

**TRAINING A MACHINE LEARNING MODEL FOR UNDERWATER
CHEMICAL SOURCE LOCALIZATION IN SIMULATED TURBULENT
FLOWS**

by

Hau Phan

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Summer 2022

© 2022 Hau Phan
All Rights Reserved

**TRAINING A MACHINE LEARNING MODEL FOR UNDERWATER
CHEMICAL SOURCE LOCALIZATION IN SIMULATED TURBULENT
FLOWS**

by

Hau Phan

Approved: _____
Austin J. Brockmeier, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Jamie Phillips, Ph.D.
Department Chair, Electrical and Computer Engineering

Approved: _____
Levi Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

ACKNOWLEDGMENTS

This Master's thesis emerged from the research work done over one and a half years. I am thankful to Dr. Austin Brockmeier, Office of Naval Research, and the Department of Electrical and Computer Engineering at the University of Delaware for providing me the opportunity to learn and work on interesting problems. I also want to thank Dr. Michael Wardlaw, Dr. Fletcher Blackmon, and Dr. Bruce Kim for sharing their insights on a wide variety of topics. I would also say thanks to Mr. Bilal Riaz, Dr. Xinjie Lan, Mr. Samet Bayram, Mr. Yuksel Karahan, Mr. Claudio Cesar Claros Olivares, and all the lab mates at Computer Neural Information Engineering Lab for providing both professional and emotional support during these two years. Again, I'm extremely grateful to Dr. Austin Brockmeier for inculcating me with a love of research and for having tons of patience.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
ABSTRACT	viii
 Chapter	
1 INTRODUCTION	1
2 UNDERWATER SIMULATION	5
2.1 Related work	5
2.2 Simulation Design	8
2.2.1 Simulated Agent Design	9
2.2.2 Simulated chemical source	10
2.2.3 Simulated water flow	11
3 SOURCE LOCALIZING MODEL	13
3.1 Related Work	13
3.2 Proposed Source Localization Model	14
3.2.1 Heatmap generator model	15
3.2.2 Wasserstein distance-based loss function	17
4 EXPERIMENT	19
4.1 Stochastic policy movement dataset	19
4.2 Comparison of two loss function versions	20
4.3 Different policy dataset	24
5 DISCUSSION	30
5.1 Limitation	31

5.2 Future work	31
6 CONCLUSION	33
BIBLIOGRAPHY	34

LIST OF TABLES

4.1	Stochastic movement policy dataset analyzing	21
4.2	Stochastic movement policy dataset visualization	22
4.3	Model performance on stochastic policy movement dataset using loss 1.	23
4.4	Model performance comparison on stochastic policy movement dataset using loss 1 and loss 2.	25
4.5	Different policy dataset analysis	28
4.6	Cross testing of models trained a stochastic, spiral, and onward datasets	29

LIST OF FIGURES

2.1	PLIF imaging system [1]	7
2.2	Mean normalized concentration field in horizontal planes obtained by PLIF. a) 4cm above a bed-level source. b) 2cm above a bed level source [1]	7
2.3	Meandering plume with centerline relative diffusion and filament growth. The area represented is 100x100 with the odor source at (5,0). Each arrow indicates the local wind vector at the tail of the arrow.[2]	8
2.4	(a) the distribution of source location (b) The distribution of the distance between the initialized agent and the source	10
2.5	(a) The distribution of water flow force (b) The distribution of water flow magnitude	12
2.6	An example of a proposed chemical plume simulation	12
3.1	Heatmap generator model architecture	16
3.2	Visualization of how the distance and angular error is calculated . .	17

ABSTRACT

Underwater chemical source localization is challenging due to the dynamic and chaotic processes involved. Averaged across long time scales, the geometry of the chemical plume is determined by the mean water flow. At shorter time scales, the turbulence of water tends to create swirls, eddies, and vortices, preventing the observation of a smooth gradient of chemical concentration. Instead, the chemical concentration in the plume downstream from a source is intermittent with mostly low-level concentrations interspersed with short high-concentration segments. Various underwater platforms could deploy chemical sensors to sample the chemical concentration and measure the water flow as they move. By traveling with a predefined trajectory, the sensors can collect observations at different positions. However, these observations may consist of only a few non-zero chemical concentration measurements along the path through the turbulent plume. It is non-trivial to process these measurements to recognize the geometry of the chemical plume and predict the chemical source's location. In order to predict the location of any chemical sources, we train recurrent neural networks to process the input time series jointly consisting of the chemical concentration observations, water flow measurements, and sensor platform movements. From there, the neural network model constructs a heatmap that represents the probability that the chemical source is located at different locations around the sensing platform. This heatmap is trained based on simulations where a sensor platform moves along different trajectories across numerous scenarios of various source locations, water flows, and turbulence characteristics. In each simulated trajectory, the heatmap at each time step (after an initial non-zero chemical concentration measurement) is compared to the true source location using a Wasserstein distance metric as the loss function. This

encourages the heatmap to minimize the expected distance given the source localization predictions and the true source location, which is known during simulation. Since Wasserstein distance keeps the geometries of distributions in consideration and it does not require the support of distributions to be the same, it provides an additional advantage in comparison to the traditional cross-entropy-based loss functions. Thus, when the source is out of the prediction range, the heatmap can still be useful to predict the direction of the chemical source location which respects the sensor platform's current location. Additionally, we show that the expected Wasserstein distance for cases where no chemical is detected leads to a regularization term that shrinks the variance of source localization predictions. In order to train and test our methodology, we created a particle-based turbulence simulation based on prior work. The simulation models the Spatio-temporal variation in water flow along with the diffusion of the dissolved chemical. In every simulation episode, the source location is randomized radially symmetric around the sensor platform. At each time step, the sensor platform moves at a fixed speed for a predefined number of steps. To assess performance, we measure the resolution-accuracy tradeoff of the heatmap prediction under various water flow characteristics. The results indicate the potential for predicting chemical source locations from chemical sensor readings from limited observations in turbulent environments.

Chapter 1

INTRODUCTION

Chemical source localizing is the process of finding the chemical source in the environment. The environment, where the chemicals could disperse in, includes underwater, above-ground, below-ground, and enclosed spaces. Chemical source localizing is crucial for many species to search for food, find mates and avoid dangers. Inspired by animal behaviors, many robots have been developed for chemical localization. Its applications vary from detecting fire or leaking gas to humanitarian activities such as searching for humans in damaged buildings after an earthquake or avalanche. In these tasks, source tracking is needed since we want the robot to get as close as possible to the target. In other searching tasks such as locating unexplored bombs and mines, we may not want to take a risk of reaching close, but remotely sensing the source location. Remote sensing either predicts the correct target location or at least narrows down the possible regions.

Chemical source localization can be divided into three stages [3]. First, the agent (organism or robot) searches for the chemical plume to detect the chemical existence. Then, the agent searches for the chemical source by following the gradients in the strength of sensed concentrations. These gradients in sensory signals are reliable to lead toward the target [4]. Finally, the agent verifies the chemical source identity. Most of the work has been done for the source tracking stage [2, 5, 6, 7, 3, 8, 9, 10]. The agent is dropped into the chemical plume and tracks the chemical source guided by the chemical gradient. This strategy is named chemotaxis [11]. Contrastively, plume searching is more difficult since the sensory signals are infrequent, noisy, and lack directional information [8]. In reality, the sparse signal may only contain a few positive concentration observations. However, the sparse signal is not totally lack of

information [12]. On the other hand, if the agent receives no concentration signal, it can narrow its assumption of the source location. On the other hand, with a strong signal, the agent can make a close prediction of the source location. The agent with predefined movements can build its belief map which represents the probabilities of the source presence. By moving over different locations, it can reduce the uncertainty of the source location. This strategy is named infotaxis [2, ?, 13, 14, 15]. We considered a search for a chemical source where the agent computes an ego-centric heatmap, representing the probabilities of the source location, at every timestep. We defined our work as an infotaxis strategy.

Localizing the chemical source in a turbulent flow environment is an ill-posed and non-linear problem in the real world. The chemical substance emitted from its source is dispersing widely into the environment and is carried by the water current. The water fluid, dominated by turbulence, tends to create swirls and vortices. In such situations, it requires determining the time-average concentration to observe a smooth gradient of chemical concentration [16]. Therefore, the chemotaxis approach for source tracking tasks becomes slow in the underwater environment. Moreover, it's problematic to measure the chemical concentration as it is rapidly decayed with increasing distance from its source. Therefore, it is crucial to set the concentration threshold as close as possible to the noise level [17]. Alternatively, intermittency [5], the rate that the concentration raise above the threshold, is also useful to detect the plume presence.

Despite the difficulty of chemical source localization in the underwater environment, many aquatic organisms rely on their olfaction to search for food or mates. The best analogy for this case is the sensing done by invertebrates, namely crayfish [18, 19] and lobster [5, 20]. They have chemoreceptor neurons that are located on the antennae and other limbs. They sense the bulk motion of the water through their large antennae. They can position the antenna adaptively to sample spatially, and quickly whisk the antenna to sample the turbulent flow without causing a laminar buildup. Additionally, the microstructure of the hairs on the antennae enables rapid sampling without laminar buildup. Further, their nervous system can take these signals and fuse

them into movement strategies (a search). A fundamental question is to what extent these organisms rely on memory.

We want to replicate these characteristics to create a sensor to perform a sort of ‘remote’ sensing to localize the chemical source from distant observations of the plume. It is remote in that the chemical is assumed to be dispersed in the water and the goal is not to navigate the sensor to the actual source, but rather remotely identify the source’s location. However, localizing the chemical source without navigating to the source is a difficult process that requires combining past observations and current information. Even with the sparse sequence of observation, it is still informative to successfully localize the source [12]. Ideally, an analysis of the remotely collected information can provide a map of potential sources [2, ?, 13, 14, 15].

Through machine learning, we train a model to compute the heatmap of potential source location given the current history of observations. We assumed that the chemical molecules emitted from its source are slowly dispersed into the water environment. Their concentrations are decayed at a fixed rate over time. Averaged across long time scales, the shape of the plume is determined by the mean water flow. With various movement policies, the agent (a sensing platform) can collect observations at different locations. The observation includes the local concentration measurement, the local water flow measurement, and the movement itself. The training is based on supervised learning by providing the chemical source location (during training). Our model consists of recurrent neural networks to encode the time series of observations, and convolutional neural networks to generate the heatmaps given the encoded states. We used the Wasserstein metric loss function to minimize the expected distance between the prediction and the true source location. We trained our model across multiple simulated episodes whether the plume is detected or not. In the situation that the true source location is outside of predicting range, the predicting heatmap is still useful to give the directional information. It can help improve the success rate of chemotaxis tasks.

In order to test such a decision process, it is vital to create a simulation environment that captures the key features of real-world aquatic environments. Therefore, we need an understanding and model of how the chemical concentration behaves in aquatic environments so that we can use this information to create the simulation environment [1, 2, 5, 13]. Also, comparing results with other experiments performed in a realistic environment will provide a baseline for the validation of our simulation-based source localization model.

This thesis contributes an underwater chemical source simulation and a novel infotaxis approach to chemical source localization problems. Our environment simulation is duplicable to generate different scenarios to train the source localization model. Our simulated water flow varies within time and space to mimic the chaotic behaviors of realistic turbulence. By unbounding the environment, the computational cost is not scaled with increasing environment space. We trained a source localization model to produce an ego-centric heatmap representing the potential source location given the time series of collected observations from the sensing platform. With the preliminary results, the model achieves a good performance in predicting chemical source locations given the sparse time series of observations where the plume is detected. Moreover, our model indicates the potential of predicting the chemical source location in the non-plume detected situation.

