Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer

Peter Ukkonen[1,2]

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

Key Points:

- Feed-forward and recurrent neural networks (NN) were developed to emulate a shortwave radiation scheme, as well as its components
- The recurrent NN has far better accuracy than usual approaches, while offering a significant speedup especially on GPUs
- Using NNs for gas optics is 3 times faster and does not sacrifice accuracy

Corresponding author: Peter Ukkonen, `peter.ukkonen@nbi.ku.dk`

Abstract

Machine learning (ML) parameterizations of subgrid physics is a growing research area. A key question is whether traditional ML methods such as feed-forward neural networks (FNNs) are better suited for representing only specific processes. Radiation schemes are an interesting example, because they compute radiative flows through the atmosphere using well-established physical equations. The sequential aspect of the problem implies that FNNs may not be well-suited for it.

This study explores whether emulating the entire radiation scheme is more difficult than its components without vertical dependencies. FNNs were trained to replace a shortwave radiation scheme, its gas optics component, and its reflectance-transmittance computations. In addition, a novel recurrent NN (RNN) method was developed to structurally incorporate the vertical dependence and sequential nature of radiation computations.

It is found that a bidirectional RNN with an order of magnitude fewer model parameters than FNN is considerably more accurate, while offering a smaller but still significant 4-fold speedup over the original code on CPUs, and a much greater speedup on GPUs. The RNN predicts fluxes with less than 1% error, and heating rates computed from fluxes have a root-mean-square-error of 0.16 K day^{-1} in offline tests using a year of global data. Finally, FNNs emulating gas optics are very accurate while being several times faster. As with RNNs emulating radiative transfer, the smaller dimensionality may be crucial for developing models that are general enough to be used as parameterizations.

Plain Language Summary

Numerical weather and climate simulations are being performed at increasingly resolution, making the energy cost of simulations significant. Computing how solar and terrestrial radiation interact with Earth's atmosphere, surface, and clouds is one of the most computationally expensive parts in climate models especially. This has invited efforts to replace these computations with predictions from a neural network, which is approximative but considerably faster than physical radiation computations.

In this paper, different ways of emulating a radiation code with neural networks have been explored. Its main contribution is developing a novel emulation method based on recurrent neural networks, which more closely resemble physical radiative transfer computations. The accuracy is found to be considerably higher than with traditional neural network approaches which use an order of magnitude more model parameters.

1 Introduction

Climate and weather simulations are being performed at increasingly high resolutions. The implications for energy use are significant: even with an atmospheric model fully ported to a state-of-the-art GPU supercomputer, kilometer-scale global simulations consume 596 MWh energy per simulated year (Fuhrer et al., 2018). This is the same as the yearly electricity consumption of 161 average EU households in 2018 (Odyssee-Mure, 2021). For the energy costs of earth system simulations not to become untenable, both hardware and and algorithmic improvements are needed.

An algorithmic development which could improve both the accuracy and computational efficiency of weather and climate simulations is the use of machine learning (ML) methods to represent sub-grid diabatic processes. Recent years has seen an influx of papers on this topic where the typical approach has been training neural networks (NNs) or random forests on coarse-grained data from cloud-resolving or high-resolution simulations, and representing all sub-grid processes with a single model (Rasp et al., 2018;

Brenowitz & Bretherton, 2018; Gentine et al., 2018; Yuval et al., 2021). Results have in many cases been promising: NN-parameterized simulations have shown to reproduce several features of high-resolution simulations not found in coarse-resolution ones (Rasp et al., 2018; Gentine et al., 2018). Issues with instability, model drift or energy conservation have been widely reported, but also overcome; for instance by using loss functions or model architectures which incorporate conservation laws (Beucler et al., 2019, 2021). This is in itself an impressive feat considering the challenge of the problem. However, all of these simulations have used highly idealized aquaplanet setups. It has yet to be demonstrated that unified NN parameterizations can improve realistic climate simulations, which are much more complex and require reliable predictions across different climates.

One of the most time-consuming components in coarse-resolution simulations (50% in ECHAM) is the radiation scheme. This has led to attempts to replace the entire radiation scheme with machine learning models, the outputs being column radiative fluxes and/or heating rates (radiation was also included in the subgrid physics emulation in many of the aforementioned studies). Impressive speed-ups (1-2 orders of magnitude) relative to the physical parameterization have been obtained using this approach, but it is unclear if the accuracy and reliability is sufficient for state-of-the-art numerical weather prediction (NWP) and climate simulations. For instance, surface fluxes deviated by 10 Wm^{-2} from the reference simulation in a prognostic evaluation with a climate model (Pal et al., 2019). Recently, Song and Roh (2021) developed NNs to emulate a radiation scheme in a regional NWP setting. In offline tests with independent data, predicted shortwave radiation had a root-mean-square-error of roughly 0.2 K day^{-1} in heating rates and 20 Wm^{-2} in fluxes.

Although these differences seem large compared to parameterization errors for clear-sky radiation (Hogan & Matricardi, 2020, Figure 5,7), they are less so relative to the noise caused by Monte Carlo Independent Column Approximation (McICA) which represents cloud sub-grid cloud variability stochastically and is used in many climate and weather models (Räisänen et al., 2005). Stable climate simulations incorporating ML have been demonstrated in several studies, with the differences in prognostic tests being similar to the models internal variability (Krasnopolsky et al., 2010). Yet again, a realistic climate is not sufficient evidence for accuracy. Detailed evaluation of fluxes and heating rates across the whole atmosphere using fully independent data is rarely presented. Heating rates are particularly prone to errors in the upper stratosphere: Yuval and OGorman (2020) emulated subgrid tendencies from specific processes and found ML predictions of radiative heating rates in upper layers to be poor, and had to use the original parameterization above 11.8 km, while Yuval et al. (2021) made the cut-off at 13.8 km.

An as alternative to emulating the entire radiation code, one can use ML for predicting optical properties while still computing fluxes using a traditional solver. This may be easier from a physical and algorithmic perspective since the former relies on empirical methods (look-up table interpolation) and has no dependency between adjacent atmospheric layers, while the latter requires solving the radiative transfer equations to compute radiative flows through an atmospheric column. (Here a parallel can be drawn to the dynamical or "resolved" part of large-scale models, since they both rely on solving well-established physical equations to compute flows). Ukkonen et al. (2020) and Veerman et al. (2021) demonstrated high accuracy of NN-predicted optical properties of the gaseous atmosphere, which were 3-4 times faster than the original RRTMGP gas optics scheme. In the former study the NN gas optics were combined with a refactored radiative transfer solver to speed up clear-sky flux computations by a factor of 2-3, while the fluxes and heating rates were almost identical to the original scheme when evaluated against line-by-line radiation computations.

While NN methods are powerful algorithms capable of modeling complex relationships, it is not clear that regular feed-forward neural networks (FNNs) are algorithmi-

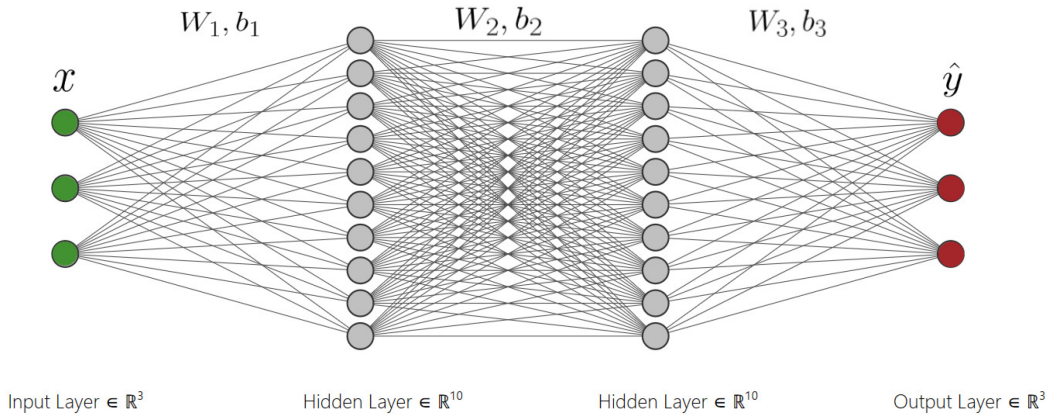

Figure 1: Feed-forward neural networks, shown to illustrate the potential algorithmic issue with using them to model radiative transfer as is commonly done by stacking the vertical profiles of input variables (such as temperature and pressure) into one FNN input column. Because the variables or nodes are only connected horizontally to nodes in other layers, the vertical dependencies between atmospheric layers (Variable 1, Variable 2...) can only be represented indirectly through the horizontal connections (weights) to shared nodes in one or more hidden layers. Information does not propagate directly in the vertical direction, as it does in radiative transfer equations. Figure adapted from Aldakheel et al. (2021).

cally well-suited for radiative transfer problems which involve computing radiative flows between mediums. In the case of radiation parameterizations used in weather and climate models, radiative flows in a column are computed layer by layer, requiring several iterations through a column. Emulating a radiation scheme by stacking vertical profiles of several variables into a single input column of a machine learning model, and predicting profiles of fluxes and/or heating rates as a single output column, means that information needs to propagate between different inputs or nodes corresponding to adjacent layers (as in the physical equations). This does not occur "directly" in an FNN where nodes are connected horizontally to nodes in other layers, but not vertically to nodes in the same layer (Figure 1). These vertical dependencies can of course be represented via the weights connected to at least one hidden layer, but it is unclear how this can be done accurately with simple neural network architectures.

