CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Brain-Mimetic Mapless Navigation Framework Integrating Visual Streams and Entorhinal-Hippocampal-Prefrontal Circuits

Yishen Liao[1] | Naigong Yu[2,3] | Hejie Yu[4] | Shufei Fu[1]

[1]Modern Industry School of Virtual Reality, Jiangxi University of Finance and Economics, Nanchang, China | [2]School of Information Science and Technology, Beijing University of Technology, Beijing, China | [3]Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing, China | [4]Department of Precision Instrument, Tsinghua University, Beijing, China

Correspondence: Naigong Yu (yunaigong@bjut.edu.cn)

Received: 1 March 2026 | **Revised:** 5 June 2026 | **Accepted:** 1 July 2026

Keywords: entorhinal-hippocampal-prefrontal | mapless navigation | mobile robots | spatial cells | visual streams

ABSTRACT

Efficient autonomous navigation in complex and unstructured environments without pre-existing maps remains a significant challenge for mobile robotics. Drawing inspiration from rodent neural architectures, this study proposes a brain-mimetic mapless navigation framework for mobile robots. It integrates three key components to achieve autonomous navigation: first, a visual stream-based localisation model utilising object-vector cells to compensate for cumulative path integration errors; second, a navigation-guiding model where the hippocampal-prefrontal circuitry outputs initial paths through exploration and learning, which are then dynamically optimised for obstacle avoidance through a self-organising strategy of hippocampal CA1 place cells incorporating boundary-vector cells' inputs; and finally, the consolidation of these optimised paths into stable navigation habits via spatial cells' theoretical firing rate and spike-timing-dependent plasticity learning rule. Extensive 2D and 3D simulation experiments demonstrate that the proposed framework not only outperforms various baseline algorithms in terms of convergence speed and path efficiency but also exhibits robust error correction under motion noise, rapid habit reshaping during task changes and strong invariance to the initial heading direction. Moreover, real-world experiments on a mobile robot in complex indoor environments further confirm its practical feasibility and effectiveness, which offers a biologically plausible and computationally efficient navigation solution.

1 | Introduction

The ability to navigate complex, unstructured environments without relying on preconstructed maps is a hallmark of biological intelligence and a long-standing goal in autonomous robotics [1]. Map-based approaches, inspired by hippocampal place and grid cells, support flexible planning and shortcuts, but they require constructing and maintaining an explicit metric map, which is costly and brittle in low-feature or dynamic settings [2]. However, in rodents, mapless navigation emerges not from explicit spatial mapping but from the gradual consolidation of task-specific behavioural habits shaped by repeated interaction with the environment [3]. Reproducing this capability requires solving three tightly coupled subproblems at the same time. The first is maintaining an accurate estimate of self-position despite the drift inherent to path integration. The second is converging from initially blind, random exploration onto a globally efficient and obstacle-free route. The third is retaining the resulting route as a stable, retrievable policy that can be

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

rapidly reshaped when the goal or obstacles change. It is the coupling among these three, in which localisation feeds planning, planning feeds memory and memory closes the loop back onto action, that makes the problem difficult, and it is precisely this coupling that the rodent brain resolves through a distributed but integrated circuit spanning the visual cortex, entorhinal-hippocampal system and prefrontal cortex [4].

The first subproblem, drift correction, is solved in the rodent brain by anchoring self-motion estimates to external landmarks. The dorsal 'where' stream computes spatial relationships regarding landmark distance and bearing, whereas the ventral 'what' stream identifies landmark attributes [5]. Downstream, object-vector cells (OVCs) in the entorhinal cortex fuse these streams into landmark-centred vectors of direction and distance relative to the rat's current pose, forming a geometric reference frame for real-time localisation [6]. Because this frame is re-anchored to stable external features rather than to integrated motion, it periodically resets accumulated drift, allowing rats to preserve positional accuracy during extended exploration even under low-feature or noisy conditions [7].

The second subproblem, route efficiency, is addressed by hippocampal CA1 dynamics. Place cells located in the hippocampal CA1 not only represent the current location but also support trajectory optimisation through self-organised remapping of their firing fields [8]. During offline replay, optimised path sequences are reactivated and consolidated, progressively refining navigation efficiency and escaping the local optima typical of greedy exploration [9], whereas boundary-vector cells (BVCs) supply obstacle geometry for collision-free adjustment [10]. This remapping is only meaningful when it is grounded on an accurate position estimate, which links it directly to the localisation stage above.

The third subproblem, stable and reusable habits, relies on experience-dependent plasticity between the prefrontal cortex and hippocampus. Spike-timing-dependent plasticity (STDP) at CA3-prefrontal synapses strengthens action–outcome associations during successful goal-reaching episodes [11], and over repeated trials, these connections converge into stable motor policies [12]. By replaying the optimised CA1 trajectories as the teaching signal for this plasticity, the brain consolidates not raw exploration but the refined route and can quickly reshape the habit when conditions change.

Taken together, these mechanisms are not independent; they form a closed perception-planning-memory loop. The corrected pose from the OVC-based localisation stage is the substrate on which CA1 remapping operates. The refined trajectory then becomes the supervisory signal that retrains the hippocampal CA3-prefrontal network. The consolidated habit, in turn, generates the motor policy that drives the next cycle of sensing and correction. Despite the elegance of this integrated loop in biology, these functions remain studied in isolation and have been imported into robotics only piecemeal, with no unified, bio-plausible system that reproduces the full loop [13]. This paper introduces a unified, bio-plausible mapless navigation framework that bridges this gap. Operating without external rewards, prior maps or GPS, it emulates the full rodent navigation loop in an efficient spiking neural architecture. Our original contributions are as follows:

- A real-time localisation module that models the 'what' and 'where' visual streams and OVC receptive fields to extract prior landmark cues from monocular images, enabling geometric correction of path integration drift with minimal computational overhead.
- A dynamic path optimisation module based on CA1 place-cell self-organisation, which iteratively shortens trajectories, avoids local optima and handles obstacles via BVC input, enabling globally efficient routes from initially random explorations.
- A memory consolidation loop that replays optimised CA1 trajectories as theoretical firing sequences to supervisely retrain the hippocampal CA3-prefrontal spiking neural network, forming stable, retrievable habits of refined paths and enabling rapid habit reshaping when goals or obstacles change.

The subsequent structure of this article is outlined below. Section 2 revisits pertinent studies and discusses previous research endeavours. Section 3 delves into the various details of model construction. Experimental findings are presented in Section 4, followed by the conclusion in Section 5.

2 | Related Works

2.1 | Landmark-Based Localisation and Error Correction Methods

Landmark-based localisation in mobile robotics has advanced from early dependence on artificial passive landmarks to using natural environmental features such as corners, edges and semantic object [14, 15]. Recent developments incorporate semantic landmarks and multisensor fusion, achieving centimetre-level accuracy and reliable navigation in complex, dynamic environments [16, 17]. Beyond 2D cues, geometry-aware 3D point cloud learning has further enabled precise target localisation in unstructured field environments, maintaining centimetre-level accuracy even under heavy occlusion [18]. In parallel, probabilistic state estimators such as the extended Kalman filter have been widely adopted for motion estimation and trajectory prediction under uncertainty, remaining effective even in dynamic, unstructured environments [19]. However, inspired by neuroscientific discoveries of visual streams and OVCs in the rat brain, biomimetic approaches have recently enhanced landmark-based techniques. In 2021, Andersson et al. employed a Bayesian model of OVCs' firing fields, computing Gaussian-uniform noise probability distributions in response to visual stimuli such as object size and contrast, enabling the identification of primitive visual features as robust vector anchors for weak-field detection [20]. In 2023, Han and Sereno extended artificial neural networks to localise multiple objects via ventral and dorsal pathways, computing shape-space representations for enhanced recognition and positioning accuracy [21]. In 2025, Piray et al. modelled OVCs as polar encoders within a compositional map, integrating self-supervised grid-cell signals to enable flexible, efficient path planning [22].

2.2 | Navigation Method Based on Brain-Inspired SLAM

Brain-inspired SLAM research originated from Milford's RatSLAM framework [23]. By mimicking the spatial representation mechanisms of the rat hippocampus, this framework couples a pose cell network with an experience map, demonstrating robust adaptability and stable mapping performance across complex indoor and outdoor scenes. On this basis, Milford incorporated a 'goal memory' navigation framework that learns temporal gradients among locations during exploration. Consequently, the robot can achieve path planning by relying on the association between the current state and the target's temporal gradient [24]. Subsequent studies have propelled brain-inspired SLAM beyond pure metric mapping towards the fusion of semantic information and cognitive representations. By explicitly introducing semantic information into SLAM, the biologically plausible spiking neural network model, SSP-SLAM [25], leverages the neural engineering framework and semantic pointer architecture to encode continuous spatial maps via vectors, seamlessly binding cross-domain visual and conceptual features. On a theoretical level, Safron et al. developed a generative SLAM architecture [26] by implementing the free energy principle and active inference framework atop Latent-SLAM. Focusing on the representation learning of cognitive maps, George et al. introduced the clone-structured cognitive graph (CSCG) [27]. This architecture spawns multiple 'clones' for an identical observation under distinct contexts, enabling highly efficient learning via probabilistic sequence models.

However, the constraint of map-based navigation methods lies in their reliance on pre-existing environmental maps. Because conventional path planning algorithms demand destinations situated strictly within traversable areas, these methods struggle to guide robots towards targets beyond explored boundaries, thereby compromising flexibility in unmapped environments. Conversely, if a model can directly establish an end-to-end mapping between perception and navigation behaviours, these aforementioned challenges could be effectively resolved.

2.3 | Reinforcement Learning-Based Mapless Navigation Methods

Advances in reinforcement learning (RL) have deepened studies on navigation habit formation, modelling path learning via exploration and reward signals. In 2004, Oudeyer and Kaplan introduced intelligent adaptive curiosity (IAC) for autonomous exploration in unknown environments [28]. In 2009, Baranes and Oudeyer proposed robust R-IAC to reduce prediction error in complex sensorimotor spaces [29], which Zhang et al. later applied to unstructured path planning with action selection, enabling obstacle avoidance and goal adaptation across initial states [30]. In 2014, Dasgupta et al. developed an RNN-based 'action-evaluation' framework, where an evaluation module computes reward-gradient error signals and an action module executes perception-driven behaviours [31].

Traditional RL suffers from the curse of dimensionality in high-dimensional state spaces. Deep reinforcement learning (DRL), integrating deep learning with RL, addresses this issue and is widely applied in robot navigation. In 2018, DeepMind used an LSTM-based network to achieve grid-cell-like vector navigation in virtual mazes via DRL, enabling autonomous localisation, multiscale representation and goal-directed navigation [32]. In 2022, Ruan et al. combined LSTM with DQN for intrinsic motivation-driven visual navigation, learning environmental structure and selecting waypoints via curiosity and temporal distance, demonstrating efficient exploration and adaptability in 3D environments [33]. In 2023, Li et al. enhanced DQN with prioritised replay, artificial potential fields and transfer learning, boosting the navigation success rate and speed in unknown obstacle-rich settings [34]. In 2024, Gao et al. advanced DRL with a dual extrinsic-intrinsic reward (using memory-decaying novelty) and LSTM, improving mapless navigation in long corridors and dead corners by up to 28% [35].

2.4 | Bioinspired Neural Circuit-Based Mapless Navigation Methods

DRL's large-scale grids and long training hinder mobile robot deployment, prompting bioinspired mapless navigation mimicking hippocampal-prefrontal circuits, especially the interactions between prefrontal cortex action cells and hippocampal place cells. As early as 2008, Kulvicius et al. used feedforward networks to connect place cells and action cells, adjusting weights via Q-learning during exploration to guide navigation [36]. In 2013, Frémaux et al. employed spiking networks with continuous-time differential learning and STDP for reward prediction, accelerating Morris water maze goal-finding [37]. In 2017, Brzosko et al. developed Sn-Plast, integrating acetylcholine and dopamine into reward-modulated STDP to enable efficient navigation under changing goals [38]. This breakthrough applied bidirectionally neuromodulated synaptic rules in computational models. In 2018 and 2021, Zannone et al. modelled acetylcholine's effects on Sn-Plast, showing enhanced behavioural exploration and greater STDP flexibility, outperforming other reward-based plasticity rules [39, 40]. In 2025, Wei et al. proposed a bio-plausible hippocampal-basal ganglia model with STDP and dopamine-modulated RL. It integrates allocentric hippocampal mapping with egocentric cortico-basal ganglia action selection, reproducing rodent maze behaviours and lesion effects, and enabling mobile robot navigation [41].

3 | Methodology

This section provides a detailed description of the mapless navigation framework. The overall architecture is illustrated in Figure 1. The proposed framework is composed of three primary modules: information perception and processing, localisation error correction and navigation habit formation. The primary functions of each component are described below.

Information perception and processing: During the process of exploring the environment, the robot collects self-motion clues through encoders and gyroscopes and collects landmark information through cameras. Subsequently, the self-motion cues are input into the entorhinal-hippocampal CA3 neural computing model [42], which achieves more accurate path integration through grid cells and CA3 place cells.

Localisation error correction: By constructing computational models of the 'what' and 'where' visual streams along with a

receptive field model for OVCs, a prior landmark from the environment is acquired to reduce drift accumulated over the course of path integration, thereby enhancing positional accuracy.

Navigation habit formation: First, the Sn-Plast learning rule modulates synaptic weights in the spiking neural network (SNN) from hippocampal CA3 place cells to prefrontal action cells [39], including its modelling of synaptic dynamics and delay variations, simulating navigational habit formation in rats. Next, a self-organisation model of CA1 place cells optimises the navigation trajectory, with BVCs supplying obstacle information. Finally, optimised path information is fed back to the SNN to establish stable trajectory memory.

3.1 | Visual Stream Calculation Model

This section sequentially models the two visual streams and the firing properties of OVCs. In this study, visual landmarks are defined as cards placed on the ground, each imprinted with distinct characters to represent the uniqueness of individual landmarks. A camera for landmark recognition is mounted on the front of the robotic platform, with its imaging plane parallel to the ground. A schematic of the robot capturing visual landmarks in the environment is shown in Figure 2. To ensure reliable conversion from pixel measurements to azimuth and distance, the camera was calibrated using Zhang's calibration method [43]. Then, each image frame is rectified for distortion before entering the visual stream, thereby ensuring the accuracy of azimuth and distance estimations.

