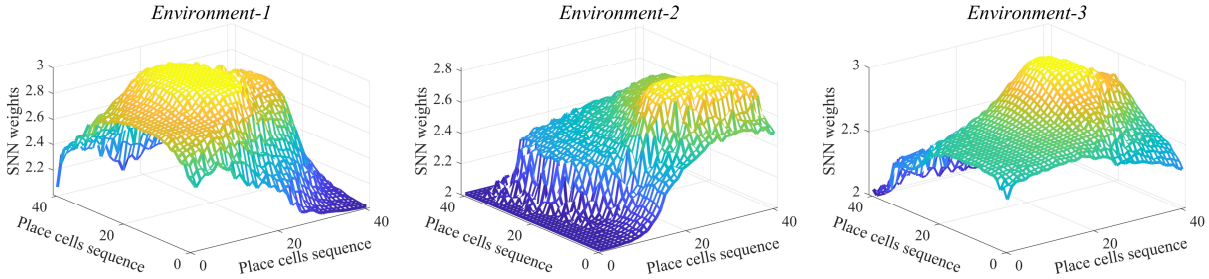
episodes is set to 30, and the maximum route length per exploration episode is limited to 80m. The process of navigation habit formation in the robot within the spatial arenas is illustrated in Figure 5(a), where panels from left to right represent an increase in the number of exploration episodes. Figure 5(b) depicts the distribution of movement policies after habit formation, while Figure 5(c) shows the distribution of SNN weights following habit formation.



(a) Process of navigation habit formation in the robot within the spatial arenas (the yellow line indicates the robot's movement route, with green and red circles marking the start and end points, respectively, while purple squares represent obstacles).



(b) Distribution of movement policies after habit formation (the arrow indicates the movement direction guided by the action cell at that location).



(c) Distribution of SNN weights after habit formation (areas near obstacles or unrelated regions have lower SNN weights).

Figure 5. Experimental results on cultivating navigation habits.

As shown in Figure 5(a), in the early stage of exploration, the robot's movement trajectories

exhibit pronounced randomness due to the stochastic nature of its environmental exploration strategy. At this phase, the motion policy and the distribution of neural network connection weights remain disorganized, hindering accurate and stable navigation. However, under the guidance of the Sn-Plast model, continuous environmental exploration enables the robot to gradually acquire the ability to navigate from the starting point to the target, forming task-adaptive navigation habits. Furthermore, as shown in Figures 5(b) and 5(c), the robot's motion policy demonstrates an obstacle avoidance tendency, with smaller connection weights near obstacles and regions distant from the navigation route. This indicates that the robot has developed the capability to accomplish collision-free navigation. However, the results reveal two limitations: 1) The model fails to converge rapidly, as the robot cannot reliably revisit previously reached goals; and 2) navigation routes are tortuous and suboptimal. Therefore, route optimization is necessary. Figure 6 illustrates the comparison of navigation routes before and after optimization, while Table 1 presents the route lengths. It is evident that the optimized routes are shorter, validating the effectiveness of the proposed model in route optimization.

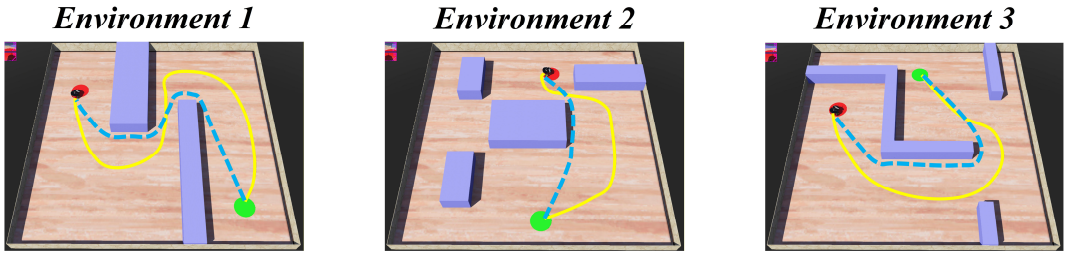


Figure 6. Comparison of navigation routes before and after optimization (the yellow line and blue dashed line represent the navigation routes before and after optimization).

Table 1. Statistical analysis of route lengths prior to optimization.

	Environment-1	Environment-2	Environment-3
Before optimization	18.94m	12.18m	16.97m
After optimization	12.48m	7.46m	13.39m

4.3. Comparison experiments

To highlight the advantages of the proposed method, it is compared with Dyna-Q [23], DQN [25], R-IAC [22], Meta-RL [32], and action curiosity based deep reinforcement learning (Ac-DRL) [35]. In each habit formation process, the number of exploration episodes is set to 30, and the maximum route length per episode is limited to 80 meters. Each algorithm guides the agent to route in three spatial regions. For ease of comparison, three quantitative metrics are used: Average route length, convergence probability, and average exploration episodes required to complete habit formation. The convergence criterion is as follows: If the robot discovers the goal and successfully reaches it in each of the subsequent four exploration episodes, the model is deemed converged (navigation habit formed); otherwise, it is considered non-converged. The convergence probability P_{con} is defined by Eq (36).

$$P_{con} = Sum_{con}/Sum_{find} \tag{36}$$

In Eq (36), Sum_{find} is the number of goal discoveries, and Sum_{con} is the number of times the

robot reaches the goal in four consecutive subsequent exploration episodes. The average route length L_{avg} is the mean length over all discovery episodes, as described in Eq (37).

$$L_{avg} = \sum_{j=1}^{Sum_{find}} L_j / Sum_{find} \tag{37}$$

In Eq (37), L_j denotes the route length for the j-th discovery of the goal. The average number of exploration episodes required for habit formation measures the mean episodes needed for the algorithm to converge, reflecting its convergence speed and robustness. For fair comparison, each algorithm is repeated 20 times in each environment. The variation of route length with the number of exploration episodes for each algorithm during navigation habit formation process is shown in Figure 7, and the results of performance indicators are shown in Table 2.

