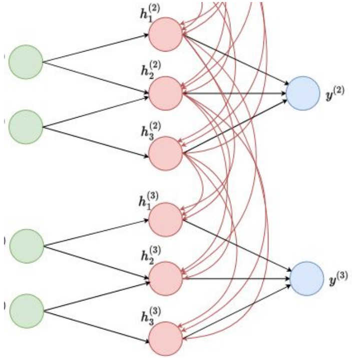
Radiative Transfer Emulation: Results So Far (and why we should move on to 3D)

MAELSTROM dissemination workshop 2023

Peter Ukkonen
Danish Meteorological Institute
peterukk@gmail.com

Danmarks
Meteorologiske
Institut

With help from Matthew Chantry (ECMWF)



Outline

1. Introduction
2. A recurrent neural network approach for emulating radiative transfer
3. Optimizing physical radiation schemes
4. Performance: state-of-the-art physical scheme on CPU vs ML on GPU
5. 3D considerations

Radiation in Earth system models: the art of approximation

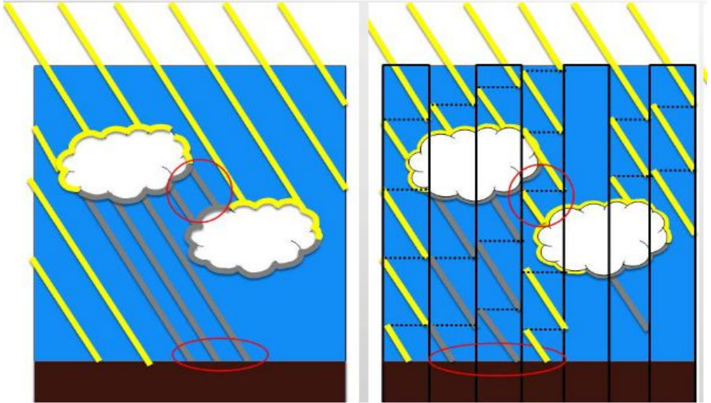
Maxwell’s equations in terms of fields **E**(**x**,t), **B**(**x**,t)

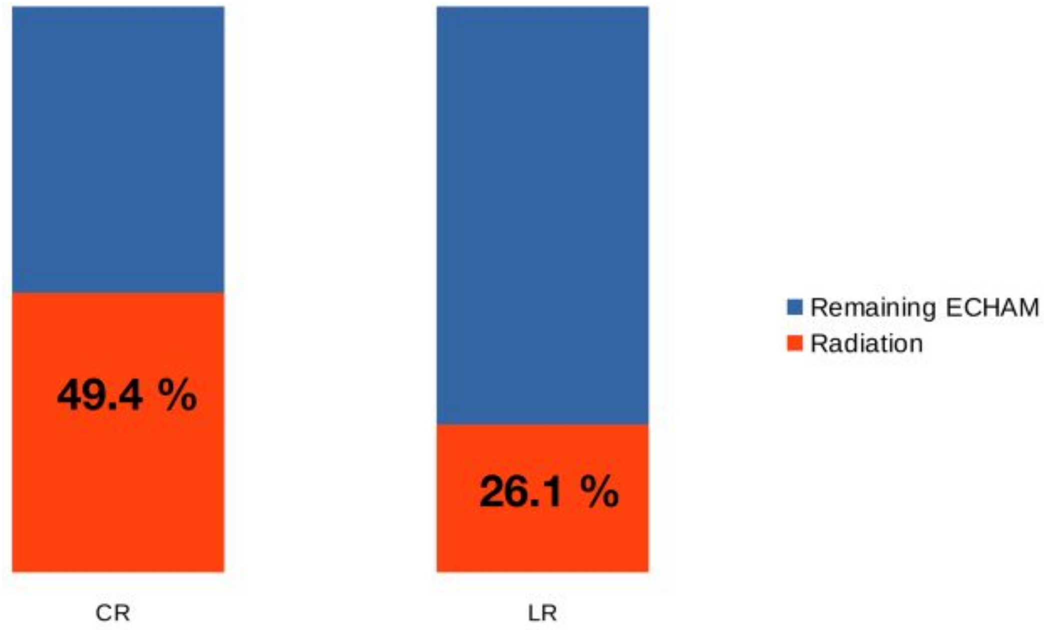
3D radiative transfer in terms of monochromatic radiances I (**x**,Ω,ν)

Weather and climate models: 1D radiative transfer in terms of two monochromatic fluxes F↓(z, ν), F↑(z, ν)

Radiative transfer in the atmosphere is **well-understood** but **approximated due to computational constraints**:

- group together spectral frequencies
- atmosphere is horizontally homogenous within a grid column (“plane-parallel”)
- consider radiation only in two directions, up and down (“two-stream”)





Coarse-res
(~460 km)

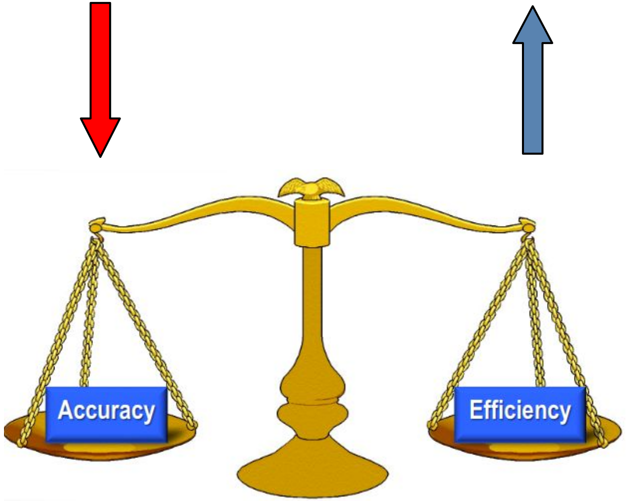
Low-res
(~170 km)

Radiation has historically been a very expensive component of Earth system models, which has been the motivation for emulation studies

However, as model resolution increases radiation can be called more infrequently (relative to model time step). IFS computes radiation every hour on a coarser grid: only a few % of runtime

Nonetheless, efficient computations would allow fewer approximations / more realism (e.g. more streams than 2!)

Cotronei & Slavig, GMD 2020: Single-precision arithmetic in ECHAM radiation reduces runtime and energy consumption.



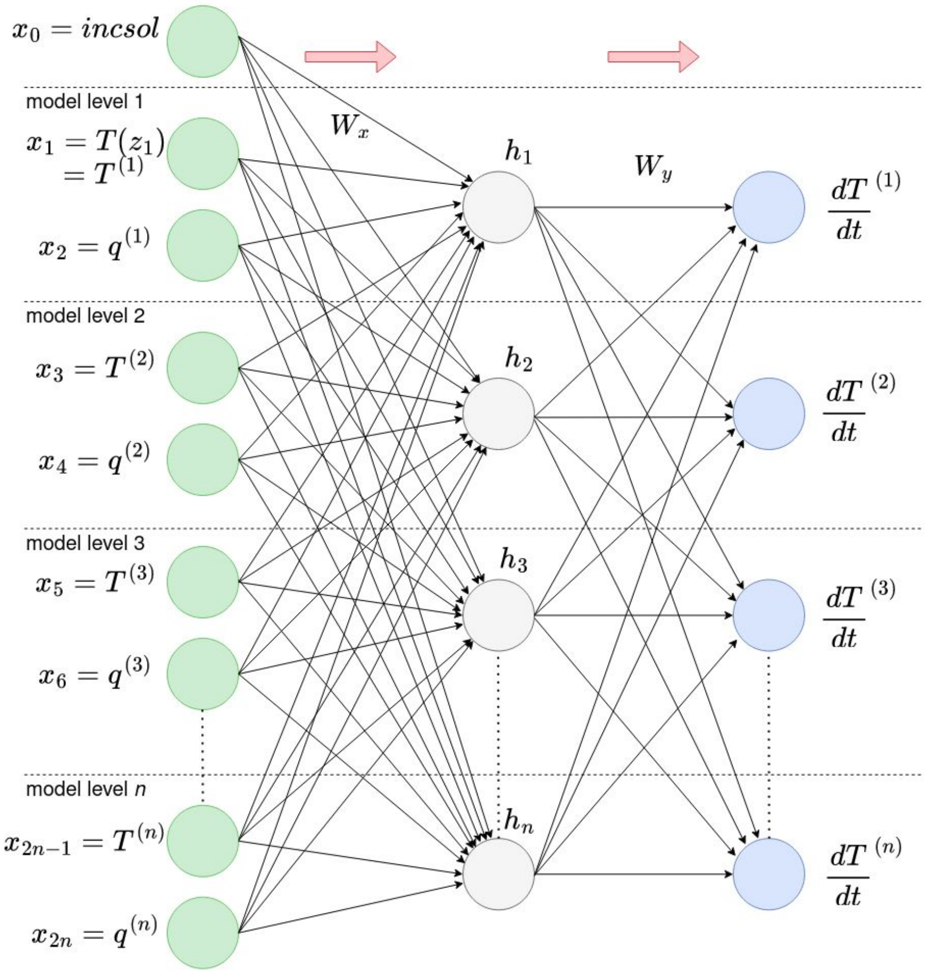
Key question: can we “improve” the accuracy/efficiency trade-off of radiation schemes by using ML?

- Longstanding (> 20 years) efforts based on dense neural nets have given large speed-ups, but sacrificed accuracy and generalization
- Some studies have used outdated and slow radiation codes as reference

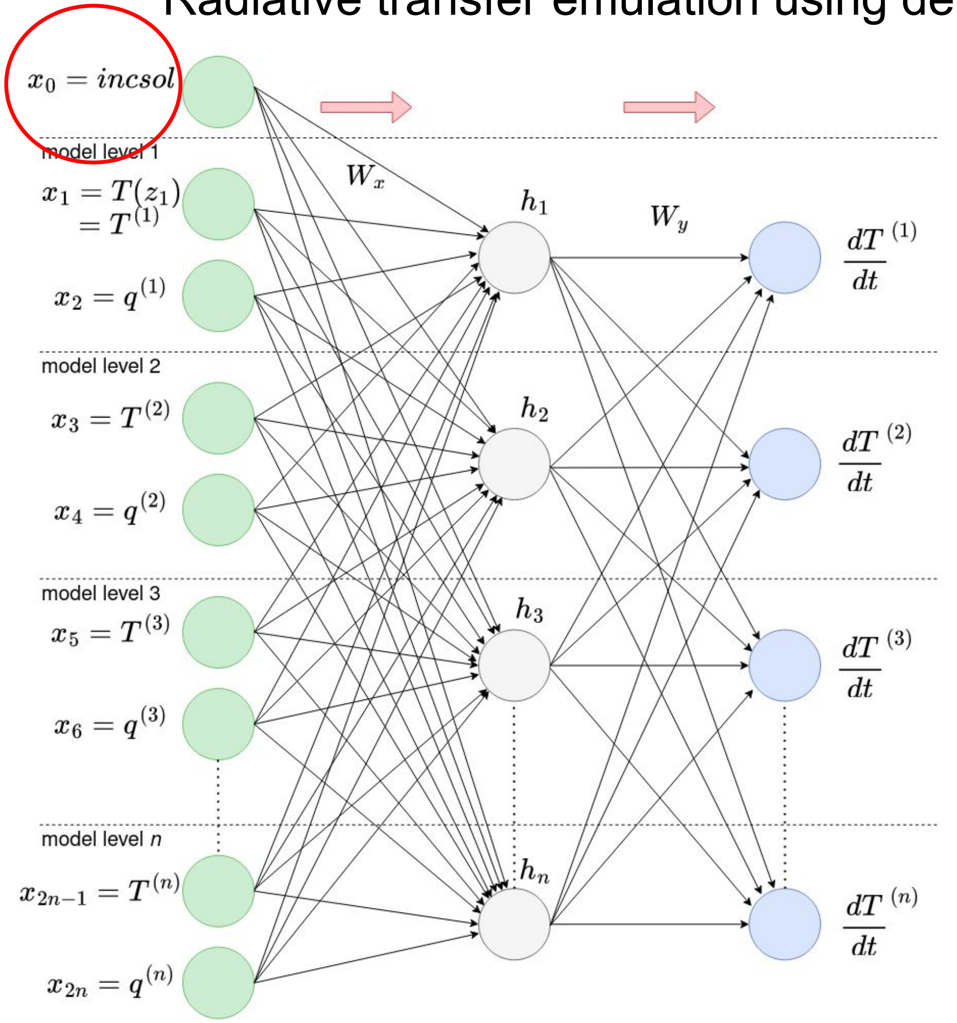
Even if we don’t achieve this goal, radiation provides a useful testbed for physics emulation:

- Well-understood, benchmark solutions exist
- Easy-to-use stand-alone radiation codes: ESM simulations not always necessary
- Non-linear (like other sub-grid physics)
- **Non-local (assumed 1D, like other sub-grid physics)**

Radiative transfer emulation using dense networks (MLPs)



Radiative transfer emulation using dense networks (MLPs)



