

Addressing out-of-sample issues in multi-layer convolutional neural-network parameterization of mesoscale eddies applied near coastlines

Cheng Zhang¹, Pavel Perezhogin², Alistair Adcroft¹, Laure Zanna²

¹Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ 08542, USA

²Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

Key Points:

- This study validates specialized boundary condition treatments in CNN models to reduce boundary artifacts in ocean parameterizations.
- This approach can be applied directly to already trained CNN models to ensure accurate and stable implementation of mesoscale eddies parameterizations.
- Replicate padding outperforms zero padding by minimizing boundary artifacts and preventing extreme values that compromise simulations.

Abstract

This study addresses the boundary artifacts in machine-learned (ML) parameterizations for ocean subgrid mesoscale momentum forcing, as identified in the online ML implementation from a previous study (Zhang et al., 2023). We focus on the boundary condition (BC) treatment within the existing convolutional neural network (CNN) models and aim to mitigate the "out-of-sample" errors observed near complex coastal regions without developing new, complex network architectures. Our approach leverages two established strategies for placing BCs in CNN models, namely zero and replicate padding. Offline evaluations revealed that these padding strategies significantly reduce root mean squared error (RMSE) in coastal regions by limiting the dependence on random initialization of weights and restricting the range of out-of-sample predictions. Further online evaluations suggest that replicate padding consistently reduces boundary artifacts across various re-trained CNN models. In contrast, zero padding sometimes intensifies artifacts in certain re-trained models despite both strategies performing similarly in offline evaluations. This study underscores the need for BC treatments in CNN models trained on open water data when predicting near-coastal subgrid forces in ML parameterizations. The application of replicate padding, in particular, offers a robust strategy to minimize the propagation of extreme values that can contaminate computational models or cause simulations to fail. Our findings provide insights for enhancing the accuracy and stability of ML parameterizations in the online implementation of ocean circulation models with coastlines.

Plain Language Summary

This study focuses on improving machine learning (ML) models used to predict ocean forces near coastlines, where errors arise because these models lack information in the area. We investigated how boundary conditions are handled in existing convolutional neural network models to reduce these errors without creating complex new architectures. By using two methods, i.e., zero padding and replicate padding, we found that replicate padding significantly decreases prediction errors in coastal areas. While zero padding sometimes worsens issues in certain models, our results show that replicate padding is more reliable for effectively minimizing extreme value errors. This work highlights the importance of proper boundary condition treatment in ML models for coastal applications, ultimately aiming to enhance the accuracy and reliability of ocean circulation predictions.

1 Introduction

Even with advances in computing over recent decades, climate models have finite resolution and must parameterize unresolved, subgrid-scale processes. Historically, these parameterizations employ a mix of theory and empirical approaches (e.g., for ocean circulation, Gent et al., 1995; Griffies et al., 1998; Juricke et al., 2017), but are imperfect so that the representation of subgrid processes continues to be a major source of bias and errors in climate projections (Stevens & Bony, 2013; Hewitt et al., 2020).

Recently, the use of machine learning methods has emerged as a promising tool for developing subgrid parameterizations in numerical models of both the atmosphere (Rasp et al., 2018; Beucler et al., 2021; Yuval et al., 2021; Wang et al., 2022; Shamekh et al., 2023) and ocean (Bolton & Zanna, 2019; Zanna & Bolton, 2020; Guillaumin & Zanna, 2021; Sane et al., 2023; A. Ross et al., 2023; Bodner et al., 2023; Perezhigin et al., 2024). Among machine learning architectures employed for parameterization of subgrid fluxes in climate models, convolutional neural networks (CNNs) have become increasingly popular due to their ability to connect local fluxes or tendencies to spatially non-local features (e.g., Bolton & Zanna, 2019; Zanna & Bolton, 2020; Guillaumin & Zanna, 2021; Bodner et al., 2023; Gregory et al., 2024). However, applying CNN-based ML parameterizations presents unique challenges in ocean circulation models which must account for complex boundary conditions and topographical features at the Earth's surface. Un-

like atmospheric models, ocean models must manage dynamic interactions of coastal water with shorelines, where CNNs often struggle to make a prediction, because they operate by sliding fixed-size kernels across images (fields) to extract features. This was highlighted in a recent study by [Zhang et al. \(2023\)](#), where significant boundary artifacts were observed when a CNN, trained to parameterize mesoscale eddies in the open ocean ([Guillaumin & Zanna, 2021](#)), was employed near coasts in an ocean circulation model. These artifacts are likely the "out of sample" problem in ML parameterizations, wherein a network trained on limited data (open ocean) might extrapolate poorly beyond its knowledge base (near the coast). In their study, the out-of-sample predictions of the CNN applied near the boundaries lead to errors that can ultimately propagate across the entire computational domain as the circulation model evolves.

Training a CNN model with global data including the near-coast regions should solve the out-of-sample problem near boundaries. However, this approach has significant challenges. Consider training a CNN to represent an open ocean (deep water) physical process that is unresolved in a coarse ocean model. Many of the eddy-resolving process-studies used to derive parameterizations of mesoscale turbulence do not consider eddy interaction with coasts, and so inherently may not rectify the parameterized physics and fluxes appropriately. Secondly, the shallow depth near the coast might modify the spatial scales of the process requiring even finer resolution in the high-resolution simulation used to obtain training data which could be too costly ([Hallberg, 2013](#)). Thirdly, the training data near the coastline is limited because the majority of the grid points correspond to the open ocean. Finally, the modification of the process near coastlines (e.g. different dynamics, scales, etc.) likely needs more sophisticated CNN models to capture the additional complexity (i.e either or both deeper and wider networks). The machine learning models that we have seen developed so far to parameterize ocean processes are based either on data from idealized simulations ([Bolton & Zanna, 2019; Zanna & Bolton, 2020; A. Ross et al., 2023](#)) or on regional data from open ocean areas in global General Circulation Model (GCM) simulations that exclude land points ([Guillaumin & Zanna, 2021; Bodner et al., 2023](#)).

All this to say, training a new ML model including coastlines remains challenging, and alternative strategies to mitigate the out-of-sample issues near shorelines should be explored. The border effect for CNNs has been extensively studied in image processing, where common remedies include filling values at the image edges, that is padding ([Innamorati et al., 2018; Nguyen et al., 2019; Huang et al., 2021; Yang et al., 2023](#)), rescaling the result of the convolution operation near the borders, that is partial convolution ([Liu et al., 2018](#)), or changing the filter kernel near the borders ([Leng & Thiyyagalingam, 2023](#)). A most straightforward approach to mitigate out-of-sample errors involves providing boundary values for land points, i.e., filling in appropriate values where no ocean field data exists. In previous research, most studies on spatiotemporal problems with boundaries apply simple, explicit rules like periodic boundaries ([Mohan et al., 2020; Guan et al., 2022; A. Ross et al., 2023](#)). The studies specifically focus on boundary treatments for CNNs remain limited ([Alguacil et al., 2021; Durand et al., 2024](#)).

[Zhang et al. \(2023\)](#) already tried zeroing-out land values in the input, or output, features of the whole CNN, but significant boundary artifacts are still observed, indicating that zero padding in the first layer is ineffective in mitigating the out-of-sample problem in coastal water. Drawing inspiration from the image processing community, which goes further and applies input filling for each layer within the network, this paper presents and compares two boundary condition treatments, i.e., zero padding and replicate padding at each layer, designed to reduce shoreline artifacts for a CNN parameterization of mesoscale eddies evaluated both offline and online. The paper is organized as follows. In Section 2, the limitations of CNN-based parameterizations near boundaries and two BC treatments in multilayer CNN models are introduced. Section 3 briefly introduces the CNN model used in this study, evaluating its performance with and without boundary con-

dition treatments in an offline setting. In Section 4, an ocean circulation model employing the ML parameterization with BC treatments is tested against an idealized wind-driven double gyre case, demonstrating the importance of boundary treatments in online CNN implementations when land points are present. Finally, conclusions and ideas for future study are discussed in Section 5.

2 Methods

In this section, we discuss the limitations of ML parameterizations using classic CNN models near ocean shorelines and introduce two straightforward yet effective methods for managing boundary conditions in multilayer CNN models.

2.1 Limitations of CNN-based parameterizations near boundaries

CNNs function by sliding fixed-size kernels over images to extract spatially-local features. Discontinuous values of a physical field across a land-sea boundary, which are not sampled in training, will lead to undetermined outputs and limit their effectiveness in environments with intricate spatial boundaries.

