JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2023MS003890

Special Section:
Machine learning application to Earth system modeling

Parameterizing Vertical Mixing Coefficients in the Ocean Surface Boundary Layer Using Neural Networks

Aakash Sane[1], Brandon G. Reichl[2], Alistair Adcroft[1], and Laure Zanna[3]

[1]Atmospheric and Oceanic Sciences Program, Princeton University, Princeton, NJ, USA, [2]NOAA – Geophysical Fluids Dynamics Laboratory, Princeton, NJ, USA, [3]Courant Institute, New York University, New York, NY, USA

Key Points:

- We improve a parameterization of vertical mixing in the ocean surface boundary layer using neural networks
- Neural networks are trained to predict the diffusivity of second moment closure and maintain energetic constraints of the original parameterization
- The improved scheme reduces biases of mixed layer depth and thermocline in an atmospherically forced ocean model

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

A. Sane,
aakash.sane@princeton.edu

Citation:

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*, e2023MS003890. https://doi.org/10.1029/2023MS003890

Received 17 JUN 2023
Accepted 27 SEP 2023

Author Contributions:

Conceptualization: Aakash Sane
Formal analysis: Aakash Sane
Investigation: Aakash Sane
Methodology: Aakash Sane
Writing – original draft: Aakash Sane

Abstract Vertical mixing parameterizations in ocean models are formulated on the basis of the physical principles that govern turbulent mixing. However, many parameterizations include ad hoc components that are not well constrained by theory or data. One such component is the eddy diffusivity model, where vertical turbulent fluxes of a quantity are parameterized from a variable eddy diffusion coefficient and the mean vertical gradient of the quantity. In this work, we improve a parameterization of vertical mixing in the ocean surface boundary layer by enhancing its eddy diffusivity model using data-driven methods, specifically neural networks. The neural networks are designed to take extrinsic and intrinsic forcing parameters as input to predict the eddy diffusivity profile and are trained using output data from a second moment closure turbulent mixing scheme. The modified vertical mixing scheme predicts the eddy diffusivity profile through online inference of neural networks and maintains the conservation principles of the standard ocean model equations, which is particularly important for its targeted use in climate simulations. We describe the development and stable implementation of neural networks in an ocean general circulation model and demonstrate that the enhanced scheme outperforms its predecessor by reducing biases in the mixed-layer depth and upper ocean stratification. Our results demonstrate the potential for data-driven physics-aware parameterizations to improve global climate models.

Plain Language Summary The upper region of the ocean is highly energetic and is responsible for transferring mass, energy and biogeochemical tracers between the atmosphere and the deeper regions of the ocean. This transport takes place because of turbulent swirling motions, which are found to be of varying sizes. Climate models cannot represent all of these motions because smaller-scale swirls are complex and require additional computational resources. As we cannot neglect those small swirls, we try to approximate their effects on larger-scale motions using mathematical models. These models have a few ad hoc or empirical assumptions that lead to uncertainty when these climate models are used to project the future climate. To reduce this uncertainty, we augment an existing model of turbulent swirling process with machine learning, which replaces some ad hoc approximations with data-driven neural networks. Neural networks can learn those missing processes more accurately than a traditional physics-based model. The neural networks are shown to improve physics in climate simulations. Although we only touch on one component in an ocean climate model, this approach can be replicated to improve any other component that was using ad hoc assumptions and replace them with data-driven models using techniques from machine learning.

1. Introduction

Vertical mixing parameterizations used in ocean general circulation models (OGCM) represent the effects of unresolved processes on the mean state. These parameterizations have theoretical deficiencies due to the lack of understanding of inadequately represented or missing processes. To overcome this deficiency, parameterizations often require ad hoc/empirical modifications either to approximate the missing processes or to fit data. Vertical mixing schemes can be constructed with various assumptions and different schemes are calibrated differently. These inconsistencies cause the schemes to disagree among themselves (Li et al., 2019) and are a major source of model uncertainty (Fox-Kemper et al., 2019; Gutjahr et al., 2021; Hawkins & Sutton, 2009; Huber & Zanna, 2017; Todd et al., 2020). Poorly parameterized mixing can result in errors that accumulate over time, leading to biases in the OGCM.

New approaches are emerging to improve various parameterizations in ocean and atmosphere models using machine learning. We have applied neural networks, a type of machine learning, to improve a vertical mixing

parameterization of the ocean surface boundary layer (OSBL). OSBL is a vital region of turbulence in the ocean. It acts as an interface between the atmosphere and the deeper ocean and it is important to accurately represent mixing in the OSBL. The atmosphere energizes the ocean through the OSBL. Mass, tracers, and momentum are transferred between the atmosphere and deep ocean via the OSBL, and inaccuracies in vertical mixing parameterizations can give rise to uncertain estimates of heat transport, sea level rise, ocean carbon uptake, etc. Including missing processes in upper ocean vertical mixing schemes impact large-scale phenomena, for example, accounting for Langmuir turbulence and submesoscale effects in the OSBL improves simulations of the Indian monsoon (Orenstein et al., 2022).

1.1. Modeling Vertical Diffusivity Within Ocean Surface Boundary Layer (OSBL) Parameterizations and the Assumption of a "Universal" Shape Function

We focus on the energetic Planetary Boundary Layer (ePBL) scheme, a first-order OSBL turbulent mixing parameterization as described in Reichl and Hallberg (2018) (see Section 2). The variation of the vertical diffusivity profile κ_ϕ (of arbitrary scalar, ϕ) within the OSBL in ePBL and similar first order schemes can be expressed as a diffusivity scale ($\hat{\kappa}_\phi$) multiplied by a prescribed normalized diffusivity profile (i.e., shape function):

$$\kappa_\phi(\sigma) = \hat{\kappa}_\phi g(\sigma), \tag{1}$$

where $\hat{\kappa}_\phi$ is often decomposed into a velocity and length scale (Large et al., 1994), $g(\sigma)$ is a dimensionless shape-function, and $\sigma = z/h$ is a dimensionless vertical coordinate, where z is the vertical coordinate and h is the depth of the boundary layer. OSBL parameterizations that follow this approach traditionally assume that $g(\sigma)$ is a universal function or has a fixed component such as a cubic polynomial that does not change (Large et al., 1994; O'Brien, 1970), and therefore is ad-hoc. In the K-profile-parameterization (KPP) scheme of Large et al. (1994), there is a cubic polynomial which is multiplied by a vertically varying turbulent velocity that sets the structure of κ_ϕ. The cubic polynomial is universal, whereas turbulent velocity mostly affects the surface layer defined by the region $0 < \sigma < 0.1$, making the cubic structure dominant below the surface layer. In ePBL scheme (Reichl & Hallberg, 2018), κ_ϕ follows similar design. However, there is no physics-based justification for a universal or ad-hoc profile to exist, and it is widely understood that characteristics of boundary layers can vary considerably with forcing conditions (Li et al., 2019). We hypothesize that capturing variations of the shape function that are not considered in first-order OSBL schemes such as ePBL will improve the overall representation of vertical mixing in ocean models. In the subsequent text, our usage of the term "universal shape function" will include shape functions which involve some ad-hoc components or approximations such as used in the ePBL scheme (see Section 2).

1.2. Second Moment Closure and an Alternative to the "Universal" Shape Function

Second Moment Closure (SMC) is an alternative approach to predict vertical diffusivity profiles within the OSBL (Rodi, 1987; Umlauf & Burchard, 2005). SMC does not require a shape function because it instead predicts the diffusivity from the turbulent kinetic energy (k) and the turbulent length scale (l_t). Various SMC approaches exist to predict k and l_t and a general formulation to infer diffusivity is expressed as:

$$\kappa_\phi(z) = c_\phi \, k^{1/2}(z) \, l_t(z), \tag{2}$$

where c_ϕ represents the model stability functions (Umlauf & Burchard, 2005).

SMC predicts a profile of vertical diffusivity based on models of physical processes that drive turbulent fluxes within the OSBL. SMC does not prescribe a shape function a priori. However, since SMC directly evaluates a diffusivity profile, the implied shape function and diffusivity scale can be diagnosed from the output. The implied shape function differs significantly from a universal shape function, as seen in Figure 1. The diagnosed shape-function and diffusivity scale from SMC can then be used to build a model for use in ePBL. We selected SMC over large eddy simulation as our "truth" because it is inexpensive compared to the latter, leading to effortless creation of training data set spanning a wide range of forcing regimes. This is required for machine learning applications as they are hungry for a large amount of data.

A natural question is why SMC is not directly used instead of ePBL in OGCM. It remains impractical to directly use SMC for vertical mixing in climate simulation due to the sensitivity of their predictions to long

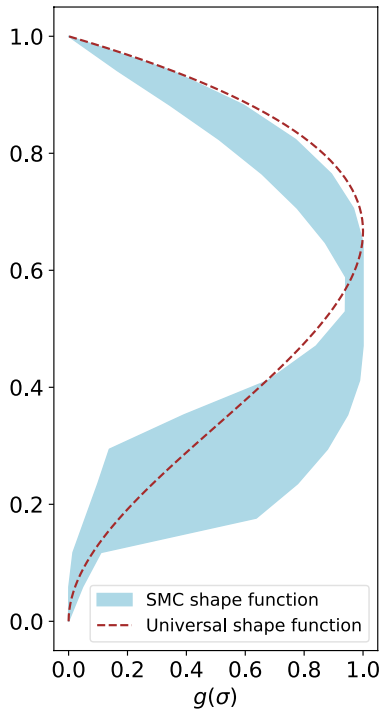

Figure 1. Shape functions derived from various forcing conditions from a Second Moment Closure (SMC) (blue, shaded region) plotted against a universal shape function (brown, dashed line) used in general circulation model vertical mixing schemes. The observed discrepancy between them reveals a limitation in existing vertical mixing schemes. However, this deficiency can be effectively addressed through the application of neural networks, which have the potential to predict the shape function and diffusivity associated with second moment closures.

time steps and coarse vertical grids often used in climate models (see Reichl & Hallberg, 2018). However, using the framework described in this article, ePBL can yield a closer approximation to the vertical diffusivity from the SMC scheme without sensitivity to the model's vertical resolution and time step. Our neural network approach allows ePBL to consider the physics-based variation in the shape function seen in SMC due to solving k and l_t. This variability in the shape function will lead to different profiles of vertical mixing within ePBL than using a prescribed universal profile.

1.3. Machine Learning Is an Emerging Tool to Improve OGCMs

Consider a physics-based parameterization that gives an output Ψ as some functional relationship $\mathcal{F}$ between physical quantities $\mathbf{x}$:

$$\Psi = \mathcal{F}(\mathbf{x}). \tag{3}$$

Finding $\mathcal{F}$ is an optimization problem. It can be set as an optimal linear fit to some combination of $\mathbf{x}$, but the fit might not work for different regimes or might implicitly depend on higher-order combinations of terms in $\mathbf{x}$ (nonlinearity) or some other neglected terms. $\mathcal{F}$ can be assumed to be a function of non-dimensional parameters requiring onerous fitting. With machine learning, $\mathcal{F}$ can be a function of multiple combinations of parameters:

$$\mathcal{F}(\mathbf{x}) = \mathcal{N}^{\mathbf{w}}(\mathbf{x}), \tag{4}$$

where $\mathcal{N}$ is a machine-learning function, $\mathbf{x} = (x_1, x_2, \ldots)$ is the input vector and $\mathbf{w}$ are parameters (weights and biases). Machine learning involves determining (learning) the correct values of $\mathbf{w}$ by tuning the hyperparameters that give the optimal $\mathcal{N}$ (Brenner et al., 2019), which is becoming routine due to advances in training algorithms. The machine learning approach provides an avenue to include as many relevant parameters as desired in the vector $\mathbf{x}$, which has been a significant challenge in traditional physics-based approaches.

Machine learning is favorable for the development and application of climate models due to the abundance of optimization algorithms and hardware (Balaji et al., 2022; Christensen & Zanna, 2022). Studies show that neural networks can be used in idealized model configurations, and recently, the use of machine learning has emerged in realistic GCMs. Artificial neural networks (ANNs) have been shown to improve sub-grid momentum transport in atmospheric models (Yuval & O'Gorman, 2023), predict precipitation (Shamekh et al., 2023) and fluxes (Shamekh & Gentine, 2023), while in ocean models they have been used to improve the parameterization of free convection (Ramadhan et al., 2023). Liang et al. (2022) applied deep neural networks to predict temperature and salinity evolution in the OSBL at a weather station (Station Papa). Partee et al. (2022) trained a deep neural network to learn subgrid kinetic energy of oceanic mesoscale eddies from a high resolution OGCM to improve their representation in a lower resolution OGCM. Convolutional neural networks (CNNs) have been used to predict parameterizations of ocean momentum backscatter in a variety of models (Bolton & Zanna, 2019; Guillaumin & Zanna, 2021; Zanna & Bolton, 2020) and have been implemented in an ocean primitive equation model (Zhang et al., 2023). Gregory et al. (2023) recently employed CNNs to learn data assimilation increments for sea-ice and showed that networks could be used to reduce biases in sea-ice.

Apart from neural networks, techniques considered part of the machine learning toolbox show potential to improve GCMs. The random forest algorithm has been used to parameterize moist convection (O'Gorman & Dwyer, 2018) and to learn small-scale processes from a high resolution atmospheric model (Yuval & O'Gorman, 2020). Mansfield and Sheshadri (2022) used Gaussian Process emulator to tune gravity wave parameterization in

an intermediate complexity atmospheric GCM. Souza et al. (2020) use a Bayesian technique to fine-tune the non-local flux terms of the KPP parameterization of Large et al. (1994).

The aforementioned examples show the potential of enhancing conventional physics-based schemes using machine learning techniques. This article draws inspiration from these demonstrations, recognizing the promise of machine learning in advancing ocean model parameterizations and prompting further investigation in this area.

1.4. Outline to Use Neural Networks and Output From SMC to Improve ePBL

ANNs are trained using output from SMC that directly predicts the profile of vertical diffusivity and do not rely on ad hoc shape functions. As neural networks are powerful approximators, they can model the variability in the vertical diffusivity profiles of the SMC, but we formulate the ANNs to fit within the simplified framework of the first-order ePBL approach. Our procedure has the following advantages:

1. We use the neural networks to modify the vertical diffusion term within ePBL instead of directly predicting turbulent flux time tendencies (e.g., temperature and salinity), guaranteeing that the scheme conserves physical quantities.
2. The neural networks are introduced in a manner that does not interfere with the potential energy-based mixing constraints of the original ePBL scheme, and therefore ePBL's robust numerical implementation is preserved.
3. The ANNs predict quantities used to compute the diffusivity: the non-dimensional structure (shape function) and a turbulent velocity, which simplifies training, implementation, and interpretability versus directly predicting the diffusivities.
4. ANNs yield strictly positive values of the vertical diffusivity, an important consideration for numerical stability (see Section 3.4.2).
5. Our ANNs are as small as possible to balance accuracy and computational costs, as they will be used in climate timescale OGCM simulations.

We structure the article as follows. Section 2 describes the ePBL scheme and briefly addresses a calibration/tuning problem. Section 3 gives details of the network structure and describes the data used to train networks with estimates of uncertainty. Section 4.1 provides details on implementing the enhanced ePBL scheme, hereafter called ePBL_NN. The new improvements in ePBL_NN are demonstrated online using free-running single-column model experiments (Section 4.2), and their impact on biases in an existing ocean-ice climate model is assessed in Section 4.3. We conclude with a summary and discussion of the broader implications of this work for applying machine learning to improve parameterizations in ocean climate models.

