MISSIONE 4
**ISTRUZIONE RICERCA**

ITINERIS

# D5.17 Data-driven reconstruction of the ocean state - Algorithm Theoretical Background Document (ATBD)

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

| | |
|---|---|
| **Deliverable number:** | D5.17 |
| **Work package:** | WP5 – Marine Domain<br>WP5.21 Prototype models for data-driven reconstruction of the ocean state |
| **Intermediate Objective:** | IO5.7 |
| **Deliverable type:** | X Document, report |
| | ☐ Websites, patent filings, videos, etc. |
| | ☐ Other: please specify ..... |
| **Dissemination level:** | X Public |
| | ☐ Restricted |
| **Estimated delivery (bimester):** | 13 |
| **Actual delivery date:** | 05/12/2024 |
| **Author(s) (Partner-OU):** | Bruno Buongiorno Nardelli, Daniele Ciani, Lorenzo della Cioppa, Claudia Fanelli, Michela Sammartino (CNR-ISMAR-NA) |
| **Reviewed by:** | Rosalia Santoleri & ITINERIS Executive Board |
| **Note:** | |

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System - CUP B53C22002150006 (D.D. n. 130/2022)
Funded by EU - Next Generation EU
Mission 4 “Education and Research” - Component 2: “From research to business” -
Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

# Table of contents



# Index of tables

# 1 RECONSTRUCTION OF VERTICAL PROFILES FROM SURFACE DATA BASED ON A KOLMOGOROV-ARNOLD NETWORK

Satellites allow for quasi-synoptic observation of several surface ocean EOVs/ECVs. However, satellite sensors are unable to observe beyond the surface layer, rendering in situ measurements still crucial for oceanographic investigations. In order to fill the gaps between satellite observations and in situ data and obtain synoptic 3D reconstructions, data-driven interpolation/extrapolation methods are required. Recently, the use of neural networks has been proposed to predict both Temperature and Chlorophyll [1], proving very effective at the task.

In this activity, we develop a novel data-driven neural network model for the 3D reconstructions of ocean state at basin scale. The proposed network architecture is based on the innovative, recently proposed, Kolmogorov-Arnold Network (KAN), which has been re-implemented and adapted to our goal. The proposed model aims at reconstructing Salinity, Temperature, Density and Chlorophyll at several depths along the water column from observed satellite surface data.

The proposed model is suitable for the integration with observations from different in situ platforms, such as argo floats, gliders, CTD profilers, fixed platforms, etc.

## 1.1 Input data selection and pre-processing

To build the reference Mediterranean database used for training and testing the Artificial Intelligence algorithm, surface and subsurface measurements of the chosen input co-predictors have been extracted from satellite regional products and concurrent in situ datasets respectively.

### *1.1.1 Surface data*

- *Satellite Absolute Dynamic Topography dataset*

The Absolute Dynamic Topography (ADT) and absolute geostrophic currents (zonal and meridian components, U & V respectively) have been downloaded from the Copernicus Marine Service web-portal. These are additional variables included in the altimeter satellite gridded data of Sea Level Anomalies (SLA) produced by the production unit of SL-CLS of Toulouse in France. The ADT is the instantaneous height above the Geoid and it is computed as the sum of the $SLA_N$ and $MDT_N$ (Mean Dynamic Topography, [2], where N, here, is the mean reference period (1993–2012). The ADT is a delayed time optimally interpolated product at a resolution of 0.125◦ × 0.125◦ processed by the DUACS multimission altimeter data processing system and coming from the merging of different altimeter missions (for more details see the Quality Information Document at https://doi.org/10.48670/moi-00141 or https://duacs.cls.fr). The two components u and v of absolute geostrophic currents are then obtained by the SLA and ADT product, above mentioned. In the present work, the surface ADT, U and V values were extracted once matched-up with profile's position and then used as input variables for the 3D neural network reconstruction.

### *1.1.2 In situ profiles*

The in situ data were derived from two primary sources: oceanographic cruise data and bio-hydrographic profiles collected by Biogeochemical-Argo floats (Fig. 1).

- *Oceanographic cruise database*

The in situ oceanographic cruise database comprises concurrent profiles of temperature, chlorophyll, and salinity collected during 26 oceanographic cruises conducted in the Mediterranean Sea between 1997 and 2017 (represented by red crosses in Fig. 1). Many of these

data were acquired and processed by the GOS group (Global Ocean Satellite monitoring and marine ecosystem studies) at the Institute of Marine Sciences (ISMAR) in Rome, part of the Italian National Research Council. Chlorophyll estimates were obtained by calibrating the fluorometer signals with bottle samples collected simultaneously at each station during each cruise.

- *Bio-Argo database*

The second in situ database consists of hydrographic (temperature and salinity) and bio-geochemical (chlorophyll-a) profiles collected by Biogeochemical-Argo (BGC-Argo) floats across the Mediterranean Sea between 2012 and 2022 (blue crosses in Fig. 1). These data have been collected and made freely available by the International Argo Program and its contributing national programs (http://www.argo.ucsd.edu, http://argo.jcommops.org), which form part of the Global Ocean Observing System.
For this work, both real-time and delayed-mode data were included for temperature and salinity profiles, while for chlorophyll-a, only delayed-mode data (CHLA_ADJUSTED) were used. All profiles were downloaded through the Coriolis FTP server (ftp.ifremer.fr).

In the Argo program, real-time profile files are available within 12–24 hours after the float completes a profile. These files undergo automatic quality control tests to detect significant errors, and any scientifically calibrated parameters are recorded as <PARAM>_ADJUSTED. These real-time data are generally consistent with ocean climatologies, as verified through monthly climatology tests and visual comparisons for profiles that fail the automated tests.

The Delayed-mode files for hydrographic profiles are typically available 1–2 years after data collection, though sometimes earlier. These profiles undergo thorough validation by oceanographic experts. Adjusted salinity values are derived by comparing float data to high-quality ship-based CTD measurements and Argo climatologies, following the procedures described in [3-6].

For biogeochemical parameters, delayed-mode files may become available within 5–6 float cycles after deployment. This speeded availability allows for early adjustment of data to address sensor calibration issues, significantly improving data accuracy. Additional delayed-mode quality controls are applied once a longer time series of float data is available.

