Data-driven parameterization of mesoscale eddies using the Eliassen-Palm flux

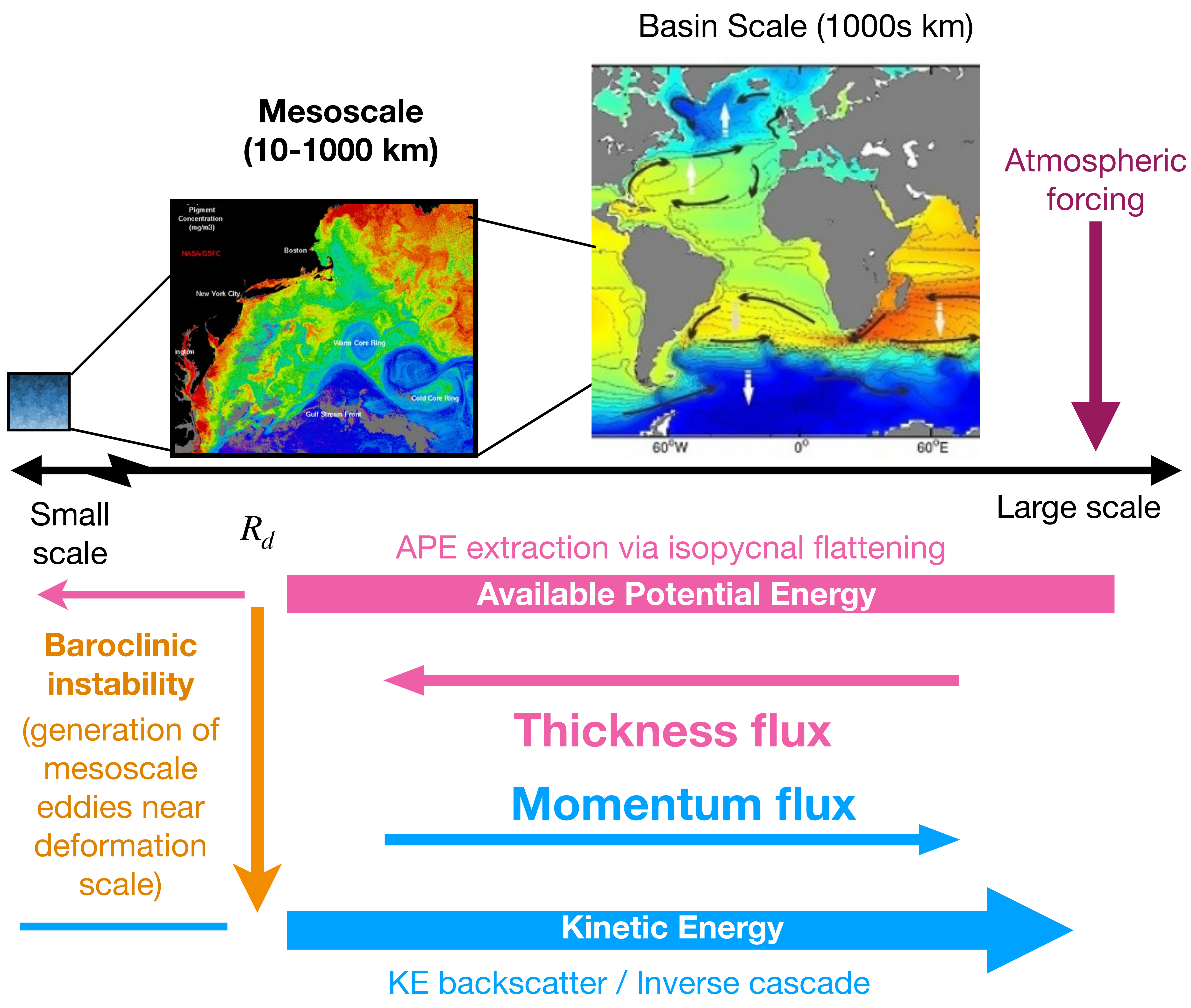
CLIVAR-OMDP Workshop, September 2024

Kelsey Everard[1], Pavel Perezhogin[1], Dhruv Balwada[2], Alistair Adcroft[3,4], Laure Zanna[1]

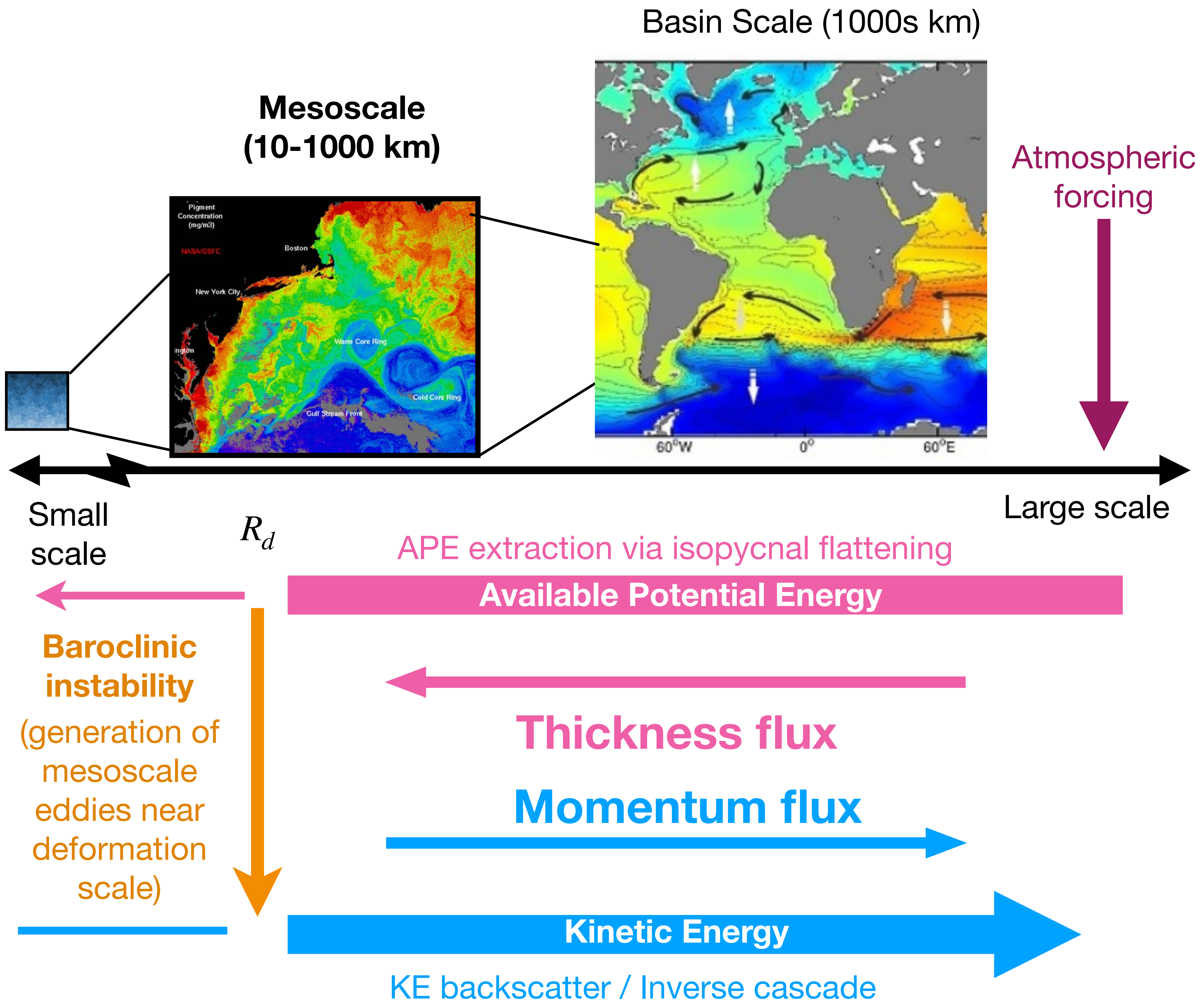
[1] Courant Institute of Mathematical Sciences, New York University
[2] Lamont-Doherty Earth Observatory, Columbia University
[3] Geophysical Fluid Dynamics Laboratory, National Oceanic and Atmospheric Administration
[4] Atmospheric and Oceanic Sciences Program, Princeton University

@M2LInES
https://m2lines.github.io

Mesoscale Eddy Energy Cycle



Energy cycle: A goal



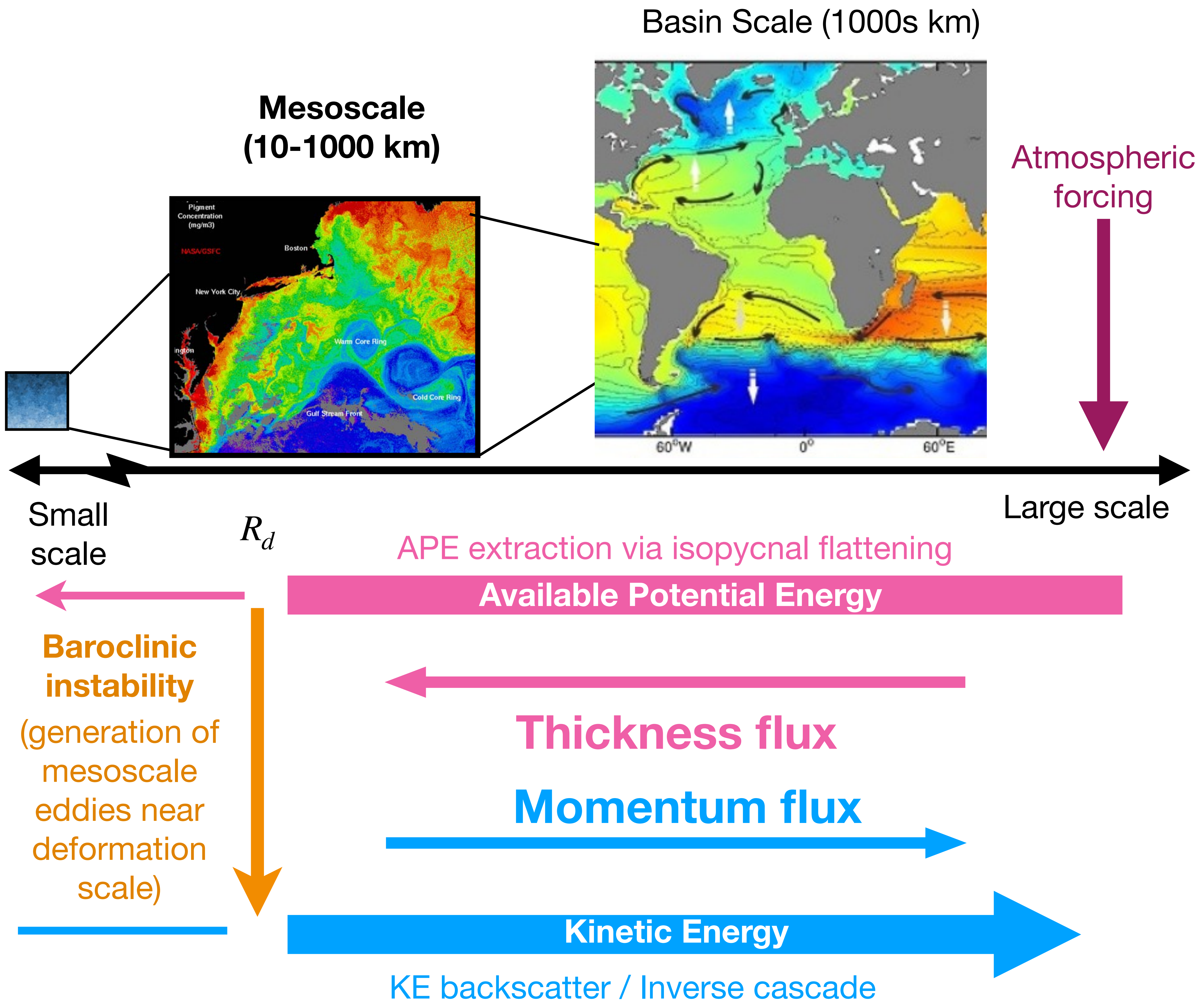
Down-scale energy transfer via APE extraction and up-scale energy transfer via KE backscatter are often parameterised separately

e.g., Kraichnan 1976; Frederiksen 1997; Berner 2009; Jansen 2014

e.g., Gent McWilliams 1990; Griffies 1998

But they're two essential aspects of one process

Energy cycle: A goal



Down-scale energy transfer via APE extraction and up-scale energy transfer via KE backscatter are often parameterised separately

e.g., Kraichnan 1976; Frederiksen 1997; Berner 2009; Jansen 2014

e.g., Gent McWilliams 1990; Griffies 1998

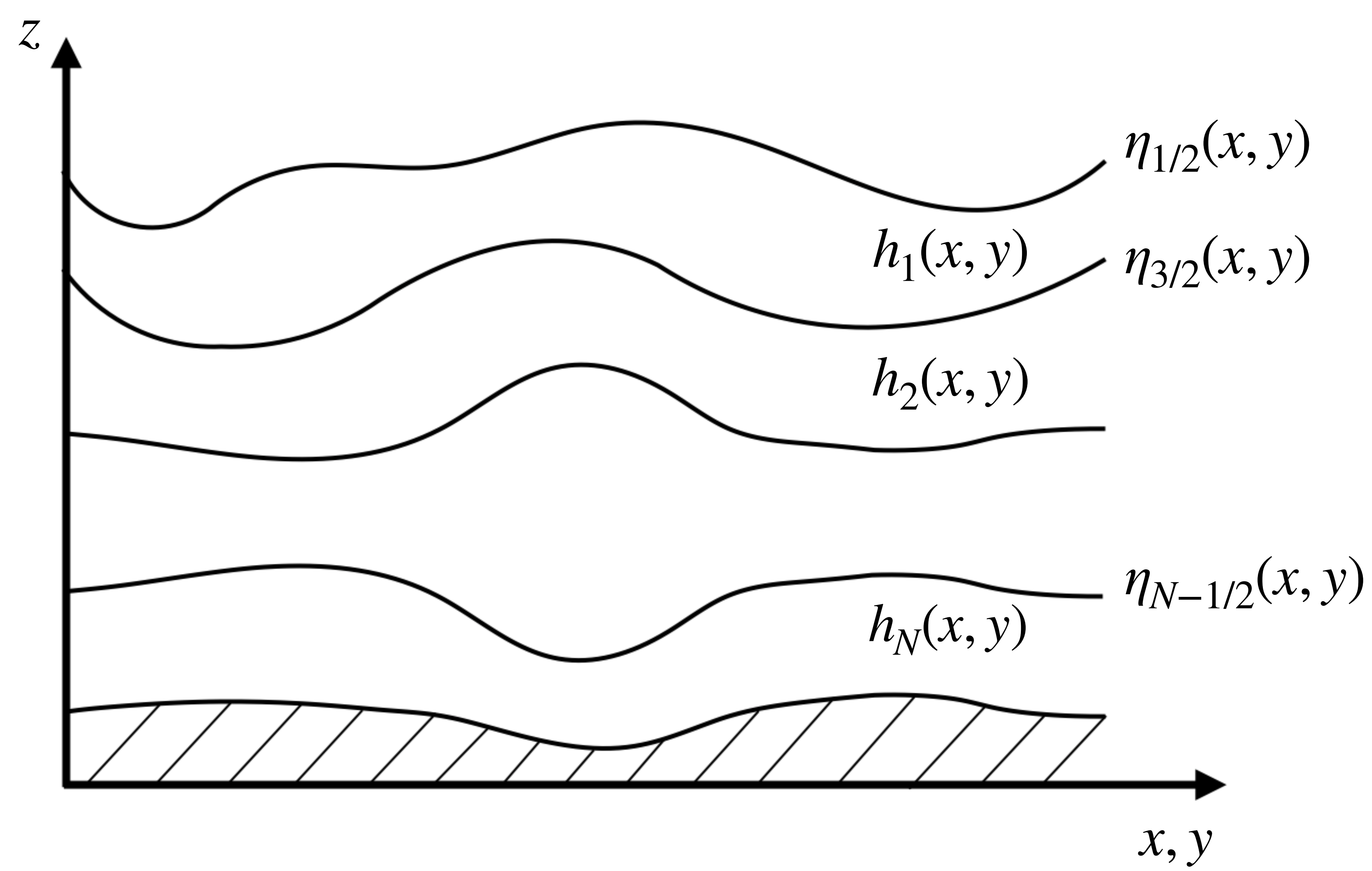
But they're two essential aspects of one process

Parameterisation of Ocean mesoscale eddies whereby **momentum** and **thickness** fluxes are considered ***together***

Energy + Momentum + Thickness = Eliassen-Palm Flux

To capture the full eddy energy cycle, we target the Eliassen-Palm Flux

Thickness-weighted average framework: $\widehat{u}_n = \dfrac{\overline{h_n u_n}}{\overline{h}_n}$ Thickness weighted average



$$\frac{\partial \widehat{u}_n}{\partial t} + \widehat{u}_n \frac{\partial \widehat{u}_n}{\partial x} + \widehat{v}_n \frac{\partial \widehat{u}_n}{\partial y} - f \widehat{v}_n + \frac{\partial \overline{p}_n}{\partial x} - \widehat{F}^{(u)}_n = -\nabla_3 \cdot \widehat{E}^{(u)}_n$$

$$\frac{\partial \widehat{v}_n}{\partial t} + \widehat{u}_n \frac{\partial \widehat{v}_n}{\partial x} + \widehat{v}_n \frac{\partial \widehat{v}_n}{\partial y} + f \widehat{u}_n + \frac{\partial \overline{p}_n}{\partial y} - \widehat{F}^{(v)}_n = -\nabla_3 \cdot \widehat{E}^{(v)}_n$$

$$\widehat{\mathbf{E}}_n = \left(\widehat{E}^{(u)}_n, \widehat{E}^{(v)}_n\right) = \begin{pmatrix} \left(\widehat{u''^2_n} + \frac{1}{2\overline{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}}\right) & \left(\widehat{u''_n v''_n}\right) \\ \left(\widehat{u''_n v''_n}\right) & \left(\widehat{v''^2_n} + \frac{1}{2\overline{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}}\right) \\ \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial x} p'_{n-1}}\right) & \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial y} p'_{n-1}}\right) \end{pmatrix}$$

Eliassen-Palm Flux

Zonal | Meridional

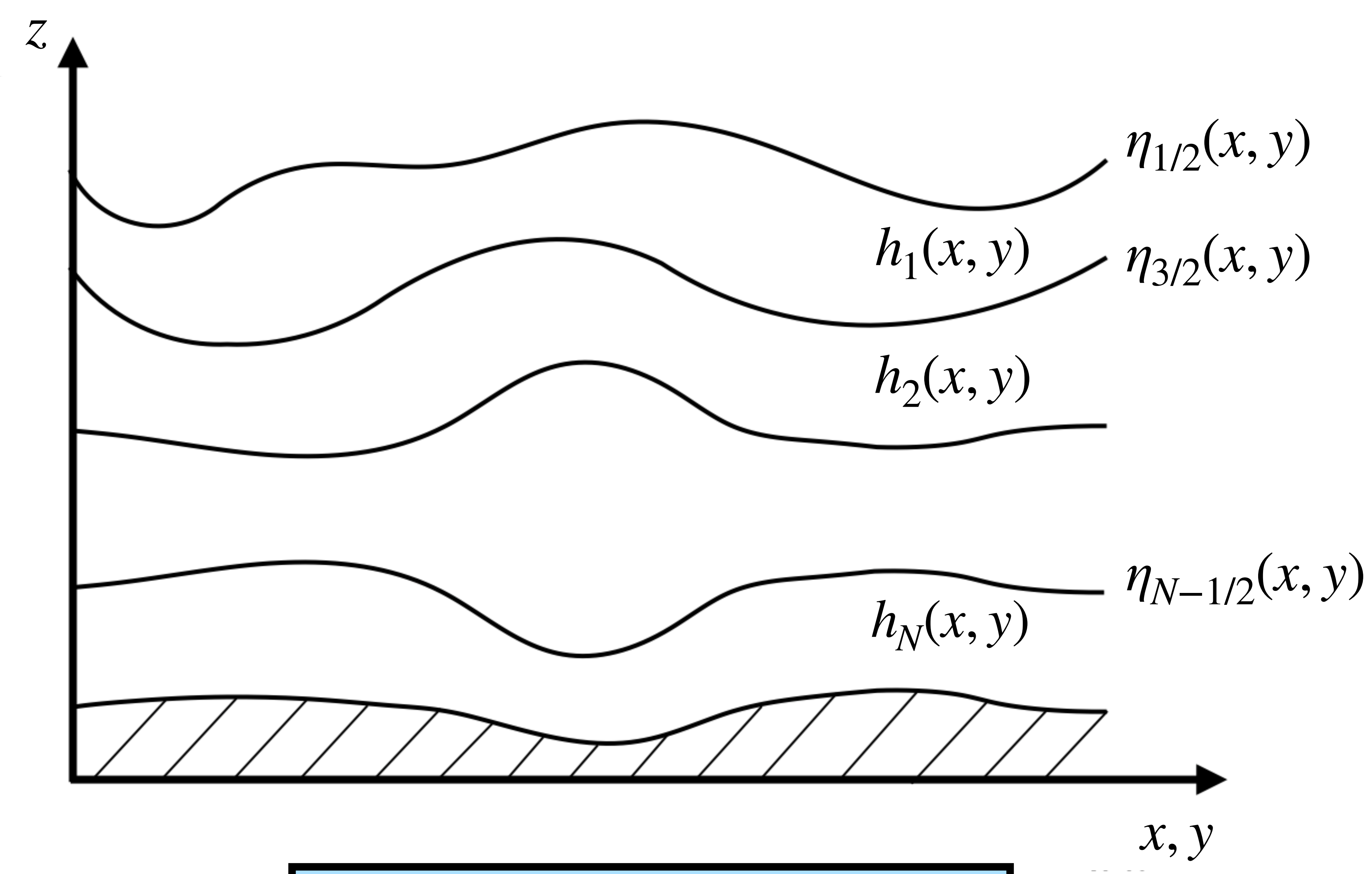
Thickness fluxes projected onto momentum equations!

e.g. Marshall et al. (2012); Young (2012);
Madison and Marshall (2013); Loose et al (2023)

Energy + Momentum + Thickness = Eliassen-Palm Flux

To capture the full eddy energy cycle, we target the Eliassen-Palm Flux

Thickness-weighted average framework: $\widehat{u}_n = \dfrac{\overline{h_n u_n}}{\overline{h}_n}$ Thickness weighted average



$$\frac{\partial \widehat{u}_n}{\partial t} + \widehat{u}_n \frac{\partial \widehat{u}_n}{\partial x} + \widehat{v}_n \frac{\partial \widehat{u}_n}{\partial y} - f \widehat{v}_n + \frac{\partial \overline{p}_n}{\partial x} - \widehat{F}_n^{(u)} = - \nabla_3 \cdot \widehat{E}_n^{(u)}$$

$$\frac{\partial \widehat{v}_n}{\partial t} + \widehat{u}_n \frac{\partial \widehat{v}_n}{\partial x} + \widehat{v}_n \frac{\partial \widehat{v}_n}{\partial y} + f \widehat{u}_n + \frac{\partial \overline{p}_n}{\partial y} - \widehat{F}_n^{(v)} = - \nabla_3 \cdot \widehat{E}_n^{(v)}$$

$$\widehat{\mathbf{E}}_n = \left(\widehat{E}_n^{(u)}, \widehat{E}_n^{(v)} \right) = \begin{pmatrix} \left(\widehat{u_n''^2} + \frac{1}{2\overline{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}} \right) & \left(\widehat{u_n'' v_n''} \right) \\ \left(\widehat{u_n'' v_n''} \right) & \left(\widehat{v_n''^2} + \frac{1}{2\overline{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}} \right) \\ \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial x} p'_{n-1}} \right) & \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial y} p'_{n-1}} \right) \end{pmatrix}$$

