Learn to Trace Odors: Robotic Odor Source Localization via Deep Learning Methods with Real-world Experiments

Lingxiao Wang
Dept. of Electrical Engineering
Louisiana Tech University
Ruston LA, 71272
lwang@latech.edu

Ziyu Yin
Dept. of Electrical Engineering
and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach FL, 32114
yinz@my.erau.edu

Shuo Pang
Dept. of Electrical Engineering
and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach FL, 32114
shuo.pang@erau.edu

SoutheastCon 2023 | 978-1-6654-7611-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/SoutheastCon51012.2023.10115175

***Abstract*—This paper presents an olfactory-based navigation algorithm via deep learning (DL) methods. The object is to obtain a neural network that navigates a mobile robot to find an odor source without explicating search strategies. Two deep neural networks (DNNs), including feedforward and long short-term memory neural networks (FNN and LSTM), are devised to calculate robot heading commands based on onboard sensor readings. The training dataset is obtained by implementing traditional olfactory-based navigation algorithms, namely moth-inspired and Bayesian-inference methods, on a mobile robot in hundreds of odor source localization (OSL) tests. After the supervised learning, DNNs are validated in real-world experiments with unseen odor source locations and airflow fields. Experiment results show that both FNN and LSTM can imitate the moth-inspired method but cannot effectively learn the complex Bayesian-inference method. In terms of the averaged search time in repeated tests, the proposed FNN and LSTM outperform the Bayesian-inference method by 18% and 14%, respectively, and both networks achieve a comparable search performance with the moth-inspired method.**

I. INTRODUCTION

Olfaction, also known as the sense of smell, is an important sensing ability that is widely used by animals in life-essential activities, such as homing, foraging, mate-seeking, evading predators, etc. Inspired by odor search behaviors of animals, mobile robots, equipped with odor-detection sensors, could locate an odor source in an unknown environment. The technology of employing robots to find odor sources is referred to as robotic odor source localization (OSL) [1]. Some robotic OSL applications include monitoring air pollution [2], locating chemical gas leaks [3], finding unexploded mines and bombs [4], and marine surveys such as locating underwater hydrothermal vents [5].

Robotic OSL has been viewed as a challenging task due to the turbulent nature of airflows and the resulting odor plume characteristics [6]. The key to correctly finding an odor source is the design of effective olfactory-based navigation algorithms. Like image-based navigation algorithms, which extract the information from images as the reference to navigate a robot, olfactory-based navigation algorithms detect odor plumes as cues to guide a robot moving toward an odor source. Existing olfactory-based navigation methods can be categorized into two types, namely bio-inspired and engineering-based (i.e., probabilistic) methods [7]. A bio-inspired method directs a plume-tracing robot to find the odor source by mimicking animals' odor search behaviors. A typical example is the mate-seeking behaviors of male moths, which could find female moths over a long distance via tracing emitted pheromones [8]. To complete this task, a male moth adopts a 'surge/casting' behavior pattern: a male moth flies upwind when it detects pheromones and traverses the wind direction when pheromones are absent. By contrast, engineering-based methods utilize math and physics approaches to deduce odor plume distributions and predict possible odor source locations. Then, a path planning algorithm is employed to construct a collision-free trajectory, navigating the robot to move to the estimated target.

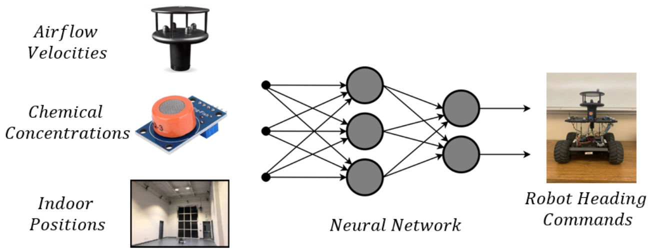


Fig. 1. The general idea of the deep learning-based navigation method. A DNN takes sensor readings as inputs, including airflow velocities, chemical concentrations, and indoor positions, and calculates robot heading commands, which navigate the robot moving toward the odor source.

Comparing existing olfactory-based navigation methods, the limitation of the bio-inspired methods is lacking the ability to estimate odor plume locations. When the robot loses plume contact, it can only conduct a time-consuming 'casting' behavior to re-detect plumes. As for an engineering-based method, the computational cost grows significantly with respect to the size of the search area and the resolution of the source mapping algorithm. The high computational cost for loop updating source estimates in every time step restrains its applications on mobile robots, which have the limited computational capacity. Thus, a desired olfactory-based navigation algorithm should be effective in different flow environments and light-weighted in computational demands for implementing on robotic agents.

Motivated by this consideration, we attempt to leverage the OSL problem via the deep learning (DL) methods. Benefits of training DNNs to solve the OSL problem include: 1) compared to engineering-based navigation methods, the query time of DNNs is predictable and unaffected by search environments, which is suitable for implementing on mobile robots; 2) DNNs can learn other successful navigation methods from demonstrations without explicating the specific navigation algorithm; 3) DNNs have the potential to continually improve the performance by adding more search examples in training data sets. However, the main challenge is collecting training data sets since OSL experiments are expensive to be repeatedly performed in different search conditions.

