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

Presenter: Tom Beucler – UCI & Columbia

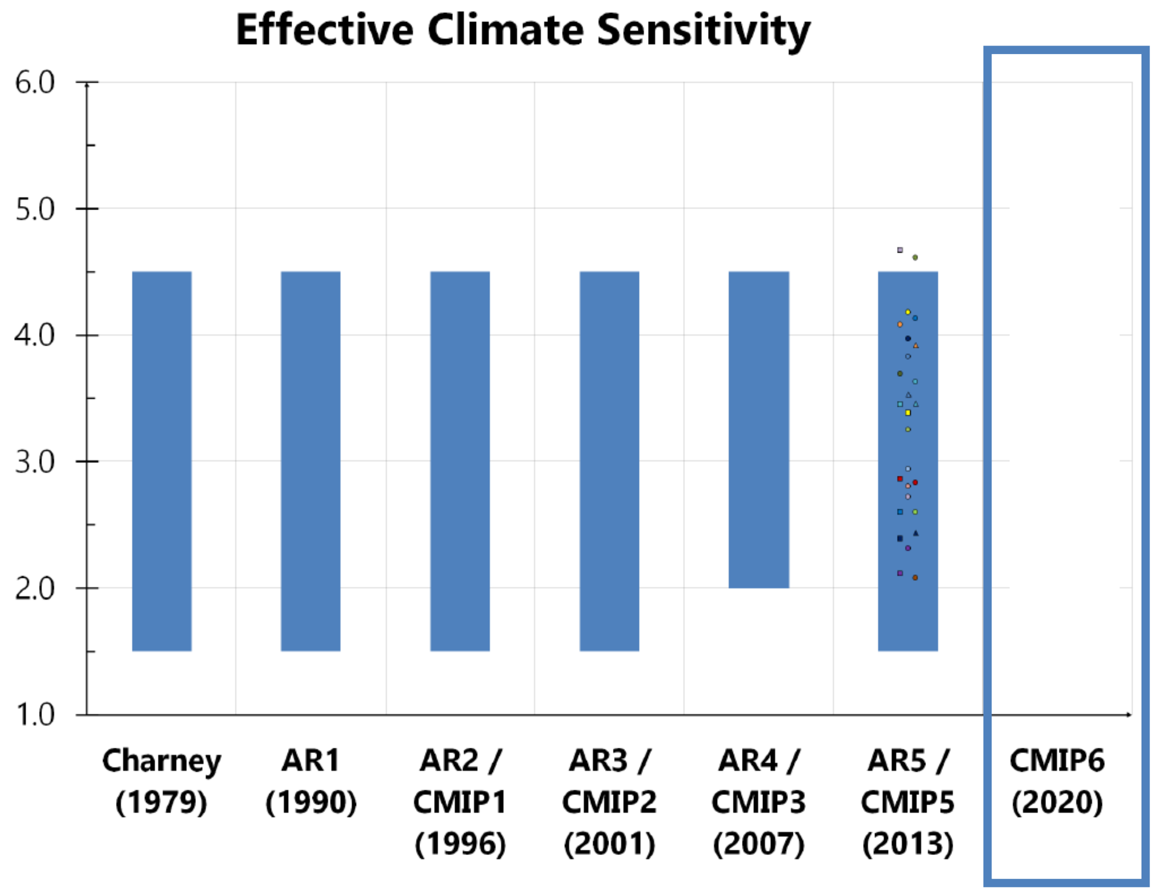
Advisors: Michael Pritchard (UCI) & Pierre Gentine (Columbia)

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

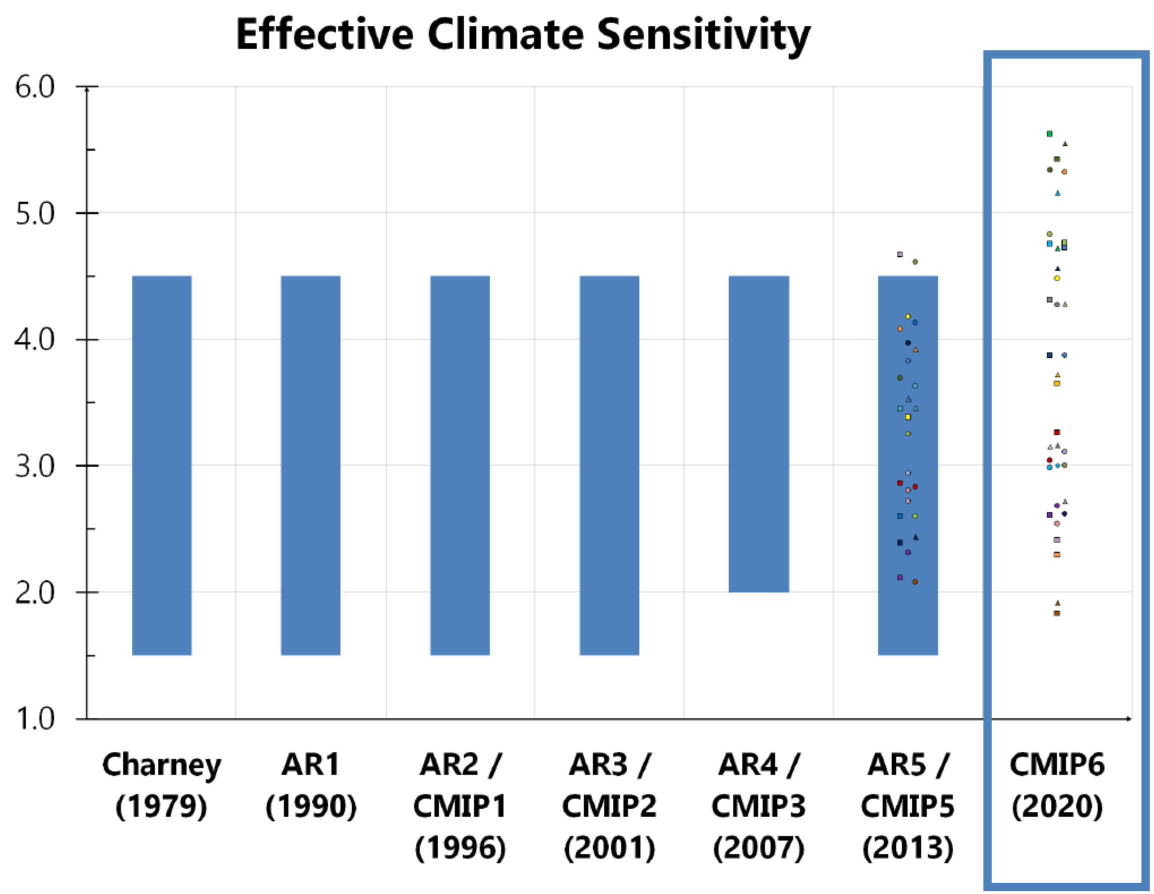
Eq. Temperature Response to CO_2 doubling



<u>Source:</u> Gentine et al. (Submitted), adapted from Meehl et al. (In Review)

Motivation 1: Largest uncertainties in climate projections come from clouds

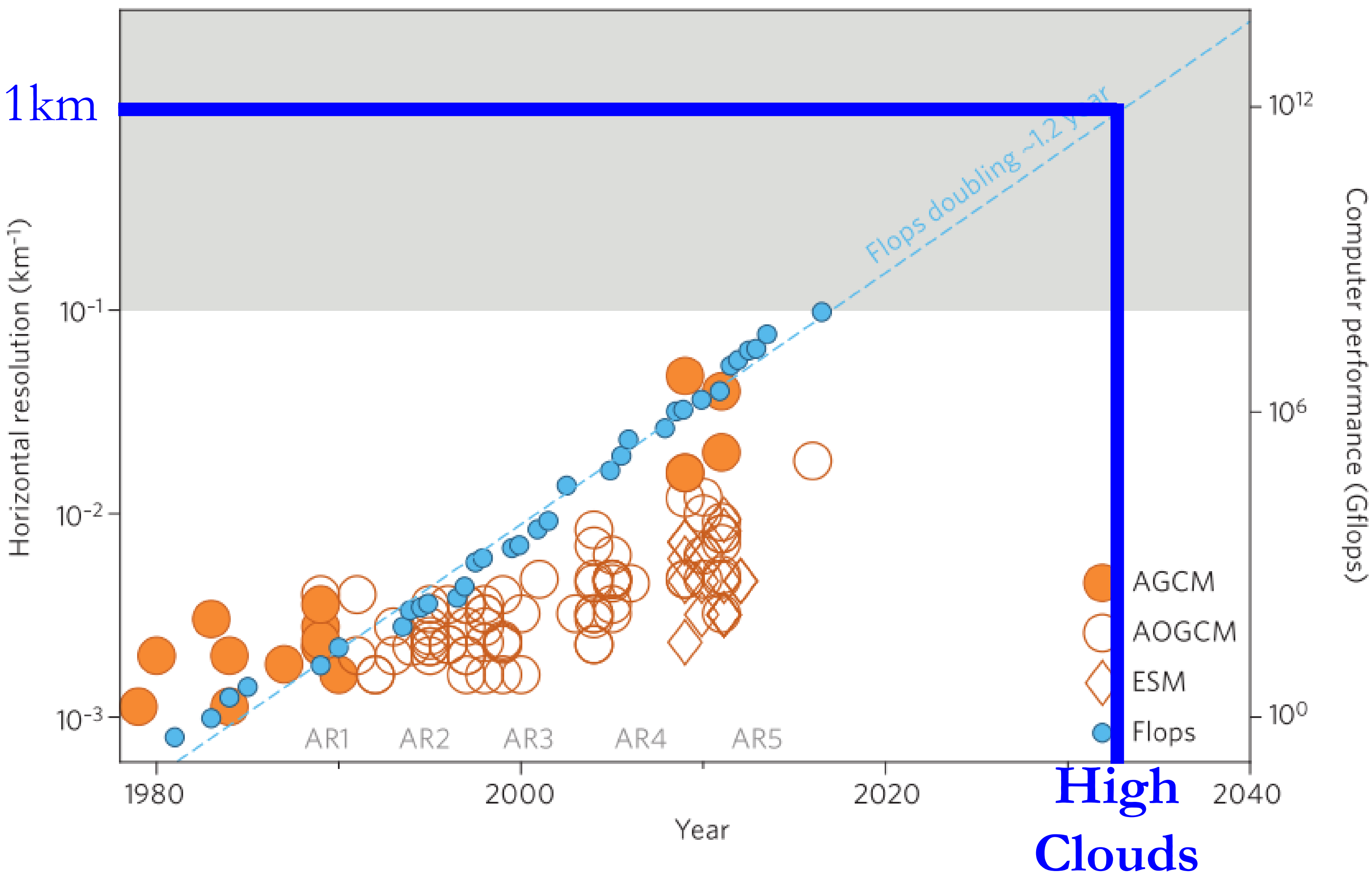
Eq. Temperature Response to CO_2 doubling



Cloud shortwave & longwave feedbacks are top contributors to spread

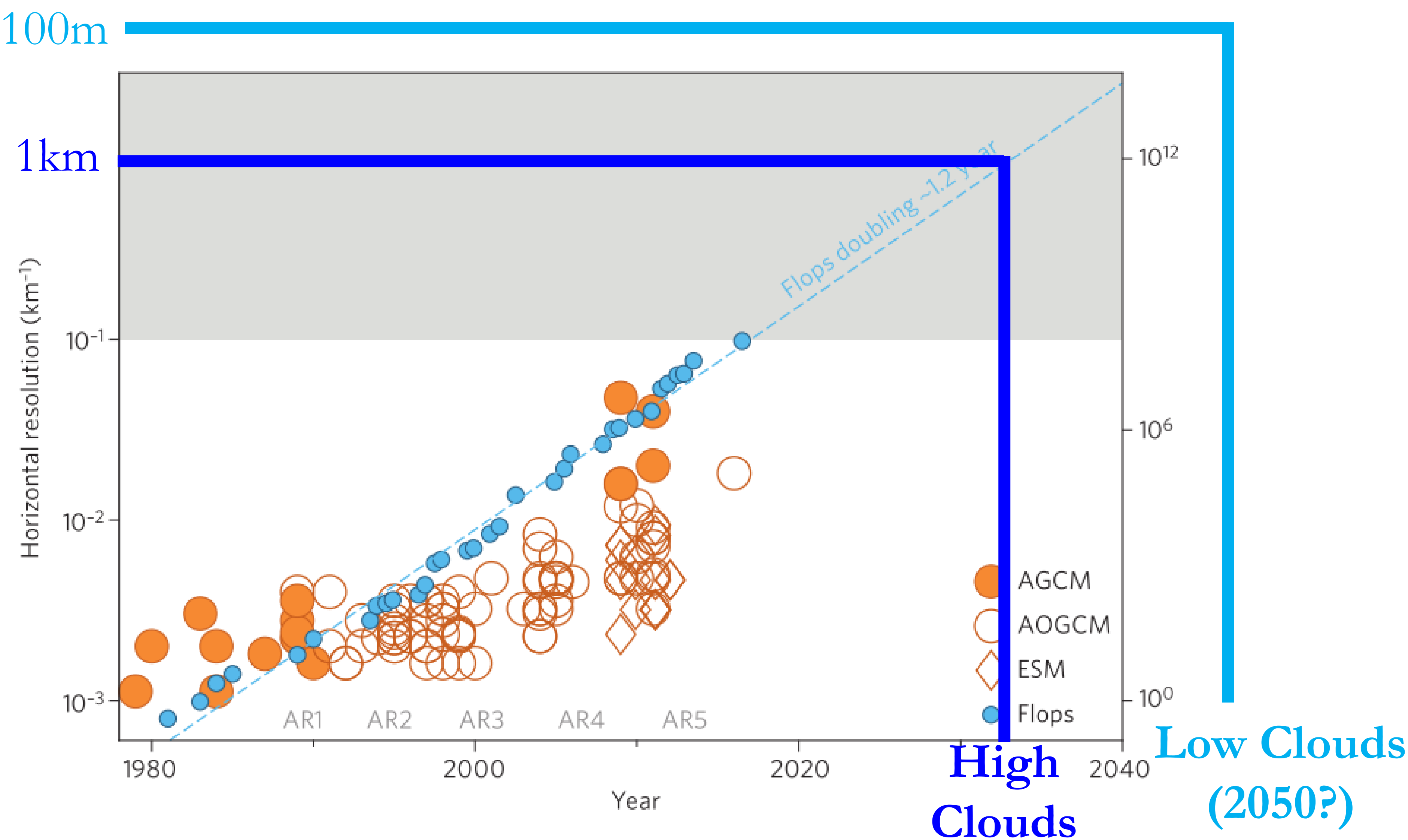
<u>Source:</u> Gentine et al. (Submitted), adapted from Meehl et al. (In Review), Zelinka et al. (2020)

Motivation 2: 100-year Climate Simulations unable to resolve low clouds before 2050



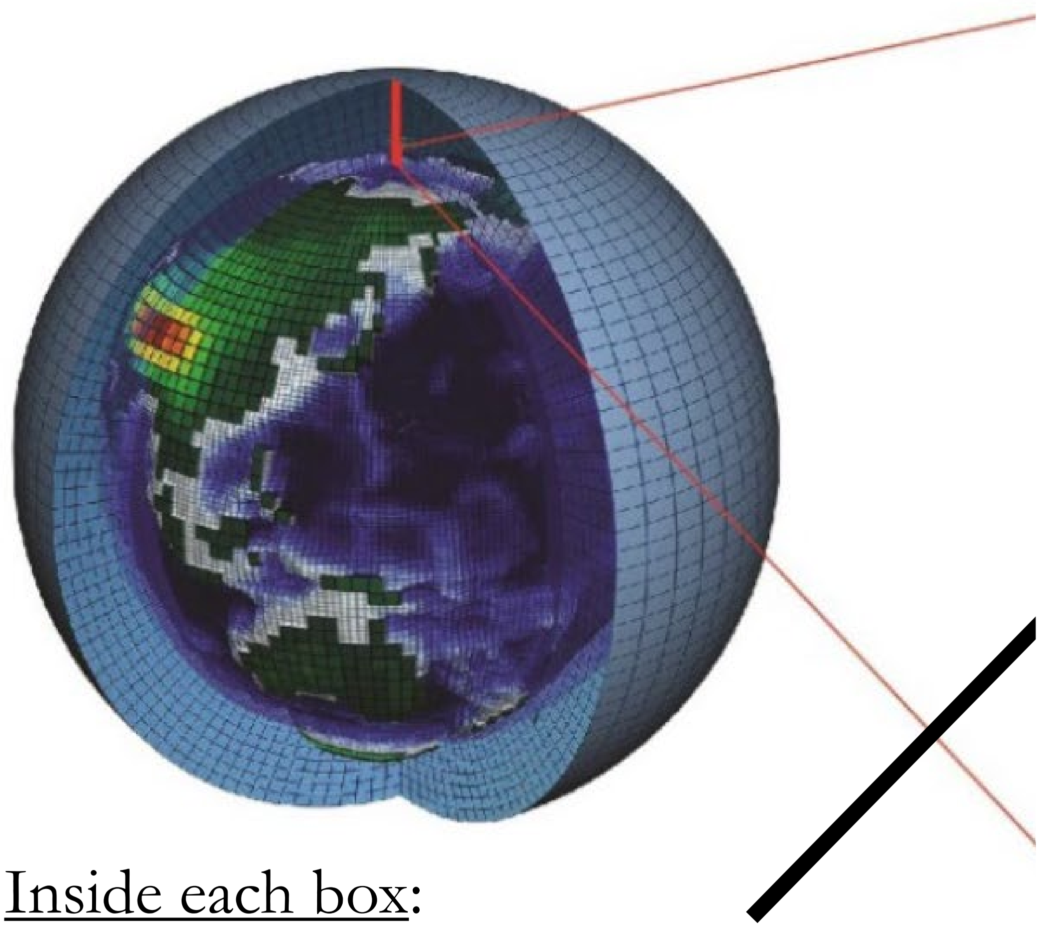
<u>Source:</u> Schneider et al., 2017

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



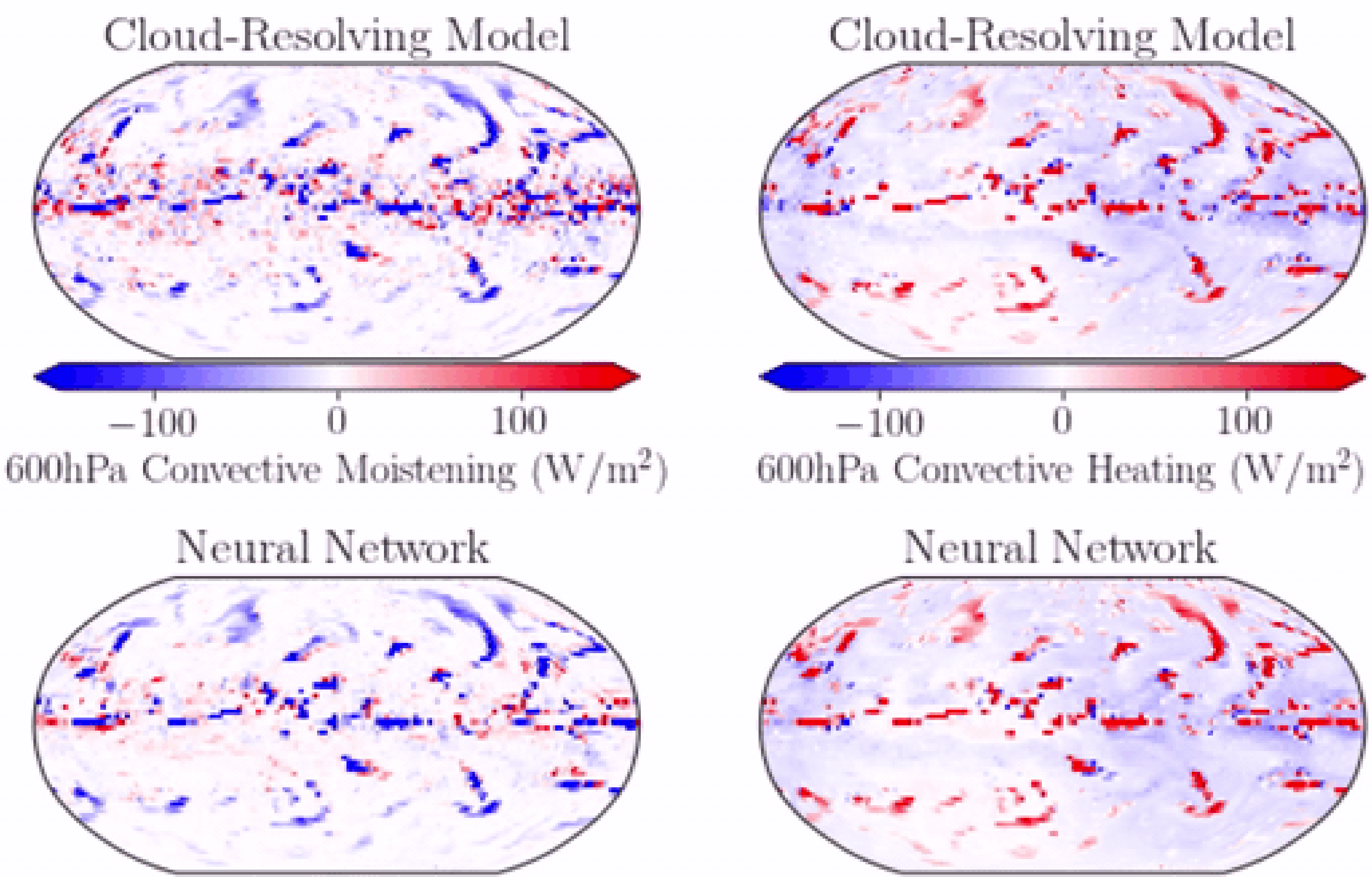
Goal

Inside each box:

Parametrization represents effects of fine-scale turbulence on coarse scale

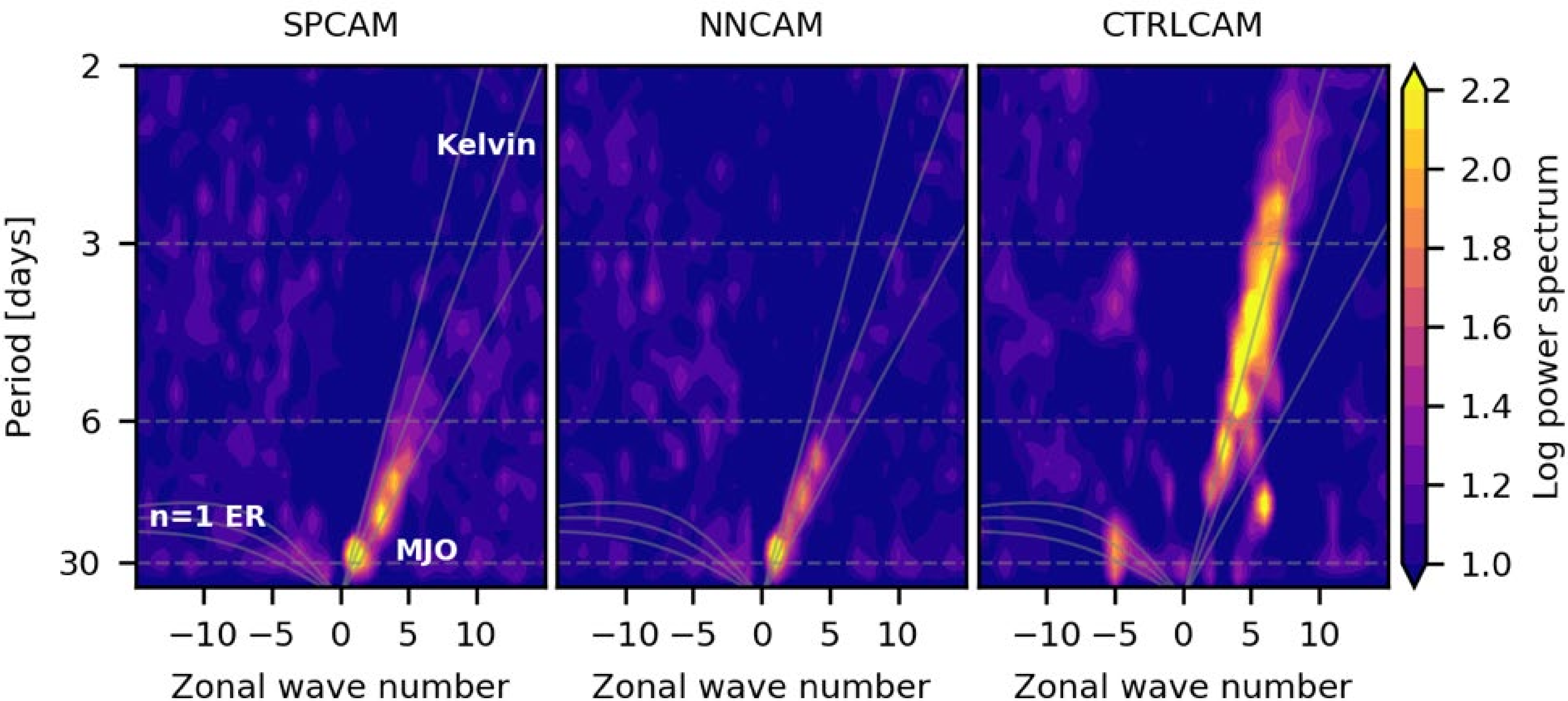
Source: Gentine et al. (Submitted)

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 ~10% CPU cost



<u>*See*</u>*: Rasp et al. (2018), Brenowitz et al. (2018,2019), O'Gorman et al. (2018), Krasnopolsky et al. (2013)*

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

<u>See</u>: 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…

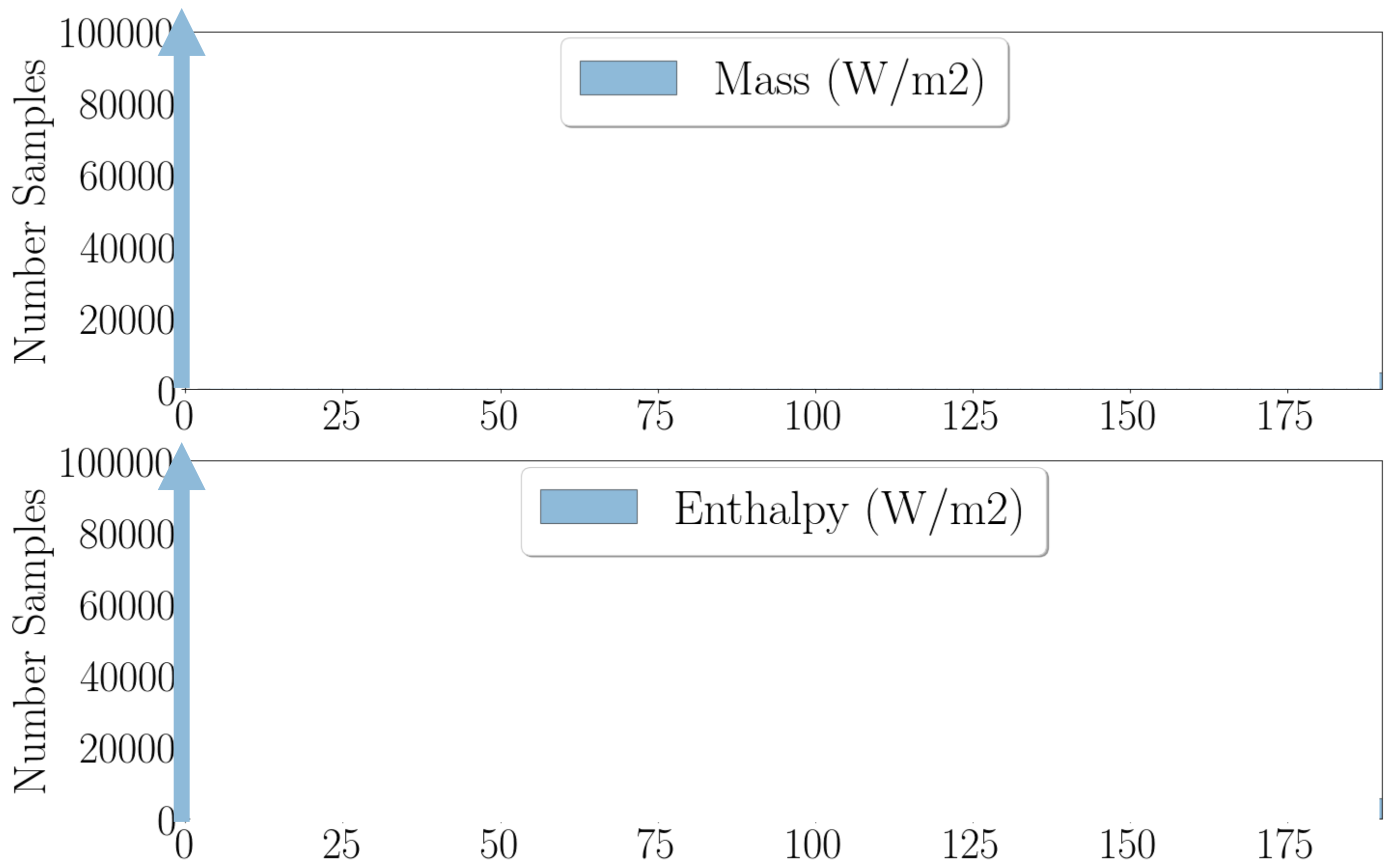
1. Are hard to **interpret** and trust, impinging their operational use
2. Violate **physical constraints** (Conservation laws, Positive definition)

<u>See</u>: Beucler 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 $O(10W/m^2)$ or negative precipitation



(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)
3. Make large errors when evaluated **outside of training** set

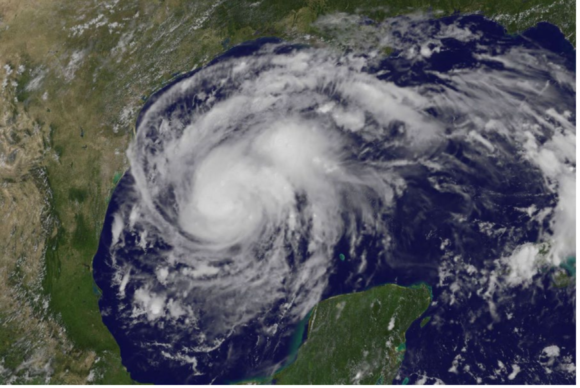
<u>*See*</u>*: Beucler et al. (2020)*

Problem: First crude attempts show that Machine-learning parametrizations…

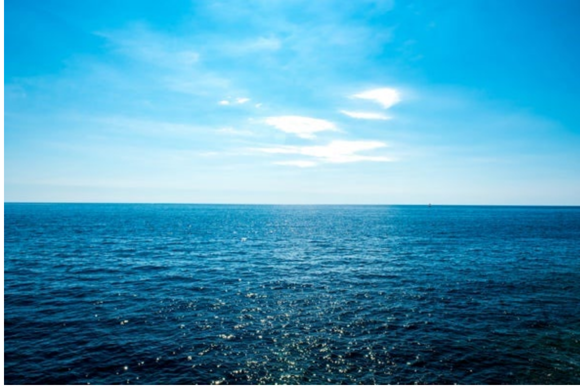
1. Are hard to **interpret** and trust, impinging their operational use
2. Violate **physical constraints** (Conservation laws, Positive definition)
3. Make large errors when evaluated **outside of training** set

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

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



1
Interpretability
Stability



Given Data-driven Parametrization, How to make it **Interpretable** & **Physically Consistent**?

2
Physical
Constraints

Here shown for
subgrid parametrization convection
but broadly applicable to
data-driven models
of physical processes

3
Generalization

1
Interpretability
Stability

Given Data-driven Parametrization,
How to make it **Interpretable**
& **Physically Consistent**?