In many ways, the results obtained in previous studies are impressive, as not only does the NN approach skip explicit layer-to-layer computations, but also explicit spectral computations. Radiation codes have evolved for many decades, and the current state-of-the-art is to combine the two-stream approximation to the one-dimensional radiative transfer equation (Meador & Weaver, 1980) with the correlated-k-distribution (CKD) method (e.g., Goody et al., 1989) for the spectral integration. CKD can accurately resolve broadband fluxes (i.e. fluxes integrated over the electromagnetic spectrum, which relate to heating rates) while reducing the number of monochromatic computations by many orders of magnitude compared to line-by-line methods. If it was true that an NN could reduce the problem further by several orders of magnitude, not incorporate any physical laws, and still be accurate and reliable, this would essentially mean that current parameterizations include wasted computations.

In this study, we aim to shed some light on the suitability of neural networks to replace radiation computations by addressing the following research questions:

1) Can FNNs closely emulate an entire radiation scheme, i.e. directly predict fluxes or heating rates with similar accuracy to existing parameterizations?

2) Is it easier to predict fluxes and heating rates accurately by only emulating computations without a vertical dependency, such as gas optics or reflectance-transmittance computations using FNNs?

3) Do recurrent neural networks (RNNs) which structurally incorporate the vertical dependence of radiation computations better emulate radiative transfer then FNNs?

4) How does the trade-off between efficiency and accuracy vary across the different emulation strategies?

To help answer these questions, we train neural nets to emulate: A. the entire radiation scheme (gas optics and radiative transfer combined), B. gas optics, and C. the reflectance-transmittance computations in the solver. Method A, which maps atmospheric conditions to fluxes or radiative heating rates, has been used in several papers but here a novel method based on RNNs is developed and compared to the standard approach using FNNs.

These emulation strategies are then compared in terms of accuracy and generalization through offline validation with independent profiles, acquired from reanalysis data, that span a wide range of atmospheric conditions. Since the goal is to evaluate how well simple neural networks can emulate complex radiative transfer computations, this paper restricts itself to shortwave computations accounting for clouds, where the need to consider scattering results in a much harder problem. Generation of training data, model implementation, and verification is carried out using the recently developed RTE+RRTMGP radiation scheme (Pincus et al., 2019).

Below, we introduce the data and codes (Section 2), followed by an overview of the different emulation strategies and associated machine learning methodologies (Section 3). The results in terms of accuracy and speed-up are then presented (Section 4) and discussed in the context of previous literature (Section 5). Finally, conclusions are given in Section 6.

2 Data and codes

2.1 Data

Global CAMS reanalysis (Inness et al., 2019) data was acquired for 2009-2018, with 4 dates and 2 times of day (03 and 15 UTC 1.2, 1.5, 1.8 and 1.11) in order to encompass seasonal and diurnal variability of atmospheric fields. Model level variables consist of temperature, pressure, cloud liquid water and ice mixing ratio, and mixing ratios of five gases that are radiatively active in the shortwave: water vapor, ozone, carbon dioxide, methane and nitrous oxide. The radiation computations also account for oxygen and nitrogen, but these are assumed constant (with mole fractions of 0.209 and 0.781, respectively) and therefore not included in NN inputs. The gases correspond to all the gas species considered by RRTMGP-SW, with the exception of nitrogen dioxide which was not available in CAMS. The single-level variables obtained were surface pressure, 2-metre temperature, and forecast albedo. We also computed true solar zenith angles for the purpose of model evaluation, but when generating training data, the solar angle of each column was assigned a random value between 0 and 90. The total solar irradiance at top of atmosphere is assumed constant at 1412 Wm^{-2}.

To avoid over-representation of polar regions in the training data, the CAMS data was interpolated from a longitude-latitude grid to a global 320 km resolution triangular grid as specified for the ICON model (Zängl et al., 2015), while keeping the the original vertical grid of 60 layers (top at 10 Pa). Each year consists of $5120 \times 8 = 40960$ columns. Data was partitioned into validation (the year 2014), testing (2015, in which 09 and 21 UTC data was additionally included), and training (remaining 8 years in 2009-2018) sub-

sets. It may be noted that the evaluation data does not sample new climate conditions, but given the high variability and dimensionality of fields associated with column-wise radiation computations, even one "in-sample" testing year should give some indication of model generalization.

The amount of training samples depends on the emulation method (Table 1). When training an emulator for the whole radiation scheme, the model inputs are columns of atmospheric variables, resulting in $40960 \times 8 = 327680$ training samples. For training other models, which take input variables defined at a single spectral and/or atmospheric layer, the potential training data is enormous, especially for the reflectance-transmittance model which operates on individual g-points. In this case, we extract random samples from the data, limiting the number of training samples to roughly 50 million (reflectance-transmittance) or 10 million (gas optics).

2.2 RTE+RRTMGP

RTE+RRTMGP (Pincus et al., 2019) is a recently developed radiation scheme for dynamical models combining two codes: Radiative Transfer for Energetics (RTE), which computes fluxes given a description of boundary conditions, source functions and optical properties of the atmosphere, and RRTM for General circulation model applicationsParallel (RRTMGP), which computes optical properties and source functions of the gaseous atmosphere. The combined package can be used to compute broadband radiative fluxes from input profiles of temperature, pressure and gas concentrations. The gas optics scheme RRTMGP uses a k-distribution based on state-of-the-art spectroscopy, and has 256 g-points in the longwave and 224 g-points in the shortwave, which is high compared to many other schemes. RRTMGP continues to evolve and preliminary reduced-resolution k-distributions with roughly half the number of g-points (similar to the predecessor code RRTMG), was available at the time of writing, but in this study the original 224 g-point model is used. When profiling code this should favour the approach of emulating the entire radiation scheme, as this method avoids explicit g-point computations while the runtime of the original code (as well as emulators of components) is proportional to number of g-points. Indeed, reducing the number of g-points, for instance by using full-spectrum correlated-k-methods, is a promising way to improve the accuracy/speed trade-off in radiation schemes (Hogan, 2010).

2.3 RTE+RRTMGP-NN

In this work, a refactored version of RTE+RRTMGP developed in tandem with NN emulators for RRTMGP (Ukkonen et al., 2020) is used in order to utilize existing NN code infrastructure and to get a more meaningful measure of the speedup given by emulators. The refactored version (RTE+RRTMGP-NN) has columns as the outermost dimension in both RRTMGP and RTE and therefore avoids expensive array transposes, and also features smaller efficiency optimizations such as an optional inlining of the broadband flux computation inside a column loop for reduced memory use. (This feature was at the time of writing available in RTE+RRTMGP).

The NN inference and I/O code in RTE+RRTGMP-NN is based on neural-Fortran (Curcic, 2019) but has been optimized for efficiency by packing (or re-interpreting using pointers, when possible) the data into batches, resulting in the core operations - multiplying layer weights with inputs - becoming a matrix-matrix multiplication that is delegated to a BLAS library. Other changes include fusing the activation and bias additions, as well as GPU support based on OpenACC directives and the NVIDIA cuBLAS library. The end result is a highly efficient Fortran implementation of feed-forward neural networks that can be used in production code.

The data generation workflow consisted of acquiring reanalysis data, pre-processing it into yearly netCDF files that can be read by RTE+RRTMGP (for instance, gas mixing ratios were converted to mole fractions), and performing shortwave cloudy-sky computations which account for gases and clouds (but not aerosols) to generate the input-output pairs for machine learning. The radiation computations account for scattering, and cloud optical properties were generated with a cloud optics extension in RTE+RRTMGP that is based on Mie calculations. Aerosols are not included; for the purpose of evaluating the ability of NNs to emulate the physical radiation code, this should not be important as the aerosol optical properties are simply added to the optical properties from clouds and gases. The NNs were designed and trained in Tensorflow (`https://www.tensorflow.org`) using the Keras front-end (`https://keras.io`), but PyTorch (`https://pytorch.org`) code was also written to facilitate further research. A Python script was used to convert the Keras models into ASCII files from which neural-Fortran loads the model weights.

3 Emulation strategies

3.1 FNN-RadScheme - emulation of the full radiation scheme using feed-forward neural networks

Emulating the full radiation scheme is the best approach from the perspective of efficiency, since explicit layer-to-layer computations as well as spectral computations can be avoided. Internally, the radiation scheme computes many intermediate variables with a higher dimensionality than the parameterization input and outputs: first RRTMGP computes gas optical properties (optical depth and single-scattering albedo) at each g-point and model level. The cloud optical properties (optical depth, single-scattering albedo, and asymmetry parameter) are then generated for each spectral band and model level and added to the gas optical properties. The radiative solver takes the optical properties and boundary conditions (incoming solar flux, zenith angle, and surface albedo) and performs radiative transfer computations for each g-point, resulting in upward and downward fluxes $F_{\downarrow}$, $F_{\uparrow}$ (total flux, given by diffuse plus direct flux) and direct shortwave fluxes $F_{\downarrow,dir}$, $F_{\uparrow,dir}$ for each g-point and model half-level. Finally, broadband fluxes are obtained $F_{\downarrow}$, $F_{\uparrow}$ by summing the spectral fluxes together. In the NN approach, the broadband fluxes are predicted directly from profiles of gas and cloud mixing ratios. This is very efficient, but assumes that the spectral and vertical dependencies can be represented by the NN mapping.

