



# Applications of deep learning to subgrid eddy parameterization in high- and low-resolution ocean model

by

**YAN Fei Er**

in partial fulfillment of the requirements for the degree of

**Master 2**  
in Physical Oceanography and Climate

at Université de Bretagne Occidentale,  
Friday September 6, 2019 .

Supervisor: Jonathan Gula

Laboratory: Laboratoire d'Océanographie Physique et Spatiale (LOPS)



# Abstract

Mesoscale eddies play a substantial role in the ocean dynamics, which is missing in low-resolution ocean models. We apply deep learning algorithms to train convolutional neural networks to represent small-scale processes, based on sub-sampled output from a realistic model that better resolves the large and small-scale dynamics. We find that this approach can give accurate generalization of spatial and temporal variability of subgrid eddy momentum forcing in both high- and low-resolution. Training by a limited amount of data is sufficient for good performance. This implies a potential way to implement data-driven parameterizations in ocean model in the future.



# Contents

<b>1</b>	<b>Introduction and Objectives</b>	<b>1</b>
<b>2</b>	<b>Data and Methods</b>	<b>5</b>
2.1	High-Resolution Ocean Model . . . . .	5
2.2	Convolutional Neural Network Algorithms and setup . . . . .	7
2.2.1	The architecture of CNNs . . . . .	7
2.2.2	Implementation in details . . . . .	7
2.3	Implementation in coarse grid . . . . .	9
<b>3</b>	<b>Results and Analysis</b>	<b>11</b>
3.1	Comparison of CNNs with BM18 . . . . .	11
3.2	Sensitivity of Neural Networks in realistic simulations . . . . .	13
3.2.1	Non-Local Predictions . . . . .	14
3.2.2	The impact of sub-region size . . . . .	18
3.2.3	The impact of the amount of training data . . . . .	19
3.3	Implementation in coarse-resolution simulation . . . . .	20
<b>4</b>	<b>Conclusion</b>	<b>23</b>
	<b>Bibliography</b>	<b>25</b>



# 1

## Introduction and Objectives

Mesoscale oceanic eddies are energetic and ubiquitous, with typical horizontal scales of 10-100 km and timescales on the order of a month. Their energy generally exceed that of the mean flow by an order of magnitude or more (e.g [Wyrtki et al., 1976], [Ferrari and Wunsch, 2009] ). Mesoscale eddies transport and distribute water mass and properties such as heat, salt and carbon around the ocean. This transport is comparable in magnitude to that of the large-scale wind- and thermohaline-driven circulation [Zhang et al., 2014]. The role of mesoscale eddies in the general ocean circulation has been studied over the past decades, both in idealized Quasi-Geostrophic (QG) models and more realistic simulations. Large-scale oceanic currents are driven by the mesoscale eddies. The interactions of oceanic turbulence at small-scales play an important role in maintaining the large-scale ocean circulation [Greatbatch et al., 2010]. In addition, several studies have suggested the impact of mesoscale eddies in the ventilation of the thermocline circulation (e.g [Lachkar et al., 2009]). Therefore, the representation of mesoscale eddies is very important in physical oceanography.

However mesoscale eddies usually missed in low-resolution ocean models (typically  $1^\circ$ ,  $\approx 110$  km), which too coarse for eddies to form. Modeling at eddy-permitting ( $(1/4)^\circ$ ,  $\approx 30$  km) or eddy-resolving ( $< (0.1)^\circ$ ) scales is therefore alternative way to represent eddy effects. According to [Hallberg, 2013], the dominant length-scale of mesoscale eddies varies greatly with latitude, stratification and ocean depth. Despite significantly increased capability of modern computing systems, ocean general circulation models (OGCMs) can not well represent all relevant eddy scales on a numerical grid without an inordinate amount of computational resources. Moreover it is challenging at present to run eddy-resolving resolutions at global scale for time scales given its huge computational cost. Hence it is important to have a good parameterization of the effects of the unsolved eddies.

In past decades, Laplacian diffusion or a purely adiabatic advection implemented in ocean models as a parameterization for the effects of unresolved eddies. This has been known as Gent-McWilliams (GM) parameterization [Gent and Mcwilliams, 1990; Gent, 2011]. Based on their work, several approaches were introduced. For instance, according to [Hallberg, 2013], common parameterizations of eddy effects via a Laplacian diffusion of the height of isopycnal surfaces (a GM diffusivity as the free parameter) suppress resolved eddies, and do not well represent them. Using a resolution function to regulate parameterizations, He proposed an approach that would work well with the magnitude of the eddy diffusivities in large-scale ocean models. Furthermore [Manz and Zanna, 2014, Zanna et al., 2017] suggested that stochastic parameterizations of eddy-mean flow interaction lead to drastic improvement in ocean state, comparing to deterministic ones. Due to encouraging results and little computational cost,

they have shown great potential in ocean climate models. Unfortunately all these approaches of eddy parameterizations have their own limits. Reducing uncertainties through these approaches to develop a “correct” global parameterization scheme for each ocean process, is hard to implement.

On the other hand, ocean models contain systematic errors because of both physical and numerical inaccuracies. The parameterization of unresolved processes reinforces uncertainties in multi-decades simulations. Therefore, it is always a challenge to improve parameterizations for the unresolved fluxes .

A subset of machine learning in artificial intelligence has networks to extract information from enormous amount of data. Due to its numerical optimization for efficiency, machine learning has been currently explored to parameterize unresolved processes with climate models, such as the Typhoon Forecast [[Jiang et al., 2018](#)], representing unresolved moist convection [[Gentine et al., 2018](#)]. There also have been multiple studies on the application of machine learning in turbulence modeling. They have shown a promising direction for estimating the subgrid forcing quantity in governing equations ( e.g [[Ling et al., 2016](#)] ). It also has been applied as part of the development and improvement of eddy parameterizations recently. [[Bolton and Zanna, 2019](#)] (referred to as BM18) have shown that convolutional neural networks (CNNs) could successfully predict sub-grid and large-scale eddy momentum forcing from a high-resolution QG ocean model. Their work also implied the potential for data-driven eddy parameterizations and for the improvement in numerical simulations in the future.

Following the strategy of BM18, we attempt to train CNNs to represent the relationship between the spatio-temporal variability of the eddy momentum forcing and that of the surface velocity. Therefore we implement the CNNs on more realistic simulation data and also explore the potential way of using machine learning to create new parameterization schemes, by fitting a coarse-grid model to the output of high-resolution models, that more accurately represent the subgrid dynamics.

The Gulf Stream (GS) is one of the most energetic current systems in the world, with intense mesoscale variability. After separating from the western boundaries at Cape Hatteras, the GS flow eastward as a strong jet with large meanders and eddies. Representation of the GS dynamics is important to improving our understanding of the ocean general circulation. The role of mesoscale eddies in GS System has been increasingly investigated during the past decade ( e.g [[Greatbatch et al., 2010](#)] ). And many ocean models (including eddy-resolving models) have been applied in this area. Our study therefore focus on the GS using the simulation data from the ocean model Coastal and Regional Ocean CCommunity model(CROCO) [Gula et al. \[2015\]](#).

A primary goal of this report is to investigate the performance of the CNNs in reproducing the sub-grid eddy momentum forcing from a more realistic model. We evaluate non-local predictions of the BM18 CNNs firstly, and try to modify it to adapt in our cases. Then we estimate the sensibility of the neural networks in different conditions ( Section 3.2 ). The aim of this study is to use sample data generated by an eddy-resolving model, to represent unresolved eddies in low-resolution model with neural networks as a subgrid-scale parametrization. We attempt to augment the performance low-resolution model via this data-driven approach ( Section 3.3 ). The basic information of the ocean model and simulations we used, the CNNs, as well as the

data processing is presented in Chapter 2. The main conclusion and discussions of this report are given in Chapter 4.



# 2

## Data and Methods

### 2.1. High-Resolution Ocean Model

We first attempt to mimic the idealized simulation in BM18. The ocean model used in this study is Coastal and Regional Ocean CCommunity (CROCO) (<http://www.croc-ocean.org>). It is a new generation of the Regional Ocean Modeling System (ROMS) [Shchepetkin and C. McWilliams, 2005], which is specially designed to accurately simulate regional oceanic systems. Here the model has a bounded-square basin with size  $L=3600\text{km}$  and a flat bottom. It has 10 vertical levels from surface to bottom. The simulation represents a wind-driven, double-gyre circulation in mid-latitude ocean. Two gyres are separated by a strong meandering jet from the western boundary. The classic double-gyre flow is known as its dynamics significantly affected by the interactions between the large-scale flow and mesoscale eddies. The model is at an eddy-resolving horizontal resolution of  $\Delta x \approx 7\text{km}$  with  $512 \times 512$  grid points. The wind stress forcing applied on surface is defined as BM18:

$$f(x, y) = \begin{cases} -\tau_0 \frac{0.92\pi}{L\rho_0 H} \sin\left(\frac{\pi y}{g(x)}\right) & y \leq g(x), \\ \tau_0 \frac{2\pi}{0.9L\rho_0 H} \sin\left(\frac{\pi(2y-g(x))}{L-g(x)}\right) & y > g(x). \end{cases}$$

where  $g(x) = L/2 + 0.2(x - L/2)$ ,  $L$  is the length of basin, and  $\rho_0 = 1025\text{kg/m}^3$  is the reference density. The wind strength  $\tau_0 = 0.3\text{N/m}^2$ , and  $H$  is the layer thickness. One snapshot and time-mean of the total velocity stream function at surface are shown in Fig.2.1. Unfortunately the jet does not extend eastwards evidently in our simulation. After the initial spin-up over 1000 days from the rest state to reach statistical equilibrium, the simulation then run for a subsequent 1000 days. 1000 daily outputs of surface zonal and meridional velocity ( $u, v$ ) are stored to train the neural networks.

