

NORi: An ML-Augmented Ocean Boundary Layer Parameterization

Xin Kai Lee^{1,2,3}, Ali Ramadhan^{1,2*}, Andre Souza¹, Gregory LeClaire Wagner^{1†},
Simone Silvestri^{1‡}, John Marshall^{1,2} and Raffaele Ferrari^{1,2}

¹Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology

²Center for Computational Science and Engineering, Massachusetts Institute of Technology

³Department of Physics, Imperial College London

Key Points:

- NORi uses neural networks to augment a downgradient physical closure to parameterize ocean boundary layer turbulence.
- *A posteriori* calibration, where loss depends on variable trajectories instead of instantaneous fluxes, ensures robust numerical stability.
- NORi is tested against large-eddy simulations and is found to be numerically stable when embedded in an idealized 3D ocean model.

*Current affiliation: atdepth

†Current affiliation: Aeolus Labs

‡Current affiliation: Department of Environment, Land and Infrastructure Engineering, Politecnico di Torino

Corresponding author: Xin Kai Lee, me@xinkailee.com

Abstract

NORi is a machine-learned (ML) parameterization of ocean boundary layer turbulence that is physics-based and augmented with neural networks. NORi stands for neural ordinary differential equations (NODEs) Richardson number (Ri) closure. The physical parameterization is controlled by a Richardson number-dependent diffusivity and viscosity. The NODEs are trained to capture the entrainment through the base of the boundary layer, which cannot be represented with a local diffusive closure. The parameterization is trained using large-eddy simulations in an *a posteriori* fashion, where parameters are calibrated with a loss function that explicitly depends on the actual time-integrated variables of interest rather than the instantaneous subgrid fluxes, which are inherently noisy. NORi is designed for the realistic nonlinear equation of state of seawater and demonstrates excellent prediction and generalization capabilities in capturing entrainment dynamics under different convective strengths, oceanic background stratifications, rotation strengths, and surface wind forcings. NORi is numerically stable for at least 100 years of integration time in large-scale simulations, despite only being trained on 2-day horizons, and can be run with time steps as long as one hour. The highly expressive neural networks, combined with a physically-rigorous base closure, prove to be a robust paradigm for designing parameterizations for climate models where data requirements are drastically reduced, inference performance can be directly targeted and optimized, and numerical stability is implicitly encouraged during training.

Plain Language Summary

Climate models struggle to represent small-scale mixing processes in the ocean that occur at scales of 1-100 meters because global simulations can only resolve features larger than about 10 kilometers. These mixing processes, driven by winds, evaporation, and surface cooling, critically affect how heat, salinity, and other properties are exchanged between the atmosphere and the interior of the ocean, influencing climate predictions. We introduce NORi, a new approach that combines traditional physics-based equations with modern machine learning to better represent these small-scale processes. Traditional methods often sacrifice accuracy for computational speed or vice versa. NORi achieves both by using neural networks to enhance simpler physics-based models where they fall short. What makes NORi effective is our training approach—using high-resolution simulations as “ground truth” and focusing on correctly predicting the outcomes over time rather than matching instantaneous variables. This approach produced a model that remains stable when run for 100 years, despite being trained on just 2 days of data. NORi also works efficiently with larger time steps, making it practical for long-term climate simulations. By bridging physics and machine learning in this way, NORi represents a new paradigm for developing accurate, computationally efficient models that require less training data.

1 Introduction

State-of-the-art global ocean models used in climate studies solve the equations that govern ocean dynamics and thermodynamics on grids with resolutions of 10 km or coarser (Hewitt et al., 2020; Silvestri et al., 2025). At this resolution, many subgrid-scale processes cannot be explicitly resolved. In the upper ocean boundary layers, these include turbulent mixing driven by cooling, evaporation, and wind stresses on scales from a few hundred meters down to centimeters. Despite their small spatial scales, this mixing has a fundamental impact on the large-scale structure of the ocean by regulating the exchange of heat, carbon, and other climatically important tracers between the atmosphere and the ocean interior. Subgrid-scale parameterizations are therefore introduced in climate models to represent the impact of these processes on large-scale variables.

Traditional parameterizations of turbulent mixing in the upper ocean rely on known physics, typically encoded in scaling laws with a number of free parameters that are determined with empirical data. The most parsimonious parameterizations are first-order diffusive closures, where vertical diffusivities are increased when surface winds, cooling, and/or evaporation are strong enough to trigger turbulent mixing. The Pacanowski-Philander parameterization (Pacanowski & Philander, 1981) is one such example where the vertical diffusivity is a function of local stratification and vertical shear via the Richardson number. Although this diffusive closure smooths local gradients of momentum, temperature, and salinity, it cannot accurately represent entrainment at the base of the boundary layer. Entrainment is driven by convective plumes formed by surface cooling and/or evaporation, which penetrate beneath the mixed layer and draw denser water from the thermocline into the mixed layer. This process sharpens the gradient at the base of the boundary layer—an inherently anti-diffusive, nonlocal mechanism, as illustrated in Figure 1. To represent entrainment in purely diffusive first-order closures, the widely-used K-Profile Parameterization (KPP, Large et al., 1994) adds a nonlocal flux term to the diffusive flux. Bulk parameterizations, originally proposed by Niiler (1977) and recently adopted for use in large-scale ocean models (ePBL, Reichl & Hallberg, 2018), are formulated only in terms of vertically-averaged properties of the boundary layer through bulk energetic arguments and can be thought of as a fully nonlocal model. Higher-order second-moment closure models, such as the celebrated k - ϵ parameterizations (Umlauf & Burchard, 2003; Mellor & Yamada, 1982) address the nonlocality of turbulent mixing by formulating equations that predict the evolution and transport of quantities such as turbulent kinetic energy (TKE) and energy dissipation. Although more accurate than first-order closure models, they are not widely used in global climate models, as they require shorter time steps than the equations for large-scale ocean dynamics and therefore add substantial computational cost (Reffray et al., 2015; Reichl & Hallberg, 2018). Most recently, the Convective Adjustment TKE parameterization (CATKE Wagner, Hillier, et al., 2025) has been introduced to leverage the benefits of higher-order models while reducing their computational cost. CATKE is a 1.5-order closure that uses a prognostic equation for TKE to compute the diffusivity. Its skill compares favorably with the k - ϵ parameterization at a computational cost comparable to KPP.

Despite much progress in the formulation of boundary layer parameterizations, substantial biases persist in the simulation of the upper ocean in large-scale ocean models, especially in the Southern Ocean (DuVivier et al., 2018; Treguier et al., 2023; Sallée et al., 2013) and the tropics (Li & Xie, 2014). In particular, the boundary layer depth bias is often of the same order as the depth itself (DuVivier et al., 2018; Treguier et al., 2023).

Machine learning methods offer a potential path to reduce these biases and have been increasingly used in recent years to reduce biases in climate model parameterization for subgrid-scale processes, due to advances in computing hardware and machine learning software (Bolton & Zanna, 2019). For upper ocean boundary layer turbulence, a Bayesian approach has been used to tune and uncover structural deficiencies of KPP (Souza et al., 2020). Neural networks have also been used to predict the evolution of temperature and salinity at Station Papa (Liang et al., 2022), improve the parameterization of ePBL (Sane et al., 2023), and calibrate the vertical diffusivity profile in KPP (Yuan et al., 2024). However, the success in reducing biases has been limited.

One of the most important challenges facing neural network-based parameterizations applied to climate modeling is the numerical stability and physical accuracy over long integration times (Chattopadhyay et al., 2023). The causes of such numerical instabilities are manifold, including the spectral bias of neural networks, which prioritize learning low frequency dynamics which may cause networks to underrepresent high-frequency dynamics which can be important for turbulence and chaotic processes (Chattopadhyay et al., 2023), as well as the use of *a priori* calibration to train the neural network (Frezat et al., 2022). In *a priori* calibration, the neural network parameterization is trained of-

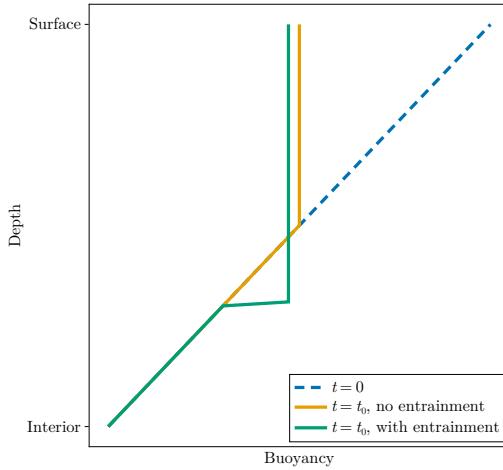


Figure 1. Schematic of the vertical profile of buoyancy as a function of depth below the ocean surface—buoyancy is negatively proportional to the density anomaly generated by a change in temperature and salinity. At time $t = 0$, the buoyancy decreases linearly with depth. After some time, in response to buoyancy loss at the surface, a well-mixed buoyancy layer develops in the upper ocean. In the absence of entrainment through the base of the mixed layer, the buoyancy profile smoothly connects to the stratified interior. Entrainment results in a further deepening of the boundary layer and the development of a sharp buoyancy jump at its base.

fine to match the diagnosed, unresolved subgrid-scale fluxes as a function of large-scale variables. Since no parameterization is perfect, once implemented in the dynamical equations of the climate model, it will introduce some random errors in addition to the desired fluxes. Over time, errors accumulate and amplify, leading to the climate variables diverging from the correct trajectories, not unlike how chaotic systems will have diverging trajectories with infinitesimal perturbations in the initial conditions. This challenge has been well documented in data-driven weather forecast models based on neural networks, which can be considered as the state-of-the-art in the application of machine learning tools to atmospheric models, including FourCastNet (Pathak et al., 2022), Graph-Cast (Lam et al., 2023), and Pangu (Bi et al., 2023), which can exhibit instabilities or error accumulation when autoregressively integrated over long times (Chattopadhyay et al., 2023).

A recently suggested alternative to obtain stable solutions is *a posteriori* training, albeit with an increase in training cost. In *a posteriori* training, the parameterization is trained to match the evolution of coarse-grained variables over consecutive time steps. Although the model still predicts subgrid-scale fluxes, the training is adjusted to identify fluxes which guarantee that the coarse-grained variables proceed similarly to the training dataset, rather than matching fluxes independently at each time step. Importantly, the parameterized fluxes do not need to match exactly those in the training dataset, as the training data is noisy, and the fluxes required to produce the correct tendencies are not unique. Instead, the parameterization is trained in an end-to-end fashion by integrating the coarse-grained variables forward in time and using the integrated solution in the loss function. Examples of parameterizations calibrated *a posteriori* include parameterizations of 2D turbulence (Kochkov et al., 2021), quasi-geostrophic turbulence (Frezzat et al., 2022), 3D turbulence (Sirignano et al., 2020; Stachenfeld et al., 2021), and the CATKE boundary layer parameterization (Wagner, Hillier, et al., 2025). Another notable example of *a posteriori* calibration in earth system modeling is NeuralGCM (Kochkov

et al., 2024), where neural networks learn holistic tendencies to augment the dynamical core rather than for targeted physical parameterizations. For many *a posteriori* calibration problems with relatively few number of parameters or have reasonably good priors, ensemble-based methods such as Ensemble Kalman Inversion (EKI) (Iglesias et al., 2013; Dunbar et al., 2022) are effective and practical (Gjini et al., 2025) as they do not require automatic-differentiability. However, neural networks generally have orders of magnitude more free parameters than physical models. If ensemble-based methods are unable to explore the entire parameter space due to the computational expense of running many instances of the forward model, then informative priors become especially important. Since neural network parameters do not explicitly represent physically interpretable processes, generating good priors for them may be challenging even with pre-training. In such circumstances, ensemble-based methods can perform poorly on *a posteriori* calibration, and the model should be trained in fully differentiable frameworks. This is becoming more common in modern languages like Python (with the use of JAX (Frostig et al., 2018)) and Julia (with software packages such as Lux.jl (Pal, 2023) and Enzyme.jl (Moses & Churavy, 2020)), but it is not routine in climate modeling. Therefore, *a posteriori* calibration is not always possible for large models as it requires the forward model to be fully automatic-differentiable and comes at a higher computational as well as memory cost. However, recent developments on building out full-fledged, general purpose automatic differentiability in Julia-based earth system models have made Oceananigans differentiable, allowing *a posteriori* tasks to possibly be carried out natively (Moses et al., 2025).

We present NORi (pronounced noh-ree, which refers to a dried edible seaweed in Japanese): an ocean boundary layer parameterization that combines the strengths of physics-based and machine-learned parameterizations. The philosophy is to start with as simple a physics-based parameterization as possible and then augment it with a data-trained neural network. The physics-based model is a first-order diffusive closure similar to the Pacanowski-Philander model (Pacanowski & Philander, 1981) where the eddy diffusivity is a simple function of the local gradient Richardson number. To capture the additional mixing physics of entrainment, the parameterization is augmented with neural networks trained *a posteriori* with high-resolution simulations of upper ocean turbulence. The *a posteriori* calibration uses a neural ordinary differential equation (NODE) framework (Chen et al., 2019; Rackauckas et al., 2021) where the parameterization is integrated forward in time within fully-differentiable ODE solvers. This is a continuation of Ramadhan et al. (2023, arXiv), which pioneered the idea of modeling turbulent fluxes in convective scenarios using a physical closure (convective adjustment) augmented with neural networks, trained in an *a posteriori* fashion. Our work expands upon Ramadhan et al. (2023, arXiv) by including effects of wind-driven mixing and taking into account realistic thermodynamics of seawater. NORi considers the full nonlinear equation of state (Roquet et al., 2015) and provides a closure for the complete suite of prognostic variables for ocean boundary layer dynamics: momentum, temperature, and salinity. We show that incorporating known physics into the parameterization reduces the amount of data required for training, allows the use of a small neural network, and improves model generalizability and numerical stability. Like all other vertical mixing schemes for the boundary layer mentioned above, NORi currently focuses on mixing processes that deepen the boundary layer, assuming that the large-scale model resolves lateral instabilities due to the mesoscales and submesoscales that drive restratification.

Section 2 introduces the high-resolution, high-fidelity, large-eddy simulations (LES) that we use to train and validate NORi. Section 3 describes the formulation of the NORi column model. Section 4 describes the local eddy-diffusivity closure which encodes the known physics. Section 5 describes the neural network formulation and the *a posteriori* training paradigm used in NORi, while we assess NORi’s performance in Section 6 with single column configurations and Section 7 in a large-scale long-time double-gyre ocean simulation. Finally, we conclude with a few closing remarks in Section 8.

2 High-resolution dataset for boundary layer turbulence

2.1 Large-eddy simulations

To train and validate NORi, high-quality, physically accurate “ground-truth” data are required. To that end, we run large-eddy simulations (LES) of upper ocean turbulence, i.e., simulations that resolve the largest turbulent eddies in the boundary layer and use a subgrid-scale model to represent the smaller, more universal 3D eddies. The LES are generated with Oceananigans.jl (Ramadhan et al., 2020; Wagner, Silvestri, et al., 2025), a finite volume ocean model written in Julia (Bezanson et al., 2017) and optimized for GPUs (Silvestri et al., 2025). The LES data suite focuses on shear-driven and convection-driven deepening of the boundary layer against a stable background stratification. Mathematically, LES solve the incompressible Boussinesq form of the Navier-Stokes equations given by

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - f \hat{\mathbf{k}} \times \mathbf{u} - \nabla p + b \hat{\mathbf{k}}, \quad (1)$$

$$\frac{\partial T}{\partial t} = -\nabla \cdot (\mathbf{u} T), \quad (2)$$

$$\frac{\partial S}{\partial t} = -\nabla \cdot (\mathbf{u} S), \quad (3)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (4)$$

$$b = -g \frac{\sigma - \sigma_0}{\sigma_0}, \quad (5)$$

$$\sigma = \sigma_{\text{TEOS-10}}(T, S, z = 0), \quad (6)$$

where \mathbf{u} are the velocities in the x , y , and z directions, f is the Coriolis parameter, $\hat{\mathbf{k}}$ is the unit vector in the vertical direction (normal to the fluid surface), p is the kinematic pressure, b is the buoyancy of the fluid, T and S are the temperature and salinity of the fluid, σ is the potential density, g is the gravitational acceleration, and $\sigma_0 = 1020 \text{ kg m}^{-3}$ is the reference density. The LES are run with laterally doubly-periodic boundary conditions, prescribed advective fluxes of T , S , and u at the top given by J_T^{top} , J_S^{top} and J_u^{top} , and no-flux boundary conditions for T and S and free-slip boundary conditions for u and v at the bottom. Additionally, we assume that the vertical velocities at the top and bottom surfaces are zero, imposing the no-penetration boundary condition given by

$$w(z = 0) = w(z = -L_z) = 0. \quad (7)$$

To model a realistic ocean, we use the TEOS-10 equation of state (Roquet et al., 2015), which takes into account the nonlinear dependence of buoyancy on temperature and salinity. This nonlinear dependence can lead to effects such as cabbeling, requiring that temperature and salinity be modeled separately. The 9th order weighted essentially non-oscillatory scheme (WENO, Balsara & Shu, 2000) is used as the advection scheme, as it reduces numerical oscillations and is gradient-preserving. At the same time, the dissipative numerics of the WENO scheme also act as an implicit viscous closure while providing a high effective resolution due to its gradient-preserving properties (Silvestri et al., 2023). We do not provide any additional explicit LES closure as this technique has been shown to be desirable when modeling sharp transitions at the thermocline with little sacrifice to numerical accuracy (Pressel et al., 2017). The prognostic variables in (1)–(4) are evolved using a third-order Runge-Kutta scheme with a pressure projection (or predictor-corrector) method that involves the solution of a three-dimensional Poisson equation for pressure (Schumann & Sweet, 1988).

The snapshots in Figure 2 show the simulated buoyancy and vertical velocity fields of two representative examples of upper ocean turbulence: free convection and shear turbulence acting on a fluid initialized with constant vertical stratification. These simulations used for illustration purposes in Figure 2 are run with an isotropic resolution of

0.5 m and a Coriolis parameter of $f = 8 \times 10^{-5} \text{ s}^{-1}$. Turbulence in the free convection simulation is generated by buoyancy loss at the ocean surface, leading to static instability, and in the shear simulation by a surface wind stress. Over time, convection- and wind-driven turbulence create a well-mixed layer at the top of the fluid, which deepens over time. Between the mixed layer and the stratified interior, a thermocline with sharp gradients is formed. This vertical structure can be seen in the horizontally-averaged temperature and salinity fields in the 3rd and 4th columns of Figure 2 for different times.

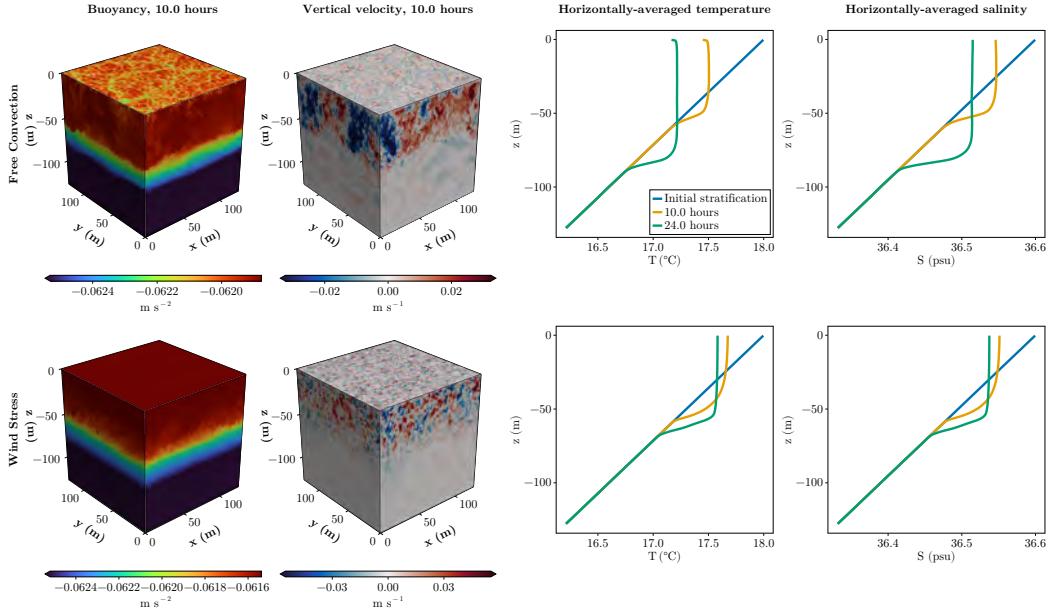


Figure 2. Large-eddy simulations (LES) for free convection and pure wind stress scenarios in a horizontally doubly-periodic domain of size $(L_x, L_y, L_z) = (128, 128, 128) \text{ m}$ with a grid resolution of 0.5 m and a Coriolis parameter of $f = 8 \times 10^{-5} \text{ s}^{-1}$. The top row shows a convective turbulence LES driven by surface cooling, while the bottom row shows a shear turbulence LES driven by surface wind stress. The first and second columns show snapshots of the buoyancy and vertical velocity fields. The third and fourth columns show the time evolution of the horizontally-averaged temperature and salinity profiles characterized by a deepening mixed layer in response to the surface forcings.

