

A knowledge-driven framework for Robotic Odor Source Localization using large language models[☆]

Khan Raqib Mahmud^a, Lingxiao Wang^{b,*}, Sunzid Hassan^a, Zheng Zhang^c

^a Department of Computer Science, Louisiana Tech University, Ruston, 71272, LA, USA

^b Department of Electrical Engineering, Louisiana Tech University, Ruston, 71272, LA, USA

^c Department of Aerospace Engineering, Embry-Riddle Aeronautical University, Daytona Beach, 32114, FL, USA

ARTICLE INFO

Keywords:

Robotic Odor Source Localization
Large Language Models (LLMs)
Knowledge driven framework
Deep Q-Network (DQN)

ABSTRACT

Robotic Odor Source Localization (OSL) technology enables mobile robots to detect and navigate unknown odor sources in diverse environments. Traditional OSL methods, including bio-inspired, engineering-based, and machine learning-based approaches, face limitations of lack of adaptability to varying environments, significant computational resource requirements, and dependence on historical data. To overcome these challenges, we present a knowledge-driven framework that leverages Large Language Models (LLMs) to improve the robot's navigation capabilities through contextual understanding and informed decision-making. A key feature of the proposed work is integrating an LLM agent with a memory module, which stores past experiences and recalls them during the decision-making process, allowing the robotic agent to make decisions based on current sensory inputs and previously acquired knowledge. Compared to traditional deep learning-based methods, such as Deep Q-Network (DQN), both simulation and real-world experiment results demonstrate that our framework significantly outperforms it in terms of accuracy, efficiency, and generalization across different environmental conditions.

1. Introduction

Olfaction, also known as the sense of smell, provides crucial information about the environment. In nature, animals depend on olfaction to perform life-essential activities, such as foraging [1], homing [2], mate-seeking [3], and navigating in the environment [4,5]. A mobile robot, integrated with gas or chemical sensors, could detect and trace odors to find an unknown odor source within an environment. The technology of using robots to locate odor sources is known as robotic Odor Source Localization (OSL) [6,7]. In practice, robotic OSL finds its applications in air pollutant localization [8], wildfire smoke tracing [9], gas emission monitoring [10], and the detection of ocean hydrothermal vents [11], among others.

The key to successfully locating an odor source is the design of an effective navigation algorithm, i.e., the OSL algorithm, which guides the robot moving toward the odor source location based on sensor observations. Present OSL algorithms [12] include bio-inspired methods [13–19] that mimic animal behaviors, engineering-based methods [20,21], that use mathematical models, and machine learning-based methods that rely on trained models. Bio-inspired methods, like the moth-inspired algorithm, command a mobile robot to find an odor source by

mimicking animal odor-searching behaviors [22–24]. However, these methods follow pre-determined behavioral patterns that may not adapt well to airflow-varying environments [25]. Engineering-based methods, such as the particle filter-based algorithm, rely on mathematical models to derive possible odor source locations and direct the robot to the estimated source locations [26]. The downside of the engineering-based method is computational complexity [27]. Since most engineering-based methods divide the search area into multiple cells, and for each cell, there will be a probability calculation to calculate the chance of this cell containing the odor source. The computational complexity will increase exponentially if the search area is large and broad. Machine learning-based methods, including deep supervised and reinforcement learning (RL) techniques, often require extensive training data and computational resources [28]. These algorithms aim to model the complex patterns within sensory data collected from various environments. However, they frequently face issues like dataset bias, overfitting, and a lack of interpretability. Addressing these challenges is crucial for gaining a deeper understanding of odor dispersion patterns and making more rational decisions, which could significantly improve the effectiveness of OSL systems.

[☆] This work is supported by Louisiana Board of Regents, United States with the contract number: LEQSF(2024-27)-RD-A-22.

* Corresponding author.

E-mail address: lwang@latech.edu (L. Wang).

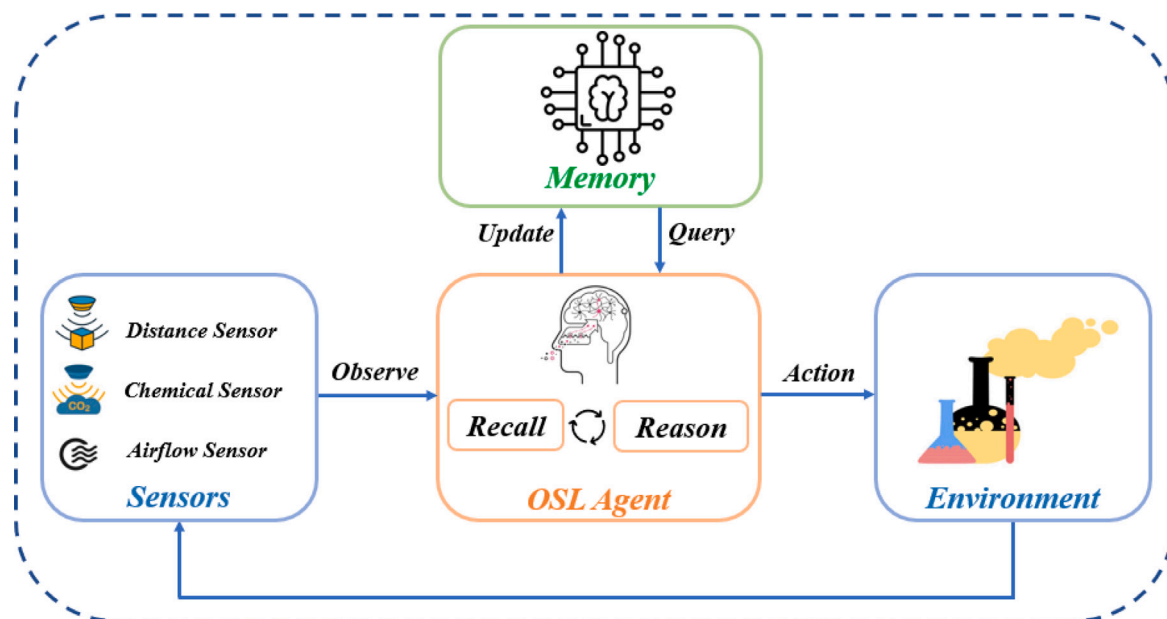


Fig. 1. The knowledge-driven paradigm for robotic odor source localization, including an interactive environment, an OSL agent with recall and reasoning abilities, and an independent memory module. The OSL agent continuously processes sensory inputs from the environment, queries and updates experiences from the memory module, and makes informed decisions to navigate toward the odor source.

Drawing inspiration from the profound capabilities of human cognition, we explore the core principles that underlie effective odor localization and raise a pivotal distinction: traditional OSL methods are fundamentally data-driven, whereas human olfactory behaviors are knowledge-driven [29]. For instance, when faced with a complex odor plume in a turbulent environment, humans can rely on contextual understanding and reasoning to navigate toward the source. Conversely, data-driven methods rely heavily on a large quantity of similar data to fit these scenarios, which limits their ability to generalize across diverse conditions. Additionally, collecting and annotating large datasets for training OSL models is labor-intensive and costly.

The knowledge-driven approach has emerged as a promising alternative to traditional methods in recent years. Unlike data-driven models that rely solely on large datasets, knowledge-driven models incorporate contextual understanding, reasoning, and decision-making capabilities similar to human cognition. This approach is particularly powerful in scenarios where understanding and interpreting complex environments is crucial. Large Language Models (LLMs), which have demonstrated exceptional abilities in various domains, embody this knowledge-driven paradigm [30,31]. LLMs leverage vast amounts of pre-existing knowledge, enabling them to generalize better across diverse scenarios and make informed decisions with limited contextual data [32,33]. Recent advancements in LLMs with emergent abilities offer an ideal embodiment of human knowledge, providing valuable insights toward addressing the challenges of OSL. LLMs possess exceptional human-level abilities and show strong performance in areas such as robotics manipulation [34], multi-modal understanding [35], and lifelong skill learning [36]. However, like humans need practice to master complex tasks, LLMs also require experience and guidance to perform effectively in specific applications.

To leverage the potential of LLMs for robotic OSL, we adapt a novel framework that integrates these models into a knowledge-driven approach [32]. This framework, illustrated in Fig. 1, incorporates several key components: a sensors module, an interactive environment, an OSL agent with a reasoning module, and a memory module to store and recall experiences. The reasoning module uses the LLM to query stored experiences from the memory module and apply common-sense knowledge to make informed decisions based on current scenarios. This process involves continuous evolution, where the agent observes

the environment, queries, and updates experiences from memory, and makes decisions.

Our research presents a novel framework for robotic OSL that leverages the capabilities of LLMs. To the best of our knowledge, we are the first to work to integrate LLM with robotic OSL problems. This knowledge-driven approach overcomes the limitations of traditional methods by integrating reasoning, contextual understanding, and scenario interpretation into the decision-making process. We compare the performance of our LLM-based framework with Deep Q-Networks (DQNs), a popular RL method [37]. Both simulation and real-world experiment results demonstrate that the LLM-based approach not only outperforms DQNs in terms of accuracy and efficiency but also exhibits superior generalization across different environments.

The contributions of our article are summarized as follows:

1. We introduce a knowledge-driven framework that leverages LLMs for robotic OSL, emphasizing the advantages of integrating reasoning, contextual understanding, and decision-making processes.
2. We validate the proposed OSL framework under realistic conditions by collecting real-world plume data using a wind tunnel and Particle Image Velocimetry (PIV) system.
3. We compare the performance of the LLM-based OSL framework with traditional DQN methods in both simulated and real-world search environments, demonstrating the superior adaptability and efficiency of the LLM-based approach.

2. Related works

2.1. Robotic Odor Source Localization (OSL)

2.1.1. Bio-inspired methods

Bio-inspired methods for robotic OSL draw inspiration from the natural world, particularly from the behaviors and mechanisms employed by various organisms to locate odor sources. These methods have been adapted into algorithms and systems to enhance the efficiency and adaptability of robots in tasks such as detecting and localizing chemical compounds or gases in the air, which is crucial for applications such as environmental monitoring, search and rescue operations, and detecting

gas leaks or hazardous substances. One approach involves mimicking the adaptability of insect brains to control a robot's movement based on sensory feedback, as demonstrated by a brain-machine hybrid system that adjusts the robot's velocity in response to neural activities descending from an insect's brain [38]. Similarly, the flight patterns of moths have inspired algorithms that trigger motion in robots upon detecting gas, integrating a repulsion function to navigate around obstacles [39]. The behavior of the adult male silk moth, particularly its method of modulating speed based on odor detection frequency, has been used to develop a robust moth-inspired algorithm for indoor and outdoor odor source localization [40]. This approach leverages the moths' ability to use smell and wind direction to locate mates, showing excellent performance in various environmental complexities [41,42]. Other organisms, such as flatworms, have also inspired bio-inspired OSL methods. Flatworms' kinesis response and tropotaxis behavior, which do not require wind direction or odor concentration information, have led to a bionic odor source localization algorithm that improves search efficiency and environmental adaptability [43]. Additionally, a method mimicking male moths tracking pheromone plumes uses simulations of their sensory, behavior, and control systems. Despite advancements, these models still fall short of natural efficiency, prompting optimization through genetic algorithms [44]. Incorporating biological components, such as moth antennae, into flying robots has provided rapid response times and high specificity and sensitivity in chemical detection due to gene editing advances [45]. Additionally, the collective behavior observed in nature has led to the development of swarm intelligence algorithms for multi-robot systems, enhancing parallelism, scalability, and robustness in odor source localization tasks [46]. Particle Swarm Optimization (PSO), based on the social behavior of birds and fish, has also been adapted for odor localization, proving effective in real-world scenarios [47–49]. Crosswind Levy Walks, spiraling, and upwind surges, inspired by animal search patterns, form a robust 3D algorithm for odor localization, outperforming 2D methods [50]. Additionally, biohybrid systems combining living materials with synthetic devices have led to robots with insect antennae for odor sensing, allowing autonomous navigation and obstacle avoidance [51].

However, bio-inspired methods face limitations. The complexity of accurately mimicking biological systems can lead to challenges in learning and implementing such methods effectively. Moreover, the reliance on specific sensor types, such as metal oxide semiconductor (MOS) sensors, can limit the sensing capacity of OSL robots, making it difficult to match the performance of their biological counterparts [52].

2.1.2. Engineering-based methods

Robotic OSL employs various engineering-based methods to enable robots to detect and locate the sources of odors. These methods range from algorithmic approaches to the utilization of unmanned aerial vehicles (UAVs) for enhanced mobility and efficiency. One primary algorithmic approach involves the use of Independent Posteriors (IP) and Dempster-Shafer (DS) theory algorithms, which rely on occupancy grid mapping to estimate the probability of a location being an odor source. The IP algorithm has shown superior performance in minimizing false source attributions in turbulent fluid flow environments [53]. For scenarios where constructing a dispersion model of the odor plume is impractical due to complex geometries, a combination of Infotaxis and Dijkstra algorithms has been proposed. This approach dynamically adjusts the robot's focus between exploration and exploitation, significantly improving success rates and reducing search times [54]. In the realm of UAVs, a multi-UAV collaboration based on a collaborative particle filter algorithm and an adaptive path planning algorithm has been developed. This method aims to quickly locate odor sources with minimal resource consumption, demonstrating superior performance in simulation platforms [52]. Olfactory quadruped robots equipped with various sensors have been developed for complex environment navigation and odor source localization, offering adaptability

and eco-friendliness [55]. These diverse engineering-based methods highlight the interdisciplinary efforts to improve robotic OSL, leveraging algorithmic precision, UAV mobility, and bio-inspired strategies to address the challenges of locating odor sources in complex environments. Engineering-based methods, such as those employing UAVs for OSL, benefit from flexible deployment and controllable movement in 3D space, offering precise source estimation and efficient navigation through collaborative algorithms and adaptive path planning [27]. These methods often rely on well-defined computational models and algorithms, like Fuzzy inference and Markov decision processes, to optimize search strategies in turbulent flow environments. While these approaches provide clear frameworks for problem-solving and optimization, they may lack the inherent adaptability and robustness to environmental variability that bio-inspired methods offer.

In summary, bio-inspired methods for robotic OSL offer adaptability, efficiency, and robustness by leveraging natural strategies, but they may struggle with the complexity of certain algorithms and sensor limitations. Conversely, engineering-based methods provide precise, optimized solutions through computational models but may lack the adaptability to environmental changes seen in bio-inspired approaches.

2.1.3. Learning-based methods

Learning-based methods for robotic OSL have significantly advanced, leveraging deep learning (DL) and reinforcement learning (RL) to enhance robots' ability to detect and locate odor sources efficiently. DL has emerged as a promising approach for robotic OSL, offering the potential to navigate and identify odor sources with high accuracy. Deep learning methods, particularly, have been instrumental in developing algorithms that enable mobile robots to navigate toward an odor source without predefined search strategies. Two notable deep neural networks (DNNs), feedforward neural networks (FNN) and long short-term memory neural networks (LSTM), have been developed to calculate robot heading commands based on sensor readings, showing promising results in real-world experiments [56]. Additionally, a deep learning-based odor compass has been designed, incorporating a deep learning-based odor attention (DL-OA) model with a separated spatial-temporal attention-based encoder-decoder structure for end-to-end odor source direction estimation (OSDE), demonstrating significant accuracy in indoor environments [57]. Another approach involves a convolutional neural network (CNN) and LSTM modules to improve the accuracy and generalization ability of odor-source direction estimation, further enhancing the performance of OSL robots [58]. On the reinforcement learning front, a multi-continuous-output Takagi-Sugeno-Kang fuzzy system tuned with reinforcement learning has been proposed. This system, designed for dynamic outdoor environments, relies on the robot's observations to guide it toward the odor source, showing comparable success rates and higher efficiency than manually tuned systems [59]. Moreover, the integration of probabilistic models, such as probabilistic gas-hit maps, into the robotic systems provides a higher level of abstraction to model the time-dependent nature of gas dispersion, aiding in source localization in complex indoor environments [60]. The integration of convolutional neural network (CNN) and LSTM modules into the odor compass design further improves accuracy and generalization ability in OSL tasks [61]. Deep learning frameworks also excel in handling sparse and unreliable spatio-temporal chemical sensor data, offering regularized solutions that accurately predict gas leak sources by conforming to the spatio-temporal structure of gas concentration distribution [62]. These advancements underscore the potential of AI-based methods in enhancing the efficiency and reliability of robotic odor source localization.

However, several challenges and limitations are associated with its application, which necessitates further research and development to address. One significant challenge is the complexity of environmental factors, such as signal noise, obstacles, and sparse fingerprints, which can hinder the modeling and localization performance of robots in indoor environments. To overcome this, a novel deep learning framework with

a localization attention module and a multi-faceted localization module integrating LSTM and GRU has been proposed, showing efficiency in capturing dynamic spatial and temporal features [63]. Another limitation is the difficulty in learning complex search strategies, such as Bayesian-inference methods, through DL. Experiments have shown that while DL models can imitate simpler moth-inspired methods, they struggle with more complex strategies [56]. Addressing this requires the development of more sophisticated neural network architectures or training methodologies.

In summary, addressing the challenges and limitations of using deep learning for robotic OSL requires a multifaceted approach, including the development of advanced DL frameworks, improved sensor technologies, sophisticated training methodologies, and the integration of biological insights into algorithm design [41].

2.2. LLM-based agent

2.2.1. Large language models

Large language models (LLMs) have significantly transformed the landscape of natural language processing (NLP), pushing the boundaries of language understanding and generation to new heights. These models, with their vast amount of parameters frequently reaching the hundreds of billions, are trained on extensive text datasets. This in-depth training enables them to grasp natural language and perform various intricate tasks, primarily focusing on text generation and comprehension. Notable examples of LLMs include GPT-3 [64], PaLM [65], LLaMA [66], and GPT-4 [67]. Constructed using the transformer architecture, LLMs have demonstrated significant advancements in performance compared to earlier versions due to utilizing extensive data and complex training methods. LLMs stand out from smaller language models due to their emergent capabilities, including in-context learning [64], following instructions [68,69], and reasoning with chain-of-thought [70]. The creation and implementation of these models have achieved state-of-the-art performance across numerous NLP tasks, introducing a paradigm shift in how these tasks are approached. They leverage pre-training on large datasets followed by fine-tuning for specific applications [71]. LLMs' capacity to produce coherent and contextually appropriate text makes them extremely useful for various applications in educational technology, computational social science, and beyond. These models have demonstrated exceptional capabilities in generating text, understanding complex language nuances, and performing tasks with minimal input through zero-shot or one-shot learning settings [72]. The evolution of LLMs, highlighted by their ability to perform tasks zero-shot without specific training data, is transforming computational social science by serving as zero-shot data annotators and bootstrapping creative generation tasks, showcasing their versatility and broad applicability [73].

Recent advancements in LLMs have showcased human-like intelligence and hold the potential to propel us closer to the realm of Artificial General Intelligence (AGI) [73]. OpenAI's pursuit of LLMs has led to significant milestones such as ChatGPT [74] and GPT-4 [67]. These milestones signify notable advancements in LLMs' capabilities, particularly in natural language understanding and generation. The continuous development of LLMs, including refining architectures and training strategies, promises further advancements in their capabilities and applications. These models have become essential tools in various domains, driving innovation and enhancing the ability to tackle complex problems through advanced language understanding and generation.

