OLFACTORYSEARCHRL: A MEASURED-PLUME BENCHMARK FOR LEARNING TO NAVIGATE BY SMELL

Anonymous authors
Paper under double-blind review

ABSTRACT

Olfactory navigation requires an agent to infer a hidden source from intermittent local measurements while choosing where to sample next. Progress in learning this behaviour depends on environments that combine realistic chemical structure with repeatable evaluation. We introduce OLFACTORYSEARCHRL (OSRL), a modular toolbox and reference benchmark that turns experimentally measured turbulent concentration fields into interactive learning environments. OSRL separates plume replay, sensing, observation processing, and control, so researchers can compare policies and representations on a common physical record. A fixed 135-episode track is used to compare navigation policies, where here we evaluate Infotaxis, PMFS, and RNN-PPO variants on three held-out recordings. We additionally retrain the RNN-PPO policy with a dynamic gas sensor model, and showcase successful robotic deployment. In particular, we demonstrate that a robotic agent equipped with chemoresistive gas sensors successfully reaches the source region in complex turbulent flow in 29 of 30 physical trials, without further policy fine-tuning. We release the toolbox and a reproduction package containing configurations, checkpoints, evaluation records, and analysis scripts. OSRL provides an experimental foundation for studying active sensing and state estimation in turbulent chemical environments.

1 INTRODUCTION

A robot searching for a chemical source must decide where to move from brief odour encounters separated by periods with little or no signal. Turbulent transport makes these observations difficult to interpret: acute signal absence can indicate distance from the source, movement outside the plume, or a temporary interruption in chemical delivery (Schmuker et al., 2016). Local concentration measurements usually do not provide a useful gradient towards the source (Vergassola et al., 2007), and the robot's next action determines where it will obtain evidence (Crimaldi et al., 2022). Olfactory navigation therefore brings together active sensing, partial observability, and sequential decision-making in a physical task, with applications in hazard localisation, environmental inspection, disaster management, precision agriculture and embodied AI (Ishida et al., 2012; Francis et al., 2022).

For learning methods, the central challenge is constructing a useful state from an ambiguous observation history. Encounter timing, airflow, movement, and earlier absences can all inform the next decision. Chemical sensors add their own response and recovery dynamics (Martinez et al., 2019; Dennler et al., 2024), so the information available to a controller also depends on transduction. These properties make olfaction a setting in which sensing, representation, and control can be studied together. They also make the experimental infrastructure consequential: changing the plume, sensor, or information supplied to the agent changes the problem being solved.

Existing tools support model-based source-tracking environments, gas-dispersion simulation, and detailed embodiment (Loisy & Eloy, 2022; Monroy et al., 2017; Wang-Chen et al., 2024; Ojeda et al., 2025). However, faithfully reproducing turbulent chemical transport remains difficult: plume dynamics depend sensitively on flow conditions, geometry, and boundary effects, and models may not fully capture the intermittency and spatial structure of physical plumes (Celani et al., 2014). A complementary route is therefore to let agents navigate within *measured turbulent fields*. Spatially resolved recordings preserve these dynamics while allowing repeated evaluation under controlled

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

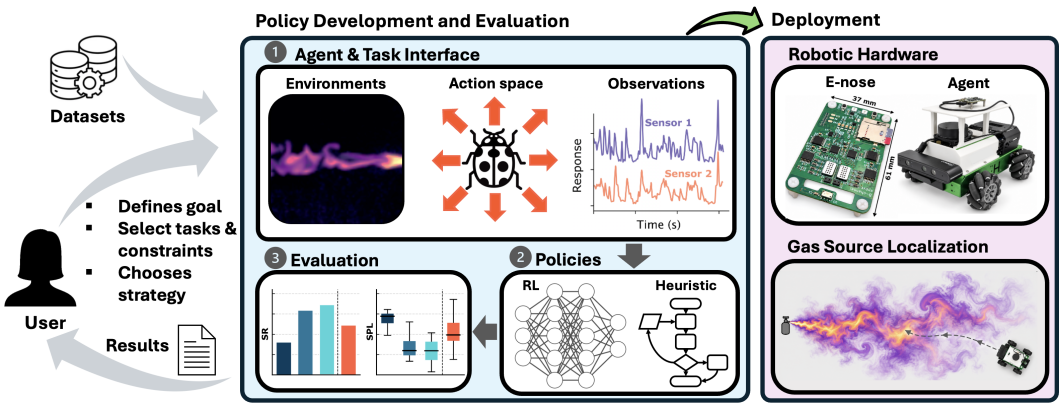

Figure 1: **From measured plumes to robotic search.** `OlfactorySearchRL` combines recorded turbulent concentration fields with configurable tasks, actions, and observations. Researchers develop navigation policies and compare them on a fixed evaluation track. The modular sensing and control interface further supports deployment with a mobile robot and electronic nose / gas sensors; the physical demonstration uses a separately trained controller.

initial conditions. Making them useful for learning requires an executable environment, explicit sensing interfaces, and a reference protocol connecting reported scores to reproducible experiments.

OLFACTORYSEARCHRL (OSRL) provides this connection. It converts planar laser-induced fluorescence (PLIF) recordings of turbulent odour plumes into environments in which actions determine sampling location while the recorded plume evolves in time. Researchers can replace a controller, sensor model, or observation processor through modular interfaces. A fixed benchmark then supplies shared episodes, success geometry, and metric definitions. This combination supports both the development of new methods and comparisons whose assumptions can be inspected. It also offers a practical entry point for researchers without access to a plume-imaging facility or chemical-sensing robot.

The initial benchmark illustrates the questions this foundation makes accessible. We implement Infotaxis (Vergassola et al., 2007), Plume Mapping and Source Localization (PMFS) (Hengenius et al., 2021; Ojeda et al., 2024), and a recurrent variant of PPO (Schulman et al., 2017) (RNN-PPO) with optional source-belief representation, which exhibit different reliability and path-efficiency profiles on the same eligible episode subset. Within RNN-PPO, removing the compact source-belief representation and retraining reduces mean success from 70.7% to 11.3%. The resulting 59.4-percentage-point gap establishes a reference challenge: learning a useful state representation under a specified sensing and training budget. We also demonstrate a route from measured-plume learning to physical robotic search: a separate controller developed with a dynamic sensor frontend operates with a real electronic nose and reaches the source region in 29 of 30 trials.

Our contributions are **(1)** a modular toolbox for interactive navigation in measured turbulent plumes; **(2)** a fixed reference benchmark with planner and learned baselines, episode-level outputs, and explicit evaluation conventions; and **(3)** a study of structured source estimation and policy behaviour, together with a robotic demonstration connecting replay-based development to real chemical sensing and control. We are releasing the toolbox together with the materials needed to regenerate the reported analyses, evaluate the supplied checkpoints, and retrain the learned baselines.