The two simulations illustrate the fundamental differences between the two scenarios of boundary layer turbulence. In free convection, the case shown in the upper row of Figure 2, coherent plumes with strong downward velocities plunge from the surface towards and through the thermocline. As the plumes reach the base of the boundary layer, they overshoot into the stratified ocean interior due to inertia and entrain denser fluid from the thermocline into the mixed layer, increasing the mixed layer's mean density. Entrainment mixing is nonlocal as it occurs at the base of the boundary layer, driven by surface air-sea fluxes. However, in the shear-driven case in the bottom row of Figure 2, surface wind stress creates shear instability, which generates local eddies that decay away from the surface. The plumes generated by wind stress are much smaller in scale than those created by free convection. Thus, from a parameterization perspective, the shear instability can be regarded as a local process. We will therefore develop NORi, starting with a diffusive closure that is known to represent well the local mixing induced by both convective and shear turbulence. To address the nonlocal entrainment of fluid from

the interior into the mixed layer, which has proven very challenging to parameterize, we will add a machine-learned neural network component to NORi.

2.2 Training and validation data generation

A LES training and validation suite is run in order to calibrate NORi. All LES are initialized with uniform temperature and salinity profiles in the horizontal direction and with a linear gradient in the vertical direction. The simulations are run under a variety of surface forcing scenarios: winds only, cooling or evaporation only, cooling + evaporation, winds + cooling and/or evaporation, winds + heating, and winds + heating + precipitation. All of these forcings produce destabilizing fluxes that mix away the ocean stratification and result in a deepening mixed layer in the upper ocean. The LES domains are $(L_x, L_y, L_z) = (512, 512, 256)$ m with an isotropic resolution of 2 m, which achieves about 12 simulated days per day on an NVIDIA V100 GPU. Finer resolutions resulted in very similar results, suggesting that the bulk of the turbulent mixing is resolved with the 2 m isotropic grid.

The initial temperature and salinity stratification is based on linear fits to summertime profiles from the ISAS reanalysis product of Argo profiles from 2002 to 2020 (Gaillard et al., 2016)—we choose summertime profiles because we are interested in simulating the deepening of the mixed layer in fall and winter. We consider profiles from three latitude bands: midlatitude Atlantic (strong temperature and salinity gradient), equatorial Pacific (strong temperature but weak salinity gradient) and Southern Ocean (salinity-dominated stratification, equation of state highly nonlinear). This ensures that we cover a representative range of oceanic stratifications that will be encountered at inference time, including regimes where the equation of state is highly nonlinear. We complement these observationally inspired initial profiles with additional ones designed to span a broader parameter space, which is key to ensuring generalizability of the parameterization and avoiding overfitting during training. In particular, some training cases lack temperature or salinity gradients, others have inverse (stabilizing) salinity stratification, and others operate in the more nonlinear regime of seawater’s equation of state at lower temperatures. The actual values of the physical parameters used in the LES suite can be found in Appendix B.

The prescribed air-sea fluxes are constant in time and space and chosen to span a representative range of conditions encountered in the ocean. The forcing suite covers the most common oceanic regimes, from free convection (no wind stress) to pure wind scenarios (no buoyancy flux) to mixed conditions. The range of air-sea flux values is very broad to ensure, as much as possible, that NORi is never run under forcing conditions outside the training dataset. The cooling rates range from 0 W m^{-2} to 2500 W m^{-2} to capture strong entrainment scenarios, noting that the mean monthly cooling rate of the Gulf Stream can go up to around 600 W m^{-2} (The Climode Group: et al., 2009), while latent heat fluxes due to evaporative cooling can reach 500 W m^{-2} during strong wind events such as tropical cyclones (Vreugdenhil & Gayen, 2021). On the other hand, evaporation rates span between 0 m yr^{-1} and around 45 m yr^{-1} . For context, the winter average of the evaporation rates over the Gulf Stream is around 3 m yr^{-1} (Yu, 2007). Wind stresses vary roughly from 0 N m^{-2} to very strong winds at 0.5 N m^{-2} , and typical monthly-averaged wind stress values are around 0.2 N m^{-2} (Copernicus Climate Change Service, 2019). The main training suite consists of 62 simulations, and the LES parameters can be found in Tables B1, B2, B3, and B4. The parameters for the 32 simulations used in the validation and testing suite are given in Tables B1 and B5. This suite of datasets is publicly available for easy download and access on Zenodo (Lee, 2025b).

All LES are run for 2 days, a compromise between a long enough simulation that captures the physics of turbulent mixing under different scenarios and short enough to limit the memory footprint of the calibration. This is because to reduce the computa-

tional cost, we do not perform checkpointing during the forward-backward automatic differentiation cycle, i.e., we save the model state at all intermediate time steps rather than at specific, sparsely distributed times. These intermediate states are needed for the serial-in-time paradigm of *a posteriori* calibration during backpropagation, since the loss function depends on the entire trajectory of the model state. Using checkpointing reduces the memory footprint at the cost of increased computational time, as the model state at a specific time has to be recomputed from its nearest checkpoint.

3 Modeling unresolved fluxes in a coarse-grained column model

A convection-resolving simulation requires a resolution of $\mathcal{O}(1 \text{ cm})$. Using an LES, one might be able to represent such processes at $\mathcal{O}(1 \text{ m})$. But even $\mathcal{O}(1 \text{ m})$ is far beyond the resolution that can be achieved in global oceanic simulations now and in the foreseeable future. Our goal is thus to develop a parameterization that predicts the deepening of the boundary layer in response to forcing without the need to resolve the small-scale horizontal motions. In practice, we want to find a closed-form 1D equation that predicts the evolution of the LES area-averaged profiles of velocity, temperature, and salinity as a function of initial conditions and air-sea fluxes.

Let us start by defining the area averaging operator of some LES variable ϕ in a doubly-periodic horizontal domain Ω ,

$$\phi(x, y, z, t) = \bar{\phi}(z, t) + \phi'(x, y, z, t), \quad (8)$$

$$\bar{\phi} \equiv \frac{1}{L_x L_y} \iint_{\Omega} \phi \, dx \, dy, \quad (9)$$

$$\bar{\phi}' \equiv 0, \quad (10)$$

where $\bar{\phi}$ is the horizontal area-average of ϕ , L_x and L_y are the lateral domain dimensions, and ϕ' is its deviation from the horizontal average. Applying this Reynolds-like averaging as well as the no-normal flow boundary conditions at the top and bottom of the LES domain according to Equation (7), the momentum, temperature, and salinity Equations (1), (2), and (3), reduce to a set of 1D equations (see complete derivation in [Appendix A](#)),

$$\frac{\partial \bar{u}}{\partial t} = -\frac{\partial}{\partial z} \bar{u}' w' + f \bar{v}, \quad (11)$$

$$\frac{\partial \bar{v}}{\partial t} = -\frac{\partial}{\partial z} \bar{v}' w' - f \bar{u}, \quad (12)$$

$$\frac{\partial \bar{T}}{\partial t} = -\frac{\partial}{\partial z} \bar{w}' T', \quad (13)$$

$$\frac{\partial \bar{S}}{\partial t} = -\frac{\partial}{\partial z} \bar{w}' S', \quad (14)$$

where $\bar{u}' w'$, $\bar{v}' w'$, $\bar{w}' T'$, and $\bar{w}' S'$ are the total unresolved momenta, temperature, and salinity fluxes due to subgrid-scale turbulence. Their surface boundary conditions are given by

$$\bar{u}' w'(z = 0) = J_u^{\text{top}}, \quad (15)$$

$$\bar{v}' w'(z = 0) = J_v^{\text{top}}, \quad (16)$$

$$\bar{w}' T'(z = 0) = J_T^{\text{top}}, \quad (17)$$

$$\bar{w}' S'(z = 0) = J_S^{\text{top}}, \quad (18)$$

$$(19)$$

while their bottom boundary conditions are given by

$$\bar{u}' w'(z = -L_z) = \bar{v}' w'(z = -L_z) = \bar{w}' T'(z = -L_z) = \bar{w}' S'(z = -L_z) = 0 \quad (20)$$

where L_z is the vertical extent of the domain. We write equations for temperature and salinity separately because we use the full nonlinear equation of state of seawater, and we cannot combine the two equations into a single one for buoyancy.

4 “Ri”: Eddy-diffusivity closure based on local gradient Richardson number

Following Kochkov et al. (2024), we first attempted to develop a parameterization of the boundary layer using a purely data-driven approach by modeling the full *fluxes* of $\overline{u'w'}$, $\overline{v'w'}$, $\overline{w'T'}$, and $\overline{w'S'}$ with neural networks (note that Kochkov et al. (2024) models the *tendencies* instead). However, we found that neural networks were not able to learn that vertical profiles of temperature and salinity should be persistently statically stable, i.e., $N^2 = \frac{\partial b}{\partial z} \geq 0$ where N^2 is the buoyancy frequency. Instead, neural networks produced a boundary layer that was nearly well-mixed near the surface, but with small random fluctuations in z where N^2 was negative. This led to numerical instability in the hydrostatic formulation, which is used in large-scale ocean models. This deficiency could likely be addressed by increasing the size of the training data set based on the evidence that noise in neural network inference can be reduced by providing more training (Subel et al., 2021). However, increasing the amount of data is costly due to the increased computational and memory requirements during training on a per-epoch basis. Alternatively, one could increase the model’s expressivity by using a deeper neural network, but this also leads to exploding memory and/or compute requirements during training because of the serial-in-time nature of the *a posteriori* calibration approach and raises the cost during inference, as the neural network needs to be evaluated very often (at least once every few time steps) during a forward ocean simulation.

A more effective solution to alleviate the triple conundrum of low training cost, strong numerical stability, and low inference cost per time step is to start by formulating a simple physics-based parameterization that captures most of the boundary layer physics and then augment it with a neural network to address its limitations. In particular, a boundary layer model that enhances the eddy diffusivity whenever the air-sea fluxes are destabilizing guarantees that unstable stratification is rapidly mixed away but fails to capture nonlocal physics, such as entrainment through the boundary layer base. We will refer to this model as the *base closure*. Training a neural network to learn the physics missing in the base closure rather than all boundary layer physics reduces the data requirement and model size, dramatically improves parameterization performance, and reduces numerical instability during training and validation. Mathematically, the base closure represents all subgrid-scale fluxes as down-gradient fluxes of momentum, temperature, and salinity,

$$\overline{u'w'} = J_{u, \text{local}} = -\nu \frac{\partial \overline{u}}{\partial z}, \quad (21)$$

$$\overline{v'w'} = J_{v, \text{local}} = -\nu \frac{\partial \overline{v}}{\partial z}, \quad (22)$$

$$\overline{w'T'} = J_{T, \text{local}} = -\kappa \frac{\partial \overline{T}}{\partial z}, \quad (23)$$

$$\overline{w'S'} = J_{S, \text{local}} = -\kappa \frac{\partial \overline{S}}{\partial z}, \quad (24)$$

where ν and κ are the (positive) eddy viscosity and diffusivity, which depend on the local gradient Richardson number Ri , given by

$$Ri = -\frac{g}{\sigma_0} \frac{\frac{\partial \overline{\sigma}}{\partial z}}{\left(\frac{\partial \overline{u}}{\partial z} \right)^2 + \left(\frac{\partial \overline{v}}{\partial z} \right)^2}. \quad (25)$$

The Ri -dependence is well supported by theory (Drazin & Reid, 2004), ocean observations (e.g. Price et al., 1986), and forms the basis of many ocean boundary layer parameterizations (e.g. Pacanowski & Philander, 1981; Large & Gent, 1999; Wagner, Hillier,

et al., 2025). When $Ri < 0$, the boundary layer stratification is convectively unstable. When the boundary layer is statically stable, but the shear is large enough such that Ri is below some critical value Ri^c , the boundary layer is shear unstable. The specific functional dependence of diffusivity and viscosity on Ri is illustrated in the upper row of Figure 3 and takes the form,

$$\nu = \begin{cases} (\nu_{\text{shear}} - \nu_{\text{conv}}) \tanh \frac{Ri}{\Delta Ri} + \nu_{\text{shear}}, & \text{for } Ri < 0 \\ (\nu_0 - \nu_{\text{shear}}) \frac{Ri}{Ri^c} + \nu_{\text{shear}}, & \text{for } 0 \leq Ri < Ri^c \\ \nu_0, & \text{for } Ri \geq Ri^c \end{cases} \quad (26)$$

$$\kappa = \begin{cases} (\kappa_{\text{shear}} - \kappa_{\text{conv}}) \tanh \frac{Ri}{\Delta Ri} + \kappa_{\text{shear}}, & \text{for } Ri < 0 \\ (\kappa_0 - \kappa_{\text{shear}}) \frac{Ri}{Ri^c} + \kappa_{\text{shear}}, & \text{for } 0 \leq Ri < Ri^c \\ \kappa_0, & \text{for } Ri \geq Ri^c \end{cases} \quad (27)$$

$$\kappa_{\text{shear}} = \frac{\nu_{\text{shear}}}{Pr_{\text{shear}}}, \quad (28)$$

$$\kappa_{\text{conv}} = \frac{\nu_{\text{conv}}}{Pr_{\text{conv}}}. \quad (29)$$

where ν_{conv} , ν_{shear} , Ri^c , ΔRi , Pr_{conv} , and Pr_{shear} are parameters that should be trained with the LES suite. They represent the viscosity under convective turbulence ν_{conv} , and under shear turbulence ν_{shear} . The Pr_{conv} and Pr_{shear} are the turbulent Prandtl numbers under the convective and shear regimes. The other parameters are instead fixed: $\sigma_0 = 1020 \text{ kg m}^{-3}$ is the reference potential density, $g = 9.80665 \text{ m s}^{-2}$ is the gravitational acceleration, $\nu_0 = 1 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$ is the background viscosity whose value is chosen to be small enough to have no impact on the training and validation tests, while κ_0 is computed using the shear Prandtl number $\kappa_0 = \frac{\nu_0}{Pr_{\text{shear}}}$. This formulation is inspired by the Pacanowski-Philander model (Pacanowski & Philander, 1981), but differs from it by using different functional forms and accounting for different mixing rates in the convective and shear regimes. Also, this is only the first step in our parameterization formulation, which is to be augmented by a neural network.

The base closure implies a change in both the magnitude of viscosity and the Prandtl numbers between the convective, shear, and stable regimes. This is supported by the LES solution and mixing length arguments. The eddy diffusivity/viscosity can be expressed as the product of a turbulent velocity times a mixing length scale. Although the turbulent velocity spans similar ranges in all regimes, the mixing length scale is significantly longer in the convective regime, spanning the entire turbulent boundary layer and resulting in stronger mixing of tracers. However, the turbulent Prandtl number is smaller than one, because rotation slows momentum mixing by establishing a thermal wind if vertical mixing is slower than the Coriolis period (Young, 1994). This is not relevant in the shear regime, which is associated with faster mixing time scales.

The free parameters in the base closure are calibrated in an *a posteriori* fashion by minimizing the mean squared difference between the LES column profiles and their vertical gradients with those predicted by the closure for all scenarios and times. The loss function $\mathcal{L}_{\text{base}}$ is defined as

$$\mathcal{L}_{\text{base}}(\bar{u}, \bar{v}, \bar{T}, \bar{S}; \boldsymbol{\theta}_{\text{base}}) = \frac{1}{N_{\text{sim}}} \sum_{a=1}^{N_{\text{sim}}} (\mathcal{L}_u^a + \mathcal{L}_v^a + \mathcal{L}_T^a + \mathcal{L}_S^a + \mathcal{L}_\sigma^a) \quad (30)$$

$$\text{with } \mathcal{L}_\phi^a = \frac{A_\phi^a}{N_t N_z} \sum_{i=1}^{N_z} \sum_{j=1}^{N_t} \left| \bar{\phi}^{a,i,j} - \bar{\phi}_{\text{LES}}^{a,i,j} \right|^2 + \frac{A_{\partial_z \phi}^a}{N_t N_z} \sum_{i=1}^{N_z} \sum_{j=1}^{N_t} \left| \partial_z \bar{\phi}^{a,i,j} - \partial_z \bar{\phi}_{\text{LES}}^{a,i,j} \right|^2, \quad (31)$$

where $\bar{\phi}$ and $\bar{\phi}_{\text{LES}}$ are the profiles of one of the variables (velocity, temperature, salinity) as a function of vertical level i , time j , and scenario a —the overbars are a reminder that we are only parameterizing the area-averaged profiles. Normalization factors A_ϕ^a are computed adaptively based on initial conditions and losses in iteration 0 before training to ensure balanced learning in all variables and loss components. This is done through

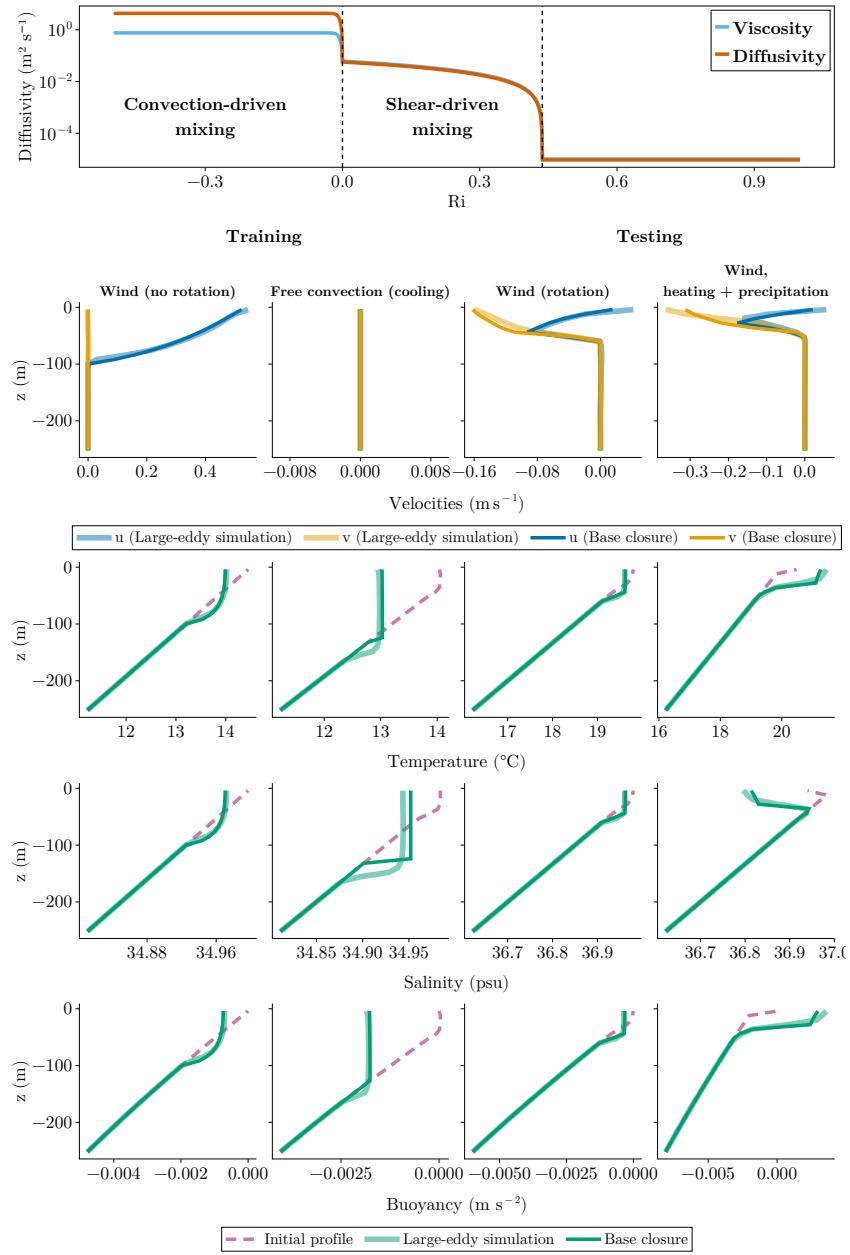


Figure 3. The top panel shows the calibrated diffusivity and viscosity values as a function of the local gradient Richardson number Ri in the base closure model. In the convective range, the viscosity and diffusivity are constant, while they decrease linearly towards a background value in the shear range. The lower panels show the vertical profiles of momentum, temperature, salinity, and buoyancy from LES simulations forced with four different air-sea fluxes and the corresponding predictions from the base closure (the dashed lines represent the initial profiles). From left to right: a wind-forced example with no rotation and a convection example with rotation used to train the base closure; a wind-forced example with rotation and a wind-forced example with heating and precipitation used for validation.

a two-stage process. Firstly, the magnitudes of temperature and salinity losses are reweighted to be inversely proportional to their contributions to density variations, preventing either variable from dominating the loss. The density loss is then weighted so that

$$\frac{\mathcal{L}_\sigma^a}{\mathcal{L}_T^a + \mathcal{L}_S^a} = \frac{1}{9}, \quad (32)$$

imposing a weaker diagnostic constraint on the density vs. temperature and salinity. Momentum losses are normalized to match the sum of density, temperature, and salinity losses. The same weights are also applied to their respective gradient loss counterparts. Secondly, after these physical weightings are applied, the total profile losses and gradient losses are rescaled to be equal. This scheme ensures that all components of the loss function contribute meaningfully to the training process. The mathematical details of how normalization factors are computed are provided in [Appendix C](#). The free parameters $\boldsymbol{\theta}_{\text{base}} = [\nu_{\text{conv}}, \nu_{\text{shear}}, Ri^c, \Delta Ri, Pr_{\text{conv}}, Pr_{\text{shear}}]^\top$ in the base closure are calibrated with Ensemble Kalman Inversion (EKI) using the `EnsembleKalmanProcesses.jl` package ([Iglesias et al., 2013](#); [Dunbar et al., 2022](#)), a gradient-free black box optimization algorithm well suited for problems with a small number of parameters. Since there are only 6 tunable parameters in the base closure, we chose to use EKI rather than differentiable methods because we are able to afford sufficient (200) ensemble members in order to more effectively explore the parameter space compared to gradient-based methods.

