Machine learning for parametrised physics

ECMWF annual seminar 2022

Matthew Chantry

With much help from:

Robin Hogan, Antonino Bonanni, Peter Ukkonen, Peter Duben,

Richard Forbes & Physical processes team

matthew.chantry@ecmwf.int

ECMWF

What is machine learning?

- Here, machine learning is a short-hand for supervised machine learning.

- Supervised machine learning:
  - Requires a dataset of inputs and outputs.
  - Learning a model to map from inputs to outputs.
  - Model has parameters, which are learnt (for neural networks this training is gradient descent).
  - The learning seeks to optimise some function.

- Given enough data, and enough model parameters, any deterministic mapping can be learnt.
  - No guarantees that this will be computationally tractable.
  - However, it often is!

How might machine learning be used for parametrised physics?

<u>Emulate existing model component</u>

Learn an operational scheme
Reduce computational cost
Port to GPUs
TL/Ad (see later)

<u>Examples</u>
Chevallier (Radiation 1990!)
Krasnoposky (Radiation + more)
Song & Roh (Radiation)
Chantry (NOGWD)
Espinosa (NOGWD)

<u>Emulate increased complexity model component</u>

Learn an unaffordable scheme
Reduce computational cost
…

<u>Examples</u>
Meyer (Radiation)
Gettelman (Cloud microphysics)

<u>Learn new parametrisation scheme</u>

Use data from high resolution simulations or observations
Greater challenges for model stability

<u>Examples</u>
Yuval & O'Gorman (Convection, subgrid momentum)
Brenowitz & Bretherton (Convection)
Beucler, Pritchard, Gentine, Rasp (Convection)

What does the emulation workflow look like?

- Identify inputs & outputs.

- Run existing model (perhaps coupled to ESM).
  - Save inputs & outputs.

- Train machine learning model to reproduce the input->output mapping.

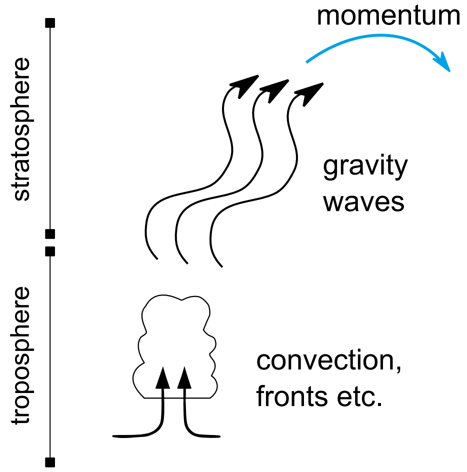
- Connect machine learning model back into ESM.

- Run simulations to understand coupled impact of emulator on forecasts.

First exploration: Non-orographic gravity wave drag

Chantry et al. 2021

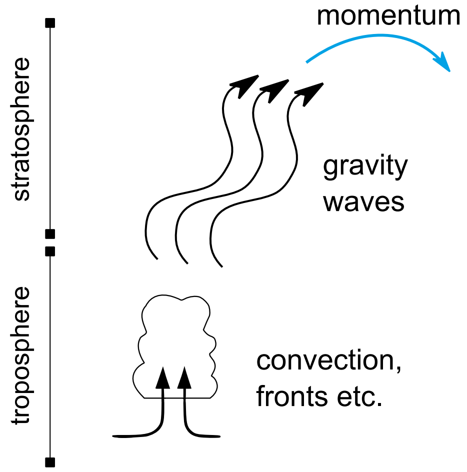
- Capture impact of unresolved momentum on resolved flow.
- Important for quasi-biennial oscillation.
- Generate data from existing scheme.
- Recreate with “simple” network.

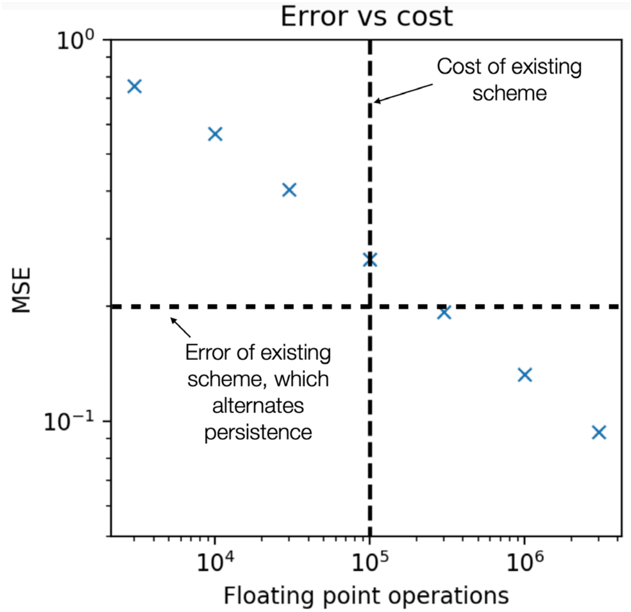


First exploration: Non-orographic gravity wave drag

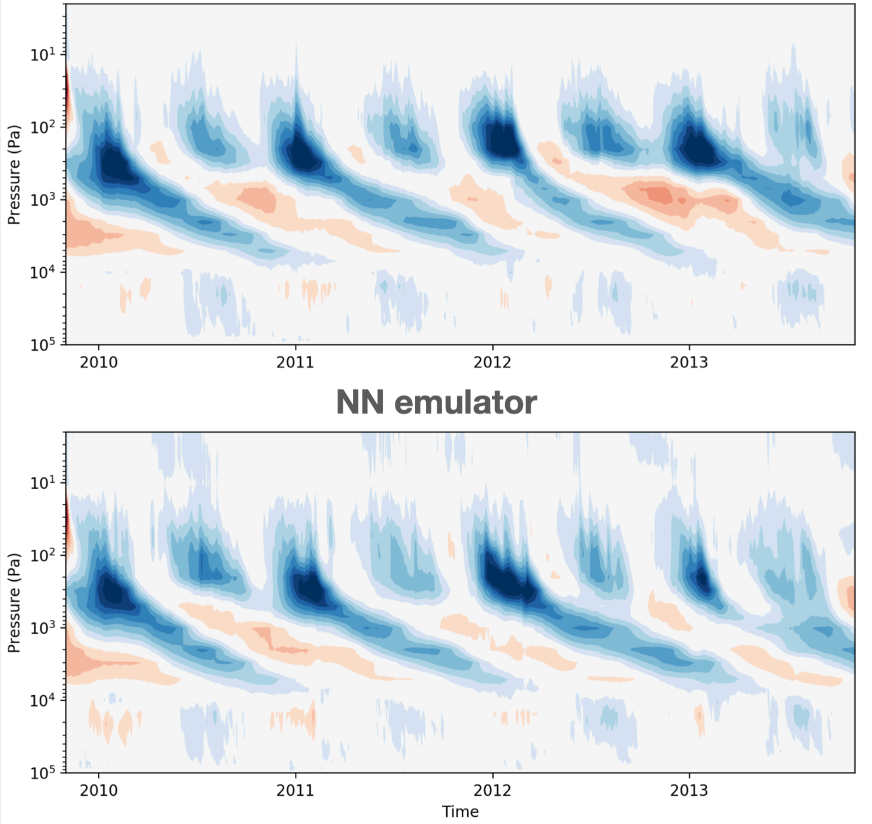
Chantry et al. 2021

- Capture impact of unresolved momentum on resolved flow.
- Important for quasi-biennial oscillation.
- Generate data from existing scheme.
- Recreate with “simple” network.
- Find that offline error is a tuneable parameter with complexity.



