
Ensembles of Neural Surrogates for Parametric Sensitivity in Ocean Modeling

Yixuan Sun

Argonne National Laboratory
yixuan.sun@anl.gov

Romain Egele

Oak Ridge National Laboratory
regele@ornl.gov

Sri Hari Krishna Narayanan

Argonne National Laboratory
snarayan@anl.gov

Luke Van Roekel

Los Alamos National Laboratory
lvanroekel@lanl.gov

Carmelo Gonzales

NVIDIA
carmelog@nvidia.com

Steven Brus

Argonne National Laboratory
sbrus@anl.gov

Balu Nadiga

Los Alamos National Laboratory
balu@lanl.gov

Sandeep Madireddy

Argonne National Laboratory
smadireddy@anl.gov

Prasanna Balaprakash

Oak Ridge National Laboratory
pbalapra@ornl.gov

Abstract

Accurate simulations of the oceans are crucial in understanding the Earth system. Despite their efficiency, simulations at lower resolutions must rely on various uncertain parameterizations to account for unresolved processes. However, model sensitivity to parameterizations is difficult to quantify, making it challenging to tune these parameterizations to reproduce observations. Deep learning surrogates have shown promise for efficient computation of the parametric sensitivities in the form of partial derivatives, but their reliability is difficult to evaluate without ground truth derivatives. In this work, we leverage large-scale hyperparameter search and ensemble learning to improve both forward predictions, autoregressive rollout, and backward adjoint sensitivity estimation. Particularly, the ensemble method provides epistemic uncertainty of function value predictions and their derivatives, providing improved reliability of the neural surrogates in decision making.

1 Introduction

The ocean is a vast reservoir of heat and plays a significant role in redistributing heat from the equatorial regions to the poles. Accurate simulation of this heat transport requires resolution of smaller scale ocean eddies [1]. While various ocean models have been constructed to model these complex dynamics [2–4], at lower resolution, these critical eddies must be modeled separately using a *parameterization*. Commonly, small scale eddies are modeled to remove baroclinic instability and also transport tracers along constant density layers [5, 6]. At smaller scales, vertical turbulent mixing is modeled as a down gradient process [7]. On the larger scale ocean, the effect of parameterizations is poorly understood, making it challenging to optimize uncertain model parameters to better match observations [8, 9]. Therefore, it is crucial to understand the model parametric sensitivity to effectively perform tuning to minimize model bias relative to observations. However, the existing physics-based ocean model codes are often too computationally expensive and not readily differentiable to

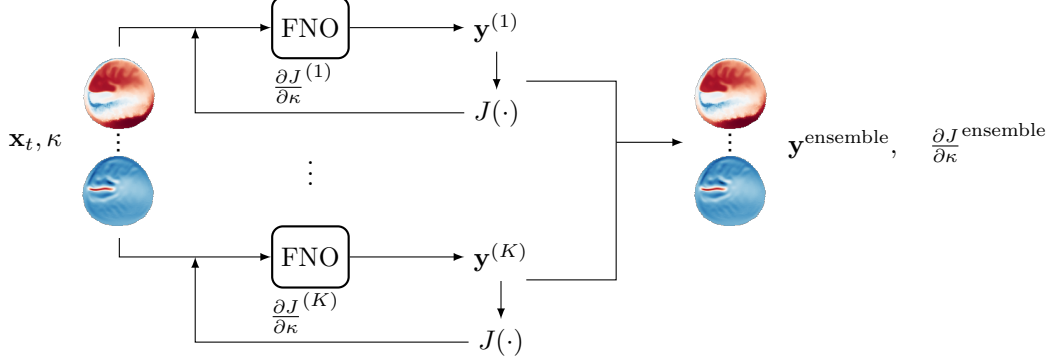


Figure 1: Overview of deep ensemble of Fourier Neural Operators (FNOs) for ocean parametric sensitivity. We train individual models with diverse hyperparameters to produce time-stepping predictions and parametric sensitivities. We aggregate these outputs to produce ensemble predictions of future ocean states and estimates of the partial derivative of objective J w.r.t. the parameterization κ .

support perturbation analysis or automatic differentiation [10]. Alternatively, neural surrogates have emerged [11–14], which can approximate the parametric sensitivity inexpensively by differentiating the trained networks [15, 16]. However, without ground truth derivatives to constrain the training, such as in [17], the estimated derivatives can deviate substantially from the true values, even when the network accurately reproduces the forward process. We propose to alleviate this issue through an ensemble of neural surrogates constructed from a large-scale hyperparameter optimization (HPO), as illustrated in Figure 1. The ensemble improves both the forward prediction performance and derivative estimates, which enables more stable autoregressive rollouts of the ocean states for long-term forecasts. Moreover, the neural surrogate ensemble provides quantified epistemic uncertainty, providing detailed reliability information for decision-making.

2 Neural Surrogate and Deep Ensemble

The target neural network surrogate model, $M(\cdot; \theta)$, approximates the true physical model, $\mathcal{M}(\mathbf{x}, \kappa)$. Here $\mathbf{x}$ and κ are the current ocean states and physical parameter of interest, and θ is set of the neural network parameters (including trainable parameters and architecture choices). We also aim to estimate the parametric sensitivity of a model-dependent objective function J by computing its derivative w.r.t. the parameterization, $\frac{dJ}{d\kappa} = \nabla_{\mathcal{M}} J \frac{\partial \mathcal{M}}{\partial \kappa}$ ¹. Specifically, for given training data set $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \kappa_i), \mathbf{y}_i\}_{i=1}^N$, generated from the true physical process $\mathbf{y}_i = \mathcal{M}(\mathbf{x}_i, \kappa_i)$, where $\mathbf{y}$ represents the ocean states at a future time, and N is the number of input-output pairs. We train the neural network, $\mathcal{N}_{\theta}$, such that $M(\mathbf{x}_i, \kappa_i; \theta) = \mathbf{y}_i$ by minimizing the loss function $\mathcal{L}(\theta; \mathcal{D}_{\text{train}}) = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}^{(i)} - \mathcal{N}_{\theta}(\mathbf{x}^{(i)}, \kappa^{(i)})\|_2^2$, and use $\frac{\partial M}{\partial \kappa}$ as the estimate of the parametric sensitivity.