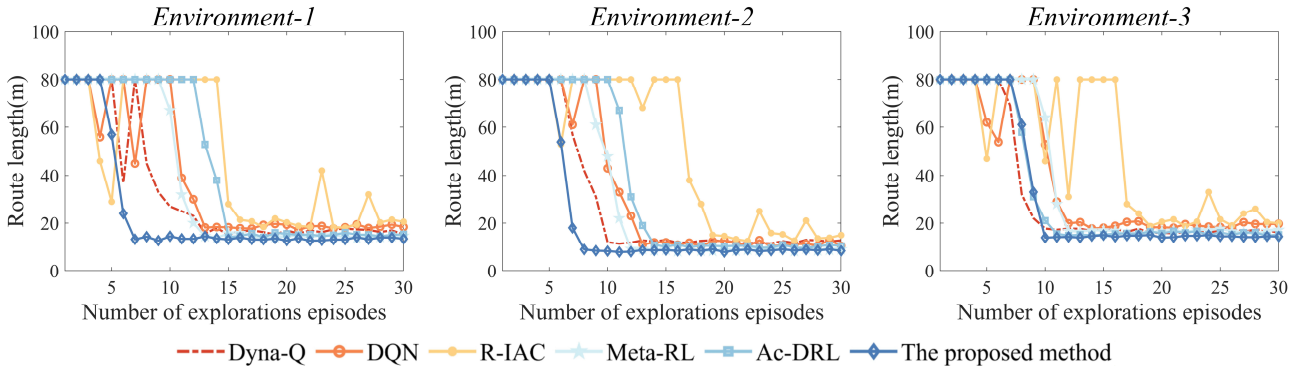


Figure 7. Variation of route length with the number of exploration episodes for each algorithm.

Table 2. Results of navigation performance indicators for various algorithms.

Environment	Algorithm	$L_{avg}(m)$	P_{con}	Average exploration episodes for habit formation
1	Dyna-Q	14.02	95.2%	9.8
	DQN	15.34	88.1%	10.7
	R-IAC	16.71	74.2%	15.1
	Meta-RL	13.67	90.6%	9.1
	Ac-DRL	14.19	92.2%	12.9
	The proposed method	**12.74**	**95.6%**	**7.5**
2	Dyna-Q	8.62	93.9%	8.8
	DQN	8.98	86.5%	9.5
	R-IAC	12.08	77.1%	14.2
	Meta-RL	8.41	91.9%	8.3
	Ac-DRL	8.74	94.0%	9.7
	The proposed method	**7.61**	**96.2%**	**7.9**
3	Dyna-Q	15.11	92.7%	8.1
	DQN	16.22	86.9%	10.3
	R-IAC	19.13	87.5%	11.8
	Meta-RL	15.43	96.8%	8.7
	Ac-DRL	14.94	93.3%	7.2
	The proposed method	**13.79**	**97.1%**	**7.0**

As shown in Figure 7 and Table 2, all evaluated algorithms enable the robot to develop task-specific navigation habits. Among them, R-IAC exhibits the lowest convergence speed and route efficiency, primarily due to its intrinsic curiosity mechanism, which prioritizes extensive exploration of novel or high prediction error states to minimize model uncertainty. This focus delays the exploitation of high-reward routes and results in frequent detours during navigation. By contrast, Dyna-Q and DQN achieve superior overall performance, as both are primarily driven by extrinsic rewards to directly maximize cumulative returns, enabling the agent to rapidly converge on high-reward trajectories. However, these methods lack the endogenous structural signals afforded by synaptic plasticity, which limits their capacity to efficiently convert sparse extrinsic rewards into persistent synaptic weight adjustments. Meta-RL and Ac-DRL show improvements in adaptability through experience replay mechanisms, achieving relatively high convergence probabilities. However, they yield longer average route lengths and require more exploration episodes compared to the proposed method, as they do not incorporate biologically inspired route optimization and memory consolidation feedback, leading to less efficient geometric refinement and slower habit stabilization. In comparison, the proposed method outperforms all baselines across environments, achieving the shortest route lengths, highest convergence probabilities, and fewest exploration episodes.

4.4. Ablation experiments

To verify the effectiveness of each module in the proposed method, the following ablation experiments are designed. Moreover, the proposed method consists of the Sn-Plast module, the Route Optimization module (RO), and the Route Feedback module (RF). To demonstrate the contribution of each module, ablation experiments are conducted to evaluate the impact of individual modules on performance. Specifically, Sn-Plast represents navigation habit formation without route optimization and feedback; Sn-Plast+RO denotes optimization of navigation routes without feedback; Sn-Plast+RF denotes feedback of navigation routes without optimization; and Sn-Plast+RO+RF corresponds to the complete proposed method. The parameter settings are consistent with the comparative experiment. The variation of route length with the number of exploration episodes for each algorithm during the navigation habit formation process is shown in Figure 8, and the statistical results of performance indicators are shown in Table 3.

