GANs for integration of deterministic model and observations in marine ecosystem

Gloria Pietropolli[1,2], Gianpiero Cossarini[2], and Luca Manzoni[1]

[1] Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste
Via Alfonso Valerio 12/1, 34127 Trieste, Italy
gloria.pietropolli@phd.units.it, lmanzoni@units.it
[2] National Institute of Oceanography and Applied Geophysics - OGS
Borgo Grotta Gigante 42/c, 34010 Sgonico, Trieste, Italy
gcossarini@ogs.it

Abstract. Monitoring the marine ecosystem can be done via observations (either in-situ or satellite) and via deterministic models. However, each of these methods has some drawbacks: observations can be accurate but insufficient in terms of temporal and spatial coverage, while deterministic models cover the whole marine ecosystem but can be inaccurate. This work aims at developing a deep learning model to reproduce the biogeochemical variables in the Mediterranean Sea, integrating observations and the output of an existing deterministic model of the marine ecosystem. In particular, two deep learning architectures will be proposed and tested: first *EmuMed*, an emulator of the deterministic model, and then *InpMed*, which consists of an improvement of the latter by the addition of information provided by in-situ and satellite observations. Results show that *EmuMed* can successfully reproduce the output of the deterministic model, while *ImpMed* can successfully make use of the additional information provided, thus improving our ability to monitor the biogeochemical variables in the Mediterranean Sea.

1 Introduction

Improving the capability of monitoring and forecasting the status of the marine ecosystem has important implications (e.g. sustainable approaches to fishing and aquaculture, mitigation of pollution, and eutrophication), especially considering the changes caused by human activities [6]. An unprecedented improvement in monitoring the oceans has arisen from satellite sensors in the 90s and in situ autonomous oceanographic instruments, such as *float* in the 2000s. Floats consist of a two meters long robotic device, that collects marine variable data by diving in the ocean and varying its depth; for more details see the GOOS (Global Ocean Observing System) website [1]. While these instruments do not need human intervention and provide profiles while the battery lasts, however, they are expensive and thus perform relatively few measurements compared to the whole area to cover (Figure 1 shows the distribution of the float measurement collected during 2015 over the entire Mediterranean sea), which, consequently, cannot

be modeled by only relying on these observations. Satellites cover with high resolution the whole marine domain but only at the surface and they suffer from cloud cover. Hence, observational data available are largely spatially sparse, and with a scarcity of series spanning more than a few decades. Deterministic models have been exploited to simulate the marine environment, as they can provide reanalyses and predictions for the whole 3D domain. However, uncertainties in parameterization and input data and high computational costs can impact their reliability and applicability. The current state-of-the-art deterministic marine ecosystem modeling merges observations (e.g. satellite ocean color, BGC argo float, and so on) with ocean model through data assimilation methods [7]. The incorporation of machine learning (ML) techniques offers alternative and stimulant opportunities for advancing the capacity of integrating theory, knowledge and observations to simulate the marine environment [13]. That is, ML is a new way, compared to existing data assimilation methods, of integrating observations and theory. The present work aims to develop a novel deep learning approach to assess spatial and temporal variability of physical and biogeochemical variables in the marine domains, that combines the knowledge provided by the deterministic model and the in-situ and satellite observations. Embedding of ML techniques to physical and biogeochemical oceanography received significant attention in recent years [12], for a comprehensive review of the current state of the art of ML application to this field the reader can refer to [13].

The deep learning method proposed in this work is based on the approach of filling missing pixels of a considered image, which is a well-known and extensively studied computer vision task, often referred to as *image inpainting* [3]. Since this method has been created specifically to synthesize visually realistic, coherent, and semantic plausible pixels for missing regions, our idea was to exploit its architecture to assemble a model capable of skilfully reconstructing the physical and biogeochemical variables and also to fill the information gap provoked by the inhomogeneity of in-situ observation. This novel approach has been implemented in the Mediterranean Sea, a semi-enclosed sea where a rich collection of model, satellite, and in-situ data are already available: a validated model [5], high-resolution satellite data from Copernicus [4] and in-situ BGC-Argo floats [14]. The first ML model that we will introduce, named *EmuMed*, exploits Generative Adversarial Networks (GAN) [8], and is based on an inpainting architecture [9]. *EmuMed* learns spatial and temporal relationship among the marine ecosystem variables starting from the deterministic model *MedBFM* output thanks to the nature of its architecture. The second ML model, that we define is *InpMed*, adds observations to *EmuMed* while maintaining the same architecture of *EmuMed*. We remark that modeling marine ecosystem variables by ML presents several challenges. First of all, marine datasets span four dimensions (i.e., temporal, vertical and two horizontal) which are characterized by different scales and units (e.g., kilometer and meter respectively for horizontal and vertical spatial dimensions). Moreover, unlike many ML applications, in geosciences we cannot rely on ground-truth data. Indeed, the deterministic model is just an approximation itself of the marine ecosystems, with the observations providing only a very sparse and scarce picture

