UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

ESCAPE 2

European Commission
Horizon 2020
European Union funding
for Research & Innovation

Danmarks
Meteorologiske
Institut

Ph.D. thesis

Improving the trade-off between accuracy and efficiency of atmospheric radiative transfer computations by using machine learning and code optimization

Peter Ukkonen

Supervisors:

Professor Eigil Kaas (NBI)
PhD Kristian Pagh Nielsen (DMI)

Submitted on: March 31, 2022

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

A Thesis Submitted for the Degree of Doctor of Philosophy

Publication: 31.03.2022

Support: European Union's Horizon 2020 research and innovation programme.
Grant Number: 800897 (ESCAPE-2)

Author: M.Sc. Peter Ukkonen
E-mail: peter.ukkonen@nbi.ku.dk, puk@dmi.dk
UCPH username: wdq538

University Supervisor: Professor Eigil Kaas
External Supervisor (DMI): PhD Kristian Pagh Nielsen

Ph.D. partners: University of Copenhagen, Faculty of Science
Danish Meteorological Institute

Keywords: Radiation parameterization, Machine learning, Code optimization

Abstract

Radiative transfer parameterizations are physically fundamental components of weather and climate models, and often represent a computational bottleneck in climate models. At the same time, the amount of energy needed to run these models is growing alongside a push towards ever-increasing resolution and physical realism. It is therefore clear that the trade-off between computational efficiency and accuracy in radiation parameterizations is very important and needs to improve. This PhD study aims to contribute to this worthy goal by approaching it from two angles: using approximative but faster machine learning methods to replace physical radiation schemes or their components, and two, improving the efficiency of existing schemes by using code restructuring techniques.

Here it is shown that by replacing only one component of a radiation parameterization with a neural network (NN), significant improvements in runtimes can be achieved without any considerable loss of accuracy. This component, known as the gas optics scheme, computes the optical properties of the gaseous atmosphere. Combining the NN gas optics with a refactored radiative transfer solver, a modern radiation scheme (RTE+RRTMGP) was made 2-3 times faster. By implementing the NN gas optics in the "ecRAD" radiation scheme used in a leading state-of-the-art weather model, the Integrated Forecast System (IFS), it was demonstrated that compared to using the original gas optics, the NN emulator does not significantly impact the model climate and speeds up ecRAD by roughly a third.

This PhD thesis also contributes to more accurate emulation of atmospheric radiative transfer (the full radiation scheme) by development of a novel method based on recurrent neural networks (RNNs), the structure of which more closely reflects the physics of radiative transfer. Shortwave fluxes and heating rates can be predicted with far greater accuracy compared to using standard feed-forward NNs, while also requiring several orders of magnitude fewer model parameters.

Finally, a significant code restructuring of ecRAD was carried out. The focus was on improving the efficiency of a radiative transfer solver capable of representing the 3-D radiative effects of clouds (SPARTACUS). These 3-D effects are currently ignored in all weather and climate models. The computational cost of optimized SPARTACUS, when combined with an advanced new gas optics scheme with a smaller spectral resolution, is actually less than the operational radiation code in the IFS! The impact of these results should be significant, assuming some remaining issues with numerical instability when running SPARTACUS in single precision can be resolved.

Sammendrag

Strålingstransport-parametriseringer er fysisk fundamentale komponenter i vejr- og klima-modeller. Ofte repræsenterer de en beregningsmæssig flaskehals i klima-modeller. På samme tid vokser mængden af energi, der kræves til at køre disse modeller, sammen med stadigt højere krav til opløsning og fysisk realisme. Det er derfor klart, at afvejningen mellem beregningsmæssig effektivitet og præcision i strålings-parametriseringer, der kan synes som et trivielt emne, faktisk er meget vigtigt og behøves at forbedres. Dette PhD-studie sigter til at bidrage til dette værdige mål ved at nærme sig det fra to indgangsvinkler: At bruge tilnærmede men hurtigere machine learning metoder til at erstatte fysiske strålings-beregninger og deres komponenter, og to - ved at forbedre effektiviteten af de eksisterende beregninger ved hjælp af refaktorerings-teknikker.

Det er vist, at ved kun at erstatte en komponent i en strålings-model med et neural netværk (NN) kan betydelige forbedringer i den tid, det tager at afvikle modellen, opnås uden at ofre præcisionen i nogen nævneværdig grad. Denne komponent - kendt som gas-optikken - beregner de optiske egenskaber af gasserne i atmosfæren. Ved at kombinere NN gas-optikken med den refaktorerede kode, der løser strålingstransporten, blev en moderne strålingstransport-model (RTE+RRTMGP) gjort 2-3 gange hurtigere. Ved at implementere NN gas-optikken i "ecRad" strålings-modellen, der bliver brugt i en førende vejr-model: Integrated Forecast System (IFS), blev det demonstreret, at påvirkningen af model-klimaet ved at bruge NN-emulatoren ikke er af betydning, og at dette gør ecRad ca. en tredjedel hurtigere i forhold til, når den oprindelige gas-optik bruges.

Denne PhD-afhandling bidgrager også til mere præcise emuleringer af atmosfærisk strålingstransport (hele strålings-modellen) ved at udvikle en ny metode baseret på rekursive neurale netværker (RNN'er), hvis strukturer bedre repræsenterer den fysiske strålingstransport. Kortbølgede fluxe og opvarmnings-rater kan blive forudsagt med langt større nøjagtighed sammenlignet med, når standard feed-forward NN'er bruges, imens der også behøves flere størrelsesordner færre antal model-parametre.

Sidst skal det nævnes, at en betydelig kode-restrukturering af ecRad blevet gennemført. Fokus har været på at forbedre effektiviteten af en strålingstransportmodel, der kan repræsentere 3-D strålings-effekterne af skyer (SPARTACUS). Disse 3-D effekter er hidtil blevet ignorerede i alle vejr- og klima-modeller. De samlede beregningsmæssige udgifter, når denne optimerede version af SPARTACUS bruges og kombineres med en ny avanceret gas-optik-model, er faktisk mindre end den udgifterne til den oprindelige strålings-model i IFS! Disse resultater er af stor betydning, under antagelsen af, at et par tilbageværende problemer med numerisk instabilitet, når SPARTACUS køres med single-præcision, kan løses.

Preface and acknowledgements

The journey of this PhD at times felt incredibly exciting and invigorative, and at other times, long and arduous. Studying a PhD during a pandemic and working for a large part independently (arguably too much so) have no doubt contributed to the latter feelings. Still, it has been an absolute privilege to do a PhD on a topic that one feels really matters, and this has been an enormous source of motivation.

I would not have come this far without the support, guidance and help from others. I would like to thank my supervisor **Kristian Pagh Nielsen** for his enthusiasm and guidance on all matters radiation, and for many interesting conversations. A heartfelt thanks goes to my other supervisor, **Eigil Kaas**, for always being supportive, kind and helping me avoid imposter syndrome. (At this stage, I don't feel like an impostor, just don't ask me to do any math).

It was an absolute pleasure to work with **Robert Pincus** on the first paper. I feel like this collaboration made me a better writer and researcher. Similarly, it has been a fantastic opportunity to work with **Robin Hogan**. To paraphrase Mark Fielding from my short but enjoyable stay at ECMWF, the real test for machine learning models is to beat Robin Hogan (this does not refer to some robot-human boxing match, but pitting ML results against his radiation schemes - ECCKD in particular feels like a game-changer).

Last but not least, I would like to thank my family for giving me all the opportunities in life, and my partner Ashleigh without whose incredible support and patience I can't imagine I would ever have come so far.

This PhD study has been mainly funded by the EU through the ESCAPE-2 project (Energy-efficient SCalable Algorithms for weather and climate Prediction at Exascale). It started in December 2018 as a collaboration between the University of Copenhagen (Niels Bohr Institute) and the Danish Meteorological Institute (DMI), where I have been employed. A funded-3 month extension was received from NBI, and I think those extra 3 months have made all the difference. In fall 2021, a two-month change of environment took place at the European Centre for Medium-Range Weather Forecasts (ECMWF) in Reading.

Paper 4 is still in early stages of preparation. Because it shows results that are significant for the conclusions of the thesis (as well as in itself) it was decided that the early manuscript should be included. A major obstacle has been issues with numerical stability in the SPARTACUS radiative transfer which manifest as floating point errors in the radiation code when running the IFS weather prediction model in single precision, subsequently crashing the model. Although these crashes are unrelated to the optimizations done in the course of the PhD, the significance of those optimizations in large part depend on being able to run SPARTACUS in single precision, which had not been tested before. To add to the frustration, SPARTACUS now runs seemingly perfectly in an offline setting, and we have been unable to reproduce the crashes when testing with tens of thousands of atmospheric profiles. This makes bug-fixing almost impossible. Still, I am hopeful, even confident, that these issues can be solved in the coming months to end up with something really exciting: a radiation scheme capable of representing the 3-D radiative effects of clouds, that is fast enough to be used in climate and NWP models.

On the bright side, without the progress of Paper 4 grinding to a halt, Paper 3 might not have happened: this paper came about in the last months of my PhD largely out of a desire to include a third finished manuscript in my thesis (I did not quite get there). The implementation of the NN gas optics in a large-scale dynamical model, and explicit demonstration that it is accurate "even when" used prognostically, should hopefully improve the chances that the NN models, or some aspect of the methods, will be used by others and contribute to faster radiation code in weather and climate models.

Outline

This PhD thesis consists of a background section (describing the theory and methods of radiative transfer, state-of-the-art radiation parameterizations, and an introduction to neural networks), four scientific papers, and a discussion and conclusion section. Two of the papers have been published, and two are in preparation, with Paper 3 close to being submitted.

Paper 1 Ukkonen, P., Pincus, R., Hogan, R. J., Pagh Nielsen, K., Kaas, E. (2020). Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. *Journal of Advances in Modeling Earth Systems*, 12(12) [Published November 2020].

Paper 2 Ukkonen, P. (2022). Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer. *Journal of Advances in Modeling Earth Systems.* [Published March 2022].

Paper 3 Implementation of machine-learned gas optics parameterization in the ECMWF Integrated Forecasting System. [In preparation]

Paper 4 Optimizing the ecRAD radiation scheme with a new gas optics scheme results in affordable computations of 3D cloud radiative effects. [In preparation]

In addition, thousands of lines of code have been written in the course of the PhD. This consists mainly of Python code infrastructure for neural network development and evaluation, as well as Fortran code featuring optimized radiative transfer code, and contributions have been made to faster neural network inference in Fortran. With the exception of code used in unpublished papers (3-4), all essential code to reproduce the results have been made available in public repositories. The ecRAD radiation scheme now has an open source license, and the optimizations featured in Paper 4 will most likely be implemented in the official ecRAD repository on Github, or otherwise released through a fork of it.

Glossary

CKD (Correlated K-Distribution) is a method used in modern radiative transfer parameterizations to treat gas absorption, which requires several orders of magnitudes fewer individual calculations than LBL methods.

CPU (Central Processing Unit) is the main processor in a conventional computer, which executes instructions in a computer program. Modern microprocessors are incredibly complex, and are nowadays implemented in multicore designs, where the CPU consists of multiple processors (cores). From the point of view of high-performance computing, CPU's have many performance features that need to be exploited, such as memory caches and SIMD-level parallelism.

GPU (Graphics Processing Unit) are specialized processors that were originally designed to accelerate graphics tasks, but today are also used for general high-performance computing due to excelling at parallel computing (a typical GPU has thousands of cores, which are much simpler than CPU cores).

LBL (Line-By-Line) radiative transfer is an "exact" method of computing radiative transfer in an absorbing and emitting atmosphere based on resolving each individual absorption line in a spectrum.

ML (Machine Learning) is a branch of computer science and artificial intelligence based on algorithms which can improve automatically by learning from data. These statistical algorithms can be used to solve various tasks and make predictions without being instructed (programmed) on how to do so.

NNs (artificial Neural Networks) are computational models and type of machine learning algorithm which are loosely modelled after biological neural networks. NNs are flexible enough that they can approximate virtually an input-output mapping, which in practice means that they can provide useful answers for various complex problems (such as recognizing a cat from a picture, or simulate atmospheric radiative transfer).

NWP (Numerical Weather Prediction) models are models of the atmosphere and ocean which are used to predict the weather based on current weather conditions.

Parameterizations in weather and climate models refers to physical processes that are too small-scale or complex to be represented in an exact manner in weather and climate models, such as clouds and radiation, and therefore need to be parameterized (indirectly represented through simplified parameters). These sub-grid processes may be contrasted with the "dynamical" part of weather and climate models which explicitly simulate larger-scale atmospheric motions by solving a set of fluid dynamics equations on the underlying grid.

SIMD (Single Instruction Multiple Data) is a type of parallelism in a computer which refers to performing the same operation on multiple data points simultaneously.

