Towards Efficient Gas Leak Detection in Built Environments: Data-Driven Plume Modeling for Gas Sensing Robots

Wanting Jin, Faezeh Rahbar, Chiara Ercolani and Alcherio Martinoli

***Abstract*— The deployment of robots for Gas Source Localization (GSL) tasks in hazardous scenarios significantly reduces the risk to humans and animals. Gas sensing using mobile robots focuses primarily on simplified scenarios, due to the complexity of gas dispersion, with a current trend towards tackling more complex environments. However, most state-of-art GSL algorithms for environments with obstacles only depend on local information, leading to low efficiency in large and more structured spaces. The efficiency of GSL can be improved dramatically by coupling it with a global knowledge of gas distribution in the environment. However, since gas dispersion in a built environment is difficult to model analytically, most previous work incorporating a gas dispersion model was tested under simplified assumptions, which do not take into consideration the impact of the presence of obstacles to the airflow and gas plume. In this paper, we propose a probabilistic algorithm that enables a robot to efficiently localize gas sources in built environments, by combining a state-of-the-art probabilistic GSL algorithm, Source Term Estimation (STE) with a learned plume model. The pipeline of generating gas dispersion datasets from realistic simulations, the training and validation of the model, as well as the integration of the learned model with the STE framework are presented. The performance of the algorithm is validated both in high-fidelity simulations and real experiments, with promising results obtained under various obstacle configurations.**

I. Introduction

Finding the source of a gaseous chemical leak released in the air and mapping its spatial distribution have several applications in various, often critical, situations [1]. Mobile robots provide an excellent platform for chemical sensing, mainly due to their autonomous nature. However, the chaotic and unpredictable nature of gas dispersion, which is further influenced by the environment, makes Gas Source Localization (GSL) operations challenging tasks.

Particularly, efficient robotic chemical sensing algorithms often rely on knowledge of the gas dispersion phenomenon to interpret the data sensed in motion and to make informed navigation decisions [2]. Gas dispersion is a combination of advection due to airflow, which carries the molecules in its direction, turbulent diffusion, which diffuses the gas by turbulent kinetics, and molecular diffusion, which moves molecules randomly [3]. This inherent complexity makes the phenomenon hard to model, which has prompted much of the research in robotic chemical sensing to focus on simplified environments, with a steady wind applied to an obstacle-free environment. This simplification allows the shape of the gas plume to be approximated by analytical models, such as the pseudo-Gaussian model [4] or particle differential equations [5], which are computationally inexpensive and efficient in simple environments. Conversely, they do not consider the effect of the environment and the surrounding airflow on the plume [6], as shown in Figure 1. A commentary on the importance of accurate plume modeling is offered in [7] and [5], where the authors evaluated the performance of GSL algorithms under inaccurate modeling of the environments and showed that the success rate drops quickly if the model does not match the higher complexity of the dispersion. Computational Fluid Dynamics (CFD) is a powerful tool to model turbulent airflow which shapes the gas plume due to advection. However, CFD simulations are too computationally expensive to run in real-time on a resource-constrained mobile robot.

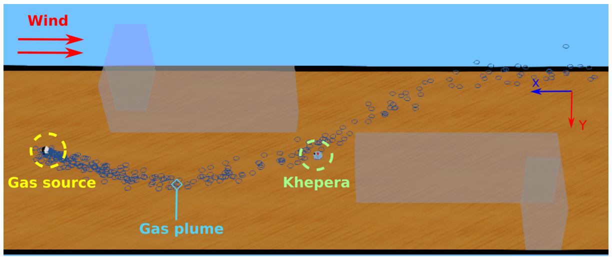


Fig. 1: Gas dispersion in a built environment reproduced in simulation: the obstacles in the environment shapes the airflow and therefore affects the gas plume.

Most state-of-the-art GSL algorithms used in cluttered environments are either reactive methods [8] [9], which are too simplistic and fragile to cope with complex environments, or map-based methods [10] [11]. Map-based algorithms first create a map of the gas distribution in a given environment and then extract the source location from the map through analytical features, such as variance [12]. Generally, map-based algorithms do not rely on assumptions about the plume, and can therefore be easily extended to built or even cluttered environments. The lack of global knowledge of gas dispersion prevents the inference of the source position from scattered measurements. Therefore, an exhaustive exploration of the environment is necessary to have the measurements in the proximity of the source [13], often resulting in a decrease in time efficiency and localization accuracy in large environments.

