MISSIONE 4
ISTRUZIONE
RICERCA

D5.31 Final report on data-driven model validation

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

Deliverable number:	D5.31
Work package:	WP5 – Marine Domain 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):	14
Actual delivery date:	28/02/2025
Author(s) (Partner-OU):	Bruno Buongiorno Nardelli, Daniele Ciani, Lorenzo della Cioppa, Claudia Fanelli, Michela Sammartino (CNR-ISMAR-NA)
Reviewed by:	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

1 4

1.1 4

1.1.1 4

1.1.2 4

1.1.3 5

1.2 6

2 11

2.1 12

2.1.1 12

2.1.2 12

2.2 13

3 16

3.1 17

3.1.1 17

3.1.2 17

3.2 Description of the algorithm **15**

3.2.1 17

3.2.2 18

3.2.3 18

Index of tables

Table 1 - Input and output variables 7

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

Satellites provide quasi-synoptic monitoring of surface ocean EOVs/ECVs. However, since satellite sensors are limited to surface-layer data (or to the first few meters in depth), in situ measurements remain essential for oceanographic research. To bridge the gap between satellite observations and in situ data and achieve comprehensive 3D reconstructions, data-driven interpolation and extrapolation techniques are necessary. Recently, neural networks have been proposed for predicting both Temperature and Chlorophyll [1], demonstrating effectiveness in this task.

In this activity, we applied a recently proposed novel neural network architecture to the task of 3D reconstruction of ocean state variables. The network architecture is built upon the recently introduced and innovative Kolmogorov-Arnold Network (KAN), which has been re-implemented and tailored to suit our objective. The proposed model takes as input observed satellite surface data and outputs Salinity (S), Temperature (T), Density (D) and Chlorophyll (CHL) at the desired depth.

The proposed model is well-suited for integration with observations from various in situ platforms, including Argo floats, CTD profilers, and others.

1.1 Input data selection and pre-processing

In this section we give a brief overview of the data that compose the dataset used to train and test the neural network. The target outputs for the network are collected from in situ measurements, while the inputs come from both satellite and in situ measurements.

1.1.1 Surface data

- *Satellite Absolute Dynamic Topography dataset*

  For Absolute Dynamic Topography (ADT) data, a Copernicus Marine Service regional product has been used (https://doi.org/10.48670/moi-00141). It also includes zonal and meridian components of absolute geostrophic currents (U & V). These satellite observations have been matched-up with in-situ measurements for dataset construction.

1.1.2 In situ profiles

Both oceanographic cruises and Biogeochemical Argo Floats have been used as a source for in situ measurements (Fig. 1).

- *Oceanographic cruise database*

  This dataset is based on 26 oceanographic cruises conducted in the Mediterranean Sea between 1997 and 2017 that have been considered for temperature, chlorophyll, and salinity profiles (red crosses in Fig. 1). Chlorophyll has been estimated by calibrating the fluorometer signals with bottle samples collected simultaneously at each station during each cruise.

- *Biogeochemical-Argo database*

  Hydrographic (temperature and salinity) and bio-geochemical (chlorophyll-a) profiles were collected by Biogeochemical-Argo (BGC-Argo) floats. Only the BGC-Argo in the 2012-2022 period were considered (blue crosses in Fig. 1). All the data are freely available from the Coriolis Global Data Assembly Centers (GDAC) FTP site (ftp.ifremer.fr; https://doi.org/10.17882/42182).

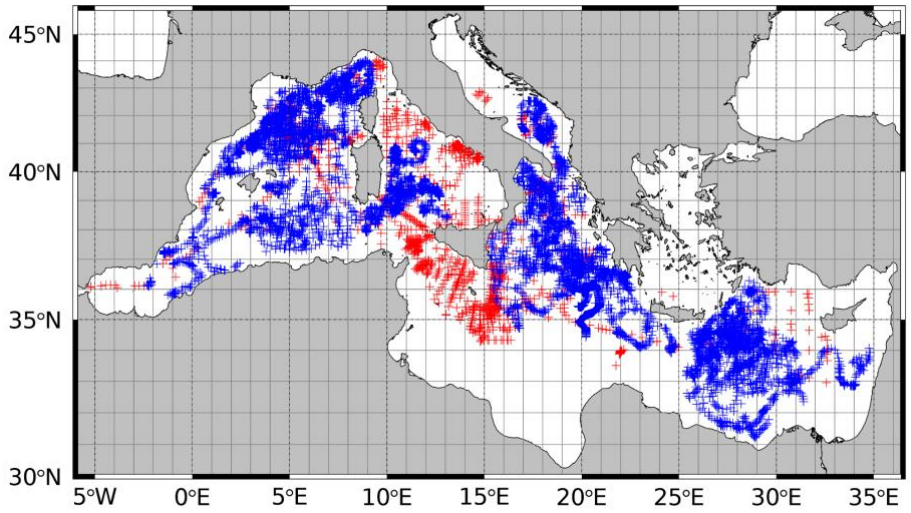


Figure 1. In situ measurements: oceanographic cruises (red crosses) and Bio-Argo (blue crosses) stations.

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. Chlorophyll and temperature/salinity profiles were processed according to the methodology detailed in [1]. In particular, it comprises:

- **Visual Inspection** to ensure the internal consistency of the dataset.
- **Verification of Initial Acquisition Depth** to reject profile with a first acquisition depth too high.
- **Evaluation of Missing Data Points** to discard profiles with an excessive number of missing points. In the case this number was low, linear interpolation has been used to fill the gaps.
- **Maximum Depth Criterion** to ensure that the selected profiles reached at least a depth of 150 m.

For BGC-Argo profiles, only those profiles with a depth of at least 150 m, including data with flags "1", "2", "5", and "8" were considered [3b]. All profiles have been then interpolated on a regular grid of 1 m spacing.
In order to simulate CHL satellite observations, which is usually approximated by the chlorophyll-a concentration integrated up to the depth of roughly 1/5 of the euphotic depth, from both oceanographic and BGC-Argo datasets the optical weighted pigment (OWP) has been estimated from chlorophyll profiles according to a modified version of Morel's model used to compute the attenuation coefficient [7].
The final in situ database was subsequently matched up with satellite absolute dynamic topography and the two components of surface geostrophic current.

Due to the lack of high-resolution satellite products prior to 2008, surface values of in situ T and S profiles were used as proxies for satellite-derived sea surface temperature (SST) and sea surface salinity (SSS), respectively. Density profiles were computed from in situ T and S data through the

state equation, and their surface values used as proxies for satellite-derived sea surface density (SSD). OWP was used as a proxy for satellite CHL.

In addition, profiles whose chlorophyll values were constant within the first 15 meters of depth were excluded. Given the broad range of chlorophyll concentrations in the Mediterranean Sea, the surface OWP and vertical CHL values have been log10-transformed before being input into the network. Data were then normalized in the 0–1 range.

Then, the 8624 selected profiles were randomly split into training (80%) and validation (20%) subsets. It has to be noted that this database was originally developed within the ESA 4DMED-Sea project and it is used here as the basis for implementing the AI algorithm detailed in the following section.

1.2 Description of the algorithm

Input network variables are eleven, while the targets are four variables.

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

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

To capture seasonal periodicity, the Julian day (JD) is represented by two variables: JD1 = cos(2πJD/365) and JD2 = sin(2πJD/365). As the other variables, JD1 and JD2 are normalized in [0, 1] as well.

The network is based on the recently proposed Kolmogorov-Arnold Network (KAN) [8], which are inspired by Kolmogorov-Arnold Theorem, stating that every multivariate function can be written as a sum of univariate functions. KANs substitute the weight matrix multiplication and successive nonlinear function of traditional MLPs with spline evaluation (see Fig. 2).

Given an input vector $X = (x_1, \ldots, x_p, \ldots, x_N)$ and an output vector $Y = (y_1, \ldots, y_q, \ldots, y_M)$, the output of a single KAN layer can be expressed in Einstein summation notation as

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

where B_k are B-spline bases and $c_k{}^{q,p}$ are the relative coefficients.

