Route-Centric Ant-Inspired Memories Enable Panoramic Route-Following in a Car-like Robot

Gabriel G. Gattaux[1], Antoine Wystrach[2*], Julien R. Serres[1,3], Franck Ruffier[1*]

[1]Aix Marseille Univ, CNRS, ISM, Marseille, France.
[2]Univ Toulouse, CRCA, CBI, UMR CNRS-UPS 5169, Toulouse, France.
[3]Institut Universitaire de France, IUF, Paris, France.

*Corresponding author(s). E-mail(s): antoine.wystrach@univ-tlse3.fr; franck.ruffier@cnrs.fr;

Abstract

Solitary foraging ants excel at route following using minimal neural resources, Robots don't. Recent biological studies proposed lateralized, nest-centric memories to explain ants' direct visual homing but did not address how ants follow curved visual routes away from their nest. We present a biologically inspired neuromorphic model for one-shot panoramic route learning and continuous route following, implemented on a compact car-like robot, Antcar. We demonstrate that route-centric lateralized memories, inspired by the insect mushroom body, enable Antcar to achieve bi-directional route-following, with motivation-driven recognition of route extremities and familiarity-based velocity control. With rigorous Lyapunov-based stability analysis and an empirical memory scalability evaluation, the model was tested over 1.6 km across 113 challenging real-world trials. The system achieves less than 25 cm median lateral error using minimal resources (800-pixel input, 300 MB RAM, 500 mW power, and 18.75 kB memory per 50 m route), offering insights into insect cognition and advancing autonomous robotics under strict resource constraints.

Introduction

Insect navigation has long intrigued researchers across various fields, from biology to robotics, driving the development of cutting-edge technologies for autonomous mobile robots [1–3]. Autonomous navigation remains a demanding and interdisciplinary challenge with applications ranging from space exploration to last miles delivery [4, 5], especially in scenarios where robots cannot rely on satellite systems [6]. Simultaneously, robots serve as valuable tools for studying insects navigation and brain structure, advancing neuromorphic engineering [7–11].

In Robotics, visual teach-and-repeat methods combined with dead-reckoning techniques have gained in popularity [12–15]. However, experienced solitary foraging ants navigate along familiar routes using only visual memories, without relying on dead reckoning (called path integration in insect literature) [16–18]. This behavior has inspired various robotic models, although current implementations are generally limited to short-range experiments of about ten meters, with modest computational efficiency, precision, and accuracy [19–24]. While ant-inspired models achieve results comparable to conventional computer vision approaches [13, 25], they struggle in dynamic environments where computational efficiency must be balanced with resource use.

Fig. 1 Biological inspiration for robotic navigation. An ant in the foreground symbolizes nature's efficient navigational strategies, while the Antcar robot in the background integrates these principles into a neuromorphic system. The blurred image captures only the large masses of the environment, similar to the low-pass spatial filter in the ant's visual system, which retains these large features even when objects obstruct the view between the robot and the building. ©Tifenn Ripoll - VOST Collectif / Institut Carnot STAR.

These challenges are partly due to early navigation models that emphasized hymenopteran behavior rather than underlying brain processes. Early models, referred to as perfect memory models, stored periodic snapshots at specific waypoints [26, 27]. Then, during autonomous route following (or exploitation), forced scanning movements compared acquired views to an image bank, using rotational image differences to establish the most familiar image and desired heading, a

process known as the insect-based visual compass [28–33]. However, these approaches has revealed two main limitations when applied in robotics.

The first limitation involves the cumulative storage of snapshots, which significantly increases memory and computational demands as the route lengthens, making it unsuitable for long-distance navigation. This issue was partially addressed by the Infomax neural network [34], which enables efficient encoding of increasing numbers of images without a corresponding rise in memory load [20, 32, 35]. However, Infomax requires substantial adjustments to synaptic weights for each input through a non-local learning mechanism, limiting its biological plausibility.

In parallel, research on the Mushroom Body (MB), a key part of the insect brain, has highlighted its essential role in olfactory and visual learning [36, 37]. In the MB, learning occurs through synaptic depression between thousands of Kenyon Cells (KCs) – intrinsic neurons that sparsely encode sensory input – and a few Mushroom Body Output Neurons (MBONs), which modulate behavioral responses based on learned associations. These processed signals are then transmitted to downstream neural circuits, influencing decision-making [38]. The first MB model simulating visual route following used a Spiking Neural Network with 20,000 KCs and one MBON to compute familiarity [39]. Despite this advancement, a second limitation remains: navigation requires forced, systematic scanning, which slows robotic movement due to non-continuous command updates [21]. Also, this limitation does not reflect natural ant behavior, where scanning occurs only occasionally [40–42].

To address the second limitation, an early robotic implementation combined a *klinokinesis* model with perfect memory, enhancing short-distance route-following by replacing cumbersome scanning with alternating, ballistic left and right turns where familiarity adjusted turn amplitude [19] (later also observed in ants [43]).

To move beyond random, undirected kinesis, an early visual homing strategy was proposed that simulates directed movement toward a goal (e.g., a nest). In this model, tested in a particle-based simulation, the firing activity of KCs—which encode visual input—is categorized based on the path integration vector into two pairs of lateralized MBONs. Specifically, views acquired when the nest lies to the left are assigned to the left MBON, and vice versa [44, 45]. This approach mirrors how insects, through continuous lateral body oscillations, sample multiple directions around their nest position [43, 46].

In this paper, we focus on route following, which is considered distinct from visual homing in the ant navigation framework. While visual homing is goal-directed and attractor-based, route following relies on on-route visual memories [47]. Some recent robotic models have attempted to extend lateralization to route following by splitting the field of view into left and right memories. However, these methods have shown limited success in real-world conditions [23, 48], and they diverge from biological evidence: in ants, the entire panoramic field is processed binocularly by the MB [49]. Our approach therefore uses the full panoramic view and categorizes visual input with right and left MBON in a route-centric manner, i.e. with respect to a local route orientation, without splitting the field-of-view.

Building on our earlier one-MBON model for route learning in an indoor robot [21, 22], and the lateralization principle for nest-centric homing in simulation [45], we present a fully-embodied, biologically realistic, and route-centric lateralized MB architecture implemented and tested on a real-world robot. This work introduces several contributions that significantly advance the field:

(i) Route-centric lateralized learning: We replace the nest-centered categorization with a local, route-aligned reference frame, enabling two lateralized MBONs (right and left) to memorize panoramas as seen on the right or left of the route. This supports curved and goal-directed navigation without requiring a global home vector. (ii) Bi-directional navigation with start/end recognition: Two additional MBONs allow the robot to recognize route extremities, enabling stop and back-and-forth route following, a behavior observed in ants [50–53]. (iii) Familiarity-based velocity control: The robot adapts its speed based on visual familiarity, accelerating on known routes and decelerating in novel environments, mirroring ant behavior [40]. (iv) Analytical stability via oscillatory learning: We show that oscillatory heading movements during learning lead to stable route following, supported by a Lyapunov-based proof of convergence in the local route frame. (v) Empirical memory scalability analysis: We propose a metric to quantify memory capacity in relation to network hyperparameters, providing insights into the scalability of Mushroom Body-based architectures for longer and more complex routes. (vi) Extensive real-world validation: Embedded in the compact Antcar robot (Fig. 1, 2a), our model was tested across 113 autonomous trajectories covering 1.6 km in both indoor and outdoor settings with different environmental conditions and challenges. The system achieved median lateral and angular errors of 24 cm and 4°, with processing rates of 16 Hz during exploitation and 38 Hz during learning — all using a single 32×32-pixel panoramic camera and minimal computational resources: 300 megabytes of RAM, 500 mW power consumption, and just 18.75 kilobytes of memory. Overall, this paper delivers a biologically grounded and resource-efficient implementation of the proposed panoramic, route-centric and lateralized visual route-following on a physical robot. It integrates neural plausibility, theoretical robustness, and practical feasibility, bridging the gap between insect-inspired models and real-world autonomous robotics.

Results

Our proposed MB model emulates ant visual processing by encoding panoramic images as low resolution neural representations (800-pixel input), enabling efficient learning and route recognition with minimal computational demands (see Methods for details, Fig. 2b). The model operates in two main phases: learning (Fig. 2c) and exploitation (Fig. 2d). During the learning phase, our self-supervised model encodes the route using two

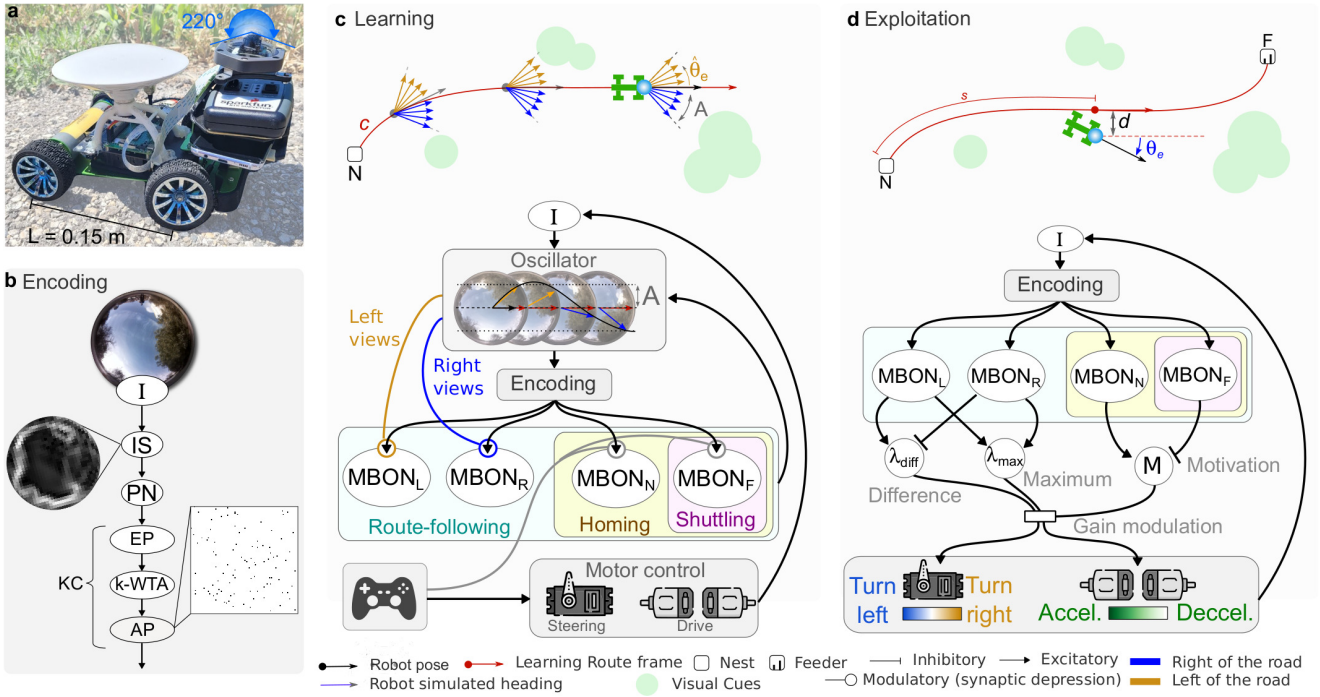


Fig. 2 Overview of the route-centric lateralized memories model implemented in the Antcar robot. This figure illustrates the processing pipeline from image encoding to navigation control, spanning both the learning and exploitation phases. **a** The Antcar robot: a compact car-like platform equipped with a panoramic camera and a GPS-RTK (Global Positioning System - Real-Time Kinematic) system for ground truth data. **b** Image encoding: this process mimics the ant's visual system. Panoramic images (**I**) are captured, blurred, sub-sampled, and edge-filtered to produce a low-resolution 32 × 32 pixel panorama (**IS**). The resulting image is then transformed into Projection Neurons (**PN**), expanded into Excitatory Post-Synaptic Projections (**EP**), and reduced into Action Potentials (**AP**) via a κ-WTA function, forming the sparse Kenyon Cell (KC) representation. **c** Learning phase: the robot follows a route (C) from a start point (N). An in silico scan (simulated image rotation) generates images with a defined angular error ($\hat{\theta}_e$), used to continuously categorize views at 38 Hz in a route-centric manner—i.e., as facing to the right or left of the local route orientation. These continuous categorizations drive the update of synaptic weights in the corresponding Mushroom Body Output Neurons (MBONs), allowing visual inputs to be associated with specific route memories in a self-supervised fashion. Joystick inputs are used to define learning boundaries at the start and end of the route, modulating plasticity. **d** Exploitation phase: the robot seeks to minimize lateral (d) and angular (θ_e) errors with respect to the learned route. The encoded image continuously activates MBONs at 16Hz based on previously learned synaptic weights, enabling the robot to infer route orientation and adjust its steering and speed accordingly. MBON familiarity indices (λ) operate in an opponent process: differences in familiarity between left and right MBONs guide steering, while the overall familiarity magnitude modulates speed. Special MBONs associated with route extremities affect motivational states, enabling route polarity correction or halting movement.

route-centric MBONs, with a in silico scanning, and stores place-specific memories for the nest and feeder as route extremities (see Methods, Fig. 2c). In the exploitation phase, the robot processes each view through all memory pathways, yielding four familiarity values (left, right, nest and feeder MBON activities). The lateralized difference of route familiarities (λ_{diff}) directs steering, while the maximum familiarity value modulates forward speed. Additionally, a motivational control modulates motor gain, allowing the robot to stop or reverse based on a familiarity thresholds set by place-specific MBONs (see Methods, Fig. 2d).

This study begins with an offline analysis of the proposed self-supervised, route-centric MB model using two route MBONs to assess stability (Fig. 3), followed by experimental route-following tasks in challenging indoor and outdoor environments (Figs. 4, 5 and 6). Next, a homing task is described, in which the robot follows a long outdoor route in reverse toward the starting area, designated as the nest (N), and stops nearby, utilizing three MBONs (Fig. 7). Finally, a shuttling task is introduced, where the robot, after a single learning trial with two route MBONs and two extremities MBONs for the nest and feeder, autonomously shuttles *to and fro* between these two locations, driving both forward and backward (Fig. 7).

Self-supervised route-centric lateralized memories model

The continuous in-silico rotation of the panoramic image, along with the route-centric hypothesis and the robot's assumption that each captured image corresponds to the route direction, drives a self-supervised route-learning mechanisms. We first evaluated the self-supervised model for route learning (using only two MBONs) with a dataset of indoor and outdoor parallel routes (Figs. 3e,h). Results demonstrated that, with a controlled oscillation amplitude during learning, the model accurately estimated its heading error based on the differential familiarity λ_{diff}, handling angular deviations up to 135° indoors and 90° outdoors (Fig. 3c,f,i). Furthermore, the maximum familiarity index $\lambda_{\max}$, used as feedback for speed control, increased proportionally with heading error, enabling the robot to slow down when misaligned with the route. This behavior was consistent even when the robot was moved laterally off-route. Outdoors, these gradients were steeper (Fig.

