Reconstructing Ocean Flow from Observed Lagrangian Trajectories

Bianca Champenois
Department of Mechanical Engineering
Center for Ocean Engineering
Massachusetts Institute of Technology
Cambridge, USA
0000-0002-3922-3055

Themistoklis P. Sapsis
Department of Mechanical Engineering
Center for Ocean Engineering
Massachusetts Institute of Technology
Cambridge, USA
0000-0003-0302-0691

***Abstract*—Accurate and computationally efficient modeling of ocean currents is important for understanding and monitoring the dispersion of marine pollution. Passive floating Lagrangian drifters, equipped with GPS, are a low-cost approach to tracking surface flow. However, assimilating drifter observations into comprehensive ocean models remains challenging, and these models are often too computationally intensive for real-time or operational use. In this work, we introduce a novel Lagrangian data assimilation method that leverages machine learning techniques to enable computationally efficient ocean flow modeling. We first train an autoencoder using existing snapshots of the flow field to create a reduced-order model, which extracts low-dimensional representations of the high-dimensional data. We then apply Bayesian optimization to minimize a cost function, defined in the low-dimensional latent space, that measures the discrepancy between observed trajectories and model-generated trajectories. Our approach is demonstrated to efficiently reconstruct regional ocean flow from sparse drifter data, with results obtained in minutes. The method can be used to identify the minimum number of drifters needed for accurate flow modeling and to determine optimal drifter launch locations.**

***Index Terms*—ocean flow; drifter trajectories; data assimilation; reduced-order modeling; machine learning; autoencoder; convolutional neural network; Bayesian optimization**

Computationally efficient modeling of ocean flow is relevant for many applications (e.g., aquaculture, search and rescue, offshore platforms, mitigating pollution dispersion, shipping). For example, knowledge of ocean flow can help track the dispersion of wastewater from the Deer Island Wastewater Treatment Plant in the Massachusetts Bay. However, accurately modeling ocean flow in real time is challenging because it is governed by complex, high-dimensional, and non-linear equations [1]. These equations are highly sensitive to initial, boundary, and forcing conditions and exhibit features across a wide range of spatial and temporal scales. Standard numerical solvers, while physically detailed, are often computationally prohibitive for real-time operational use.

Ocean flow can be observed using a variety of methods, including satellite altimetry, acoustic Doppler current profilers (ADCPs), moored instruments, and Lagrangian drifters. Among these, Lagrangian drifters offer the most cost-effective means of capturing near-surface circulation over large spatial and temporal scales. However, integrating these Lagrangian observations into models is challenging because drifter data are sparse, indirect, and influenced by chaotic advection. Ocean models are usually formulated in an Eulerian framework, complicating the assimilation of Lagrangian observations. Addressing these challenges requires advanced Lagrangian data assimilation techniques, which aim to optimally combine observations with models to improve predictions [2]. This paper presents a novel method for real-time, computationally efficient reconstruction of ocean flow from observed Lagrangian trajectories, making accurate predictions more feasible for operational applications.

A. Datasets and Application

We are motivated by a dataset of measurements of Lagrangian trajectories from *Microstar* drifters collected by the MIT Sea Grant office (Figure 2). Microstar drifters are low-cost and designed to track currents at a depth of 1 meter below the ocean's surface. The *Student Drifters* educational program, run by James Manning, offers an additional dataset of drifter trajectories dating back to 1988, including observations from both sail-equipped drifters and those with drogues (sock-like attachments that help the drifter follow deeper currents), all of which are designed, built, and deployed by students [3], [4].

B. Lagrangian Data Assimilation

Lagrangian ocean data assimilation, the process of incorporating drifter and float trajectories into numerical models to improve state estimation, has been an area of active research for decades. Broadly, data assimilation seeks to optimize the model state (or model parameters) by minimizing the mismatch between observations and model outputs. Lagrangian assimilation has demonstrated advantages over Eulerian methods in capturing transport dynamics [5].

Researchers have developed several approaches to perform Lagrangian data assimilation, including optimal interpolation (OI), variational assimilation (3D-Var or 4D-Var), and the Ensemble Kalman Filter (EnKF). OI uses a least-squares approach to minimize the difference between the model state and the observations, typically by weighting the errors based on their respective covariances [6]–[8]. Variational assimilation

This work was funded by the MIT NOAA Sea Grant office and the National Science Foundation Graduate Research Fellowship.

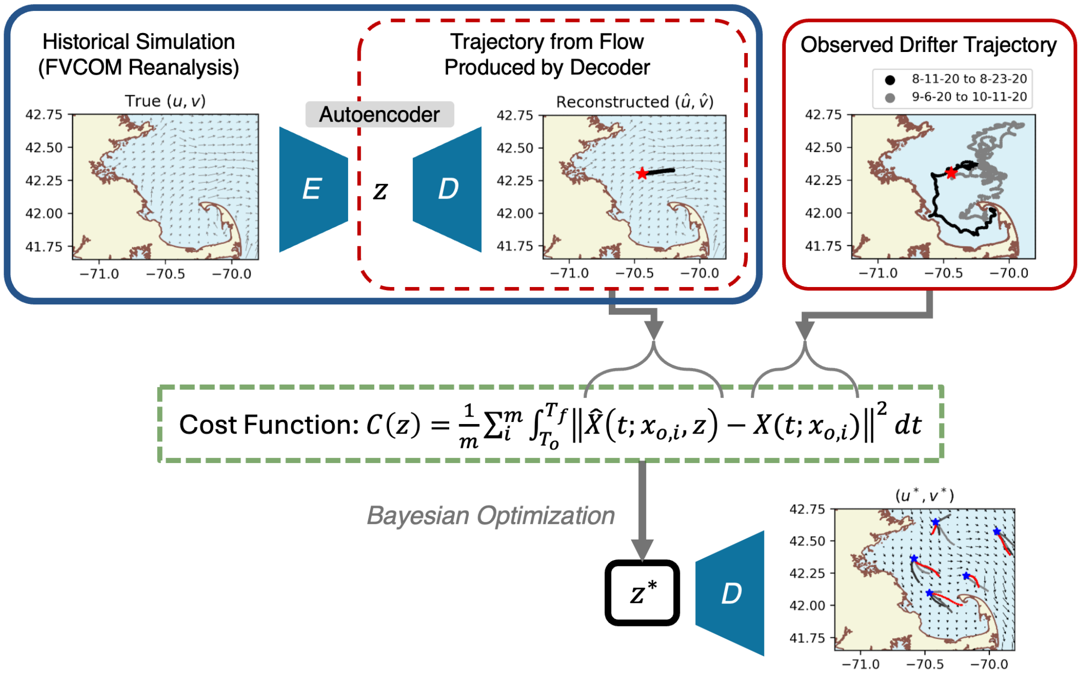

Fig. 1. Summary of framework. First, we train an autoencoder on snapshots from a numerical simulation of the region of interest. The decoder D can be used to generate new flow snapshots from the latent space z. Then, we use Bayesian optimization to identify the optimal z^* that generates a trajectory most closely aligned with the observed trajectory. Finally, we pass this optimal z^* through the decoder to reconstruct the full flow field.