In the next chapters, we will propose our underwater simulation, source sensing model, and testing experiment.

Chapter 2

UNDERWATER SIMULATION

2.1 Related work

Previous work on olfaction in aquatic environments relied on understanding the chemical dispersion in a turbulent flow. In particular, there can be regions with streamlined dynamics adjacent to turbulent patches. Due to turbulence and non-laminar flow, at a particular instant of time, there can be regions of high chemical density far from the chemical source and relatively lower chemical density right next to it. These are created by vortices in turbulent flow that maintain localized regions of high concentrations of chemicals. Vortices can carry the chemical to the edges of the flow or downstream. This means that the instantaneous chemical concentration is not informative as to the distance from the chemical source. It also means that the relationship between the concentration signal as a time series is strongly determined by the characteristic of the flow.

Laboratory-based flow imaging data is one way to capture the chemical dispersion in the entire scene. Planar laser-inductive fluorescence (PLIF) provides high-resolution images from three different orientations (horizontal, vertical, and traverse) of a chemical plume as it is shown in figure 2.1 [1]. Crimaldi *et al.* [1] showed the mean normalized chemical concentration field provided by the PLIF image system. Averaging images along a horizontal plane with different depths gives different peaks of concentration levels depending on the height of the plane above the source. The PLIF images shown in figure 2.2 were collected from an open channel flow. A water pump was used to provide nearly constant water flow. The freestream turbulence levels are only 2% of the flow velocity. This rate quantifies relative turbulent flows compared to

the average flow. However, this is just one case of a realistic turbulent environment and the average concentration. Even if we change conditions of this realistic test environment such as boundary conditions, water flow force then we will get different levels of the turbulent flow. Relying on previous data collection or new data collection is limiting to understand the processing across a variety of realistic test scenarios. Instead, we need a simulation environment where we can change these conditions to generate a variety of test environments virtually.

Farrell *et al.* [2] created a simulation environment with a Monte-Carlo type analysis of dispersion in a turbulent medium. The model considered instantaneous concentrations rather than time-averaged exposure. The method aims to produce a time-varying plume centerline and dispersion about the centerline to mimic a realistic environment. The statistical analysis of the plume environment is focused on the temporal and amplitude properties of the plume structure. The model adopted the sub-grid scale velocity characteristics based on the features of the advective velocity to adjust the scale of the puff for creating the plume. Farrell *et al.*'s odor plume simulation is shown in figure 2.3. However, Farrell *et al.*'s simulation did not model the fine-scale flow characteristic. Moreover, Farrell *et al.*'s simulation was built under the assumption that the vehicle with low bandwidth (≈ 1 Hz) flow sensor can capture the detail of the plume.

Michaelis *et al.* [5] analyzed the temporal and spatial intermittency of the turbulent plume in physical seawater and developed a plume tracking algorithm inspired by sea creatures such as lobster, *Panulirus Argus*. Intermittency is the rate at which the concentration rises above and falls below a certain threshold. It is useful to unravel the relationship between the chemical concentration in turbulence flow. Michaelis *et al.* [5] calculated intermittency across the full 240 s, 10 Hz concentration time series. They repeated calculating intermittency with lower temporal resolution concentration time series at 5 Hz, 2 Hz, 1 Hz, 0.5 Hz and 0.25 Hz by using every 2nd, 5th, 10th, 20th and 40th observation. For the purpose of observing concentration spikes, sampling concentration at 2 Hz and 5 Hz is effectively equivalent to 10 Hz. Moreover, they also stated that the

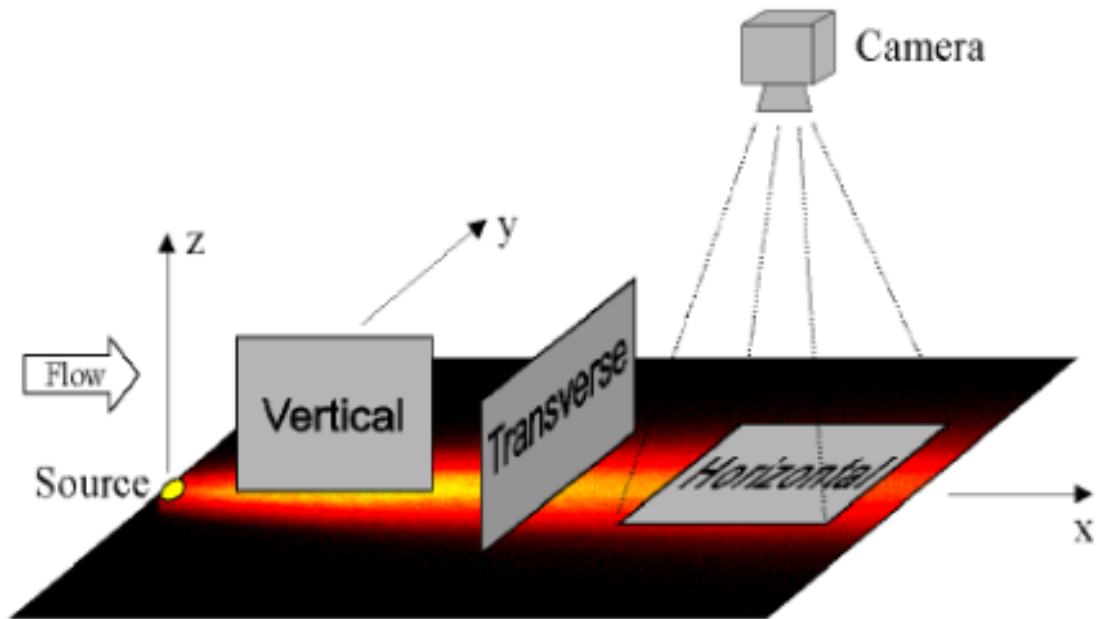


Figure 2.1: PLIF imaging system [1]

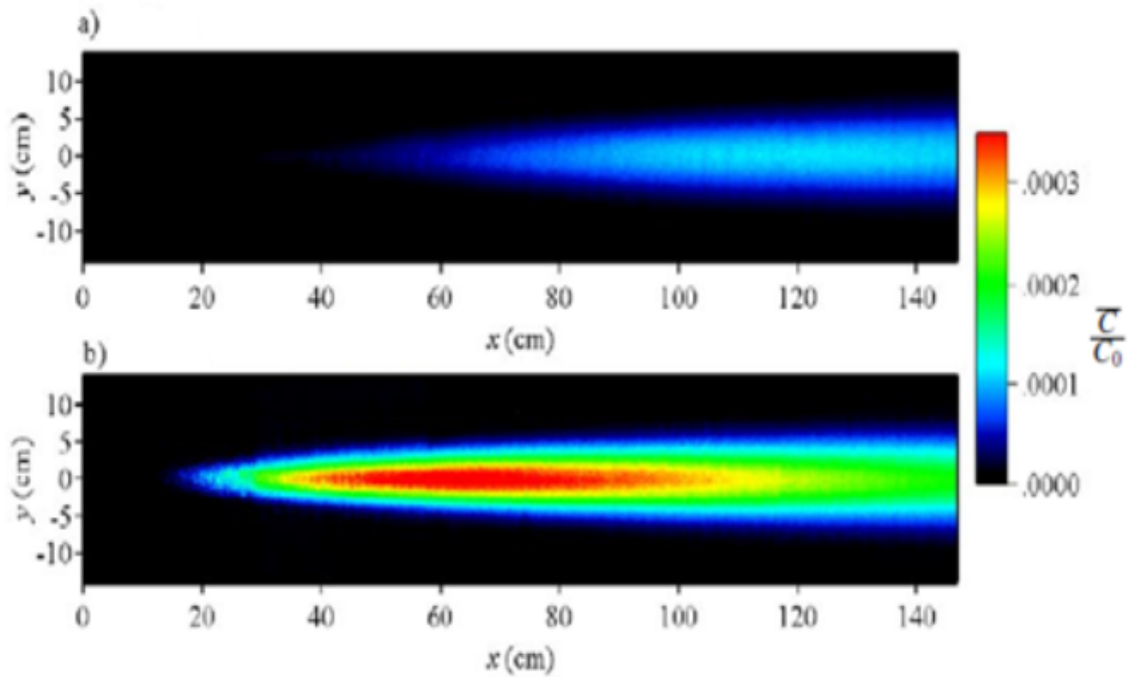


Figure 2.2: Mean normalized concentration field in horizontal planes obtained by PLIF. a) 4cm above a bed-level source. b) 2cm above a bed level source [1]

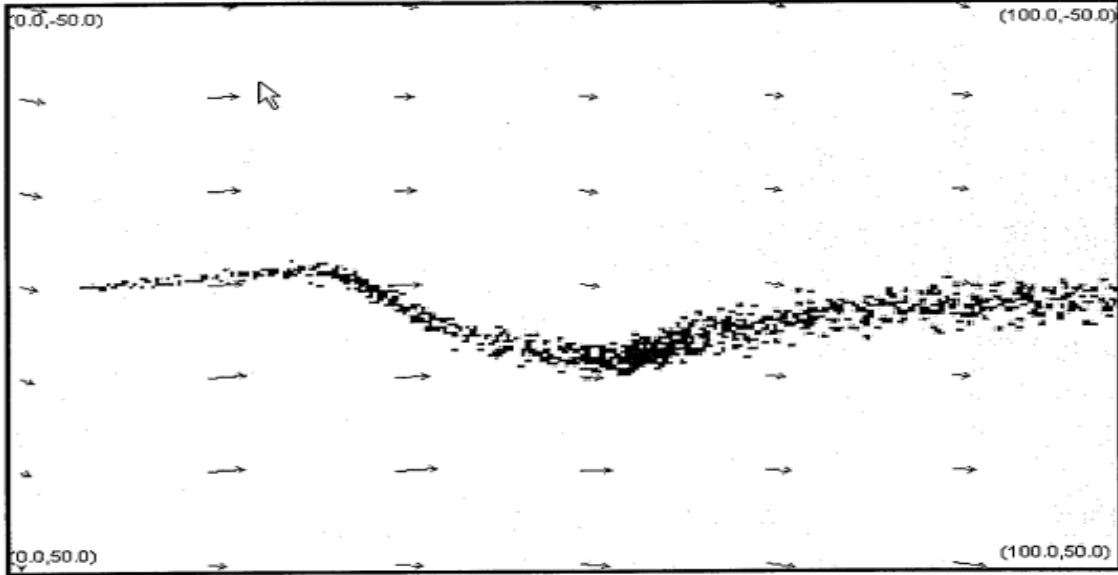


Figure 2.3: Meandering plume with centerline relative diffusion and filament growth. The area represented is 100x100 with the odor source at (5,0). Each arrow indicates the local wind vector at the tail of the arrow.[2]

majority of concentration spikes can be captured at a slower sampling rate at 1 Hz. Indeed, sea creatures such as lobster and *Panulirus Argus* have sampling frequency at 1 Hz or 2 Hz, much slower than the ideal sampling rate 10 Hz [20]. Moreover, Michaelis *et al.*'s plume searching algorithm can successfully navigate along the edge of the plume with the agent moving at speed 0.05 m/s.