RTE+RRTMGP was used to generate output downward and upward flux profiles from profiles of gas concentrations, temperature, pressure, cloud ice and water mixing ratios, as well as the scalar variables surface albedo and cosine of the solar zenith angle. The NN outputs in this study consist only of downward and upward fluxes, and is smaller compared to other studies. Direct downward flux is omitted; while this variable would likely be needed in the host model, its computation is more straightforward and it's not needed for heating rates, and therefore less interesting for NN emulation.

Earlier studies (Krasnopolsky et al., 2010; Roh & Song, 2020; Pal et al., 2019) have predicted heating rates (HR) profiles directly as NN output, often omitting prediction of flux profiles completely and instead adding scalar flux variables at the surface and top-of-atmosphere as additional NN outputs (Krasnopolsky et al., 2010; Roh & Song, 2020). Here we test predicting fluxes, while HR is given by the vertical divergence of net fluxes at each model layer i as in physical radiation codes:

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

where $F_{\mathrm{i+1/2,\ SW}}$ is the difference between the downward and upward SW fluxes at the interface between model layers i and $i+1$, c_p is the specific heat a constant pressure, g is the gravitational constant and $\frac{\partial T}{\partial t}$ is the rate of temperature change.

Computing heating rates from fluxes ensures physical consistency and energy conservation Yuval et al. (2021). On the other hand, it can result in large errors in HR because NN-predicted fluxes tend to be noisy and HR are very sensitive to the vertical gradient in fluxes, especially in the stratosphere where pressure is low. The problem can be alleviated by taking special care in the NN design. Firstly, normalizing the fluxes by the downward direct flux at the top layer of each column (incoming flux multiplied with the cosine of the solar zenith angle) is found to reduce errors in fluxes. Effectively this physically re-scales the output values to a range between 0 and 1. Although in some cases the flux at a lower layer can exceed the incoming flux (Jiang et al., 2005), the training data only had a handful of values above 1. Therefore the flux scaling is combined with a sigmoid activation in the output layer to constrain outputs within the 0-1 range, which was found to reduce errors. Second, a custom loss function can be used to explicitly minimize the error in both flux and heating rates:

$$loss = \alpha(y - y_{pred})^2 + (1-\alpha)(HR - HR_{pred})^2,$$

where y is the target value (scaled flux), y_{pred} is the NN output, HR is the heating rate computed using equation (1), and α is a manually tuned coefficient controlling how much heating rates are weighted relative to fluxes. In practice, the benefit from using a hybrid loss function was limited by the heating rates being very noisy when not predicted directly, and the sensitivity of computed HR to flux errors in the upper atmosphere. This issue with noisy heating rates when predicting fluxes, manifesting in large swings in the training losses (not shown), seems to be specific to FNNs as it was not seen with RNNs (section 3.2).

Figure 2 compares the flux and heating rate errors for models using different predictands and scaling methods. Included in the comparison is a model which predicts heating rates profiles directly in addition to fluxes at the boundaries, as in Krasnopolsky et al. (2010); Roh and Song (2020). Heating rate errors are much smaller using this method. However, adding the full flux profiles as output in addition to heating rate profiles (182 outputs in total) led to very poor predictions at the surface in quick tests with models with up to 256 neurons in two layers (not shown). To avoid the issue with physically inconsistent heating rates and fluxes at the boundaries, and to allow an equal footing with other emulation strategies (sections 3.2 - 3.4) which can all produce flux profiles, method c) in Figure 2 is used in the final evaluation despite the larger heating rate errors. This may be a questionable choice, but for operational implementation the conservation of energy is important, and is only guaranteed when predicting fluxes and computing heating rates from those (the incoming solar radiation will then be equal to the energy heating the atmosphere and the surface).

The FNNs have 128 neurons in three hidden layers. Two hidden layers produced only slightly worse results, but a model with a single hidden layer and 128-192 neurons had substantially larger errors.

3.2 RNN-RadScheme - emulation of the full radiation scheme using bidirectional recurrent neural networks

While the feed-forward neural network can predict heating rate profiles and scalar fluxes reasonably well, on paper it still appears ill-suited for predicting radiative flows due to the lack of inter-node connections in a NN layer. The FNN approach also has the drawback of being tied to vertical resolution of the training data, as the number of inputs and outputs are fixed. A type of NN which can avoid this problem is found in the

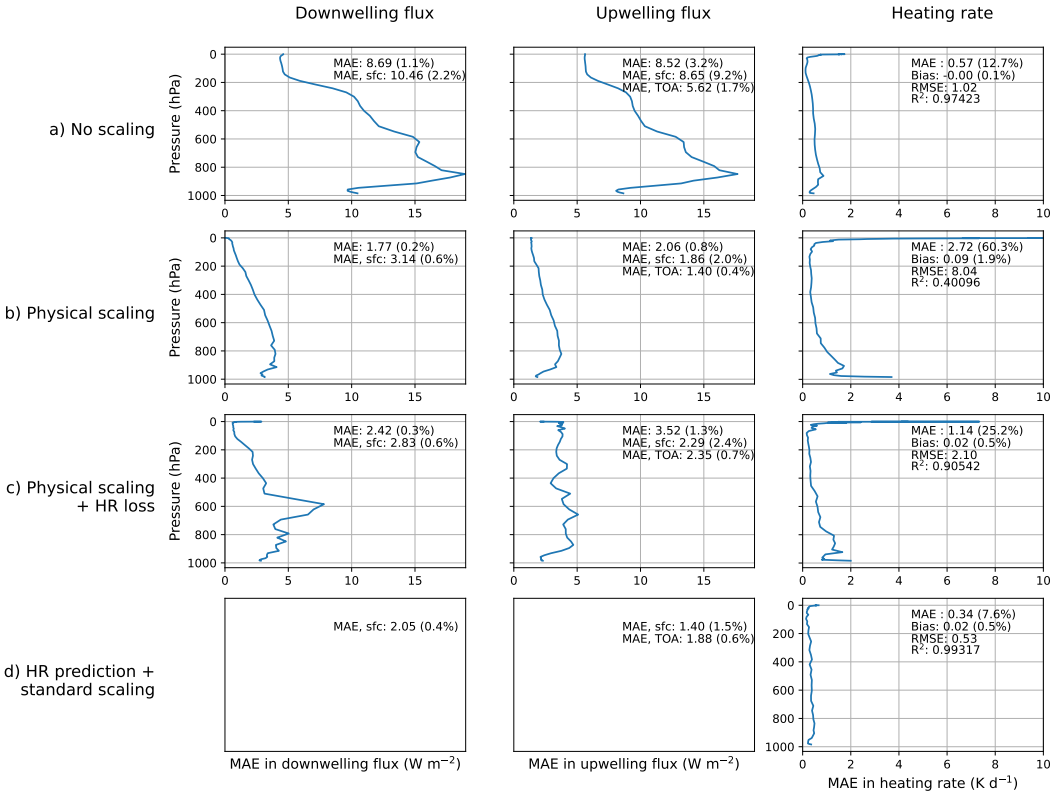

Figure 2: Impact of scaling, loss function and predictand on the vertical profiles of mean absolute error in downwelling flux (left column), upwelling flux (middle column) and heating rate (right column) for the validation data from 2014 with randomly sampled solar zenith angles. The outputs of the different feed-forward NN models are unnormalized fluxes (a), fluxes scaled by the incoming flux (b-c), and heating rate profiles plus three flux scalar variables (d). Adding heating rate to the loss function is helpful when predicting scaled fluxes (c); with a regular loss function (b) the heating rate errors reach up to 20 K day^{-1} at the top of atmosphere (the x-axis has been cropped at 10 K day^{-1}). All fluxes are total (direct + diffuse) shortwave fluxes. Overall mean absolute error (MAE) is annotated, with the number in parenthesis indicating the MAE value as a percentage of the column and layer mean of the variables, which only have positive values for physically computed SW radiation. When testing each method, three separate NNs were trained, and the results with smallest heating rate errors were saved and compared.

recurrent neural network (RNN), in which connections form a directed graph. RNNs are usually applied to problems associated with temporal sequences. A RNN layer takes the input at a given sequence, updates its internal state, and then processes the next point in the sequence. The sequential nature is not present in an FNN where the output of one layer forms the input of a separate NN layer with different weights. The internal state allows the RNN to have memory so that prior inputs, i.e. from earlier in time when dealing with a temporal problem, can influence the current prediction.

This idea can be exploited for radiative transfer by letting the sequence be represented by vertical levels. However, a basic RNN is not appropriate because the radiative fluxes at a given level depend not only on conditions at the levels above but also on the levels below. Fortunately, information can propagate from future states in a bidirectional RNN (BiRNN). A BiRNN comprises of two RNNs of opposite directions connected to the same output, meaning that one RNN begins from the beginning of the sequence and moves in the positive direction, while the other begins from the end of the sequence and moves in the negative direction. A single BiRNN layer approach, as illustrated in Figure 3 was tested. In this method the input for a given atmospheric layer is used to predict the scaled downward flux at the bottom of this layer (the next half-level) as well as the upward flux at the top of the layer (the previous half-level). Two output variables then remain; the downward flux at the top and upward flux above the surface. The first of these is actually an input and used here for scaling the fluxes, while the latter can be physically computed from the downward flux above the surface times the surface albedo.