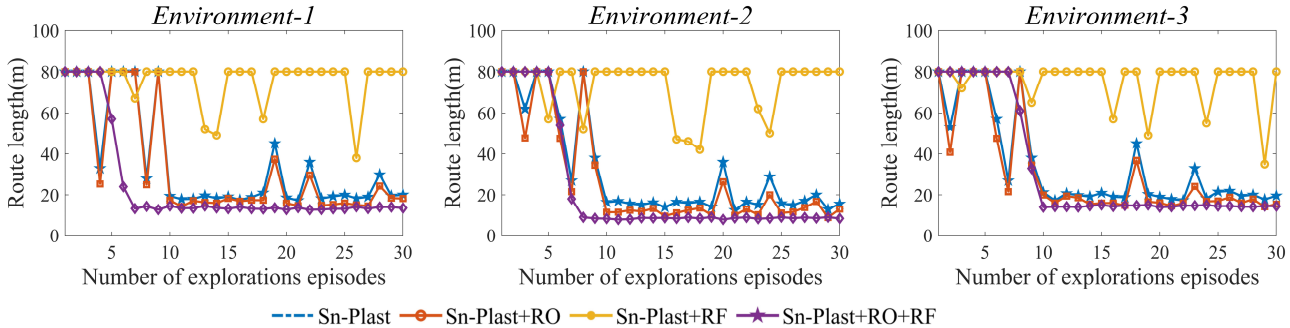


Figure 8. Variation of route length with the number of exploration episodes for each algorithm.

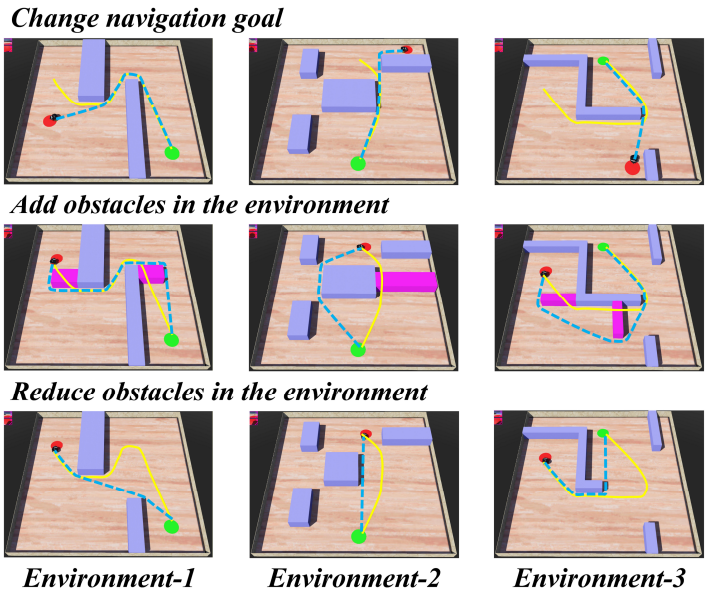
Table 3. Results of navigation performance indicators for various algorithms.

Environment	Algorithm	$L_{avg}(m)$	P_{con}	Average exploration episodes for habit formation
1	Sn-Plast	19.23	71.9%	14.9
	Sn-Plast+RO	13.11	71.9%	14.9
	Sn-Plast+RF	61.32	5.8%	N/A
	Sn-Plast+RO+RF	**12.74**	**95.6%**	**7.5**
2	Sn-Plast	13.03	78.7%	12.7
	Sn-Plast+RO	7.89	78.7%	12.7
	Sn-Plast+RF	55.24	6.9%	N/A
	Sn-Plast+RO+RF	**7.61**	**96.2%**	**7.9**
3	Sn-Plast	20.55	71.1%	16.7
	Sn-Plast+RO	15.12	71.1%	16.7
	Sn-Plast+RF	58.51	5.2%	N/A
	Sn-Plast+RO+RF	**13.79**	**97.1%**	**7.0**

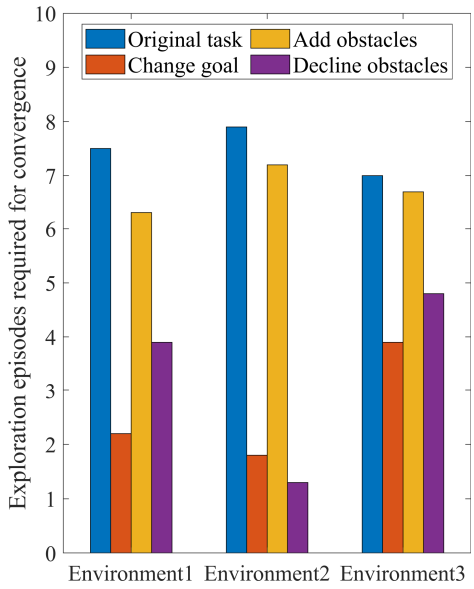
As shown in Figure 8 and Table 3, Sn-Plast+RO displays synchronized route length variations with Sn-Plast but yields shorter average lengths owing to its route optimization capability. Nevertheless, the absence of a feedback mechanism prevents the formation of stable memory for these optimized routes. Sn-Plast+RF performs the worst because it directly uses unoptimized, tortuous, and inefficient routes to generate supervisory signals for SNN feedback, severely disrupting or preventing stable convergence of navigation habits. However, Sn-Plast+RO+RF demonstrates the best navigation performance, achieving an optimal balance between convergence speed and route efficiency through the synergistic integration of RO and RF mechanisms. The RO component enables dynamic route optimization and shortening via biologically inspired synaptic plasticity, whereas the RF component reinforces and stabilizes retention of the optimized routes by incorporating extrinsic reward signals, thereby resolving the memory persistence limitation observed in Sn-Plast+RO.

