AntFlie: Frugal Visual Teach and Repeat on Narrow FoV Micro-Drones

Gabriel G Gattaux, Hamidou Diallo, Julien R Serres, Antoine Wystrach, Franck Ruffier

AntFlie: Frugal Visual Teach and Repeat on Narrow FoV Micro-Drones

Gabriel G. Gattaux[1*], Hamidou Diallo[2], Julien R. Serres[1,3], Antoine Wystrach[4] and Franck Ruffier[1,5]

***Abstract*— We present AntFlie, an ant-inspired Visual Teach-and-Repeat (VT&R) framework demonstrated on a 33-gram, 9-cm MAV equipped with an ultra-low-resolution (24×24 px) camera and narrow 87° field of view (FoV). During a one-shot teach flight along an outbound route, the MAV performs periodic physical scans and uses a local compass based on inertial and optic flow cues to categorize views as left or right relative to the path, storing compact, lateralized visual memories in a Mushroom Body (MB) neural network with a footprint under 4 kB. In the repeat phase, the MAV flies the inbound route by retracing the outbound path, and autonomously lands at its home location using only visual familiarity through direct sensorimotor coupling, rather than map-based reasoning. Offline simulations show that the Route Lateralized (RoLa) algorithm in AntFlie matches the accuracy of a state-of-the-art insect visual compass (ViCo) while running up to 20× faster and supporting narrow FoVs. Real-world indoor experiments further demonstrate 24 autonomous inbound repeats totaling 110 meters of flight, with a 13-cm median lateral error and 34-cm landing precision on average. These results highlight the feasibility of frugal, bio-inspired, vision-only navigation for MAVs operating under strict size, weight, power, and cost (SWaP-C) constraints, inspired by the navigation of *Cataglyphis* and *Melophorus* ants.**

***Index Terms*— Aerial Systems: Perception and Autonomy, Micro/Nano Robots, Biologically-Inspired Robots, Vision-Based Navigation, Insect vision**

I. Introduction

Micro Aerial Vehicles (MAVs) hold promise for tasks like subterranean exploration, ecological monitoring, or last-mile delivery, but their ultralight designs impose strict size, weight, power and cost (SWaP-C) constraints [1], [2]. These limitations rule out heavy sensors such as LiDAR, making vision-based navigation more attractive due to its compactness and efficiency. In GPS-denied missions, homing (the ability to autonomously return to the starting point) is critical, especially when the outbound path offers the safest return route, as in Martian lava tubes exploration [3]. Visual Teach-and-Repeat (VT&R) algorithms can achieve this by recording landmarks during an outbound *teach* phase and using them to retrace the path during a *repeat* phase, forming a visual Ariadne's thread [4], [5].

However, typical VT&R pipelines rely on high-resolution imagery, stereo cameras, and compute-heavy algorithms such as Visual Odometry (VO), visual Simultaneous Localization and Mapping (vSLAM), or Visual Place Recognition (VPR) [6], [7], which are unsuitable for MAVs lighter than 50 grams.

In parallel, insects like ants and bees navigate long routes through complex environments using low-resolution vision and minimal neural resources [8]. Their strategies rely on egocentric visual familiarity rather than localization, answering "Where am I going?" rather than "Where am I?" [9]. Inspired by this, biologically grounded VT&R alternatives have emerged that avoid explicit mapping and localization [10].

In this letter, we extend the Route Lateralized (RoLa) framework previously tested on a ground robot with panoramic camera [11] to support arbitrary camera field of view (FoV), without requiring calibration or panoramic vision. Experiments were preformed with AntFlie, a 33-gram MAV quadcopter under SWaP-C constraints, operating with a narrow 87° FoV and ultra-low-resolution (24×24 px) vision. This lightweight and biologically inspired VT&R (see Fig. 1) is divided into two phases. During the outbound (teach) phase, AntFlie performs periodic scans and uses onboard Inertial Measurement Unit (IMU) and optic flow to categorize views as "left" or "right" of the route forming the local compass, encoding compact route-lateralized visual memories in a Mushroom Body (MB) neural network, with a memory footprint under 4 kB. The local compass provides heading computation relative to the local route frame. In the inbound (repeat) phase, navigation is achieved purely through visual familiarity as a feedback signal, without maps, localization, or odometry.

The main contributions are: (1) Extension of RoLa framework to arbitrary camera FoVs by using a local compass, removing the need for panoramic input; (2) Real-world MAV demonstration of familiarity-based VT&R with autonomous landing under extreme SWaP-C constraints, including a Lyapunov stability experimental proof; and (3) Benchmark against ViCo, showing comparable accuracy while running up to 20 times faster and better supporting narrow FoVs. The experimental data are available online[1] , the visualization code is provided in the supplementary material, and a video is available online[2]

*G.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.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).

[1]Aix Marseille Univ, CNRS, ISM, Marseille, France gabriel.gattaux@univ-amu.fr

[2]INRIA and University of Lyon, CITI, INSA Lyon, France hamidou.a.diallo@inria.fr

[3]Institut Universitaire de France, IUF, Paris, France julien.serres@univ-amu.fr

[4]Univ Toulouse, CRCA, CBI, UMR CNRS-UPS 5169, Toulouse, France antoine.wystrach@cnrs.fr

