Accelerating radiation computations for dynamical models with targeted machine learning and code optimization

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE
10.1029/2020MS002226

Key Points:
- Neural networks (NNs) were trained to predict the optical properties of the gaseous atmosphere
- Training data were generated with a recently developed radiation scheme for dynamical models (RRTMGP)
- RRTMGP-NN is roughly 3 times faster than the reference code and has a similar accuracy, also in future climate scenarios

Correspondence to:
P. Ukkonen
peter.ukkonen@nbi.ku.dk

Citation:
Ukkonen, P., Pincus, R., Hogan, R. J., Nielsen, K. P., & Kaas, E. (2020). Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. *Journal of Advances in Modeling Earth Systems*, *12*, e2020MS002226. https://doi.org/10.1029/2020MS002226

Received 30 JUN 2020
Accepted 11 NOV 2020
Accepted article online 15 NOV 2020

Accelerating Radiation Computations for Dynamical Models With Targeted Machine Learning and Code Optimization

Peter Ukkonen[1,2], Robert Pincus[3,4], Robin J. Hogan, Kristian Pagh Nielsen[15], and Eigil Kaas[2]

[1]Danish Meteorological Institute, Copenhagen, Denmark, [2]Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark, [3]Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, CO, USA, [4]NOAA Physical Sciences Laboratory, Boulder, CO, USA, [5]European Centre for Medium-Range Weather Forecasts, Reading, UK

Abstract Atmospheric radiation is the main driver of weather and climate, yet due to a complicated absorption spectrum, the precise treatment of radiative transfer in numerical weather and climate models is computationally unfeasible. Radiation parameterizations need to maximize computational efficiency as well as accuracy, and for predicting the future climate many greenhouse gases need to be included. In this work, neural networks (NNs) were developed to replace the gas optics computations in a modern radiation scheme (RTE+RRTMGP) by using carefully constructed models and training data. The NNs, implemented in Fortran and utilizing BLAS for batched inference, are faster by a factor of 1–6, depending on the software and hardware platforms. We combined the accelerated gas optics with a refactored radiative transfer solver, resulting in clear-sky longwave (shortwave) fluxes being 3.5 (1.8) faster to compute on an Intel platform. The accuracy, evaluated with benchmark line-by-line computations across a large range of atmospheric conditions, is very similar to the original scheme with errors in heating rates and top-of-atmosphere radiative forcings typically below 0.1 K day^{-1} and 0.5 W m^{-2}, respectively. These results show that targeted machine learning, code restructuring techniques, and the use of numerical libraries can yield material gains in efficiency while retaining accuracy.

Plain Language Summary Solar and terrestrial radiation interact with Earth's atmosphere, surface, and clouds and provide the energy which drives climate and weather. Simulating these radiative flows in climate and weather models is crucial and can also be very time-consuming. One possible way to model radiative effects more efficiently is to use neural networks or similar machine learning algorithms, but predictions are not guaranteed to be realistic because such models do not use physical equations. Here we investigate using neural networks to replace only one part of traditional radiation code, where the optical properties of the atmosphere are computed. We have found that this approach can be several times faster, while still being accurate in various situations, such as simulating future climate.

1. Introduction

Atmospheric radiation is the fundamental energy source which drives weather and climate. For this reason, representing the exchanges of radiation is crucial to models of the atmosphere. Net radiative fluxes at the surface, in the atmosphere, and at the top of the atmosphere provide the main diabatic forcing to these models. In climate models it is particularly important to capture the changes in Earth's radiative equilibrium over time as accurately as possible. For medium-range weather forecasts—from days to weeks ahead—the accumulated radiative heating or cooling is important for changes in the large-scale weather patterns (Shepherd et al., 2018). For short-range forecasts, radiative effects can have a large impact on local surface temperature and the evolution of convective systems such as tropical cyclones (Mandal et al., 2004) and supercell storms (Markowski & Harrington, 2005).

Unlike many other parameterized processes in dynamical models, such as clouds and convection, atmospheric radiative transfer is a well-understood problem that can be very accurately modeled. The absorption spectra of atmospheric constituents, however, consist of hundreds of thousands of spectral lines.

To get around this complexity, modern radiation codes usually rely on the k-distribution method and the correlated-k approximation (e.g., Goody et al., 1989). The method entails reordering the highly variable spectrum of absorption coefficient as a function of wavelength, $k(\lambda)$, by k so that it is replaced by a monotonically increasing function $k(g)$, where $g(k)$ is the cumulative distribution function. This smooth function can be integrated using a small number of quadrature points, known as g-points, reducing the number of monochromatic computations required to retrieve fluxes in the shortwave and longwave (LW) spectra by many orders of magnitude compared to line-by-line (LBL) methods. K-distributions can be applied to an inhomogeneous medium such as the atmosphere by assuming that the mapping from wavelengths to g-space is perfectly correlated for adjacent atmospheric layers, an approximation which typically allows fluxes and heating rates to be calculated with errors of less than 1% (Fu & Liou, 1992).

Despite the efficiency of the correlated k-distribution (CKD) method, radiation computations remain expensive enough that they are often performed on a coarser horizontal and temporal grid than other computations. For example, in the high-resolution forecast model of the European Centre for Medium-Range Weather Forecasts (ECMWF), the radiation scheme is called every hour on a grid with 10.24 times fewer columns than the rest of the model (Hogan & Bozzo, 2018). A reduced grid for radiation is also used in the superparameterized Energy Exascale Earth System Model (SP-E3SM) (Hannah et al., 2020). This is because radiation is often one of the most expensive components in climate models, accounting for nearly 50% of the runtime of the ECHAM atmospheric model in coarse-resolution configurations (Cotronei & Slawig, 2020), for example.

How, then, can the efficiency of radiation computations be further improved? One option is to reduce the amount of g-points. For NWP, the full-spectrum correlated-k method (Hogan, 2010) is promising as it requires fewer quadrature points to achieve a given accuracy, although this can also be achieved in other ways as seen in the evolution from RRTM to RRTMG (Iacono et al., 2008). Doing computations in single-precision can also reduce runtime by about 40% (Cotronei & Slawig, 2020). Likewise, code optimization can play an important role: Current NWP and climate models often have low arithmetic intensity and underutilize the computational power of modern supercomputers, but code restructuring techniques can significantly improve performance by alleviating memory bottlenecks and improving vectorization (Michalakes et al., 2016).

Another interesting alternative is machine learning (ML), which is currently a popular research topic in the context of physical modeling, as it has the potential to reduce key sources of uncertainty in dynamical models. ML algorithms such as deep neural networks (NNs) can learn complex nonlinear relationships from data, sidestepping any structural assumptions and simplifications. This may lead to, for instance, accurate convective or unified parameterizations by learning from cloud-resolving simulations (Brenowitz & Bretherton, 2018; Gentine et al., 2018; Rasp et al., 2018), improved bias correction of smartphone pressure observations (McNicholas & Mass, 2018), more skillful thunderstorm predictions by learning from lightning data (Ukkonen & Mäkelä, 2019), and many applications in remote sensing (Boukabara et al., 2019). NNs also have a key advantage in computational efficiency. The underlying matrix operations have been optimized for various kinds of hardware in external software libraries, enabling high performance across different platforms with little or no changes to code. NNs are particularly fast on accelerators such as Graphics Processing Units (GPUs), which are already being used for high-resolution weather simulations (Lapillonne et al., 2017) and superparameterized climate simulations (Hannah et al., 2020).

NNs have previously been used to emulate the entire radiation scheme in a dynamical model, with the outputs being the radiative fluxes and heating rates for all layers in an atmospheric column (Krasnopolsky et al., 2010; Pal et al., 2019). This approach has yielded considerable speed-ups of one (Pal et al., 2019) or several (Krasnopolsky et al., 2010) orders of magnitude compared to the original scheme. On the other hand, prognostic testing by Pal et al. (2019) revealed differences in radiative fluxes that were in some regions larger than the internal variability of the original scheme, and the differences in surface net fluxes reached 20 W m^{-2}. Another drawback is that the NN is tightly tied to the model configuration and has to be rebuilt when, for example, the vertical grid is changed. With some notable exceptions in idealized aquaplanet studies (Rasp et al., 2018), top-down ML approaches to subgrid physics have had issues pertaining to physical realism, numerical stability, and generalization. A key challenge is identifying an appropriate loss function, as minimizing the instantaneous error of a given variable does not ensure numerical stability over large time steps, realistic variability, or the conservation of energy, moisture, and momentum.

Here we explore a targeted approach, where conceptually different processes remain separated and physical equations are used where they are available. Modern radiation codes first compute the optical properties of the gaseous clear-sky atmosphere, aerosols, clouds, and surface and then compute the transfer of radiation through the atmosphere, often in a separate component known as the solver, by using the two-stream approximation. In this study we focus on the gas optics, which accounts for roughly a third of the runtime of one typical code (Hogan & Bozzo, 2018). Our aim is to accelerate a state-of-the-art radiation scheme for dynamical models by replacing the gas optics computations with NNs, while retaining accuracy for numerous applications such as numerical weather prediction and simulating past, present, and future climates. We also explore optimizations to remaining parts of the code which make it easier to exploit efficiencies obtained with the NNs.