The main contribution of this work is to investigate the feasibility of implementing the DL algorithms on the OSL problem. The objective is to obtain a deep neural network (DNN) that navigates a plume-tracing robot to find the odor source without explicating odor search algorithms. Two types of DNNs, namely feedforward and long-short term memory [9] (FNN and LSTM), are devised to guide a mobile robot in finding the odor source. During the plume tracing process, DNNs calculate robot commands based on onboard sensor measurements. Two paradigms from categories of bio-inspired and engineering-based methods, namely moth-inspired [10] and Bayesian-inference [11] methods, are employed as expert methods to generate training data sets. Expert methods are implemented on a mobile robot (as shown in Fig. 1) in hundreds of repeated OSL tests. After the supervised training, the proposed DNNs are implemented in previously unseen search environments, i.e., new odor source locations and airflow fields, to validate their performance in the real-world scenario.

II. Related Works

Robotic OSL is a heated research topic in recent decades [12]. Thanks to the development of technologies in robotics and autonomous systems, implementing mobile robots to find an odor (or chemical) source becomes feasible. Early works of robotic OSL attempt to complete this task via a simple gradient following algorithm, i.e., chemotaxis. A common implementation is to install a pair of chemical sensors on the left and right sides of a plume-tracing robot, where the robot is commanded to steer to the side with the higher concentration measurement [13]. Many research works [14], [15], [16], [17] have proved the validity of chemotaxis in laminar flow environments (i.e., low Reynolds numbers).

In turbulent flow environments (i.e., high Reynolds numbers), bio-inspired and engineering-based methods were proposed to solve the OSL problem. Inspired by mate-seeking behaviors of male moths, the moth-inspired method has demonstrated its validity on many robotic OSL applications. Lochmatter *et al.* [18] implemented the moth-inspired method on a mobile robot in a laminar flow environment. Li *et al.* [19] applied the moth-inspired method on an autonomous underwater vehicle (AUV) to find an underwater chemical source over a large search area. Recently, Shigaki *et al.* [20] presented a time-varying moth-inspired method, where the duration of the 'surge' behavior is controlled via an equation obtained from observations of biological experiments. Besides, many other bio-inspired search strategies have been proposed, such as zigzag [21], spiral [22], fuzzy-inference [23], and multi-phase exploratory [24].

Engineering-based methods, on the other hand, navigate a robot relying on odor source estimations. To indicate the odor source distribution, constructing a source probability map is the common approach. This map divides the search area into multiple small regions and assigns every region with a probability, indicating how likely this region contains the odor source. Algorithms that construct a source probability map include the Bayesian-inference theory [11], particle filter [25], hidden Markov model (HMM) [26], occupancy grid mapping [27], source term estimation [28], [29], and partially observable Markov decision process (POMDP) [30]. After a source probability map is obtained, the robot is commanded to move toward the estimated source location (i.e., the region with the highest probability of containing the odor source) via a path planning algorithm. Possible path planners include the artificial potential field (APF) [31] and A-star [32] algorithms. Besides, Vergassola et al. [33] proposed the 'infotaxis' algorithm, which uses the information entropy to guide the robot searching for the odor source. In this method, the robot selects a future movement that mostly reduces the information uncertainty of the odor source.

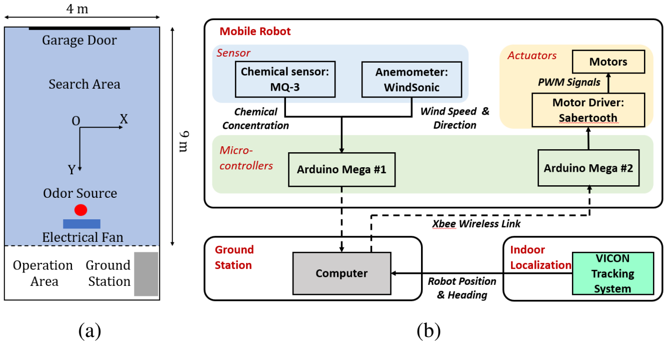
In the field of DL-based OSL methods, limited works have been carried. Recent developments include designing DNNs to predict the gas leaking locations from a stationary sensor network or adopting reinforcement learning algorithms to learn a plume tracing strategy. Kim *et al.* [34] trained a recurrent neural network (RNN) to predict possible odor source locations via the data obtained from a stationary sensor network, where the training data is acquired from a simulation program. Hu *et al.* [35] presented a plume tracing algorithm based on the model-free reinforcement learning algorithms. The deterministic policy gradient (DPG) is employed to train an actor-critic network (built upon LSTM networks), where the actor-network controls an AUV to search odor plumes and find the odor source (i.e., hydrothermal vents). Wang *et al.* [36] trained and implemented an adaptive neuro-fuzzy inference system (ANFIS) to solve the OSL problem in simulations, where real-world implementations are needed to prove its validity.