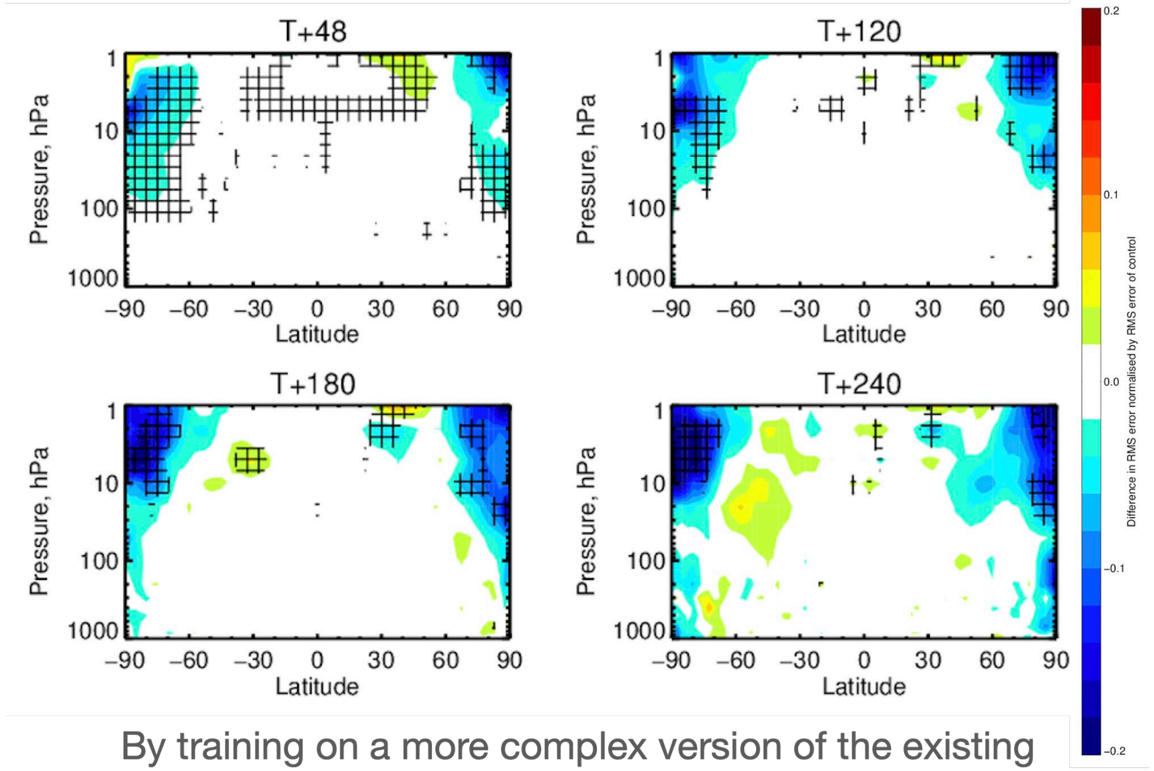


Non-orographic gravity wave drag: coupled results



Visualisation of the equatorial zonal jet showing the quasi-biennial oscillation.

Temperature predictions errors relative to existing parameterisation scheme



By training on a more complex version of the existing parametrisation scheme we are able to reduce forecast errors with our neural network solutions

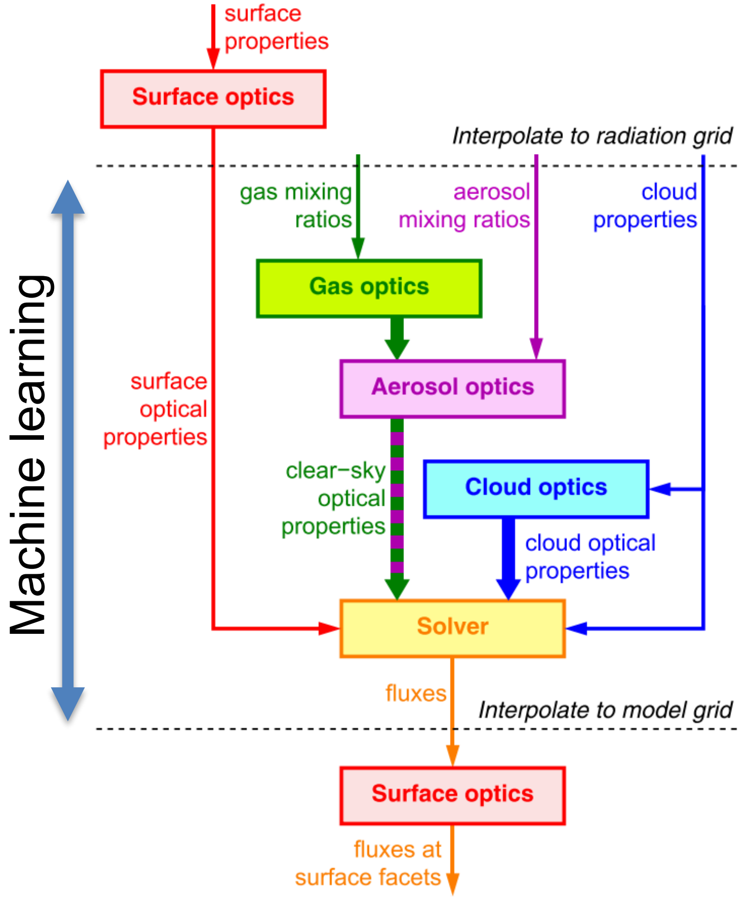
- IFS forecasts suffer no degradation when using NN for NOGWD.
- NN-based forecasts capture improvement from using more complexity in physical scheme.

NOGWD great test-bed for proof of concept…

… now onto bigger components.