4.5. Dynamic navigation experiments

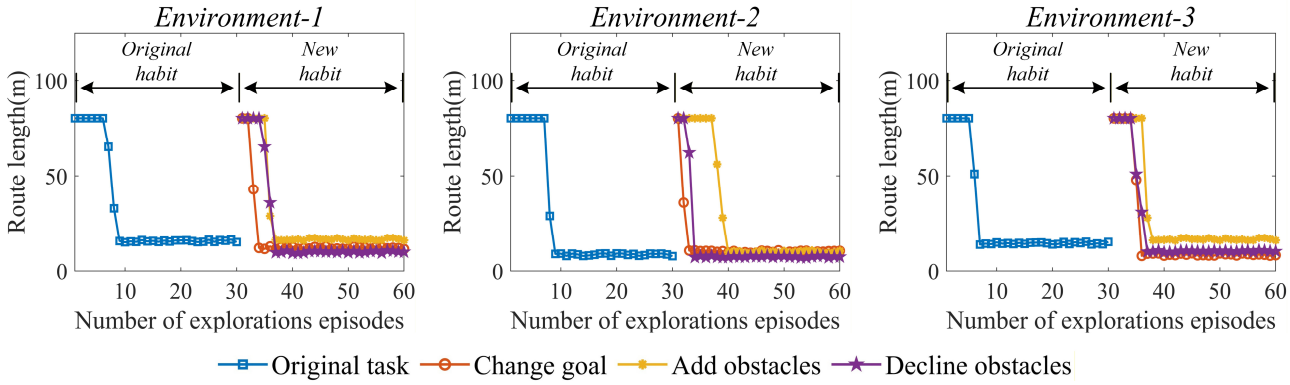
Navigation tasks are typically dynamic, necessitating robust adaptability in robots. To evaluate the proposed method's adaptive capabilities, we design the following experiment: The robot performs 60 exploration episodes, with the first 30 dedicated to habit formation in the original navigation task and the subsequent 30 to habit reshaping in a modified task. Task changes are implemented by adding obstacles, removing obstacles, or altering the goal position. To ensure robustness, each experimental condition is repeated 20 times. Figure 9(a) illustrates the comparison of navigation routes before and after the task change, Figure 9(b) illustrates the average number of exploration episodes needed for reshaping habits, and Figure 9(c) presents the variation of route length with the number of exploration episodes.



(a) Comparison of navigation routes before and after the task change (the pink cuboids represent added obstacles, and the yellow and blue dashed lines denote the original route and the reshaped route).



(b) Comparison of exploration episodes (the number of episodes required decreases significantly after the task is altered).



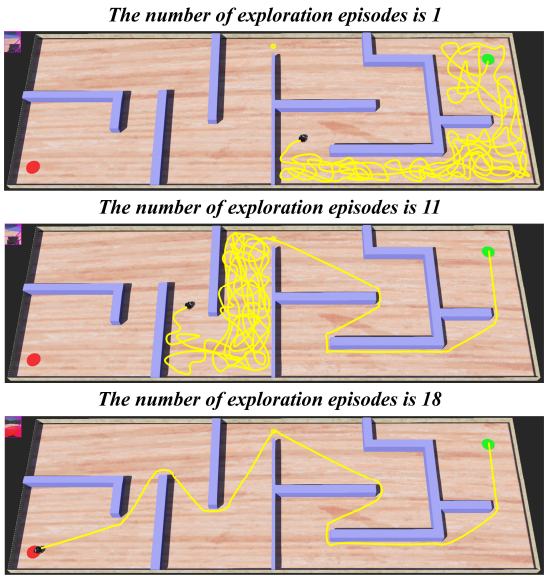
(c) Variation of route length with the number of exploration episodes (the first 30 episodes represent the formation of the initial navigation habit, while the subsequent 30 episodes represent the habit formation process after the navigation task is altered).

Figure 9. Results of the dynamic navigation experiments.

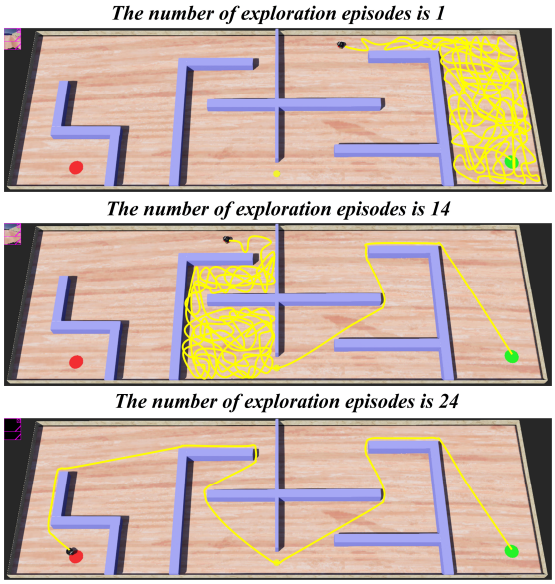
As shown in Figure 9, when the navigation task changes, the proposed method can re-guide the robot to explore the environment and rapidly develop navigational habits adapted to the new task. Compared to the initial task, the number of exploration episodes required for the navigation policy to converge after the task change is significantly reduced. The fundamental reason is that the robot has accumulated rich environmental experiential knowledge during the exploration phase of the original task. Additionally, there exists a certain degree of similarity between the new task and the original task. This similarity enables the robot to effectively reuse prior experience through an experience replay mechanism, thereby rapidly adjusting its navigation policy and achieving more efficient adaptation to the new task.