By summarising these works, it can be discovered that despite the high-level intelligence and potential benefits of DL technologies, using DL methods to solve an OSL problem is still in its infancy and requires further research. Besides, most

TABLE I
DEFINITIONS OF VARIABLES

Symbols of Variables	Definitions of Variables
t (s)	Algorithm running time
u (m/s)	Wind speed at the robot position
ϕ (rad)	Wind direction at the robot position
ρ (mmpv)	Odor concentration at the robot position
x (m)	Robot horizontal position
y (m)	Robot vertical position
ψ (rad)	Robot heading angle
v (m/s)	Robot speed
v_c (m/s)	Robot speed command
ψ_c (rad)	Robot heading command

mmpv: million molecules per cm^3



(a) (b)

Fig. 2. (a) Search area. Possible odor source locations in the training dataset are labeled with the red dot. (b) System configuration. This system contains three main components, including mobile robot, ground station, and indoor localization system. The solid connection line represents physical cables, and the dotted connection line represents wireless link.

DL-based OSL methods are validated in virtual environments, where flow fields and plume distributions are simulated. Real-world implementations are needed to proof the validity of DL methods in the OSL problem.

III. METHODOLOGY

A. DNN-Based Plume Tracing Algorithm

In this work, the OSL is considered as a two-dimensional (2-D) problem since the aimed robotic platform is a ground mobile robot. The main objective of this work is to obtain a DNN model that guides the mobile robot to locate an odor source in an unknown environment. To achieve this goal, the DNN model is trained to calculate robot commands, i.e., $\mathbf{C}$, based on states, i.e., $\mathbf{S}$:

$$\mathbf{C} = \mathcal{F}_\theta(\mathbf{S}). \tag{1}$$

This DNN model is parametrized by a parameter vector θ, and the optimal θ is found during the process of supervised training, which minimizes the difference between outputs of the DNN and the ones demonstrated by expert methods.

B. Generate Training Data Sets

1) Inputs and Outputs of DNNs: Two expert methods, namely moth-inspired [10] and Bayesian-inference methods [11], are employed to generate training data sets. These two navigation methods are chosen to generate the training dataset since they are paradigms in categories of bio-inspired and engineering-based methods, respectively, and they have been proofed to be valid in real-world experiments [37], [38].

To learn expert methods, DNNs should be offered with similar input information. In the moth-inspired method, odor concentrations (ρ) and wind directions (ϕ) are required to determine the robot search behaviors. For the Bayesian-inference method, robot positions (x and y), wind speeds (u_x and u_y), and algorithm running time (t) are essential to estimate odor source locations. To properly learn expert methods, aforementioned variables are included in DNN inputs. For outputs of DNNs, only speed and yaw angle commands (v_c and ψ_c) are needed to control the mobile robot on a 2-D plane, thus, these two variables are included in DNNs' outputs. It should be mentioned that all angle-related variables are converted into the vector forms, including wind directions (ϕ), robot yaw angles (ψ), and yaw angle commands (ψ_c):

$$\begin{cases} u_x = u\cos\phi, v_x = v\cos\psi, v_{c,x} = v_c\cos\psi_c \\ u_y = u\sin\phi, v_y = v\sin\psi, v_{c,y} = v_c\sin\psi_c \end{cases}. \tag{2}$$

This is because angles do not make a good DNN input: one angle could refer to two different values such as $-\pi$ and π, and angles should not matter if the corresponding speed is zero. For the easy reference, Table I lists variables and corresponding definitions in $\mathbf{S}$ and $\mathbf{C}$. Therefore, the input vector $\mathbf{S}$ can be defined as:

$$\mathbf{S} = (t, u_x, u_y, \rho, x, y, v_x, v_y), \tag{3}$$

where u_x, u_y, v_x and v_y are wind and robot speeds in x and y directions, respectively. Besides, DNN's outputs are defined as:

$$\mathbf{C} = (v_{c,x}, v_{c,y}). \tag{4}$$

2) Collect Training Data from Real OSL Experiments: To collect training data, the mobile robot, implemented with the expert navigation methods, was repeated tested in real-world OSL experiments.

The experiment field is in the indoor autonomous robots testing lab at the Embry-Riddle Aeronautical university. The lab is divided into two areas, including a search area where the robot can move and an operation area for accommodating the ground station. As shown in Fig. 2(a), the size of the search area is 9 × 4 m^2, containing an odor source and an electrical fan. Ethanol was employed as the odor source since it is minimally toxic and commonly implemented in the OSL research [39]. Besides, ethanol was put in a humidifier to consistently release odor plumes.