Contents

Abstract **i**

Sammendrag **iii**

Preface and acknowledgements **v**

Outline **vii**

Glossary **ix**

1 Introduction **1**

2 Background **3**

2.1 Atmospheric radiation in weather and climate models 4

2.1.1 Emission and absorption 5

2.1.2 Atmospheric transmission 8

2.1.3 Scattering and radiative transfer equation 12

2.1.4 Two-stream approximation 12

2.1.5 K-distribution method 13

2.1.6 State-of-the-art radiation parameterizations 16

2.2 Artificial Neural networks 19

2.2.1 Feed-forward networks 19

2.2.2 Recurrent networks 22

3 Paper 1: Accelerating Radiation Computations for Dynamical Models With Targeted Machine Learning and Code Optimization **27**

3.1 Motivation 27

3.2 Supplementary results 45

4 Paper 2: Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer 49

4.1 Motivation 49

5 Paper 3: Implementation of machine-learned gas optics parameterization in the ECMWF Integrated Forecasting System) 69

6 Paper 4: Optimizing the ecRAD radiation scheme with a new gas optics scheme results in affordable computations of 3D cloud radiative effects 89

6.1 Abstract 89

6.2 Introduction 90

6.3 Higher-level refactoring to expose more parallelism 91

6.3.1 SPARTACUS-SW 91

6.3.2 SPARTACUS-LW 92

6.3.3 TripleClouds-SW 92

6.3.4 TripleClouds-LW 92

6.4 Lower-level optimizations 93

6.4.1 Single precision computation of reflectance and transmittance 93

6.4.2 Pipelining and vectorization 93

6.4.3 Hand-optimized matrix operations: loop unrolling and avoiding redundancy 94

6.4.4 Declaring ng at compile time 94

6.4.5 Other optimizations 94

6.5 Results 96

7 Discussion and conclusion 100

7.1 Which emulators for radiative transfer? 101

7.2 Future perspectives 102

Chapter 1

Introduction

Shortwave radiation from the Sun and Earth's longwave thermal radiation interact with the atmosphere, surface, and clouds and provide the energy which drives climate and weather. At the top of atmosphere, these radiative fluxes are roughly in balance, and when they are not, the planet either warms or cools in response to the energy imbalance. It is therefore crucial that these radiative flows in the atmosphere are accurately represented in weather and climate models: our ability to predict the weather and future changes in climate (as a result of greenhouse gas emissions) depends on it. However, the complexity of atmospheric radiative transfer means that it is computationally far too expensive to represent these radiative processes in an exact manner, even if we have the tools to do so. This means that weather and climate models are required to make simplifications and approximations in the way radiative transfer is represented. Even so, radiation computations remain computationally very demanding, and in climate models, can constitute around 50% of the runtime of the entire model. Were such computational resources to be freed up, they could instead be used to increase fidelity of climate simulations by increasing the model resolution, for example.

All of this means that radiation parameterizations are an integral part of weather and climate models, and attempting to improve the tradeoff between accuracy and computational cost of radiation computations is a remarkably direct exercise in improving the models as a whole. How, then, can we achieve better efficiency, without sacrificing crucial accuracy? This is the subject of the PhD thesis. Here the starting point is existing state-of-the-art radiation parameterizations, and the tools that are investigated for improving upon them consist of machine learning (which can be used to *emulate* a radiation scheme or its components) and code optimization (refactoring existing code to make it run faster). Although seemingly distinct, these two approaches have the same goal and are in some ways related: the computational efficiency of neural networks on modern computer hardware is what makes them useful as a code acceleration tool, but at the same time, the computational efficiency of existing scientific codes can in many cases be improved. The research presented here aims to shed light on which of these approaches is more promising for accelerating radiation computations, and investigate the accuracy/efficiency trade-off offered by different emulation approaches.

The objectives of this thesis are as follows:

- Advance the state-of-the-art in radiation parameterizations by use of machine-learning, code optimization, or both to improve the accuracy/speed trade-off of such schemes (primary objective)
- Study different ways of emulating a radiation scheme, and compare their trade-offs
- Develop new ML methods to emulate a radiation scheme more closely

The last two of these are addressed specifically by Paper 2, while Paper 4 concerns code refactoring of a state-of-the-art radiation scheme.

The use of machine learning (particularly neural networks) to emulate radiative transfer parameterizations has been the subject of more than a dozen studies and actually goes back more than two decades, with growing interest in recent years. Despite this, to the authors knowledge there is no NWP or climate model out there today using such emulators operationally. One potential issue of past approaches, which have typically used simple neural networks to replace the entire radiation scheme, is a focus on *speed* instead of *accuracy*. In a hope to contribute to real-world applications and avoid overlap with past studies, the focus of the present work is on approaches that emphasize accuracy over speed.

Chapter 2

Background

The first section will introduce the topic of atmospheric radiation in weather and climate models, with an emphasis on a basic conceptual understanding of how radiation and radiative transfer in the atmosphere "works" as well as a brief overview of methods used in state-of-the-art radiation schemes. A particular emphasis will be on concepts mentioned in the subsequent publications, such as the correlated-k method. Such a basic, high-level understanding should in many cases be enough to develop machine learning emulators of a physical scheme in an informed manner (and certainly, attempts can be made without almost any domain knowledge, but such "blind" approaches are less likely to be successful or substantial).

Here, "informed manner" is difficult to define. At the minimum, however, it should probably include knowledge of: what is the state-of-the-art in physical parameterizations (to know what schemes to target and what the potential applications are), what the inputs and outputs in radiation schemes and their internal components represent physically, and the approximate range of the input distributions across different applications (to be able to generate representative data sets). In addition, physical or structural understanding of radiative transfer codes may guide the machine learning (ML) development process by being able to identify the most promising ML models and structures, and construct well-designed training data sets. (These requirements, of course, differ from that of a scientist developing physical parameterizations, who needs a much deeper understanding of the physical processes and intimate knowledge of the underlying equations.)

The second section will introduce the artificial neural networks to the reader, assuming only a basic knowledge of statistics and linear algebra. The emphasis will be on the two types of neural networks (NNs) used in this work to parameterize atmospheric radiation: feedforward NNs and recurrent NNs. Previous attempts to emulate radiative transfer schemes in literature have almost exclusively used the former of these, although convolutional networks have featured in a few studies. While NNs is only one class of non-linear machine algorithms among many (others including, for example, support vector machines and ensemble methods based on decision trees) it is a particularly powerful method for regression problems where sufficient training data is available. Because radiation codes used in weather and climate models can be run (and are often evaluated) "offline", that is independently of a numerical weather prediction (NWP) or climate model, it becomes easy to produce large amounts of training data. Therefore, a working assumption is that NNs represent the most promising machine learning algorithm for modeling atmospheric radiative transfer.

2.1 Atmospheric radiation in weather and climate models

In this section, some fundamental laws are introduced and the processes governing atmospheric radiation (emission, absorption and scattering) described, before briefly describing the radiative transfer equation and typical approximations and solutions employed in radiation parameterizations.

Electromagnetic radiation can be described using either "wave language" or "photon language", as it exhibits characteristics of both. In the macroscopic world where we wish to understand how radiation in the atmosphere interacts with gases, clouds, aerosols and the surface, wave language is in general more useful, but there are exceptions: for instance, to understand the spectral absorption of gases a quantum (photon) view is needed, and one accurate method of modeling radiative transfer also deals with photons (Monte Carlo models).

A fundamental property of radiation is its wavelength λ, which is related to frequency ν by

$$\lambda\nu = c, \tag{2.1}$$

where c is the speed of light.

Some other basic definitions of electromagnetic (EM) radiation which are used throughout this dissertation include (Petty, 2006, Section 2.7):

- Flux density, or more commonly *flux* or *irradiance*, gives the total energy per unit time (power) per unit area transported by EM radiation through a plane and has the unit W m^{-2}. Often the magnitude of this energy depends on the orientation of the plane or surface (in atmospheric models, flux on mainly horizontal surfaces is considered). Since our fundamental concern with radiation in the atmosphere is that it carries energy, flux is a key concept.

- Radiant *intensity* is a more detailed measure than flux, giving the contribution to flux from a specific direction. Therefore, flux incident on a surface is obtained by integrating the contributions of intensity of all directions visible from that surface. The concept of intensity is related to that of a *solid angle*, which describes how much of the field of view is covered by an object and has units steradian (sr). Intensity is defined as the flux per unit solid angle traveling in a given direction per unit solid angle (W sr^{-1}).
- Broadband and monochromatic radiation. EM radiation composed only of a single frequency is referred to as *monochromatic*, while radiation over a wider spectral region - an integrated quantity - is known as *broadband* radiation. Monochromatic (or *spectral*) flux can be defined as:

  $$F_\lambda = \lim_{\Delta\lambda \to 0} \frac{F(\lambda, \lambda + \Delta\lambda)}{\Delta\lambda} \tag{2.2}$$

  where $F(\lambda, \lambda + \Delta\lambda)$ is the flux contributed by radiation with wavelengths between λ and $\lambda + \Delta\lambda$. The unit of monochromatic flux is power per unit area per unit wavelength, i.e. $Wm^{-2}\mu m^{-1}$. In reality no radiation is truly monochromatic.

2.1.1 Emission and absorption

All objects emit radiation. The intensity of radiation emitted by a blackbody in thermal equilibrum is given by Planck's function (or Planck's law):

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kT}} - 1} \tag{2.3}$$

where $c = 2.998 \times 10^8$ m s^{-1} is the speed of light, $h = 6.626 \times 10^{-34}$ J s is Planck's constant, and $k_b = 1.381 \times 10^{-23}$ J/K is Boltzmann's constant (Petty, 2006, p. 118).

At a given temperature, Planck's function peaks at a wavelength that is inversely proportional to the temperature (Wien's Displacement law, illustrated in Fig. 2.1), meaning that the peak emission of the sun occurs at much shorter wavelengths than the emission of Earth. This has given rise to the terms *shortwave radiation* and *longwave radiation*, referring to solar and terrestrial radiation, respectively. The blackbody emission curves (Planck function) as a function of wavelength at temperatures characteristic for the sun and Earth's surface and

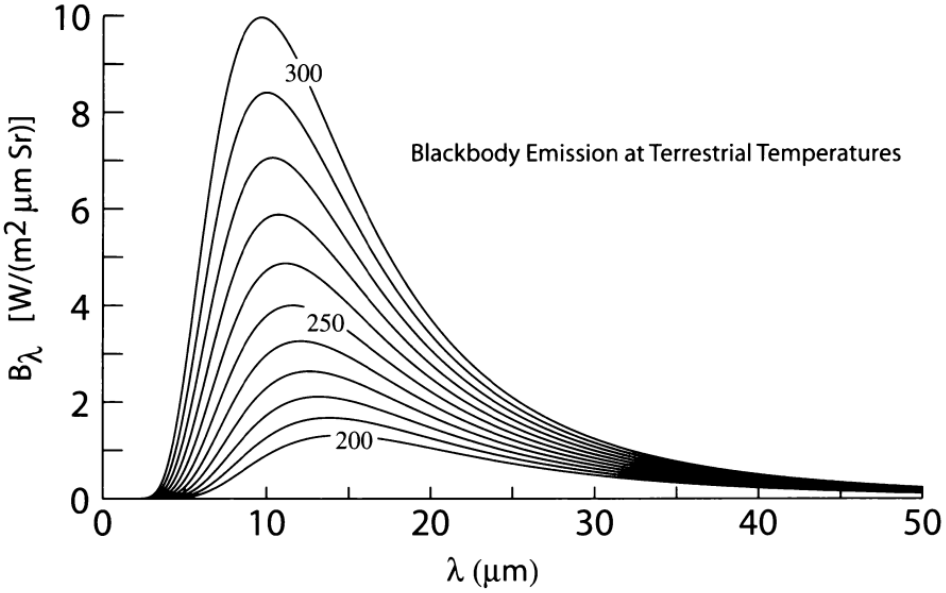


Figure 2.1: Planck function B_λ at temperatures that are typical for Earth's atmosphere. Source: Petty, 2006, Fig. 6.3.

atmosphere are shown in the top panel of Fig. 2.2. The curves for solar radiation (solid red line towards the left) and terrestrial radiation (solid lines towards the right) have been normalized to have equal areas. The separation of atmospheric radiation into shortwave and longwave components, with the cut-off around 4 μm, is a good approximation in the sense that more than 99% of the radiative energy of both sources is accounted for due to the overlap being small (Petty, 2006, p. 146) (this small overlap is actually accounted for by radiation schemes). Moreover, it allows radiation schemes to treat the two independently. For instance, in the shortwave (SW) it can be assumed that the only source is incident at the top-of-atmosphere, while longwave (LW) radiation computations must account for sources at each layer in the atmosphere, but often ignore scattering (which cannot be done in the SW).

