Towards Physically-Consistent, Data-driven, and Interpretable Parametrizations of Convection

Presenter: Tom Beucler – UCI & Columbia

Advisors: Michael Pritchard (UCI) & Pierre Gentine (Columbia)

Collaborators: Ankitesh Gupta, Jordan Ott, Pierre Baldi, Liran Peng (UCI), Stephan Rasp (TUM), Noah Brenowitz, Chris Bretherton (UW)

Image source: NASA (Stratocumulus deck above the northwestern Pacific Ocean, 2013)
Motivation 1: Largest uncertainties in climate projections come from clouds

Eq. Temperature Response to CO₂ doubling

Source: Gentine et al. (Submitted), adapted from Meehl et al. (In Review)
**Motivation 1: Largest uncertainties in climate projections come from clouds**

**Eq. Temperature Response to CO₂ doubling**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Effective Climate Sensitivity</strong></td>
<td>4.2</td>
<td>4.3</td>
<td>4.1</td>
<td>4.2</td>
<td>4.0</td>
<td>4.1</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Cloud shortwave & longwave feedbacks are top contributors to spread

*Source: Gentine et al. (Submitted), adapted from Meehl et al. (In Review), Zelinka et al. (2020)*
Motivation 2: 100-year Climate Simulations unable to resolve low clouds before 2050

Source: Schneider et al., 2017
Motivation 2: 100-year Climate Simulations unable to resolve low clouds before 2050

Source: Schneider et al., 2017
Motivation 2: 100-year Climate Simulations unable to resolve low clouds before 2050

Inside each box:
Parametrization represents effects of fine-scale turbulence on coarse scale

Source: Gentine et al. (Submitted)
Motivation 3: Data-driven parametrizations accurately mimic subgrid-scale thermodynamics (offline)

See: Beucler et al. (2019), Gentine et al. (2018)
Motivation 3: Accurately enough to correct convective biases in climate models for $\sim$10\% CPU cost.
Problem: First crude attempts show that Machine-learning parametrizations...

1. Are hard to interpret and trust, impinging their operational use. E.g. Lead to unexpected instabilities when coupled to fluid dynamics.

See: Brenowitz et al. (2020)
Problem: First crude attempts show that Machine-learning parametrizations...

1. Are hard to interpret and trust, impinging their operational use

E.g. Lead to unexpected instabilities when coupled to fluid dynamics

See: Brenowitz et al. (2020)
Problem: First crude attempts show that Machine-learning parametrizations…

1. Are hard to interpret and trust, impinging their operational use
2. Violate physical constraints (Conservation laws, Positive definition)

See: Bencler et al. (2019, 2020)
Problem: First crude attempts show that Machine-learning parametrizations...

1. Are hard to interpret and trust, impinging their operational use
2. Violate physical constraints (Conservation laws, Positive definition)
   E.g. Generate spurious O(10W/m²) or negative precipitation
Problem: First crude attempts show that Machine-learning parametrizations…

1. Are hard to interpret and trust, impinging their operational use
2. Violate physical constraints (Conservation laws, Positive definition)
3. Make large errors when evaluated outside of training set

See: Beucler et al. (2020)
Problem: First crude attempts show that Machine-learning parametrizations…

1. Are hard to interpret and trust, impinging their operational use
2. Violate physical constraints (Conservation laws, Positive definition)
3. Make large errors when evaluated outside of training set

E.g. Double convective heating if evaluated in warmer climate (+4K)

See: Beucler et al. (2020)
Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?

