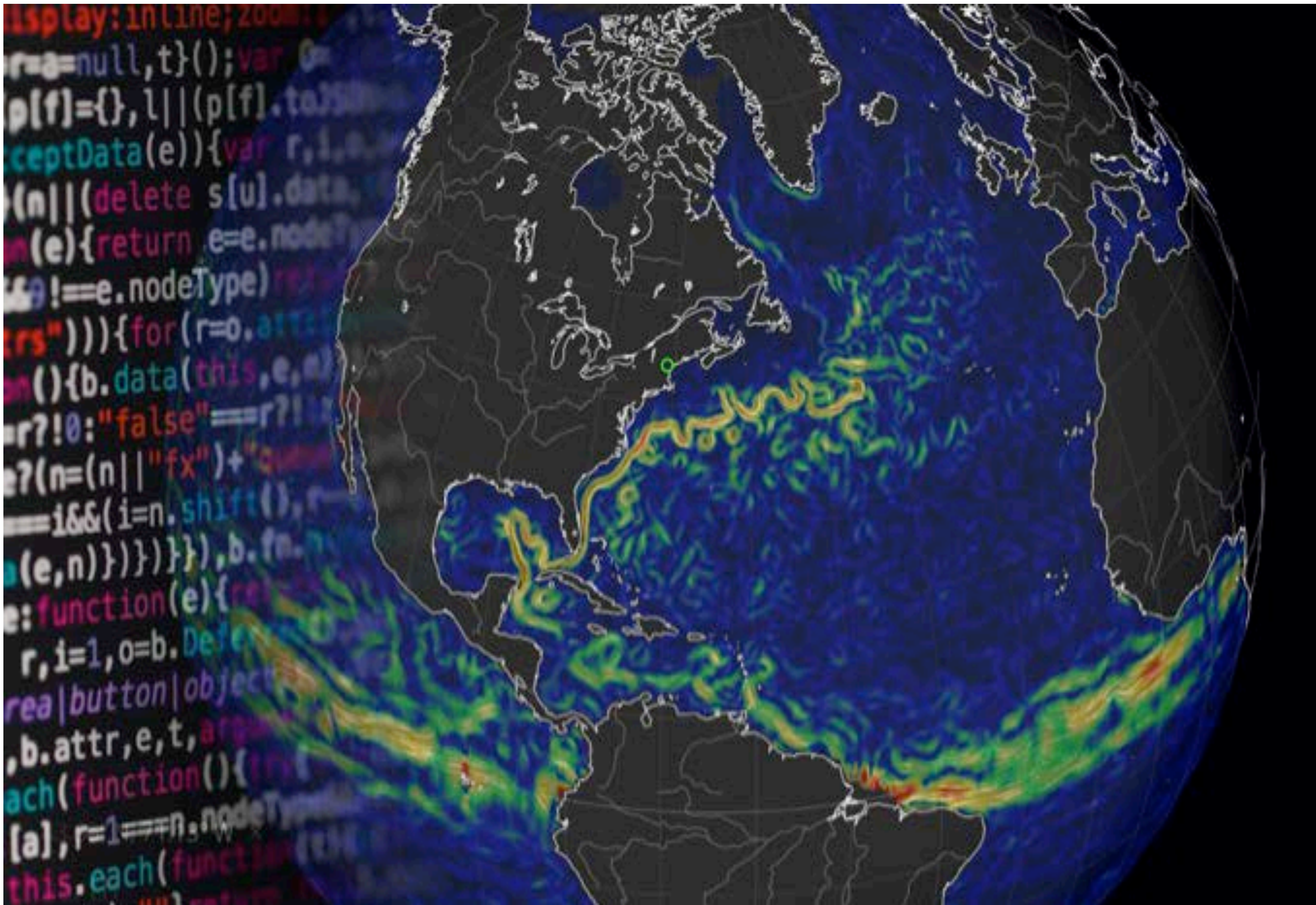
Deep Learning & Parametrization of Ocean Turbulence

Laure Zanna *Dept of Physics, Oxford & (soon) Courant, NYU*

Tom Bolton *Dept of Physics, Oxford*



Outline

- Introduction: parametrization of ocean turbulence

- Two data-driven — deep learning — tools for progress on parametrization

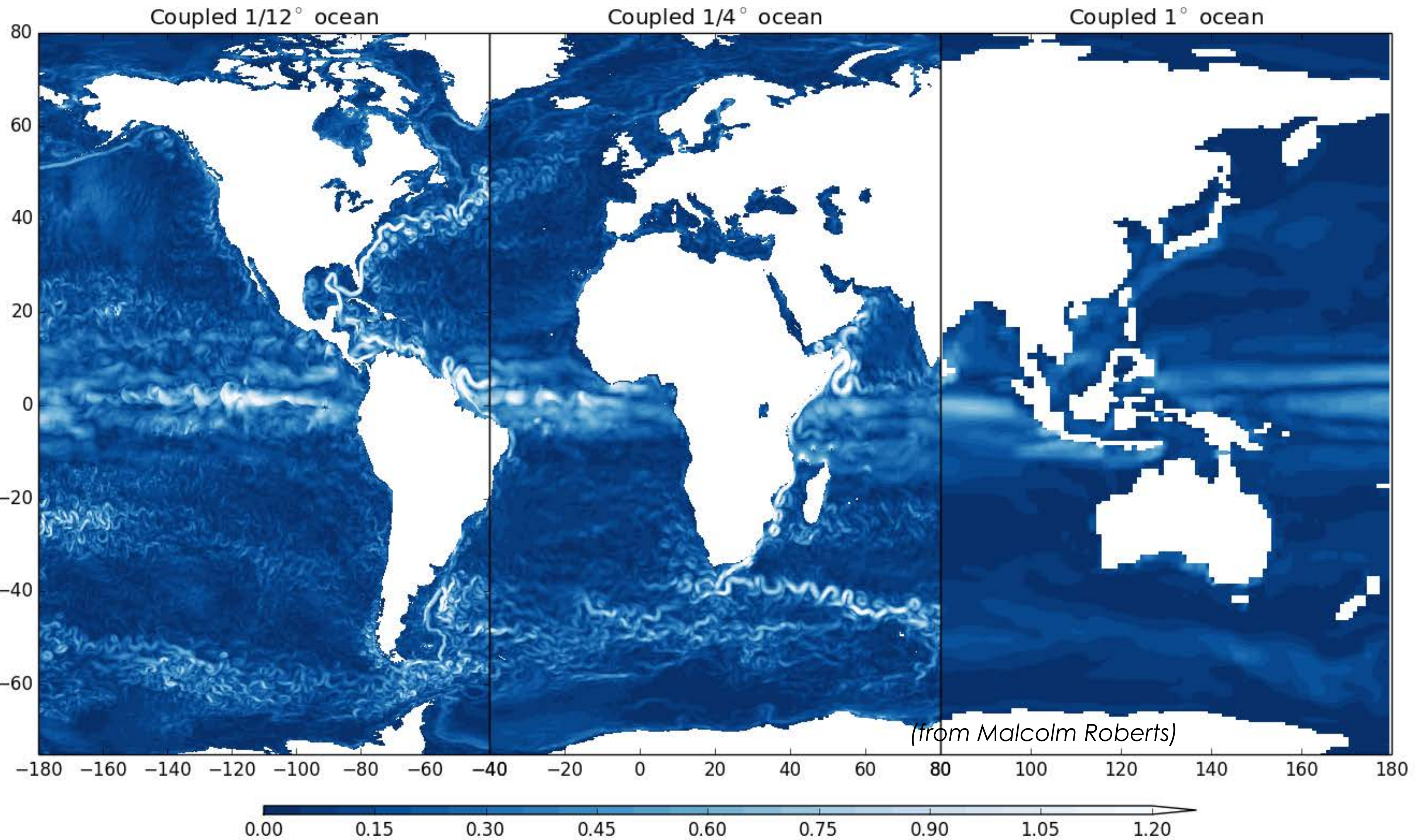
    ➡ Convolutional Neural Network *(Bolton & Zanna, JAMES 2019)*

    ➡ ODE/PDE discovery *(Bolton & Zanna, Proceedings Climate Informatics, Submitted)*

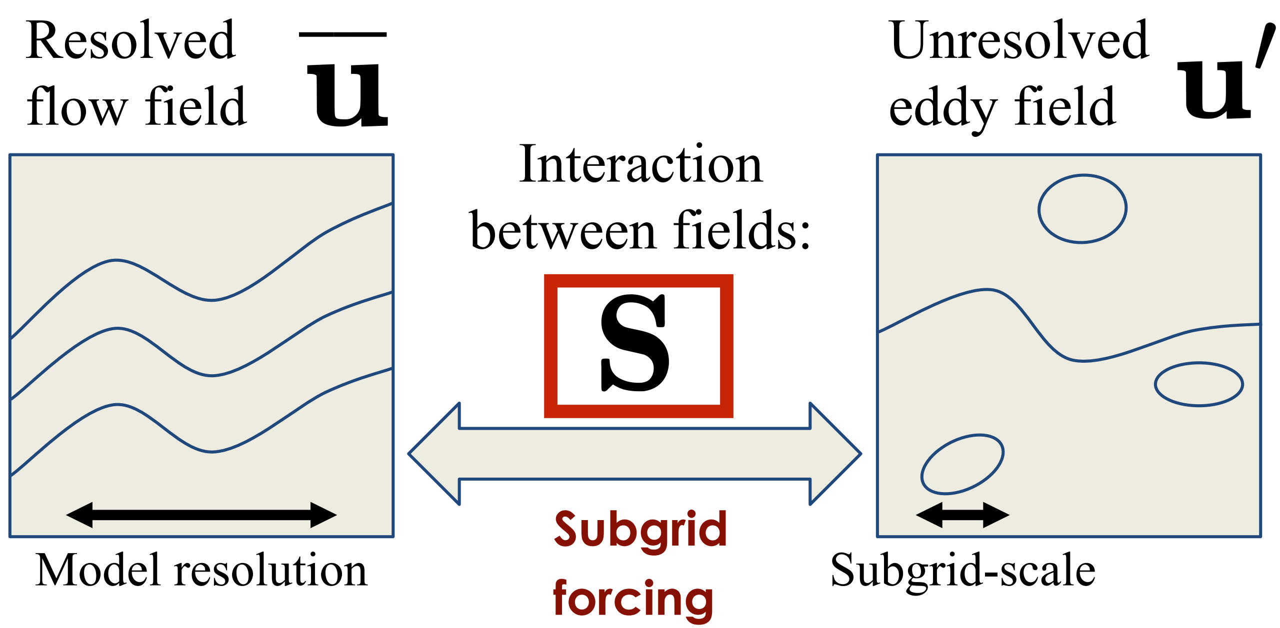
    → Preliminary Implementation in idealized models

