sensors

MDPI

Article

Chasing Ghosts: A Simulation-to-Real Olfactory Navigation Stack with Optional Vision Augmentation

Kordel K. France *, Ovidiu Daescu, Latifur Khan and Rohith Peddi

Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, USA; ovidiu.daescu@utdallas.edu (O.D.); latifur.khan@utdallas.edu (L.K.); rohith.peddi@utdallas.edu (R.P.)
* Correspondence: kordel.france@utdallas.edu

Abstract

Autonomous odor source localization remains a challenging problem for aerial robots due to turbulent airflow, sparse and delayed sensory signals, and strict payload and computation constraints. While prior unmanned aerial vehicle (UAV)-based olfaction systems have demonstrated gas distribution mapping or reactive plume tracing, they rely on predefined coverage patterns, external infrastructure, or extensive sensing and coordination. In this work, we present a complete, open-source UAV system for online odor source localization using a minimal sensor suite. The system integrates custom olfaction hardware, onboard sensing, and a learning-based navigation policy that we train in simulation and deploy on a real quadrotor. Through our minimal framework, the UAV is able to navigate directly toward an odor source without constructing an explicit gas distribution map or relying on external positioning systems. We incorporate vision as an optional complementary modality to accelerate navigation under certain conditions. We validate the proposed system through real-world flight experiments in a large indoor environment using an ethanol source, demonstrating consistent source-finding behavior under realistic airflow conditions. The primary contribution of this work is a reproducible system and methodological framework for UAV-based olfactory navigation and source finding under minimal sensing assumptions. We elaborate on our hardware design and open-source our UAV firmware, simulation code, olfaction–vision dataset, and circuit board to the community.

Keywords: olfactory navigation; robotics; UAV; odor source localization; reinforcement learning; computer vision; sim-to-real transfer; stereo olfaction; electrochemical sensors; metal oxide sensors

Check for updates

Academic Editor: Bingbing Gao

Received: 10 March 2026
Revised: 18 April 2026
Accepted: 27 April 2026
Published: 2 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

How does one find what one cannot see, hear or touch? Researchers have likened navigating by scent to "chasing ghosts" due to humans' lack of high-bandwidth olfactory perception. Olfaction is the most primitive form of perception, yet artificial intelligence systems have predominantly focused on visual, audio, and linguistic data. This oversight largely stems from the scarcity of olfactory data and the absence of standardized benchmarks, which pose significant challenges for developing and evaluating machine learning models in this domain.

In this study, we introduce a model for olfaction-only navigation, but demonstrate how vision can complement performance. We show how these models can be trained via simulation to run at the edge for scent-based navigation to an odor source. Olfactory

navigation is largely absent from robotic functions today, and our motivation for this work was to establish methodology that allows robots to, for example, localize the source of a specified chemical compound, inform automated surgery by mapping olfactory cues to visual components, and inspire alternative forms of underwater navigation via chemical plume tracking. We demonstrate our methodology on a real UAV navigating to a chemical compound.

Due to the hysteresis and non-linearity associated with gas dynamics, navigating by scent is inherently difficult. Current olfaction sensors suffer from drift and long sampling times [1–3], and the inability to accurately perceive the presence of a chemical compound analogizes to "chasing ghosts". Here, we detail a confluence of adaptive learning techniques, hardware design, sensor selection, and computation optimizations that make this feasible. We provide a detailed account of our hardware, methodologies, and evaluation metrics to ensure transparency and reproducibility. The Appendices A–H detail a full bill of materials and code to reproduce our results. Figure 1 shows frontal views of the two configurations we used for the UAV.

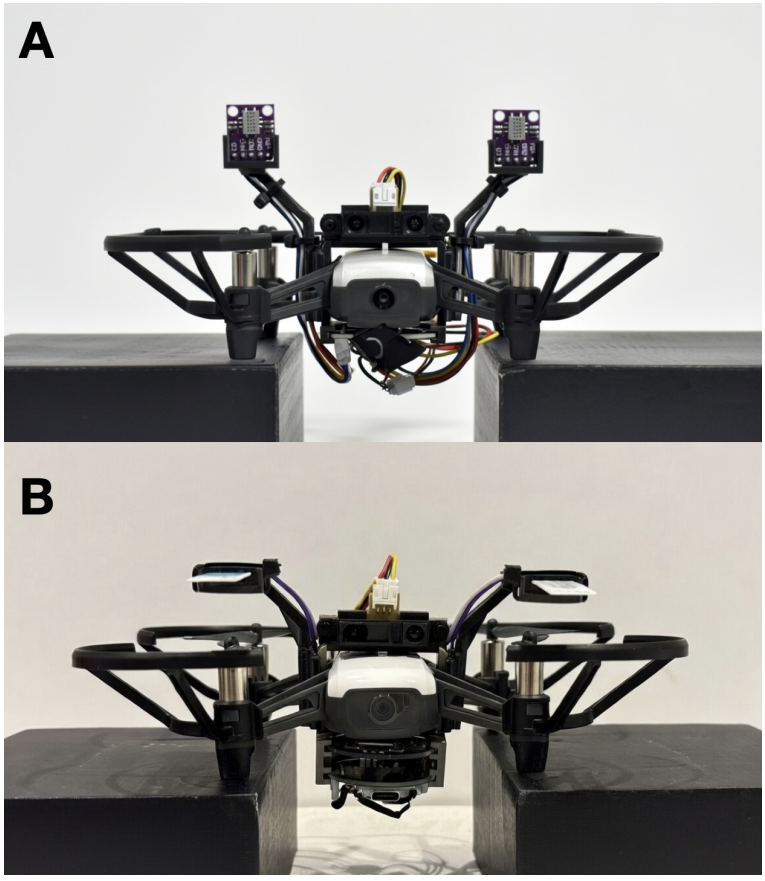


Figure 1. The UAV equipped with the olfactory processing unit (OPU) and sensor harnesses for scent-based navigation. (**A**) shows the metal oxide sensor configuration, and (**B**) shows the electrochemical sensor configuration.

The contributions of our work are as follows:

1. A complete, open-source UAV system for odor source localization, integrating custom olfaction hardware, onboard sensing, simulation, and real-world deployment under payload and computation constraints.
2. A learning-based navigation method that we train in simulation and deploy on a real UAV, enabling online odor source localization without prior mapping or external positioning infrastructure.

3. A minimal-sensor navigation formulation, demonstrating reliable source-seeking behavior using only a small number of olfaction sensors and a single range sensor, with vision serving optionally as a complementary modality.
4. Real-world experimental validation in a large indoor environment, showing consistent source localization behavior and providing qualitative and quantitative insights into plume navigation under realistic airflow conditions.
5. A demonstration that our framework generalizes across sensing mediums.

This work emphasizes system design and methodological integration, on which one could expand with statistically powered evaluations across a wider range of environments and conditions.

The code, data, and designs are available at https://github.com/KordelFranceTech/ChasingGhosts (accessed on 26 April 2026).

2. Related Work

Early UAV-based olfaction systems have focused on gas distribution mapping (GDM), where a robot follows a predefined flight pattern to reconstruct a concentration field and estimate a source location post hoc. Burgués et al. [4] presented a representative example, employing a quadrotor to sample gas concentrations along a sweeping trajectory and infer the source location from the reconstructed map. While effective for environmental monitoring, such approaches do not perform online source-seeking behavior and require substantial spatial coverage prior to localization.

Reactive plume-tracing methods draw inspiration from biological odor tracking, enabling robots to move toward an odor source based on local concentration cues. Shigaki et al. [5] demonstrated three-dimensional chemical plume tracing using a quadcopter equipped with dual alcohol sensors and onboard ranging, achieving successful localization in controlled indoor environments. Their work highlights the feasibility of UAV-based plume tracking, but focuses on reactive control strategies rather than learning-based policies trained across environments.

More recently, researchers have explored learning-based and swarm-based approaches to improve robustness in turbulent environments. Duisterhof et al. [6] introduced a multi-agent "Sniffy Bug" system that combines bio-inspired navigation with particle swarm optimization, leveraging inter-agent communication and additional sensing modalities. While effective at the swarm level, such systems rely on greater sensing and coordination infrastructure than a single lightweight UAV typically carries. Research from Zhang et al. [7,8] demonstrates how chemical and visual sensing can work collaboratively on a UAV, but does not demonstrate it in navigation tasks.

In contrast to prior work, this paper focuses on single-agent, learning-based odor source localization under minimal sensing assumptions. Rather than reconstructing gas distributions or relying on swarm coordination, the proposed system learns a navigation policy in a simulation and deploys it directly on a real UAV, enabling online source-seeking behavior in a large indoor environment. The emphasis is on system integration, reproducibility, and sim-to-real transfer, providing a foundation for future statistically powered evaluations and extensions. The works in [4,6] leverage ultrawide-bandwidth (UWB) radio tags placed throughout each course to help the UAV navigate. In contrast, we only allow the UAV to use olfaction (and in the last mile, vision) to guide itself to the odor source, and it has no prior knowledge about the course. Unlike the post-mapping over 160 m^2 from [4], we achieved real-time localization over 200 m^2, the largest-sized experimental course so far, without positional aids.

The work by Hassan et al. [9] shows how robots can navigate by olfactory and visual capabilities, but leverages closed-source online large language models to perform

the computations as well as the interpretation of the chemical and vision data, methods impractical for real-time edge robotics. In [4–6,9], the authors use metal oxide sensors as the olfactory receptor. We demonstrate our framework with metal oxide sensors to show the improvement on previous work, but also demonstrate that our framework generalizes across sensing mediums with the introduction of electrochemical sensors.

Feng et al. constructed a dataset of vision–olfactory relationships in SmellNet [10], but the dataset related exclusively to fruits, nuts and other common foods. Ozguroglu et al. [11] showed how vision and olfaction can link together through contrastive learning, but the scope was exclusive to botany. We built off many of the methods established by [4,6,12]. Namely, we used ethanol as our target compound to maintain parity with their work and provide a level of experimental control.

To our knowledge, this is the first demonstration of stereo-olfactory sensing combined with vision on a UAV over two different chemical sensing mediums. Appendix A presents a concise comparison of our work to related work.

3. Methods

Our primary goal was to demonstrate that UAVs can navigate purely with olfactory sensors to track a target chemical back to its source. However, we recognize that visual sensors can complement olfactory navigation, as seen in several mammals and insects. As a result, we show how understanding the aromas emitted by certain objects can decrease the time needed to localize the target compound through visual confirmation of capturing the odor source.

3.1. Hardware

We desired a solution for olfactory navigation that is financially and technically accessible to robotics and AI researchers while also reaching the precision required to localize a chemical compound. Therefore, we selected off-the-shelf hardware where reasonably possible. Figure 2 shows critical additions to the base UAV airframe, while Appendix H discusses a full bill of materials.

