Embedding machine-learnt sub-grid variability improves climate model precipitation patterns

<u>Daniel Giles [1]</u>, James Briant [2], Cyril Morcrette [3] & Serge Guillas [2]

1. Department of Computer Science, University College London, UK
2. Department of Statistical Science, University College London, UK
3. Met Office, Exeter, UK

Motivation

Improve the representation of clouds/precipitation in climate simulations

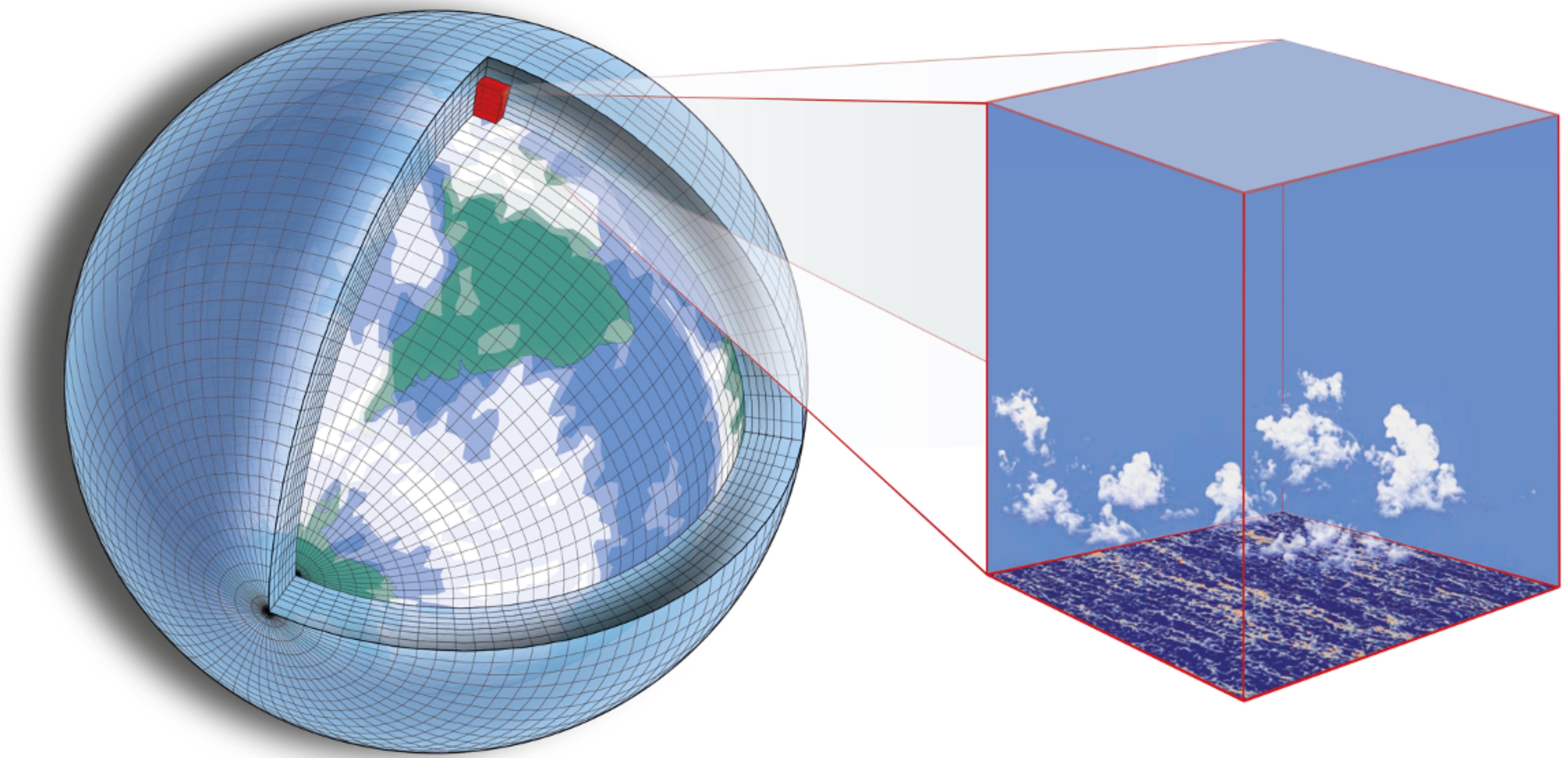
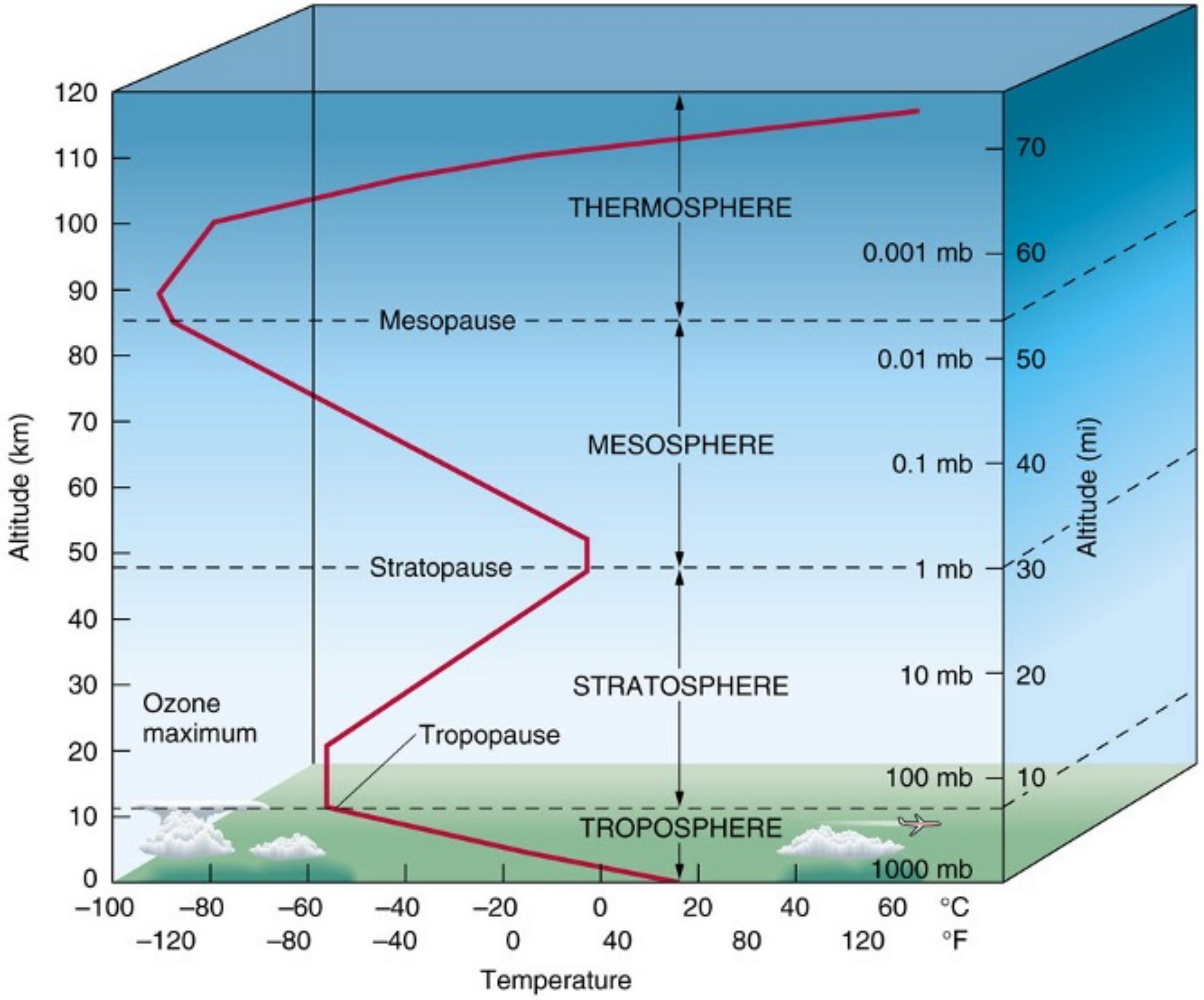


Image credit: https://climate.nasa.gov/internal_resources/1974/

Approach

Coupling Statistical ML with a Climate Model

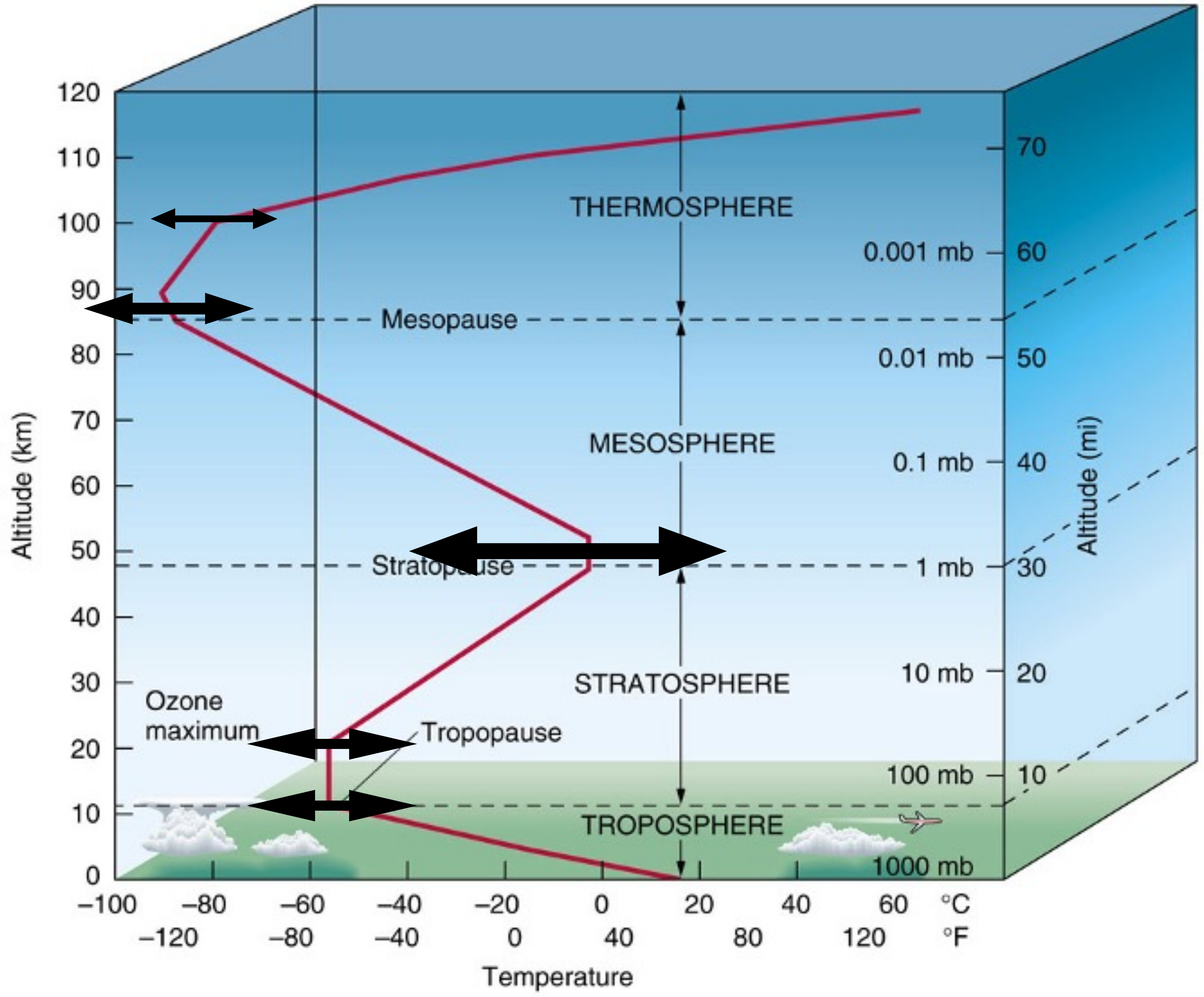
- Use a statistical learned model to predict local **variability** on mean cell values - Temperature (T) and Specific Humidity (Q)



Approach

Coupling Statistical ML with a Climate Model

- Use a statistical learned model to predict local **variability** on mean cell values - Temperature (T) and Specific Humidity (Q)



+/- σ

Approach

- Use a statistical learned model to predict local **variability** on mean cell values

Inputs $\left[p_s, T_1, T_2, \ldots, T_n, Q_1, Q_2, \ldots, Q_n, \mu_{oro}, \sigma_{oro}, lsm\right]$

Approach

- Use a statistical learned model to predict local **variability** on mean cell values

Inputs $\left[p_s, T_1, T_2, \ldots, T_n, Q_1, Q_2, \ldots, Q_n, \mu_{oro}, \sigma_{oro}, lsm\right]$

Statistical Model

Approach

- Use a statistical learned model to predict local **variability** on mean cell values

Inputs $\left[p_s, T_1, T_2, \ldots, T_n, Q_1, Q_2, \ldots, Q_n, \mu_{oro}, \sigma_{oro}, lsm\right]$