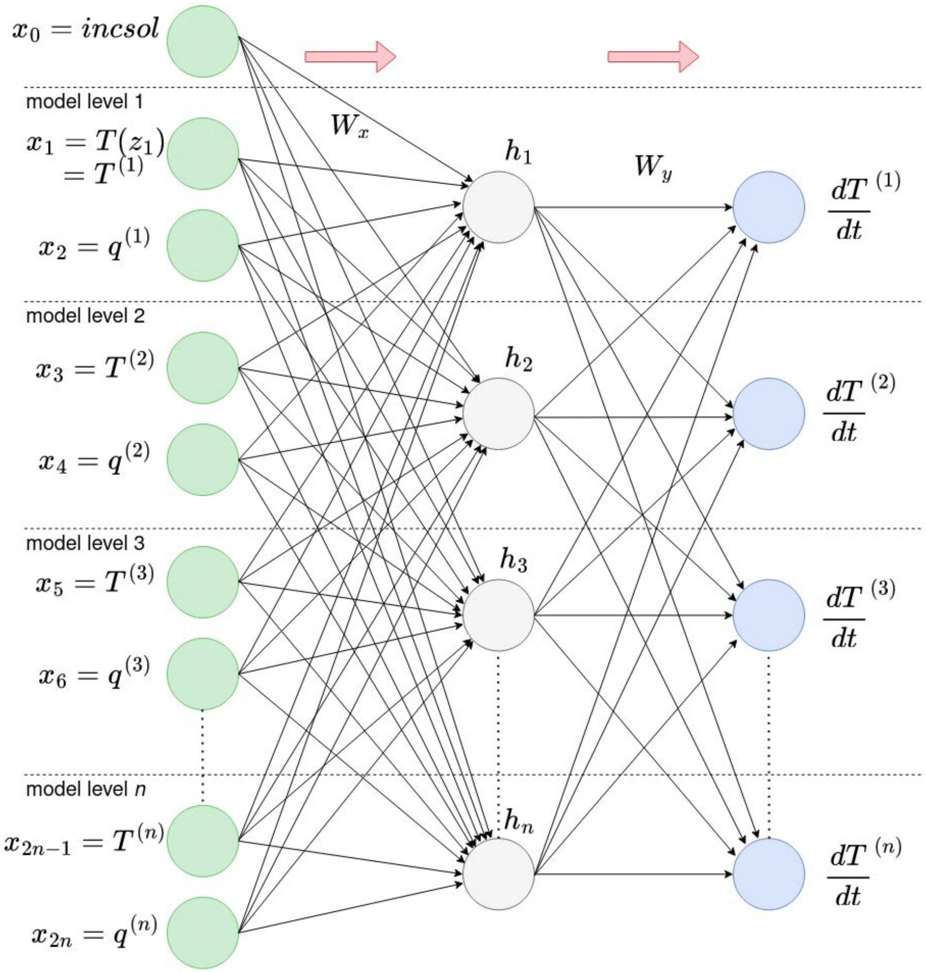
Inputs are **vertical profiles** of pressure, temperature, gases, cloud water and ice, and a few scalar variables such as incident solar radiation (shortwave only)

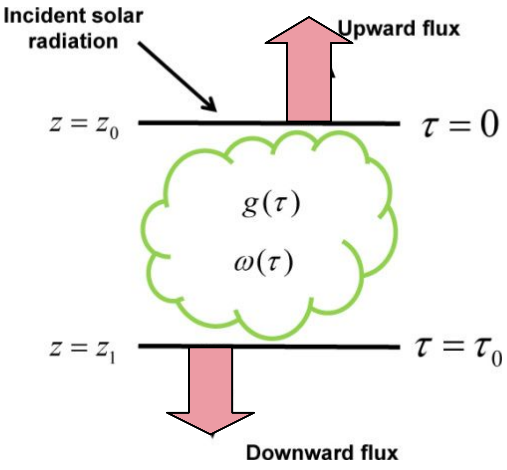
Outputs are **vertical profiles of heating rates (HR) = dT/dt**

Radiation codes compute HR from upward and downward **fluxes,** but this approach gives noisy heating rates with dense NNs, so typically the outputs are HR profile + surface and top-of-atmosphere flux

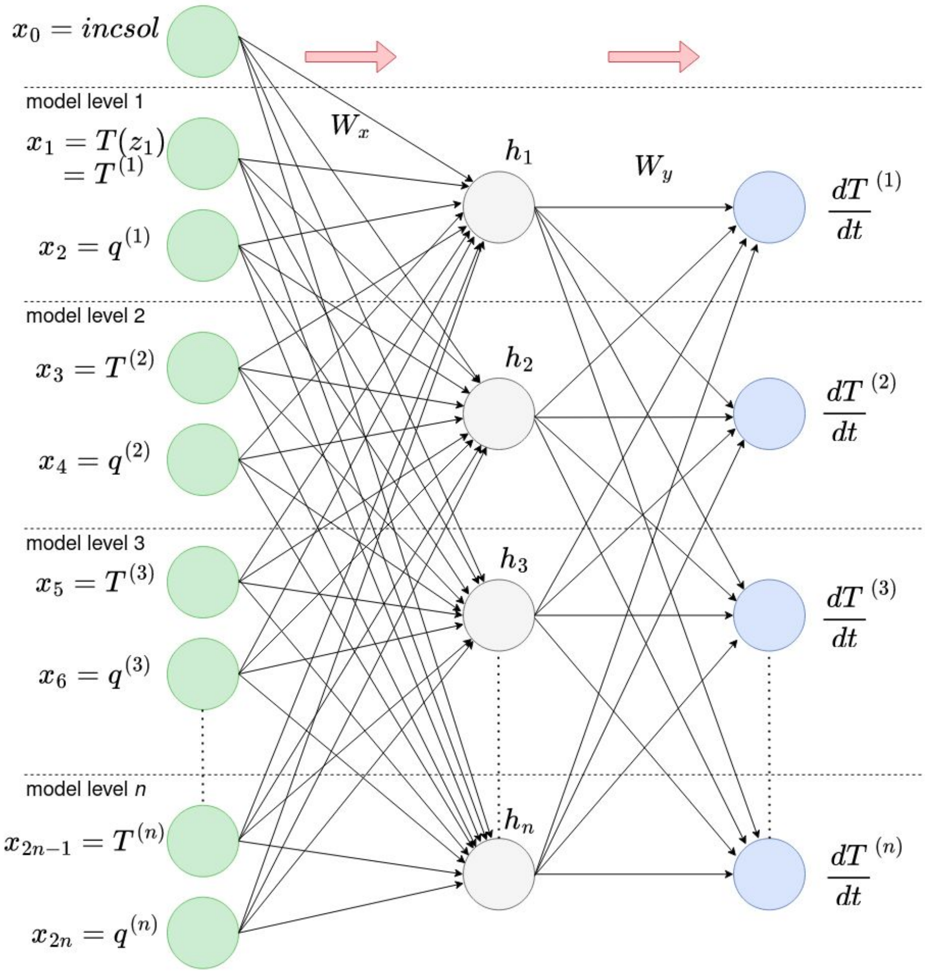
→ less noisy HRs **but breaks energy conservation**

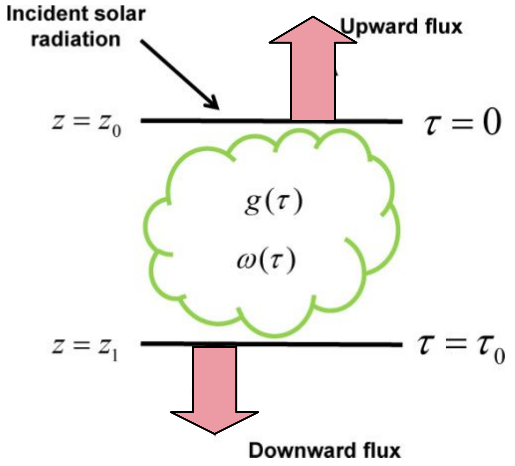
Radiative transfer emulation using dense networks (MLPs)





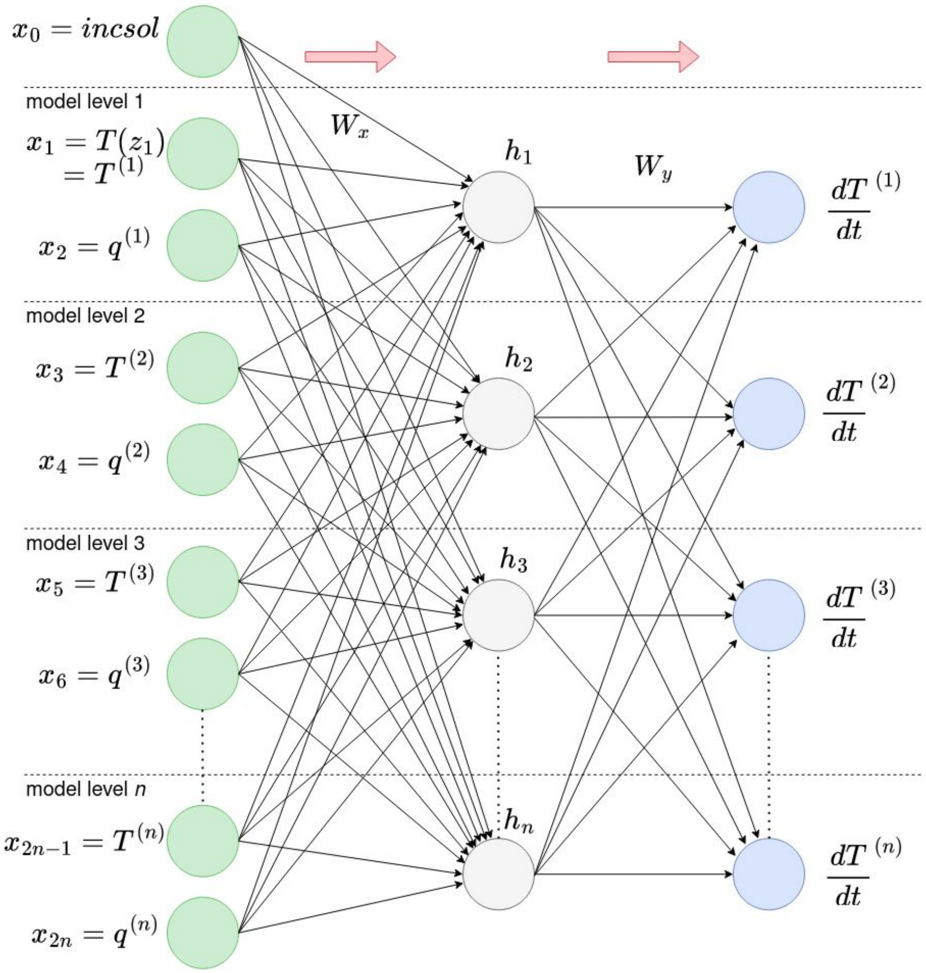
Radiative transfer emulation using dense networks (MLPs)

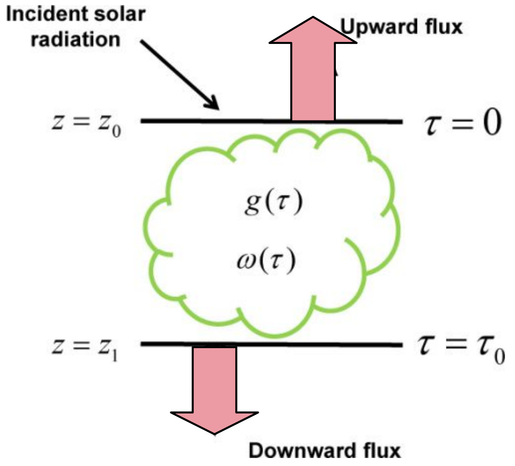




Comparing the two figures, what problem does the model have?

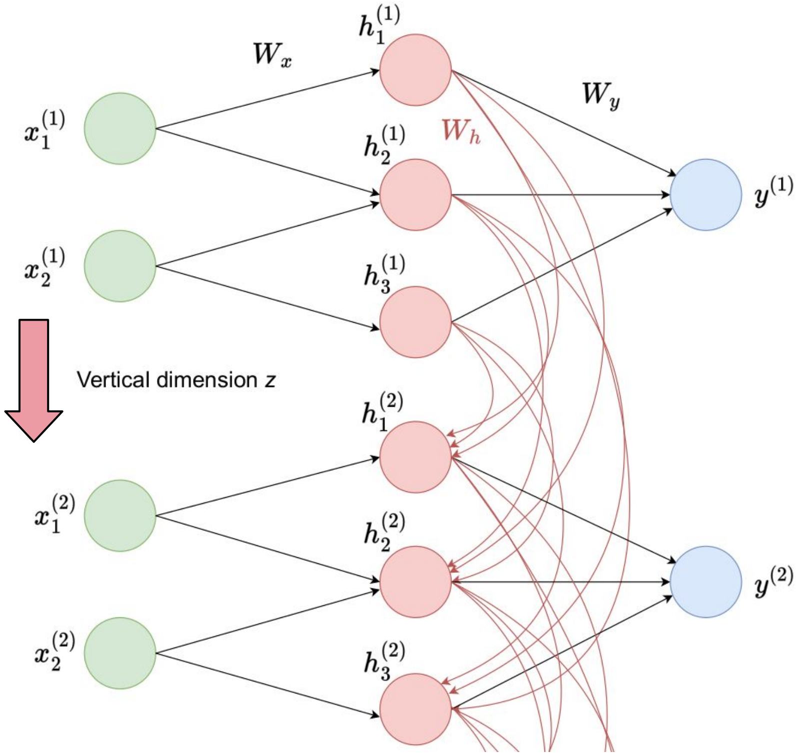
Radiative transfer emulation using dense networks (MLPs)

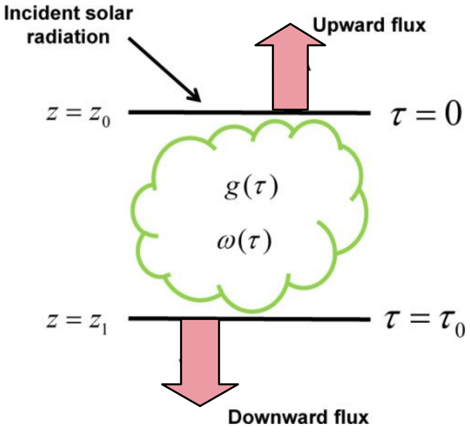




- **Mismatch in the direction of information flow between the model and the process!**
- Dense nets **lack the connections to directly propagate information up/down the column** (non-locality)

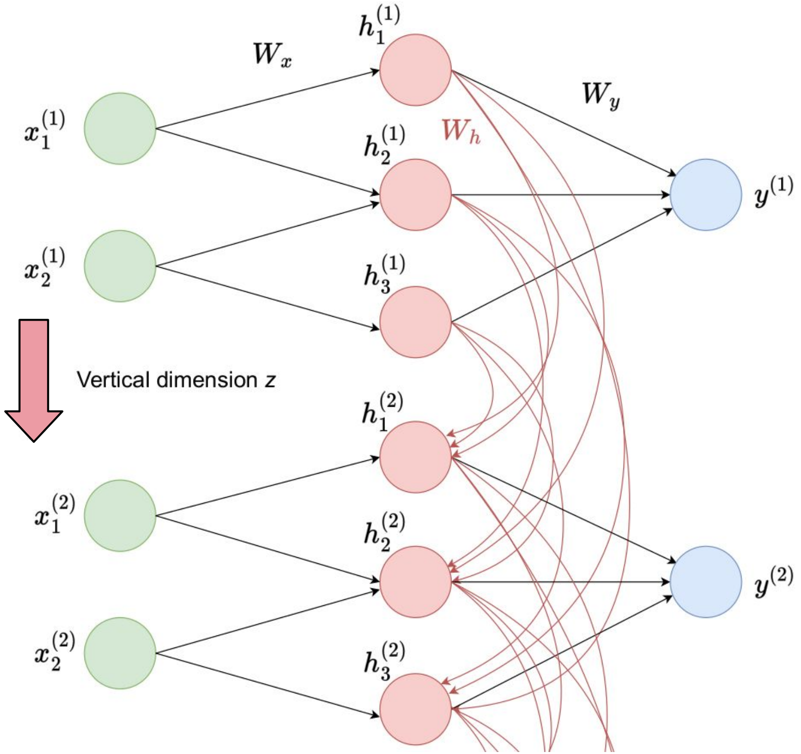
Recurrent neural networks for radiation (the solution?)