We selected the DJI Tello as the UAV [13], a small open-sourced drone that one can program, but that contains minimal onboard computations. Unlike prior works [4,6] that leverage the Crazyflie UAV platform, we required a slightly larger UAV platform with a camera and higher payload that enables us to carry more sensing capability and computations. Three infrared sensors provide basic stability: two downward-looking sensors for leveling and one forward-looking sensor for obstacle detection. We designed a custom olfactory processing unit (OPU) that contains an ESP-32 microcontroller and a PalmSens EmStat Pico potentiostat and attaches to the bottom of the UAV. The OPU processes the infrared data for obstacle detection and olfaction sensors to inform navigation, and wirelessly streams all data back to a laptop ground station for telemetry. We mechanically modified the Tello UAV to accommodate the harnessing, battery, and circuitry that the OPU requires. This required adding 71 g of ballast to the aft end of the drone in order to counter the forward moment that the additional components create; no additional ballast was needed to adjust for lateral moments. We designed the mechanical brackets and harnessing needed to attach the OPU, olfactory sensors, and infrared sensors to the airframe in SolidWorks [14]. We designed the OPU motherboards using EasyEda [15]. Figure 3 gives a snapshot of the hardware additions while Appendices G and H present more details on the electrical and mechanical hardware, respectively.

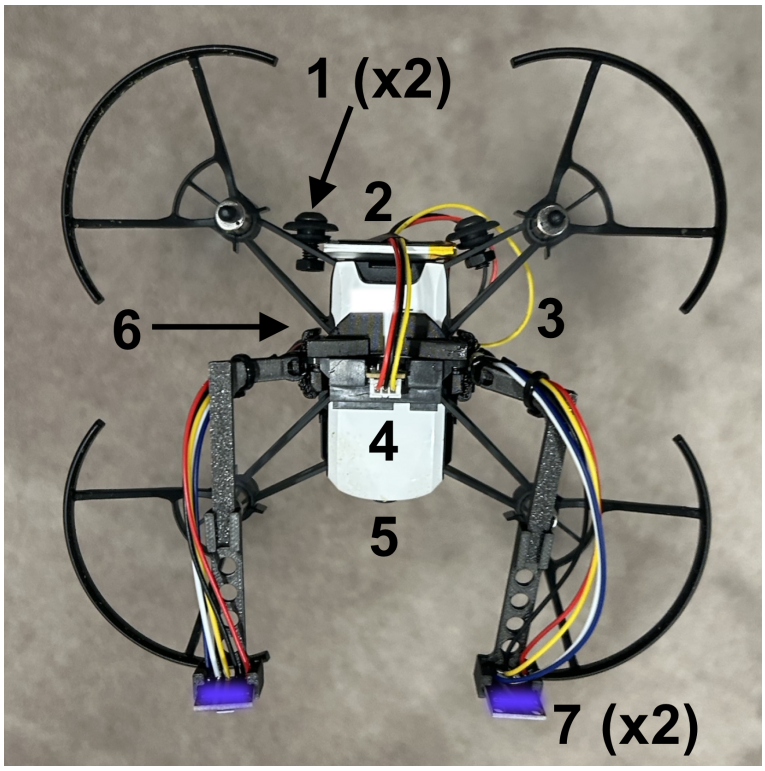


Figure 2. The UAV equipped with the olfactory processing unit (OPU) and sensor harnesses. The figure shows a top-down view of the aircraft: (1) the 2 ballast points needed to balance the aircraft; (2) the battery to power the motherboard required for the olfactory sensors; (3) the wire harnessing leading to the motherboard attached to the belly of the aircraft; (4) the forward-looking time-of-flight sensor for obstacle avoidance; (5) the forward-looking camera; (6) the motherboard and downward-looking time-of-flight infrared sensors attach to the belly, and (7) the two olfactory sensor antennae (MOX sensors shown, but EC sensors lie at the same location when in the proper configuration).

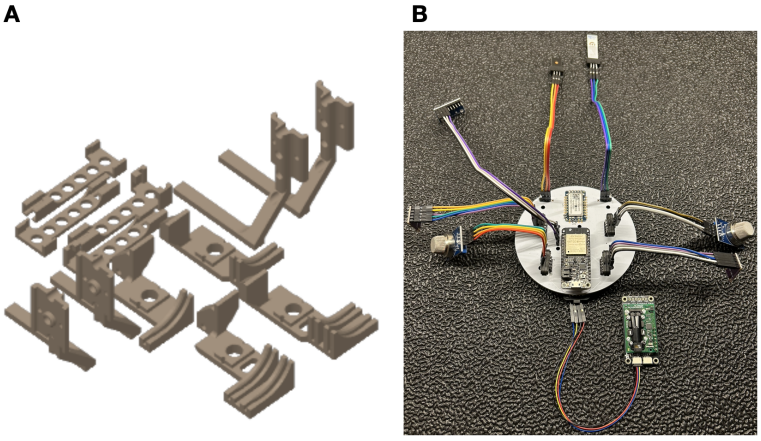


Figure 3. (**A**) One can additively manufacture the full UAV modification kit for both electrochemical and metal oxide sensing configurations on a single 200 mm × 200 mm print bed. (**B**) An image of the breakout board we designed to down-select the olfaction sensors for ethanol detection.

3.2. Simulation

We created a digital twin of the modified UAV for two different simulations. We built a physics simulation via SimuLink to model the control loop algorithms for the UAV, test various flight conditions, and establish a control envelope. Figure 4 shows a series of plots from this simulation, establishing the performance of the autopilot control algorithms (for more information on these algorithms, see Appendix B and Tables A2 and A3). This simulation contains a configuration file that enables extrapolation of our control laws to

any other UAV platform with our OPU modification. We built the second simulation using Python 3.12 and the Gymnasium framework [16], which we discuss in detail in the next sections. We leveraged and tuned Kalman filters to smooth signals in the control responses. Taking inspiration from olfactory navigation in insects, we positioned two sensors at the top and fore of the UAV in analogous positions, as the antennae on the silk moth. This differs from previous approaches of olfactory navigation [4,6], where the sensor sat at the top of the body and below the rotors, but it is similar to the locations proposed by [5]. Our positioning allowed us to perform stereo sensing to ascertain the plume direction and avoid volatility from the rotor wash, maximizing the odor signal.

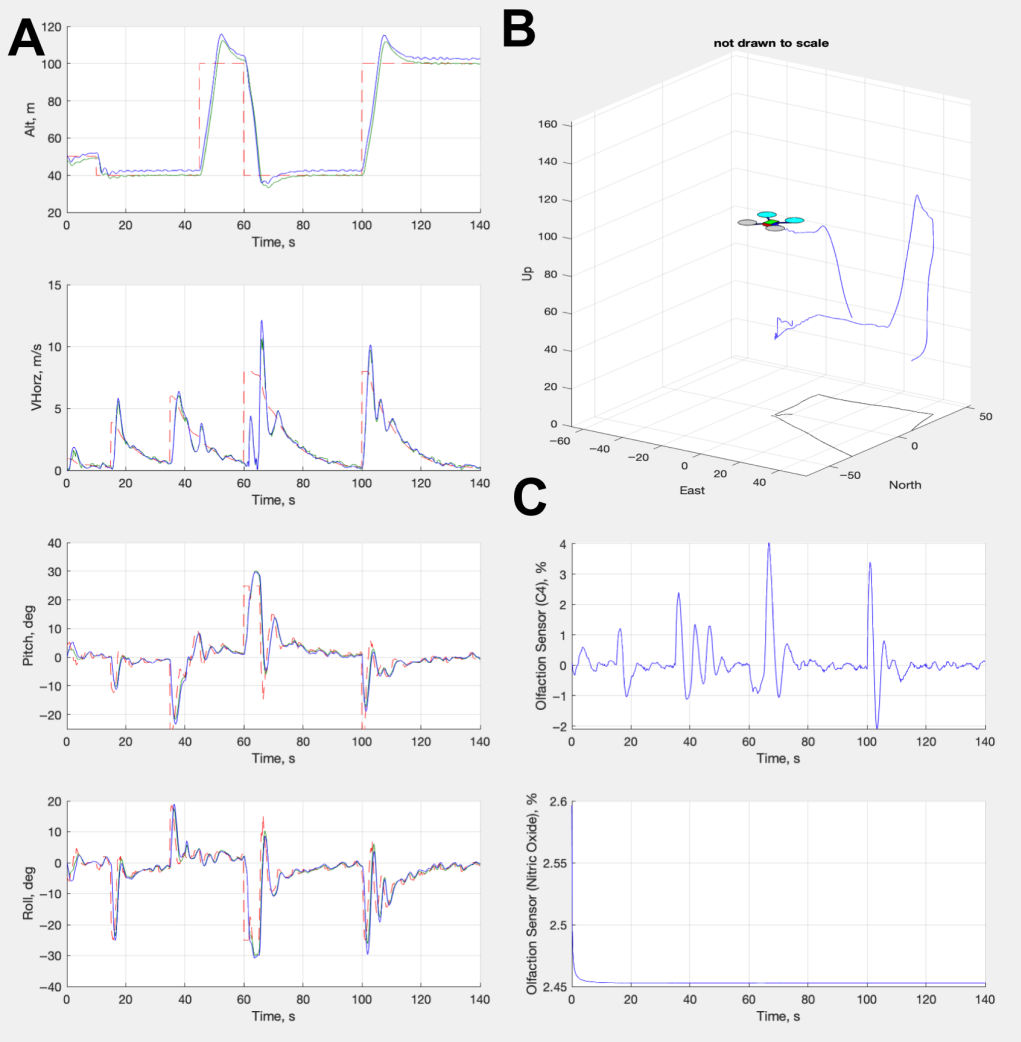


Figure 4. (**A**) From top to bottom, this panel shows the altitude, velocity, pitch, and roll from the controller response (blue line) and the command (red line) from the control algorithms. (**B**) A diagram showing a partial flight path of the UAV in the simulation. (**C**) From top to bottom, this panel shows the olfactory signal responses for ethanol and nitrogen dioxide. Red dotted lines indicate the desired value. Green solid lines indicate the values executed by the autopilot. Blue solid lines indicate the actual achieved values by the UAV.

3.3. Sensors

We desired a generalized solution for olfactory navigation. To aid with down-selection of off-the-shelf olfaction sensors, we designed a small electrical breakout board (see Appendix G). This allowed us to manually test detection algorithms without having to reconfigure the mounting hardware and ballast on the drone. The repository associated with this paper contains the circuitry and design of this board. The results of the experiments

we performed with this olfactory board showed that metal oxide (MOX) and electrochemical (EC) sensors were the best for tracking ethanol, so we selected them as the final olfactory effectors.

Metal oxide sensors respond rapidly (1–100 Hz) [17] and are typically sensitive to families of molecules, allowing for more general chemical tracking. We selected a pair of SGX Sensortech MICS 6814 sensors (Corcelles-Cormondreche, Switzerland) [18] due to their prevalence in the recent literature [10,17] and the wide array of compounds they detect. These sensors respond to a variety of gases, but we focused on their sensitivity to ethanol for our work here, since ethanol was the target compound.