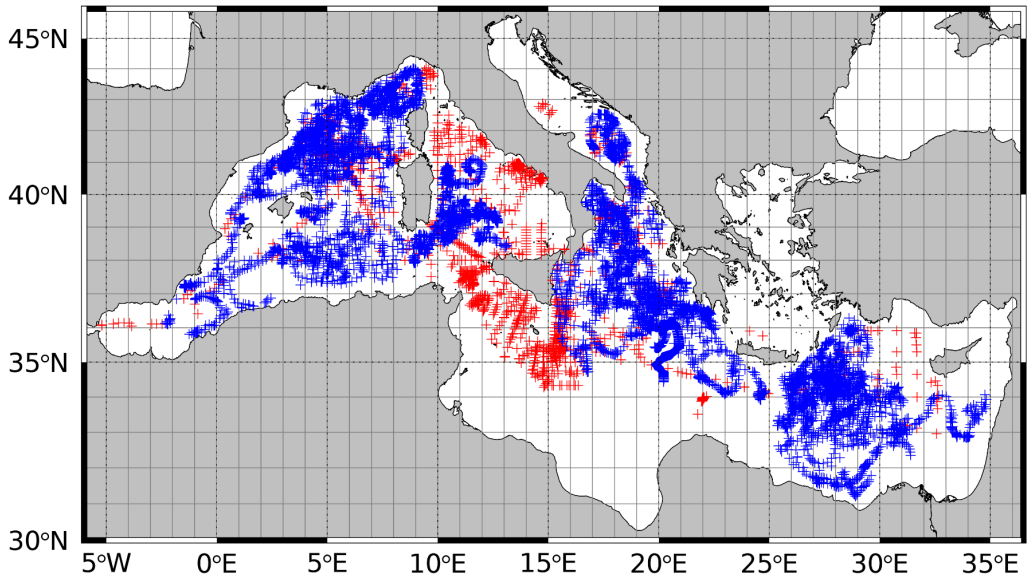


**Figure 1.** Map of in situ database comprising oceanographic in situ (red crosses) and Bio-Argo (blue crosses) stations.

### *1.1.3 Input data preprocessing*

All in situ data from oceanographic cruises underwent quality controls based on the SeaDataNet standards (refer to the quality control manual at https://www.seadatanet.org/Standards/DataQuality-Control) along with additional criteria specifically designed for the objectives of this work. The same quality control methodology detailed in [1] was applied to both in situ chlorophyll and temperature/salinity profiles, comprising the following steps:

- **Visual Inspection**: Ensured the internal consistency of the dataset.
- **Verification of Initial Acquisition Depth**: Profiles were accepted only if the first recorded depth was between 3 and 4 m to minimize noise in the initial measurement.
- **Evaluation of Missing Data Points**: Profiles with excessive missing data were discarded. In other cases, missing points were filled via linear interpolation.
- **Maximum Depth Criterion**: Profiles were included only if their acquisition depth reached at least 150 m.

For the Bio-Argo dataset, in addition to the quality controls and post-processing conducted by Argo providers (http://www.argodatamgt.org/Documentation), further checks were implemented. Profiles with a depth of at least 150 m were selected and interpolated onto a regular vertical grid with 1 m spacing.

From both datasets (oceanographic and Bio-Argo), the optical weighted pigment (OWP) concentration was derived from the chlorophyll profiles using a modified version of the Morel [7] model. This step allowed us to obtain from in situ data a chlorophyll concentration closer to that represented by satellite-observed signal. It is usually approximated to the chlorophyll-a concentration integrated over the upper portion of the water column, roughly corresponding to 1/5 of the euphotic depth (depth where light levels are reduced to 1% of surface irradiance).

The final in situ database was then matched with satellite altimeter-derived sea surface height (ADT) and the two components of surface absolute geostrophic current products.

Given the absence of ultra-high resolution sea surface temperature and sea surface salinity/density data further back than 2008, the surface values of in situ temperature (T) and salinity (S) profiles were used as proxies for satellite-derived sea surface temperature (SST) and sea surface salinity (SSS), respectively, while OWP was used as a proxy for satellite chlorophyll (CHL). Consequently, density profiles were computed from in situ T and S data through the equation state, and their surface values served as proxies for satellite-derived sea surface density (SSD).

To enhance data quality, profiles showing constant chlorophyll values within the first 15 meters of depth were excluded.

After completing the quality control process, a total of 8624 profiles were retained. These data were randomly split into training (80%) and testing (20%) subsets, with the latter serving as an independent dataset for evaluating the network's performance. It has to be noted that this database, originally developed within the ESA 4DMED-Sea project, is used here as the basis for implementing the AI algorithm detailed in the following section.

All profiles were scaled within the 0–1 interval before feeding the network by using the following formula:

$$x_{norm} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$$

## 1.2 Description of the algorithm

The model consists of a neural network which takes eleven variables as predictors and four variables as targets. The input variables and output variables are detailed in Table 1.

**Table 1.** Input and output variables

| **Inputs** |
|---|
| JD1 (Julian Day Embedding 1) |
| JD2 (Julian Day Embedding 2) |
| Lat (Latitude) |
| Lon (Longitude) |
| SST/T(0) (Sea Surface Temperature) |
| SSS/S(0) (Sea Surface Salinity) |
| LOWP (Log10 of Optical Weighted Pigment) |
| ADT (Absolute Dynamic Topography) |
| Zonal Velocity (U) |
| Meridional Velocity (V) |

| **Outputs** |
|---|
| T (Temperature) |
| S (Salinity) |
| LChl (Log10 of Chl) |
| D (Density) |

| SSD/D(0) (Sea Surface Density) |
|---|

The Julian day JD is embedded in two variables $JD_1 = cos(2\pi JD/365)$ and $JD_2 = sin(2\pi JD/365)$ to account for seasonal periodicity. All of the variables are normalized to be in the range [0, 1].

The network is based on the recently proposed Kolmogorov-Arnold Network (KAN) [8]. A classic Multi-Layer Perceptron (MLP) is a neural network consisting of one input layer, one output layer and one or more hidden layers. Each of the hidden layers multiplies the input vector by a matrix of weights and then processes each element of the resulting vector through a scalar nonlinear function. MLPs are then trained using gradient-descent-based optimizers to minimize an aptly defined loss function. The relation between input vector $X = (x_1, x_2, \ldots x_N)$ and output vector $Y = (y_1, y_2, \ldots y_M)$ of a single MLP layer is given by $Y = \sigma(WX)$, where W is the weight matrix and $\sigma$ is the nonlinear function.