2.2.2. Robotic Transformer

The concept of a "Robotic Transformer" encompasses a range of innovative approaches in robotics, focusing on enhancing the capabilities of robots through advanced machine learning models, particularly transformers, for tasks such as 3D object manipulation, robotic grasping, and skill assessment in robot-assisted activities. In the realm of 3D object manipulation, the Robotic View Transformer (RVT) represents

a significant advancement, offering a scalable and accurate multi-view transformer model that outperforms existing methods in both training speed and inference speed, demonstrating its effectiveness across a variety of real-world tasks with minimal demonstrations required [75]. Similarly, the Robotics Transformer model emphasizes the importance of transferring knowledge from large, diverse datasets to solve specific downstream tasks, showcasing the potential for generalization in robotics through open-ended task-agnostic training [76]. For robotic grasping, extending transformer models to 6-Degree-of-Freedom (6-DoF) grasping has shown promising results, with methods that efficiently learn both global and local features, significantly improving success rates in challenging datasets [77]. Additionally, Act3D introduces a manipulation policy transformer that excels in 3D detection for end-effector pose prediction, setting new benchmarks in manipulation tasks through its innovative use of 3D feature clouds and adaptive spatial computation [78]. These transformer-based approaches demonstrate significant advancements in robotic manipulation tasks, showcasing superior performance, scalability, and generalization abilities.

2.2.3. LLM-based agent in robotic tasks

Current advancements in Large Language Model (LLM)-based agent technology have significantly enhanced the capabilities of robots in understanding and executing complex tasks. These advancements leverage the vast knowledge encoded in LLMs, extending beyond simple prompt engineering to enable more nuanced and situationally aware interactions between robots and their environments. A cognitive-agent approach has been developed to mitigate the limitations of prompt engineering, allowing robots to acquire new task knowledge that aligns with their native language capabilities, embodiment, environment, and user preferences. This approach has demonstrated the ability of robots to achieve high task completion rates in one-shot learning scenarios, with and without human oversight [79].

Moreover, the integration of LLMs with reinforcement learning has led to the development of mediator models that optimize the cost and frequency of interactions between agents and LLMs. These models enable agents to consult LLMs only when necessary, significantly reducing interaction costs and improving performance in complex decision-making tasks [80]. LLMs have shown significant promise in enhancing robotic task learning, particularly by providing a rich knowledge source that can be tapped into for novel task acquisition. One promising approach leverages LLMs for object goal navigation in a zero-shot manner, where an embodied agent navigates to a target object described in natural language within an unexplored environment. This method has demonstrated substantial improvements in success rates over current baselines [81], utilizing the implicit knowledge of LLMs about the semantic context of the environment and mapping it into sequential inputs for robot motion planning. Furthermore, LLMs integrated with visual and natural language understanding have been explored for incremental decision-making in real-world environments, such as Vision and Language Navigation (VLN), where an embodied agent follows navigation instructions grounded in real-world observations [82].

The concept of using LLMs as a 'robotic brain' to unify memory and control within an embodied AI system has been introduced, demonstrating the potential of LLMs in active exploration and embodied question answering tasks [83]. This approach not only streamlines the interaction between perception, planning, and control but also significantly boosts the efficiency and accuracy of robotic tasks. A closed-loop technique, AdaPlanner, has been proposed to allow LLM agents to adaptively refine their plans in response to environmental feedback, showing improved performance in sequential decision-making tasks [84].

In the context of memory utilization, Retrieval-Augmented Generation (RAG) has emerged as a highly relevant approach that combines retrieval mechanisms with LLM generation tasks. RAG typically retrieves external documents or passages to directly augment LLM outputs in a single-stage process. RAG has demonstrated advantages in enhancing

knowledge integration, processing complex queries, and improving retrieval efficiency [85]. Its scalability and adaptability to dynamic environments, facilitated by mechanisms such as topology-aware retrieval and relevant information gain, make it a strong theoretical foundation for structured memory-driven tasks [86]. However, RAG's dependency on external data sources and the challenges of filtering retrieved information demonstrate the importance for domain-specific adjustments to maximize its effectiveness [87].

LLM-based agents demonstrate exceptional proficiency in conversational engagement and adherence to instructions across various downstream tasks, outperforming rule-based systems that require explicit programming for each specific task [88]. These agents can process and generate natural language instructions, adapt to new tasks with minimal data, and engage in complex decision-making processes. However, LLMs lack the inherent ability to engage with and comprehend the complexities of odor source localization as effectively as humans. Robotic OSL systems require active interaction with and understanding of their environment.

To bridge this gap, we propose a novel knowledge-driven framework for robotic OSL that enables LLMs to comprehend and navigate odor source localization tasks by incorporating human knowledge. By combining the semantic understanding and decision-making capabilities of LLMs with the sensory data from traditional OSL methods, robots could achieve higher levels of accuracy and efficiency in detecting and navigating toward odor sources. The integration of LLMs into robotic OSL could revolutionize the field by providing a more intuitive and flexible method for robots to understand and navigate their environment. This approach has the potential to lead to significant improvements in safety and rescue operations, pollution control, and environmental monitoring, addressing the urgent need for efficient, safe, and accurate localization of hazardous chemical gas leaks.

3. Methodology

3.1. Problem statement

The primary objective of Robotic OSL is to develop a mobile robotic system that can detect and navigate toward an unknown odor source within a specific environment. This task involves determining a sequence of actions that will effectively guide the robot to the odor source. Mathematically, we can represent this process as follows:

$$a = F(o, s) \quad (1)$$

where a denotes the robot's action, F represents the navigation algorithm, o is the past observation, and s refers to the current observation. The main challenge lies in finding the optimal function F that generates a sequence of actions a to navigate the robot toward the odor source successfully. This function must effectively leverage past experiences o and real-time sensory data s to make informed decisions that lead the robot to the odor source.

To address this challenge, we propose a knowledge-driven paradigm that leverages the capabilities of Large Language Models (LLMs) to enhance the robot's navigation and decision-making processes. Our approach, illustrated in Fig. 1, aims to utilize the generalization ability of LLMs, along with their contextual understanding of environmental dynamics, to develop a more robust and adaptable function F . This function integrates past experiences stored in memory with real-time sensory inputs to guide the robot in efficiently locating odor sources through advanced navigation strategies.

3.1.1. The search area and plume field

The search area is defined as a two-dimensional search space that contains an odor source, the location of which is unknown to the robot. This study focuses on navigating within this space to locate the unknown odor source. The odor plume is dispersed within this search area, creating a plume field characterized by varying concentrations

of odor. These concentrations are influenced by environmental factors such as wind and obstacles, making the search process more complex.

In this study, the plume field was generated by using a wind tunnel and a Particle Image Velocimetry (PIV) system, as illustrated in Fig. 2(a). The PIV system utilized green lasers to identify plume positions and velocities, enabling precise measurements of the odor plume distribution. These measurements are crucial for identifying the real-world plume distribution. The data collection process involved continuous emission of odor plumes (mineral oil) [89] into the wind tunnel through a nozzle, as shown in Fig. 2(b). The PIV system was employed to record plume positions and velocities in the downstream area, which had dimensions of $2.2 \times 5.0 \text{ m}^2$ ($x \times y$). The resolution of PIV measurements was 220 by 500, providing detailed spatial information about the plume dynamics [89]. For this study, the search area extends along the x -axis from 0 to 200 resolution units and the y -axis from -110 to $+110$ resolution units, corresponding to approximately $2.2 \times 2.0 \text{ m}^2$. The odor source is located at positions $x=2$ and $y=12$, as illustrated in Fig. 2(c).

To derive the odor concentration at each point in the plume, we used a mathematical model that calculates concentration based on the velocity data obtained from the Particle Image Velocimetry (PIV) system, Fig. 2(c). We calculated the concentration at each point in the plume field using the formula:

$$c = \sqrt{u^2 + v^2} \quad (2)$$

where u and v represent the velocity components along the x -axis and y -axis, respectively, and c represents the odor concentration at specific position. This approach assumes that higher velocities correspond to denser plume areas, thus indicating higher concentrations of the odor. This calculated concentration data and the velocity plots provided by the PIV form the basis for creating a detailed concentration map of the plume field. This map is crucial for simulating realistic odor dispersion patterns in our experimental setup.

The detailed setup and measurement process ensure high-quality data collection, which is essential for conducting realistic simulations and evaluating the effectiveness of the proposed OSL framework. This dataset is crucial for testing and validating the effectiveness of the OSL algorithms under real-world conditions, ensuring that the developed algorithms are robust and applicable in practical scenarios.

3.1.2. The robotic agent

The robotic agent is designed to detect chemical compounds in the air and efficiently navigate toward the odor source. The key sensors integrated into the robotic agent include chemical sensors for detecting the presence and concentration of odors and anemometers for measuring wind direction and speed. These sensors enable the robot to gather comprehensive environmental data, essential for effective navigation toward the odor source.

As detailed in Table 1, the robot has an original speed v , starting with a robot heading command of zero degrees. Only speed and heading commands are required to control a robot in a 2D plane. The robot employs an adaptive speed mechanism to calculate its speed command (v_c) that adjusts its speed based on the concentration of the detected odor. The adaptive speed mechanism adjusts the robot speed command using the following formula:

$$\beta = \max(0.1, 1 - c_{norm}) \quad (3)$$

$$v_c = v \times \beta,$$

where β is the speed factor, calculated based on the normalized odor concentration c_{norm} at the robot's position and eight adjacent points. v represents the robot's original speed, and v_c is the robot speed command. This adaptive mechanism ensures that the robot reduces its speed as it nears the odor source for better exploitation accuracy while preventing the speed from reaching zero to maintain constant movement.

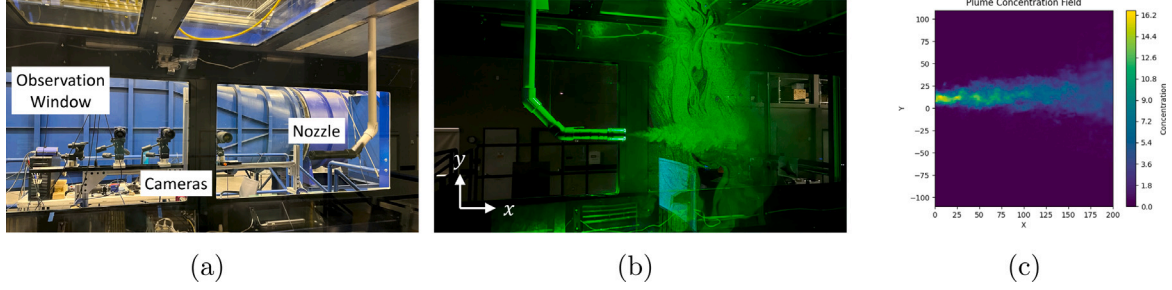


Fig. 2. (a) Experiment setup. (b) The nozzle continuously releases odor plumes into the wind tunnel. Green lasers help the PIV identify plume positions and velocities. (c) Diagram of the search area, illustrating the plume field and the locations of the odor source. The color scale ranges from deep purple (low concentration, 0) to bright yellow (high concentration, 16.4), depicting the intensity of the odor plume across the search area. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Definition of Parameters.

Parameters	Definitions
α	Wind Direction
β	Speed Factor
v	Original Speed
v_c	Robot Speed Command
ϕ_c	Robot Heading Command
c_{norm}	Normalized Concentration

We determine the robot's heading command (ϕ_c) by two search behaviors inspired by the mate-seeking behaviors of male moths, including surge and casting behaviors [90]. Surge behavior is activated when the moth is inside the plume, which commands the moth to move upwind; Casting behavior is activated when the moth moves out of the plume area, and the moth will move across the wind to increase the chance of re-detecting plumes. By iterating these two search behaviors, a male moth can find a female moth from a considerable distance.

In this work, we convert these search behaviors into two robot action commands, denoted as 0 and 1. Action 0 represents surge behavior, where the robot moves against the wind direction, and action 1 represents casting behavior, where the robot moves across the wind direction. During the search process, the LLM agent decides which action to perform based on the current sensor reading and past sensor observations. If the LLM agent selects action 0, it moves upwind by adding 180 degrees to the current wind direction, simulating the moth's direct approach toward the odor source. Conversely, if action 1 is selected, it moves across the wind by adding 90 degrees to the wind direction. Mathematically, we calculated the robot heading command as follows:

$$\phi_c = \begin{cases} \alpha + 180 & \text{if } action = 0; \\ \alpha + 90 & \text{if } action = 1, \end{cases} \quad (4)$$

This method ensures that the robot can adeptly adjust its trajectory in response to changes in wind direction, effectively using bio-inspired maneuvers to locate the odor source. It is important to note that the key difference between the moth-inspired method and our approach is that we use LLM to decide whether the robot should surge or cast, whereas the moth-inspired method makes decisions based on the current sensor reading. The LLM makes the decisions based not only on the current sensor reading but also on past sensory observations.

The robot updates its position once the heading command is determined based on the obtained heading command. As illustrated in Fig. 3, the robot operates within a global frame (xoy), with its localized adjustments made relative to its body frame ($x_r o y_r$). This updating process is determined mathematically in Eq. (5), which utilizes trigonometric functions to determine the robot's new position:

$$\begin{aligned} x^{k+1} &= x^k + (v_c^k \times \cos(\phi_c^k)) \\ y^{k+1} &= y^k + (v_c^k \times \sin(\phi_c^k)). \end{aligned} \quad (5)$$

Here, ϕ_c^k is the robot's heading command at the k th time step, calculated by Eq. (4). The variables (x^k, y^k) denote the robot's current coordinates at the k th time step within the plume field. The adaptive speed, v_c^k , is determined using Eq. (3) based on the odor concentration detected at the k th time step. After executing an action, the robot's updated coordinates (x^{k+1}, y^{k+1}) represent its new position at time step $k + 1$.

The primary goal of the robotic agent is to locate the odor source within the defined search area. The robot starts from a random initial position, denoted by (x^0, y^0) , and uses its sensors to detect and follow the odor plume. The robot's task involves making informed decisions about its movement direction and distance based on the odor concentration and wind direction at its current location. Upon detecting the odor concentration at its current position, the robot can choose between moving against the wind or across the wind direction. The chosen action and the wind direction determine the robot's heading command using Eq. (4).

3.2. Proposed knowledge-driven OSL framework

In this section, we present the knowledge-driven framework for robotic OSL, which integrates LLMs to enhance robots' decision-making and navigational capabilities. As depicted in Fig. 4, the framework comprises three interconnected modules: the Environment, the Reasoning module, and the Memory module.

The Environment module acts as the primary interface between the robot and its surroundings and captures real-time sensory data. The Reasoning module is the core of the decision-making process. It consists of several interdependent components that work together to process the sensory data and generate actionable commands for the robot. The Memory Module stores a diverse array of past experiences, which the Prompts Generator can recall when necessary. This module ensures that the decision-making process is not solely reliant on current sensory inputs but also benefits from past experiences, allowing the robot to make more informed and nuanced decisions. The following subsections explain the reasoning and memory modules in detail.

The reflection module is designed to evaluate and correct individual actions. For our robotic odor source localization (OSL) task, it is impossible to judge the correctness of each produced action since the robotic OSL task is a long horizon task, which means the final result (i.e., successful finding the odor source or fail finding the odor source) is determined by the accumulation of series of actions. The impact of single action is less significant to the final result. Thus, we remove the reflection module for the robotic OSL task to accelerate the decision-making process.

3.2.1. Reasoning module

The Reasoning module is the core of our knowledge-driven OSL framework. It uses the LLM to analyze sensory inputs, such as odor concentration and wind direction, to make informed navigation

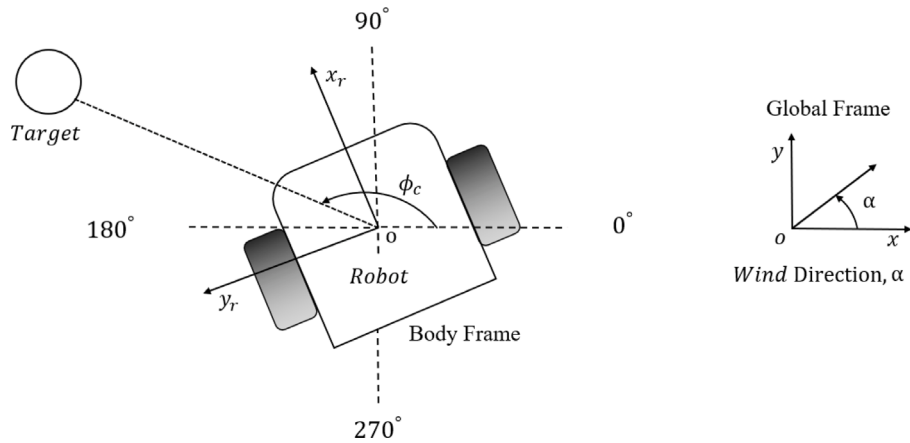


Fig. 3. Control of a ground mobile robot.

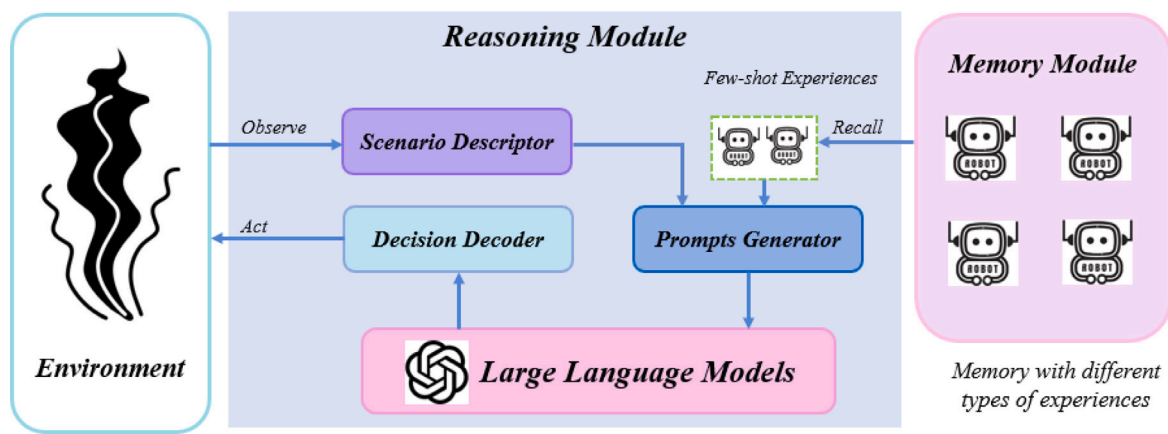


Fig. 4. The framework of our knowledge-driven robotic odor source localization system. It consists of three modules: Environment, Reasoning, and Memory. The Reasoning module processes sensory inputs, combines scenario descriptions with experiences from the Memory module to generate prompts, and interprets responses from the LLM to make navigation decisions.

