ORIGINAL PAPER

Identification of moving sources in stochastic flow fields: A bayesian inferential approach with application to marine traffic in the mediterranean sea

Issam Lakkis[1] · Alexios Rustom[1] · Mohamad Abed El Rahman Hammoud[2] · Leila Issa[3] · Omar Knio[4] · Olivier Le Maitre[5] · Ibrahim Hoteit[6]

Published online: 10 April 2025
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2025

Abstract
A Bayesian inference approach for inferring the source of marine pollution released from a moving source in an uncertain flow field is proposed. A Markov Chain Monte Carlo (MCMC) algorithm is developed and applied for inferring single and multiple release events from vessels moving at known velocity along a predefined path in the Mediterranean Sea. The likelihood is based on a logistic regression cost function that measures the discrepancy between the modeled spill distribution and a binary representation of the observed images. We assess the performance of the proposed methodology using a synthetic release scenario employing realistic ocean currents to drive a stochastic Lagrangian Particle Tracking (LPT) algorithm to generate a probabilistic representation of the spill distribution. The MCMC algorithm employs an adaptive scheme to robustly ensure convergence and well-mixed chains. The proposed Bayesian framework is tested by inferring the location, or injection time, and relative contributions of single and multiple moving sources, contributing to separate and common observation patches, with a focus on various scenarios that demonstrate the efficiency of our sampling algorithm. The performance of the proposed framework was further assessed by comparing the model predictions with the most probable release parameters predicted by a global optimization algorithm.

Keywords Uncertainty quantification · Bayesian inference · Source reconstruction · Stochastic flow field · Moving sources · Marine pollution · Mediterranean sea

1 Introduction

Marine pollution poses a critical risk to living resources, human health and marine activities. Recent research is primarily focused on marine oil, plastic and microplastic pollution, where these are considered as the major hazards to the marine ecosystem [1]. For instance, Jambeck et al. [2] reports that millions of tons of plastic waste enters the sea each year. Furthermore, microplastics have emerged as a hazardous marine pollutant, where because of its small size (<5mm), it is able to interact with marine organisms, such as zooplankton and and phytoplankton and make its way, in traces, up the food chain reaching the human diet [3, 4]. In addition, oil spills have posed a threat to marine ecosystems worldwide with an average of 1.25 million tons of oil being released annually from sea-based sources [5].

The Mediterranean Sea, hereafter referred to as MS, has one of the most densely populated coastlines, hosting a range of activities including tourism, fishing, shipping and industrial applications, to name a few. It is also one of the busiest waterways being responsible for approximately 15% of the global shipping activity [6], and offers a

passageway for different types of ships, such as recreational, shipping and oil tankers [7]. Furthermore, since the MS is a semi-enclosed basin with limited outflow of surface waters [8, 9], it is generally considered as a hotspot for plastic pollution [10, 11]. Unfortunately, the MS has further experienced a number of catastrophic marine pollution incidents, for instance, [12] report that 14 oil spill incidents have occurred in the Med between 1970 and 2016 resulting in a total of 10 tons of oil contaminating the sea. The largest oil spill incident occurred on April 11, 1991, when an explosion aboard the MT Haven tanker resulted in the spill of approximately 50,000 tonnes of crude oil along the coast of Genoa, Italy.

Identifying the potential sources of marine pollution is crucial to contain the associated risk and mitigate its consequences [13]. Furthermore, efficient identification of marine pollution sources is important to set policies to take reasonable measures against potential culprits. Extensive studies have been conducted for the purpose of determining quickly and accurately the probable sources of contaminants; e.g. [14]. This inverse problem, also known as the Source Term Estimation (STE) problem, involves the identification of the source characteristics given some set of observed or measured data [15, 16]. Two different approaches [17] have been proposed to tackle the source reconstruction problem: the deterministic optimization approach that seeks obtaining a single optimized solution for the inverse source reconstruction problem without taking into account the uncertainties associated with the release event, and the stochastic Bayesian approach that yields a probabilistic representation of the source parameters and quantifies the degree of plausibility of possible solutions.

STE problems are ill-posed, and have been applied in different contexts including the identification of sources of pollutants in the atmosphere [18, 19], and land mines [20] to name a few. In oceanic settings, STE problems have been generally tackled using backward in time integration methodology. For example, Mohtar et al. [21] examined the identification of a single source location in the Red Sea in the presence of uncertain currents. Zodiatis et al. [22] also relied on a backward integration machinery to infer multiple sources in a deterministic current field. Furthermore, Hammoud et al. [23] investigated the identification of a single moving source in the MS using backward integration in the presence of uncertainties.

Bayesian inference methodologies, however, were limited to parameter calibration of fixed sources, such as location and strength in idealized settings. For instance, it was used in STE problems for determining the origin and decay rate of a non-conservative scalar [24], parameters of a known single fixed source in a complex urban environment [25], or known fixed release source scenarios in simplified source-detector configurations [26, 27]. Similarly, Xue et al. [28] adopted the Bayesian inference framework for source identification involving wind tunnel experiments numerically simulated using large eddy simulations. On larger scales, Yee et al. [29] inferred single fixed source parameters in a real world application problem, while Kopka and Wawrzynczak [30] investigated several sampling algorithms to identify the model parameters of a single fixed source of atmospheric contaminants in Copenhagen. Yee et al. [25] also inferred the parameters of a single fixed source on a European continental level. In a related work, Kopka et al. [31] investigated a Bayesian methodology for the inference of the model parameters of a mobile source releasing atmospheric contaminants over an 8 min period in a specific day in the presence of a deterministic time-dependent wind field.

Considering uncertainties in the flow field is crucial for studying marine pollution as it helps model the unpredictable dispersion and transport of pollutants, building confidence in the predictions of contamination pathways [32]. Accounting for uncertainties enhances the reliability of risk assessments and informs effective mitigation strategies, ensuring a comprehensive and resilient approach to addressing marine pollution challenges [33, 34]. In the context of particle tracking, uncertainties are generally considered in nonlinear processes pertaining to the Lagrangian particle or the oceanic flow field [35–38]. Uncertainties in the flow field often stem from the imperfect descriptions of oceanic forcing, internal physics, and initial conditions [39, 40]. Flow field uncertainties are commonly accounted for by advecting particles using an ensemble of flow field realizations, where at each time step, an ensemble of velocity field realizations representing a finite set of potential scenarios are employed. This strategy has been adopted in the context of atmospheric [40] and marine pollution [21, 23].

In this work, we introduce and implement a novel source reconstruction approach with an application to Marine traffic in the MS in the presence of an uncertain velocity field. A new Bayesian inference methodology is also

proposed, allowing the inference of the parameters of single and multiple release events due to a surface vessel moving along a known pathway. The approach is based on introducing a likelihood function derived from information theory to describe the discrepancy between observed images and model predictions. In addition, a new MCMC sampling algorithm is proposed, which enables us to effectively build a posterior density and consequently characterize the properties of the release event(s). The performance of the resulting algorithm is analyzed in both simplified and realistic settings, and the predictions are contrasted with results obtained using a global optimization algorithm.

The manuscript is organized as follows. Section 2 provides a description of the ensemble of ocean current fields used in our numerical experiments. In Section 3 outlines the numerical model employed to track the evolution of the released quantities. Section 4 presents the marine traffic database adopted to identify a high-density pathway along which release events are chosen. We then present in details the proposed Bayesian inference algorithm, and discuss key elements of its construction in Section 7, followed by the analysis of our numerical results and performance of the resulting inversion algorithm in Section 8. Finally, main conclusions are summarized in Section 10.

2 Ocean current ensemble

To generate an uncertain flow field, an ensemble of realizations of the MS currents was generated using 30 years of reanalysis fields from the Copernicus Marine Services (CMS). The CMS data assimilated reanalysis fields for the basin covers the sea from 6°W to 36.25°E, 30.17°N to 45.9375°N and reaches a depth of 5334.65m below sea level. We consider the time period between July 14^{th} and August 12^{th}, which covers the period of the Jiyyeh oil spill that occurred in 2006. The realizations are randomly sampled from a Gaussian distribution, $\mathcal{N}$, with a covariance computed from the CMS fields over the period 1987-2016 and a mean ocean current equal to that of the CMS reanalysis fields for that given day for the year 2006. The covariances of the CMS currents over the 30 years were computed using the a window of ± 1 day about the day of interest for a total of 90 realizations.

Following the procedure in [23], the small-scale features are filtered out using an Empirical Orthogonal Function (EOF) strategy. Specifically, we only retain the first m dominant modes of the Empirical covariance, where m is chosen to capture 95% of the empirical variability. The generation of the ensemble members of the flow fields is achieved by taking a low-rank approximation of the covariance matrix as $\tilde{C} = \tilde{L}\tilde{L}^T$. Each realization of the longitudinal and latitudinal currents is then sampled as:

$$u_i^d = u^{d,y} + \tilde{L}\gamma_i^d, \quad \gamma_i^d \sim \mathcal{N}(0, I_{M\times M}) \tag{1}$$

where d and y are respectively the day and year of interest, M the number of eigendirections, $i = 1, 2, \cdots, N_e$ is the index of the realization, and N_e is the ensemble size. Here, $N_e = 50$, $d \in$ July 14 - August 12 and $y = 2006$. The uncertain velocity field is represented on a grid as:

$$\vec{u}_i(\vec{x}_g, t_a), \ t_a = nT_a, \ n \in \mathbb{N}, \ i = 1, \ldots, N_e \tag{2}$$

where $\vec{x}_g$ are the grid coordinates, and the assimilation time, t_a, is an integer multiple of the assimilation interval, T_a, taken to be 1 day. The indices i refer to the sampled realizations. The computational grid is uniformly spaced horizontally with a resolution of $0.0625° \times 0.0625°$, and non-uniformly spaced vertically with 72 depth levels, for a computational mesh of size $677 \times 253 \times 72$ grid points.

3 Particle transport model

Given a known release source location and release time $(\vec{x}_s, t_R)$, the forward map consists in identifying probable locations of passive particles carried by the ensemble of velocity fields, $\vec{u}_i(\vec{x}_g, t_a)$. We denote by $P(\vec{x}, t; \vec{x}_s, t_R)$

the probability density at $t > t_R$ generated by Lagrangian tracking of passive tracers, originating from release sources located at $\vec{x}_s$, and advected with an ensemble of flow fields. To this end, we employ the Lagrangian Particle Tracking (LPT) method presented in [21]. This LPT relies on an efficient parallel implementation of a high-order particle advection scheme. The method overcomes the challenge of exponential growth in the number of particles which arise from considering all possible combinations of the different realizations belonging to the sequence of ensembles. To this end, it incorporates an adaptive binning procedure that conserves the zeroth, first and second moments of probability (total probability, mean position, and variance). The adaptive binning procedure offers a trade-off between speed and accuracy by limiting the number of particles to a desired maximum. Although the proposed framework can be applied to any transport model, we adopt a simple transport model that only accounts for advection by the local velocities. As such, this model ignores the influence of other physical phenomena on the transport of released particles.

4 Trajectories and source prior

The moving sources considered in this work are assumed to traverse a single trajectory in the MS. The path is extracted from https://www.marinetraffic.com, a platform that provides density maps showcasing global vessel traffic intensity [41]. The path studied here corresponds to the heaviest traffic of oil tankers in the MS starting at the Suez Canal and ending at the Strait of Gibraltar, as illustrated by the automatic identification system. By varying the opacity of the density map as shown in Fig. 1a, the path considered in this work and presented in Fig. 1b was extracted by selecting its coordinates in locations of high density.

We consider a prior that consists of a set of discrete sources lying on a single trajectory. The trajectory is discretized in N_s segments. The segments are associated with pairs of release location and time: $(x, t)_m$, $m = 1, \ldots, N_s$. We define the canonical set of events $\mathcal{E}^* \doteq \{(x, t)_m, m = 1, \ldots, N_s\}$, where N_s is the total number of possible events.