- Conclusion & thoughts for the future of climate modelling

Limitations of Current Computing



The Closure/Parametrization Problem *(e.g., momentum)*

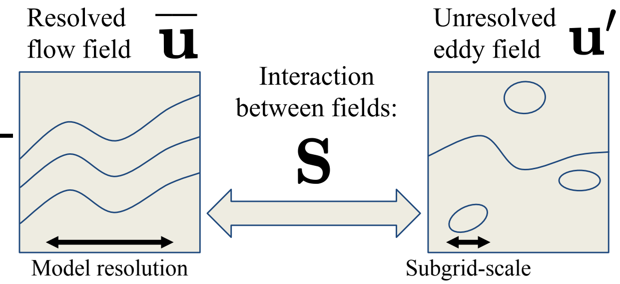


$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + (\overline{\mathbf{u}} \cdot \overline{\nabla})\overline{\mathbf{u}} = \overline{\mathbf{F}} + \overline{\mathbf{D}} + \boxed{\mathbf{S}}$$

resolved flow

Subgrid forcing

Traditional approach



▪ Consider physical mechanisms to approximate the bulk effect *(e.g. Smagorinsky 1963, Leith 1968)*

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + (\overline{\mathbf{u}} \cdot \overline{\nabla})\overline{\mathbf{u}} = \overline{\mathbf{F}} + \overline{\mathbf{D}} + \mathbf{S}$$

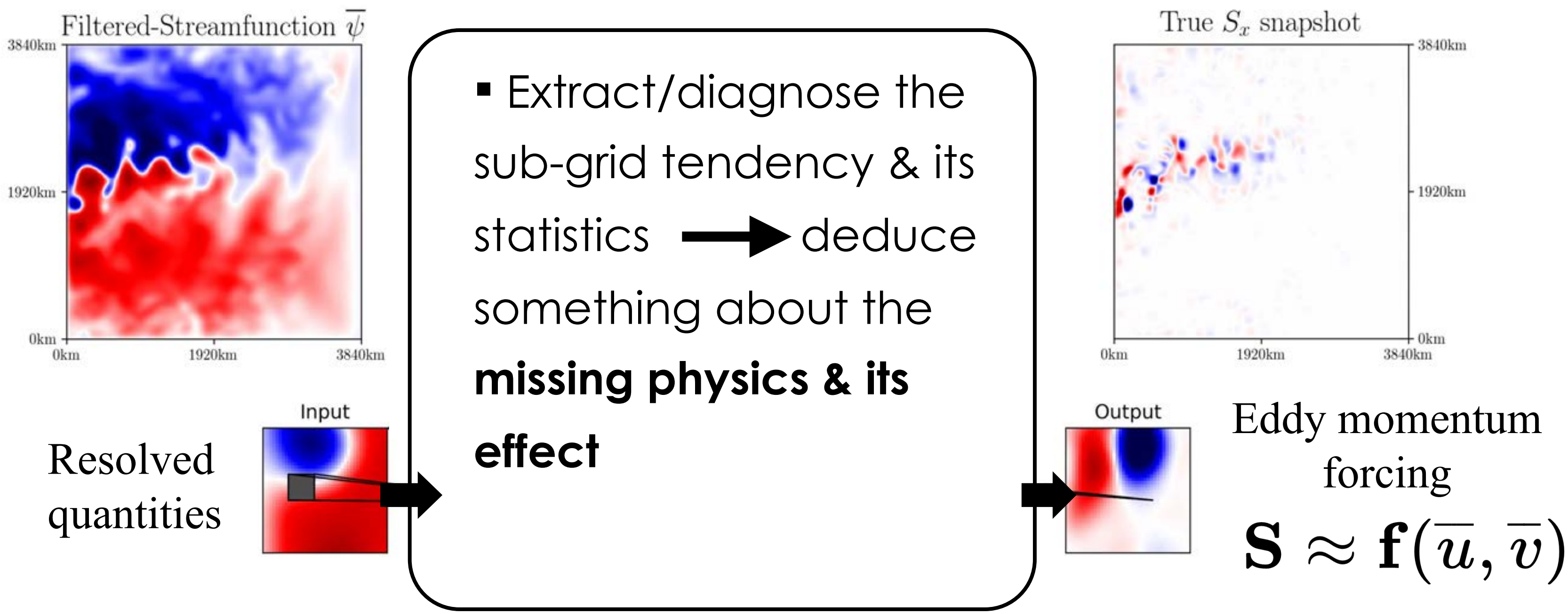
▪ Clear physical interpretation

▪ Computationally cheap to implement

▪ Parameter from theory, observations or tuning

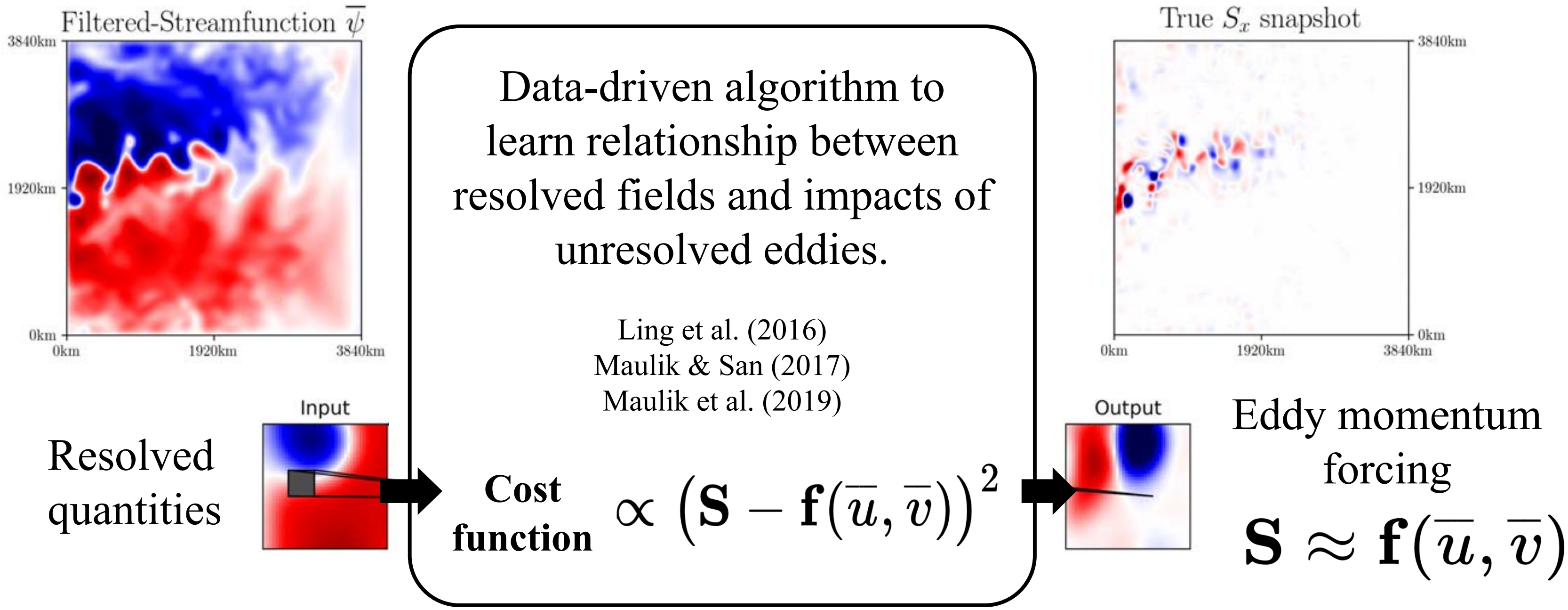
▪ *Caveats*: often assumes down-gradient fluxes, difficult to capture all dynamical & thermodynamic flow-dependent + local/non-local effects

Data-driven Approach



- Could capture physical processes that current parametrizations do not
- Can represent highly nonlinear spatio-temporal variability

Data-driven Approach



- Could capture physical processes that current parametrizations do not
- Can represent highly nonlinear spatio-temporal variability
- *Caveats*: act as a black-box, may not work outside the training data, & may not respect physical/conservation laws

Turbulence: Ling et al., 2016; Wang et al., 2017
Atm: Brenowitz & Bretherton, 2018; Gentine et al., 2018; Jiang et al., 2018; O'Gorman & Dwyer, 2018
Ocean: Bolton & Zanna 2019