$\downarrow$

Statistical Model

$\downarrow$

Outputs $\left[\sigma_{T_1}, \sigma_{T_2}, \ldots, \sigma_{T_n}, \sigma_{Q_1}, \sigma_{Q_2}, \ldots, \sigma_{Q_n}\right]$

Workflow

1. Generate the high resolution training data - Unified Model (Met Office)

Workflow

1. Generate the high resolution training data - Unified Model (Met Office)
2. Train a statistical model to predict σ_T and σ_Q at each layer of the atmosphere

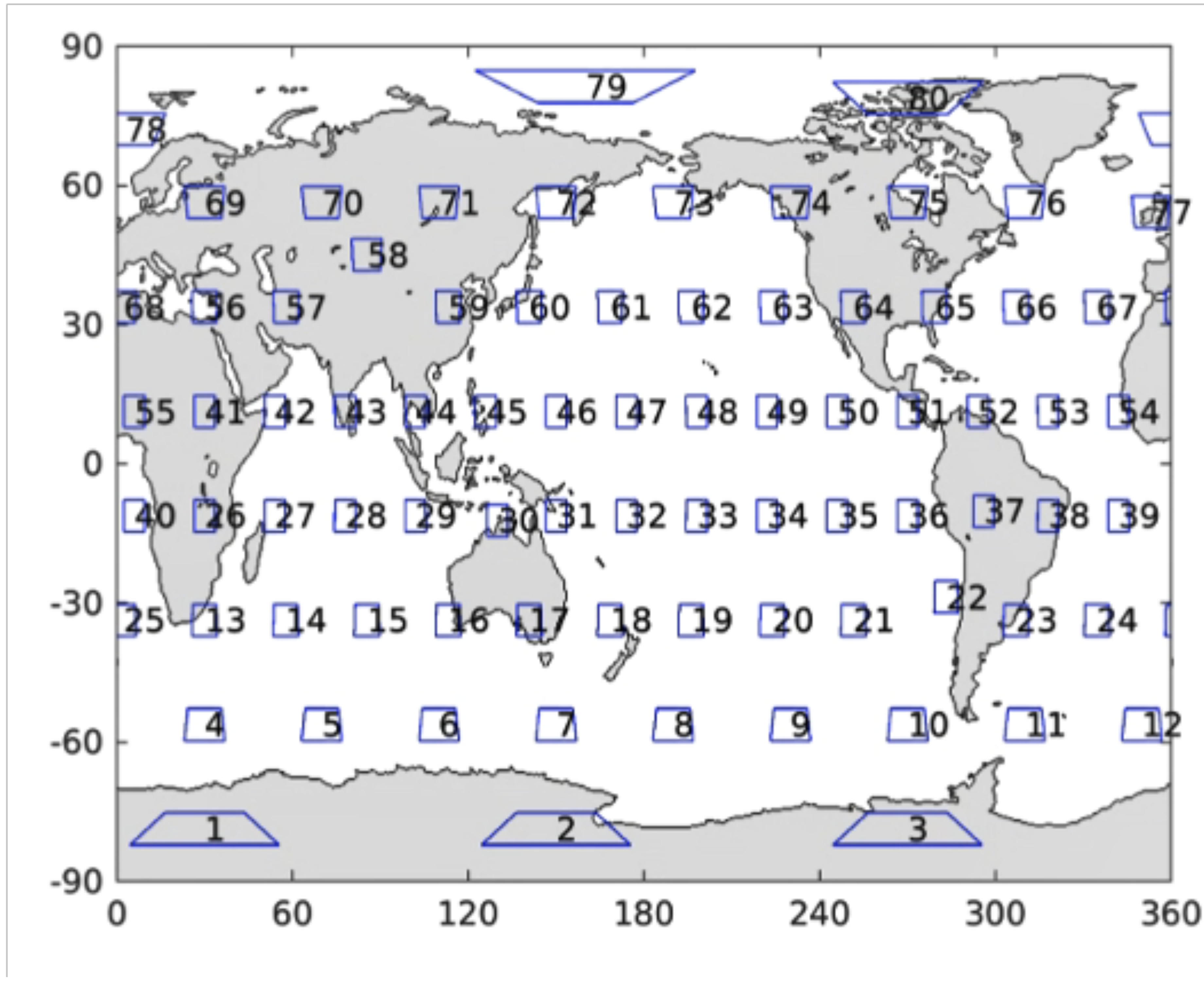
Workflow

1. Generate the high resolution training data - Unified Model (Met Office)
2. Train a statistical model to predict σ_T and σ_Q at each layer of the atmosphere
3. Embed the trained model into a coarse resolution climate model

Training Data

UK Met Office Unified Model (MetUM)

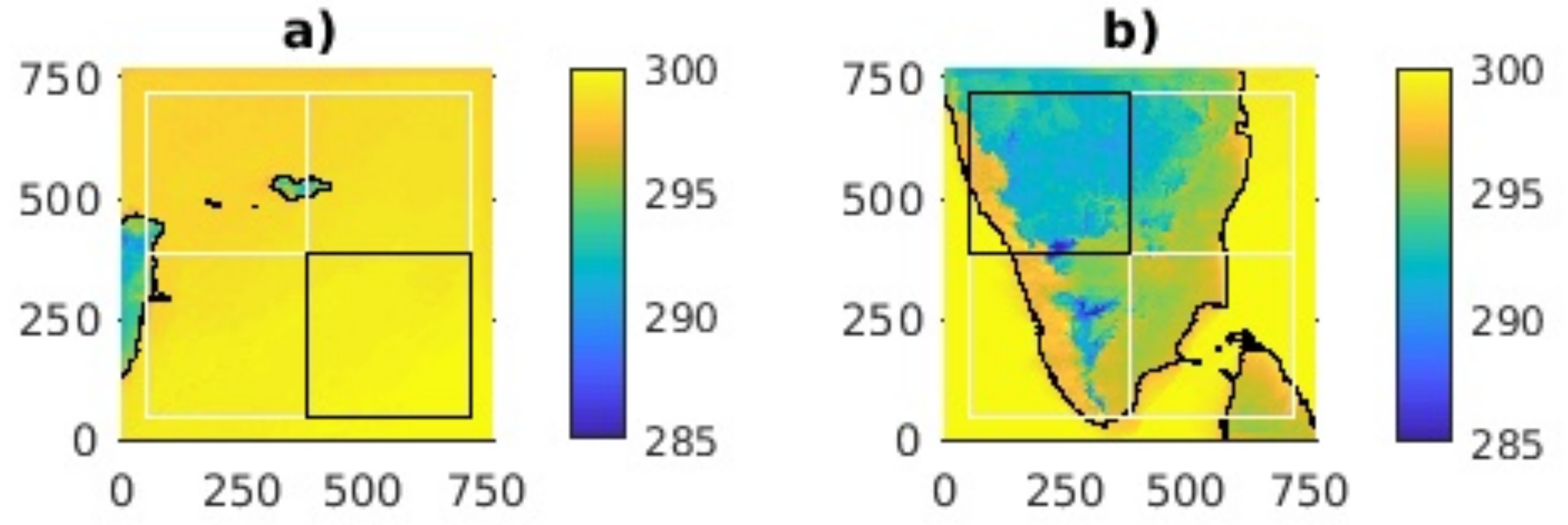
- Unified Model high-resolution simulations, 80 uniformly-spaced Limited-Area Models (LAMs) spread across the globe
- Nested within a global model hindcast (GA6 configuration)
- Each LAM is 512 x 512 grid-points wide and uses a horizontal grid length of 1.5 km
- This combination of LAM and global configuration has been used to study a range of atmospheric processes on the kilometer-scale.

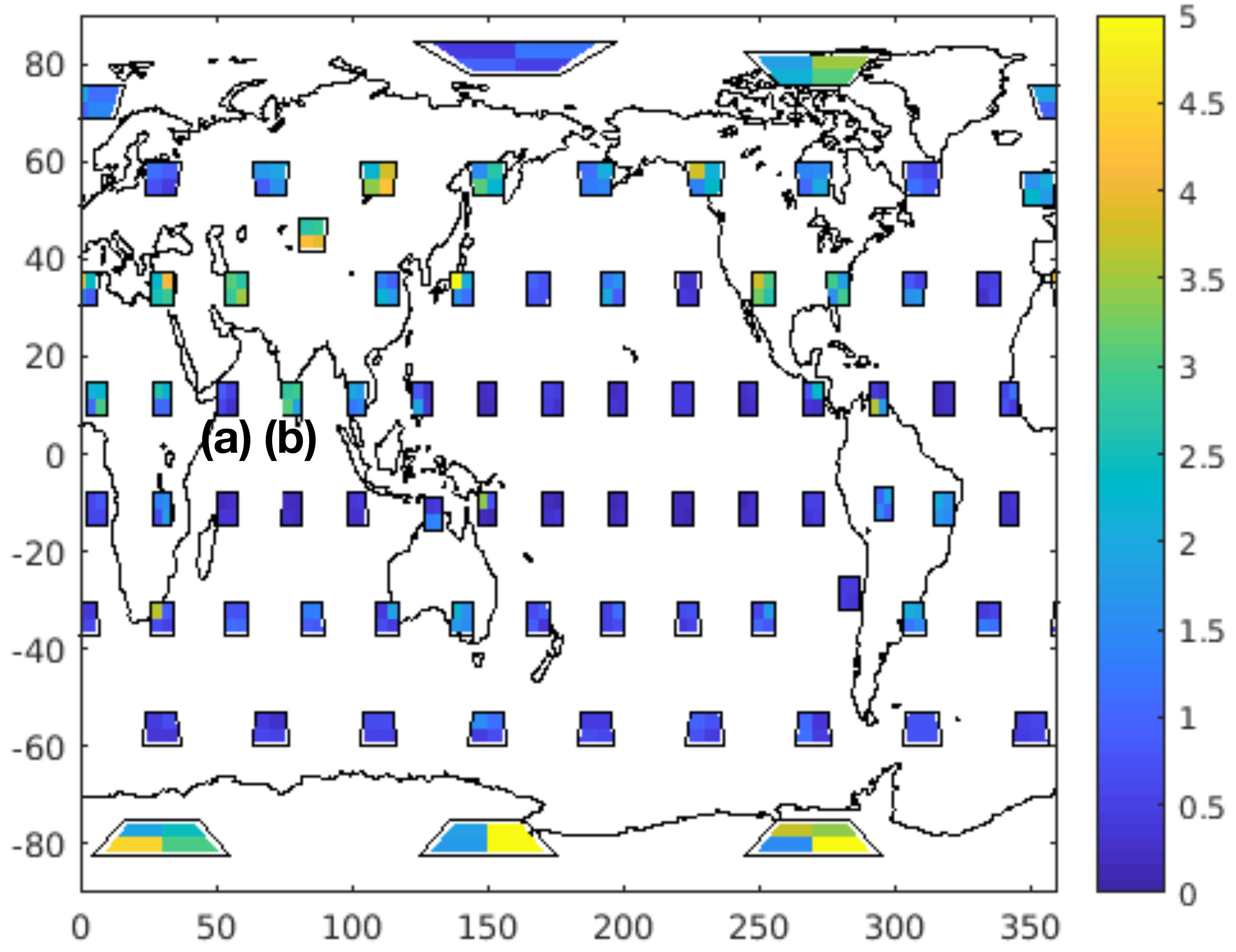


Data Coarse Graining

- Split each LAM into coarse patches (2x2) calculate the average and standard deviation over each patch.