Fig 2(b) presents the configuration of the implemented robotic system, containing a mobile robot, a ground station, and an indoor localization system. The robot is equipped with a chemical sensor (MQ-3, Waveshare) and an anemometer (WindSonic, Gill Instruments), which are connected to a micro-controller (Arudino Mega, Arduino) for reading sensor measurements. The second onboard micro-controller controls

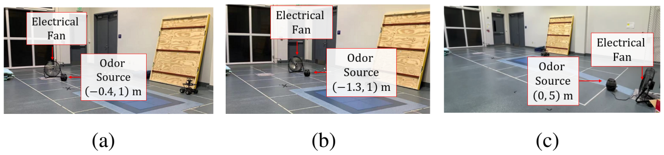


Fig. 3. Different odor source locations and airflow fields in the training dataset. (a) $(-0.4, 1)$ m. (b) $(-1.3, 1)$ m. (c) $(0, 5)$ m. The airflow direction for (a) and (b) points to the positive side of y axis, while the airflow direction for (c) points to the negative side of y axis.

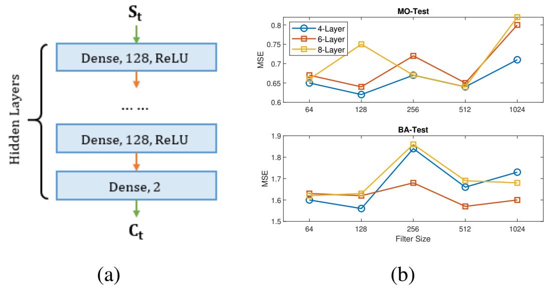


Fig. 4. (a) The structure of the proposed FNNs. Notations inside a blue layer represent the layer type, filter size, and activation function, respectively. A dense layer indicates a fully-connected neural network; (b) MSEs of FNNs with varying filter sizes and hidden layers on two testing data sets, including MO-test and BA-test.

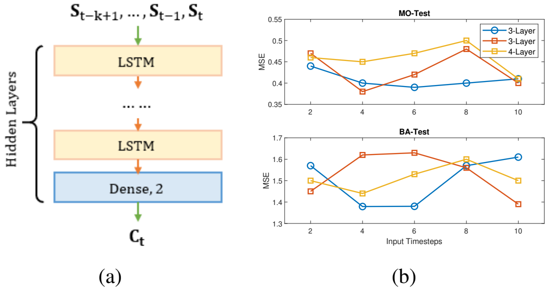


Fig. 5. (a) The structure of the proposed LSTM network. Inputs of the LSTM network include previous sensor readings, i.e., $\mathbf{S}_{t-k+1}$ to $\mathbf{S}_t$, where the length of inputs is k. Multiple LSTM cells are stacked to form a deep LSTM network; (b) MSEs of LSTM networks with the varying time steps of inputs (i.e., k) on testing data sets.

robot motors via a motor driver (Sabertooth, Dimension Engineering). Two micro-controllers can communicate with the ground station via a wireless communication network (Xbee, Digi international). The Vicon tracking system (Vicon Inc.) is employed to determine indoor positions, which sends robot positions and orientations to the ground station. The response time of all sensors were set to 0.25 s.

During an OSL trial, the robot sends sensor measurements to the ground station, where the navigation algorithm is implemented. Since the chemical sensor has a long recovery time, an adaptive concentration threshold [25] is employed to distinguish the odor detection and non-detection events. After the calculation in the ground station, robot commands will be transmitted to the mobile robot via the wireless communication network. Then, the robot moves toward the target heading and collects information at the new location. The transmission time between the robot and ground station is negligible (less than 1 ms). These processes repeat until the robot finds the odor source.

To collect training data, over 200 OSL trials were conducted, i.e., 100 trials for each expert method. In these OSL trials, the odor source was placed at three different positions as shown in Fig. 3, and the robot initial position was randomly selected in the downwind regions. During the plume tracing process, a data tuple that consists of the input state vector $\mathbf{S}_t$, which is obtained from sensor measurements, and expert command $\mathbf{C}_{exp,t}$, which is produced by the implemented expert method, is recorded at every time t. An OSL trial is considered as complete if the robot reaches the odor source location or the algorithm running time is beyond the time limit, i.e., 200 s. The odor source is considered as located if the robot is in vicinity of it (within 0.5 m). In real-world applications, the source declaration step could be completed with aids of external sensors such as cameras, which could recognize an odor source from a close distance.

3) Training Data Specifications: Depending on the type of the expert method, two training data sets, namely MO-Train (obtained from the moth-inspired method [10]) and BA-Train (obtained from the Bayesian-inference method [11]), are acquired.

During the training process, only 80% of training data is used to train DNN models, while 10% of the remaining data, termed testing data set, is used to test DNN models after the training, and the last 10% data, termed validation data set, is used to compute validation errors. The training process is terminated once the validation error is not improved in 20 episodes or the training epoch reaches the limit. The validation error is defined as mean absolute errors (MAEs), i.e., $1/n \cdot \sum_{i=1}^{n} |\mathcal{F}_\theta(\mathbf{S}_i) - \mathbf{C}_{i,exp}|$, where n is the size of validation data set.