Metal oxide sensors change their resistance based on their exposure to the target gas. This resistance indicates the gas intensity in the surrounding environment, and one can compute it according to the following equation:

$$R_s = \left[\frac{V_c}{V_{RL}} - 1\right] \cdot R_L \tag{1}$$

where R_S indicates the sensor resistance, V_c is the circuit voltage, V_{RL} is the voltage drop over the load resistor, and R_L is the resistance of said load resistor. As Equation (1) is fundamental to the function of the MOX sensor, we designed our optimizations around it. We built off of the methodology from Burgués et al. in [4], in that we designed our algorithm to not depend on absolute concentrations in the parts-per-million or parts-per-billion quantity. Rather, we analyzed the relative change between time steps of V_{RL} (the temporal difference) and used this to inform the UAV on how to move relative to gas measurements.

Electrochemical sensors respond less rapidly, but much more specifically, to individual compounds, allowing for more precise tracking. We leveraged a two-electrode electrochemical sensor from ItalSens designed to maximize diffusivity of compounds over a certain voltage range for chronoamperometry. Chronoamperometry is the process by which one can measure the change in electric current with respect to time over a controlled electrical potential. The Cottrell equation shows this relationship:

$$I = \frac{n_e F A c_k \sqrt{D_k}}{\sqrt{\pi t}} \tag{2}$$

where I denotes the electrical current, measured in amperes; n_e is the number of electrons needed to oxidize one molecule of analyte k; F is the Faraday constant of 96,485 Coulombs per mol; A denotes the planar area of the electrode in square centimeters; c_k defines the initial concentration of the target analyte k in mols per cubic centimeter; D_k defines the diffusion coefficient for analyte k in square centimeters per second; and t is simply the time that the chronoamperometric sequence is running in seconds. Our sensor contains an electrode surface area of 2.25 square centimeters and specifically tracks a target analyte with a reduction potential of 0.8 V according to Fick's law. We utilized 1-ethyl-3-methylimidazolium tetrafluoroborate ($[EMIM][BF_4]$) as the ionic liquid to act as a transducer over our electrodes.

Typically, a single chronoamperometric sequence operates for anywhere between 6 and 60 s. We borrowed the method from [3] to infer the analyte diffusion in order to speed up the sampling rate to 0.5–1 Hz. Appendix C presents more details on this technique. Unlike our approach with the MOX sensors, we did adjust the fundamental operation of the EC sensors in Equation (2).

3.4. Dataset

We performed pattern recognition over the olfactory data via adaptive temporal difference learning. We did not accumulate a dataset a priori for training the olfactory sensors, as we determined that adaptively learning the plume via real-time exploration approximation was sufficient, following many of the methods that Sutton set forth in [19]. The data for the olfaction–vision models comes from [20], which fuses trusted and peer-reviewed olfactory [21–23] and computer vision [24] datasets, and is also the only known olfaction–vision dataset.

While modern reasoning models can associate from which objects an *aroma* (e.g., lingual descriptors such as "fruity" and "musky") is coming in an image, others find that they have difficulty in specifying from where a *chemical compound* (e.g., CO_2, CH_4) is coming [10,20,25]. In addition, leveraging a large vision–language reasoning model—most of which are billions or trillions of parameters in size—is neither practical nor necessary for real-time edge robotics. To this end, we constructed our own encoder to model olfactory–vision relationships, discussed in Section 3.5.2.

3.5. Models

3.5.1. Olfaction-Only Navigation

To track a plume, we must make some inference on its direction. We used the time-delay estimation between the left and right sensors to inform the horizontal plume angle of attack against the sensor surface. Let x_L and x_R represent the left and right lateral positions of the sensors onboard the aircraft, respectively. For a planar odor front advecting with velocity u, wind speed s, and sensor separation $\Delta x = x_L - x_R$, we can define the expected sensor time constant $\hat{\tau}$ as:

$$\hat{\tau} \approx -\frac{\Delta x \cdot u}{s^2} \tag{3}$$

For a pure lateral baseline Δx and known wind speed s, the angle ϕ between the plume travel direction and the longitudinal axis of the aircraft resolves to:

$$\phi = \arcsin \frac{\hat{\tau} s}{d} \tag{4}$$

This effectively converts a measured time lag into a heading command for the aircraft. We clipped ϕ such that $\phi \in [-90°, -5°] \cup [5°, 90°]$ such that the UAV will hold the current heading for all $\phi \in (-5°, 5°)$. Large angles can be likened to casting (exploratory) behaviors in insects, while smaller angles (meaning that the UAV stays its current course) can be likened to surging (exploitive) behavior.

The UAV samples the readings every second from each of the two MOX sensors and every two seconds for the electrochemical sensors. We applied dual-timescale exponential averaging to detect plume bouts while suppressing sensor drift and turbulence noise. This parallels filtering mechanisms observed in insect olfactory neurons. We calculated this divergence D by subtracting a long exponential moving average (EMA) with period β from a short EMA with period α, denoted by E_β and E_α, respectively:

$$D = E_\alpha - E_\beta \tag{5}$$

D can be considered as a temporal-prediction-difference filter, which is mathematically adjacent to TD error computation in reinforcement learning, albeit not a learning mechanism. Although D does not constitute temporal-difference learning in the reinforcement-learning sense, it produces a temporal-prediction error—the difference between fast and slow odor expectations. This is conceptually similar to biological temporal difference filters that insect olfactory pathways use to emphasize odor onset and plume entry events.

We additionally computed a signal line S, which is a smoothed EMA of D with period ρ, where $\alpha < \rho < \beta$. This signal line serves as a dynamic expectation of odor-trend behavior. A positive deviation between lines D and S indicates an accelerating odor concentration, consistent with entry into the plume, whereas a negative deviation reflects loss of odor momentum and triggers casting behavior. This dual-timescale architecture parallels both biological olfactory adaptations and momentum-based filtering methods in quantitative finance.

Formally, let $C(t)$ define a function over time t for the olfaction sensor readings, where $C_L, C_R \in C$ define functions for the left and right sensors, respectively. Then,

$$D(t) = E_\alpha(C(t)) - E_\beta(C(t)) \tag{6}$$

And the signal line is

$$S(t) = E_\rho(D(t)) \tag{7}$$

While ϕ gives the UAV the incident angle of the plume against the sensors, the bookkeeping of the D and S sequences allows the UAV to determine its proximity relative to the plume. For the experiments shown here, we selected the empirically deduced values of 3, 8, and 5 for α, β, and ρ, respectively.

3.5.2. Olfaction–Vision Navigation

For navigation augmented by vision, we trained and evaluated a multimodal machine learning model for understanding olfaction–vision relationships. Figure 5 shows the main architecture. It employs a CLIP-based vision encoder [26] to extract visual features from images. A separate 12-layer fully connected encoder processes chemical data from the olfactory sensors, converting it into 138-dimension olfactory embeddings. We call this model *COLIP (contrastive olfaction–language–image pre-training)* since it builds on the original CLIP model. A learned linear layer projects both visual and olfactory embeddings into a shared 512-dimension latent space, ensuring compatibility before further processing. A graph associator learns joint relationships between the projected visual and olfactory embeddings. We evaluated two such models to assess tradeoffs in different architectures: a simple fully connected network and a graph attention network (GAT). Both models use the same 12-layer encoder. The GAT captured more complex associations between the projected visual and olfactory embeddings, but we found difficulties in a reliable performance when exporting to a format that could run at the edge on the UAV. As a result, we did not leverage it for the final analysis because the extra computation and inference time required to run the model would contrast with our desire to run all computations fully at the edge. Appendix D presents more details on the training parameters for these models.

Just like its predecessor, COLIP underwent contrastive training using an InfoNCE (information noise contrastive estimation) loss to align olfactory and visual embeddings, encouraging the model to associate corresponding pairs closely in the latent space. These modality pairs encode together, effectively binding them in the process. We optimized these encoders through the InfoNCE loss [27] shown below.

$$L_{I,M} = -\log \frac{\exp\left(q_i^\intercal k_i / \tau\right)}{\exp\left(q_i^\intercal k_i / \tau\right) + \sum_{j \neq i} \exp\left(q_i^\intercal k_j / \tau\right)} \tag{8}$$

where τ denotes a scalar temperature that controls the smoothness of the softmax distribution and j defines "negatives", or unrelated observations. We presumed each example $j \neq i$ in the mini-batch to be a negative. The loss makes the embeddings q_i and k_i closer in Euclidean distance in the joint embedding space. This consequently aligns I with M conveniently for joint learning. A binary classification loss from the graph head learns

relationships between modalities, distinguishing between matching and non-matching olfaction–vision pairs.

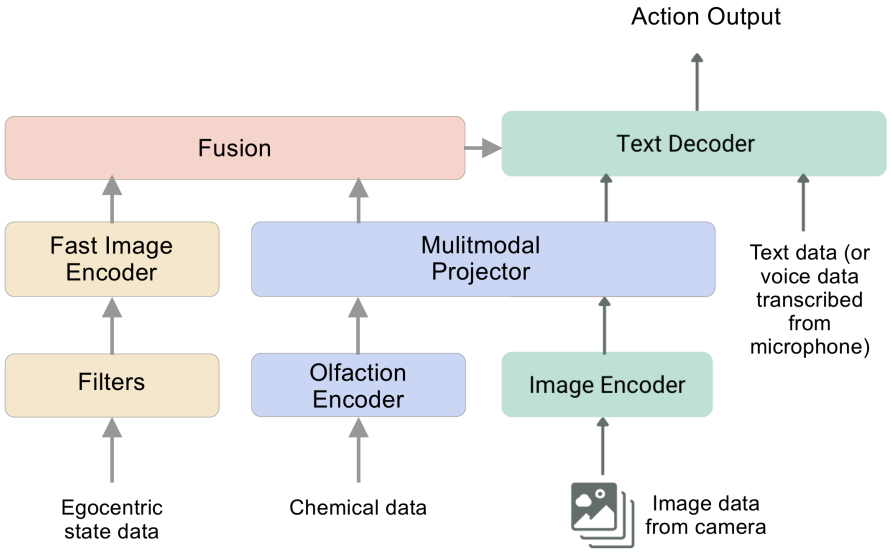


Figure 5. Olfaction–vision model architecture. The blue and green boxes construct *Component 1* based on *COLIP*. The yellow boxes construct *Component 2*. The output from both models informs reasoning for navigation.

Many approaches to robotic control struggle with a core tradeoff: the backbones of large vision models are highly general, but too slow for real-time operation, while traditional robot visuomotor policies are fast, but lack generalization [28]. We resolved this tradeoff by introducing a dual-system architecture. Component 1 is the OVM that runs at 0.1–0.2 Hz. It handles high-level scene understanding and multimodal comprehension, enabling broad generalization across objects, environments, and goals. Component 2 is a high-speed reactive vision–olfaction module containing a fine-tuned YOLOv11 [29] for rapid vision inference, running inference at 1–2 Hz. Component 1 mainly confirms whether the acceptance criteria for locating the odor source have been achieved (see Section 3.8 for more on this). Component 2 synchronizes at the same sampling rate as the olfaction sensors, and helps convert the latent representations from the OVM into precise, continuous UAV actions.

This decoupled design lets each component function at its optimal temporal scale. Component 2 can deliberate slowly over abstract goals, while Component 1 executes rapid, low-latency commands in response to dynamic conditions. This relieves the need to model increased complexity in our plume models and simulations. Furthermore, the separation of both components allows for independent development and iterations.