2. A Physics-Based Vertical Mixing Framework: The Energetic Planetary Boundary Layer (ePBL)

The ePBL framework, as described by Reichl and Hallberg (2018), is designed for climate applications of OGCMs and emphasizes robust solutions to changes in model time stepping and vertical resolution. The scheme is simple enough to implement efficiently within implicit diffusion solvers often used in OGCMs while maintaining important physical constraints on ocean mixing. The ePBL scheme performs with high skill in idealized models and OGCMs (Li et al., 2019; Reichl & Li, 2019), and has been implemented in NOAA—Geophysical Fluid Dynamics Laboratory (GFDL)'s MOM6-based climate models: OM4, CM4, and ESM4 (Adcroft et al., 2019; Dunne et al., 2020; Held et al., 2019).

ePBL builds on the paradigm of bulk mixed layer models (Kraus & Turner, 1967; Niiler, 1977), which constrain the boundary layer depth (h) via energetic implications of vertical mixing. The scheme therefore constrains the mixing based on parameterizing the rate by which turbulent kinetic energy is converted to potential energy within the OSBL:

$$\overbrace{\int_{-h}^{0} \min\left(0, \overline{w'b'}\right) dz}^{\text{integrated PE conversion}} = \mathcal{G}\left(\overbrace{f}^{\text{Coriolis}}, \overbrace{u_*}^{\text{Wind}}, \overbrace{B_0}^{\text{Buoyancy}}, \overbrace{h}^{\text{BLD}}, \overbrace{\int_{-h}^{0} \max\left(0, \overline{w'b'}\right) dz}^{\text{Convective PE release}} \right). \tag{5}$$

Here, h is the (positive) depth of the boundary layer as defined in Reichl and Li (2019), $\overline{w'b'}$ is the vertical turbulent buoyancy flux, overbar represents an averaging procedure (e.g., over ensembles), and $\mathcal{G}$ is a relation that depends on the Coriolis parameter f, surface friction velocity u_*, surface buoyancy flux $B_0 = \overline{w'b'}_0$, boundary layer depth h, and integrated release of potential energy by convective buoyancy fluxes. Reichl and Hallberg (2018) find $\mathcal{G}$ using simulations from single column models using SMC under a range of forcing scenarios. Later, this function was enhanced to include Langmuir turbulence using large eddy simulations (LES) (Reichl & Li, 2019).

ePBL extends the bulk mixed layer formulation to resolve vertical structure within the OSBL by applying a down-gradient flux profile using the vertical diffusivity given by

$$\overline{w'\phi'} = -\kappa_\phi \frac{\partial \overline{\phi}}{\partial z}, \tag{6}$$

where κ_ϕ is the variable diffusivity of a scalar ϕ. The diffusivity varies with depth and is given in the following form:

$$\kappa_\phi(\sigma) = L(\sigma)v_0(\sigma), \tag{7}$$

where L and v_0 are length and velocity scales. In the present implementation of ePBL (Reichl & Hallberg, 2018), the turbulent Prandtl number is assumed to be one and hence the diffusivity and viscosity are identically modified. Both L and v_0 are expressed as functions of position σ within the boundary layer. The length scale in Equation 7 is set as (following O'Brien, 1970; Large et al., 1994):

$$L(\sigma) = (z_o + |z|)(1-\sigma)^\gamma. \tag{8}$$

By assuming a fixed constant for γ, the expressions given by Equations 7 and 8 may be expressed in the same form as Equation 1, which reveals the role of the shape function as $g(\sigma) = \sigma(1-\sigma)^\gamma$. γ should not be a fixed constant. Constructing γ as a data-driven function is challenging and the form $\sigma(1-\sigma)^\gamma$ does not have a physical basis. The velocity scale $v(\sigma)$ uses a similar formulation motivated to generally agree with the model $k-\epsilon$ (see Equations 43–45 in Reichl & Hallberg, 2018).

Although the integrated mixing in ePBL is constrained via the function $\mathcal{G}$, the stratification resulting from the mixing is sensitive to the assumptions for γ and v_0 that set the diffusivity profile within the boundary layer. Differences in the diffusivity profile mean that even when the energetic constraints are accurate, inconsistent OSBL evolution and stratification can emerge when comparing ePBL with SMC such as $k-\epsilon$ (see Figures 6 and 7). In this article, we enhance the physics-based ePBL approach by improving these velocity- and length-scale formulations with ANNs (see Section 3).

3. Artificial Neural Networks and Training Procedures

ANNs are one of the most widely used forms of machine learning models. ANNs are universal approximators and can find hidden nonlinear relations between quantities (Cybenko, 1989; Hornik, 1991; Hornik et al., 1989). In this section, we describe the fundamentals of ANNs and provide details describing the training procedures for the neural networks used to supplement ePBL's eddy diffusivity model.

3.1. Fundamentals

ANNs consist of nodes arranged in layers. Nodes are elements of a vector $\mathbf{x}$ that constitute a layer. See Figure 2 for a schematic. Each vector is connected to its adjacent vector via a transformation that involves multiplying with coefficients, called weights $\mathbf{w}$, and adding an offset, called biases $\mathbf{b}$. After transforming the vector with weights and biases, a nonlinear operation yields the next vector (or layer). The nonlinear operator is an activation function $\mathcal{A}$. For an input layer consisting of a vector $\mathbf{x_1}$, one hidden layer $\mathbf{x_2}$ and an output layer $\mathbf{y}$, ANN can be written as$_2$ and an output layer $\mathbf{y}$, ANN can be written as

$$\begin{aligned} \mathbf{x}_2 &= \mathcal{A}(\mathbf{w}_1\mathbf{x}_1 + b_1), \\ \mathbf{y} &= (\mathbf{w}_2\mathbf{x}_2 + b_2). \end{aligned} \tag{9}$$

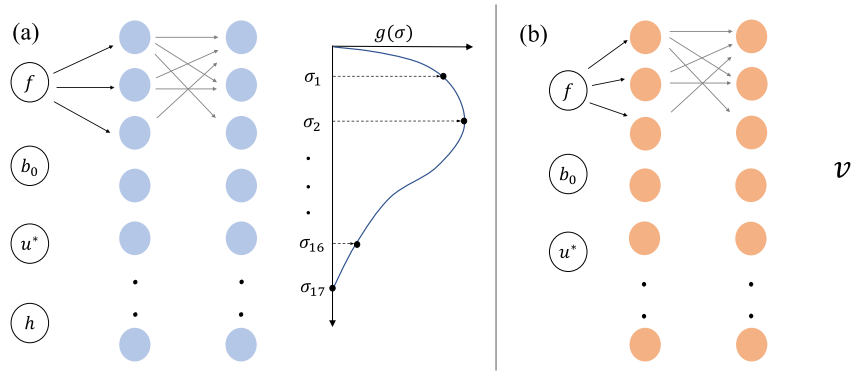

Figure 2. (a) Neural network $\mathcal{N}_1$. It requires four inputs (f, B_0, u_*, h) and output layer consists of 16 nodes giving values of $g(\sigma)$ at those locations. (b) Neural network $\mathcal{N}_2$ requires three inputs (f, B_0, u_*) and output is a scalar velocity scale v_0. Diffusivity is obtained by: $\kappa(\sigma) = g(\sigma) \cdot h \cdot v_0$. Here, h is the boundary layer depth which is evaluated in the vertical mixing parameterization of ocean surface boundary layer in ocean general circulation model using physical arguments.

Deeper networks are expanded versions of the above Equation set 9 and are obtained by adding additional layers. ANNs can capture nonlinear relationships within certain tolerances and can interpolate with high accuracy within the range of training data. We employ ANNs to learn the nonlinear relationship between chosen input parameters, described below, and the vertical diffusivity profile predicted using SMC.

3.2. Learning Diffusivity Using Two Neural Networks $\mathcal{N}_1$ and $\mathcal{N}_2$

To train the ANN model to predict the diffusivity profile, we use the sigma coordinate defined as $\sigma = z/h$. Therefore, at the surface $\sigma = 0$, and at h, $\sigma = 1$. We define the diffusivity in the sigma coordinate in terms of a velocity scale, v_0, the boundary layer depth, h, and the nondimensional shape function, g:

$$\kappa_\phi(\sigma) = g(\sigma) \cdot h \cdot v_0, \tag{10}$$

where $g(\sigma)$ is defined to give values between [0,1]. We could have introduced vertical structure in few or all of the terms on the right hand side in Equation 10. Instead we use only $g(\sigma)$ to provide vertical structure as we found out that it was convenient to train one profile than two or more. The benefit of adopting the sigma coordinate is in removing the dependence on the vertical coordinate (e.g., grid spacing in z) that varies in different ocean models. This allows us to train and infer (feed-forward) without depending on the model's vertical grid, which makes it practical to implement in an ocean model with an adaptive vertical grid (e.g., Bleck, 2002).

The velocity scale v_0 in Equation 10 does not vary with σ. The entire vertical structure of κ_ϕ is captured by $g(\sigma)$ alone. This is in contrast to Equation 7 where both the length scale and the velocity scale vary in vertical direction and contribute to the vertical structure of κ_ϕ. We made this choice to simplify the approach so that only one neural network is needed to capture the vertical structure of κ_ϕ.

We choose to obtain the shape function and velocity scale using two separate neural networks:

$$\begin{aligned} g(\sigma) &= \mathcal{N}_1(f, B_0, u_*, h), \\ v_0 &= \mathcal{N}_2(f, B_0, u_*), \end{aligned} \tag{11}$$

where $\mathcal{N}_1$ and $\mathcal{N}_2$ represent two distinct neural networks that are trained independently. $\mathcal{N}_1$ requires inputs f, B_0, u_*, and h, while $\mathcal{N}_2$ is found to depend on f, B_0, and u_*. We chose this strategy rather than combining the two outputs into one ANN for a couple of reasons. First, it is straightforward to cleanly diagnose $g(\sigma)$ and v_0 from the data, as will be explained in Sections 3.4 and 3.5. Second, we anticipate that having separate networks will make the individual networks easier to interpret, which allows us to better understand physical processes modeled by the network.

Both neural networks $\mathcal{N}_{1,2}$ are trained using the Pytorch package (Paszke et al., 2019). Rectified Linear Unit (Nair & Hinton, 2010) has been used as the activation function due to its simplicity and rapid convergence in training.

Table 1
Range of Parameters Used to Generate Training Data

Inputs	$\mathcal{N}_1$	$\mathcal{N}_2$
Surface heat flux	−600 to 600 W/m^2	−2,000 to 2,000 W/m^2
Wind stress	0–1.2 N/m^2	0–20 N/m^2
Surface friction velocity	0 to 0.034 m/s	0 to 0.034 m/s
Latitude	−90° to 90°	−90° to 90°
Boundary layer depth	20–300 m	–
Reference density	1,027 kg/m^3	1,027 kg/m^3
Specific heat capacity	3,985 J kg^{-1}K^{-1}	3,985 J kg^{-1}K^{-1}
Equation of state	Linear	Linear
Stratification at initial conditions	0.005°K/m	0.005°K/m

Note. We have added additional details about GOTM runs to Supporting Information S1.

3.3. Data for Training

The SMC data used to train the networks is generated using the single column model framework implemented in the General Ocean Turbulence Model (GOTM; Umlauf & Burchard, 2005; Umlauf et al., 2014). GOTM provides numerous SMC options to predict the fluxes of turbulence and the vertical diffusivity. We employ a two-equation model: $k-\epsilon$, with stability function closure following Schumann and Gerz (1995). The choice of this specific SMC parameterization is made to be consistent with Reichl and Hallberg (2018). Vertical mixing parameterizations remain an active research topic, and currently used schemes, including SMC, can exhibit biases in different forcing regimes and regions (Damerell et al., 2020; Li et al., 2019; Peters & Baumert, 2007; Sane et al., 2021; Tirodkar et al., 2022). Any biases in the training data are inherited by the neural networks. However, our neural networks can be trained using the output of different mixing schemes, including the improved schemes developed in future research.

The GOTM column model consists of a vertical grid with forcing applied at the surface grid point. It is applicable for flows with horizontal homogeneity, that is, horizontal fluxes are zero or constant. GOTM simulations are performed by changing the following parameters: latitude (Coriolis), surface wind stress (surface friction velocity), and surface heat flux (surface buoyancy flux). Salinity is kept constant and temperature is the only active tracer, though the results are general for any combination of buoyancy fields and forcing. Our initial analysis indicates that the diffusivity of $k-\epsilon$, κ_{ke}, depends on the Coriolis parameter f, the surface buoyancy flux B_0, the surface friction velocity u_* and the depth of the boundary layer, h. We can only specify f, B_0, and u_* in single column simulations, and h is diagnosed from the time evolution simulated by GOTM.

Each GOTM case runs with a set of constant forcings. The time step is set at 60 s, and the vertical grid spacing is 1 m. The depth of the column is 800 m. The simulation results for the $k-\epsilon$ model are converged at this time step and resolution (see Figure 1 in Reichl & Hallberg, 2018). The initial conditions consist of zero horizontal velocity, the surface temperature is set at 20°C, and the initial temperature stratification is set at 0.005°C/m. Data was saved at hourly intervals. For every f, B_0, and u_*, we included one hundred instantaneous profiles of diffusivity at each hour from day 2 to day 6 in the training data set.

We found (f, B_0, u_*, h) to strongly affect diffusivity compared to the background stratification established by the initial conditions. Stratification acts as a barrier to the deepening of the mixed layer, and therefore it is challenging to obtain deeper layers with strong stratification at the bottom of the mixed layer, and this limits the generation of training data spanning a wide range of h. Therefore, we choose a weak initial stratification. The effects of stratification on diffusivity in $k-\epsilon$ most directly impact the rate of deepening of the boundary layer (which is already captured by the energetic constraint of the ePBL), compared to the shape function itself.

Table 1 shows the range of forcing parameters of the training data. Forcing range is different for $\mathcal{N}_1$ and $\mathcal{N}_2$ because we found that the shape function does not vary significantly outside the range stated in Table 1. Hence we do not train on data outside that range, and the inputs to the network can be capped inside the mixing scheme in MOM6. For example, if the wind stress is 1.3 N/m^2, capping prevents the wind stress from going

beyond 1.2 N/m^2 as the shape function does not vary significantly beyond 1.2 N/m^2. A similar argument can be made about the surface heat flux. The range selected to perform the sweep has been informed using the observed forcings in the JRA atmospheric reanalysis data set (Tsujino et al., 2018). The range shown in Table 1 covers most of the forcing space certainty as explained in Appendix B. For h, maximum variations for $g(\sigma)$ were observed between 20 and 300 m and beyond 300 m g(σ) is found to vary marginally. Randomizing the training data and splitting it into two sets (train and validation) could result in very similar elements from similar experiments being present in both the train and the validation data. This is undesirable since a fully independent validation data set is required to monitor overfitting when training a neural network. To prevent this issue, a validation data set is independently generated having 10% the size of the training data using a fully independent set of forcing parameters. No single element between (f, B_0, u_*) is common between the training data set and the validation data set to strictly ensure the independence between the training and validation sets.

