ESA-ECMWF Workshop 2021

∂³AWN
∂ata-∂riven
Atmospheric & Water
∂yNamics

ATMOSPHERIC PHYSICS-GUIDED MACHINE LEARNING

Tom Beucler
University of Lausanne

November 16th, 2021

Atmospheric Science & Machine Learning at UNIL

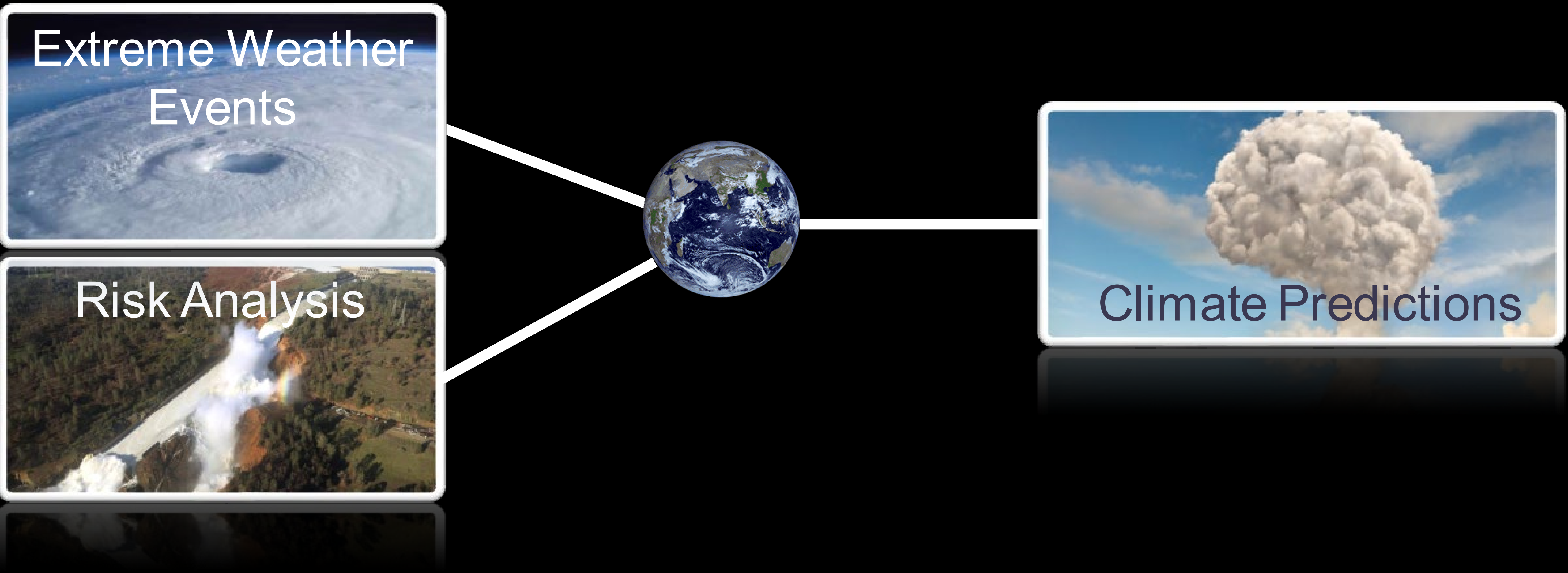


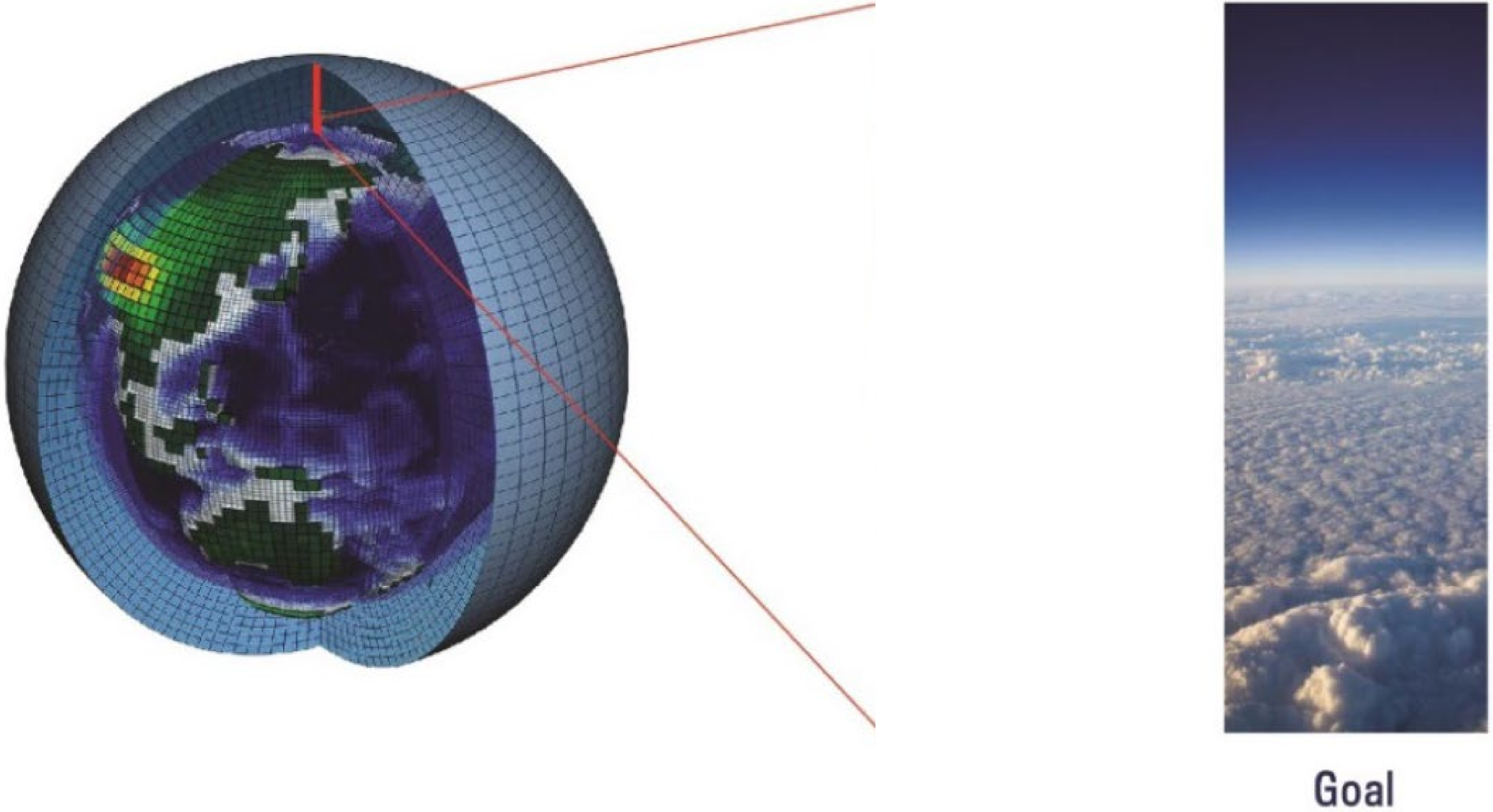
Image Sources: Adhithya Sandeep, John Lund, Cal. Dep. Wat. Res.

ML for Climate Modeling

How to best combine ML & physical knowledge?

Towards Data-Driven and Physically-Consistent Models of **Atmospheric Convection**

<u>Motivation 1</u>: Largest uncertainties in climate projections from clouds



<u>Source:</u> *Zelinka et al. (2020), Meehl et al. (In Review), Gentine, Eyring & Beucler (2020)*

<u>Motivation 1</u>: Largest uncertainties in climate projections from clouds

<u>Motivation 2</u>: Global cloud-resolving models can resolve convection & clouds at ~1km, but only for short period (1 year)

<u>Source:</u> Stevens et al. (2019), Sato et al. (2009), SAM: Khairoutdinov and Randall (2003), Lee and Khairoutdinov

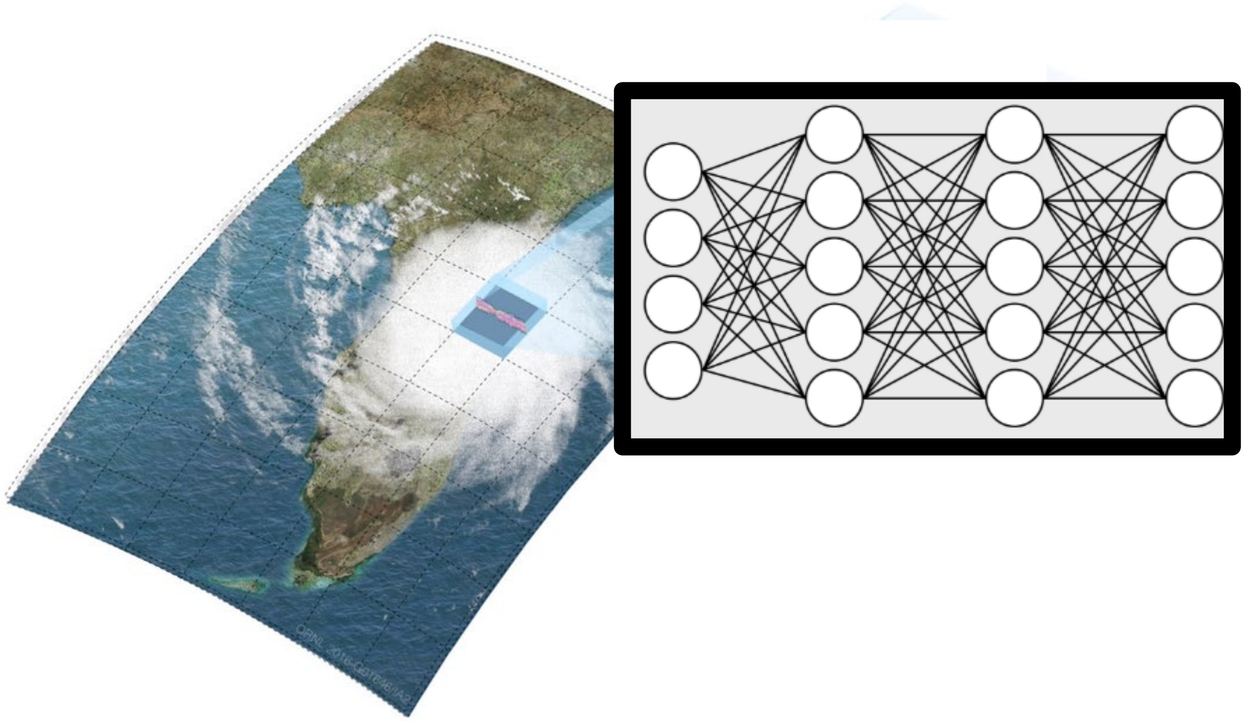
Motivation 1: Largest uncertainties in climate projections from clouds

Motivation 2: Global cloud-resolving models can resolve convection & clouds at ~1km, but only for short period (1 year)

Motivation 3: ML can accurately mimic ~1km convective processes

See: Rasp et al. (2018), Brenowitz et al. (2018,2019), Gentine et al. (2018), Yuval et al. (2020), Krasnopolsky et al.

ML of Subgrid-Scale Thermodynamics



Neural Network:
100 times faster

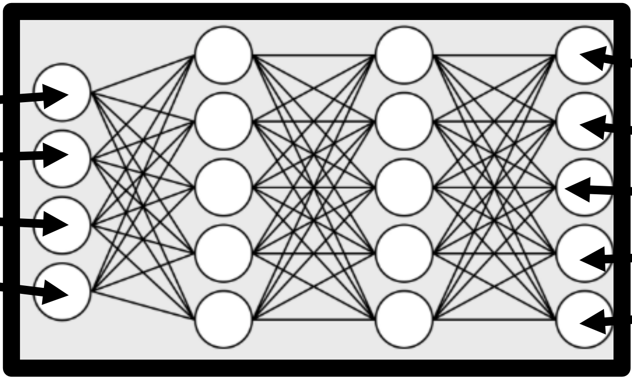
Setup : Super-Parameterized climate model with prescribed surface temp.

Year 1 for training (42M samples), Year 2 for validation/test

Image source: e3sm.org, Model source: Khairoutdinov et al. (2004)

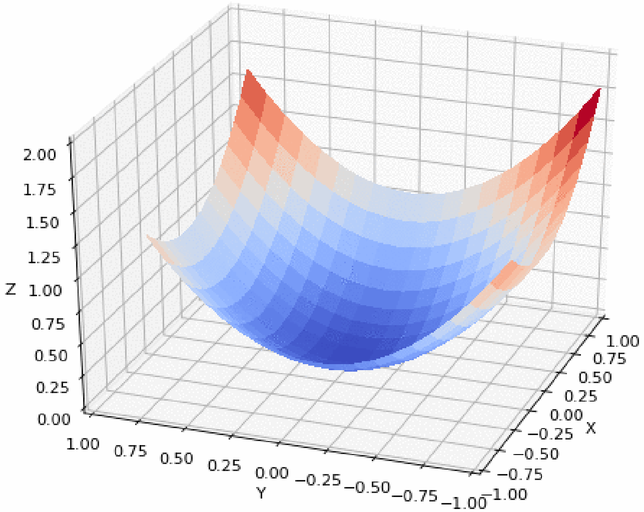
Neural Network = Non-linear regression tool

High-dimensional
Inputs **x**



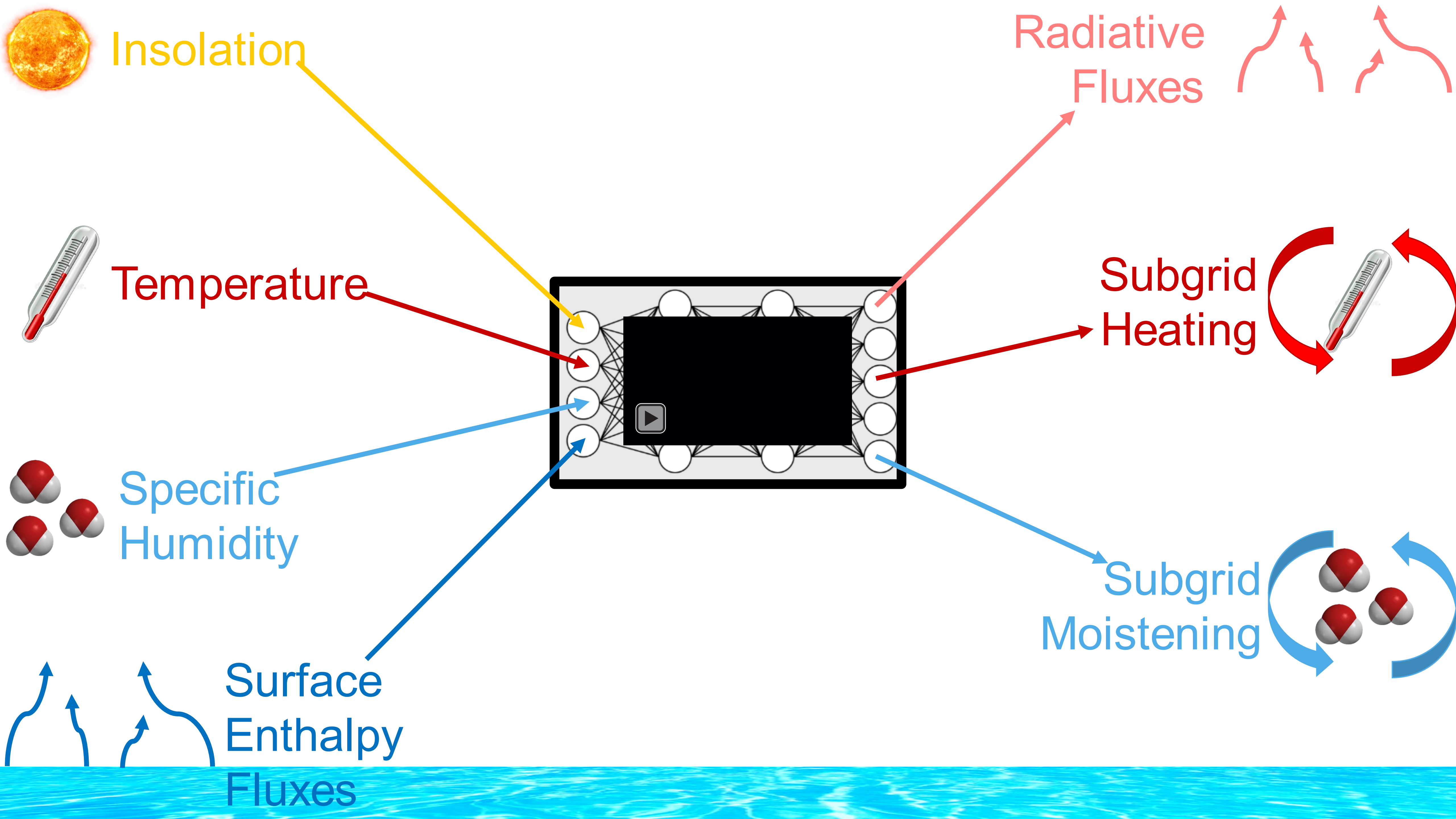
High-dimensional
Outputs **y**

$$\boldsymbol{x} \mapsto \boldsymbol{y}$$



$$\min \ \text{Loss function}\left(\boldsymbol{y}_{\mathbf{Predicted}}, \boldsymbol{y}_{\mathbf{Truth}}\right)$$

<u>Image source</u>: Kathuria (Paperspace)


Insolation
Radiative Fluxes
Temperature
Subgrid Heating
Specific Humidity
Subgrid Moistening
Surface Enthalpy Fluxes

Truth
Super-
param.
simulation

Prediction
NN
(offline)

Source: Mooers, Pritchard, Beucler et al. (2021)

See: Rasp et al. (2018), Brenowitz et al. (2018,2019), Gentine et al. (2018), Yuval et al. (2020), Krasnopolsky et al.

Can we eliminate physics entirely?

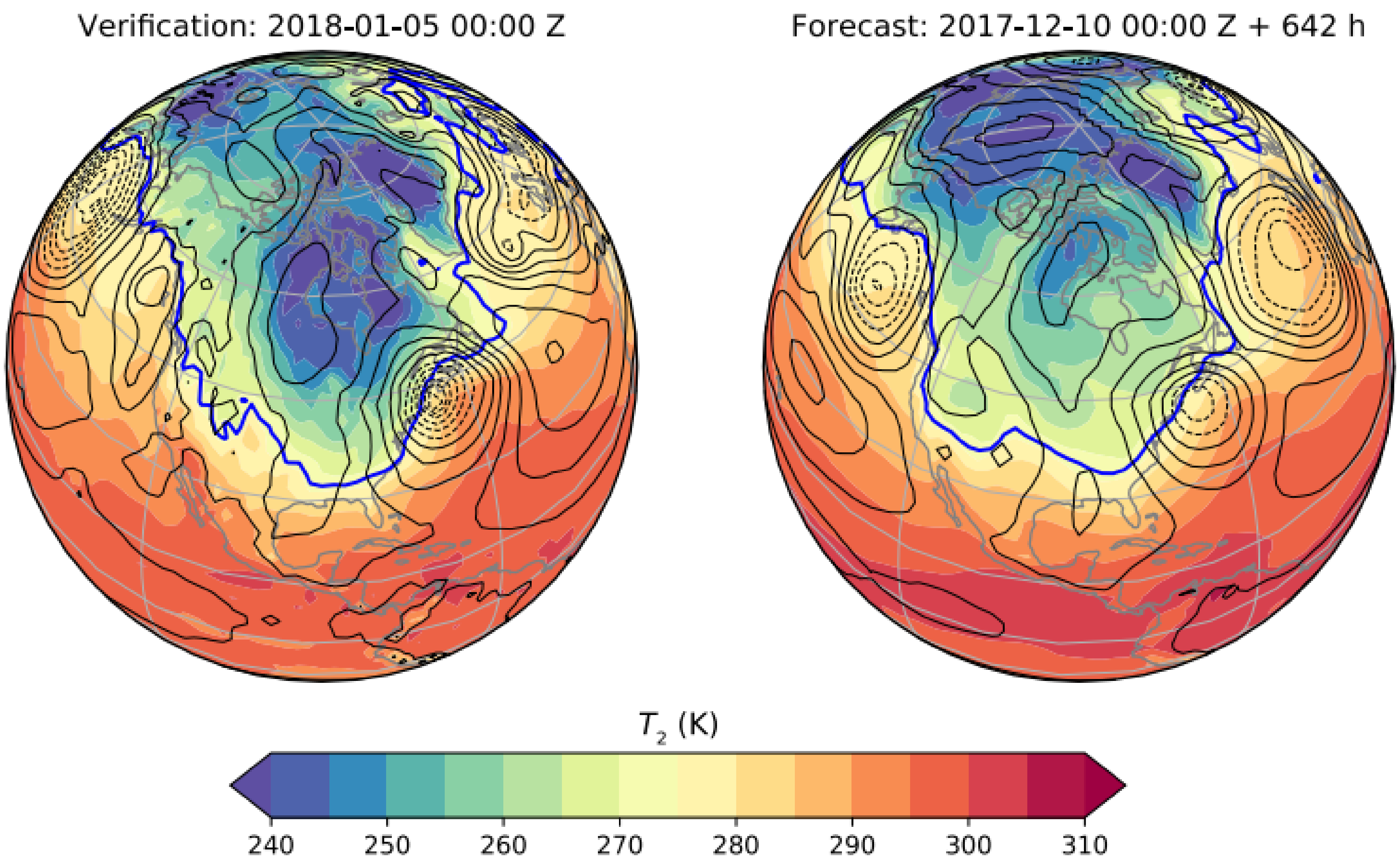


Image Source: Weyn et al. (2020), See also: Rasp et al. (2020)

Can we eliminate physics entirely?

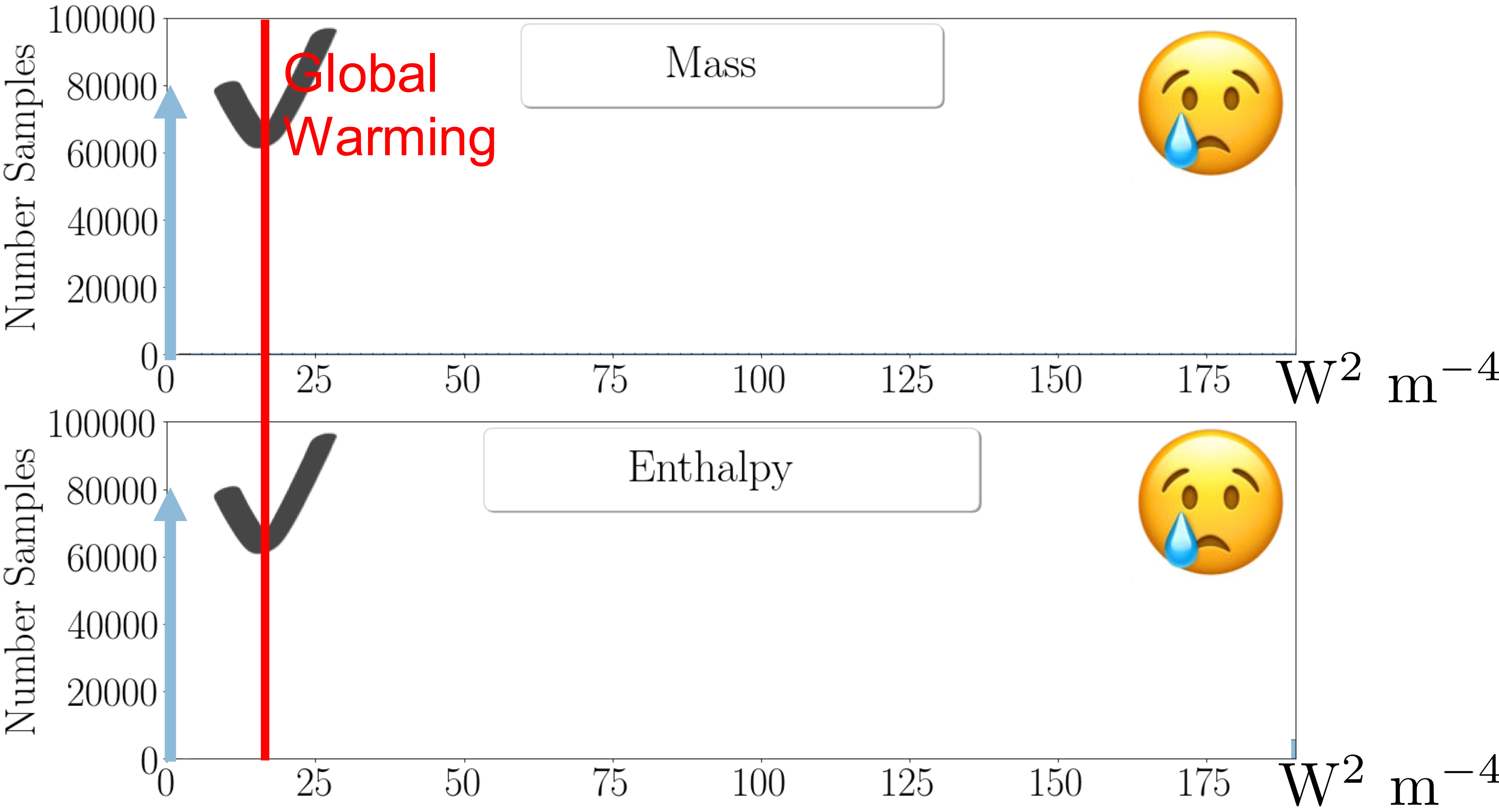
Maybe for meteorology
Not for climate

Problem 1: ML algorithms violate conservation laws

Problem 2: ML parametrization hard to interpret/trust

Problem 3: ML algorithms fail to generalize

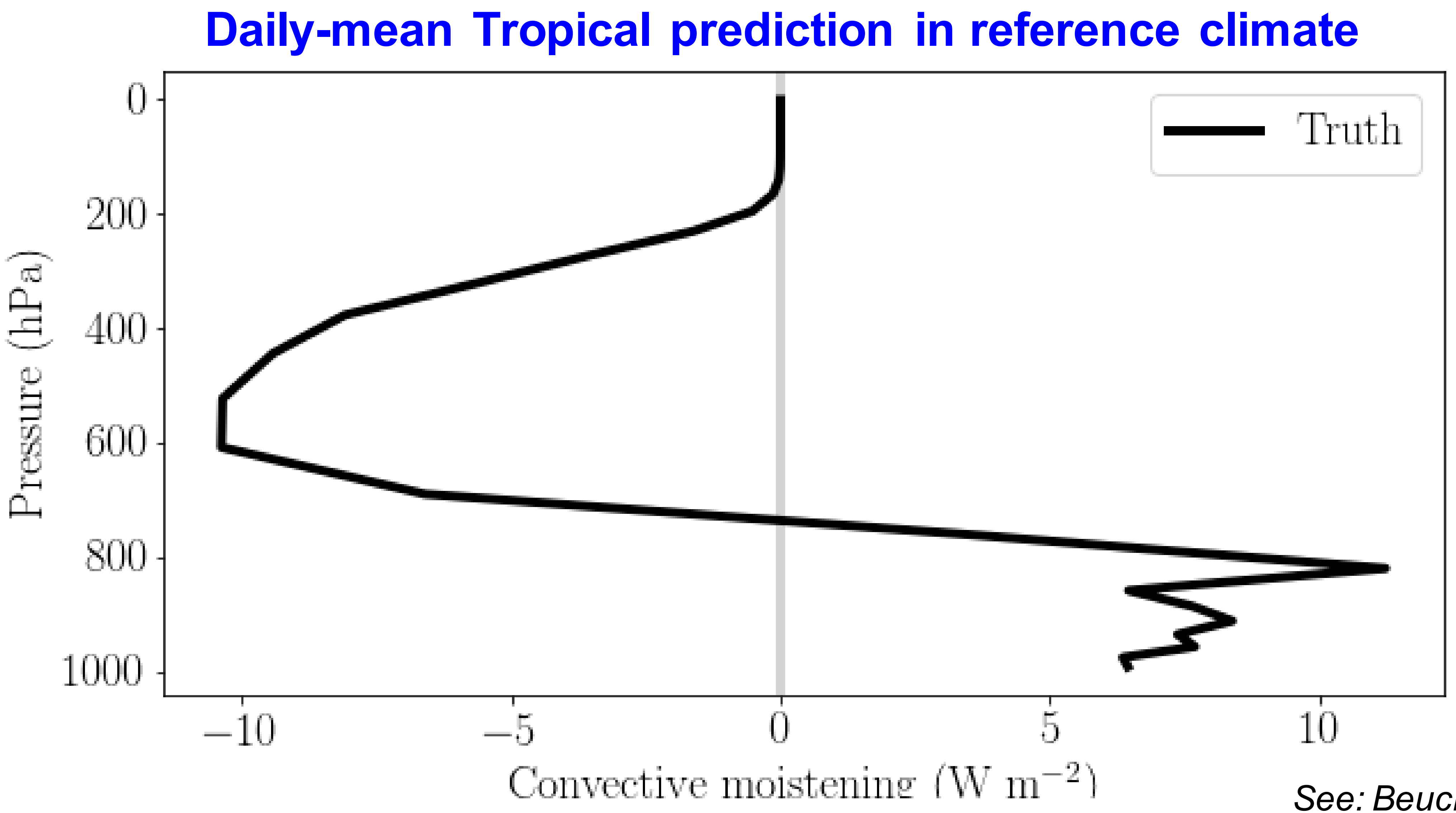
Problem 1: Neural Nets typically violate conservation laws



Problem 2: ML parametrizations are hard to interpret/trust

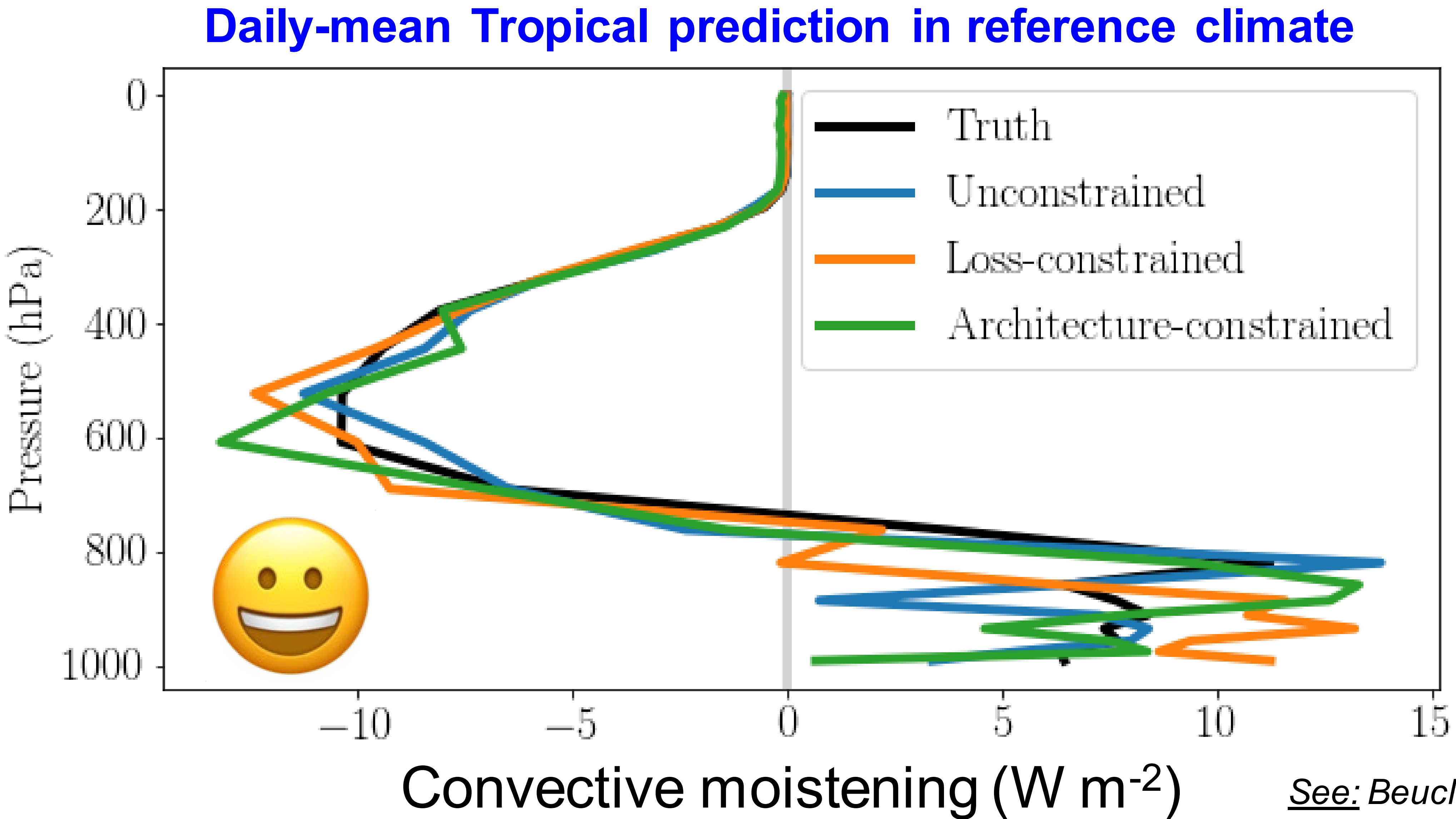
<u>See:</u> Brenowitz, Beucler et al. (2020)