The variables in Figure 2 are defined as follows: The position of the landmark in the world coordinate system is $\left(x_{\text{obj}}, y_{\text{obj}}\right)$, the angle between the landmark's character and the positive x-axis of the world coordinate system is θ_{wor}, the height of the camera above the ground is h_{cam}, and the distance from the landmark to the camera's projection on the ground is d_{obj}. In the camera-captured image coordinate system, the pixel distance from the landmark to the image centre is d_{cam}, the angle between the landmark's character and the positive x-axis of the camera-captured image coordinate system is θ_{obj}, and the orientation of the landmark relative to the robot's heading is θ_{dir}. Here, the landmark position $\left(x_{\text{obj}}, y_{\text{obj}}\right)$, inclination angle θ_{wor} and camera height h_{cam} are known quantities. The computational model for each component is derived as follows:

- 'What' visual stream computational model
  The 'what' visual stream is responsible for accurately recognising characters within landmarks, and its computational model operates as follows: First, GIST descriptors are employed to extract and represent character images in a

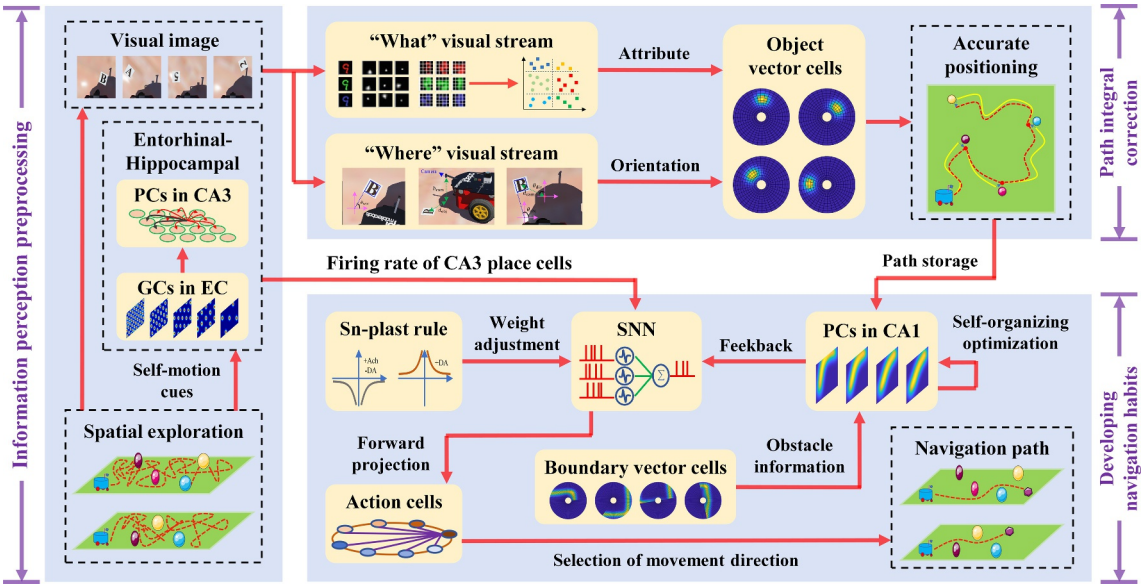


FIGURE 1 | Comprehensive structure of the proposed framework.

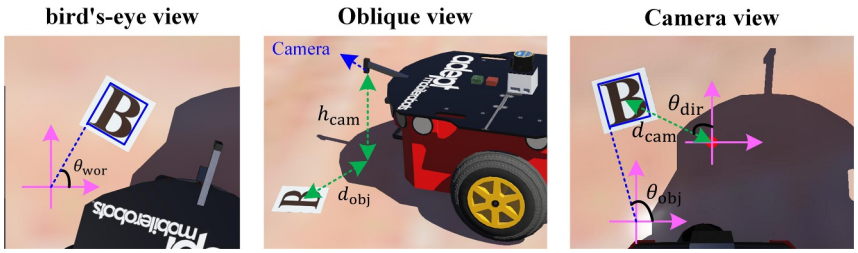


FIGURE 2 | Schematic diagram of the robot collecting visual landmarks in the environment.

high-dimensional space. Subsequently, to balance detection efficiency and accuracy, a linear-based multiclass random oblique decision tree algorithm [44] is utilised for character recognition in the images.

- 'Where' visual stream computational model
  The 'where' visual stream infers the coordinates of landmarks relative to the robot's position from both egocentric and distance information, achieving specific steps as follows: First, the 'what' visual stream identifies landmarks in the image and determines their positions. Subsequently, a binary segmentation is performed on the character regions in the image, and the smallest outer rectangular bounding box algorithm is employed to obtain the outer rectangular frame of the characters. The pixel distance d_{cam} from the image element to the camera can then be calculated based on this outer rectangular frame. Thereafter, the ratio coefficient α_{cam} is defined to represent the conversion from the true distance to the pixel distance, enabling the derivation of the projection of the object on the ground and its distance d_{obj} from the landmark, as shown in Equation (1).

$$d_{\text{obj}} = \alpha_{\text{cam}} \cdot (h_{\text{cam}} + h_{\text{con}}) \cdot d_{\text{cam}}. \tag{1}$$

In Equation (1), the constant distance h_{con} and the conversion coefficient α_{cam} are determined based on the inherent calibration of the camera. Finally, the relative directional angle θ_{dir} of the character with respect to the camera is computed, as shown in Equation (2).

$$\theta_{\text{dir}} = \arctan(y_{\text{cam}}/x_{\text{cam}}) - \pi/2. \tag{2}$$

In Equation (2), x_{cam} and y_{cam} represent the horizontal and vertical coordinates of the landmark in the camera-acquired image coordinate system, respectively.

- Firing properties of OVCs
  As downstream neurons of the two visual streams, OVCs integrate outputs from both streams to encode the attributes of environmental landmarks and their direction and distance relative to the rat. The firing activity of OVCs is organised in the form of receptive fields, which define the range of external visual stimuli to which the neuron responds. Each receptive field consists of multiple receptive units, with each unit specialised in encoding a specific location within the rat's visual field. These receptive units operate collaboratively to form a receptive field network, enabling comprehensive capture of landmark information in the surrounding environment. The network structure of the receptive fields in OVCs is illustrated in Figure 3.

In Figure 3, each rectangular frame represents one receptive unit. It can be observed that the receptive field of OVCs is constructed by building a multilayered ring network structure, where each layer consists of a group of independent ring-shaped receptive unit clusters. The specific position encoded by the jth receptive unit in the ith layer is defined as Su_pos_{ij}, and the mathematical expressions for its horizontal coordinate $\text{Su_pos}_{ij} \cdot x$ and vertical coordinate $\text{Su_pos}_{ij} \cdot y$ are given in Equations (3) and (4).

$$\text{Su_pos}_{ij} \cdot x = (d_{\text{su}} i/\text{Su_layer}) \times \cos(2\pi j/\text{Su}_{\text{ring}}), \tag{3}$$

$$\text{Su_pos}_{ij} \cdot y = (d_{\text{su}} i/\text{Su_layer}) \times \sin(2\pi j/\text{Su_ring}). \tag{4}$$

In Equations (3) and (4), d_{su} represents the farthest distance perceivable by perception, which is dictated by the radius of the camera's field of view. Su_layer denotes the number of layers in the receptive network, and Su_ring indicates the number of units in each layer. Regarding these two parameters, larger values yield more precise encoding at the expense of increased computational overhead; conversely, smaller values result in coarser encoding but reduce computational overhead. Then, $d_{\text{su}} i/\text{Su_layer}$ and $2\pi j/\text{Su_ring}$, respectively, represent the encoding distances and encoding directions of the receptive units. Subsequently, the landmark distance d_{obj} and directional angle θ_{dir} are used to compute the overall receptive strength of all receptive units. The receptive strength of the jth receptive unit in the ith layer is defined as Su_streng_{ij}, as shown in Equation (5).

$$\text{Su_streng}_{ij} = e^{-\left\| \text{obj_pos} - \text{Su_pos}_{ij} \right\|^2/\sigma_{\text{Su}}}. \tag{5}$$

In Equation (5), σ_{Su} represents the tuning factor of the receptive unit, and $\left\| \text{obj_pos} - \text{Su_pos}_{ij} \right\|$ denotes the distance between the encoded position of the receptive unit and the position of the landmark relative to the camera. The mathematical expression for the position coordinate obj_pos is given in Equation (6).

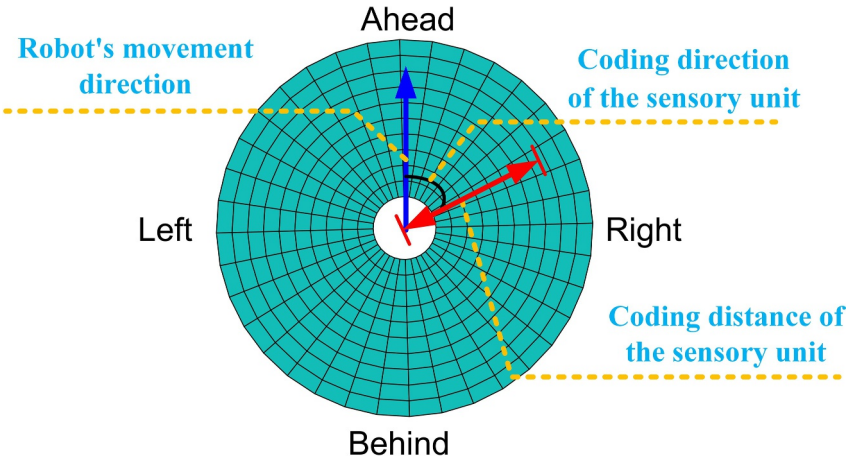


FIGURE 3 | Schematic diagram of the object-vector cells' receptive field.

$$\text{obj_pos} = \left[d_{\text{obj}} \cos(\theta_{\text{dir}}), d_{\text{obj}} \sin(\theta_{\text{dir}})\right]. \quad (6)$$

3.2 | Localisation Error Correction Method

When OVCs detect landmarks within the robot's visual field, they utilise the information from these landmarks to precisely compute the robot's position in the environment. This not only helps correct the cumulative errors arising from path integration throughout the exploration process, thereby establishing a more accurate position estimate, but also further enhances the robustness of the robot during the formation of navigation habits. The geometric relationships of the cumulative error correction process are illustrated in Figure 4a, whereas the operational mechanism of the cumulative error correction process is shown in Figure 4b.

In Figure 4a, the robot first perceives Landmarks A and B. Here, t^A_{loss} and t^B_{loss} denote the moments when the robot loses sight of Landmarks A and B during navigation, respectively, whereas t^B_{enter} represents the moment when the robot discovers Landmark B during navigation. When $t \in \left[t^B_{\text{enter}}, t^B_{\text{loss}}\right]$, the robot can use the path information to accurately locate itself. The calculation steps are as follows: First, the inclination angle θ_{obj} of the character's outer rectangle in the world coordinate system is computed based on the angles θ_{obj} and θ_{wor}, thereby obtaining the robot navigation angle θ^{vis} inferred from the landmark, as shown in Equation (7).

$$\theta^{\text{vis}} = \pi/2 - \theta_{\text{obj}} + \theta_{\text{wor}}. \quad (7)$$

The position $r^{\text{vis}}(t) = \left(x^{\text{vis}}, y^{\text{vis}}\right)$ represents the robot's position at time t, as calculated based on the landmark. d_{cr} denotes the distance between the camera and the centroid of the robot on the 2D plane. Then, x^{vis} and y^{vis} can be computed through spatial geometric relations, as shown in Equations (8) and (9).

$$x^{\text{vis}} = x_{\text{obj}} - d_{\text{obj}} \cos\left(\theta_{\text{dir}} + \theta^{\text{vis}}\right) - d_{\text{cr}} \cos\left(\theta^{\text{vis}}\right), \quad (8)$$

$$y^{\text{vis}} = y_{\text{obj}} - d_{\text{obj}} \sin\left(\theta_{\text{dir}} + \theta^{\text{vis}}\right) - d_{\text{cr}} \sin\left(\theta^{\text{vis}}\right). \quad (9)$$

To mitigate the impact of inaccurate single-position calculations on path integration, the least squares method is employed to optimise all position points within the interval $t \in \left[t^B_{\text{enter}}, t^B_{\text{loss}}\right]$. When $t \in \left[t^B_{\text{enter}}, t^B_{\text{loss}}\right]$, the robot relies solely on path integration to perform localisation, which inevitably leads to cumulative errors and requires correction based on the positions at moments t^A_{loss} and t^B_{enter} derived from landmark calculations. The position $r^{\text{int}}(t) = \left(x^{\text{int}}, y^{\text{int}}\right)$ represents the robot's position at time t obtained through path integration, whereas $r^{\text{cor}}(t) = \left(x^{\text{cor}}, y^{\text{cor}}\right)$ denotes the robot's position at time t after correction, as given in Equations (10) and (11).

$$x^{\text{cor}} = \beta\left[\Delta x^{\text{int}} \cos\Delta\gamma - \Delta y^{\text{int}} \sin\Delta\gamma\right] + x^{\text{vis}}\left(t^A_{\text{loss}}\right), \quad (10)$$

$$y^{\text{cor}} = \beta\left[\Delta x^{\text{int}} \sin\Delta\gamma + \Delta y^{\text{int}} \cos\Delta\gamma\right] + y^{\text{vis}}\left(t^A_{\text{loss}}\right). \quad (11)$$

In Equations (10) and (11), β represents the correction ratio, and $\Delta\gamma$ denotes the correction angle, with their mathematical expressions given as follows:

$$\beta = \left|r^{\text{vis}}\left(t^A_{\text{loss}}\right) r^{\text{vis}}\left(t^B_{\text{enter}}\right)\right| / \left|r^{\text{vis}}\left(t^A_{\text{loss}}\right) r^{\text{int}}\left(t^B_{\text{enter}}\right)\right|, \quad (12)$$

$$\Delta\gamma = \gamma^{\text{vis}} - \gamma^{\text{int}}. \quad (13)$$

In Equations (12) and (13), $\left|r^{\text{vis}}\left(t^A_{\text{loss}}\right) r^{\text{vis}}\left(t^B_{\text{enter}}\right)\right|$ represents the distance between points $r^{\text{vis}}\left(t^A_{\text{loss}}\right)$ and $r^{\text{vis}}\left(t^B_{\text{enter}}\right)$ along the line segment, and $\left|r^{\text{vis}}\left(t^A_{\text{loss}}\right) r^{\text{int}}\left(t^B_{\text{enter}}\right)\right|$ is the same. γ^{vis} denotes the angle between the line connecting points $r^{\text{vis}}\left(t^A_{\text{loss}}\right)$ and $r^{\text{vis}}\left(t^B_{\text{enter}}\right)$ and the positive direction of the world coordinate system x-axis, whereas γ^{int} represents the angle between the line connecting points $r^{\text{vis}}\left(t^A_{\text{loss}}\right)$ and $r^{\text{int}}\left(t^B_{\text{enter}}\right)$ and the positive direction of the world coordinate system x-axis. In Equations (10) and (11), Δx^{int} and Δy^{int} represent the coordinate differences, as given in Equations (14) and (15).

$$\Delta x^{\text{int}} = x^{\text{int}} - x^{\text{vis}}\left(t^A_{\text{loss}}\right), \quad (14)$$

$$\Delta y^{\text{int}} = y^{\text{int}} - y^{\text{vis}}\left(t^A_{\text{loss}}\right). \quad (15)$$

Through the aforementioned computational process, the position output from path integration can be updated, thereby achieving the correction of localisation errors.

3.3 | Optimisation of the Navigation Trajectory

Once navigation habits are established, the navigation path is stored in CA1 place cells, with each cell encoding information for a single waypoint. Unfortunately, the Sn-Plast model tends to get trapped in local optima, resulting in suboptimal paths.

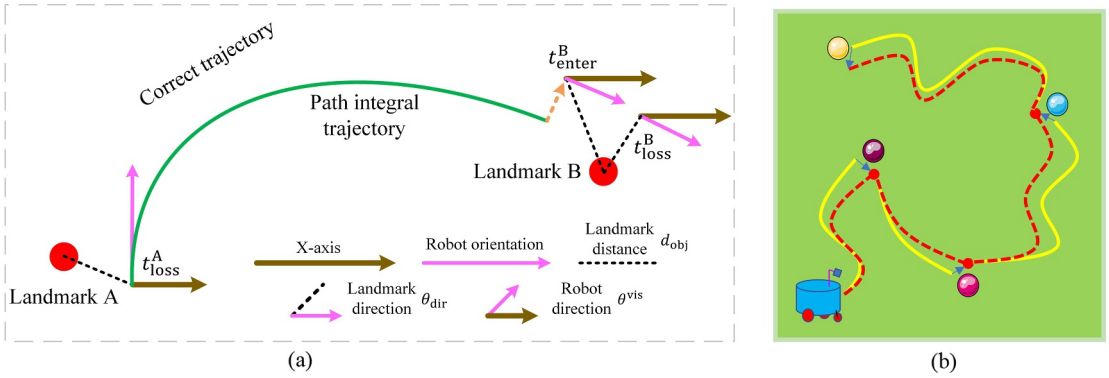


FIGURE 4 | Schematic diagram of the cumulative error correction method. (a) Geometric relationship of cumulative error correction process. (b) Operational mechanism.