The number K of spline functions is set by the user and is related to the size of the underlying grid. An additional silu nonlinearity is added to the B-splines, weighted by a trainable parameter. The weights are randomly initialized, according to either an uniform distribution or a normal distribution.

The grid refinement procedure proposed in the original KAN formulation has been implemented here as well: the B-spline computation grid is refined at predefined times during training, and the spline coefficients are recomputed by least square error minimization (see Fig.3).

We stack several KAN layers to form a deep KAN.

We don't include the original entropic regularisation term and pruning steps that the original formulation presents. As they were introduced to achieve sparsity, in our case proved detrimental to the training.

We train the network using SGD-based AdamW optimizer [10].

The code is implemented in Python using the TensorFlow machine learning package, which offers both functional and traditional APIs for easily defining customized networks. Indeed, due to the specificness of the KAN, a custom layer has been implemented.

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. The tensor multiplication described by (*) is computed in parallel across the batch.

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

1.3 Results

Two different training experiments have been performed on the described dataset with the same network. In addition to the above mentioned input predictors, depth has been added.

The network is composed of 3 concatenated KAN layers, with output dimensions of, respectively, 20, 10 and 4. Spline order has been set to 6 for all the layers, and grid dimension has been initialized as 2 for the first 2 layers and as 5 for the last layer. The grid size is doubled each 25 epochs, and grid sizes have been limited to 10 for the first 2 layers and to 20 for the last layer. The total number of trainable parameters is 4400. The training is carried out for 500 epochs, and after the training the number of parameters has grown to 8560. We stress that the number of parameters is lower than other ANN approaches.

The only difference between the two training experiments is the value of the weight parameter w in the optimization algorithm AdamW. In the first experiment w is set to 0.01, while in the second it is set to 0.1.

In order to find a good starting point for the training, a pre-training simulated annealing procedure has been performed for a total of 30 epochs, with a linear temperature decay schedule.

For both training and pre-training a mean absolute value (MAE) function has been used as loss.

The two training experiments do not present visible differences in terms of results. The loss values across epochs are shown in fig. 4. As it can be seen the network is capable of good generalization.

In figures 5 and 6 the error profiles for the two experiments are shown. The error profiles have been computed with both MAE and RMSE, to be consistent with previous studies. The error profiles have been de-normalized using minima and maxima of the training dataset.

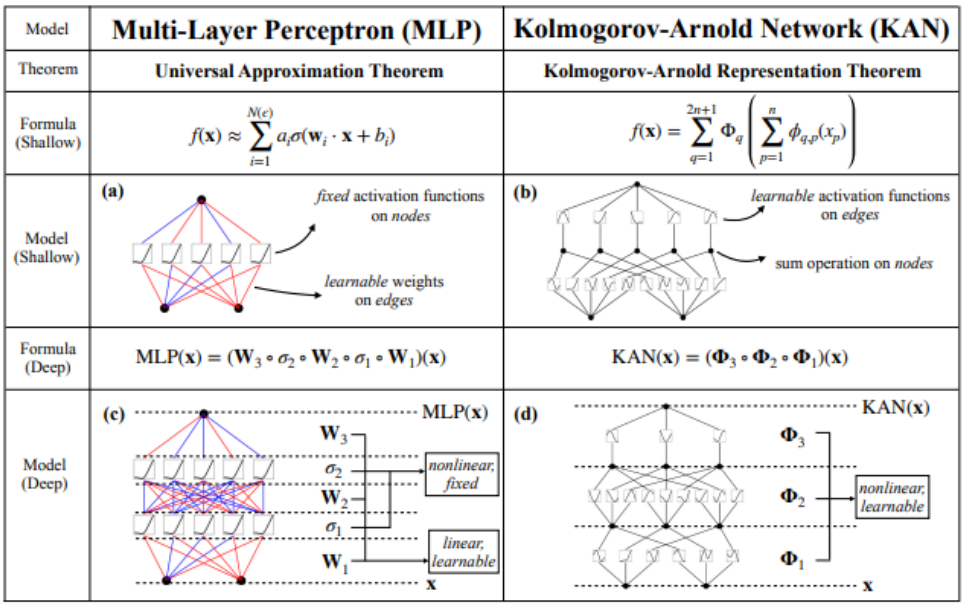


Figure 2. Comparison between MLP and KAN [8].

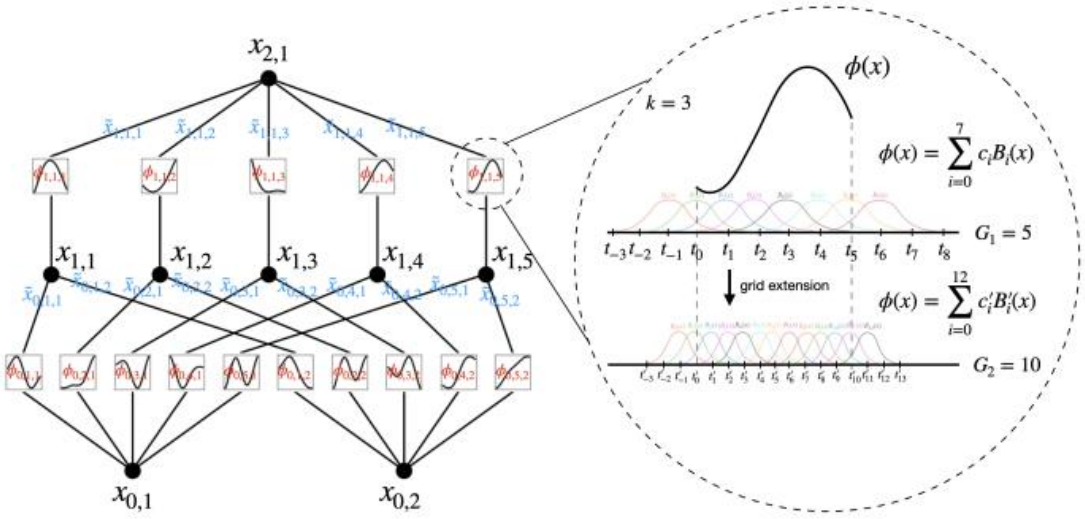
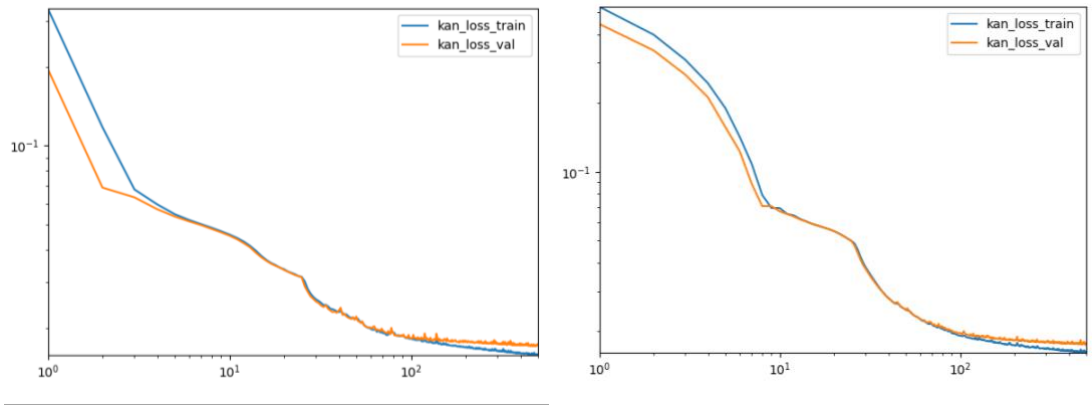


Figure 3. Computation of the splines in the KAN nodes, with grid update detailed [8].

1.