The above approach is elegant, but requires the albedo to be a broadband quantity. This happens to be true for the data used here, but may not be a valid assumption generally. Furthermore, the inconsistency in how upward fluxes are computed led to larger heating rate errors at the surface for a BiRNN model which otherwise performed well (not shown). To remedy these issues, the model structure can be refined to output the full flux profile at layer interfaces ($nlay + 1$), despite the inputs being defined at layers ($nlay$). One way of achieving this is by concatenating layer-wise RNN outputs with the output from a dense NN layer, which takes as input the albedo(s) and/or other surface quantities. This more complex approach is illustrated in Figure 4. A third RNN layer, where the information propagates downward, has also been added; this was found to work better than just two RNNs (one BiRNN). The structure in Figure 4 was inspired by the physical equations in the radiative transfer solver, and resembles them quite closely. Three vertical iterations are used there, too: one to compute direct downwelling flux starting from top-of-atmosphere, one starting from the surface and computing the albedos at each level using the adding-doubling method (Hansen, 1971), and a final downward pass from the top-of-atmosphere to compute upward and downward fluxes.

While three vertical iterations within the NN model reduce the potential for speedup, on the other hand the number of hidden neurons needed for accurate results is very small. Here we evaluate a model using only 16 neurons in each of the three RNN layers. For the RNN layers, we use gated recurrent units (GRU), which are more complex than simple RNN layers. A GRU layer consists of an "update gate" and a "reset gate". Here the former decides if the cell state should be updated with the past (accumulated) state or not, while the reset gate allows the network to forget past information. It is not clear how these mechanisms specifically benefit radiative transfer, but they have been found to alleviate problems with vanishing gradients by allowing information to be passed without going through a nonlinear activity, thus helping preserve information from earlier states. For radiation such information could relate to optical properties, or reflectances and transmittances, as computed in prior states. In practice, GRU layers gave substantially better results than simple RNN layers.

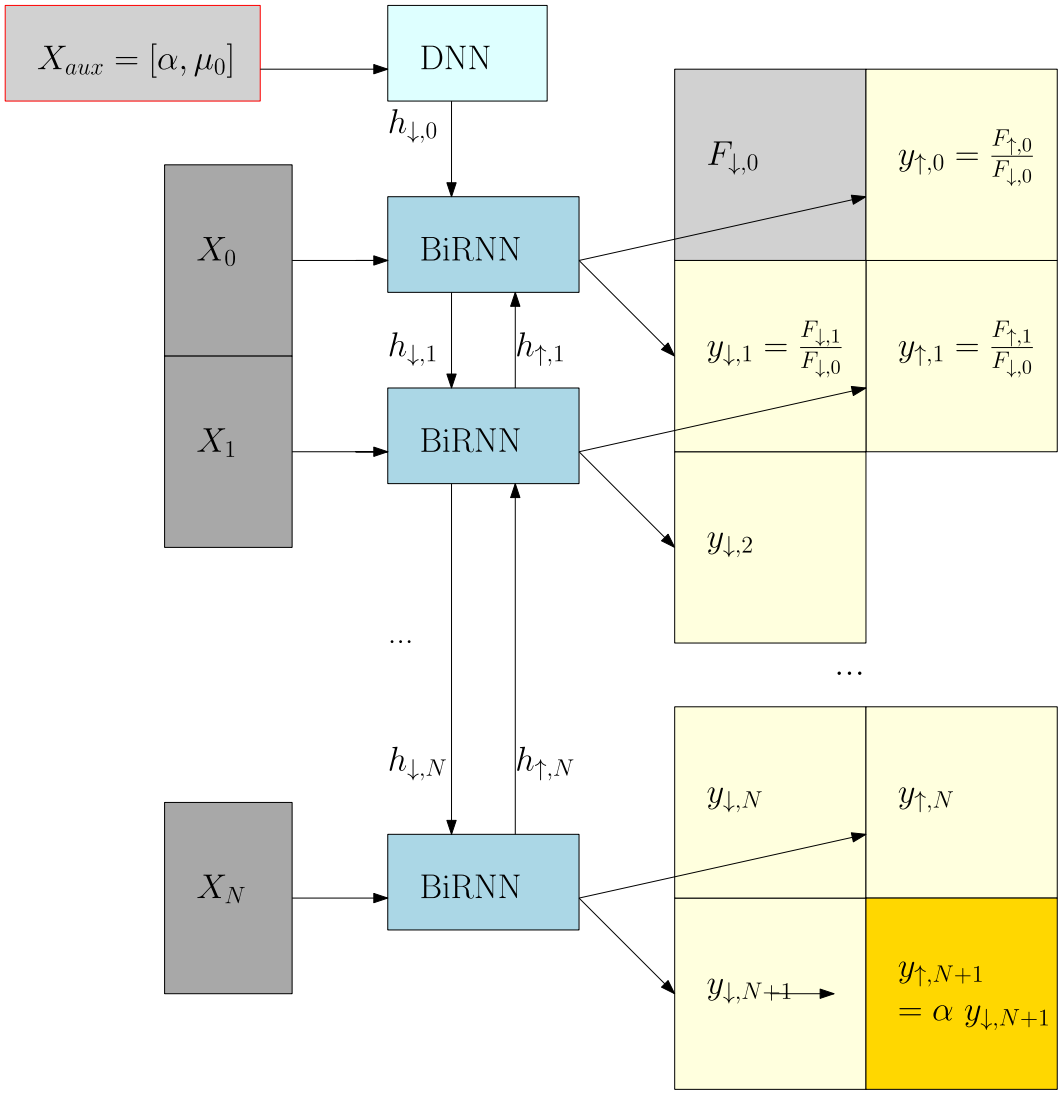

Figure 3: A RNN-based approach to predicting radiative fluxes. Input variables defined at N model layers ($X_0, X_1 ... X_N$) form the sequential input to the Bidirectional RNN (BiRNN), while the output consist of two scalar values: the (scaled) upward flux at the upper layer boundary (the $N+1$ layer boundaries are referred to as *levels*), and downward flux at the lower boundary. Note that the figure shows the *unrolled* network structure; there is actually just one BiRNN layer which forms a directed graph to itself by saving a hidden state h; or two hidden states $h_\downarrow, h_\uparrow$ in the case of the BiRNN which internally comprises of a forward and backward RNN (not shown). The auxiliary scalar inputs, albedo α and cosine of the solar zenith angle μ, are incorporated through a dense layer (DNN) which predicts the initial states of the BiRNN $h_{0,\downarrow}, h_{N,\uparrow}$. The diagram depicts input variables in gray, output variables in light yellow, and NN layers in light blue. The upward flux near the surface (dark yellow) is not an NN output but computed explicitly from the albedo and the downward flux at the surface.