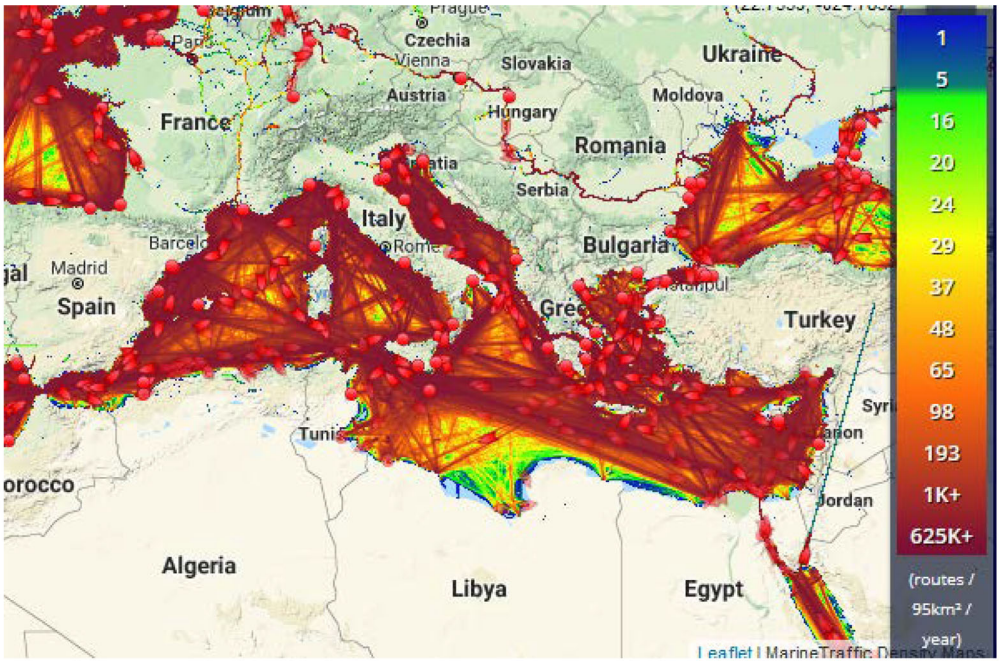
Extension to the case of multiple trajectories along several paths is possible. In this case, the prior is based on N_τ trajectories along N_π paths. Each trajectory $\mathcal{T}_{k,p}$, $k = 1, \ldots, N_\tau$, $p = 1, \ldots, N_\pi$ is discretized in $N_{s,k,p}$ segments. The segments are associated with pairs of release location and time: $(x, t)_{i,k,p}$, $i = 1, \ldots, N_{s,k,p}$. This essentially allows us to invert for the release source location, initial release time and duration through the posterior probability distribution. Introducing the single index $m = i + \sum_{q<p} \sum_{l<k} N_{s,l,q}$, the canonical set of events is defined as $\mathcal{E}^* \doteq \{(x, t)_m, m = 1, \ldots, N_E\}$, where $N_E = |\mathcal{E}^*|$ is the total number of possible events.

5 Forward probability map

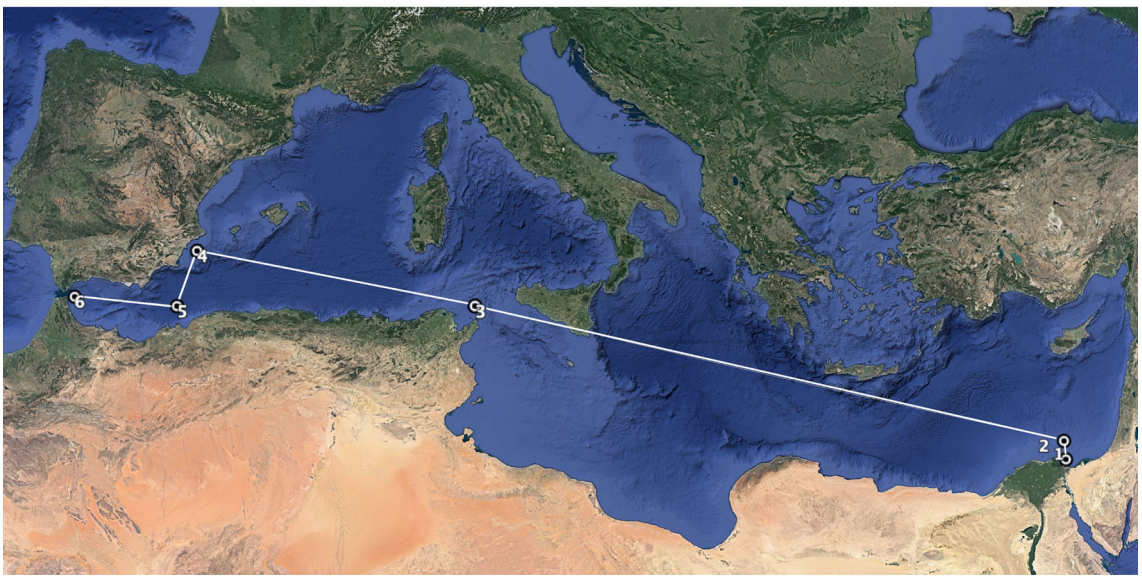
The prior $\mathcal{E}_O$ for an observation at time t_O includes all elements of $\mathcal{E}^*$ satisfying $t_m < t_O$, i.e. $\mathcal{E}_O = \{(\vec{x}, t)_m \in \mathcal{E}^*, t_m < t_O\}$, which follows from the fact that release events that take place after the observation time do not contribute to the observation. Given a release event, $(\vec{x}, t)_m$ and an observation time, $t_O > t_m$, the forward probability map, $P(\vec{x}, t_O; (\vec{x}, t)_m)$, is generated by applying the adaptive LPT method, described in Section 3, in the time interval $[t_m, t_O]$. This amounts to advecting passive tracers originating from $(\vec{x}, t)_m$ by the finite ensemble $\vec{u}_i(\vec{x}_g, t_a)$, $i = 1, \ldots, N_e$.

For each event belonging to $\mathcal{E}_O$, the LPT algorithm is used to compute the forward probability map that characterizes the corresponding spill scenario. The advantage of this approach is that the forward maps of all the individual release events can be generated independently and in parallel, making the associated computational cost of the inference algorithm affordable. The forward map of any combination of release events $(\vec{x}, t)_i, i \in \mathcal{R}$, $\mathcal{R} = \{m_1, m_2, \ldots\}$, can then be constructed as the weighted sum of the forward maps of the individual events

$$P_E(\vec{x}, t_O) = \sum_{m \in \mathcal{R}} w_m P(\vec{x}, t_O; (\vec{x}, t)_m), \tag{3}$$



(a)



(b)

Fig. 1 (a) Density map of marine traffic in the MS, obtained from https://www.marinetraffic.com [41]. (b) The selected path of the ship in the MS. The path starts at the Suez Canal and ends in the Strait of Gibraltar; it is assumed to consist of five straight segments, as illustrated

where w_m is the weight of release event m denoting the ratio of the pollutant amount released in event m to the total amount released in the events $i \in \mathcal{R}$. In other words, letting $Q_i, i \in \mathcal{R}$, denote the pollutant amount released in event i, then

$$w_m = \frac{Q_m}{\sum_{i \in \mathcal{R}} Q_i}. \tag{4}$$

The objective is to infer the set, $\mathcal{I}$, of discrete release events $i \in \mathcal{I}$, $(\vec{x}, t)_i \in \mathcal{E}_O$, together with the associated weights, w_i, contributing to a given observation.

6 Observations

The observations considered in this work are ideally satellite images of pollutant patches. Mapping such images to the probability distributions obtained from the forward map is a challenge, because the former is an optical measure of the pollutant spatial distribution in the form of intensity $I(\vec{x}, t_O), \vec{x} \in \Omega_O$, where Ω_O is the spatial observation domain, whereas the latter is a probability distribution. To overcome this challenge, we map the intensity to a scalar function Y, which we call the "indicator function" henceforth and define as follows:

$$Y(\vec{x}, t_O) = 1 \quad \text{if} \quad I(\vec{x}, t_O) > I^*,$$
$$0 \quad \text{otherwise},$$

With the definition above, a value $Y = 1$ indicates the presence of the pollutant, whereas $Y = 0$ indicates its absence. The cutoff value of the intensity, I^*, is chosen to be a small fraction of the maximum intensity. Note that, in this work, we adopt a conservative cutoff where every grid cell with an intensity greater than $I^* = 10^{-3}$ is accepted as potentially a pollutant. An example of the mapping from I to Y is shown Fig. 2-a to b.

7 Bayesian inference

Bayesian methods enable identification of model parameters, given limited noisy observations and some prior knowledge on the parameters [42]. These approaches are based on Bayes' rule

$$\mathcal{P}(\mathbf{M}|\mathbf{D}) \propto \mathcal{L}(\mathbf{D}|\mathbf{M})\, P(\mathbf{M}), \tag{5}$$

which expresses the posterior probability, $\mathcal{P}$, of the model parameters, $\mathbf{M}$, given the observations $\mathbf{D}$ and the prior distribution, P, of the model parameters. The likelihood $\mathcal{L}(\mathbf{D}|\mathbf{M})$ denotes the probability of recovering the observations with the forward model given $\mathbf{M}$ as input. For our problem, the model parameters $\mathbf{M}$ to be inferred are the set, $\mathcal{I}$, of discrete release events $i \in \mathcal{I}$, $(\vec{x}, t)_i \in \mathcal{E}_O$, together with the associated weights, w_i. The observations $\mathbf{D}$ are represented by the observation function $Y(\vec{x}, t_O)$ defined in Section 6. The prior $\mathcal{E}^\star$ is the collection of possible release events, which are all treated as equi-probable. For each possible release event belonging to the prior, the forward map is constructed using the stochastic LPT algorithm, as described in Section 5. The algorithm caps the number of particles at a maximum of 10^7 and uses an advection time step that is equal to 3 hours. These values, informed by studies reported in [21], offer a trade off between the computational cost and accuracy.

In the case of complex high-dimensional models, the posterior PDF is high dimensional and is commonly not analytically tractable. In this case, the posterior distribution is numerically estimated by adequately sampling over the model parameters space, generally using a Markov Chain Monte Carlo (MCMC) method, [43]. MCMC algorithms start with an initial guess of the model parameters. Randomly generated samples of the model parameters are then either accepted or rejected based on some acceptance criterion. The accepted samples, generated from a Markov chain whose distribution will eventually tend, asymptotically with the number of accepted samples, to the desired posterior distribution.

In the remainder of this section, we first define the likelihood and introduce a cost function that is suitable for contrasting the output of the LPT model with the observation field. We then outline an MCMC sampling algorithm that we use to simulate the posterior.

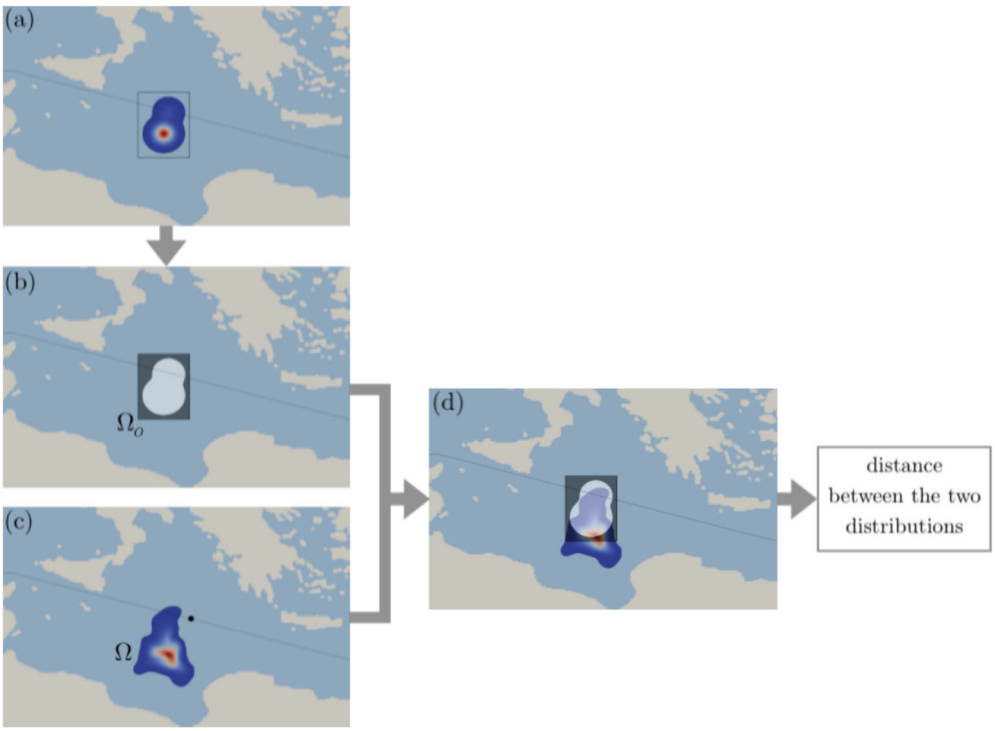


Fig. 2 (a) Simulated image of a pollutant patch. (b) Observation function discretized on a rectangular grid containing the observation patch. (c) Forward probability map of release event denoted by filled black circle. (d) Sets $\mathcal{Y}$ and $\mathcal{Z}$ of the observation grid cells and the forward probability map are used to calculate the cost function measuring the distance between the observation and model output

7.1 Likelihood

For a realization of the model parameters, the likelihood measures how close the model prediction is to the observation(s). In Section 6, we expressed the observations in terms of the "indicator function", Y, to render them compatible with the model output, expressed in terms of the forward probability map, presented in Section 5.