2 RELATED WORK

Search policies and representations. Infotaxis chooses actions using expected information gain about source location (Vergassola et al., 2007). PMFS combines probabilistic mapping and online dispersion simulation (Ojeda et al., 2024). Learned strategies include deep RL for the source-tracking POMDP (Loisy & Heinonen, 2023) and recurrent agents whose emergent behaviours and internal dynamics illuminate plume tracking (Singh et al., 2023; France et al., 2026). OSRL builds on these lines of work by making their environment and sensing assumptions explicit. The RNN-PPO

baselines establish reference performance and expose a representation-learning challenge within this framework.

Olfactory environments. OTTO offers simulation, heuristic and learned policies, a Gym interface, and quantitative evaluation of a model-defined source-tracking POMDP (Loisy & Eloy, 2022). GADEN provides three-dimensional gas-dispersion simulation for mobile robot olfaction (Monroy et al., 2017; Ojeda et al., 2025). NeuroMechFly v2 couples sensory inputs, including olfaction, to a detailed model of fly embodiment and control (Wang-Chen et al., 2024). OSRL contributes a complementary experimental capability: repeated policy evaluation within spatially resolved physical measurements, coupled to configurable sensing and observation processing. Its reference track fixes these interfaces and links scores to episode-level records. This positions measured replay alongside model-defined transport and detailed embodiment as a distinct basis for olfactory-learning experiments.

Measured fields and reusable benchmarks. PLIF supplies spatially resolved scalar measurements (Crimaldi, 2008; Connor et al., 2018); the recordings used here come from an existing turbulent-plume measurement resource (Dennler et al., 2026a;b). Our contribution is their use in an executable decision-making and evaluation framework. As in modular RL environments (Chevalier-Boisvert et al., 2023), reusable interfaces reduce the work required to pose a new learning question. Measured replay adds a particular benefit: policies can be exposed to the same recorded turbulent evolution, even though their actions lead to different local observations.

3 A Toolbox for Learning under Measured Turbulence

3.1 The replay environment

An episode is a finite-horizon POMDP

$$\mathcal{M}_\theta = (\mathcal{S}, \mathcal{A}, P_\theta, R_\theta, \mathcal{O}, Z_\theta, \rho_{0,\theta}, H), \tag{1}$$

where θ specifies the recording, source, geometry, spatial calibration, wind, sensor model, and initial conditions. Let $s_t = (q_t, k_t, z_t; \theta)$ comprise agent pose q_t, recording index k_t, and sensor or processor state z_t. At each step, the environment samples the recorded field at the agent's location, applies the sensor and observation processor, and advances the pose in response to the selected action. Thus policies generate their own observation sequences through movement; the recording supplies the external concentration dynamics.

The measured field follows a *passive-probe approximation*,

$$P_\theta(k_{t+1} \mid s_t, a_t) = P_\theta(k_{t+1} \mid k_t), \tag{2}$$

which excludes agent-induced flow perturbations. Reusing a recording gives different policies a common evolving environment while preserving the consequences of their different sampling decisions. It also permits trajectory inspection against the measured field after evaluation.

The underlying resource contains ten PLIF recordings with source, flow, geometry, and spatial metadata. Conversion prepares the fields for sampling at continuous positions. The evaluated split uses six recordings for training and three for testing; the remaining recording is outside the reported reference results. Configurable spatial and concentration transformations define task variants, and the concentration-field interface permits additional datasets to be connected. The acquisition and replay specification is completed in Appendix D; the present evaluation uses one dataset family. The concentration-field interface is not restricted to the present PLIF resource. Any time-resolved scalar field supplied with spatial calibration, frame timing, source location, and wind metadata can define a new OSRL environment. This includes independent experimental datasets (e.g. Connor et al. (2018) or Stark et al. (2025)) as well as planar fields exported from CFD or gas-dispersion simulators such as GADEN (Monroy et al., 2017). These additions can reuse the same sensing, policy, task, and evaluation interfaces, while remaining explicitly versioned as separate benchmark tracks.

3.2 Interfaces as experimental controls

Table 1 connects the software decomposition to the experiments it enables. The plume interface specifies the external field; the sensor model specifies transduction; the observation processor defines

Table 1: **What a researcher can change in OSRL.** The interfaces expose separate experimental choices. The reported reference track fixes their settings.

Component	Configurable element	Question enabled
Plume	Measured or simulated scalar field and calibration	Does a policy transfer across recordings or plume-generation methods?
Sensor	Identity or stateful nonlinear response; placement	How does transduction alter available evidence?
Observation	Gas, wind, clearance, optional pose, source belief	Which representation supports control?
Task and control	Proximity/declaration; motion and stopping rules	What constitutes finding the source?
Reward and policy	Reward terms; planner or learned controller	How do objectives and strategies affect search?
Evaluation	Fixed manifest and episode artifacts	Can results and failure modes be compared?

the information supplied to a policy. Tasks, actions, and rewards specify the decision problem. These distinctions allow a researcher to change one component without implicitly redefining all the others.

The CLI exposes `convert`, `train`, `record`, and `eval` for data preparation, optimisation, qualitative inspection, and configurable evaluation. YAML configurations specify the components. The separate `benchmark-heuristic` and `benchmark-learned` commands execute reference protocols. Outputs include a machine-readable summary, episode table, trajectory CSVs, and trajectory images; learned runs also identify the checkpoint and observation processor. These artifacts connect each aggregate result to the policy and episodes that produced it.

A new controller can therefore be evaluated on the existing track while retaining its own internal state and action-selection method. Planner-specific action adapters connect native commands to the task's physical motion constraints. A new sensor or task can instead define a separate experiment through the same construction path. Scores from such altered settings must be identified as a different protocol, preserving the distinction between extending the toolbox and changing a benchmark.

3.3 A REUSABLE WORKFLOW AND A FIXED REFERENCE TRACK

OSRL supports three entry points. A researcher can *evaluate a controller* using the reference episodes and inspect its trajectories; *study a representation* by replacing the observation processor while keeping the environment fixed; or *define a sensing experiment* by changing the sensor frontend and recording the resulting protocol. The belief ablation in Section 5 demonstrates the second workflow, and the physical frontend in Section 6 demonstrates integration through the sensing interface.

Our release will pair the general toolbox with a versioned paper-reproduction package. It will provide a small runnable example, the data-access and conversion procedure, baseline configurations and checkpoints, evaluation records, and scripts for the paper's tables and figures. Saved-output analysis, checkpoint evaluation, and retraining will be documented separately, so using the benchmark need not begin with another training run. Appendix F specifies this package and its remaining access details.

4 REFERENCE TRACK AND LEARNED BASELINES

4.1 REFERENCE BENCHMARK

The reference benchmark manifest combines three held-out recordings (`r56-r58`), three domain widths (3, 6, and $10\,\mathrm{m}$), and 15 fixed normalised starts, giving 135 saved episodes per policy checkpoint. Two matched starts (episode indices 58 and 103) already lie inside the success region. All reference scores use the other 133 episodes, selected by $D_i^{\mathrm{goal}} > 0$ identically across methods; the original 135-episode run records remain available for audit. The source concentration is set to $20\,\mathrm{ppm}$; success is entry of the agent centre into a radius of $0.5\,\mathrm{m}$ around the source. The learned policies train on six disjoint recordings (`r59`, `r60`, `r61`, `r70`, `r71`, `r75`). This tests held-out recordings

within the same acquisition family. Starts and spatial transformations reuse those recordings and are not independent physical plume samples.