Another main class of GSL algorithms is probabilistic algorithms, which are able to reflect uncertain conditions in the

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. This work was funded by the Swiss National Science Foundation under grant 200020_175809. Videos accompanying this paper and additional information about the research can be found here: `https://www.epfl.ch/labs/disal/research/gassensingstructure/`

measurements and underlying models by designing a likelihood function [14]. Thus, they are able to leverage a plume model combined with samples of gas acquired in motion to produce an estimation of the location of the gas source [15]. They generate a belief of the gas source location without necessarily having to reach its proximity. One of the main algorithms in this category is called Infotaxis [16], which guides the robots to the target that yields the smallest entropy. Another widely used method is Source Term Estimation (STE) [17], which is based on an underlying plume model and estimates model parameters, using the measurements collected by a mobile robot. After each new observation, the probability distribution map of the gas source location is updated by a recursive Bayesian estimation. In [18] [19], the authors successfully conducted the GSL tasks using a mobile ground robot based on the STE framework. A combination of map-based and probabilistic algorithms for simultaneous GSL and gas distribution mapping is proposed in [20], which exploits the synergies between the STE framework and a state-of-the-art mapping algorithm. Most of the contributions that use STE for GSL rely on the pseudo-Gaussian model with strong assumptions on the environment, which prevents their application to more realistic scenarios. In [21], the STE is applied to a cluttered environment. However, the plume model used is a simple isotropic model, since the obstacles employed in this work are small enough to neglect their effects on the airflow. Another existing alternative solution is to update the belief of the estimation only based on local measurement. In [22], the authors use the wind direction and the direction between a *hit* and *miss* position to update the probability of the surrounding cells to contain the source for each measurement. However, given the exclusive dependence on local information and the lack of global information to guide the robot exploration, the efficiency of the algorithm can be low in a large space where lots of *misses* are sensed.

In this work we propose a GSL method that extends state-of-the-art probabilistic algorithms to gas sensing applications in built environments by exploiting a Data-Driven Plume Model (DDPM). This algorithm enables a ground robot to perform GSL tasks in various built environments where large obstacles are present. A DDPM is initially generated by feeding realistic gas dispersion simulation data to a Deep Convolutional Neural Network (CNN). Once the plume model is produced and validated, it is integrated into the STE framework to estimate the source of the gas. This algorithm is tested in high-fidelity simulation and real experiments, with the assumption that the geometry feature and the boundary conditions of the environment are known. The following contributions are presented in this paper:

- A pipeline to generate a gas dispersion dataset and the subsequent training to produce a novel DDPM.
- An extension of the STE framework to built environments by replacing the analytical model with DDPM.
- A thorough evaluation of the proposed algorithm with a high-fidelity simulator and physical experiments.

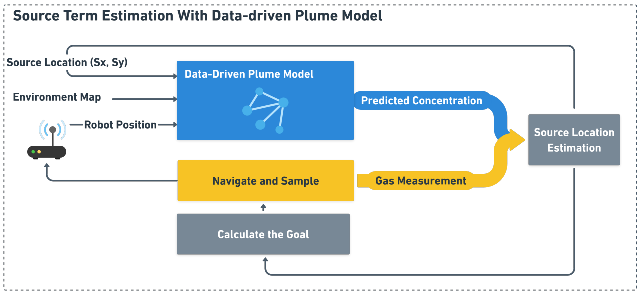


Fig. 2: Overview of the STE with a DDPM

The proposed method is introduced in Section II. The performance evaluation of the system, in both simulation and physical experiments, is reported in Section III. Lastly, a discussion of the outcome and the outlook for this work are presented in Section IV.

II. Methods

In this section, the pipeline of generating the dataset as well as the training of the DDPM is presented. Next, the integration of the DDPM to the STE framework is introduced.

A. Overview

The gas dispersion phenomenon can be modeled as a function of several variables that describe source terms, such as the release rate, the wind speed, and the location of the gas source. STE is a well-known inverse modeling technique that estimates these source terms [17]. It relies on a steady-state plume model and gas observations from the environment to update the probabilistic distribution of estimated source terms. In most STE implementations in the literature, the leveraged model, such as pseudo-Gaussian model [**?**] and isotropic model [23], is successful in (almost) uncluttered environments, where the plume is not affected. The extension of the STE framework to cluttered environments requires the estimation to be based on a more sophisticated plume model than the ones mentioned above, which takes into consideration the characteristics of the environment and the source. Many mathematical plume models exist for gas dispersion in a time-dependent fashion, such as the filament-based model [24] or the SCIPUFF model [25]. These models can faithfully simulate the dispersion of a known gas release in any environment and are usually used for gas dispersion simulations, but are too computationally expensive to be integrated in the STE framework. Therefore, in this work, we propose to leverage a DDPM generated with machine learning techniques for gas concentration prediction using simulated gas dispersion data. A global schematics of this approach is shown in Figure 2.

B. Data-Driven Plume Model

A DDPM is a surrogate plume model that predicts the gas concentration based on previously seen and learned samples. In our case, the surrogate model should take the environmental map and the estimated location of the gas source as input, and produce the gas dispersion map as output.