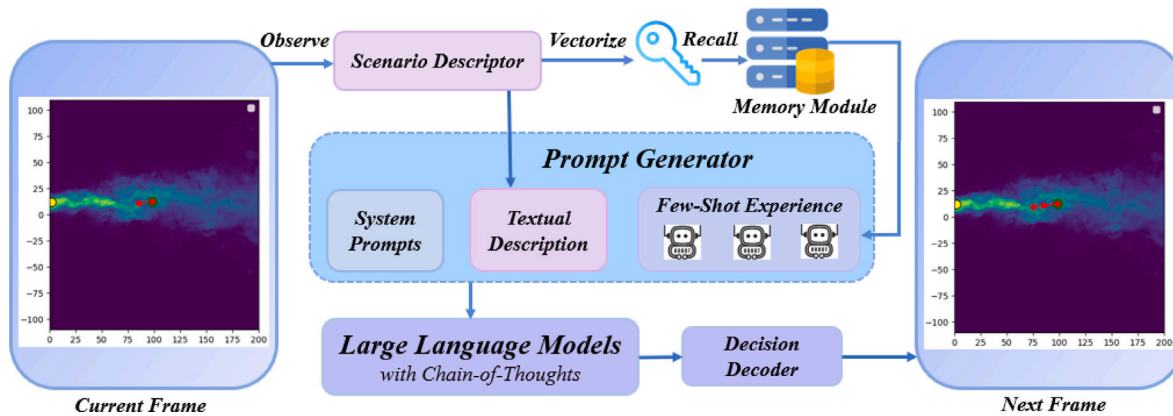


Fig. 5. Reasoning module for robotic odor source localization that utilizes the common-sense knowledge of the LLM and retrieves experiences from the Memory module to make informed decisions based on the observed scenario.

decisions. The reasoning agent can generate effective search strategies by combining current scenario descriptions with stored experiences. In the Reasoning module, we utilize experiences derived from the Memory module and the contextual understanding of the LLM to make decisions for the current odor localization scenario. The reasoning procedure is illustrated in Fig. 5, and includes the following steps: (1) scenario encoding; (2) experience retrieval; (3) prompt generation; (4) Prompt Processing; and (5) action decoding.

Scenario Encoding: The reasoning module begins by acquiring the sensor data of the current environmental scenario into a structured scenario descriptor, as illustrated in Fig. 6. By utilizing the natural language, the scenario descriptor transcribes these sensor data into descriptive text that describes the current scenario of the environment. This description contains detailed information about the Environment, such as the robot’s position and odor concentration within the search area. These descriptions are then fed into the prompt generator and

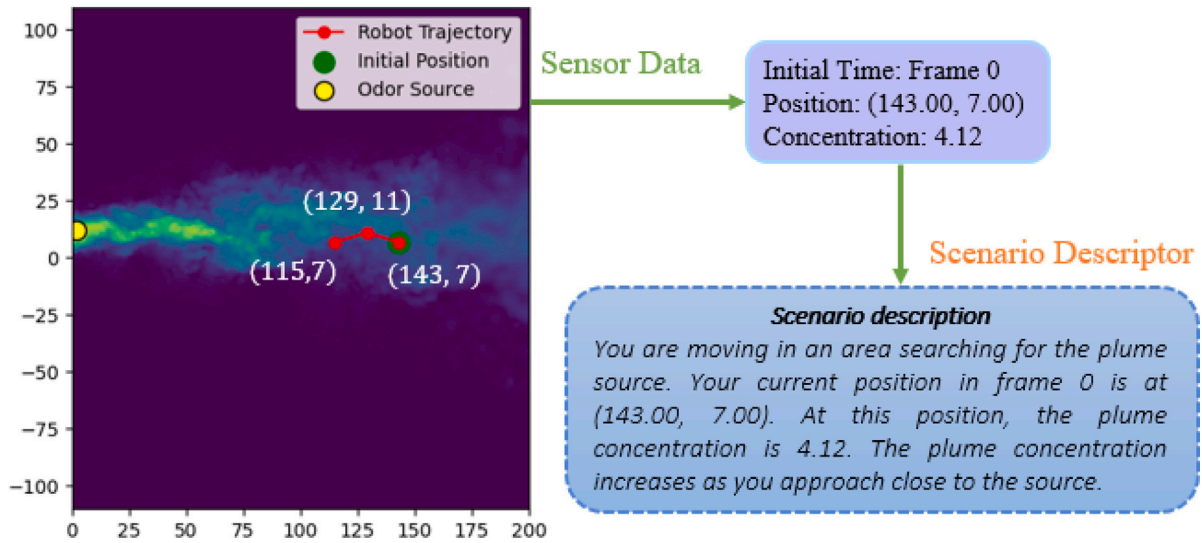


Fig. 6. Scenario Descriptor transcribes the current environmental scenario from the sensor data into descriptive text.

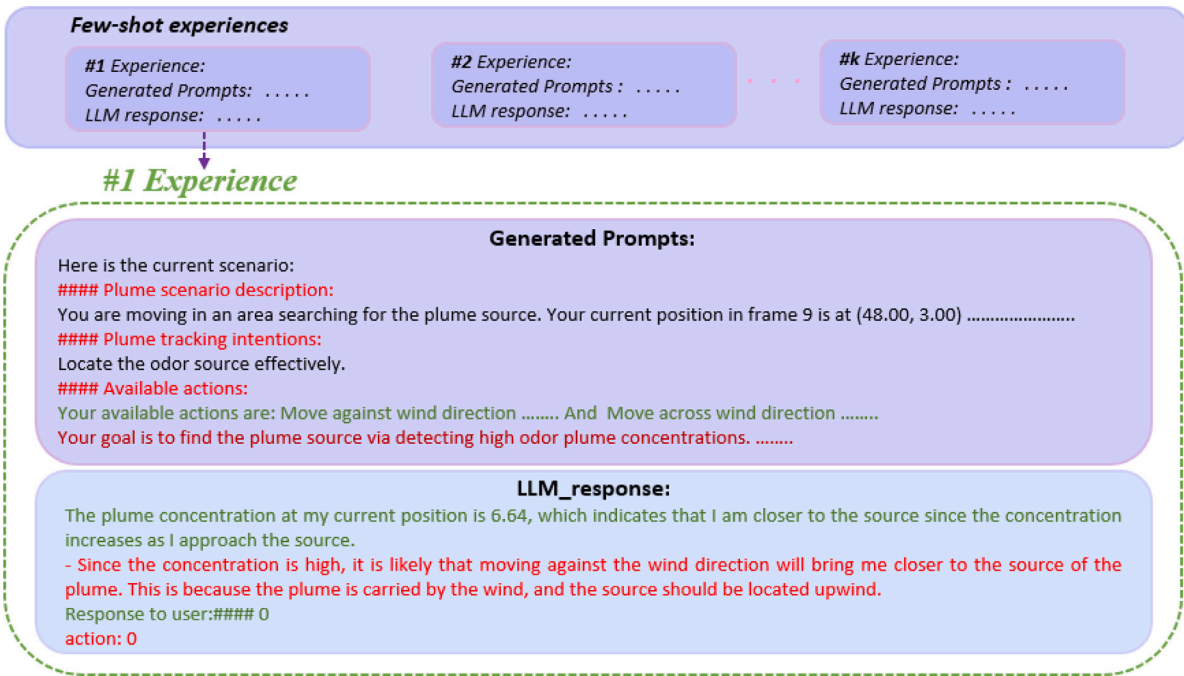


Fig. 7. Few-shot experiences where each experience consists of a human-LLM dialogue pair.

used as the keys to retrieve the relevant few-shot experiences from the Memory module.

Experience Retrieval: In this step, the current scenario is transformed into a vector representation using an embedding model. This vector, derived from a detailed natural language description of the current environment (including factors like robot position and odor concentration), serves as a query to access our Memory module. By leveraging cosine similarity [91], this query vector is compared against the embeddings of past scenarios stored in the database. The system retrieves the top k most similar entries, representing past experiences with contexts closely related to the current scenario. These entries, or “few-shot experiences”, as illustrated in Fig. 7, are then integrated with the present scenario description to assist in the reasoning process. This approach ensures that the robot’s decision-making is informed by historical data, improving its responses to new yet similar conditions.

Prompt Generation: This stage involves constructing comprehensive prompts that are pivotal for the LLM’s reasoning process. The

prompts are generated by combining system prompts, textual scene descriptions, and relevant past experiences drawn from few-shot experiences. The system prompts summarize the task, detailing the expected inputs and outputs and specific constraints governing the reasoning process. These prompts, illustrated in Fig. 8, tailored specifically for each decision-making instance, capture the specific details of the current situation, enabling the LLM to apply its reasoning capabilities effectively to deduce the most appropriate navigational actions.

Prompt Processing: The complexity and variability of odor localization tasks demand a detailed reasoning process for accurate decision-making. To address this, we utilize Chain-of-Thought (CoT) prompting techniques [70], which guide the LLM to articulate its reasoning step-by-step, helping to clarify the decision-making path and reduce potential inaccuracies. By structuring the LLM’s responses in this manner, we ensure that each decision is based on a logical progression of thought appropriate for the variable and complex scenarios encountered in odor localization tasks.

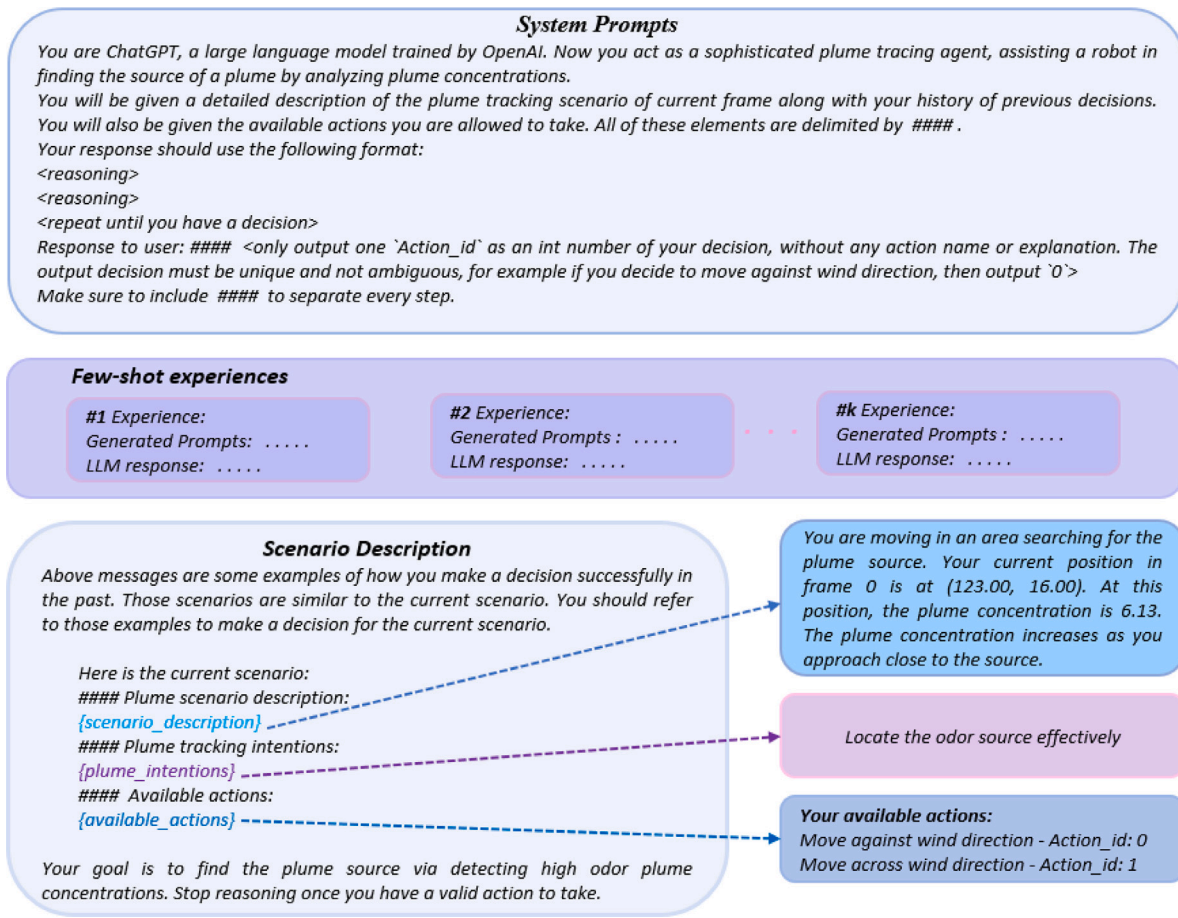


Fig. 8. Prompts generator consists of system prompts, textual description, and few-shot experiences.

Action Decoding: The output from the LLM is interpreted by the action decoder within the reasoning module, translating the model’s responses into specific actions for the robotic agent. The decoder is critical in converting high-level decision outcomes into specific navigational commands that the robot executes, facilitating interaction with the Environment. It creates a robust closed-loop system that refines the robot’s navigational abilities through a cycle of continuous feedback and learning, leveraging both new experiences and past experiences stored in the Memory module. Fig. 9 illustrates the functionality of the reasoning module, showing how environmental data is transformed into a decision-making prompt for the LLM, and subsequently, how the LLM’s output is decoded into actions for the robot. Detailed information on the construction and design of these prompts can be found in Appendices A.1 and A.2.

3.2.2. Memory module

The memory module of our knowledge-driven OSL framework is crucial for the robotic agent’s ability to reason effectively in complex odor source localization tasks. It archives extensive records of past scenarios, including scene descriptions and corresponding reasoning processes, which are instrumental in informed decision-making in new situations. When the robotic agent faces a new scenario, it retrieves relevant past experiences to aid decision-making. This retrieval process involves transforming the current scenario description into a vector. This vector acts as a key, enabling the agent to search through the memory module for scenarios with similar conditions and their corresponding successful strategies.

To optimize this retrieval process, our framework employs pre-filtering technique based on cosine similarity. While providing as many recordings as possible directly in the LLM prompt and leaving the LLM

to select the best n candidates may allow the LLM to make much more nuanced choices, adopting this in our framework could introduce significant challenges for the decision-making process of LLM agent, discussed below:

- **Real-time Decision-Making:** Adding an extra step for the LLM to dynamically select relevant memories would increase computational overhead, slow down the decision-making process, and compromise performance in time-sensitive scenarios. The proposed pre-filtering technique significantly enhances performance by reducing the complexity of input processing, thereby minimizing inference time.
- **Finite Token Limits:** LLMs have a finite token limit, which poses significant challenges to process extensive contexts. Providing all memory recordings directly to the LLM could exceed these limits as the number of experiences grows, which can lead to the truncation of valuable data. By pre-filtering using cosine similarity, we ensure that only the most relevant and manageable subset of experiences is included within the token limit, thereby preserving critical information.
- **Scaling Memory Module:** Due to token limits and computational overhead, LLMs face significant challenges in managing large numbers of experiences. As the memory module grows, providing all recordings directly to the LLM becomes increasingly unfeasible. A pre-filtering mechanism is essential to ensure scalability while maintaining decision quality. This mechanism allows LLMs to efficiently manage and retrieve relevant information without overwhelming computational resources.

By utilizing pre-filtered past experiences, the robot applies correct strategies to current environmental scenarios, ensuring a higher like-

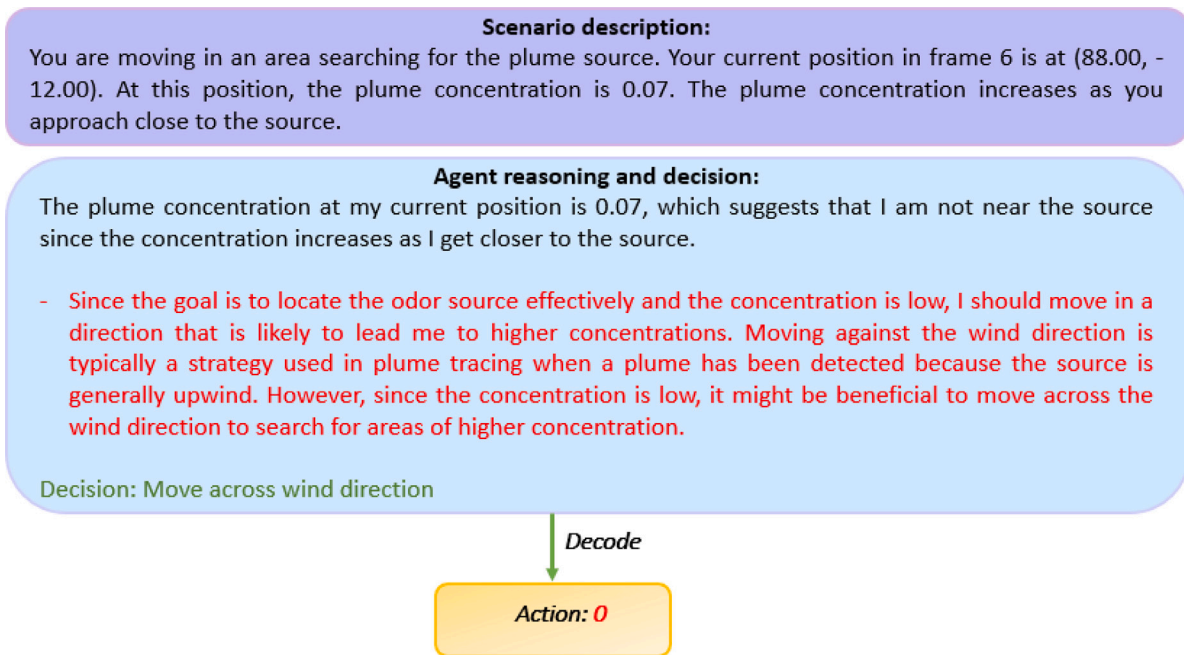


Fig. 9. The scenario description provided by the scenario descriptor and the decision decoder determines the action based on the LLM's reasoning output for robotic odor source localization.

likelihood of successful navigation. This method of leveraging past experiences enables the robot to maintain consistent and reliable decision-making, enhancing its operational efficiency without constant updates to the memory database. This static approach avoids the complexities and potential errors associated with continuous learning models, focusing instead on applying well-established knowledge to new situations.

4. Experiments and results

This section presents our study's experimental setup and results on robotic odor source localization (OSL). We designed the experiments to evaluate the performance of the proposed knowledge-driven OSL framework under different configurations and conditions. We conducted an ablation study, comparing results across various scenarios, including different odor source locations and varying levels of memory utilization. Additionally, we compared our approach to a reinforcement learning method using Deep Q-Networks (DQN).

4.1. Experimental setup

We evaluated the performance of our knowledge-driven OSL framework by conducting experiments in a simulated environment, described in Section 3.1.1, which incorporates real-world plume data. We designed the environment to replicate the plume behavior observed in the wind tunnel experiments, ensuring that the search area and odor dispersion closely mimic actual conditions.