However, mammals possess the ability to flexibly adjust their navigation paths, enabling adaptive and efficient navigation in complex environments. To address this, a dynamic self-organisation strategy of CA1 place cells is proposed for path optimisation. The core idea of this strategy is to treat the stored path as a chain of CA1 place cells in which each cell is coupled to its neighbours so that the whole chain behaves like an elastic string: Cells iteratively pull one another towards the straight line connecting them, progressively removing detours and shortening the trajectory while preserving its endpoints. Equations (16–22) formalise this neighbour-coupled update, and Equations (23–27) then constrain it so that the contraction never violates obstacle boundaries.

Let the ith cell be represented by the set $e^i = \{\beta^i, p^i\}$, where $p^i = \left(p_x^i, p_y^i\right)$ denotes the coordinates of the place field centre, and β^i is the agent angle at p^i relative to the heading direction. Initially, the line connecting the centres of cells e^i and e^{i+1} forms a continuous clamp angle along the x-axis. At time t, the update to the ith place field centre is $\Delta p^i(t) = \left(\Delta p_x^i(t), \Delta p_y^i(t)\right)$, with the corresponding angle correction $\Delta\beta^i(t)$. The mathematical expressions are given below.

$$\Delta p_x^i(t) = p_x^i(t) + \tau_r d^i \cos\left(\beta^i(t) + \varphi_i\right), \tag{16}$$

$$\Delta p_y^i(t) = p_y^i(t) + \tau_r d^i \sin\left(\beta^i(t) + \varphi_i\right), \tag{17}$$

$$\Delta\beta^i(t) = \mathrm{Delt}\left(\beta^{i+1}(t), \beta^i(t)\right). \tag{18}$$

In Equations (16–18), τ_r denotes the relaxation factor. An excessively large τ_r will result in insufficient path shortening, leaving the path overly long; conversely, a value that is too small will lead to path distortion and induce iterative oscillations that prevent convergence. Delt stands for the angle-difference-solving function, and d^i stands for the initial distance between the firing field centres of place cells e^i and e^{i+1}, with φ_i being the initial difference between β_i and β_{i+1}. Together, they quantify how much the current waypoint's position and heading should be adjusted to reduce the local deviation of the path. Then, the coordinate correction formulas for the firing field centre are provided below.

$$p^i(t+1) = p^i(t) + \delta\left(p^{i+1}(t) - \Delta p^i(t)\right), \tag{19}$$

$$p^{i+1}(t+1) = p^{i+1}(t) - \delta\left(p^{i+1}(t) - \Delta p^i(t)\right). \tag{20}$$

The mathematical expression for heading-direction angle correction is as follows:

$$\beta^i(t+1) = \beta^i(t) + \delta\,\Delta\beta^i(t), \tag{21}$$

$$\beta^{i+1}(t+1) = \beta^{i+1}(t) - \delta\,\Delta\beta^i(t). \tag{22}$$

In Equations (19–22), δ is the correction increment. An excessively large value of δ will cause the path length to oscillate during the optimisation process and fail to converge; conversely, an overly small value will reduce the optimisation speed of the path. The ith cell and its neighbour are displaced by equal and opposite increments of δ along the deviation direction, which symmetrically draws the two adjacent waypoints towards their connecting line; repeated over the whole chain, this monotonically reduces the total path length. Because the update is driven purely by local geometric deviation rather than by the reward signal that originally trapped Sn-Plast in a local optimum, the optimisation refines the route without reintroducing that local-optimum problem.

Because real environments usually contain obstacles, their effects must be incorporated into path optimisation. Therefore, this study divides the optimisation process according to the firing properties of BVCs. To keep the contraction collision-free, obstacle geometry sensed by BVCs is injected into the same chain through the following steps: The receptive response to obstacles (Equation 23) yields, at each cell, the distance to the nearest obstacle (Equation 24); a cell whose distance falls below a safety threshold is locked in place (Equations 25 and 26); and the elastic straightening of Equations (19–22) is then applied only to the free cells between consecutive locked cells (Equation 27). In this way, the path is optimised piecewise between obstacle-anchored points, so shortening never pushes the trajectory into an obstacle.

During robotic exploration, LiDAR is used to acquire obstacle data, which are then discretised. The obstacle set is denoted as $\mathrm{OBS} = \left\{\mathrm{obs}^1, \mathrm{obs}^2, ..., \mathrm{obs}^k\right\}$ and is subsequently input to the BVCs' receptive-field model. As with OVCs, BVCs' activity is expressed via receptive fields. d_{Bu} represents the farthest distance perceivable by perception, which correlates with the sensing radius of the LiDAR. Bu_layer denotes the number of layers in the receptive network, and Bu_ring indicates the number of units in each layer. Regarding these two parameters, larger values yield more precise encoding at the expense of increased computational overhead; conversely, smaller values result in coarser encoding but reduce computational overhead. Within the BVCs' receptive-field network, $\mathrm{Bu_streng}_{ij}$ represents the receptive intensity of the jth receptive unit in the ith layer, as shown in Equation (23).

$$\mathrm{Bu_streng}_{ij} = \sum_{k=1}^{\mathrm{Num_{obs}}} e^{-\left\| \mathrm{obs}^k - \mathrm{Bu_{pos}}_{ij} \right\|^2 / \sigma_{\mathrm{Bu}}}. \tag{23}$$

In Equation (23), $\mathrm{Num_{obs}}$ is the cardinality of OBS, $\mathrm{Bu_pos}_{ij}$ is the location encoded by the sensory unit, and σ_{Bu} is the receptive-field tuning parameter. Throughout navigation path optimisation, the system must continuously assess in real time the distance from the current trajectory to nearby obstacles. Let the minimum distance between the ith cell's firing centre and the nearest obstacle at time t be defined as $l_{\min}^i(t)$. The formula is presented below.

$$l_{\min}^i(t) = \min\left(\left\| p^i(t) - \mathrm{obs}^k \right\| \right). \tag{24}$$

As the navigation path is optimised, once a place cell's firing field centre gets sufficiently close to an obstacle and the distance falls below a threshold, that centre is locked in place. Let $P_{\mathrm{fix}}(t)$ denote the set of place cells with fixed firing fields at time t. When $t = 0$, $P_{\mathrm{fix}}(t) = \left\{p0(t), p^{N_{\mathrm{hpc}}}(t)\right\}$. The update rule for $P_{\mathrm{fix}}(t)$ across each iteration is given below.

$$P_{\mathrm{fix}}(t+1) = \mathrm{sort}\left(P_{\mathrm{fix}}(t) \cup \left\{p^{i \in i_{\mathrm{fix}}}(t+1)\right\}\right), \tag{25}$$

$$i_{\text{fix}} = \left\{ l^{i}_{\min}(t+1) \le l_{\text{th}} \,\middle|\, l^{i}_{\min}(t+1) - l^{i}_{\min}(t) < 0 \right\}. \tag{26}$$

In Equations (25) and (26), the 'sort' operation sorts the elements in the set from smallest to largest in ascending order, with l_{th} representing the shortest-distance judgement threshold. l_{th} is configured according to the physical size of the robot. Setting this threshold too large will cause the robot to stay excessively far from obstacles, resulting in an overly long optimised path; conversely, setting it too small may lead to collisions between the robot and obstacles during navigation. Let $\text{Cor}(i, i_{\text{start}}, i_{\text{end}})$ represent the deviation correction term for each place cell located between the firing centre indices i_{start} and i_{end}. The set $P_{\text{fix}}(t)$ is defined such that its ith component is $p^{i}_{\text{fix}}(t)$. The complete correction mechanism for all place cells is therefore described by the equation below.

$$\text{Cor}\left(i, p^{i}_{\text{fix}}(t), p^{i+1}_{\text{fix}}(t)\right) \text{ for } p^{i}_{\text{fix}}(t), p^{i+1}_{\text{fix}}(t) \in P_{\text{fix}}(t). \tag{27}$$

The above steps can optimise the navigation path in segments to ensure that the method can operate correctly in environments with obstacles.

3.4 | Feedback of the Navigation Trajectory

After path optimisation, the optimised navigation trajectory must be fed back into the SNN to establish a stable memory representation of the optimised path within the model. The principle of this feedback is supervised consolidation. The optimised trajectory is converted into the firing pattern that the network should produce when traversing it, termed the theoretical firing sequence, and this sequence is then used as a teaching signal to retrain the CA3-to-action-cell synapses so that the optimised behaviour becomes the network's spontaneous output and need not be rediscovered. Equations (28–33) construct this theoretical sequence, and Equation (34) performs the weight update that imprints it.

Set the path point stored by the currently active place cell as p^{i} and the next consecutive path point as p^{i+1}, which is then denoted as l^{i}. The angle θ_{te} between l^{i} and the positive x-axis (world coordinate system) is given below.

$$\theta_{\text{te}} = \arctan\left(\left(p^{i}_{y} - p^{i+1}_{y}\right) \Big/ \left(p^{i}_{x} - p^{i+1}_{x}\right)\right). \tag{28}$$

Next, the theoretical firing sequences of action cells and CA3 place cells are calculated along the optimised trajectory, as formulated in Equations (29) and (30).

$$R^{j}_{\text{ac}} = \begin{cases} 1 & e^{-\vartheta_{\text{AC}} \cdot \text{Delt}\left(\theta_{\text{te}}, \frac{2j\pi}{N_{\text{AC}}}\right)} \ge R^{\text{TH}}_{\text{AC}} \\ 0 & e^{-\vartheta_{\text{AC}} \cdot \text{Delt}\left(\theta_{\text{te}}, \frac{2j\pi}{N_{\text{AC}}}\right)} < R^{\text{TH}}_{\text{AC}} \end{cases}, \tag{29}$$

$$R^{k}_{\text{pc}} = \begin{cases} 1 & e^{-\frac{\left\| p^{i} - r^{k}_{\text{pc}} \right\|^{2}}{\sigma_{\text{CA3}}}} \ge \text{rand} \\ 0 & e^{-\frac{\left\| p^{i} - r^{k}_{\text{pc}} \right\|^{2}}{\sigma_{\text{CA3}}}} < \text{rand} \end{cases}. \tag{30}$$

In Equation (29), R^{j}_{ac} is the firing rate of the jth action cell, $R^{\text{TH}}_{\text{AC}}$ is the threshold, ϑ_{AC} is the action-cell firing-rate adjustment factor, and N_{AC} is the number of action cells. The effects of $R^{\text{TH}}_{\text{AC}}$ and ϑ_{AC} on the module are analogous. Excessive values lead to oversharp tuning, meaning that too few firing action cells are available to drive navigation behaviour; deficient values cause overactivation of firing action cells, destroying directional selectivity and causing actions to be averaged out and smeared. In Equation (30), R^{k}_{pc} is the firing rate of the kth place cell, r_{pc} denotes the population firing centre of place cells, σ_{CA3} is the place-field-radius adjustment factor, and rand is a random number drawn from the interval [0, 1]. An excessively large σ_{CA3} results in a place field that is too wide, leading to low trajectory accuracy; conversely, an overly small value causes the place field to be too narrow and the firing to be excessively sparse, making the position representation prone to gaps.

At each discrete simulation step of duration Δt, a place cell is assigned an instantaneous firing probability equal to its tuning value $\exp\left(-\left\| p_i - r_{\text{pc}_k} \right\|^{2} / \sigma_{\text{CA3}}\right)$, and a spike is emitted whenever this probability exceeds a random number, which is exactly a Bernoulli realisation of a Poisson process. The action cells are correspondingly activated when their directional tuning $\exp\left(-\vartheta_{\text{AC}} \cdot \text{Delt}\left(\theta_{\text{te}}, 2j\pi / N_{\text{AC}}\right)\right)$ exceeds the threshold $R^{\text{TH}}_{\text{AC}}$ (Equation 29), selecting the cells aligned with the required heading. The spikes emitted over the dwell time at each waypoint are collected into the spike train $Y_j(t) = \sum_f \delta\left(t - t_j\right)$ used in Equation (31).

Taken together, Equations (29) and (30) define the theoretical firing pattern that the action cells and CA3 place cells should exhibit at each point along the optimised trajectory: The place cells encode where the agent is on the path, whereas the action cells, tuned to their preferred directions, encode the movement required at that location. This firing pattern is what drives behaviour. At each time step, the population activity of the action cells is decoded into a concrete movement so that the cell most aligned with the required direction contributes most to the action. The resulting action $a(t)$ is therefore obtained by combining the action cells' firing through their spiking sequences and a filter kernel, as formulated in Equation (31).

$$a(t) = \frac{1}{N_{\text{ac}}} \sum_{j} \left(Y_j \circ \gamma\right)(t) a_j. \tag{31}$$

In Equation (31), $a(t)$ represents the action guided by the jth action cell, Y_j denotes the spiking sequence of the jth action cell, and γ is the filter kernel. The corresponding detailed mathematical expressions are provided in Equations (32) and (33).

$$a_j = a_0\left(\sin\left(\theta_j\right), \cos\left(\theta_j\right)\right), \tag{32}$$

$$\gamma(t) = \frac{e^{\frac{-t}{\tau_\gamma}} - e^{\frac{-t}{\nu\gamma}}}{\tau_\gamma - \nu_\gamma} \Theta(t). \tag{33}$$

In Equation (32), a_0 denotes the action step length, and θ_j represents the preferred movement direction of the jth action cell. In Equation (33), $\Theta(t)$ is the reward function, whereas τ_γ and ν_γ are time-constant and gain parameters of the low-pass

filter, respectively. When the agent reaches the target, synaptic weights are updated using a plasticity rule that combines STDP with eligibility traces, as described in Equation (34).

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

In Equation (34), η denotes the learning rate, A denotes the neuromodulatory gating factor, and W stands for the STDP learning window, mathematically expressed as $W(x) = e^{-|x|/\tau}$. ψ is the polarity factor. The variables $\bar{t}_j$ and $\bar{t}_i$ represent the arrival times of the jth action-cell spike and the ith place-cell spike. Driven by this theoretical sequence, Equation (34) strengthens the synapses from CA3 place cells to action cells through STDP gated by eligibility traces so that presenting a given place-cell activity later elicits the action that the optimised path prescribes. The optimised route is thereby consolidated into the SNN as a stable habit. During subsequent navigation, CA3 place cells' firing drives the retrained SNN to emit the correct action cells, the resulting motion is kept accurate by the visual stream correction described in Section 3.2, and any residual suboptimality is again refined by the CA1 self-organisation described in Section 3.3, closing the 'perception-optimisation-consolidation' loop. The operational mechanism of navigation path feedback is illustrated in Figure 5.

3.5 | Dynamic Navigation Tasks

In a static setting, a robot can determine a viable navigation route after a limited number of explorations. In contrast, dynamic real-world conditions demand the capacity for rapid adaptation, requiring the robot to continuously sense, process and respond to unforeseen changes. Changes in navigation tasks typically involve two scenarios: (1) The original navigation path becomes blocked due to the addition of obstacles in the environment, and (2) the goal of the navigation task changes. In such cases, the robot must re-explore the environment to search for a new navigation path. Figure 6 depicts the overarching workflow of the proposed navigation method.

4 | Results and Evaluations

4.1 | Experimental Procedure

To evaluate the performance of the proposed method, this study designs experiments across three levels: 2D MATLAB simulations for controlled comparisons and ablation studies under noise-free conditions; 3D Webots simulations for localisation, noise robustness and habit reshaping; and real-world TurtleBot2 trials for feasibility validation. The proposed method is compared against several RL baselines (Q-learning, SARSA, R-IAC, meta-RL and Ac-DRL) and classical planners (A* and rolling-window RRT). Additionally, ablation variants (Sn-Plast, Sn-Plast + PO and Sn-Plast + PO + PF) are evaluated to isolate the contribution of each module.