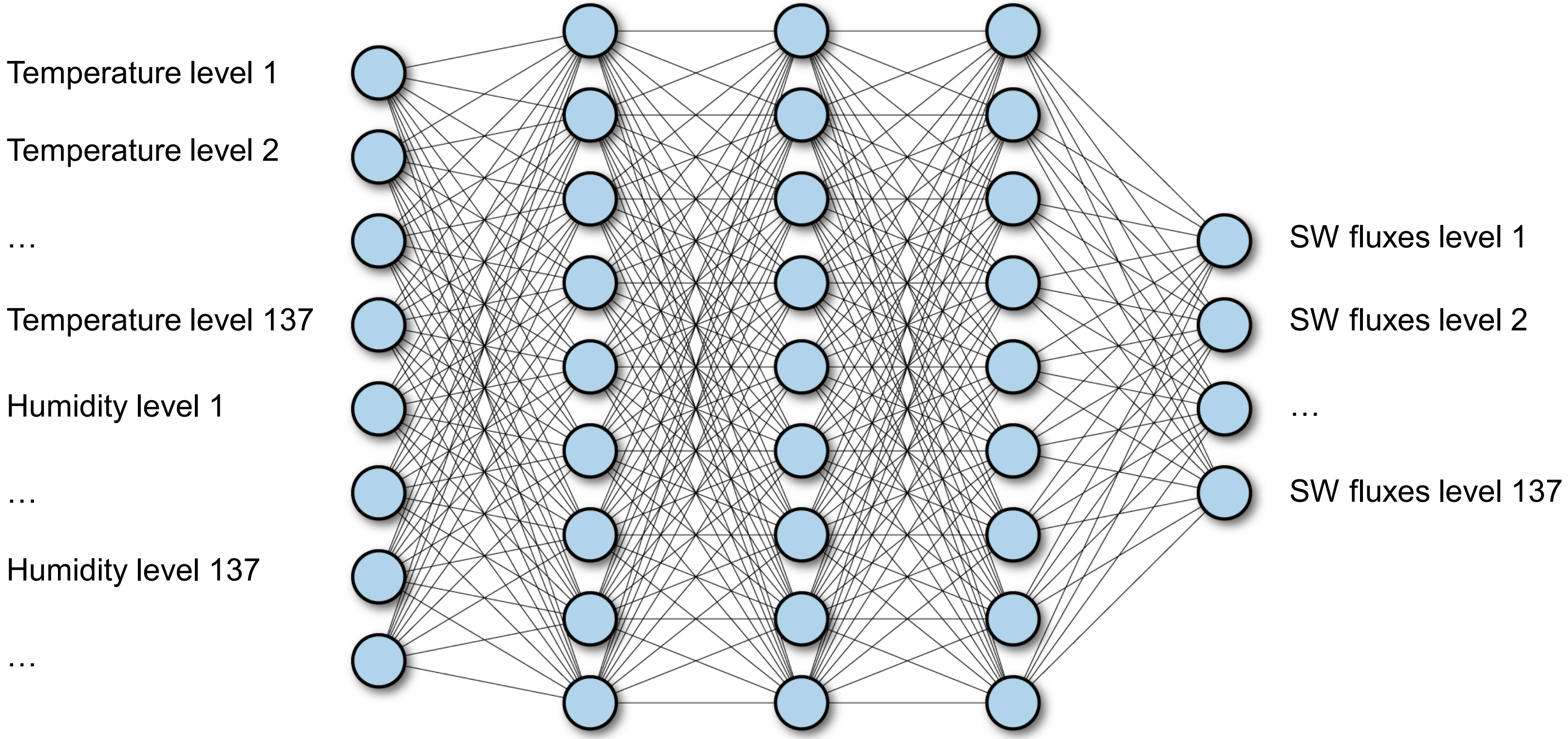
Radiation

- Much more expensive and impactful component of the IFS.
  - Specifically, TripleClouds solver (upcoming operational scheme).
- Include gas & aerosol mixing ratios to learn dependence.
- Existing scheme run at reduced temporal & spatial resolution.
  - Opportunity to use ML to increase this resolution?
- More complex (unaffordable) scheme exists, SPARTACUS, which includes 3D cloud effects.
  - See Meyer (2022) for more on this application.



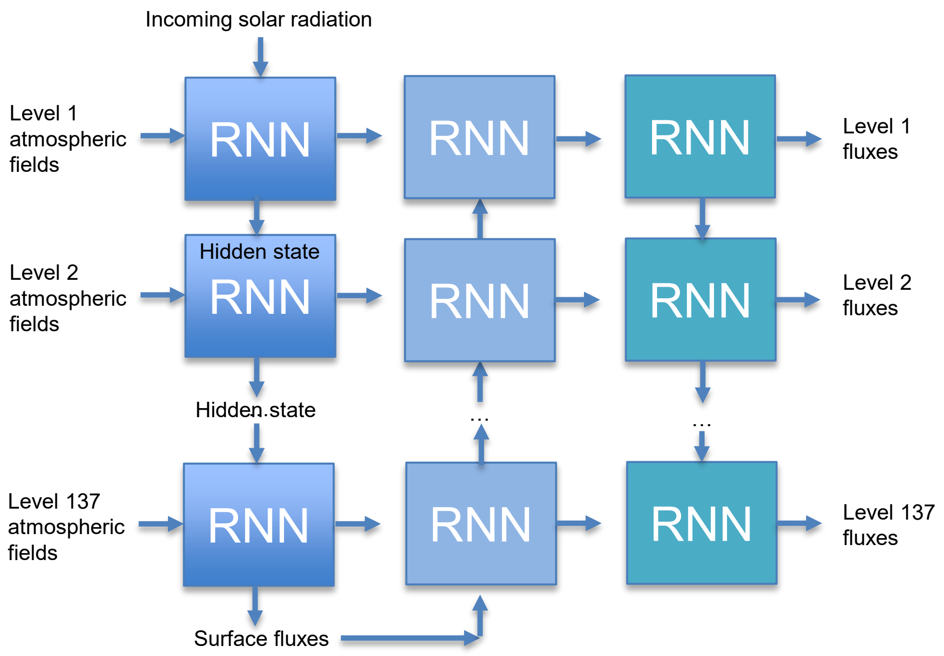
Keeping a human in the loop

- Toy image of a neural network is a fully-connected network.
  - Might make us nervous about spurious correlations.



Keeping a human in the loop

- Toy image of neural network is a fully-connected network.
  - Might make us nervous about spurious correlations.
- For shortwave heating, Ukkonen (2022) proposed a model to mimic the data flow of the existing solver.



Keeping a human in the loop

- Toy image of neural network is a fully-connected network.
  - Might make us nervous about spurious correlations.
- For shortwave heating, Ukkonen (2022) proposed a model to mimic the data flow of the existing solver.
- Results in relatively small number of trainable parameters.
- Imposes same mechanisms at each vertical level.

- Similar ideas used for longwave process.
  - Current best model based on convolutional layers.
  - Imposes similar ideas of information propagation & vertical invariance.

Keeping a human in the loop #2

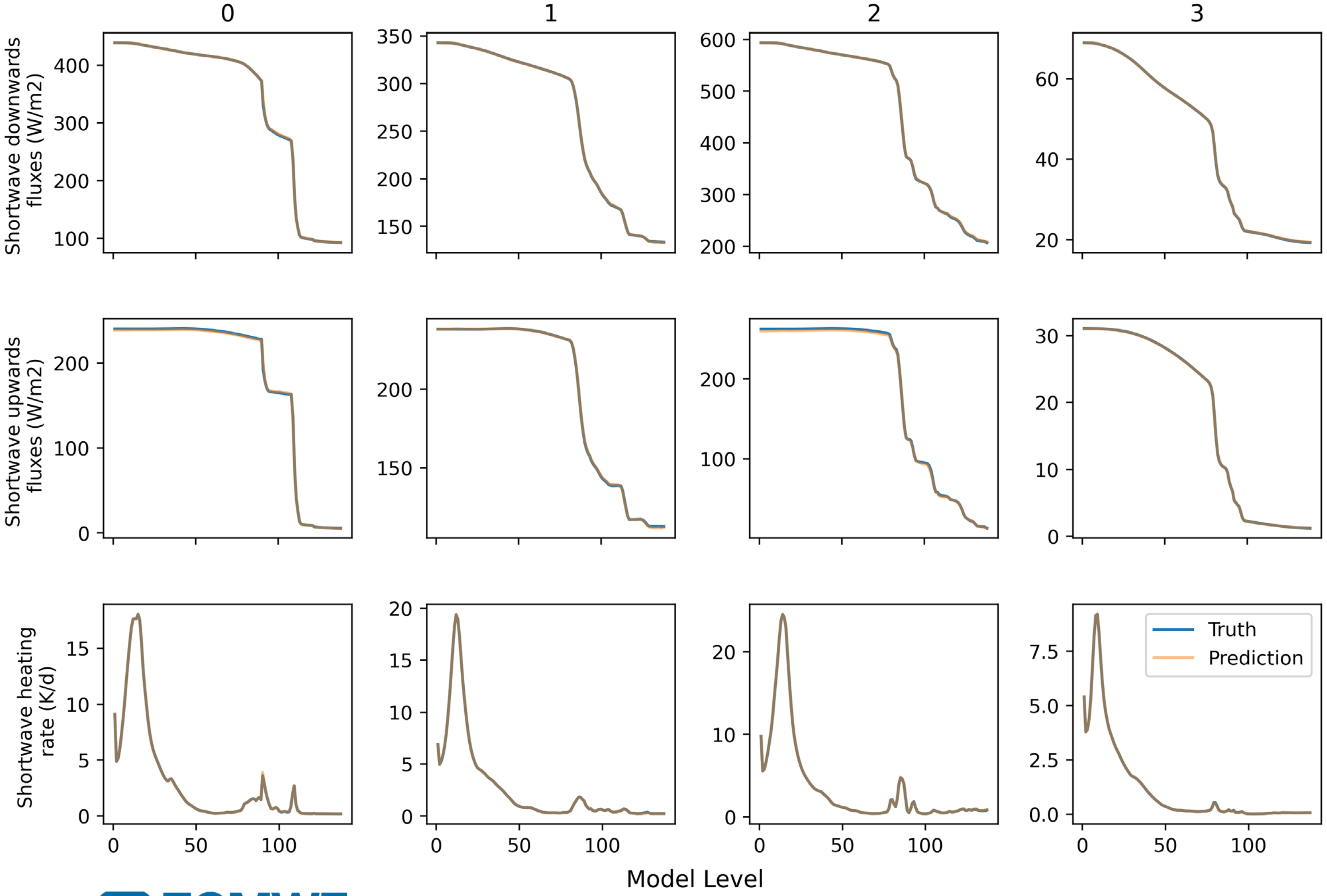
Radiative transfer has a “duality” between fluxes & heating rates.

