sensors

MDPI

Article

Variational Autoencoder for the Prediction of Oil Contamination Temporal Evolution in Water Environments

Alejandro Casado-Pérez *, Samuel Yanes, Sergio L. Toral, Manuel Perales-Esteve and Daniel Gutiérrez-Reina

Department of Electronics Engineering, University of Seville, 41009 Seville, Spain; syanes@us.es (S.Y.); storal@us.es (S.L.T.); mperales@us.es (M.P.-E.); dgutierrezreina@us.es (D.G.-R.)
* Correspondence: acasado4@us.es

Abstract: The water quality monitoring of large water masses using robotic vehicles is a complex task highly developed in recent years. The main approaches utilize adaptative informative path planning of fleets of autonomous surface vehicles and computer learning methods. However, water monitoring is characterized by a highly dynamic and unknown environment. Thus, the characterization of the water quality state of a water mass becomes a challenge. This paper proposes a variational autoencoder structure, trained in a model-free manner, that aims to provide a dynamic contamination model from partial observations of a homogeneous fleet of autonomous surface vehicles. To train the proposed approach, an oil spillage simulator based on heuristics is provided for world building. The proposed variational autoencoder was tested in three different environments characterized by different oil spill movements and twp different fleets equipped with different sensors. The results show accurate future contamination distribution predictions with a mean squared error ranging from 3 to 9% of the baseline at validation, defined as the static approach. Further tests addressed the overfit of the proposed neural network, showing a high robustness against unseen scenarios, and the effects of the gathered monitoring information in the variational autoencoder performance.

Keywords: VAE; prediction; neural networks; contamination model

check for updates

Academic Editor: Arturo de la Escalera Hueso

Received: 13 January 2025
Revised: 17 February 2025
Accepted: 5 March 2025
Published: 7 March 2025

Citation: Casado-Pérez, A.; Yanes, S.; Toral, S.L.; Perales-Esteve, M.; Gutiérrez-Reina, D. Variational Autoencoder for the Prediction of Oil Contamination Temporal Evolution in Water Environments. *Sensors* **2025**, *25*, 1654. https://doi.org/10.3390/s25061654

1. Introduction

Water plays an irreplaceable role in activities like the conservation of biodiversity, agriculture, tourism and industry, among others. However, besides being the most abundant liquid on the planet, there is a severe scarcity of quality usable water. This problem is becoming more accentuated due to climate change in recent years [1], requiring huge investments and difficult treatments for finding new water sources and acquiring quality water. The United Nations reflected the need for cooperation between worldwide organizations in Sustainable Development Goal (SDG) 6 [2]. This goal aims to ensure the availability and sustainable management of water and sanitation for all. SDG target 6.3 addresses another problem related to the release of hazardous chemicals and materials.

Approximately half of all the wastewater generated worldwide is released without treatment [3]. This, in addition to the accidental waste of residues like organic matter, oil spillage, heavy metals, and even radioactive substances, makes the situation a real environmental hazard. If these sources of contamination are not detected and treated accordingly in time, they can extend and cover the whole water surface, contaminating it directly, affecting its biodiversity, or indirectly promoting the appearance of invasive species or algae blooms [4], which, with time, can make the water unhealthy for human use. To avoid reaching this state of environmental crisis, water quality values must remain within water quality standards [5], and governments and organizations need to continuously monitor water masses. Monitoring is the preemptive measure against water contamination and degradation, as recovery is a process that takes several years [6].

Traditional water quality monitoring approaches focus on taking manual measures and analyzing samples in laboratories, requiring a lot of effort and human resources [7]. Recently, traditional methods are being replaced with superficial satellite methods [8] or intelligent robots, such as submarines and surface or aerial vehicles, that can be equipped with water quality sensors and robotic actuators [7]. Thus, vehicles are able to perform tasks ranging from exploration to actuation on water masses [9], involving detection, chasing and cleaning pollutants, and other monitoring tasks in real time. Furthermore, the time taken in laboratories to analyze samples induces a delay that, in the case of emergent contaminants, can cause a public health problem [10] not present when monitoring with autonomous vehicles. Therefore, it is envisioned that autonomous vehicles will play a crucial role in SDG target 6.6 in protecting water masses bodies [2].

The improvements in battery autonomy and computation power have made autonomous vehicles able to take intelligent actions. Thus, tasks that previously required an operator remotely controlling the vehicle are being replaced with a programmed movement policy that dictates the behavior of the vehicle [11]. The objective of these policies is to provide vehicles with target points or waypoints to travel to, making obtaining a policy a path planning problem. Another objective is to optimize the monitoring task assigned, which can be exploration or actuation, while taking into account factors present in the vehicle, such as battery, sensing, and actuating constraints. Thus, the path planning problem needs to take into account information about the environment. Developing a policy becomes a complex challenge due to the highly dynamic scenario of water masses. Since water is a fluid affected by several forces that facilitate the movement of particles through the whole mass, determining how a mass of water and its properties evolve through time is difficult. Therefore, vehicles need to adapt to this environment. Vehicles do not have prior knowledge about the environment. As a consequence, information is gathered by the vehicles during its mission and processed inside the vehicle or at a base station that the vehicle is able to communicate with, making offline planning an invalid solution.

With the advancements of neural networks, this field was able to provide solutions to the Adaptative Informative Path Planning (AIPP) problem based on deep architectures that have been developed [12]. As a clear example, Deep Reinforced Learning (DRL) approaches are able to solve the informative path planning problem, providing a valid policy with which vehicles are able to carry out the designated task [12], offering more robust and scalable solutions that adapt to the complexities and uncertainties of the environment. There are various optimization tasks, but regarding AIPP algorithms, some previous works have focused on water quality monitoring, contamination phenomenon exploration, and search tasks [13]. Among these previous works, some have focused on the contamination detection of algae blooms [14] and oil spills [15] using autonomous surface vehicles (ASVs), which are also called agents in the field of AIPP [16], equipped with specialized sensors [17,18]. However, most of the previous works made the assumption of lentic waters [19]. This means that the evolution of water properties and contaminants is slow and it can be considered that they do not change throughout a monitoring mission. However, this is not the case in larger water bodies like seam rivers and larger lakes, where the scenario is highly dynamic due to currents and wind, among other factors. Therefore, water quality conditions may evolve at the same time or faster than the actuation of the vehicles, and consequently, measures can become easily outdated. Although the use of multiple vehicles can alleviate the problem by increasing the data samples [20] or processing the age of the data collected [21], the obtained models will not reflect the real evolution of water quality parameters. Therefore, the planning actions will be sub-optimal in real scenarios.

The aim of a contamination model is to solve the estimation problem, obtaining the whole contamination map from partial observations and estimating its evolution. Several contamination models have been studied in the past [22]. In [23], the evolution of a contaminant in a river was modeled using mathematical hydrodynamic equations and solving the inverse model, reducing the potential harm caused by pollution accidents. In [24], several numerical models based on advection–dispersion equations or transport models for vulnerability assessment were used. However, characterizing the evolution in a larger water mass like a lake with partial observations cannot be explicitly described with equations, as it is affected by several chaotic effects. Bayesian contamination models like the Gaussian process [25] are able to provide valid solutions to the static problem, with the downside of a high computational cost that increases with the number of samples. In [26], a contamination model was obtained using a variational autoencoder (VAE) neural network, providing a more scalable solution with water samples. In the same paper, the results showed that a good contamination model is able to provide improvement in policy performance of approximately 50%. Thus, offering a forecasting module of the contamination that provides not only the present state of the contamination but makes a prediction of the future state of contaminants is likely to improve policy performance even further.

This paper proposes a variational autoencoder architecture based on the popular UNet network [27] combined with a prior and posterior convolutional neural network (CNN) architecture [28]. In [26], a similar architecture was proposed for the static case. In this paper, it was extended to a dynamic case, analyzing its capacity to estimate future distributions. This network was trained in a model-free manner, using only simulated interactions of the agents with the environment. The aim of the simulator is to provide a spatio-temporal distribution of pollutants in water bodies, replicating an oil spill accident. The simulator was used to create training and test datasets. The proposed VAE-UNet architecture will be a tool for any AIPP algorithm to plan ahead. The VAE works as a model that can provide accurate information about the contamination state from partial observations. Simultaneously, it captures the temporal-dependent behavior of the contamination, providing foresight for future contamination states.

To summarize, this paper contributes the following:

- A novel VAE neural network following the U-Net architecture that aims to provide future state estimations of water pollutant evolution.
- A comparison of the network performance against a naive baseline prediction.
- A further study of the limitations and overfitting of the suggested architecture.

This paper is organized as follows: Section 2 presents the materials and methods and describes the problem that the proposed VAE wants to solve, as well as how to set up the environment, contamination, agents, and the simulator. Lastly, the architecture of the VAE will be discussed. In Section 3, the results of the VAE obtained will be analyzed, and the model's behavior will be compared with that of a naive model. In Section 4, the main contributions of this paper and future lines of work will be discussed.

2. Materials and Methods

2.1. Problem Formulation

The aim of the proposed variational autoencoder (VAE) is to predict and detect the evolution of an accidental water contamination, providing a visual image of the current and future states of contamination, detailing its evolution. The environment analyzed evolves dynamically, increasing in size and expanding with time. An example of an oil spill contamination accident is showed in green in Figure 1 at different timestamps.

The information available is the data gathered by a fleet of autonomous surface vehicles equipped with sensors, which are called agents from now on. This information is assumed to be incomplete and dispersed like that seen in Figure 1c. Furthermore, this information can become easily outdated as the contamination value measured by an agent is instantaneous. In previous works like [29], contamination evolution was estimated through the Gaussian process. In [26], a neural network model was used to estimate the contamination distribution at the current timestamp. An example of this estimation can be seen in Figure 1d.

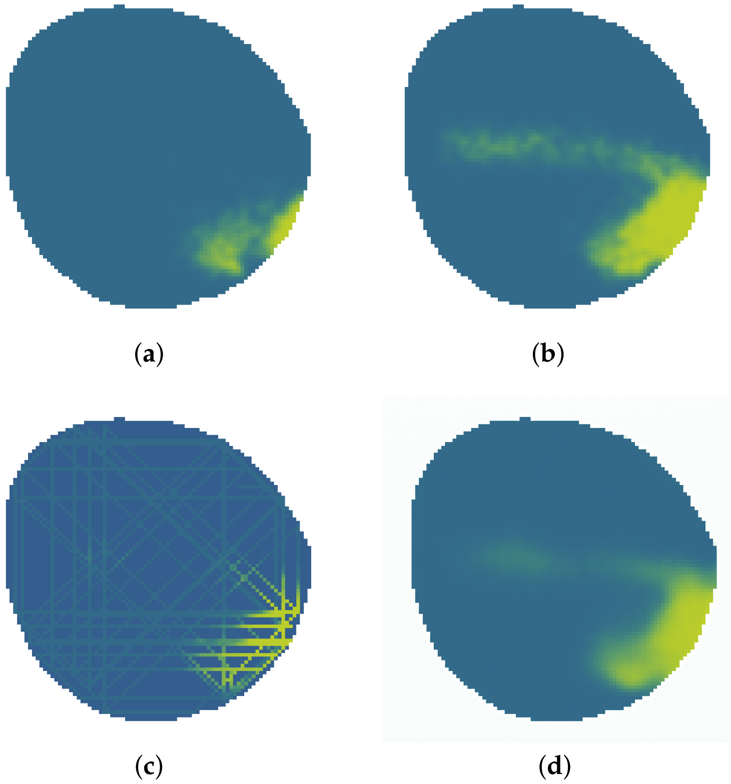

(**a**) (**b**) (**c**) (**d**)

Figure 1. Oil spill evolution and contamination prediction problem where oil is showed in green. (**a**) Oil spill at timestamp 50. (**b**) Oil spill at timestamp 200. (**c**) Data measured by agents. (**d**) Possible estimation at timestamp 200.

