



HAL
open science

Deep learning for Lagrangian drift simulation at the sea surface

Daria Botvynko, Carlos Granero-Belinchon, Simon Van Gennip, Abdesslam Benzinou, Ronan Fablet

► **To cite this version:**

Daria Botvynko, Carlos Granero-Belinchon, Simon Van Gennip, Abdesslam Benzinou, Ronan Fablet. Deep learning for Lagrangian drift simulation at the sea surface. ICASSP 2023: IEEE International Conference on Acoustics, Speech, and Signal Processing, Jun 2023, Rhodes, Greece. 10.1109/ICASSP49357.2023.10094622 . hal-03852489

HAL Id: hal-03852489

<https://hal.science/hal-03852489v1>

Submitted on 16 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DEEP LEARNING FOR LAGRANGIAN DRIFT SIMULATION AT THE SEA SURFACE

Daria Botvynko¹, Carlos Granero-Belinchon², Simon van Gennip³, Abdesslam Benzinou¹, Ronan Fablet²

¹ ENIB, Lab-STICC, UMR CNRS 6285, 29238 Brest

² IMT Atlantique, Lab-STICC, UMR CNRS 6285, 29238 Brest, France

³ Mercator Ocean International, 31400 Toulouse, France

ABSTRACT

We address Lagrangian drift simulation in geophysical dynamics and explore deep learning approaches to overcome known limitations of state-of-the-art model-based and Markovian approaches in terms of computational complexity and error propagation. We introduce a novel architecture, referred to as DriftNet, inspired from the Eulerian Fokker-Planck representation of Lagrangian dynamics. Numerical experiments for Lagrangian drift simulation at the sea surface demonstrates the relevance of DriftNet w.r.t. state-of-the-art schemes. Benefiting from the fully-convolutional nature of DriftNet, we explore through a neural inversion how to diagnose model-derived velocities w.r.t. real drifter trajectories.

Index Terms— Lagrangian drift * Deep learning * Advection scheme * Fokker-Planck * Sea Surface Dynamics

1. INTRODUCTION

The modeling and simulation of Lagrangian drift at sea surface is a key scientific challenge in operational oceanography [1]. Applications of interest range from plastic and other debris tracking [2], algae or plankton drift [3], to the forecasting of drifting objects for search and rescue operations [4]. The analysis of Lagrangian drifts is also widely used to diagnose ocean numerical models w.r.t. Lagrangian trajectories observed in the field [5].

From a methodological point of view, we may distinguish two broad categories of approaches for Lagrangian drift simulation: model-based approaches [6] and Markov data-driven schemes [7]. Model-based approaches rely on the implementation of a sequential advection process knowing the underlying velocity fields [8]. Due to its sequential nature, these schemes are not highly-scalable to simulate large ensembles of drift trajectories. Besides, small errors in the underlying velocity fields may strongly affect the trajectory patterns. The latter is a critical issue as, for operational applications, these velocity fields are prone to inversion errors. By contrast, Markov data-driven schemes naturally account for uncertainties in the drift process but mainly apply to relatively coarse space-time resolutions. Their extension to fine-scale patterns is a challenge, [9]. We also note that in both approaches the simulation relies at each time-step on a location-wise velocities only, which may only apply for smooth velocity fields.

Recently, a growing literature has emerged for applications of deep learning to trajectory data, including among others, generative models [10], [11], [12], [13], [14], short-term forecasting [15]

and classification issues [16]. Most of the studies using generative approaches rely on sequential approaches and recurrent neural networks, which may face the same limitations as the ones identified above for the state-of-the-art schemes for Lagrangian drift simulation. To our knowledge, only very few previous studies have explored the conditional simulation of Lagrangian drift [17]. We may however note that some studies stress the potential for considering higher-dimensional latent representation of trajectory data [18].