Problem 3: ML algorithms fail to generalize



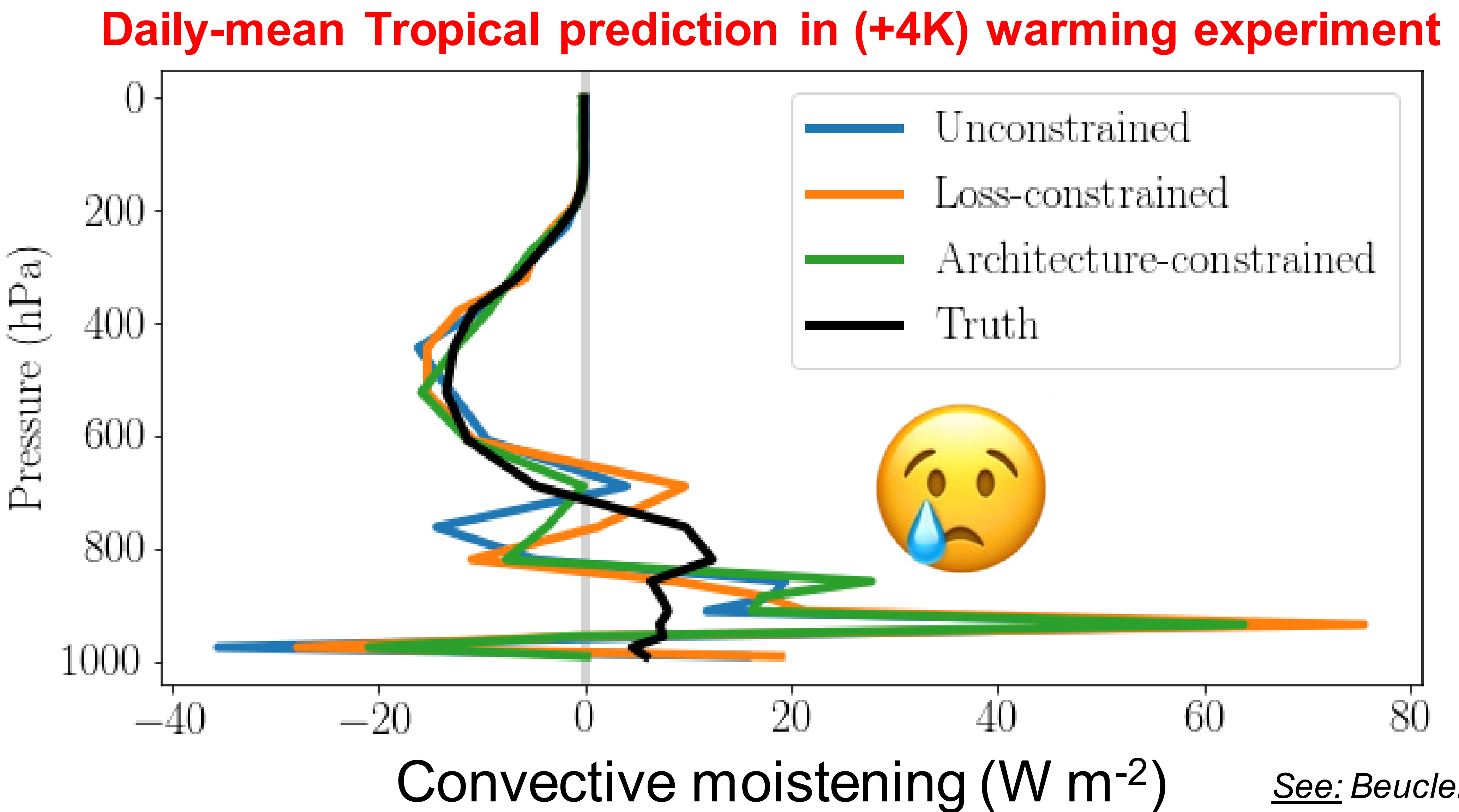
See: Beucler et al. (2019)

Problem 3: ML algorithms fail to generalize



See: Beucler et al. (2019)

Problem 3: ML algorithms fail to generalize



<u>See:</u> *Beucler et al. (2019)*

Problem 1: ML algorithms violate conservation laws

Problem 2: ML parametrization hard to interpret/trust

Problem 3: ML algorithms fail to generalize

How can we design
interpretable, physically-consistent & data-
driven
models of convection?

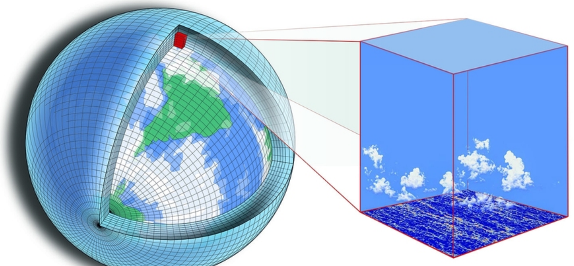
How to best combine ML & physical knowledge?

Physics-Guided ML: Add physical structure to restrict ML output to physically-plausible solutions

Physical Structure

<u>Reviews</u>: Willard et al. (2020), Reichstein et al. (2019), Karpatne et al. (2017), Beucler et al. (2021)

Physics-Guided ML: Add physical structure to restrict ML output to physically-plausible solutions

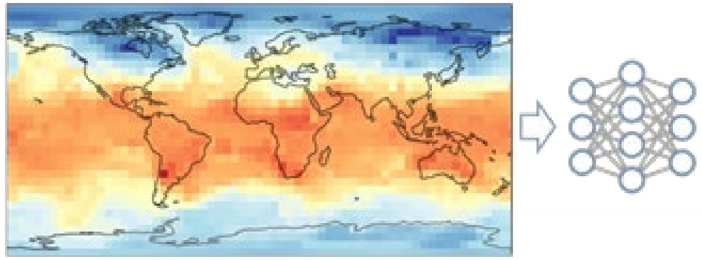


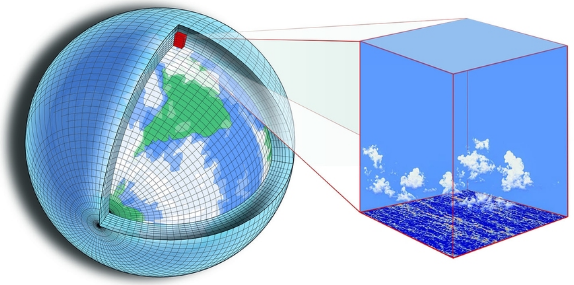
Physical Structure

Learn Parameters of Physical Model

__See__: Schneider et al. (2017), Reichstein et al. (2019), Camps-Vall et al. (2018), __Image Source__: CliMA, Caltech

Physics-Guided ML: Add physical structure to restrict ML output to physically-plausible solutions





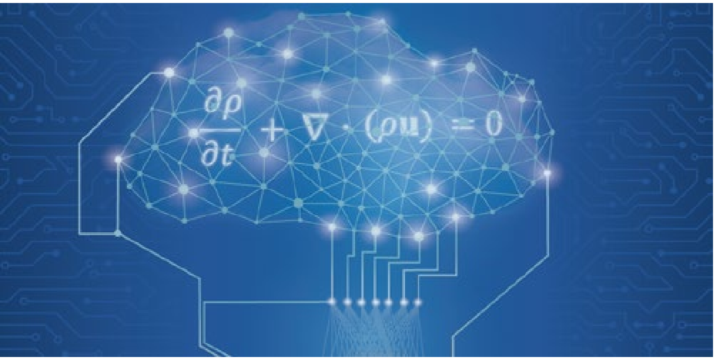
Physical Structure

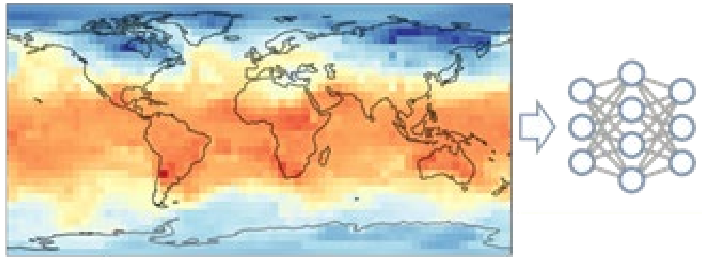
Bias Correction of Physical Model

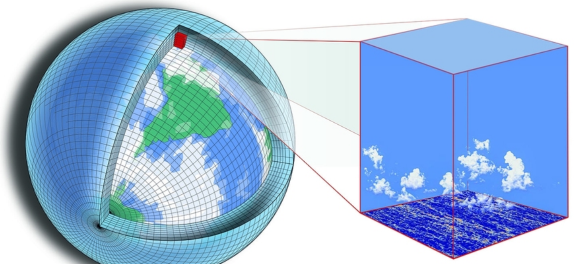
Learn Parameters of Physical Model

See: Rasp and Lerch (2018), Grönquist et al. (2021), Bonavita and Laloyaux (2020), Image Source: Rasp et al. (2020)

Physics-Guided ML: Add physical structure to restrict ML output to physically-plausible solutions







Physical Structure

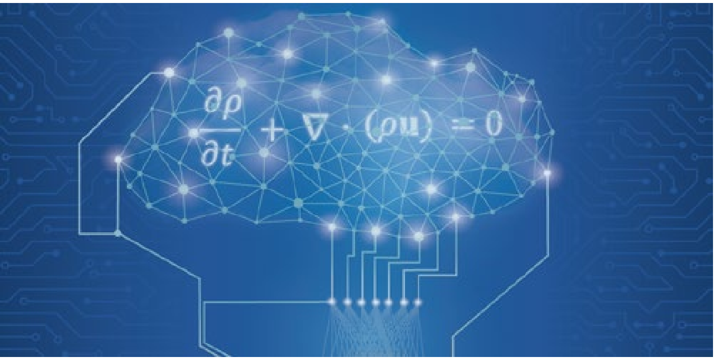
Physics-Constrained Loss or Architecture

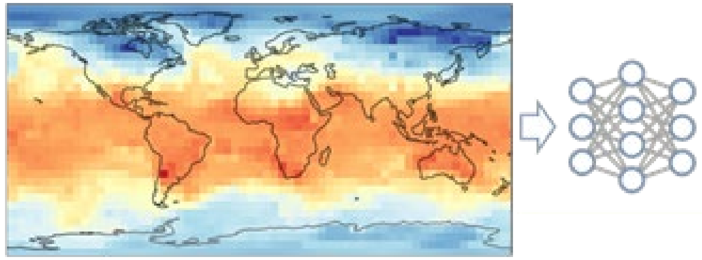
Bias Correction of Physical Model

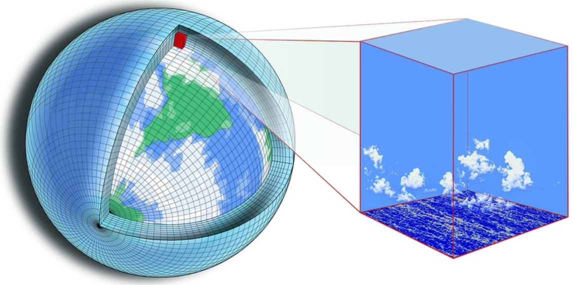
Learn Parameters of Physical Model

__See__: Karpatne et al. (2017), Wu et al. (2020), Raissi et al. (2019), __Image Source__: R. Gauthier-Butterfield, UCI (2021)

Problem 1: Neural Nets typically violate conservation laws







Physical Structure

Physics-Constrained Loss or Architecture

Bias Correction of Physical Model

Learn Parameters of Physical Model

See: Karpatne et al. (2017), Wu et al. (2020), Raissi et al. (2019), Image Source: R. Gauthier-Butterfield, UCI (2021)

Physics-Constrained **Loss Function**

Idea: Introduce a penalty for violating conservation ($\sim$ Lagrange multiplier):

$$\boxed{\mathrm{Loss} = \alpha\,(\mathrm{Squared\ Residual\ from\ conservation\ laws}) + (1-\alpha)\,(\mathrm{Mean\ squared\ error})}$$

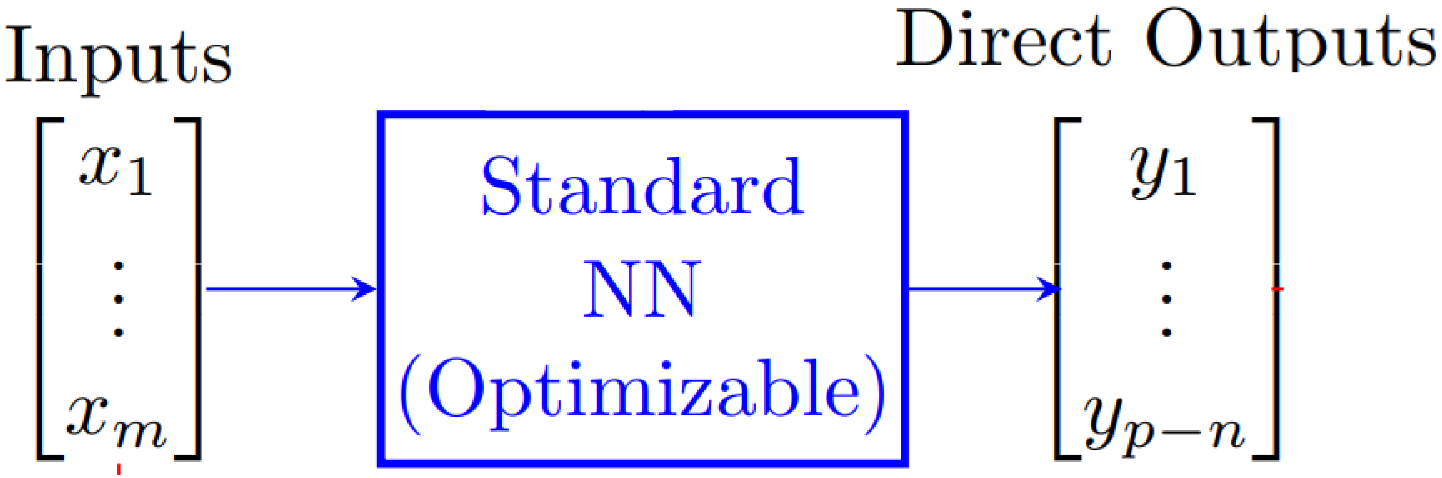
See: Beucler et al. (2021)

Physics-Constrained **Architecture**

<u>Idea</u>: Introduce a penalty for violating conservation (~ Lagrange multiplier):

$$\boxed{\text{Loss} = \alpha\,(\text{Squared Residual from conservation laws}) + (1-\alpha)\,(\text{Mean squared error})}$$

Constraint layers to enforce conservation laws to within machine precision!



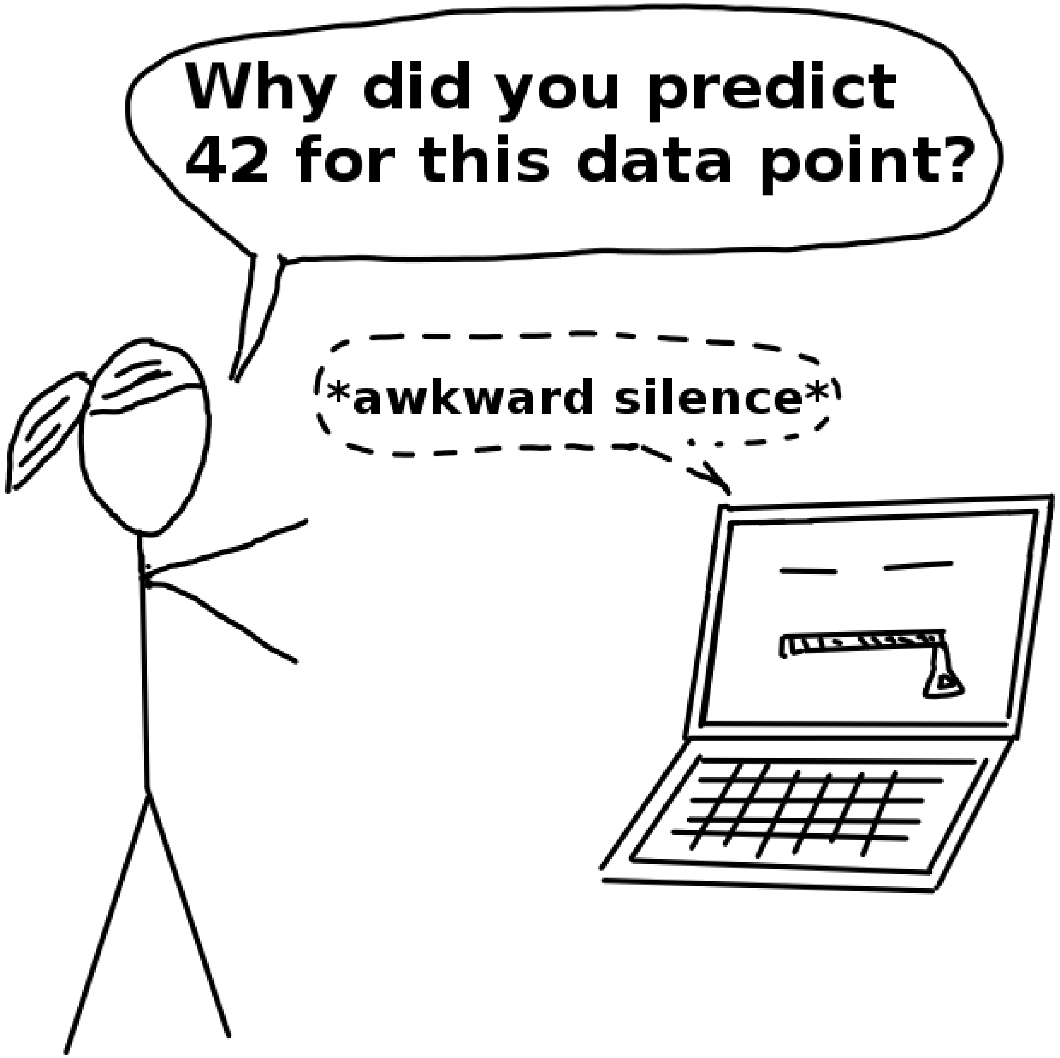
<u>See:</u> Beucler et al. (2021)

Problem 1: Neural Nets typically violate conservation laws

We can enforce conservation laws in NNs
Conservation of mass, energy, and radiation

<u>See:</u> *Beucler et al. (2021)*

Problem 2: For climate modeling, we need trustworthy/interpretable parametrizations



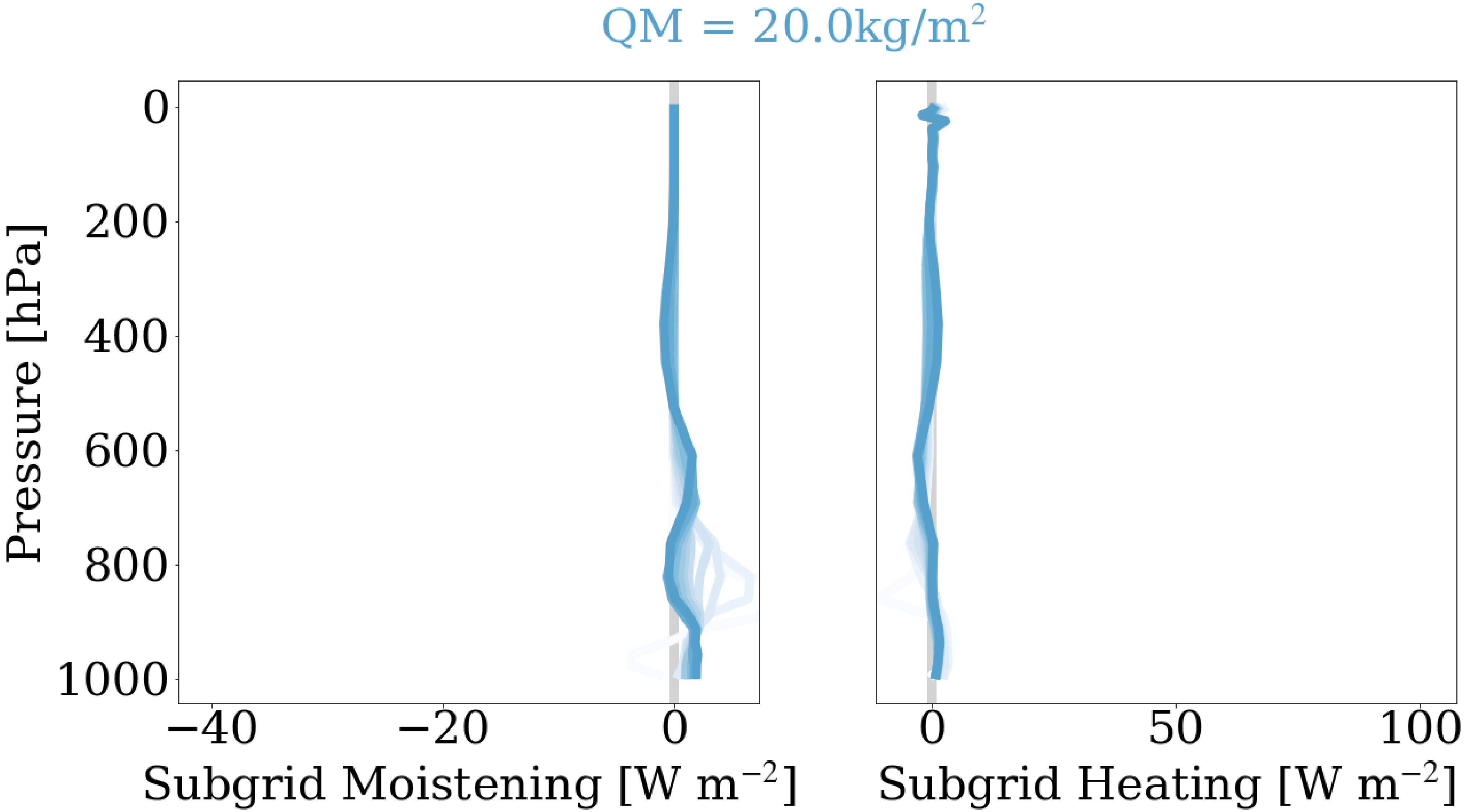
Source: Interpretable Machine Learning, C. Molnar (2019)

Problem 2: ML parametrizations are hard to interpret/trust

Idea: Tailor 2 NN interpretability methods to parameterization convection

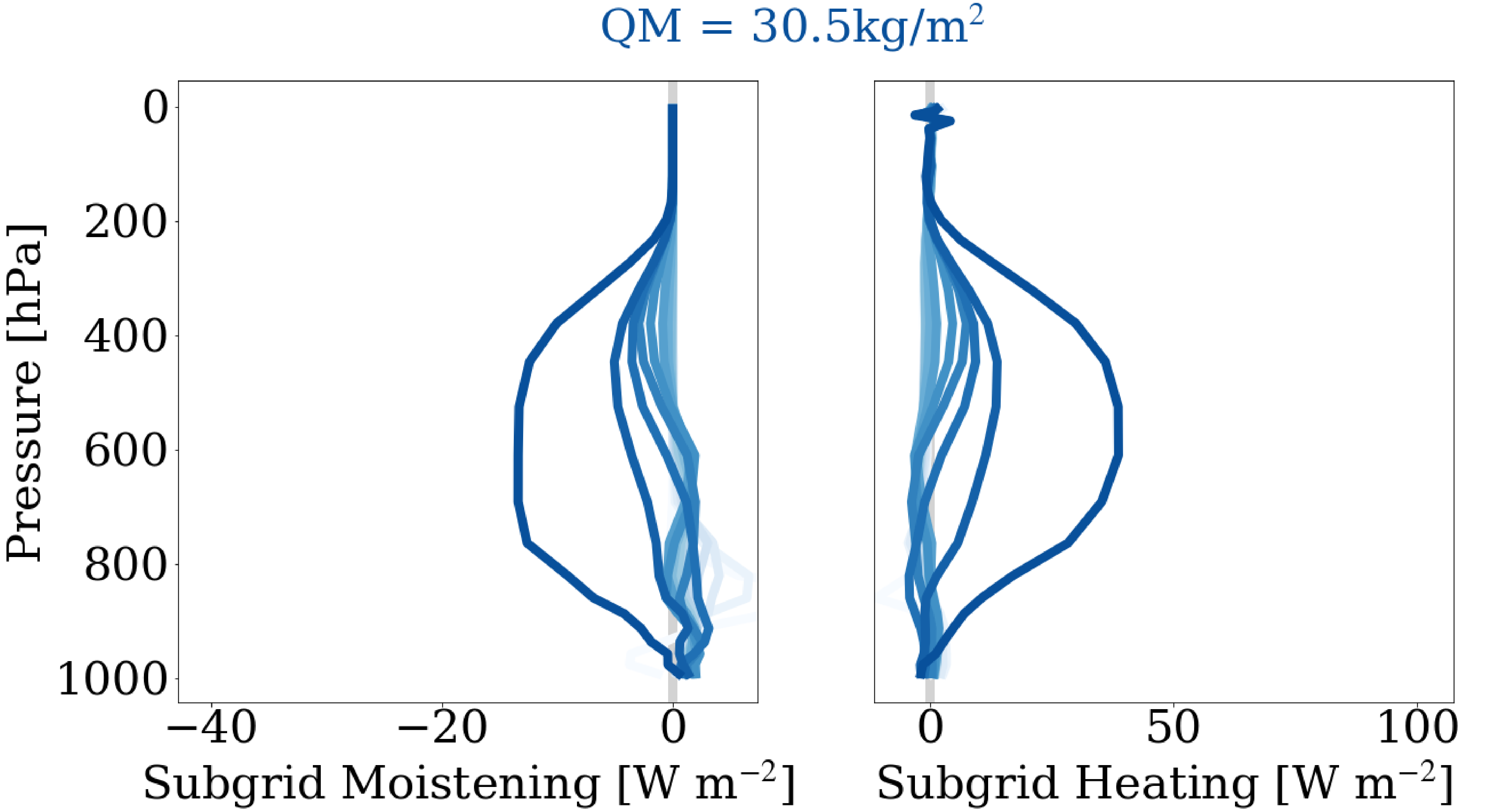
See: McGovern et al. (2019), Toms et al. (2019), Montavon et al. (2018), Molnar et al. (2018)

fixed l.t. stability, mid-tropospheric moisture fuels convection



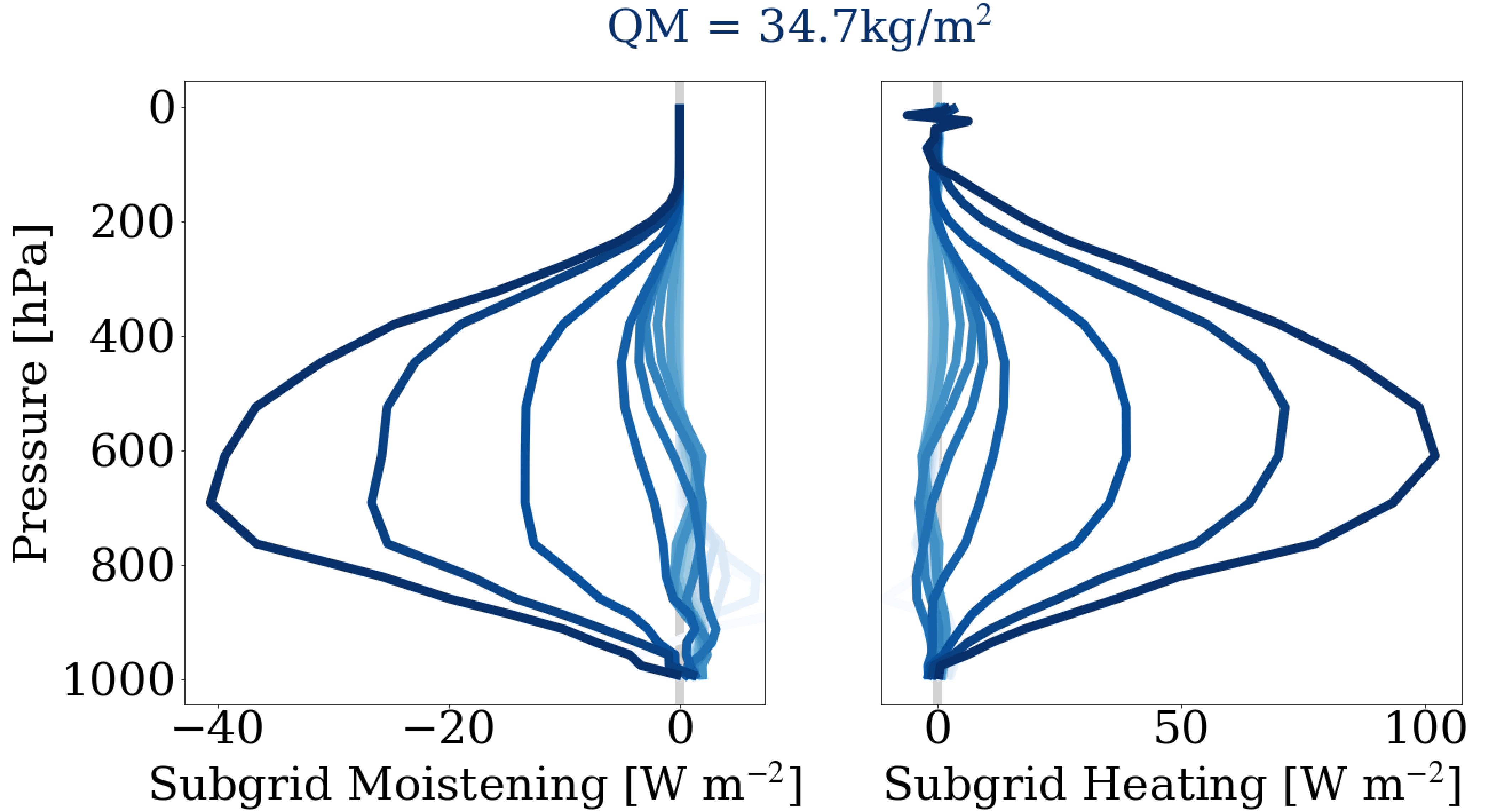
<u>See</u>: Brenowitz, Beucler et al. (2020)