In this paper, this problem is further addressed. The data gathered by the agents need to estimated not only for the current timestamp contamination distribution but also the distributions at future timestamps. The complexity added by the estimation of future states of contamination requires a contamination model. Due to the uncertainties of the water environment and the unknown factors that affect oil spill contamination, establishing a mathematical model with the available data is not possible. To solve this problem, this paper proposes a VAE as a contamination model to provide future contamination states.

The reliability of this estimation is measured using the mean squared error (MSE) of the contamination distribution. At a given coordinate, the ground-truth contamination is compared with the estimated contamination distribution, and the MSE is calculated.

2.2. Environment Characterization

Given a water body affected by oil spillage, the 3D environment is reduced to a 2D horizontal distribution of surface pollutants as in previous works [12], assuming that pollutants in the vertical (z-direction) are thoroughly mixed, with a negligible concentration diffusion coefficient. Space will be divided using an arbitrary-sized square grid $H \times W$, as seen in Figure 2a. The environment is hence represented as a graph $G = (V, E)$ defined as follows:

- $V = \{v_{i,j} | 1 \leq i \leq H, 1 \leq j \leq W\}$, where each node $v_{i,j}$ represents a specific position in the grid.
- $E \subseteq V \times V$ is a subset of edges that connect adjacent nodes indicating possible movements between positions. Node adjacency is defined under the assumption that the grid is 8-connected.

From the environment representation, we can analyze navigable waters. For each node $v_{i,j}$, we will assign a binary value {0, 1}, defining matrix M of size $H \times W$ as the occupancy of the node $v_{i,j}$ in graph G as seen in Figure 2b:

$$M[i,j] \in \{0,1\} \quad \text{where} \quad M[i,j] = \begin{cases} 1 & \text{if } v_{i,j} \text{ is navigable} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

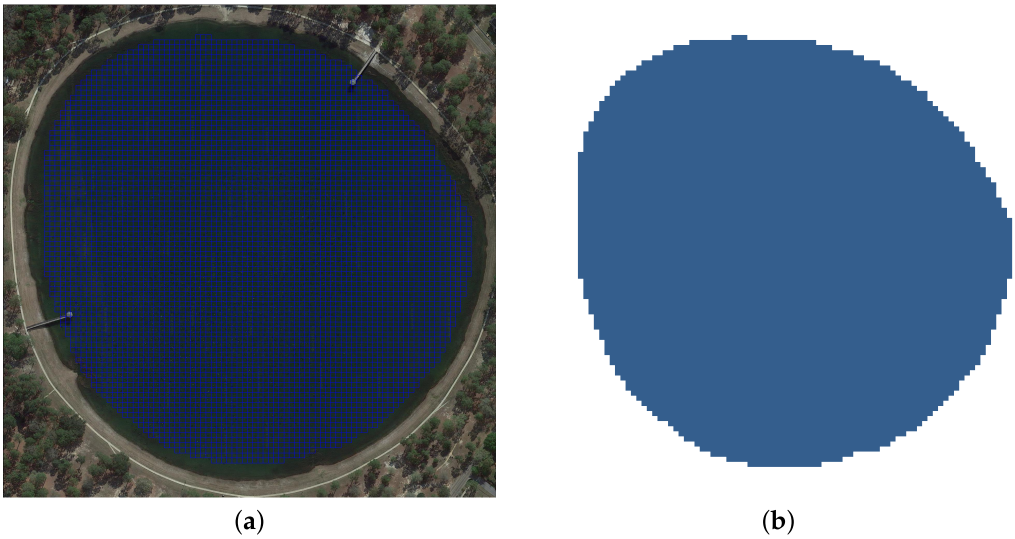


Figure 2. Environment characterization. (**a**) Environment grid V with M detailed. (**b**) Navigable water occupancy grid M.

2.3. Oil Spill Contamination Simulator

Contamination is defined as the concentration of a contaminant like crude oil measured by a sensor. The aim of the simulator is to provide contamination behavior that follows the movement of real contamination. To represent the contamination in a computationally efficient way, the oil contamination in the water body is modeled as a set of discrete particles represented by a set of real positions. With K as the total number of contamination positions, the set B is defined as

$$B = \{b_k = (x_k, y_k) \mid k = 1, 2, \ldots, K\} \tag{2}$$

where each element $b_k \in B$ represents a specific contamination position, and (x_k, y_k), the coordinates of b_k in a continuous reference system. A representation of B over M can be seen in Figure 3a. A matrix $\mathring{Y}$ of size $H \times W$ is defined as the contamination particle matrix:

$$\mathring{Y}[i,j] = |\{b_k \in B \mid (x_k, y_k) \in Area(i,j)\}| \tag{3}$$

where each element $\mathring{Y}[i,j]$ represents the amount of contamination positions contained in the $v_{i,j}$-associated area using the B set of contamination positions. $Area(i,j)$ represents the area of node $v_{i,j}$ in M contained within the real coordinates that surround the node, and $|\cdot|$ denotes the size of the subset, the quantity of contamination positions in that area. $\mathring{Y}$ is depicted in Figure 3b.

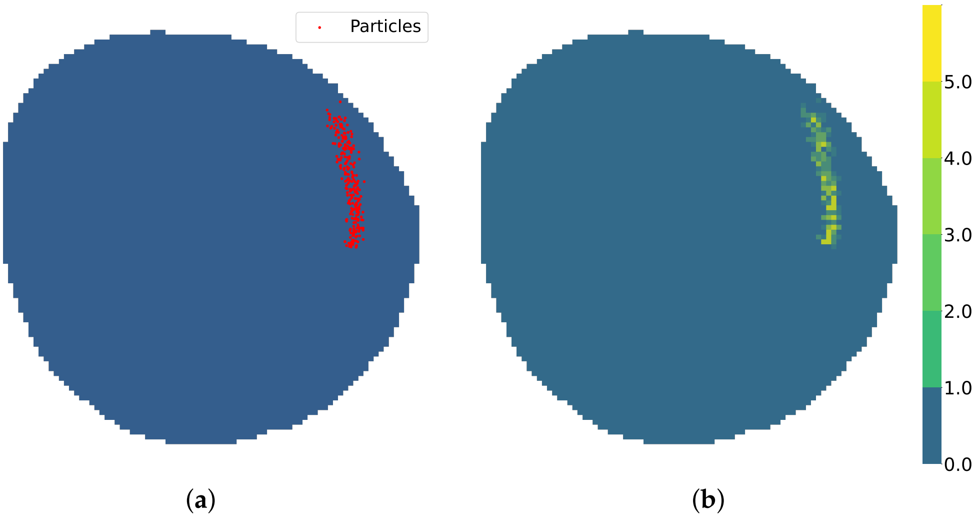


Figure 3. Contamination particle distribution. (**a**) Details of set of real contamination positions B over M. (**b**) Contamination particle matrix $\mathring{Y}$.

This simulator assumes that oil spill contamination must have a source from where all contamination particles originate from before moving through the water body, like in the case of water shipping [30]. With K as the total number of contamination sources. The set of contamination sources S is defined as

$$S = \{s_k = (v_{ij}) \mid k = 1, 2, \ldots, K\} \tag{4}$$

where each element s_k represents the node where a contamination source is located. At each given time step, each contamination source liberates Q contamination particles at position s_k contributing to set B.

In [23], a spatio-temporal model of the migration and dispersion of pollutants in a river was created using an empirical water quality hydrodynamic equation [23]:

$$\frac{\partial hc}{\partial t} + u\frac{\partial hc}{\partial x} + v\frac{\partial hc}{\partial y} = \frac{\partial}{\partial x}\left(E_x\frac{\partial hc}{\partial x}\right) + \frac{\partial}{\partial y}\left(E_y\frac{\partial hc}{\partial y}\right) + H\sum S_i \tag{5}$$

where hc is the current flow, E is the sum of molecular dispersion coefficients, H is the river bottom elevation, and S is the sink-source term of contaminants. This equation for a mass of water can also be described mathematically as in [31]:

$$b_k^{t+1} = b_k^t + \Delta t\left(\nu_{current} + g \cdot \nu_{wind} + \nu_{corretion} + \nu_{difusion} + \nu_{mechanical}\right) \tag{6}$$

This model has been further simplified to only consider the effects of the wind ν_{wind} and currents $\nu_{current}$ on the particle. The last three elements (horizontal turbulent diffusion velocity $\nu_{difusion}$, correction $\nu_{corretion}$, and mechanical spreading velocity $\nu_{mechanical}$) are represented by a random effect ν_{random} following the Brownian movement of particles.

$w_{wind}, w_{current}$, and w_{random} are the weights associated to each effect of the movement of particles. The particle movement model is then defined as

$$\hat{b}_k^{t+1} = b_k^t + \Delta t(w_{current} \cdot v_{current} + w_{wind} \cdot v_{wind} + w_{rand} \cdot v_{rand}) \tag{7}$$

The effect of the wind is modeled as a constant 2D vector that affects the whole scenario uniformly, as seen in Figure 4a. The current effect is modeled as a constant force field that assigns each node $v_{i,j}$ a 2D vector force. This force field of currents is depicted in Figure 4b and can be modeled as

$$w_{current}[i,j] = \begin{bmatrix} sin(i-a) * cos(j-b) \\ -cos(i-a) * sin(j-b) \end{bmatrix} for[a,b] \in V \tag{8}$$

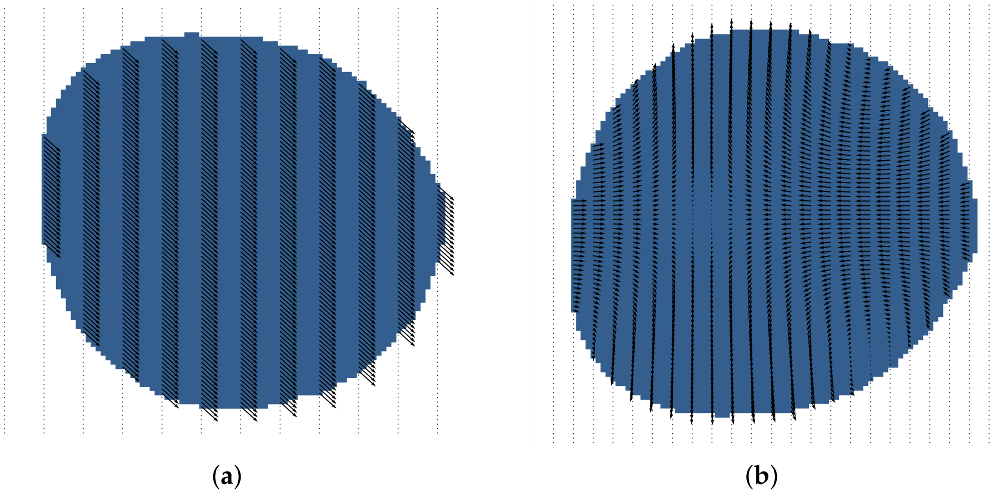


Figure 4. Particle movement effects. (**a**) Wind force field distribution. (**b**) Current force field distribution.