Here, we propose a novel deep learning scheme for the conditional simulation of Lagrangian drifts. We introduce a convolutional architecture with a spatially-explicit latent representation, inspired by Eulerian Fokker-Planck representations of drift processes. It results in a non-local simulation of drift trajectories conditionally to velocity fields. In our numerical experiments, we consider an application to Lagrangian drift simulation at the sea surface for a case-study region off the coast of California. The reported results support the relevance of this approach. Through an inverse problem formulation, we further illustrate how our approach provides new means to inform sea surface velocities from real drift trajectories.

The remainder is organized as follows. Section 2 provides some background. We introduce the proposed deep learning approach in Section 3 and report numerical experiments in Section 4. Section 5 discusses further our main contributions.

2. PROBLEM STATEMENT

Lagrangian fluid dynamics consider the fluid as an ensemble of particles and describe the flow properties through the trajectories of the particles advected by the flow [19]. This results in particle trajectories governed by the following ordinary differential equation:

$$\frac{\partial \vec{r}(\vec{r}_0, t)}{\partial t} = \vec{v}(\vec{r}_0, t) = \vec{u}(\vec{r}(t), t) \quad (1)$$

where $\vec{r}(\vec{r}_0, t)$ and $\vec{v}(\vec{r}_0, t)$ are respectively the position and velocity at time t of the particle initially located at \vec{r}_0 , and $\vec{u}(\vec{r}(t), t)$ is the Eulerian velocity of the flow at position $\vec{r}(t)$ and time t . State-of-the-art model-driven schemes implement such Lagrangian drift dynamics using explicit time integration schemes, such as Runge-Kutta 4. The sequential nature of the above equation clearly propagates and accumulates errors if the underlying velocities are not perfectly known [20]. Besides, from a computational point of view, the location-dependent parameterization of the right-hand term in eq.(1) leads to a relatively poor parallelization potential for the simulation of large ensembles of trajectories.

Given eq.(1), learning-based approaches for Lagrangian drift simulation naturally consider recurrent neural networks as investigated in [21] using LSTM and broadly used from trajectory simulations [15]. This is also the theoretical background for Markov models [22] which can be regarded as an extension of eq.(1) accounting for

This work was supported by LEFE program (LEFE MANU and IMAGO projects IA-OAC), CNES (OSTST DUACS-HR and SWOT ST DIEGO) and ANR Projects Melody (ANR-19-CE46-0011) and OceaniX (ANR-19-CHIA-0016). It benefited from HPC and GPU resources from Azure (Microsoft Azure grant) and from GENCI-IDRIS (Grant 2021-101030)

uncertainties. Both categories of approaches state the targeted issue as the simulation of a time sequence of positions $\vec{r}(\vec{r}_0, t)$ according to a discrete-time generalization of eq.(1):

$$\vec{r}(t + \delta) = \mathcal{F}(\vec{r}(t), \mathbf{z}_t, t) \quad (2)$$

with δ the simulation time step, \mathcal{F} some deterministic or stochastic operator to be trained and \mathbf{z}_t conditioning variables. In general $\mathbf{z}_t = \vec{u}(\vec{r}(t), t)$. This representation suffers from the same limitation as eq.(1) in terms of computational complexity and error propagation.

Interestingly, we can also derive an Eulerian formulation of Lagrangian dynamics. Through Fokker-Planck formalism [23], we can generalize eq.(1) to the time propagation of the pdf of the moving particle as follows:

$$\frac{\partial}{\partial t} p_{\vec{r}}(x, t) = - \frac{\partial}{\partial x} [\mu(x, t) p_{\vec{r}}(x, t)] \quad (3)$$

where x represents spatial position in the Eulerian framework, t the time, $p_{\vec{r}}(x, t)$ the probability density function of the position \vec{r} of the moving particle and $\mu(x, t)$ a drift field. In our case, this drift term relates to the underlying velocity field $\vec{u}(x, t)$. Solving eq.(3) relies on a time integration scheme from some initial pdf $p_{\vec{r}}(x, t = 0) = p_{\vec{r}_0}(x, 0)$. Computationally speaking, this Eulerian representation is no more state-dependent as in eq.(1) but a partial differential equation (PDE). This is particularly appealing from a learning point of view as numerous studies have stressed the relevance of convolutional architectures as PDE solvers [24].