We select the ensemble members from the results of a large-scale HPO, where the mapping between the hyperparameters and model performance is treated as a black-box function, $F = h(\lambda)$, $\lambda \in \Lambda$. F is the objective function, and Λ denotes the hyperparameter search space of which a point defines the data preprocessing, neural architectures, and training strategies. We simply use the validation loss as the search objective to be minimized, $F^* = \min_{\theta} \mathcal{L}(\theta; \mathcal{D}_{\text{val}})$. During the search, we model the relationship using a tree-based surrogate, $\hat{h}$, from the current known hyperparameter set and objective pairs, (λ_i, F_i) , s.t. $F_i = \hat{h}(\lambda_i)$, and perform centralized Bayesian optimization to select the next best evaluation point that can potentially lead to the global optimum [18]. Once the HPO has completed, we then select K top-performing models as the ensemble members.

Let $M_k(\mathbf{x}, \kappa; \theta_k)$ denote the k th member of the ensemble parameterized by θ_k . The ensemble prediction of the ocean states are $\mathbf{y}_{\text{ens}} = \sum_{k=1}^K w_k M_k(\mathbf{x}, \kappa, \theta_k)$, where w_k is the weight associated with the k th member of the ensemble. Then the model uncertainty is expressed as the empirical distribution variance, $\sigma^2 = \sum_{k=1}^K w_k (\mathbf{y}_k - \mathbf{y}_{\text{ens}})^2$.

¹The total derivative equals the partial derivative with respect to κ in this case, as the current state $\mathbf{x}$ is assumed to be independent of κ .

Table 1: RMSE ($\downarrow$) of single step forward predictions.

	Constant Pred.	Baseline	Top-1	Top10 Ensemble	Top10 Ensemble (weighted)
Layer Thickness	0.0004	0.0011	0.0008	0.0004	0.0004
Zonal V.	0.0122	0.0022	0.0024	0.0016	0.0016
Meridional V.	0.0075	0.0024	0.0019	0.0013	0.0013
Temp.	0.3876	0.0494	0.0484	0.0385	0.0392
Salinity	0.0072	0.0026	0.0024	0.0018	0.0017

Regarding the parametric sensitivity, we use $J_k = \frac{1}{2} \|\mathbf{y}_k\|_2^2$ as the scalar-valued objective² and compute the individual model sensitivity estimate as $\frac{\partial J_k}{\partial \kappa} = \frac{\partial J}{\partial \mathbf{y}_k} \frac{\partial \mathbf{y}_k}{\partial \kappa} = \mathbf{y}_k \frac{\partial \mathbf{y}_k}{\partial \kappa}$. Similar to the ensemble prediction, we aggregate the individual model sensitivity estimates to obtain the ensemble sensitivity estimate as $\frac{\partial J^{\text{ensemble}}}{\partial \kappa} = \sum_{k=1}^K w_k \mathbf{y}_k \frac{\partial \mathbf{y}_k}{\partial \kappa}$, and the uncertainty of the ensemble sensitivity estimate as $\sigma_{\partial J}^2 = \sum_{k=1}^K w_k \left(\frac{\partial J_k}{\partial \kappa} - \frac{\partial J^{\text{ensemble}}}{\partial \kappa} \right)^2$.

3 Numerical Experiments

We adopt the Fourier Neural Operator (FNO) [19] as the surrogate and apply our framework to the Simulating Mesoscale Ocean Activity (SOMA) model [9]. The FNO is trained on five ocean state variables, salinity, temperature, layer thickness, and meridional and zonal velocities, using the current state and the bolus kappa parameter (κ represents its strength) [5] as input, and predicting the next-step state. Hyperparameter optimization is performed with DeepHyper [18], and the top $K = 10$ models form our ensemble. We compare two weighting schemes: uniform weights and regression-based weights learned on the validation set. Appendix B provides details on the SOMA setup, data generation, preprocessing, normalization, and training. Evaluation is conducted on three tasks: single-step prediction, autoregressive rollout, and parametric sensitivity estimation.

Single-step prediction We report the model prediction RMSE over the testing set in Table 1. To highlight the effect of learning, the results also include the constant predictor, which is the temporal mean of ocean states. Aside from the layer thickness, all trained models present improved predictions over the constant predictor. In particular, the optimal model is superior to the baseline in predicting four out of the five states. The performance improvement continues for the constructed ensembles, which achieve the lowest RMSE scores. These results showcase the proposed framework’s high accuracy in modeling the forward process of the physical model over all single surrogates.

Rollout We investigate the model’s ability for producing longer-term forecasts by recursively making predictions of the ocean state variables at the next time step using the previous prediction as input. In the ideal situation, when the model prediction for single-step forward matches the truth state exactly, with the fixed time resolution, we expect to see infinite long, accurate rollout. However, due to the accumulation of prediction errors, the duration of producing accurate and stable rollout shortens. To evaluate the rollout performance, we compute the spatially averaged ocean state time series from the true and predicted states and compare the discrepancies between the two. Figure 2 shows the rollout performance comparison among models. The optimal and ensemble models show a significant rollout performance improvement over the baseline despite the slight improvement in single time-stepping prediction task. The long-term autoregressive prediction amplifies the subtle difference in error accumulation.