(a) (b)

Figure 4. Loss functions for the two training experiments: panel (a) refers to the training with weight parameter w = 0.01, panel (b) refers to the training with weight parameter 0.1. X axis is the epoch number, Y axis is the value of the loss function.

2.

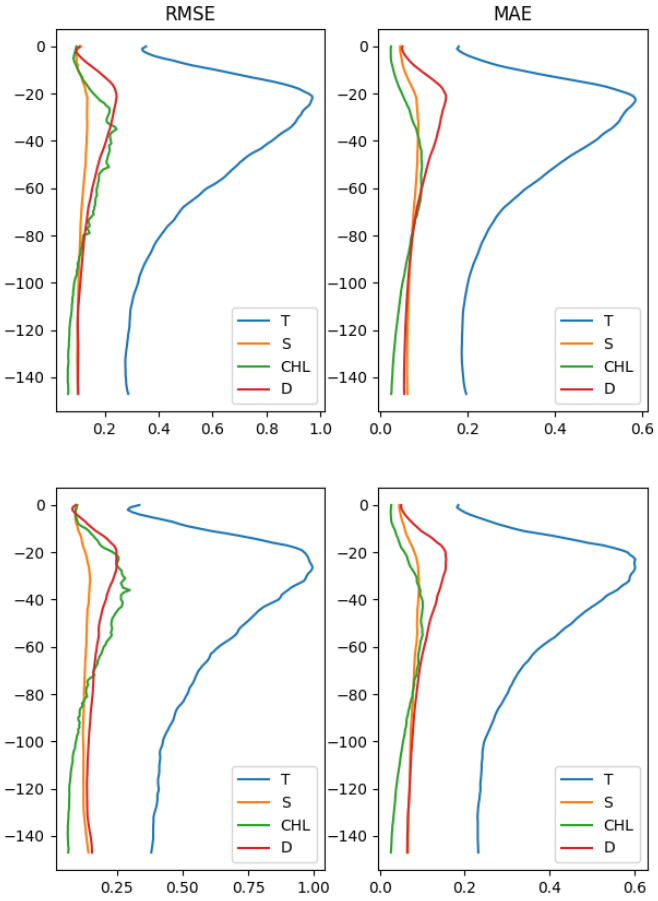


Figure 5. Error profiles for the first experiment (w = 0.01). Top row is the error profiles computed on the training dataset, bottom row is the error profiles computed on the test dataset

3.

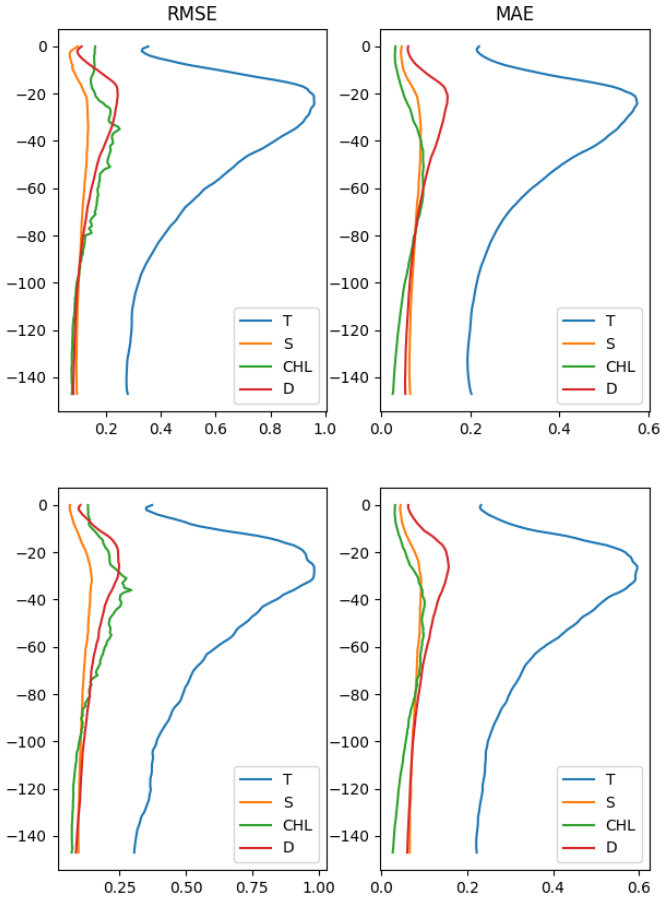


Figure 6. Error profiles for the first experiment (w = 0.1). Top row is the error profiles computed on the training dataset, bottom row is the error profiles computed on the test dataset

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

Our study focuses on improving the mapping of Absolute Dynamic Topography (ADT) and Sea Surface Temperature (SST) from satellite data, addressing challenges like instrument limitations, sampling gaps, and interpolation errors in gap-free (L4) analyses. Using a Convolutional Neural

Network (CNN) adapted from image super-resolution models, we conducted an Observing System Simulation Experiment (OSSE) with Copernicus model outputs. Unlike earlier studies, which relied on idealized SST data, we introduced realistic L4 SST processing errors and adapted the CNN to jointly predict high-resolution SST and ADT from synthetic satellite L4 datasets. By incorporating physics-informed loss functions, this approach enhances our ability to reconstruct high-resolution ocean surface dynamics while maintaining dynamical consistency.

2.1 Input data description

The network was trained on OSSE data and applied to satellite-derived ADTs and SSTs from the Copernicus Marine Service to reconstruct super-resolved ADTs and geostrophic currents. Focusing on the Mediterranean Sea in 2017—a region characterized by small Rossby deformation radii (~10 km)—40 test dates were selected to cover various dynamical regimes throughout the year, while the remaining data were divided into 76 × 100 pixel tiles (~300 × 400 km) with 50% spatial overlap, yielding 42,250 samples. These samples were normalized and filtered to remove signals larger than ~200 km, preserving local SST variations linked to horizontal current advection. By extracting patterns from SSTs, the network enhanced ADT reconstruction, capturing ocean surface dynamics. The dataset was split for training (85%) and validation (15%).

2.1.1 Simulated SST/SSH data

We used daily outputs of SSH and SST provided by the Mediterranean Forecasting System (MFS) through the Copernicus Marine Service (Product ID: MEDSEA-ANALYSIS-FORECAST-PHY-006-013), focusing on the Mediterranean Basin within the coordinates 30° to 46°N and -6° to 37°E, available on a 1/24° regular grid with 125 vertically unequally spaced levels. From these data, we therefore produced one year (2017) of synthetic Satellite-Equivalent altimeter-derived Absolute Dynamic Topography (referred to as SE-ADT) maps 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:

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

The mean dynamic topography (MDT) was provided as a static field, with a constant adjustment of 0.344 m subtracted to ensure the 2017 Mediterranean Sea SLA had a zero spatio-temporal average. A Loess filter was applied to synthetic data to reduce large-scale, high-frequency variability typically addressed by dynamic atmospheric correction (DAC). SLA was sampled along synthetic altimeter tracks from Jason-3, Sentinel-3A, SARAL/Altika, and CryoSat-2, using the SWOT simulator to account for orbit and noise characteristics. These measurements were processed with the DUACS system to create L4 SLA maps, employing the DUACS DT2018 optimal interpolation (OI) configuration for the Mediterranean. The final L4 SLA maps were combined with filtered large-scale maps to generate Absolute Dynamic Topography (ADT) and provided daily on a 1/8° grid [15].

2.1.2 Satellite-derived SST/SSH data

Here we briefly recall the satellite data used as input for the network:

- Sea surface geostrophic currents, derived from optimally interpolated absolute dynamic topography data, were obtained from the Copernicus Marine Service (ID product: SEALEVEL_EUR_PHY_L4_MY_008_068/cmems_obs-sl_eur_physsh_my_allsat-l4-duacs-0.125deg_P1D). These datasets integrate observations from a radar altimeter constellation (four to six instruments) covering 2008–2019 [14]. The currents are available as daily fields at a 1/8° horizontal resolution, and the full time series was extracted for this study.