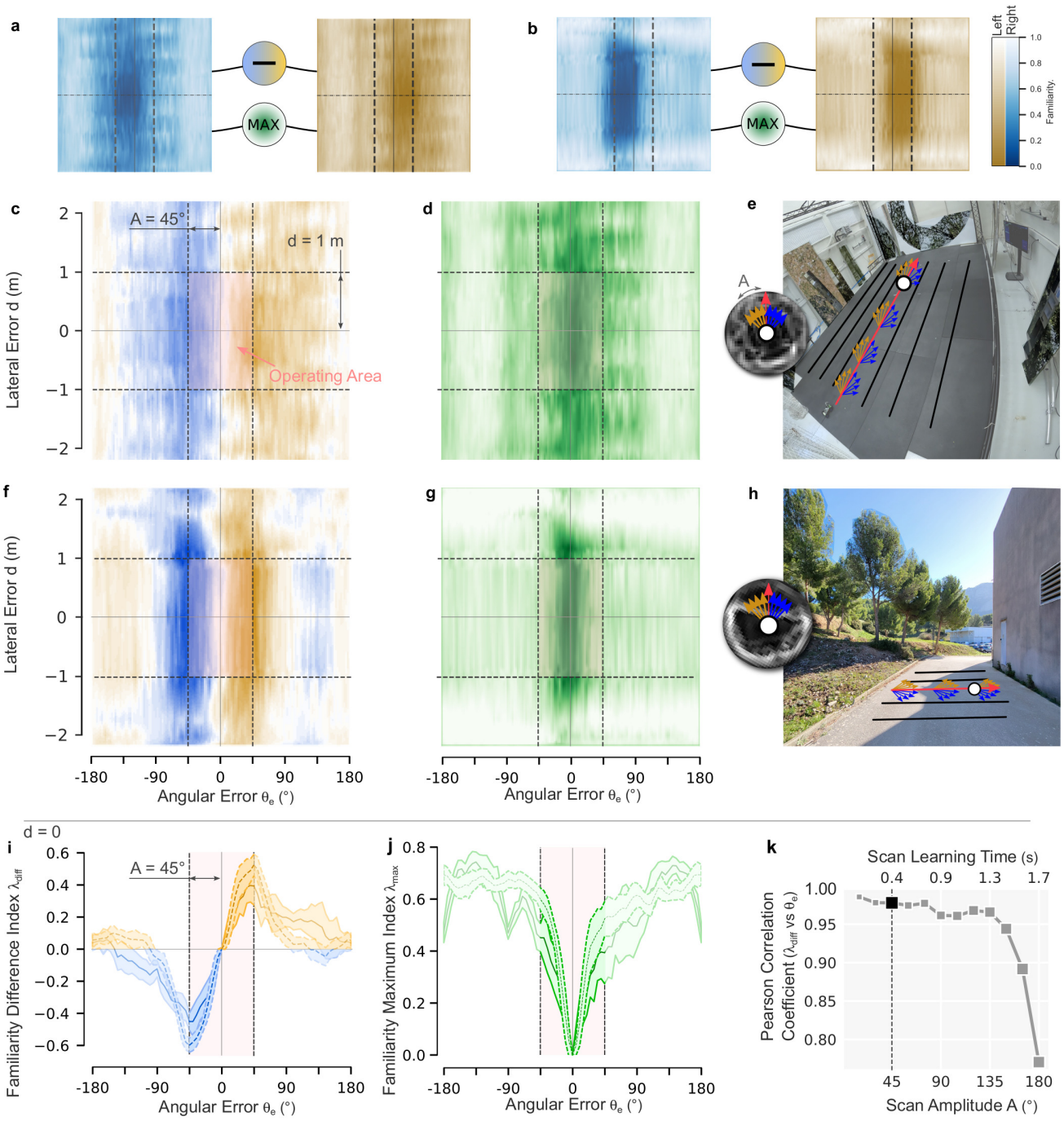


Fig. 3 Offline familiarity mapping for learning of indoor and outdoor routes. This figure illustrates the familiarities differentiation and maximum value of route Mushroom Body Output Neurons (MBONs) during offline analysis of panoramic images and positional data from indoor (Mediterranean Flight Arena) and outdoor (Luminy Campus, Marseille, France) environments. The mapping was performed using an oscillation amplitude (A) of 45°. **a,b** Raw familiarities from the left and right MBONs. **c,f** Familiarity difference index (λ_{diff}) mapped in the route's frame of reference, showing variations with both lateral (d) and angular (θ_e) errors. The defined operating area is highlighted in pink. **d,g** familiarity maximum index ($\lambda_{\max}$) mapped similarly. **e** Overview of the indoor environment with the learned route highlighted in red. **h** Overview of the outdoor environment **i,j** Cross-sectional view of λ_{diff} and $\lambda_{\max}$, respectively, plotted against θ_e when d is zero. Indoor conditions are represented by a solid line, and outdoor conditions by a dotted line. **k** Pearson correlation coefficient illustrating the linear relationship between λ_{diff} and θ_e as a function of oscillation amplitude A. This plot also indicates the learning time required for a single oscillation cycle per image captured by the robot.

3c,d,f and g), indicating a higher visual contrast with larger landmarks.

The model's ability to identify heading error accurately across training oscillation amplitudes up to 135° (Fig. 3k, see also Supplementary note 1 and Fig. S1) suggests that this parameter may not require further tuning below this threshold. However, larger oscillation amplitudes increased computation time, especially on the Raspberry Pi platform (0.4s for ±45°, Fig. 3k). Notably, the familiarity difference index (Fig. 3i) closely matched the spatial derivative of the maximum familiarity index, corresponding to the catchment area and turn rate amplitude observed in ants (Fig. 3j, Supplementary note 1, 2, Fig. S1 and S2 [44]).

This analysis helped establish the operational limits of our MB model, maintaining stable behavior within a lateral error (d) of 2 meters and an angular error (θ_e) within the learning oscillation amplitude, set here at

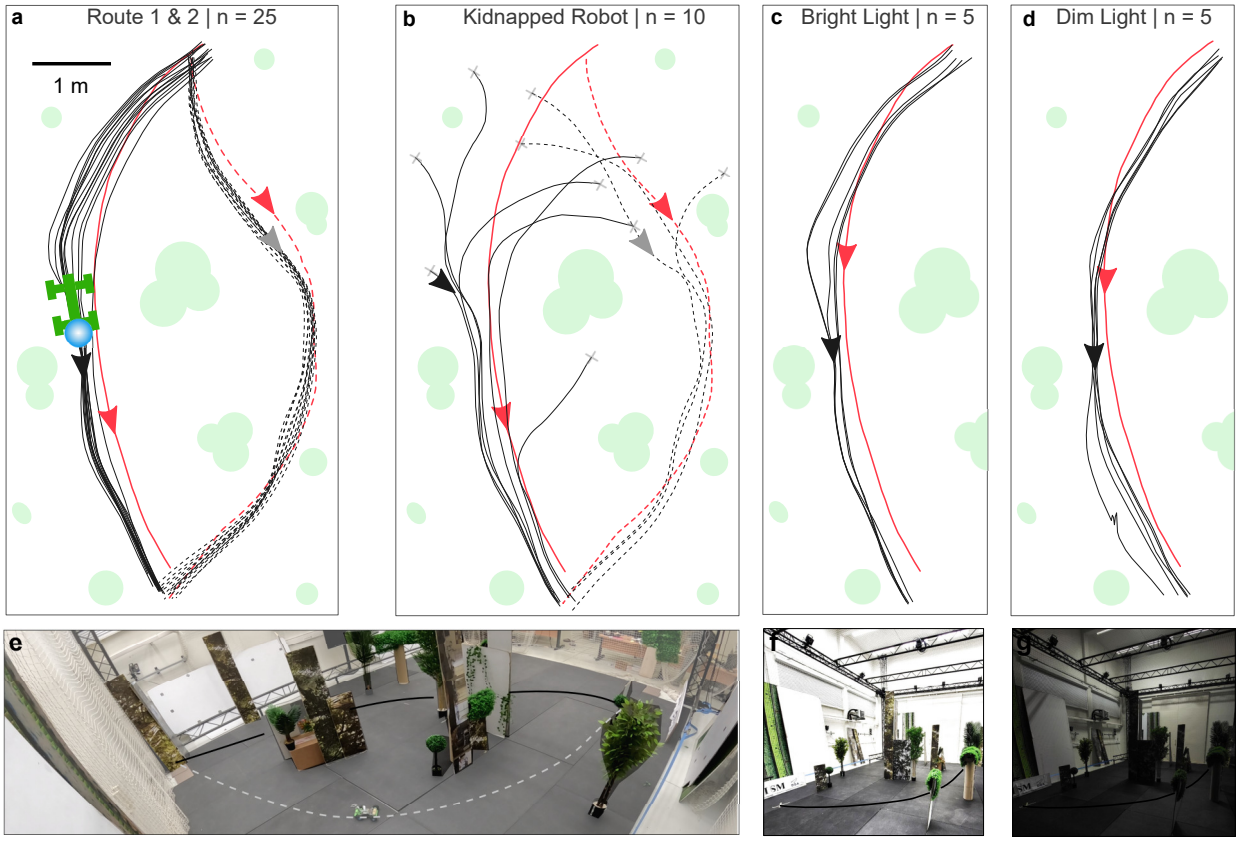


Fig. 4 Indoor route following experiments: Effects of steering direction and lighting variation with artificial visual cues. Detailed environmental configurations and familiarity data are shown in Supplementary Fig. S4 and the accompanying video. **a** Autonomous route following experiments for two distinct learned routes (route 1: solid black line, route 2: dashed black line), each approximately 8 m long and represented independently by separate pairs of route MBONs. Learned trajectories are indicated in red. **b** Kidnapped robot tests performed independently for each learned route, assessing robustness to positional uncertainty. **c,d** Route following tests performed under bright **(c)** and dim **(d)** lighting conditions, while routes were learned under standard office lighting. **e-g** Overview images of the indoor experimental environment.

45°. For asymptotic stability (i.e., the system's ability to return to equilibrium), we assumed a proportional relationship between λ_{diff} and θ_{e}, supported by the Pearson correlation coefficient being close to 1 (Fig. 3k) and expressed as $K_{\text{diff}} \cdot \lambda_{\text{diff}} = -\theta_{\text{e}}$, where K_{diff} is a tuned negative gain. Integrating this relationship into the robot's motion equations, we applied a Lyapunov function for stability analysis. Results confirmed that the system converged to equilibrium points at $d^e = 0$ and $\theta_{\text{e}}^e = 0$, effectively correcting small deviations and enabling the robot to remain aligned with the learned route. The full derivation of these equations and Lyapunov stability proof are provided in the Methods (section 6) and Supplementary note 3,4 and Fig. S3.

Route-following: robustness to visual changes

The proposed self-supervised approach for route learning was validated through a series of indoor and outdoor route-following tasks in fully autonomous mode, with only two MBONs. After a first outbound route with online learning, where images were captured continuously to update synaptic weights in real-time, the robot demonstrated robust route-following in various configurations (Figs. 4, 5 and 6). First, the Antcar robot successfully navigated two routes in a cluttered indoor environment of approximately 8 meters (median lateral error ±median absolute deviation (MAD) = 0.21 ± 0.09 m, angular error ±MAD = 3.4 ± 6.2°, Fig. 4a,e and Fig. 8a). Moreover, the robot showed resilience in a kidnapped robot scenario, realigning with the learned route after being displaced from 1 to 2m away and oriented from 0 to 50 degree, (lateral error ±MAD = 0.26 ± 0.14 m, angular error ±MAD = 6.45 ± 4.19°, Fig. 4b, and Fig. 8a). Only one predictable crash occurred when the robot exceeded theoretical angular limits (see Supplementary Fig. S5).

Further tests assessed the robot's adaptability to bright and dim light conditions (Figs. 4c,f and Figs. 4d,g). Despite a single learning trial under standard lighting (815 Lux), the robot accurately followed its route in bright (1,340 Lux) and dim (81 Lux) lighting, with similar lateral and angular errors across tests (Fig. 8). This indicates that the route-centric MB-based control system is robust to significant changes in illumination.

In indoor dynamic conditions with pedestrians and camera occlusions (Figs. 5a,b), the robot maintained reliable route-following performance. When encountering pedestrians, the lateral error was ± MAD = 0.27 ± 0.15 m and the angular error was ± MAD = 4 ± 2.8° (Figs. 5a,e and Fig. 8a). Similarly, in the presence of dynamic occlusions, the robot achieved a lateral error of ± MAD = 0.22 ± 0.13 m and an angular error of ± MAD = 4.7 ± 3.3° (Figs. 5b,f and Fig. 8a). The presence of pedestrians and occlusions resulted in a reduction of maximum familiarity, leading to slower speeds and increased oscillatory motion—approximately 15%

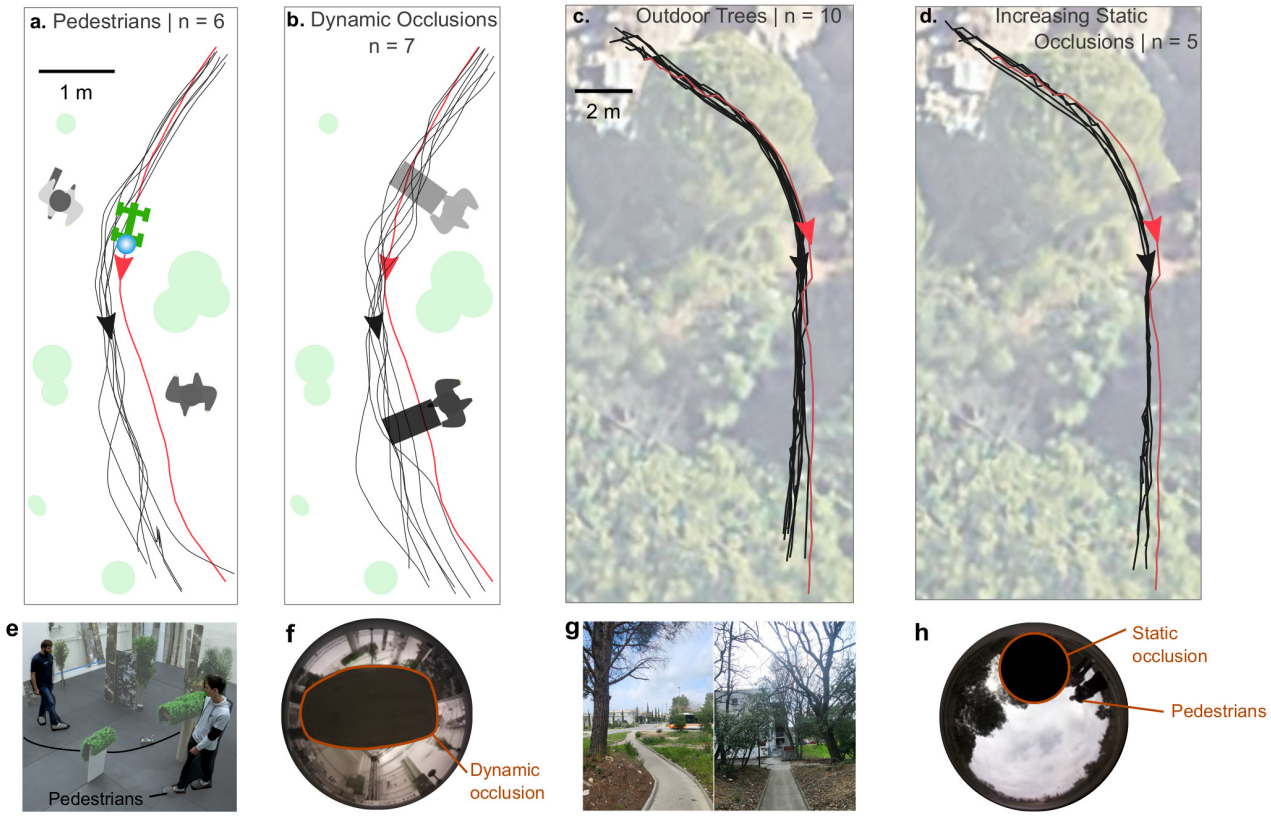


Fig. 5 Robust visual route following in dynamic environments. Antcar robot navigating indoor and outdoor routes under challenging conditions after a single learning phase. **a-b** Indoor experiments along an 8 m route marked with artificial visual cues, featuring walking pedestrians **(a)** and large dynamical occlusions **(b)**. **c-d** Outdoor experiments along a narrow (1.5 m) forest-like footpath of 21 m length, without **(c)** and with **(d)** static occlusions. **e, g** Third-person views of indoor and outdoor experimental environments. **f, h** Robot's first-person views illustrating dynamic and static occlusions in the frontal field of view.

slower than in experiments without these disturbances (Figs. 5a,b, Supplementary Figs S5 and Supplementary Movie 1). Across all seven autonomous routes with dynamic occlusions, occlusions occupied on average 11% of the image area, with peaks up to 56% (see Methods and Supplementary Fig S4 for occlusion detection using YOLOv11 [54]).

Outdoor experiments demonstrated the model's robustness in diverse settings—from a narrow 1.5 m pathway in a tree-like environment to a wider 5 m semi-urban route—while maintaining stable performance under static occlusions and altered conditions. On a semi-cloudy day, a 20 m route was learned and accurately recapitulated with low errors (lateral error ±MAD = 0.3 ± 0.12 m; angular error ±MAD = 4 ± 2.3°; Figs. 5c and 8a). Then, a static occlusion was introduced, consisting in a fixed black spot in the frontal field of view, covering 4 to 15% of the entire visual field (and 8 to 30% of the front view), the performance remained robust (lateral error ±MAD = 0.38 ± 0.10 m; angular error ±MAD = 5.6 ± 3°; Figs. 5d and 8a).