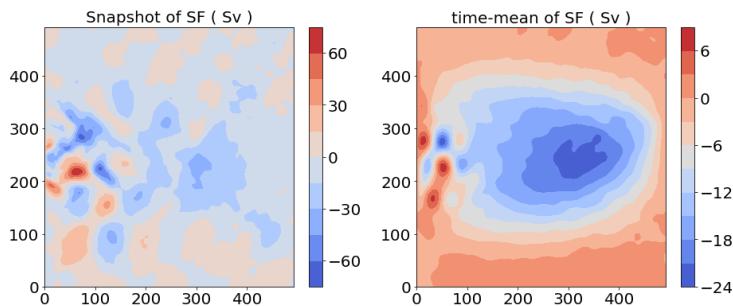


Figure 2.1: The instantaneous ( left ) and time-mean ( right ) total velocity stream function at surface

Another dataset is from a realistic, very high-resolution simulation of the GS region, realized with the ocean model ROMS. The data are from a nested grid covering the whole GS region

( $1600 \times 1000$  grid points) with  $\Delta x \approx 2.5\text{km}$  at 5-day intervals. The effect of the topography is considered and bathymetry of the domain is constructed from the SRTM30\_PLUS dataset (more details of the model in [Gula et al., 2015]). We only extract a smaller region with  $512 \times 512$  grid points ( $L=1280\text{ km}$  in horizontal) shown in Fig.2.2, and 1000 snapshots of  $u, v$  are used in this study.

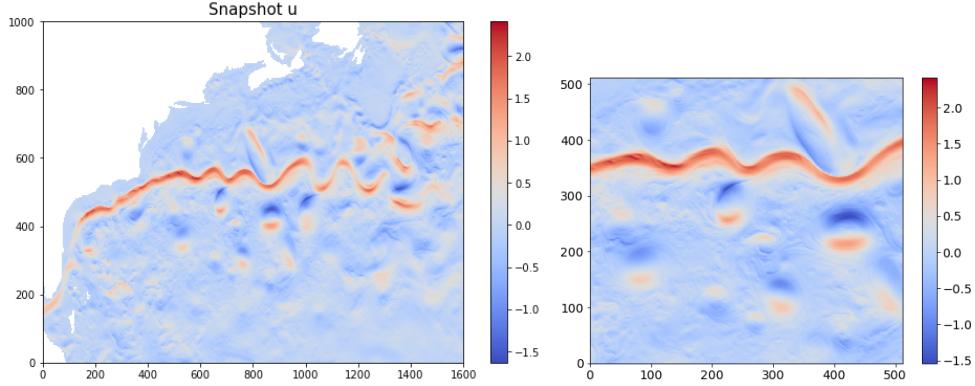


Figure 2.2: The snapshots of zonal component of the surface velocity  $u$  ( m/s ), in the whole domain of the original simulation of [Gula et al., 2015] (left) and a sample of studied region (right).

Following the definitions in BM18 and [Zanna et al., 2017], ignoring vertical effects and planetary vorticity, the horizontal momentum equation is given by:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla) u = F + D \quad (2.1)$$

where  $u$  is the velocity.  $F$  and  $D$  are the momentum of the applied wind stress curl forcing and the dissipation respectively.

We take the approach of the Large-Eddy Simulation to separate velocity field between large-scale and eddy components of the flow ([Berloff, 2005]). The velocity field is split into large and small scales by applying a two-dimensional spatial low-pass Gaussian filter  $G_\Delta(x)$  (Leonard 1974). The filter eliminates scales smaller than the length-scale  $\Delta$ . Here  $\Delta$  is set as  $\approx 30\text{km}$  so that mesoscale eddies information can be kept. Therefore the filtered velocity  $\bar{u}$  (large scale) is obtained by  $G_\Delta * u$ . The eliminated scales are called subgrid-scale (or subfilter)  $u'$ . The full flow is now represented as:

$$u = \bar{u} + u' \quad (2.2)$$

Therefore the filtered momentum equation is given by:

$$\frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla) \bar{u} = \bar{F} + \bar{D} + (\bar{u} \cdot \nabla) \bar{u} - \overline{(u \cdot \nabla) u} \quad (2.3)$$

$$\frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla) \bar{u} = \bar{F} + \bar{D} + S \quad (2.4)$$

where the divergence of a Reynolds stress  $S = (S_x, S_y) = (\bar{u} \cdot \nabla) \bar{u} - \overline{(\bar{u} \cdot \nabla) u}$ , representing the effects of subgrid-scale eddies and their interaction with the large-scale flow. Hence, the effect of the unresolved eddy field is introduced as a forcing term in this resolved-scale momentum equation. Depending on the subgrid variables, the subgrid eddy momentum forcing  $S$  is usually represented with a physical parameterization or closure in ocean models. As

studied in BM18, the machine-learning algorithm based on artificial neural networks (ANNs), is introduced to provide an alternative approach.

## 2.2. Convolutional Neural Network Algorithms and setup

ANNs can approximate any nonlinear deterministic function, a property called the universal approximation theorem [Schmidhuber, 2015]. Machine-learning algorithms have been wildly applied in many different fields in the past decade, where a physically relationship could not be clearly defined. Given the recent development of deep learning algorithm with neural networks, there have been also applications in the geosciences, such as for rainfall prediction ( e.g Miao et al. [2015] ), weather forecast, and convection parameterization [Gentine et al., 2018], etc. One of the advantages of ANNs is that the training of the neural networks is only done once. As most of the computational cost is dedicated to the training phase, they can be applied efficiently after being trained. Eddy parameterization therefore can be an ideal problem to apply machine-learning algorithms and especially with deep learning.

In BM18, they used deep-learning algorithms with the Convolutional Neural Networks (CNNs) to reproduce the eddy momentum forcing. The CNNs are a class of deep neural networks. Compared to standard feedforward neural networks, CNNs have much fewer connections and parameters, so they are easier to train, while still having great quality [Krizhevsky et al., 2012]. Due to their superior performance in two-dimensional image processing, they are applied to successfully represent the subgrid eddy momentum forcing in BM18.

### 2.2.1. The architecture of CNNs

BM18 used a simple kind of CNNs, which includes three convolution layers, a max-pooling layer, and a fully-connected layer. It based on the CNN proposed by Lecun et al. [1998], but without the classification on the output. However with the final fully-connected layer, the total number of parameters is relatively large ( 325 728 in BM18 ). Inspired by the image super-resolution using CNNs [Dong et al., 2014], which directly learns an end-to-end mapping between the two two-dimensional images, we construct another CNNs in this study.

The CNNs we used take two-dimensional surface velocity ( zonal and meridional component  $u, v$  ) as the inputs, and output the map of eddy momentum forcing. It consists of four layers. The first is the input layer, which in our case is a two-dimensional image of snapshots. The first convolutional layer extracts 32-dimensional features from each input variable. In the second, we map each of these 32-dimensional vectors into a 16-dimensional one. Between two convolutional layers, the regular Rectified Linear Unit (ReLU) is applied, which makes convergence fast while still shows good quality. The output layer is transposed convolution layer (also called deconvolution ) to maintain the same shape the input, which is purely linear without an activation function. In the end, the total number of parameters to train is only 4449 compared to 325 728 in BM18. It reduces significantly computational cost. This kind of CNNs has a lightweight structure, and achieves fast speed with good quality.

### 2.2.2. Implementation in details

Before the training of CNNs, the data from the high-resolution model have to been pre-processed. As described in Section 2.1, using a low-pass Gaussian filter with the standard deviation  $\sigma =$

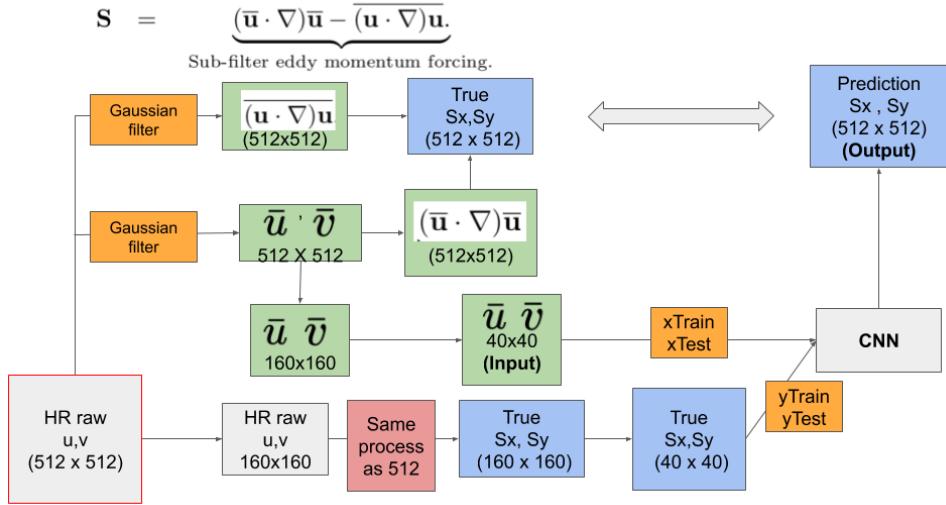


Figure 2.3: Illustration of the data processing and

28 km in two-gyre case, and 30 km in realistic simulations, we remove information of dynamics smaller than mesoscale in space and obtain the zonal and meridional component of filtered velocity ( $\bar{u}$  and  $\bar{v}$ ) respectively. The two-dimensional  $\bar{u}$  and  $\bar{v}$  are separately passed into the CNNs as inputs. The outputs are the corresponding zonal  $S_x$  and meridional  $S_y$  components of the subgrid momentum forcing  $S$ . In order to train efficiently, we subtracted the mean (both temporal and spatial) and divided by the standard deviation of all input values, to keep them on the same order of magnitude.