- Former needed for boundaries.
- Latter needed to increment atmospheric temperature.
- Related by: $$\frac{dT}{dt} = -\frac{g}{c_p}\frac{F^n_{i+1/2} - F^n_{i-1/2}}{p_{i+i-1/2} - p_{i-i-1/2}},$$
- At the top of the atmosphere pressure differences are small, heating rate is very sensitive to small changes in fluxes.

- Encode this relationship in the neural network.
  - Predict both quantities, inherently coupled through the above.
  - Loss in training aims to minimise both.

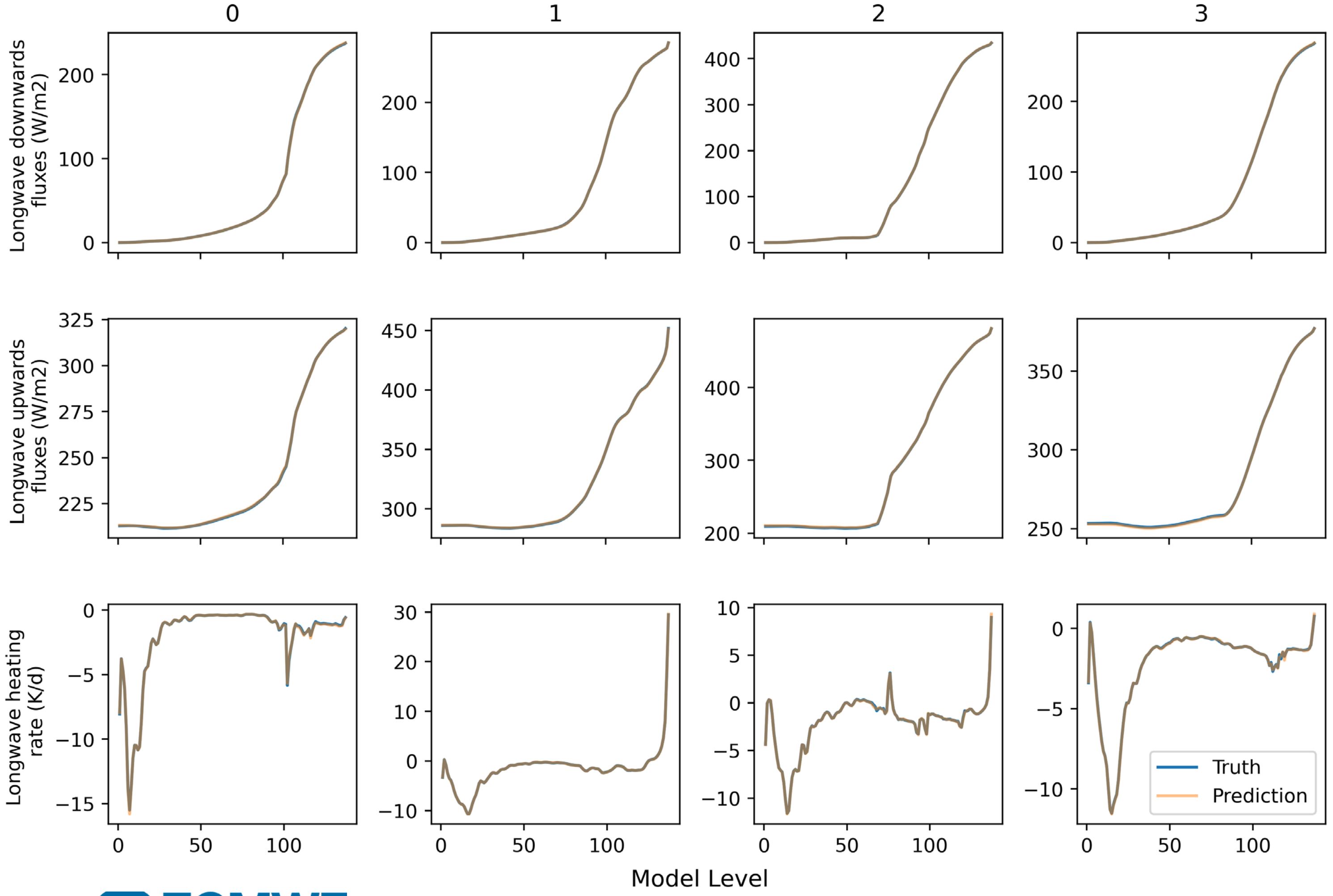
What does it look like? Shortwave

Sample columns, unseen in training.
On average, flux errors ~1W/m^2, heating rate errors ~0.02K/d.
Biases ~0.01W/m^2 & 0.002K/d



What does it look like? Longwave

Sample columns, unseen in training.
On average, flux errors ~0.3W/m^2, heating rate errors ~0.05K/d.
Biases ~0.01W/m^2 & ~0.001K/d



Offline looks nice, but how do we do online?

Infero library - A lower-level API for ML Inference in Operations – Antonino Bonanni (ECMWF)

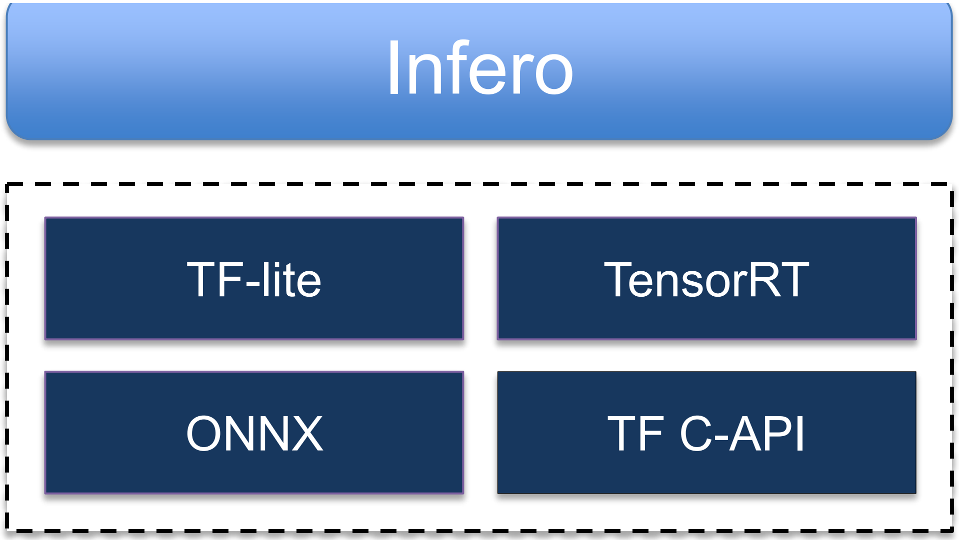
- One Interface, multiple backends
  - TF-lite
  - TensorRT
  - ONNX
  - TF C-API
- Infero provides API's:
  - C, C++, Fortran, Python
- Supports C and Fortran tensor
- Open-Source:
  - github.com/ecmwf-projects/infero