2
Physical
Constraints

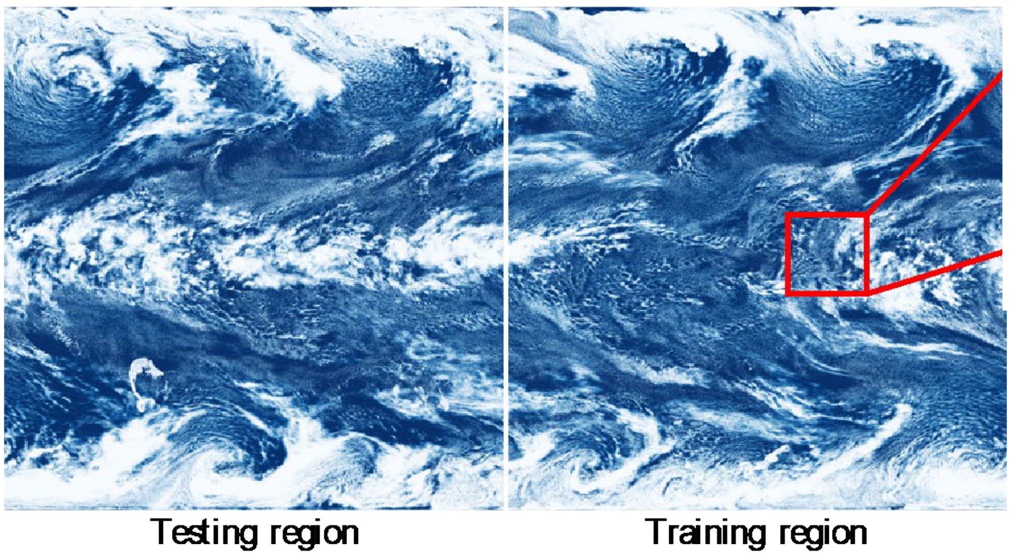
3
Generalization

1) Global Cloud-Resolving Model (UW)

<u>See</u>: Brenowitz et al. (2018, 2019, 2020), <u>Movie source</u>: Noah Brenowitz

1) Global Cloud-Resolving Model (UW)

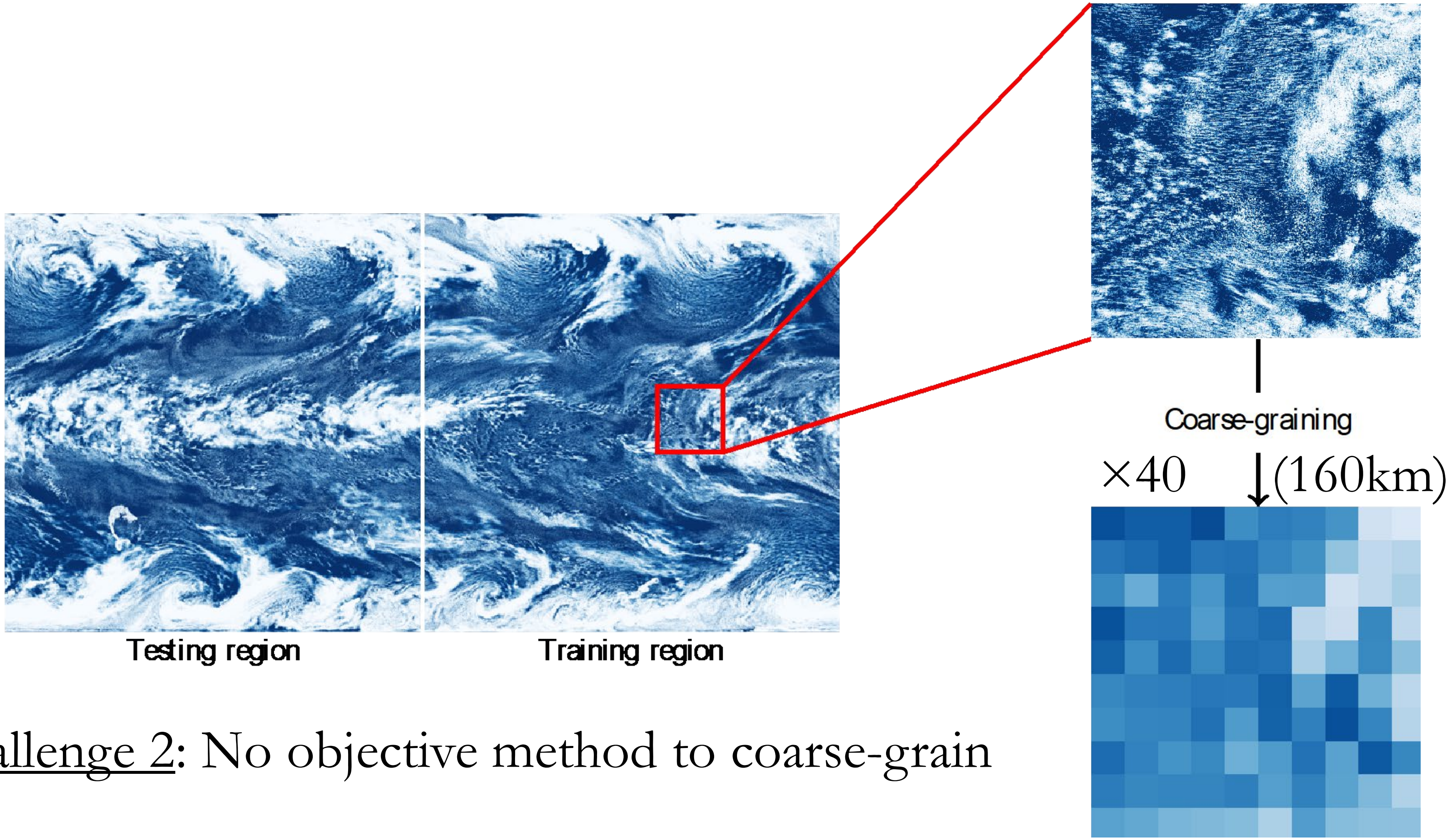
<u>Challenge</u>: No clear “truth”/ “target” → Requires coarse-graining data



<u>See</u>: Brenowitz et al. (2018), <u>Slide source</u>: Noah Brenowitz

1) Global Cloud-Resolving Model (UW)

<u>Challenge</u>: No clear "truth"/ "target" → Requires coarse-graining data

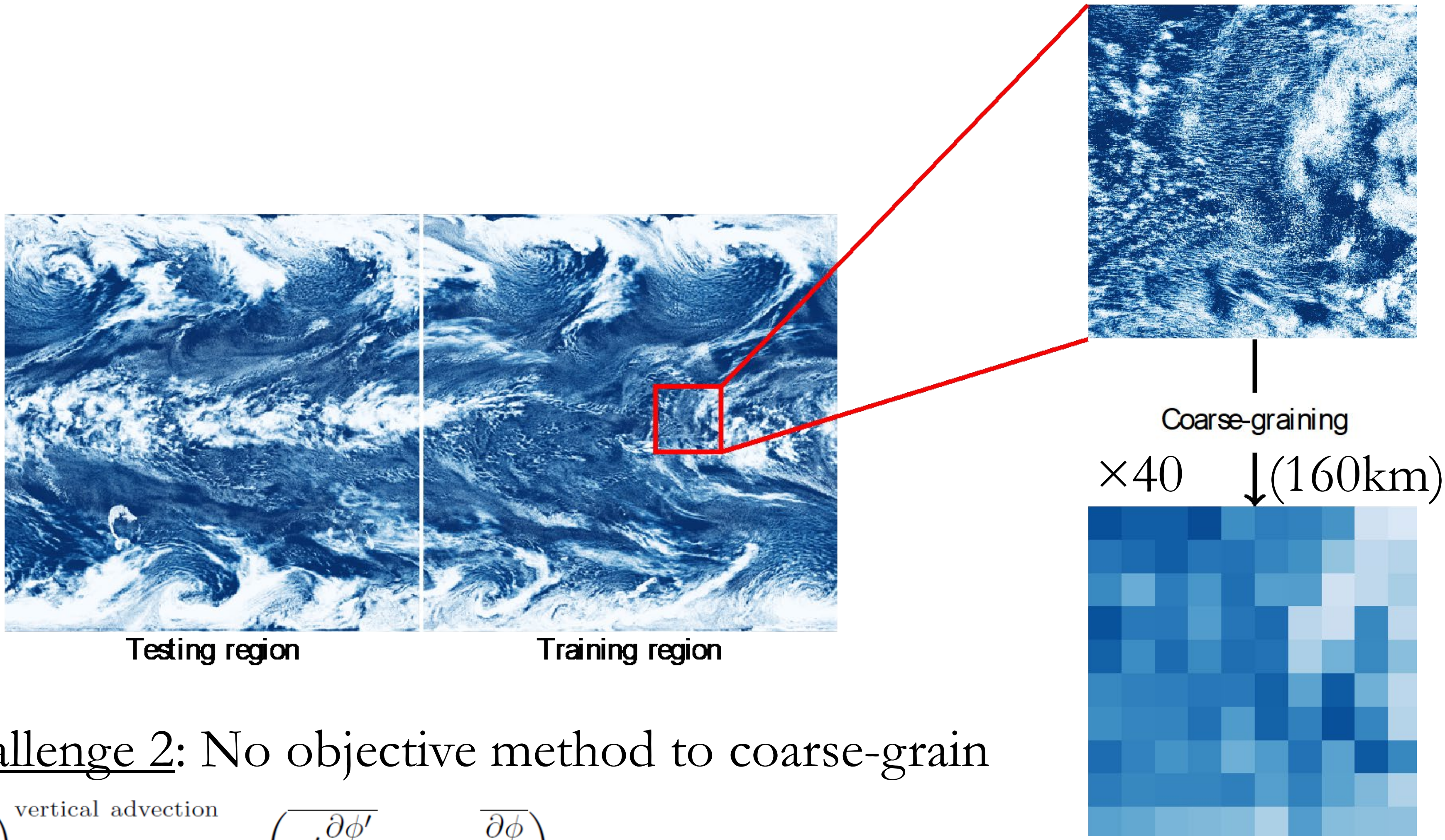


<u>Challenge 2</u>: No objective method to coarse-grain

<u>See</u>: Brenowitz et al. (2018), <u>Slide source</u>: Noah Brenowitz

1) Global Cloud-Resolving Model (UW)

<u>Challenge</u>: No clear "truth"/ "target" → Requires coarse-graining data



<u>Challenge 2</u>: No objective method to coarse-grain

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

<u>See</u>: Brenowitz et al. (2018), Yuval et al. (2020), <u>Slide source</u>: Noah Brenowitz

2) "Truth": Super-Parametrized CAM

Exterior Model:

Does not resolve convection

$\Delta t = 30\text{m}, \Delta x = 2°$

Interior Model:

Resolves response convection

$\Delta t = 20\text{s}, \Delta x = 4\text{km}$

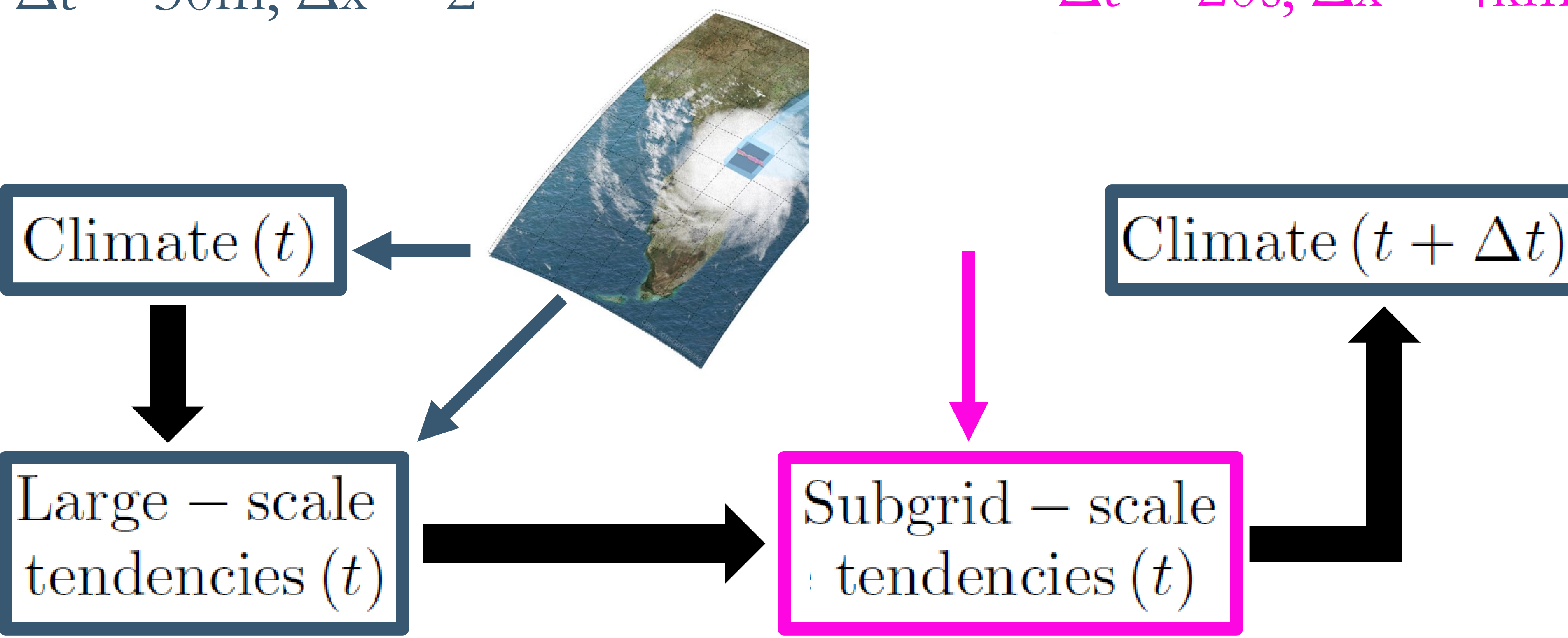
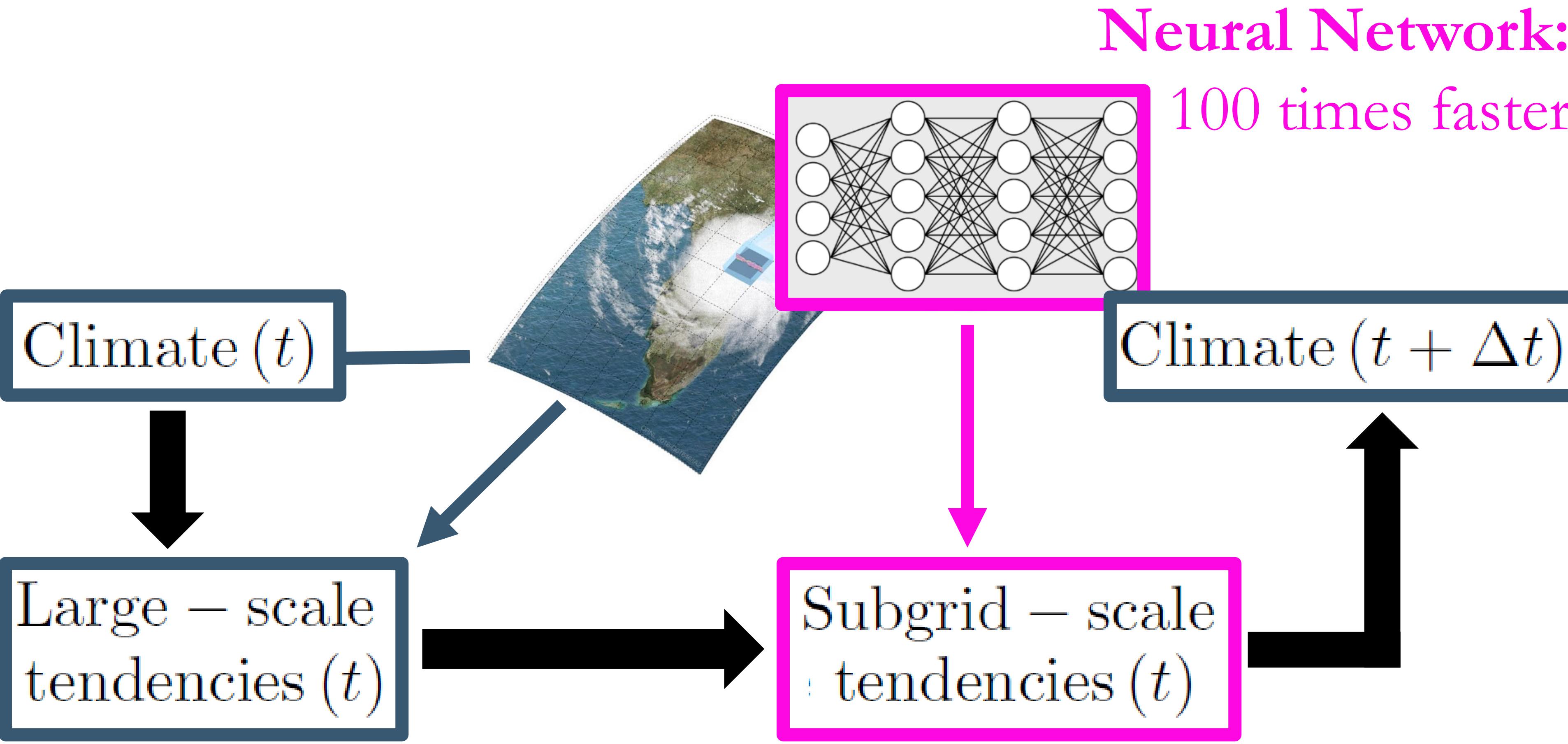


Image source: e3sm.org, Model source: Khairoutdinov et al. (2004)

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)

Fully connected network
layers & nodes decided via
e.g. formal hyperparameter tuning
Loss = Mass-weight. MSE (W/m^2)
Adam optimizer
7 layers of 128 $\rightarrow$ ~100k param.
Trained for 15-20 epochs
Train. = yr1 (42,369,024 samples)
Valid. = yr2 (42,369,024 samples)

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)

<u>*See:*</u> *SHERPA (Github), Hertel et al. (2018), Gentine et al. (2018), Rasp et al. (2018), Beucler et al. (2019)*

1
Interpretability
Stability

Given Data-driven Parametrization,
How to make it **Interpretable**
& **Physically Consistent**?

2
Physical
Constraints

3
Generalization

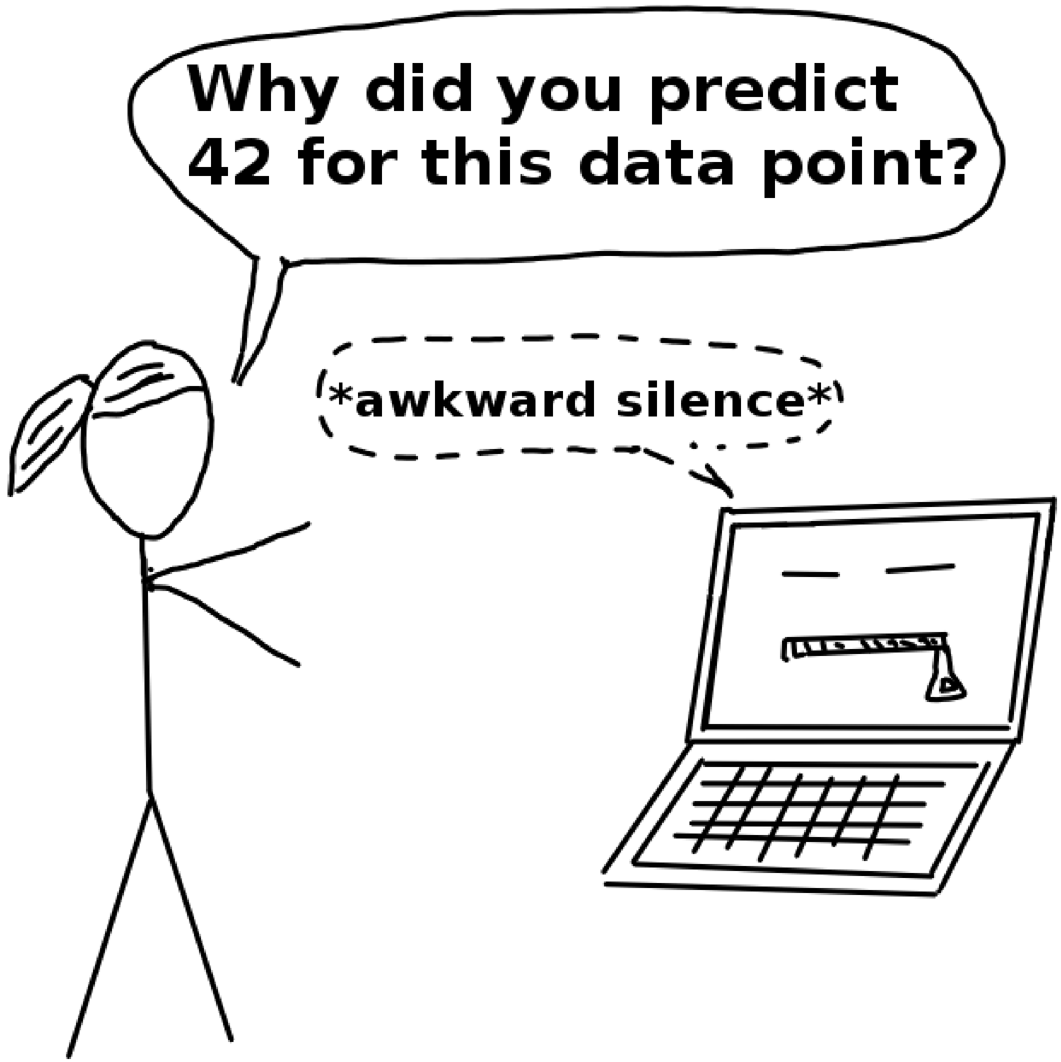
1
Interpretability
Stability

Given Data-driven Parametrization,
How to make it **Interpretable**
& **Physically Consistent**?

2
Physical
Constraints

3
Generalization

Problem: For climate modeling, we need interpretable parametrizations



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

Tailor NN interpretability techniques to parametrization task

<u>*See*</u>*: McGovern et al. (2019), Toms et al. (2019), Montavon et al. (2018), Molnar et al. (2018)*

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: $$\boldsymbol{x} \overset{\mathrm{NN}}{\mapsto} \boldsymbol{y} = \boldsymbol{f}\left(\boldsymbol{x}\right)$$

Isolate inputs of interest: $$\boldsymbol{x} = \begin{bmatrix}\boldsymbol{x}_{\mathbf{PDP}} & \boldsymbol{x}_{\mathbf{Other}}\end{bmatrix}^{\boldsymbol{T}}$$

Partial dependence fx: $$\boldsymbol{x}_{\mathbf{PDP}} \mapsto \mathbb{E}\left\{\boldsymbol{f}\begin{bmatrix}\boldsymbol{x}_{\mathbf{PDP}} & \boldsymbol{x}_{\mathbf{Other}}\end{bmatrix}^{\boldsymbol{T}} \middle| \boldsymbol{x}_{\mathbf{PDP}}\right\}$$

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

<u>See</u>: Molnar et al. (2018), Friedman (2001)

1) Partial Dependence Plot

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

Image source: *NOAA Photo library (fly00890)*

1) Partial Dependence Plot

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

Input 1:
Mid-tropospheric
Moisture (QM)

Image source: NOAA Photo library (fly00890), See: Bretherton et al. (2004), Rushley et al. (2018)

1) Partial Dependence Plot

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

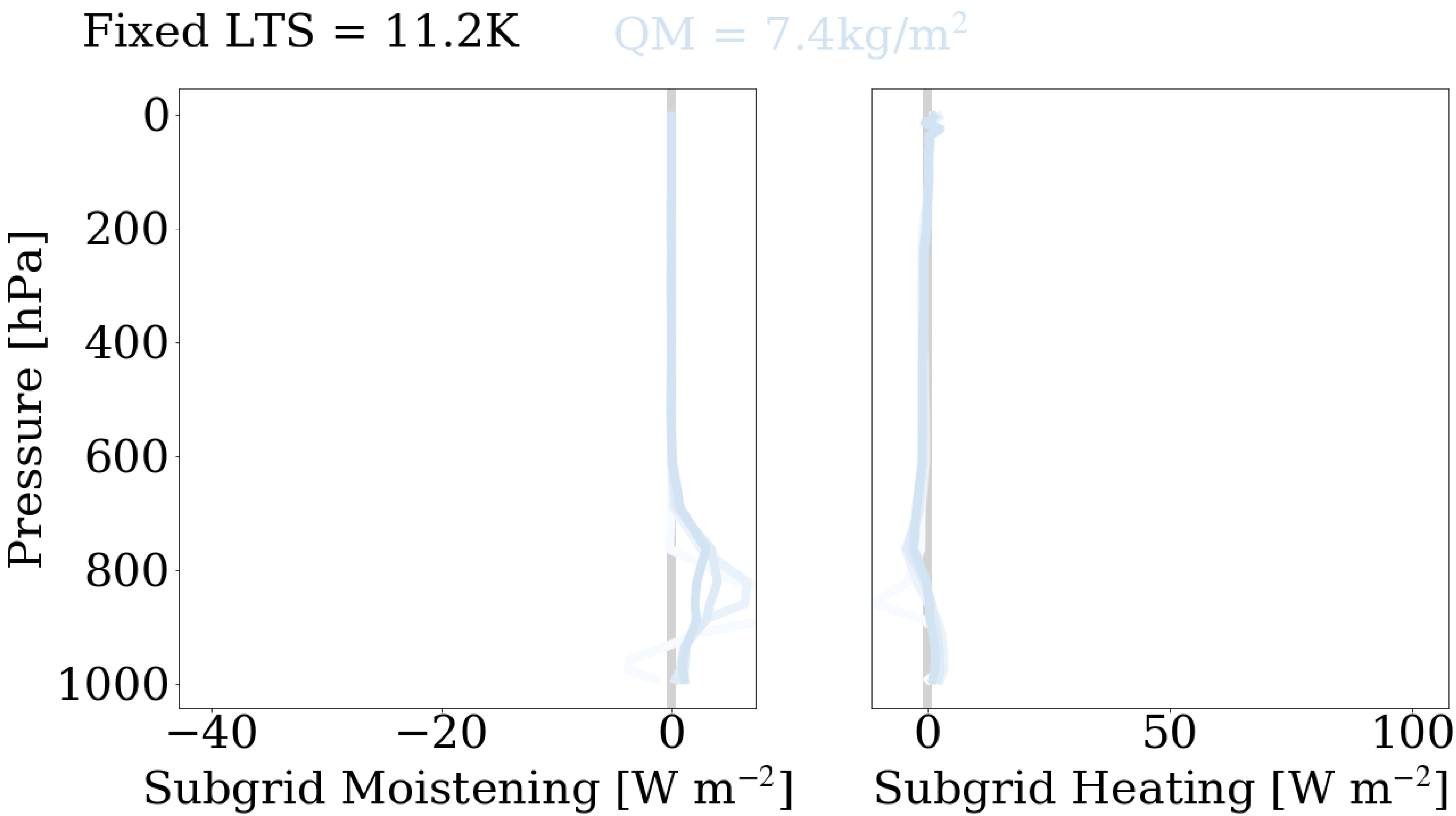
Input 1:
Mid-tropospheric Moisture (QM)

Input 2:
Lower-tropospheric Stability (LTS)

Image source: NOAA Photo library (fly00890), See: Wood and Bretherton (2006)

1) Partial Dependence Plot

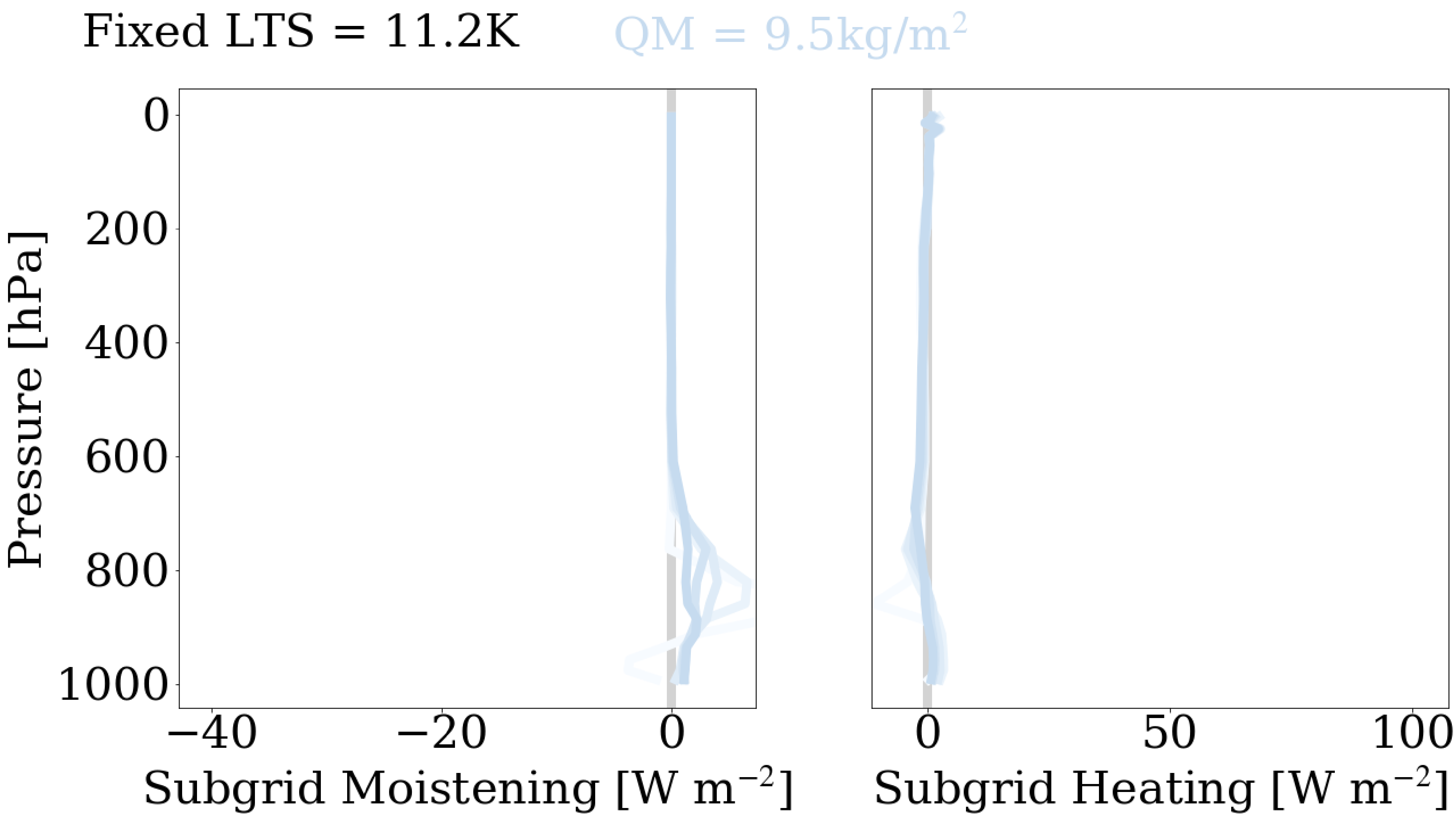
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

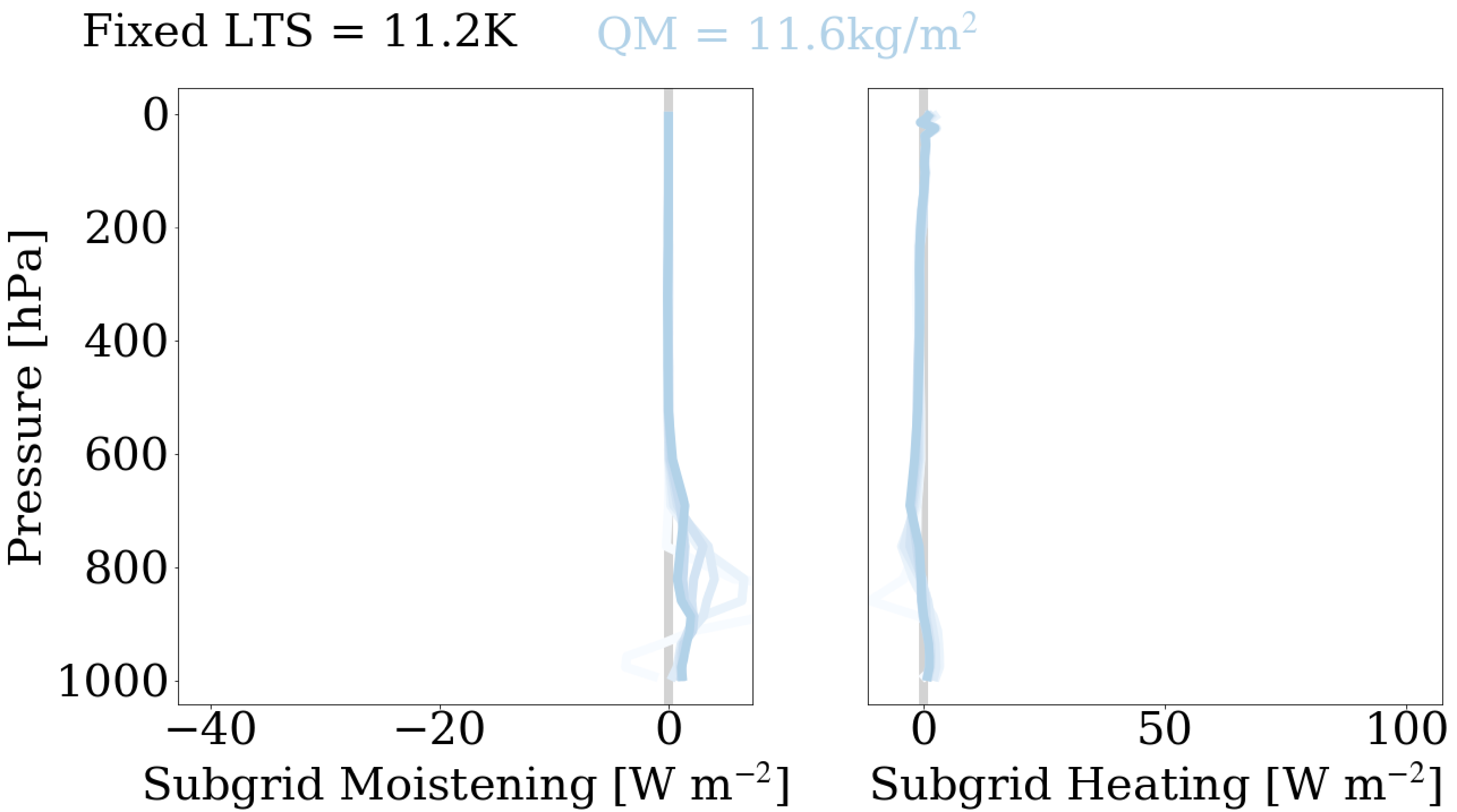
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

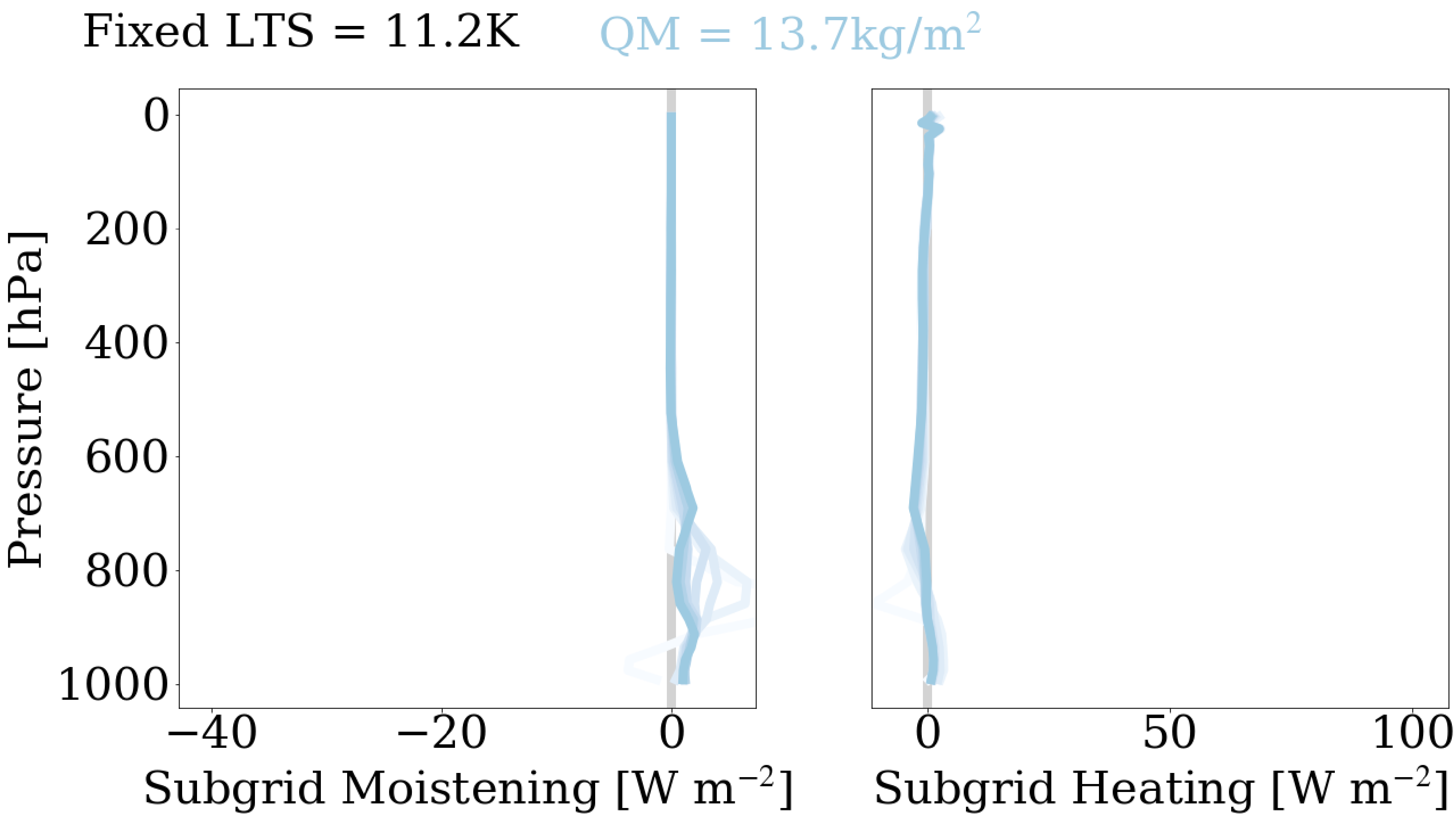
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

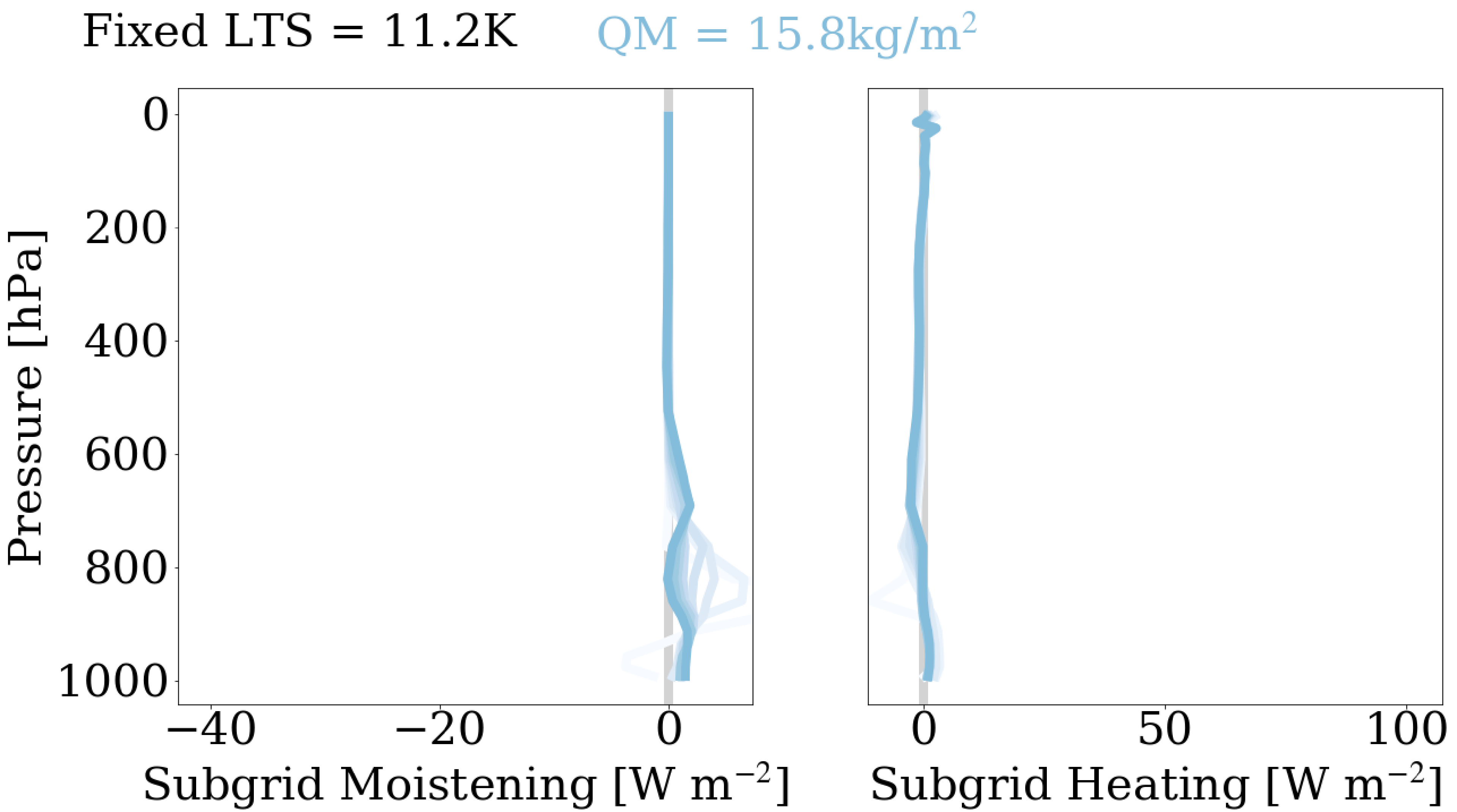
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