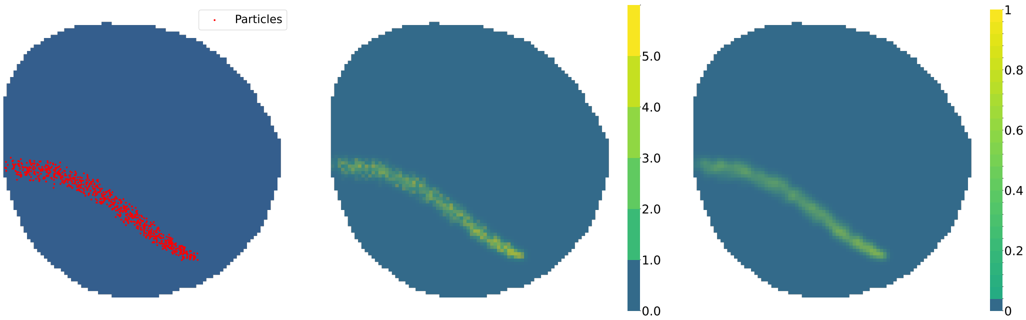
This simulator applies two rules before updating particle positions: No particle can be displaced beyond the water limits, staying at the last position if so. To model the concentration saturation of a node, which happens under nonlinear viscosity conditions, quantity C is defined as the maximum node capacity. If a contamination particle enters a node where there are already C particles present, the node is said to be saturated. The new particle position will update to the closest node following Algorithm 1, where O is defined as a set of nodes sorted in ascending order by Euclidean distance, with $\hat{b}_k^{t+1}$ as the expected new particle position for particle b_k.

$$b_k^{t+1} = \begin{cases} b_k^t & if\ \hat{b}_k^{t+1} \notin M \\ ParticleInteraction(\hat{b}_k^{t+1}) & if\ \mathring{Y}_{b_k^{t+1}} >= C \\ \hat{b}_k^{t+1} & otherwise \end{cases} \tag{9}$$

Working with discrete particles is efficient from the simulator's point of view. However, in a real-world scenario, an oil spill cannot be measured with particles. A good approximation to a valid real-world measure is contamination concentration. Given the amount of contamination particles present in a node $\mathring{Y}[i,j]$ and the maximum number of particles allowed to be present in a node C, node contamination density can be calculated. The shape of a real oil spill contamination area is characterized by a continuous contamination concentration [30]. The node contamination density provided by this simulator is a sparse matrix that presents sharp changes in the contamination density between adjacent nodes. To smooth the output of the simulator, a 5 × 5 kernel Gaussian mask is applied to the contamination density, providing the contamination concentration $Y[i, j]$ as the final output, as seen in Figure 5c.

Algorithm 1 *ParticleInteraction*

```
Input: $\hat{b}_k^{t+1}$
Output: $b_k^{t+1}$
Require: $V$
$\tilde{P} \leftarrow \hat{b}_k^{t+1}$
for $\mathring{v}_{ij} \in O(\hat{b}_k^{t+1})$ do
    if $\mathring{Y}(\mathring{v}_{ij}) >= C$ then
        $b_k^{t+1} = \mathring{v}_{ij}$
        break
    end if
end for
```



(**a**) (**b**) (**c**)

Figure 5. Simulator model. (**a**) Particle positions B. (**b**) Contamination particle matrix $\mathring{Y}$. (**c**) Oil contamination concentration Y.

2.4. Agent Description

The vehicles that monitor the environment are defined as a fleet of N agents. Each agent is represented by a variable p_n, where n is the vehicle index.

$$\mathcal{P} = \{p_n \mid n = 1, 2, \ldots, N\} \tag{10}$$

p_n is characterized by several factors:

- Position (p_n): Each position of the fleet corresponds to a node. A position can be described as $p_n = v_{i,j} \in V$
- Speed: An agent is able to move along the 8-connected node grid. However, some agents may be able to move several nodes along the same direction.

For an agent, moving along a given node of the map, whether horizontally, vertically, or diagonally, requires one temporal unit. At any given instant, agents must stay withing navigable water nodes M and no more than one agent can be simultaneously situated in a node to avoid crashes between them.

Agents are able to take measures of the oil contamination concentration, be it by electrochemical or spectral sensors like multi-spectral cameras [7]. Agents that use electrochemical sensors are able to take punctual measures. This measure will be representative of the oil contamination concentration in the node. This hypothesis will hold, given that the smoothness of the contamination phenomena maintains the locality of every measurement. Multi-spectral cameras are able to cover larger areas, allowing us to measure contamination concentration in nodes adjacent to the agent.

Agent monitoring sensing capability is hence characterized by an influence radius ρ_n. At any time step, the agent will provide measured data of the node occupied by it and all the nodes inside a circular area of radius ρ_n. As seen in Figure 6a, Θ is defined as the set of nodes inside the vehicle influence radius.

$$\Theta(p_n) = \{v \in V \mid |v, p_n| <= \rho_n\} \tag{11}$$

As agents explore the map, vehicles measure the contamination concentration in the environment. $\tilde{Y}$ is defined as the measured vehicle contamination concentration model:

$$\hat{Y}[i,j] \leftarrow Y[i,j] \iff v_{ij} \in \Theta(p_n) \tag{12}$$

This matrix is initialized to -1, as vehicles have no prior knowledge about contamination positions. If at a given instant, a node $v_{i,j}$ is contained by an agent p_n or its influence radius $\Theta(p_n)$, the value of $Y[\hat{i,j}]$ will be filled with instantaneous values of $Y[i,j]$, as seen in Figure 6. As contamination positions B evolve dynamically, the values of $\hat{Y}$ become outdated. To provide temporal information to the vehicle contamination model $\hat{Y}$, a time dependence matrix $U[i,j]$ is defined as

$$U^{t+1}[i,j] \leftarrow U^t \cdot \gamma + \Theta(\mathcal{P}) \tag{13}$$

This matrix is initialized to 0, denoting that the point has not been visited yet. As $U[i,j]$ is visited by an agent or an agent is inside its influence radius $\Theta(p_n)$, it is updated to 1, denoting that the cell has been visited recently. Then, each temporal unit $U[i,j]$ is multiplied by a forgetting factor γ in the range $(0,1)$. This translates to a value of $U[i,j]$ closer to 0 the older the measure present in the contamination model $\hat{Y}[i,j]$, as seen in Figure 6c.

The monitoring data of a fleet of four agents with influence radius $\rho = 1$ can be seen in Figure 6. The same fleet with influence radius $\rho = 4$ monitoring data can be seen in Figure 7. The fine estimation of the real contamination distribution will rely upon the VAE model, as explained in the following Section. Ultimately, all environment parameters are summed up in Table 1.

Table 1. Environment description.

Parameter	Description
$G(V,E)$	Environment graph.
V, v_{ij}	Set of environment nodes.
E	Set of paths between nodes.
$M[i,j]$	Navigation matrix. Indicates if a node i,j can be visited.
δ_t	Internal simulator time constant.
B, b_k	Set of contamination particles . Each particle has a position and is associated with a node v.
S, s_k	Set of source points. Each source point has a position and is associated with a node v.
Q	Pollutant liberated from the scenario by the contamination source each time step.
$\mathring{Y}[i,j]$	Contamination particle matrix.
C	Maximum number of particles per node.
$Y[i,j], \hat{Y}[i,j]$	Contamination concentration matrix and vehicle model of the contamination particle matrix.
w_{wind}, ν_{wind}	Wind gain and wind force vector.
$w_{currents}, \nu_{currents}$	Current gain and current force field.
w_{random}, ν_{random}	Brownian movement gain and Brownian movement force effect.
$\mathcal{P}, p_j$	Fleet of agents. Set of nodes where a vehicle is present.
ρ	Maximum distance of water measurement.
N	Number of agents present in the fleet.
$\Theta(p_j)$	Set of nodes inside a distance ρ.
O	Set of nodes sorted in ascending order by Euclidean distance.
$U[i,j]$	Matrix that indicates the age of the data associated with the value of node $\hat{Y}[i,j]$.

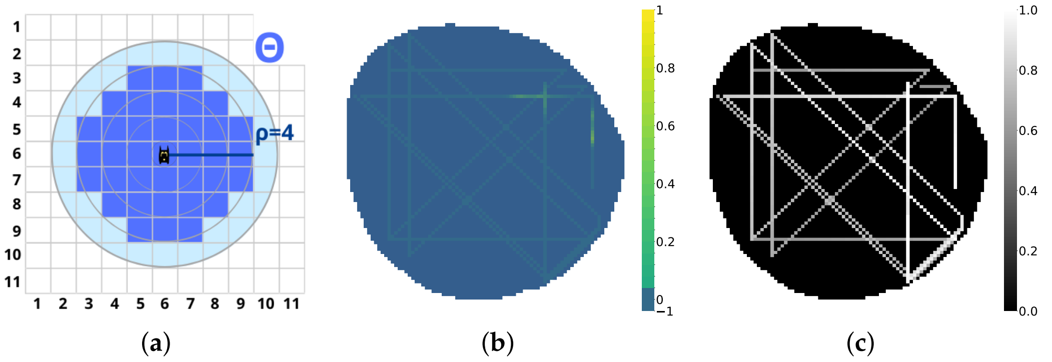


Figure 6. Agent Model. (**a**) Influence radius Θ. (**b**) Model contamination $\hat{Y}$ ($\rho = 1$). (**c**) Time dependence U ($\rho = 1$).

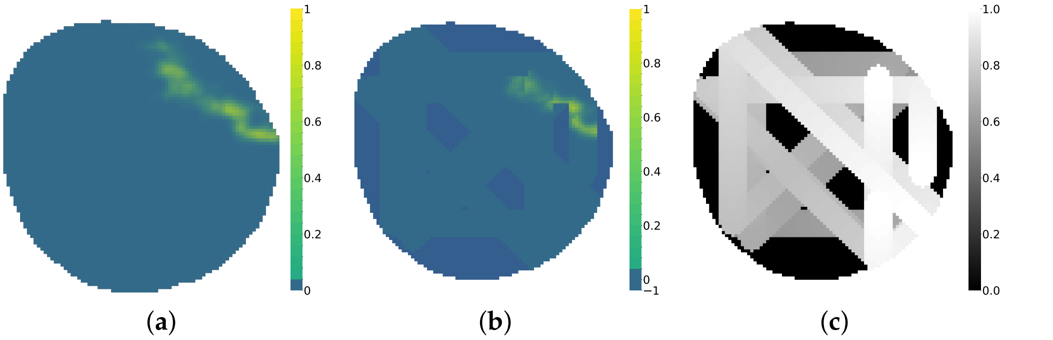


Figure 7. Agent exploration. (**a**) Contamination concentration Y. (**b**) Model contamination $\hat{Y}$ ($\rho = 4$). (**c**) Time dependence U ($\rho = 4$).

2.5. VAE-UNet Model

The proposed variational autoencoder architecture is an improvement of the one proposed in [26]. It is composed by a fully convolutional neural network (CNN) with an encoder and decoder phase comprising four convolutional and max-pooling layers inherited from UNet shape [27]. The variational side comprises 2 separate CNNs that produce the prior $N_{prior}(\mu, \delta)$ and posterior $N_{posterior}(\tilde{\mu}, \tilde{\delta})$ probabilistic Gaussian distributions. The network architecture is depicted in Figure 8.