Consider an input data field illustrated in Figure 1(a) where the northwest portion consists of land points (grey), and the remaining points are ocean points. In this example, the ocean points have known values of 0.5, while the land points contain unknown (and physically meaningless) values. Typically, without special BC treatment, a value of zero is applied on the land points through masking of inputs (multiplying by one or zero), as shown in Figure 1(b)i. In the case of a one-layer CNN, this masking is equivalent to setting a Dirichlet boundary condition, where an input field y is set to zero at the boundary point b , that is $y|_b = 0$. However, merely setting the land points to zero before the first convolutional layer can introduce bias in multilayer CNN models.

Consider a two-layer CNN model, featuring two consecutive convolutional layers, each with a 3×3 kernel and uniform weights of 1. The first convolutional layer processes a 5×5 input field around an ocean point (the blue cell in Figure 1(a)), and assigns values to all cells in the output 3×3 field including land points. In this example, the land cells were assigned values of 0.5, 2, and 2 (as indicated in the gray cells of Figure 1(b)ii). These values in land cells do not represent any physical value or boundary conditions. These artificial values at the land points then influence the computation of the value in the ocean point when the 3×3 stencil passes the second convolutional layer (Figure 1(b)iii). This propagating contamination from zeroed inputs highlights the need for more nuanced BC treatments in CNN-based models used for spatial parameterizations.

2.2 Special treatments of boundary conditions in multilayer CNNs

To address the artifacts introduced by values at land points, we have implemented two padding strategies that incorporate information into these points at each convolutional layer in multilayer CNN models. It should be noted that the term 'padding' used here differs from the traditional usage in CNN terminology, where 'padding' typically refers to adding values around the borders of the input images. In our context, 'padding' specifically denotes the replacement of values at points designated as land within the land masks, which may include locations around or within the borders of the input image.

The first padding strategy we consider is "zero padding". This is the simplest approach, where values are replaced by zeros on land points for each layer. To illustrate, consider the scenario described in Section 2.1, where no boundary condition treatments were initially applied other than zero masking the land values of the input layer. This results in the values 0.5, 2, and 2 at land points after the first convolutional layer (Figures 1(b,c)ii). These land values are reset to zero before the subsequent layer (Figure 1(c)iii),

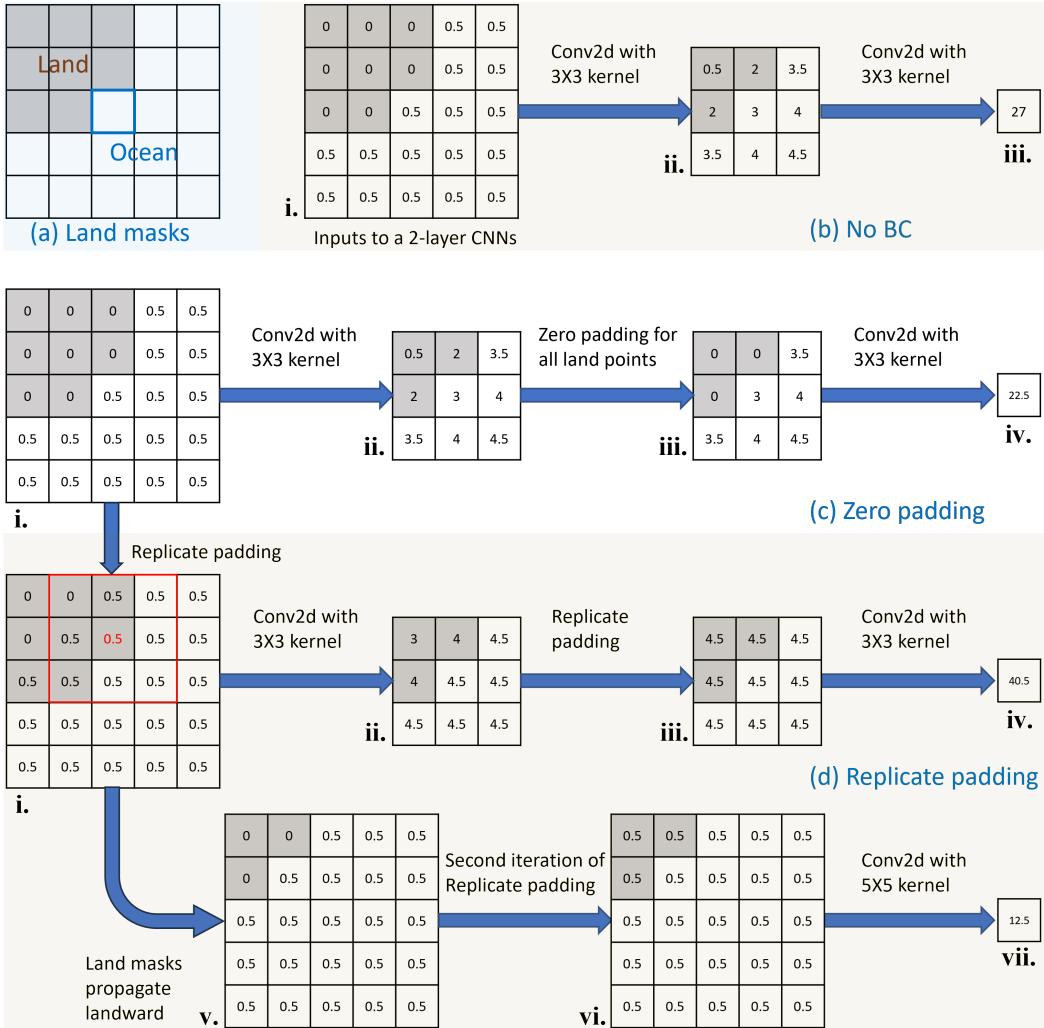


Figure 1. Examples of three boundary condition strategies employed in an idealized two-layer CNN where all weights are set to 1: (a) the layout of land masks; (b) no specific treatment at land points (no padding); (c) filling land points with zeros (zero padding); (d) filling land points with values averaged from the nearest ocean points (replicate padding).

thus ensuring that a Dirichlet-like boundary condition is maintained at each convolutional layer of the CNN.

The second strategy we consider is "replicate padding", where the value at a land point is calculated as the average of values from the nearest ocean points. This method approximates a Neumann boundary condition, where $(\partial y / \partial \mathbf{n})|_b = 0$, and \mathbf{n} represents the vector normal to the boundary. For example, the red box in Figure 1(d)i depicts a 3×3 stencil used to compute the value for the central land point, averaging the values from neighboring ocean points, i.e., 0.5 from the ocean points northeast, east, southeast, and south to the land point. This averaging process is repeated after each convolutional layer to consistently apply a quasi-Neumann boundary condition across all layers of the CNN.

2.3 Replicate padding for larger kernels size

In Section 2.2, we explored two padding strategies applied to sweeping a 3×3 kernel over the computational domain. In many CNN architectures, larger kernels such as 5×5 and 7×7 might also be employed. For zero padding, no additional effort is required to assign a value of zero to land points encompassed by a larger kernel.

For larger kernels using replicate padding, iterations of replication are applied to fill values beyond the first layer of land points. Figure 1(d)vii illustrates the mechanism of replicate padding with a 5×5 kernel. After the first application of replicate padding on the original image, the field with the first layer of replicated land points is established (Figure(d)i). To address the values beyond these first-layer land points, we propagate the land masks landward, creating a new field of land masks (the stencil with three gray cells in Figure 1(d)v). A second replicate padding iteration follows in Figure 1(d)vi, updating the values in the land points based on the nearest "ocean points", as indicated by the new land masks.

This iterative process of propagating land masks and updating land values is necessary when using larger kernels in CNN models. It ensures that all land points within the kernel reach are appropriately filled, maintaining the integrity of the computational model across various kernel sizes.

3 Offline evaluations of CNN+BC treatments

To illustrate the effectiveness of BC treatments in CNN models, we employ an existing CNN model and test it against a global dataset from a high-resolution GCM ocean simulation. In this section, we first outline the CNN model adopted for this study. Then we compare the offline performance of the CNN model with and without the implementation of BC treatments.

3.1 CNN model and dataset descriptions

The CNN model used in this study is the stochastic-deep learning model from [Guillaumin and Zanna \(2021\)](#) (hereafter referred to as GZ21). The model was trained on the high-resolution surface horizontal velocities \mathbf{u} from GFDL CM2.6 ocean simulations, spanning over 7,000 days. The surface velocities \mathbf{u} , with the nominal grid size of $1/10^\circ$ and sampled daily, were filtered and coarse-grained to yield $\bar{\mathbf{u}}$. The subgrid momentum forcing is diagnosed as

$$\mathbf{S} = (\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} - \overline{(\mathbf{u} \cdot \nabla) \mathbf{u}} \quad (1)$$

where the overbar indicates the horizontal filtering and coarse-graining, and ∇ is the horizontal gradient. GZ21 was trained using the first 80% of the data (approximately 16 years), drawn from selected four regions to represent different dynamical regimes. The test dataset

for this section comprises the remaining 20% (approximately 4 years), covering the global domain.