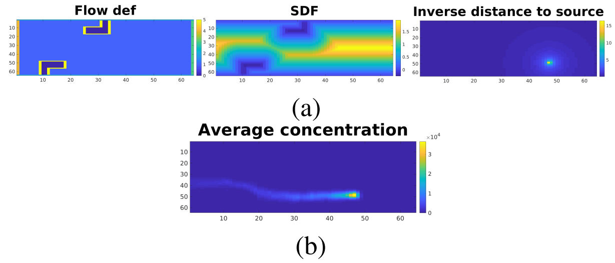
a) Dataset Generation: A diverse dataset that represents several factors influencing the gas dispersion needs to be gathered to train the DDPM. Due to the nature of the gas dispersion phenomenon, ground truth acquisition for each environment in real world would be a cumbersome and time-consuming task. A calibrated simulation is an alternative tool to generate the training dataset. For this purpose, we used OpenFOAM, an open-source CFD simulator, and Webots [26], a open-source high-fidelity robotic simulator equipped with a gas dispersion plugin [27]. The basic steps to synthesize the dataset are the following:

1) Generate environment maps with obstacles of randomized rectilinear shapes and locations;
2) Simulate wind flow in the environment maps in OpenFOAM and obtain the associated wind maps;
3) Load the wind maps into Webots and place the gas source at different randomized locations to gather the gas dispersion results;
4) Produce the feature maps for neural network training.

Two hundred environments with randomly placed obstacles have been prepared. On each map, there could be one rectangle, two separated rectangles, as well as L-shaped and U-shaped obstacles. The size of the obstacles was also randomly set on length and width. The height, however, was fixed for all the cases at 1 m. For each environment, a wind map with a fixed wind speed is generated. For each wind map, 20 gas dispersion simulations are conducted with the gas source placed randomly outside the obstacles, with a fixed release rate. In total, the training dataset is composed of 4000 dispersion maps.

OpenFOAM Simulation: OpenFOAM [28] is an open-source CFD software with a wide range of applications for simulating complex flows. The type of solver, the properties of the simulated fluid, the boundary conditions, and the solver parameters are important factors for obtaining simulation results that are faithful to reality. In our case, for a steady-state simulation of airflow around obstacles, the $simpleFoam$ solver was initially chosen, which is designed for steady-state simulations of incompressible, turbulent flow. However, the existence of the Karman Vortex, caused by obstacles, makes the environment highly transient, making the steady state solutions unsteady, and thus preventing the convergence of an iterative solver. Therefore, to find promising results under different scenarios, we approximated the steady-state solution by averaging the results of several time steps from a transient solver, $pimpleFoam$. The boundary conditions, the inlet and outlet positions, as well as the shape of the area ($14 \times 4 \times 2$ m^3) are identical for all training data to emulate the conditions of our experimental facility. For the inlet, a constant velocity of 0.75 m/s is set according to the real experimental conditions. For the outlet, the zero-gradient condition is set to simulate the free exit. The no-slip boundary is set for walls and obstacles.

Webots Simulation: Webots is a flexible open-source high-fidelity robotic simulator. A gas dispersion simulator plugin [27] based on the filament gas dispersion model [24] has been successfully developed which allows importing wind flow simulation results from OpenFOAM. For each randomly generated environment, a gas concentration map was generated by recording the gas measurements using a static sensor network placed on a 64×64 grid. Since we focus on 2D, the height of the sensor network is the same as that of the robot, at 10 cm. Each sensor gathers 5 s of the gas concentration data with a sampling rate of 10 Hz, and then both the average reading and the standard deviation are logged, together with the location of the sensors.



(a)

(b)

Fig. 3: An example of the feature maps (a) and concentration map (b) for the same simulation environment.

Feature Maps: The plume model is required to take, as input, the source position as well as the environment map and generate, as output, a map corresponding to the average gas concentration. To increase the efficiency of the learning process, for each environment map, we create three feature maps, inspired by [29] and [30]. The geometry of the environment is the most important factor in shaping the steady-state flow which, in turn, shapes the plume. A steady-state flow is mostly defined by the boundary conditions, which include the properties of each region in the map. Therefore, the first feature map is dedicated to labeling each region of the map according to its flow properties. For instance, the free space is labeled 1, the walls 2, the inlet 3, the outlet 4, and the cells inside obstacles 0. Additionally, a second feature map is designed to reflect the geometry of the map, namely the position of obstacles, where a Signed Distance Function (SDF) [30], is used to represent the distance from each cell in the environment to a surface. Moreover, points that are inside an obstacle are given negative values to make the distinction between different cells inside and outside of the obstacle. Finally, the position of the source is required to shape the plume. Therefore, the last feature map gives the inverse Euclidean distance from the source position for every cell. An example of the three feature maps is shown in Figure 3(a). To match the size of the gas concentration maps collected in Webots, we also contained the feature maps in 64×64 cells.