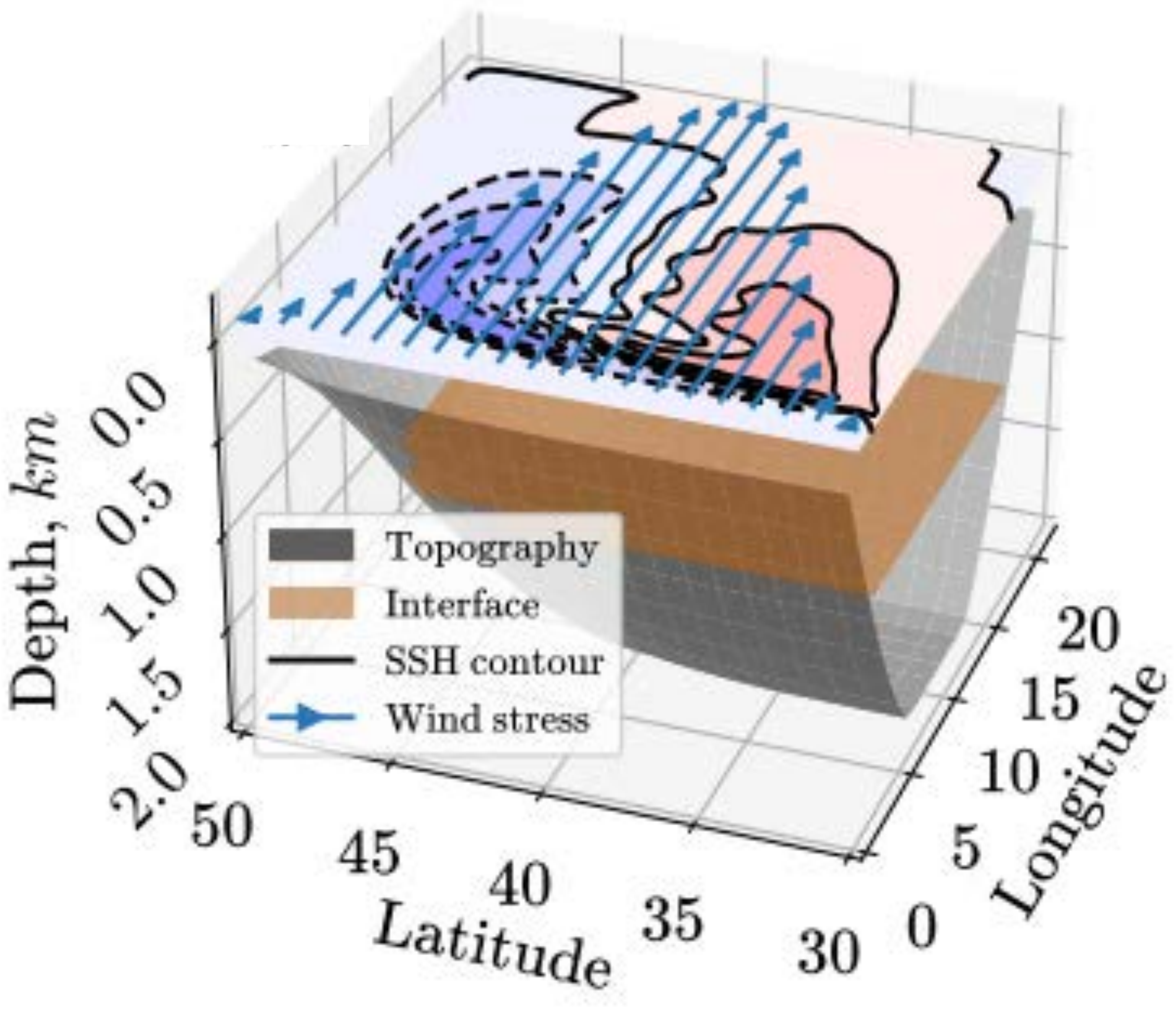
Zonal | Meridional

Upscale energy transfer via KE backscatter

Downscale energy transfer via APE extraction

e.g. Marshall et al. (2012); Young (2012);
Madison and Marshall (2013); Loose et al (2023)

MOM6 Double Gyre Configuration



Perezhogin et al (2024)

Commonly used configuration as stepping stone e.g.,

Dhruv's presentation;

Perezhogin et al (2024);

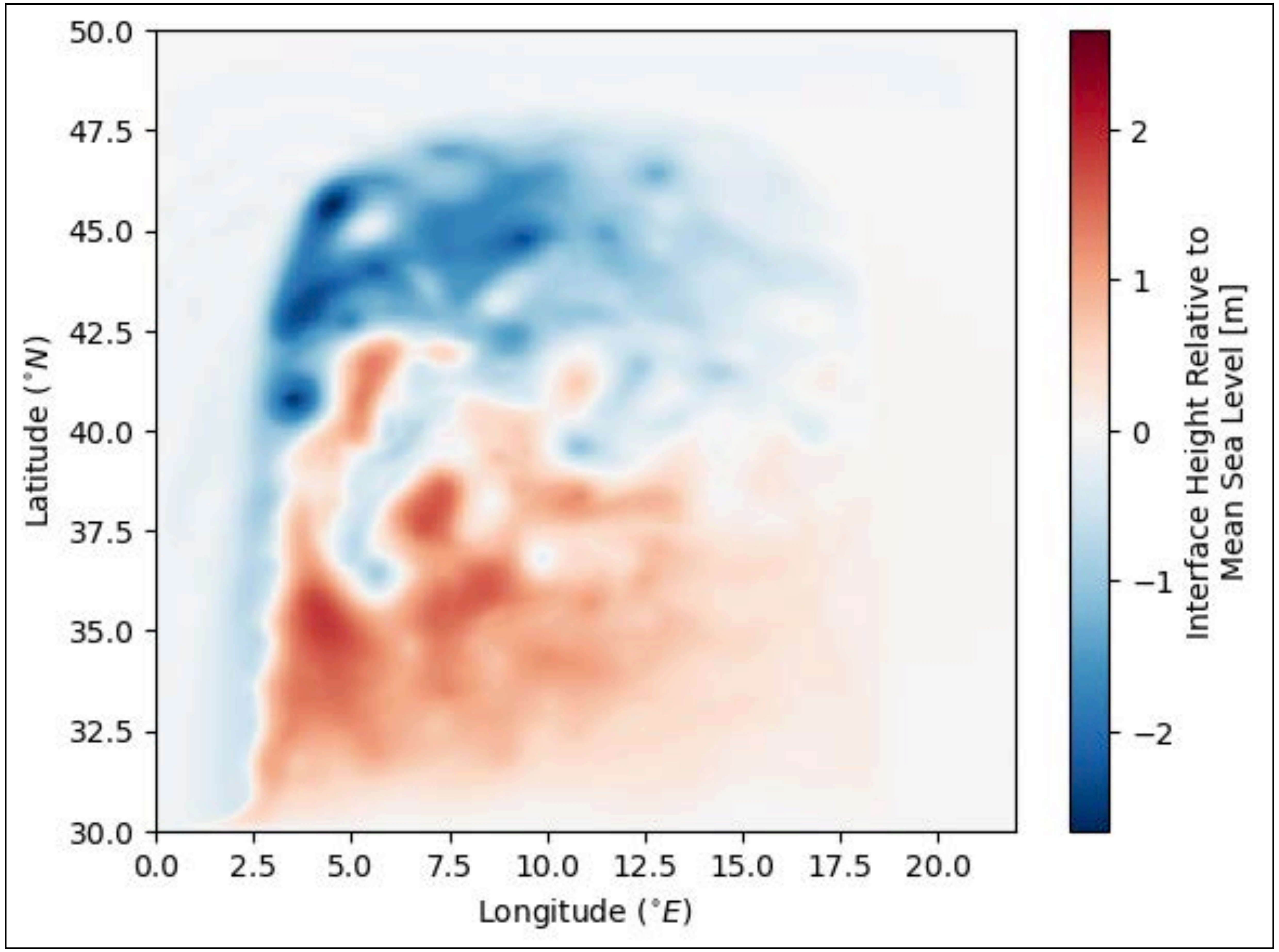
Zhang et al (2023)

GFDL MOM6 ocean model in Double Gyre configuration

- Two layers
- Initialized from rest
- Driven by wind and equilibrated by bottom friction
- Biharmonic Smagorinsky model

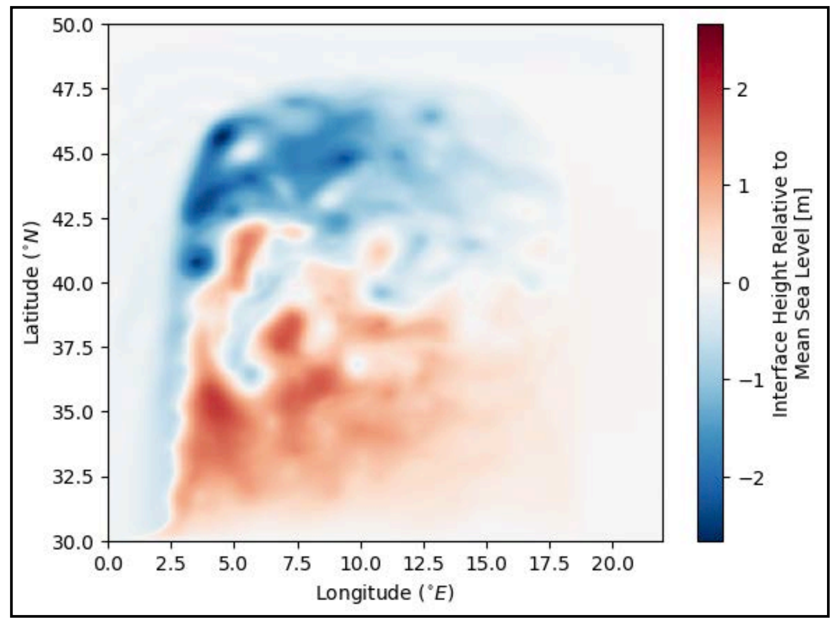
Parameterisation Development: 1) High resolution data

Double Gyre simulation data at $1/32^{\circ}$ resolution

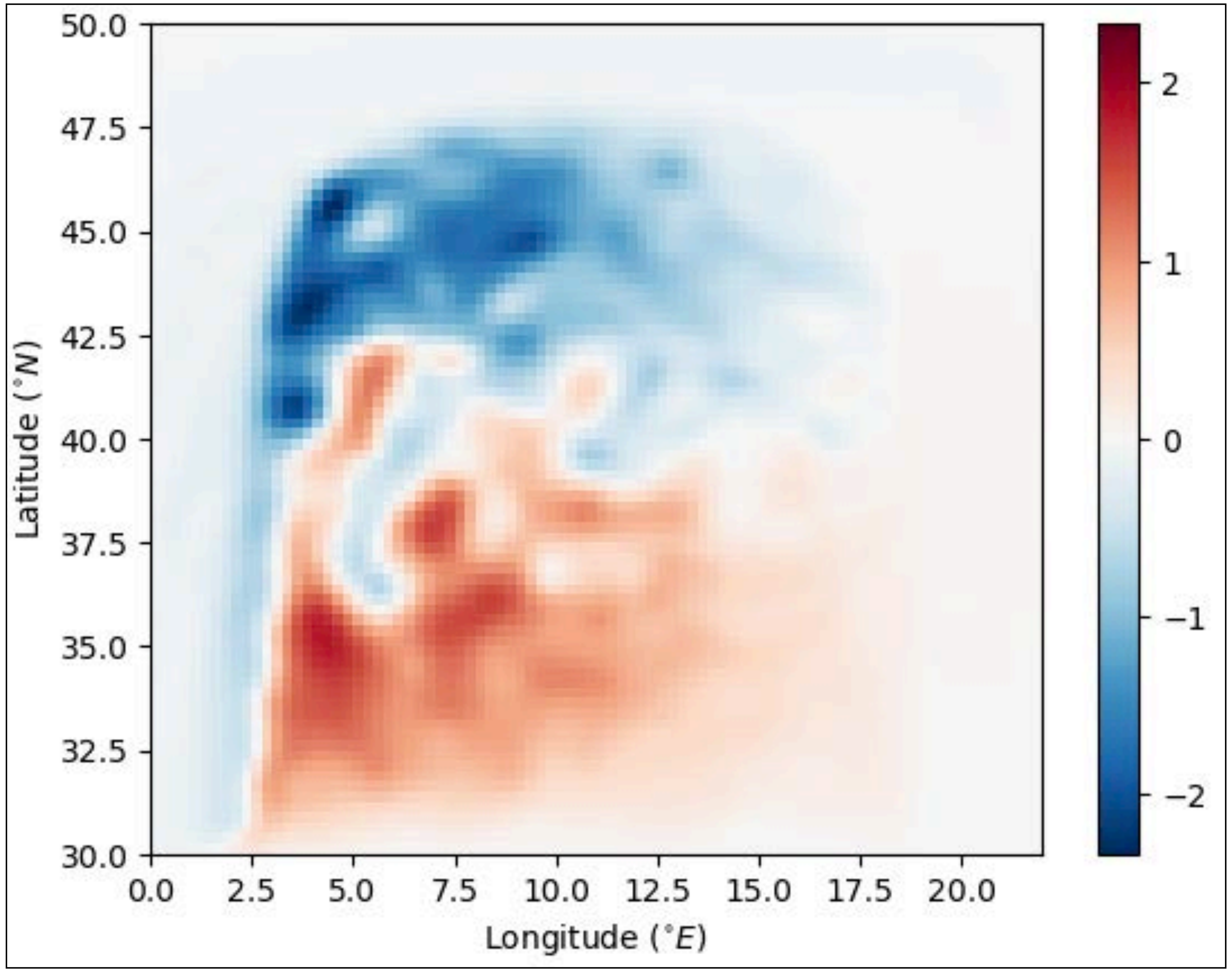


Parameterisation Development: 2) Filter and coarsen

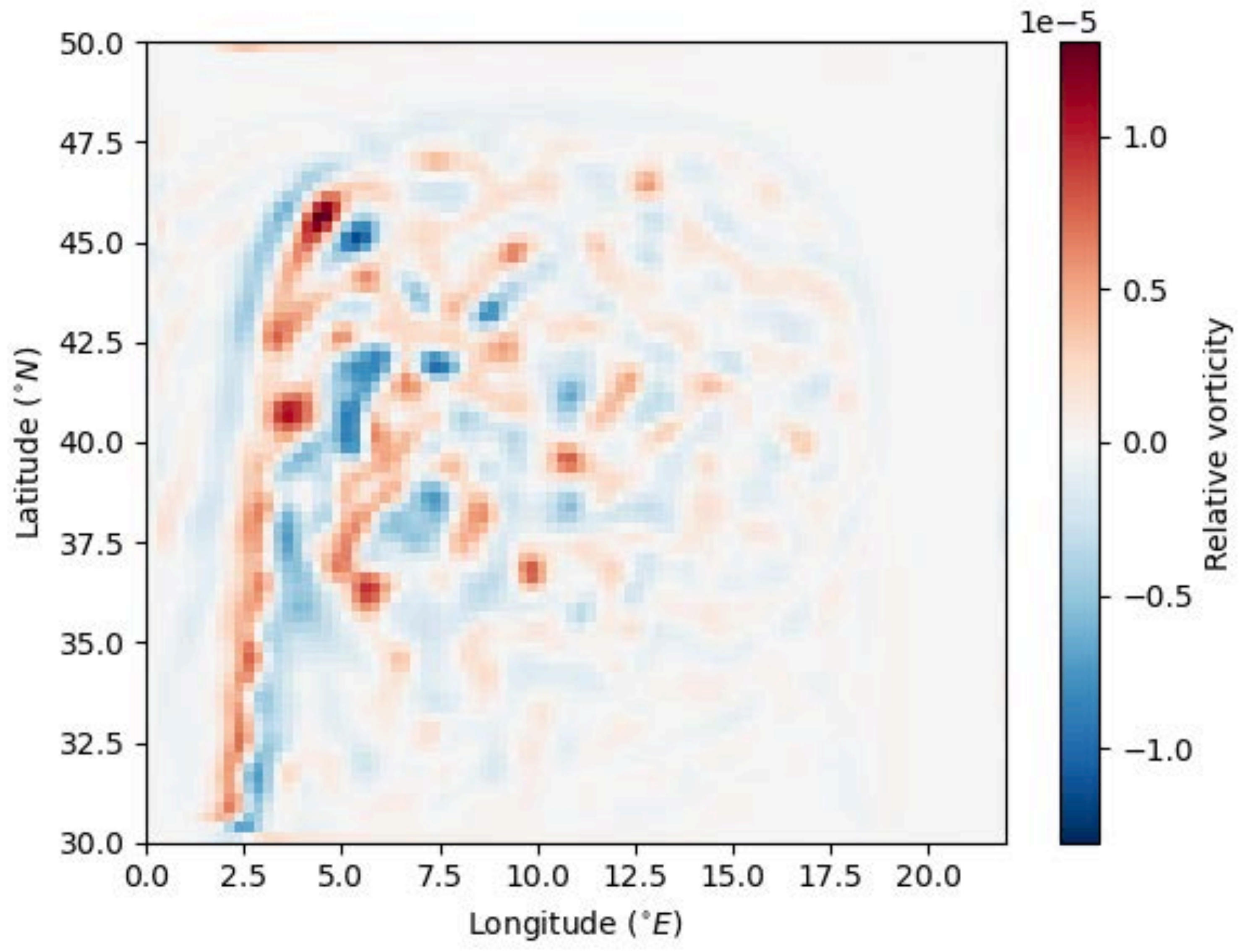
Double Gyre simulation data at $1/32°$ resolution



Filter and coarsen to $1/4°$



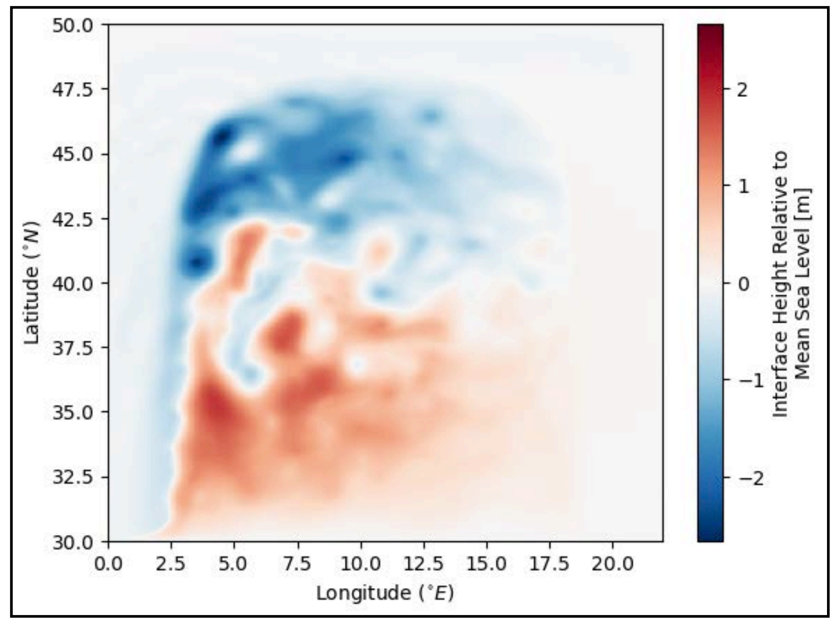
Parameterisation Development: 3) Coarse flow-field representations

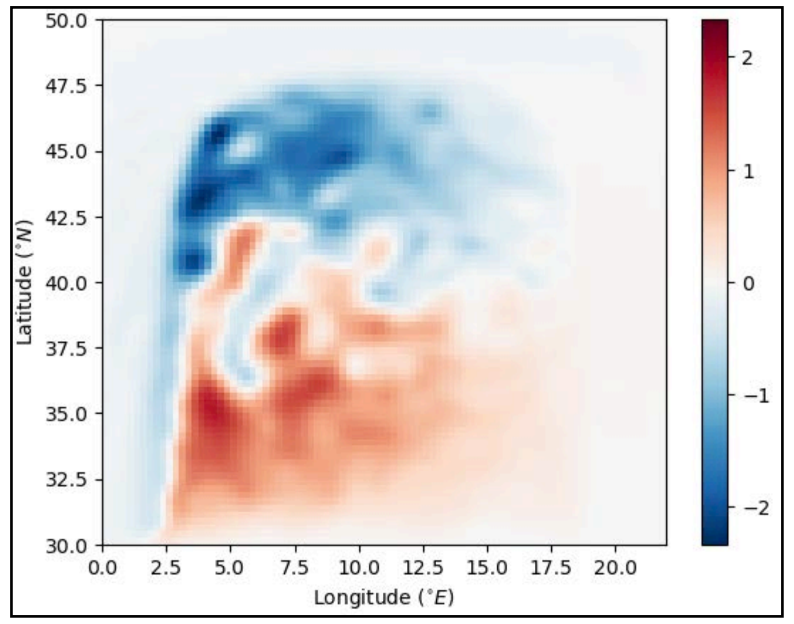


Calculate Spatial gradients of flow field variables

Double Gyre simulation data at $1/32^\circ$ resolution

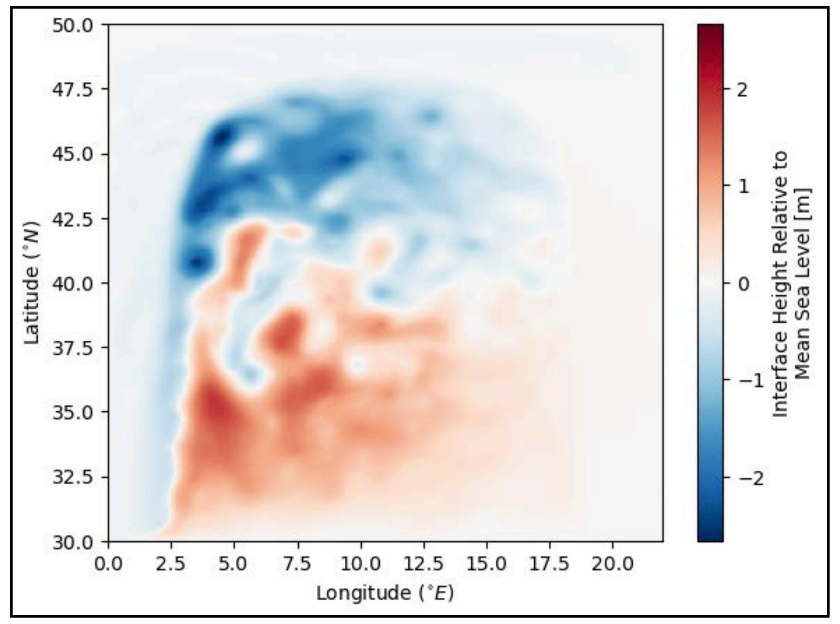
Filter and coarsen to $1/4^\circ$

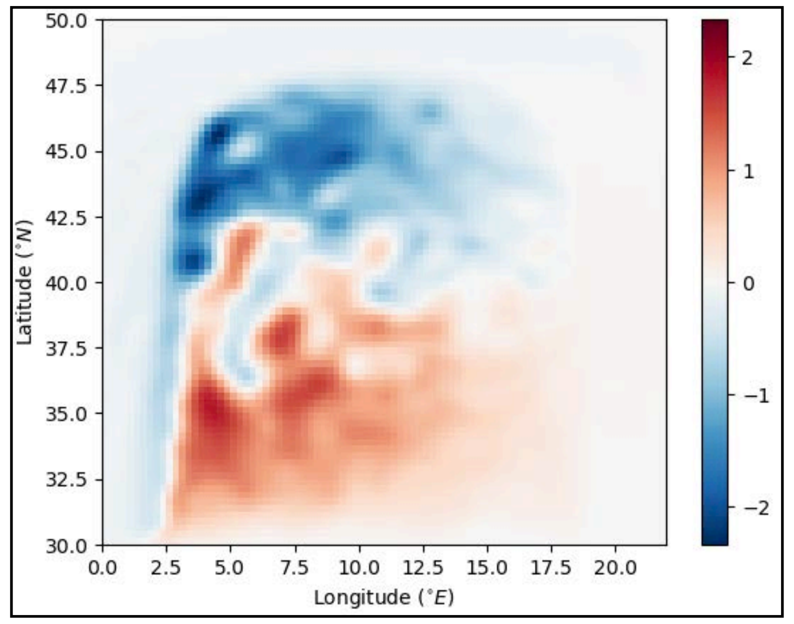




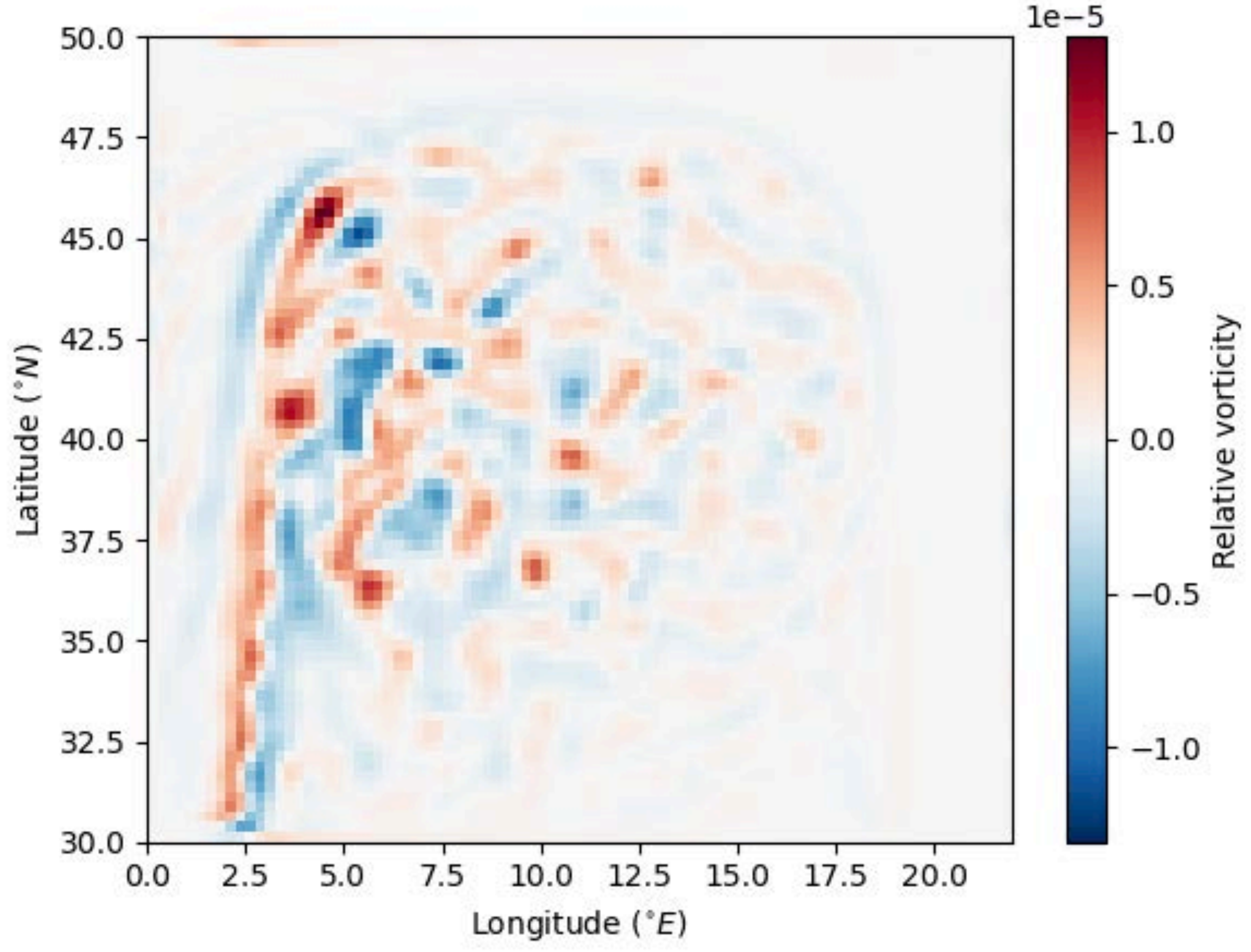
Parameterisation Development: 4) Diagnose Eliassen-Palm Fluxes