Here, we draw on this Eulerian Fokker-Planck representation of Lagrangian dynamics to explore Eulerian convolutional neural architectures for Lagrangian drift simulation.

3. PROPOSED METHOD

This section describes the proposed deep learning approach for Lagrangian drift simulation, referred to as DriftNet. We first introduce our neural architecture before the learning setup and the neural inversion approach.

3.1. DriftNet architecture

Formally, let us introduce \mathcal{D} the spatial domain of interest and $\mathbf{u} = \{\mathbf{u}_{t_0}, \mathbf{u}_{t_0+\Delta}, \dots, \mathbf{u}_{t_0+K\Delta}\}$ a sequence of velocity fields over domain \mathcal{D} from time t_0 to $t_0 + K\Delta$ with time resolution Δ and number of time steps K . Similarly, we denote by $\vec{r} = \{\vec{r}_{t_0}, \vec{r}_{t_0+\Delta}, \dots, \vec{r}_{t_0+K\Delta}\}$ a Lagrangian drift given as a time series of locations in \mathcal{D} from t_0 to $t_0 + K\Delta$.

Inspired by Fokker-Planck representation eq.(3), we consider the following latent representation for a Lagrangian drift \vec{r} :

$$\begin{cases} \mathbf{y} &= \mathcal{E}(\mathbf{u}, \mathbf{y}_0) \\ \vec{r} &= \mathcal{M}(\mathbf{y}) \end{cases} \quad (4)$$

where \mathbf{y} is a space-time-explicit latent embedding of \vec{r} , and \mathbf{y}_0 some initial latent embedding to encode the known initial position \vec{r}_{t_0} . Eulerian neural operator \mathcal{E} computes latent embedding \mathbf{y} given velocity conditions \mathbf{u} and initial representation \mathbf{y}_0 . Neural operator \mathcal{M} maps latent representation \mathbf{y} to the targeted Lagrangian drift \vec{r} . As such, it aims at mapping a multi-dimensional tensor to a time series of locations in \mathcal{D} .

Regarding the parameterization of operator \mathcal{E} , we do not constrain the latent representation to be univariate similarly to a pdf

in Fokker-Planck formulation. We explore the computational efficiency of deep learning schemes for higher-dimensional latent representations. Overall, operator \mathcal{E} is composed of four elementary units: a series of 3D convolution layers (with the number of channels evolving as follows: $2 \rightarrow 11 \rightarrow 16$) with a non-linear LeakyReLU activation, a 2D convolutional LSTM block and the last series of 3D convolution layers (number of channels evolving as follows: $16 \rightarrow 8$) with a non-linear LeakyReLU activation. Mapping operator \mathcal{M} then combines a *Softmax* layer and a last block of 1D convolutional layers (number of channels evolving as follows: $8 \rightarrow 2$) to transform latent variable \mathbf{y} into a set of positions \vec{r} .

The initial latent representation \mathbf{y}_0 provided to operator \mathcal{E} encodes the initial position of the particle. For time t_0 , we consider a 2D map where the value at each point of the reference domain \mathcal{D} is given by a normalized version of the distance to the initial position of the particle \vec{r}_{t_0} . For the next time steps the encoded normalized distances from the initial position to each grid of the map are attenuated by the dispersion factor computed for the area of study (with regards to the maximum speed in the zone).

Through the different convolutional blocks and the convolutional LSTM unit, we expect the resulting architecture to capture relevant information at different space-time scales, and not only in a point-wise manner, as would be the case for the straightforward implementation of Fokker-Planck representation eq.(3).