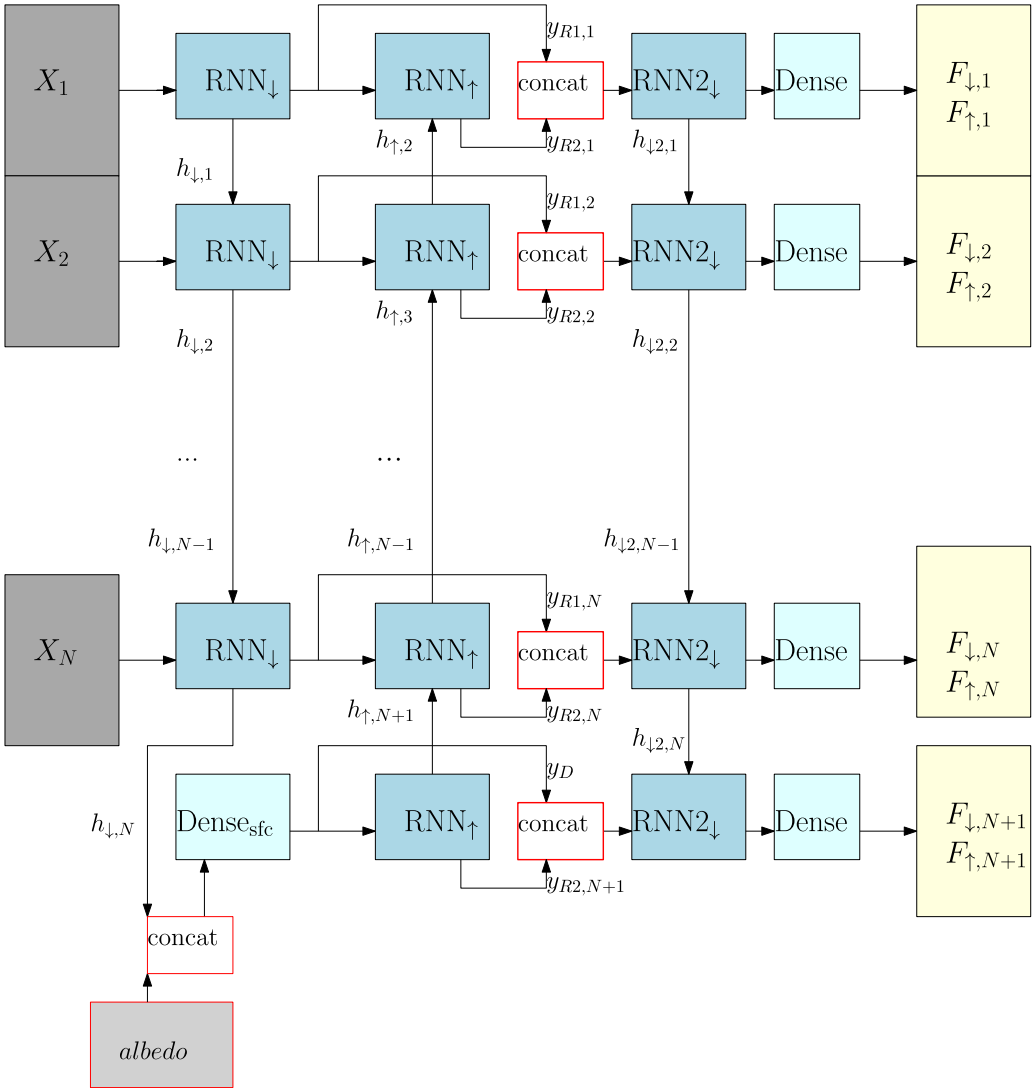

Figure 4: A RNN-based approach to predicting fluxes at layer interfaces (N+1) from layer-wise inputs (N), consisting of three RNNs to mimic two-stream radiative transfer equations with scattering. The first RNN ($RNN_\downarrow$) has a forward (downward) direction, and when it reaches the end of the sequence, i.e. the last vertical layer at N, its hidden state $h_{\downarrow,N}$ is concatenated ("concat") with the surface albedo and fed to a dense layer, whose output is then concatenated with the RNN sequence. The dense layer essentially replaces the RNN at the boundary, where layer-wise inputs are missing. Hereafter the sequence has a length of (N+1), and is connected to a backward/upward RNN ($RNN_\uparrow$). Then, the first two sequences are concatenated (as is usually done in a bidirectional RNN) and connected to a third and final RNN ($RNN2_\downarrow$). Finally, the sequential output from this RNN is connected to a dense layer which predicts two values, the upwelling and downwelling fluxes scaled by incoming flux.

3.3 FNN-RRTMGP - emulation of gas optics only

Successful NN emulators for RRTMGP gas optics have been developed in earlier work: in Veerman et al. (2021), average flux errors were within 0.5 Wm^{-2} of RRTMGP, while in Ukkonen et al. (2020) root-mean-square-errors (RMSE) in heating rate with respect to line-by-line results were virtually identical with RRTMGP. Here an identical NN methodology as in Ukkonen et al. (2020) is used, which involves predicting absorption and Rayleigh cross-sections with two separate NNs.

The main advantage of using neural networks for gas optics is efficiency: whereas the original kernel computes optical properties separately for each band and each minor gas species (the absorption due to two major gases in a band is computed separately and parameterized to account for overlap in the absorption spectra), the NN can take all gases as one input vector and predict the optical properties for all g-points as one output vector. Consequently, minor greenhouse gases can be included with almost no additional cost. NNs are also suitable for predicting optical properties from a physical perspective, since the original kernel relies on empirical look-up-tables and incorporates no physical laws explicitly. Further benefits are generalization to arbitrary vertical grids by predicting layer-wise optical properties normalized by number of molecules (cross-sections), and that a NN can treat gas overlap implicitly. In theory, a novel NN gas optic model could be trained directly on data generated with line-by-line radiation codes to avoid errors associated with gas overlap assumptions, but the data generation would be a significant computational challenge.

3.4 FNN-RefTrans - emulation of reflectance-transmittance computations

Training NNs to emulate the radiative transfer solver was considered for this work, but because RTE and other solvers perform computations per g-point, an emulator which respects the underlying physics and similarly operates on g-points is unlikely to be more efficient (broadband fluxes could be predicted directly, but the inputs are still defined per g-point).

An alternative is focusing on computations of reflectance and transmittance in the shortwave solver. While the efficiency drawback of explicit g-point computations remain, this may be more promising for FNNs since the problem has a simpler nonlinear input-output mapping which does not include vertical dependencies. The reflectance-transmittance computations (kernel sw_two_stream) are furthermore the slowest part of RTE and exhibit a high sensitivity to numerical precision.

Simple neural networks are able to predict direct and diffuse reflectance and transmittances with high accuracy (Figure 5). However, when implementing the NNs into the radiation code it was discovered that even very small inaccuracies overall (with R-squared > 0.999 for each variable) can translate into significant RMSE and maximum errors in net fluxes; typically tens and hundreds of Wm^{-2} respectively. A possible explanation is a larger sensitivity for errors at specific values of reflectance and transmittance, specific g-points (which contribute to broadband flux more strongly than others), or specific atmospheric levels, or just a high sensitivity in the dependence of flux on reflectance and transmittance in general. For instance, predicting intermediate values of transmittance accurately may be more important than values near zero, since the latter case is likely to be associated with radiation being fully extinguished (reflected or absorbed). The distribution of the predictands is highly skewed with such intermediate values being rare, and as a result are also associated with larger errors when employing a regular loss function.

To combat this problem, one could devise custom loss functions, data transformation, or synthetic data generation to create more samples for the important but under-

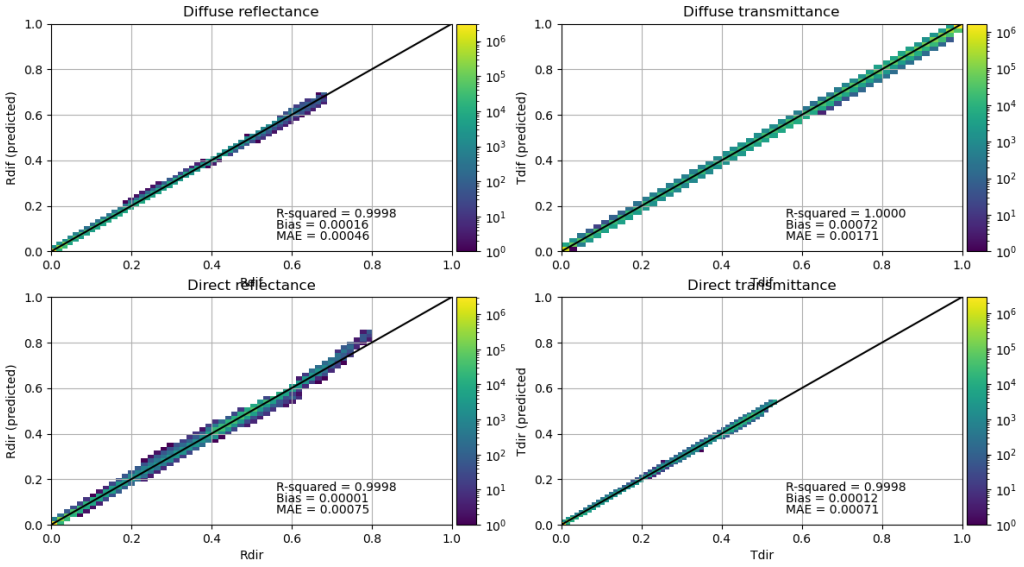

Figure 5: Comparison of the predicted (y-axis) and true (x-axis) reflectance and transmittance values using the validation data set and final REFTRANS model, which has 12 neurons in two hidden layers. The colors on the scatter plot correspond to the occurrence on a log-scale.

represented parts of the distribution. A simple data transformation which reduced errors in radiative flux was to take the square root of the output prior to training, which makes the distribution more Gaussian (albeit still highly non-Gaussian). We then tested custom loss functions which give smaller weights to intermediate values of all four outputs, and/or a smaller weight to diffuse transmittance, but found no clear improvements in predicted fluxes. Figure 5 shows the validation performance of the final model.

3.5 Summary of model architectures and methodologies

The architecture and pre-processing used for the different NN emulators are described in Table 1. The model hyperparameters (number of hidden neurons, hidden layers and activation functions) as well as suitable pre-processing methods - which were often found to be more important than NN hyperparameters - were tuned by hand. The objective of this laborious tuning process was to find a reasonable trade-off between accuracy and model complexity, which determines the computational cost. This restricted the reflectance-transmittance emulator to a very simple NN model, as it turned out to be difficult to surpass the efficiency of the original computations. For the FNN emulating the entire radiation scheme, efficiency was less of a consideration, as the inference code using this method was very fast regardless.

The hyperparameters of the gas optics emulator were taken from Ukkonen et al. (2020). All models were trained using the Adam optimizer (Kingma & Ba, 2015) and the early-stopping method, which stops training when the validation error has not improved for a certain number of epochs (here 28). The batch size was set to 1024.

