Parametric Sensitivities of a Wind-driven Baroclinic Ocean using Neural Surrogates

Yixuan Sun
yixuan.sun@anl.gov
Mathematics and Computer Science Division, Argonne National Laboratory
Lemont, Illinois, USA

Elizabeth Cucuzzella
Tufts University
419 Boston Ave., Massachusetts, USA
ebcucuzzella@gmail.com

Steven Brus
Mathematics and Computer Science Division, Argonne National Laboratory
Lemont, Illinois, USA
sbrus@anl.gov

Sri Hari Krishna Narayanan
Mathematics and Computer Science Division, Argonne National Laboratory
Lemont, Illinois, USA
snarayan@anl.gov

Balasubramanya Nadiga
Los Alamos National Laboratory
Los Alamos, California, USA
balu@lanl.gov

Luke Van Roekel
Los Alamos National Laboratory
Los Alamos, New Mexico, USA
lvanroekel@lanl.gov

Jan Hückelheim
Mathematics and Computer Science Division, Argonne National Laboratory
Lemont, Illinois, USA
jhuckelheim@anl.gov

Sandeep Madireddy
Mathematics and Computer Science Division, Argonne National Laboratory
Lemont, Illinois, USA
smadireddy@anl.gov

Patrick Heimbach
University of Texas at Austin
Austin, Texas, USA
heimbach@oden.utexas.edu

ABSTRACT

Numerical models of the ocean and ice sheets are crucial for understanding and simulating the impact of greenhouse gases on the global climate. Oceanic processes affect phenomena such as hurricanes, extreme precipitation, and droughts. Ocean models rely on subgrid-scale parameterizations that require calibration and often significantly affect model skill. When model sensitivities to parameters can be computed by using approaches such as automatic differentiation, they can be used for such calibration toward reducing the misfit between model output and data. Because the SOMA model code is challenging to differentiate, we have created neural network-based surrogates for estimating the sensitivity of the ocean model to model parameters. We first generated perturbed parameter ensemble data for an idealized ocean model and trained three surrogate neural network models. The neural surrogates accurately predicted the one-step forward ocean dynamics, of which we then computed the parametric sensitivity.

CCS CONCEPTS

• **Applied computing** → **Environmental sciences**; • **Computing methodologies** → **Neural networks**; • **Mathematics of computing** → **Automatic differentiation**.

PASC '24, June 3–5, 2024, Zurich, Switzerland

ACM ISBN 979-8-4007-0639-4/24/06…$15.00
https://doi.org/10.1145/3659914.3659920

KEYWORDS

Ocean Modeling, Adjoints, Neural Opeartor

ACM Reference Format:
Yixuan Sun, Elizabeth Cucuzzella, Steven Brus, Sri Hari Krishna Narayanan, Balasubramanya Nadiga, Luke Van Roekel, Jan Hückelheim, Sandeep Madireddy, and Patrick Heimbach. 2024. Parametric Sensitivities of a Wind-driven Baroclinic Ocean using Neural Surrogates. In *Platform for Advanced Scientific Computing Conference (PASC '24), June 3–5, 2024, Zurich, Switzerland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3659914.3659920

1 INTRODUCTION

The oceans are important in mitigating anthropogenic climate change, primarily by absorbing significant amounts of carbon dioxide and heat. Concurrently, oceanic processes redistributing mass, heat, and salt are key drivers in phenomena such as hurricanes, extreme precipitation, and droughts. Therefore, much effort is devoted to modeling and understanding the behavior of the ocean under various scenarios [7, 19, 41, 46]. Of particular interest are the long-term changes in critical ocean circulation patterns, which could have wide-ranging climate impacts. Understanding the stability of the circulation is critical to our ability to predict the conditions that could cause its weakening.

We consider the Model for Prediction Across Scales MPAS-Ocean, a numerical model designed for the simulation of the ocean system from time scales of days to millennia and spatial scales from sub-1 km to global circulations [12, 35, 39]. Being able to represent interactions across a wide range of scales, it can directly capture mesoscale ocean activity at sufficiently high resolutions or be used to study anthropogenic climate change when used as part of a comprehensive climate or Earth system model. In this context, subgrid-scale

parameterization, for example, of unresolved eddies or vertical mixing, plays an important role. We are interested in understanding the sensitivities of MPAS-Ocean's output to the model parameters. Estimating this sensitivity is time- and resource-consuming by running the model multiple times while varying each parameter in isolation. Other approaches include the use of Gaussian process, which suffers greatly from the curse of dimensionality [4, 16], and parameter study methods [2].

Alternatively, adjoint models have shown great promise in uncovering the sensitivity of the model to its parameters [8, 9, 28, 29]. They simultaneously calculate the derivatives of a single model output with respect to all model parameters. Adjoint models can be created manually or via automatic differentiation (AD), a technique to compute the adjoints of mathematical functions expressed as source code [13]. The reverse mode of AD, which computes adjoints, is also known as backpropagation and underpins machine learning [3]. Most popular machine learning frameworks such as PyTorch [33], which are used to build neural networks, provide the ability to perform the reverse mode of AD and thus compute adjoint sensitivity with little effort on the part of the end user. While AD tools exist for several programming languages, an AD tool can be challenging to apply for highly parallel, pre-existing codes, including MPAS-Ocean. The resulting derivative computation may exhibit poor performance and can be difficult to validate [18].

Neural networks (NNs), as universal function approximators [17], create models of physical processes from observational or simulation data and may act as surrogates for numerical models [30, 42, 43]. Once trained, NNs can infer outcomes at unseen parameter points compatible with the training data. Recently, adjoints obtained by differentiation of the surrogate NN model have been shown to match the accuracy of those obtained by conventional AD of the original forward simulation code in particular settings that were considered [6, 14]. We therefore explore how to generate an accurate NN surrogate for the idealized Simulating Ocean Mesoscale Activity (SOMA) test case within MPAS-Ocean [1]. We then use our NN model to generate adjoint versions of the original model to obtain parametric sensitivities.

The contributions of this work can be summarized as follows. (1) We generated a SOMA perturbed parameter ensemble dataset for deep learning model development and benchmarking; (2) we trained neural network surrogates with large-scale distributed training, aiming to recreate the timestepping behavior of the forward (true) model; and (3) we computed and verified neural adjoints from the neural surrogates and gained insight into the model sensitivity to four parameters. The rest of this paper is organized as follows. Section 2 presents the SOMA simulation, problem setting, and neural network surrogate generation. Section 3 presents the experiment setup, and Section 4 presents the results and discussion. Section 5 concludes the paper with a summary and a brief look at future work.

2 METHODS

This section describes the default SOMA configuration of MPAS-Ocean, problem formulation, and neural network surrogate.