Outline

- Introduction: parametrization of ocean turbulence

- Two data-driven — deep learning — tools for progress for parametrization

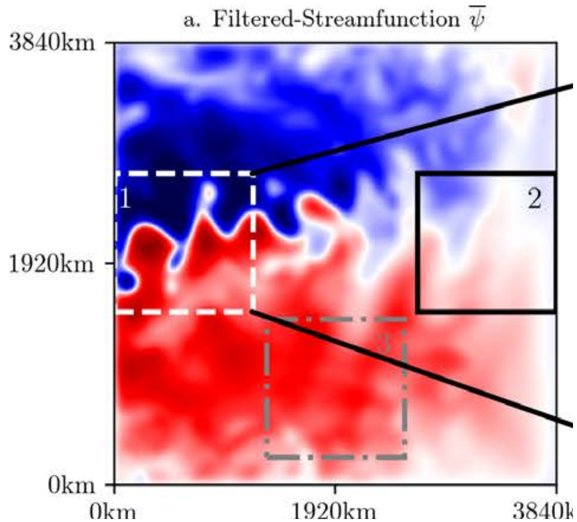
➡ **Convolutional Neural Network** ***(Bolton & Zanna, JAMES 2019)***

➡ ODE/PDE discovery *(Bolton & Zanna, Proceedings Climate Informatics, Submitted)*

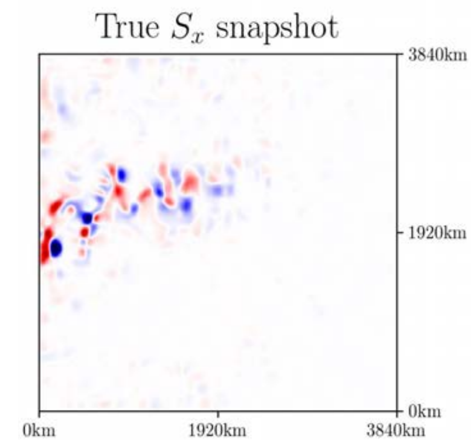
Preliminary Implementation in idealized models

- Conclusion & thoughts for the future of climate modelling

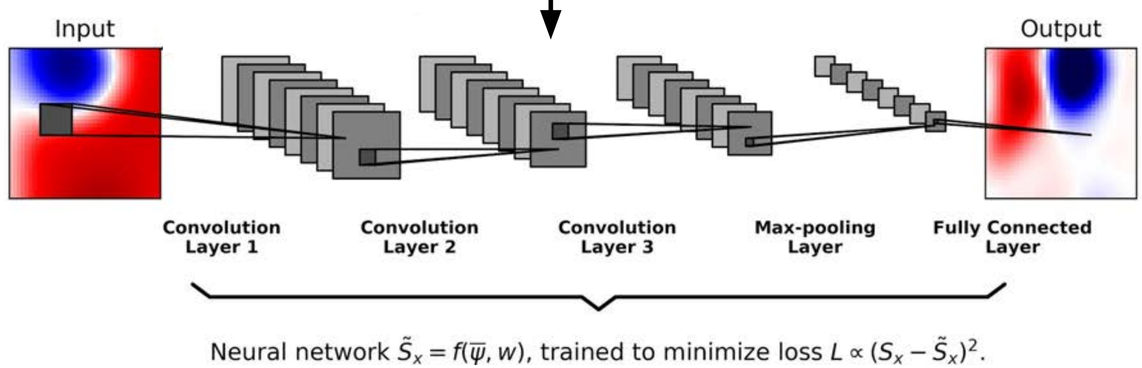
Architecture of the Convolutional Neural Network