fixed l.t. stability, mid-tropospheric moisture fuels convection



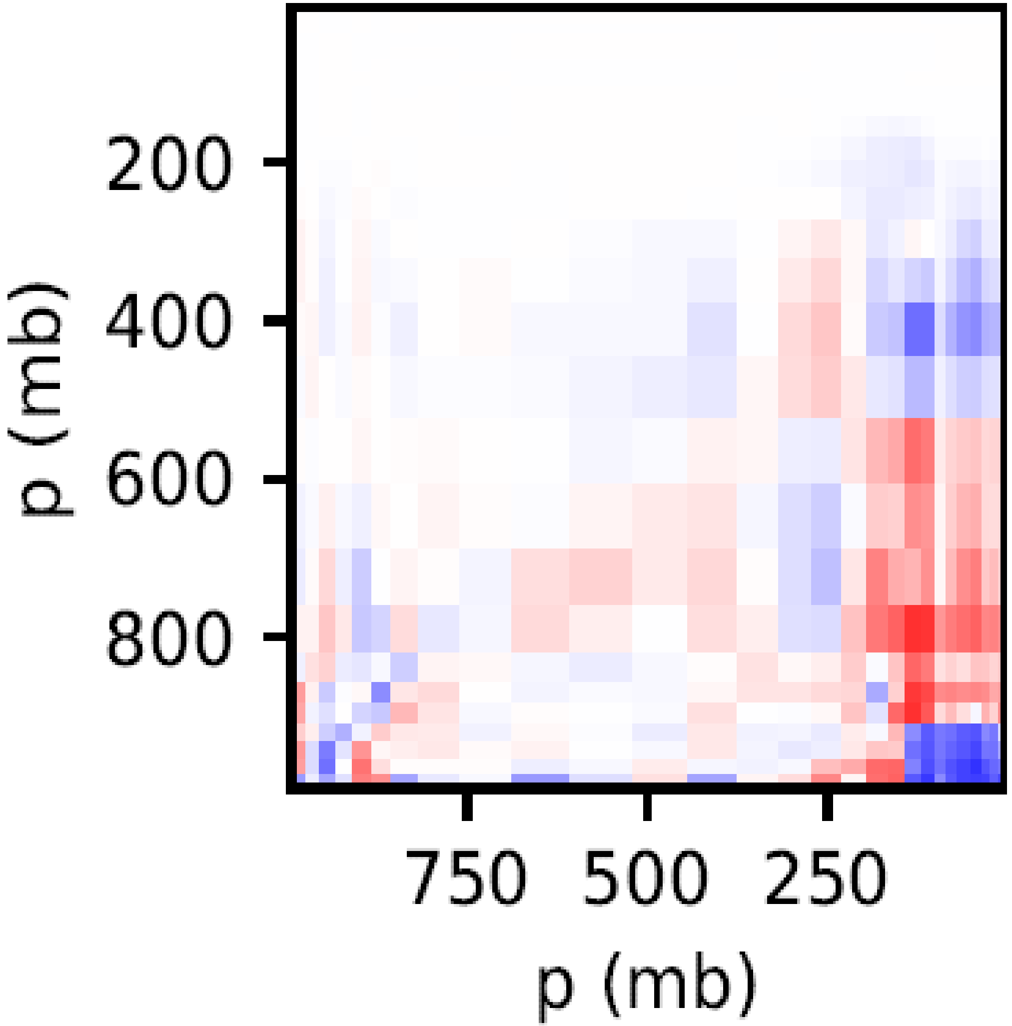
See: Brenowitz, Beucler et al. (2020)

fixed l.t. stability, mid-tropospheric moisture fuels convection



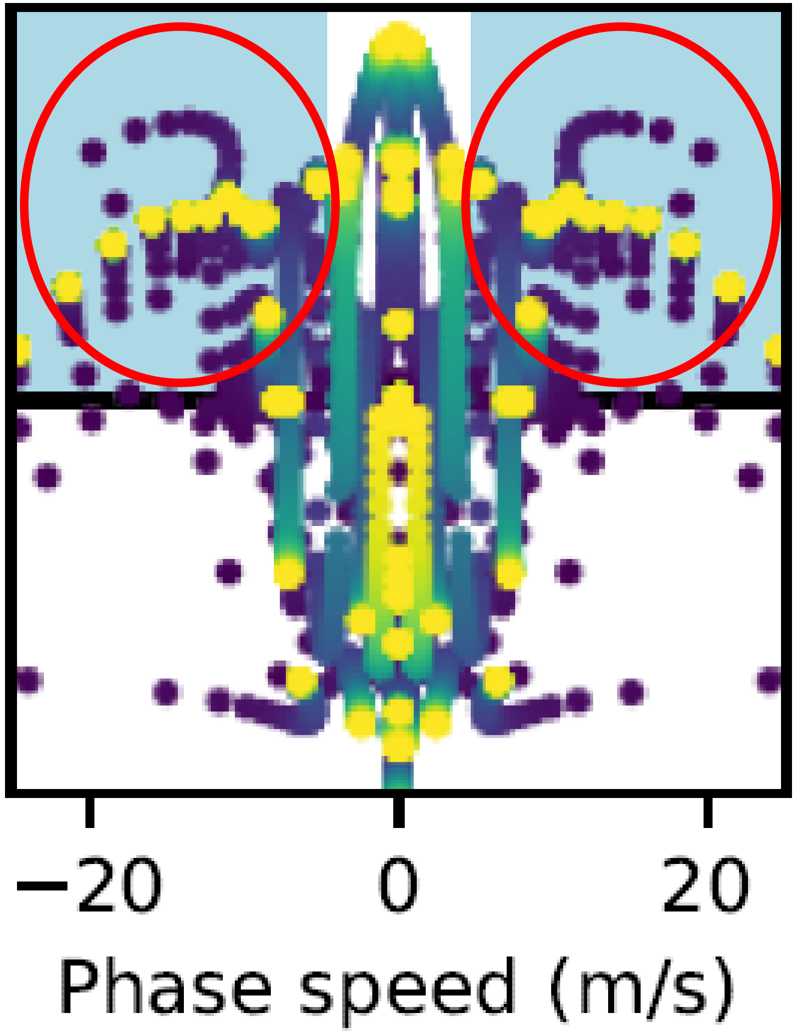
See: Brenowitz, Beucler et al. (2020)

Jacobian calculated via **automatic differentiation** helps interpret and stabilize parameterization



Spurious unstable propagating modes

Couple to linearized gravity wave dynamics



See: Kuang (2018, 2007), Herman and Kuang (2013), Beucler et al. (2018), ***Brenowitz, Beucler et al. (2020)***

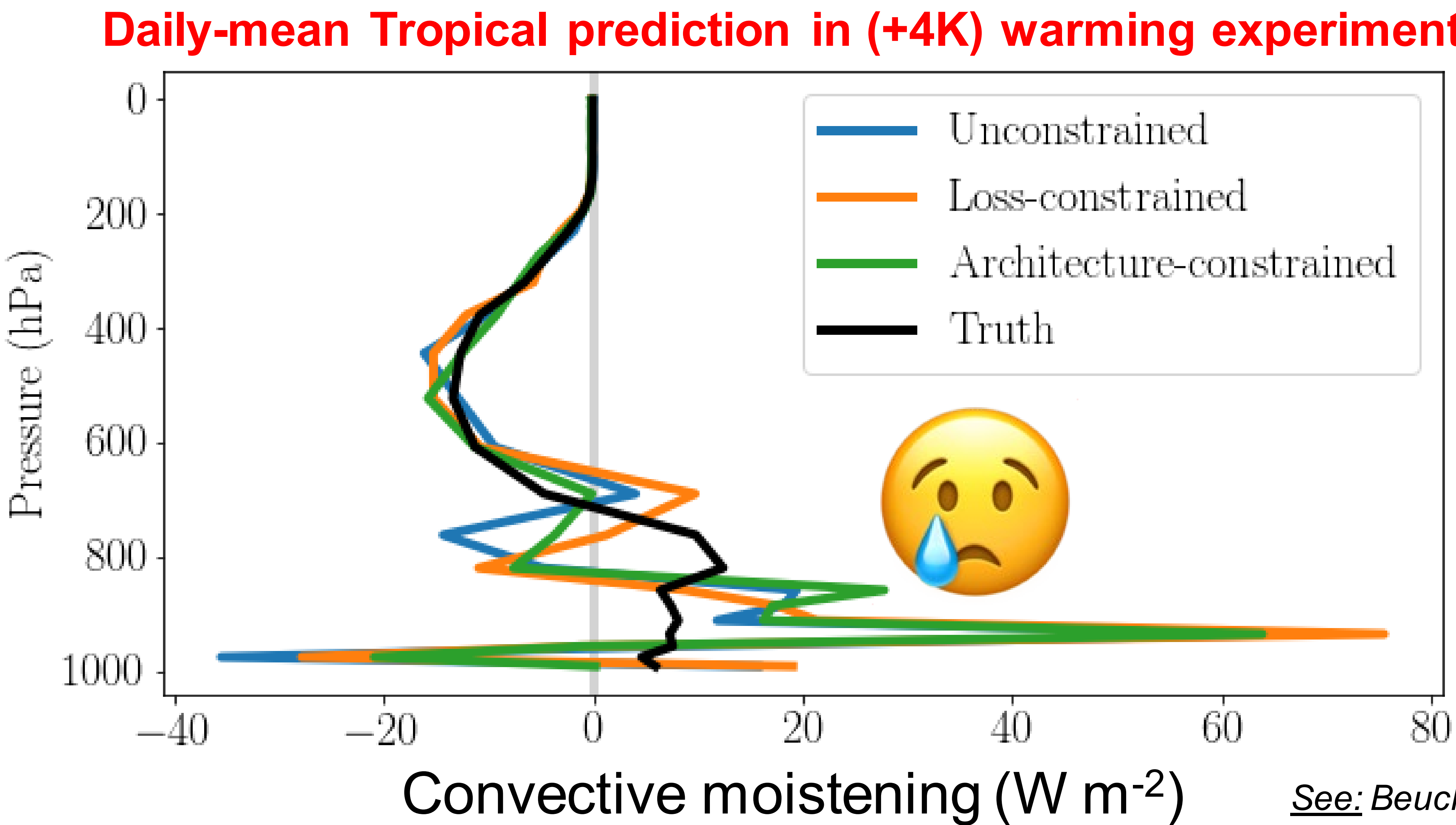
Problem 2: ML parametrizations are hard to interpret/trust

We can tailor interpretability methods
Partial Dependence Plots + Gradients

Attribution Maps

<u>See</u>: Brenowitz, Beucler et al. (2020)

Problem 3: ML algorithms fail to generalize



See: Beucler et al. (2019)

Idea: Break the model even more!

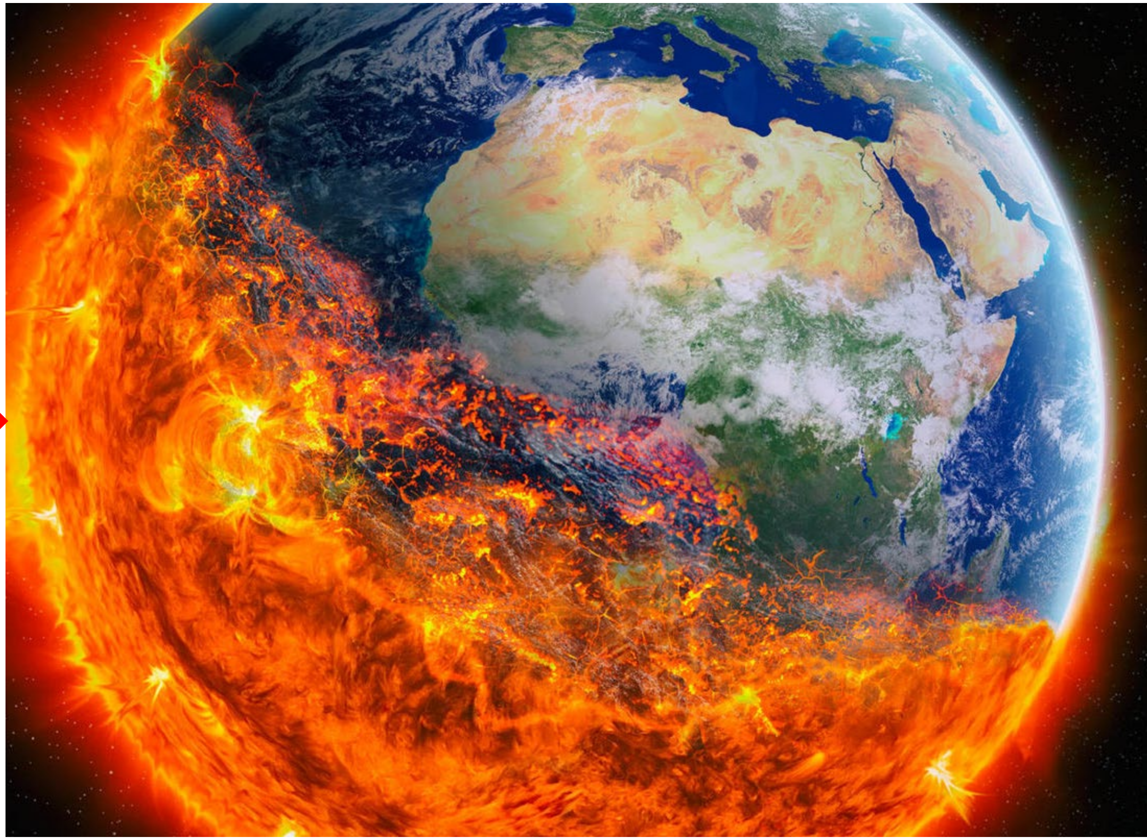
Image: IT Biz Advisor

Generalization Experiment: Uniform +8K warming

Training and Validation on cold aquaplanet simulation

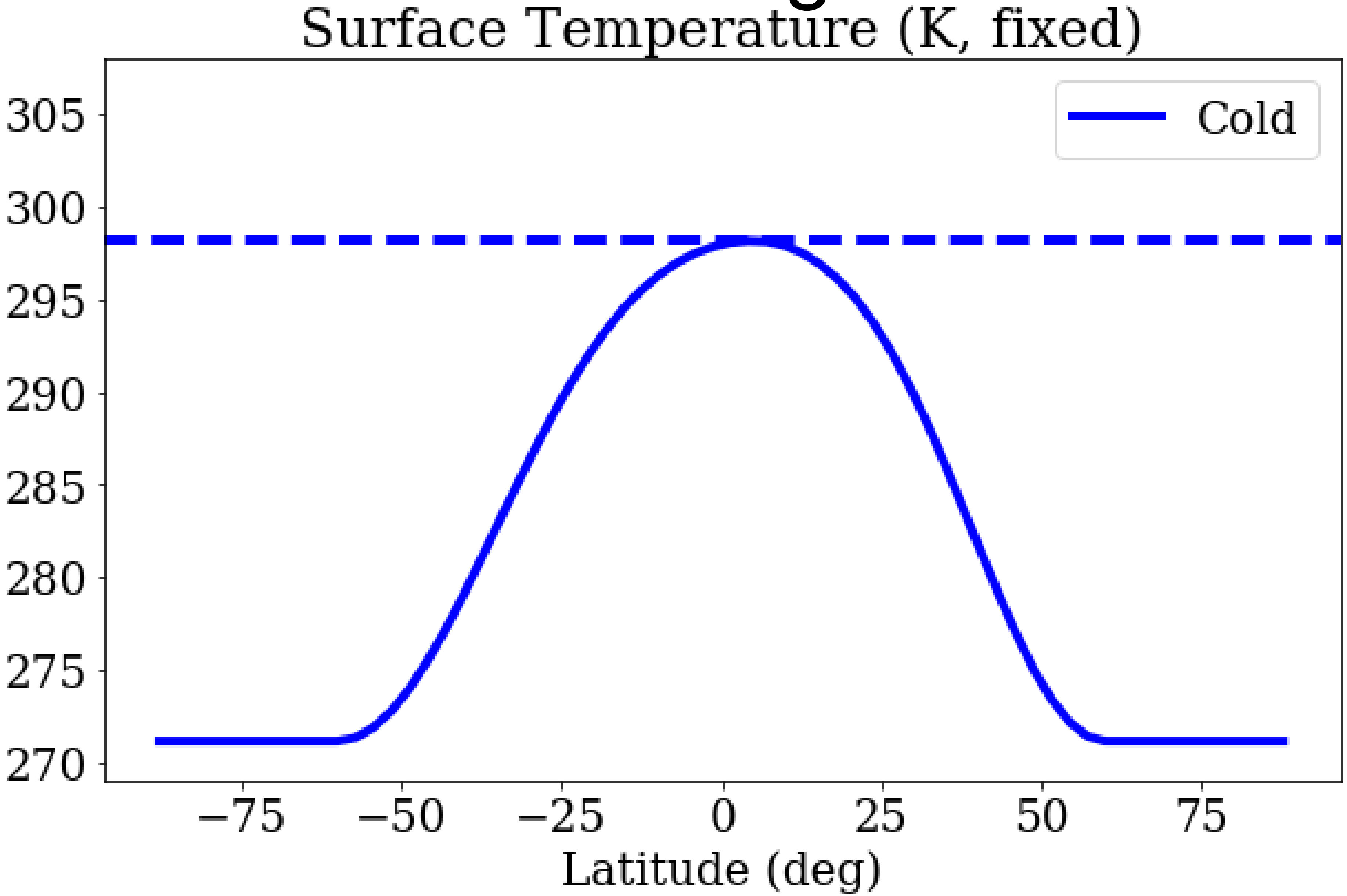
Test on warm aquaplanet simulation

+8K →

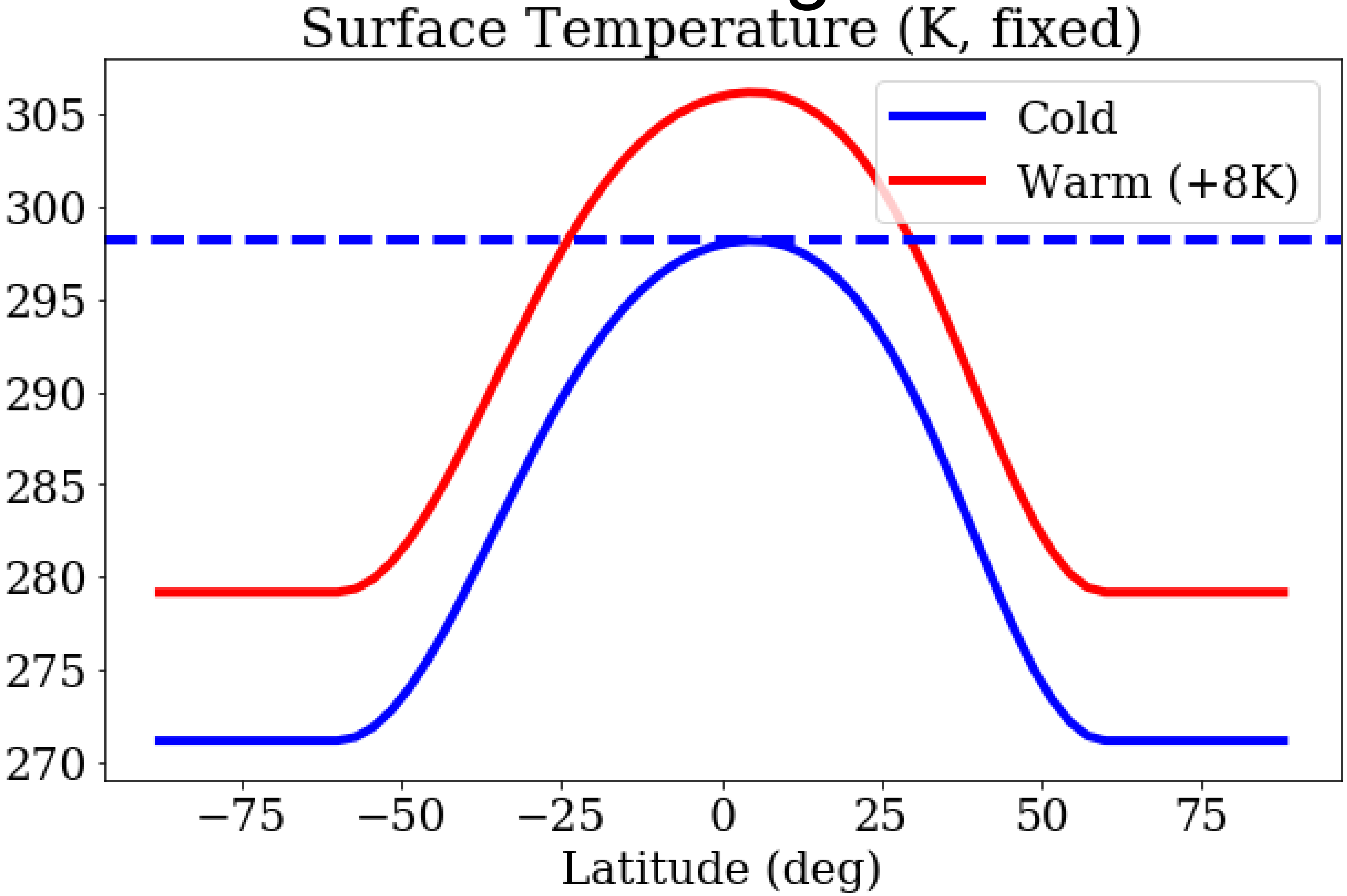


Images: Rashevskyi Viacheslav, Sebastien Decoret

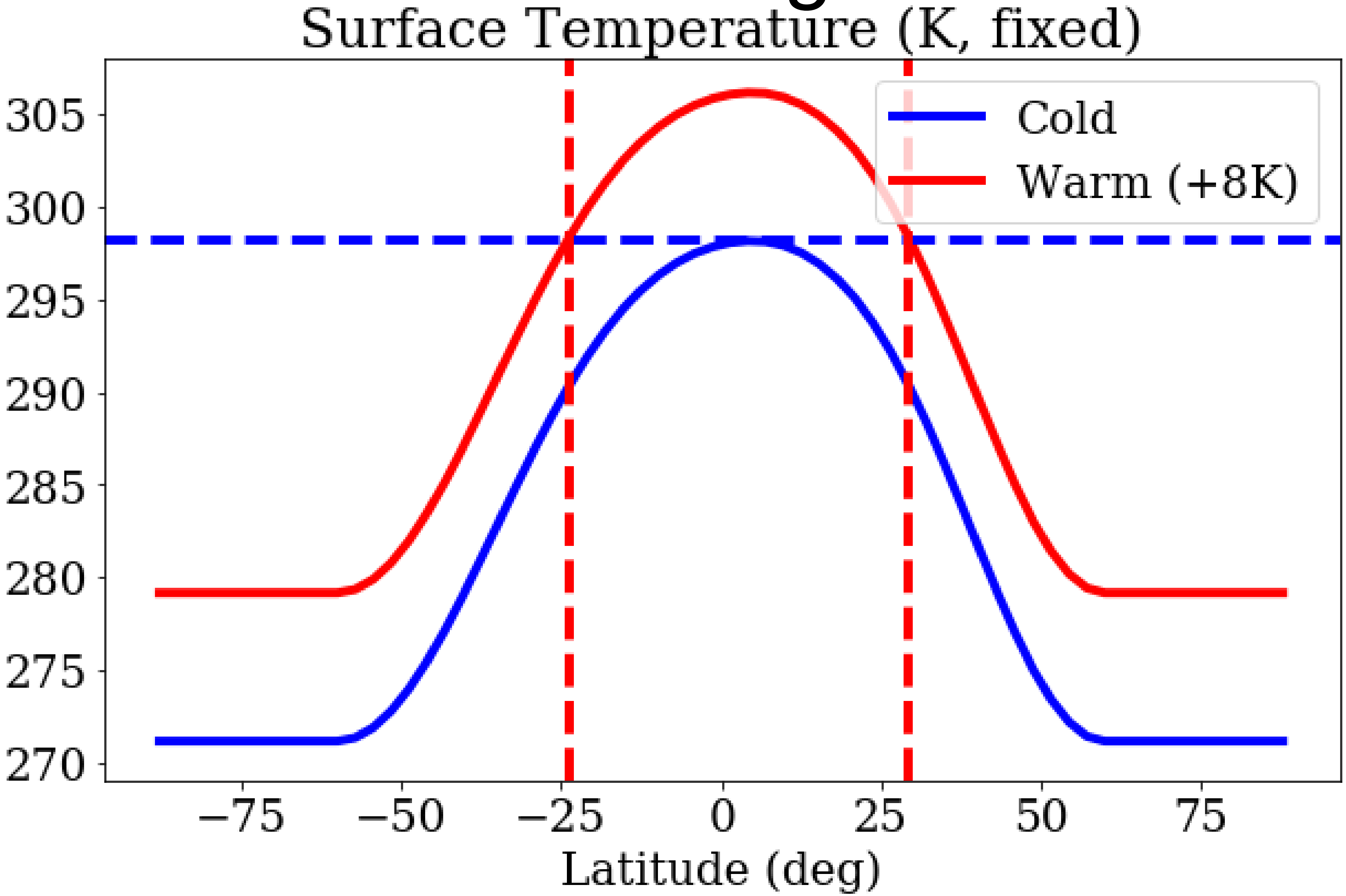
Generalization Experiment: Uniform +8K warming



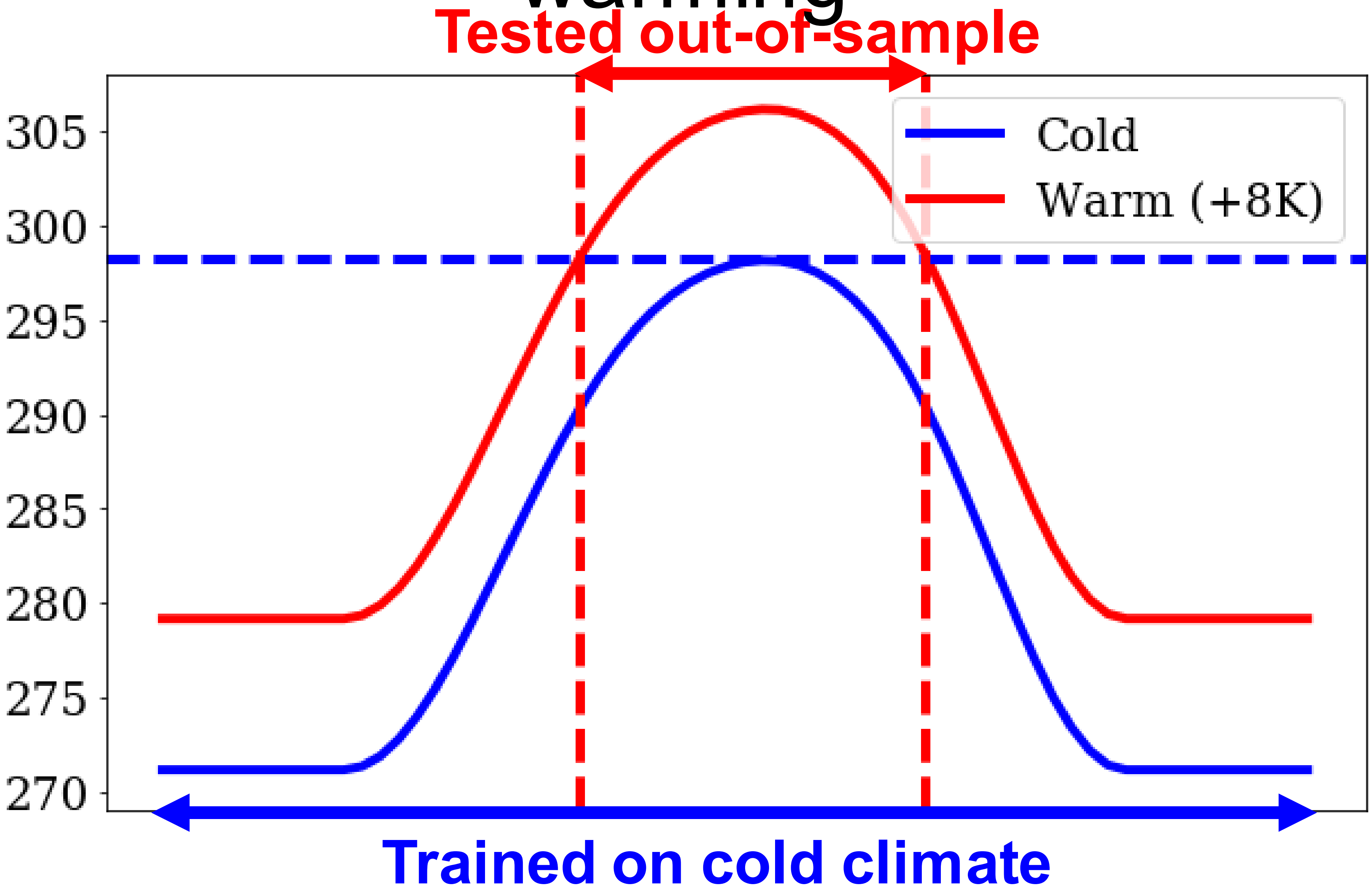
Generalization Experiment: Uniform +8K warming



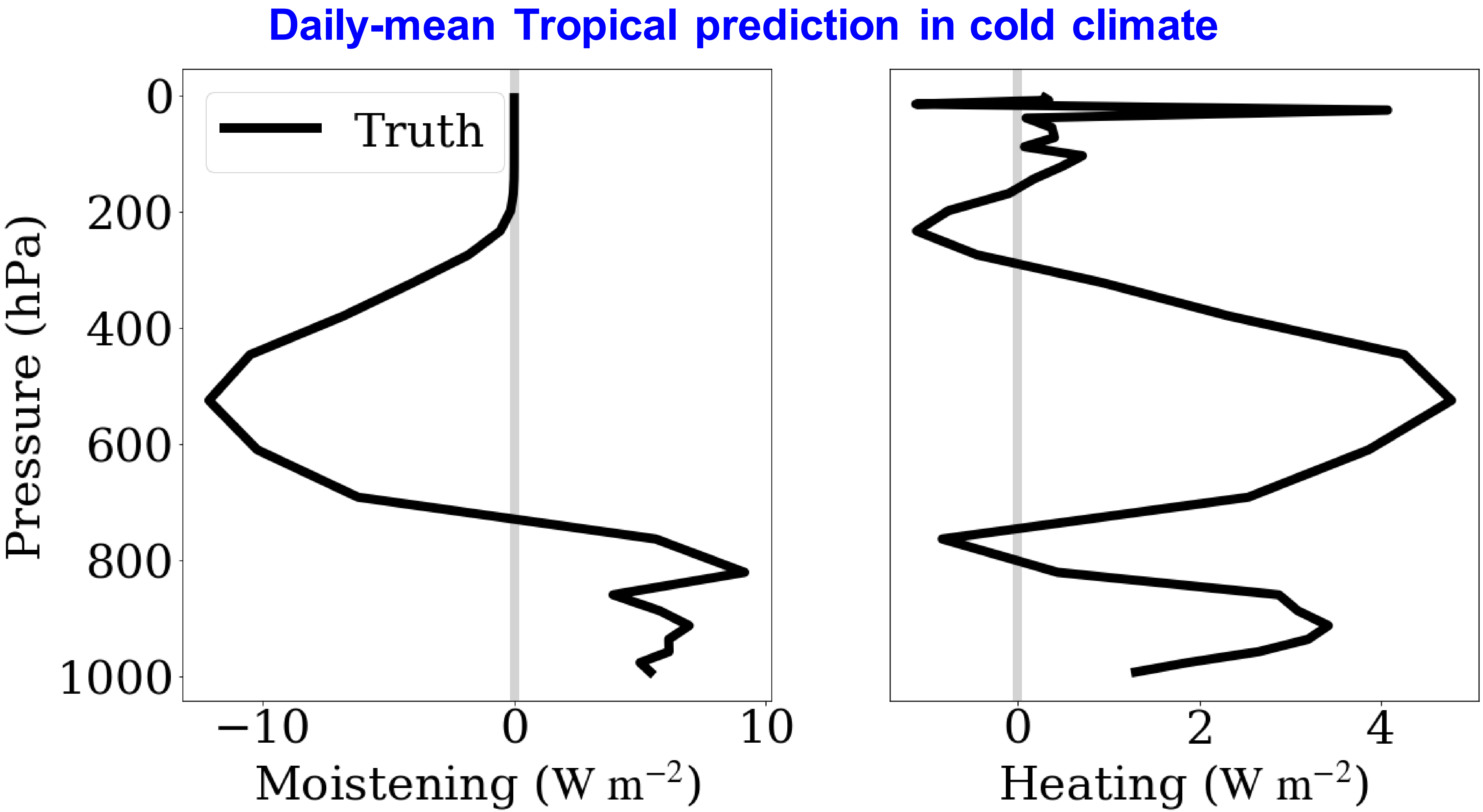
Generalization Experiment: Uniform +8K warming