Over a 55 m route under altered conditions, the robot first navigated a sunny day with low errors (lateral error ±MAD = 0.39 ± 0.13 m; angular error ±MAD = 5.8 ± 2.8°; Figs. 6a and 8a). The following day, at the same hour, with parked cars removed and a higher speed gain (cruising at 1.5 m/s versus 1 m/s), errors increased slightly (lateral error ±MAD = 1.3 ± 0.5 m; angular error ±MAD = 6.2 ± 3.2°; Figs. 6e and 8a) yet remained within acceptable limits. During exploitation, the difference and maximum index values were marginally higher on the second day (Figs. 6g and 6h) than during the first day (Figs. 6c and 6d), likely due to the increased lateral error. These results underscore the system's resilience and adaptability under challenging conditions (See environment's picture taken at similar place from day one Fig.6b to day two Fig. 6f).

Homing: homeward route and stop

Building on the validated route-following strategy, further tests refined the robot's behavior, focusing on ant-like homing. Homing, by definition, includegraphicsis the ability to return to a specific location after displacement. To test this, we evaluated the robot's ability to follow a 50 m outdoor route in reverse, stopping at a designated nest area (point N in Fig. 7a). During learning, a 180° shift in the visual oscillation pattern simulated the "turn back and look" behavior observed in ants and led to homeward route following.

The robot successfully followed the 50 m route in reverse under cloudy outdoor conditions (lateral error ±MAD = 0.9 ±0.5 m, angular error ±MAD = 6.3 ±4.2°, Fig. 6a and Fig. 8a). Although maximum familiarity was higher than in previous outdoor experiments (see Supplementary note 6, Fig. S5 and Table S1), overall accuracy remained stable and emerging oscillatory movements was demonstrated (see Supplementary Movie 1).

To enable autonomous stopping at the nest, a place-specific MBON was used to learn 'nest-views' at the starting point of the route. Subsequent 'recognition' in

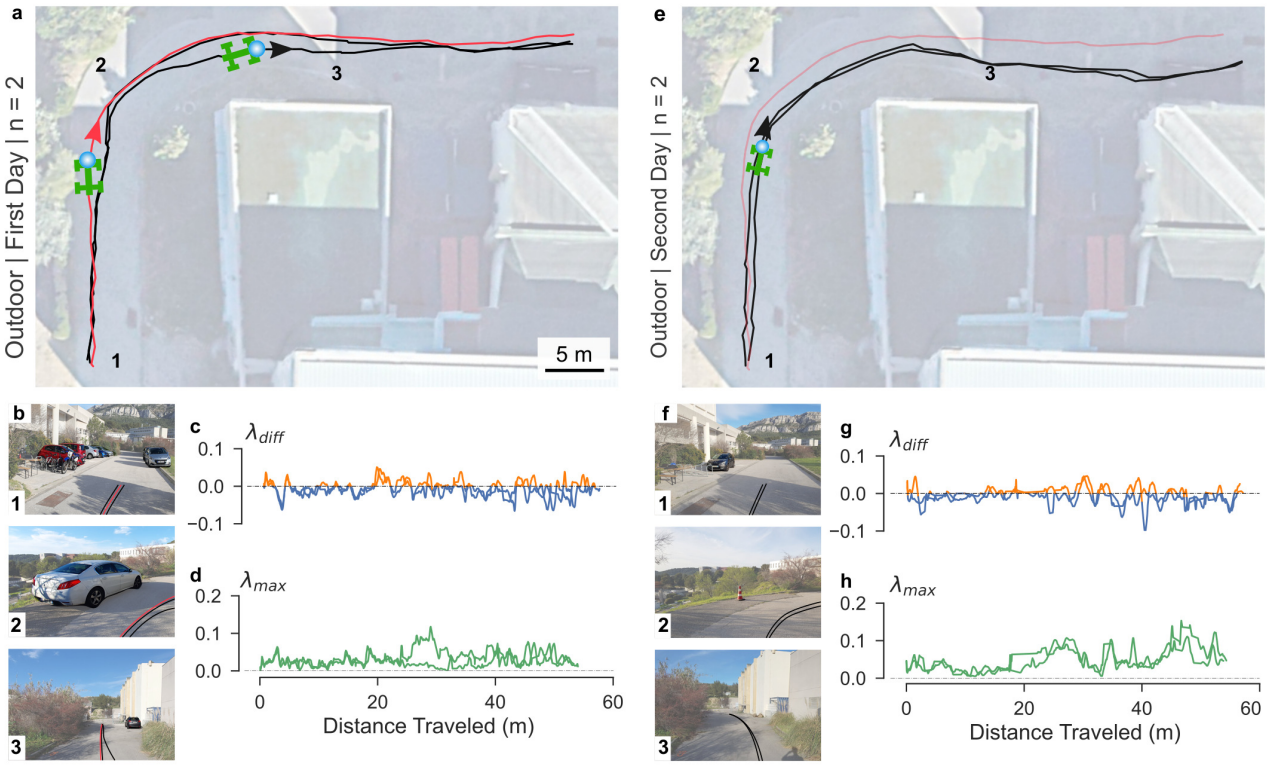


Fig. 6 Real-world experiments of outdoor route-following with shared memories in a semi-urban environment. a First-day experiments: learning and autonomous route following with several cars along the 55m road. **b,f** Overviews of the outdoor environment on the first and second days, respectively. **c,d,g,h** Familiarities difference (λ_{diff}) and maximum (λ_{max}) values plotted against distance traveled on the first and second days, respectively. **e** Second-day experiments conducted at the same hour demonstrate autonomous route following using day-one memories in an altered, car-free environment.

this MBON, based on a familiarity threshold, acted as a motivational cues to halt route-following behavior and reducing the robot's linear velocity. This mechanisms was sufficient for the robot to successfully reach and stop at the nest area in 4 out of 5 trials, with a median stopping distance of 1.4 m (Fig. 7c, see also Supplementary Fig. S7b for detailed familiarities values over distance).

Shuttling: foodward and homeward routes

Reverse route-following is also commonly observed in ants and was successfully replicated on board Antcar. Homing ants can pull food items backward when it is too large to carry forward, maintaining body alignment with the outbound route learned forward, and using outbound memories with an opposite valance [52]. Shuttling tests show the robot's ability to switch movement direction and drive backward while maintaining alignment with the outbound route (Fig. 6b).

This foraging behavior was made possible by incorporating two additional place MBONs, which learned a series of panoramic views defining each endpoint of the route (feeder and nest). During shuttling, the model triggered a switch in motor gain polarity upon recognizing these panoramic views corresponding to the feeder or nest areas. In a cluttered indoor environment along a 6-meter learned route, the robot autonomously shuttled *to and fro* between the feeder and the nest, covering a total distance of 160 meters without interruption. Using a similar familiarity threshold on the two route-extremity MBONs, the robot detected the endpoints 22 times, achieving a median stopping distance of 0.31 m (Fig. 7d) (See Supplementary Fig. S7a for detailed familiarities values over distance).

This continuous shuttling revealed distinct differences in error profiles between forward and backward movement (Fig. 7b). During forward motion, the robot maintained stable control with minimal deviations (lateral error ±MAD = 0.1 ± 0.03 m, angular error ±MAD = 1.26 ± 0.83°, Fig. 7b). However, during backward motion, the traction-driven setup amplified steering effects, resulting in slightly larger deviations from both accuracy and precision, though overall performance remained acceptable (lateral error ±MAD = 0.19 ± 0.08 m, angular error ±MAD = 2.7 ± 2.1°, Fig. 7b and 8a). The increased 'motor' variability led to lower visual recognition signal and thus usefully affected speed, which decreased by 14% compared to forward motion (see Supplementary note 6, Fig. S5 and Table S1). Nonetheless, the robot consistently realigned with the correct path after such minor deviations. These results highlight the model's versatility across different driving dynamics, capability to implement inverted steering, and adaptability to variations in motor kinematics and propulsion.

Performance summary

Across all experiments, including both indoor and outdoor route-following, homing and shuttling tasks, the model demonstrated robust and stable navigation performance, completing 113 autonomous trajectories with a total of 1.6 km traveled. The theoretical limits of the system were validated, with convergence toward

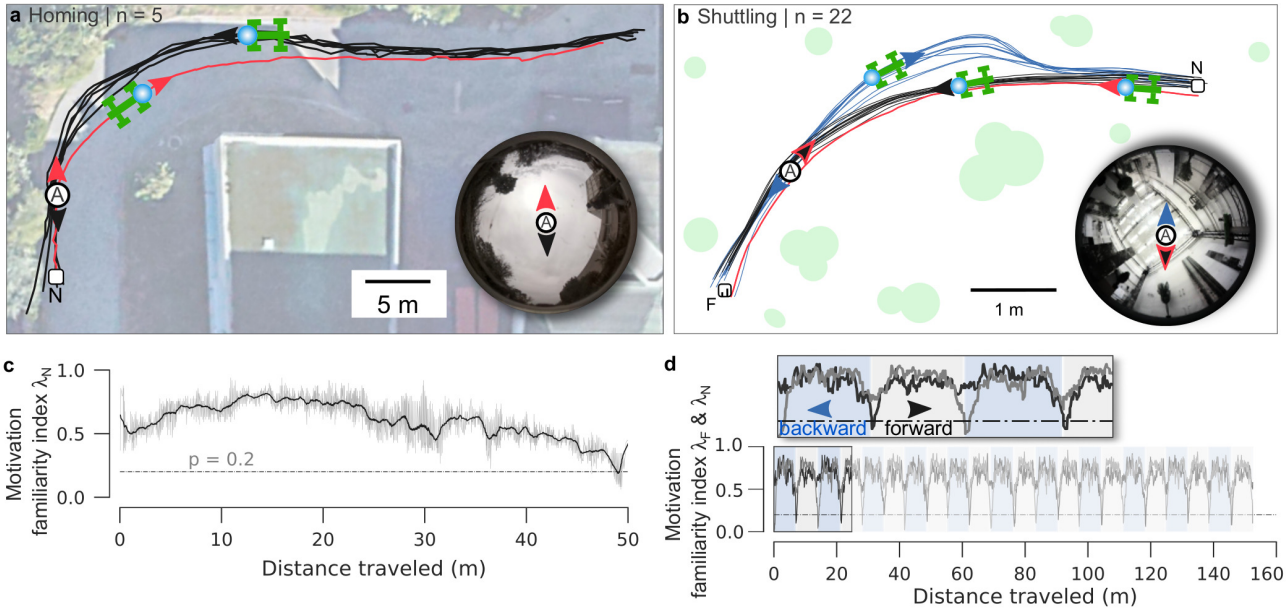


Fig. 7 Real-world experiments of outdoor homing and indoor shuttling. a Outdoor homing experiments along a 50 m L-shaped route in a cloudy environment. Two route MBONs and one motivation MBON guided an autonomous return route in the opposite direction. **b** Indoor shuttling experiments with artificial visual cues, using two route MBONs and two place MBONs. Autonomous routes swing back (blue) and forth (black). In both figures, the inset shows the robot's point of view. **c** Familiarity nest index (λ_N) plotted against distance traveled under a fixed stopping condition ($p = 0.2$). **d** Familiarity nest (λ_N) and feeder (λ_F) indices plotted against the traveled distance, with an inset zooming in on the values to highlight the backward and forward movements.

equilibrium points consistently achieved under various environmental conditions, even in the presence of noise (lateral error ±MAD = 0.24 ±0.13 m, angular error ±MAD = 4 ±2.7°, Fig. 8b). Lateral errors were within acceptable margins for both indoor and outdoor contexts, aligning within the standard widths of roads in France (5m) and typical indoor corridor (1.5m).

Additionally, statistical analysis showed no significant differences in the lateral or angular errors across the eleven test scenarios (Kruskal-Wallis test, $H = 2.10$ for lateral error, $p = 0.99$; $H = 8$ for angular error, $p = 0.76$), underscoring the system's reliability across diverse conditions (see Statistical Information). These results highlight the robustness and adaptability of the route-centric MB model in dynamic environments, confirming its potential applicability in a variety of navigation contexts.

Steering memory capacity

In classical single-MBON familiarity models, memory capacity (m) is defined by an error probability (P_{error}), the chance of confusing an unfamiliar visual pattern with a learned one [39]. With our chosen parameters ($N = 15,000, \kappa = 0.01$), theoretical estimates predict a memory capacity of $m = 346$ KC activity patterns stored for a 1% error probability equivalent to roughly 38m of route in cluttered environment [39]. However, such binary classification metrics fail to capture the complexity of our route-centric lateralized MBON steering mechanism.

To better quantify our model's steering-specific memory performance, we propose an empirical metric ($\mathrm{Pl}_{\mathrm{error}}$), defined as the proportion of visual patterns incorrectly activating the opposite MBON (e.g., images associated with a leftward orientation producing lower familiarity on the left MBON than on the right MBON, and vice versa). Therefore, we define a steering-specific memory capacity, (see Fig. 9, Methods section and Supplementary Fig. S8).

Under our metric, during extended route-following experiments in a semi-urban setting (55 m, 430 unique KC activity patterns, Fig.6), our network achieved 0.7% of error, below the critical threshold of 1% confusion probability (Fig.9a). Further analysis of memory scalability using a larger dataset (up to 250 m; Supplementary Fig. S8) indicated that under our current network parameters ($N = 15,000, \kappa = 0.01$), a 1% error probability yields an estimated 500 patterns, corresponding to about 65 m of stored route in this conditions. A basic linear extrapolation indicated that with a larger network ($N = 100,000$, $\kappa = 0.005$), routes as long as 600 m could be reliably encoded at the error threshold of 1% (Fig. 9b).

Discussion

We introduced a fully embodied, biologically grounded route-following system that unifies route-centric lateralized visual memory, panoramic input encoding, and neuromorphic control within a real-world robotic platform. Unlike prior ant-inspired models, which remained limited to short-range tasks or simulations, our model delivers continuous, one-shot learning and stable bi-directional navigation of visual routes — using only a single 800-pixel sensor, four MBONs, and light computational resources. These results mark a significant step forward in insect-inspired embodied cognition for real-world robotic autonomy.

The angular error between the agent's head direction and the dynamic local route orientation (i.e route-centric, defined in the Methods as the Frenet frame [55]) emerged as both a challenge during exploitation—where

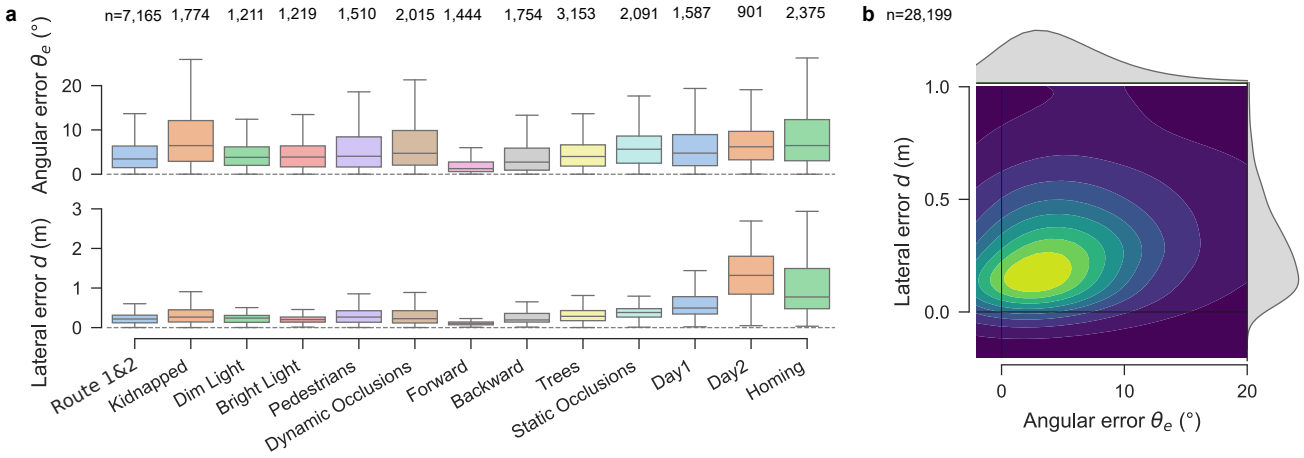


Fig. 8 Performance during route following overview a Detailed error analysis for each experiment. **b** Weighted bi-variate distribution for lateral (d) and angular errors (θ_e) across 13 experimental configurations.

the system tends to minimizes this error—and a cue during learning, where the categorization process depends on its polarity. Our model demonstrated homing behavior using either a 180° shift in visual oscillation or by inverting motor gains, thus enabling forward and backward movements with only a single foodward learning route. Additionally, visual place memories stored in supplementary MBONs, paired with a motivational control system, allowed the robot to recognize route endpoints and modulate motor gains, halting movement or reversing foraging motivation. With a single learning pass in one direction, the agent could follow the route forward, backward, and in reverse, controlled by oscillation parameters and motivational cues. Only motivational rules required adjustment to switch between route following, homing, and shuttling, underscoring the model's flexibility.