To quantify performance, six quantitative metrics are used. Average navigation path length L_{avg} (Equation 36) and average exploration episodes for habit formation measure path efficiency and convergence speed. Convergence success rate P_{con} (Equation 35) measures the reliability of habit formation. Trajectory error E_{traj}, the mean absolute error between the corrected and ground-truth trajectories, measures localisation accuracy under motion noise. Convergence count N_{con}, the number of successful convergences over repeated trials, measures robustness to motion noise. For the maze comparison, the standard deviation of path length across different initial headings quantifies the invariance of the planned route to the initial motion angle. In addition, single-step time and memory consumption measure the computational overhead and practical deployability on robot platforms. Qualitatively, we assess landmark and obstacle perception accuracy, trajectory smoothness, behavioural consistency with rodent studies and the heading-direction and linear-velocity stability of the physical robot.

Simulation experiments are conducted on a Dell OptiPlex 7080 server equipped with an NVIDIA RTX 3070 GPU, an Intel Core i9-10900K CPU, 32 GB of RAM and the Windows 10 operating system. To accelerate computation, all matrix operations are performed on the GPU. Parameter settings involve five modules: robot structural parameters, entorhinal-hippocampal CA3 path integration model, visual pathway calculation

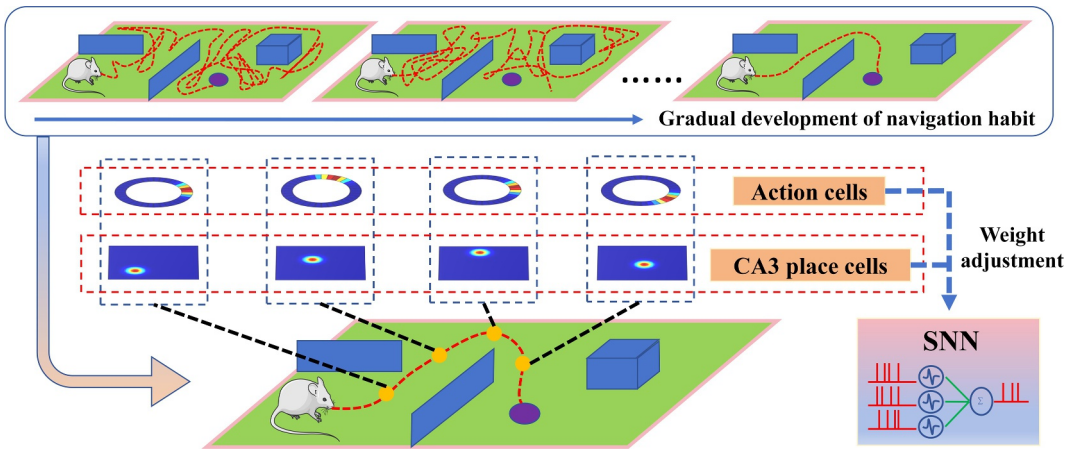


FIGURE 5 | Operational mechanism of navigation path feedback.

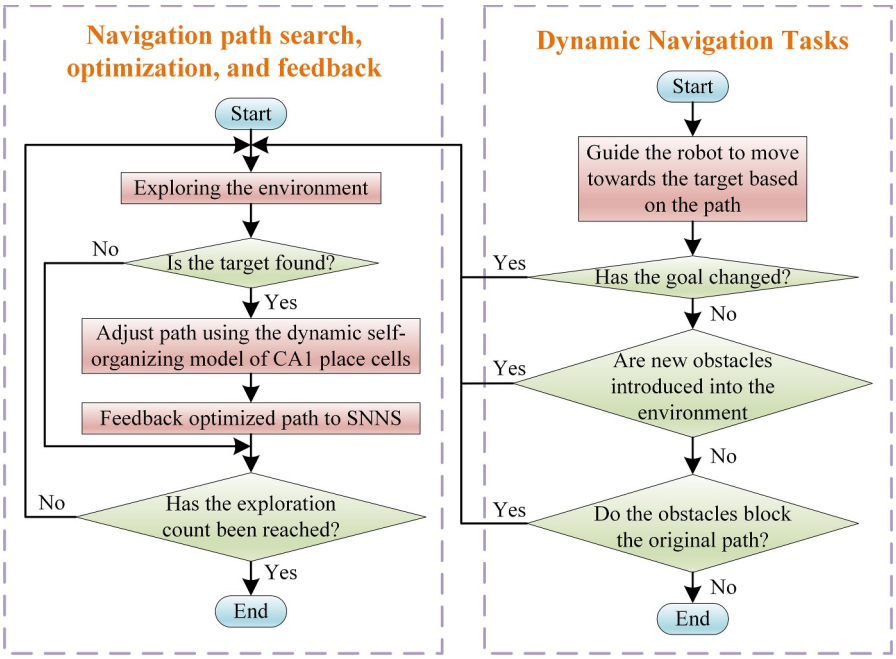


FIGURE 6 | Overarching workflow of the proposed navigation method.

model, hippocampal-prefrontal navigation model and path optimisation and feedback method. The Sn-Plast model is configured following Ref. [39]; the entorhinal-hippocampal CA3 model is configured following Ref. [42], whereas the 'what' stream model is configured following Ref. [44]. The remaining parameter settings are listed in Table 1.

4.2 | 2D Simulation Experiment

The 2D simulation experiments aim to evaluate the method's performance under ideal conditions (i.e., without motion noise or cumulative error), including convergence speed, navigation path length and parameter variations during model execution. To highlight the method's effectiveness, comparative experiments and ablation experiments are designed in this section.

4.2.1 | Navigation Habit Formation and Path Optimisation Experiments

Four distinct 2D spatial regions, each measuring 10 m × 10 m, are constructed for navigation experiments. Obstacles, start points and goal points are placed within each region. The Sn-Plast model guides the agent in exploring all four regions to develop corresponding navigation habits. Each habit formation process consists of 30 exploration episodes, with a maximum path length of 80 m per episode. The process of navigation habit formation without the path optimisation module is illustrated in Figure 7, where panels from left to right represent increasing exploration episodes. Figure 8 illustrates the navigation path optimisation process, with panels from left to right representing increasing iterations of the optimisation model. Figure 9a shows the variation in path length during optimisation, whereas Figure 9b presents the statistical comparison of path lengths before and after optimisation.

As shown in Figure 7, during the initial exploration phase, the agent employs a random exploration strategy, resulting in trajectories with pronounced random characteristics. Under the guidance of the Sn-Plast model, the agent gradually learns to navigate from the start point to the goal point through continuous exploration, thereby forming a navigation habit tailored to the current task. However, the navigation results reveal two key limitations: slow convergence and suboptimal paths. Slow convergence means that the agent may still fail to reach the goal in subsequent episodes, even though it has previously explored the goal. Suboptimal paths are characterised by tortuous trajectories that are far from the optimal route. Therefore, path optimisation is required. Figures 8 and 9 demonstrate that, as the number of optimisation iterations increases, the optimised path is significantly shorter than the original path, validating the effectiveness of the model in navigation path optimisation.

TABLE 1 | Relevant parameter settings.

Parameters	Value
h_{cam}	0.34 m
d_{cr}	0.42 m
d_{su}	0.29 m
Su_layer	10
Su_ring	36
d_{Bu}	3 m
Bu_layer	20
τ_r	0.3
δ	0.5
$R_{\text{AC}}^{\text{TH}}$	0.9
ϑ_{AC}	1000
σ_{CA3}	0.16
l_{th}	0.2 m
Bu_ring	72

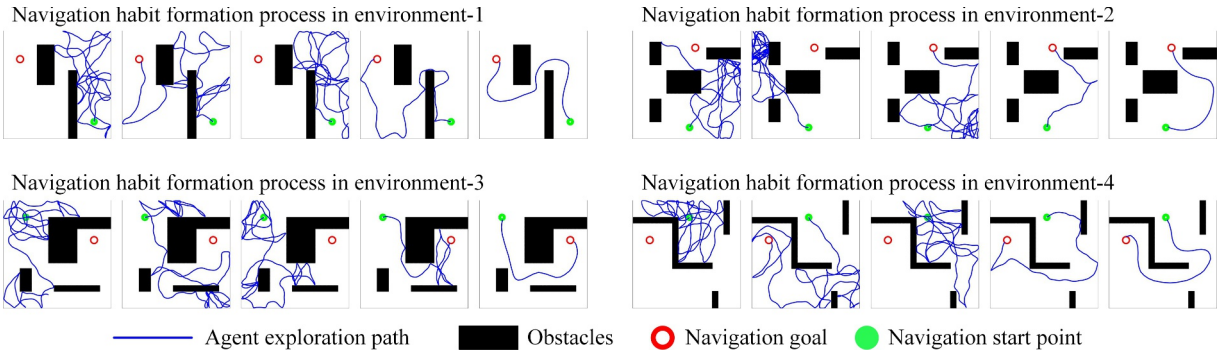


FIGURE 7 | Formation process of navigation habits without path optimisation.

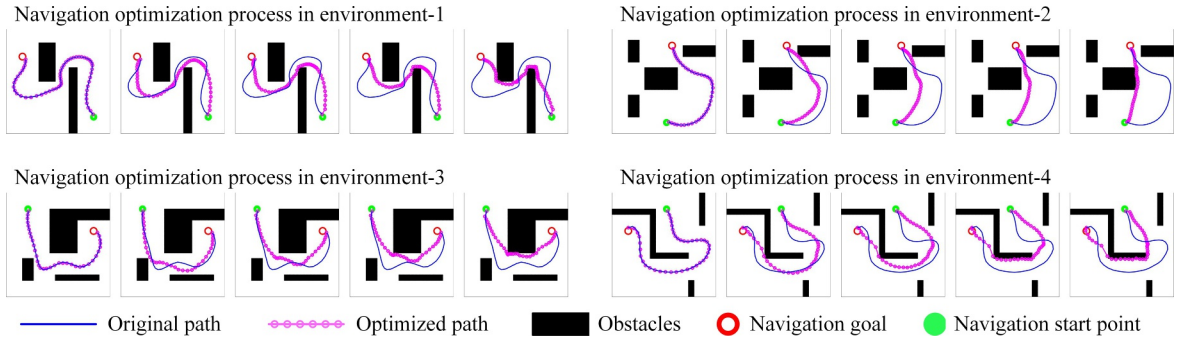


FIGURE 8 | Optimisation process of the navigation path.

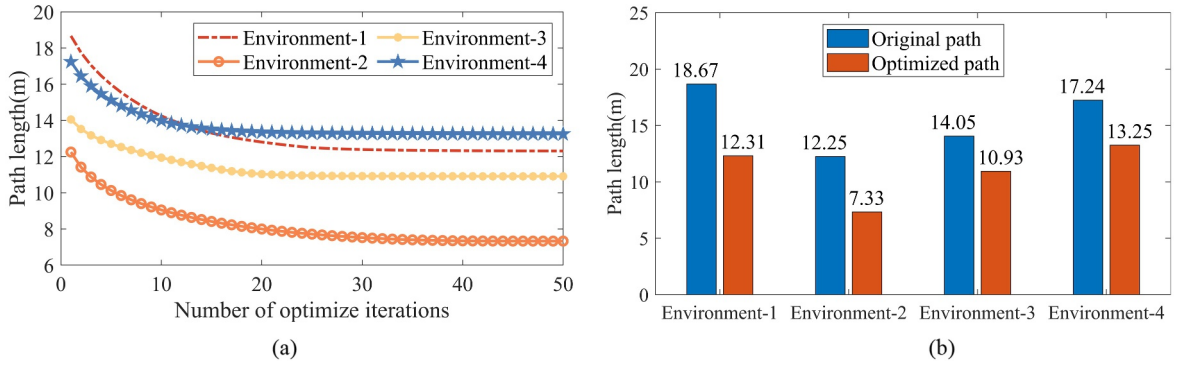


FIGURE 9 | Optimisation effect of navigation path length. (a) Path length variation with optimisation iterations. (b) Path length before and after optimisation.

4.2.2 | *Comparison and Ablation Experiments*

To highlight the advantages of the proposed method, it is compared with Q-learning [36], SARSA [45], R-IAC [30], meta-RL [46] and Ac-DRL [47]. The proposed navigation method comprises three modules: the Sn-Plast module, the path optimisation (PO) module and the path feedback (PF) module. To evaluate the contribution of each module, ablation studies are conducted. Specifically, Sn-Plast represents habit formation without path optimisation, Sn-Plast + PO denotes path optimisation without feedback, and Sn-Plast + PO + PF corresponds to the complete proposed method. Each habit formation process consists of 30 exploration episodes, with a maximum path length of 80 m per episode. The agents, guided by each algorithm, explore the four spatial regions to develop corresponding navigation habits. Figure 10 presents the variation in navigation path length with exploration episodes. To further validate the proposed method's advantages in convergence speed and navigation path length, this study conducts comparative experiments from three perspectives: average path length, probability of discovering the goal after navigation habit formation and average number of exploration episodes required to form the habit. To quantify convergence, it is judged to have converged (habit formed) when the agent, upon target discovery, successfully navigates to it across the next four episodes. The convergence success rate P_{con} is defined by Equation (35).

$$P_{\text{con}} = \frac{\text{Sum}_{\text{con}}}{\text{Sum}_{\text{find}}}. \tag{35}$$

In Equation (35), Sum_{find} represents the number of times the goal region is discovered, and Sum_{con} denotes the number of

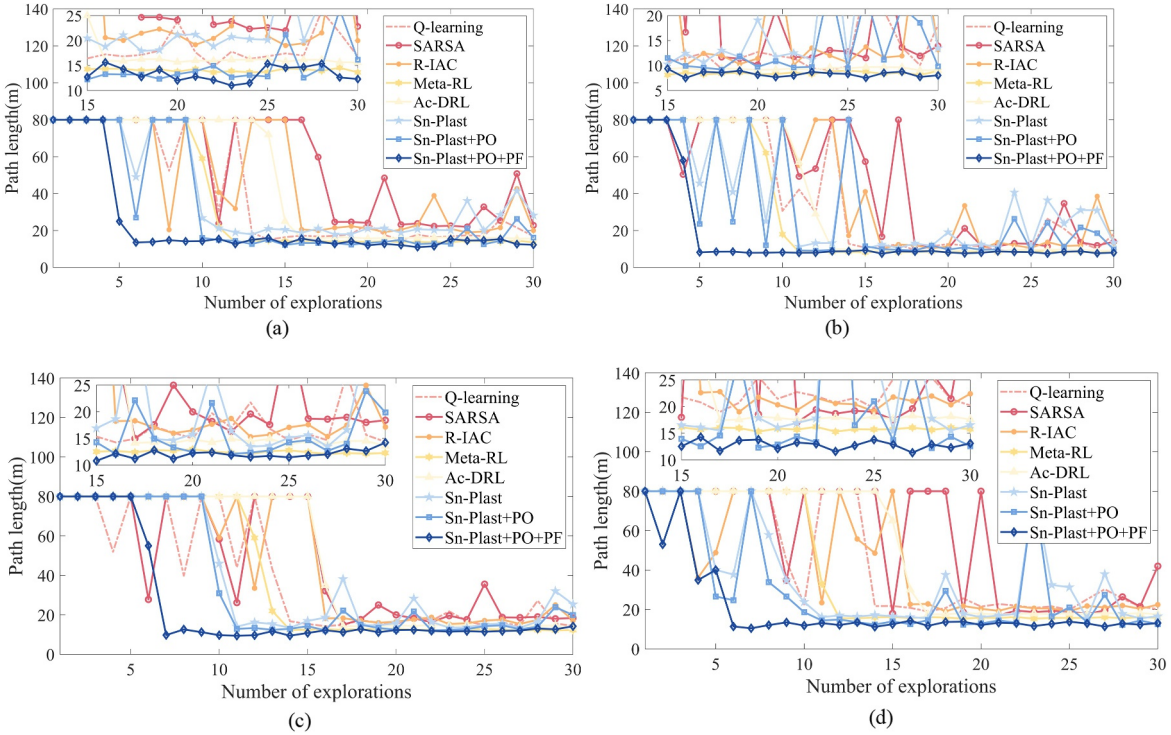


FIGURE 10 | Statistical results of navigation path length variation for each algorithm. (a) Environment-1 navigation results. (b) Environment-2 navigation results. (c) Environment-3 navigation results. (d) Environment-4 navigation results.

times the agent can reach the goal in the next four consecutive episodes. The average navigation path length L_{avg} is the average path length over all episodes in which the goal region is discovered, as defined by Equation (36).

$$L_{\text{avg}} = \frac{\Sigma_{j=1}^{\text{Sum}_{\text{find}}} L_j}{\text{Sum}_{\text{find}}}. \tag{36}$$

In Equation (36), L_j represents the exploration path length for the jth discovery of the goal region. The total number of exploration episodes required to complete habit formation is defined as the average number of episodes needed for the algorithm to converge and consistently reach the goal region. This metric reflects the convergence speed and robustness of the algorithm in discovering the goal. For a fair comparison, each group of experiments is repeated 20 times, with the results presented in Table 2.