[5]ENSTA|IP Paris, CNRS, Lab-STICC, Brest, France franck.ruffier@cnrs.fr

[1]https://doi.org/10.57745/AC9ILI

[2]https://youtu.be/hQBL9Npc6cQ

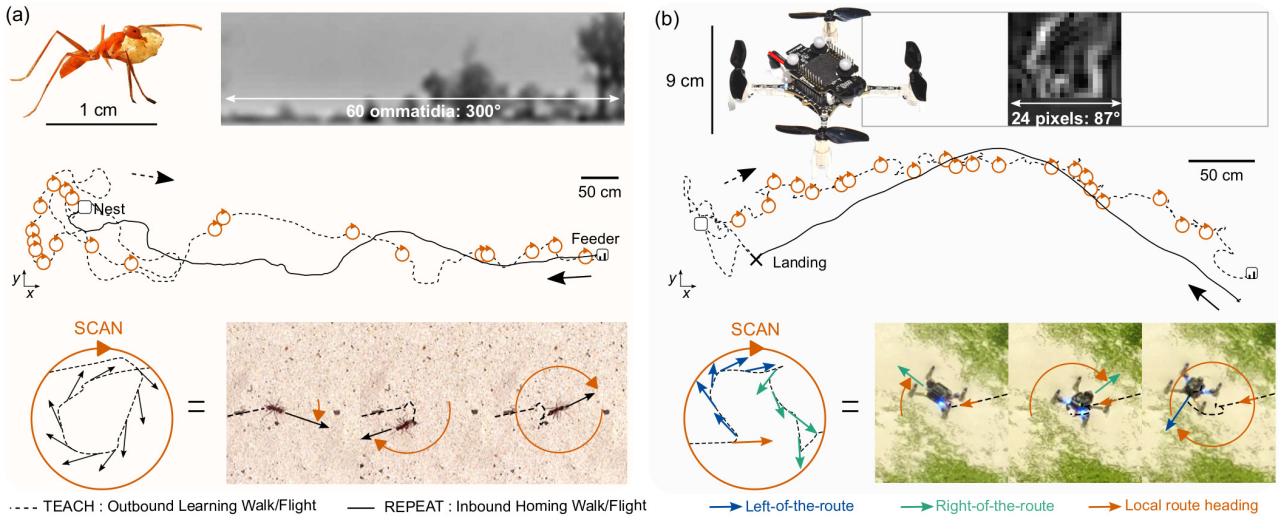

Fig. 1. Bio-inspired homing via outbound scans and route-lateralized visual memories bird-eye views. (a) Desert ant navigation (adapted from [12], [13]). (b) AntFlie quadcopter mimics this with low-res, narrow-FoV scans for learning and visual familiarity for homing/landing.

II. Related Work

VT&R methods fall broadly into two categories: map-based and mapless. Map-based approaches rely on explicit state estimation and localization, whereas mapless approaches use direct image-driven or reactive control, often based on visual familiarity.

A. Map-Based VT&R

Global Metric Maps: These methods build globally consistent pose graphs using vSLAM or VO. Mahdavian et al. [14] implemented VT&R with ORB-SLAM and YOLOv3 on a ground robot, requiring stereo vision and GPU. While accurate, such solutions are computationally heavy. Nourizadeh et al. [15] proposed a lightweight alternative using odometry and visual offsets to maintain a global pose. However, it lacks robustness to kidnapped-robot scenarios and remains highly wheel odometry-dependent.

Local Metric Maps: These approaches constrain localization to a taught trajectory. Furgale and Barfoot [4] built overlapping local submaps but required stereo vision and 358.4 kB/m storage. For MAVs, Warren et al. [5] and Pfrunder et al. [16] demonstrated VO-based VT&R using stereo and monocular vision respectively, but relied on calibrated cameras and GPU processing, making them unsuitable for ultra-light MAVs.

Topological Maps: Topological VT&R represents environments as graphs of visual places, avoiding full metric accuracy [17]. Early systems by Matsumoto et al. [18] relied on view databases and local image interpolation to generate motion commands. While efficient, they required omnidirectional or wide FoV cameras. These systems also scaled poorly in memory and often needed initial pose alignment or route indexing. Modern approaches use deep VPR: Sun et al. [19] employed self-supervised monocular descriptors for day-night matching, while Camara et al. [20] combined CNN-based features with image offsets for localization and heading correction. However, these methods depend on 1080p input, pre-trained networks, and GPU compute, unsuitable for ultra-light MAVs.

B. Mapless VT&R

Mapless approaches avoid explicit localization by relying on egocentric visual cues for reactive control. Krajník et al. [21] proposed a system replaying recorded velocities and correcting heading via image offsets, while Image Based Visual Servoing (IBVS) methods like Caron et al.'s photometric control law [22] use image gradients directly for control—but demand high-resolution input and heavy computation, or velocity profile encoding, making them unsuitable for ultra-light MAVs.

Insect-Inspired Navigation. Ants and bees navigate long distances using low-resolution vision and minimal neural resources, inspiring biologically grounded mapless VT&R. The Insect Visual Compass model (ViCo) [23], [24] stores panoramic outbound views and performs in silico scans during repeat, but suffers from high computational cost and cumulative storage. Van Dijk et al. [25] implemented a Fourier-based familiarity method on a MAV. However, it used a panoramic camera (adding $\approx 48\%$ mass) and relied on odometry, which contrasts with evidence that insects navigate via egocentric (retinotopic), orientation-specific visual familiarity rather than whole-image signatures or odometry [26]. To reduce storage, insect MB models, which are sparse associative memories for visual familiarity [27], were used alongside ViCo [28]. However, they remain limited to ground robots, require panoramic vision and scanning at each step. While aerial adaptations with downward-facing views [29] or tree detection [30] offer partial solutions; real-world VT&R in MAVs remains unachieved.

Prior Work. Inspired by studies showing that the Central Complex (CX) in insect brains (also in desert ant) modulates memory encoding in the MB based on the nest's relative direction [31], [32], the **RoLa (Route-Lateralized) model** was previously introduced using panoramic images [11]. The

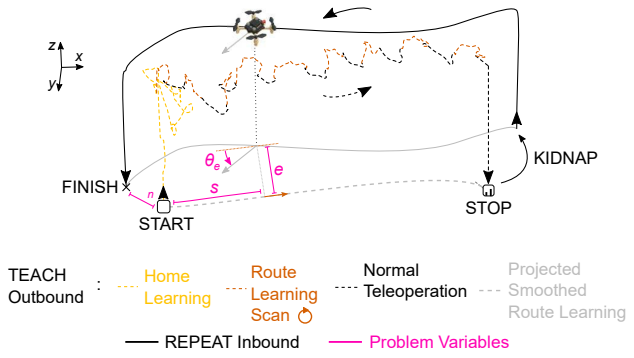

Fig. 2. Problem statement overview. After the outbound flight, AntFlie is displaced and attempts to autonomously return while minimizing lateral (e), angular (θ_e), and along-track (s) error, reducing final landing error (d_n).

approach consisted in categorize views as "left" or "right" relative to the route direction, enabling egocentric route following on a ground robot equipped with panoramic vision and in silico scans during the learning phase.