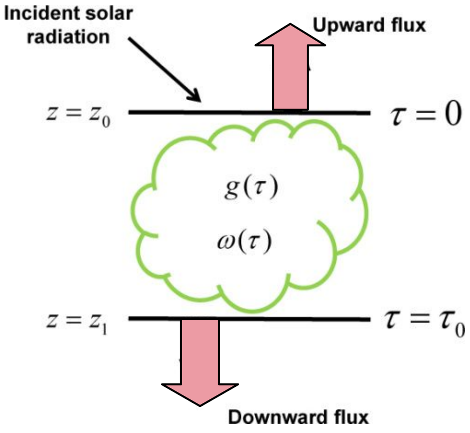




RNNs are usually applied to **temporal sequences**, here they iterate through a **vertical column**, updating a *hidden memory* when processing each vertical level

Recurrent neural networks for radiation (the solution?)



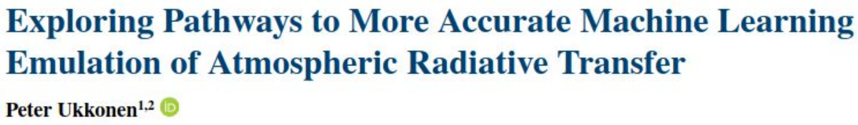


RNNs are usually applied to **temporal sequences**, here they iterate through a **vertical column**, updating a *hidden memory* when processing each vertical level

Characteristics of atmospheric radiative transfer respected by RNNs:

- **Correct directionality**, however radiation flows both upward and downward, so we need **bidirectional RNNs** (BiRNN)
- **Sequential** from one level to the next – unlike DNN, which (unphysically) connects the top directly to the surface
- **Invariable** physical laws by height – unlike DNN, which (unphysically) uses level-specific weights

RNNs for shortwave radiation (JAMES 2022)



Exploring Pathways to More Accurate Machine Learning Emulation of Atmospheric Radiative Transfer

Peter Ukkonen[1,2]

- Compare different ways of emulating the RTE+RRTMGP scheme

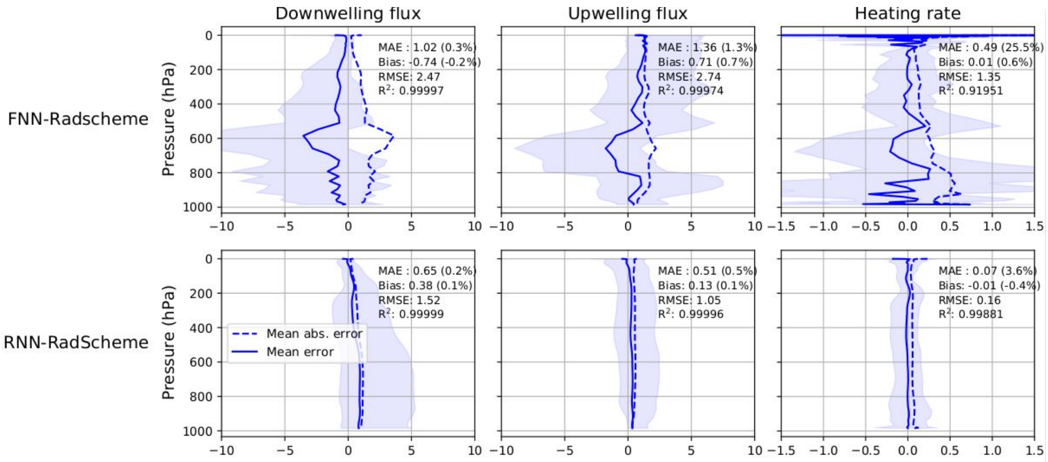
- To ensure energy conservation, the NNs predict **full flux profiles (like physical radiation schemes)**, from which heating rate (HR) is physically derived:

$$\text{HR} = \left(\frac{dT}{dt}\right)_{\text{SW radiation}} = -\frac{g}{c_p}\frac{F_{\mathrm{i}+1/2,\,\mathrm{SW}} - F_{\mathrm{i}-1/2,\,\mathrm{SW}}}{p_{\mathrm{i}+1/2} - p_{\mathrm{i}-1/2}},$$

- A hybrid loss function to reduce HR errors: $loss = \alpha(y - y_{pred})^2 + (1-\alpha)(HR - HR_{pred})^2,$

- For shortwave it's helpful to **normalize all the fluxes by the incoming solar radiation**

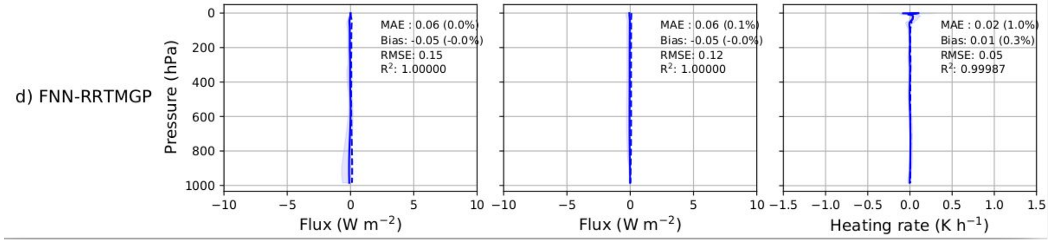
RNNs for shortwave radiation (JAMES 2022)



2-layer deep MLP, 128+128 neurons:
100,000 parameters
RMSE **1.35 K / day**
~50x speedup

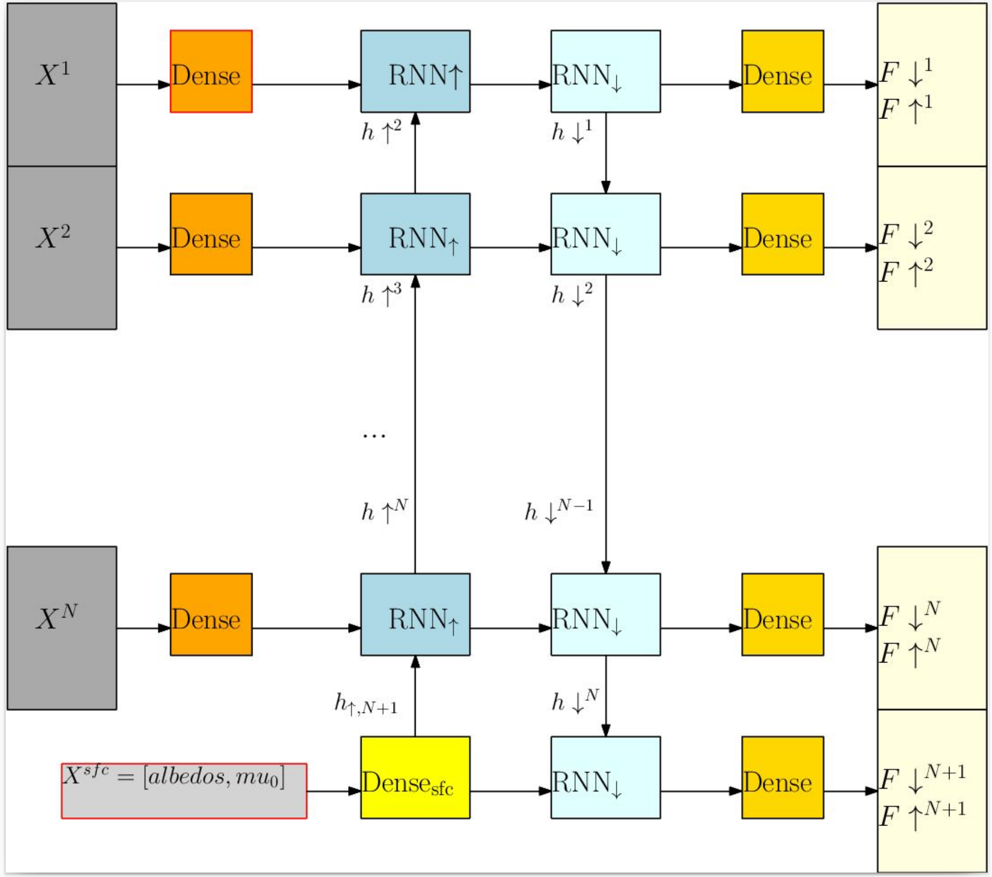
Bidirectional GRU, 16+16 neurons:
5600 parameters
RMSE **0.16 K / day**
~5x speedup

! speed-ups are on CPU and relative to a modern but somewhat expensive radiation scheme with high spectral resolution (RTE+RRTMGP, 224 SW g-points).
*On GPU, RNN was **~7x** slower than MLP*



MLPs for predicting optical properties:
RMSE 0.05 K / day
4200 parameters
~1.3x speedup (but better generalization, interpretability and flexibility)

Predicting *nlay+1* up-and-downwelling fluxes from layer-wise inputs using bidirectional RNNs



We can emulate a two-stream radiation scheme using a bidirectional RNN that maps ***(nx, nlay)* inputs** to ***(2, nlay+1)* outputs** , i.e. down- and upwelling fluxes at layer interfaces

Surface information can be used to initialize the first (upward) RNN, and concatenated with its *nlay* outputs to get a sequence length of *nlay+1.* Mimics the computation of albedo in physical radiation code

The second RNN starts at top-of-atmosphere and iterates downwards computing the fluxes, just like in ecRad

Prognostic evaluation of radiation emulators in the IFS

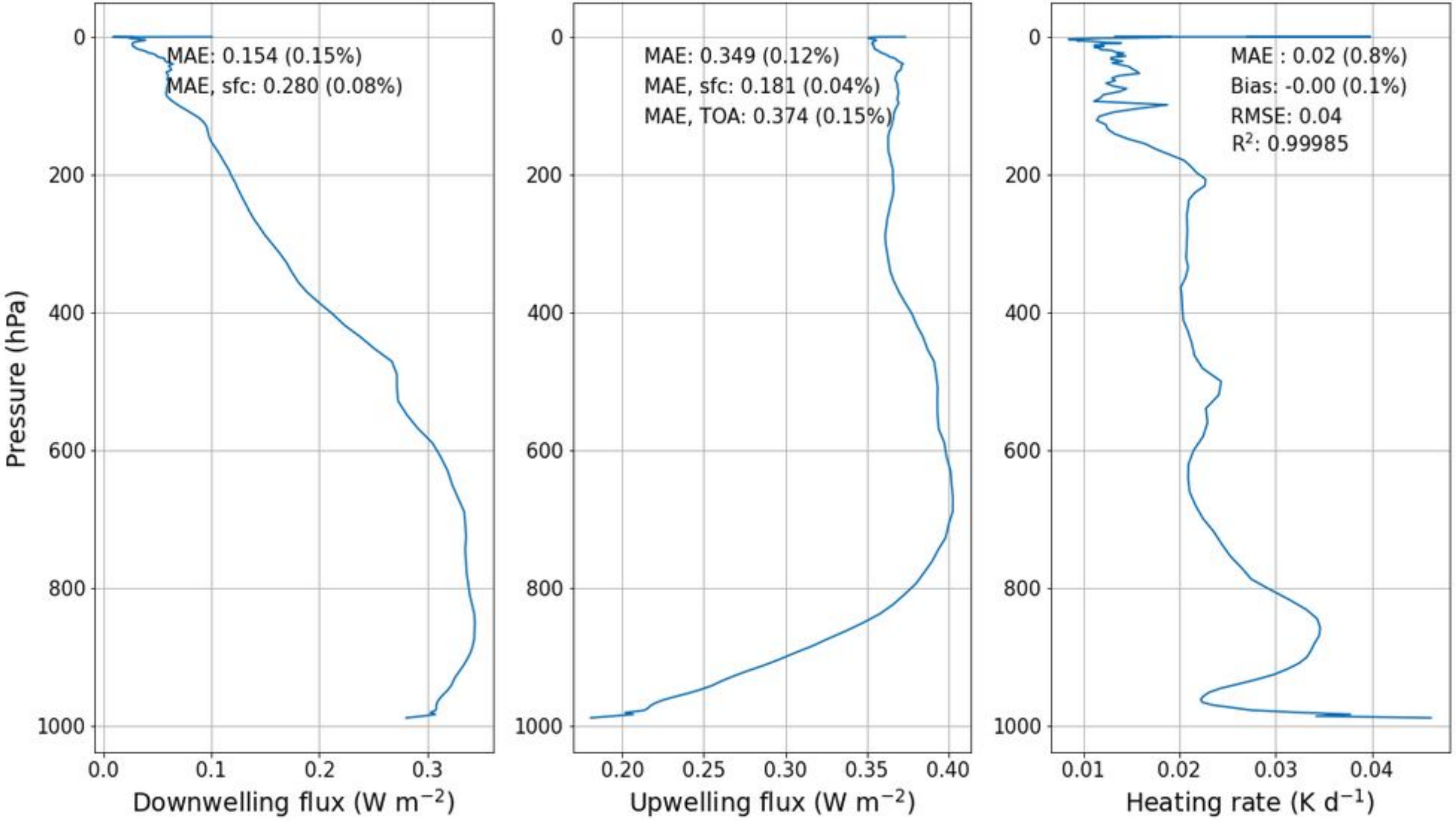
- RNNs were trained on the inputs and outputs of ecRad (TripleClouds solver) using a hybrid loss incorporating heating rate.
- Training - 2020, Evaluation – 2021
- IFS implementation / online inference by using Infero, a lower-level ML library developed at ECMWF that supports different back-ends

github.com/ecmwf-projects/infero

Work mainly by Matthew Chantry (ECMWF)