Figure 10 and Table 2 show that all algorithms enable the agent to develop task-specific navigation habits. Among them, SARSA performs the worst in terms of convergence speed and path efficiency. Q-learning, although producing shorter navigation paths, is slower in initial goal finding than Sn-Plast. The primary issue with Sn-Plast is its lack of convergence reliability, as it fails to consistently reach the goal despite occasional rapid success. The Sn-Plast + PO variant follows a similar convergence trend to Sn-Plast but achieves shorter paths than Q-learning due to the path optimisation module. However, without feedback, it cannot form a stable memory of the optimised path, leading to delayed convergence. Meta-RL and Ac-DRL show improvements in adaptability through experience replay mechanisms, achieving high convergence probabilities. However, they still yield longer average route lengths and require more exploration episodes compared to the proposed method, as they do not incorporate biologically inspired route optimisation and memory consolidation feedback, leading to less efficient geometric refinement and slower habit stabilisation. In comparison, Sn-Plast + PO + PF achieves the best performance across all three metrics, validating the effectiveness of the PO and PF modules in path optimisation and convergence speed.

These results stem from the complementary mechanisms of the three modules. The Sn-Plast rule augments an STDP window with eligibility traces and a goal-triggered neuromodulatory gating factor so that when reward is delivered at the goal, credit is retroactively assigned to the recently co-active CA3 place cells and prefrontal action cells. This temporal bridging links actions to their delayed outcomes after only a few trials, steering subsequent exploration towards goal-relevant regions and locating the goal in fewer episodes. Sn-Plast alone, however, tends towards local optima and never consolidates the discovered route, yielding suboptimal paths and unreliable convergence. The PO module refines the stored path by treating the CA1 place-cell sequence as a self-organising chain that straightens while locking cells near obstacles sensed through the BVC receptive fields, thereby shortening the route. Yet, as the Sn-Plast + PO ablation shows, this gain remains confined to CA1 and does not improve convergence until the PF module closes the loop. The PF module replays the optimised trajectory as theoretical action-cell and CA3 place-cell firing sequences that drive the same STDP-with-eligibility rule to retrain the SNN, imprinting the path into the synaptic weights as a stable, retrievable habit. This consolidation simultaneously raises the post-habit success probability and shortens the time required to stabilise.

TABLE 2 | Statistical results of navigation experiments for various algorithms.

Environment	Algorithm	$L_{avg}(m)$	P_{con}	Average exploration episodes for navigation habit formation
1	Q-learning	16.59	84.2%	11.2
	SARSA	19.56	67.6%	16.5
	R-IAC	17.44	85.8%	17.3
	Meta-RL	13.98	91.3%	8.5
	Ac-DRL	15.52	86.7%	12.4
	Sn-Plast	18.97	73.3%	13.8
	Sn-Plast + PO	13.02	73.3%	13.8
	Sn-Plast + PO + PF	**12.48**	**96.1%**	**6.7**
2	Q-learning	11.87	79.9%	8.3
	SARSA	12.62	76.5%	12.4
	R-IAC	11.14	81.3%	7.9
	Meta-RL	8.02	93.5%	8.2
	Ac-DRL	9.15	89.8%	10.2
	Sn-Plast	12.90	80.7%	12.9
	Sn-Plast + PO	7.73	80.7%	12.9
	Sn-Plast + PO + PF	**7.52**	**97.8%**	**7.1**
3	Q-learning	16.28	77.4%	14.2
	SARSA	17.55	71.5%	22.8
	R-IAC	17.13	89.1%	16.1
	Meta-RL	12.36	90.9%	11.7
	Ac-DRL	13.64	85.5%	15.5
	Sn-Plast	18.93	78.4%	20.7
	Sn-Plast + PO	11.40	78.4%	20.7
	Sn-Plast + PO + PF	**11.16**	**97.3%**	**8.4**
4	Q-learning	18.51	85.7%	14.1
	SARSA	19.38	70.2%	20.9
	R-IAC	18.96	91.5%	9.6
	Meta-RL	14.91	94.6%	10.2
	Ac-DRL	16.80	90.1%	13.5
	Sn-Plast	19.73	74.8%	17.4
	Sn-Plast + PO	14.06	74.8%	17.4
	Sn-Plast + PO + PF	**13.70**	**98.2%**	**6.1**

4.3 | 3D Simulation Experiment

The 3D robot platform simulation experiments aim to evaluate the localisation error correction capability of the visual stream, navigation performance under path integration errors and navigation habit reshaping performance. Additionally, comparative navigation experiments are designed in the classic Tolman maze environment to further assess the method's performance.

4.3.1 | *Localisation Error Correction Experiments*

The 3D simulation area is set to 10 m × 10 m, with various visual landmarks placed in the environment. The robot explores arbitrarily and acquires landmark information from the camera. To investigate the impact of landmark quantity N_{obj} and noise on performance, multiple experiments are conducted. The parameters are defined as follows: N_{obj} is set to 0, 4, 7, 10 and 13. Angular and linear velocity noise are Gaussian noise with a mean of zero, and the specific formulas are as follows:

$$\tilde{\omega}(t) = \omega(t) + n_{\omega}(t), n_{\omega}(t) \sim \mathcal{N}\left(0, d_{\tilde{\omega}}\right), \tag{37}$$

$$\tilde{v}(t) = v(t) + n_{v}(t), n_{v}(t) \sim \mathcal{N}\left(0, d_{\tilde{v}}\right). \tag{38}$$

In Equations (37) and (38), $\omega(t)$ and $v(t)$ represent the true angular velocity and linear velocity, $n_{\omega}(t)$ and $n_{v}(t)$ represent noise values, and $\mathcal{N}\left(0, d_{\tilde{\omega}}\right)$ and $\mathcal{N}\left(0, d_{\tilde{v}}\right)$ represent Gaussian distributions with a mean of 0 and variances of $d_{\tilde{\omega}}$ and $d_{\tilde{v}}$, respectively. These noise parameters can be custom-configured within Webots by setting the encoder noise and gyroscope noise, respectively. The combinations of noise variances $\left(d_{\tilde{\omega}}, d_{\tilde{v}}\right)$ are specifically set to (0.002, 0.005), (0.005, 0.01) and (0.008,

0.015). For consistency, each parameter group is repeated 20 times, and the average absolute error between the corrected and ground-truth trajectories is computed as the trajectory error. The robot explores the environment for 500 s per trial. Figure 11 shows the landmark recognition and OVC responses during exploration. Figure 12a presents the trajectories within the first 50 s of a representative experiment, and Figure 12b displays the trajectory error statistics for the entire path.

Figure 11 shows that the visual stream model can accurately recognise character information and provides the corresponding bounding rectangles. Meanwhile, the OVC model precisely perceives and encodes landmark information in the environment. Figure 12a reveals that, as the robot moves within the designated area, noise in self-motion signals causes the raw path integration trajectory to gradually deviate from the ground-truth trajectory. In contrast, the landmark-corrected trajectory closely follows the actual path. As shown in Figure 12b, larger variances in angular and linear velocity noise result in greater error between the corrected and ground-truth trajectories. Conversely, a higher number of visual landmarks increases the opportunities for self-localisation during motion. Thus, under the same noise level, more landmarks lead to smaller localisation errors. These experimental results align with observed rodent behaviour.

4.3.2 | Navigation Habit Formation Experiments With Motion Noise

This subsection aims to evaluate the robot's navigation performance under motion noise. Three simulation environments are configured, each with an area of 10 m × 10 m, containing obstacles, landmarks, start points and goal points. Because the robot must perform real-time obstacle avoidance during exploration while simultaneously acquiring obstacle information for path optimisation, the obstacle perception capability is first validated, as shown in Figure 13. The results demonstrate that BVCs accurately detect and store obstacle information during exploration, thereby laying the foundation for subsequent path optimisation.

Then, multiple experiments are conducted to investigate the effects of landmark quantity N_{obj} and motion noise magnitude on navigation convergence performance. The landmark quantity N_{obj} is kept consistent with the localisation error correction experiments, whereas noise levels $d_{\tilde{\omega}}$ and $d_{\tilde{v}}$ are increased, with $d_{\tilde{\omega}}$ and $d_{\tilde{v}}$ set to 0 for the control group. The number of exploration episodes per habit formation process is set to 30, with a maximum path length of 80 m per episode. Each parameter group is repeated 30 times, and the convergence count during the experiments is recorded. Under varying noise levels, the convergence count of the navigation model varies with landmark quantity N_{obj}, as shown in Figure 14. Figure 15 illustrates the navigation process of the robot after developing navigation habits under $d_{\tilde{\omega}} = 0.008$, $d_{\tilde{v}} = 0.015$ and $N_{\text{obj}} = 13$. In the figure, purple cuboids represent environmental obstacles; red and green solid circles represent the goal and start points, respectively; yellow dashed lines represent the navigation path; and black dashed circles represent the robot's position. Figure 16a shows the corresponding navigation path under $d_{\tilde{\omega}} = 0.008$, $d_{\tilde{v}} = 0.015$ and $N_{\text{obj}} = 0$. In the figure, the blue dashed line represents the robot's actual navigation path when $N_{\text{obj}} = 0$, whereas the yellow dashed line indicates the path when $N_{\text{obj}} = 13$. Figure 16b presents the variation in navigation path length with exploration episodes under $d_{\tilde{\omega}} = 0.008$ and $d_{\tilde{v}} = 0.015$ for N_{obj} values of 0 and 13.

It can be observed that, in the absence of motion noise, the number of visual landmarks does not affect the convergence of the navigation process. Under the same noise level, a greater number of visual landmarks leads to more frequent convergence. Conversely, with the same number of landmarks, higher motion noise results in fewer convergence instances. The analysis of the experimental results is as follows: First, the navigation model computes the firing responses of all action cells based on the robot's current position to determine its next movement. The presence of motion noise introduces localisation errors, causing the robot to misjudge its position and issue incorrect movement commands at the current time step. The accumulation of such errors prevents the robot from reaching the goal correctly and may even lead to collisions with obstacles during motion, as

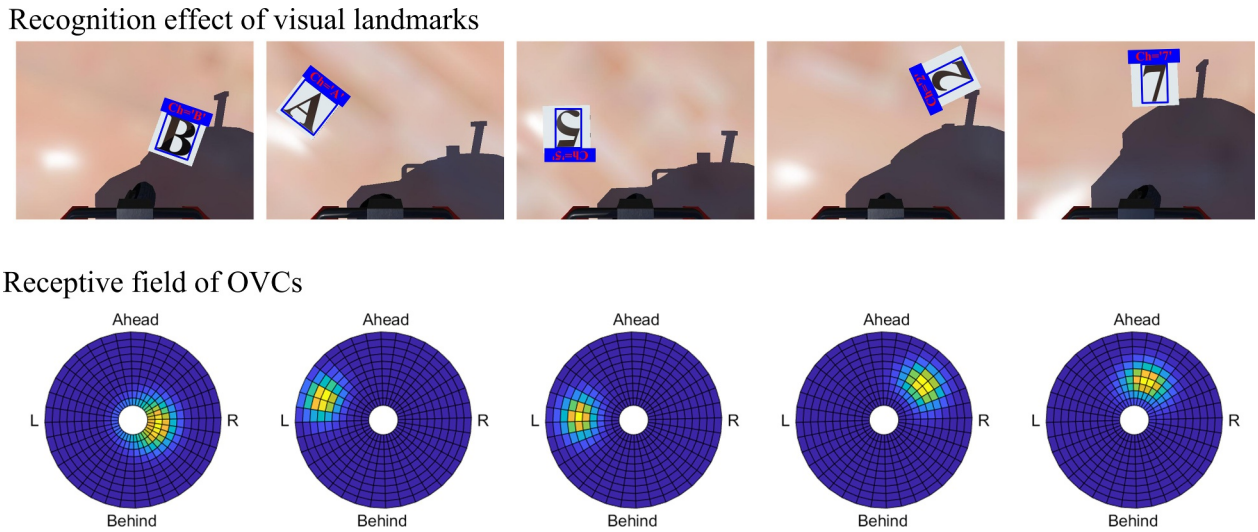


FIGURE 11 | Visual perception effect of robots during exploration.

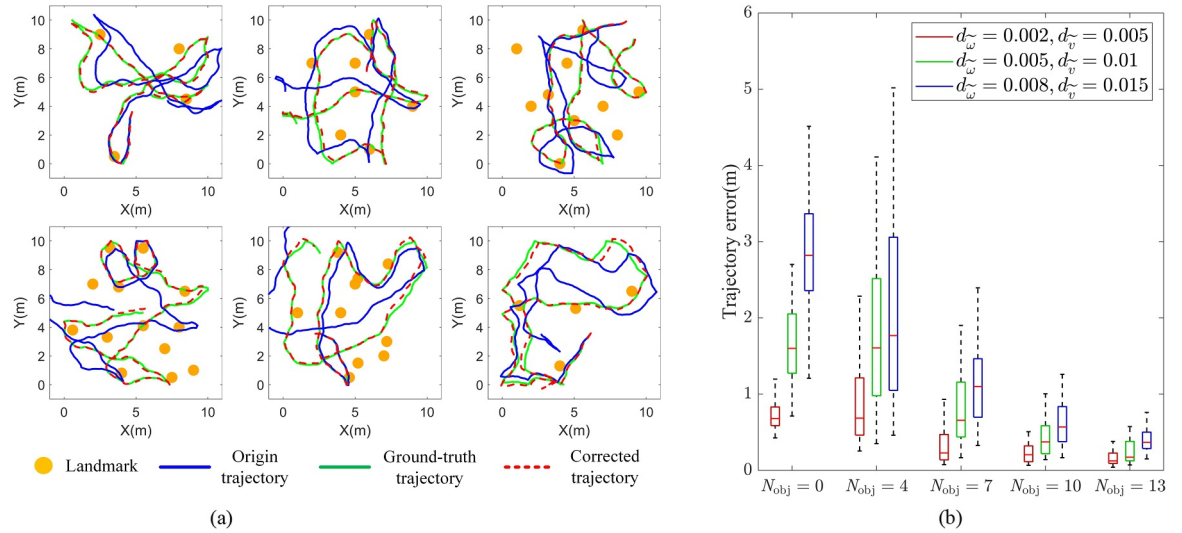


FIGURE 12 | Experimental results of trajectory correction. (a) Effect of partial trajectory correction. (b) Trajectory error statistics.