2.1 SOMA Configuration

The SOMA configuration is designed to investigate equilibrium mesoscale eddy activity in a setting similar to how ocean climate models are deployed. SOMA simulates an idealized, eddying, midlatitude, double-gyre ocean basin with latitudes ranging from 21.58 to 48.58N and longitudes ranging from 16.58W to 16.58E [45]. The circular basin features curved coastlines with a 150 km wide, 100 m deep continental shelf and slope (see Figure 1). SOMA can be run at four different resolutions, where a coarser resolution is more granular: 4 km, 8 km, 16 km, and 32 km.

We have chosen to run SOMA at a resolution of 32 km. At this resolution, the mesoscale eddy field is unresolved and is parametrized instead. The diagnostic output is computed from five prognostic outputs (layer thickness, salinity, temperature, zonal velocity, and meridional velocity) that, in turn, are influenced by four model parameters. The parameterization of mesoscale tracer transport employed in SOMA is a combination of isopycnal diffusion known as Redi parameterization [38], with an associated parameter κ_{redi}, and eddy-induced advection known as Gent–McWilliams parameterization [10, 11], with an associated parameter κ_{GM}. The vertical mixing model uses the Richardson number–based parameterization [32], with an associated parameter κ_{bg}. Bottom drag parameterization, with an associated parameter C_D, is used to extract energy supplied by wind forcing. The prognostic variables whose temporal evolution we are interested in emulating are layer thickness, salinity, meridional velocity, zonal velocity, and temperature.

The original SOMA simulation runs for constant values of the scalar parameters that are being studied. For training our NN surrogates, we are interested in varying the parameters. Figure 2 shows the variation in ocean temperature for different values of the Gent–McWilliams parametrization. Significant variation in output for different parameter values is clearly observed. The SOMA setup is therefore suitable for creating a perturbed parameter ensemble, as described in Section 3.1.

2.2 Problem Setting

This section describes the problem setup. We use $\mathbf{x}$ to denote the state variables, p the physical model parameters, and θ the learnable parameters in the neural network.

On a high level, SOMA starts with the initial state, $\mathbf{x}_0$, and the model parameter(s), p, and solves for the state $\mathbf{x}(t)$ (i.e., the prognostic variables) at time t. The initial value problem, $d\mathbf{x}(t,p)/dt = f(\mathbf{x}(t))$, with $\mathbf{x}(0) = \mathbf{x}_0$, leads to the solution at time t, $\mathbf{x}_t = \mathbf{x}_0 + \int_0^t f(\mathbf{x}(\tau,p))d\tau$. The solution at the discrete time step $t+1$ can be expressed as

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \int_t^{t+1} f(\mathbf{x}(\tau,p))d\tau. \tag{1}$$

Building neural surrogates aims to model the one-step solving process in (1). We aim to use a neural network $\mathcal{N}_\theta$, parameterized by the neural network trainable parameters θ to approximate the solution operator. In particular, we trained the neural network to approximate the following mapping,

$$\mathbf{x}_{t+1} = \mathcal{N}_\theta(\mathbf{x}_t, p), \tag{2}$$

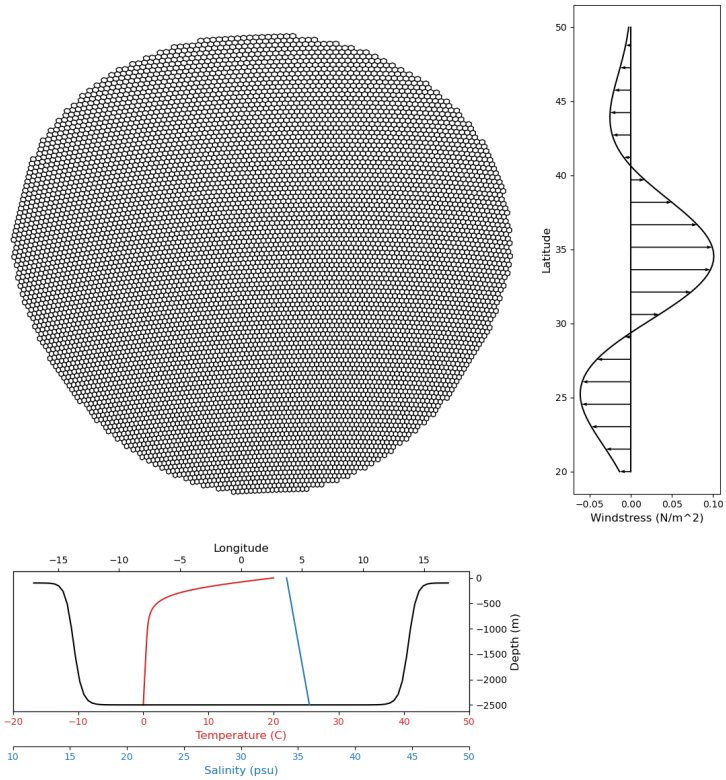


Figure 1: SOMA domain shown with the 32 km mesh. Below is the depth profile of the basin, along with the horizontally constant initial temperature and salinity profiles. To the right is the longitudinally constant (zonal-mean) imposed wind stress forcing.

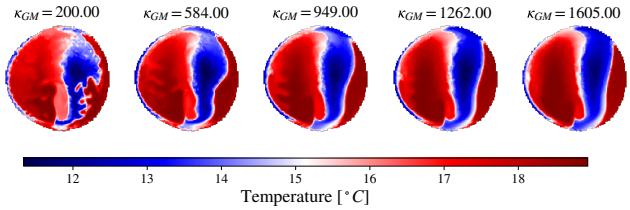


Figure 2: Variation shown in temperature with different values of Gent–McWilliams (GM) parameterization. The temperature at a depth of approximately 43 m at the end of simulations of the same initial condition is shown.

where $\mathbf{x}$ is the state variable vector, t represents the current time, and p is the parameter that impacts the trajectories of state variables. The dataset $\mathcal{D}$, obtained from each ensemble with varying p, described in Section 3.1, contains the input-output pairs $\{[\mathbf{x}_t^{(k)}, p^{(k)}], \mathbf{x}_{t+1}^{(k)}\}_{k=1}^{N}$. Now, the learning objective is to minimize the empirical loss function,

$$\mathcal{L}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{k=1}^{N} \|\mathbf{x}_{t+1}^{(k)} - \mathcal{N}_\theta(\mathbf{x}_t^{(k)}, p^{(k)})\|_2^2, \tag{3}$$

where N is the number of data points in the training set. We expect the trained neural network to accurately predict the state variables one step forward. Furthermore, with an accurate neural surrogate for the forward process, we are interested in understanding the adjoint sensitivity of the model with respect to the parameter p, $\partial \mathcal{N}_\theta / \partial p$, which can be easily calculated through backpropagation, once a skillful, trained neural network is available. Because the SOMA simulation is not readily differentiable using AD, we verify the calculation of neural adjoints by performing a dot-product test, described in Section 4.2.1, as the first step toward matching the model true adjoint $\partial \mathcal{M} / \partial p$, when a differentiable physical model $\mathcal{M}$ is available.