Overall, most VT&R approaches are typically evaluated on ground robots, either with GPU computation, using panoramic cameras, or rely heavily on odometry during repeat phase. Note that transitioning from panoramic views with virtual scans on a ground robot to a physically constrained MAV with a narrow FoV remains a challenge. For this reason, we introduce a compact VT&R framework enabling robust inbound route following and autonomous landing on a ultra-low-resolution narrow-FoV MAV (AntFlie, Fig. 1) using minimal memory.

III. Methodology

A. Problem Statement

We address a route-based visual homing task in which a MAV must autonomously return to its starting point after a single outbound, teleoperated "teach" flight. During teaching, the MAV forms visual memories of (1) the nest area and (2) the outbound route. Before the return ("repeat") flight, the MAV is displaced, creating a kidnapped-robot scenario, and must navigate home using only vision, without maps, localization, nor global positioning.

The objective is to minimize the final landing error d_n, defined as the Euclidean distance between the takeoff and autonomous landing points (Fig. 2). In the Frenet–Serret frame of the outbound path, this requires reducing lateral error e, angular error θ_e, and longitudinal error s. If the MAV re-enters the taught path near its original heading, a bearing-only controller minimizing θ_e can stabilize e, while constant forward velocity and robust end-route detection can reduce s and thus d_n.

We assess stability via the Lyapunov candidate function:

$$V(t) = \tfrac{1}{2}\left(K_h\theta_e^2 + K_e e^2 + K_s s^2\right), \qquad (1)$$

where positive gains K_h, K_e, and K_s weight heading, cross-track, and along-track errors, here fixed to 1 for simplicity. We assume that convergence of $V(t)$ over time indicates stable homing.

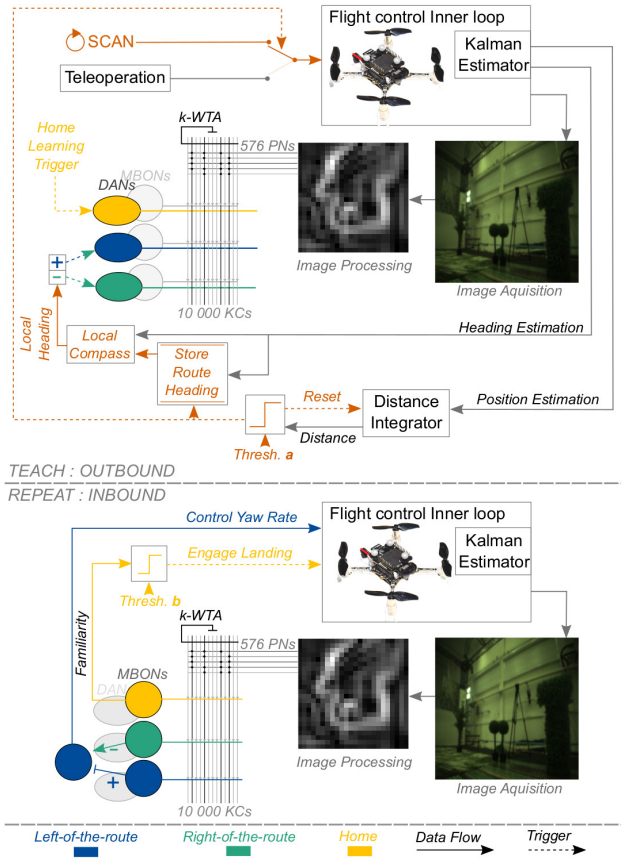

Fig. 3. AntFlie's bio-inspired visual homing framework. During the teleoperated Teach phase, a home memory is externally triggered, followed by automatic 360° scans for route memory encoding aligned with the local route direction. Left/right views are stored in lateralized MBONs via dopaminergic modulation. In Repeat, left/right familiarity steers yaw and home familiarity triggers landing.

B. Route Lateralized (RoLa): Image Processing & Encoding

The image processing pipeline (Fig. 3) follows the biologically inspired approach described in [11]. Its objective is to transform raw visual input into a high-dimensional, sparse binary representation, mimicking the visual pathway from the insect optic lobes to the MB.

Each views are encoded into a sparse, high-dimensional binary code inspired by the insect MB (Fig. 3). A 87° FoV grayscale image (green channel, 324×324 px) is Gaussian-blurred, downsampled to 24×24 px, and edge-enhanced with Sobel filter. The result is flattened to a 576-D Projection Neuron (PN) vector. PNs project to N=10,000 Kenyon Cells (KCs) via a fixed binary matrix $PNtoKC \in \{0,1\}^{10{,}000\times 576}$ with 4 PN inputs per KC, producing $EPSP = PNtoKC \cdot PN$. A k-WTA with k=1% keeps the top N_u=100 KCs, yielding a 10,000-bit activation vector with exactly 100 ones named Action Potential (AP). This fixed-size code underlies image encoding and sparsification to be learned into home or routes memories (Teach Phase) or compared as a familiarity index (Repeat Phase).

C. Route Lateralized (RoLa): Teach Phase

The outbound Teach phase consists of two externally selected subphases: Home Learning and Route Learning (see Fig. 2 and Fig. 3).

1) Home Learning: Immediately after takeoff, Home Learning is triggered via an external dopaminergic signal (DAN), designating a specific MB Output Neuron (MBON) to encode the "home" location. Synaptic plasticity is implemented through an anti-Hebbian depression rule, whereby co-active synapses between Kenyon Cells (KCs) and the selected MBON are weakened. Each MBON (home or route) maintains a binary synaptic weight vector $KCtoMBON \in \{0,1\}^{10,000}$, initialized to 1 (fully connected). Given the current KC activation vector AP, learning is performed as:

$$KCtoMBON_i = \begin{cases} 0, & \text{if } AP_i = 1 \\ KCtoMBON_i, & \text{otherwise} \end{cases} \quad (2)$$

Where i is the neuron and synapse number. This binary synaptic update encodes the current visual scene by removing connections from the active KCs.

2) Route Learning: Route Learning applies the same anti-Hebbian mechanism to two additional MBONs, encoding views to the *left* and *right* of the route direction. While the transition between Home and Route Learning is externally triggered, the assignment to left or right MBONs during Route Learning is self-supervised during periodic physical scans. These scans, inspired by desert ant behavior (Fig. 1), consist of yaw rotations of fixed amplitude (either full 360° or $\pm 45°$ here). The goal is to categorize each view as either left- or right-facing relative to the current route direction.