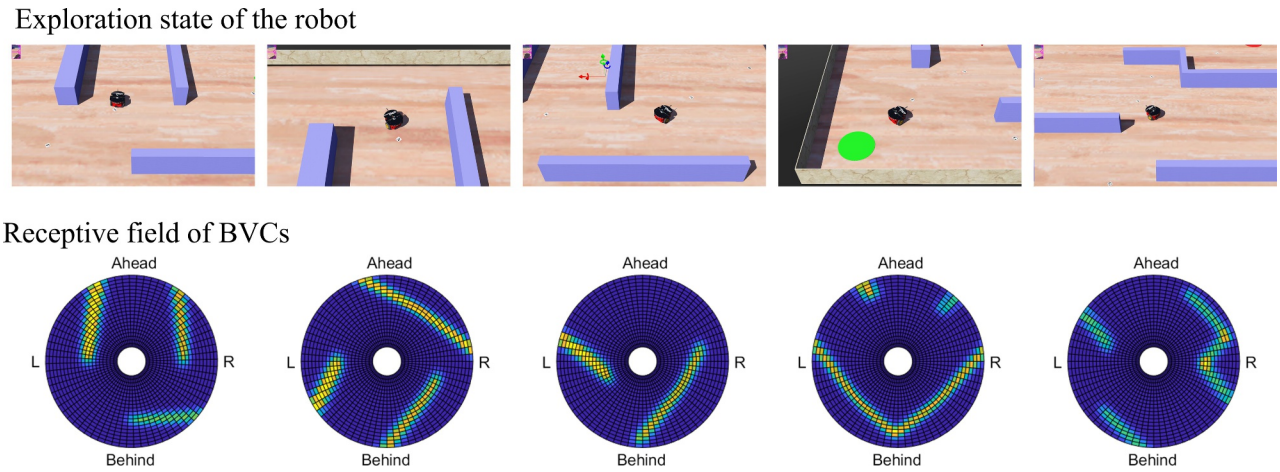


FIGURE 13 | Obstacle perception by robots during exploration.

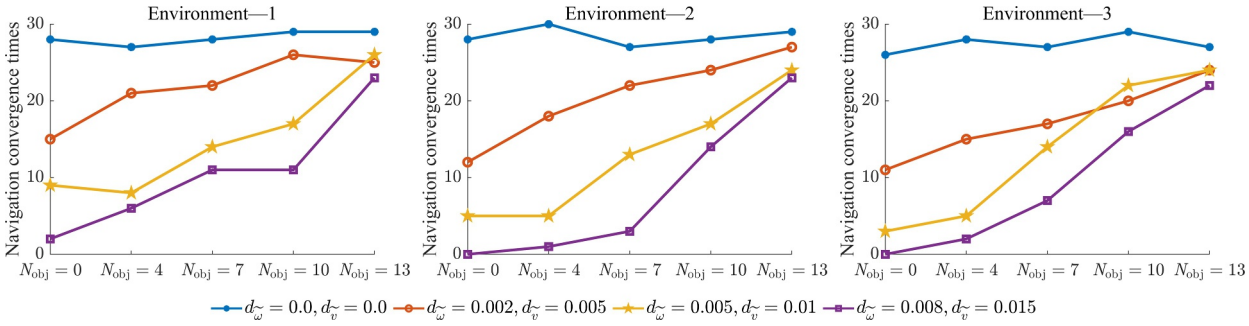


FIGURE 14 | Number of convergence times varies with the number of landmarks.

shown in Figure 16a. Thus, without visual landmarks, the navigation model fails to converge and cannot form a stable navigation habit. In contrast, the presence of visual landmarks improves localisation accuracy, reduces the probability of erroneous movement decisions and enables the stable formation of task-specific habits, as illustrated in Figure 16b. This stable habit, in turn, reliably guides the robot to complete goal-directed navigation, as demonstrated in Figure 15.

4.3.3 | *Navigation Habit Reshaping Experiments*

Navigation tasks are inherently nonstationary, requiring robots to possess strong adaptive capabilities. Upon changes in the navigation task, the robot must re-explore the environment to reform its navigation habits. To validate the adaptive performance of the proposed method, the following experiment is designed: The exploration process consists of 60 episodes, with the first 30 episodes dedicated to forming the habit for the

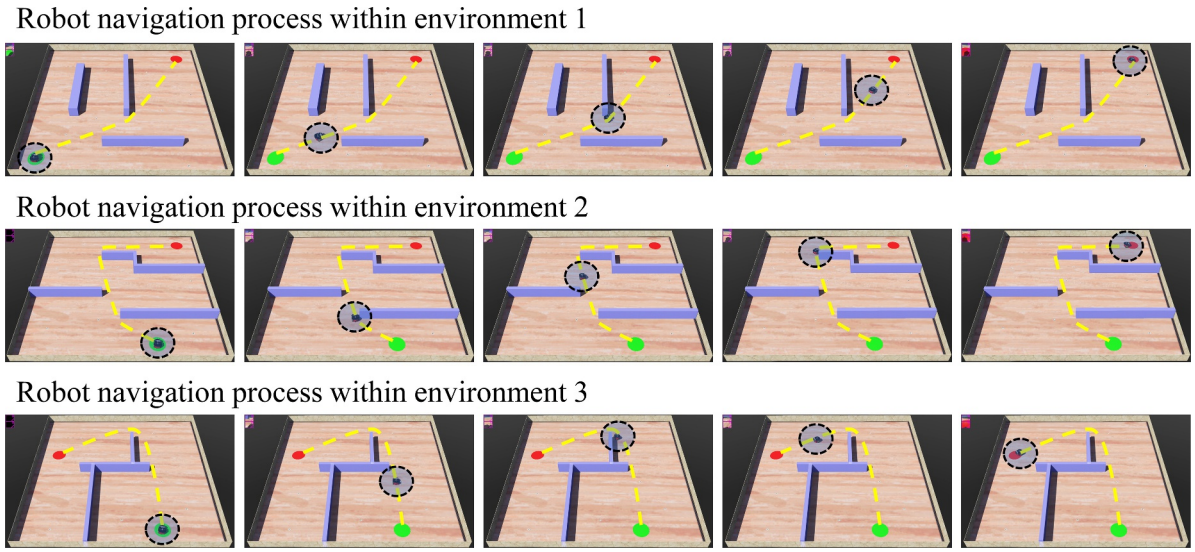


FIGURE 15 | Navigation process of robots after forming navigation habits.

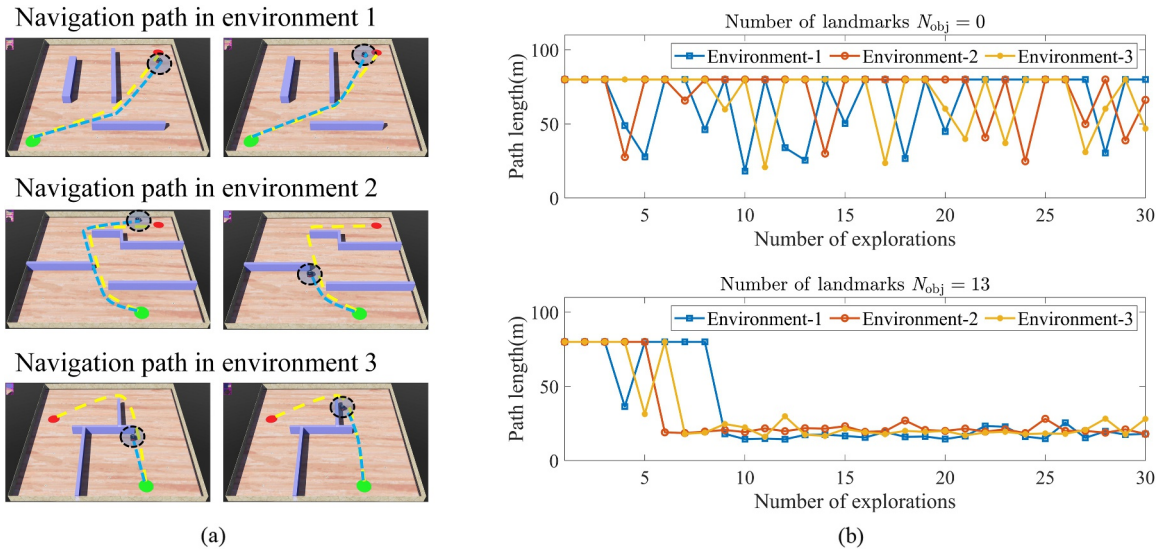


FIGURE 16 | Impact of visual landmarks on navigation performance. (a) Trajectory comparison. (b) Comparison of changes in path length.

original navigation task, followed by 30 episodes for reforming the habit for the new navigation task. The update involves adding obstacles or altering the goal position to complete the new task. To avoid contingency, each parameter group is repeated 20 times. The number of landmarks $N_{obj} = 13$ is set to 13, and the motion noise levels $d_{\tilde{\omega}}$ and $d_{\tilde{v}}$ are set to 0.005 and 0.01, respectively, consistent with previous experiments. Figure 17a presents the comparison of navigation paths before and after the change, where the pink cuboid represents the newly added obstacle and the blue dashed line indicates the navigation habit formed after reforming. Figure 17b presents the comparison of the average exploration episodes required for navigation habit convergence. Figure 18 shows the variation in navigation path length.

Figure 17a demonstrates that, upon a change in the navigation task, the proposed method effectively reguides the robot to re-explore the environment and rapidly form a habit adapted to the new task. Figure 17b reveals that the number of exploration episodes required for convergence after the task change is significantly lower than that required for the original task. This is primarily because the robot has already accumulated relevant experiential knowledge during the original task exploration. The new task shares similarities with the original task, allowing the robot to leverage prior experience through the experience replay mechanism to swiftly adjust its navigation strategy, thereby adapting more quickly and effectively to the new task.

4.3.4 | *Navigation Experiments in the Tolman Maze*

To facilitate comparison with animal experiments and further validate the method's effectiveness, a classic Tolman three-channel maze environment is constructed in Webots, incorporating visual landmarks, obstacles, navigation start points and goal points. The number of landmarks N_{obj} is set to 13, and the motion noise levels $d_{\tilde{\omega}}$ and $d_{\tilde{v}}$ are set to 0.005 and 0.01, respectively, consistent with prior experiments. Three navigation tasks are conducted within the environment, with differences between tasks arising from obstructions by different

channel obstacles in the maze. The spatial layout of the Tolman maze and the start, goal and final navigation paths for the three tasks are shown in Figure 19. In the figure, the yellow dashed line represents the robot's navigation path, pink cuboids denote obstructing channel obstacles, and red and green solid circles indicate the goal and start points, respectively.

Figure 19 shows that, in the face of complex maze environments, the robot can gradually learn and adaptively form effective navigation paths through learning. This observation aligns with phenomena observed in physiological studies: Rats gradually develop habitual behaviours during maze navigation. To demonstrate the superiority, it is compared with the A* algorithm [48] and the rolling-window rapid-exploring random tree (RRT) algorithm [49]. The comparison is conducted under different initial motion angles. Specifically, the initial motion angles are set to 0, $\pi/2$, π and $3\pi/2$. Figure 20a presents the navigation path length statistics for each method under different initial motion angles, whereas Figure 20b shows the average navigation path length statistics for each method across different initial motion angles.

Figure 20 shows that the path lengths planned by A* and rolling-window RRT are highly sensitive to the initial motion angle, with different angles causing significant variations in path length. In contrast, the paths generated by the proposed

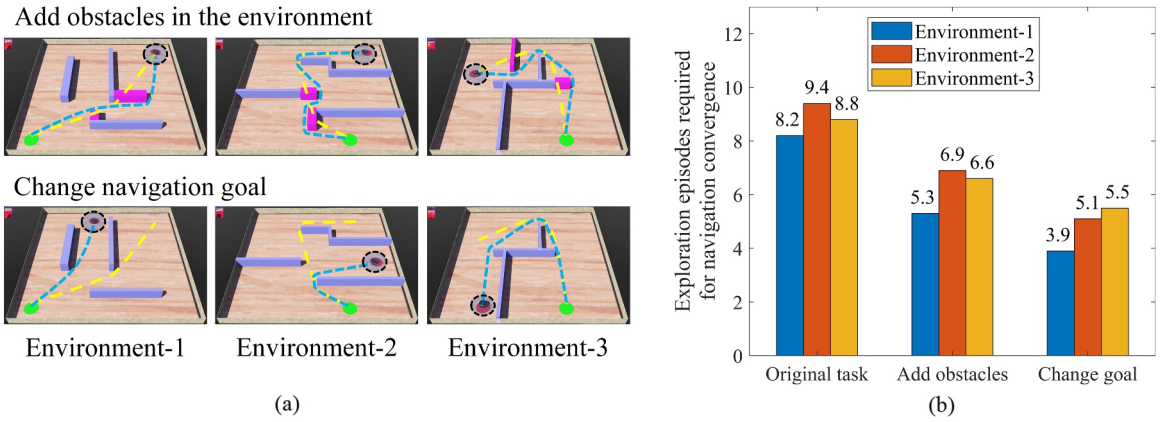


FIGURE 17 | Experimental results of reshaping navigation habits. (a) Comparison of paths before and after changes. (b) Comparison of exploration episodes.

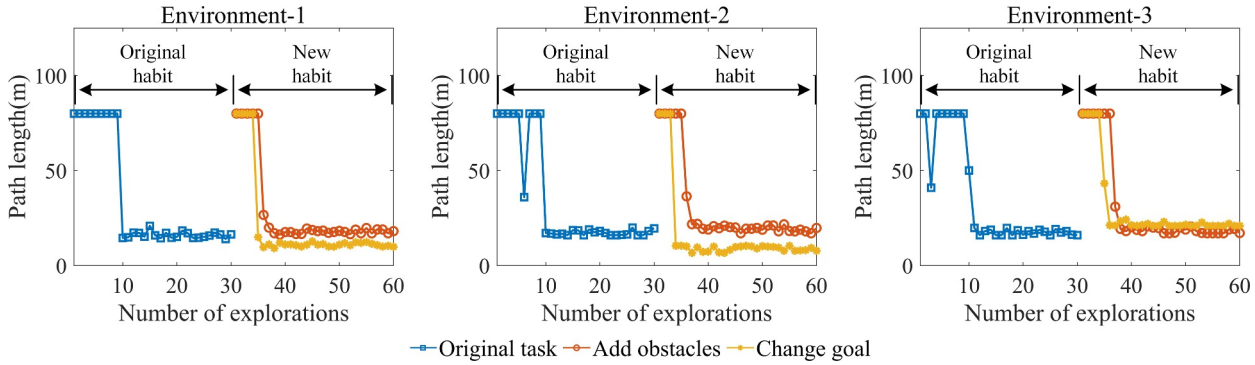


FIGURE 18 | Variation in path length during navigation habit reshaping experiments.

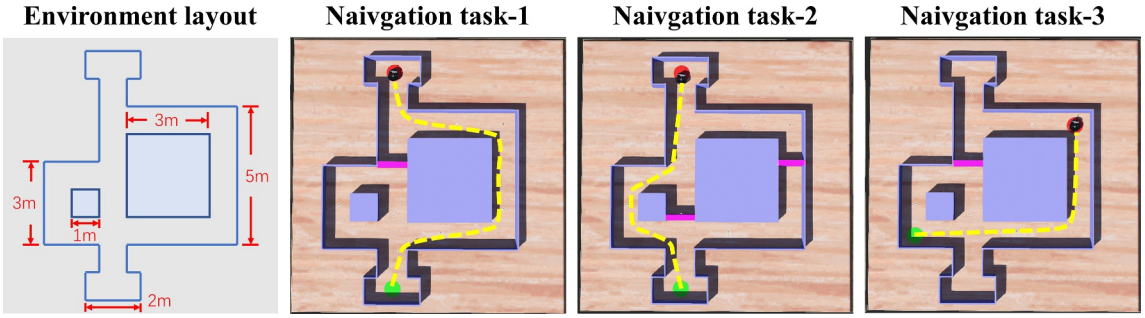


FIGURE 19 | Spatial layout and navigation path results of the maze environment.

method are virtually unaffected by changes in the initial motion angle, demonstrating strong robustness. Notably, this robustness does not stem from the inherent properties of the hippocampal-prefrontal model (Sn-Plast) but rather from the CA1 place-cell dynamic self-organisation model introduced in this chapter. This model dynamically optimises the navigation path in real time, ensuring that the optimal path adapted to the current task is produced regardless of the initial motion angle.