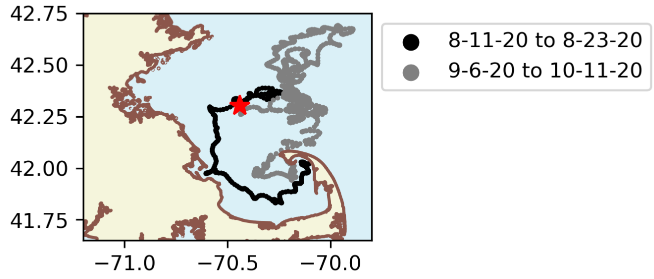

Fig. 2. Trajectories of two Sea Grant Microstar drifters. The first was released on August 11th, 2020 and recovered on August 23rd, 2020. The second was released on September 6th, 2020 and recovered on October 11th, 2020.

formulates the problem as an optimization of a cost function, often using adjoint models to efficiently compute gradients [9]–[11]. Meanwhile, EnKF evaluates an ensemble of model states and uses the posterior distribution to estimate the mean state, with the variance of the posterior providing a measure of uncertainty [12]–[14]. Each of these methods has challenges: OI does not scale well with increasing dimensionality, variational assimilation requires the use of complicated or expensive adjoint models, and EnKF is costly due to the need to maintain and propagate an ensemble of model states.

Another challenge in Lagrangian data assimilation is the high dimensionality of ocean models. Strategies to address this include localized EnKF techniques for dimensionality reduction [15] and hybrid Lagrangian-Eulerian approaches [16]. More recently, machine learning has been explored for reduced-order modeling, such as using recurrent neural networks to predict modal coefficients [17]. Other advancements involve the coupling of Lagrangian data with satellite imagery for improved surface transport estimates [18], [19] or the use of Lagrangian observations to recover Eulerian statistics [20], [21].

Beyond state estimation, some studies have focused on optimizing drifter deployments for targeted observations [22]. Several studies have focused on applying these methods to specific ocean regions, demonstrating their effectiveness in diverse environments such as the Adriatic Sea [23], the Mediterranean [24], and the Gulf of Mexico during the GLAD experiment [25]. Lagrangian assimilation is a useful tool for studying ocean transport processes, including pollution dispersion. For example, [26] applied these techniques to investigate plastic transport in Massachusetts Bay, while [27] investigated plastic dispersion through numerical simulations with modeled drifter trajectories.

These applications highlight the broad utility of Lagrangian data assimilation in both operational and research contexts, motivating novel approaches to further improve prediction accuracy and computational efficiency.

C. Contributions

Given the growing interest in using machine learning for fluid mechanics, we introduce a novel method for Lagrangian data assimilation , outlined in Figure 1 [28]. First, we train an autoencoder using existing snapshots of the flow obtained from a numerical simulation (Section I). This reduced-order model, which uses deep neural networks to encode and decode the data, can be likened to techniques such as principal component analysis (PCA) or empirical orthogonal functions (EOF), which also extract low-dimensional representations of high-dimensional datasets. Next, we apply Bayesian optimization to minimize a cost function that quantifies the difference between the observed Lagrangian trajectory and the trajectory generated by the flow derived from the low-dimensional latent space of the reduced-order model (Section II).

This proposed method is most similar to the framework in [17] which uses a reduced-order model based on EOF. In both methodologies, the data assimilation is applied to the latent space of the selected reduced-order model. We find that the autoencoder is superior to PCA/EOF for modeling historical ocean flow patterns because it is better able to capture nonlinear dynamics. The papers differ because [17] uses a recurrent neural network to capture the temporal dynamics of the EOF coefficients, allowing for the use of an EnKF to assimilate the observations in time. We only consider individual snapshots in time, and we use Bayesian optimization to perform the assimilation.

We recognize that because our method is based on a reduced-order model and our method uses black-box optimization, the resulting predictions may not be as high-fidelity as those of a traditional numerical ocean model. However, we believe that our method is useful for making predictions in real time given new observations. Furthermore, our data assimilation method produces “global” estimates for the whole region as opposed to other data assimilation methods which only correct models near the observations.

We first demonstrate the success of the method on a simulated Kolmogorov flow (Section III) before applying it to a real-world dataset of the Massachusetts and Cape Cod Bays (Section IV). Modeling ocean flow in this region is important for tracking the dispersion of wastewater from the Deer Island Wastewater Treatment Plant.

I. Reduced Order Modeling with an Autoencoder

We first develop a reduced-order surrogate model for the region of interest by training an autoencoder on snapshots from a numerical simulation of the region. Autoencoders are neural networks that learn low-dimensional representations of data, known as the latent space z. They consist of two parts: an encoder E, which maps the input to z, and a decoder D, which reconstructs the input from this low-dimensional representation (Figure 3). Because these neural networks use nonlinear activation functions, they are better able to learn nonlinear manifolds for the latent space compared to standard linear modal decompositions [29], [30].

The architecture of the encoder and the decoder, shown in Figure 3, consists of four convolutional layers. The kernel of each convolutional layer has dimension 3×3 and the stride is 1. The encoder starts with two channels, one for each component of velocity (u, v). The encoder has increasing channel dimensions (2 to 32 to 64 to 128 to 256), and the layers are separated by a ReLU activation function and a max-pooling operation, progressively reducing the spatial resolution of the input and extracting high-level features while retaining essential information. In the final layer of the encoder, the tensor is flattened and passed through a fully connected layer to be reduced to the latent dimension. The decoder reverses the encoding process, expanding the latent space back into the original spatial dimensions. It begins with a fully connected layer that expands the latent representation back into the feature space of $256 \times 4 \times 4$. This is followed by four convolutional layers with decreasing channel dimensions (256 to 128 to 64 to 32 to 2). Instead of max-pooling, the decoder uses bilinear upsampling to progressively increase the spatial resolution of the data back to its original size, and ReLU activations are applied after each convolution. This structure allows the decoder to utilize the compressed latent representation to accurately reconstruct the flow fields.

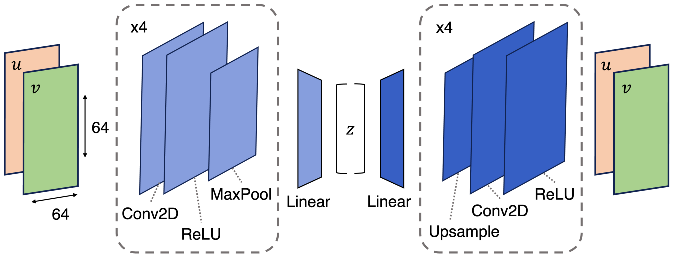

Fig. 3. Architecture of autoencoder. The autoencoder has an encoder and a decoder, each made up of blocks consisting of convolutional layers and nonlinear activation functions. The encoder uses max pooling to reduce the dimensionality while the decoder uses upsampling to recover the original dimensionality. The dimension of the latent space z is a flexible parameter.