In prior work [11], this classification was computed from in silico scan of panoramic images under the assumption that the robot moved in the route direction. Here, due to the MAV's narrow FoV, this assumption no longer holds. Instead, we implement an onboard local compass to compute the heading error θ_e, which drives DAN-based route memory categorization during learning.

A local compass to categorize left/right views: The so-called **local compass** employs a Kalman filter to fuse IMU, ventral optic flow, and time-of-flight measurements, enabling (i) estimation of the MAV's yaw and (ii) distance gauging in the local route frame during scans by integrating the estimated ground speed, thereby performing local dead-reckoning odometry. A scan is triggered every forward displacement of $a = 30\,\mathrm{cm}$, storing the yaw as the reference route direction; the instantaneous estimated heading is compared to this value: $\theta_e = \theta_{\text{current}} - \theta_{\text{route}}$. Based on the sign of θ_e, the algorithm determines whether the current view is facing left or right for categorization : it applies the corresponding DAN signal to update the associated MBON via the same rule in Eq. 2. Periodic triggering and reset based on traveled distance, coupled with onboard heading estimation, reduce the effects of odometric drift during Route Learning thanks to local estimations.

D. Route Lateralized (RoLa): Repeat Phase

During the Repeat phase, the current visual input is processed into a Kenyon Cell activation vector (AP), which is then compared against all three memory traces encoded in the MBONs: left ($KCtoMBON_L$), right ($KCtoMBON_R$), and home ($KCtoMBON_H$).

A familiarity score $\lambda \in [0,1]$ is computed as the normalized count of active KCs that remains connected to a given MBON. This measure captures visual similarity between the current scene and a stored memory, such as:

$$\lambda = \frac{1}{N_u} \sum_{i=1}^{N} AP_i \cdot KCtoMBON_i \quad (3)$$

where N is the total number of Kenyon Cells and $N_u = \sum AP_i$ is the number of active KCs. A value of $\lambda = 0$ indicates perfect familiarity (all active KCs were previously depressed), while $\lambda = 1$ indicates complete novelty. To steer the MAV, a familiarity difference λ_{diff} index is computed between the left and right route memories:

$$\lambda_{\text{diff}} = \lambda_L - \lambda_R \quad (4)$$

For all repeat experiments, the linear velocity was fixed to 20 cm/s. The yaw rate command signal ω is computed as follows:

$$\omega = k \cdot \lambda_{\text{diff}} \quad (5)$$

Where k is a proportional gain. Finally, when the familiarity score of the home MBON (λ_H) falls below a fixed threshold $b = 0.1$, the system initiates autonomous landing. This mechanism ensures that the MAV recognizes the learned home vicinity based solely on visual input.

Note that during the Repeat phase, no local odometry or heading estimation is used for navigation. All high-level decisions are based purely on visual familiarity. The onboard Kalman filter is used exclusively for low-level stabilization and good execution of yaw rate and landing maneuvers.

IV. Simulation

We conducted offline simulations to compare RoLa and ViCo algorithms under controlled variations of world complexity, FoV, scan amplitude, and lateral error (Table I). The synthetic images were captured in Gazebo ROS simulator. In the simulation context, we refer to the phases as *train* and *test*. The training route was L-shaped, 10 m long, and discretized into 20 positions (0.5 m spacing). The scan step was fixed to 5°.

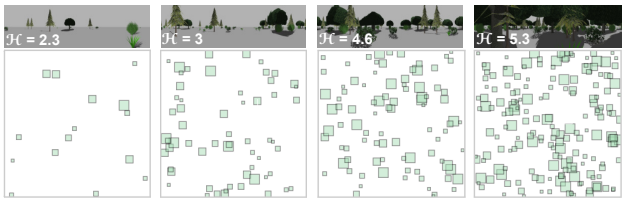

Fig. 4. Different world visual complexity measured by entropy (H), tested in Simulation.

TABLE I

SIMULATION PARAMETERS AND TESTED VALUES.

Parameter	Values
World complexity (H)	[2.3, 3.0, 4.6, 5.3]
FoV (deg)	[360, 315, 270, 225, 180, 135, 90, 45, 15]
Scan amplitude (deg)	[360 ($\pm$180), $\pm$135, $\pm$90, $\pm$45, $\pm$15]
Lateral error (m)	[-0.8, -0.4, -0.2, 0, 0.2, 0.4, 0.8]

A. Simulation Setup

1) Train Phase: During training, a key difference between the two methods is when the scan amplitude is used: RoLa applies it during training only, while ViCo applies it during testing. In ViCo algorithm, the number of stored views per memory is fixed at 20 (one per route position), independent of scan amplitude. With 4 worlds (Fig. 4) and 9 FoVs, this produces 36 distinct training setups. In RoLa algorithm, the number of stored views per memory depends on scan amplitude: for example, $\pm 45°$ with $5°$ steps yields 9 views per side (180 per memory, 360 total for left/right memories). A full $360°$ scan stores 3600 views per memory (7200 total). With 4 worlds, 9 FoVs, and 5 scan amplitudes, RoLa produces 180 distinct training setups. Though RoLa algorithm's training set size increases with scan amplitude, the size of the MB network and the storage footprint of the learned synaptic weights remain constant. It is noteworthy that each image passes through the model only once, as a one-shot learning.

2) Evaluation Metrics: For both methods, we treat the MB-model output as a binary classifier of egocentric route direction sign and compute $\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$, $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$.

Here TP/FP/TN/FN are defined for left/right turn decisions. For RoLa algorithm, positive (P)= turn-left and negative (N) = turn-right (TP: right view $\rightarrow$ turn-left; FP: left view $\rightarrow$ turn-left; TN/FN analogously). Similarly for ViCo algorithm, but turn direction is derived from the sign of the desired bearing. We also record mean computation time per decision and a qualitative deployability for a narrow-FoV MAV.