4.2 Policy evaluation via reliability and successful-path efficiency

Let $S_i \in \{0, 1\}$ indicate success, L_i be travelled path length, and

$$D_i^{\text{goal}} = \max\left(\|x_i^0 - x^{\text{source}}\| - r_{\text{succ}}, 0\right) \tag{3}$$

be the Euclidean distance from the start to the success region. It is a geometric reference length; in an obstructed task, a feasible shortest-path distance would be needed. We set $N = 133$ by excluding the two initially successful episodes ($D_i^{\text{goal}} = 0$) for every policy and seed. We report

$$\text{SR} = \frac{1}{N}\sum_i S_i, \qquad \text{PE}_{\text{succ}} = \frac{1}{N_{\text{succ}}}\sum_{i:S_i=1} \frac{D_i^{\text{goal}}}{\max(D_i^{\text{goal}}, L_i)}. \tag{4}$$

SR measures reliability; PE_{succ} describes path directness conditional on success. It is undefined when there are no successes. A policy can attain high conditional efficiency by solving only easy episodes, so the two metrics must be read together. They do not compare identical success subsets across policies, nor does path length measure search time or computation.

For compatibility we retain success weighted by inverse path length (SPL) (Anderson et al., 2018), using the same goal-region distance, in Appendix C. Within any run with at least one success,

$$\text{SPL} = \text{SR}\,\text{PE}_{\text{succ}}. \tag{5}$$

The primary pair exposes the reliability and conditional-directness factors of the joint score. Learned-policy metrics are computed within each seed and then averaged across seeds, with sample SD. The product of the two across-seed means need not equal mean SPL.

4.3 Implemented policies and their information access

As a benchmark example, we implement two heuristic navigation policies, namely *Infotaxis* (Vergassola et al., 2007) and *PMFS* (Hengenius et al., 2021; Ojeda et al., 2024), and two variants of a learned (PPO) policy, here called *belief RNN-PPO* and *no-belief RNN-PPO*. All of the policies share the same measured-world manifest and success geometry, with common physical speed constraints and planner-specific action adapters. The comparison is between configured navigation systems, with each planner's priors and action interface documented alongside the learned-policy inputs in Appendix D. The track uses identity concentration sensing to isolate navigation from gas-sensor dynamics. Dynamic MOx gas sensor models are used in the separate deployment study, see Section 6.

The learned policy receives 15 features: one gas measurement, two body-frame wind components, two local clearance features, and ten source-belief features. Absolute position is not a direct network input. However, the belief processor uses agent pose and domain geometry to register observations and express source hypotheses in the body frame. The complete belief-augmented system therefore uses localisation and geometry alongside local gas sensing. The full plume field is available for environment sampling and diagnostics only, and not available as an image observation.

4.4 Structured source estimation and the no-belief control

Belief RNN-PPO implements a belief processor that maintains weights b_t^i over candidate source cells s^i. A wind-conditioned observation model assigns an expected detection intensity $\mu_{i,t}$ from candidate-agent geometry. The update has the form

$$b_t^i \;\propto\; b_{t-1}^i\,\mathcal{L}(n_t; \mu_{i,t}), \qquad \mathcal{L}(n; \mu) \;\propto\; \exp(n \log \mu - \mu), \tag{6}$$

followed by normalisation and numerical flooring. Here n_t is scaled, count-like gas evidence. The Poisson-inspired weighting is an approximate evidence model; measured turbulent observations are temporally correlated and the weights are not asserted to be a calibrated posterior.

The ten-feature summary contains egocentric displacements to the modal and mean source candidates, peak mass, normalised entropy, and first- and second-order circular moments of directional mass. It

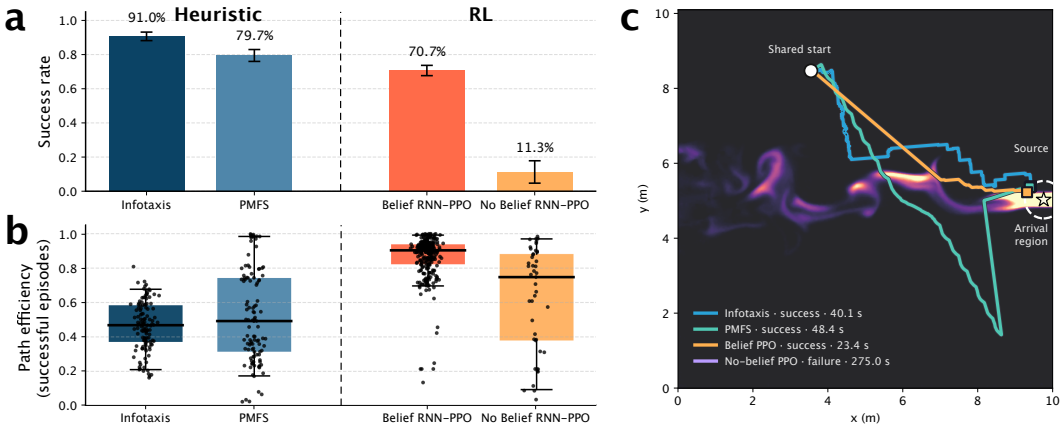

Figure 2: **Reliable search and direct routes are distinct strengths. (a)** Success rate across 133 matched starts outside the arrival region. Infotaxis and PMFS are each evaluated once (error bars computed via 68% Wilson score intervals over tested episodes); belief and no-belief RNN-PPO are shown across three training seeds. **(b)** Path efficiency on each policy's own successful episodes; dots show individual episodes, and boxes summarize their distributions. **(c)** Illustrative trajectories from the same start in episode 81 (recording r57). Infotaxis, PMFS, and belief RNN-PPO reach the source region; no-belief RNN-PPO does not. The measured plume snapshot provides spatial context and is not synchronized with the trajectories.

supplies a compact, interpretable state estimate to a recurrent controller trained with proximal policy optimisation (PPO) (Schulman et al., 2017). Each checkpoint is trained for 20×10^6 transitions using ten parallel environments; each learned condition has three independent training seeds. The detailed algorithmic implementation as well as optimisation settings are reported in Appendix A and Appendix B respectively.

No-belief RNN-PPO uses the same recurrent architecture and 15-dimensional input width, with all ten belief coordinates fixed to zero throughout training and evaluation. Gas, wind, clearance, reward, training recordings, and training budget are retained. Retraining avoids an inference-time distribution shift caused by masking a trained policy. The intervention removes the entire engineered estimate, including accumulated evidence and its pose- and geometry-dependent encoding. The intervention tests the contribution of that complete representation under a fixed training procedure; it does not isolate temporal memory from its geometric and transport priors.

