Parameterizing Vertical Turbulent Mixing Coefficients In The Ocean Surface Boundary Layer Using Neural Networks

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

[1] Princeton University (GFDL Affiliate)
[2] NOAA - Geophysical Fluid Dynamics Laboratory
[3] New York University

Highlights:

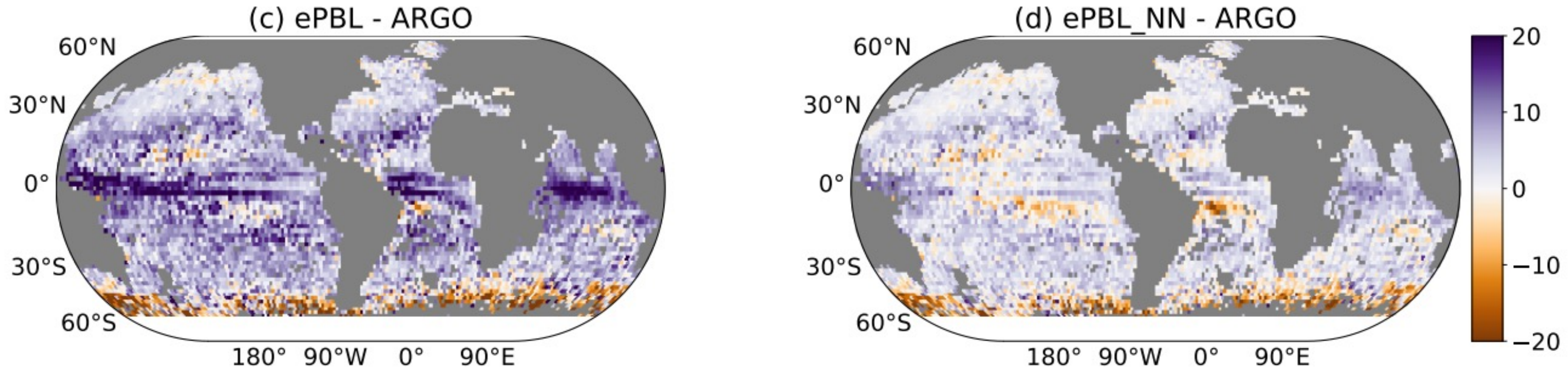
1. Neural networks - within the existing energetics based physics framework to improve vertical diffusivity in the OSBL.
2. Scheme implemented in MOM6.
3. JRA forced simulations performed: Bias reduction in shallow mixed layer depth and upper ocean stratification.

Highlights:

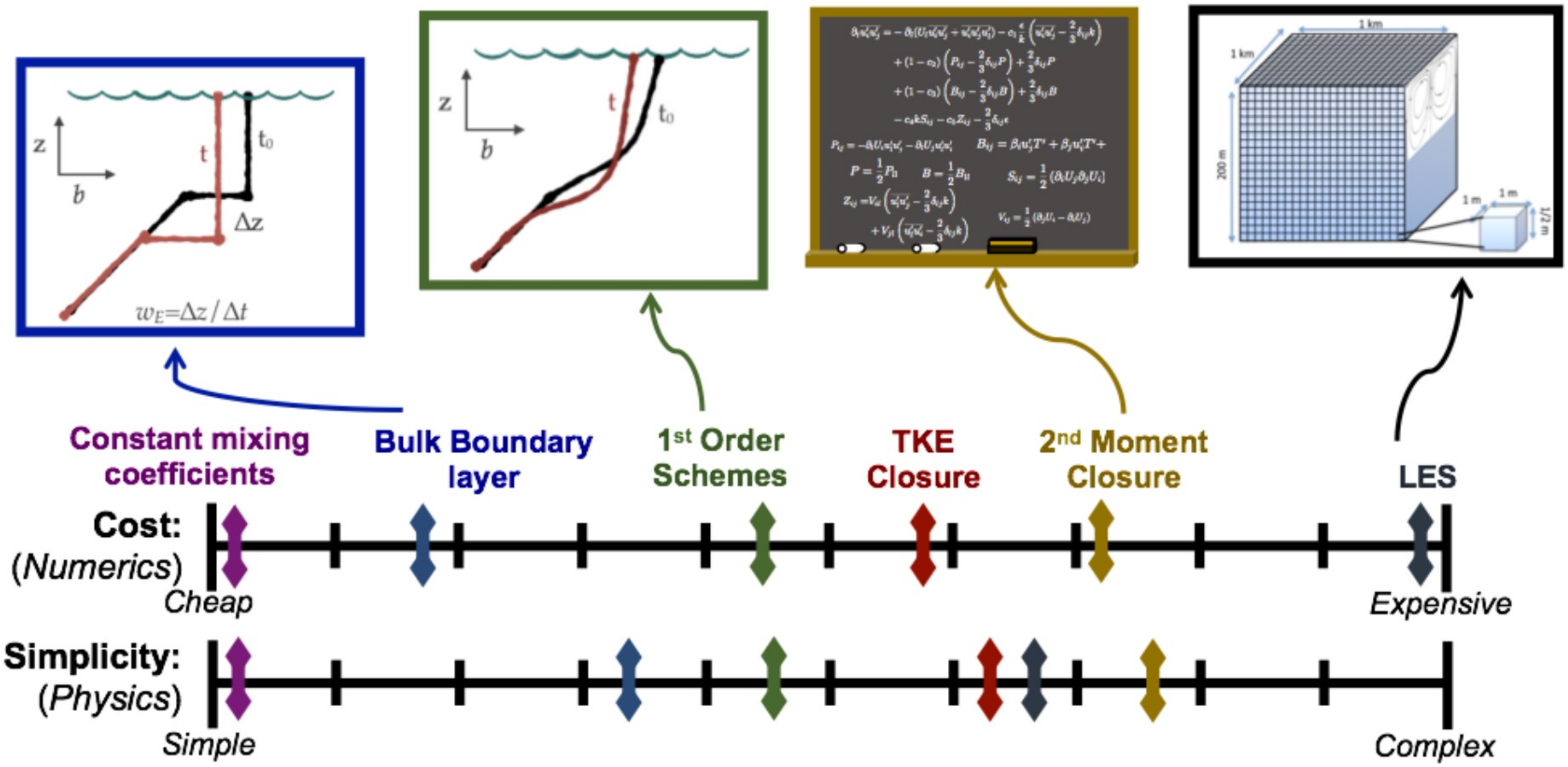
1. Neural networks - within the existing energetics based physics framework to improve vertical diffusivity in the OSBL.
2. Scheme implemented in MOM6.
3. JRA forced simulations performed: Bias reduction in shallow mixed layer depth and upper ocean stratification.