We conducted our experiments using the search area configuration detailed in Section 3.1.1, following the setup of our simulated environment. Initially, we used the original odor source location at a specific point within the search area, where the nozzle continuously released odor plumes into the wind tunnel. To further assess the system's robustness, we tested the framework by flipping the odor source location along the x -axis, as shown in Fig. 10. For each experimental setting, we conducted 10 trials to capture the variability and ensure the robustness of our results. The performance metrics for evaluation included Success Rate (SR), Averaged Travel Distance (TD), and Averaged Search Time (ST). The success rate measures the proportion of trials where the robot successfully located the odor source. We define the Averaged Travel

Distance and the Averaged Search Time as the mean values of total distance traveled and time taken across the trials.

4.2. LLM-based OSL agent configuration

We tested the robotic agent's navigation strategy under three different configurations: (i) Adaptive-Hint, where the robot adjusts its speed based on odor concentration and receives directional hints; (ii) Adaptive-No Hint, where the robot adjusts its speed but does not receive any hints; and (iii) Hint Only, where the robot receives hints about movement direction but does not adjust its speed. To assess the impact of prior experiences on performance, we varied memory utilization across three levels: 0-shot, 3-shot, and 5-shot experiences where these experience levels refer to the number of relevant past scenarios retrieved from the memory module to inform the robot's decision-making process. In the 0-shot experience, the robot makes decisions based solely on the current scenario without using any past experiences. In the 3-shot experience, the robot retrieves three past experiences to guide its current decision, while in the 5-shot experience, the robot retrieves five, providing more context to inform its actions.

4.3. DQN-based OSL agent

In our study, to benchmark the performance of our knowledge-driven OSL framework, we implemented a Deep Q-Network (DQN) model based on the architecture and methodology outlined in [37]. The DQN maps the robot's sensor input states to output actions that guide the robot's navigation toward the odor source.

We chose DQN, a classic deep reinforcement learning algorithm with a discrete action space, as a baseline to compare our framework's performance with a learning-based algorithm. The purpose of this comparison is not to design a state-of-the-art reinforcement learning agent but to highlight the generalization capability of our proposed method. The LLM in our framework also outputs discrete actions, including moving against the wind or crosswind. Therefore, for a meaningful comparison, we selected DQN as it aligns with the discrete action space of the LLM-driven framework, which also outputs discrete actions.

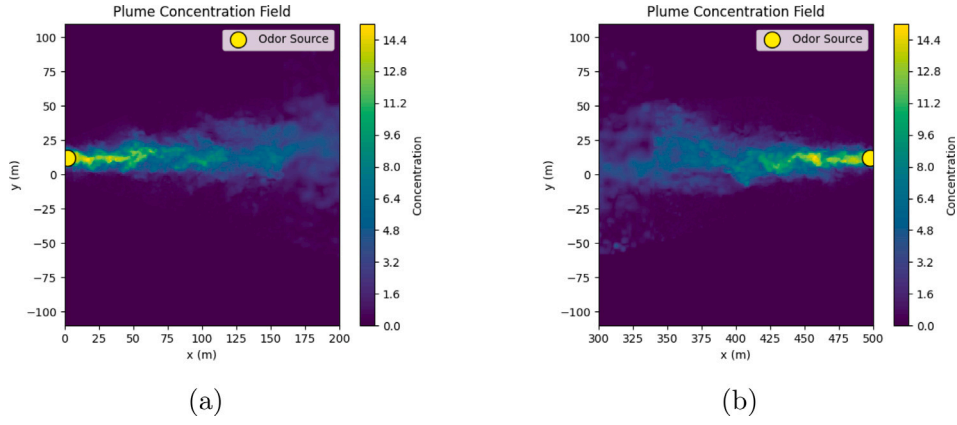


Fig. 10. (a) Diagram of the search area, illustrating the plume field and the original location of the odor source. (b) Diagram of the search area after flipping the search area and odor source location along the x -axis.

The DQN implemented in our project is a fully connected neural network designed to process input states—comprising the robot’s position, velocity components, and concentration of the odor—and output actions that determine the robot’s heading direction. The network architecture, illustrated in Fig. 11, consists of the following:

Input Layer: The input to the DQN is a state vector that includes the robot’s current x and y positions, the velocity components (u and v), and the odor concentration at the robot’s location. This state vector encapsulates the essential environmental information needed for making navigation decisions.

Hidden Layers: The network includes two fully connected layers with 128 neurons each, using ReLU (Rectified Linear Unit) activation functions to introduce non-linearity, allowing the model to capture complex relationships between the input state and the optimal action.

Output Layer: This layer comprises four neurons, each representing the Q-value of one of four possible actions: moving left, down, right, or up. The action with the highest Q-value is selected as the robot’s next move.

Training DQN Network: We trained the DQN in the same simulated environment used to evaluate our knowledge-driven OSL framework, ensuring a consistent comparison. This environment replicates the plume behavior observed in wind tunnel experiments, with the search area defined as a 2.2×2.0 meter space. Initially, we fixed the odor source at a specific location and later flipped its position along the x -axis to test the model’s robustness. During each training step, the DQN receives a state vector that includes the robot’s position (x , y), velocity components (u , v), and odor concentration. Based on this input, the network generates Q-values for the four possible actions: moving left, down, right, or up. The robot then selects and executes the action with the highest Q-value.

We designed the reward system to encourage the robot to locate the odor source efficiently. Mathematically, we define the reward r as follows:

$$r = \begin{cases} 100 & \text{if } d < \delta; \\ -100 & \text{if } OB \text{ or } t < t_{max}; \\ -0.1 & \text{if } dist(P_4, P_1) < 3; \\ c_{norm} & \text{if } c_{norm} > 0 \text{ and } dist(P_4, P_1) \geq 3; \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where d is the distance of the robot agent to the odor source, δ is the distance threshold to consider the goal reached, OB indicates that the robot moves out of bounds, t is the current time step, t_{max} is the maximum allowed time steps, P_1 and P_4 are the robot’s positions at the first and fourth most recent steps, $dist(P_4, P_1)$ represents the Euclidean distance between the positions P_1 and P_4 , c_{norm} is the normalized odor concentration at the robot position.

The robot receives a high positive reward, $r = 100$ if it successfully reaches the odor source, and a negative reward of $r = -100$ if the robot either moves outside the designated search area (i.e., moves beyond the boundaries of the environment) or exceeds the maximum allowed time for the search. If the robot’s movement over the last four steps is minimal, indicating a lack of significant progress (i.e., it has not moved much from its position four steps ago), we apply a small penalty of $r = -0.1$ to discourage it from staying stationary. When the robot detects an odor concentration and has moved significantly, we reward it based on the concentration at its current position, encouraging it to move toward areas with higher odor concentrations. If none of these conditions are met, the reward defaults to $r = 0$, indicating no progress or penalty. The goal is to maximize cumulative future rewards by minimizing the distance to the odor source and reducing travel time.

We conducted training over multiple episodes, each representing a complete search sequence starting from a random initial position until the robot either finds the odor source or reaches the episode’s time or boundary limits. We trained the DQN with 1,000, 3,000, and 5,000 episodes using an epsilon-greedy policy, where the robot initially explores randomly (high epsilon) and gradually shifts to exploiting the learned policy (epsilon decreases). We set the discount factor for future rewards to 0.99, ensuring that the robot values immediate rewards while considering long-term benefits. We evaluated the DQN’s performance using the same metrics as in our knowledge-driven OSL framework: Success Rate (SR), Averaged Travel Distance (TD), and Averaged Search Time (ST). These metrics comprehensively assess the DQN’s ability to navigate the robot to the odor source.

4.4. Sample trials

To illustrate the performance of our knowledge-driven OSL framework under various memory settings, we demonstrated sample trials using 0-shot, 3-shot, and 5-shot memory configurations. We design the trials to illustrate the impact of different levels of memory integration on the robot’s ability to navigate the plume field and accurately locate the odor source.

The LLM agent can choose one of two actions: moving cross wind or moving against the wind. When the action is crosswind, the robot moves in a direction that is 90 degrees to the wind direction. In Fig. 12, the robot initially performs crosswind movements as no plume concentration is detected. When the robot still fails to detect any plume in subsequent positions, it continues to choose the crosswind action. This results in back-and-forth movements, which are a natural part of the exploratory process during crosswind actions, allowing the robot to search for the plume in a broad area. When the robot detects a plume concentration that is larger than the threshold, the LLM agent switches to the against-wind action, guiding the robot upwind to follow

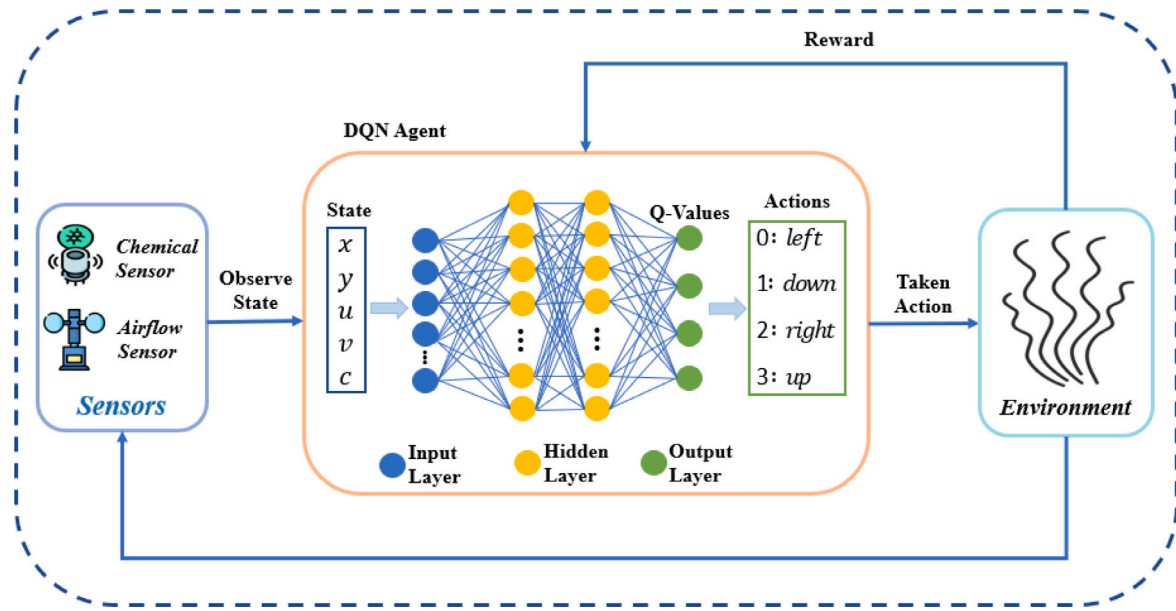


Fig. 11. The Deep Q-Network (DQN) architecture for robotic odor source localization. The framework consists of sensors, a DQN agent, and an environment.

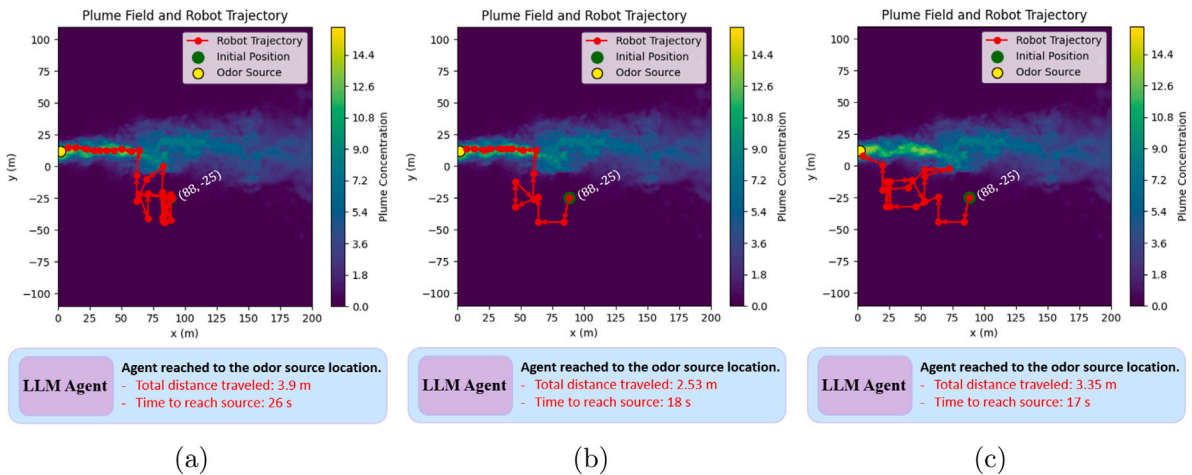


Fig. 12. Sample trials of the knowledge-driven OSL framework for past experience: (a) 0-shot memory, (b) 3-shot memory, and (c) 5-shot memory.

the plume. The observed redundancy in the trajectory is expected and reasonable for this exploratory strategy. If the plume had been detected during the initial crosswind movements, the robot would have immediately switched to the upwind (surge) behavior, avoiding repeated movements. The behavior reflects the LLM’s decision-making process and the inherent challenges of locating odor sources in dynamic environments.

In the 0-shot trial (Fig. 12(a)), the robot operates without leveraging prior experiences, relying solely on the LLM’s real-time reasoning to navigate toward the odor source. The trajectory shows the robot navigating toward the odor source with some exploration due to the lack of memory. Despite this, the robot successfully locates the odor source, demonstrating the inherent capability of the LLM to interpret and act on sensory inputs effectively. The 3-shot trial (Fig. 12(b)) exhibits enhanced navigation efficiency as the robotic agent utilizes memory recall to retrieve three similar past experiences from the memory module. The trajectory is more direct, with fewer deviations compared to the 0-shot trial, indicating the benefits of incorporating relevant past experiences into the decision-making process. This results in quicker and more accurate localization of the odor source. Conversely, the 5-shot trial (Fig. 12(c)) incorporates five past experiences, providing a

richer historical context for the decision-making process. While this setup allows for a highly informed approach, it also introduces the risk of over-reliance on historical data, occasionally leading to suboptimal paths that may not perfectly align with the current environmental setup. It highlights the potential drawbacks of excessive memory usage, where too much reliance on past data can reduce the adaptability and efficiency of navigation.

Overall, these sample trials provide valuable insights into how different levels of memory integration affect the performance of robotic odor source localization. They illustrate a clear trade-off between memory use and navigational efficiency, emphasizing the importance of finding an optimal balance to maximize the effectiveness of the knowledge-driven OSL framework in dynamic and varied environments.

4.5. Ablation study

The ablation study aimed to rigorously evaluate the impact of different memory utilization levels and configuration settings on the performance of our knowledge-driven OSL framework. This study provides key insights into how varying the integration of past experiences

Table 2
Performance metrics of different levels of memory utilization for OSL.

Memory	Success Rate (SR) ↑	Averaged Travel Distance (TD) ↓	Averaged Search Time (ST) ↓
0-shot	8/10	2.35	15.5
3-shot	10/10	2.19	14.5
5-shot	9/10	2.20	14.7

Table 3
Performance metrics of different settings of configurations for OSL.

Configuration	Success Rate (SR) ↑	Averaged Travel Distance (TD) ↓	Averaged Search Time (ST) ↓
Adaptive-Hint	10/10	1.93	13.5
Adaptive-No Hint	8/10	2.01	13.2
Hint Only	4/10	1.44	6.8

and adaptive strategies influences the robot's ability to locate odor sources effectively.

Table 2 presents the performance metrics across different levels of memory utilization within the OSL framework: 0-shot, 3-shot, and 5-shot. The 3-shot memory setting demonstrated the best overall performance, achieving a high success rate (SR) of 10/10, along with the lowest averaged travel distance (TD) and search time (ST). This suggests that utilizing a moderate amount of past experiences (3-shot) optimally enhances the robot's efficiency in navigating and locating the odor source. In contrast, the 0-shot setting, which depends solely on real-time reasoning without memory recall, resulted in lower success rates and longer travel distances. The 5-shot setting, while slightly better than 0-shot, did not outperform the 3-shot configuration, indicating that over-reliance on historical data does not necessarily lead to improved performance.

Our results show that, the 3-shot configuration perform optimally in our OSL framework as it retrieves three highly relevant examples with a higher similarity threshold, maintaining a concise focus on the most aligned scenarios. This setup enhances the LLM's reasoning process by providing precise, contextually relevant guidance.

In contrast, the 0-shot configuration lacks contextual memory, relying solely on generalized reasoning, which often leads to suboptimal navigation and reduced success rates. On the other hand, the 5-shot configuration introduces additional diversity by retrieving two more past experiences. While this can offer a broader historical context, it also raises the likelihood of including less relevant or conflicting scenarios, especially in environments with limited variability like ours. This increased diversity sometimes leads to over-reliance on historical data, which can produce suboptimal decision-making and less efficient navigation paths. Furthermore, the 5-shot configuration faces several other challenges that impact the LLM's reasoning process. One such challenge is scenario redundancy, which occurs when additional memories overlap with existing ones, contributing limited new insights and introducing unnecessary complexity to the LLM's reasoning process. Additionally, including more memories in low-variability environments increases computational and decision-making burden, which overwhelms the LLM and reduces its ability to generalize effectively to the current scenario. Lastly, using a larger set of examples increases the risk of overfitting, where the reasoning process becomes too focused on specific patterns from the retrieved data, which makes it less adaptable to new or varied scenarios.

Table 3 summarizes the performance metrics for different configuration settings: Adaptive-Hint, Adaptive-No Hint, and Hint Only, all utilizing the 3-shot memory setting, which showed the best performance. Among these, the Adaptive-Hint configuration demonstrated superior performance, achieving the highest success rate (SR) of 10/10 and the most efficient travel distance (TD) and search time (ST). This configuration adjusts the robot's speed based on the detected odor

Table 4
Performance metrics of different OSL algorithms for the original odor source location.

OSL Algorithms	Training episodes	Success Rate (SR) ↑	Averaged Travel Distance (TD) ↓	Averaged Search Time (ST) ↓
(LLM) Adaptive-Hint (3-shot)	–	10/10	1.93	13.5
Moth-inspired method [92]	–	7/10	2.28	16.2
DQN [37]	1000	8/10	1.45	13.1
	3000	9/10	1.91	16.3
	5000	9/10	3.5	22.8

concentration (adaptive speed) and provides hints or guidance on the robot's movement direction based on this concentration data (hints about the robot's movement). This combined approach of adapting speed and using directional hints proved to be the most effective in guiding the robot to locate the odor source efficiently. In contrast, the Adaptive-No Hint configuration showed a slightly lower success rate and efficiency, indicating that the absence of hints affects the robot's ability to navigate optimally. The Hint Only configuration, which relied solely on directional hints without adaptive speed adjustments, had the lowest success rate and performance metrics. This underscores the importance of combining adaptive strategies with memory utilization to enhance the robot's navigation capabilities.

The results from the ablation study reveal that the optimal approach for robotic odor source localization involves a moderate level of memory utilization (3-shot) combined with adaptive strategies (Adaptive-Hint). This balance maximizes the robot's ability to accurately and efficiently locate odor sources by effectively combining past experiences with real-time environmental feedback.