Here shown for subgrid parametrization convection but broadly applicable to data-driven models of physical processes.
Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?
1) Global Cloud-Resolving Model (UW)

See: Brenowitz et al. (2018, 2019, 2020), Movie source: Noah Brenowitz
1) Global Cloud-Resolving Model (UW)

Challenge: No clear “truth”/ “target” → Requires coarse-graining data

See: Brenowitz et al. (2018), Slide source: Noah Brenowitz
1) Global Cloud-Resolving Model (UW)

Challenge: No clear “truth”/ “target” → Requires coarse-graining data

Challenge 2: No objective method to coarse-grain
1) Global Cloud-Resolving Model (UW)

Challenge: No clear “truth”/“target” → Requires coarse-graining data

Challenge 2: No objective method to coarse-grain

\[
\left( \frac{\partial \overline{\phi}}{\partial t} \right)_{\text{vertical advection}} = \left( w' \frac{\partial \phi'}{\partial z} - \overline{w} \times \frac{\partial \phi}{\partial z} \right)
\]

See: Brenowitz et al. (2018), Yuval et al. (2020), Slide source: Noah Brenowitz
2) “Truth”: Super-Parametrized CAM

Exterior Model:
Does not resolve convection
$\Delta t = 30\text{m}, \Delta x = 2^\circ$

Interior Model:
Resolves response convection
$\Delta t = 20\text{s}, \Delta x = 4\text{km}$
2) “Target”: Emulate Super-Parametrization

Setup: SPCAM3 in **aqua-planet** configuration with **fixed SST**

*See: Collins et al. (2006) Khairoutdinov et al. (2005), Rasp et al. (2018)*
Neural Network maps large-scale climate to how convection vert. redistributes energy

- Specific humidity (kg/kg, 30 lev)
- Temperature (K, 30 lev)
- Surf. Pressure (Pa)
- Sol. Insol. (W/m²)
- SHF (W/m²)
- LHF (W/m²)
- Spec. hum. tend. (kg/kg/s, 30 lev)
- Temperature tend. (K/s, 30 lev)
- Net SW t (W/m²)
- Net SW b (W/m²)
- Net LW t (W/m²)
- Net LW b (W/m²)

Fully connected network

# layers & nodes decided via e.g. formal hyperparameter tuning
Loss = Mass-weight. MSE (W/m²)
Adam optimizer
7 layers of 128 → ~100k param.
Trained for 15-20 epochs
Train. = yr1 (42,369,024 samples)
Valid. = yr2 (42,369,024 samples)

See: SHERPA (Github), Hertel et al. (2018), Gentine et al. (2018), Rasp et al. (2018), Beucler et al. (2019)
Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?
Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?
Problem: For climate modeling, we need interpretable parametrizations

Why did you predict 42 for this data point?

*awkward silence*

Source: Interpretable Machine Learning, C. Molnar (2019)
Tailor NN interpretability techniques to parametrization task

Tailor NN interpretability techniques to parametrization task

1) Partial Dependence Plot

“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Neural-network = Non-linear Mapping: \( x \xrightarrow{\text{NN}} y = f(x) \)

Isolate inputs of interest:

\[
\begin{bmatrix}
  x_{\text{PDP}} \\
  x_{\text{Other}}
\end{bmatrix}^T
\]

Partial dependence fx:

\( x_{\text{PDP}} \xrightarrow{\text{E}} \mathbb{E} \left\{ f \begin{bmatrix} x_{\text{PDP}} & x_{\text{Other}} \end{bmatrix}^T | x_{\text{PDP}} \right\} \)

Estimated by averaging over chunks of data with similar inp. of interest

See: Molnar et al. (2018), Friedman (2001)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Input 1:
Mid-tropospheric Moisture (QM)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

**Input 1:**
Mid-tropospheric Moisture (QM)

**Input 2:**
Lower-tropospheric Stability (LTS)

*Image source: NOAA Photo library (fly00890), See: Wood and Bretherton (2006)*
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 7.4kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 9.5kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot

“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  
QM = 11.6kg/m²

See: Brenowitz et al. (2020)
1) **Partial Dependence Plot**

“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed $LTS = 11.2K$ \hspace{1cm} $QM = 13.7\text{kg/m}^2$

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 15.8kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K   QM = 17.9kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 20.0kg/m²