GZ21 is structured as a fully convolutional neural network with eight convolutional layers. The kernel sizes for the first two layers are 5×5 , and 3×3 for the subsequent layers. The layers contain 128, 64, 32, 32, 32, 32, 32, and 4 filters, respectively, each of the first 7 layers followed by ReLU activation functions. The loss function employed is the full negative Gaussian log-likelihood, which estimates the distribution of subgrid momentum forcing given the local velocity field. The CNN outputs both the mean and standard deviation of this distribution, $S_{C,i,j}^{(mean)}$ and $S_{C,i,j}^{(std)}$, where the stochastic subgrid momentum forcing is calculated as

$$S_{C,i,j} = S_{C,i,j}^{(mean)} + \epsilon_{C,i,j} \cdot S_{C,i,j}^{(std)}; \quad C = x, y; \quad i = 1, \dots, M; \quad j = 1, \dots, N. \quad (2)$$

Here, i and j are the grid spatial indices, M and N are grid sizes in two directions, C indicates the component of momentum forcing (zonal "x" or meridional "y"), and $\epsilon_{C,i,j}$ are random 2D fields sampled from the standard normal distribution, independent for each grid cell, zonal/meridional component, vertical layer, and time step. For further details on model training and data generation, see Section 2 of [Guillaumin and Zanna \(2021\)](#).

3.2 Metrics for offline evaluation

To assess the accuracy of the CNN predictions, we employ the standard root mean square error (RMSE) to measure the absolute error between the predicted values and the ground truth. The RMSE is time-averaged at each location as follows

$$RMSE_{C,i,j} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(S_{C,i,j,t}^{(mean)} - S_{C,i,j,t}^{(true)} \right)^2}; \quad C = x, y \quad (3)$$

where t is the time index of the snapshots and T is the total number of snapshots (days) in the test dataset. The RMSE averaged over both time and space is given by

$$RMSE_C = \sqrt{\frac{1}{MNT} \sum_{i=1}^M \sum_{j=1}^N \sum_{t=1}^T \left(S_{C,i,j,t}^{(mean)} - S_{C,i,j,t}^{(true)} \right)^2}; \quad C = x, y \quad (4)$$

Additionally, we employ a R^2 coefficient, as outlined in [Guillaumin and Zanna \(2021\)](#), as a measure similar to the correlation between the predictions and the truth. Values close to 1 signify strong predictions, while values near 0 indicate weaker predictions. The time-averaged R^2 at each location is calculated as

$$R_{C,i,j}^2 = 1 - \frac{\sum_{t=1}^T \left(S_{C,i,j,t}^{(mean)} - S_{C,i,j,t}^{(true)} \right)^2}{\sum_{t=1}^T S_{C,i,j,t}^{(true)2}}; \quad C = x, y \quad (5)$$

The R^2 averaged over both time and space is determined by

$$R_C^2 = 1 - \frac{\sum_{i=1}^M \sum_{j=1}^N \sum_{t=1}^T \left(S_{C,i,j,t}^{(mean)} - S_{C,i,j,t}^{(true)} \right)^2}{\sum_{i=1}^M \sum_{j=1}^N \sum_{t=1}^T S_{C,i,j,t}^{(true)2}}; \quad C = x, y \quad (6)$$

These metrics provide a quantitative framework for evaluating the performance of GZ21 with or without BC treatments in predicting subgrid momentum forces.

3.3 Tests against global data

GZ21 was trained using data from regions devoid of land, posing a significant out-of-sample problem when predicting subgrid forces near shorelines. GZ21 has a stencil size of 21×21 for predicting subgrid momentum forcing at an ocean grid point. This wide stencil results in a broad coastal band (within 10 cells from the shore, approximately 60-100 km in distance when eddy-permitting resolution is applied), where the stencil includes land points, potentially affecting predictions. In their offline evaluation on global datasets, [Guillaumin and Zanna \(2021\)](#) excluded these problematic points, focusing model evaluation exclusively on open ocean areas.

In our analysis, we divide the global domain into two distinct areas: the open ocean domain and the coastal domain. RMSE maps in Figure 2 (plots a and b) illustrate the absolute prediction errors without additional BC treatments for each domain. In this evaluation, subgrid forcing predictions in one direction are sufficiently representative of the predictions in both directions; thus, only zonal predictions are presented in this section. The space- and time-averaged prediction errors in the coastal domain are 3 times higher than those in the open ocean (b versus a). The errors increase as fewer layers of coastal water points are included in the evaluation, with the errors in the layer of grid points closest to the shore being an order of magnitude higher than those in the open ocean (Table 1).

RMSE maps (plots c and d) in Figure 2 show the result of implementing either zero padding or replicate padding which reduces the errors by approximately 11-12%, with both strategies performing comparably. The reduction percentage increases to about 25% for the layer closest to the shore (Table 1). These padding strategies do not alter predictions in the open ocean domain. The changes are not clearly apparent in the global maps of RMSE but zooming into the Malay Archipelago (plots (e) to (g) of Figure 2) shows responses in ocean cells with significant numbers of land in the vicinity. These detailed views indicate that immediate proximity to land led to larger errors without padding, and that as land cells occupy a smaller fraction of the 21×21 stencil (i.e. further away from the coast), the padding has less impact.

Number of Layers	No Padding RMSE to truth ($10^{-7} ms^{-2}$)	Zero Padding		Replicate Padding	
		RMSE to truth ($10^{-7} ms^{-2}$)	Improved %	RMSE to truth ($10^{-7} ms^{-2}$)	Improved %
10	0.883	7.882	10.72	7.717	12.58
7	1.029	0.905	12.05	0.883	14.19
5	1.188	1.028	13.47	0.999	15.88
4	1.305	1.114	14.64	1.081	17.16
3	1.472	1.232	16.30	1.192	19.02
2	1.734	1.404	19.03	1.358	21.69
1	2.209	1.677	24.08	1.634	26.03

Table 1. Offline evaluation results by inferring the subgrid forcing $S_x^{(mean)}$ using GZ21 in the coastal domain with different BC padding strategies, based on the metric of RMSE averaged over both time and space, as well as the improved percentage of using these BC padding strategies. 'Number of Layers' refers to the number of layers of water points near coastlines that are included in the evaluation.

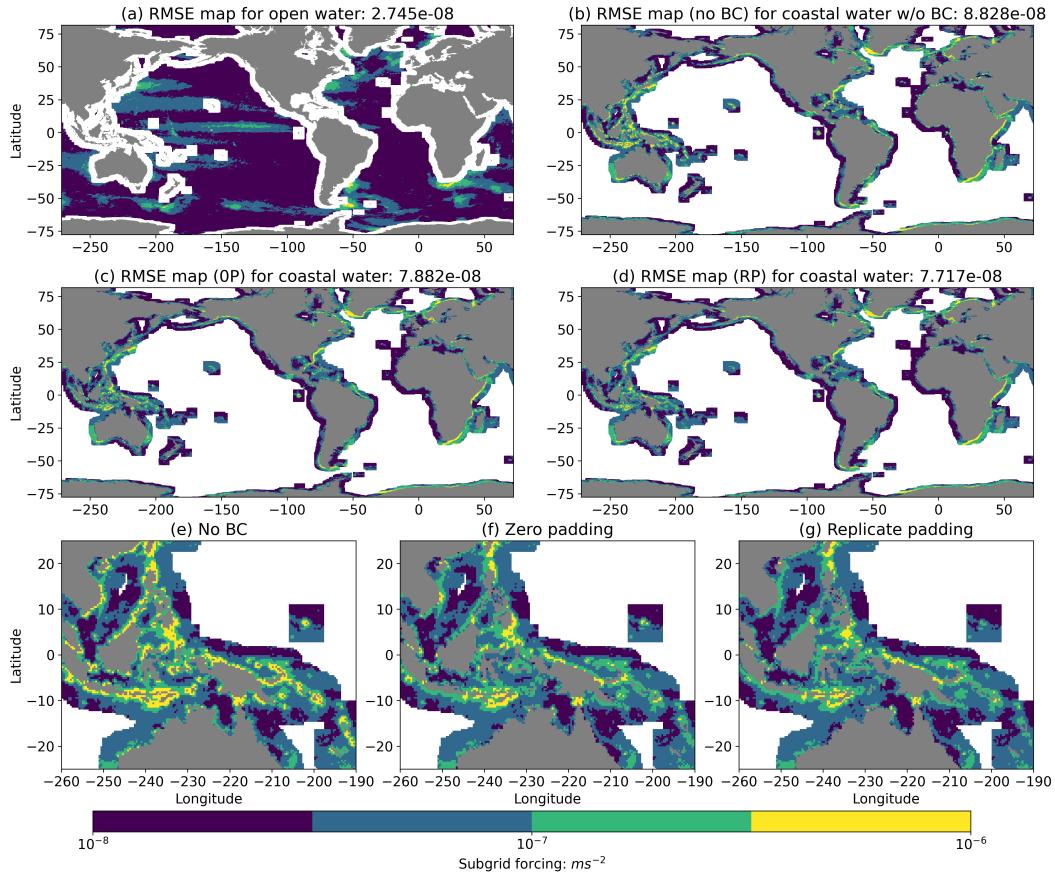


Figure 2. Time-averaged global RMSE maps to true forcing for GZ21 inference of subgrid forcing $S_x^{(\text{mean})}$ in open ocean domain (a), in coastal domain without special BC treatment (b), with zero padding (0P) treatment (c) and replicate padding (RP) treatment (d). (e) to (g) are zoom-in maps of (b) to (d), respectively. The subtitles include the RMSE averaged over both time and space.