The base closure is able to capture the evolution of the vertical profile of the boundary layers under training and testing scenarios where instabilities are driven purely by surface wind stresses. As expected from a purely local parameterization, the base closure underestimates the mixed layer depth when convection is present, since entrainment is a nonlocal process (as discussed in [Section 2](#)). The bottom panels of [Figure 3](#) show examples of vertical profiles of temperature, salinity, and potential density from LES simulations forced with four different air-sea fluxes and the corresponding predictions from the base closure. The base closure is unable to fully capture the deepening of the boundary layer in the free convection training case (second from left), which could be attributed to the missing nonlocal entrainment physics. The other three training and validation cases shown in [Figure 3](#) include purely wind-driven mixing in rotational and irrotational cases, as well as (destabilizing) wind stress combined with (stabilizing) heating and precipitation. In all three cases, mixing is purely local and the base closure is able to predict accurately the depth of the mixed layer.

5 “NO”: Using neural networks to capture nonlocal entrainment

From [Section 4](#), we have seen that the base closure is able to capture local mixing but is unable to represent nonlocal entrainment processes driven by surface buoyancy fluxes. We will now use neural networks as a residual model to capture nonlocal entrainment physics.

5.1 Enforcing physical knowledge and constraints on neural networks with architecture design

In NORi, the neural networks are used to predict the missing entrainment fluxes of temperature and salinity at the base of the boundary layer. Initially, additional neural networks for momenta u and v were included, but they did not improve the parameterization skill despite a substantial increase in training cost. This is likely because the momentum fluxes are dominated by inertia-gravity wave signals rather than turbulent fluctuations, obscuring the entrainment signal that contributes to vertical mixing. Despite the absence of neural networks in the momentum equations, we found that due to the base closure, entrainment fluxes in T and S would still lead to enhanced momenta mixing at the base of the boundary layer.

Mathematically, we add nonlocal terms J_{NN_T} and J_{NN_S} which are computed using neural networks NN_T and NN_S to Equations (23) and (24),

$$\overline{w'T'} = J_{T, \text{local}} + J_{T, \text{nonlocal}} = -\kappa \frac{d\bar{T}}{dz} + J_{\text{NN}_T} \quad (33)$$

$$\overline{w'S'} = J_{S, \text{local}} + J_{S, \text{nonlocal}} = -\kappa \frac{d\bar{S}}{dz} + J_{\text{NN}_S}. \quad (34)$$

By construction, the surface and bottom fluxes of J_{NN_T} and J_{NN_S} are set to zero, ensuring that the neural network only redistributes tracers within the water column without adding or removing them from the domain. The inputs to NN_T and NN_S are the temperature gradient, salinity gradient, potential density gradient, and gradient Richardson number of 5 grid points within the neighborhood of the grid point intended for inference, as well as the surface buoyancy flux. The information of local shear is encoded in the local Richardson number Ri while information regarding nonlinear seawater thermodynamics from TEOS-10 (Roquet et al., 2015) is explicitly encoded in the potential density gradient inputs. The nonlocality of the neural networks is reflected in their dependence on the surface buoyancy flux regardless of the location of inference within the vertical column.

The output of each neural network NN_T and NN_S consists of only a scalar value that represents the nonlocal flux at the intended point of the grid. To infer the missing entrainment fluxes at each of these grid points, the same neural networks are “convolved” along the vertical dimension, i.e., the weights of the neural networks are shared among all vertical grid points. Similarly, location-dependent inputs (neighborhood tracer gradients and Ri) change accordingly as the neural networks are “vertically convolved”. This means that neural networks are depth-independent, implicitly assuming that the entrainment physics do not depend on depth. However, it is important to note that rotational effects (which are ignored in NORi) suppress entrainment in boundary layers deeper than a kilometer as the time scale for plumes to reach the base of the boundary layer becomes comparable to the Coriolis time scale (Marshall & Schott, 1999). Such deep boundary layers are only found in regions of dense water formation at high latitudes and require a dedicated parameterization that includes both vertical mixing by convection and lateral restratification by lateral baroclinic instabilities (Boccaletti et al., 2007). NORi, like all other ocean boundary layer models, is designed to capture turbulence in low- to mid-latitude boundary layers.

Taking advantage of the fact that entrainment affects only the base of the boundary layer, the neural networks are only active within five grid points below and ten grid points above the boundary layer base, diagnosed as the nearest point from the surface where the Richardson number exceeds the critical Richardson number Ri^c . We dedicate a larger active zone above the base of the mixed layer than the bottom as we found through experimentation that this allows the entrained tracers to mix uniformly across the entire mixed layer column above the entrainment zone more effectively. The nonlocality is reflected by the neural networks’ dependence on air-sea fluxes, in addition to local gradients at the base of the boundary layer, consistent with evidence that the entrainment fluxes scale nonlocally with air-sea fluxes (Deardorff et al., 1980; Van Roekel et al., 2018). More technical details on the neural network architecture are provided in Appendix D.

5.2 Neural ODEs: online calibration to promote accuracy, generalizability, and stability

The boundary layer models consist of a system of neural ODEs (two velocity components, temperature, and salinity), i.e., ODEs with embedded neural networks. Two separate neural networks that represent the nonlocal turbulent fluxes of temperature and salinity are embedded in their respective equations. The neural ODEs (NODEs) are calibrated *a posteriori* to promote numerical stability and generalizability of the solutions,

as explained in the introduction. This means that the loss \mathcal{L} is a function of the full time history of the evolving variables rather than the subgrid fluxes. Mathematically, for some integration time from t_0 to t_1 where the variables are time-marched forward,

$$\mathcal{L}(\bar{u}, \bar{v}, \bar{T}, \bar{S}; \boldsymbol{\theta}_T, \boldsymbol{\theta}_S) = \mathcal{L}(\bar{u}(t), \bar{v}(t), \bar{T}(t), \bar{S}(t), \bar{\sigma}(t)), \quad (35)$$

where $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_S$ are the trainable weights of the neural networks, and the variables in \mathcal{L} are the full time series from t_0 to t_1 where $t \in [t_0, t_1]$. The potential density σ is included in the loss because it is a nontrivial function of temperature and salinity due to nonlinearities in the equation of state.

The neural networks are trained with *a posteriori* calibration. We utilize automatic differentiation (AD) to compute how loss changes with respect to the weights of the neural networks. AD propagates derivatives through the ODE solver, enabling gradient-based updates to parameters. In the context of the parameterization problem addressed in this study, the climate model acts as the forward model, implying that online calibration requires the climate model to be entirely differentiable. However, contemporary climate models are not differentiable. Additionally, due to its serial-in-time calibration nature, online training requires substantial computational power and memory. We describe in the following how we were able to overcome both challenges. However, first, we wish to explain why we did not resort to *a priori* calibration, which is a common strategy to simplify the calibration task.

A priori (or offline) calibration trains the neural networks to match the turbulent fluxes diagnosed from the LES as a function of the inputs with no information about the temporal evolution of the system. Mathematically, offline training for our problem would take the form

$$\mathcal{L}_{\text{offline}}(\bar{u}, \bar{v}, \bar{T}, \bar{S}; \boldsymbol{\theta}_T, \boldsymbol{\theta}_S) = \mathcal{L}(\bar{u}'\bar{w}', \bar{v}'\bar{w}', \bar{w}'\bar{T}', \bar{w}'\bar{S}'). \quad (36)$$

This approach led to numerical instabilities within a few time steps when applied to our parameterization problem. The reason is that the diagnosed LES turbulent fluxes contain noise in addition to the signal relevant to entrainment: linear internal waves generate fluctuations in all variables but have no impact on entrainment, and discretization errors are unavoidable. Although regularization techniques can be used to reduce the impact of noise contamination in training data, this is not appropriate for our purposes because the target signal, the entrainment flux, is characterized by sharp gradients at the base of the boundary layer. Any smoothing of its vertical profile would affect the stratification at the base of the boundary layer, which is a crucial variable in setting the deepening of the boundary layer itself. As a result, *a priori* calibration results in parameterizations of fluxes with a small amount of noise. This noise inevitably accumulates as the NODEs are integrated forward in time, eventually leading to a finite-time blowup.

The loss function used for *a posteriori* calibration has the same form as the one used to train the base closure and given in Equation (31) except for the omission of the momentum fields, \mathcal{L}_u^a and \mathcal{L}_v^a . Including the momentum terms in the loss function significantly deteriorates the quality of the training, because the velocity fields are dominated by inertial oscillations, which have no impact on entrainment, but in an attempt to capture them, neural networks deteriorate the quality of the predictions T and S . We also found that including vertical derivatives of tracers during optimization promotes smooth solutions free of tiny wiggles around the training data, acting as a form of regularization. Training is carried out over $N_{\text{sim}} = 62$ simulations outlined in Tables B1, B2, B3, and B4. The details of how normalization factors are calculated are discussed in Section 4 and Appendix C.

When we were trying to train NORi, there were no fully-differentiable ocean models which provides the functionality and interface required for our *a posteriori* calibration. Therefore, we wrote a custom-built, standalone implementation of the 1D boundary layer model outlined in Equations (11) through (14) in the Julia programming lan-

guage (Bezanson et al., 2017). However, it is worthwhile to note that at the time of writing, it is likely that in the near future, it will be possible to perform such training directly in Oceananigans.jl due to recent developments in its automatic differentiation capabilities (Moses et al., 2025).

The neural ODEs (11) through (14) are quite stiff (Kim et al., 2021) due to the multiscale nature of the boundary layer physics, where the diffusion terms, Coriolis terms, and neural network entrainment terms evolve on three significantly different length and time scales. Explicit time stepping schemes are thus prohibitively expensive for both the forward and backward passes. Instead, we implemented a split implicit-explicit algorithm, where the diffusion term is time stepped with the implicit Euler method and the non-local flux is time stepped with the explicit Euler method. The NODEs are time stepped by iterating over the non-dimensional, discretized versions of the NODEs with a constant $\Delta t_{\text{train}} = 10$ minutes during training and $\Delta t_{\text{validation}} = 5$ minutes during validation. (The discretized equations are provided in [Appendix E](#) and the nondimensionalization is derived in [Appendix F](#).) As we will also show in [Section 6](#), NORi is insensitive to time step size up to 1 hour, so with a training time step of 10 minutes, higher-order explicit time stepping schemes are not required.

The neural networks are built employing Lux.jl (Pal, 2023), with the Glorot method (Glorot & Bengio, 2010) used to initialize their weights. During training, both input and output data are normalized to a mean of zero and a variance of one across all training simulations. The backpropagation of the weight gradients of the neural network with respect to the loss function is accomplished through the Enzyme.jl AD tool (Moses & Churavy, 2020) in the Julia language—see Equations (E1) through (E4). The weights of the neural network are updated using the Adam optimizer (Kingma & Ba, 2017). Through a brief hyperparameter sweep over different activation functions, neural network layers, and the number of units in the hidden layers, we found that 3 hidden layers with 128 units each and a rectified linear unit (ReLU, Glorot et al., 2011) provided the best performance as computed following the loss function introduced right above. To avoid local minima due to the complex loss landscape to be sampled, we employ a “curriculum learning” strategy. This involves training the NODEs to make accurate and numerically stable predictions over progressively extended integration time frames. Integration times are increased in 3 stages with integration windows of 15-hour, 23.3-hour, and 43.3-hour while decreasing the learning rate. The numerical values of the chosen integration windows do not carry any significance apart from them being the 90th, 140th, and 260th time step given 10-minute time steps. In each stage, the NODEs are trained for 2000 epochs, and the initial model weights of each stage are selected based on the lowest training loss in the previous stage. At the beginning of each stage, the losses are renormalized, i.e. the A_ϕ^α factors in the loss function are recalculated to promote a good gradient flow towards optimal model performance. The final model weights are selected on the basis of the lowest validation loss in the final stage. We used a total of 62 training examples to train the NODEs, which is equivalent to 62 different initial and boundary conditions integrated forward in each epoch. A “full-batch gradient descent” approach (where the loss is computed across all training simulations for each gradient descent step) is taken, as we found that performing minibatching worsens loss convergence. General training takes approximately 90 hours wall time on a single CPU, trained on the Supercloud computing cluster (Reuther et al., 2018). We chose to train it on a CPU due to the more complete CPU coverage of Enzyme.jl at the time of training NORi, but in future iterations it would be useful to train natively on Oceananigans.jl with GPUs.

6 Training, validating, and comparing NORi in single column contexts

6.1 Training and validation loss

We trained NORi on 62 cases listed in Tables B1, B2, B3, and B4 and validated it on an additional 30 cases shown in Table B5 to select the final model weights. We note that the last 2 testing cases in Table B1 are not used to validate the neural network components. This is because these cases do not have convection, and in such scenarios the neural networks would not be turned on anyway. Figure 4 shows that the neural network losses decreased during the training epochs, approaching convergence after about 4,000 epochs. Training losses remain small in cases that begin with small losses, indicating that neural networks augment base closure only when there are significant residual errors. We also find that the “curriculum” learning approach, where neural networks are trained on progressively longer integration windows, is well-balanced, with the decrease in loss similar across the three stages. This is likely because the physics of entrainment within shallow mixed layers is independent of the depth of the boundary layer, so each stage provides the same training information. The validation losses follow a similar trend with epochs to the training losses, indicating that NORi interpolates well across the physical regimes it is trained on.

6.2 Inference performance

Once trained, NORi matches the LES solutions in all training cases and demonstrates strong predictive skills in the validation cases. This includes the cases where the base closure was inadequate, indicating that NORi effectively captures entrainment, which is the primary deficiency in the base closure.

We illustrate the improvement of NORi over the base closure in six representative cases—three training cases and three validation cases—in Figure 5. NORi matches the LES solutions in the training cases and approximates well the validation cases, showing substantial improvement over the base closure in simulations where the entrainment is strong. The first column of Figure 5 shows the evolution of a boundary layer subject to strong winds and cooling. The initial stratification is strong and is derived from a linear fit of the summer Argo profiles from the midlatitude Atlantic (Gaillard et al., 2016). During the first hours, a turbulent Ekman layer driven by wind develops on the surface $L_{Ek} \simeq 0.25f^{-1}\sqrt{J_u^{\text{top}}} \sim 70$ m (McWilliams, 2011). But soon afterwards convective plumes start punching deeper, resulting in substantial entrainment captured by NORi, but not by the base closure. In contrast, the third column of Figure 5 presents a simulation in which both NORi and the base closure match the LES solution. This solution is representative of the Equatorial Pacific where $f = 0$ and therefore $L_{Ek} \rightarrow \infty$ so Ekman dynamics do not apply. Despite strong evaporation, momentum shear reaches the base of the boundary layer and dominates over convection, resulting in minimal entrainment. These two examples demonstrate that NORi correctly reduces to base closure when shear dominates turbulence and deepening scales with Ri , while it improves on base closure when convection drives entrainment.

The fourth through sixth columns of Figure 5 show validation examples that were not seen during training. NORi generalizes well to these unseen scenarios, confirming that the neural network effectively interpolates within the range of air-sea fluxes and stratification values seen in training. Should new cases emerge where NORi fails to match LES simulations or observational measurements of boundary layer evolution, its high expressivity and flexibility give us confidence that it can be retrained through fine-tuning or transfer learning to capture the new physics.

Importantly, NORi works well in the nonlinear regime of the equation of state, as demonstrated in the second and fifth columns of Figure 5, where conditions similar to those of the Southern Ocean with low temperatures, destabilizing temperature strati-

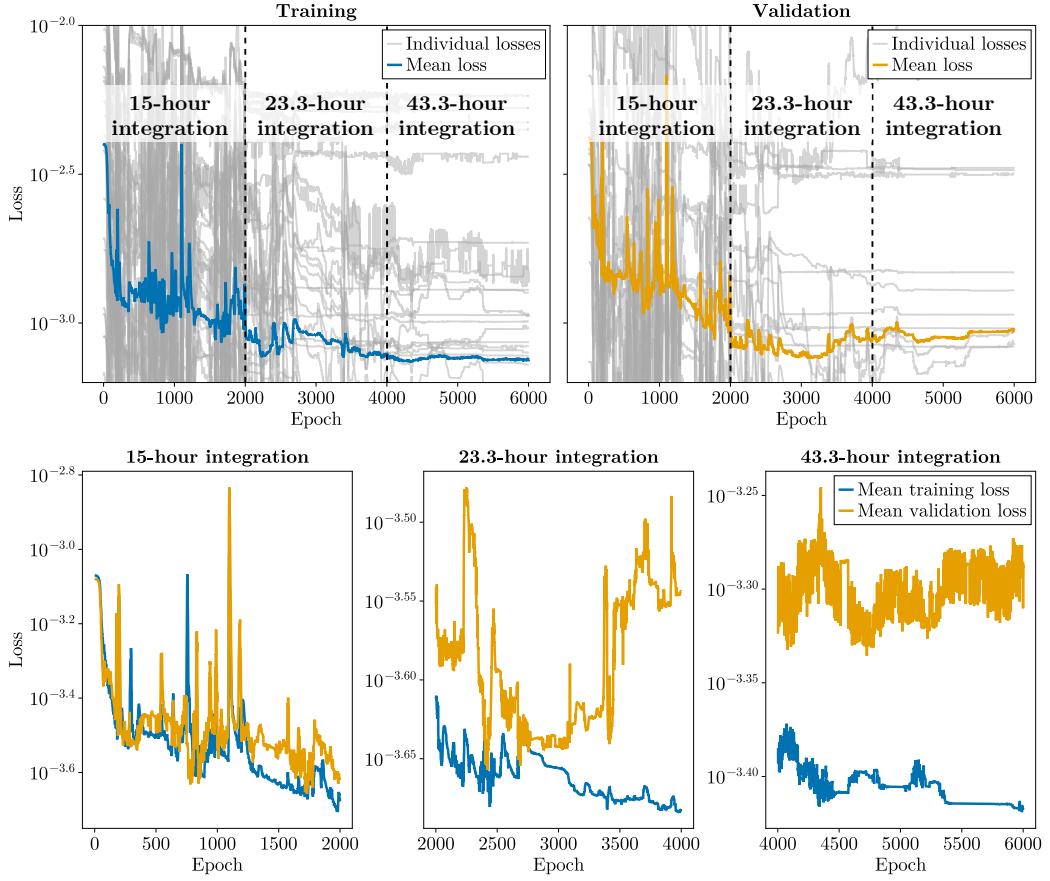


Figure 4. Training and validation losses of NORi. Top row: mean and individual losses against epoch over the final integration horizon of 43.3 hours for the training suite (Tables B2, B3, and B4) in the left panel and the validation suite (Table B5) in the right panel. All losses are normalized only once at epoch 0—an epoch is defined as one iteration over the entire training suite. The gray lines are the individual losses for each training/validation case, while the colored lines are their means. Bottom row: mean training and validation losses over the three stages of “curriculum learning,” where each stage covers a longer integration period: 15 hours, 23.3 hours, and 43.3 hours. The training loss is re-normalized at the start of each stage, as explained in Appendix C. The epoch with the lowest training loss at each stage is used to initialize network weights for the next stage. The final model weights are selected from the model with the lowest validation loss in the final stage.