To arrive at this final architecture, we experimented with the size of the kernel, the stride, the number of layers, the number of channels, the activation function, and the dimension of the latent space. We also experimented with applying the autoencoder to a dataset of vorticity instead of velocity.

To train the autoencoder, we split the available data into training (30%) and validation (70%) sets. The neural network is optimized with respect to the mean squared error (MSE) loss function using the Adam optimizer. The training and validation losses are computed during each epoch, and training is interrupted when the validation loss does not improve for more than 10 consecutive epochs.

II. Inferring the Flow with Bayesian Optimization

A. Problem Setup

Given a random flow field that has been parametrized through an autoencoder in the form:

$$u(x, t; \omega) = \varphi(x; z(t; \omega)), \tag{1}$$

where $\omega \in \Omega$ is the random argument, and $z(t) \in \mathbb{R}^n$ is the latent variable which can be modeled as stochastic processes with known probability density function (pdf), $f_z(z)$. This pdf can be obtained by using the encoder on a historical record of the flow. In what follows, we will focus on the case where the flow is either steady or slowly varying, i.e. over the interval that we observe it, the latent variables can be assumed to be constant. For ocean applications, this is a good assumption as the flow is varying slowly, and we can look at the characteristic timescales of the flow to determine an appropriate length of time for which this assumption is appropriate.

Suppose we have a realization of the flow, $v(x)$, that can only be observed through a set of Lagrangian trajectories (with arbitrary but known initial conditions, $(t_{0,i}, x_{0,i})$, and length, T_i):

$$\mathcal{X}(t; x_{0,i}), \quad t \in [t_{0,i}, t_{0,i} + T_i], \quad i = 1, ..., m, \tag{2}$$

where for each trajectory i:

$$\frac{d}{dt}\mathcal{X}(t; x_{0,i}) = v(\mathcal{X}(t; x_{0,i})), \qquad \text{with} \;\; \mathcal{X}(t_0; x_{0,i}) = x_{0,i}, \tag{3}$$

our goal is to infer the value of the stochastic vector $z \in \mathbb{R}^n$ that corresponds to the flow field $v(x)$.

We define a cost function $C(z)$ that depends on the latent variable z and measures the integrated distance between m trajectories $\hat{\mathcal{X}}_i(z)$ obtained from a flow produced by decoding the latent variable z and the observed drifter trajectories $\mathcal{X}_i$.

$$\mathcal{C}(z) = \frac{1}{m}\sum_{i=1}^{m}\int_{t_{0,i}}^{t_{0,i}+T_i} \left\| \hat{\mathcal{X}}(t; x_{0,i}; z) - \mathcal{X}(t; x_{0,i}) \right\|^2 dt \tag{4}$$

B. Algorithm

We use Bayesian optimization (BO) to identify the optimal latent variable z^* to reconstruct the flow. To perform BO, a small initial dataset is constructed by computing the cost function $C(z)$ for a random set of possible latent variables z. The surrogate is trained with the initial dataset, and this surrogate is used to calculate the acquisition function. Points with the lowest acquisition function are sequentially added to the training set, and the surrogate model is updated. After some number of iterations of the algorithm, the optimal z^* is the point in the dataset that results in the lowest $C(z)$. The algorithm is also described in pseudocode below.

Algorithm 1 Bayesian Optimization for Minimizing $C(z)$

```
Require: Function $C(z)$ to minimize, GP prior, acquisition function $a(z)$
    Initialize $\mathcal{D}_0 = \{(z_j, C(z_j))\}_{j=1}^{n}$ with $n$ initial samples
    Fit GP surrogate model to $\mathcal{D}_0$

    for $t = 1, 2, \ldots, T$ do
        Compute surrogate GP posterior: $\mu_t(z)$ and $\sigma_t^2(z)$
        Define acquisition function $a(z)$ using $\mu_t(z)$ and $\sigma_t^2(z)$
        Find next query point: $z_{t+1} = \arg\min_z a(z)$
        Evaluate $C(z_{t+1})$ and update dataset:
            $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(z_{t+1}, C(z_{t+1}))\}$
        Refit GP surrogate model to $\mathcal{D}_{t+1}$
    end for

    Return $z^* = \arg\min_{z \in \mathcal{D}_T} C(z)$
```

1) Surrogate: To model the relationship between z and $C(z)$ we test both standard Gaussian Process (GP) Regression (GPR) from `GPy` for which the kernel is the radial basis function and deep GPR from `GPyTorch` [31]. We also tested using an ensemble of neural networks as the surrogate model for $C(z)$, and we also experimented with particle swarm optimization and a genetic algorithm, but we found the results to be best with GPR.

2) Acquisition Function: We test both the lower confidence bound (LCB) criterion and the expected improvement (EI) criterion to minimize the cost function [32]–[34]. These acquisition functions guide the exploration-exploitation tradeoff in Bayesian optimization by deciding where to sample next in the search space. The LCB criterion is given by

$$a_{\text{LCB}}(x) = \mu_t(x) - \kappa \cdot \sigma_t(x), \tag{5}$$

where $\mu_t(x)$ and $\sigma_t(x)$ represent the posterior mean and standard deviation of the Gaussian process at iteration t, and the parameter $\kappa > 0$ controls the balance between exploration (sampling points with high uncertainty) and exploitation (sampling near the predicted minimum).

We also experiment with the EI criterion, which seeks to maximize the expected gain over the current best observed value. The EI is given by

$$a_{\text{EI}}(x) = (\mu(x) - y_{\text{best}})\,\Phi\left(z\right) + \sigma(x)\phi\left(z\right), \tag{6}$$

where y_{best} is the best observed value so far, Φ is the cumulative distribution function (CDF) of the standard normal distribution, ϕ is the probability density function (PDF) of the standard normal distribution, and $z = (\mu(x) - y_{\text{best}})/\sigma(x)$.

C. Characteristic Timescale

We determine the characteristic timescale by analyzing the autocorrelation of the velocity components u and v. For each component, we compute the autocorrelation function and define the characteristic timescale T as the time lag at which the autocorrelation drops to $1/e$. This timescale reveals how persistent the flow dynamics remain over time; longer timescales correspond to more persistent patterns. By selecting an appropriate timescale T, we assume that the flow remains steady or varies slowly over the time domain of interest.

III. Kolmogorov Flow

We first test the proposed method on a protoyipical two-dimensional Kolmogorov flow, a simple model of fluid flow driven by sinusoidal forcing. The simplicity of the flow allows for controlled testing while still providing a good example of key features found in more complex ocean dynamics.