The significant errors observed in coastal domain predictions are a manifestation of the out-of-sample problem. The particular predictions in the coastal domain should vary from training to training due to the random initialization of weights. To check this, we retrained a new CNN model, GZ21-T2, following the exact procedures described in [Guillaumin and Zanna \(2021\)](#). Figure 3 compares the root mean square differences (RMSD) for coastal domain predictions between GZ21 and GZ21-T2, calculated as

$$RMSD_{x,i,j}^{(\text{model})} = \sqrt{\frac{1}{T} \sum_{t=1}^T \left(S_{x,i,j,t}^{(\text{GZ21,mean})} - S_{x,i,j,t}^{(\text{GZ21-T2,mean})} \right)^2}; \quad (7)$$

with and without BC treatments. The results indicate that BC treatments effectively reduce the randomness of out-of-sample predictions, indicated by the notably smaller overall differences when BC treatments are applied.

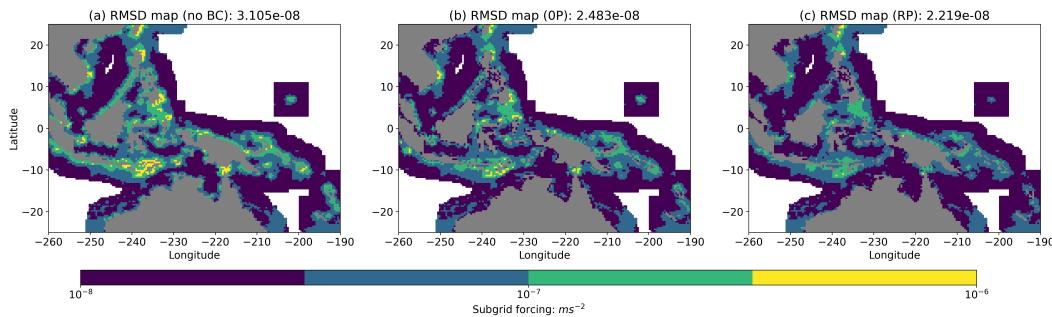


Figure 3. Time-averaged $RMSD^{(\text{model})}$ maps, focusing on the Malay Archipelago, of subgrid forcing $S_x^{(\text{mean})}$ between inference of the original CNN model GZ21 and the retrained CNN model GZ21-T2 with global data (a) without special BC treatment, (b) with zero padding treatment, and (c) replicate padding treatment. The RMSD values in subtitles are global averaging over both time and space.

The R^2 coefficient, as described in Section 3.2, serves as an effective indicator of the overall performance of ML models within the area of interest. The global R^2 values averaged over both time and space (Eq. 6) are 0.381 for GZ21 without BC treatments, 0.563 for GZ21 with zero paddings, and 0.558 for GZ21 with replicate padding. The variability in R^2 averaged in space reflects varying levels of bias from out-of-sample predictions near the coasts under different BC treatments. For example, at a location ($-74.15^\circ, 39.41^\circ$) near Atlantic City, New Jersey, GZ21 tends to predict significantly higher absolute values. Figure 4(a) contrasts the predicted forcing from GZ21 with the true forcing, where the blue lines represent true forcing, the orange lines represent the GZ21 predictions, and the dashed green lines represent the 95% confidence interval. The true forcing is in the range of $[-4, 0] \times 10^{-7} \text{ms}^{-2}$, while the mean part of the forcing from the GZ21 predictions is in the range of $[-20, 0] \times 10^{-7} \text{ms}^{-2}$. The BC treatments can effectively reduce the prediction range, and both strategies reduce the range of forcing to $[-4, 0] \times 10^{-7} \text{ms}^{-2}$. These plots highlight the efficacy of BC treatments in narrowing the prediction range to more closely align with the actual measurements, despite not perfectly matching the truth.

To verify the reproducibility of GZ21, we repeated the training process 6 additional times (GZ21-T3 to GZ21-T8). It is important to note that each model may exhibit significant performance variations in coastal domains (out-of-sample predictions) due to the random initialization of weights during each training session. Table 2 lists the global R^2 coefficients, coastal RMSE between predictions and truth, and coastal RMSD between

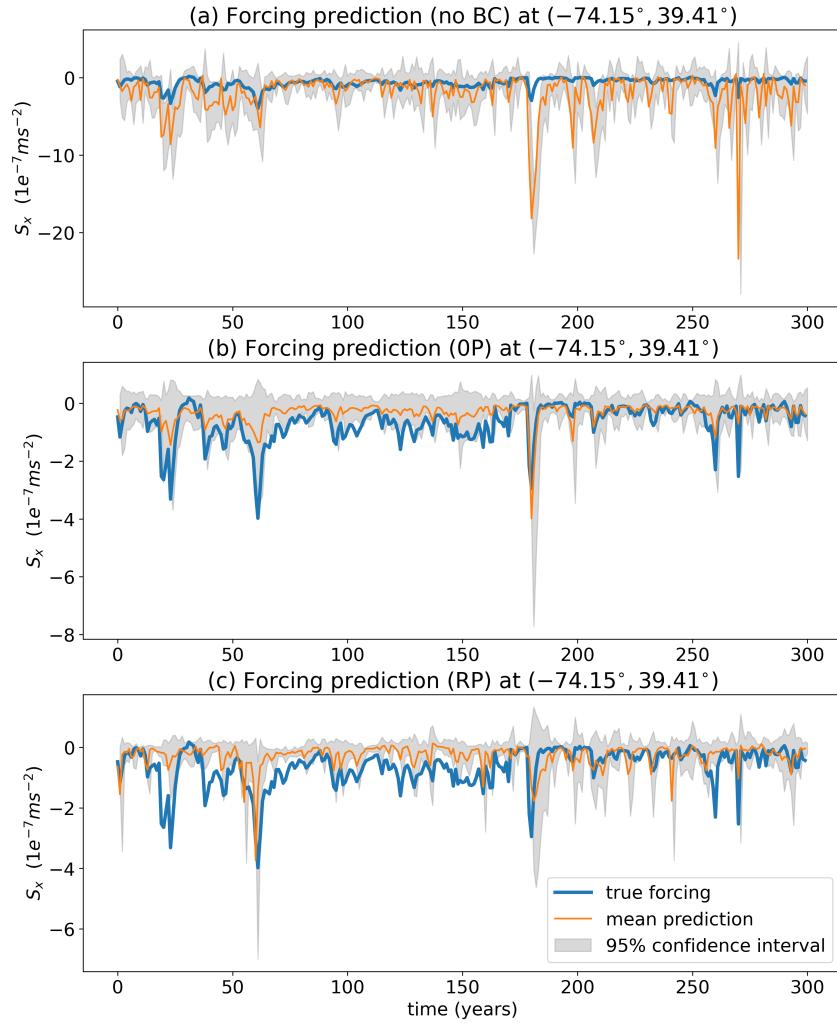


Figure 4. Time series of true forcing (blue), of the mean predictions ($S_x^{(\text{mean})}$, orange), and of the 95% confidence interval ($\pm 1.96 S_{x,i,j}^{(\text{std})}$, shaded in grey) at a location $(-74.15^\circ, 39.41^\circ)$ near Atlantic City, New Jersey.

different model predictions, both with and without BC treatments. The standard deviation of the R^2 column for no padding, zero padding, and replicate padding are 0.0179, 0.0049, 0.0056, respectively. The significantly lower numbers from the offline evaluations with BC treatments confirm the improved reproducibility of GZ21 with BC treatments compared to GZ21 without BC treatments.