In order to compute the distance between the observation function and the forward map, the observation function is discretized on a rectangular grid as shown in Fig. 2-b. The extension of this rectangular domain from a tight-fit rectangle is denoted by B. Note that unless otherwise specified, we use $B = 0$, associated with the smallest number of cells with $Y = 0$, will be used in the numerical experiments. We denote by $\mathcal{Y}$ the set of all grid cells i in the rectangular domain Ω_O for which $Y(\vec{\xi}_i) = 1$, where $\vec{\xi}_i$ is the position vector of cell i. Similarly, we denote by $\mathcal{Z}$ as the set of all grid cells j in the rectangular domain Ω_O for which $Y(\vec{\xi}_j) = 0$.

The distance between the observation (e.g. Fig. 2-b) and model output (e.g. Fig. 2-c) is measured using the binary cross-entropy loss function [44], which is the basis of the standard loss function for logistic regression in classification problems in machine learning applications [45, 46].

Given the observation $Y(\vec{x}, t_O)$, a model realization corresponding to the events $l \in \mathcal{S}, (\vec{x}, t)_l, \in \mathcal{E}_O$ and associated weights w_l, the cost function is expressed as

$$\mathbb{J}(\mathbf{D}|\mathbf{M}) = -\sum_{i\in\mathcal{Y}} \log\left(\sum_{l\in\mathcal{S}} w_l \frac{P(\vec{\xi}_i, t_O; (\vec{x}, t)_l)}{C}\right) - \sum_{j\in\mathcal{Z}} \log\left(1 - \sum_{l\in\mathcal{S}} w_l \frac{P(\vec{\xi}_j, t_O; (\vec{x}, t)_l}{C}\right), \tag{6}$$

where $P(\vec{\xi}, t_O; (\vec{x}, t)_l)$ is the forward probability at a observation grid point $\vec{\xi}$ and observation time t_O due to release event $(\vec{x}, t)_l$.

The scaling coefficient, C, is determined so that the integral of the scaled probability over the forward probability map domain, Ω, is equal to the integral of the observation function Y over the observation patch domain, Ω_O, (see Fig. 2-d) i.e.

$$\sum_{l\in\mathcal{S}} \int_{\Omega} w_l \frac{P(\vec{\xi}, t_O; (\vec{x}, t)_l)}{C} dA = \int_{\Omega_O} Y(\vec{\xi}, t_O) dA. \tag{7}$$

Noting that the integral on the right is the fraction of the observation area for which $Y = 1$, and upon using the observation grid as a subset of the grid discretizing Ω, C can be numerically approximated as

$$C \simeq \frac{1}{N_{\mathcal{Y}}} \sum_{l\in\mathcal{S}} w_l \sum_{j=1}^{N_\Omega} P(\vec{\xi}_j, t_O; (\vec{x}, t)_l)), \tag{8}$$

where $N_{\mathcal{Y}}$ is the number of cells in $\mathcal{Y}$ and N_Ω is the number of grid cells discretizing Ω. Note that only cells with non-zero probability need to be included in the inner summation.

Using the cost function defined in Eq. 6, we model the likelihood probability as an exponentially decaying function of the distance between the observation and model output [24], expressed as

$$\mathcal{L}(\mathbf{D}|\mathbf{M}) = \exp\left(-\frac{\mathbb{J}(\mathbf{D}|\mathbf{M}^{(s)})}{\lambda}\right), \tag{9}$$

where λ is a hyperparameter that is adaptively tuned to enhance convergence of the MCMC sampling algorithm. Intuitively, one would expect that the performance of the algorithm would peak whenever λ scales with the range of the cost function. The impact of a suitable initialization of the hyperparameter is assessed in light of numerical experiments presented below.

We finally point out the crucial role that the relative weights play in dealing with scenarios involving multiple release events, particularly in guiding the MCMC sampling algorithm (presented in Section 7.2) towards lower values of the cost function. In the MCMC algorithm, any sampled release event along the path of the moving source that is not contributing to the observed patch will end up with a relative weight of zero. This effectively endows the proposed inference methodology with the ability to infer the number of sources, in addition to their location and relative weights. As a result, one does not require sophisticated techniques, such as the reversible jump MCMC algorithm in [47] or with a simulated annealing strategy in [48]. Instead, we employ simpler yet efficient strategies in the sampling process, as described next.

7.2 The sampling algorithm

In order to infer for the probable release events contributing to the observations, as well as their relative contributions, an accept-reject sampling algorithm is introduced. The latter is implemented in a similar way to the Metropolis Hastings algorithm ([49–51]), and aims to determine the distribution of the vector of sampled model parameters, $\mathbf{M}$, which consists of the set of sampled events $\mathcal{S}$ and corresponding sampled weights $w_l, l \in \mathcal{S}$. This is provided in the form of N_c parallel and independent chains of accepted samples, $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \ldots$. To ensure that the acceptance ratio (AR) is within the desired range, the algorithm updates a hyperparameter λ every M_s samples, where M_s is an input to the algorithm. Once the AR criterion is satisfied, the algorithm uses only the last M_s samples of each of the N_c chains for inference statistics.

To construct the chain, the sampling algorithm explores random jumps in the space of model parameters. Unlike conventional MCMC algorithms that generally perturb all components of the parameter vector, the presently-introduced algorithm samples one component of $\mathbf{M}$ in each jump. This approach avoids the obvious difficulty of defining suitable proposal distributions in the multi-dimensional parameter space, which may lead to rejection of a large proportion of proposed samples and/or poorly mixed chains To define the jumps in individual parameters,

normal proposal distributions are used, with standard deviation σ_e when the events are sampled and standard deviation σ_w when the weights are sampled. Note that the vector of relative weights is normalized so that its l_1 norm is always equal to one. Also, in case the proposed model parameters fall outside their respective ranges, we resample until the proposed parameter vector becomes admissible.

At any position $\mathbf{M}^{(c)}$ in the chain, the proposed sampling algorithm is used to generate new values of the model parameters. To decide whether a proposal is accepted or not, we (i) evaluate the likelihood ratio

$$AC = \frac{e^{-\frac{\mathbb{J}(\mathbf{D}|\mathbf{M}^{(p)})}{\lambda}}}{e^{-\frac{\mathbb{J}(\mathbf{D}|\mathbf{M}^{(c)})}{\lambda}}} \tag{10}$$

(ii) randomly draw a scalar, α, from a uniform distribution over [0, 1], and (iii) test whether $AC > \alpha$. In this case, the proposal is accepted, i.e. $\mathbf{M}^{(p)}$ is included as a new sample in the chain; otherwise, a new proposal is made and the process is repeated. The iterations are carried out until the number of accepted samples in a given chain reaches the preset value, and the number chains is chosen to be sufficiently large to ensure a convergence and well-fixed conditions. Note that as further discussed below, during the iterations we adapt the values of the hyperpameter λ and the standard deviation σ_e of the events' proposal distribution. This is performed in order to enhance the efficiency of the algorithm and to ensure a well-mixed chain.

7.2.1 The hyperparameter λ

To illustrate the impact of λ on the behavior of the algorithm, we consider an inference problem involving two distinct release events, specified as Experiment 2 in Table 1. Figure 3 shows profiles of the cost function plotted against accepted samples, for different fixed values of λ. It can be seen from Fig. 3a that when the hyperparameter λ is large ($\lambda = 500$), the cost function of the accepted samples in the MCMC chain undergoes abrupt changes and is biased to high values. For small $\lambda = 1$, Fig. 3b shows that the cost function fluctuates around the minimum value with no significant changes (the highest encountered difference in the cost function of the consecutive samples is around 8). This indicates the need for an optimum value of λ (in this case, $\lambda = 15$) that yields relatively significant changes in the cost function of the accepted samples, as illustrated in Fig. 3c, without causing the algorithm to bias the cost function to regions of high or low values.

To ensure that the value of λ remains in a suitable range, an adaptive algorithm is used, based on monitoring the acceptance rate, AR, of the samples generated by the MCMC algorithm. Algorithm 1 adapts the value of λ every M_s samples as follows. If the AR is less than 30%, λ is increased by a factor $l_\lambda > 1$ to increase the acceptance rate of samples. On the other hand, if the AR is greater than 50%, λ is decreased by a factor $1/u_\lambda$, where $u_\lambda < 1$. The values of the fixed parameters l_λ and u_λ are specified by the user. The adaptation of the value of λ keeps the AR in the desired range, as depicted in Fig. 4 which shows the values of λ adapted every M_s samples for Experiment 3 in Table 1 along with the corresponding AR for one of the (parallel) chains.

7.2.2 The standard deviations

Mixing of the chains is enhanced by adapting the standard deviation of the proposal distribution of the event indices, σ_e, according to the autocovariances at lag 0, $s_0^{(l)}$, of the sequences of sampled events indices for release events $l \in \mathcal{S}$. The autocovariance is computed for sequences of M_u samples of indices $i = (k-1)M_u + 1, \ldots, kM_u$, $k = 1, 2, \ldots$, where M_u is an input to the algorithm. For each chain, Algorithm 2 starts with a large value of σ_e to allow for big jumps in the sample event indices, which enables the algorithm to efficiently explore the event space. The value of σ_e is updated every M_u samples as follows. If the autocovariance is zero, then the chain of the samples has been locked on a sample event for the past M_u samples, and the algorithm is not accepting new model parameters. This is because sampling with a large value of σ_e introduces sample events far from the set of contributing events, and as such these events will be rejected, which locks the chain and negatively impacts the mixing of the Markov

 https://doi.org/10.1007/s10596-025-10350-0

Table 1 Parameters of the observations patches (described in Section 8.2) and the MCMC and sampling algorithms (Algorithms 1 and 2) used in the experiments

Experiment	1	2	3	4
Observations patches				
No. of sources	2	2	1	4
Obs. type	OP1	OP1	OP2	OP2
Indices of events sequences	300, 400	549, 599	349 − 353	200 − 203 349 − 353 668 − 671 799 − 802
Release times (days)	9.222, 9.917	8.188, 7.840	9.549 − 9.576	10.590 − 10.611 9.549 − 9.576 7.340 − 7.361 6.431 − 6.451
Figures	6, 7, 8		9, 10, 11	12, 13, 14
Sampling algorithm				
n_{Chains}	5	5	5	5
u_λ	0.5	0.5	0.1	0.5
l_λ	2	2	2	2
M_s	10000	10000	10000	10000
M_u	100	50	10	50
f_σ	0.7	0.7	0.7	0.3
burn-in no. of samples	1000	1000	1000	1000
Inferred events				
$\bar{r} \pm 1$ std.	[299, 307] [392, 393]	[544, 546] [595, 596]	[345, 354]	[201, 206] [347, 352] [665, 672] [795, 800]
r_{gs}	302, 393	544, 595	349	205 349 670 798

Indices of the most probable inferred events are also listed along with the those within one standard deviation from the mean of the marginal posterior event index distribution

chain. To unlock the chain and improve mixing, σ_e is decreased by a factor $1/f_\sigma$, where $f_\sigma < 1$ is specified by the user, as presented in Algorithm 2. An example is shown in Fig. 5, where the value of σ_e, adapted every $M_u = 100$ samples, is plotted against the sample index over one of the parallel chains for Experiment 3 listed in Table 1.