Kolmogorov Arnold Networks are inspired by Kolmogorov-Arnold Theorem, which states that every multivariate function can be written as a sum of univariate functions. KANs substitute the weight matrix multiplication and successive nonlinearization with spline evaluation (see Fig. 2). Each element of the input vector is fed into B-spline functions of order $N_o$ on a predefined grid. The number of spline functions is determined by the user. For a single KAN layer the functional relation between input vector $X = (x_1, x_2, \ldots, x_p, \ldots, x_N)$ and output vector $Y = (y_1, y_2, \ldots, y_q, \ldots, y_M)$ is given by

$$y_q = \sum_{p=1}^{N} \phi_{q,p}(x_p) \qquad q = 1, 2, \ldots, M$$

where $\phi_{q,p}(x) = \sum_{k=1}^{K} c_k{}^{q,p} B_k(x)$. The $B_k$'s are B-spline basis functions and the $c_k{}^{q,p}$'s are the trainable B-spline coefficients (see Fig. 3). In Einstein summation notation

$$y_q = c_k{}^{q,p} B_k(x_p). \qquad (*)$$

The number K of spline functions is determined by the number of discretization points of the grid where the spline functions are defined. This parameter can be set by the user. In the implementation, an additional residual nonlinearity is added to the B-spline in the form of a silu function ( $silu(x) = x/(1 + e^{-x})$), weighted by a trainable parameter. All the weights are initialized randomly, drawing from either an uniform distribution or a normal distribution.

As in the original paper, a grid refinement procedure has been implemented here. This procedure updates the grid at predefined time steps during the training, increasing the number of grid points and adapting the grid on the input. As a result, the K dimension of the tensor $c_k{}^{q,p}$ changes and new parameters $\underline{c}_k{}^{q,p}$ are computed by least squares error minimization (see Fig.3).

Several KAN layers are composed to form a deep KAN. The number of outputs/inputs of a KAN layer are the equivalent of neurons in MLPs.

The original formulation of KANs included both a pruning step and a regularisation term based on the spline outputs rather than on weights values, both proposed to increase sparseness in the B-spline coefficient tensor $c_k{}^{q,p}$ . They are not included in our implementation as we found that, for our application, they make the training more difficult without offering any advantage.

In the original paper the KAN is trained using second order quasi-Newton L-BFGS algorithm [9] while in our implementation we use SGD-based AdamW optimizer [10].

The code has been implemented in python, using the machine learning package Tensorflow. Tensorflow provides both functional and traditional API that allow for easy definition of customized networks. However, due to the specificness of the KAN layer, a custom layer has been implemented. The KAN layer is defined by the number of inputs, the number of outputs, the order of the splines, the grid range and the grid discretization size. Additional parameters can be set: a scale factor for random parameter initialization, the residual activation function, and a coefficient $\varepsilon$, which weights between uniform and adaptive grids during grid updates.

The KAN layer takes as input a tensor of dimensions [Batch, Inputs] and outputs a tensor of size [Batch, Outputs]. The spline bases values $B_k(x_p)$ are computed using De Boor's algorithm [11], which allows for parallel computation of the basis values on the whole batch. Then the bases values are right-multiplied by the coefficient tensor ${c_k}^{q,p}$, as per eq. (*). The tensor multiplication is computed in parallel across the batch.

The Tensorflow automatic differentiation system automatically manages to compute the gradient of the network through backpropagation [12].

