A targeted machine learning approach to accelerate radiation computations

Peter Ukkonen[1,2]
[1]Danish Meteorological Institute
[2]Niels Bohr Institute, University of Copenhagen

DMI
Danish Meteorological Institute

UNIVERSITY OF COPENHAGEN

DMI
Danish Meteorological Institute

Climate models are slow, and we should care

- Trend towards higher resolution in weather/climate models

- High-resolution simulations are highly energy intensive: on the state-of-the-art COSMO model run on efficient GPUs at 1 km resolution, 596 MWh per simulated year[1]

  → energy required to simulate 100 years could power 16000 UK homes for a year

- Trend towards more heterogenous and parallel hardware which legacy codes make poor use of (current NWP models realize a few % of peak performance[2])

DMI
Danish Meteorological Institute

Atmospheric radiation: understood, but complex

- Radiative transfer is often one of the most expensive components in large-scale models (50% of runtime in ECHAM[3])

- This is despite using several tricks to simplify the well-understood but complex physics in modern radiation schemes

  - Treat radiation as 1D despite being 3D
  - To get around spectral complexity (millions of absorption lines), use the *correlated-k* method to reorder absorption coefficients in ascending order → *O*(2-3) pseudo-monochromatic spectral point (g-points)

DMI
Danish Meteorological Institute

Machine learning to the rescue?

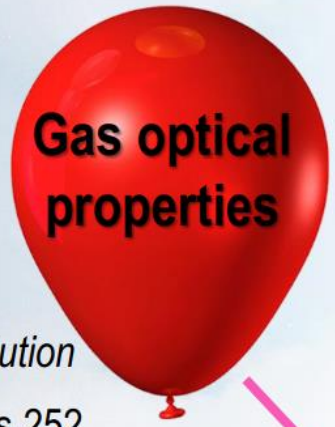
- Neural nets have been used to emulate entire radiation schemes (Krasnopolsky et al. (2010), Pal et al., (2019)..)

- Large speed-ups (10-100x) but at the cost of accuracy and generalization (20 W m^{-2} error in surface flux in Pal et al.)
  - Even small errors can accumulate

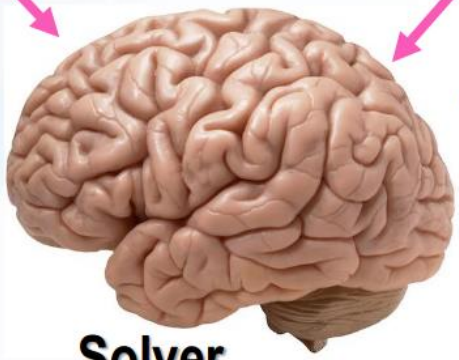
- Operational use would require high accuracy, stability and generalisation (to e.g. warmer climates)

- Alternative to the tall task of emulating the entire radiation code?

The four components of a radiation scheme



- *Determines spectral resolution*
- RRTM-G uses 252 spectral intervals



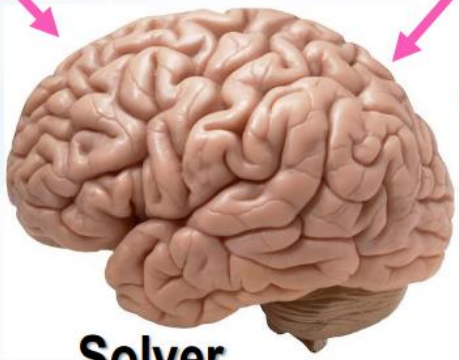
- *Determines how sophisticated interactions with clouds will be*

- Codes should be *modular*, allowing components to be changed independently

The four components of a radiation scheme

Interpolated from look-up-table

- *Determines spectral resolution*
- RRTM-G uses 252 spectral intervals



Physical equations for radiative transfer between layers

- *Determines how sophisticated interactions with clouds will be*

- Codes should be *modular*, allowing components to be changed independently

DMI
Danish Meteorological Institute

Work in a nutshell

- Develop neural networks to replace the computation of optical properties in a new radiation scheme, RTE+RRTMGP
  - RRTMGP (RRTM for General circulation model applications—Parallell) is successor to RRTMG, used widely in GCM and NWP
  - RRTMGP computes optical properties of the atmosphere using (256+224=480) g-points
  - RTE (Radiative Transfer for Energetics) uses two-stream equations to compute radiative transfer
- Combine this with refactoring/optimizing other parts of the code

DMI
Danish Meteorological Institute

Why NNs are (hopefully) more efficient

Original code maps atmospheric conditions to optical properties using LUT interpolation within lots of loops

- For each layer $j = 1...J$ in each column $k = 1...K$
  - For band $b = 1...B$
    - Compute the g-point vector $\overline{\tau, maj}_{b,j,k}$ by 3D linear interpolation in T, p and η.
    - For each minor gas
      - Compute τ_{min} by 2D linear interpolation in temperature and η
    - $\tau_{i,j,g} = \tau_{maj} + \tau_{min}$