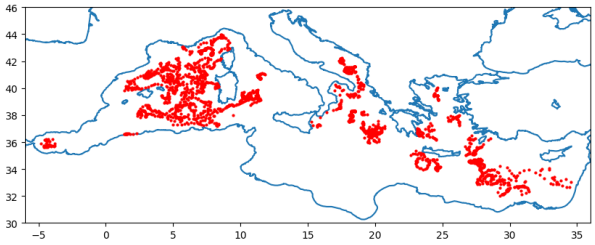


Fig. 1. Map of the *float* measurements over the Mediterranean Sea collected during the year 2015.

of it. These motivations encourage us to select a convolutional-based architecture, as it is naturally suitable for dealing with spatial data. The main idea was to treat horizontal maps of the considered domain as images that capture the marine environment as if it were photography, where the classical RGB channels are substituted with channels representing the marine variables. Indeed, in images, the three colors channels are strongly interrelated and dependent on each other, as they need to collaborate to produce a whole range of colors. Similarly, we aim to introduce an intrinsic strong relation between marine ecosystem variables as they are also naturally correlated.Then, considering that dealing with in-situ measurements leads also to the aforementioned problem of the insufficient spatial coverage of information, an architecture capable of filling areas where measurements are missing becomes essential. These considerations lead us to choose as learning architecture GAN specifically constructed to deal with inpainting tasks to deal with horizontal sections of the marine domain.

The paper is structured as follows: Section 2 provides a description of the proposed models. In particular, Section 2.1 introduces the deep learning architecture, Section 2.2 defines and describes *EmuMed*, while Section 2.3 illustrates *InpMed*. In Section 3 the experimental settings are provided and experimental results are reported in Section 4. Section 5 recalls the main contributions of the paper and provides directions for further research.

2 Material and Method

In this section, we introduce the deep learning architecture employed. In Section 2.2 we discuss an intermediary version of the method we build: *EmuMed*. Finally, a final and improved model, *InpMed*, will be presented and discussed in Section 2.3.

2.1 Deep Learning Architecture

The deep learning architecture employed will take advantage of Convolutional Neural Network (CNN) [10]. CNN performs proficiently in machine learning problems dealing with multiple dimensional input domains, such as image data,

since they conserve the spatial structure of the input; for further details, the reader can refer to [2]. The models introduced in this work are based on a convolutional inpainting architecture [9], which is in turn based on Generative Adversarial Networks (GAN) [8]. The original purpose of GAN is to train the generative model by using an auxiliary network, called discriminator, which serves to distinguish real images with respect to the one generated by the generative model. The general inpainting architecture consists of the training of a generative network to "fill-in" in the most realistic way possible an image with one (or even more) parts of it masked. In this paper, we will consider an inpainting model composed of three interacting convolutional neural networks: the *completion network* used to complete the image; the *global discriminator*, and the *local discriminator*, which are two auxiliary networks. The completion and the discriminators compete in a two-player game, where simultaneous improvements are made to both of them during the training phase. Thus, while the completion network learns how to fill the holes in a realistic and coherent way, discriminators are trained to understand whether or not the provided input has been completed. The improvement of the completion implies a betterment of the discriminators' performance; and vice-versa, the improvement of the discriminators' capability to recognize completed input implies a rise in the completion performance, to fool the discriminators.

Completion Network. The completion network is a convolutional neural network, consisting of 17 layers, as detailed in [9]. The architecture exploits an encoder-decoder technique that initially decreases the resolution of the input features to reduce the computational effort, and then restores the original resolution. Like in image generation task, the input of the completion network is an RGB image with binary channels, where 1 indicates that a mask is applied to the input pixel, and the output is an RGB image, properly completed.