Model	FNN-RadScheme	RNN-RadScheme	FNN-RRTMGP	FNN-RefTrans
Emulated component	Radiation scheme with gas and cloud optics	Radiation scheme with gas and cloud optics	Gas optics	Reflectance-transmittance computations
Input	scalars α and μ_0 + vertical profiles of gas mole fractions, T, p, cloud ice and cloud water	same as for FNN-RadScheme but one layer at a time	gas mole fractions, T, p	τ, ssa, g, mu, T_{noscat}
Input size	$2 + 9 \cdot nlay = 542$	$2 + 9 = 11$	5	7
Output	vertical profiles of broadband fluxes	vertical profiles of broadband fluxes	absorption/Rayleigh cross-sections as a vector of g-points	$R_{dif}, T_{dif}, R_{dir}, T_{dir}$
Output size	$2 \cdot nlev = 122$	2	$ngpt = 224$	4
Hidden layers	Dense, Dense, Dense	RNN, Dense, RNN, RNN, Dense	Dense, Dense	Dense, Dense
Activation functions in hidden layers and output layer	relu, relu, relu, sigmoid	tanh, linear, tanh, tanh, sigmoid	softsign, softsign, linear	softsign, softsign, hard sigmoid
Neurons in each hidden layer	128	16	16	12
Total parameters	118,266	5,698	4,208	280
Required iterations	$ncol$	$ncol$ ($\times$ 3 $nlay$ internallly)	$ncol \times nlay$ ($\times$ 2 NN models)	$ncol \times nlay \times ngpt$
Flexible with regards to vertical grid	No	Yes	Yes	Yes
Input scaling	$x_i = \frac{x_i}{max(x_i)}$	$x_i = \frac{x_i}{max(x_i)}$	$x = log(x)$ for p; $x = x^{\frac{1}{4}}$ for H_2O and O_3; $x_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}$	$x = x^{\frac{1}{8}}$ for τ; $x = \frac{x}{max(x)}$ for τ (other features already in 0-1 range)
Output scaling	$y_i = \frac{y_i}{F_{\downarrow,0}}$	$y_i = \frac{y_i}{F_{\downarrow,0}}$	$y = y^{\frac{1}{8}}$; $y_i = \frac{y_i - \bar{y}_i}{\sigma_y}$	$y = y^{\frac{1}{4}}$

Table 1: Description of the different models. Abbreviations: α = surface albedo, μ_0 = cosine of solar zenith angle, T = temperature, p = pressure, ssa = single-scattering albedo, g = asymmetry parameter, T = transmittance, R = reflectance, H_2O = mixing ratio of water vapor, O_3 = mixing ratio of ozone.

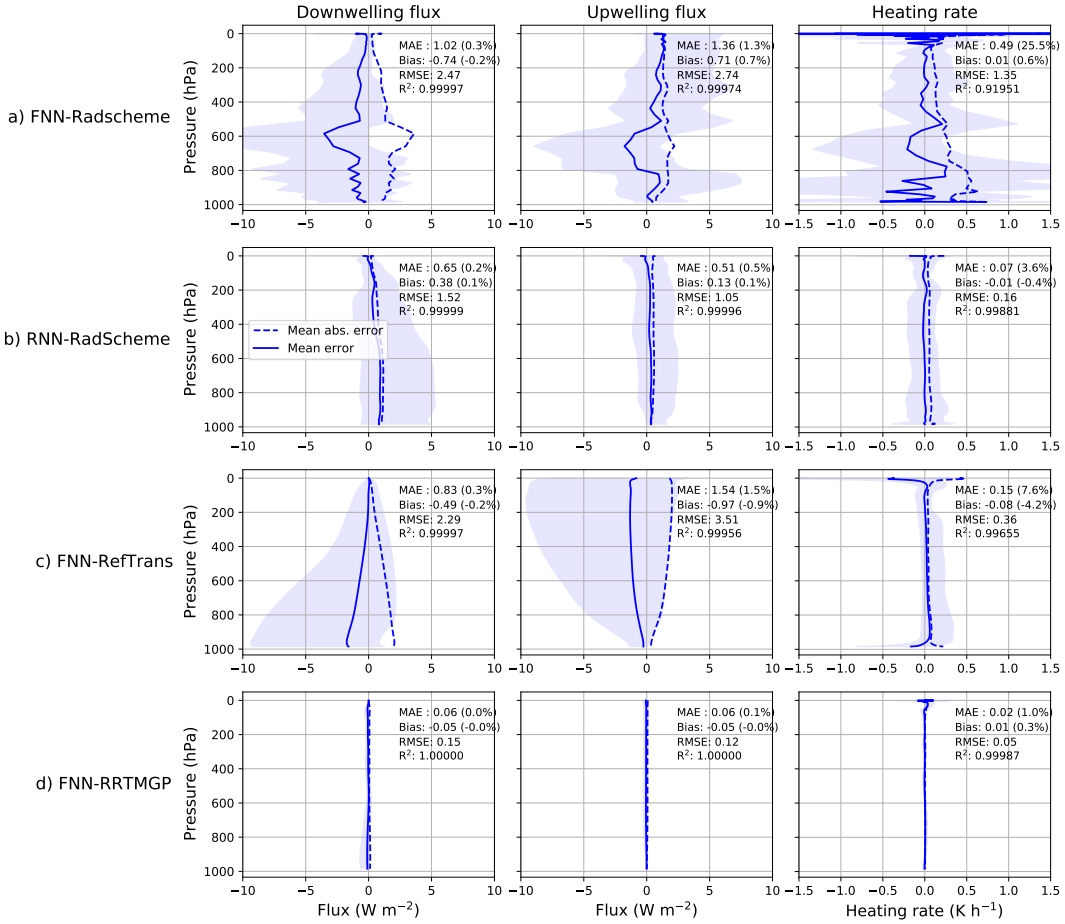

Figure 6: Vertical profiles of the error in shortwave downwelling flux (left column), upwelling flux (middle column) and heating rates (right column) for the test data (2015) using different emulation methods: replacing the radiation scheme with (a) a feed-forward NN or (b) a bidirectional recurrent NN, (c) replacing only the radiative solver's reflectance-transmittance computations with a FNN, or (d) replacing the gas optics computations with a FNN. The solid and dotted lines show the mean error and mean absolute error, respectively, while the shaded area indicates the 5th and 95th percentile of mean absolute error. For FNN-Radscheme (a) the mean heating rate errors at TOA (0.01 Pa) reach around 3.5 K day^{-1} (the x-axis has been cropped). In the annotated statistics, the number in parenthesis gives the error as a percentage of the column and layer mean of the variable.

4 Results

4.1 Accuracy

The models are evaluated by comparing the final output of the radiation code, fluxes and heating rates, to a reference result computed in double precision using RTE+RRTMGP. (Comparison to a single precision result would be very similar, as the NN errors are larger than those from using reduced precision.)

The errors in flux and heating rate using different emulators is shown in Figure 6. In this offline evaluation based on one year of independent global data, all emulation methods produce fluxes with R-squared values very close to 1 and mean absolute errors around

1% or less. In the case of gas optics emulation (FNN-RRTMGP), there is practically no error in fluxes. The emulation of the whole scheme (FNN-RadScheme) gives a similar accuracy in flux compared to emulating only reflectance-transmittance computations (FNN-RefTrans), which is a poor result for the latter method, as it is far more expensive.

Heating rates computed from these fluxes show much larger differences across emulators. FNN-RadScheme has the largest heating rate errors with a mean absolute error (MAE) of 0.50 K day^{-1}, or 25.5% when expressed as a percentage of the mean HR in the dataset. The radiation scheme emulator based on recurrent NNs (RNN-RadScheme) produces far more accurate heating rates despite not predicting them directly, with a MAE of 0.07 and RMSE of 0.16 K day^{-1}. FNN-RefTrans reproduces heating rates well relative to fluxes; with errors well below 0.5 K day^{-1} throughout most of the atmosphere despite the flux errors being comparable to FNN-RadScheme. The most accurate heating rates are seen with FNN-RRTMGP with a MAE of only 0.02 K day^{-1}.

For simulating climate, the upwelling flux at the top-of-atmosphere (TOA) is an important quantity. All emulators have small errors in TOA upwelling flux (Figure 7): less than 1 Wm^{-2} for all models but FNN-RefTrans. Likewise, the downwelling flux at surface is predicted within roughly 1% by all emulators (Figure 8).

4.2 Speed-up

Speedup of the radiation codes were measured on a modern workstation with both reference and NN computations performed in single precision. A fair comparison is ensured by implementing all NN models, with the exception of the RNN, in the RTE+RRTMGP-NN Fortran code and including the overhead from pre- and post-processing. Principal timings were done using the AMD Ryzen 7 5800H processor and GNU compiler version 11 (compiler options *-march=native -O3*). The matrix-matrix computations in RTE+RRTGMP-NN were accelerated using AMD BLIS (`https://developer.amd.com/amd-aocl/`) version 3.0.6. The Fortran code uses blocking of the columns for better cache performance; for each emulator, an optimal block size was used. All timings represent the best result from three trials.

The computation of cloudy-sky fluxes for the 81920 test columns took roughly 18.5 seconds using the reference code a single CPU core. By comparison, the FNN-RadScheme computed fluxes in just 0.35 seconds, a 52-fold speed-up. This is similar to what has been reported in other studies (e.g., Song & Roh, 2021). Replacing only the gas optics component with a FNN reduces the the runtime of the gas optics by a factor of 3, but the total runtime by only 25%. This reflects that the solver is the most expensive part of SW radiation computations in optimized RTE+RRTMGP (Ukkonen et al., 2020). Finally, the reflectance-transmittance emulator is not faster than the original code, but 40-45% slower. This is despite the FNN being a very simple model with only 280 parameters. The slowness of the method can be attributed to it operating on individual spectral points as does the original code, but not being tailored as the physical equations, resulting in redundant computations.