DMI
Danish Meteorological Institute

Why NNs are (hopefully) more efficient

Original code maps atmospheric conditions to optical properties using LUT interpolation within lots of loops

- For each layer $j = 1...J$ in each column $k = 1...K$
  - For band $b = 1...B$
    - Compute the g-point vector $\overline{\tau, maj}_{b,j,k}$ by 3D linear interpolation in T, p and η.
    - For each minor gas
      - Compute τ_{min} by 2D linear interpolation in temperature and η
    - $\tau_{i,j,g} = \tau_{maj} + \tau_{min}$

With NNs we can:

- Predict all g-points (224-256) and gas contributions (4-16) simultaneously
  $\mathbf{Y}_{ng} = f(\mathbf{X}_{ngas})$, where Y and X are vectors and *f* is the NN mapping
- (!) Since optical property computations are independent for model levels and columns, we can furthermore collapse nlev and ncol into one dimension *k* and do batch predictions for $\mathbf{Y}_{ng,k}$
- The core computations are then **matrix-matrix multiplications** which we can delegate to an optimized library (BLAS)

DMI
Danish Meteorological Institute

Developing NNs to replace a physics scheme: lots of data needed!

- Obtain atmospheric profiles (T, p, q(H_2O), q(O_3), q(CO_2),..) from many sources
  - Reanalyses
  - climate projections
  - Idealized profiles
  - Sample present-day, preindustrial, future, LGM..
- Expand data synthetically: hypercube sampling of gas concentrations, tweak temps and humidities etc.

DMI
Danish Meteorological Institute

Developing NNs to replace a physics scheme: lots of data needed!

- Obtain atmospheric profiles (T, p, q(H_2O), q(O_3), q(CO_2),..) from many sources
  - Reanalyses
  - climate projections
  - Idealized profiles
  - Sample present-day, preindustrial, future, LGM..
- Expand data synthetically: hypercube sampling of gas concentrations, tweak temps and humidities etc.
- Use RRTMGP to compute outputs (absorption coefficients, emission)

  → 7 million training samples spanning **a large range** of atmospheric conditions

Accuracy

DMI
Danish Meteorological Institute

After lots of experimenting, good results (R^2 >0.99) by using separate neural nets to predict molecular (=independent of layer thickness):

- LW (=longwave, terrestrial) absorption
- LW emission
- SW (=shortwave,solar) absorption
- SW scattering

Each NN outputs a vector of sizes 224 or 256, has 2 hidden layers and 18-64 neurons, and both inputs and outputs were scaled