4.6. Comparative analysis with DQN and moth-inspired method

In this section, we compared the performance of our knowledge-driven OSL framework with a DQN based approach [37] and a Moth-inspired method [92]. The comparison focuses on key performance metrics, including success rate (SR), averaged travel distance (TD), and averaged search time (ST), across both original and flipped odor source locations.

Table 4 presents the performance metrics for the original odor source location. The knowledge-driven OSL framework with the Adaptive-Hint (3-shot) configuration achieved the highest success rate (10/10). It demonstrated the most efficient navigation, with the shortest averaged travel distance (1.93) and averaged search time (13.5). This performance underscores the effectiveness of combining memory recall with adaptive strategies in the knowledge-driven approach. In contrast, the DQN approach, despite showing competitive performance in some metrics, exhibited inconsistencies. While the DQN trained with 3,000 episodes achieved a success rate of 9/10, its performance in terms of averaged travel distance (1.91) and averaged search time (16.3) was less efficient than that of the knowledge-driven approach. Additionally, as the training episodes increased to 5,000, the DQN's performance declined, indicating the model's limitations in maintaining consistent efficiency and accuracy across varying conditions. While showing reasonable performance with a success rate of 7/10, the Moth-inspired method fell short compared to the knowledge-driven and DQN approaches, particularly in terms of average travel distance and search time. It suggests that while biologically inspired methods can be effective, they may not fully capture the complexities of navigating varied and dynamic environments.

Table 5 summarizes the performance metrics after flipping the odor source location along the x -axis. The knowledge-driven OSL framework, particularly the Adaptive-Hint (3-shot) configuration, continued

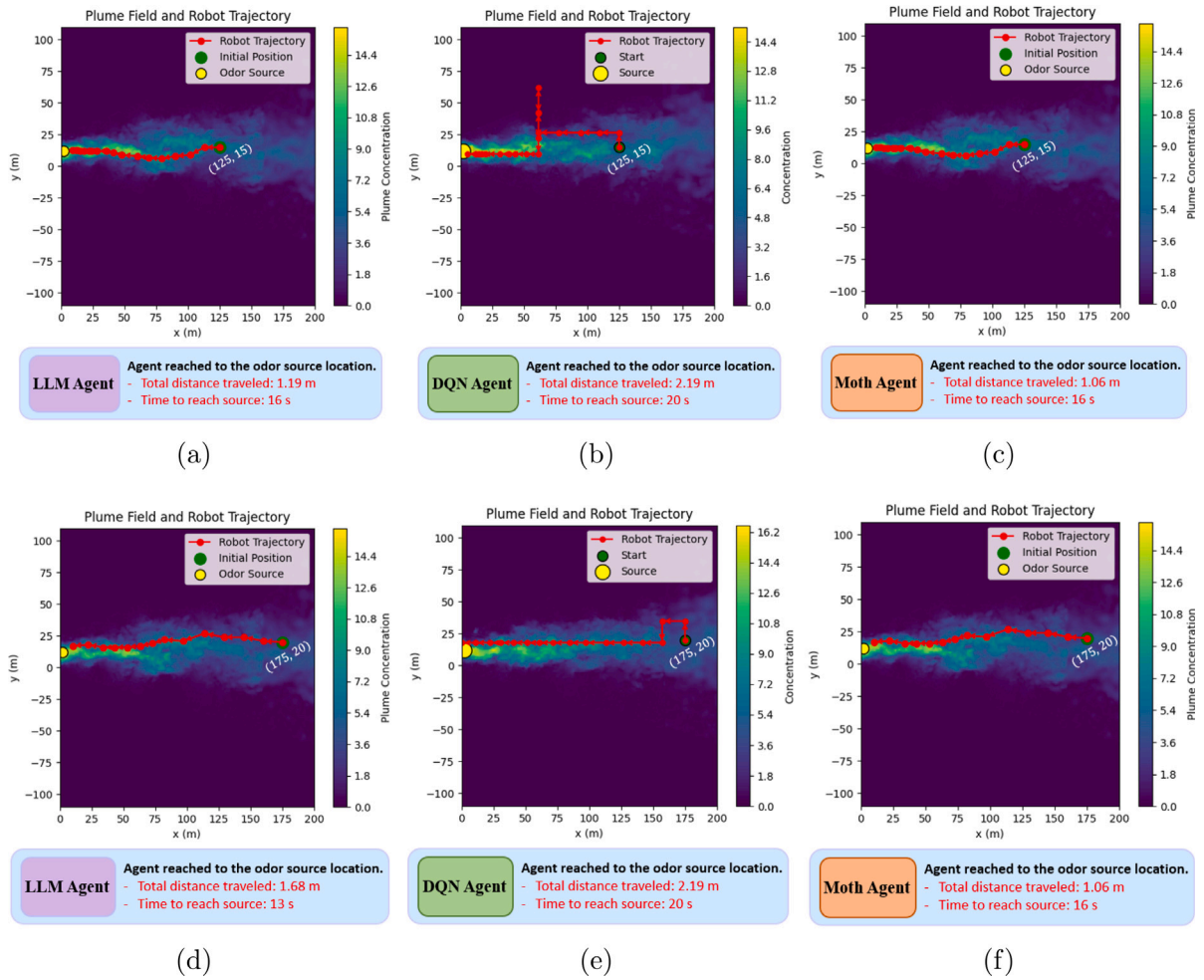


Fig. 13. Examples of robotic agent navigation to the odor source location. (a) and (d) show that the robotic agent reaches the odor source location using the LLM's reasoning, demonstrating efficient navigation and effective decision-making, (b) and (e) show that the robotic agent reaching the odor source location using the DQN approach, which, while sometimes successful, often results in less efficient paths and higher search times, and (c) and (f) show that the robotic agent reaching the odor source location using the Moth-inspired method, demonstrating efficient navigation in which agent is in the plume.

to show strong performance with an 8/10 success rate. It highlights the framework's adaptability to changes in environmental conditions. In contrast, the DQN approach completely failed, with a 0/10 success rate across all training levels, underscoring its inability to generalize to new, unseen conditions. While maintaining a 7/10 success rate, the Moth-inspired method again showed less efficiency in terms of travel distance and search time compared to the knowledge-driven approach. These results indicate that the DQN and Moth-inspired methods lack the robustness and adaptability required for effectively handling dynamic and altered environments, as evidenced by the significant drop in performance when the odor source location was flipped.

Figs. 13 and 14 visually depict the trajectories of the robotic agents using the different approaches for both the original and flipped odor source locations. For the original odor source location, the knowledge-driven agent (Figs. 13(a) and 13(d)) demonstrated efficient and direct navigation to the odor source, showcasing its effective decision-making and adaptability. On the other hand, the DQN agent (Figs. 13(b) and 13(e)) exhibited less consistent and often less direct paths, with a notable decline in performance when training episodes were increased, highlighting the limitations in its generalization capabilities. The Moth-inspired method (Figs. 13(c) and 13(f)) showed a consistent trajectory but was less efficient overall compared to the knowledge-driven approach.

In the flipped odor source scenario (Fig. 14), the knowledge-driven agent (Fig. 14(a)) continued to perform effectively, maintaining adaptability to the new environmental conditions. In contrast, the DQN agent

Table 5

Performance metrics of different OSL algorithms for the flipped odor source location.

OSL Algorithms	Training Episodes	Success Rate (SR) ↑	Averaged Travel Distance (TD) ↓	Averaged Search Time (ST) ↓
(LLM) Adaptive-Hint (0-shot)	–	7/10	2.53	15.7
(LLM) Adaptive-Hint (3-shot)	–	8/10	2.65	16.2
(LLM) Adaptive-No Hint (3-shot)	–	4/10	3.28	19.7
Moth-inspired method [92]	–	7/10	3.85	23.8
DQN [37]	1000	0/10	51.8	99
	3000	0/10	46.12	99
	5000	0/10	56.68	99

failed to locate the odor source entirely (Fig. 14(b)), reinforcing its struggle with generalization. The Moth-inspired method (Fig. 14(c)), although maintaining some success, again demonstrated inefficiency in navigating the altered environment, with the robot often taking longer paths to the odor source.

The comparative analysis clearly demonstrates the superior performance of the knowledge-driven OSL framework over the DQN and

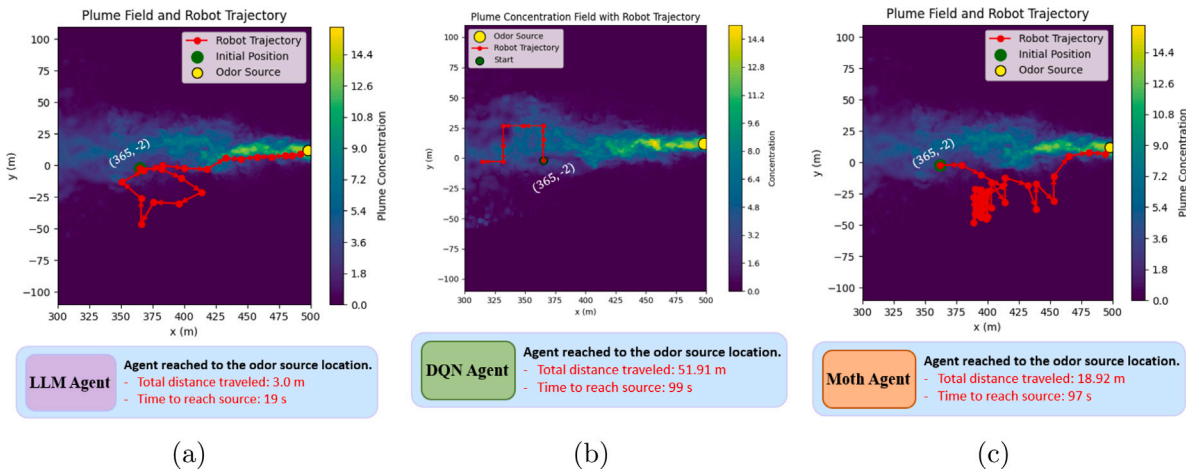


Fig. 14. a) Robotic agent reached to the odor source location using LLM’s reasoning: 3-shot memory, (b) shows that DQN agent failed to reach the odor source location where the odor source location is flipped along the x-axis, and (c) show that the robotic agent reaching the odor source location using the Moth-inspired method, often results in less efficient paths and higher search times.

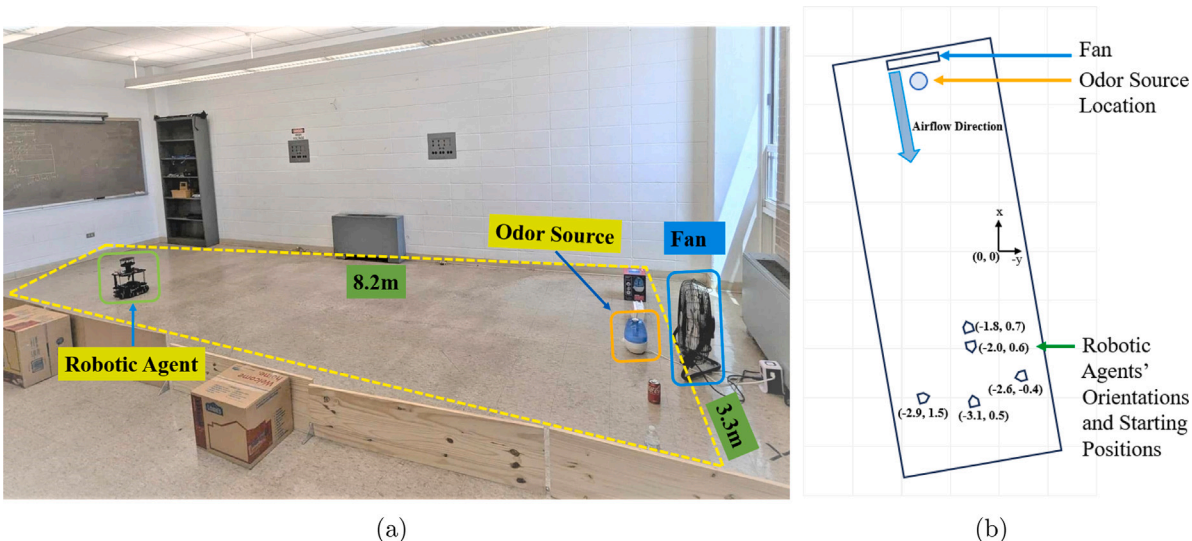


Fig. 15. a) An experimental setup shows the robotic agent in the search area, with a humidifier releasing ethanol vapor as the odor source and a fan to establish airflow. The agent starts from a downwind location, navigating toward the ethanol source. (b) A schematic representation of the search area with a fan setup creates a uniform airflow. The diagram includes the odor source location and the initial robot positions, illustrating the experimental conditions for OSL tests.

Moth-inspired methods. Integrating large language models (LLMs) with memory-assisted decision-making enhances the robot’s ability to locate odor sources efficiently and provides robust adaptability across varied environmental scenarios. The DQN approach, while effective under certain conditions, needs to improve with generalization, particularly in dynamic or altered environments, making it less reliable for real-world applications. The Moth-inspired method, although consistent, lacks the efficiency and adaptability needed for optimal performance. Overall, the knowledge-driven framework stands out as the most effective solution for robotic odor source localization in complex and dynamic environments.

4.7. Real world experiment results of OSL

4.7.1. Search area

In the real-world experimental setup of our odor source localization (OSL) study, we defined the search area to test the robot’s navigation and detection capabilities in a controlled environment. The search area covered a two-dimensional space of 8.2 meters by 3.3 m, as shown in Fig. 15(a). Ethanol vapor, chosen for its non-toxic properties

and frequent use in OSL research [93], served as the odor source. We employed a humidifier as the plume generator to disperse the ethanol vapor throughout the experiments. To ensure a uniform airflow direction across the search area, we placed a single fan behind the odor source, facilitating the diffusion of the plume, as presented in Fig. 15(b).

4.7.2. Robotic agent specifications

The Turtlebot3 mobile robot platform was utilized in our experiments, equipped with built-in sensors and components tailored for navigation and sensory tasks in odor source localization. The robot’s sensor suite includes a 360-degree LDS-02 Laser Distance Sensor for measuring the distance, a WindSonic Anemometer for measuring wind speed and direction, and an MQ3 alcohol sensor to detect chemical plume concentrations. Fig. 16 shows the robotic agent used in the experiments.

Powered by a Raspberry Pi 4 CPU, the Turtlebot3 runs on Ubuntu 20.04 with the Robot Operating System (ROS) Noetic, allowing for seamless integration and real-time communication. The robot operates in conjunction with a remote personal computer (PC) via a local area

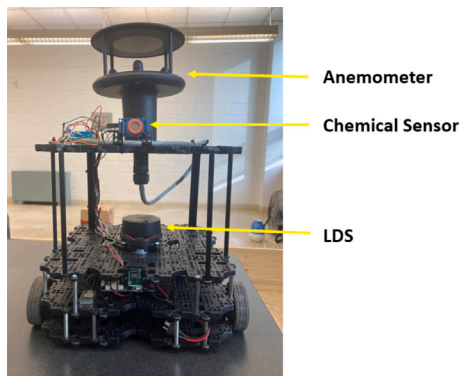


Fig. 16. Robotic agent used in the experiments, equipped with various sensors including a chemical sensor and an anemometer to measure odor concentrations and airflow, respectively.

network, which facilitates the control and data exchange necessary for effective robotic navigation and data processing in complex odor localization scenarios. This setup ensures robust performance in dynamic environments, enabling precise control and responsiveness during the localization tasks.

4.7.3. OSL experimental design

We designed the experiments to evaluate the effectiveness of our proposed Knowledge-driven OSL framework, which leverages Large Language Models (LLMs) in a real-world setup. Additionally, we assessed the performance of robotic OSL using a Deep Q-Network (DQN) model based on Reinforcement Learning (RL). Both approaches were tested under controlled conditions to determine their performance and adaptability in real-world scenarios.

We incorporated a dynamic memory setting in the knowledge-driven framework to enhance decision-making across various environmental conditions. We tested three configurations: Adaptive-Hint, Adaptive-No Hint, and Hint Only, with memory settings varied across 0-shot, 3-shot, and 5-shot conditions to assess the impact of accumulated knowledge on performance. Each configuration was designed to evaluate the framework's efficiency and success rate, with detailed specifics provided in Section 4.1. Trials were conducted under standardized conditions, with success defined by the robot's proximity to the odor source—a trial was deemed successful if the robot reached within 0.6 meters of the target location. If the robot failed to locate the odor source within 120 s, the trial was considered a failure. The robotic agent's control parameters were set to ensure optimal performance during operation. The initial linear velocity (l_v) and angular velocity (ω_c) were set at 0.1 m/s and 0.3 rad/s, respectively, for the Knowledge-driven OSL framework. To improve navigation toward the odor source, the robot employed an adaptive speed mechanism, adjusting its linear velocity based on the concentration of the detected odor. The adaptive speed was determined using the following formula:

$$l_v = \begin{cases} 0.1 & \text{if } c > \delta_c; \\ 0.08 & \text{otherwise,} \end{cases} \quad (7)$$

where c represents the concentration of the odor at the robot's current position, and δ_c is the threshold concentration level used to determine whether the robot should increase its linear velocity.

For the DQN algorithm, trained over 5000 episodes, the initial values of l_v and ω_c were set at 0.02 m/s and 0.02 rad/s, respectively. This training aimed to optimize the robot's ability to navigate toward the odor source based on real-time sensor inputs, without relying on a pre-existing knowledge base.

Each experimental setup involved five test runs, initialized from the same five starting positions. Fig. 15(b) shows these starting positions

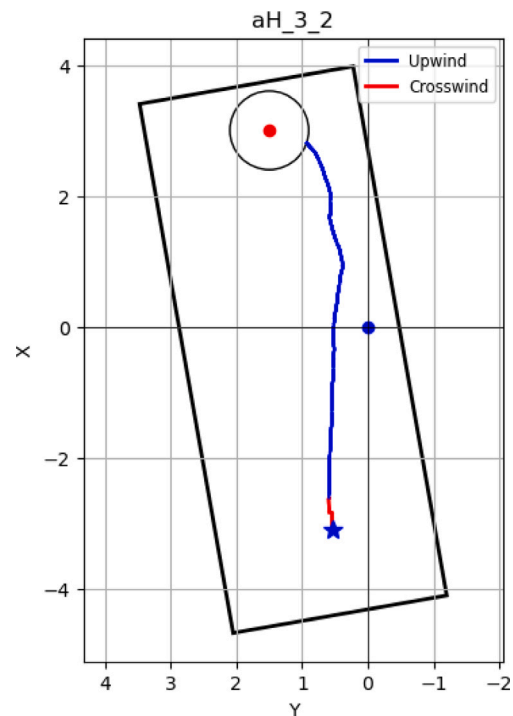


Fig. 17. Robot's navigation path during an OSL test under the Adaptive-Hint configuration with a 3-shot memory setting. The robot's trajectory is depicted, showing its approach toward the odor source, marked by the red dot. The blue and red lines represent the robot's movement upwind and crosswind, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

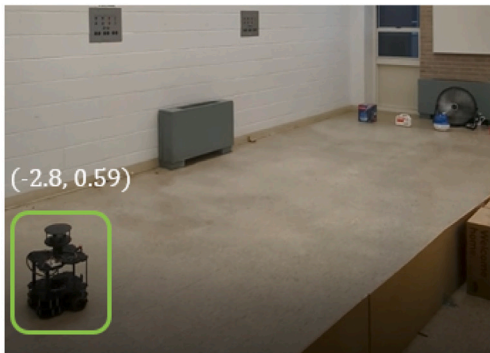
and the airflow setups used during the experimental runs. The experiments evaluated each algorithm using predefined performance metrics: Success Rate (SR), averaged Travel Distance (TD), and averaged Search Time (ST), as detailed in Section 4.1. The comprehensive testing of both approaches allows us to comprehensively assess the practical applicability, efficiency, and adaptability of each method in dynamic environments, offering insights into their potential uses in challenging OSL tasks.