2.2 Simulation Design

We adapted the simulation design from both Crimaldi *et al.* and Farrell *et al.* to create our simulation. Our 2D simulation can be understood as the horizontal view of a chemical plume or a slice of a 3D realistic environment. Our water flow velocity has two components: a constant vector and a fluctuation created by Fourier models. We adopted three characteristic structural features from Farrell *et al.*'s simulation. First, the chemical source at a fixed location should have an intermittent structure that closely duplicates experimental observations. However, our chemical source is initialized at different positions in different episodes and does not need to be inside of

the region of interest as Farrell *et al.*'s. Second, the plume created from a continuous release of odor should be sinusoidal and time-varying. Moreover, we still can keep track of the plumes which are out of the region of interest. Third, the plume shape and water flow should be coherent. Instead of defining the water flow vector based on the region, we created turbulent fluctuations which vary within time and space and can be measured regardless of the region. Then, we place our mobile sensor/agent in the center of the interesting region. The agent can travel along a predefined policy. By traveling, it can collect observations at different positions which consists of local chemical concentration measurement and local water flow force measurement. Our simulated chemical source is dispersing at rate 1 Hz. The chemical odor packets are also generated and decayed at rate 1 Hz. Therefore, we kept the same assumption that the sensor with low bandwidth 1 Hz can capture the detail of the plume. Nonetheless, we also converted our simulation unit into real-world units by scaling the simulated agent's speed to a real-world underwater vehicle's speed.

2.2.1 Simulated Agent Design

The agent position l_{agent} is initialized the same in every episode which $l_{agent_{initial}} = (0, 0)$. We assume that the agent movement is not affected by the water flow force. The agent follows a predefined trajectory. At every time step it can move a constant speed $d_{agent} = 5$ units/timestep. We want to make our simulation more realistic by converting all simulation units into real-world units (meter, second). We assume that the sensor with sampling rate 1 Hz can capture the detail of the plume. Therefore we chose the temporal scale at 1 timestep = 1 second. Moreover, we used the agent's speed as a baseline to convert simulation units into meters. Wardlaw and his team at MIT designed a deep ocean underwater glider (DOUG) [21] with reasonable operating costs. Even though DOUG cannot freely move in underwater, it gives us an idea that the real underwater vehicle can reach a speed of 0.5-2 knots or approximately 0.25-1 m/s. Therefore, we adopt the scale of 5 simulation units = 0.5 m (1 unit to 10 cm). At the result, our simulated agent is moving with speed of 0.5 m/s. At every time step, the

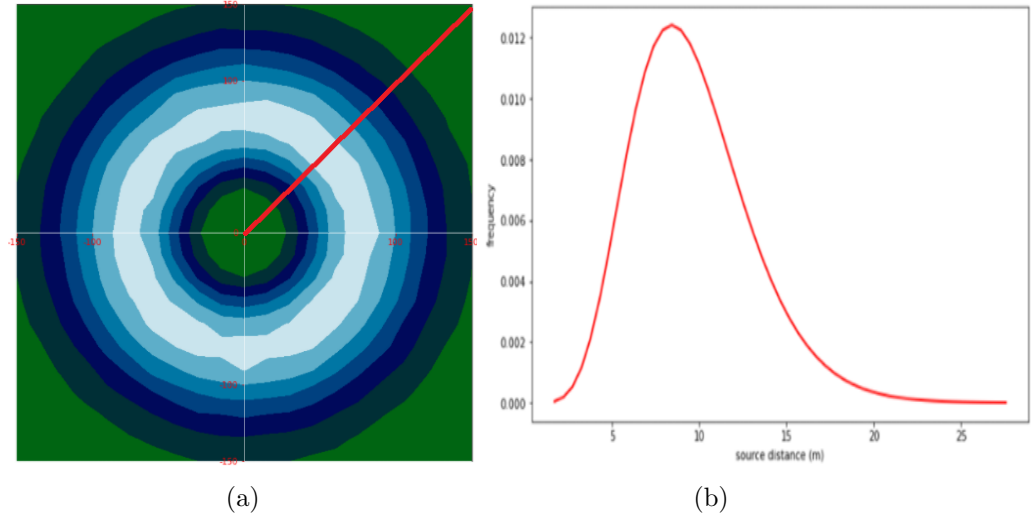


Figure 2.4: (a) the distribution of source location (b) The distribution of the distance between the initialized agent and the source

agent is able to sense odor packets within its range $r_{sense} = 1$ m. We will introduce the sensing process in the next section.

2.2.2 Simulated chemical source

The chemical source is initialized radially symmetric about the agent’s initial position. The position of the source is at a random angle. The distance between initial chemical source and the initial agent is Gamma distributed with shape $\alpha_{source} = 8$ and scale $\alpha_{source} = \frac{1}{12}$. The source location is shown in figure 2.4.

The chemical source concentration is initialized uniformly as $c_{source} \sim U(0.6; 1)$ and is kept fixed during the episode. At every time step, the simulated chemical source produces a random number (possibly zero) of chemical odor packets that mimic a local emission of chemical concentration. The number of generated odor packets $n_{packet} \sim \lambda(\frac{c_{source}}{spawnRate})$. We chose $spawnRate = 0.2$ when generated, every odor packet’s concentration is the same as the chemical source, and is decayed over time $c_{packet_t} = c_{packet_{t-1}} \times decayRate$. We chose $decayRate = 0.97$. The odor packets will be eliminated from the simulation when their concentration decays below the $c_{threshold} = 0.2$. The decay rate is meant to mimic the stability of vortices in the

turbulent flow that maintains a high chemical concentration far from the source. Eventually, vortices decay into smaller vortices when a vortex loses its energy and part of the chemical concentration is dispersed. Each chemical packet will travel based on the local water flow. Let l_{packet_t} denote the position of a chemical packet, $l_{packet_t} = l_{packet_{t-1}} + v_{water}(l_{packet_t}, t)$, where v_{water} is water velocity which varies within space and time. The water velocity function will be introduced in the next section. The simulated agent is able to sense odor packet within its sensing range r_{sense} at every time step $c_{agent_t} = \int_{l_{agent}-r_{sense}}^{l_{agent}+r_{sense}} c_{packet_t}$.

2.2.3 Simulated water flow

The water flow velocity has two components which are a mean water flow V_{water} and turbulent fluctuations $v_f(p, t)$ varying across space and time $v_{water}(p, t) = V_{water} + v_f(p, t)$. The mean water flow is at a random angle. Its magnitude is Gamma distributed with shape $\alpha_{water} = 2$ and scale $\alpha_{water} = \frac{1}{2}$. The source location is shown in figure 2.5. We adopt the stochastic flow using a Fourier mode as the fluctuating component of the velocity field [22]. The flow characterizes by a single scale L , obtained by superimposing eight Fourier modes $\mathbf{k} = (k_x, k_y) \in K = K_1 \cup K_2 = (k_s, 0), (0, k_s) \cup (k_s, \pm k_s)$, where $k_s = \frac{2\pi}{L}$. The stream function Ψ is computed at each odor packet position by $\Psi(p, t) = \sum_{\mathbf{k} \in K} (A(\mathbf{k}, t)e^{i\mathbf{k} \cdot \mathbf{p}} + c.c)$ where *c.c* stands for the complex conjugate. The amplitudes of the Fourier modes $A(\mathbf{k}, t)$ are Gaussian random complex variables evolving with the following Ornstein-Uhlenbeck process.

An example of the simulation within one episode is shown in figure 2.6, in which, the water is flowing from left to right. The initial concentration of the source is 0.8. The agent has a predefined trajectory that repeats going straight from right to left. The odor packets are traveling under the effect of the mean water flow and a sinuous fluctuation. In general, the odor plume has a cone shape.

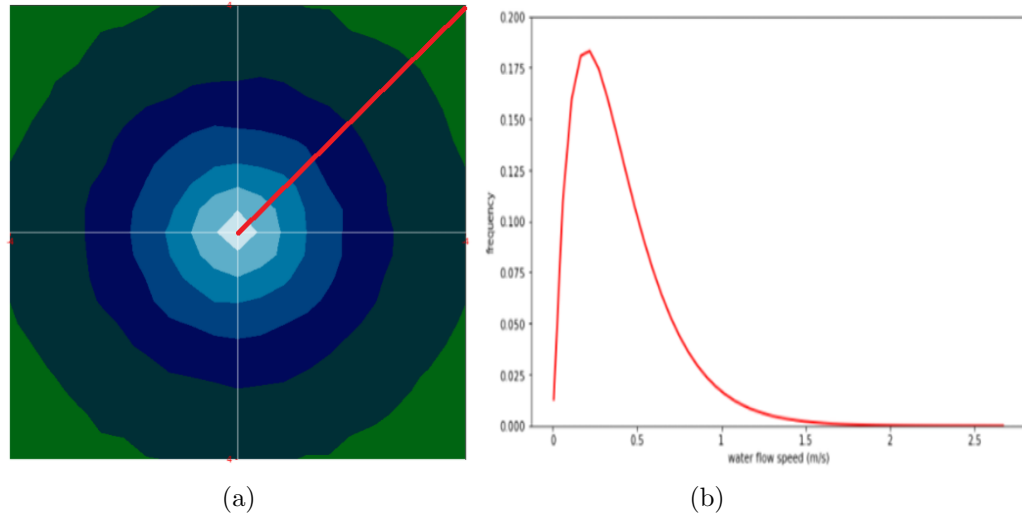
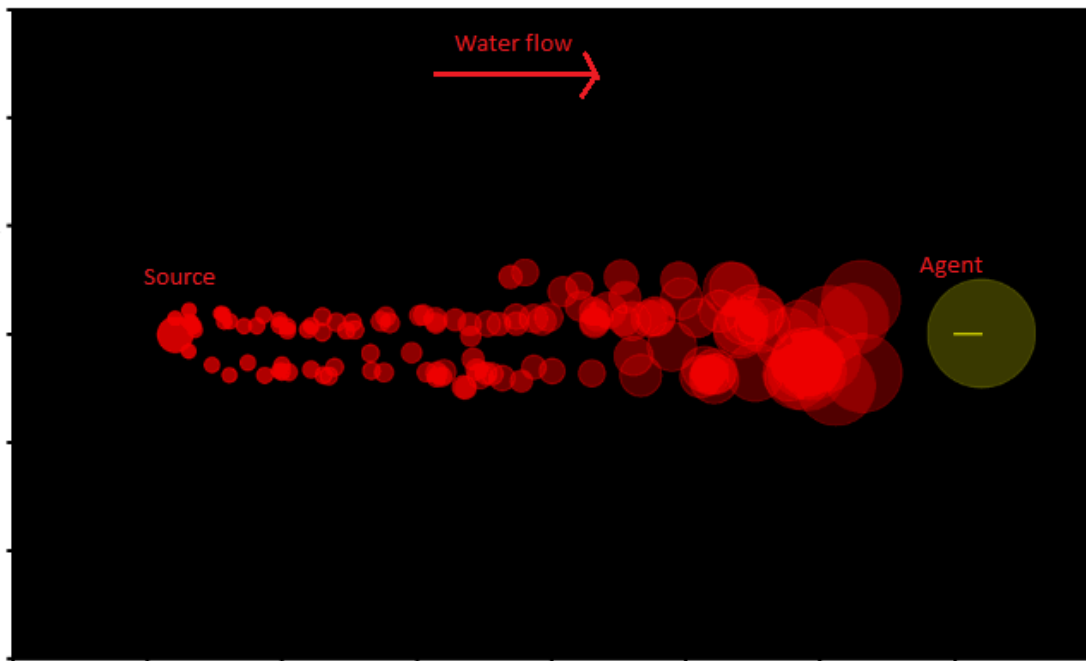


Figure 2.5: (a) The distribution of water flow force (b) The distribution of water flow magnitude



(a)

Figure 2.6: An example of a proposed chemical plume simulation

Chapter 3

SOURCE LOCALIZING MODEL

3.1 Related Work

Many chemical source tracking/tracing with underwater simulation have been introduced [2, 3, 5, 6, 7, 8, 9, 10]. Indeed, olfactory-based search mechanisms [2, 5, 6] are mainly used to detect the chemical plume when the measurement rises above the predefined threshold. Michaelis *et al.* [5] used two sensors to mimic the lobster’s two antennas. The source tracking movement is guided by turning 8° towards the stronger sensor reading. Singh *et al.* [7] applied reinforcement learning to perform the plume tracking task. The agent is given partial observation and a fixed length memory, then trained by a deep deterministic policy gradient [23, 24]. Singh *et al.* stated that the longer the memory is, the more likely the agent is able to return to the plume right track, and have a higher success rate in reaching the source. Moreover, through a reinforcement learning algorithm, the agent can learn to adapt to the sudden change in mean water flow which will change the current shape of the chemical plume.