- Results in a dataset of 12,800 profiles





The standard deviation of near-surface temperature [K] at 00Z on 1 Jan 2020

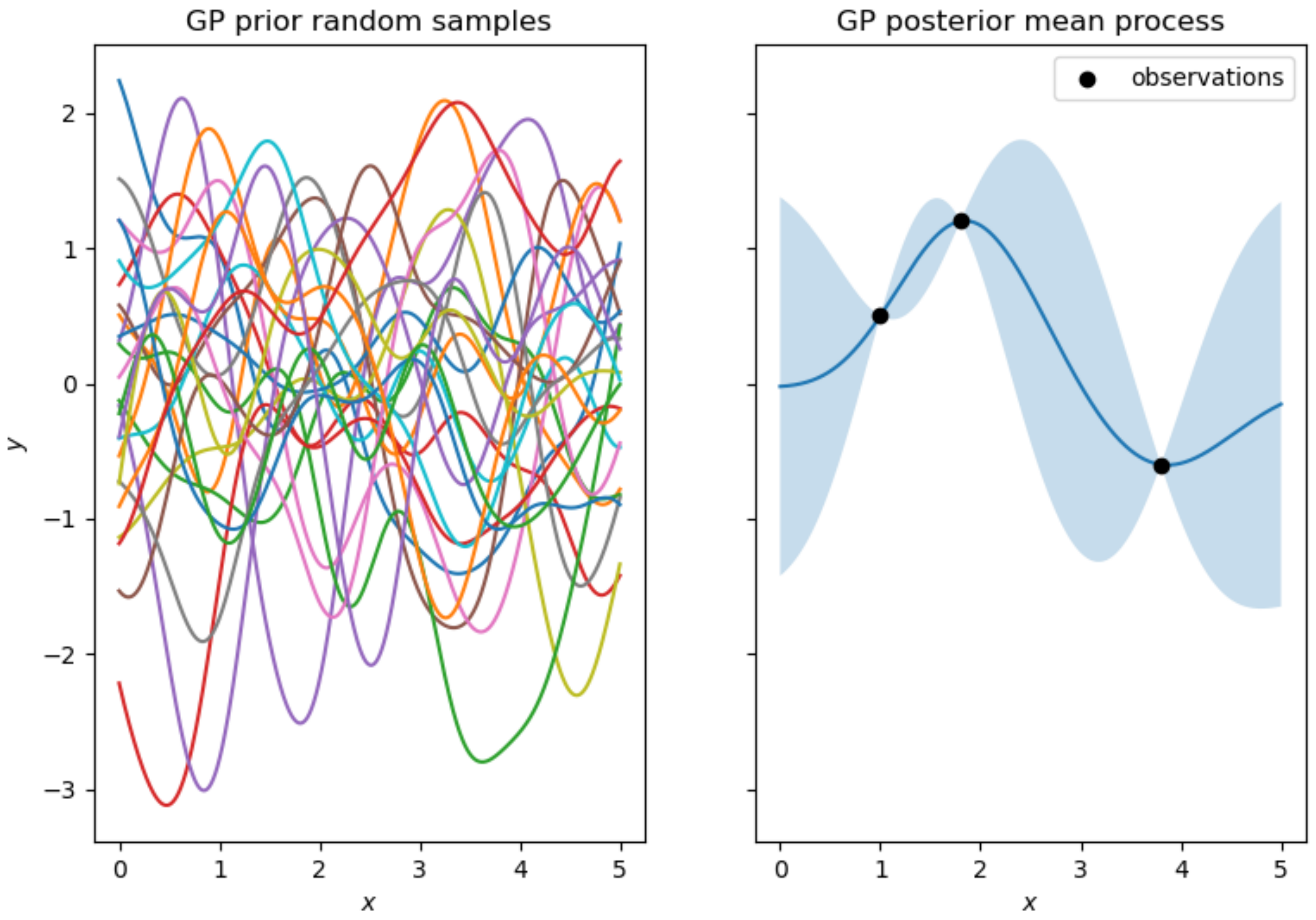
Workflow

1. Generate the high resolution training data - Unified Model (Met Office)
2. Train a statistical model to predict σ_T and σ_Q at each layer of the atmosphere

Statistical Model

Gaussian Process

- In simple terms, a Gaussian process can be thought of as a **distribution over functions.**
- Instead of having a single function that maps inputs to outputs, **a GP defines a probability distribution over the possible functions** that fit the data.
- Key ingredients:
  - Mean function
  - Covariance function (encodes assumptions about smoothness and periodicity of the data)
- A key advantage of GPs is that they provide not only predictions but also **uncertainty estimates**.



Example of a single output GP

MOGP-Emulator[1]

Key Features

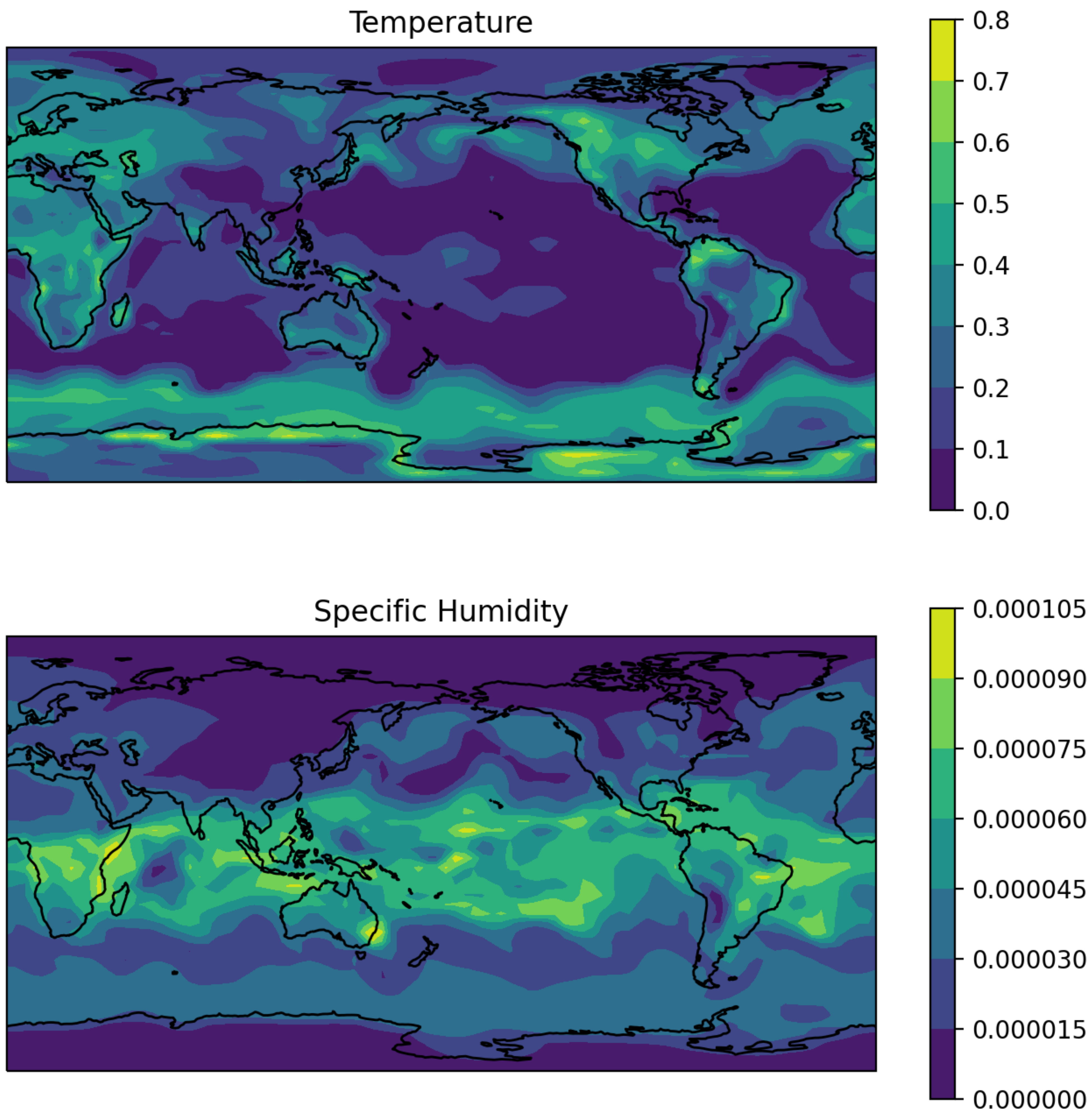
- Multiple target outputs
- Parallelized optimization and fitting procedures – Multiprocessing (python) and GPUs (CUDA)
- Advanced experimental design methods (MICE)

[1] Multiple-Output Gaussian Process: https://github.com/alan-turing-institute/mogp-emulator

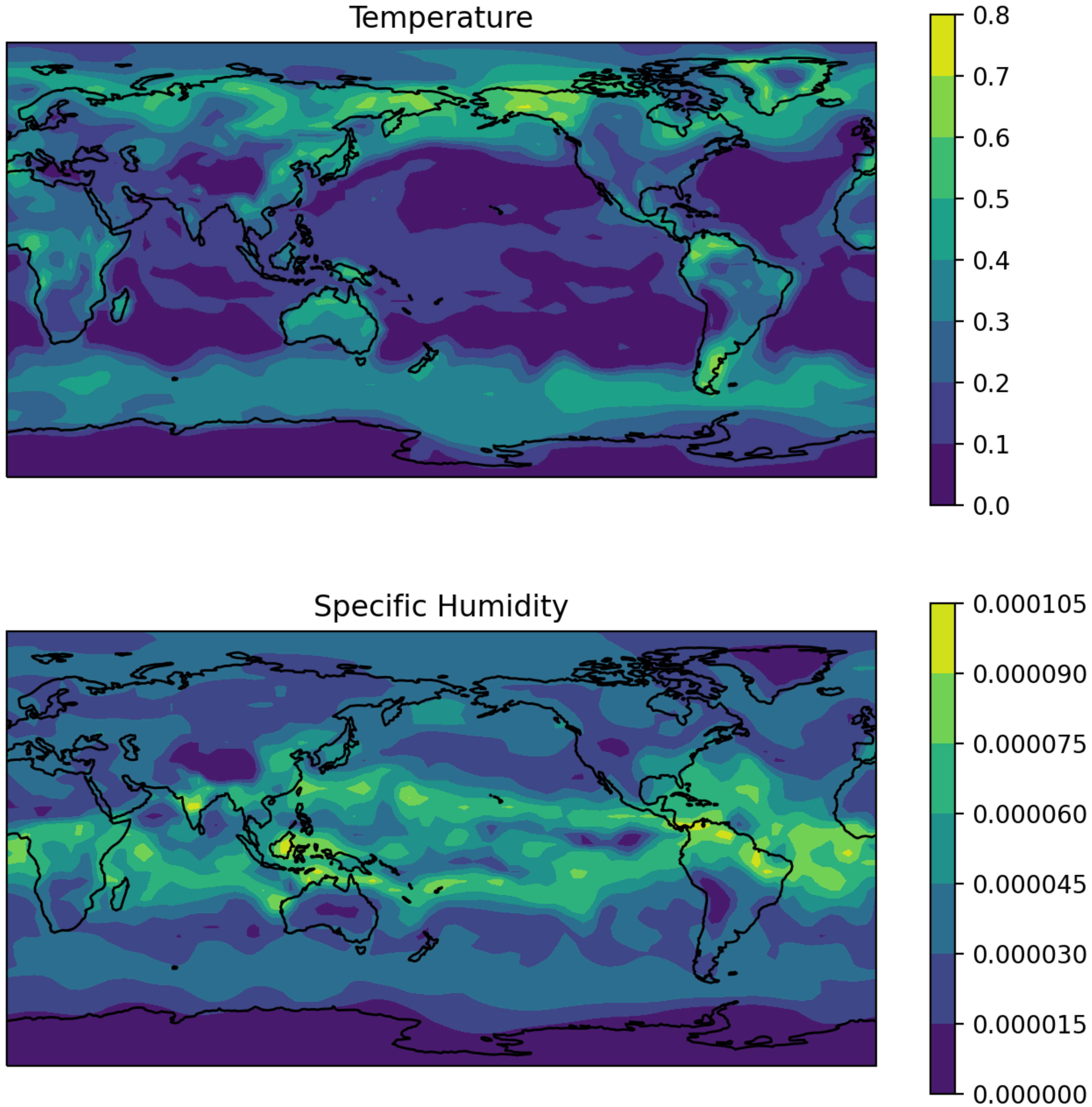
Results - (Sanity Check)

MOGP ‘One-shot’ Prediction Samples - 925hPa (bottom layer)

1st January

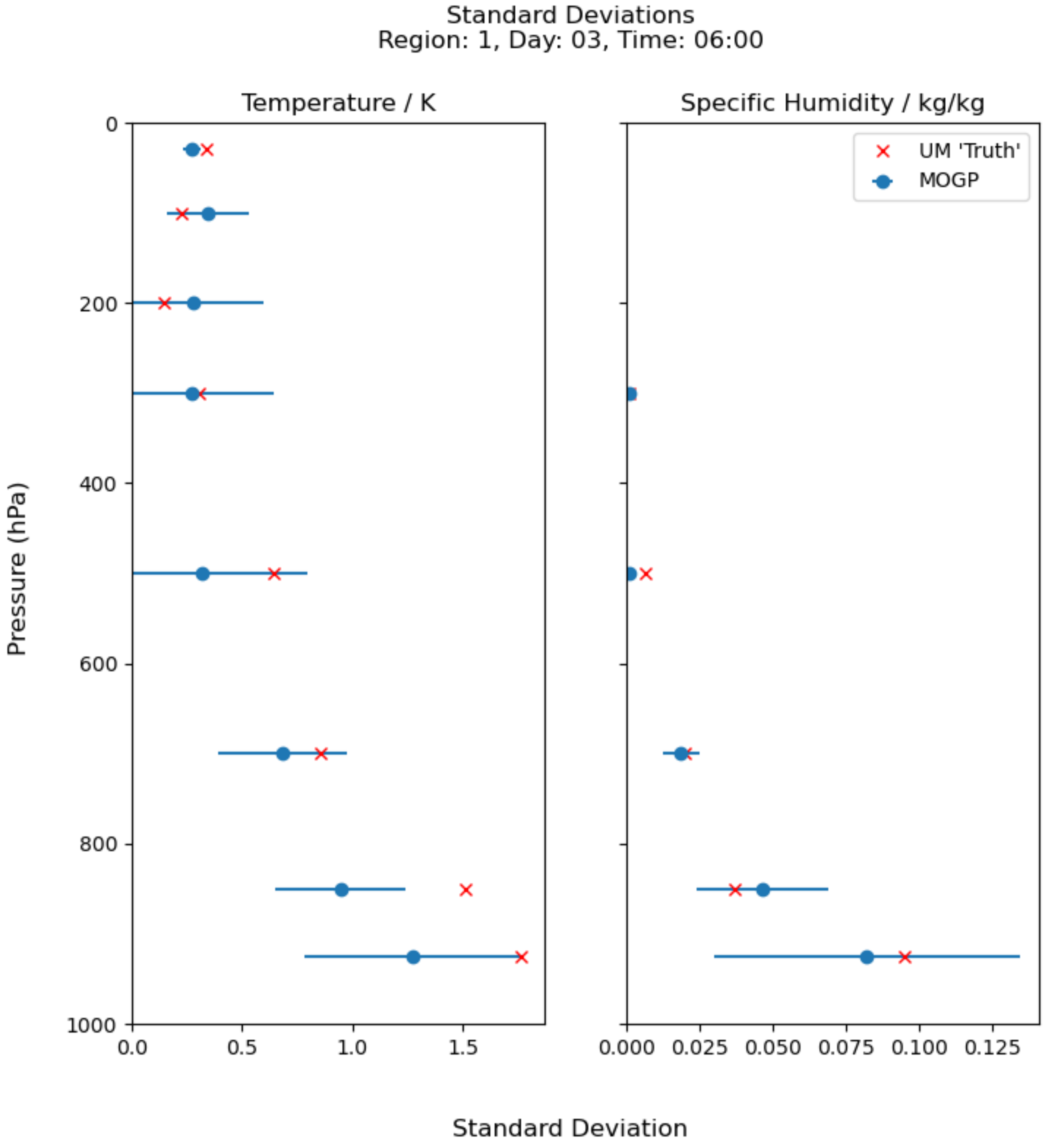


1st June



Validation

- An example of a testing profile
- MOGP captures the **overall trend** in the vertical direction of the standard deviations