3.6. Navigation and Plume Models

Plume models are inherently complex and require advanced computational fluid dynamics engines to achieve reasonable approximations. We sought a set of algorithms that generalize well to all chemical compounds and follow simple rules based on environmental evidence, not unlike those that insects follow [12,30,31]. The primary navigation model was a simple bout detection algorithm [6,32] in which the first and second derivatives align with the short and long moving averages of the olfactory signal coupled with inertial data to determine the UAV actions. For brevity, we denote this model as *olfactory inertial odometry* (*OIO*), building off the principles from [33]. We modeled plumes with simple Gaussian time-series processes, inserting blanks [34] into the data to resemble wind shifts and plume volatility. The robot has seven actions available depending on temporal plume

dynamics: surge forward, cast in one of four directions, pause, or land. Selecting the "land" action indicates that the UAV is confident that it has found the plume source. Our perception reasoning is highly susceptible to hysteresis due to the ability of shifting plume dynamics. We employed tuned Kalman filters and sensor synchronization to counter this. We implemented a custom plume environment using the Gymnasium framework [16] for additional simulations in plume tracking. We designed factors such as the temperature, relative humidity, barometric pressure, wind direction, wind magnitude, wind sparsity, and air chemical composition to all be tunable hyperparameters of the plume. The design follows Gaussian plume principles [31] and a Dryden turbulence model [35]. Appendix F presents more details about this plume environment and its construction.

We implemented a second navigation model based on reinforcement learning (RL) to provide a more comprehensive assessment of the navigation. *Q-learning* [36] is a standard temporal difference RL algorithm that many applications use. Due to the dynamism of plumes and the UAV's frequent probability to travel off plume, we employed eligibility traces to harmonize immediate and long-term rewards. $TD(\lambda)$ algorithms leverage eligibility traces to give credit to more recent λ states instead of only the current state ($\lambda = 0$). Consequently, we leveraged the *TD(λ)* derivatives of *Q*-learning and Expected SARSA, called $Q(\lambda)$ and *Expected SARSA(λ)*, respectively [37]. While we trained both models in the simulation, we found Expected SARSA (λ) to be the model most robust to reasoning about blank pockets of air, and thus selected it as a finalist for evaluation in our experiments. Expected SARSA(λ) proved most robust for air with continuous gradients, but this scenario is not practical for real-world applications. Appendix E presents the exact training parameters for these models.

With either model, we discretized the aircraft's movements such that it can only move 10 cm/s or $\frac{\pi}{2}$ radians/s at each time step. In essence, we gave identical linear and angular acceleration commands for every command. We followed a policy that allows the UAV to move and wait for 2 s to sample continually until done. For each "move" command, the magnitudes of acceleration and velocity were identical to provide a level of experimental control and reproducibility. The consistency here also ensured that we did not encounter further nonlinearities induced by the payload bias. We found that making the actions continuous was not necessary, as doing so added more complexity to our simulations and did not substantially improve them. In addition, the infrared sensors enabled the UAV to make a decision on how to move forward. We acknowledge that enabling continuous action spaces could present an opportunity for future work.

3.7. Course

Finally, we transferred our solution to the real world with flight tests on the UAV. We deployed our UAV over a 200 m^2 course, depicted in Figure 6. We started by placing 100 mL of ethanol within a home diffuser in Room 2, as the checkered flag indicates. The diffuser diffused the ethanol at a rate of 2.16 mL per minute. We placed the diffuser near an 80 cm home fan that exhausted air at 1415 L per minute in order to develop a small plume for traceability of the compound. The robot started at the far adjacent end of the map, as the UAV icon in the figure indicates. We observed 5 runs each with the target compound lying in both Rooms 1 and 2. At the beginning of each run, we gave the compound 5 min to disperse and form the start of a plume before we commanded the UAV for takeoff. After takeoff was initiated, no humans intervened with the UAV until it sent a response back to the ground station that it had found the source of the target compound. Between each run, we vented the course and took air measurements to ensure that the next iteration would begin with a clean baseline and to encourage as reproducible results as possible. Due to the length of time needed to properly vent the course, the uncontrollable HVAC schedules

within the test building, and other experimental controls needed to re-initialize to a stable environment, the number of statistical runs we could perform was limited. As a result, for each of the four tasks, we analyzed five trials. We note that prior work from [4,6] had a low count of statistical runs (identical to ours), which we suspect these factors also caused.

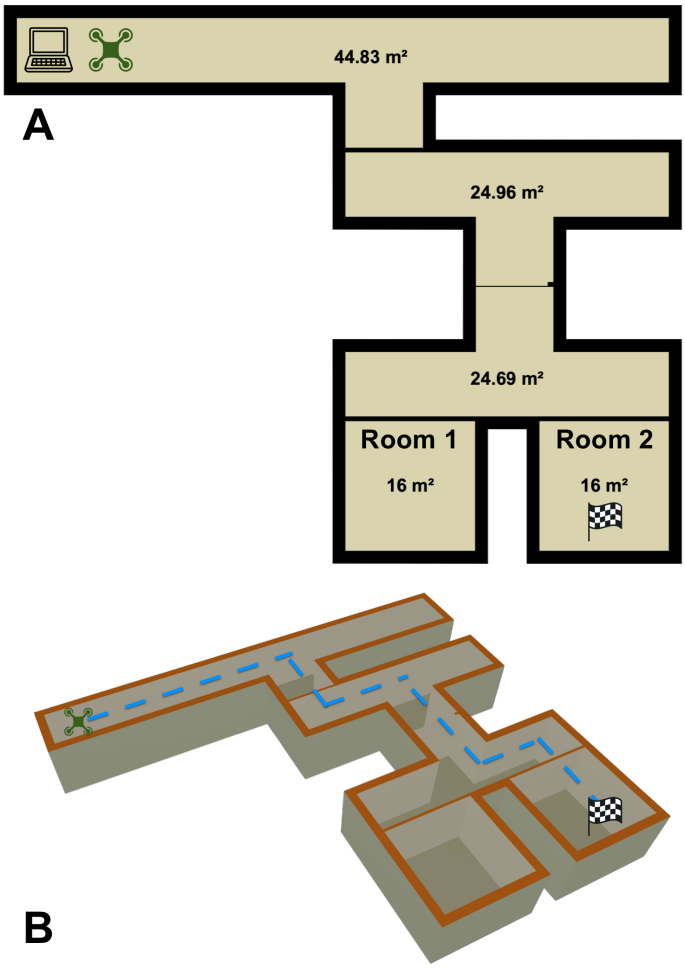


Figure 6. Illustration of the course developed for olfactory navigation by the UAV. (**A**) shows a bird's-eye view of the course, where the ground station (laptop icon) is located at the top left and the UAV (quadcopter icon) takes off next to it; Room 2 hosts the target compound and the plume is formed from here. (**B**) shows an isometric view of the ideal path that the UAV should pursue to locate the source. The blue dotted line indicates the minimum optimal path that the UAV should travel to localize the odor source, though this was never achieved.

3.8. Task Termination Criteria

One defining contribution of our work is how a UAV decides that it is "done". For the robot to determine whether it has found the source of the gas, it must evaluate a hypothesis about whether it believes it has found the highest concentration of the plume given its observed samples. On our selected UAV, the IMU bias from the added payload was significant, making long-term positional estimation very difficult. The UAV could perform all of the maneuvers commanded, but the added payload dampened the magnitude of each command. We found this dampening factor non-linear and partially dependent on battery life. This made the error of the commands difficult to model, but we found short-term estimation more practical to filter via extended Kalman filters and reasoned via OIO accordingly. See Figure 7 as an example of the IMU replay that occurred over the map detailed in Figure 6.

We concluded that the UAV reached the gas source when it landed within 1 m of the diffuser. To evaluate when the UAV reached the gas source, we constructed a minimum-variance unbiased point estimator $\hat{C}$ to approximate the true highest concentration of the gas within the environment, given by:

$$\hat{C} = m\frac{k}{k+1} - 1 \tag{9}$$

where m is the largest concentration that the olfaction sensors observed and k is the total number of samples observed. We assumed that, once a chemical packet has been observed, it can be observable again, as in drawing with replacement [38,39].

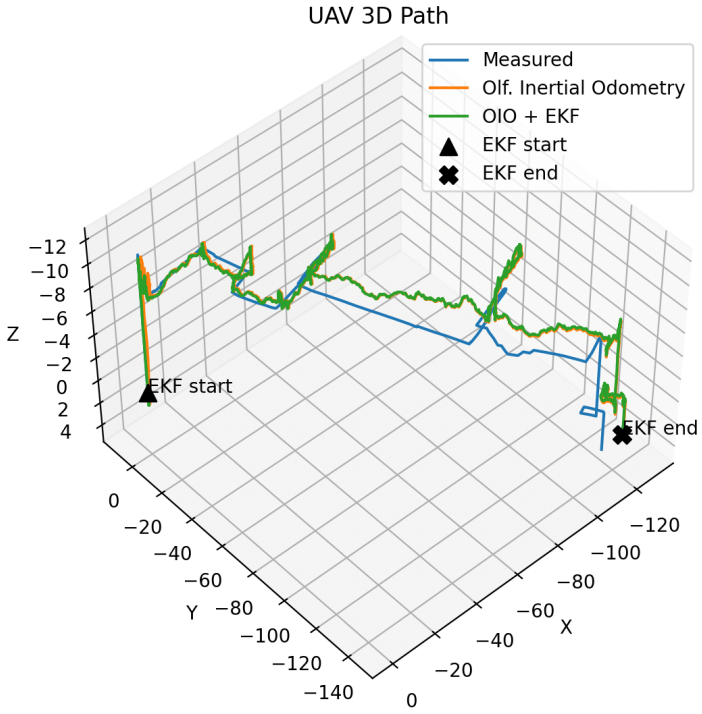


Figure 7. The IMU replay of the UAV traveling over the whole course is shown here. Unfiltered olfactory inertial odometry (OIO) [33,40] and filtered OIO via EKF are shown alongside naked measurements from the IMU. The addition of the olfaction sensors and OPU as payload significantly biased the IMU. This made compound mapping via long-term inertial odometry nearly impossible and drove our decision for the task termination criteria in Section 3.8. Measurements are in centimeters.

The point estimate becomes reliable only after long sessions with several observed samples. However, because we desired a generalized solution to both short and long exploration sessions, we computed a confidence interval for $\hat{C}$. We computed the p-th and q-th quantiles of the sample maximum m to form the interval $[\hat{C}q^{1/k}, \hat{C}p^{1/k}]$, which yielded the corresponding confidence interval for the maximum estimated gas source concentration over the full population of gas samples:

$$\left[\frac{m}{q^{1/k}}, \frac{m}{p^{1/k}}\right] \tag{10}$$

As an example, assume that the UAV has observed 20 samples and believes that it has found the source of the gas with an observed final concentration m. The point estimate

for $k = 20$ samples would be $1.05m$ based on Equation (9). By Equation (10), taking a 95% confidence interval would return

$$[\frac{m}{0.975^{1/20}}, \frac{m}{0.025^{1/20}}] = [1.001m, 1.2025m] \approx [m, 1.2m] \quad (11)$$

This indicates that, given the observed plume concentrations, we can be 95% confident that the true highest concentration of the plume lies between the current observed highest concentration m and a concentration 20% larger. From here, the UAV can determine whether to keep navigating or to end. In general, we strive to stay within 25% of the expected maximum odor concentration based on a 95% confidence interval, so in this case, the UAV would land and stop navigating.