Double Gyre simulation data at $1/32°$ resolution → Filter and coarsen to $1/4°$



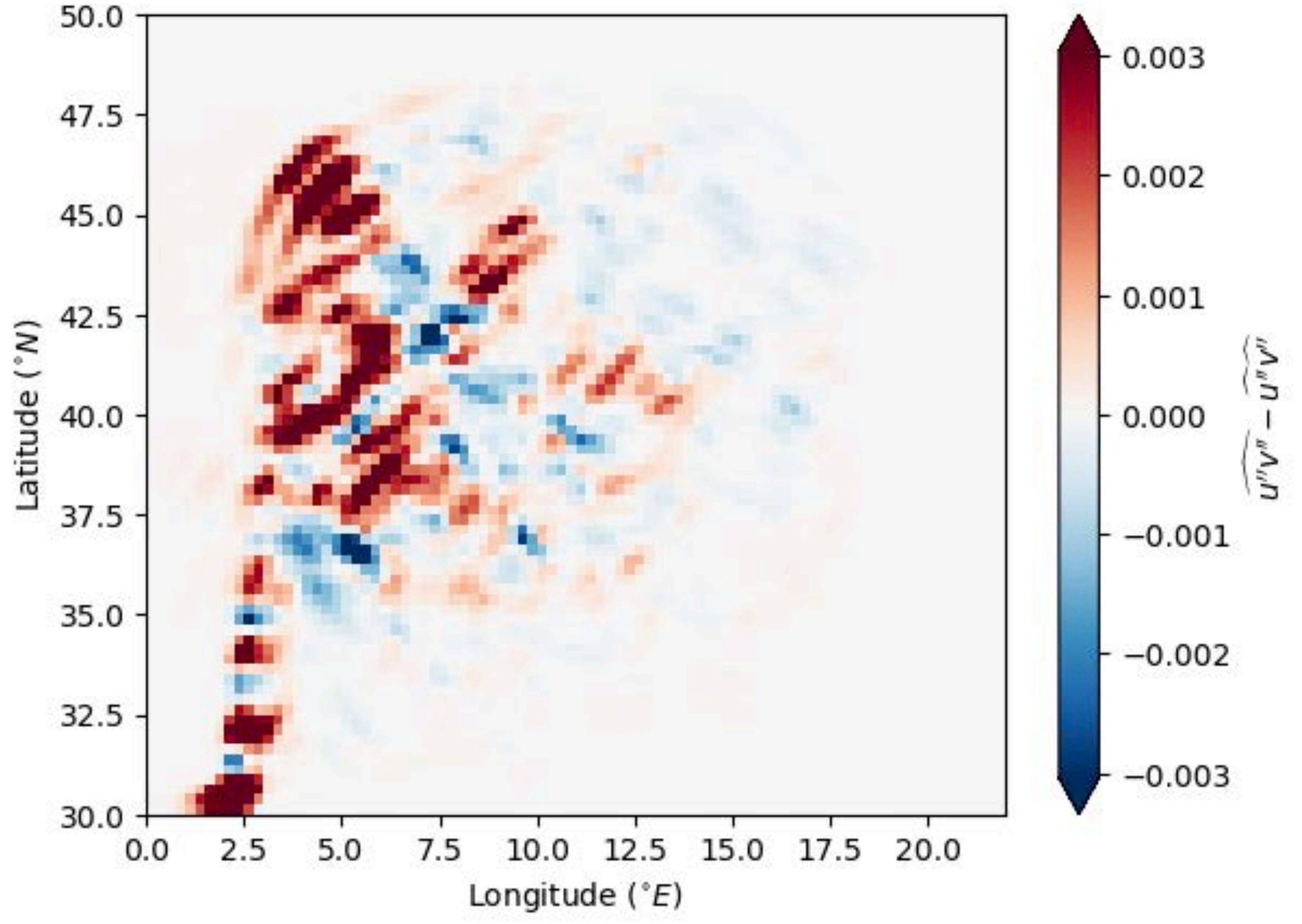


Calculate Spatial gradients of flow field variables

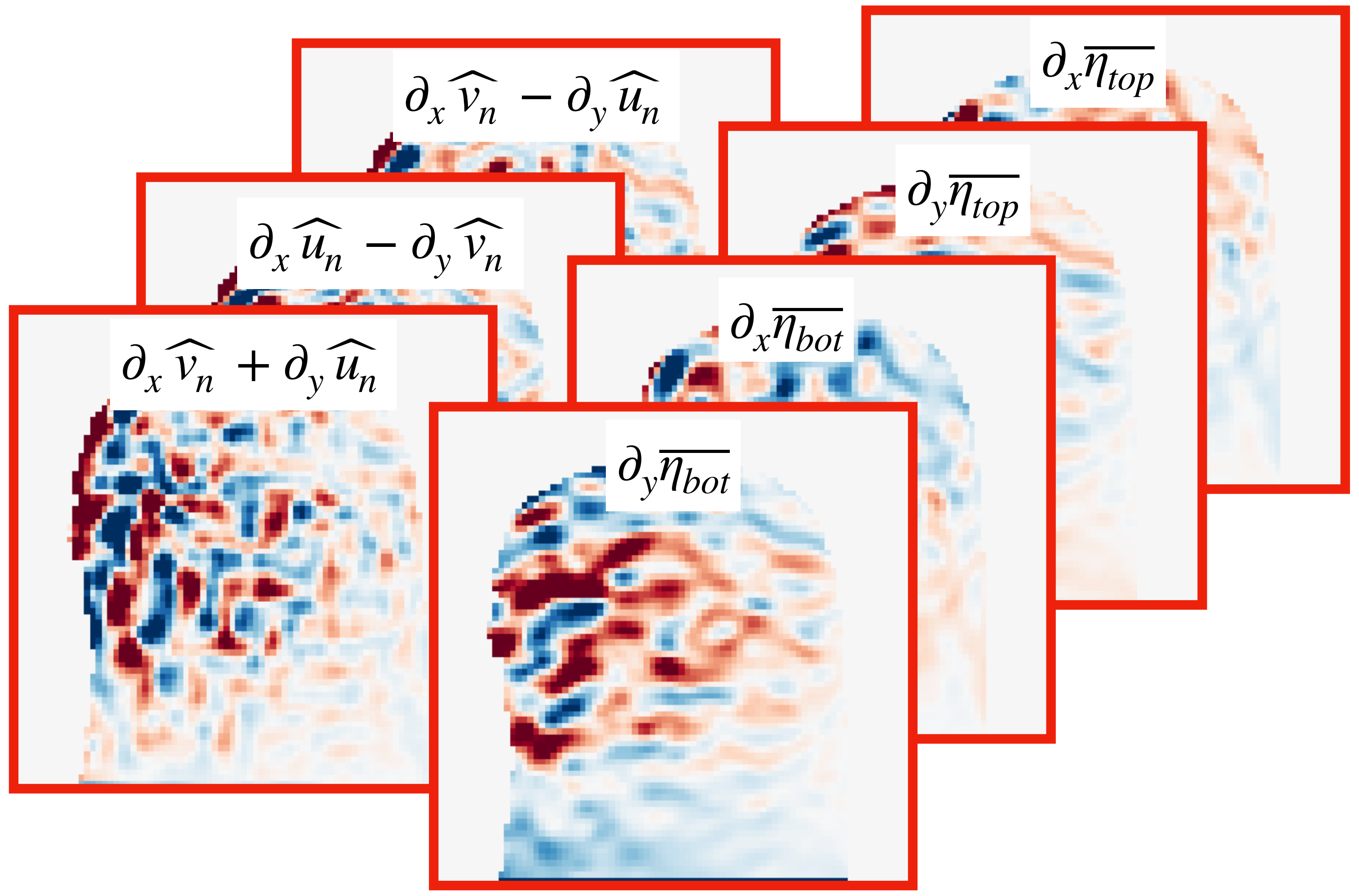


AND

Calculate Sub-filter EPF tensor components

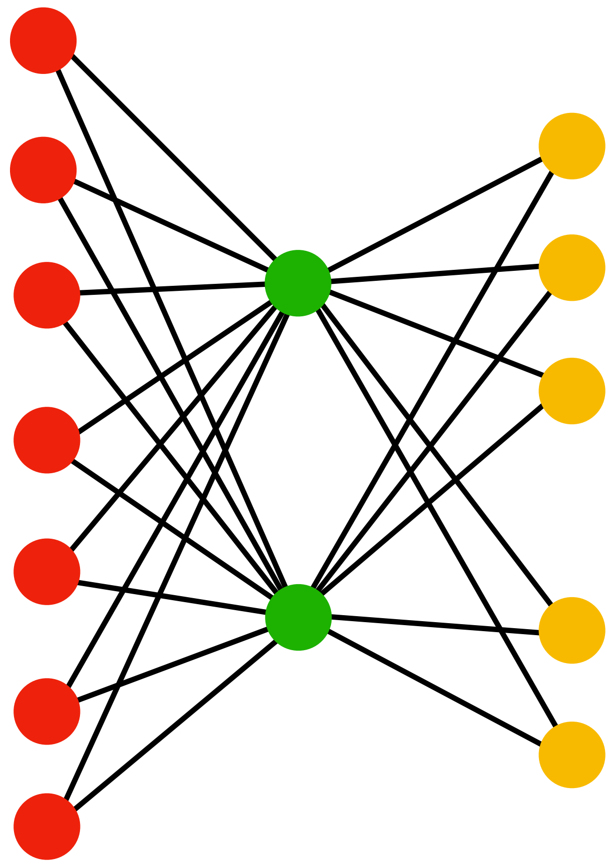


Parameterisation Development: 5) Artificial Neural Network

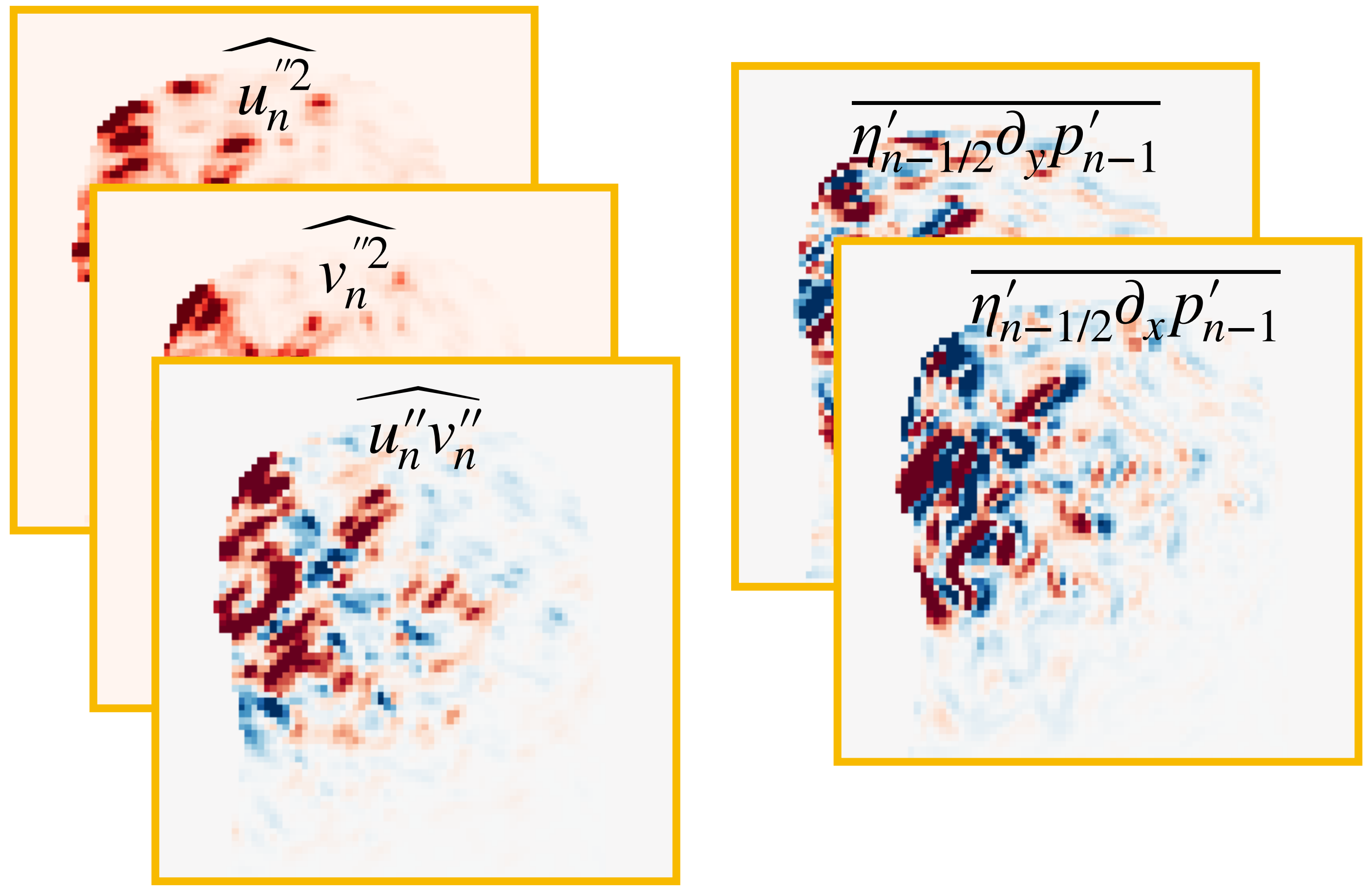


INPUTS:

- Coarse-grain derivatives of filtered flow variables



2 hidden layers
8 neurons each



OUTPUTS:

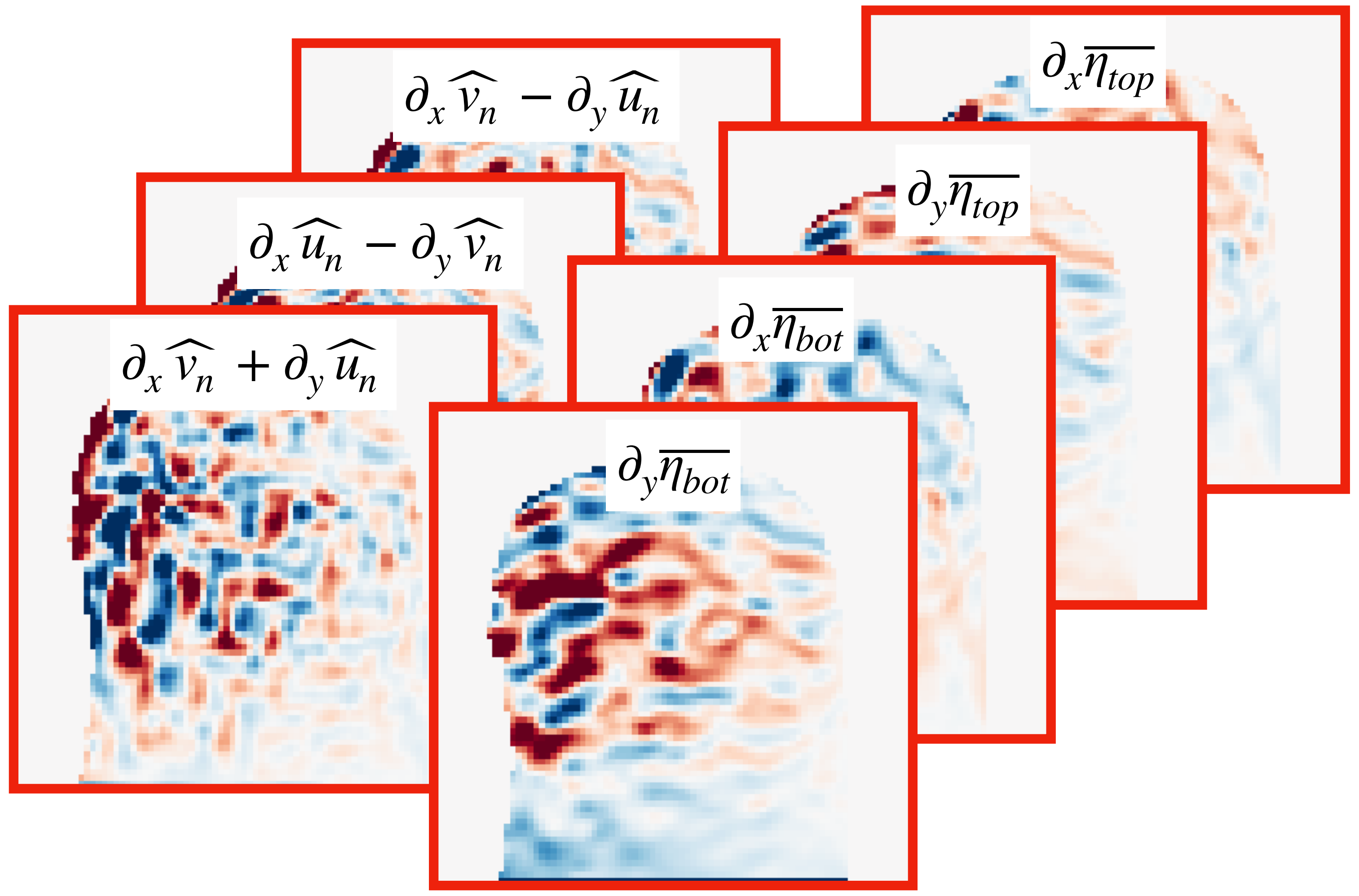
- Reynolds' stresses and dual form stresses

Trained on ~4×10^6 samples (288 snapshots)

~5×10^5 samples for validation (36 snapshots)

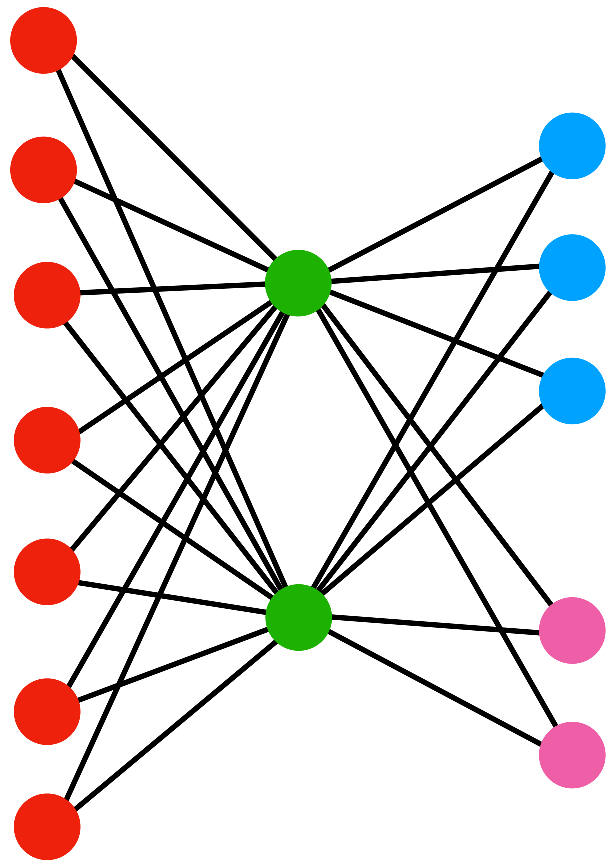
~5×10^5 samples for testing (36 snapshots)

Parameterisation Development: 5) Artificial Neural Network

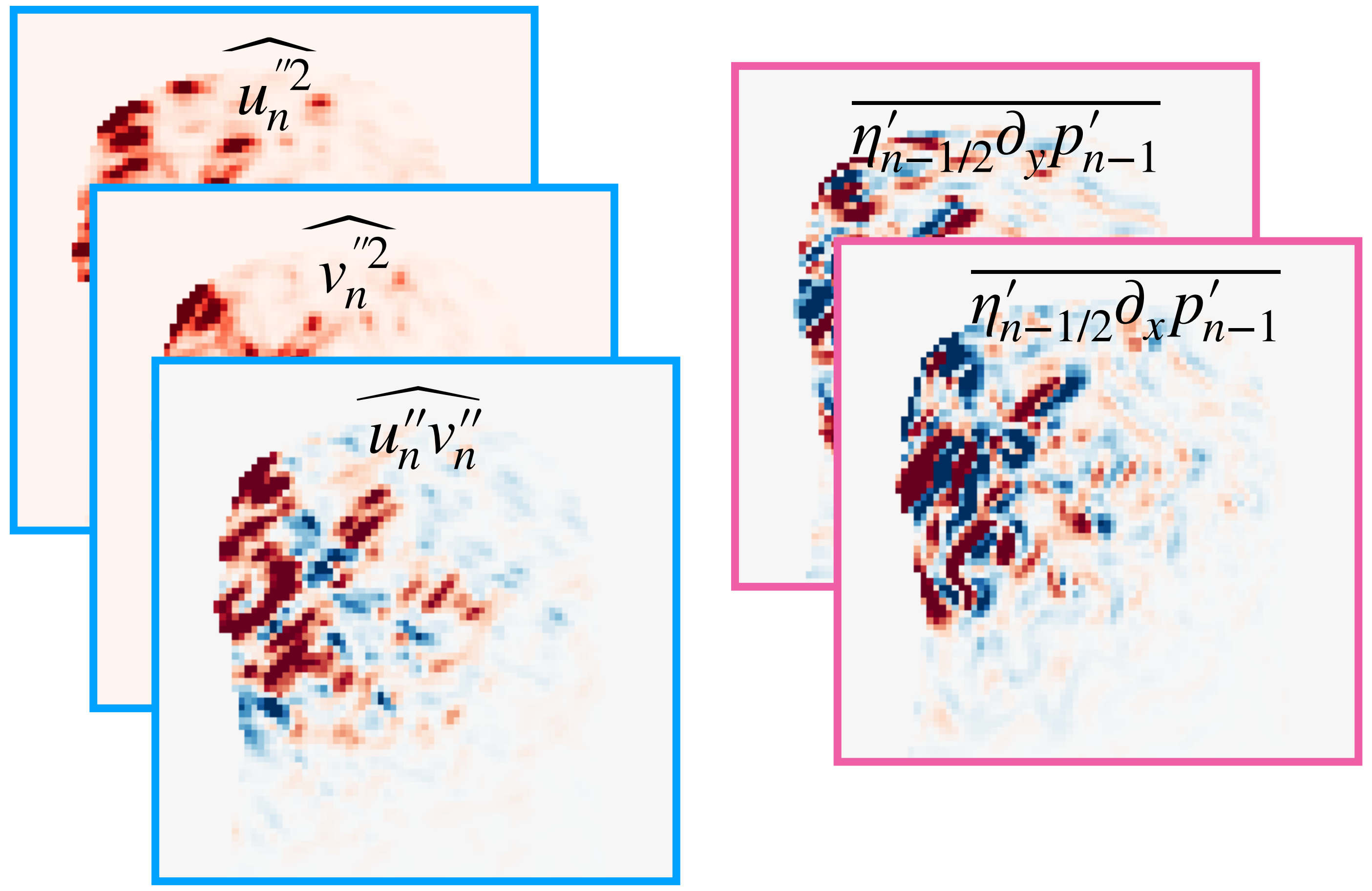


INPUTS:

- Coarse-grain derivatives of filtered flow variables



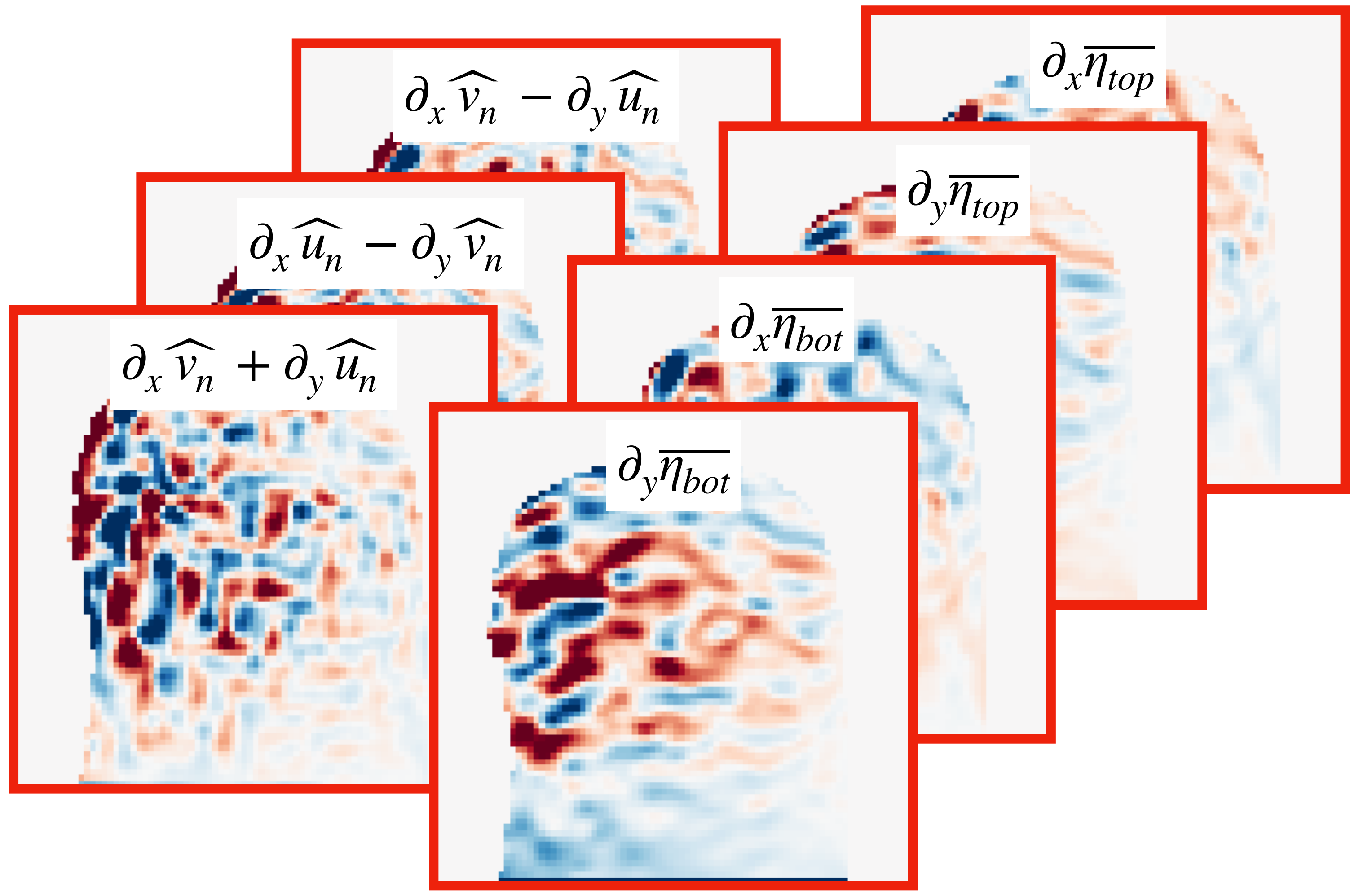
2 hidden layers
8 neurons each



Upscale energy transfer via KE backscatter

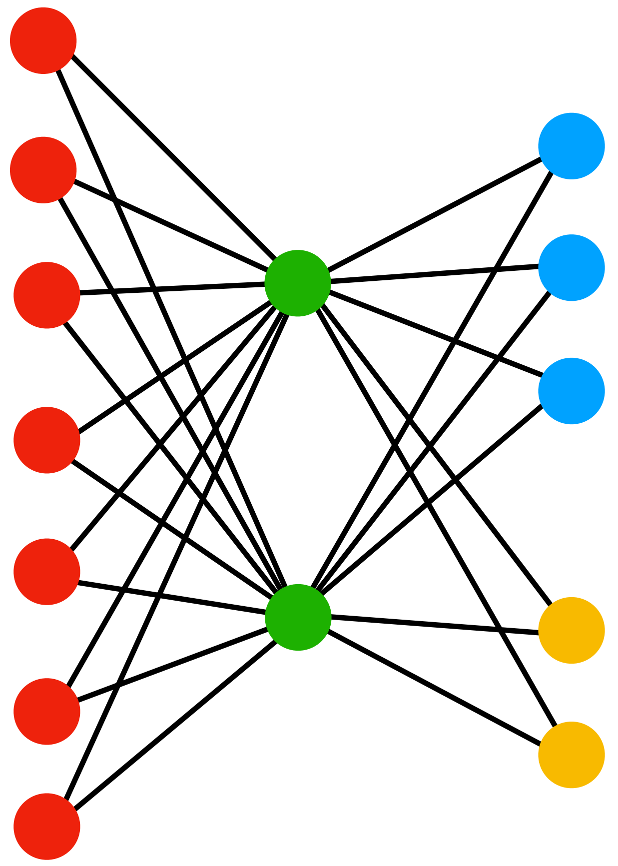
$$\widehat{\mathbf{E}}_n = \left(\widehat{E}_n^{(u)}, \widehat{E}_n^{(v)}\right) = \begin{pmatrix} \left(\widehat{u_n''^2} + \frac{1}{2\bar{h}_n} g^r_{n-1/2}\overline{\eta'^2_{n-1/2}}\right) & \left(\widehat{u_n''v_n''}\right) \\ \left(\widehat{u_n''v_n''}\right) & \left(\widehat{v_n''^2}+\frac{1}{2\bar{h}_n}g^r_{n-1/2}\overline{\eta'^2_{n-1/2}}\right) \\ \left(\overline{\eta'_{n-1/2}\frac{\partial}{\partial x}p'_{n-1}}\right) & \left(\overline{\eta'_{n-1/2}\frac{\partial}{\partial y}p'_{n-1}}\right)\end{pmatrix}$$

Parameterisation Development: 5) Artificial Neural Network

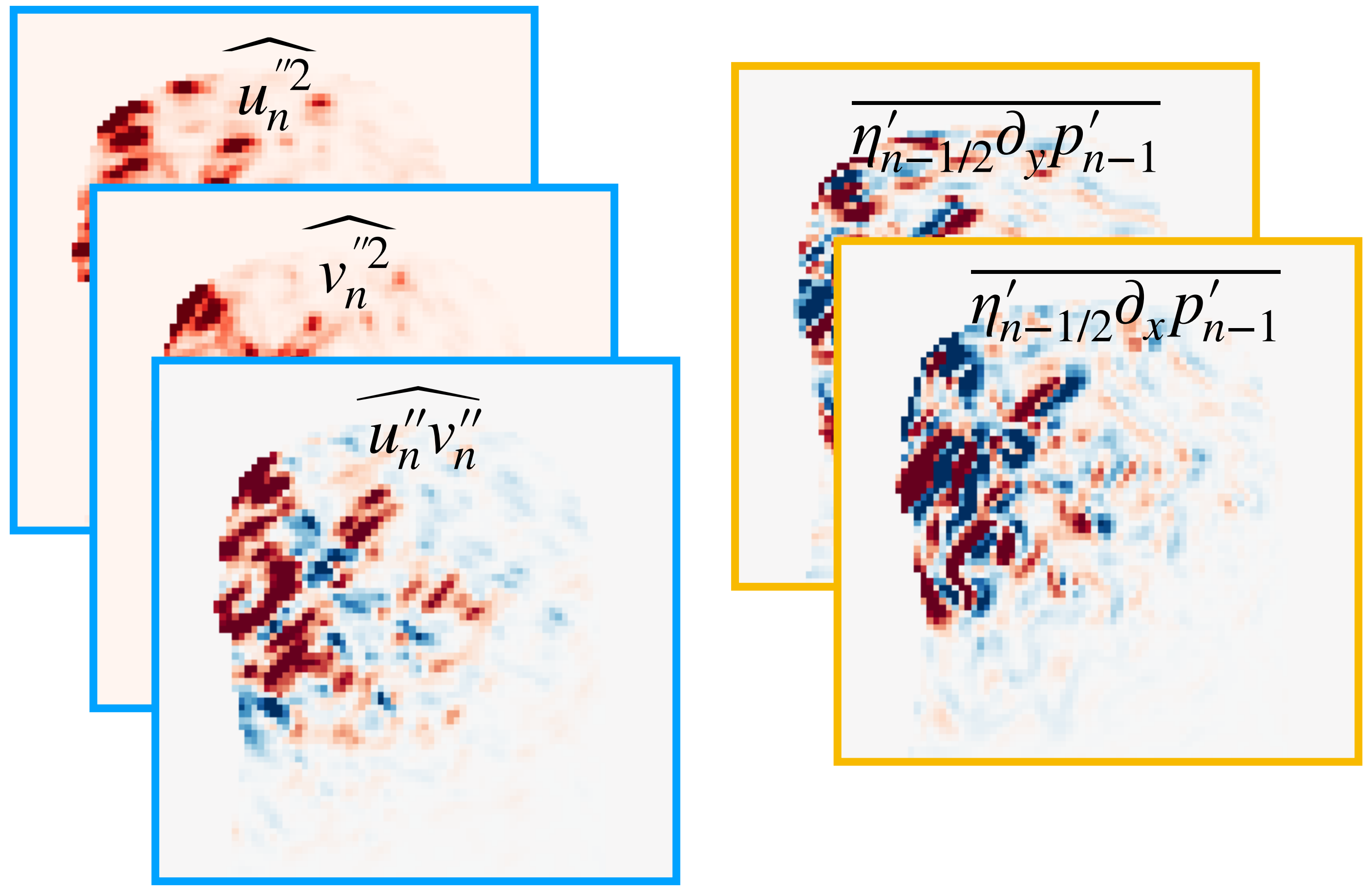


INPUTS:

- Coarse-grain derivatives of filtered flow variables



2 hidden layers
8 neurons each



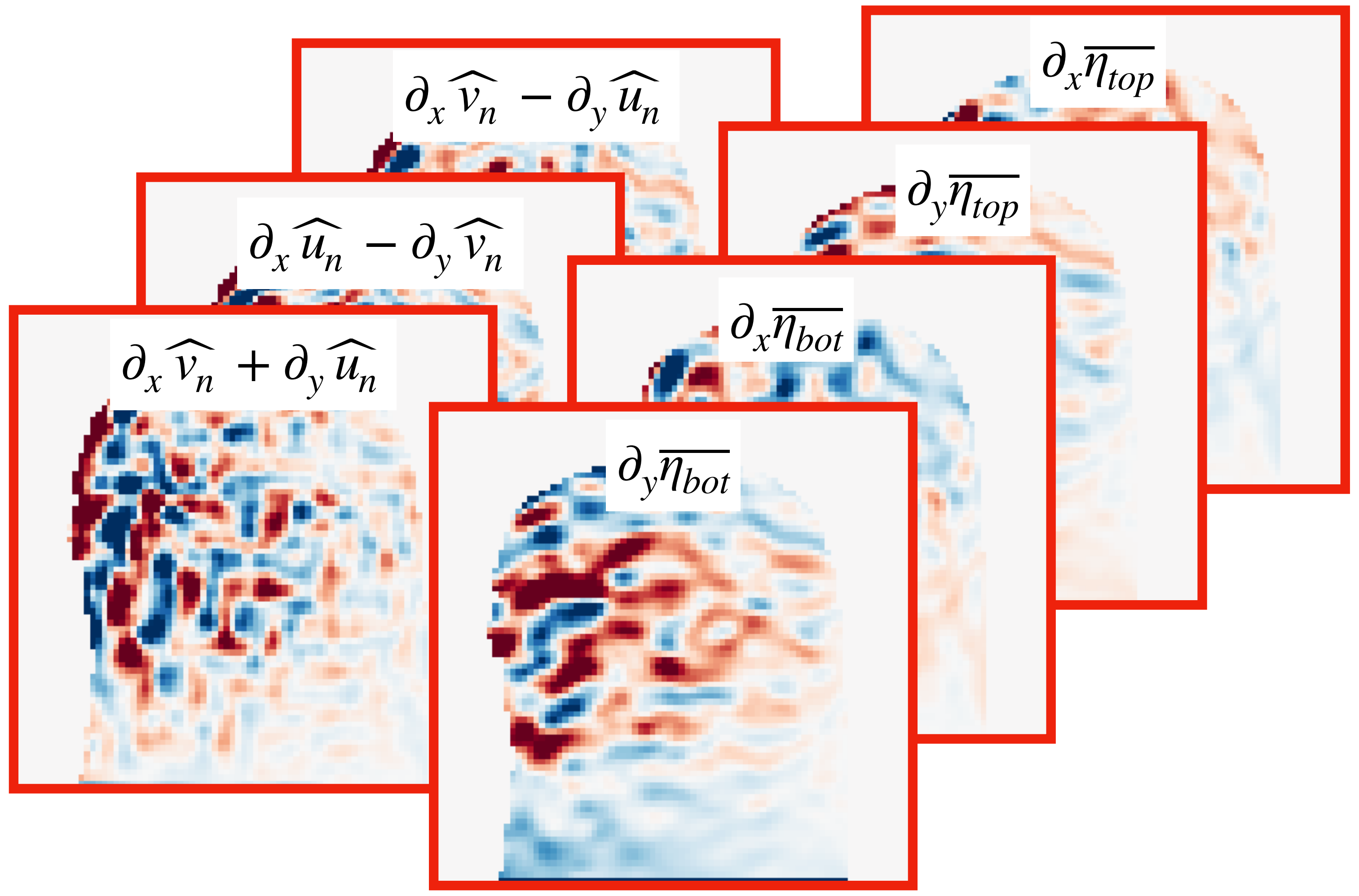
Upscale energy transfer via KE backscatter

$$\widehat{\mathbf{E}}_n = \left(\widehat{E}_n^{(u)}, \widehat{E}_n^{(v)}\right) = \begin{pmatrix} \left(\widehat{u_n''^2} + \frac{1}{2\bar{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}}\right) & \left(\widehat{u_n'' v_n''}\right) \\ \left(\widehat{u_n'' v_n''}\right) & \left(\widehat{v_n''^2} + \frac{1}{2\bar{h}_n} g^r_{n-1/2} \overline{\eta'^2_{n-1/2}}\right) \\ \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial x} p'_{n-1}}\right) & \left(\overline{\eta'_{n-1/2} \frac{\partial}{\partial y} p'_{n-1}}\right) \end{pmatrix}$$

Neglecting sub-filter potential energy

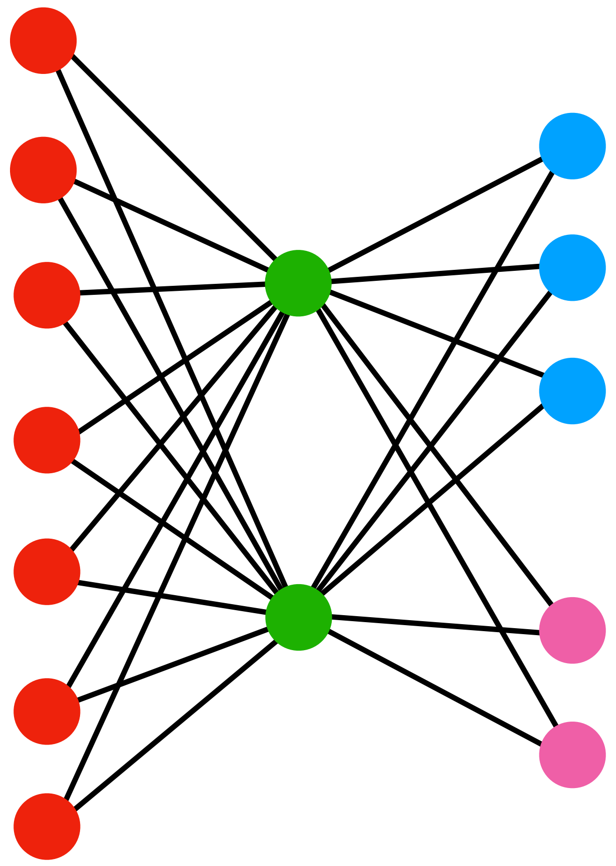
e.g., Marshall et al (2012)

Parameterisation Development: 5) Artificial Neural Network

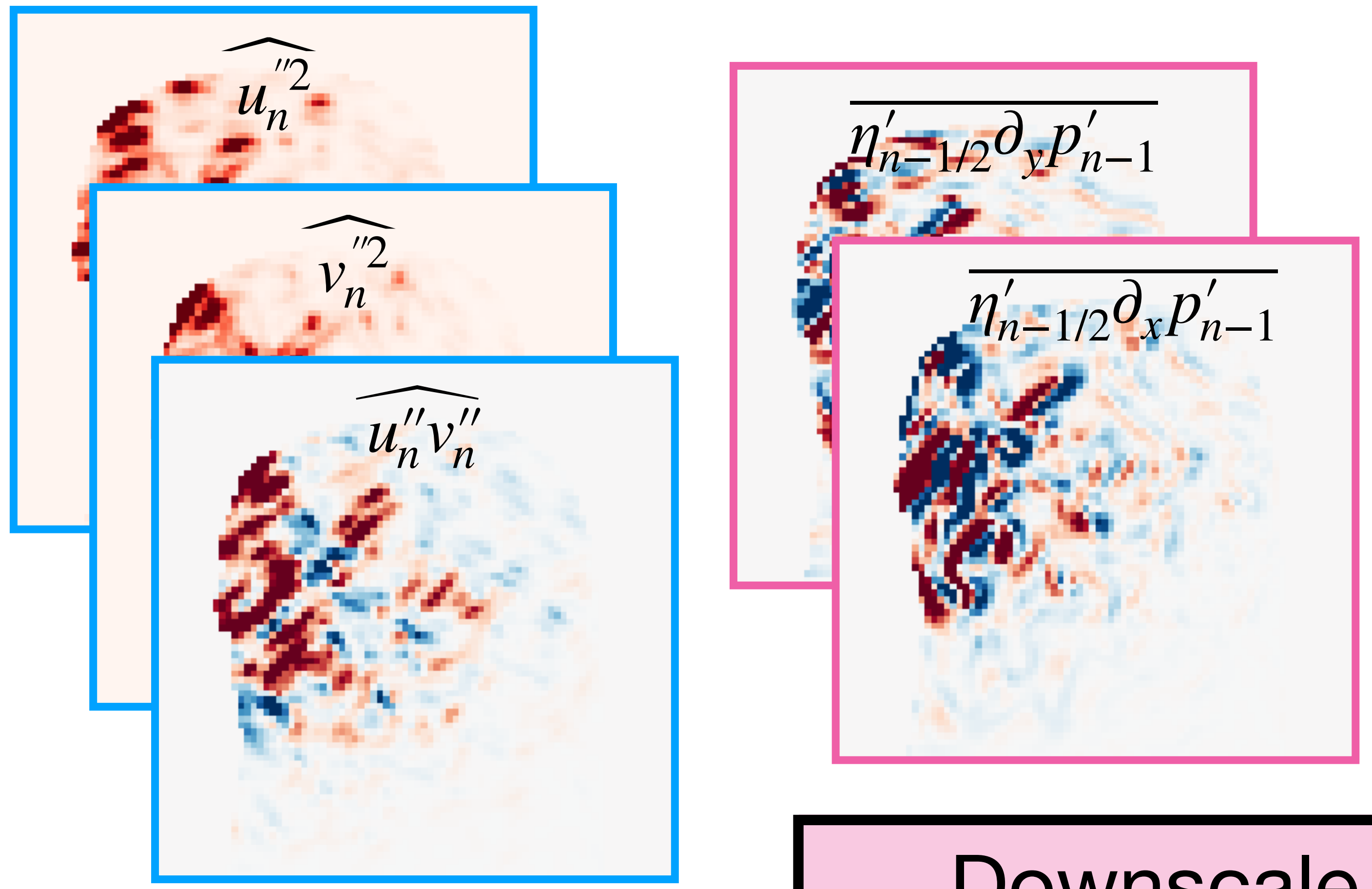


INPUTS:

- Coarse-grain derivatives of filtered flow variables



2 hidden layers
8 neurons each

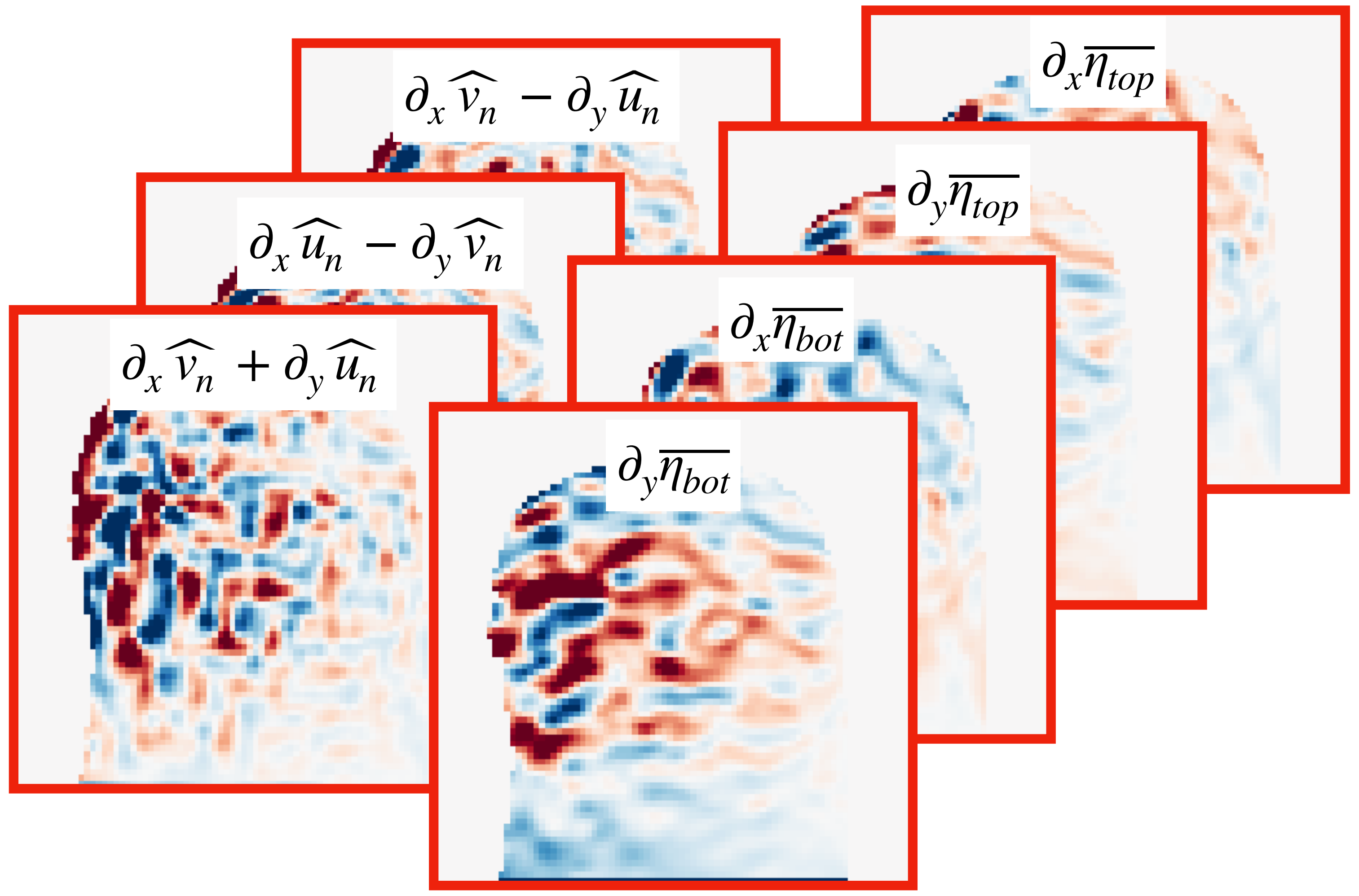


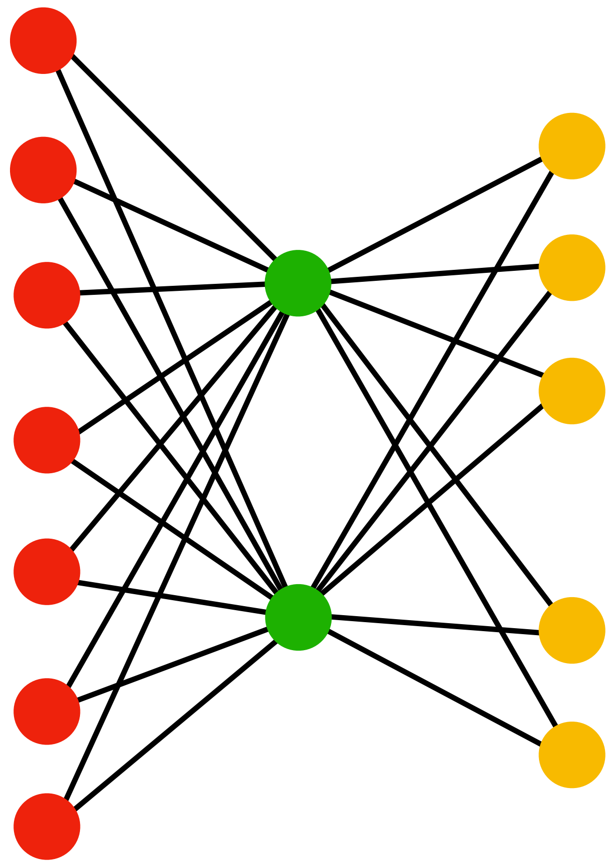
Upscale energy transfer via KE backscatter