A. Data

To generate data, we start with the incompressible 2D Navier Stokes

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \tag{7}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{8}$$

In component form, the equations are:

$$\frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} = -\frac{\partial p}{\partial x} + \nu \nabla^2 u_x + f_x \tag{9}$$

$$\frac{\partial u_y}{\partial t} + u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} = -\frac{\partial p}{\partial y} + \nu \nabla^2 u_y + f_y \tag{10}$$

with vorticity

$$\omega = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \tag{11}$$

The equations can be transformed into Fourier space where

$$\hat{u}_x = \mathcal{F}(u_x) \text{ and } \hat{u}_y = \mathcal{F}(u_y) \tag{12}$$

and

$$\hat{\omega}(k_x, k_y) = ik_x\hat{u}_y - ik_y\hat{u}_x \tag{13}$$

To find a solution to the equation with a numerical solver, we write the time evolution of the velocity components $\hat{u}_x$ and $\hat{u}_y$ in Fourier space:

$$\frac{\partial}{\partial t}\begin{bmatrix}\hat{u}_x \\ \hat{u}_y\end{bmatrix} = \begin{bmatrix}-\hat{u}_y\hat{\omega} \\ \hat{u}_x\hat{\omega}\end{bmatrix} - \nu k^2 \begin{bmatrix}\hat{u}_x \\ \hat{u}_y\end{bmatrix} + \begin{bmatrix}\hat{f}_x \\ \hat{f}_y\end{bmatrix} \tag{14}$$

where

$$k^2 = k_x^2 + k_y^2 \tag{15}$$

We set the forcing in one direction

$$f_x = \sin(k_f y) \text{ and } f_y = 0 \tag{16}$$

We use a fourth order Runge-Kutta (`MATLAB ode45`) to solve the equation for $\hat{u}_x$ and $\hat{u}_x$ from which we obtain u_x and u_y. We run the simulation on a torus $[0, 2\pi)^2$ with a timestep of 0.5 to generate 20,000 snapshots.

B. Autoencoder Results

We test latent dimensions of 20 and 50, and we use 6000 snapshots for training and 14000 for validation. The validation root mean squared errors (RMSE) for the Kolmogorov flow are listed in Table I. The errors from the CNN autoencoder are lower than both those from PCA and those from fully connected autoencoder (which does not use convolutional layers). We also recover the maximum vorticity from the predictions and show in Figure 4 that the CNN is better able to capture the important large transient features. Similarly, Figure 5 shows the autoencoder reconstruction for a snapshot with an average error. Overall, the CNN autoencoder with a latent dimension of 20 is a suitable reduced-order model for the Kolmogorov flow.

TABLE I
RMSE VALUES FOR DIFFERENT REDUCED-ORDER MODELS FOR THE KOLMOGOROV DATASET WITH LATENT DIMENSIONS 20 AND 50

Model	**Latent 20**			**Latent 50**		
	RMSE u	**RMSE v**	**RMSE ω**	**RMSE u**	**RMSE v**	**RMSE ω**
PCA	0.226	0.322	1.378	0.108	0.168	0.973
FCNN	0.394	0.182	1.336	0.378	0.150	1.276
CNN	0.117	0.124	0.736	0.084	0.105	0.615

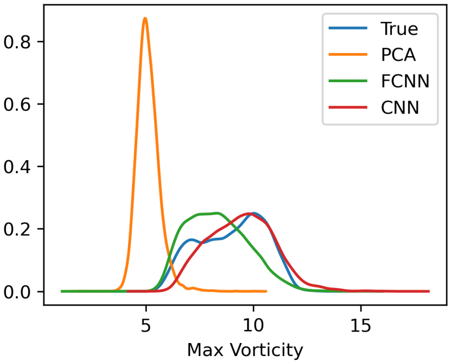

Fig. 4. Probability density function of the predicted maximum vorticity obtained for the three dimensionality reduction methods (PCA, FCNN, CNN) for a latent dimension of 20. The CNN best captures the true maximum vorticity.

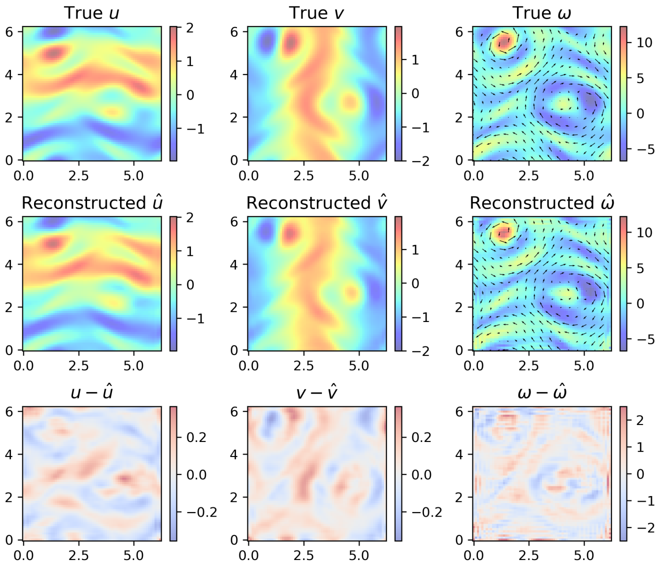

Fig. 5. Comparison between true flow fields and fields reconstructed from the autoencoder with latent dimension of 20 at a timestep with an average model error. The first row shows the true flow field, the second row shows the predicted flow field, and the third row shows the difference between the true and predicted fields. The left column shows the horizontal velocity component, the middle column shows the vertical velocity component, and the right column shows the vorticity.

C. Optimization Results

We randomly select three trajectories for a random timestep, and we run the optimization for 100 iterations using a standard GP surrogate with a latent dimension of 20 and the LCB acquisition function. During the analysis, we compare the reconstruction from the optimal solution z^* (Figure 6) to the reconstruction that results from taking the average of the top five solutions (Figure 7). We find that taking the top five solutions can be used to estimate uncertainty, and in some cases can increase the overall accuracy.

1) Convergence Analysis: To further investigate the method, we performed tests examining the effect of the

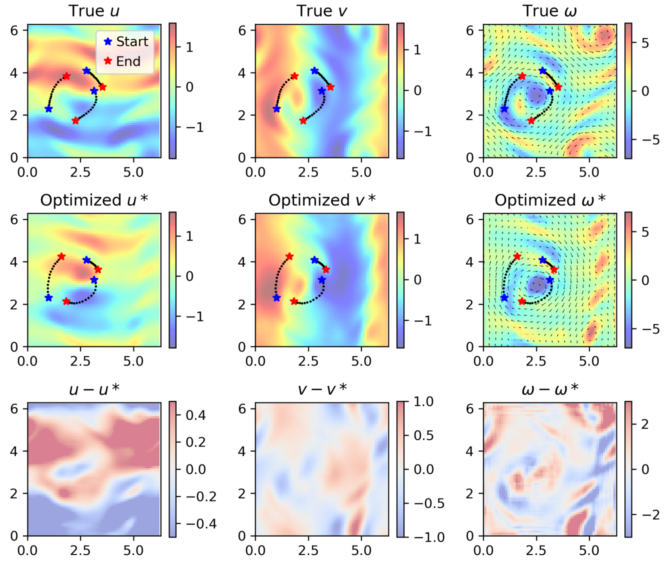

Fig. 6. Results from running the optimization with three drifter trajectories for 150 iterations with the LCB acquisition function, a latent dimension of 20, and a standard GP surrogate. The top row shows the true flow field with the true trajectories. The second row shows the reconstructed flow field obtained by decoding the optimal latent variable, along with the corresponding trajectories that minimized the cost function. The third row shows the error. The left column shows the horizontal velocity component, the middle column shows the vertical velocity component, and the right column shows the vorticity.