C. Design DNNs for OSL Problems

As mentioned, two types of neural networks, i.e., FNN and LSTM, are employed for the representation of $\mathcal{F}_\theta$. The motivation for choosing FNN is that we want to use a simple DNN structure to investigate the viability of implementing DL approaches on OSL problems. Besides, the intuitive FNN could also be employed as the baseline to evaluate the performance of other types of DNN models in the OSL problem. To determine the optimal structure of the FNN network, varying numbers of hidden layers and filter sizes are evaluated with the testing data sets. Fig. 4 shows the mean square errors (MSEs, i.e., $1/m \cdot \sum_{i=1}^{m} (\mathcal{F}_\theta(\mathbf{S}_i) - \mathbf{C}_{i,exp})^2$, where m is the size of the testing dataset) of implementing different FNNs with varying hidden layers on testing data sets. It can be observed that larger models (i.e., more layers) achieve better performances (i.e., lower MSEs) but overfit (i.e., the MSE increases) when the model is too complicated. Based on plots, the FNN with 4-layer and 128 filters is selected for training both MO-train and BA-train.

Besides the simple FNN network, we also train a LSTM network. Unlike standard FNNs, LSTMs have feedback connections, which bring the previous output to the input at the current time step. In this work, inputs of a neural network are onboard sensor readings and outputs are robot commands. By using LSTMs, the sensor data history can be involved in the calculation of robot commands at the current time step. This feature may be useful in learning the Bayesian-inference method, which deduces possible odor source locations via sensed airflow history. Similar to FNNs, we determine the optimal structure of LSTM by examining multiple combinations of layer numbers and the size of input (i.e., how many time steps k are included in the input). Fig. 5 shows the plot of MSEs of implementing different structure LSTMs on testing data sets. Based on the plot, we choose the input time step as 4 for both MO-Train and BA-Train, where the number of LSTM layers is 3 for the MO-Train and 2 for the BA-Train.

D. Training DNN Models

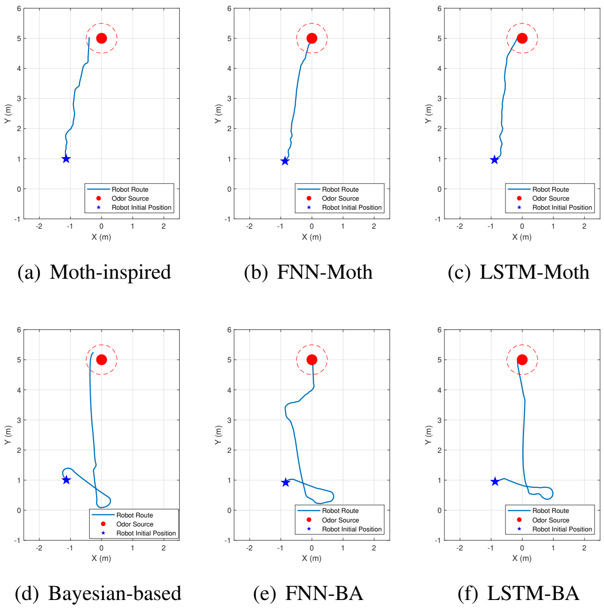
The supervised learning [40] is employed as the training algorithm to train DNNs, which aims to find the optimal parameter vector θ^* that minimizes the loss function J. The loss function J is defined as the mean square error between DNN outputs $\mathcal{F}_\theta(\mathbf{S})$ and expert demonstrations $\mathbf{C}_{exp}$, which can be represented as:

$$J\left(\Gamma_B\right)=\frac{1}{N_B}\cdot\sum_{i=j}^{j+N_B}\left(\mathcal{F}_\theta\left(\mathbf{S}_i\right)-\mathbf{C}_{i,exp}\right)^2, \tag{5}$$

where Γ_B is a mini-batch that contains N_B (32 in our work) samples from a training data set. The gradient of the cost function with respect to model parameters is calculated using the backpropagation algorithm [41], and the optimization algorithm that updates model parameters is the Adam optimizer [42].

Additionally, inputs and outputs are scaled before training a neural network. Unscaled input variables can result in a slow or unstable learning process, whereas unscaled target variables on regression problems can result in exploding gradients causing the learning process to fail [43]. Here, we normalize inputs and outputs before the training by subtracting the mean and dividing by the standard deviation of each feature. The training process is considered as complete if one of the following two conditions is satisfied: 1) the training epoch reaches the limit (i.e., 200 in implementations); 2) the validation error does not improve in 20 consecutive epochs. Google® TensorFlow [44] is employed as the framework to construct and train DNN models, and the training process is complete on an Intel® i7-8750 CPU with the Nvidia® GeForce GTX 1070 GPU acceleration.

After training with the training data sets generated by two expert methods, i.e., MO-Train and BA-Train, four DNNs are obtained, termed FNN-Moth, LSTM-Moth, FNN-BA, and LSTM-BA. This notation consists of the type of DNN and the corresponding training data set, e.g., FNN-Moth is the FNN trained with MO-Train.