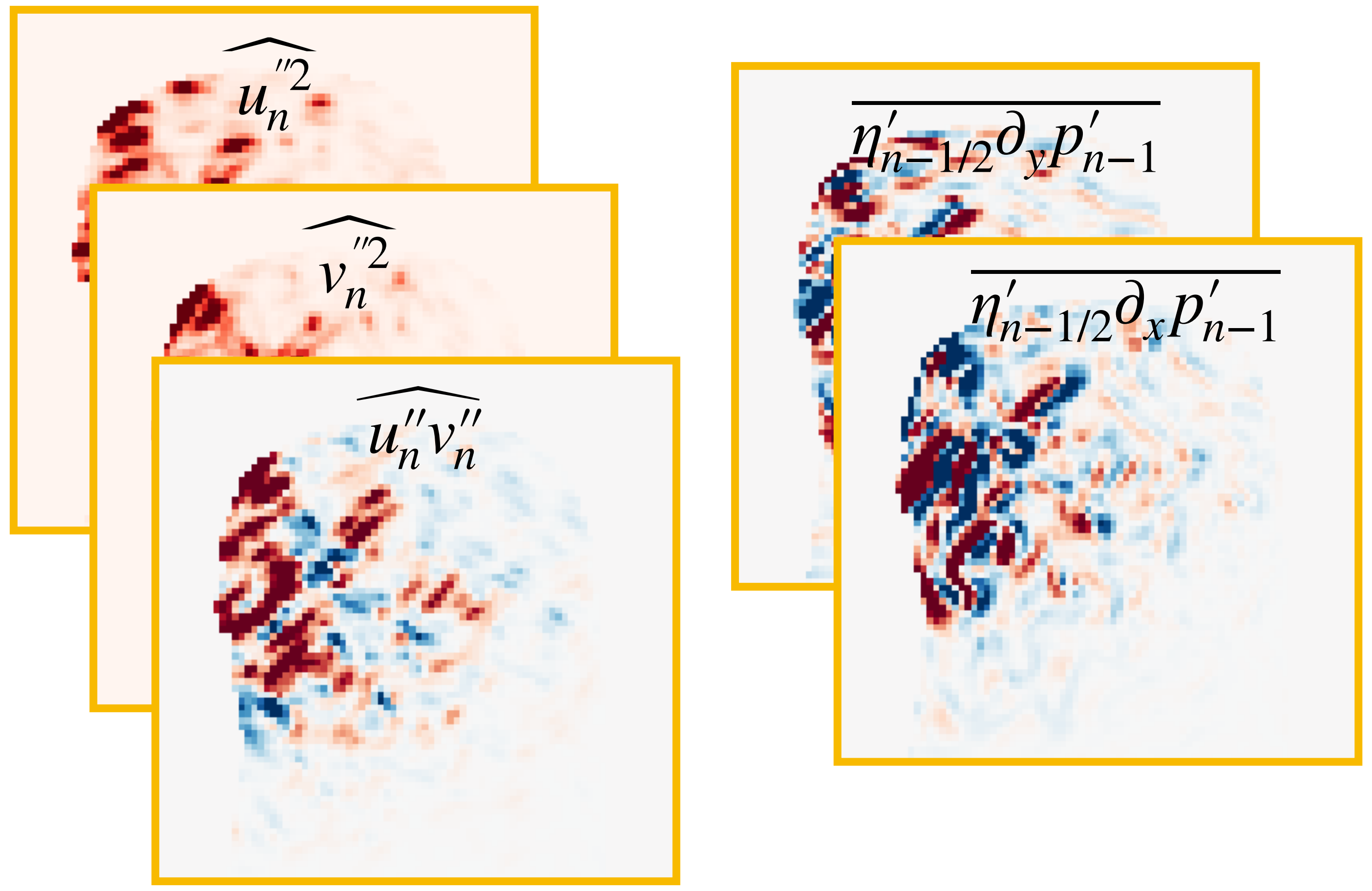
Downscale energy transfer via APE extraction

$$\widehat{\mathbf{E}}_n = \left(\widehat{E}_n^{(u)}, \widehat{E}_n^{(v)}\right) = \begin{pmatrix} \left(\widehat{u_n''^2}\right) & \left(\widehat{u_n''v_n''}\right) \\ \left(\widehat{u_n''v_n''}\right) & \left(\widehat{v_n''^2}\right) \\ \left(\overline{\eta'_{n-1/2}\frac{\partial}{\partial x}p'_{n-1}}\right) & \left(\overline{\eta'_{n-1/2}\frac{\partial}{\partial y}p'_{n-1}}\right) \end{pmatrix}$$

Parameterisation Development: 5) Artificial Neural Network







Momentum flux components

$$\Delta x^2 ||\nabla u||^2 \,\textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{mom}$$

Normalized inputs

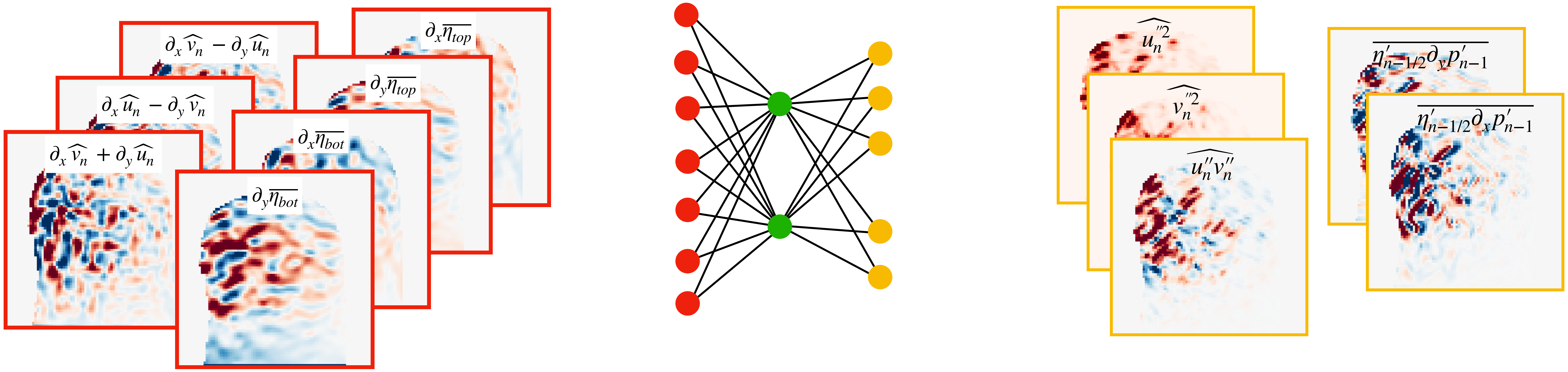
Thickness flux components

$$\Delta x^2 f ||\nabla u|| \, ||S|| \,\textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{thick}$$

$$||\nabla u|| = \left(\left(\partial_x \hat{v} - \partial_y \hat{u}\right)^2 + \left(\partial_x \hat{u} - \partial_y \hat{v}\right)^2 + \left(\partial_x \hat{v} + \partial_y \hat{u}\right)^2\right)^{1/2}$$

$$||S|| = \left(\left(\partial_x \overline{\eta_{top}}\right)^2 + \left(\partial_y \overline{\eta_{top}}\right)^2 + \left(\partial_x \overline{\eta_{bot}}\right)^2 + \left(\partial_y \overline{\eta_{bot}}\right)^2\right)^{1/2}$$

Parameterisation Development: 5) Artificial Neural Network



Momentum flux components

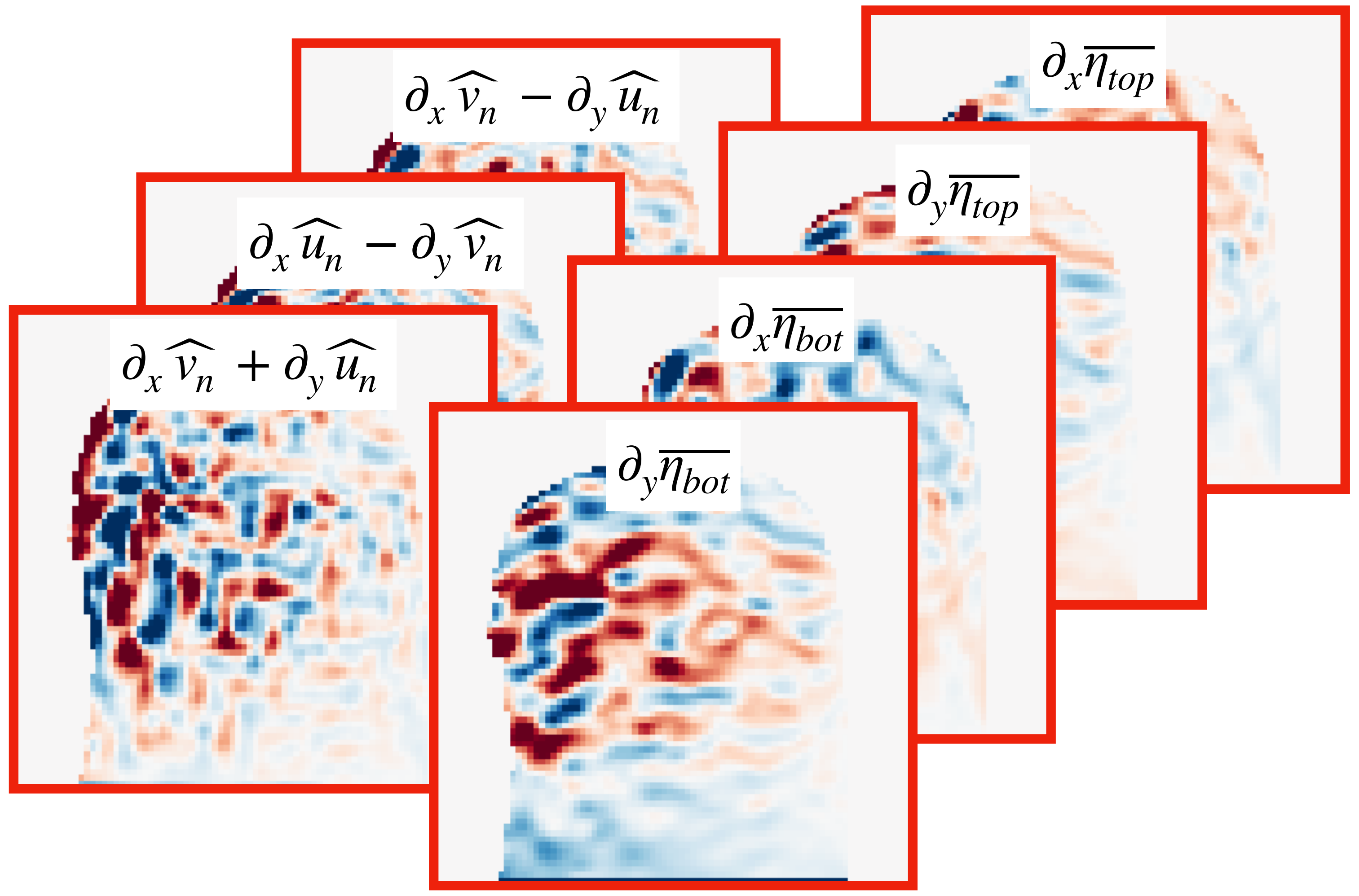
$$\Delta x^2 ||\nabla u||^2 \, \textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{mom}$$

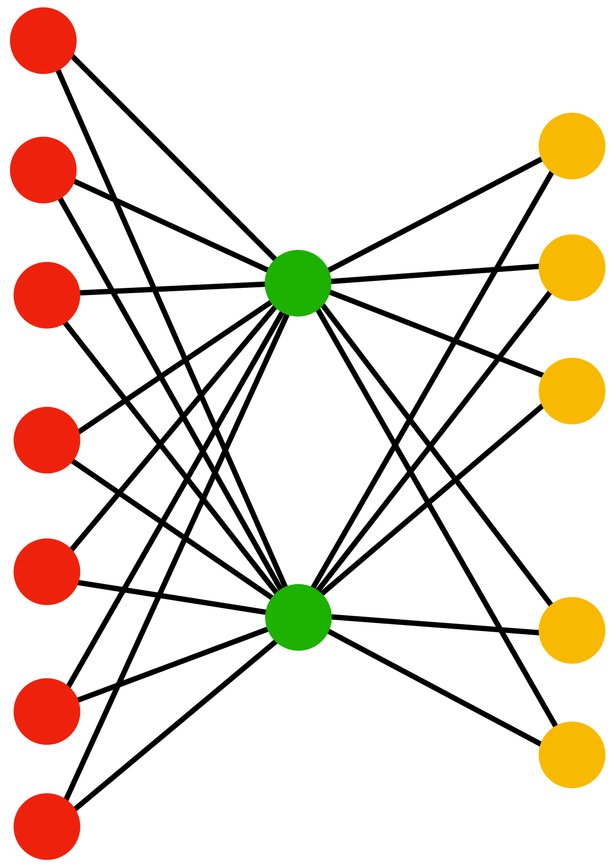
Normalized inputs → **ANN** → **Non-dimensional outputs**

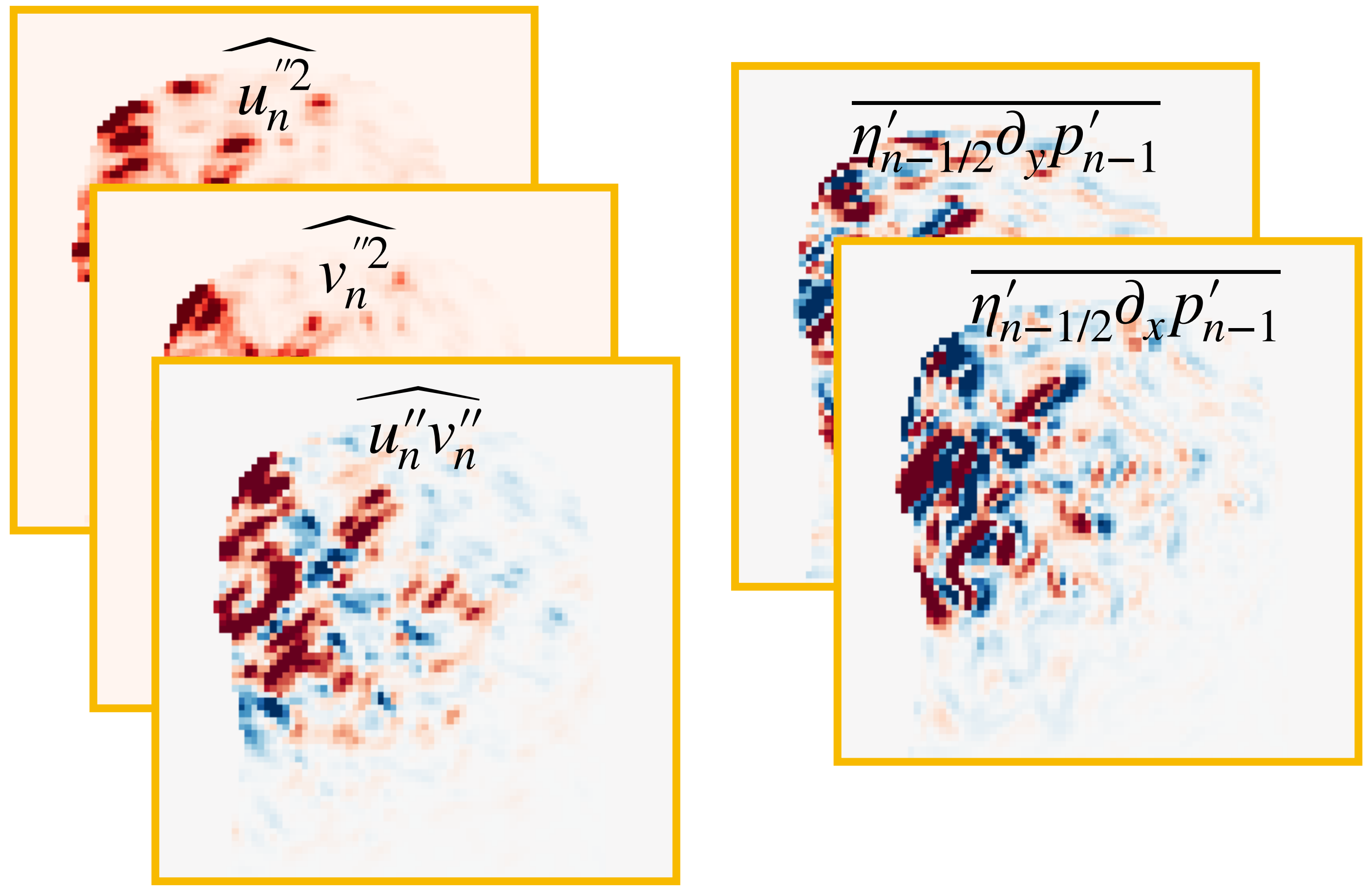
Thickness flux components

$$\Delta x^2 f ||\nabla u|| \, ||S|| \, \textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{thick}$$

Parameterisation Development: 5) Artificial Neural Network







Momentum flux components

$$\Delta x^2 ||\nabla u||^2 \, \text{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{mom}$$

Thickness flux components

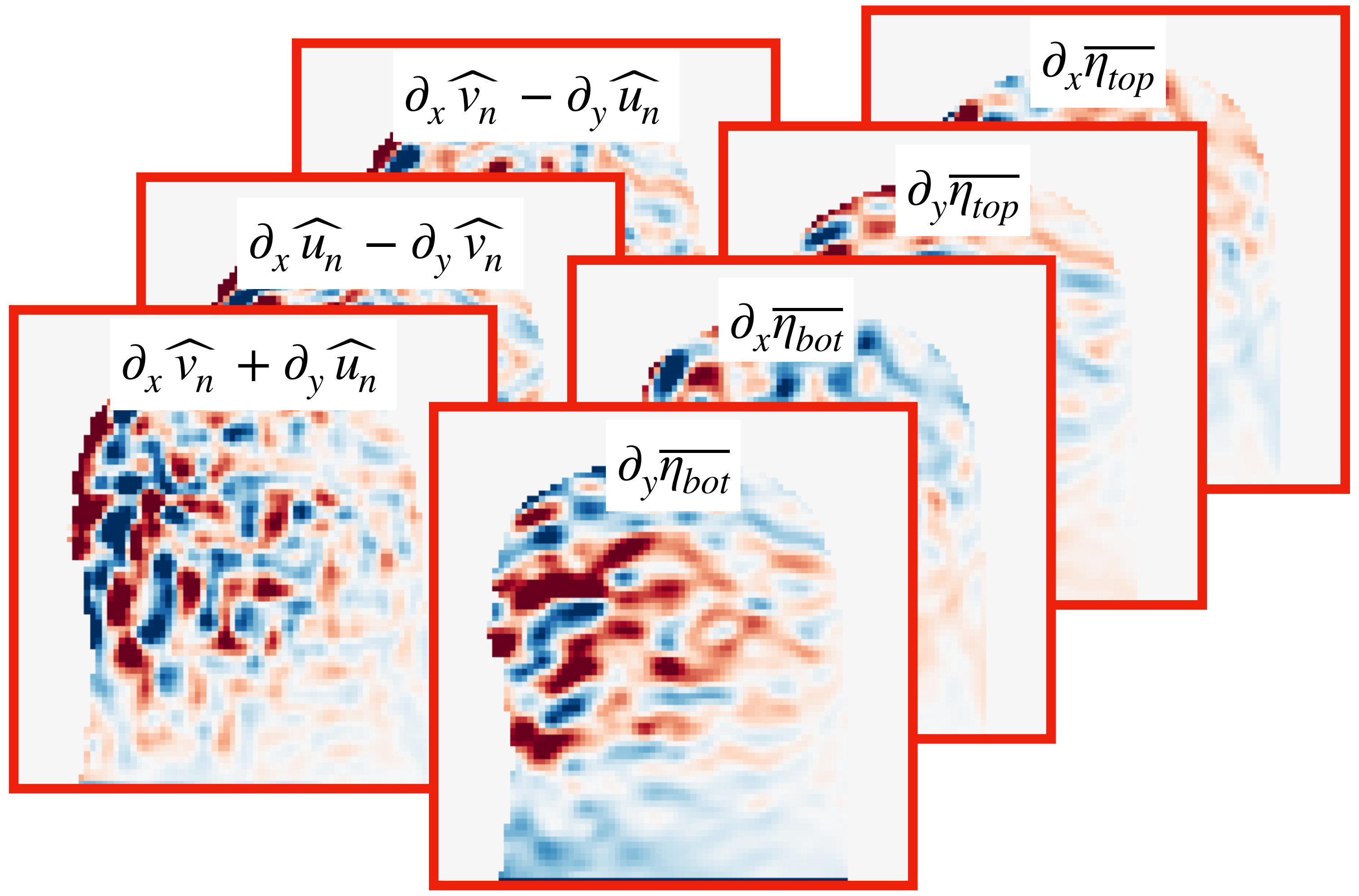
$$\Delta x^2 f ||\nabla u|| \, ||S|| \, \text{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{thick}$$

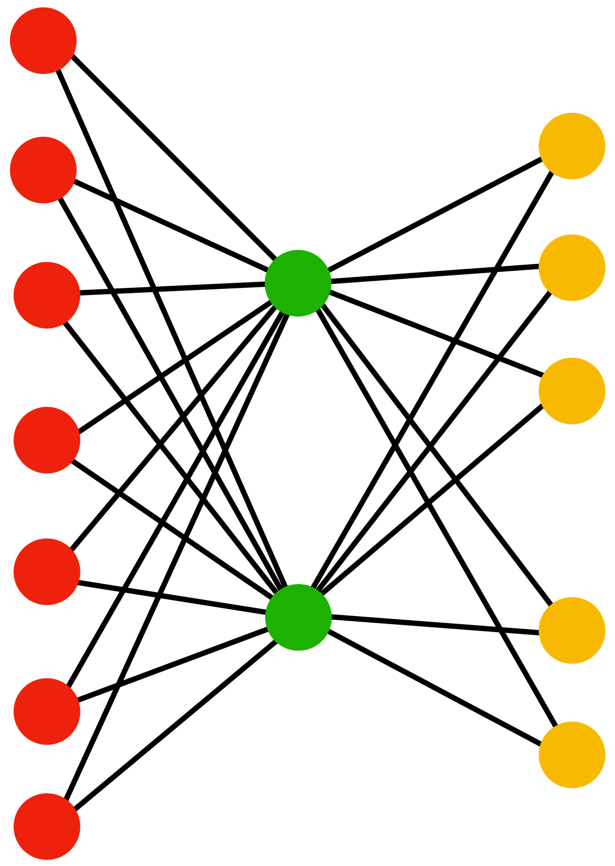
$$\Delta^2 = 2\left(\delta x \delta y\right)^2 / \left((\delta x)^2 + (\delta y)^2\right)$$