Compared to earlier ant-inspired familiarity-based models, generally limited to short, indoor routes or stop-and-scan strategies, our system demonstrates practical advantages in scalability, efficiency, and real-world adaptability [19–24]. Our memory footprint and command computation time are significantly lower than those reported in panoramic visual route-following methods [13], which require 3 megabytes per kilometer and 400 ms per control update. In contrast, our model operates at 0.3 megabytes per kilometer and produces commands within 75 ms on a lightweight embedded board—enabling real-time operation in constrained platforms. Additionally, it achieves competitive performance compared to snapshot-based visual route-following methods that use odometry [15].

This route-centric lateralized MB model distinguishes itself through reduced time and space complexity for route direction processing compared to perfect memory, snapshot, and insect-based visual compass approaches. Whereas time and space complexity increase with the number of images in perfect memory or snapshot models, our MB model maintains constant space complexity, relying only on the synaptic matrix size **KCtoMBON**. Additionally, in contrast to visual compass approaches, where computational complexity scales with in-silico scan range and resolution during exploitation ($\mathcal{O}(n)$), our MB model maintains a constant factor ($\mathcal{O}(1)$) since in-silico scanning is only required during learning. For instance, while a insect-based visual compass scanning a ±45° range at 1° resolution requires 90 comparisons per image, our model requires only two comparisons, eliminating the need for angular scanning in exploitation. Notably, our model produced commands five times faster than the insect-based visual compass approach on the same robot platform [21, 22].

Our contribution also aligns well with current biological observations, particularly highlighting the effectiveness of latent learning, where continuous learning bypass the need to control "when to learn" [32, 45]. The opposed event-triggered and snapshot-based learning models producing place learning [15, 56] where used

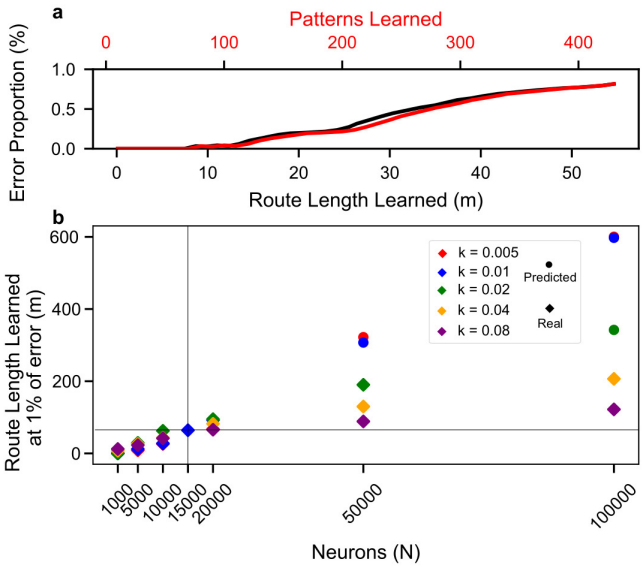


Fig. 9 Memory capacity analysis. a Error proportion $\mathrm{Pl}_{\mathrm{error}}$ (defined as cases where the current visual pattern activates the incorrect lateralized MBON) as a function of route length and visual patterns learned, from outdoor experiments conducted on day one. **b** Route length that our MB architecture can store maintaining an error proportion below 1%, plotted against the number of neurons and k-WTA parameter. The dataset is from a 250 m outdoor route (real) and extrapolation is made on two couple of parameter (because it did not reach the 1% before 250m, see Supplementary Fig. S8). Under our current network parameters ($N = 15,000, \kappa = 0.01$), our route memory capacity with a 1% error probability corresponds to approximately 65 m.

here only to recognise place of interests such as the nest and the feeder to switch motivation, but were not engaged for route guidance. Also, our MB model prioritized body orientation within the route frame rather than splitting the visual field [23, 48], aligning with biological observations in ants with unilateral visual impairment, showing that these insects store and recognise fundamentally binocular views [49]. Interestingly, the linear relationship observed between familiarity measures and route-centric angular error during exploitation closely mirrors experimental findings in ants with nest-centric models [44]. This relationship enabled us to demonstrate the asymptotic stability of the system within a defined domain, ensuring the consistent and predictable behavior essential for a robotic navigation model [57].

Furthermore, oscillatory learning behavior mirrors ant behavior, where initial routes involve slow, rotational movements, transitioning to direct paths on subsequent journeys [40]. These oscillations typically fall within $\pm 100°$, with peaks around $\pm 45°$ in unfamiliar terrain [41, 43]. The robot's ability to slow down and produce emerging mechanical oscillation upon entering unfamiliar areas (see Figs. 5 a,b and Supplementary Movie 1) are consistent with such naturalistic behaviors. Finally, Antcar's homing capability was maintained even when navigating backward, closely mirroring ant behavior while dragging food [50–52, 58]. Overall, our attempt to integrate multiple MBONs, oscillations, "turn back and look" behavior, and motivational control mechanisms echoes insect mechanisms [2, 59], and the resulting expression when implemented in the robot echoes insect behaviours.

This study addresses several core needs identified in research on embodied neuromorphic intelligence [6, 8], such as robustness to visual changes, adaptability to real-world environments, and support for extended route learning. Our algorithm's efficiency allows computational power for additional tasks, making it valuable in GPS-compromised or SLAM-disrupted scenarios (Simultaneous Localization And Mapping). The robot's low-resolution, wide-angle vision proves resilient against moving objects that often disrupt SLAM. Our model is particularly effective in dynamic environments or scenarios where traditional odometry—whether visual, inertial, step-counting, or wheel-rotation—is unreliable. It can serve as a low-resource backup system (running on a 4-core CPU) alongside other processes, or be integrated with additional sensors to enhance state estimation robustness [60, 61].

Interestingly, the semi-random encoding process, specifically the **PNtoKC** synaptic projections, introduces a "fail-secure" memory-sharing mechanism. If synaptic weights for encoding differ, memory sharing becomes inaccessible, an advantageous feature for swarm robotics or cross-robot memory sharing. Future research could enhance this approach but also transitioning this model to a spiking neural network on neuromorphic hardware could further enhance computational efficiency and biological fidelity [11]. Additionally, incorporating obstacle avoidance [62, 63], would improve performance in dynamic environments.

Our upward-facing fisheye camera provides a 360° view at minimal cost, and in silico rotation keeps the learning route straight. However, if the ground is not planar, in silico rotation may not accurately represent physical rotation, thus learn image that the robot will never encounter. Moreover, bypassing image unwrapping and calibration speeds computation but reduces the influence of distal cues, especially in large open fields where the horizon is severely deformed and can appear uniform. Interestingly, salt lake ants in such environments have developed enhanced horizontal resolution and accuracy, and build artificial visual cues near the nest entrance [64].

A reduced visual field, as seen in more general cases, may preclude in silico scanning, necessitating an estimation of the angular error between the road frame and the agent. Collett et al. [65] proposed that ants use route segment odometry for navigation, suggesting a potential alternative. In insects, dopaminergic neurons (DAN) modulates MBON synapses from Kenyon cell inputs in response to motor stimuli, effectively acting as a "supervisor" for sensory cue categorization [66]. There are plenty of feedbacks from motor areas towards the MB (both DANs and MBON)[67] that could enable to orchestrate visual memory formation , for instance, based on whether the current body orientation is left or right from a goal compass direction [45]. Similarly, our model's left–right classification functionally mimics the effect of dopaminergic feedbacks (onto MB), which activity would be based on whether the current body orientation is left or right from the local route heading.

Our approach does not cover beeline homing post-foraging or search behaviors near points of interest, although these could be added by adding path integration mechanisms [68] or using the current visual mechanism but adding "learning walk" behaviors around place of interest [45]. Although anti-Hebbian learning effectively improves familiarity discriminability by sparsifying **KCtoMBON** synapses, there is a theoretical limit to this benefit. Excessive sparsification would lead to memory saturation (dramatic forgetting), where the MBON output no longer discriminates between familiar and unfamiliar inputs. Therefore, it suggest an opportunity for further exploration by adjusting KCs numbers or connectivity by testing different MB learning mechanisms with feedback or prediction error [69].

Finally, our system's memory capacity metric (500 patterns) is higher than the theoretical estimation from Ardin et al. (346 patterns) [39]. This discrepancy may arises because the theoretical model assumes random, independent KC activations, whereas in continuous route learning, KC activations are correlated and structured. Additionally, our empirical error calculation depends on our visual preprocessing, hardware and the semi-urban environment. Also, the memory capacity error computed here can be linked to the sparsity of the **KCtoMBON** synaptic weight, avoiding a possible dramatic forgetting when all the KCs are used (i.e the **KCtoMBON** vector is 0). To extend route length while reducing error, we could increase the number of KCs (see Fig. 9); however, this may slow processing due to larger matrices—though the impact

may be minimal on a GPU. Alternatively, incorporating additional pairs of MBONs for different route segments, could enable longer and multiple route memories without affecting processing time and therefore considering an extrapolated memory footprint of only 0.3 megabyte per kilometer. Also, expanding the number of MBONs, akin to the 100 in honey bees [70], along with a mechanism (such as a context-based selection) to dynamically activate the appropriate MBON pair would enable more complex motivational states, multi route memory storage, and broader navigational abilities [71].

Inspired by the neuroethology of ants, our route-centric, lateralized MB model provides an effective bridge between theoretical insights and practical application in insect-inspired robotic navigation. This egocentric architecture demonstrates how biologically plausible neural circuits can support robust, scalable, and adaptive behavior in real-world robots—using accessible hardware, panoramic input, and minimal computing resources. These results reinforce the promise of neuromorphic approaches for embodied autonomy, and broaden opportunities for both robotic systems and experimental studies in comparative cognition.

Methods

This section describes the methodology used in the present study, focusing on the Encoding, Learning, and Exploitation processes of the proposed MB model (Figs. 2b-d and Fig. 10). We also provide details on the hardware setup, control architecture, and stability analysis.

Image Encoding

Inspired by the visual system of ants [72], the model encoded real-world images into sparse, binary neural representations to efficiently handle visual input.

The encoding function (Fig. 2b, Fig. 10 and Supplementary Note 5) processed panoramic images from a camera with a 220° vertical and 360° horizontal field of view. This wide field of view enabled the camera to capture from slightly below the horizon to nearly directly below itself. To enhance natural contrast, the green channel of each image was selected [72], followed by Gaussian smoothing ($\sigma = 3$ pixels) to reduce noise. The chosen σ corresponds to an acceptance angle of about 4.4° ($\sigma + \mathrm{FOV}/r$), slightly larger than the estimated visual acceptance angle of the *Melophorus* species [73]. The image was then downsampled to an low-resolution 32×32 pixel thumbnail (0.145 pixel per degree), approximating the visual resolution of ants at 7.1° between adjacent photoreceptors [64]. Next, a Sobel filter extracted edges, mimicking lateral inhibition as seen in insect optical lobes [74]. These processed images were flattened into $n = 800$ Visual Projection Neurons ($\mathbf{PN} \in \mathbb{R}^n$), comparable to the number of ommatidia in *Cataglyphis* ants. Theses processing steps, with the aim of mimicking the output of the optic lobes, are described in details in Supplementary note 5 and in Fig. 10.

The **PN**s were further expanded into Kenyon Cells (KCs) using a fixed, sparse pseudo-random binary synaptic matrix (**PNtoKC**, equation 1). Each KC received input from four **PN**s, enhancing the visual encoding's discriminative power within the MB, forming an Excitatory Post Synaptic Projection (**EP**) vector (see equation 2 and Fig. 10).

$$\mathbf{PNtoKC} = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{N \times n} \tag{1}$$

So, the product

$$\mathbf{EP} = \mathbf{PNtoKC} \cdot \mathbf{PN} \tag{2}$$

will be a N-dimensional vector, i.e., $\mathbf{EP} \in \mathbb{R}^N$. The **EP** vector size was set to $N = 15,000$ for the route MBONs (MBON_R and MBON_L), while for place-specific MBONs (MBON_N and MBON_F), which required fewer images, N was set to 5,000.

A κ-Winner-Take-All (WTA) mechanism was applied to capture the highest contrasts, creating a high-dimensional, sparsified binary vector (see Fig. 10). This vector, referred to as the Action Potential (**AP**, equation 3), consequently activated only 1% of KCs ($\kappa = 0.01$), giving $N_u = N * \kappa$ active neurons, such that :

$$\mathbf{AP}_i = \begin{cases} \mathbf{EP}_i, & \mathbf{EP}_i \in \{k \text{ largest element of} \mathbf{EP}\} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where i is the neurons number, and $\mathbf{AP} \in \{0,1\}^N$. This final binary representation served as the encoded visual input to be learned. All parameters were predefined by literature and experimental tests, but not further optimized.

Routes and places learning

The learning process is governed by synaptic depression through anti-Hebbian learning. For each MBONs, their synaptic weight vector (**KCtoMBON**) dynamically adjusted their binary weight based on input from the **AP** layer.

$$\mathbf{KCtoMBON}_i = \begin{cases} 0, & \text{if } \mathbf{AP}_i = 1 \\ \mathbf{KCtoMBON}_i, & \text{otherwise} \end{cases} \tag{4}$$

Where $\mathbf{KCtoMBON} \in \{0,1\}^N$. At the beginning, for each MBONs, their **KCtoMBON** vector are fully connected (i.e. all **AP** neuron are connected to the MBONs). During learning, each time the **KCtoMBON** have the same weight as the current **AP** (1), the respective synaptic connection is depressed (weight from 1 to 0) in the designed **KCtoMBON** (equation 4). For the route, the **KCtoMBON** vector is designed (therefore depressed) according to the route-centric left or right looking direction (equation 4). The model assumed the robot perfectly aligned to the route being learned. The body rotation was estimated as $\hat{\theta}_e = \theta_e + \theta_c$, where therefore $\theta_e = 0$ during learning, and θ_c is the image rotation angle. The encoded binary image (**AP**) was

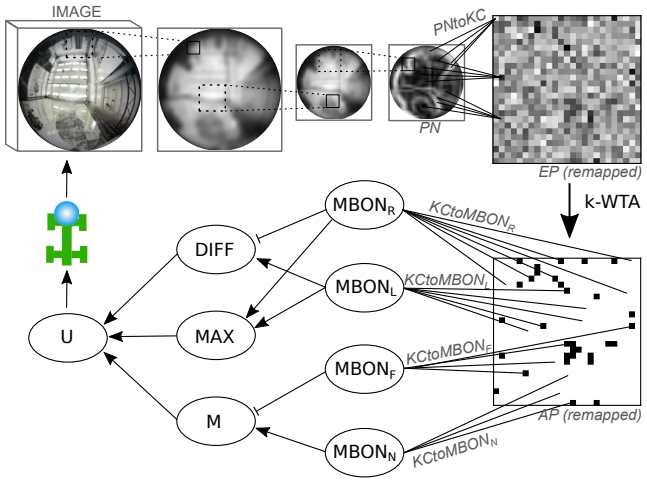


Fig. 10 Simplified mushroom body neural network. The network includes four MBONs fitted during the exploration phase. For visualization, the **EP**, **AP**, and **KCtoMBON** connections are reshaped into matrix-like formats. **KCtoMBON** synapses are initially fully connected prior to learning.

learn in one MBONs based on the polarity of $\hat{\theta}_e$, such that:

$$\begin{cases} \text{Learn}(\mathbf{AP}, \mathbf{KCtoMBON}_R), & \text{if } \hat{\theta}_e \leq 0 \\ \text{Learn}(\mathbf{AP}, \mathbf{KCtoMBON}_L), & \text{if } \hat{\theta}_e \geq 0 \end{cases} \tag{5}$$

Where Learn() is the function in equation 4. The simulated oscillatory movements during learning were obtained by rotating each captured image in steps, creating a sweep of rotations (θ_c) described by the following function:

$$\theta_c(n) = A \cdot \sin\left(n \cdot \Delta\theta + \phi\right) \quad \text{for } n = 0, 1, 2, \ldots, \frac{2A}{\Delta\theta} \tag{6}$$

where A represents the oscillation amplitude, $\Delta\theta$ the step size, and ϕ the phase shift. The step size was fixed at $\Delta\theta = 5°$, with $A = 45°$ for route MBONs and $A = 30°$ for place MBONs. The phase shift was $\phi = 180°$ only for the homing task (Fig. 7). The place MBONs learning was triggered by an external stimulus, specified by the robot operator.