2.3 Neural Network Surrogates

We trained three types of NNs that are commonly used in learning dynamical systems [5, 31]—U-Net [40], ResNet [15], and FNO [24]—to learn the forward dynamics of the simulation. The values of normalized root mean square error (NRMSE) of the predicted prognostic variables from the three NNs trained for κ_{GM}, listed in Table 1, show that the Fourier neural operator (FNO) greatly outperforms (with lower NRMSEs) the other two NNs. We therefore primarily adopted FNO as the surrogate model to train on data from

Table 1: The normalized root mean square error (%) of the predicted prognostic variables from U-Net, ResNet, and FNO trained for κ_{GM}.

	Layer Thickness	Salinity	Temperature	Zonal Velocity	Meridional Velocity
U-Net	0.8961	0.6288	2.190	0.8679	0.5518
ResNet	4.273	3.978	3.889	1.624	1.914
FNO	0.2944	0.1706	0.2176	0.3195	0.2059

the dynamics of all varied parameters. FNO aims to learn the operator between infinite-dimensional function spaces and can handle high-dimensional data efficiently in the spectral space. FNO has had many successes in modeling physical systems [25, 34, 37]. In particular, FourCastNet [34], powered by FNO backbones, achieved outstanding performance in short- to mid-range global weather forecasting. The key component of FNO is the kernel integral operator implemented as a convolution in the Fourier domain. By compositing such kernel integral operators along with nonlinear lift and projection operations, FNO effectively approximates such an operator and can generalize beyond the data resolution in the training set. Figure 3 shows the data flow with FNO in our problem, where the input consists of 3D representations of the state variables and the external parameter, $\mathbf{x}$, p. The output is the same state variables at the next time step. We zero-padded the region outside the circular domain, mitigating the edge effect and avoiding artifacts from the periodic boundary assumption for the fast Fourier transform.

In the Fourier layer, the integral kernel performs convolution in the frequency domain, capturing global dependencies. In particular, high-frequency modes are filtered out to preserve the main structure of the solution, increasing the generalization ability. Because of the high-dimensional nature of our data (three-dimensional in space with multiple variables), we use a variant of FNO, called TFNO [23], which incorporates Tucker decomposition for more efficient learning. Tucker decomposition essentially does a principal component analysis for higher-order tensors [22], reducing the complexity of the trainable parameters in the FNO backbones, resulting in better training efficiency and generalization [23].

3 EXPERIMENTS

This section describes the data generation and postprocessing, neural network training, and evaluation metrics for the trained surrogate models.

3.1 Perturbed Parameter Ensemble Data Generation and Transformation

We derived from the literature a set of reasonable ranges for perturbing the parameters. Table 2 lists the parameters and maximum and minimum values for uniform sampling. We independently perturbed each parameter in the list and performed simulations to form one ensemble per parameter. For each ensemble, we trained a neural network surrogate. For each parameter, using uniformly sampled values in the range, we created 100 forward runs. In each run, we executed the simulation from the same initial condition with different parameter configurations for three years and saved the *daily* snapshots of the last year. At the 32 km resolution, there are 8,521 hexagonal cells on the grid, each with 60 vertical levels. Each month resulted in over 15 million (8521×60×30) cell values for each spatially and temporally varying output variable in the data set. Each run of the simulation was done with 128 cores via MPI. The problem setup and simulation code can be found on GitHub.

Table 2: Range of Perturbed Parameter Values.

SOMA Parameter	Symbol	Minimum	Maximum
GM_constant_kappa	κ_{GM}	200.0	2000.0
Redi_constant_kappa	κ_{redi}	0.0	3000.0
cvmix_background_diff	κ_{bg}	0.0	1e-4
implicit_bottom_drag	C_D	1e-4	1e-2

The generated data with its original fidelity and representation posed challenges for training FNO-based models. Therefore, the data generated for the mesh grid was converted to a standard latitude and longitude grid through spatial interpolation, and the values were populated to standard array entries. As a result, we obtained the data represented with regular grids and stored as arrays, each instance of shape (6, 60, 100, 100). The first dimension contains the five prognostic variables and one model parameter, and the last three dimensions represent the spatial axes of the domain. We post-processed the *first month* of saved data for neural surrogate training and evaluation purposes. In total, we have 3,000 data instances (30 time steps × 100 runs) for the ensemble per perturbed parameter.

3.2 Neural Surrogate Training

We implemented the data loaders, data transformations, and TFNOs in PyTorch [33]. The data loaders streamed the input-output pairs in batches, alleviating memory usage. Based on independent runs of SOMA, the data was randomly split into training, validation, and testing sets with a ratio of 0.6, 0.2, and 0.2. That is, the training set contained 60 forward runs, and 20 forward runs for both validation and testing sets. Our objective was to predict all five prognostic variables from a single model simultaneously. The state variables lie in dramatically different ranges. The range difference can cause the learning to be biased toward certain variables. Therefore, we normalized the data to ensure the resulting values had similar ranges. In particular, we used the minimum and maximum variable values from the simulation instances to perform the normalization with a target range of [0, 1]. With the high dimensionality of the data and complexity of the neural network, we distributed the training and data loader onto ten computing nodes equipped with 4 Nvidia A100 GPUs each. We trained all four models with a batch size of 10, L_2 norm as loss function, and Adam optimizer [21]