3) Test Phase: In RoLa algorithm, each memory setup is tested with $7 \times 20 \times 72 = 10{,}080$ views (7 lateral errors $\times$ 20 positions $\times$ 72 orientations from $5°$ steps over $360°$). In ViCo algorithm, the number of tested views grows with scan amplitude: for example, 725,760 views for a $360°$ scan vs. 60,480 for $\pm 15°$. Both methods have 180 distinct testing setups.

B. Simulation Results

Figure 5 summarizes the comparison between RoLa and ViCo algorithms across FoV, scan amplitude, lateral error (e), and scene complexity. In Figure 5a (d = 0), scan amplitude is the dominant factor for accuracy, with FoV only becoming limiting below $90°$ for both methods. RoLa algorithm maintains consistent performance through scene complexities, reaching a plateau at $\pm 90°$, while ViCo algorithm benefits more from maximum scan amplitudes. For a fixed FoV of $90°$

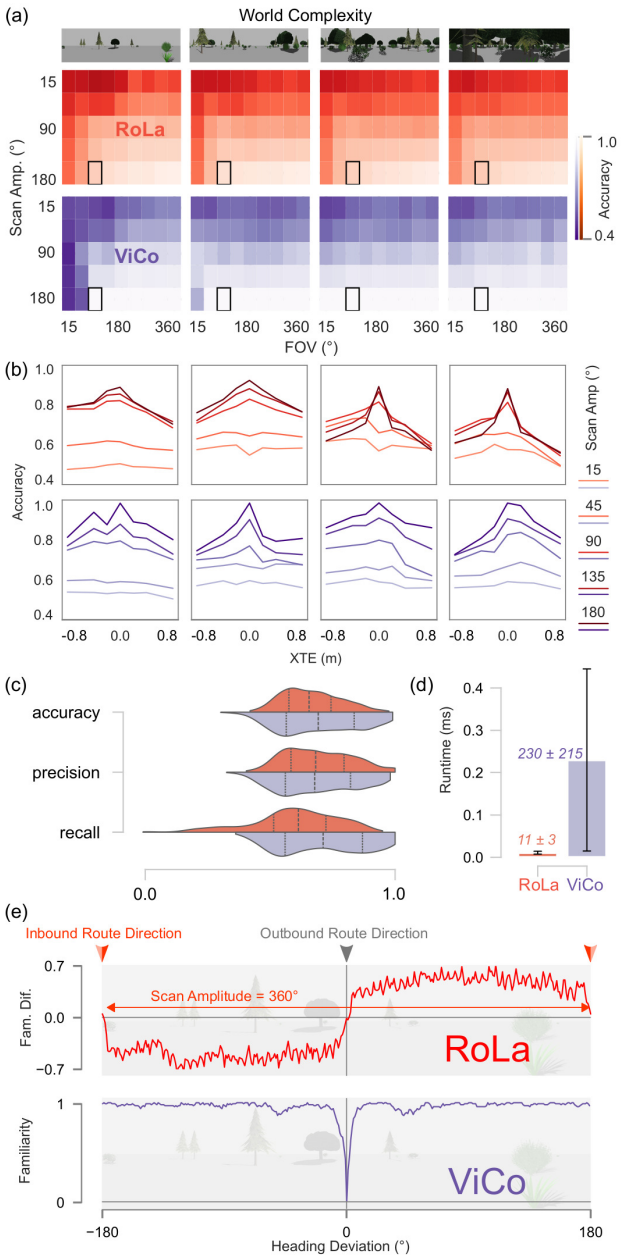

Fig. 5. Offline simulation analysis of RoLa vs. ViCo algorithms across FoV, scan amplitude, lateral error, and scene complexity. (a) Accuracy heatmaps; black box = AntFlie setup. (b) Accuracy vs. lateral error. (c) Accuracy/precision/recall. (d) Mean runtime. (e) Familiarity response

(Fig. 5b), ViCo shows slightly higher tolerance to large cross-track error in high-complexity worlds, but the gap closes for smaller scan amplitudes.

Overall accuracy (Fig. 5c) varies less for RoLa algorithm, while ViCo algorithm attains a slightly higher mean. Precision is similar between methods, but RoLa's lower recall suggests occasional missed turning decisions, which low occurrence would be corrected in closed-loop navigation. Runtime analysis (Fig. 5d) reveals a clear advantage for RoLa (11 ± 3 ms) over ViCo (230 ± 215 ms), as RoLa's evaluation cost is constant, whereas ViCo's scales directly with scan amplitude. Familiarity responses (Fig. 5e) show

RoLa producing a strong signed signal with clear zero crossings at 0° (outbound) and 180° (inbound), enabling direction discrimination, unlike ViCo's single-view storage.

We compared scan requirements for panoramic and narrow-FoV versions of RoLa and ViCo algorithms. In the training phase, panoramic RoLa and ViCo algorithms obtain multiple orientations via software rotation of stored panoramic images, while narrow-FoV versions must perform a physical yaw sweep through the scan amplitude (Fig. 5e). In RoLa algorithm, the number of scan steps equals the scan amplitude divided by the angular resolution, with local compass needed only for the narrow-FoV case. In ViCo algorithm, both panoramic and narrow-FoV store only the outbound-facing view at each location; the panoramic version generates heading in silico, whereas the narrow-FoV version must physically sweep toward the outbound-facing view, requiring local compass.

In the testing phase, RoLa (both panoramic and narrow) requires no scans and no local compass, evaluating only two familiarity scores per location. ViCo, however, must scan at test time: panoramic ViCo does this in silico, while narrow-FoV ViCo performs a continuous physical sweep through the scan amplitude, again requiring local compass.

In summary, RoLa and ViCo algorithms achieve comparable accuracy for FoVs above 90°, with scan amplitude being the main determinant of performance. Scene complexity has limited influence, but RoLa offers major benefits in computation time, direction discrimination, and narrow-FoV feasibility.

V. Real-World Experiments

A. Experimental Setup

Experiments were conducted in the Mediterranean Flight Arena (MFA), equipped with artificial visual cues and an 18-camera VICON motion capture system for ground-truth tracking. The AntFlie quadcopter (Fig. 6) is a 33 g Crazyflie 2.1 micro-drone carrying an AI-deck with an ultra-low-power Himax camera (0.2g, 87° FoV, 324 × 324 px), a PMW3901 optic flow sensor, and a VL53L1x ToF for altitude control, powered by a 1S–250 mAh LiPo battery.