e.g. log(p), $\sqrt{q(H_20)}$

+normalization

Accuracy

DMI Danish Meteorological Institute

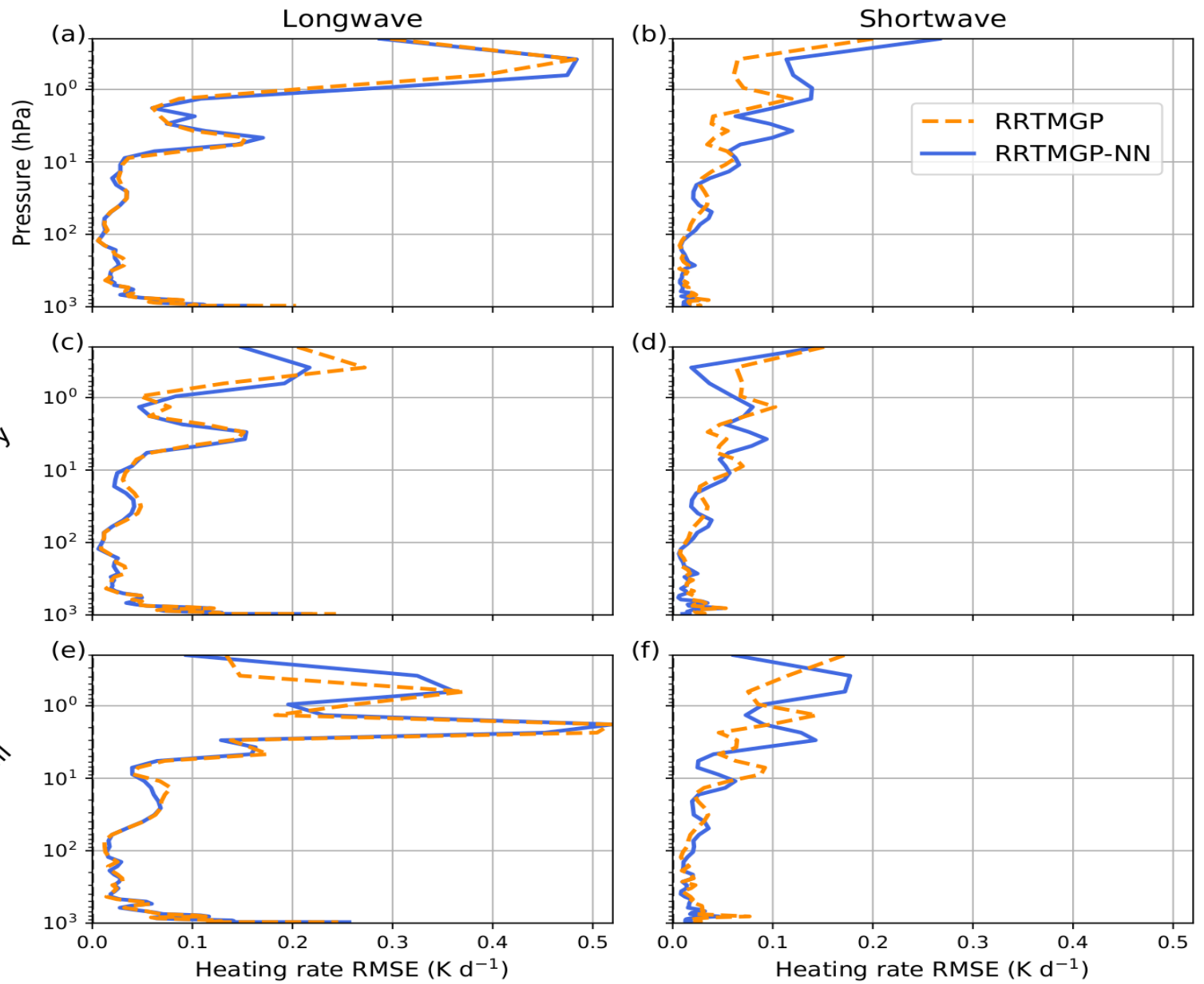
After lots of experimenting, good results (R^2 >0.99) by using separate neural nets to predict molecular (=independent of layer thickness):

- LW (=longwave, terrestrial) absorption
- LW emission
- SW (=shortwave,solar) absorption
- SW scattering

Each NN outputs a vector of sizes 224 or 256, has 2 hidden layers and 18-64 neurons, and both inputs and outputs were scaled

e.g. log(p), $\sqrt{q(H_20)}$

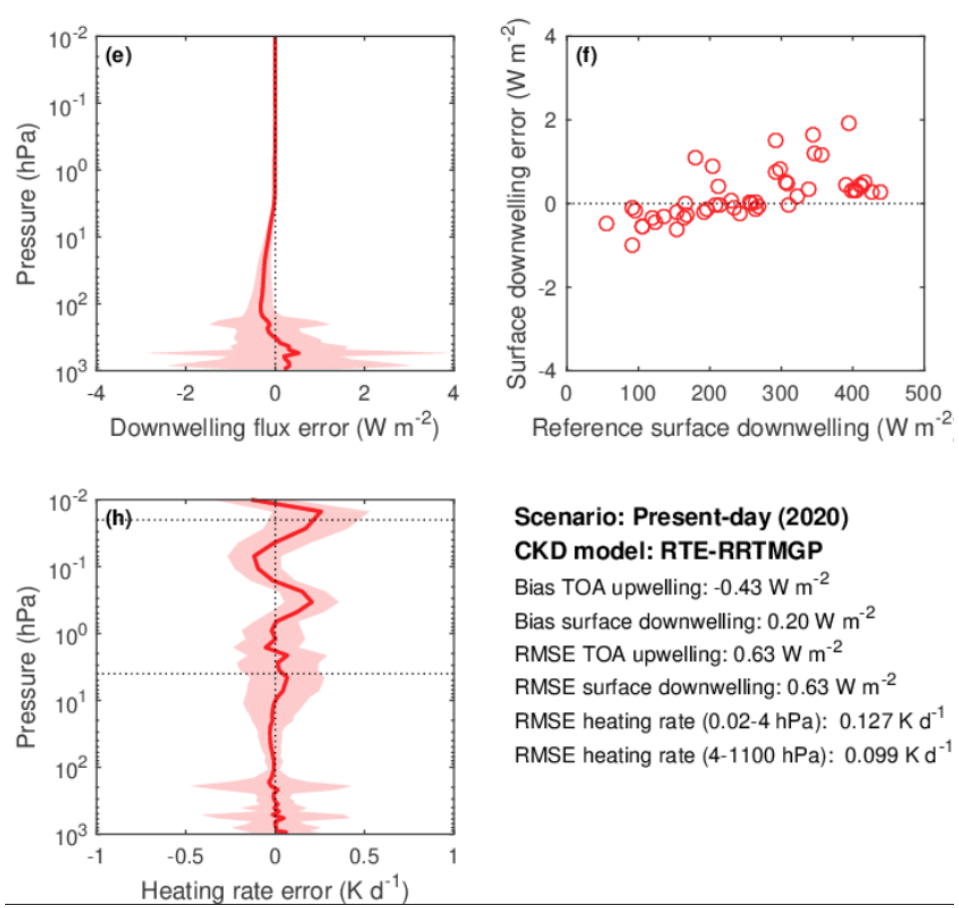
+normalization



Accuracy (LW)