Algorithm 1 Pseudocode of the MCMC algorithm.

Run N_{chains} $(c = 1, \cdots, N_{chains})$
Initialize $\mathbf{M}^{(0)}(c) = \left[r_1^{(0)}, r_2^{(0)}, \cdots, r_{N_s}^{(0)}, w_1^{(0)}, w_2^{(0)}, \cdots, w_{N_s}^{(0)}\right](c)$ and M_s
Set $(\sigma_e^{(0)}(c), \sigma_w^{(0)}(c))$
Set $\lambda^{(0)}(c)$, $u_\lambda(< 1)$, and $l_\lambda(> 1)$
while $AR_0 < 0.3$ or $AR_0 > 0.5$ **do**
    $[\mathbf{M}^{(M_s)}(c)$,AR] = `Sample`$(\mathbf{M}^{(0)}(c), M_s, \sigma_e^{(0)}(c), \sigma_w^{(0)}(c), \lambda^{(0)}(c))$ ▷ Alg. 2
    if $AR < 0.3$ **then**
        $\lambda^{(0)}(c) = l_\lambda \lambda^{(0)}(c)$
    end if
    if $AR > 0.5$ **then**
        $\lambda^{(0)}(c) = u_\lambda \lambda^{(0)}(c)$
    end if
end while

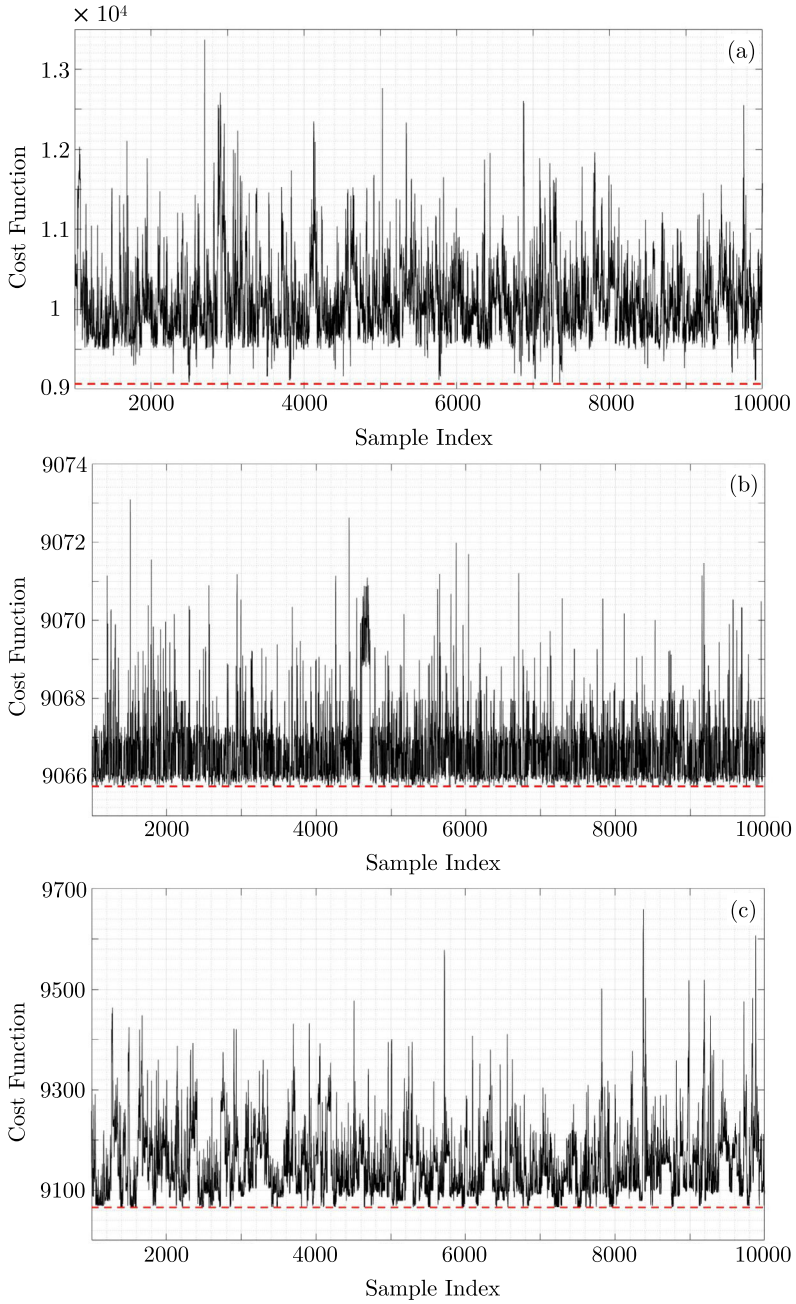


Fig. 3 Effect of λ on the sampling algorithm. Plotted are chains of the cost function for (a) $\lambda = 500$, (b) $\lambda = 1$, and (c) $\lambda = 15$. The dashed line corresponds to the minimum encountered value of the cost function

In preliminary experiments (not shown), we briefly analyzed the impact of the standard deviation of the relative weights, σ_w, on the behavior of the sampling algorithm. The results suggest that σ_w does not have a significant impact on the behavior of the algorithm, so long as a reasonable value, in the range [0.1, 0.9], is selected. Consequently, σ_w was not adapted by the sampling algorithm.

7.2.3 Input and initialization

The input to Algorithm 1 includes the observation function, $Y(\vec{x}, t_O)$, the prior, $\mathcal{E}^\star$, the stochastic velocity map Eq. 2, and $N(\mathcal{S})$, defined as the number of events contributing to the observations. Also provided as input are (i)

 https://doi.org/10.1007/s10596-025-10350-0

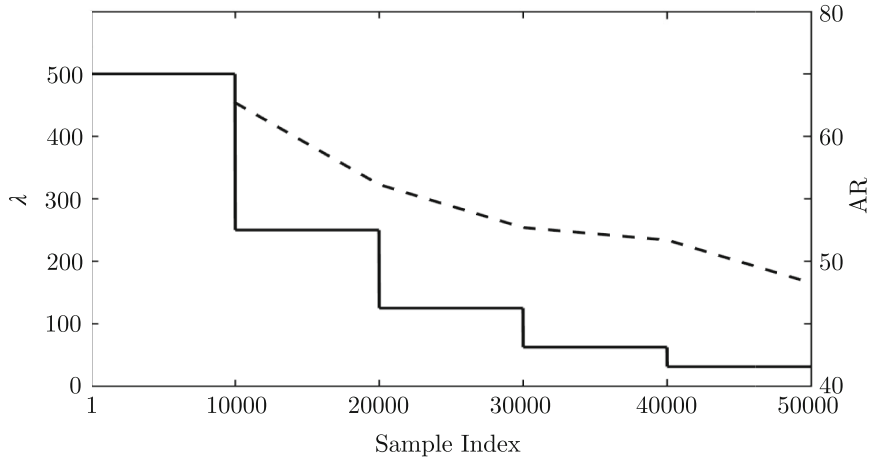


Fig. 4 The solid line is a plot of the values of λ adapted every $M_s = 10000$ samples. The dashed line shows how the % acceptance ratio approaches the desired range as λ is adapted. Adaptation is performed using Algorithm 1 and the curves are generated using results of Experiment 3 from Table 1

the number of parallel chains N_c, (ii) the initial guess the vector of the model parameters $\mathbf{M}^{(0)}$, randomly sampled from the uniform distributions over the corresponding ranges, (iii) the initial value $\lambda^{(0)}$ of the hyperparameter λ and its adaptation parameter M_s and factors u_λ and l_λ, (iv) the initial value $\sigma_e^{(0)}$ of the standard deviation σ_e and its adaptation parameters M_u and f_σ, and (v) the standard deviation σ_w.

8 Experimental setup

We explore the performance of the proposed algorithm in inferring the release events based on observations arising form the single and multiple sources described in Experiments 1-4 in Table 1. Probable locations of passive particles originating from these sources, and carried by the ensemble velocity field described in Section 2 using the transport model of Section 3, are determined using the LPT algorithm described in Section 5. Table 1 also provides the sampling algorithm parameters for all cases considered.

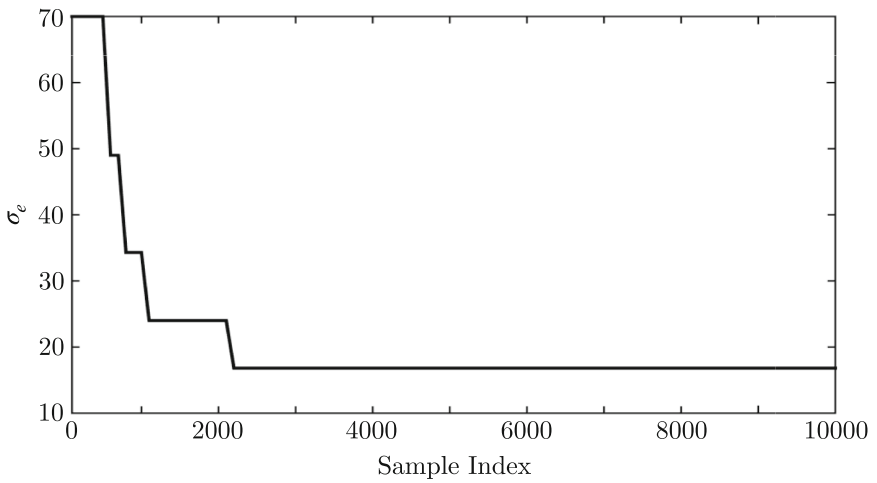


Fig. 5 Plot of the values of σ_e adapted every $M_u = 100$ samples. Adaptation is performed using Algorithm 2 and the curves are generated using results of Experiment 3 from Table 1

Algorithm 2 Pseudocode of the sampling subroutine.

$[\mathbf{M}^{(M_s)}, AR]$ = `Sample`$(\mathbf{M}^{(in)}, M_s, \sigma_e, \sigma_w, \lambda)$
$q = 0$
$\mathbf{M}^{(a)} = \mathbf{M}^{(in)}$
Set f_σ (< 1) and M_u
for i=1 to M_s **do**
  for $d = 1$ to $2N_s$ **do**
    $\mathbf{M}^{(s)} = \mathbf{M}^{(a)}$
    if $1 \leq d \leq N_s$ **then**
      Sample $\mathbf{M}^{(s)}(d) \sim \mathcal{N}(\mathbf{M}^{(a)}(d), \sigma_e)$
    end if
    if $N_s \leq d \leq 2N_s$ **then**
      Sample $\mathbf{M}^{(s)}(d) \sim \mathcal{N}(\mathbf{M}^{(a)}(d), \sigma_w)$
    end if
    Sample again or Reflect when out of range
    Calculate $\mathbb{J}^{(\mathbf{M}^{(s)})}$ and $\mathbb{J}^{(\mathbf{M}^{(a)})}$ ▷ Eq. 6
    Sample $\alpha \sim \mathcal{U}(0, 1)$
    if $\frac{e^{-\frac{\mathbb{J}^{(\mathbf{M}^{(s)})}}{\lambda}}}{e^{-\frac{\mathbb{J}^{(\mathbf{M}^{(a)})}}{\lambda}}} > \alpha$ **then**
      $q \Leftarrow q + 1$ ▷ increment the number of accepted samples
      $\mathbf{M}^{(a)} = \mathbf{M}^{(s)}$
    end if
  end for
  if mod $(i, M_u) == 0$ **then**
    Calculate $s_0^{(l)}$ for $l \in \mathcal{S}$ ▷ autocovariance at lag 0
    if any $s_0^{(l)}$ is zero **then**
      ▷ update the standard deviation of the event index proposal distribution
      $\sigma_e \Leftarrow f_\sigma \sigma_e$
    end if
  end if
  $AR = \frac{q}{2N_s M_s}$
end for

The trajectory of the probable moving sources is described next, together with its representation as a set of discrete events, which leads to the specification of the discrete prior. This is followed by a description of the synthetic observation patches, used for validation and testing.