4.7.4. Sample trials

To demonstrate the practical application and validate the performance of our Knowledge-driven OSL framework, we conducted specific trial runs and monitor each step and record the robot's actions and decisions. In a representative trial run using the Knowledge-driven OSL framework with the 3-shot memory setting under the Adaptive-Hint configuration, the robot was initialized at a starting point on the edge of the search area. Upon initialization, the robot immediately began processing environmental data using its chemical sensor and anemometer. At 5 s, it detected a significant chemical concentration, indicating the presence of an odor plume and triggering its navigation toward the source. The robot adjusted its path based on the intensity of ethanol vapor and wind direction data. By 56.2 s, the robot successfully reached within 0.6 meters of the odor source, well within the 120-second time limit set for the experiment.

The integration of prior experiences allowed the robot to optimize its path efficiently, avoiding unnecessary detours and steadily approaching the odor source. The trial conclusively demonstrated the framework's effectiveness, with the robot achieving its goal significantly under the allotted time. The robot's trajectory during this trial is presented in Fig. 17, while snapshots of the robot's navigation at different times are shown in Fig. 18. These images illustrate the robot's decision-making process as it responds to varying plume concentrations.

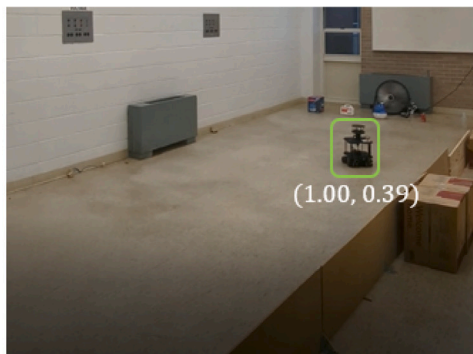
Scenario description:
 You are moving in an area searching for the plume source. Your current position is at (-2.8, 0.59). At this position, the plume concentration is 649.00.



Agent reasoning and decision:
 The plume concentration at the current position is 649.00, which is considered low. The intention is to locate the odor source effectively, which suggests that I should move to areas with potentially higher concentration.
 - Since the concentration is low, it is less likely that moving against the wind direction will lead me to the source. The source should be in the direction where the concentration increases.
 - Moving across the wind direction could potentially lead to areas with higher concentration as it suggests exploring new areas where the plume might be more concentrated.
 Decision: Move across the wind direction

(a)

Scenario description:
 You are moving in an area searching for the plume source. Your current position is at (1.00, 0.39). At this position, the plume concentration is 667.00.



Agent reasoning and decision:
 The plume concentration at the current position is 667.00, which indicates that I am in the presence of the plume since the concentration increases as I approach the source.
 - Since the goal is to locate the odor source effectively and the concentration is already high, I should move against the wind direction to trace the plume back to its source.
 Decision: Move against the wind direction

(b)

Fig. 18. Snapshots of robotic agent navigation to the odor source location in different time. (a) represents the LLM's reasoning and decision-making at position (-2.8, 0.59) of the robotic agent. (b) represents the LLM's reasoning and decision-making at position (1.00, 0.39) of the robotic agent.

4.7.5. Experimental results

In our real-world experiments, the Knowledge-driven OSL framework under different configurations— Adaptive-Hint, Adaptive-No Hint, and Hint Only—each tested with 0-shot, 3-shot, and 5-shot memory settings. The performance was primarily measured using Success Rate (SR), averaged Travel Distance (TD), and averaged Search Time (ST) metrics.

The Adaptive-Hint configuration with the 3-shot memory setting emerged as the most effective, achieving a perfect success rate with optimized travel distance and search time. This configuration struck a successful balance between utilizing historical data and adapting to real-time environmental changes, leading to efficient navigation and odor source localization. In contrast, the Adaptive-No Hint configuration demonstrated moderate success. While it maintained adaptability, the absence of hints led to slightly less efficient performance, particularly in the 3-shot setting, where longer search times and increased travel distances were observed. The Hint Only configuration showed that strategic hints alone could enhance decision-making, though it was less effective compared to the adaptive configurations, which dynamically adjusted based on environmental inputs. The results of these configurations are summarized in Table 6.

Table 6

Performance metrics for the original odor source location.

Configuration	Memory	Success Rate (SR) ↑	Averaged Travel Distance (TD) ↓	Averaged Search Time (ST) ↓
Adaptive-Hint	0-shot	0.8	5.16	64.18
	3-shot	1.0	5.79	53.03
	5-shot	1.0	5.85	53.81
Adaptive-No Hint	0-shot	0.8	5.09	65.59
	3-shot	1.0	6.07	56.85
Hint Only	3-shot	1.0	5.59	51.36

Overall, these experiments highlighted the significant impact of integrating memory and adaptive strategies into the LLM-driven decision-making process, resulting in more accurate and effective navigation to the odor source. We visually presented the results of the navigation strategies and the robot's trajectories in Figs. 19.

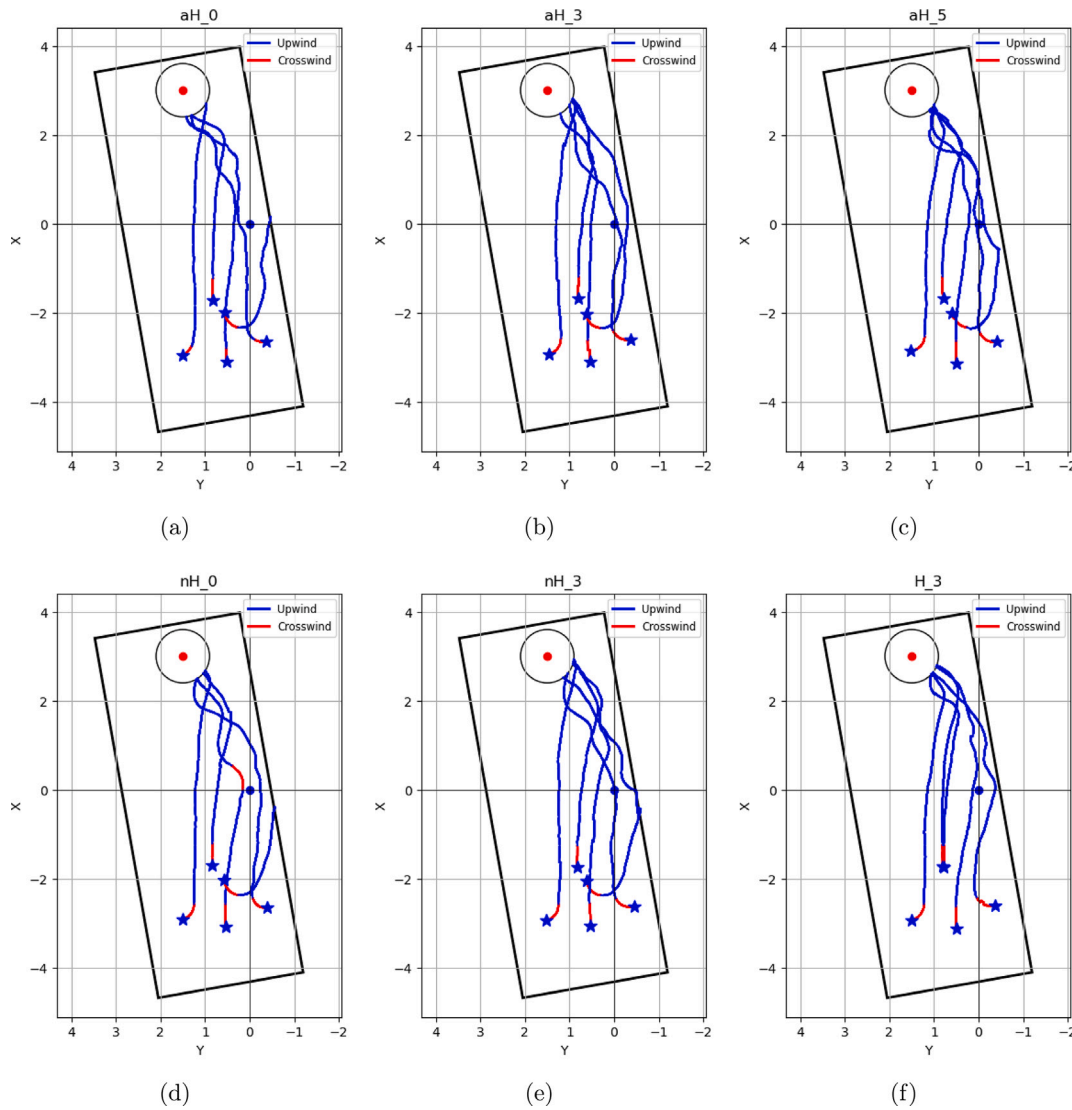


Fig. 19. (a), (b) and (c) show the results of LLM within the Adaptive-Hint configuration in 0-shot, 3-shot and 5-shot setting respectively. (d) and (e) show the results of LLM within the Adaptive-No Hint configuration in 0-shot and 3-shot setting respectively. (f) shows the results of LLM within the Only Hint configuration in 3-shot setting.

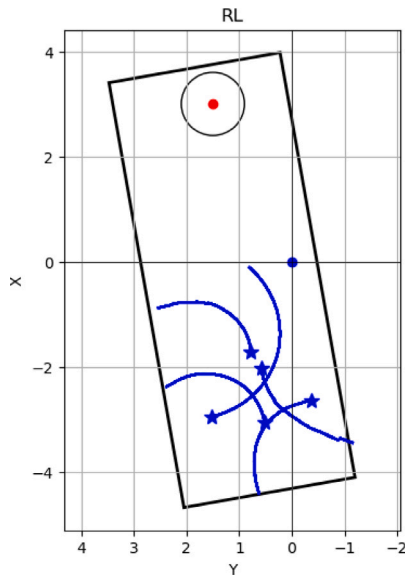


Fig. 20. Shows the results of DQN trained with 5000 episodes.

To provide a comparative analysis, we also evaluated the DQN algorithm trained over 5000 episodes under similar experimental conditions. The DQN demonstrated a lower success rate and less reliable results compared to the Knowledge-driven OSL framework. It struggled with longer travel distances and extended search times, underscoring its limitations in dynamic environments. We presented these results in Fig. 20, which illustrates the trajectories of the robotic agent.

The experimental results demonstrate the superior performance of the Knowledge-driven OSL framework, particularly within the 3-shot setting of the Adaptive-Hint configuration. This configuration not only achieved a perfect success rate but also delivered optimal metrics for travel distance and search time. In contrast, the DQN approach lacked consistency and adaptability for robust real-world odor localization tasks.

We measured the inference time of the LLM during navigation and observed that the minimum inference time is 3.59 s. While this introduces a delay compared to traditional reinforcement learning or sensor-based navigation methods, it did not significantly affect navigation accuracy in our experiments. More importantly, our knowledge-driven approach demonstrates strong generalization capability, performing effectively even when the odor source is flipped during the tests. In contrast, the DQN struggled to generalize across different search environments, as illustrated in our experimental results. For instance,

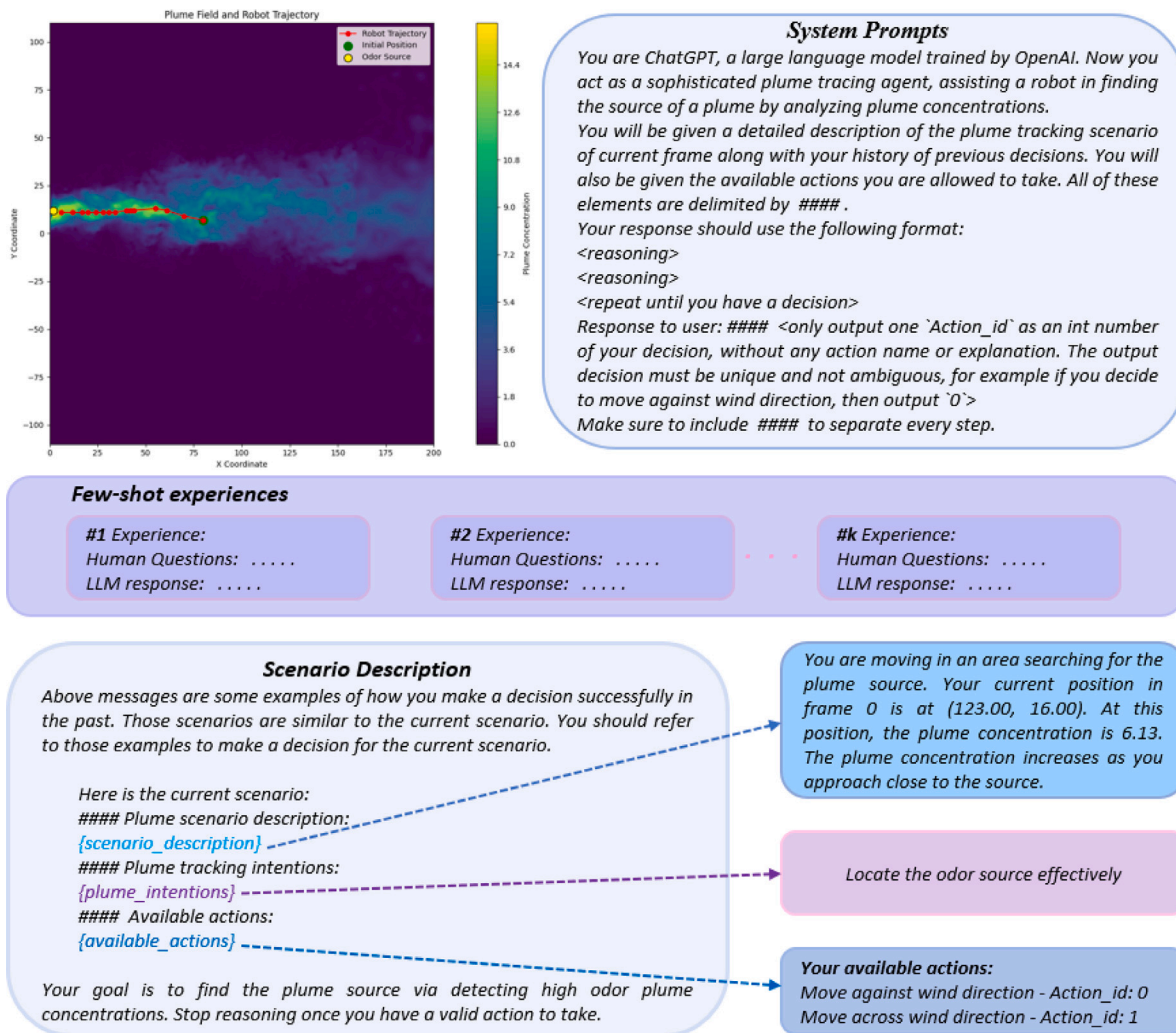


Fig. A.21. The prompt template for the reasoning module in our OSL experiment. The prompts: System Prompts box and Scenario Description box are fixed, while the prompts in the colored boxes vary depending on the current scenario. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 14 in Section 4.6 compares the results of our method with DQN, while Figs. 19 and 20 further emphasize this difference, particularly in real-world adaptation scenarios.

The comprehensive analysis underscores the superior adaptability and efficiency of the Knowledge-driven OSL framework compared to traditional reinforcement learning approaches like DQN. This demonstrates the framework’s practical applicability and effectiveness in dynamic real-world odor localization tasks, suggesting promising potential for future enhancements and applications.

5. Conclusion and future work

In this research project, we developed a knowledge-driven framework for robotic odor source localization (OSL) by integrating large language models (LLMs) to enhance the robot’s navigation capabilities. Our framework leverages the contextual understanding and decision-making abilities of LLMs, augmented with a memory module that stores and recalls past experiences. Through a series of simulations and real-world experiments, we demonstrated the effectiveness of our approach compared to a traditional reinforcement learning method using Deep Q-Networks (DQN). The results from both simulated and real-world studies indicate that the inclusion of memory, especially with 3-shot experiences, significantly improves the robot’s performance in locating the odor source. In real-world tests, the adaptive-hint configuration with 0-shot and 3-shot memory settings consistently achieved

the highest success rates, shortest travel distances, and fastest search times, demonstrating the framework’s efficiency and robustness across various environments. The adaptive speed mechanism further enhances navigation efficiency, leading to shorter travel distances and reduced search times in simulations and physical experiments. Additionally, our framework exhibited robust generalization capabilities, maintaining high performance even when we alter the odor source location or change the experimental conditions. Overall, the knowledge-driven approach outperformed the DQN-based method in both success rate and efficiency and showcased superior adaptability to dynamic environmental conditions. It highlights the potential of integrating LLMs and memory modules for complex robotic tasks in varied and unpredictable environments.

Building on the success of our current framework, we can explore several future research directions. First, investigating more advanced memory mechanisms, such as dynamic or hierarchical memory networks, could enhance the robot’s ability to recall and utilize past experiences in more complex scenarios. The memory retrieval process is optimized using cosine similarity to identify contextually relevant scenarios. While effective for general semantic alignment, it does not prioritize decision-critical features such as wind direction, odor concentration gradients, or inferred proximity to the odor source. Enhancements like weighted embeddings or tagging memories with critical metadata could refine retrieval accuracy and allow the framework to adapt dynamically to varying environmental complexities.