Test Dataset: Eight test maps, shown in Figure 4, are designed with different complexity levels to evaluate the DDPM. The shape and size of each obstacle were chosen in such a way that the obstacle is large enough to have a significant effect on the plume shape, while leaving enough space for the plume to disperse in the room. Depending on the source position with respect to the obstacles, different plume shapes can be seen for a given map. Similarly to the training set, for each test map, the gas dispersion maps with

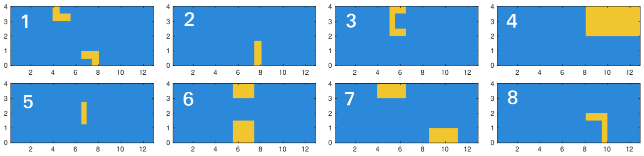


Fig. 4: The schematics of the 8 test maps.

20 random source positions are gathered as ground truth and used for the validation of DDPM.

b) Neural Network Architecture: Since the nature of the phenomenon that we seek to learn depends on the geometry of maps, a CNN [31] is leveraged, which has been widely tested for tasks that have strong spatial and/or temporal dependencies. In recent works on fluid dynamics surrogate models, such as [29] [30] [32], CNNs drastically improve computational time compared to traditional CFD simulations, while resulting in high accuracy. In this work, the desired output of the network, as shown in Figure 3(b), is a spatial gas concentration map composed of 64×64 cells, where each cell represents the average gas concentration in a steady-state flow. Since both the input features and the desired output of the network are maps, the architecture of the employed CNN requires an encoder-decoder function. We adapted the network architecture from [29], where a U-Net architecture is used to predict wind velocity and pressure based on the geometry of the map of the environment. The resulting architecture is composed of a four-block encoder-decoder, where each block contains three convolution layers followed by a max pooling and a ReLU layer. The kernel size of five was chosen, and the number of filters for the four blocks are 8, 16, 32, 32 for the encoder blocks, and the same, but in the opposite order for the decoder blocks. The optimizer is chosen to be AdamW with a learning rate of 0.001, weight decay of 0.005, and without batch and weight normalization. Since the architecture used in [29] was well-investigated and the application domain is close to our case, we have conducted the learning process with the same network and parameters along with the provided open-source code. The model is trained in 1000 epochs with the 4000 samples, divided into 90% as the training set and 10% as the validation set.

C. Source Term Estimation

To estimate the gas source, the robot autonomously navigates in the room, collects the gas concentrations $D_{1:k} = \{d_1 ... d_k\}$ at discrete time iteration $1 ... k$ and at known position $\mathbf{P}_{1:k} = \{\mathbf{p}_1 ... \mathbf{p}_k\}$. In our case, since the underlying gas dispersion model simulates the steady state, the robot stops at each measurement point for 5 s while sampling at 10 Hz and the mean concentration value is used as the point measurement. At each iteration k, the probability distribution $p(\mathbf{\Theta}|D_{1:k})$ of the estimations of source term $\mathbf{\Theta}$ is updated. The next goal position $\mathbf{p}_{k+1}$ is calculated based on the current estimation of the source term. The robot navigates toward the next goal position and starts the next iteration of the algorithm loop. In this work, we extend our previous STE [15] by replacing the pseudo-Gaussian plume model with a DDPM. For more details about the STE algorithm employed in this work, please refer to [18].

a) Inference Engine: The Bayesian framework is used to update the posterior probability distribution $p\left(\mathbf{\Theta} \mid D_{1:k}\right)$ of an estimated source term $\mathbf{\Theta}$ when a new measurement d_k is available. In our case, the parameters that we seek are the source position coordinates, thus $\mathbf{\Theta} = \{S_x, S_y\}$.

$$p\left(\mathbf{\Theta} \mid D_{1:k}\right) = \frac{p\left(D_{1:k} \mid \mathbf{\Theta}\right) p\left(\mathbf{\Theta}\right)}{p(D_{1:k})} \tag{1}$$

We consider the evidence $p(D_{1:k})$ to be a normalization factor and the prior $p(\mathbf{\Theta})$ a uniform distribution in between each parameter limit. Therefore, the posterior $p(\mathbf{\Theta} \mid D_{1:k})$ is proportional to the likelihood $p(D_{1:k} \mid \mathbf{\Theta})$ in the parameters limit. Outside of this limit and inside the obstacle regions it is equal to 0.

b) Likelihood: The likelihood, presented in Equation 2, defines the probability of collecting an observation d_k, given the predicted concentration value c_k by the plume model, based on a set of source terms $\mathbf{\Theta}$. In our previous works, c_k was obtained from a pseudo-Gaussian plume model, but in this work, it is replaced by our novel DDPM. σ_M and σ_D represent standard deviations of model and measurement errors, respectively.

$$p(D_{1:k}|\mathbf{\Theta}) \propto exp\Bigg(-\frac{1}{2}\sum_{k=0}^{N}\frac{(d_k - c_k(\mathbf{\Theta}))^2}{\sigma_M^2 + \sigma_D^2}\Bigg) \tag{2}$$

c) Navigation: The navigation strategy, based on our previous work [15] uses a navigation vector that encompasses the exploration-exploitation trade-off through the weighted sum of two components. The exploration component points to the most informative point, obtained using the Kullback-Leibler divergence [33]. The exploitation component points to the location that strikes a balance between having the highest potential of containing the source and being close to the current robot position. This prevents the robot from traveling far distances when the uncertainty on the source location is still high. On top of the previous navigation strategy, the robot needs to plan its trajectory with collision avoidance. Since the environment map is known, a state-of-the-art path planning algorithm, Visibility Graph (VG) [34] algorithm is leveraged, which allows the robot to travel around the edges of the obstacles to reach its goal position.