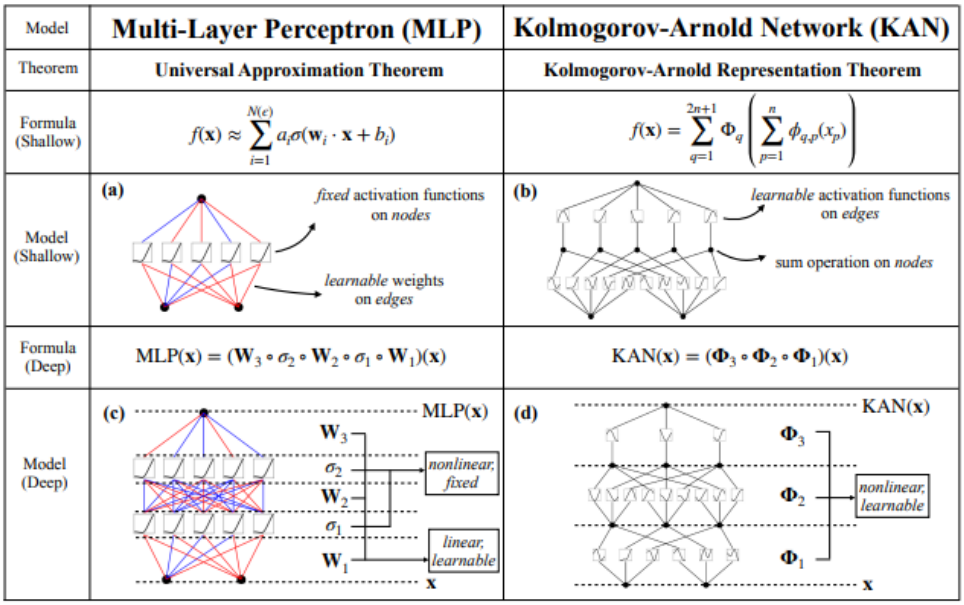


**Figure 2**. Comparison between MLP and KAN [8].

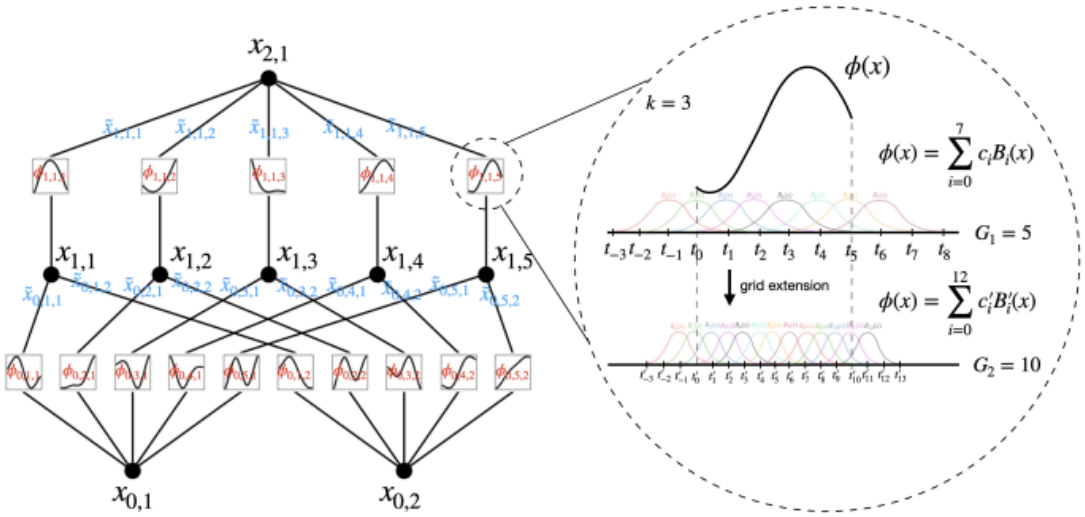


**Figure 3.** Computation of the splines in the KAN nodes, with grid update detailed [8].

# 2 IMPROVEMENT OF ABSOLUTE DYNAMIC TOPOGRAPHY EFFECTIVE RESOLUTION WITH A CONVOLUTIONAL AUTOENCODER

Our study aims to enhance the mapping of Absolute Dynamic Topography (ADT) and Sea Surface Temperature (SST) using satellite observations. Obtaining consistent, high-resolution ADT and SST data from space is challenging due to limitations of satellite instruments, sampling constraints, and inaccuracies introduced by interpolation algorithms that create gap-free (Level 4 - L4) analyses. To tackle these challenges, we developed and tested a deep learning approach, based on Convolutional Neural Network (CNN) models originally designed for single-image super-resolution.

Building on recent research carried out in the framework of the European Space Agency World Ocean Circulation project, we conducted an Observing System Simulation Experiment (OSSE) using outputs from the Copernicus numerical model. We tested a new network architecture and also devised a strategy for further refinement with respect to earlier OSSEs. In fact, previous tests utilized low-resolution L4 satellite-equivalent ADTs combined with high-resolution "perfectly known" SSTs to extract high-resolution sea surface dynamics. In this study, we introduced realistic L4 SST processing errors and adapted CNN to simultaneously predict high-resolution SST and ADT from synthetic satellite-equivalent L4 datasets. This approach enables us to evaluate potential improvements in ADT and SST mapping while incorporating dynamical constraints through physics-informed loss functions.

## 2.1 Input data description

The network is trained using OSSE data and then applied to satellite-derived ADTs and SSTs from the Copernicus Marine Service. The primary objective is to reconstruct super-resolved ADTs and geostrophic currents. This investigation focuses on the Mediterranean Sea, a region characterized by its challenging small Rossby deformation radius (~10 km), during the year 2017. We selected for each dataset 40 dates distributed along the year in order to cover all dynamical regimes and used as independent test dataset. The remaining data has been processed as follows: we performed a resampling into 76 ×100 pixels (corresponding to ≃ 300 ×400 km) tiles. The tiles extraction also

considered a 50% spatial overlap, resulting in a total of 42250 samples. All the samples then undergo a normalization procedure with respect to the maximum observed in the time series. In addition, for the satellite equivalent data, we compute anomalies filtering out signals larger than approximately 200 km. This operation is intended to preserve signals associated with local SST variations caused by horizontal current advection. By extracting patterns from the SST field, it contributes to ADT reconstruction, primarily capturing the effects of ocean surface dynamics. These samples were subsequently utilized for the network's training and validation processes, whose relative amount is respectively 85% and 15% of the 42250 (total) samples.

### *2.1.1 Simulated SST/SSH data*

The Mediterranean Forecasting System (MFS) is a hydrodynamic model covering the Mediterranean Basin and the Atlantic Ocean near the Strait of Gibraltar [13]. It provides outputs ranging from monthly to 15-minute intervals for 3D horizontal currents and sea surface height (SSH), as well as monthly to hourly estimates of 3D temperature and salinity fields. These datasets are accessible through the Copernicus Marine Service web portal (Product ID: MEDSEA-ANALYSIS-FORECAST-PHY-006-013). In this study, we used daily outputs of SSH and SST, focusing on the Mediterranean Basin within the coordinates 30° to 46°N and -6° to 37°E. The data are available on a 1/24° regular grid with 125 vertically unequally spaced levels. The simulations are generated using the NEMO (Nucleus for European Modelling of the Ocean) model, coupled with Wave Watch-III to include wave dynamics. Additionally, the MFS incorporates data assimilation, including 2D satellite-derived SST, vertical salinity profiles, and along-track sea-level anomaly observations.

We therefore produced one year (2017) of synthetic Satellite-Equivalent altimeter-derived Absolute Dynamic Topography (referred to as SE-ADT) maps using outputs from the Copernicus Marine Service MFS hydrodynamic simulation and applying the Data Unification and Altimeter Combination System (DUACS) mapping method. To calculate sea level anomaly (SLA) from model outputs, we used the following formula:

$$SLA = SSH - (MDT - 0.344).$$

Here, the mean dynamic topography (MDT) is provided as a static field along with the model outputs. A constant value of 0.344 m was subtracted to adjust SLA values in the Mediterranean Sea, ensuring the spatio-temporal average of SLA is zero for 2017. To reduce large-scale, high-frequency variability typically handled by dynamic atmospheric correction (DAC), we applied a Loess filter to the synthetic data. The SLA was then sampled along the actual paths of a synthetic radar altimeter constellation consisting of four missions: Jason-3, Sentinel-3A, SARAL/Altika, and CryoSat-2. This step utilized the SWOT simulator software, which accounts for the orbits, errors, and noise characteristics of each mission. The selected constellation mirrors the configuration used in Copernicus Marine Service processing during 2017. These along-track synthetic measurements were processed through the DUACS system to create L4 SLA maps. Finally, the optimal interpolation (OI) scheme followed the DUACS DT2018 (Delayed Time) configuration for the Mediterranean region, as detailed in [14]. The reconstructed L4 SLA maps were combined with the filtered large-scale maps to produce Absolute Dynamic Topography (ADT). The resulting data are provided daily on a 1/8° regular grid, with further details available in [15].

### *2.1.2 Satellite-derived SST/SSH data*

Sea surface geostrophic currents were sourced from the Copernicus Marine Service, derived from optimally interpolated absolute dynamic topography data. These data combine observations from a constellation of radar altimeters, comprising four to six instruments during the 2008–2019 period [14]. The geostrophic currents are provided as daily fields with a nominal horizontal resolution of

1/8°. For this study, we extracted the 2008–2019 time series. The associated Copernicus Marine Service product and dataset ID are SEALEVEL_MED_PHY_L4_REP_OBSERVATIONS_008_051/dataset-duacsrep-medsea-merged-allsat-phy-l4, accessed on 1 March 2021. This dataset is now part of the SEALEVEL_EUR_PHY_L4_MY_008_068/cmems_obs-sl_eur_physsh_my_allsat-l4-duacs-0.125deg_P1D product.