Source tracking algorithm can only be applied after finding plumes, however, finding plumes is still a challenging problem. Without any prior knowledge, the optimal plume searching strategy is moving cross-wind [11]. Croon *et al.* [25] set up a simulated odor plume searching experiment with a genetic algorithm to analyze the best searching policy. In Croon *et al.*’s simulation, the position of the agent is initialized randomly, while the position of the chemical source is kept fixed. The wind always flows from left to right. Moreover, their environment is bounded, so when the agent goes beyond the boundary it will reappear at another random location. It’s problematic to figure out an optimal search strategy if there is one chemical source in the environment. Instead,

Croon *et al.*'s toroid world implies that the agent is moving in a large environment with an unevenly spaced chemical source. Due to the "toroid world" property, the best search strategy the agent use is steering to the main 90° angle to the wind direction.

Some odor localization research has been done with a probabilistic approach [2, 13, 14, 15]. The goal is to generate a spatial map that keeps track of the probabilities for source presence. The agent's moving trajectory is used to reduce the heatmap's entropy. Farrell *et al.* [2, 13] used Bayesian inference methods to generate the source-likelihood map. The heatmap represents the likelihood of source presence in a region of interest. The heatmap is updated responding to both detection and non-detection events. Kim *et al.*[26] trained long short-term memory (LSTM) [27] to process the time series of observations to localize the leaking gas position. Kim placed 40 sensors at different locations to collect wind direction, wind speed, and gas concentration. The trained LSTM will process the time series and classify them into 41 classes (40 positions and no leakage detection) in real-time.

3.2 Proposed Source Localization Model

We want to train a source localization model which processes the time series of observations to produce the agent-centered heatmap that represent the distribution of source location. Our hypothesis is that, by collecting local chemical concentration and local water flow measurements, the agent can infer the source location. Unlike [2, 5, 6], we do not want to limit our heatmap to any region of interest. Our heatmap can also show the possible direction that the source may appear. Since the heatmap is centered at the agent's current location, we also included the agent's movement in the observations. These observations may consist of only a few positive chemical concentration measurements. The generated heatmap is updated at every timestep corresponding to both detection and non-detection events. The non-detection events help the model eliminate the possible source location. We train a gated recurrent neural network to process the time series of observations, then build the heatmap based on the hidden state of the recurrent heatmap. With the state of art in natural language

processing, neural networks are able to generate images given semantic sentences [28, 29]. The system consists of a recurrent neural network to encode the given sentences, and a convolutional neural network to generate a realistic image given the encoded hidden states. We adopt this model design to encode the time-series of observation and generate the heatmap.

We propose a Wasserstein distance metric loss function to train the heatmap generator model. Since the Wasserstein distance keeps the geometries of distributions in consideration and it does not require the support of distributions to be the same, it provides an additional advantage in comparison to the traditional cross-entropy-based loss function. Additionally, when the source is out of prediction range, the predicted heatmap is still useful to predict the direction of the chemical source location which respects to agent’s current location. However, due to the lack of positive concentration observation within an episode, we want to investigate whether we should generate a heatmap at every time step or just start generating it after an initial positive chemical concentration measurement.

3.2.1 Heatmap generator model

We used an LSTM, a gated recurrent neural network, to process the observation time series to generate the heatmap, which represents the probability that the chemical source is located at different locations. The agent is able to sense the chemical concentration within 1m. By collecting observations $o_t \in R^3$ at different positions, the agent can measure the mean water flow. On the one hand, when the agent receives consecutive zero concentration observations, it can eliminate possible source locations upstream. On the other hand, when it receives positive concentration observations, combined with mean water flow measurement, it can infer the source location upstream within a range of 20 m. The heatmap is separated into $H = n_y^2$ areas to show the potential source locations. The more separated areas yield a higher resolution and more precise heatmap. Since the heatmap is ego-centric to the agent’s current location, the agent has to learn to shift the heatmap based on its movement. Therefore,

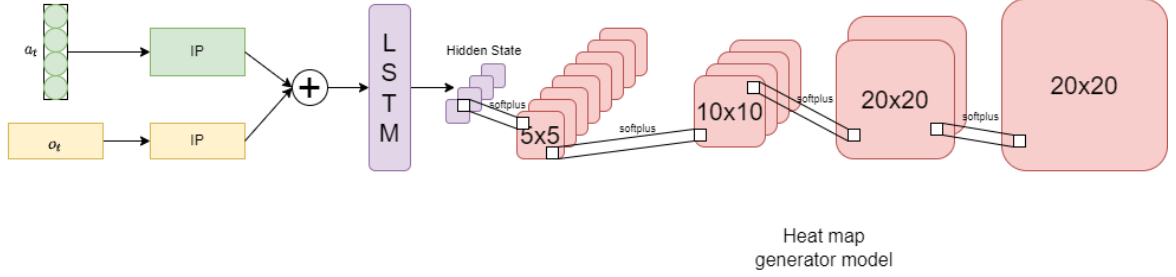


Figure 3.1: Heatmap generator model architecture

we added action variable encoded as a one hot vector of length 4 into the observation time series. The model network is shown in the figure 3.1. Based on the hidden state of the LSTM, we used a convolutional neural network to generate the heatmap. We adapted the architecture from the generator model in a GAN [28]. The hidden state and predicted heatmap are

$$h_t = f_{\theta_1}(o_{1 \rightarrow t}, a_{1 \rightarrow t});$$

and

$$\hat{P}_t = f_{\theta_2}(h_t);$$

respectively, where $\hat{P}_t \in R^{n_y \times n_y}$ is the predicted heatmap of the source location. Heatmap \hat{P}_t has the solution $n_y \times n_y$. θ_1 and θ_2 respectively are the parameters of the LSTM model and the convolutional generator model.

The heatmap is uniformly divided in order to minimize the distance error between the centroid (bin center) of the predicted area and the source location when the source is close and to minimize the angular error between the centroid of the predicted area and the source location when the source is far away. The distance and angular error are calculated as shown in figure 3.2. When the prediction of the source is in the central area, the angular error cannot be calculated accurately and the angular error of the model when the source is close to the agent will be high. Alternatively, we considered taking the central area prediction as 0° degree error. However, if the model keeps making random predictions that the source is in the central area, it will have a low expected angular error, which is not true. We also considered separating the heatmap into non-uniform Voronoi regions, where the distal areas are larger than the

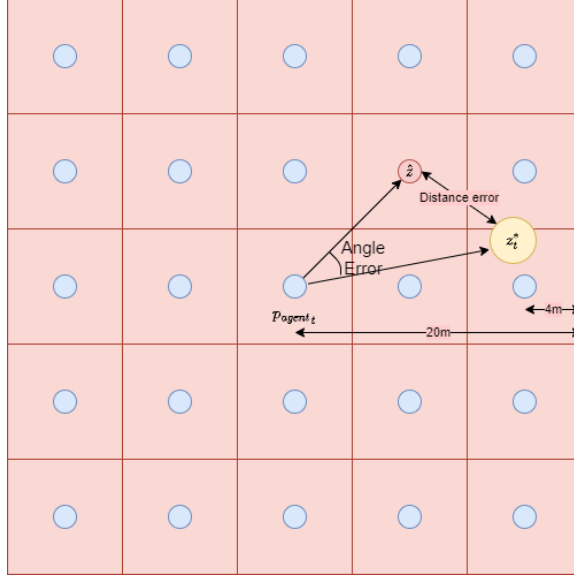


Figure 3.2: Visualization of how the distance and angular error is calculated

proximal ones. With this approach, we can minimize the angular error regardless of whether the source is close or not. However, in practice, the angular error is useful to tell which direction the source is with respect to the current location when the source is out of predicting the range. When the source is inside of the predicted range, we want an accurate location of the source. Therefore, we decided to maintain a uniform grid for the heatmap.

3.2.2 Wasserstein distance-based loss function

The model is trained on different episodes e generated from the simulation $e \in p_{data}$. All episode has the same length T . We proposed a Wasserstein distance-based loss function to train our models to minimize the expected distance between the predicting and true source location across an episode

$$L(\theta_1, \theta_2) = \mathbb{E}_{e \in p_{data}} \left(\frac{1}{T} \sum_t \sum_{z \in G} \hat{P}_t(z) \times \|z - z_t^*\|^2 \right), \quad (1)$$

where z is the centroid of each separated areas, and z_t^* is the relative source location to current position of agent $z_t^* = l_{sourceinitialize} - l_{agent_t}$. Compared to traditional cross-entropy loss, even when the prediction is wrong, our Wasserstein loss can be informative

in predicting the true source location. As stated above, most of the observations have zero sensing concentration. Therefore applying the loss across all the time steps may produce a bias. We modified the loss function to evaluate after an initial non-zero chemical concentration observation as

$$L(\theta_1, \theta_2) = \mathbb{E}_{e \in p_{data}} \left(\frac{\sum_t^T \sum_{z \in G} \hat{P}_t(z) \times \|z - z_t^*\|^2 \times \mathbb{1}_t}{\sum_t^T \mathbb{1}_t} \right), \quad (2)$$

and

$$\mathbb{1}_t = \begin{cases} 1 & \text{if } t = T, \text{ or } t > t' \text{ where } c_{t'} > 0 \\ 0 & \text{Otherwise} \end{cases}$$

Moreover, in case the agent does not receive any positive concentration observation, we do not want the model is too confident on its prediction. Therefore, we add a entropy regularizer to which penalize the model's confidence

$$\Omega(\hat{P}_t(z)) = \lambda \sum_{z \in G} \hat{P}_t(z) \log \hat{P}_t(z),$$

where λ is the regularization constant.

In the next experiment, we want to compare the performance of two loss functions. Based on the result, we can conclude that evaluating the heatmap given the plume detected can help increase the accuracy of the model.