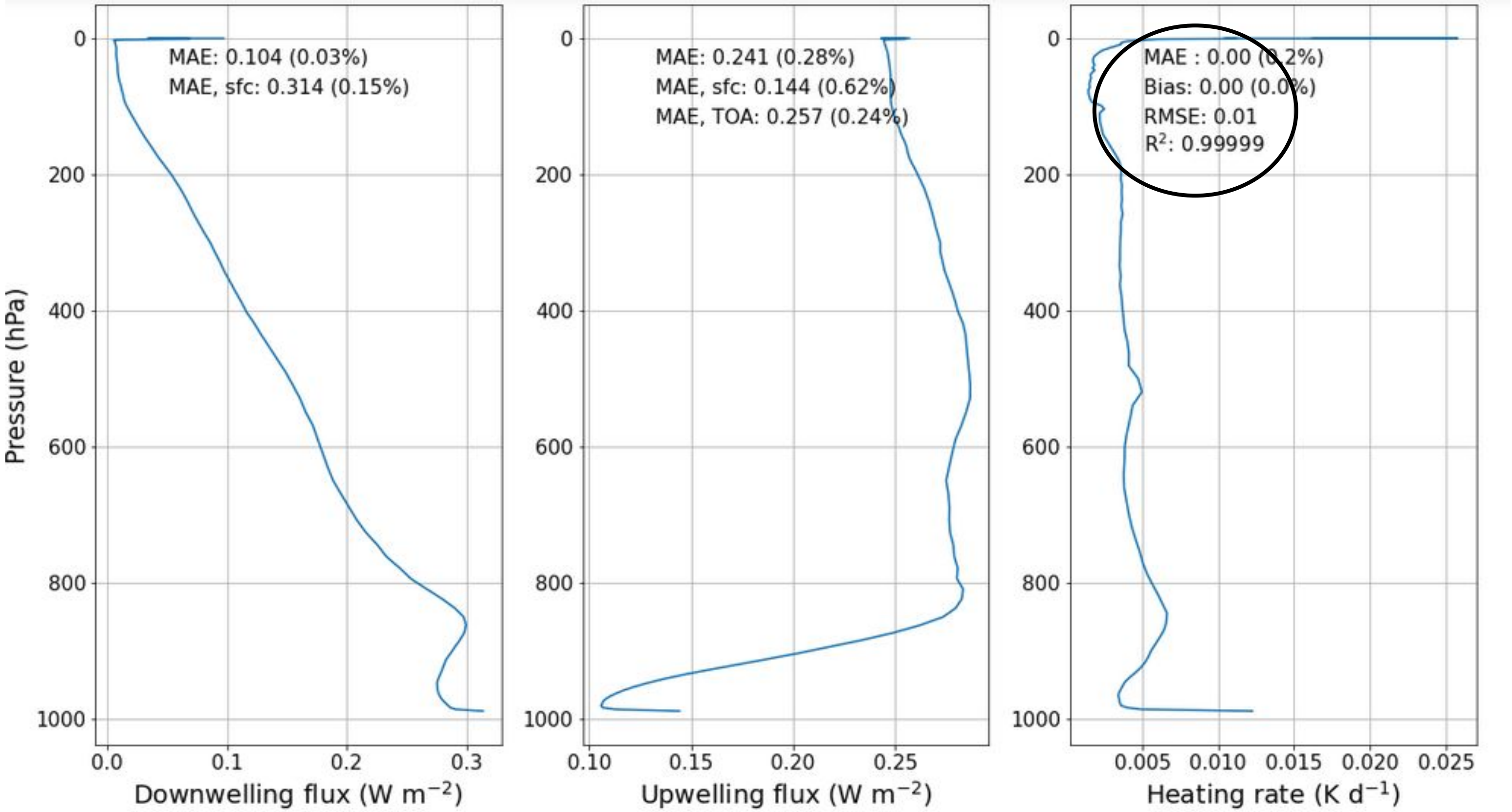
RNNs emulating ecRad, tested in the IFS

Offline errors, 64-neuron **longwave** LSTM (~60k parameters)



RNNs emulating ecRad, tested in the IFS

Offline errors, 64-neuron **shortwave** LSTM (~60k parameters)

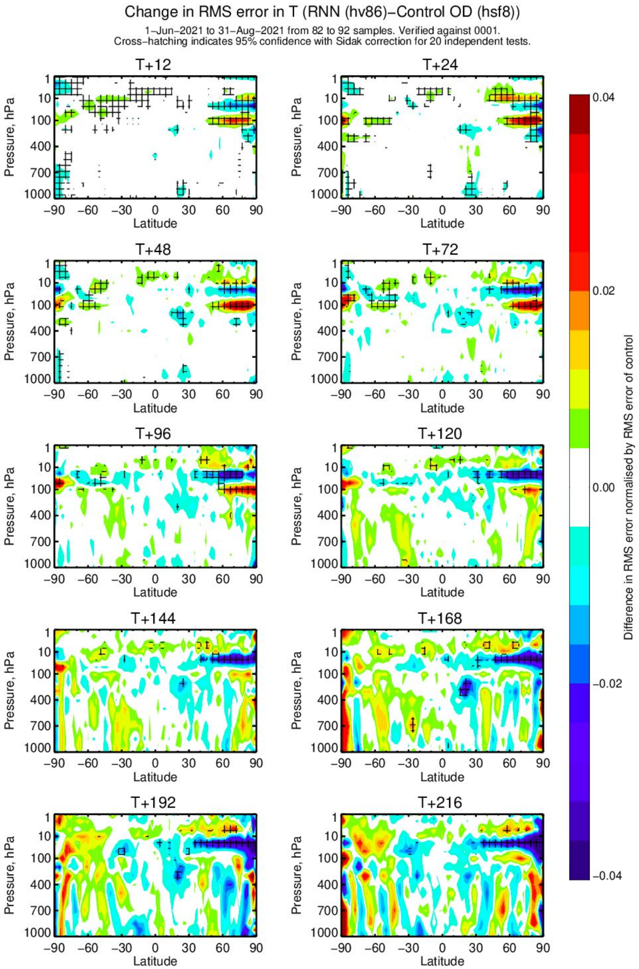


RNN vs TripleClouds

Depicted: change in temperature RMSE using a suite of JJA IFS experiments at ~30km resolution

Red = degradation

Blue = improvement



Change in RMS error in T (RNN (hv86)–Control OD (hsf8))

1–Jun–2021 to 31–Aug–2021 from 82 to 92 samples. Verified against 0001.
Cross–hatching indicates 95% confidence with Sidak correction for 20 independent tests.

Depicted: change in temperature RMSE using a suite of JJA IFS experiments at ~30km resolution

Red = degradation

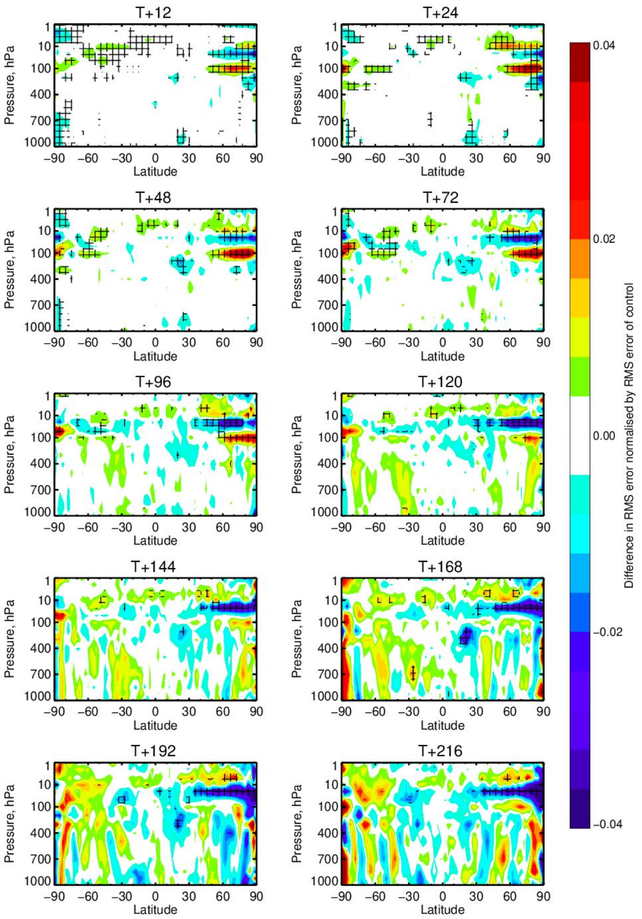
Blue = improvement

Using emulators has a statistically small impact on forecasts: **similar to changing the radiative transfer solver to McICA** (which is similar to TripleClouds but treats sub-grid cloud variability stochastically)

RNN vs TripleClouds

Change in RMS error in T (RNN (hv86)–Control OD (hsf8))

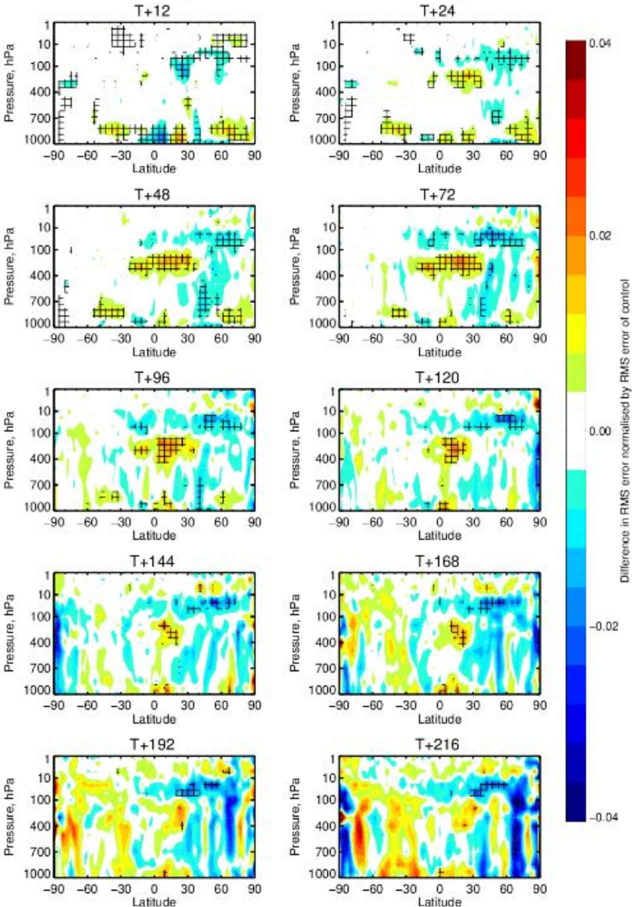
1–Jun–2021 to 31–Aug–2021 from 82 to 92 samples. Verified against 0001.
Cross–hatching indicates 95% confidence with Sidak correction for 20 independent tests.

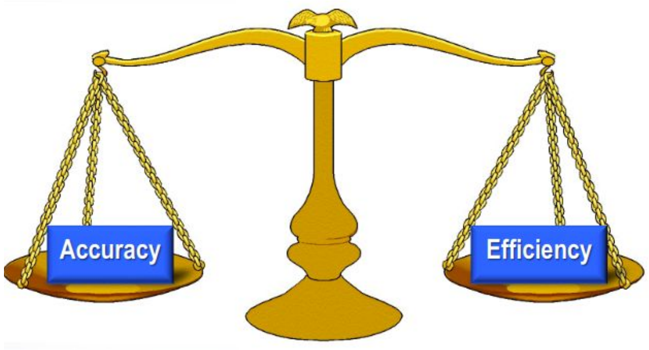


McICA vs TripleClouds

Change in RMS error in T (MCICA (hs6i)–TC (hryb))

1–Jun–2021 to 31–Aug–2021 from 82 to 92 samples. Verified against 0001.
Cross–hatching indicates 95% confidence with Sidak correction for 20 independent tests.



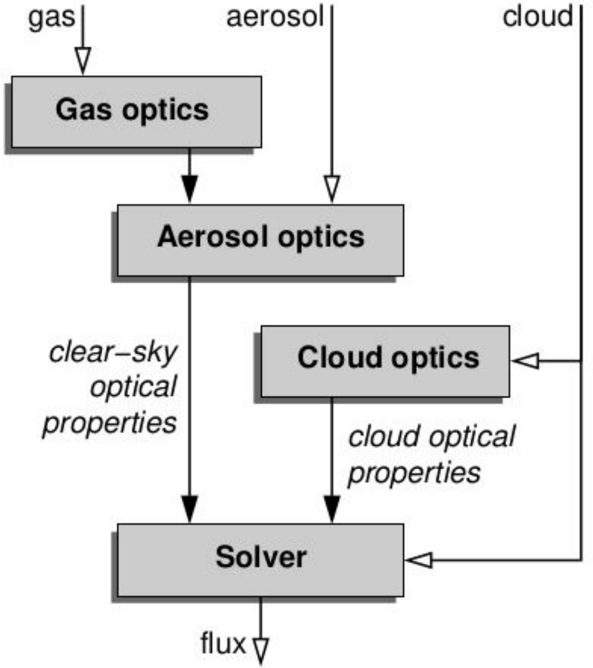


Can machine learning "improve" the trade-off between accuracy and efficiency for radiation?

Answer: **no free lunch with ML.** Recurrent NNs can emulate a radiation scheme very closely but are also slower than (quite inaccurate) feed-forward networks

Before we do a performance comparison, let's revisit where physical codes spend computations and how we can optimize them!

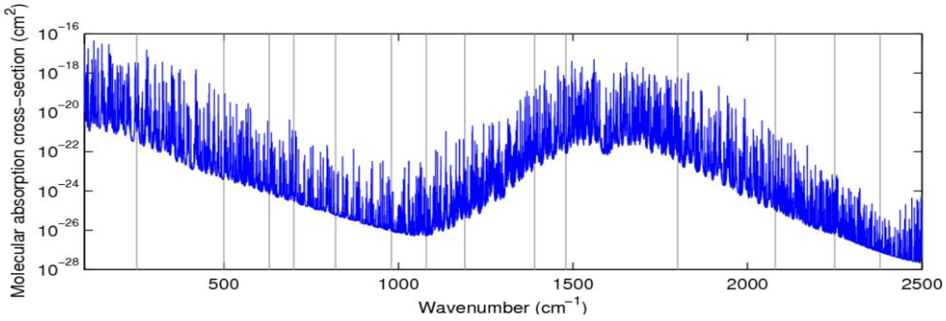
The ecRad scheme



ecRad is a modular, highly configurable radiation scheme developed at ECMWF.