To this end, we collect input data spanning a wide range of atmospheric conditions and greenhouse gas (GHG) concentrations and train NN on the data generated by the gas optics scheme RRTMGP. Efficient NN Fortran code for both CPUs and GPUs is implemented. The speed and accuracy of the new gas optics code, which is coupled to a refactored solver, is then evaluated against the original scheme using benchmark LBL computations.

2. RRTMGP

RRTMGP (RRTM for GCM applications-Parallel) is a newly developed package for predicting the optical properties of the gaseous atmosphere, freely available together with the radiative solver RTE (Radiative Transfer for Energetics). RTE+RRTMGP has been designed as an open-source code base for radiation calculations for dynamical models, including current and future numerical weather and climate models (Pincus et al., 2019). The toolbox, written in modern Fortran, aims to balance accuracy, efficiency, and flexibility in a modern software package which continues to evolve. Like other schemes, it separates solar "shortwave" (SW) from thermal LW radiation.

Where the scheme differs from less recent parameterizations is that the k-distribution is based on state-of-the-art spectroscopy and that it uses a high number of g-points; 256 within 16 LW bands (10–3,250 cm^{-1}) and 224 within 14 SW bands (820–50,000 cm^{-1}). (RRTMG, the predecessor to RRMTGP widely used in large-scale models, has 140 [LW] and 112 [SW] g-points.) As a result, the new scheme is more accurate than RRTMG but also slower in the LW by a factor of roughly 2.2 or 20% slower per g-point on one tested platform (Pincus et al., 2019). In the shortwave, the code is about twice as fast despite the higher spectral resolution.

The main computational kernel in RRTMGP is based on a linear 3-D interpolation of optical depth from values stored in a lookup table for various temperatures, pressures, and mixing fractions. The overlapping absorption of the two most absorptive gases in each band is treated via the parameter η, which is the relative mixing fraction of two *major* species which dominate the absorption in a given band. The lookup table values were determined by averaging output from an accurate LBL model, assuming atmospheres which contain only these two gases (dry air is used for bands with only one major species). The major gases in RRTMGP are H_2O, O_3, CO_2, CO_2, CH_4, and N_2O.

The contribution from other absorbing gases in a given band is treated more coarsely, with the tabulated values coming from LBL computations which include only this gas and a single reference pressure and the interpolated value for each of these *minor* gases added to the major gas value. Despite the simpler 2-D interpolation for minor gases, they can be more expensive than the major gas computations when looping over each minor gas in each band (in addition to inner loops over columns and layers), as RRTMGP supports up to 11 minor LW species such as CFC_{11} and CFC_{12} (Table A1 in Pincus et al., 2019).

Besides computing absorption optical depth for each layer, the LW code uses an interpolation routine to predict the Planck fraction, which is the fraction of the Planck function associated with each g-point in a given band. This is then multiplied with band-wise Planck functions (which depend on temperature) to output four emission variables used in RTE: Planck source functions at layer centers and the surface and upward and downward Planck source functions at levels (interfaces between layers). In the shortwave, Rayleigh scattering optical depths are interpolated from another table and combined with absorption optical depth to compute the single-scattering albedo and extinction optical depth. The optical properties of RTE+RRTMGP are defined on a 3-D grid (column, height, spectral).

3. RRTMGP-NN: A NN Emulation of RRTMGP

3.1. Background

NNs are a class of ML algorithms which map inputs to outputs by one or more layers of nodes—*neurons*—connected to each other by adjustable parameters and nonlinear functions. The input-output mapping therefore represents a series of adjustable nonlinear transformations. Mathematically, the transformation in each layer of a feedforward NN may be described as follows:

$$a_j = h\left(\sum_{i=1}^{D} w_{ji} x_i + w_{j0}\right) \tag{1}$$

where h is a nonlinear and differentiable function known as the activation function (e.g., a sigmoid function), $j = 1, \ldots, M$ and M is the number of neurons in the layer, w_{ji} are referred to as weights, and w_{j0} as biases. $x_1, \ldots x_D$ are the inputs to each layer, which for the first layer is the model inputs and for subsequent layers the outputs from previous layers. The outputs of the last layer are the model outputs $a_k = y_k$, where $k = 1, \ldots, K$ and K is the number of outputs.

The goal when training a network is to find a set of weights which minimize some measure of difference between the model output and training labels, such as root-mean square error. In theory any nonlinear mapping with finite discontinuities can be emulated with NNs, but larger models (with more layers and neurons) are needed for more difficult problems. Complex models are in turn more likely to suffer from overfitting, which means that the errors are small on the training data but large for new, unseen data. The ability to adapt to unseen data, generalization, is a key issue in problems such as ours with nonlinearity and a wide and high-dimensional input space. For a machine-learned radiation scheme to perform well in simulations of future climate, for example, it is important that future concentrations of GHGs and warmer and moister atmospheres are readily sampled during training, as NNs are unlikely to extrapolate beyond the trained input space with much skill.

3.2. Data

Our aim is to develop a model that can reproduce the full range of sensitivities of RRTMGP, which includes the sensitivity to a wide range of temperatures, pressure, and minor gases. In order to retain accuracy across such a wide range of states, we need a carefully constructed data set for training that is both broad and dense. In practice, our data set was expanded numerous times in a lengthy, iterative process, often after discovering a particular weakness in the model with respect to certain atmospheric conditions, GHG concentrations, and/or metrics such as heating rates or radiative forcings. The data we used came from the following:

- Data provided by the Radiative Forcing Model Intercomparison Project (RFMIP; see Pincus et al., 2016) and used in experiment *rad-irf*, consisting of 100 carefully chosen profiles from around the world and 18 experiments sampling different atmospheric conditions and GHG concentrations, such as present-day and future scenarios.
- CAMS (Inness et al., 2019) global reanalysis data for 00 and 12 UTC 1.2.2003, 1.7.2003, 1.2.2017, and 1.7.2017. Adjacent grid cells were left out, and random samples were drawn from what remained.
- To sample future climate, we obtained data for the years 2045 and 2100 derived from climate projections under the Shared Socioeconomic Pathways 2-4.5 and 5-8.5 (SSP2-4.5 and SSP5-8.5) in the CMIP6 archive. The data came from the Max Planck Institute Earth System Model Version 1.2 (Mauritsen et al., 2019).
- Forty-two atmospheric profiles (Garand et al., 2001) which were used by Pincus et al. (2019) to tune RRTMGP.
- Artificial profiles from the Correlated K-Distribution Model Intercomparison Project or CKDMIP (Hogan & Matricardi, 2020) designed to sample median, maximum, and minimum values of temperature, water vapor, and ozone (the "MMM" data set). We extended this data to also sample the mean, maximum, and minimum values of CO_2 and CH_4 found in RFMIP data, resulting in $3^5 = 243$ profiles instead of the original 3^3.

From each of these initial data sets, larger training data sets were created by extending the atmospheric profiles into tens to hundreds of different *experiments* where gas concentrations, and occasionally temperature and humidity, were varied in different ways. These experiments include 16 from the RFMIP protocol

(Tables 3 and 4 in Pincus et al., 2016) consisting mainly of preindustrial, present-day, or future values of specific or all GHGs; the two experiments with vertical dependent changes in atmospheric conditions ("future-all" and "preindustrial-all") were ignored as they are harder to apply to other data sets. We also created many new experiments inspired by RFMIP, such as perturbed temperature (up to -2 or +4K) while keeping relative humidity constant, and experiments where the concentration of individual gases was uniformly sampled, across different columns, between the minimum and maximum of RFMIP values. Most minor gases were missing in the original data sets, for such gases we again took guidance from RFMIP.

Using RFMIP as an example, we created 400 additional experiments to supplement the original 18. Many of these came from a Halton sequence, a design of experiments (DOE) method (Kocis & Whiten, 1997) similar to Latin hypercube sampling but deterministic and better at filling space uniformly in high-dimensional spaces. A Halton sequence of 140 samples—experiments—was generated with the DOEPY Python package (https://github.com/tirthajyoti/doepy) using all gases except water vapor and ozone, which were set to present-day values. This meant sampling a 14-dimensional cube. Although such samples contain unrealistic combinations of inputs, it may help the model learn the underlying physics and therefore improve generalization. Specifically, our aim was to present the NN with realistic conditions for current and future climate and also data with large variability so that the model may learn how individual gases contribute to the absorption across the spectrum.

In total, our extended data set consists of more than 7.5 million input-output pairs, sourced from roughly 200 000 atmospheric profiles with 20–60 vertical layers. The data were divided into training, validation, and testing subsets, using 15 randomly selected RFMIP profiles for testing, and the Garand data, supplemented with a random 2% of other data, for validation. The remaining data were used for training (roughly 90% of the whole, while the other subsets each make up 5%) and are comprised mostly of RFMIP and CAMS profiles.