(a) Moth-inspired (b) FNN-Moth (c) LSTM-Moth

(d) Bayesian-based (e) FNN-BA (f) LSTM-BA

Fig. 6. Six sample OSL trials with different navigation methods.

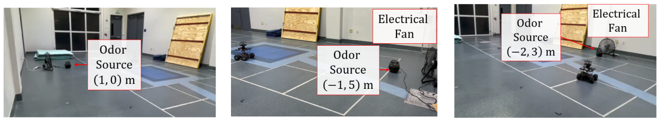
IV. Experiments

A. Sample OSL Trials

We first demonstrate the search trajectories generated by implementing the trained DNNs in a seen environment (i.e., Fig. 3(c)), where the odor source location is placed at $(0, 5)$ m and the airflow direction points to the negative side of the y axis.

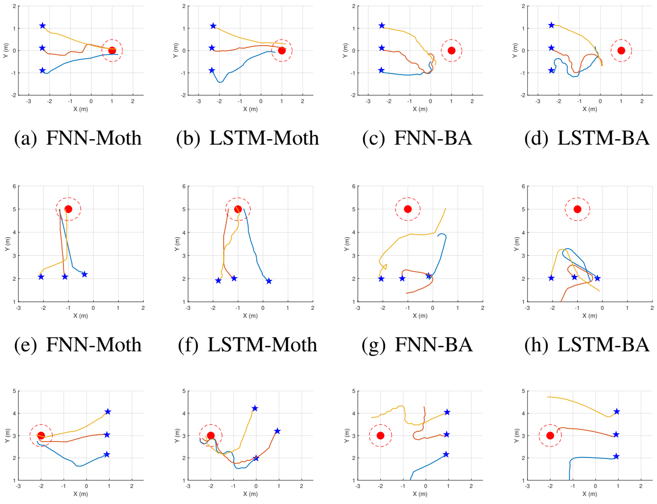
Fig. 6 presents search trajectories generated by expert methods and DNNs. Fig. 6(a), 6(b), and 6(c) show search trajectories generated by the original moth-inspired method, FNN-Moth, and LSTM-Moth, respectively. It can be seen that three search trajectories are very similar and all end up with the odor source location, indicating that the proposed DNNs can mimic the moth-inspired method to correctly find the odor source. As for the Bayesian-inference method (presented in Fig. 6(d)), the search trajectory produced by the LSTM-BA (i.e., Fig. 6(f)) achieves a shorter search time than the FNN-BA (i.e., Fig. 6(e)), indicating a better search performance. This is because the context understanding ability of the LSTM network is more suitable for learning the Bayesian-inference method, which requires the knowledge of the sensor reading history to estimate possible odor source locations.

In this group of tests, search trajectories generated by DNNs are similar to the expert methods, and both FNN and LSTM can find the odor source location in these sample OSL trials. However, it is possible that DNNs memorize the final odor source location instead of learning the actual searching strategies demonstrated by the expert methods. Thus, trained DNNs need to be examined in previously unseen environments to verify their validities.



(a) (b) (c)

Fig. 7. Unseen search environment where odor source locations are located at (a) $(1, 0)$ m, (b) $(-1, 5)$ m, and (c) $(-2, 3)$ m. New airflow directions are created in unseen environments, where the airflow direction points to (a) negative side of x axis, (b) negative side of y axis, and (c) positive side of x axis.



(a) FNN-Moth (b) LSTM-Moth (c) FNN-BA (d) LSTM-BA

(e) FNN-Moth (f) LSTM-Moth (g) FNN-BA (h) LSTM-BA

(i) FNN-Moth (j) LSTM-Moth (k) FNN-BA (l) LSTM-BA

Fig. 8. Robot search trajectories generated by the proposed DNNs with different robot initial positions in previous unseen environments. In these diagrams, blue stars indicate robot initial positions, the red dot represents the odor source location, and the red dotted line shows the source localization range (i.e., 0.5 m): the odor source is considered as located if the robot is inside this range.

B. Search Results in Unseen Environments

In this group of tests, trained DNNs are tested in unseen environments. As shown in Fig. 7, we choose three new odor source locations and airflow directions to construct the testing environment, and for each new odor source location, various OSL trials were conducted with different robot initial positions.

Fig. 8 shows the search results of trained DNNs in unseen environments. It can be observed that both FNN-Moth and LSTM-Moth networks, trained by the moth-inspired method, can correctly find the odor source in new environments. In these trials, the robot demonstrates the ability to mimic the moth-inspired method, which can effectively find the odor source by performing the upwind movement (similar to the 'surge' behavior) or the crosswind excursion (like the 'casting' behavior) to move toward the odor source. By contrast, the performance of FNN-BA and LSTM-BA, trained with the Bayesian-inference method, is not satisfied, where only a few of successful tests.