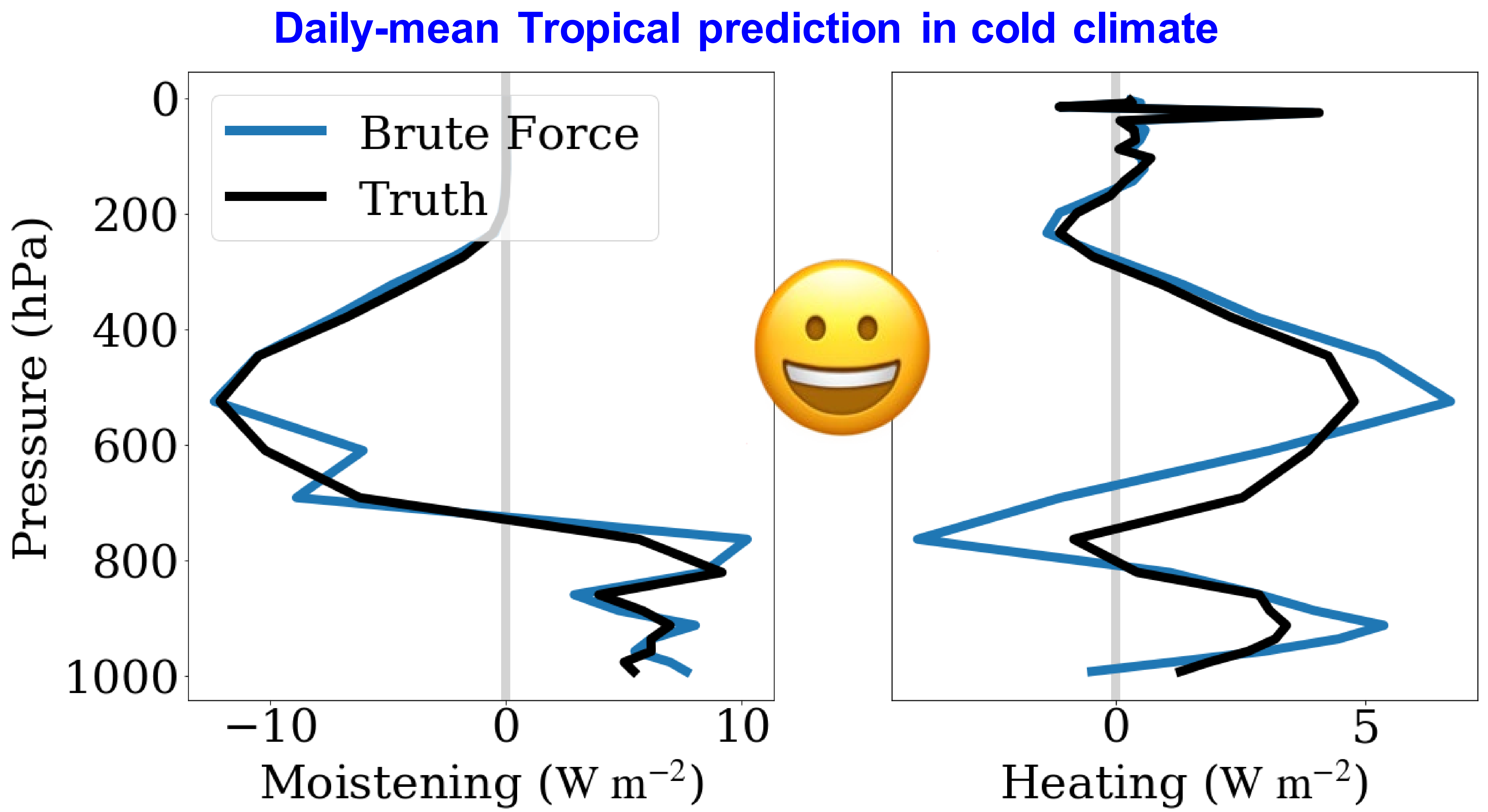
Generalization Experiment: Uniform +8K warming



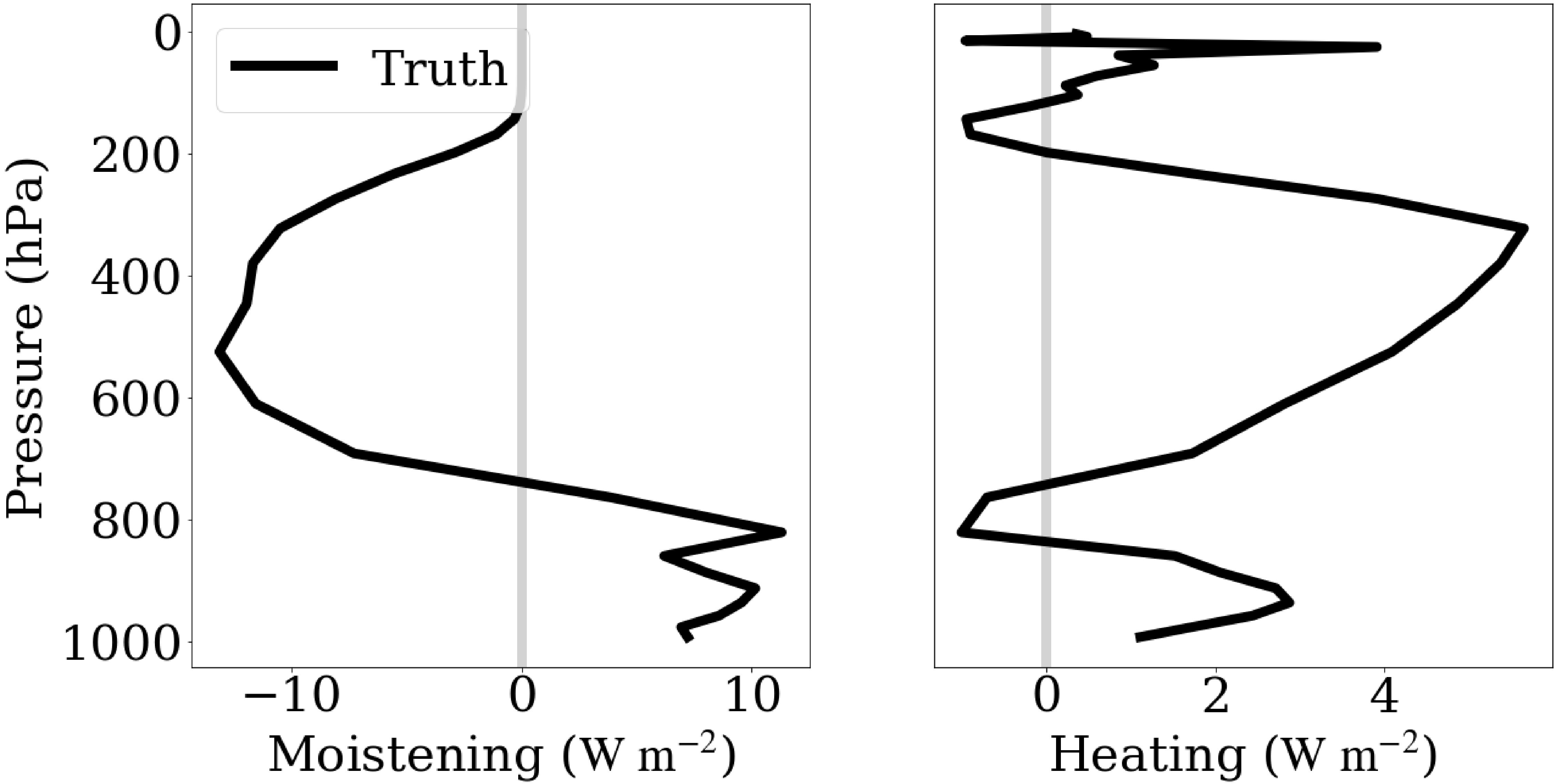
Problem 3: NNs fail to generalize to unseen climates



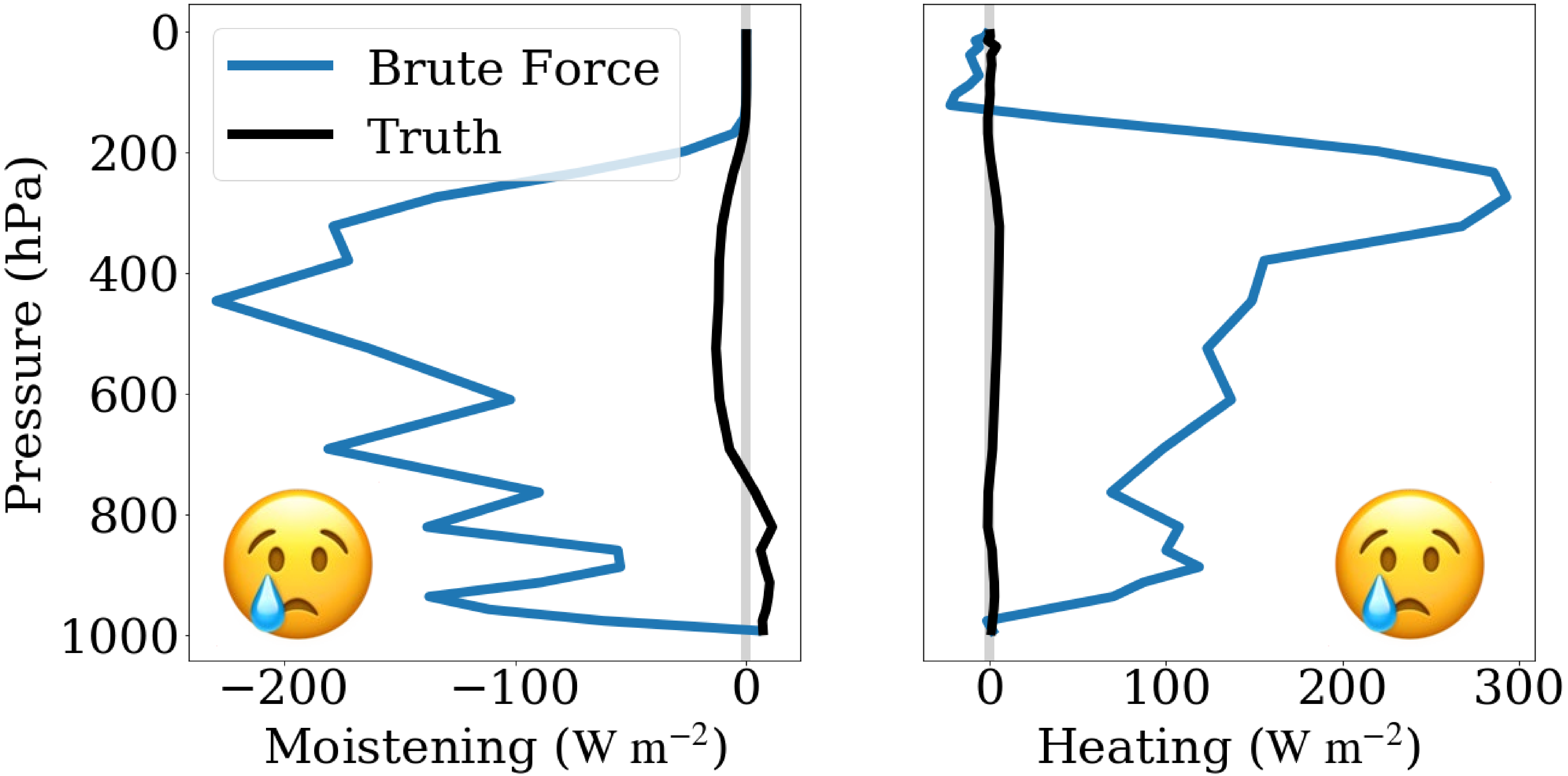
Problem 3: NNs fail to generalize to unseen climates



Daily-mean Tropical prediction in warm climate



Daily-mean Tropical prediction in warm climate



Physically rescale the data to convert extrapolation into interpolation

$$\begin{bmatrix} \text{Specific humidity}\,(p) \\ \text{Temperature}\,(p) \\ \text{Surface Pressure} \\ \text{Solar Insolation} \\ \text{Latent Heat Flux} \\ \text{Sensible Heat Flux} \end{bmatrix} \overset{\text{NN}}{\longmapsto} \begin{bmatrix} \text{Subgrid moistening}\,(p) \\ \text{Subgrid heating}\,(p) \\ \text{Radiative fluxes} \end{bmatrix}$$

<u>Brute Force</u>: Not Climate-Invariant

Physically rescale the data to convert extrapolation into interpolation

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

$$\begin{bmatrix} \widetilde{\text{Specific humidity}}\,(\widetilde{p}) \\ \widetilde{\text{Temperature}}\,(\widetilde{p}) \\ \widetilde{\text{Surface Pressure}} \\ \widetilde{\text{Solar Insolation}} \\ \widetilde{\text{Latent Heat Flux}} \\ \widetilde{\text{Sensible Heat Flux}} \end{bmatrix} \overset{\widetilde{\text{NN}}}{\mapsto} \begin{bmatrix} \widetilde{\text{Subgrid moistening}}\,(\widetilde{p}) \\ \widetilde{\text{Subgrid heating}}\,(\widetilde{p}) \\ \widetilde{\text{Radiative fluxes}} \end{bmatrix}$$

Goal: Climate-Invariant

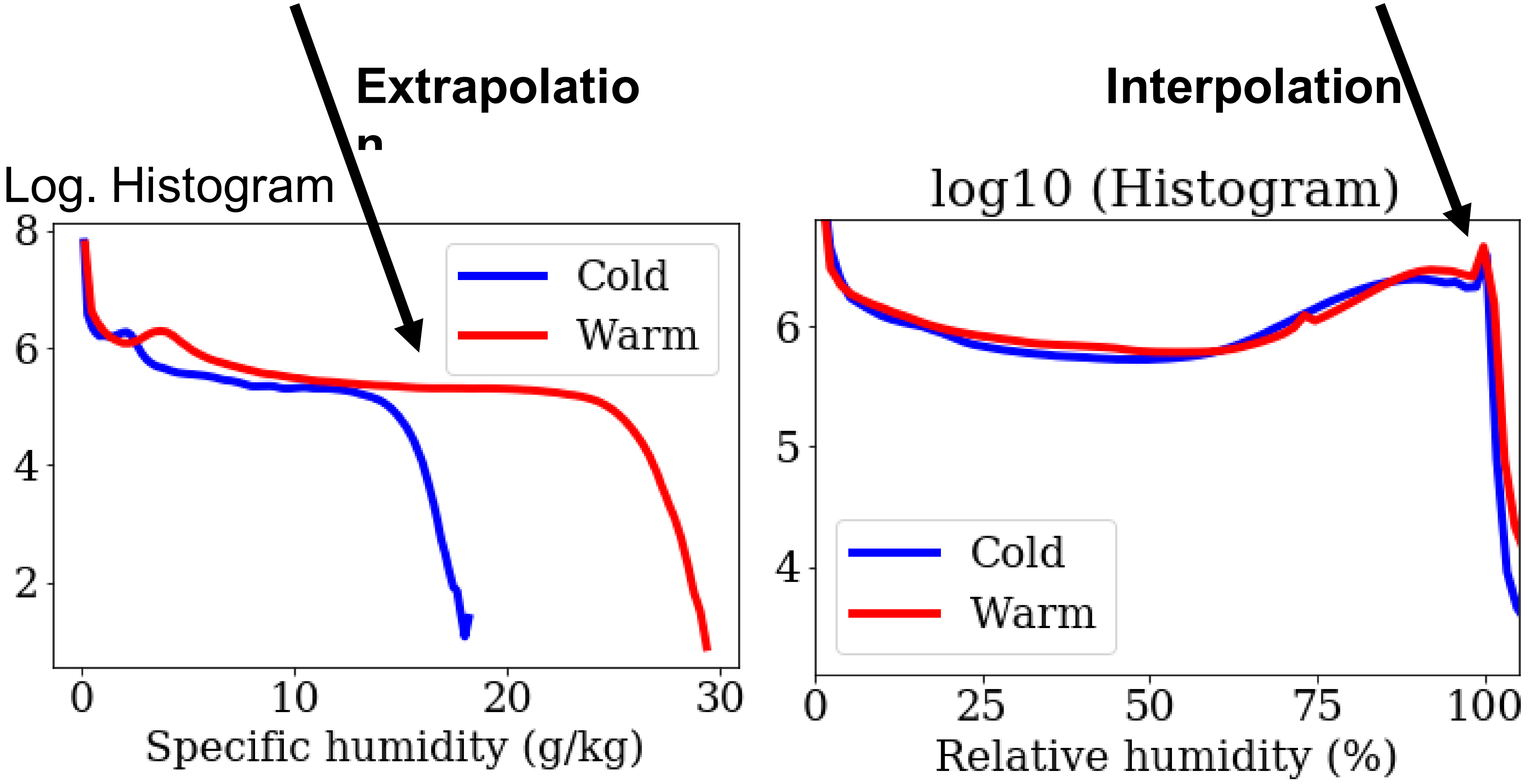
Physically rescale the data
to convert extrapolation into interpolation

<u>Goal</u>: Uncover **climate-invariant** mapping from climate to convection

$$\begin{bmatrix} \widetilde{\text{Specific humidity}}\,(\tilde{p}) \\ \widetilde{\text{Temperature}}\,(\tilde{p}) \\ \widetilde{\text{Surface Pressure}} \\ \widetilde{\text{Solar Insolation}} \\ \widetilde{\text{Latent Heat Flux}} \\ \widetilde{\text{Sensible Heat Flux}} \end{bmatrix} \overset{\widetilde{\text{NN}}}{\longmapsto} \begin{bmatrix} \widetilde{\text{Subgrid moistening}}\,(\tilde{p}) \\ \widetilde{\text{Subgrid heating}}\,(\tilde{p}) \\ \widetilde{\text{Radiative fluxes}} \end{bmatrix}$$

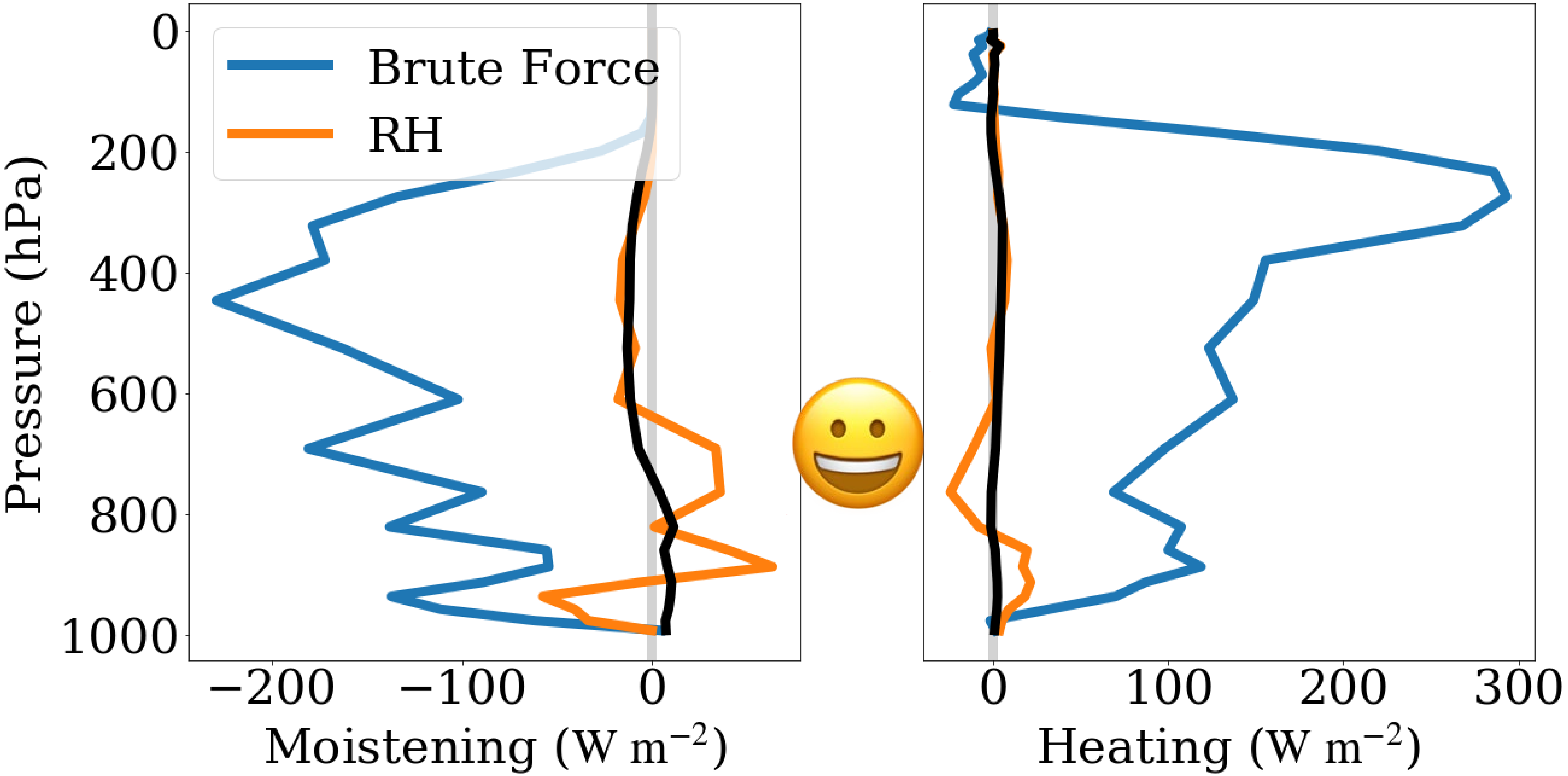
How to choose the physical

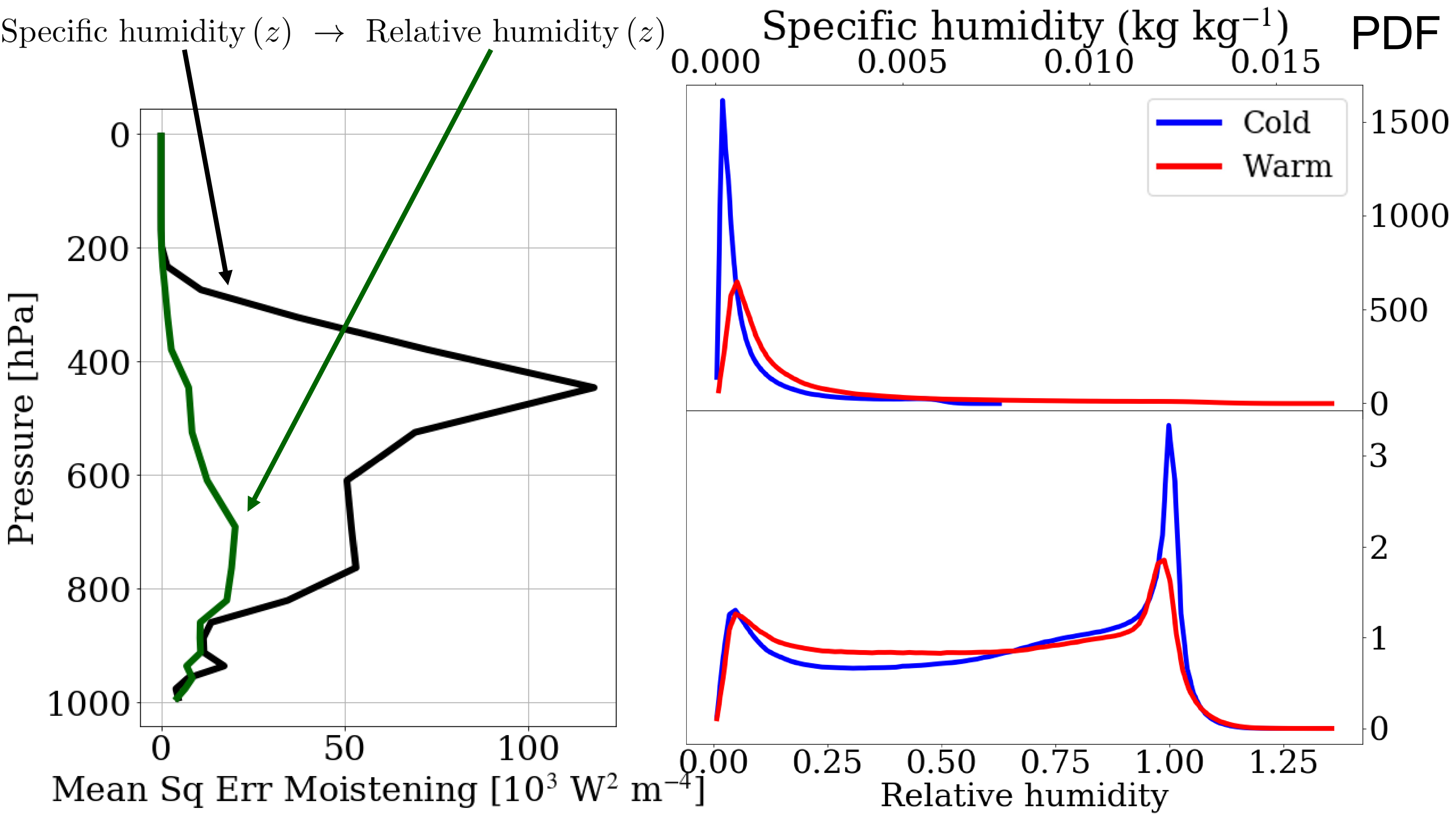
$\text{Specific humidity}\,(z) \;\rightarrow\; \text{Relative humidity}\,(z)$



Specific humidity $(z) \rightarrow$ Relative humidity (z)

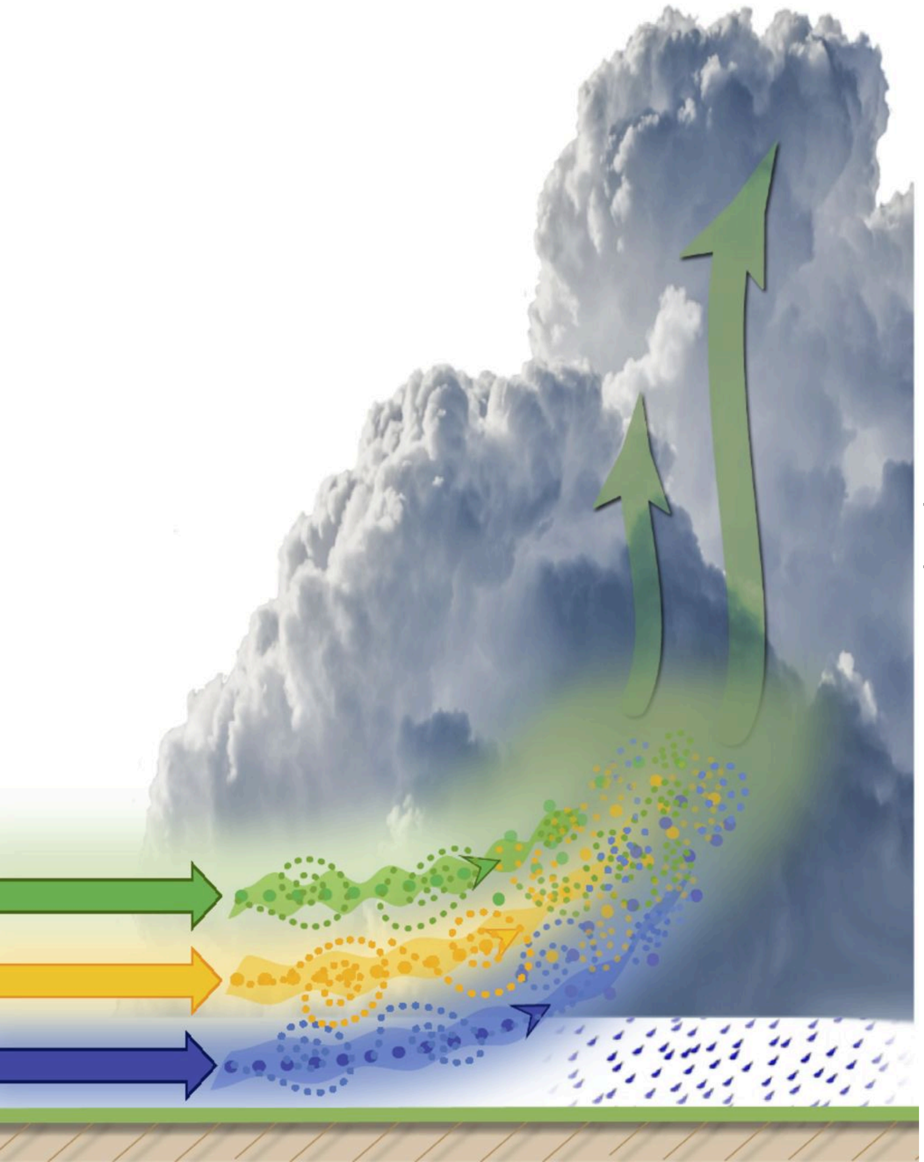
Generalization improves dramatically!