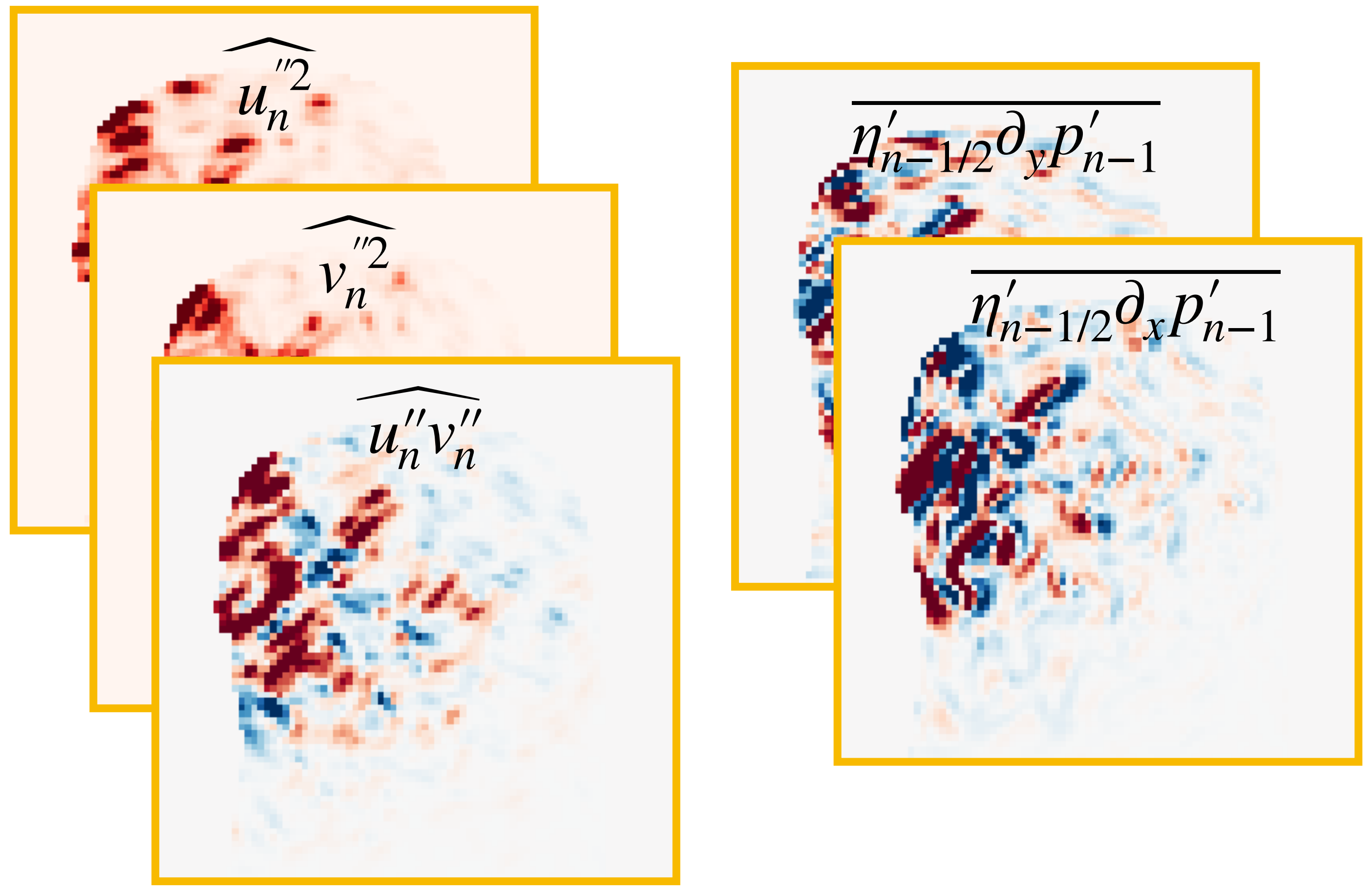
Normalized inputs → **ANN** → **Non-dimensional outputs** → **Momentum flux scaling** / **Thickness flux scaling**

$$f = 2\Omega \sin\left(\frac{\pi Latitude}{180^\circ}\right)$$

Parameterisation Development: 5) Artificial Neural Network







Based off of eddy energy system

Momentum flux components

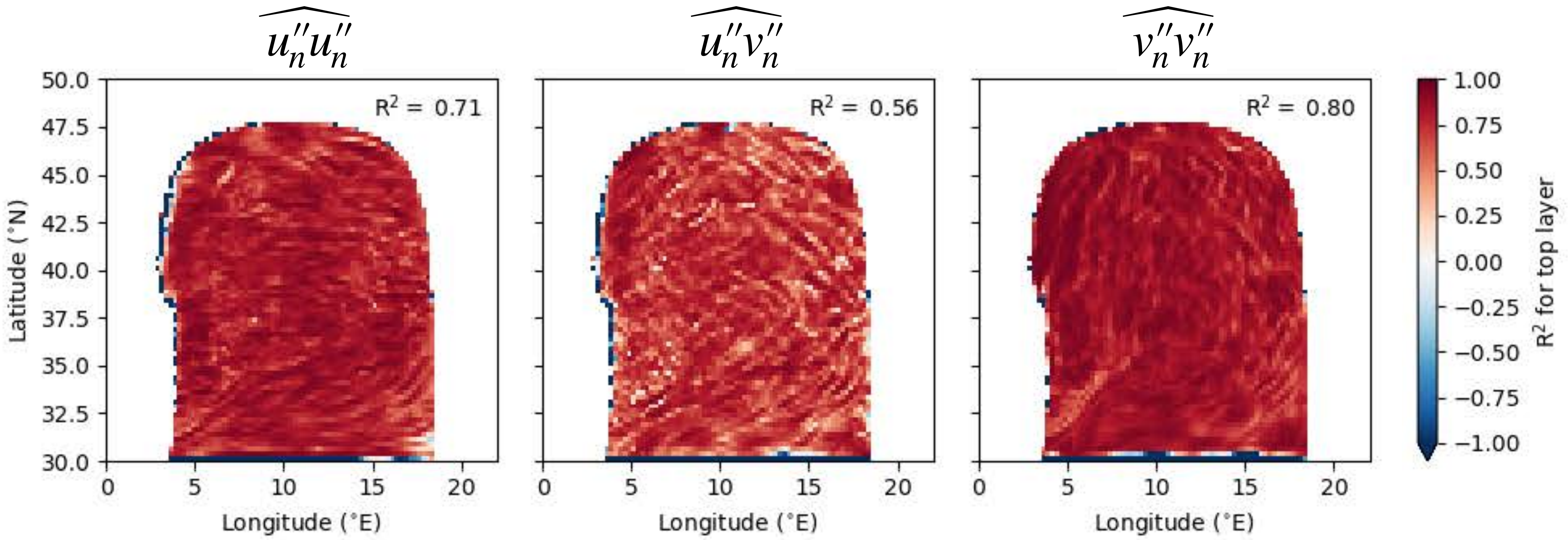
$$\Delta x^2 ||\nabla u||^2 \, \textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{mom}$$

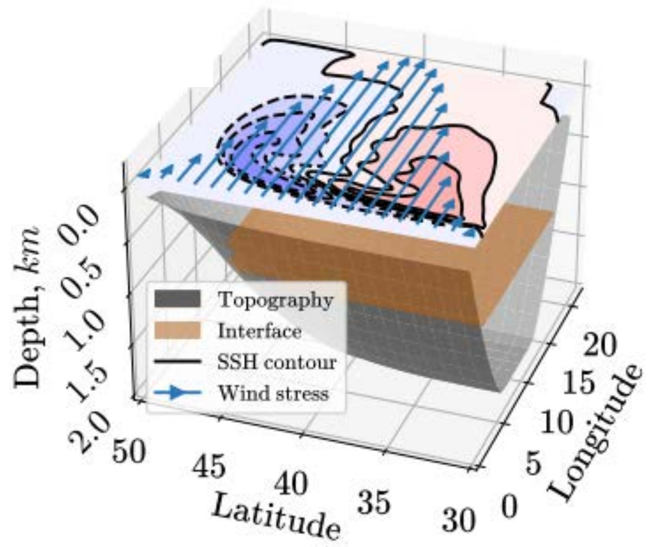
Thickness flux components

$$\Delta x^2 f ||\nabla u|| \, ||S|| \, \textbf{ANN}\left(\frac{\nabla u}{||\nabla u||}, \frac{S}{||S||}\right) \approx \widehat{E}_{thick}$$

Normalized inputs → **ANN** → **Non-dimensional outputs** → **Momentum flux scaling** / **Thickness flux scaling** → **EPF tensor**

Offline performance on test data

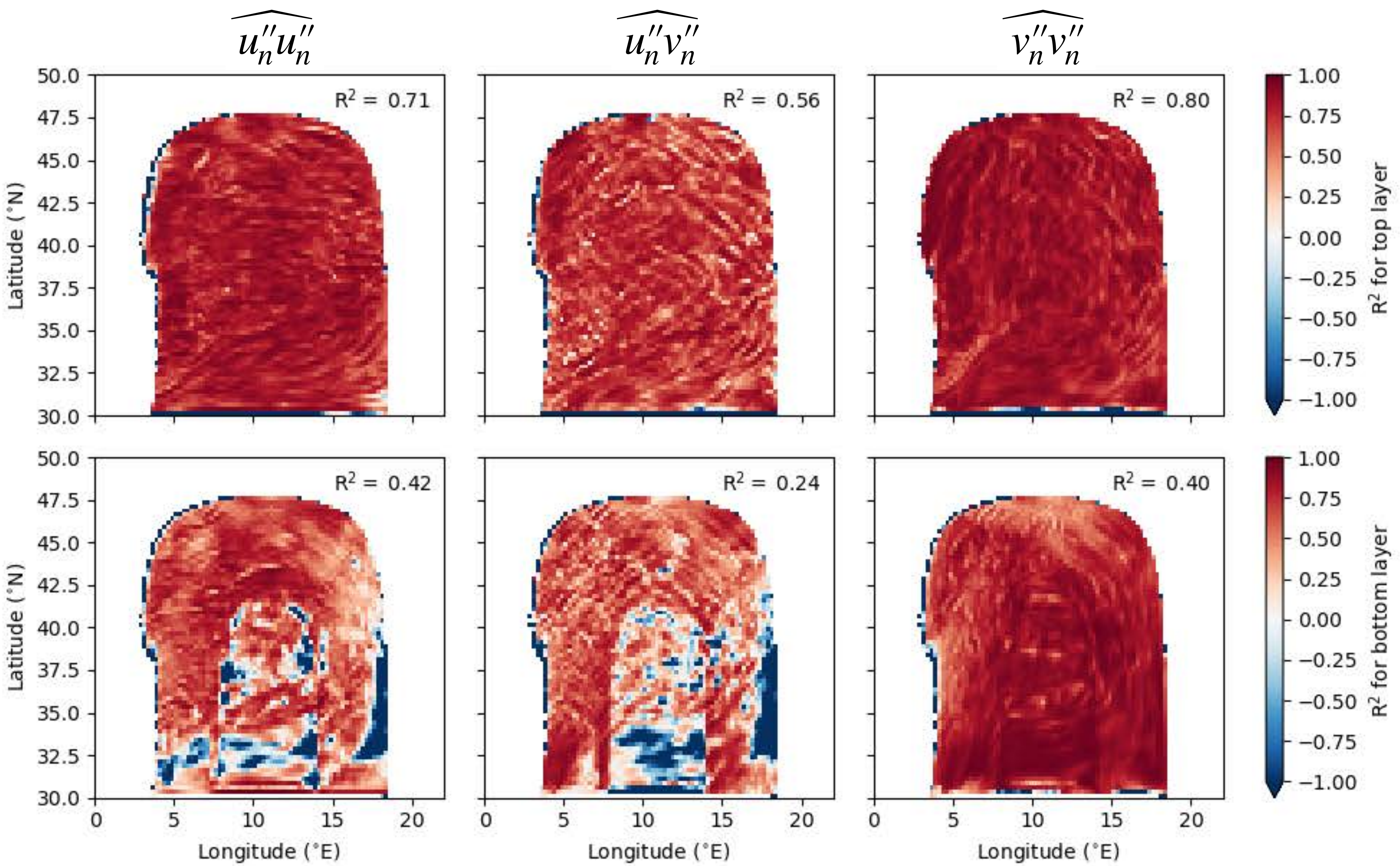


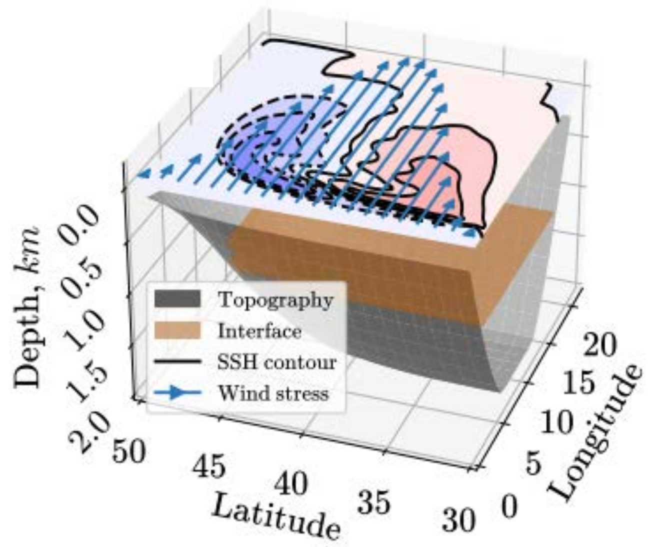


High R^2 for Reynolds' stresses in top layer

*The R^2 is set to 0.0 when the spatial average $R^2 < 0$

Offline performance on test data





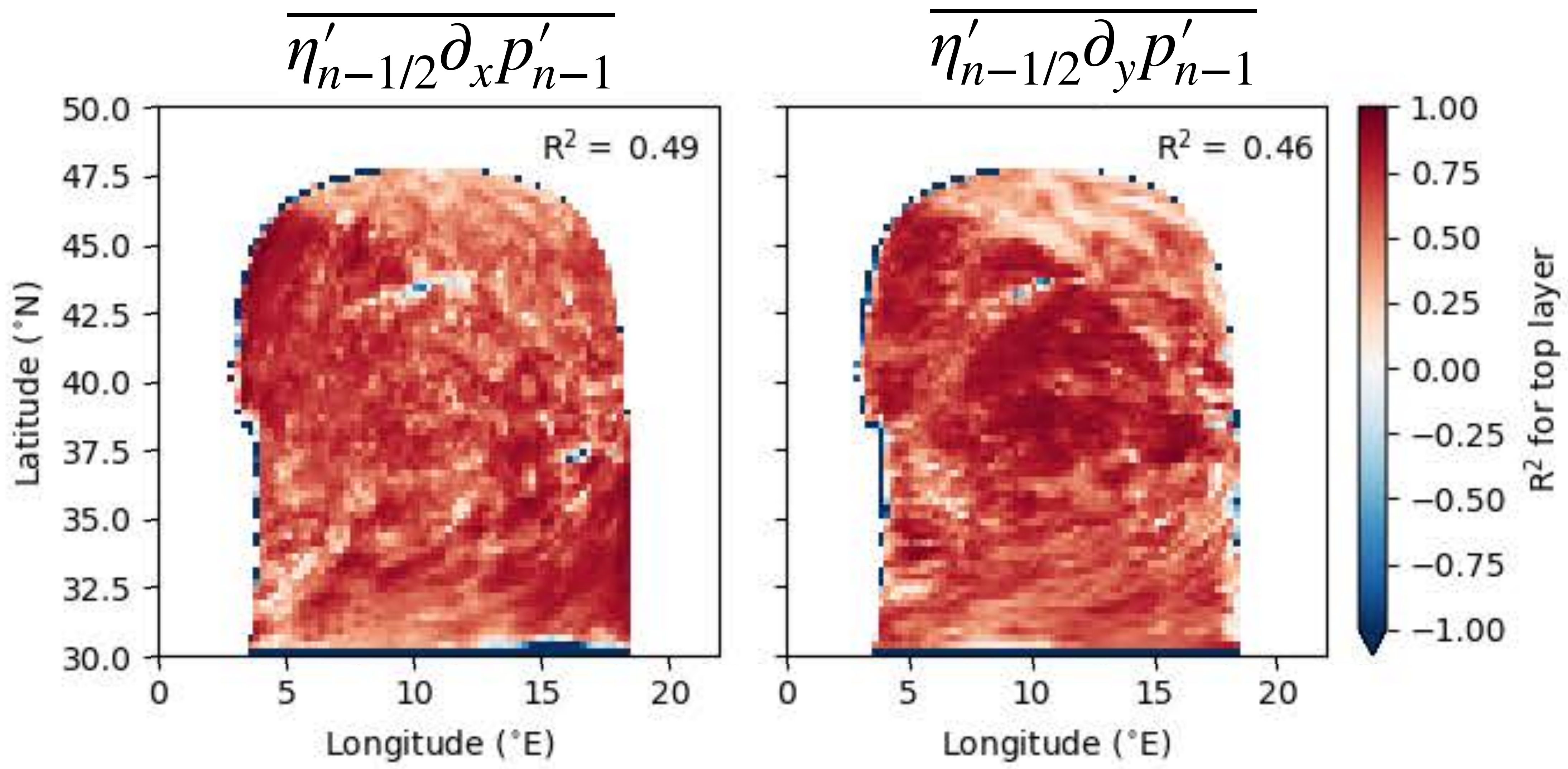
High R^2 for Reynolds' stresses in top layer

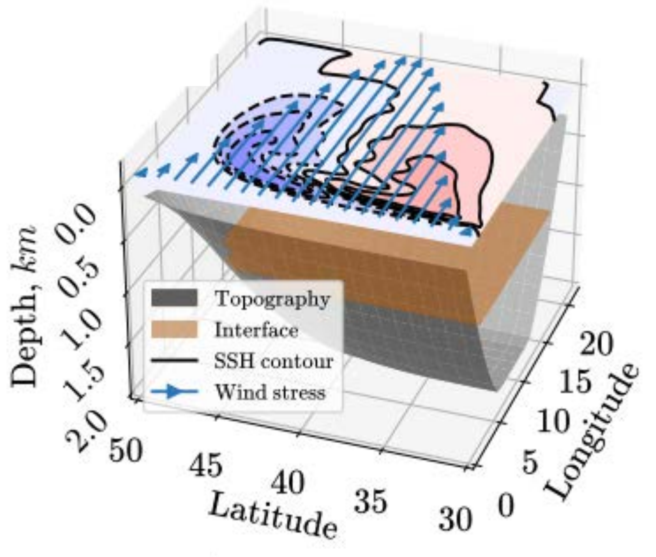
Worse performance in bottom layer

*The R^2 is set to 0.0 when the spatial average $R^2 < 0$

Offline performance on test data

Moderate R^2 for dual form stresses in top layer



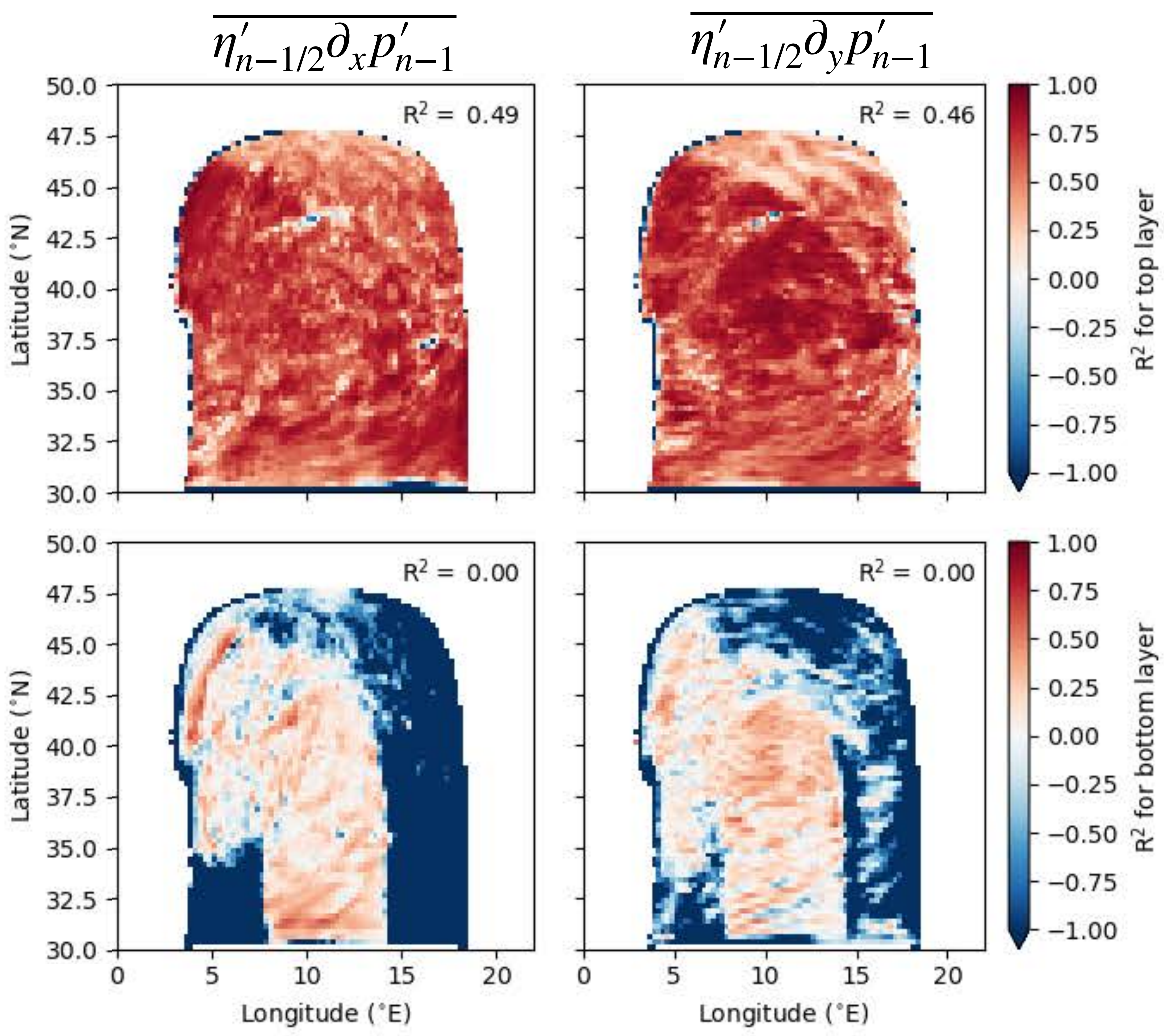


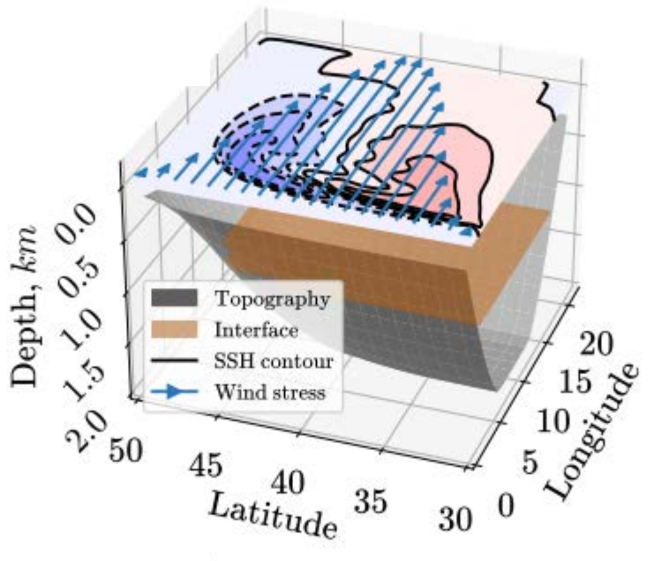
*The R^2 is set to 0.0 when the spatial average $R^2 < 0$