1000-days daily data are split into the first 800-days as training datasets to directly train CNNs, and the remainder 200-days are stored for validation. No latitude or longitude information is used to train neural networks. BM18 have determined the CNNs can generalize successfully the dynamics of the full-domain from local region. Following their strategy: training neural networks by one region to predict surrounding regions, we therefore extract a smaller region (size  $160 \times 160$  grid points) from the whole study domain ( $512 \times 512$ ). In addition to reduce the computational cost and increase the amount of training and validation data, we split further the region spatially from the initial  $160 \times 160$  into sixteen  $40 \times 40$  sub-regions, as illustrated in Fig.2.3. Therefore 800-days data provide 12800 snapshots of sub-regions as inputs to train the CNNs and 3200 for validation.

Training a network means to optimize the weight matrices and bias vectors of each layer, to minimize the difference between outputs of CNNs and the given desired targets. To measure this difference, we use the mean-squared error as the loss function with a stochastic gradient descent called Ada optimization algorithm [Kingma and Ba, 2014]. The CNNs are trained by 200 epochs, which means that the input will be passed forward and backward through the neural networks 200 times. For the reason of computational efficiency, we divide the inputs into smaller sets (batch size = 128) before passing them into CNNs each epoch. We started with a learning rate of  $10^{-3}$ . In total, we trained for 200 epochs. We train and implement CNNs to separately predict  $S_x$  and  $S_y$  using Keras [Chollet et al., 2015], with the Tensorflow backend [Abadi et al., 2016]. After training, the CNNs can extract the most important information from

the input to predict the output.

After the CNNs being trained, we use the validation dataset of the region to predict the whole area. Since the input and output size are  $40 \times 40$  grid points from the trained CNNs, we need to tile together all predictions by overlapping small patches with a stride of 2. Then they are averaged to produce the final full domain ( $512 \times 512$ ).

### 2.3. Implementation in coarse grid

Machine learning can be used to learn new parameterizations of small-scale processes directly from high-resolution model outputs. For instance, [O'Gorman and Dwyer, 2018] have successfully applied machine learning-based parameterization of moist convection to assist in simulations with a general circulation model. Our aim here is to apply the deep learning techniques to parameterize unresolved turbulent processes in coarse grid, by directly learning from high-resolution simulations.

Using sub-sampled data generated by an eddy-resolving model, we attempt to represent subgrid momentum forcing in low resolution by the identical architecture of CNNs we construct in the Section.2.2. The realistic data of the GS region are described in Section.2.1. They are produced by ROMS model at  $\Delta x \approx 2.5\text{km}$  resolution , which resolves all the necessary scales resulting in the correct coupling between mesoscale eddies and large-scale flows.

A larger domain has been used ( $720 \times 720$ ) here. In order to stimulate a low-resolution ocean model and obtain the coarse-grained samples of  $u$  and  $v$ , we resize each original two-dimensional snapshot of velocity to  $40 \times 40$  by averaging every  $18 \times 18$  points. One example of coarse-grained zonal velocity is shown in Fig.2.4. And  $\Delta x$  turns to  $45\text{km}$ , where mesoscale eddies dominate around this range of scales.

Instead of training the CNNs by a small region to reproduce the whole domain as discussed previously, the full region of coarse-grained data is passed as the input variables, so the output subgrid momentum forcing in size  $40 \times 40$  as well. Remarkably, the parameters of CNNs reduces sharply and it makes the CNNs much faster to learn.

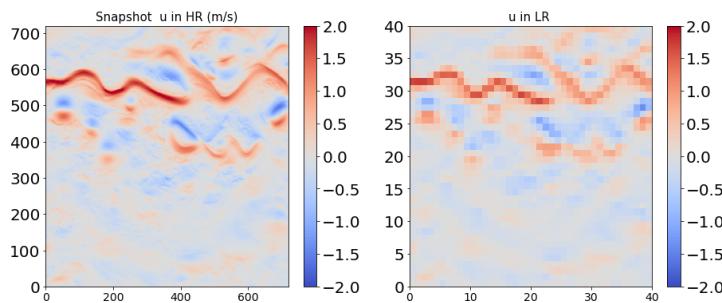


Figure 2.4: The instantaneous spatial maps of zonal velocity  $u$  in realistic simulation  $720 \times 720$  (HR) (left), and down-scaled into  $40 \times 40$  (LR) (right)



# 3

## Results and Analysis

### 3.1. Comparison of CNNs with BM18

BM18 have trained CNNs on degraded data from a high-resolution QG ocean model, and successfully predicted eddy momentum forcing on the two-gyre case. As described in Section 2.2.2, we only train the neural networks using datasets from one specific region, but generalize eddy momentum forcing across the whole domain. BM18 chose three regions: Region 1 is near the western boundary with a strong eastwards jet; Region 2 is near the eastern boundary; Region 3 is the southern gyre area which is much less turbulent than the other two. They showed that training on the western boundary produces the best performance. Therefore, using our data from the western boundary in the gyre simulation, the CNNs of BM18 train and then compare to another architecture of CNNs. The two neural networks are trained by the same training dataset, and test their performance with the 200-days validation dataset.

Our results demonstrate that the CNNs of BM18 can be applied in different datasets and still reproduce great generalizations ( Fig.3.1 first two columns). The true zonal eddy momentum forcing  $S_x$  calculated directly by using the validation dataset across the full domain. The jet dynamics dominates around the western boundary and extends to eastwards in both spatial and temporal variability  $S_x$  (Fig.3.1 1a,2a,3a). The amplitude of instant  $S_x$  is significantly higher than the time-averaged one.

The spatial patterns of the true  $S_x$  and prediction  $\tilde{S}_x$  in one single snapshot and the time-mean are almost identical, except the amplitudes are slightly weaker in  $\tilde{S}_x$ . As in BM18, our CNNs also successfully represent both the spatial and temporal variability of the eddy momentum forcing from surface horizontal velocity (Fig.3.1 third column ). In general, the performances of these two CNNs are really similar both in patterns and amplitudes. For the standard deviation, these two neural networks underestimate the generalization of variance of sub-grid eddy forcing. However, the CNNs of BM18 suppress visibly high variance of  $S_x$  in the western boundary, while our CNNs can still capture some of the high values (Fig.3.1 3a-c ).

To compare different CNNs more easily, we calculate the Pearson correlation between the true  $S_x$  and the prediction  $\tilde{S}_x$  (Fig.3.1 4b,4c ). The two CNNs both have high correlations ( $r > 0.9$ ) at the western boundary, while low or negative values near the eastern boundary due to the reason that has been explained in BM18: The larger spatial-scale at the eastern boundary reduces the sub-grid eddy momentum forcing to almost zero, and the wave-like behavior causes neural networks to badly react. Poor correlations are within gyres as well, this suggests that the dynamical processes have an opposite effect between the western boundary and gyres [Waterman and Jayne, 2010].