Absorption and emission are related (or inverse) processes, as is evident in Kirchhoff's law:

$$\varepsilon_\lambda(\theta, \phi) = a_\lambda(\theta, \phi) \tag{2.4}$$

where θ and ϕ are the zenith and azimuthal angles, respectively, in a spherical coordinate system. Therefore, the law states that directional, monochromatic emissivities ε and absorptivities a are equal (a good absorber is also a good emitter, and vise versa). Absorptivity and emissivity are unitless, fractional quantities. A perfect absorber ($\varepsilon = a = 1$) is known as a blackbody and emits the theo-

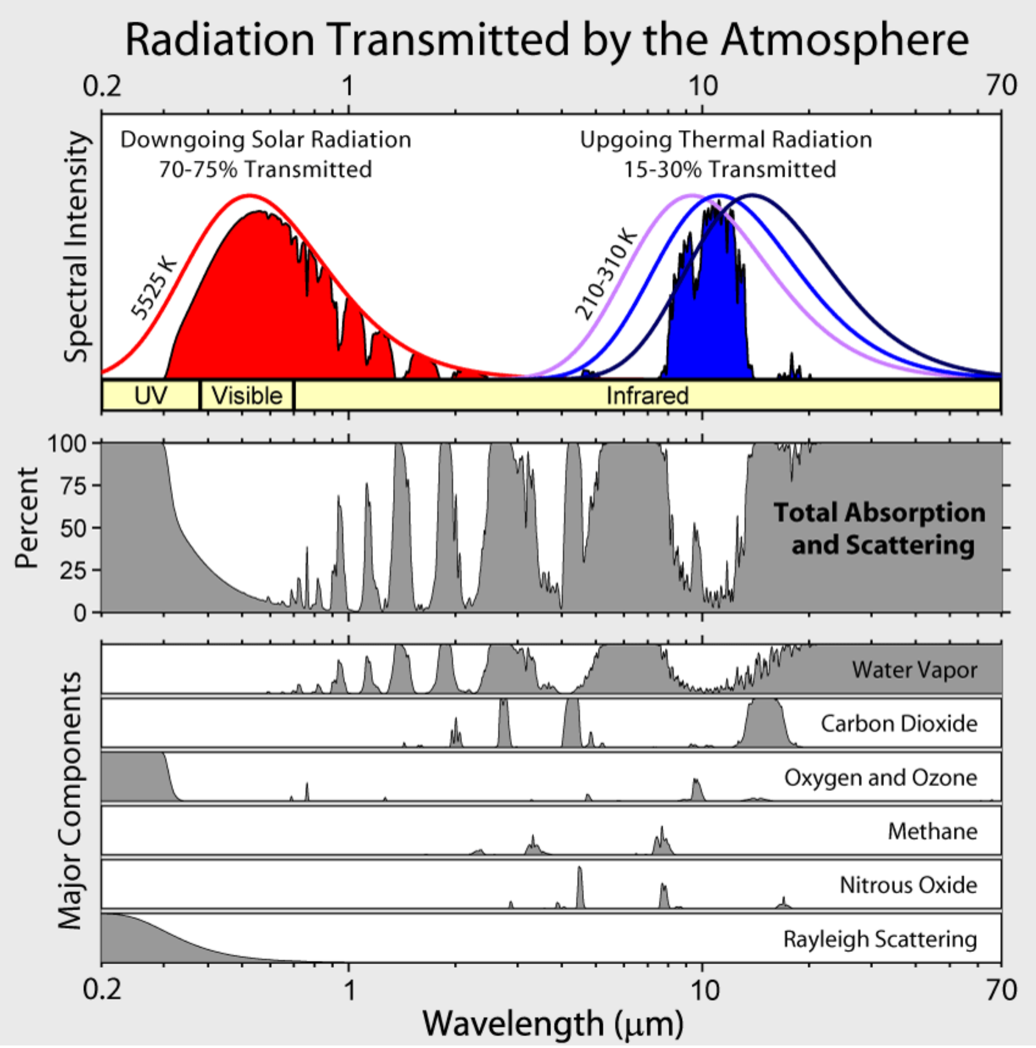


Figure 2.2: Absorption and emission of radiation in the cloud-free atmosphere by wavelength and different constituents. Source: Wikimedia commons, originally prepared by Robert A. Rohde for the Global Warming Art project.

retical maximum amount of thermal radiation as described by Planck's function.

The radiatively dominant gases in the atmosphere and their individual contributions to absorption across the shortwave and longwave spectrum in clear-sky conditions are shown in Fig. 2.2 (lower panels). Radiatively active gases in the thermal infrared range are known as greenhouse gases due to their absorption and re-emission of longwave radiation having a warming effect on the surface temperature of the Earth (not a perfect analogy, since actual greenhouses mainly trap heat by suppressing convective heat transfer, i.e. movement of fluid). Also shown in the Figure is the combined effect of the major constituents on downwelling solar radiation and upwelling longwave radiation, as drawn underneath the Planck functions described earlier in the upper panel. It is worth noting that data used to produce the figure is based on calculations which only include the effect of absorption and Rayleigh scattering on direct vertical transmission. In particular, the upgoing longwave radiation based on satellite observations differs from that in the Figure not only due to the presence of clouds and aerosols, but because the atmosphere itself emits radiation. In polar regions, the dips in the emission spectra in regions of strong gas absorption actually become bumps of enhanced emission due to the surface and lower atmosphere being colder than the atmosphere above (Fig. 2.3).

Understanding the absorption spectra of molecules requires delving into laws of quantum mechanics and the electronic, vibrational and rotational energy levels of molecules. This is not done here; for the purposes of the present work it is more relevant to simply know that these spectra - the intensity of absorption as a function of wavelength - are very complex and highly variable, consisting of up to millions of narrow *spectral lines* that are empirically determined (through absorption spectroscopy). Atmospheric radiative transfer models cannot afford to resolve this kind of detail in the absorption and emission spectrum of atmospheric constituents, and instead rely on parameterization (Section 2.1.5). Although it is not necessary to delve into the details of absorption by molecules, the physical coefficients related to absorption and scattering, and how they relate to atmospheric transmission are important and described in the next section.

2.1.2 Atmospheric transmission

A fundamental law of radiative transfer is that of exponential attenuation. Consider a monodirectional, monochromatic beam of radiation directed along the x-axis in a homogeneous medium, where the irradiance at $x = 0$ is F_0. The law may easily be derived by subdividing the distance x into N identical slices of thickness $\Delta x = x/N$ and recognizing that if Δx is sufficiently small the attenuation of the beam is proportional to Δx and F_0 (Bohren and Clothiaux, 2006, p. 51):

$$F_0 - F_1 \propto F_0 \Delta x \tag{2.5}$$

$$F_0 - F_1 = F_0 \beta_a \Delta x \tag{2.6}$$

where the proportionality constant is the *absorption coefficient* β_a in the case of attenuation by absorption. Rewriting as $F_1 = F_0(1 - \beta\Delta x)$, it is clear that the transmission over a distance $x = N\Delta x$, assuming the transmission by each slab is independent, is

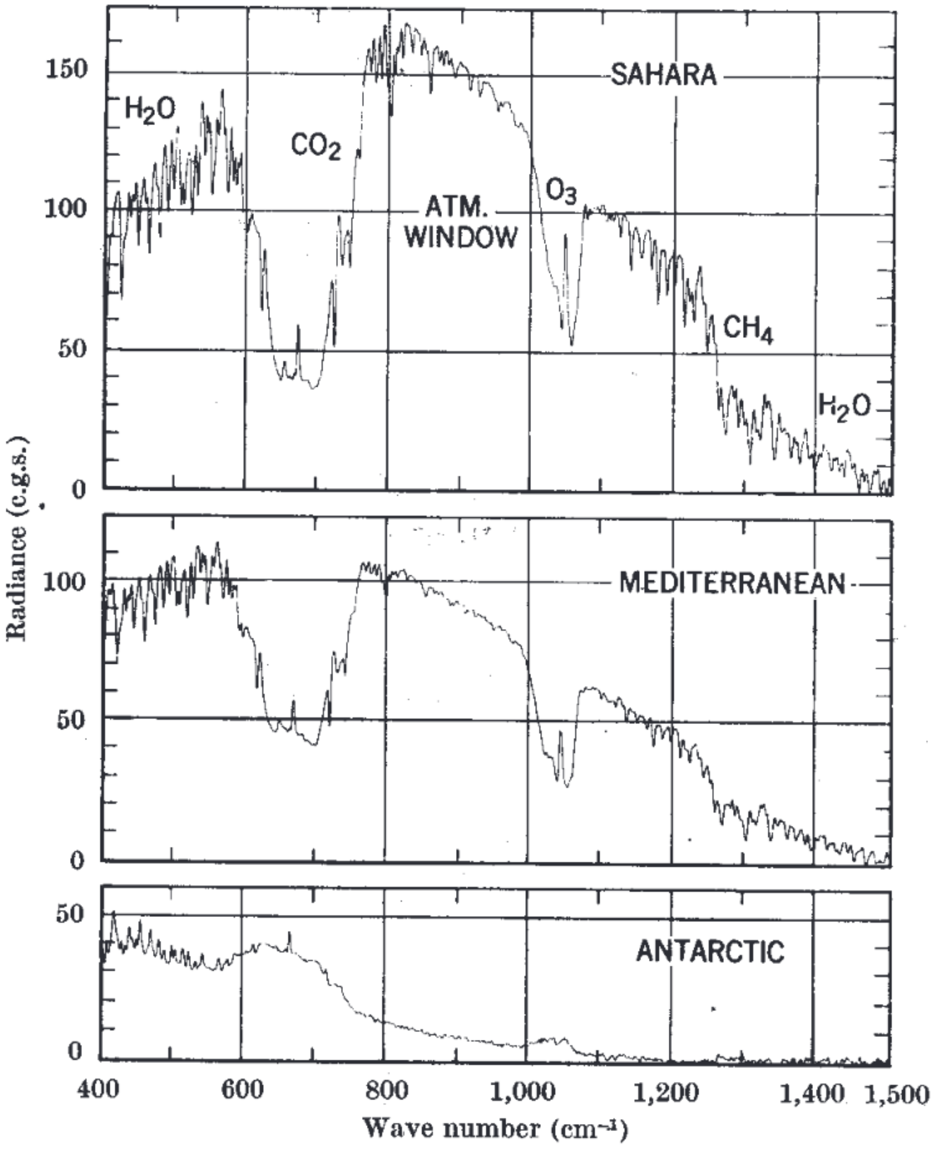


Fig. 1. Selected spectra obtained with the Michelson interferometer flown on the Nimbus 4 satellite. The three spectra shown here, obtained over the Sahara Desert, the Mediterranean and the Antarctic, are apodized, resulting in a resolution equivalent to 2·8 cm^{-1}. Atmospheric constituents responsible for the various absorption features are indicated in the upper spectrum.

Figure 2.3: Terrestrial emission spectra over different regions measured by the Nimbus 4 satellite. Figure taken from Hanel and Conrath (1970).

$$F_N = F_0(1 - \beta\Delta x)^N = F_0(1 - \beta_a x/N)^N \tag{2.7}$$

In the limit as N approaches infinity

$$F = \lim_{N\to+\infty} F_0(1 - \beta_a \Delta x)^N \tag{2.8}$$

Recognizing that the right hand side resembles the definition of the exponential function $exp(\xi) = \lim_{n\to+\infty}(1 + \xi/n)^n$,, we arrive at the law of exponential attenuation

$$F = F_0 \exp(-\beta_a x) \tag{2.9}$$

More commonly, the medium will not be homogeneous, and the extinction is obtained by integrating over the path:

$$F(x_1) = F(x_0) \exp\left[\int_{x_0}^{x_1} -\beta_a(x)dx\right] \tag{2.10}$$

The integral quantity is called the *optical depth* between points x_0 and x_1

$$\tau(x_0, x_1) = \int_{x_0}^{x_1} -\beta_a(x)dx \tag{2.11}$$

and the *transmittance* is

$$T(x_0, x_1) = e^{-\tau(x_0,x_1)}, \tag{2.12}$$

where T ranges from zero (for $\tau \to +\infty$) to 1 at $\tau = 0$, and Eq. 2.10 becomes $F(x_1) = T(x_0, x_1)F(x_0)$.

The exponential attenuation law, also known as Beer's law (or the Beer-Lambert law) is a fundamental aspect of radiative transfer, and significant also in a computational sense due to the large computational expense of the exponential function. As a consequence, the way it's implemented in the radiation code becomes important, as is discussed in Paper 4.

A very useful assumption in modeling atmospheric radiative transfer is that the extinction coefficient (and other optical properties) does not vary in the horizontal direction, but only in the vertical direction *z*. More generally, treating the atmosphere as *plane parallel* ignores horizontal variations in the structure of the atmosphere so that radiative properties are assumed to only depend on the vertical coordinate (Petty, 2006, p. 170). The plane-parallel assumption allows expressing slant paths at a zenith angle θ as $s = \frac{z}{\mu} = \frac{z}{cos\theta}$, and as well as using optical depth as a vertical coordinate in radiative transfer computations.

Radiation can be attenuated as it passes through a medium not only due to absorption but also *scattering*, which refers to photons being redirected from their original direction of propagation, usually due to interactions with particles. A coefficient similar to β_a but for scattering, which also in general depends on both the medium and the wavelength, can be defined: *scattering coefficient* β_s. The contributions of absorption and scattering to extinction can furthermore be combined in an *extinction coefficient* β_e:

$$\beta_e = \beta_a + \beta_s \tag{2.13}$$

The relative importance of scattering versus absorption in a medium is characterized by the *single-scattering albedo* ω:

$$\omega = \frac{\beta_s}{\beta_e} = \frac{\beta_s}{\beta_a + \beta_s} \tag{2.14}$$

Optical depth and single-scattering albedo are examples of **optical properties** which characterize how a material or medium interacts with radiation. The optical properties of the atmosphere are determined by its physical properties, including temperature, pressure, and the concentrations of gases, aerosols and cloud droplets. The first step in determining how the atmosphere interacts with radiation is computing such optical properties at every layer in the atmosphere. In the absence of scattering (an assumption often made in longwave radiation computations) only the optical depth is required. For computations with scattering under the two-stream approximation, which is also described further below, the optical properties are determined by three variables: optical depth, single-scattering albedo, and the *asymmetry parameter* which will be defined later. A particular focus in this dissertation is the optical properties of gases. These consist of optical depth in the longwave, and of optical depth and single-scattering albedo in the shortwave, where scattering is in the Rayleigh regime due to gas molecules being much smaller than the wavelength of the incoming light.