DMI
Danish Meteorological Institute

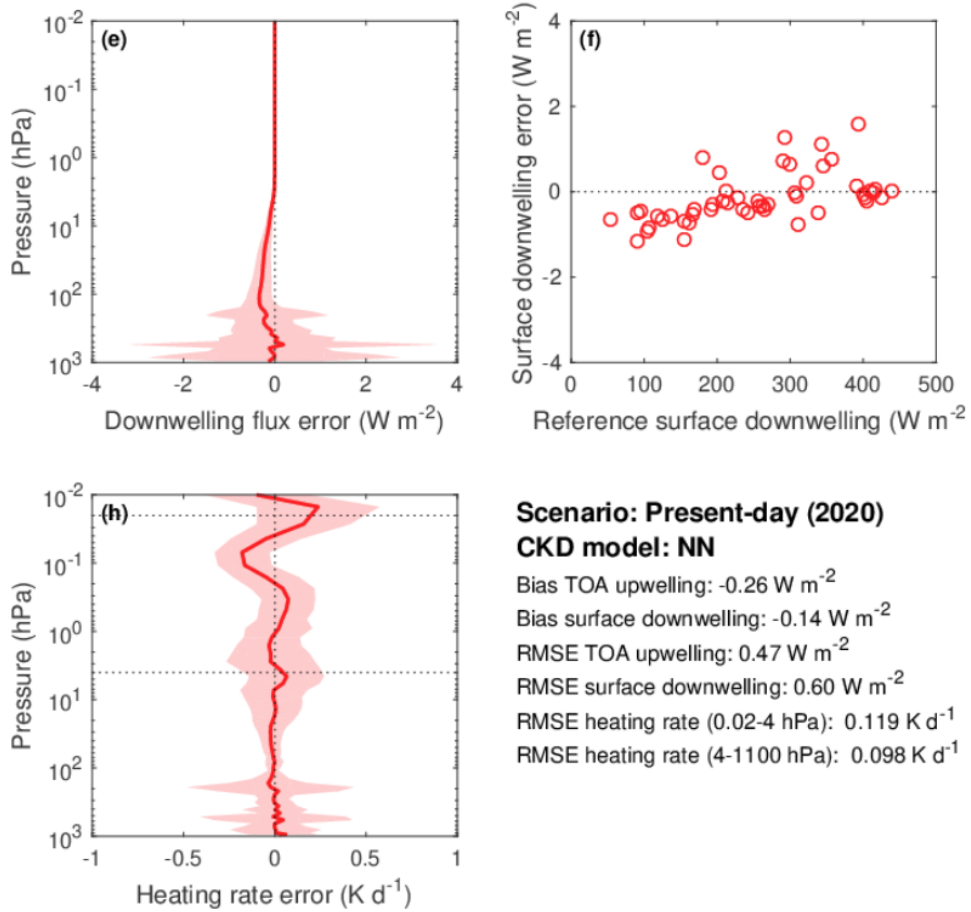
REFERENCE



Scenario: Present-day (2020)
CKD model: RTE-RRTMGP

Bias TOA upwelling: -0.43 W m^{-2}
Bias surface downwelling: 0.20 W m^{-2}
RMSE TOA upwelling: 0.63 W m^{-2}
RMSE surface downwelling: 0.63 W m^{-2}
RMSE heating rate (0.02-4 hPa): 0.127 K d^{-1}
RMSE heating rate (4-1100 hPa): 0.099 K d^{-1}

NEURAL NET



Scenario: Present-day (2020)
CKD model: NN

Bias TOA upwelling: -0.26 W m^{-2}
Bias surface downwelling: -0.14 W m^{-2}
RMSE TOA upwelling: 0.47 W m^{-2}
RMSE surface downwelling: 0.60 W m^{-2}
RMSE heating rate (0.02-4 hPa): 0.119 K d^{-1}
RMSE heating rate (4-1100 hPa): 0.098 K d^{-1}

DMI
Danish Meteorological Institute

Implementation: neural networks in Fortran is simple and performance portable

- Neural nets were trained in Python, need to be used in Fortran

- I took an existing OO Fortran implementation of dense neural networks (neural-Fortran[4]), wrote new kernels using BLAS for batched inference