Fortran

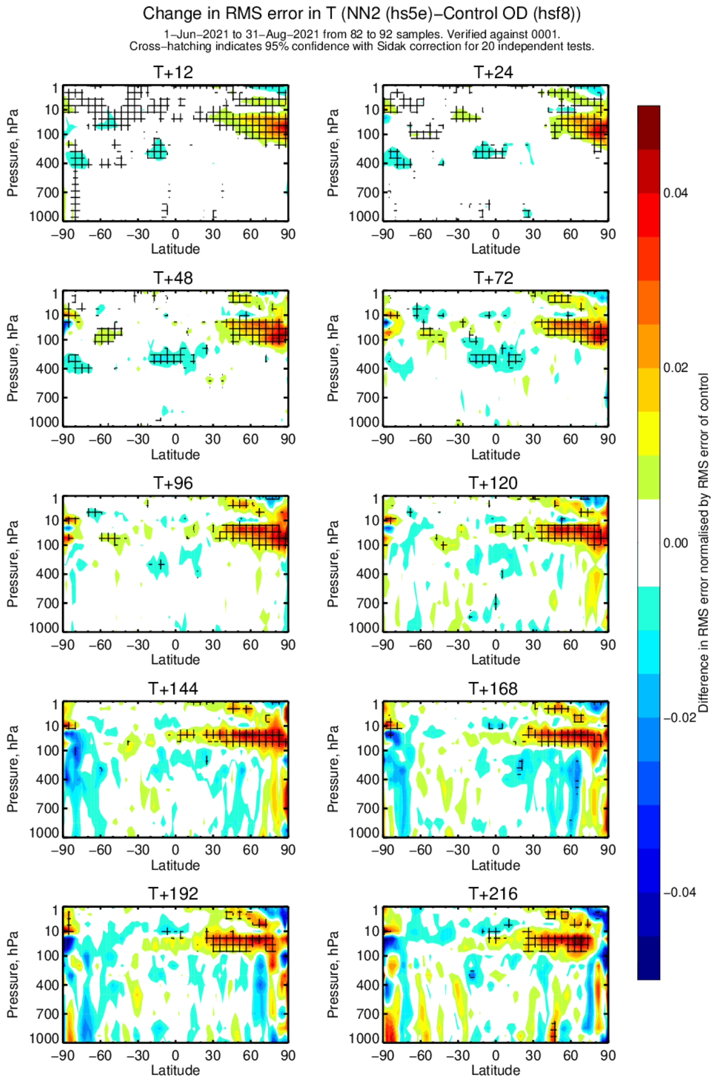
```
model%initialise_from_yaml_file(yaml_path)
model%infer(input_tensor, output_tensor)
```

Python

```
model = pyinfero.Infero(model_path, model_type)
output = model.infer(input_tensor, output_shape)
```



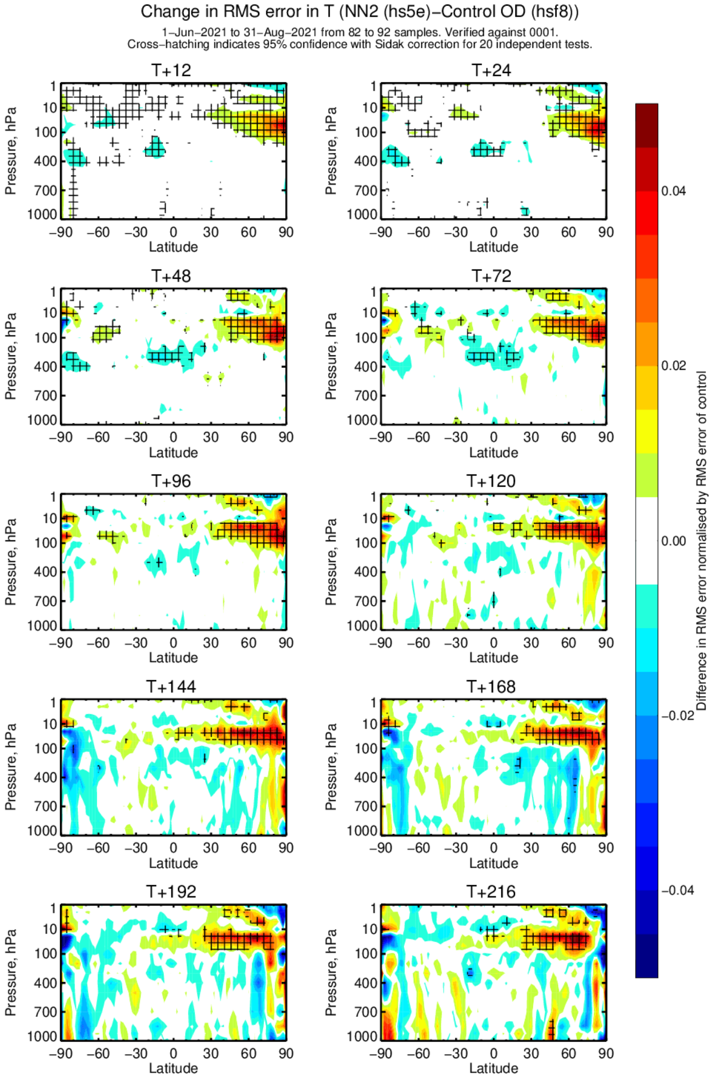
T RMSE NN vs TripleClouds



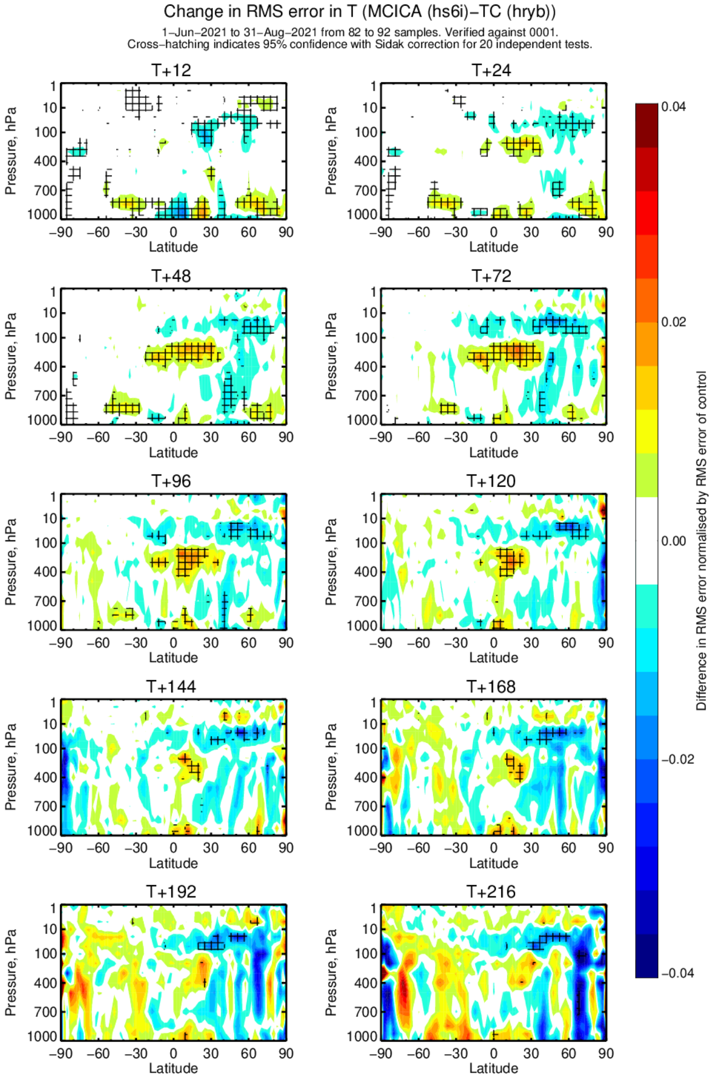
- Suite of JJA TCo399L137 (~30km) forecast experiments.
- Compare with TripleClouds scheme.
- Red degradation, blue improvement (spurious)
- Below 100hPa no strong degradation.
- Evident when only using NN for SW or LW components.
- Strong results, but room for further improvement in NN.