d) End of Algorithm: The entropy on the posterior probability function of the source position, which reflects the uncertainty of the estimation, is used as the main criterion to stop the algorithm. When it goes below a certain threshold, the algorithm starts the source declaration procedure. Additionally, a timeout stops the algorithm when the estimation exceeds a predefined number of iterations.

III. Performance Evaluation

This section presents the evaluation of our STE algorithm coupled with the DDPM, performing the GSL tasks in 2D in both high-fidelity simulation and real physical experiments.

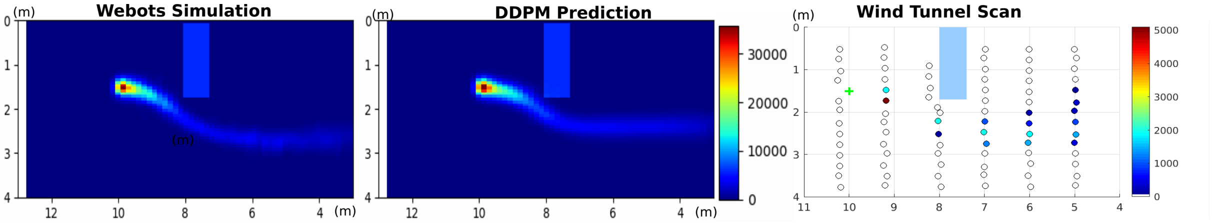


Fig. 5: An illustration of the model validation result. Left: the gas dispersion ground truth gathered by a sensor network simulated in Webots by importing the wind map from OpenFoam; middle: the gas dispersion predicted by using DDPM; right: the physical gas scan with a Khepera IV in the wind tunnel.

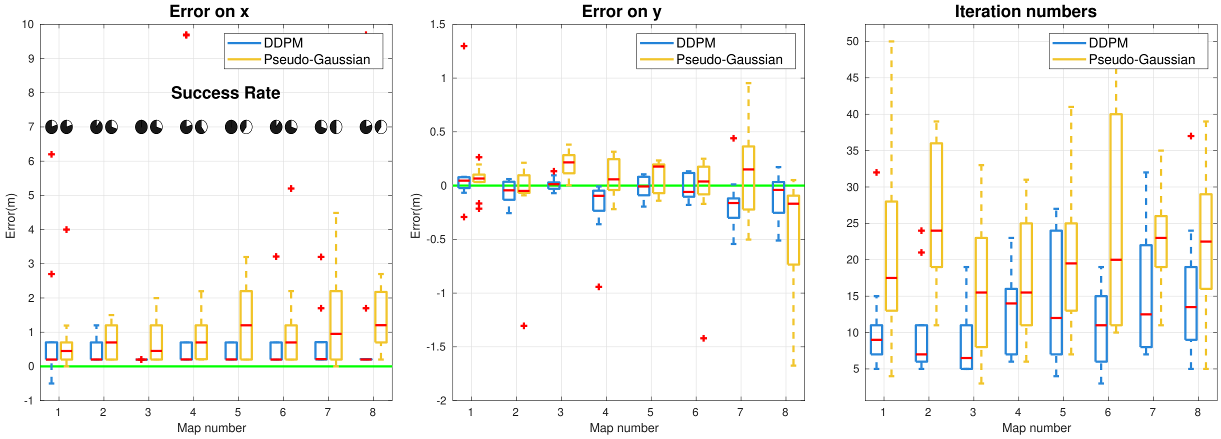


Fig. 6: The comparison of the simulation results of STE with the DDPM and the pseudo-Gaussian model.

A. *Evaluation of DDPM*

The accuracy of the model was evaluated by comparing the concentration map predicted by the model and the simulated one in Webots, used as ground truth. Additionally, we evaluate the DDPM against real gas dispersion data captured in our testbed through a lawnmower scan with the robotic platform. Once the model was validated, it was exported and used as the plume model in our STE framework for the GSL task. An example of the DDPM validation using Test Map 2, with the source located at S_x =10 m and S_y =1.5 m, can be seen in Figure 5.

B. *STE Evaluation: Simulation*

Webots is employed for the simulation evaluation of the STE algorithm. The test maps, shown in Figure 4, are recreated in Webots, and the algorithm is evaluated 10 times in each of them. A simulated Khepera IV robot equipped with gas and wind sensors runs the algorithm. A view of the simulated environment can be seen in Figure 1.