DMI
Danish Meteorological Institute

Implementation: neural networks in Fortran is simple and performance portable

- GPUing the code is trivial

1) Call NVIDIAs cuBLAS instead of BLAS

2) OpenACC directives above loops

```
#ifdef USE_OPENACC
  use cublas
  use openacc
#define sgemm cublassgemm
#endif
```

```
! 1. Multiply inputs with weights of first layer

!$acc host_data use_device(wt, x, a)
call sgemm('N','N', neurons, nbatch, nx, 1.0, wt, neurons, x, nx, 0.0, a, neurons)
!$acc end host_data

! 2. Add biases and activation

!$acc parallel loop gang vector collapse(2) default(present)
do j = 1, nbatch
  do i = 1, neurons
    a(i, j) = a(i, j ) + b(i)
    call activation_softsign(a(i, j))
  end do
end do

! 3. Repeat steps 1-2 until final layer reached

do n = 3, nlayers-1
```

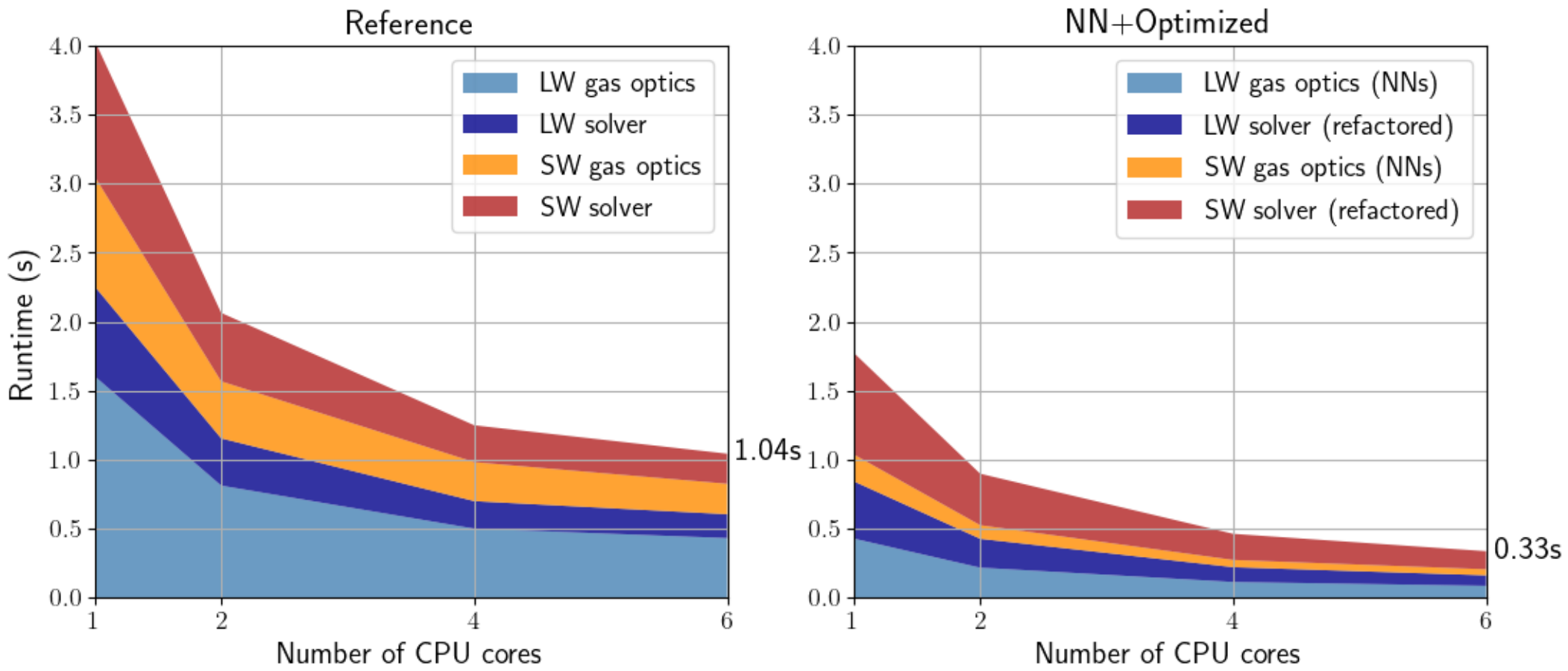
Code refactoring: switch around dimensions, pretty much

- The original RTE code uses columns as inner dimension, but RRTMGP has columns as outer dimension

  optical_depth(ng, nlev, ncol), radn_up(ncol, nlev, ng)

- In between expensive array transposes are needed. This motivated rewriting RTE to columns-last