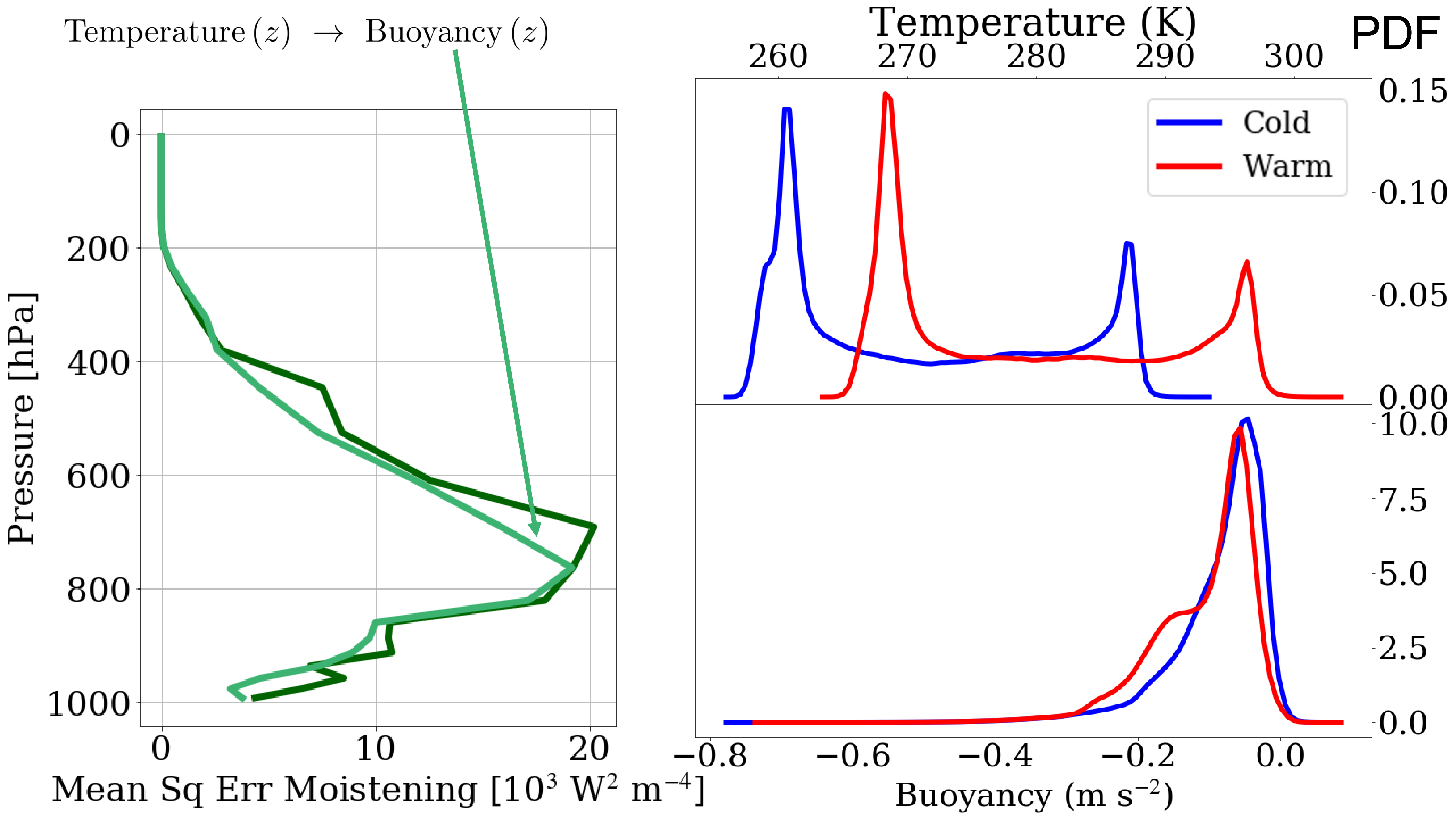
Specific humidity (z) → Relative humidity (z)
Pressure [hPa]
0
200
400
600
800
1000
0
50
100
Mean Sq Err Moistening [10^3 W^2 m^{-4}]
Specific humidity (kg kg^{-1})
0.000
0.005
0.010
0.015
PDF
Cold
Warm
1500
1000
500
0
3
2
1
0
0.00
0.25
0.50
0.75
1.00
1.25
Relative humidity

Observations suggest a strong relationship between buoyancy & moist convection across scales

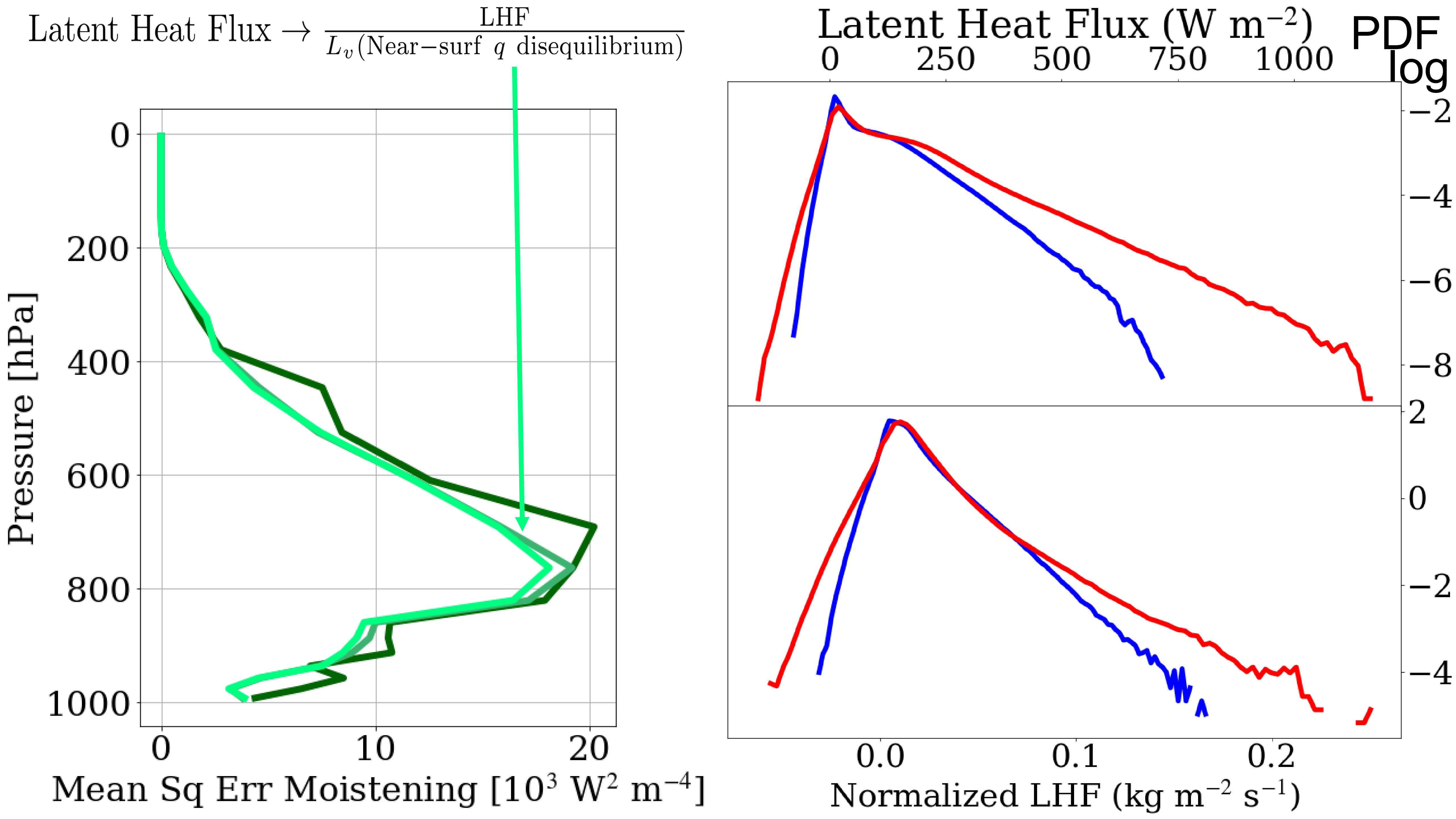


$$\text{Buoyancy}\,(z) \stackrel{\text{def}}{=} g \times \frac{\text{Temp parcel} - \text{Temp}(z)}{\text{Temp}(z)}$$

<u>See:</u> Schiro et al. (2018), Ahmed & Neelin (2018), Ahmed et al. (2020)

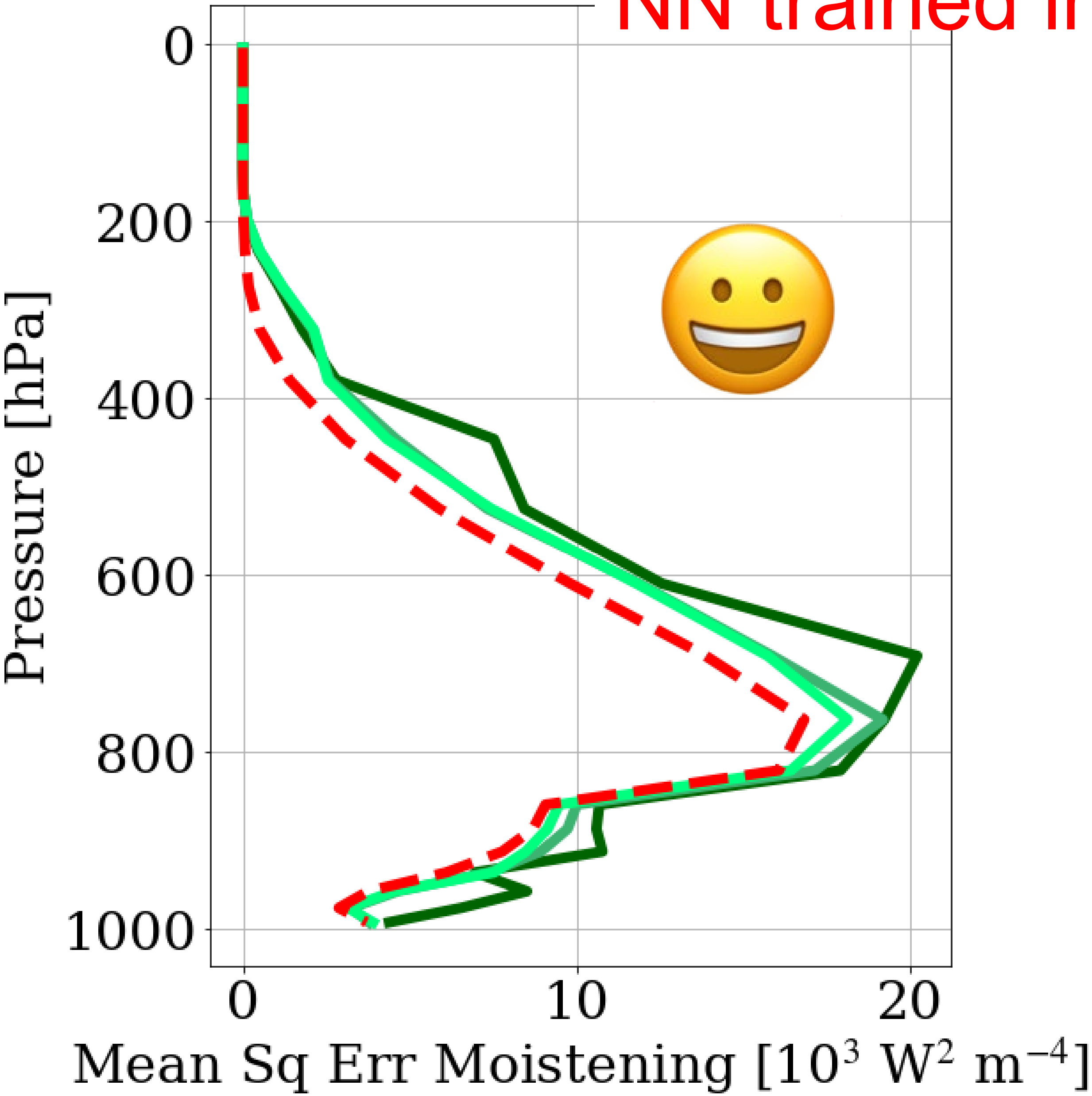


$\mathrm{Temperature}\,(z) \rightarrow \mathrm{Buoyancy}\,(z)$
Pressure [hPa]
0
200
400
600
800
1000
0
10
20
Mean Sq Err Moistening [10^3 W^2 m^{-4}]
Temperature (K)
260
270
280
290
300
PDF
0.15
0.10
0.05
0.00
10.0
7.5
5.0
2.5
0.0
Cold
Warm
−0.8
−0.6
−0.4
−0.2
0.0
Buoyancy (m s^{-2})

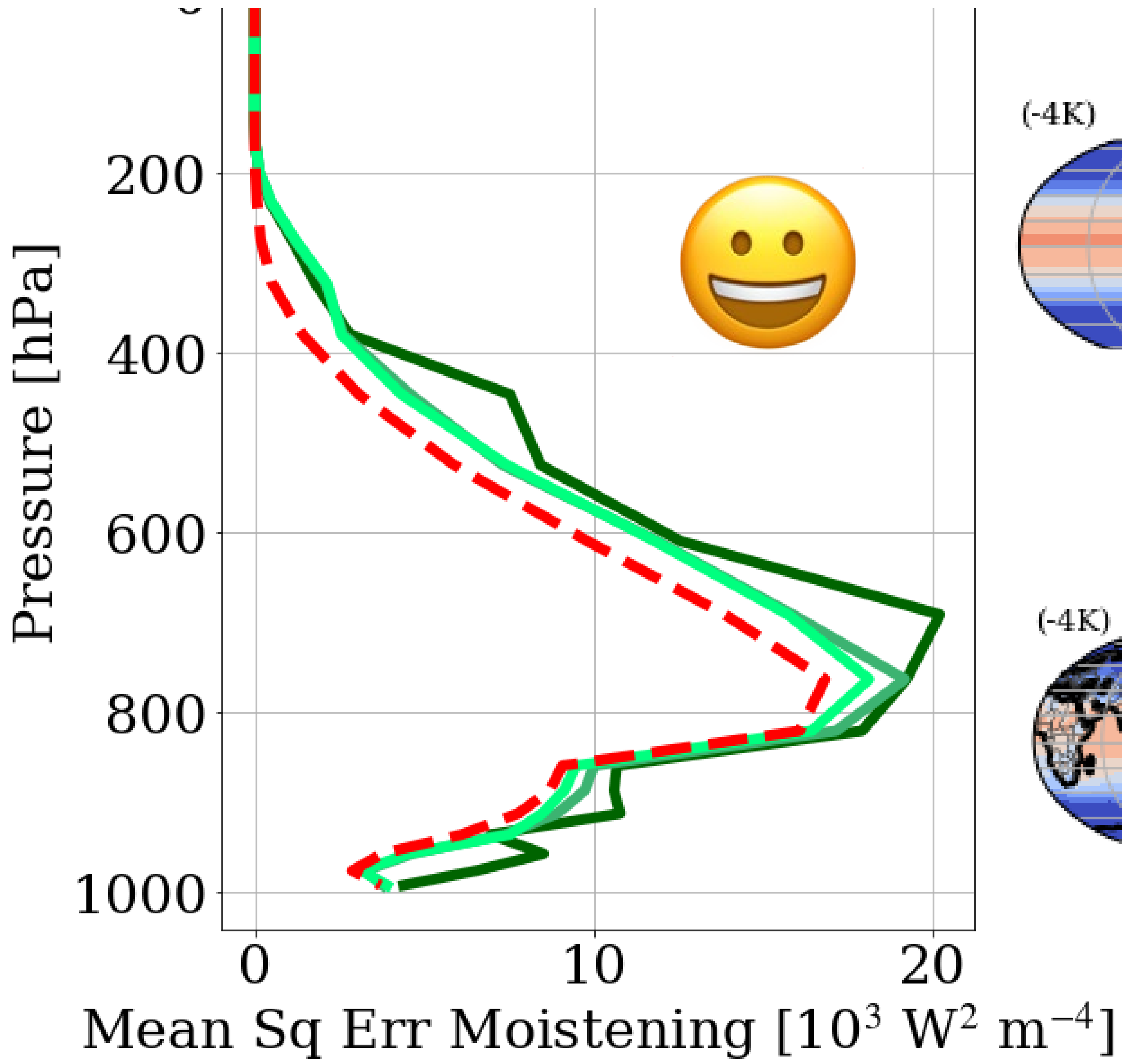


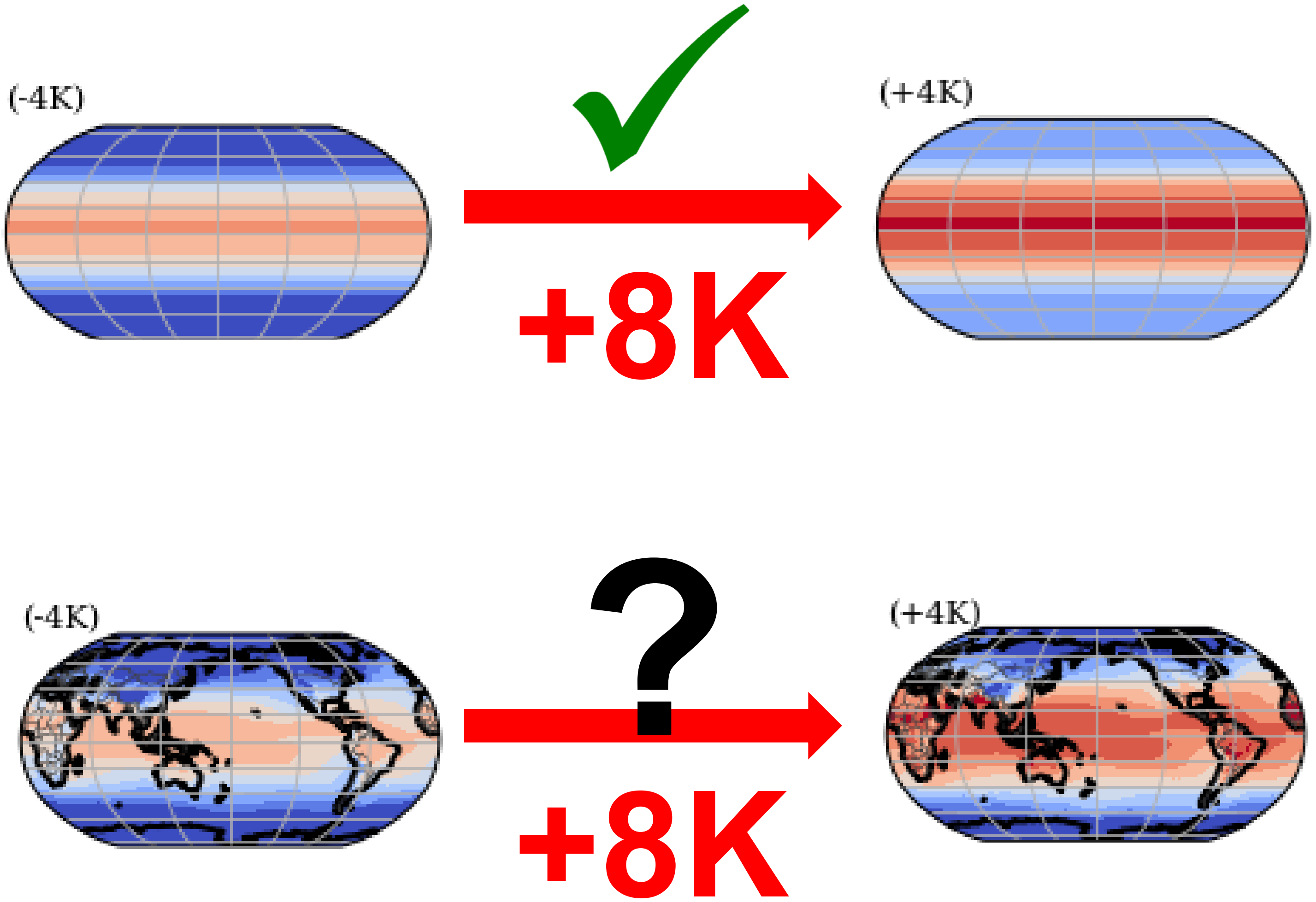
Latent Heat Flux → $\frac{\text{LHF}}{L_v(\text{Near-surf } q \text{ disequilibrium})}$
Pressure [hPa]
0
200
400
600
800
1000
0
10
20
Mean Sq Err Moistening [10^3 W^2 m^{-4}]
Latent Heat Flux (W m^{-2})
0
250
500
750
1000
PDF
log
−2
−4
−6
−8
2
0
−2
−4
0.0
0.1
0.2
Normalized LHF (kg m^{-2} s^{-1})

Climate-Invariant NNs generalization error close to NN trained in warm climate

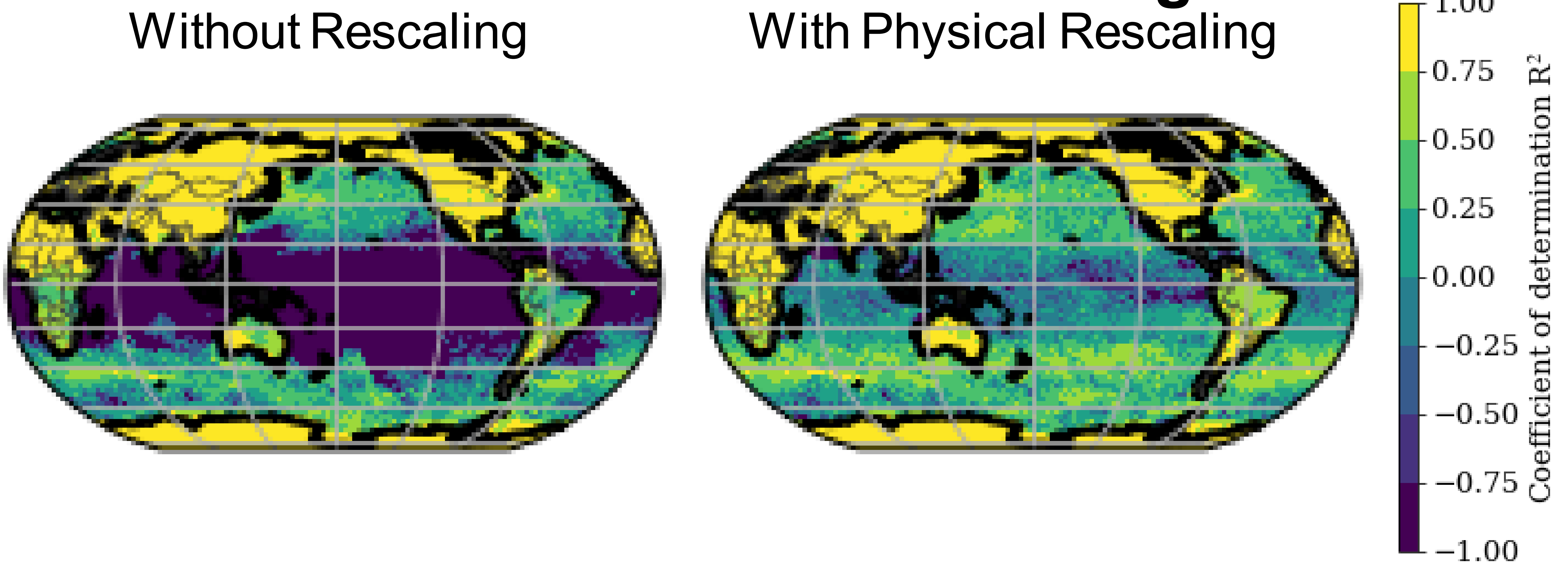


Problem 3: Physically Rescaling Inputs allows NNs to generalize from cold to warm climate



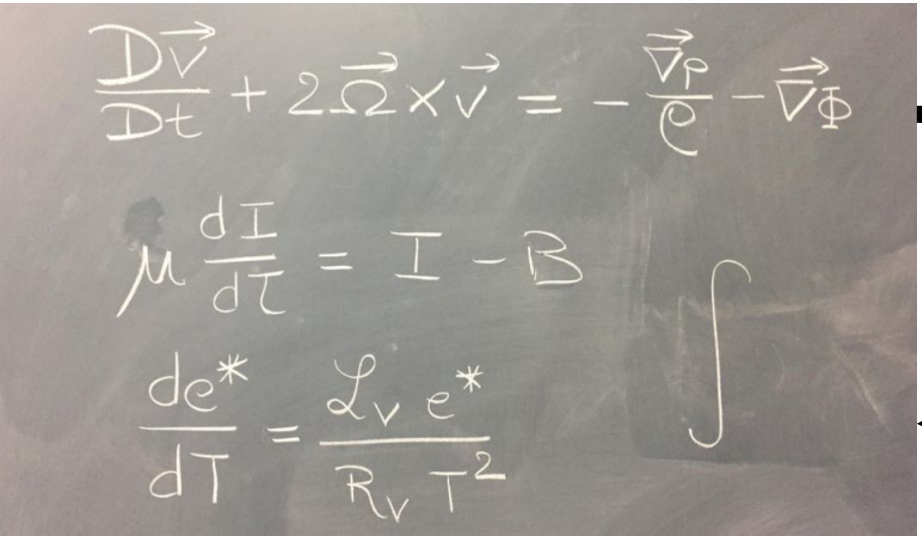


Physically-Rescaled Neural Networks Generalize Better Across Climates **in Earth-like configurations**



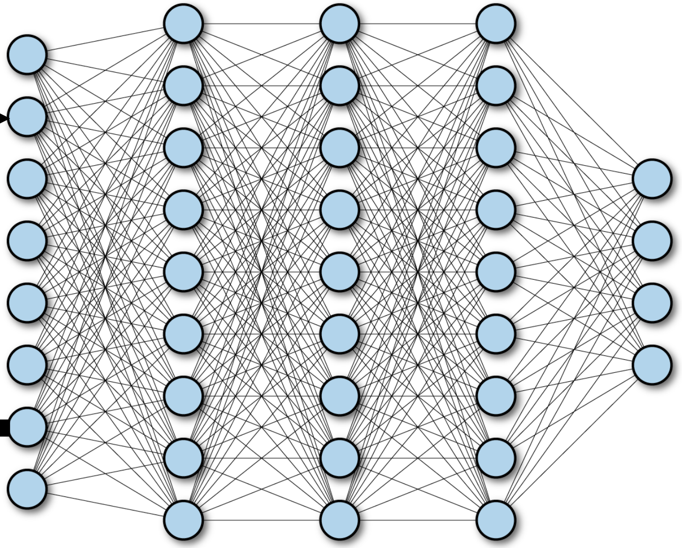
Near-Surface Subgrid Heating

Outlook 1: Extracting Physics from Data



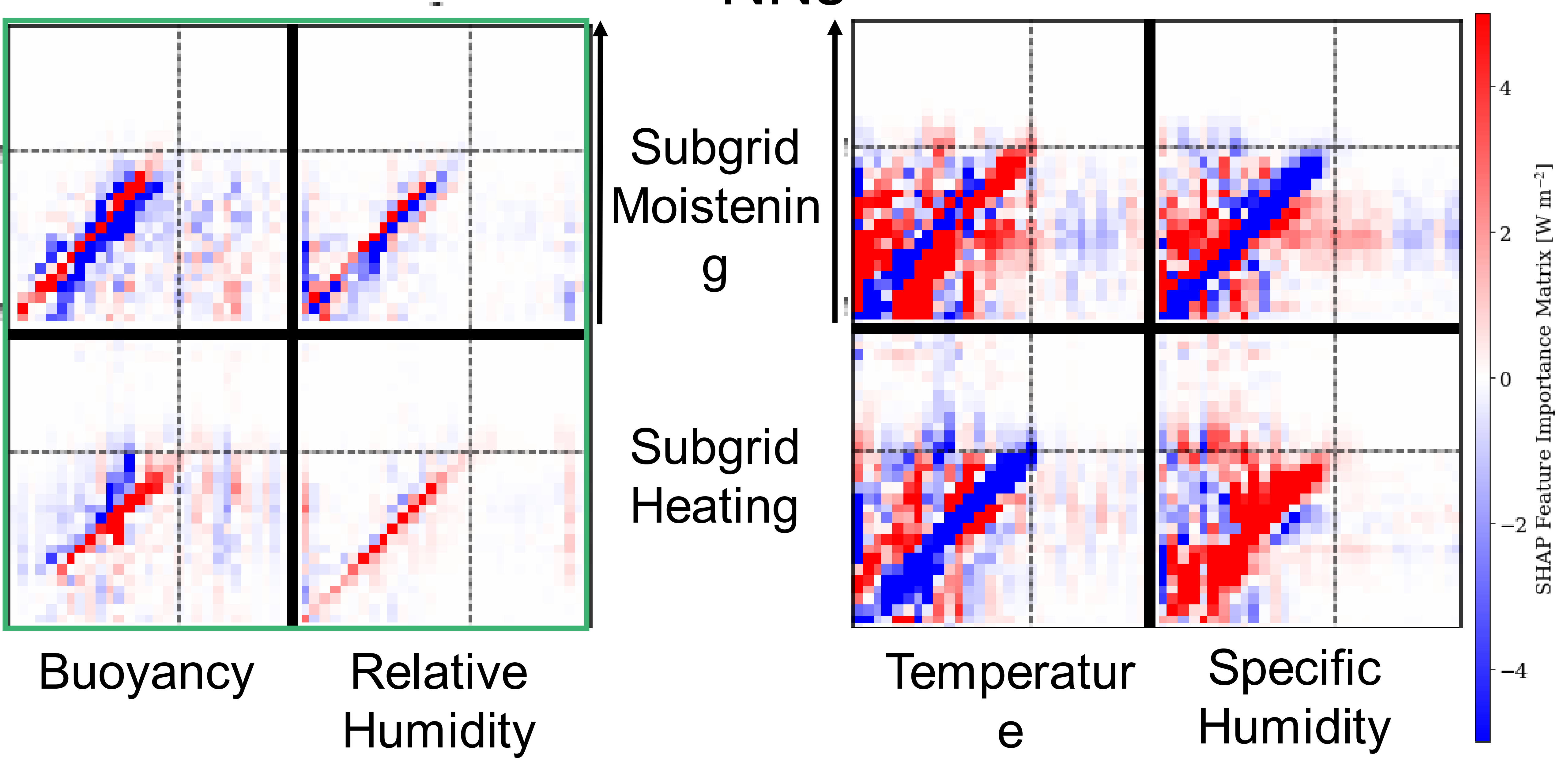
Incorporate physical knowledge into ML

Use ML to extract physical knowledge from data

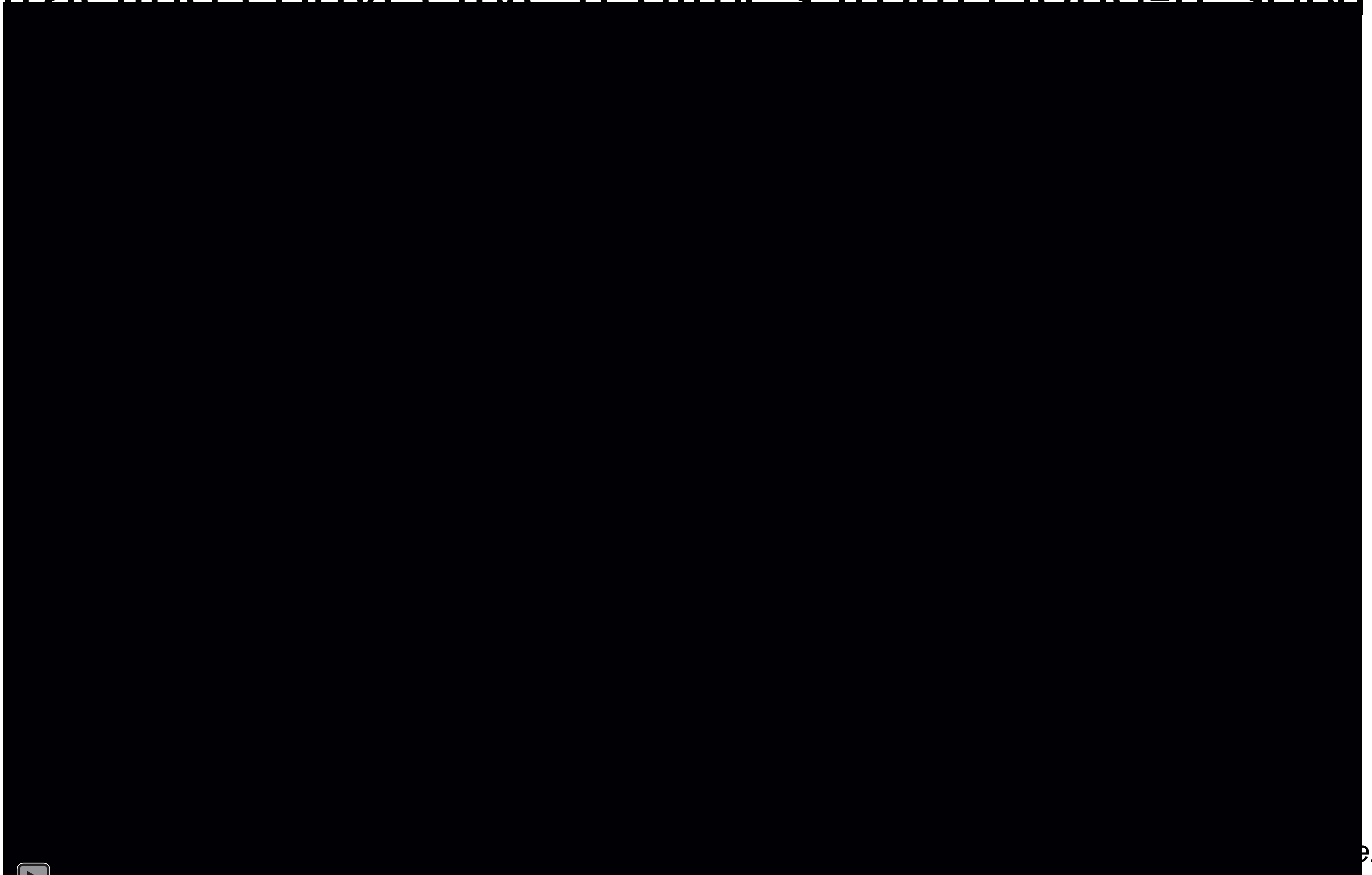


<u>See</u>: Barnes & Ebert-Uphoff (2020)