Discriminator Networks. Two discriminators play against the completion network introduced above: the global discriminator and the local discriminator. The former tests the reliability of the input in its entirety, while the latter focuses on a particular and smaller area, thus paying more attention to details. The discriminators take as input the complete image (adequately re-scaled), both of them are implemented using convolutional neural networks followed by a fully-connected layer producing a real-valued vector as output. Finally, the two resulting vectors are concatenated and passed again as input of a fully-convolutional layer, that returns a continuous value indicating the probability that the provided input is real or fake.

Training. The loss function employed to train the completion network, introduced in [11], is the weighted MSE defined as follows:

$$L(x, M_c) = ||M_c \odot (C(x, M_c) - x)|| \tag{1}$$

where $\odot$ stands for the pixel-wise multiplication and $||\cdot||$ is the Euclidean norm. Furthermore, the GAN loss [8] is used for training together completion and

discriminators network. While discriminators aim to maximize the average of the log-probability of real images and the log of the inverse probability for fake images, the generator aims to minimize the log of the inverse probability predicted by the discriminator for fake images. Therefore, the generator tries to minimize the following function while the discriminator tries to maximize it:

$$\min_C \max_D \mathbf{E}[\log D(x, M_d) + log(1 - D(C(x, M_c), M_c))] \tag{2}$$

where M_c is the input mask, M_d is a random mask, $D(x, M_d)$ is the discriminator's estimate of the probability for the real input x with mask M_d to be real, $D(C(x, M_c))$ is the discriminator's estimate of the probability for the fake input x to be real, and $\mathbf{E}$ indicate the average over the training input. Finally, taking in account both Equation 1 and Equation 2, the resulting loss function is:

$$\min_C \max_D \mathbf{E}[L(x, M_c) + \log D(x, M_d) + \alpha log(1 - D(C(x, M_c), M_c))] \tag{3}$$

where α is a fixed hyperparameter. The training of the algorithm, which is schematized in Algorithm 1, consists into three main phases: during *phase 1* the completion network is trained among all the features of the training set for T_C epochs; then, during *phase 2*, the completion network is fixed and the discriminator network is trained for T_D epochs; finally, during *phase 3* both the completion network and the discriminators are trained at the same time for T_{CD} epochs.

2.2 EmuMed

The *EmuMed* is the first model that we present in this paper, named after the fact that it behaves as an emulator (meaning that it is learning information from) of the deterministic model *MedBFM* [15,5]. The architecture underlying *EmuMed* is the one presented in Section 2.1: a generative convolutional neural network trained through adversarial loss. The input (tensors) employed for the training are obtained from a discretization of data generated through a simulation of the deterministic model. These tensors represent 2-dimensional maps of a fixed region of the Mediterranean Sea (here, with the term map we denote a horizontal section of the region at a fixed depth). The role that pixels accomplished for the image completion task is carried out by rectangles that represent a discretization area of the Mediterranean Sea, while the standard RGB channels are substituted with channels that contain values representing the oceanographic physical and the biogeochemical variables that we aim to reproduce. Thus, *EmuMed* consists of a generative model capable of reconstructing the biological and chemical interactions for the whole Mediterranean Sea domain considered.

2.3 InpMed

InpMed is the second model presented in this paper, obtained starting from *EmuMed* and then performing a further training phase adding both in-situ

Algorithm 1 Pseudocode of the training steps for the deep learning architecture underlying *EmuMed*, and, consequently *InpMed*.

for $t = 0 \dots T_C$ **do** ▷ phase 1
  for all x in the training set **do**
    Generate masks M_c with random holes.
    Compute $C(x, M_c)$.
    Update completion network weights through Equation 1.
  end for
end for
for $t = 0 \dots T_D$ **do** ▷ phase 2
  for all x in the training set **do**
    Generate masks M_c with random holes.
    Compute $D(x, M_d)$ and $D(C(x, M_c), M_c)$.
    Update discriminator network weights through binary cross entropy loss
  end for
end for
for $t = 0 \dots T_{CD}$ **do** ▷ phase 3
  for all x in the training set **do**
    Generate masks M_c with random holes.
    Generate masks M_d with random holes.
    Compute $D(x, M_d)$ and $D(C(x, M_c), M_c)$.
    Update discriminator network weights through binary cross entropy loss.
    Update completion network weights through Equation 3.
  end for
end for