Convolutional neural network for $\mathbf{f}(\overline{u}, \overline{v})$



Resolved quantities

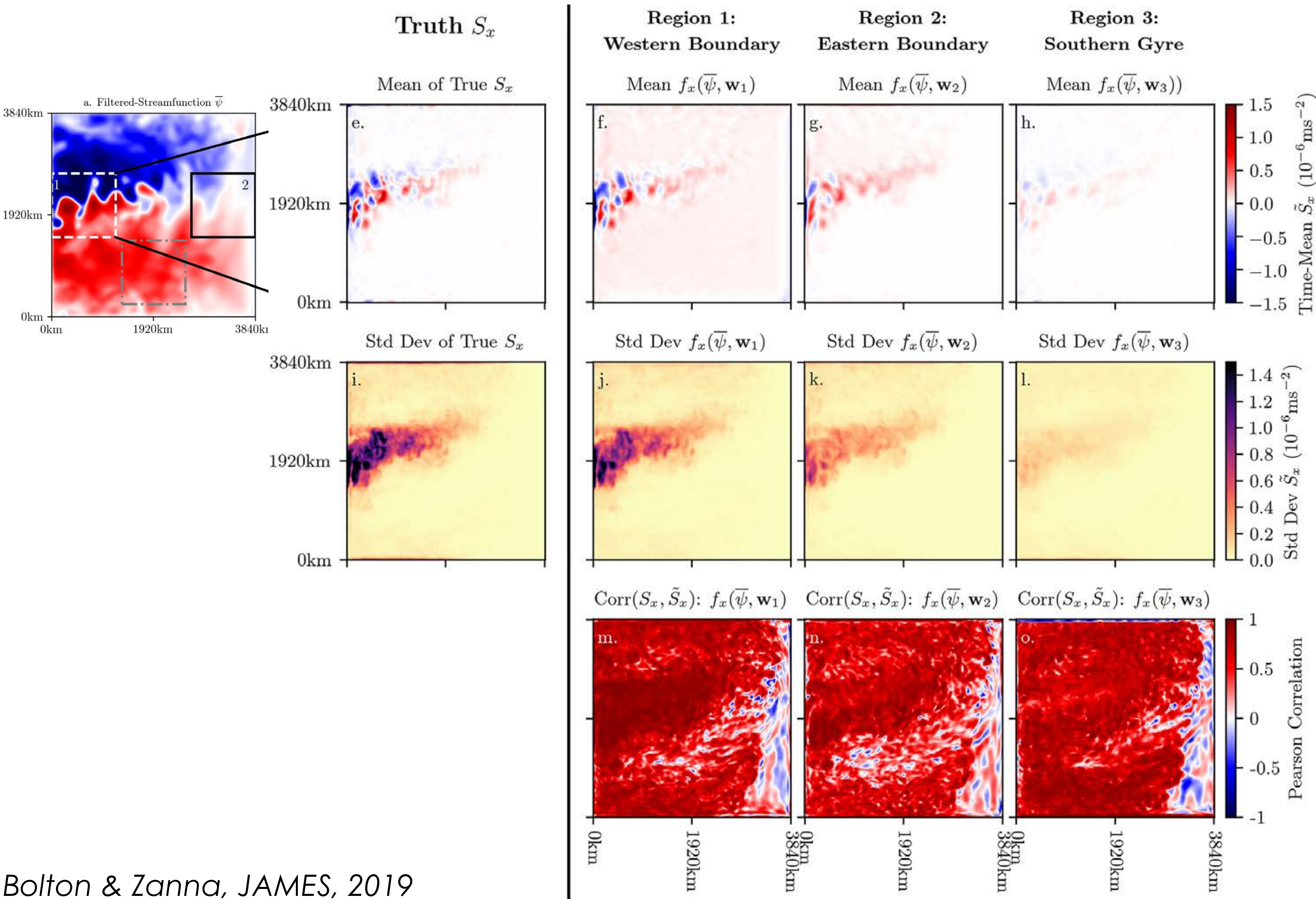


Eddy momentum forcing

$$\mathbf{S} \approx \mathbf{f}(\overline{u}, \overline{v})$$

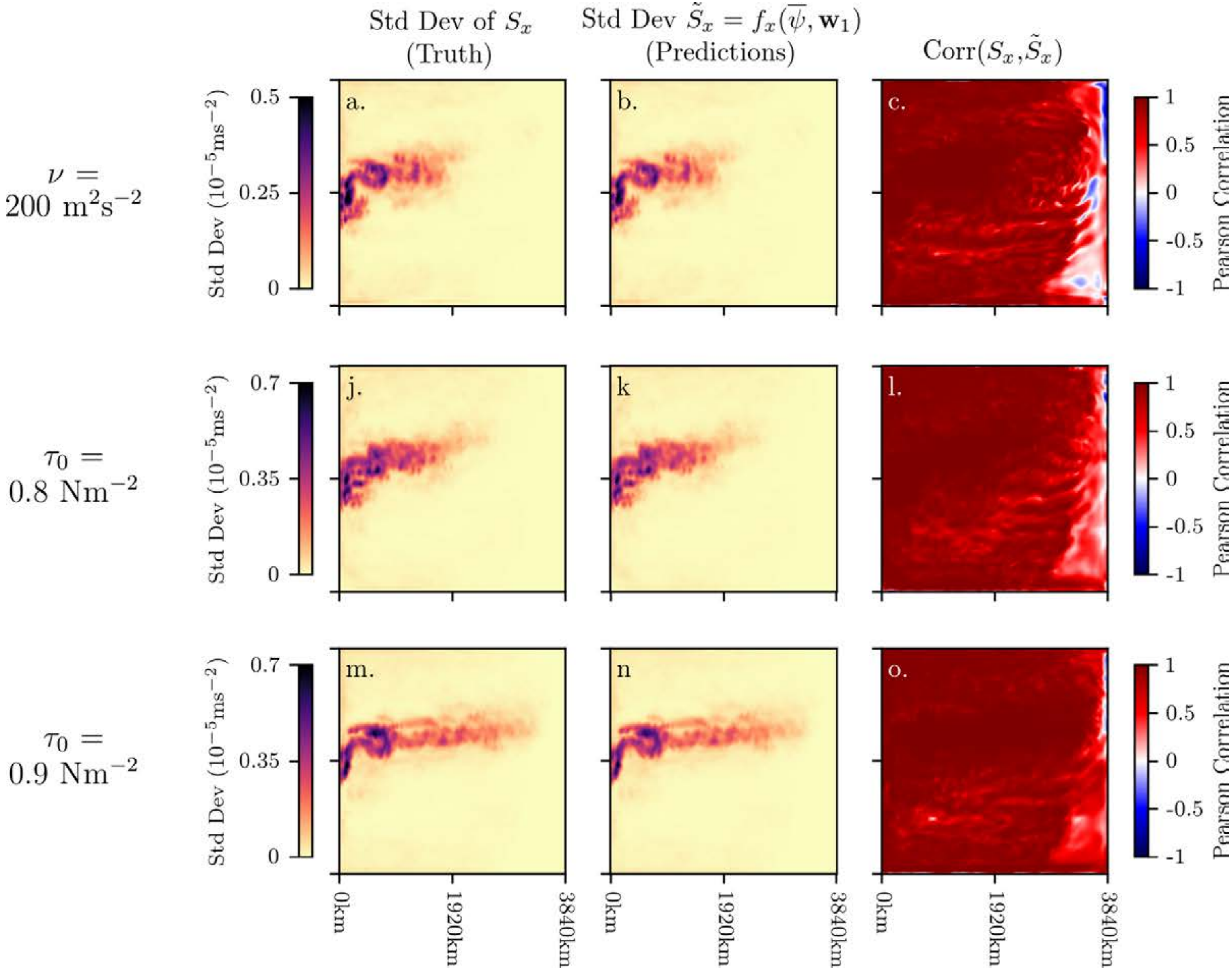
Bolton & Zanna, JAMES, 2019

Non-local Generalisation



Bolton & Zanna, JAMES, 2019

Generalization to other dynamical regimes



Bolton & Zanna, JAMES, 2019

Summary of CNN-based Eddy Parametrization

- Convolutional Neural Networks can be successfully trained to mimic eddy momentum forcing

  - *Caveats ?*

  ➡ act as a black-box: *extracting derivatives*

  ➡ may not work outside the training data: *generalisation to different Reynolds numbers*

  ➡ *may not respect physical laws: conservations can be imposed within the architecture*

➡ **CNNs = a "good" basis for a new set of physics-aware machine learning-derived sub-grid parametrizations *(to complement traditional approaches)***

Yet, CNNs cannot be written as a "mathematical operator" which is well-defined &can be studied …

Bolton & Zanna, JAMES, 2019; Code on GitHub

Outline

- Introduction: parametrization of ocean turbulence

- Two data-driven — deep learning — tools for progress on parametrization

➡ Convolutional Neural Network *(Bolton & Zanna, JAMES 2019)*

➡ **ODE/PDE discovery** ***(Bolton & Zanna, Proceedings Climate Informatics, Submitted)***

Preliminary Implementation in idealized models

- Conclusion & thoughts for the future of climate modelling

Data-driven PDE discovery

1. Spatio-temporal data

High-resolution models or observations

→

2. Library of functions

Gradients and products of resolved velocities.

→

3. Iterative sparse regression

Repeatedly prune library of functions.

→

4. Data-driven equation

Remaining functions form the final result.

High-resolution dada from idealised geometry MITgcm barotropic & baroclinic simulations

$$\mathbf{S} = (\overline{\mathbf{u}} \cdot \overline{\nabla})\overline{\mathbf{u}} - \overline{(\mathbf{u} \cdot \nabla)\mathbf{u}}$$

Data-driven PDE discovery

1. Spatio-temporal data

High-resolution models or observations

→

2. Library of functions

Gradients and products of resolved velocities.

→

3. Iterative sparse regression

Repeatedly prune library of functions.

→

4. Data-driven equation

Remaining functions form the final result.

Divergence: $\dot{\sigma} = \overline{\nabla} \cdot \overline{\mathbf{u}}$

Vorticity: $\zeta = \overline{\nabla} \times \overline{\mathbf{u}}$

Shearing deformation: $D = \dfrac{\partial \overline{u}}{\partial y} + \dfrac{\partial \overline{v}}{\partial x}$

Stretching deformation: $\tilde{D} = \dfrac{\partial \overline{u}}{\partial x} - \dfrac{\partial \overline{v}}{\partial y}$

Enforce solution to be divergence of a flux: $f(\overline{u}, \overline{v}) = \overline{\nabla} \cdot \mathbf{F}$

Data-driven PDE discovery

1. Spatio-temporal data

High-resolution models or observations

2. Library of functions

Gradients and products of resolved velocities.

3. Iterative sparse regression

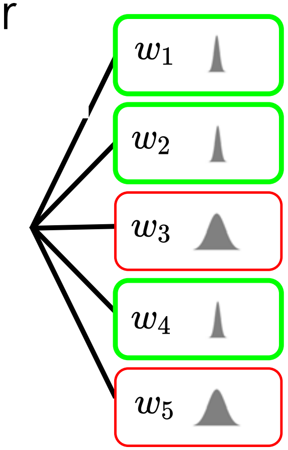
Repeatedly prune library of functions.

4. Data-driven equation

Remaining functions form the final result.

Sparse **Bayesian** learning, using relevance vector machines (RVM)

$$f(\overline{u},\overline{v}) = \sum_i w_i \phi_i(\overline{u},\overline{v})$$



Inspired by: Zhang & Lin (2018)
"Robust data-driven discovery of governing physical laws with error bars"

Data-driven PDE discovery

1. Spatio-temporal data

High-resolution models or observations

2. Library of functions

Gradients and products of resolved velocities.

3. Iterative sparse regression

Repeatedly prune library of functions.

4. Data-driven equation

Remaining functions form the final result.

An equation for S, the unresolved eddy momentum forcing, based on resolved variables

$$\mathbf{S} = \overline{\nabla} \cdot \mathbf{T}$$

$$\mathbf{T} = \mathbf{T}(\overline{u}, \overline{v})$$

Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

$$\mathbf{f}(\overline{u}, \overline{v}) = \underset{\text{(scalar)}}{\kappa}\overline{\nabla} \cdot \begin{pmatrix} \zeta^2 - \zeta D & \zeta\tilde{D} \\ \zeta\tilde{D} & \zeta^2 + \zeta D \end{pmatrix}$$

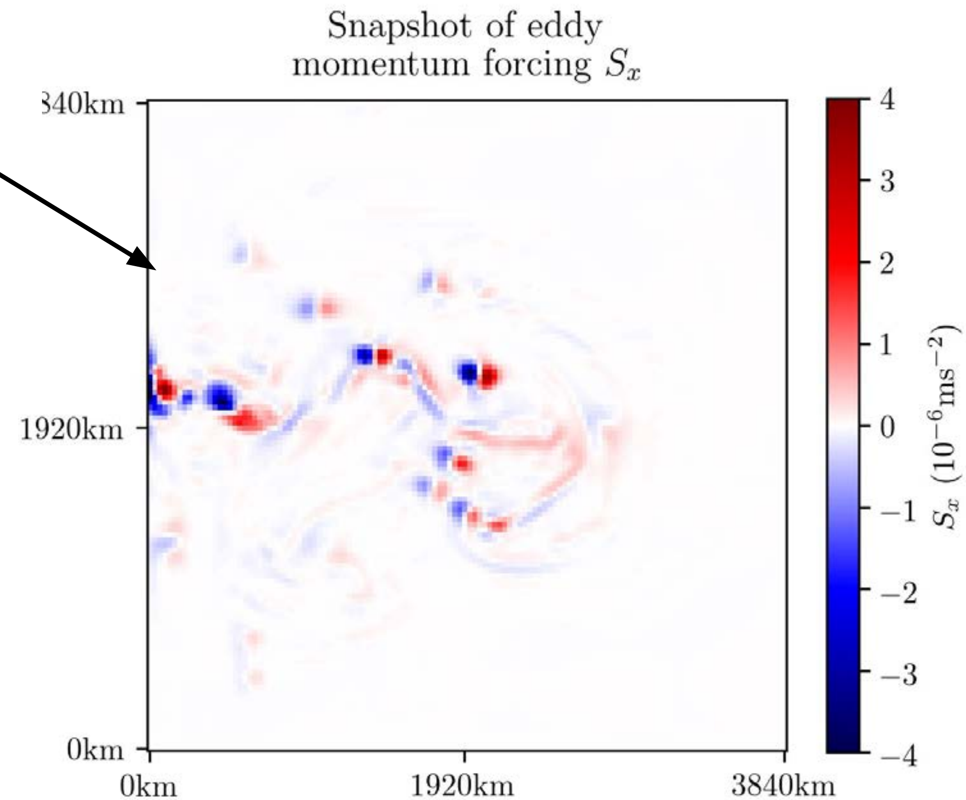
Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

$$\mathbf{f}(\overline{u}, \overline{v}) = \underbrace{\kappa}_{\text{(scalar)}} \overline{\nabla} \cdot \begin{pmatrix} \zeta^2 - \zeta D & \zeta \tilde{D} \\ \zeta \tilde{D} & \zeta^2 + \zeta D \end{pmatrix}$$