State estimation via an onboard Kalman filter ran on the AI-deck, while the neural network was executed on an external PC (Python) for rapid prototyping, using the Intel i7 (8th gen) CPU. Images were transmitted over Wi-Fi via ROS, and the PC computed the desired commands during Repeat phase, relaying them to the MAV via radio dongle. We report the metrics using medians and Median Absolute Deviation (MAD).

B. Teach-Phase Validation

We first confirmed that AntFlie could align its outbound scans with the instantaneous route direction using only onboard IMU and ventral optic flow without access to the global smoothed path. In both ±45° and ±180° scans (Fig. 7), measured local headings (orange) closely tracked deviations from the smoothed outbound path (gray), confirming that the local compass provides accurate egocentric orientation within the route frame. This capability is a prerequisite for fully local, odometry-light Route Learning with narrow FoV vision.

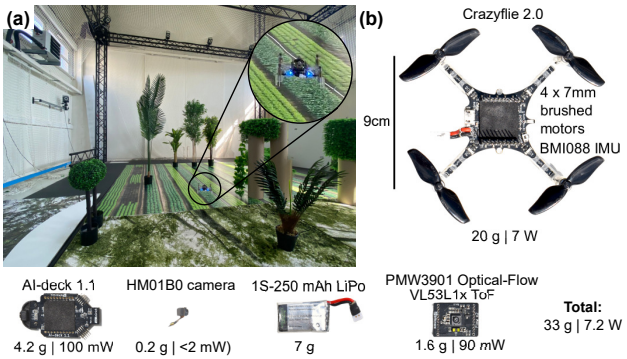

Fig. 6. Experimental setup comprising (a) the Mediterranean Flight Arena (MFA) with motion capture and visual cues, and (b) the 33 g AntFlie drone featuring low-power sensors for autonomous flights.

C. Experimental Results

1) Baseline Repeatability on a Straight Route: We validated the complete VT&R pipeline on a 3 m straight route using periodic ±45° scans (Fig. 7a, Fig. 8). After a single outbound learning flight, AntFlie completed n = 3 autonomous repeats with a median lateral error of 4 cm and a median heading error of 5.0° (Table II, Exp. S). This baseline confirmed that route memories and familiarity steering achieve highly repeatable path reproduction under minimal geometric complexity.

2) Inbound L-Shaped Routes and Autonomous Landing: We next evaluated AntFlie on four distinct L-shaped routes (Fig.9), each learned in a single outbound flight with periodic 360° scans. In total, $n = 24$ autonomous inbound flights were performed from displaced start positions (kidnapped-robot condition). Start conditions spanned absolute lateral errors from 5 mm to 73 cm and absolute heading errors from 0.5° to 67°. Due to the battery constraints of the 33 g platform, each outbound smoothed route was restricted to ∼4.7 m in length because of the scanning maneuvers consuming energy, even though the drone physically traveled ∼10m during learning. During the Repeat phase, the

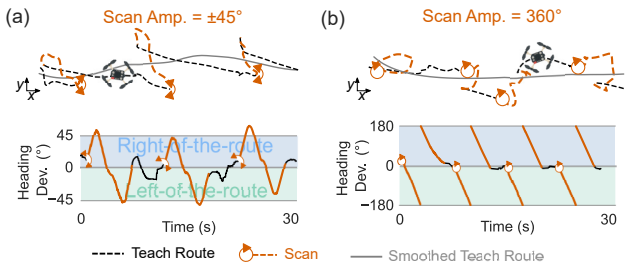

Fig. 7. Real-world validation of local heading estimation during teach-phase scans. With ±45° (a) and 360° (b) scans, AntFlie centers rotations on the instantaneous route direction from IMU and optic flow, and the resulting deviations from the smoothed path (gray) enable accurate view categorization on the right or on the left of the route without global route knowledge.

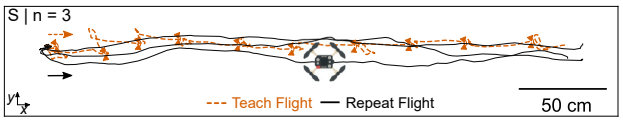

Fig. 8. Teaching a straight route with ±45° scans and autonomous inbound repeats.

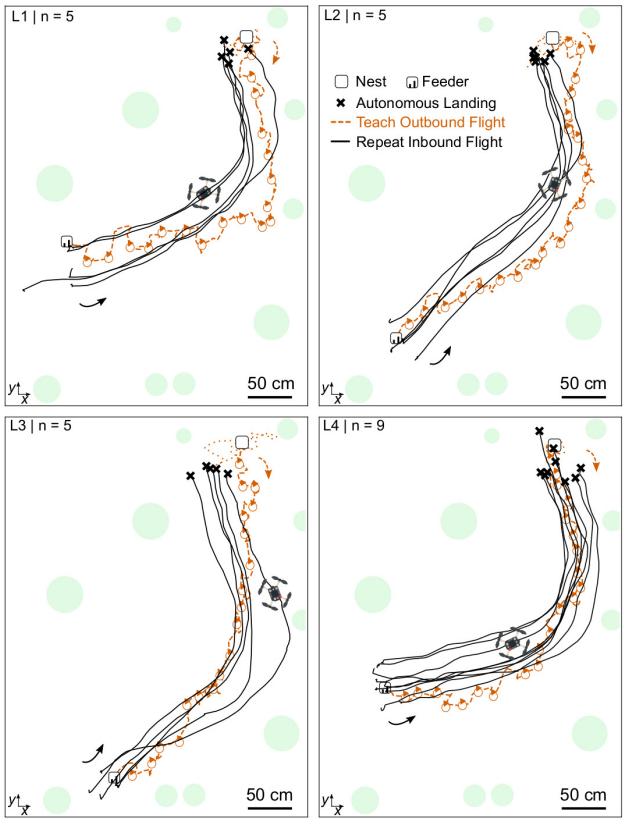

Fig. 9. Teaching four different L-shaped outbound routes with 360° scans and autonomous inbound repeats and landing. The MAVs consistently returned from the "feeder" to the "nest" using only visual route memories. Green circles indicate arena landmarks.

closed-loop control pipeline, including image capture, Wi-Fi transfer, neural network computation, and radio transmission of commands, operated at 8 Hz.