measurements collected by the float devices and by satellite observations. This additional training phase is performed according, again, to phase 1 of the Algorithm 1 described in Section 2.1. The weights of *EmuMed* are updated to fit these new data, producing a more reliable prediction that is closer to the real marine ecosystem conformation. *InpMed* ensures an improvement in the simulating capability of the model, as the convolutional structure guarantees a local distribution of information provided by observation also in neighboring areas of these measurements. Another crucial point is that there are certain marine indicators that are not measured either through in-situ devices or via satellite information, such as the primary production, which prediction can be improved anyway by taking advantage of a combination of the ML architecture and of the observed data. In fact, relations between variables are learned through the training of *EmuMed*; subsequently, *InpMed* exploits the information provided by the measured variables and the relations learned from the deterministic model to improve the prediction for both the measured variables and the ones that cannot be measured.

3 Experimental Setting

The geographical area considered in this work is the western Mediterranean portion, specifically, the one with latitude ranging between 36 and 44 and

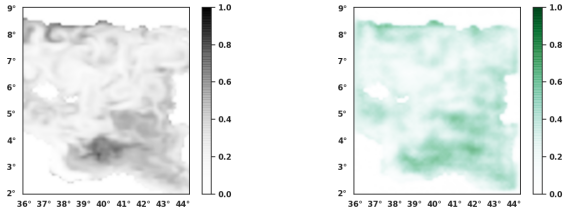
Table 1. Experimental settings values. The table on the left describes the parameters concerning the deep learning architecture. Horizontal lines separate: common parameters, *EmuMed* parameters and *InpMed* parameters. The table on the right represents the parameters referring to the definition of the input structure.

Parameter	**Value**
Comp. input size	$30 \times 65 \times 75$
Loc. Disc. input size	$20 \times 50 \times 50$
lr_c	0.01
lr_d	0.01
T_c	5000
T_d	200
T_{cd}	1000
α	4×10^{-4}
lr_{float}	0.001
T_{float}	200

Parameter	**Value**
time interval	weekly
latitude interval	$36^\circ - 44^\circ$
longitude interval	$2^\circ - 9^\circ$
depth interval	$0 - 600$ m
time resolution	weekly
latitude resolution	12 km
longitude resolution	12 km
depth resolution	20 m

longitude varying between 2 and 9 and the vertical dimension covers a depth ranging from 0 to 600 meters. It consists of the portion between southern France and northern Africa, delimited on the east limit by Corsica and Sardinia and on the west limit by Balearic islands. The spatial resolution is 12 km in both latitude and longitude axes and 20 meters in the vertical one. The time period covers the year 2015 which is discretized on a weekly basis. Therefore, the 4-dimensional input tensor consists of a horizontal map, of the central-western Mediterranean Sea area, whose dimensions are: length, height, width, and channel. Each channel of the tensor, in turn, collects one marine ecosystem variable. Namely, the variable considered are temperature, salinity, oxygen, chlorophyll-a, and primary production. All the variables can be obtained via the deterministic model *MedBFM*, while only the first four are collected via float measurement (as it is not possible to measure primary production via any sensor), and only chlorophyll-a can be inferred through satellite. Each location in the 3D field has 5 variables associated and a time resolution one week is used (thus, 52 weekly observations are available). Due to their nature, each float provides a 1D profile, where latitude and longitude are fixed and only the depth can change. Finally, since satellites can observe only the surface of the water, they provide 2D data (with holes due to cloud covers). For the training of *EmuMed*, we used the following hyper-parameters (summarized in Table 1): the completion network is trained for 5000 epochs, the discriminator network is trained for 200 epochs, and finally the two networks are trained simultaneously for 1000 epochs. These hyperparameters have been fixed after an appropriate preliminary study. The optimizer chosen is ADADELTA [16], which set the learning rate for each weight in the network automatically. The learning rate initial value for both the completion and the discriminator is set to 0.01. Subsequently, *InpMed* is trained for 200 epochs with a learning rate value set initially to 0.01.