MLD Bias (summer)

Control *Neural networks*



Ocean Boundary Layer Parameterizations: Cost vs complexity

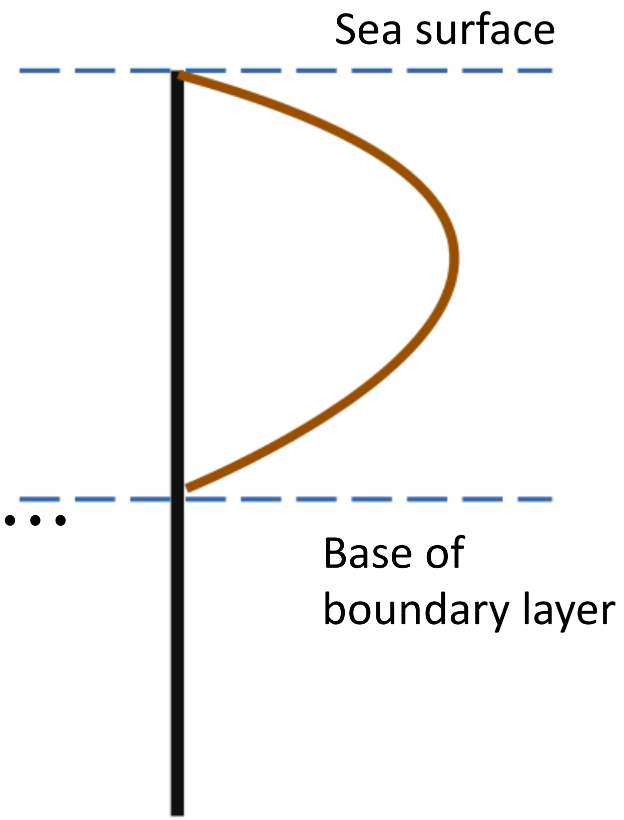


Vertical mixing:

$$\frac{D\phi}{Dt} = -\frac{\partial\langle w'\phi'\rangle}{\partial z}$$

$$\langle w'\phi'\rangle = -\kappa\frac{\partial\phi}{\partial z} + \cdots$$

$$\kappa = g(z)\cdot L\cdot v$$

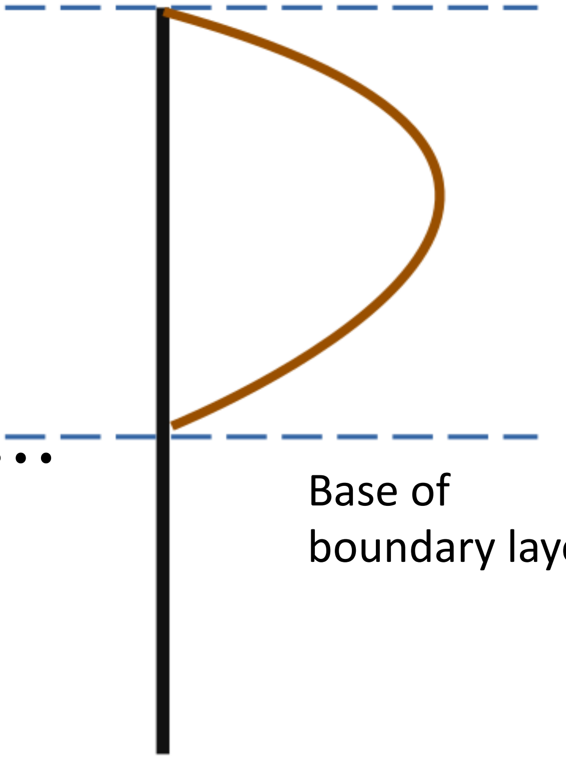


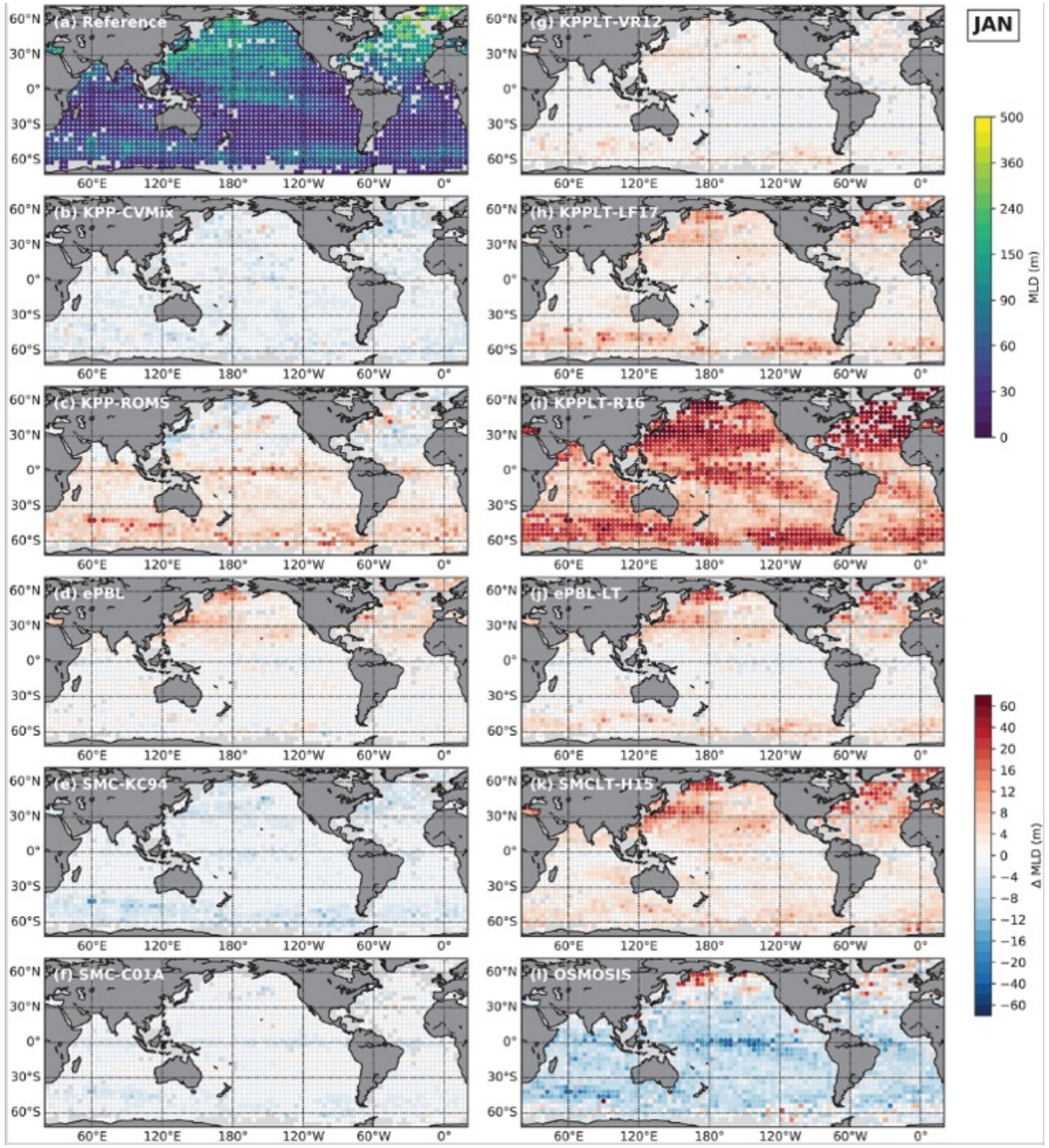
Vertical mixing:

$$\frac{D\phi}{Dt} = -\frac{\partial \langle w'\phi' \rangle}{\partial z}$$

$$\langle w'\phi' \rangle = -\kappa \frac{\partial \phi}{\partial z} + \cdots$$

$$\kappa = g(z) \cdot L \cdot v$$





Parameterizations disagree!
(Li et. al., 2019)

Existing vertical mixing parameterization: ePBL

Ocean Modelling 132 (2018) 112–129

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

A simplified energetics based planetary boundary layer (ePBL) approach for ocean climate simulations.

Brandon G. Reichl[*,a], Robert Hallberg[b,a]

[a] Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ, USA
[b] NOAA Geophysical Fluid Dynamics Laboratory, Princeton, NJ, USA

$$\int_{-h}^{0} \langle w'\phi' \rangle = G(f, u*, B_0, h)$$

$$L = (z)\left(\frac{h - z}{h}\right)^{\gamma}$$

$$\kappa = c_0 \cdot L(z) \cdot v(z)$$

Existing vertical mixing parameterization: ePBL

Ocean Modelling 132 (2018) 112–129

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

A simplified energetics based planetary boundary layer (ePBL) approach for ocean climate simulations.