Subgrid Moistening [W m⁻²]  Subgrid Heating [W m⁻²]

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 22.1kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K    QM = 24.2kg/m²
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 26.3kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  \[\text{QM} = 28.4\text{kg/m}^2\]

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 30.5kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 32.6kg/m²

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed LTS = 11.2K  QM = 34.7kg/m^2

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7 kg/m²  
LTS = 7.4 K

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7kg/m²    LTS = 8.3K

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7kg/m²   LTS = 9.1K

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7kg/m²   LTS = 9.9K

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7kg/m²  LTS = 10.8K

See: Brenowitz et al. (2020)
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7 kg/m²   LTS = 11.6 K
1) Partial Dependence Plot
“Shows the marginal effect of 1 or 2 inputs on the predicted outputs”

Fixed QM = 33.7kg/m²  LTS = 12.5K

See: Brenowitz et al. (2020)
Tailor NN interpretability techniques to parametrization task

2) Saliency Map
Gradient of the NN

Calculate the NN’s Jacobian via automatic differentiation:

\[
J \overset{\text{def}}{=} \left( \frac{\partial \text{Output}}{\partial \text{Input}} \right)_{\text{Input}_0}
\]

Image source: flashtorch (Github) See: Paszke et al. (2017), Springenberg et al. (2015)
Jacobian = Linear Response Function

\[
\left( \frac{\partial \text{Output}}{\partial \text{Input}} \right) = \frac{\partial (\text{Convective Moistening})}{\partial (\text{Moisture})} \quad [1/\text{day}]
\]

Jacobian = Linear Response Function

\[
\left( \frac{\partial \text{Output}}{\partial \text{Input}} \right) = \frac{\partial (\text{Convective Moistening})}{\partial (\text{Moisture})} \quad [\text{1/day}]
\]

Local anomalies are removed and redistributed in the lower atmosphere.

Jacobian = Linear Response Function

Stable NN

Unstable NN

Coupling Linear Response Function to Gravity Waves gives Stability Diagram offline

See: Kuang (2018), Brenowitz et al. (2020)

Spurious unstable propagating modes
Stability diagram helped stabilize NNs offline

Super Parametrized

“Regularize” Inputs by adding Gaussian noise

Global Cloud-Resolv.

Remove upper-atmos. Inputs
Both stabilized NN ran without crashing for 1 month+ when coupled to climate models.
We can make data-driven parametrizations interpretable by tailoring existing interpretability tools to parametrization. **BUT** Generate spurious $O(10W/m^2)$ or negative precipitation.

See: Brenowitz et al. (2020) "Interpreting and Stabilizing Machine-learning Parametrizations of Convection." 
We can make data-driven parametrizations **interpretable** by tailoring existing interpretability tools to parametrization

**BUT** Generate spurious $O(10W/m^2)$ or negative precipitation
Adapting NN's Architecture/Loss Fx

Context: Conservation of mass, energy and radiation

First step = Add all terms of conservation laws to NN’s inputs/outputs

Loss: Introduce a penalty for violating conservation (~Lagrange mult.):

\[ \text{Loss} = \alpha (\text{Squared residual from conservation laws}) + (1 - \alpha) (\text{Mean squared error}) \]

Architecture: Constraints lay. to enforce cons. laws to machine precision

\[
\begin{bmatrix}
 x_1 \\
 \vdots \\
 x_m
\end{bmatrix}
\xrightarrow{\text{Standard NN (Optimizable)}}
\begin{bmatrix}
 y_1 \\
 \vdots \\
 y_{p-n}
\end{bmatrix}
\]
Loss: Trade-off between **performance** and **physical constraints**

\[
\text{Loss} = (1 - \alpha) \text{[Mean squared error]} + \alpha \text{[Squared residual from conservation laws]}
\]
Loss: Trade-off between performance and physical constraints