5 What the Reference Benchmark Reveals

5.1 The benchmark separates reliability and path directness

Table 2 shows that the configured methods occupy different positions in this comparison. Infotaxis succeeds in 121/133 eligible episodes (91.0%), followed by PMFS with 106/133 (79.7%) and Belief RNN-PPO with $(70.7 \pm 3.0)\%$. On their respective successful episodes, Belief RNN-PPO has higher path efficiency: 0.864 ± 0.004, compared with 0.464 for Infotaxis and 0.517 for PMFS. Its mean goal-region SPL is also higher (0.610 versus 0.422 and 0.412; Appendix C). These results expose a useful benchmark target: improving reliability while retaining direct successful paths. They also show why a common evaluation should preserve both outcomes, rather than select a single policy ranking for every application.

5.2 A concrete challenge for learned state representations

Removing the source-belief features reduces mean success by 59.4 percentage points, from $(70.7 \pm 3.0)\%$ to $(11.3 \pm 6.6)\%$. Conditional path efficiency also decreases, from 0.864 ± 0.004 to 0.647 ± 0.081, on the respective successful subsets. Under the evaluated training budget, recurrent PPO

Table 2: **Reference benchmark.** Each run saves the same 135 episodes; scores use the same 133 nontrivial starts. RNN-PPO entries are mean ± sample SD over three training seeds, with efficiency calculated within each seed's successful subset. Planner entries describe one evaluation of each configured policy. SD describes training variation on the fixed track. Successful subsets differ across policies.

Policy	Success rate ↑	Path efficiency | success ↑
Infotaxis	91.0%	0.464
PMFS	79.7%	0.517
Belief RNN-PPO	$(70.7 \pm 3.0)\%$	0.864 ± 0.004
No-belief RNN-PPO	$(11.3 \pm 6.6)\%$	0.647 ± 0.081

operating on gas, wind, and clearance alone does not recover the performance of the controller supplied with structured source estimates.

This makes the observation interface a substantive research opportunity. The explicit estimator combines encounter history, wind-conditioned evidence, localisation, and a source prior into ten features. The result establishes the utility of this combined representation; it leaves open which ingredients a learned estimator can recover and under what information budget. OSRL provides a fixed environment in which such alternatives can be implemented and compared.

Where does useful memory reside? With belief features retained, resetting the policy's recurrent state at every step achieves $(72.4 \pm 2.8)\%$ success and 0.878 ± 0.016 successful-path efficiency, compared with $(70.7 \pm 3.0)\%$ and 0.864 ± 0.004 for Belief RNN-PPO. Thus the large loss from removing the belief contrasts with similar descriptive performance when only policy-state persistence is removed. This is consistent with useful history being carried by the explicit estimator. The control retains the recurrent architecture and the belief's own memory; it tests policy-state persistence, not a memory-free agent. Three seeds do not establish equivalence (Appendix B).

5.3 Inspecting behaviour on a shared episode

The common manifest also supports trajectory-level comparison. Figure 2c is designed to display one matched episode for all four policies, with the same recording, start, source, and replay initialisation. A common time-averaged plume background provides spatial context; it is not the instantaneous field encountered along the entire path. This diagnostic view connects aggregate scores to sampling behaviour and illustrates the inspectable outputs available to a new benchmark user.

6 From Measured-Plume Learning to Robotic Search

To demonstrate the toolbox beyond replay evaluation, we connect a learned controller to a mobile robot searching for an acetone source with a real electronic nose. This exercises the complete path from sensor measurements through state estimation and policy inference to physical movement. A separate deployment checkpoint is trained with a MOx-like model combining asymmetric rise and recovery filtering with nonlinear transduction (Martinez et al., 2019). The sensor interface therefore supports developing a controller under response dynamics relevant to the physical frontend, alongside the identity-sensor benchmark.

The holonomic robot (*Hiwonder JetAuto*[1], Figure 3a) carries a high-speed MOx electronic nose (Dennler et al. (2024), Figure 3b) and planar LiDAR. Gas measurements acquired near $1\,\mathrm{kHz}$ are filtered and decimated to $20\,\mathrm{Hz}$, then converted to a baseline-relative proxy. Localisation supports belief geometry, logging, and arena handling; wind direction is prescribed. Independent arena and LiDAR safety mediation can modify policy commands before execution. Appendix E specifies the signal and control path.

In a nominal $4.0\,\mathrm{m} \times 2.0\,\mathrm{m}$ arena (Figure 3c), the robot reaches a $0.30\,\mathrm{m}$ source region in 10/10, 9/10, and 10/10 attempts at nominal flow conditions of 2, 3, and $6\,\mathrm{m\,s^{-1}}$, respectively (Figure 3d &

[1] https://www.hiwonder.com/products/jetauto

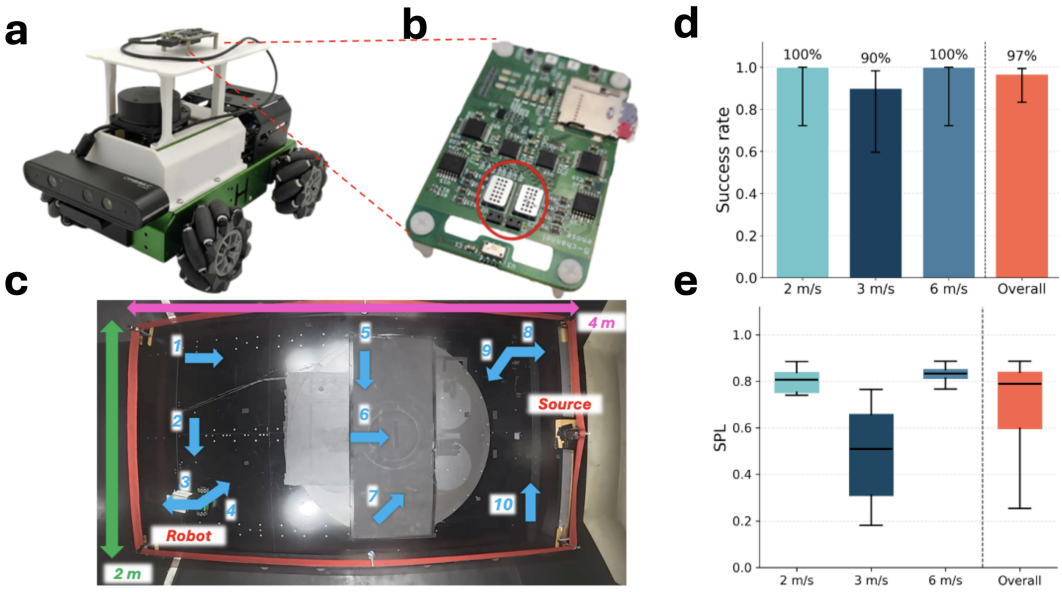

Figure 3: **Physical deployment of the learned olfactory-search policy. a,** Holonomic mobile robot used for the wind-tunnel experiments. **b,** High-speed MOx electronic nose (Dennler et al., 2024) mounted on the robot, with the sensing elements highlighted. **c,** Overhead view of the $4 \times 2\,\mathrm{m}$ search arena showing the ten prescribed starting poses and headings and the odour-source location. **d,** Source-localisation success rates for nominal wind speeds of 2, 3, and $6\,\mathrm{m\,s^{-1}}$, and pooled across all 30 trials; error bars denote 95% Wilson confidence intervals. **e,** Corresponding distribution of SPL values for each wind condition and overall. The robot reached the source region in 29 of 30 trials.

e). The 29/30 arrivals demonstrate repeated source-region reaching by the integrated robot. For a toolbox intended to support embodied learning, this is an important practical result: the development workflow connects to real chemical measurements, localisation, timing, safety, and actuation through the sensing-policy interface. The separate checkpoint and deployment conditions are documented in Appendix E.