3.3. Model Design

The inputs to the NNs are similar to those in the original scheme: temperature, pressure, and the concentrations of all gases represented in RRTMGP, excluding oxygen and nitrogen which are assumed to be constant with mole fractions of 0.209 and 0.781. This leads to seven inputs in the shortwave which only uses H_2O, CO_2, CH_4, O_3, and N_2O and a total of 18 inputs in the LW where many trace gases contribute to the absorption.

To choose what variables to predict, we follow the underlying kernels in RRTMGP to respect a physical separation of concerns. Separate models are trained to predict absorption optical depth and Planck fraction in the LW and absorption and Rayleigh optical depths in the shortwave. These are multi-output networks which predict all g-points simultaneously, so vectors of sizes 256 (LW) or 224 (SW). This is more efficient than predicting a single g-point or band at a time, despite a band approach having the benefit of reducing the number of inputs as some bands only have one or two contributing gases. RRTMGP treats the troposphere and stratosphere separately, using different lookup tables and sets of gases, which complicates the code and reduces efficiency. The NNs treat stratosphere/troposphere differences implicitly, and a single model predicts optical properties of arbitrary layers.

3.4. Model Training and Tuning

We developed NNs in Python using the high-level Keras library (https://keras.io) and the MXNet (https://mxnet.apache.org) back end. The final models are summarized in Table 1. In order to maximize accuracy, we tested many different optimizers, activation functions, model architectures, loss functions, and preprocessing methods. NN tuning is a laborious process and is often considered more art than science. We initially performed Bayesian optimization using the Hyperas wrapper around Hyperopt (Bergstra et al., 2015), but this quickly became too expensive as the training data grew larger. We then tuned our model by hand; this could mean computing the error in transmittance, but often we evaluated models more thoroughly by implementing them in RRTMGP and computing the flux errors with respect to LBL results for the original RFMIP dataset. Our main findings were as follows:

- RRTMGP computes absorption optical depth τ as the product of the absorption cross section k (m^2 mol^{-1}) and the path number of molecules in a column N (mol m^{-2}). We follow suit, normalizing τ by N before training, so that our models can support arbitrary vertical discretizations.
- Preprocessing both inputs and outputs was found to be critical for obtaining good results and making training faster. Some variables span many orders of magnitude and have a skewed distribution which may impede training. For these variables we found that that power scaling using the Nth square root

Table 1
Summary of the Different Models and Their Inputs and Outputs

Process	Predicted variable	Inputs	Neurons	Outputs	Output scaling
LW absorption	absorption cross section	18	58–58	256	$y = y^{\frac{1}{8}}; y_i = \frac{y_i - \bar{y}_i}{\sigma}$
LW emission	Planck fraction	18	16–16	256	$y = y^{\frac{1}{2}}$
SW absorption	absorption cross section	7	48–48	224	$y = y^{\frac{1}{8}}; y_i = \frac{y_i - \bar{y}_i}{\sigma}$
SW scattering	Rayleigh cross section	7	16–16	224	$y = y^{\frac{1}{8}}; y_i = \frac{y_i - \bar{y}_i}{\sigma}$

Note. All inputs were scaled to a range between 0 and 1. In the final column, y_i refers to the ith output, while σ is the standard deviation of all outputs. This variant of standardization was applied after taking the Nth root of the raw outputs.

was sufficient and computationally more efficient than log scaling (pressure was still log scaled). We transformed water vapor and ozone mixing ratios using $N = 4$, across sections using $N = 8$, and Planck fraction using $N = 2$. After this the inputs were scaled to 0–1.

- The absorption and Rayleigh cross sections were further normalized by subtracting each g-point with its mean and dividing by the standard deviation across all g-points, as described in Krasnopolsky (2013).
- When the outputs were normalized in this manner, standard loss functions such as the mean square error (MSE) and the mean absolute error (MAE) worked well. Before using normalization, we had relied on hybrid loss functions which explicitly computed the transmittance T in order to address the problem that simply minimizing the error in optical depth τ does not lead to accurate predictions of $T = \exp(-\tau)$ when the data are dominated by either small or large values of τ for which $T \approx 1$ and $T \approx 0$, respectively.
- After finding a reasonable architecture using Hyperopt, we manually tested larger and smaller networks with an emphasis on models with a constant number of hidden neurons due to these being more efficient to implement. We discovered that for all models besides LW absorption, 16 neurons in two hidden layers (16-16) were sufficient for obtaining accurate optical depths and fluxes. For predicting LW absorption, a more complex model with 50–60 neurons in two layers was needed.
- The soft sign activation $f(x) = \frac{x}{1+|x|}$ was associated with a lower loss than the widely used rectified linear unit (ReLU) while also being faster to compute than other well-performing functions such as the sigmoid function.

The NNs we trained were quite easily able to predict optical depth and transmittance with very high overall accuracy, with R squared (R^2) values larger than 0.99 and 0.999, respectively. For the Planck fractions R^2 was above 0.9995. Obtaining accurate fluxes proved more challenging, as even this level of accuracy in transmittance was not necessarily sufficient for predicting LW fluxes within 1 W m^{-2}. To obtain accurate fluxes, careful tuning was necessary, and our final models predict transmittance with an R^2 value of around 0.9995.

To avoid overfitting, we used early stopping, a regularization method which stops the training when the performance on a separate data set (the validation data) has no longer improved after a certain number of epochs. This resulted in accurate fluxes and heating rates for most RFMIP experiments. However, we later found a problem with unrealistic LW surface forcings by minor gases. This was mostly corrected by slightly increasing the size of the LW absorption model, producing more training data which targeted increased variability for these gases, and loosening the early stopping criteria to 20 epochs which led to substantially more epochs trained and lower losses. Our final LW absorption and emission models were trained for roughly 300–400 epochs, which took a few hours on a NVIDIA GTX 1070 GPU using a batch size of 1024. These models were trained with MSE loss at first, followed by another round using MAE after the early stopping condition was first met. The SW models were much easier to train: The early stopping criteria were reached sooner, and substantially less tuning was required to produce accurate fluxes.

4. Implementation and Code Optimization

With a NN the underlying computations for processing one atmospheric layer consist of a series of matrix-vector dot products, where the matrix is the NN weights and the initial vector is the input array, followed by an activation function and addition of biases. However, the fastest implementation collapses the vertical and horizontal dimensions into one dimension k and feeds the $N_x \times N_k$ array to the matrix-matrix