Finally, the RNN and FNN models predicting fluxes was evaluated within Python using the ONNX Runtime Library (ORT) version 1.9.0, first using a CPU (single core). This was necessary because the neural-Fortran library does not support RNNs. These timings do not include pre- and post-processing, but those accounted for less than 10% of the runtime of FNN-RadScheme in Fortran. The inference with the RNN emulator took roughly 4.1 seconds using ORT, representing a speed-up of 4.5X over the reference code in Fortran. This is a significant speed-up, but much smaller than obtained with the FNN model. To compare the FNN and RNNs on a single platform, the ONNX timings were also done for FNN-RadScheme, which in this instance took 0.21 seconds. It can be concluded that the recurrent NN approach is roughly 20 times slower than an FNN-based approach on CPUs. The performance on a GPU (RTX 3060 Mobile) was then briefly

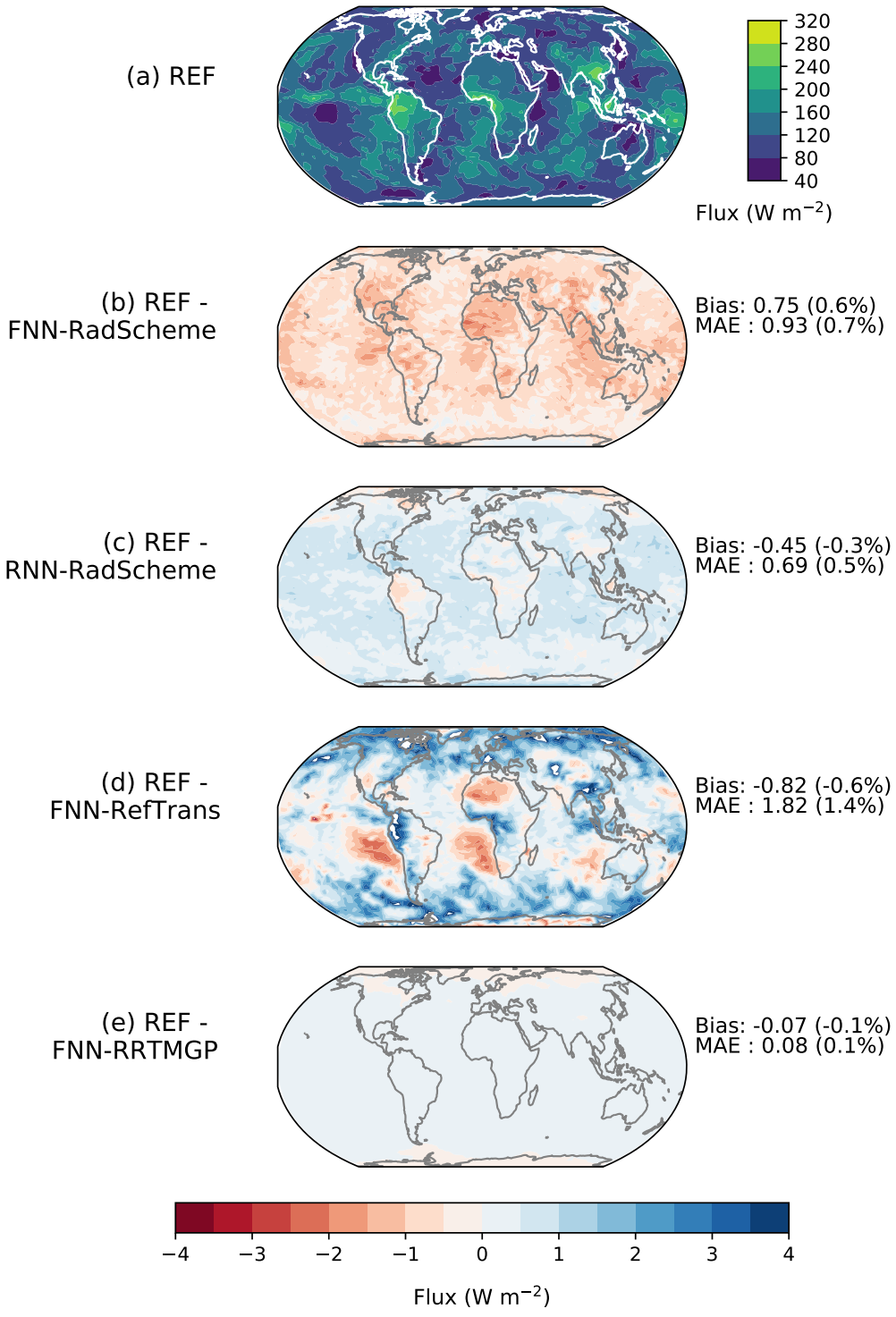

Figure 7: Global upwelling shortwave flux at top-of-atmosphere for the testing year 2015 as computed with RTE+RRTMGP (a) and the grid box mean differences in this quantity using different emulators (b - e). Bulk error statistics with respect to individual columns are displayed on the right hand side.

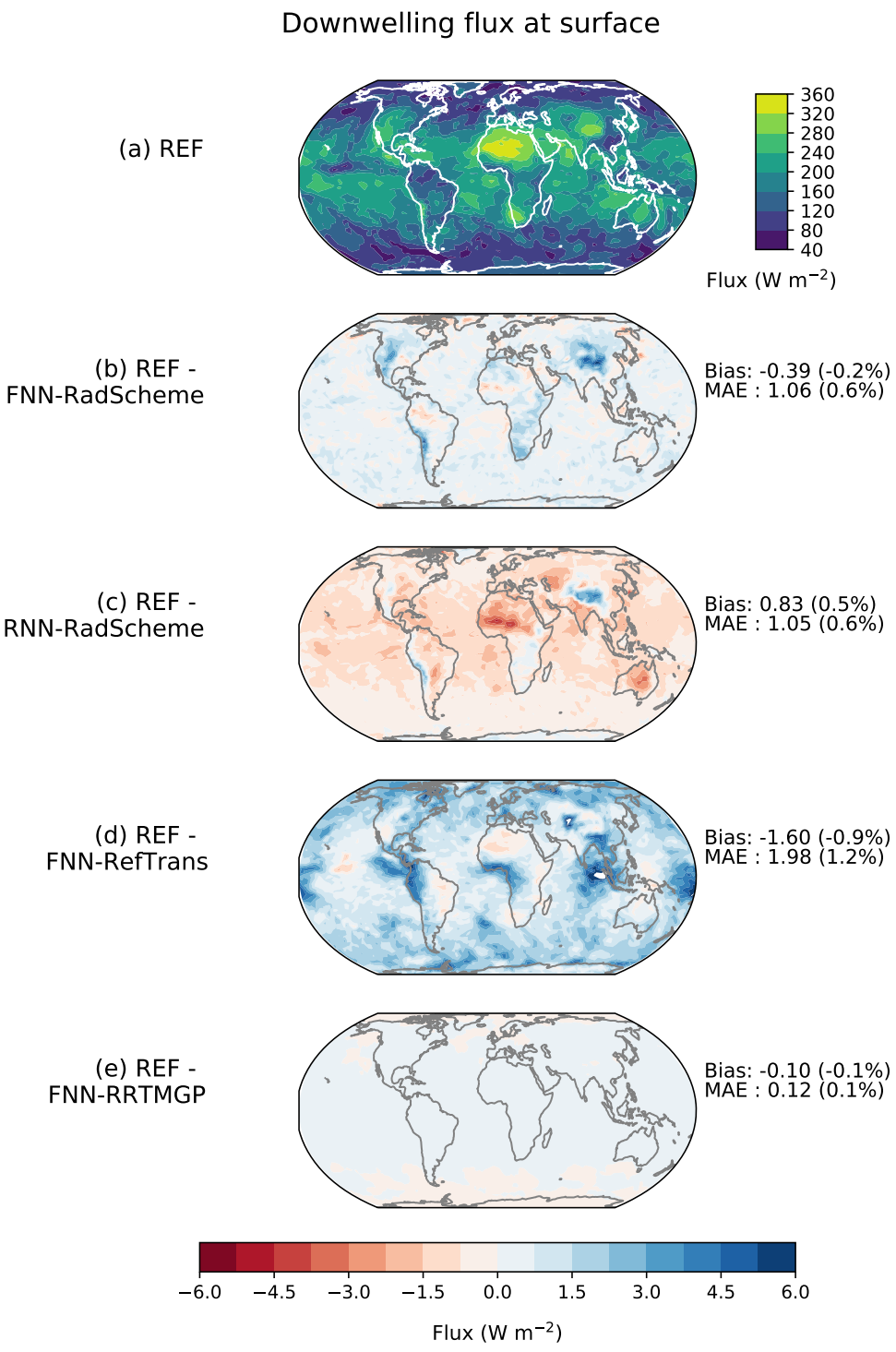

Figure 8: As in Figure 7, but for downwelling flux at surface.

evaluted using ORT. The RNN inference time is reduced to 0.34 seconds on the GPU, while the FNN inference took a mere 0.046 seconds. The performance gap between the FNN and RNN-based approaches for radiative transfer is therefore reduced considerably when using GPUs, here to roughly 7.4X.

5 Benefits of targeted and physics-informed machine learning

All the emulators evaluated here produce very reasonable fluxes, but the large sensitivity of heating rates and noise in the fluxes predicted by feed-forward NNs results in relatively large heating rate errors. Some other studies have sidestepped this issue by predicting heating rates directly, implying a lack of energy conservation which may or may not be an issue in practice but is nonetheless undesirable in an operational setting.