drifters' initial release point with the goal of evaluating the stability and robustness of the method and the sensitivity of the reconstruction to inputs. In both tests, we ran each optimization for 75 iterations, with the LCB acquisition function, a latent dimension of 50, and a deep GP surrogate.

a) Fixing the Initial Release Position of the Drifters for Different Timesteps: In this experiment, we performed the optimization for trajectories released from the same three spatial positions for 1000 different timesteps. Averaged over the 1000 timesteps, the reconstruction error is lowest in the neighborhood near the release of the drifters, suggesting that the model performs better in regions with more observations (Figure 8). However, the error far from the drifters' release points remains reasonable, confirming that our method is useful for obtaining a global estimate of the flow field.

b) Varying the Initial Release Position of the Drifters for a Fixed Timestep: Here, we fixed the timestep and varied the initial spatial location at which the drifters are released by randomly selecting sets of three locations for each of the 1000 experiments. This experiment revealed that prediction is improved when drifters are released near coherent flow structures, such as eddies or fronts (Figure 9).

IV. Massachusetts and Cape Cod Bays

Having demonstrated the success of the method on the Kolmogorov flow, we apply the framework to a reanalysis dataset of the Massachusetts and Cape Cod Bays, and we eventually show how the method can be used with real-world observed trajectories.

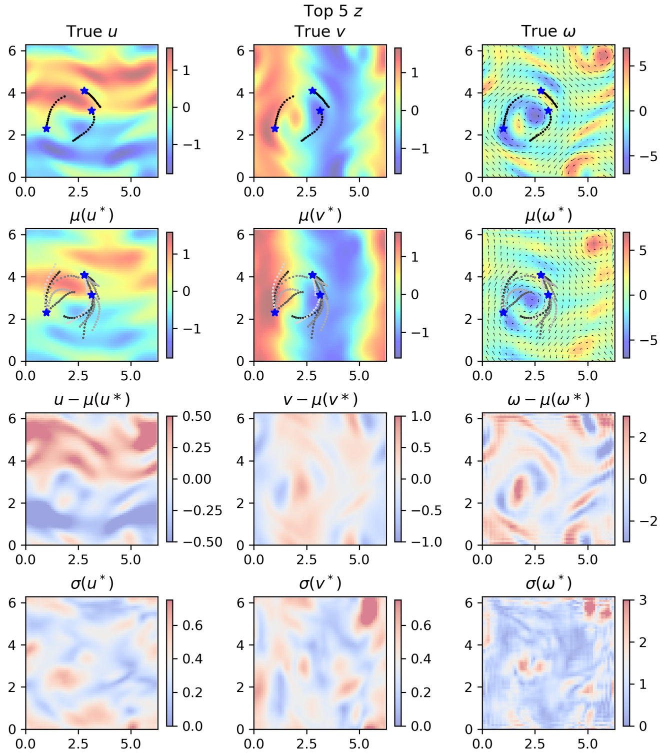

Fig. 7. Results from taking the five best z that were obtained from running the optimization with three trajectories for 150 iterations with the LCB acquisition function, a latent dimension of 20, and a standard GP surrogate. The top row shows the true field with the true trajectories. The second row shows the reconstructed field, obtained by taking the average of decoding the top five optimal latent variables, along with the corresponding trajectories that minimized the cost function. The third row shows the error. The last row shows the standard deviation of the top five flows, serving as a way to estimate uncertainty.

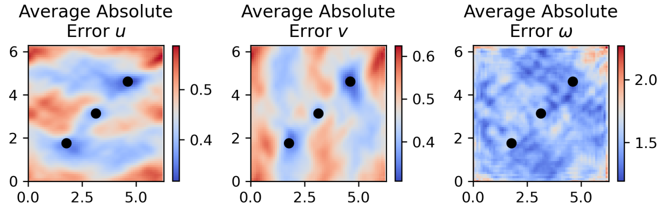

Fig. 8. Average absolute error for u (left), v (center), ω (right) reconstructed with three trajectories for 1000 different time steps of the Kolmogorov flow with the same drifter initial position. The black points represent the initial release point of the three drifters. The average absolute error is lowest near the release point for u and v. In this experiment, we use the autoencoder with latent dimension 50, a deep GP surrogate, and the LCB acquisition function.

A. Data: Finite Volume Community Ocean Model

The Finite Volume Community Ocean Model is a comprehensive high resolution physics-based reanalysis model that integrates real-world measurements in the numerical simulation [35]. This model has been used for the Northeast Coastal Ocean Forecast System (NECOFS), a region with complex coastlines, freshwater sources, and a significant fishing industry. The model consists of daily estimates for eastward,

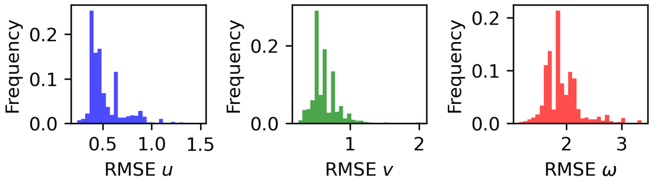

Fig. 9. Histogram of the root mean square error for u (left), v (center), ω (right) at one time step reconstructed with three trajectories for 1000 different sets of random drifter initial position. In this experiment, we use the autoencoder with latent dimension 50, a deep GP surrogate, and the LCB acquisition function.

northward, and upward velocity among other variables from 2005 to 2013 at 45 sigma layers, but we only consider the 2D velocity at the surface. While FVCOM is accurate and fine-grained, it is a very complex model that requires significant parameter tuning. The data are only available for past years, and the model is too expensive and slow to run in real time.

In addition to the data from the model, we have measurements of Lagrangian trajectories from passive drifters, some of which were collected by the MIT Sea Grant office (Figure 2) and others by the Student Drifter Program. There also exist other sensor measurements including wind speed from buoys, sea surface height from satellites, and tide charts. We hope to incorporate this information into our model as needed in the future.

B. Autoencoder Results

For the Massachusetts and Cape Cod Bays, we test latent dimensions of 50 and 75 to achieve reasonable reconstruction accuracies; however, these choices require the use of a deep GP surrogate, as standard GPs do not scale well with higher dimensions. From the nine years of reanalysis data, we use the first 4642 snapshots for training and the following 10833 for validation. We applied a mask in the loss function to ignore any values that are predicted over land. The validation RMSE for u and v are listed in Table II. Again, the CNN autoencoder outperforms PCA.