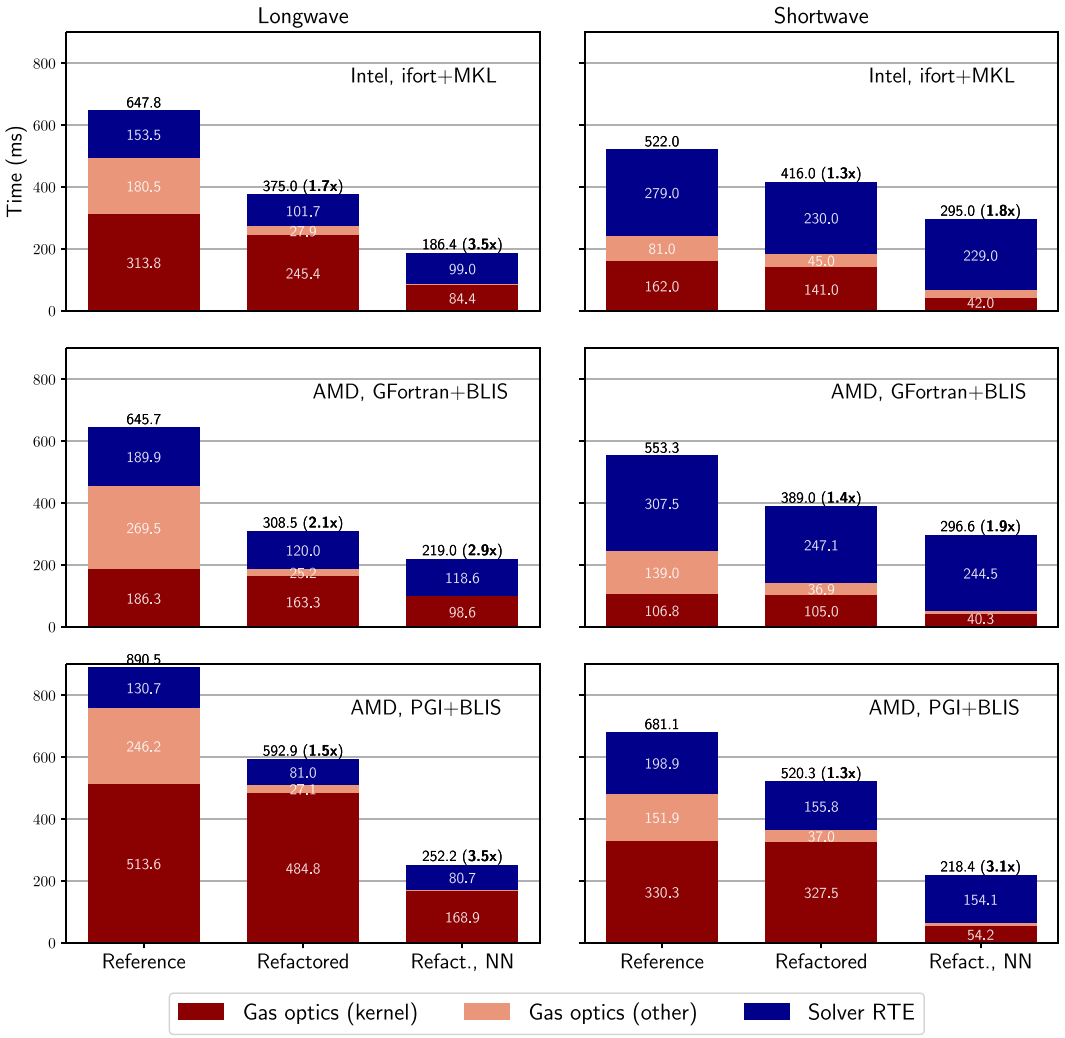

Figure 1. Mean elapsed time to compute the clear-sky longwave fluxes (left, without scattering) and shortwave fluxes (right, with scattering) for 1,600 RFMIP profiles. Top: Intel CPU, Intel Fortran compiler 19.1, and Intel MKL. Middle: AMD Ryzen CPU, GNU Fortran compiler 9.3, and AMD-Optimized BLIS 2.2. Bottom: AMD Ryzen CPU, PGI Fortran 19.10 compiler, and AMD-optimized BLIS. The gas optics is separated into the computational kernel and remaining parts which includes preprocessing and the transposing of arrays in the reference code. The postprocessing of neural network (NN) outputs is done inside the kernel. The number above columns indicates total runtime with speed-up in brackets. All code was run using single precision, a single CPU core, a block size of 32 columns, and the -O3 optimization flag, in addition to *'-march=native -funroll-loops –fast-math'* on Gfortran. To reduce the impact of noise, computations were repeated 10 times in an outer loop, and the best result of three tests was chosen. We used GPTL to profile the code.

multiplication routine in a BLAS library (GEMM). The kernel is implemented in single precision (using SGEMM), which is sufficient for NNs. For a fair comparison of computational cost we also ran the reference code in single precision. Note, though, that the two-stream solver used in the RTE shortwave code uses hard-coded floors for numerical stability in the two-stream calculation of layer reflectance and transmittance, which makes shortwave results for both implementations currently incorrect when run in single precision.

When timing the code, we discovered that any acceleration of the gas optics kernel would by itself have a modest impact on the time taken to compute fluxes. The radiative transfer solver RTE has columns as the fastest-varying dimension in the solver to let users be able to tune the problem size to the hardware at hand by processing a block of columns at a time. However, RRTMGP uses g-points as the fastest-varying dimension internally and transposes optical depth and source function arrays after they are computed. This transposition is an expensive operation and, depending on the hardware and compiler, can take as much

time as the actual computations. To maximize the benefit from accelerating the kernel with NNs, we refactored RTE to be consistent with RRTMGP in its array structure, resulting in all 3-D variables existing on a (spectral, height, and column) grid.

Further optimizations, with impacts on runtimes of tens of percent, are possible with some loss of generality. To reduce memory use, for example, the source functions at g-points can be computed within a column loop in the solver from Planck fractions and source functions by band, as opposed to allocating large 3-D arrays for the spectral source functions at layers and levels and passing these to the solver. Performing the spectral reduction inside a column loop also reduces memory use, so we implemented this common use case as an optional feature. Similarly, some loops in the LW code could be merged (like the computation of source terms within the downward transport loop), thereby reducing memory accesses by iterating over arrays on fewer occasions. On modern computers, computational inefficiency often stems from memory bottlenecks which cause the processor to be underutilized.

Figure 1 compares the computational performance of RTE+RRTMGP, the refactored code, and the refactored code with NNs, when using a block size of 32 which usually resulted in the lowest total runtime with RTE+RRTMGP. On an Intel platform, computing clear-sky LW fluxes for RFMIP profiles were 1.7 times faster with the refactored code and 3.5 times faster when additionally using NNs. The NN kernel itself was nearly 3 times faster than the lookup table method when using all minor gases. In the shortwave, the gas optics had a similar speed-up, but a relatively more expensive solver whose runtime was often dominated by computations of the exponential function meant that the overall speed-up was smaller. One exception to this was the PGI compiler, where the total runtime decreased by a factor of 3.1 due to poor performance with RRTMGP. In general, the speed-up depends greatly on the hardware, compiler, and the BLAS library. On the AMD platform, using GNU Fortran with aggressive compiler options, the NN was only 65% faster than the original LW kernel. While slower than the Intel Math Kernel Library in our tests, we found BLIS (Van Zee & van de Geijn, 2015) to be faster than some other open-source BLAS libraries. The reason NNs are faster than the original code is principally due to high performance of BLAS; while the NNs use more floating-point operations than the original kernel (roughly 4 times more in the LW code), the number of floating-point operations per second is drastically higher (Figure A1). Most of these operations are in the last NN layer, where the inner dimensions of the matrix multiplication have the shape $N_{gpt} \times N_{neurons} = 256 \times 58$ for the LW absorption model. Reducing the number of spectral points or neurons could make the code much faster still, as we found to be the case when using smaller models with BLIS.

The refactored code was substantially faster on all tested platforms. On the other hand, many of the optimizations are specific to the task of computing broadband fluxes and to RRTMGP's specific representation of the Planck source function and are associated with trade-offs. In particular, the in-line computation of Planck sources in the solver breaks the separation of concerns between RRTMGP and RTE, while the in-line broadband integration—in itself reducing the solvers runtime by up to 20%—leads to repeated code and hence trades some simplicity for efficiency. This highlights how it can be difficult to balance flexibility, simplicity, and efficiency in scientific code, particularly when performance is conditioned on many other factors. The only clear lesson is that transposing large arrays is expensive and should be avoided if possible. Appendix A explores the performance in more detail with an emphasis on the impact of block size, which is the innermost dimension in RTE.

Our accelerated radiation code for dynamical models, RTE+RRTMGP-NN, is like its parent code written in modern Fortran and uses object-oriented programming to provide flexibility and to separate computations from flow control. The NN models are loaded from data files specified at runtime, similar to the k-distribution in RRTMGP. The NN code is implemented as its own Fortran class, which we based on neural Fortran (Curcic, 2019) but wrote optimized kernels for which use GEMM. The postprocessing of outputs is done inside these kernels, but the preparation of inputs is delegated to a routine in the gas optics module. The currently implemented models use all RRTMGP gases as input. Gases not provided by the user are by default assumed to be zero but can also be specified to use a reference scalar concentration such as preindustrial, present day, or future.

5. Accuracy

We investigate the accuracy of RRTMGP-NN by implementing all models and comparing the resulting fluxes and heating rates to accurate LBL results alongside the original scheme. We use as our reference fluxes