Workflow

1. Generate the high resolution training data - Unified Model (Met Office)
2. Train a statistical model to predict σ_T and σ_Q at each layer of the atmosphere
3. Embed the trained model into a coarse resolution climate model

Proof of concept coupling

SPEEDY[2] a simplified AGCM

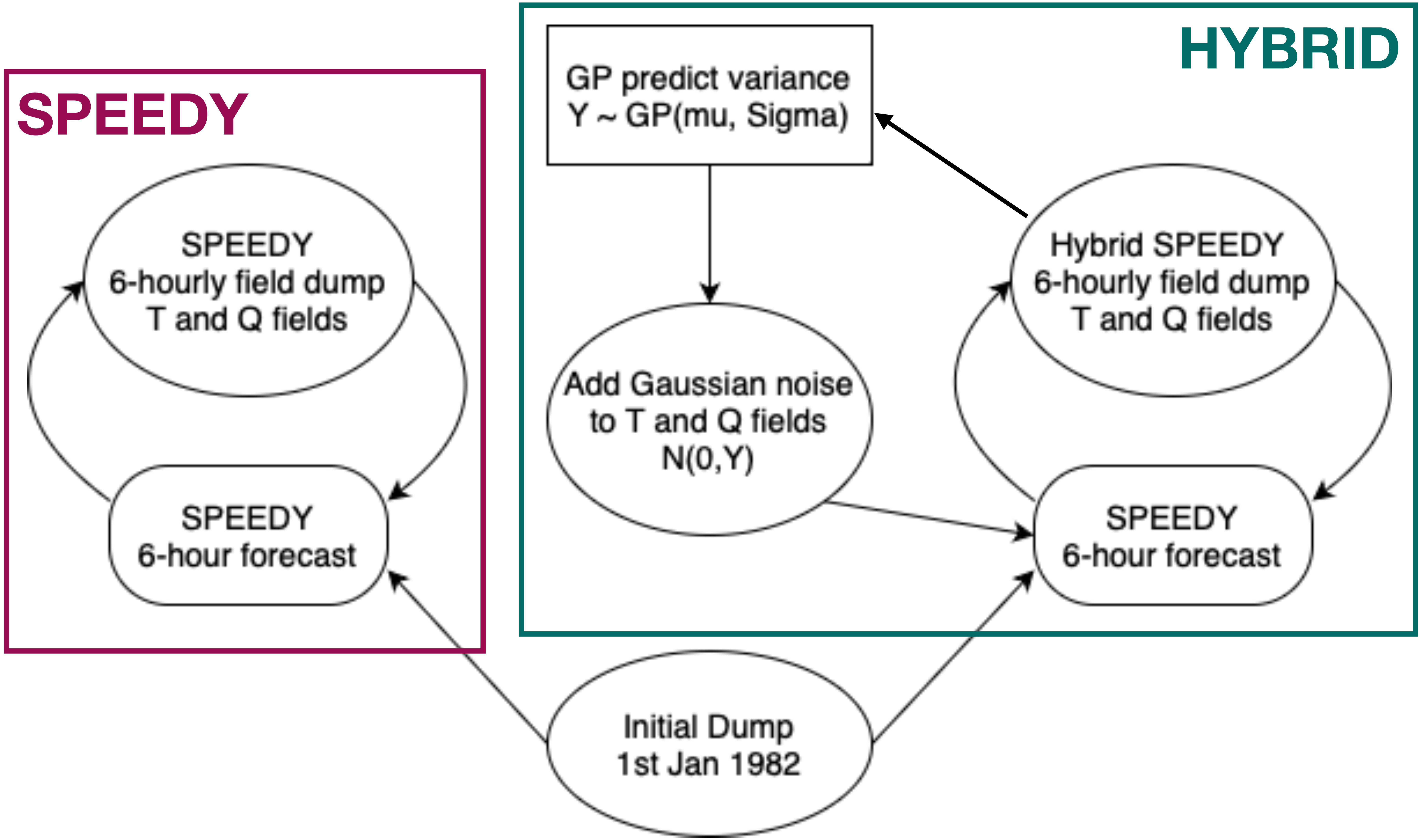
- SPEEDY (Simplified Parameterizations, privitivE-Equation Dynamics) is a simplified GCM (Fortran)
- Run at a very coarse resolution T30 (96 x 48 x 8 grid)
- Computationally cheap (can be run on a laptop)
- Coupled with MOGP by using in-house developed python wrappers

[2] SPEEDY: https://www.ictp.it/research/esp/models/speedy.aspx
http://users.ictp.it/~kucharsk/speedy-net.html

Experimental set up

10-year simulation
(1982 – 1992)

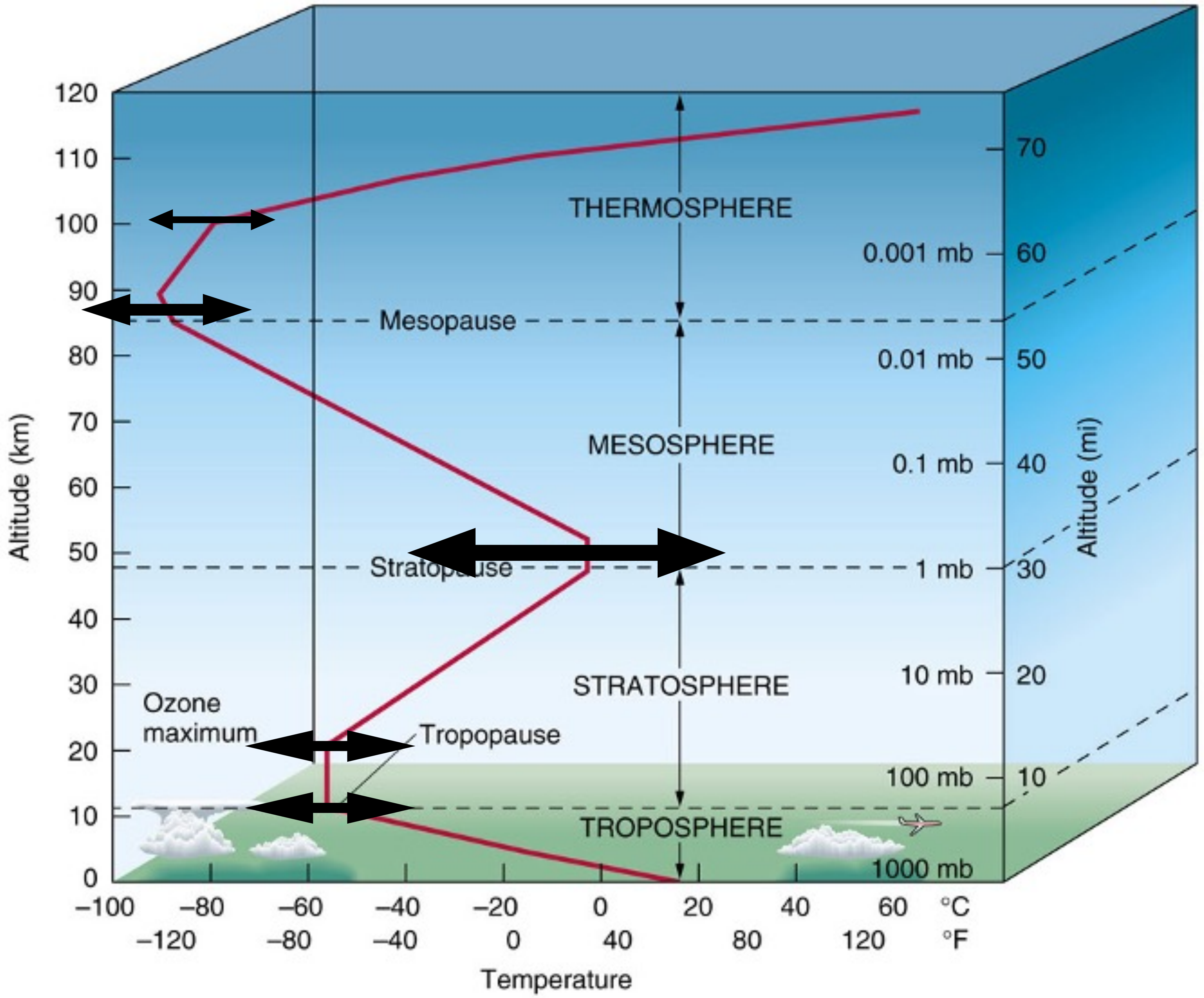
Every 6 hours σ_T
and σ_Q are predicted



Experimental set up

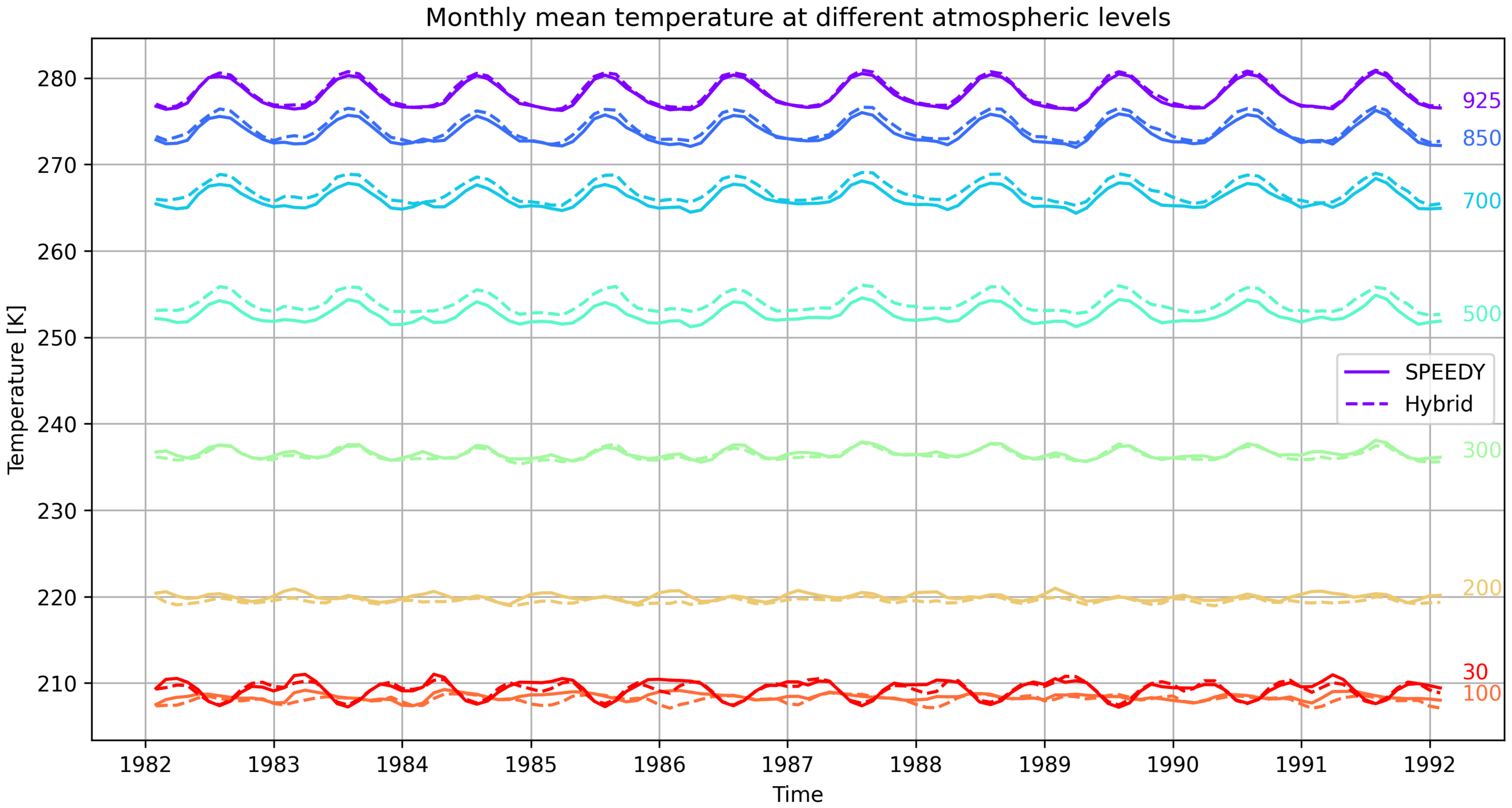
- Every 6 hours σ_T and σ_Q are predicted
- **Cell values are perturbed**
  - For cell I: $x_i = x_i + \epsilon_i$
  - $\epsilon_i \sim N(0,\sigma_i + \tau_i)$