- Captures ~54% of the variance
- Extracted symmetric stress tensor with no a priori knowledge
- Expression conserves global momentum & vorticity

$$\begin{pmatrix} \zeta & = \text{vorticity} \\ D & = \text{shearing deformation} \\ \tilde{D} & = \text{stretching deformation} \end{pmatrix}$$



Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

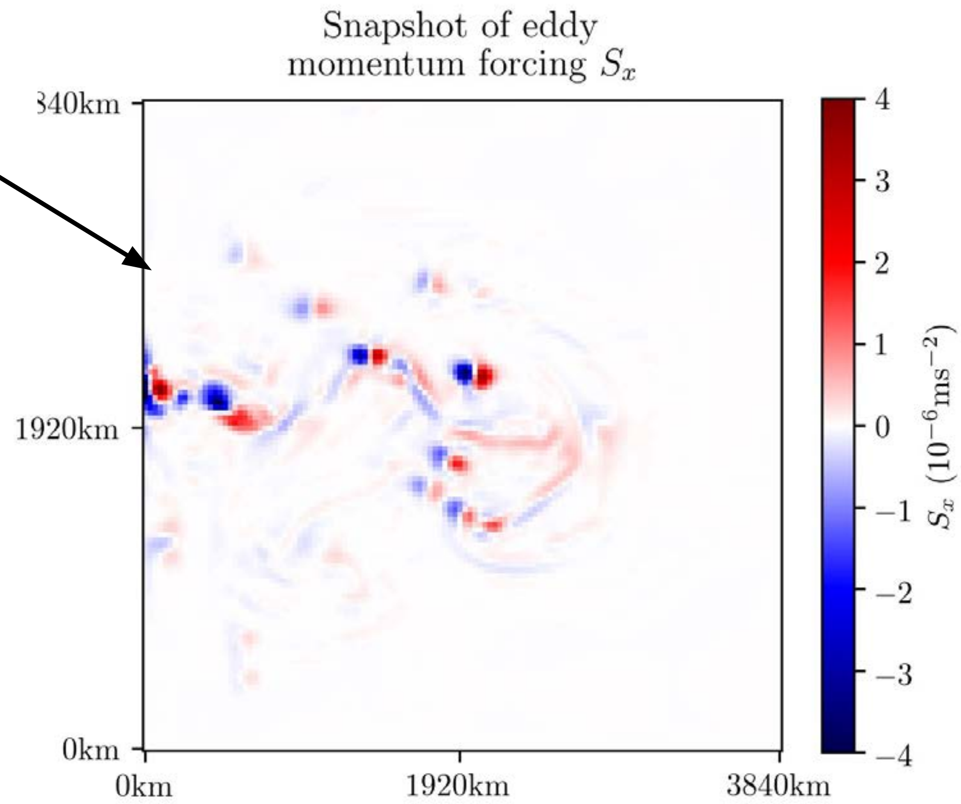
$$\mathbf{f}(\overline{u}, \overline{v}) = \underset{\text{(scalar)}}{\kappa} \overline{\nabla} \cdot \begin{pmatrix} \zeta^2 - \zeta D & \zeta \tilde{D} \\ \zeta \tilde{D} & \zeta^2 + \zeta D \end{pmatrix}$$

Contains the deformation based parameterisation of Anstey & Zanna (2017)

$$\overline{\nabla} \cdot \begin{pmatrix} -\zeta D & \zeta \tilde{D} \\ \zeta \tilde{D} & \zeta D \end{pmatrix}$$

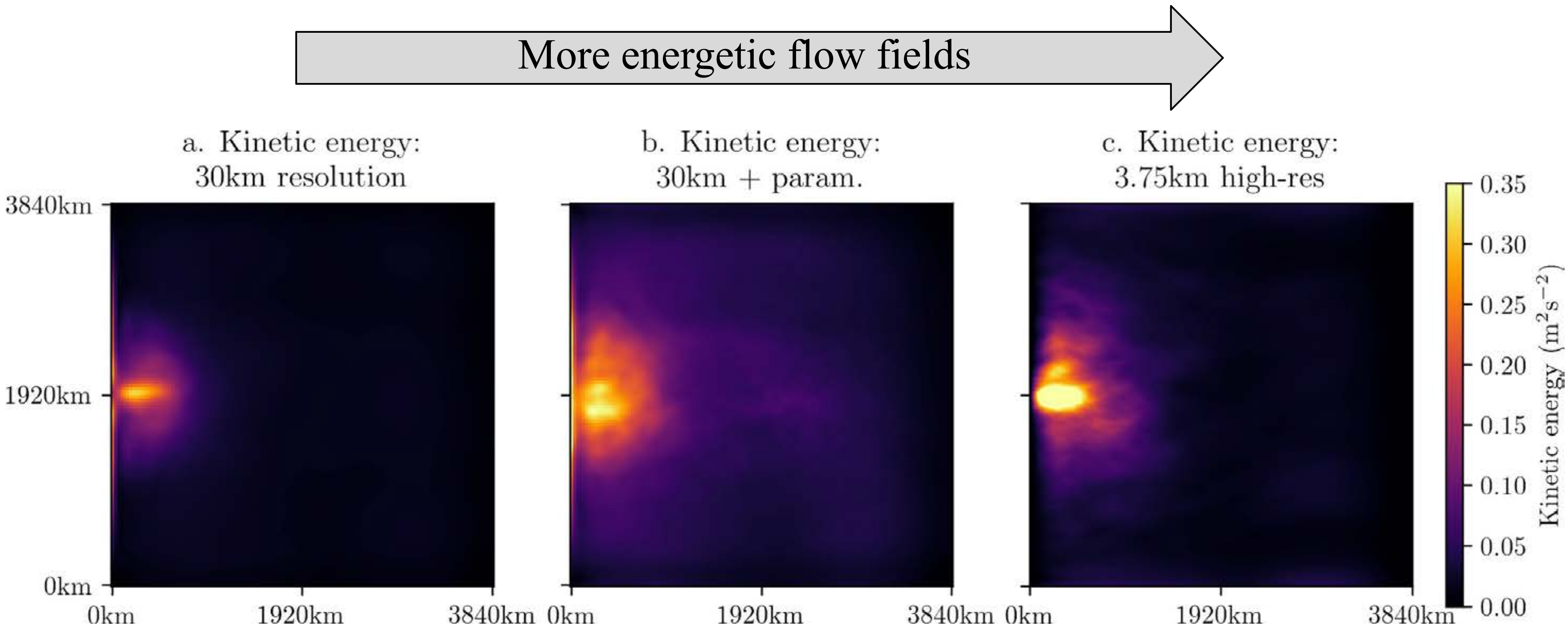
- Captures ~54% of the variance
- Extracted symmetric stress tensor with no a priori knowledge
- Expression conserves global momentum & vorticity

ζ = vorticity
D = shearing deformation
$\tilde{D}$ = stretching deformation



Implementation in barotropic model

Must correct spurious loss of kinetic energy (Mana & Zanna 2014; Jansen et al 2015; Zanna et al 2017)



Resolution: 30km	Resolution: 30km	Resolution: 3.75km
No parameterisation	+ $\overline{\overline{\nabla}} \cdot \begin{pmatrix} \zeta^2 - \zeta D & \zeta\tilde{D} \\ \zeta\tilde{D} & \zeta^2 + \zeta D \end{pmatrix}$	High resolution

Conclusions & Thoughts

- *Machine learning can reveal implicit (CNNs) or explicit (RVM) novel mesoscale eddy parameterisations for use in ocean & climate models*
- Can respect physical conservation law & generalise well to other regimes
- Implementation: More energetic flow, correcting biases due to limited resolution & accuracy of numerical schemes

Conclusions

- *Machine learning can reveal implicit (CNNs) or explicit (RVM) novel mesoscale eddy parameterisations for use in ocean & climate models*