Path fidelity: Despite variations in outbound curvature and bending across experiments (Fig. 9), inbound trajectories consistently reproduced each specific learned path, rather than converging on a fixed beeline, indicating that control was driven by local visual familiarity relative to the local route learning direction.

Error stability: Across all repeats, the absolute heading error was 17.0° ± 12.3° and lateral error was 0.13 m ± 0.08 m (Table II, Fig. 10a). The Lyapunov function $V(t)$ (Eq. III-A) decreased monotonically in all cases, reaching $V(t)/V(0) < 1\%$ within a median of 16.6 s (IQR: 15.3–19.0 s) (Fig. 10b), confirming rapid convergence of cross-track (lateral), heading, and along-track errors.

Landing repeatability: Visual home familiarity triggered autonomous descent when $\lambda_H < 0.1$. Across all 24 landings, the median error was 0.34 m (MAD: 0.09 m), with a best case

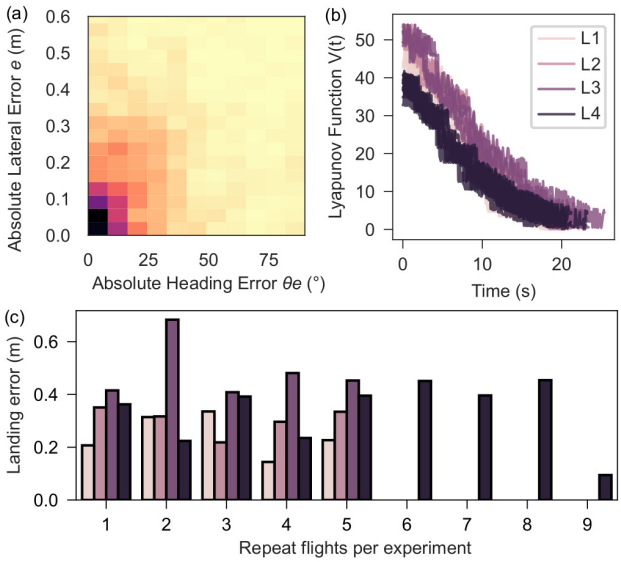

Fig. 10. Overall performance. (a) Lateral/heading error heatmap. (b) Lyapunov $V(t)$ over normalized time. (c) Landing errors per trajectory and experiment.

of 0.09 m (equal to body length) and worst case of 0.68 m (7 times the body length). The coefficient of variation (C_V) was 37%, indicating strong repeatability under purely vision-based narrow-FoV and ultra-low resolution conditions.

Memory footprint. Each Mushroom Body memory (i.e., the synaptic weight vector KCtoMBON) is stored as a bit-packed 10,000-bit vector (1 bit/synapse), requiring 1.22 kB per memory. Applying `gzip` to the packed bytes further reduces this to 1.1–1.2 kB. Thus, the three memories together occupy only 3.3–3.6 kB on disk, while supporting constant-time decoding. This fixed-size representation is more efficient than Compressed Sparse Row (CSR) formats (tested in [11]), whose size depends on sparsity. For comparison, storing the same synaptic weights as CSV files requires ≈120 kB, about 34 times larger, and storing the raw image would be even more memory-intensive.

VI. Conclusion

We demonstrated that route-lateralized visual memories learned during outbound physical scans, combined with a local compass, enable a 33-gram MAV with narrow FoV and ultra-low-resolution vision to perform reliable autonomous inbound homing and landing—without maps, global localization, or odometry during repeat, and using only local

TABLE II
Median (± MAD) of error metric per experiment.

Exp.	Heading error[°]	Lateral error[m]	Landing error[m]
S	5±3	0.04±0.02	–
L1	23±13	0.21±0.07	0.23±0.08
L2	27±20	0.22±0.09	0.32±0.02
L3	31±20	0.14±0.09	0.45±0.04
L4	14±10	0.12±0.07	0.39±0.06
ALL	17±12	0.13±0.08	0.34±0.09

segment odometry during teach. In simulation, our framework matched a state-of-the-art insect visual compass under diverse conditions. In real flights, AntFlie reproduced taught routes over a cumulative distance of 110 m with low lateral error (0.13 ± 0.08 m), high repeatability using less than 4 kB of memory, and executed 24 autonomous landings with high precision (0.34 ± 0.09 m). The route lateralized (RoLa) model is biologically plausible even using narrow FoV : the physical scans account for the head/body movements of ants and bees, whose visual fields also contain blind spots [13], and the local odometry resides in the insect central complex [33]. The RoLa model is practical across platforms and cameras, requiring no deblurring, calibration, stereo or panoramic vision. Current limitations include susceptibility to perceptual aliasing in visually repetitive environments and the inability to store multiple routes without memory overwriting. Future work will address forgetting via complementary memory architectures, integrate coarse place-recognition to improve aliasing resilience, and implement the system fully onboard microcontrollers. A few kilobytes of ant-inspired memory suffice for reliable, empirically Lyapunov-stable homing and landing on a micro-drone—no maps, no SLAM, only vision.

REFERENCES

[1] T. A. Ward, C. J. Fearday, E. Salami, and N. Binti Soin, "A bibliometric review of progress in micro air vehicle research," *International Journal of Micro Air Vehicles*, vol. 9, no. 2, pp. 146–165, Jun. 2017.

[2] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, B. J. Nelson, B. Scassellati, M. Taddeo, R. Taylor, M. Veloso, Z. L. Wang, and R. Wood, "The grand challenges of *Science Robotics*," *Science Robotics*, vol. 3, no. 14, p. eaar7650, Jan. 2018.

[3] A. Patel, S. Karlsson, B. Lindqvist, C. Kanellakis, A.-A. Agha-Mohammadi, and G. Nikolakopoulos, "Towards energy efficient autonomous exploration of Mars lava tube with a Martian coaxial quadrotor," *Advances in Space Research*, vol. 71, no. 9, pp. 3837–3854, May 2023.

[4] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, Sep. 2010.

[5] M. Warren, M. Greeff, B. Patel, J. Collier, A. P. Schoellig, and T. D. Barfoot, "There's No Place Like Home: Visual Teach and Repeat for Emergency Return of Multirotor UAVs During GPS Failure," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 161–168, Jan. 2019.