The attenuation in eq. 2.6 was defined in terms of an absorption coefficient β_a and path length dx. Another way of defining extinction, which can be useful in atmospheric models, is with respect to the *number concentration* (or number density) of absorbing (or scattering) particles N

$$\beta_e = \sigma_e N \tag{2.15}$$

where β_e is the extinction coefficient from before with units m^{-1}, and the new constant of proportionality σ_e is the extinction *cross-section* with units m^2 (and N has units of m^{-3}). The extinction cross-section can be interpreted as the effective area of a particle that results in some of the incident beam being absorbed or scattered. σ_e can be larger than the geometrical cross-sectional area of the particle.

2.1.3 Scattering and radiative transfer equation

Previously, the effect of scattering to attenuate radiation was considered. However, scattering can also be a *source* of radiation as photons of different light beams are deflected into the direction being considered. Moreover, for many applications, such as shortwave radiative transfer in the presence of clouds, it must be considered that photons can be scattered more than once (*multiple scattering*), which makes the problem considerably more difficult.

Combining the effects of extinction due to absorption and scattering (a sink), a source term due to emission, and a source term due to radiation being scattered *into* the beam, the change in intensity dI along an infinitesimal path is given by:

$$dI = dI_{ext} + dI_{emis} + dI_{scat} \tag{2.16}$$

where $dI_{ext} = -\beta_e I ds$ as in eq. 2.6. The source term needs to include the scattering from all possible directions Ω' into the direction of interest Ω:

$$dI_{scat} = \frac{\beta_s}{4\pi} \int_{4\pi} p(\Omega', \Omega) I(\Omega') d\omega' ds \tag{2.17}$$

In this expression the 4π steradians of solid angle $d\Omega'$ in a full sphere have been integrated, and $p(\Omega', \Omega)$ is the probability that radiation from direction Ω' is scattered into Ω, known as the *phase function.* The phase function must satisfy the normalisation condition

$$\frac{1}{4\pi} \int_{4\pi} p(\Omega', \Omega) I(\Omega') d\Omega' = 1. \tag{2.18}$$

Dividing by $d\tau = -\beta_e ds$ yields the *general* form of the radiative transfer equation (RTE) (Petty, 2006, p. 323):

$$\frac{dI(\Omega'}{d\tau} = I(\Omega) - (1-\omega)B - \frac{\omega}{4\pi} \int_{4\pi} p(\Omega', \Omega) I(\Omega') d\Omega' \tag{2.19}$$

A useful simplification for the phase function can be made when particles are spherical or randomly oriented, in which case only the angle Θ between Ω and Ω' matters and $p(\Omega', \Omega)$ can be replaced with $p(cos\Theta)$. *Isotropic* scattering, with all directions equally as likely to be scattered into, is the simplest phase function ($p(cos\Theta)$ = 1).

2.1.4 Two-stream approximation

The general form of the RTE is too complex to lend itself to analytical solutions. The major step done to simplify the RTE in atmospheric models is to assume that the radiation field only consists of irradiances in two directions, upward and downward. The two-stream approximation is useful when we are concerned with upward and downward fluxes $F_\downarrow$, $F_\uparrow$ instead of intensities.

Immediately this simplifies the phase function, as only the relative proportion of photons scattered in forward versus backward direction (relative to the original direction of travel) is of interest. This is captured in the *asymmetry parameter* g:

$$g = \frac{1}{4\pi}\int_{4\pi} p(cos\Theta)cos\Theta d\Omega \tag{2.20}$$

which can be interpreted as the average cosine of the scattering angle $cos\Theta$ and varies between -1 (photons scatter fully in the backward direction) and 1 (photons scatter fully forward). In the case of isotropic scattering, scattering into both hemispheres is equally as likely and $g = 0$.

The two-stream equations can be derived from the RTE when integrating radiances over two hemispheres and making further assumptions such as elastic scattering, and are (Meador and Weaver, 1980):

$$\frac{dF\downarrow(\tau)}{d\tau} = \int_0^1 I(\tau,\mu)d\mu - \frac{1}{2}\int_0^1\int_{-1}^1 p(\mu,\mu')d\mu' d\mu - \pi F\omega_0\beta_0 e^{-\tau/\mu_0} \tag{2.21}$$

$$\frac{dF\uparrow(\tau)}{d\tau} = -\int_0^1 I(\tau,-\mu)d\mu + \frac{1}{2}\int_0^1\int_{-1}^1 p(-\mu,\mu')d\mu' d\mu + \pi F\omega_0(1-\beta_0)e^{-\tau/\mu_0} \tag{2.22}$$

$$\tag{2.23}$$

where the last terms on the right-hand side represent the contribution from direct radiation incident at a layer. The exact solutions to these equations come in different flavours depending on specific assumptions regarding the phase function, etc., and are described in Meador and Weaver (1980).

2.1.5 K-distribution method

Because the gas absorption spectra of the atmosphere contains hundreds of thousands of absorption lines, computing broadband fluxes across the full shortwave and longwave spectrum *line by line* entails doing this many independent monochromatic computations. This is far too costly to do in weather and climate models which need to perform radiative transfer computations at a high spatial and temporal frequency (for instance in every grid column). Fortunately, alternatives to line-by-line methods exist, such as *band transmission models*, which are based on finding analytic approximations to the band-averaged transmittance over a path by using a statistical model for the distribution of line positions and strengths in the spectral interval, or band (Petty, 2006, p. 299). However, the most accurate and efficient method to deal with the spectral integration to date is the *k*-distribution method, and the related correlated-*k* approximation.

The *k*-distribution method is based on the idea that the highly variable function of absorption coefficient as a function of wavenumber $k(\nu)$ (or wavelength or frequency; here wavenumber is used following Petty, 2006) in a spectral interval can be *reordered* into a cumulative probability distribution which is far easier to integrate numerically. The spectral-mean transmission in a homogeneous atmospheric layer of mass path u is given by:

$$T(u) = \frac{1}{\nu_2 - \nu 1} \int_{\nu_1}^{\nu_2} exp\left[-k(\nu)u\right] d\nu \tag{2.24}$$

The numerical integration would typically have a sum of the form (Petty, 2006, p. 300):

$$T(u) \approx \sum_{i=1}^{N} \alpha_i exp\left[-k(\nu_i)u\right] \tag{2.25}$$

where α_i are weights associated with a specific quadrature method and *N* is the number of wavenumbers, which would typically need to be large so that individual lines can be resolved. Crucially, the *order* in which the terms are summed does not matter. Therefore, the *k*-distribution method replaces the function $k(\nu)$ with a new function over g that ranges from $g = 0$ for the smallest value of k to $g = 1$ for the largest value of k:

$$T(u) = \int_{0}^{1} exp\left[-k(g)u\right] dg \tag{2.26}$$

Here $g(k)$ is the cumulative probability function, giving the fraction of the absorption coefficients that are smaller than *k* in the interval. Because the new function is smooth and monotonically increasing, it can be numerically integrated using a small number of quadrature terms. These discrete *g*-points, also known as *k*-terms, represent similar *k*, which are likely to correspond to many different frequencies, grouped together.

This method is exact, but applying it to an inhomogenous path like an atmospheric column requires making an approximation. This is because a change in temperature and pressure changes $k(\nu)$ somewhat, and therefore also the mapping $\rightarrow g$ (for instance, a given *g*-point may correspond to different wavenumbers at two different pressure levels). In particular, large changes in pressure affect the mapping due to broadening of absorption lines due to collisions between molecules, an effect known as pressure broadening. In the correlated-*k*

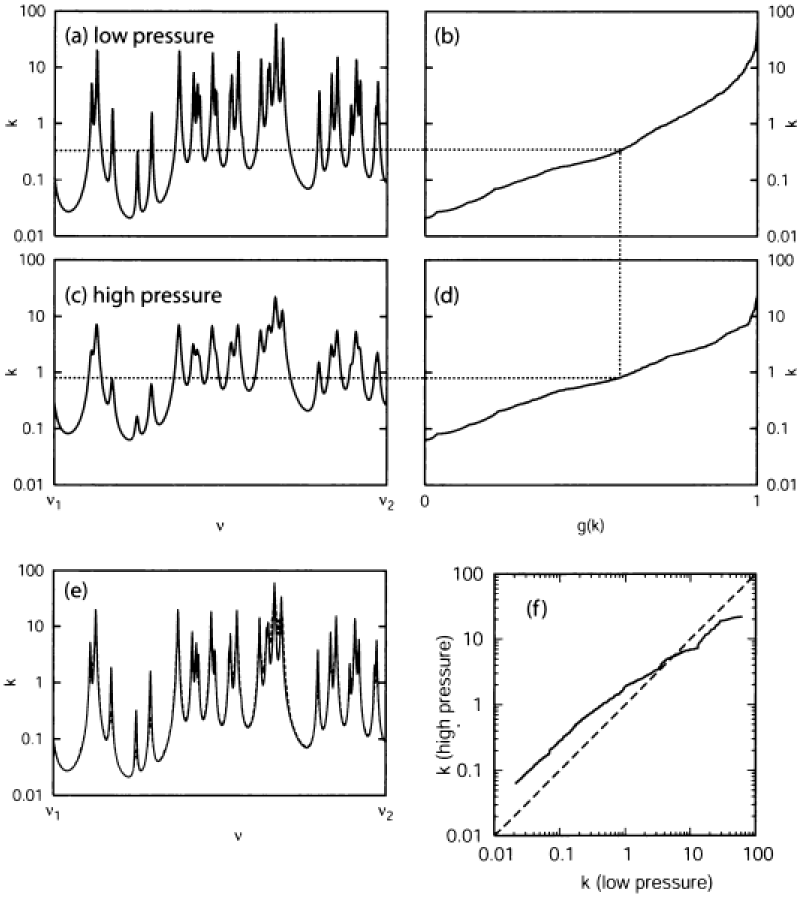


Fig. 10.5: Illustration of the *k*-distribution method and its extension, the correlated-*k* method. (a) A hypothetical spectrum of absorption coefficient *k* at relatively low pressure. (b) By sampling the spectrum at fine intervals and then sorting the results so that *k* increases monotonically, we define the function $0 \leq g(k) \leq 1$ (horizontal axis). Panels (c) and (d) are the same as (a) and (b) except with stronger pressure broadening. (e) Comparison of the actual spectrum for low pressure [from panel (a)] (solid curve) with one estimated from the spectrum at higher pressure [panel (c)] (dotted curve), using the mapping in panel (f). (f) The mapping between *k* values at the two pressure levels, based on equal values of *g*.

Figure 2.4: Figure and caption from Petty, 2006, p. 301, illustrating the k-distribution method and correlated-k approximation.

approximation, it is assumed that the $k \rightarrow g$ mapping is identical in adjacent layers. The error associated with this approximation is typically quite small, as is nicely illustrated in Petty, 2006, Fig. 10.5 (reprinted here in Figure 2.4), which compares the k-distributions obtained for a hypothetical spectral interval at a low pressure to that obtained at a pressure which is three times higher. As seen in the lower right subfigure, the $k \rightarrow g$ mapping at the two pressure levels are highly correlated overall.

Modern radiative transfer codes used in NWP and climate models generally use some variant of the correlated *k*-distribution (CKD) method. However, CKD schemes may differ quite substantially from one another. Among the many considerations in how to design a CKD model is how many *g*-points and bands to use, which band boundaries to select, and how the contributions to absorption from multiple gases are combined. For example, bands may be chosen so as to limit the number of active gases in each band, or longwave bands may be required to be narrow enough that the Planck function can be assumed to be constant (Hogan and Matricardi, 2020). Another design choice is in the reordering from wavenumber to *g*-space which may be done separately for each pressure level, or just using a single mapping. Using a single mapping means that the error from assuming correlated mappings is eliminated, but each discrete *g*-point maps to a greater range of *k* and the discrete *k* become less representable for each subinterval, which is another source of error in CKD (Mlawer et al., 1997). How such choices affect the accuracy and efficiency of CKD models is explored in the Correlated K-Distribution Model Intercomparison Project, or CKDMIP (Hogan and Matricardi, 2020). Hogan and Matricardi (2022) point out that the gas optics module of the radiation scheme (which today is generally based on CKD) may be the most fundamental part of a climate model, due to its importance in determining the climatic impact of greenhouse gases. Crucially, since the gas optics scheme dictates the spectral resolution, it also largely determines the computational cost of the whole radiation scheme. Moreover, since radiation is one of the most expensive components in climate models, it follows that the gas optics scheme plays an outsized role in its computational expense and that reducing the number of *g*-points can make the entire climate model significantly cheaper to run!

2.1.6 State-of-the-art radiation parameterizations

Two radiation schemes featured in this work are RTE+RRTMGP (Pincus et al., 2019) and ecRAD (Hogan and Bozzo, 2018). Both of these are modern radiation codes designed around flexibility and modularity, and now include recently developed CKD-based gas optics schemes based on state-of-the-art spectroscopy. The radiation schemes also utilise the same variant of the two-stream method to compute direct and diffuse reflectances and transmittances. The main difference is that ecRAD is a more complete and mature radiation package, and includes several radiative transfer solvers, ways to treat cloud overlap and heterogeneity, and different gas, aerosol and cloud optics schemes. This separation of physical concerns in radiative transfer computations, which entails computing the optical properties of gases, aerosols and clouds separately and then combine them in the *solver* which solves the radiative transfer equation, is what is meant by modularity. A modular structure is highly convenient as it allows changing individual components independently of other components, and for the user to combine them freely in various configurations.