8.1 Trajectory

A single trajectory is considered in the experiments. The ship path, selected based on the criterion stated in Section 4 and shown in Fig. 1b, is composed of 5 stages. The ship leaves the Suez Canal on July 19, 2006, taken to be respectively the reference position and reference time. The ship travels for 7 days in the MS and reaches the Strait of Gibraltar on July 26, 2006, i.e. 17 days prior to the observation time, which is August 12, 2006. The trajectory is divided into $N_s = 1037$ segments of equal length, Δs, chosen to be equal $4km$, which is of the same order as the longitude-latitude grid size; the scale at which the velocity field is resolved. The number of events in the prior $\mathcal{E}^\star$ is then $N_E = N_s = 1037$. The locations along the trajectory of release events $(\vec{x}, t)_m$, $m = 0, \ldots, N_E - 1$ belonging to $\mathcal{E}^\star$ are the midpoints of the segments m along the trajectory, so that $\vec{x}_m = \vec{x}_{m-1} + \frac{\Delta s}{2}(\hat{s}_m + \hat{s}_{m-1})$, $m > 0$ and $\vec{x}_0 = \frac{\Delta s}{2}\hat{s}_0$, where $\hat{s}_m$ is the unit vector along trajectory segment m. The release time is $t_m = \left(m + \frac{1}{2}\right)\Delta t$, where Δt is time separating two consecutive events on the trajectory. As such, Δt is equal to the time it takes a ship to traverse a distance Δs. The ship speed is set to $V_s = 24$ km/hr, which is within the range of typical oil tanker speeds (12-15 knots), so that $\Delta t = 10$ minutes.

Because a single trajectory is considered, the location and time of a release event are one to one, given that the velocity is known, which is typically the case when AIS data is available. The model parameters are therefore the locations (or times) of the release events and their corresponding relative spill contributions.

8.2 Observations

For the considered experiments listed in Table 1, we use synthetic observation patches that originated from one or more sources, where each source comprises a set of consecutive release events. The observation patches are constructed in one of the following two ways:

- *Using the probability maps* (OP1):
  To validate the implementation of the proposed MCMC algorithm, Experiments 1 and 2, listed in Table 1, employ observations constructed using the probability maps obtained from the forward LPT of pre-specified release events in the ensemble flow field. If these maps intersect, an observation patch can be synthesized in the intersection zone, which allows us to assess the performance of the algorithm in inferring probable release events from different sources that contribute to a single observation patch. Experiments 1 and 2 of Table 1 are designed to validate the proposed algorithm in a double source inference problem. For each experiment, the $Y = 1$ observations belong to the intersection zone of the two probability maps originating from two distinct sources, where each source comprises a single release event, as shown in Fig. 6 for experiment 2.
- *Deterministic simulation using the mean of the stochastic velocity field* (OP2):
  In this case, the observation patch is constructed from the scalar distribution generated by the forward advection-diffusion problem of particles emitted in pre-specified release events. The particles are advected by the mean of the stochastic velocity field and diffusion is modeled using the core spreading method [52, 53]. Observation patches generated in this manner are used in Experiments 3 and 4, listed in Table 1, to explore the performance of the proposed algorithm in realistic spill events from single and multiple sources where the advection flow field is uncertain. In Experiment 3, the algorithm was applied to the single source inference problem, where the observation patch, shown in Fig. 9-a, was generated for release events 349 to 353. Experiment 4 is tailored to assess the robustness of the Algorithm in the inference of four sources contributing to four separate observation patches generated using the OP2 method. The four sources comprise respectively the following sequences of release events: 200 to 203, 349 to 353, 668 to 671, and 799 to 802. Note that the $Y = 0$ observations, shown in Fig. 12, are within a $B = 0$ outside the patches. As mentioned earlier, the observation is made on August 12, 2006, i.e. the observation time $t_O = 29$ days.

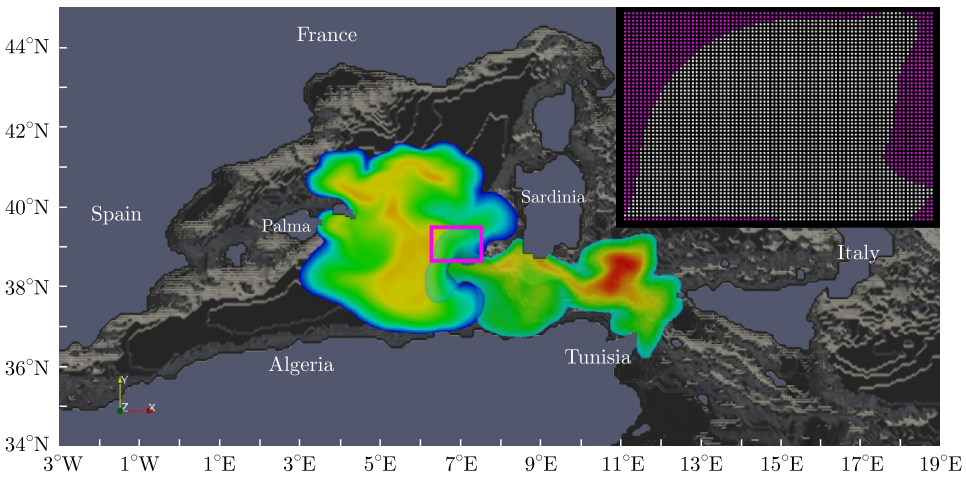


Fig. 6 The observation patch, with $B = 0$, due to events 300 and 400 in Experiment 1, generated using the OP1 method. The patch, bound by the shown rectangle, is a subset of the intersection between the forward probability maps of the two events

9 Results and discussion

This section analyzes the behavior of the proposed MCMC and sampling algorithms (Algorithms 1 and 2) and assesses their performance based on the experiments listed in Table 1. As stated earlier, Experiment 1 and 2 were constructed to validate the proposed inference algorithm and its implementation, while Experiments 3 and 4 showcase the performance of the proposed algorithm in a realistic setting. For each case, we use the Algorithm 1 to determine the posterior distribution of the events indices and their relative weights. In all cases, we see that the algorithm quickly reaches stationary distributions with well-mixed chains. To verify the quality of the inference, we compare the most probable model parameters inferred by the MCMC algorithm with those obtained by solving the optimization problem of minimizing the cost function in Eq. 6. The "Global Search" optimization solver in MATLAB is used for this purpose. The most probable event indices predicted by the MCMC and the optimal indices obtained from global search algorithm are respectively presented in the last two rows ($\bar{r} \pm 1$ std. and r_{gs}) of Table 1, which also provides the selected parameters for the sampling algorithm.

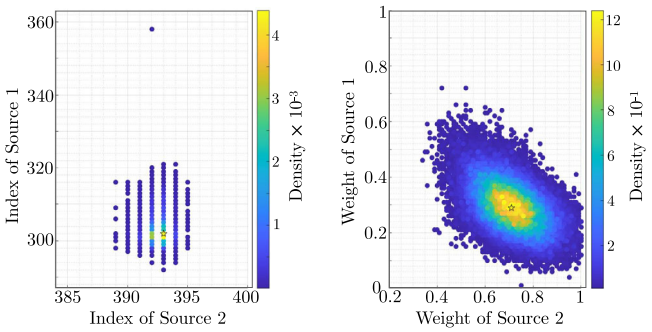
9.1 Validation of the algorithm

For Experiment 1 of Table 1, the observation patch generated using OP1 from events 300 and 400 is shown in Fig. 6. The patch is arbitrarily selected to lie within the intersection region of the forward probability maps of the two events, and bound as illustrated. Figure 7a shows the joint densities of the two events indices (left) and their weights (right). Samples from the MCMC chains are used for the purpose of constructing these densities. Also note that for clarity and consistency of the presentation, we have adopted the convention that source 1 is associated with highest indexed events, followed by source 2, and so on as applicable. For source 1, the marginal posteriors of the event index and weight are plotted in Fig. 7b-left and 7b-right, respectively. Similar plots for source 2 are shown in Fig. 7c. According to the results presented in Fig. 7, the inferred events indices are 393 and 301 and the respective inferred weights are 0.7 and 0.28. Furthermore, the plots suggest that the solution for source 1 is more uncertain than that of source 2, as indicated by the wider distribution of the first. This might be attributed to a strong correlation between the true release events of source 1 and nearby events, which is lacking in the case of source 2. The actual events indices contributing to the observation patch are 400 and 300. Note that the posterior distributions are not perfectly Gaussian, which may be attributed to the highly nonlinear forward model, and the selection of prior distribution and likelihood functions. Figure 8 shows the forward maps of the highest posterior probable events, with the inset figure showing the forward map of the combination of the two events, P_E, in the observation domain Ω_O. The similarity with the observation patch can be inspected by comparing Fig. 8 with Fig. 6.

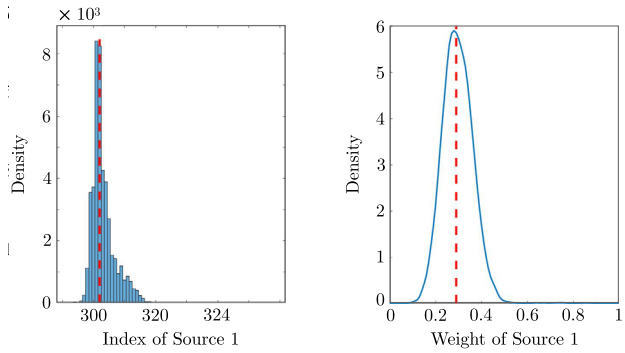
Application of our methodology to Experiment 2 led to observations that are similar to those drawn from the analysis of the computational results of Experiment 1. Consequently, we omit a detailed discussion of the results on Experiment 2, but still provide in Table 1 the inferred probable sources. Note that in this example one still observes concentrated posterior distributions, but unlike Experiment 1 a small bias towards lower indexed events is clearly seen. On the other hand, the most probable sources obtained by the Bayesian algorithm are still consistent with the global optimization results. This suggests that the bias observed in both approaches is likely due to the partial observation of a patch generated by different sources.

For completeness, we mention the computational costs required by the present machinery. The numerical forward model was developed in Fortran 90 and is fully parallelized using openMP. The codes were run on a standard office workstation, with an Intel Xeon E5-2680 v4 2.4 GHz Fourteen-Core (28 threads) LGA 2011-3 Processor with 64 Gb RAM. Each forward run takes around 4 hours, and additional details about the computational cost of the LPT algorithm can be found in [21]. The Bayesian inference codes were written in Python, and this validation experiment inverting for a single source using 10000 samples takes around 30 minutes per chain.

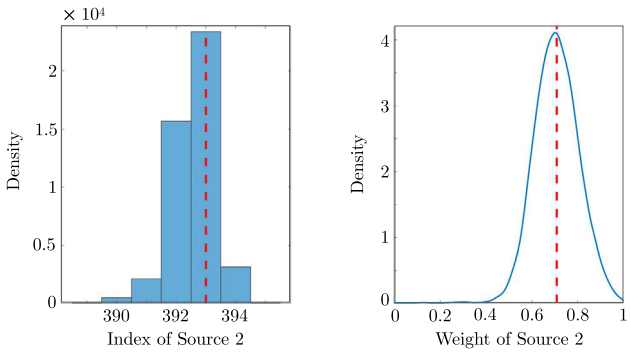
 https://doi.org/10.1007/s10596-025-10350-0



(a) Joint distributions of the event indices (left) and weights (right) of sources 1 (x-axis) and 2 (y-axis).



(b) Marginal posterior distributions of the event index (left) and weight (right) for source 1 in Experiment 1.



(c) Marginal posterior distributions of the event index (left) and weight (right) for source 2 in Experiment 1.