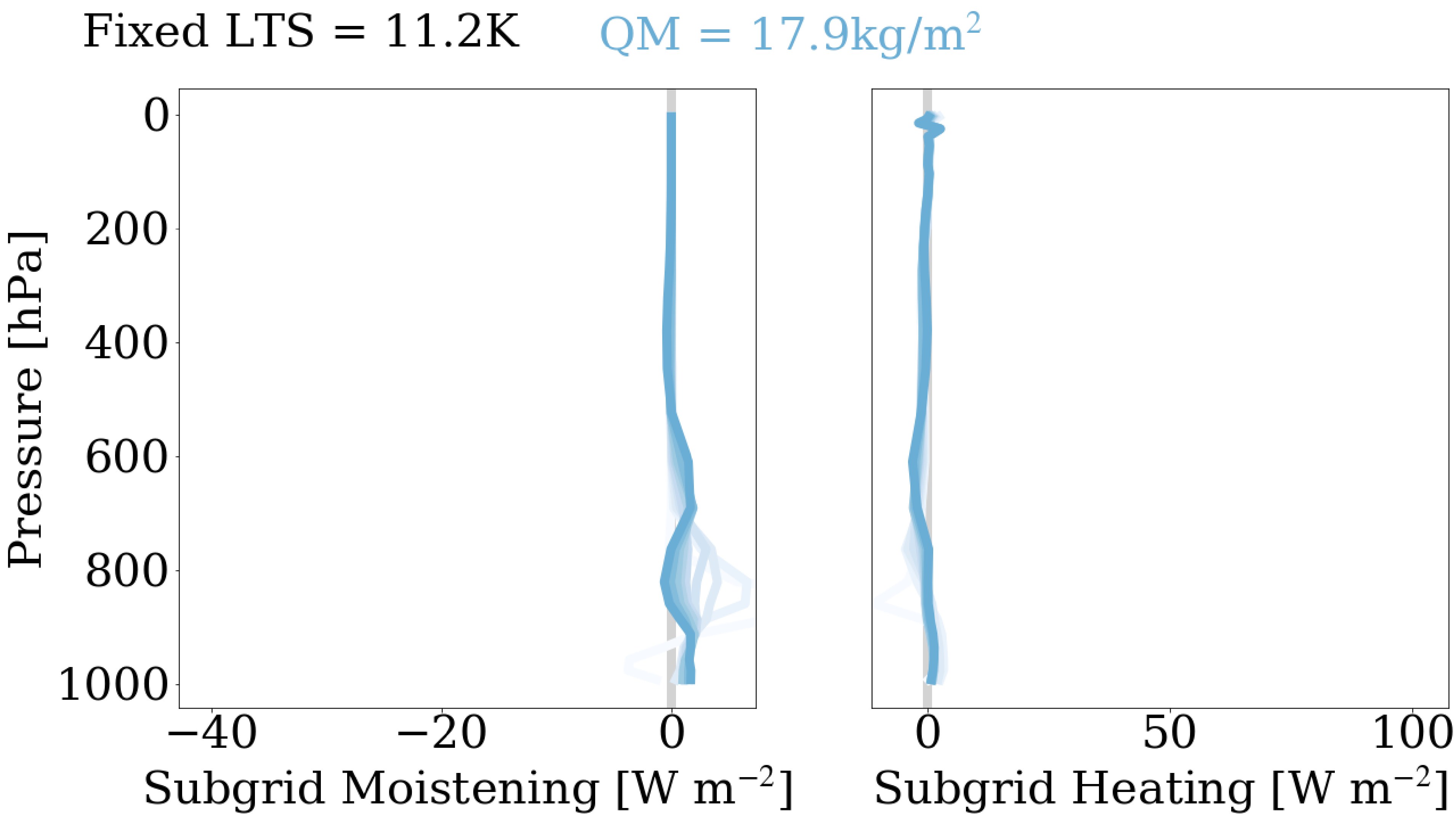
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

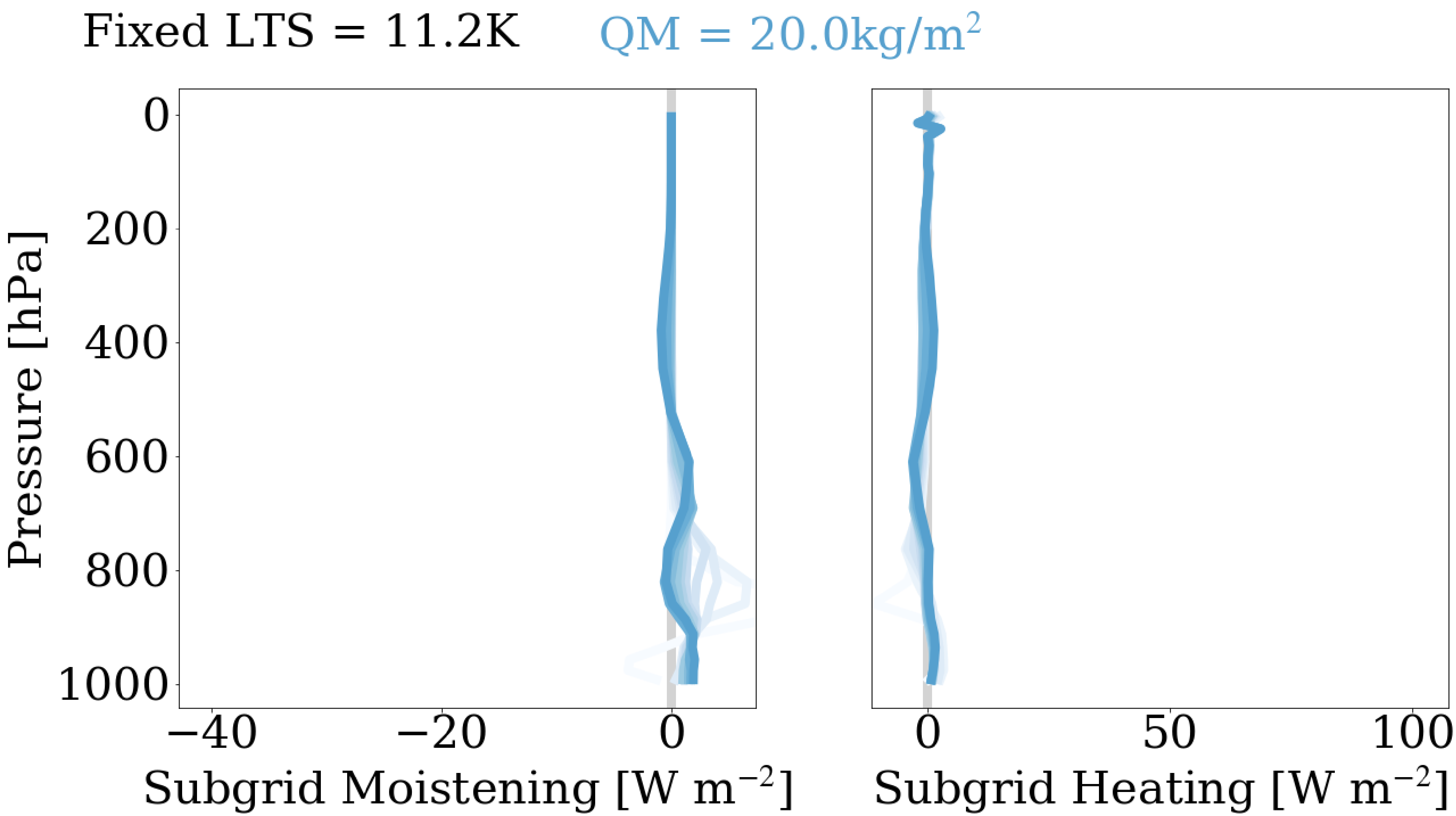
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

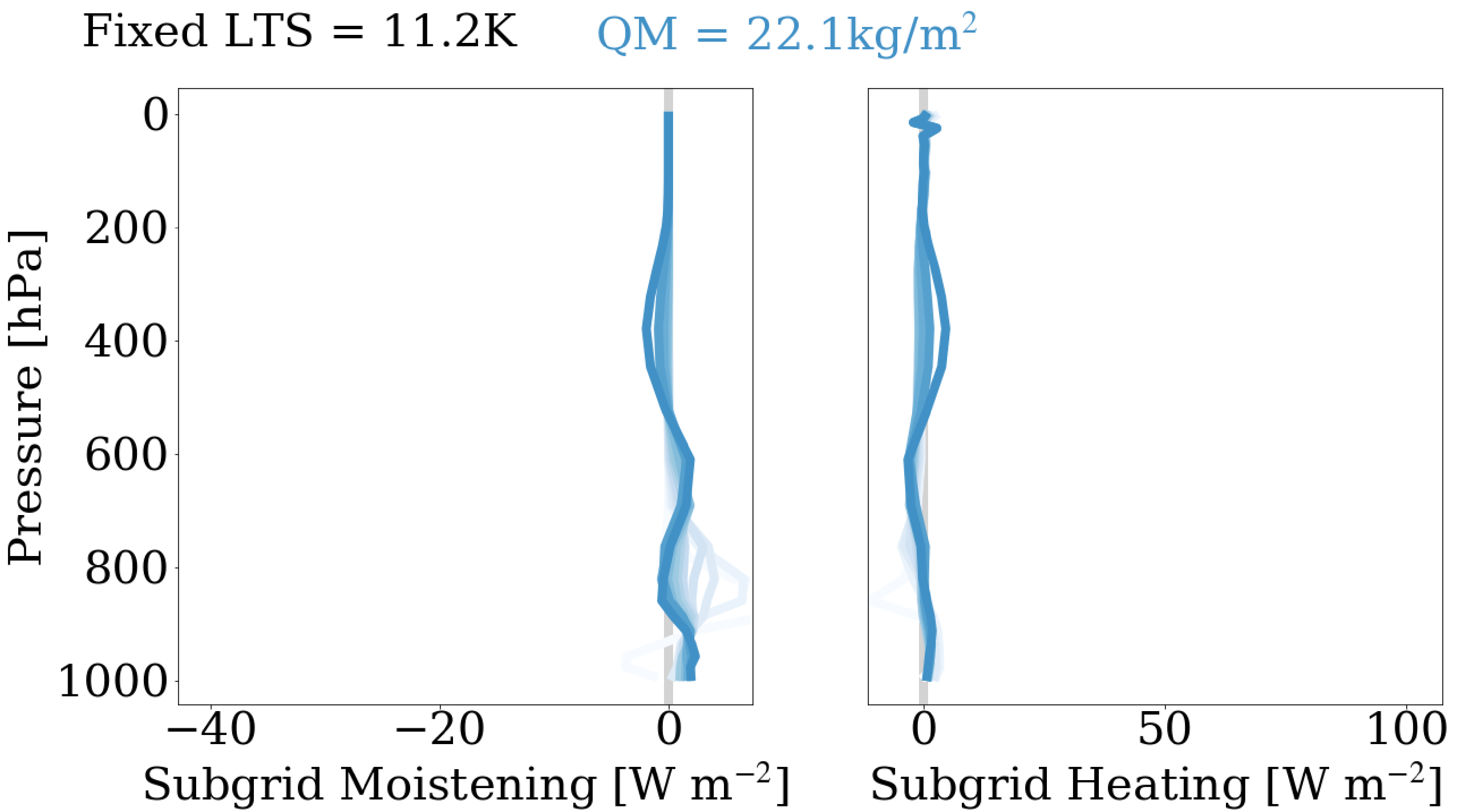
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

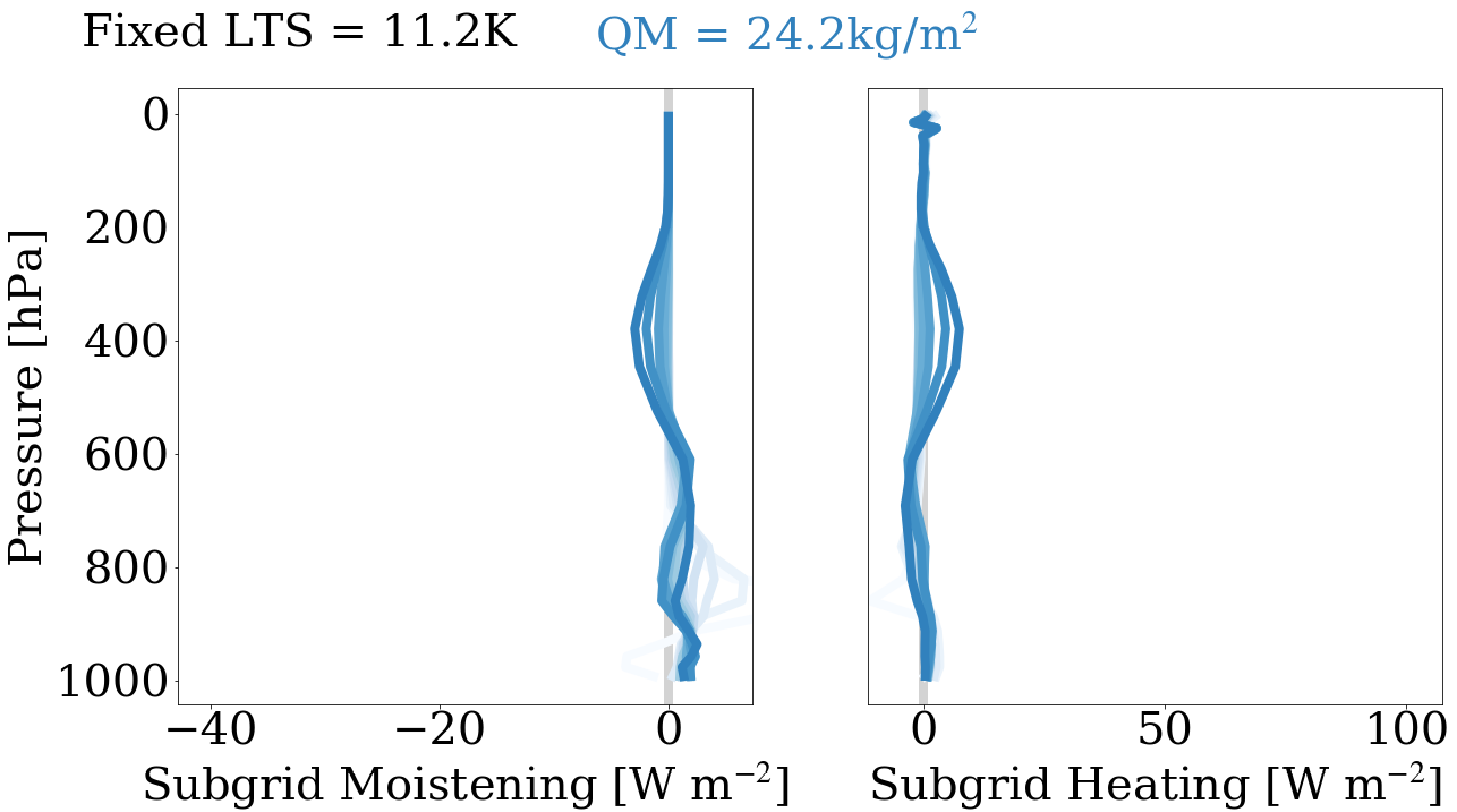
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

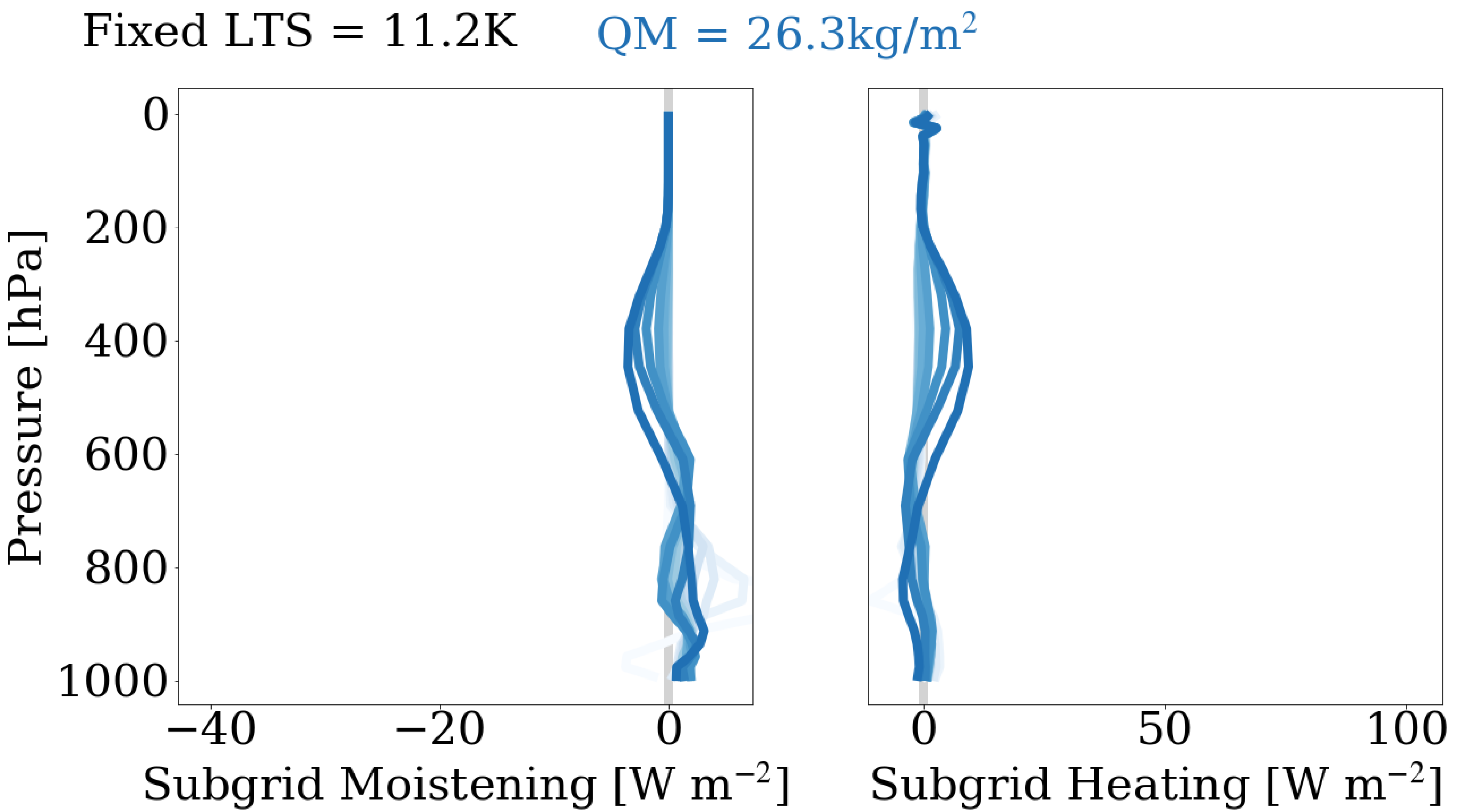
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



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

1) Partial Dependence Plot

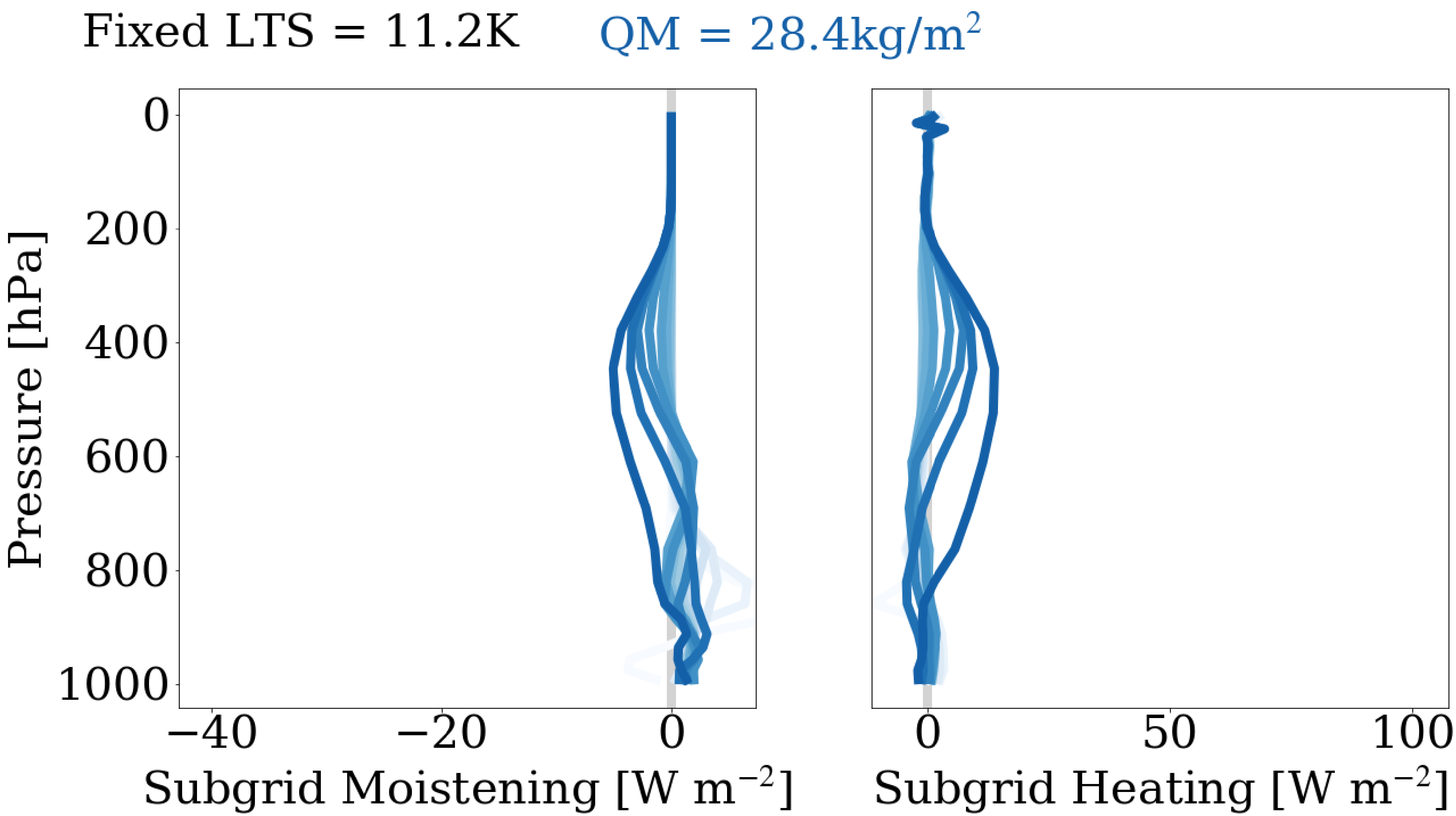
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

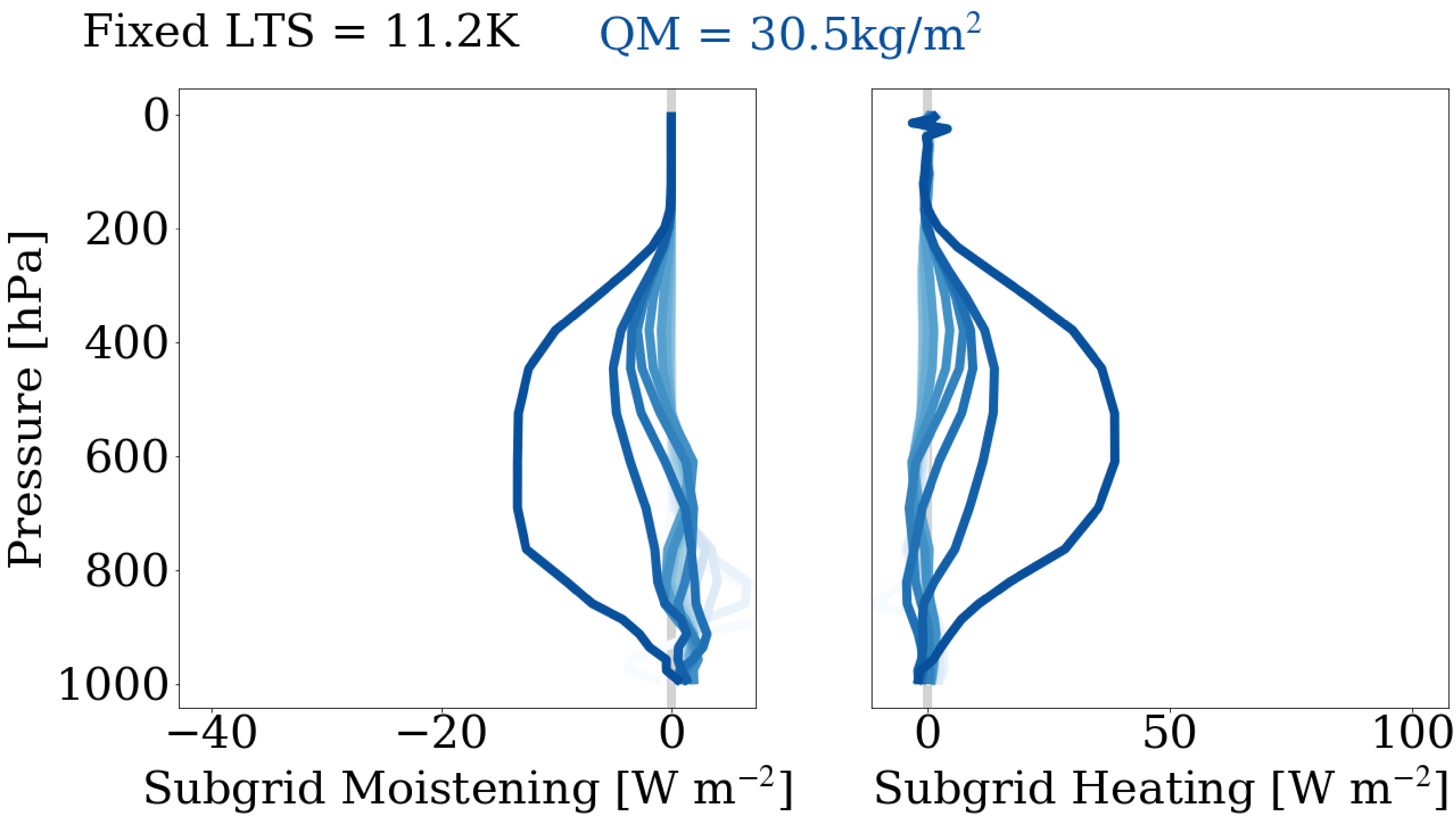
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



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

1) Partial Dependence Plot

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

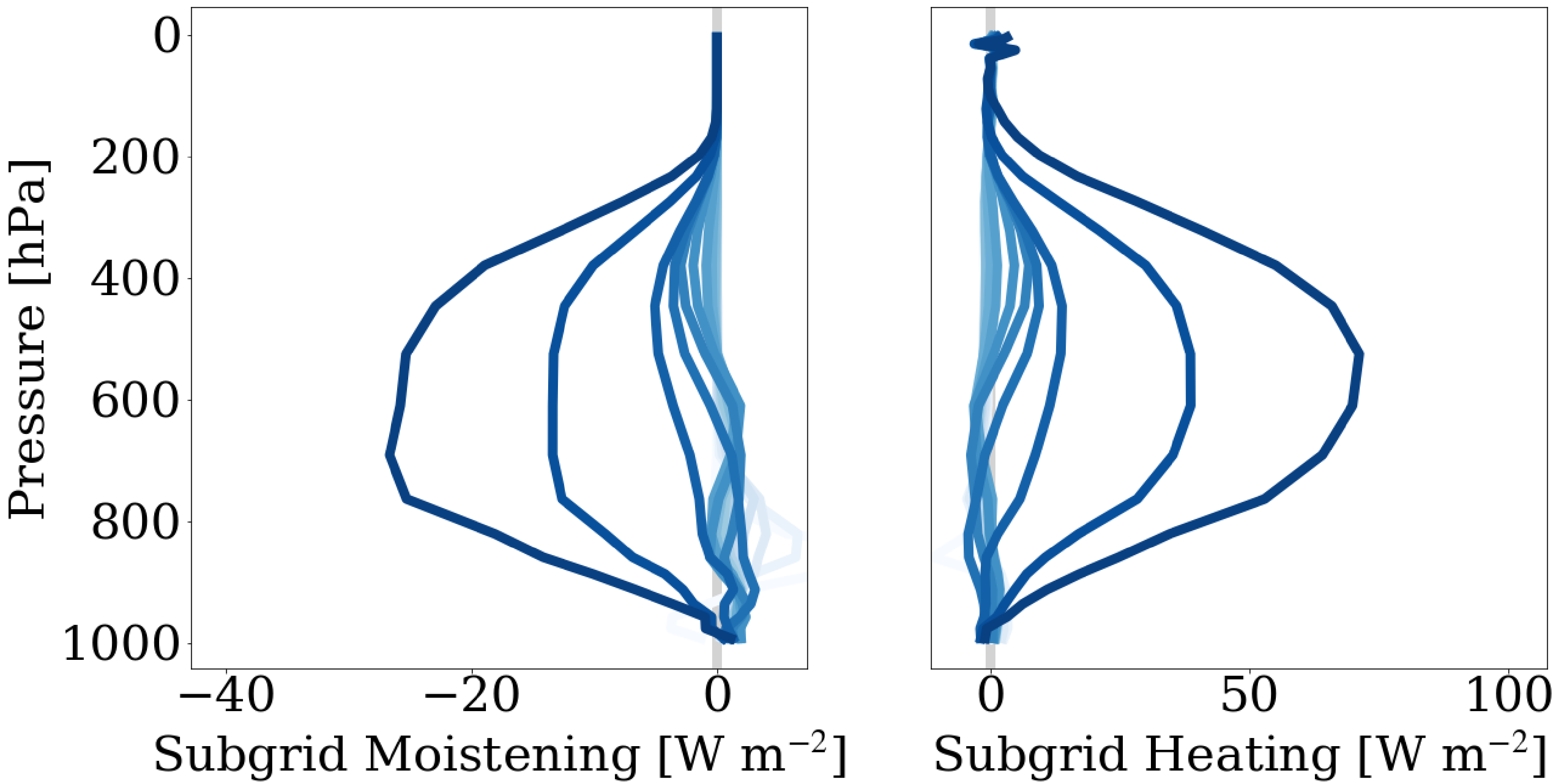


See: Brenowitz et al. (2020)

1) Partial Dependence Plot

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

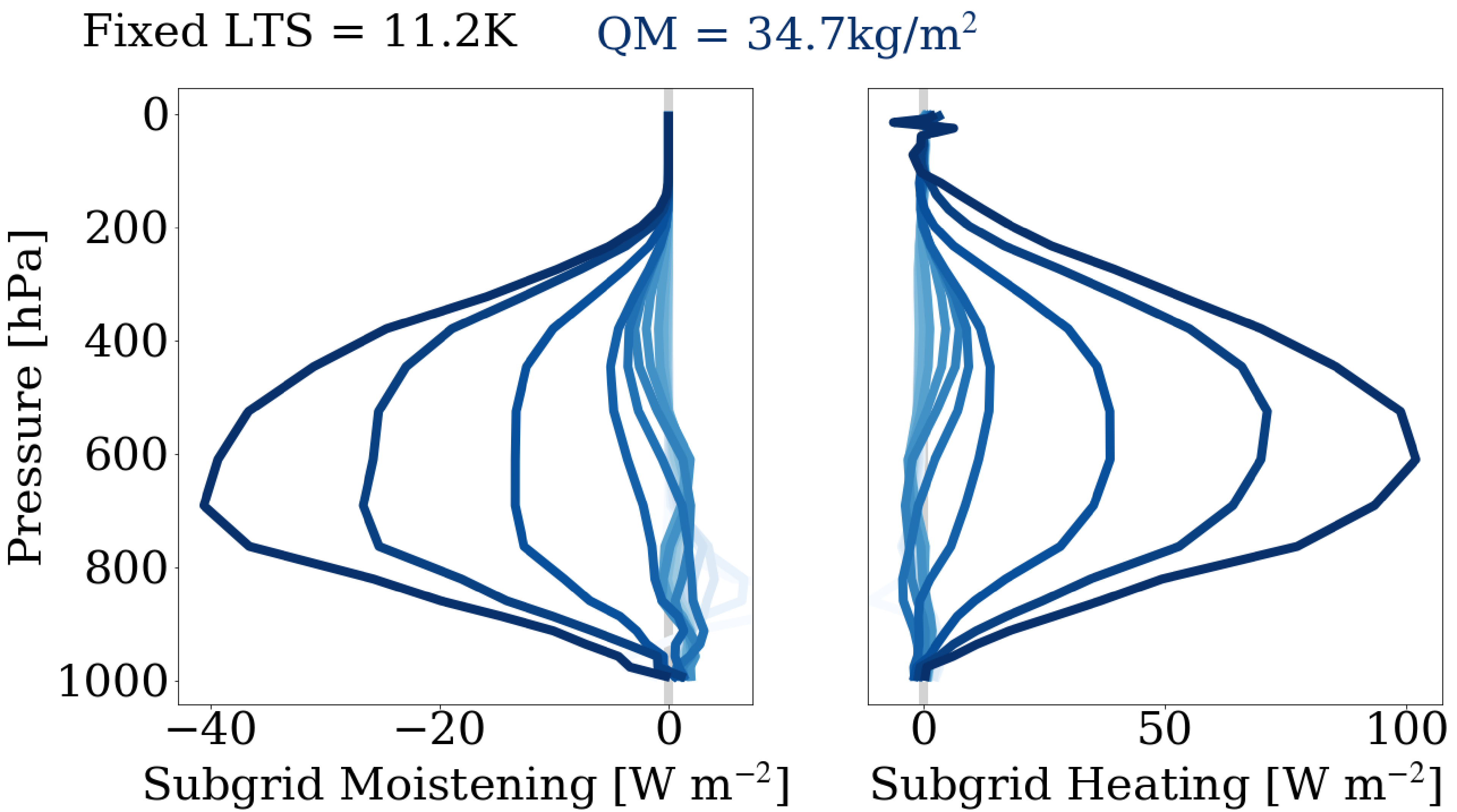
Fixed LTS = 11.2K QM = 32.6kg/m^2



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

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

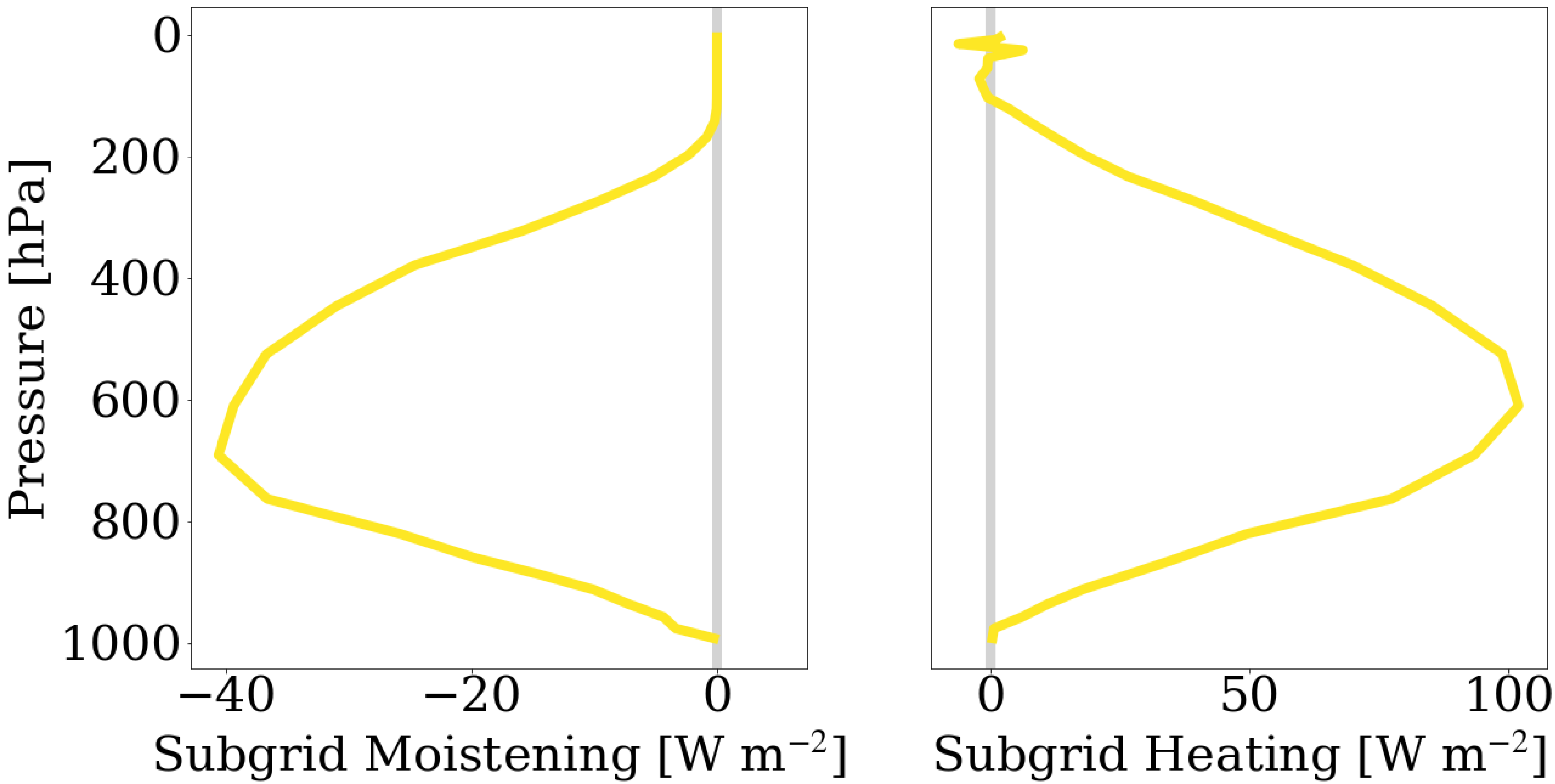


See: Brenowitz et al. (2020)