4.6. Chaining navigation experiments

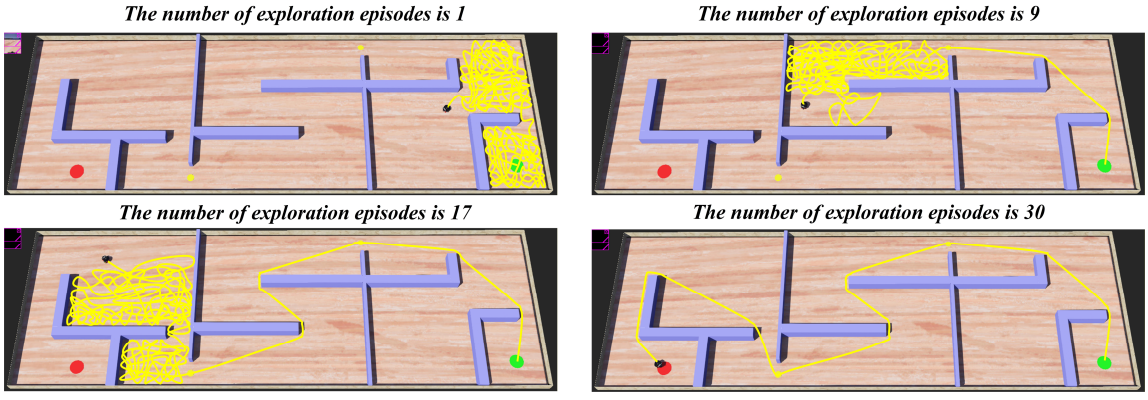
When rats perform navigation tasks in complex environments, they typically decompose the task into multiple subtasks. Inspired by this biological chaining strategy, the framework proposed in this paper introduces a multi-stage navigation method, which decomposes complex long-distance tasks into sequential subtasks in a hierarchical manner. To validate its effectiveness, three relatively complex navigation environments are designed within the Webots platform. Environments 1 and 2 each contain two independent spatial regions with an area of 15m × 10m, while Environment 3 contains three independent spatial regions with an area of 10m × 10m. The independent spatial regions are connected via narrow passages. These environments feature diverse obstacle configurations, including wall partitions, to enforce hierarchical decomposition. To facilitate robot recognition, yellow circular landmarks are placed at the connections between each independent spatial region. Relevant parameters are consistent with those described earlier. Figure 10 shows the changes in the robot's navigation routes as it develops navigation habits in the three environments, while Figure 11 shows the variation of route length with the number of exploration episodes.



(a) Environment-1 navigation training.



(b) Environment-2 navigation training.



(c) Environment-3 navigation training.

Figure 10. Cultivating navigation habits in chaining navigation tasks.

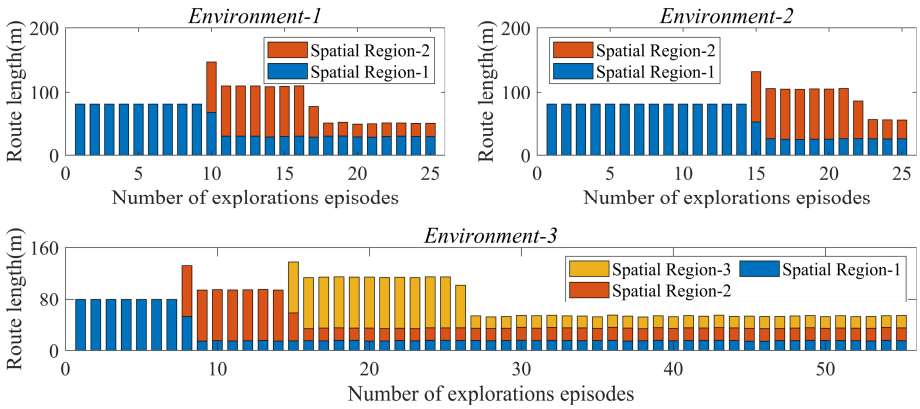


Figure 11. Variation of route length with the number of exploration episodes (the bars of different colors represent the lengths of navigation routes within different sub-regions).

As shown in Figures 10 and 11, during the initial exploration phase, the robot performs random exploration within the current independent spatial compartment to locate passages leading to the next compartment. With increasing exploration episodes, the robot gradually develops local navigation habits from the current start point (or the exit passage of the previous compartment) to the endpoint (or the entrance passage of the next compartment). Furthermore, upon discovering the entrance to the next compartment, the robot immediately commences exploration in that subsequent space. Ultimately, these local habits across compartments are progressively chained, forming an efficient global navigation route that traverses multiple independent spaces from the start to the goal.