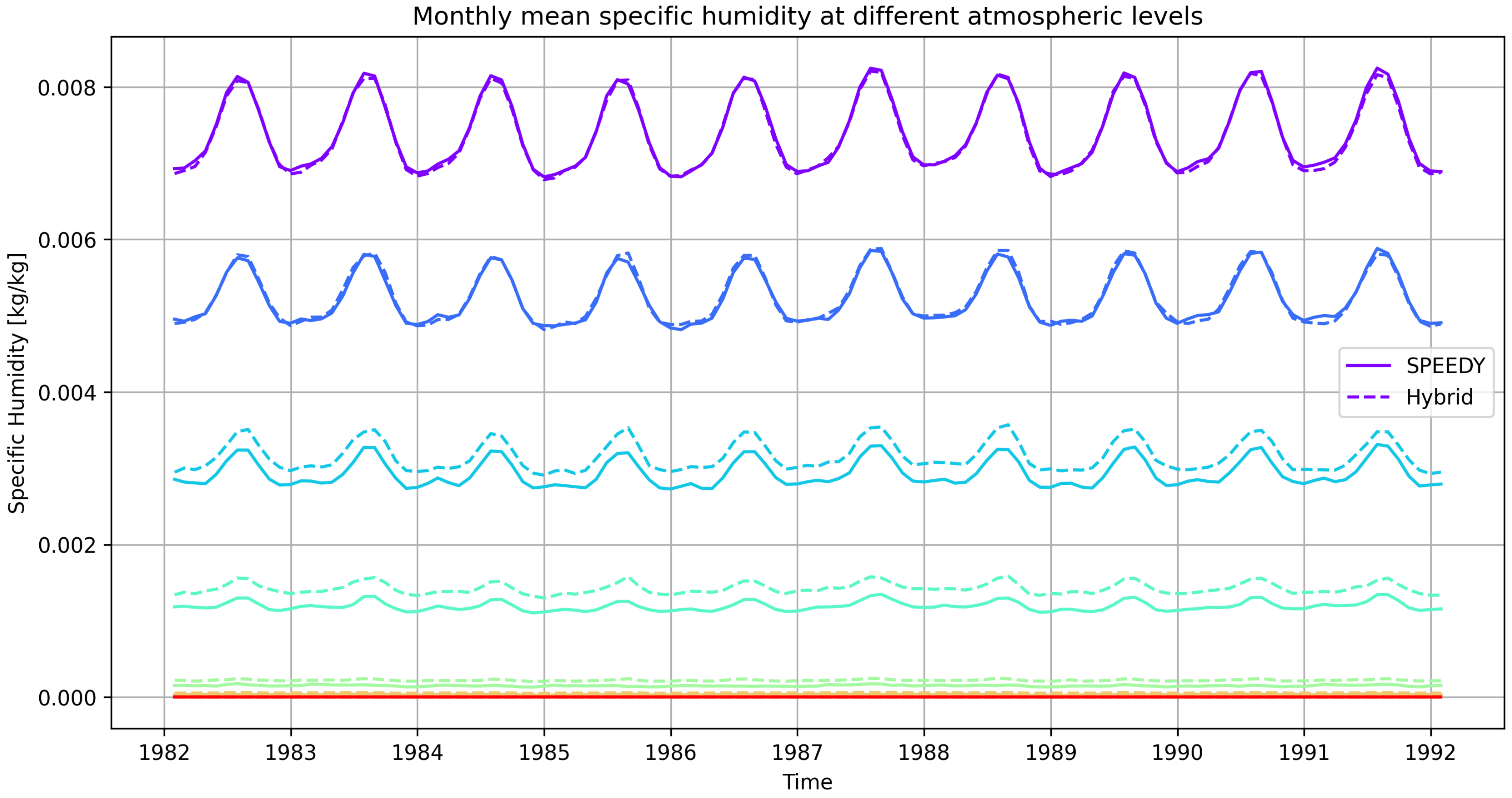
GP's predicted uncertainty



Sanity check for long-term drift

Temperature and Specific Humidity



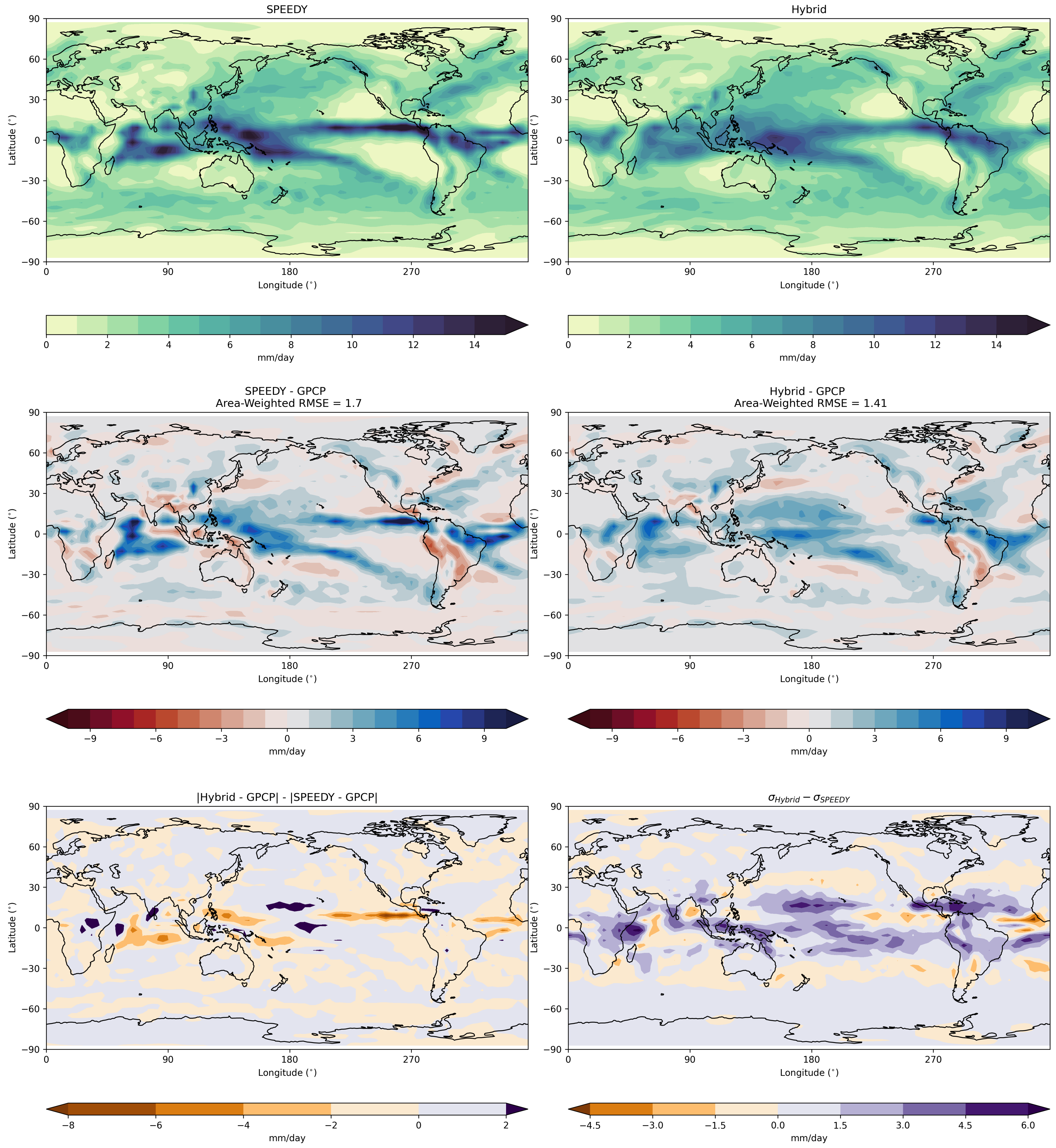


Monthly average at each level of the vertical for temperature and specific humidity

Results - Comparison to GPCP

Mean Precipitation

- GPCP is a satellite derived global precipitation product and is considered to be the ground truth here.
- Reductions in the strength of the double Inter-Tropical Convergence Zone (ITCZ) and Indian Ocean dipole can be seen
- These are also long standing biases of fully fledged GCMs
- 17% reduction in global area-weighted RMSE (20% reduction in the tropics)



Results - Comparison to ERA5

Robustness and naive perturbations

- Hybrid run is **repeated** to test **robustness**
- Perturbing the profiles in a **naive** manner acts as the baseline model to compare against.
  - **Naive approach:** The profiles are perturbed by additive Gaussian noise given by $\mathcal{N}(0, \epsilon\mu_{T/Q})$, where $\mu_{T/Q}$ is the mean values and $\epsilon \in \{0.01, 0.05, 0.1\}$

	Global Area-weighted RMSE			Tropics Area-weighted RMSE		
Ground truth source	IPCP	ERA5	ERA5	IPCP	ERA5	ERA5
Data Field	Precipitation [mm/day]	T [K]	Q [kg/kg] $\times 10^{-3}$	Precipitation [mm/day]	T [K]	Q [kg/kg] $\times 10^{-3}$
SPEEDY Control	1.702	**3.352**	1.657	3.104	**3.450**	2.489
Hybrid	1.412 (17.01%)	3.397	1.638	**2.487** (19.87%)	3.691	2.474
	1.411 (17.06%)	3.394	1.637	2.495 (19.61%)	3.691	2.469
	1.420 (16.50%)	3.410	1.637	2.514 (19.01%)	3.688	2.467
Naive ($\epsilon = 0.01$)	1.584 (6.92%)	3.520	1.580	2.860 (7.87%)	3.991	2.316
	1.590 (6.56%)	3.538	1.583	2.872 (7.48%)	3.995	2.318
	1.580 (7.16%)	3.534	1.582	2.857 (7.96%)	3.996	2.321
Naive ($\epsilon = 0.05$)	5.049	3.563	1.213	8.917	4.811	2.111
	5.044	3.577	1.213	8.902	4.842	2.115
	5.037	3.574	**1.204**	8.885	4.849	**2.095**
Naive ($\epsilon = 0.1$)	N/A	N/A	N/A	N/A	N/A	N/A