Parametric sensitivity Our final objective is to leverage the trained models to compute parametric sensitivity in the form of derivatives. As described in Section 2, we use the simple sum-of-square of the ocean states as the scalar-valued function for which we study the sensitivity. We compute and visualize the parametric sensitivity for temperature fields at $\kappa = 1429.81$. Figure 3 shows the comparison of estimated sensitivity using the trained models on the sea surface (depth 0). The models show substantial differences in magnitudes and the value distribution in their estimate for the parametric sensitivity. The baseline and ensemble models show similar sensitivity ranges

²The specific choice of the objective is for a proof of concept, which doesn’t necessarily correspond to a meaningful physical quantity. A more realistic objective function can be used based on the specific application.

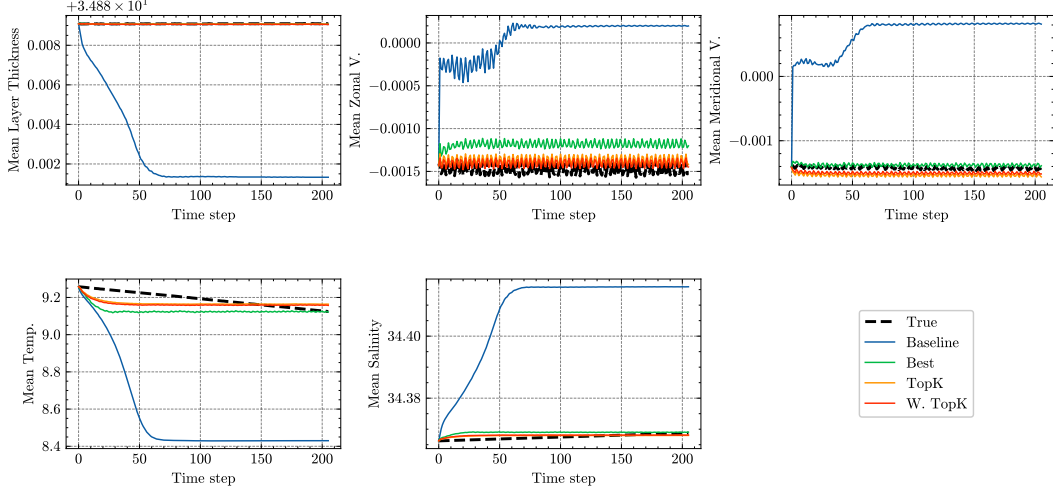


Figure 2: Autoregressive rollout performance comparisons among the three models. The curves are the spatially averaged ocean states for 8 years. For the baseline model, the forecasts quickly diverge from the true state due to error accumulation, whereas the ensembles generate stable prediction for all ocean states.

and present value concentration at certain areas while the best model from the HPO displays a wider range and more uniform distribution of values. Meanwhile, the ensemble models offer quantified uncertainties for the sensitivity estimate, suggesting evident variations among the ensemble members. This observation validates that a well-trained neural network for the forward model does not necessarily provide accurate derivative estimates. As the ground truth sensitivity is not available, we instead indirectly validate the estimated sensitivity using the linearized models to make predictions for nearby points. We expect the model having a better sensitivity estimation to produce a more accurate linear model. We linearize the trained model around each $\kappa_{in} \in \mathcal{D}_{test}$, to obtain $M^{linearized}(\mathbf{x}, \kappa) = M(\mathbf{x}, \kappa_{in}) + \frac{\partial M}{\partial \kappa_{in}}(\kappa - \kappa_{in})$. We then use $M^{linearized}(\mathbf{x}, \kappa)$ to make predictions of ocean state variables under a κ value in the testing set such that $\kappa = \min\{\kappa' \in \mathcal{D}_{test} | \kappa_{in} < \kappa'\}$. Figure 4 shows the testing RMSE of the linearized models on the temperature fields, where we also report the full model performance for reference. The baseline and uniformly weighted ensemble result in the lower RMSE, while the linearized best model presents the highest errors, despite its improved performance in the forward prediction (as the full model) over the baseline. By evaluating both forward-prediction accuracy and sensitivity estimates, our ensembles achieve superior performance while delivering quantified predictive uncertainty.

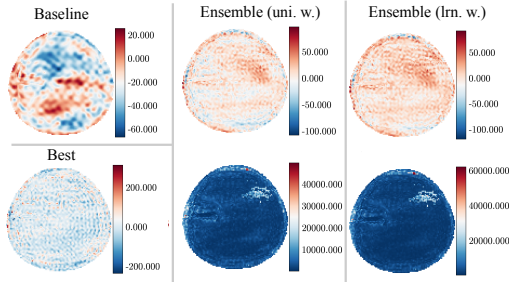


Figure 3: Time averaged model estimated sensitivity of J calculated using the temperature fields to κ .

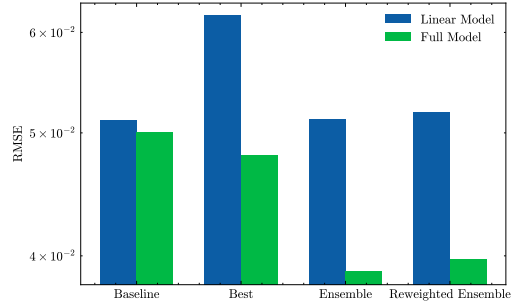


Figure 4: RMSE of the temperature predictions from linearized models.