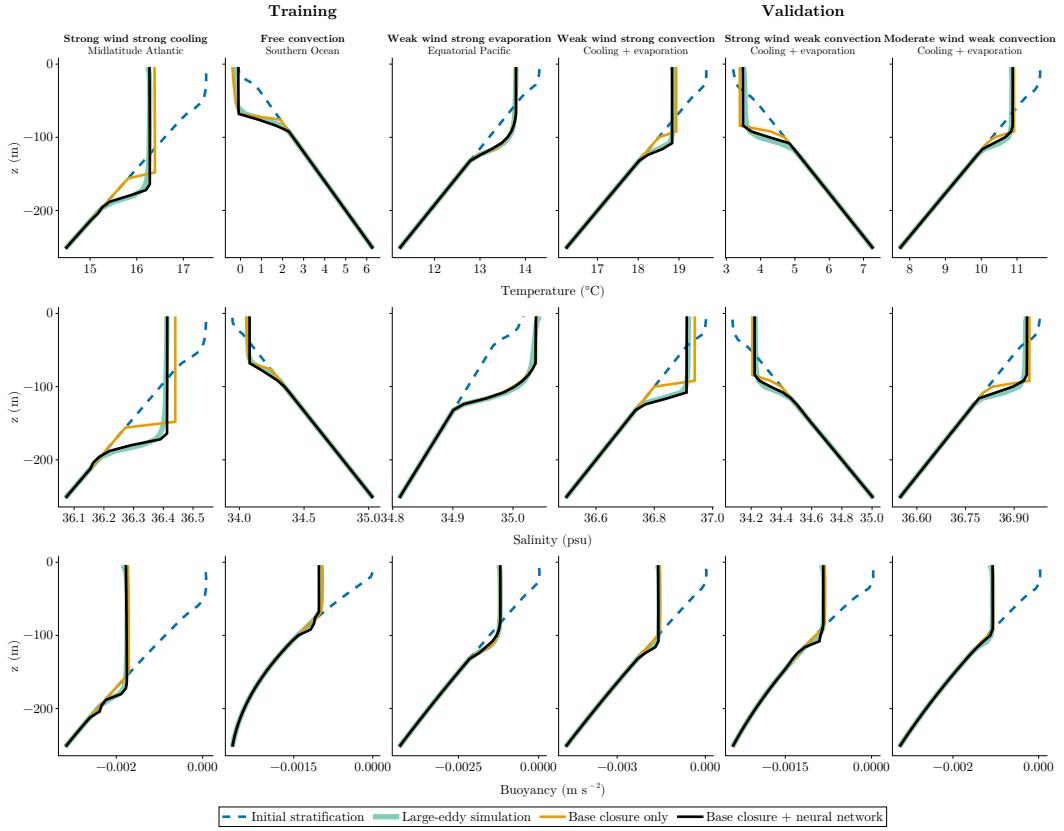


Figure 5. Temperature, salinity, and buoyancy profiles for selected training (columns 1 through 3) and validation (columns 4 through 6) examples generated with the calibrated NORi model (black lines), the base closure (orange lines) as well as area-averaged large-eddy simulation (LES) solutions (green lines). The profiles are computed 1.75 days after the initial conditions (dashed lines).

fications, and salinity-dominated buoyancy structures are illustrated. The initial temperature and salinity stratifications are linear with depth, but the resulting buoyancy is not. NORi captures the entrainment of temperature, salinity, and buoyancy, which behave differently due to the nonlinear equation state and are missed by the base closure.

It is worth remarking that the base closure matches the buoyancy LES profiles much better than those of temperature and salinity, which compensate strongly in these examples. This supports our argument that boundary layer parameterization must be validated with solutions and observations of both temperature and salinity profiles, rather than just buoyancy as is often done in the literature.

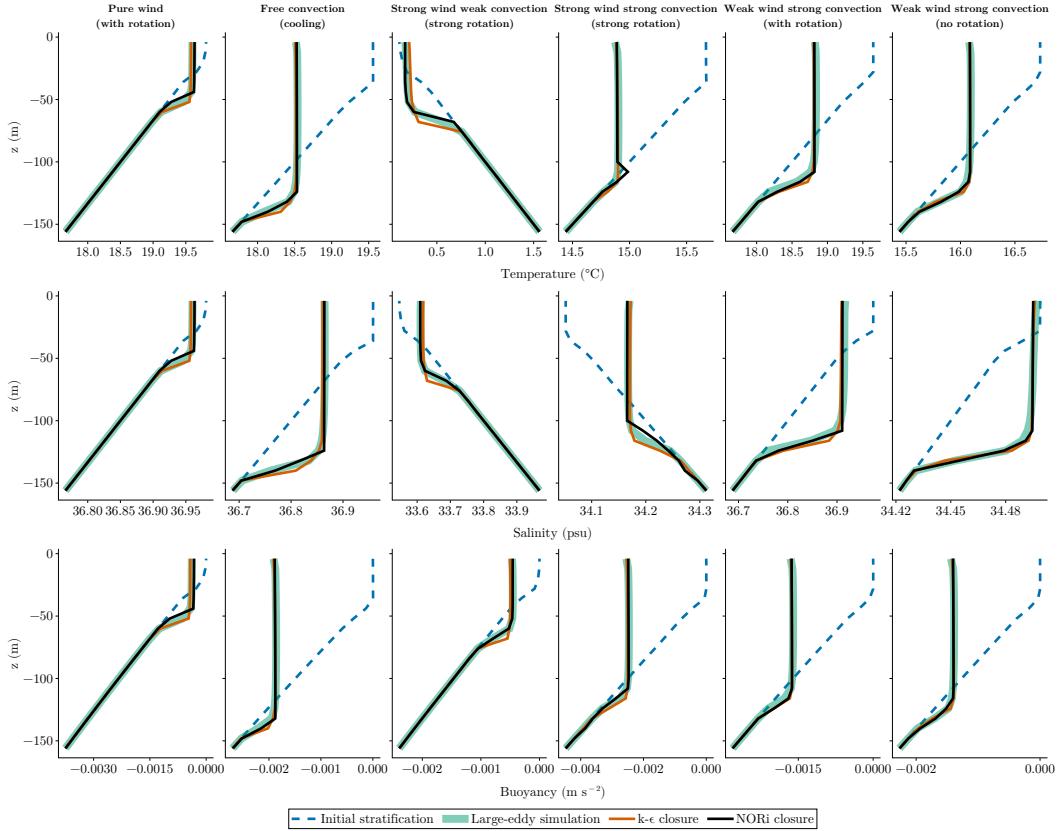


Figure 6. Comparison of NORi versus k - ϵ performance in a column model setting. The model profiles of temperature, salinity, and buoyancy are plotted together with the area-averaged LES profiles. The examples shown are samples of validation cases not seen by NORi during training (see Table B5 for a complete list). The profiles are computed 1.75 days after the initial conditions (dashed lines).

So far, we have focused on demonstrating that NORi has the skill to match LES solutions of boundary layer turbulence. In Figure 6 we further show that NORi compares very favorably with a state-of-the-art boundary layer closure. The k - ϵ model (Umlauf & Burchard, 2003) is a second-order turbulence model that solves two additional prognostic equations for the turbulent kinetic energy k and its dissipation ϵ to predict the turbulent fluxes $\overline{u'w'}$, $\overline{v'w'}$, $\overline{w'T'}$, and $\overline{w'S'}$. Despite its much simpler formulation, NORi, a leaner first-order model, produces single-column solutions that are practically indistinguishable from k - ϵ and the LES solutions within the validation suite used in this work.

An additional favorable comparison of NORi against CATKE (Wagner, Hillier, et al., 2025), a recently-developed 1.5-order closure that estimates turbulent fluxes using only an additional prognostic equation for turbulent kinetic energy (TKE) is shown in Figure H1 in [Appendix H](#). This makes NORi a compelling alternative in scenarios for coarse-resolution, long-time step simulations where lower-order models are preferred. If we wish to apply NORi to high-resolution, short-time step simulations, it is possible to extend NORi to higher orders by solving additional prognostic equations for turbulent quantities as well.

Before testing NORi in more general settings, we next evaluate NORi’s numerical stability, an important consideration in neural network-based models, and its sensitivity to the time step used to march the boundary layer equations forward.

6.3 Numerical stability

To assess whether NORi is numerically stable on time scales longer than the training horizon of 2 days, we run NORi for 60 days in a single 768 m deep column with a constant grid resolution of 8 m and a time step of 5 minutes. The simulation is initialized with a constant temperature gradient of $\frac{\partial T_0}{\partial z} = 9.77 \times 10^{-3} \text{ }^{\circ}\text{C m}^{-1}$ and a salinity gradient of $\frac{\partial S_0}{\partial z} = 1.46 \times 10^{-3} \text{ psu m}^{-1}$, and surface values $T_0(z=0) = 30 \text{ }^{\circ}\text{C}$ and $S_0(z=0) = 37 \text{ psu}$. Surface forcing is imposed through a constant momentum flux of $J_u^{\text{top}} = -1 \times 10^{-4} \text{ m}^2 \text{ s}^{-2}$ and time-dependent sinusoidal temperature and salinity fluxes

$$J_T^{\text{top}} = \left[2 \cos \left(\frac{2\pi}{1 \text{ day}} t \right) + 1 \right] \times 10^{-4} \text{ }^{\circ}\text{C m s}^{-1}, \quad (37)$$

$$J_S^{\text{top}} = -2 \cos \left(\frac{2\pi}{2.63158 \text{ days}} t \right) \times 10^{-5} \text{ psu m s}^{-1}, \quad (38)$$

with a Coriolis parameter of $f = 1 \times 10^{-4} \text{ s}^{-1}$. The “magic number” 2.63158 days is chosen ad-hoc so that when combined, the temperature and salinity fluxes result in irregular fluctuations in the surface buoyancy flux, primarily destabilizing, but occasionally stabilizing. Because they have different periods, the temperature and salinity fluxes are not in phase. At certain times they act in opposite directions (one stabilizing, another destabilizing), while at other times they are jointly stabilizing or destabilizing. So, in addition to testing numerical stability, we also test whether NORi can simulate the evolution of a boundary layer under time-dependent air-sea fluxes despite being trained only in settings with constant air-sea fluxes. NORi correctly turns on the neural network to capture entrainment only when the buoyancy flux drives convection (destabilizing).

To test whether NORi is not only stable but also accurate on long time scales, in [Figure 7](#) we compare the NORi profiles of the velocity, temperature, and salinity fields at 60 days with the profiles of an LES simulation and the k - ϵ model forced with the same time-dependent air-sea fluxes. We note that the momentum components u and v can be highly noisy in LES due to internal oscillations that do not contribute to entrainment, so the analysis of results against LES should focus on the temperature and salinity fields. This also illuminates why the loss function used to train NORi does not include momentum terms (see our discussion in [Section 5](#)).

The profiles are quite similar, but differences do emerge: the k - ϵ model produces a mixed layer that is around 50 m deeper than that of NORi by the end of 60 days. Comparing both models with the LES solution, we see that NORi is slightly undermixed while k - ϵ is slightly overmixed, where the biases are around 25 m in each direction. This suggests that the NORi representation of the entrainment may need further tuning over longer integration times. However, the k - ϵ model has not been tuned against the same suite of LES simulations used to train NORi, so any statement about its capabilities would not be entirely fair. That said, this experiment shows that despite NORi and k - ϵ being virtually indistinguishable over short time scales (see [Figure 6](#)), tiny biases will accumu-

late, leading to differences over longer time scales. Although these biases might appear minor at 60 days, they are in fact of great significance, as ocean models are typically integrated for at least decades.

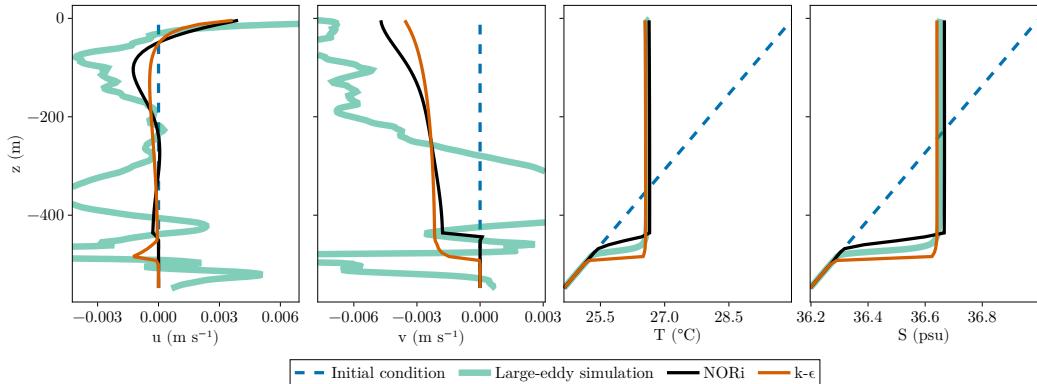


Figure 7. Velocity, temperature, and salinity profiles 60 days after initialization (dashed lines) for simulations forced with a constant wind stress, $J_u^{\text{top}} = -1 \times 10^{-4} \text{ m}^2 \text{ s}^{-2}$ and time-dependent heat and freshwater fluxes given by Equations (37) and (38). The Coriolis parameter is set to $f = 1 \times 10^{-4} \text{ s}^{-1}$. Solutions are reported from the NORi and k - ϵ models together with the area-averaged LES.

6.4 Time step dependence

An important consideration in the development of parameterizations for large-scale ocean models is the time step required to get stable and accurate solutions. To address this point, we run NORi and the k - ϵ model with increasing time steps from 1 minute to 2 hours. To focus on the limitations of the time steps associated with the parameterizations and not with the inertial oscillations, we set $f = 0$. The simulations are initialized with a temperature gradient of $\frac{\partial T_0}{\partial z} = 1.5 \times 10^{-2} \text{ }^{\circ}\text{C m}^{-1}$ and a salinity gradient of $\frac{\partial S_0}{\partial z} = 2 \times 10^{-3} \text{ psu m}^{-1}$, with surface values $T_0(z = 0) = 20 \text{ }^{\circ}\text{C}$ and $S_0(z = 0) = 37 \text{ psu}$. The surface fluxes are set to $J_u^{\text{top}} = -1 \times 10^{-4} \text{ m}^2 \text{ s}^{-2}$, $J_T^{\text{top}} = 2 \times 10^{-4} \text{ }^{\circ}\text{C m s}^{-1}$, and $J_S^{\text{top}} = -2 \times 10^{-5} \text{ psu m s}^{-1}$, respectively. The models are integrated for 4 days. Figures 8 and 9 show the final velocity, temperature and salinity profiles for the different time steps computed using the NORi closure and the k - ϵ closure.

As shown in Figure 8, the NORi solutions are independent of time step if shorter than 1 hour. The solution, instead, begins to deviate from the true solution when the time step is increased to 2 hours; the mixed layer becomes too deep because NORi begins to overestimate mixing. In contrast, the model k - ϵ shows time step dependence even below 1 hour. Interestingly, k - ϵ tends to underestimate the mixing as the time step increases. This is unsurprising as it is well known that higher-order boundary layer models, such as k - ϵ , require shorter time steps than their lower-order counterparts, and that is the main reason why they are not commonly used in large-scale coarse resolution ocean models that require longer time steps (Reichl & Hallberg, 2018; Wagner, Hillier, et al., 2025). For NORi, with 1 hour time steps, we find that the solutions of T and S remain accurate compared to the LES solution, indicating that NORi can be used with time steps up to 1 hour without loss of accuracy. The same could also be said for k - ϵ at or below 30 minutes time steps. Additionally, regardless of time steps, the velocity profiles of both NORi and k - ϵ show greater deviations from the LES solution. However, velocities are

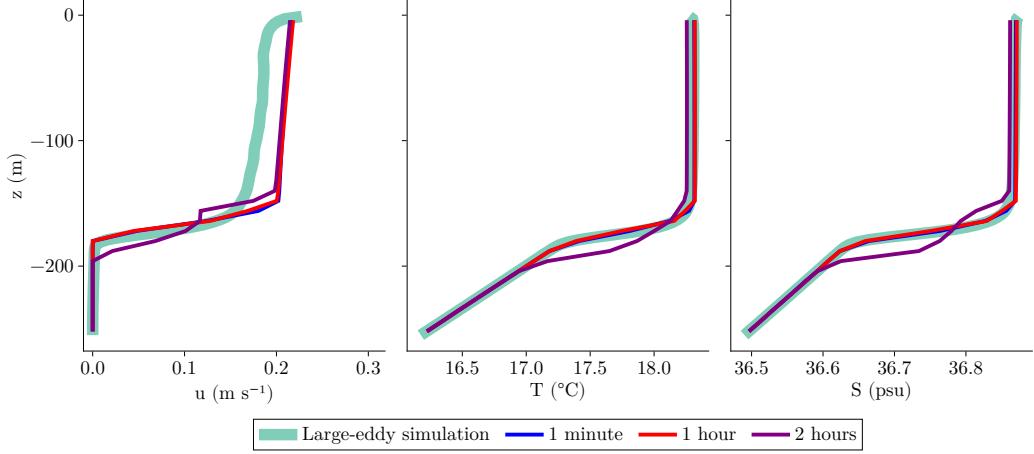


Figure 8. Velocity, temperature, and salinity profiles 4 days after initialization generated with NORi using a range of time steps. The surface forcing parameters are given in the text. There is no v velocity component, because the simulation is not rotating and the surface momentum stress acts in the x -direction. The LES solution is also shown for reference.

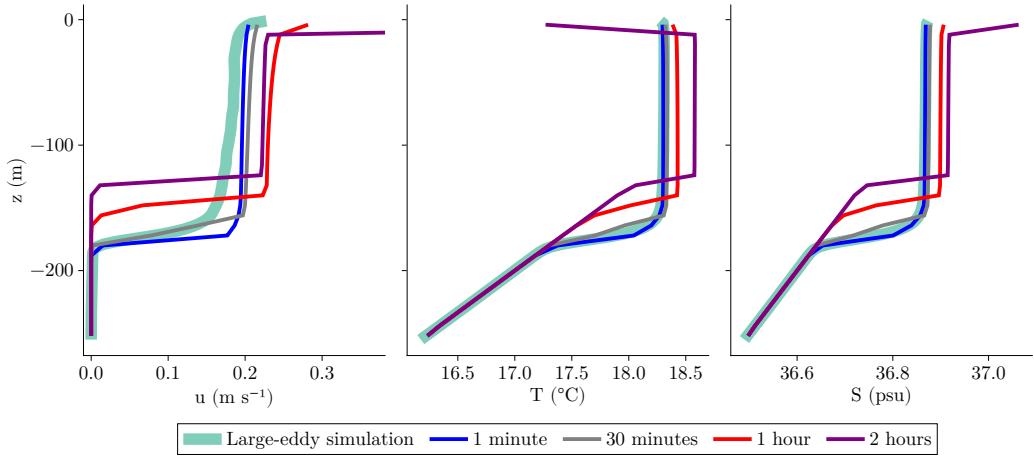


Figure 9. Same as Figure 8 but generated with the k - ϵ model.

less important for capturing the vertical mixing processes in which we are primarily interested, since both models can capture the stratification accurately despite not getting momentum entirely right.

In a highly unoptimized, naive first implementation of NORi in Oceananigans.jl where data is copied to and from the working arrays for neural network inference, NORi is around 15% slower than k - ϵ on a per-time step basis. Given that NORi remains accurate at time steps more than twice as long as k - ϵ , we conclude that there are computational advantages to using NORi over higher-order models in large-scale simulations. At the same time, we have seen in Figure 7 that this set of trained NORi weights produces biases that are of the same magnitude as k - ϵ when compared to LES in our simulation over 60 days, making the case for using NORi in large-scale simulations even more compelling.

7 Testing NORi on a large-scale, centennial double-gyre simulation

To test NORi’s skill in large-scale simulations, an idealized double-gyre simulation is set up that includes many key components of realistic ocean simulations. Implementing NORi in Oceananigans.jl was straightforward; it is only necessary to set up input and output fields for neural network inference, as the neural network package Lux.jl easily integrates into Oceananigans.jl without the need for intermediate wrappers. NORi works flexibly with both CPU and GPU backends in Oceananigans.jl, allowing us to take advantage of hardware acceleration for large-scale simulations.

The double-gyre has dimensions of $(L_x, L_y, L_z) = (4000, 6000, 1.6)$ km with a grid size of $(N_x, N_y, N_z) = (100, 100, 200)$, which fits on a single NVIDIA V100 GPU. We do not include any other parameterizations in order to isolate and observe only the effects of boundary layer schemes. The setup uses a β -plane approximation with $f = 2\Omega \sin 45^\circ \approx 1.0 \times 10^{-4} \text{ s}^{-1}$ at the center of the domain and $\beta = \frac{2\Omega \cos 45^\circ}{R_{\text{Earth}}} \approx 1.6 \times 10^{-11} \text{ m}^{-1} \text{ s}^{-1}$, where Ω is the rotation rate of the Earth and R_{Earth} is the radius of the Earth. The meridional extent of the gyre is 6000 km and corresponds approximately to 54° in latitude. Forcing consists of restoring on an 8 day time scale of temperature and salinity to linear profiles decreasing with latitude from 30°C to 0°C and from 37 psu to 34 psu, respectively (see Figure 10). Momentum forcing is through a sinusoidal wind stress of magnitude $1 \times 10^{-4} \text{ m}^2 \text{ s}^{-2} \approx 0.1 \text{ N m}^{-2}$ driving westerlies north and south and easterlies at the center latitude as shown in Figure 10. A linear drag with a damping time scale of 30 days is imposed at the deepest grid level. The double-gyre simulation is run for 100 years with a constant time step of 5 minutes, chosen short enough to compare the results with the NORi and k - ϵ boundary layer models. The simulation is initialized with a uniform temperature field in the horizontal direction and with a linear vertical gradient decreasing from 30°C at the top to 10°C at the bottom. The salinity field is initialized with no vertical gradient but with a meridional linear gradient decreasing from 37 psu to 34 psu with latitude. Figure 10 shows the equilibrated 3D buoyancy field, a horizontal temperature slice at a depth of 164 m, and the vertically-integrated barotropic streamfunction Ψ averaged over 10 years defined as

$$\frac{\partial \Psi}{\partial y} = - \int_{-L_z}^0 u dz, \quad \frac{\partial \Psi}{\partial x} = \int_{-L_z}^0 v dz, \quad (39)$$

$$\bar{\Psi} = \frac{1}{10 \text{ years}} \int_{\text{year 90}}^{\text{year 100}} \Psi dt. \quad (40)$$

The barotropic streamfunction is characterized by two intensified gyres on the west. A thermocline develops in the southern half of the domain consistent with the Ventilated Thermocline Theory (Pedlosky, 1996; Vallis, 2017). Deep convection generates a well-mixed water column at the northern edge of the domain.