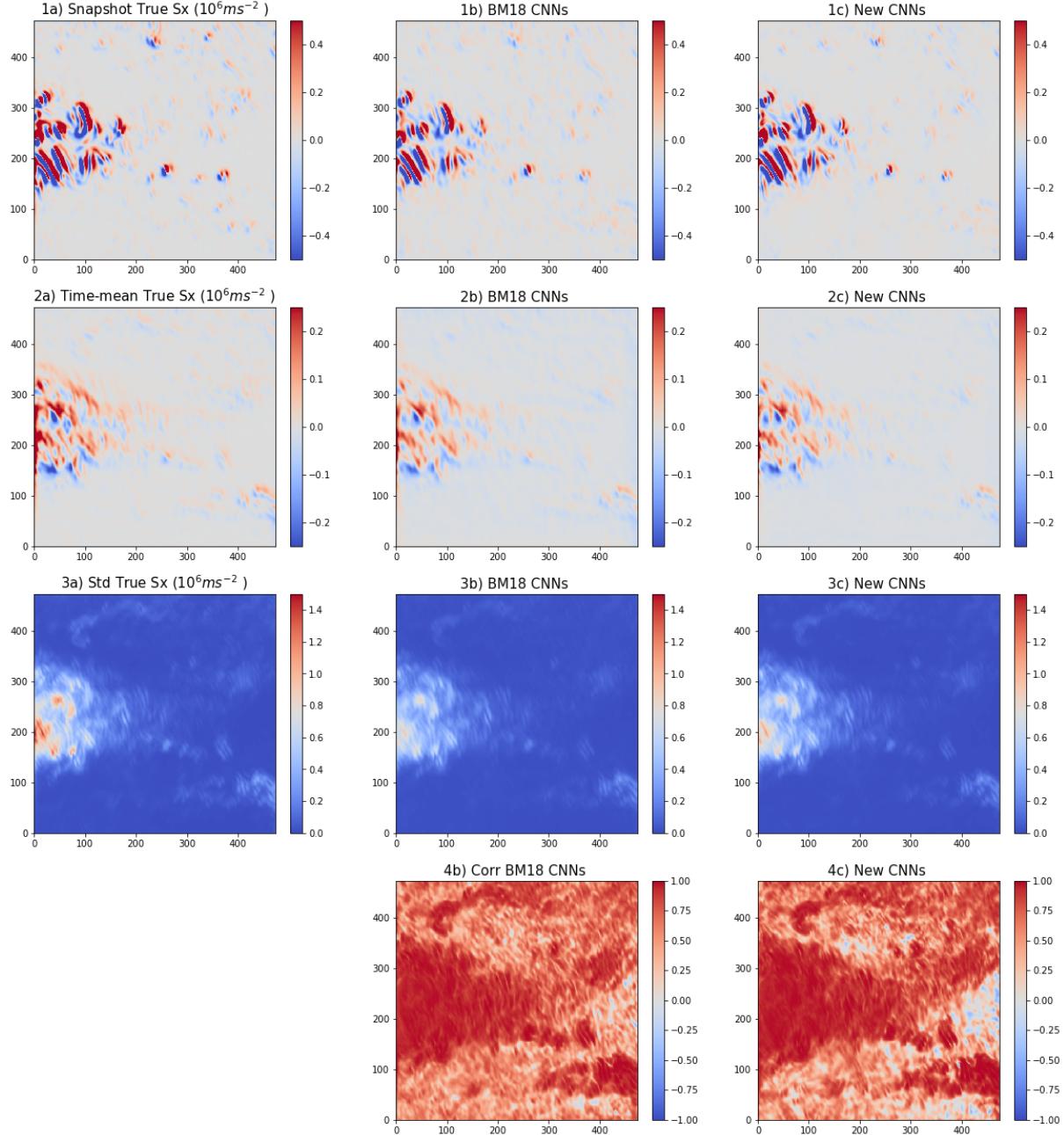


Figure 3.1: Comparisons of the true zonal component of the zonal eddy momentum forcing  $S_x$  with the CNNs trained by the western boundary. The first column shows the true  $S_x$  calculated from surface velocity; the second for the CNNs of BM18; the third for the our CNNs. Then the first three rows show the snapshots  $S_x$  at one single point, time-means, the standard deviation, respectively. The last row for the correlation between the true  $S_x$  and the predictions  $\tilde{S}_x$

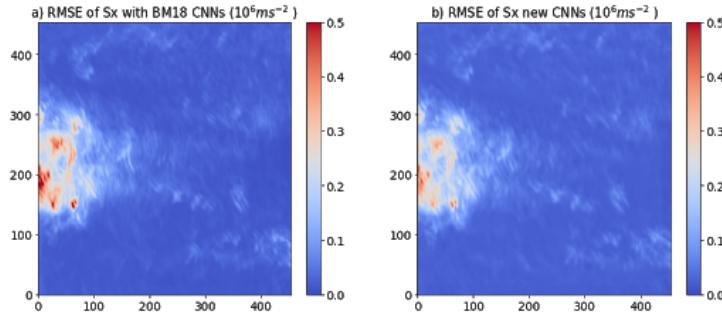


Figure 3.2: The spatial maps of time-averaged RMSE between the true zonal component of the subgrid eddy momentum forcing  $S_x$  the predictions  $\tilde{S}_x$  with the CNNs of BM18 (left), and with a new CNNs (right)

In order to assess the performance of the two neural networks further, we compare the root-mean-square error (RMSE) on validation datasets over the full domain. After being time averaged, the spatial maps of RMSE are shown in Fig.3.2. They have similar spatial patterns, and low values generally. But higher RMSE areas emerge around the western boundary (Fig.3.2), where there is an eastward jet and many intense meanders. However, the extreme values are lower in our CNNs ( $\text{RMSE} < 0.5 \times 10^{-6} \text{ms}^{-2}$ ). The complexity of intense turbulence makes it difficult for CNNs to learn and represent its dynamics.

These two CNNs can both successfully represent temporal variability of the eddy momentum forcing  $\tilde{S}_x$  as well. For instance, we show that time series of  $S_x$  at one point with two CNNs in Fig.3.3. The two time series  $\tilde{S}_x$  generally match the fluctuation of true  $S_x$ , but our CNNs have better performance in amplitude most parts.

Overall our CNNs show similar performance as that in BM18, but it takes much less time to compute, due to much fewer parameters in CNNs layers. So we apply it for the following studies.

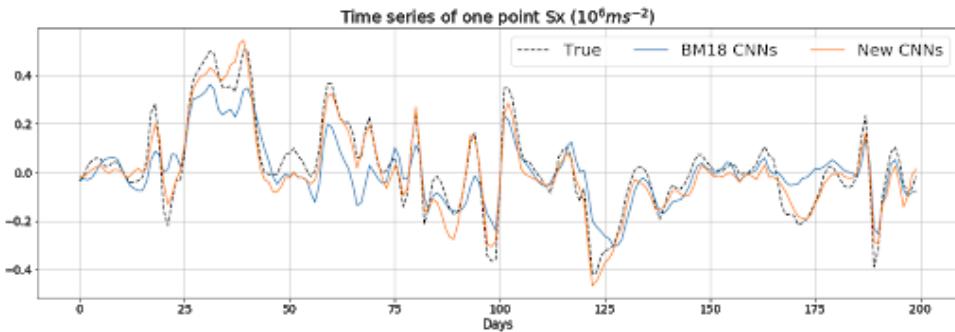


Figure 3.3: Time series of the true zonal component of the subgrid eddy momentum forcing  $S_x$  at one single point with CNNs of BM18 ( in blue ) and our CNNs ( in orange ), both trained by the western boundary.

## 3.2. Sensitivity of Neural Networks in realistic simulations

After testing that our CNNs can be applied successfully in the idealized gyre case, we move on to more realistic simulations in the North Atlantic from this section. BM18 has proven that the training datasets from different regions significantly impact the accuracy. In the study domain

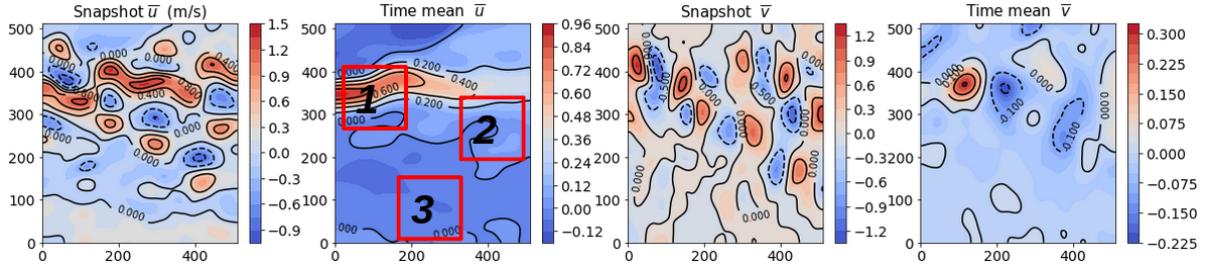


Figure 3.4: The snapshot (a) and time-average (b) of filtered zonal velocity  $u$ ; The snapshot (c) and time-average (d) of filtered meridional velocity  $v$ . These study regions have been shown in (b) with red line.

that we choose, the strong GS can be seen clearly in the north of Fig.3.4, and intense meanders extend towards northeast. The whole domain is turbulent as expected (Fig.3.4 a,c). The time-mean zonal velocity  $u$  is positive in the jet area. Its amplitude is highest ( $\approx 1\text{m/s}$ ) near the western boundary, and decreases eastwards (Fig.3.4(b)). For meridional velocity  $v$ , the highest spots are near the western boundary as well (Fig.3.4 d) and are weaker ( $\approx 0.3\text{m/s}$ ).

Following the strategy of BM18, we choose three regions as shown in (Fig.3.4 (b) ). Region 1 is the most energetic due to the eastwards jet, Region 2 part of the GS extension, is turbulent with meanders, while Region 3 is calmest and close to the gyre.

As suggested in BM18, training neural networks on different positions leads to different dynamical processes being learned. To successfully reproduce the correct amplitude and variability across the domain, training data must contain a diverse range of scale interactions. Hence, we mix three regions in a proportion : 33% of Region 1, 33% of Region 2, 34% of Region 3 to reconstruct a new 1000-snapshots dataset to train the CNNs. We furthermore create an artificial region: randomly sub-sampling 1000 snapshots of  $160 \times 160$  size from the full domain ( $512 \times 512$ ) to make the samples effectively independent, then test the neural networks using this reconstructed region. In this way, each snapshot of velocity is independent, time series of each point have been reconstructed, and have lost their temporal history. But we use the same 200-snapshot validation data from the original dataset, to make representations of eddy momentum forcing in all the tests.