For a challenging and fair performance evaluation, the initial position of the robot is randomized within the entire arena. Similarly for the source position, the x coordinate remains in an upwind direction, close to the inlet, and the y coordinate is set randomly.

The performance of STE with DDPM as the underlying model is compared to the same algorithm exploiting the traditionally used pseudo-Gaussian plume model. We consider a run successful if the estimated source position is within 1 m of the true source position on the X-axis and 0.5 m on the Y-axis. From the results in Figure 6, we can conclude that STE with DDPM outperforms the pseudo-Gaussian model in most of the cases, with a higher success rate, smaller errors in the source position estimation, and fewer required iterations for convergence. When the gas source is not occluded by obstacles, the plume maintains a Gaussian-like shape and yields good results also for the traditional model-based STE (e.g., Test Map 1). However, when the gas source is not in line-of-sight of the robot, the DDPM clearly outperform the pseudo-Gaussian model by encouraging the robot to explore the region close to the obstacle(s).

C. *STE Evaluation: Physical Experiments*

Physical experiments were conducted in our wind tunnel, which facilitates the evaluation of the system in a repeatable fashion with controllable wind. The wind tunnel has a volume of 18 × 4 × 1.9 m^3 and is equipped with a Motion Capture System (MCS) that provides localization to the robot. A Khepera IV robot, equipped with a wind sensor board [35] and a gas sensor module sampling at 10 Hz, which is composed of a MiCS-5521 gas sensor and a mini active sniffing pump, is used as robotic platform. The gas source is represented by an electric pump vaporizing ethanol. Due to the limited computational resources of the robot, the estimation part currently runs on an external laptop and the goal positions are sent to the robot. The robot navigates and sends the measurements in return. The Test Maps {1,2,3,5} were selected to be reproduced in reality, as representative examples with different obstacle sizes and

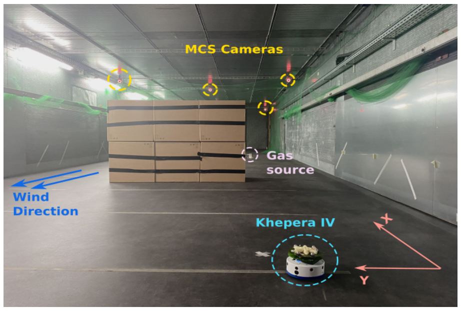


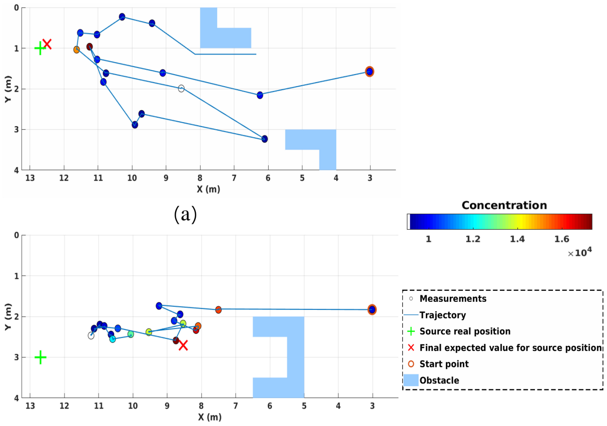
Fig. 7: A view of the experimental setup in the wind tunnel: Khepera IV robot, the obstacles, the gas source, as well as the MCS cameras.

locations. Similarly to the simulation, the x position of the source is maintained to a fixed upwind location, while the y position is selected within a predefined set, namely $S_y = \{1$ m, 2 m, 3 m$\}$ to roughly cover the Y-axis. For each setup, two experiments were conducted, totaling 24 experiments, with the results shown in Figure 9. An example of the robot trajectory of a successful run is shown in Figure 8(a).

The localization results are satisfying in most scenarios, despite the degradation in performance compared to simulation due to the non-negligible reaction and recovery time of the MOX sensor [36], which is not simulated in simulation, as well as the highly intermittent nature of gas dispersion. An example of a stark discrepancy between simulation and reality was noted in Test Map 3, where the gas leaks towards the center of the U obstacle, as shown in Figure 8(b). In this scenario, the gas will accumulate inside the U region during physical experiments, generating a locally high concentration that misleads the system, eventually resulting in a high estimation error in the x direction. Since the model used in our work is trained by time-averaged gas concentration maps, the artifact of local gas accumulation is not captured by it. Moreover, most runs failed for Test Map 5, where an obstacle is in the middle of the airflow, which gets split into two parts. The presence of the Karman Vortex in this particular case makes the airflow behind the obstacle highly dynamic, scattering the gas further. The dynamic wind flow exacerbates the intermittent properties of the gas, adding noise to the gas concentration measurements.