The heating rate errors with flux-predicting FNN seems to be caused by the fact that the NN outputs at different atmospheric levels are not structurally correlated with outputs at other levels. The recurrent NN, which does have incorporate the vertical dependence, produces far more accurate heating rates. The RMSE of 0.16 K day^{-1}, evaluated across the whole atmosphere with the uppermost layer at 10 Pa, is smaller than the errors reported in other studies. For instance, shortwave heating rates had an offline RMSE of 0.5 K day^{-1} in Roh and Song (2020) and 0.17 K day^{-1} in Song and Roh (2021). In both of these studies, the vertical grid only reached 50 hPa and heating rates were predicted directly with an FNN. With this in mind these initial results with an RNN are very promising, and the errors are in fact similar in magnitude to parameterization errors associated with the correlated-k distribution method (Hogan & Matricardi, 2020, Figure 7). The drawback of the RNN approach is that its sequential nature, which lets it emulate a radiation parameterization more closely, also makes it less efficient than FNNs. However, a speed-up of more than 4 times is still significant, and when testing with a GPU a much larger speed-up was seen (a factor of 54 relative to a single CPU core).

Smooth flux profiles, associated with small heating rate errors, are also seen with FNN-RRTMGP and FNN-RefTrans, demonstrating the advantage of retaining the radiative transfer equations. While the FNN-RefTrans model is considerably slower than the original code, and therefore found to be an unsuccessful emulation target, the gas optics emulation produces extremely accurate results while speeding up the original look-up-table by several factors.

Regarding the choice of output, while it may seem attractive to predict heating rates directly in addition to fluxes at boundaries, it should also be noted that it could lead to larger errors in fluxes: the RMSE in SW flux was around 15 Wm^{-2} in offline evaluation in both Roh and Song (2020) and Song and Roh (2021). By comparison, the MAE in SW upwelling flux at TOA and downwelling flux at surface were around 1 Wm^{-2} or less for both FNN-RadScheme and RNN-RadScheme. It is unclear, however, why tests with a heating rate predicting FNN had relatively small errors in the boundary fluxes in this study (Figure 2). Our experience is that hyperparameters (number of hidden layers, activation functions used in the output layer) and other technical details in how ML models are developed can have a substantial impact on the results. Unfortunately, these are not always well documented. A great example is pre-processing of both inputs and outputs, which can have a major impact. Besides such more overlooked aspects, the quantity of training data can obviously be an important factor. In this study, the number of training profiles was initially an order of magnitude smaller, and model errors significantly worse.

How the NN emulators would perform in a prognostic evaluation when embedded in a large-scale model is a critical question. Such experiments were considered to be out of the scope of the present work. Nonetheless, comparison with studies which have performed both offline and online errors should give some indication. These include the stud-

ies mentioned above with larger offline errors, where prognostic evaluation based on a squall-line simulation (Roh & Song, 2020) and a regional NWP simulation (Song & Roh, 2021) did not show a significant degradation for precipitation and temperature forecasts, and forecasts were improved relative to infrequent calls of the original scheme at the same computational cost. In another study, NN output consisting of both flux and heating rate profiles had mean errors of a few percentage points in an offline setting (Figure 1 in Pal et al., 2019). In year-long climate simulations, the NN parameterization resulted in time- and area-averaged SW surface downwelling fluxes that differed substantially from the reference simulation, but the differences were comparable to the internal variability of the model.

6 Conclusions

Emulating a sub-component of a physics scheme reduces the potential to speed-up, but can greatly improve accuracy and generalization. For operational implementation, the fact that the dimensionality is much smaller is important, because it allows sampling the input space more thoroughly. Accelerating computations of reflectance and transmittance using NNs was not successful, but the gas optics component is relatively straightforward to emulate at high accuracy, and the FNNs are much faster than the look-up-table method of the original code.

It was also found that transforming inputs and outputs prior to training can have a substantial impact on the accuracy in both the physical output variable as well as derived variables which are not directly predicted. Scaling shortwave fluxes by the incoming TOA flux reduces flux errors substantially, but at the expense of heating rate errors when using a feed-forward NN. A loss function which computes the heating rate error alleviated the issue, but predicting heating rates directly (as opposed to fluxes) may be necessary to produce accurate heating rates with a feed-forward NN.

Finally, this study has contributed to more accurate emulation of radiation computations by developing a recurrent NN method that can predict fluxes at layer interfaces from inputs defined at levels and the surface. The author is not aware of previous work using recurrent NNs to compute radiative fluxes in a vertical column. This method is in principle flexible with regards to the vertical grid. A model of roughly 5,600 parameters which consists of three RNN layers, propagating information in both directions of the vertical column (mimicking radiative transfer computations), is able to predict fluxes and heating rates far better than a FNN with more than 100,000 parameters. Fewer parameters, in turn, makes it much easier to build general models which can replace parameterizations in real applications. While the speedup offered by the RNN is smaller than with FNNs, it still offered a 4-fold speedup on a CPU and a 54-fold speedup on GPU relative to running the original scheme a single CPU core. Future work should investigate the RNN approach further by implementation in a large-scale model.

Acknowledgments

The author would like to thank Kristian Pagh Nielsen for helping him understand radiation and how radiation schemes interact with other parts of large-scale models, as well as for reviewing the manuscript. The code used in this work is available on Github on a dedicated branch of the RTE+RRTMGP-NN package (`https://github.com/peterukk/rte-rrtmgp-nn/tree/nn_dev/examples/emulator-training`); which includes Python and Fortran code for data retrieval, pre-processing, data generation, model training, and model evaluation. The machine learning input-output data can be found on Zenodo (`https://doi.org/10.5281/zenodo.5564314`). The author acknowledges 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.

References

Aldakheel, F., Satari, R., & Wriggers, P. (2021, July). Feed-forward neural networks for failure mechanics problems. , *11*(14), 6483. Retrieved from `https://doi.org/10.3390/app11146483` doi: 10.3390/app11146483

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, *126*(9), 098302.

Beucler, T., Rasp, S., Pritchard, M., & Gentine, P. (2019). Achieving conservation of energy in neural network emulators for climate modeling. *arXiv preprint arXiv:1906.06622*.

Brenowitz, N. D., & Bretherton, C. S. (2018). Prognostic validation of a neural network unified physics parameterization. *Geophysical Research Letters*, *45*(12), 6289–6298. doi: 10.1029/2018gl078510

Curcic, M. (2019). A parallel fortran framework for neural networks and deep learning. In *Acm sigplan fortran forum* (Vol. 38, pp. 4–21).

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., ... Vogt, H. (2018, May). Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 GPUs with COSMO 5.0. , *11*(4), 1665–1681. Retrieved from `https://doi.org/10.5194/gmd-11-1665-2018` doi: 10.5194/gmd-11-1665-2018

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751. doi: 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.

Hansen, J. E. (1971). Multiple scattering of polarized light in planetary atmospheres. part i. the doubling method. *Journal of Atmospheric Sciences*, *28*(1), 120–125.

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., & Matricardi, M. (2020). Evaluating and improving the treatment of gases in radiation schemes: the correlated k-distribution model intercomparison project (ckdmip). *Geoscientific Model Development*, *13*(12), 6501–6521.

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

Jiang, S., Stamnes, K., Li, W., & Hamre, B. (2005). Enhanced solar irradiance across the atmosphere–sea ice interface: a quantitative numerical study. *Applied optics*, *44*(13), 2613–2625.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings.* Retrieved from `http://arxiv.org/abs/1412.6980`

Krasnopolsky, V., Fox-Rabinovitz, M., Hou, Y., Lord, S., & Belochitski, 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.

Meador, W., & Weaver, W. (1980). Two-stream approximations to radiative transfer in planetary atmospheres: A unified description of existing methods and a new improvement. *Journal of Atmospheric Sciences*, *37*(3), 630–643.

Odyssee-Mure. (2021). *Sectoral profile households - electricity consumption per dwelling.* Retrieved 2021-02-01, from `https://www.odyssee-mure`

.eu/publications/efficiency-by-sector/households/electricity-consumption-dwelling.html

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*(11), 6069–6079.

Pincus, R., Mlawer, E., & Delamere, J. (2019). Balancing accuracy, efficiency, and flexibility in radiation calculations for dynamical models. *Earth and Space Science Open Archive*. Retrieved from https://www.essoar.org/doi/abs/10.1002/essoar.10500964.2 doi: 10.1002/essoar.10500964.2

Räisänen, P., Barker, H. W., & Cole, J. (2005). The monte carlo independent column approximations conditional random noise: Impact on simulated climate. *Journal of climate*, *18*(22), 4715–4730.

Rasp, S., Pritchard, M. S., & Gentine, P. (2018, September). Deep learning to represent subgrid processes in climate models. , *115*(39), 9684–9689. Retrieved from https://doi.org/10.1073/pnas.1810286115 doi: 10.1073/pnas.1810286115

Roh, S., & Song, H.-J. (2020). Evaluation of neural network emulations for radiation parameterization in cloud resolving model. *Geophysical Research Letters*, *47*(21), e2020GL089444.

Song, H.-J., & Roh, S. (2021). Improved weather forecasting using neural network emulation for radiation parameterization. *Journal of Advances in Modeling Earth Systems*, e2021MS002609. doi: 10.1002/essoar.10506992.1

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*(12), e2020MS002226. doi: 10.1029/2020ms002226

Veerman, M. A., Pincus, R., Stoffer, R., Van Leeuwen, C. M., Podareanu, D., & Van Heerwaarden, C. C. (2021). Predicting atmospheric optical properties for radiative transfer computations using neural networks. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200095.

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021). Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision. *Geophysical Research Letters*, *48*(6), e2020GL091363.

Yuval, J., & OGorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature communications*, *11*(1), 1–10.

Zängl, G., Reinert, D., Rípodas, P., & Baldauf, M. (2015). The icon (icosahedral non-hydrostatic) modelling framework of dwd and mpi-m: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, *141*(687), 563–579.