7 Discussion and Scope

A shared problem for embodied learning. OSRL makes measured turbulence available at the level where learning researchers work: observations, actions, representations, policies, and evaluation records. The reference results identify two concrete objectives for subsequent methods: retaining direct paths while improving reliability, and learning state estimates that approach the utility of the engineered belief representation. The same interfaces support experiments on active information gathering and sensor dynamics. These questions can be pursued on a common physical record with explicit changes to the experimental configuration.

Accumulating comparable evidence. The toolbox and benchmark serve complementary roles. Flexible components support new questions; a fixed reference track preserves a meaningful comparison point. Our release will document both a route for adding a policy to the reference track and the settings that define a new track. This makes the contribution useful beyond the particular baselines evaluated here, while keeping changes in sensing, information access, and task assumptions visible. The same field contract also provides a route to future tracks built from independent measurements or simulated plume fields.

Limits of the current evidence. The reference track uses three held-out recordings from one acquisition family. Additional starts and spatial transformations increase the set of decision problems but do not create independent turbulent realisations; rescaling also does not establish fluid-dynamic similarity to a larger arena. Planar measurements, finite measurement bandwidth, identity sensing,

and the passive-probe assumption bound the present benchmark. Its baselines provide an initial comparison rather than exhaustive algorithm coverage. The belief ablation tests a combined state-estimation package, as specified in Section 4.4. The physical study uses a separate checkpoint, repeated trials in one installation, prescribed wind, evolving background, and mediated actions. Its arrival counts establish integration; causal use of live odour and transfer across new installations would require additional controls and environments.

8 Conclusion

OSRL turns recorded turbulent concentration fields into a reusable setting for learning to navigate by smell. Modular sensing and representation interfaces make experimental choices explicit, while a fixed reference track connects policy comparisons to inspectable episodes. The initial results reveal both a reliability–directness trade-off and a substantial gap associated with structured source estimation. The robotic demonstration extends this workflow to real chemical sensing and movement, with 29 source-region arrivals in 30 trials. Together with the toolbox and reproduction release, these results provide a practical foundation for the learning community to develop, test, and compare olfactory-search methods.

Reproducibility Statement

We release the toolbox and the paper-reproduction package, including conversion utilities, data-access instructions, configurations, episode manifests, trained checkpoints with preprocessing state, and episode-level outputs. The reproduction workflow distinguishes regenerating analyses from saved outputs, evaluating released checkpoints, and retraining baselines (Appendix F). Methods and implementation details are specified in Sections 3–4.2 and Appendices A–E. The complete toolbox and reproduction package, including configurations, evaluation scripts, episode manifests, trained policy checkpoints, and instructions for reproducing the reported results and figures, are available at `https://anonymous.4open.science/r/OSRL/`. All materials are provided anonymously for review.

Ethics Statement

This work involves no human or animal subjects. Physical trials use controlled acetone release and a mobile robot in a laboratory installation. Chemical-source localisation supports monitoring and safety applications; use outside the evaluated setting requires application-specific validation.

AI Use Statement

Generative AI tools assisted with code development and debugging, methodological feedback, literature-search support, and manuscript drafting and editing. They were not used to generate experimental measurements or conduct physical experiments. The authors take responsibility for the final text, claims, analyses, citations, and software, including content produced with AI assistance.

References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. URL `https://arxiv.org/abs/1807.06757`.

Antonio Celani, Emmanuel Villermaux, and Massimo Vergassola. Odor landscapes in turbulent environments. *Physical Review X*, 4(4):041015, 2014.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.48550/arXiv.2306.13831.

Erin G. Connor, Margaret K. McHugh, and John P. Crimaldi. Quantification of airborne odor plumes using planar laser-induced fluorescence. *Experiments in Fluids*, 59(9):137, September 2018. doi: 10.1007/s00348-018-2591-3.

John P. Crimaldi. Planar laser induced fluorescence in aqueous flows. *Experiments in Fluids*, 44(6): 851–863, June 2008. doi: 10.1007/s00348-008-0496-2.

John P. Crimaldi, Hong Lei, Andreas Schaefer, Michael Schmuker, Brian H. Smith, Aaron C. True, Justus V. Verhagen, and Jonathan D. Victor. Active sensing in a dynamic olfactory world. *Journal of Computational Neuroscience*, 50(1):1–6, February 2022. doi: 10.1007/s10827-021-00798-1.

Nik Dennler, Damien Drix, Tom P. A. Warner, Shavika Rastogi, Cecilia Della Casa, Tobias Ackels, Andreas T. Schaefer, André Van Schaik, and Michael Schmuker. High-speed odor sensing using miniaturized electronic nose. *Science Advances*, 10(45):eadp1764, November 2024. doi: 10.1126/sciadv.adp1764.

Nik Dennler, Elle Stark, Saimon Collaku, Lars Larson, André van Schaik, Michael Schmuker, John P. Crimaldi, Andreas T. Güntner, and Aaron True. Odour sensing in turbulent plumes with high-speed electronic nose and non-invasive ground truth. *arXiv preprint*, 2026a. doi: 10.48550/arXiv.2604.19626.

Nik Dennler, Elle Stark, Andreas Güntner, and Aaron True. Dataset: Odour sensing in turbulent plumes with high-speed electronic nose and non-invasive ground truth. ETH Zurich Research Collection, January 2026b. URL http://hdl.handle.net/20.500.11850/782611.

Kordel K France, Ovidiu Daescu, Latifur Khan, and Rohith Peddi. Chasing ghosts: A simulation-to-real olfactory navigation stack with optional vision augmentation. *Sensors*, 26(9):2849, 2026.

Adam Francis, Shuai Li, Christian Griffiths, and Johann Sienz. Gas source localization and mapping with mobile robots: A review. *Journal of Field Robotics*, 39(8):1341–1373, 2022. doi: 10.1002/rob.22109.

James B. Hengenius, Erin G. Connor, John P. Crimaldi, Nathaniel N. Urban, and G. Bard Ermentrout. Olfactory navigation in the real world: Simple local search strategies for turbulent environments. *Journal of Theoretical Biology*, 516:110607, 2021. ISSN 0022-5193. doi: https://doi.org/10.1016/j.jtbi.2021.110607. URL https://www.sciencedirect.com/science/article/pii/S0022519321000291.

H. Ishida, Y. Wada, and H. Matsukura. Chemical Sensing in Robotic Applications: A Review. *IEEE Sensors Journal*, 12(11):3163–3173, November 2012. doi: 10.1109/JSEN.2012.2208740.

Aurore Loisy and Christophe Eloy. OTTO: A Python package to simulate, solve and visualize the source-tracking POMDP. *Journal of Open Source Software*, 7(74):4266, 2022. doi: 10.21105/joss.04266.