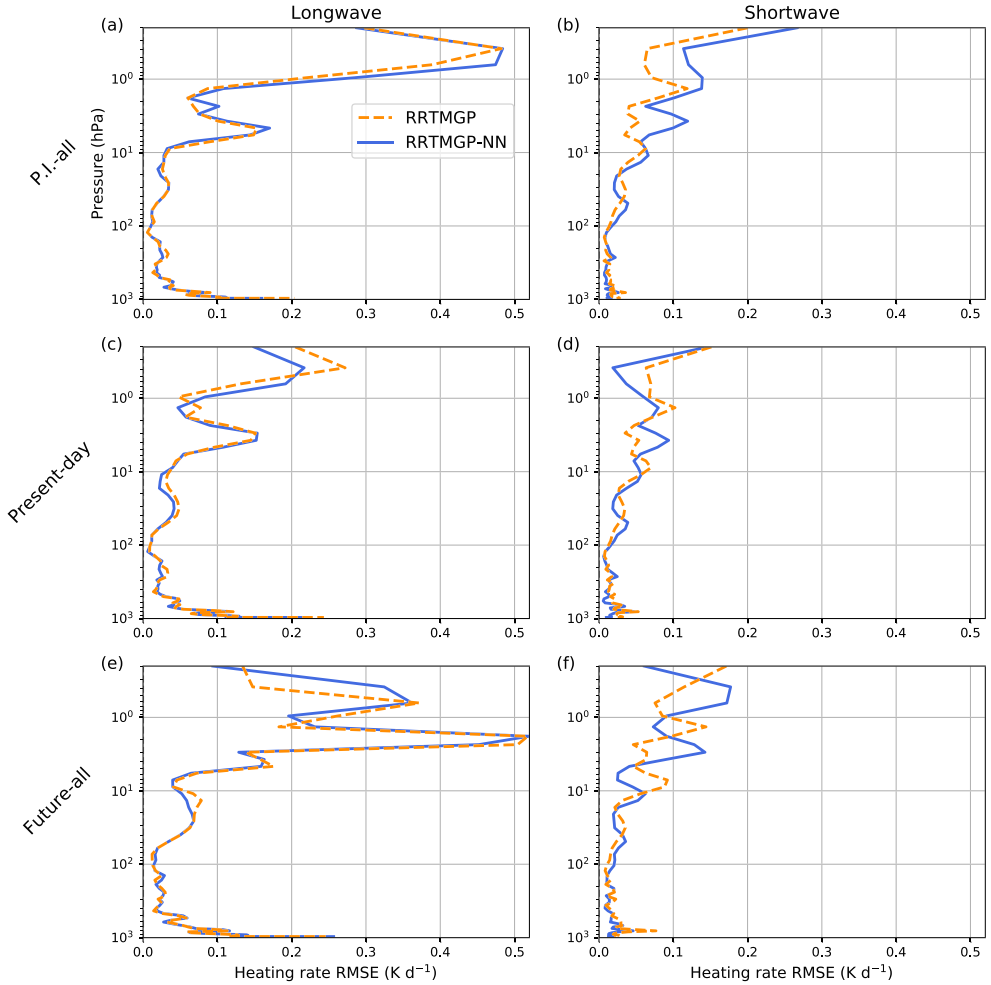

Figure 2. Root-mean-square errors in longwave (a, c, e) and shortwave (b, d, f) heating rates shown for both RRTMGP and RRTMGP-NN using 15 test profiles and 3 different experiments from RFMIP: preindustrial-all (a, b), present day (c, d), and future-all (e, f). The "all" suffix refers to perturbed temperature and humidity in addition to perturbed gas concentrations. The errors were computed relative to the LBLRTM line-by-line model using the 15 RFMIP test profiles.

computed by LBLRTM 12.8 (Clough et al., 2005), the model on which RRTMGP was trained, obtaining results for the RFMIP examples from the Earth System Grid (Pincus et al., 2020). Heating rate errors averaged over 15 RFMIP profiles set aside for testing are first shown in Figure 2 for three different experiments. The error profiles for RRTMGP and RRTMGP-NN are very similar and virtually identical in the LW. Only in the upper atmosphere, where the errors are larger, can the curves be clearly discerned. This is an indication that RRTMGP-NN matches the original scheme very closely.

Figure 3 depicts the top-of-atmosphere (TOA) radiative forcing between preindustrial and future RFMIP experiments. The forcing predicted by RRTMGP-NN across different sites is highly accurate and again almost indistinguishable from RRTMGP. As expected, the differences between the two schemes are smaller than the errors with respect to LBL results, but the latter are also very small. RRTMGP-NN predicted TOA fluxes with substantially smaller RMSE and biases in some experiments (not shown), in particular future and future-all, where RRTMGP has a bias of -0.31 and -0.53 W m^{-2} but RRTMGP-NN only -0.05 and -0.21 W m^{-2}. RRTMGP generally had a smaller bias and RMSE in other experiments, particularly present-day and preindustrial (PI) RFMIP experiments such as PI CO_2 and PI-all, but the global TOA flux biases and RMSE of RRTMGP-NN did not exceed 0.3 and 0.41 W m^{-2}, respectively.

Net surface fluxes are also predicted accurately. In the future-all experiment, RRTMGP-NN performs particularly well, with a lower RMSE and bias than RRTMGP (Figure 4). The performance is similar for training and test profiles, which suggests that the models have not been overfitted to the training data. This was expected given the diverse training data and use of early stopping.

For a fully independent evaluation, we have participated in CKDMIP with RRTMGP-NN. The purpose of CKDMIP is to evaluate current CKD models using benchmark LBL calculations and explore how accuracy varies with the number of *g*-points and other choices in how CKD models are generated. This should be a more difficult test for our model since CKDMIP only includes water vapor, ozone, methane, CFC12, and an artificially increased CFC11 concentration to represent 38 further GHGs (CFC11-equivalent). Such CKDMIP-style experiments were sampled in only a small portion of our training data.

In the CKDMIP evaluation, RRTMGP-NN performs again very similar to RRTMGP, with the differences generally being smaller than those between RTE+RRTMGP and ecRAD+RRTMG. RRTGMP-NN had slightly more accurate upwelling fluxes at TOA than RRTMGP in three of the four experiments. While it may seem curious that the NN would perform better than the scheme it was trained on, we consider this a lucky accident. In our tests a single training epoch at the end of training could make the difference between better or worse net fluxes for RFMIP data compared to RRTMGP. The two codes had virtually the same heating rate RMSE in all CKDMIP scenarios (preindustrial, present day, future, and Last Glacial Maximum).

RRTMGP-NN is, unfortunately, notably worse with respect to the sensitivity of surface net LW fluxes to changes in some individual gas concentrations (Figure 5). The surface forcings for N_2O, CFC11-equivalent, and CFC12 deviate from LBL results, while RRTMGP represents the forcings of all CKDMIP gases well. Obtaining accurate surface forcings for minor gases with NNs turned out to be challenging. Sensitivity tests showed that the surface forcing errors varied considerably from one training epoch to the next. Using custom loss functions to minimize such errors remains to be explored. We also did not test using other regularization methods besides early stopping, since we do not attribute the issue to overfitting (such forcing errors were also found for training and validation data). Finally, the TOA forcings are generally quite accurate and excellent for carbon dioxide and methane.

6. Fast and Flexible Models

Our NN emulation of a recently developed gas optics scheme is considerably faster than the original kernel which uses a lookup table, while retaining high accuracy across diverse atmospheric states. Comparing our work to previous studies in literature, these top-down ML emulations of model radiation have led to larger speed-ups but at the cost of accuracy and generalization. For instance, the surface net fluxes in Pal et al. (2019) deviated from the original scheme by up to 20 W m^{-2} in a prognostic validation with a dynamical model. In earlier work by Krasnopolsky et al. (2010), the differences between NN-emulated full radiation and the original scheme were more reasonable, with root-mean-square errors in LW heating rates reaching 0.8 K day^{-1}.

Our work has important parallels with that of Veerman et al. (2020). However, the authors used a much smaller range of atmospheric conditions and only a few gases, targeting large-eddy simulations and NWP. They obtained downwelling LW flux errors within 0.5 W m^{-2} with respect to RRTMGP, again indicating that a targeted approach which retains the radiative transfer equation can yield high accuracy.

We have found that optical depths and Planck function can be predicted accurately across a wide range of atmospheres using fairly small NNs of around 5,000–20,000 parameters, as long as great care is taken in preparing and preprocessing data and tuning the model. The models we trained have a similar complexity to those in Veerman et al. (2020), who had a much narrower focus and only included water vapor and ozone. We briefly tested using a smaller set of input gases and did not find any clear improvement in accuracy, generalization, or required model size, possibly because most of the complexity in atmospheric absorption comes from water vapor. Computationally, the number of inputs is in itself unimportant, since most of the floating-point operations are in the last NN layer which outputs a large array. This suggests that NNs are an efficient way to include a large number of gases when computing the radiative transfer for climate modeling applications.

NNs are also attractive because they can make efficient use of specialized hardware like GPUs. We have developed an initial GPU implementation of our NN gas optics parameterization using cuBLAS and OpenACC and find speed-ups, relative to an OpenACC implementation of RTE+RRTMGP, comparable to the CPU results.