- Gap-free (L4) SST data provided by the Copernicus Marine Service at Near Real Time (Product ID: SST-MED-SST-L4-NRT-OBSERVATIONS-010-004-c-V2) were also acquired. 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°. The data were then processed by applying a low-pass Lanczos filter and remapped onto the final model grid (1/24° resolution) using bilinear interpolation.

2.2 Description of the algorithm

The architecture, illustrated in Figure 4 and known as an autoencoder, consists of two main components: an encoder and a decoder. The encoder processes low-resolution input data to generate a latent representation of key features by down-sampling the dataset. The decoder then generalizes these features to produce high-resolution SST image candidates.

The encoder extracts features from input tiles (76×100×6), which include SE-ADT, SE-ADT error, SE-SST, SE-SST error, ∂t(SE-SST), and ∂t(SE-SST) error. The encoder consists of multiple convolutional layers, progressively reducing the input dimensions. The decoder mirrors the encoder's structure, employing transposed convolutions to reconstruct high-resolution outputs from the feature vector, restoring the missing region's original size. To mitigate overfitting, a 20% dropout is applied to the feature vector and all convolutional layers in the network are connected with LeakyReLU activation functions with a slope coefficient of 0.2.

The autoencoder is trained using a physics-informed loss function (LF), defined as a weighted combination of multiple terms:

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

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.

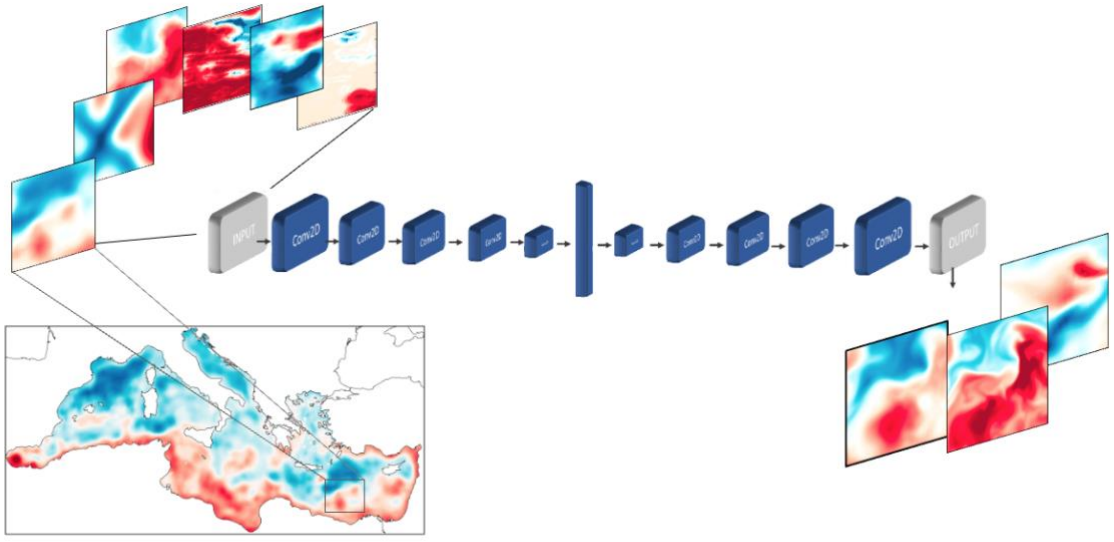


Figure 4. Autoencoder architecture. The set of input tiles, from left to right, respectively indicate: SE-ADT, SE-ADT error, SE-SST, SE-SST error, ∂t(SE-SST) and the ∂t(SE-SST) error. The three output tiles, from left to right, respectively indicate the SR-ADT, SR-SST and SR- ∂tSST. Conv2D stands for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

2.3 Results

The autoencoder was trained considering six predictors (SE-ADT, SE-ADT error, SE-SST, SE-SST error, ∂t(SE-SST) and the ∂t(SE-SST) error) and three targets (SR-ADT, SR-SST and SR- ∂tSST) over 1 year of data (2017). We then employed it to reconstruct the latter from the test dataset left totally independent from the training one. The results were compared with the reconstructions obtained from the network developed by [19] and called dilated Adaptive Deep Residual Network for Super Resolution (dADR-SR).

The neural network model trained with the OSSE is now evaluated for its ability to predict super-resolved ADTs, utilizing state-of-the-art L4 ADTs derived from satellite altimetry and high-resolution L4 satellite SSTs. The input satellite data, generated by the Copernicus Marine Service as detailed in Sec. 2.1.2, spans the period from 2008 to 2019. The performance of the CNN is assessed by deriving super-resolved (SR) geostrophic currents from the SR-ADTs using the geostrophic approximation. This process involves applying a finite-central-differences operator to the SR-ADTs to compute spatial gradients, with drifting buoy measurements serving as a validation benchmark. Validation is conducted through root mean square error inter-comparisons, achieved by interpolating the gridded altimeter-derived and SR currents to match the drifter time and location during 6-hourly acquisitions for both circulation components, as described in [20].

To maximize the number of matchups between gridded and in situ-measured currents, statistics are presented in 2° × 2° grid boxes using a 12-year time series (2008–2019). This approach ensures coverage of in situ measurements, as shown in Fig. 5, with each box containing a minimum of approximately 100 observations.

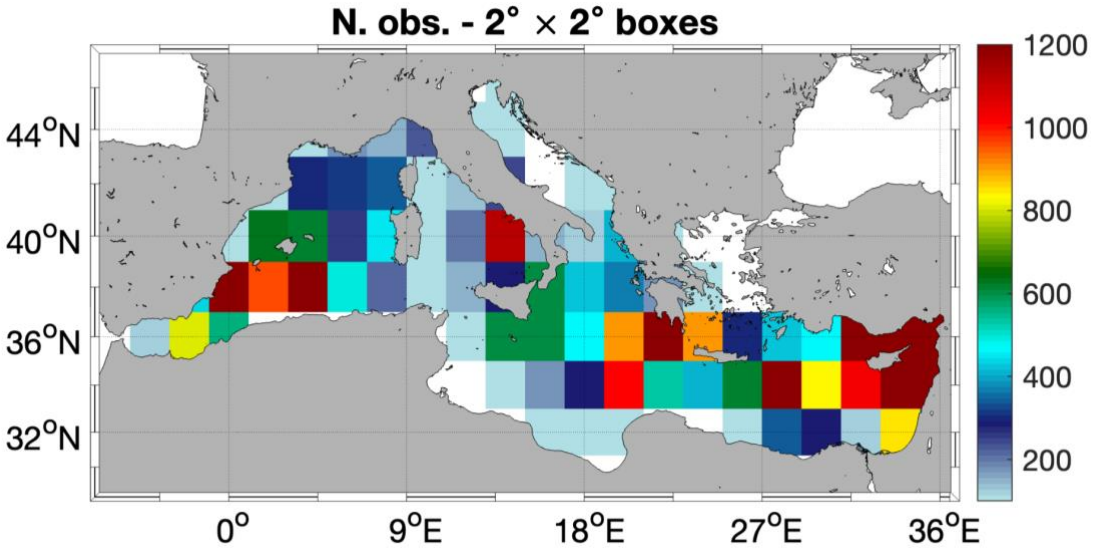


Figure 5. Number of in situ measurements from drifting buoys in the 2008–2019 time frame.

Compared to the standard altimeter-derived mapping, AE currents reduce the RMS error by approximately 2 to 8 cm s^{-1} across most of the basin for both zonal and meridional flows (Fig. 6). However, degradations relative to the altimeter (Alti) currents are primarily observed in coastal regions, with the most significant impacts occurring at the eastern edge of the Levantine Basin. On a basin-wide scale, the RMS errors of the Alti and SR currents highlight an improvement in the super resolved currents presented here, showing a reduction in RMS error of up to 1 cm s^{-1}.