4 Experimental Results



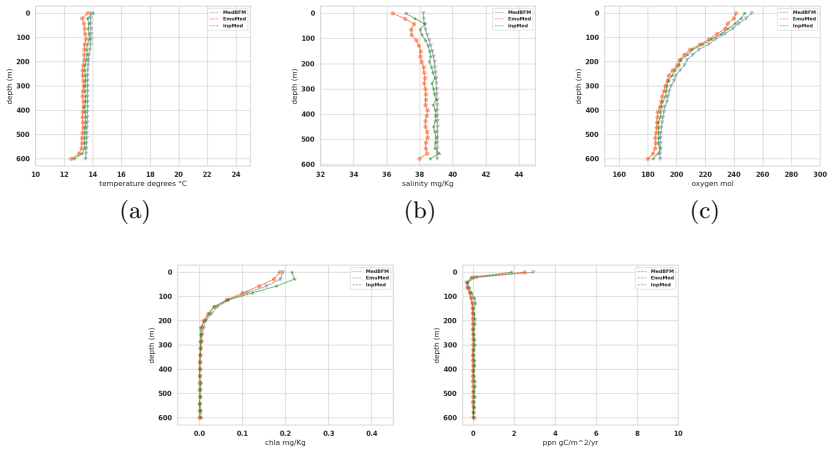
(a) (b)

Fig. 2. Map of surface chlorophyll produced by (a) *MedBFM* and reconstructed using (b) *EmuMed*.

The Mediterranean case study aims at demonstrating the reliability of ML reconstruction for marine ecosystem variables and at showing how the different components of the ML architecture contribute to the reconstruction quality. To assess the goodness of the proposed reconstruction, the analysis focus on some statistical properties (averages and variances) of the simulated fields. In particular, Figure 2 reports maps of one of the variables (i.e., surface chlorophyll) demonstrating *InpMed* capability to emulate the intense spatial variability of surface marine fields. Figure 3 shows the vertical profiles of the spatial averages among two given weeks of the year (e.g., one in winter and one in summer), assessing the capability of the technique to simulate different seasonal periods for all modeled variables. These plots compare the original deterministic model *MedBFM* profile with the *EmuMed* and *InpMed* reconstructions, showing the benefits provided by the different architectural components. Finally, Figure 4 compares, via box-plot, the distributions of the standard deviation of the spatial variability of *MedBFM* and *InpMed*. Four weeks of the year are displayed in order to study how the spatial heterogeneity of the marine proprieties varies throughout the year and how it is handled by the different models.

Results show that the *EmuMed* has learned well to reproduce the typical mean vertical profiles, simulated by *MedBFM*, for all variables but salinity (Figure 3). Deviations of *InpMed* from *EmuMed* profiles (e.g., orange and green lines in oxygen, chlorophyll, and primary production in Figure 3) highlight how the inclusion of the observations in the ML architecture introduced possible corrections (i.e., new information) to the *MedBFM* simulated fields. Temperature shows that the inclusion of observations had a marginal effect while salinity shows that observations bring *InpMed* profiles closer to *MedBFM* highlighting a possible inaccurate reconstruction of *EmuMed*, that anyway is corrected in the second phase of the training, confirming the added values of the two-step architecture implemented in *InpMed*. Regarding the spatial variability of horizontal fields of marine variables, maps of Figure 2 show qualitatively the good performance of the ML reconstruction. From a quantitative point of view, the comparison of

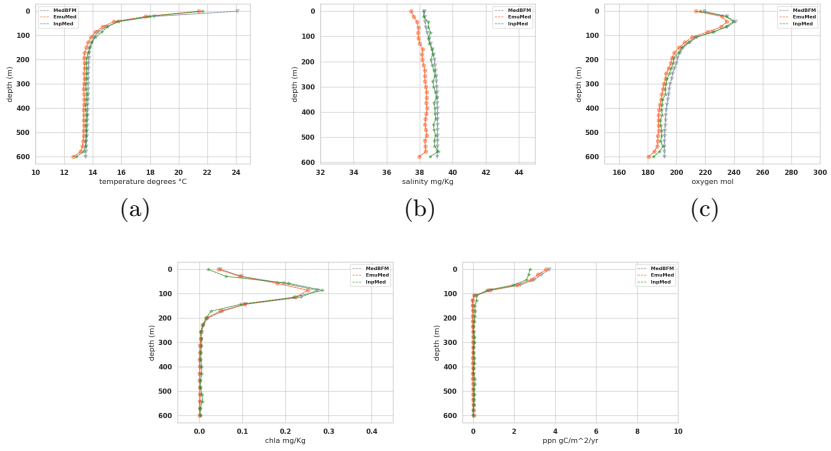
Week 2



(a) (b) (c)