- Can respect physical conservation law & generalise well to other regimes

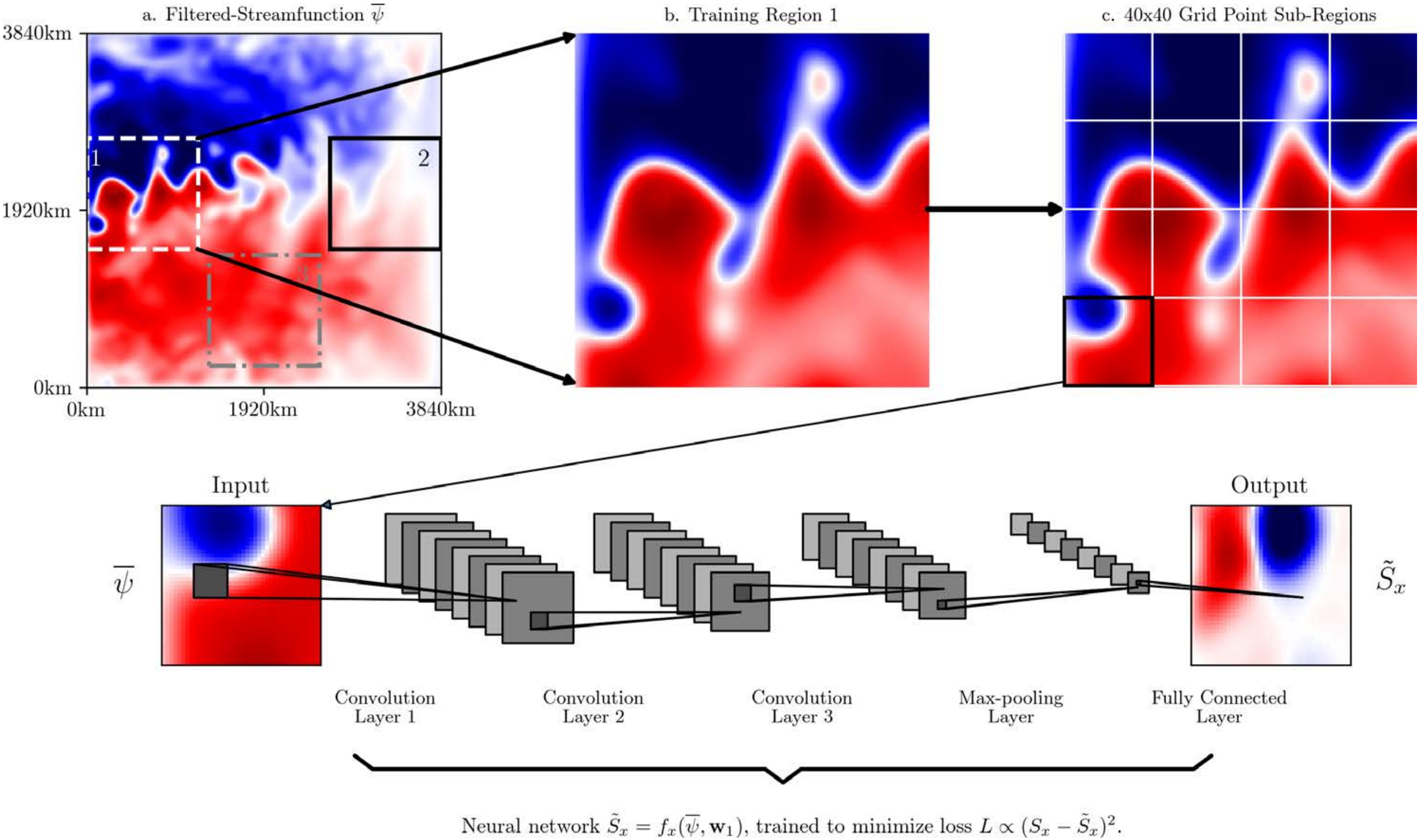
- Implementation: More energetic flow, correcting biases due to limited resolution & accuracy of numerical schemes

Some Current/Future Work + Thoughts

- **Long-term aim for model improvements:**
  - a closed set of data-driven sub-grid parametrizations *(in addition to current approaches)*
  - Expand the search for closure using high-resolution & complex output from models & observations *(e.g., new satellite missions or in situ data)*
  - Error estimates (Bayesian) for different closures (model uncertainty)

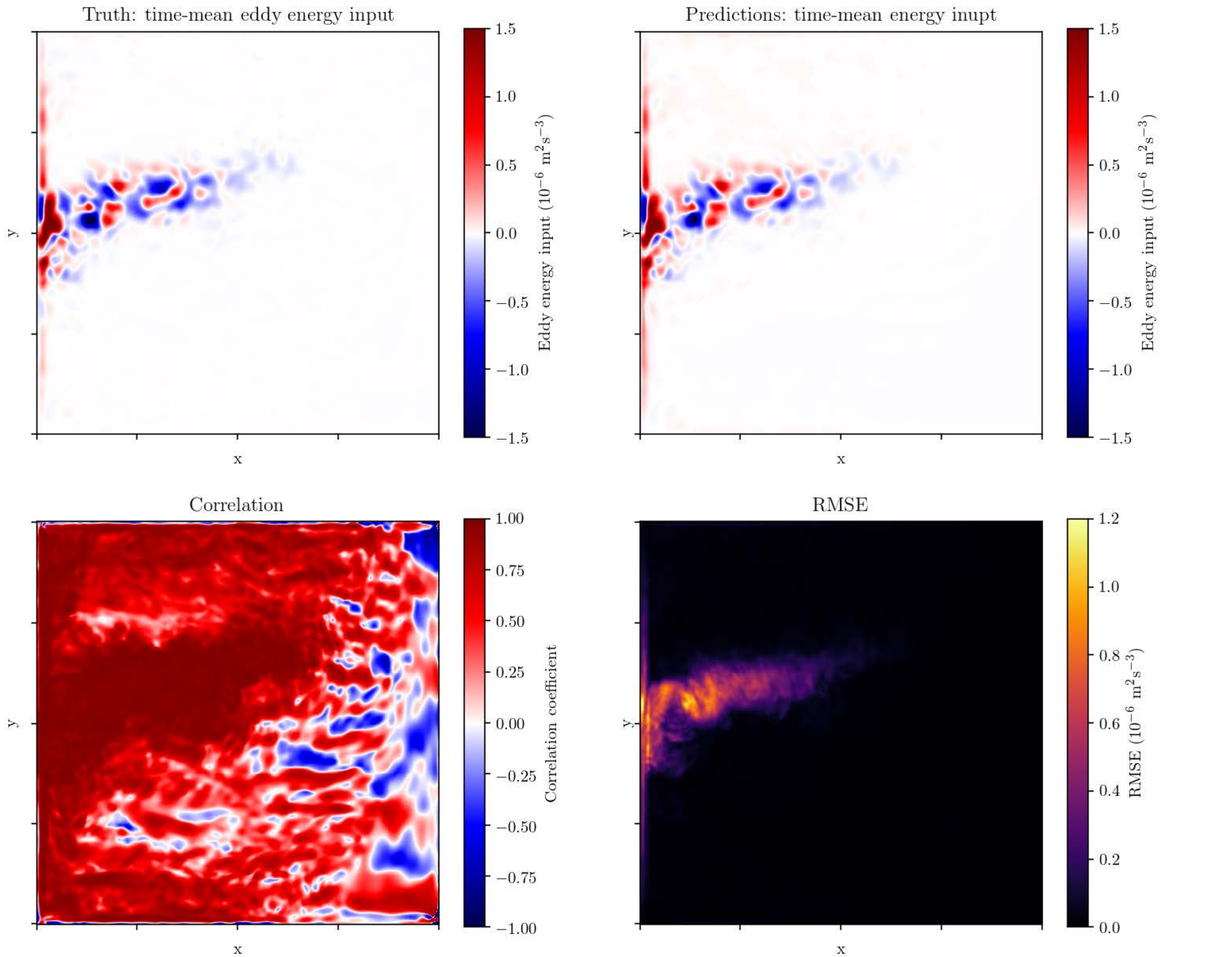
Opportunity: merging traditional thinking (physics constraints, stable implementation) with new avenues from data-driven algorithms (new parametrizations - not just parameters, stochastic physics, estimates of model errors)