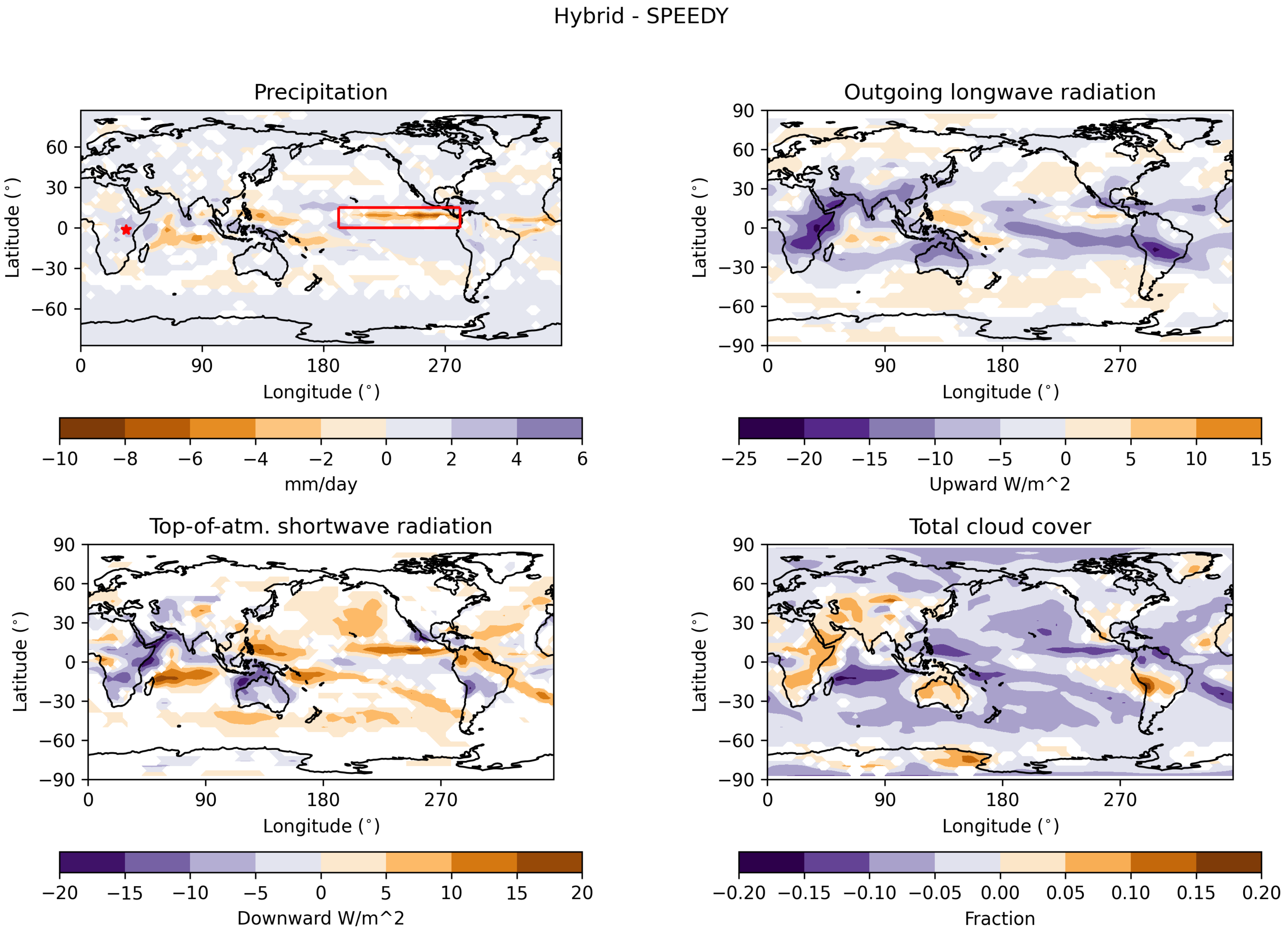
Table 1 Global and tropics area weighted precipitation, temperature (T at 925 hPa) and specific humidity (Q at 925 hPa) RMSE values against GPCP and ERA5 data for the control SPEEDY, hybrid and naively perturbed runs. For cases of improvement the percentage relative error against the SPEEDY control run are also calculated. Note that for $\epsilon = 0.05$ there is a large degradation in the precipitation RMSE despite having lower RMSE values for Q.

Understand the drivers

Understand the drivers

Field differences

- Increase/decrease in precipitation is consistent with the increases/ decreases in cloud cover and radiation patterns.

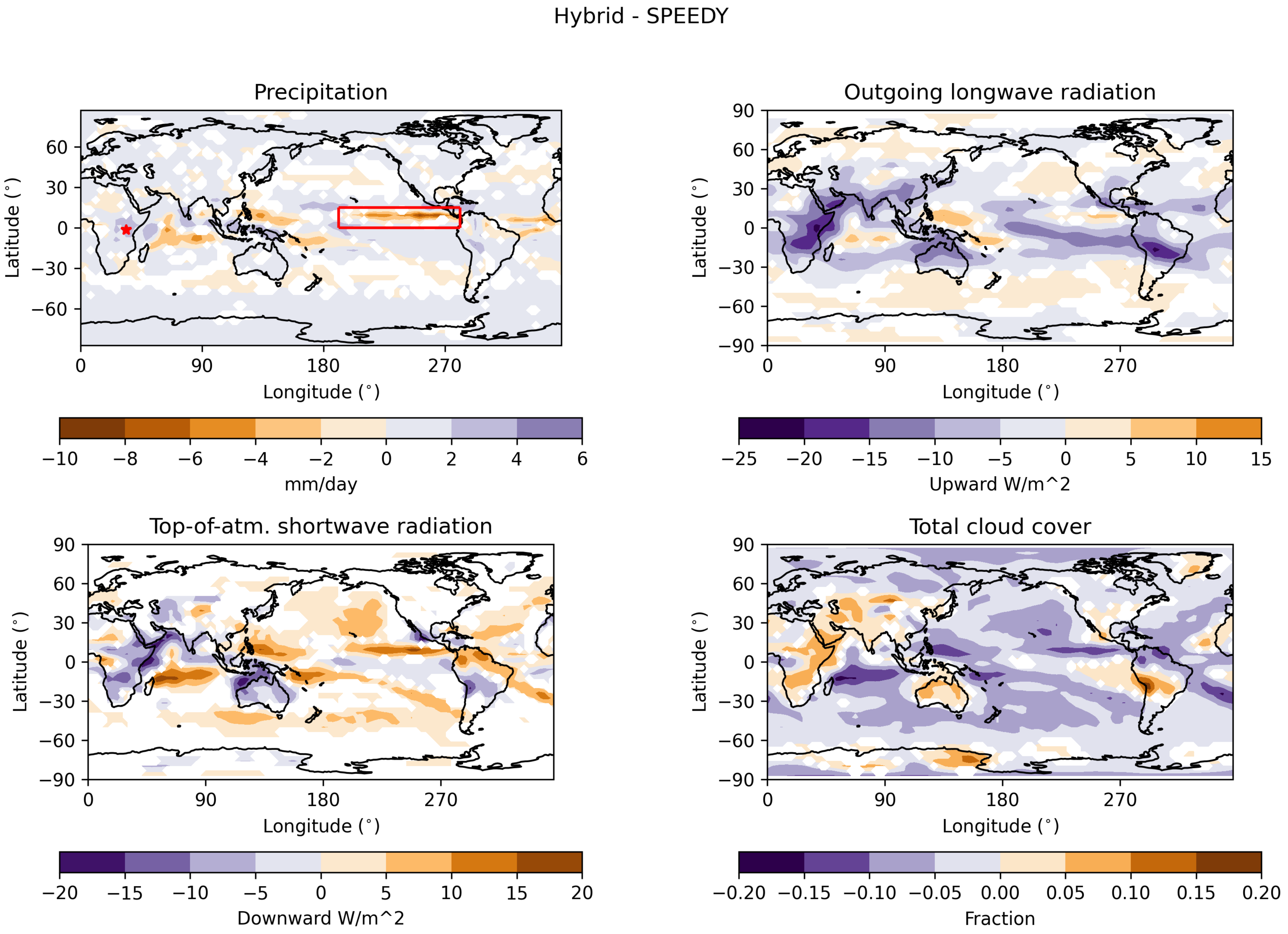


Results

Field differences

- Increase/decrease in precipitation is consistent with the increases/ decreases in cloud cover and radiation patterns.

- Key areas of interest for further analysis:
    - Central Africa
    - Pacific region off the coast of Central America



Results - Central Africa

Increased precipitation

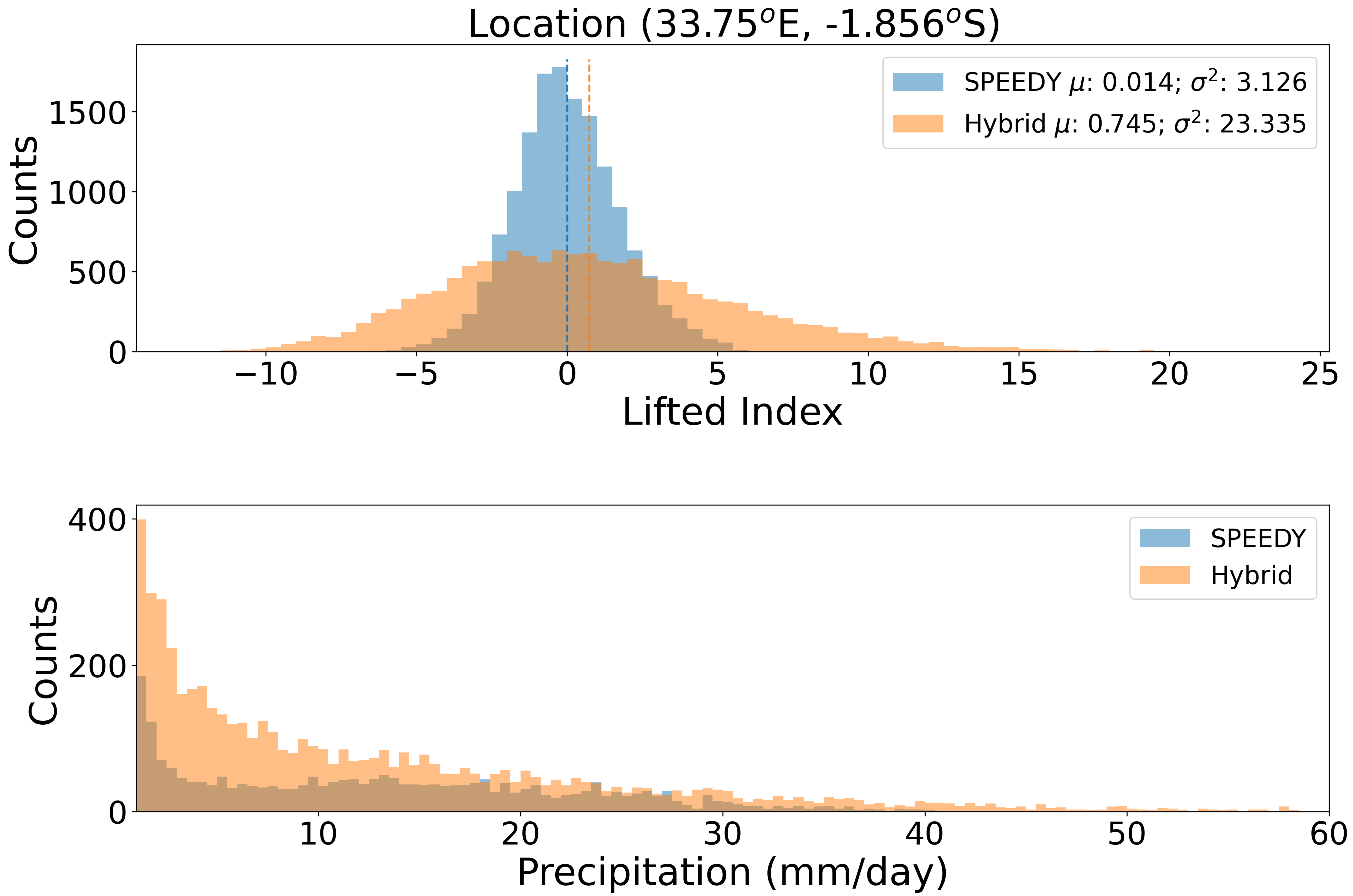
- Lifted index is a **measure of the atmospheric instability** and is used to estimate the development of thunderstorms.
- The lifted index is calculated by taking a parcel of air from the surface and lifting it adiabatically (without heat transfer) to a certain level in the atmosphere.
- The temperature of this lifted parcel is then compared to the temperature of the surrounding environment at that level.

Lifted Index	Stability
Greater than 0	Stable
0 to -4	Marginal Stability
-4 to -7	Large Instability
-7 or less	Extreme Instability

Results - Central Africa

Increased precipitation

- The lifted index is calculated for each time step of the 10 year simulations.
- Histogram shows an increase in large unstable values (< -4) which contribute to extreme precipitation events.



Results - Pacific Region

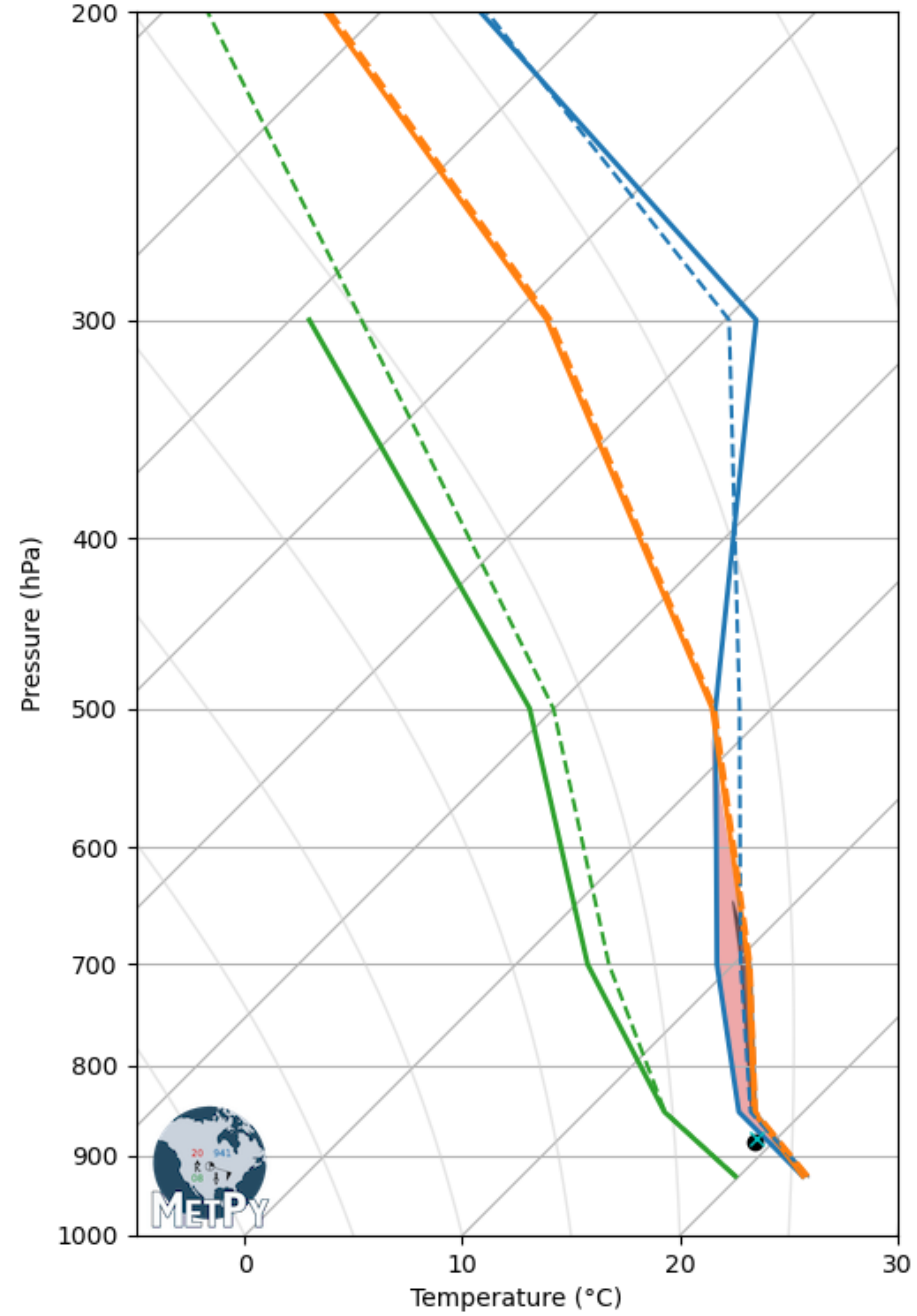
Skew-T log-p diagram for thermodynamic profile

- In this area the precipitation is reduced in the hybrid run.