4 Conclusion

This paper presents a deep ensemble approach for improved parametric sensitivity estimates for ocean models. Utilizing a large-scale HPO, we select top performing models and use two weighting schemes to construct the ensembles. The trained models are evaluated in single time-stepping prediction, long-range autoregressive rollout, and parametric sensitivity estimation. Without access to ground

truth sensitivity, we evaluate the linearized models to assess the quality of sensitivity estimates. The results show that the proposed ensemble models outperform the baseline model and best model from the HPO in all three tasks and provide quantified uncertainty at the same time. Future work involves extending the framework to multiple parameterizations and sensitivity evaluation through numerical differentiation of the physical model or data assimilation.

Acknowledgments and Disclosure of Funding

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] Stephen M. Griffies, Michael Winton, Whit G. Anderson, Rusty Benson, Thomas L. Delworth, Carolina O. Dufour, John P. Dunne, Paul Goddard, Adele K. Morrison, Anthony Rosati, Andrew T. Wittenberg, Jianjun Yin, and Rong Zhang. Impacts on ocean heat from transient mesoscale eddies in a hierarchy of climate models. *Journal of Climate*, 28(3):952 – 977, 2015. doi: 10.1175/JCLI-D-14-00353.1. URL <https://journals.ametsoc.org/view/journals/clim/28/3/jcli-d-14-00353.1.xml>.
- [2] Kristin E Hoch, Mark R Petersen, Steven R Brus, Darren Engwirda, Andrew F Roberts, Kevin L Rosa, and Phillip J Wolfram. Mpas-ocean simulation quality for variable-resolution north american coastal meshes. *Journal of Advances in Modeling Earth Systems*, 12(3):e2019MS001848, 2020.
- [3] Jean-Christophe Golaz, Luke P Van Roekel, Xue Zheng, Andrew F Roberts, Jonathan D Wolfe, Wuyin Lin, Andrew M Bradley, Qi Tang, Mathew E Maltrud, Ryan M Forsyth, et al. The doe e3sm model version 2: Overview of the physical model and initial model evaluation. *Journal of Advances in Modeling Earth Systems*, 14(12):e2022MS003156, 2022.
- [4] Shreyas Sunil Gaikwad, Sri Hari Krishna Narayanan, Laurent Hascoet, Jean-Michel Campin, Helen Pillar, An Nguyen, Jan Huckelheim, Paul Hovland, and Patrick Heimbach. Mitgcm-ad v2: Open source tangent linear and adjoint modeling framework for the oceans and atmosphere enabled by the automatic differentiation tool tapenade, 2024. URL <https://arxiv.org/abs/2401.11952>.
- [5] Peter R. Gent and James C. McWilliams. Isopycnal mixing in ocean circulation models. *Journal of Physical Oceanography*, 20(1):150 – 155, 1990. doi: 10.1175/1520-0485(1990)020<0150:IMIOCM>2.0.CO;2.
- [6] Martha H. Redi. Oceanic isopycnal mixing by coordinate rotation. *Journal of Physical Oceanography*, 12(10):1154–1158, 1982. doi: [https://doi.org/10.1175/1520-0485\(1982\)012<1154:OIMBCR>2.0.CO;2](https://doi.org/10.1175/1520-0485(1982)012<1154:OIMBCR>2.0.CO;2).
- [7] W. G. Large, J. C. McWilliams, and S. C. Doney. Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Reviews of Geophysics*, 32(4):363–403, 1994. doi: <https://doi.org/10.1029/94RG01872>.
- [8] Pavel Perezhogin, Arun Balakrishna, and Rahul Agrawal. Large eddy simulation of ocean mesoscale eddies, 2025. URL <https://arxiv.org/abs/2501.05357>.
- [9] Phillip J Wolfram, Todd D Ringler, Mathew E Maltrud, Douglas W Jacobsen, and Mark R Petersen. 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):2114–2133, 2015.

- [10] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, 2008. doi: 10.1137/1.9780898717761.
- [11] Taeyoon Kim and Woo-Dong Lee. Review on applications of machine learning in coastal and ocean engineering. *Journal of Ocean Engineering and Technology*, 36(3):194–210, 2022.
- [12] Marko Radeta, Agustin Zuniga, Naser Hossein Motlagh, M. Liyanage, Rúben Freitas, Maged A. Youssef, S. Tarkoma, Huber Flores, and P. Nurmi. Deep learning and the oceans. *Computer*, 55: 39–50, 2022. doi: 10.1109/mc.2022.3143087.
- [13] M. J. Er, Jie Chen, Yani Zhang, and Wenxiao Gao. Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. *Sensors (Basel, Switzerland)*, 23, 2023. doi: 10.3390/s23041990.
- [14] Ashesh Chattopadhyay, Michael Gray, Tianning Wu, Anna B Lowe, and Ruoying He. Oceannet: A principled neural operator-based digital twin for regional oceans. *Scientific Reports*, 14(1): 21181, 2024.
- [15] Yixuan Sun, Elizabeth Cucuzzella, Steven Brus, Sri Hari Krishna Narayanan, Balu Nadiga, Luke Van Roekel, Jan Hückelheim, and Sandeep Madireddy. Surrogate neural networks to estimate parametric sensitivity of ocean models. *arXiv preprint arXiv:2311.08421*, 2023.
- [16] Yixuan Sun, Elizabeth Cucuzzella, Steven Brus, Sri Hari Krishna Narayanan, Balasubramanya Nadiga, Luke Van Roekel, Jan Hückelheim, Sandeep Madireddy, and Patrick Heimbach. Parametric sensitivities of a wind-driven baroclinic ocean using neural surrogates. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–10, 2024.
- [17] Thomas O’Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-informed neural operator: An efficient framework for high-dimensional parametric derivative learning, 2023. URL <https://arxiv.org/abs/2206.10745>.
- [18] Prasanna Balaprakash, Romain Egele, Misha Salim, Romit Maulik, Venkat Vishwanath, Stefan Wild, et al. "deephpyer: A python package for scalable neural architecture and hyperparameter search", 2018. URL <https://github.com/deephpyer/deephpyer>.
- [19] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [20] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18, 2004.
- [21] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Timo Ewalds, Andrew El-Kadi, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. GenCast: Diffusion-based ensemble forecasting for medium-range weather, December 2023. URL <http://arxiv.org/abs/2312.15796>. arXiv:2312.15796 [physics].
- [22] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.
- [23] PhysicsNeMo Contributors. Nvidia physicsnemo: An open-source framework for physics-based deep learning in science and engineering. <https://github.com/NVIDIA/physicsnemo>, February 2023. If you use this software, please cite it as below.
- [24] Romain Egele, Isabelle Guyon, Yixuan Sun, and Prasanna Balaprakash. Is One Epoch All You Need For Multi-Fidelity Hyperparameter Optimization? In *ESANN 2023 proceedings*, pages 555–560, Bruges (Belgium) and online, 2023. Ciaco - i6doc.com. ISBN 978-2-87587-088-9. doi: 10.14428/esann/2023.ES2023-84. URL <https://www.esann.org/sites/default/files/proceedings/2023/ES2023-84.pdf>.