Architecture of the Convolutional Neural Network



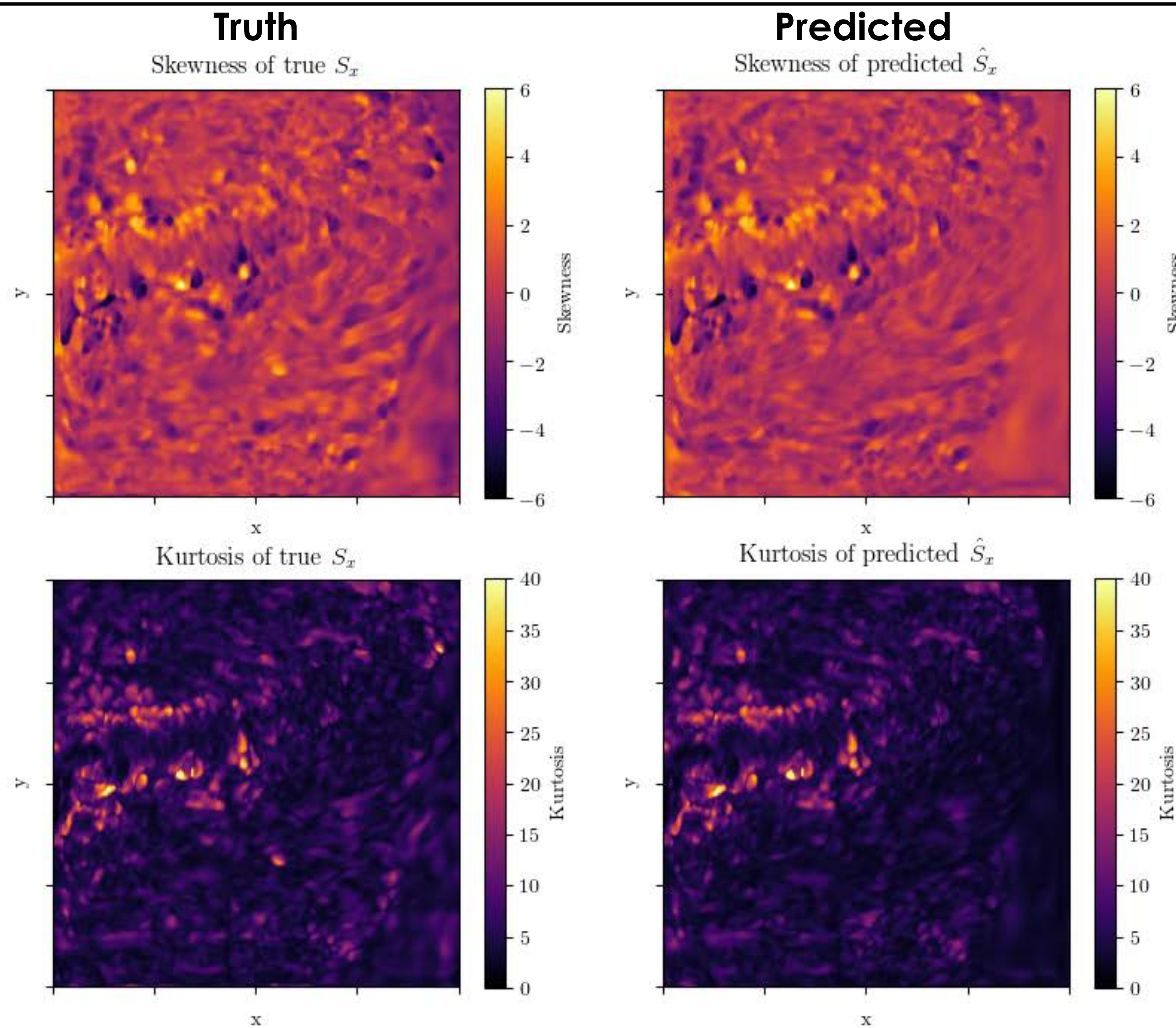
Bolton & Zanna, JAMES, 2019

Eddy Energy

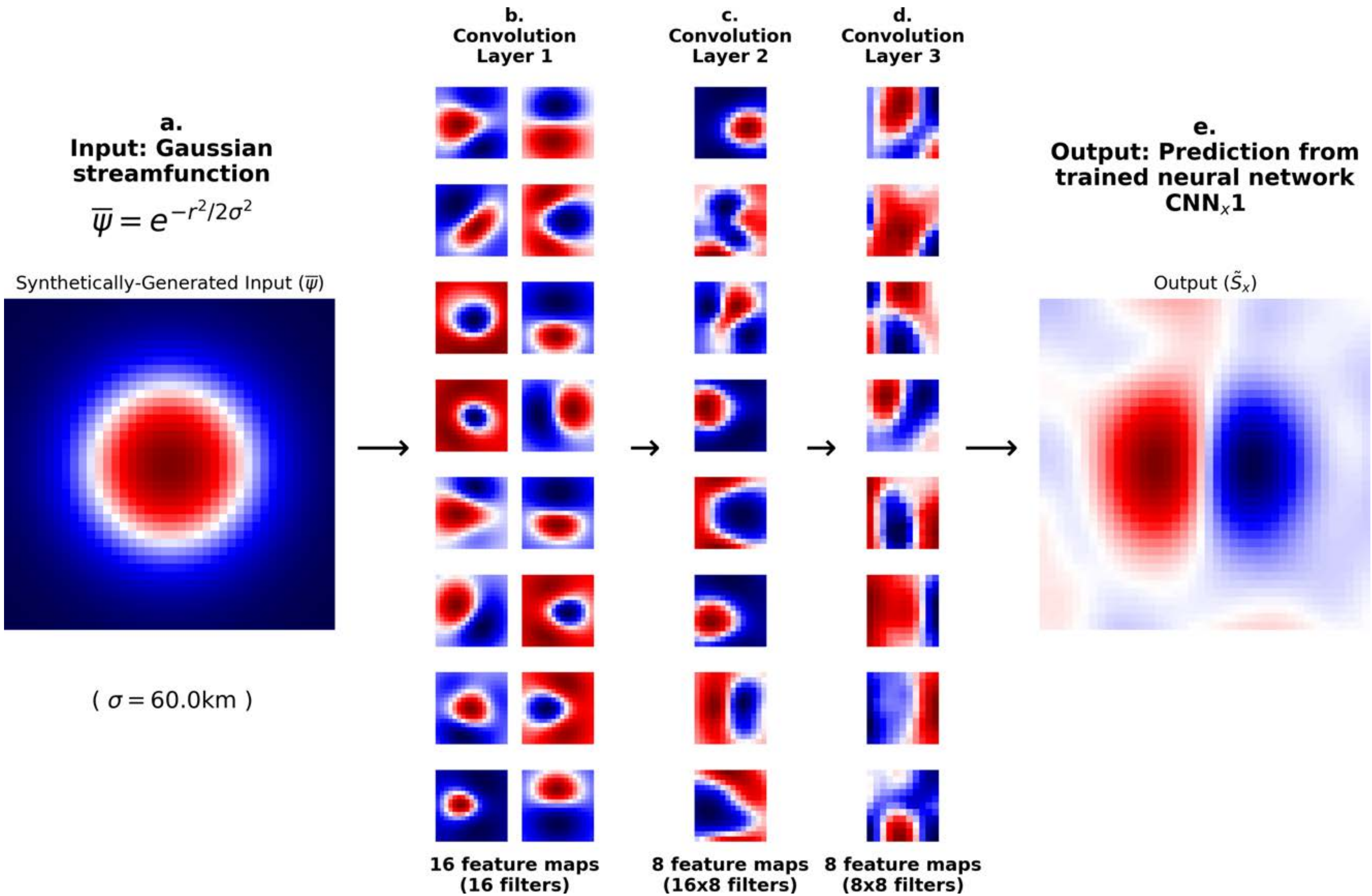


Bolton & Zanna, 2019; Zanna et al 2018, 2019

Higher Order Statistics

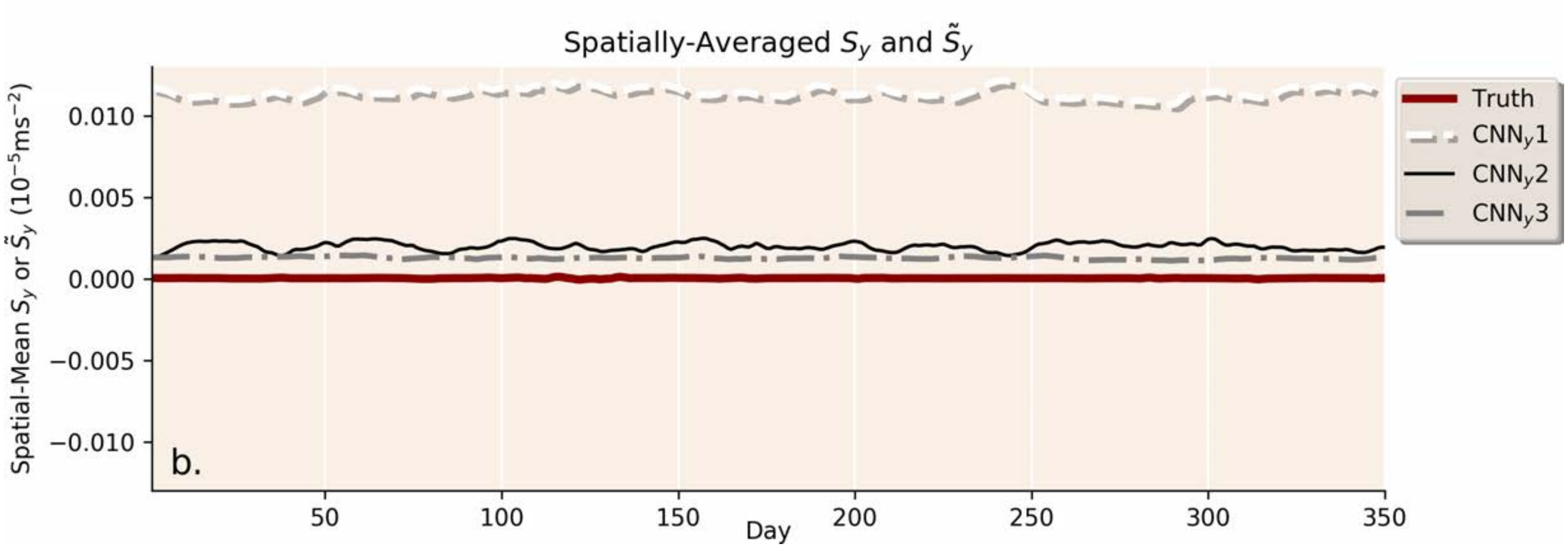


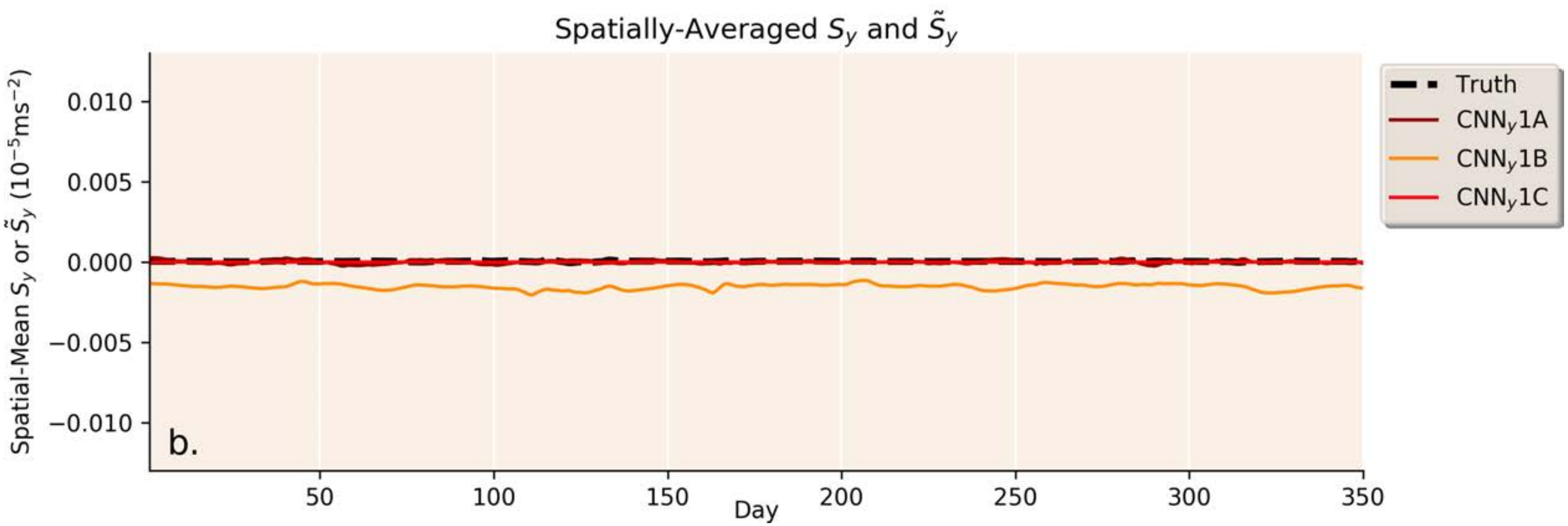
Extracting derivatives



Bolton & Zanna, JAMES, 2019

Momentum Conservation





Bolton & Zanna, JAMES, 2019

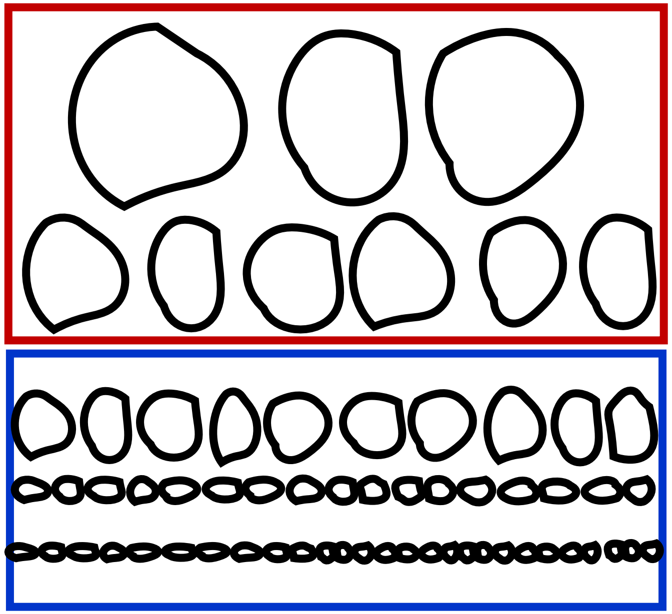
Network Details

Neural Network Data Details	
Data source	Quasi-geostrophic ocean model
Input variable (feature)	Filtered-stream function $\bar{\psi}$
Output variables (targets)	Subfilter momentum forcing S_x, S_y
Training region 1	Western boundary
Training region 2	Eastern boundary
Training region 3	Southern gyre
Number of training samples	5,800 (years 1–9)
Number of validation samples	5,600 (year 10)
Standardization method	Zero mean, unit variance
Neural Network Architecture	
Input size	40×40
Number of convolution layers	3
Number of filters for each convolution layer	16, 16*8, 8*8
Size of filter for each convolution layer	8×8, 4×4, 4×4
Filter stride for each convolution layer	2, 1, 1
Activation function for each convolution layer	SELU, SELU, SELU
Max pooling kernel size	2
Output layer activation function	None/Linear
Output size	40×40
Neural Network Training Parameters	
Loss function	Mean-square error
Optimizer	Adam
Learning rate	0.001
Momentum	0.9
Batch size	16
Training epochs	200

Bolton & Zanna, JAMES, 2019

The Parametrization or Closure Problem

- Including unresolved processes at low computational cost



$\overline{(\ )}$ = slow- /large-scale fluctuations > grid-box size

$(\ )'$ = fast/small-scale (eddy) fluctuations < grid-box size

- E.g., momentum (the same applies to heat, etc)