Model Name	R^2	No Padding		Zero Padding		Replicate Padding	
		RMSE to truth (10^{-8} m s^{-2})	RMSD to GZ21 (10^{-8} m s^{-2})	RMSE to truth (10^{-8} m s^{-2})	RMSD to GZ21 (10^{-8} m s^{-2})	R^2	RMSE to truth (10^{-8} m s^{-2})
GZ21	0.381	8.828	/	0.563	7.882	/	0.558
GZ21-T2	0.385	8.861	3.105	0.559	7.783	2.483	0.546
GZ21-T3	0.401	8.846	3.210	0.557	7.823	2.617	0.540
GZ21-T4	0.419	8.635	2.969	0.572	7.781	2.505	0.545
GZ21-T5	0.367	8.874	2.990	0.561	7.832	2.472	0.547
GZ21-T6	0.400	8.679	3.062	0.558	7.760	2.458	0.546
GZ21-T7	0.367	8.938	3.110	0.558	7.891	2.558	0.540
GZ21-T8	0.395	8.793	3.042	0.559	7.878	2.532	0.546

Table 2. Offline evaluation results by inferring the subgrid forcing $S_x^{(mean)}$ using original CNN model GZ21 or retrained model GZ21-T2 to GZ21-T8 with different BC padding strategies, based on two metrics of global R^2 and coastal RMSE averaged over both time and space, as well as coastal RMSD of the forcing prediction between retrained models and GZ21.

4 Online implementations of CNN+BC with MOM6

The ultimate goal of developing ML parameterizations is to improve the online solution of numerical models. While the overall offline performance is excellent in Section 3, it does not assure comparable online success (A. S. Ross et al., n.d.). When these parameterizations are incorporated into a coarse-resolution ocean model and executed over extended periods, local errors introduced by the parameterization can accumulate and/or spread throughout the simulation. This can subsequently contaminate the entire domain or, in severe cases, cause the simulation to fail.

In this section, we further explore the effectiveness of boundary condition (BC) treatments within the GZ21 parameterization for online inference. We test the model both with and without BC treatments in an idealized case for which we can afford to run a fine-resolution "truth": a wind-driven double gyre (Hallberg & Rhines, 2000). We will first briefly outline the ocean model used for this study and the setup of the case. Then we will examine the evaluation results and discuss the computational costs associated with implementing BC treatments in online simulations.

4.1 Ocean model and case setup

The numerical model employed in this study is the Modular Ocean Model version 6 (MOM6) (Adcroft et al., 2019), the ocean component of the NOAA coupled global climate and earth system models developed at GFDL. We apply MOM6 under the assumption of an adiabatic limit with no buoyancy forcing, which simplifies the ocean dynamics into the stacked shallow water equations. This assumption facilitates the testing of the ML parameterization of subgrid momentum forcing (Eq. 1) in an idealized setting of a primitive equation model. The governing equations are discretized on a C-type staggered grid, positioning the velocity components ($\bar{\mathbf{u}}$ in Eq. 1) on cell faces and the subgrid forcing (\mathbf{S} in Eq. 1) at the cell centers. During the implementation of the ML parameterization, velocity components from MOM6 are interpolated to the cell centers and then used as inputs to the ML model to infer subgrid forcing, which is then interpolated

back to the cell faces. Further details on the model descriptions are available in Section 2.1 of [Zhang et al. \(2023\)](#).

For this study, the ocean model is configured to simulate two idealized wind-driven double gyre scenarios. The first configuration (hereafter referred to as C1) features a bowl-shaped basin ([Hallberg & Rhines, 2000](#)) extending from 0° to 22° in longitude and from 30° to 50° in latitude, with a depth ranging from -2000m to 0m in vertical. A vertical wall is placed at the southern boundary. The flow includes two vertical layers with constant water density in each layer, and no computations involving equation of state, temperature and salinity. The circulation is driven by wind and balanced by bottom friction. The simulations start from rest and continue for a duration of 10 years. Further details can be found in Section 3.1 of [Zhang et al. \(2023\)](#).

The second configuration (hereafter referred to as C2) introduces a box-shaped island in the center of the domain based on the first configuration, located between 8.5° to 13.5° in longitude and 37.5° to 42.5° in latitude (see Figure 17 in [Zhang et al., 2023](#)). This island represents a significant topographic obstacle in the path of the wind-driven jet and we expect the abrupt nature of the obstacle to test the limits of the ML parameterization near boundaries.

The evaluations are conducted using a coarse grid model with $1/4^\circ$ horizontal resolution (hereafter referred to as R4), which is "eddy-permitting" but not fine enough to resolve all mesoscale eddy dynamics. By applying the ML parameterization in R4 (hereafter referred to as R4-P), we compare its performance to that of a fine grid model with a $1/32^\circ$ horizontal resolution (hereafter referred to as R32), which is capable of fully capturing mesoscale eddy processes. In this section, the total Kinetic Energy (KE) of the flow is the only metric used to quantitatively evaluate the online performance of the CNN model.

4.2 Results with various BC treatments

The study of [Zhang et al. \(2023\)](#) found that when GZ21 is applied in MOM6 to predict the subgrid mesoscale momentum forcing near boundaries, it generates artifacts that cause the over-energization of the flow. The structures highlighted within black rectangles in Figure 5 illustrate the typical artifacts, which are absent in the higher-resolution model R32.

We first consider online evaluations of the original GZ21 model (not retrained) but with different BC treatment during inference. Figure 6(a-d) shows the relative vorticity snapshots of the upper flow from R32, R4-P without padding, R4-P with zero padding, and R4-P with zero and replicate padding strategies in the C1 scenario. Figure 6(e,d) compares the time series of KE in both the upper and lower layer flows under different padding strategies, with line colors corresponding to the edges of each plot from (a) to (d). [Zhang et al. \(2023\)](#) demonstrated that the parameterizations based on GZ21 tend to over-energize the flow in the upper layer while under-energizing the flow in the lower layer. Neither of the two padding strategies affect the overall energy injection in each layer, but they are effective in mitigating artifacts near the southern boundary. In C1, the performance of both treatments appears similar.

The ML parameterizations were also tested in C2. Figure 7(a-d) shows the relative vorticity snapshots from the ground truth and ML parameterizations with different BC treatments. Without special BC treatments, obvious sheared structures appear both around the box island, as well as at the southern boundary as we observe in C1 (Figure 7(b)). Zero padding removes the relatively weak artifacts near the southern boundary but does not affect the strong sheared structures around the island (Figure 7(c)). In contrast, replicate padding effectively eliminates the artifacts in both boundary regions, aligning the better relative vorticity snapshot with the ground truth. This is also evi-

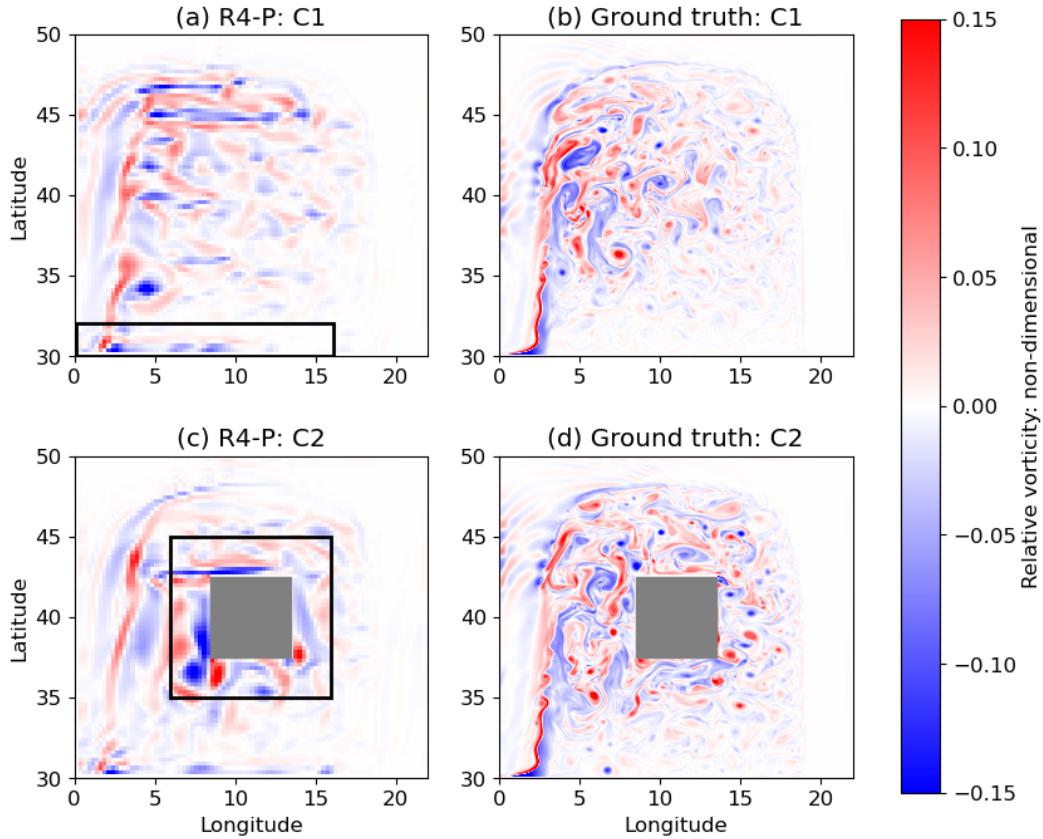


Figure 5. Snapshots of the upper layer relative vorticity at the end of C1 and C2 simulations from the coarse resolution model with ML parameterizations R4-P (a,c) and the fine resolution R32 (b,d). The ML parameterizations do not use special BC treatments. The black rectangle indicates the region where the unrealistic eddies are generated.