TABLE II
RMSE VALUES FOR DIFFERENT REDUCED-ORDER MODELS FOR THE FVCOM DATASET WITH LATENT DIMENSIONS 50 AND 75

Model	**Latent 50**		**Latent 75**	
	RMSE u (m/s)	**RMSE v (m/s)**	**RMSE u (m/s)**	**RMSE v (m/s)**
PCA	0.0605	0.0635	0.0515	0.0544
CNN	0.0564	0.0576	0.0512	0.0518

C. Optimization Results

1) Validation with Synthetic Trajectories: We first validate the method on synthetic trajectories — trajectories generated by numerically simulating the advection of passive particles through a known velocity field. We generate five synthetic drifter trajectories with random initial positions from the flow field at a random timestep, and we run the optimization for 100 iterations with the LCB criterion. We show the results from taking the average of the top five solutions in Figures 11 and 12.

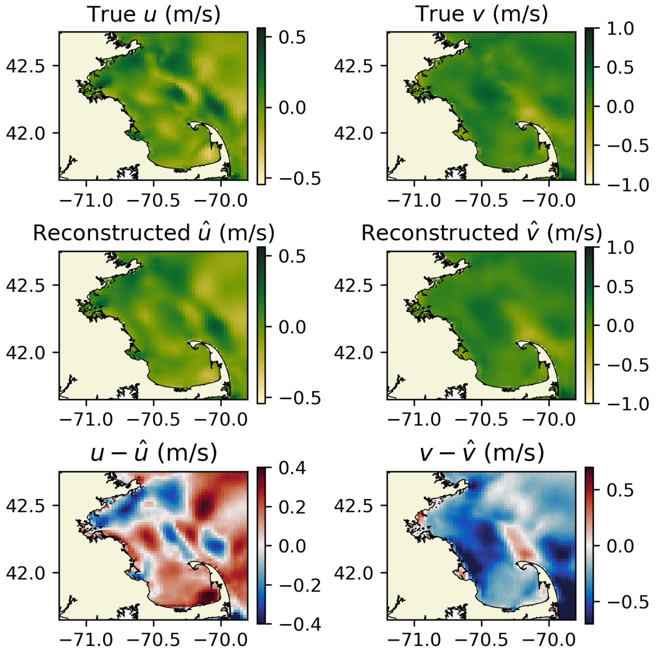

Fig. 10. Comparison between true field and field reconstructed from autoencoder with a latent dimension of 75 at a timestep with an average model error (June 14th, 2010). The first row shows the true field, the second row shows the predicted field, and the third row shows the difference between the true and predicted fields. The left and right columns correspond to the horizontal and vertical velocity components, respectively.

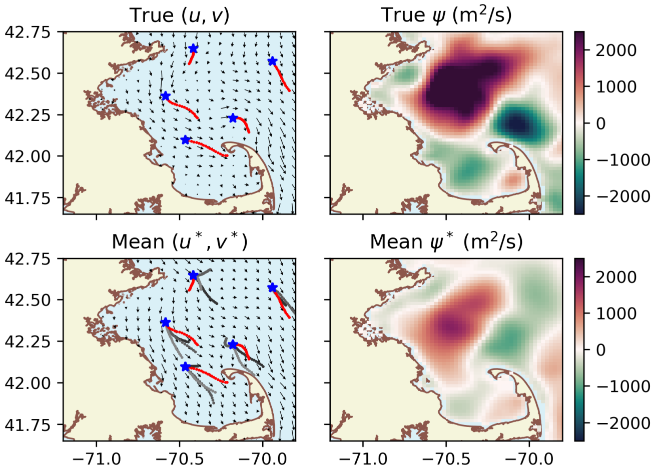

Fig. 11. Results from taking the five best z that were obtained from running the optimization with five synthetic trajectories (June 8th, 2008) for 100 iterations with the LCB acquisition function, an autoencoder with latent dimension 75, and a deep GP surrogate for the cost function. The top row shows the true (u, v) field and the streamfunction ψ. The second row shows the $(u*, v*)$ and $\psi*$ reconstructed from taking the average of decoding the top five optimal latent variables, along with the corresponding trajectories that minimized the cost function.

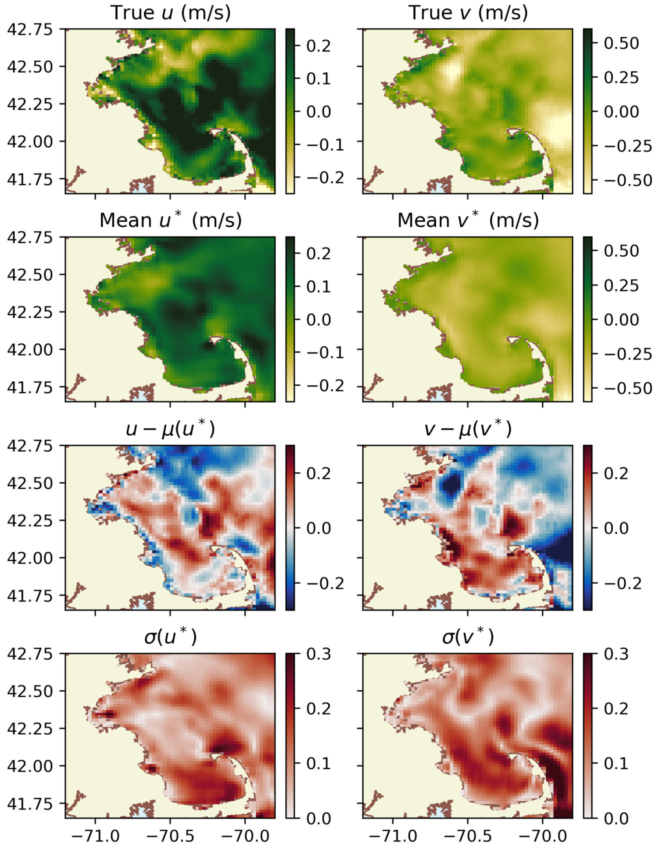

Fig. 12. Results from taking the five best z that were obtained from running the optimization with five synthetic trajectories (June 8th, 2008) for 100 iterations with the LCB acquisition function, an autoencoder with latent dimension 75, and a deep GP surrogate for the cost function. The first row shows the true velocities, the second row shows the mean of the predictions, the third row shows the difference, and the fourth row shows the standard deviation of the predictions. This standard deviation serves as a useful way to measure uncertainty because regions with high standard deviation match regions with high error.

2) Application to Real-World Observed Trajectories: Finally, we apply the method to the real observed trajectories. We were unable to use the Microstar trajectories from 2020 because there is only one observation at any given time, and our previous experiments demonstrated that optimization is best when using several simultaneous trajectories. Given this constraint, our best option was to use observations from the Student Drifter Program in 2010 for which we have more simultaneous trajectories in the Massachusetts Bay. We run the optimization to generate a flow field that could have potentially generated the observed trajectory. We show the results from taking the mean of the top three predictions in Figure 13. Given that we only have measurements from one drifter, we have no way to validate the predictions. By launching more simultaneous drifters, we could better evaluate the success of the method in the real world.