RTE+RRTMGP is recently released radiation code currently consisting of a gas optics package (RRTMGP) and radiative transfer solver (RTE). As this scheme is described in the published papers (Chapters 3-4), only ecRAD is described here.

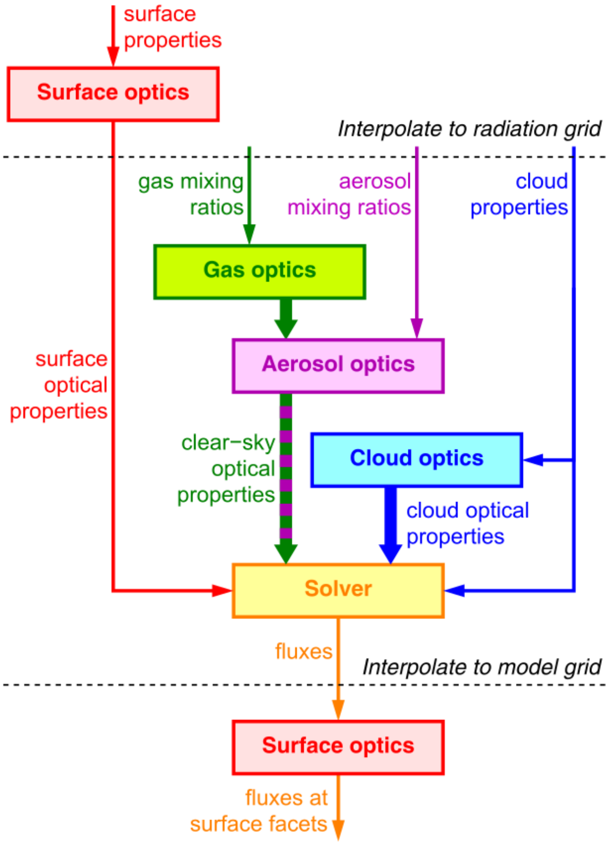


Figure 2.5: Schematic of the ecRAD scheme and the flow of data between different components. Figure taken from Hogan and Bozzo (2018).

2.1.6.1 ecRAD

ecRAD is a radiation scheme developed at the European Centre for Medium-Range Weather Forecasts (ECMWF), and used in their global weather model, the Integrated Forecast System (IFS). The structure of the scheme and the way it interacts with the IFS is shown is illustrated in Figure 2.5. In the high-resolution deterministic forecasts, the radiation scheme is called every hour, and the atmospheric variables are interpolated to a grid with 10.24 times fewer columns than the rest of the model (Hogan and Bozzo, 2018). First, the gas optics scheme computes gas absorption optical depth (SW and LW), single-scattering albedo ω from Rayleigh scattering (SW), and sources (Planck functions in the LW and incoming flux in the SW) on a *g*-point / height / column grid. The aerosol optics scheme increments these optical properties with the contribution from aerosols, and optionally computes ω and asymmetry factor g at each LW *g*-point. The cloud optics scheme computes cloud τ in each SW and LW band, and ω and g in SW band (optionally LW). (Longwave scattering can either be turned off, turned on for clouds, or turned on for both aerosols and clouds). These optical properties are then passed to the solver alongside surface optical properties. The solver combines the cloud and clear-sky optical properties in a specific manner depending on assumptions of cloud overlap and structure, and computes irradiances (*g*-point fluxes) and finally broadband fluxes.

ecRAD improves upon the previous ECMWF radiation scheme, McRAD (Morcrette et al., 2008), in various ways and is roughly 40% faster (Hogan and Bozzo, 2018). The radiative transfer solvers in ecRAD, and a new gas optics scheme which makes the whole radiation scheme much faster still, are described below.

McICA (The Monte Carlo Independent Column Approximation) is used in many atmospheric models. It represents cloud heterogeneity stochastically via a cloud generator that uses N *g*-points to sample N sub-grid columns, where a sub-grid column is either cloudy or clear-sky (Pincus et al., 2003). This introduces noise, which has no measurable impact on seasonal forecasts, but can affect short-range forecasts of near-surface temperature Hogan and Bozzo (2018).

TripleClouds is a radiative transfer solver which represents sub-grid cloud structure in a deterministic way by dividing each atmospheric layer into three regions, two regions of cloud and one clear-sky region (Shonk and Hogan, 2008). It can represent arbitrary vertical overlap between the three regions in two adjacent levels (Shonk and Hogan, 2008). Because it represents cloud inhomogeneity deterministically, it does not suffer from the noise associated with McICA.

SPARTACUS (Speedy Algorithm for Radiative Transfer through Cloud Sides) is recently developed radiative transfer solver which accounts for cloud 3D radiative effects within model columns. Such 3D effects can be significant, increasing e.g. the longwave cloud radiative effect at the surface locally by around 30% (Hogan et al., 2016), but have previously been too expensive to represent in weather and climate models. SPARTACUS offers a fast way to compute 3D effects, while matching full 3D radiative transfer calculations for cumulus cloud fields quite closely. The solver uses the same subgrid cloud model as TripleClouds, i.e. two cloudy and one-clear sky region, but computes radiative transport through cloud sides by adding extra terms to the two-stream equations to represent lateral transport between clear and cloudy regions (Hogan et al., 2016). The coupled system of equations can be solved accurately by using matrix exponentials. These matrix exponentials are relatively expensive to compute, and on the whole ecRAD with SPARTACUS is roughly 6-8 times slower than with McICA, which is considered too costly to use in the IFS. A major focus of Paper 4 is rewriting the algorithm to improve the efficiency of these computations.

ECCKD is an innovative new gas optics scheme available in the development branch of ecRAD. The scheme is described in a paper by Hogan and Matricardi which is currently under review. Technically ECCKD is actually not a gas optics scheme but a tool for generating gas optics schemes that allows the user to define the bands and specify the range of greenhouse gas concentrations. However, here the ECCKD models available in the development branch of ecRAD are discussed.

ECCKD achieves high efficiency by using the full-spectrum correlated-*k* (FSCK) method in the longwave, which is a variant of CKD where bands are not used and instead the reordering is done across the whole spectrum. Furthermore, the method used in ECCKD differs from classical CKD in that a unique mapping from wavelength to *g*-space, instead of using a different reordering at each height. The spectral variation in cloud absorption in the near-infrared is represented by partitioning the parts of the spectrum that are optically thin to gases into three or more sub-bands, while allowing k terms for the optically thicker parts of the spectrum to span the entire near-infared (Hogan and Matricardi, 2022). By using the FSCK approach and selecting k-terms in a clever way, the number of k-terms can be kept very small: preliminary ECCKD use only 16-32 k-terms in the shortwave and 32 in the longwave. Yet, the evaluation from CKDMIP suggests that ECCKD models have a similar accuracy as other CKD schemes, which might use an order of magnitude more k-terms. The specific trick done in ECCKD to achieve high accuracy with fewer k-terms in doing the spectral reordering terms of the height of the peak heating/cooling rate, which ensures that the reordering is always done most accurately at the height where it matters most (Hogan 2022, personal communication). That is, the reordering is done so as to explicitly minimize a cost function that is very closely tied to the requirements of radiation schemes in dynamical models (which is not generally the case in the development of CKD models).

2.2 Artificial Neural networks

2.2.1 Feed-forward networks

Artificial neural networks are a type of computing system which very loosely resemble biological neural networks. NNs are essentially highly adaptive nonlinear statistical models, in fact it has been demonstrated that NNs can approximate virtually any smooth function, also known as being "universal approximators" (Gardner and Dorling, 1998). Structurally the NN can be regarded as a series (or *layers*) of functional transformations which together map input to output data. Each layer comprises of a certain number of neurons, or nodes, which are connected to the nodes in the next layer (Fig. 2.6). What makes the NN adaptable is that the layer-wise transformations depend on adjustable parameters (*weights*) which are optimized through a training process which seeks to minimize the difference between the model output and the outputs in some data set; that is, the model gradually becomes better by learning from labeled training examples. This procedure is known as supervised learning, algorithms which make use of unlabelled data (unsupervised learning) are not discussed here.

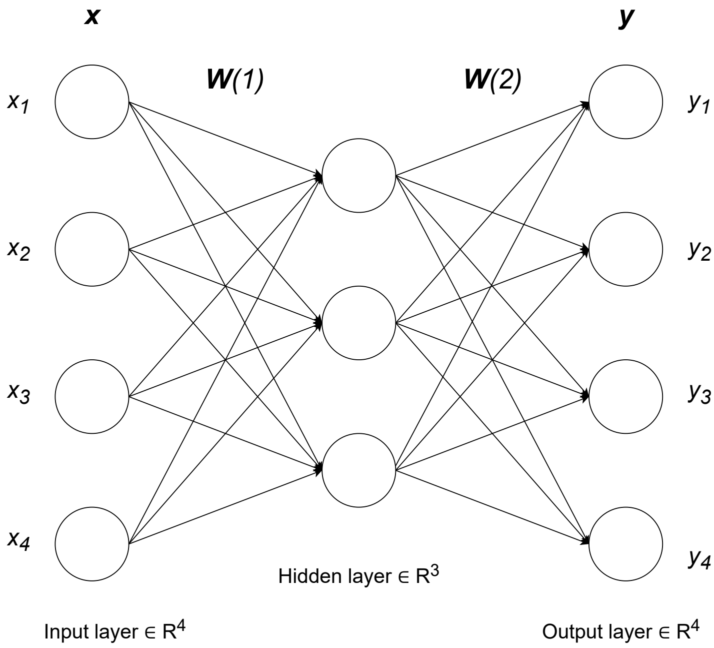


Figure 2.6: Illustration of a simple feed-forward NN with four inputs, three hidden neurons, and four outputs.

Following (Bishop, 2006), NNs can be described by first considering simple linear models for regression:

$$y(x, w) = f\left(\sum_{j=1}^{M} w_j \phi_j(x)\right) \tag{2.27}$$

Here $\phi_j(x)$ is called a *basis function* and the coefficients w_j can be adjusted to give the best fit to a dataset by minimizing an error function such as sum-of-squares . As can be seen from the equation, these models are based on linear combinations of basis functions. While the basis function itself can be non-linear (for instance, a polynomial), the models are linear in the coefficients *w*. In the case of simple regression, *f* is identity. In the case of classification, *f* is a nonlinear *activation function* such as a sigmoid function $f(a) = \frac{1}{1+e^{-a}}$.

If we now wish to make this model more general we can make the basis functions $\phi_j(x)$ themselves depend on adjustable parameters, along with the coefficients w_j. There are many ways to construct parametric nonlinear basic functions, but neural networks use basic functions of the form 2.27, so that the basic function itself is a non-linear function *f* of linear combinations of inputs Bishop (2006). In the first layer of the network, those inputs are the model inputs x_i:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{2.28}$$

where $j = 1, ..., M$ and the superscript (1) indicates that these parameters are in the first layer (Bishop, 2006). The parameters w_{ji} are referred to as weights and the parameters w_{j0} as biases. This output is then transformed using a nonlinear and differentiable activation function *f*(.):

$$h_j = f(a_j) \tag{2.29}$$

The transformed outputs h_j are called hidden units, which can again be linearly combined in a second layer:

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} h_j + w_{k0}^{(2)} \tag{2.30}$$

where $k = 1, ...K$ and K is the number of outputs. The output unit activations can also be transformed by an activation to give the final network outputs y_k, but for regression this function is usually identity: $y_k = a_k$. For classification problems, a logistic sigmoid function is typically used to constrain the output to a range of 0 to 1:

$$y_k = \sigma(a_k), \sigma(a) = \frac{1}{a + exp(-a)} \tag{2.31}$$

The above equations can be combined so that the overall network function takes the form (for sigmoidal output activations)

$$y_k(x, w) = \sigma(a_k) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} f\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{2.32}$$

The weights and bias parameters are here combined into the vector w. The layer structure has given rise to the name *multi-layer perceptrons* (MLP) for such models. The second often used term, *feed-forward* network, refers to the fact that information only flows in one direction (information only flows towards the right in Figure 2.6).

The objective during network training is to find a set of weights w which minimize an error function such as

$$E(w) = \frac{1}{2}\sum_{n=1}^{N} ||y(x_n, w) - t_n||^2. \tag{2.33}$$

The minimization of this function, sum of squares, is equivalent to maximizing the likelihood function (Bishop, 2006, p. 233). (Mean-squared-error has the same minimum, and is the most commonly used error or *loss* function.) Different strategies exist for this optimizing the weights, but the most successful and efficient are generally based on gradient descent. This entails computing the local gradients of the error surface and nudging the weights in the direction of the steepest gradient. After each nudge (updating the weights based on the gradient), the final output of the network is again calculated and compared with the right answer. This iterative procedure is followed until a minimum of the error surface is found. Depending on the problem it can be very difficult task to find a global minimum, as it's instead common to end up in local minimum. Various tricks can be used to try to avoid the latter from happening, such as adding a "momentum" term to the gradient (Rumelhart et al., 1986).