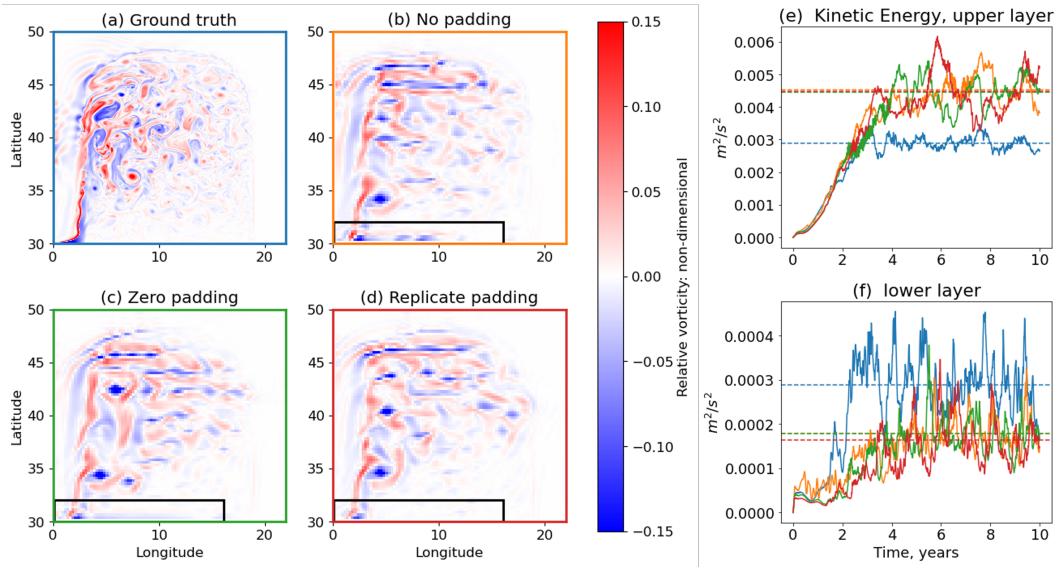


Figure 6. (a)-(d): Snapshots of the upper layer relative vorticity at the end of C1 simulations from: (a) the fine resolution R32; (b) the coarse resolution model using ML parameterization R4-P without special BC treatment; (c) with zero padding strategy; or (d) with replicate padding. (e) & (f): Comparison of KE time series for the flow upper layer and lower layer between the four simulations from (a) to (d). The dashed lines are the time-mean values of KE over the last 5 years and the colors of solid lines correspond to the edge colors of plots (a) to (d), where blue is for ground truth, orange for no padding, green for zero padding and red for replicate padding. The black rectangle indicates the region where the unrealistic eddies are generated.

dent in the time series of KE (Figure 7(e,f)), where no treatment results in a more energetic flow, zero padding reduces energy from artifacts, and replicate padding closely matches the KE to that in the ground truth. Thus, replicate padding is the most effective approach considered in C2. It should be noted that the zonal elongation of eddies is observed in Figures 6(b-d) and 7(b-d). We hypothesize that the four regions selected in GZ21 exhibit a tendency towards zonal flows, as discussed in Section 4.4 of [Zhang et al. \(2023\)](#). The discussion of this issue is beyond the scope of this paper.

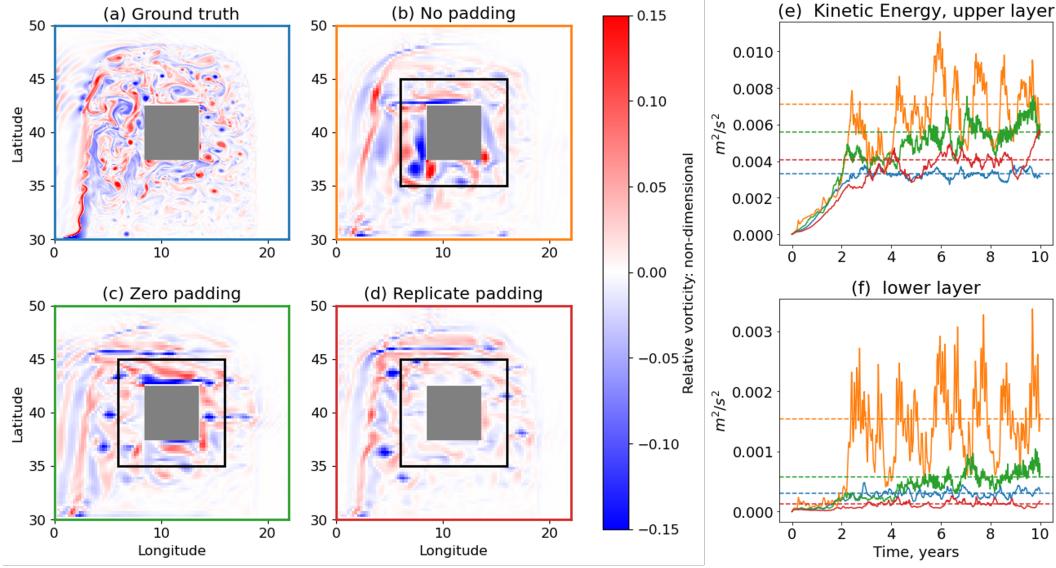


Figure 7. Same to Figure 6 but tested on C2.

We now consider whether the padding strategy can reduce the sensitivity of network performance to the random initialization of weights in training processes; the GZ21-T2 model is retrained with the same data as GZ21 and without padding used during training (just like what GZ21 does), but reevaluated with the various padding treatments. Figure 8 shows results with the retrained model in C1. It is interesting to see that the zero padding strategy fails to eliminate the artifacts near the southern boundary (Figure 8(c)) as it does with GZ21. After a 5-year run, the flow becomes even more energetic compared to simulations that did not use any BC treatments. In contrast, replicate padding with GZ21-T2 performs well in C1 for artifact eliminations. Tests are also conducted on configuration C2; however, simulations using GZ21-T2 with no padding and zero padding failed in C2, as excessive energy was injected into the flow, causing the models to blow up. This issue of over-energization near boundaries is not isolated to GZ21-T2; some other retrained models (models in Table 2) also struggle to eliminate artifacts using the zero padding method. The KE time series depicted in Figure 9(c,d) demonstrate that using zero padding in GZ21-T4 and GZ21-T8 similarly leads to poorer predictions of energy injection in configuration C1. In contrast, the replicate padding BC treatment consistently reduces energy injection to a more reasonable range across all retrained models, aligning more closely with the ground truth.