4. Results

Table 1 shows the final results for our experiments. For all trials, the UAV successfully found the source of the target compound, which was our minimum bar for success. We observed that electrochemical-based olfactory sensors require more time to complete the task in general, and we suspect that this is due to the fact that EC sensors are more sensitive to perturbations in the target compound's concentration, which could cause them to overfit to noise or small eddy currents in the plume. For both MOX and EC sensors, we noted a small bump in performance from adding vision as a modality by which to localize the odor. In all tasks, the UAV only used olfaction to navigate. OIO showed more consistency in navigation times than RL. We suspect that this is due to RL over-optimizing the navigation path by attempting to predict ahead or weighing states too far in the past for future decisions due to the eligibility traces. In future work, we expect that wrapping OIO with RL will become more competitive (or objectively better) than naked OIO with further tuning. In the *olfaction + vision* tasks, the UAV only activated the camera when it could no longer maximize the olfactory gradient, which usually occurred when the UAV was confident that it had reached the target compound based on the termination criteria. When the UAV used only olfaction, we observed that it would spend several seconds slowly moving around the chemical source, trying to maximize the olfactory signal. Because diffusion of the gas is more uniform the closer one is to the gas's source, an apparent gradient may not be easy to determine, and the UAV will waste several seconds moving around very close to the source. This is where vision became additive: using a visual confirmation that the UAV has found the source allows the UAV to not waste time and battery chasing small gradients. As a consequence, vision shortened the navigation times by shaving a few seconds off the end of each trial, where olfaction-only navigation would have kept the UAV flying.

Table 1. Final results.

Method	Sensor Type	Nav. Algorithm	Average Time μ_t (s)	Sigma Time σ_t (s)	Best Time β_t (s)
Olfaction	MOX	OIO	98.38	14.84	71.59
Olfaction	EC	OIO	112.22	16.70	97.65
Olfaction	MOX	E.SARSA(λ)	103.99	14.89	82.73
Olfaction	EC	E.SARSA(λ)	121.85	16.77	97.10
Olfaction + Vision	MOX	OIO	94.03	14.68	66.12
Olfaction + Vision	EC	OIO	107.71	17.07	80.50

5. Discussion and Limitations

The most important conclusion we drew from our experiments was that olfaction is enough to navigate when localizing to an odor source. We believe that this is the first documented attempt to enable a robot to navigate purely by scent without any planning,

a priori knowledge of the environment, or location aids such as UWB radio tags. We attempted to show that our results generalize by performing experiments with two different olfaction sensor types. However, we acknowledge that a larger variety of chemical sensors exists. In addition, we performed our work here over only one chemical compound: ethanol. This was intentional in order to build off of the assumptions and work by Burgues et al. [4] and Duisterhof et al. [6], as they demonstrated olfactory navigation with radio tags and metal oxide sensors in tracking ethanol. With our work here, we attempted to establish a generalizable framework that builds off the accepted state of the art, and that can enable intuitive extrapolation of our hardware and algorithms to other compounds with little or no tuning. We are performing ongoing work to support this and demonstrate said framework on a variety of gases in outdoor conditions and wider kinematic envelopes.

We note that our results show that the UAV localized itself to the chemical source on *every* trial, while the work of Shigaki et al. [5] noted a non-perfect localization rate. We are also performing ongoing work to demonstrate the generalization of our framework over other environments, chemical compounds, and sensor types such as graphene-based sensors, spectroscopy sensors, and photo-ionization detectors. We acknowledge that exact replication of any olfactory navigation task is difficult due to the inherent non-linearities of gas dynamics and the difficulty in replicating exact initial chemistry conditions. As a result, the focus of our work here was to establish a simple, reproducible, and generalized framework on which other researchers can build to conduct additional experiments that will lead to further improvements and an increased reproducibility.

6. Conclusions

Scent-based navigation is a very nuanced robotics task. Organisms that leverage olfaction to perceive the world optimize over various criteria, yet none of them are standardized. With our framework and experiments here, we show that, to localize a UAV to a chemical source, olfactory perception is enough. We incorporated many nature-inspired principles to define methods for the first instance of stereo olfaction–vision odor localization and demonstrate their practicality with real aerial robots. We make our code, simulations, hardware schematics, and datasets available to the community to encourage advancement in the field. We hope that the techniques presented here encourage more work in olfactory robotics and bring us one step closer to enabling the sense of smell for machines.

Author Contributions: Conceptualization, K.K.F. and O.D.; methodology, K.K.F.; software, K.K.F.; validation, K.K.F., O.D. and L.K.; formal analysis, K.K.F.; investigation, K.K.F.; resources, K.K.F.; data curation, K.K.F.; writing—original draft preparation, K.K.F.; writing—review and editing, K.K.F., O.D., L.K. and R.P.; visualization, K.K.F.; supervision, O.D. and L.K.; project administration, K.K.F.; funding acquisition, O.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All code, data, hardware schematics, and designs are available at https://github.com/KordelFranceTech/ChasingGhosts (accessed on 26 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Overview of Related Work

We present a high-level comparison of our work to previous research in olfaction navigation. We strove to maintain parity with existing research and build off of their assumptions and methods for a level of experimental control while integrating our own optimizations. These papers are the closest we found to our research.

Table A1. Comparison to relevant prior works.

Aspect	Shigaki et al. [5]	Burgués et al. (2019) [4]	Duisterhof et al. (2021) [6]	Our Work
Robot Type	Pocket-sized quadcopter (2018) or ground crawler (2023); single unit.	Nano quadcopter (Crazyflie).	Swarm of nano quadcopters (Crazyflie).	Modified DJI Tello UAV; single unit, but with custom harness for stereo sensors.
Sensors	Single MOX sensor (2018) or dual alcohol + wind sensors (2023).	Single MOX (TGS 8100).	Single MOX per drone + laser rangers.	Dual MOX/EC sensors in stereo; IR for obstacles; vision complementary.
Navigation	Bio-inspired (surge-cast/zigzag).	Predefined sweeping path; bout detection.	Mapless bug algorithm with PSO; real-time swarming.	Adaptive TD learning (OIO/Q(λ)); real-time plume modeling with temporal differences.
Course Size	Small (1.5–2 m start distance).	160 m^2 indoor.	10 $\times$ 10 m (100 m^2) with obstacles.	200 m^2 indoor with rooms.
Success Rate	80–90%.	N/A (mapping error 1–2 m).	92% in sim; 11/12 real runs.	100% across 20+ trials.
Positioning Aids	None (onboard only).	External UWB anchors.	Onboard UWB for relative localization.	None (only onboard IR + olfaction).

Appendix B. Control Algorithms

We based the autopilot control loop on a proportional-integral controller with rate feedback, or *PIR* for short. Figure A1 below shows the control diagram. Table A2 explains each parameter. Table A3 shows the values for each gain of each controller.

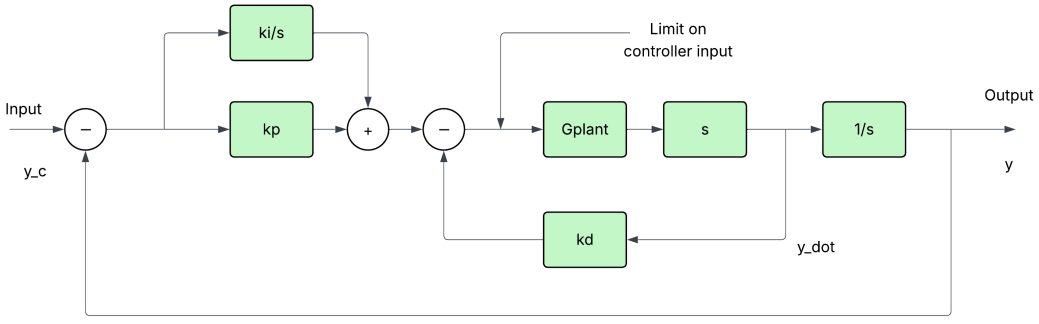


Figure A1. We designed the control loop to model the DJI Tello control algorithms and understand how to sequence the olfaction sensors with the other sensors.

Table A2. PI Controller with rate feedback.

Parameter	Definition
G_{plant}	Controller
y_c	Closed-loop command
y	Current system response
$\dot{y}$	Rate feedback
dt	Timestep, in seconds
k_i	Integral gain
k_p	Proportional gain
k_d	Derivative gain

Table A3. PID gains for each PIR controller.

Parameter	Value
k_p (roll)	0.010
k_i (roll)	0.002
k_d (roll)	0.006
k_p (altitude)	0.070
k_i (altitude)	0.015
k_d (altitude)	0.060–0.080
k_p (pitch)	0.010–0.012
k_i (pitch)	0.002
k_d (pitch)	0.010
k_p (yaw)	0.100–0.200
k_i (yaw)	0.012
k_d (yaw)	0.100
k_p (velocity lateral)	0.100
k_i (velocity lateral)	0.050
k_d (velocity lateral)	0.000–0.010
k_p (velocity longitudinal)	0.100
k_i (velocity longitudinal)	0.050
k_d (velocity longitudinal)	0.000–0.010

Appendix C. Inference Technique for Electrochemical Sensors

For metal oxide sensor measurements, we used the principles from Dennler et al. in [17]. For measurements with electrochemical sensors, we borrowed the fast inference technique for chronoamperometry from [3]. We evaluated two sampling rates for chronoamperometry: 10 Hz and 100 Hz. We determined that, while 10 Hz was computationally cheaper, 100 Hz gave us more points from which the fast inference algorithm can infer, so we set 100 Hz as the sampling rate for deployment on the UAV and cut the chronoamperometry measurement after 1.0 s. Figure A2 shows plots for assessing how well the inference technique predicted the full 6-s diffusion sequence for a 100 Hz measurement frequency, and Figure A3 shows the same for a 10 Hz measurement frequency.