$$\frac{\partial \overline{u}}{\partial t} + \overline{u}\frac{\partial \overline{u}}{\partial x} + \overline{v}\frac{\partial \overline{u}}{\partial y} - f\overline{v} = -\frac{1}{\rho_0}\frac{\partial \overline{p}}{\partial x} + \frac{\overline{F}_x}{\rho_0} - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y}$$

$$\frac{\partial \overline{u}}{\partial t} + \overline{u}\frac{\partial \overline{u}}{\partial x} + \overline{v}\frac{\partial \overline{u}}{\partial y} - f\overline{v} = -\frac{1}{\rho_0}\frac{\partial \overline{p}}{\partial x} + \frac{\overline{F}_x}{\rho_0} + \boxed{\text{S = Turbulence closure for sub-grid eddy forcing}}$$

Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

Anstey & Zanna (2017)

$$\mathbf{f}(\overline{u},\overline{v}) = \kappa\overline{\nabla}\cdot\left[\begin{pmatrix} -\zeta D & \zeta\tilde{D} \\ \zeta\tilde{D} & \zeta D \end{pmatrix} + \frac{1}{2}\mathbf{I}(\zeta^2 + D^2 + \tilde{D}^2)\right]$$

(scalar)

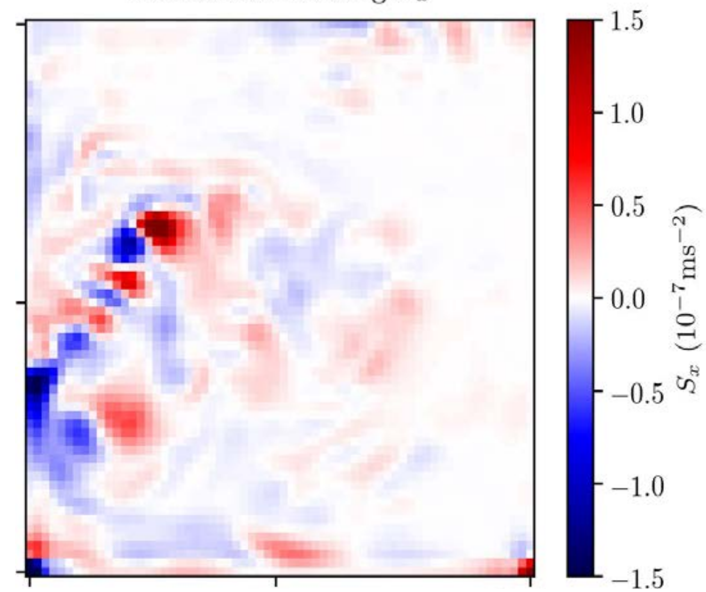
ζ = vorticity

D = shearing deformation

$\tilde{D}$ = stretching deformation

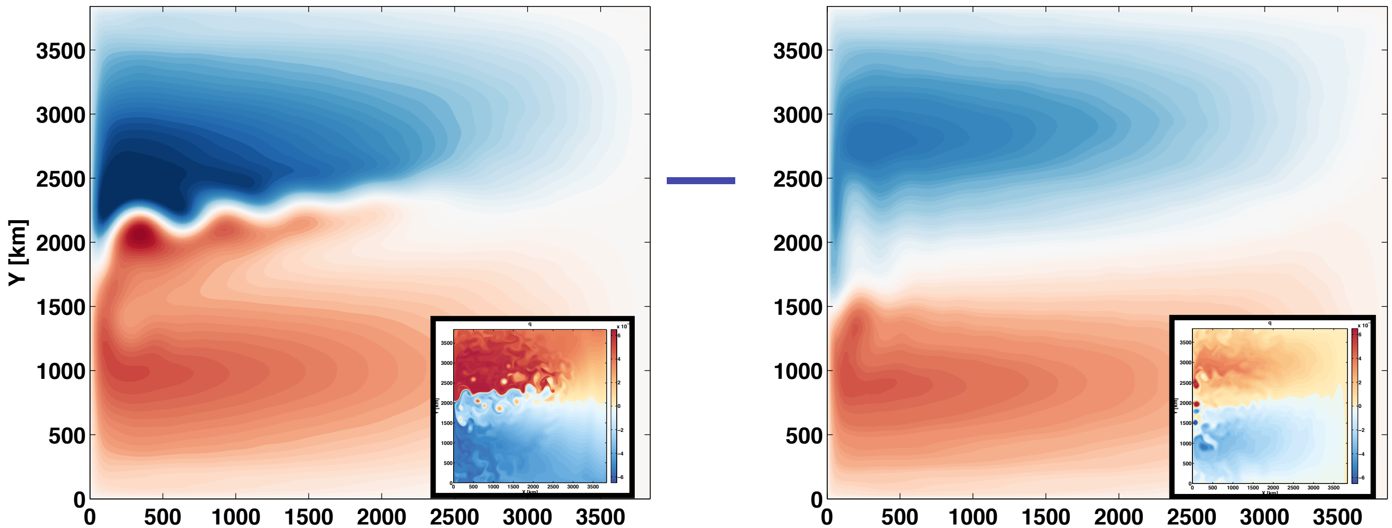
- Captures ~50% of the variance across all 15 vertical levels
- Baroclinic expression = barotropic expression + additional terms
- Symmetric stress tensor & conserves global momentum

Snapshot of eddy momentum forcing S_x



High-resolution simulations & Coarse-Graining/Filtering

Eddy Resolving (7.5km) — **Eddy Permitting (30km)**



- Diagnostics in a baroclinic 3 layers quasi-geostrophic model *(Zanna et al, 2017)*

- Filtering of high-resolution variables: $$\overline{(.)} \propto \int (.)\, e^{-\frac{(\mathbf{x}-\mathbf{x}_0)^2}{2\sigma^2}}\, d\mathbf{x}$$

- Eddy sub-grid momentum forcing: $$\mathbf{S} = (\overline{\mathbf{u}} \cdot \nabla)\overline{\mathbf{u}} - \overline{(\mathbf{u} \cdot \nabla)\mathbf{u}}$$