To validate the effectiveness of the chaining strategy, it is compared against the original non-chaining method, which directly forms global navigation habits across the environment. Given the increased complexity and larger state space in multi-compartment environments, the non-chaining method often struggles to converge within previous exploration limits. To ensure reliable navigation habit formation in the original method while maintaining fair comparison conditions, the maximum route length per exploration episode is set to 200 m, and the maximum number of exploration episodes is limited to 70. Without loss of generality, each method is repeated 40 times for navigation habit formation in each environment. Figure 12 shows the variation of navigation route length with the number of exploration episodes for methods with and without the chaining strategy. Table 4 presents the statistical results of average route length and average exploration episodes required for habit formation under two methods.

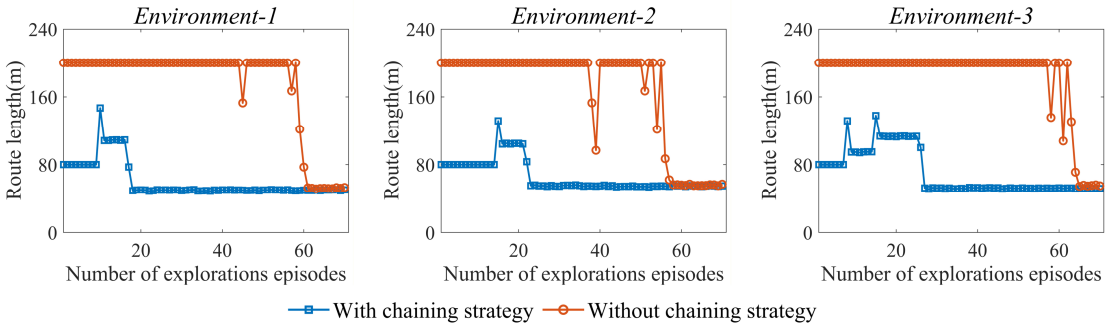


Figure 12. Variation of route length with the number of exploration episodes (the chain-like strategy (blue line) shows a distinct stepwise decrease and converges faster, whereas the non-chain-like strategy (orange line) exhibits a prolonged plateau period).

Table 4. Results of route length and average exploration episodes required for habit formation.

	Spatial regions	Average route length	Average exploration episodes required for habit formation
With the chaining strategy	Environment-1	48.91m	17.7
	Environment-2	55.13m	22.4
	Environment-3	51.41m	30.6
Without the chaining strategy	Environment-1	51.82m	46.8
	Environment-2	59.74m	53.1
	Environment-3	55.51m	64.2

As shown in Figure 12, the chaining strategy exhibits a characteristic stepwise decrease in route length, reflecting the robot's progressive mastery and integration of local navigation habits within individual compartments, thereby achieving faster and more stable convergence. In contrast, the non-chaining method displays prolonged plateau periods, indicating difficulties in locating the navigation goal and forming corresponding habits. Then, as shown in Table 4, the chaining strategy significantly outperforms the non-chaining method, reducing the exploration episodes required for habit formation by 52.3%–62.2% and shortening the route length by 5.6%–7.7%. Moreover, the performance gap widens with increasing environmental complexity and number of independent compartments. These results confirm that the chaining mechanism effectively mitigates navigation challenges in large-scale unstructured environments, enabling mobile robots to achieve more efficient, robust, and scalable formation of navigation habits in complex settings.

4.7. Runtime statistics and computational complexity analysis

To address the practical deployability of the proposed method, we provide a theoretical analysis of its computational complexity. The time complexity analysis of the navigation framework, when evaluated per robot position (i.e., at each decision-making or state-update step during navigation, excluding cumulative exploration steps), is dominated by the real-time computation in the Sn-Plast spiking neural network. The primary bottleneck arises from updating the membrane potentials of all prefrontal action cells by summing contributions from all hippocampal CA3 place cells, along with lateral interactions among action cells, resulting in a complexity of $O(N_{CA3\_PC} \times N_{AC} + N_{AC} \times N_{AC})$, with N_{CA3_PC} and N_{AC} representing the numbers of CA3 place cells and action cells. When $N_{CA3\_PC} \gg N_{AC}$, it is approximately equal to $O(N_{CA3\_PC} \times N_{AC})$. Path integration in the entorhinal-hippocampal model, which updates CA3 place cell activations based on self-motion cues, contributes the complexity of $O(N_{CA3\_PC} \times N_{GC})$, with N_{GC} representing the numbers of grid cells. Notably, route optimization, route feedback, and subtask transitions are triggered only upon goal arrival and thus incur zero additional cost at regular position steps. Furthermore, it is worth noting that route optimization, route feedback, and navigation task decomposition are triggered only upon reaching the goal, and therefore do not incur any additional computational cost during regular position steps. Overall, the dominant computational complexity is $O(N_{CA3\_PC} \times N_{AC} + N_{CA3\_PC} \times N_{GC})$.

To evaluate the time consumption of the proposed navigation framework in real-world execution, we statistically analyze the average computational time at each sampled position during task execution. The measured average processing time per position point is 84.51 ms, which corresponds to a real-time operating frequency of approximately 12 Hz.