3.2. Learning setting

We consider a supervised training of the proposed neural architecture according to the following two losses:

1. the Mean Square Error (MSE) between the reference and simulated trajectories for the considered K -step simulation of N_T particles

$$\mathcal{L}_{MSE} = \frac{1}{N_T K} \sum_{i=0}^{N_T} \sum_{t=0}^K (\vec{r}_R(\vec{r}_{0,i}, t) - \vec{r}_S(\vec{r}_{0,i}, t))^2 \quad (5)$$

2. Liu index between the reference and simulated trajectories [25]

$$\mathcal{L}_{Liu} = \frac{1}{N_T} \sum_{i=1}^{N_{total}} \frac{\sum_{j=1}^K d_{i_j}}{\sum_{j=1}^K l_{i_j}} \quad (6)$$

where d_{i_j} is the Euclidean distance between the reference and the simulated trajectories number i at time step j and l_{i_j} is the length of reference trajectory i between the initial position and the position at time step j .

Overall, the training loss \mathcal{L} is a weighted sum of these losses: $\mathcal{L} = \alpha \cdot \mathcal{L}_{MSE} + \beta \cdot \mathcal{L}_{Liu}$. From cross-validation experiments, we set α and β to 0.2 and 0.8. Using Pytorch¹, our learning setup relies on Adam optimizer with a learning rate of $5e-3$ over 100 epochs.

3.3. Neural inversion of sea surface currents

Contrary to previous works, the proposed neural architecture for Lagrangian drift simulation does not rely on a point-wise relationship but may exploit space-time features of interest in the conditioning velocity fields. Here, we explore how a trained DriftNet can provide new means to diagnose model-driven velocities w.r.t. real Lagrangian trajectories.

¹The open source code ofis available at <https://github.com/CIA-Oceanix/DriftNet>

Let us consider a trained DriftNet with operators $\{\mathcal{E}^*, \mathcal{M}^*\}$ and a real Lagrangian trajectory \vec{r}^R . We aim at identifying the correction $\hat{d}\mathbf{u}$ to the conditioning velocity fields \mathbf{u} such that DriftNet trajectory simulated with velocity condition $\mathbf{u} + \hat{d}\mathbf{u}$ and initial position $\vec{r}_{t_0}^R$ best matches trajectory \vec{r}^R . This leads to the following minimization problem:

$$\hat{d}\mathbf{u} = \arg \min_{d\mathbf{u}} \|\vec{r}^R - \mathcal{M}^*(\mathcal{E}^*(\mathbf{u} + d\mathbf{u}, \mathbf{y}_0))\|^2 \quad (7)$$

Thanks to the automatic differentiation natively embedded in deep learning framework, we directly solve this minimization using a fixed-step gradient descent, typically over 200 gradient steps with stepsize of $5e-2$.

4. RESULTS AND DISCUSSION

In this section we introduce our operational oceanography case study. We present DriftNet results for Lagrangian drift simulation at sea surface and compare them to those of state-of-the-art models. We also report neural inversion examples for the retrieval of corrected sea surface velocities.

4.1. Case study and Datasets

The focus region for our Lagrangian drift simulations is the California Current System (CCS) within the North East Pacific (20°N to 60°N and 160°E to 110°E, see Fig.1). This zone is a major eastern boundary upwelling system wherein cold nutrient-rich water upwelled along the coast fuel one of the most productive marine ecosystem. At the surface it is crossed by the equatorward-flowing California Current (CC) and is a part of the anticyclonic North Pacific gyre [26]. It is characterised by intense mesoscale activity with long-lived eddies generated along the coast travelling offshore westward. The strong mesoscale and submesoscale dynamics of the region together with the presence of a large number of drifters [27] make it particularly well suited for our study.

Two datasets are used: Global Ocean General Circulation Model reanalysis product GLORYS12V1 and Drifters trajectories records from the Copernicus Marine Services dataset. GLORYS12V1² is a reanalysis product of the E.U. Copernicus Marine Service with eddy-resolving resolution of $1/12^\circ$ and 50 vertical layers [28] from 1993 to present. We focus here on GLORYS12V1 sea surface velocities.

Satellite-tracked surface drifting buoys, referred to as drifters, data is collected from the Surface Drifter Data Assembly Centre (NOAA AOML), participating in Global Drifter Program (GDP) and is distributed by Copernicus E.U. Marine Service³. Drifters trajectories records provide sea surface currents observations gridded at a 6-hour resolution, from 1990 to present for the entire globe [27], [29]. The mean trajectory length of drifters deployed in the zone of study is 280 days. As the focus here is set on short-time series, trajectories have been divided into 9-day segments. Overall, our dataset comprises a total of 12.436 9-day trajectories segments with a 6-hour resolution from 1993 until 2020.