[6] B. Al-Tawil, T. Hempel, A. Abdelrahman, and A. Al-Hamadi, "A review of visual SLAM for robotics: Evolution, properties, and future applications," *Frontiers in Robotics and AI*, vol. 11, p. 1347985, Apr. 2024.

[7] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[8] B. Webb and A. Wystrach, "Neural mechanisms of insect navigation," *Current Opinion in Insect Science*, vol. 15, pp. 27–39, Jun. 2016.

[9] R. Wehner, M. Boyer, F. Loertscher, S. Sommer, and U. Menzi, "Ant Navigation: One-Way Routes Rather Than Maps," *Current Biology*, vol. 16, no. 1, pp. 75–79, Jan. 2006.

[10] G. C. H. E. De Croon, J. J. G. Dupeyroux, S. B. Fuller, and J. A. R. Marshall, "Insect-inspired AI for autonomous robots," *Science Robotics*, vol. 7, no. 67, p. eabl6334, Jun. 2022.

[11] G. G. Gattaux, A. Wystrach, J. R. Serres, and F. Ruffier, "Route-centric ant-inspired memories enable panoramic route-following in a car-like robot," *Nature Communications*, 2025, in press.

[12] C. A. Freas and K. Cheng, "Visual learning, route formation and the choreography of looking back in desert ants, Melophorus bagoti," *Animal Behaviour*, vol. 222, p. 123125, Apr. 2025.

[13] S. Schwarz, A. Narendra, and J. Zeil, "The properties of the visual system in the Australian desert ant Melophorus bagoti," *Arthropod Structure & Development*, vol. 40, no. 2, pp. 128–134, Mar. 2011.

[14] M. Mahdavian, K. Yin, and M. Chen, "Robust Visual Teach and Repeat for UGVs Using 3D Semantic Maps," Jun. 2022.

[15] P. Nourizadeh, M. Milford, and T. Fischer, "Teach and Repeat Navigation: A Robust Control Approach," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. Yokohama, Japan: IEEE, May 2024, pp. 2909–2916.

[16] A. Pfrunder, A. P. Schoellig, and T. D. Barfoot, "A Proof-of-Concept Demonstration of Visual Teach and Repeat on a Quadrocopter Using an Altitude Sensor and a Monocular Camera," in *2014 Canadian Conference on Computer and Robot Vision*. Montreal, QC, Canada: IEEE, May 2014, pp. 238–245.

[17] E. Garcia-Fidalgo and A. Ortiz, "Vision-based topological mapping and localization methods: A survey," *Robotics and Autonomous Systems*, vol. 64, pp. 1–20, Feb. 2015.

[18] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1. Minneapolis, MN, USA: IEEE, 1996, pp. 83–88.

[19] L. Sun, M. Taher, C. Wild, C. Zhao, Y. Zhang, F. Majer, Z. Yan, T. Krajnik, T. Prescott, and T. Duckett, "Robust and Long-term Monocular Teach and Repeat Navigation using a Single-experience Map," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, Sep. 2021, pp. 2635–2642.

[20] L. G. Camara, T. Pivonka, M. Jilek, C. Gabert, K. Kosnar, and L. Preucil, "Accurate and Robust Teach and Repeat Navigation by Visual Place Recognition: A CNN Approach," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 6018–6024.

[21] T. Krajnik, F. Majer, L. Halodova, and T. Vintr, "Navigation without localisation: Reliable teach and repeat based on the convergence theorem," Jul. 2018.

[22] G. Caron, E. Marchand, and E. M. Mouaddib, "Photometric visual servoing for omnidirectional cameras," *Autonomous Robots*, vol. 35, no. 2-3, pp. 177–193, Oct. 2013.

[23] J. Zeil, M. I. Hofmann, and J. S. Chahl, "Catchment areas of panoramic snapshots in outdoor scenes," *Journal of the Optical Society of America A*, vol. 20, no. 3, p. 450, Mar. 2003.

[24] J. C. Knight, D. Sakhapov, N. Domcsek, A. D. M. Dewar, P. Graham, T. Nowotny, and A. Philippides, "Insect-Inspired Visual Navigation On-Board an Autonomous Robot: Real-World Routes Encoded in a Single Layer Network," 2019.

[25] T. Van Dijk, C. De Wagter, and G. C. H. E. De Croon, "Visual route following for tiny autonomous robots," *Science Robotics*, vol. 9, no. 92, p. eadk0310, Jul. 2024.

[26] M. Mangan and B. Webb, "Spontaneous formation of multiple routes in individual desert ants (Cataglyphis velox)," *Behavioral Ecology*, vol. 23, no. 5, pp. 944–954, 2012.

[27] P. Ardin, F. Peng, M. Mangan, K. Lagogiannis, and B. Webb, "Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments," *PLOS Computational Biology*, vol. 12, no. 2, p. e1004683, Feb. 2016.

[28] G. Gattaux, A. Wystrach, F. Ruffier, and J. Serres, "Enhancing ant-inspired visual compass with focused visual scan in a compact robot," in *The 7th IEEE International Conference on Artificial Intelligence Circuits and Systems*. IEEE, 2025, pp. 110–114.

[29] J. Stankiewicz and B. Webb, "Looking down: A model for visual route following in flying insects," *Bioinspiration & Biomimetics*, vol. 16, no. 5, p. 055007, Sep. 2021.

[30] W. Kuang, H. W. Ho, Y. Zhou, and S. A. Suandi, "ForaNav: Insect-Inspired Online Target-Oriented Navigation for MAVs in Tree Plantations," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 7063–7069, Jul. 2025.

[31] A. Wystrach, F. Le Moël, L. Clement, and S. Schwarz, "A lateralised design for the interaction of visual memories and heading representations in navigating ants," Animal Behavior and Cognition, Preprint, Aug. 2020.

[32] A. Wystrach, "Neurons from pre-motor areas to the mushroom bodies can orchestrate latent visual learning in navigating insects," *Biorxiv*, pp. 2023–03, 2023.

[33] J. D. Seelig and V. Jayaraman, "Neural dynamics for landmark orientation and angular path integration," *Nature*, vol. 521, no. 7551, pp. 186–191, 2015.