1) Partial Dependence Plot

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

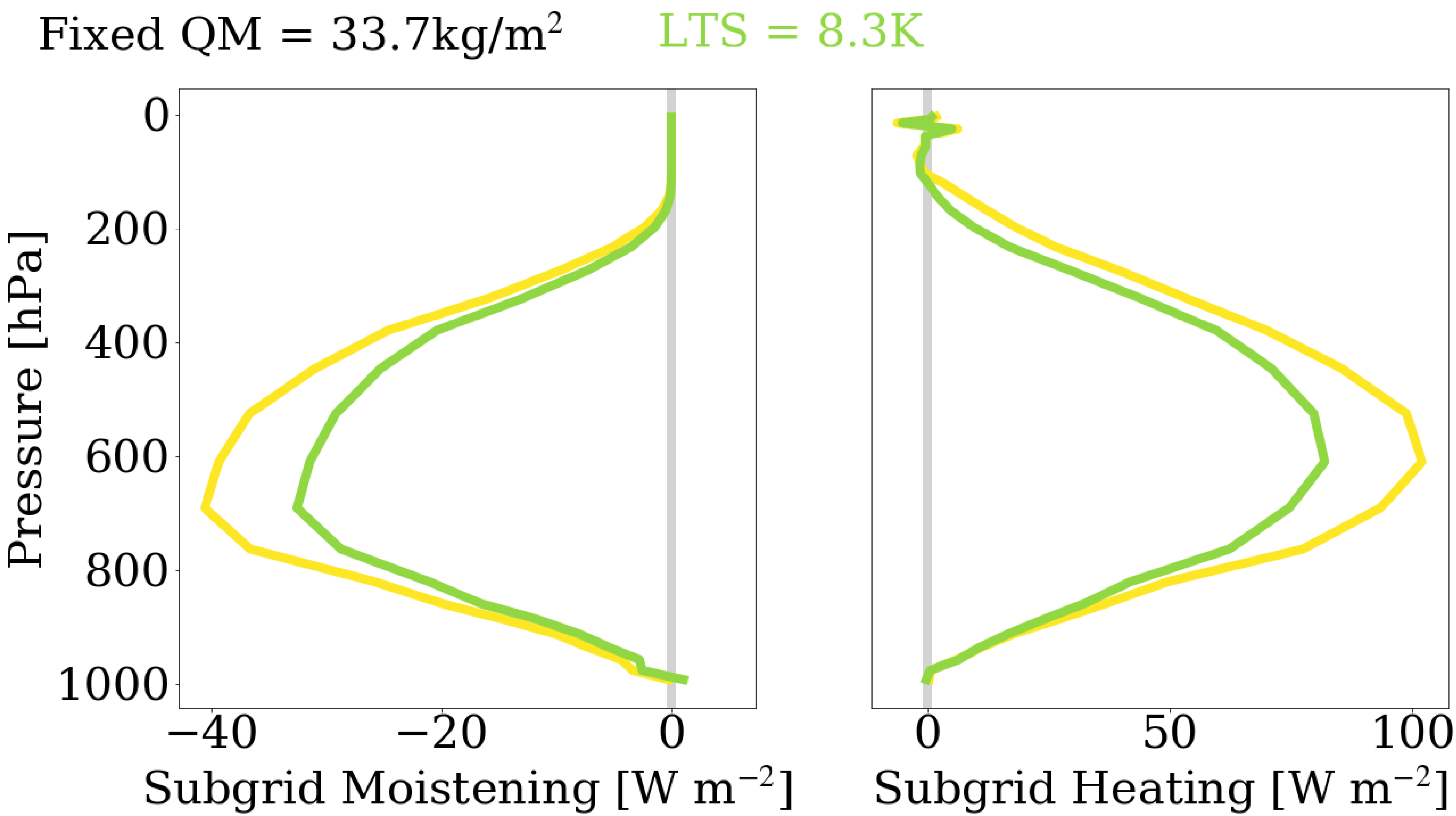
Fixed QM = 33.7kg/m^2 LTS = 7.4K



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

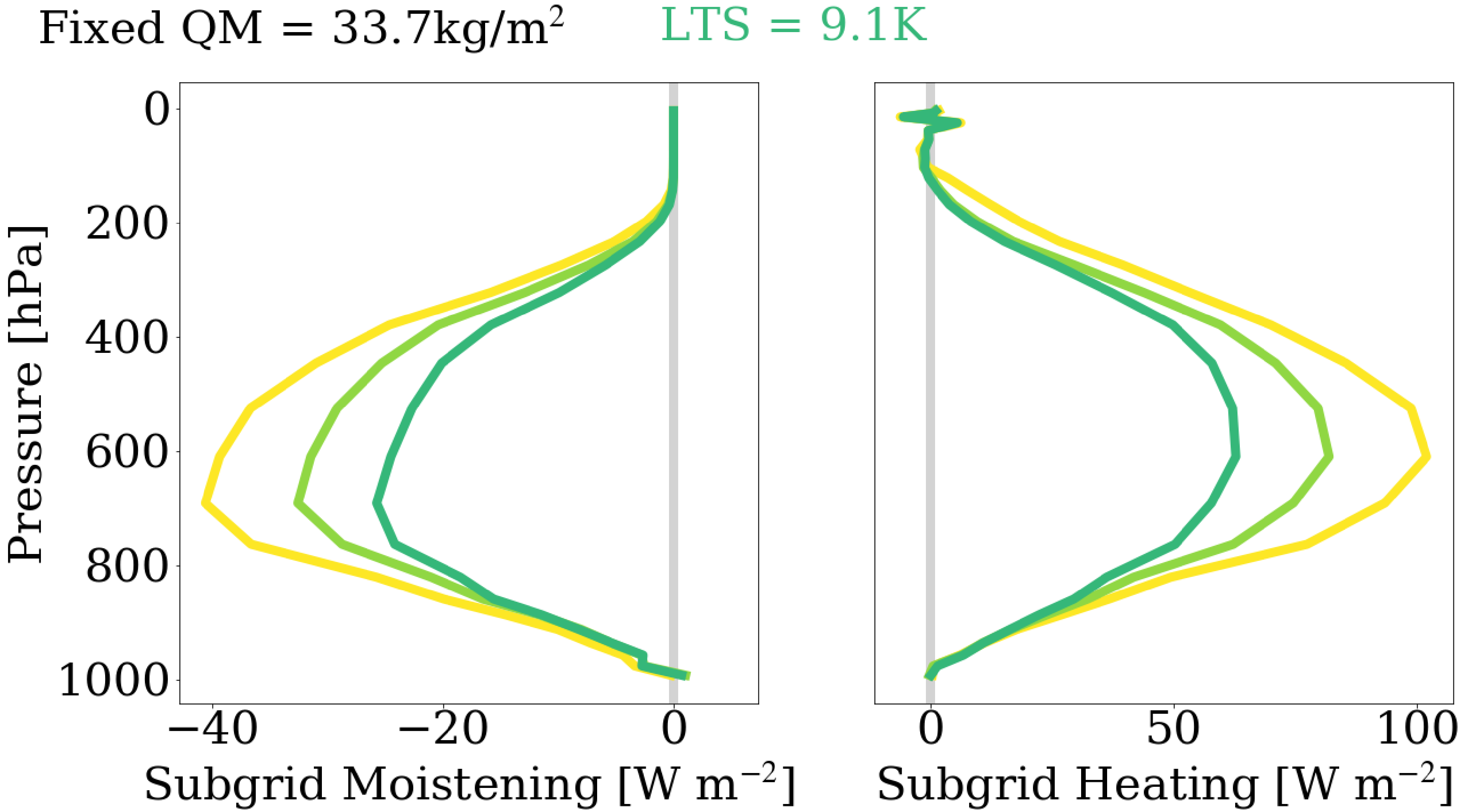
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



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

1) Partial Dependence Plot

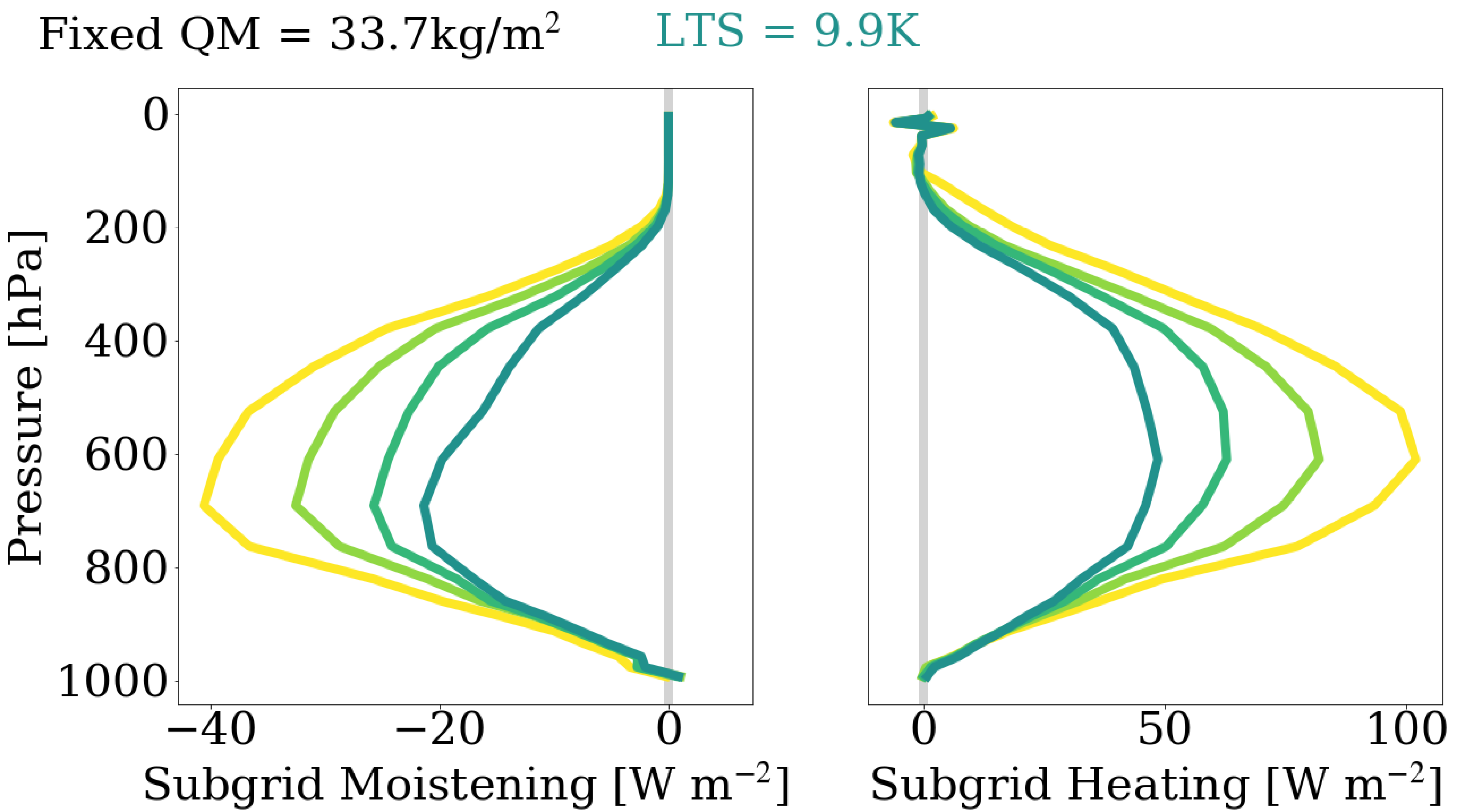
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



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

1) Partial Dependence Plot

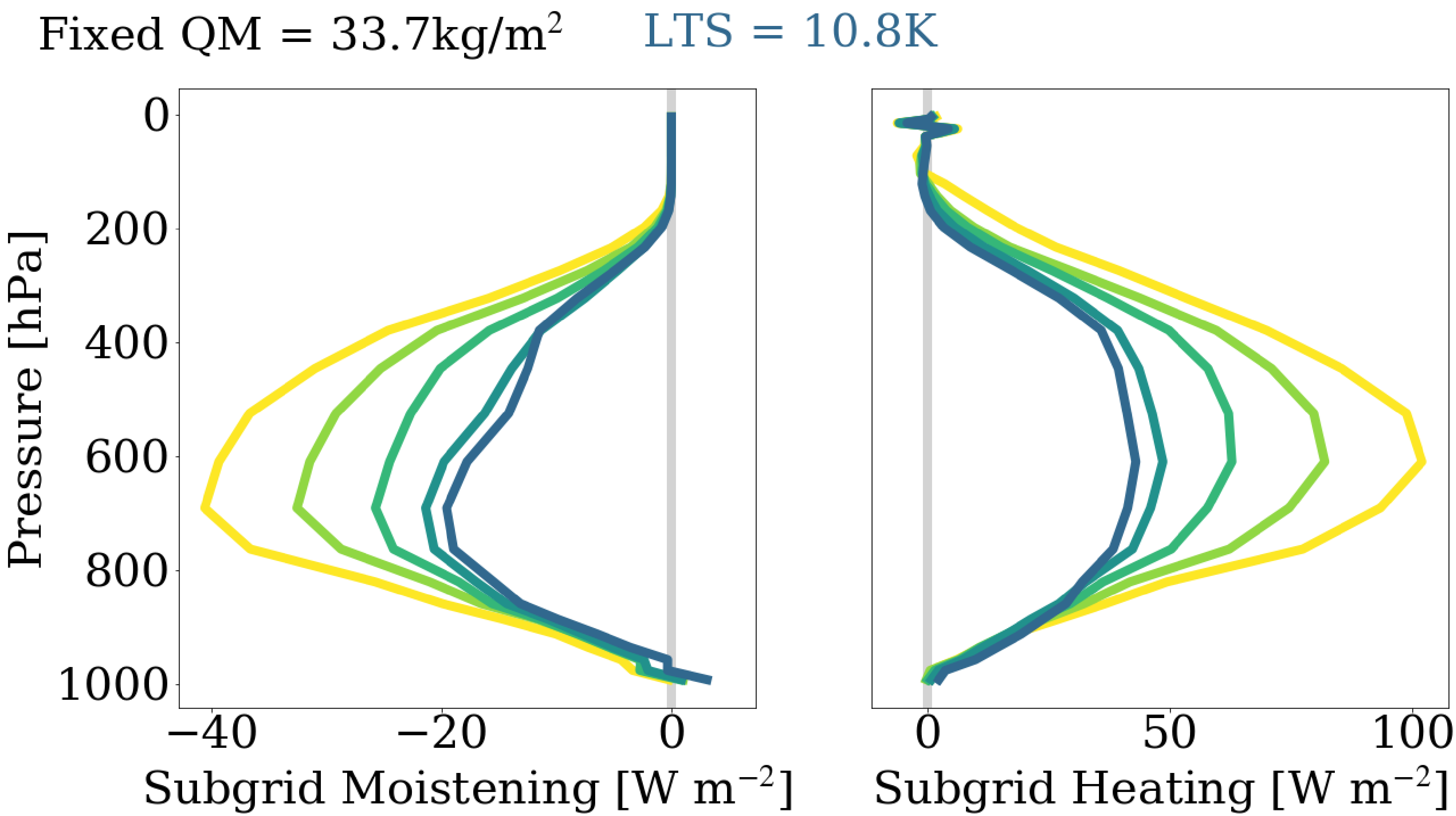
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

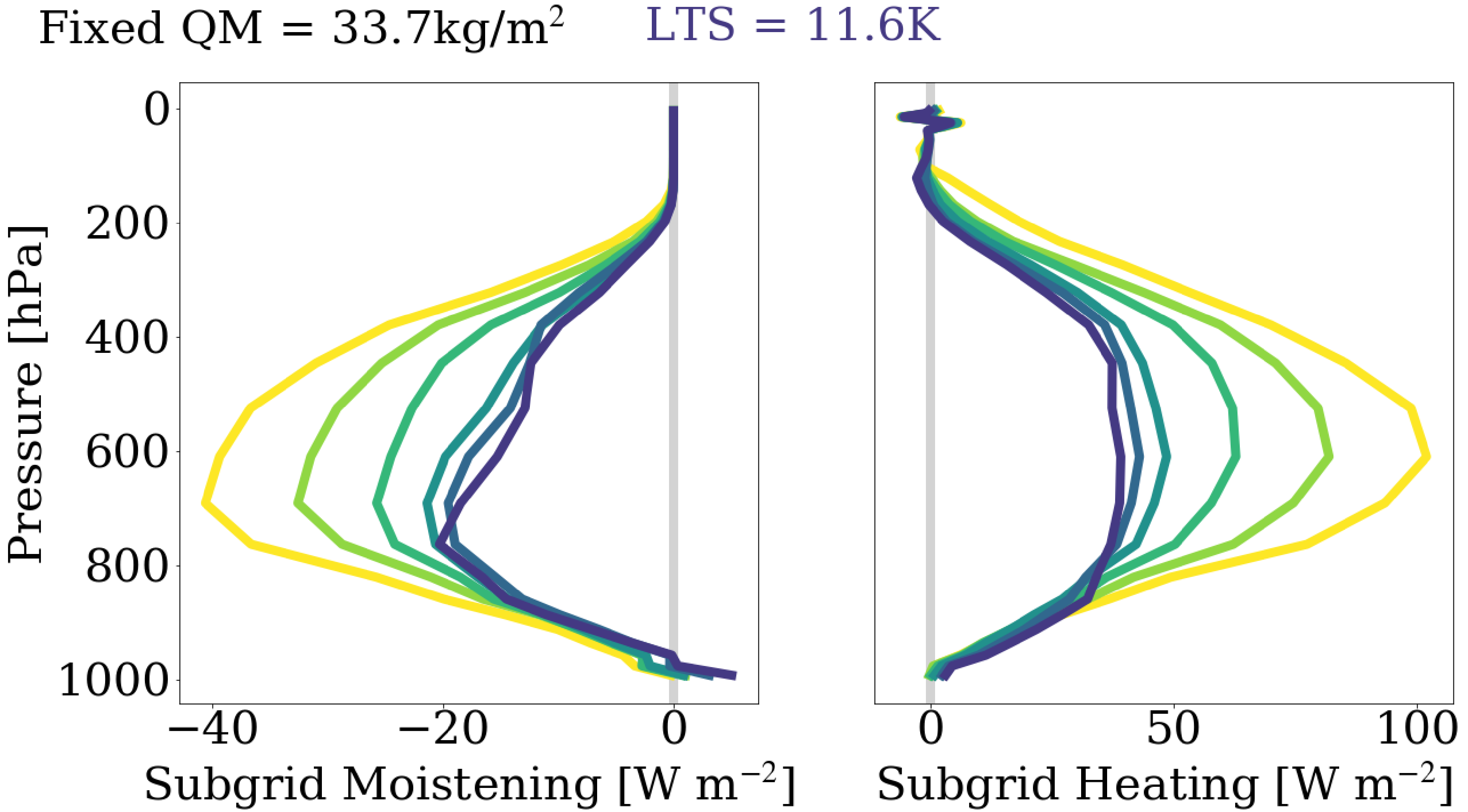
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

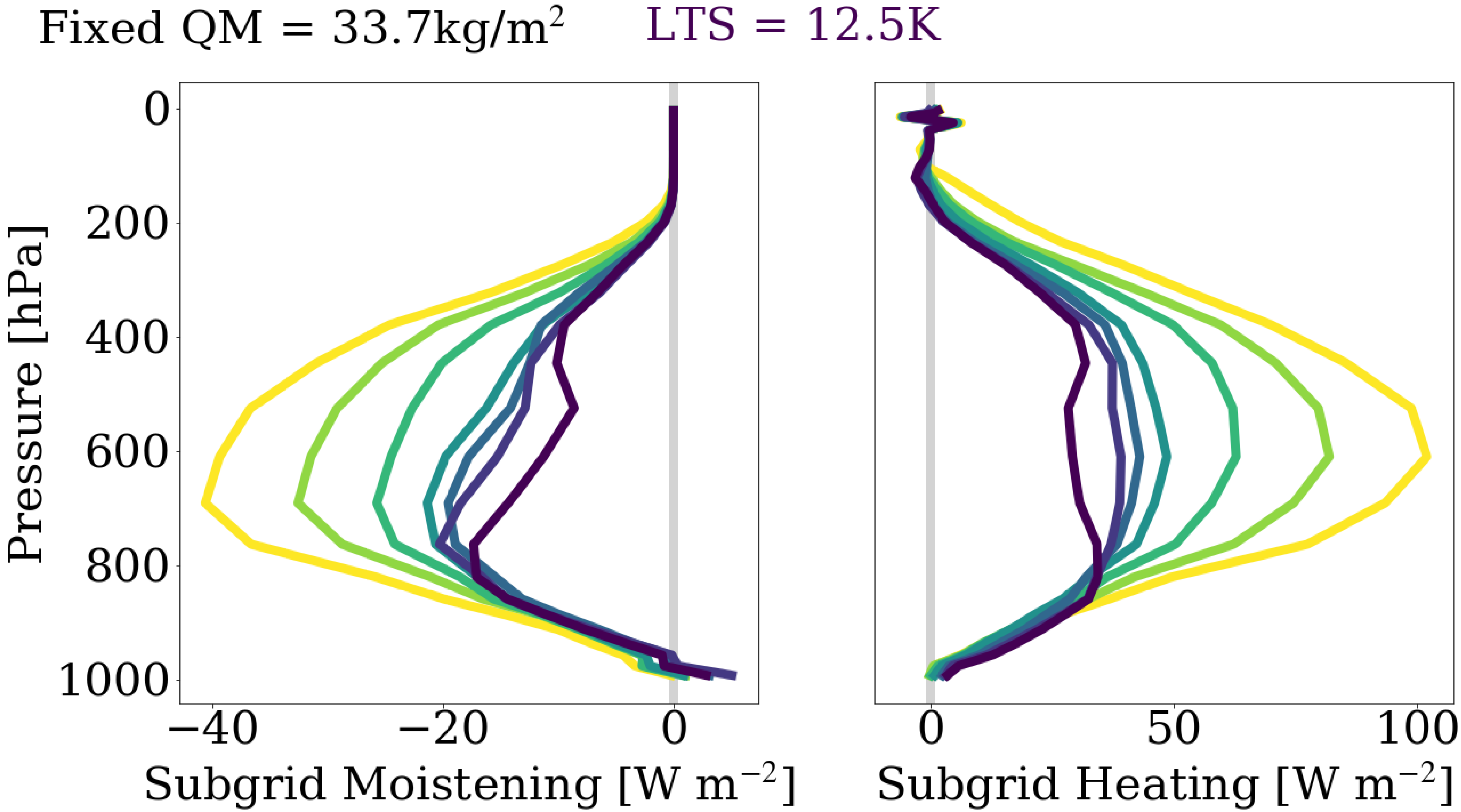
"Shows the marginal effect of 1 or 2 inputs on the predicted outputs"



See: Brenowitz et al. (2020)

1) Partial Dependence Plot

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



<u>See</u>: *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:

$$\boldsymbol{J} \stackrel{\text{def}}{=} \left(\frac{\partial \textbf{Output}}{\partial \textbf{Input}}\right)_{\textbf{Input}_0}$$

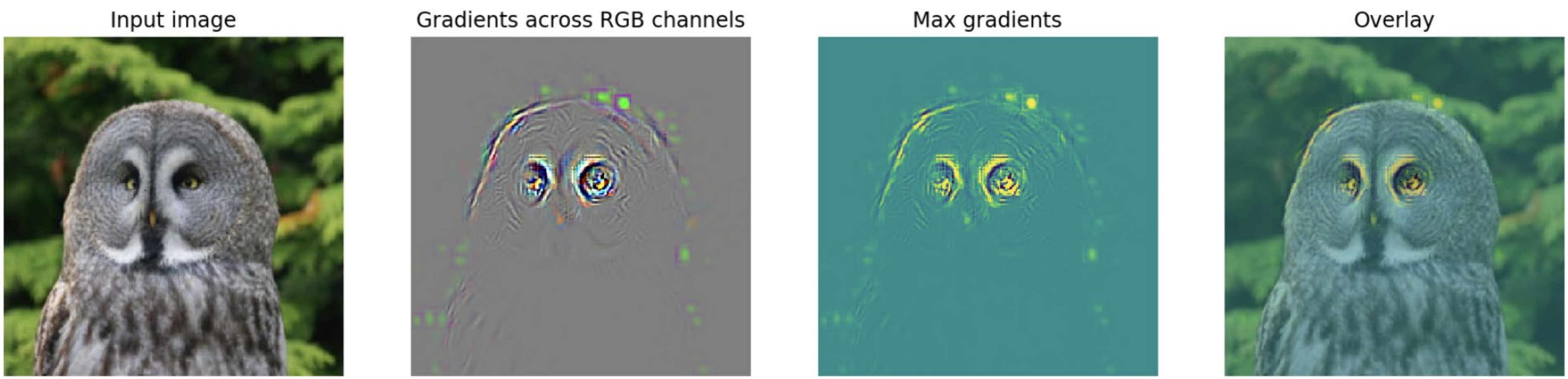
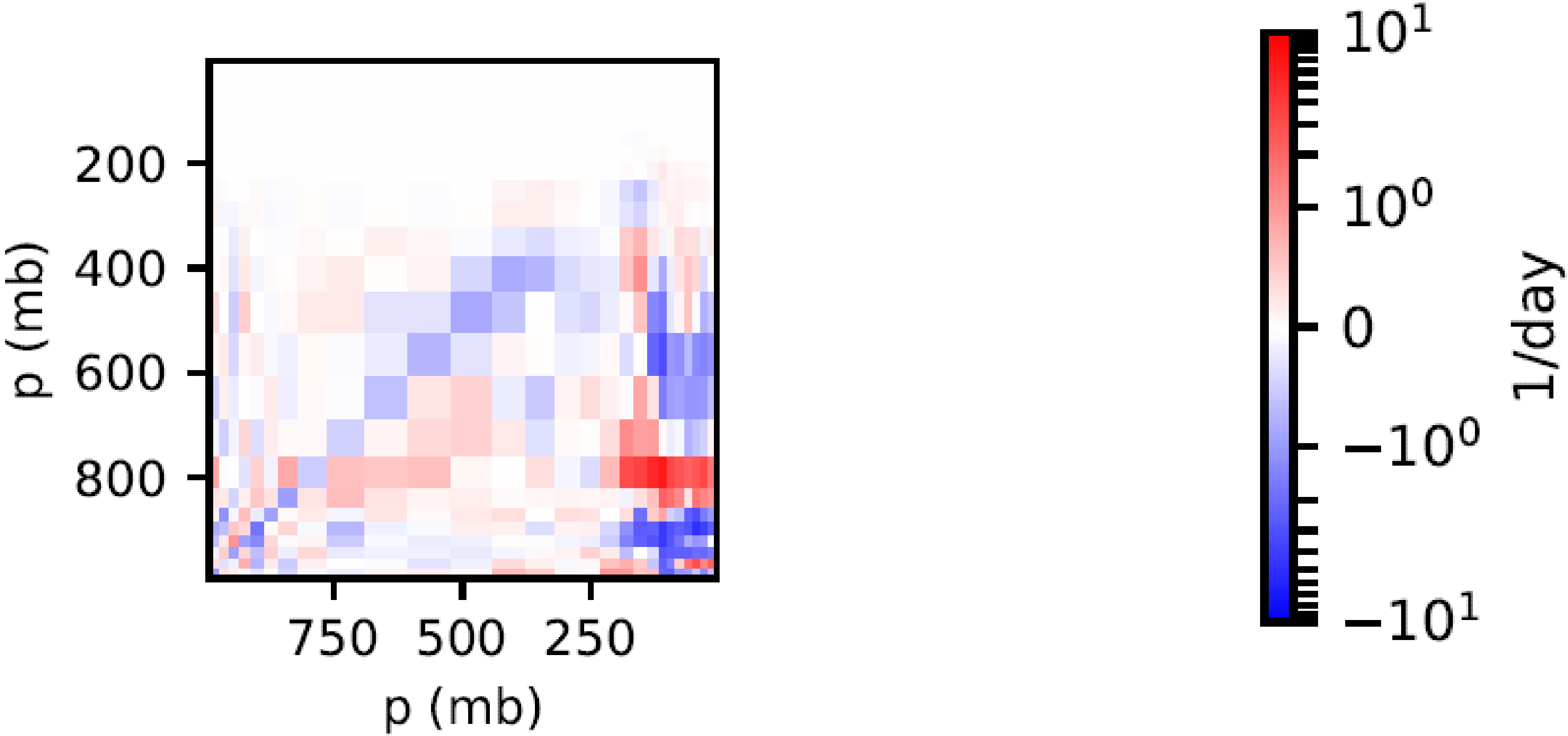


Image source: *flashtorch (Github)* *See:* *Paszke et al. (2017), Springenbert et al. (2015)*

Jacobian = Linear Response Function

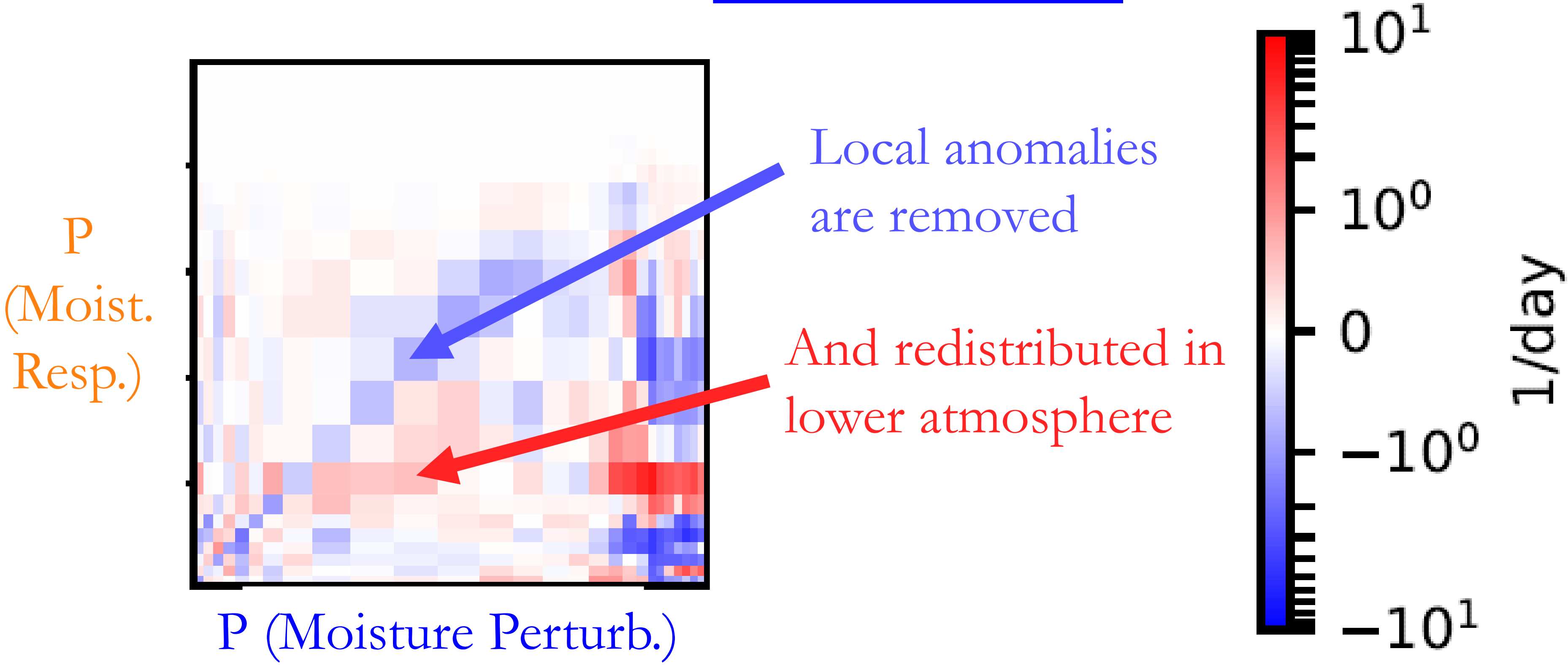
$$\left(\frac{\partial \textbf{Output}}{\partial \textbf{Input}}\right) = \frac{\partial\,(\text{Convective Moistening})}{\partial\,(\text{Moisture})} \quad [1/\text{day}]$$



