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

(from Malcolm Roberts)
Towards a data-driven mesoscale eddy parameterisation.

Model resolution

Resolved flow field $\overline{u}$

Unresolved eddy field $u'$

Interaction between fields:

Subgrid forcing

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

resolved flow

Subgrid forcing

PROS
- Clear physical interpretation.
- Computationally cheap to implement.

CONS
- Hard to capture all dynamical and thermodynamic effects.

Traditional route to eddy parameterisations

Interaction between fields:

E.g. Smagorinsky (1963), Leith (1968).
Traditional approach

- Consider physical mechanisms to approximate the bulk effect (e.g. Smagorinsky 1963, Leith 1968)
- Clear physical interpretation
- Computationally cheap to implement
- Parameter from theory, observations or tuning

\[ \frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} = \bar{F} + \bar{D} + \mathbf{S} \]

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

- Extract/diagnose the sub-grid tendency & its statistics → deduce something about the missing physics & its effect

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

Resolved quantities

Input

Output

Eddy momentum forcing

\[ S \approx f(\bar{u}, \bar{v}) \]
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

**Resolved quantities**

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

Eddy momentum forcing $S \approx f(\bar{u}, \bar{v})$

Neural network $\hat{S}_x = f(\overline{\psi}, w)$, trained to minimize loss $L \propto (S_x - \hat{S}_x)^2$. 

**Towards a data-driven mesoscale eddy parameterisation.**

Our previous work using CNNs

- Trained convolutional neural networks in an idealised QG model*.
- Regional dynamics impacted accuracy.
- Generalised very well to higher Reynolds number regimes.

(*Bolton & Zanna, 2019, "Applications of deep learning to ocean data inference...", JAMES)
Generalization to other dynamical regimes

\[ \nu = 200 \text{ m}^2\text{s}^{-2} \]

\[ \tau_0 = 0.8 \text{ Nm}^{-2} \]

\[ \tau_0 = 0.9 \text{ Nm}^{-2} \]
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 data from idealised geometry MITgcm barotropic & baroclinic simulations

\[ S = (\bar{u} \cdot \nabla)\bar{u} - (u \cdot \nabla)u \]
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: \[ \sigma = \nabla \cdot \bar{u} \]

Vorticity: \[ \zeta = \nabla \times \bar{u} \]

Shearing deformation:
\[ D = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \]

Stretching deformation:
\[ \tilde{D} = \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \]

Enforce solution to be divergence of a flux:
\[ f(\bar{u}, \bar{v}) = \nabla \cdot \bar{F} \]
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:

$$ S = \nabla \cdot T $$

$$ T = T(\bar{u}, \bar{v}) $$
Discovered expression using data from barotropic MITgcm

\[
\mathbf{f}(\bar{u}, \bar{v}) = \kappa \nabla \cdot \begin{pmatrix}
\zeta^2 - \zeta D & \zeta \tilde{D} \\
\zeta \tilde{D} & \zeta^2 + \zeta D
\end{pmatrix}
\]

(Scalar)
Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

\[
f(\bar{u}, \bar{v}) = \kappa \nabla \cdot \left( \begin{array}{cc}
\zeta^2 - \zeta D & \zeta \tilde{D} \\
\zeta \tilde{D} & \zeta^2 + \zeta D
\end{array} \right)
\]

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

\[
\begin{align*}
\zeta &= \text{vorticity} \\
\tilde{D} &= \text{shearing deformation} \\
\tilde{D} &= \text{stretching deformation}
\end{align*}
\]
Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

\[ \mathbf{f}(\tilde{u}, \tilde{v}) = \kappa \nabla \cdot \begin{pmatrix} \zeta^2 - \zeta \tilde{D} & \zeta \tilde{D} \\ \zeta \tilde{D} & \zeta^2 + \zeta D \end{pmatrix} \]

(scalar)

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

\[ \begin{align*}
    \zeta & = \text{vorticity} \\
    \tilde{D} & = \text{shearing deformation} \\
    \tilde{D} & = \text{stretching deformation}
\end{align*} \]

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

\[ \nabla \cdot \begin{pmatrix} -\zeta D & \zeta \tilde{D} \\ \zeta \tilde{D} & \zeta D \end{pmatrix} \]

Snapshot of eddy momentum forcing \( S_x \)
Implementation in barotropic model

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

More energetic flow fields

Resolution: 30km
No parameterisation

Resolution: 30km
+ \nabla \cdot \left( \begin{array}{c}
\zeta^2 - \zeta D \\
-\zeta \tilde{D} \\
\zeta^2 + \zeta D
\end{array} \right)

Resolution: 3.75km
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)*
Neural network $\tilde{S}_x = f_x(\overline{\psi}, w_1)$, trained to minimize loss $L \propto (S_x - \tilde{S}_x)^2$. 