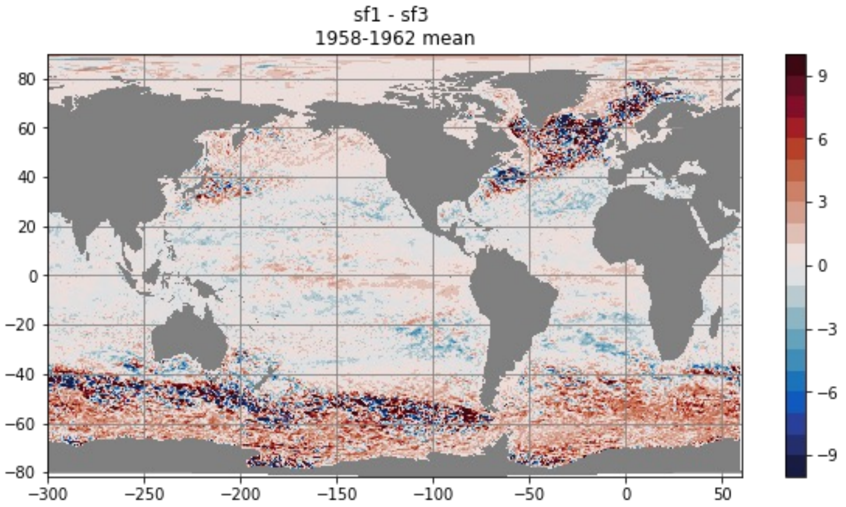
Brandon G. Reichl[*,a], Robert Hallberg[b,a]

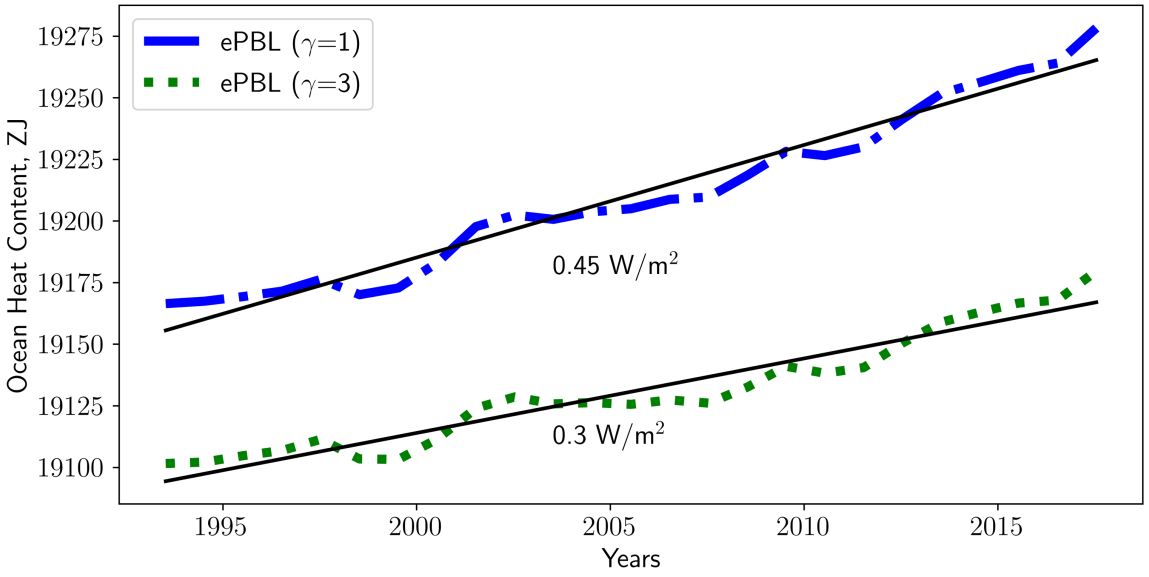
[a] Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ, USA
[b] NOAA Geophysical Fluid Dynamics Laboratory, Princeton, NJ, USA

$$\int_{-h}^{0} \langle w'\phi' \rangle = G(f, u*, B_0, h)$$

$$L = (z)\left(\frac{h-z}{h}\right)^{\gamma}$$

$$\kappa = c_0 \cdot L(z) \cdot v(z)$$





Existing vertical mixing parameterization: ePBL

Ocean Modelling 132 (2018) 112–129

Contents lists available at ScienceDirect

ELSEVIER

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

A simplified energetics based planetary boundary layer (ePBL) approach for ocean climate simulations.

Brandon G. Reichl[*,a], Robert Hallberg[b,a]