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

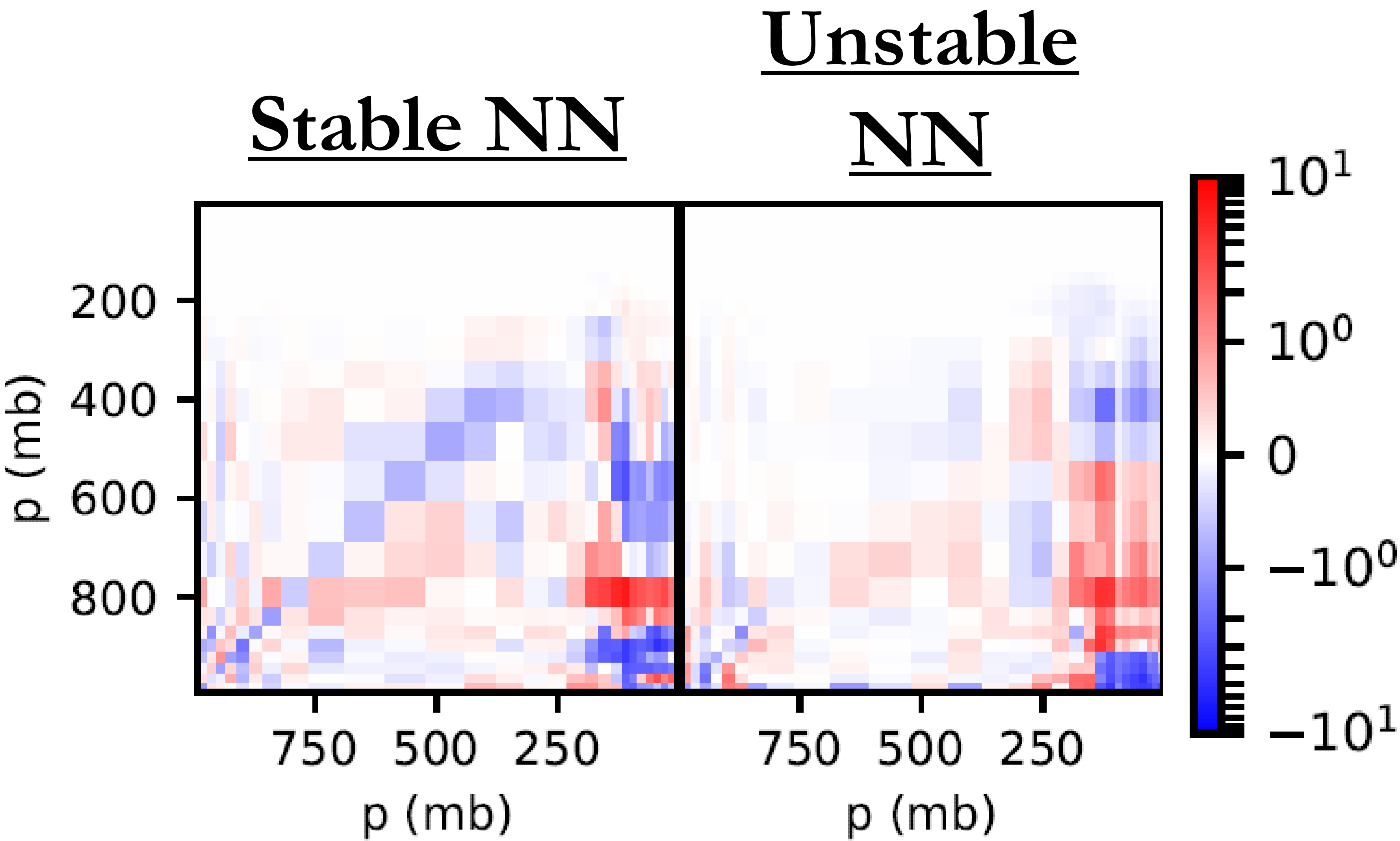
Jacobian = Linear Response Function

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



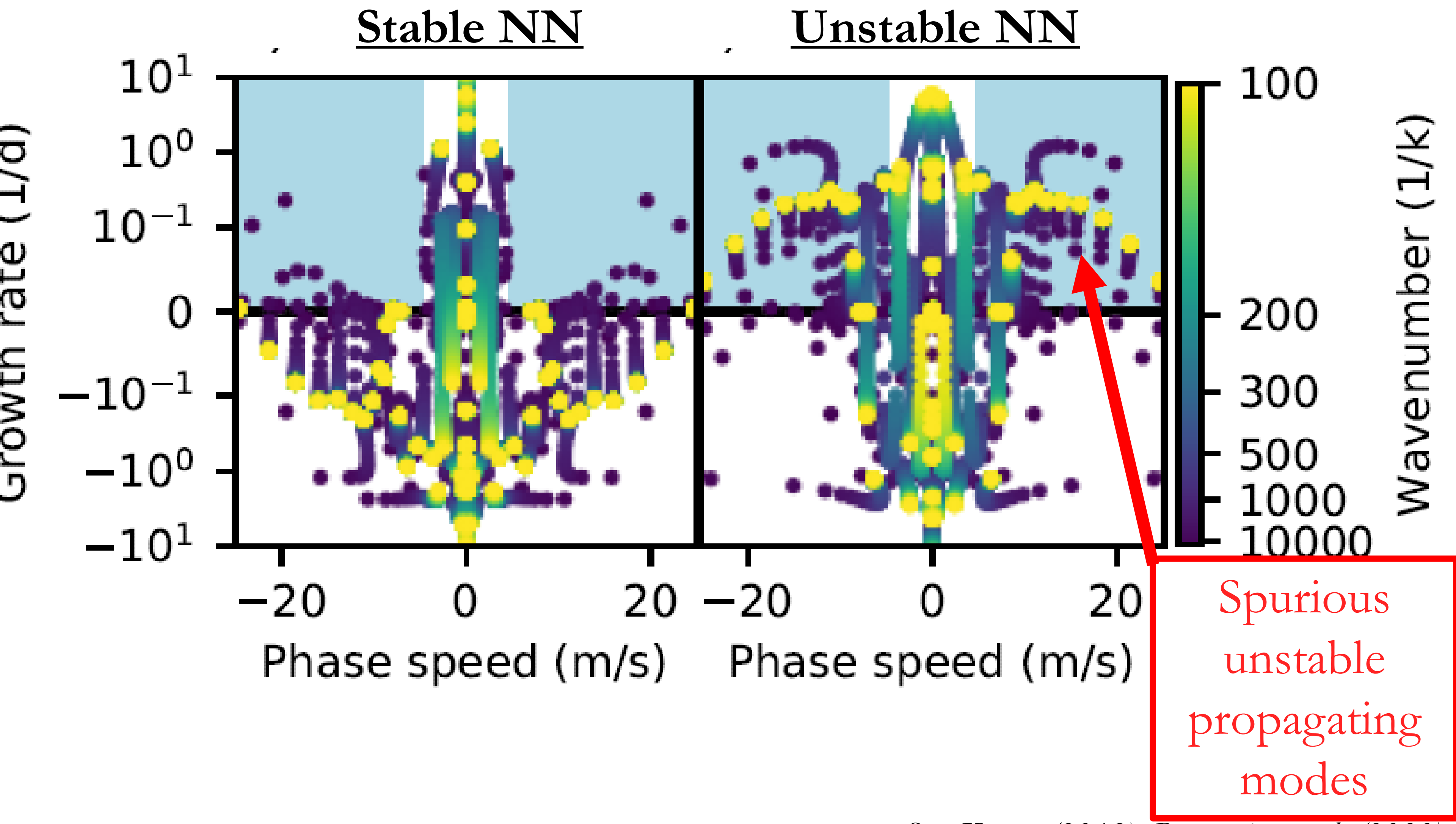
See: Kuang (2018, 2007), Herman and Kuang (2013), Beucler et al. (2018)

Jacobian = Linear Response Function



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

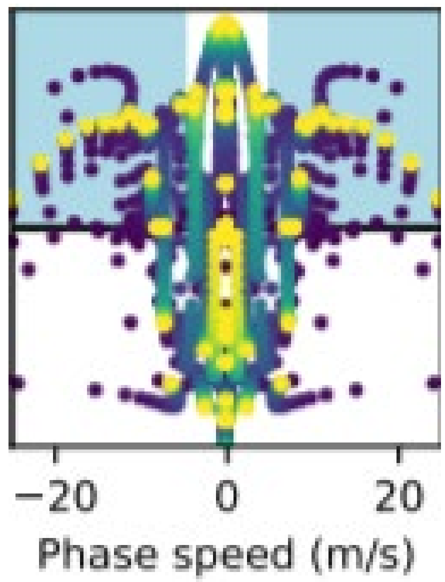
Coupling Linear Response Function to Gravity Waves gives Stability Diagram **offline**



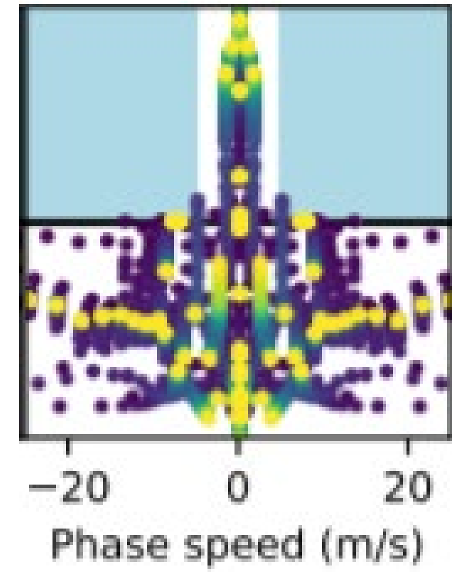
See: Kuang (2018), Brenowitz et al. (2020)

Stability diagram helped stabilize NNs **offline**

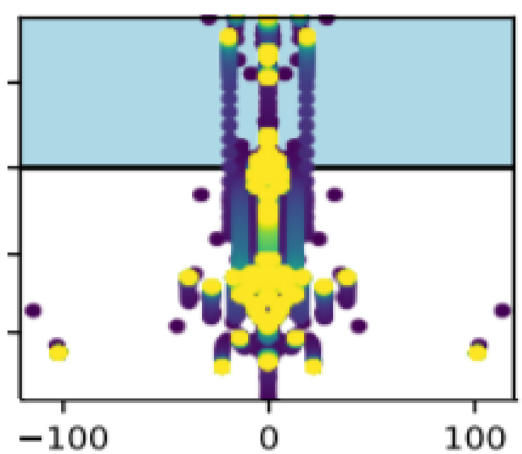
Super Parametrized



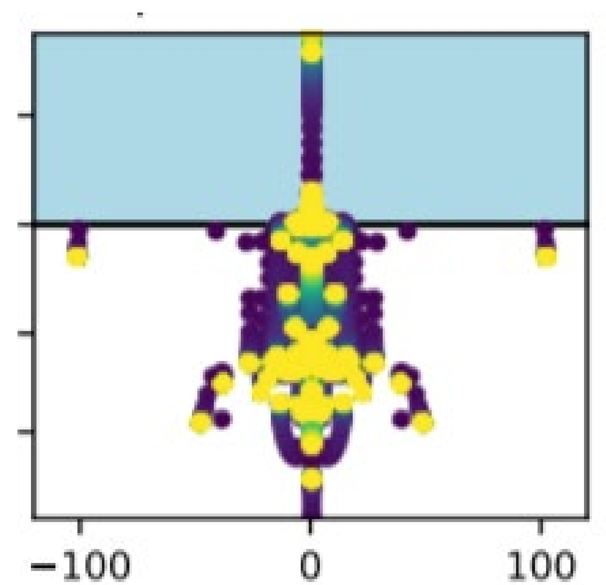
“Regularize” Inputs by adding Gaussian noise



Global Cloud-Resolv.

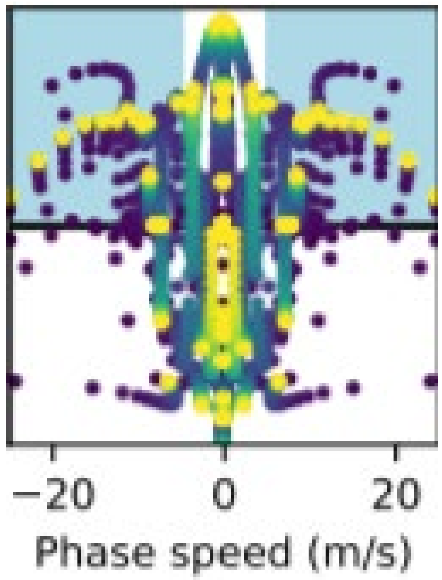


Remove upper-atmos. Inputs

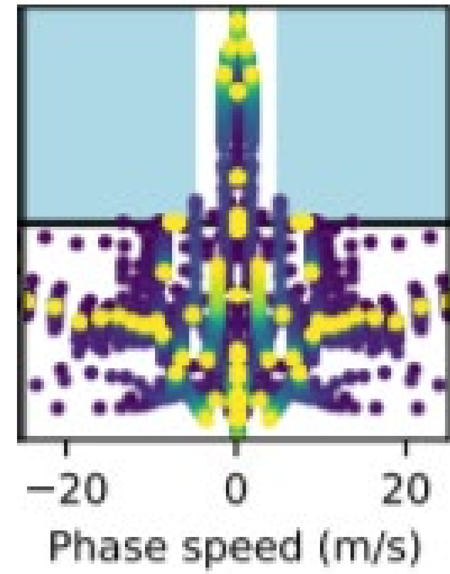


Both stabilized NN ran without crashing for **1month+** when coupled to climate models

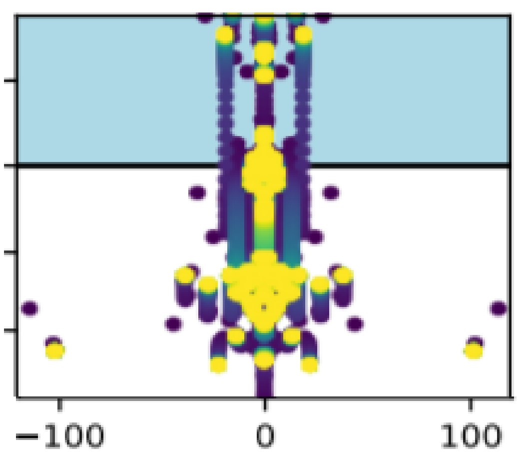
Super Parametrized



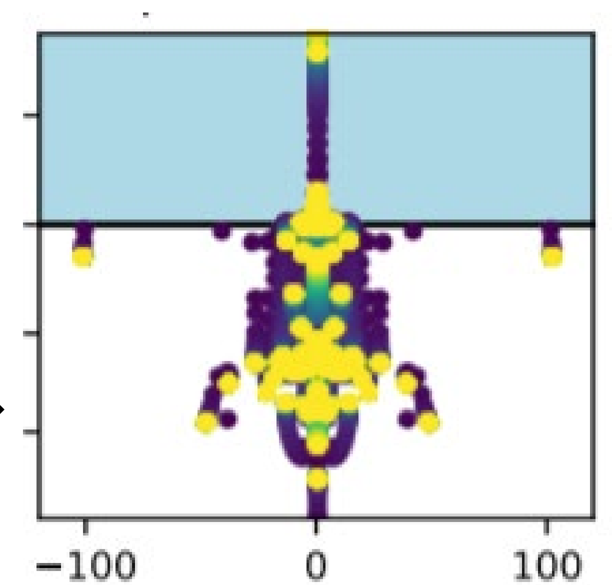
“Regularize” Inputs by adding Gaussian noise



Global Cloud-Resolv.



Remove upper-atmos. Inputs



1
Interpretability
Stability

We can make data-driven parametrizations **interpretable** by tailoring existing interpretability tools to parametrization
BUT Generate spurious O(10W/m^2) or negative precipitation

2
Physical
Constraints

See: Brenowitz et al. (2020) "Interpreting and Stabilizing Machine-learning Parametrizations of Convection." arXiv:2003.06549

3
Generalization

1

Interpretability

Stability

We can make data-driven parametrizations **interpretable** by tailoring existing interpretability tools to parametrization

BUT Generate spurious O(10W/m^2) or negative precipitation

2

Physical Constraints

3

Generalization

Adapting NN's Architecture/Loss Fx

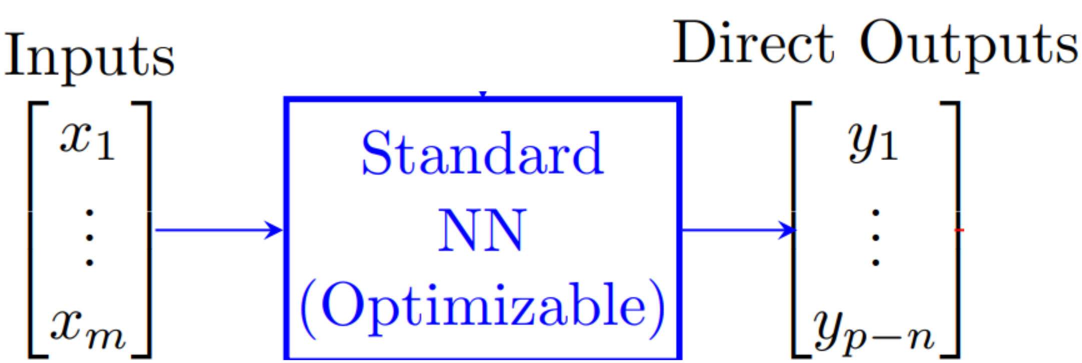
Context: Conservation of mass, energy and radiation

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

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

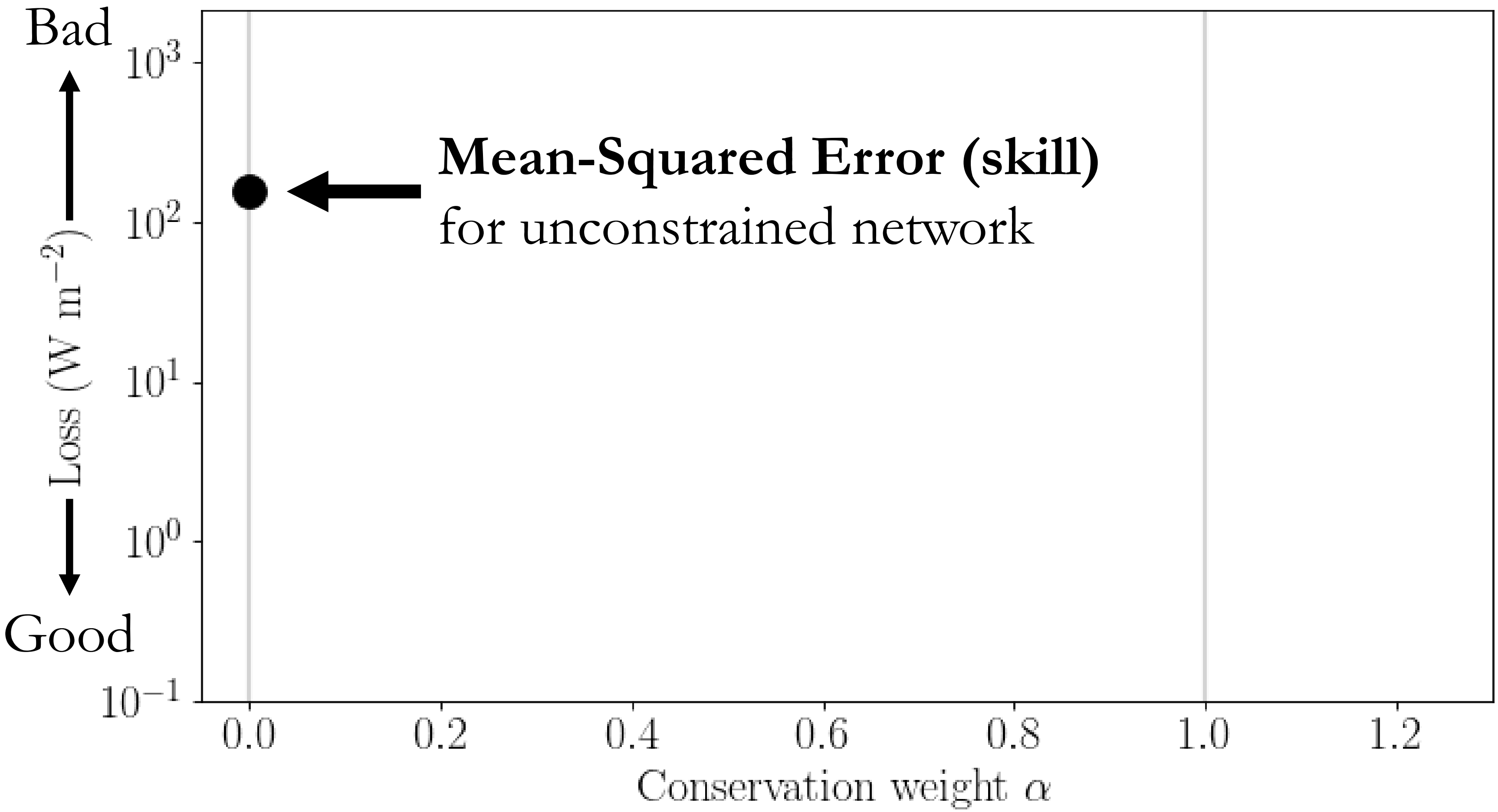
$$\text{Loss} = \alpha\,(\text{Squared residual from conservation laws}) + (1-\alpha)\,(\text{Mean squared error})$$

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



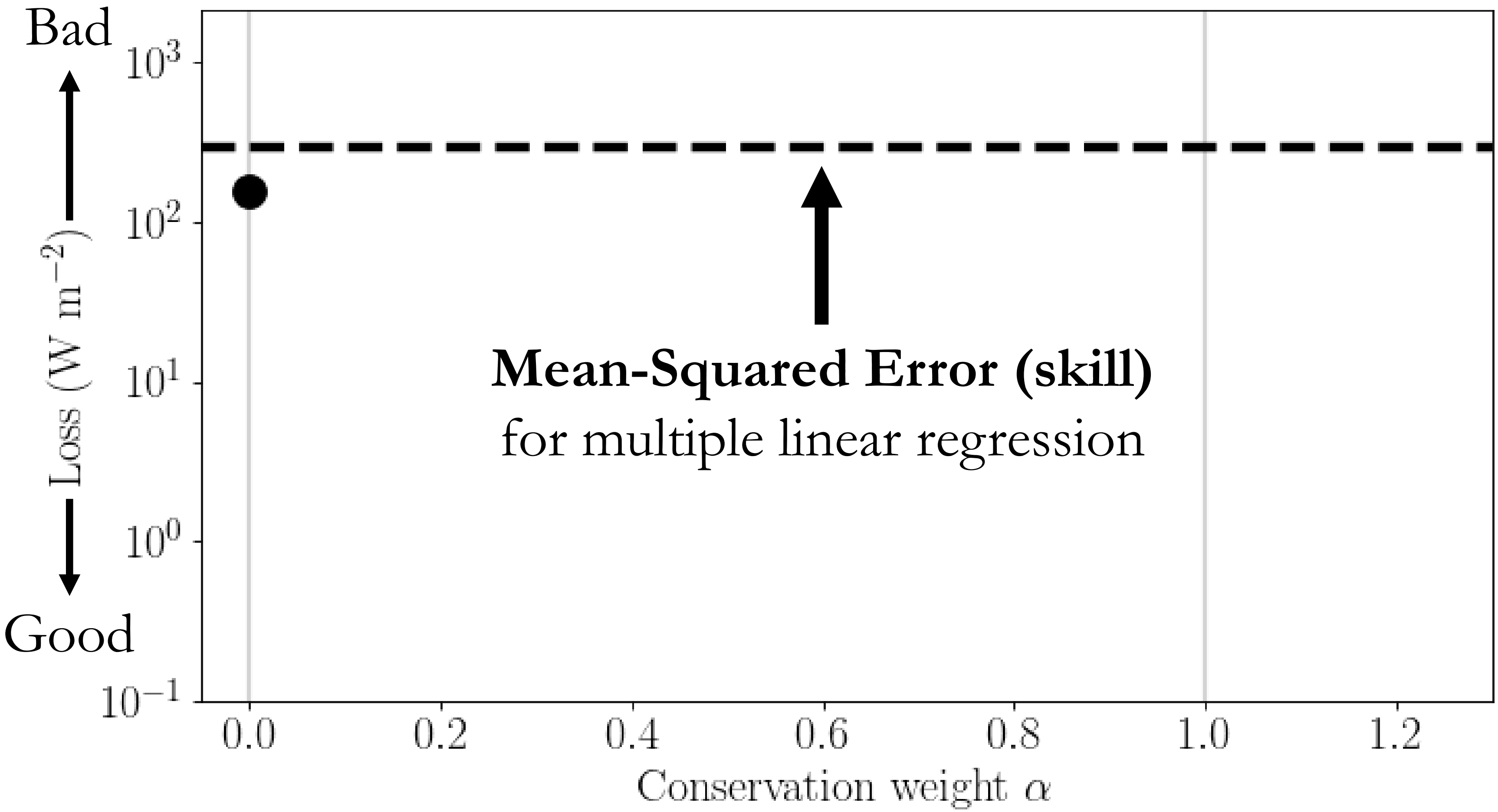
Loss: Trade-off between **performance** and **physical constraints**

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



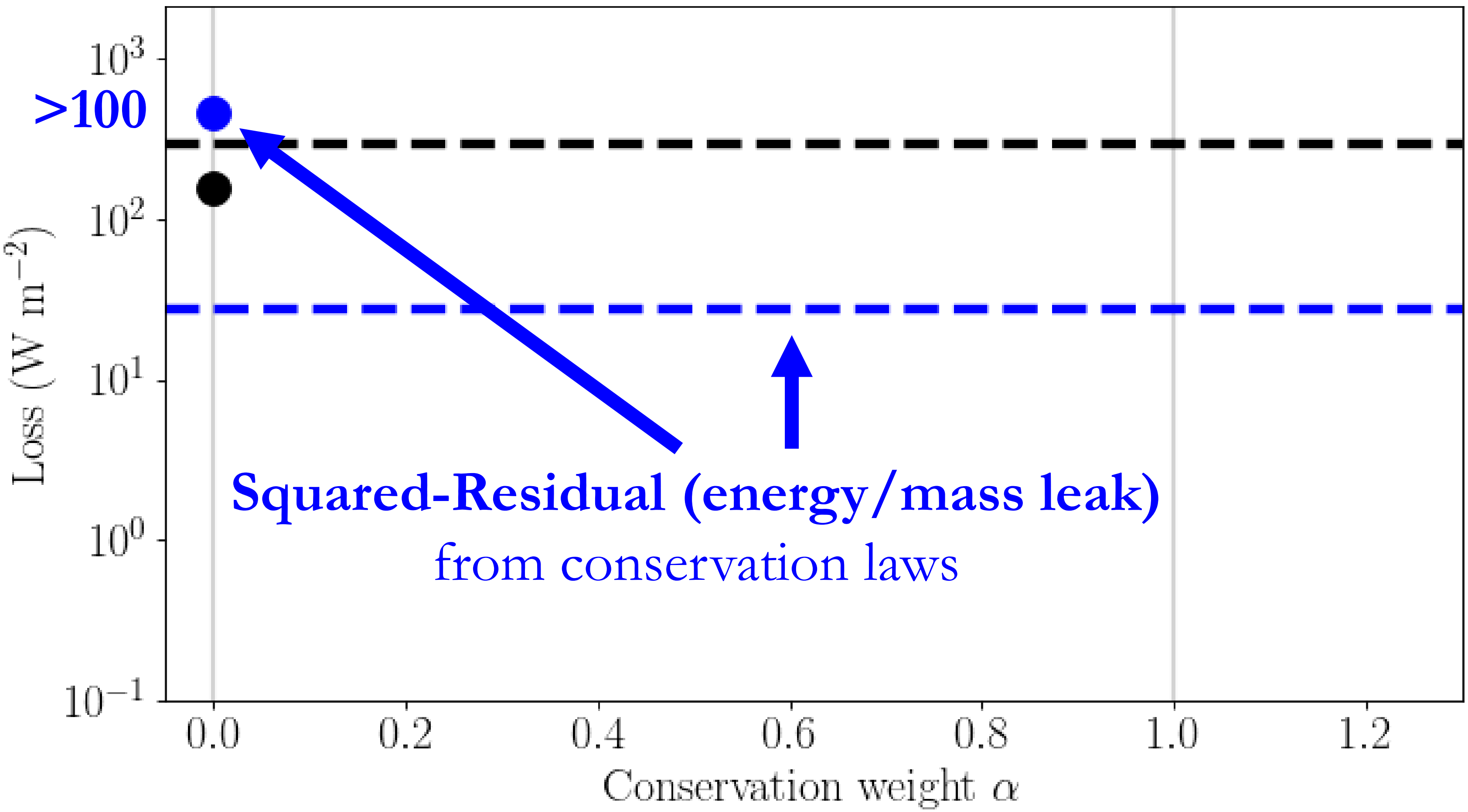
<u>Loss</u>: Trade-off between **performance** and **physical constraints**

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



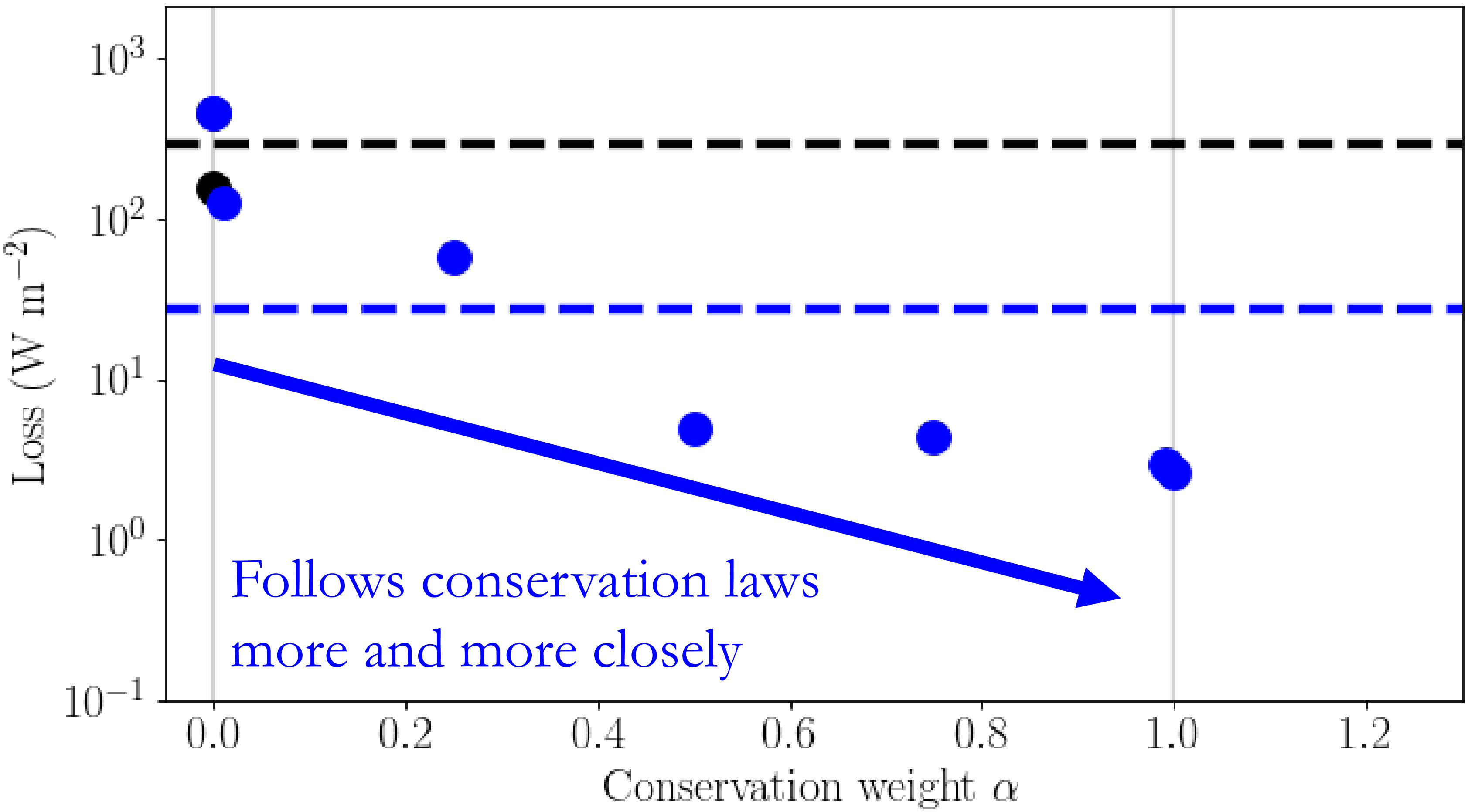
<u>Loss</u>: Trade-off between **performance** and **physical constraints**

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



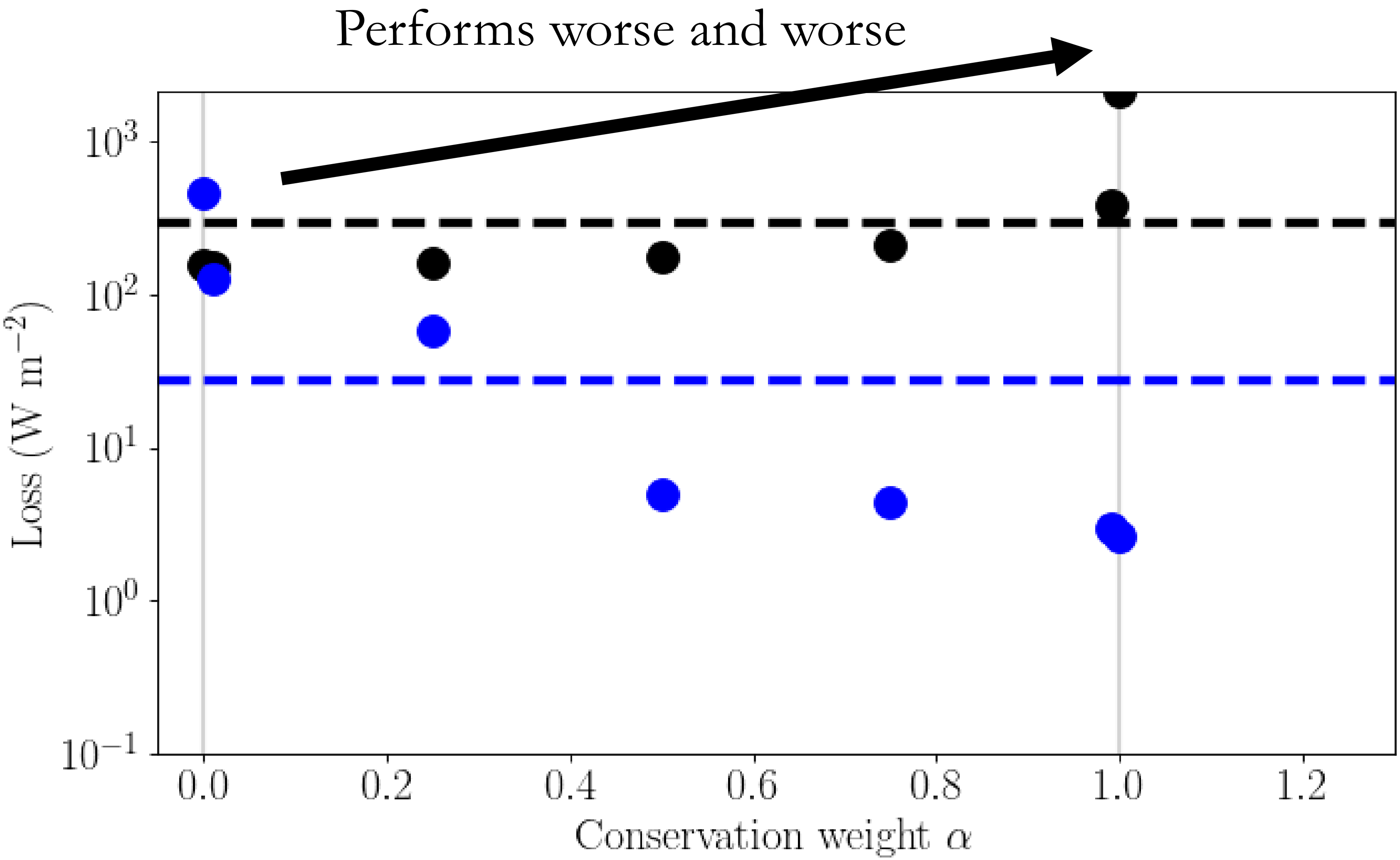
<u>Loss</u>: Trade-off between **performance** and **physical constraints**

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



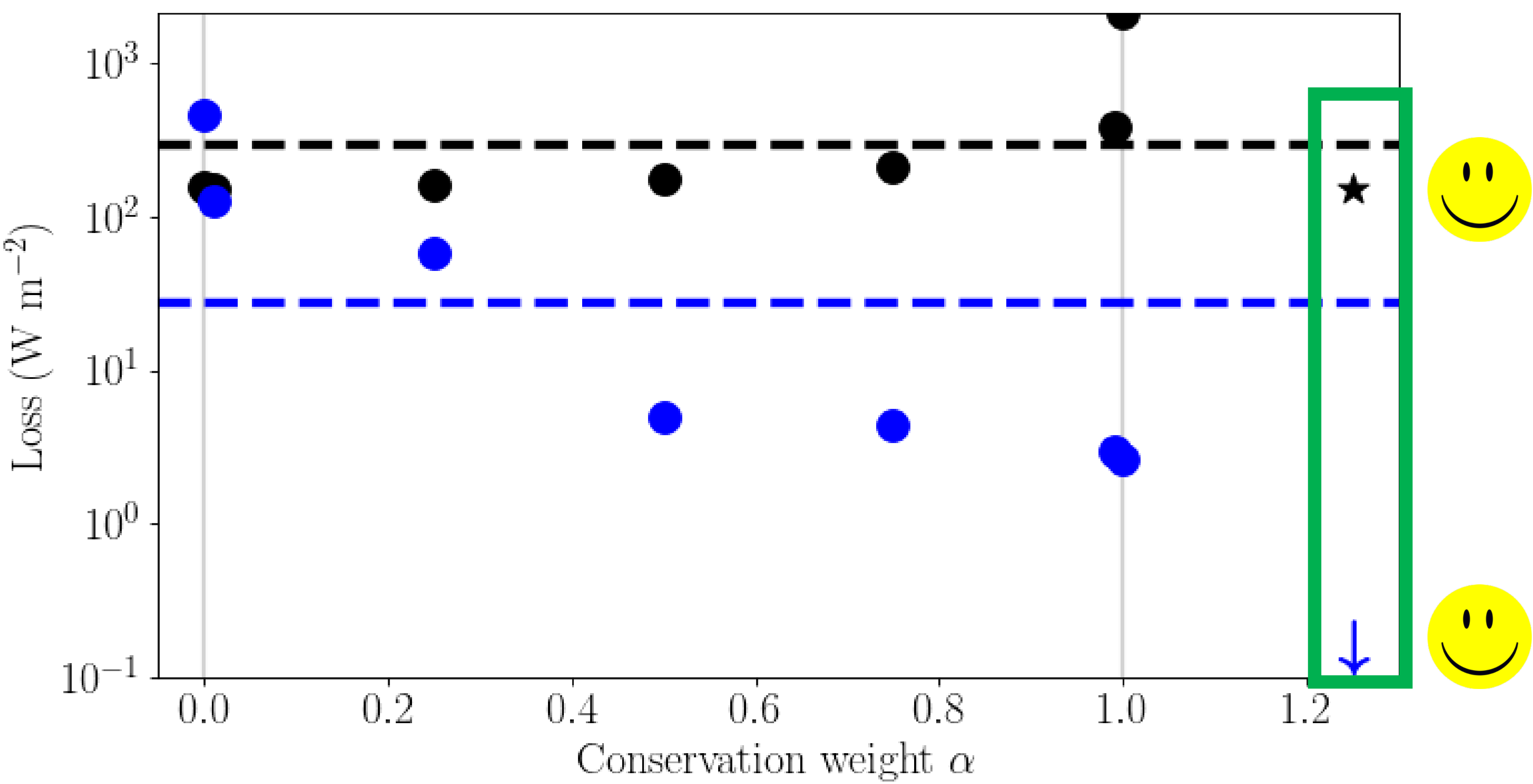
<u>Loss</u>: Trade-off between **performance** and **physical constraints**