Synaptic weights (**KCtoMBON**) were stored in CSR format (Compressed Sparsed Row), achieving significant data compression to 148 kilobits independently of the route length (from 6m to 55m), reducing memory requirements by 99.97% from cumulative image storage. This self-supervised model continuously learned visual input at high throughput without memory overload, as only novel views (i.e., newly recruited KCs) modulated synapses. Several panoramic views were learned to define the start and finish areas in their respective MBONs, serving as motivational cues.

Exploitation process and control architecture

During exploitation, the model calculated familiarity scores (λ) by comparing the current input (**AP**) with each MBON's synaptic weight matrix (**KCtoMBON**):

$$\lambda = \frac{1}{N_u} \sum_{i=1}^{N} \mathbf{AP}_i \cdot \mathbf{KCtoMBON}_i \tag{7}$$

This familiarity score, ranging from 0 (unfamiliar) to 1 (familiar), was used to assess route alignment. The lateralized difference in familiarities between the left and right MBONs (λ_{diff}, equation 8 and Fig. 10), which indicates whether the current view is more oriented to the left or right of the route, guided the robot's steering angle (φ). Meanwhile, the maximum familiarity ($\lambda_{\max}$), representing how familiar the current view is, modulated its speed (v).

$$\lambda_{\text{diff}} = \lambda_L - \lambda_R \quad \textit{and} \quad \lambda_{\max} = \max(\lambda_L, \lambda_R) \tag{8}$$

Thus, the control input $\mathbf{U}$ was defined as:

$$\mathbf{U} = \begin{bmatrix} v \\ \varphi \end{bmatrix} = \begin{bmatrix} M \cdot K_v \cdot \text{sat}(1 - \lambda_{\max}) \\ M \cdot K_\varphi \cdot \lambda_{\text{diff}} \end{bmatrix} \tag{9}$$

Here, K_v and K_φ are proportional gains that control linear and angular velocities, while the saturation function (sat()) establishes a minimum throttle level, ensuring minimum speed even at low familiarity levels to allow the robot to turn (due to his mechanical constraint). The motivational state (M) regulated transitions between behaviors, based on a familiarity thresholds within place-specific MBONs. During route following, M was consistently set to 1. In homing experiments, where the objective was to stop at the nest, M initially started at 1 and switched to 0 once the familiarity of the nest-specific MBON (λ_{N}) fell below a fixed threshold ($p = 0.2$), signaling arrival at the nest. For shuttling tasks, M alternated between values of 1 and -1 as the robot reached each route extremity, driven by a familiarity thresholds of the two place-specific MBONs (λ_{N} and λ_{F}).

Theoretical analysis of the robot stability

Stability in mobile agents, biological or robotic, is essential for reliable, predictable behavior. In control theory, an agent's motion is generally modeled as $\dot{x} = f(\mathbf{x}, \mathbf{U})$, where $\mathbf{x}$ is the state vector (e.g., position or velocity), $\mathbf{U}$ is the control input, and f describes system dynamics. A desired equilibrium point $\mathbf{x}_{\text{e}}$ is achieved by defining a control input $\mathbf{U}_{\text{e}}$ such that $f(\mathbf{x}_{\text{e}}, \mathbf{U}_{\text{e}}) = 0$, allowing the system to maintain stability and return to equilibrium after disturbances. Stability is typically assessed using a Lyapunov function [57], which ensures the system converges to a stable state over time.

In contrast to conventional control approaches, we applied a neuroethologically inspired control input derived from ant behavior and assessed its stability via an a posteriori Lyapunov analysis. The robot's motion was modeled in a Frenet frame, a moving reference frame coincident with the nearest point on the route, to minimize lateral and angular errors, defined by $\mathbf{x} = [d, \theta_{\text{e}}]$. Empirical data for stability assessment was collected in indoor and outdoor environments (paths of approximately 6 meters with 855 learned images

each), providing distinct visual contexts (Figs. 2, 3). The robot's equations of motion from a global to the Frenet frame are [75]:

$$\begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v\left(\cos\theta_\mathrm{e} - \tan\varphi \sin\theta_\mathrm{e}\right) \\ v\left(\sin\theta_\mathrm{e} + \tan\varphi \cos\theta_\mathrm{e}\right) \\ v\frac{\tan\varphi}{L} \end{bmatrix}, \tag{10}$$

where s is the arc length along the route, d is the lateral error, and θ_e is the angular error.

This kinematic model, along with by empirical observations (Fig. 3), enabled us to establish an asymptotically stable domain for lateral and angular errors (d and θ_e), ensuring reliable route-following performance even with minor disturbances. The full theoretical stability proof and derivations of the model in the Frenet frame are provided in the Supplementary note 3 and 4.

Antcar robot and ground truth system

The experiments were conducted using Antcar (Fig. 1 and Fig. 2a), a PiRacer AI-branded car-like robot. Antcar features four wheels, with two rear drive wheels powered by 37-520 DC motors (12V, 1:10 reduction rate) and a front steering mechanism controlled by an MG996R servomotor (9kg/cm torque, 4.8V). The robot's chassis measures $13 \times 24 \times 19.6$ cm and is powered by three rechargeable 18650 batteries (2600mAh, 12.6V output). Antcar's primary sensor is a 220° Entaniya™ fisheye camera, mounted upward to capture panoramic images at $160 \times 160px \times 3$ resolution, processed using OpenCV on a Raspberry Pi 4 Model B (Quad-core Cortex-A72, 1.8GHz, 4GB RAM), running Ubuntu 20.04. Note that there was no closed-loop control on the wheel rotation speed. Raspberry Pi manages real-time performance and controls the motors through a custom ROS architecture.

Real-time communication is facilitated by ROS Noetic, either via Wi-Fi (indoor) or a 4G dongle (outdoor). The robot can be controlled manually using a keyboard, joystick or with GPS waypoint, but in autonomous visual-only mode, it follows its own internal control law. Control inputs—steering angle (φ) and throttle (v) are processed using the PyGame library. Real-time data visualization and post-experiment monitoring are achieved via Foxglove.

Antcar has a maximum velocity of 1.5 m/s and a maximum steering angle of 1 rad, with a wheelbase of 0.15 m. The robot's configuration states $\mathbf{q} = (x, y, \theta)$ were tracked using different systems. Indoor experiments utilized eighteen Vicon™ motion capture cameras, with infrared markers on Antcar providing precise tracking at 50 Hz with 1 mm accuracy. Outdoor experiments employed a GPS-RTK system with a SparkFun GPS-RTK Surveyor, providing 14 mm accuracy at 2 Hz (GPS-RTK stands for Global Positioning System - Real-Time Kinematic). Ground speed and angular speed were calculated through position differentiation. The base station used for GPS corrections was a Centipede LLENX station located at 24 km (Marseille Provence Airport) from the experiment site in Marseille. Note that the ground truth acquisition system was run on the Raspberry Pi along with the MB model.

Lateral error was calculated by finding the nearest point on the learning route using the Euclidean distance, with the shortest distance representing the absolute lateral error. Angular error was defined as the absolute difference in heading between the nearest learning route point and the current position. The euclidean distance between the agent and the nest or feeder areas was calculated to estimate the distance when the robot switched behavior (i.e familiarity dropped below the threshold).

Memory capacity computation

To compute our memory capacity and $\mathrm{Pl}_\mathrm{error}$, we used a 250 m dataset (Supplementary Fig. S8). The metric was computed over the operating area defined by the learning oscillation. The computation was performed incrementally: for each new learned image (rotated from $-45°$ to $45°$ in $5°$ steps), all previously encountered images were tested (rotated from $-45°$ to $45°$ in $1°$ steps), we didn't take into account the value between -1° and 1°. For each tested image i, the familiarity difference $\lambda_\mathrm{diff}[i]$ was calculated. An error indicator e_i was then defined by comparing the sign of $\lambda_\mathrm{diff}[i]$ with that of the heading error $\theta_\mathrm{e}[i]$:

$$e_i = \begin{cases} 1 & \text{if } \mathrm{sign}(\lambda_\mathrm{diff}[i]) \neq \mathrm{sign}(\ \theta_\mathrm{e}[i]) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Finally, the cumulative mean error up to test image i is given by:

$$\mathrm{Pl}_\mathrm{error}(i) = \frac{1}{i+1} \sum_{j=0}^{i} e_j \tag{12}$$

Statistical informations

The errors used for statistics were recorded at each command decision timing. Due to non-normality in error values (with outliers retained), Box-Cox transformations were applied to stabilize variance across experiments, reducing the impact of outliers caused by indoor obstacles that hid the robot from the motion capture system or by GPS-RTK inaccuracies outdoors. The groups was compared using the Kruskal-Wallis test [76], and median values are reported with median absolute deviation (MAD), as median ± MAD. The package python SciPy [77] was used for the statistics. The overall medians and bivariate distribution plots were weighted by the number of measurements per experiment for the Fig. 8.

Acknowledgments

The authors thank David Wood for revising the English in this study, Guillaume Caron for providing the camera reference, and Thomas Gaillard, Clément Serrasse, and Hamidou Diallo for their assistance during the robotic tests.

Declarations

- Funding: G.G. was supported by a doctoral fellowship grant from Aix Marseille University (amU) and the

French Ministry of Defense (AID - Agence Innovation Défense, agreement #A01D22020549 ARM/DGA /AID). G.G., J.R.S. and F.R. were also supported by Aix Marseille University and the CNRS Institutes (Biology, Informatics as well as Engineering). The facilities for the experimental tests has been mainly provided by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015).

- Conflict of interest: the authors declare no competing interests.
- Data availability: The dataset, including image banks and experimental data, is available at Figshare: https://doi.org/10.6084/m9.figshare.27708105[78]
- Code availability: The code for the MB model, dataset analysis, and figure generation is available on GitHub: https://doi.org/10.5281/zenodo.15783472 [79]. The ROS Python code is available upon request.
- Supplementary Movie 1: https://youtu.be/6YIkDhN4SaY
- Author contribution: G.G.G., A.W., J.R.S., and F.R. designed this research work; G.G.G, A.W., J.R.S., and F.R. got funding for this study; A.W. was supported by ERC Consolidator Grant RESILI-ANT no. 101125881. G.G.G. performed experiments, collected and visualized the data; G.G., A.W., J.R.S., and F.R. analyzed data; G.G.G. wrote the first full draft. All authors reviewed the results and approved the final version of the manuscript.

References

[1] Franceschini, N. Small Brains, Smart Machines: From Fly Vision to Robot Vision and Back Again. *Proceedings of the IEEE* **102**, 751–781 (2014).

[2] Webb, B. & Wystrach, A. Neural mechanisms of insect navigation. *Current Opinion in Insect Science* **15**, 27–39 (2016).

[3] Denuelle, A. & Srinivasan, M. V. A sparse snapshot-based navigation strategy for uas guidance in natural environments (2016).

[4] Glick, P. E., Balaram, J. B., Davidson, M. R., Lyons, E. & Tolley, M. T. The role of low-cost robots in the future of spaceflight. *Science Robotics* **9**, eadl1995 (2024).

[5] Yang, G.-Z. *et al.* The grand challenges of *Science Robotics*. *Science Robotics* **3**, eaar7650 (2018).

[6] de Croon, G. C., Dupeyroux, J., Fuller, S. B. & Marshall, J. A. Insect-inspired ai for autonomous robots. *Science Robotics* **7**, eabl6334 (2022).

[7] Mangan, M. *et al.* A virtuous cycle between invertebrate and robotics research: Perspective on a decade of Living Machines research. *Bioinspiration & Biomimetics* **18**, 035005 (2023).

[8] Bartolozzi, C., Indiveri, G. & Donati, E. Embodied neuromorphic intelligence. *Nature Communications* **13**, 1024 (2022).

[9] Webb, B. Robots in invertebrate neuroscience. *Nature* **417**, 359–363 (2002).

[10] Franz, M. O. & Mallot, H. A. Biomimetic robot navigation. *Robotics and Autonomous Systems* (2000).

[11] Sandamirskaya, Y., Kaboli, M., Conradt, J. & Celikel, T. Neuromorphic computing hardware and neural architectures for robotics. *Science Robotics* **7**, eabl8419 (2022).

[12] Simon, M., Broughton, G., Rouček, T., Rozsypálek, Z. & Krajník, T. in *Performance Comparison of Visual Teach and Repeat Systems for Mobile Robots* (eds Mazal, J. *et al.*) *Modelling and Simulation for Autonomous Systems*, Vol. 13866 3–24 (Springer International Publishing, Cham, 2023).

[13] Stelzer, A., Vayugundla, M., Mair, E., Suppa, M. & Burgard, W. Towards efficient and scalable visual homing. *The International Journal of Robotics Research* **37**, 225–248 (2018).

[14] Nourizadeh, P., Milford, M. & Fischer, T. Teach and Repeat Navigation: A Robust Control Approach. *2024 IEEE International Conference on Robotics and Automation (ICRA)* 2909–2916 (2024).

[15] Van Dijk, T., De Wagter, C. & De Croon, G. C. H. E. Visual route following for tiny autonomous robots. *Science Robotics* **9**, eadk0310 (2024).

[16] Mangan, M. & Webb, B. Spontaneous formation of multiple routes in individual desert ants (Cataglyphis velox). *Behavioral Ecology* **23**, 944–954 (2012).

[17] Kohler, M. & Wehner, R. Idiosyncratic route-based memories in desert ants, Melophorus bagoti: How do they interact with path-integration vectors? *Neurobiology of Learning and Memory* **83**, 1–12 (2005).

[18] Wystrach, A., Schwarz, S., Schultheiss, P., Beugnon, G. & Cheng, K. Views, landmarks, and routes: how do desert ants negotiate an obstacle course? *Journal of Comparative Physiology A* **197**, 167–179 (2011).

[19] Kodzhabashev, A. & Mangan, M. Wilson, S. P., Verschure, P. F., Mura, A. & Prescott, T. J. (eds) *Route Following Without Scanning.* (eds Wilson, S. P., Verschure, P. F., Mura, A. & Prescott, T. J.) *Biomimetic and Biohybrid Systems*, Vol. 9222, 199–210 (Springer International Publishing, Cham, 2015).

[20] *Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network*, Vol. ALIFE 2019: The 2019 Conference on Artificial Life of *Artificial Life Conference Proceedings*.

[21] Gattaux, G., Vimbert, R., Wystrach, A., Serres, J. R. & Ruffier, F. Antcar: Simple route following task with ants-inspired vision and neural model. *Preprint HAL* (2023). https://hal.science/hal-04060451v1.

[22] Gattaux, G., Wystrach, A., Ruffier, F. & Serres, J. Enhancing ant-inspired visual compass with focused visual scan in a compact robot. *2025 IEEE 7th International Conference on AI Circuits and Systems (AICAS)* 110–114 (2025).

[23] Lu, Y., Cen, J., Alkhoury Maroun, R. & Webb, B. Insect-inspired Embodied Visual Route Following. *Journal of Bionic Engineering* (2025).

[24] Jesusanmi, O. O. *et al.* Investigating visual navigation using spiking neural network models of the insect mushroom bodies. *Frontiers in Physiology* **15**, 1379977 (2024).

[25] Caron, G., Marchand, E. & Mouaddib, E. M. Photometric visual servoing for omnidirectional cameras. *Autonomous Robots* **35**, 177–193 (2013).