A Methodology details

We list the detailed steps of the proposed framework in Algorithm 1. Note that this framework is not limited to top-K selection with uniform weighting or re-weighting via linear regression. We plan to explore other model selection criteria, such as greedy selection [20], in future work.

Algorithm 1 Deep ensemble for ocean dynamics and parametric sensitivity

- 1: Given training and validation datasets, $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$
 - 2: Define baseline neural network, $M_{\theta_{\text{base}}}$.
 - 3: Select HPO surrogate S and acquisition function α .
 - 4: Select ensemble size K .
 - 5: Initialize HPO with $M_{\theta_{\text{base}}}$, $\theta' \leftarrow \theta_{\text{base}}$.
 - 6: **for** n in number of search steps **do**
 - 7: Evaluate candidate hyperparameters θ' using $\mathcal{D}_{\text{val}}$.
 - 8: Fit S with θ' and compute $\alpha(\theta')$.
 - 9: Return new model candidates $\theta' = \arg \max_{\theta \in \mathcal{X}} \alpha(\theta)$
 - 10: **end for**
 - 11: Rank models in HPO results and keep first K models.
 - 12: Fully train the selected models
 - 13: **if** Selection criterion is Top-K **then**
 - 14: Return model weights $w_1 = \dots = w_K$, $\sum w_k = 1$.
 - 15: **else if** Selection criterion is weighted **then**
 - 16: Perform linear regression
 - 17: Return updated weights, $w_1, \dots, w_K$.
 - 18: **end if**
 - 19: Produce ensemble predictions of $\mathbf{x}_{i+1}$ and $\frac{\partial J}{\partial \kappa}$.
-

B Experiment details

B.1 Data generation

Following [16], we perturb the Gent–McWilliams parameterization while keeping all other parameterizations at their nominal values to run independent simulations for model training and evaluation. Specifically, we sample values from a uniform distribution (range in Table 2) and create 100 forward runs. In each run, the simulation is initialized from the same condition and integrated for 23 years, with *monthly* snapshots saved. At the 32 km resolution, the grid consists of 8,521 hexagonal cells (a nearly circular horizontal domain), each with 60 vertical levels. Each simulation year produces over 13 million ($8521 \times 60 \times 26$) cell values for each spatially and temporally varying output variable in the dataset. We select five ocean states—*layer thickness*, *zonal velocity*, *meridional velocity*, *temperature*, and *salinity*—as targets and truncate the trajectories to retain the last 8 years, discarding the necessary spin-up stage. Each simulation run was performed using the publicly available code, which can be found at *Anonymous*.

Table 2: Range of Perturbed Parameter Values.

SOMA Parameter	Symbol	Minimum	Maximum
GM_constant_kappa	κ	200.0	2000.0

The generated data, in its original fidelity and representation, poses challenges for training FNO-based models because it is defined on an irregular grid. To address this, the mesh-grid data was converted to a standard latitude–longitude grid through spatial interpolation, and the values were mapped to regular array entries. As a result, we obtain data on regular grids stored as arrays, each instance having shape (6, 60, 100, 100). The first dimension contains the five ocean states and one model parameter κ , while the last three dimensions represent the spatial axes of the domain. We then convert the last 8 years of trajectories from the 100 independent simulations to these regular grid representations. Each simulation consists of 208 time steps covering the full trajectory. Finally, we split the trajectories into

training, validation, and testing sets and prepare input–output pairs (Section 2) using two consecutive time steps.

B.2 Data preprocessing

The raw data are three-dimensional in space, forming a nearly circular horizontal domain with multiple vertical layers indicating depth. Similar to most weather and climate modeling tasks [21, 22], we treat the vertical layers as additional feature dimensions. The data are therefore represented with shape (360, 100, 100), where the first axis corresponds to various ocean state values across depths. For example, the first 60 entries store layer thickness values from the sea surface (depth 0) to the 60th layer, while the last 60 represent the parameterization for this particular simulation across the vertical layers, which remain spatially constant in this work.