Fig. 7 Posterior distribution of the source parameters in Experiment 1. Indices of the inferred events are marked by the vertical dashed lines

9.2 Inference of single and multiple sources in a realistic setting

For Experiment 3 of Table 1, Fig. 9a presents the observation patch generated using OP2 from events 349 – 353. To investigate the impact of the box extension on the cost function, Fig. 9b shows the dependence of the cost function on the event index for different values of B. For each event, the LPT algorithm is employed to generate the forward probability map, which, along with the observation patch, is used to compute the cost function according to Eq. 6, It can be observed that the event index that minimizes the cost function is largely insensitive to the value of B over the range 0 - 5 W_O, where W_O is the width of the rectangle tightly fitting the observation patch, shown in Fig. 9a.

Figure 10a shows that the chain resulting from the MCMC algorithm, using the parameters listed in Table 1, is well mixed and convergent, as a result of adaptation of λ and σ_e. Figure 10b plots the marginal posterior probability

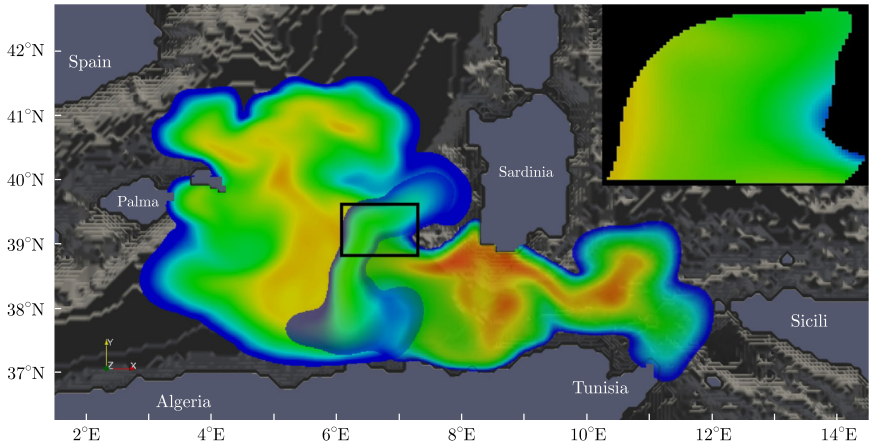


Fig. 8 Forward probability maps of the combination of events with maximum posterior probability, for Experiment 1

density of the event indices. The histogram shows that the event index with maximum posterior probability density (MAP) is 349, which matches that predicted using the global optimization algorithm (represented using a red dashed line). Events indices within one standard deviation away from the mean lie in the range [345, 354] (shown in Table 1 for Experiment 1), indicating that the probability of predicting the event index to be within this range is around

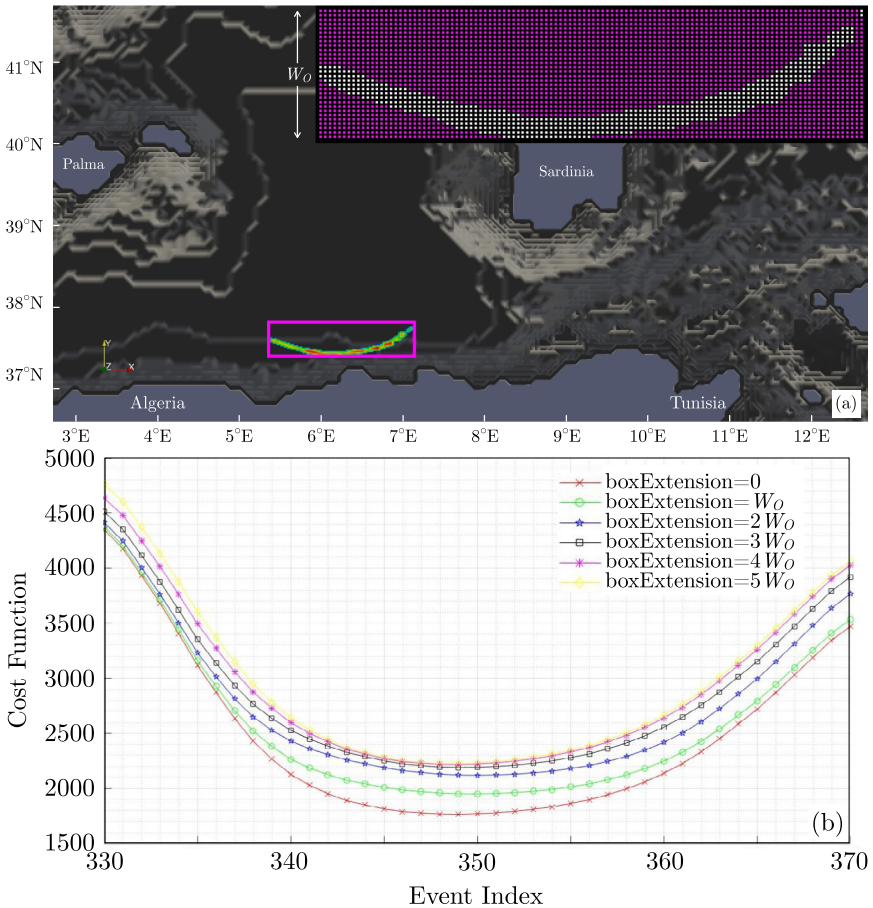


Fig. 9 (a) The observation patch, with $B = 0$, due to events $349 - 353$ in Experiment 3, generated using the OP2 method. (b) Cost function versus event index. Curves are generated for different values of B as indicated

 https://doi.org/10.1007/s10596-025-10350-0

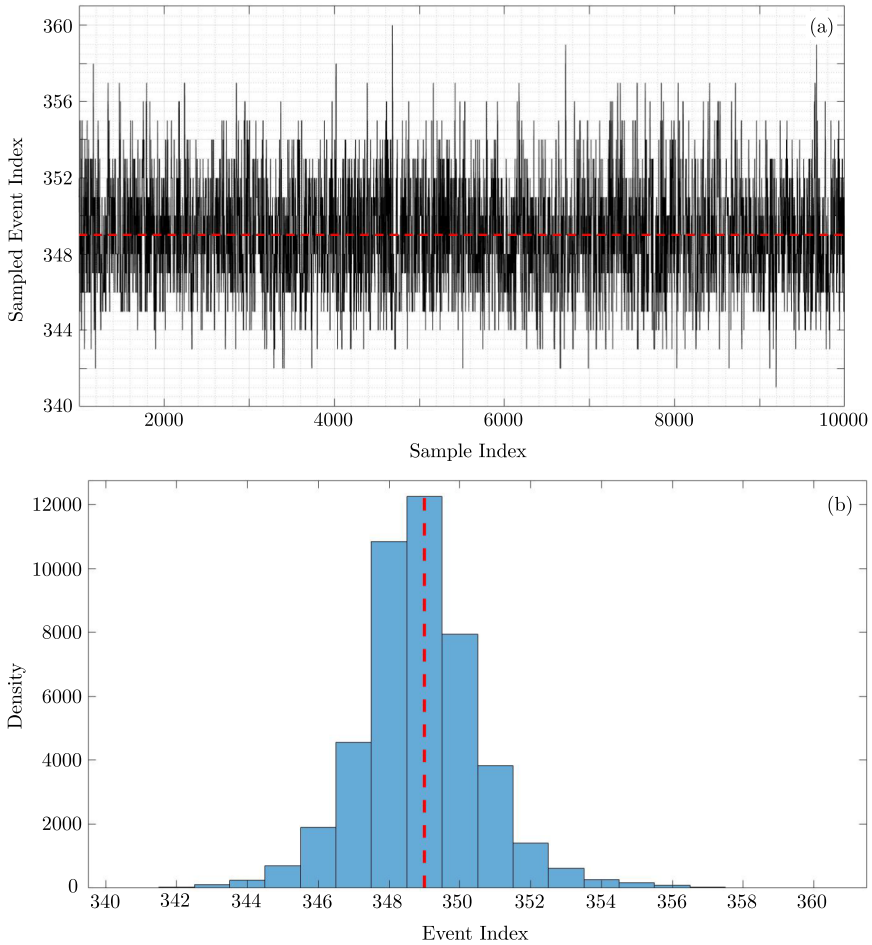


Fig. 10 Event index chain (a) and marginal posterior event index distribution (b) for Experiment 3. The observation patch is generated from release events 349-353. The index of the inferred event is marked by the dashed line

68%. Note that the predicted solution with highest posterior probability matches the true release event and that most of the remaining predictions sampled by the algorithm fall within ±7 events.

Figure 11 overlays the observation patch on top of the forward map of the event index with highest posterior probability. The figure shows that the observation patch lies inside the high probability region of the pushed forward MAP. This provides further confidence in the quality of the inferred solution.

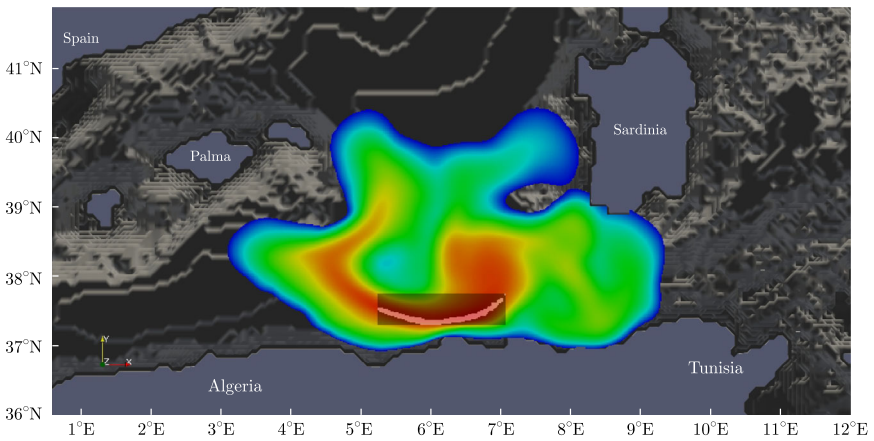


Fig. 11 Observation patch (Fig. 9a) overlaid on the forward probability map of the event of maximum posterior probability, for Experiment 3

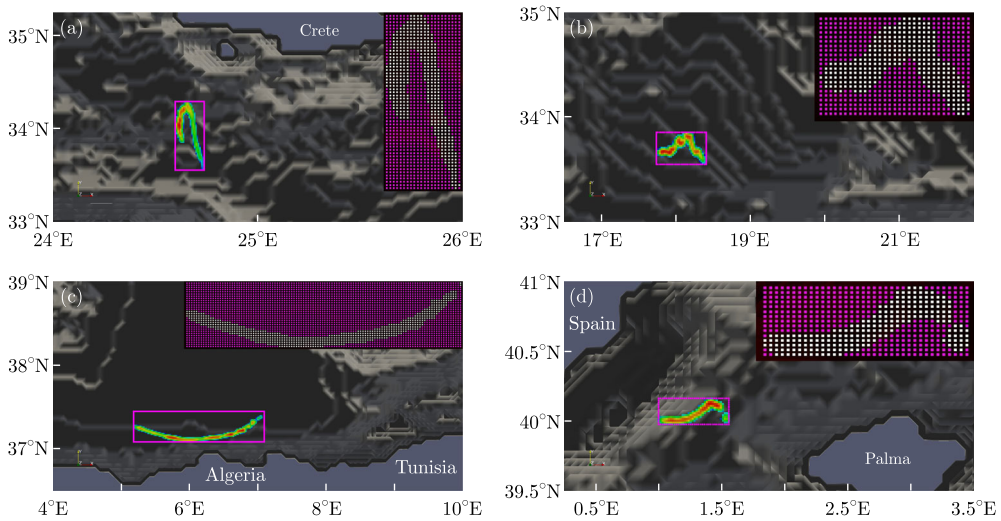
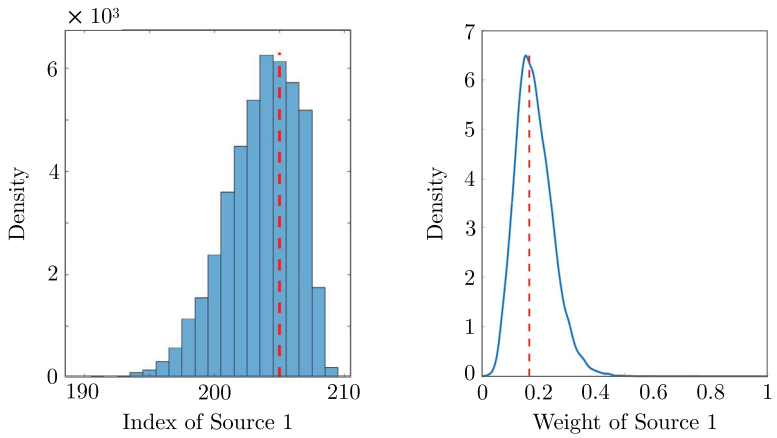
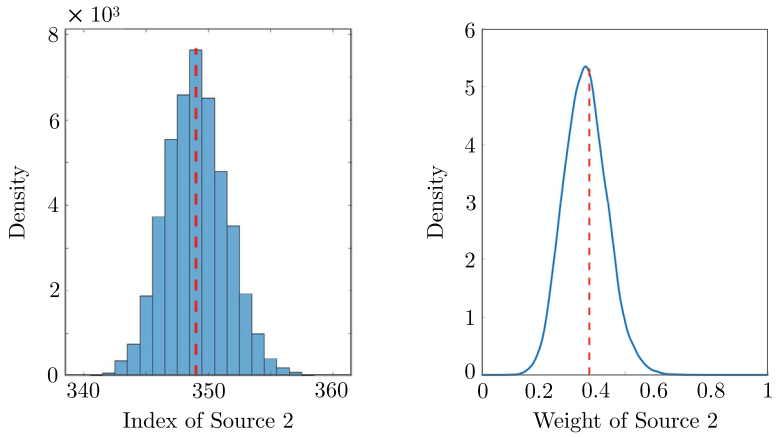