(d) (e)

Week 35



(a) (b) (c)

(d) (e)

Fig. 3. Vertical profile of the spatial averages, varying with depth, over the considered domain. Variables represented are: (a) temperature, (b) salinity, (c) oxygen, (d) chlorophyll, (e) primary production. The gray line represents the deterministic vertical profile *MedBFM*, the orange line represents the one inferred by *EmuMed*, and the green represents the vertical profile predicted by *InpMed*. Above are reported results relative to week 2 (winter), on the right relative to week 35 (summer), in order to demonstrate the capability of introduced models to predict different seasonal periods.

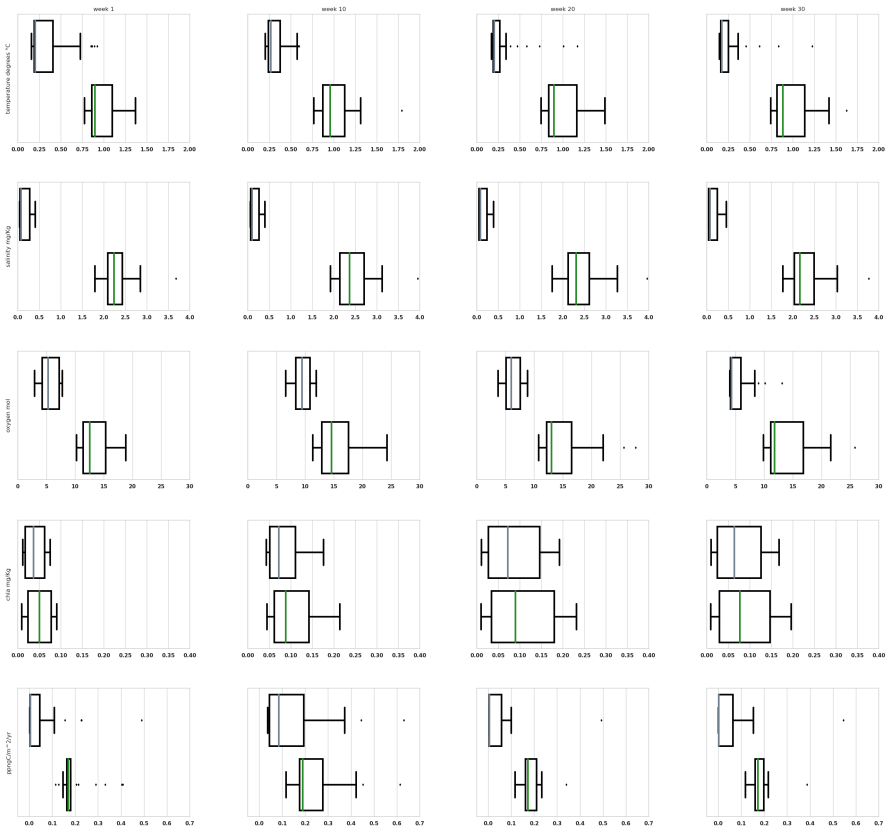


Fig. 4. Box-plots showing the distributions of the standard deviation of the marine variables computed from the spatial maps at different depths. From top to bottom are shown, respectively, temperature, salinity, oxygen, chlorophyll (only layers between 0 and $200m$), primary production (only layers between 0 and $200m$). The box-plot with the median gray line represents deterministic model *MedBFM*, while the box-plot with the green median line represents *InpMed*. From left to right are shown, respectively, week 1, week 10, week 20 and week 30

the standard deviation boxplots (Figure 4) shows that the spatial variability of *InpMed* is generally higher than *MedBFM* for all variables in all selected weeks. This highlights that, when observations are included in the reconstruction, the ML model *InpMed* simulates horizontal fields characterized by more complicated gradients and spatial structures w.r.t. a possibly too smooth output of the deterministic model.