Chapter 4

EXPERIMENT

In this section, we generate various episodes from the proposed simulation to train and test our proposed chemical source localization model. In each episode, the source is initialized at different positions with different initial concentrations. The water flow is also initialized with different directions and magnitudes. The agent starts at the same position and is free to move in 4 directions. Since the model performance depends on the trajectory of the agent, we first train our model with a stochastic policy. Then we test and compare this model to other movement policy datasets to see if the model can generalize and understand the alternative trajectories. Moreover, we will also test the correlation between the model accuracy across different choices of the heatmap resolution. When the agent receives a detection, a positive concentration observation, we expect our model to localize the chemical source, and remember this positive as it keeps moving. However, if the agent does not receive any detection yet, it is difficult for the agent to make an accurate prediction. We also test the two loss function proposed above, to see whether it's necessary to generate a heatmap only after the agent receives a positive concentration observation. The model that uses the first loss function will be trained every timestep of the simulated training data. On the other hand, the model that uses the second loss function will only be trained on the timesteps after detection.

4.1 Stochastic policy movement dataset

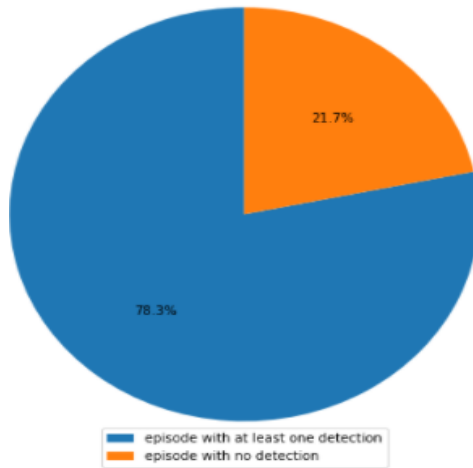
We simulated 22,000 episodes of length 120 time steps (2 minutes) as a dataset to train our model. In each episode, the agent is moving with a stochastic policy where it has a 70% chance to keep the same direction, a 10% chance to turn 90 degrees, a

10% chance to turn -90 degrees, and a 10% chance to reverse its current direction. There are 17,229 episodes (78% of episodes) that the agent detects plume as shown in figure 4.1a. Across every timestep and episode, the chance that the agent receives a positive concentration observation is 0.08%. We separate the training data into two types: the timesteps before and after receiving a detection as shown in figure 4.1b. The episode where the agent doesn't receive any positive concentration observation will be considered as timesteps before detection. The distance between the agent and the chemical source at each timestep is shown in figures 4.1c, 4.1d. In general, the distribution of the agent's location is shown in figure 4.2a. To better visualize the training dataset, we also show the agent's location distribution with respect to source location and water flow direction. For the source perspective, we rotate and shift the coordinate to make the source place at origin as shown in figure 4.2c. From a water flow perspective, we rotate the coordinate to make the water flow direction always $[-1, 0]$ as shown in figure 4.2c.

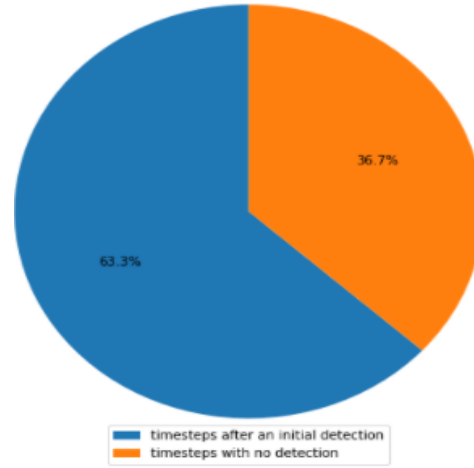
4.2 Comparison of two loss function versions

We used 80% of the episodes as training data and the rest as testing data. We evaluate the model's performance by the expected distance and angular error against the distance between the current agent's position and the source location. Moreover, if the source is outside of predicting range, the distance error is not really informative. Therefore, we only evaluate the distance error when the source is inside the predicting range. Moreover, we also want to see the relationship between the resolution of the predicted heatmap and the accuracy of the models. As shown in the figure 3.2, with the heatmap of resolution 5×5 , we expect the distance error should be in the range $[0, 4\text{m}]$ if the source is inside predicting range, and the angular error should be in the range $[0, 11.25^\circ]$. As we increase the heatmap resolution to 10×10 , 20×20 , we expect the expected distance error and expected angular error would be reduced.

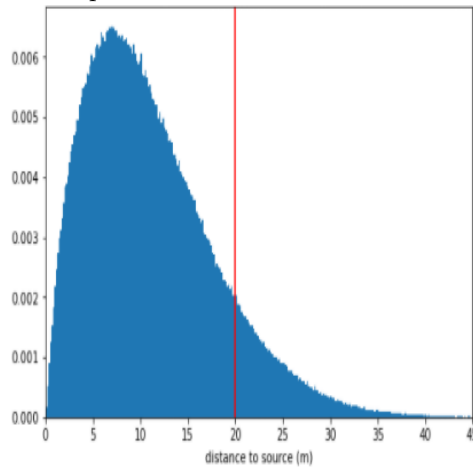
We train our model with the loss 1, which generates and evaluates the heatmap at every timestep, then calculate the expected distance error and angular error of the



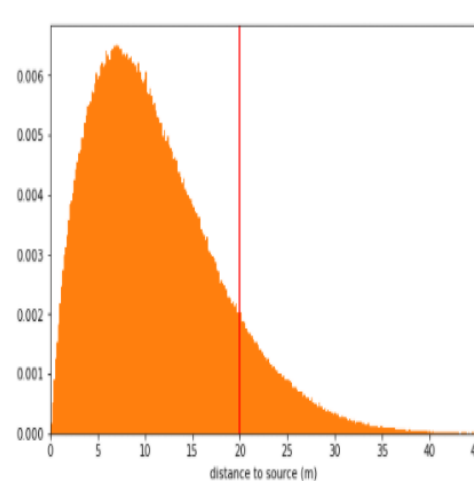
(a) the percentage of episodes with detected plume



(b) the percentage of timesteps after and before the detection event



(c) the distribution of the distance between agent and source of timesteps after detection event. The vertical red line shows whether the source is inside the predicting range.



(d) the distribution of the distance between agent and source of timesteps before the detection event. The vertical red line shows whether the source is inside the predicting range.

Table 4.1: Stochastic movement policy dataset analyzing

model. The performance of the model is shown in figure 4.3, where the x-axis is the distance between the agent and the source, and the y-axis is the expected distance/angular error. According to figures 4.3a and 4.3c, with heatmap resolution of 5×5 , if the agent receives an initial positive concentration observation, the expected distance error is around 4 m if the source is inside predicting range and the expected angular error is around 11.25° if the source is outside. We can say the model can localize the source

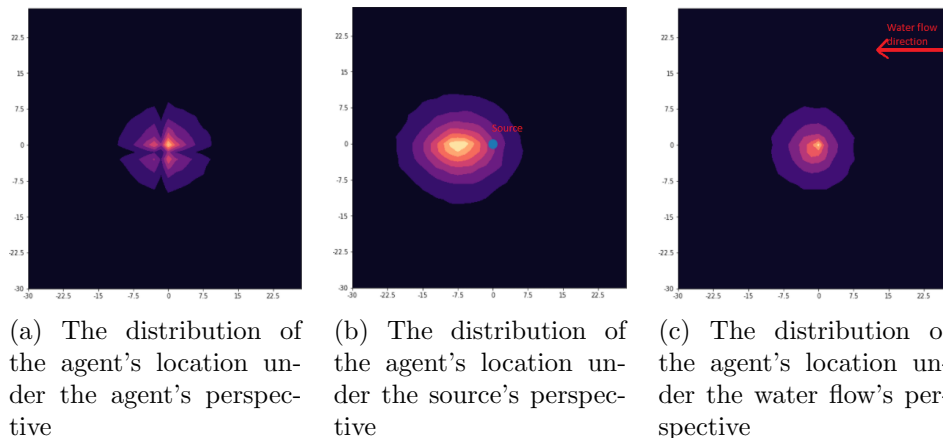
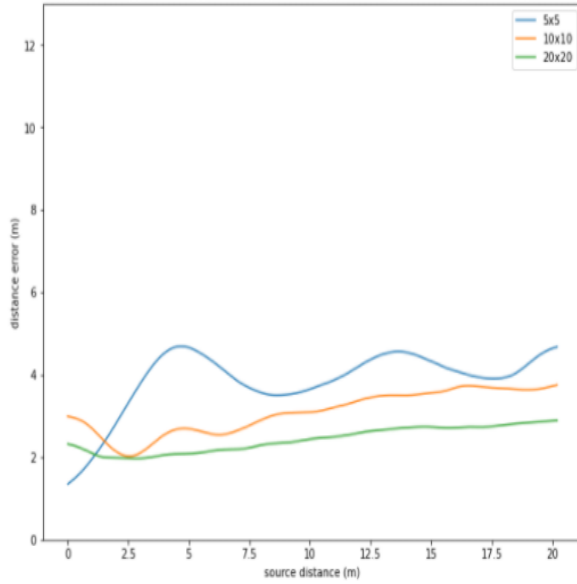


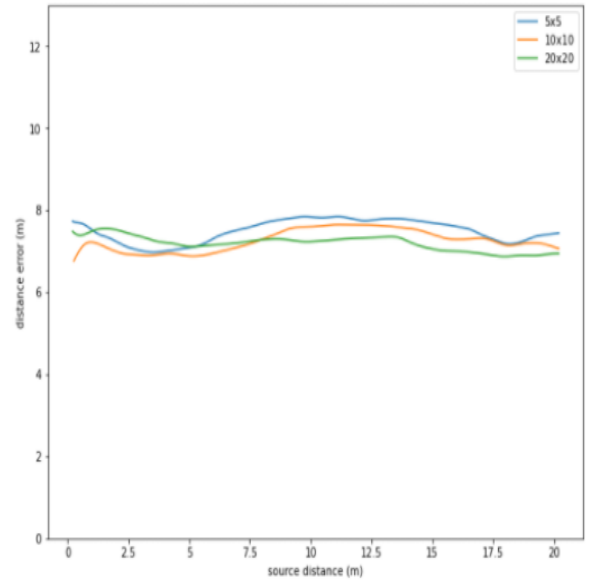
Table 4.2: Stochastic movement policy dataset visualization

well with a heatmap resolution of 5×5 . As we increase the heatmap resolution, there is a slight reduction in the distance/angular error. Contrastively, according to figures 4.3b and 4.3d, if the agent does not receive any positive concentration observation, regardless of heatmap resolution, the expected distance error is around 7 m if the source is inside, and the expected angular error is still around 11° if the source is outside. However, when the agent doesn't receive any positive concentration observation, there is no difference in performance for different resolution heatmaps. Moreover, when the source is inside and close to the agent, even though the distance error is acceptable, the angular error is high. For example, when the agent comes very close to the source, the angle between consecutive areas is also high. Therefore, this model may not be good to show the agent which direction it should move to find the chemical source.