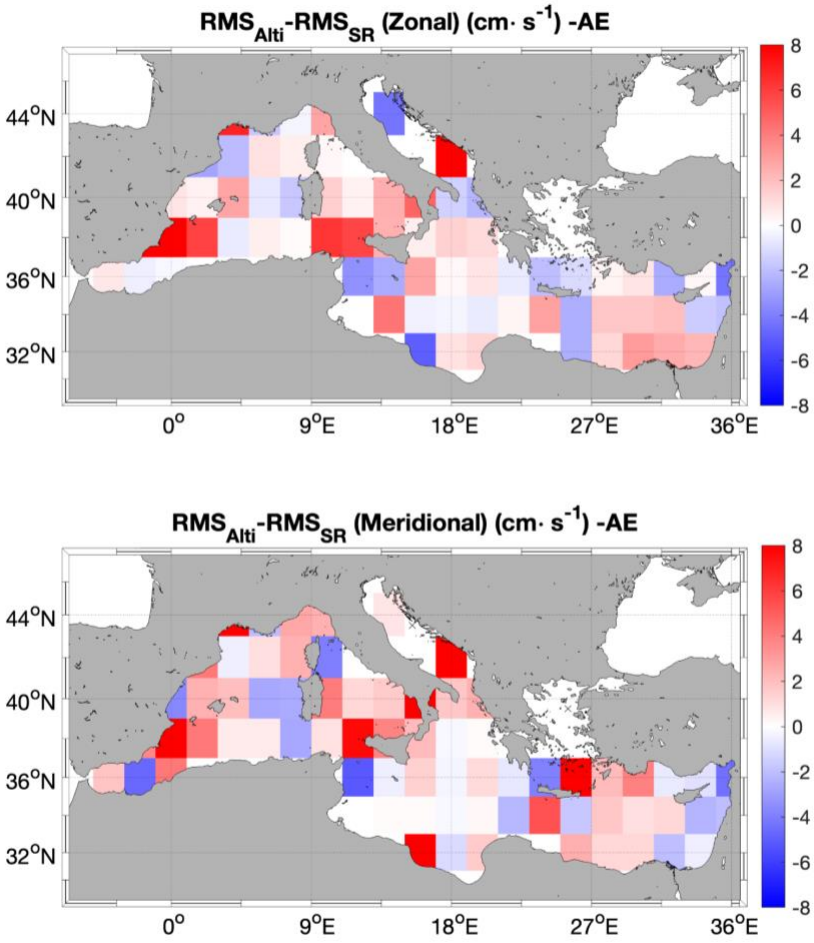


Figure 6. Differences of rms errors between the altimeter-derived (Alti) and SR currents reconstructed by the autoencoder (AE) architecture: (**top**) zonal flow and (**bottom**) meridional flow. Red areas express an improvement with respect to standard altimetry.

To obtain a quantitative assessment of our methodology, the two performances were evaluated comparing independent surface current estimates from in situ drifting floats with corresponding altimeter and model data collected at the same locations and times (Fig. 7). For both networks the error associated with the super-resolved and standard altimetric estimates (calculated here as the absolute difference compared to drifter data) does not exhibit a consistent pattern along individual drifter trajectories. Instead, there are alternating instances of improvement and degradation in the current velocity components, without a clear geographical trend. Overall, both fields' reconstructions appear to offer slightly more accurate values with respect to altimeter data in certain (mainly offshore) regions of the western basin, whereas, on average, they seem to perform less effectively in

the easternmost part of the Levantine Basin, near the coasts of Israel and Lebanon. However, the autoencoder reconstruction exhibits a smaller error with respect to the dADR-SR outputs, where blue pixels (corresponding to a degradation of the network's reconstructed field) are largely present, especially in the Alboran Sea and the Levantine basin.

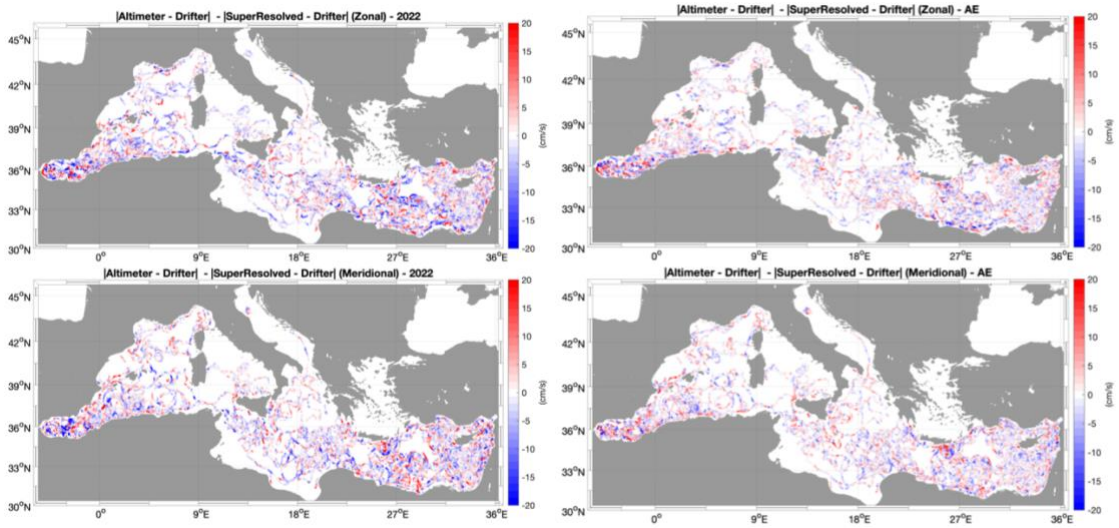


Figure 7. Performances of (**left**) dADR-SR [19] and (**right**) the autoencoder (AE) reconstructions of the geostrophic currents with respect to standard altimeter L4 data along drifter trajectories. The plots show the difference between the absolute error of altimetry and that of super-resolved currents, estimated vs the drifter velocities: (**top**) zonal component; (**bottom**) meridional component. Positive values indicate an improvement with respect to altimetry.

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

The Lagrangian approach plays a crucial role in various scientific and operational applications, including debris advection forecasting, dynamical analysis, and model validation.

Since the number of deployed drifters is often inadequate for the required time and space sampling in these applications, trajectory simulations are essential.

While satellites provide synoptic observation of the ocean surface, their coverage is often limited in both time and space. Additionally, current velocities are typically estimated under the assumption of geostrophic balance, which can lead to inaccuracies and is ineffective in regions where the Coriolis force is weak, such as near the Equator. Conversely, General Circulation Models (GCMs) generate high-resolution Eulerian fields, but their accuracy remains constrained when compared to real-world data. In fact, trajectories derived from synthetic fields generally exhibit lower accuracy than those based on geostrophic calculations. Ultimately, the selection of the underlying model depends on which is best suited for the specific task.

A novel Artificial Neural Network for Lagrangian trajectories prediction has been proposed as part of this activity. Taking inspiration from computer vision algorithms we propose and hybrid U-net/LSTM deep network that takes as input initial condition and Eulerian fields (velocity and sst in this study) and outputs the desired trajectory.

3.1 Input data description

The inputs of the model are the initial point $x_0$and the chosen Eulerian fields, that is zonal velocity U, meridional velocity V and sea surface temperature T. Eulerian fields have been divided into 24x24 tiles and trajectory length for training is 8 time steps (corresponding to 2 days).

3.1.1 Simulated model data

Synthetic Eulerian fields used for trajectory training dataset are computed by the Mediterranean Forecasting System (MFS) [13]. Spatial resolution is 1/24° and time resolution is 1 day (daily). Only the surface layer has been used. The data are freely available by the Copernicus Marine Service (Product ID: MEDSEA-ANALYSIS-FORECAST-PHY-006-013).