We generate a synthetic dataset of Lagrangian drift trajectories according to GLORYS12V1 sea surface velocity fields using as initial positions those of the 9-day trajectories of real drifters. This dataset generated with Ocean Parcels tool [8] leads to a dataset of 12.436 9-day simulated trajectories with a 6-hour resolution. We

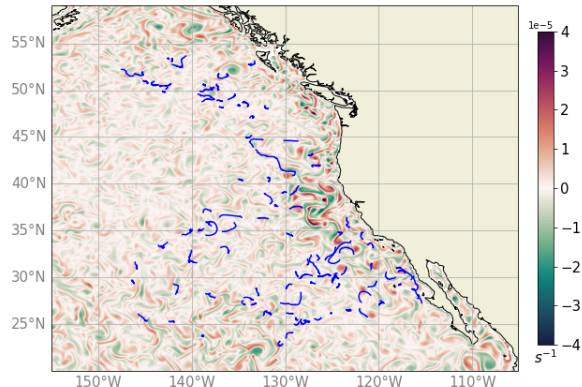


Fig. 1. Snapshot of 9-days GDP drifters trajectories deployed close in time (blue) superimposed on relative vorticity of Glory12v1 velocity field for the case-study area on 20/10/1993.

use this dataset to assess the proposed deep learning scheme according to the following train-validation-test split: we randomly select 80% of the trajectories as training data, and the remaining 20% are equally split into the validation and test datasets.

4.2. Benchmarking experiments

Our numerical experiments benchmark DriftNet (amounting to 2.230.491 trainable parameters) with the following state-of-the-art neural architectures:

- a LSTM architecture with a one-dimensional latent space, inspired from [15] (1.185.448 trainable parameters);
- a 2D Convolutional LSTM architecture with an additional ConvTranspose layer to map the 2D LSTM latent space to a sequence of positions (980.410 trainable parameters);
- a CNN architecture inspired from [11], which combines 3D convolutional blocks and ConvTranspose layers to map the embedded vector to the targeted temporal resolution (2.069.718 trainable parameters).

As evaluation metrics, we consider the Liu index introduced in eq.(6), and the RMSE (root mean square error), defined as the root of the MSE defined in eq.(5), between the reference and the simulated positions ($\gamma(\vec{r}_t)$) and Lagrangian time scales ($\gamma(\mathcal{T})$) [30]. The Lagrangian time scale, $\mathcal{T} = \int R_{\vec{v}}(\tau) d\tau$, with $R_{\vec{v}}(\tau)$ the autocorrelation function, is a measure of the time during which the velocities are correlated with themselves .

Figure 2 shows separation distance statistics between simulated and reference Lagrangian drifts for the benchmarked models. DriftNet clearly outperforms the other deep learning models. It reduces mean separation distance at 9-th day by 66% compared to LSTM-1D, 5% compared to LSTM-2D model and by 65% compared to CNN-2D (see Table 1). Importantly, the simulation performance is of the same order as the one corresponding to Ocean Parcels with a initial position uncertainty of $1/12^\circ$ (Fig.2b). This is a direct consequence of the chaos in turbulent flows, which is widely documented [20]. Similar conclusions can be drawn for Liu index, which provides an assessment of trajectory geometry similarities. DriftNet outperforms the second best model by 23%.