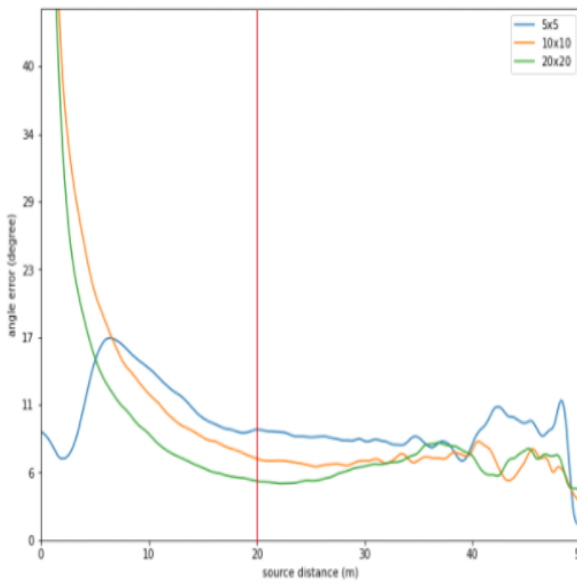
Subsequently, we test if only evaluating the predicting heatmap when receiving an initial positive concentration observation will help the model improve its performance. We trained another model which uses the loss function 2, which only optimizes the heatmap when the agent receives an initial positive concentration. Then, we calculate the expected distance error and angular error of this model and compare to the previous results. The performance of the second model is shown in figure 4.4. According to the results shown in figures 4.4a and 4.4c, when trained on timesteps after



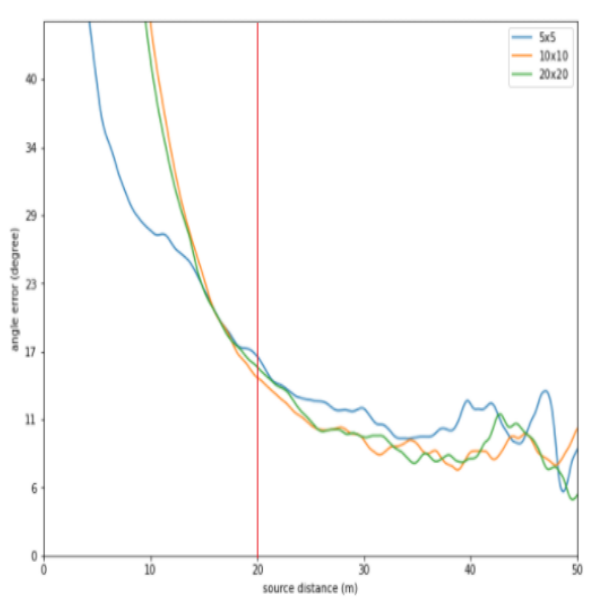
(a) Average predicted heatmap distance error versus the distance between the agent and the source when the agent receives a detection event.



(b) Average predicted heatmap distance error versus the distance between the agent and the source when the agent doesn't receive any detection event.



(c) Average predicted heatmap angular error versus the angular between the agent and the source when the agent receives a detection event.



(d) Average predicted heatmap distance error versus the angular between the agent and the source when the agent doesn't receive any detection event.

Table 4.3: Model performance on stochastic policy movement dataset using loss 1.

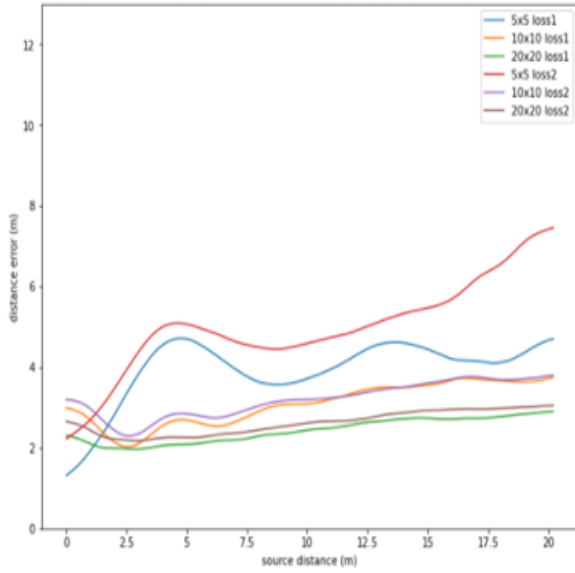
receiving an initial positive concentration observation, the model has worse performance compared to the one trained on every timestep, especially in the low-resolution

heatmap. As we increase the heatmap resolution, there is no difference in performance between the two models. On the other hand, when we test two models on the timesteps when the agent doesn't receive any positive concentration observation, the model with loss function 2 performs worse than the one with loss function 1 since it's not trained to deal with these cases. Therefore, we conclude that the loss function 1 is superior.

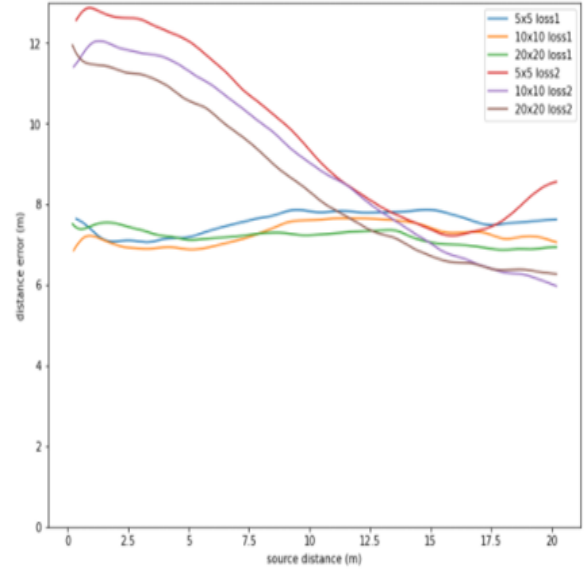
4.3 Different policy dataset

In this section, we want to test the model trained on the stochastic movement dataset to better understand the effect of the agent's trajectory on the source localization. We simulated episodes where the agent is controlled with different policies. We chose spiral policy, where the agent will take a 90° after a fixed amount of steps, and onward policy, where the agent commits to one direction and keeps moving. We want to test whether the model can understand rotation action and localize itself in the environment. We trained specific models on these datasets, then compare their performances to the stochastic model on these dataset. Just like with the stochastic trajectory dataset, we will analyze the spiral dataset and onward dataset by showing the percentage of episodes where the agent detects the plume, the percentage of timesteps after receiving a detection, and the agent position distribution under agent, source, and water flow perspective as shown in figure 4.5. For all models, we will use the loss function 1 and produce heatmap predictions with the resolution of 5x5. The performance comparison of each model is shown in figure 4.4.

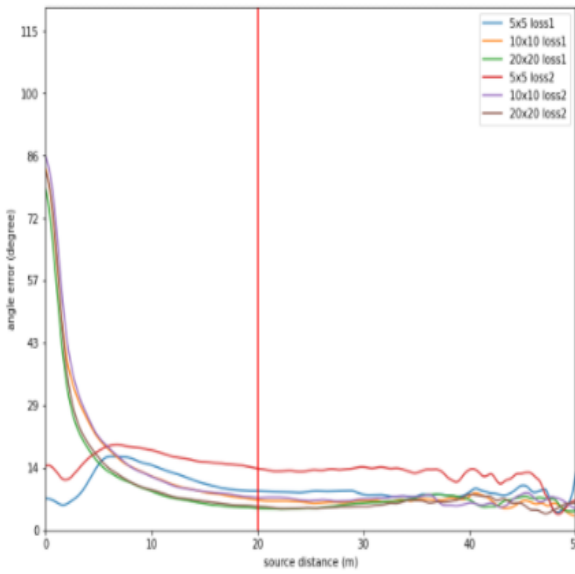
We trained three models respectively on the spiral, stochastic, and onward datasets, then tested them on each dataset. As the results shown in figure 4.6, the stochastic model performed as well as or even better than other models trained on their own dataset. One thing we noticed is that the distance error of the spiral model increases linearly as the agent goes far away from the source. If the agent does not receive any detection, the model will predict the source location in the central area. As mentioned, we don't count these predictions into calculating angular error. With



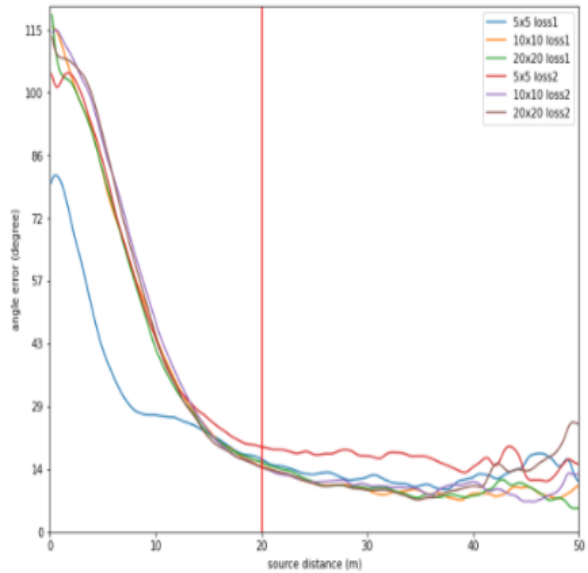
(a) Average predicted heatmap distance error versus the distance between the agent and the source when the agent receives a detection event.



(b) Average predicted heatmap distance error versus the distance between the agent and the source when the agent doesn't receive any detection event.



(c) Average predicted heatmap angular error versus the angular between the agent and the source when the agent receives a detection event.

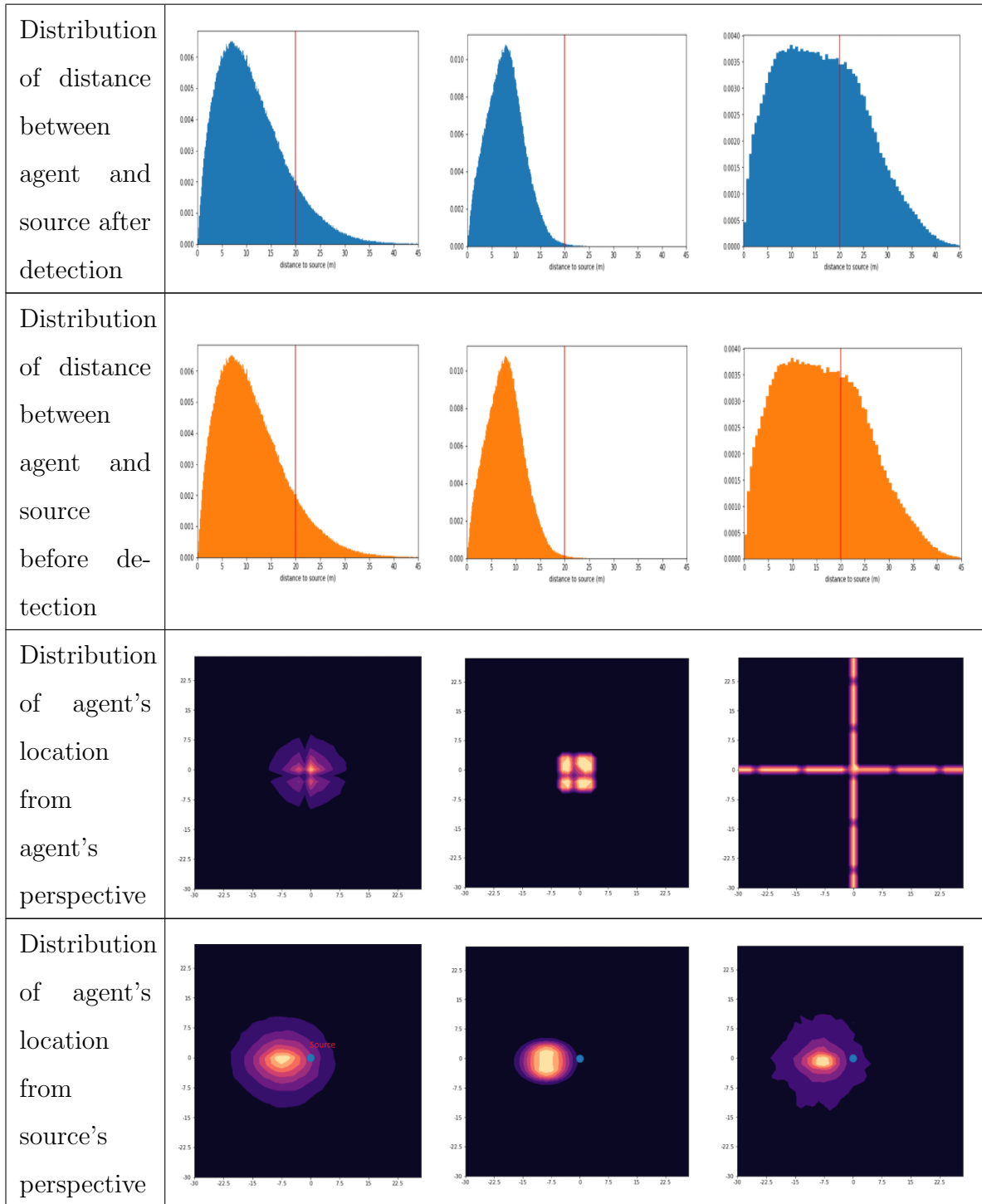


(d) Average predicted heatmap distance error versus the angular between the agent and the source, when the agent doesn't receive any detection event.