Fig. 12 The observation domain in Experiment 4 contains four separate observation patches, generated using the OP2 method with $B = 0$, due to event sequences $200 - 203$ (a), $349 - 353$ (b), $668 - 671$ (c), and $799 - 802$ (d), respectively



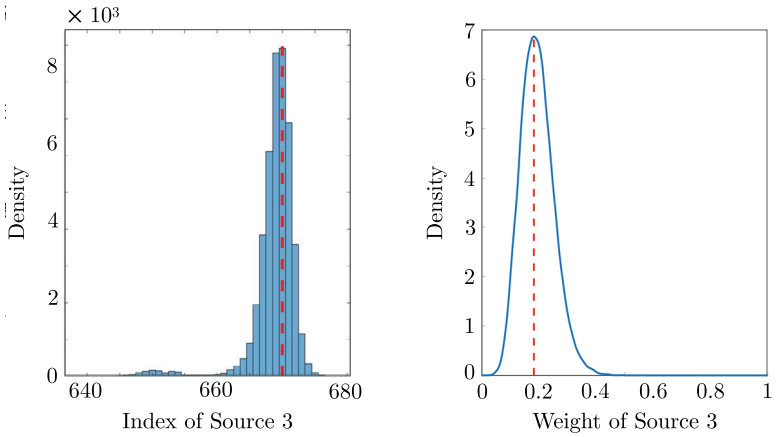
(a) Marginal posterior event index (left) and weight (right) distributions for source 1.



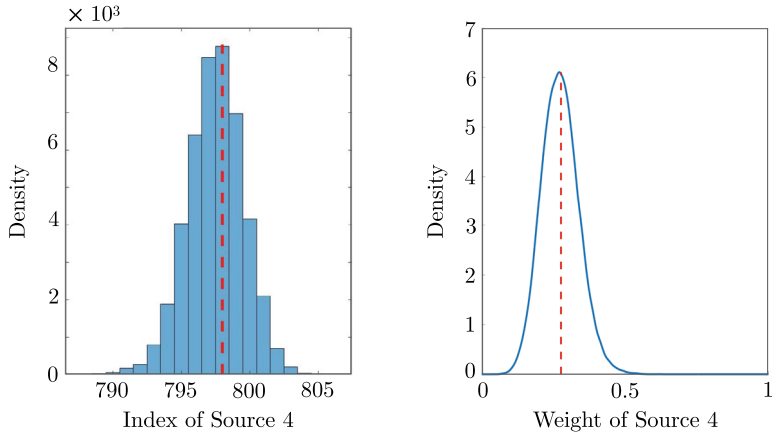
(b) Marginal posterior event index (left) and weight (right) distributions for source 2.

Fig. 13 Posterior distributions of the source parameters in Experiment 4. Indices of the inferred events are marked by the vertical dashed lines

 https://doi.org/10.1007/s10596-025-10350-0



(c) Marginal posterior event index (left) and weight (right) distributions for source 3.



(d) Marginal posterior event index (left) and weight (right) distributions for source 4.

Fig. 13 continued

In Experiment 4, we employ the algorithm to infer 4 events contributing to 4 separate observation patches, shown in Fig. 12. The number of sources to be inferred is set to be equal to the number of observations patches (i.e. 4), and, as such, is not a result of the inference procedure. Note, however, that if a smaller number is preset, the inference procedure will converge to a subset of the contributing sources. If, on the other hand, a larger number is selected, the four contributing sources will be inferred with positive weights, and the algorithm will assign negligible weights to the remaining sources. The computational cost of the algorithm increases with the number of number of sources to be inferred. Thus, one would generally aim to limit the number of sources to be inferred to be at most slightly larger than the number of observation patches.

Also note that in Experiment 4, the patches, generated using the OP2 method, originate from the sources consisting of the following sequences of event indices: 200-203, 349-353, 668-671, and 799-802. The corresponding release times (t_m) are shown in Table 1. The posterior probability density functions of the events indices and weights are plotted in Fig. 13a-d. The events with maximum posterior probability are respectively 205, 349, 664, and 798. The corresponding ranges of indices within 1 standard deviation from the mean are 201-206, 347-352, 665-672, and 795-800. Note that this experiment highlights the potential of Bayesian inversion for STE, where by fine tuning the Bayesian inversion machinery, the more complicated problem of inverting for 4 different release events may be solved with reasonable accuracy, such that the predictions are within approximately 10 release events from the reference solution. Figure 14 shows the observation patches of Fig. 12 overlaid on the forward maps of the events with indices and weights of highest posterior probability. The figure shows that each observation patch is, spatially, a subset of the high probability region of corresponding the forward map. This illustrates the

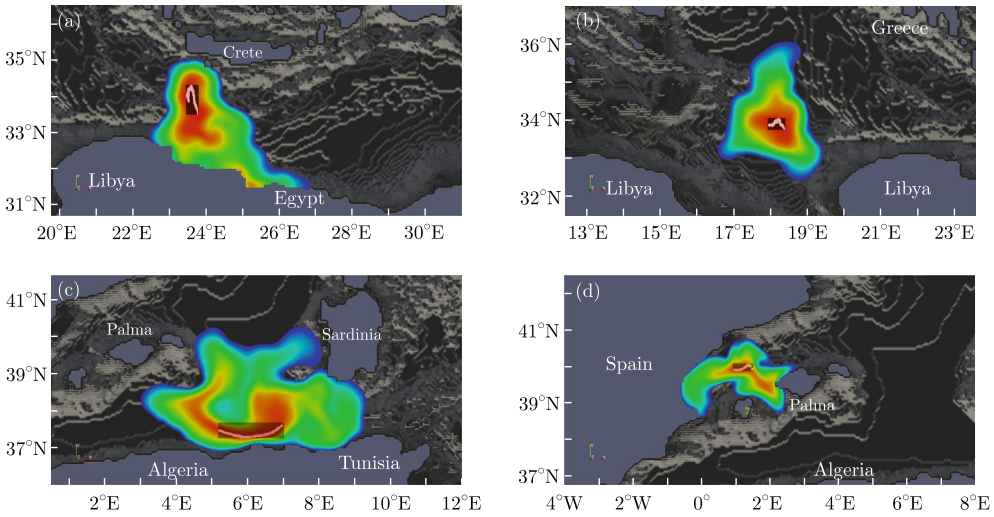


Fig. 14 Observation patches of Fig. 12 overlaid on the corresponding forward probability maps of the events of maximum posterior probability, for Experiment 4

ability of the proposed inference machinery to identify the sources of multiple patches that are well separated, and that originated by spills occurring a different times prior to the observation.

10 Conclusions

We proposed an efficient Bayesian inference framework for the identification of the release locations, release times, and relative weights of moving sources of passive tracers contributing to single and multiple observation patches in a stochastic flow field. The framework is based on an adaptive MCMC sampling algorithm where the prior consists of a finite set of possible release events discretizing preset trajectories. The likelihood is a function of the weighted sum of the probability maps generated by the Lagrangian Particle Tracking, in a stochastic flow field, of passive tracers emitted by the sampled events. To alleviate the cost of computing the forward probability map associated with the sampled events, we adopted a Green's function approach whereby the forward probability maps of all the events belonging to the prior are generated individually, once and for all, in a preprocessing step and stored. The maps of the sampled events are then queried and used by the MCMC algorithm. The latter evaluates proposals based on a likelihood that employs logistic regression cost function that quantifies the distance between the observation and model output. Computing this distance is made possible by the proposed logistic (or binary) representation of the observation patches. The likelihood, expressed through the cost function, involves a hyper parameter λ that is adaptively tuned to ensure an acceptance ratio in the range 30-50 %. The width of the proposal distribution is also adaptively updated by the MCMC algorithm. This yields well-mixed chains and helps ensure that stationary distributions are efficiently reached.

Performance of the proposed algorithm is assessed for various scenarios where single and separate observation patches originate from a single or a multiple sources, with each source consisting of a sequence of release events belonging to a prescribed trajectory. The trajectory, extending from the Suez Canal till the Strait of Gibraltar in the MS, is assumed to be traversed by vessels moving at a constant speed. For each experiment, the observation patches are synthesized from a prescribed set of release sequences. For all experiments considered, the proposed method succeeded in inferring the model parameters (location and relative contribution) and quantifying the associated uncertainty. In particular, the inferred events indices are very close to the indices of events sources used to synthesize the observation patches. Additional confidence in the performance of the proposed algorithm was gained by comparing the most probable events inferred with those predicted by a global search optimization algorithm.

Future work will focus on extending the scope of the experimental setting, namely through the inclusion of sources associated with multiple trajectories, and tackling the associated challenges. To improve efficiency and expand applicability, we plan to explore alternative formulations of the inference algorithm based on sequential inference of multiple sources, as well as the inclusion of multiple observations. We consequently anticipate that the proposed framework would provide effective means for fast and accurate identification of multiple sources of pollutants, and thus constitute a valuable resource in rapid response scenarios.

Funding This work is supported in part by the University Research Board of the American University of Beirut, and by King Abdullah University of Science and Technology (KAUST) under Award No. REP/1/3268-01-01, and Award No. OSR-CRG2018-3711.

Data Availability Data will be made available upon reasonable request.

Code Availibility The code is being prepared to be available publicly via GitHub.