4.4 | Real-World Physical Experiments on a Robot Platform

To demonstrate the practical application performance of the proposed method in real physical environments, it is implemented on a mobile robot equipped with perception and computing capabilities, and physical experiments are conducted in indoor settings. Figure 21 illustrates the hardware structure and information acquisition schematic of the robot. The main body is a TurtleBot2 mobile robot. Self-motion information is obtained via encoders and IMU, landmark information is captured by an overhead camera, and obstacle information is acquired using a LiDAR. The laptop runs Ubuntu 18.04 with the following specifications: an Intel Core i7-7820HK CPU, 16 GB of RAM and an NVIDIA GeForce GTX 1070 GPU. The brain-inspired navigation system runs computations in MATLAB (R2020b) under Linux. Camera images are directly acquired and processed by MATLAB. Self-motion information and LiDAR data are first collected by the ROS system and then transmitted to MATLAB. In turn, motion commands generated by MATLAB are sent to ROS, which forwards them to the robot chassis to drive movement and steering.

The experiments are conducted in a conference room and an activity room. The maximum single-exploration distance for the robot is set to 40 m in the conference room and 60 m in the activity room. The number of visual landmarks is configured as 12 in the conference room and 10 in the activity room, with landmarks randomly distributed throughout each environment. To facilitate recognition of the navigation goal, the endpoint is designated as a red circular landmark. Figure 22a presents a snapshot of the real scene in each environment, with annotations indicating all key elements. Figure 22b shows the corresponding 2D layout of the environment, including detailed annotations of the spatial dimensions.

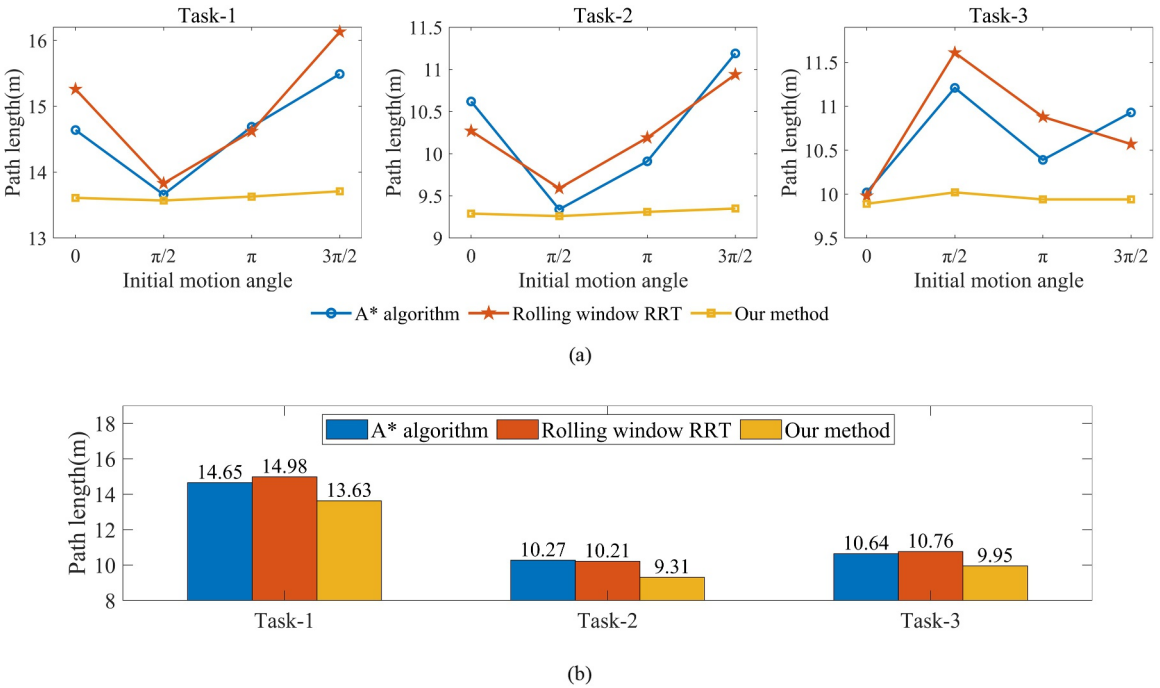


FIGURE 20 | Comparison of the navigation performance of different methods in the maze environment. (a) Navigation path length under different initial motion angles. (b) Average navigation path length statistics for each method.

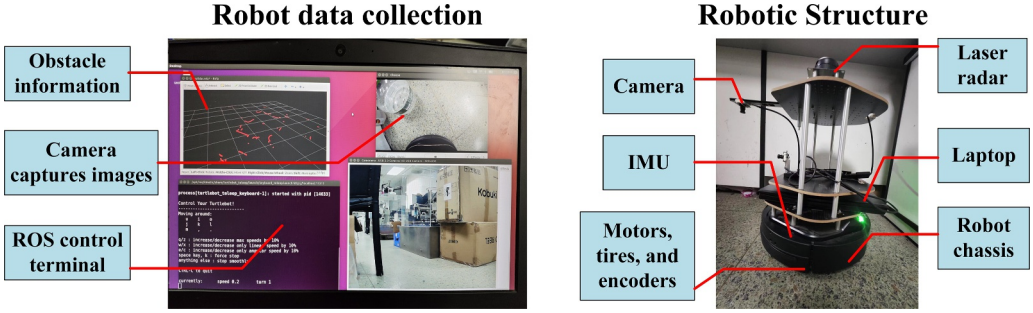


FIGURE 21 | Presentation of and introduction to the mobile robot platform.

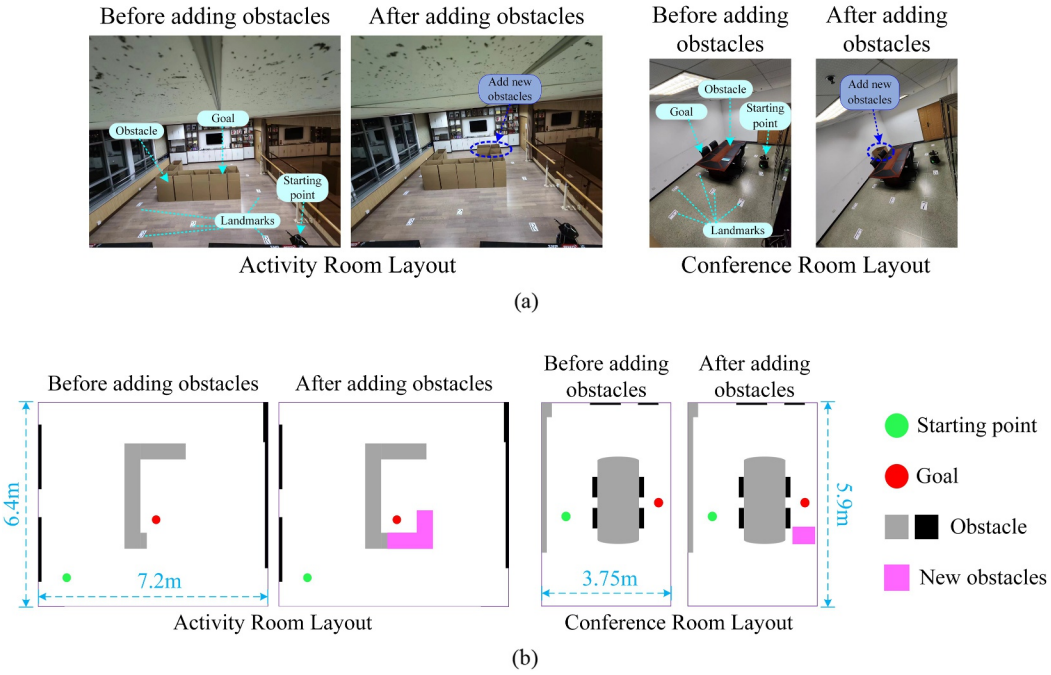


FIGURE 22 | Spatial layout of the experimental environment. (a) Actual scene snapshot of the environment with labelled key elements. (b) 2D layout display effect of the spatial environment.

The results of the mapless navigation experiments are shown in Figure 23. Figure 23a,b illustrates the navigation outcomes in the conference room and activity room, respectively, after habit formation, including the robot's navigation trajectory and real-time snapshots of the robot's states. Figure 23c depicts the changes in heading direction and linear velocity of the robot during navigation in each environment after habit formation. Figure 23d shows the variation in the length of the robot's single-exploration trajectory during the habit formation process.

As shown in Figure 23, the proposed bioinspired navigation method demonstrates excellent performance in real-world physical environments. It effectively enables the mobile robot to rapidly form stable, task-specific navigation habits through a limited number of exploration episodes, as evidenced by the clear convergence of the exploration trajectory length with increasing episodes in Figure 23d. The resulting navigation paths are relatively smooth (Figure 23a,b), allowing reliable goal-directed movement from the starting point to the designated target in complex indoor settings. Furthermore, Figure 23c reveals that, once habits are established, the robot exhibits continuous changes in heading angle and relatively controlled fluctuations in linear velocity, indicating the formation of a stable motion policy.

However, compared with conventional map-based navigation approaches, Figure 23c still exhibits more pronounced velocity jumps and brief pauses. This difference mainly stems from the fact that mapless navigation relies on real-time perception and computation: Upon reaching each new position encoded by place cells, the robot must momentarily pause to perform path integration, visual landmark recognition, position correction and action-cell decision-making. As a result, the linear velocity displays rhythmic fluctuations after the 'computation–decision–execution' cycle. This observation underscores the inherent trade-off between real-time responsiveness and computational overhead in the current bioinspired mapless navigation framework, representing a key direction for future optimisation.

Finally, to quantitatively characterise this trade-off and the associated memory cost, the single-step time and memory consumption of the robot are recorded during the real-world experiments, as summarised in Table 3. The average per-step time consumption is 115 ms, whereas the maximum reaches 683 ms. This gap mainly arises from the goal-reaching step: Once the robot reaches the goal, the PO and PF modules are triggered to optimise the discovered route and consolidate it into the SNN, which incurs a markedly higher computational cost than ordinary exploration steps. In contrast, typical exploration steps only perform path integration, landmark recognition, position correction and action decision-making and thus correspond to the much lower average time consumption. Regarding memory, the consumption remains low and highly stable, with an average of 1192 MB and a maximum of 1216 MB. The narrow gap of only 24 MB indicates that the framework maintains a bounded memory footprint throughout navigation because it neither constructs nor stores an explicit metric map and instead encodes the environment in a fixed set of spatial-cell populations and synaptic weights. This bounded and predictable memory behaviour, together with the low average per-step latency, confirms that the proposed framework is computationally lightweight and practically deployable on resource-constrained mobile platforms.

5 | Conclusion

The core innovation of this study lies in bridging previously segregated brain functions. The dual visual streams and OVC receptive fields enable real-time landmark encoding with minimal computational overhead, achieving centimetre-level localisation correction even under sensor noise. This mirrors

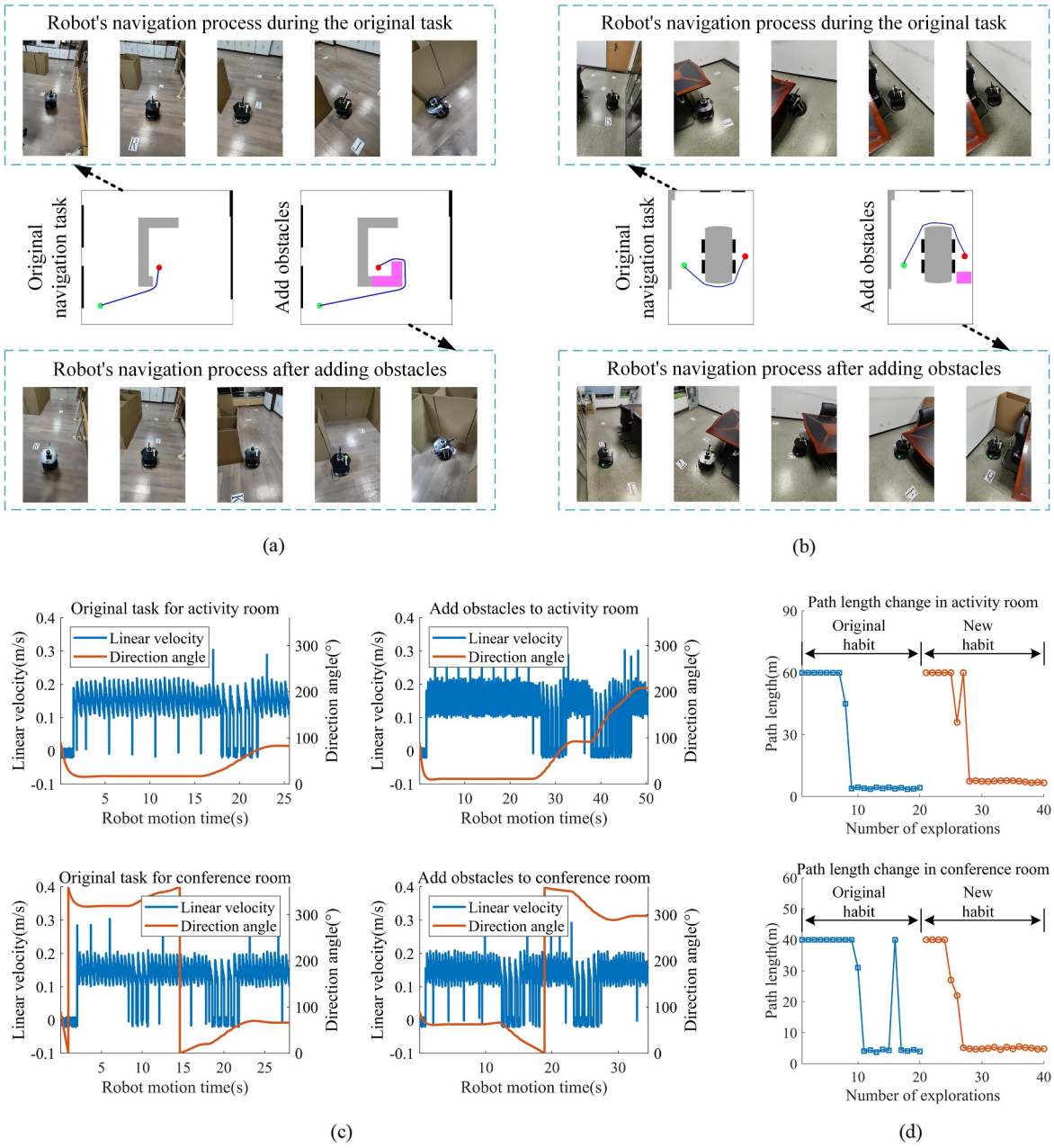


FIGURE 23 | Operational results of the robot navigation experiment. (a) Navigation results for the activity room setting. (b) Navigation results for the conference room setting. (c) Changes in the robot's velocity and heading direction during the navigation process. (d) Navigation path length changes.

TABLE 3 | Single-step time and memory consumption on the physical robot platform.

	Time consumption		Memory consumption	
Index	**Average**	**Maximum**	**Average**	**Maximum**
Value	115 ms	683 ms	1192 MB	1216 MB

rodents' reliance on peripheral visual cues for drift compensation. The CA1 self-organisation module dynamically shortens trajectories while respecting BVC-derived obstacle constraints, reproducing the iterative refinement observed during offline replay. Finally, supervised replay of optimised CA1 sequences via theoretical spike trains consolidates stable motor policies in the SNN, emulating experience-dependent plasticity. Unlike prior works that model isolated components, this closed-loop integration enables emergent habit formation without external rewards or a priori maps. This study not only validates through 2D and 3D simulation experiments that the proposed method significantly outperforms traditional RL approaches (e.g., Q-learning, SARSA, R-IAC, meta-RL and Ac-DRL) and path planning methods (e.g., A* and rolling-window RRT) in terms of convergence speed, path efficiency, localisation robustness and task adaptability but also demonstrates, via real-world experiments on the TurtleBot2 mobile robot platform in indoor physical environments, the feasibility and practical effectiveness of the presented bioinspired mapless navigation framework.