3.4. Training $\mathcal{N}_1$

The parameters f, B_0, u_*, and h are inputs to $\mathcal{N}_1$, while the output consists of a vector having values of $g(\sigma)$ at 16 evenly distributed nodes, as shown in Figure 2. For each set of forcing (i.e., latitude, heat flux, and surface stress), the GOTM output consists of the evolution of the initial conditions into a developed boundary layer. The boundary layer deepens and variations in $\kappa_{k\epsilon}$ emerge. Ignoring the initial ≈2 days of data, h is diagnosed for each model output with a frequency of 60 min by analyzing the profile of the vertical buoyancy flux. Here, h is defined by the depth at which $\overline{w'b'}$ reaches and stays close to zero. This is the maximum extent to which the effect of surface forcing penetrates the upper layer through turbulent buoyancy flux. The diffusivity profile, $\kappa_{k\epsilon}(\sigma)$, is normalized by its maximum to find the shape function:

$$g(\sigma) = \kappa_{k\epsilon}(\sigma)/\max(\kappa_{k\epsilon}(\sigma)). \tag{12}$$

The neural network cannot learn a continuous profile in σ, but instead we train it to learn on a subsampled $g(\sigma)$ grid that consists of 18 equally spaced σ points (0, 1/17, 2/17, …, 16/17, 1). σ at 0 and 1 is ignored in the training because $g(\sigma) = 0$ at the surface ($\sigma = 0$). At $\sigma = 1$, $g(\sigma) \to 0$ and hence is assumed to be zero for training purposes. Therefore, the network predicts $g(\sigma)$ at the 16 interior locations. Subsampling $g(\sigma)$ to 18 evenly distributed points was found to be sufficient to capture essential features of $g(\sigma)$ while maintaining a small enough network to later implement in an OGCM.

3.4.1. Overcoming Limitation in h Using Synthetic Data

ANNs show high prediction skill when input is within the range of training data. GOTM experiments can cover a wide range of data points that span latitude, heat flux, and surface wind stress, such as those historically observed in the real ocean. However, this is not true for h as it evolves prognostically and we cannot set its range for each run. We have chosen h to vary from 20 to 300 m (see Table 1) for training purposes, but for some surface heating conditions the boundary layer depth will saturate toward the Monin-Obukhov length L_{MO}, which might be less than 300 m. As h is an input to the network and if for a particular case $L_{MO} < h < 300$ m, then profiles will not exist and the network will have to predict outside the range of the training data set. The network might end up predicting spurious profiles.

We address this issue by supplementing the training with synthetic data. For a particular case, if h saturates to, for example, 200 m, then an additional 10 profiles are added to cover the missing range of 200–300 m in the training data. The shape function for these synthetic profiles is assumed to be the same as when $h = 200$ m, that is, $g(\sigma)$ for $h \in (200, 300)$ will have the same values as $g(\sigma)$ for $h = 200$ m. This assumption is reasonable, since $g(\sigma)$ was found to vary little for deeper boundary layers with surface heating.

Strong convection can cause a related issue due to quick deepening of the boundary layer within the spin-up phase of the turbulent OSBL. This gap is filled in the same way as described for deep boundary layer gaps. If the lowest value of h is, for example, 100 m, then 10 profiles are added that cover 20–100 m. The shape function for these 10 profiles is assumed to be the same as that when $h = 100$ m. This fill-up of gaps in h is necessary to stabilize ANNs trained with our existing data sets. Knowing the exact bounding box of the training data set is imperative for a successful and stable implementation in a GCM.

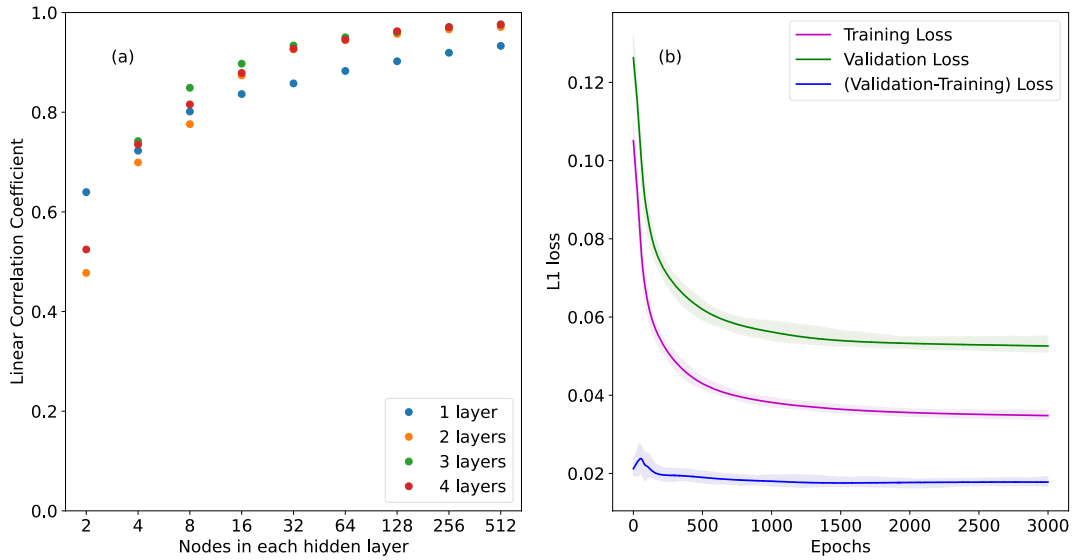

Figure 3. (a) Average linear correlation coefficient between network output and the true values from data. Averaging has been done over all the 16 nodes. Different color represent different number of layers and x-axis shows the nodes in each hidden layer. (b) L1 loss curves for training and validation.

3.4.2. Forcing $\mathcal{N}_1$ to Be Strictly Positive

The network $\mathcal{N}_1$ consists of 4 input nodes, 2 hidden layers, and 16 output nodes (sensitivity to network hyperparameters is described in the next subsection). The four input nodes correspond to (f, B_0, u_*, h). The output nodes predict the shape function as described above. The output of $\mathcal{N}_1$, $g(\sigma)$, is a vector of length 16. If $g(\sigma)$ predicts a negative value of the shape function for any σ, it would lead to negative diffusivity values. We prevent this by training on the logarithm of $g(\sigma)$. $\mathcal{N}_1$ predicts $\log(g(\sigma))$ and, while inferring, the exponential function is used. This ensures that the shape function is strictly positive.

The four inputs to the network (f, B_0, u_*, h) are normalized by their respective mean and standard deviation of the training data. For the 16 output nodes, each output was normalized by its own mean and standard deviation. For output node i, $\log(g(\sigma_i))$ was transformed into

$$\log(g(\sigma_i)) \rightarrow \frac{\log(g(\sigma_i)) - \overline{\log(g(\sigma_i))}}{\langle \log(g(\sigma_i)) \rangle} \tag{13}$$

before training. The overbar denotes the mean, and the angled brackets denote the standard deviation.

3.4.3. Network Skill and Hyperparameter Sweep

To train $\mathcal{N}_1$ two hyperparameters need to be tuned: the number of hidden layers and the number of nodes in those layers. For simplicity, we chose the same number of nodes in each hidden layer. A sweep was performed to test the accuracy of different networks. We varied the number of hidden layers from 2 to 4 and the number of hidden nodes in each layer from 2 to 512. Training data was randomized and provided as a single batch to train networks.

To measure the network's performance, linear correlation coefficient between the validation data and its prediction was calculated (see Figure 3a). The linear correlations for the 16 nodes were weight averaged with the mean value of $g(\sigma)$ at the corresponding node. The weight-averaged correlation is a better estimate of the network's skill for the given set of hidden nodes, as it reduces the influence of noisy values at the bottom of the boundary layer. The noisy values might be due to interpolation of the shape function profiles from the GOTM data. Based on hyperparameter sweep, we chose 2 hidden layers with 32 hidden nodes for $\mathcal{N}_1$, for which average correlation $\approx$0.9, and it is reasonably close to more expensive networks. For a deeper and wider network than 32 nodes, the average correlation score does not vary significantly, but the cost of using the network in an OGCM increases.

Figure 3b shows the loss curves for training the network. Training loss (magenta) and validation loss (green) decrease with the training epoch. The validation loss is higher than the training loss, but both eventually plateau.

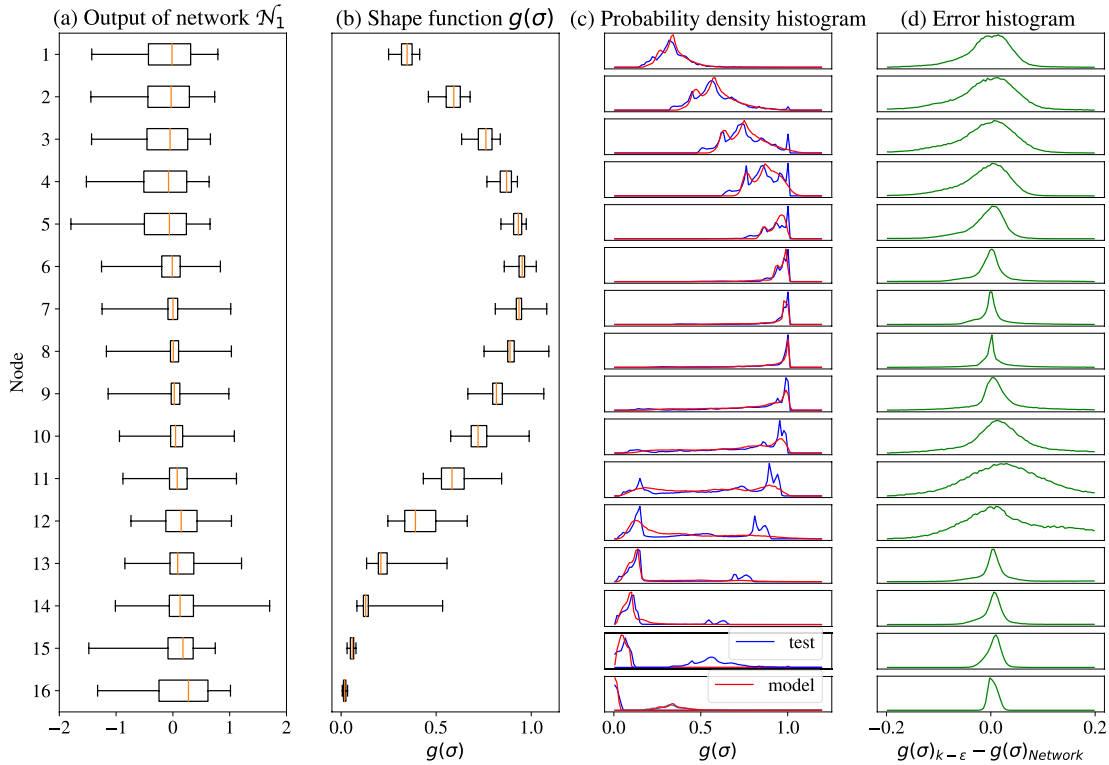

Figure 4. Performance of $\mathcal{N}_1$ for all the 16 sigma points. (a) Difference between network prediction and data. (b) Difference between network prediction and data in the physical space. Percentile ranges have been superimposed over the mean shape function from the training data set. (c) Probability density histogram between network prediction and data. (d) Histogram of error defined by differences between the network prediction and data. For nodes 10, 11, and 12 networks exhibits poor performance. This could be due to the strong multi-modal nature of data at those locations.

The difference between validation and training loss, shown in blue, remains constant in later epochs, signifying when training should be stopped. The validation loss does not increase, ensuring that the network is not overfitting the training data. The performance of the network is further tested by comparing it with the validation data. Strong agreement with validation data can be seen through the average correlation scores in Figure 3a and the error statistics in column (d) of Figure 4.

Figure 4 displays the performance of $\mathcal{N}_1$. The first column (a) shows the error statistics between the validation data and the network's prediction in the normalized space for each output node. The boxes show the interquantile range, while the whiskers show the 5th and 95th percentiles of the error. The second column (b) shows the same percentile range as in column (a) but in the physical space of $g(\sigma)$. The medians are superposed over the mean $g(\sigma)$ profile of the entire data set. This helps to visualize the skill of $\mathcal{N}_1$ with respect to each σ value. Nodes 11 and 12 have a high error variance compared to other nodes. The error variances in column (a) are different from those in column (b) because the data have different variances along the nodes. The last node 16 has a high variance in (a) but because the $g(\sigma)$ values at that node are small, poor performance at that node does not penalize the overall performance of $\mathcal{N}_1$. Node 16 is in the transition layer, which may have a large gradient of the tracer that might amplify the error in diffusivity at node 16. However, implementing this version of the network in ePBL yields an acceptable improvement in overall performance, suggesting that the error in node 16 is acceptable. Sensitivity in the transition layer will be investigated in more detail in future work. Column (c) has histogram plots of the validation data and its prediction. The network performs reasonably well and only shows inaccurate behavior when the data is multimodal. Column (d) shows the error histogram. The error has the highest variance at node 12, and is approximately Gaussian everywhere implying randomness.

For the neural network $\mathcal{N}_1$, Figure 4 shows the ability to predict the shape function offline. In general, the network shows high skill, as seen by the scores in Figure 3. The network $\mathcal{N}_1$ shows some inaccuracies in predicting multimodal distributions for output nodes 10–15. A single network predicts the value of the shape function at all the nodes, and it could compensate for the accuracy at one node over the other. Increasing the size of the network (i.e.,

number of layers and nodes in them) slightly reduces this error, but the cost of computation increases significantly with size rendering them unusable for longer time-scale simulations.

$\mathcal{N}_1$ is trained in *all* of the forcing regimes: surface heating, neutral, and convection. Perhaps, this adds a limitation to the network, which falls short of having very high skill for all the regimes. In our training experiments (not described in this article), training and predicting separately on the stable and unstable regimes gave higher skill than training on all regimes at once. Having two networks to predict $g(\sigma)$ alone could lead to higher skill without increasing the number of hyperparameters. This might be a cost-effective way to increase the overall accuracy of ePBL_NN without expanding the size of the network. Increasing the number of hidden nodes in the hidden layers increases the cost of forward computation, while switching between the networks based on the forcing regime has a similar cost to using a single network. For simplicity, in this work we prefer to train all data using a single neural network and have not pursued this any further.

We used the L1 loss function (mean absolute error) for training, as it gave better training performance than the L2 (root mean square error [RMSE]). We also increased the convergence of the network parameters (weight and biases) by tweaking the loss function. The loss values at nodes 8 to 13 were amplified by a factor of 100. This made the loss gradients steeper at the nodes that show the highest variance (seen in columns 1 and 2). This forces the network to put more weight on reducing errors on the nodes that are otherwise difficult to learn. The ADAM optimization algorithm (Kingma & Ba, 2014) has been used to train the weights and biases of the network with a learning rate of 10^{-3}.

3.5. Training $\mathcal{N}_2$

The second neural network $\mathcal{N}_2$ as shown in Figure 2 predicts the characteristic velocity, v_0. Velocity is diagnosed from the training data using the following *jugaad*:

$$v_0 = \overline{\left(\frac{\max(\kappa_{k\epsilon}(\sigma))}{h}\right)} \tag{14}$$

where the overbar denotes the average of all the values of v_0 for a set (f, B_0, u_*). The spread of $\max(\kappa_{k\epsilon}(\sigma))/h$ for a constant (f, B_0, u_*) is small, and averaging assists the neural network in training to predict the mean value (see Appendix A).