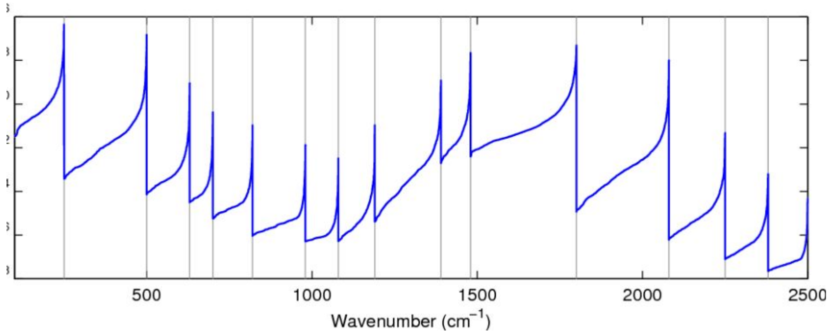
The **gas optics module determines the spectral resolution** and therefore the cost

The correlated-k method

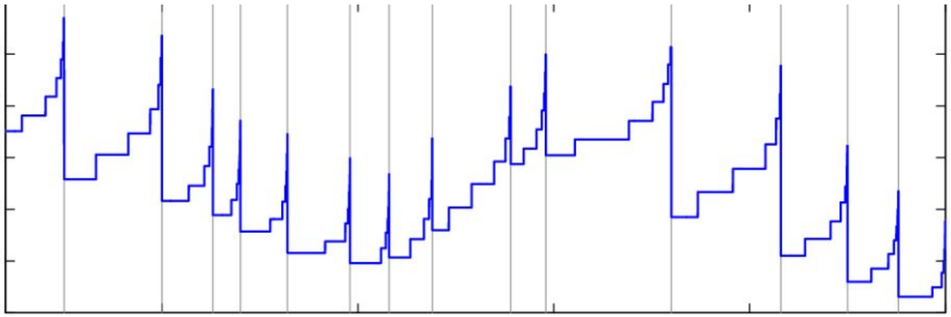


10^6 - 10^7 points needed to numerically integrate the absorption spectrum line-by-line

Divide into bands (or not) and reorder by absorption coefficient



10^2 - 10^3 ”g-points” needed to numerically integrate the cumulative probability functions

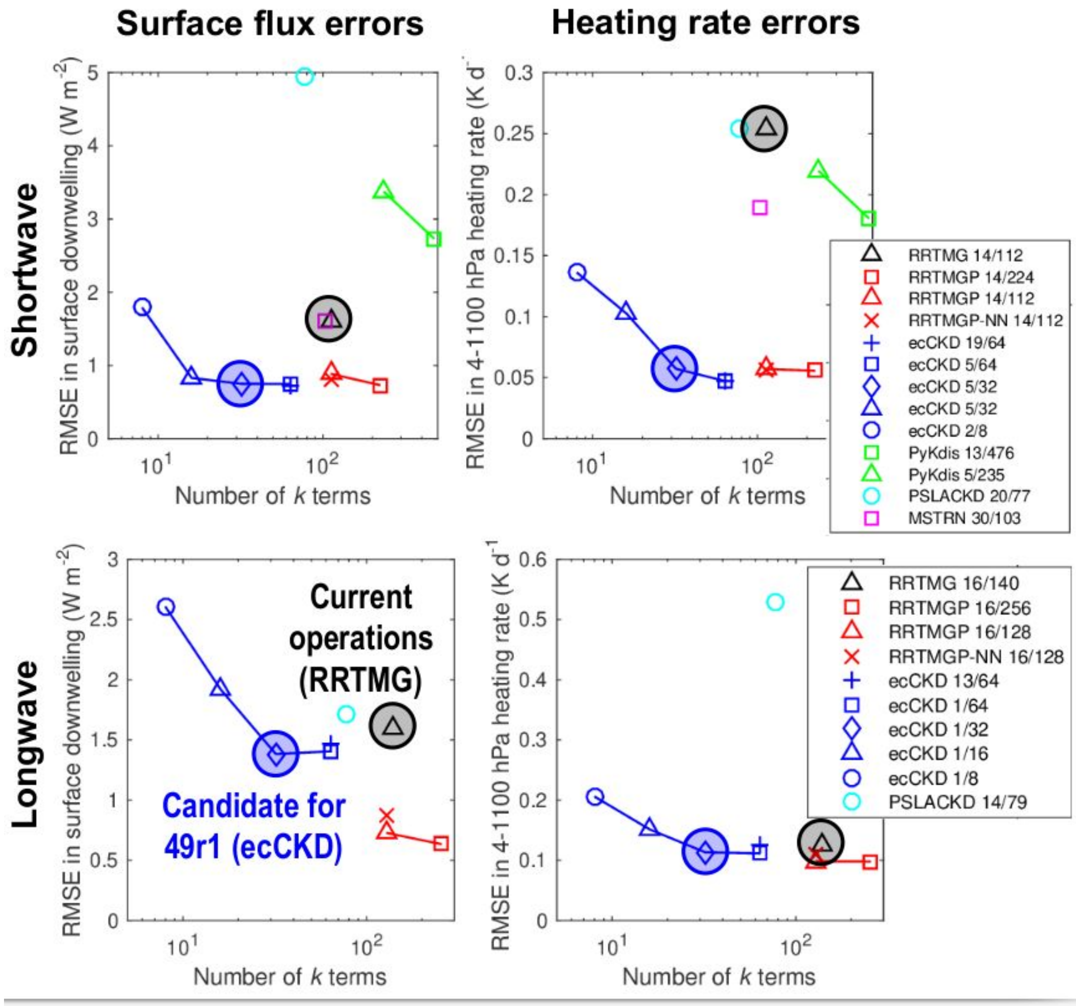


If we want to go beyond two-streams, we need cost savings somewhere else

Recently it’s been shown that we can reasonably reduce **spectral resolution:**

While there is a positive correlation with accuracy, improved methods allow similar accuracy with fewer quadrature points (ecCKD, Hogan & Matricardi 2022)

→3-8x saving in floating point operations compared to operational codes!



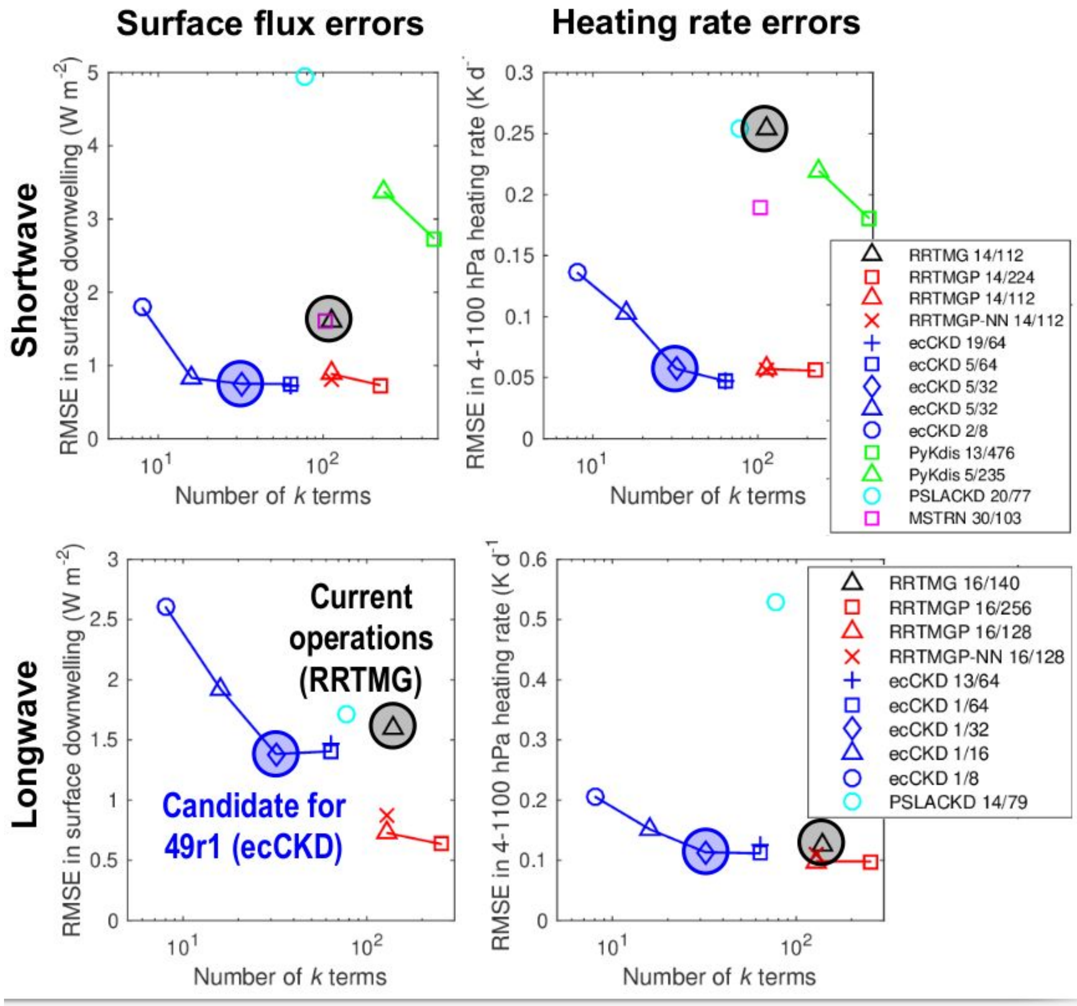
If we want to go beyond two-streams, we need cost savings somewhere else

Recently it’s been shown that we can reasonably reduce **spectral resolution:**

While there is a positive correlation with accuracy, improved methods allow similar accuracy with fewer quadrature points (ecCKD, Hogan & Matricardi 2022)

→3-8x saving in floating point operations compared to operational codes!

...does not result in as big a runtime reduction due to short vectorized loops inhibiting vectorization (but wait!)



Improving the efficiency of radiation schemes

For radiation, the motivation for exploring the use of emulators has always been to improve speed, but there are other ways to achieve this

Ukkonen and Hogan (*in review*): How far can we push efficiency of traditional radiation schemes if we combine

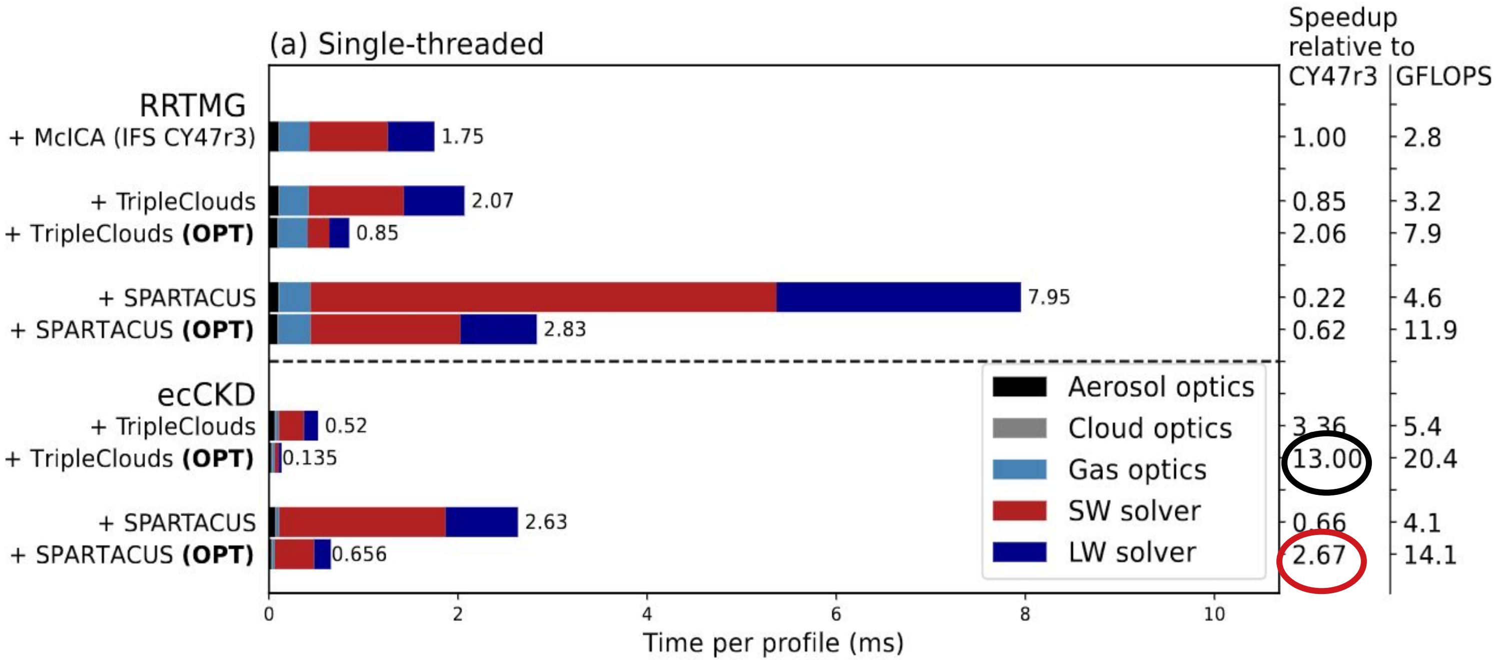
1. Spectrally reduced correlated-*k* gas optics models

   with

2. Code optimization
   1. Higher-level refactoring to expose more parallelism - **important at reduced spectral resolution**
   2. Low-level optimization (avoid double precision, fusing kernels, loop unrolling etc)

https://doi.org/10.22541/essoar.168298700.07329865/v1

A new state-of-the-art in speed/accuracy by combining spectral and code optimization



Spectral reduction yields **~4x speedup** and **code optimization an additional ~4x**

TripleClouds + ecCKD: similar or better accuracy than operational schemes and **13X faster**