We transform the data for model training to improve stability and generalization. For each ocean state at each horizontal location (including the model output $\mathbf{y} = \mathbf{x}_{t+1}$), we use *per-depth* statistics to standardize the data as follows, for all t ,

$$\mathbf{x}_{i,d}^{\text{scale}} = \frac{\mathbf{x}_{i,d} - \mu_{i,d}}{\sigma_{i,d}},$$

where $\mathbf{x}_{i,d}$ denotes the i th ocean state at depth d , and $\mu_{i,d}$ and $\sigma_{i,d}$ are the corresponding sample mean and standard deviation obtained from the training set. At evaluation time, after the trained model produces forecasts, we apply inverse scaling to map the values back to the original physical space, for all t ,

$$\mathbf{x}_{i,d} = \mathbf{x}_{i,d}^{\text{scale}} \cdot \sigma_{i,d} + \mu_{i,d}.$$

For the variance of predictions from the ensemble model, the corresponding value in the original space can be obtained as

$$\text{Var}[\mathbf{x}_{i,d}] = \sigma_{i,d}^2 \cdot \text{Var}[\mathbf{x}_{i,d}^{\text{scale}}].$$

We leverage automatic differentiation to compute the parametric sensitivity. Since the trained networks operate on normalized data and $\frac{\partial J(\mathbf{x}_{t+1})}{\partial \kappa}$ is not computable from $\frac{\partial J(\mathbf{x}_{t+1}^{\text{scale}})}{\partial \kappa^{\text{scale}}}$ without the explicit access to $\frac{\partial \mathbf{x}_{t+1}^{\text{scale}}}{\partial \kappa^{\text{scale}}}$ (explanation below), we wrap the network operation with additional normalization and its inverse transform and differentiate it with respect to the parameterization κ . Therefore, the output directly reflects the estimated $\frac{\partial J(\mathbf{x}_{t+1})}{\partial \kappa}$.

Rescaling computed sensitivity Given that we train the neural networks using normalized data, the most straightforward way to compute the adjoint sensitivity is to pass the model output to the scalar-value objective function and differentiate this function with respect to the input parameterization, $\frac{\partial J^{\text{scale}}}{\partial \kappa^{\text{scale}}}$, and perform associated transformations to reach $\frac{\partial J}{\partial \kappa}$. However, we now show that this is not feasible without explicit access to $\frac{\partial \mathbf{x}_{t+1}^{\text{scale}}}{\partial \kappa^{\text{scale}}}$. For simplicity, we now use $\mathbf{x}$ to denote the model output $\mathbf{x}_{t+1}$.

With $J(\mathbf{x}) = \frac{1}{2} \|\mathbf{x}\|_2^2$ and normalization transformation listed in Section 3, in the normalized space, we can easily compute

$$\frac{\partial J(\mathbf{x}^{\text{scale}})}{\partial \kappa^{\text{scale}}} = (\mathbf{x}^{\text{scale}})^\top \frac{\partial \mathbf{x}^{\text{scale}}}{\partial \kappa^{\text{scale}}}. \quad (1)$$

Our final objective is to compute the sensitivity in the original scale, plugging (1) in,

$$\begin{aligned} \frac{\partial J(\mathbf{x})}{\partial \kappa} &= \mathbf{x}^\top \frac{\partial \mathbf{x}}{\partial \kappa} = \mathbf{x}^\top \frac{\sigma_{\mathbf{x}}}{\sigma_{\kappa}} \frac{\partial \mathbf{x}^{\text{scale}}}{\partial \kappa^{\text{scale}}} \\ &= \frac{(\sigma_{\mathbf{x}} \mathbf{x}^{\text{scale}} + \mu_{\mathbf{x}})^\top \frac{\sigma_{\mathbf{x}}}{\sigma_{\kappa}} \frac{\partial \mathbf{x}^{\text{scale}}}{\partial \kappa^{\text{scale}}} \frac{\partial J(\mathbf{x}^{\text{scale}})}{\partial \kappa^{\text{scale}}}}{(\mathbf{x}^{\text{scale}})^\top \frac{\partial \mathbf{x}^{\text{scale}}}{\partial \kappa^{\text{scale}}}}. \end{aligned} \quad (2)$$

Since $\mu_{\mathbf{x}} \neq 0$ and $\mathbf{x}$ is not a scalar, we require the access to $\frac{\partial \mathbf{x}^{\text{scale}}}{\partial \kappa^{\text{scale}}}$ to perform the transformation in (2), which requires additional and less efficient steps to obtain in the reverse-mode auto-differentiation

process. Therefore, we use the objective function directly taking the input in the original scale and add additional normalizing and unnormalizing steps to directly compute $\frac{\partial J(\mathbf{x})}{\partial \kappa}$.

B.3 Model training and evaluation

We adopt the FNOs implemented in PhysicsNeMo [23] and use the default hyperparameters as the baseline for all surrogates in this work. The domain of the ocean states is nearly circular in the horizontal direction, and the horizontal mask is not constant across vertical layers because the domain of interest is bowl-shaped, with horizontal area decreasing as depth increases. As a result, we apply masking during training and compute the loss only over values inside the domain. No special constraints are placed outside the domain; both true and predicted values are simply set to zero.

Once the models are trained, we evaluate them on the testing set and report the root mean squared error (RMSE) of model performance across various state variables for single-step forward predictions. In addition, we use the *time-averaged* ocean states as a constant predictor to set a benchmark for model evaluation. Different from training, we do not set values outside the domain to zero; instead, we exclude them from the calculation and only account for values within the domain of interest.

We randomly select a simulation associated with a unique κ in the testing set to evaluate and visualize model performance on autoregressive rollout and adjoint sensitivity estimation. For the rollout, we compute the spatially averaged values of each ocean state at each time step and compare them to the ground truth. For adjoint sensitivity, since true derivatives are not available, we report only the time-averaged adjoint sensitivities of variables at different vertical levels.