Similarly to $\mathcal{N}_1$, network is trained on logarithm of v_0 and exponential function is used while inferring to ensure that the predicted v_0 is strictly positive. The data is divided into 80%–20% to train and test the performance of the network. As seen in Table 1, the training data cover a wide range compared to that of $\mathcal{N}_1$, including extreme forcing conditions anticipated in a realistic OGCM. When the network sees conditions outside this range, the input is capped at the nearest extremum data point. This is to prevent the network from extrapolating, which is less skillful than interpolation. The trained network has high skill (linear correlation of 0.99) as seen in Figure 5.

4. Evaluating Impacts in a Prognostic OGCM

Training, testing, and validation data provide one method for testing the network and its ability to reproduce training data. However, to fully test the network's potential for OGCM experiments the neural networks must also be implemented in free-running, prognostic models. Our implementation does not cause simulation to fail due to any spurious effects or instabilities which is a known problem with implementing neural networks in a GCM (see Brenowitz et al., 2020 and references therein). Stability might result because we implement neural networks as a component within the existing ePBL framework. We demonstrate the success of our implementation using both free running column model experiments and forced ice-ocean global OGCM climate model experiments.

4.1. Implementation of Neural Networks in MOM6

We now describe the implementation of our networks in the MOM6 ocean model. The weights and biases of the network are generated offline and stored in NetCDF files. Feedforward (inference) calculation of the network involves matrix multiplications and activation functions. These have been coded as subroutines in MOM6's vertical mixing module (ePBL). A flag activates the neural networks to predict $g(\sigma)$ and v_0. All inputs to the network

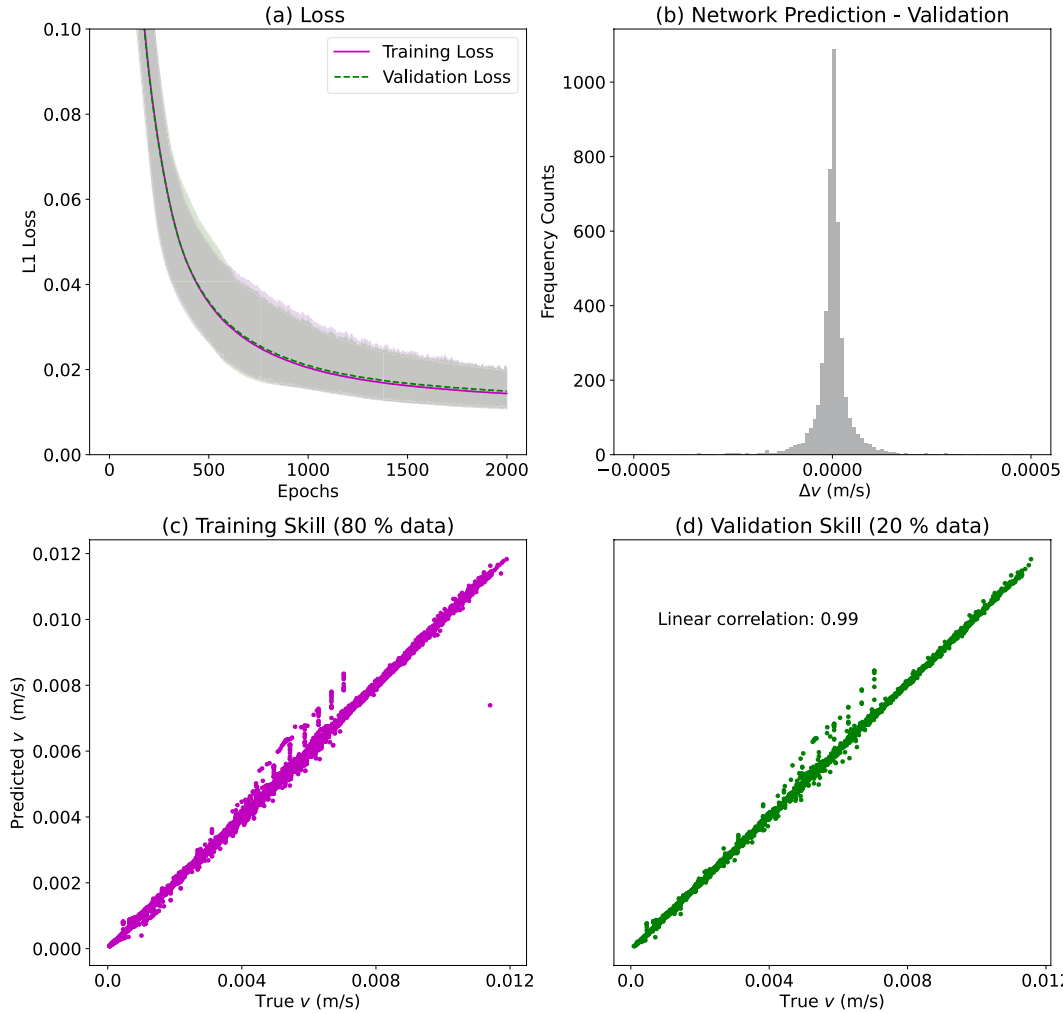

Figure 5. Performance of $\mathcal{N}_2$. (a) Loss curves. (b) Histogram of difference between network's prediction and data. (c) Predicted versus true values for the training data set. (d) Predicted versus true values for the validation data set.

are readily available within the ePBL module. The neural networks require the depth of the boundary layer h, which is provided by the ePBL scheme as described in Reichl and Hallberg (2018). The neural networks function alongside the algorithm by which ePBL derives h and therefore they do not interfere with any energy constraints set by the original scheme. Additionally, in MOM6, the diffusivity derived from ePBL and the neural network subroutines is passed to a main diabatic mixing module which combines diffusivities from various mixing parameterizations (such as Jackson et al., 2008) within MOM6. More details can be found in Reichl and Hallberg (2018). $g(\sigma)$ is obtained at 16 points between the surface and h. At the surface, $g(\sigma = 0) = 0$ to satisfy zero diffusivity. At h, $g(\sigma = 1)$ is set as a small number by assuming $g(\sigma = 1) = c \cdot g(\sigma = 16/17)$, where c is a small positive constant set as 0.1. GCM and single column runs were found to be insensitive to small and non-zero values of c.

Shape function on σ is converted to the model's vertical grid by linear interpolation. The use of the sigma coordinate makes our scheme grid independent of the vertical coordinate. The shape function on the model grid is multiplied by $v_0 \cdot h$ according to Equation 10 to recover the diffusivity profile of the $k - \epsilon$ model. The subroutines pass on the diffusivity profile to the ePBL module. In MOM6, there are other parameterizations active along with ePBL to incorporate strong shearing regions found at the equator and also that handle background diffusivity. Both networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are shallow, as they have 2 hidden layers with 32 nodes in each. OM4 model with ePBL_NN requires ≈5%–10% more runtime than ePBL. This cost may not warrant a need for graphical processing units (GPUs) to speed-up the inference in this version of the scheme, but this option will be explored in the future.

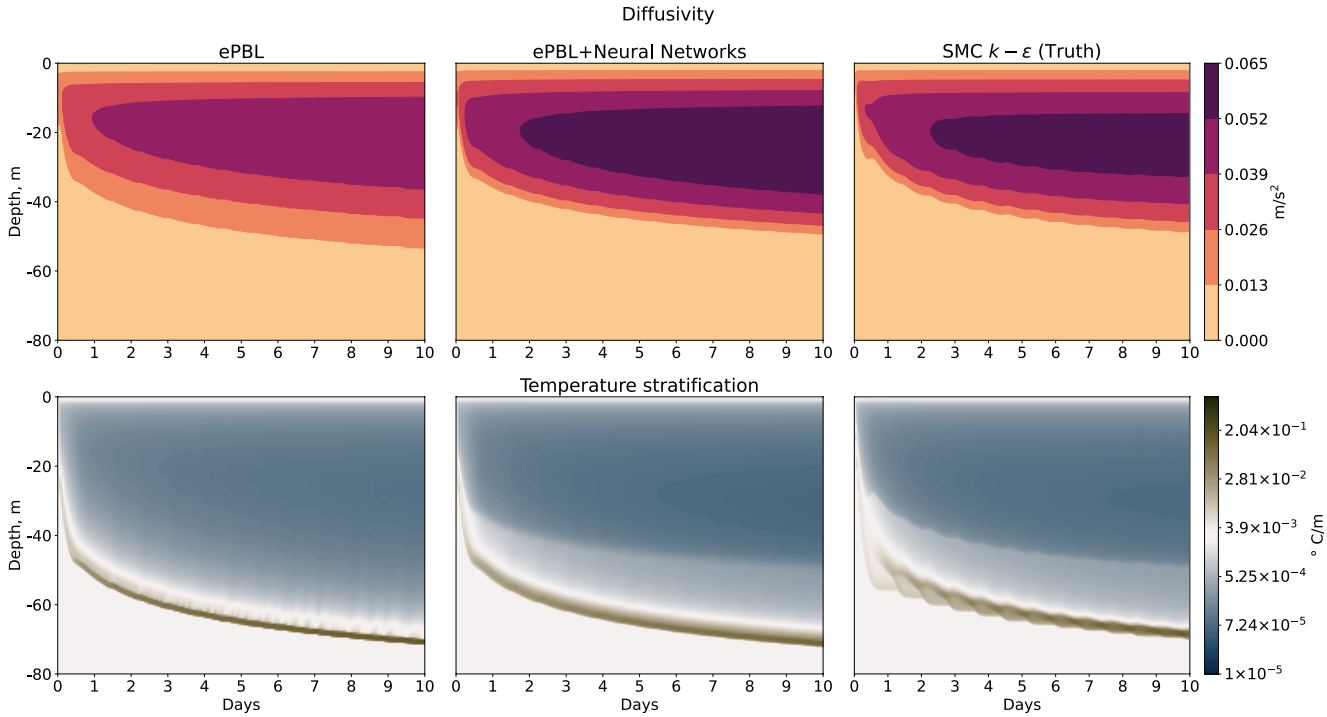

Figure 6. Time series comparison for single column model configuration. Latitude is set to 40°, surface heat flux is 50 W/m^2, and wind stress is 0.2 N/m^2. The upper row compares diffusivity and the bottom row compares stratification. In both the cases, ePBL_NN is in better agreement to the second moment closure scheme $k-\epsilon$ than energetic Planetary Boundary Layer (ePBL).

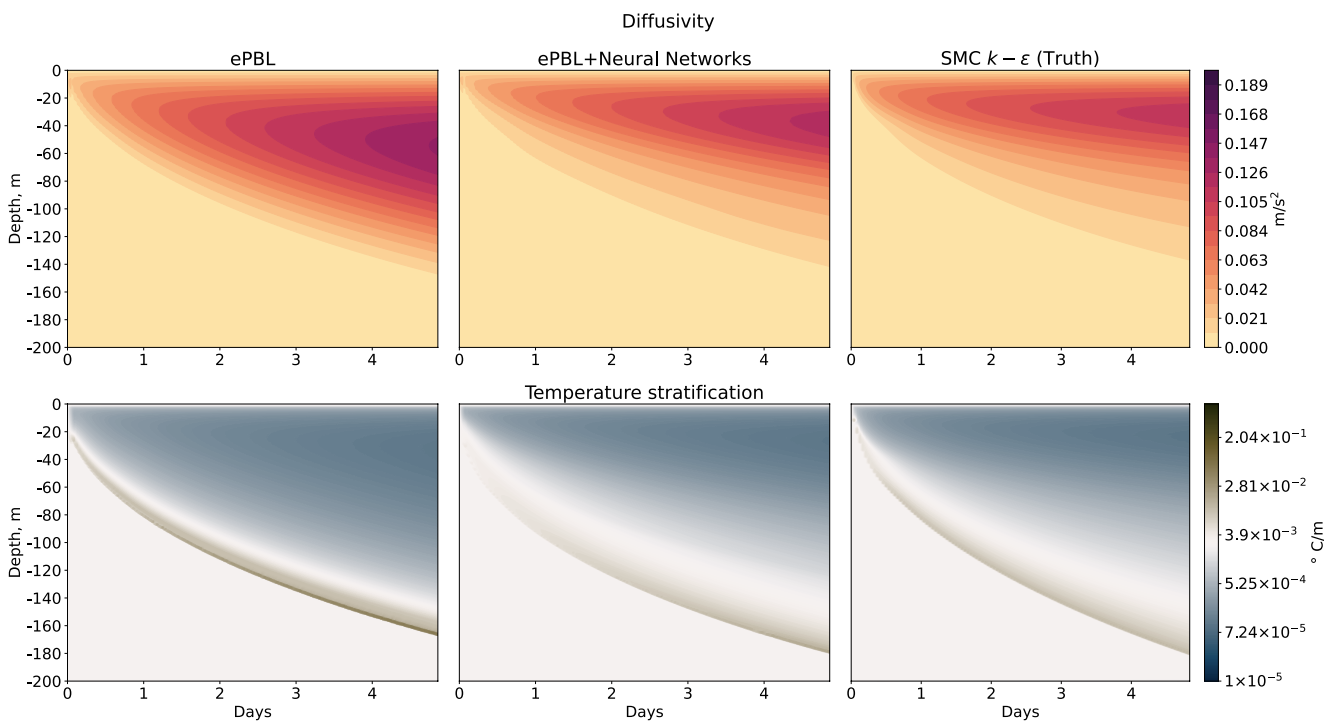

Figure 7. Time series comparison for single column model configuration. Latitude is set to 1°, surface heat flux is 50 W/m^2, and wind stress is 0.2 N/m^2. The upper row compares diffusivity and the bottom row compares stratification. In both the cases, ePBL_NN is in better agreement to the second moment closure scheme $k-\epsilon$ than energetic Planetary Boundary Layer (ePBL).

The inputs to the neural network are also capped inside the subroutine to ensure the networks do not make predictions outside their training range. For $\mathcal{N}_1$, if any of the inputs (f, B_0, u_*, h) are outside the known range, then the subroutine limits the inputs and changes them to the nearest point in the four-dimensional hypercube formed by the four inputs. Our training data covers a reasonable space of the forcing parameter regime as observed among realistic present conditions (as it will be applied in this study). Data points outside the range are less probable, allowing the network to perform effectively for nearly all of the tested forcing conditions (see Appendix B). Capping the inputs prevents the network's output from being unphysical. If the network is applied for simulations in substantially different climate regimes (e.g., paleoclimate or for other planetary bodies) the training data could be enhanced. If the network receives inputs outside the known range, the shape function can have spurious values with irregular vertical structure. Capping the inputs ensures that this spurious behavior is prevented. The training on logarithm and using exponential function while inferring described in the earlier sections prevents non-positive behavior for both $\mathcal{N}_1$ and $\mathcal{N}_2$.

4.2. Single Column Model Results

We compare three schemes to examine the performance of the network in single column model: GOTM $k-\epsilon$, ePBL, and ePBL_NN. MOM6 in single column configuration (as in Reichl & Hallberg, 2018) is used to run ePBL and ePBL_NN, while GOTM is used for the $k-\epsilon$ experiments. The column models are forced at the surface grid interface with constant buoyancy forcing (surface heating of 50 W/m^2) and constant wind surface stress (0.2 N/m^2). Stratification is constant throughout the column in the initial conditions. To have the same entrainment in all the three cases, the m* value is diagnosed from the $k-\epsilon$ output and imposed in MOM6. The quantity m* is the non-dimensional integral of the entrainment flux and is given by $\int_{-h}^{0}\min\left(0,\overline{w'b'}\right)dz = m^* u^{*3}$ for surface heating conditions. In Reichl and Hallberg (2018), m* has been parameterized using a function G as in Equation 5. Instead of using the parameterized m* from Reichl and Hallberg (2018) and Reichl and Li (2019), we use a diagnosed and time varying m* from $k-\epsilon$ to perform a controlled comparison with identical forcing conditions. This prevents deficiencies in the parameterized m* from causing any disagreements between MOM6 and GOTM. By matching the surface forcing and integral of the entrainment flux, the differences between all the three cases can only be due to diffusivities.