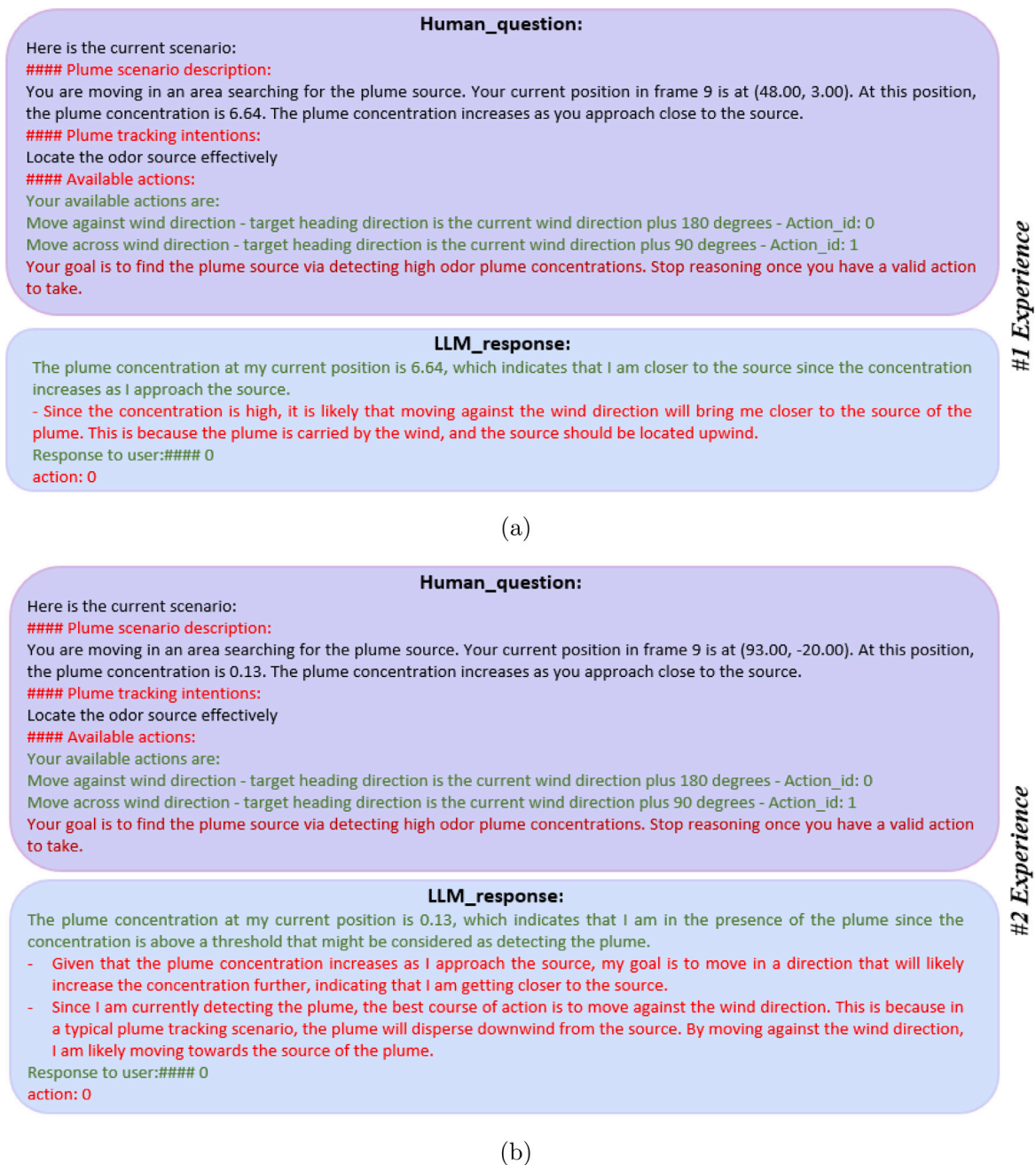


Fig. A.22. The 3-shot prompts example for the scenario shown.

The framework currently relies on training data from low-variability scenarios, which may limit its effectiveness in dynamic or complex environments. Expanding the training data to include more diverse and high-variability scenarios, such as through synthetic or augmented data, could improve its generalization to unseen conditions.

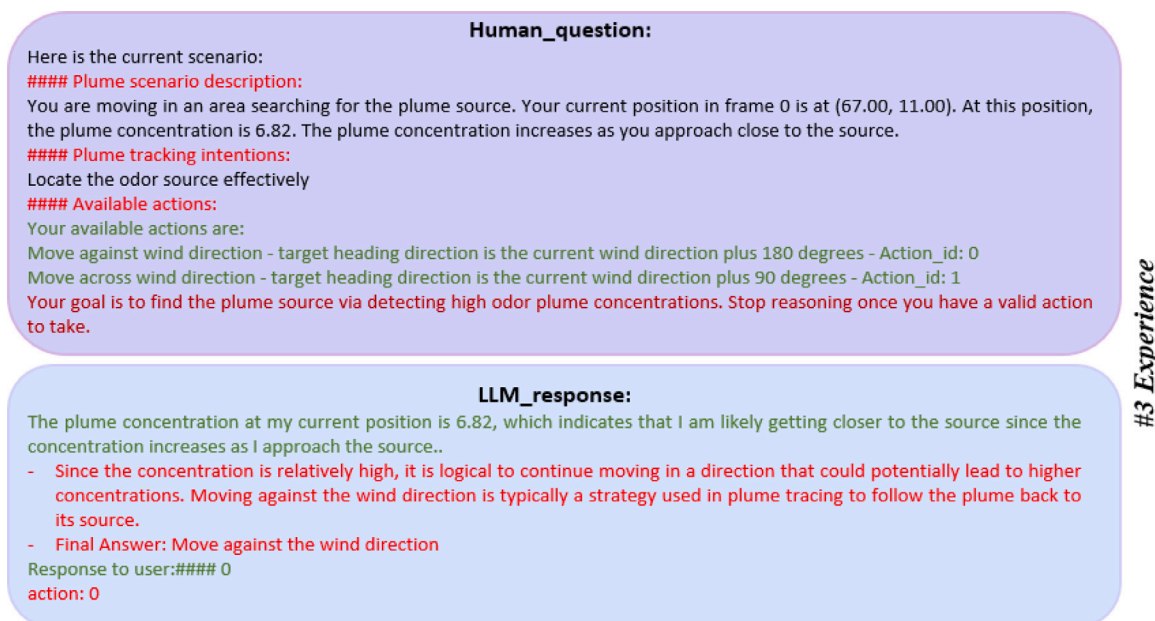
While this study focuses on single-source odor localization to establish a baseline for the framework's effectiveness, investigating multiple and distracting odor source localization is one of our future research directions. To address the challenges of multi-source environments, improvements would be needed in decision-making processes, filtering mechanisms, and learning algorithms. Additionally, integrating Kalman filters with LLM pre-filtered information offers a promising direction for enhancing real-time tracking and state estimation, improving computational efficiency and robustness in dynamic scenarios.

Moreover, extending the framework to support multiple collaborative robots could improve efficiency and accuracy in locating odor

sources, especially in more significant or more challenging environments. Lastly, additional sensory inputs, such as thermal imaging or infrared sensors, could provide a more comprehensive understanding of the environment and enhance the robot's decision-making process. Exploring reinforcement learning techniques that can learn effectively from sparse rewards could complement our knowledge-driven approach, particularly when odor concentrations are infrequent or difficult to detect. Finally, enhancing the framework's few-shot learning capabilities by fine-tuning the LLMs with domain-specific data could lead to more accurate and contextually relevant decision-making.

CRediT authorship contribution statement

Khan Raqib Mahmud: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lingxiao Wang:** Writing – review & editing, Validation,



(c)

Fig. A.22. (continued).

Supervision, Resources, Project administration, Methodology, Funding acquisition, Data curation, Conceptualization. **Sunzid Hassan:** Software. **Zheng Zhang:** Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lingxiao Wang reports financial support was provided by State of Louisiana Board of Regents. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Prompts generation in knowledge-driven OSL framework

In this section, we detail the specific design of prompts used in the reasoning module. As mentioned in the article, the prompts for the reasoning module primarily consist of three parts: system prompts, scenario description, and few-shot experience. Each part plays a crucial role in guiding the LLM to make informed decisions for the robotic odor source localization (OSL) task.

A.1. Reasoning prompts generation

The system prompts section is entirely fixed and includes the foundational information necessary for the LLM to understand the task at hand. This section provides an introduction to the odor source localization task, detailed instructions for input and output formats, and specific formatting requirements for LLM responses to ensure consistency and clarity. Fig. A.21 illustrates the prompt template used in our OSL experiment, highlighting the fixed nature of the system prompts and the dynamically generated scenario descriptions.

The scenario description, while mostly fixed, contains dynamically generated parts based on the current decision frame. This section provides information about the robot's current position within the search area, details about wind direction and speed, odor concentrations detected by the robot's sensors, and environmental conditions that may

affect navigation. These dynamic elements are crucial as they provide the real-time context necessary for the LLM to generate appropriate navigation strategies. The scenario description is embedded into vectors and used as query inputs to the memory module to retrieve relevant experiences. Available actions include moving against the wind direction or moving across the wind direction. The default navigation intention is: Your navigation intention is to locate the odor source efficiently.

As for the few-shot experience, it is entirely obtained from the memory module. Each experience consists of a human-LLM dialogue pair, where the human question includes the scenario description at that decision frame, and the LLM response represents the correct reasoning and decision made by the robotic agent. The extracted experiences are directly utilized with a few-shot prompting technique to input into the LLM, enabling in-context learning. This approach allows the reasoning module to adapt its decision-making process based on previous experiences stored in the memory module, ensuring continuous improvement and adaptability in locating the odor source.

The few-shot experience component is derived entirely from the memory module and consists of a human-LLM dialogue pair. The human question includes the scenario description at that decision frame, and the LLM response represents the reasoning and decision made by the robotic agent. These experiences are used with a few-shot prompting technique to input into the LLM, enabling in-context learning. This process allows the reasoning module to adapt its decision-making based on previously encountered scenarios, ensuring continuous learning and improvement. Fig. A.22 demonstrates the results of a 3-shot experience query, which includes three "move against the wind direction" decisions.

To illustrate, let us consider an example scenario where the robotic agent is tasked with navigating toward an odor source. The prompt generation process would include system prompts, such as a task description and instructions, a dynamically generated scenario description, and relevant few-shot experiences. The combination of these elements enables the LLM to generate well-informed decisions for the current scenario.

By leveraging LLMs and memory for contextual understanding and decision-making, our framework provides a significant advancement over traditional OSL methods, offering improved adaptability and efficiency in dynamic environments. Figs. A.21 and A.22 illustrate the

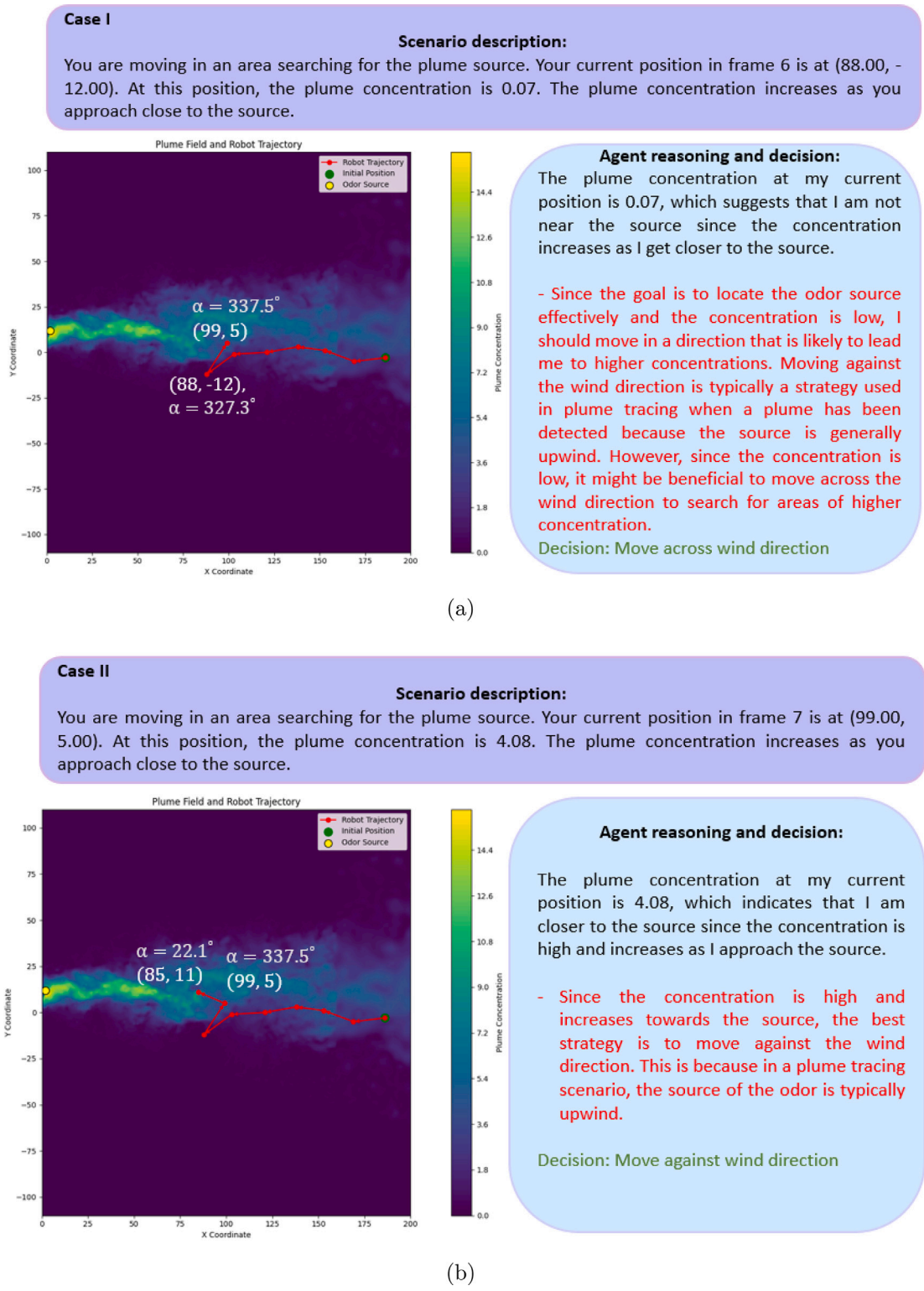


Fig. A.23. (a) Case Study I: Reasoning Module for action 1 - Move across wind direction. (b) Case Study II: Reasoning Module for action 2 - Move against wind direction.

detailed structure and examples of the prompts used in the reasoning module, showcasing how they guide the LLM in generating effective navigation strategies.

A.2. Case study of reasoning module

First, we present the results of the reasoning module for two cases, as shown in Fig. A.23. In Case Study I (Fig. A.23(a)), the robotic agent

decides to move across the wind direction based on the current environmental data and stored experiences. The reasoning module processes the sensory inputs, retrieves relevant past experiences, and generates a prompt that guides the robot to move across the wind direction. This decision is based on the logic that moving across the wind direction will likely lead to encountering a higher concentration of the odor plume.

In Case Study II (Fig. A.23(b)), the scenario is different, prompting the robotic agent to move against the wind direction. The reasoning

module again processes the current sensory inputs, retrieves relevant past experiences, and generates a prompt that instructs the robot to move against the wind direction. This decision is made because moving against the wind direction is assessed as the best strategy to locate the odor source based on the environmental conditions and the agent's experiences.

These case studies illustrate how the reasoning module enables the robotic agent to adapt its navigation strategy based on real-time data and stored knowledge, thereby improving the efficiency and accuracy of locating the odor source.

Data availability

Data will be made available on request.