Offline performance on test data

Moderate R^2 for dual form stresses in top layer

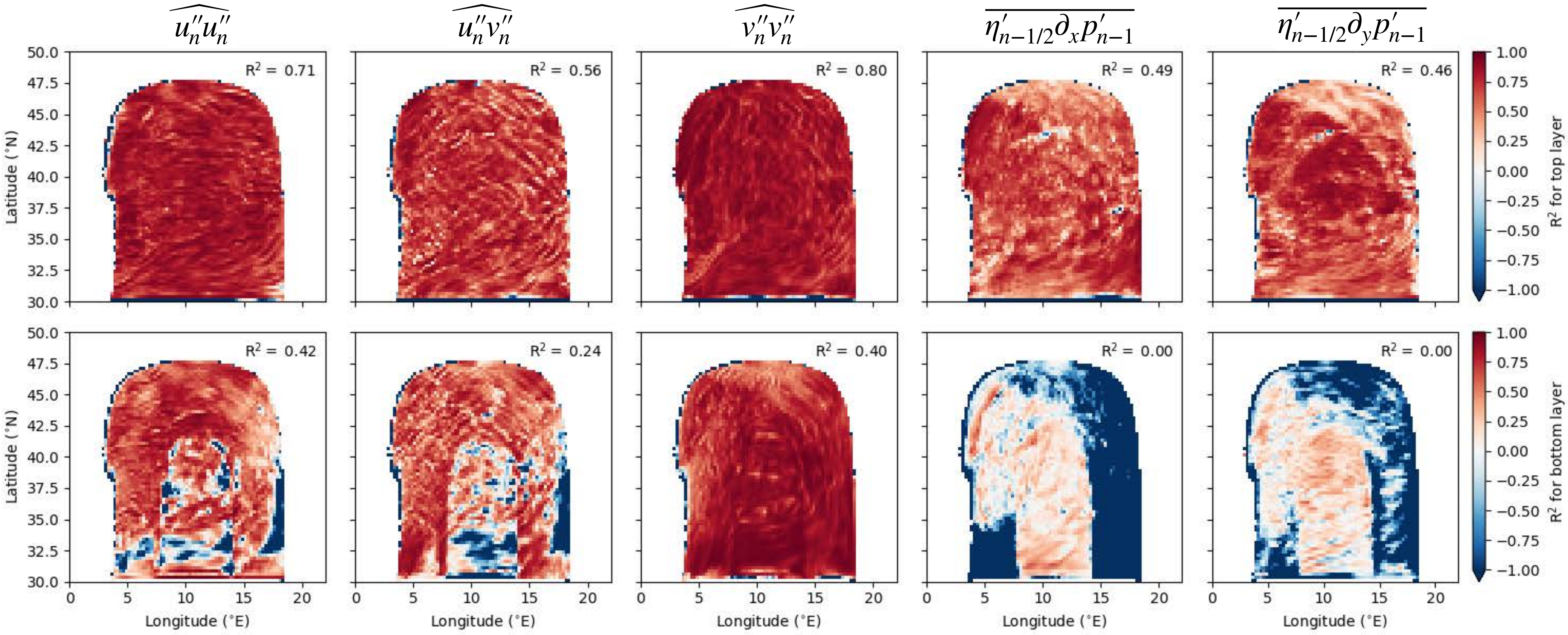
Poor R^2 for dual form stresses in bottom layer





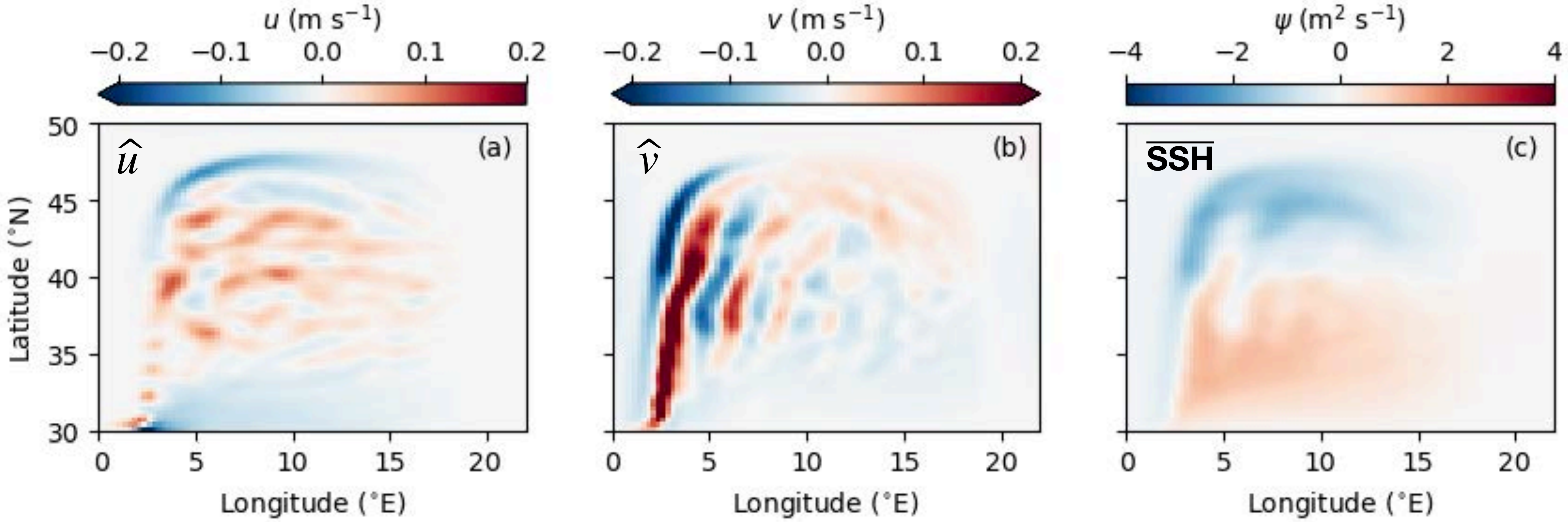
*The R^2 is set to 0.0 when the spatial average $R^2 < 0$

Offline performance on test data



*The R^2 is set to 0.0 when the spatial average $R^2 < 0$

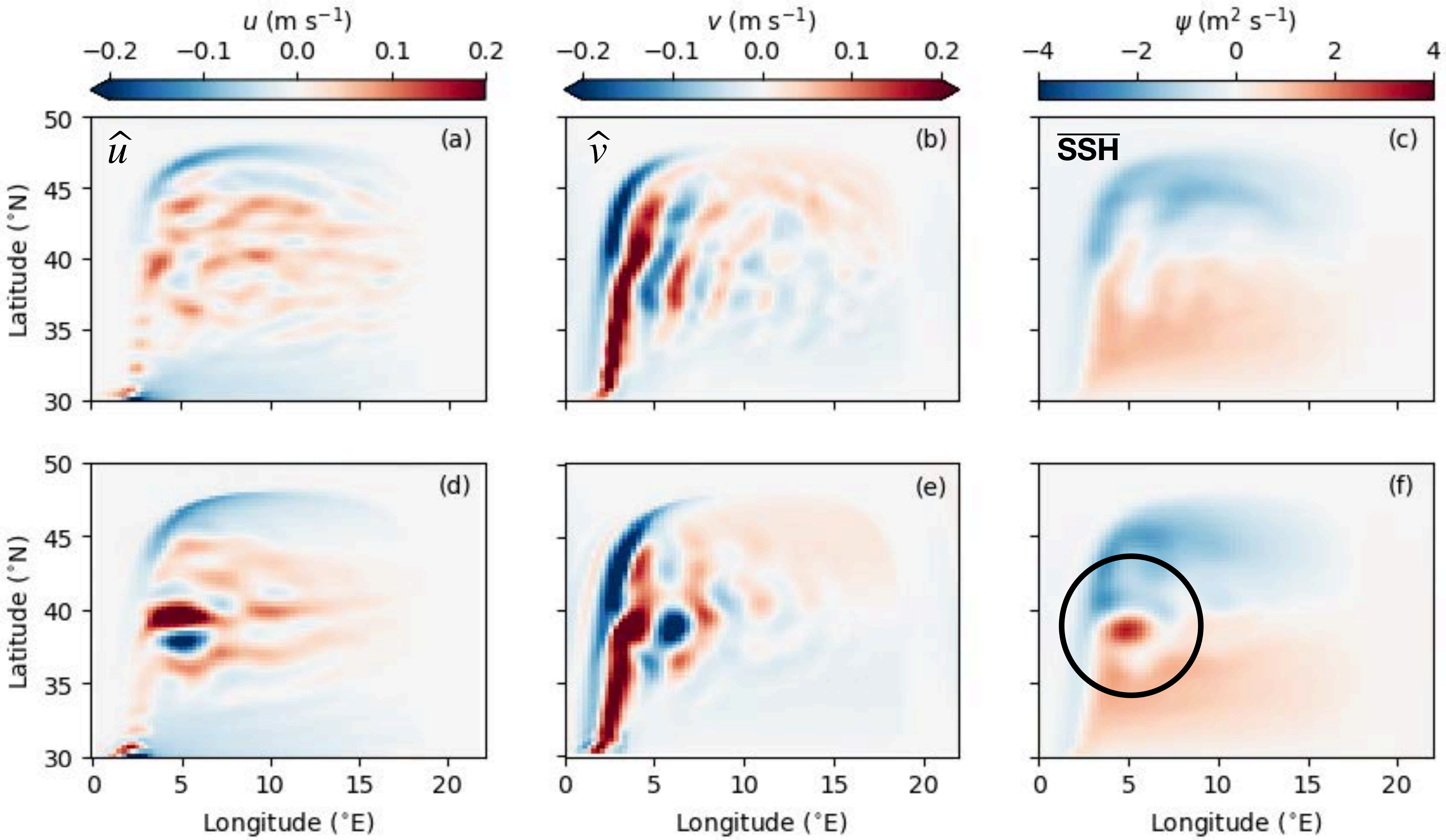
Online Implementation: Top Layer Mean Flow



$1/32^\circ$ flow field filtered and coarsened to $1/4^\circ$ resolution.

Averages over daily snapshots of ~ 10 simulation years

Online Implementation: Top Layer Mean Flow



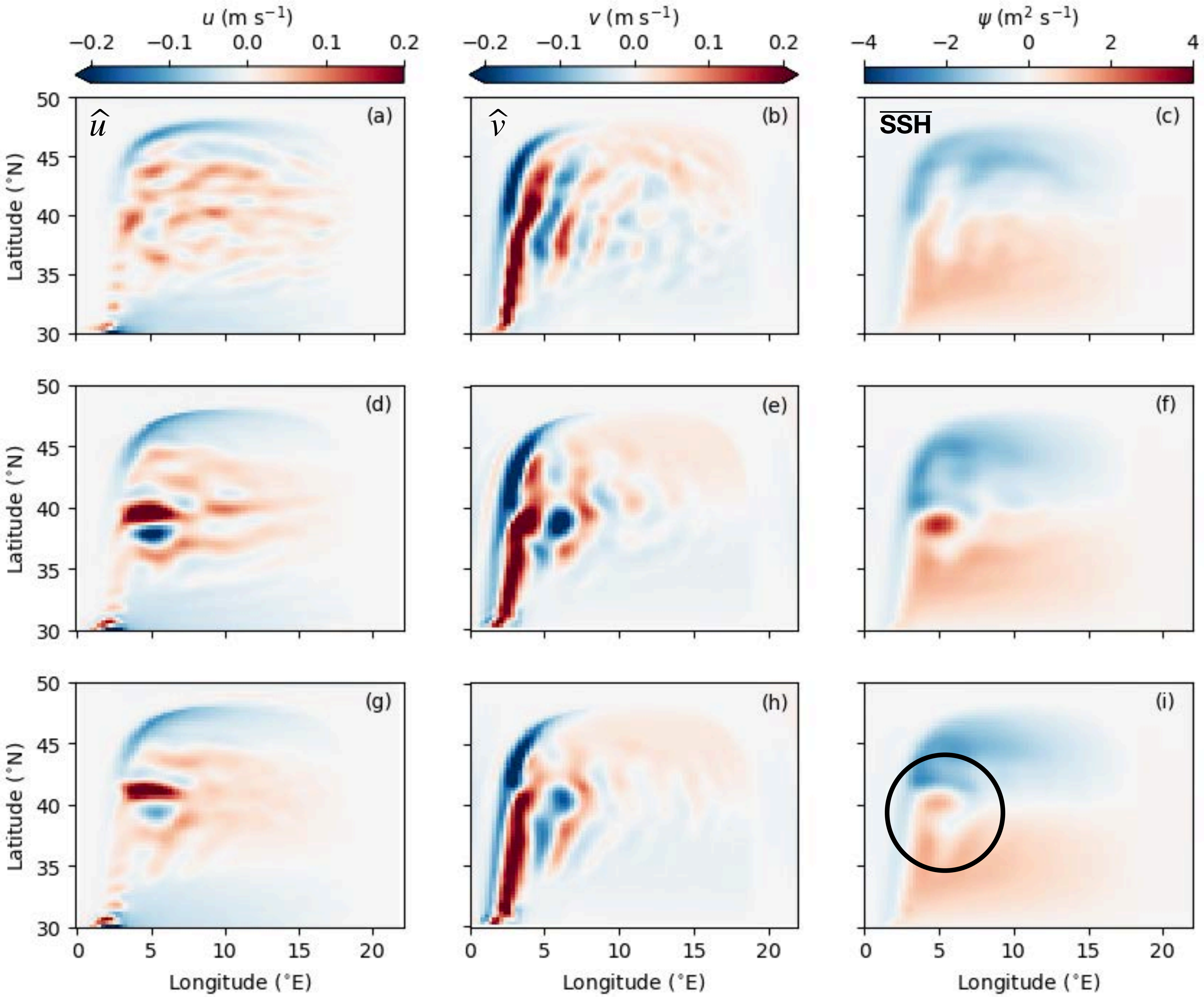
$1/32^\circ$ flow field filtered and coarsened to $1/4^\circ$ resolution.

$1/4^\circ$ resolution run with **no parameterisation** (only Smagorinsky)

Pesky persistent eddy

Averages over daily snapshots of ~ 10 simulation years

Online Implementation: Top Layer Mean Flow



$1/32°$ flow field filtered and coarsened to $1/4°$ resolution.

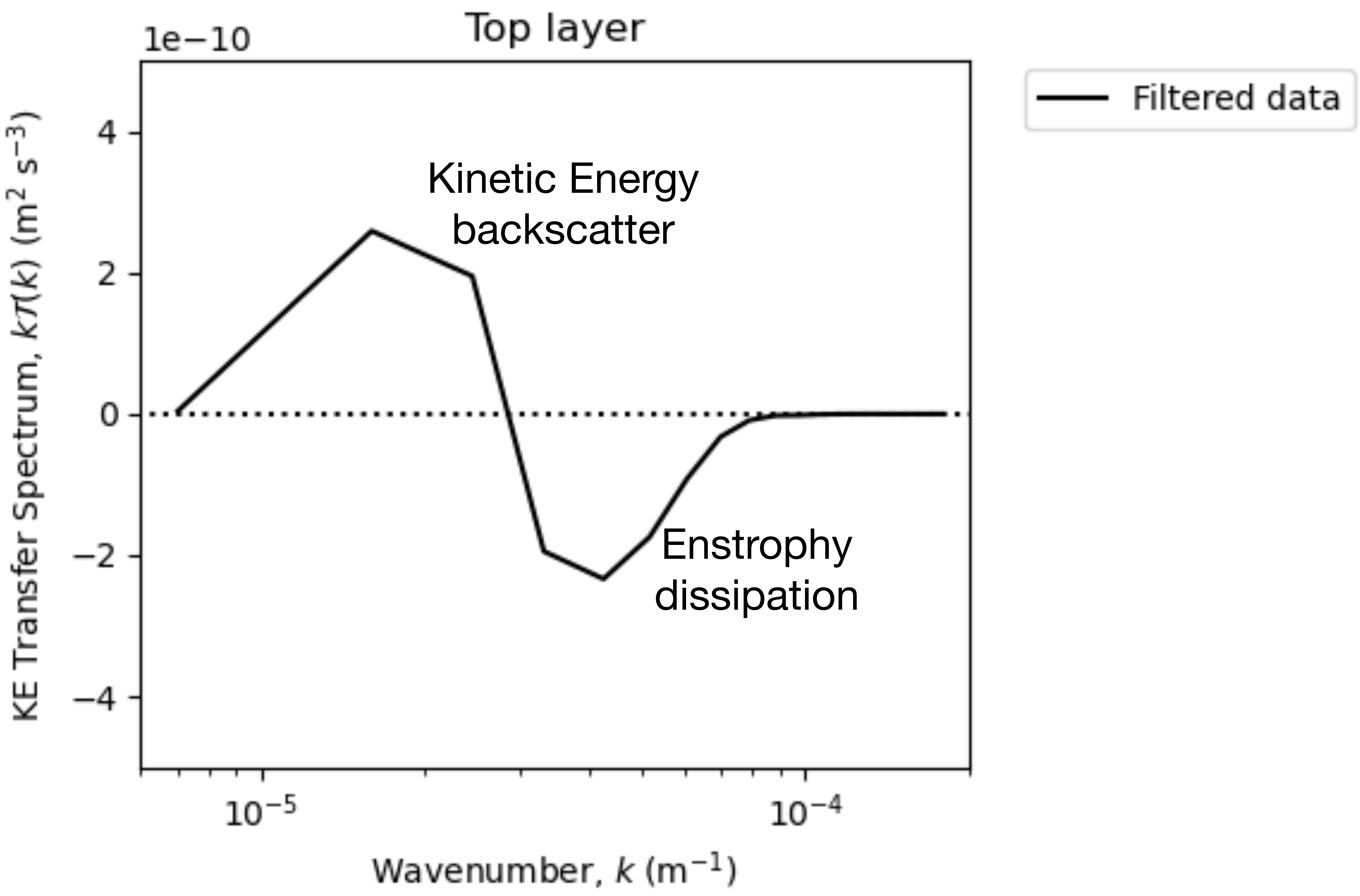
$1/4°$ resolution run with **no parameterisation** (only Smagorinsky)

$1/4°$ resolution run with **EPF ANN parameterisation** (in addition to Smagorinsky)

Pesky persistent eddy is **weakened by parameterisation!**

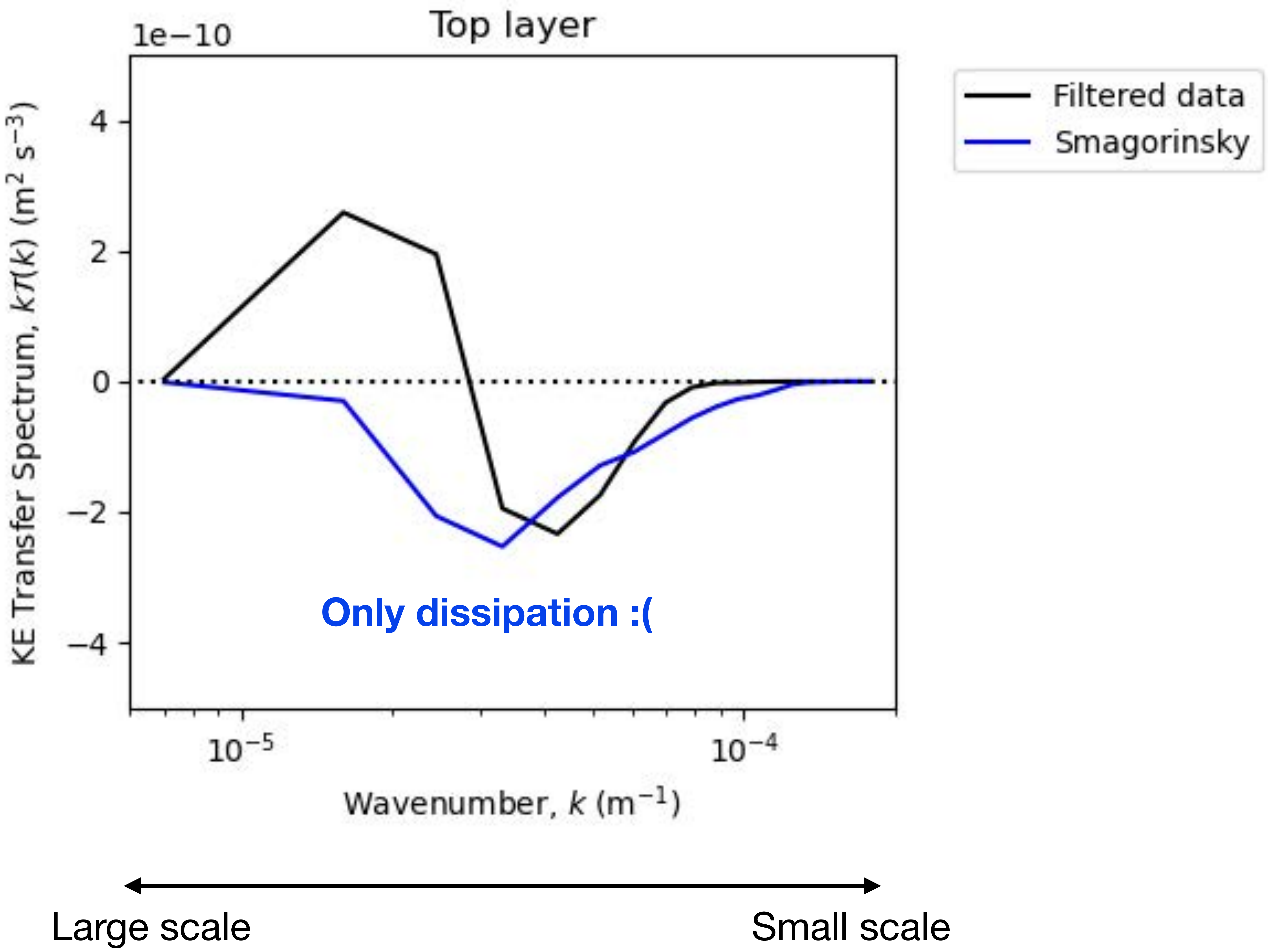
Averages over daily snapshots of ~ 10 simulation years