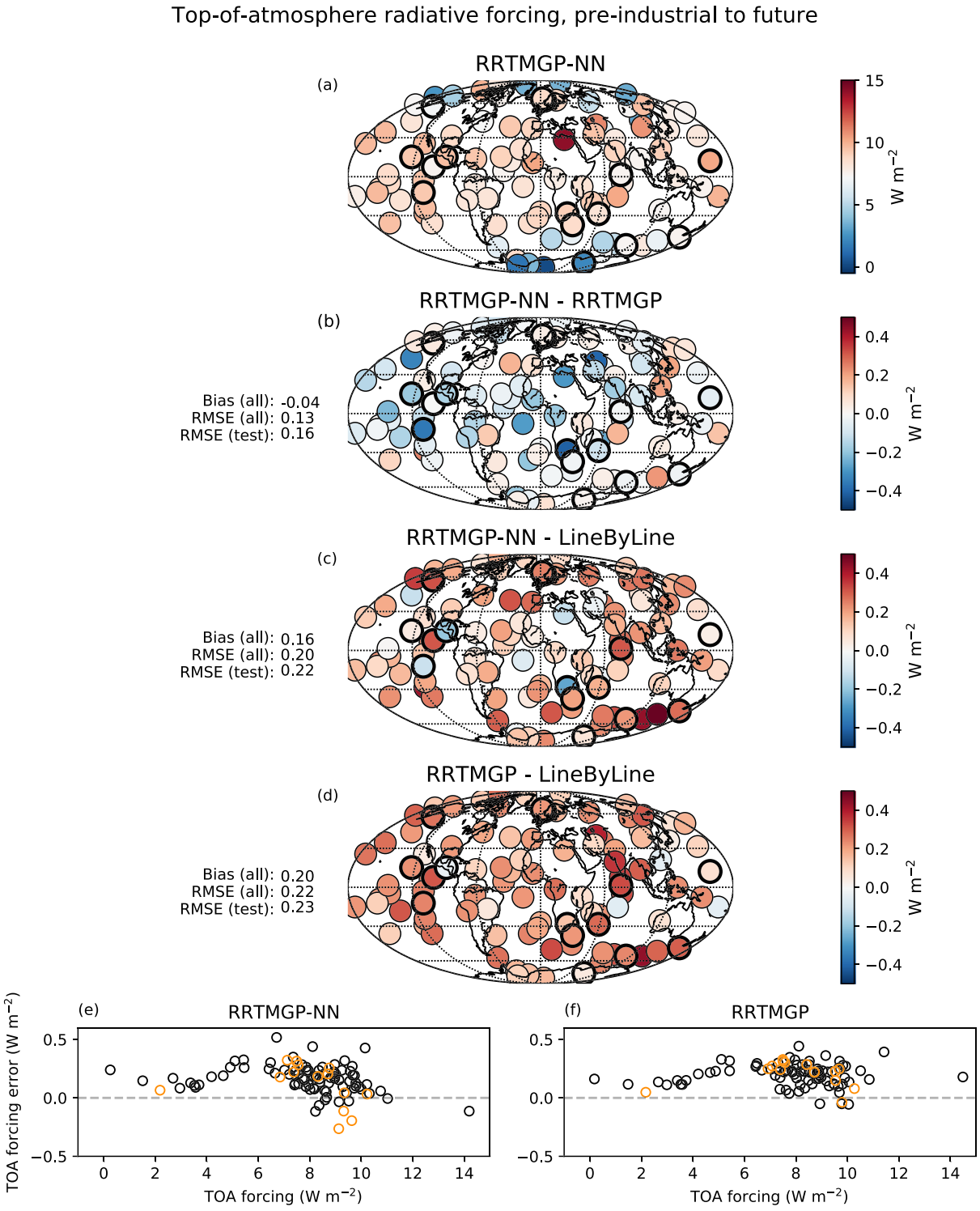

Figure 3. Instantaneous radiative forcing (IRF) i.e. the change in net fluxes at top of atmosphere between preindustrial and future RFMIP experiments for NN (a) and the difference in IRF between the NN and RRTMGP (b), the NN and LBL (c), and RRTMGP and LBL (d). The 15 test profiles are indicated with a bolded circle. Below the main figure a scatterplot of the forcings against errors is shown for the NN (e) and RRTMGP (f) with test profiles in orange.

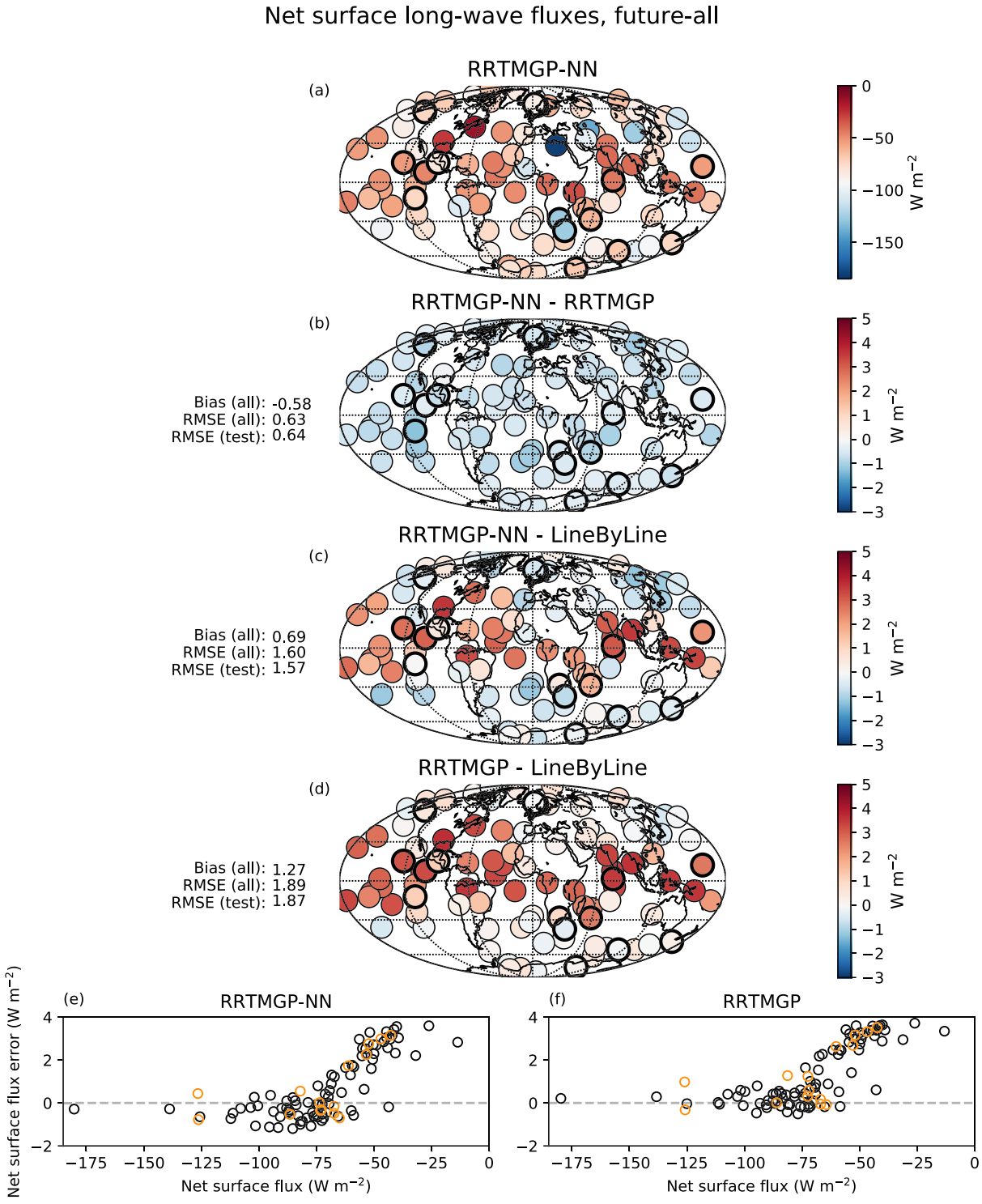

Figure 4. As in Figure 3 but for the net longwave fluxes at surface for the "future-all" experiment.