Loss = (1 - \(\alpha\)) (Mean squared error) + \(\alpha\) (Squared residual from conservation laws)
Loss: Trade-off between **performance** and **physical constraints**

\[
\text{Loss} = (1 - \alpha) \text{(Mean squared error)} + \alpha \text{(Squared residual from conservation laws)}
\]
Loss: Trade-off between **performance** and **physical constraints**

\[
\text{Loss} = (1 - \alpha) \text{Mean squared error} + \alpha \text{Squared residual from conservation laws}
\]
Loss: Trade-off between performance and physical constraints

\[
\text{Loss} = (1 - \alpha) \text{(Mean squared error)} + \alpha \text{(Squared residual from conservation laws)}
\]
Loss: Trade-off between performance and physical constraints

Loss = \((1 - \alpha)\) \(\text{Mean squared error}\) + \(\alpha\) \(\text{Squared residual from conservation laws}\)
Loss: Trade-off between performance and physical constraints

Architecture: Constraints enforced & competitive performance
Loss: Trade-off between performance and physical constraints

Architecture: Nonlinear Constraints enforced & competitive perf. & Biases mitigated via loss function
Problem: Even when physically constrained, NNs fail to generalize
**Problem:** Even when physically constrained, NNs fail to generalize

Daily-mean Tropical prediction in reference climate

See: Beucler et al. (2019)
**Problem:** Even when physically constrained, NNs fail to generalize

*Daily-mean Tropical prediction in reference climate*

*See: Beucler et al. (2019)*
**Problem:** Even when physically constrained, NNs fail to generalize

Daily-mean Tropical prediction in (+4K) warming experiment

See: Beucler et al. (2019)
We can enforce **physical constraints** in NN parametrizations of convection by changing the architecture (strict) or the loss fx (soft) **BUT** Make large error when evaluated outside of training conditions

We can enforce physical constraints in NN parametrizations of convection by changing the architecture (strict) or the loss fx (soft) BUT Make large error when evaluated outside of training conditions
Generalization Experiment:
Idea = Break the model
Generalization Experiment: Uniform +8K warming

Training and Validation on cold aquaplanet simulation (Cold, -4K)

Test on warm aquaplanet simulation (Warm, +4K)

Images: Rashevskyi Viacheslav, Sebastien Decoret
Generalization Experiment:
Uniform +8K warming

Surface Temperature (K, fixed)
Generalization Experiment:
Uniform +8K warming

Surface Temperature (K, fixed)

- Cold
- Warm (+8K)

Latitude (deg)
Generalization Experiment:
Uniform +8K warming
Generalization Experiment: Uniform +8K warming

Trained on cold climate

Tested out-of-sample
Physically rescale the data to convert extrapolation into interpolation

Goal: Uncover climate-invariant mapping from climate to convection

<table>
<thead>
<tr>
<th>Specific humidity ($p$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature ($p$)</td>
</tr>
<tr>
<td>Surface Pressure</td>
</tr>
<tr>
<td>Solar Insolation</td>
</tr>
<tr>
<td>Latent Heat Flux</td>
</tr>
<tr>
<td>Sensible Heat Flux</td>
</tr>
</tbody>
</table>

$\rightarrow$ $\nabla$ $\rightarrow$

<table>
<thead>
<tr>
<th>Subgrid moistening ($p$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subgrid heating ($p$)</td>
</tr>
<tr>
<td>Radiative fluxes</td>
</tr>
</tbody>
</table>

**Brute Force: Not Climate-Invariant**
Physically rescale the data to convert extrapolation into interpolation

Goal: Uncover **climate-invariant** mapping from climate to convection

Steps:
1. Data analysis to find climate-invariant mapping
2. Train NN on physically-rescaled data
Algorithms: Custom Data Generators & Custom Layers

- Only one training/validation/test data despite multiple rescalings
- Build NNs using different physical rescalings (trial & error)
- Keep the rescalings that yield the best generalization
Clausius Clapeyron implies exponential scaling of moisture with temperature.
Clausius Clapeyron implies exponential scaling of moisture with temperature.
Clausius Clapeyron implies exponential scaling of moisture with temperature.