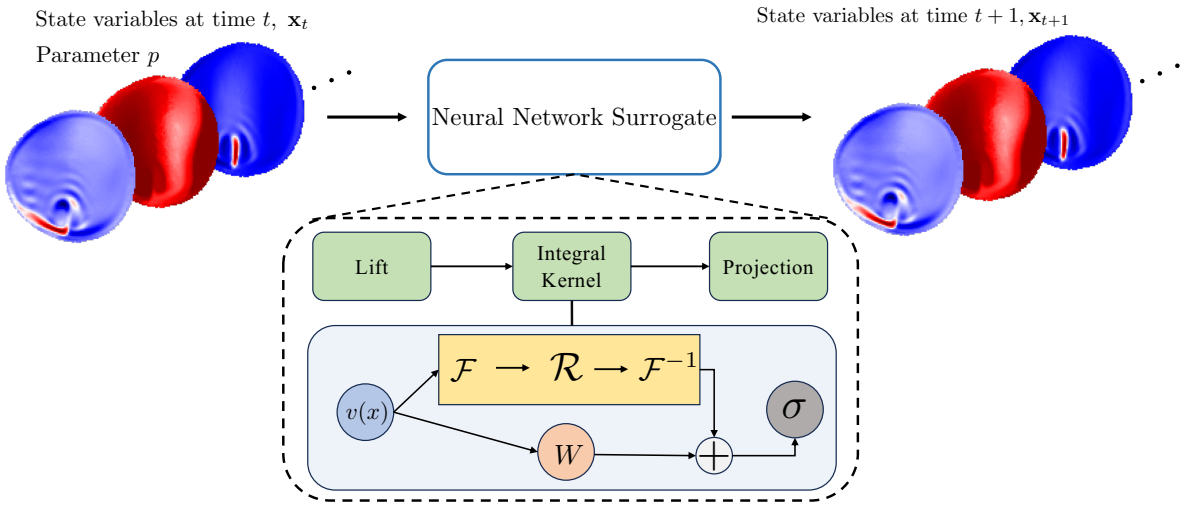


Figure 3: Fourier neural operator surrogate schematic for one-step forward forecasting of prognostic variables. The input state variable fields and parameters are in 3D representations. The lift block performs pointwise projection to raise the dimensions of the input. A Fourier layer performs fast Fourier transform (FFT) of the input and conducts convolution operations in the frequency domain. The high-frequency modes from the convolution are truncated. At the end of the Fourier layer, an inverse FFT transforms the data onto its original physical domain. Another layer of pointwise projection then aligns with the dimension of the solution function.

for over 2,000 epochs with observed convergence. Then, based on the validation loss, we selected the best model snapshot with the lowest validation loss for further evaluation with the testing set. The implementation and training scripts are available on GitHub.

3.3 Metrics

We use three metrics to evaluate the predictions of the model based on ground truth: coefficient of determination (R^2), NRMSE, and anomaly correlation coefficient (ACC). R^2 is defined as $R^2 = 1 - SS_{res}/SS_{tot}$, where $SS_{res} = \sum(y - \hat{y})^2$; $SS_{tot} = \sum(y - \bar{y})^2$; and y, $\hat{y}$, and $\bar{y}$ are true values, predicted value, and the average of true values, respectively. It describes the proportion of variance in the target that the model can explain. A number closer to 1 is associated with a better model. NRMSE is defined as $NRMSE = \frac{1}{N}\sqrt{\sum(y - \hat{y})^2}/(y_{max} - y_{min})$. The NRMSE accounts for the average percentage deviation of the predicted values from the true values. It is normalized by using the true data range to compare different models and predictions for different state variables fairly. The third metric, ACC, implies the similarity between two state variable fields, including the pattern-contrast variations. It is defined as $ACC = \sum(y-\bar{y})(\hat{y}-\bar{y})/\sqrt{\sum(y-\bar{y})^2(\hat{y}-\bar{y})^2}$ [44]. ACC ranges from -1 to 1, with 1 indicating a perfect positive correlation between the anomalies from predicted and true values. All three metrics are robust to state variable types and scales.

4 RESULTS AND DISCUSSION

This section discusses the accuracy of our trained NNs and the computation of adjoints.

4.1 Forward Prediction

Table 3 shows the performance metrics of the TFNOs for each output prognostic variable (layer thickness, salinity, temperature, zonal velocity, and meridional velocity) with varying parameters. The trained models, overall, accurately predict the prognostic variables one step in the future. The model performance on zonal and meridional velocities among the four trained networks is less accurate than other variables. In particular, the network taking varying κ_{bg} is the least performant in predicting zonal and meridional velocities. One possible reason is that zonal and meridional velocities have high contrast profiles in small regions, whereas other variables lack sharp changes in values. Since we trained the network to minimize the total loss for all variables, it was more challenging for the network to capture high-frequency features present in zonal and meridional velocity profiles.

Figure 4 shows the true, predicted, and absolute error for one-step forward values for the prognostic variables of the trained TFNO for the parameter, κ_{GM}. The predicted profiles follow the same range of variable values and closely resemble the actual distribution, with a slight loss of details. Moreover, we are interested in the model performance for more extended horizon rollout, where the trained networks predict prognostic variables in an autoregressive way. Figure 5 presents the four neural surrogates' rollout performance on the testing set. The NRMSE indicates accurate rollout from three of the four surrogates more than 10 steps (days), and the model taking varying κ_{redi} has the least accumulated error for temperature and meridional velocity profiles. The model taking κ_{bg} exhibits the fastest error accumulation compared with other models, holding up a relatively accurate rollout of only about 5 steps. This could be due to the trained neural surrogate not fully capturing

Table 3: Single-step forward prediction results (TFNO): Varying κ_{GM}, κ_{redi}, κ_{bg}, and C_D.

	κ_{GM}			κ_{redi}		
	R^2	NRMSE (%)	ACC	R^2	NRMSE (%)	ACC
Layer Thickness	0.9999	0.2944	0.9999	0.9987	0.6119	0.9993
Salinity	0.9999	0.1706	1.000	0.9993	0.4992	0.9997
Temperature	0.9999	0.2176	1.000	0.9921	0.1028	0.9960
Zonal Vel.	0.9611	0.3195	0.9806	0.9314	0.2007	0.9806
Meridional Vel.	0.9928	0.2059	0.9969	0.9346	0.1903	0.9969
	κ_{bg}			C_D		
	R^2	NRMSE (%)	ACC	R^2	NRMSE (%)	ACC
Layer Thickness	0.9997	0.3463	0.9998	0.9997	0.4050	0.9990
Salinity	0.9998	0.3439	0.9999	0.9999	0.2281	1.000