We also acquired remotely sensed SST data from the Copernicus Marine Service (https://doi.org/10.48670/moi-00172, accessed on 14 January 2022). These Level 4 products provide gap-free estimates of the foundation temperature (approximately 10 m depth) on a regular grid, produced and distributed operationally in near real-time. For this study, we used 12 years (2008–2019) of the ultra-high spatial resolution (UHR) Mediterranean dataset with a nominal resolution of 1/100° (Product ID: SST-MED-SST-L4-NRT-OBSERVATIONS-010-004-c-V2). This SST product is derived from nighttime satellite infrared sensor images, subjected to rigorous quality control, cloudy pixel removal, and an optimal interpolation algorithm. To ensure consistency with the SST data used to train the neural network model, we evaluated the effective spatial scales represented in the model SST and applied a low-pass Lanczos filter to the UHR data to achieve comparable effective resolutions. The filtered satellite SSTs were then remapped onto the final model grid (1/24° resolution) using bilinear interpolation.

## 2.2 Description of the algorithm

The overall architecture, sketched in Figure 4 and generally known as autoencoder, follows a pipeline structure that can be divided into two parts: an encoder and a decoder. The main idea is that the encoder processes the low resolution input and generates a latent representation of key features by down-sampling the dataset. In the second stage, a decoder learns to generalize these features and generates high-resolution SST image candidates.

In particular, the encoder is designed to extract all relevant features from the input tiles (with dimensions 76×100×6) corresponding to SE-ADT, SE-ADT error, SE-SST, SE-SST error, ∂t(SE-SST) and the ∂t(SE-SST) error. The encoder consists of multiple convolutional layers with a stride of 2, progressively reducing the input dimensions to a 6×9 feature map. The final encoder layer, composed of 256 convolutional filters, is reshaped into a one-dimensional feature vector of size 13824, which serves as input to the decoder. To mitigate overfitting, a 20% dropout is applied to the feature vector. The decoder part mirrors the encoder's architecture. It uses several layers of transposed convolutions to expand the feature vector back to the original missing region's size. All convolutional layers in the network are connected with LeakyReLU activation functions with a slope coefficient of 0.2.

The autoencoder is trained to minimize a physics-informed loss function (LF) which is a weighted combination of several terms and is defined as follows:

$$\mathrm{LF} = \alpha\overline{[(\mathrm{SST}_{\mathrm{pred}} - \mathrm{SST}_{\mathrm{ref}})^2]} + \beta\overline{[(\mathrm{ADT}_{\mathrm{pred}} - \mathrm{ADT}_{\mathrm{ref}})^2]} + \gamma\{\overline{[(\partial_t \mathrm{SST})_{\mathrm{pred}} - (\partial_t \mathrm{SST})_{\mathrm{ref}}]^2}\}$$

where $\alpha = 1, \beta = 0.25, \gamma = 0.67$ have been determined via preliminary CNN training over a few epochs and the subscripts "pred" and "ref" respectively stand for the output of the CNN prediction and the validation dataset used during training.

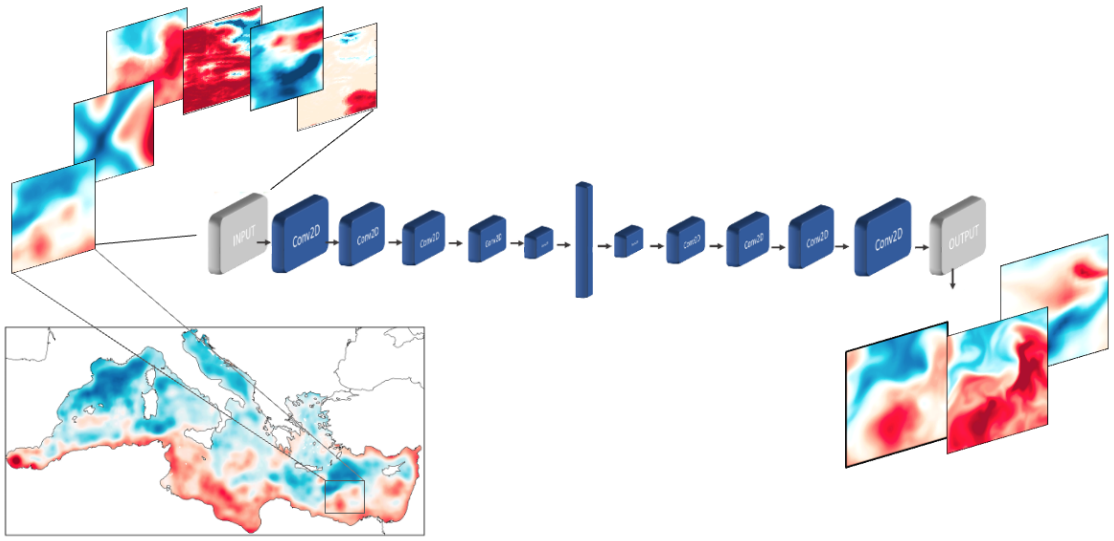


**Figure 4.** Autoencoder architecture. The set of input tiles, from left to right, respectively indicate: SE-ADT, SE-ADT error, SE-SST, SE-SST error, ∂t(SE-SST) and the ∂t(SE-SST) error. The three output tiles, from left to right, respectively indicate the SR-ADT, SR-SST and SR- ∂tSST. Conv2D stands for 2D convolutional filter. The ADT output file is emphasized by a thick edge to indicate the focus of the present study.

# 3 PREDICTION OF OCEANIC LAGRANGIAN TRAJECTORIES WITH HYBRID SPACE-TIME CNN ARCHITECTURE

The Lagrangian approach is a fundamental part of several scientific and operational tasks, such as debris advection forecasting, dynamical studies, validation of models, etc.

As the number of deployed drifters is often insufficient for the time/space sampling needed in the applications, trajectory simulations are of great importance.

The most common method to simulate lagrangian trajectories is to integrate the differential equation

$$\frac{d}{dt}x(t) = U(t, x(t)), \quad x(0) = x_0,$$

where U(t, x(t)) is the Eulerian velocity field.