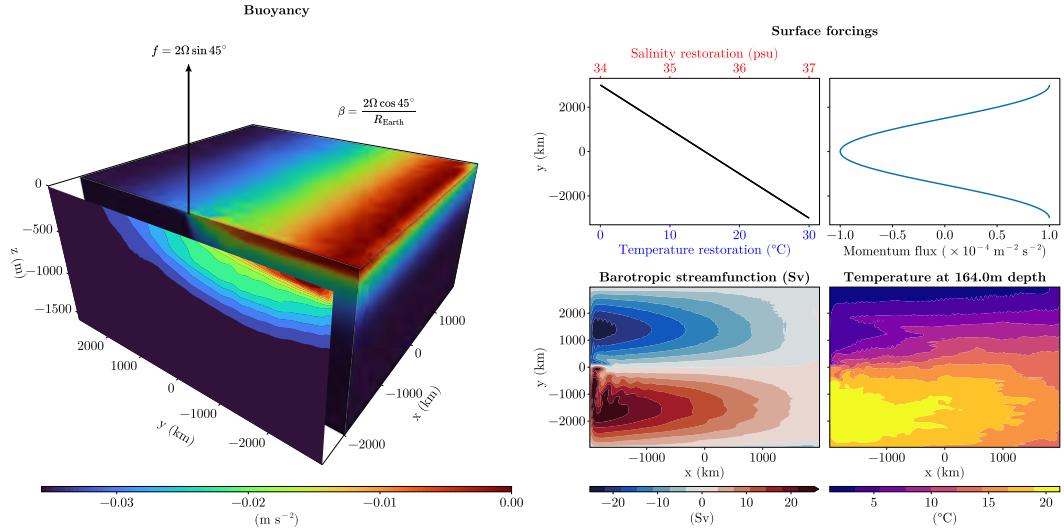


Figure 10. The configuration of the double-gyre simulation. The results shown in this figure are produced using the NORi closure implemented in Oceananigans.jl, run on GPUs, taken at $t = 100$ years. The subfigure on the left shows a 3D snapshot of the buoyancy field as well as the zonally-averaged buoyancy of the simulation. The subfigures on the right show the surface temperature, salinity surface restoring profiles (top left), the surface wind stress (top right), the barotropic streamfunction averaged over the last 10 years (bottom left), and a horizontal temperature slice at a depth of 164 m (bottom right). The barotropic streamfunction reaches a maximum of around 40 Sv in the subtropical gyre.

We run simulations using NORi, the base closure model, and the k - ϵ model. In this naive first implementation of NORi in Oceananigans.jl, we allocated additional working arrays to store the inputs and outputs of the neural networks; thus a neural network inference is a forward pass over the input working array. Admittedly, this is a highly inefficient approach as it incurs severe memory overhead as well as data duplications. However, this simple implementation has already yielded a performance that is barely 15% slower than k - ϵ on a per-time step basis. Combining a more efficient implementation with the larger time steps with which NORi can be run (see Figures 8 and 9), NORi has the potential to be significantly faster than higher-order closures when deployed in large-scale simulations.

NORi is stable for the full 100 years of simulation despite being trained with an integration time of less than 2 days. In the many different setups we have run during experimentation, NORi has never once crashed due to numerical instabilities. Figure 11 compares the zonally-averaged temperature at $t = 2.5$ years from double-gyre simulations using the three boundary layer models. NORi simulates deeper mixed layers than the base closure as expected, because the base closure lacks the representation of entrainment. Instead, NORi produces shallower mixed layers than k - ϵ , especially in the northern part of the domain. Interestingly, this bias pattern is the same as what we observed in the single-column comparison at 60 days with LES in Figure 7. Although NORi and k - ϵ produce very similar solutions over the 2-day training periods, important differences can emerge over longer time periods, as also discussed in Section 6.3. However, the discrepancies in this case may also arise from interactions between the parameterizations and the large-scale ocean dynamics. Figure G1 in Appendix G confirms that these discrepancies persist in time by showing the same zonally-averaged temperature after 100 years. In Appendix H we further show that NORi generates deeper mixed layers than CATKE (Wagner,

Hillier, et al., 2025) at 100 years, suggesting that more studies are needed to determine which model is more accurate.

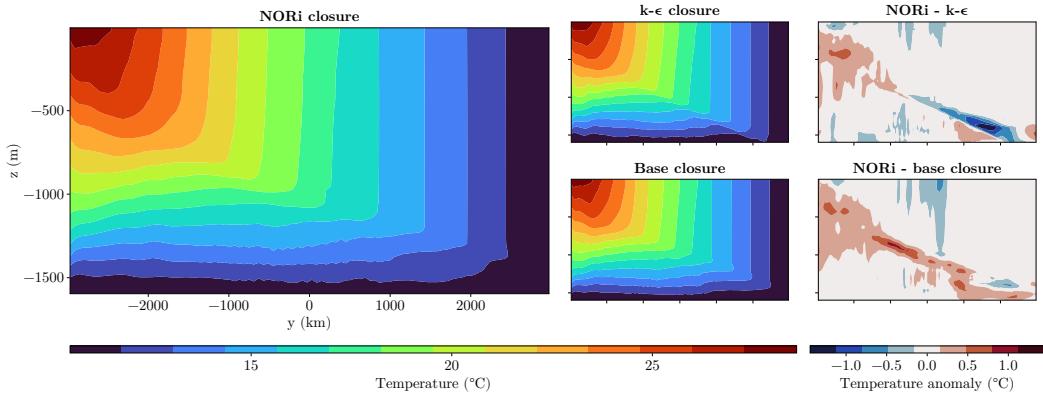


Figure 11. Zonally-averaged temperature slice from simulations using NORi, $k-\epsilon$, and the base closure in the double-gyre simulation. The snapshots above are taken at $t = 2.5$ years. The rightmost column shows the temperature differences between NORi and the other two models.

The results of the double-gyre solutions highlight the importance of recalibrating boundary layer parameterizations in global simulations with climatological observations, because the small residuals in the calibration with LES simulations can build up over longer times and larger scales. However, calibration with short-term, high-resolution simulations or observations provides essential initial estimates for the model parameters, reducing the amount of data required for calibration in the global context.

8 Conclusion

We introduced NORi, a parameterization of oceanic surface boundary layer turbulence formulated with neural ODEs where neural networks augment a simple physical closure. Some important overarching themes guided the development, iteration, and calibration of NORi. We recognized that to overcome the triple conundrum of high numerical stability, fast model inference, and relatively cheap training, the neural networks, as highly expressive as they are, must be constrained by our physical understanding of the vertical mixing process. With many more free parameters and training data than what we attempted, using a purely neural network approach may yield higher model accuracy and generalization. However, the computational cost of generating the training data and performing inference renders this approach unrealistic. It is well understood that incorporating physical knowledge into model design drastically reduces the amount of data required and the size of the model (Goyal & Bengio, 2022). Through numerous experiments, we designed a physical closure that is “simple but not any simpler”. Layered on top of it are highly expressive neural networks that capture more complicated processes, where good representations have eluded traditional parameterizations for decades. Specifically, in the context of NORi, we used neural networks to capture nonlocal entrainment fluxes which are missing from the local eddy-diffusivity closure.

During the design process of NORi, we paid special attention to ensuring that every choice is physically principled. Since neural networks predict fluxes instead of tendencies, NORi satisfies conservation laws to machine precision by construction. The non-linear thermodynamics of seawater are explicitly incorporated, thus NORi can be read-

ily deployed in realistic ocean simulations. By sharing the same neural network weights across all vertical levels, we ensure that the entrainment dynamics of NORi are independent of mixed layer depth. However, this also means that NORi cannot capture deep convection dynamics (see our discussion in Section 5.1), but a simple extension of including the mixed layer depth in the neural network inputs can be used to address this limitation. To reduce inference cost, the neural networks in NORi are relatively lean, with only three hidden layers and 128 neurons per hidden layer. With further work on distillation and pruning, the neural networks could be made even smaller without a significant loss in accuracy. NORi has excellent training and generalization performance, as it is able to accurately predict the depth of the mixed layer in a variety of realistic oceanic forcings across a wide range of wind, cooling, evaporation, and rotational strengths under various stratifications with different thermodynamic regimes. NORi is also time step-independent below 1 hour, which is appropriate for large-scale, coarse-resolution ocean simulations.

Good calibration techniques go hand-in-hand with good model design, since the right calibration techniques promote desirable model features beyond the direct dependencies within the loss function. From the outset, we designed NORi within the *a posteriori* paradigm, where the loss function explicitly depends on the entire time history of coarse-grained variables rather than on instantaneous turbulent fluxes alone, without considering the resulting trajectories. This ensures that we optimize the model directly to our desired target. By using the neural ODE approach, we also explicitly optimized for numerical stability, since models which are numerically unstable will incur a high loss and are thus avoided. Using this approach, NORi is numerically stable for at least 100 years, despite being trained on LES with timespans of roughly 2 days. In the many double-gyre configurations we have explored during experimentation, NORi has never once crashed due to numerical instabilities. However, in pursuit of numerical stability, we have made sacrifices in the scalability of the calibration process because of the serial-in-time nature of neural ODEs. Furthermore, we speculate that the loss landscape for *a posteriori* training is likely to be noisier than *a priori* training, making it less amenable to gradient-based optimization which could lead to worse training outcomes. Despite these drawbacks, we have found that *a posteriori* training is paramount for building a model that is usable in practice, as training neural networks directly on instantaneous turbulent fluxes leads to models which are highly numerically unstable and exhibit finite time blowup. This also means that models pre-trained on instantaneous fluxes were not good priors for *a posteriori* training. Without good priors, ensemble-based methods are likely to struggle as well when we cannot afford sufficient ensemble members to explore the entire parameter space. Eventually, we found that providing more structure and physical inductive biases in the model design seems to be essential for improving training outcomes.

We have designed NORi to be relatively straightforward to implement and deploy in GPU-based ocean models. Using the Julia software stack, we were able to integrate NORi into Oceananigans.jl with minimal effort and no intermediate wrappers. Owing to the GPU-oriented design of Oceananigans.jl, neural network inference is straightforward and efficient as we do not require data transfers between CPU and GPU. However, in our first naive implementation of NORi in Oceananigans.jl, additional working arrays are allocated for neural network inputs and outputs, leading to memory-bound performance due to large transfers of values between working arrays and native Oceananigans.jl fields. Despite such inefficiencies, NORi’s performance is only 15 % slower than the state-of-the-art k - ϵ closure on a per-time step basis. A more efficient implementation of NORi would involve writing custom forward pass kernels that access native Oceananigans.jl arrays directly, thereby eliminating the need for large working arrays. This performance engineering task should be done in the future once NORi is fully production-ready.

There remain a few important challenges not addressed in this study that are important to address in the next step. A major caveat in this work is that NORi has been

trained and deployed with a fixed grid resolution of 8 m. For NORi to be flexibly used in large-scale ocean simulations, it has to be able to support variable vertical grid sizes. The most straightforward approach is to interpolate between the dynamics grid and the physics grid, running NORi inference at a fixed 8 m resolution and then interpolating the fluxes back to the dynamics grid which is coarser and potentially variable, similar to that of Molod (2009). A first attempt at this approach has been carried out using simple linear interpolation, but the results were unsatisfactory. Unsurprisingly, entrainment fluxes weaken with coarser vertical resolutions, leading to shallower mixed layers. This is because a coarser grid is less able to support the sharp gradients at the base of the mixed layer characteristic of entrainment. Thus, more sophisticated reconstruction techniques or a resolution-independent representation of entrainment fluxes may be potential solutions.

After variable vertical grid support is implemented and validated, we can then run NORi in a regional and/or global ocean simulation with realistic forcings, setups that are verifiable against observations. This has to be done in order to assess whether NORi is successful in alleviating known biases in the mixed layer depth of large-scale climate simulations. Ultimately, we want to calibrate NORi *a posteriori* within the regional and/or global ocean simulation contexts, where the model is initialized, forced, and verified against observations. This would allow for the fine-tuning of NORi to account for biases found in large-scale simulations, which are not present in idealized column setups. Importantly, the neural network weights we have learned so far will serve as good priors, which are paramount for *a posteriori* calibration given the complexity of the forward model. Extending this holistic approach, a thorough and unbiased comparison between current state-of-the-art boundary layer parameterizations calibrated against a common suite of ocean scenarios can be carried out to rigorously assess their respective biases.

So far, we have only focused on the deepening phases of the mixed layer, which occur when the surface buoyancy fluxes are destabilizing. Restratiification during the warming phases due to submesoscale processes that shallow the mixed layer have been ignored as they are governed by processes at very different length scales than those that deepen it. In the future, we aim to extend NORi to include restratiification processes within a similar framework, thereby allowing for a more complete representation of the mixed layer dynamics across the entire seasonal cycle.

Acknowledgments

This project is supported by Schmidt Sciences, LLC. X. K. L. is also supported by the Norman C. Rasmussen Fellowship and the Callahan-Dee Fellowship. This work was initiated while X. K. L. was an undergraduate student in the Department of Physics at Imperial College London under the Imperial-MIT student exchange program, working with A. R. and J. M.

Appendix A Column model derivation of the ocean surface boundary layer

For easier reference, we first rewrite equations (1), (2), (3), and (4) into its component form:

$$\frac{\partial u}{\partial t} = -\nabla \cdot (\mathbf{u}u) + fv - \partial_x p, \quad (\text{A1})$$

$$\frac{\partial v}{\partial t} = -\nabla \cdot (\mathbf{u}v) - fu - \partial_y p, \quad (\text{A2})$$

$$\frac{\partial w}{\partial t} = -\nabla \cdot (\mathbf{u}w) - \partial_z p + b, \quad (\text{A3})$$

$$\frac{\partial T}{\partial t} = -\nabla \cdot (\mathbf{u}T), \quad (\text{A4})$$

$$\frac{\partial S}{\partial t} = -\nabla \cdot (\mathbf{u}S), \quad (\text{A5})$$

$$\nabla \cdot \mathbf{u} = 0. \quad (\text{A6})$$

We again note that by decomposing a field into its horizontally-averaged component and perturbations away from the average $\phi = \bar{\phi}(z, t) + \phi'(x, y, z, t)$, in a laterally doubly-periodic domain, the horizontal averaging operator $(\bar{\cdot})$ has the property that

$$\frac{\partial \bar{\phi}}{\partial x} = \frac{\partial \bar{\phi}}{\partial x} = 0, \quad (\text{A7})$$

$$\frac{\partial \bar{\phi}}{\partial y} = \frac{\partial \bar{\phi}}{\partial y} = 0, \quad (\text{A8})$$

$$\bar{\phi}' \equiv 0. \quad (\text{A9})$$

By applying the horizontal averaging to Equation (A6) and noting properties (A7) and (A8), we obtain

$$\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} = 0, \quad (\text{A10})$$

$$\frac{\partial \bar{w}}{\partial z} = 0, \quad (\text{A11})$$

$$\therefore \bar{w} = C \quad (\text{A12})$$

where C is a constant. We now apply the horizontal averaging operator to the impenetrability boundary condition at the surface $z = 0$, i.e., Equation (7), yielding

$$\bar{w}(z = 0) = 0, \quad (\text{A13})$$

$$\therefore \bar{w} = C = 0, \quad (\text{A14})$$

since \bar{w} is a constant everywhere in the column.

Expanding the terms in Equation (A1) and applying the horizontal averaging operator, we have (in Cartesian coordinates)

$$\frac{\partial \bar{u}}{\partial t} = -\partial_x \bar{u} \bar{u} - \partial_y \bar{v} \bar{u} - \partial_z \bar{w} \bar{u} + f \bar{v} - \partial_x \bar{p}, \quad (\text{A15})$$

$$= -\frac{\partial}{\partial z} \overline{\left(\bar{u} \bar{w} + u' \bar{w} + \bar{u} w' + u' w' \right)} + f \bar{v}, \quad (\text{A16})$$

$$= -\frac{\partial}{\partial z} \overline{\left(\bar{u} \bar{w} + \bar{u}' w' \right)} + f \bar{v}, \quad (\text{A17})$$

$$= -\frac{\partial \bar{u}' w'}{\partial z} + f \bar{v}, \quad (\text{A18})$$

where $\overline{u'w'}$ is the vertical flux of horizontal momentum in the x -direction, arriving at Equation (11). Similarly, we can apply the same procedure above to Equations (A2), (A4), and (A5) to arrive at Equations (12), (13), and (14) respectively.

Appendix B Training and testing suite

In this section, we provide the complete training and testing suite used to train, validate and test NORi. Table B1 shows the training cases driven by shear instability (wind-driven mixing) and two testing cases used to evaluate the base closure, Table B2 shows the training free convection cases driven by buoyancy instability (no winds), Table B3 shows the training cases driven by both shear and convective instabilities, while Table B4 shows the remaining miscellaneous training cases that include cases with the same forcings and stratifications as some of those before but in different thermodynamic regimes (lower temperatures) as well as cases with no temperature or salinity gradient. Finally, Table B5 shows the validation cases used to select the final weights of the neural network after training. Note that $J_v^{\text{top}} = 0$ in all simulations as imposing nonzero J_v^{top} is dynamically equivalent to imposing nonzero J_u^{top} during training and inference since momenta is only implicitly provided to NORi via the Richardson number.

J_T^{top} ($^{\circ}\text{C m s}^{-1}$)	J_S^{top} (psu m s^{-1})	J_u^{top} ($\text{m}^2 \text{s}^{-2}$)	$\frac{dT}{dz}$ ($^{\circ}\text{C m}^{-1}$)	$\frac{dS}{dz}$ (psu m^{-1})	f (s^{-1})	T^{top} ($^{\circ}\text{C}$)	S^{top} (psu)
0	0	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	0	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	0	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	0	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	0	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	1.25×10^{-4}	0	33.9
0	0	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	1.25×10^{-4}	0	33.9
-3×10^{-4}	3×10^{-5}	-5×10^{-4}	1.5×10^{-2}	1.5×10^{-3}	1×10^{-4}	20	37
0	0	-3.5×10^{-4}	1.5×10^{-2}	1.5×10^{-3}	1×10^{-4}	20	37

Table B1. Cases driven by shear instability used to train and evaluate NORi. The last two rows in the table denote testing cases for the base closure.

Appendix C Loss scalings

As seen in Equations (30) and (31), during training of both the base closure and neural networks, we use a weighted loss function to balance the contributions of each variable using normalization factors A_{ϕ}^a for the field ϕ in simulation a within the training suite. The difference between the loss functions for the base closure and neural network training is that velocity components u and v are not included in the neural network training loss as these fields are dominated by inertial oscillations which do not affect vertical mixing. The values of A_{ϕ}^a are recalculated at the beginning of each training stage to renormalize the training loss. A_{ϕ}^a is dependent on the relative contribution of temperature and salinity to the potential density in the initial condition of the neural ODEs, as well as the relative contribution of each loss component to the total loss. A_T^a and A_S^a are calculated in this way:

$$A_T^a = \frac{\alpha \|\Delta T_0^a\|_{\infty} + \beta \|\Delta S_0^a\|_{\infty}}{\alpha \|\Delta T_0^a\|_{\infty}}, \quad (\text{C1})$$

$$A_S^a = \frac{\alpha \|\Delta T_0^a\|_{\infty} + \beta \|\Delta S_0^a\|_{\infty}}{\beta \|\Delta S_0^a\|_{\infty}} \quad (\text{C2})$$

J_T^{top} ($^{\circ}\text{C m s}^{-1}$)	J_S^{top} (psu m s^{-1})	J_u^{top} ($\text{m}^2 \text{s}^{-2}$)	$\frac{dT}{dz}$ ($^{\circ}\text{C m}^{-1}$)	$\frac{dS}{dz}$ (psu m^{-1})	f (s^{-1})	T^{top} ($^{\circ}\text{C}$)	S^{top} (psu)
5×10^{-4}	0	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
1×10^{-4}	0	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-5×10^{-5}	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-1×10^{-5}	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
3×10^{-4}	-3×10^{-5}	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
3×10^{-4}	-3×10^{-5}	0	1.4×10^{-2}	-2.1×10^{-3}	8×10^{-5}	18	36.6
1×10^{-4}	-1×10^{-5}	0	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	10	36.6
5×10^{-4}	0	0	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
1×10^{-4}	0	0	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-5×10^{-5}	0	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-1×10^{-5}	0	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
3×10^{-4}	-3×10^{-5}	0	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
3×10^{-4}	-3×10^{-5}	0	1.3×10^{-2}	-7.5×10^{-4}	0	14.5	35
1×10^{-4}	-1×10^{-5}	0	1.3×10^{-2}	7.5×10^{-4}	0	10	35
5×10^{-4}	0	0	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
1×10^{-4}	0	0	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-5×10^{-5}	0	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-1×10^{-5}	0	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
3×10^{-4}	-3×10^{-5}	0	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9

Table B2. Free convection cases used to train NORi.

where ΔT_0^a and ΔS_0^a are relative contributions of temperature and salinity to the total potential density variation in the initial condition of the training case a , $\|\Delta \phi_0^a\|_\infty = \max \bar{\phi}_0^a - \min \bar{\phi}_0^a$ is the maximum difference in the field of temperature or salinity initially throughout the vertical column, while α and β are the thermal expansion and haline coefficients of seawater respectively. Other components of A_ϕ^a are chosen such that

$$\frac{\mathcal{L}_\sigma^a}{\mathcal{L}_T^a + \mathcal{L}_S^a} = \frac{1}{9}, \quad (\text{C3})$$

$$\mathcal{L}_{\text{profile}}^a = \mathcal{L}_{\text{gradient}}^a, \quad (\text{C4})$$

$$\mathcal{L}_{\text{gradient}}^a = A_{\partial_z T}^a \delta(\partial_z \bar{T})^a + A_{\partial_z S}^a \delta(\partial_z \bar{S})^a + A_{\partial_z \sigma}^a \delta(\partial_z \bar{\sigma})^a \quad (\text{C5})$$

$$= A_{\text{gradient}}^a \left(A_T^a \delta(\partial_z \bar{T})^a + A_S^a \delta(\partial_z \bar{S})^a + A_\sigma^a \delta(\partial_z \bar{\sigma})^a \right), \quad (\text{C6})$$

$$\delta\phi \equiv \frac{1}{N_t N_z} \sum_{i=1}^{N_z} \sum_{j=1}^{N_t} \left| \phi^{i,j} - \phi_{LES}^{i,j} \right|^2 \quad (\text{C7})$$

where ϕ and ϕ_{LES} are one of the variables' (temperature, salinity, potential density) vertical gradient profile as a function of vertical level i , time j , and scenario a —the over-bars are a reminder that we are only parameterizing the area-averaged profiles.