Isotropic KE transfer spectra

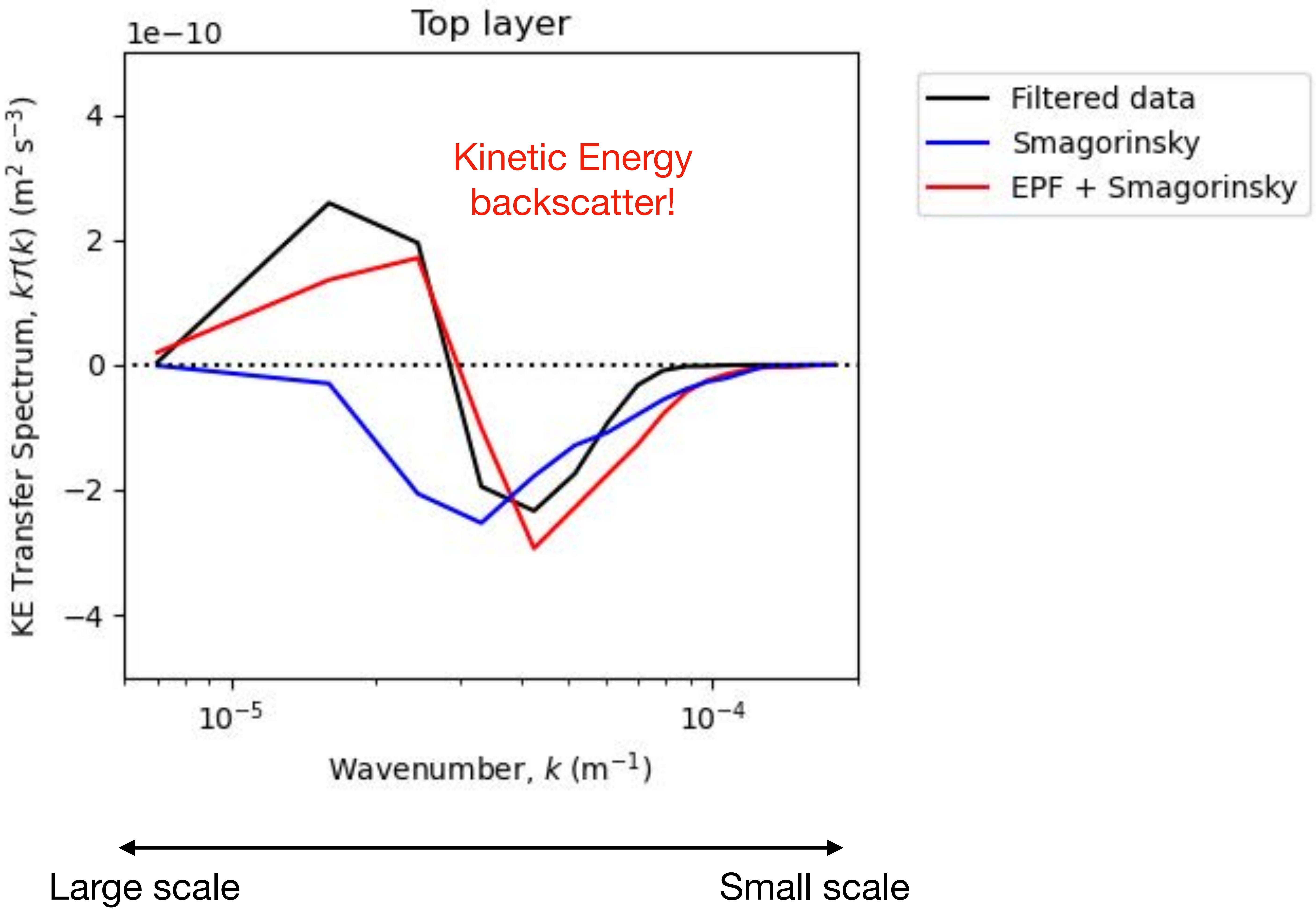


Large scale ⟷ Small scale

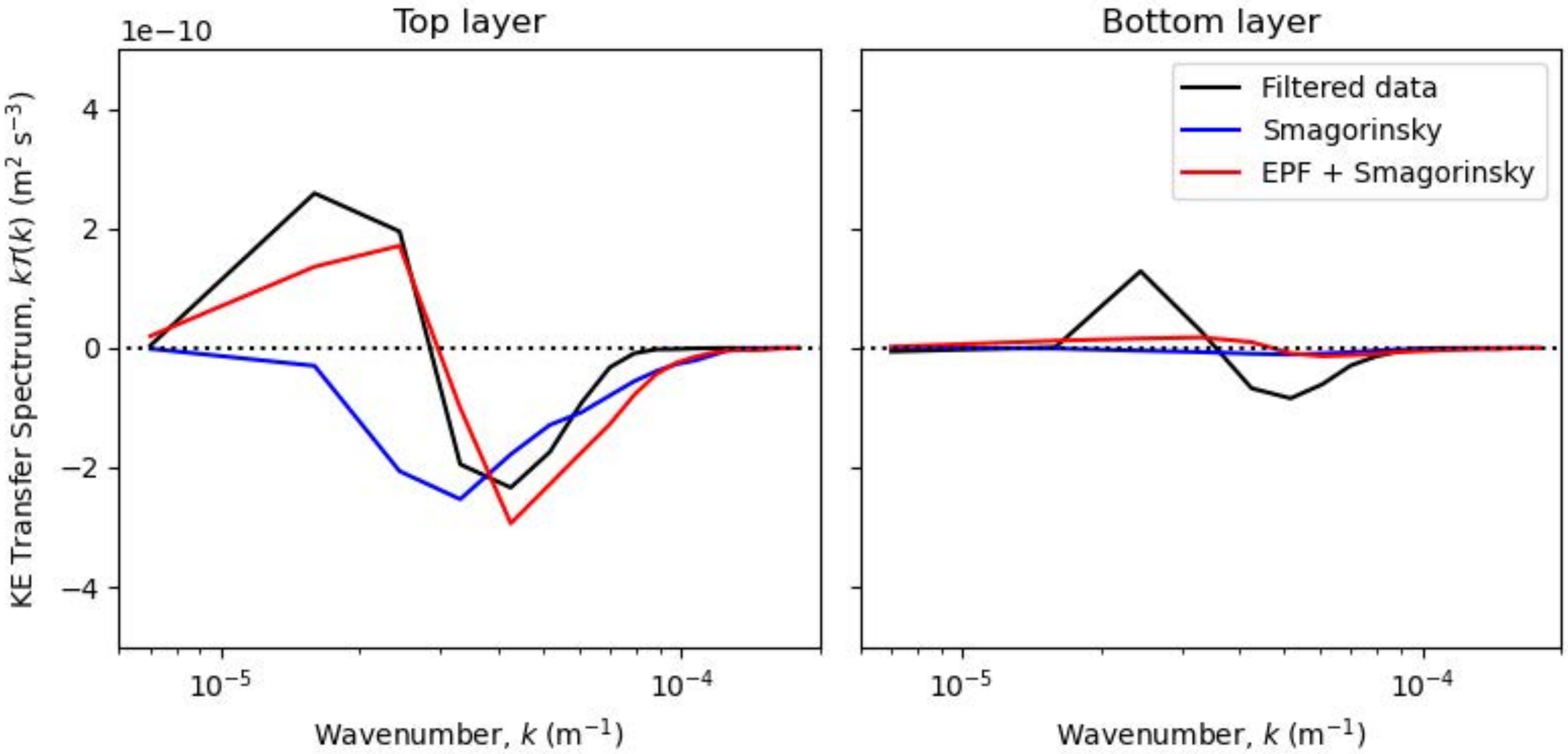
Isotropic Kinetic Energy transfer spectra



Isotropic Kinetic Energy transfer spectra



Isotropic Kinetic Energy transfer spectra



Not much improvement in bottom layer

Concluding remarks

Initial data-driven parametrization to capture the mesoscale energy cycle using the Eliassen-Palm Flux tensor components:

Idealised runs, double gyre

- **Offline performance**:
  - Room for improvement in training (more epochs, more data, regularization techniques, etc)
  - … however, improvements offline do not guarantee improvements online
- **Online performance**:
  - Doesn’t blow up!
  - Weakens pesky persistent eddy in mean flow
  - Better representation of energy spectral transfer
  - Still room for improvement

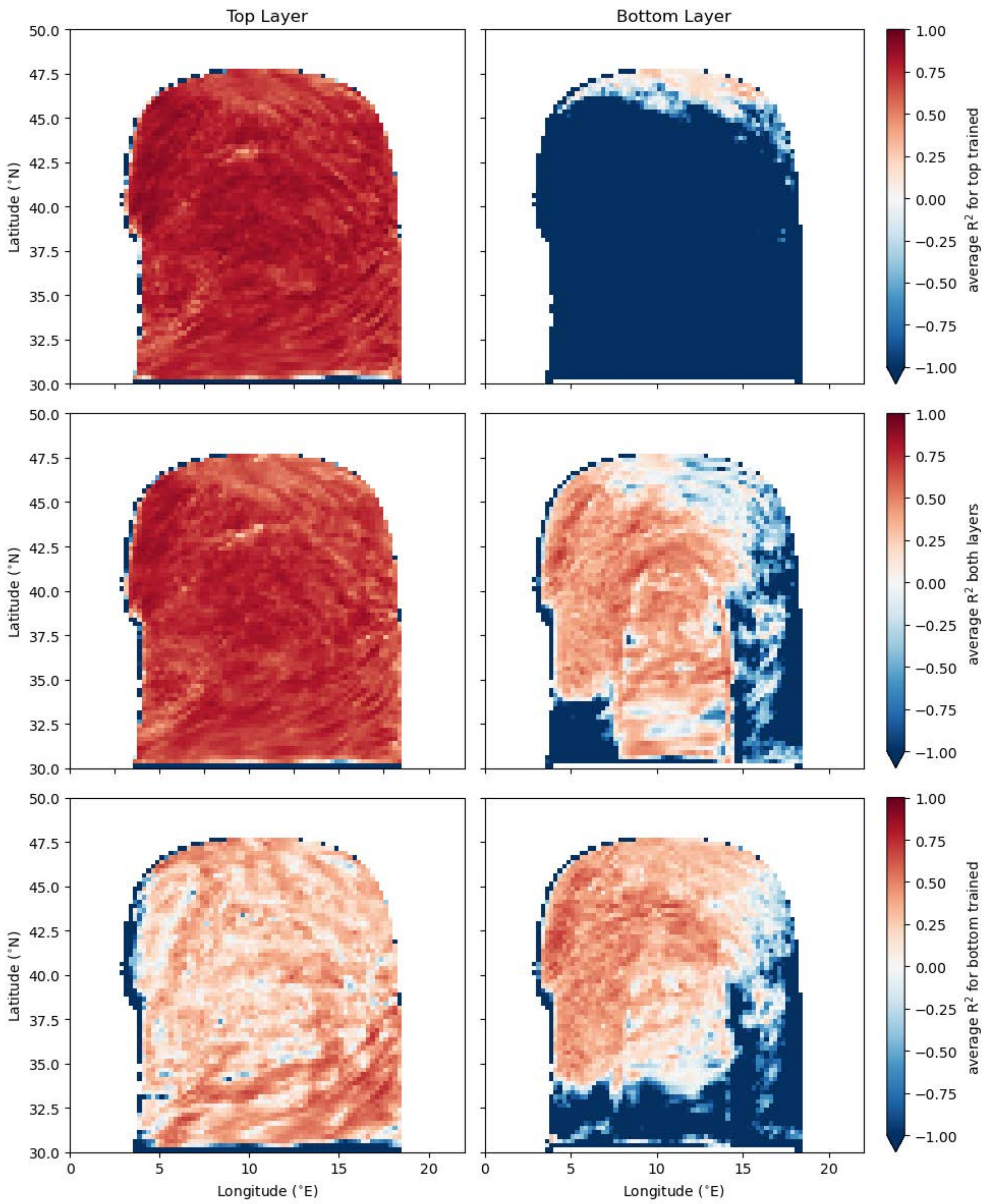
Upcoming:

- **How can we improve representation of thickness fluxes?**
  - Improve ANN to be implemented

Use of eddy energy system to produce a single parameterisation that captures both momentum and thickness fluxes seems promising 😊

Supplementary Slides

Supplementary Slide: **Weighting layer influence during training**

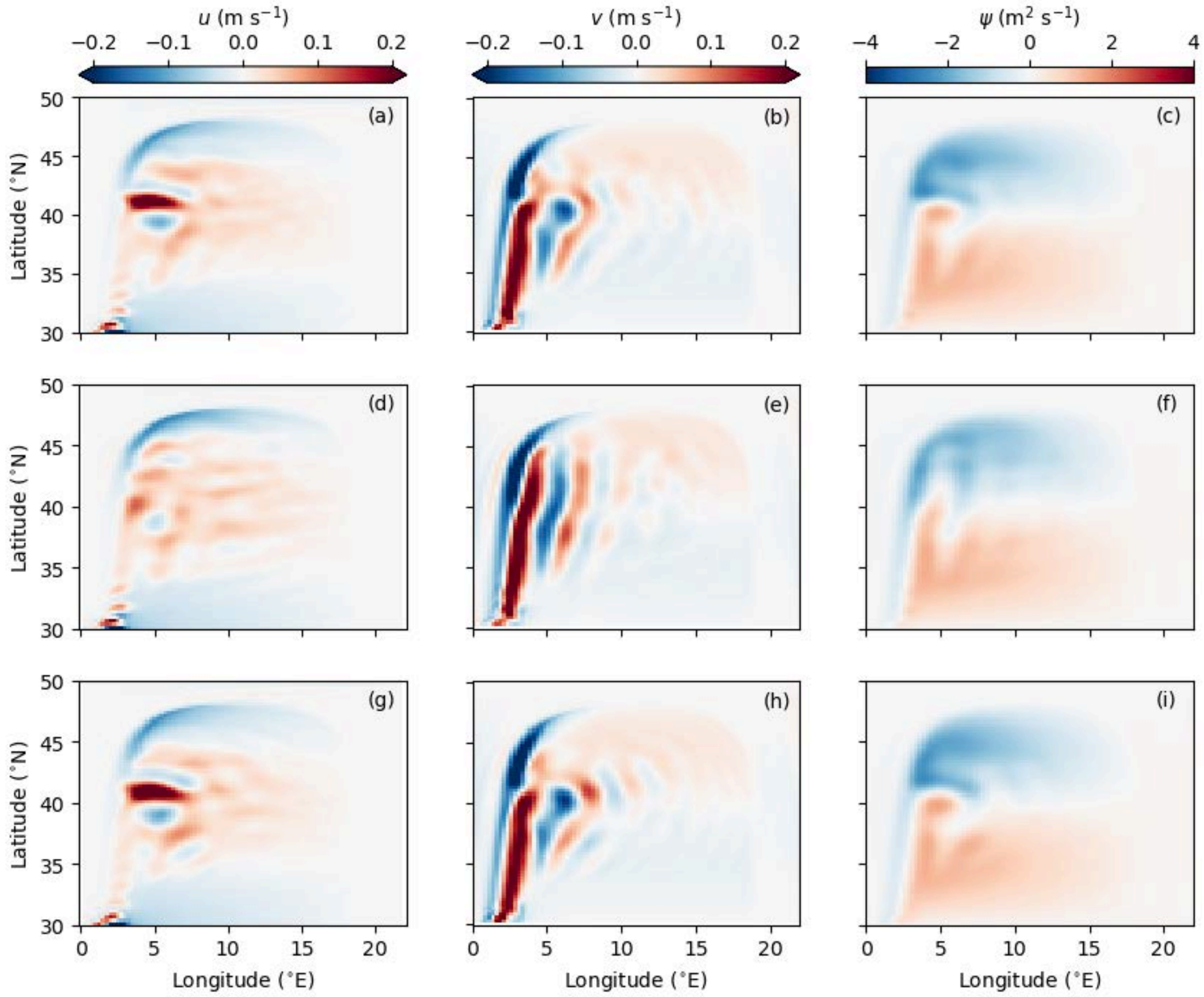


Alternate ANN trained, where back-propagated loss depends **only** on the **top layer**

'Original' ANN trained, where back-propagated loss depends **equally** on **top and bottom layer**

Alternate ANN trained, where back-propagated loss depends **only** on the **bottom layer**

Supplementary Slide: **EPF parameterization contributions** for top layer

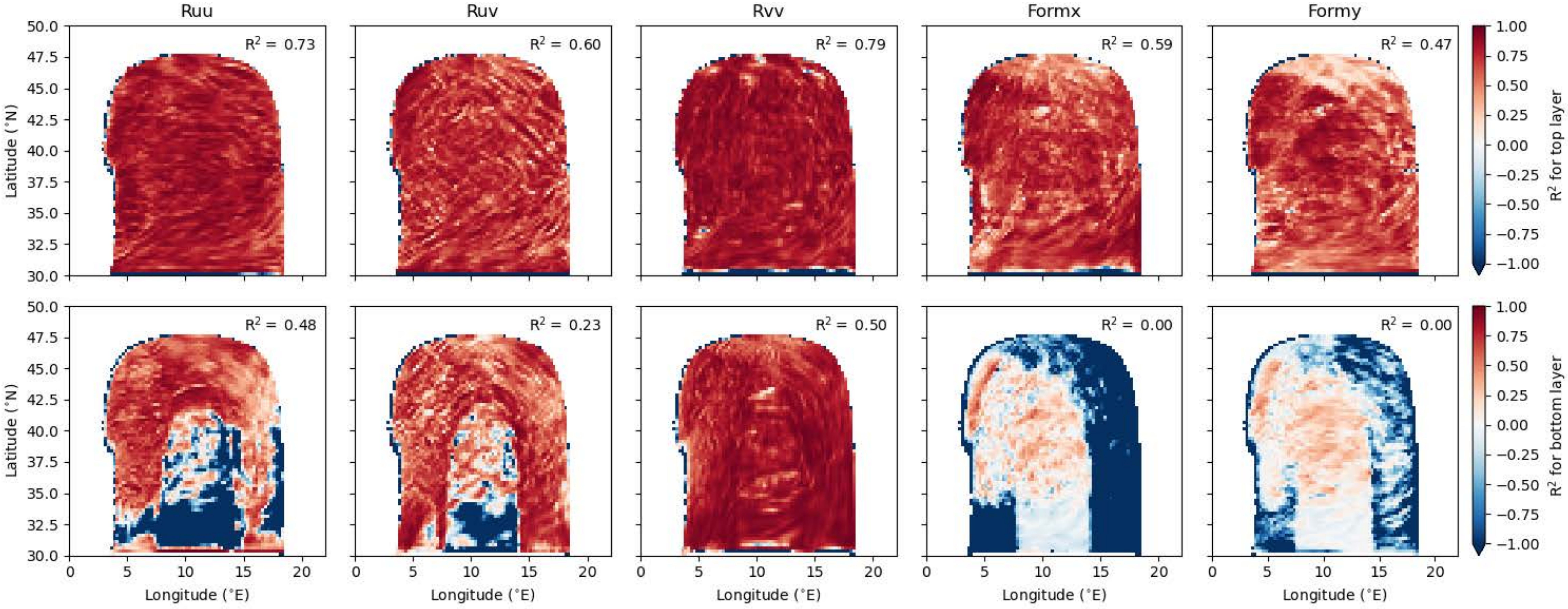


$1/4^{\circ}$ resolution DG run with **full EPF ANN parameterisation** (in addition to Smagorinsky)

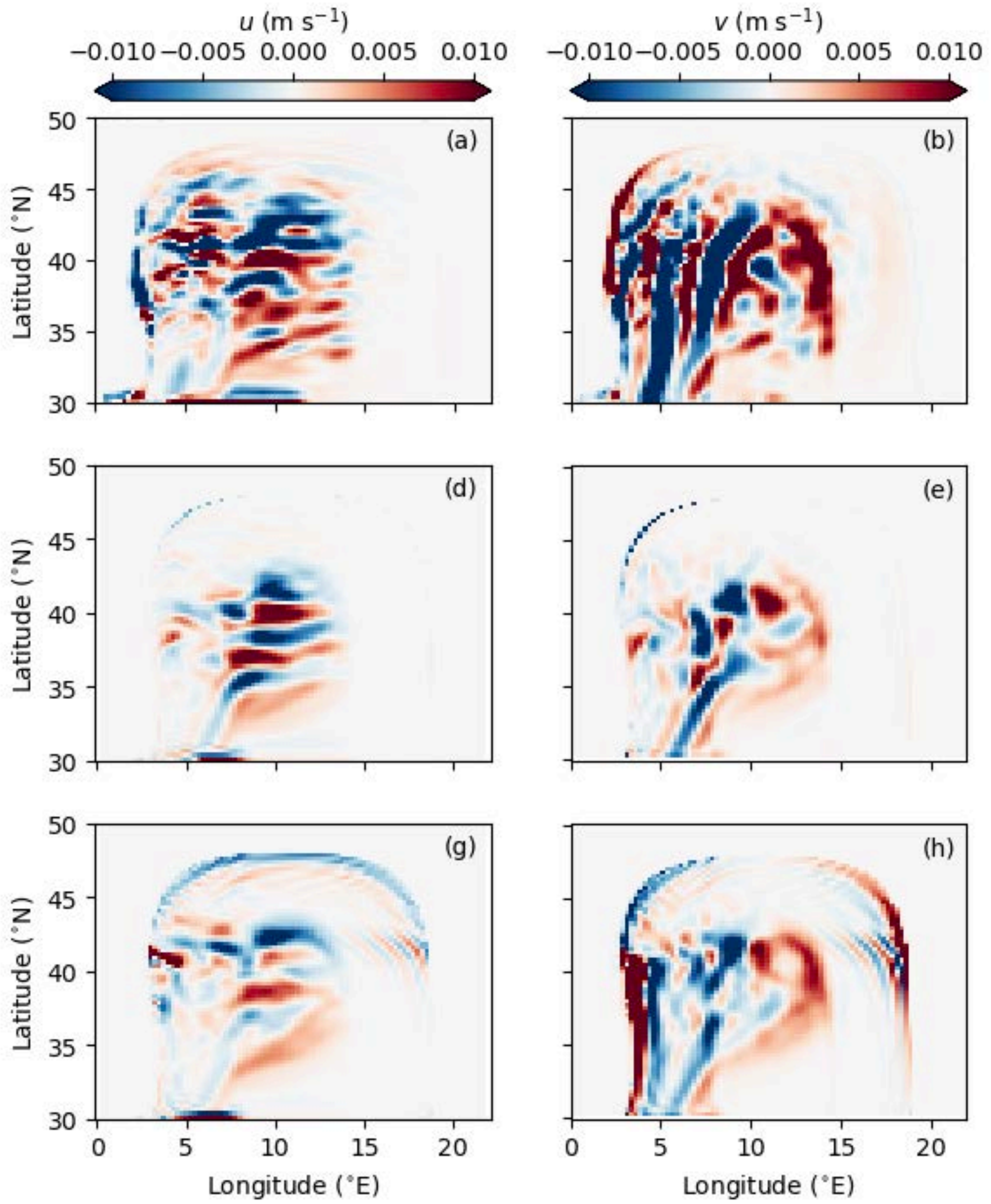
$1/4^{\circ}$ resolution DG run with **horizontal divergence of EPF** (in addition to Smagorinsky)

$1/4^{\circ}$ resolution DG run with **vertical divergence of EPF** (in addition to Smagorinsky)

Supplementary Slide: **ANN with bottom topography ignored**



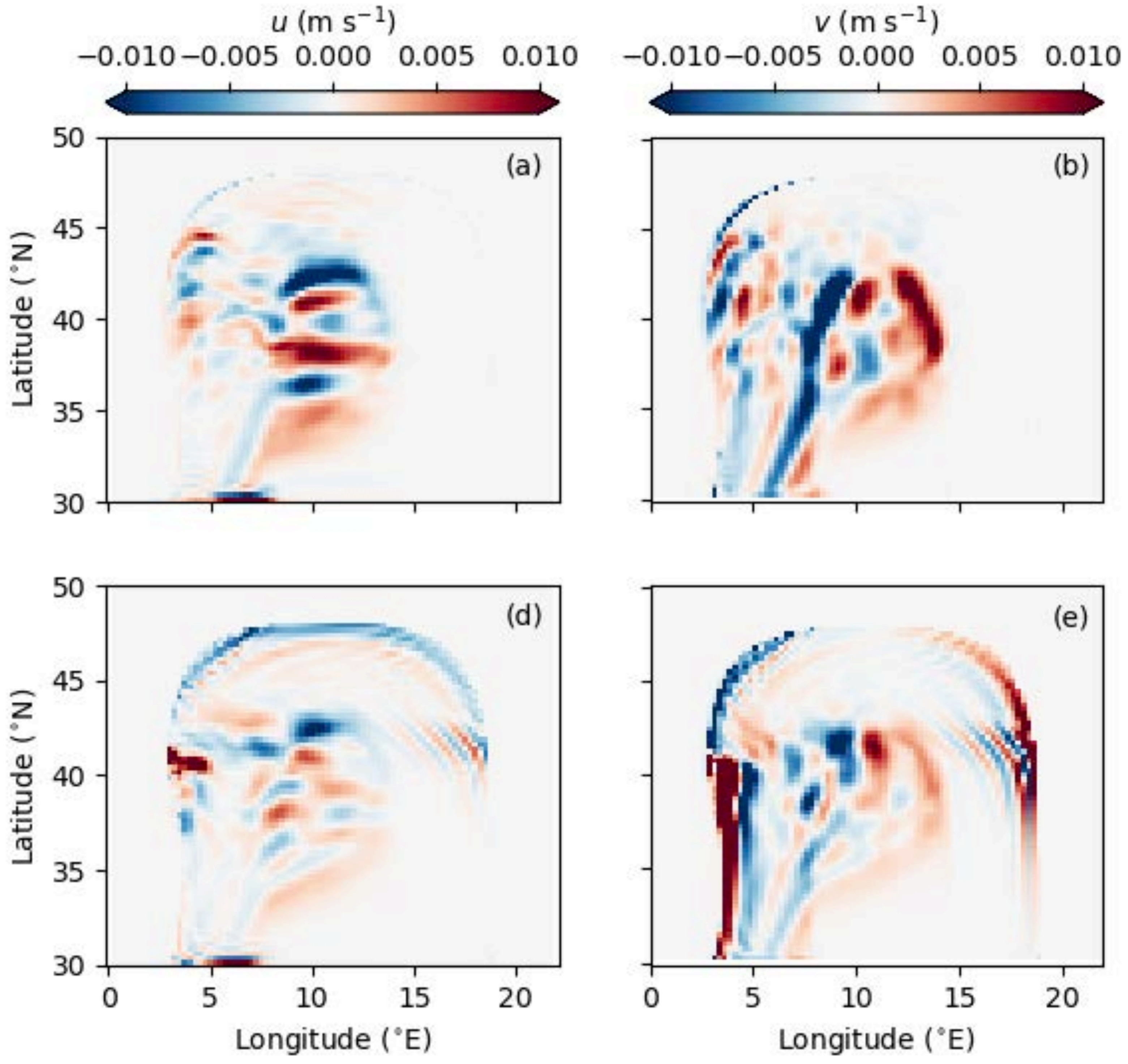
Large-scale flow field: Bottom layer



1/32° flow field filtered and coarsened to 1/4° resolution.

1/4° resolution run with **no parameterisation**

1/4° resolution run with EPF ANN parameterisation



Momentum flux only

Thickness flux only

Strength of Parameterisation

Does increasing the strength of the acceleration due to the parameterisation improve representation of the mean flow?