Bolton & Zanna, JAMES, 2019
Eddy Energy

Bolton & Zanna, 2019; Zanna et al 2018, 2019
Higher Order Statistics

**Truth**

Skewness of true $S_x$

**Predicted**

Skewness of predicted $\hat{S}_x$

Kurtosis of true $S_x$

Kurtosis of predicted $\hat{S}_x$
Extracting derivatives

Figure 10. The feature maps at each stage of the neural network CNN

(a) Input: Gaussian streamfunction
\[ \bar{\psi} = e^{-r^2/2\sigma^2} \]

(b) Convolution Layer 1

(c) Convolution Layer 2

(d) Convolution Layer 3

(e) Output: Prediction from trained neural network CNN, 1

Synthetically-Generated Input (\(\bar{\psi}\))

Output (\(\tilde{S}_x\))

\(\sigma = 60.0\text{km} \)

16 feature maps (16 filters)

8 feature maps (16x8 filters)

8 feature maps (8x8 filters)

Figure 11. Correlation between the (momentum-form) parameterisation of Mana and Zanna [2014], with coarse-grained sub-filter momentum forcing. The parameterisation was calculated from the coarse-grained potential vorticity, which was calculated from the coarse-grained streamfunction.
Figure 7. Comparing spatial-averages of the neural network predictions with the truth. Panels (a) and (b) compare the spatial-averages of the zonal and meridional components respectively; the spatial-averages of $S_y$ and $\tilde{S}_y$ indicate how the sub-filter momentum forcing impacts the global momentum budget, i.e. the contributions to the spatially-averaged filtered-momentum tendency $D_u/Dt$. All diagnostics are calculated from the validation data.

Figure 9. Comparing spatial-averages of the neural network predictions with the truth. Panels (a) and (b) compare the spatial-averages of the zonal and meridional components respectively. The same as Figure 7, but comparing the neural networks of the momentum-conserving approaches A, B, and C.
## Network Details

### Neural Network Data Details

<table>
<thead>
<tr>
<th>Data source</th>
<th>Quasi-geostrophic ocean model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input variable (feature)</td>
<td>Filtered-stream function $\tilde{\psi}$</td>
</tr>
<tr>
<td>Output variables (targets)</td>
<td>Subfilter momentum forcing $S_x, S_y$</td>
</tr>
<tr>
<td>Training region 1</td>
<td>Western boundary</td>
</tr>
<tr>
<td>Training region 2</td>
<td>Eastern boundary</td>
</tr>
<tr>
<td>Training region 3</td>
<td>Southern gyre</td>
</tr>
<tr>
<td>Number of training samples</td>
<td>5,800 (years 1–9)</td>
</tr>
<tr>
<td>Number of validation samples</td>
<td>5,600 (year 10)</td>
</tr>
<tr>
<td>Standardization method</td>
<td>Zero mean, unit variance</td>
</tr>
</tbody>
</table>

### Neural Network Architecture

<table>
<thead>
<tr>
<th>Input size</th>
<th>$40 \times 40$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of convolution layers</td>
<td>3</td>
</tr>
<tr>
<td>Number of filters for each convolution layer</td>
<td>16, 16<em>8, 8</em>8</td>
</tr>
<tr>
<td>Size of filter for each convolution layer</td>
<td>$8 \times 8, 4 \times 4, 4 \times 4$</td>
</tr>
<tr>
<td>Filter stride for each convolution layer</td>
<td>2, 1, 1</td>
</tr>
<tr>
<td>Activation function for each convolution layer</td>
<td>SELU, SELU, SELU</td>
</tr>
<tr>
<td>Max pooling kernel size</td>
<td>2</td>
</tr>
<tr>
<td>Output layer activation function</td>
<td>None/Linear</td>
</tr>
<tr>
<td>Output size</td>
<td>$40 \times 40$</td>
</tr>
</tbody>
</table>

### Neural Network Training Parameters

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Mean-square error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Momentum</td>
<td>0.9</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
</tr>
<tr>
<td>Training epochs</td>
<td>200</td>
</tr>
</tbody>
</table>
The Parametrization or Closure Problem

- Including unresolved processes at low computational cost

\[ \bar{u} = \text{slow- } /\text{large-scale fluctuations} \]
\[ \bar{u} = \text{grid-box size} \]

\[ (\cdot) = \text{fast/small-scale (eddy) fluctuations} \]
\[ (\cdot)' = \text{grid-box size} \]

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

\[
\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} - f \bar{v} = - \frac{1}{\rho_0} \frac{\partial p}{\partial x} + \frac{F_x}{\rho_0} - \frac{\partial u'u'}{\partial x} - \frac{\partial u'v'}{\partial y}
\]

\[
\frac{\partial \bar{u}}{\partial t} + \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} - f \bar{v} = - \frac{1}{\rho_0} \frac{\partial p}{\partial x} + \frac{F_x}{\rho_0} + \frac{\partial u'u'}{\partial x} + \frac{\partial u'v'}{\partial y}
\]

\[ S = \text{Turbulence closure for sub-grid eddy forcing} \]
Eddy momentum forcing expressions

Discovered expression using data from barotropic MITgcm

Anstey & Zanna (2017)

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

- Captures \(~50\%\) of the variance across all 15 vertical levels
- Baroclinic expression = barotropic expression + additional terms
- Symmetric stress tensor & conserves global momentum
Diagnostics in a baroclinic 3 layers quasi-geostrophic model (Zanna et al, 2017)

Filtering of high-resolution variables:
\[
\overline{(.)} \propto \int e^{-\frac{(x-x_0)^2}{2\sigma^2}} \, dx
\]

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