TABLE II
Search Results of Four Navigation Methods in Repeated Tests. μ: Mean Search Time; σ: Standard Deviation

Test Name	Moth-inspired Method (s)	Bayesian-inference Method (s)	FNN-Moth (s)	LSTM-Moth (s)
Test 1	18	40	21	20
Test 2	21	24	19	20
Test 3	19	23	20	21
Test 4	20	22	20	21
Test 5	19	23	20	21
Test 6	26	28	25	30
Test 7	27	28	28	28
Test 8	28	42	25	26
Test 9	25	30	30	29
Test 10	27	27	27	30
μ (s)	**23.0**	28.7	**23.5**	24.6
σ	**3.9**	7.0	**4.0**	4.4

This result is anticipated since the search strategy in the moth-inspired method is much simpler than the one demonstrated by the Bayesian-inference method, which finds the odor source relying on mathematical calculations. For the Bayesian-inference method, the underlying calculation process cannot directly reflect on the robot search behaviors, making it difficult for DNNs to mimic its search strategy. Given the same amount of training data sets, DNNs are more likely to learn the simple moth-inspired method than the Bayesian-inference counterpart. One possible solution is adding more demonstrations with different odor source locations to diversify the training data set. Still, the increasing time and efforts to collect training data and raise training time are also significant.

C. Compare with Expert Methods in Repeated Tests

In this group of tests, we implement the FNN-Moth and LSTM-Moth networks in an unseen environment, where the odor source is located at $(-1, 5)$ m. To evaluate the search performance, two DNNs are compared with two expert methods. For each navigation method, 10 OSL trials were conducted. The robot initial position is at $(-0.7, 2)$ m in Test 1-5 and changes to a far position at $(-1.2, 1)$ m in Test 6-10. Search results are reported in Table II.

It can be observed in Table II that all navigation methods can correctly navigate the robot to find the odor source. In terms of the averaged search time (μ) and standard deviation (σ), the moth-inspired method achieves the best performance among other methods. Two DNNs attain a comparable search performance with the moth-inspired method. Moreover, the FNN-Moth is slightly better than the LSTM-Moth network due to the lower averaged search time (23.5 s v.s. 24.6 s) and standard deviation (4.0 v.s. 4.4). Repeated tests reflect that both neural networks can effectively learn the moth-inspired method to locate an odor source in a new search environment.

V. Conclusion and Future Works

This paper presents DL-based plume tracing algorithms. Two DNNs are devised, including FNN and LSTM networks. DNNs calculate robot commands that navigate the robot to find the odor source based on onboard sensor readings during the plume tracing process. Two expert methods, namely moth-inspired and Bayesian-inference methods, are employed as expert methods to generated training data sets. Hundreds of OSL trials were conducted to collect training data, and after training, DNNs are validated in unseen search environments. Experiment results show that both FNN and LSTM can mimic the moth-inspired method but cannot effectively learn the Bayesian-inference method given the same amount of training data. In repeated tests, the search performance of DNNs is comparable with the expert method.

The experimental idea presented in this paper shows a new direction to approach the robotic OSL problem: using DNNs to learn an olfactory-based navigation algorithm. With the limited training data set, experiment results show that DNNs can effectively learn a simple plume tracing strategy but not a complex one. In the future, the training data set could be diversified with more demonstrations, including different odor source locations and navigation methods. Besides, other AI algorithms, such as reinforcement learning, could also be implemented to solve the robotic OSL problem.

REFERENCES

[1] G. Kowadlo and R. A. Russell, "Robot odor localization: a taxonomy and survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[2] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.

[3] S. Soldan, G. Bonow, and A. Kroll, "Robogasinspector-a mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results," *IFAC Proceedings Volumes*, vol. 45, no. 8, pp. 33–38, 2012.

[4] R. A. Russell, "Robotic location of underground chemical sources," *Robotica*, vol. 22, no. 1, pp. 109–115, 2004.

[5] G. Ferri, M. V. Jakuba, and D. R. Yoerger, "A novel method for hydrothermal vents prospecting using an autonomous underwater robot," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1055–1060.

[6] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. T. Cardé, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," *Environmental fluid mechanics*, vol. 2, no. 1-2, pp. 143–169, 2002.

[7] X. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robotics and Autonomous Systems*, vol. 112, pp. 123–136, 2019.

[8] R. T. Cardé and A. Mafra-Neto, "Mechanisms of flight of male moths to pheromone," in *Insect pheromone research*. Springer, 1997, pp. 275–290.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] J. A. Farrell, S. Pang, and W. Li, "Chemical plume tracing via an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 428–442, 2005.

[11] S. Pang and J. A. Farrell, "Chemical plume source localization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1068–1080, 2006.

[12] T. Jing, Q.-H. Meng, and H. Ishida, "Recent progress and trend of robot odor source localization," *IEEJ Transactions on Electrical and Electronic Engineering*, 2021.

[13] G. Sandini, G. Lucarini, and M. Varoli, "Gradient driven self-organizing systems," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 1. IEEE, 1993, pp. 429–432.

[14] F. W. Grasso, T. R. Consi, D. C. Mountain, and J. Atema, "Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 115–131, 2000.