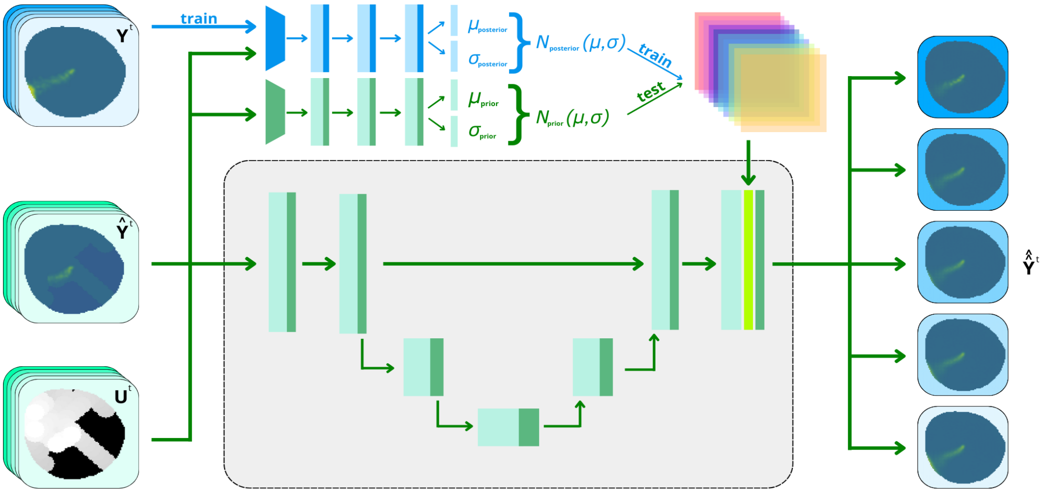


Figure 8. VAE-UNet architecture.

The main new developments from [26] add new channels providing past information of the contamination model in the input, aiming to obtain future states of the environment.

Hence, at time t, the input of the VAE will contain a temporal window of the measured contamination $\hat{Y}^t$ constructed from agent samples and the time dependence U^t associated with them. This set of 2 inputs is concatenated with K samples of data at present and past states $Input = \hat{Y}^{t-k}, U^{t-k}, \hat{Y}^{t-k+1}, U^{t-k+1}, \ldots, \hat{Y}^t, U^t \; k \in \mathbb{K}$. These windows can be formed by consecutive timestamps or be selected with asymmetric timestamps. The outputs of the network are the expected real contamination index at current time $\hat{\hat{Y}}^t$ and future time $\hat{\hat{Y}}^{t+1}, \ldots \tilde{I}^{t+K}$. This scheme can be seen in Figure 9.

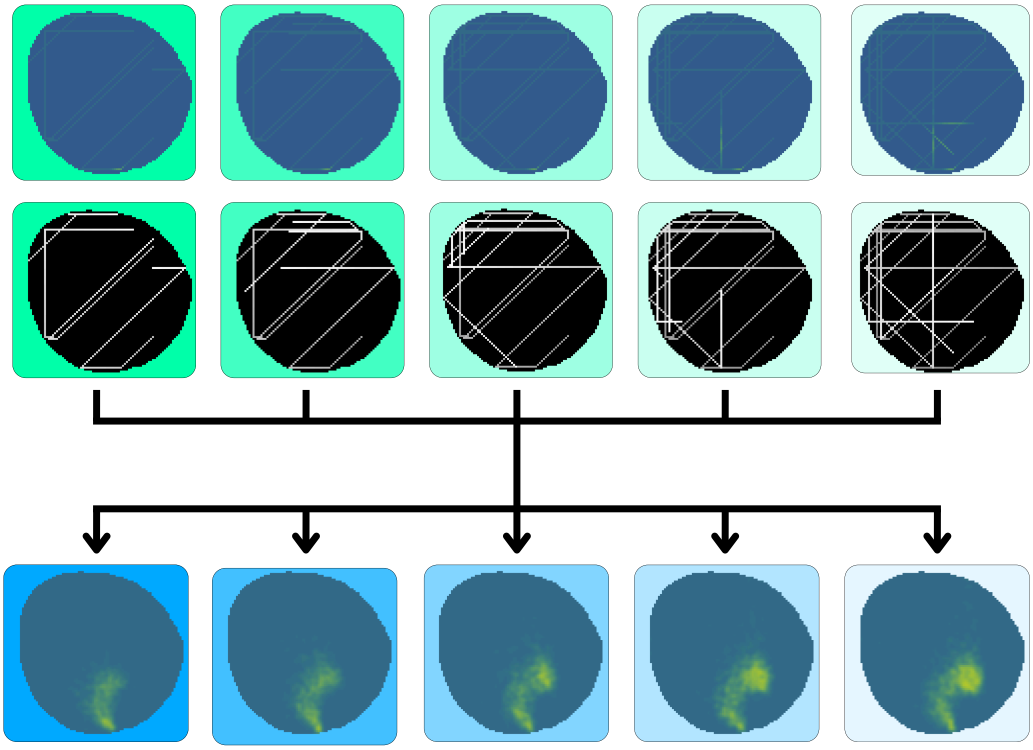


Figure 9. Expected inputs and outputs.

2.5.1. VAE-UNet Architecture

The proposed VAE needs to learn from a vehicle model contamination matrix and extract not only the data provided but also the implicit information about the dynamics that rule the environment and invert them to provide the desired output contamination estimation. Locating an oil spillage on a map is a task that involves visual information. The use of CNNs facilitates the extraction of high-level features from the input channels and, combined with a UNet shape [27], is able to perform the image segmentation of details very well. On the other hand, a structure broadly used in neural networks when the desired output replicates the input with variations is the autoencoder [32]. This structure is able to encode the main aspects that dictate the behavior of the oil spill to a reduced dimension layer, called the latent space, and use the same rules to reconstruct the same inputs. The module in charge of reducing dimensions is called the encoder, and the one in charge of reconstructing the input is called the decoder. An enhanced version of the autoencoder, called a variational autoencoder [33], parameterizes the latent space into n Gaussian distributions, with its mean μ and covariance σ^2 being the output layers of its encoder. This approach is able to provide a structured architecture to fit the data into Gaussian distributions, with the subsequent benefits of the sampling capacity allowing us to provide as many different outputs as samples taken from these Gaussian distributions. In [28], this VAE architecture was improved through the addition of another latent state called the posterior, introducing the prior–posterior architecture.

The VAE used in this study presents 2 different latent spaces produced in the encoder–decoder phase. The one generated by the prior has as inputs the measured contamination $\hat{Y}$ and time dependence U, providing a latent space N_{prior} originated from the minimization of the prediction error from the partial observations. The second latent space $N_{posterior}$ has as inputs not only the partial observations but also the actual real contamination concentration Y. The information provided by training with the desired output makes the network learn from data not available originally, providing a lower loss and hence, a better latent space. However, complete contamination information is not available in a real-world experiment, with the prior network being the only feasible solution. The aim of this architecture is to track both latent spaces during training and try to reduce the Kullback–Leibler divergence [28] present between the latent spaces of the prior and posterior network. In this way, the prior is able to induce the data only provided to the posterior. This makes the generated estimations provided by the VAE follow behaviors closer to the ground-truth when partial observations are provided. A scheme of the architecture is shown in Figure 8.

2.5.2. VAE-UNet Loss Function

The loss function here is an adaptation of the loss proposed in [34]. The output of the VAE ($\hat{\hat{Y}}$) must match the real ground truth (Y). However, during training, neural networks are biased to first minimize the largest loss [35]. Due to the nature of a temporal prediction, the farther it is into the future, the higher the probability to miss in the prediction. This motivates the network to minimize loss in most future predictions (higher by origin) and less in the near future. To avoid this and improve the accuracy of close future predictions, the MSE of the future predictions is weighted with a forgetting factor $\phi \in (0,1]$. The reconstruction loss at instant $t+k$ is thus defined as

$$\mathbb{L}_{recon} = \sum \mathrm{MSE}(Y^{t+k}, \hat{\hat{Y}}^{t+k}) \cdot \phi^{k} \quad k \in \mathbb{N} \tag{14}$$

As mentioned before, the latent space is doubled by 2 variational latent spaces. The output of the network is composed using the posterior latent space during training (blue lines in Figure 8) and prior latent space during testing (green lines in Figure 8). The divergences between both latent spaces leads to the definition of the Kullback–Leibler (KL) divergence loss.

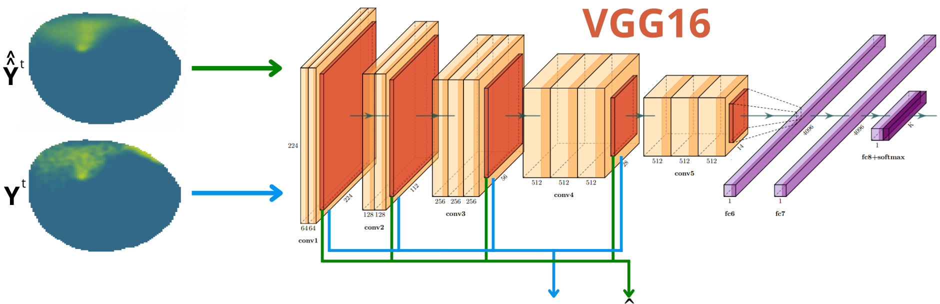
$$\mathbb{L}_{KL} = KL(N_{prior}(\mu,\delta), N_{posterior}(\mu,\delta)) \tag{15}$$

Lastly, As the VAE needs to process an image as input and provide an image as output, a style transfer strategy [36] is adopted utilizing the $VGG16$ [37]. The $VGG16$ is a fully trained CNN for image recognition on a huge dataset. Through studying this network, it is observed that the initial layers contain the low-level features of the image (color, edges, texture, etc.), and deeper layers contain higher-level features (objects and their arrangement in the input image) [36]. In [36], it was demonstrated that pixel-by-pixel comparisons, like the one we performed with $\mathbb{L}_{recon}$, show a low performance when training a new neural network using a small dataset. However, if training is assisted using a fully trained network like the $VGG16$, it results in a much better-performing network. This is defined as style transfer [36] and is a strategy used in image transformation problems. The $VGG16$ is used to compare the output of the proposed VAE-UNet against the actual ground truth using the features extracted by the $VGG16$. By doing this, we encourage the output oil spill to cover the same nodes that are polluted in the ground truth, preserving the same spatial structure. The mean square error (MSE) of the $VGG16$ layers $relu1\_2$, $relu2\_2$, $relu3\_3$, and $relu4\_3$ is defined as Perceptual Loss, as described in Figure 10.

$$\mathbb{L}_{perceptual} = \mathrm{MSE}(\xi(Y), \xi(\hat{\hat{Y}})) \tag{16}$$

The total loss of the network is defined as the weighted sum of all 3 losses.

$$\mathbb{L} = \omega_{recon}\mathbb{L}_{recon} + \omega_{KL}\mathbb{L}_{KL} + \omega_{perceptual}\mathbb{L}_{perceptual} \tag{17}$$



$$\mathbb{L}_{perceptual} = MSE(\xi(Y), \xi(\hat{Y}))$$

Figure 10. Depiction of the feature comparison performed in $\mathbb{L}_{perceptual}$.

2.6. Agent Planner

The fleet model plays a crucial role in the modeling of the contamination. The input of the VAE is directly related to the fleet's information-gathering performance. It is necessary to address the model at the same time that the path planner is designed. The unavoidable consequence of an ill-designed model is poor monitoring and decision making. From the model perspective, poor information acquisition will result in a catastrophic scarcity of examples to learn from.

Agent information policies have been broadly studied [13], and the optimal policy is out of the scope of this paper. The aim of this study was to use the VAE to predict future states of contamination from partial observations that are assumed to provide relevant enough data of the contamination source. The proposed policy was selected to be a safe random coverage policy that provides enough information without considering the optimal policy in the long range. This puts the focus on the necessary objective of obtaining a model that works even when the policy is not perfect. Consequently, the model will be robust enough to serve other purposes in other scenarios. At the start, agents choose a random direction in the 8-connected grid to move toward. The selection takes into account that the agent will stay within navigable water M limits and the destination cell is not already occupied by another agent. When an agent movement will cause the agent to leave M or crash with another agent, the agent chooses a new safe random direction in the 8-connected grid. To avoid the overlapping of positions and possibilities of agents occupying the same node due to simultaneous movement, agents follow a priority order defined randomly at the start of the simulation, where an agent executes the policy just after the previous one has established a target position.