Table 4.4: Model performance comparison on stochastic policy movement dataset using loss 1 and loss 2.

a spiral dataset, we want to test whether the model can understand the agent’s trajectory when the agent makes multiple turning actions. With the onward dataset, we want to test whether the agent can localize itself when the agent doesn’t make any turning action and just keeps moving forward. Since the stochastic model has a good performance compared to another model, we will continue to use the stochastic model as a navigator in simulation.

	Stochastic Dataset	Spiral Dataset	Onward Dataset												
# episode	22000	12000	10000												
Episode length	120	120	40												
Detection chance	11%	9%	3%												
% episodes with plume detection	<table border="1"> <tr> <td>episode with at least one detection</td> <td>78.3%</td> </tr> <tr> <td>episode with no detection</td> <td>21.7%</td> </tr> </table>	episode with at least one detection	78.3%	episode with no detection	21.7%	<table border="1"> <tr> <td>episode with at least one detection</td> <td>55.9%</td> </tr> <tr> <td>episode with no detection</td> <td>44.1%</td> </tr> </table>	episode with at least one detection	55.9%	episode with no detection	44.1%	<table border="1"> <tr> <td>episode with at least one detection</td> <td>31.3%</td> </tr> <tr> <td>episode with no detection</td> <td>68.7%</td> </tr> </table>	episode with at least one detection	31.3%	episode with no detection	68.7%
episode with at least one detection	78.3%														
episode with no detection	21.7%														
episode with at least one detection	55.9%														
episode with no detection	44.1%														
episode with at least one detection	31.3%														
episode with no detection	68.7%														
% timesteps before and after detection	<table border="1"> <tr> <td>timesteps after an initial detection</td> <td>63.3%</td> </tr> <tr> <td>timesteps with no detection</td> <td>36.7%</td> </tr> </table>	timesteps after an initial detection	63.3%	timesteps with no detection	36.7%	<table border="1"> <tr> <td>timesteps after an initial detection</td> <td>35.8%</td> </tr> <tr> <td>timesteps with no detection</td> <td>64.2%</td> </tr> </table>	timesteps after an initial detection	35.8%	timesteps with no detection	64.2%	<table border="1"> <tr> <td>timesteps after an initial detection</td> <td>28.6%</td> </tr> <tr> <td>timesteps with no detection</td> <td>71.4%</td> </tr> </table>	timesteps after an initial detection	28.6%	timesteps with no detection	71.4%
timesteps after an initial detection	63.3%														
timesteps with no detection	36.7%														
timesteps after an initial detection	35.8%														
timesteps with no detection	64.2%														
timesteps after an initial detection	28.6%														
timesteps with no detection	71.4%														



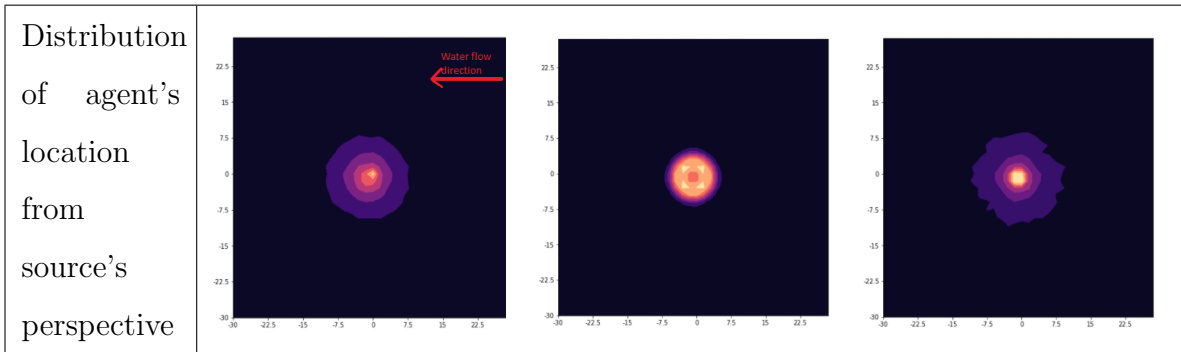
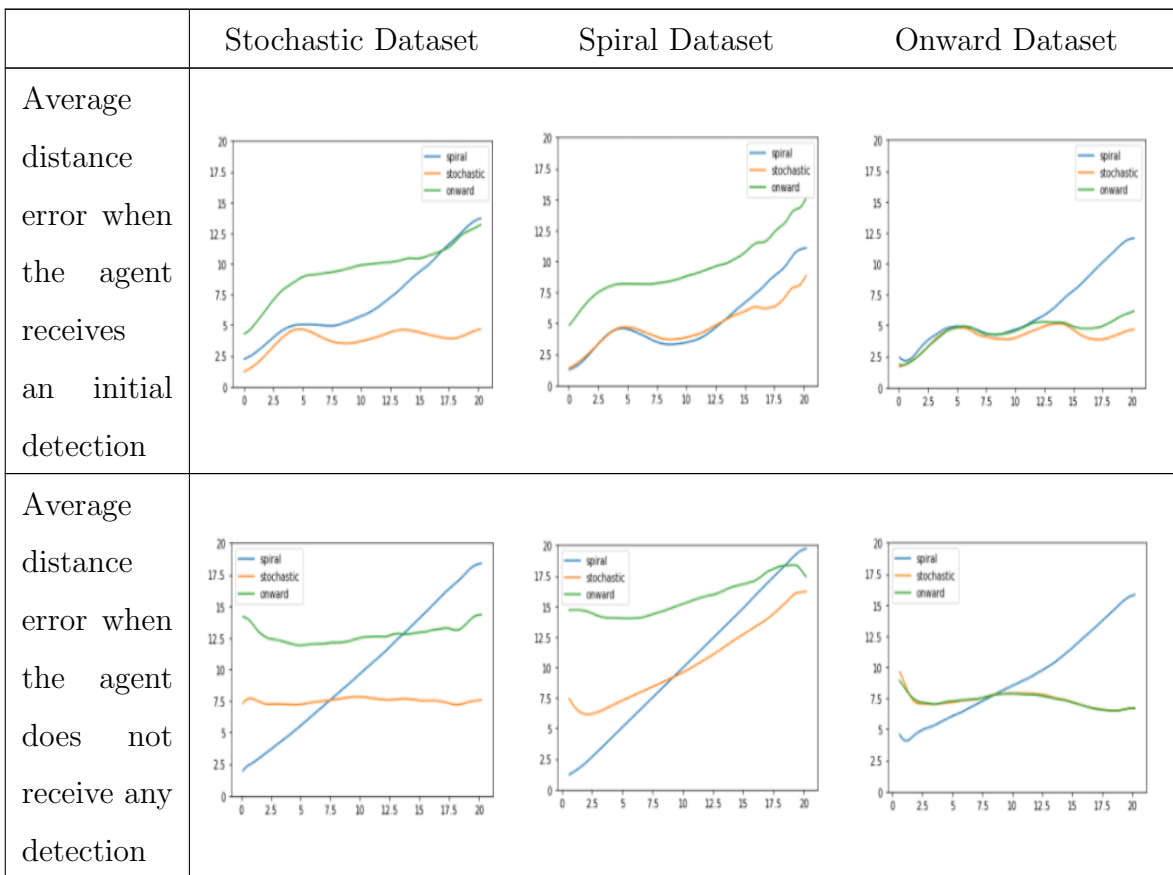


Table 4.5: Different policy dataset analysis



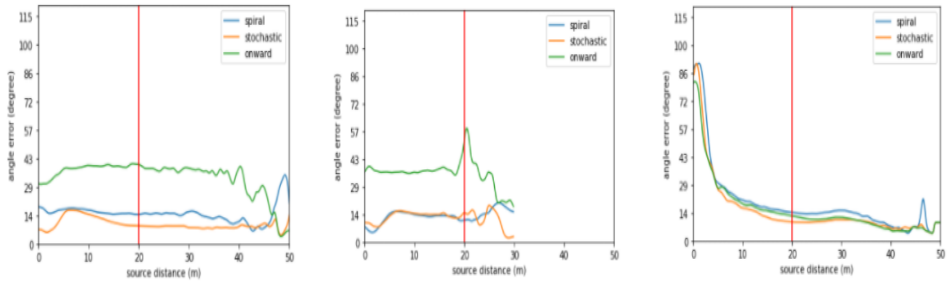
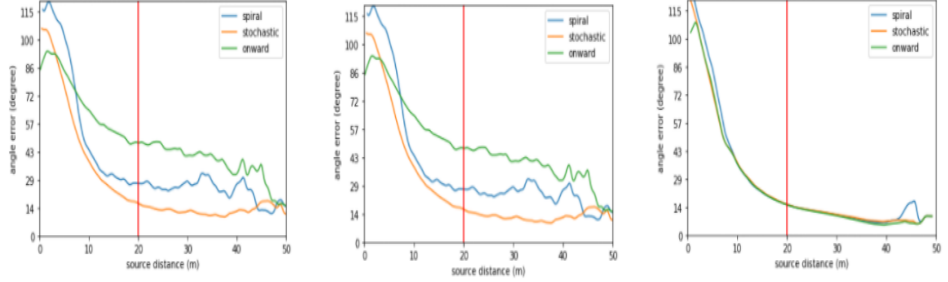
<p>Average angular error when the agent receives an initial detection</p>	 <p>The top row contains three line graphs. Each graph plots 'angle error (degree)' on the y-axis (0 to 115) against 'source distance (m)' on the x-axis (0 to 50). A vertical red line is at 20m. The legend indicates three models: spiral (blue), stochastic (orange), and onward (green). In all three graphs, the error is relatively low and stable, generally staying below 45 degrees, with a slight increase or spike at the 20m mark.</p>
<p>% Average angular error when the agent does not receive any initial detection</p>	 <p>The bottom row contains three line graphs with the same axes and legend as the top row. In these graphs, the error starts very high (around 115 degrees) at 0m and decreases significantly as distance increases. The 'onward' model (green) generally maintains the highest error, while the 'stochastic' model (orange) maintains the lowest. All models show a sharp drop in error before the 20m mark and then stabilize between 14 and 45 degrees.</p>

Table 4.6: Cross testing of models trained a stochastic, spiral, and onward datasets

Chapter 5

DISCUSSION

This thesis proposes an alternative underwater chemical plume simulation and a source localization model which achieves a good performance. Our proposed simulation has a cheaper computational cost than Farrell *et al.*'s simulation's but still satisfies their three requirements for under water chemical simulation [2]. Our simulated water flow varies within space and time and is coherent to the chemical plume shape. Moreover, our simulation is duplicable to generate various scenarios to test the source localization model. Unlike Farrell *et al.*'s simulation, we do not bound our simulation. We can keep track of the agent and chemical source when they are outside of the region of interest, and measure the water flow at any position. Therefore, the computational cost does not scale as we increase the size of the environment. There are many source tracking methods that achieve a high source-seeking rate. However, the plume finding is still problematic. Our proposed source localizing model can achieve accurate prediction if the agent can detect the presence of the chemical plume. On the other hand, without any initial positive concentration observation, the model can eliminate the potential position of the chemical source and reach acceptable performance. Either way, the predicted heatmap is still informative about the direction in which the agent should go to find the source. We also show that our model can process and understand the movement of the sensor platform. In testing, the model trained on stochastic movement policy can achieve a good performance on other data sets such as spiral where, the agent frequently takes 90⁰ turn, onward, where the agent keeps moving forward. Moreover, the model trained on stochastic policy can outperform models trained specifically on spiral and onward datasets. Therefore, we will continue to use the stochastic model as a navigator in future work.