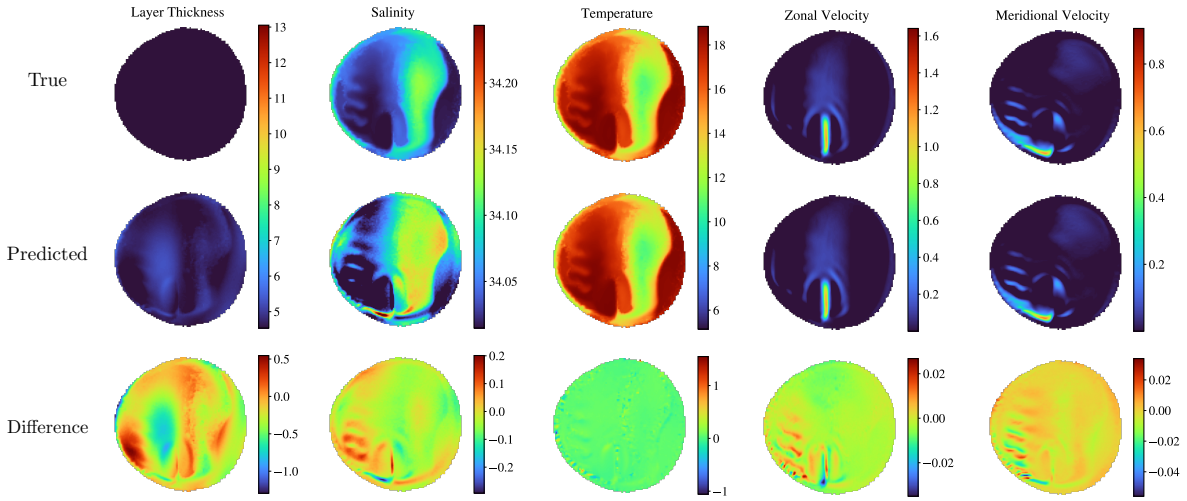
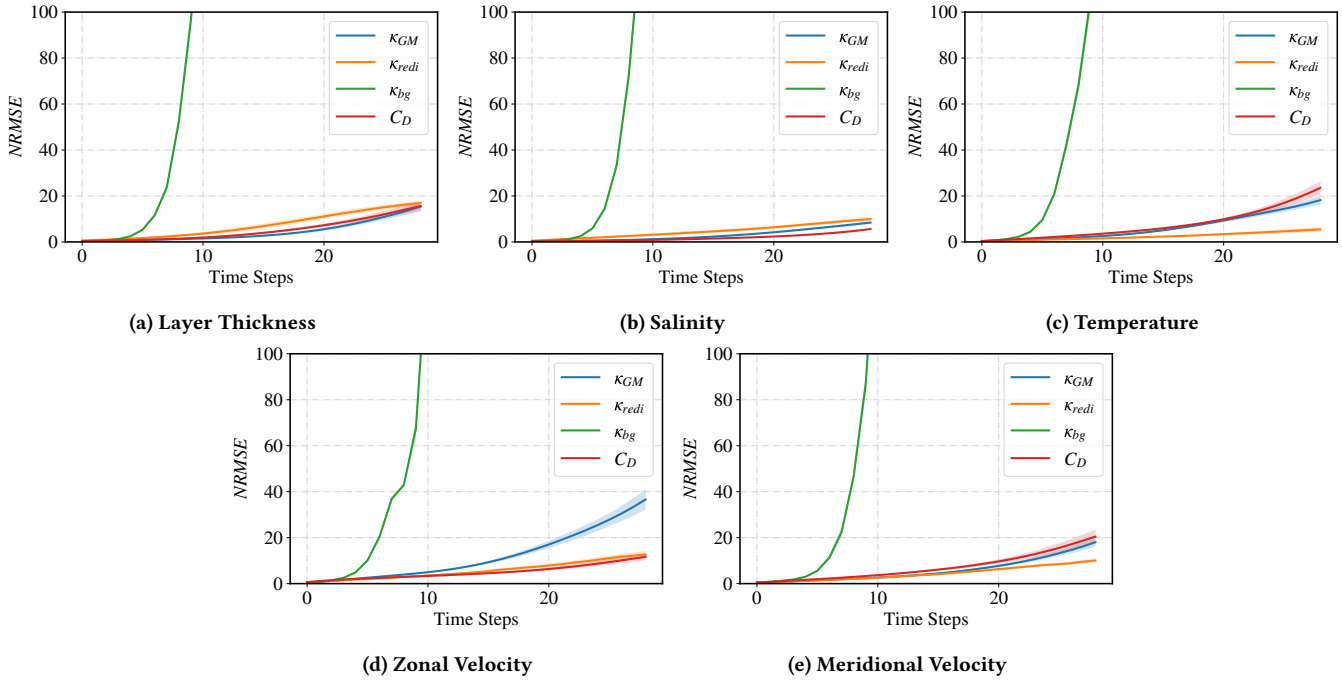


Figure 4: One-step forward predictions from the trained TFNO for all five prognostic variables at a depth of approximately 43 m and at timestep 15 (days). Units are (from left to right column): m, g/kg, °C, m/s, m/s.

the possibly more intricate and nonlinear interactions between the prognostic variables and κ_{bg}, resulting in a small error (comparable to other models) in single-step forward prediction but large error accumulation in the multistep rollout.

4.2 Neural Adjoints

We are focused on understanding how sensitive the prognostic variables are to the parameter using the neural surrogates. Using the accurate neural surrogate makes it possible to determine the adjoint sensitivity through backpropagation. We have computed the sensitivity of the neural surrogates to the parameters of the physical model by calculating the Jacobian of the surrogate's output w.r.t. the parameter. The output field is of shape $\mathbf{x}_{t+1} \in \mathbb{R}^{n\times 60\times 100\times 100}$, where n is the number of state variables (five in our case). The input physical model parameters, which do not vary spatially, have the shape of $p \in \mathbb{R}^m$, where $m = 1$ is the number of model parameters in our case. The Jacobian, $\frac{\partial \mathcal{N}_\theta}{\partial(\mathbf{x}_t, p)}$,of the the trained NNs, $\mathcal{N}_\theta : \mathbb{R}^{(n+m)\times 60\times 100\times 100} \mapsto \mathbb{R}^{n\times 60\times 100\times 100}$, is computationally expensive to compute given its size. However, to obtain the adjoint sensitivity to the parameter, $\frac{\partial \mathcal{N}_\theta}{\partial p}$, by differentiating the NNs involves differentiating w.r.t the state variables as well, because the parameters and state are stored in a single tensor. Therefore, we calculated the Jacobians at various randomly selected locations in the output domain and obtained the gradient with respect to p, per location, per state variable. It leads to calculating the gradient of a function, $\mathcal{N}^*_\theta : \mathbb{R}^{(n+m)\times 60\times 100\times 100} \mapsto \mathbb{R}^1$, which is straightforward and efficient using backpropagation. Then, we extracted the portion related to p. Because of its spatial uniformity, for a



(a) Layer Thickness (b) Salinity (c) Temperature

(d) Zonal Velocity (e) Meridional Velocity

Figure 5: Rollout performance, reported in NRMSE, of the trained surrogates for the prognostic variables. The neural surrogates for varying κ_{GM}, κ_{redi}, and C_D make accurate predictions in an autoregressive way up to 20 steps (days) until the error accumulation starts to rapidly increase. Meanwhile, the performance of neural surrogate for varying κ_{bg} quickly degrades in the first few steps.

selected location and prognostic variable, we had $J_{loc.} \in \mathbb{R}^{1\times m}$ as the measure of the sensitivity. These Jacobians were used to show the sensitivity of output state variables to the physical model parameters, summarized in Table 4. The values were the averaged gradient of the state variables to the parameters κ_{GM}, κ_{redi}, κ_{bg}, and C_D over 5 randomly selected horizontal locations with a fixed vertical level in the domain and were extended for over a month in the testing set. The results suggest, for the trained neural surrogate, that the outputs are least affected by κ_{GM} since its sensitivity values are an order of magnitude less than those of the other parameters. For the neural surrogates with varying κ_{GM} and κ_{redi}, we observe higher sensitivities for temperature, zonal, and meridional velocity values than for layer thickness and salinity. Zonal and meridional velocity sensitivities to κ_{bg} are the highest among the prognostic variables, while temperature takes the lowest value. For the neural surrogate taking C_D, the sensitivities of all prognostic variables are at a similar level.