### 3.2.1. Non-Local Predictions

As in the gyre case, we first examine one single snapshot, the time-mean and the standard deviation by training 5 regions respectively, shown in Fig.3.5,3.6. Both the spatial and temporal variability of the true  $S_x$  are dominated by strong meanders along the path of the GS, which cause complex spatial patterns. The amplitude of the time-mean of  $S_x$  is lower by an order of magnitude than the spatio-temporal variability ( Fig.3.5). The turbulent regions exhibit high variance ( $\approx 310^{-6}\text{ms}^{-2}$  in spatial average ), and the variability of  $S_x$  expands strongly eastwards (Fig.3.6 (a)).

Trained different regions respectively, the 5 CNNs all reproduce good generalization in spatial variability with correct pattern and amplitude ( Fig.3.5 left panel). In time-mean  $S_x$ , the mixed dataset represents the best performance ( Fig.3.5 E), while the others poorly generalize along the path of currents: Region 1 to 3 underestimate it, and the reconstructed case over-

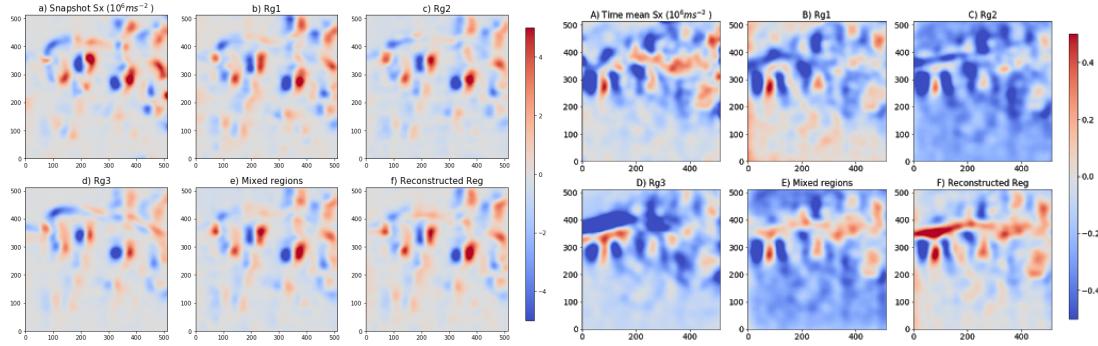


Figure 3.5: Comparisons of one snapshot (Left panel) and the time-mean (right panel) a) the true eddy momentum forcing  $S_x$  and  $\tilde{S}_x$  with different regions trained : b) Region 1 ,c) Region 2, d) Region 3, e) the mixed regions, f) reconstructed region

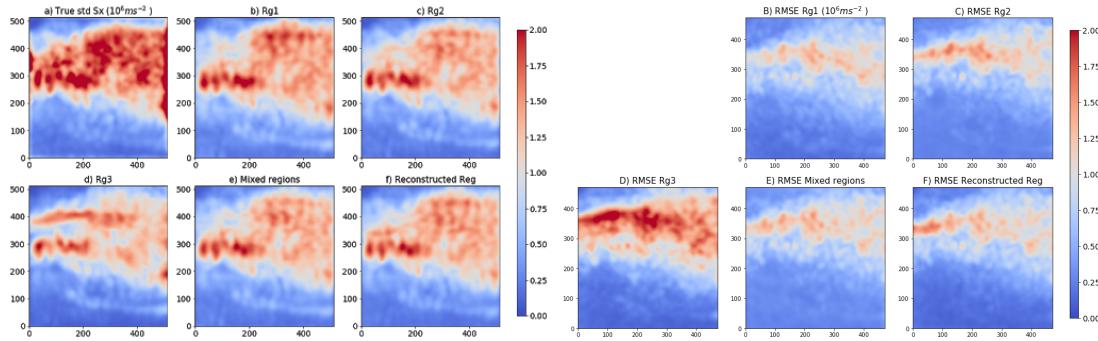


Figure 3.6: Comparisons of standard deviation (left panel) and RMSE (right panel) a) the true zonal eddy momentum forcing  $S_x$  and  $\tilde{S}_x$  with different regions trained : b) Region 1 ,c) Region 2, d) Region 3, e) the mixed regions, f) reconstructed region

estimates on the western boundary. This implies that training on an adaptive combination of different regions, can effectively improve the performance of the neural networks. Generally, standard deviations of  $\tilde{S}_x$  are much lower than true  $S_x$  (Fig.3.6 left panel). Training on the most energetic area ( Region 1 ) produces the best performance ( Fig.3.6(b) ). However, it does not fully translate high variance along the strong jet. It is obvious that more complex dynamical processes in realistic simulations are difficult to be learned and represented by neural networks, compared to the idealized gyre case.

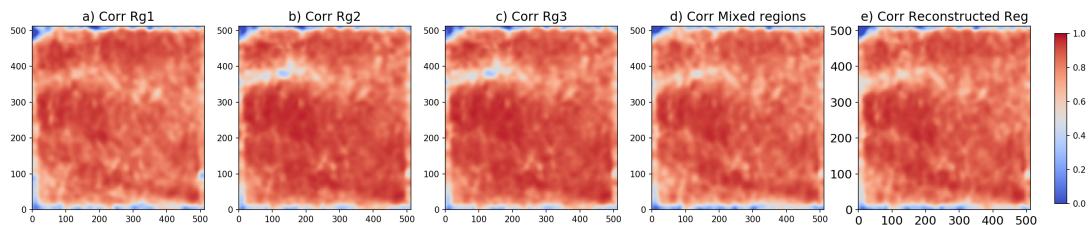


Figure 3.7: Comparisons of the correlation between the true  $S_x$  and  $\tilde{S}_x$  with different regions trained : a) Region 1 ,b) Region 2, c) Region 3, d) the mixed regions, e) reconstructed region. The zero represents both zeros and negative values

For the Pearson correlation between the true  $S_x$  and the predictions  $\tilde{S}_x$  , the figures (3.7) have shown how the performance varies between regions of differing energetic activity. Given

the edge effect of the spatial filter applied, low correlations are around the boundaries. Overall, the predictions are highly correlated with the true  $S_x$  (spatial means  $r= 0.85 \pm 0.01$ ), except for Region 3 ( $0.72$ ). For all neural networks, the correlation decreases visibly along the path of the jet near the western boundary. The values tend towards low or negative correlation there in Fig.3.7 (b,c,d,e). This suggests that it is difficult to make predictions with strong meanders and eddies (high velocity in particular), which could deteriorate the performance of the neural networks. Therefore, training the CNNs directly on the most turbulent and energetic Region 1 has the most successful generalization along the currents.

However, the high performance in Region 1 does not fully appear outside the area, Region 2 and 3 can reproduce best correlations in the southern (Fig.3.7 (b,c)). Considering the whole domain, the mixed and randomly reconstructed region (Fig.3.7 (d,e)) effectively improve the performance of the neural networks. But determining an adaptive combination of regions still needs to be discussed in the future.

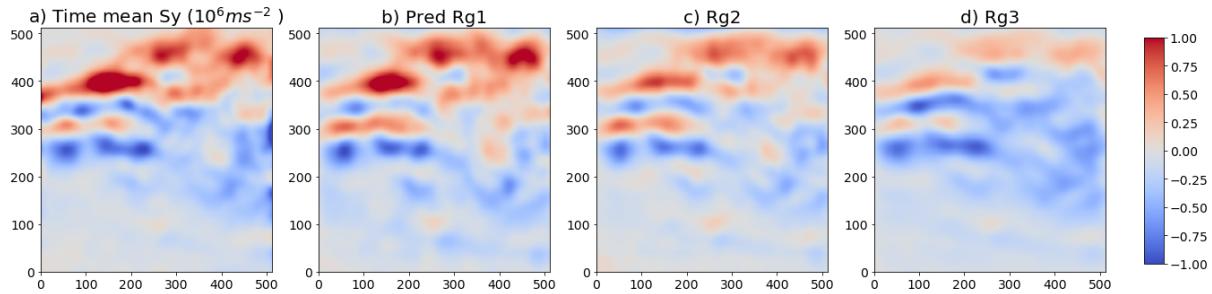


Figure 3.8: Comparisons of time-mean of a) the true meridional eddy momentum forcing  $S_y$  and  $\tilde{S}_y$  by training different regions respectively : b) Region 1 , c) Region 2, d) Region 3

Unlike the complex spatial structure of the zonal component  $S_x$ , the time-mean patterns of the meridional eddy momentum forcing  $S_y$  have nearly zonal sharp gradients, which suggests the presence of a force that tends to tear the mean flow away from its mean axis [Ducet and Le Traon, 2001]. The intersections between positive and negative gradients are very well defined curves (Fig.3.8 a). Generally, the CNNs have better performance in reproducing  $S_y$ . Training by Region 1, the predictions , are almost identical to the true values in spatial variability (not shown), and in time-mean spatial patterns (Fig.3.8 b), better than the representation of  $S_x$  (Fig.3.5 ). Training by three regions respectively, Region 1 leads to the best predictions in time-mean including the correct amplitude and spatial pattern of  $\tilde{S}_y$  across the full domain. In the contrary, Region 3 reproduces part of the pattern, but underestimates heavily within the GS jet. As the case of its zonal component  $S_x$ , training only Region 3 can not make the CNNs to generalize correctly high amplitude of  $\tilde{S}_y$ .