Aurore Loisy and Robin A Heinonen. Deep reinforcement learning for the olfactory search POMDP: a quantitative benchmark. *The European Physical Journal E*, 46(3):17, 2023. doi: 10.1140/epje/s10189-023-00277-8.

Dominique Martinez, Javier Burgués, and Santiago Marco. Fast Measurements with MOX Sensors: A Least-Squares Approach to Blind Deconvolution. *Sensors*, 19(18):4029, September 2019. doi: 10.3390/s19184029.

Javier Monroy, Victor Hernandez-Bennetts, Han Fan, Achim Lilienthal, and Javier Gonzalez-Jimenez. GADEN: A 3D Gas Dispersion Simulator for Mobile Robot Olfaction in Realistic Environments. *Sensors*, 17(7):1479, June 2017. doi: 10.3390/s17071479.

Pepe Ojeda, Javier Monroy, and Javier Gonzalez-Jimenez. Robotic gas source localization with probabilistic mapping and online dispersion simulation. *IEEE Transactions on Robotics*, 40: 3551–3564, 2024. doi: 10.1109/TRO.2024.3426368.

Pepe Ojeda, Javier Monroy, and Javier Gonzalez-Jimenez. Gaden-rt: A real time and interactive gas dispersion simulator for mobile robotics. *SoftwareX*, 32:102388, 2025.

Michael Schmuker, Viktor Bahr, and Ramón Huerta. Exploiting plume structure to decode gas source distance using metal-oxide gas sensors. *Sensors and Actuators B: Chemical*, 235:636–646, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Satpreet H. Singh, Floris Van Breugel, Rajesh P. N. Rao, and Bingni W. Brunton. Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nature Machine Intelligence*, 5(1): 58–70, January 2023. doi: 10.1038/s42256-022-00599-w.

Elle Stark, Lars Larson, Aaron C True, and John P Crimaldi. Characterizing airborne odor landscapes using simultaneous stereo PIV and PLIF. In *Division of Fluid Dynamics Annual Meeting 2025*. APS, 2025.

Massimo Vergassola, Emmanuel Villermaux, and Boris I Shraiman. 'Infotaxis' as a strategy for searching without gradients. *Nature*, 445(7126):406–409, 2007. doi: 10.1038/nature05464.

Sibo Wang-Chen, Victor Alfred Stimpfling, Thomas Ka Chung Lam, Pembe Gizem Özdil, Louise Genoud, Femke Hurtak, and Pavan Ramdya. NeuroMechFly v2: simulating embodied sensorimotor control in adult Drosophila. *Nature Methods*, 21:2353–2362, 2024. doi: 10.1038/s41592-024-02497-y.

A SOURCE-BELIEF REPRESENTATION

The processor discretises the longest domain side into 32 cells and adds eight padded cells beyond each physical boundary. At reset, 99% of prior mass is assigned within the domain and 1% to the padded region. This avoids treating the rectangular boundary as an absolute source constraint while retaining an explicit domain prior. The identity-sensor configuration uses likelihood scale $\ell = 0.08 L_{\mathrm{dom}}$, where $L_{\mathrm{dom}} = \max(W, H)$, wind-bias coefficient $\beta_w = 0.5$, gas evidence scale 1.0, minimum source intensity 1.0, and a probability floor of 10^{-6}. These settings define an approximate estimator; they do not calibrate its weights as probabilities of the measured transport process.

Let q_t denote agent position and $R(-\vartheta_t)$ rotation into the body frame. For $i_t^{\max} = \arg\max_i b_t^i$ and $\bar{s}_t = \sum_i b_t^i s^i$, the four location features are

$$d_t^{\max} = L_{\mathrm{dom}}^{-1} R(-\vartheta_t)(s^{i_t^{\max}} - q_t), \tag{7}$$

$$d_t^{\mathrm{mean}} = L_{\mathrm{dom}}^{-1} R(-\vartheta_t)(\bar{s}_t - q_t). \tag{8}$$

Two additional features encode the peak probability within the physical domain and clipped normalised entropy. Candidate cells are also grouped into eight body-frame sectors with directional masses $m_{k,t}$ and angles $\alpha_k = k\pi/4$. Their first- and second-order circular moments are

$$u_{j,t} = \left[\sum_k m_{k,t} \cos(j\alpha_k),\ \sum_k m_{k,t} \sin(j\alpha_k) \right], \qquad j \in \{1, 2\}. \tag{9}$$

The complete feature vector is

$$\phi_t = [d_t^{\max}, d_t^{\mathrm{mean}}, b_t^{\max}, \bar{S}_t, u_{1,t}, u_{2,t}] \in \mathbb{R}^{10}. \tag{10}$$

The second circular moment can retain axial structure when opposing directional masses cancel in the first. The remaining policy inputs comprise gas, two body-frame wind components, and two clearance values clipped at $0.5\,\mathrm{m}$.

Exact implementation. The processor clips negative centre-sensor readings to zero and uses $n_t = \mathrm{round}(\max(g_t, 0)/1)$ as count-like evidence (the identity sensor reads the transformed nominal ppm field). For candidate i at distance $d_{i,t}$, let $a = \frac{1}{2}\min(W/32, H/32)$, $\ell = 0.08 \max(W, H)$, and $I = \max(1, c_{\mathrm{peak}})$. The implemented wind-conditioned intensity is

$$\mu_{i,t} = \max\left[10^{-12}, \frac{I\, K_0(\max(d_{i,t}, a)/\ell)}{\log(\max(\ell/a, 1 + 10^{-6}))} \exp(0.5 \cos \psi_{i,t}) \right], \tag{11}$$

where $\cos\psi$ projects the candidate direction onto the upwind unit vector, and K_0 is the modified Bessel function. The code evaluates $n_t \log \mu_{i,t} - \mu_{i,t} - \log \Gamma(n_t + 1)$, subtracts the maximum log likelihood, exponentiates, floors the result at 10^{-12}, multiplies the prior cell weights, normalises, then smooths each cell as $(b_i + 10^{-6})/(1 + M10^{-6})$ for M padded-grid cells. The mode breaks ties in grid order within the physical domain; mean, peak probability, and eight sector masses renormalise physical-domain mass, while entropy uses the full padded-grid belief divided by $\log M_{\mathrm{inside}}$ and clipped to $[0, 1]$. The update precedes feature extraction; episode reset clears accumulated evidence.

B TRAINING CONFIGURATION AND SECONDARY CONTROL

Each learned checkpoint uses 20×10^6 environment transitions and ten parallel environments. The conditions share training recordings, task distribution, reward composition, action space, and PPO settings. Table 3 records the optimisation settings supplied for the study.

Architecture and optimisation. The JAX/Flax actor applies two tanh dense layers of width 64, a 96-unit GRU, and a nine-logit discrete-action head. A separate feedforward value path applies two tanh dense layers of width 64 and a scalar output; the critic therefore receives no recurrent state, despite the retained legacy configuration key `enable_critic_rnn`. RNN-PPO uses Adam, a clipping range of 0.2, a maximum gradient norm of 0.5, and the schedule in Table 3. The recurrence-reset control evaluates the same trained belief-policy checkpoints while replacing the incoming and outgoing GRU carry with zero at every step; in the standard condition, the carry and belief state are

Table 3: PPO training configuration. Training is repeated for three seeds per reported condition.