We analyse further the relevance of DriftNet simulations in terms of auto-correlation patterns (Fig.2c-d). We report a very

²<https://doi.org/10.48670/moi-00021>

³<https://doi.org/10.17882/86236>

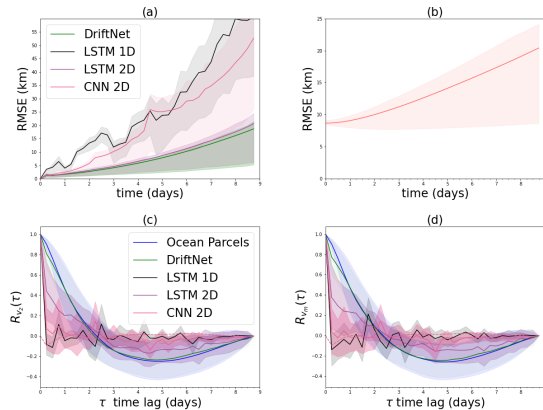


Fig. 2. (a) Mean separation distance in km between trajectories simulated with Ocean Parcels, DriftNet, LSTM and CNN models as a function of time in days. (b) Distance in km between trajectories seeded in drifters launching positions and trajectories seeded in the $1/12^\circ$ radius around them, both simulated with Ocean Parcels. The shaded area correspond to the distances comprised between the first and the third quantile. (c), (d) Autocorrelation for zonal (left) and meridional (right) velocities as a function of time lag (τ) for different simulation methods and Ocean Parcels as reference.

good match between DriftNet simulations and reference Lagrangian drifts. These results are in line with the good performance in terms of Lagrangian time scales discussed above.

| Model | $\gamma(\mathcal{T})$, days | Liu | $\gamma(\vec{r}_{t=9})$, km |
|----------|------------------------------|------|------------------------------|
| LSTM 1D | 3.74 | 0.8 | 59.2 |
| LSTM 2D | 1.78 | 0.26 | 20.7 |
| CNN 2D | 12 | 0.64 | 52 |
| DriftNet | 0.7 | 0.2 | 19.6 |

Table 1. Metrics comparison between DriftNet and state-of-the-art methods. The used metrics are the Liu index, the RMSE between positions $\gamma(\vec{r}_{t=9})$, and Lagrangian time scales $\gamma(\mathcal{T})$. Each metric is averaged over the trajectories of the test dataset.

4.3. Inversion examples

We report inversion examples using the approach described in Section 3.3. As illustrated in Fig.3, we observe a clear mismatch between the real drifter trajectory and the one simulated from GLORYS12V1 sea surface velocities. It reveals that GLORYS12V1 product cannot recover all fine-scale patterns from the assimilation of available satellite-derived observation datasets [28].

As reported for the two examples in Fig.3, DriftNet-based neural inversion identifies velocity anomalies such that the simulated trajectories match closely the real ones. By contrast, this inversion scheme does not apply relevantly when considering the other neural architectures. DriftNet-based velocity anomalies exhibit fine-scale vorticity patterns. Such structures are too small to be well resolved by reanalysis products which do not capture the small meso and submesoscale [28]. Equally, limitations also exist in resolving the upper mesoscale

with for instance eddy features potentially poorly positioned, misdimensioned, or incoherent over time, all of which having a direct impact on the Lagrangian transport [5].