We observe the correlations between the true  $S_y$  and the predictions  $\tilde{S}_y$  in Fig.3.9. The negative and low correlations ( $r \approx -0.1$ ) evidently appear again within the jet in the case of Region 3 (Fig.3.9 c), consistent with  $S_x$ . The best spatial-averaged correlation ( $r= 0.81$ ) is by training Region 2, but the values drop slightly within the upstream of the GS jet (Fig.3.8 c). While Region 1 does not correlate well in the southern.

Overall, Training on the most turbulent and energetic region (Region 1) leads to the best predictions including the correct amplitude and spatial pattern of the eddy momentum forcing

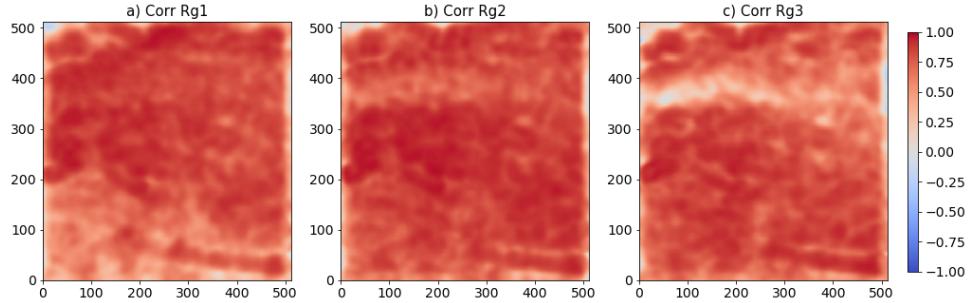


Figure 3.9: Comparisons of the correlation between the true  $S_y$  and  $\tilde{S}_y$  by training different regions respectively : a) Region 1 ,b) Region 2, c) Region 3

over the whole domain. Training by Region 3 generalizes the worst  $\tilde{S}_x$  in all discussed terms. It is consistent with the idealize gyre case in Section 3.1. The dynamics of the GS jet and its recirculation gyres have an nonlinear effect between each other. Eddies and eddy-flow interactions impact mean jet–gyre strength, structure, and stability through the mechanism of nonlinear eddy rectification [Waterman and Jayne, 2010].

On the other hand, Region 1 and Region 2 both have higher sub-grid eddy momentum forcing given the path of the GS currents. However, Region 2 is relatively easier to be predicted from any trained CNNs. It may cause by highest time-mean zonal velocity in Region 1 (Fig.3.4(b)). As discussed in BM18, two regions may be as turbulent or energetically active as each other, but the nature of the eddy-mean flow interactions within them, may differ.

Due to the good performances of the mixed and the reconstructed region, they suggests that temporal dynamic behavior may not impact the performance of neural networks. As feed-forward neural networks, the CNNs have good generalization ability. They can treat each snapshot independently, no need to feed time-continuous slices of data. Our work implies the flexibility of choosing data to train neural networks.

### Conservation of Energy

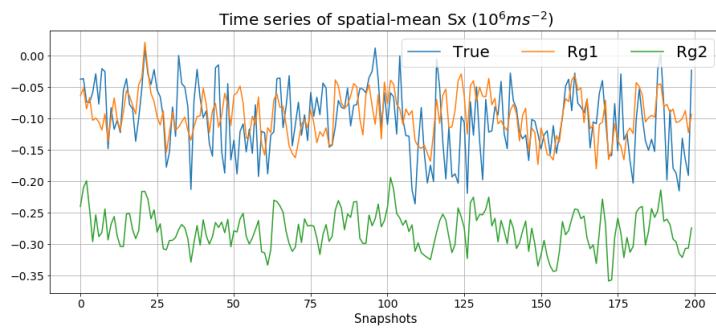


Figure 3.10: Time series of the zonal subgrid momentum forcing  $S_x$  ( blue), the prediction training by Region 1 ( orange ), by Region 2 ( green ), in spatially-averaged over the entire domain.

Neural networks do not intrinsically preserve energy, and cause momentum biases. To respect physical conservation laws, each studying region should have a zero spatially-integrated momentum tendency in the ideal gyre case. In BM18, they therefore found the best and easiest method to constrain the neural networks is post-processing of output, which means to remove

the spatial-temporal mean. We apply the same approach in our gyre case. But in realistic simulation of the GS region, the spatially-integrated momentum in the study domain we extract, is not zero (Fig.3.10), and overall  $-0.10 \times 10^{-6} ms^{-2}$ . In contrast to other regions, the CNNs training by Region 1 reproduces overall momentum  $-0.9 \times 10^{-6} ms^{-2}$ , close to the true value. Therefore, it does not need a postprediction correction.

### 3.2.2. The impact of sub-region size

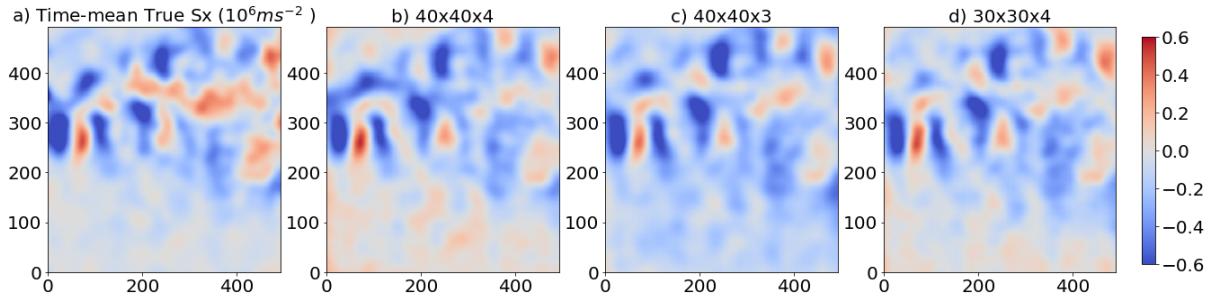


Figure 3.11: The spatial map of time-mean of a) the true  $S_x$  and the predictions  $\tilde{S}_x$  b) in  $40 \times 40$  sub-region of original Region 1 ( $160 \times 160$ ) , c) in  $40 \times 40$  sub-region of a small region ( $120 \times 120$ ) from original Region 1, d) in  $30 \times 30$  sub-region of a small region as c).

One disadvantage of CNNs is the computational cost of the matrix operations of each convolution layer. Using smaller size of the input and output can skillfully reduce the number of parameters in CNNs layers and computational cost. As discussed in BM18, both velocity and velocity shears are important features in accurate predictions of the eddy momentum forcing. Reducing the size of sub-regions of the inputs could lose the local features to predict across the full domain, even using the same amount of data to train.

Here we discuss how the performance varies from the sub-region size in time-mean eddy momentum forcing  $S_x$ . At first, we extract a smaller region ( $120 \times 120$ ) from the original Region 1 ( $160 \times 160$ ), but keep the sub-region size as  $40 \times 40$ . There is theoretically less information in total to train the CNNs, but the performances of two tests do not vary obviously. The spatial-averaged correlation is 0.79 in  $120 \times 120$  and 0.80 in the original, and the RMSE is  $0.63 \times 10^{-6} ms^{-2}$  and  $0.61 \times 10^{-6} ms^{-2}$  respectively. On the contrary, training on the smaller region reproduces better spatial patterns across the full domain in time-mean comparing to the true  $S_x$  (Fig.3.11 c). While the original Region 1 has a positive bias in the southern shown in Fig.3.11 b. This suggests that the smaller region we choose may contain the key information of eddy momentum forcing to train neural networks, and the CNNs can be applied more widely if we choose the training data wisely.

Keeping this same small region ( $120 \times 120$  from Region 1 ), we then reduce the sub-region size from  $40 \times 40$  to  $30 \times 30$ . The CNNs are trained by the same amount of data in total in this comparison. The RMSE and spatial-mean correlation of two cases are almost identical ( $0.63 \times 10^{-6} ms^{-2}$  , 0.79 respectively). The generalizations in time-mean  $\tilde{S}_x$  are very similar in both amplitude and spatial pattern (Fig.3.11 c, d ). Furthermore in the case of  $30 \times 30$ , the spatial-averaged momentum of the whole domain ( $-1.03 \times 10^{-7} ms^{-2}$  ) is most similar to the value diagnosed directly from velocity ( $-1.01 \times 10^{-7} ms^{-2}$  ) in three cases. Therefore using smaller

sub-region does not decrease obviously the accuracy of CNNs in our tests. But it can increase effectively computing speed and reduce computational cost.

Using another smaller training region  $150 \times 150$  from the original Region 1 ( $160 \times 160$ ), we increase the size of sub-region from  $40 \times 40$  to  $50 \times 50$ . Comparing to the original case, the test does not show any evident improvement, but the RMSE increases from 0.61 to  $0.63 \times 10^{-6} ms^{-2}$ . Considering on the increase of computational complexity, larger sub-region is not necessary to be applied.