IV. Conclusion and Outlook

In this work, we presented a novel data-driven surrogate plume model that allows for a real-time application of a well-known probabilistic algorithmic framework to estimate the source position efficiently in realistic built indoor environments. The size of the obstacles in the environment was chosen to significantly affect the shape of the plume, therefore hindering the reliance on classical plume models. We generated the dataset of the gas dispersion maps by leveraging open-source simulation tools for the fluid dynamic and robotic aspects, and used them to train the DDPM based on a CNN. By integrating the learned CFD-based model into our previously developed and validated STE framework,



(a)

(b)

Fig. 8: Trajectory examples, from real experiments, performed by a Khepera IV carrying out GSL with a DDPM. (a) a successful run and (b) a failed run due to gas accumulation in the cavity of the U-shape obstacle.

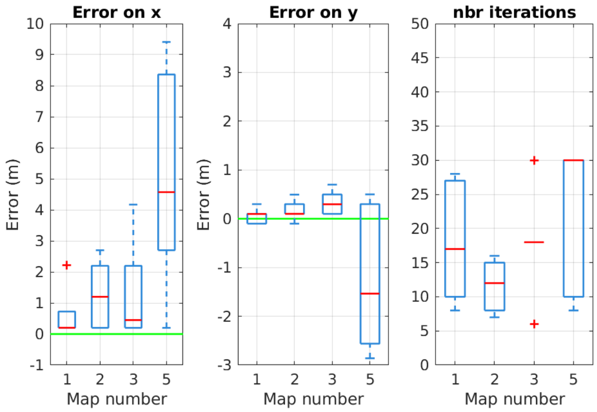


Fig. 9: Results of real experiments in the wind tunnel.

the resulting algorithm shows promising results in both simulation and physical reality characterized by various built environments.

In the future, the capacity of the algorithm to estimate additional source terms can be obtained by extending our dataset, in particular by including data gathered under different conditions (e.g., wind speed and source release rate). The simulation-to-reality gap of the gas dispersion phenomenon in cluttered environments will also be among the key topic to be investigated in the future. Other features of gas detection could be exploited, such as the variance of the concentration [12] or the gas hit frequency [37], since the absolute gas concentration is tricky to be modeled in different scenarios. Additionally, the integration of information about obstacle placement and geometric dimensions, as well as fluid dynamic considerations will be explored to improve the efficiency of the robotic sensing system.

Acknowledgement

This article has benefited from technically insightful discussions with Tristan Revaz and Prof. Fernando Porté-Agel of the Wind Engineering and Renewable Energy Laboratory, EPFL, about CFD theory and simulation.

References

[1] T. Jing, Q. Meng, and H. Ishida, "Recent Progress and Trend of Robot Odor Source Localization," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 16, no. 7, pp. 938–953, July 2021.

[2] A. Keats, E. Yee, and F.-S. Lien, "Bayesian inference for source determination with applications to a complex urban environment," *Atmospheric Environment*, vol. 41, no. 3, pp. 465–479, Jan. 2007.

[3] M. Rossi and D. Brunelli, "Gas Sensing on Unmanned Vehicles: Challenges and Opportunities," in *2017 New Generation of CAS*. Genova, Italy: IEEE, Sept. 2017, pp. 117–120.

[4] S. P. Arya *et al.*, *Air pollution meteorology and dispersion*. Oxford University Press, 1999, vol. 310.

[5] T. Wiedemann, A. Lilienthal, and D. Shutin, "Analysis of Model Mismatch Effects for a Model-Based Gas Source Localization Strategy Incorporating Advection Knowledge," *Sensors*, vol. 19, no. 3, p. 520, Jan. 2019.

[6] C. S. Li, Z. H. Yang, G. Q. Cui, and B. Jin, "Odor Source Localization Research of Mobile Robots in Indoor Environments," *Applied Mechanics and Materials*, vol. 441, pp. 796–800, Dec. 2013.

[7] J. D. Rodríguez, D. Gómez-Ullate, and C. Mejía-Monasterio, "On the performance of blind-infotaxis under inaccurate modeling of the environment," *The European Physical Journal Special Topics*, vol. 226, no. 10, pp. 2407–2420, July 2017.

[8] T. Lochmatter, N. Heiniger, and A. Martinoli, "Localizing an Odor Source and Avoiding Obstacles: Experiments in a Wind Tunnel using Real Robots," in *Proc. of the 13th Int. Symp. on Olfaction and Electronic Nose*. Brescia (Italy): AIP Conference Proceedings Series, Apr. 2009, pp. 69–72.

[9] B. P. Duisterhof, S. Li, J. Burgués, V. J. Reddi, and G. C. H. E. de Croon, "Sniffy bug: A fully autonomous swarm of gas-seeking nano quadcopters in cluttered environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 9099–9106.

[10] J. Burgués, V. Hernández, A. Lilienthal, and S. Marco, "Smelling Nano Aerial Vehicle for Gas Source Localization and Mapping," *Sensors*, vol. 19, no. 3, p. 478, Jan. 2019.

[11] J. G. Monroy, J.-L. Blanco, and J. Gonzalez-Jimenez, "Time-variant gas distribution mapping with obstacle information," *Autonomous Robots*, vol. 40, pp. 1–16, Jan. 2016.