5. Conclusions

In this study, we present a brain-inspired navigation framework for mobile robots that integrates key mechanisms from the rodent entorhinal-hippocampal-prefrontal system. By combining path integration, Sn-Plast-based habit formation, route optimization, feedback-driven memory consolidation, and hierarchical chaining navigation strategy, the proposed method achieves efficient goal-directed navigation while addressing limitations of traditional SLAM and reinforcement learning methods, such as route inefficiency, poor adaptability, and scalability issues in large or dynamic environments. Simulation experiments in diverse arenas demonstrate superior performance in route length reduction, convergence speed, and robustness compared to baseline algorithms. Ablation studies confirm the critical contributions of route optimization and feedback consolidation. Dynamic navigation experiments highlight rapid habit reshaping, while chaining navigation experiments show substantial reductions in required exploration episodes and final route length in complex multi-compartment settings.

Overall, the framework validates the effectiveness of biologically plausible mechanisms in enabling adaptive, efficient, and scalable robotic navigation. These findings not only advance bio-inspired robotics but also provide new insights into the computational principles underlying mammalian spatial cognition. Future work includes extending the method to real-world robotic platforms for practical validation, integrating multi-modal sensory inputs such as vision, tactile, and auditory feedback to improve perception in noisy or partially observable environments, exploring additional neuromodulatory effects to achieve more flexible and diverse learning behaviors, and combining the framework with lifelong learning mechanisms to enable continuous adaptation and knowledge accumulation across tasks and environments.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62566024 and Grant 62076014, in part by the Science and Technology Research Project of Jiangxi Provincial Department of Education under Grant GJJ2400402, and in part by the Early-Career Young Scientists and Technologists Project of Jiangxi Province under Grant 20252BEJ730116.

Conflict of interest

The authors declare there are no conflict of interest.

References

1. A. Ibrayev, B. Omarov, Recent advances in visual SLAM: taxonomy, comparative analysis, and open challenges, *Eng. Technol. Appl. Sci. Res.*, **15** (2025), 29069–29076. https://doi.org/10.48084/etasr.13116

2. C. A. Freas, M. L. Spetch, Varieties of visual navigation in insects, *Anim. Cognit.*, **26** (2023), 319–342. https://doi.org/10.1007/s10071-022-01720-7
3. P. Yin, A. Abuduweili, S. Zhao, L. Xu, C. Liu, S. Scherer, Bioslam: A bioinspired lifelong memory system for general place recognition, *IEEE Trans. Rob.*, **39** (2023), 4855–4874. https://doi.org/10.1109/TRO.2023.3306615
4. W. Cothi, N. Nyberg, E. M. Griesbauer, C. Ghanamé, F. Zisch, J. M. Lefort, et al., Predictive maps in rats and humans for spatial navigation, *Curr. Biol.*, **32** (2022), 3676–3689. https://doi.org/10.1016/j.cub.2022.06.090
5. L. M. Giocomo, M. B. Moser, E. I. Moser, Computational models of grid cells, *Neuron*, **71** (2011), 589–603. https://doi.org/10.1016/j.neuron.2011.07.023
6. R. M. Grieves, É. Duvelle, P. A. Dudchenko, A boundary vector cell model of place field repetition, *Spatial Cognit. Comput.*, **18** (2018), 217–256. https://doi.org/10.1080/13875868.2018.1437621
7. A. A. Fenton, Remapping revisited: how the hippocampus represents different spaces, *Nat. Rev. Neurosci.*, **25** (2024), 428–448. https://doi.org/10.1038/s41583-024-00817-x
8. A. Arleo, W. Gerstner, Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity, *Biol. Cybern.*, **83** (2000), 287–299. https://doi.org/10.1007/s004220000171
9. F. Gobbo, R. Mitchell-Heggs, D. Tse, M. Al Omrani, P. A. Spooner, S. R. Schultz, et al., Neuronal signature of spatial decision-making during navigation by freely moving rats by using calcium imaging, *Proc. Natl. Acad. Sci. U.S.A.*, **119** (2022), e2212152119. https://doi.org/10.1073/pnas.2212152119
10. F. K. Qian, Y. Li, J. C. Magee, Mechanisms of experience-dependent place-cell referencing in hippocampal area CA1, *Nat. Neurosci.*, **28** (2025), 1486–1496. https://doi.org/10.1038/s41593-025-01930-5
11. W. Tang, J. D. Shin, S. P. Jadhav, Geometric transformation of cognitive maps for generalization across hippocampal-prefrontal circuits, *Cell Rep.*, **42** (2023), 112246. https://doi.org/10.1016/j.celrep.2023.112246
12. J. Widloski, D. J. Foster, Flexible rerouting of hippocampal replay sequences around changing barriers in the absence of global place field remapping, *Neuron*, **110** (2022), 1547–1558. https://doi.org/10.1016/j.neuron.2022.02.002
13. Y. Zheng, X. Zhou, S. C. Moseley, S. M. Ragsdale, L. J. Alday, W. Wu, et al., A Hippocampal–Parietal Network for Reference Frame Coordination, *J. Neurosci.*, **45** (2025), e1782242025. https://doi.org/10.1523/JNEUROSCI.1782-24.2025
14. H. Mailka, M. Abouzahir, M. Ramzi, An efficient end-to-end EKF-SLAM architecture based on LiDAR, GNSS, and IMU data sensor fusion for autonomous ground vehicles, *Multimedia Tools Appl.*, **83** (2024), 56183–56206. https://doi.org/10.1007/s11042-023-17595-w
15. C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, J. D. Tardós, Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap SLAM, *IEEE Trans. Rob.*, **37** (2021), 1874–1890. https://doi.org/10.1109/TRO.2021.3075644
16. M. J. Milford, G. F. Wyeth, Mapping a suburb with a single camera using a biologically inspired SLAM system, *IEEE Trans. Rob.*, **24** (2008), 1038–1053. https://doi.org/10.1109/TRO.2008.2004520