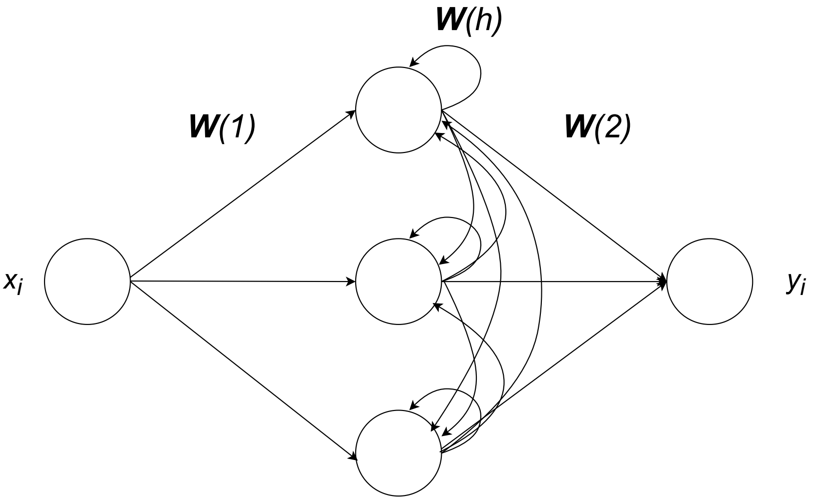
While neural networks are very flexible, they are prone to suffer from overfitting. Overfitting refers to the situation where a model has good performance on the training data, but performs poorly when presented with new data that was not seen during training. Overfitting is linked to model complexity (this is more generally also the case with linear regression methods, for example), as more model parameters makes it easier for the model to overfit to the training data. Overfitting can be combated with *regularization* methods such as drop-out, where randomly selected neurons are dropped out during training (set to zero). Another way of avoiding overfitting is using *early stopping*. Here, the error with respect to an independent validation data set is monitored during training, and training is stopped when the validation error begins to increase. Naturally, increasing the size of the training data can also help prevent overfitting.

2.2.2 Recurrent networks

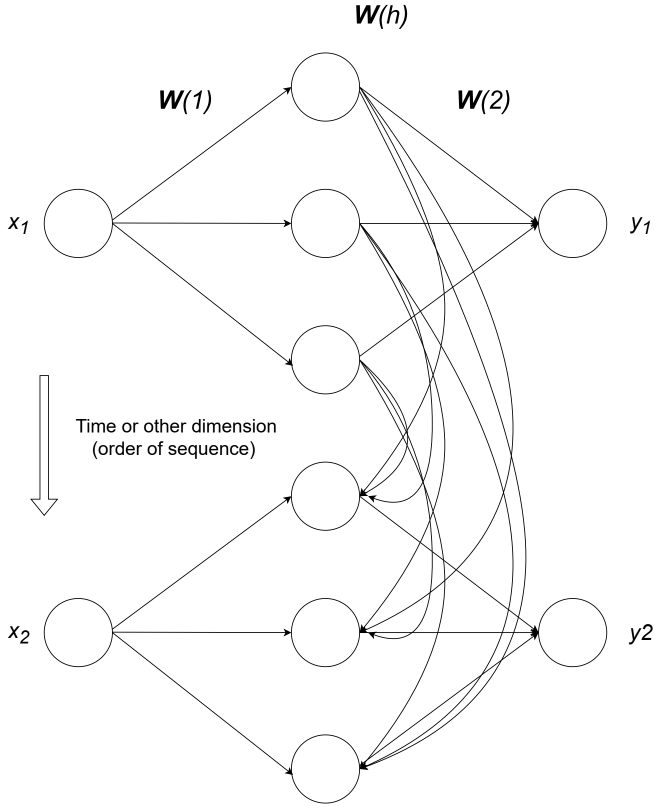
Let's say that we're trying to solve a problem of a sequential nature, such as trying to predict a likely next word in a sentence based on the previous words in the sentence. We could try to solve this problem with a FNN by concatenating all of the words (or some lower-dimensional vector representation of the words, known as word embeddings) into one long input vector of an FNN, and the output would then be a single word embedding. However, there are some clear problems with this approach. Since sentences have variable lengths, the length of the input vector would also change, which cannot happen in an FNN. One could get around this by some sort of padding, where all the sentences corresponding to samples in a training data set are padded to the length of the longest sentence, but this seems like a clumsy and inefficient solution, and the results would likely not be very good. The real problem is that the neural network structure does not account for the sequential nature of the problem (that sentences comprise a sequence of words).

A better strategy, then, would be to use another type of NN that could process the information sequentially, support varying input sizes, and whose model size does would not depend on the size of the input. These are characteristics of a recurrent neural network (RNN). RNNs consist of an internal state (memory) that allow it to be sequential by updating this internal state, which is just another *weight*, each time the RNN receives a new point in a sequence. This is similar to taking the layer output of equation 2.29 and feeding it back to the same layer as input, instead of to another layer as in the FNN. The RNN is schematically illustrated in Figure 2.7.

As alluded to by the example, RNNs are very popular for natural language processing (NLP). A quick literature search suggests their use in atmospheric science has been limited, with most examples dealing with various time series prediction problems. For instance, RNN have been applied to air quality prediction (Athira et al., 2018) and long-lead seasonal rainfall forecasting (Karamouz et al., 2008). In such cases, the RNNs process temporal sequences.



(a) A simple recurrent network with *sequential* scalar input x, scalar output y and three nodes in a single hidden layer. In addition to the weights connected to the inputs and outputs ($W(1)$ and $W(2)$ respectively), another set of weights $W(h)$ is used to update the hidden state of the hidden layer, h, upon each sequential iteration. The hidden state acts as memory. In the figure the weights of the hidden state are represented by the lines connecting the hidden nodes to themselves (a cyclical connection).



(b) *Unfolded* view of the same network.

Figure 2.7: A simple recurrent neural network (a), and the same network "unfolded in time" (b).

References

Athira, V., Geetha, P., Vinayakumar, R., and Soman, K. (2018). Deepairnet: Applying recurrent networks for air quality prediction. *Procedia computer science*, 132:1394–1403.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

Bohren, C. F. and Clothiaux, E. E. (2006). *Fundamentals of atmospheric radiation: an introduction with 400 problems*. John Wiley & Sons.

Gardner, M. W. and Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

Hanel, R. A. and Conrath, B. J. (1970). Thermal emission spectra of the earth and atmosphere from the nimbus 4 michelson interferometer experiment. *Nature*, 228(5267):143–145.

Hogan, R. J. and Bozzo, A. (2018). A flexible and efficient radiation scheme for the ecmwf model. *Journal of Advances in Modeling Earth Systems*, 10(8):1990–2008.

Hogan, R. J. and 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.

Hogan, R. J., Schäfer, S. A., Klinger, C., Chiu, J. C., and Mayer, B. (2016). Representing 3-d cloud radiation effects in two-stream schemes: 2. matrix formulation and broadband evaluation. *Journal of Geophysical Research: Atmospheres*, 121(14):8583–8599.

Karamouz, M., Razavi, S., and Araghinejad, S. (2008). Long-lead seasonal rainfall forecasting using time-delay recurrent neural networks: a case study. *Hydrological Processes: An International Journal*, 22(2):229–241.

Meador, W. and 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.

Mlawer, E. J., Taubman, S. J., Brown, P. D., Iacono, M. J., and Clough, S. A. (1997). Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *J. Geophys. Res.*, 102(D14):16663–16682.

Morcrette, J., Barker, H. W., Cole, J., Iacono, M. J., and Pincus, R. (2008). Impact of a new radiation package, mcrad, in the ecmwf integrated forecasting system. *Monthly weather review*, 136(12):4773–4798.

Petty, G. W. (2006). *A first course in atmospheric radiation*. Sundog Pub.

Pincus, R., Barker, H. W., and Morcrette, J.-J. (2003). A fast, flexible, approximate technique for computing radiative transfer in inhomogeneous cloud fields. *Journal of Geophysical Research: Atmospheres*, 108(D13).

Pincus, R., Mlawer, E. J., and Delamere, J. S. (2019). Balancing accuracy, efficiency, and flexibility in radiation calculations for dynamical models. *Journal of Advances in Modeling Earth Systems*, 11(10):3074–3089.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Shonk, J. K. and Hogan, R. J. (2008). Tripleclouds: An efficient method for representing horizontal cloud inhomogeneity in 1d radiation schemes by using three regions at each height. *Journal of Climate*, 21(11):2352–2370.

3

Paper 1: Accelerating Radiation Computations for Dynamical Models With Targeted Machine Learning and Code Optimization

3.1 Motivation

A report from a workshop on radiation in the next generation of weather forecast models from 2018 states that: "Although neural networks may struggle to reproduce the features of a full radiation scheme, machine learning might be useful to provide an efficient means to correct the broadband fluxes to account for, for example, 3D radiative effects" (Hogan, 2018). While the indication that radiation experts are skeptical of machine learning eventually replacing physical radiation codes may seem incongruous with the growing number of studies on this topic seen in the last decade - and some might even say at odds with some of the published results - it should not be surprising. Domain experts are after all painfully aware of the sensitive and precise nature of radiation computations and the sensitivity and stability of large-scale models with respect to the results from those computations. For instance, since radiation schemes compute radiative flows of energy, it is highly important that they are energy conserving. Climate models used to study future changes in climate also require the radiative forcings of greenhouse gases to be computed at high accuracy. In general, radiation parameterizations differ from many other parameterizations in that they use equations which represent an "exact" solution to the radiative transfer problem (albeit under highly simplifying assumptions). While neural networks can be accurate, because they are data-driven and do not traditionally incorporate any physical laws or equations (although they can be trained to minimize a physical cost function), they can hardly be claimed to be "exact" nor energy conserving.

It is against this backdrop of the challenge of emulating a radiative transfer scheme using ML that the initial direction of the research was settled towards working on a smaller problem - a subcomponent of a radiation scheme which is more empirical, and therefore in theory highly suitable for NNs (gas optics). This direction was taken despite it offering a smaller potential speedup and several papers, dating back even several decades (Chevallier et al., 1998) demonstrating that NNs can in fact predict broadband flux and/or heating rate profiles with seemingly decent accuracy. (For instance in terms of small mean percentage errors with respect to the predicted quantities in an offline evaluation, or being able to produce climate simulations that are stable and realistic.)

The counter-argument to this is that "accurate" is both difficult to define and a relative concept when common metrics and data sets are absent. For an NN-based radiative transfer model to be used as a parameterization in a weather or climate model, in most cases it would need to be energy conserving, reliable and have a similar level of accuracy as existing parameterizations (with respect to benchmark radiation computations) across a wide range atmospheric conditions and even model configurations - in other words, "reproduce the features of a full radiation scheme" as described in the workshop report.

In the absence of this being demonstrated by existing literature, it was decided that a safer and more instructive approach would be to start with a smaller problem. The goal of the following paper was to develop a NN version of a modern gas optics scheme that would ideally reproduce its full features. Another seemingly independent but actually related research question which arose spontaneously when carrying out the work is the acceleration of radiation computations by refactoring (optimizing) existing radiation code.

ADVANCING EARTH AND SPACE SCIENCE

JAMES | Journal of Advances in Modeling Earth Systems

RESEARCH ARTICLE

10.1029/2020MS002226

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

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

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), $j = 1, \dots, M$ and M is the number of neurons in the layer, w_{ji} are referred to as weights, and $w_{j\,0}$ as biases. $x_1, \dots x_D$ 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 $k = 1, \dots, K$ 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

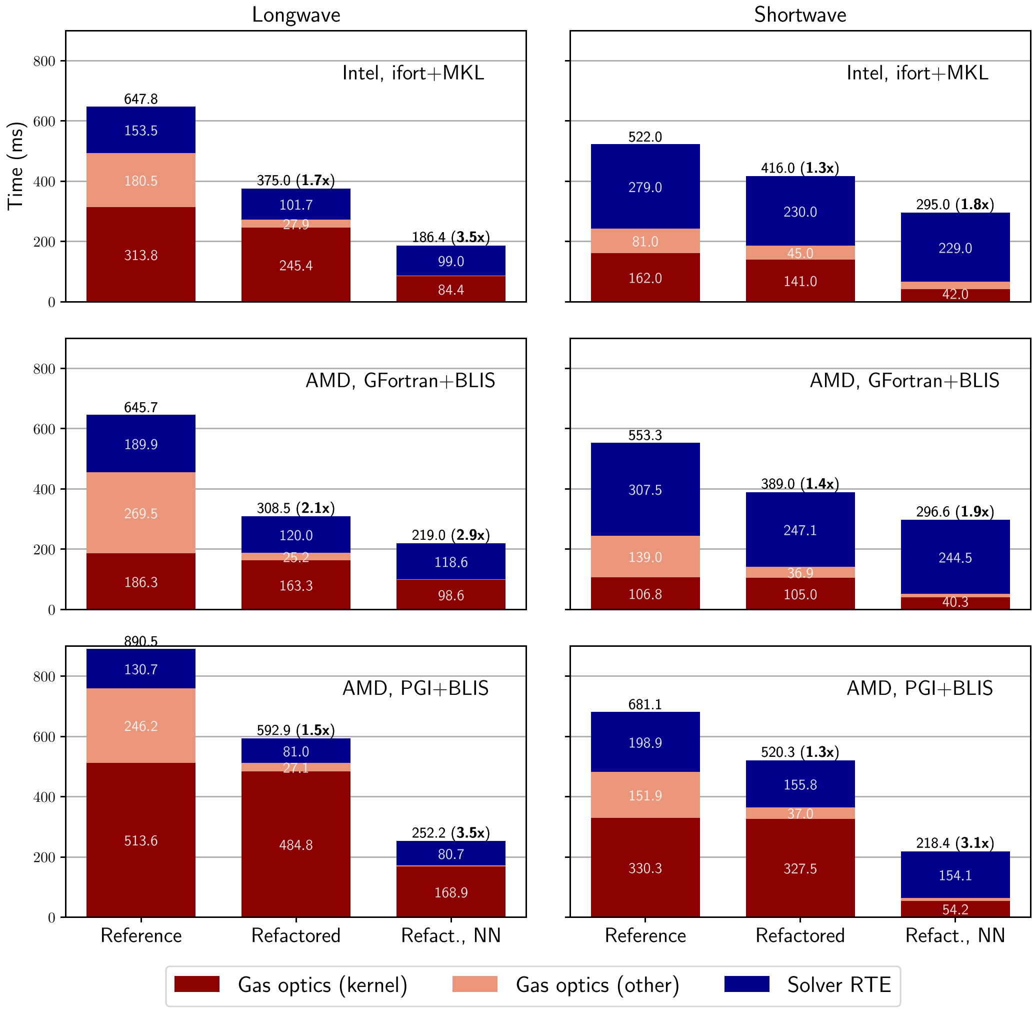


Figure 1. Mean elapsed time to compute the clear-sky longwave fluxes (left, without scattering) and shortwave fluxes (right, with scattering) for 1,600 RFMIP profiles. Top: Intel CPU, Intel Fortran compiler 19.1, and Intel MKL. Middle: AMD Ryzen CPU, GNU Fortran compiler 9.3, and AMD-Optimized BLIS 2.2. 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