However, the knowledge of Eulerian velocity fields is far from complete. Although satellites allow for synoptic observation of ocean surface, their coverage is often sparse in both time and space. In addition, current velocities are computed assuming geostrophic balance, which does not always yield accurate results and cannot be used where the Coriolis force is too small (i.e. near the Equator). On the other hand, GCM are capable of producing high resolution Eulerian fields, but their accuracy is limited when compared with real-world ground truth: accuracy of trajectories integrated from synthetic fields is generally lower than of those integrated from geostrophy-derived fields. Ultimately, the choice of the underlying model is determined by which is more suited for the task at hand.

Recently, AI approaches have been proposed in order to overcome these problems. AI approaches leverage on the possibility to easily integrate real world data in the training procedures.

In this activity we develop an innovative data-driven model for the generation of synthetic Lagrangian trajectories. The model we propose is based on a convolutional hybrid U-net/LSTM deep network, and it is inspired by well-established computer vision algorithms for image-to-image translation. The proposed method allows to combine in-situ drifters with data derived from satellite remote sensing in the training process. In this preliminary stage we focus on training on a synthetic dataset using zonal and meridional velocities and temperature as model inputs.

## 3.1 Input data description

The model is designed to take as input the initial point $x_0$and the chosen Eulerian fields, that is zonal velocity U, meridional velocity V and sea surface temperature T. The length of the trajectory in this preliminary step has been fixed at 8 time steps and the dimension of the Eulerian fields is fixed at 24 x 24 points. The effective physical dimensions depend on the data used, and are detailed in the next sections.

For each trajectory it is necessary to select the Eulerian field tiles that encompass it.

In order to do this, each Eulerian field (U, V, T) is divided into 24 x 24 points tiles with 12 points overlap. For each tile all the trajectories completely contained in it are selected. Each of these trajectories is normalized in the range [0, 1] in coordinates relative to the tile. Then the normalized trajectory and a pointer to the tile are stored. It is necessary to store the tile with the trajectory, and not vice versa, to guarantee the randomization of the dataset. In addition, for each tile the latitude and longitude information is stored, so that the lat-lon coordinates of the trajectory can be retrieved.

### *3.1.1 Simulated model data*

The chosen synthetic Eulerian fields data are generated by the Mediterranean Forecasting System (MFS) [13]. The MFS is a simulation model based on the NEMO (Nucleus for European Modelling of the Ocean) engine, coupled with the WaveWatch-III model.

The spatial resolution of the data is 1/24° and the time resolution is 1 day (daily). Only the surface layer has been used. The data have been retrieved from the Copernicus Marine Service, and are freely available:

(Product ID: MEDSEA-ANALYSIS-FORECAST-PHY-006-013).

### *3.1.2 Simulated trajectory data*

The training dataset has been generated using the Runge-Kutta 4th order integration scheme (RK4), and then been sampled at a time-step of 6 hours (four time steps per day), similarly to how real drifters are sampled, for a total of 48 steps (12 days). The code has been written in C++ (C++20) and compiled with g++. The domain considered is the whole Mediterranean Sea. Although only a small part of the dataset has been used for training, a whole year of simulation has been computed, releasing 60000 particle couples (for a total of 12000 particles) uniformly randomly each seven days. The releases used for the training are those of 1st January and 3rd September.

## 3.2 Description of the algorithm

The model is designed to take as input the initial coordinate $x_0$, the velocity and temperature fields, and output the generated trajectory. The architecture of the network and the training setup is inspired by the popular pix2pix image translation network [16].

### 3.2.1 Architecture

The model is composed of three parts (Fig. 5):

1. Trajectory Embedder Ɛ
2. Generator Network G
3. Trajectory De-Embedder $Ɛ^{-1}$

The generator is the only part that is trained during the training procedure. The embedder is fixed a priori and the de-embedder is pre-trained.

The role of the embedder is to encode a two-dimensional spatial position into a 24 x 24 image, so that it can be fed to the generator along with the Eulerian fields, fully exploiting the convolutional nature of the network. Each pixel $p_{ij}$ is determined as

$$p_{ij} = exp(-|x_0 - x_{ij}|^2),$$

where $x_{ij}$ is the position of the pixel in the image rescaled in the range [0, 1]. In the input, the initial position is embedded and then repeated for the necessary time steps.

The role of the de-embedder is to retrieve the two dimensional coordinates from embedded images. The de-embedder is a convolutional neural network composed of five 2D convolutional layers, each fed into a LeakyRelu activation function, and then the result is flattened (spatially, preserving the channels) and fed into a 1D convolution followed by a sigmoid nonlinearity. Regarding the 2D convolutions, the size of the kernels is 3x3 for the first three layers, then 2x2 and 1x1, with stride 1 except for the first two convolutions, which have stride 2. All the convolutions are computed without padding. The number of the filters is 16, 32, 64, 64, 64. The final 1D convolution has a 1x1 kernel, stride 1 and 2 filters, which are outputted as the desired coordinates. Details about the training of the de-embedder are given in the next sections.

The generator G is a hybrid U-net/LSTM network [17-18] . It takes as input a 5D tensor, whose dimensions are [**B**atch, **T**ime, **X** dimension, **Y** dimension, **C**hannel]. The first channel is the meridional velocity, the second channel is the zonal velocity, the third channel is the temperature, the fourth channel is the embedding of the initial position.

The overall architecture of the generator is shown in fig. 5. The U-net is composed of three levels, and in place of the skip connections an LSTM is placed.

The second and third downsampling steps are followed by a batch normalization layer, while the second and third upsampling steps are followed by a dropout layer, with probability p = 0.5.

The convolutional downsample and upsample blocks are shown in Fig. 6. In the downsample block the input is processed by a 3D convolution and a swish activation function, whose output is both sent to the skip connection (i.e. the LSTM input) and to a 3D convolution with stride 2, followed by another swish activation function, whose output is sent to the next level. In the upsample block the input from the lower level is processed by a 3D transposed convolution with stride 2 and a swish activation function, it is the concatenated to the skip connection input (i.e. the output of the LSTM) and the processed by a 3D convolution and a swish activation function, whose output is sent to the next level. The number of filters (for both downsample and upsample blocks convolutions) in the first level is 48, in the second level is 64 and in the third level is 96. The number of LSTM filters is 48 for the first skip connection, 64 for the second and 96 for the third and fourth. A last 3D convolution with kernel size 1 and 1 filter follows to produce a 5D tensor with one channel. The output of the generator is the embedding of the generated trajectory.