Two latitudes are compared: Latitude 40° (Figure 6) and 1° (Figure 7). The figures show the time series of diffusivity and temperature stratification. For both latitudes, the diffusivity and stratification in ePBL_NN are in closer agreement with the $k-\epsilon$ model than the original ePBL model, showing the ability of the neural networks to match $k-\epsilon$. ePBL_NN has a diffusivity profile closer to $k-\epsilon$ than ePBL throughout the OSBL. In $k-\epsilon$ (SG), the turbulent diffusivity is computed from the simulated turbulent kinetic energy and turbulent length scale, using stability functions that relate the Prandtl number to the Richardson number (Schumann & Gerz, 1995). The neural networks have "learned" those relationships (without direct knowledge of either parameter) that set the structure of diffusivity and hence show high skill in predicting the profile.

The upper ≈20% of the diffusivity profile is able to learn traditional constraints, such as the law of the wall scaling, since they are features of the training data. The bottom ≈40% of the OSBL shows more variability and is an important region for the entrainment process. In deepening of the boundary layer, the entrainment process mixes the higher density water masses (usually cold) from below the mixed layer with the lower density mixed layer above it (usually warmer). Outside of the polar regions, this process cools the mixed layer along with the sea surface temperature (and warms the interior) and has implications for ocean-atmosphere energy exchange and feedbacks.

4.3. Ice-Ocean JRA Forced Model Results

We next tested the ePBL_NN scheme using the GFDL's OM4.0 ocean/sea ice model. The model has a nominal 1/4° resolution and is forced using the JRA forcing product (Tsujino et al., 2018). JRA forced simulations constrain the atmospheric fields that force the ocean model with the observed/reanalysis atmospheric data. This is different from the atmosphere-ocean coupled model as there is no feedback from the ocean response to the atmosphere. However, this approach is beneficial for testing parameterizations since two experiments can be more carefully compared without considering the complications of those feedbacks. Future work will examine the performance of these schemes in fully coupled climate models.

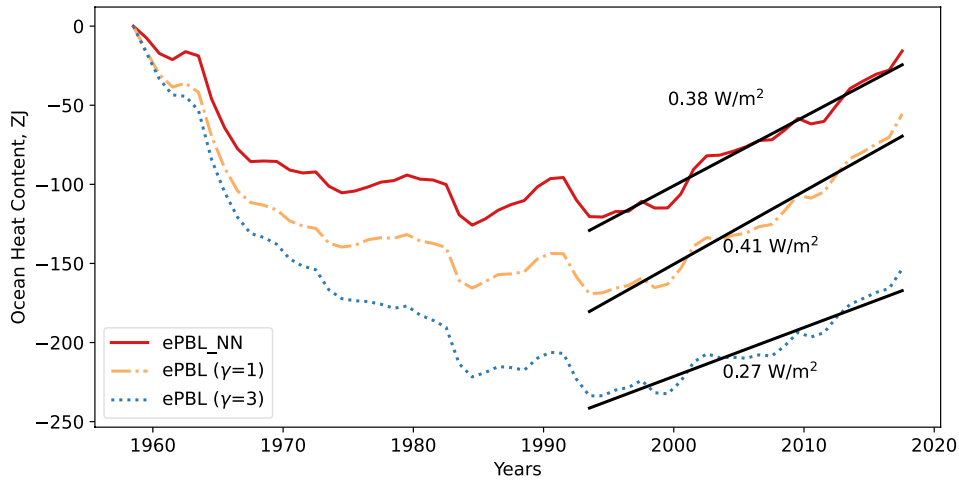

Figure 8. Total ocean heat content compared between energetic Planetary Boundary Layer (ePBL) and ePBL_NN for the duration of 1958–2017. Initial 30 years (1958–1988) can be considered as spin up. For ePBL, γ from Equation 8 has been set to 1 and 3. For ePBL_NN, the tunable parameter γ does not exist. We can observe that the ocean's total heat content is sensitive to the vertical diffusivity set by the ocean surface boundary layer mixing scheme. ePBL_NN replaces ad-hoc diffusivity of ePBL with a physics informed data-driven neural network.

Two sets of OGCM experiments have been performed: one using the ePBL scheme as a control run (e.g., as described by Adcroft et al. (2019)) and the second with the neural networks active to replace the shape function and velocity scale in ePBL. The simulations were performed for a period of 1958–2017.

In this study, we compare the two runs with observations to analyze the impact on: (a) Ocean heat uptake, (b) Sea surface temperature, (c) Mixed layer depth, and (d) Upper ocean temperature stratification in the Tropical Pacific. For sea surface temperature, data from the World Ocean Atlas (Boyer et al., 2018; Locarnini et al., 2019) has been used to compare the two schemes. For the subsurface comparison: mixed layer depth and stratification, ARGO float measurements have been utilized (Argo, 2023).

4.3.1. Ocean Heat Uptake and Sea Surface Temperature Comparison

Figure 8 shows the global ocean heat content for the three runs: one with ePBL_NN shown in red-solid line, and the other two with ePBL by setting γ from Equation 8 as 1 and 3 shown as blue-dashed line and green-dotted line respectively. ePBL_NN shows more heat uptake than the original scheme, and rate of warming is between ePBL runs with γ set as 1 or 3. This highlights the sensitivity of the total ocean heat content to the shape function and to boundary layer mixing schemes.

Figure 9 shows the sea surface temperature (SST) bias averaged over the years 2003–2017 for each 1° grid point. SST biases are similar in the two runs with minimal differences, which is expected since the atmospheric fields are prescribed and not coupled. SST around the eastern Pacific and Atlantic equatorial regions shows a slightly warmer bias for the ePBL_NN run than for the ePBL. In the Indian Ocean, the bias is slightly colder. The SST bias in the Gulf Stream and Kuroshio current is slightly warmer in ePBL_NN by about 0.5°C. The response of the SST to ePBL_NN in the boundary current regions indicates that changes in the vertical viscosity or diffusivity also impacts the circulation in certain regions.

Changes in the patterns of SST can be due to changes in the mixed layer depth and the surface heat fluxes. The heat fluxes are computed as a function of SST, surface ocean velocity and ice cover as stated in Adcroft et al. (2019) and Griffies et al. (2016).

4.3.2. Mixed Layer Depth Comparison

Summer and winter mixed layer depths (MLD) are compared, a metric usually used to indicate the depth at which atmospheric influences are directly felt in the ocean. Here, winter (summer) mixed layer depth is the maximum (minimum) of the monthly averaged MLDs for each grid point over the period 2003–2017. The MLD depends on the definition, and we evaluate it using two criteria: Reichl et al. (2022) and de Boyer Montégut et al. (2004). The criterion from de Boyer Montégut et al. (2004) uses a threshold potential density of 0.03 kg/m^3 whereas Reichl

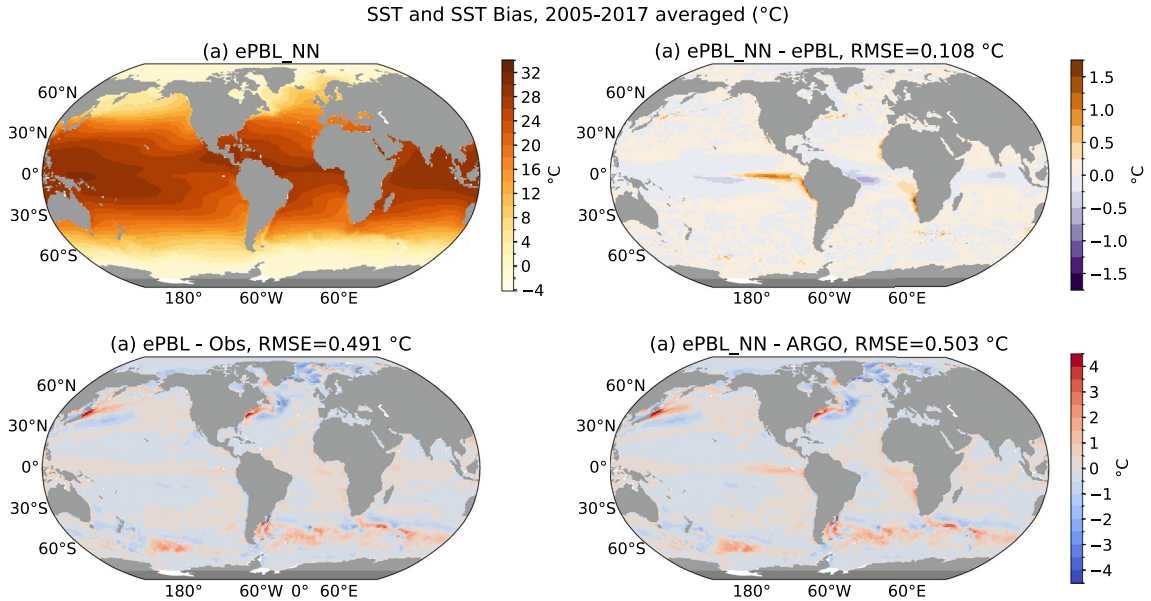

Figure 9. Sea surface temperature and biases. (a) Sea surface temperature (SST) from model runs using ePBL_NN. (b) SST difference between energetic Planetary Boundary Layer (ePBL) and ePBL_NN. (c) SST bias between ePBL and observations (Obs). (d) SST bias between ePBL_NN and observations. Bias plots also show mean and standard deviation. Observations are from World Ocean Atlas data set (Locarnini et al., 2019). SST biases are similar for ePBL and ePBL_NN. At the equatorial region, ePBL_NN shows slightly colder bias than ePBL.

et al. (2022) uses a threshold potential energy anamoly of 25 J/m^2 to define the MLD. Figures 10 and 11 show the MLD using the potential energy anomaly criterion and the potential density respectively.

Figures 10 and 11 show summer time MLD. The summer time MLD bias has reduced significantly in ePBL_NN as compared to ePBL. The average bias reduced from 7.22 to 5.73 m as seen in Figure 11. Between −20° and 20° latitude, the average RMSE for MLD bias in ePBL was about 7.94 m. In ePBL_NN, the bias was reduced to 5.18 m. We have shown the latitude dependency of RMSE between model and observations in Supporting

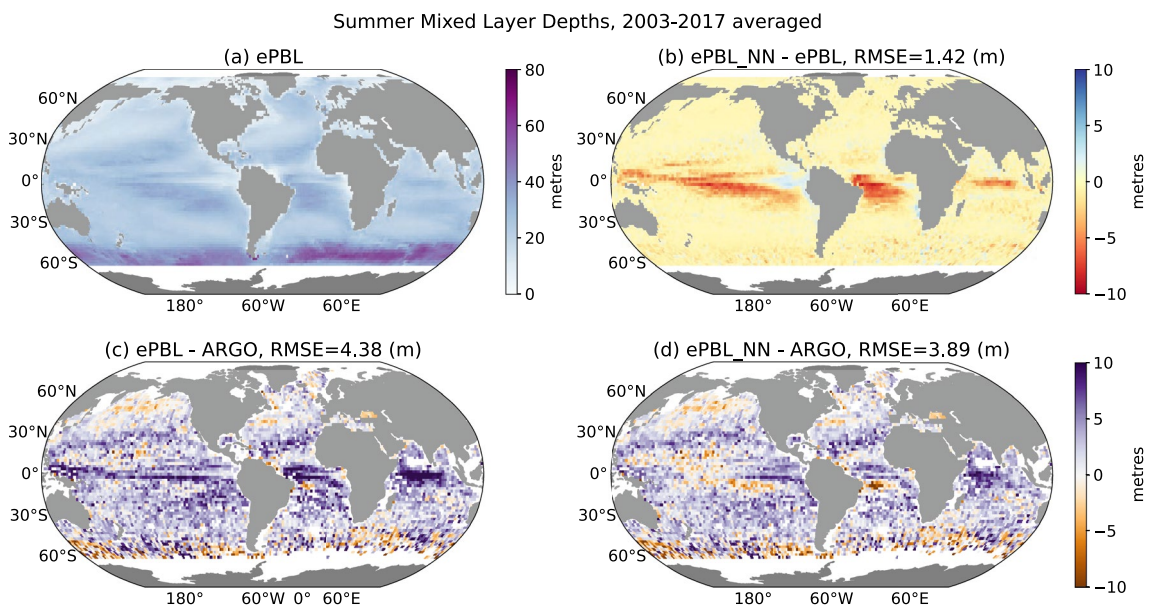

Figure 10. Summer time (shallow) mixed layer depth biases using the Potential anomaly criterion of Reichl et al. (2022). (a) Mixed layer depths (MLD) from energetic Planetary Boundary Layer (ePBL), (b) Difference of MLD between ePBL and ePBL_NN. (c) Bias of ePBL with respect to ARGO data, (d) Bias of ePBL_NN with respect to ARGO data. We can notice the bias reduction from (c) to (d).

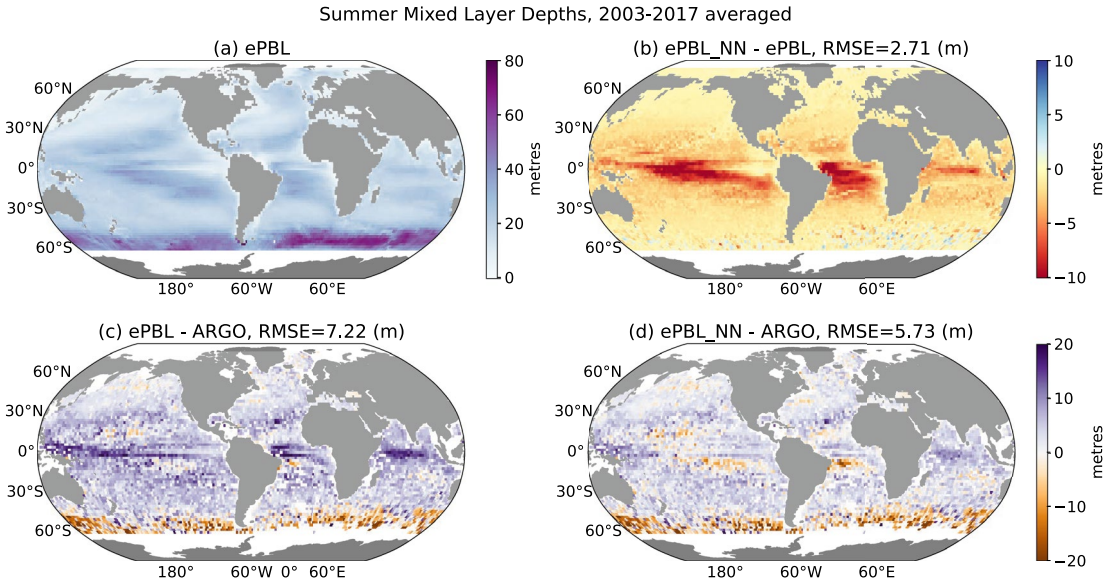

Figure 11. Summer time (shallow) mixed layer depth biases using the density criterion of de Boyer Montégut et al. (2004). (a) Mixed layer depths (MLD) from energetic Planetary Boundary Layer (ePBL), (b) Difference of MLD between ePBL and ePBL_NN. (c) Bias of ePBL with respect to ARGO data, (d) Bias of ePBL_NN with respect to ARGO data. We can notice the bias reduction from (c) to (d) and is consistent with that observed in Figure 10.