B.4 Hyperparameter optimization details

With the baseline model, we aim to conduct large-scale HPO to (1) further improve predictive accuracy and (2) create a candidate pool for constructing a deep ensemble for stable rollout and improved adjoint sensitivity estimation with quantified uncertainty. We utilize a *12-dimensional* hyperparameter search space spanning FNO architectures, data transformations, and training strategies. A detailed description of the search space is provided in Table 3. The optimization objective is set to minimizing the validation loss, and we adopt the 1-epoch strategy [24] to avoid excessive computational overhead during the search. We employ 40 NVIDIA A100 GPUs for parallelized evaluations, obtaining over 500 hyperparameter configurations. Figure 5 shows the search trajectory, which gravitates toward hyperparameter configurations outperforming the baseline (marked in red). This is accompanied by an increased number of high-performing models compared to the baseline. We then construct ensembles from the top- K configurations and train each member for 64K steps. Finally, we apply linear regression to re-weight the ensemble members, as discussed in Section 3, to further improve performance.

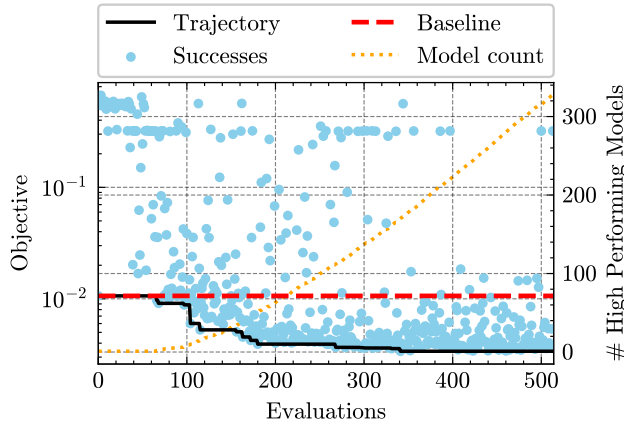


Figure 5: Hyperparameter search trajectory. Starting with the baseline configuration, the search balances the exploration and exploitation which leads to increasingly better models depicted by the solid black line. The number of high performing models increases accordingly.

Table 3: Hyperparameter search space

Variable Names.	Type	Range/Choice	Baseline
padding	int	[1, 16]	8
padding_type	str	['constant', 'reflect', 'replicate', 'circular']	constant
coord_feat	bool	[True, False]	True
lift_act	str	['relu', 'leaky_relu', 'prelu', 'relu6', 'elu', 'selu', 'silu', 'gelu', 'sigmoid', 'logsigmoid', 'softplus', 'softshrink', 'softsign', 'tanh', 'tanhshrink', 'threshold', 'hardtanh', 'identity', 'squareplus']	gelu
num_FNO	int	[2, 32]	4
num_modes	int	[2, 32]	16
latent_ch	int	[2, 64]	32
num_projs	int	[1, 16]	1
proj_size	int	[2, 32]	32
proj_act	str	['relu', 'leaky_relu', 'prelu', 'relu6', 'elu', 'selu', 'silu', 'gelu', 'sigmoid', 'logsigmoid', 'softplus', 'softshrink', 'softsign', 'tanh', 'tanhshrink', 'threshold', 'hardtanh', 'identity', 'squareplus']	silu
lr	float	$[10^{-6}, 10^{-2}]$	10^{-3}
weight_decay	float	[0, 0.1]	0

B.5 Additional results

We list the visualizations of the single time-stepping prediction, autoregressive rollout, and parametric sensitivity estimates of all ocean states in this section.

For the single time-stepping prediction of all ocean states, Figure 6 visualizes the baseline performance on a testing data point. At the sea surface (depth 0), the baseline model captures state fields that closely resemble the true ocean states. However, compared to the zonal and meridional velocities, forecasts of the other states exhibit relatively larger fraction errors. Such errors can reduce the reliability of adjoint sensitivity estimates and accelerate error accumulation during autoregressive rollout, underscoring the need for improved modeling of ocean dynamics. For this particular case, the best model from HPO does not show a clear improvement over the baseline, reflecting only marginal performance gains. In contrast, the Top-10 ensemble improves the forecast of layer thickness, yielding lower fraction errors than both the baseline and optimal models. The weighted Top-10 ensemble performs similarly to the uniform Top-10 ensemble, with both ensembles outperforming the baseline and single best models.

Figure 7 shows the parametric sensitivity estimates from all models considered in this work across the ocean states. Although the baseline and best models produce similar forward predictions, their sensitivity estimates differ substantially. In contrast, the two ensembles exhibit only marginal differences in sensitivity estimates and show strong agreement in predictive uncertainty.