### 3.2.2 Training setup

The proposed network has been trained in both GAN and nonGAN frameworks.The generator G is the only part that is trained, hence the loss function is defined using the inputs and outputs of G. The loss function follows [16] and is defined as

$$\begin{aligned}\mathcal{L}_{cGAN} &= -\mathrm{H}(1, D(x,y)) - \mathrm{H}(0, D(x, G(x))) \\ &= \mathbb{E}_{x,y}\left[\log(D(x,y)\right] + \mathbb{E}_{x}\left[\log(1 - D(x, G(x))\right].\end{aligned}$$

Here D(x, y) is a discriminator network that is trained to distinguish between ground truth and generated images. An $L^1$ loss term is added, weighted by a parameter $\lambda$, which in our implementation is fixed at $\lambda = 0.2$. The total loss is

$$\mathcal{L} = \mathcal{L}_{cGAN} + \lambda \mathbb{E}_{x,y}\left[\|G(x) - y\|_1\right].$$

In the nonGAN case the cGAN loss is ignored, leaving only the $L^1$ term.

The optimizer is the AdamW optimizer, with learning rate $\eta = 2 \cdot 10^{-5}$, weight decay $\mu = 0.1$ and the AMSgrad option. The batch size is **B** = 64.

The dataset is composed of 40000 trajectories randomly extracted from each of the two selected releases, for a total of 80000 trajectories. For each of the considered releases, 32000 trajectories are selected for training and the remaining 8000 for validation. Hence the training dataset is 64000 trajectories and the validation dataset is 16000 trajectories. Each epoch the training dataset is shuffled.

The training procedure for both the GAN and nonGAN cases is shown in Fig. 7.

### 3.2.3 De-Embedder training

The de-embedder is pre trained to invert the embedding Ɛ and retrieving 2D coordinates from gaussian images.

The training is carried out in a monte carlo fashion: each iteration step a batch of random couples $x$ is drawn uniformly in [0, 1] x [0, 1] and then embedded into a batch of images $\mathcal{E}(x)$. Then the loss function is computed, gradients are computed and Adam optimizer is applied. A random jitter $z$ is applied with probability p = 0.5 to increase robustness of de-embedding. The loss function is then design to force $\mathcal{E}^{-1}$ to be both left and right inverse of Ɛ:

$$\mathcal{L}_{\mathrm{EMB}} = \mathbb{E}_{x,z}\left[\left\|\mathcal{E}(x) - \mathcal{E}(\mathcal{E}^{-1}(\mathcal{E}(x) + z))\right\|_2^2\right] + \mathbb{E}_{x,z}\left[\left\|x - \mathcal{E}^{-1}(\mathcal{E}(x) + z)\right\|_2\right]$$