Setting	Value
Transitions / checkpoint	20×10^6
Parallel environments	10
Rollout steps / environment	680
Batch size; PPO epochs	512; 4
Discount; GAE parameter	0.995; 0.98
Entropy coefficient; value-loss coefficient	0.008; 0.5
Target KL; gradient-norm limit	0.012; 0.5
Learning-rate schedule	Linear, 1.5×10^{-4} to 2×10^{-5}
Recurrent units	96
Policy and value heads	Separate $[64, 64]$ heads

reset only at episode boundaries. The implementation performs per-update advantage normalisation but applies no additional observation, return, or reward normalisation. Checkpoints contain the policy parameters and model metadata required for deterministic inference.

The released package supports Python 3.11 and 3.12 and declares the complete dependency set in `pyproject.toml`. We additionally provide the versions of the core numerical stack used to validate the cleaned release. The exact package versions of the original training environment were not retained; the released configurations, checkpoint hashes, and deterministic evaluation protocol therefore define the archived trained policies more precisely than an inferred reconstruction of that environment.

Task distribution and reward. The supplied training configurations draw from the six named recordings with domain widths in $[4, 5]\,\mathrm{m}$, nominal source concentration $20\,\mathrm{ppm}$, and random initial positions and orientations. The agent has radius $0.162\,\mathrm{m}$ and eight translation directions at $0.30\,\mathrm{m\,s^{-1}}$ plus stop. The shaped reward includes source-distance progress (on-plume/off-plume scales 6/3, clipped to ± 0.15, with an eight-step blank grace), a -0.008 step cost, a near-wall penalty reaching -0.1, a -2 collision penalty, $+200$ at source-region success, and -60 on dataset exhaustion. The progress reward uses the true source coordinates *during training*; those coordinates are not a direct inference-time PPO input. All reported training conditions use a control interval of $\Delta t = 0.05\,\mathrm{s}$, matching the native $20\,\mathrm{Hz}$ PLIF cadence. The field is sampled at the post-action agent position using bilinear spatial interpolation and a piecewise-constant temporal frame, with frame index $\lfloor t/\Delta t \rfloor$. Each converted recording contains 5,500 retained frames, giving a maximum episode duration of $275\,\mathrm{s}$ or 5,500 control steps. In the proximity task, an episode terminates successfully when the agent centre enters the $0.5\,\mathrm{m}$ source region and is truncated when the recording is exhausted. Boundary crossings are clamped to the valid domain and counted as collisions but do not terminate the episode; there is no additional step limit or failure termination. Declaration actions are disabled for this task. Belief and no-belief conditions use the same timing, task, action, reward, and truncation rules.

Selection and compute. Each learned condition has training seeds 42, 43, and 44; its reported run evaluates the matching saved checkpoint. The tenth recording is outside the reported split. Each 20-million-step training run required approximately three hours on a MacBook Pro with an Apple M1 Pro processor and 32 GB of unified memory.

The primary no-belief control replaces all ten belief coordinates with zeros during training and inference. Observation width and the remaining local features are preserved. It is a separately trained condition, not a masked deployment of the full checkpoint.

For the secondary recurrence control, the belief features are retained while recurrent state is reset before every policy step. The network still has recurrent parameters but cannot propagate policy hidden state across steps; the belief processor continues to accumulate evidence. This condition therefore is not a memory-free system. Across three training seeds it achieves $(72.4 \pm 2.8)\%$ success, 0.878 ± 0.016 successful-path efficiency, and 0.636 ± 0.015 goal-region SPL. These descriptive results are close to those of Belief RNN-PPO; three seeds do not establish statistical equivalence. No result is claimed for a joint no-belief/no-recurrence condition.

C GOAL-REGION SPL AND AGGREGATION

The joint metric follows the SPL construction (Anderson et al., 2018) with distance measured to the success region:

$$\mathrm{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{D_i^{\mathrm{goal}}}{\max(D_i^{\mathrm{goal}}, L_i)}. \tag{12}$$

Failures contribute zero. This is distinct from using start-to-source-centre distance when the success rule permits stopping at a finite radius. The same $D_i^{\mathrm{goal}} > 0$ filter is used as for the primary metrics. The two excluded rows were recorded as automatic successes with stored SPL zero; assigning them conditional efficiency one, as a plotting utility did, produces different numbers. Our saved-output analysis checks all 135 source rows and applies the documented 133-row rule consistently.

Table 4: Joint success and path efficiency on the reference track. Learned-policy values are mean ± sample SD over training seeds.

Policy	Goal-region SPL ↑
Infotaxis	0.422
PMFS	0.412
Belief RNN-PPO	0.610 ± 0.029
No-belief RNN-PPO	0.070 ± 0.038
Belief RNN-PPO, recurrent state reset	0.636 ± 0.015

For each seed j, compute SR_j, $\mathrm{PE}_{\mathrm{succ},j}$, and SPL_j first, then report their separate means and sample standard deviations. Multiplying the reported SR and efficiency means does not in general recover mean SPL because the two seed-level metrics may covary. Pooling successful episodes also differs from equal-seed aggregation: it gives more weight to seeds with more successes. A condition with no successes has zero SPL and undefined conditional efficiency; it should not be assigned zero conditional efficiency or silently excluded from seed summaries. The stored per-episode outcomes are the appropriate source for exact successful-episode counts and unrounded summaries.

D REFERENCE PROTOCOL AND REPRODUCTION ARTIFACTS

Table 5: Reference settings established in the manuscript. Additional implementation-specific settings must be read from the exact configuration and manifest associated with a run.

Component	Setting
Training recordings	`r59, r60, r61, r70, r71, r75`
Evaluation recordings	`r56, r57, r58`
Evaluation domain widths	3, 6, 10 m
Initial positions	15 fixed normalised starts
Episodes / checkpoint	135 saved; 133 scored ($D_i^{\mathrm{goal}} > 0$)
Nominal source concentration	20 ppm
Success region	Agent centre within 0.5 m of source
Learned gas input	Identity sensor at agent centre
Other learned inputs	Body-frame wind, local clearance, optional belief summary
Learned actions	Eight translation directions and stop, without rotation
Learned inference	Deterministic
Primary metrics	Success rate and successful-path efficiency
Joint metric	Goal-region SPL

Data and execution contract. The source recordings contain 5,500 processed acetone-PLIF frames acquired at 20 Hz after removal of the first 500 frames (25 s) used for source stabilisation (Dennler et al., 2026a). During dataset preparation, the two alternating laser-head sequences are corrected separately by background subtraction, flat-field division, and normalisation to the measured source intensity. The resulting frames are geometrically unwarped, median-filtered, merged in temporal order,

and stored as source-normalised `float32` arrays with axis order $[\mathrm{time}, y, x]$. The accompanying HDF5 metadata provide the physical coordinate grids, pixel-to-millimetre calibration, and source reference position.