Climate-invariant NNs more local than Brute-Force NNs

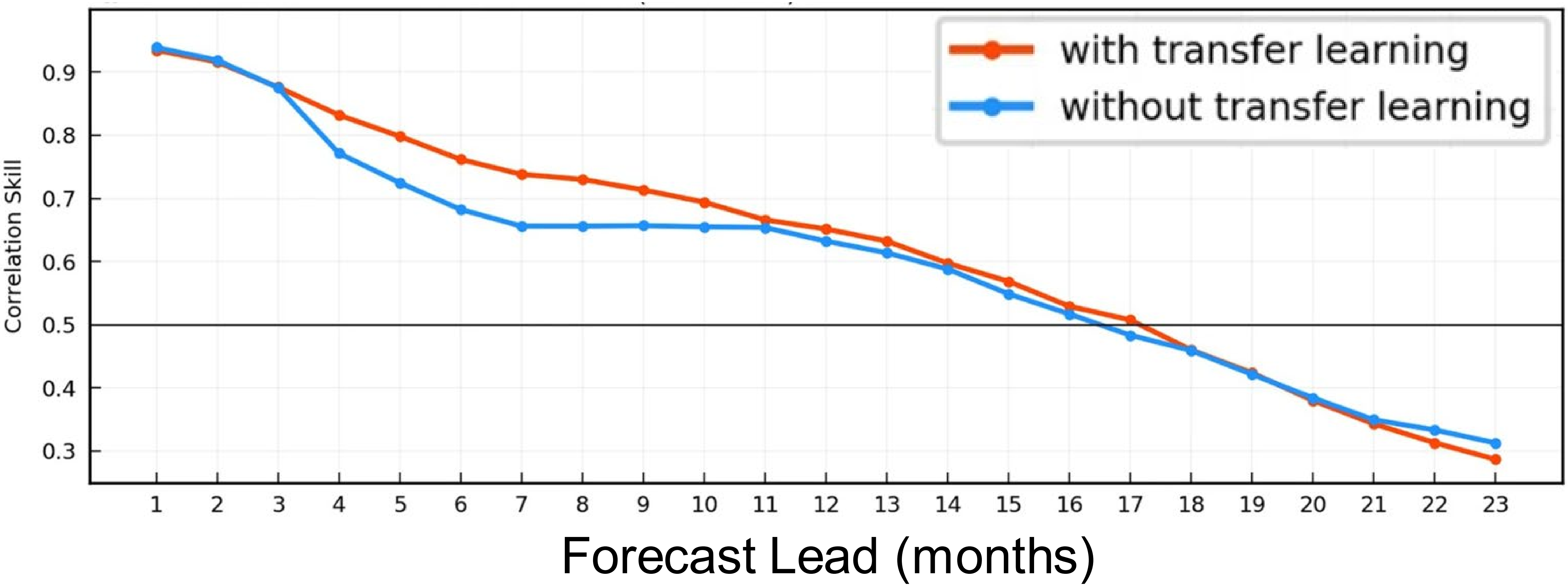


Extracting convective regimes from cloud-resolving



...er (2020)

Outlook 2: Transferring knowledge across climates/geographies/models/observations



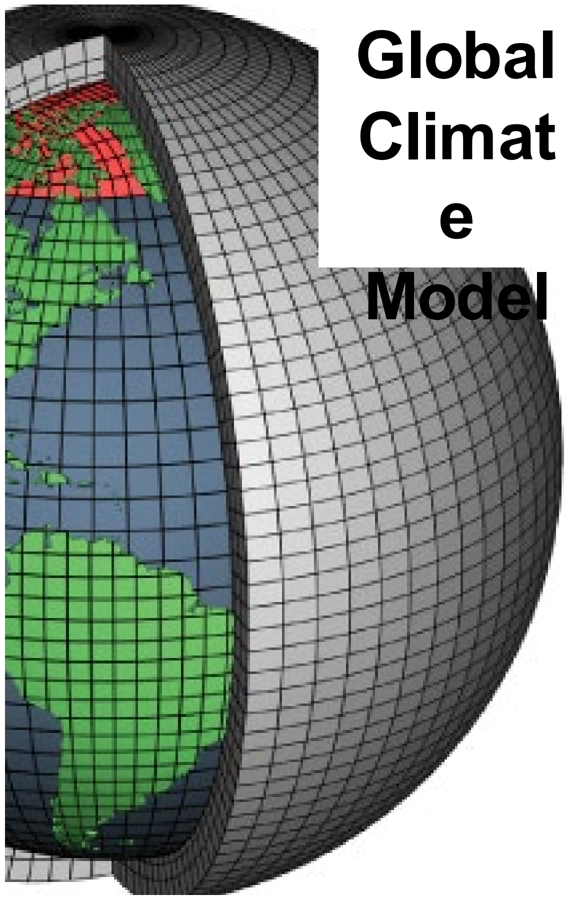
Adapted from: *Ham et al. (2019), See: Barnes et al. (2020), Rasp & Thuerey (2021)*

Problem: Observations of convection are sparse

Global Observing System

Specific humidity
(kg/kg)

Temperature
(K)



Global Climat
e
Model

Images: NASA, NOAA

Problem: Observations of convection are sparse

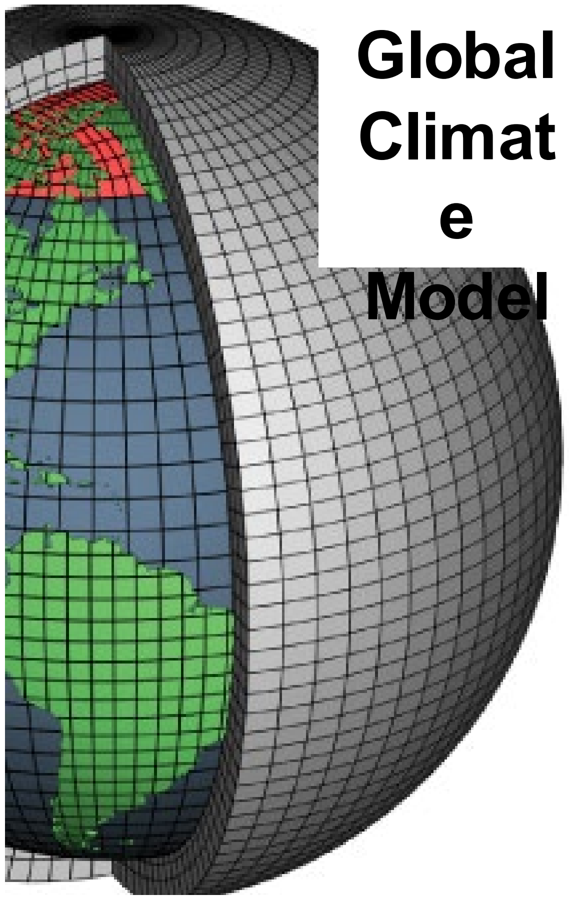
Global Observing System

Moistening tendency (W/m^2)

Heating tendency (W/m^2)

? 🤔 ?

Global Climate Model

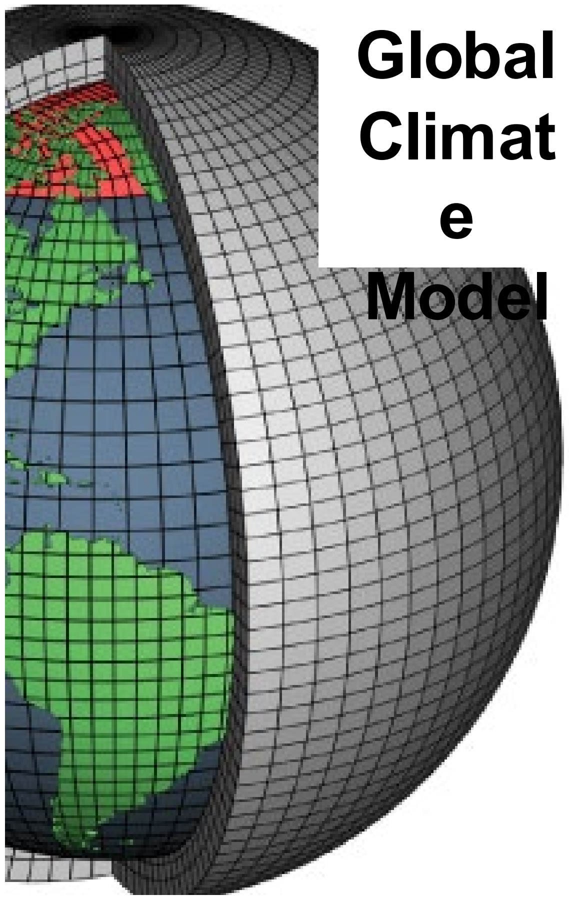


Images: NASA, NOAA

Problem: Observations of convection are sparse

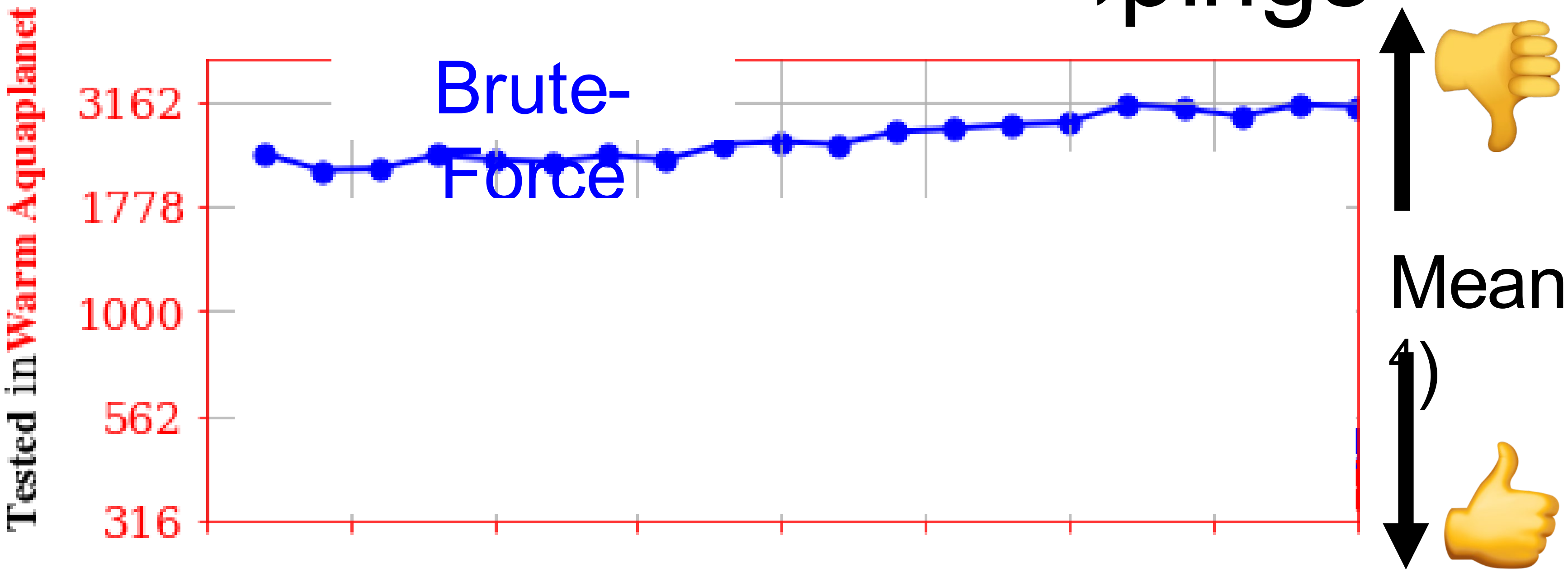
Moistening tendency (W/m^2)

Heating tendency (W/m^2)



Images: EUREC^{4}A, NOAA

Climate-Invariant NNs learn transferable mappings

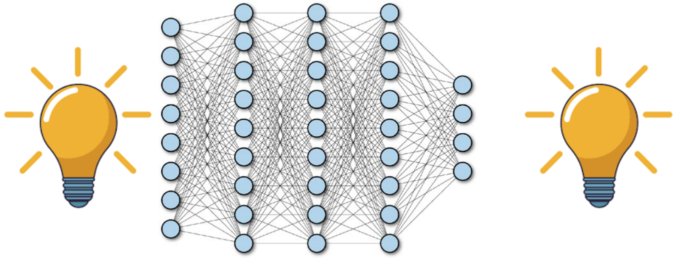


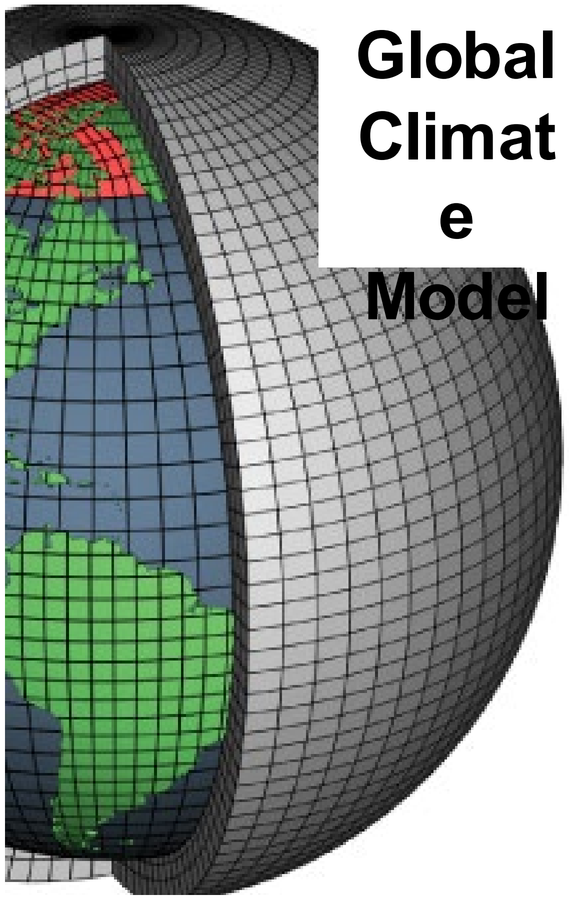
Including from Aquaplanet to Earth-like simulations!

Outlook 2: Physics-informed ML may assist the data assimilation of sparse observations

Moistening tendency (W/m^2)

Heating tendency (W/m^2)





Images: EUREC^{4}A, NOAA

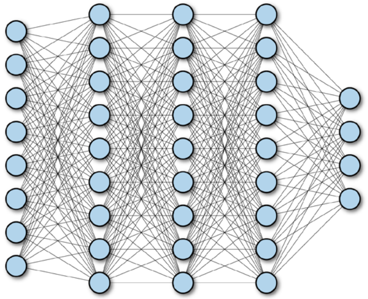
Atmospheric Physics can Help Machine Learning

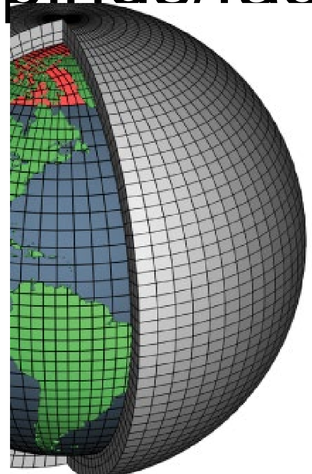
1) Enforce physical constraints approx. (loss) or exactly (architecture)

2) Tailor ML interpretability methods for emulation of physical processes

3) Help NNs generalize by physically rescaling inputs & outputs

4) Rescaled ML learns more general mappings/facilitates transfer learning





Images: NASA, NOAA

Thank you

UNIL | Université de Lausanne

∂^3AWN

∂ata-∂riven Atmospheric & Water ∂yNamics

COLUMBIA UNIVERSITY

www.unil.ch/dawn

tom.beucler@unil.ch

GitHub

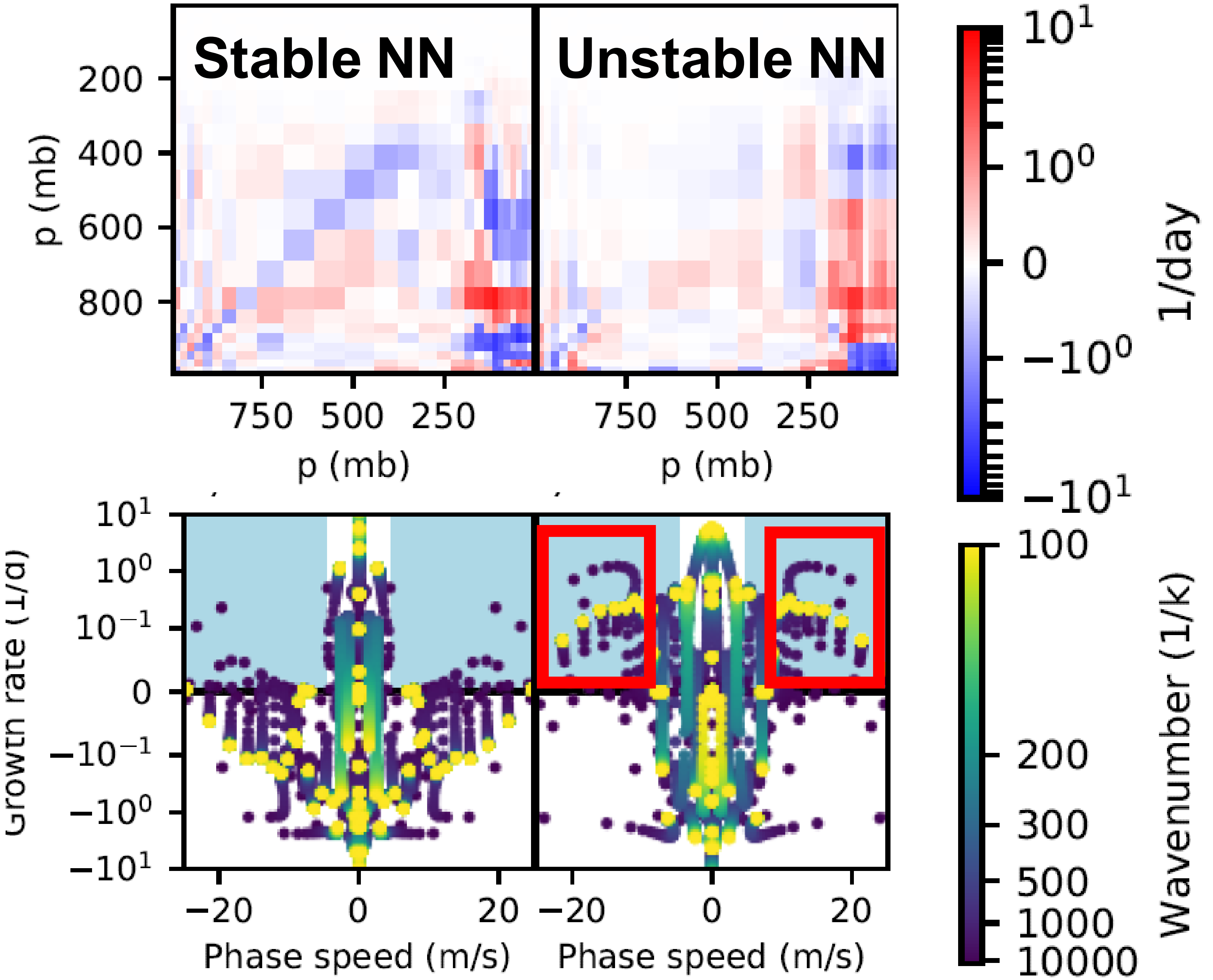
arXiv

Bonus Slides

Summary

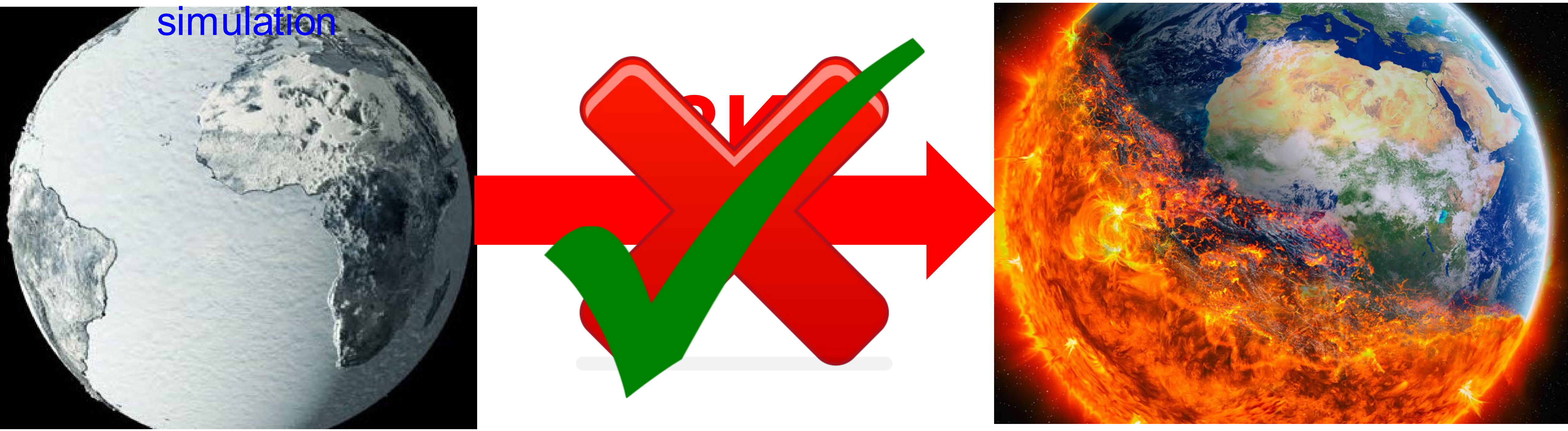
Linear Response Function

Stability Diagram (Offline)

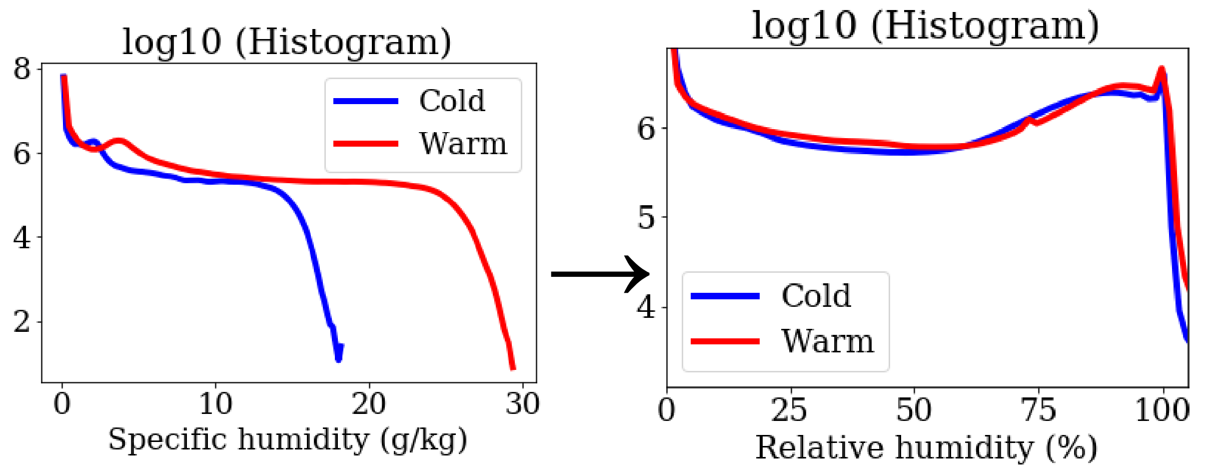


Training/Validation on
cold aquaplanet
simulation

Test on
warm aquaplanet simulation



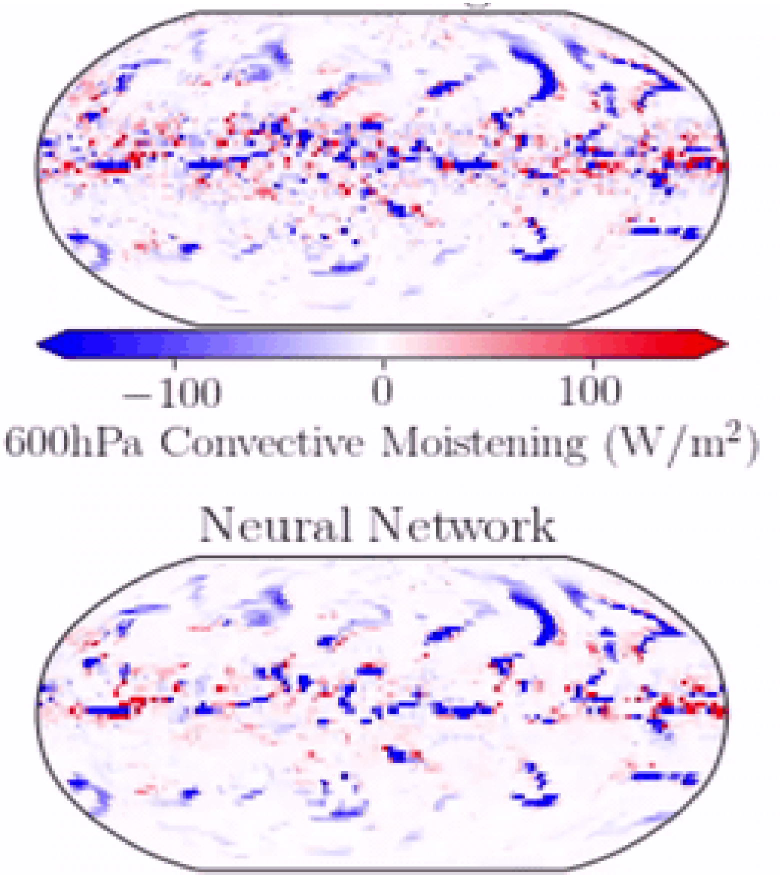
Climate-Invariant nets: Rescale inputs/outputs so that (extrapolation)→(interpolation)



Climate-Invariant neural networks:

- Learn more general mappings

- Facilitate transfer learning

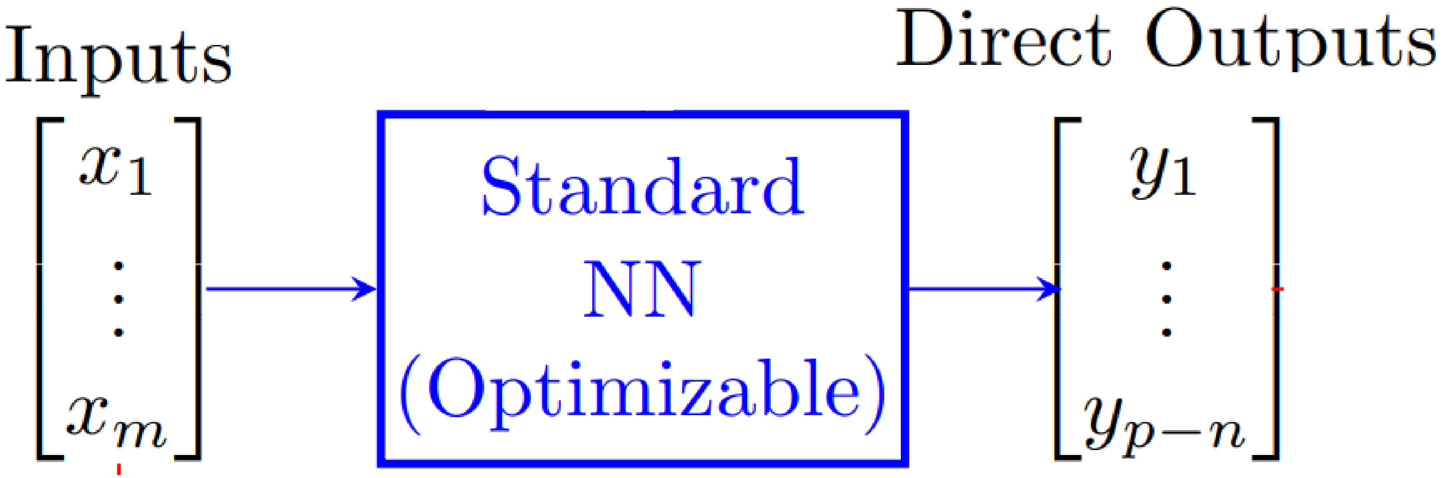


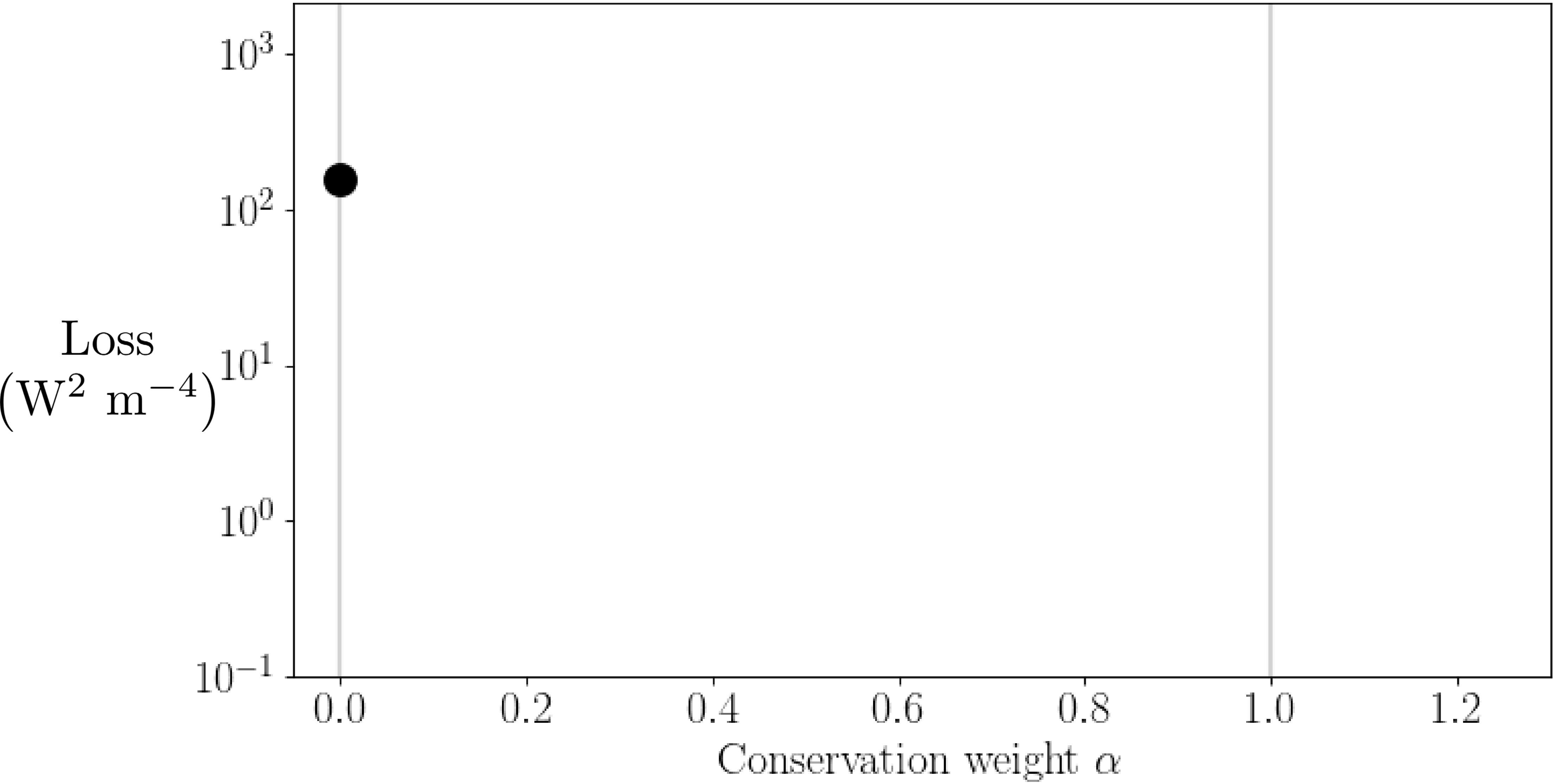
Soft Constraints (Loss) vs Hard Constraints (Architecture)