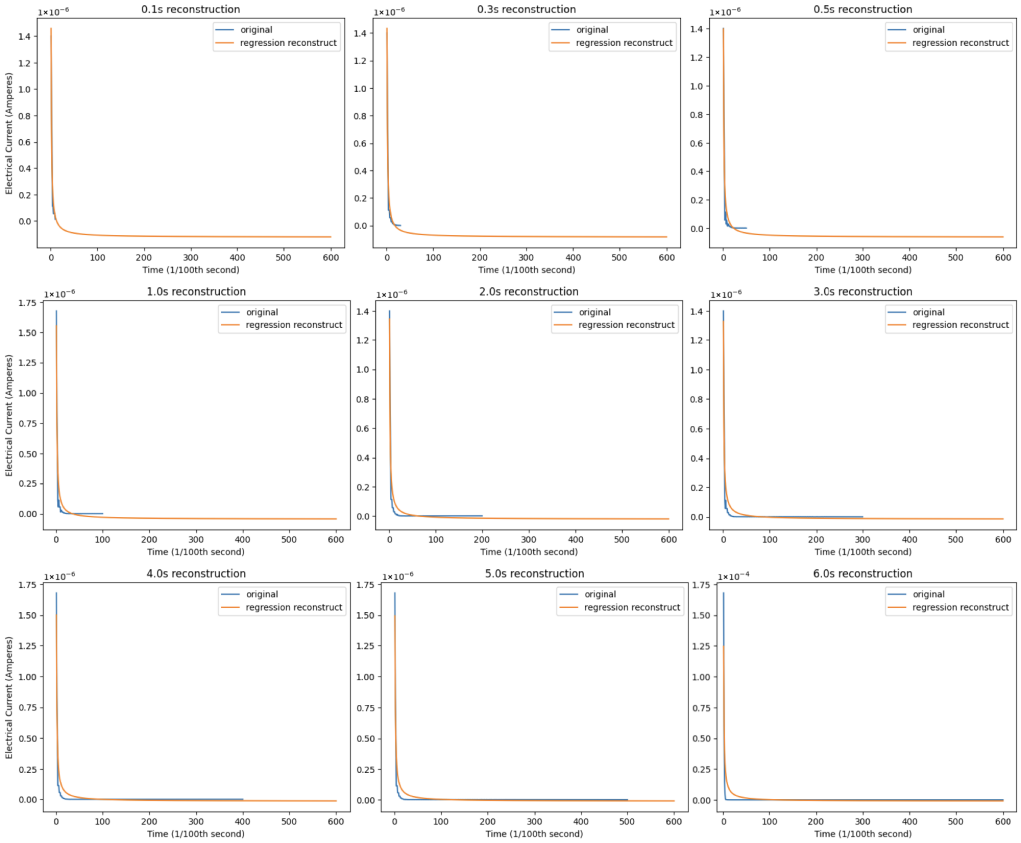


Figure A2. Prediction of the full 6-s diffusion current for chronoamperometry over various time periods at a 100 Hz measurement frequency.

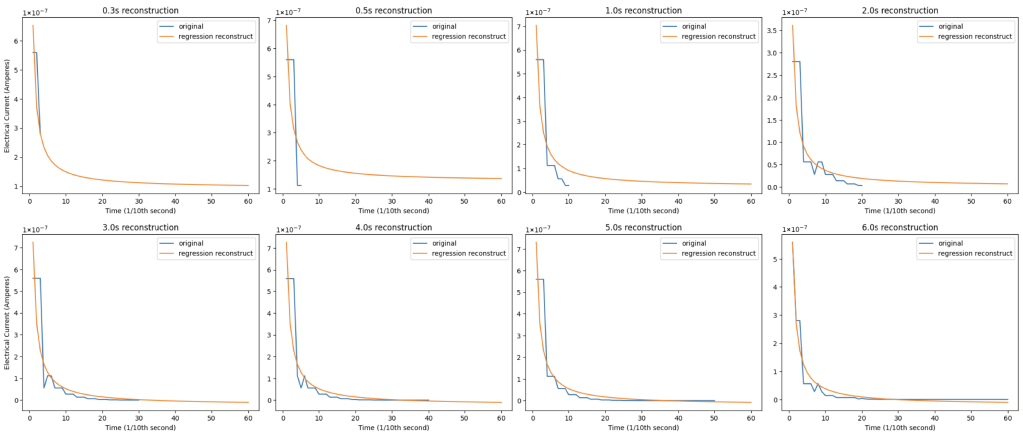


Figure A3. Prediction of the full 6-s diffusion current for chronoamperometry over various time periods at a 10 Hz measurement frequency.

Appendix D. Training of Olfaction–Vision Models

We trained an encoder to compress olfactory data into a format that could jointly learn with CLIP through a graph associator model. We intentionally kept this encoder simple in architecture due to issues with exporting advanced layers that we observed with the graph associator models (more on this below). The encoder is a very deep model, but we intentionally over-parameterized it due to the low additional computational overhead and because there is a general lack of open-sourced olfactory training data from which to learn (depth and over-parameterization have been shown to help models generalize in the absence of plentiful data [41]). Details on the encoder training appear below:

Table A4. Olfaction encoder.

Parameter	Value
Layers	12
Layer types	Fully connected only
Input dimension	138
Output dimension	512
Batch size	16
Learning rate	1×10^{-4}
Activation functions	ReLU
Epochs	1100
Loss function	Information noise-contrastive estimation
Loss function temperature	0.07
Optimizer	Adam

We trained two graph associator machine learning models to learn olfaction–vision relationships: a fully connected (FC) neural network and a graph-attention (GAT) neural network. We optimized the simple FC network to export for use at the edge without external API calls. This model has a lower computational demand, lower layer complexity (and thus, easier exportability), and faster inference. Details on its architecture appear below:

Table A5. FC graph associator.

Parameter	Value
Layers	4
Layer types	Fully connected only
Input dimension	512
Output dimension	512
Embedding dimension	512
Batch size	16
Learning rate	1×10^{-4}
Activation functions	ReLU
Epochs	1100
Loss function	Information noise-contrastive estimation
Loss function temperature	0.07
Optimizer	Adam

The GAT associator learned more complex relationships between CLIP and olfactory representations. However, we noted a lack of consistency in the inference performance between what we observed when training the models and what we actually observed when exported to the UAV firmware. As a result, we did not move forward with this architecture as the baseline model. Details on GAT training appear below:

Table A6. GAT graph associator.

Parameter	Value
Layers	4
Layer types	2 fully connected, 2 GAT convolution
Input dimension	512
Output dimension	512
Embedding dimension	512
Batch size	16
Learning rate	1×10^{-4}

Table A6. *Cont.*

Parameter	Value
Activation functions	ReLU
Epochs	1100
Loss function	Information noise-contrastive estimation
Loss function temperature	0.07
Optimizer	Adam

Figure A4 below shows the loss plot of the final training run for both graph associator models jointly trained with the olfactory encoder + CLIP.

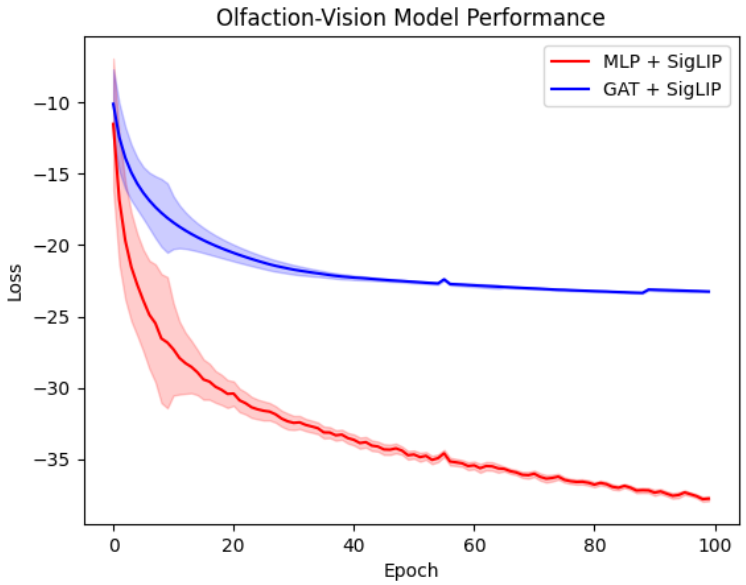


Figure A4. Training losses for both olfaction–vision models. Note that the GAT-based model is more accurate, but requires more computations and inference time, and thus violates our constraints for full edge computation. As a result, we used the MLP-based model for our final results.

References [20,42] respectively provide full code to reproduce the training data and results for both models.

Appendix E. Training of Reinforcement Learning Algorithms

In order to construct any learning algorithm for navigation, we need to solidify the states for which the UAV could possibly reside and the actions available to it in each state. We deduced that, for each of nine different states, seven actions were possible: turn slightly left (45°), turn slightly right (45°), turn hard left (90°), turn hard right (90°), surge forward, cast left, or cast right. Table A7 below details the nine states.

Table A7. State table for each navigation agent.

	Off-Plume	On-Plume (Low Conc.)	On-Plume (High Conc.)
Left < Right	0	3	6
Left = Right	1	4	7
Left > Right	2	5	8

We selected two TD(λ) algorithms for the simulation as a possible navigation algorithm: Q(λ) and Expected SARSA(λ).

Q-learning is an off-policy temporal-difference (TD) method that directly estimates the optimal action–value function without requiring adherence to the policy that drives exploration [36]. It learns a policy that selects state–action transitions yielding the highest reward and ranks among the most widely adopted TD algorithms in reinforcement learning due to its computational simplicity and its effectiveness in approximating the long-term cumulative return.

Greedy Q-learning performs well in settings where rewards are clearly defined for each state. From an intuitive standpoint, an agent that optimizes the expected reward is better positioned to approximate the latent reward structure of an environment than an agent that strictly maximizes instantaneous rewards, as in standard Q-learning. This distinction becomes particularly relevant in environments that contain continuous gradients of a target compound. However, complications arise when the environment contains regions devoid of the target signal. During early training, reward assignments in such regions are highly uncertain, causing a Q-learner to maximize an unreliable objective. In contrast, an Expected SARSA learner [37] estimates the expectation over possible actions, thereby approximating the policy itself and effectively reducing loss. Although regularization can partially mitigate this issue, it does not fully resolve the instability that erroneous reward estimates introduce.

In much of contemporary reinforcement learning research—especially in domains with directly observable environments—Q-learning and its variants represent a practical compromise between the computational cost and the performance. Our experimental results, however, indicate that the effectiveness of Q-learning depends strongly on the agent's ability to observe rewards that faithfully reflect the ground truth. When reward signals are reliable, a greedy Q-learner is likely to achieve a superior performance. For this reason, we posit that Expected SARSA constitutes a more robust choice in general, with its update rule given by

$$Q^*(s,a) = Q(s,a) + \alpha \left[R(s,a,s') + \frac{\gamma}{n} \sum_{i=1}^{n} Q(s_i', a_i') - Q(s,a) \right]$$

Expected SARSA further allows explicit control over the degree of greediness in the policy. As ϵ converges to 1, Sutton and Barto [37] demonstrate that the method converges to selecting the action with the maximal cumulative reward, thereby becoming equivalent to Q-learning. This flexibility enables Expected SARSA to combine the benefits of both paradigms by adjusting policy learning according to the reliability of reward estimates and the self-paced learning parameter λ. Accordingly, the early stages of training resemble SARSA:

$$Q^*(s,a) = Q(s,a) + \alpha[R(s,a,s') + \gamma Q(s',a') - Q(s,a)]$$

which dampens excessive loss and gradually transitions toward Q-learning, moderating the high loss as the policy improves:

$$Q^*(s,a) = Q(s,a) + \alpha \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

This formulation allows the algorithm to operate in both on-policy and off-policy regimes.