[12] V. Hernandez Bennetts, A. J. Lilienthal, P. P. Neumann, and M. Trincavelli, "Mobile Robots for Localizing Gas Emission Sources on Landfill Sites: Is Bio-Inspiration the Way to Go?" *Frontiers in Neuroengineering*, vol. 4, 2012.

[13] J. Burgués, V. Hernández, A. J. Lilienthal, and S. Marco, "Gas distribution mapping and source localization using a 3D grid of metal oxide semiconductor sensors," *Sensors and Actuators B: Chemical*, vol. 304, p. 127309, Feb. 2020.

[14] J. R. Bourne, M. N. Goodell, X. He, J. A. Steiner, and K. K. Leang, "Decentralized Multi-agent information-theoretic control for target estimation and localization: finding gas leaks," *The International Journal of Robotics Research*, vol. 39, no. 13, pp. 1525–1548, Nov. 2020.

[15] F. Rahbar, "Source Term Estimation Algorithms for Gas Sensing Mobile Robots," Ph.D. dissertation, EPFL, Lausanne, 2021. [Online]. Available: https://infoscience.epfl.ch/record/288390?ln=en

[16] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, Jan. 2007.

[17] M. Hutchinson, H. Oh, and W.-H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Information Fusion*, vol. 36, pp. 130–148, July 2017.

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

[19] F. Rahbar, A. Marjovi, and A. Martinoli, "Design and Performance Evaluation of an Algorithm Based on Source Term Estimation for Odor Source Localization," *Sensors*, vol. 19, no. 3, p. 656, Feb. 2019.

[20] C. Ercolani, L. Tang, and A. Martinoli, "GaSLAM: An algorithm for simultaneous gas source localization and gas distribution mapping in 3D," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Kyoto Japan, Oct. 2022, pp. 333–340.

[21] C. Rhodes, C. Liu, and W.-H. Chen, "Autonomous Source Term Estimation in Unknown Environments: From a Dual Control Concept to UAV Deployment," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2274–2281, Apr. 2022.

[22] P. Ojeda, J. Monroy, and J. Gonzalez-Jimenez, "Information-Driven Gas Source Localization Exploiting Gas and Wind Local Measurements for Autonomous Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1320–1326, Apr. 2021.

[23] M. Hutchinson, C. Liu, and W. Chen, "Source term estimation of a hazardous airborne release using an unmanned aerial vehicle," *Journal of Field Robotics*, vol. 36, no. 4, pp. 797–817, June 2019.

[24] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. Carde, "Filament-Based Atmospheric Dispersion Model to Achieve Short Time-Scale Structure of Odor Plumes:," Defense Technical Information Center, Fort Belvoir, VA, Tech. Rep., Jan. 2002.

[25] R. I. Sykes, S. F. Parker, D. S. Henn, and R. S. Gabruk, "SCIPUFF — A Generalized Dispersion Model," in *Air Pollution Modeling and Its Application XI*, ser. NATO · Challenges of Modern Society, S.-E. Gryning and F. A. Schiermeier, Eds. Boston, MA: Springer US, 1996, pp. 425–432.

[26] O. Michel, "Cyberbotics Ltd. Webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[27] "Webots Odor Simulation/Model - Wikibooks, open books for an open world." [Online]. Available: https://en.wikibooks.org/wiki/Webots_Odor_Simulation/Model

[28] "OpenFOAM." [Online]. Available: https://www.openfoam.com/

[29] M. D. Ribeiro, A. Rehman, S. Ahmed, and A. Dengel, "DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks," *arXiv:2004.08826 [physics]*, May 2020, arXiv: 2004.08826.

[30] X. Guo, W. Li, and F. Iorio, "Convolutional Neural Networks for Steady Flow Approximation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA, Aug. 2016, pp. 481–490.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[32] T. Georgiou, S. Schmitt, M. Olhofer, Y. Liu, T. Bäck, and M. Lew, "Learning Fluid Flows," in *2018 International Joint Conference on Neural Networks*, July 2018, pp. 1–8, iSSN: 2161-4407.

[33] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[34] K. J. Obermeyer and Contributors, "VisiLibity: A c++ library for visibility computations in planar polygonal environments," http://www.VisiLibity.org, 2008, v1.

[35] T. Lochmatter, "Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots," Ph.D. dissertation, EPFL, Lausanne, 2010. [Online]. Available: https://infoscience.epfl.ch/record/143049?ln=en

[36] S. Shigaki, M. Fikri, and D. Kurabayashi, "Design and experimental evaluation of an odor sensing method for a pocket-sized quadcopter," *Sensors*, vol. 18, no. 11, p. 3720, 2018.

[37] M. Schmuker, V. Bahr, and R. Huerta, "Exploiting plume structure to decode gas source distance using metal-oxide gas sensors," *Sensors and Actuators B: Chemical*, vol. 235, pp. 636–646, 2016.