Information S1 (see Figures S1 and S2 in Supporting Information S1). The ePBL_NN scheme performs better under stable surface heating conditions than the ePBL scheme. The shallow MLD bias reduction has implications for equatorial oceanic regions and its effect on large-scale ocean-atmosphere feedbacks (Adcroft et al., 2019). Winter MLD biases (Figures 12 and 13) are very similar for both runs. The ePBL_NN predicts diffusivity close to a second moment scheme but does not significantly impact the winter time bias simulated by the model with the original ePBL scheme. This is likely because other model physics and factors can dominate in setting the deep convective mixed layers and water properties.

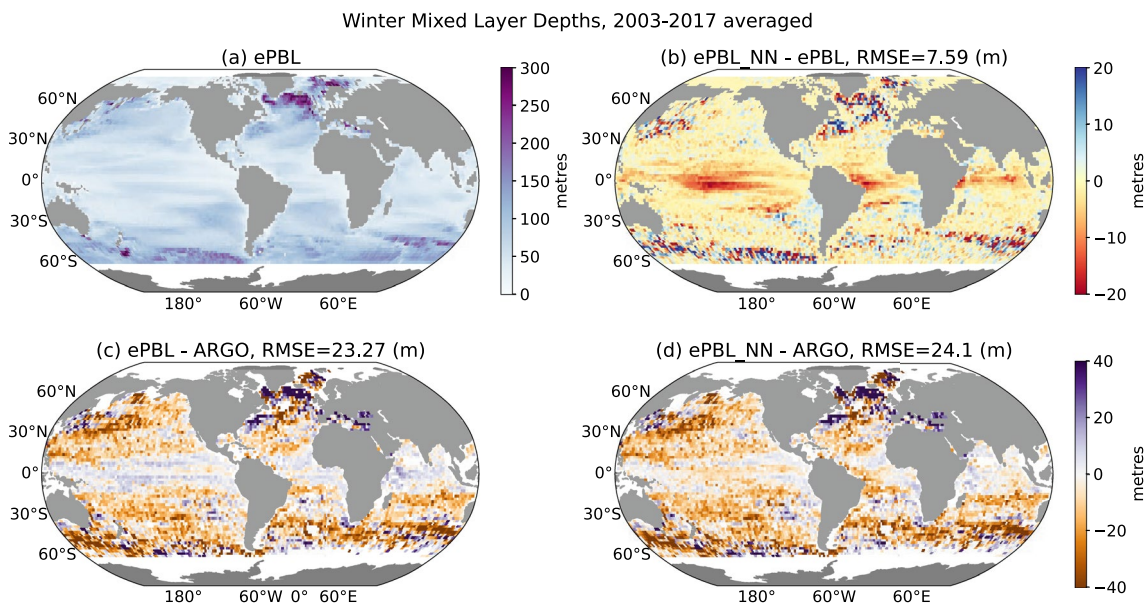

Figure 12. Winter time (deep) mixed layer depth biases using the Potential anomaly criterion of Reichl et al. (2022). (a) Mixed layer depths (MLD) from energetic Planetary Boundary Layer (ePBL), (b) Difference of MLD between ePBL and ePBL_NN. (c) Bias of ePBL with respect to ARGO data, (d) Bias of ePBL_NN with respect to ARGO data. Biases are similar in (c) and (d) and ePBL_NN does not significantly worsen any biases.

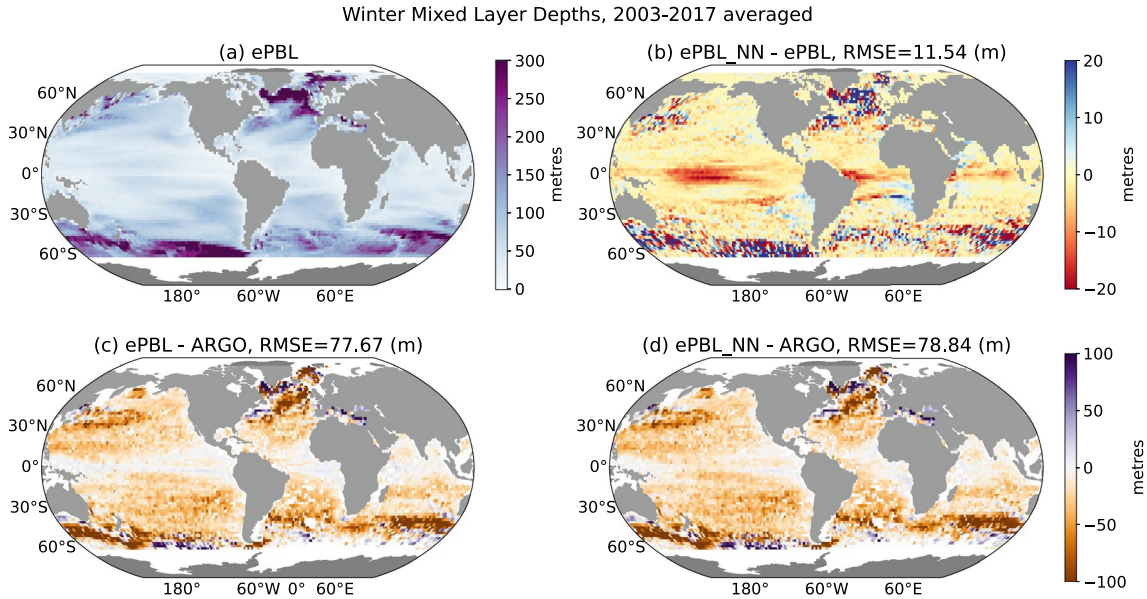

Figure 13. Winter time (deep) mixed layer depth biases using the density criterion of de Boyer Montégut et al. (2004). (a) Mixed layer depths (MLD) from energetic Planetary Boundary Layer (ePBL), (b) Difference of MLD between ePBL and ePBL_NN. (c) Bias of ePBL with respect to ARGO data, (d) Bias of ePBL_NN with respect to ARGO data. Biases are similar in (c) and (d) and ePBL_NN does not significantly worsen any biases. This observation is consistent with that observed in Figure 12.

Although ePBL_NN has been trained on all the regimes including surface cooling, a different scheme or process might be compensating the effects of improved diffusivity. This could also be due to higher sensitivity of shallow mixed layers to changes in surface forcing than deep mixed layers. For shallow mixed layer depth, any perturbations in the atmospheric forcing will reach the base of the boundary layer quicker than it would reach in deeper layers. In Reichl and Hallberg (2018), the rate of conversion of turbulent kinetic energy to potential energy within the boundary layer (left hand side in Equation 7) uses a parameterization that depends on h. Changing the diffusivity can alter h which in turn modulates the rate of energy conversion. This can lead to changes in the MLD.

The winter time MLD biases are similar for both runs. The summer time MLD bias shows a further reduction when evaluated using de Boyer Montégut et al. (2004) than with Reichl et al. (2022). It is not unusual to get different values of MLD using different definitions. For both definitions, winter biases in ePBL_NN are not worsened. Qualitative agreement of the reduction in summer bias in ePBL_NN using two different criteria provides strong evidence of ePBL_NN performing better than ePBL in terms of MLD bias reduction under fixed atmospheric forcing conditions.

4.3.3. Comparison of Upper Ocean Stratification in the Tropical Pacific

The final comparison we use to assess the impact of the neural network diffusivities on the model result is the upper ocean temperature stratification ($\partial\Theta/\partial z$, where Θ is the potential temperature) in the Equatorial Pacific region. The thermocline in the Equatorial Pacific region plays an important role in El Niño Southern Oscillation dynamics with implications for the Earth's climate system (for example) (Jin & An, 1999). The temperature stratification is shown for a vertical cross section along the equator spanning −220° to −80°E. Figure 14c shows the observational data from ARGO floats (Roemmich & Gilson, 2009). Figure 14b is the $\partial\Theta/\partial z$ from the original ePBL and shows lower stratification as compared to ARGO observations. The $\partial\Theta/\partial z$ from ePBL_NN, as seen in Figure 14a, shows significant improvements in the stratification of the upper 50 m of the ocean. Stratification in ePBL_NN is closer to ARGO data in the equatorial region of the Pacific Ocean. The neural network predicted diffusivities help to increase the stratification and make it closer to observations than the simulation with the original ePBL with the ad hoc shape function for diffusivity.

Stratification acts as a barrier to mixing, this warrants further investigation into how ePBL_NN changes transport pathways of heat through the OSBL in different regions of the world's oceans. Overall, the MLD bias is reduced,

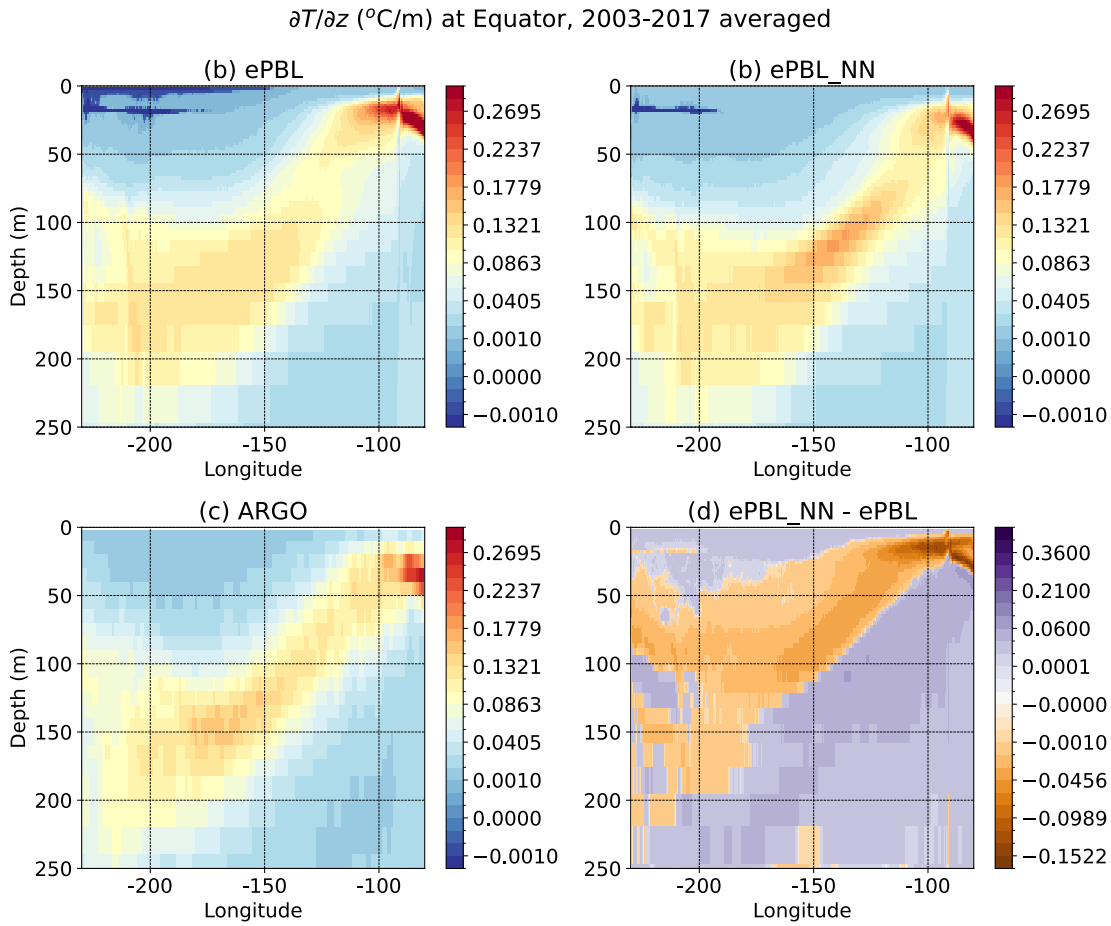

Figure 14. Temperature stratification at a transect along the equator in the Pacific Ocean. (a) Energetic Planetary Boundary Layer (ePBL) output, control run. (b) ePBL_NN output. (c) ARGO data (d) ePBL—ePBL_NN. Note that (b) is closer to (c) than (a). ePBL_NN has been instrumental in enhancing the stratification of the upper ocean.

and stratification has improved for the upper ≈50 m suggesting that ePBL_NN works to fix these two biases in conjunction.

5. Concluding Remarks

5.1. Summary

In this study, we apply neural networks to improve the parameterization of the vertical diffusivity in the OSBL. The data used to train the neural networks is obtained using second-moment closure simulations by running single-column model under various forcing scenarios, spanning the possible range of present-day and future conditions. The neural networks are implemented within the existing physics-based parameterization from the ePBL framework of Reichl and Hallberg (2018). The neural networks augment the method to determine the vertical diffusivity in the ePBL scheme with data-driven relations but maintain the physically motivated energetic constraints on mixing from the original scheme. A benefit of our approach is that it yields a stable implementation in the OGCM (MOM6). This enables us to perform decade-scale simulations spanning 1958 to 2017.

Atmospherically forced Ice-Ocean experiments using the GFDL's MOM6 1/4° model suggest an overall improved performance due to the enhancements in ePBL_NN relative to the original scheme. There is a reduction in biases of summer-time MLD and no exacerbation of the winter-time biases compared to ePBL. The stratification of the upper ocean in the tropical pacific shows improvements in the thermocline

compared to the ARGO float observations. This analysis indicates that the resulting scheme is suitable for implementation in future OGCM configurations and experiments and is expected to reduce biases in climate simulations. Further analysis using a wider range of diagnostics in additional model configurations will be particularly beneficial.

The ePBL framework is already optimized for GCMs, providing larger time stepping capabilities ($\approx O(1)$ hr) and ePBL_NN leverages these advances with improved diffusivity profiles. It is computationally expensive to run a second-moment scheme in a GCM due to time-stepping restrictions ($\approx O(10-100)$ s), but ePBL_NN can yield eddy diffusivity profiles more consistent with a SMC within the original ePBL framework. This is significant for GCMs as we are achieving closer results to a model having a second-order turbulence closure scheme, but able to maintain coarse resolutions and long time steps needed for climate scale simulations. We note that the longer implicit time step used in the numerics of ePBL (see Reichl & Hallberg, 2018) can lead to a smoothing effect which can complicate resolving small-scale structure, but we observe that the large-scale evolution is tracked accurately.