OSRL streams the processed image tensor into an adjacent NumPy array without temporal resampling. Conversion additionally records an estimated source coordinate, wind direction, and a two-dimensional plume mask under `/plif-rl`. The source is estimated from the centre of the highest-energy connected region above the 99.9th percentile of the log time-averaged field, subject to a minimum area of 40 pixels. The plume mask retains the highest-energy connected component above the 80th percentile after morphological closing and hole filling, subject to a minimum area of 5,000 pixels. This mask is used for plume-membership features and reward logic; concentration observations are sampled from the complete scalar field.

At runtime, stored pixel coordinates $[y, x]$ are mapped to world coordinates (x, y) in metres. Selecting a domain width applies one uniform metres-per-pixel scale to both image axes and therefore preserves the recorded aspect ratio. Source-normalised concentrations are clipped below at zero and scaled to the configured source concentration, $20\,\mathrm{ppm}$ in the reference benchmark. Sampling is bilinear in space and piecewise constant in time at $\Delta t = 0.05\,\mathrm{s}$; samples outside the image are zero padded. Episodes begin at the first retained frame and are truncated when simulated time reaches the end of the 5,500-frame recording.

Information available to each system. The inspected inference paths have these inputs and action modes:

System	Inference-time inputs and prior	Control
Infotaxis	Local gas and wind; pose, domain geometry, peak concentration and internal source prior; binary detections at 0.5% of peak	Discrete movement
PMFS	Local gas and wind; pose, domain geometry, peak concentration and model-based mapping/dispersion prior; 2% detection threshold	Continuous body-frame velocity
Belief RNN-PPO	Gas, body-frame wind, clearance, ten belief features; processor additionally uses pose, geometry, peak and wind	Nine discrete actions
No-belief RNN-PPO	Gas, body-frame wind, clearance; ten zero-filled feature slots	Nine discrete actions

The evaluated policies do not receive the true source coordinates or the complete concentration field as inference-time inputs. A new heuristic planner is instantiated for every episode using the domain extent, wind direction, peak concentration, success radius, body radius, control interval, and resolved action layout; its internal state is therefore reset between episodes. At each step, the heuristic interface supplies the current pose and concentration-sensor measurements but not the source location. Infotaxis converts the concentration measurement into a binary hit using a threshold of 0.5% of the episode peak, updates a 32-cell-long-side source posterior, and selects among stop and four cardinal translation commands by expected entropy reduction. PMFS uses a threshold of 2% of the episode peak, maintains 25×25 hit and source-probability grids, and returns a body-frame continuous velocity directed toward its selected target. The environment scales this command to the common maximum translational speed of $0.30\,\mathrm{m\,s^{-1}}$. RNN-PPO is evaluated deterministically by selecting the highest-probability action among stop and eight translation directions. All four conditions use proximity-based stopping: success is registered automatically upon entering the $0.5\,\mathrm{m}$ arrival region, and no policy is given a source-declaration action. Differences in planner state, action representation, and low-level adapters are part of the configured systems being compared.

Numerical provenance. All paths in Figure 2c were regenerated with the cleaned release from the corresponding checkpoint or heuristic implementation. The fixed protocols produce 135 matched episodes per run; indices 58 and 103 begin inside the $0.5\mathrm{m}$ arrival region and are excluded before analysis, leaving the same 133 eligible starts for every condition. Infotaxis and PMFS succeed in 121 and 106 episodes, respectively. Belief RNN-PPO succeeds in 98, 90, and 94 episodes for seeds 42, 43, and 44, whereas no-belief RNN-PPO succeeds in 21, 5, and 19 episodes. Reported RNN-PPO values

are equal-seed means with sample standard deviations; conditional path efficiency is first averaged over the successful episodes within each seed. The analysis script independently recomputes success rate, conditional path efficiency, and SPL from the episode-level records.

The reproduction package will link each result to its configuration, protocol version, manifest, and policy/checkpoint identity. The episode table and trajectories support recomputation of both primary metrics and goal-region SPL. Configurable `eval` runs and altered sensor/task settings must be distinguished from scores on the fixed reference track. Physical deployment uses a separate checkpoint and success radius and is not numerically interchangeable with replay evaluation.

E Physical Deployment Details

Deployment-oriented training uses a forward sensor model with asymmetric filtering and nonlinear transduction, similar to Martinez et al. (2019), whereas parameters were fitted using known gas stimuli (not shown here). For concentration $c(k)$ and filtered state $\widetilde{c}(k)$,

$$\widetilde{c}(k) = \begin{cases} \alpha_r c(k) + (1-\alpha_r)\widetilde{c}(k-1), & \dot{c}(k) > 0, \\ \alpha_f c(k) + (1-\alpha_f)\widetilde{c}(k-1), & \text{otherwise}, \end{cases} \qquad g(k) = \beta \widetilde{c}(k)^{\gamma}. \tag{13}$$

This models response dynamics and nonlinear signal generation; it is not an inverse concentration estimator. The exact coefficients belong to the deployment configuration.

The physical frontend acquires the e-nose channels at approximately $1\,\mathrm{kHz}$. A 401-tap FIR low-pass filter with $20\,\mathrm{Hz}$ cutoff precedes decimation by 50. One resistance channel is converted to a bounded baseline-relative proxy, not calibrated concentration. At each $20\,\mathrm{Hz}$ control tick, gas, localisation-derived belief geometry, clearance, and prescribed wind are combined for the policy. A predictive arena mediator may substitute unsafe actions, while a direction-dependent LiDAR interlock can command zero velocity. Requested and executed commands are logged separately.

A pump-activated washing bottle containing acetone provides the source in the nominal $4.0\,\mathrm{m} \times 2.0\,\mathrm{m}$ robot-accessible arena. Entry within $0.30\,\mathrm{m}$ of the registered outlet defines arrival. Ten attempts are performed at each nominal flow of 2, 3, and $6\,\mathrm{m\,s^{-1}}$, yielding 10, 9, and 10 arrivals. Closed-loop tunnel operation leads to background accumulation across acquisition order. The deployment reproduction package will include the trial record and available logs distinguishing requested from executed commands.

F Toolbox Release and Reproduction Package

We release the general-purpose toolbox and a versioned reproduction package tied to this paper. The package distinguishes three operations with different resource requirements:

1. **Regenerate the paper analyses.** Read the stored episode outputs, apply the documented goal-region metric and seed aggregation, and regenerate the tables and plots. This route requires no policy training or simulation rerun.
2. **Evaluate released checkpoints.** Load the measured fields, fixed manifests, trained policies, and associated preprocessing state; execute the reference protocols; compare episode and aggregate outputs with the archived results.
3. **Retrain the learned baselines.** Use the released training configurations, seeds, dependencies, and checkpoint-selection rule; evaluate the resulting policies with the same reference protocol.

The toolbox release includes installation instructions, an environment specification, a minimal measured-data example, and examples of adding a controller or observation processor. Data documentation specifies recording identifiers, calibration, units, coordinate systems, splits, conversion steps, and checksums. Each reported result is associated with a code version, configuration, manifest, and policy/checkpoint identity. Figure-generation scripts identify the plotted runs and the matched trajectory episode.