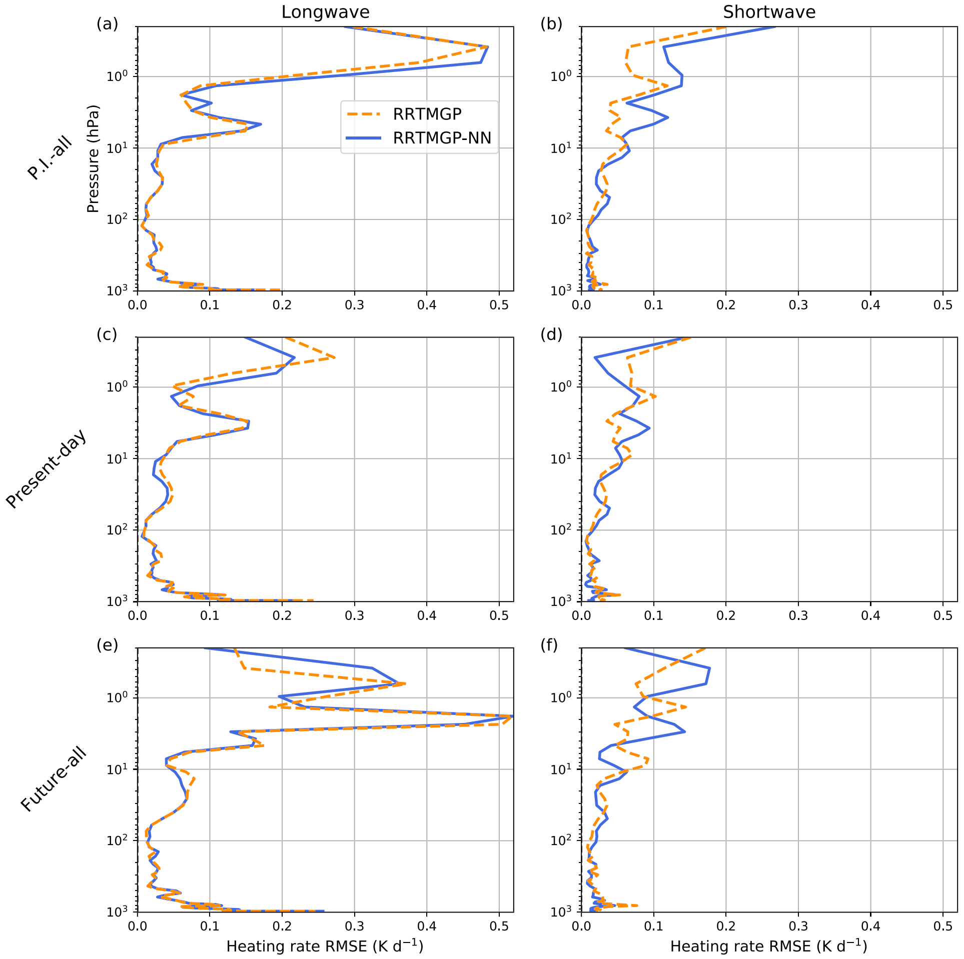


Figure 2. Root-mean-square errors in longwave (a, c, e) and shortwave (b, d, f) heating rates shown for both RRTMGP and RRTMGP-NN using 15 test profiles and 3 different experiments from RFMIP: preindustrial-all (a, b), present day (c, d), and future-all (e, f). The “all” suffix refers to perturbed temperature 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.

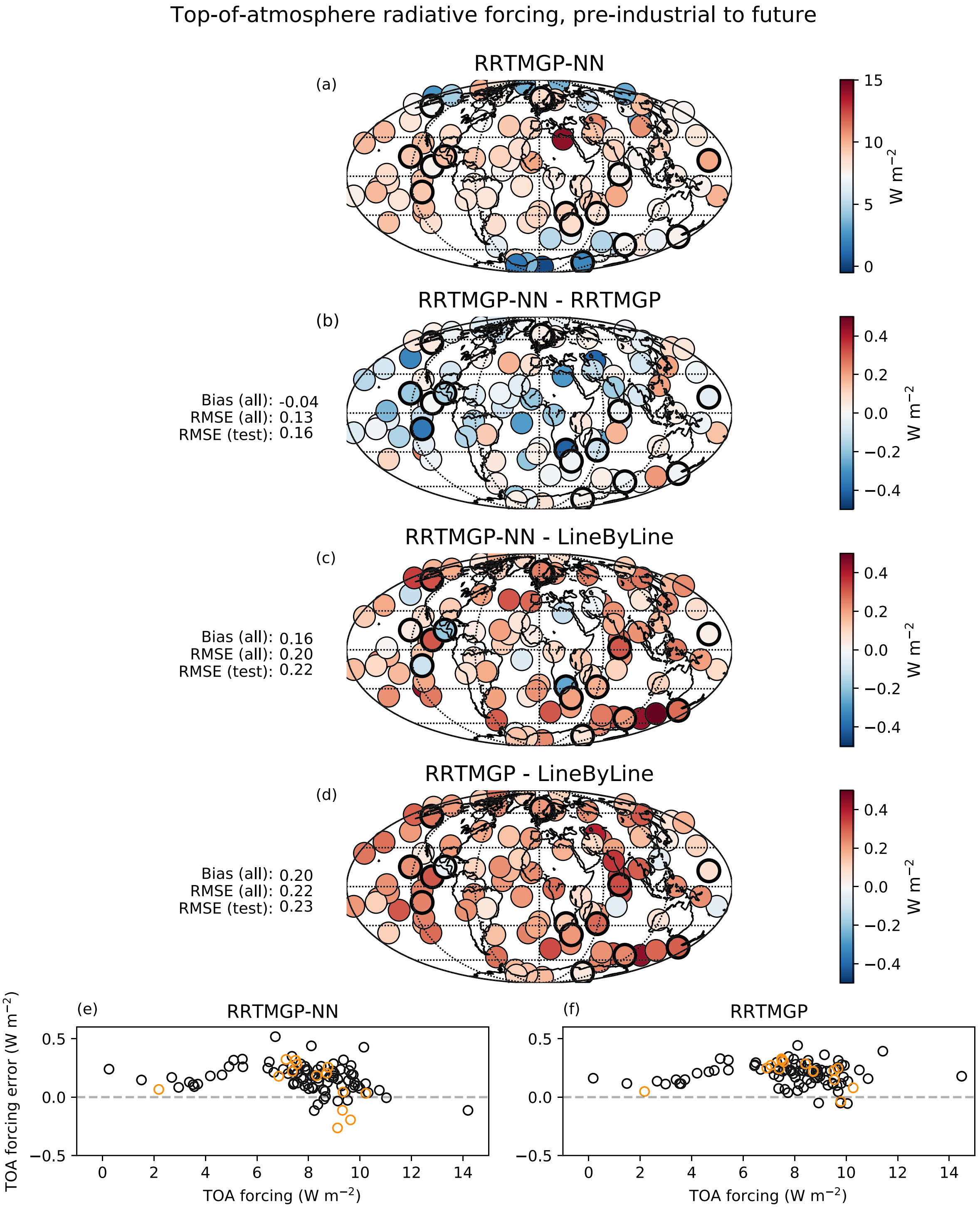


Figure 3. Instantaneous radiative forcing (IRF) i.e. the change in net fluxes at top of atmosphere between preindustrial and future RFMIP experiments for NN (a) and the difference in IRF between the NN and RRTMGP (b), the NN and LBL (c), and RRTMGP and LBL (d). The 15 test profiles are indicated with 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.

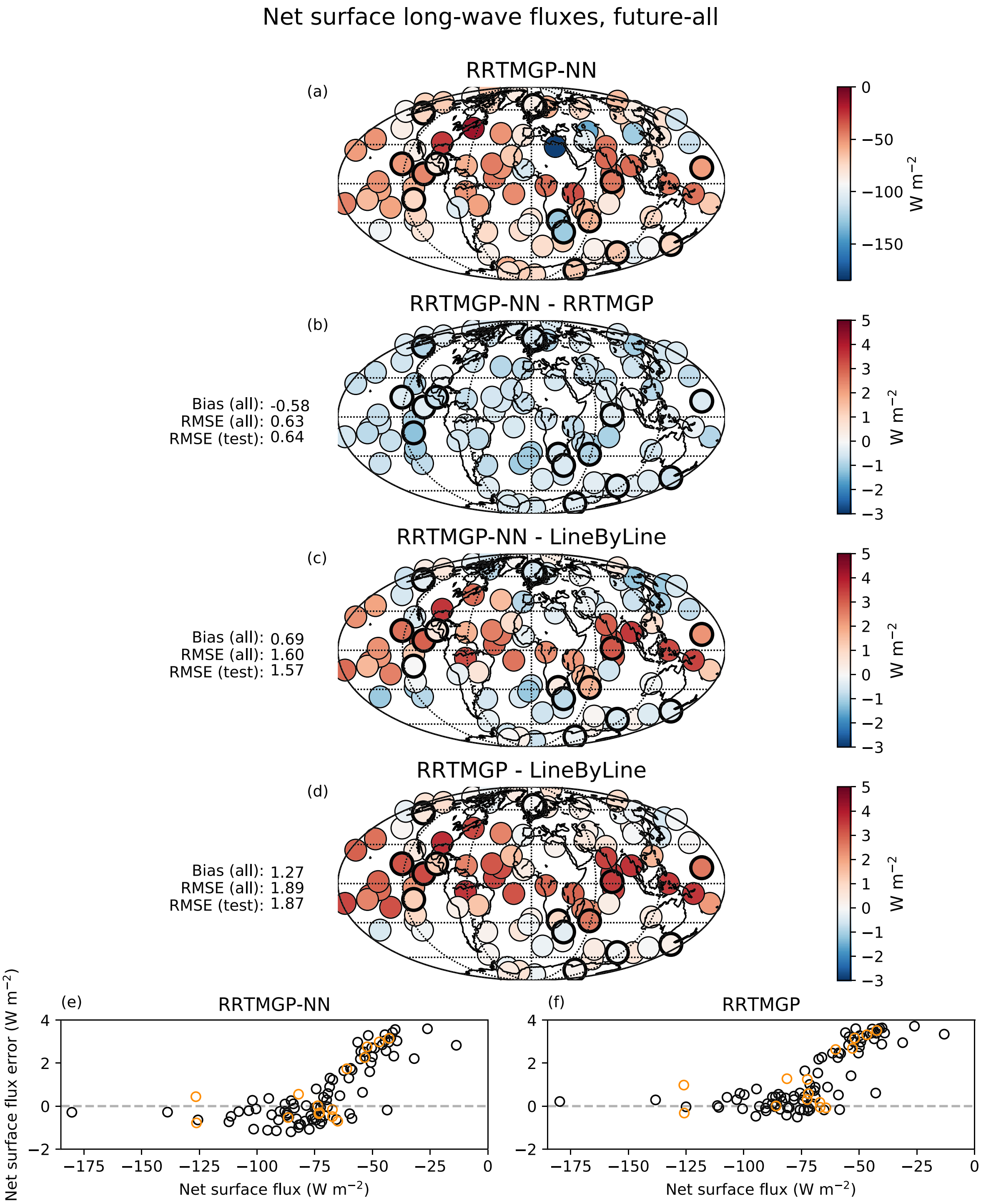


Figure 4. As in Figure 3 but for the net longwave fluxes at surface for the “future-all” experiment.