A separate comment can be done for primary production (i.e., a variable that is not observed). Despite the mean vertical profiles are not substantially changed by ML architecture (Figure 3), it is possible to notice that the *InpMed* model differs from the *EmuMed* even if, during the training, no observed data have been provided for primary production. The fact that the variability introduced by the observed variable is clearly propagated by *InpMed* can be also observed in Figure 4. This evidence confirms that information provided by observed data improves the *InpMed* capability to simulate the unobserved variable, thanks to the relations among variables learned from the output of the deterministic model by the deep learning architecture exploited.

5 Conclusion

We investigated the integration of an existing ecosystem deterministic model with in-situ and satellite information through a convolutional generative deep learning architecture. Merging these two different kinds of information allows us to combine their strengths and exploit them to lessen each other's limits. We remark that a deep learning model can also be less computationally expensive, once trained, with respect to a deterministic one, as it does not require an entire simulation in order to get specific variable estimations (e.g., primary production). Such a comparison will be one of the aspects that will be investigated in future works. Moreover, exploiting the intrinsic structure of the architecture, the learning framework makes possible the spread of information provided from the observed variables (temperature, salinity, oxygen, chlorophyll) also to variables that are not possible to directly collect via in-situ or satellite measurements (e.g., primary production). Experimental results on both *EmuMed* and *ImpMed* have confirmed the validity of the proposed approach, showing that our models can infer correctly information from the deterministic model and, in the case of *ImpMed*, also from observations. This work represents the first step to exploiting deep learning architecture aimed at merging large deterministic model output with observations to reconstruct the marine ecosystem's temporal and spatial variability. Our main goal will be to extend this architecture by inserting a larger number of channels so that it became able to reproduce the whole set of marine ecosystem variables, in particular exploiting this architecture to model unobserved variables (as we did for primary production) also with the information provided by observed data. The extension of the present ML model to the entire Mediterranean Sea represents another important computational challenge given the significant increase in data volume to handle.

References

1. The global ocean observing system. https://www.goosocean.org/, accessed: 2022-03-22
2. Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET). pp. 1–6 (2017). https://doi.org/10.1109/ICEngTechnol.2017.8308186
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 417–424 (2000)
4. Colella, S., Falcini, F., Rinaldi, E., Sammartino, M., Santoleri, R.: Mediterranean ocean colour chlorophyll trends. PloS one **11**(6), e0155756 (2016)
5. Cossarini, G., Feudale, L., Teruzzi, A., Bolzon, G., Coidessa, G., Solidoro, C., Di Biagio, V., Amadio, C., Lazzari, P., Brosich, A., et al.: High-resolution reanalysis of the mediterranean sea biogeochemistry (1999–2019). Frontiers in Marine Science p. 1537 (2021)
6. Euzen, A., Gaill, F., Lacroix, D., Cury, O.: The ocean revealed (2017)
7. Fennel, K., Gehlen, M., Brasseur, P., Brown, C.W., Ciavatta, S., Cossarini, G., Crise, A., Edwards, C.A., Ford, D., Friedrichs, M.A., et al.: Advancing marine biogeochemical and ecosystem reanalyses and forecasts as tools for monitoring and managing ecosystem health. Frontiers in Marine Science **6**, 89 (2019)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
9. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017)
10. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4), 541–551 (1989)
11. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. CoRR **abs/1604.07379** (2016), http://arxiv.org/abs/1604.07379
12. Sauzède, R., Johnson, J., Claustre, H., Camps-Valls, G., Ruescas, A.: Estimation of oceanic particulate organic carbon with machine learning. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences **2**, 949–956 (2020)
13. Sonnewald, M., Lguensat, R., Jones, D.C., Dueben, P., Brajard, J., Balaji, V.: Bridging observations, theory and numerical simulation of the ocean using machine learning. Environmental Research Letters (2021)
14. Teruzzi, A., Bolzon, G., Feudale, L., Cossarini, G.: Deep chlorophyll maximum and nutricline in the mediterranean sea: emerging properties from a multi-platform assimilated biogeochemical model experiment. Biogeosciences **18**(23), 6147–6166 (2021)
15. Teruzzi, A., Di Cerbo, P., Cossarini, G., Pascolo, E., Salon, S.: Parallel implementation of a data assimilation scheme for operational oceanography: The case of the medbfm model system. Computers & Geosciences **124**, 103–114 (2019)
16. Zeiler, M.D.: Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)