Appendix D Neural network architecture

In NORi, neural networks are used to predict the entrainment fluxes of temperature and salinity, augmenting the physical base closure. This is reflected in the design of neural networks, where they are only active within the zone where entrainment occurs and are inactive everywhere else in the vertical column. For interior face-centered cell indices $i = 2, 3, \dots, N$, where $i = 1$ indicates the face location at the ocean sur-

J_T^{top} ($^{\circ}\text{C m s}^{-1}$)	J_S^{top} (psu m s^{-1})	J_u^{top} ($\text{m}^2 \text{s}^{-2}$)	$\frac{dT}{dz}$ ($^{\circ}\text{C m}^{-1}$)	$\frac{dS}{dz}$ (psu m^{-1})	f (s^{-1})	T^{top} ($^{\circ}\text{C}$)	S^{top} (psu)
5×10^{-4}	0	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
1×10^{-4}	0	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
5×10^{-4}	0	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
1×10^{-4}	0	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-5×10^{-5}	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-1×10^{-5}	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-5×10^{-5}	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
0	-1×10^{-5}	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
3×10^{-4}	-3×10^{-5}	-2×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
3×10^{-4}	-3×10^{-5}	-5×10^{-4}	1.4×10^{-2}	2.1×10^{-3}	8×10^{-5}	18	36.6
5×10^{-4}	0	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
1×10^{-4}	0	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
5×10^{-4}	0	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
1×10^{-4}	0	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-5×10^{-5}	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-1×10^{-5}	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-5×10^{-5}	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
0	-1×10^{-5}	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
3×10^{-4}	-3×10^{-5}	-2×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
3×10^{-4}	-3×10^{-5}	-5×10^{-4}	1.3×10^{-2}	7.5×10^{-4}	0	14.5	35
5×10^{-4}	0	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
1×10^{-4}	0	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
5×10^{-4}	0	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
1×10^{-4}	0	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-5×10^{-5}	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-1×10^{-5}	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-5×10^{-5}	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
0	-1×10^{-5}	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
5×10^{-4}	-5×10^{-5}	-2×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9
5×10^{-4}	-5×10^{-5}	-5×10^{-4}	-2.5×10^{-2}	-4.5×10^{-3}	-1.25×10^{-4}	0	33.9

Table B3. Wind + convection cases used to train NORi.

J_T^{top} ($^{\circ}\text{C m s}^{-1}$)	J_S^{top} (psu m s^{-1})	J_u^{top} ($\text{m}^2 \text{s}^{-2}$)	$\frac{dT}{dz}$ ($^{\circ}\text{C m}^{-1}$)	$\frac{dS}{dz}$ (psu m^{-1})	f (s^{-1})	T^{top} ($^{\circ}\text{C}$)	S^{top} (psu)
3×10^{-4}	0	-1×10^{-4}	1×10^{-2}	0	1×10^{-4}	10	35
5×10^{-4}	0	-1×10^{-4}	1.4×10^{-2}	0	1×10^{-4}	10	37
0	-5×10^{-5}	-1×10^{-4}	0	-5×10^{-3}	1.5×10^{-4}	0	34
0	-5×10^{-5}	-1×10^{-4}	0	-1×10^{-3}	1×10^{-4}	10	35
5×10^{-4}	-5×10^{-5}	-1×10^{-4}	6×10^{-2}	8×10^{-3}	1×10^{-4}	30	37
3×10^{-4}	-3×10^{-5}	-1×10^{-4}	6×10^{-2}	8×10^{-3}	1×10^{-4}	30	37
1×10^{-4}	-1×10^{-5}	-1×10^{-4}	6×10^{-2}	8×10^{-3}	1×10^{-4}	30	37

Table B4. Miscellaneous training cases, including cases with no temperature or salinity gradients, and cases with very strong stratification at warm temperatures used to train NORi.

J_T^{top} ($^{\circ}\text{C m s}^{-1}$)	J_S^{top} (psu m s^{-1})	J_u^{top} ($\text{m}^2 \text{s}^{-2}$)	$\frac{dT}{dz}$ ($^{\circ}\text{C m}^{-1}$)	$\frac{dS}{dz}$ (psu m^{-1})	f (s^{-1})	T^{top} ($^{\circ}\text{C}$)	S^{top} (psu)
3.5×10^{-4}	0	0	1.5×10^{-2}	2×10^{-3}	1×10^{-4}	20	37
0	-3.5×10^{-5}	0	1.5×10^{-2}	2×10^{-3}	1×10^{-4}	20	37
2×10^{-4}	-2×10^{-5}	-1×10^{-4}	1.5×10^{-2}	2×10^{-3}	1×10^{-4}	20	37
2×10^{-4}	-1.5×10^{-5}	-3.5×10^{-4}	1×10^{-2}	1.5×10^{-3}	1×10^{-4}	17	36
1.5×10^{-4}	-2×10^{-5}	-2.5×10^{-4}	1.7×10^{-2}	1.8×10^{-3}	1×10^{-4}	12	37
5×10^{-5}	-5×10^{-6}	-4×10^{-4}	1.2×10^{-2}	1.2×10^{-3}	1×10^{-4}	10	37
3.5×10^{-4}	-3.5×10^{-5}	-3×10^{-4}	1×10^{-2}	-2×10^{-3}	1.5×10^{-4}	16	34
4.5×10^{-4}	-4×10^{-5}	-4.5×10^{-4}	1.3×10^{-2}	-1×10^{-3}	7×10^{-5}	13	35
3.5×10^{-4}	0	0	1×10^{-2}	5×10^{-4}	0	17	34.5
0	-3.5×10^{-5}	0	1×10^{-2}	5×10^{-4}	0	17	34.5
2×10^{-4}	-2×10^{-5}	-1×10^{-4}	1×10^{-2}	5×10^{-4}	0	17	34.5
2×10^{-4}	-1.5×10^{-5}	-3.5×10^{-4}	1.2×10^{-2}	7×10^{-4}	-1×10^{-5}	13	36
1.5×10^{-4}	-2×10^{-5}	-2.5×10^{-4}	1.6×10^{-2}	6×10^{-4}	-2×10^{-5}	16	34
5×10^{-5}	-5×10^{-6}	-4×10^{-4}	1.1×10^{-2}	3×10^{-4}	3×10^{-5}	10	36.5
3.5×10^{-4}	-3.5×10^{-5}	-3×10^{-4}	1.5×10^{-2}	-5×10^{-4}	-5×10^{-5}	20	37
4.5×10^{-4}	-4×10^{-5}	-4.5×10^{-4}	1.7×10^{-2}	-8×10^{-4}	-1×10^{-4}	12	35
3.5×10^{-4}	0	0	-2×10^{-2}	-4.7×10^{-3}	-1.5×10^{-4}	0	34.5
0	-3.5×10^{-5}	0	-2×10^{-2}	-4.7×10^{-3}	-1.5×10^{-4}	0	34.5
2×10^{-4}	-2×10^{-5}	-1×10^{-4}	-2×10^{-2}	-4.7×10^{-3}	-1.5×10^{-4}	0	34.5
2×10^{-4}	-1.5×10^{-5}	-3.5×10^{-4}	-1.7×10^{-2}	-4×10^{-3}	1.5×10^{-4}	3	34
4×10^{-4}	-4×10^{-5}	-2.5×10^{-4}	-1.5×10^{-2}	-4×10^{-3}	1×10^{-4}	-1	36
5×10^{-5}	-5×10^{-6}	-4×10^{-4}	-1×10^{-2}	-3×10^{-3}	1.25×10^{-4}	0	33.5
3.5×10^{-4}	-3.5×10^{-5}	-3×10^{-4}	-2×10^{-2}	-4×10^{-3}	-1.4×10^{-4}	1	35.5
4.5×10^{-4}	-4×10^{-5}	-4.5×10^{-4}	-2×10^{-2}	-4.5×10^{-3}	-1×10^{-4}	2	34
4.5×10^{-4}	0	-2×10^{-4}	3×10^{-2}	0	1×10^{-4}	20	37
1.5×10^{-4}	0	-3×10^{-4}	1×10^{-2}	0	1×10^{-4}	15	35
0	-4×10^{-5}	-3×10^{-4}	0	-4.5×10^{-3}	-1×10^{-4}	17	36
0	-2×10^{-5}	-1.5×10^{-4}	0	-2.5×10^{-3}	-1×10^{-4}	16	34
4×10^{-4}	-4×10^{-5}	-4×10^{-4}	5×10^{-2}	-7×10^{-3}	1×10^{-4}	25	36
3×10^{-4}	-2×10^{-5}	-2×10^{-4}	3×10^{-2}	-7.5×10^{-3}	1×10^{-4}	27	35

Table B5. Validation cases to evaluate performance and select final trained weights of NORi.

face and $i = N + 1$ indicates the grid point at the bottom of the domain,

$$J_{\text{NN}_\phi}^i = \begin{cases} \text{NN}_\phi(\mathbf{h}^i; \boldsymbol{\theta}_\phi), & \text{for } i = i_{\min}, i_{\min} + 1, \dots, i_{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (\text{D1})$$

$$\mathbf{h}^i = \left[(\partial_z \bar{T})_{\text{zone}}^i, (\partial_z \bar{S})_{\text{zone}}^i, (\partial_z \bar{\sigma})_{\text{zone}}^i, \arctan(\mathbf{Ri}_{\text{zone}}^i), J_b^{\text{top}} \right]^{\top}, \quad (\text{D2})$$

$$\phi_{\text{zone}}^i \equiv \left[\phi^{\min\{i+2,N\}}, \phi^{\min\{i+1,N\}}, \phi^i, \phi^{\max\{i-1,2\}}, \phi^{\max\{i-2,2\}} \right] \quad (\text{D3})$$

where NN_ϕ is a neural network to predict the turbulent flux of tracer ϕ , \mathbf{h}^i is the input vector to the neural network at index i , $\boldsymbol{\theta}_\phi$ are the trainable weights for the neural network NN_ϕ , i_{\min} and i_{\max} are the first and last grid index inclusive of which the neural networks are active, ϕ_{zone}^i denotes the 5-element vector of variable ϕ around the local neighborhood of index i , ∂_z is the discrete gradient operator in z , and J_b^{top} is the surface buoyancy flux. For any face-centered grid point within the “entrainment zone”, the value of the temperature and salinity fluxes are predicted by NN_T and NN_S , respectively, where they each output a scalar which is the predicted value of temperature or salinity flux at the grid point.

The inputs of NN_T and NN_S are the temperature gradient, salinity gradient, potential density gradient, and gradient Richardson number of 5 grid points within the neighborhood of the grid point intended for inference, as well as the surface buoyancy flux. The inputs and outputs of the neural networks are normalized using a z-score normalization against the entire training suite to ensure that for each variable, the distribution of their values within the training suite has zero mean and unit variance. More details on feature normalization can be found in [Appendix F](#).

We adopt the principle of “simple, but not any simpler” to selecting the appropriate input variables to be included in the feature vector \mathbf{h}^i , and their reasons are as such:

1. The rate of entrainment, which is the process we want to model using neural networks, is primarily dependent on the strength of the buoyancy flux J_b^{top} at the ocean surface. This is because the surface buoyancy flux is the source of entrainment mixing through the generation of vertically coherent plumes. The buoyancy flux is a function of the temperature and salinity fluxes, but we provide only the buoyancy flux to the neural network to eliminate the need for neural networks to learn the implicit correlation between temperature, salinity and buoyancy fluxes. This is because the strength of the penetration of the entrainment is a function of buoyancy, not temperature or salinity separately.
2. The strength of entrainment mixing is strongly dependent on the ratio between local shear and background stratification at the base of the mixed layer. A strong stratification in the thermocline would reduce the extent of penetrative convection where convective plumes overshoot into the interior due to inertia, while stronger local shear leads to stronger plume “rollup”, which also reduces the plume penetration. The information about local shear and stratification is encoded in the local Richardson number Ri .
3. The mixing rate due to entrainment depends on local tracer gradients regardless of their absolute values. To enforce this tracer invariance, the temperature, salinity and potential density gradients $\frac{\partial \bar{T}}{\partial z}$, $\frac{\partial \bar{S}}{\partial z}$ and $\frac{\partial \bar{\sigma}}{\partial z}$ as well as the Richardson number Ri are used as input of the neural network instead of \bar{u} , \bar{v} , \bar{T} , \bar{S} , and $\bar{\sigma}$, since their actual values do not play a direct role in mixing dynamics.
4. NORi is designed for the nonlinear equation of state TEOS-10 ([Roquet et al., 2015](#)). Therefore, the correlation between temperature and salinity gradients with stratification depends on the actual values of temperature and salinity. To encode information on nonlinearities in the equation of state explicitly in neural networks, the potential density gradient $\frac{\partial \bar{\sigma}}{\partial z}$ is also provided.

5. Entrainment can be thought of in terms of the interaction between nonlocal buoyancy fluxes with local tracer gradients. The mixing occurs only locally due to a nonlocal flux source. Therefore, tracer gradients are provided locally, while the surface buoyancy fluxes are provided nonlocally to the neural networks.

The weights of NN_T and NN_S are shared between all points on the grid. The local gradient Richardson number is provided as the inputs to the neural networks in the form of $\arctan(Ri)$ so that it is bounded from above and below by $\pm\frac{\pi}{2}$, while having the largest gradients where $Ri \sim \mathcal{O}(1)$, which is in the fluid regime where turbulent dynamics are most sensitive to shear. The “entrainment zone” where the neural network is active is diagnosed from the diffusivity field κ by

$$i_\kappa = \min\{i : \kappa[i] = \kappa_0 \text{ for } i = 1, 2, \dots, N + 1\}, \quad (\text{D4})$$

$$i_{\min} = \max\{i_\kappa - 10, 2\}, \quad (\text{D5})$$

$$i_{\max} = \min\{i_\kappa + 5, N\}, \quad (\text{D6})$$

where i_κ is the grid point closest to the ocean surface where the base closure diffusivity κ is equal to the background diffusivity κ_0 . This is equivalent to the nearest point from the surface where $Ri \geq Ri^c$ where Ri^c is given by the base closure. The neural networks do not predict any turbulent fluxes at the grid point on the ocean surface $i = 1$ and the bottom grid point $i = N + 1$. At these boundary points, the temperature and salinity fluxes are given by the boundary conditions. This restriction of the prediction zone of the neural networks reduces inference time, but implicitly assumes that convective plumes cannot penetrate deeper than 40 m, which has been the case in all the LES generated. However, should it be found that the extent of entrainment exceeds 40 m in the cases we have not considered, such as very deep mixed layers, we can easily increase the depth of the neural network prediction zone by changing the values of i_{\min} and i_{\max} without any modifications to the neural network architecture. Figure D1 illustrates a schematic of the neural network architecture.

This design enforces some important physical principles:

1. The top and bottom fluxes are prescribed in order to satisfy the surface boundary conditions. This enforces tracer conservation as a hard constraint; NORi can only redistribute tracers in the interior.
2. In NORi, the local neighborhood visible to the neural networks as inputs spans 48 m, as we have found this to be a vertical extent that is sufficient for the neural networks to characterize the location of inference relative to the base of the boundary layer while ensuring generalizability across different locations within the vertical column.
3. Entrainment has local effects: it enhances mixing in a small neighborhood near the base of the boundary layer as it brings denser fluid from the ocean interior into the mixed layer. Therefore, the neural network only needs to know the information around its neighborhood to determine if entrainment occurs, and produces a local flux to represent any mixing.

Appendix E PDE discretization of the 1D column model

To solve the partial differential equations (11), (12), (13), and (14) numerically, we perform numerical discretizations both spatially and temporally. The spatial dimensions are discretized using the finite volume method, while the temporal discretization is done using the split implicit-explicit (IMEX) method. The diffusion terms are time stepped with the implicit backward Euler method, while the advection terms ($f\bar{u}$ and $f\bar{v}$) and neural network fluxes are time stepped with the explicit forward Euler method. The dis-

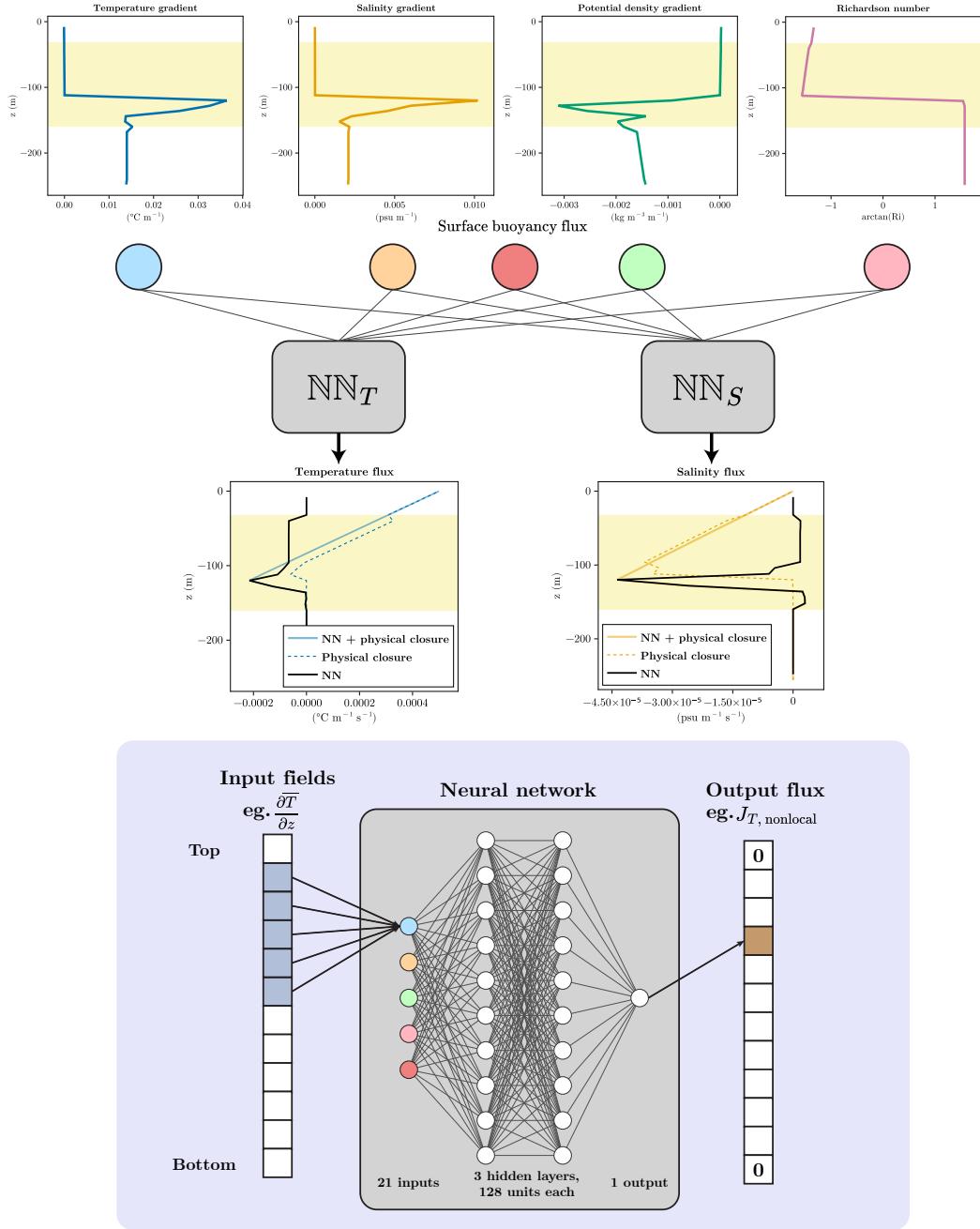


Figure D1. Schematic of NORi's neural network architecture used to augment the base closure to capture missing entrainment fluxes. Top panel: the inputs of the neural networks $\text{NN}_T : \mathbb{R}^{21} \rightarrow \mathbb{R}$ and $\text{NN}_S : \mathbb{R}^{21} \rightarrow \mathbb{R}$, as well as their outputs, which are the entrainment temperature and salinity fluxes $J_{T, \text{nonlocal}}^i, J_{S, \text{nonlocal}}^i$. The light yellow zones indicate the depth range where the neural networks are active. Beyond these zones neural network fluxes are zero. The temperature and salinity output fluxes show the neural network augmenting the base closure due to entrainment. Bottom panel: neural network architecture. The neural networks each produce one output which are the residual fluxes at grid point i given input $\mathbf{h}^i \in \mathbb{R}^{21}$, consisting of tracer gradients 5 grid points in the local neighborhood of grid point i and the surface buoyancy flux (see Appendix D). The neural networks do not predict fluxes at the top and bottom grid points.

cretized equations are given by

$$\frac{\bar{u}^{n+1} - \bar{u}^n}{\Delta t} = \partial_z (\nu \partial_z \bar{u}^{n+1}) + f \bar{v}^n, \quad (\text{E1})$$

$$\frac{\bar{v}^{n+1} - \bar{v}^n}{\Delta t} = \partial_z (\nu \partial_z \bar{v}^{n+1}) - f \bar{u}^n, \quad (\text{E2})$$

$$\frac{\bar{T}^{n+1} - \bar{T}^n}{\Delta t} = \partial_z (\kappa \partial_z \bar{T}^{n+1}) - \partial_z \mathbf{J}_{\text{NN}_T}^n, \quad (\text{E3})$$

$$\frac{\bar{S}^{n+1} - \bar{S}^n}{\Delta t} = \partial_z (\kappa \partial_z \bar{S}^{n+1}) - \partial_z \mathbf{J}_{\text{NN}_S}^n, \quad (\text{E4})$$

where $n = 0, 1, \dots$ indicate discrete time steps with $t^{n+1} - t^n = \Delta t$ and ∂_z is the discrete gradient operator in z .