T RMSE

NN vs TripleClouds

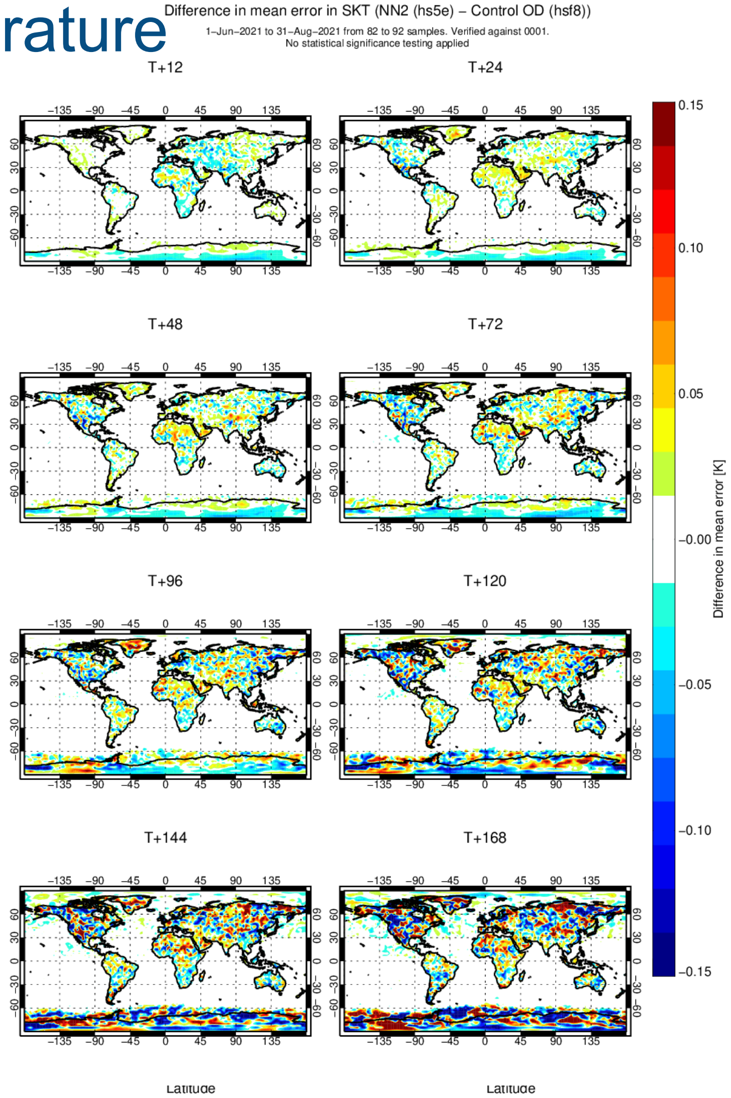


McICA vs TripleClouds



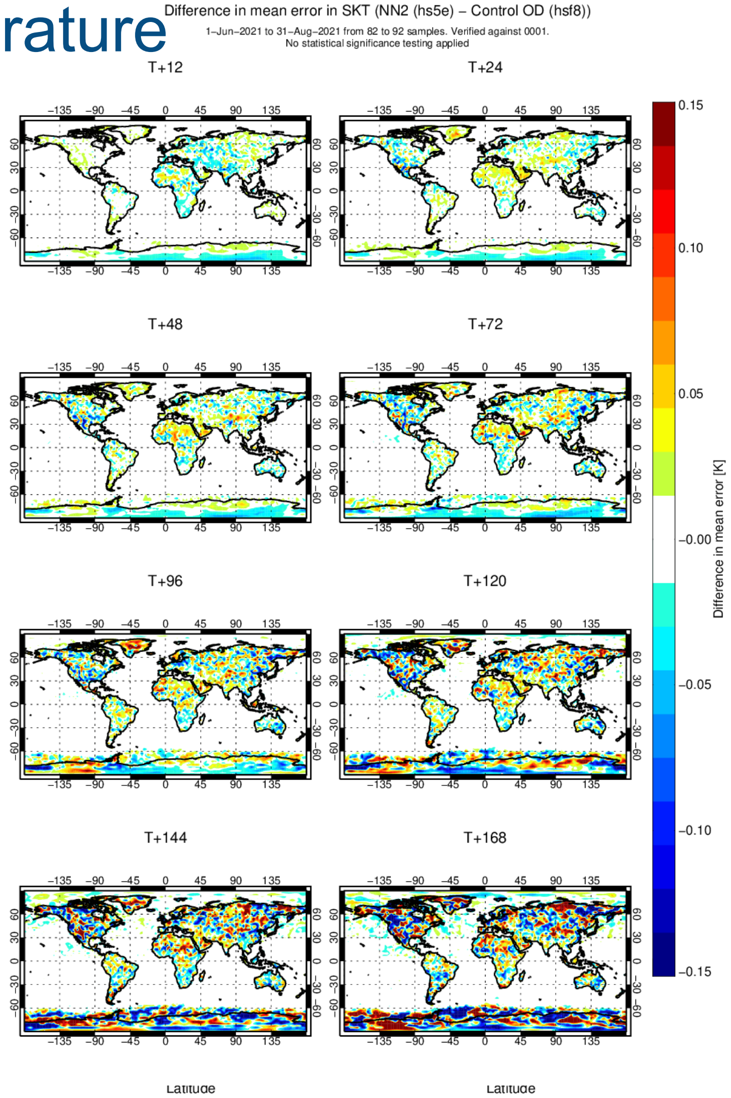
Mean skin temperature error

NN vs TripleClouds

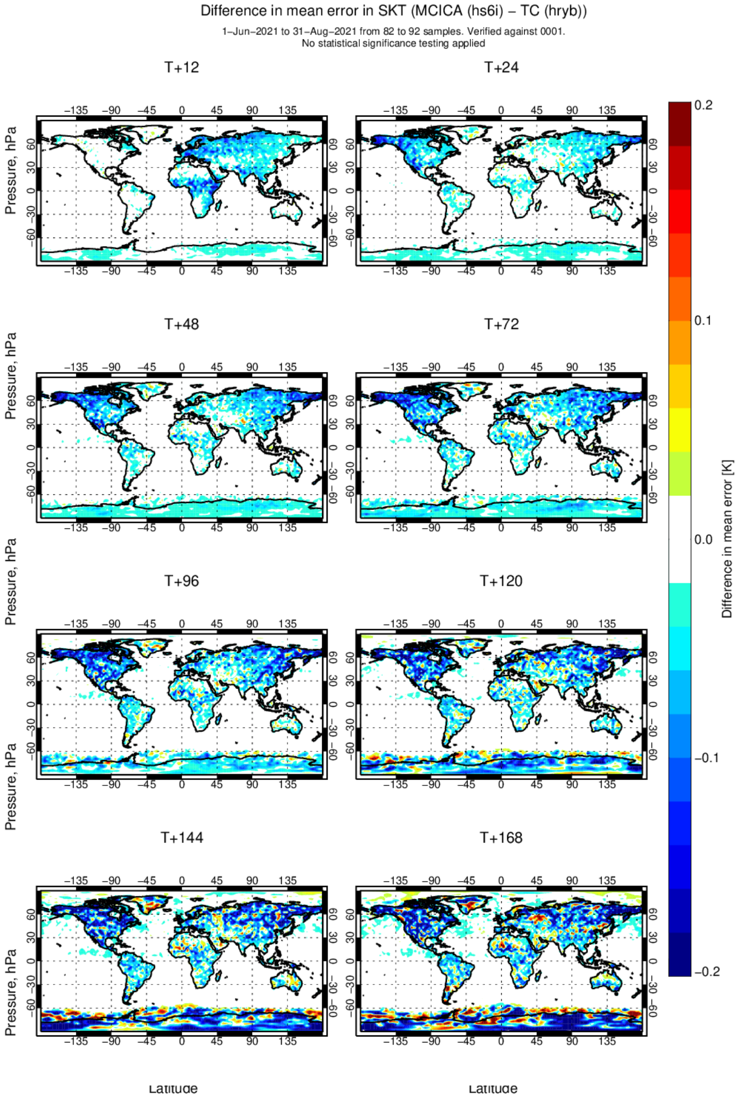


Mean skin temperature error

NN vs TripleClouds