3.1.2 Simulated trajectory data

The training dataset has been generated in C++ using the Runge-Kutta 4th order integration scheme (RK4), and then subsampled at a time-step of 6 hours for a total of 48 steps (12 days). Simulations have been carried out by releasing 60000 particle couples (for a total of 12000 particles) uniformly randomly each seven days for a year in the mediterranean basin.

3.2 Description of the algorithm

Here we describe the proposed architecture, inspired by computer vision approaches to image to image translation [16].

3.2.1 Architecture

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

4. Trajectory Embedder $\mathcal{E}$
5. Generator Network G
6. Trajectory De-Embedder $\mathcal{E}^{-1}$

The generator is trained while the embedder is fixed a priori and the de-embedder is pre-trained.

The embedder encodes each two dimensional initial position into a 24 x 24 image as a gaussian centered in the (normalized) initial positions.

The de-embedder is a deep CNN which is trained in advance using a monte carlo self-supervised learning setup.

The generator takes as input the concatenation of the Eulerian Fields for the reference time frame, along with the embedded initial position, replicated for each time step.

The architecture is composed of a U-net, in which the skip connections are replaced by LSTMs.

The number of filters evolves across levels as 48, 64, 96. The filters in the LSTM evolve as 4, 64, 96. A 3D convolution with kernel size 1 and 1 filter finally produces the outputwhich is the embedding of the generated trajectory. Architecture and CNN blocks are shown in figures 5 and 6.

3.2.2 Training setup

The proposed network has been trained in both GAN and nonGAN fashions. The generator G is trained along with a discriminator D, whose task is to correctly classify G's outputs as real or fake. The loss function follows [16] and is defined as the sum of L^1 norm between generated trajectory and ground truth and a binary cross entropy term.

The binary cross entropy term evaluates the correctness of the classification made by D.

In the nonGAN case the binary cross entropy term is ignored, leaving only the L^1 term.

The dataset consists of 80,000 trajectories equally divided between each of the two selected releases for training. For each release, 32,000 trajectories are allocated for training, while the remaining 8,000 are used for validation. As a result, the training dataset comprises 64,000 trajectories, and the validation dataset consists of 16,000 trajectories. The training dataset is shuffled at the beginning of each epoch.

The training procedure is shown in Fig. 7.

i.

3.3 Methods

Results have been evaluated using several widely accepted metrics: Root Mean Square Error (RMSE), Liu Index [17], and Finite Size Lyapunov Exponents [18].

RMSE is computed as

$$\mathrm{RMSE} = \left(\frac{1}{K} \frac{1}{T} \sum_{k=1}^{K} \sum_{t=1}^{T} d_k(t)^2 \right)^{1/2}$$

where $d_k\,(t)$ is the distance between the k-th reference trajectory and its simulated counterpart, computed using Haversine Distance.

Liu index for the k-th simulated particle is defined as

$$\mathrm{Liu}_k = \frac{\sum_{t=1}^{T} d_k(t)}{\sum_{t=1}^{T} l_k(t)}$$

where $d_k\,(t)$ is computed as before and $l_k\,(t)$ is the length of the k-th reference trajectory at time t. In Liu index the simulation error is divided by the length of the trajectory, hence introducing a weighting that takes into account the size of the simulation.

FSLE spectra have been introduced for the study of dynamical systems. In [18] they have been used for the evaluation of Lagrangian simulations. FSLE spectra analyze trajectories according to the separation scale between couples of particles. Given an initial scale δ_0 and an amplification ratio ρ, a sequence $\delta_n\,, n = 1, \dots, N$ of scales is defined by $\delta_n = \rho^n\,\delta_0$. The FSLE at scale δ_k is defined as

$$\lambda(\delta_n) = \frac{1}{\langle \tau_k(\delta_n) \rangle} \log(\rho)$$

where τ_k is the time it takes for the k-th couple to separate from scale δ_n to scale δ_{n+1} and the angled braces denote the average across all couples.

FSLE decay characterises dynamical regimes, for a more detailed discussion see [18].

Simulations are evaluated with FSLE-I and FSLE-II. The first type of FSLE is the standard FSLE, computed for both ground truth and simulation, and then compared. The second type of FSLE considers couples formed by a reference trajectory and a simulated trajectory with the same initial condition. FSLE-I assesses the capability of the model to correctly reproduce dynamical regimes.

FSLE-II indicates how much the simulation error rate grows across scales. It also indicates the correlation between simulation and ground truth. A ballistic decay ($\lambda(\delta) \sim \delta^{-1}$) is expected up to a certain scale δ^*, after which turbulent ($\lambda(\delta) \sim \delta^{-3/2}$) or diffusive ($\lambda(\delta) \sim \delta^{-2}$) regimes occur, as the simulation becomes more and more uncorrelated to the ground truth.

In [18] the mean error growth velocity $\gamma(\delta)$ is defined based on FSLE

$$\gamma(\delta) = \delta\lambda(\delta)\frac{\rho - 1}{\log(\rho)}$$

To compare different models, we average $\gamma(\delta)$ across scales, accordingly to the scale size:

$$\mathrm{ASV} \approx \frac{1}{\delta_{\bar{N}} - \delta_0} \sum_{k=0}^{\bar{N}} (\delta_{n+1} - \delta_n)\gamma(\delta_n)$$

We refer to this integrated measure as Average Separation Velocity (ASV).

3.4 Results

Two different sub-datasets have been considered for training: a subset of 80000 trajectories 8 times steps long and a subset of 40000 trajectories 16 time steps long. For each of the sub-datasets the examples have been sampled in equal parts from the releases of 1-12 January and 3-15 September. The disparity in terms of number of examples is due to computational constraints. The sub-datasets are split in training dataset and validation dataset with a ratio of 80/20. Each training has been carried out for 1000 epochs, with batch size 64. The Eulerian fields have been subdivide in 24x24 tiles with 50% overlap. For each of the training setups, both GAN and non-GAN losses have been used. Independent from the time steps training length, during inference the network is iterated several times to generate longer trajectories. After each iteration the tiles for the next inference are chosen such that the initial condition is in the center, so that the risk of the trajectory exiting the tile is minimized.

Testing has been carried out on two different time periods: 1 - 12 January and 21 May - 3 June. For the first time period, only trajectories not used for training nor validation have been used. For the first period 33400 trajectories have been used, while for the second period the number of trajectories is 38400.

The results for the considered metrics are shown in table 1 (1 - 12 January) and table 2 (21 May - 3 June). Confidence intervals for Liu index and RMSE have been computed using bootstrap as two

times the standard deviation. Though these are widely accepted metrics for Lagrangian trajectory model evaluation, they fail to accurately capture differences between the different networks.

Hence a more in-depth FSLE analysis has been carried out. FSLE spectra are shown in figures 8 and 9. From FSLE analysis it results that gan-trained models better reproduce dynamics across the scale range. Observe that FSLE performances are consistent across the two different time periods, indicating that the network is capable of learning the task on a time period and reproduce it in the other.

Network			**Metrics**		
Type	**Tr.**	**Inf.**	**ASV [km/day]**	**Liu index [km]**	**RMSE [km]**
gan	8	8	10.44	0.392 ± 0.068	6.09 ± 0.46
gan	16	8	10.62	0.400 ± 0.075	6.02 ± 0.46
nogan	8	8	8.65	0.383 ± 0.072	5.97 ± 0.48
nogan	16	8	10.52	0.401 ± 0.063	6.04 ± 0.46
gan	8	16	3.34	0.204 ± 0.031	11.24 ± 0.52
gan	16	16	3.32	0.219 ± 0.036	12.21 ± 0.51
nogan	8	16	3.36	0.196 ± 0.032	11.01 ± 0.56
nogan	16	16	3.39	0.211 ± 0.032	12.11 ± 0.53
gan	8	48	0.89	0.096 ± 0.015	30.51 ± 1.03
gan	16	48	0.85	0.089 ± 0.011	28.54 ± 0.93
nogan	8	48	0.92	0.094 ± 0.016	29.78 ± 1.02
nogan	16	48	0.88	0.086 ± 0.012	28.61 ± 0.90

Table 1: Metrics for the time period 1st - 13th January.