While the results of this work are promising, numerous aspects remain important for future work. For example:

1. ePBL, ePBL_NN, and the SMC considered here assume downgradient diffusion and hence have no nonlocal flux terms. The representation of nonlocal fluxes could improve the scheme further and potentially affect convective regions and Langmuir turbulence (e.g., Chor et al., 2021). In this application we do not explicitly consider the impact of Langmuir turbulence within ePBL (though it is part of setting the energy available for entrainment, see Reichl & Li, 2019).
2. The neural networks can be made larger to capture more complex relationships in the data by increasing the number of hyper-parameters (hidden nodes). In this work we chose small networks for initial investigation. The successful use of small neural networks as efficient surrogate models of SMCs proves that we can replicate the behavior of complex models with high fidelity. Increasing the network size will be explored in the future and will likely require using GPUs for implementation in OGCM (Zhang et al., 2023).
3. The performance of the modified vertical mixing scheme in a coupled model (atmosphere-ocean-ice) may not show the same impact on model bias as observed in this forced ocean-ice model. The atmosphere-ocean feedbacks will require exploration in future work.
4. Improving the representation of the diffusivity profile has implications for many quantities that have gradients within the boundary layer. For example, changing the diffusivity of nutrients within the euphotic layer has implications for biogeochemical processes such as primary production. The implications of improved diffusivity for ecological modeling will be explored in future work.
5. Finally, we have trained on one SMC, the $k-\epsilon$ model with stability functions following Schumann and Gerz (1995). This parameterization was chosen for consistency with Reichl and Hallberg (2018), but alternative SMC models may yield different results. In future work this process will be repeated with different SMC schemes to understand the influence of SMC diffusivities on the performance of GCMs. One disadvantage is that SMCs have been assumed to be the "truth" but it might lack realism and hence future work will focus on including data from LES studies and observations.

5.2. Applications for First Order Ocean Surface Boundary Layer Parameterizations

One key achievement of this work is that it establishes a relationship between the shape function of upper ocean vertical mixing and the forcing parameters. Previous work in similar first-order upper ocean mixing parameterizations assumes that the shape function is fixed, or was set by ad-hoc approximations. This work further suggests that models that consider this variation in the shape function are more skillful at simulating upper ocean stratification and ocean mixed layers. The physics-informed function (network) developed in this work for determining the shape function from the forcing parameters is applied here in ePBL as an example. However, the function is not specific to ePBL and can also be used within other first order OSBL parameterizations (such as KPP, Large et al., 1994; Van Roekel et al., 2018).

It is also important to consider that the neural network based model used in this work is not the only approach to find a relationship between the forcing terms and the vertical mixing profile. The neural network is able to establish the existence of a relationship between its input and outputs, which is learned during the training process. While the neural network can be applied in ocean models as-is to improve simulations, we also desire

an in-depth understanding of the patterns in the inputs that the network used to make its skillful predictions. In future work, we seek to relate the network's findings to the processes that govern the OSBL's behavior (e.g., with equation discovery). This may ultimately lead to a simpler, interpretable and computationally low-cost physics based model for the shape function that can be learned from the neural network and applied in ocean models.

5.3. Implications for Augmenting Ocean Parameterizations With Machine Learning

A second implication of this work is demonstrating the potential for neural networks to improve parameterizations in ocean models. This implication is in agreement with several similar previous studies in earth system modeling (e.g., O'Gorman & Dwyer, 2018; Yuval & O'Gorman, 2020). As neural networks are not limited to individual processes, future avenues of research on ocean parameterizations will benefit from their usage. For example, neural networks can be applied to incorporate different mixed layer processes such as non-local fluxes during convection, entrainment, Langmuir turbulence, symmetric instability, surface wave effects, etc. into a single neural network model. Further improvements can be made which incorporate time history to improve predictions under transient forcings. Many existing ocean/atmosphere parameterizations have a physics based parent scheme with a few ad-hoc components or approximations. These components can be replaced or re-tuned using our approach or other emerging approaches such as Ensemble Kalman methods, posteriori criteria matching, etc. (e.g., Frezat et al., 2022; Lopez-Gomez et al., 2022, and references therein). Parameterizations in the form of weights and biases are advantageous because they can be re-tuned and further optimized to train as additional data, observations, and processes are presented.

The successful application of neural networks in an OGCM simulation unlocks the potential to test the importance of improving a certain process/parameterization in the model. For example, consider a case where the process studies' data exist, but a physics-based parameterization might be challenging to develop. Neural networks can parameterize that process and its impacts in an OGCM can be explored before going into a detailed parameterization development, which can be resource-consuming.

One of the major sources of uncertainty in climate models arise from parameterizations due to their inadequate representation of sub-grid physics. Perhaps, high resolution or shorter time-steps can attenuate the effects of structural uncertainties in sub-grid parameterizations. Computational limitations often impose constraints on factors such as resolution, ensemble size, and integration time scales within models. These limitations underscore the need for improving the current generation of climate models, while steering away from relying on higher resolution models or shorter time steps. Combining traditional process-oriented studies with the emerging field of machine learning offers the potential for synergistic advancements, leading to the refinement of sub-grid models. We have established a pipeline whereby an existing parameterization is augmented to harness the capabilities of neural networks. The successful integration of neural network within the ePBL, and its application to an ocean model, introduces opportunities for enhancing parameterizations that govern upper ocean mixing in climate models.

Appendix A: Why Does v_0 Change Due To Coriolis Parameter, f?

In general, turbulent velocity scales are related to turbulent kinetic energy and depend on boundary forcing, u_* and B_0. Here, in addition to u_* and B_0, we find a dependency of the bulk turbulent velocity scale v_0 on f. The bulk velocity is diagnosed by using diffusivity and boundary layer depth from the training data as per Equation 14. To predict v_0 using $\mathcal{N}_2$, the Coriolis parameter f has been used because we found the model improves in its ability to predict variations in v_0 in the training data. This is evident from Figure A1.

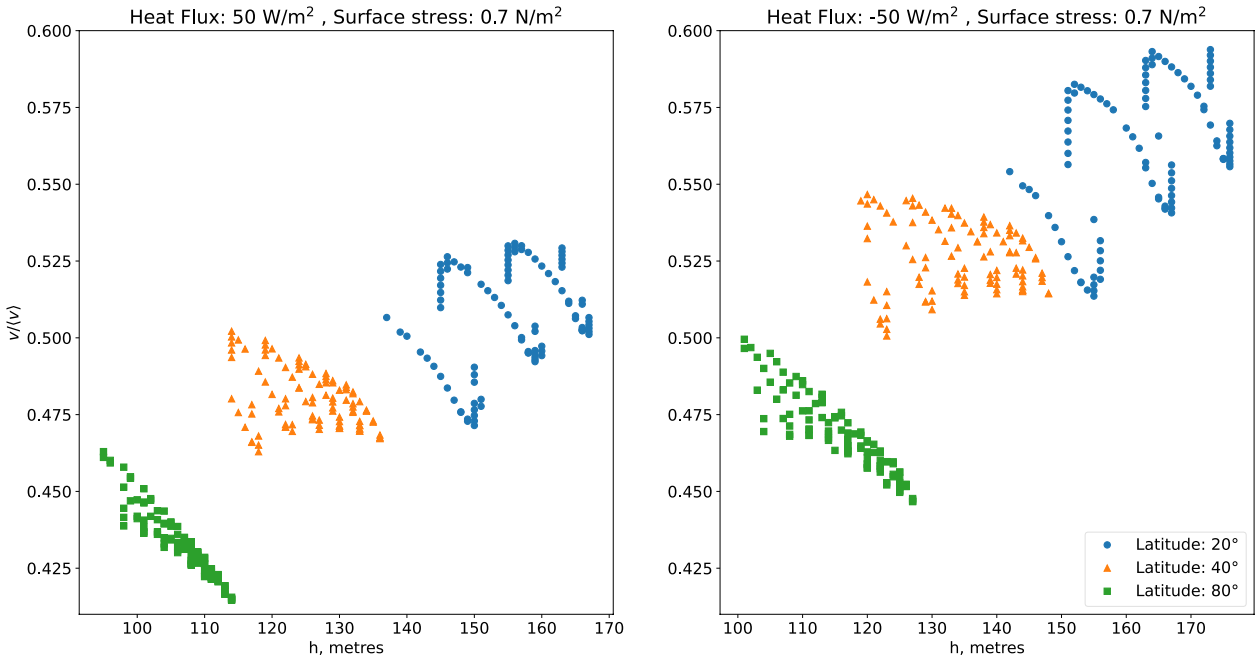

Figure A1. Variation of normalized v with respect to its mean value. v_0 varies due to heat flux, surface stress, latitude. Variations due to h are within 5% of the mean value of v_0 and hence it is reasonable to exclude h from being an input to $\mathcal{N}_2$. Note that v_0 is a diagnosed quantity from the output of $k-\epsilon$ solely used to reconstruct the diffusivity profile.

Figure A1 shows the variation of v_0 with respect to latitudes and h under surface heating and cooling conditions. This indicates that f is a useful input for accurately predicting v_0. Physically, the inclusion of f is related to the role of rotation in limiting the wind-input of energy and the shear production of turbulence in the boundary layer through Ekman effects. The variation due to h is smaller than due to f, around 5% of the mean value of v_0 for any particular set of forcing (f, B_0, u_*). Since the implementation and generalization is significantly easier if the network only depends on external forcing parameters, we choose to include f as an input to the network and neglect h.

Appendix B: Quantifying Uncertainty Range Covered in the Forcing Data

Table 1 gives the range of the forcing parameters covered in the training data set. A natural question is how much of the variability observed in GCM simulations is covered in the training data. We can estimate this using Shannon entropy (Shannon, 1948) which measures the amount of uncertainty and variability in a variable (Carcassi et al., 2021; Sane et al., 2020, 2021).

Shannon entropy of an event x_i is given by $H(x) = \Sigma_{i=1}^{N} p_i \log_2(1/p_i)$ (Cover, 1999) and measures the average amount of *information* or surprise related to the event. We only use discrete probability distributions. Low probability events have high Shannon entropy because they cause more surprise compared to high probability events. It is a non-parametric measure and does not make any assumption about the distribution. u_* and B_0 are non-Gaussian (Figure B1).

For u_*: H($u_* > 0.03$ m/s) $\approx$ 95.5% and for B_0: H($|B_0| > 2.1 \times 10^{-7}$ m^2/s^3) $\approx$ 86%. This can be interpreted as the values $u_* > 0.03$ have 95.5% uncertainty associated with them. So leaving out values of u_* for which $u_* > 0.03$ removes 95.5% uncertainty from the training data. This is a simplistic estimate and assumes u_* and B_0 are independent. These estimates show that our training data covers 95.5% variability for u_* and 86% of B_0 as observed under realistic conditions in a GCM.

The training data points are uniform and although they cover most of the range seen in realistic conditions, the training data does not follow the same marginal probability distribution of u_* and B_0 as well as the joint proba-

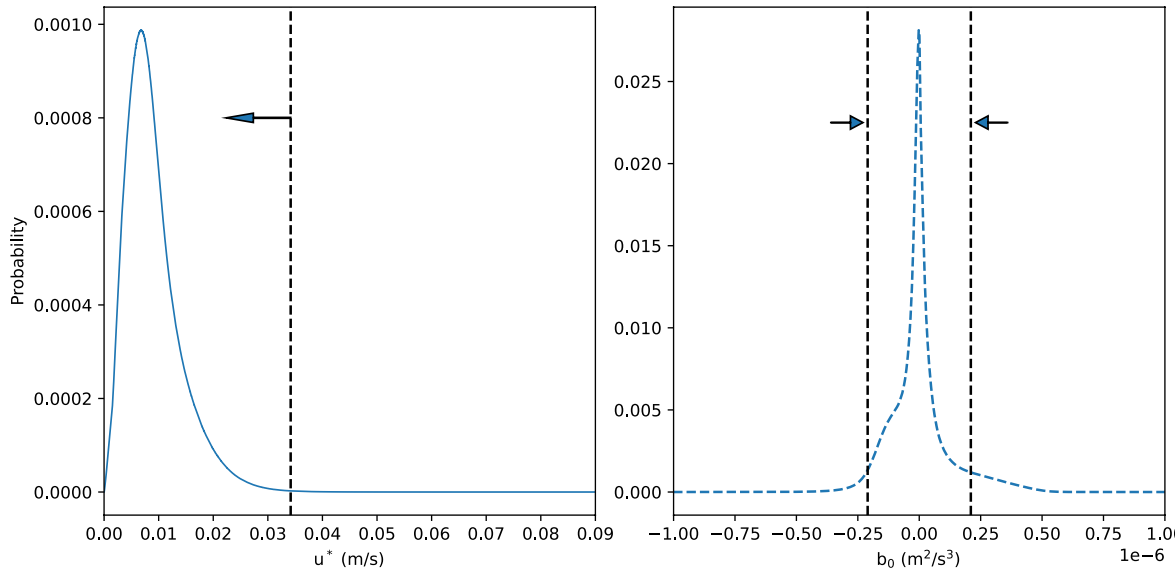

Figure B1. Left: Probability density curve of surface friction velocity u_*. Right: Probability density curve of surface buoyancy flux B_o. The arrows denote the range covered in the training data set.

bility distribution between u_*, B_0. For machine learning application of parameterization development the consequence of sampling from joint distribution of variables from realistic conditions versus having uniformly spaced forcing is unknown as of now and will be left for future study.

Appendix C: List of Symbols and Abbreviations

See Table C1.

Table C1
List of Symbols

Symbol	Description	Units (if applicable)
Ψ	Generic output	–
$\mathcal{F}$	Generic Function	–
$\mathcal{N}$	Neural Network function	–
$\mathbf{w}$	Hyperparameters in a Neural Network	–
f	Coriolis parameter	s^{-1}
w	Vertical velocity	m/s
u_*	Surface friction velocity	$m\ s^{-1}$
b	Buoyancy	$m\ s^{-2}$
B_0	Surface Buoyancy Flux $B_0 = \overline{w'b'}_0$	$m^2\ s^{-3}$
h	Boundary layer depth	m
ϕ	Generic tracer	–
κ_ϕ	Diffusivity of a variable ϕ	$m^2\ s^{-1}$
b	Buoyancy	$m\ s^{-2}$
L	Length scale used in diffusivity	m
z	z co-ordinate, aligned with the local gravitational acceleration	m
σ	Sigma co-ordinate, defined by z/h	–

Table C1
Continued

Symbol	Description	Units (if applicable)
$g(\sigma)$	Shape function which sets variation of diffusivity	–
v_0	Velocity diagnosed from $k-\epsilon$ single column model runs	m s^{-1}
MLD	Mixed layer depth	m
k	Turbulent kinetic energy	m^2/s^2
ϵ	Dissipation of turbulent kinetic energy	m^2/s^3

Data Availability Statement

The code and the data can be obtained from https://doi.org/10.5281/zenodo.8293998. The code includes scripts for generating the training data, training the neural network model, and code for vertical mixing scheme which has been modified to use neural networks. We have also provided code and data for plotting.

Acknowledgments
AS, AA and LZ received M^{2}LInES research funding through the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. AA was also supported by award NA18OAR4320123, from the National Oceanic and Atmospheric Administration (NOAA), U.S. Department of Commerce. 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. We were intellectually supported by various other members of the M^{2}LInES project. We used the Stellar computational resources provided by Princeton University and the National Oceanic and Atmospheric Administration (NOAA) Geophysical Fluid Dynamics Laboratory (GFDL). We thank Dr. Enrico Zorzetto and Dr. Robert Hallberg for providing feedback for this article. We also thank Dr. Jun-Hong Liang and two anonymous reviewers for reviewing and providing profound insights, which led us to improve this manuscript. The authors thank the international Argo project and the various associated national programs for collecting and freely distributing the data set. AS thanks his wife for showing remarkable grit in helping to improve the plain language summary.