4.2.1 Adjoint Dot-Product Tests. We aim to examine the correctness of the calculated neural adjoints. The Jacobian of the neural surrogate at a specific spatial location has the form $J_{loc} = \frac{\partial \mathcal{N}_\theta}{\partial h_N}\frac{\partial h_N}{\partial h_{N-1}}\cdots\frac{\partial h_0}{\partial p} \in \mathbb{R}^{n\times m}$, where hs are the outputs of the hidden layers in the NN. Surrogate adjoints are easily accessible by reverse-mode differentiation of trained neural networks. Given the complexity of the neural surrogate and data representations, we decided to implement a test that compares the neural adjoints with the Jacobian obtained by direct differentiation (forward differentiation). To this end, we formulated a dot-product test that allows us to efficiently verify the consistency between the two differentiation modes, shown in Equation 4. In particular, with random vectors $\mathbf{v} \in \mathbb{R}^m$ and $\mathbf{w} \in \mathbb{R}^n$, we tested whether the equality sign holds. Instead of explicitly calculating the Jacobian matrices, which can require a large amount of memory and be computationally prohibitive, we focused on computing the vector-Jacobian product (VJP), associated with adjoint calculation, and the Jacobian-vector product (JVP), associated with direct differentiation. We then generated 100 random vector $\mathbf{w}, \mathbf{v}$ pairs and compared the results from the left-hand side (LHS) and right-hand side (RHS) of Equation 4.

$$w^\top \cdot \underbrace{\left(\frac{\partial \mathcal{N}_\theta}{\partial p} v\right)}_{\text{JVP}} \overset{?}{=} \underbrace{\left(w^\top \frac{\partial \mathcal{N}_\theta}{\partial h_N}\frac{\partial h_N}{\partial h_{N-1}}\cdots\frac{\partial h_0}{\partial p}\right)}_{\text{VJP}} \cdot v, \tag{4}$$

Figure 6 shows the scatter plot of the LHS (direct differentiation) and RHS (adjoint calculation). The data points form a visually straight line, suggesting the calculated values are close or equal on both sides. Furthermore, the difference curve (in blue) and percentage error curve (in orange) show great alignment of the neural adjoints and direct differentiation, verifying the neural adjoint sensitivity based on the accurate surrogate.

Table 4: Average adjoint sensitivities computed for the five prognostic variables and the four parameters. The average was calculated across five randomly selected horizontal locations at a fixed vertical level (depth of approximately 43 m) within the domain and extended over a period of 30 days.

	Layer Thickness	Salinity	Temperature	Zonal Velocity	Meridional Velocity
$\partial\mathcal{N}/\partial\kappa_{GM}$	1.642e-09	2.353e-09	3.379e-09	2.615e-09	3.014e-09
$\partial\mathcal{N}/\partial\kappa_{redi}$	1.521e-08	2.319e-08	6.857e-08	9.578e-08	8.248e-08
$\partial\mathcal{N}/\partial\kappa_{bg}$	3.920e-08	5.233e-08	2.321e-08	6.345e-08	7.051e-08
$\partial\mathcal{N}/\partial C_D$	1.886e-08	2.367e-08	1.995e-08	1.848e-08	2.413e-08

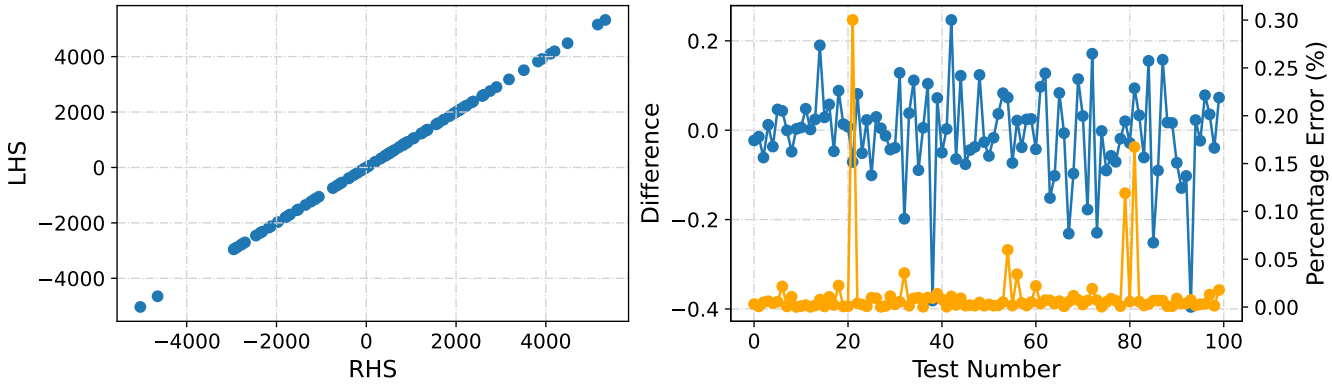


Figure 6: Dot-product test results of the trained neural surrogate. In Equation 4, the right-hand side calculates the neural adjoints via reverse-mode differentiation. The left-hand side uses forward differentiation to calculate the Jacobian-vector product. w and v are random vectors. The difference between the two sides and the corresponding percentage error suggest a good match between the neural adjoint and direct differentiation.

5 CONCLUSION

Based on perturbed parameters from SOMA runs, the neural surrogates we developed accurately predict the prognostic variables in one step forward compared with the original SOMA model. Although trained for one-step forward predictions, three of four neural surrogates can produce accurate rollouts of up to 10 timesteps. Additionally, we have successfully computed the neural adjoints from the trained models and obtained initial insights into the sensitivities. The results show that zonal and meridional velocity profiles are the most sensitive to κ_{GM}, κ_{redi}, and κ_{bg}. At the same time, the temperature is more sensitive to κ_{GM} and κ_{redi}.