A closer inspection reveals that the principal distinction between Expected SARSA and Q-learning lies in replacing reward maximization with reward averaging. Consequently, Expected SARSA incurs a greater computational cost, since the algorithm must evaluate an expectation over actions at each update. Nevertheless, this added expense yields an increased robustness and accuracy during co-training by reducing the volatility that

early reward mis-estimation causes. This advantage is particularly pronounced in the initial phases of learning, when unrewarded trajectories may receive highly inaccurate values due to frequent blank observations. Because the objective is to infer rewards for unobservable states with greater fidelity, the additional computational burden may be warranted. Expected SARSA thus provides a principled mechanism for interpolating between on-policy and off-policy learning while offering an inherent means of controlling the reward loss. In our experiments, ϵ in Expected SARSA increased as a function of the self-paced learning parameter.

In light of the above, the simulation showed that Expected SARSA(λ) was more robust to perturbations in the environment, reasoning about blanks, and training parameter changes, so we selected it as the finalist to deploy onto the UAV for evaluation. We trained both algorithms with the following configuration. Details on its architecture appear below:

Table A8. Q(λ) and Expected SARSA(λ) configurations.

Parameter	Value
Number of states	9
Number of actions	7
Episodes	10,000
γ (discount factor)	0.9
α (learning rate)	1×10^{-4}
ϵ (greediness)	1.0
ϵ-decay	0.999
λ (TD step size)	0.8

Figures A5 and A6 present the training losses for the final configurations of both algorithms. Figure A5 shows how $Q(\lambda)$ performs better in a simulation when no blanks are modeled. In other words, no spots within the navigation environment are absent of the target compound. In reality, this is not probable, and it is prudent to model "blanks" of pockets of air that contain virtually no trace of the target compound [34]. Note the tighter variance bands around $Q(\lambda)$ toward the end of training for the simulation we modeled without blanks.

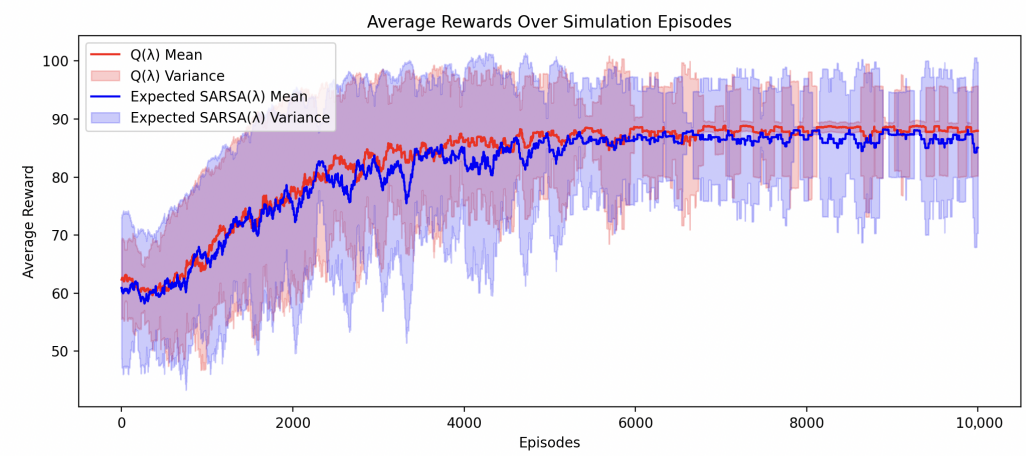


Figure A5. Training losses for both Q(λ) and Expected SARSA(λ) algorithms, with no blanks modeled.

When we modeled blanks in the environment, Expected SARSA(λ) out-performed $Q(\lambda)$. We expect that this is due to Expected SARSA being more robust to noise than greedy Q-learning, as noted by [37,43,44].

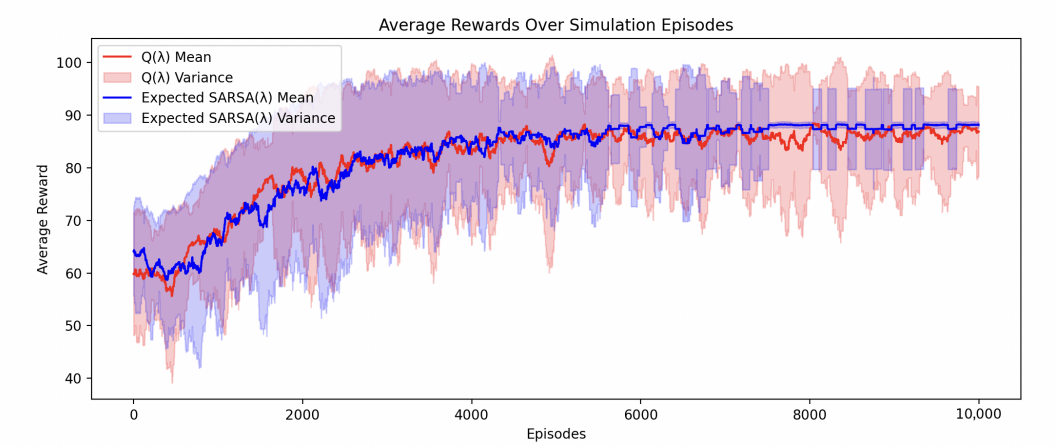


Figure A6. Training losses for both Q(λ) and Expected SARSA(λ) algorithms, with blanks modeled.

At the conclusion of training, a Q-value table emerged, showing the credit assignment to each action. Figure A7 below shows the final Q-values for the final policy.

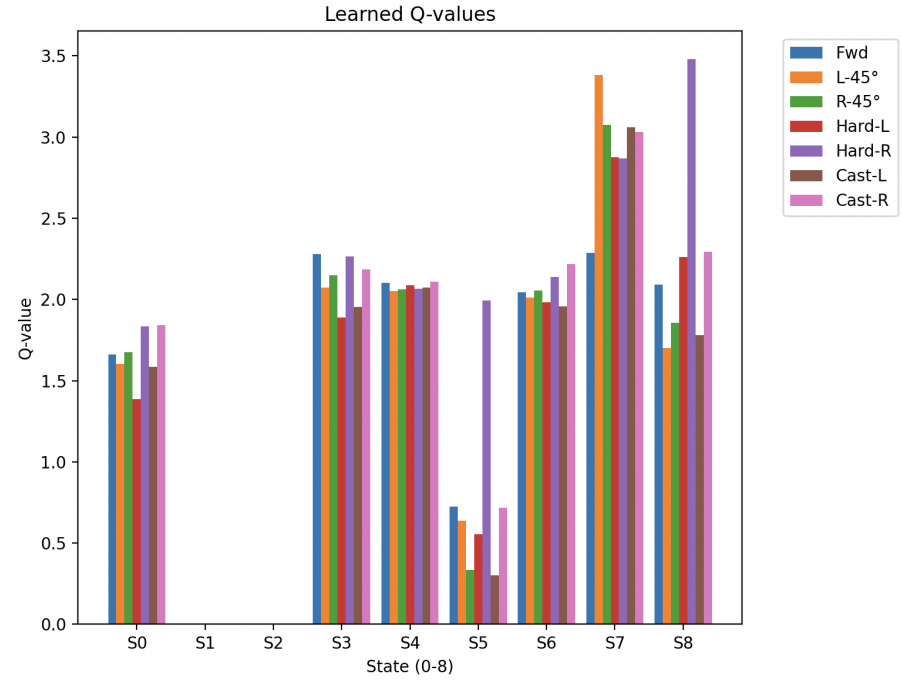


Figure A7. Q-values of the final reinforcement learning policy.

Appendix F. Plume Model Environment

We built a plume environment to understand the behavioral dynamics of gases over time. We used the environment as a model to evaluate the effects on air volatility from changing different parameters. We later used it to help model navigation algorithms for the UAV. We built the environment on top of the Gymnasium framework and included it within the code repository associated with this paper.

We note the configuration to reproduce our results below.

Table A9. Plume environment parameters.

Parameter	Value	Units	Description
Diffusion	1.0	Dimensionless	Rate of gas dispersion per timestep
Sparsity	0.0	Dimensionless	Useful for modeling blanks over long distances
Temperature	20.0	°C	Assume constant air temperature
Relative humidity	50.0	%	Assume constant relative humidity
Air density	1.225	kg/m^3	Assume constant air density
Wind speed	1.0	m/s	Average windspeed along x-direction
Emission rate	1.0	kg/s	Rate of gas emission from source
Pasquill stability	D	Class A–G	General air volatility characteristic
Source height	1.0	m	Max 3.0 m due to flight envelope
Obstacles	5	Integer count	Only square block obstacles modeled

Appendix G. Olfaction Sensor Breakout Board

Several different olfaction sensors exist. In order for us to understand which ones are best suited for the detection of ethanol, we designed a breakout board that assessed several olfaction sensors in parallel. We documented and compared the responses to concentration changes in ethanol, leading to our ultimate selection of the metal oxide and electrochemical sensors as the best fit for tracking ethanol. The code repository associated with this paper contains the electrical schematics and Gerber files for this circuit board in order to encourage olfactory research. Figure 3B shows the final assembled breakout board. Figure A8 shows the electrical schematic of the board. We accumulated data for assessing the maximum sensor response with this board via the Scentience application [45].

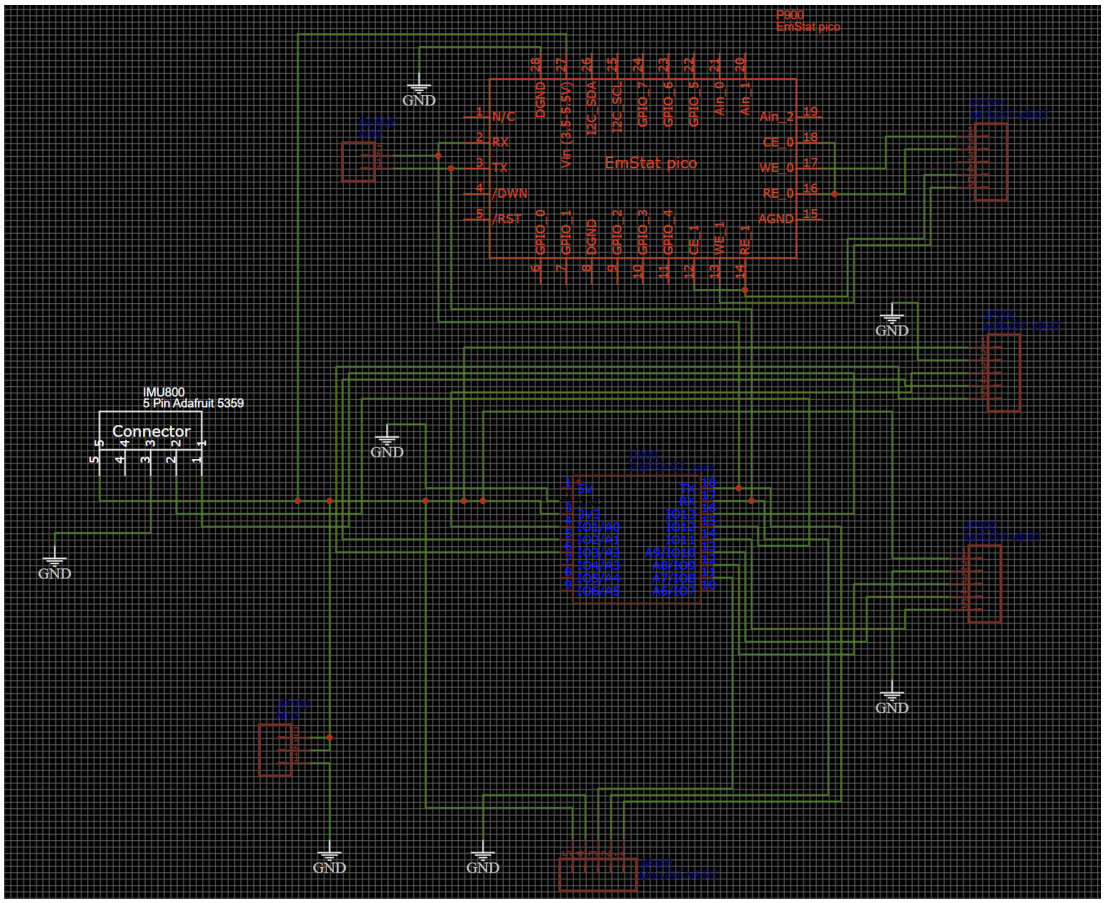


Figure A8. Electrical schematic of the breakout board above. The code repository associated with this manuscript provides Gerber files and full schematics for this board.