Network			**Metrics**		
Type	**Tr.**	**Inf.**	**ASV [km/day]**	**Liu index [km]**	**RMSE [km]**

gan	8	8	10.63	0.252 ± 0.046	5.89 ± 0.39
gan	16	8	8.91	0.261 ± 0.059	5.81 ± 0.42
nogan	8	8	8.93	0.241 ± 0.057	5.79 ± 0.45
nogan	16	8	9.38	0.267 ± 0.069	5.81 ± 0.39
gan	8	16	3.27	0.158 ± 0.019	11.19 ± 0.48
gan	16	16	3.20	0.164 ± 0.027	12.03 ± 0.50
nogan	8	16	3.22	0.151 ± 0.019	10.84 ± 0.49
nogan	16	16	3.39	0.162 ± 0.023	12.02 ± 0.48
gan	8	48	0.92	0.087 ± 0.015	31.22 ± 0.89
gan	16	48	0.84	0.078 ± 0.011	28.83 ± 0.89
nogan	8	48	0.91	0.082 ± 0.012	29.48 ± 0.88
nogan	16	48	0.85	0.081 ± 0.015	29.99 ± 0.80

Table 2: Metrics for the time period 21st May - 3rd June.

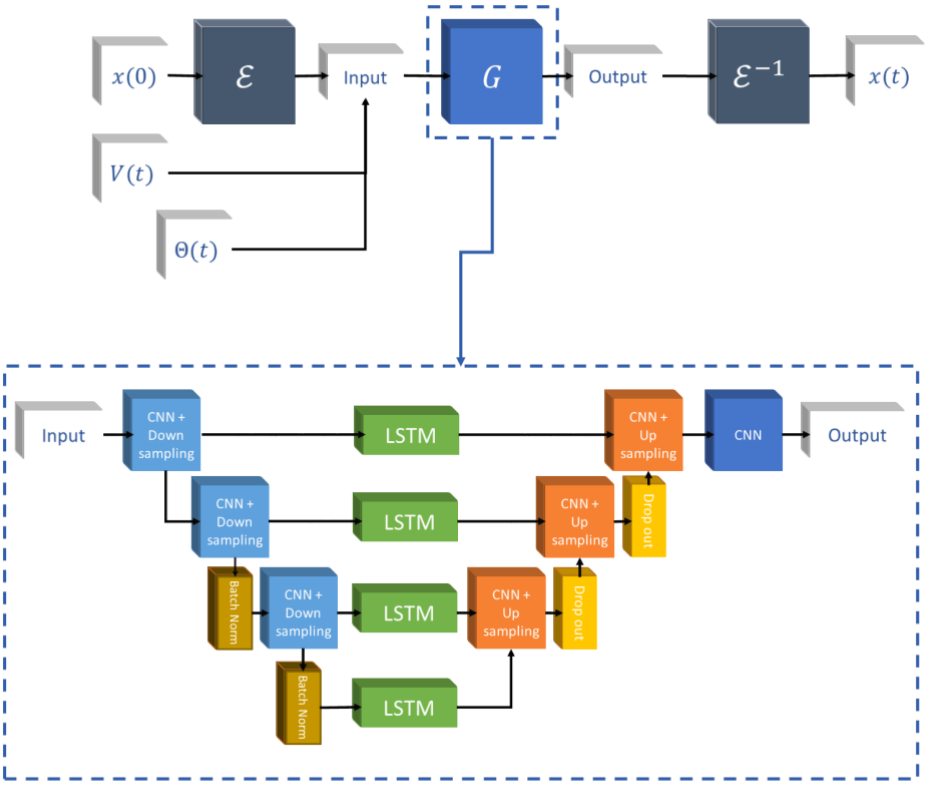


Figure 5. Architecture overview

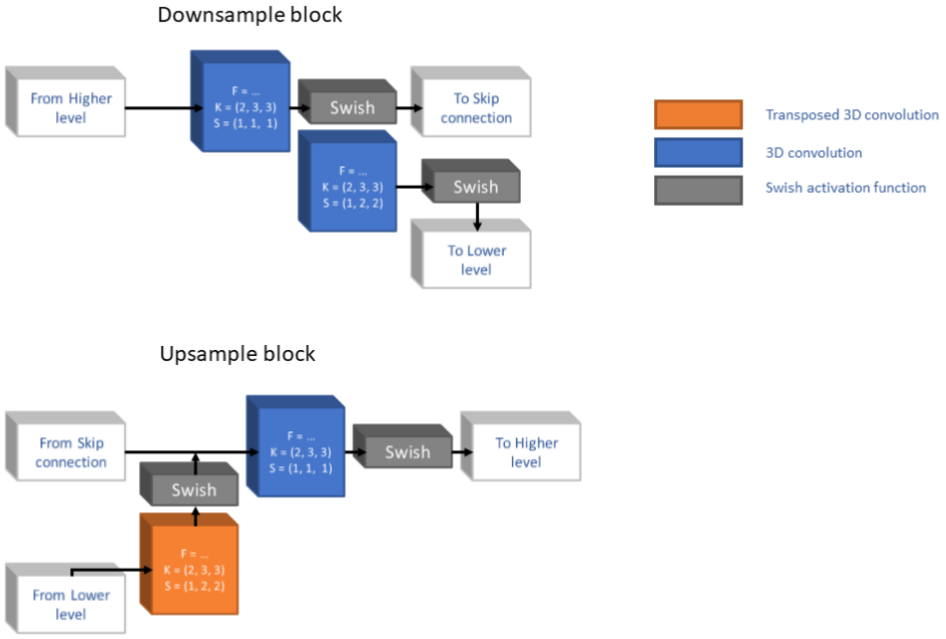


Figure 6. Details of convolutional blocks

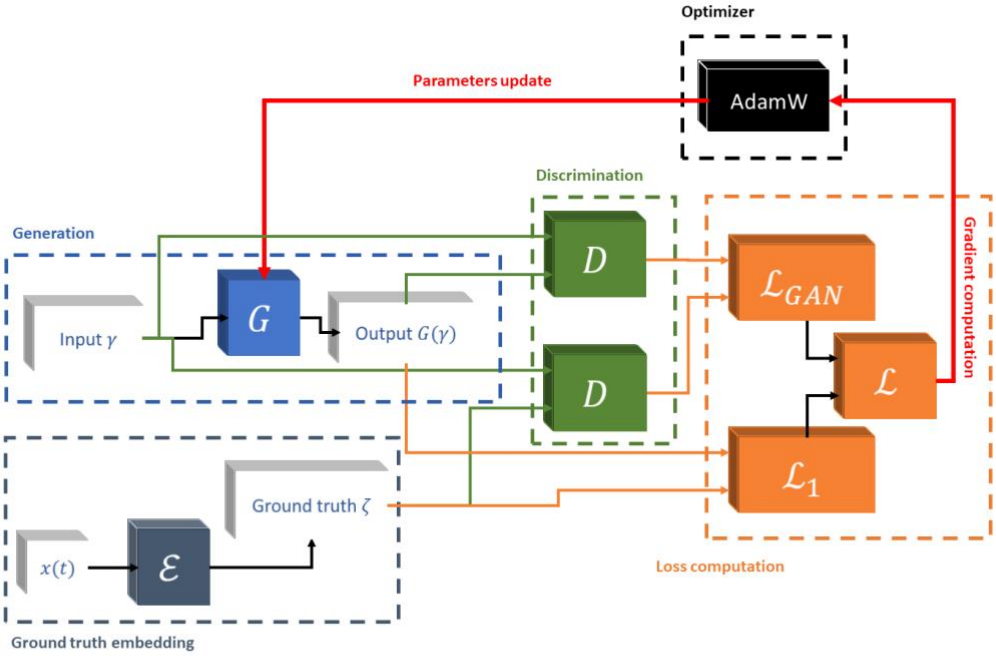


Figure 7. Training setup

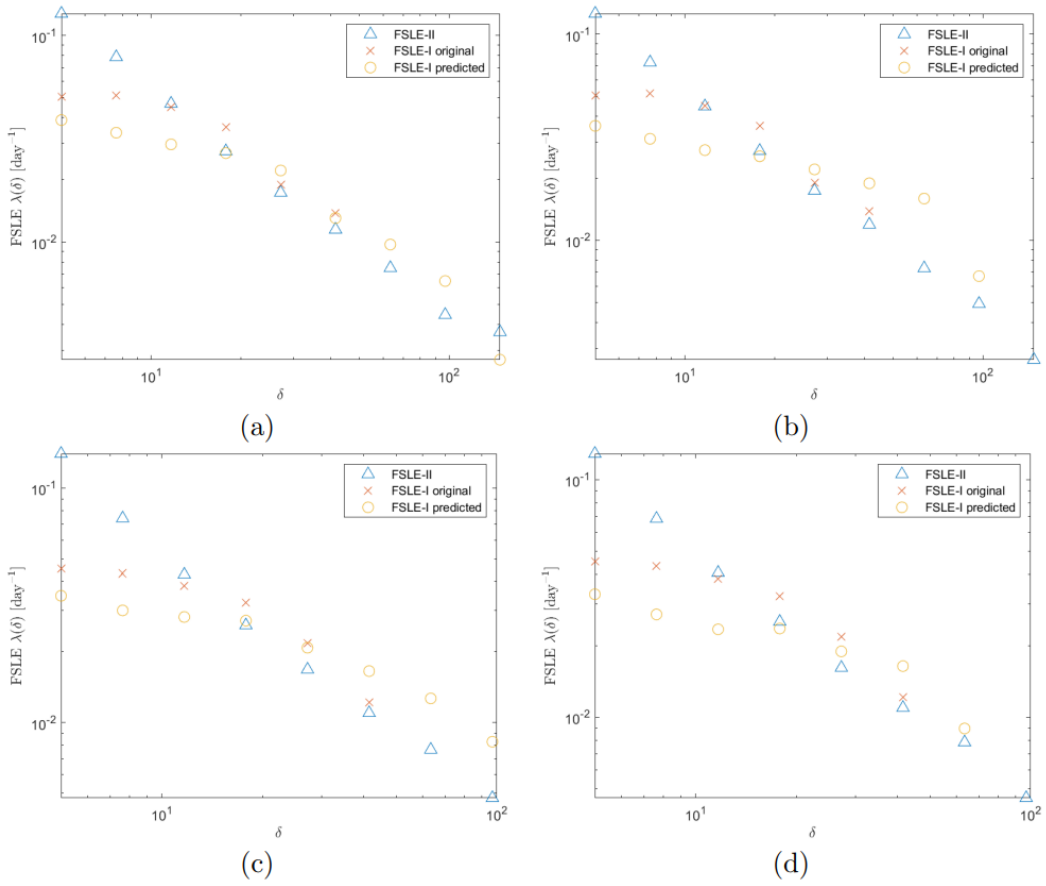


Figure 8 First and second type FSLE spectra for the gan and non-gan models trained for 8 time steps and inferred for 48 time steps. Upper panels show the spectra for gan (a) and non-gan (b) training setups referring to the first time period (1st - 13th January). Lower panels show the spectra for gan (a) and non-gan (b) training setups referring to the second time period (21st May - 3rd June)

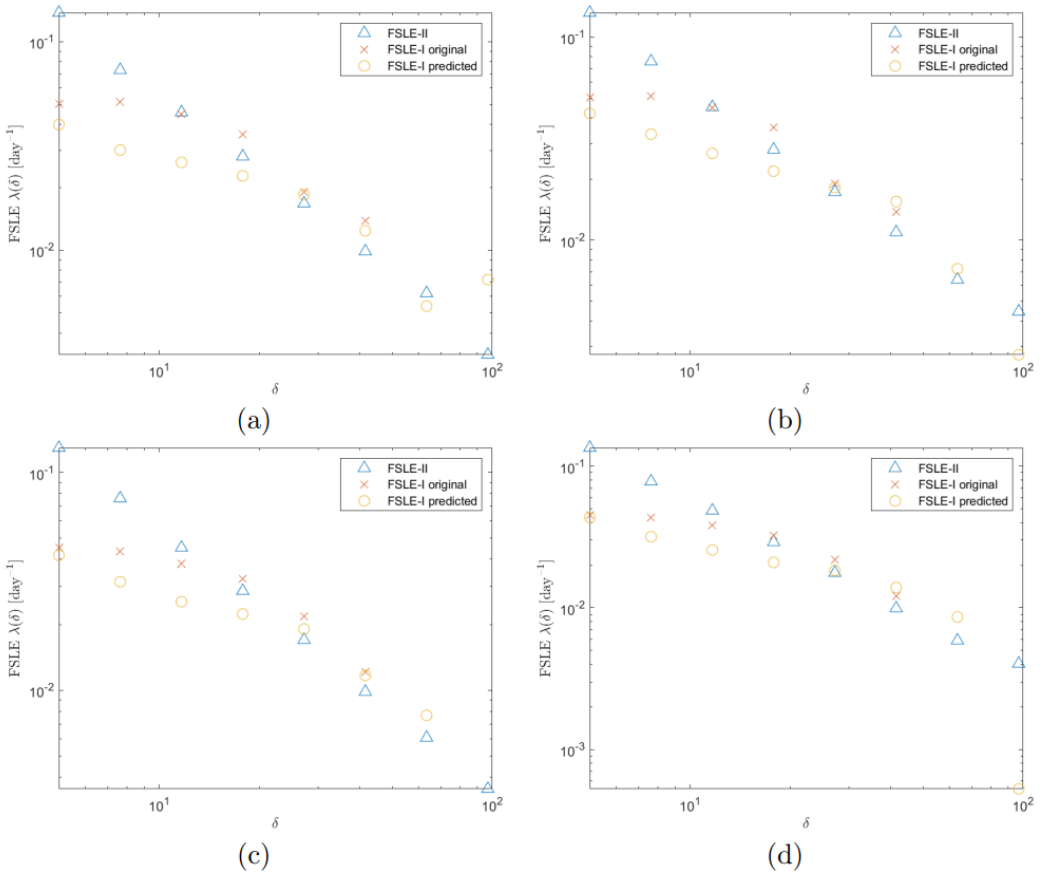