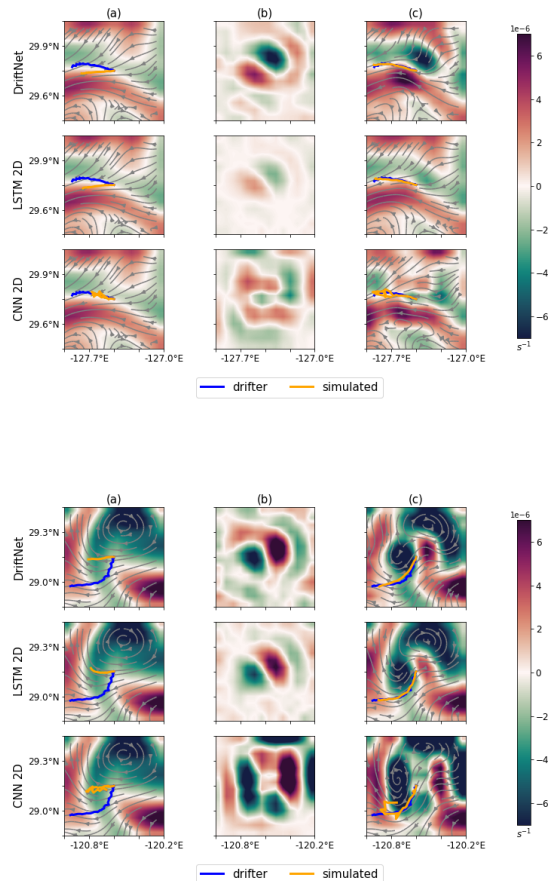


Fig. 3. Neural inversion examples for velocity anomaly fields from real drifter trajectories: vorticity of the velocity field at time t_0 with the reference (blue) and simulated (orange) drift (a), vorticity of the estimated velocity anomaly field at time t_0 (b) and vorticity of the corrected velocity field at time t_0 with reference (blue) and simulated (orange) drift. The first three rows relate to the first example using different deep learning architectures in inversion scheme eq.(7), and the last three to the second example.

5. CONCLUSIONS

In this study we proposed a novel method for Lagrangian drift simulation and analysis at sea surface based on a Deep Learning scheme. It relates to Eulerian Fokker-Planck interpretation of Lagrangian dynamics. We introduce a fully-convolutional architecture and an Eulerian latent representation of Lagrangian trajectories. The resulting Lagrangian drift simulations outperform previously explored state-of-the-art methods for trajectory data. We also point out its potential to diagnose sea surface velocities derived by operational models w.r.t. real drifter trajectories.

We believe this study to open new research avenues for the development of deep learning schemes for Lagrangian drift simulation. From a methodological point, the extension of the proposed

architecture to a conditional GAN setting [31] naturally arises to account the variability of real drift trajectories. Regarding operational oceanography, we may envision extensions of the proposed scheme with satellite-derived observations and other model-driven variables, including among others sea surface temperature fields, satellite altimetry data, wave models' outputs... Our neural scheme may also provide new means in movement ecology to study the interactions between animal movement trajectories and their environment [32].