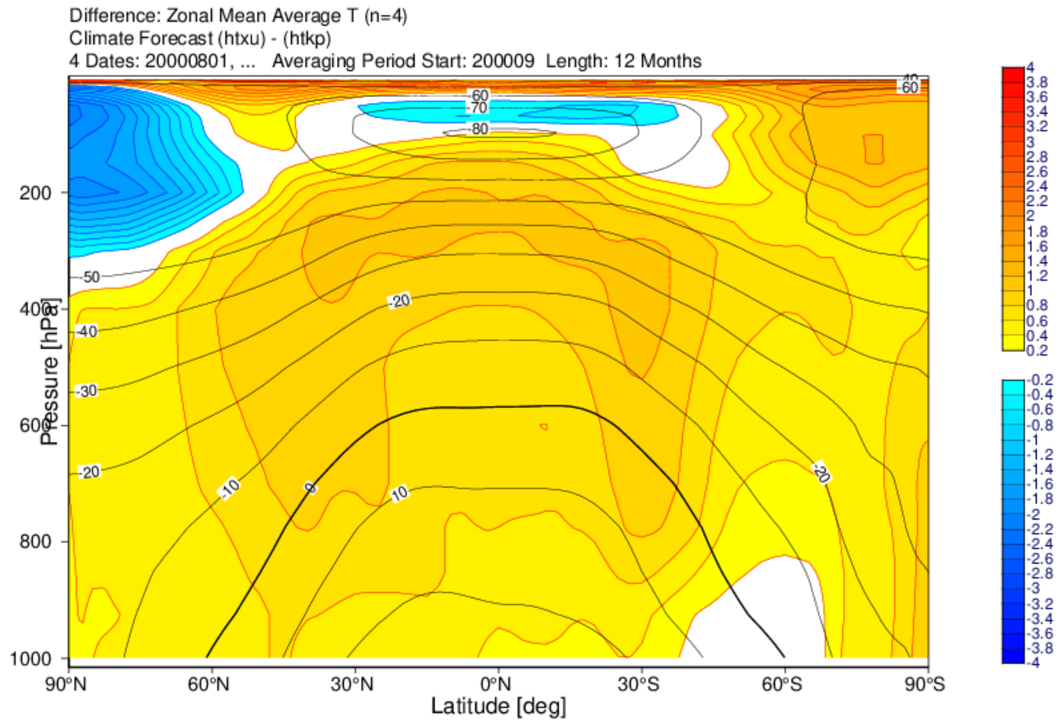
McICA vs TripleClouds



What happens to the model climate?

Climate experiment

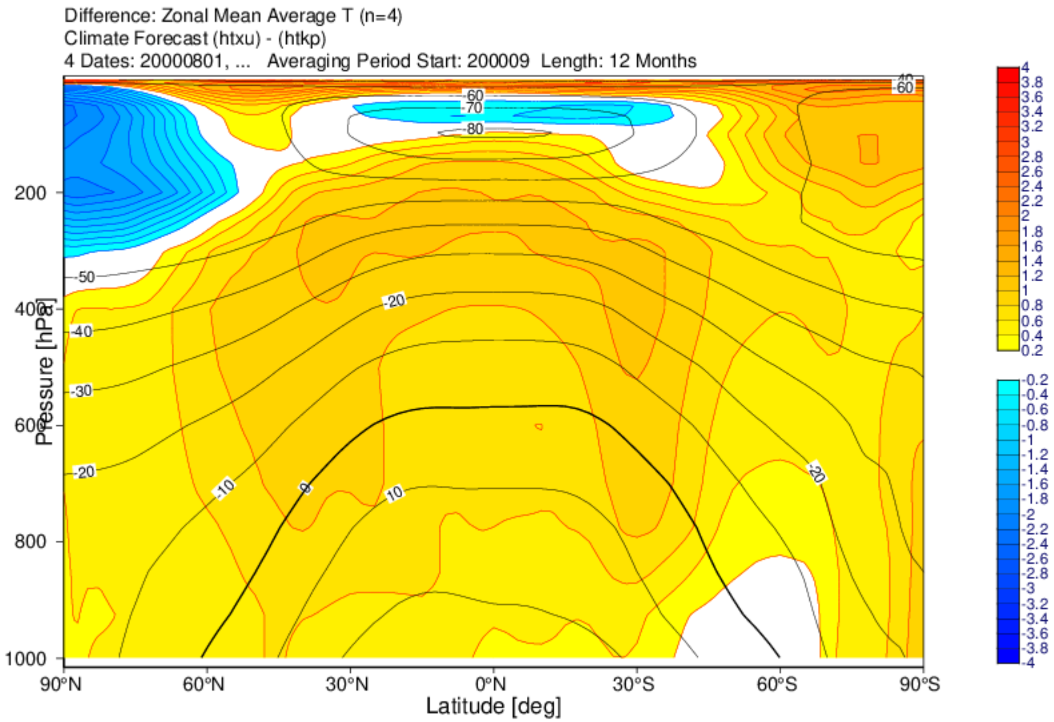
- 2000-2004, TCo199L137, 4x1 year simulations
- Diff zonal mean temperature vs TripleClouds



- Uh oh, what's different from the medium range runs?