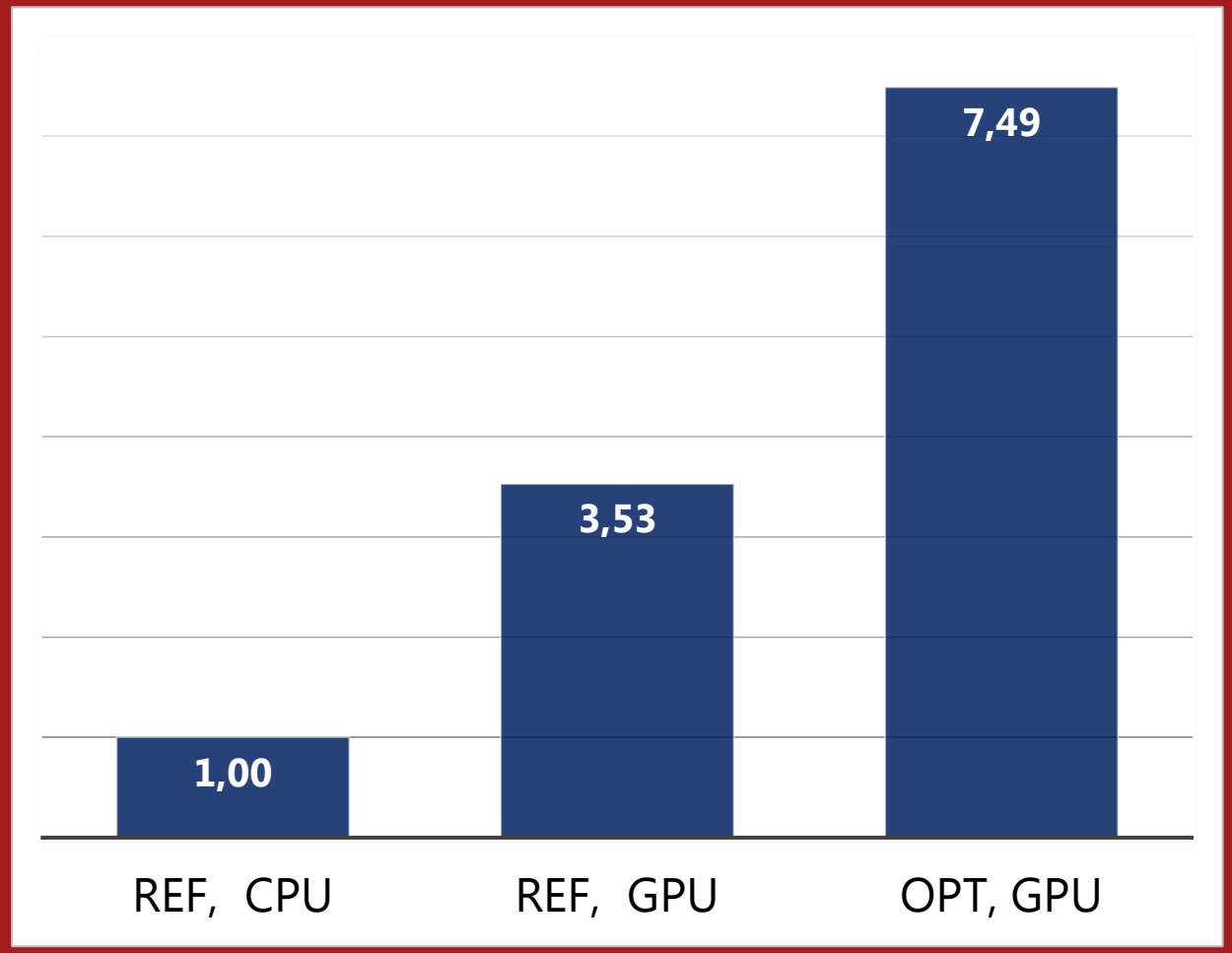
Time to solution, clear-sky computations for 6400 columns