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



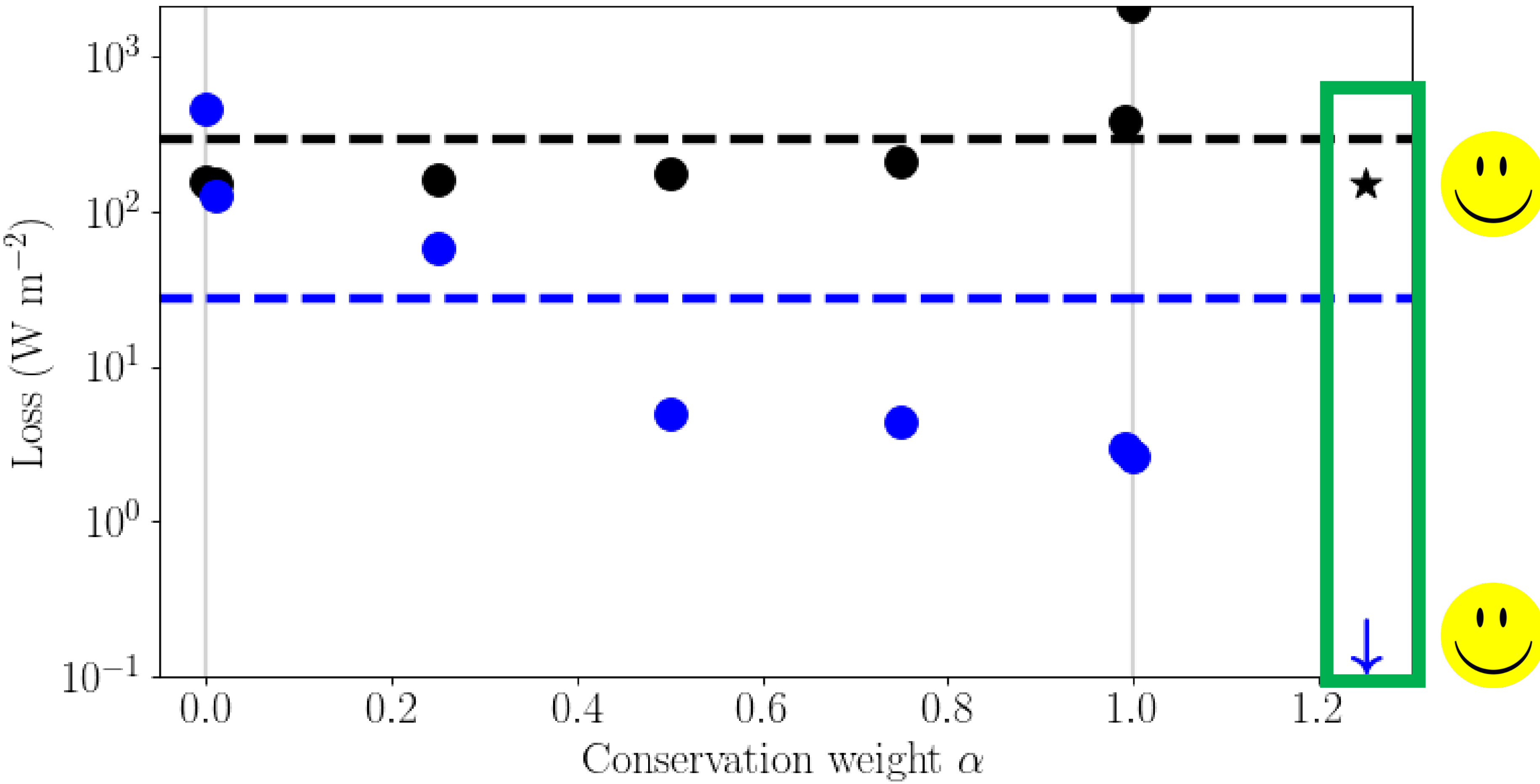
Loss: Trade-off between **performance** and **physical constraints**

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



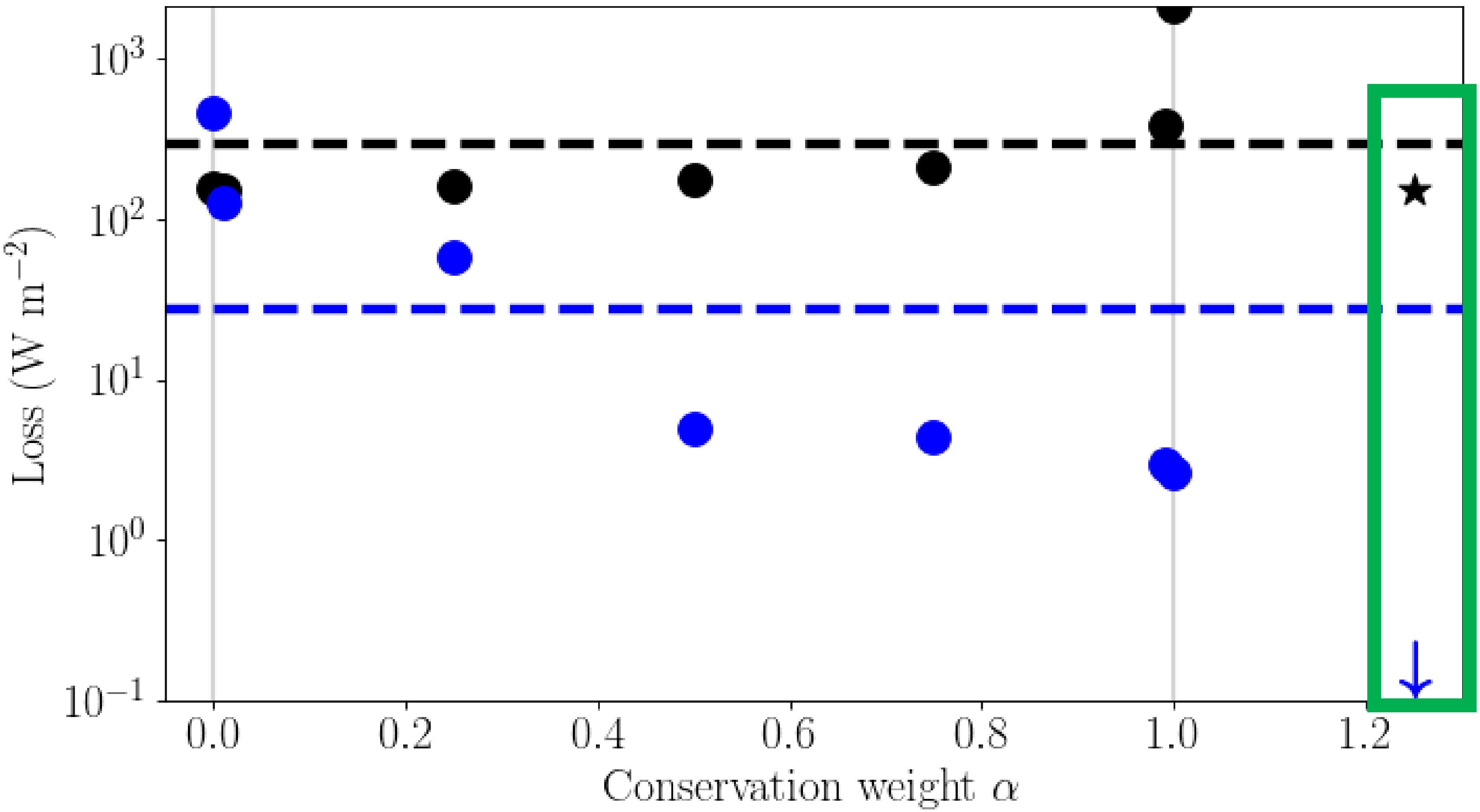
<u>Loss</u>: Trade-off between **performance** and **physical constraints**

<u>Architecture</u>: **Constraints enforced & competitive performance**



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

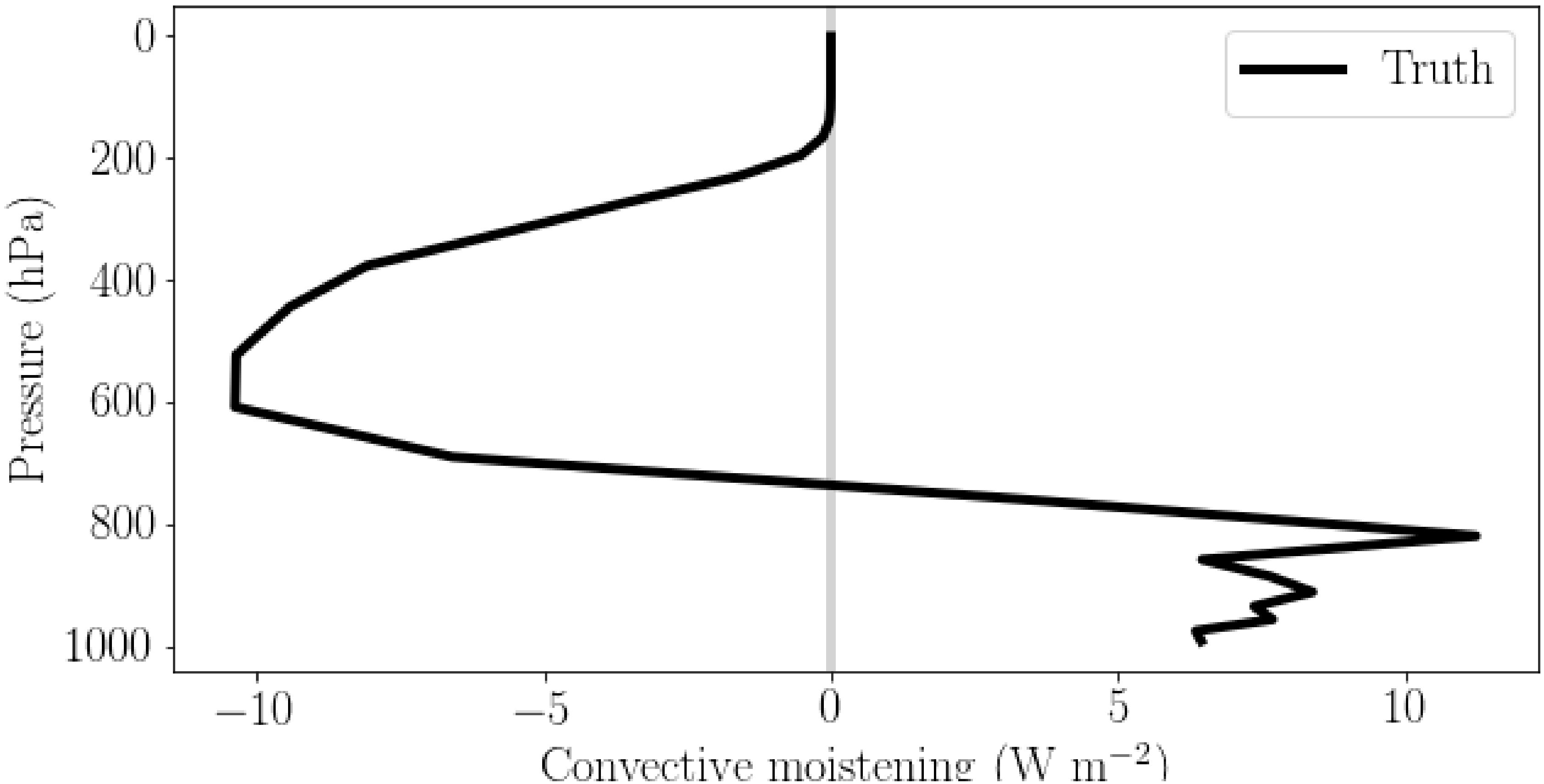
Architecture: **Nonlinear** **Constraints enforced & competitive perf.**
& Biases mitigated via loss function



Problem: Even when physically constrained, NNs fail to generalize

Problem: Even when physically constrained, NNs fail to generalize

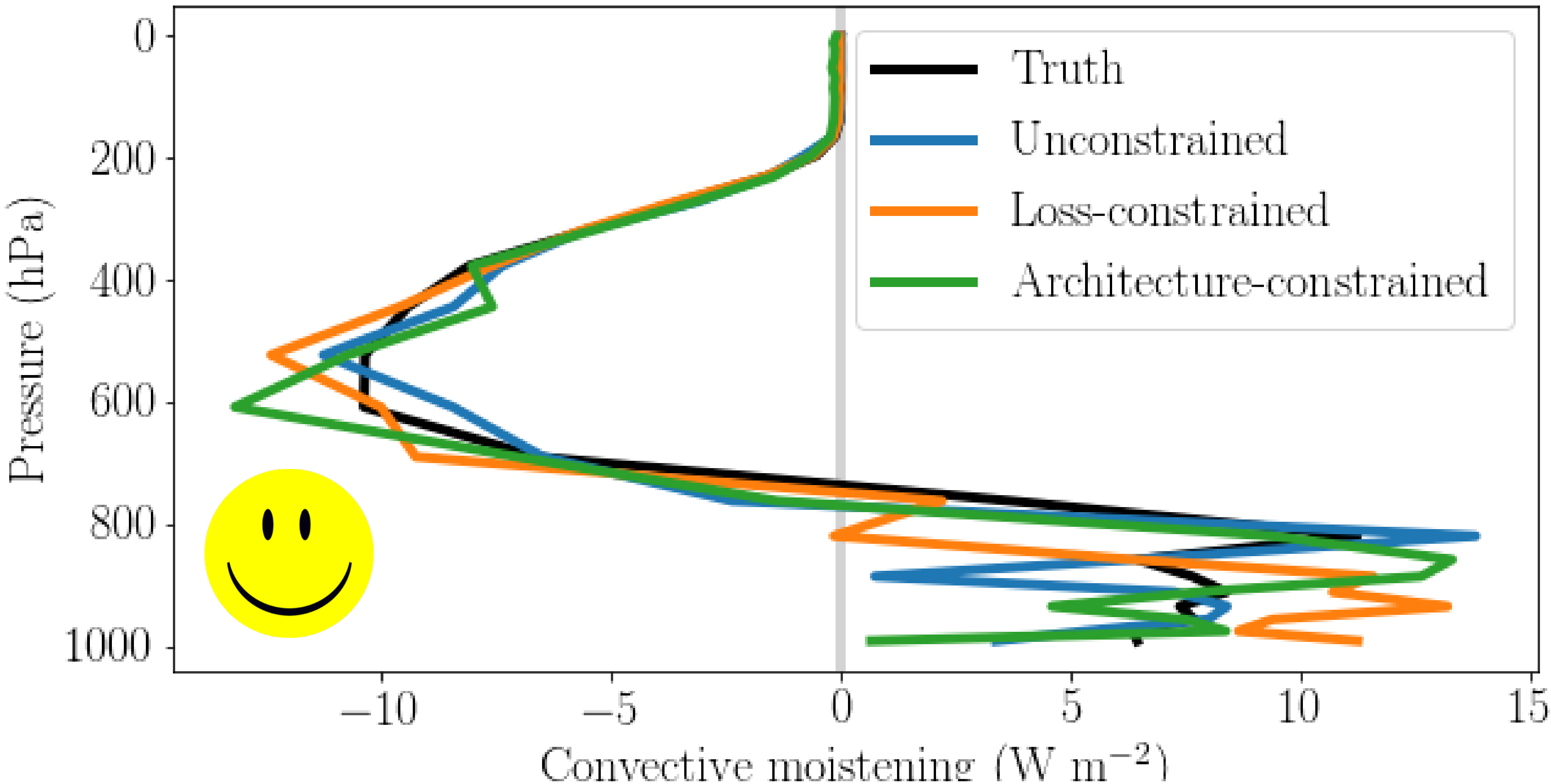
Daily-mean Tropical prediction in reference climate



See: Beucler et al. (2019)

Problem: Even when physically constrained, NNs fail to generalize

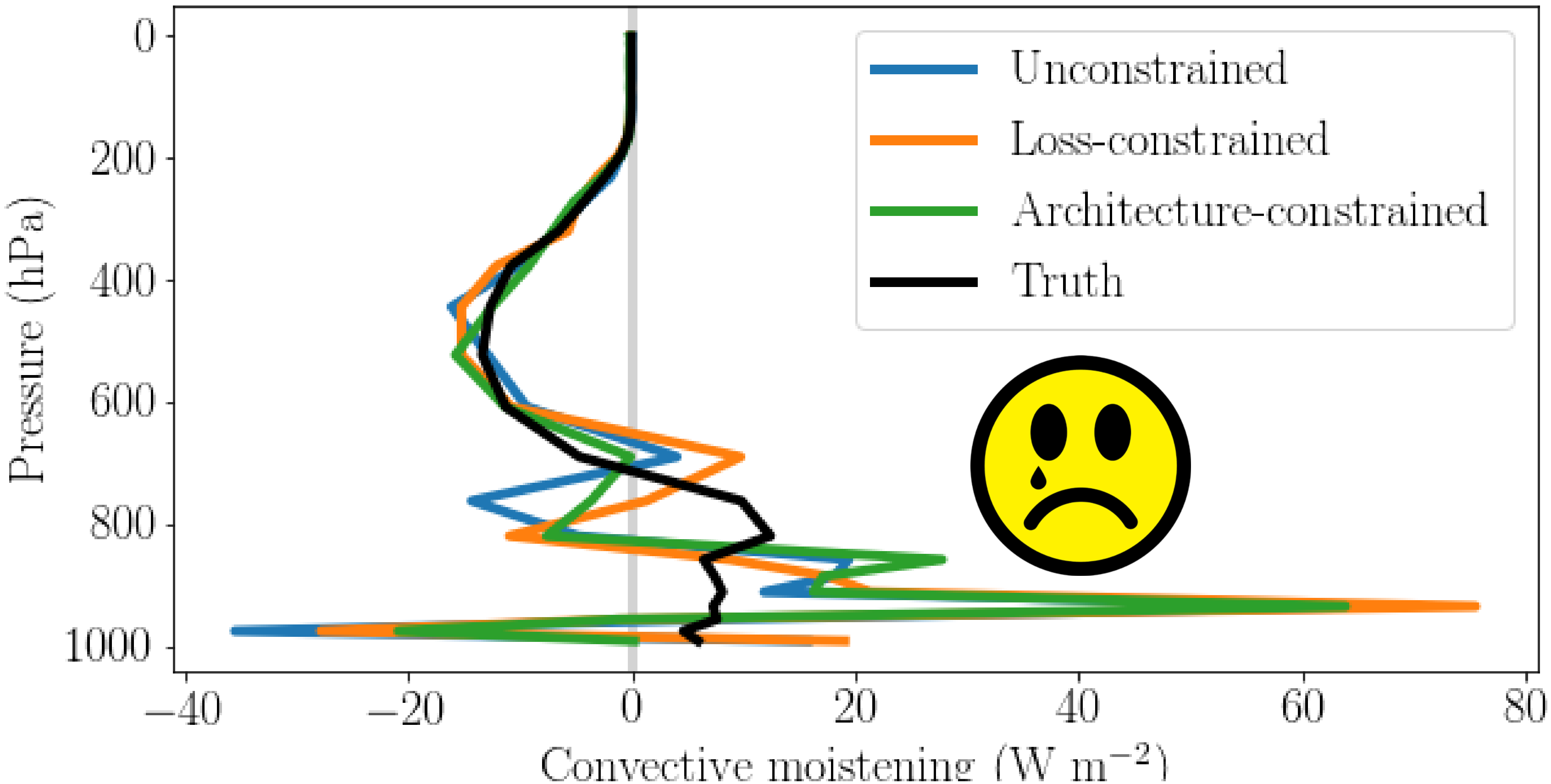
Daily-mean Tropical prediction in reference climate



<u>See:</u> *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)

1
Interpretability
Stability

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

2
Physical
Constraints

See: Beucler et al. (2019)
"Enforcing Analytic Constraints in Neural-Networks Emulating Physical Systems."
arXiv:1909.00912

3
Generalization

1
Interpretability
Stability

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

2
Physical
Constraints

3
Generalization

Generalization Experiment: Idea = Break the model

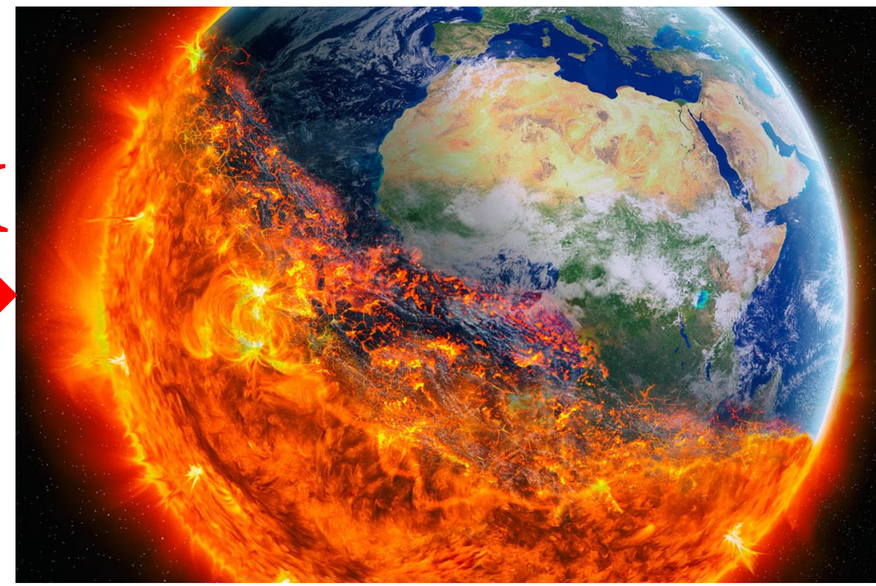
Image: IT Biz Advisor

Generalization Experiment: Uniform +8K warming

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

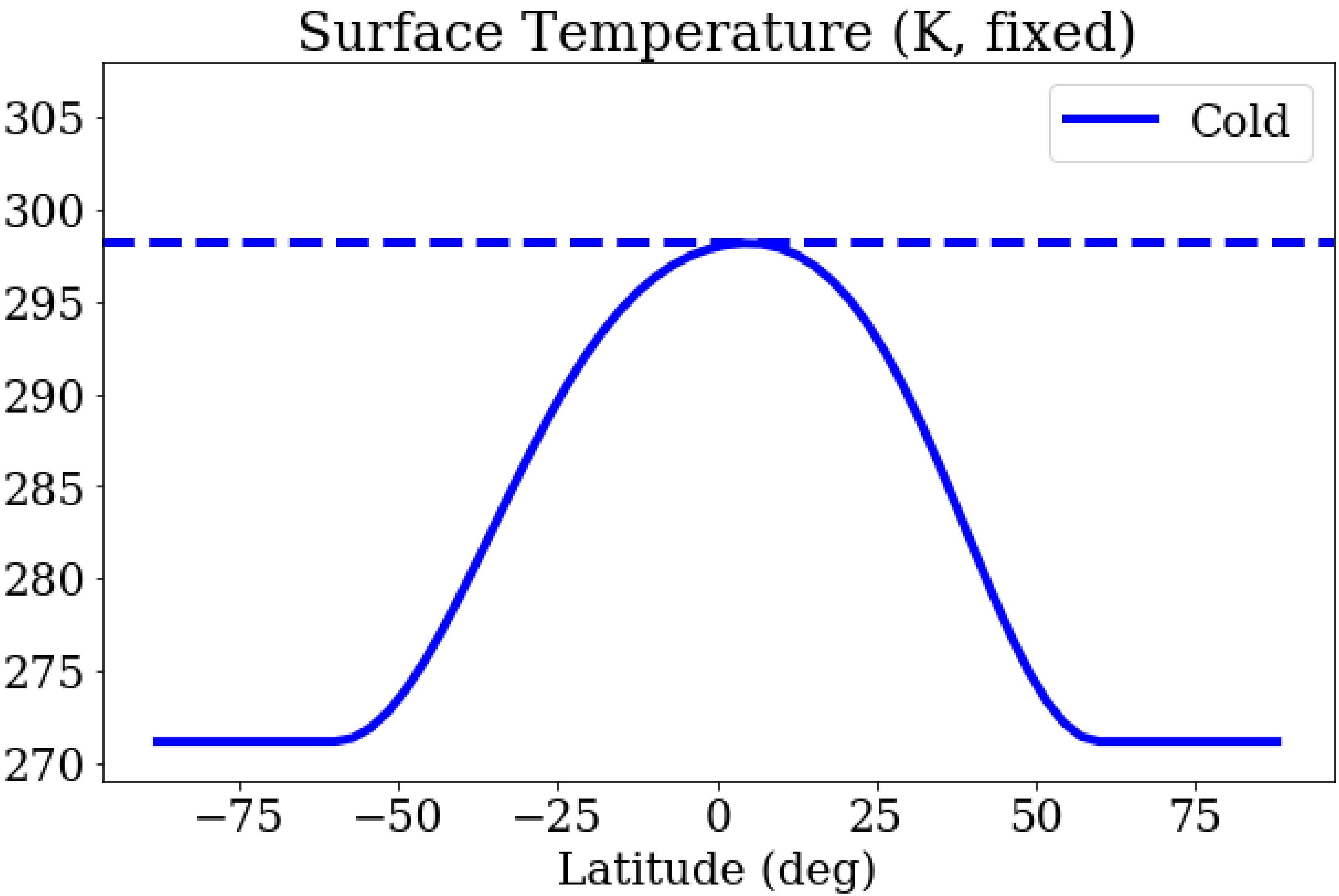
Test
on warm aquaplanet simulation
(Warm, +4K)

+8K

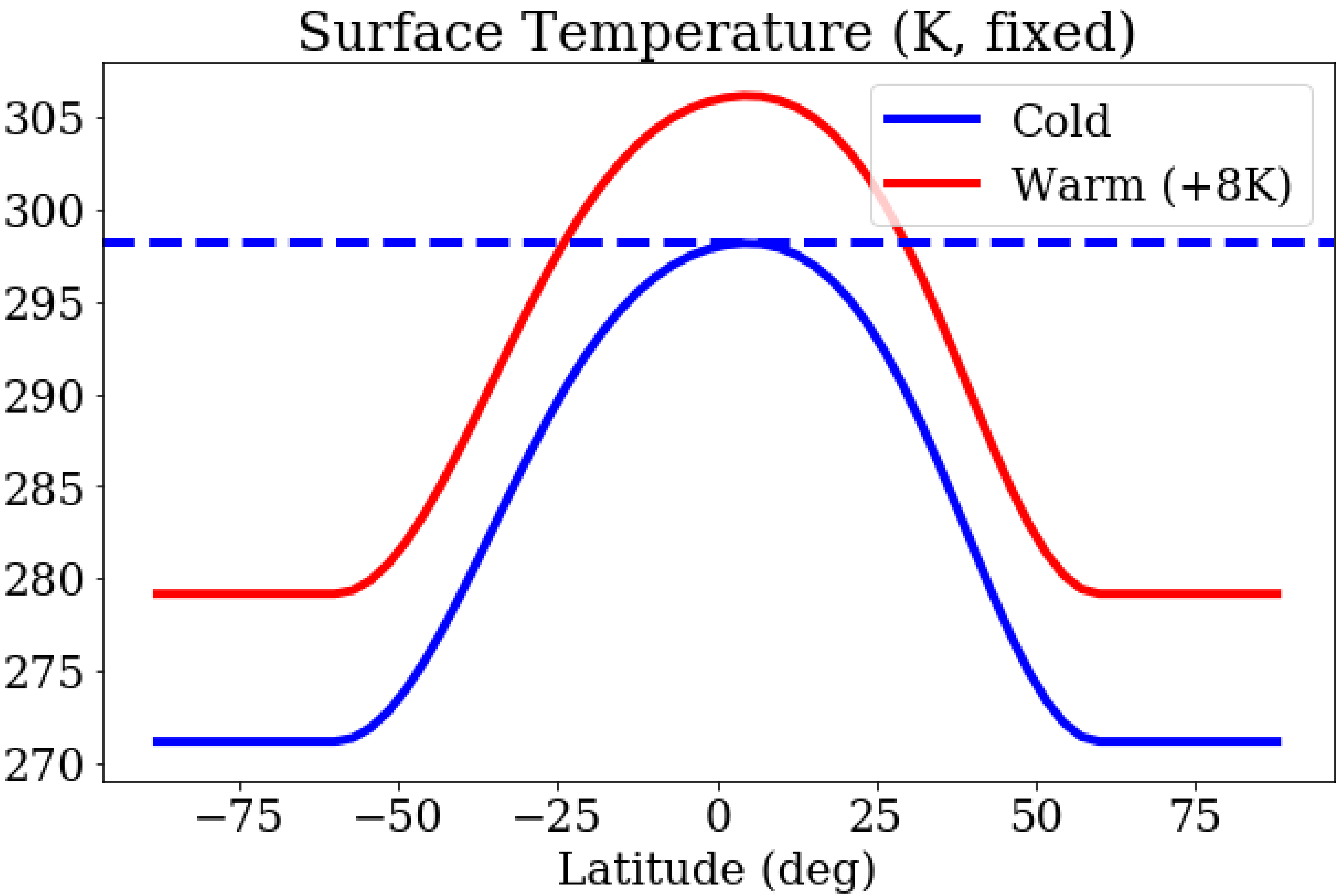


<u>Images</u>: Rashevskyi Viacheslav, Sebastien Decoret

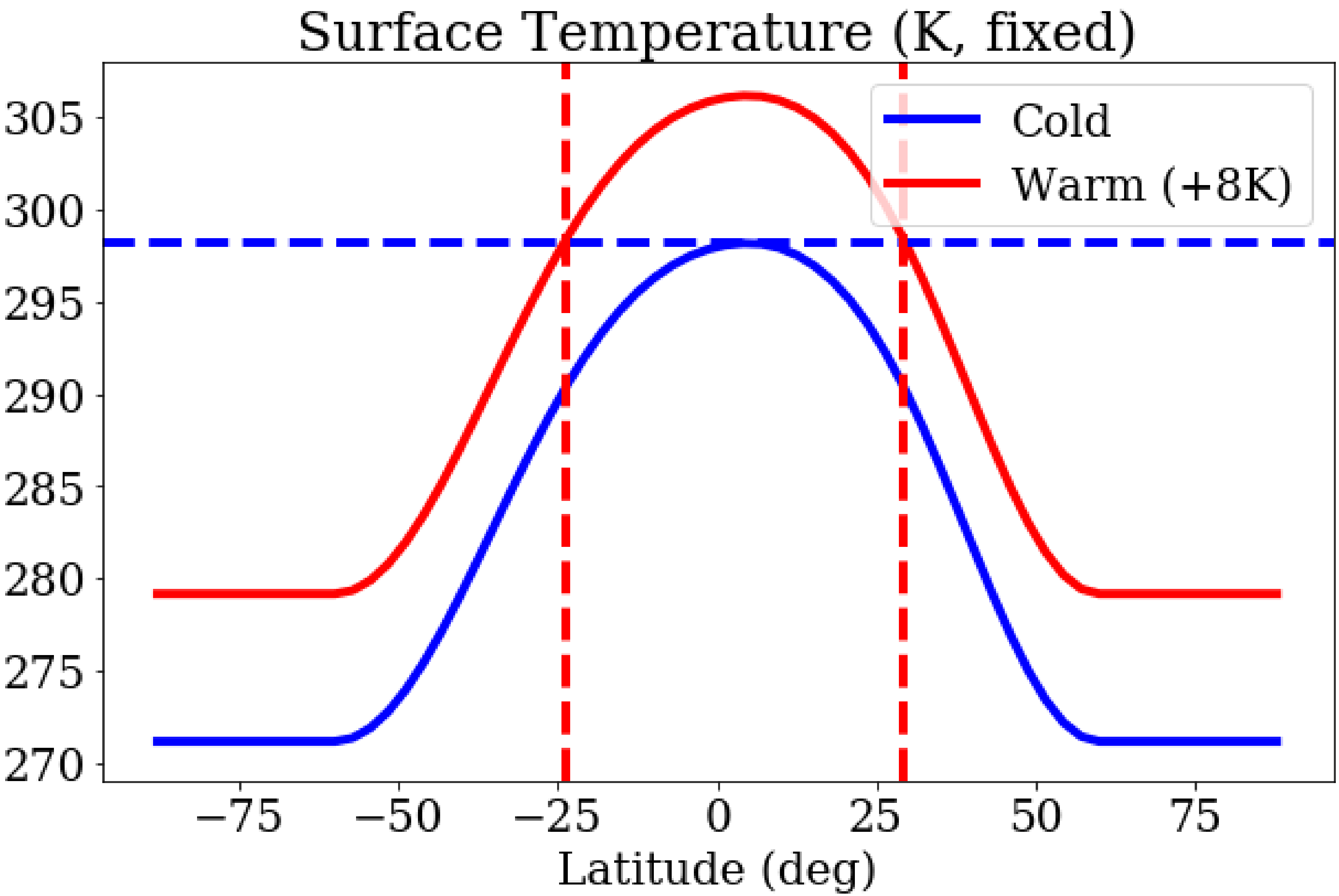
Generalization Experiment: Uniform +8K warming



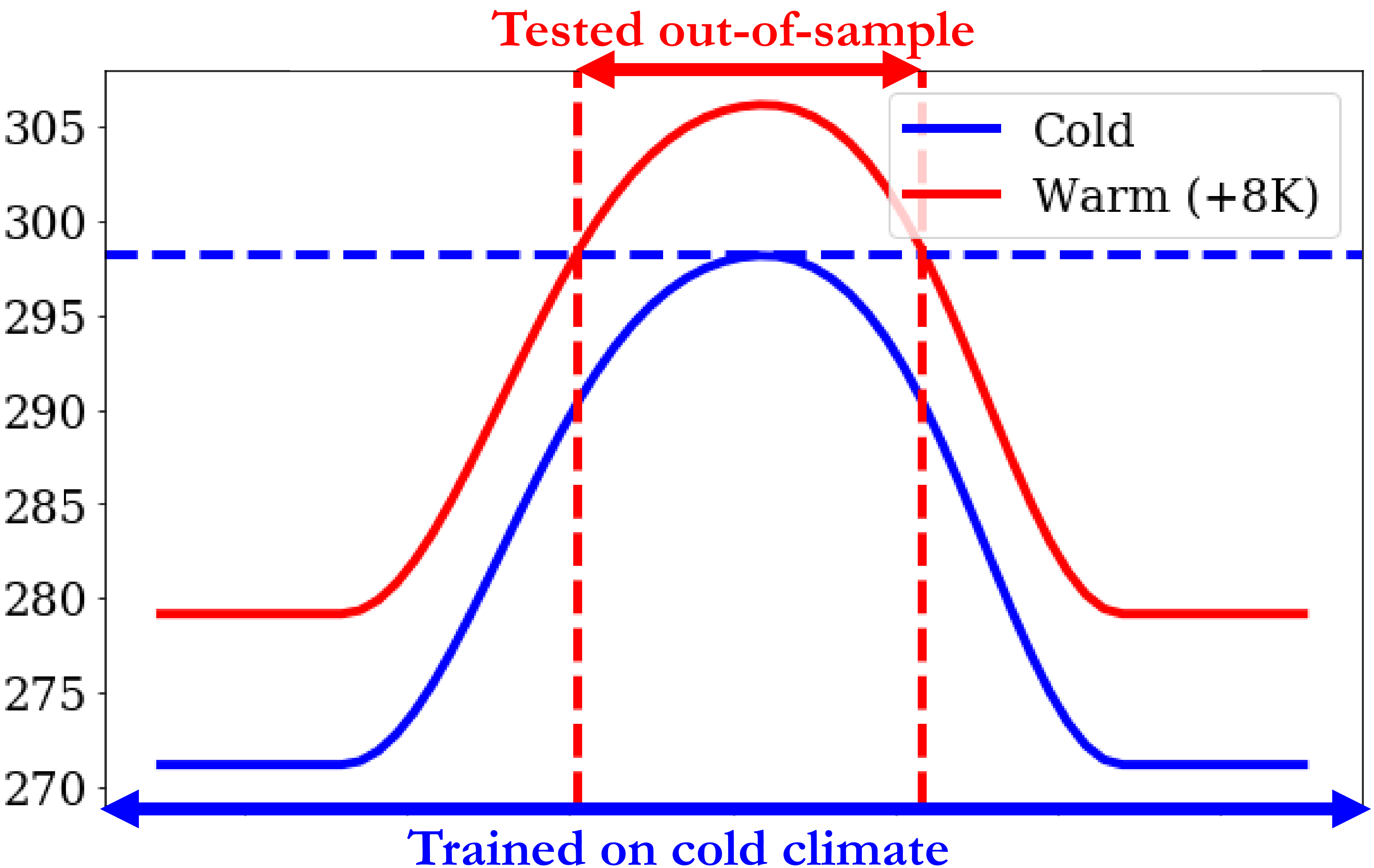
Generalization Experiment: Uniform +8K warming