In addition to the effectiveness of the BC strategies, their computational cost is also a critical consideration. The zero padding strategy does not significantly increase the computational load compared to the approach without BC treatments. Implementing zero padding across each CNN convolutional layer for GZ21 results in approximately a 10% increase in wall clock time for CNN inference. In contrast, the process of filling the nearest value to the land points is considerably more costly due the stencil operations needed

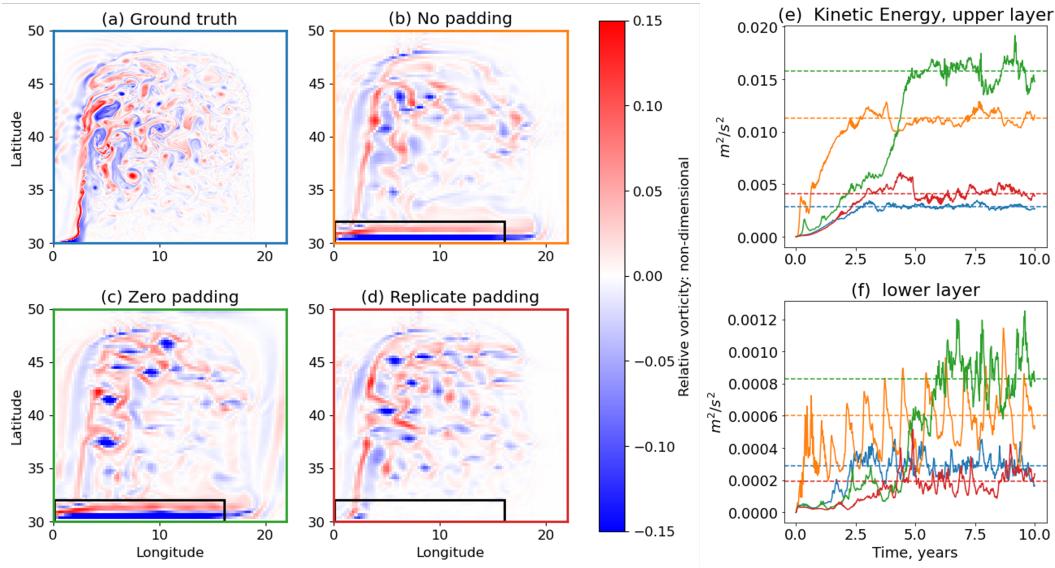


Figure 8. Same to Figure 6 but tested using GZ21-T2 for the ML parameterizations.

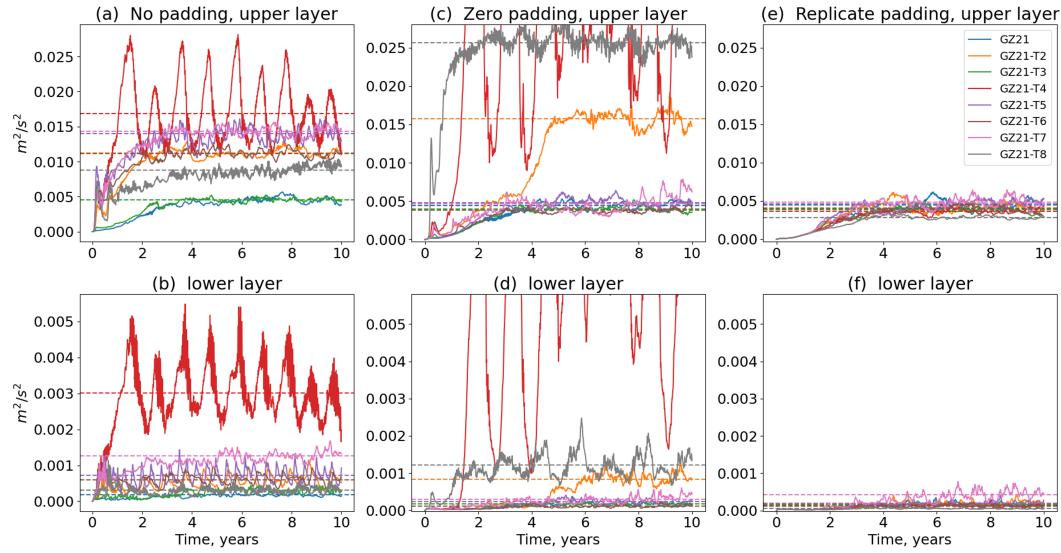


Figure 9. Comparison of KE time series for the flow upper layer and lower layer between the simulations using GZ21 and retrained CNN models. The first column of plots (a) and (b) represent the simulation results without any BC treatment, the second column of plots (c) and (d) represent the results with zero padding BC treatment, and the third column of plots (e) and (f) represent the results with replicate padding BC treatment. The dashed lines are the time-mean values of KE over the last 5 years and each row shares the same axis range to better compare the results with different BC strategies.

to propagate values. We tried optimizing this operation as a pre-computed sparse matrix, which significantly improved performance, yet only to within a factor of 2 compared to the no-padding model. While this cost represents a substantial increase compared to the computational costs associated with no padding and zero padding, it renders the strategy feasible. Furthermore, optimizations in sparse matrix operations could potentially further reduce the time required for inference.

5 Conclusions

This study was motivated by a notable issue identified in the previous study of [Zhang et al. \(2023\)](#), where distinct artifacts near boundaries were observed in the online implementation of the ML parameterization based on GZ21. These artifacts were suspected to be generated by out-of-sample predictions near boundaries because GZ21 was only trained in the open ocean. Developing a new network capable of accommodating the complex flow regimes near intricate shorelines likely requires substantial effort and sophisticated architectures with uncertain convergence. We instead explored the use of specialized BC treatments to use with existing CNN models to address over-energization caused by out-of-sample predictions near coasts.

Our offline evaluations of the existing CNN model GZ21 with the global dataset CM2.6, demonstrated that both zero padding and replicate padding strategies can effectively reduce RMSE near coastlines. The significant RMSE in coastal domains is primarily due to the unconstrained out-of-sample predictions in these regions with the original GZ21. For example, force predictions near Atlantic City, New Jersey, indicated that GZ21 without padding tends to produce a wide range of values, whereas GZ21 with zero or replicate padding can mitigate this randomness and narrow the range of out-of-sample predictions. In Addition, it is important to note that the unconstrained out-of-sample error varies significantly among GZ21 and the retained models of GZ21 due to the random initialization of weights during their training processes. Implementing boundary treatments can help effectively restrict the error for all models within a reasonable range.

Online evaluations of two configurations, the wind-driven double gyre (C1) and double gyre with an island in the center (C2), confirmed that using the replicate padding strategy as a BC treatment can effectively eliminate boundary artifacts in the online implementation of ML parameterizations, outperforming both the no padding and zero padding approaches. Our reproducibility tests indicated that GZ21 was fortuitously trained such that its predictions near boundaries do not generate overly strong sheared artifacts, which would lead to excessive energy within the flow or even simulation blowup. Although the no padding strategy was effective in offline evaluations and for several retrained models in this online evaluation, it was not universally useful, whereas the replicate padding strategy proved effective at avoiding artifacts across all retrained models in this study.

The objective of this study is to develop strategies that limit prediction errors in regions where the ML parameterizations lack sufficient ‘knowledge’. Typically, ML parameterizations for ocean subgrid forcing neglect consideration of land points during training due to the complexities of coastal regions and the intricacies of managing land point values. This is true for many conventional parameterizations also. In essence, employing replicate padding in a CNN model for coastal regions minimizes implied gradients near coasts, which naturally reduces the magnitude of predictions from the model. This approach offers a viable pathway whereby an existing CNN model trained on open water data can be used to predict forces in coastal areas without generating strongly anomalous outputs. This capability is crucial in online implementations of ML parameterizations because any extreme value introduced by CNN inference can eventually propagate throughout the domain, contaminating the solution, or even leading to failure of the simulation.

Open Research

The source code of the MOM6 version used for implementing the ML parameterization is accessible through Zenodo ([Hallberg et al., 2024](#)), while the CNN model files used for the online evaluation in this study (GZ21) can also be accessed via Zenodo ([Zhang, 2024](#)). The files for offline global evaluations can be accessed via Zenodo ([Zhang & Guillaumin, 2024](#)). To facilitate the setup process for the wind-driven double gyre case in the study, we have made the setup files available online ([Zhang, 2023](#)).

Acknowledgments

We thank all members of the M²LInES team for helpful discussions and their support throughout this project. We thank Qian Shao and Wenda Zhang for useful comments on a draft of this manuscript, and Arthur Guillaumin for assistance with the networks. This research received support through Schmidt Sciences, LLC. AA was also supported by award NA18OAR4320123, from the National Oceanic and Atmospheric Administration (NOAA), U.S. Department of Commerce and which funded the Princeton Stellar computer resources used for the inference stage of the research. The statements, findings, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the National Oceanic and Atmospheric Administration, or the U.S. Department of Commerce. This research was also supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