Hard to extrapolate.
**Problem:** NNs fail to generalize to unseen climates

**Daily-mean Tropical prediction in cold climate**

![Graph showing daily-mean Tropical prediction in cold climate](image)
Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in cold climate
Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in warm climate
Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in warm climate
Specific humidity ($z$) = Relative humidity ($z$) = \( \frac{\text{Partial water vapor pressure} (z)}{\text{Saturation water vapor pressure} (T, \rho)} \)

Extrapolation

Interpolation

**log10 (Histogram)**

- Blue: Cold
- Red: Warm

**log10 (Histogram)**

- Blue: Cold
- Red: Warm

Specific humidity (g/kg) vs. Relative humidity (%)
Generalization improves dramatically!

\[
\text{Specific humidity (z)} = \frac{\text{Partial water vapor pressure (z)}}{\text{Saturation water vapor pressure}(T, \rho)}
\]
Temperature out-of-sample in lower atmosphere
Can we similarly normalize Temperature?
Temperature = Temperature - (Near-surface Temperature)

Extrapolation

Interpolation

Histogram

Temperature (K)

Cold
Warm

Histogram

Temp. minus near-surface Temp. (K)

Cold
Warm
Temperature = Temperature - (Near-surface Temperature)

Generalization improved in lower atmosphere
\[ \widetilde{\text{Temperature}} = \text{Temperature} - (\text{Near} \,-\, \text{surface Temperature}) \]

**BUT**

Problem with the sign of heating
Latent Heat Flux = \frac{\text{Saturation specific humidity} (T_s, \rho_s)}{\text{Extrapolation}}

\text{Interpolation}

\text{log10 Histogram}

Latent heat flux (W m}^{-2} \begin{cases} \text{Cold} & \text{Cold} \\ \text{Warm} & \text{Warm} \end{cases}

LHF scaled by near-surface saturation (W kg}^{-1} \begin{cases} \text{Cold} & \text{Cold} \\ \text{Warm} & \text{Warm} \end{cases}
Latent Heat Flux = \frac{Latent Heat Flux}{Saturation specific humidity (T_s, p_s)}

Sign of convective heating mostly corrected!
Latent Heat Flux = \frac{\text{Saturation specific humidity} (T_s, p_s)}{\text{Scaled LHF}}

BUT

NN does not capture upwards shift with warming
Sign of improvements in lower atmosphere

\[
\text{Pressure} = \frac{(\text{Surface Pressure}) - \text{Pressure}}{(\text{Surface Pressure}) - (\text{Tropopause Pressure})}
\]
We can make data-driven parametrizations climate-invariant by rescaling the data so as to transform an extrapolation into interpolation.

Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?
We can make data-driven parametrizations:

1) **Interpretable**: Tailor existing interpretability tools to parametrization
2) **Physically-constrained**: Adapt NN’s architecture or loss function
3) **Climate-invariant**: Phys. rescale data (extrapolation) → (interpolation)

1) Brenowitz et al. (2020, 2003.06549)
2) Beucler et al. (2019, 1909.00912)
3) Beucler et al. (2020, 2002.08525)
Outlook

- Rescaled vertical coordinate makes NN applicable across models

- Encouraging generalization results in real geography → observations

- Same NN working for dif. simul. → Test bed for transfer learning

- Stay tuned for coupled simulations with new NNs!
We can make data-driven parametrizations:

1) **Interpretable**: Tailor existing interpretability tools to parametrization
2) **Physically-constrained**: Adapt NN’s architecture or loss function
3) **Climate-invariant**: Phys. rescale data (extrapolation)→(interpolation)

1) Brenowitz et al. (2020, 2003.06549)
2) Beucler et al. (2019, 1909.00912)
3) Beucler et al. (2020, 2002.08525)