Generalization Experiment: Uniform +8K warming



Generalization Experiment: Uniform +8K warming



Physically rescale the data to convert extrapolation into interpolation

Goal: Uncover climate-invariant mapping from climate to convection

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

Brute Force: Not Climate-Invariant

Physically rescale the data
to convert extrapolation into interpolation

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

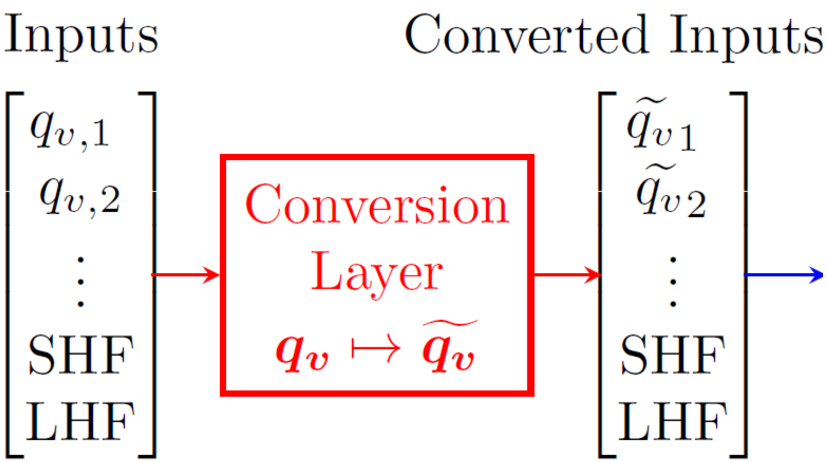
$$\begin{bmatrix} \widetilde{\text{Specific humidity}}\,(\widetilde{p}) \\ \widetilde{\text{Temperature}}\,(\widetilde{p}) \\ \widetilde{\text{Surface Pressure}} \\ \widetilde{\text{Solar Insolation}} \\ \widetilde{\text{Latent Heat Flux}} \\ \widetilde{\text{Sensible Heat Flux}} \end{bmatrix} \overset{\widetilde{\text{NN}}}{\longmapsto}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

<u>Goal</u>: Climate-Invariant

<u>Steps</u>:

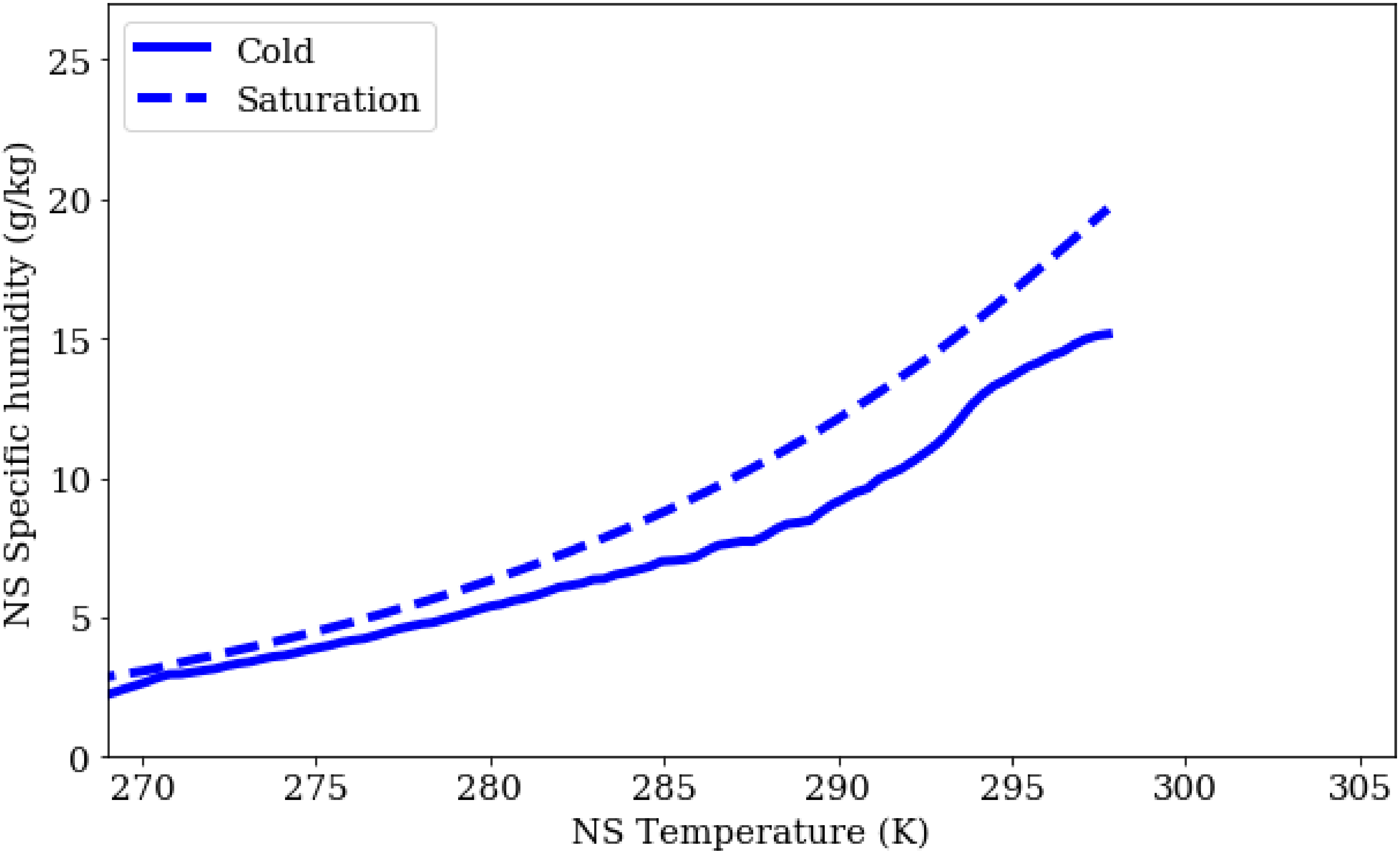
1. Data analysis to find climate-invariant mapping
2. Train NN on physically-rescaled data

Algorithms: Custom Data Generators & Custom Layers

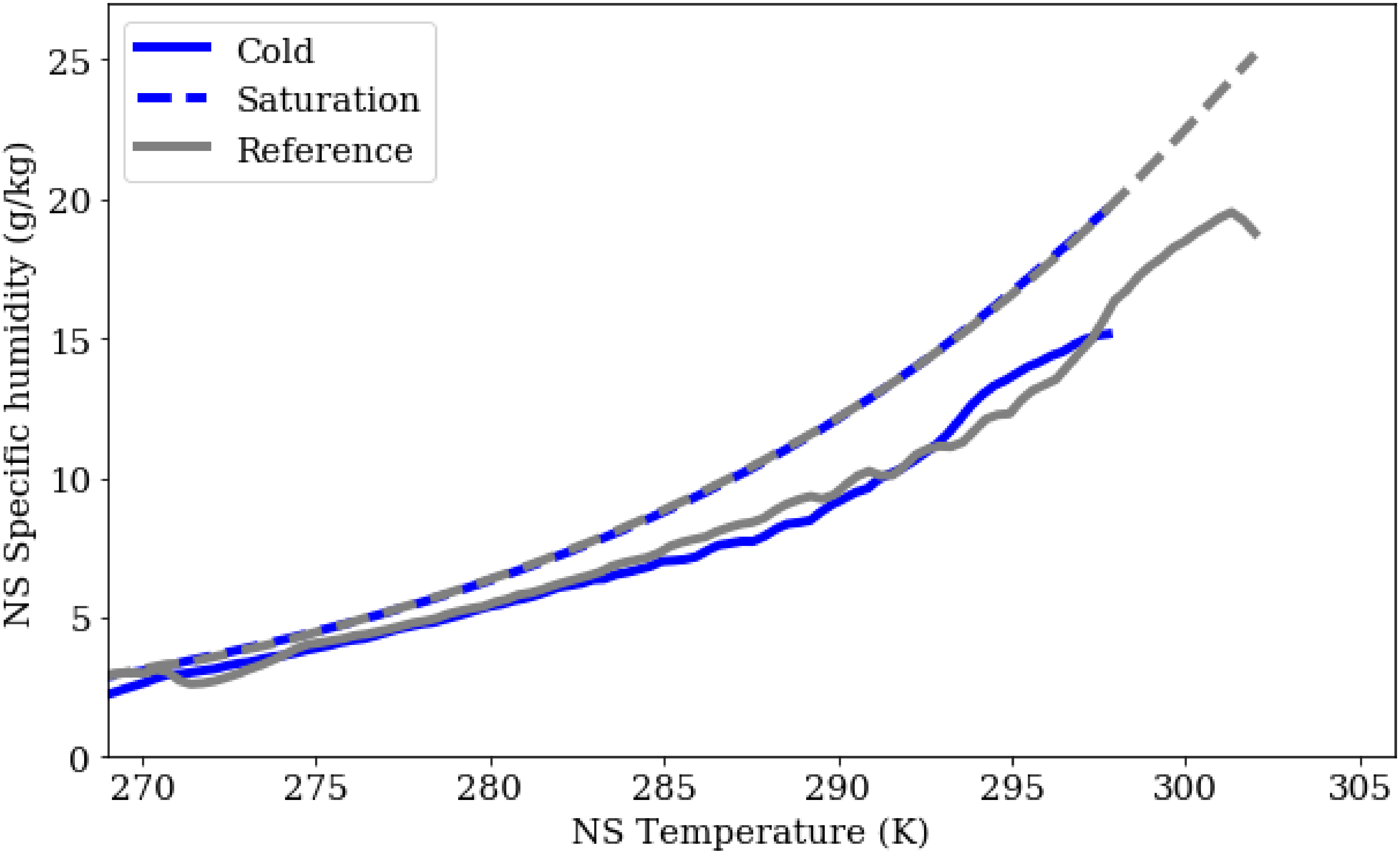


- Only one training/validation/test data despite multiple rescalings
- Build NNs using different physical rescalings (trial & error)
- Keep the rescalings that yield the best generalization

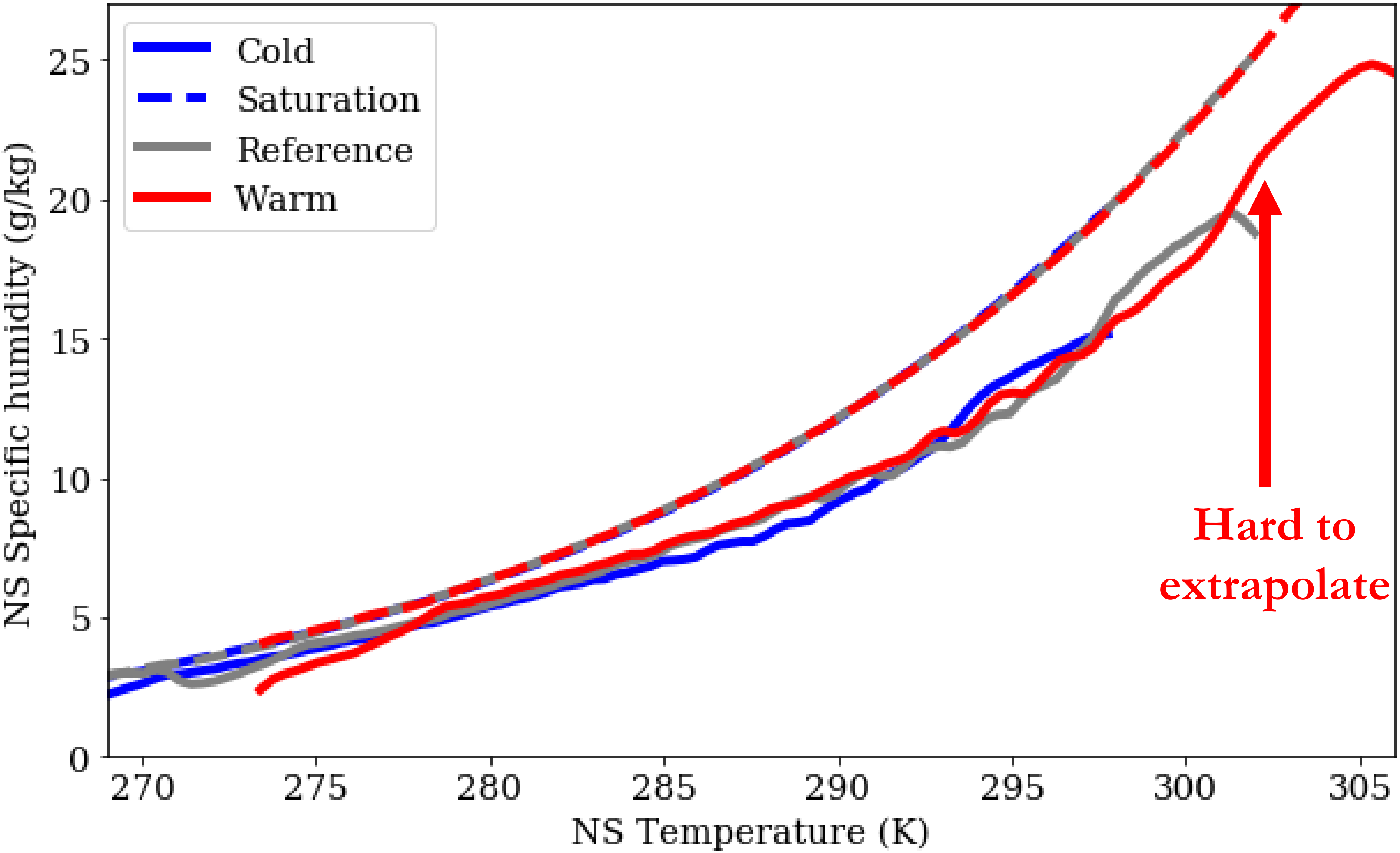
Clausius Clapeyron implies exponential scaling of moisture with temperature



Clausius Clapeyron implies exponential scaling of moisture with temperature

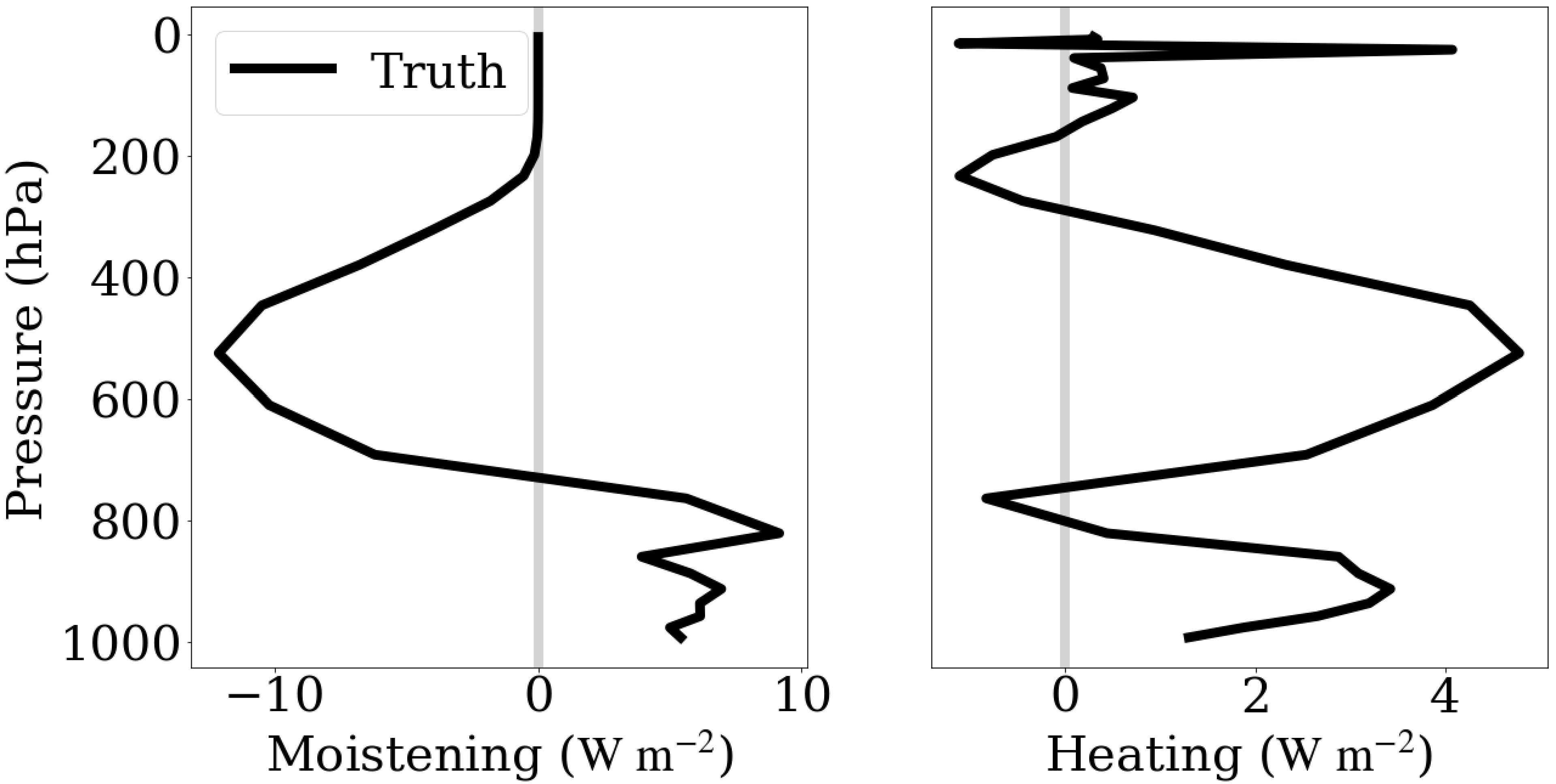


Clausius Clapeyron implies exponential scaling of moisture with temperature



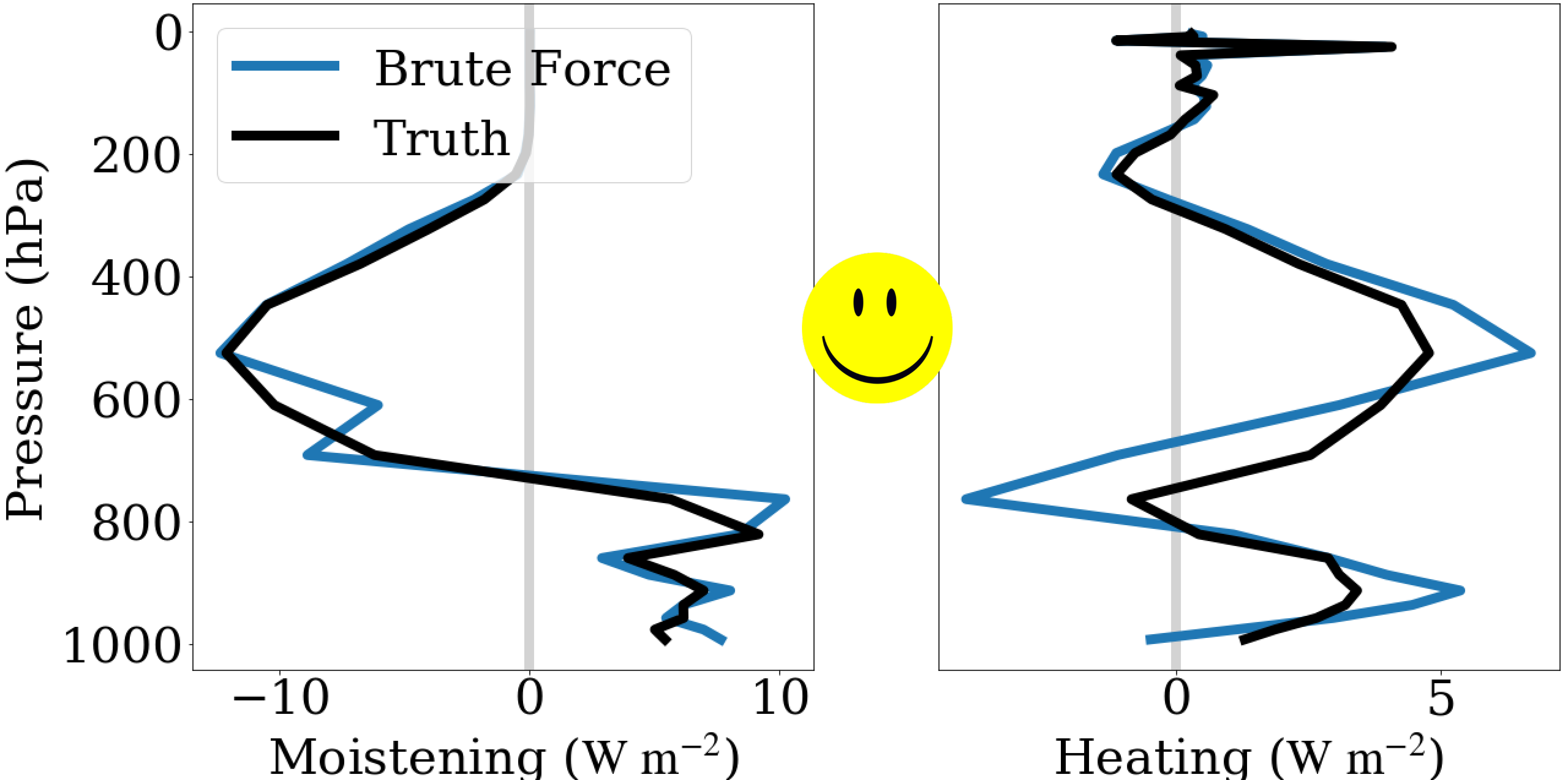
Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in cold climate



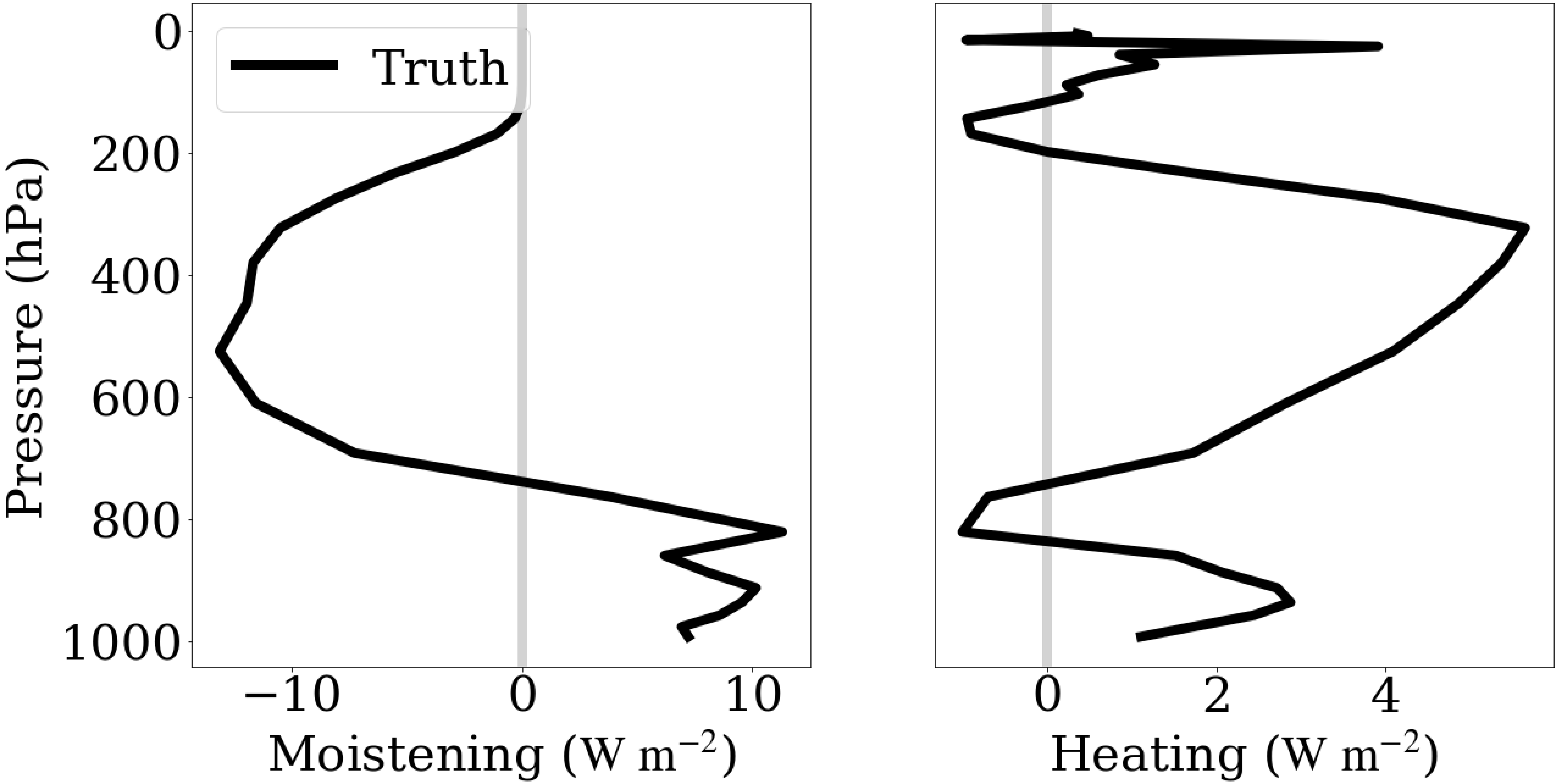
Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in cold climate



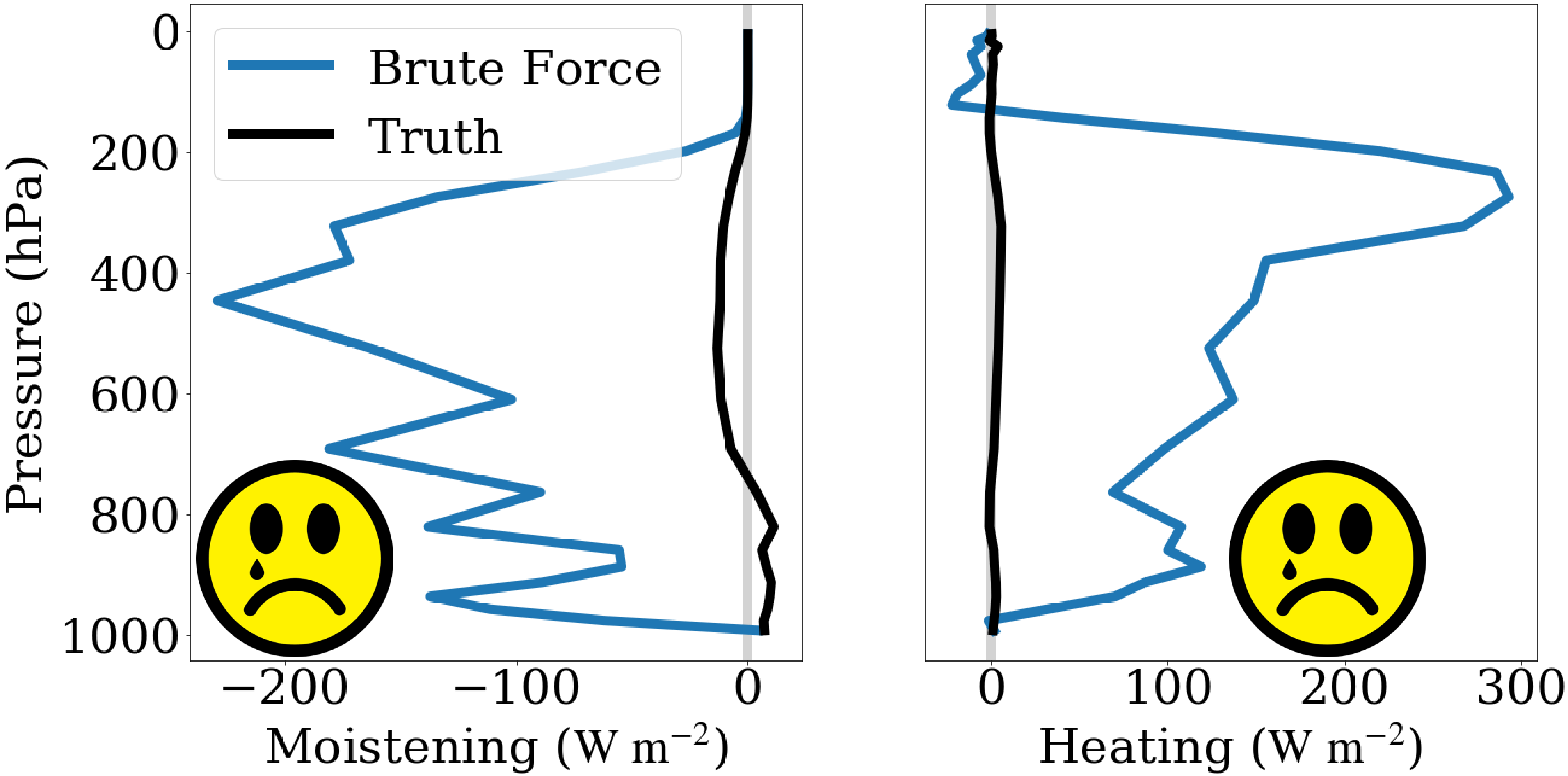
Problem: NNs fail to generalize to unseen climates

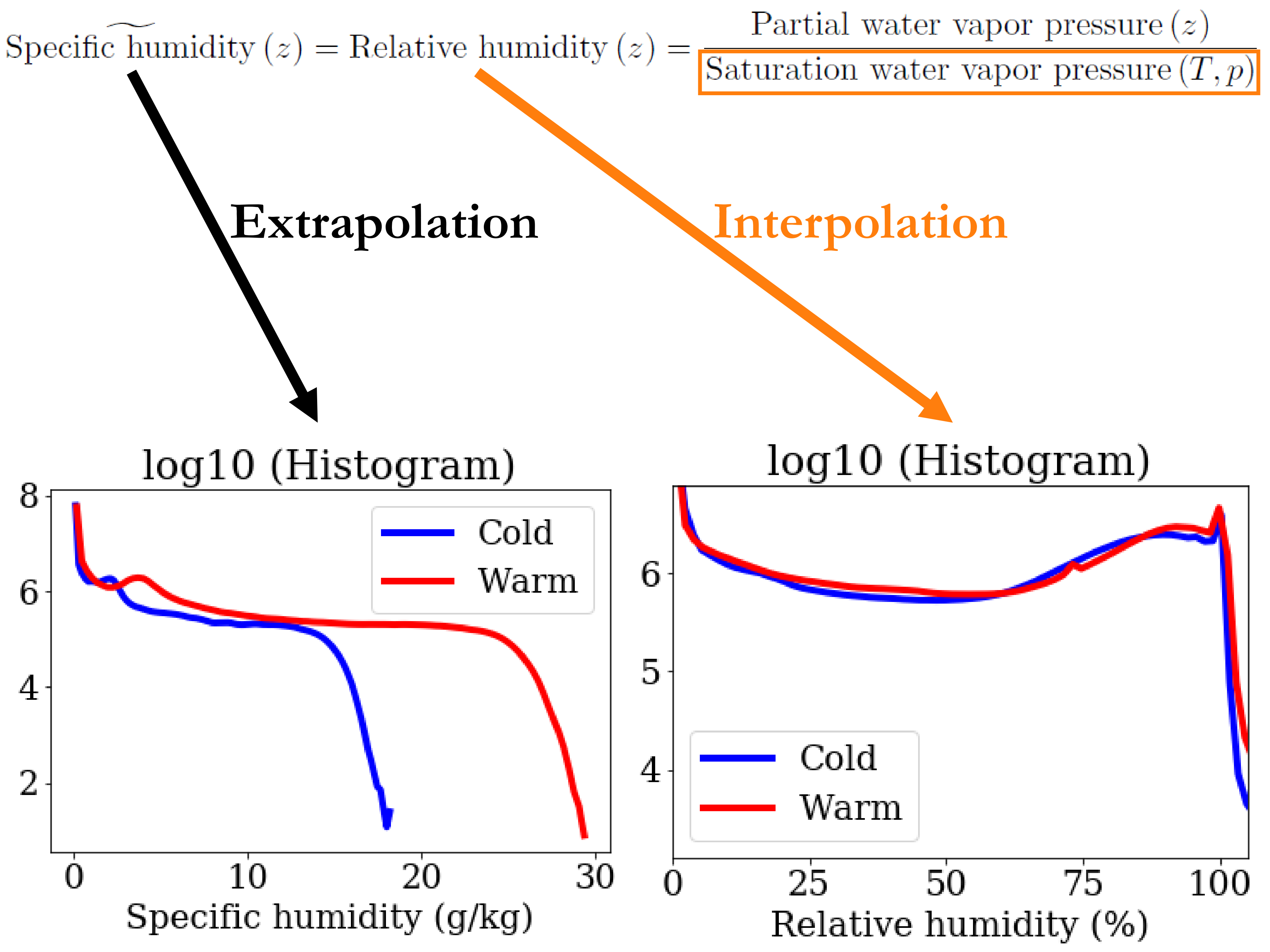
Daily-mean Tropical prediction in warm climate