Our future work includes improving adjoint-aware training by incorporating known physics and investigating the feasibility of applying our methodology to the MPAS-O code, specifically in a configuration simulating the AMOC. The governing equations of MPAS-O are well established. Utilizing known physics in the training of neural surrogates helps improve accuracy, reduce the requirement of large data size, and regularize learning for better generalization [20, 26, 27, 36].

ACKNOWLEDGMENTS

We gratefully acknowledge the computing resources provided on Bebop, a high-performance computing cluster operated by LCRC at Argonne National Laboratory. This research used resources from the NERSC, a U.S. Department of Energy Office of Science User Facility located at LBNL. Material based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research and Office of BER, Scientific Discovery through Advanced Computing (SciDAC) program, under Contract DE-AC02-06CH11357. We are grateful to the Sustainable Horizons Institute's Sustainable Research Pathways workforce development program.

REFERENCES

[1] [n. d.]. ocean/soma test group. https://mpas-dev.github.io/compass/latest/users_guide/ocean/test_groups/soma.html. Accessed: 2023-11-29.
[2] Brian M. Adams, William J. Bohnhoff, Keith R. Dalbey, Mohamed S. Ebeida, John P. Eddy, Michael S. Eldred, Russell W. Hooper, Patricia D. Hough, Kenneth T. Hu, John D. Jakeman, Mohammad Khalil, Kathryn A. Maupin, Jason A. Monschke, Elliott M. Ridgway, Ahmad A. Rushdi, Daniel Thomas Seidl, John Adam Stephens, and Justin G. Winokur. 2021. Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.15 User's Manual. (11 2021). https://doi.org/10.2172/1829573
[3] Atılım Günes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. 2017. Automatic Differentiation in Machine Learning: A Survey. *J. Mach. Learn. Res.* 18, 1 (Jan. 2017), 5595–5637.
[4] Mickael Binois and Nathan Wycoff. 2022. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization* 2, 2 (2022), 1–26.
[5] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. 2023. Spherical Fourier Neural Operators: Learning Stable Dynamics on the Sphere. *arXiv preprint arXiv:2306.03838* (2023).

[6] Austin Chennault, Andrey A. Popov, Amit N. Subrahmanya, Rachel Cooper, Anuj Karpatne, and Adrian Sandu. 2021. Adjoint-Matching Neural Network Surrogates for Fast 4D-Var Data Assimilation. *CoRR* abs/2111.08626 (2021). https://arxiv.org/abs/2111.08626

[7] Brad deYoung, Mike Heath, Francisco Werner, Fei Chai, Bernard Megrey, and Patrick Monfray. 2004. Challenges of Modeling Ocean Basin Ecosystems. *Science* 304, 5676 (2004), 1463–1466. https://doi.org/10.1126/science.1094858

[8] Ronald M Errico and Tomislava Vukicevic. 1992. Sensitivity analysis using an adjoint of the PSU-NCAR mesoseale model. *Monthly Weather Review* 120, 8 (1992), 1644–1660. https://doi.org/10.1175/1520-0493(1992)120<1644:SAUAAO>2.0.CO;2

[9] David Ferreira, John Marshall, and Patrick Heimbach. 2005. Estimating Eddy Stresses by Fitting Dynamics to Observations Using a Residual-Mean Ocean Circulation Model and Its Adjoint. *Journal of Physical Oceanography* 35, 10 (2005), 1891 – 1910. https://doi.org/10.1175/JPO2785.1

[10] Peter R. Gent and James C. Mcwilliams. 1990. Isopycnal Mixing in Ocean Circulation Models. *Journal of Physical Oceanography* 20, 1 (1990), 150–155. https://doi.org/10.1175/1520-0485(1990)020<0150:IMIOCM>2.0.CO;2

[11] Peter R. Gent, Jurgen Willebrand, Trevor J. McDougall, and James C. McWilliams. 1995. Parameterizing Eddy-Induced Tracer Transports in Ocean Circulation Models. *Journal of Physical Oceanography* 25, 4 (1995), 463 – 474. https://doi.org/10.1175/1520-0485(1995)025<0463:PEITTI>2.0.CO;2

[12] Jean-Christophe Golaz, Peter M. Caldwell, Luke P. Van Roekel, Mark R. Petersen, Qi Tang, Jonathan D. Wolfe, Guta Abeshu, Valentine Anantharaj, Xylar S. Asay-Davis, David C. Bader, Sterling A. Baldwin, Gautam Bisht, Peter A. Bogenschutz, Marcia Branstetter, Michael A. Brunke, Steven R. Brus, Susannah M. Burrows, Philip J. Cameron-Smith, Aaron S. Donahue, Michael Deakin, Richard C. Easter, Katherine J. Evans, Yan Feng, Mark Flanner, James G. Foucar, Jeremy G. Fyke, Brian M. Griffin, Cécile Hannay, Bryce E. Harrop, Mattthew J. Hoffman, Elizabeth C. Hunke, Robert L. Jacob, Douglas W. Jacobsen, Nicole Jeffery, Philip W. Jones, Noel D. Keen, Stephen A. Klein, Vincent E. Larson, L. Ruby Leung, Hong-Yi Li, Wuyin Lin, William H. Lipscomb, Po-Lun Ma, Salil Mahajan, Mathew E. Maltrud, Azamat Mametjanov, Julie L. McClean, Renata B. McCoy, Richard B. Neale, Stephen F. Price, Yun Qian, Philip J. Rasch, J. E. Jack Reeves Eyre, William J. Riley, Todd D. Ringler, Andrew F. Roberts, Erika L. Roesler, Andrew G. Salinger, Zeshawn Shaheen, Xiaoying Shi, Balwinder Singh, Jinyun Tang, Mark A. Taylor, Peter E. Thornton, Adrian K. Turner, Milena Veneziani, Hui Wan, Hailong Wang, Shanlin Wang, Dean N. Williams, Phillip J. Wolfram, Patrick H. Worley, Shaocheng Xie, Yang Yang, Jin-Ho Yoon, Mark D. Zelinka, Charles S. Zender, Xubin Zeng, Chengzhu Zhang, Kai Zhang, Yuying Zhang, Xue Zheng, Tian Zhou, and Qing Zhu. 2019. The DOE E3SM Coupled Model Version 1: Overview and Evaluation at Standard Resolution. *Journal of Advances in Modeling Earth Systems* 11, 7 (2019), 2089–2129. https://doi.org/10.1029/2018MS001603

[13] Andreas Griewank and Andrea Walther. 2008. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* (2nd ed.). Number 105 in Other Titles in Applied Mathematics. SIAM, Philadelphia, PA. http://bookstore.siam.org/ot105/

[14] Sam Hatfield, Matthew Chantry, Peter Dueben, Philippe Lopez, Alan Geer, and Tim Palmer. 2021. Building Tangent-Linear and Adjoint Models for Data Assimilation With Neural Networks. *Journal of Advances in Modeling Earth Systems* 13, 9 (2021), e2021MS002521. https://doi.org/10.1029/2021MS002521

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Takeyuki Hida and Masuyuki Hitsuda. 1993. *Gaussian processes*. Vol. 120. American Mathematical Soc.