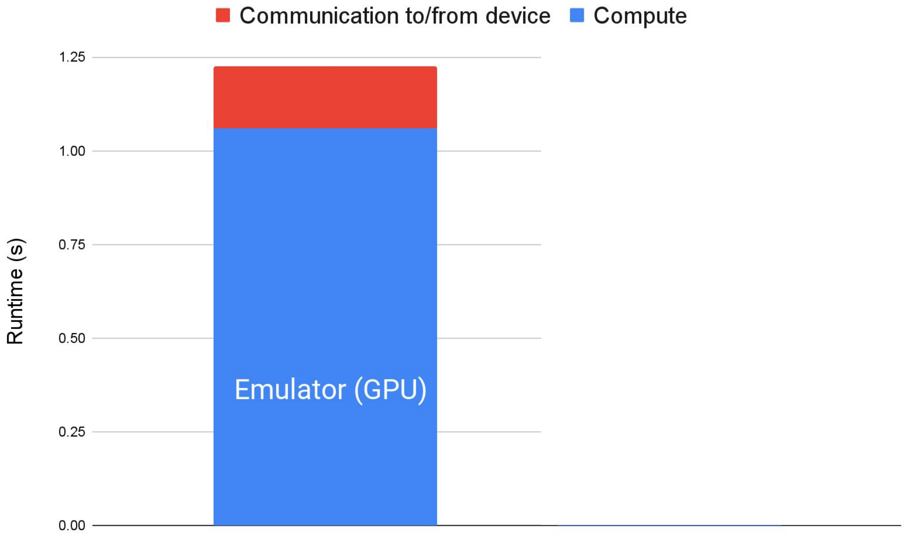
SPARTACUS + ecCKD: accounts for previously ignored 3D cloud radiative effects, **2.6X faster than operational scheme**

Performance comparison: RNN-based emulators on GPU vs optimized TripleClouds on CPU

Time to solution for 400,000 columns (lower is better)

bi-LSTM, 64 neurons
Offline setup with little overhead, ONNX runtime
Very large batch size
(40000 columns)

1x NVIDIA A100 40 GB
Released: May 2020
MSRP: ?
Price: $6800 - $10000
TDP: 400W

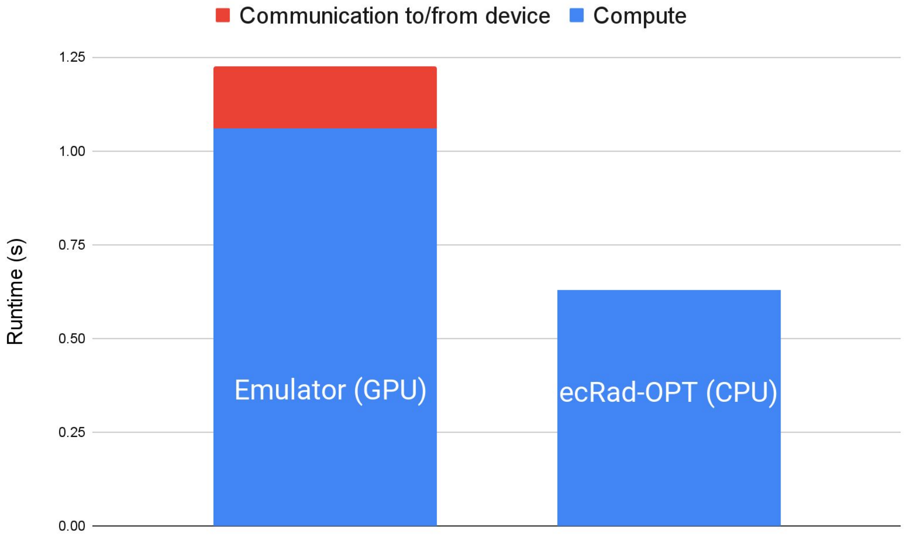


Performance comparison: RNN-based emulators on GPU vs optimized TripleClouds on CPU

bi-LSTM, 64 neurons
Offline setup with little overhead, ONNX runtime
Very large batch size (40000 columns)

1x NVIDIA A100 40 GB
Released: May 2020
MSRP: ?
Price: $6800 - $10000
TDP: 400W

Time to solution for 400,000 columns (lower is better)



Opt. TripleClouds + ecCKD
GCC 9.3, -O3
128 cores = threads
OpenMP loop batch size: 8 columns

2x AMD EPYC 7H12
Released: September 2019
MSRP: ??
Price: $7600 - $14000
TDP: 560W

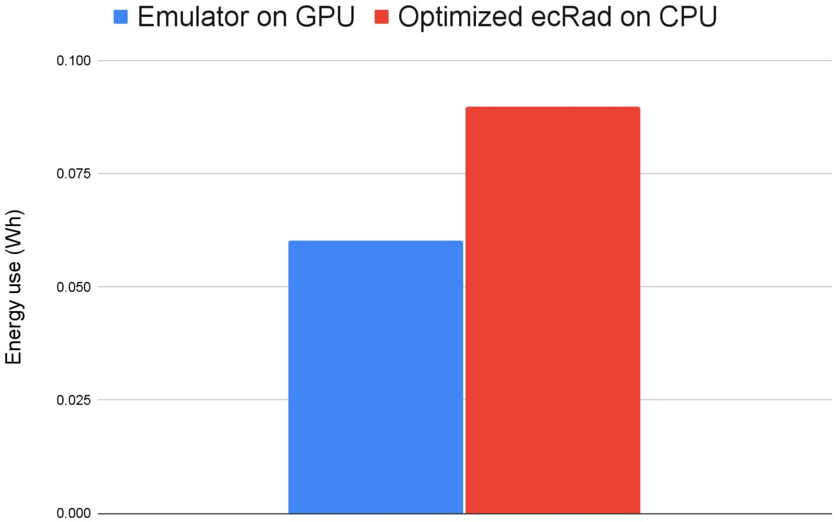
Despite the emulation-friendly setup, optimized radiation code on CPU has almost double the throughput at 635 columns ms^{-1}

Performance comparison: RNN-based emulators on GPU vs optimized TripleClouds on CPU

Energy to solution for 400,000 columns (lower is better)

bi-LSTM, 64 neurons
Offline setup with little overhead, ONNX runtime
Very large batch size
(40000 columns)

1x NVIDIA A100 40 GB
Released: May 2020
MSRP: ?
Price: $6800 - $10000
TDP: 400W

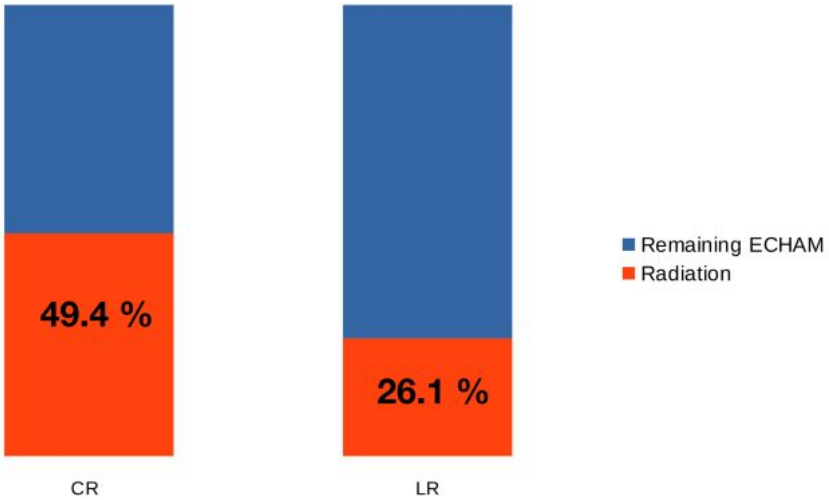


Opt. TripleClouds + ecCKD
GCC 9.3, -O3
128 cores = threads
OpenMP loop batch size: 8 columns

2x AMD EPYC 7H12
Released: September 2019
MSRP: ??
Price: $7600 - $14000
TDP: 560W

Emulators on GPU do consume less power (but smaller batches would make it less efficient)

Moving beyond 1D treatment of radiation



Coarse-res GCM (~460 km)

Low-res (170 km)

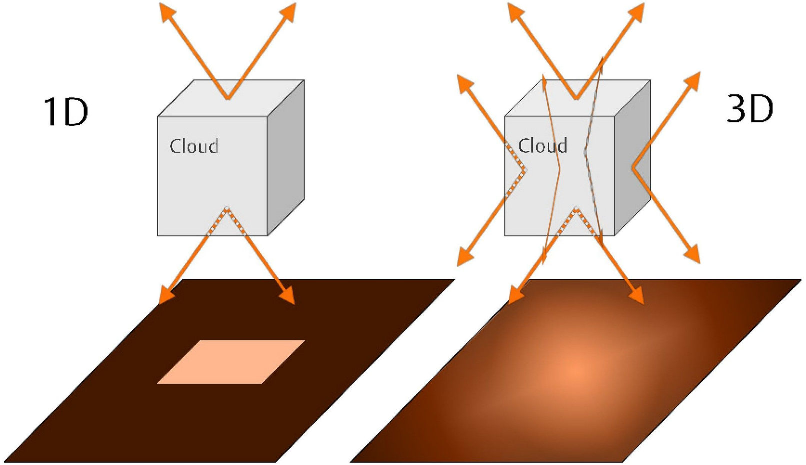
- As simulations are being performed at ever higher resolution, the relative cost of radiation decreases
- In the IFS global weather model, radiation is already only a few % of total runtime (but computed on a coarser grid)
- Combine this with new reduced *k*-distributions and optimization, **radiation is arguably not that expensive anymore - is emulation of operational parameterizations addressing a real problem?**
- Even if it is, it looks like **emulators are not faster than optimized two-stream code,** even when using GPU

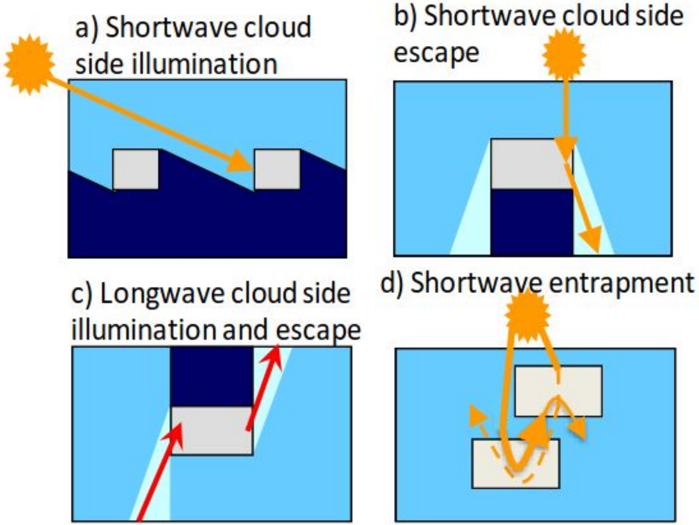
Moving beyond 1D treatment of radiation

Where are we at?

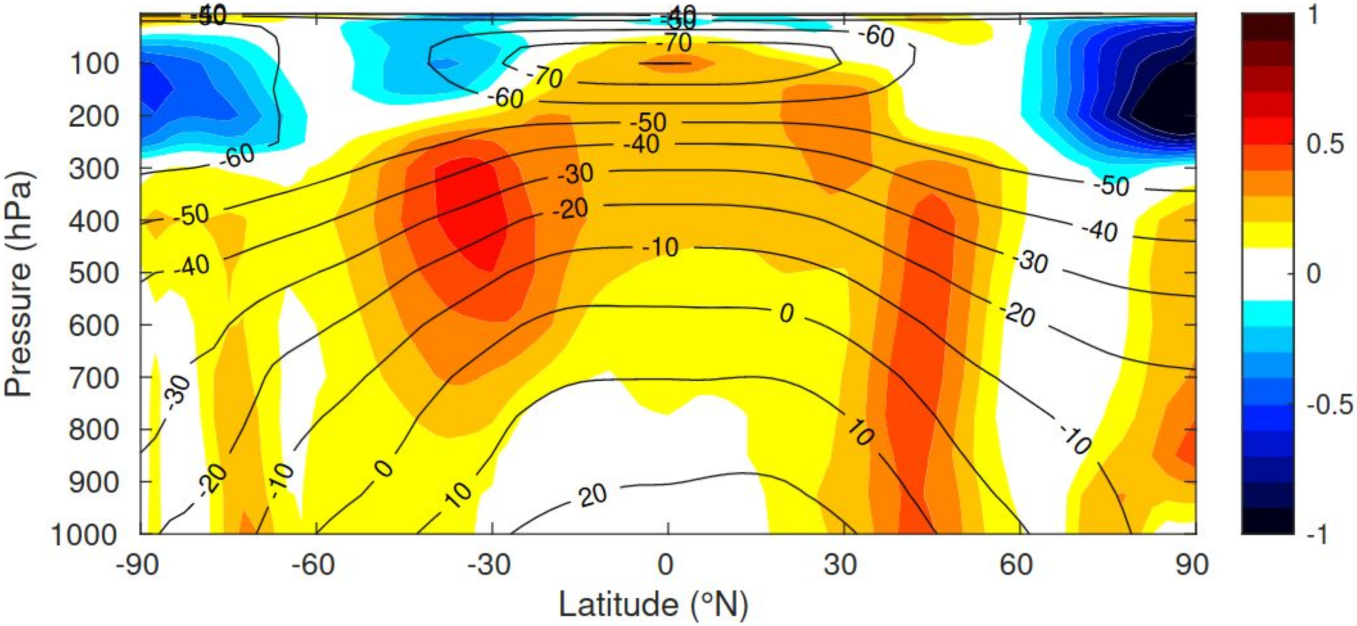
As shown earlier, **SPARTACUS** (Hogan et al, JAMES 2016) is now affordable for weather and climate applications. It computes **sub-grid cloud 3D radiative effects** (radiative flows from cloud sides) by adding extra terms to the two-stream equations

SPARTACUS still operates on 1D columns, so NOT a 3D solver!




a) Shortwave cloud side illumination
b) Shortwave cloud side escape
c) Longwave cloud side illumination and escape
d) Shortwave entrapment