CPU: AMD Ryzen 2600
Compiler: GCC 9.3, -O3
BLAS library: BLIS

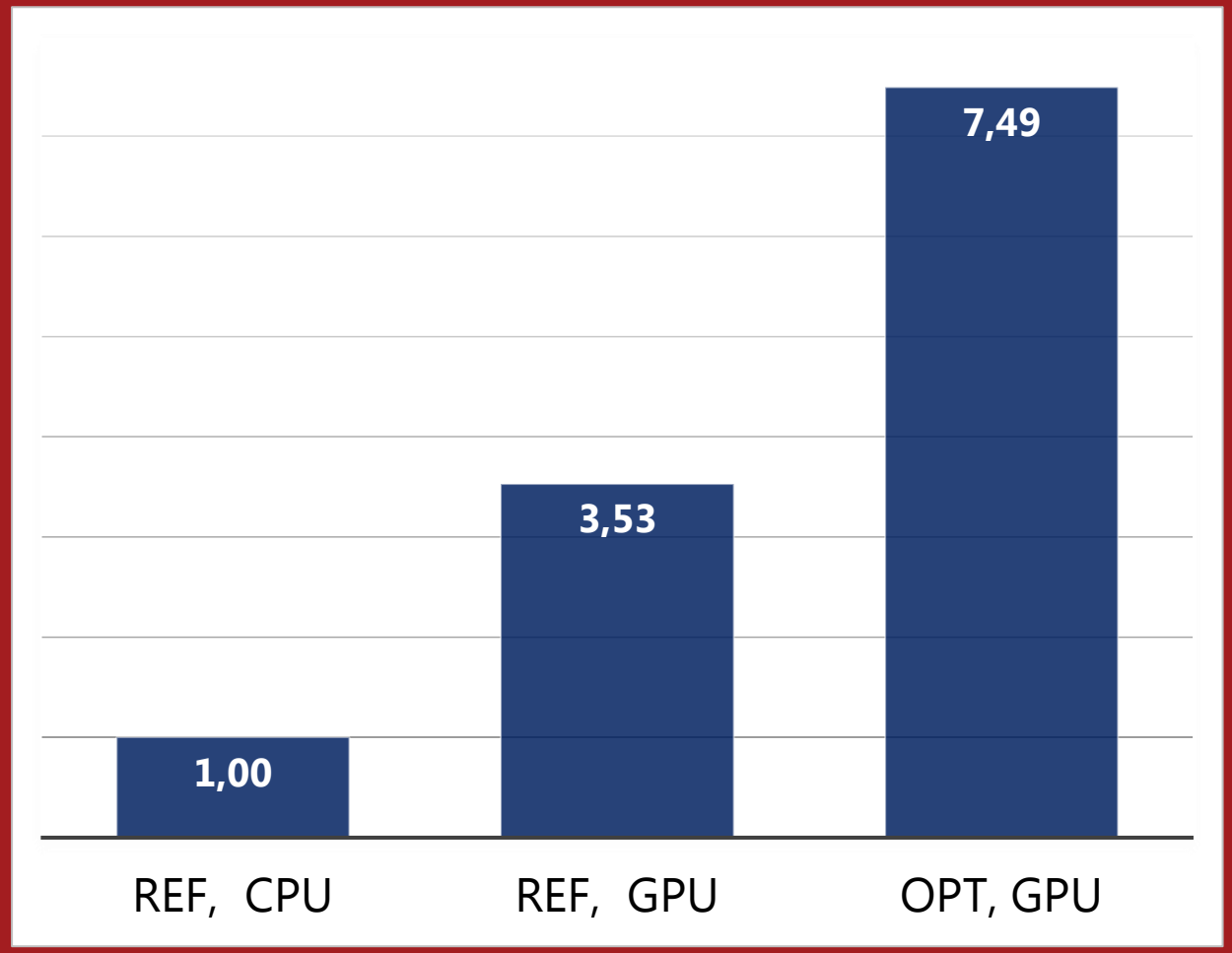
Only shortwave (SW) computations include scattering

Speed-up factor over REF (CPU)



CPU: Ryzen 2600, 6 cores
GPU: GTX 1070 (no tensor cores!)

Speed-up factor over REF (CPU)



The new gas optics has high computational efficiency due to BLAS
- 50 GFLOPS vs 5 GFLOPS for original kernel!

The ECMWF radiation scheme is even worse: 1-2 GFLOPS (but improvements are coming)