References

- Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., ... others (2019). The GFDL global ocean and sea ice model OM4. 0: Model description and simulation features. *Journal of Advances in Modeling Earth Systems*, 11(10), 3167–3211. doi: <https://doi.org/10.1029/2019MS001726>
- Alguacil, A., Pinto, W. G., Bauerheim, M., Jacob, M. C., & Moreau, S. (2021). Effects of boundary conditions in fully convolutional networks for learning spatio-temporal dynamics. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 102–117). doi: https://doi.org/10.1007/978-3-030-86517-7_7
- Beucler, T., Pritchard, M., Yuval, J., Gupta, A., Peng, L., Rasp, S., ... others (2021). Climate-invariant machine learning. *arXiv preprint arXiv:2112.08440*. doi: <https://doi.org/10.48550/arXiv.2112.08440>
- Bodner, A., Balwada, D., & Zanna, L. (2023). A data-driven approach for parameterizing submesoscale vertical buoyancy fluxes in the ocean mixed layer. *arXiv preprint arXiv:2312.06972*.
- Bolton, T., & Zanna, L. (2019). Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1), 376–399. doi: 10.1029/2018MS001472
- Durand, C., Finn, T. S., Farchi, A., Bocquet, M., Boutin, G., & Ólason, E. (2024). Data-driven surrogate modeling of high-resolution sea-ice thickness in the arctic. *The Cryosphere*, 18(4), 1791–1815. doi: <https://doi.org/10.1016/j.jcpc.2022.111090>
- Gent, P. R., Willebrand, J., McDougall, T. J., & McWilliams, J. C. (1995). Parameterizing Eddy-Induced Tracer Transports in Ocean Circulation Models. *Journal of Physical Oceanography*, 25(4), 463–474. doi: 10.1175/1520-0485(1995)025<0463:PEITTI>2.0.CO;2
- Gregory, W., Bushuk, M., Zhang, Y., Adcroft, A., & Zanna, L. (2024). Machine learning for online sea ice bias correction within global ice-ocean simulations. *Geophysical Research Letters*, 51(3), e2023GL106776. doi: <https://doi.org/10.1029/2023GL106776>
- Griffies, S. M., Gnanadesikan, A., Pacanowski, R. C., Larichev, V. D., Dukowicz,

- J. K., & Smith, R. D. (1998). Isoneutral Diffusion in a z-Coordinate Ocean Model. *Journal of Physical Oceanography*, 28(5), 805–830. (Publisher: American Meteorological Society Section: Journal of Physical Oceanography) doi: 10.1175/1520-0485(1998)028(0805:IDIAZC)2.0.CO;2
- Guan, Y., Chattopadhyay, A., Subel, A., & Hassanzadeh, P. (2022). Stable a posteriori les of 2d turbulence using convolutional neural networks: Backscattering analysis and generalization to higher re via transfer learning. *Journal of Computational Physics*, 458, 111090. doi: <https://doi.org/10.1016/j.jcp.2022.111090>
- Guillaumin, A., & Zanna, L. (2021). Stochastic deep learning parameterization of ocean momentum forcing. *Journal of Advances in Modeling Earth Systems*, 13(9), e2021MS002534. doi: <https://doi.org/10.1002/essoar.10506419.1>
- Hallberg, R. (2013). Using a resolution function to regulate parameterizations of oceanic mesoscale eddy effects. *Ocean Modelling*, 72, 92–103. doi: <https://doi.org/10.1016/j.ocemod.2013.08.007>
- Hallberg, R., Adcroft, A., Marques, G., Ward, M., Hedstrom, K., Shao, A., ... Zhang, C. (2024). chzhangudel/mom6: Second release (forpy_ss/2024.05.01)[software]. Zenodo. doi: <https://doi.org/10.5281/zenodo.7663074>
- Hallberg, R., & Rhines, P. B. (2000). Boundary sources of potential vorticity in geophysical circulations. In R. M. Kerr & Y. Kimura (Eds.), *IUTAM Symposium on Developments in Geophysical Turbulence* (pp. 51–65). Dordrecht: Springer Netherlands. doi: https://doi.org/10.1007/978-94-010-0928-7_5
- Hewitt, H. T., Roberts, M., Mathiot, P., Biastoch, A., Blockley, E., Chassignet, E. P., ... others (2020). Resolving and parameterising the ocean mesoscale in earth system models. *Current Climate Change Reports*, 6(4), 137–152. doi: <https://doi.org/10.5281/zenodo.3685918>
- Huang, Y.-H., Proesmans, M., & Van Gool, L. (2021). Context-aware padding for semantic segmentation. *arXiv preprint arXiv:2109.07854*.
- Innamorati, C., Ritschel, T., Weyrich, T., & Mitra, N. J. (2018). Learning on the edge: Explicit boundary handling in cnns. *arXiv preprint arXiv:1805.03106*.
- Juricke, S., Palmer, T. N., & Zanna, L. (2017). Stochastic subgrid-scale ocean mixing: impacts on low-frequency variability. *Journal of Climate*, 30(13), 4997–5019.
- Leng, K., & Thiyyagalingam, J. (2023). Padding-free convolution based on preservation of differential characteristics of kernels. *arXiv preprint arXiv:2309.06370*.
- Liu, G., Shih, K. J., Wang, T.-C., Reda, F. A., Sapra, K., Yu, Z., ... Catanzaro, B. (2018). Partial convolution based padding. *arXiv preprint arXiv:1811.11718*.
- Mohan, A. T., Lubbers, N., Livescu, D., & Chertkov, M. (2020). Embedding hard physical constraints in neural network coarse-graining of 3d turbulence. *arXiv preprint arXiv:2002.00021*.
- Nguyen, A.-D., Choi, S., Kim, W., Ahn, S., Kim, J., & Lee, S. (2019). Distribution padding in convolutional neural networks. In *2019 ieee international conference on image processing (icip)* (pp. 4275–4279). doi: <https://doi.org/10.1109/ICIP.2019.8803537>
- Perezhogin, P., Zhang, C., Adcroft, A., Fernandez-Granda, C., & Zanna, L. (2024). Implementation of a data-driven equation-discovery mesoscale parameterization into an ocean model. *Journal of Advances in Modeling Earth Systems*, 16(10), e2023MS004104. doi: <https://doi.org/10.1029/2023MS004104>
- Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39), 9684–9689. doi: <https://doi.org/10.1073/pnas.1810286115>
- Ross, A., Li, Z., Perezhogin, P., Fernandez-Granda, C., & Zanna, L. (2023). Benchmarking of machine learning ocean subgrid parameterizations in an idealized model. *Journal of Advances in Modeling Earth Systems*, 15(1). doi:

- <https://doi.org/10.1029/2022MS003258>
- Ross, A. S., Li, Z., Perezhigin, P., Fernandez-Granda, C., & Zanna, L. (n.d.). Benchmarking of machine learning ocean subgrid parameterizations in an idealized model. *Journal of Advances in Modeling Earth Systems*, n/a(n/a), e2022MS003258. doi: <https://doi.org/10.1029/2022MS003258>
- Sane, A., Reichl, B. G., Adcroft, A., & Zanna, L. (2023). Parameterizing vertical mixing coefficients in the ocean surface boundary layer using neural networks. *Journal of Advances in Modeling Earth Systems*, 15(10), e2023MS003890. doi: <https://doi.org/10.1029/2023MS003890>
- Shamekh, S., Lamb, K. D., Huang, Y., & Gentine, P. (2023). Implicit learning of convective organization explains precipitation stochasticity. *Proceedings of the National Academy of Sciences*, 120(20), e2216158120. doi: <https://doi.org/10.1073/pnas.2216158120>
- Stevens, B., & Bony, S. (2013). What are climate models missing? *science*, 340(6136), 1053–1054. doi: <https://doi.org/10.1126/science.1237554>
- Wang, P., Yuval, J., & O’Gorman, P. A. (2022). Non-local parameterization of atmospheric subgrid processes with neural networks. *Journal of Advances in Modeling Earth Systems*, 14(10), e2022MS002984. doi: <https://doi.org/10.1029/2022MS002984>
- Yang, N., Zhong, L., Huang, F., Bao, W., & Yuan, D. (2023). Random padding data augmentation. In *Australasian conference on data science and machine learning* (pp. 3–18). doi: https://doi.org/10.1007/978-981-99-8696-5_1
- Yuval, J., O’Gorman, P. A., & Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letters*, 48(6), e2020GL091363. doi: <https://doi.org/10.1029/2020GL091363>
- Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, 47(17), e2020GL088376. doi: <https://doi.org/10.1002/essoar.10503535.1>
- Zhang, C. (2023). *chzhangudel/double_gyre: Initial release (v1.0.1)[software]*. Zenodo. doi: <https://doi.org/10.5281/zenodo.7663041>
- Zhang, C. (2024). *chzhangudel/forpy_cnn_gz21: Second release (v2.0.0)[software]*. Zenodo. doi: <https://doi.org/10.5281/zenodo.7663061>
- Zhang, C., & Guillaumin, A. (2024). *chzhangudel/gz21: Release for offline evaluations (v0.0.0)[software]*. Zenodo. doi: <https://doi.org/10.5281/zenodo.1395892>
- Zhang, C., Perezhigin, P., Gultekin, C., Adcroft, A., Fernandez-Granda, C., & Zanna, L. (2023). Implementation and evaluation of a machine learned mesoscale eddy parameterization into a numerical ocean circulation model. *Journal of Advances in Modeling Earth Systems*, 15(10), e2023MS003697. doi: <https://doi.org/10.1029/2023MS003697>