SPARTACUS impact in coupled year-long IFS simulations

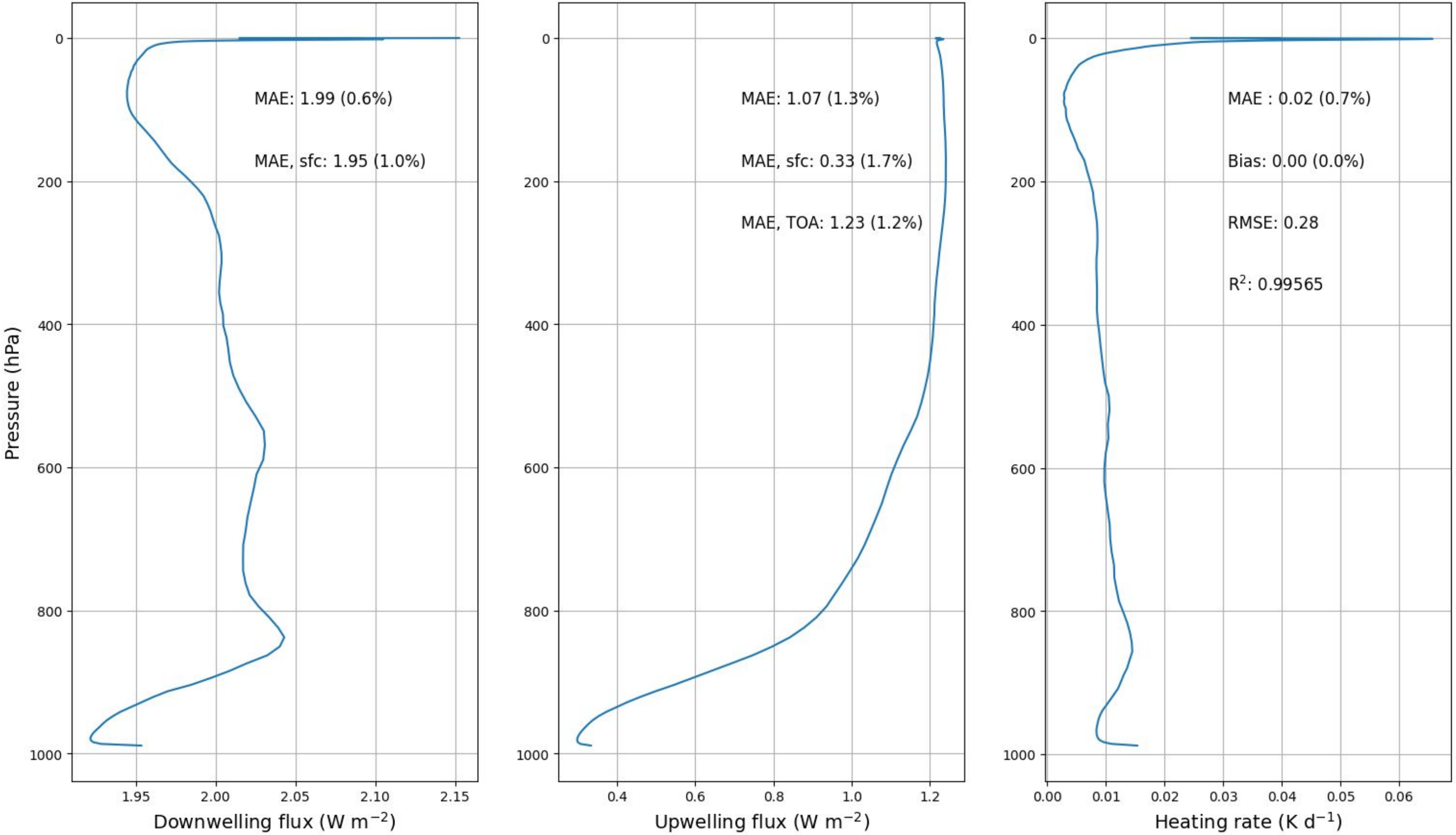


3D effects substantially warm the troposphere, especially in midlatitudes

Caveats:

- SPARTACUS currently overestimates warming effect in the longwave
- Simulations too short to capture ocean response

SPARTACUS can be emulated with similar accuracy as TripleClouds, without increasing RNN complexity

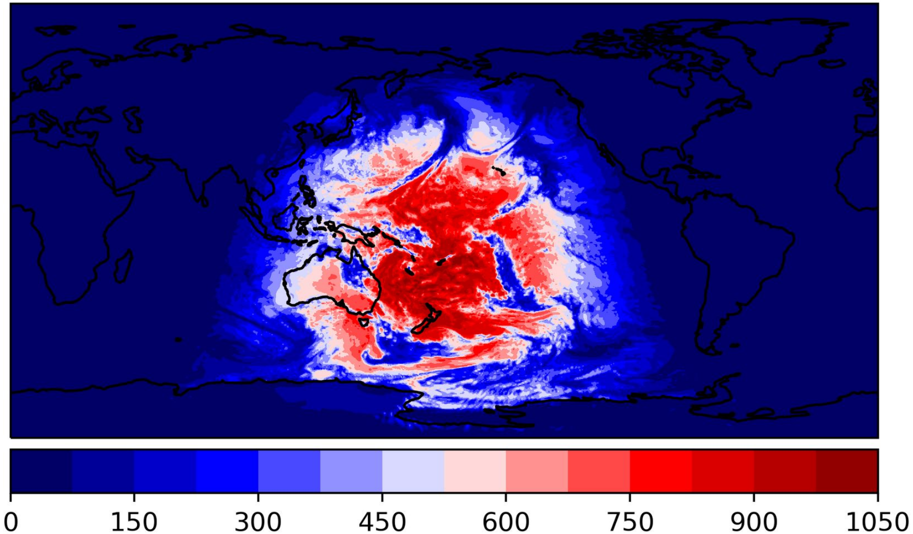


SPARTACUS can be emulated with similar accuracy as TripleClouds, without increasing RNN complexity

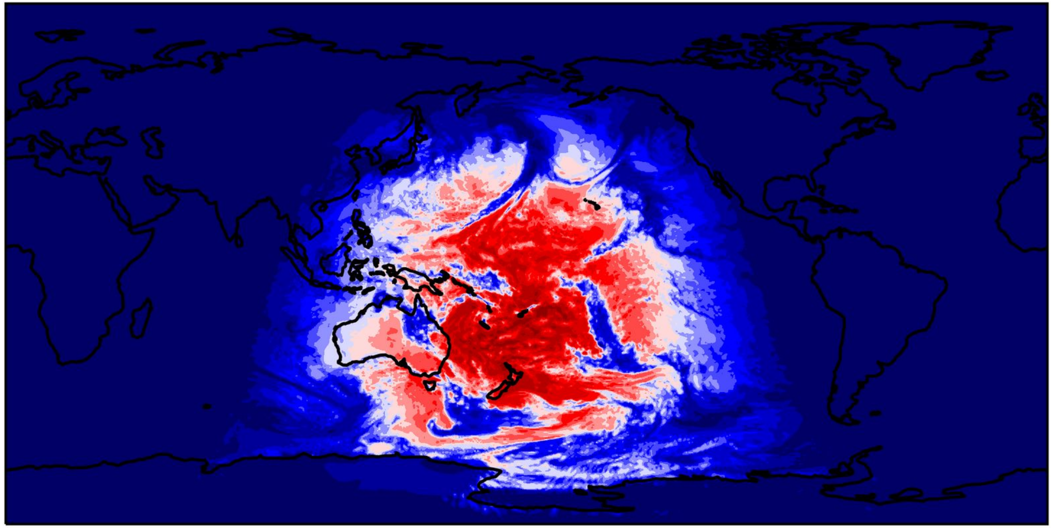
Testing accuracy, fluxes: **R^2 0.996**
Testing accuracy, 3D signal **R^2** ($\text{flux}_{\text{SPART}}$- $\text{flux}_{\text{TripleClouds}}$, flux_{NN}- $\text{flux}_{\text{TripleClouds}}$): **0.998**

Mean 3D signal, true: **0.233**
Mean 3D signal, pred. **0.261**

Optimized SPARTACUS is ~5x slower than TripleClouds, so in the optimal case (with large batches and little communication or other overhead) RNNs emulators on GPU could be faster, and definitely more energy efficient

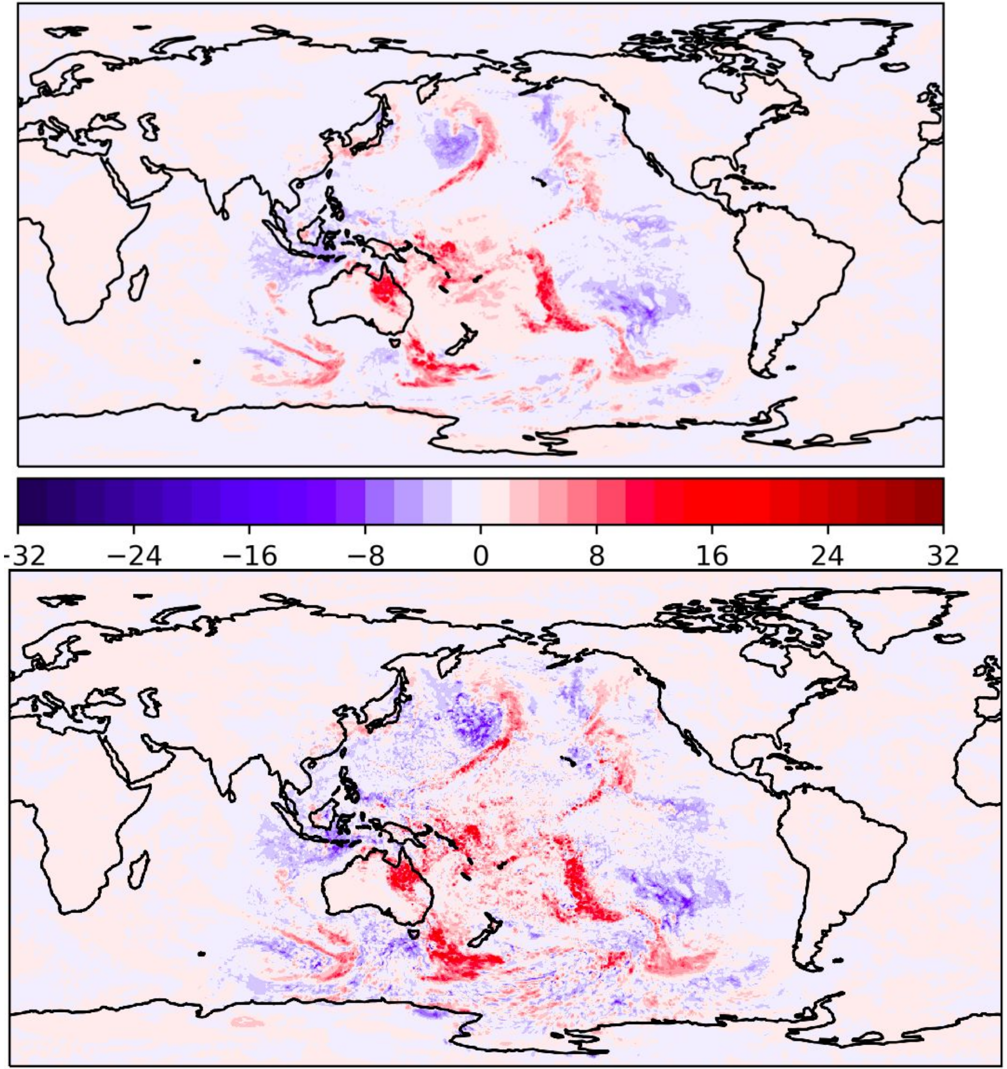


Surface net SW flux, SPARTACUS



Surface net SW flux, RNN

00 UTC 31.1.2019

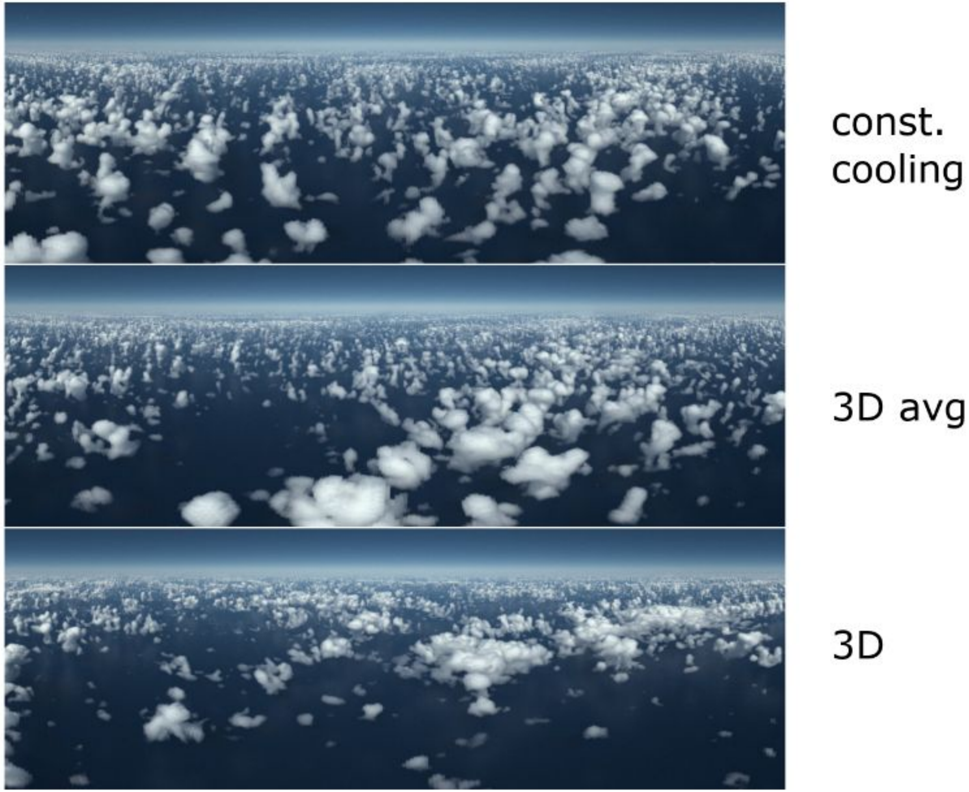


3D effect (difference in surface net flux, SPARTACUS - TripleClouds)