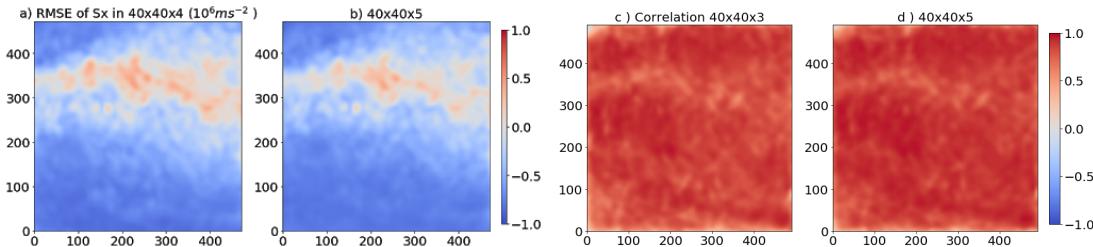


Figure 3.12: The spatial map of time-mean of a) the true  $S_x$  and the predictions  $\tilde{S}_x$  b) in  $40 \times 40$  sub-region in Region 1 ( $160 \times 160$ ), c) in  $40 \times 40$  sub-region in small region ( $120 \times 120$ ), d) in  $30 \times 30$  sub-region in the small region ( $120 \times 120$ )

Retaining the same size of sub-regions, we increase the training region from the original Region 1 to  $200 \times 200$ . Therefore there are more amount of training data in total, but the representation of spatial and temporal  $S_x$  does not improve obviously (not shown). However, with larger training region, the spatial-averaged RMSE drops from 0.61 to  $0.55 \times 10^{-6} ms^{-2}$ , and the difference at downstream of the GS jet between these two cases is clear in Fig3.12 (a,b). And the mean correlation increases from 0.81 to 0.83, the improvement can be seen over the full domain (Fig3.12 c, d) . This suggests that the CNN can have better performance by training with larger amount of data, though the prediction of subgrid momentum forcing in spatial and temporal variability, does not turn more accurate.

In summary, the origin combination of region and sub-region size from BM18 may not be the best choice in our realistic simulation. Reducing suitably the input size of training data can represent similar or even better eddy momentum forcing , and increase the compute speed significantly.

### 3.2.3. The impact of the amount of training data

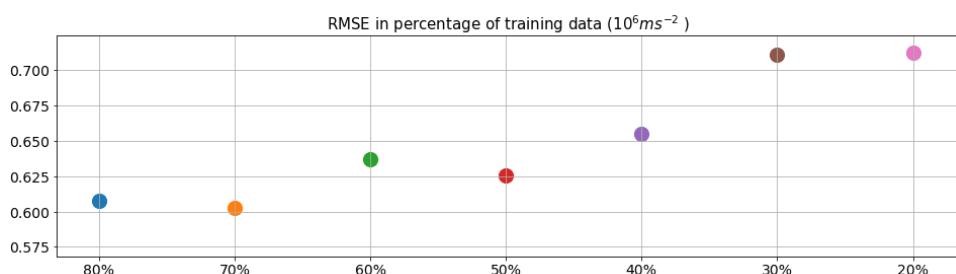


Figure 3.13: The spatial-averaged RMSE as a function of the percentage of the training data between the true  $S_x$  and the predictions  $\tilde{S}_x$

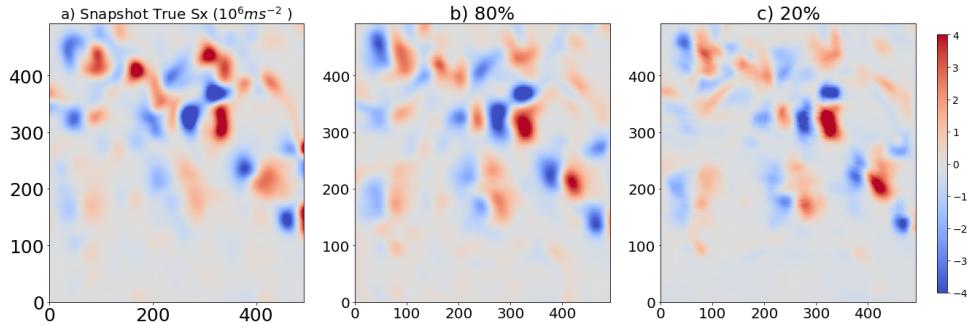


Figure 3.14: The spatial map of one single snapshot of a) true  $S_x$  and the predictions  $\tilde{S}_x$  b) using 80% and c) 20% of whole dataset to train the CNNs

BM18 used 9-years and 1-year daily data as training and validation dataset respectively. Before this section, we always use the 800 snapshots of data to train the CNNs and 200 for validation, and the CNNs remain good performance shown in previous sections. To explore further the sensitivity of the neural networks, we vary the amount of training data from 80 % to 20 % of the original 1000 snapshots from Region 1 to train the CNNs, and we use the last 200 snapshots to make predictions of  $S_x$  over the full domain. We determinate the impact of the amount of training data in Fig.3.13. Besides variances caused by the stochastic nature of the training procedure, we find that the spatial averaged RMSE increases gradually from 0.6 to  $0.71 \times 10^{-6} ms^{-2}$  while the percentage of training data reduces. However, the CNNs trained by a low amount of data ( 20% ), can still generalize temporal and spatial  $\tilde{S}_x$  consistent with the true value, in terms of the amplitude and patterns, but in blurry condition ( Fig.3.14 c ). This implies that the amount of training data can improve significantly the output quality, but not eventually the accuracy in patterns. It may be sufficient for the neural networks to generalize sub-grid scale processes by using only a limited amount of high-resolution data. However, due to the limit duration of our data, we have not yet checked if larger training data could have better performance of CNNs.

### 3.3. Implementation in coarse-resolution simulation

Here we attempt to apply the CNNs into a coarse resolution version of simulations to test their performance as a subgrid-scale parametrization. Using data sub-sampled from the eddy-resolving ( $\Delta x \approx 2.5 km$ ) ocean model, we demonstrate that the CNNs can represent both the spatial and temporal variability of the subgrid eddy momentum forcing in coarse-grid. The spatial filter causes heavy edge effect, which makes the representation of eddy momentum forcing at boundaries unrealistic. As a result, the coarse-grained data generalize similar spatial pattern to the true  $S_x$  calculated directly from sample high-resolution data in Fig.3.15. After sub-sampling, the coarse data remain the features of subgrid eddies. The true  $S_x$  highly correlates to the  $\tilde{S}_x$  along the path of the jet. The spatial averaged local correlation  $r \approx 0.76$  compared to the overall 0.53, due to the poorly predictions in the gyre. Same for  $S_y$  with the local and whole  $r$  0.78 and 0.51 respectively. The neural networks are trained by the full domain, which contains a diverse range of scales. As discussed in high-resolution cases, the amplitude of subgrid momentum forcing is orders of magnitude higher along the jet, and dynamics around

the jet and the gyres have opposing effects to reduce sharply the performance of the neural networks. The data from the region more turbulent, the predictions are more precise.

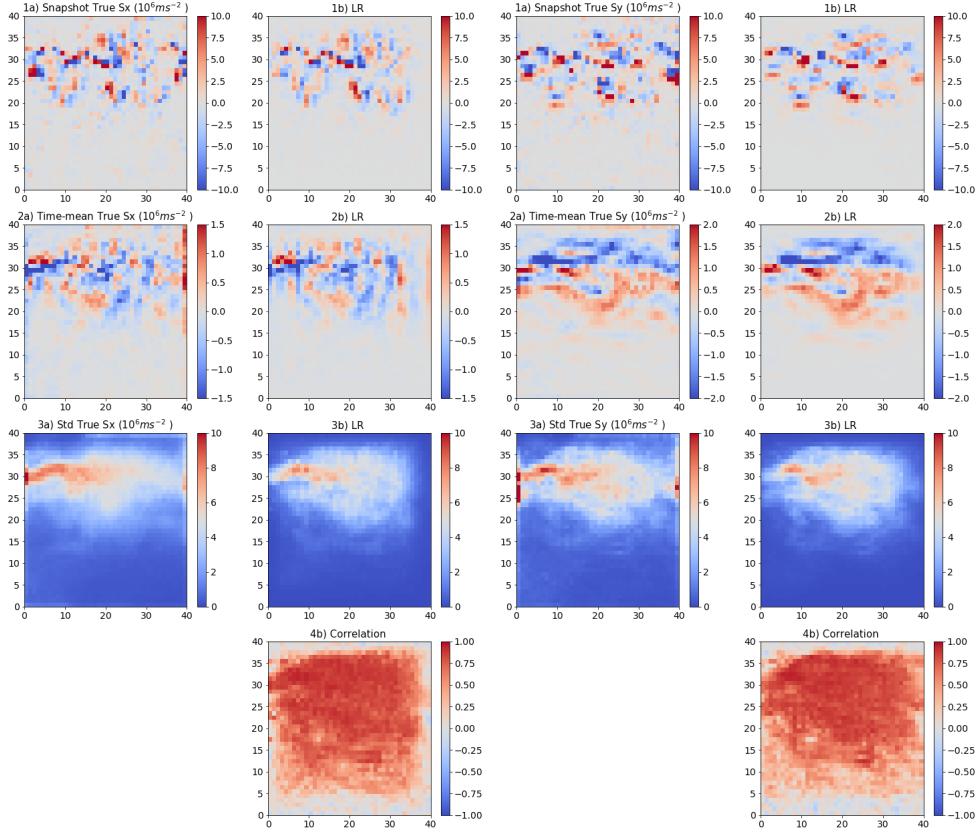


Figure 3.15: Comparisons of the true zonal component of the subgrid eddy momentum forcing  $S_x$  ( left panel ) with the CNNs trained by the western boundary. The first column shows the true  $S_x$  calculated from surface velocity at one given snapshot, time-means, the standard deviation,respectively; the second for the prediction  $\tilde{S}_x$ . while the last row for the correlation between  $S_x$  and  $\tilde{S}_x$ . Same for  $S_y$  in right panel

Fig.3.16 is two scatterplots between time-averaged true eddy momentum forcing and corresponding prediction. The coefficient of determination ( $R^2$ ) for  $S_x$  and  $S_y$  are 0.77 and 0.87 respectively. The mean bias is both negligible at  $-2.66$  and  $2.66 \times 10^{-8}$  respectively. The generalization has good accuracy in both variables, better in  $S_y$  (the coefficient of linear regression is 0.91, closed to 1). Our results demonstrate that the CNNs have the ability to represent eddy momentum forcing by using a coarse-grained data as input. This suggests the potential for data-driven oceanic turbulence closures in the future. Only a small amount of sub-sampled high-resolution data are sufficient for neural networks to successfully represent subgrid scale processes. To implement the CNNs as an approach of eddy parameterizations to augment the low-resolution model is therefore worth investigating.