References

Adcroft, A., Anderson, W., Balaji, V., Blanton, C., Bushuk, M., Dufour, C. O., et al. (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. https://doi.org/10.1029/2019MS001726

Argo. (2023). *Argo float data and metadata from Global Data Assembly Centre (Argo GDAC)* [Dataset]. https://doi.org/10.17882/42182

Balaji, V., Couvreux, F., Deshayes, J., Gautrais, J., Hourdin, F., & Rio, C. (2022). Are general circulation models obsolete? *Proceedings of the National Academy of Sciences*, *119*(47), e2202075119. https://doi.org/10.1073/pnas.2202075119

Bleck, R. (2002). An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modelling*, *4*(1), 55–88. https://doi.org/10.1016/S1463-5003(01)00012-9

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. https://doi.org/10.1029/2018ms001472

Boyer, T. P., García, H. E., Locarnini, R. A., Zweng, M. M., Mishonov, A. V., Reagan, J. R., et al. (2018). World ocean atlas 2018 [Dataset]. Retrieved from https://www.ncei.noaa.gov/archive/accession/NCEI-WOA18

Brenner, M., Eldredge, J., & Freund, J. (2019). Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, *4*(10), 100501. https://doi.org/10.1103/physrevfluids.4.100501

Brenowitz, N. D., Beucler, T., Pritchard, M., & Bretherton, C. S. (2020). Interpreting and stabilizing machine-learning parametrizations of convection. *Journal of the Atmospheric Sciences*, *77*(12), 4357–4375. https://doi.org/10.1175/JAS-D-20-0082.1

Carcassi, G., Aidala, C. A., & Barbour, J. (2021). Variability as a better characterization of Shannon entropy. *European Journal of Physics*, *42*(4), 045102. https://doi.org/10.1088/1361-6404/abe361

Chor, T., McWilliams, J. C., & Chamecki, M. (2021). Modifications to the k-profile parameterization with nondiffusive fluxes for Langmuir turbulence. *Journal of Physical Oceanography*, *51*(5), 1503–1521. https://doi.org/10.1175/JPO-D-20-0250.1

Christensen, H., & Zanna, L. (2022). Parametrization in weather and climate models. *Oxford Research Encyclopedia of Climate Science*. https://doi.org/10.1093/acrefore/9780190228620.013.826

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314. https://doi.org/10.1007/bf02551274

Damerell, G. M., Heywood, K. J., Calvert, D., Grant, A. L., Bell, M. J., & Belcher, S. E. (2020). A comparison of five surface mixed layer models with a year of observations in the North Atlantic. *Progress in Oceanography*, *187*, 102316. https://doi.org/10.1016/j.pocean.2020.102316

de Boyer Montégut, C., Madec, G., Fischer, A. S., Lazar, A., & Iudicone, D. (2004). Mixed layer depth over the global ocean: An examination of profile data and a profile-based climatology. *Journal of Geophysical Research*, *109*(C12), C12003. https://doi.org/10.1029/2004JC002378

Dunne, J. P., Horowitz, L. W., Adcroft, A. J., Ginoux, P., Held, I. M., John, J. G., et al. (2020). The GFDL earth system model version 4.1 (GFDL-ESM 4.1): Overall coupled model description and simulation characteristics. *Journal of Advances in Modeling Earth Systems*, *12*(11), e2019MS002015. https://doi.org/10.1029/2019MS002015

Fox-Kemper, B., Adcroft, A., Böning, C. W., Chassignet, E. P., Curchitser, E., Danabasoglu, G., et al. (2019). Challenges and prospects in ocean circulation models. *Frontiers in Marine Science*, *6*, 65. https://doi.org/10.3389/fmars.2019.00065

Frezat, H., Le Sommer, J., Fablet, R., Balarac, G., & Lguensat, R. (2022). A posteriori learning for quasi-geostrophic turbulence parametrization. *Journal of Advances in Modeling Earth Systems*, *14*(11), e2022MS003124. https://doi.org/10.1029/2022MS003124

Gregory, W., Bushuk, M., Adcroft, A., Zhang, Y., & Zanna, L. (2023). Deep learning of systematic sea ice model errors from data assimilation increments. *Journal of Advances in Modeling Earth Systems*, *15*, e2023MS003757. https://doi.org/10.1029/2023MS003757

Griffies, S. M., Danabasoglu, G., Durack, P. J., Adcroft, A. J., Balaji, V., Böning, C. W., et al. (2016). OMIP contribution to CMIP6: Experimental and diagnostic protocol for the physical component of the ocean model intercomparison project. *Geoscientific Model Development*, *9*(9), 3231–3296. https://doi.org/10.5194/gmd-9-3231-2016

Guillaumin, A. P., & Zanna, L. (2021). Stochastic-deep learning parameterization of ocean momentum forcing. *Journal of Advances in Modeling Earth Systems*, *13*(9), e2021MS002534. https://doi.org/10.1029/2021ms002534

Gutjahr, O., Brüggemann, N., Haak, H., Jungclaus, J. H., Putrasahan, D. A., Lohmann, K., & von Storch, J.-S. (2021). Comparison of ocean vertical mixing schemes in the Max Planck Institute Earth System Model (MPI-ESM1.2). *Geoscientific Model Development*, *14*(5), 2317–2349. https://doi.org/10.5194/gmd-14-2317-2021

Hawkins, E., & Sutton, R. (2009). The potential to narrow uncertainty in regional climate predictions. *Bulletin of the American Meteorological Society*, *90*(8), 1095–1108. https://doi.org/10.1175/2009BAMS2607.1

Held, I. M., Guo, H., Adcroft, A., Dunne, J. P., Horowitz, L. W., Krasting, J., et al. (2019). Structure and performance of GFDL's CM4.0 climate model. *Journal of Advances in Modeling Earth Systems*, *11*(11), 3691–3727. https://doi.org/10.1029/2019MS001829

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. https://doi.org/10.1016/0893-6080(91)90009-t

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huber, M. B., & Zanna, L. (2017). Drivers of uncertainty in simulated ocean circulation and heat uptake. *Geophysical Research Letters*, *44*(3), 1402–1413. https://doi.org/10.1002/2016gl071587

Jackson, L., Hallberg, R., & Legg, S. (2008). A parameterization of shear-driven turbulence for ocean climate models. *Journal of Physical Oceanography*, *38*(5), 1033–1053. https://doi.org/10.1175/2007jpo3779.1

Jin, F.-F., & An, S.-I. (1999). Thermocline and zonal advective feedbacks within the equatorial ocean recharge oscillator model for ENSO. *Geophysical Research Letters*, *26*(19), 2989–2992. https://doi.org/10.1029/1999GL002297

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kraus, E. B., & Turner, J. S. (1967). A one-dimensional model of the seasonal thermocline: II. The general theory and its consequences. *Tellus A: Dynamic Meteorology and Oceanography*, *19*(1), 98. https://doi.org/10.3402/tellusa.v19i1.9753

Large, W. G., McWilliams, J. C., & Doney, S. C. (1994). Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Reviews of Geophysics*, *32*(4), 363–403. https://doi.org/10.1029/94rg01872

Li, Q., Reichl, B. G., Fox-Kemper, B., Adcroft, A. J., Belcher, S. E., Danabasoglu, G., et al. (2019). Comparing ocean surface boundary vertical mixing schemes including Langmuir turbulence. *Journal of Advances in Modeling Earth Systems*, *11*(11), 3545–3592. https://doi.org/10.1029/2019MS001810

Liang, J.-H., Yuan, J., Wan, X., Liu, J., Liu, B., Jang, H., & Tyagi, M. (2022). Exploring the use of machine learning to parameterize vertical mixing in the ocean surface boundary layer. *Ocean Modelling*, *176*, 102059. https://doi.org/10.1016/j.ocemod.2022.102059

Locarnini, R. A., Mishonov, A. V., Baranova, O. K., Boyer, T. P., Zweng, M. M., García, H. E., et al. (2019). World ocean atlas 2018 (Vol. 1: Temperature) [Dataset]. Retrieved from https://www.ncei.noaa.gov/archive/accession/NCEI-WOA18

Lopez-Gomez, I., Christopoulos, C., Langeland Ervik, H. L., Dunbar, O. R. A., Cohen, Y., & Schneider, T. (2022). Training physics-based machine-learning parameterizations with gradient-free ensemble Kalman methods. *Journal of Advances in Modeling Earth Systems*, *14*(8), e2022MS003105. https://doi.org/10.1029/2022MS003105

Mansfield, L. A., & Sheshadri, A. (2022). Calibration and uncertainty quantification of a gravity wave parameterization: A case study of the quasi-biennial oscillation in an intermediate complexity climate model. *Journal of Advances in Modeling Earth Systems*, *14*(11), e2022MS003245. https://doi.org/10.1029/2022MS003245

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *ICML*.

Niiler, P. (1977). One-dimensional models. In *Modeling and prediction of the upper layers of the ocean* (pp. 143–172). Pergamon Press.

O'Brien, J. J. (1970). A note on the vertical structure of the eddy exchange coefficient in the planetary boundary layer. *Journal of the Atmospheric Sciences*, *27*(8), 1213–1215. https://doi.org/10.1175/1520-0469(1970)027<1213:anotvs>2.0.co;2

O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548–2563. https://doi.org/10.1029/2018MS001351

Orenstein, P., Fox-Kemper, B., Johnson, L., Li, Q., & Sane, A. (2022). Evaluating coupled climate model parameterizations via skill at reproducing the monsoon intraseasonal oscillation. *Journal of Climate*, *35*(6), 1873–1884. https://doi.org/10.1175/JCLI-D-21-0337.1

Partee, S., Ellis, M., Rigazzi, A., Shao, A. E., Bachman, S., Marques, G., & Robbins, B. (2022). Using machine learning at scale in numerical simulations with SmartSim: An application to ocean climate modeling. *Journal of Computational Science*, *62*, 101707. https://doi.org/10.1016/j.jocs.2022.101707

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library [Software]. 8024–8035. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Peters, H., & Baumert, H. Z. (2007). Validating a turbulence closure against estuarine microstructure measurements. *Ocean Modelling*, *19*(3–4), 183–203. https://doi.org/10.1016/j.ocemod.2007.07.002

Ramadhan, A., Marshall, J., Souza, A., Lee, X. K., Piterbarg, U., Hillier, A., et al. (2023). Capturing missing physics in climate model parameterizations using neural differential equations. *arXiv preprint arXiv:2010.12559*. https://doi.org/10.48550/arXiv.2010.12559

Reichl, B. G., Adcroft, A., Griffies, S. M., & Hallberg, R. (2022). A potential energy analysis of ocean surface mixed layers. *Journal of Geophysical Research: Oceans*, *127*(7), e2021JC018140. https://doi.org/10.1029/2021JC018140

Reichl, B. G., & Hallberg, R. (2018). A simplified energetics based planetary boundary layer (ePBL) approach for ocean climate simulations. *Ocean Modelling*, *132*, 112–129. https://doi.org/10.1016/j.ocemod.2018.10.004

Reichl, B. G., & Li, Q. (2019). A Parameterization with a constrained potential energy conversion rate of vertical mixing due to Langmuir turbulence. *Journal of Physical Oceanography*, *49*(11), 2935–2959. https://doi.org/10.1175/JPO-D-18-0258.1

Rodi, W. (1987). Examples of calculation methods for flow and mixing in stratified fluids. *Journal of Geophysical Research*, *92*(C5), 5305–5328. https://doi.org/10.1029/jc092ic05p05305

Roemmich, D., & Gilson, J. (2009). The 2004-2008 mean and annual cycle of temperature, salinity, and steric height in the global ocean from the Argo program. *Progress in Oceanography*, *82*(2), 81–100. https://doi.org/10.1016/j.pocean.2009.03.004

Sane, A., Fox-Kemper, B., & Ullman, D. (2020). Internal vs forced variability metrics for geophysical flows using information theory. *Earth and Space Science Open Archive*, *34*. https://doi.org/10.1002/essoar.10505545.4

Sane, A., Fox-Kemper, B., Ullman, D. S., Kincaid, C., & Rothstein, L. (2021). Consistent predictability of the ocean state ocean model using information theory and flushing timescales. *Journal of Geophysical Research: Oceans*, *126*(7), e2020JC016875. https://doi.org/10.1029/2020JC016875

Schumann, U., & Gerz, T. (1995). Turbulent mixing in stably stratified shear flows. *Journal of Applied Meteorology and Climatology*, *34*(1), 33–48. https://doi.org/10.1175/1520-0450-34.1.33

Shamekh, S., & Gentine, P. (2023). Learning atmospheric boundary layer turbulence. *ESS Open Archive*. https://doi.org/10.22541/essoar.168748456.60017486/v1

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. https://doi.org/10.1073/pnas.2216158120

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(April 1928), 379–423. 623–656. https://doi.org/10.1002/j.1538-7305.1948.tb00917.x

Souza, A. N., Wagner, G. L., Ramadhan, A., Allen, B., Churavy, V., Schloss, J., et al. (2020). Uncertainty quantification of ocean parameterizations: Application to the k-profile-parameterization for penetrative convection. *Journal of Advances in Modeling Earth Systems*, *12*(12), e2020MS002108. https://doi.org/10.1029/2020MS002108

Tirodkar, S., Murtugudde, R., Behera, M. R., & Balasubramanian, S. (2022). A comparative study of vertical mixing schemes in modeling the Bay of Bengal dynamics. *Earth and Space Science*, *9*(8), e2022EA002327. https://doi.org/10.1029/2022ea002327

Todd, A., Zanna, L., Couldrey, M., Gregory, J., Wu, Q., Church, J. A., et al. (2020). Ocean-only FAFMIP: Understanding regional patterns of ocean heat content and dynamic sea level change. *Journal of Advances in Modeling Earth Systems*, *12*(8), e2019MS002027. https://doi.org/10.1029/2019ms002027

Tsujino, H., Urakawa, S., Nakano, H., Small, R. J., Kim, W. M., Yeager, S. G., et al. (2018). JRA-55 based surface dataset for driving ocean–sea-ice models (JRA55-do) [Dataset]. Ocean Modelling, *130*, 79–139. https://doi.org/10.1016/j.ocemod.2018.07.002

Umlauf, L., & Burchard, H. (2005). Second-order turbulence closure models for geophysical boundary layers. A review of recent work. *Continental Shelf Research*, *25*(7–8), 795–827. https://doi.org/10.1016/j.csr.2004.08.004

Umlauf, L., Burchard, H., & Bolding, K. (2014). GOTM source code and test case documentation. Retrieved from https://gotm.net/portfolio/software/

Van Roekel, L., Adcroft, A. J., Danabasoglu, G., Griffies, S. M., Kauffman, B., Large, W., et al. (2018). The KPP boundary layer scheme for the ocean: Revisiting its formulation and benchmarking one-dimensional simulations relative to LES. *Journal of Advances in Modeling Earth Systems*, *10*(11), 2647–2685. https://doi.org/10.1029/2018ms001336

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, *11*(1), 1–10. https://doi.org/10.1038/s41467-020-17142-3

Yuval, J., & O'Gorman, P. A. (2023). Neural-network parameterization of subgrid momentum transport in the atmosphere. *Journal of Advances in Modeling Earth Systems*, *15*, e2023MS003606. https://doi.org/10.1029/2023MS003606

Zanna, L., & Bolton, T. (2020). Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, *47*(17), e2020GL088376. https://doi.org/10.1029/2020gl088376

Zhang, C., Perezhogin, 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*, e2023MS003697. https://doi.org/10.1029/2023MS003697