[a] Program in Atmospheric and Oceanic Sciences, Princeton University, Princeton, NJ, USA
[b] NOAA Geophysical Fluid Dynamics Laboratory, Princeton, NJ, USA

$$\int_{-h}^{0} \langle w'\phi' \rangle = G(f, u*, B_0, h)$$

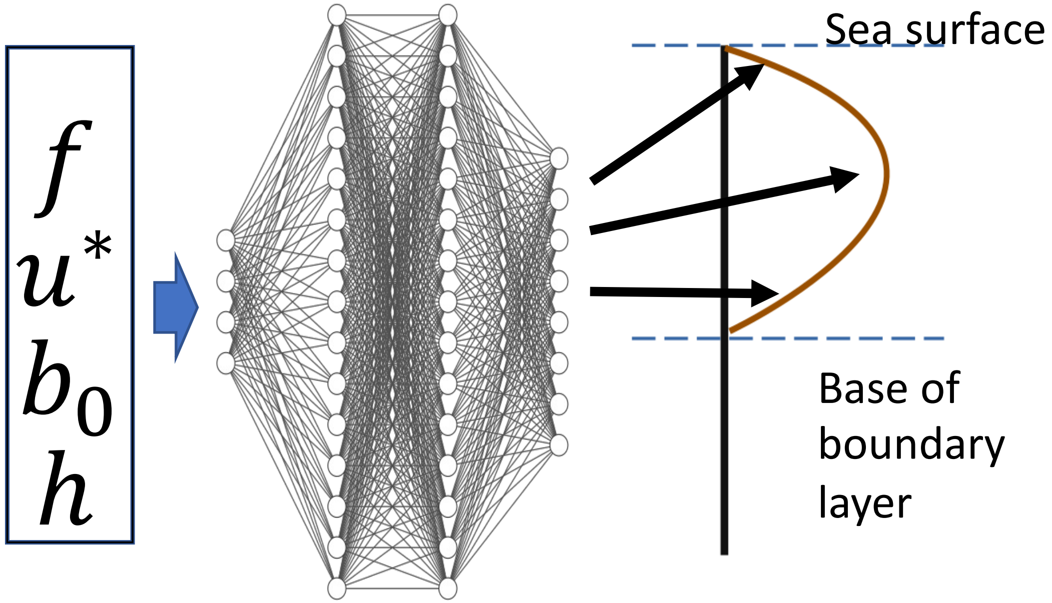
$$L = (z)\left(\frac{h-z}{h}\right)^{\gamma}$$

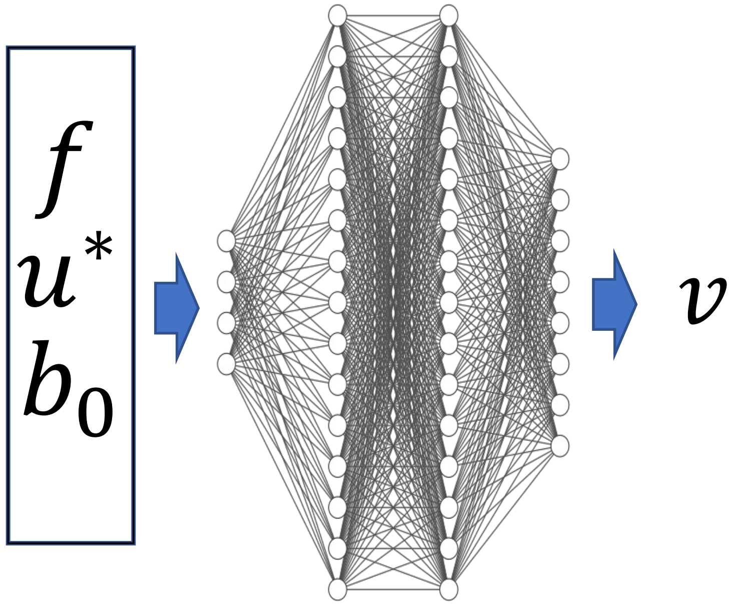
$$\kappa = c_0 \cdot L(z) \cdot v(z)$$

→ ***Replace this ad-hoc assumption with machine learning***

Neural network approach:

$$\kappa = g(\sigma) \cdot h \cdot v$$





Training data: $k - \epsilon$ model in GOTM

Surface Heat Flux	-600 to +600 W/m^2
Surface Wind Stress	0 to 1.2 N/m^2
Latitude	-90^o to +90^o
Boundary Layer Depth	20 m to 300 m

Training, Implementation in MOM6:

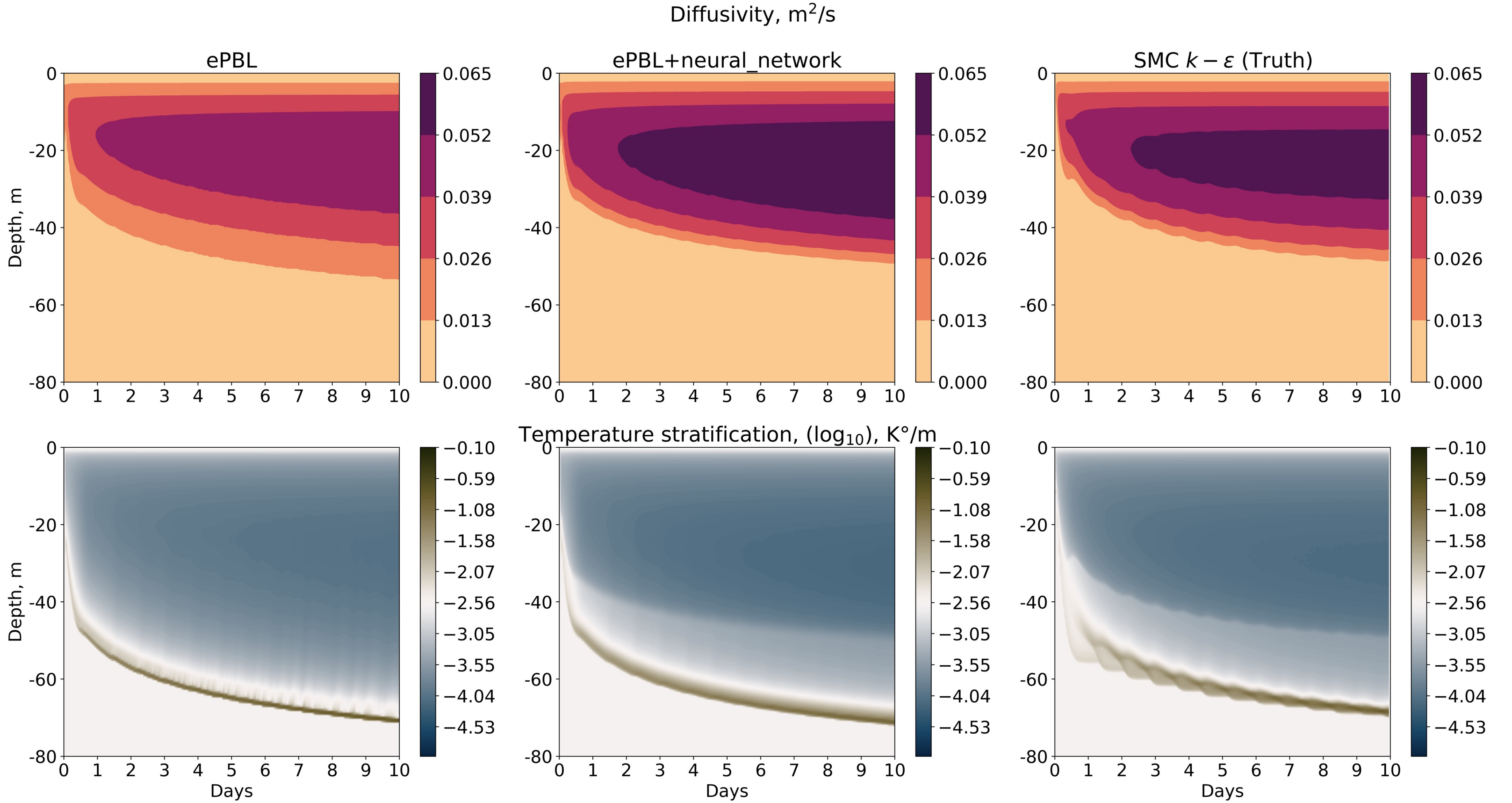
1. Pytorch used for training the networks
2. Networks are small, consists of weights and biases – saved them in netCDF files alongwith normalization values (means, standard variations).
3. Inference (matrix multiplications) algorithm coded in Fortran in the existing ePBL module of MOM6.
4. NN subroutines use h from ePBL --> energetic constraints of original scheme maintained.

Implementation in MOM6 (single column results):