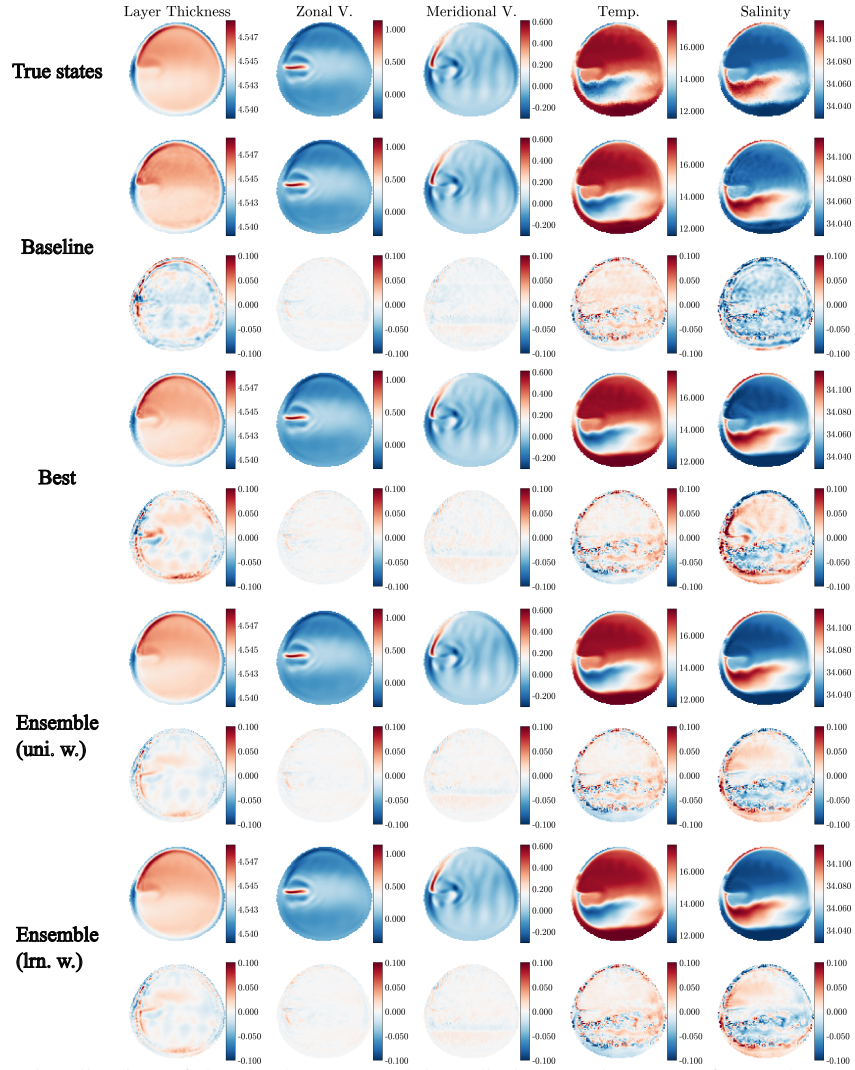


Figure 6: Visualization of the single-step model predictions at the sea surface. The top most row shows the true ocean states; in each inset thereafter, the top row is the predicted states and bottom row shows the fractional error of the predicted fields.

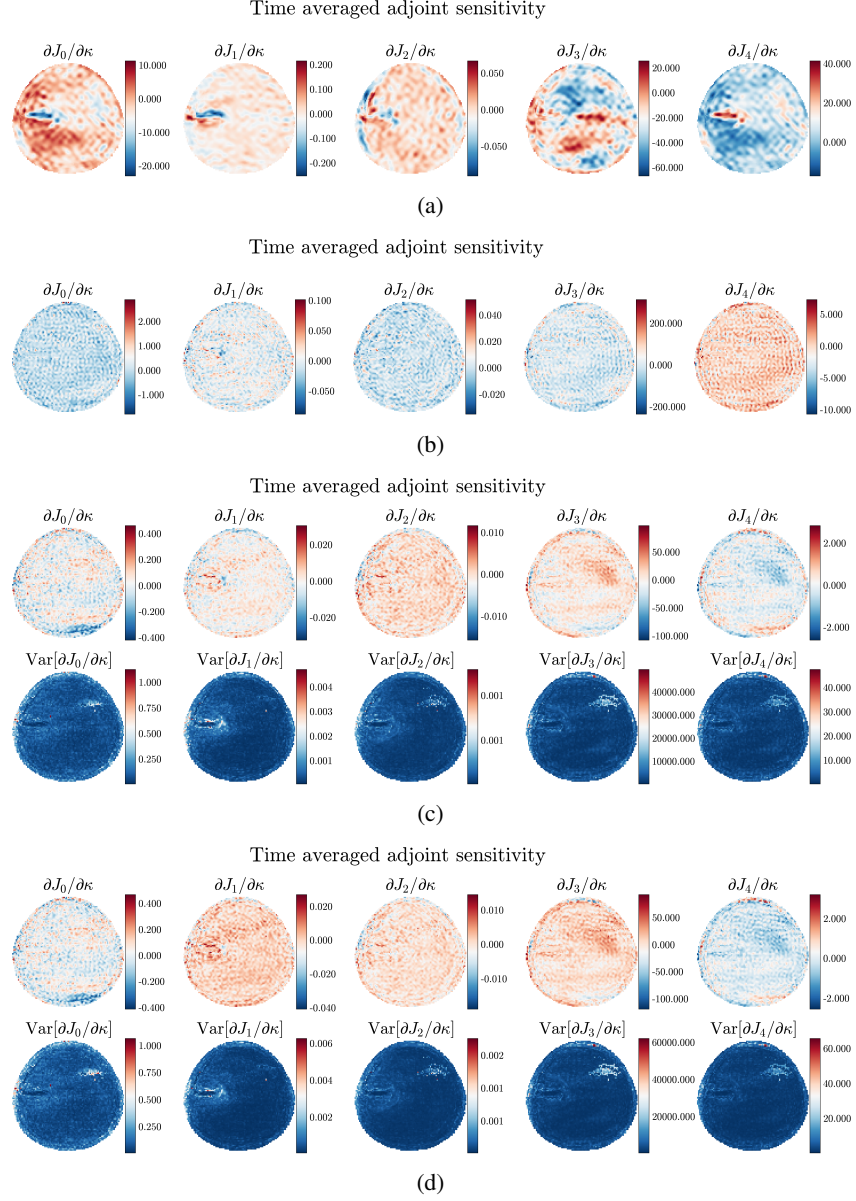


Figure 7: Time-averaged parametric sensitivity of ocean states at the sea surface with respect to the parameterization. The order of the states are layer thickness, zonal velocity, meridional velocity, temperature, and salinity. (a) baseline; (b) optimal model; (c) Top-10 ensemble; (d) Weighted Top-10 ensemble.