3. Results

Experiments were performed using a 97 × 93 node scenario in a circular shape. The simulator was used offline to create datasets for $\hat{Y}^t$ and U^t as inputs of the network and Y^t as the ground truth of the scenario and the value we want to compare to as output from the variational autoencoder (VAE). To evaluate the performance of the VAE in different environments, the simulator was configured to create scenarios where oil spill evolution can present three different behaviors: Linear dispersion, currents and wind affect particles moving in a general direction, allows erratic behaviors, as seen in Figure 11c. Circular expansion, wind and currents forces are minimal and the sources have a high flow of particles, allowing for particles to grow in a circular shape as seen in Figure 11a. Triangular diffusion is a rule-based behavior that mimics an oil spill caused by a flow coming from a broken cross pipe that follows a cross shape, as seen in Figure 11b. Oil presents a fast release that slows down once it has been liberated to the water body. This last environment presents the most artificial behavior but adds more complexity to the problem. The simulator (https://github.com/AloePacci/cpp_oil_simulator, accessed on 11 January 2025) and VAE (https://github.com/AloePacci/VAEPOCTEWE, accessed on 11 January 2025) codes are available on GitHub.

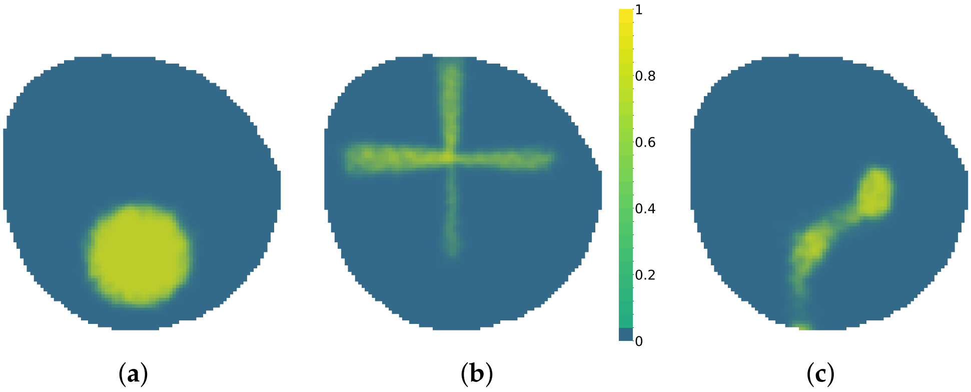


Figure 11. Oil spill behaviors. (**a**) Circular expansion. (**b**) Triangular diffusion. (**c**) Linear dispersion.

The VAE was configured to have five window frames $(\hat{Y}^t, U^t)$ as inputs expatiated uniformly five time steps between each other ranging from t_{-20} to t_0, and another five frames as output ranging from t_0 to t_{20}. Several datasets were created for each of the oil spill behaviors; 20,000 different contamination scenarios were synthesized for training, 4000 for testing and 200 for validation. These include monitoring situations with agents equipped with electrochemical sensors, with influence radius $\rho = 1$, and agents equipped with cameras, with influence radius $\rho = 4$. An example of a contamination instance dataset can be seen in Figure 12.

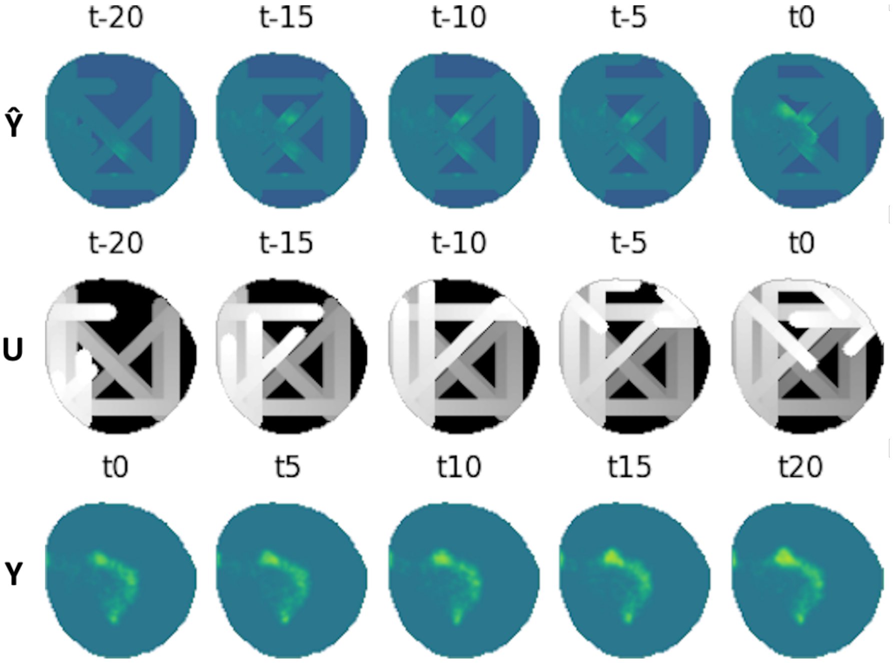


Figure 12. Dataset example containing inputs $(\hat{Y}^t, U^t)$ for $\rho = 4$ and ground truth (Y^t).

All simulations and training were carried out on a server running Ubuntu 22.04.4 LTS (Universidad de Sevilla, Sevilla, Spain), equipped with an Intel Dual Xeon Gold 5220R CPU 2.20 GHz, 192 GB of RAM and two GPUs: Nvidia Quadro A4000 48 GB and Nvidia RTX 3090 25 GB. Training loss was calculated using Equation (17) and forgetting factor $\phi = 0.9$. The hyperparameters $\omega_{recon}, \omega_{KL}, \omega_{perceptual}$ and the learning rate lr were optimized utilizing Optuna [38] to minimize the reconstruction loss $\mathbb{L}_{recon}$ in order to address the final objective of model accuracy.

Given the datasets and different combinations of agents and oil spill evolution behaviors, different networks were trained to calculate the cross losses and validate the effectiveness and generalizing capabilities of the proposed VAE. Thus, results were divided into a fleet of agents characterized by an influence radius $\rho = 1$ and another characterized by an influence radius $\rho = 4$. For each fleet of agents, four different models were trained: one for each of the oil spill behaviors for cross validation, and one containing a fusion of all three oil spill behaviors, from now on called the generalized network. The weights were chosen for the network at the epoch that showed the lowest test loss.

3.1. Performance Metrics

As mentioned previously, the aim of the VAE-UNet is to provide future states of oil spill contamination. The baseline taken for comparison to evaluate the performance of the network is the static approach, where the environment is considered non-dynamic and the expected future state of the contamination position is equal to the current one $\hat{\hat{Y}}^{t+k} = \hat{Y}^0$ $k \in \mathbb{N}$. The loss at time step 0 at areas recently visited by agents is minimal. However, in areas with data measured several steps ago, or with predictions at future timestamps, the error using this approach increases at a high rate. The visual of this loss establishes this baseline as a solution that underperforms but provides a valid estimation.

Figure 13 shows the results of evaluating the reconstruction loss MSE $\left(Y^t - Y^0\right)$ before using the baseline approach. The loss incurred by the VAE-UNet was calculated with respect to the baseline and expressed in a percentage value of the baseline loss.

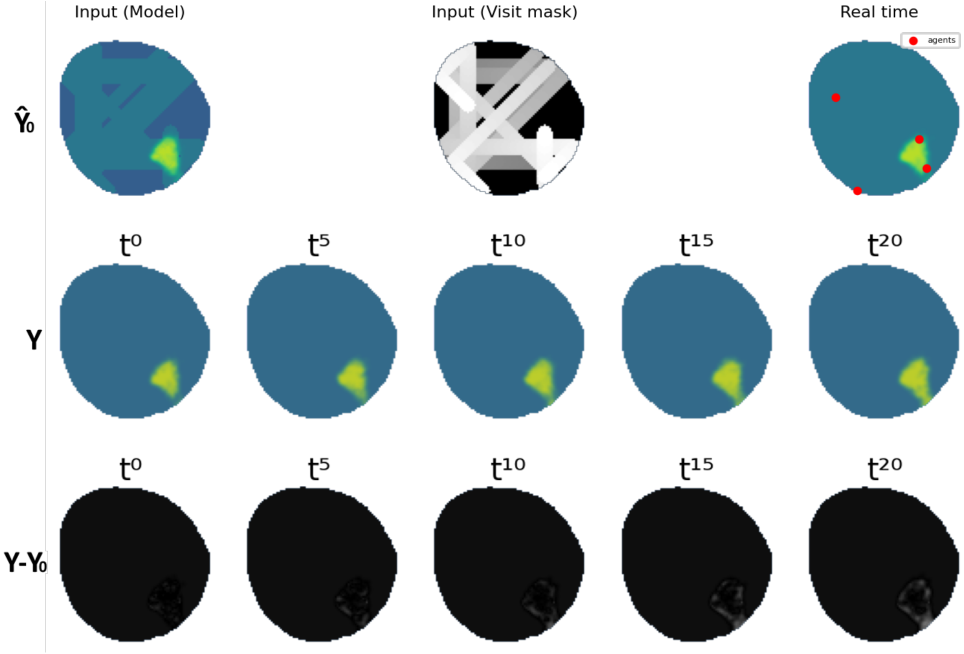


Figure 13. Baseline error $\left(Y^t - Y^0\right)$.

3.2. Fleet with $\rho = 1$

This fleet is characterized by an influence radius $\rho = 1$ and is able to take measurements in the nodes where the agents are currently located at, like the one used in [39] equipped with electrochemical sensors. It is made up of four different agents that are able to move through three cells each time step. An example of the dataset can be seen in Figure 14.

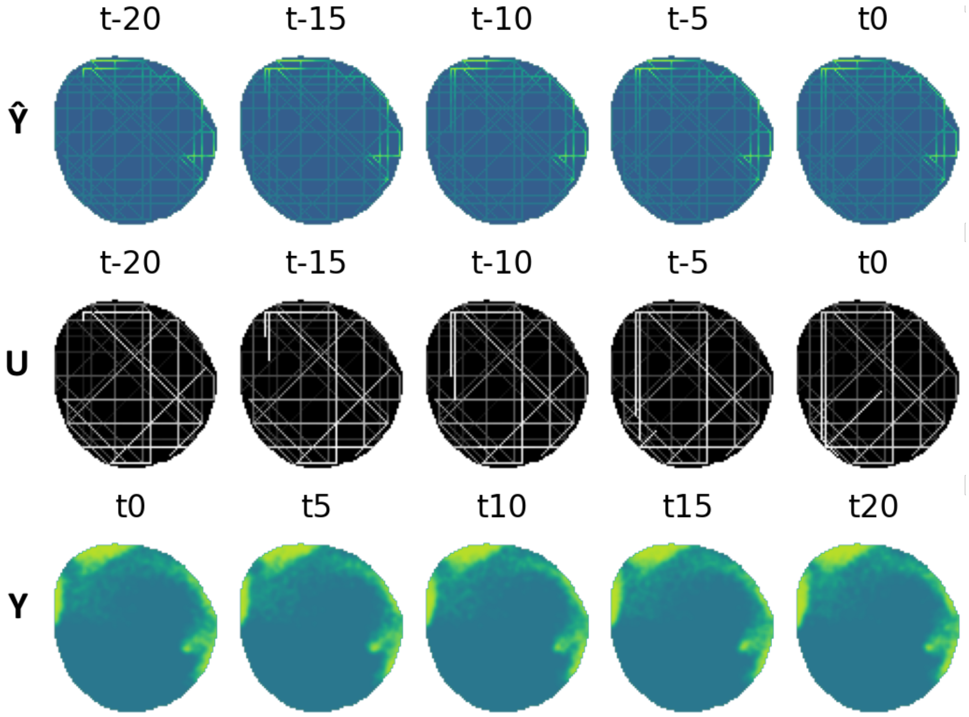


Figure 14. Dataset example containing inputs $(\hat{Y}^t, U^t)$ for $\rho = 1$ and ground truth (Y^t).