- In the boundary layer, the temperatures and humidities are similar leading to similar lifting condensation levels for the SPEEDY control (black dot) and hybrid experiment (cyan cross) and similar moist adiabats for the ascents.

- However, as a result of the different tropospheric temperature profiles, the fates of the two ascents are quite different.

Dry-bulb temperature in blue, dew-point in green and surface-parcel ascent in orange.

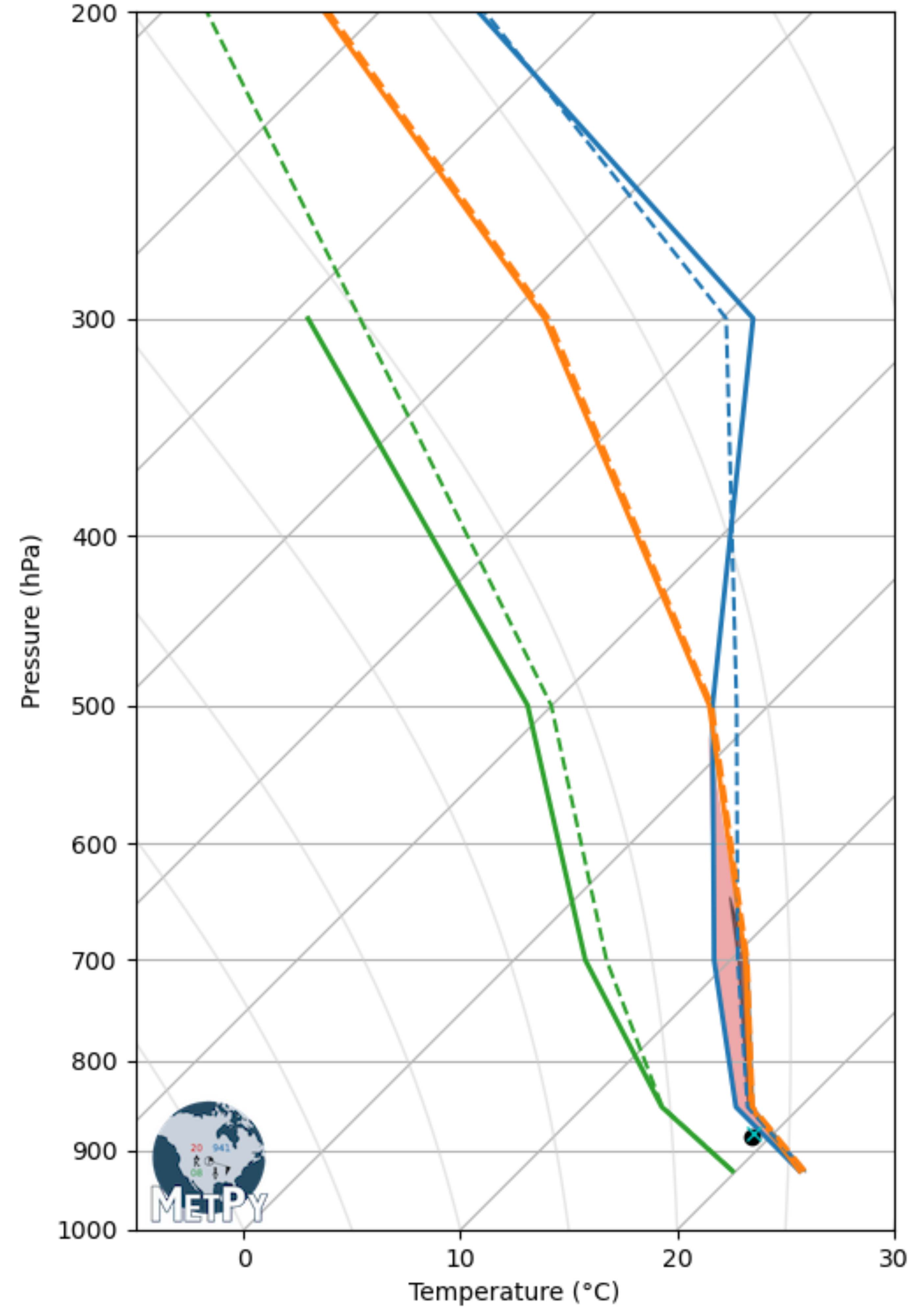


Results - Pacific Region

Skew-T log-p diagram for thermodynamic profile

- In the SPEEDY control these parcel ascents can rise to 520 hPa, corresponding to a convective cloud top of around 5.3 km where the environment temperature is -2˚C.
  - The shaded pink area, the Convective Available Potential Energy (CAPE), is 122 J/kg.
- The hybrid run has a moister troposphere and importantly a warmer troposphere. Due to the warmer mid-troposphere the hybrid experiment has a parcel than can only rise as far as 645 hPa.
  - This is a height of around 3.6 km where the temperature is +7˚C. The CAPE is now only 25 J/kg (and the black shading can hardly be seen).

Dry-bulb temperature in blue, dew-point in green and surface-parcel ascent in orange.



Conclusions

- **Coarse grained high resolution GCM runs** to generate a training data set.
- **Trained a MOGP model to predict σ_T and σ_Q** at each level of the atmosphere.
- **Coupled** the trained model to a **simplified** GCM to augment the dynamics.
- Showcased improvements to the simplified GCM biases, in particular biases associated with precipitation patterns.

Conclusions

- **Coarse grained high resolution GCM runs** to generate a training data set.
- **Trained a MOGP model to predict σ_T and σ_Q** at each level of the atmosphere.
- **Coupled** the trained model to a **simplified** GCM to augment the dynamics.
- Showcased improvements to the simplified GCM biases, in particular biases associated with precipitation patterns.
- Adding **mean zero noise has a nonlinear effect** to the dynamics.
  - Technique could be used in conjunction with **ensemble approaches** to diversify members.
  - Apply to **other modelling settings (ocean profiles).**

Ongoing Work

Fully Fledged GCM

- Integrating the technique into a **fully fledged GCM** (CESM CAM)
  - This requires **a lot of software engineering**
- Aim is to use the latest in heterogenous computing architectures (Nvidia Grace Hopper chip)
- Efforts have been led by **Scott O'Connell (UCL)**

ML Model and Training Data

- DYAMOND dataset
- Investigating alternative ML models to predict the standard deviations
- Physics constraints and spatial covariances
- In collaboration with the IIT Delhi Climate modelling group - Prof. Saroj Kanta Misha and Debi Prasad

Integration into CAM

Heterogeneous Computing

- Nvidia Grace Hopper Superchip
- Ideal for this application where the Fortran code can run on the Grace CPU and the ML model can run on the Hopper GPU (H100)
- Hurdles to get this to work
    - ARM based architecture
    - Integrate Fortran with Python

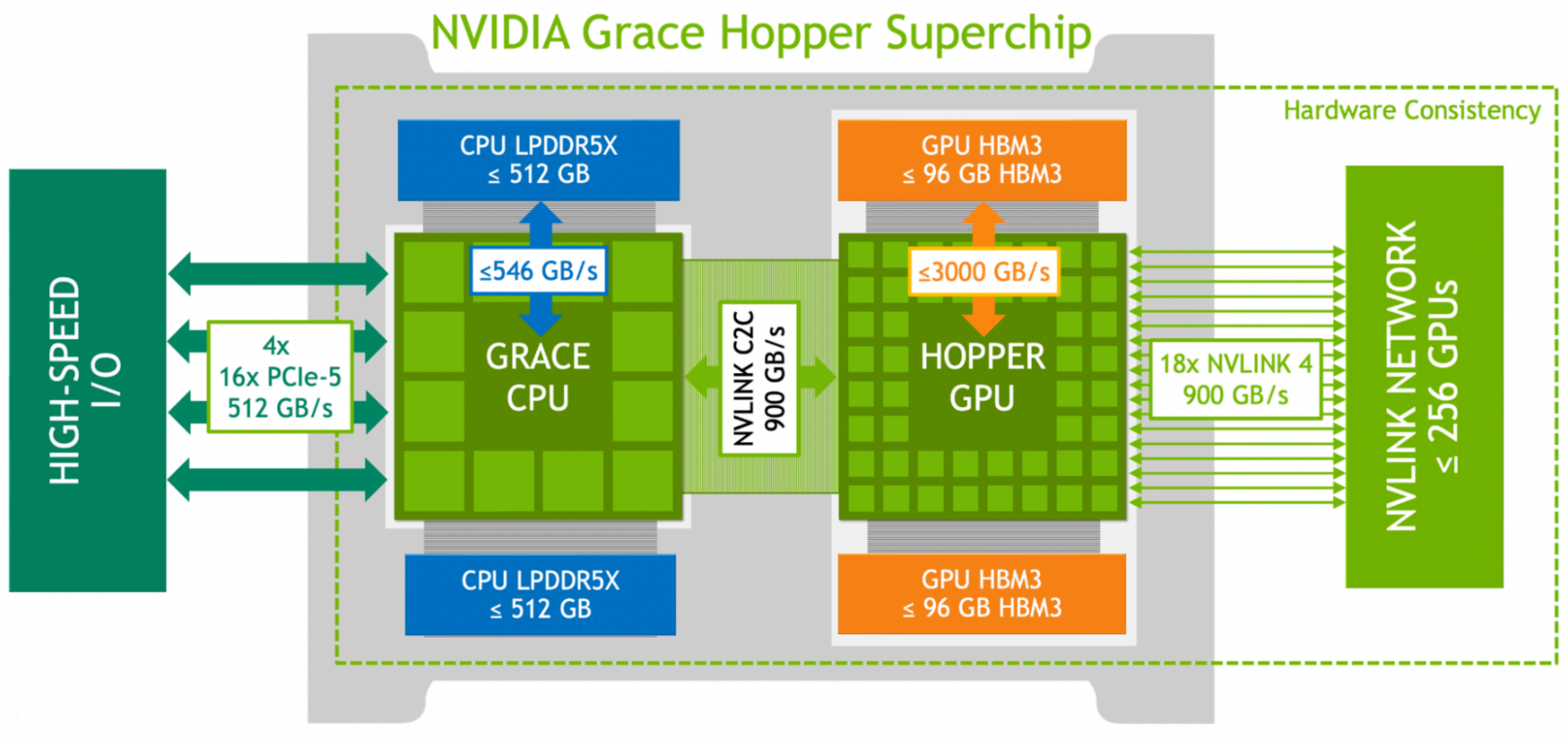
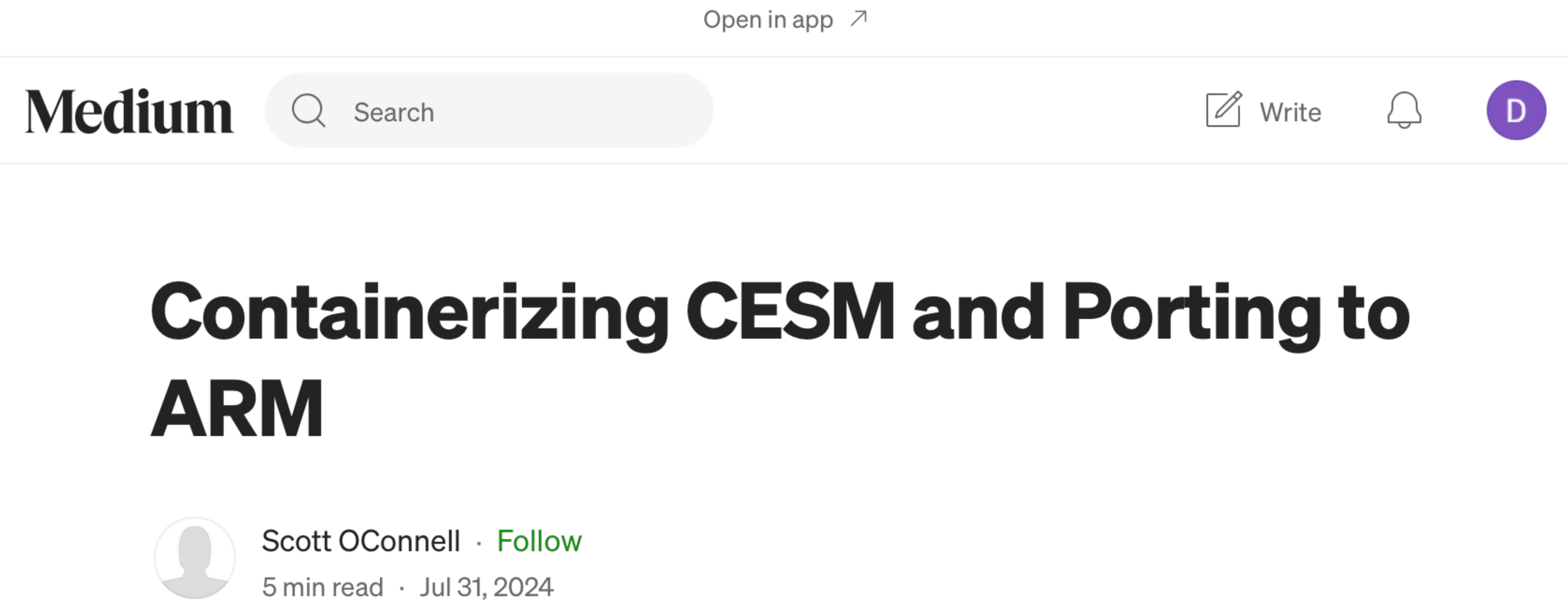