Appendix F Variable normalization and model non-dimensionalization

In order to promote better training behavior of neural networks and to allow for the addition of field quantities with different units, we normalize the field and flux variables using a Z-score normalization such that the variables in the training suite have a zero mean and unit variance. This approach allows for all features to be treated equally during neural network training. Z-score normalization can be expressed as

$$\bar{u}^* = \frac{\bar{u} - \mu_{\bar{u}}}{\sigma_{\bar{u}}}, \quad (\text{F1})$$

$$\bar{v}^* = \frac{\bar{v} - \mu_{\bar{v}}}{\sigma_{\bar{v}}}, \quad (\text{F2})$$

$$\bar{T}^* = \frac{\bar{T} - \mu_{\bar{T}}}{\sigma_{\bar{T}}}, \quad (\text{F3})$$

$$\bar{S}^* = \frac{\bar{S} - \mu_{\bar{S}}}{\sigma_{\bar{S}}}, \quad (\text{F4})$$

$$\bar{u}'w'^* = \frac{\bar{u}'w' - \mu_{\bar{u}'w'}}{\sigma_{\bar{u}'w'}}, \quad (\text{F5})$$

$$\bar{v}'w'^* = \frac{\bar{v}'w' - \mu_{\bar{v}'w'}}{\sigma_{\bar{v}'w'}}, \quad (\text{F6})$$

$$\bar{w}'T'^* = \frac{\bar{w}'T' - \mu_{\bar{w}'T'}}{\sigma_{\bar{w}'T'}}, \quad (\text{F7})$$

$$\bar{w}'S'^* = \frac{\bar{w}'S' - \mu_{\bar{w}'S'}}{\sigma_{\bar{w}'S'}}, \quad (\text{F8})$$

where μ_ϕ and σ_ϕ are the mean and standard deviation of the field ϕ throughout the training suite in space and time. Using the normalization above, starred quantities ϕ^* are nondimensional. We also non-dimensionalize the time and space dimensions using

$$t^* = \frac{t}{\tau}, \quad (\text{F9})$$

$$z^* = \frac{z}{H}, \quad (\text{F10})$$

where τ is the duration of the training of each simulation in the training suite, while H is the total depth of the domain. Using the nondimensional variables above and expressing the fluxes fully in terms of the local fluxes as predicted by the base closure as well as the nonlocal entrainment fluxes as predicted by the neural networks, the nondimen-

sional equations which are solved during the training of NORi closure are

$$\frac{\partial \bar{u}^*}{\partial t^*} = \frac{\tau}{H^2} \frac{\partial}{\partial z^*} \left(\nu \frac{\partial \bar{u}^*}{\partial z^*} \right) + \frac{f\tau}{\sigma_{\bar{u}}} (\sigma_{\bar{v}} \bar{v}^* + \mu_{\bar{v}}), \quad (\text{F11})$$

$$\frac{\partial \bar{v}^*}{\partial t^*} = \frac{\tau}{H^2} \frac{\partial}{\partial z^*} \left(\nu \frac{\partial \bar{v}^*}{\partial z^*} \right) - \frac{f\tau}{\sigma_{\bar{v}}} (\sigma_{\bar{u}} \bar{u}^* + \mu_{\bar{u}}), \quad (\text{F12})$$

$$\frac{\partial \bar{T}^*}{\partial t^*} = \frac{\tau}{H^2} \frac{\partial}{\partial z^*} \left(\kappa \frac{\partial \bar{T}^*}{\partial z^*} \right) - \frac{\tau}{H} \frac{\sigma_{w'T'}}{\sigma_{\bar{T}}} \frac{\partial}{\partial z^*} J_{\text{NN}_T}, \quad (\text{F13})$$

$$\frac{\partial \bar{S}^*}{\partial t^*} = \frac{\tau}{H^2} \frac{\partial}{\partial z^*} \left(\kappa \frac{\partial \bar{S}^*}{\partial z^*} \right) - \frac{\tau}{H} \frac{\sigma_{w'S'}}{\sigma_{\bar{S}}} \frac{\partial}{\partial z^*} J_{\text{NN}_S}. \quad (\text{F14})$$

Appendix G Double-gyre solution at 100 years

Here we show a snapshot of the zonal average of the double-gyre simulation at $t = 100$ years to complement Figure 11 in the main text. At $t = 100$ years, the double-gyre stratifications begin to show significant differences between NORi, its unaugmented base closure, and k - ϵ closure. From Figure G1, when compared with k - ϵ and the base closure, we see that NORi produces deeper mixed layers along the entire zonal extent of the basin except for in the northernmost regions where the deep mixed layers reach the bottom of the domain. However, we do not have a “ground truth” solution as a reference since this is an idealized setup. Therefore, further qualification of each model’s skill requires running realistic ocean simulations, where observational data are available for comparison. However, this is beyond the present scope, and we aim to provide a thorough and systematic evaluation of various closures in realistic ocean simulations in the future.

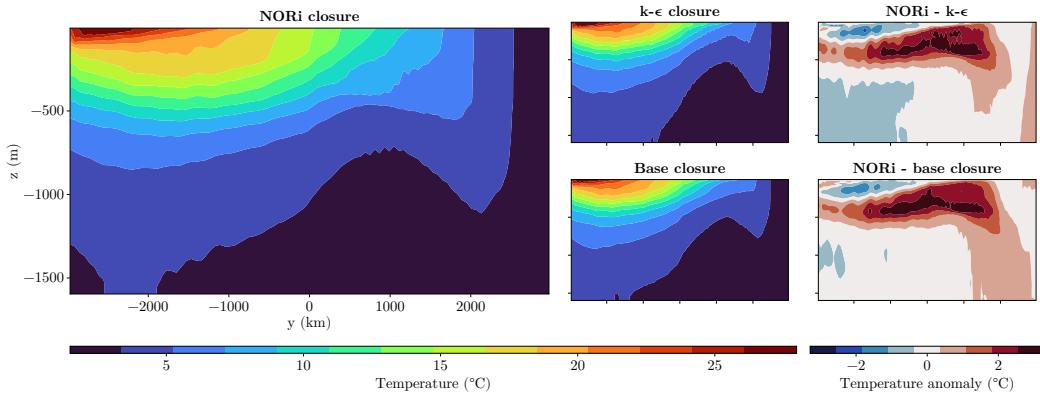


Figure G1. Zonally-averaged temperature stratification comparison between NORi, k - ϵ (Umlauf & Burchard, 2003), and the unaugmented base closure in the double-gyre simulation. The snapshots above are taken at $t = 100$ years. At this time, the simulation has not fully equilibrated yet. The rightmost column shows the temperature differences of the stratification between the NORi closure solution with that of k - ϵ ’s and the unaugmented base closure.

Appendix H NORi solution comparison with CATKE

In this section, we show some results comparing between NORi and CATKE (Wagner, Hillier, et al., 2025), a physics-based boundary layer parameterization that uses a diagnostic mixing length with a convective adjustment component and a prognostic turbu-

lent kinetic energy (TKE). CATKE demonstrated superior capabilities against commonly-used boundary layer parameterizations such as K-Profile Parameterization (KPP) (Large et al., 1994) and the Langmuir turbulence second moment closure (SMC-LT) described in (Harcourt, 2015) when benchmarked against idealized and realistic LES (Wagner, Hillier, et al., 2025).

Before making any comparisons, one important caveat is that CATKE is trained with an LES suite which takes into account the effects of additional mixing due to surface waves under shear-driven mixing scenarios, i.e., Langmuir turbulence. In addition, CATKE and NORi are calibrated using different training suites with LES of different background stratification, surface forcings, equations of state, and rotation rates. Under the same surface wind stress and initial conditions, LES which takes into account wave effects will in general create a deeper mixed layer. By extension, we would expect that CATKE would produce deeper mixed layers than NORi in strongly wind-driven scenarios, since NORi is trained on simulations that do not take into account wave effects. This is because the intention of developing NORi is to illustrate a new design and training paradigm for data-driven ocean modeling, and we have chosen to demonstrate it using a reduced set of physics that capture the leading order effects of vertical mixing due to winds and convection. Although the specific physical details of the training data are crucial, the NORi framework can be flexibly adapted to variants of microturbulence closures that take into account other physical processes through fine-tuning and/or transfer learning. Using the approach outlined in this work, other versions of NORi which account for additional complications such as wave effects and rotation-dependent deep convection or solve additional prognostic turbulent quantities can be developed in future work.

As expected, CATKE produces deeper mixed layers in single column training and validation cases driven by strong winds. In Figure H1, the first and fourth columns are cases of strong wind, where CATKE produces slightly deeper mixed layers than NORi. As noted earlier, the LES solutions shown in Figure H1 do not take into account wave effects; therefore, it is not surprising that they are generally shallower than the predictions of CATKE's. Discrepancies between CATKE and NORi are expected given the differences in their training data.

In the double-gyre simulation, the differences between CATKE and NORi are similar to those between k - ϵ and NORi (see Figures 11 and G1). At 2.5 years, shown by the zonally-averaged temperature in Figure H2, NORi produces shallower mixed layers than CATKE in the northern regions of the simulation. Interestingly, NORi also produces slightly deeper mixed layers in areas near the equator. At $t = 100$ years, we can see from Figure H3 that NORi generates mixed layers which are much deeper than CATKE in most regions except at the northernmost part of the basin where deep mixed layers reach the bottom, similar to the comparison with k - ϵ . As we do not have a ground truth solution for this idealized setup, a thorough evaluation of NORi and CATKE would require running realistic regional/global ocean simulations where observational data is available for comparison. However, it is not surprising that CATKE and k - ϵ behave similarly, as CATKE is heavily inspired by higher-order closures such as k - ϵ .

Open Research Section

The LES data used for the training and validation of NORi are generated from open source software `Oceananigans.jl` (Ramadhan et al., 2020; Wagner, Silvestri, et al., 2025). These LES data are packaged into a publicly available data set **SOBLLES - A Salty Ocean Boundary Layer Large-Eddy Simulation Dataset** hosted on Zenodo (Lee, 2025b). The base closure in NORi is calibrated using open source software `EnsembleKalmanProcesses.jl` (Dunbar et al., 2022). The neural networks in NORi are implemented using open source software `Lux.jl` (Pal, 2023) and calibrated using `Enzyme.jl` (Moses & Churavy, 2020). Scripts for

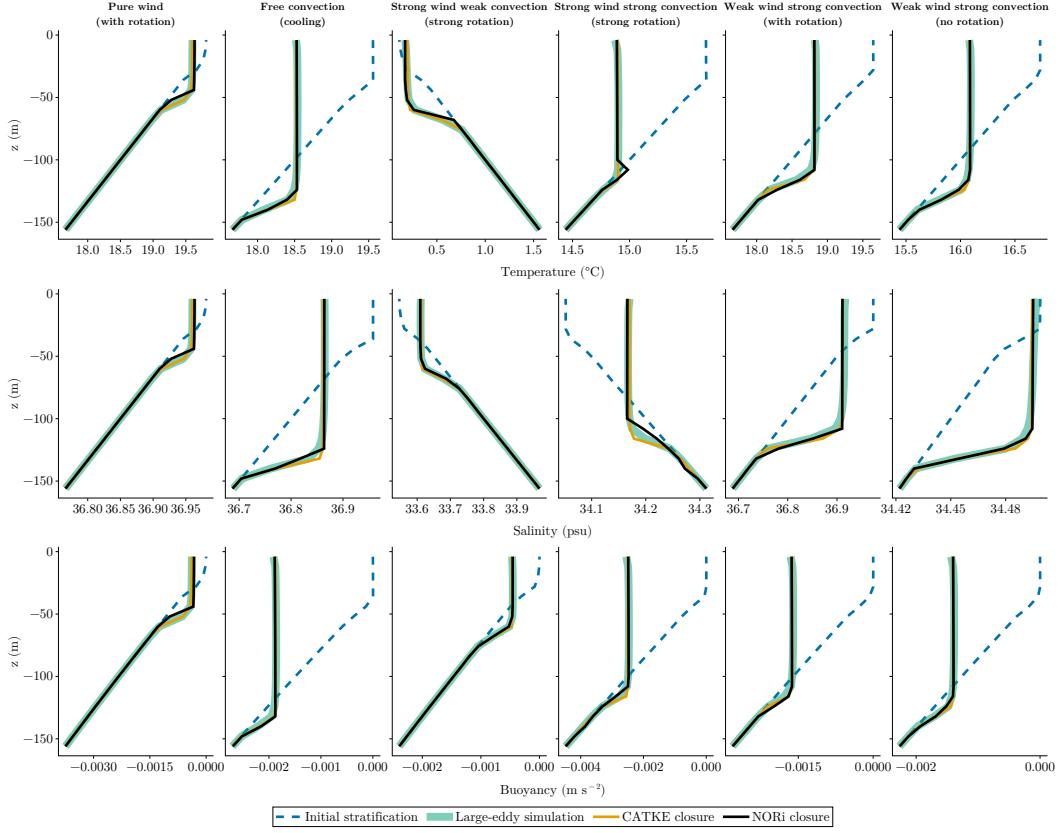


Figure H1. Comparison of NORi performance in the column model setting with CATKE (Wagner, Hillier, et al., 2025), a physics-based boundary layer parameterization that uses a diagnostic mixing length with a convective adjustment component and a prognostic turbulent kinetic energy (TKE). The comparison is made between the temperature, salinity, and buoyancy fields with respect to LES solutions. All examples shown are validation cases which were not seen by NORi during training.

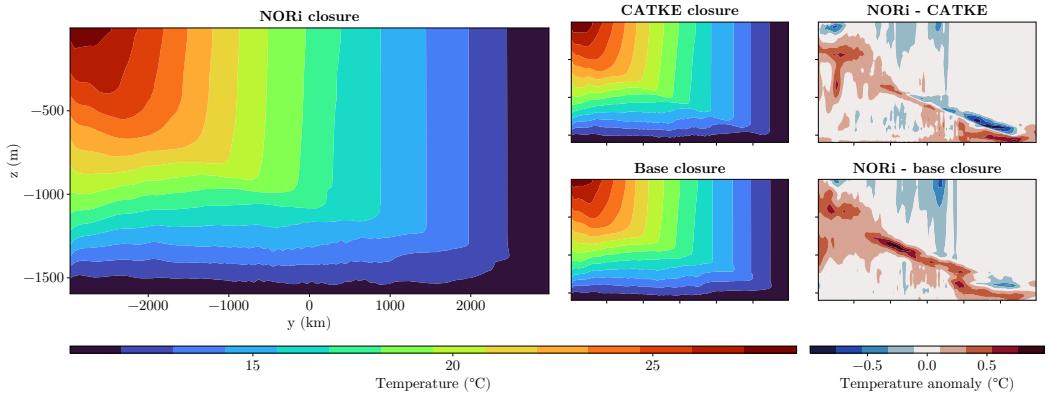


Figure H2. Zonally-averaged temperature stratification comparison between NORi, CATKE (Wagner, Hillier, et al., 2025), and the unaugmented base closure in the double-gyre simulation. The snapshots above are taken at $t = 2.5$ years, when the mixed layer extends to almost the entire water column in the north. The rightmost column shows the temperature differences of the stratification between the NORi closure solution with that of CATKE's and the unaugmented base closure.

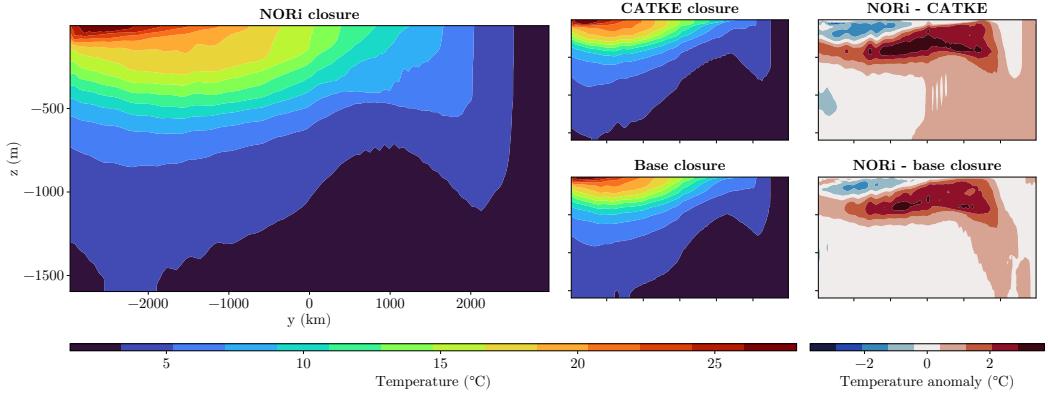


Figure H3. Same as Figure H2 above, but at $t = 100$ years.

the training and validation of NORi, as well as its implementation in Oceananigans.jl, together with the scripts to reproduce all figures in this paper are available at the Github repository **NORiOceanParameterization.jl** (Lee, 2025c). A data companion to **NORiOceanParameterization.jl** which hosts the data required to reproduce all figures in this paper is available on Zenodo (Lee, 2025a).

References

Balsara, D. S., & Shu, C.-W. (2000, May). Monotonicity Preserving Weighted Essentially Non-oscillatory Schemes with Increasingly High Order of Accuracy. *Journal of Computational Physics*, 160(2), 405–452. Retrieved 2023-12-08, from <https://linkinghub.elsevier.com/retrieve/pii/S002199910096443X> doi: 10.1006/jcph.2000.6443

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017, January). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1), 65–98. Retrieved 2024-04-25, from <https://pubs.siam.org/doi/10.1137/141000671> doi: 10.1137/141000671

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023, July). Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970), 533–538. Retrieved 2025-04-25, from <https://www.nature.com/articles/s41586-023-06185-3> doi: 10.1038/s41586-023-06185-3

Boccaletti, G., Ferrari, R., & Fox-Kemper, B. (2007, September). Mixed Layer Instabilities and Restratification. *Journal of Physical Oceanography*, 37(9), 2228–2250. Retrieved 2024-12-09, from <http://journals.ametsoc.org/doi/10.1175/JPO3101.1> doi: 10.1175/JPO3101.1

Bolton, T., & Zanna, L. (2019, January). Applications of Deep Learning to Ocean Data Inference and Subgrid Parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1), 376–399. Retrieved 2024-04-25, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2018MS001472> doi: 10.1029/2018MS001472

Chattopadhyay, A., Sun, Y. Q., & Hassanzadeh, P. (2023). *Challenges of learning multi-scale dynamics with AI weather models: Implications for stability and one solution*. arXiv. Retrieved 2025-04-25, from <https://arxiv.org/abs/2304.07029> (Version Number: 2) doi: 10.48550/ARXIV.2304.07029

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2019, December). *Neural Ordinary Differential Equations*. arXiv. Retrieved 2024-04-25, from <http://arxiv.org/abs/1806.07366> (arXiv:1806.07366 [cs, stat])

Copernicus Climate Change Service. (2019). *ERA5 monthly averaged data on single levels from 1940 to present*. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). Retrieved 2025-01-21, from <https://cds.climate.copernicus.eu/doi/10.24381/cds.f17050d7> doi: 10.24381/CDS.F17050D7

Deardorff, J. W., Willis, G. E., & Stockton, B. H. (1980, September). Laboratory studies of the entrainment zone of a convectively mixed layer. *Journal of Fluid Mechanics*, 100(1), 41–64. Retrieved 2025-11-06, from https://www.cambridge.org/core/product/identifier/S0022112080001000/type/journal_article doi: 10.1017/S0022112080001000

Drazin, P. G., & Reid, W. H. (2004). *Hydrodynamic Stability* (2nd ed.). Cambridge University Press. Retrieved 2023-05-22, from <https://www.cambridge.org/core/product/identifier/9780511616938/type/book> doi: 10.1017/CBO9780511616938

Dunbar, O. R. A., Lopez-Gomez, I., Garbuno-Iñigo, A., Huang, D. Z., Bach, E., & Wu, J.-l. (2022, December). EnsembleKalmanProcesses.jl: Derivative-free ensemble-based model calibration. *Journal of Open Source Software*, 7(80), 4869. Retrieved 2024-04-26, from [https://joss.theoj.org/papers/7\(80\), 4869](https://joss.theoj.org/papers/7(80), 4869)

[10.21105/joss.04869](https://doi.org/10.21105/joss.04869) doi: 10.21105/joss.04869

DuVivier, A. K., Large, W. G., & Small, R. J. (2018, October). Argo Observations of the Deep Mixing Band in the Southern Ocean: A Salinity Modeling Challenge. *Journal of Geophysical Research: Oceans*, 123(10), 7599–7617. Retrieved 2024-04-24, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2018JC014275> doi: 10.1029/2018JC014275

Frezat, H., Le Sommer, J., Fablet, R., Balarac, G., & Lguensat, R. (2022, November). A Posteriori Learning for Quasi-Geostrophic Turbulence Parametrization. *Journal of Advances in Modeling Earth Systems*, 14(11), e2022MS003124. Retrieved 2023-09-29, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2022MS003124> doi: 10.1029/2022MS003124

Frostig, R., Johnson, M. J., & Leary, C. (2018). Compiling machine learning programs via high-level tracing.. Retrieved from <https://mlsys.org/Conferences/doc/2018/146.pdf>

Gaillard, F., Reynaud, T., Thierry, V., Kolodziejczyk, N., & Von Schuckmann, K. (2016, February). In Situ-Based Reanalysis of the Global Ocean Temperature and Salinity with ISAS: Variability of the Heat Content and Steric Height. *Journal of Climate*, 29(4), 1305–1323. Retrieved 2024-04-25, from <http://journals.ametsoc.org/doi/10.1175/JCLI-D-15-0028.1> doi: 10.1175/JCLI-D-15-0028.1

Gjini, R., Morzfeld, M., Dunbar, O. R. A., & Schneider, T. (2025). *The Ensemble Kalman Inversion Race*. arXiv. Retrieved 2025-12-01, from <https://arxiv.org/abs/2511.15853> (Version Number: 1) doi: 10.48550/ARXIV.2511.15853

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256). JMLR Workshop and Conference Proceedings.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323). JMLR Workshop and Conference Proceedings.