Figure 9 First and second type FSLE spectra for the gan and non-gan models trained for 16 time steps and inferred for 48 time steps. Upper panels show the spectra for gan (a) and non-gan (b) training setups referring to the first time period (1st - 13th January). Lower panels show the spectra for gan (a) and non-gan (b) training setups referring to the second time period (21st May - 3rd June)

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.

[3a] 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.

[3b] Wong, A., Keeley, R., & Carval, T. (2024). Argo quality control manual for CTD and trajectory data.

[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.

[17] Liu, Y & Weisberg, R. H. (2011) Evaluation of trajectory modeling in different dynamic regions using normalized cumulative lagrangian separation. *Journal of Geo-physical Research: Oceans*, 116(C9).

[18] Lacorata, G., Corrado, R., Falcini, F. & Santoleri, R. (2019) Fsle analysis and validation of lagrangian simulations based on satellite-derived globcurrent velocity data. *Remote Sensing of Environment*, 221:136–143.

[19] Buongiorno Nardelli, B., Cavaliere, D., Charles, E., & Ciani, D. (2022). Super-resolving ocean dynamics from space with computer vision algorithms. Remote Sensing, 14(5), 1159.

[20] Rio, M. H., & Santoleri, R. (2018). Improved global surface currents from the merging of altimetry and sea surface temperature data. Remote sensing of Environment, 216, 770-785.