Climate experiment

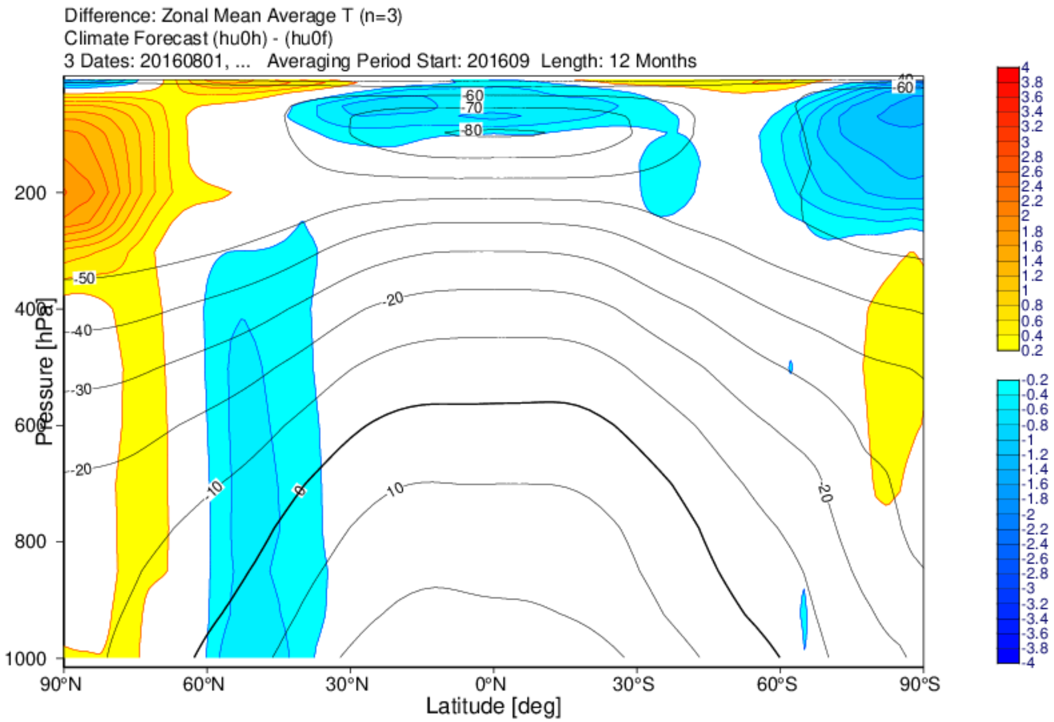
- 2000-2004, TCo199L137, 4x1 year simulations
- Diff zonal mean temperature vs TripleClouds



- Uh oh, what's different from the medium range runs? **CO2**
- Training only used 2020 data, and testing on 2019 & 2021.

Climate experiment v2!

- 2016-2018, TCo199L137, 3x1 year simulations
- Diff zonal mean temperature vs TripleClouds



- Much improved (although not yet perfect).
- Next step, train on a wider range of climates, make networks climate robust.

What about speed?

CPU – marginally faster

GPU – ongoing work, results coming soon

Summary

- Accurate emulators for complex physical processes can be trained.
  - Incorporate structure of existing solver to avoid spurious causations.
- Moving towards neutrality, but still some work left.
  - Recent improvements in NN offline scores, another cycle of online testing imminent.
- Climate robustness will require larger training set.
  - Easy to do with existing frameworks for data generation and
- Remains to be seen what computational advantages we get on GPU.
  - Cost of TripleClouds is a moving target, Robin (and collaborators) regularly making significant cost reductions in physical scheme. (Man vs Machine)
  - NOGWD results showed that cost/accuracy can be traded, further work need to find best compromise for radiation.
- Computationally neutral results bode well for learning more complex schemes (e.g. SPARTACUS).
- Interesting challenges **if** work moves towards operations.
  - Monitoring drift/errors, retraining for cycle upgrades?

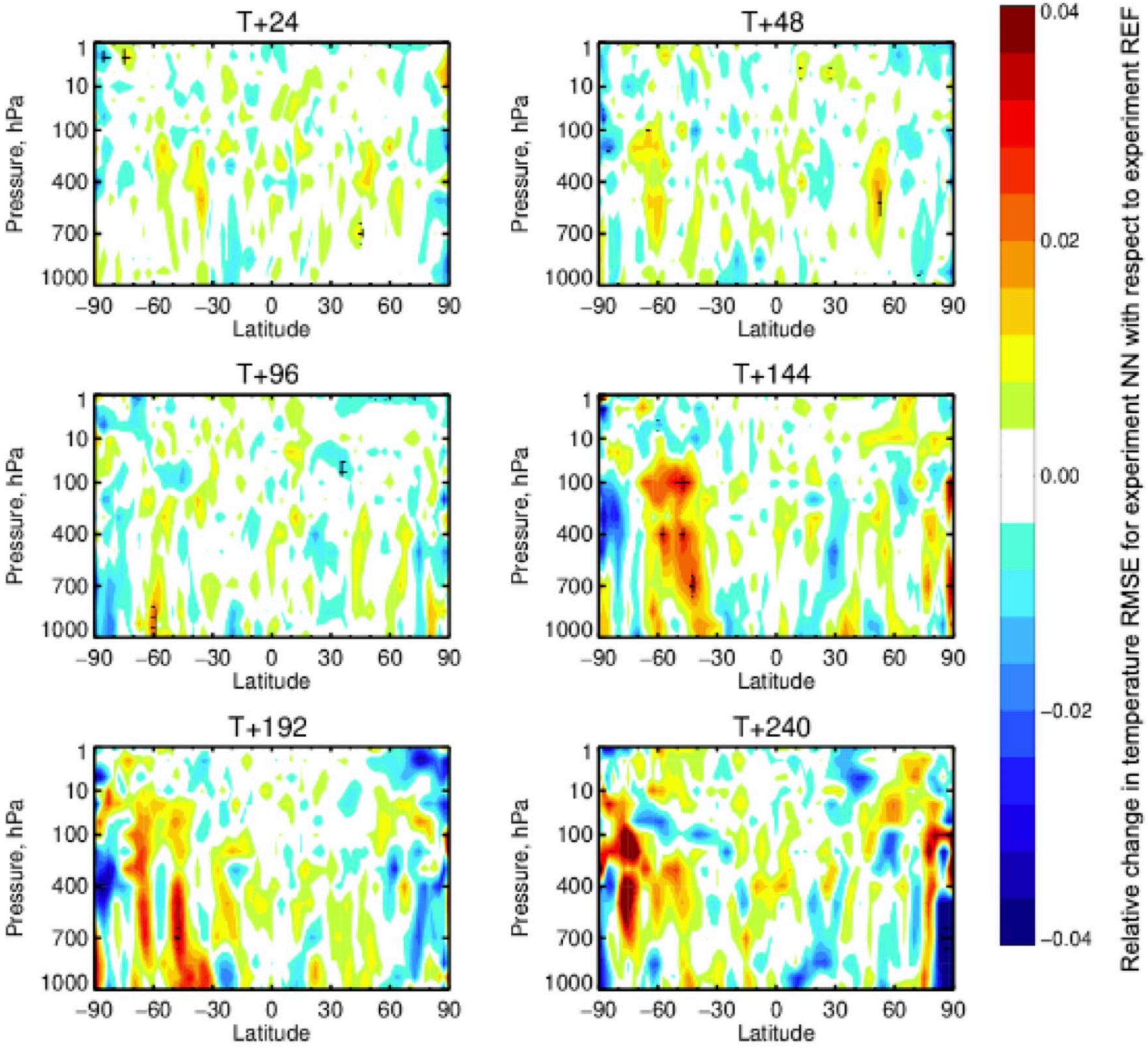
One more thing…

Emulators for variational data assimilation

- Hatfield et al. (2021) used the NOGWD neural network to derive tangent-linear and adjoint models of NOGWD process.
- TL & Ad models accurately satisfied the adjoint test.

$$\left[\mathbf{M}(x_0)\delta x\right]^{\mathrm{T}}\delta y = \delta x^{\mathrm{T}}\left[\mathbf{M}^{\mathrm{T}}(x_0)\delta y\right]$$

- Experiments using NN TL/Ad showed no change in forecast error.
- Upcoming work will apply this idea to radiative heating, where the existing TL/Ad codes are based on older version of the nonlinear scheme.
- No guarantees that learning nonlinear scheme will provide accurate enough gradient information.



That’s all folks…

…any questions?