Figure 2. NVIDIA Grace Hopper Superchip logical overview

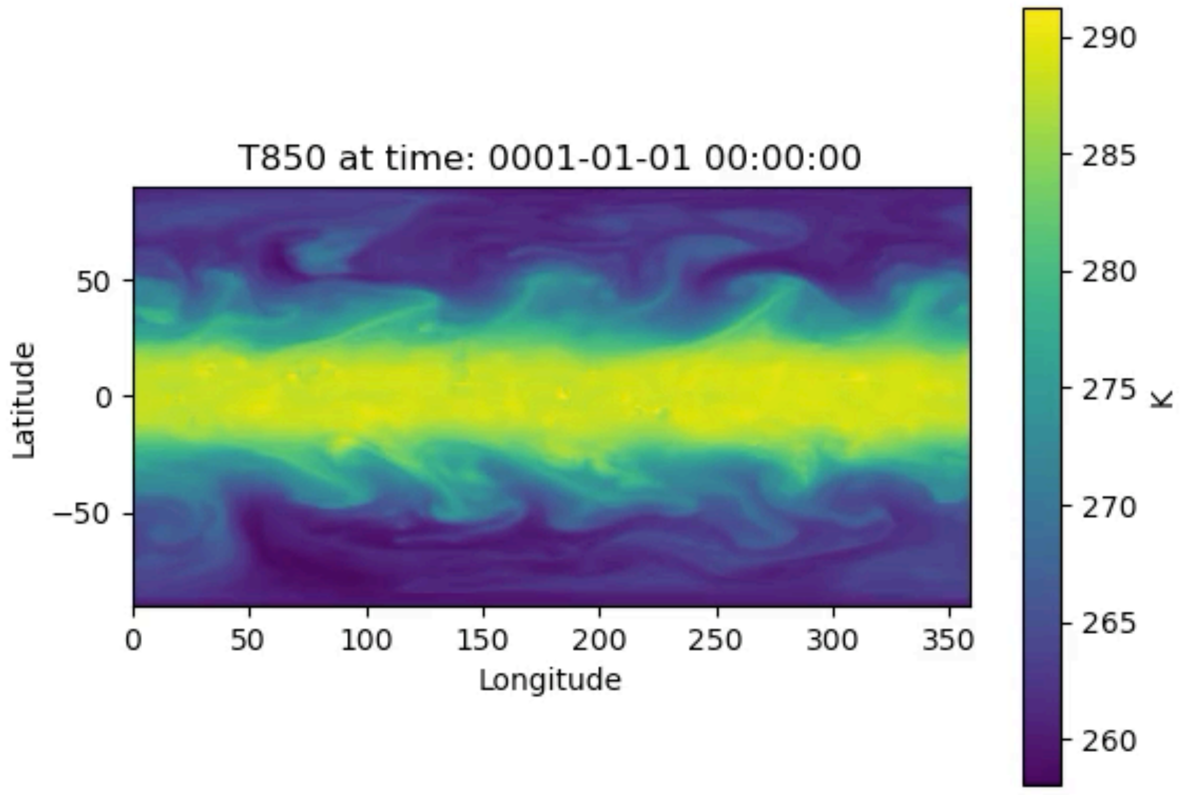
Integration into CAM

ARM based architecture

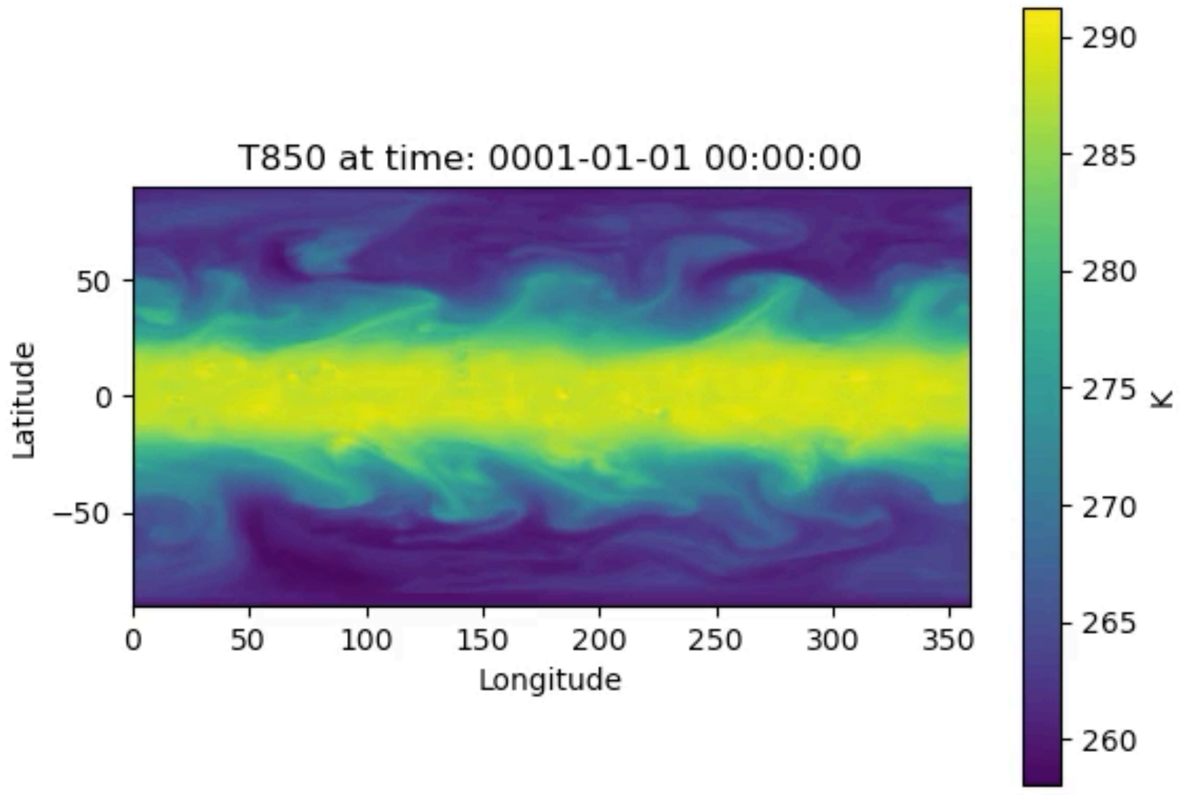
- Containerised development of CESM
  - First instance of CESM running on the Grace CPU
  - Ease the transition from an x86 instruction set to ARM.
  - Making CESM accessible to new users on new machines.
  - Aids with reproducibility
- Github: https://github.com/scotty110/CESM_Docker



Medium blogpost: https://medium.com/@twins.corgi.0a/containerizing-cesm-and-porting-to-arm-b9419ed939af



Aqua Planet Simulation on ARM

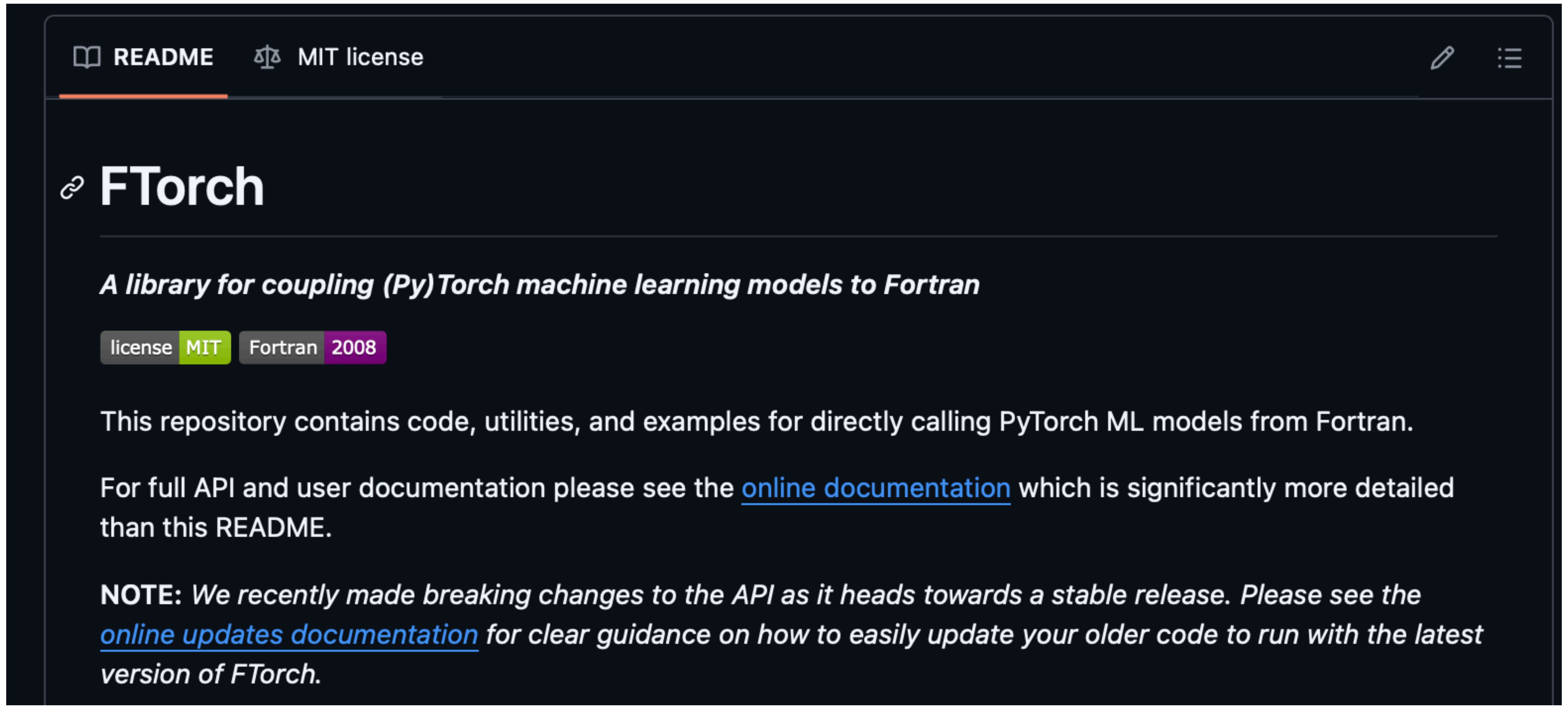


Aqua Planet Simulation on x86

Integration into CAM

Integrate Fortran with Python

- Built upon the the FTorch library which enables trained PyTorch ML models to be called directly from Fortran.
- Early results showcase minimal overheads on runtimes.


README MIT license

FTorch

A library for coupling (Py)Torch machine learning models to Fortran

license MIT Fortran 2008

This repository contains code, utilities, and examples for directly calling PyTorch ML models from Fortran.

For full API and user documentation please see the online documentation which is significantly more detailed than this README.

NOTE: *We recently made breaking changes to the API as it heads towards a stable release. Please see the online updates documentation for clear guidance on how to easily update your older code to run with the latest version of FTorch.*

We strongly encourage anyone who is interested in integrating ML models with CESM to reach out!

Thank you