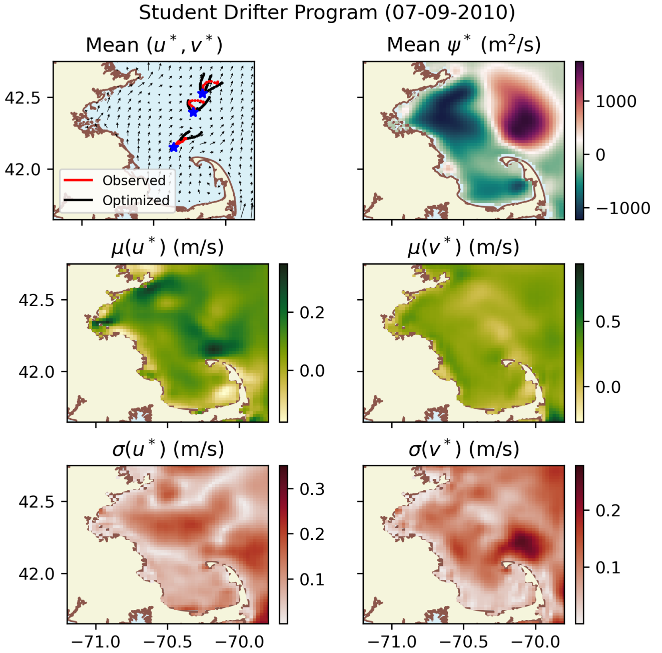

Fig. 13. Results from taking the three best z that were obtained from running the optimization using three real-world observed trajectories from July 9th, 2010 as input. The top left shows the predicted velocity field, the top right shows the predicted streamfunction, the second row shows the predicted velocities, and the third row shows the standard deviation of the three predicted velocities which can be used for uncertainty quantification. The velocity field also shows the observed trajectories in red and the trajectories found from the optimization in black.

V. Conclusion

Our methodology leverages a combination of reduced-order modeling and Bayesian optimization to reconstruct regional ocean surface flow from sparse Lagrangian drifter observations. First, we train an autoencoder using snapshots from a numerical simulation to create a reduced-order model that efficiently captures the essential flow dynamics in a low-dimensional latent space. Then, we apply Bayesian optimization to minimize a cost function that quantifies the discrepancy between observed drifter trajectories and trajectories generated by the latent-space flow representation. This optimization guides the reconstruction of the underlying flow field.

Overall, the method is able to reconstruct a regional ocean surface flow in minutes using observed Lagrangian drifter trajectories, and it is useful for determining the minimum number of trajectories needed to accurately describe the flow. This research will inform the purchase and deployment of future Microstar drifters for improved understanding of wastewater dispersion in the Massachusetts Bay. Going forward, the method can be used to determine where to launch drifters, and the reconstructed surface flow can be used to estimate the full vertical profile of the region of interest [36].

Acknowledgment

This research was funded by MIT NOAA Sea Grant and the National Science Foundation Graduate Research Fellowship.

We thank the scientists at the MIT NOAA Sea Grant office for motivating the project. We thank Michael DeFilippo for collecting the Microstar drifter observations. We thank James Manning for providing the observations from the Student Drifter Program.

REFERENCES

[1] P. F. J. Lermusiaux, P. Malanotte-Rizzoli, D. Stammer, J. Carton, J. Cummings, and A. M. Moore, "Progress and Prospects of U.S. Data Assimilation in Ocean Research," *Oceanography*, Mar. 2006. [Online]. Available: https://doi.org/10.5670/oceanog.2006.102

[2] D. Ciani, E. Charles, B. Buongiorno Nardelli, M.-H. Rio, and R. Santoleri, "Ocean Currents Reconstruction from a Combination of Altimeter and Ocean Colour Data: A Feasibility Study," *Remote Sensing*, vol. 13, no. 12, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/12/2389

[3] J. Manning, E. Pelletier, A. Smith, and C. Stymiest, "Student built, Fishermen deployed, Satellite tracked Drifters," Aug. 2014. [Online]. Available: https://www.gulfofmaine.org/public/ecosystem-indicator-partnership/monthly-journals/2014-08/

[4] I. I. Rypina, A. Macdonald, S. Yoshida, J. P. Manning, M. Gregory, N. Rozen, and K. Buesseler, "Spreading pathways of Pilgrim Nuclear Power Station wastewater in and around Cape Cod Bay: Estimates from ocean drifter observations," *Journal of Environmental Radioactivity*, vol. 255, p. 107039, Dec. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0265931X22002302

[5] L. Sun, S. G. Penny, and M. Harrison, "Impacts of the Lagrangian Data Assimilation of Surface Drifters on Estimating Ocean Circulation during the Gulf of Mexico Grand Lagrangian Deployment," *Monthly Weather Review*, vol. 150, no. 4, pp. 949–965, Apr. 2022. [Online]. Available: https://journals.ametsoc.org/view/journals/mwre/150/4/MWR-D-21-0123.1.xml

[6] A. Molcard, L. I. Piterbarg, A. Griffa, T. M. Özgökmen, and A. J. Mariano, "Assimilation of drifter observations for the reconstruction of the Eulerian circulation field," *Journal of Geophysical Research: Oceans*, vol. 108, no. C3, p. 2001JC001240, Mar. 2003. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2001JC001240

[7] T. M. Özgökmen, A. Molcard, T. M. Chin, L. I. Piterbarg, and A. Griffa, "Assimilation of drifter observations in primitive equation models of midlatitude ocean circulation," *Journal of Geophysical Research: Oceans*, vol. 108, no. C7, p. 2002JC001719, Jul. 2003. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2002JC001719

[8] A. Molcard, A. Griffa, and T. M. Özgökmen, "Lagrangian Data Assimilation in Multilayer Primitive Equation Ocean Models," *Journal of Atmospheric and Oceanic Technology*, vol. 22, no. 1, pp. 70–83, Jan. 2005. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/JTECH-1686.1

[9] J. Mead and A. Bennett, "Towards regional assimilation of Lagrangian data: the Lagrangian form of the shallow water model and its inverse," *Journal of Marine Systems*, vol. 29, no. 1-4, pp. 365–384, May 2001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924796301000240

[10] M. Nodet, "Variational assimilation of Lagrangian data in oceanography," *Inverse Problems*, vol. 22, no. 1, pp. 245–263, Feb. 2006. [Online]. Available: https://iopscience.iop.org/article/10.1088/0266-5611/22/1/014

[11] V. Taillandier, A. Griffa, and A. Molcard, "A variational approach for the reconstruction of regional scale Eulerian velocity fields from Lagrangian data," *Ocean Modelling*, vol. 13, no. 1, pp. 1–24, Jan. 2006. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1463500305000776

[12] L. Kuznetsov, K. Ide, and C. K. R. T. Jones, "A Method for Assimilation of Lagrangian Data," *Monthly Weather Review*, vol. 131, no. 10, pp. 2247–2260, Oct. 2003. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/1520-0493(2003)131¡2247:AMFAOL¿2.0.CO;2

[13] H. Salman, L. Kuznetsov, C. K. R. T. Jones, and K. Ide, "A Method for Assimilating Lagrangian Data into a Shallow-Water-Equation Ocean Model," *Monthly Weather Review*, vol. 134, no. 4, pp. 1081–1101, Apr. 2006. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/MWR3104.1