Goyal, A., & Bengio, Y. (2022, October). Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2266), 20210068. Retrieved 2025-11-28, from <https://royalsocietypublishing.org/doi/10.1098/rspa.2021.0068> doi: 10.1098/rspa.2021.0068

Harcourt, R. R. (2015, January). An Improved Second-Moment Closure Model of Langmuir Turbulence. *Journal of Physical Oceanography*, 45(1), 84–103. Retrieved 2025-01-01, from <https://journals.ametsoc.org/view/journals/phoc/45/1/jpo-d-14-0046.1.xml> doi: 10.1175/JPO-D-14-0046.1

Hewitt, H. T., Roberts, M., Mathiot, P., Biastoch, A., Blockley, E., Chassignet, E. P., ... Zhang, Q. (2020, December). Resolving and Parameterising the Ocean Mesoscale in Earth System Models. *Current Climate Change Reports*, 6(4), 137–152. Retrieved 2025-11-06, from <https://link.springer.com/10.1007/s40641-020-00164-w> doi: 10.1007/s40641-020-00164-w

Iglesias, M. A., Law, K. J. H., & Stuart, A. M. (2013, April). Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4), 045001. Retrieved 2024-04-26, from <https://iopscience.iop.org/article/10.1088/0266-5611/29/4/045001> doi: 10.1088/0266-5611/29/4/045001

Kim, S., Ji, W., Deng, S., Ma, Y., & Rackauckas, C. (2021, September). Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9), 093122. Retrieved 2024-04-02, from <https://pubs.aip.org/cha/article/31/9/093122/1077547/Stiff-neural-ordinary-differential-equations> doi: 10.1063/5.0060697

Kingma, D. P., & Ba, J. (2017, January). *Adam: A Method for Stochastic Optimization*. arXiv. Retrieved 2024-04-11, from <http://arxiv.org/abs/1412.6980> (arXiv:1412.6980 [cs])

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021, May). Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), e2101784118. Retrieved 2024-04-01, from <https://pnas.org/doi/full/10.1073/pnas.2101784118> doi: 10.1073/pnas.2101784118

Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., ... Hoyer, S. (2024, August). Neural general circulation models for weather and climate. *Nature*, 632(8027), 1060–1066. Retrieved 2025-03-21, from <https://www.nature.com/articles/s41586-024-07744-y> doi: 10.1038/s41586-024-07744-y

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., ... Battaglia, P. (2023, August). *GraphCast: Learning skillful medium-range global weather forecasting*. arXiv. Retrieved 2024-11-24, from <http://arxiv.org/abs/2212.12794> (arXiv:2212.12794 [cs]) doi: 10.48550/arXiv.2212.12794

Large, W. G., & Gent, P. R. (1999, March). Validation of Vertical Mixing in an Equatorial Ocean Model Using Large Eddy Simulations and Observations. *Journal of Physical Oceanography*, 29(3), 449–464. Retrieved from https://journals.ametsoc.org/view/journals/phoc/29/3/1520-0485_1999_029_0449_vovmia_2.0.co_2.xml doi: [https://doi.org/10.1175/1520-0485\(1999\)029%3C0449:VOVMIA%3E2.0.CO;2](https://doi.org/10.1175/1520-0485(1999)029%3C0449:VOVMIA%3E2.0.CO;2)

Large, W. G., McWilliams, J. C., & Doney, S. C. (1994, November). Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization. *Reviews of Geophysics*, 32(4), 363–403. Retrieved 2024-04-24, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/94RG01872> doi: 10.1029/94RG01872

Lee, X. K. (2025a, November). *Data Companion for NORiOceanParameterization.jl* [Data set]. Zenodo. Retrieved 2025-12-01, from <https://zenodo.org/doi/10.5281/zenodo.17605195> doi: 10.5281/ZENODO.17605195

Lee, X. K. (2025b, November). *SOBLLES: A Salty Ocean Boundary Layer Large-Eddy Simulations Dataset* [Data set]. Zenodo. Retrieved 2025-11-22, from <https://zenodo.org/doi/10.5281/zenodo.17677802> doi: 10.5281/ZENODO.17677802

Lee, X. K. (2025c, December). *xkykai/NORiOceanParameterization.jl: initial release* [Software]. Zenodo. Retrieved 2025-12-01, from <https://zenodo.org/doi/10.5281/zenodo.17773381> doi: 10.5281/ZENODO.17773381

Li, G., & Xie, S.-P. (2014, February). Tropical Biases in CMIP5 Multimodel Ensemble: The Excessive Equatorial Pacific Cold Tongue and Double ITCZ Problems*. *Journal of Climate*, 27(4), 1765–1780. Retrieved 2024-04-24, from <http://journals.ametsoc.org/doi/10.1175/JCLI-D-13-00337.1> doi: 10.1175/JCLI-D-13-00337.1

Liang, J.-H., Yuan, J., Wan, X., Liu, J., Liu, B., Jang, H., & Tyagi, M. (2022, August). Exploring the use of machine learning to parameterize vertical mixing in the ocean surface boundary layer. *Ocean Modelling*, 176, 102059. Retrieved 2023-10-03, from <https://linkinghub.elsevier.com/retrieve/pii/S1463500322000890> doi: 10.1016/j.ocemod.2022.102059

Marshall, J., & Schott, F. (1999, February). Open-ocean convection: Observations, theory, and models. *Reviews of Geophysics*, 37(1), 1–64. Retrieved 2024-04-24, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/98RG02739> doi: 10.1029/98RG02739

McWilliams, J. C. (2011). *Fundamentals of geophysical fluid dynamics*. Cambridge University Press. Retrieved from <https://www.cambridge.org/>

us/universitypress/subjects/earth-and-environmental-science/oceanography-and-marine-science/fundamentals-geophysical-fluid-dynamics?format=HB&isbn=9781107404083#description

Mellor, G. L., & Yamada, T. (1982, November). Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics*, 20(4), 851–875. Retrieved 2024-04-24, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/RG020i004p00851> doi: 10.1029/RG020i004p00851

Molod, A. (2009, May). Running GCM physics and dynamics on different grids: algorithm and tests. *Tellus A*, 61(3), 381–393. Retrieved 2024-11-27, from <http://tellusa.net/index.php/tellusa/article/view/15602> doi: 10.1111/j.1600-0870.2009.00394.x

Moses, W., Cheng, G., Churavy, V., Gelbrecht, M., Klöwer, M., Kump, J., ... Heimbach, P. (2025, November). *DJ4Earth: Differentiable, and Performance-portable Earth System Modeling via Program Transformations*. Preprints. Retrieved 2025-12-02, from <https://essopenarchive.org/users/1000301/articles/1360053-dj4earth-differentiable-and-performance-portable-earth-system-modeling-via-program-transformations?commit=b41d12401ff9fbfce8bb85c4a3c9cd8681e142a0> doi: 10.22541/essoar.176314951.18114616/v1

Moses, W., & Churavy, V. (2020). Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 12472–12485). Curran Associates, Inc. Retrieved from <https://proceedings.neurips.cc/paper/2020/file/9332c513ef44b682e9347822c2e457ac-Paper.pdf>

Niiler, P. (1977). *One-dimensional models of the upper ocean, Modelling and Prediction of the Upper Layers of the Ocean* EB Kraus, 143–172. Pergamon, New York.

Pacanowski, R., & Philander, S. (1981). Parameterization of vertical mixing in numerical models of tropical oceans. *Journal of Physical Oceanography*, 11(11), 1443–1451.

Pal, A. (2023, April). *Lux: Explicit Parameterization of Deep Neural Networks in Julia*. [object Object]. Retrieved 2024-04-26, from <https://zenodo.org/record/7808904> doi: 10.5281/ZENODO.7808904

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., ... Anandkumar, A. (2022, February). *FourCastNet: A Global Data-driven High-resolution Weather Model using Adaptive Fourier Neural Operators*. arXiv. Retrieved 2023-10-03, from <http://arxiv.org/abs/2202.11214> (arXiv:2202.11214 [physics])

Pedlosky, J. (1996). *Ocean circulation theory*. Berlin ; New York: Springer.

Pressel, K. G., Mishra, S., Schneider, T., Kaul, C. M., & Tan, Z. (2017, June). Numerics and subgrid-scale modeling in large eddy simulations of stratocumulus clouds. *Journal of Advances in Modeling Earth Systems*, 9(2), 1342–1365. Retrieved 2023-12-08, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1002/2016MS000778> doi: 10.1002/2016MS000778

Price, J. F., Weller, R. A., & Pinkel, R. (1986, July). Diurnal cycling: Observations and models of the upper ocean response to diurnal heating, cooling, and wind mixing. *Journal of Geophysical Research: Oceans*, 91(C7), 8411–8427. Retrieved 2025-11-07, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/JC091iC07p08411> doi: 10.1029/JC091iC07p08411

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., ... Edelman, A. (2021, November). *Universal Differential Equations for Scientific Machine Learning*. arXiv. Retrieved 2024-04-02, from <http://arxiv.org/abs/2001.04385> (arXiv:2001.04385 [cs, math, q-bio, stat])

Ramadhan, A., Marshall, J., Souza, A., Lee, X. K., Piterbarg, U., Hillier, A., ...

Ferrari, R. (2023, March). *Capturing missing physics in climate model parameterizations using neural differential equations*. arXiv. Retrieved 2024-04-25, from <http://arxiv.org/abs/2010.12559> (arXiv:2010.12559 [physics])

Ramadhan, A., Wagner, G., Hill, C., Campin, J.-M., Churavy, V., Besard, T., ...

Marshall, J. (2020, September). Oceananigans.jl: Fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, 5(53), 2018. Retrieved 2024-04-22, from <https://joss.theoj.org/papers/10.21105/joss.02018> doi: 10.21105/joss.02018

Reffray, G., Bourdalle-Badie, R., & Calone, C. (2015, January). Modelling turbulent vertical mixing sensitivity using a 1-D version of NEMO. *Geoscientific Model Development*, 8(1), 69–86. Retrieved 2024-04-24, from <https://gmd.copernicus.org/articles/8/69/2015/> doi: 10.5194/gmd-8-69-2015

Reichl, B. G., & Hallberg, R. (2018, December). A simplified energetics based planetary boundary layer (ePBL) approach for ocean climate simulations. *Ocean Modelling*, 132, 112–129. Retrieved 2024-04-24, from <https://linkinghub.elsevier.com/retrieve/pii/S1463500318301069> doi: 10.1016/j.ocemod.2018.10.004

Reuther, A., Kepner, J., Byun, C., Samsi, S., Arcand, W., Bestor, D., ... Michaleas, P. (2018, September). Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)* (pp. 1–6). Waltham, MA: IEEE. Retrieved 2025-01-02, from <https://ieeexplore.ieee.org/document/8547629/> doi: 10.1109/HPEC.2018.8547629

Roquet, F., Madec, G., McDougall, T. J., & Barker, P. M. (2015, June). Accurate polynomial expressions for the density and specific volume of seawater using the TEOS-10 standard. *Ocean Modelling*, 90, 29–43. Retrieved 2023-10-09, from <https://linkinghub.elsevier.com/retrieve/pii/S1463500315000566> doi: 10.1016/j.ocemod.2015.04.002

Sallée, J., Shuckburgh, E., Bruneau, N., Meijers, A. J. S., Bracegirdle, T. J., & Wang, Z. (2013, April). Assessment of Southern Ocean mixed-layer depths in CMIP5 models: Historical bias and forcing response. *Journal of Geophysical Research: Oceans*, 118(4), 1845–1862. Retrieved 2025-01-28, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1002/jgrc.20157> doi: 10.1002/jgrc.20157

Sane, A., Reichl, B. G., Adcroft, A., & Zanna, L. (2023, October). Parameterizing Vertical Mixing Coefficients in the Ocean Surface Boundary Layer Using Neural Networks. *Journal of Advances in Modeling Earth Systems*, 15(10), e2023MS003890. Retrieved 2024-03-04, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2023MS003890> doi: 10.1029/2023MS003890

Schumann, U., & Sweet, R. A. (1988, March). Fast Fourier transforms for direct solution of poisson's equation with staggered boundary conditions. *Journal of Computational Physics*, 75(1), 123–137. Retrieved 2023-01-05, from <https://linkinghub.elsevier.com/retrieve/pii/0021999188901027> doi: 10.1016/0021-9991(88)90102-7

Silvestri, S., Wagner, G. L., Campin, J.-M., Constantinou, N. C., Hill, C. N., Souza, A. N., & Ferrari, R. (2023, November). *A new WENO-based momentum advection scheme for simulations of ocean mesoscale turbulence* (preprint). Preprints. Retrieved 2023-12-05, from <https://essopenarchive.org/users/703084/articles/688784-a-new-weno-based-momentum-advection-scheme-for-simulations-of-ocean-mesoscale-turbulence?commit=3fa224d45d29525b0d6d8b8963e725ff5de12724> doi: 10.22541/essoar.170110657.76489860/v1

Silvestri, S., Wagner, G. L., Constantinou, N. C., Hill, C. N., Campin, J., Souza,

A. N., ... Ferrari, R. (2025, April). A GPU-Based Ocean Dynamical Core for Routine Mesoscale-Resolving Climate Simulations. *Journal of Advances in Modeling Earth Systems*, 17(4), e2024MS004465. Retrieved 2025-05-02, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2024MS004465> doi: 10.1029/2024MS004465

Sirignano, J., MacArt, J. F., & Freund, J. B. (2020, December). DPM: A deep learning PDE augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423, 109811. Retrieved 2024-04-25, from <https://linkinghub.elsevier.com/retrieve/pii/S0021999120305854> doi: 10.1016/j.jcp.2020.109811

Souza, A. N., Wagner, G. L., Ramadhan, A., Allen, B., Churavy, V., Schloss, J., ... Ferrari, R. (2020, December). Uncertainty Quantification of Ocean Parameterizations: Application to the K-Profile-Parameterization for Penetrative Convection. *Journal of Advances in Modeling Earth Systems*, 12(12), e2020MS002108. Retrieved 2023-12-25, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2020MS002108> doi: 10.1029/2020MS002108

Stachenfeld, K., Fielding, D. B., Kochkov, D., Cranmer, M., Pfaff, T., Godwin, J., ... Sanchez-Gonzalez, A. (2021). Learned Coarse Models for Efficient Turbulence Simulation.. Retrieved 2024-04-25, from <https://arxiv.org/abs/2112.15275> (Version Number: 3) doi: 10.48550/ARXIV.2112.15275

Subel, A., Chattopadhyay, A., Guan, Y., & Hassanzadeh, P. (2021, March). Data-driven subgrid-scale modeling of forced Burgers turbulence using deep learning with generalization to higher Reynolds numbers via transfer learning. *Physics of Fluids*, 33(3), 031702. Retrieved 2025-01-21, from <https://pubs.aip.org/pof/article/33/3/031702/973200/Data-driven-subgrid-scale-modeling-of-forced> doi: 10.1063/5.0040286

The Climode Group:, Marshall, J., Ferrari, R., Forget, G., Maze, G., Andersson, A., ... Thomas, L. (2009, September). The Climode Field Campaign: Observing the Cycle of Convection and Restratiification over the Gulf Stream. *Bulletin of the American Meteorological Society*, 90(9), 1337–1350. Retrieved 2025-01-21, from <http://journals.ametsoc.org/doi/10.1175/2009BAMS2706.1> doi: 10.1175/2009BAMS2706.1

Treguier, A. M., De Boyer Montégut, C., Bozec, A., Chassignet, E. P., Fox-Kemper, B., McC. Hogg, A., ... Yeager, S. (2023, July). The mixed-layer depth in the Ocean Model Intercomparison Project (OMIP): impact of resolving mesoscale eddies. *Geoscientific Model Development*, 16(13), 3849–3872. Retrieved 2024-04-23, from <https://gmd.copernicus.org/articles/16/3849/2023/> doi: 10.5194/gmd-16-3849-2023

Umlauf, L., & Burchard, H. (2003). A generic length-scale equation for geophysical turbulence models. *Journal of Marine Research*, 61(2), 235–265. (Publisher: Sears Foundation for Marine Research)

Vallis, G. K. (2017). *Atmospheric and Oceanic Fluid Dynamics: Fundamentals and Large-Scale Circulation* (2nd ed.). Cambridge University Press. Retrieved 2023-03-28, from <https://www.cambridge.org/core/product/identifier/9781107588417/type/book> doi: 10.1017/9781107588417

Van Roekel, L., Adcroft, A. J., Danabasoglu, G., Griffies, S. M., Kauffman, B., Large, W., ... Schmidt, M. (2018, November). 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. Retrieved 2023-12-25, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2018MS001336> doi: 10.1029/2018MS001336

Vreugdenhil, C. A., & Gayen, B. (2021, October). Ocean Convection. *Fluids*, 6(10), 360. Retrieved 2025-04-25, from <https://www.mdpi.com/2311-5521/6/10/360> doi: 10.3390/fluids6100360

Wagner, G. L., Hillier, A., Constantinou, N. C., Silvestri, S., Souza, A., Burns, K. J., ... Ferrari, R. (2025, April). Formulation and Calibration of CATKE, a One-Equation Parameterization for Microscale Ocean Mixing. *Journal of Advances in Modeling Earth Systems*, 17(4), e2024MS004522. Retrieved 2025-04-25, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2024MS004522> doi: 10.1029/2024MS004522

Wagner, G. L., Silvestri, S., Constantinou, N. C., Ramadhan, A., Campin, J.-M., Hill, C., ... Ferrari, R. (2025). *High-level, high-resolution ocean modeling at all scales with Oceananigans*. arXiv. Retrieved 2025-04-25, from <https://arxiv.org/abs/2502.14148> (Version Number: 1) doi: 10.48550/ARXIV.2502.14148

Young, W. R. (1994, August). The Subinertial Mixed Layer Approximation. *Journal of Physical Oceanography*, 24(8), 1812–1826. doi: [https://doi.org/10.1175/1520-0485\(1994\)024%3C1812:TSMLA%3E2.0.CO;2](https://doi.org/10.1175/1520-0485(1994)024%3C1812:TSMLA%3E2.0.CO;2)

Yu, L. (2007, November). Global Variations in Oceanic Evaporation (1958–2005): The Role of the Changing Wind Speed. *Journal of Climate*, 20(21), 5376–5390. Retrieved 2025-01-21, from <http://journals.ametsoc.org/doi/10.1175/2007JCLI1714.1> doi: 10.1175/2007JCLI1714.1

Yuan, J., Liang, J., Chassignet, E. P., Zavala-Romero, O., Wan, X., & Cronin, M. F. (2024, September). The K-Profile Parameterization Augmented by Deep Neural Networks (KPP_dnn) in the General Ocean Turbulence Model (GOTM). *Journal of Advances in Modeling Earth Systems*, 16(9), e2024MS004405. Retrieved 2025-01-02, from <https://agupubs.onlinelibrary.wiley.com/doi/10.1029/2024MS004405> doi: 10.1029/2024MS004405