Statements and Declarations

Conflict of Interests/Competing Interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Mittal, H.V.R., Hammoud, M.A.E.R., Carrasco, A.K., Hoteit, I., Knio, O.M.: Oil spill risk analysis for the neom shoreline. Sci. Reports **14**(1) (2024). https://doi.org/10.1038/s41598-024-57048-4
2. Jambeck, J.R., Geyer, R., Wilcox, C., Siegler, T.R., Perryman, M., Andrady, A., Narayan, R., Law, K.L.: Plastic waste inputs from land into the ocean. Science **347**(6223), 768–771 (2015). https://doi.org/10.1126/science.1260352, https://arxiv.org/abs/www.science.org/doi/pdf/10.1126/science.1260352
3. Tsiaras, K., Hatzonikolakis, Y., Kalaroni, S., Pollani, A., Triantafyllou, G.: Modeling the pathways and accumulation patterns of micro- and macro-plastics in the mediterranean. Front. Marine Sci. **8** (2021). https://doi.org/10.3389/fmars.2021.743117
4. Gkanasos, A., Tsiaras, K., Triantaphyllidis, G., Panagopoulos, A., Pantazakos, G., Owens, T., Karametsis, C., Pollani, A., Nikoli, E., Katsafados, N., Triantafyllou, G.: Stopping macroplastic and microplastic pollution at source by installing novel technologies in river estuaries and waste water treatment plants: The claim project. Front. Mar. Sci. **8** (2021). https://doi.org/10.3389/fmars.2021.738876
5. GESAMP: Estimates of oil entering the marine environment from sea-based activities. UK. (Joint group of experts on the Scientific Aspects of Marine Environmental Protection), Reports and studies GESAMP, Vol. 75, London (2019)
6. Kostianoy, A.G., Carpenter, A.: History, sources and volumes of oil pollution in the mediterranean sea. In: Carpenter, A., Kostianoy, A.G. (eds.) Oil Pollution in the Mediterranean Sea: Part I: The International Context, pp. 9–31. Springer, Cham (2018). https://doi.org/10.1007/698_2018_369
7. Girin, M., Carpenter, A.: Shipping and oil transportation in the mediterranean sea. In: Carpenter, A., Kostianoy, A.G. (eds.) Oil Pollution in the Mediterranean Sea: Part I: The International Context, pp. 33–51. Springer, Cham (2018). https://doi.org/10.1007/698_2017_6
8. Lacombe, H., Gascard, J.C., Gonella, J., Bethoux, J.P.: Response of the mediterranean to the water and energy fluxes across its surface, on seasonal and interannual scales. Oceanol. Acta **4**(2), 247–255 (1981)
9. Cozar, A., Sanz-Martin, M., Marti, E., Gonzalez-Gordillo, J.I., Ubeda, B., Galvez, J.A., Irigoien, X., Duarte0, C.M.: Plastic accumulation in the mediterranean sea. PLOS ONE **10**(4), 1–12 (2015). https://doi.org/10.1371/journal.pone.0121762
10. Lebreton, L.C.-M., Greer, S.D., Borrero, J.C.: Numerical modelling of floating debris in the world's oceans. Mar. Pollut. Bullet. **64**(3), 653–661 (2012). https://doi.org/10.1016/j.marpolbul.2011.10.027
11. Sebille, E., Wilcox, C., Lebreton, L., Maximenko, N., Hardesty, B.D., Franeker, J.A., Eriksen, M., Siegel, D., Galgani, F., Law, K.L.: A global inventory of small floating plastic debris. Environ. Res. Lett. **10**(12), 124006 (2015). https://doi.org/10.1088/1748-9326/10/12/124006
12. Kostianoy, A.G., Carpenter, A.: Oil and gas exploration and production in the mediterranean sea. In: Carpenter, A., Kostianoy, A.G. (eds.) Oil Pollution in the Mediterranean Sea: Part I: The International Context, pp. 53–77. Springer, Cham (2018). https://doi.org/10.1007/698_2018_373

13. Ouyang, W., Hao, F.-H., Wang, X.-l.: Regional non point source organic pollution modeling and critical area identification for watershed best environmental management. Water, Air, Soil Pollut. **187**(1), 251–261 (2008). https://doi.org/10.1007/s11270-007-9513-y
14. Jiang, X., Ma, R., Wang, Y., Gu, W., Lu, W., Na, J.: Two-stage surrogate model-assisted bayesian framework for groundwater contaminant source identification. J. Hydrol. **594**, 125955 (2021)
15. Lane, R., Briers, M., Copsey, K.: Approximate bayesian computation for source term estimation. Math. Defence **2009** (2009)
16. Wawrzynczak, A., Kopka, P.: Approximate bayesian computation for estimating parameters of data-consistent forbush decrease model. Entropy **20**(8) (2018). https://doi.org/10.3390/e20080622
17. Zheng, X., Chen, Z.: Inverse calculation approaches for source determination in hazardous chemical releases. J. Loss Prevention Process Industries **24**(4), 293–301 (2011)
18. Zhou, X., Amaral, V., Albertson, J.D.: Source characterization of airborne emissions using a sensor network: Examining the impact of sensor quality, quantity, and wind climatology. In: 2017 IEEE International Conference on Big Data (Big Data), pp. 4621–4629 (2017). IEEE
19. Humphries, R., Jenkins, C., Leuning, R., Zegelin, S., Griffith, D., Caldow, C., Berko, H., Feitz, A.: Atmospheric tomography: A bayesian inversion technique for determining the rate and location of fugitive emissions. Environ. Sci. Technol. **46**(3), 1739–1746 (2012)
20. Chhadé, H., Abdallah, F., Mougharbel, I., Gning, A., Julier, S., Mihaylova, L.: Localisation of an unknown number of land mines using a network of vapour detectors. Sensors **14**(11), 21000–21022 (2014)
21. Mohtar, S.E., Hoteit, I., Knio, O., Issa, L., Lakkis, I.: Lagrangian tracking in stochastic fields with application to an ensemble of velocity fields in the red sea. Ocean Model. **131**, 1–14 (2018). https://doi.org/10.1016/j.ocemod.2018.08.008
22. Zodiatis, G., Lardner, R., Solovyov, D., Panayidou, X., De Dominicis, M., et al.: Predictions for oil slicks detected from satellite images using myocean forecasting data. Ocean Sci. (OS) (2012)
23. Hammoud, M.A.E.R., Lakkis, I., Knio, O., Hoteit, I.: Moving source identification in an uncertain marine flow: Mediterranean sea application. Ocean Eng. **220**, 108435 (2021)
24. Keats, A., Yee, E., Lien, F.-S.: Efficiently characterizing the origin and decay rate of a nonconservative scalar using probability theory. Ecol Modell **205**(3–4), 437–452 (2007)
25. Yee, E., Lien, F.-S., Keats, A., D'Amours, R.: Bayesian inversion of concentration data: Source reconstruction in the adjoint representation of atmospheric diffusion. J. Wind Eng. Indust. Aerodyn. **96**(10–11), 1805–1816 (2008)
26. Yee, E.: Bayesian probabilistic approach for inverse source determination from limited and noisy chemical or biological sensor concentration measurements. In: Chemical and Biological Sensing VIII, vol. 6554, pp. 65540 (2007). International Society for Optics and Photonics
27. Yee, E., Flesch, T.K.: Inference of emission rates from multiple sources using bayesian probability theory. J. Environ. Monitoring **12**(3), 622–634 (2010)
28. Xue, F., Kikumoto, H., Li, X., Ooka, R.: Bayesian source term estimation of atmospheric releases in urban areas using les approach. J. Hazardous Mater. **349**, 68–78 (2018)
29. Yee, E., Hoffman, I., Ungar, K.: Bayesian inference for source reconstruction: A real-world application. Int. Scholarly Res. Notices **2014** (2014)
30. Kopka, P., Wawrzynczak, A.: In search of an effective monte carlo method for identification of atmospheric contamination source. In: Journal of Physics: Conference Series, vol. 1391, pp. 012106 (2019). IOP Publishing
31. Kopka, P., Wawrzynczak, A., Borysiewicz, M.: Application of the approximate bayesian computation methods in the stochastic estimation of atmospheric contamination parameters for mobile sources. Atmos. Environ. Atmospheric **145**, 201–212 (2016)
32. Chen, N., Lunasin, E., Wiggins, S.: Lagrangian descriptors with uncertainty. Phys. D: Nonlinear Phenomena **467**, 134282 (2024). https://doi.org/10.1016/j.physd.2024.134282
33. García-Sánchez, G., Mancho, A.M., Ramos, A.G., Coca, J., Wiggins, S.: Structured pathways in the turbulence organizing recent oil spill events in the eastern mediterranean. Sci. Reports **12**(1) (2022). https://doi.org/10.1038/s41598-022-07350-w
34. García-Sánchez, G., Mancho, A.M., Wiggins, S.: A bridge between invariant dynamical structures and uncertainty quantification. Commun. Nonlinear Sci. Numer. Simul. **104**, 106016 (2022). https://doi.org/10.1016/j.cnsns.2021.106016
35. Groves, D.G., Yates, D., Tebaldi, C.: Developing and applying uncertain global climate change projections for regional water management planning. Water Resour. Res. **44**(12) (2008)
36. Bennett, K.E., Werner, A.T., Schnorbus, M.: Uncertainties in hydrologic and climate change impact analyses in headwater basins of british columbia. J. Climate **25**(17), 5711–5730 (2012)
37. Edwards, C.A., Moore, A.M., Hoteit, I., Cornuelle, B.D.: Regional ocean data assimilation. Ann. Rev. Mar. Sci. **7**, 21–42 (2015)
38. Hoteit, I., Luo, X., Bocquet, M., Kohl, A., Ait-El-Fquih, B.: Data assimilation in oceanography: Current status and new directions. New Front. Oper. Oceanography 465–512 (2018)

39. Beaudoin, A., Dreuzy, J.-R., Erhel, J.: An efficient parallel particle tracker for advection-diffusion simulations in heterogeneous porous media. In: Euro-Par 2007 Parallel Processing: 13th International Euro-Par Conference, Rennes, France, August 28-31, 2007. Proceedings 13, pp. 717–726 (2007). Springer
40. Guo, H., Yuan, X., Huang, J., Zhu, X.: Coupled ensemble flow line advection and analysis. IEEE Trans. Vis. Comput. Graph. **19**(12), 2733–2742 (2013)
41. MarineTraffic: What exactly is the Density Map layer? (2023). https://support.marinetraffic.com/en/articles/9552882-what-exactly-is-the-density-map-layer. Accessed on 25 Nov 2024
42. Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis, 3rd edn. Chapman & Hall/CRC Texts in Statistical Science. Chapman & Hall/CRC, Philadelphia, PA (2013)
43. Gamerman, D., Lopes, H.F.: Markov Chain Monte Carlo, 2nd edn. Chapman & Hall/CRC Texts in Statistical Science. Chapman & Hall/CRC, Philadelphia, PA (2006)
44. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
45. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. Adaptive Computation and Machine Learning series. MIT Press, London, England (2016)
46. Bishop, C.M.: Pattern Recognition and Machine Learning, 1st edn. Information Science and Statistics. Springer, New York, NY (2006)
47. Yee, E.: Theory for reconstruction of an unknown number of contaminant sources using probabilistic inference. Boundary-layer Meteorol. **127**(3), 359–394 (2008)
48. Yee, E.: Source reconstruction: A statistical mechanics perspective. Int. J. Environ. Pollut. **48**(1–4), 203–213 (2012)
49. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. **21**(6), 1087–1092 (1953)
50. Hastings, W.K.: Monte carlo sampling methods using markov chains and their applications (1970)
51. Chib, S., Greenberg, E.: Understanding the metropolis-hastings algorithm. Am. Stat. **49**(4), 327–335 (1995)
52. Kuwahara, K., Takami, H.: Numerical studies of two-dimensional vortex motion by a system of point vortices. J. Phys. Soc. Japan **34**(1), 247–253 (1973)
53. Leonard, A.: Vortex methods for flow simulation. J. Comput. Phys. **37**(3), 289–335 (1980)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Issam Lakkis[1] · Alexios Rustom[1] · Mohamad Abed El Rahman Hammoud[2] · Leila Issa[3] · Omar Knio[4] · Olivier Le Maitre[5] · Ibrahim Hoteit[6]

✉ Issam Lakkis
issam.lakkis@aub.edu.lb

Alexios Rustom
abr03@mail.aub.edu

Mohamad Abed El Rahman Hammoud
mohamedabed.hammoud@kaust.edu.sa

Leila Issa
leila.issa@lau.edu.lb

Omar Knio
Omar.Knio@kaust.edu.sa

Olivier Le Maitre
olivier.le-maitre@polytechnique.edu

Ibrahim Hoteit
ibrahim.hoteit@kaust.edu.sa

1 Department of Mechanical Engineering, American University of Beirut, Beirut, Lebanon

2 Department of Mechanical Engineering, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

3 Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

4 Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 10587, Saudi Arabia

5 Centre de Mathématiques Appliquées, Ecole Polytechnique, Paris, France

6 Physical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal 10587, Saudi Arabia