6. REFERENCES

- [1] J. Röhrs, G. Sutherland, G. Jeans, M. Bedington, A. K. Sperrevik, K.-F. Dagestad, Y. Gusdal, C. Mauritzen, A. Dale, and J. H LaCasce, "Surface currents in operational oceanography: Key applications, mechanisms, and methods," *Journal of Operational Oceanography*, 2021.
- [2] N. Maximenko, J. Hafner, and P. Niiler, "Pathways of marine debris derived from trajectories of lagrangian drifters," *Marine pollution bulletin*, vol. 65, 2012.
- [3] Y. B. Son, B.-J. Choi, Y. H. Kim, and Y.-G. Park, "Tracing floating green algae blooms in the yellow sea and the east china sea using goci satellite data and lagrangian transport simulations," *Remote Sensing of Environment*, vol. 156, 2015.
- [4] Ø. Breivik, A. A. Allen, C. Maisondieu, and M. Olagnon, "Advances in search and rescue at sea," *Ocean Dynamics*, vol. 63, pp. 83–88, 2013.
- [5] C. N Barron, L. F Smedstad, J. M Dastugue, and O. M. Smedstad, "Evaluation of ocean models using observed and simulated drifter trajectories: Impact of sea surface height on synthetic profiles for data assimilation," *Journal of Geophysical Research: Oceans*, vol. 112, 2007.
- [6] S. Liubartseva, G. Coppini, R. Lecci, and E. Clementi, "Tracking plastics in the mediterranean: 2d lagrangian model," *Marine pollution bulletin*, vol. 129, 2018.
- [7] E. Zambianchi, M. Trani, and P. Falco, "Lagrangian transport of marine litter in the mediterranean sea," *Frontiers in Environmental Science*, vol. 5, 2017.
- [8] M. Lange and E. van Sebille, "Parcels v0. 9: prototyping a lagrangian ocean analysis framework for the petascale age," *Geoscientific Model Development*, vol. 10, 2017.
- [9] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine learning*, vol. 32, 1998.
- [10] A. Roy, R. Fablet, and S. L. Bertrand, "Using generative adversarial networks (gan) to simulate central-place foraging trajectories," *Methods in Ecology and Evolution*, 2022.
- [11] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," vol. 33, 2019.
- [12] J. Gan, P. Liu, and R. K Chakrabarty, "Deep learning enabled lagrangian particle trajectory simulation," *Journal of Aerosol Science*, vol. 139, 2020.
- [13] S. Julka, V. Sowrirajan, J. Schloetterer, and M. Granitzer, "Conditional generative adversarial networks for speed control in trajectory simulation," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2021.
- [14] D. Paz, H. Zhang, and H. I Christensen, "Tridentnet: A conditional generative model for dynamic trajectory generation," in *International Conference on Intelligent Autonomous Systems*. Springer, 2022.
- [15] P. Zegers and M. K Sundareshan, "Trajectory generation and modulation using dynamic neural networks," *IEEE Transactions on Neural Networks*, vol. 14, 2003.
- [16] Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, "Deep learning for remote sensing image classification: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, 2018.
- [17] J. F Quinting and C; M Grams, "Eulerian identification of ascending airstreams (elias 2.0) in numerical weather prediction and climate models—part 1: Development of deep learning model," *Geoscientific Model Development*, vol. 15, 2022.
- [18] D. Nguyen, R. Vadaine, G. Hajduch, R. Garelo, and R. Fablet, "Geotracknet—a maritime anomaly detector using probabilistic neural network representation of ais tracks and a contrario detection," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [19] R. E Davis, "Lagrangian ocean studies," *Annual Review of Fluid Mechanics*, vol. 23, 1991.
- [20] U. Callies, M. Kreuz, W. Petersen, and Y. G Voynova, "On using lagrangian drift simulations to aid interpretation of in situ monitoring data," *Frontiers in Marine Science*, 2021.
- [21] N. O Aksamit, T. Sapsis, and G. Haller, "Machine-learning mesoscale and submesoscale surface dynamics from lagrangian ocean drifter trajectories," *Journal of Physical Oceanography*, vol. 50, 2020.
- [22] W. Zucchini and I. L MacDonald, *Hidden Markov models for time series: an introduction using R*, Chapman and Hall/CRC, 2009.
- [23] A. W Visser, "Lagrangian modelling of plankton motion: From deceptively simple random walks to fokker–planck and back again," *Journal of Marine Systems*, vol. 70, 2008.
- [24] Z. Long, Y. Lu, X. Ma, and B. Dong, "Pde-net: Learning pdes from data," in *International Conference on Machine Learning*. PMLR, 2018.
- [25] Y. Liu and R. H Weisberg, "Evaluation of trajectory modeling in different dynamic regions using normalized cumulative lagrangian separation," *Journal of Geophysical Research: Oceans*, vol. 116, 2011.
- [26] R. J Lynn and J. J Simpson, "The california current system: The seasonal variability of its physical characteristics," *Journal of Geophysical Research: Oceans*, vol. 92, 1987.
- [27] Lumpkin R. et al. Elipot, S., "A global surface drifter data set at hourly resolution," *Journal of Geophysical Research: Oceans*, vol. 121, 2016.
- [28] J.M. Lellouche and E. et al. Greiner, "Recent updates to the copernicus marine service global ocean monitoring and forecasting real-time 1/ 12 high-resolution system," *Ocean Science*, vol. 14, 2018.
- [29] R. et al. Lumpkin, "Evaluating where and why drifters die," *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 2, pp. 300–308, 2012.

- [30] W. Krauß and C. W Böning, “Lagrangian properties of eddy fields in the northern north atlantic as deduced from satellite-tracked buoys,” *Journal of Marine Research*, vol. 45, no. 2, pp. 259–291, 1987.
- [31] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [32] T. Maekawa, K. Ohara, Y. Zhang, M. Fukutomi, S. Matsumoto, K. Matsumura, M. Shidara, S. J Yamazaki, R. Fujisawa, K. Ide, et al., “Deep learning-assisted comparative analysis of animal trajectories with deeppl,” *Nature communications*, vol. 11, no. 1, 2020.