The networks training loss curve shows a high descending slope that stabilizes around epoch 100, as seen in Figure 15. A value of 200 epochs for training was considered sufficient. Figure 16 shows the relation between the three different losses during training. In analyzing the reconstruction loss $\mathbb{L}_{recon}$ curve, the loss associated with each of the future estimations shows a similar value despite the loss reduction applied to future predictions. This justifies the assumption taken before that the further the network looks into the future, the higher the loss.

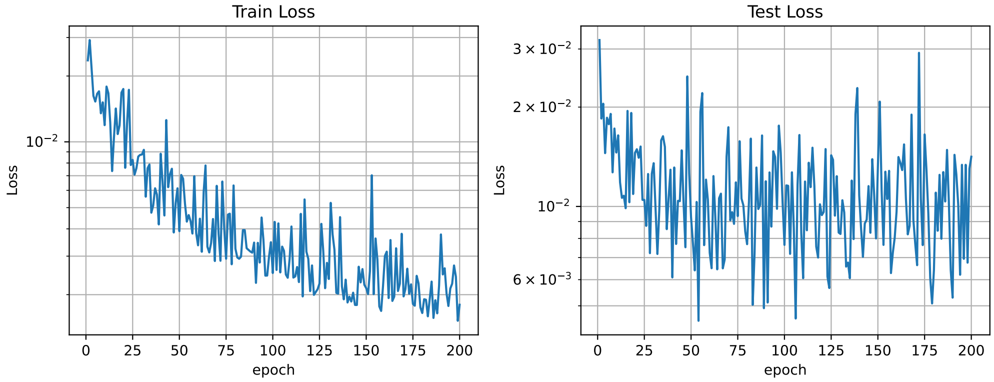


Figure 15. Training and test loss curves ($\rho = 1$).

In Table 2, the reconstruction loss value of each trained network is shown. The network trained using only triangular diffusion data shows the lowest loss. This could be due to the simplicity of the contamination behavior for this case. It is followed by circular expansion; being simpler, it presents no effects of the wind or currents. Lastly, the linear dispersion case shows the highest loss. This could be due to the high variety and random evolution of this contamination behaviour. The generalized network, presenting data from all the different datasets, presents a middle value.

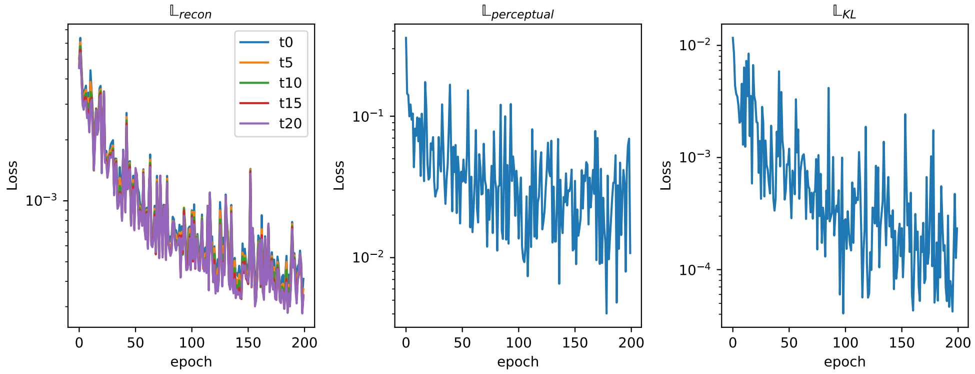


Figure 16. Training loss curves ($\rho = 1$).

Table 2. Train and test reconstruction loss (fleet $\rho = 1$).

VAE-UNet	Test Loss	Train Loss
Circular expansion	5.216×10^{-3}	1.449×10^{-3}
Triangular diffusions	3.479×10^{-3}	1.624×10^{-3}
Linear dispersion	12.394×10^{-3}	10.139×10^{-3}
Generalized	6.637×10^{-3}	2.124×10^{-3}

Once the different networks were trained, their performances were evaluated. Table 3 shows the reconstruction loss results of evaluating each trained network against each of the different validation datasets. In view of the results, the solution presented in this paper is able to provide a prediction of the evolution of an oil spill with an error of less than 10% of the naive baseline approach for each assigned contamination behaviour. This shows that the network is able to predict oil spill evolution with high accuracy in environments similar to those it was trained with.

To test the adaptability of the trained VAE-UNet in unseen behaviour, the network was evaluated against datasets in which the oil spill behaves very differently compared to the dataset that it was trained on. The results show that the VAE behaves better than the baseline prediction in all individual cases, except for the circular expansion cross-loss against the triangular diffusion case, which underperformed. This occurs mostly due to overfitting, as there is no wind or current effect in the circular expansion dataset. The opposite can be seen in the linear dispersion network, where contamination particles evolve in diverse ways, allowing for a better adaptability and lower cross-losses. A special case is triangular diffusion error, where the generalized VAE performs better than in triangular case in its own error. This could be due to loss hyperparameters being optimized for the generalized case and a better understanding of particle behavior due to a more varied dataset.

Table 3. Cross-losses VAE-UNet architecture (fleet $\rho = 1$).

VAE-UNet	Circular Expansion Environment Loss	Triangular Diffusion Environment Loss	Linear Dispersion Environment Loss
Circular Expansion	1.54%	100.16%	74.22%
Triangular Diffusions	20.34%	4.71%	20.76%
Linear Dispersion	17.48%	11.73%	7.71%
Generalized	1.82%	4.53%	8.72%
Baseline	0.8551	0.3194	0.2815

Percentages were calculated with respect to the baseline.

Lastly, Table 4 shows the evolution of $\mathbb{L}_{recon}$ along the different prediction timestamps for each VAE-UNet trained through the complete validation dataset. The values show that as the VAE-UNet predicts further into the future, the higher the error in the prediction. The increase in this loss is higher in the specific contamination behaviors of VAE-UNets due to them not being trained for the whole dataset. However, the loss value remains low, presenting a better estimation than the baseline. The Supplementary Material includes a video showing the evolution of the VAE-UNet as the fleet explores the map.

Table 4. VAE-UNet performance comparison by time step (fleet $\rho = 1$).

VAE-UNet	Loss t_0	Loss t_5	Loss t_{10}	Loss t_{15}	Loss t_{20}
Circular Expansion	36.04%	37.66%	39.08%	40.59%	42.27%
Triangular Diffusions	14.65%	15.7%	16.73%	17.66%	18.45%
Linear Dispersion	12.41%	13.25%	14.11%	14.96%	15.82%
VAE-UNet Generalized	3.51%	3.68%	3.84%	4%	4.16%
Baseline	1.5173	1.5687	1.6205	1.6726	1.7252

Percentages were calculated with respect to the baseline.

3.3. Fleet with $\rho = 4$

The fleet is characterized by an influence radius $\rho = 4$; this could be the case of agents equipped with spectral sensors like the ones present in [40]. It is made up of four different agents that are able to move through three cells each time step. An example of this fleet's dataset can be seen in Figure 12. This fleet is able to provide contamination information about nodes adjacent to the agents in a four-node radius, providing more information than the one equipped with electrochemical sensors, resulting in a lower loss, as seen in Figure 17. The analysis of the reconstruction loss for each time step in Figure 18 shows similar values despite the loss reduction. This result enhances the assumption that the further into the future the estimation, the higher the error committed by the VAE in the prediction. Table 5 presents the training and testing reconstruction losses. The losses present the same relationships. However, due to having more information, the magnitude of the loss is lower.

Table 5. Train and rest reconstruction loss (fleet $\rho = 4$).

VAE-UNet	Test Loss	Train Loss
Circular Expansion	0.930×10^{-3}	0.756×10^{-3}
Triangular Diffusions	1.719×10^{-3}	1.427×10^{-3}
Linear Dispersion	7.580×10^{-3}	1.057×10^{-3}
Generalized	3.707×10^{-3}	1.249×10^{-3}

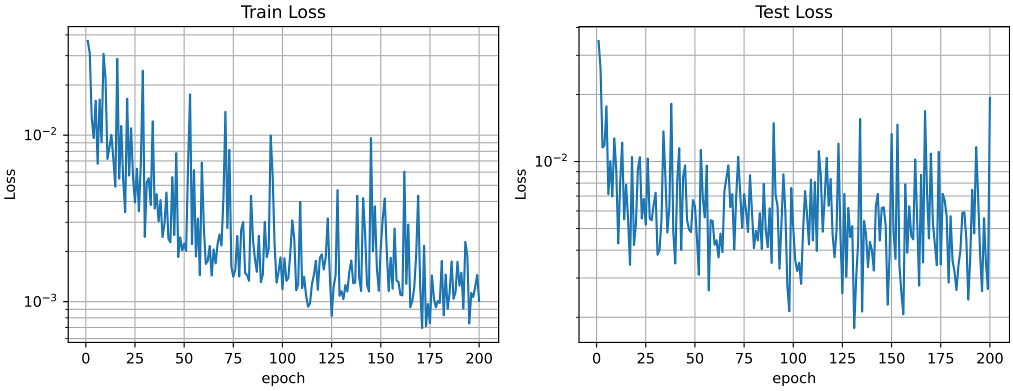


Figure 17. Training and test loss curves ($\rho = 4$).

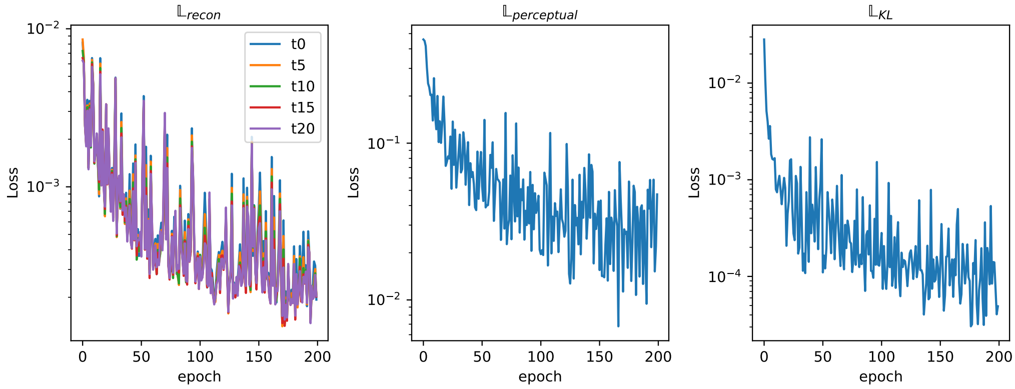


Figure 18. Training loss curves ($\rho = 4$).

The performance of the network was evaluated, and the results are shown in Table 6. The static approach selected as the baseline provided by this fleet has more information, providing an estimation loss that is six times lower on average. The VAE is able to process the new information to provide better estimations. However, even though the absolute loss values were reduced. The cross-losses show a performance worse than those of the baseline in environments different from those in the trained cases. This overfit is more present in the circular expansion case trained without the influences of the wind or currents. The opposite is seen in the linear dispersion case, presenting a more varied training dataset and environment effects.