Although the proposed method represents significant progress in bioinspired mapless navigation, several limitations remain. Navigation is currently confined to predefined, fixed-size environments and cannot yet scale to arbitrarily large or unbounded spaces. Moreover, cumulative error correction relies exclusively on salient visual landmarks, without incorporating other rich environmental cues. Additionally, when the environment changes (e.g., through obstacle addition/removal or goal relocation), the system still requires local re-exploration and complete path re-optimisation, lacking the capacity for direct, incremental adaptation of existing navigation habits. To address these limitations, future work should explore biologically plausible reset mechanisms for spatial-cell activity to enable navigation in arbitrarily large environments, integrate multiple complementary exogenous sensory inputs to progressively build more robust and comprehensive environmental representations during prolonged robot–environment interaction and develop adaptive incremental learning strategies that allow localised, efficient updates to established navigation policies in response to dynamic environmental changes. Beyond these, future work can also focus on designing an architecture that flexibly switches between map-based and mapless navigation based on situational demands, in line with the multiple strategies rodents employ when solving complex navigation tasks.

In summary, this study establishes a solid foundation for incorporating rat-inspired, brain-like cognitive mechanisms into autonomous robot navigation, paving the way for more flexible, robust and scalable bioinspired spatial intelligence in real-world applications.

Funding

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62566024 and 62076014), in part by the Science and Technology Research Project of Jiangxi Provincial Department of Education (Grant No. GJJ2400402) and in part by the Early-Career Young Scientists and Technologists Project of Jiangxi Province (Grant No. 20252BEJ730116).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. B. Cai, C. Wei, and Z. Ji, "Deep Reinforcement Learning With Multiple Unrelated Rewards for AGV Mapless Navigation," *IEEE Transactions on Automation Science and Engineering* 22 (2024): 4323–4340, https://doi.org/10.1109/tase.2024.3410162.

2. Q. Zou, C. Wu, M. Cong, and D. Liu, "Brain Cognition Mechanism-Inspired Hierarchical Navigation Method for Mobile Robots," *Journal of Bionic Engineering* 21, no. 2 (2024): 852–865, https://doi.org/10.1007/s42235-023-00449-4.

3. M. Hicks, T. Lei, C. Luo, D. W. Carruth, and Z. Bi, "A Bio-Inspired Goal-Directed Cognitive Map Model for Robot Navigation and Exploration," *IEEE Transactions on Cognitive and Developmental Systems* 17, no. 5 (2025): 1125–1140, https://doi.org/10.1109/tcds.2025.3552085.

4. L. C. S. Tavares and A. B. L. Tort, "Hippocampal-Prefrontal Interactions During Spatial Decision Making," *Hippocampus* 32, no. 1 (2022): 38–54, https://doi.org/10.1002/hipo.23394.

5. M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object Vision and Spatial Vision: Two Cortical Pathways," *Trends in Neurosciences* 6, no. 10 (1983): 414–417, https://doi.org/10.1016/0166-2236(83)90190-x.

6. Ø A. Høydal, E. R. Skytøen, S. O. Andersson, M. B. Moser, and E. I. Moser, "Object-Vector Coding in the Medial Entorhinal Cortex," *Nature* 568, no. 7752 (2019): 400–404, https://doi.org/10.1038/s41586-019-1077-7.

7. A. S. Etienne, R. Maurer, V. Boulens, A. Levy, and T. Rowe, "Resetting the Path Integrator: A Basic Condition for Route-Based Navigation," *Journal of Experimental Biology* 207, no. 9 (2004): 1491–1508, https://doi.org/10.1242/jeb.00906.

8. T. Geiller, S. Sadeh, S. V. Rolotti, et al., "Local Circuit Amplification of Spatial Selectivity in the Hippocampus," *Nature* 601, no. 7891 (2022): 105–109, https://doi.org/10.1038/s41586-021-04169-9.

9. J. D. Shin and S. P. Jadhav, "Prefrontal Cortical Ripples Mediate Top-Down Suppression of Hippocampal Reactivation During Sleep Memory Consolidation," *Current Biology* 34, no. 13 (2024): 2801–2811, https://doi.org/10.1016/j.cub.2024.05.018.

10. R. M. Grieves, É. Duvelle, and P. A. Dudchenko, "A Boundary Vector Cell Model of Place Field Repetition," *Spatial Cognition and Computation* 18, no. 3 (2018): 217–256, https://doi.org/10.1080/13875868.2018.1437621.

11. K. F. Messanvi, K. Berkun, A. Perkins, and Y. Chudasama, "Parallel Pathways Provide Hippocampal Spatial Information to Prefrontal Cortex," *Journal of Neuroscience* 43, no. 1 (2023): 68–81, https://doi.org/10.1523/jneurosci.0846-22.2022.

12. J. D. Shin, W. Tang, and S. P. Jadhav, "Dynamics of Awake Hippocampal-Prefrontal Replay for Spatial Learning and Memory-Guided Decision Making," *Neuron* 104, no. 6 (2019): 1110–1125, https://doi.org/10.1016/j.neuron.2019.09.012.

13. S. Viganò and M. Piazza, "The Hippocampal-Entorhinal System Represents Nested Hierarchical Relations Between Words During Concept Learning," *Hippocampus* 31, no. 6 (2021): 557–568, https://doi.org/10.1002/hipo.23320.

14. B. J. Choi, B. Kim, S. M. Jin, et al., "Magnetic Landmark-Based Position Correction Technique for Mobile Robots With Hall Sensors," *Intelligent Service Robotics* 3, no. 2 (2010): 99–113, https://doi.org/10.1007/s11370-010-0062-7.

15. F. Shamsfakhr, B. Sadeghi Bigham, and A. Mohammadi, "Indoor Mobile Robot Localization in Dynamic and Cluttered Environments Using Artificial Landmarks," *Engineering Computations* 36, no. 2 (2019): 400–419, https://doi.org/10.1108/ec-03-2018-0151.

16. W. Chai, C. Li, and Q. Li, "Multi-Sensor Fusion-Based Indoor Single-Track Semantic Map Construction and Localization," *IEEE Sensors Journal* 23, no. 3 (2022): 2470–2480, https://doi.org/10.1109/jsen.2022.3226821.

17. J. Yin, F. Yan, Y. Liu, G. He, and Y. Zhuang, "An Overview of Simultaneous Localisation and Mapping: Towards Multi-Sensor Fusion," *International Journal of Systems Science* 55, no. 3 (2024): 550–568, https://doi.org/10.1080/00207721.2023.2282409.

18. H. Wang, G. Zhang, H. Cao, et al., "Geometry-Aware 3D Point Cloud Learning for Precise Cutting-Point Detection in Unstructured Field Environments," *Journal of Field Robotics* 42, no. 7 (2025): 3063–3076, https://doi.org/10.1002/rob.22567.

19. C. Wang, M. Yang, Q. Han, et al., "Dynamic Litchi Fruit Trajectory Prediction Using Extended Kalman Filter in Simulated Wind

Environment," *Computers and Electronics in Agriculture* 247 (2026): 111666, https://doi.org/10.1016/j.compag.2026.111666.

20. S. O. Andersson, E. I. Moser, M. B. Moser, et al., "Visual Stimulus Features That Elicit Activity in Object-Vector Cells," *Communications Biology* 4, no. 1 (2021): 1219, https://doi.org/10.1038/s42003-021-02727-5.

21. Z. Han and A. Sereno, "Identifying and Localizing Multiple Objects Using Artificial Ventral and Dorsal Cortical Visual Pathways," *Neural Computation* 35, no. 2 (2023): 249–275, https://doi.org/10.1162/neco_a_01559.

22. P. Piray and N. D. Daw, "Reconciling Flexibility and Efficiency: Medial Entorhinal Cortex Represents a Compositional Cognitive Map," *Nature Communications* 16 (2025): 7444, https://doi.org/10.1038/s41467-025-62733-7.

23. M. J. Milford and G. F. Wyeth, "Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System," *IEEE Transactions on Robotics* 24, no. 5 (2008): 1038–1053, https://doi.org/10.1109/tro.2008.2004520.

24. M. Milford, G. Wyeth, and D. Prasser, "Efficient Goal Directed Navigation Using RatSLAM," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (IEEE, 2005), 1097–1102, https://doi.org/10.1109/ROBOT.2005.1570262.

25. N. S. Y. Dumont, P. M. Furlong, J. Orchard, and C. Eliasmith, "Exploiting Semantic Information in a Spiking Neural SLAM System," *Frontiers in Neuroscience* 17 (2023): 1190515, https://doi.org/10.3389/fnins.2023.1190515.

26. A. Safron, O. Çatal, and T. Verbelen, "Generalized Simultaneous Localization and Mapping (G-SLAM) as Unification Framework for Natural and Artificial Intelligences: Towards Reverse Engineering the Hippocampal/Entorhinal System and Principles of High-Level Cognition," *Frontiers in Systems Neuroscience* 16 (2022): 787659, https://doi.org/10.3389/fnsys.2022.787659.

27. D. George, R. V. Rikhye, N. Gothoskar, J. S. Guntupalli, A. Dedieu, and M. Lázaro-Gredilla, "Clone-Structured Graph Representations Enable Flexible Learning and Vicarious Evaluation of Cognitive Maps," *Nature Communications* 12, no. 1 (2021): 2392, https://doi.org/10.1038/s41467-021-22559-5.

28. P. Y. Oudeyer and F. Kaplan, "Intelligent Adaptive Curiosity: A Source of Self-Development," in *Procedings of the International Workshop on Epigenetic Robotics* (Lund University Cognitive Studies, 2004), 127–130.

29. A. Baranes and P. Y. Oudeyer, "R-IAC: Robust Intrinsically Motivated Exploration and Active Learning," *IEEE Transactions on Autonomous Mental Development* 1, no. 3 (2009): 155–169, https://doi.org/10.1109/TAMD.2009.2037513.

30. X. Zhang, X. Ruan, Y. Xiao, et al., "Mobile Robot Autonomous Path Planning Method Based on Intrinsic Motivation Mechanism," *Control and Decision* 33, no. 9 (2018): 1605–1611, 10.13195/j.kzyjc.2017.0620.

31. S. Dasgupta, F. Wörgötter, and P. Manoonpong, "Neuromodulatory Adaptive Combination of Correlation-Based Learning in Cerebellum and Reward-Based Learning in Basal Ganglia for Goal-Directed Behavior Control," *Frontiers in Neural Circuits* 8 (2014): 126, https://doi.org/10.3389/fncir.2014.00126.

32. A. Banino, C. Barry, B. Uria, et al., "Vector-Based Navigation Using Grid-Like Representations in Artificial Agents," *Nature* 557, no. 7705 (2018): 429–433, https://doi.org/10.1038/s41586-018-0102-6.

33. X. Ruan, P. Li, X. Zhu, and P. Liu, "A Target-Driven Visual Navigation Method Based on Intrinsic Motivation Exploration and Space Topological Cognition," *Scientific Reports* 12, no. 1 (2022): 3462, https://doi.org/10.1038/s41598-022-07264-7.

34. W. Li, M. Yue, J. Shangguan, and Y. Jin, "Navigation of Mobile Robots Based on Deep Reinforcement Learning: Reward Function Optimization and Knowledge Transfer," *International Journal of Control, Automation and Systems* 21, no. 2 (2023): 563–574, https://doi.org/10.1007/s12555-021-0642-7.

35. Y. Gao, F. Lin, B. Cai, et al., "Mapless Navigation via Hierarchical Reinforcement Learning With Memory-Decaying Novelty," *Robotics and Autonomous Systems* 182 (2024): 104815, https://doi.org/10.1016/j.robot.2024.104815.

36. T. Kulvicius, M. Tamosiunaite, J. Ainge, P. Dudchenko, and F. Wörgötter, "Odor Supported Place Cell Model and Goal Navigation in Rodents," *Journal of Computational Neuroscience* 25, no. 3 (2008): 481–500, https://doi.org/10.1007/s10827-008-0090-x.

37. N. Frémaux, H. Sprekeler, and W. Gerstner, "Reinforcement Learning Using a Continuous Time Actor-Critic Framework With Spiking Neurons," *PLoS Computational Biology* 9, no. 4 (2013): e1003024, https://doi.org/10.1371/journal.pcbi.1003024.

38. Z. Brzosko, S. Zannone, W. Schultz, C. Clopath, and 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.

39. S. Zannone, Z. Brzosko, O. Paulsen, and C. Clopath, "Acetylcholine-Modulated Plasticity in Reward-Driven Navigation: A Computational Study," *Scientific Reports* 8, no. 1 (2018): 9486, https://doi.org/10.1038/s41598-018-27393-2.

40. G. W. Y. Ang, C. S. Tang, Y. A. Hay, S. Zannone, O. Paulsen, and C. Clopath, "The Functional Role of Sequentially Neuromodulated Synaptic Plasticity in Behavioural Learning," *PLoS Computational Biology* 17, no. 6 (2021): e1009017, https://doi.org/10.1371/journal.pcbi.1009017.

41. H. Wei, L. Yin, S. Hou, Y. Yu, and Q. Wang, "Dynamical Modeling of Hippocampal-Basal Ganglia Interactions for Spatial Navigation," *Chaos, Solitons & Fractals* 200 (2025): 117014, https://doi.org/10.1016/j.chaos.2025.117014.

42. Y. Liao and N. Yu, "A Real-Time Cognitive Map Construction Method Based on the Entorhinal-Hippocampal Working Mechanism of the Rat's Brain," *Cognitive Computation and Systems* 6, no. 4 (2024): 49–64, https://doi.org/10.1049/ccs2.12101.

43. Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, no. 11 (2000): 1330–1334, https://doi.org/10.1109/34.888718.

44. T. N. Do and N. K. Pham, "Handwritten Digit Recognition Using GIST Descriptors and Random Oblique Decision Trees," in *The National Foundation for Science and Technology Development (NAFOSTED) Conference on Information and Computer Science* (Springer International Publishing, 2014), 1–15, https://doi.org/10.1007/978-3-319-14633-1_1.

45. S. Adam, L. Busoniu, and R. Babuska, "Experience Replay for Real-Time Reinforcement Learning Control," *IEEE Transactions on Systems, Man, and Cybernetics* 42, no. 2 (2011): 201–212, https://doi.org/10.1109/tsmcc.2011.2106494.

46. X. Lv and D. Wang, "A Bionic Memory Replay Model Based on Hippocampal-Prefrontal Collaboration: Design, Mechanism, and Robot Navigation Verification," *IEEE Transactions on Automation Science and Engineering* 23 (2026): 2852–2863, https://doi.org/10.1109/tase.2026.3657422.

47. J. Xue, J. Chen, and S. Zhang, "Action-Curiosity-Based Deep Reinforcement Learning Algorithm for Path Planning in a Nondeterministic Environment," *Intelligent Computing* 4 (2025): 0140, https://doi.org/10.34133/icomputing.0140.

48. F. Duchoň, A. Babinec, M. Kajan, et al., "Path Planning With Modified a Star Algorithm for a Mobile Robot," *Procedia Engineering* 96 (2014): 59–69, https://doi.org/10.1016/j.proeng.2014.12.098.

49. L. Kang, C. X. Zhao, and J. H. Guo, "Improved Path Planning Based on Rapidly-Exploring Random Tree for Mobile Robot in Unknown Environment," *Pattern Recognition and Artificial Intelligence* 22, no. 3 (2009): 337–343, https://doi.org/10.16451/j.cnki.issn1003-6059.2009.03.021.