Problem: NNs fail to generalize to unseen climates

Daily-mean Tropical prediction in warm climate

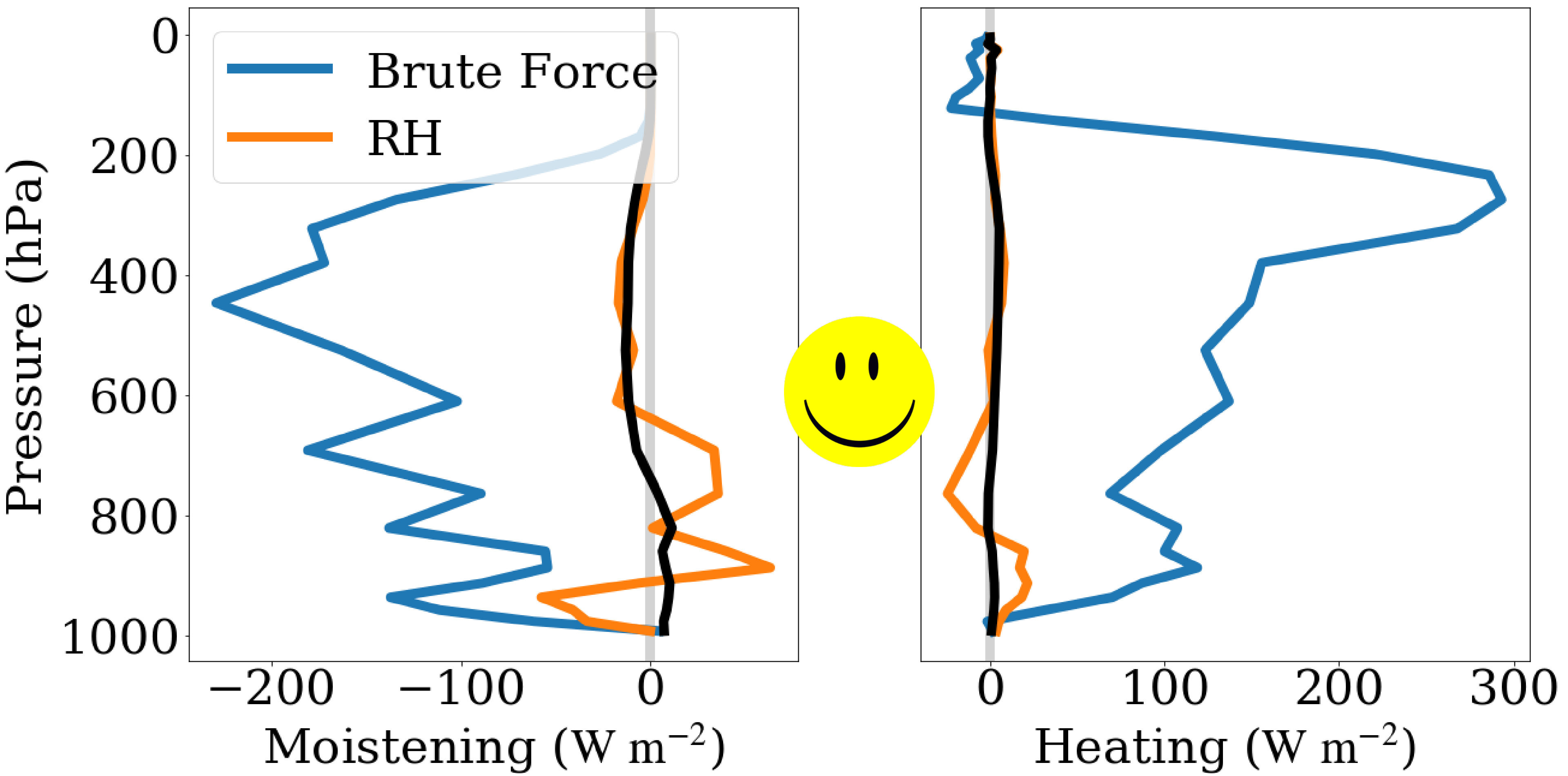




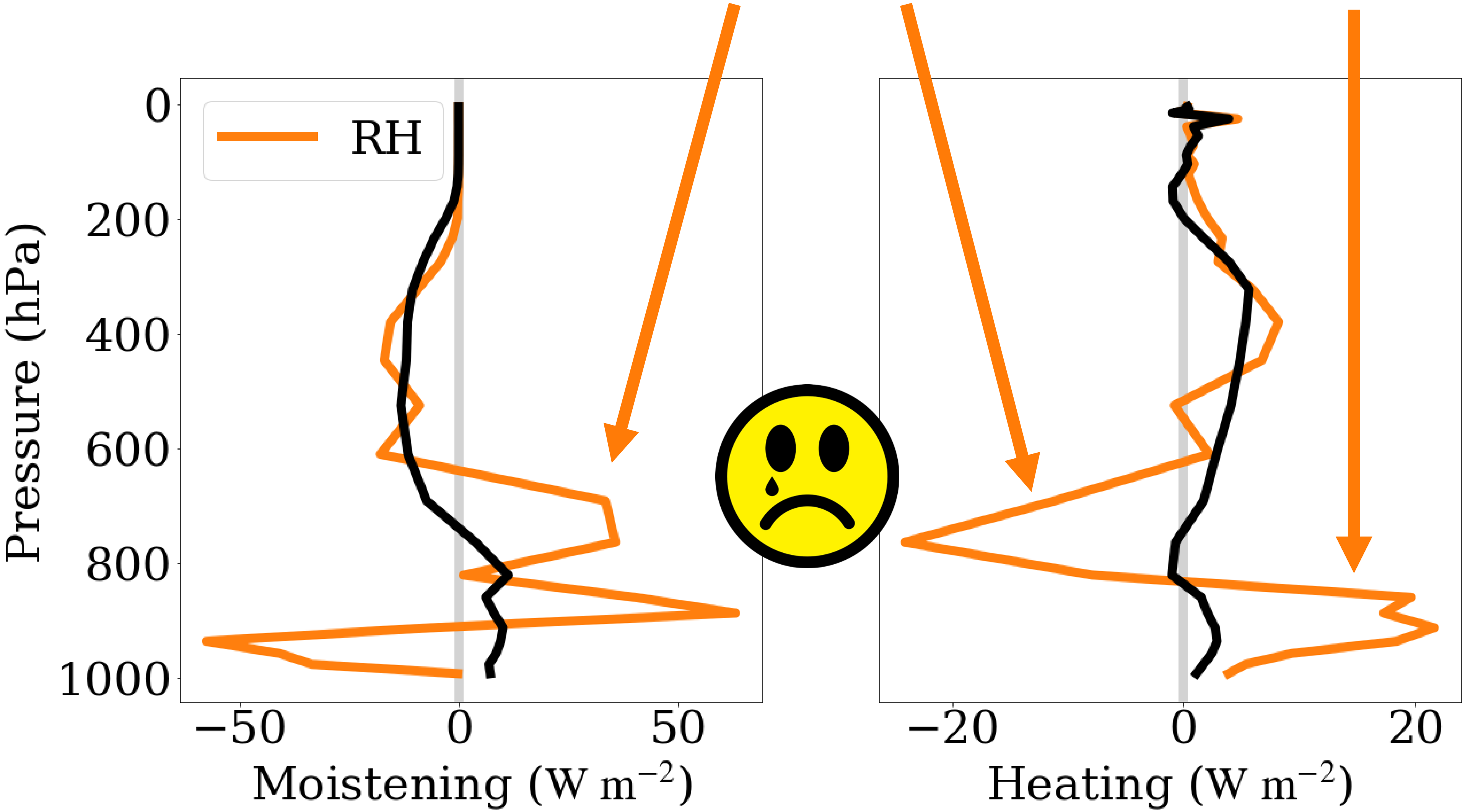
$\widetilde{\text{Specific humidity}}(z) = \text{Relative humidity}(z) = \dfrac{\text{Partial water vapor pressure}(z)}{\boxed{\text{Saturation water vapor pressure}(T,p)}}$
Extrapolation
Interpolation
log10 (Histogram)
Cold
Warm
8
6
4
2
0
10
20
30
Specific humidity (g/kg)
log10 (Histogram)
Cold
Warm
6
5
4
0
25
50
75
100
Relative humidity (%)

$$\widetilde{\text{Specific humidity}}\,(z) = \text{Relative humidity}\,(z) = \frac{\text{Partial water vapor pressure}\,(z)}{\boxed{\text{Saturation water vapor pressure}\,(T, p)}}$$

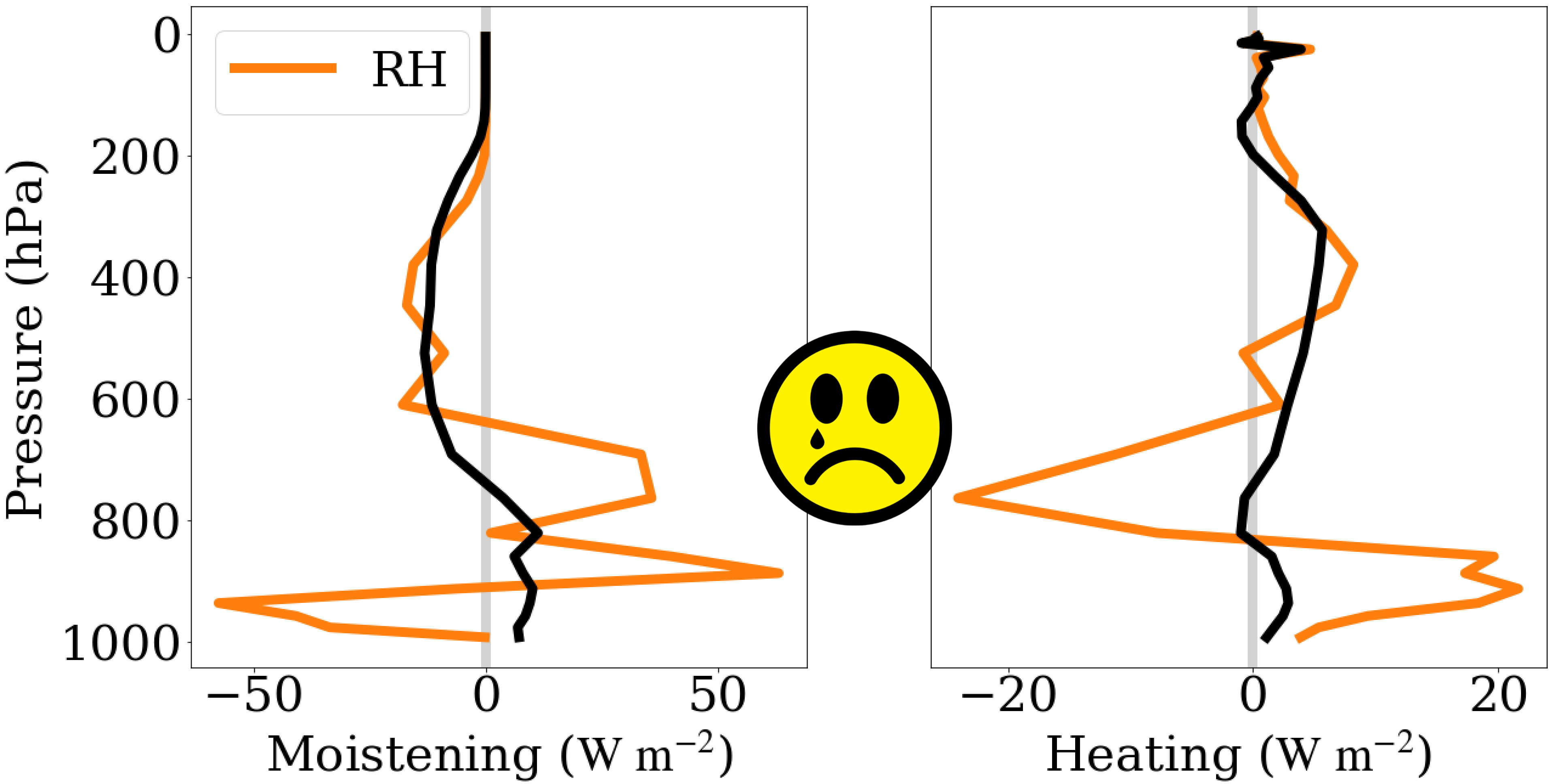
Generalization improves dramatically!

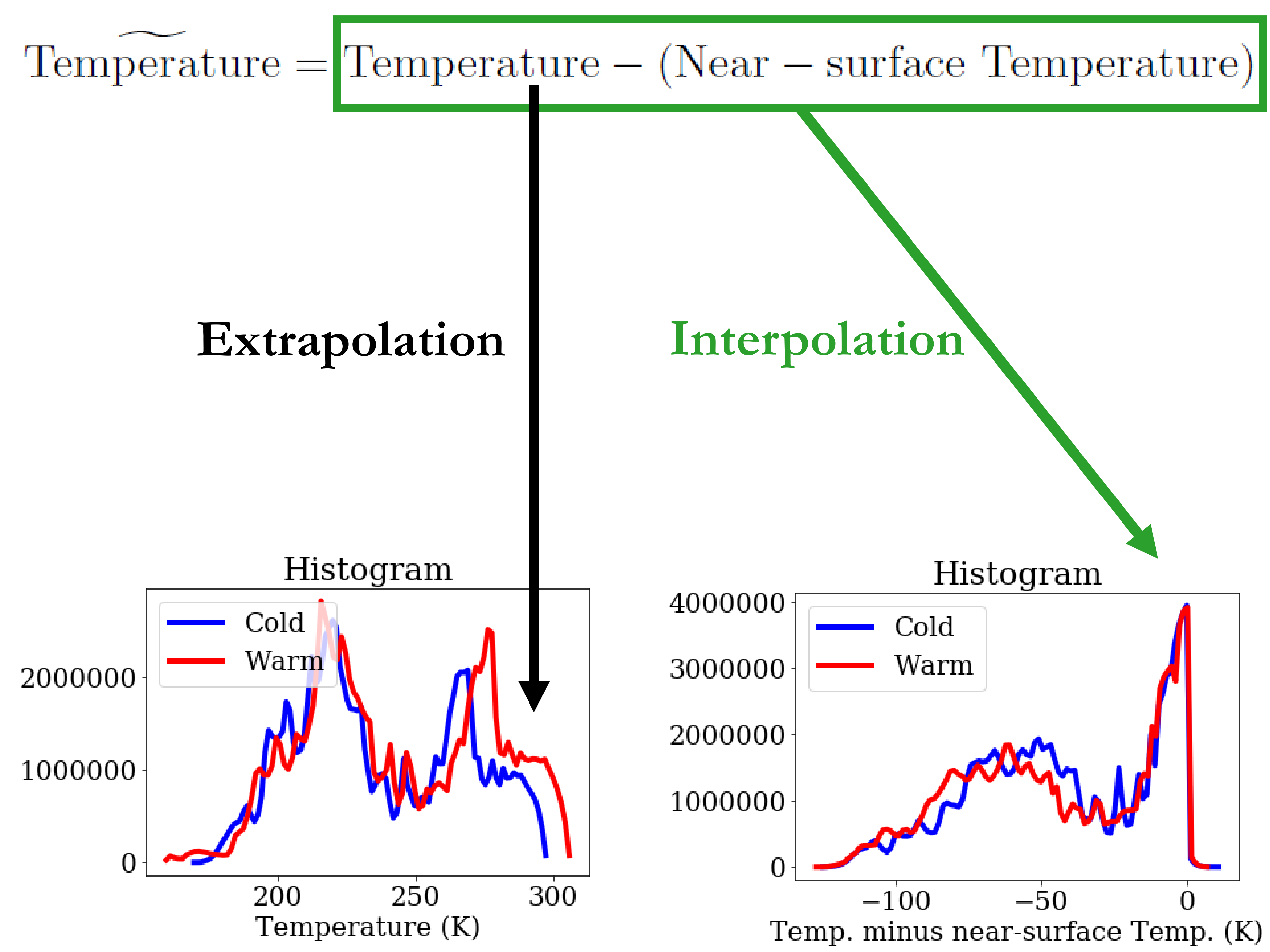


Temperature out-of-sample in lower atmosphere



Can we similarly normalize Temperature?

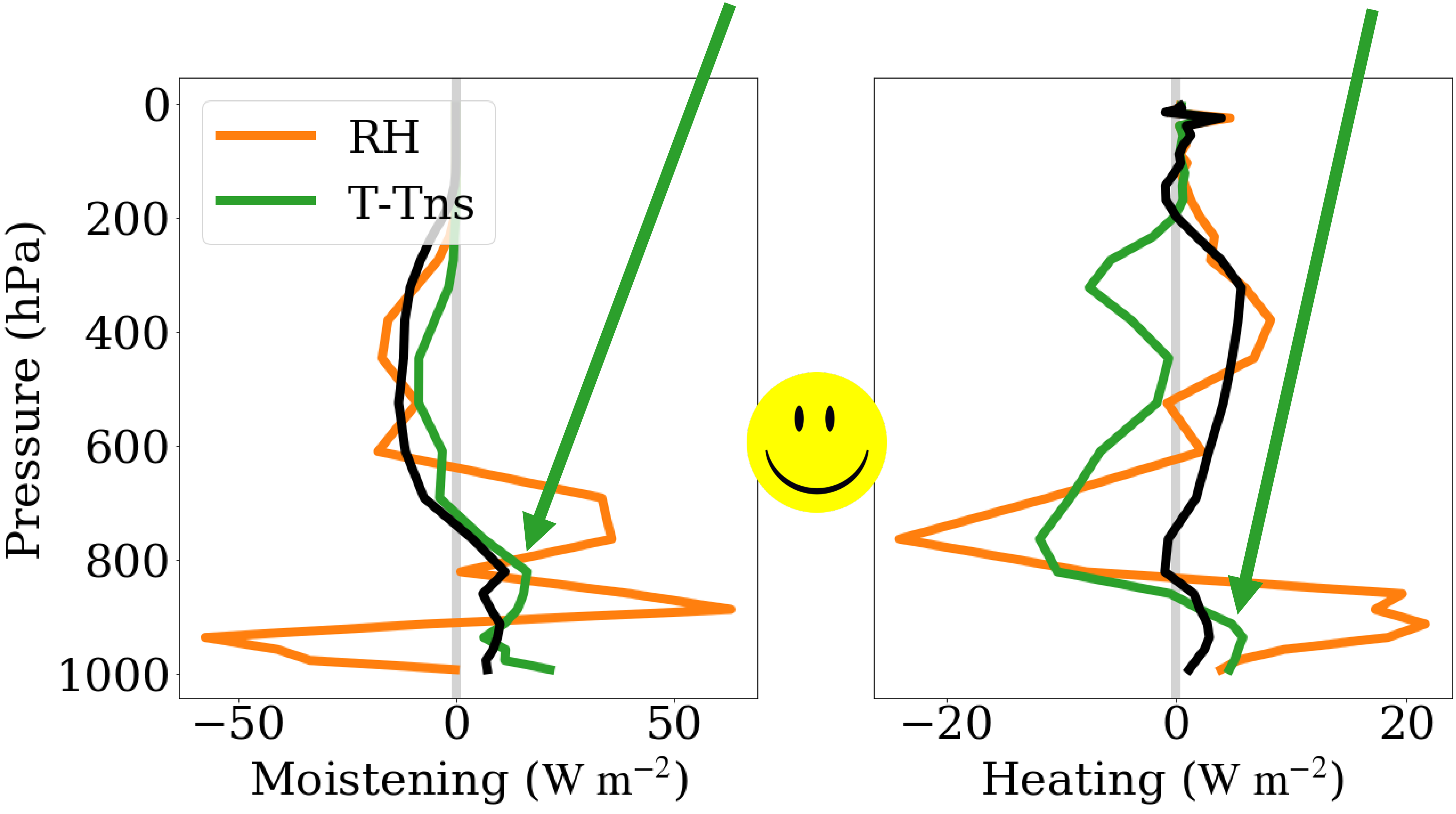




Temperature = Temperature − (Near − surface Temperature)
Extrapolation
Interpolation
Histogram
Cold
Warm
2000000
1000000
0
200
250
300
Temperature (K)
Histogram
4000000
3000000
2000000
1000000
0
Cold
Warm
−100
−50
0
Temp. minus near-surface Temp. (K)

$$\widetilde{\text{Temperature}} = \boxed{\text{Temperature} - (\text{Near} - \text{surface Temperature})}$$

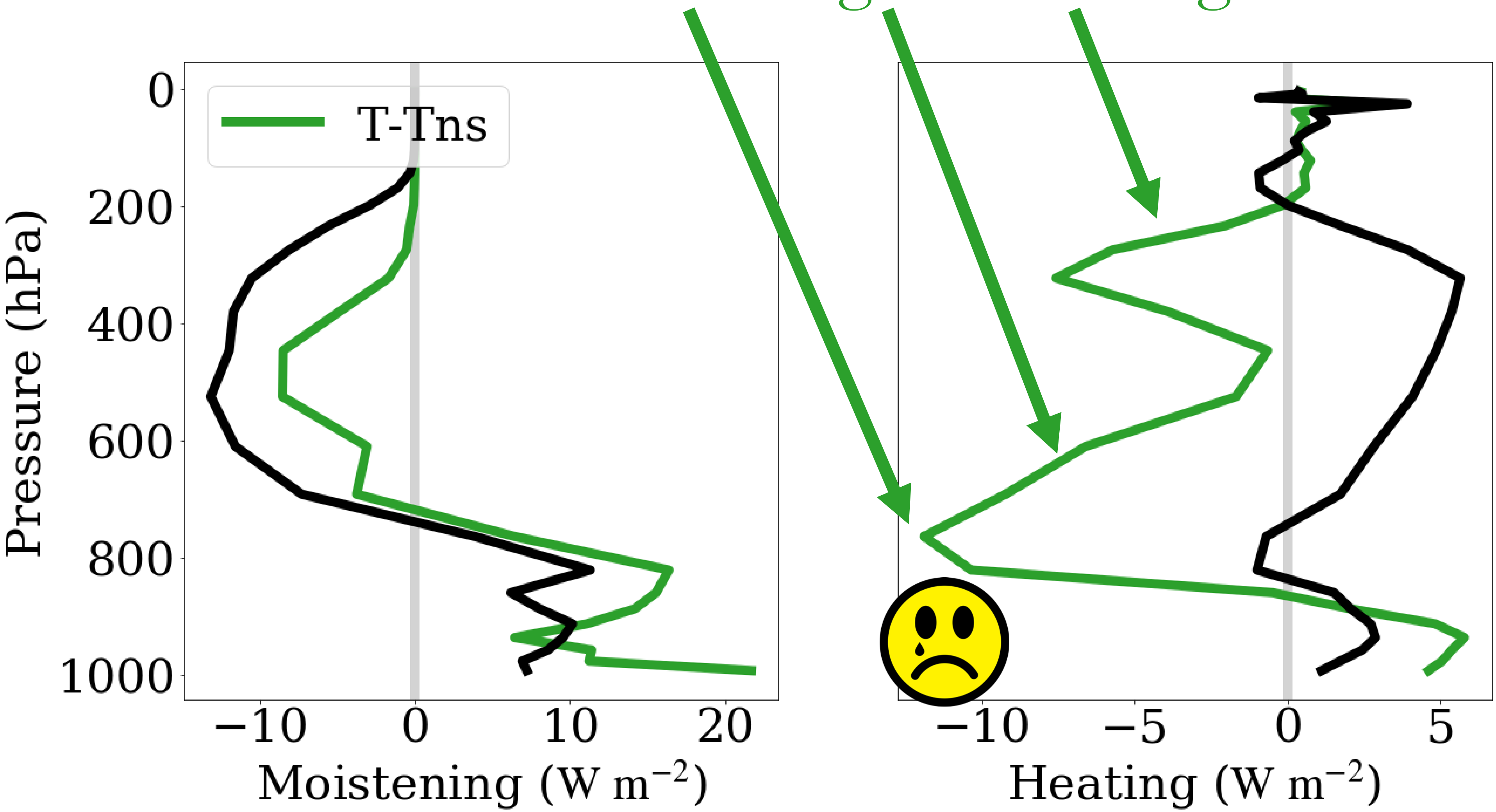
Generalization improved in lower atmosphere



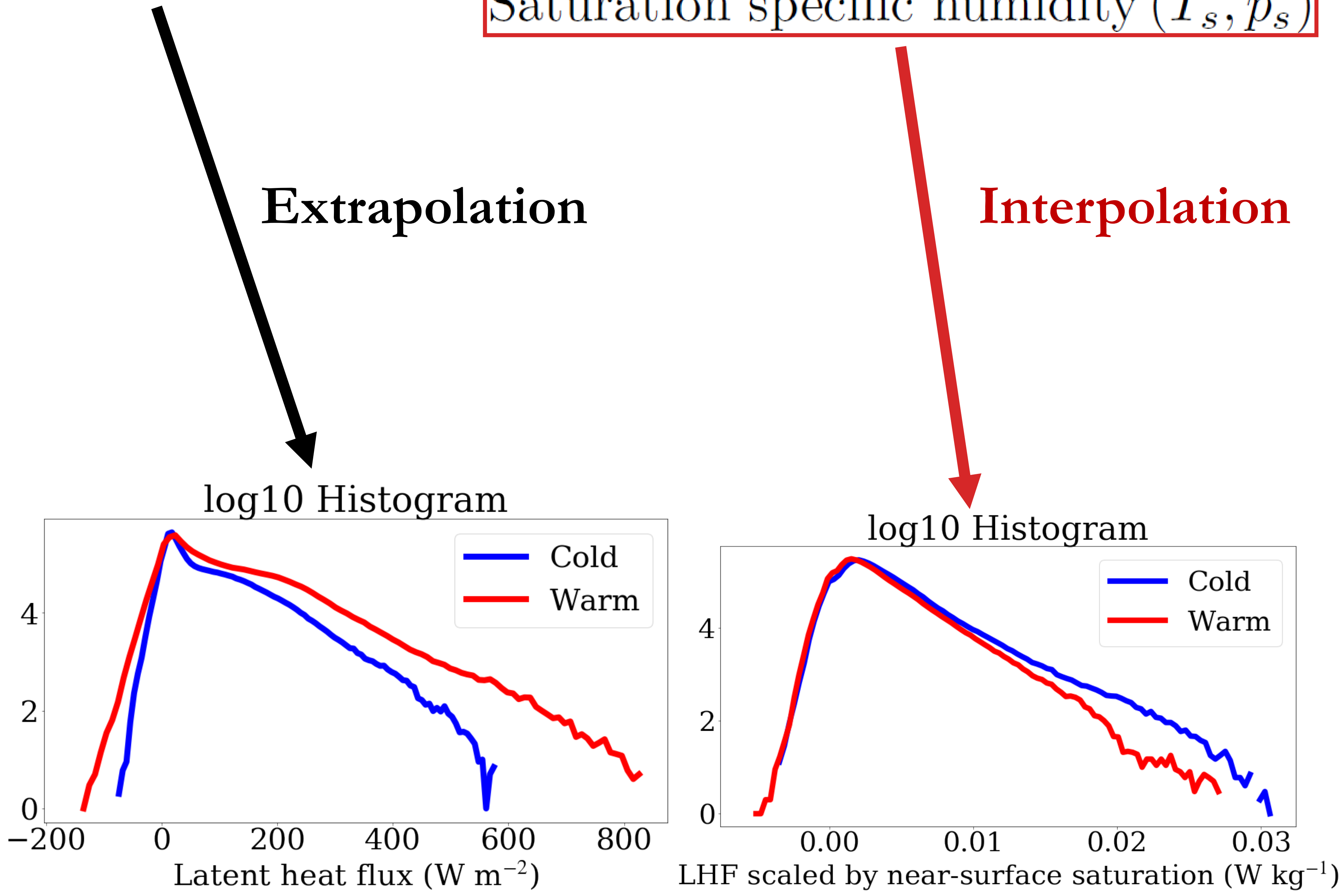
$$\widetilde{\text{Temperature}} = \boxed{\text{Temperature} - (\text{Near} - \text{surface Temperature})}$$

BUT

Problem with the sign of heating

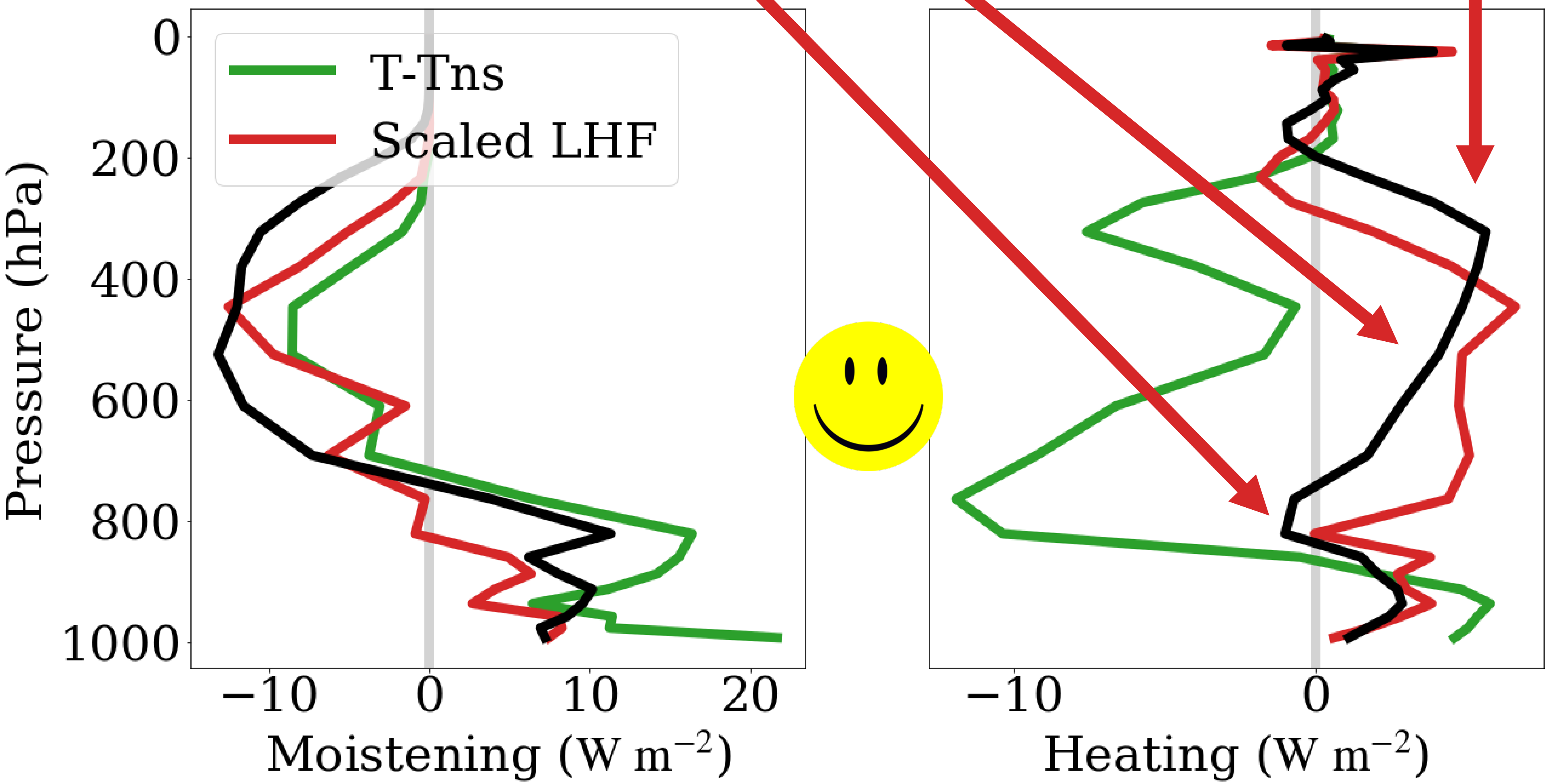


$$\widetilde{\text{Latent Heat Flux}} = \frac{\text{Latent Heat Flux}}{\boxed{\text{Saturation specific humidity}\,(T_s, p_s)}}$$



$$\widetilde{\text{Latent Heat Flux}} = \frac{\text{Latent Heat Flux}}{\boxed{\text{Saturation specific humidity}\,(T_s, p_s)}}$$

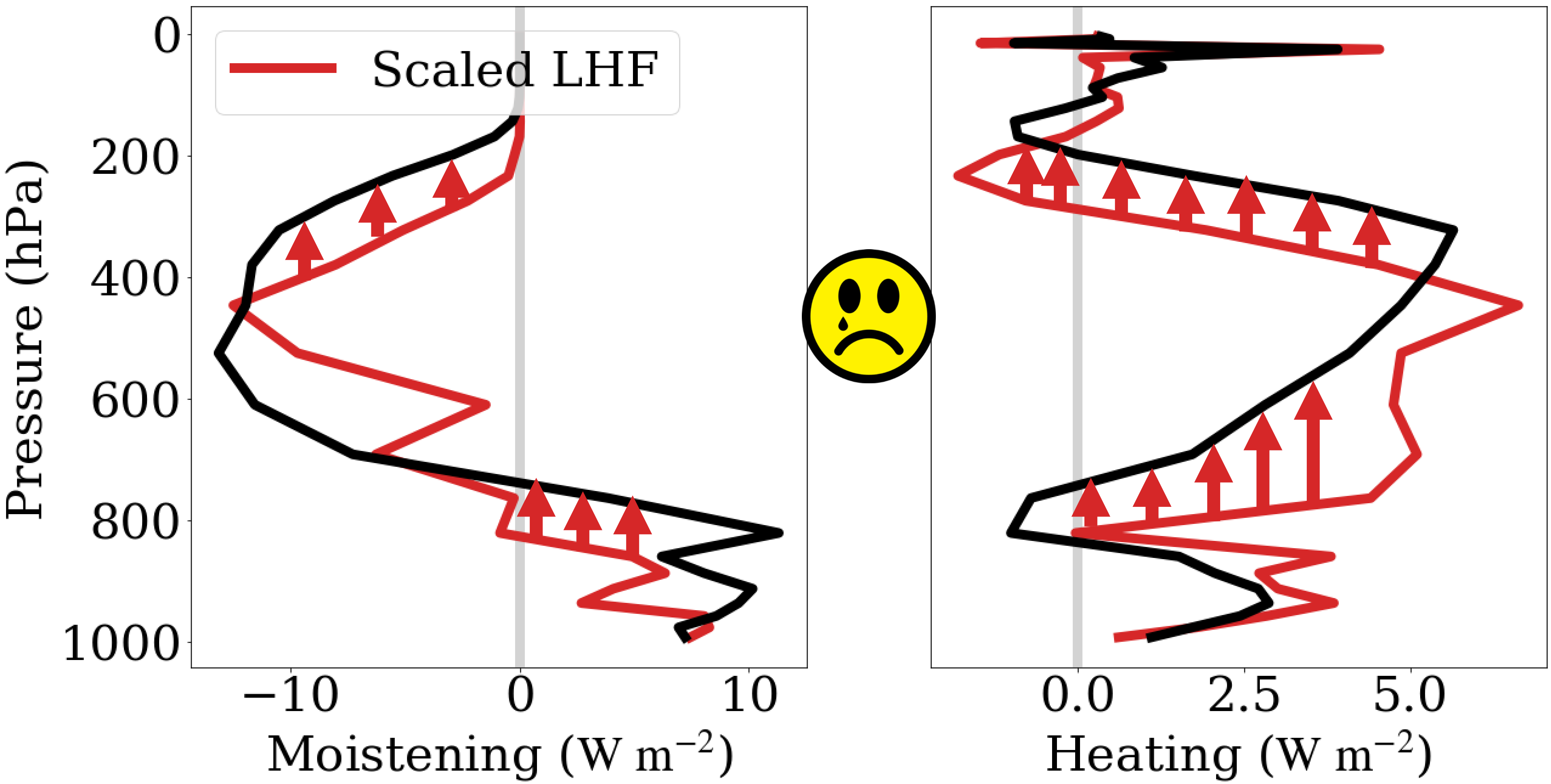
Sign of convective heating mostly corrected!



$$\widetilde{\text{Latent Heat Flux}} = \frac{\text{Latent Heat Flux}}{\boxed{\text{Saturation specific humidity}\,(T_s, p_s)}}$$

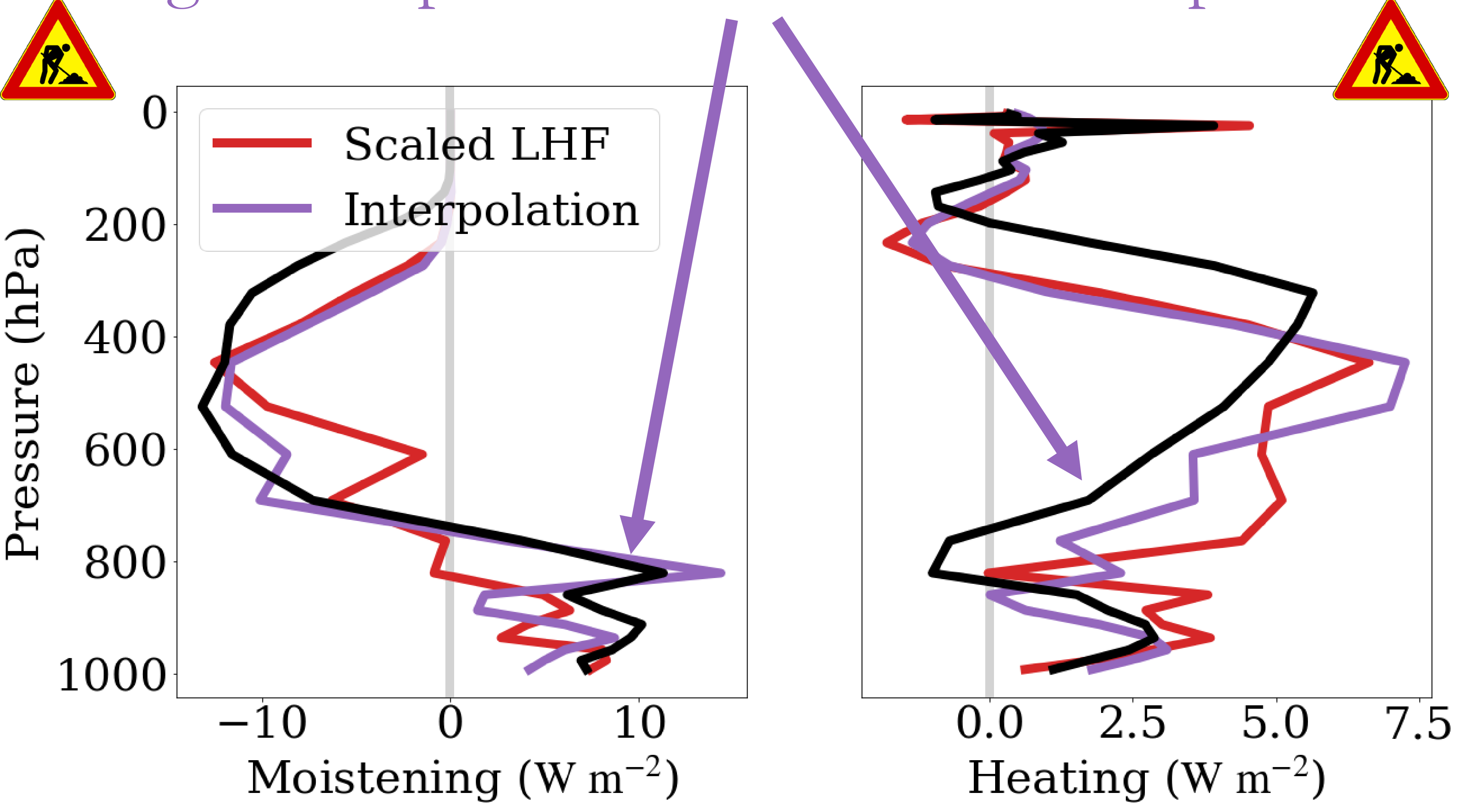
BUT

NN does not capture upwards shift with warming



$$\widetilde{\text{Pressure}} = \frac{(\text{Surface Pressure}) - \text{Pressure}}{(\text{Surface Pressure}) - (\text{Tropopause Pressure})}$$

Sign of improvements in lower atmosphere



1
Interpretability
Stability

We can make data-driven parametrizations **climate-invariant** by rescaling the data so as to transform an extrapolation into interpolation

2
Physical
Constraints

See: Beucler et al. (2020) "Towards Physically-consistent, Data-driven Models of Convection." arXiv:2002.08525

3
Generalization

1
Interpretability
Stability

Given Data-driven Parametrization,
How to make it **Interpretable**
& **Physically Consistent**?

2
Physical
Constraints

3
Generalization

1
Interpretability
Stability

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)

2
Physical
Constraints

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

3
Generalization

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!

1
Interpretability
Stability

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)

2
Physical
Constraints

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

3
Generalization