Table 6. Cross-losses of VAE-UNet architecture (fleet $\rho = 4$).

VAE-UNet	Circular Expansion Environment Loss	Triangular Diffusion Environment Loss	Linear Dispersion Environment Loss
Circular Expansion	4.31%	386.17%	601.95%
Triangular Diffusions	43.26%	18.09%	146.96%
Linear Dispersion	46.38%	53.41%	47.86%
Generalized	4.11%	16.77%	60.90%
Baseline	0.1615	0.0479	0.0233

Percentages were calculated with respect to the baseline.

Table 7 shows the evolution of $\mathbb{L}_{recon}$ along the different prediction timestamps through the complete validation dataset. The overfit can be easily seen in the circular expansion network. The rest of the trained networks present loss values lower than those of the baseline. The generalized dataset presents the lowest reconstruction loss value, being the network trained with the most varied dataset. These results proclaim that the more varied the dataset, the better and the more robust the network, leading to better adaptation to unknown environments and lower losses.

Table 7. VAE-UNet Performance comparison by time step (fleet $\rho = 4$).

VAE-UNet	Loss t_0	Loss t_5	Loss t_{10}	Loss t_{15}	Loss t_{20}
Circular Expansion	78.00%	85.32%	91.31 %	96.34%	100.96%
Triangular Diffusions	27.17%	28.36%	29.64%	31.18%	32.56%
Linear Dispersion	25.34%	27.58%	29.75%	31.81%	33.76%
VAE-UNet Generalized	7.19%	7.49%	7.75%	7.99%	8.21%
Baseline	0.2470	0.2701	0.2948	0.3206	0.3474

Percentages were calculated with respect to the baseline.

A visual representation of the VAE-UNet generalized network for the fleet equipped with electrochemical sensors can be seen in Figure 19, showing the partial input $\hat{Y}$ and the output of the VAE $\hat{\hat{Y}}$ against the ground-truth data Y and the difference between both $\hat{\hat{Y}} - Y$. As mentioned previously, the agent policies are not the objective of this study. Figure 20 shows an example where agents have yet to discover the oil spill contamination. The VAE-UNet predicts contamination to be in an erroneous position. This addresses the effect of the fleet's information-gathering performance present both during testing and training in the modeling of the contamination.

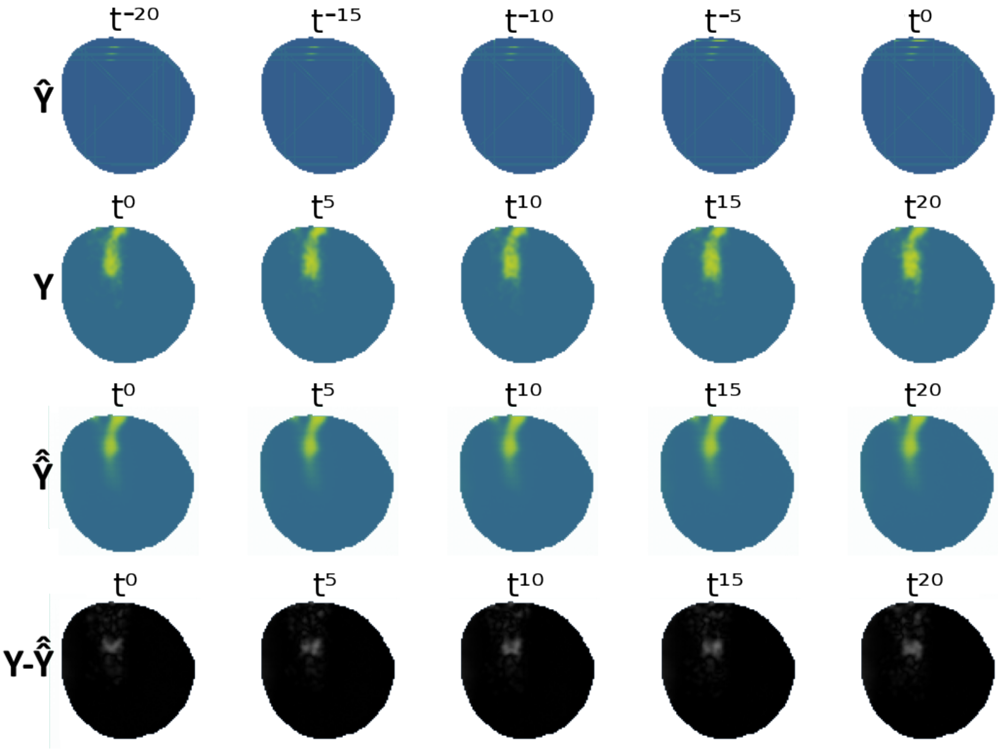


Figure 19. Comparison of VAE (generalized) output against real ground truth.

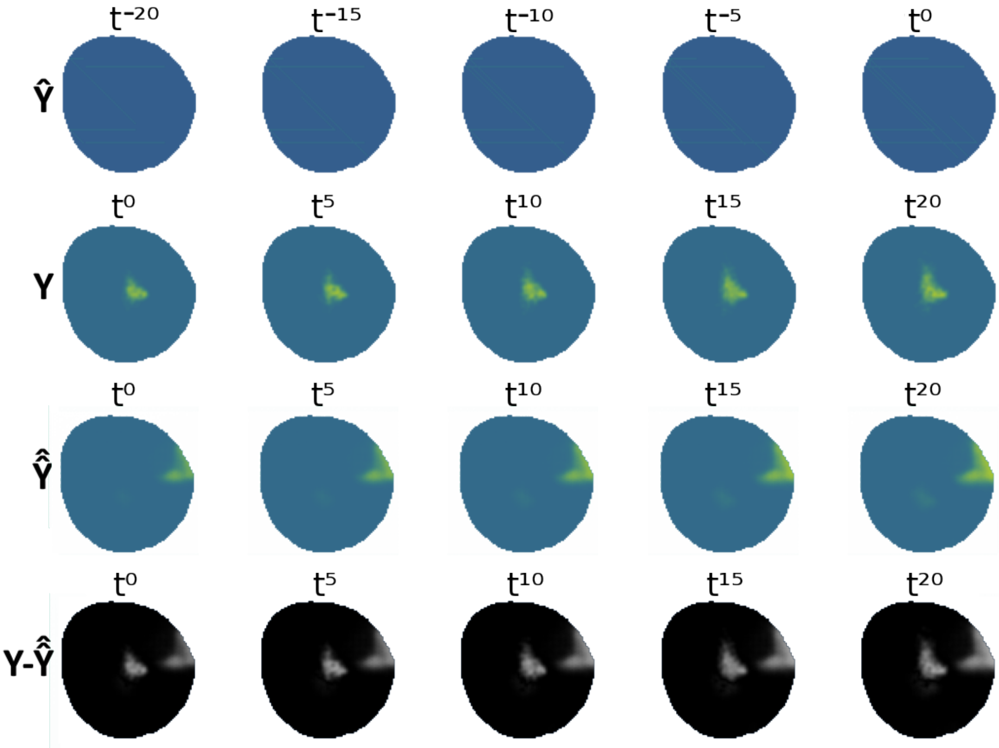


Figure 20. Comparison of VAE output against real ground truth, unfavorable case.

Figure 21 shows the evolution of the different losses along three different oil spill environments. Initially, reconstruction loss increases until the contamination area is detected and then decreases sharply. The generalized network presents instances where the fleet with $\rho = 1$ presents a lower loss than $\rho = 4$. This is due to the planner policy of the agents that provides different monitoring information to each fleet.

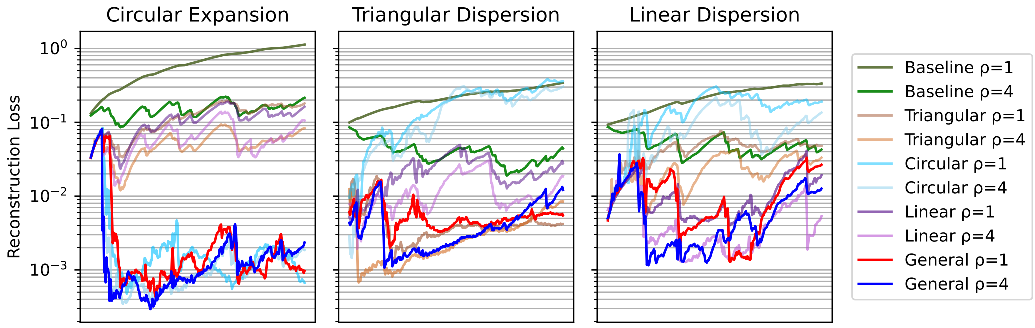


Figure 21. Reconstruction loss during a synthesized oil spill contamination accident.

4. Conclusions

This paper proposes a variational autoencoder to predict the evolution of oil spill contamination in water bodies from partial observations. To assess performance, it was tested on several scenarios presenting three different simulated oil contamination environments: circular expansion, presenting minimal wind and current forces; triangular diffusion, where contamination is exposed to biased currents; and linear dispersion, allowing random behaviors with high wind and current effects. Furthermore, the test was duplicated using two fleets of autonomous surface vehicles with different monitoring capabilities: a fleet equipped with electrochemical sensors able to take punctual measurements and a fleet equipped with spectral cameras able to monitor an area close to the vehicle.

According to Tables 4 and 7, the validation results of the proposed generalized VAE show a prediction loss as low as 3.51%, the baseline at current time, by the fleet equipped with electrochemical sensors, and as high as 8.21%, the baseline 20 time steps into the future, by the fleet equipped with spectral cameras. The magnitude of this loss increases with the age of the predictions, presenting an increase the further into the future that the predictions are made. The overfit of the network to the data trained on was tested using networks trained with datasets presenting only one of the three available environments. The results show a lower loss at the specific environments and a higher loss at different ones. A further study showed that this overfit decreases when the network is trained with a more varied dataset, presenting validation losses as high as six times the baseline for the circular expansion case in the fleet equipped with spectral cameras, or 11.73% for the baseline, in the linear dispersion case with the fleet equipped with electrochemical sensors. Thus, it is expected that the proposed generalized network trained with a varied dataset performs very well in new environments.

The gathering performance of the agents affects the proposed VAE in two different ways. The fleet equipped with spectral cameras is able to cover a wider monitoring area, providing more monitoring data and allowing for a reconstruction loss six times lower on average. Furthermore, the wider coverage allows for detecting the contamination position with more certainty. The path-planning policy is random, presenting cases where the vehicles have not detected any contamination and the prediction erroneously locates the contamination. Thus, in a monitoring scenario, the initial losses of the proposed VAE show an underperforming solution.

Future lines of work diverge into two lines of investigation. On the one hand, an analysis of the effect of the proposed VAE-UNet structure on informative path planning should be performed, providing numeric data of the effects of taking a future state of contamination particles into account in agent policy estimation. On the other hand, the limits of the VAE should be addressed, evaluating the effects on the prediction accuracy of different agent policies and the input requirements regarding number and age of window frames.

Supplementary Materials: A video was added as Supplementary Data to provide a better visual of the VAE-UNet performance: https://youtu.be/xtE6pfyCSVo (accessed on 14 February 2025). Video S1: Variational autoencoder tests fleet electrochemical sensors. https://youtu.be/KH9hu6ksXp8 (accessed on 14 February 2025). Video S2: Variational autoencoder tests fleet spectral sensors.