[26] Cartwright, B. A. & Collett, T. S. Landmark learning in bees: Experiments and models. *Journal of Comparative Physiology ? A* **151**, 521–543 (1983).

[27] Möller, R. & Vardy, A. Local visual homing by matched-filter descent in image distances. *Biological cybernetics* **95**, 413–430 (2006).

[28] Zeil, J., Hofmann, M. I. & Chahl, J. S. Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A* **20**, 450 (2003).

[29] Wystrach, A., Cheng, K., Sosa, S. & Beugnon, G. Geometry, features, and panoramic views: ants in rectangular arenas. *Journal of Experimental Psychology: Animal Behavior Processes* **37**, 420 (2011).

[30] Gaffin, D. D. & Brayfield, B. P. Autonomous Visual Navigation of an Indoor Environment Using a Parsimonious, Insect Inspired Familiarity Algorithm. *PLOS ONE* **11**, e0153706 (2016).

[31] Philippides, A., Baddeley, B., Cheng, K. & Graham, P. How might ants use panoramic views for route navigation? *Journal of Experimental Biology* **214**, 445–451 (2011).

[32] Baddeley, B., Graham, P., Husbands, P. & Philippides, A. A Model of Ant Route Navigation Driven by Scene Familiarity. *PLoS Computational Biology* **8**, e1002336 (2012).

[33] Wystrach, A., Beugnon, G. & Cheng, K. Ants might use different view-matching strategies on and off the route. *Journal of Experimental Biology* **215**, 44–55 (2012).

[34] Linsker, R. Self-organization in a perceptual network. *Computer* **21**, 105–117 (1988).

[35] Wystrach, A., Mangan, M., Philippides, A. & Graham, P. Snapshots in ants? new interpretations of paradigmatic experiments. *Journal of Experimental Biology* **216**, 1766–1770 (2013).

[36] Heisenberg, M. Mushroom body memoir: From maps to models. *Nature Reviews Neuroscience* **4**, 266–275 (2003).

[37] Eichler, K. *et al.* The complete connectome of a learning and memory centre in an insect brain. *Nature* **548**, 175–182 (2017).

[38] Aso, Y. *et al.* Mushroom body output neurons encode valence and guide memory-based action selection in Drosophila. *eLife* **3**, e04580 (2014).

[39] Ardin, P., Peng, F., Mangan, M., Lagogiannis, K. & Webb, B. Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments. *PLOS Computational Biology* **12**, e1004683 (2016).

[40] Haalck, L. *et al.* CATER: Combined Animal Tracking & Environment Reconstruction. *SCIENCE ADVANCES* (2023).

[41] Deeti, S., Cheng, K., Graham, P. & Wystrach, A. Scanning behaviour in ants: An interplay between random-rate processes and oscillators. *Journal of Comparative Physiology A* (2023).

[42] Wystrach, A., Philippides, A., Aurejac, A., Cheng, K. & Graham, P. Visual scanning behaviours and their role in the navigation of the australian desert ant melophorus bagoti. *Journal of Comparative Physiology A* **200**, 615–626 (2014).

[43] Clement, L., Schwarz, S. & Wystrach, A. An intrinsic oscillator underlies visual navigation in ants. *Current Biology* **33**, 411–422 (2023).

[44] Wystrach, A., Le Moël, F., Clement, L. & Schwarz, S. A lateralised design for the interaction of visual memories and heading representations in navigating ants. *Preprint bioRxiv* (2020). https://www.biorxiv.org/content/10.1101/2020.08.13.249193v1.

[45] Wystrach, A. Neurons from pre-motor areas to the Mushroom bodies can orchestrate latent visual learning in navigating insects. *Preprint bioRxiv* (2023). https://www.biorxiv.org/content/10.1101/2023.03.09.531867v1.

[46] Stürzl, W., Zeil, J., Boeddeker, N. & Hemmi, J. M. How Wasps Acquire and Use Views for Homing. *Current Biology* **26**, 470–482 (2016).

[47] Zeil, J. Visual homing: an insect perspective. *Current opinion in neurobiology* **22**, 285–293 (2012).

[48] Steinbeck, F. *et al.* Familiarity-taxis: A bilateral approach to view-based snapshot navigation. *Adaptive Behavior* 10597123231221312 (2024).

[49] Schwarz, S., Clement, L., Haalck, L., Risse, B. & Wystrach, A. Compensation to visual impairments and behavioral plasticity in navigating ants. *Proceedings of the National Academy of Sciences* **121**, e2410908121 (2024).

[50] Ardin, P. B., Mangan, M. & Webb, B. Ant Homing Ability Is Not Diminished When Traveling Backwards. *Frontiers in Behavioral Neuroscience* **10** (2016).

[51] Pfeffer, S. E. & Wittlinger, M. How to find home backwards? Navigation during rearward homing of *Cataglyphis fortis* desert ants. *Journal of Experimental Biology* **219**, 2119–2126 (2016).

[52] Schwarz, S., Clement, L., Gkanias, E. & Wystrach, A. How do backward-walking ants (Cataglyphis velox) cope with navigational uncertainty? *Animal Behaviour* **164**, 133–142 (2020).

[53] Freas, C. A. & Spetch, M. L. Terrestrial cue learning and retention during the outbound and inbound foraging trip in the desert ant, cataglyphis velox. *Journal of Comparative Physiology A* **205**, 177–189 (2019).

[54] Jocher, G., Qiu, J. & Chaurasia, A. Ultralytics YOLO (2023). URL https://github.com/ultralytics/ultralytics.

[55] Frenet, F. Sur les courbes à double courbure. *Journal de mathématiques pures et appliquées* **17**, 437–447 (1852).

[56] Möller, R. & Vardy, A. Local visual homing by matched-filter descent in image distances. *Biological Cybernetics* **95**, 413–430 (2006).

[57] Lyapunov, A. M. The general problem of the stability of motion. *International journal of control* **55**, 531–534 (1992).

[58] Webb, B. The internal maps of insects. *Journal of Experimental Biology* **222**, jeb188094 (2019).

[59] Aso, Y. *et al.* Mushroom body output neurons encode valence and guide memory-based action selection in drosophila. *elife* **3**, e04580 (2014).

[60] Delarboulas, P., Gaussier, P., Caussy, R. & Quoy, M. Robustness study of a multimodal compass inspired from hd-cells and dynamic neural fields (2014).

[61] Sun, X., Yue, S. & Mangan, M. A decentralised neural model explaining optimal integration of navigational strategies in insects. *eLife* **9**, e54026 (2020).

[62] Schoepe, T. *et al.* Finding the gap: Neuromorphic motion-vision in dense environments. *Nature Communications* **15**, 817 (2024).

[63] Sun, X., Fu, Q., Peng, J. & Yue, S. An insect-inspired model facilitating autonomous navigation by incorporating goal approaching and collision avoidance. *Neural Networks* **165**, 106–118 (2023).

[64] Zollikofer, C. P., Wehner, R. & Fukushi, T. Optical scaling in conspecific cataglyphis ants. *Journal of Experimental Biology* **198**, 1637–1646 (1995).

[65] Collett, T. S. & Collett, M. Route-segment odometry and its interactions with global path-integration. *Journal of Comparative Physiology A* **201**, 617–630 (2015).

[66] Aso, Y. *et al.* The neuronal architecture of the mushroom body provides a logic for associative learning. *elife* **3**, e04577 (2014).

[67] Brezovec, B. E. *et al.* Mapping the neural dynamics of locomotion across the drosophila brain. *Current Biology* **34**, 710–726 (2024).

[68] Stone, T. *et al.* An anatomically constrained model for path integration in the bee brain. *Current Biology* **27**, 3069–3085 (2017).

[69] Webb, B. Beyond prediction error: 25 years of modeling the associations formed in the insect mushroom body. *Learning & Memory* **31**, a053824 (2024).

[70] Rybak, J. & Menzel, R. Anatomy of the mushroom bodies in the honey bee brain: the neuronal connections of the alpha-lobe. *Journal of Comparative Neurology* **334**, 444–465 (1993).

[71] Sommer, S., Von Beeren, C. & Wehner, R. Multiroute memories in desert ants. *Proceedings of the National Academy of Sciences* **105**, 317–322 (2008).

[72] Aksoy, V. & Camlitepe, Y. Spectral sensitivities of ants–a review. *Animal Biology* **68**, 55–73 (2018).

[73] Schwarz, S., Mangan, M., Zeil, J., Webb, B. & Wystrach, A. How ants use vision when homing backward. *Current Biology* **27**, 401–407 (2017).

[74] Wystrach, A., Dewar, A., Philippides, A. & Graham, P. How do field of view and resolution affect the information content of panoramic scenes for visual navigation? A computational investigation. *Journal of Comparative Physiology A* **202**, 87–95 (2016).

[75] Applonie, R. & Jin, Y.-F. A novel steering control for car-like robots based on lyapunov stability. *2019 American Control Conference (ACC)* 2396–2401 (2019).

[76] Kruskal, W. H. & Wallis, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* **47**, 583–621 (1952).

[77] Virtanen, P. *et al.* Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods* **17**, 261–272 (2020).

[78] Gattaux, G. G., Wystrach, A., Serres, J. R. & Ruffier, F. Dataset for "route-centric ant-inspired memories enable panoramic route following" (2025). URL https://doi.org/10.6084/m9.figshare.27708105. Research data.

[79] Gattaux, G. G., Wystrach, A., Serres, J. R. & Ruffier, F. Code for "route-centric ant-inspired memories enable panoramic route following" (2025). URL https://doi.org/10.5281/zenodo.15783472. Computer software.

Supplementary Notes for Route-Centric Ant-Inspired Memories Enable Panoramic Route-Following in a Car-like Robot

Gabriel G. Gattaux[1], Antoine Wystrach[2], Julien R. Serres[1,3], Franck Ruffier[1]

[1]Aix Marseille Univ, CNRS, ISM, Marseille, France.
[2]Univ Toulouse, CRCA, CBI, UMR CNRS-UPS 5169, Toulouse, France.
[3]Institut Universitaire de France, IUF, Paris, France.

Supplementary note 1: Left and right familiarities with varying oscillation amplitude

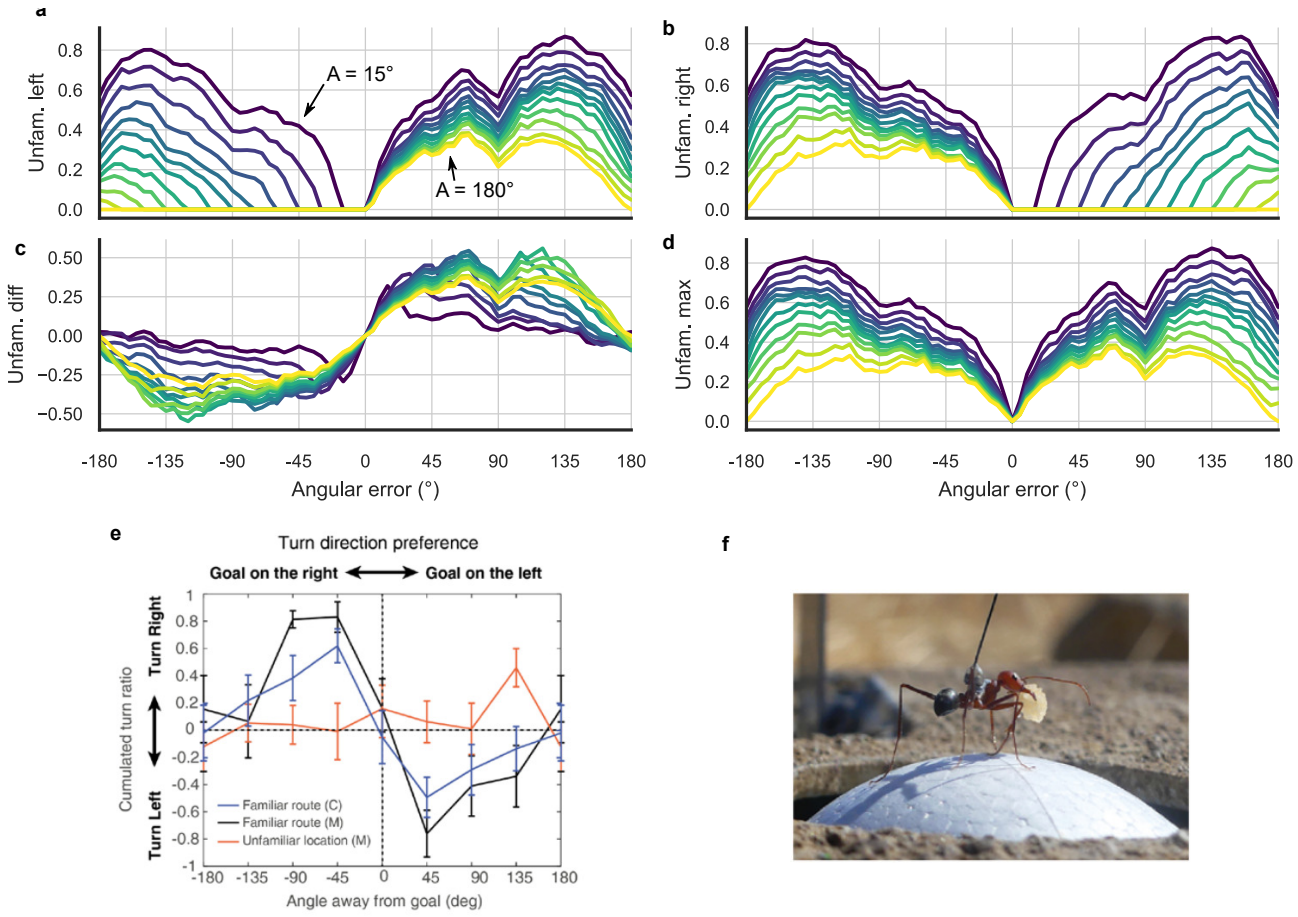


Fig. S1: **Indoor familiarity value relative to the oscillation amplitude and ants experiments**. The experimental setup was conducted indoors with artificial visual cues. **a** Familiarity from the left MBON. **b** Familiarity from the right MBON. **c** Differential familiarity. **d** Maximum familiarity. **e** Turn direction preference of various *Cataglyphis velox* (C) and *Myrmecia croslandi* (M) ants [1]. **f** A *Cataglyphis* ant carrying a cookie while moving on a ball [1].

To assess the self-supervised hypothesis, the images were captured and learned offline along a route (here indoor), and for each image, the familiarity (λ_L and λ_R) were computed for all orientation on the route from the two route MBONs, and plotted against the angular error (see supplementary Figs. S1a and S1b).

The familiarity was plotted for different oscillation amplitude during learning. Interestingly, we observed a similar pattern in the difference of familiarity (λ_{diff}) when the oscillation amplitude was approximately 45 degrees, resembling the turn direction preferences seen in ants [1] (see supplementary Figs. S1c, S1e and S1f).

Supplementary note 2: The advantages of several lateralized memories

The familiarity function when the visual compass is used – when only the direction of the route is learned in one MBON– resembles closely to the maximum familiarity of the route MBONs (see Fig. S1, S2a and S2b).

When using a pure visual compass [2], where one familiarity value was used, the agent had to navigate by minimizing the catchment area (area in the familiarity function where the familiarity drop), and thus recover the route direction. However, since forced scanning in all direction is too much unnecessary calculation, gradient descent approach or klinokinesis were proposed as well [3, 4]. However, errors can occur when the catchment area contains local minima, as in indoor settings here (see Fig. S2d), revealed by the spatial derivative (along the angular error axis) of the familiarity value (See Figs. S2b and S2e). Therefore, the correct sign of the function within the catchment area appeared outdoors (See Fig. S2b) but is not clear indoor (See Fig. S2e), necessitating temporal filtering (in this case, a Savitzky-Golay filter with a window size of 21 inputs). A higher refreshing rate could be required to maintain an adequate control rate after filtering to keep this filtered derivative. Additionally, since the two catchment areas are not identical, the value of the derivative will not be similar, and therefore proportional gains must be fine-tuned for in different environments.