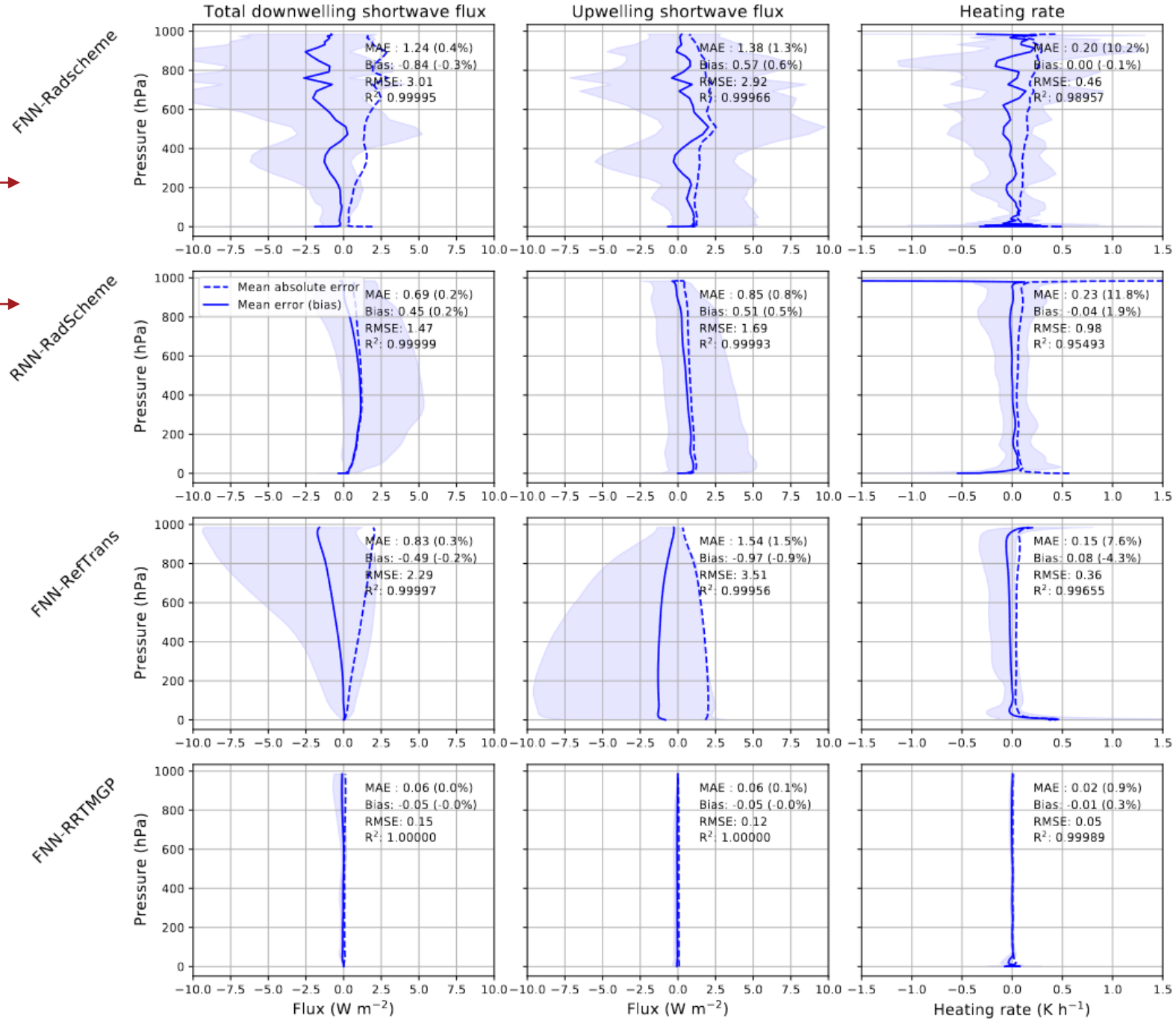
CPU: Ryzen 2600, 6 cores
GPU: GTX 1070 (no tensor cores!)

Bonus slide – UNPUBLISHED RESULTS

NN Emulation of entire radiation scheme

NN Emulation of one component in the solver (reflectance-transmittance)

NN Emulation of gas optics



Take-home message

- Climate and weather models make poor use out of modern computer hardware
- NNs can be faster than traditional parameterizations, but focusing on less exact, more empirical components may give good results mor easily
- Refactoring existing codes could also go a long way
- We combined the two to speed up a new radiation scheme by ~3x on CPUs and GPUs, seemingly without losing accuracy
- RTE+RRTMGP-NN is designed to be a usable replacement of the original code

For any questions, feel free to

- Read our paper : Ukkonen, P., Pincus, R., Hogan, R. J., Pagh Nielsen, K., & Kaas, E. (2020). Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. *Journal of Advances in Modeling Earth Systems*, *12*(12)

- Check out the code at https://github.com/peterukk/rte-rrtmgp-nn
- Email me (peterukk@gmail.com / puk@dmi.dk)

ESCAPE-2: Energy-efficient SCalable Algorithms for weather and climate Prediction at Exascale

European Union's Horizon 2020 research and innovation programme. Grant Number: 800897

DMI
Danish Meteorological Institute

References

1) Fuhrer et al. (2018). Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0. https://doi.org/10.5194/gmd-11-1665-2018
2) Michalakes et al. (2016). Optimizing weather model radiative transfer physics for intel's many integrated core (MIC) architecture. Parallel Processing Letters, 26(04), 1650019
3) Cotronei, A., & Slawig, T. (2020). Single-precision arithmetic in ECHAM radiation reduces runtime and energy consumption. https://doi.org/10.5194/gmd-13-2783-2020
4) Curcic, M. (2019). A parallel Fortran framework for neural networks and deep learning. https://doi.org/10.1145/3323057.3323059

Pal et al. (2019). Using deep neural networks as Cost-Effective surrogate models for Super-Parameterized E3SM radiative transfer.

Krasnopolsky et al.. (2010). Accurate and fast neural network emulations of model radiation for the NCEP coupled climate forecast system: Climate simulations and seasonal predictions. Monthly Weather Review, 138(5), 1822–1842.