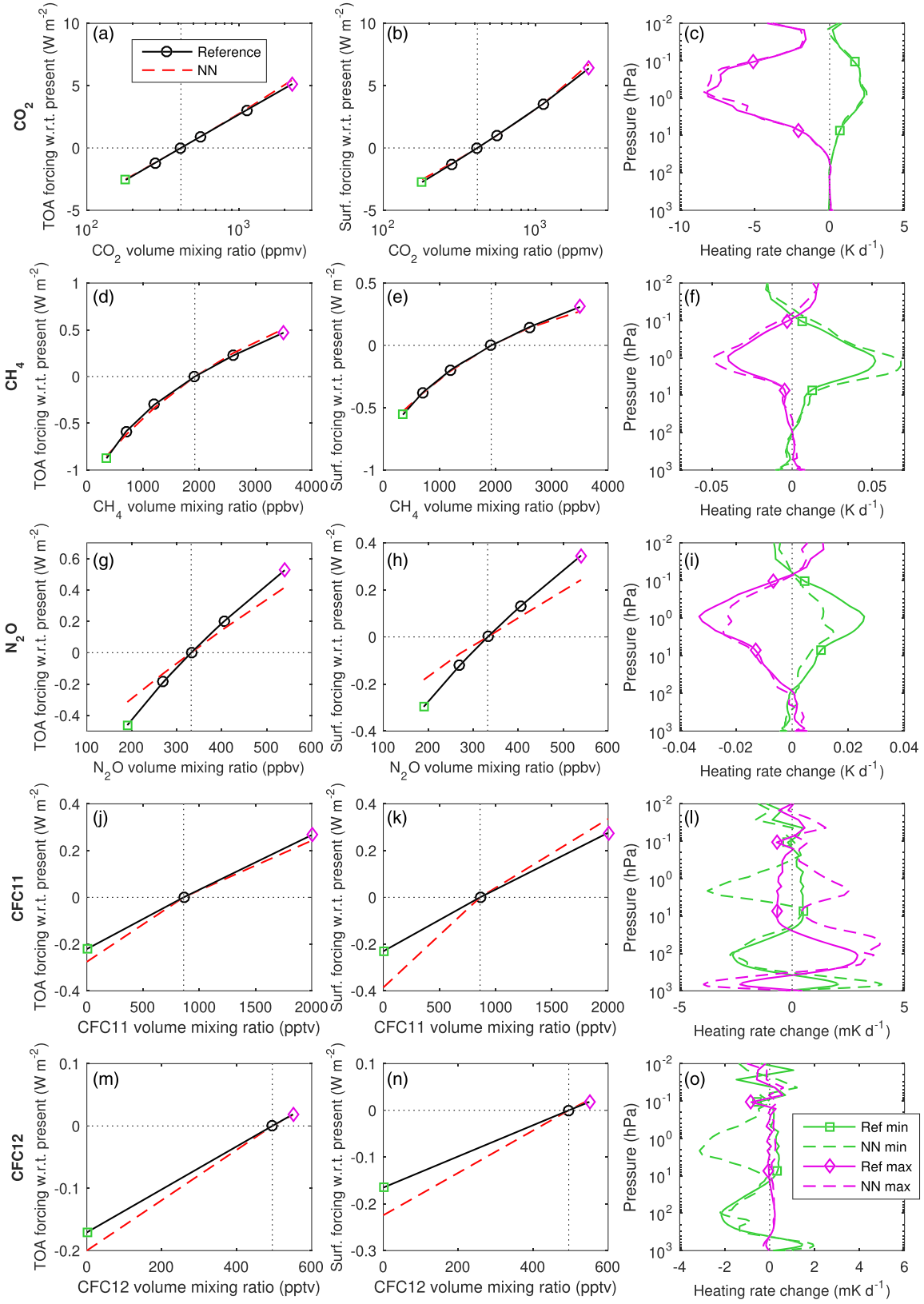

Figure 5. (a–l) Comparison of RRTMGP-NN and reference line-by-line computations of instantaneous radiative forcing at top of atmosphere and the surface when perturbing the concentrations of individual well-mixed greenhouse gases from their present-day values, averaged over 50 profiles in the Evaluation-1 CKDMIP data set. For the minimum and maximum concentrations, the change to the mean atmospheric heating rate is also shown.

We spent a considerable time optimizing the remaining parts of RRTMGP+RTE. These efforts have in common a focus on accelerating calculations; both will depend on the software and hardware platform on which they are deployed. Though NNs inherit the structural assumptions and simplifications of the schemes on which they are trained, these need not be made explicit: In our example, there is no direct dependence on the η parameter used to account for the spectrally overlapping absorption of a gas mixture. It may be possible to omit such assumptions altogether by training directly on k-distributions sourced from LBL data, in effect creating a NN-based CKD model which is capable of treating gas overlap implicitly. However, this would require an extreme LBL modeling effort.

Due to their flexibility and nonlinearity, NNs have the promise to improve physics parameterizations, with the added benefit of high performance across different processors and particularly on accelerators. For some problems it is possible to design models that remain rooted in fundamental physics; such models can then be accurate and efficient as well as interpretable and physically consistent. Separating physical concerns, as was done here for the radiative transfer and gas optics problem, may yield good results elsewhere, too.

Appendix A: Which Dimension Order for Radiation Calculations?

RTE uses columns as the fastest-varying dimension so that users may tune the problem size B to their hardware, enabled by outer loops over $N_{blocks} = \frac{N_{columns}}{B}$. Figure A1 shows how computational performance changes with B for the original and refactored shortwave codes. The only difference between the shortwave

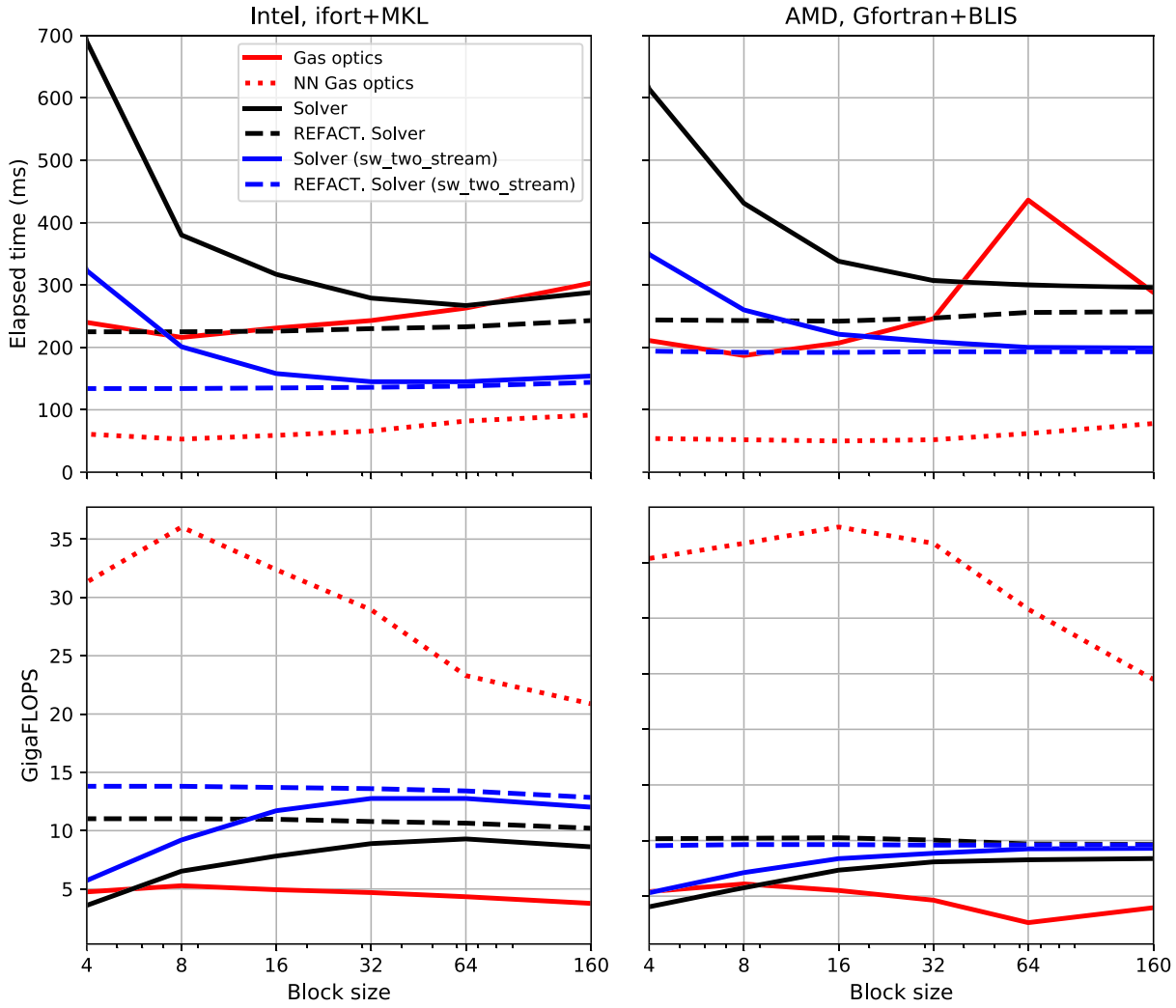

Figure A1. As in Figure 1, but showing the impact of block size on the elapsed time (top row) and floating-point operations per second (bottom row) of the shortwave codes for the Intel (left) and AMD platform (right). NN = neural network, REFACT = refactored.

solvers is the in-line computation of broadband fluxes and that the refactored solver uses a first-dimension size of 224 (the number of *g*-points in the RRTMGP *k*-distribution).

The top row of Figure A1 shows the time to solution as a function of B. The reference solver displays poor performance for very small values of B. A likely explanation for this is short inner loops of length B which inhibit efficient instruction-level parallelism and vectorization. (The number of calls to subroutines, given by $N_{blocks} \times N_{g-points}$, also becomes large, but the overhead from this was only significant at $B = 4$). Most of the time in the shortwave calculation is taken in the two-stream calculation of layer reflectance and transmittance (subroutine *sw-two-stream*) which is nearly constant for block sizes above 16. A total run-time difference of 20–25% remains between the solvers due to the inline broadband flux summation and the greater efficiency of this computation when *g*-points are the innermost dimension (reduction operations being faster for contiguous memory). Inlining the broadband flux computation is only possible when columns are outermost and avoids allocation of 3-D flux variables.

The bottom row shows the calculation rate in billions of floating-point operations per second (FLOPS). The NNs achieve 6–7 times more FLOPS than the lookup table method, resulting in faster run times despite having more computations. FLOPS peak around $B = 8$ and drop considerably after this, which may be attributable to better cache use for smaller data blocks.