Interestingly, the raw difference between the two routes familiarity MBONs (see Figs. S2c and S2f) produced a function similar to the spatially filtered catchment area. This similarity was consistent across indoor and outdoor settings without the need for additional filtering. This demonstrates the importance of the learning in a lateralized manner, which enabled robust and consistent behavior in several environments without extensive tuning. These findings suggest that using the raw familiarity difference is well-suited for directly inputting into an autonomously controlled system, as it simplifies the process and eliminates the need for complex filtering mechanisms.

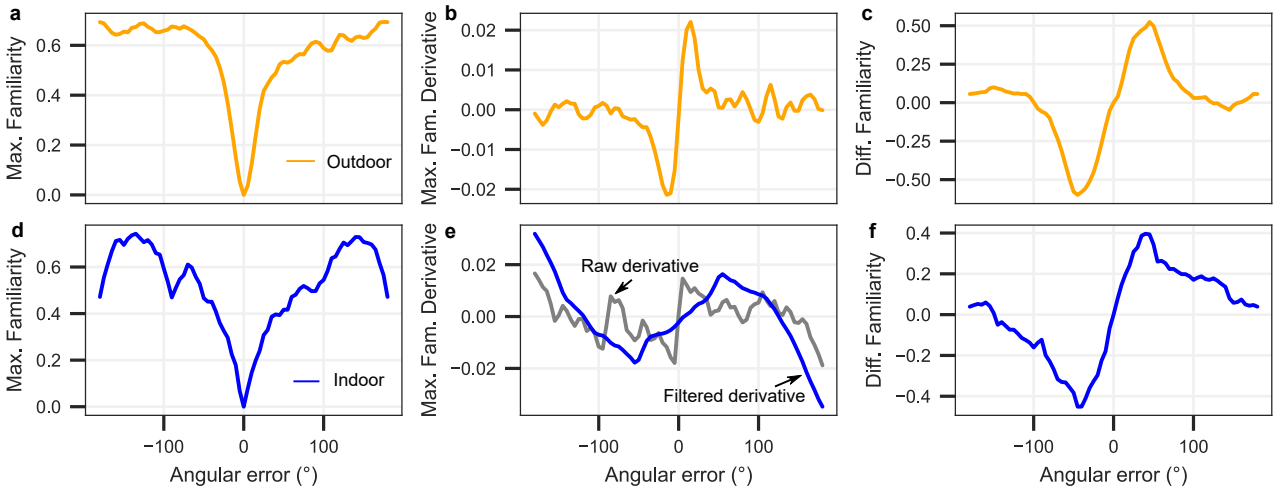


Fig. S2: **Comparison of familiarity derivative over angular error**. **a-c** Outdoor. **d-f** Indoor. **a,d** Maximum familiarity. **b,e** Spatial derivative of the maximum familiarity values. **c,f** Difference between left and right familiarities.

Supplementary note 3: Robot kinematic model in the Frenet frame

This note derives the simplified kinematic model of a car-like robot in the Frenet frame of reference and propose a control input. Unlike unicycle or differential drive robots, a car-like robot employs the Ackermann steering principle to prevent front-wheel slippage (see Fig. S3a). The initial configuration space of the car-like robot is described by $q = (x, y, \theta, \varphi)$, where (x, y) represents the position of point P_r (midpoint of the rear axle), θ is the robot's orientation, and φ is the steering angle defined at point P_a (midpoint of the front axle). The kinematic equations of the car-like robot are given by [5]:

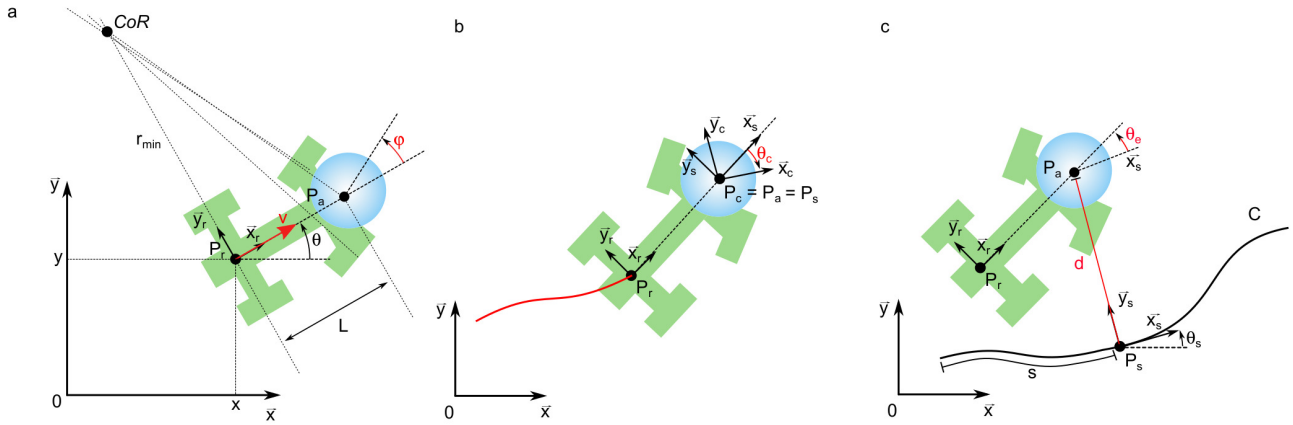


Fig. S3: **Car-like robot kinematics and frame. a** The configuration variables of a car like robot in a global frame of reference. **b** The camera frame relative to the robot frame while learning a route (in red) **c** The coordinate of the robot in the Frenet frame (dynamic local route point).

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ \frac{tan(\varphi)}{L} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \tag{1}$$

where v is the longitudinal velocity (See Fig. S3a), and ω is the steering rate ($\dot{\varphi}$). Assuming the steering rate ω_{max} is high enough to allow instantaneous steering angle changes, the model simplifies to $q = (x, y, \theta)$, resulting in:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cdot \cos(\theta) \\ v \cdot \sin(\theta) \\ v \cdot \frac{\tan(\varphi)}{L} \end{bmatrix}, \tag{2}$$

In practice, the throttle input τ (ranging from 0 to 1) controls the (Pulse Width Modulation) PWM values sent to motors, thus modifying velocity v. For simplicity, we assume v is a function of throttle, $v = f(\tau, v_{max})$, where v_{max} is the maximum speed at full throttle in a straight line. Furthermore non-holonomic constraints prevent the car from moving sideways, resulting in:

$$\dot{x} \cdot \sin(\theta) - \dot{y} \cdot \cos(\theta) = 0 \tag{3}$$

indicating that the robot can only turn if it has a non-zero linear speed.

To address the route-following problem, the Frenet-Serret frame is used as a reference [6]. Let C be a curve with an attached frame $F_s = P_s, \vec{x_s}, \vec{y_s}$, where $\vec{x_s}$ is tangent to C. The motion of P_a with respect to F_s is described by the states $q = (s, d, \theta_e)$, where s is the distance traveled along C, d is the lateral error (distance from P_a to C), and θ_e is the angular error between the robot's orientation and the tangent of the dynamic local route (See Fig. S3). Thus,

$$\theta_e = \theta - \theta_s \tag{4}$$

where θ_s is the dynamic local route orientation in the global frame (See Fig. S3). The motion equations for $\dot{\theta}_e$ are derived from (2) and (4), assuming a momentarily constant route orientation ($\dot{\theta}_s = 0$). This assumption [7] simplifies the calculation of motion equations by assuming that the route orientation does not vary significantly over a small time interval. Thus,

$$\begin{aligned} \dot{\theta}_e &= \dot{\theta} \\ &= v \cdot \frac{tan(\varphi)}{L} \end{aligned} \tag{5}$$

Using the transport theorem, we derive the full set of motion equations in the Frenet frame (as in [6] but simplified here) :

$$\dot{q} = \begin{bmatrix} \dot{s} \\ \dot{d} \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v \cdot (\cos\theta_e - \tan\varphi \sin\theta_e) \\ v \cdot (\sin\theta_e + \tan\varphi \cos\theta_e) \\ v \cdot \frac{\tan\varphi}{L} \end{bmatrix}, \tag{6}$$

The angular error must remain below 90° to avoid losing control due to perpendicular alignment (tangent effect). Based on the empirical results in the main paper, we proposed that the model stayed valid within a 2-meter lateral corridor around the learned route and a 45° angular error, in relation with the learning oscillation amplitude A, leading to:

$$D = \{(d, \theta_e) \mid |d| < 1 \text{ m and } |\theta_e| < 45°\} \tag{7}$$

Within this domain, we proposed a proportional relationship between the familiarity difference λ_{diff} and angular error θ_e thanks to the observed linearity in the main paper:

$$K_{diff} \cdot \lambda_{diff} = -\theta_e \tag{8}$$

where K_{diff} is a negative gain. Then, we propose the following control input U:

$$U = \begin{bmatrix} v \\ \varphi \end{bmatrix} = \begin{bmatrix} M \cdot K_v \cdot sat(1 - \lambda_{max}) \\ M \cdot K_\varphi \cdot \lambda_{diff} \end{bmatrix}, \tag{9}$$

Supplementary note 4 : Lyapunov stability analysis

To evaluate the stability of the overall system, we introduce a Lyapunov function [8] that is positive definite. The function $V(d, \theta_e)$ is constructed such that it is positive definite ($V(d, \theta_e) \geq 0$) for all $d \neq 0$ and $\theta_e \neq 0$, and it equals zero only when $d = 0$ and $\theta_e = 0$. The proposed Lyapunov function is:

$$V(d, \theta_e) = \frac{1}{2}\left(d^2 + \theta_e^2\right) \tag{10}$$

For the system to be considered asymptotically stable, the derivative of this Lyapunov function, $V(\dot{d}, \theta_e)$, must be negative for all $d \neq 0$ and $\theta_e \neq 0$, and zero when $d = 0$ and $\theta_e = 0$. This implies that the system not only remains near equilibrium but eventually converges to it . The derivative of the Lyapunov function is:

$$\dot{V}(d, \theta_e) = d\dot{d} + \theta_e\dot{\theta_e} \tag{11}$$

Then, by integrating the model' equation 2 to the derivative Lyapunov function 11 and the control actions equation 9 for φ (we set $v = 1$ for simplification, knowing that v will be always positive or negative but without change on a route-following task) and fixed the motivation state $M = 1$, we get :

$$\begin{aligned} \dot{V}(d, \theta_e) &= dv(\sin\theta_e + \tan\varphi\cos\theta_e) + \theta_e\frac{\tan\varphi}{L} \\ &= d(\sin\theta_e + \tan(K_\varphi\lambda_{diff})\cos\theta_e) \\ &+ \theta_e\frac{\tan(K_\varphi\lambda_{diff})}{L} \end{aligned} \tag{12}$$

Then, we apply the proposition depicted in equation 8, we obtain :

$$\begin{aligned} \dot{V}(d, \theta_e) &= d(\sin\theta_e + \tan -\theta_e\cos\theta_e) + \theta_e \cdot \frac{\tan -\theta_e}{L} \\ &= d(\sin\theta_e + \frac{\sin -\theta_e}{\cos -\theta_e}\cos\theta_e) + \theta_e \cdot \frac{\tan -\theta_e}{L} \\ &= d(\sin\theta_e + \sin -\theta_e) + \theta_e\frac{\tan -\theta_e}{L} \\ &= d(\sin\theta_e - \sin\theta_e) + \theta_e\frac{\tan -\theta_e}{L} \\ &= -\theta_e \cdot \frac{\tan\theta_e}{L} \end{aligned} \tag{13}$$

The θ_e sign will cancel itself out if it is within plus or minus 90 degrees, due to the tangent function and the similar sign of λ_{diff} relative to θ_e. Consequently, the negative sign of the resulting derivative of the Lyapunov function in (13) satisfies the conditions $\dot{V}(d, \theta_e) < 0$ for all $d \neq 0$ and $\theta_e \neq 0$, and $\dot{V}(0, 0) = 0$ within the domain limits specified in (7). Therefore, the system is proven to be asymptotically stable within these conditions.

Below is a concise, coherent presentation of the entire image-processing pipeline. We use an enumerated list and place the key "calculation" formulas at the start of each item, followed by brief explanatory text. The pipeline takes an input image I and outputs the final vector PN.

Supplementary note 5: Image Processing Steps

Given a digital RGB image

$$I \in \mathbb{R}^{r\times r\times 3},$$

where $\{1,\dots,r\}\times\{1,\dots,r\}$ spans the spatial coordinates and the third dimension corresponds to the red, green, and blue color channels, the goal is to generate a Projection Neuron vector PN suitable for subsequent processing in the Mushroom Bodies (MBs).

1. **Green Channel Extraction**

$$I_g(i,j) \;=\; I(i,j,2),$$

for $1 \le i,j \le r$.

This isolates the green channel from the RGB image I. Consequently, $I_g \in \mathbb{R}^{r\times r}$ contains the intensity of the green component at each pixel (i,j).

2. **Gaussian Smoothing**

$$I_G \;=\; I_g \;*\; Gg(x,y), \quad \text{where} \quad Gg(x,y) \;=\; \frac{1}{2\pi\,\sigma^2}\,\exp\!\Big(-\tfrac{x^2+y^2}{2\sigma^2}\Big).$$

We choose $\sigma = 3$, and the Gaussian kernel size is

$$\text{ksize} \;=\; 2 \times \big\lceil 3\times\sigma \big\rceil \;+\; 1.$$

Convolving I_g with a 2D Gaussian kernel of standard deviation σ reduces high-frequency noise and smooths the image. This step enhances subsequent edge extraction.

3. **Subsampling**

Let $rs = 32$. Define the scale factor

$$\alpha \;=\; \frac{r}{rs}.$$

Then construct $I_s \in \mathbb{R}^{rs\times rs}$ by nearest-neighbor sampling:

$$I_s(i,j) \;=\; I_G\big(\text{round}\big[\alpha\,(i-1)\big]+1,\, \text{round}\big[\alpha\,(j-1)\big]+1\big), \quad 1 \le i,j \le rs.$$

This maps the smoothed image I_G of size $r\times r$ to a smaller image I_s of size $rs\times rs$ (here, 32×32). Using round$(\cdot)$ ensures each new pixel (i,j) comes from the closest pixel in I_G.

4. **Edge Extraction**

$$I_c \;=\; \sqrt{\big(G_x * I_s\big)^2 + \big(G_y * I_s\big)^2},$$

where the Sobel kernels are

$$G_x \;=\; \begin{bmatrix} -1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1 \end{bmatrix}, \quad G_y \;=\; \begin{bmatrix} -1 & -2 & -1\\ 0 & 0 & 0\\ 1 & 2 & 1 \end{bmatrix}.$$

Convolving I_s with Sobel filters in the horizontal and vertical directions approximates the image gradient, thus emphasizing edges. This is analogous to lateral inhibition in the optic lobes.

5. **Flattening**

Let $m = \frac{rs+1}{2}$ and define the circular region

$$C \;=\; \{(i,j) \mid (i-m)^2 + (j-m)^2 \;\le\; r^2\}.$$

Enumerate its coordinates from top-left to bottom-right: $\{(i_k,j_k)\}_{k=1}^{n}$. Then form

$$PN(k) \;=\; I_c\big(i_k,j_k\big), \quad k = 1,\dots,n,$$

where $n \approx 800$.

This step merges the circular region (radius $r = 32$) of the edge-detected image into a one-dimensional vector $PN \in \mathbb{R}^n$. The circular cropping and flattening offer a simplified, biologically inspired encoding of spatial information.

After these five steps, the final PN-vector serves as input to the subsequent MB processing.