References

- [1] G.A. Nevitt, Olfactory foraging by antarctic procellariiform seabirds: Life at high Reynolds numbers, *Biol. Bull.* 198 (2) (2000) 245–253.
- [2] A.D. Hasler, A.T. Scholz, *Olfactory Imprinting and Homing in Salmon*, Springer-Verlag, New York, 1983.
- [3] R.T. Cardé, A. Mafra-Neto, Mechanisms of flight of male moths to pheromone, in: *Insect Pheromone Research: New Directions*, Springer US, 1997, pp. 275–290.
- [4] N. Vickers, Mechanisms of animal navigation in odor plumes, *Biol. Bull.* 198 (2) (2000) 203–212.
- [5] R.T. Cardé, Odour plumes and odour-mediated flight in insects, in: *Ciba Foundation Symposium 200 - Olfaction in Mosquito-Host Interactions*, John Wiley & Sons, Ltd, 1996, pp. 54–70.
- [6] L. Marques, H. Magalhães, R. Baptista, J. Macedo, Mobile Robot Olfaction State-Of-The-Art and Research Challenges, *Institution of Engineering and Technology eBooks*, 2022, pp. 213–248, http://dx.doi.org/10.1049/pbce097e_ch9.
- [7] G. Kowadlo, R.A. Russell, Robot odor localization: A taxonomy and survey, *Int. J. Robot. Res.* 27 (8) (2008) 869–894, <http://dx.doi.org/10.1177/0278364908095118>.
- [8] M. Dunbabin, L. Marques, Robots for environmental monitoring: Significant advancements and applications, *IEEE Robot. Autom. Mag.* 19 (1) (2012) 24–39.
- [9] L. Wang, S. Pang, M. Noyela, K. Adkins, L. Sun, M. El-Sayed, Vision and olfactory-based wildfire monitoring with uncrewed aircraft systems, in: *2023 20th International Conference on Ubiquitous Robots, UR, IEEE, 2023*, pp. 716–723.
- [10] S. Soldan, G. Bonow, A. Kroll, Robogasinspector-a mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results, *IFAC Proc. Vol.* 45 (8) (2012) 33–38.
- [11] G. Ferri, M.V. Jakuba, 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.
- [12] X. Chen, J. Huang, Odor source localization algorithms on mobile robots: A review and future outlook, *Robot. Auton. Syst.* 112 (2019) 123–136.
- [13] J.H. Belanger, M.A. Willis, Adaptive control of odor-guided locomotion: Behavioral flexibility as an antidote to environmental Unpredictability1, *Adapt. Behav.* 4 (3–4) (1996) 217–253.
- [14] J.H. Belanger, E.A. Arbas, Behavioral strategies underlying pheromone-modulated flight in moths: lessons from simulation studies, *J. Comp. Physiol. A* 183 (1998) 345–360.
- [15] F.W. Grasso, T.R. Consi, D.C. Mountain, J. Atema, Biomimetic robot lobster performs chemo-orientation in turbulence using a pair of spatially separated sensors: Progress and challenges, *Robot. Auton. Syst.* 30 (1) (2000) 115–131.
- [16] F.W. Grasso, Invertebrate-inspired sensory-motor systems and autonomous, olfactory-guided exploration, *Biol. Bull.* 200 (2) (2001) 160–168.
- [17] H. Ishida, Y. Kagawa, T. Nakamoto, T. Moriizumi, Odor-source localization in the clean room by an autonomous mobile sensing system, *Sensors Actuators B* 33 (1) (1996) 115–121.
- [18] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, J. Janata, Plume-tracking robots: A new application of chemical sensors, *Biol. Bull.* 200 (2) (2001) 222–226.
- [19] Y. Kuwana, S. Nagasawa, I. Shimoyama, R. Kanzaki, Synthesis of the pheromone-oriented behaviour of silkworm moths by a mobile robot with moth antennae as pheromone sensors1This paper was presented at the Fifth World Congress on Biosensors, Berlin, Germany, 3–5 June 1998.1, *Biosens. Bioelectron.* 14 (2) (1999) 195–202.
- [20] S. Pang, J.A. Farrell, Chemical plume source localization, *IEEE Trans. Syst. Man Cybern. B* 36 (5) (2006) 1068–1080.
- [21] L. Wang, S. Pang, J. Li, Olfactory-based navigation via model-based reinforcement learning and fuzzy inference methods, *IEEE Trans. Fuzzy Syst.* 29 (10) (2021) 3014–3027, <http://dx.doi.org/10.1109/TFUZZ.2020.3011741>.
- [22] W. Li, J.A. Farrell, S. Pang, R.M. Arrieta, Moth-inspired chemical plume tracing on an autonomous underwater vehicle, *IEEE Trans. Robot.* 22 (2) (2006) 292–307.
- [23] S. Shigaki, S. Haigo, C.H. Reyes, T. Sakurai, R. Kanzaki, D. Kurabayashi, H. Sezutsu, Analysis of the role of wind information for efficient chemical plume tracing based on optogenetic silkworm moth behavior, *Bioinspiration Biomim.* 14 (4) (2019) 046006.
- [24] A.M. Matheson, A.J. Lanz, A.M. Medina, A.M. Licata, T.A. Currier, M.H. Syed, K.I. Nagel, A neural circuit for wind-guided olfactory navigation, *Nature Commun.* 13 (1) (2022) 4613.
- [25] R.T. Cardé, A. Mafra-Neto, Mechanisms of Flight of Male Moths to Pheromone, Springer, MA, 1996, pp. 275–290, http://dx.doi.org/10.1007/978-1-4615-6371-6_25.
- [26] J. Li, Q. Meng, Y. Wang, M. Zeng, Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm, *Auton. Robots* 30 (3) (2011) 281–292.
- [27] L. Wang, S. Pang, J. long Li, Olfactory-based navigation via model-based reinforcement learning and fuzzy inference methods, *IEEE Trans. Fuzzy Syst.* 29 (10) (2021) 3014–3027, <http://dx.doi.org/10.1109/TFUZZ.2020.3011741>.
- [28] S. Hassan, L. Wang, K.R. Mahmud, Robotic odor source localization via vision and olfaction fusion navigation algorithm, *Sensors* 24 (7) (2024) <http://dx.doi.org/10.3390/s24072309>.
- [29] Y. LeCun, A path towards autonomous machine intelligence, *Open Rev.* (62) (2022) 0.9. 2, 2022-06-27.
- [30] D. Driess, F. Xia, M.S.M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, P. Florence, PaLM-E: An embodied multimodal language model, 2023, [arXiv:2303.03378](https://arxiv.org/abs/2303.03378).
- [31] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, H. Li, Instruct2Act: Mapping multi-modality instructions to robotic actions with large language model, 2023, [arXiv:2305.11176](https://arxiv.org/abs/2305.11176).
- [32] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, Y. Qiao, Dilu: A knowledge-driven approach to autonomous driving with large language models, 2023, [arXiv preprint arXiv:2309.16292](https://arxiv.org/abs/2309.16292).
- [33] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, L. Fei-Fei, VoxPoser: Composable 3D value maps for robotic manipulation with language models, 2023, [arXiv:2307.05973](https://arxiv.org/abs/2307.05973).
- [34] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, A. Anandkumar, Voyager: An open-ended embodied agent with large language models, 2023, [arXiv:2305.16291](https://arxiv.org/abs/2305.16291).
- [35] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue, H. Li, Y. Qiao, LLaMA-Adapter V2: Parameter-efficient visual instruction model, 2023, [arXiv:2304.15010](https://arxiv.org/abs/2304.15010).
- [36] X. Zhu, Y. Chen, H. Tian, C. Tao, W. Su, C. Yang, G. Huang, B. Li, L. Lu, X. Wang, Y. Qiao, Z. Zhang, J. Dai, Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory, 2023, [arXiv:2305.17144](https://arxiv.org/abs/2305.17144).
- [37] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M.A. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, *Nature* 518 (2015) 529–533.
- [38] R. Minegishi, Y. Takahashi, A. Takashima, D. Kurabayashi, R. Kanzaki, Adaptive Control System of an Insect Brain During Odor Source Localization, *IEEE*, 2013, pp. 357–362, <http://dx.doi.org/10.1109/ROS.2013.6696376>.
- [39] C. Ercolani, A. Martinoli, 3D Odor Source Localization using a Micro Aerial Vehicle: System Design and Performance Evaluation, *IEEE*, 2020, pp. 6194–6200, <http://dx.doi.org/10.1109/ROS45743.2020.9341501>.
- [40] G. Lu, Q. Zhang, T. Qie, Q. Feng, A robot odor source localization strategy based on bionic behavior, *IOP Conf. Ser. Mater. Sci. Eng.* 470 (1) (2019) 012033, <http://dx.doi.org/10.1088/1757-899X/470/1/012033>.
- [41] S. Shigaki, M. Yamada, D. Kurabayashi, K. Hosoda, Robust moth-inspired algorithm for odor source localization using multimodal information, *Sensors* 23 (3) (2023) <http://dx.doi.org/10.3390/s23031475>, 1475–1475.
- [42] J. Bailey, M. Willis, R. Quinn, A multi-sensory robot for testing biologically-inspired odor plume tracking strategies, in: *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2005, pp. 1477–1481, <http://dx.doi.org/10.1109/AIM.2005.1511219>.
- [43] F. Rahbar, A. Marjovi, P. Kibleur, A. Martinoli, A 3-D bio-inspired odor source localization and its validation in realistic environmental conditions, *IEEE*, 2017, pp. 3983–3989, <http://dx.doi.org/10.1109/ROS.2017.8206252>.
- [44] J.-G. Li, Q.-H. Meng, Y. Wang, M. Zeng, Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm, *Auton. Robots* (2011) <http://dx.doi.org/10.1007/S10514-011-9219-2>.
- [45] M.J. Anderson, J. Sullivan, T.K. Horiuchi, S.B. Fuller, T.L. Daniel, A bio-hybrid odor-guided autonomous palm-sized air vehicle, *Bioinspir. Biomim.* (2020) <http://dx.doi.org/10.1088/1748-3190/ABB81>.
- [46] D.W.D. Muhamad Raasyan Fikri, Palm-sized quadrotor source localization using modified bio-inspired algorithm in obstacle region, *Int. J. Electr. Comput. Eng.* 12 (4) (2022) <http://dx.doi.org/10.11591/ijece.v12i4.pp3494-3504>, 3494–3494.

- [47] J. Belanger, M. Willis, Biologically-inspired search algorithms for locating unseen odor sources, in: Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) Held Jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell, 1998, pp. 265–270, <http://dx.doi.org/10.1109/ISIC.1998.713672>.
- [48] W. Jatmiko, F. Jovan, R.Y.S. Dhiemas, A.M. Sakti, F.M. Ivan, A. Febrian, T. Fukuda, K. Sekiyama, Robots implementation for odor source localization using PSO algorithm, WSEAS Trans. Circuits Syst. Arch. (2011) <http://dx.doi.org/10.5555/2001161.2001162>.
- [49] K. Gaurav, A. Kumar, R. Singh, Single and multiple odor source localization using hybrid nature-inspired algorithm, Sadhana-Academy Proc. Eng. Sci. 45 (1) (2020) 1–19, <http://dx.doi.org/10.1007/S12046-020-1318-3>.
- [50] D. Gong, C. liang Qi, Y. Zhang, M. Li, Modified particle swarm optimization for odor source localization of multi-robot, 2011, <http://dx.doi.org/10.1109/CEC.2011.5949609>.
- [51] Y. Zou, D. Luo, W. Chen, Swarm Robotic Odor Source Localization Using Ant Colony Algorithm, IEEE, 2009, pp. 792–796, <http://dx.doi.org/10.1109/ICCA.2009.5410516>.
- [52] S. Wang, S. Sun, M. Liu, B. Gao, Y. Wang, Resource-aware probability-based collaborative odor source localization using multiple UAVs, 2023, <http://dx.doi.org/10.48550/arxiv.2303.03830>.
- [53] M. Staples, C.H. Hugenholtz, T.E. Barchyn, M. Gao, A comparison of multiple odor source localization algorithms, Sensors 23 (10) (2023) <http://dx.doi.org/10.3390/s23104799>, 4799–4799.
- [54] L.D. Nhat, D. Kurabayashi, Odor source localization in obstacle regions using switching planning algorithms with a switching framework, Sensors 23 (3) (2023) <http://dx.doi.org/10.3390/s23031140>, 1140–1140.
- [55] L. Cheng, Y. Li, Multi-sensory olfactory quadruped robot for odor source localization*, 2023, pp. 332–335, <http://dx.doi.org/10.1109/CBS55922.2023.10115389>.
- [56] L. Wang, Z. Yin, S. Pang, Learn to trace odors: Robotic odor source localization via deep learning methods with real-world experiments, in: SoutheastCon 2023, 2023, pp. 524–531, <http://dx.doi.org/10.1109/SoutheastCon51012.2023.10115175>.
- [57] Z. Yan, Q.-H. Meng, T. Jing, S.-W. Chen, H.-R. Hou, A deep learning-based indoor odor compass, IEEE Trans. Instrum. Meas. 72 (2023) 1–10, <http://dx.doi.org/10.1109/TIM.2023.3238053>.
- [58] Z. Yan, T. Jing, S. Chen, M. Jabeen, Q.-H. Meng, A novel odor source localization method via a deep neural network-based odor compass, in: ROBOT2022: Fifth Iberian Robotics Conference, Springer International Publishing, 2023, pp. 189–200, http://dx.doi.org/10.1007/978-3-031-21062-4_16.
- [59] X. Chen, B. Yang, J. Huang, Y. Leng, C. Fu, A reinforcement learning fuzzy system for continuous control in robotic odor plume tracking, Robotica 41 (2022) 1039–1054, <http://dx.doi.org/10.1017/S0263574722001321>.
- [60] P. Ojeda, J. Monroy, J. Gonzalez-Jimenez, Robotic gas source localization with probabilistic mapping and online dispersion simulation, 2023, <http://dx.doi.org/10.48550/arxiv.2304.08879>.
- [61] D. Badawi, I. Bassi, S. Ozev, A.E. Cetin, Deep-learning-based gas leak source localization from sparse sensor data, IEEE Sens. J. 22 (21) (2022) 20999–21008, <http://dx.doi.org/10.1109/JSEN.2022.3202134>.
- [62] A.S.A. Yeon, A. Zakaria, S.M.M.S. Zakaria, R. Visvanathan, K. Kamarudin, L.M. Kamarudin, Gas source localization via mobile robot with gas distribution mapping and deep neural network, in: 2022 2nd International Conference on Electronic and Electrical Engineering and Intelligent System, ICE3IS, 2022, pp. 120–124, <http://dx.doi.org/10.1109/ICE3IS56585.2022.10010251>.
- [63] Y. Shan, H. Lu, W.T. Lou, Multi-faceted deep learning framework for dynamics modeling and robot localization learning, J. Intell. Fuzzy Systems 45 (2023) <http://dx.doi.org/10.3233/jifs-230895>.
- [64] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), in: Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.
- [65] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A.M. Dai, T.S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, PaLM: Scaling language modeling with pathways, 2022, [arXiv:2204.02311](http://arxiv.org/abs/2204.02311).
- [66] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: Open and efficient foundation language models, 2023, [arXiv:2302.13971](http://arxiv.org/abs/2302.13971).
- [67] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, other authors, GPT-4 technical report, 2024, [arXiv:2303.08774](http://arxiv.org/abs/2303.08774).
- [68] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P.F. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), in: Advances in Neural Information Processing Systems, vol. 35, Curran Associates, Inc., 2022, pp. 27730–27744.
- [69] J. Wei, M. Bosma, V.Y. Zhao, K. Guu, A.W. Yu, B. Lester, N. Du, A.M. Dai, Q.V. Le, Finetuned language models are zero-shot learners, 2022, [arXiv:2109.01652](http://arxiv.org/abs/2109.01652).
- [70] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023, [arXiv:2201.11903](http://arxiv.org/abs/2201.11903).
- [71] A. Caines, L. Benedetto, S. Taslimipoor, C. Davis, Y. Gao, O. Andersen, Z. Yuan, M. Elliott, R. Moore, C. Bryant, M. Rei, H. Yannakoudakis, A. Mullooly, D. Nicholls, P. Buttery, On the application of large language models for language teaching and assessment technology, 2023, [arXiv:2307.08393](http://arxiv.org/abs/2307.08393).
- [72] J. Göpfert, J.M. Weinand, P. Kuckertz, D. Stolten, Opportunities for large language models and discourse in engineering design, 2023, <http://dx.doi.org/10.48550/arxiv.2306.09169>, [arXiv.org abs/2306.09169](http://arxiv.org/abs/2306.09169).
- [73] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023, [arXiv:2303.18223](http://arxiv.org/abs/2303.18223).
- [74] OpenAI. Introducing chatgpt, 2023, URL <https://openai.com/blog/chatgpt/>.
- [75] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, D. Fox, RVT: Robotic view transformer for 3D object manipulation, 2023, [arXiv:2306.14896](http://arxiv.org/abs/2306.14896).
- [76] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N.J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, B. Zitkovich, RT-1: Robotics transformer for real-world control at scale, 2023, [arXiv:2212.06817](http://arxiv.org/abs/2212.06817).
- [77] Z. Zhao, H. Yu, H. Wu, X. Zhang, 6-DoF robotic grasping with transformer, 2023, [arXiv:2301.12476](http://arxiv.org/abs/2301.12476).
- [78] T. Gervet, Z. Xian, N. Gkanatsios, K. Fragkiadaki, Act3D: 3D feature field transformers for multi-task robotic manipulation, 2023, [arXiv:2306.17817](http://arxiv.org/abs/2306.17817).
- [79] D. Seamon, Improving knowledge extraction from LLMs for robotic task learning through agent analysis, 2023, <http://dx.doi.org/10.48550/arxiv.2306.06770>.
- [80] B. Hu, C. Zhao, P. Zhang, Z. Zhou, Y. Yang, Z. Xu, B. Liu, Enabling intelligent interactions between an agent and an LLM: A reinforcement learning approach, 2023, <http://dx.doi.org/10.48550/arxiv.2306.03604>, [arXiv.org abs/2306.03604](http://arxiv.org/abs/2306.03604).
- [81] V.S. Dorbala, J.F. Mullen, D. Manocha, Can an embodied agent find your “cat-shaped mug”? LLM-based zero-shot object navigation, IEEE Robot. Autom. Lett. 9 (2023) 4083–4090.
- [82] R. Schumann, W. Zhu, W. Feng, T.-J. Fu, S. Riezler, W.Y. Wang, VELMA: Verbalization embodiment of LLM agents for vision and language navigation in street view, in: AAAI Conference on Artificial Intelligence, 2023.
- [83] J. Mai, J. Chen, B. Li, G. Qian, M. Elhoseiny, B. Ghanem, LLM as a robotic brain: Unifying egocentric memory and control, 2023, [arXiv:2304.09349](http://arxiv.org/abs/2304.09349).
- [84] H. Sun, Y. Zhuang, L. Kong, B. Dai, C. Zhang, AdaPlanner: Adaptive planning from feedback with language models, 2023, [arXiv:2305.16653](http://arxiv.org/abs/2305.16653).
- [85] S. Yu, M. Cheng, J. Yang, J. Ouyang, Y. Luo, C. Lei, Q. Liu, E. Chen, Multi-source knowledge pruning for retrieval-augmented generation: A benchmark and empirical study, 2024, [arXiv:2409.13694](http://arxiv.org/abs/2409.13694).
- [86] X. Li, Application of RAG model based on retrieval enhanced generation technique in complex query processing, Adv. Comput. Signals Syst. 8 (6) (2023).
- [87] C. Yao, S. Fujita, Adaptive control of retrieval-augmented generation for large language models through reflective tags, Electronics 13 (23) (2024).
- [88] S.E. Spatharioti, D. Rothschild, D.G. Goldstein, J.M. Hofman, Comparing traditional and LLM-based search for consumer choice: A randomized experiment, 2023, <http://dx.doi.org/10.48550/arxiv.2307.03744>, [arXiv.org abs/2307.03744](http://arxiv.org/abs/2307.03744).
- [89] Z. Zhang, D. Seth, S. Artham, J.G. Leishman, E.P. Gnanamanickam, Time-resolved flowfield measurements of momentum-driven pulsed transient jets, AIAA J. 56 (4) (2017) 1434–1446.
- [90] N. Ando, R. Kanzaki, A simple behaviour provides accuracy and flexibility in odour plume tracking—the robotic control of sensory-motor coupling in silkworms, J. Exp. Biol. 218 (23) (2015) 3845–3854.
- [91] W. Huang, P. Abbeel, D. Pathak, I. Mordatch, Language models as zero-shot planners: Extracting actionable knowledge for embodied agents, 2022, [arXiv:2201.07207](http://arxiv.org/abs/2201.07207).
- [92] J. Farrell, S. Pang, W. Li, Chemical plume tracing via an autonomous underwater vehicle, IEEE J. Ocean. Eng. 30 (2) (2005) 428–442.
- [93] Q. Feng, H. Cai, Z. Chen, Y. Yang, J. Lu, F. Li, J. Xu, X. Li, Experimental study on a comprehensive particle swarm optimization method for locating contaminant sources in dynamic indoor environments with mechanical ventilation, Energy Build. 196 (2019) 145–156.



Khan Raqib Mahmud is a Ph.D. student of department of Computer Science at Louisiana Tech University in Ruston, Louisiana. He received Master of Science (M. Sc.) degree in Computer Simulation for Science and Engineering, from KTH Royal Institute Technology, Sweden and M. Sc. degree in Computational Engineering from University of Erlangen–Nuremberg, Germany with Erasmus Mundus Scholarship. His research involves Robot Navigation, Computer Vision and Pattern Recognition, Machine Learning, Artificial Intelligence and Autonomous Systems.



Sunzid Hassan is a Ph.D. student in the Department of Computer Science at Louisiana Tech University, Ruston, Louisiana 71270, USA. He received his Master's degree in Computer Science from Louisiana Tech University. His current research interests are in the fields of embodied artificial intelligence and robotic odor source localization.



Lingxiao Wang received the B.S. degree in Electrical Engineering from the Civil Aviation University of China, Tianjin, China in 2015, M.S. degree in Electrical and Computer Engineering, and Ph.D. in Electrical Engineering and Computer Science from Embry-Riddle Aeronautical University, Daytona Beach, FL, USA in 2017 and 2021, respectively. Currently, he is Assistant Professor of Electrical Engineering at Louisiana Tech University. His current research interests include autonomous systems, robotic applications, and artificial intelligence. He focuses on developing intelligent decision-making models to navigate and control robots using AI methods.



Dr. Zheng Zhang is an active researcher in experimental aerodynamics, fluid–structure interaction, and developing relevant instrumentation for low-speed wind tunnel testing. He obtained his Ph.D. in Aerospace Engineering and Mechanics from the University of Alabama in 2014. Following his doctorate, he joined Neuroscience Collaborative Center, University of Louisville as an instrumentation developer. Since August 2015, he has been working as a Postdoctoral Scholar, Research Assistant Professor, and then Senior Research Scientist at Department of Aerospace Engineering at Embry-Riddle Aeronautical University, teaching Experimental Aerodynamics and attending developing of ERAU's new low-speed wind tunnel. Dr. Zhang is a Member of AIAA and APS. His researches have been sponsored by funds from DoD, including ONR and ARO.