[17] Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* 4, 2 (1991), 251–257.

[18] Jan Hückelheim, Harshitha Menon, William Moses, Bruce Christianson, Paul Hovland, and Laurent Hascoët. 2023. Understanding Automatic Differentiation Pitfalls. arXiv:2305.07546 [math.NA]

[19] IPCC. 2021. *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Vol. In Press. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA. https://doi.org/10.1017/9781009157896

[20] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3, 6 (2021), 422–440. https://doi.org/10.1038/s42254-021-00314-5

[21] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[22] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM Rev.* 51, 3 (2009), 455–500.

[23] Jean Kossaifi, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. 2023. Multi-Grid Tensorized Fourier Neural Operator for High Resolution PDEs. https://openreview.net/forum?id=po-oqRst4Xm

[24] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2021. Fourier Neural Operator for Parametric Partial Differential Equations. arXiv:2010.08895 [cs.LG]

[25] Zhijie Li, Wenhui Peng, Zelong Yuan, and Jianchun Wang. 2022. Fourier neural operator approach to large eddy simulation of three-dimensional turbulence. *Theoretical and Applied Mechanics Letters* 12, 6 (2022), 100389.

[26] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. [n. d.]. Physics-Informed Neural Operator for Learning Partial Differential Equations. https://doi.org/10.48550/arXiv.2111.03794

[27] Xin-Yang Liu, Hao Sun, Min Zhu, Lu Lu, and Jian-Xun Wang. 2022. Predicting parametric spatiotemporal dynamics by multi-resolution PDE structure-preserved deep learning. arXiv:2205.03990 [cs.LG]

[28] Guokun Lyu, Armin Köhl, Ion Matei, and Detlef Stammer. 2018. Adjoint-Based Climate Model Tuning: Application to the Planet Simulator. *Journal of Advances in Modeling Earth Systems* 10, 1 (2018), 207–222. https://doi.org/10.1002/2017MS001194 arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2017MS001194

[29] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid Control Using the Adjoint Method. *ACM Trans. Graph.* 23, 3 (aug 2004), 449—-456. https://doi.org/10.1145/1015706.1015744

[30] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K. Gupta, and Aditya Grover. [n. d.]. ClimaX: A foundation model for weather and climate. arXiv:2301.10343 [cs] http://arxiv.org/abs/2301.10343 version: 1.

[31] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. 2023. ClimaX: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343* (2023).

[32] R. C. Pacanowski and S. G. H. Philander. 1981. Parameterization of Vertical Mixing in Numerical Models of Tropical Oceans. *Journal of Physical Oceanography* 11, 11 (1981), 1443–1451. https://doi.org/10.1175/1520-0485(1981)011<1443:POVMIN>2.0.CO;2

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* 32 (2019).

[34] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. 2022. FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214* (2022).

[35] Mark R. Petersen, Xylar S. Asay-Davis, Anne S. Berres, Qingshan Chen, Nils Feige, Matthew J. Hoffman, Douglas W. Jacobsen, Philip W. Jones, Mathew E. Maltrud, Stephen F. Price, Todd D. Ringler, Gregory J. Streletz, Adrian K. Turner, Luke P. Van Roekel, Milena Veneziani, Jonathan D. Wolfe, Phillip J. Wolfram, and Jonathan L. Woodring. 2019. An Evaluation of the Ocean and Sea Ice Climate of E3SM using MPAS and Interannual CORE-II Forcing. *Journal of Advances in Modeling Earth Systems* 11, 5 (2019), 1438–1458. https://doi.org/10.1029/2018MS001373 arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018MS001373

[36] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. [n. d.]. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. https://doi.org/10.48550/arXiv.1711.10561 arXiv:1711.10561 [cs, math, stat]

[37] Meer Mehran Rashid, Tanu Pittie, Souvik Chakraborty, and NM Anoop Krishnan. 2022. Learning the stress-strain fields in digital composites using Fourier neural operator. *Iscience* 25, 11 (2022).

[38] Martha H. Redi. 1982. Oceanic Isopycnal Mixing by Coordinate Rotation. *Journal of Physical Oceanography* 12, 10 (1982), 1154–1158. https://doi.org/10.1175/1520-0485(1982)012<1154:OIMBCR>2.0.CO;2

[39] Todd Ringler, Mark Petersen, Robert L. Higdon, Doug Jacobsen, Philip W. Jones, and Mathew Maltrud. 2013. A multi-resolution approach to global ocean modeling. *Ocean Modelling* 69 (2013), 211–232. https://doi.org/10.1016/j.ocemod.2013.04.010

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[41] Albert J. Semtner. 1995. Modeling Ocean Circulation. *Science* 269, 5229 (1995), 1379–1385. https://doi.org/10.1126/science.269.5229.1379

[42] Yixuan Sun, Christian Moya, Guang Lin, and Meng Yue. 2023. DeepGraphONet: A deep graph operator network to learn and zero-shot transfer the dynamic response of networked systems. *IEEE Systems Journal* (2023).

[43] Jeyan Thiyagalingam, Mallikarjun Shankar, Geoffrey Fox, and Tony Hey. 2022. Scientific machine learning benchmarks. *Nature Reviews Physics* 4, 6 (2022), 413–420.

[44] Ralph RB von Frese, Michael B Jones, Jeong Woo Kim, and Jeong-Hee Kim. 1997. Analysis of anomaly correlations. *Geophysics* 62, 1 (1997), 342–351.

[45] Phillip J. Wolfram, Todd D. Ringler, Mathew E. Maltrud, Douglas W. Jacobsen, and Mark R. Petersen. 2015. Diagnosing Isopycnal Diffusivity in an Eddying, Idealized Midlatitude Ocean Basin via Lagrangian, in Situ, Global, High-Performance Particle Tracking (LIGHT). *Journal of Physical Oceanography* 45, 8 (2015), 2114–2133. https://doi.org/10.1175/JPO-D-14-0260.1

[46] Xiaoqin Yan, Rong Zhang, and Thomas R. Knutson. 2018. Underestimated AMOC Variability and Implications for AMV and Predictability in CMIP Models. *Geophysical Research Letters* 45, 9 (2018), 4319–4328. https://doi.org/10.1029/2018GL077378