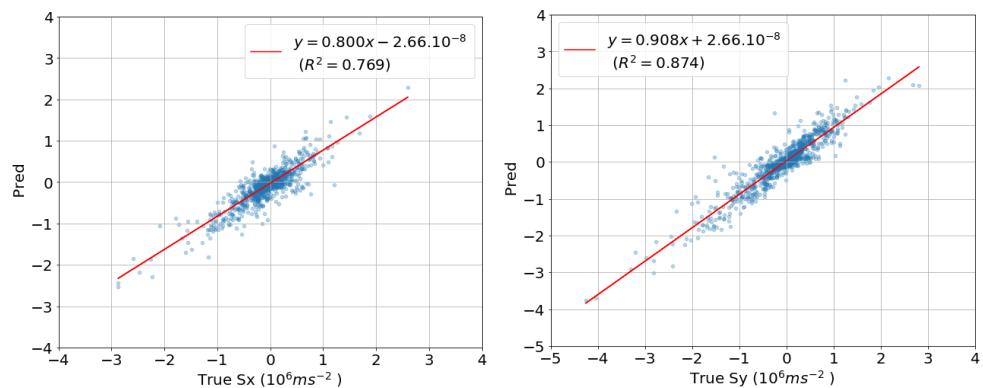


Figure 3.16: Scatterplot between time-averaged true  $S_x$  (left),  $S_y$  (right) and corresponding prediction in coarse grid.

# 4

## Conclusion

Proved in the idealized gyre case in BM18, we explore the machine learning algorithms to represent unresolved turbulent processes, in particular, the eddy momentum forcing, in a realistic eddy-solving simulation of the GS region. We have demonstrated that deep learning with the CNNs, can skillfully predict the relationship between the eddy momentum forcing and surface velocity. By minimizing the error between predictions and the known output over training datasets, the CNNs can successfully generalize subgrid and large-scale eddy momentum forcing in both the spatial and temporal variability from the velocity at surface. Considering on the spatial patterns and the amplitude, the CNNs reproduce the meridional eddy momentum forcing  $S_y$  better than its zonal component  $S_x$ . For  $S_y$ , the predictions in spatial and temporal variability , are almost identical to the true values. On the contrary, the CNNs we use have relatively bad performance in the map of time-mean  $\tilde{S}_x$ .

We have investigated several factors that affect the generalization of the CNNs in this study. For the training region, we have shown how the performance varies between training regions regarding in energetic level in the GS region. The performance of the neural network depends on the data. If the data from the region more turbulent, the predictions are more precise. Besides, high velocity also reduces the accuracy of the neural networks, in particular upstream of the GS currents. The CNNs have proven to represent successfully the dynamics of the full-domain from local region in BM18. We vary the size of training region ( original  $160 \times 160$  ) to test the sensitivity of the CNNs. With larger region being passed to train, more information being learned, so that we will have better prediction. Then we change the size of sub-regions which are directly passed into the neural networks. Velocities and velocity shears are key features to reproduce the eddy momentum forcing, so it is important to preserve enough local information for the CNNs in sub-regions. The original combination of region and sub-region size in BM18 is not the optimist choice in our realistic simulation. Smaller size of training data can represent similar or even better eddy momentum forcing , and increase the compute speed significantly. Regarding to the amount of training data, BM18 used 10-year daily data to train the neural network. Reducing in size of training data from 9 years to 800 days, our results have proven sufficient to have good performance of the CNNs with fewer number of inputs. Using only 200-days to train, the spatial-averaged mean-square root error increases by 15%, and still reproduce the correct spatial pattern of eddy momentum forcing. Overall, we have to make a compromise between the accuracy of prediction and computing efficiency.

A promising approach is to apply machine learning to represent small-scale processes in a low-resolution models, based on output from high-resolution models, which have been investigated in climate studies and the turbulent modeling ( e.g [O'Gorman and Dwyer, 2018]).

Machine learning therefore represents a powerful way to reproduces unresolved turbulent processes comparable to the high-resolution model. In contrast to conventional parameterizations, the training of the neural networks is only done once. It is computationally efficient, even for relatively large data. Using sub-sampled data from high-resolution, we show that the deep learning with CNNs can accurately represent both the spatial and temporal variability of the eddy momentum forcing comparable to the high-resolution model. This implies potential of a new parameterization for unresolved processes. Our work provides insights in using machine learning as an eddy parameterization to augment the low-resolution ocean models in the future.

# Bibliography

- M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.
- P. S. Berloff. On dynamically consistent eddy fluxes. *Dynamics of Atmospheres and Oceans*, 2005.
- T. Bolton and L. Zanna. Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, 2019.
- F. Chollet et al. Keras, 2015.
- C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, 2014.
- N. Ducet and P.-Y. Le Traon. A comparison of surface eddy kinetic energy and reynolds stresses in the gulf stream and the kuroshio current systems from merged topex/poseidon and ers-1/2 altimetric data. *Journal of Geophysical Research: Oceans*, 106(C8), 2001.
- R. Ferrari and C. Wunsch. Ocean circulation kinetic energy: Reservoirs, sources, and sinks. *Annual Review of Fluid Mechanics*, 41(1):253–282, 2009.
- P. R. Gent. The gent–mcwilliams parameterization: 20/20 hindsight. *Ocean Modelling*, 2011.
- P. R. Gent and J. C. Mcwilliams. Isopycnal mixing in ocean circulation models. *Journal of Physical Oceanography*, 20(1):150–155, 1990.
- P. Gentine, M. Pritchard, S. Rasp, G. Reinaudi, and G. Yacalis. Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, 2018.
- R. J. Greatbatch, X. Zhai, M. Claus, L. Czeschel, and W. Rath. Transport driven by eddy momentum fluxes in the gulf stream extension region. *Geophysical Research Letters*, 37(24), 2010.
- J. Gula, M. J. Molemaker, and J. C. McWilliams. Gulf stream dynamics along the southeastern u.s. seaboard. *Journal of Physical Oceanography*, 2015.
- R. Hallberg. Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects. *Ocean Modelling*, 72:92 – 103, 2013.
- G.-Q. Jiang, J. Xu, and J. Wei. A deep learning algorithm of neural network for the parameterization of typhoon-ocean feedback in typhoon forecast models. *Geophysical Research Letters*, 45(8):3706–3716, 2018.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- Z. Lachkar, J. Orr, J.-C. Dutay, and P. Delecluse. On the role of mesoscale eddies in the ventilation of antarctic intermediate water. *Deep Sea Research Part I: Oceanographic Research Papers*, 56, 2009.
- Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

- J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016. doi: 10.1017/jfm.2016.615.
- P. P. Mana and L. Zanna. Toward a stochastic parameterization of ocean mesoscale eddies. *Ocean Modelling*, 79:1 – 20, 2014.
- C. Miao, H. Ashouri, K.-L. Hsu, S. Sorooshian, and Q. Duan. Evaluation of the persiann-cdr daily rainfall estimates in capturing the behavior of extreme precipitation events over china. *Journal of Hydrometeorology*, 16(3):1387–1396, 2015.
- P. A. O’Gorman and J. G. Dwyer. Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, 10(10):2548–2563, 2018.
- J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
- A. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following- coordinate ocean model. *Ocean Modelling*, 9:347–404, 12 2005.
- S. Waterman and S. R. Jayne. Eddy-mean flow interactions in the along-stream development of a western boundary current jet: An idealized model study. 2010.
- K. Wyrtki, L. Magaard, and J. Hager. Eddy energy in the oceans. *Journal of Geophysical Research (1896-1977)*, 81(15):2641–2646, 1976.
- L. Zanna, P. P. Mana, J. Anstey, T. David, and T. Bolton. Scale-aware deterministic and stochastic parametrizations of eddy-mean flow interaction. *Ocean Modelling*, 111:66 – 80, 2017.
- Z. Zhang, W. Wang, and B. Qiu. Oceanic mass transport by mesoscale eddies. *Science*, 345(6194): 322–324, 2014.