Appendix H. Mechanical Modifications to the UAV

One can additively manufacture the full UAV modification kit for both electrochemical and metal oxide sensing configurations on a single 200 mm × 200 mm print bed. Figure 3B gives a diagram of the CAD developed for this project. To adapt the UAV for olfactory navigation, we designed a mechanical harness that equipped the machine with the OPU, olfaction sensors, battery, ballast, and infrared sensors. We designed all components in SolidWorks and iterated several times; the final iterations reside in the code repository under the */cad* subdirectory. We followed generally accepted mechanical design principles for aeronautics. One can 3D-print the full kit on a single 500 mm × 500 mm print bed using PLA plastic. We used a 35% infill to balance the durability and weight. The code repository contains a full bill of the materials.

References

1. Dennler, N.; Rastogi, S.; Fonollosa, J.; van Schaik, A.; Schmuker, M. Drift in a popular metal oxide sensor dataset reveals limitations for gas classification benchmarks. *Sens. Actuators B Chem.* **2022**, *361*, 131668. [CrossRef]
2. Dennler, N.; van Schaik, A.; Schmuker, M. Limitations in odour recognition and generalization in a neuromorphic olfactory circuit. *Nat. Mach. Intell.* **2024**, *6*, 1451–1453. [CrossRef]
3. France, K.K. Chronoamperometry with Room-Temperature Ionic Liquids: Sub-Second Inference Techniques. *arXiv* **2025**. [CrossRef]
4. Burgués, J.; Hernández, V.; Lilienthal, A.J.; Marco, S. Smelling Nano Aerial Vehicle for Gas Source Localization and Mapping. *Sensors* **2019**, *19*, 478. [CrossRef] [PubMed]
5. Shigaki, S.; Fikri, M.R.; Kurabayashi, D. Design and Experimental Evaluation of an Odor Sensing Method for a Pocket-Sized Quadcopter. *Sensors* **2018**, *18*, 3720. [CrossRef] [PubMed]
6. Duisterhof, B.P.; Li, S.; Burgués, J.; Reddi, V.J.; de Croon, G.C.H.E. Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 9099–9106. [CrossRef]
7. Zhang, L.; Shi, J.; Wei, X.; Feng, L. OEVS-fusion: Olfactory-enhanced visual semantic recognition framework for ground stain detection in indoor environments. *Sens. Actuators B Chem.* **2025**, *440*, 137902. [CrossRef]
8. Zhang, L.; Feng, L.; Wei, X.; Chen, S.; Zhang, Y. MSA-TICNN: A dual-modal framework integrating exhaled e-nose data to assist ECG-based psychological stress evaluation for cognitive enhancement. *Sens. Actuators B Chem.* **2026**, *450*, 139215. [CrossRef]
9. Hassan, S.; Wang, L.; Mahmud, K.R. Robotic Odor Source Localization via Vision and Olfaction Fusion Navigation Algorithm. *Sensors* **2024**, *24*, 2309. [CrossRef]
10. Feng, D.; Li, C.; Dai, W.; Liang, P.P. SMELLNET: A Large-scale Dataset for Real-world Smell Recognition. *arXiv* **2025**. [CrossRef]
11. Ozguroglu, E.; Liang, J.; Liu, R.; Chiquier, M.; DeTienne, M.; Qian, W.W.; Horowitz, A.; Owens, A.; Vondrick, C. New York Smells: A Large Multimodal Dataset for Olfaction. *arXiv* **2025**. [CrossRef]
12. Singh, S.H.; van Breugel, F.; Rao, R.P.N.; Brunton, B.W. Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nat. Mach. Intell.* **2023**, *5*, 58–70. [CrossRef]
13. Escoté, D.F. DJITelloPy: Python Library to Interact with the DJI Tello Drone. 2019. Available online: https://github.com/damiafuentes/DJITelloPy (accessed on 23 July 2025).
14. Systèmes, D. Solidworks. Computer-Aided Design (CAD) Software. 2024–2025. Available online: https://solidworks.com (accessed on 31 December 2025).
15. EasyEDA Contributors. EasyEDA—Online PCB Design Tool. 2024–2025. Available online: https://easyeda.com/ (accessed on 19 January 2026).
16. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**. [CrossRef]
17. Dennler, N.; Drix, D.; Warner, T.P.; Rastogi, S.; Casa, C.D.; Ackels, T.; Schaefer, A.T.; van Schaik, A.; Schmuker, M. High-speed odor sensing using miniaturized electronic nose. *Sci. Adv.* **2024**, *10*, eadp1764. [CrossRef]
18. EasyEDA Contributors. Compact MOS Sensor for Pollution Detection—1143_Datasheet MiCS-6814 rev 8. 2015. Available online: https://www.sgxsensortech.com/content/uploads/2015/02/1143_Datasheet-MiCS-6814-rev-8.pdf (accessed on 26 April 2026).
19. Sutton, R.S. Learning to Predict by the Methods of Temporal Differences. *Mach. Learn.* **1988**, *3*, 9–44. [CrossRef]
20. France, K.K.; Daescu, O. Diffusion Graph Neural Networks and Dataset for Robust Olfactory Navigation in Hazard Robotics. *arXiv* **2025**. [CrossRef]
21. Company, T.G.S. The Good Scents Company Information System. Available online: http://www.thegoodscentscompany.com/ (accessed on 8 March 2025).

22. Associates, L. PMP 2001—Database of Perfumery Materials and Performance. Available online: http://www.leffingwell.com/bacispmp.htm (accessed on 8 March 2025).
23. Lee, B.K.; Mayhew, E.J.; Sanchez-Lengeling, B.; Wei, J.N.; Qian, W.W.; Little, K.A.; Andres, M.; Nguyen, B.B.; Moloy, T.; Yasonik, J.; et al. A principal odor map unifies diverse tasks in olfactory perception. *Science* **2023**, *381*, 999–1006. [CrossRef]
24. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**. [CrossRef]
25. Zhong, S.; Zhou, Z.; Dawes, C.; Brianz, G.; Obrist, M. Sniff AI: Is My 'Spicy' Your 'Spicy'? Exploring LLM's Perceptual Alignment with Human Smell Experiences. *arXiv* **2024**. [CrossRef]
26. Zhai, X.; Mustafa, B.; Kolesnikov, A.; Beyer, L. Sigmoid Loss for Language Image Pre-Training. *arXiv* **2023**. [CrossRef]
27. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2019**. [CrossRef]
28. Eschmann, J.; Albani, D.; Loianno, G. Learning to Fly in Seconds. *IEEE Robot. Autom. Lett.* **2024**, *9*, 6336–6343. [CrossRef]
29. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**. [CrossRef]
30. Singh, G.; Lofaro, D.; Sofge, D. Pursuit-evasion with Decentralized Robotic Swarm in Continuous State Space and Action Space via Deep Reinforcement Learning. In Proceedings of the 12th International Conference on Agents and Artificial Intelligence, Valletta, Malta, 22–24 February 2020; Volume 1, pp. 226–233. [CrossRef]
31. Crimaldi, J.; Lei, H.; Schaefer, A.; Schmuker, M.; Smith, B.H.; True, A.C.; Verhagen, J.V.; Victor, J.D. Active sensing in a dynamic olfactory world. *J. Comput. Neurosci.* **2022**, *50*, 1–6. [CrossRef]
32. Schmuker, M.; Bahr, V.; Huerta, R. Exploiting plume structure to decode gas source distance using metal-oxide gas sensors. *Sens. Actuators B Chem.* **2016**, *235*, 636–646. [CrossRef]
33. France, K.K.; Daescu, O. Olfactory Inertial Odometry: Methodology for Effective Robot Navigation by Scent Compensation. *arXiv* **2025**. [CrossRef]
34. Dennler, N.; True, A.; van Schaik, A.; Schmuker, M. Neuromorphic principles for machine olfaction. *Neuromorphic Comput. Eng.* **2025**, *5*, 023001. [CrossRef]
35. Liepmann, H.W. On the Application of Statistical Concepts to the Buffeting Problem. *J. Aeronaut. Sci.* **1952**, *19*, 793–800. [CrossRef]
36. Watkins, C. Learning from Delayed Rewards. Ph.D. Thesis, Department of Computer Science, Kings College, Cambridge University, Cambridge, UK, 1989.
37. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
38. Clark, G.; Gonye, A.; Miller, S.J. Lessons from the German Tank Problem. *arXiv* **2021**. [CrossRef]
39. Simon, C.M. A Bayesian Treatment of the German Tank Problem. *Math. Intell.* **2024**, *46*, 117–127. [CrossRef]
40. France, K.K.; Daescu, O.; Paul, A.; Prasad, S.Olfactory Inertial Odometry: Sensor Calibration and Drift Compensation. *arXiv* **2025**. [CrossRef]
41. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
42. France, K.K. Scentience-OVLC-v1: Joint Olfaction-Vision-Language Classifiers. Hugging Face. 2025. Available online: https://huggingface.co/kordelfrance/Olfaction-Vision-Language-Classifiers (accessed on 26 April 2026).
43. France, K.K.; Sheppard, J.W. Factored Particle Swarm Optimization for Policy Co-training in Reinforcement Learning. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23*; ACM: New York, NY, USA, 2023; pp. 30–38. [CrossRef]
44. France, K.K.; Paul, A.; Banga, I.; Prasad, S. Emergent Behavior in Evolutionary Swarms for Machine Olfaction. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '24*; ACM: New York, NY, USA, 2024; pp. 1479–1487.
45. Scentience. Scentience App: Olfactory Interface Instrument. 2025. Available online: https://apps.apple.com/us/app/scentience/id6741092923 (accessed on 2 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.