[15] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robotics and Autonomous Systems*, vol. 45, no. 2, pp. 83–97, 2003.

[16] A. Lilienthal and T. Duckett, "Experimental analysis of gas-sensitive Braitenberg vehicles," *Advanced Robotics*, vol. 18, no. 8, pp. 817–834, 2004.

[17] H. Ishida, G. Nakayama, T. Nakamoto, and T. Moriizumi, "Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors," *IEEE Sensors Journal*, vol. 5, no. 3, pp. 537–545, 2005.

[18] T. Lochmatter, X. Raemy, L. Matthey, S. Indra, and A. Martinoli, "A comparison of casting and spiraling algorithms for odor source localization in laminar flow," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1138–1143.

[19] W. Li, J. A. Farrell, S. Pang, and R. M. Arrieta, "Moth-inspired chemical plume tracing on an autonomous underwater vehicle," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 292–307, 2006.

[20] S. Shigaki, T. Sakurai, N. Ando, D. Kurabayashi, and R. Kanzaki, "Time-varying moth-inspired algorithm for chemical plume tracing in turbulent environment," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 76–83, 2017.

[21] T. Lochmatter and A. Martinoli, "Tracking odor plumes in a laminar wind field with bio-inspired algorithms," in *Experimental robotics*. Springer, 2009, pp. 473–482.

[22] F. Rahbar, A. Marjovi, P. Kibleur, and A. Martinoli, "A 3-d bio-inspired odor source localization and its validation in realistic environmental conditions," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3983–3989.

[23] S. Shigaki, Y. Shiota, D. Kurabayashi, and R. Kanzaki, "Modeling of the adaptive chemical plume tracing algorithm of an insect using fuzzy inference," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 1, pp. 72–84, 2019.

[24] P. Pyk, S. B. i Badia, U. Bernardet, P. Knüsel, M. Carlsson, J. Gu, E. Chanie, B. S. Hansson, T. C. Pearce, and P. F. Verschure, "An artificial moth: Chemical source localization using a robot based neuronal model of moth optomotor anemotactic search," *Autonomous Robots*, vol. 20, no. 3, pp. 197–213, 2006.

[25] J. Li, Q. Meng, Y. Wang, and M. Zeng, "Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm," *Autonomous Robots*, vol. 30, no. 3, pp. 281–292, 2011.

[26] J. A. Farrell, S. Pang, and W. Li, "Plume mapping via hidden Markov methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 850–863, 2003.

[27] M. V. Jakuba, "Stochastic mapping for chemical plume source localization with application to autonomous hydrothermal vent discovery," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

[28] F. Rahbar, A. Marjovi, and A. Martinoli, "An algorithm for odor source localization based on source term estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 973–979.

[29] M. Hutchinson, C. Liu, and W.-H. Chen, "Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2388–2402, 2018.

[30] H. Jiu, Y. Chen, W. Deng, and S. Pang, "Underwater chemical plume tracing based on partially observable markov decision process," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419831874, 2019.

[31] S. Pang and F. Zhu, "Reactive planning for olfactory-based mobile robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4375–4380.

[32] L. Wang and S. Pang, "Chemical plume tracing using an auv based on pomdp source mapping and a-star path planning," in *OCEANS 2019 MTS/IEEE SEATTLE*. IEEE, 2019, pp. 1–7.

[33] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, p. 406, 2007.

[34] H. Kim, M. Park, C. W. Kim, and D. Shin, "Source localization for hazardous material release in an outdoor chemical plant via a combination of lstm-rnn and cfd simulation," *Computers & Chemical Engineering*, vol. 125, pp. 476–489, 2019.

[35] H. Hu, S. Song, and C. P. Chen, "Plume tracing via model-free reinforcement learning method," *IEEE transactions on neural networks and learning systems*, 2019.

[36] L. Wang and S. Pang, "An implementation of the adaptive neuro-fuzzy inference system (anfis) for odor source localization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021.

[37] J. A. Farrell, S. Pang, W. Li, and R. Arrieta, "Chemical plume tracing experimental results with a remus auv," in *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, vol. 2. IEEE, 2003, pp. 962–968.

[38] H.-f. Jiu, S. Pang, J.-l. Li, and B. Han, "Odor plume source localization with a pioneer 3 mobile robot in an indoor airflow environment," in *IEEE SOUTHEASTCON 2014*. IEEE, 2014, pp. 1–6.

[39] Q. Feng, H. Cai, Z. Chen, Y. Yang, J. Lu, F. Li, J. Xu, and X. Li, "Experimental study on a comprehensive particle swarm optimization method for locating contaminant sources in dynamic indoor environments with mechanical ventilation," *Energy and buildings*, vol. 196, pp. 145–156, 2019.

[40] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[41] H. J. Kelley, "Gradient theory of optimal flight paths," *Ars Journal*, vol. 30, no. 10, pp. 947–954, 1960.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[44] S. S. Girija, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *Software available from tensorflow. org*, vol. 39, no. 9, 2016.