17. Q. Zou, M. Cong, D. Liu, Y. Du, A neurobiologically inspired mapping and navigating framework for mobile robots, *Neurocomputing*, **460** (2021), 181–194. https://doi.org/10.1016/j.neucom.2021.07.025
18. J. Li, T. Zeng, B. Si, Brain-inspired visual topometric localization via roadnetwork-constraint hidden markov model, *IEEE Rob. Autom. Lett.*, **10** (2025), 11062–11069. https://10.1109/LRA.2025.3604706
19. T. Lu, Z. Wang, X. Fan, Y. Shen, X. Shao, Hybrid-NeuroSLAM: A neurobiologically inspired hybrid visual-inertial SLAM method for large scale environment, *IEEE Rob. Autom. Lett.*, **10** (2025), 7484–7491. https://doi.org/10.1109/LRA.2025.3574995
20. R. Bai, H. Guo, W. Y. Yau, L. Xie, Graph-based slam-aware exploration with prior topo-metric information, *IEEE Rob. Autom. Lett.*, **9** (2024), 7597–7604. https://doi.org/10.1109/LRA.2024.3420817
21. P. Y. Oudeyer, F. Kaplan, Intelligent adaptive curiosity: A source of self-development, in *Proceedings of the International Workshop on Epigenetic Robotics*, (2004), 127–130.
22. X. Zhang, X. Ruan, Y. Xiao, Q. Sie, J. Chai, Mobile robot autonomous path planning method based on intrinsic motivation mechanism, *Control Decis.*, **33** (2018), 1605–1611. https://10.13195/j.kzyjc.2017.0620
23. M. Pei, H. An, B. Liu, C. Wang, An improved dyna-q algorithm for mobile robot path planning in unknown dynamic environment, *IEEE Trans. Syst. Man Cybern.: Syst.*, **52** (2021), 4415–4425. https://10.1109/TSMC.2021.3096935
24. A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, et al., Vector-based navigation using grid-like representations in artificial agents, *Nature*, **557** (2018), 429–433. https://doi.org/10.1038/s41586-018-0102-6
25. H. Li, Q. Zhang, D. Zhao, Deep reinforcement learning-based automatic exploration for navigation in unknown environment, *IEEE Trans. Neural Networks Learn. Syst.*, **31** (2019), 2064–2076. https://doi.org/10.1109/TNNLS.2019.2927869
26. L. Zhang, J. Peng, W. Yi, H. Lin, L. Lei, X. Song, A state-decomposition DDPG algorithm for UAV autonomous navigation in 3-D complex environments, *IEEE Internet Things J.*, **11** (2023), 10778–10790. https://doi.org/10.1109/JIOT.2023.3327753
27. T. Kulvicius, M. Tamosiunaite, J. Ainge, P. Dudchenko, F. Wörgötter, Odor supported place cell model and goal navigation in rodents, *J. Comput. Neurosci.*, **25** (2008), 481–500. https://doi.org/10.1007/s10827-008-0090-x
28. N. Frémaux, H. Sprekeler, W. Gerstner, Reinforcement learning using a continuous time actor-critic framework with spiking neurons, *PLoS Comput. Biol.*, **9** (2013), e1003024. https://doi.org/10.1371/journal.pcbi.1003024
29. Z. Brzosko, S. Zannone, W. Schultz, C. Clopath, O. Paulsen, Sequential neuromodulation of Hebbian plasticity offers mechanism for effective reward-based navigation, *eLife*, **6** (2017), e27756. https://doi.org/10.7554/eLife.27756
30. G. W. Y. Ang, C. S. Tang, Y. A. Hay, S. Zannone, O. Paulsen, C. Clopath, The functional role of sequentially neuromodulated synaptic plasticity in behavioural learning, *PLoS Comput. Biol.*, **17** (2021), e1009017. https://doi.org/10.1371/journal.pcbi.1009017
31. H. Wang, Behavioral decision-making of mobile robots simulating the memory consolidation mechanism of human brain, *Adapt. Behav.*, **33** (2025), 1–15. https://doi.org/10.1177/10597123241302508

32. X. Lv, D. Wang, A bionic memory replay model based on hippocampal-prefrontal collaboration: design, mechanism, and robot navigation verification, *IEEE Trans. Autom. Sci. Eng.*, **23** (2026), 2852–2863. https://doi.org/10.1109/TASE.2026.3657422
33. E. Huang, D. Xu, H. Zhu, Z. Chen, Y. Chen, S. Lei, et al., Hierarchical replay of multi-regional sequential spiking associated with working memory, *Neurosci. Bull.*, (2025), 1–14. https://doi.org/10.1007/s12264-025-01524-y
34. Y. Liao, N. Yu, A real-time cognitive map construction method based on the entorhinal-hippocampal working mechanism of the rat's brain, *Cogn. Comput. Syst.*, **6** (2024), 49–64. https://doi.org/10.1049/ccs2.12101
35. J. Xue, J. Chen, S. Zhang, Action-curiosity-based deep reinforcement learning algorithm for path planning in a nondeterministic environment, *Intell. Comput.*, **4** (2025), 0140. https://doi.org/10.34133/icomputing.0140

AIMS Press