[14] A. Apte, C. K. R. T. Jones, and A. M. Stuart, "A Bayesian approach to Lagrangian data assimilation," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 60, no. 2, p. 336, Jan. 2008. [Online]. Available: https://a.tellusjournals.se/article/10.1111/j.1600-0870.2007.00295.x/

[15] L. Sun and S. G. Penny, "Lagrangian Data Assimilation of Surface Drifters in a Double-Gyre Ocean Model Using the Local Ensemble Transform Kalman Filter," *Monthly Weather Review*, vol. 147, no. 12, pp. 4533–4551, Dec. 2019. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/MWR-D-18-0406.1

[16] Q. Deng, N. Chen, S. N. Stechmann, and J. Hu, "LEMDA: A Lagrangian-Eulerian Multiscale Data Assimilation Framework," *Journal of Advances in Modeling Earth Systems*, vol. 17, no. 2, p. e2024MS004259, Feb. 2025. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2024MS004259

[17] Z. Zhang, M. Hou, F. Zhang, and C. R. Edwards, "An LSTM based Kalman Filter for Spatio-temporal Ocean Currents Assimilation," in *Proceedings of the International Conference on Underwater Networks & Systems*. Atlanta GA USA: ACM, Oct. 2019, pp. 1–7. [Online]. Available: https://dl.acm.org/doi/10.1145/3366486.3366522

[18] D. Béréziat and I. Herlin, "Coupling Dynamic Equations and Satellite Images for Modelling Ocean Surface Circulation," in *Computer Vision, Imaging and Computer Graphics - Theory and Applications*, S. Battiato, S. Coquillart, J. Pettré, R. S. Laramee, A. Kerren, and J. Braz, Eds. Cham: Springer International Publishing, 2015, vol. 550.

[19] A. Sinha and R. Abernathey, "Estimating Ocean Surface Currents With Machine Learning," *Frontiers in Marine Science*, vol. 8, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmars.2021.672477

[20] M. A. Mohamad and A. J. Majda, "Eulerian and Lagrangian statistics in an exactly solvable turbulent shear model with a random background mean," *Physics of Fluids*, vol. 31, no. 10, p. 105115, 2019, _eprint: https://doi.org/10.1063/1.5121705. [Online]. Available: https://doi.org/10.1063/1.5121705

[21] ——, "Recovering the Eulerian energy spectrum from noisy Lagrangian tracers," *Physica D: Nonlinear Phenomena*, vol. 403, p. 132374, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167278918305505

[22] M. Tukan, E. Biton, and R. Diamant, "An Efficient Drifters Deployment Strategy to Evaluate Water Current Velocity Fields," *IEEE Journal of Oceanic Engineering*, vol. 49, no. 4, pp. 1455–1471, Oct. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10547293/

[23] S. Castellari, A. Griffa, T. M. Özgökmen, and P.-M. Poulain, "Prediction of particle trajectories in the Adriatic Sea using Lagrangian data assimilation," *Journal of Marine Systems*, vol. 29, no. 1-4, pp. 33–50, May 2001. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0924796301000082

[24] J. A. U. Nilsson, S. Dobricic, N. Pinardi, P.-M. Poulain, and D. Pettenuzzo, "Variational assimilation of Lagrangian trajectories in the Mediterranean ocean Forecasting System," *Ocean Science*, vol. 8, no. 2, pp. 249–259, Mar. 2012. [Online]. Available: https://os.copernicus.org/articles/8/249/2012/

[25] E. F. Coelho, P. Hogan, G. Jacobs, P. Thoppil, H. Huntley, B. Haus, B. Lipphardt, A. Kirwan, E. Ryan, J. Olascoaga, F. Beron-Vera, A. Poje, A. Griffa, T. Özgökmen, A. Mariano, G. Novelli, A. Haza, D. Bogucki, S. Chen, M. Curcic, M. Iskandarani, F. Judt, N. Laxague, A. Reniers, A. Valle-Levinson, and M. Wei, "Ocean current estimation using a Multi-Model Ensemble Kalman Filter during the Grand Lagrangian Deployment experiment (GLAD)," *Ocean Modelling*, vol. 87, pp. 86–106, Mar. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1463500314001577

[26] P. F. J. Lermusiaux, J. Marshall, T. Peacock, C. Noble, M. Doshi, C. S. Kulkarni, A. Gupta, P. J. Haley, C. Mirabito, F. Trotta, S. J. Levang, and G. R. Flierl, "Plastic Pollution in the Coastal Oceans: Characterization and Modeling," in *OCEANS 2019 MTS/IEEE SEATTLE*. Seattle, WA, USA: IEEE, Oct. 2019, pp. 1–10. [Online]. Available: https://ieeexplore.ieee.org/document/8962786/

[27] A. Peytavin, B. Sainte-Rose, G. Forget, and J.-M. Campin, "Ocean Plastic Assimilator v0.2: assimilation of plastic concentration data into Lagrangian dispersion models," *Geoscientific Model Development*, vol. 14, no. 7, pp. 4769–4780, Jul. 2021. [Online]. Available: https://gmd.copernicus.org/articles/14/4769/2021/

[28] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine Learning for Fluid Mechanics," *Annual Review of Fluid Mechanics*, vol. 52, no. Volume 52, 2020, pp. 477–508, 2020,

publisher: Annual Reviews Type: Journal Article. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-010719-060214

[29] K. Fukami, T. Nakamura, and K. Fukagata, "Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data," *Physics of Fluids*, vol. 32, no. 9, p. 095110, Sep. 2020. [Online]. Available: https://doi.org/10.1063/5.0020721

[30] W. Obayashi, H. Aono, T. Tatsukawa, and K. Fujii, "Feature extraction of fields of fluid dynamics data using sparse convolutional autoencoder," *AIP Advances*, vol. 11, no. 10, p. 105211, Oct. 2021. [Online]. Available: https://doi.org/10.1063/5.0065637

[31] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: Biologische Kybernetik, Jan. 2006, backup Publisher: Max-Planck-Gesellschaft.

[32] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec. 1998. [Online]. Available: https://doi.org/10.1023/A:1008306431147

[33] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[34] A. Blanchard and T. Sapsis, "Bayesian optimization with output-weighted optimal sampling," *Journal of Computational Physics*, vol. 425, p. 109901, Jan. 2021, publisher: Elsevier BV.

[35] C. Chen, H. Liu, and R. C. Beardsley, "An Unstructured Grid, Finite-Volume, Three-Dimensional, Primitive Equations Ocean Model: application to Coastal Ocean and Estuaries," 2003, volume: 20 Pages: 159 - 186 Publication Title: Journal of Atmospheric and Oceanic Technology Issue: 1 Place: Boston MA, USA Publisher: American Meteorological Society.

[36] B. Champenois and T. Sapsis, "Machine learning framework for the real-time reconstruction of regional 4D ocean temperature fields from historical reanalysis data and real-time satellite and buoy surface measurements," *Physica D: Nonlinear Phenomena*, vol. 459, p. 134026, Mar. 2024.