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

$$\boxed{\text{Loss} = \alpha\,(\text{Squared Residual from conservation laws}) + (1-\alpha)\,(\text{Mean squared error})}$$

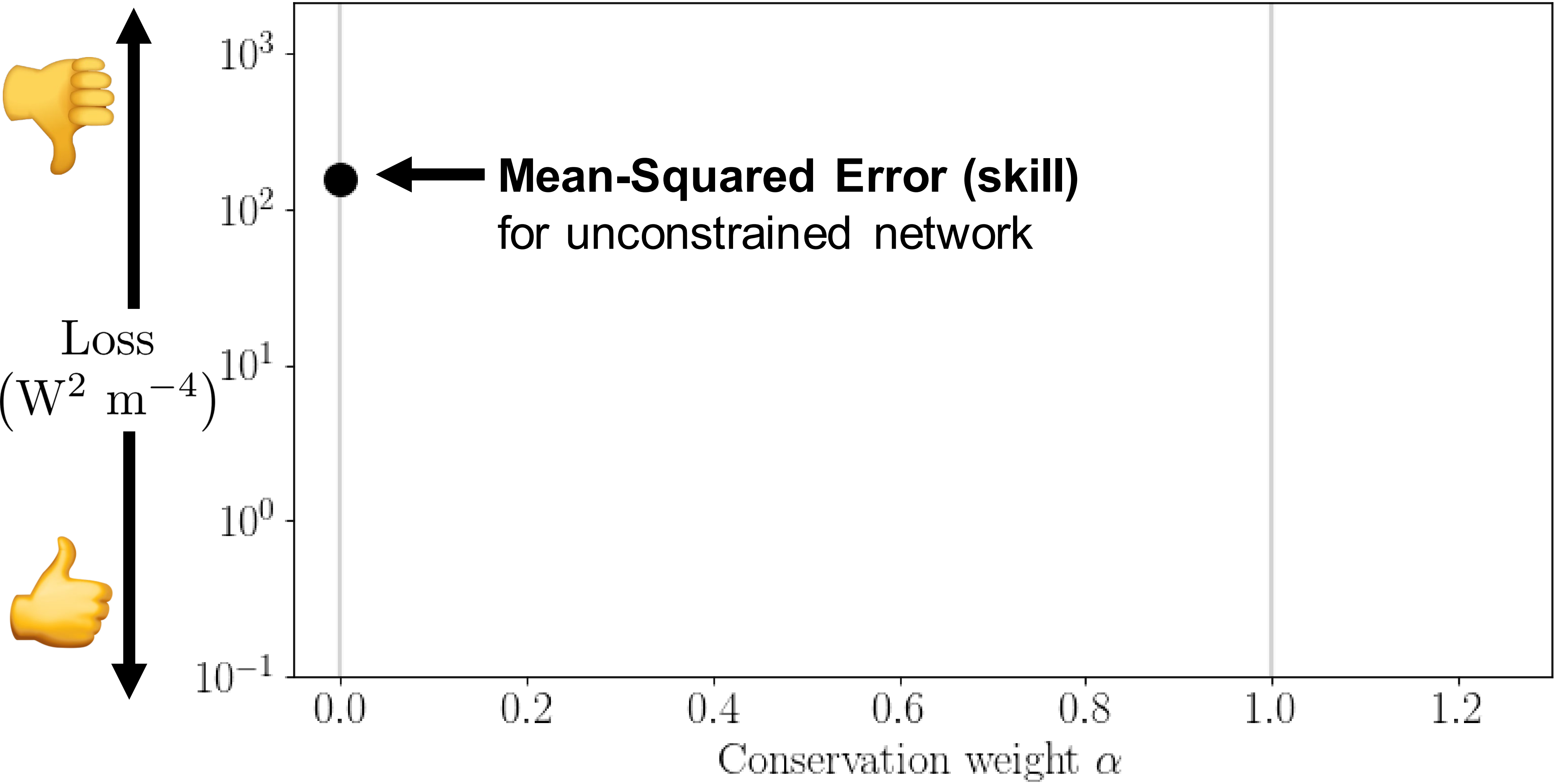
Architecture: Constraints layers to enforce conservation laws to machine precision





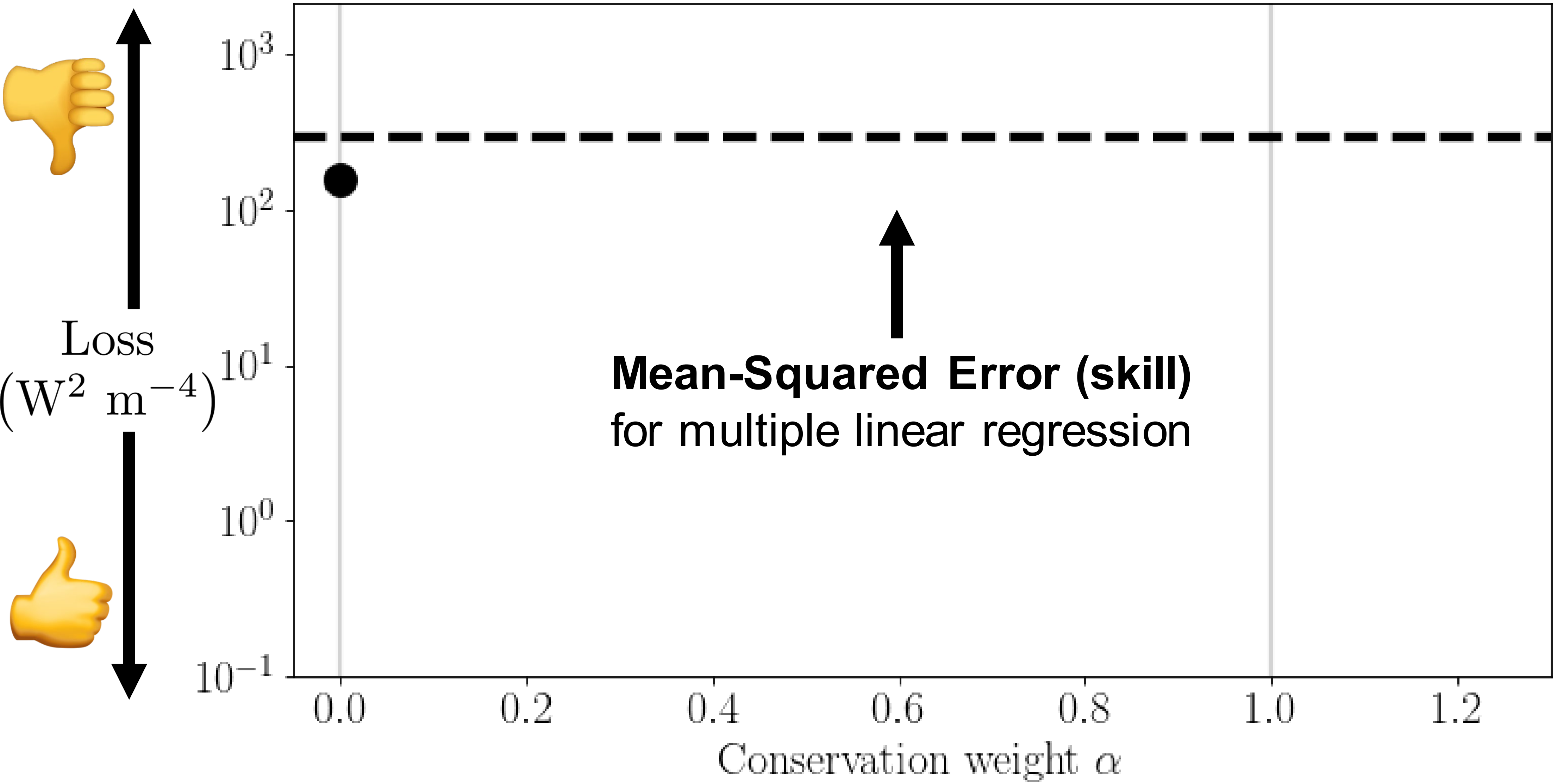
Loss: Trade-off between **physical constraints** and **performance**

$$\text{Loss} = \alpha\ \boxed{(\text{Squared Residual from conservation laws})} + (1-\alpha)\ \boxed{(\text{Mean squared error})}$$



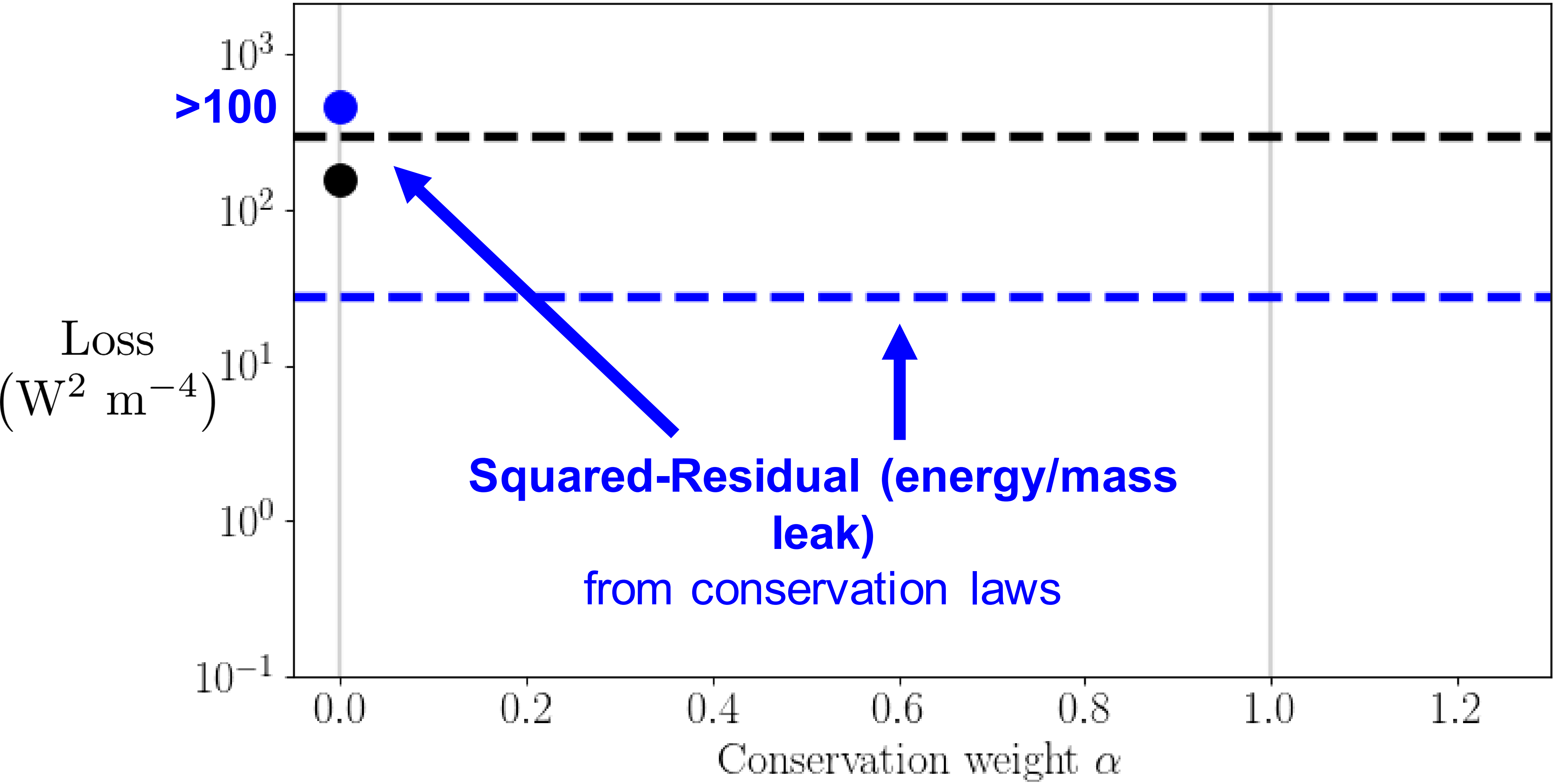
<u>Loss</u>: Trade-off between **physical constraints** and **performance**

$$\text{Loss} = \alpha\,\boxed{\text{(Squared Residual from conservation laws)}} + (1-\alpha)\,\boxed{\text{(Mean squared error)}}$$



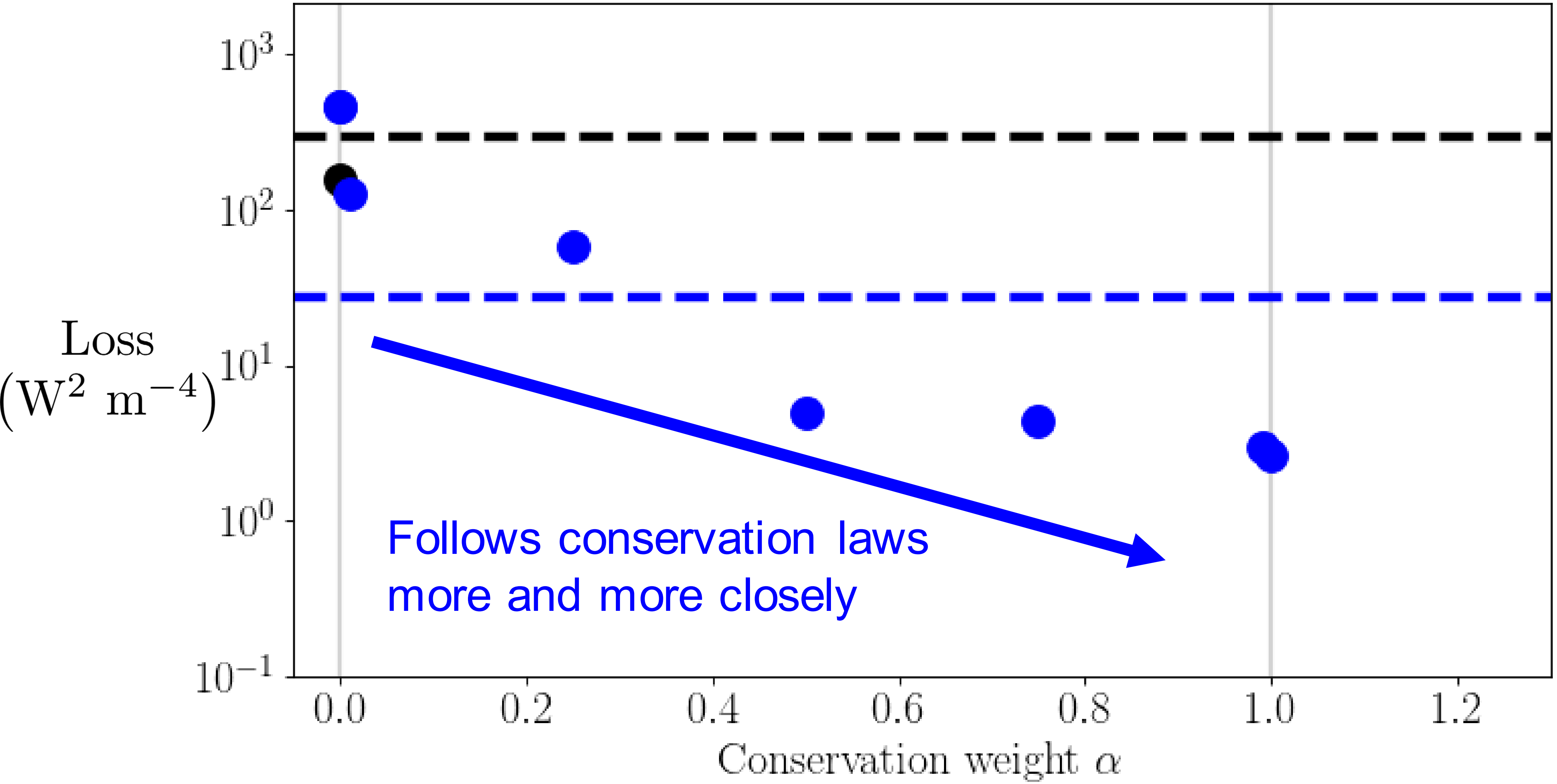
<u>Loss</u>: Trade-off between **physical constraints** and **performance**

$$\text{Loss} = \alpha \boxed{\text{(Squared Residual from conservation laws)}} + (1-\alpha) \boxed{\text{(Mean squared error)}}$$



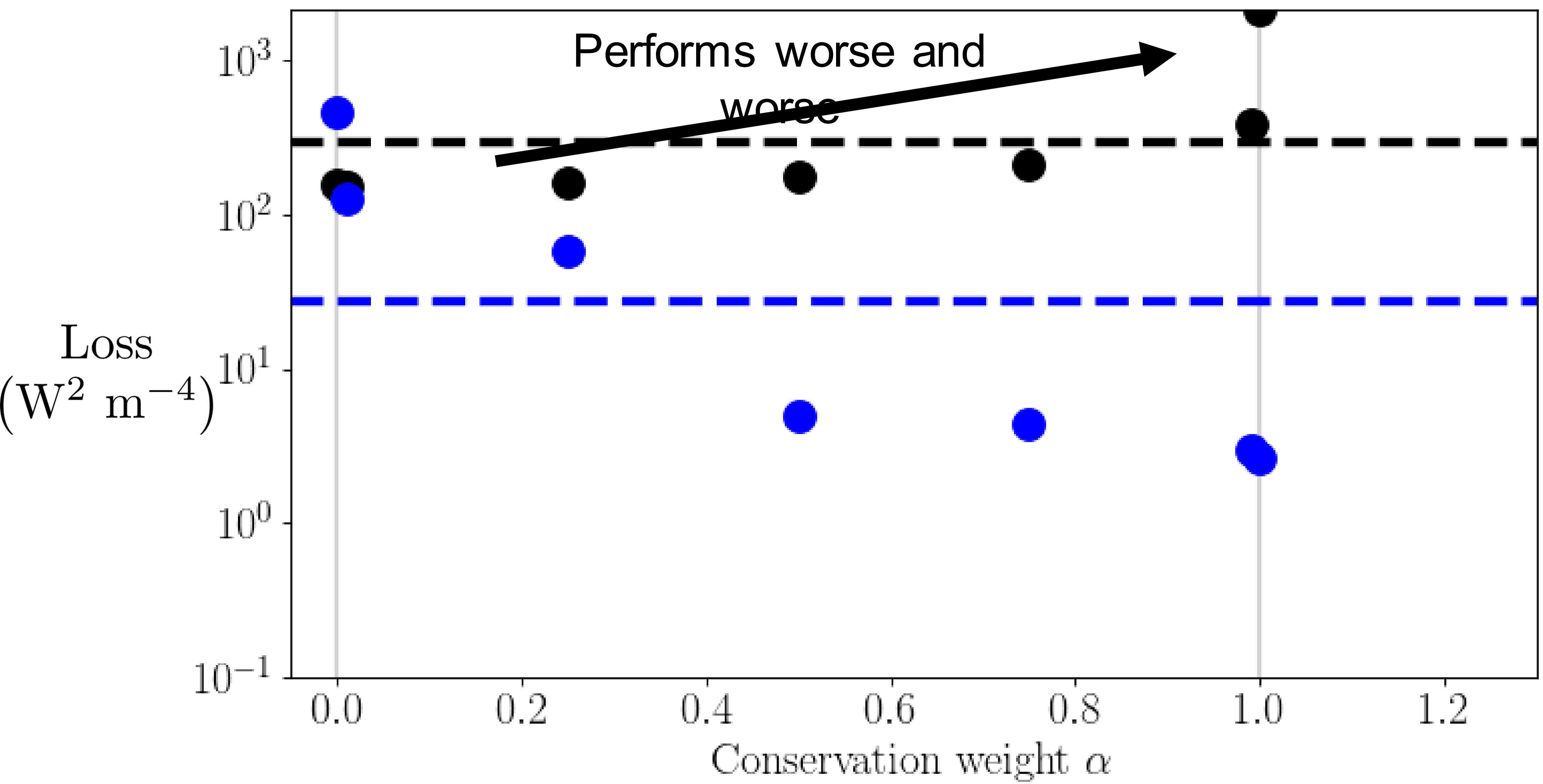
<u>Loss</u>: Trade-off between **physical constraints** and **performance**

$$\text{Loss} = \alpha \boxed{(\text{Squared Residual from conservation laws})} + (1-\alpha) \boxed{(\text{Mean squared error})}$$



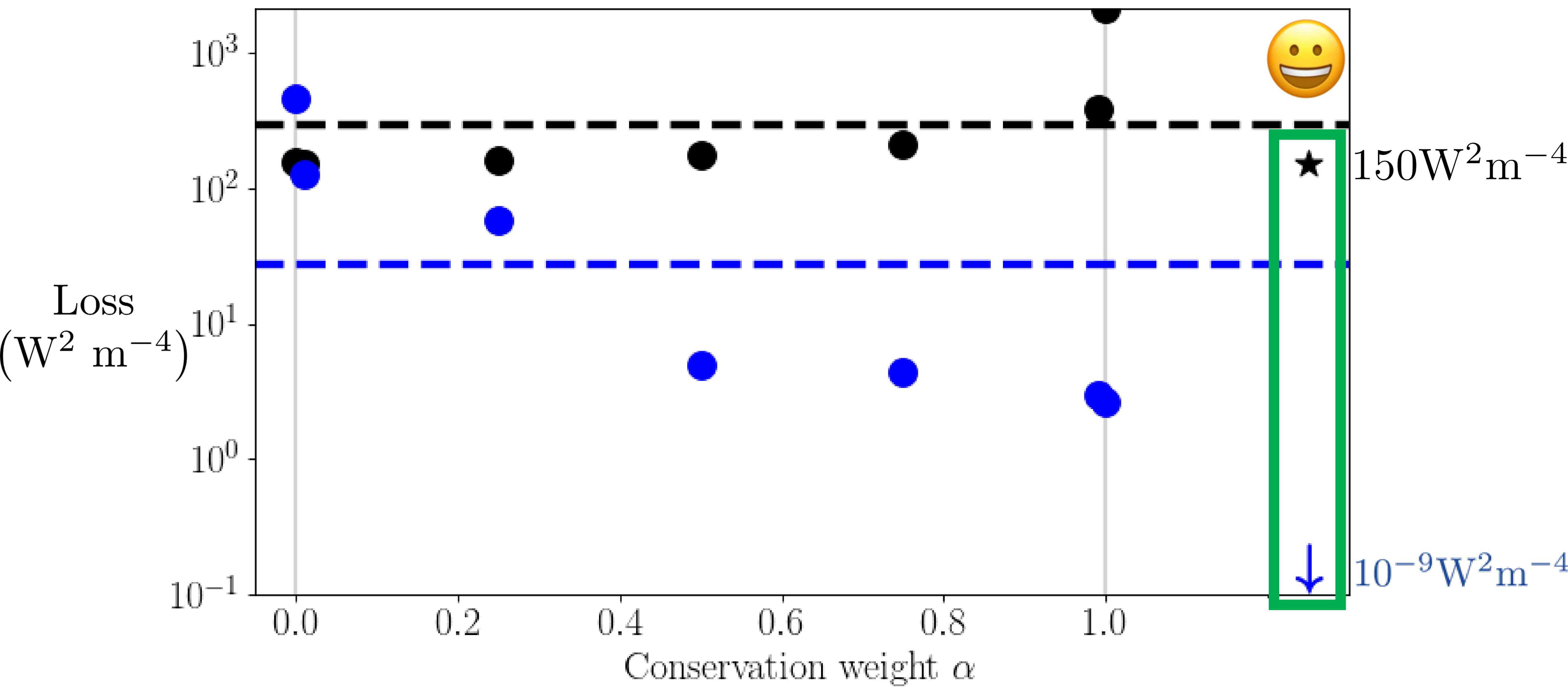
<u>Loss</u>: Trade-off between **physical constraints** and **performance**

$$\text{Loss} = \alpha \boxed{(\text{Squared Residual from conservation laws})} + (1-\alpha) \boxed{(\text{Mean squared error})}$$



Loss: Trade-off between **physical constraints** and **performance**

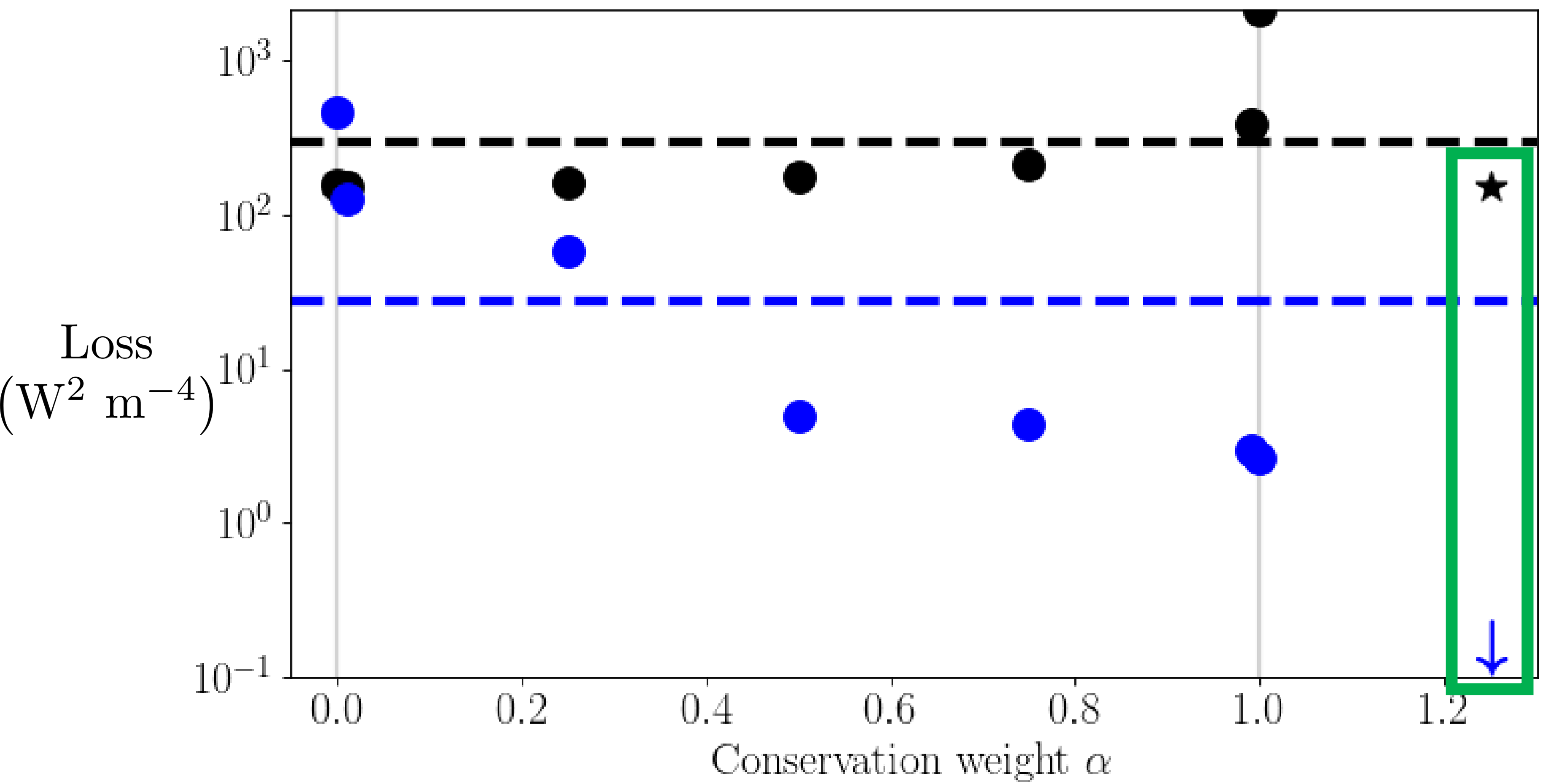
$$\text{Loss} = \alpha \boxed{(\text{Squared Residual from conservation laws})} + (1-\alpha) \boxed{(\text{Mean squared error})}$$



Loss: Trade-off between **physical constraints** and **performance**

Architecture: **Constraints enforced & competitive performance**

See: Beucler et al. (2019)



Problem 2: Even when physically constrained, NNs fail to generalize

Algorithms: Custom Data Generators/Layers

$$\underset{\text{Inputs}}{\begin{bmatrix} q_{v,1} \\ q_{v,2} \\ \vdots \\ \mathrm{SHF} \\ \mathrm{LHF} \end{bmatrix}} \longrightarrow \boxed{\mathrm{NN}} \longrightarrow \underset{\text{Outputs}}{\begin{bmatrix} \dot{q}_{v,1} \\ \dot{q}_{v,2} \\ \vdots \\ P \\ P_i \end{bmatrix}}$$