In general B had limited impact on runtime for $B > 16$, although performance of the solvers degrades somewhat for first-dimension sizes which are not multiples of 16 (not shown). Our experiments take the number of *g*-points, which for RRTMGP is both large and a multiple of 16, as given. A solver using *g*-points as the first dimension would be subject to some of the same performance trade-offs for very small numbers of *g*-points.

We note that these tests were carried out on somewhat modern commodity CPUs (AMD Ryzen 2600 and Intel i5-8250U) but may show some dependence on specific hardware.

Data Availability Statement

RTE+RRTMGP-NN is available on Github (https://github.com/peterukk/rte-rrtmgp-nn); this manuscript was produced with the version archived online (https://doi.org/10.5281/zenodo.4029138). Scripts and data used in this paper are available online (https://doi.org/10.5281/zenodo.3909653).

Acknowledgments
The authors thank Menno Veerman for useful communication which helped us avoid overlapping efforts. We acknowledge the financial support from the European Commission for the project: "Energy-efficient SCalable Algorithms for weather and climate Prediction at Exascale" (ESCAPE-2) Horizon 2020 project, which has enabled us to do this work (grant agreement ID 800897). R. P. is grateful for support from the U.S. Department of Energy Office of Biological and Environmental Research (Award DE-SC0016593).

References

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. (2015). Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Boukabara, S.-A., Krasnopolsky, V., Stewart, J. Q., Maddy, E. S., Shahroudi, N., & Hoffman, R. N. (2019). Leveraging modern artificial intelligence for remote sensing and NWP: Benefits and challenges. *Bulletin of the American Meteorological Society*, *100*, ES473–ES491.

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*, 6289–6298. https://doi.org/10.1029/2018GL078510

Clough, S. A., Shephard, M. W., Mlawer, E. J., Delamere, J. S., Iacono, M. J., Cady-Pereira, K., et al. (2005). Atmospheric radiative transfer modeling: A summary of the AER codes. *Journal of Quantitative Spectroscopy & Radiative Transfer*, *91*(2), 233–244. https://doi.org/10.1016/j.jqsrt.2004.05.058

Cotronei, A., & Slawig, T. (2020). Single-precision arithmetic in ECHAM radiation reduces runtime and energy consumption. *Geoscientific Model Development*, *13*(6), 2783–2804. https://doi.org/10.5194/gmd-13-2783-2020

Curcic, M. (2019). A parallel Fortran framework for neural networks and deep learning. *ACM SIGPLAN Fortran Forum*, *38*(1), 4–21. https://doi.org/10.1145/3323057.3323059

Fu, Q., & Liou, K. N. (1992). On the correlated k-distribution method for radiative transfer in nonhomogeneous atmospheres. *Journal of the Atmospheric Sciences*, *49*(22), 2139–2156.

Garand, L., Turner, D. S., Larocque, M., Bates, J., Boukabara, S., Brunel, P., et al. (2001). Radiance and jacobian intercomparison of radiative transfer models applied to HIRS and AMSU channels. *Journal of Geophysical Research*, *106*(D20), 24,017–24,031. https://doi.org/10.1029/2000JD000184

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*, 5742–5751. https://doi.org/10.1029/2018GL078202

Goody, R., West, R., Chen, L., & Crisp, D. (1989). The correlated-k method for radiation calculations in nonhomogeneous atmospheres. *Journal of Quantitative Spectroscopy and Radiative Transfer*, *42*(6), 539–550.

Hannah, W. M., Jones, C. R., Hillman, B. R., Norman, M. R., Bader, D. C., Taylor, M. A., et al. (2020). Initial results from the super-parameterized E3SM. *Journal of Advances in Modeling Earth Systems*, *12*, e2019MS001863. https://doi.org/10.1029/2019MS001863

Hogan, R. J. (2010). The full-spectrum correlated-k method for longwave atmospheric radiative transfer using an effective planck function. *Journal of the atmospheric sciences*, *67*(6), 2086–2100.

Hogan, R. J., & Bozzo, A. (2018). A flexible and efficient radiation scheme for the ECMWF model. *Journal of Advances in Modeling Earth Systems*, *10*, 1990–2008. https://doi.org/10.1029/2018MS001364

Hogan, R. J., & Matricardi, M. (2020). Evaluating and improving the treatment of gases in radiation schemes: The correlated K-Distribution model intercomparison project (CKDMIP). *Geoscientific Model Development Discussions*, *2020*, 1–29. https://doi.org/10.5194/gmd-2020-99

Iacono, M. J., Delamere, J. S., Mlawer, E. J., Shephard, M. W., Clough, S. A., & Collins, W. D. (2008). Radiative forcing by long-lived greenhouse gases: Calculations with the AER radiative transfer models. *Journal of Geophysical Research*, *113*, D13103. https://doi.org/10.1029/2008JD009944

Inness, A., Ades, M., Agusti-Panareda, A., Barré, J., Benedictow, A., Blechschmidt, A.-M., et al. (2019). The CAMS reanalysis of atmospheric composition. *Atmospheric Chemistry and Physics*, *19*(6), 3515–3556.

Kocis, L., & Whiten, W. J. (1997). Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)*, *23*(2), 266–294.

Krasnopolsky, V. M. (2013). *The application of neural networks in the Earth system sciences: Neural networks emulations for complex multidimensional mappings*. Dordrecht: Springer.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., Hou, Y. T., Lord, S. J., & Belochitski, A. A. (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.

Lapillonne, X., Osterried, K., & Fuhrer, O. (2017). Using OpenACC to port large legacy climate and weather modeling code to GPUs. In *Parallel Programming with OpenACC* (pp. 267–290). Boston: Morgan Kaufmann.

Mandal, M., Mohanty, U. C., & Raman, S. (2004). A study on the impact of parameterization of physical processes on prediction of tropical cyclones over the bay of bengal with NCAR/PSU mesoscale model. *Natural Hazards*, *31*(2), 391–414.

Markowski, P. M., & Harrington, J. Y. (2005). A simulation of a supercell thunderstorm with emulated radiative cooling beneath the anvil. *Journal of the atmospheric sciences*, *62*(7), 2607–2617.

Mauritsen, T., Bader, J., Becker, T., Behrens, J., Bittner, M., Brokopf, R., et al. (2019). Developments in the MPI-M Earth system model version 1.2 (MPI-ESM1. 2) and its response to increasing CO2. *Journal of Advances in Modeling Earth Systems*, *11*, 998–1038. https://doi.org/10.1029/2018MS001400

McNicholas, C., & Mass, C. F. (2018). Smartphone pressure collection and bias correction using machine learning. *Journal of Atmospheric and Oceanic Technology*, *35*(3), 523–540.

Michalakes, J., Iacono, M. J., & Jessup, E. R. (2016). Optimizing weather model radiative transfer physics for intel's many integrated core (MIC) architecture. *Parallel Processing Letters*, *26*(04), 1650019.

Pal, A., Mahajan, S., & Norman, M. R. (2019). Using deep neural networks as Cost-Effective surrogate models for Super-Parameterized E3SM radiative transfer. *Geophysical Research Letters*, *46*, 6069–6079. https://doi.org/10.1029/2018GL081646

Pincus, R., Buehler, S. A., Brath, M., Crevoisier, C., Jamil, O., Evans, K. F., et al. (2020). Benchmark calculations of radiative forcing by greenhouse gases. *Journal of Geophysical Research: Atmospheres*, *125*, e2020JD033483. https://doi.org/10.1029/2020JD033483

Pincus, R., Forster, P. M., & Stevens, B. (2016). The radiative forcing model intercomparison project (RFMIP): Experimental protocol for CMIP6. *Geoscientific Model Development*, *9*(9), 3447–3460. https://doi.org/10.5194/gmd-9-3447-2016

Pincus, R., Mlawer, E., & Delamere, J. (2019). Balancing accuracy, efficiency, and flexibility in radiation calculations for dynamical models. *Journal of Advances in Modeling Earth Systems*, *11*, 3074–3089. https://doi.org/10.1029/2019MS001621

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, *115*(39), 9684–9689.

Shepherd, T. G., Polichtchouk, I., Hogan, R. J., & Simmons, A. J. (2018). Report on stratosphere task force: ECMWF. https://doi.org/10.21957/0vkp0t1xx

Ukkonen, P., & Mäkelä, A. (2019). Evaluation of machine learning classifiers for predicting deep convection. *Journal of Advances in Modeling Earth Systems*, *11*, 1784–1802. https://doi.org/10.1029/2018MS001561

Van Zee, F. G., & van de Geijn, R. A. (2015). BLIS: A framework for rapidly instantiating BLAS functionality. *ACM Transactions on Mathematical Software*, *41*(3), 14:1–14:33. https://doi.org/10.1145/2764454

Veerman, M. A., Pincus, R., Stoffer, R., van Leeuwen, C., Podareanu, D., & van Heerwaarden, C. C. (2020). Predicting atmospheric optical properties for radiative transfer computations using neural networks. *Philosophical Transactions A*. https://doi.org/10.1098/rsta.2020.0095