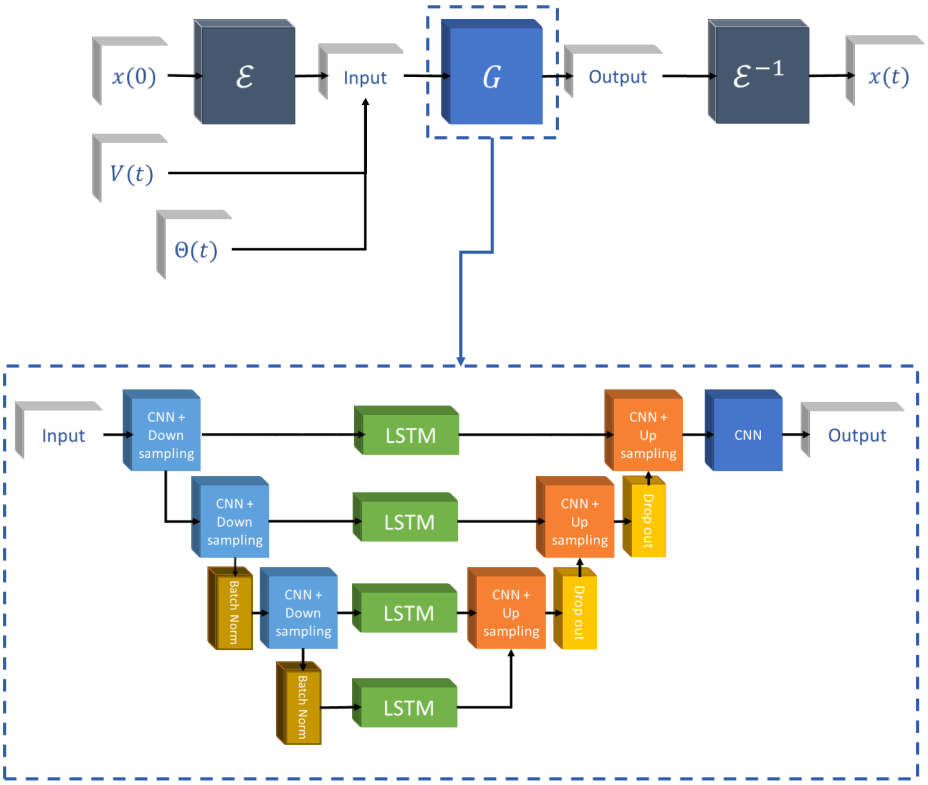


**Figure 5.** Architecture overview

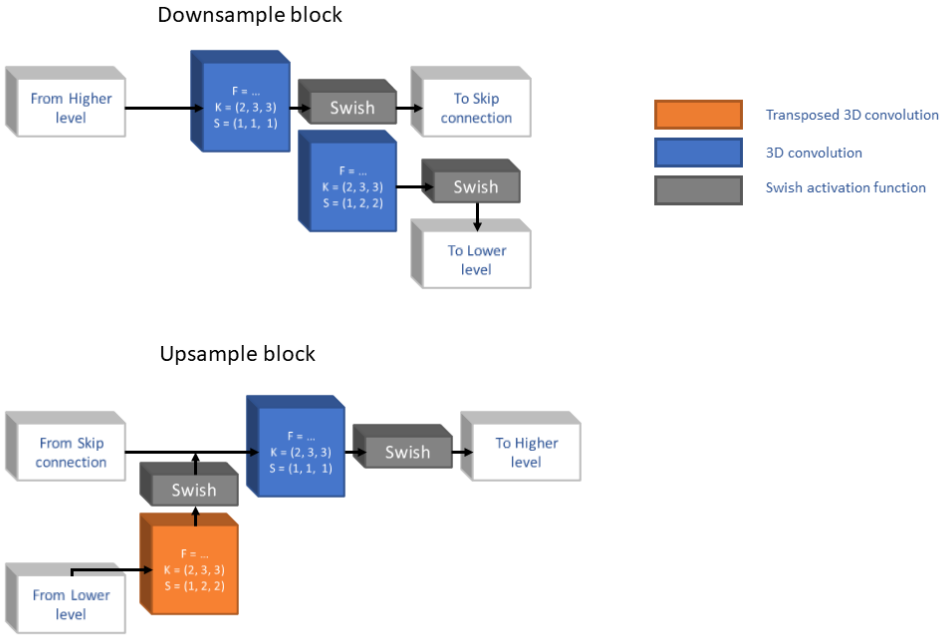


**Figure 6.** Details of convolutional blocks

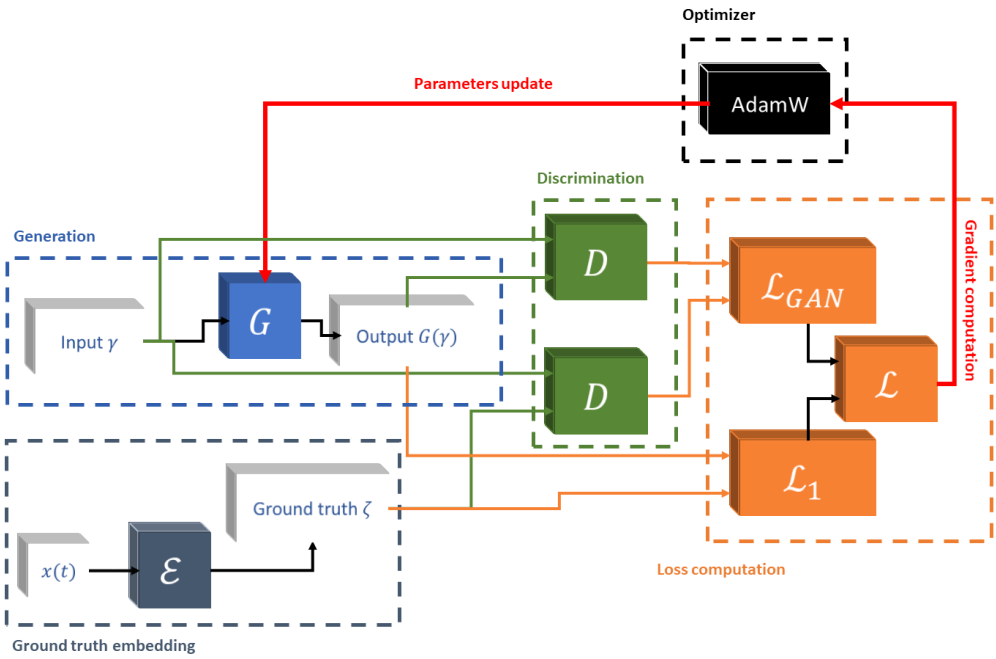


**Figure 7.** Training setup

**References:**

[1] Sammartino, M., Buongiorno Nardelli, B., Marullo, S., & Santoleri, R. (2020). An artificial neural network to infer the Mediterranean 3D chlorophyll-a and temperature fields from remote sensing observations. *Remote Sensing, 12*(24), 4123. https://doi.org/10.3390/rs12244123.

[2] Rio, M. H., Pascual, A., Poulain, P.-M., Menna, M., Barceló-Llull, B., & Tintoré, J. (2014). Computation of a new mean dynamic topography for the Mediterranean Sea from model outputs, altimeter measurements, and oceanographic in situ data. *Ocean Science, 10*(5), 731–744.

[3] Wong, A. P., Johnson, G. C., & Owens, W. B. (2003). Delayed-mode calibration of autonomous CTD profiling float salinity data by θ–S climatology. *Journal of Atmospheric and Oceanic Technology, 20*(2), 308–318.

[4] Böhme, L., & Send, U. (2005). Objective analyses of hydrographic data for referencing profiling float salinities in highly variable environments. *Deep Sea Research Part II: Topical Studies in Oceanography, 52*(3–4), 651–664.

[5] Owens, W. B., & Wong, A. P. (2009). An improved calibration method for the drift of the conductivity sensor on autonomous CTD profiling floats by θ–S climatology. *Deep Sea Research Part I: Oceanographic Research Papers, 56*(3), 450–457.

[6] Cabanes, C., Thierry, V., & Lagadec, C. (2016). Improvement of bias detection in Argo float conductivity sensors and its application in the North Atlantic. *Deep Sea Research Part I: Oceanographic Research Papers, 114*, 128–136.

[7] Morel, A., & Berthon, J.-F. (1989). Surface pigments, algal biomass profiles, and potential production of the euphotic layer: Relationships reinvestigated in view of remote-sensing applications. *Limnology and Oceanography, 34*(8), 1545–1562.

[8] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). KAN: Kolmogorov-Arnold Networks. *arXiv.* https://arxiv.org/abs/2404.19756.

[9] Liu, D. C., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming, 45*(1), 503–528. https://doi.org/10.1007/BF01589116.

[10] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *International Conference on Learning Representations.*

[11] De Boor, C. (2001). *A practical guide to splines.* Springer.

[12] Griewank, A., & Walther, A. (2000). *Evaluating derivatives: Principles and techniques of algorithmic differentiation* (2nd ed.). Frontiers in Applied Mathematics.

[13] Clementi, E., Pistoia, J., Escudier, R., Delrosso, D., Drudi, M., Grandi, A., Lecci, R., Cretí, S., Ciliberti, S., Coppini, G., et al. (2021). Mediterranean Sea Analysis and Forecast (CMEMS MED-Currents 2016–2019) (Version 1) [Data Set]. Copernicus Monitoring Environment Marine Service (CMEMS).

[14] Taburet, G., Sanchez-Roman, A., Ballarotta, M., Pujol, M.-I., Legeais, J.-F., Fournier, F., Faugere, Y., & Dibarboure, G. (2019). DUACS DT2018: 25 years of reprocessed sea level altimetry products. *Ocean Science, 15*(6), 1207–1224.

[15] Ciani, D., Charles, E., Buongiorno Nardelli, B., Rio, M.-H., & Santoleri, R. (2021). Ocean currents reconstruction from a combination of altimeter and ocean colour data: A feasibility study. *Remote Sensing, 13*(12), 2389.

[16] Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5967–5976. https://doi.org/10.1109/CVPR.2017.632.

[17] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *arXiv.* https://arxiv.org/abs/1505.04597.

[18] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM—A tutorial into long short-term memory recurrent neural networks. *arXiv.* https://arxiv.org/abs/1909.09586.