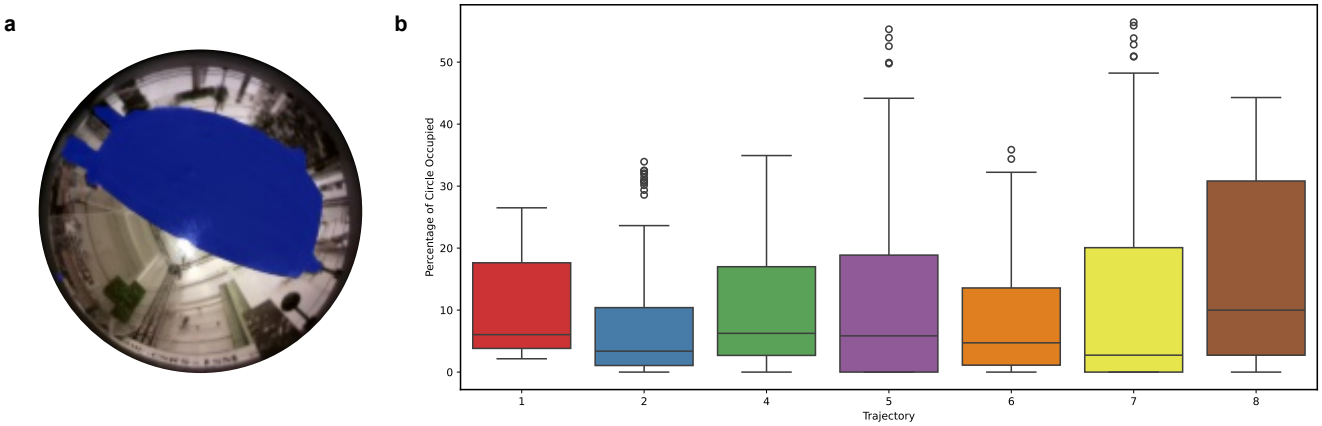


Fig. S4: **Detailed occlusions sensitivity**. To define the nature and sensitivity to dynamics occlusions, we used Yolo v11m "segment" [9]) and we trained it on a manually segmented 75 images, the segmentation covered the plank that was the field of view, see Supplementary Video. Then, we predict for all our trajectories and images the mask for the plank **a**, and we defined the percentage of occlusions by the percentage of the predicted mask on the total image area **b**.

Supplementary note 6: Extended experimental data

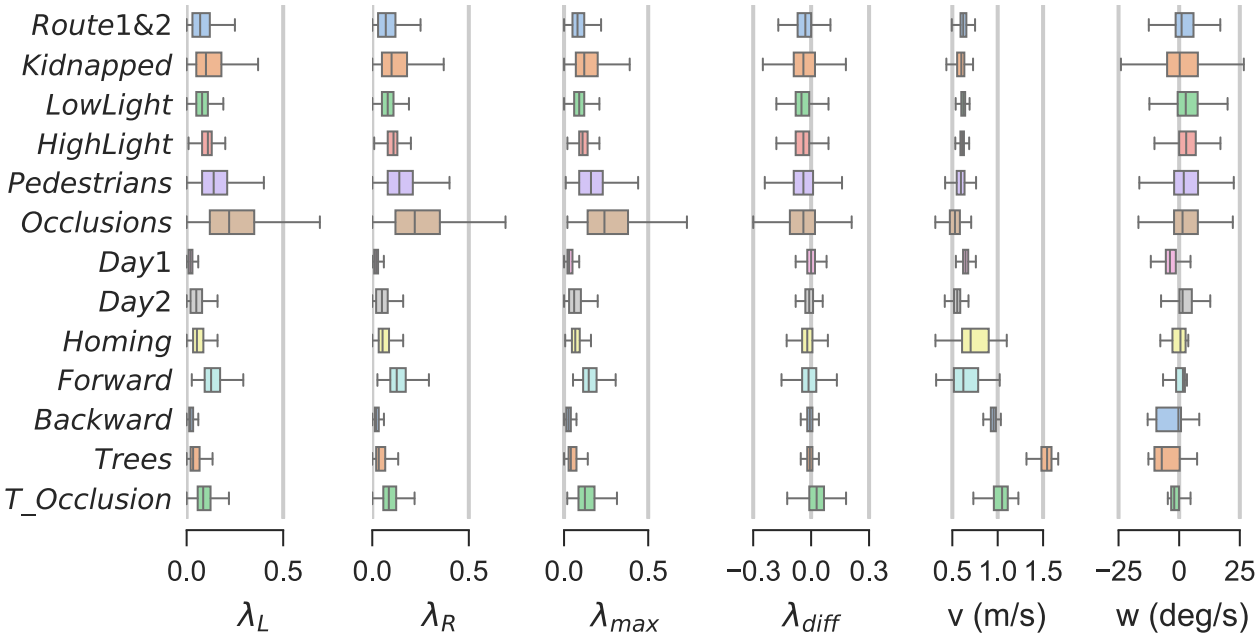


Fig. S5: **Detailed variables during experiments**. Boxplots representing the left and right familiarity (λ_L and λ_R), the difference (λ_{diff}), the maximum λ_{max}, the linear speed (v), and the angular speed (w) for each experiment.

The overview of the left and right familiarities, with the difference, the maximum, the linear and angular velocities are depicted in Fig. S5 and in the Table S1. Furthermore, the crash which occurred during kidnapped robot and during occlusion are shown in Fig. S6, with two other indoor route following experiments not shown in the main paper but taken into account in *Route*1&2 statistics. The poses learned during shuttling and homing experiments are also shown in Fig. S7 with the endpoint familiarity plotted in the background.

Table S1: Summary statistics by experiment.

Experiment ID	Lateral Error (m)		Angular Error (°)		Diff. Familiarity	
	Median	MAD	Median	MAD	Median	MAD
Route 1&2	0.22	0.10	3.45	2.25	-0.03	0.04
Kidnapped	0.27	0.14	6.46	4.19	-0.04	0.05
Low Light	0.24	0.08	3.81	2.09	-0.05	0.03
High Light	0.20	0.06	3.88	2.33	-0.04	0.03
Pedestrians	0.27	0.15	4.05	2.82	-0.04	0.05
Occlusions	0.23	0.13	4.72	3.29	-0.04	0.06
Day1	0.40	0.13	5.77	2.79	-0.00	0.01
Day2	1.32	0.47	6.19	3.22	-0.01	0.01
Homing	0.89	0.54	6.28	4.16	0.03	0.04
Forward	0.11	0.04	1.26	0.84	0.00	0.02
Backward	0.19	0.08	2.73	2.16	-0.01	0.02
Trees	0.28	0.12	4.00	2.36	-0.02	0.02
T_Occlusions	0.38	0.11	5.64	3.09	-0.01	0.04

Experiment ID	Max Familiarity		Angular Speed (°/s)		Linear Speed (m/s)	
	Median	MAD	Median	MAD	Median	MAD
Route1&2	0.08	0.03	1.00	3.46	0.62	0.03
Kidnapped	0.12	0.06	0.22	6.32	0.60	0.04
Low Light	0.09	0.03	2.72	4.01	0.62	0.02
High Light	0.11	0.03	2.89	3.24	0.61	0.02
Pedestrians	0.16	0.07	1.89	4.60	0.59	0.04
Occlusions	0.24	0.11	1.39	4.75	0.53	0.06
Day1	0.02	0.01	-5.90	5.86	0.98	0.02
Day2	0.04	0.02	-7.25	3.67	1.55	0.05
Homing	0.10	0.03	-2.65	0.94	1.08	0.05
Forward	0.03	0.01	-3.80	1.97	0.65	0.03
Backward	0.06	0.03	1.42	1.91	0.55	0.03
Trees	0.07	0.02	0.53	2.41	0.70	0.13
T_Occlusions	0.15	0.04	1.56	0.97	0.32	0.14

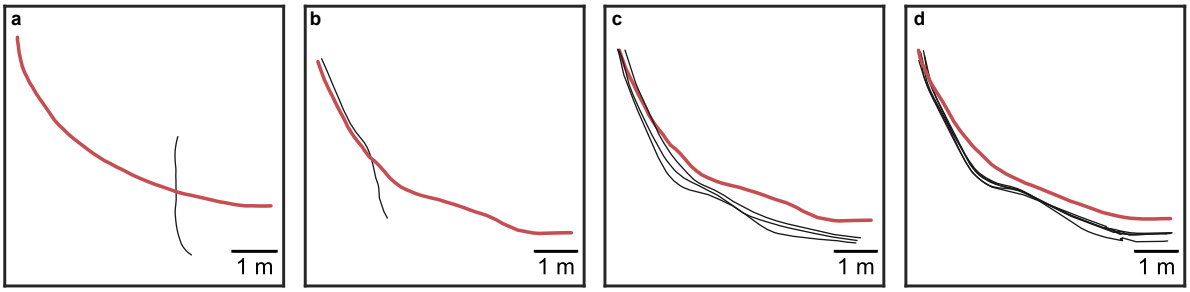


Fig. S6: **Supplementary trajectories. a** The crash during the kidnapped robot experiment. **b** Crash during dynamic occlusion. **c** Three trajectories, in normal conditions before pedestrians experiment, grouped in *Route*1&2. **d** Five trajectories in normal conditions grouped in *Route*1&2 cases.

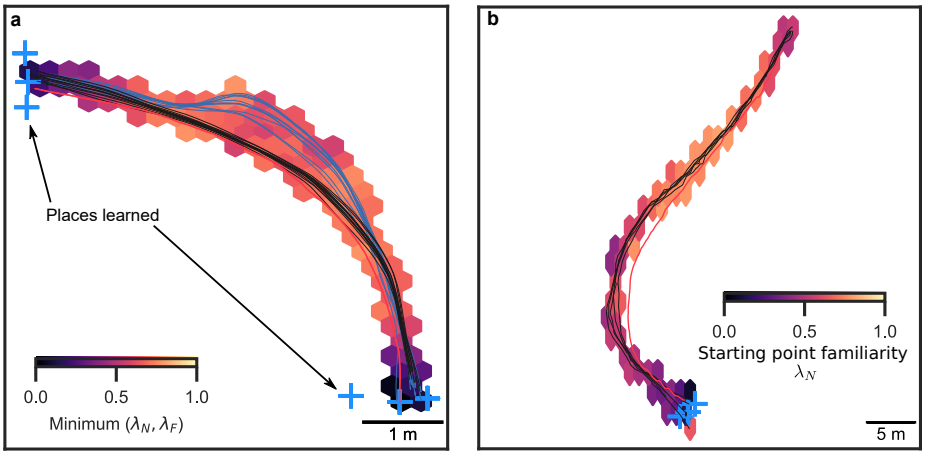
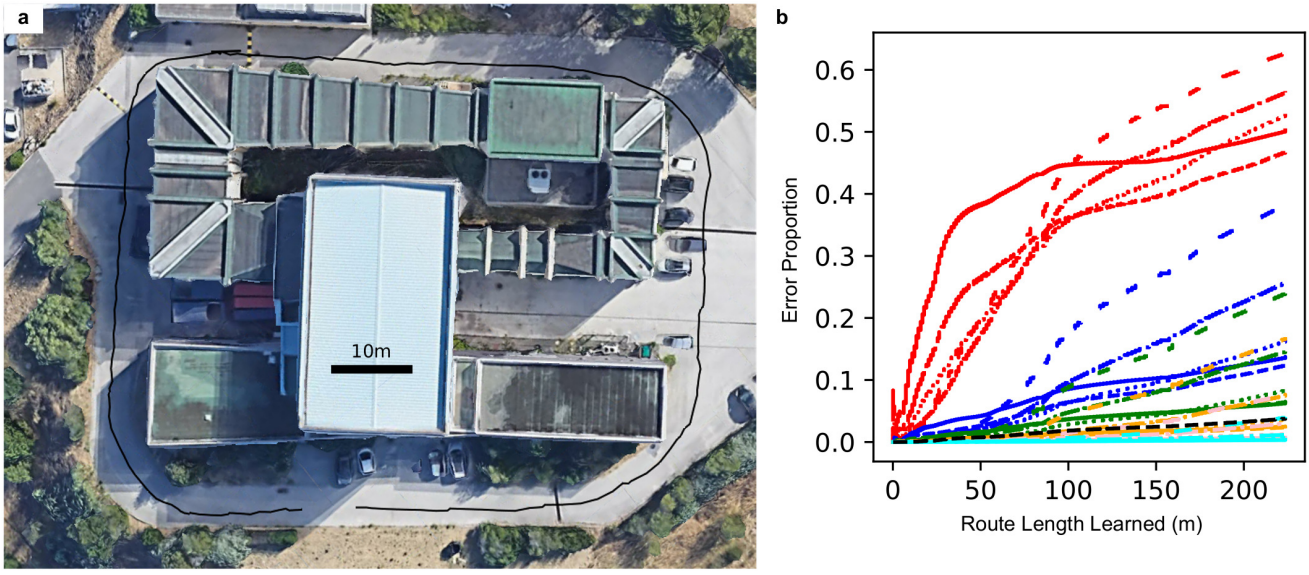


Fig. S7: **Endpoint learning in shuttling and homing experiments. a** Shuttling experiments: the background shows the minimum familiarity between the two endpoint's MBONs (Nest and Feeder), with blue crosses indicating the learned poses. **b** Homing experiments: Nest MBONs familiarity in background and trajectories in foreground.



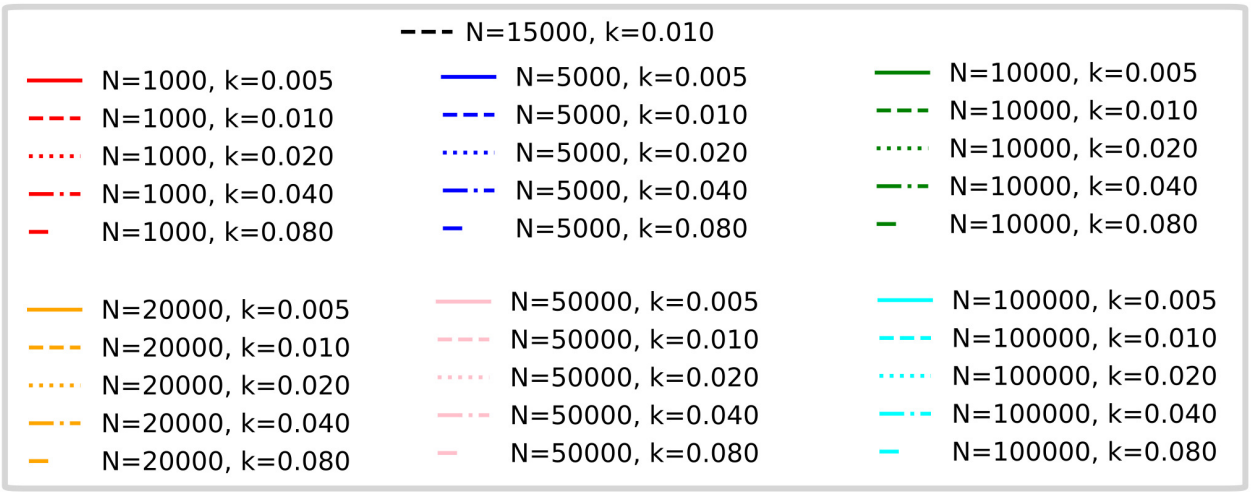


Fig. S8: **Dataset gathered in square around the laboratory**. **a** Google earth view of the trajectory for the 250m route dataset. **b** Curve of the Error proportion according to the Route Length Learned with the detailled legend below.

Supplementary References

[1] Wystrach, A., Le Moël, F., Clement, L. & Schwarz, S. A lateralised design for the interaction of visual memories and heading representations in navigating ants. *Preprint BioRxiv* (2020).

[2] Gattaux, G., Vimbert, R., Wystrach, A., Serres, J. R. & Ruffier, F. Antcar: Simple Route Following Task with Ants-Inspired Vision and Neural Model. *Preprint HAL* (2023).

[3] Möller, R. & Vardy, A. Local visual homing by matched-filter descent in image distances. *Biological Cybernetics* **95**, 413–430 (2006).

[4] Kodzhabashev, A. & Mangan, M. Wilson, S. P., Verschure, P. F., Mura, A. & Prescott, T. J. (eds) *Route Following Without Scanning*. (eds Wilson, S. P., Verschure, P. F., Mura, A. & Prescott, T. J.) *Biomimetic and Biohybrid Systems*, Vol. 9222, 199–210 (Springer International Publishing, Cham, 2015).

[5] Lynch, K. *Modern Robotics* (Cambridge University Press, 2017).

[6] Morin, P. & Samson, C. Motion control of wheeled mobile robots. *Springer handbook of robotics* **1**, 799–826 (2008).

[7] Applonie, R. & Jin, Y.-F. A novel steering control for car-like robots based on lyapunov stability. *2019 American Control Conference (ACC)* 2396–2401 (2019).

[8] Lyapunov, A. M. The general problem of the stability of motion. *International journal of control* **55**, 531–534 (1992).

[9] Jocher, G., Qiu, J. & Chaurasia, A. Ultralytics YOLO (2023). URL https://github.com/ultralytics/ultralytics.