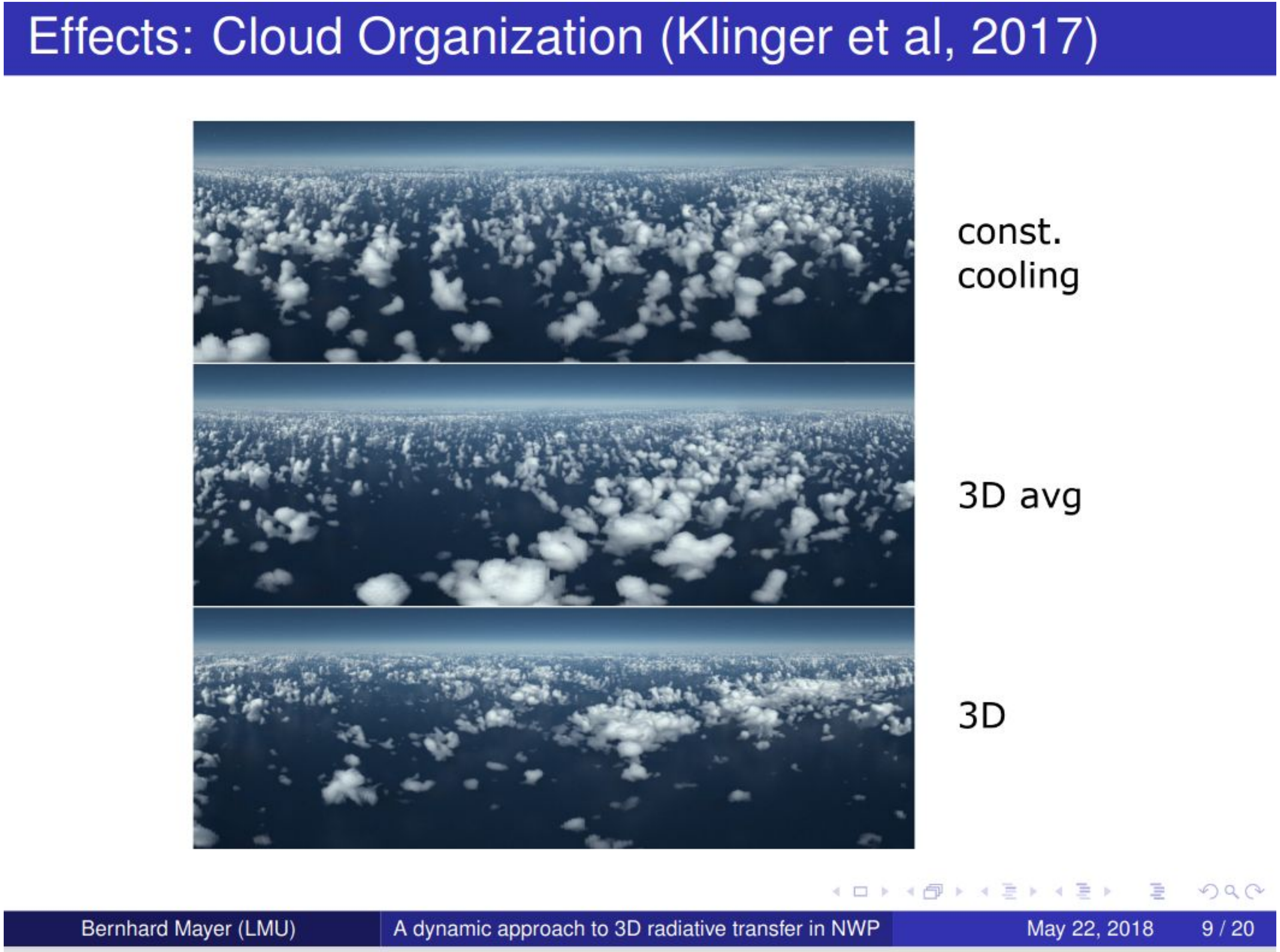
3D effect (difference in surface net flux, RNN - TripleClouds)

00 UTC 31.1.2019

Effects: Cloud Organization (Klinger et al, 2017)



Klinger et al, 2017. Effects of 3-D thermal radiation on the development of a shallow cumulus cloud field



Effects: Cloud Organization (Klinger et al, 2017)

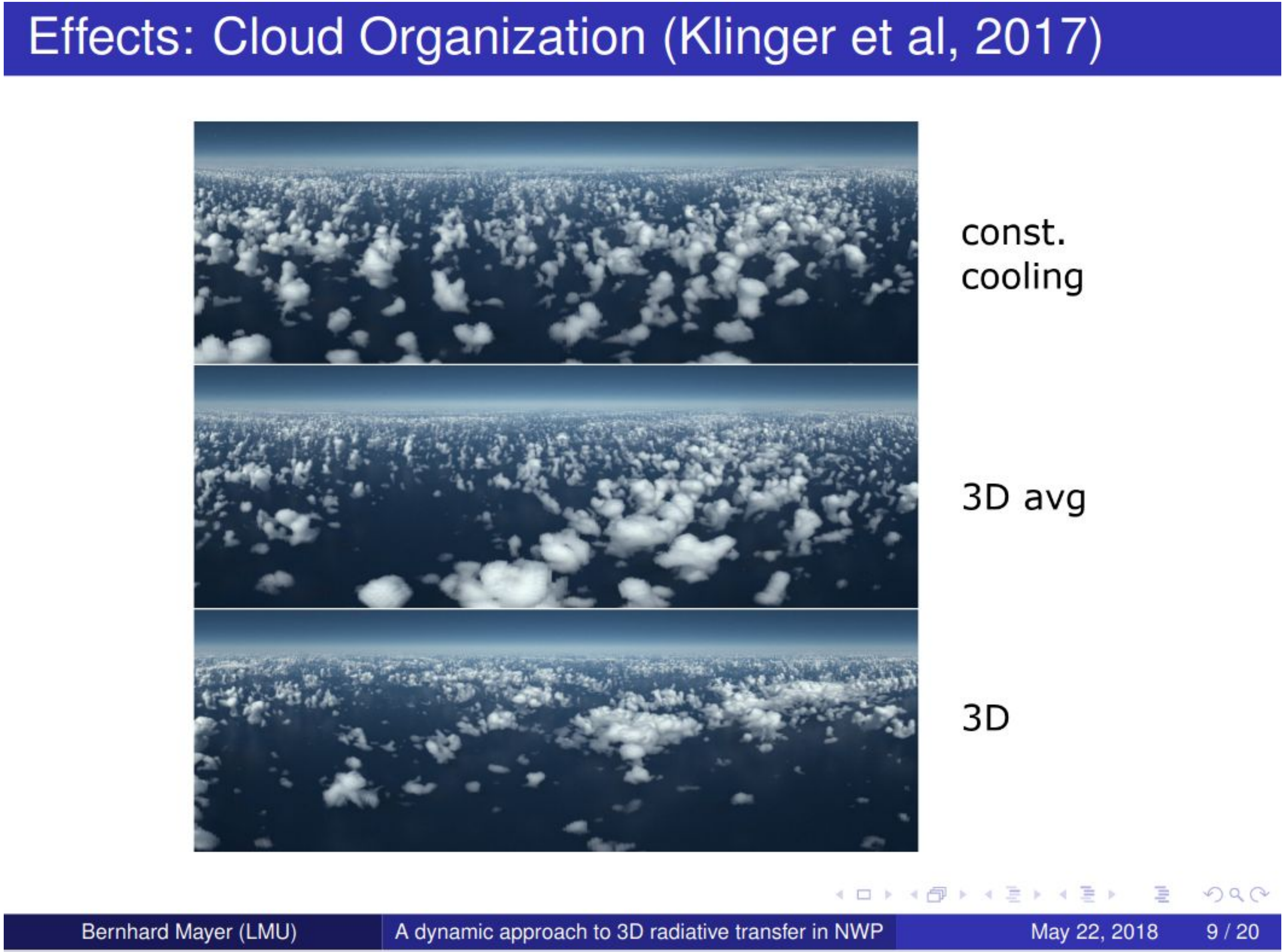
const. cooling

3D avg

3D

← SPARTACUS?

Klinger et al, 2017. Effects of 3-D thermal radiation on the development of a shallow cumulus cloud field



Effects: Cloud Organization (Klinger et al, 2017)

const. cooling

3D avg

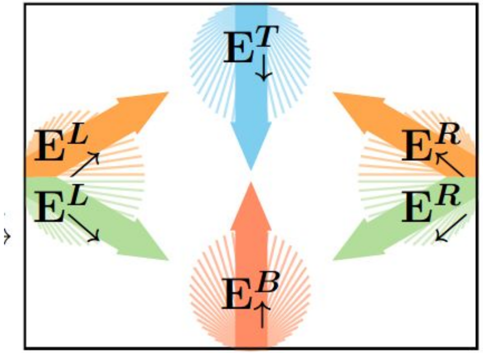
3D

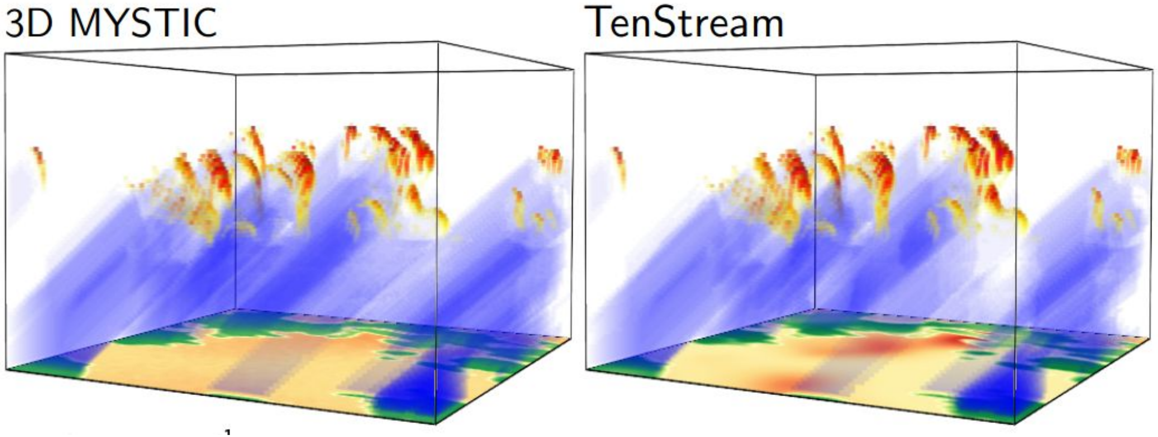
← SPARTACUS?

← What if we want this?

Klinger et al, 2017. Effects of 3-D thermal radiation on the development of a shallow cumulus cloud field

The TenStream solver (Jakub and Mayer, 2015)


$\mathbf{E}^{T}_{\downarrow}$
$\mathbf{E}^{L}_{\nearrow}$
$\mathbf{E}^{R}_{\nwarrow}$
$\mathbf{E}^{L}_{\searrow}$
$\mathbf{E}^{R}_{\swarrow}$
$\mathbf{E}^{B}_{\uparrow}$



3D radiative transfer solver which is 20-70x faster than Monte Carlo; 20-50x slower than two-stream

Fabian Jakub and Bernhard Mayer, 2015. A three-dimensional parallel radiative transfer model for atmospheric heating rates for use in cloud resolving models – The TenStream solver (JQSRT)

Moving beyond 1D treatment of radiation:
Could emulating more complex solvers be more useful?

Yes, in that **RNNs can be used to emulate SPARTACUS** (1D solver with sub-grid 3D effects) **with similar accuracy** as TripleClouds

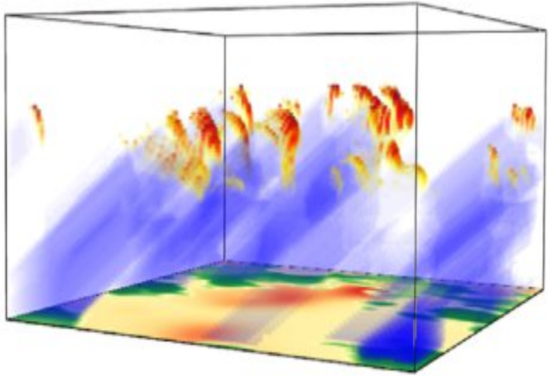
Similarly, RNNs could probably skillfully emulate a more expensive 4-stream 1D-solver

Maybe?, for emulating a full 3D solver - no published results so difficult to say. E.g. transformers would have bad parameter scaling given high dimensionality (N_y*N_x*N_y ~ 100*1000*1000)

Given that local 3D radiative effects have been shown to change cloud circulation, we should try, but perhaps a hybrid approach would work better -

A possible hybrid approach for 3D radiative transfer

Idea: use ML to “downscale” SPARTACUS fluxes on a coarse grid (statistical, domain-average 3D effects) **to “resolved” 3D on a finer grid** (with shadows in correct places)



1. Train NN on TenStream fluxes normalized by the domain-mean flux, with an output softmax layer so that predictions e.g. at a given vertical level sum to 1
2. Run SPARTACUS on coarsened grid (20 km+), computing cloud variability inputs from fine grid (10m-1km scale) similar to that used in training
3. Run NN using fine grid cloud field and multiply the predictions with the SPARTACUS flux and *N* so that the domain average flux is conserved

+ The NN would only downscale/redistribute fluxes, accuracy need not be high. Could adapt ML approaches from other fields

Conclusions

RNNs can emulate 1D schemes closely and are quite fast on GPU - but if we **compare against state-of-the-art radiation code on CPU** (**optimized TripleClouds+ecCKD)**, it’s actually faster (offline - online speed-up even more unlikely)

Given small or negative speed gains (assuming ML architecture which is accurate enough for global NWP), **we should probably move from emulation from 1D schemes! Been studied more than 20 years and still not used operationally.**

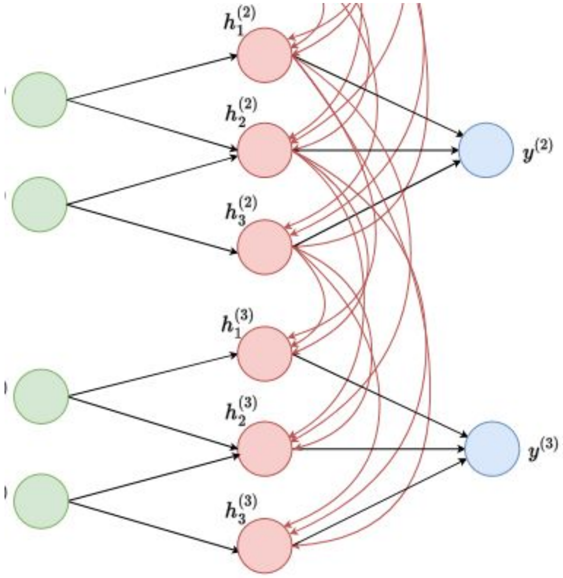
Also worth noting **for climate applications:** crucial metrics such as **radiative forcing have not been evaluated in any full-emulation study!**

Instead, we should move towards the emulation of more sophisticated schemes (e.g. 4-streams, SPARTACUS, 3D solver): easier to get a substantial **speed-up**; prospect to **improve realism** compared to current models

Danmarks
Meteorologiske
Institut

Thanks for listening!

Any questions?


$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
$y^{(2)}$
$h_1^{(3)}$
$h_2^{(3)}$
$h_3^{(3)}$
$y^{(3)}$