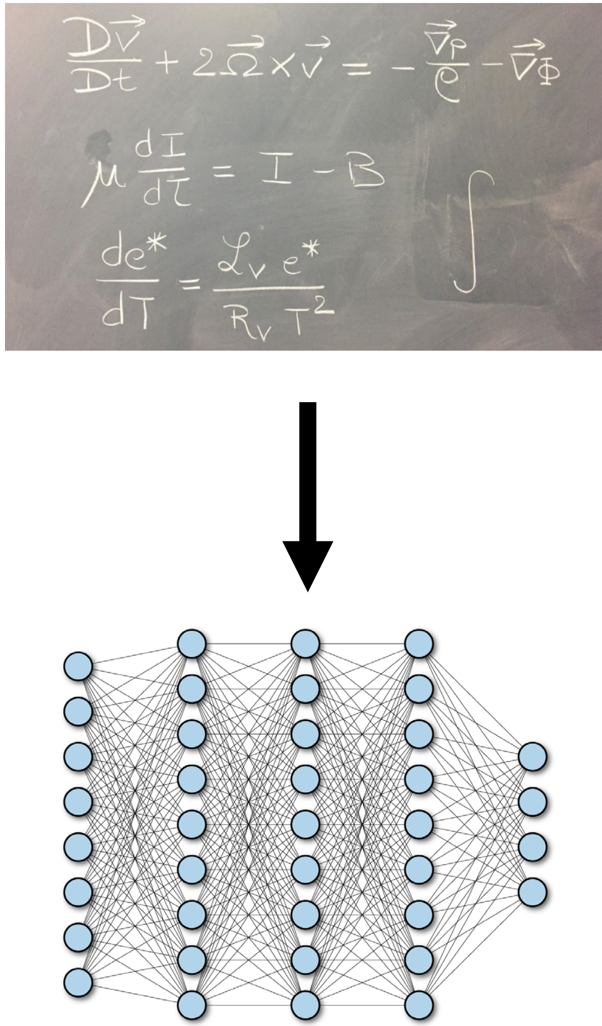
- Only one training/validation/test data despite multiple rescalings
- Test different rescalings quickly using multi-linear/logistic regressions
- Keep the rescalings that yield the best generalization

Start with clear link to climate impact/remote sensing

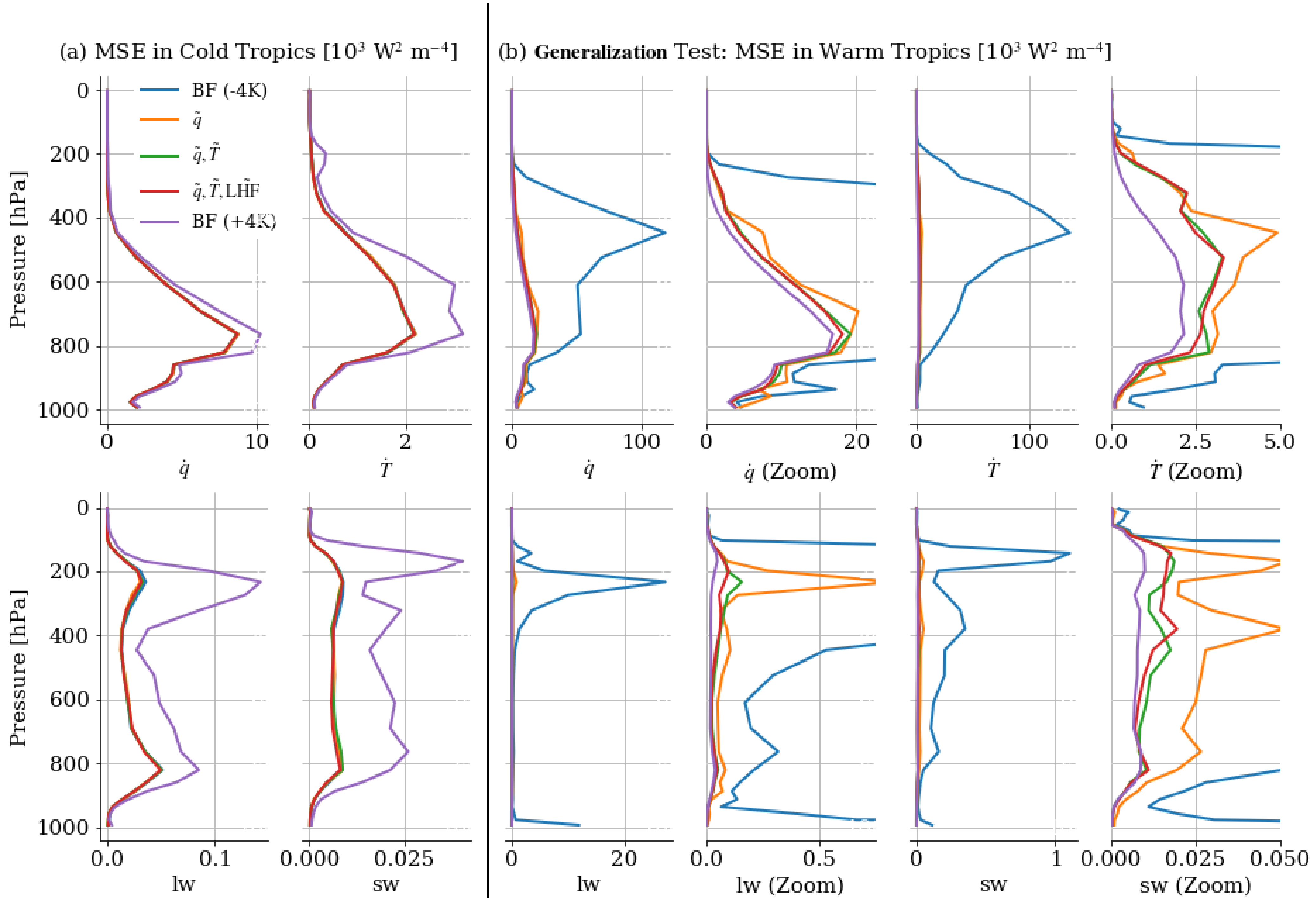
Link = Transfer Learning

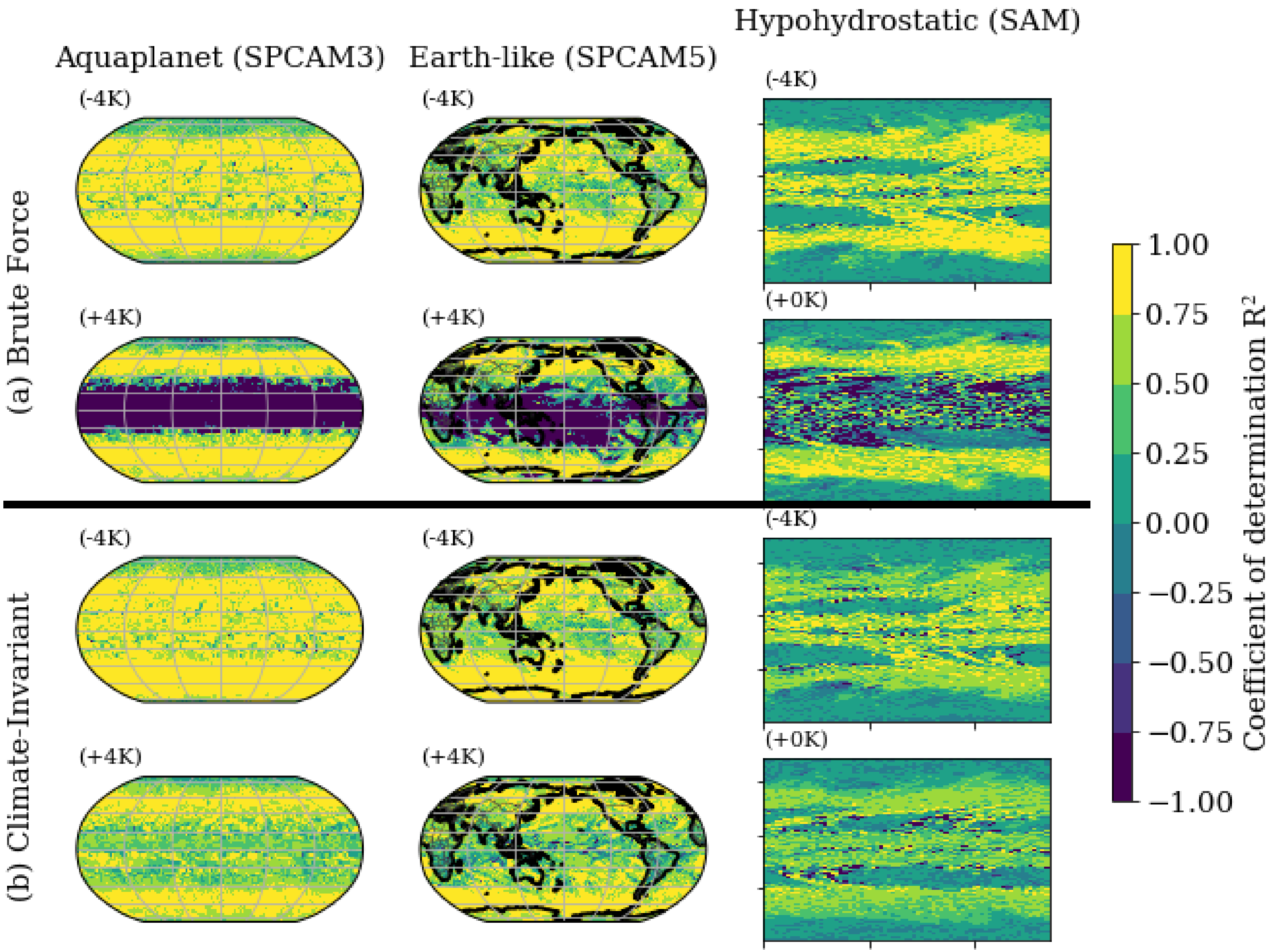
Why Integrate Physics into ML/Stat Algorithms?

- Physical consistency
  (definitions, conservation laws…)
- Ability to generalize outside of the training set
- Interpretability
- Stability
- Data limitations

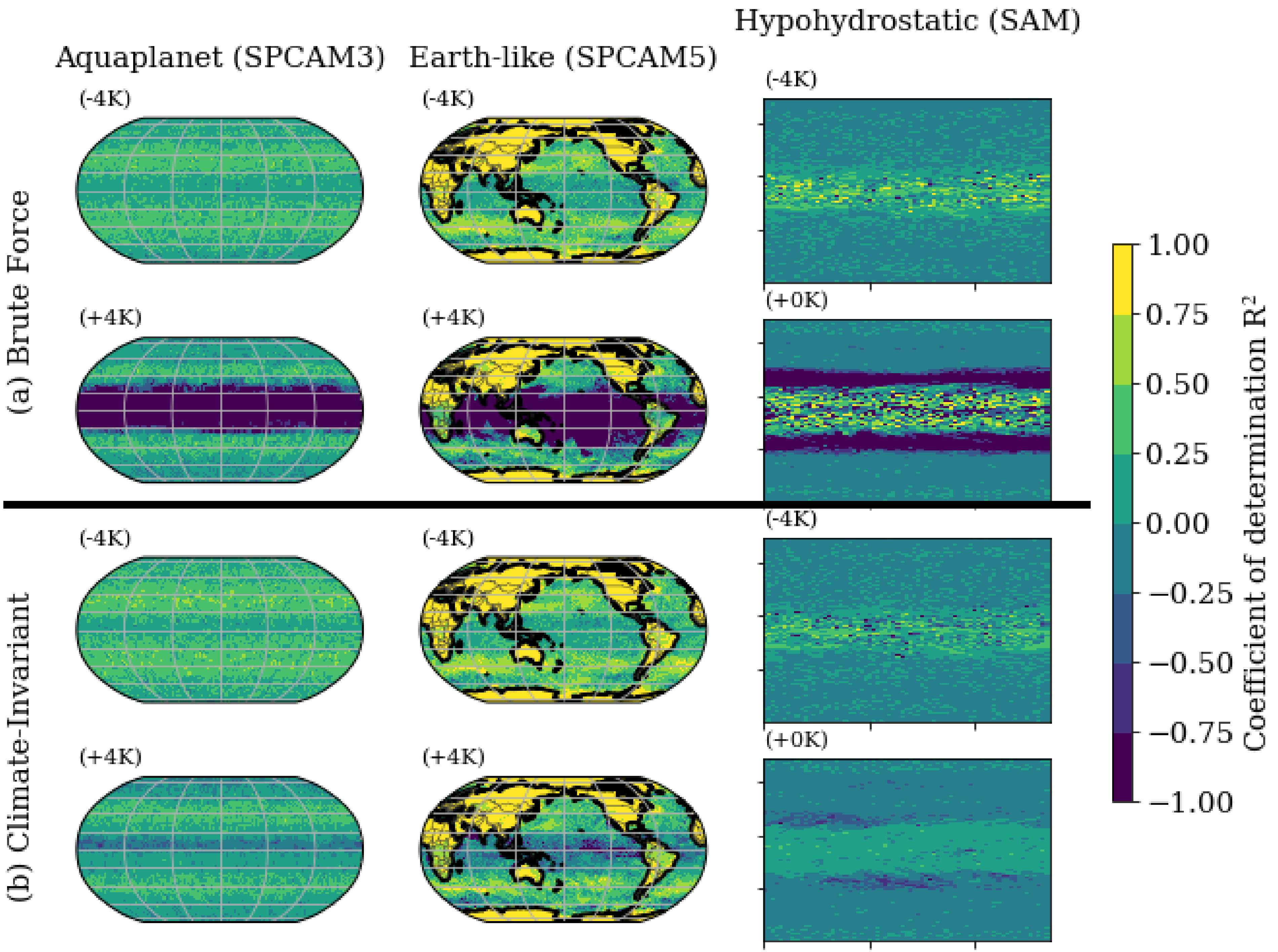


Reviews: Willard et al. (2020), Reichstein et al. (2019), Karpatne et al. (2017), Beucler et al. (2021)

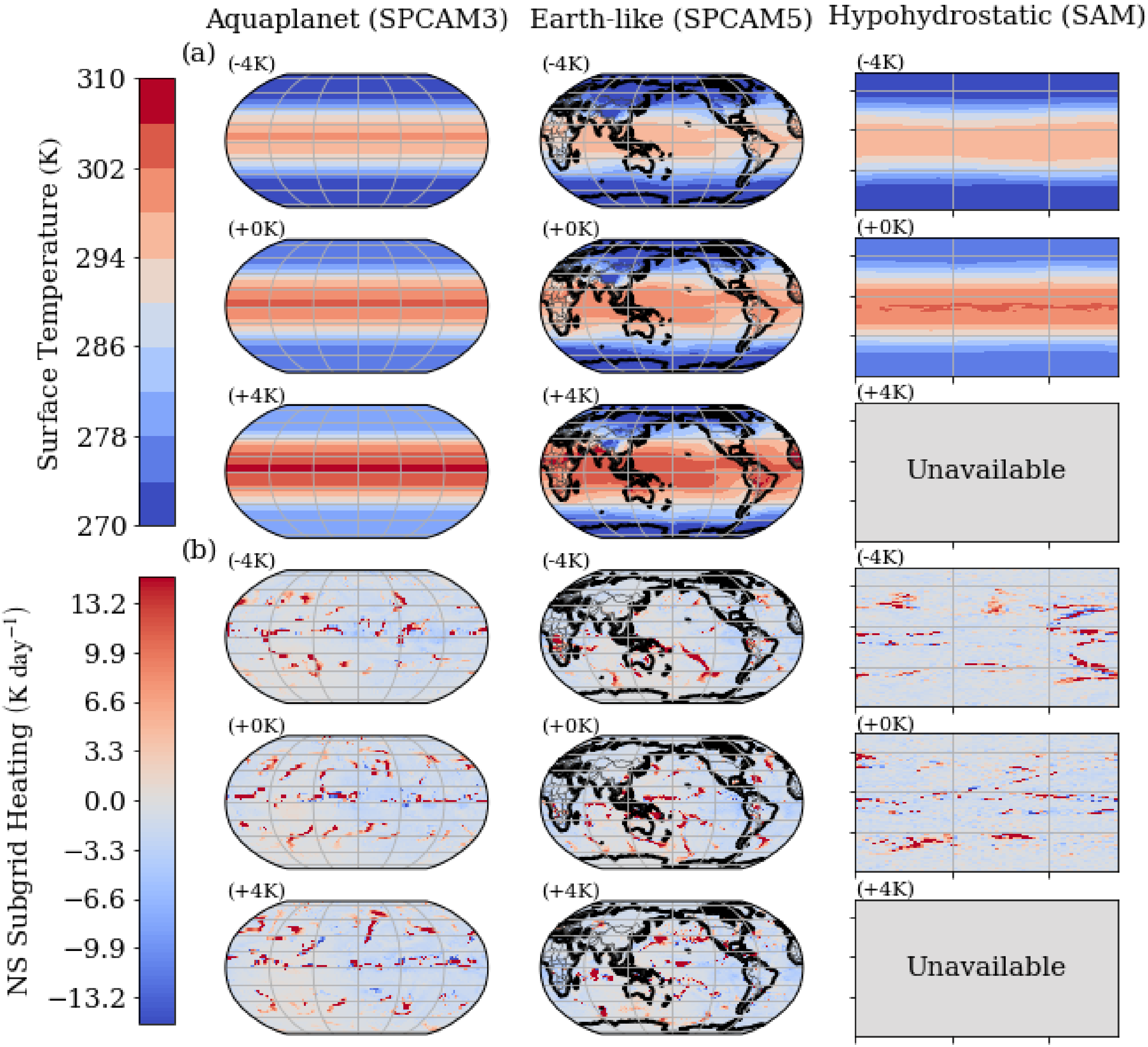

(a) MSE in Cold Tropics [10^3 W^2 m^{-4}]
(b) Generalization Test: MSE in Warm Tropics [10^3 W^2 m^{-4}]
BF (-4K)
$\tilde{q}$
$\tilde{q}, \tilde{T}$
$\tilde{q}, \tilde{T}, \tilde{\mathrm{LHF}}$
BF (+4K)
Pressure [hPa]
$\dot{q}$
$\dot{T}$
$\dot{q}$ (Zoom)
$\dot{T}$ (Zoom)
lw
sw
lw (Zoom)
sw (Zoom)



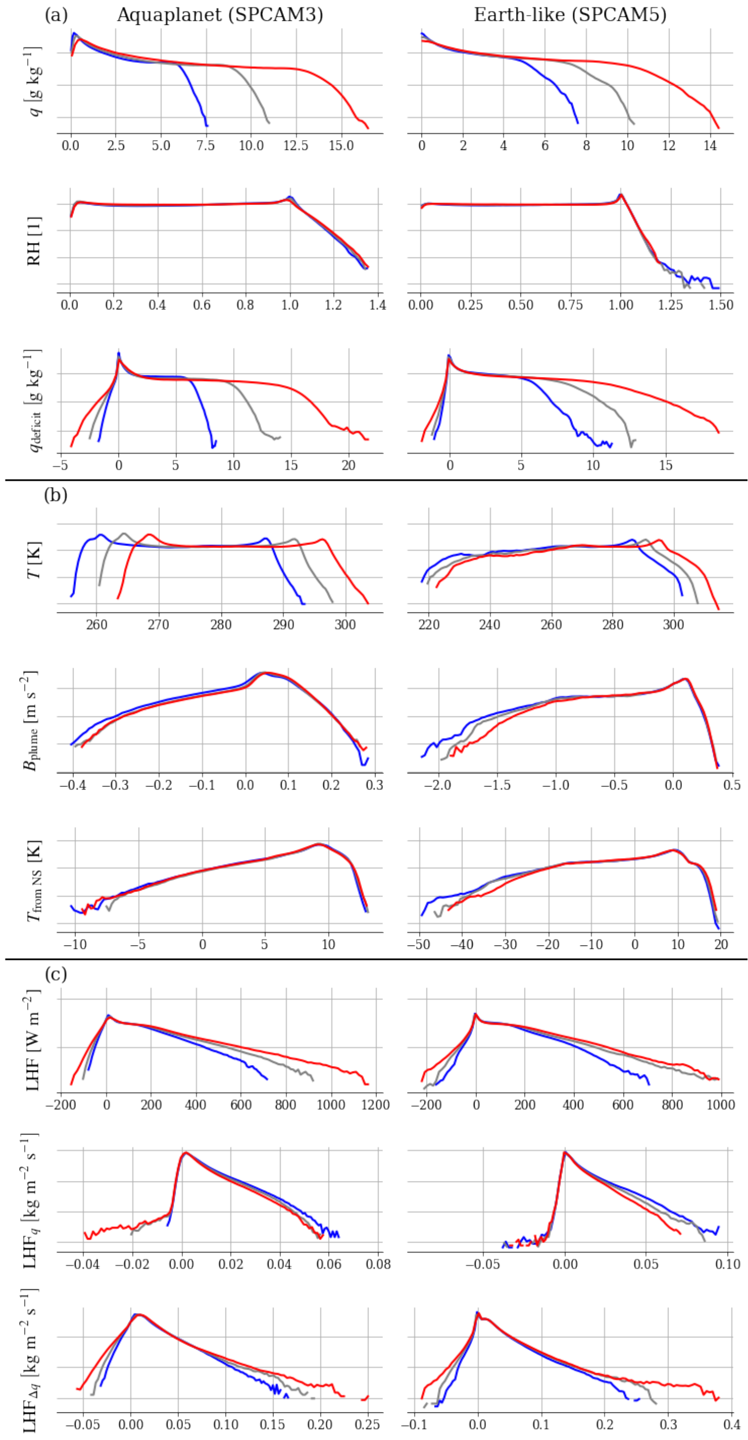
500-hPa
Subgrid
Heating



Near-surface Subgrid Heating



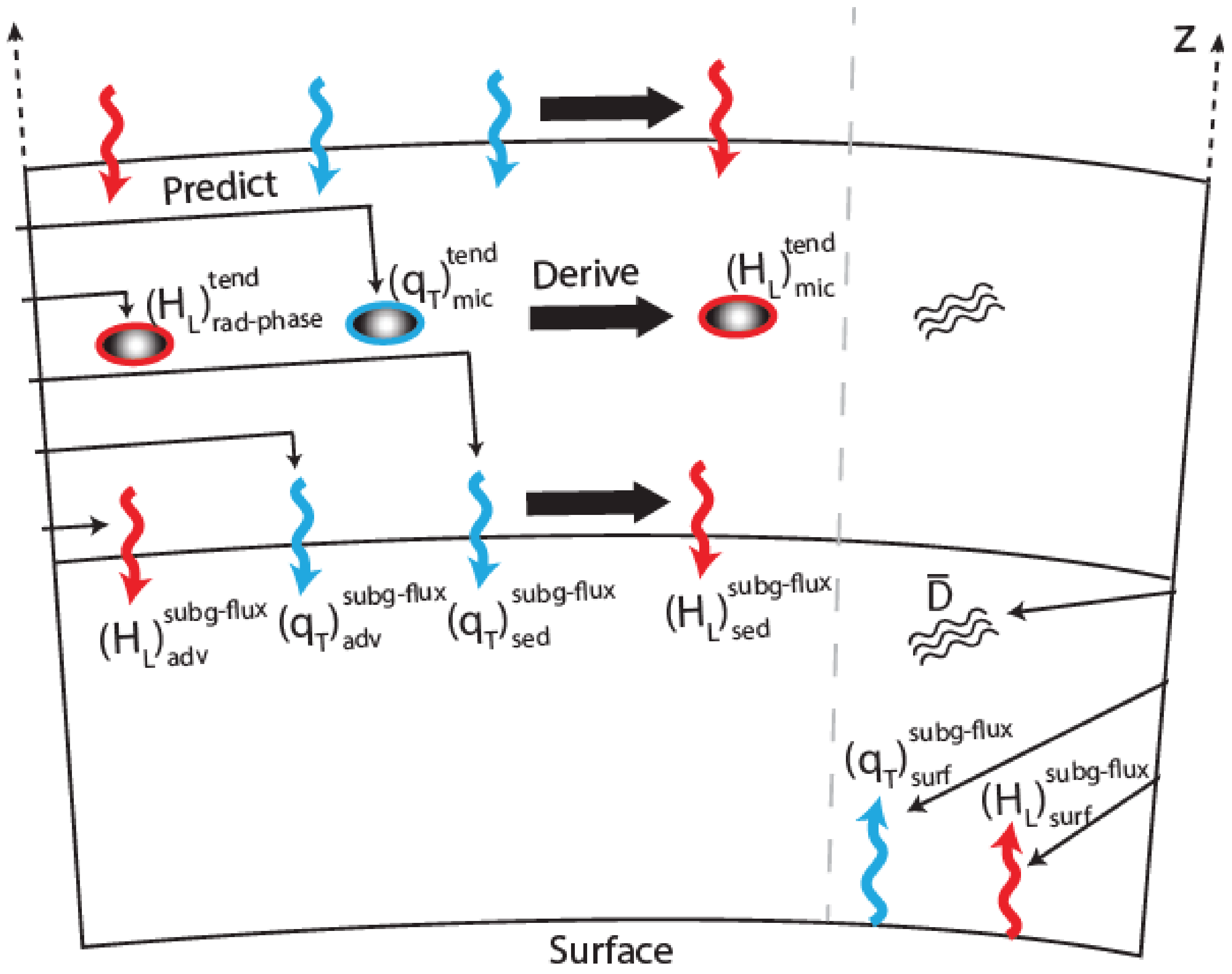
Aquaplanet (SPCAM3)
Earth-like (SPCAM5)
Hypohydrostatic (SAM)
(a)
(-4K)
(+0K)
(+4K)
Surface Temperature (K)
310
302
294
286
278
270
Unavailable
(b)
NS Subgrid Heating (K day^{-1})
13.2
9.9
6.6
3.3
0.0
−3.3
−6.6
−9.9
−13.2



(a)
Aquaplanet (SPCAM3)
Earth-like (SPCAM5)
q [g kg^{-1}]
RH [1]
q_{deficit} [g kg^{-1}]
(b)
T [K]
B_{plume} [m s^{-2}]
$T_{\mathrm{from\ NS}}$ [K]
(c)
LHF [W m^{-2}]
LHF$_q$ [kg m^{-2} s^{-1}]
LHF$_{\Delta q}$ [kg m^{-2} s^{-1}]

Limits of Physics-Guided ML

Modeling: Process-based ML stable & interpretable



<u>See</u>: *Yuval et al. (2021), Brenowitz et al. (2021)*

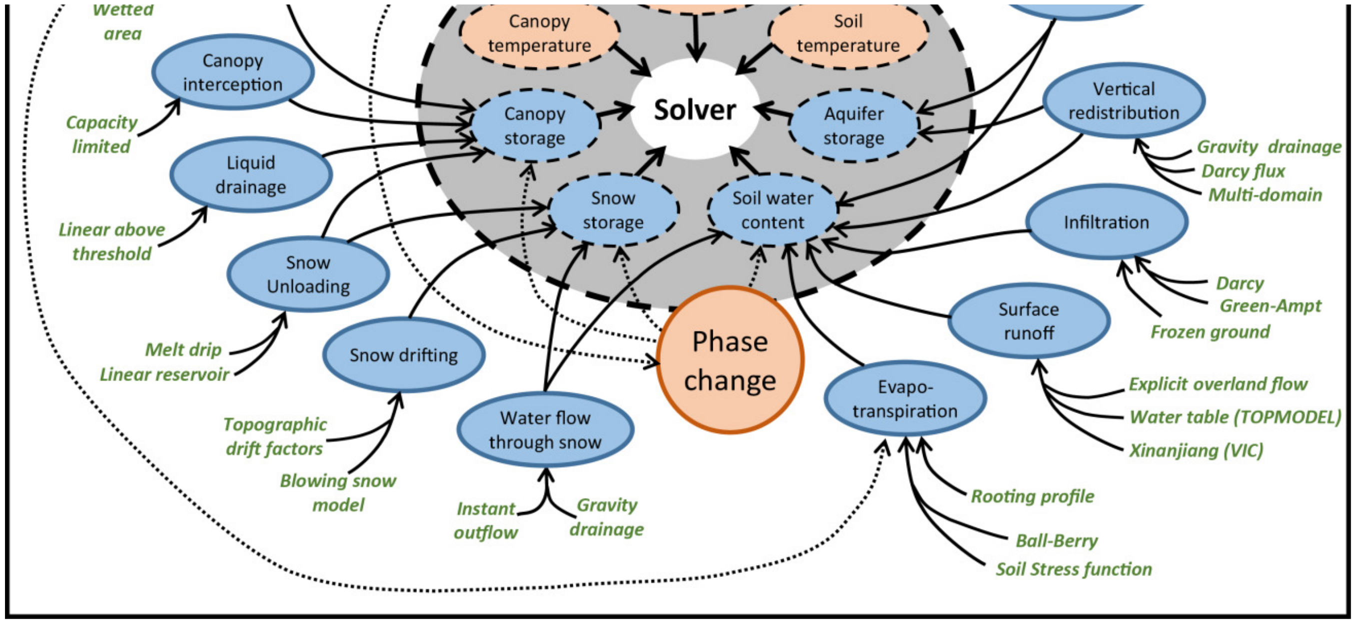
Some Limitations of Physics-Guided ML

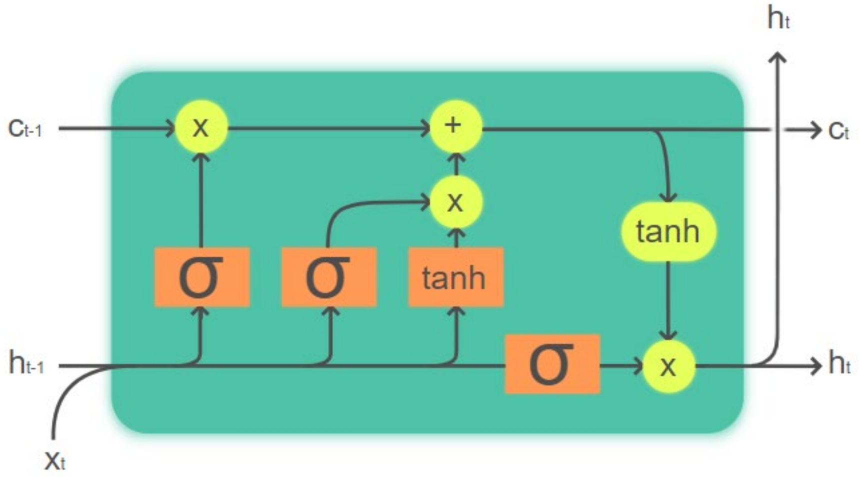
Water Resources Research

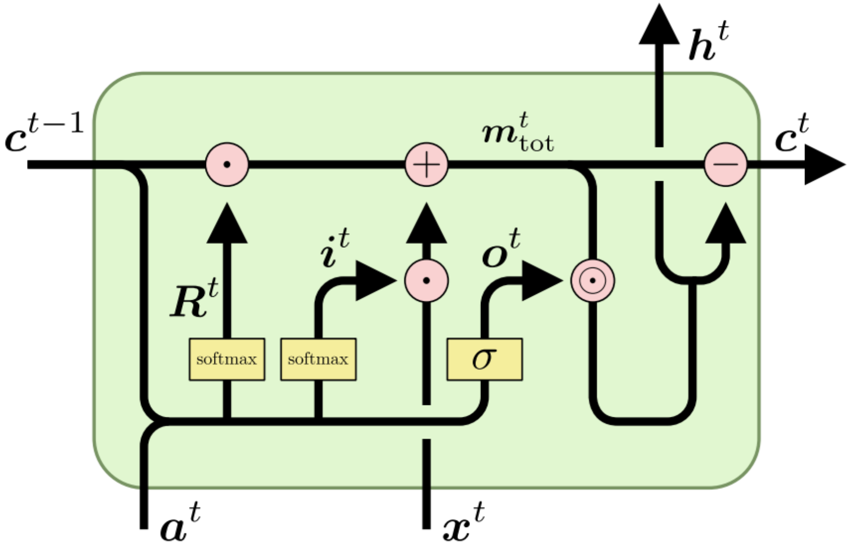
Commentary | Free Access

What Role Does Hydrological Science Play in the Age of Machine Learning?

Grey S. Nearing, Frederik Kratzert, Alden Keefe Sampson, Craig S. Pelissier, Daniel Klotz, Jonathan M. Frame, Cristina Prieto, Hoshin V. Gupta







See: Nearing et al. (2021), Clark et al. (2015), Chevalier (2018), Hoedt et al. (2021)