5.1 Limitation

The search for the presence of the chemical plume, which decides the performance of the source localization model, is still problematic. As shown in the experiment, the gap between the model’s performance with and without plume detection is significant. In reality, the percentage of episodes with at least one positive concentration observation is not as high as we used in the generated dataset. The stochastic movement is not an ideal policy to search for the plume. We used stochastic movement for the purpose of training the model to understand the movement of the sensor platforms. With a better plume search policy, the accuracy of the heatmap will be increased overall. For example, if we want to search for a specific region, the space-filling curve could be a better option [30]. Otherwise, without any prior knowledge, the best strategy for plume search is moving across the direction of flow [11].

5.2 Future work

In the future, we want to analyze the dependency of the sensor’s sampling rate to our model’s performance. In our simulation, our source’s dispersing rate and agent’s sampling rate are the same as 1 Hz. In fact, sea creatures also have a low sampling rate, only 1-2 Hz [5]. Michaelis *et al.*’s works showed that the majority of concentration spikes can be captured at 1 Hz. The concentration intermittency calculated at a sampling rate of 2 Hz and 5 Hz is effectively equivalent to the ideal sampling rate of 10 Hz. In future works, we will increase the sampling of agents and run the test again. Since we scale our simulation’s units to realistic units based on the agent’s velocity, changing the temporal resolution will lead to spatial resolution. In other words, our simulation will be smaller, and both agent’s sensing range and predicting range will shrink.

We also want to test our model on a realistic dataset. In order to avoid the risk of using physical hardware, we plan to test our localization model on the PLIF imaging system data. The PLIF images recorded the realistic movement of water flow and the chemical plume. We can use consecutive PLIF images as a simulation to test our model. In this experiment, the local concentration can be measured as the total

value of neighbor pixels. Moreover, analyzing the PLIF images can help improve our simulation to be more realistic.

In the situation of no plume detection, we want to introduce regularization to reduce the confidence of the model. The regularization is formulated as $\Omega(\hat{P}_t(z)) = \lambda \sum_{z \in G} \hat{P}_t(z) \log \hat{P}_t(z)$ where λ is the regularization constant. By maximizing the entropy of the heatmap prediction, we can reduce the agent's confidence in its prediction when it does not detect the plume yet.

In a realistic environment, there may be measurement errors, such as local concentration measurement or agent movement, can happen that may hurt the model performance. Indeed, the agent's velocity is not fixed but affected the water flow. Therefore, depending on the heatmap resolution, there may be a minimal shift in prediction. Since the observation also includes the local measurement of local flow, we can analyze its impact on the real sensing platform to correct predicting heatmap. Most types of sensing platforms only work when the chemical molecules are brought to the surfaces of the sensors [31]. However, the number of chemical molecules reaching the surface of the sensor significantly decreases when the sensor is slightly off from the chemical distribution. Moreover, the realistic sensors need an amount of time to absorb the local chemical molecules. Depending on how long the sensor takes, the measurement may not be exactly the local chemical concentration. Another analysis must be done to deal with the local concentration measurement error.

Chapter 6

CONCLUSION

Underwater chemical source localization in an environment with turbulent flow is an ill-posed problem. We developed an underwater simulation to represent the movement of the chemical molecules under the chaotic water flow. Our proposed chemical source localization model trained on our simulation achieved a good performance. The proposed model can process the sequence of local measurements of chemical concentration and water flows along with the movement of the agent to produce a heatmap representing the potential local of the chemical source. Even though the source localization depends on the plume search, lacking a concentration signal does not mean lacking information. Our model can eliminate the potential location of the source where it does not receive a concentration signal. We also demonstrated the accuracy of our localization model under the distance error and angular error between the predicting and the true source location.

BIBLIOGRAPHY

- [1] John Crimaldi, Megan Wiley, and Jeffrey Koseff. The relationship between mean and instantaneous structure in turbulent passive scalar plumes. *Journal of Turbulence*, 3:27–29, 07 2002.
- [2] Jay Farrell, John Murlis, Xuezhong Long, Wei Li, and Ring Cardé. Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes. *Environmental Fluid Mechanics*, 2:143–169, 06 2002.
- [3] Gideon Kowadlo and R. Russell. Robot odor localization: A taxonomy and survey. *The International Journal of Robotics Research*, 27:869–894, 08 2008.
- [4] Michael D. Manson and William Margolin. Howard berg’s random walk through biology. *Journal of Bacteriology*, 202(23):e00494–20, 2020.
- [5] Brenden Michaelis, Kyle Leathers, Yuriy Bobkov, Barry Ache, Jose Principe, Raheleh Baharloo, Il Park, and Matthew Reidenbach. Odor tracking in aquatic organisms: the importance of temporal and spatial intermittency of the turbulent plume. *Scientific Reports*, 10:7961, 05 2020.
- [6] Jay Farrell, Shuo Pang, and Wei Li. Chemical plume tracing via an autonomous underwater vehicle. *Oceanic Engineering, IEEE Journal of*, 30:428 – 442, 05 2005.
- [7] Satpreet Singh, Floris van Breugel, Rajesh Rao, and Bingni Brunton. Understanding biological plume tracking behavior using deep reinforcement-learning. volume ALIFE 2020: The 2020 Conference on Artificial Life of *ALIFE 2022: The 2022 Conference on Artificial Life*, pages 750–752, 07 2020.
- [8] Massimo Vergassola, Emmanuel Villermaux, and Boris Shraiman. ‘infotaxis’ as a strategy for searching without gradients. *Nature*, 445:406–9, 02 2007.
- [9] Jiu Haifeng, Chen Yu, Deng Wei, and Pang Shuo. Underwater chemical plume tracing based on partially observable markov decision process. *International Journal of Advanced Robotic Systems*, 16, 2019.
- [10] Mihir Durve, Lorenzo Piro, Massimo Cencini, Luca Biferale, and Antonio Celani. Collective olfactory search in a turbulent environment. *Physical Review E*, 102 1-1:012402, 2020.

- [11] R. Andrew Russell, Alireza Bab-Hadiashar, Roderick Shepherd, and Gordon G. Wallace. A comparison of reactive robot chemotaxis algorithms. *Robotics and Autonomous Systems*, 45:83–97, 2003.
- [12] Andrew M. Hein and Scott A. McKinley. Sensing and decision-making in random search. *Proceedings of the National Academy of Sciences*, 109:12070 – 12074, 2012.
- [13] S. Pang and J.A. Farrell. Chemical plume source localization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(5):1068–1080, 2006.
- [14] Eduardo Moraud and Elisabetta Chicca. Toward neuromorphic odor tracking: Perspectives for space exploration. *Acta Futura*, 4:9–19, 01 2011.
- [15] Eduardo Martin Moraud and Dominique Martinez. Effectiveness and robustness of robot infotaxis for searching in dilute conditions. *Frontiers in Neurorobotics*, 4, 2010.
- [16] R.L. Woodfin. *Trace Chemical Sensing of Explosives*. 05 2006.
- [17] Víctor Pomareda, Rudys Magrans, Juan Manuel Jiménez-Soto, Dani Martínez, Marcel Tresánchez, Javier Burgués, Jordi Palacín, and Santiago Marco. Chemical source localization fusing concentration information in the presence of chemical background noise. *Sensors*, 17:904, 04 2017.
- [18] Ryuichi Takemura, Hiromi Sakata, and Hiroshi Ishida. Active chemical sampling system for underwater chemical source localization. *Journal of Sensors*, 2016:1–11, 02 2016.
- [19] Petr Denissenko, Sergei Lukaschuk, and Thomas Breithaupt. The flow generated by an active olfactory system of the red swamp crayfish. *Journal of Experimental Biology*, 210:4083 – 4091, 2007.
- [20] J.A. Goldman and M.A.R. Koehl. Fluid Dynamic Design of Lobster Olfactory Organs: High Speed Kinematic Analysis of Antennule Flicking by *Panulirus argus*. *Chemical Senses*, 26(4):385–398, 05 2001.
- [21] Mike Wardlaw. Deep ocean underwater glider. <https://www.dropbox.com/s/0x3ss3farb0591i/2022%20-%20202.014%20Final%20Presentation.pdf?dl=0>, 2022. Accessed: 2022-07-18.
- [22] L. Biferale, J. Bec, G. Boffetta, A. Celani, M. Cencini, A. Lanotte, S. Musacchio, and F. Toschi. Inertial particles in turbulence. In Martin Oberlack, George Khujadze, Silke Günther, Tanja Weller, Michael Frewer, Joachim Peinke, and Stephan Barth, editors, *Progress in Turbulence II*, pages 207–212, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

- [23] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Beijing, China, 22–24 Jun 2014. PMLR.
- [24] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *ArXiv*, abs/1509.02971, 2016.
- [25] Guido Croon, L.M. O’Connor, C. Nicol, and Dario Izzo. Evolutionary robotics approach to odor source localization. *Neurocomputing*, 121:481–497, 12 2013.
- [26] Hyunseung Kim, Myeongnam Park, Chang Won Kim, and Dongil Shin. Source localization for hazardous material release in an outdoor chemical plant via a combination of lstm-rnn and cfd simulation. *Computers Chemical Engineering*, 125:476–489, 03 2019.
- [27] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.
- [28] Scott E. Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *ArXiv*, abs/1605.05396, 2016.
- [29] Kaiqin Hu, Wentong Liao, Michael Ying Yang, and Bodo Rosenhahn. Text to image generation with semantic-spatial aware gan. *ArXiv*, abs/2104.00567, 2021.
- [30] Siddharth H. Nair, Arpita Sinha, and Leena Vachhani. Hilbert’s space-filling curve for regions with holes. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 313–319, 2017.
- [31] David Brock Dusenbery. Sensory ecology: How organisms acquire and respond to information. W. H. Freeman, New York, NY, USA, 1992.