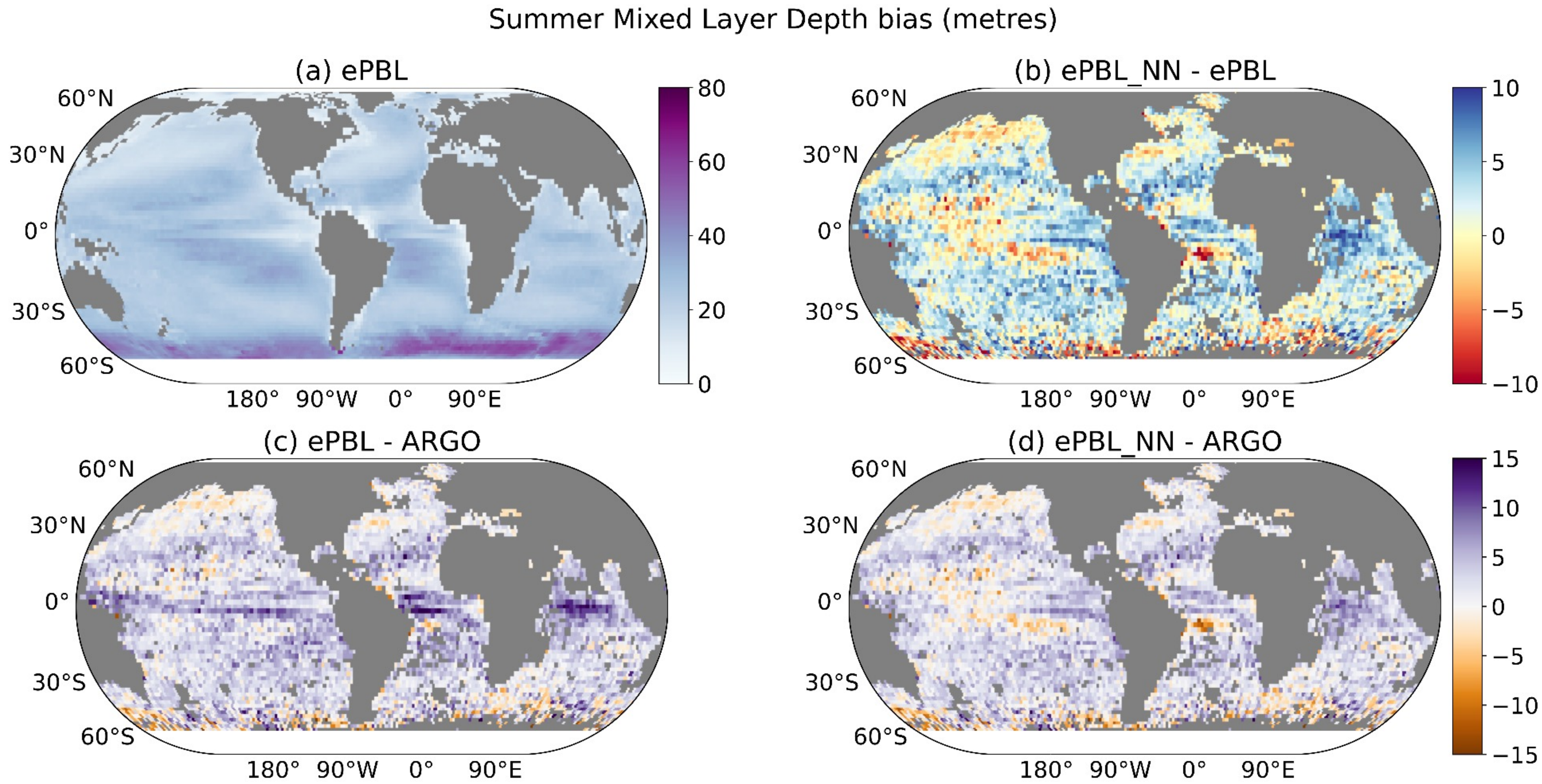
Surface shear,
Surface heating

Diffusivity

$$\frac{\partial T}{\partial z}$$



JRA forced simulations results:

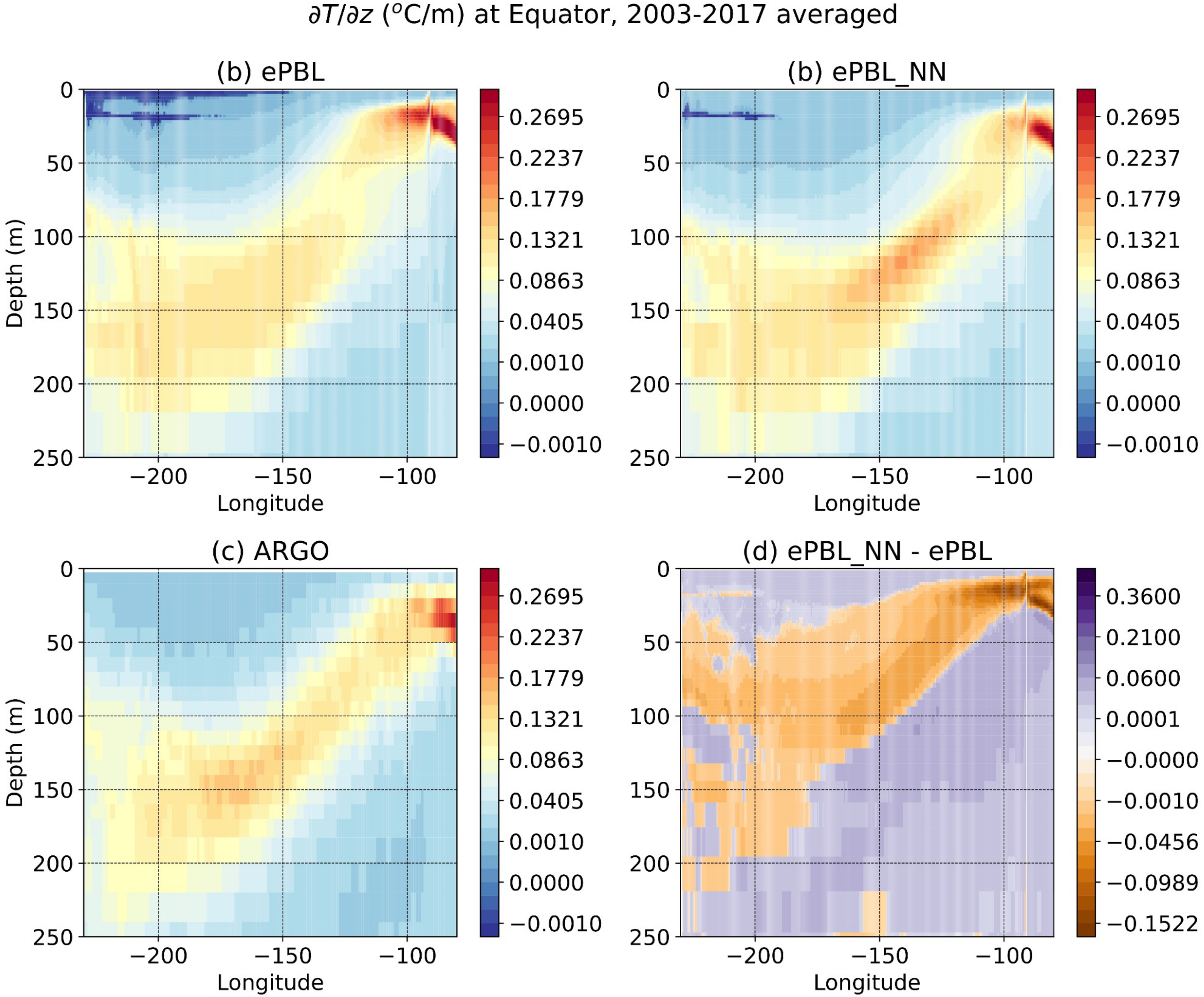


JRA forced simulations results:

Control run with physics based parameterization

Physics based parameterization enhanced with neural networks

ARGO observations

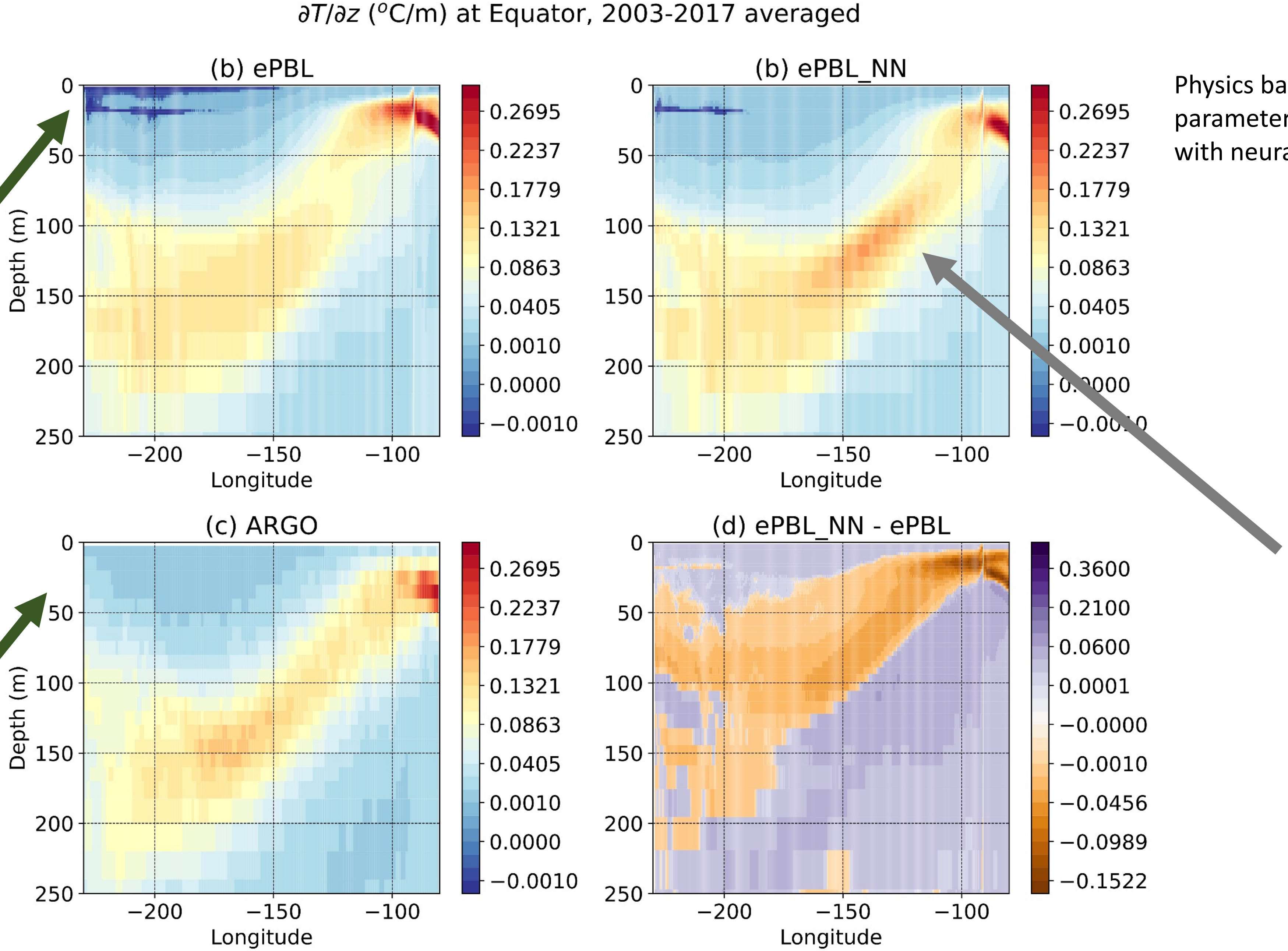


JRA forced simulations results:

Control run with physics based parameterization

Physics based parameterization enhanced with neural networks

ARGO observations



Conclusions:

1. A neural network framework has been developed which uses second moment closure for training and gives output of diffusivity for the OSBL.

2. Neural network shows skill in predicting the shape function.

3. Scheme has been implemented in GFDL's MOM6 within the existing EPBL physics framework. Only ad-hoc portions of original scheme replaced with machine learning to maintain the energy constraints of the original scheme.

4. Machine learning allows ePBL to use diffusivity from SMC with coarse temporal resolution as compared to directly using a SMC which requires fine time stepping.

Thank You