Author Contributions: Conceptualization, A.C.-P. and S.Y.; data curation, A.C.-P.; formal analysis, A.C.-P., S.Y. and D.G.-R.; funding acquisition, S.L.T., M.P.-E. and D.G.-R.; investigation, A.C.-P., S.Y. and D.G.-R.; methodology, A.C.-P.; project administration, S.L.T., M.P.-E. and D.G.-R.; resources, S.Y. and D.G.-R.; software, A.C.-P. and S.Y.; supervision, S.L.T., M.P.-E. and D.G.-R.; validation, A.C.-P., S.Y., S.L.T., M.P.-E. and D.G.-R.; visualization, A.C.-P., S.Y. and D.G.-R.; writing—original draft, A.C.-P.; writing—review and editing, A.C.-P., S.Y., S.L.T., M.P.-E. and D.G.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article and Supplementary Material. The dataset can be downloaded from https://bender.us.es/acasado/dataset-vae (accessed on 14 February 2025).

Acknowledgments: The authors would like to thank the following projects: TED2021-131326B-C21 of the Spanish Ministry of Science MCIN/AEI/10.13039/501100011033, the European Union with "NextGenerationEU/PRTR" funds, TED2021-131326A-C22 of the Spanish Ministry of Science MCIN/AEI/10.13039/501100011033, the European Union with "Next Generation EU/PRTR" funds, and the "Junta de Andalucía: Consejería de Universidad, Investigación e Innovación" through the project "Monitorization of Environmental Dangers with Unmanned Surface Agents: (MEDUSA)" under Grant PCM_00019. Activity: C23.I1.P03.S01.01 Andalucía Public subsidy for the development of the «INVESTIGO Programme», financed with funds from the «Recovery and Resilience Mechanism», Regional Government of Andalucía and the European Union.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two Dimensional
AIPP	Adaptative Informative Path Planning
ASV	Autonomous Surface Vehicle
MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism
VAE	Variational Autoencoder
CNN	Convolutional Neural Network
SDG	Sustainable Development Goal
MSE	Mean Square Error
KL	Kullback-Leibler

References

1. Zhang, L.; Yu, X.; Zhou, T.; Zhang, W.; Hu, S.; Clark, R. Understanding and attribution of extreme heat and drought events in 2022: Current situation and future challenges. *Adv. Atmos. Sci.* **2023**, *40*, 1941–1951. [CrossRef]

2. Statistics, U. Global indicator framework for the sustainable development goals and targets of the 2030 agenda for sustainable development. In *Developmental Science and Sustainable Development Goals for Children and Youth*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 439.
3. Tariq, A.; Mushtaq, A. Untreated wastewater reasons and causes: A review of most affected areas and cities. *Int. J. Chem. Biochem. Sci.* **2023**, *23*, 121–143.
4. Mercado, J.M.; Gómez-Jakobsen, F.; Korbee, N.; Aviles, A.; Bonomi-Barufi, J.; Munoz, M.; Reul, A.; Figueroa, F.L. Analyzing environmental factors that favor the growth of the invasive brown macroalga Rugulopteryx okamurae (Ochrophyta): The probable role of the nutrient excess. *Mar. Pollut. Bull.* **2022**, *174*, 113315. [CrossRef]
5. Dettori, M.; Arghittu, A.; Deiana, G.; Castiglia, P.; Azara, A. The revised European Directive 2020/2184 on the quality of water intended for human consumption. A step forward in risk assessment, consumer safety and informative communication. *Environ. Res.* **2022**, *209*, 112773. [CrossRef]
6. Qu, J.; Fan, M. The current state of water quality and technology development for water pollution control in China. *Crit. Rev. Environ. Sci. Technol.* **2010**, *40*, 519–560. [CrossRef]
7. Kwon, D.Y.; Kim, J.; Park, S.; Hong, S. Advancements of remote data acquisition and processing in unmanned vehicle technologies for water quality monitoring: An extensive review. *Chemosphere* **2023**, *343*, 140198. [CrossRef] [PubMed]
8. Mittal, K.; Gill, K.S.; Upadhyay, D.; Singh, V.; Chythanya, K.R. Neural Networks Utilization for Oil Spill Classification Using a Sequential CNN Model. In Proceedings of the 2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM), Noida, India, 21–23 February 2024; pp. 1–5. [CrossRef]
9. Bogue, R. Robots for monitoring the environment. *Ind. Robot. Int. J.* **2011**, *38*, 560–566. [CrossRef]
10. Storey, M.V.; Van der Gaag, B.; Burns, B.P. Advances in on-line drinking water quality monitoring and early warning systems. *Water Res.* **2011**, *45*, 741–747. [CrossRef]
11. Valada, A.; Velagapudi, P.; Kannan, B.; Tomaszewski, C.; Kantor, G.; Scerri, P. Development of a low cost multi-robot autonomous marine surface platform. In *Field and Service Robotics: Results of the 8th International Conference*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 643–658.
12. Luis, S.Y.; Reina, D.G.; Marín, S.L.T. A Deep Reinforcement Learning Approach for the Patrolling Problem of Water Resources Through Autonomous Surface Vehicles: The Ypacarai Lake Case. *IEEE Access* **2020**, *8*, 204076–204093. [CrossRef]
13. Popović, M.; Ott, J.; Rückin, J.; Kochenderfer, M.J. Learning-based methods for adaptive informative path planning. *Robot. Auton. Syst.* **2024**, *179*, 104727. [CrossRef]
14. Arzamendia, M.; Gregor, D.; Reina, D.G.; Toral, S.L. An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of Ypacarai Lake. *Soft Comput.* **2019**, *23*, 1723–1734. [CrossRef]
15. González-Reolid, I.; Molina-Molina, J.C.; Guerrero-González, A.; Ortiz, F.J.; Alonso, D. An autonomous solar-powered marine robotic observatory for permanent monitoring of large areas of shallow water. *Sensors* **2018**, *18*, 3497. [CrossRef] [PubMed]
16. Fuchi, T.R.; Mokam, L.; Kombe, T. Drone equipment and configuration for crude oil spill detection in water. *Eur. J. Technol.* **2022**, *6*, 1–14. [CrossRef]
17. Dolenko, T.A.; Fadeev, V.V.; Gerdova, I.V.; Dolenko, S.A.; Reuter, R. Fluorescence diagnostics of oil pollution in coastal marine waters by use of artificial neural networks. *Appl. Opt.* **2002**, *41*, 5155–5166. [CrossRef] [PubMed]
18. Lin, B.; An, J.; Carl, B.; Zhang, H. Neural Networks in Detection and Identification of Littoral Oil Pollution by Remote Sensing. In Proceedings of the Advances in Neural Networks—ISNN, Dalian, China, 19–21 August 2004; Yin, F.L., Wang, J., Guo, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 977–982.
19. Carazo-Barbero, G.; Besada-Portas, E.; Girón-Sierra, J.M.; López-Orozco, J.A. EA-based ASV Trajectory Planner for Pollution Detection in Lentic Waters. In *Applications of Evolutionary Computation: Proceedings of the 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual, 7–9 April 2021*; Proceedings 24; Springer: Berlin/Heidelberg, Germany, 2021; pp. 812–827.
20. Luis, S.Y.; Peralta, F.; Córdoba, A.T.; del Nozal, Á.R.; Marín, S.T.; Reina, D.G. An evolutionary multi-objective path planning of a fleet of ASVs for patrolling water resources. *Eng. Appl. Artif. Intell.* **2022**, *112*, 104852. [CrossRef]
21. de Silva, I.; de Souza, A.; Hossri, J.; Zago, M. Identification of the level of contamination and degradation of oil by artificial neural networks. In Proceedings of the Conference Record of the 2000 IEEE International Symposium on Electrical Insulation (Cat. No.00CH37075), Anaheim, CA, USA, 5 April 2000; pp. 275–279. [CrossRef]
22. Bahadur, R.; Amstutz, D.E.; Samuels, W.B. Water contamination modeling—A review of the state of the science. *J. Water Resour. Prot.* **2013**, *5*, 27683. [CrossRef]
23. Sang, J.; Liu, Z.; Ding, X.; Yin, Y. Prediction model for pollution accidents trend in drinking water source areas: Enhancing water safety and comprehensive applications. *Process. Saf. Environ. Prot.* **2024**, *184*, 12–24. [CrossRef]
24. Banerjee, A.; Creedon, L.; Jones, N.; Gill, L.; Gharbia, S. Dynamic Groundwater Contamination Vulnerability Assessment Techniques: A Systematic Review. *Hydrology* **2023**, *10*, 182. [CrossRef]

25. Kathen, M.J.T.; Flores, I.J.; Reina, D.G. An informative path planner for a swarm of asvs based on an enhanced pso with gaussian surrogate model components intended for water monitoring applications. *Electronics* **2021**, *10*, 1605. [CrossRef]
26. Luis, S.Y.; Basilico, N.; Antonazzi, M.; Gutiérrez-Reina, D.; Marín, S.T. Variational model-based Deep Reinforcement Learning for Non-Homogeneous Patrolling aquatic environments with multiple unmanned surface vehicles. *Expert Syst. Appl.* **2025**, *270*, 126483. [CrossRef]
27. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: Proceedings of the 18th international conference, Munich, Germany, 5–9 October 2015*; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
28. Kohl, S.A.A.; Romera-Paredes, B.; Meyer, C.; Fauw, J.D.; Ledsam, J.R.; Maier-Hein, K.H.; Eslami, S.M.A.; Rezende, D.J.; Ronneberger, O. A Probabilistic U-Net for Segmentation of Ambiguous Images. *arXiv* **2019**, arXiv:1806.05034.
29. Peralta, F.; Reina, D.G.; Toral, S. Towards an online water quality monitoring system of dynamic environments using an autonomous surface vehicle. In Proceedings of the International Conference on Optimization and Learning (OLA2022), Sicilia, Italy, 18–20 July 2022; Science Conferences.
30. El-Magd, I.A.; Zakzouk, M.; Ali, E.M.; Abdulaziz, A.M. An open source approach for near-real time mapping of oil spills along the mediterranean coast of Egypt. *Remote Sens.* **2021**, *13*, 2733. [CrossRef]
31. Hodgson, Z.; Esnaola, I.; Jones, B. Model-Based Optimal Adaptive Monitoring of Oil Spills. *IEEE Trans. Control Syst. Technol.* **2022**, *30*, 2115–2130. [CrossRef]
32. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. In *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook*; Springer International Publishing: Cham, Switzerland, 2023; pp. 353–374.
33. Doersch, C. Tutorial on variational autoencoders. *arXiv* **2016**, arXiv:1606.05908.
34. Yi, K.; Guo, Y.; Fan, Y.; Hamann, J.; Wang, Y.G. Cosmo vae: Variational autoencoder for cmb image inpainting. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
35. Guo, Z.; Yang, G.; Zhang, D.; Xia, M. Rethinking gradient operator for exposing AI-enabled face forgeries. *Expert Syst. Appl.* **2023**, *215*, 119361. [CrossRef]
36. Johnson, J.; Alahi, A.; Li, F.-F. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Computer Vision—ECCV 2016: Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 694–711.
37. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
38. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019.
39. Yanes-Luis, S. Deep Reinforcement Learning for Adaptive Monitarizacion and Patrolling of Water Resources with Unmanned Surface Vehicles. Ph.D. Thesis, Universidad de Sevilla, Sevilla, Spain, 2024.
40. Dias, A.; Mucha, A.; Santos, T.; Oliveira, A.; Amaral, G.; Ferreira, H.; Martins, A.; Almeida, J.; Silva, E. Oil Spill Mitigation with a Team of Heterogeneous Autonomous Vehicles. *J. Mar. Sci. Eng.* **2024**, *12*, 1281. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.