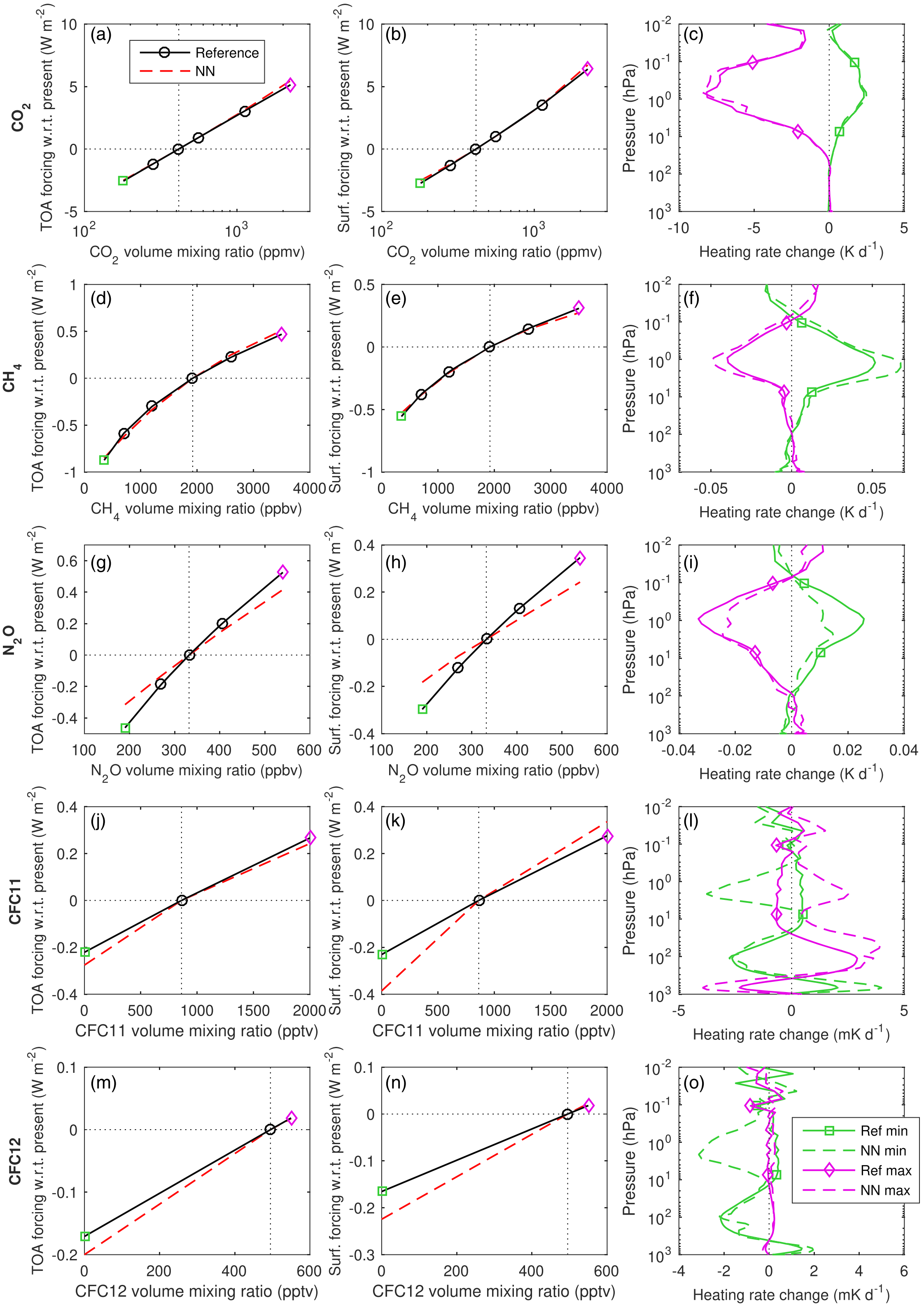


Figure 5. (a–l) Comparison of RRTMGP-NN and reference line-by-line computations of instantaneous radiative forcing at top of atmosphere and the surface when perturbing the concentrations of individual well-mixed greenhouse gases from their present-day values, averaged over 50 profiles in the Evaluation-1 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

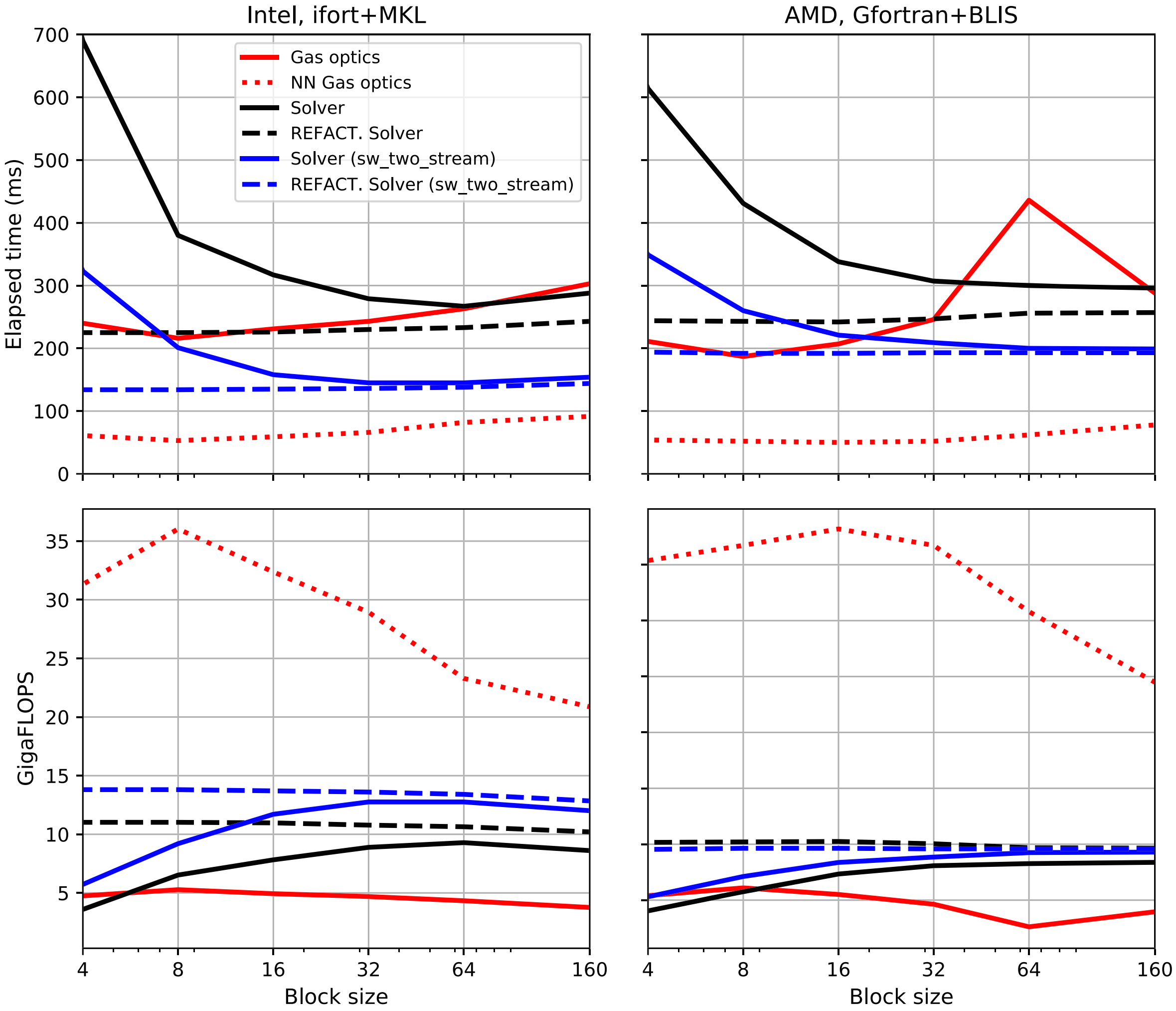


Figure A1. As in Figure 1, but showing the impact of block size on the elapsed time (top row) and floating-point operations per second (bottom row) of the shortwave codes for the Intel (left) and AMD platform (right). NN = neural network, REFACT = refactored.

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 runtime 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

3.2 Supplementary results

During the time after Paper 1 was published, RTE+RRTMGP has been under continuous development. New correlated-k distributions have been released with roughly half the number of g-points as the models used in Paper 1, reducing the computational expense of the entire code. In addition, the code has seen various optimizations to improve efficiency. Some of these may have arose from possible improvements described in Paper 1 (inlining of broadband flux computations) but the most significant is in the gas optics code, which has been rewritten to be consistent with the solver in its array structure (columns as the innermost dimension), therefore avoiding expensive array transposes.

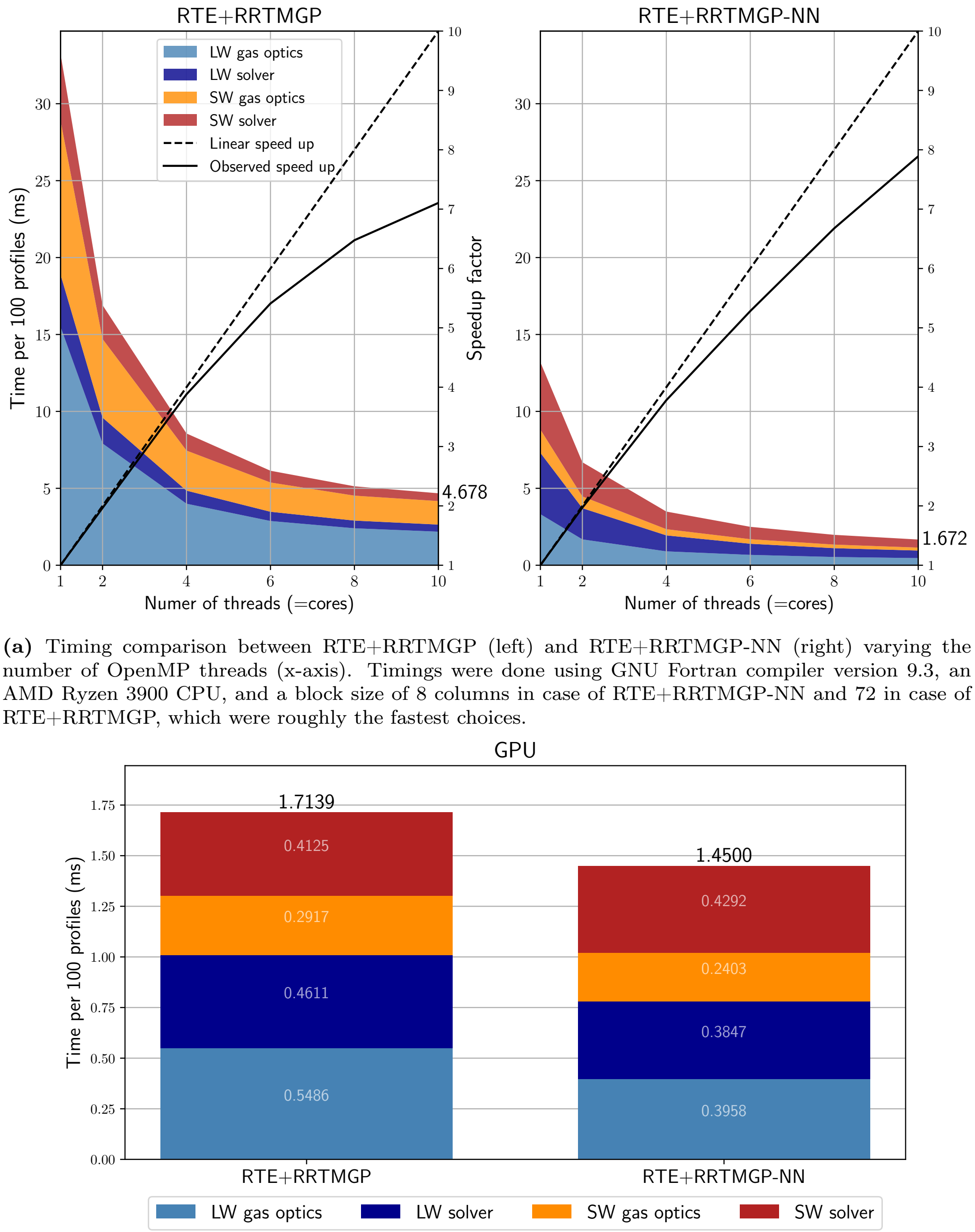
With these developments in mind it's of interest to compare the NN fork of the code to a recent version of the reference code, to see if the NN version of the radiation code (with columns outermost) is still faster.

On CPUs, RTE+RRTMGP-NN still has a similar speedup compared to RTE+RRTMGP as was reported in Paper 1, with the former code being 2.5x - 2.7x faster in total. This is despite the refactored gas optics in the reference code being significantly faster than before. Not shown here is the RTE+RRTMGP-NN code using look-up-tables instead of NNs; this was around 25% faster than the reference RRTMGP gas optics (on CPUs), with most of this the difference probably attributable to RTE+RRTMGP-NN using one fewer Planck source functions (one for levels or "layers", and one for half-levels, instead of two for half-levels) than the reference code. The computation of spectral Planck sources in RTE+RRTMGP-NN is now done fully within the gas optics, and not in the solver, which previously broke the separation of concerns as discussed in the paper.

On GPU's, however, the NNs are now only slightly faster than the reference code. In general, the performance of reference RTE+RRTMGP is very good, probably due to the recent refactoring to avoid transposes and continued collaboration with NVIDIA to optimize the code for GPUs. Still, the fact that the LUT gas optics code is now almost as fast as the NN code is somewhat surprising considering the widespread notion of GPU's exceling at machine learning computations, and thae fact that these timings include all RRTMGP gases, which include many minor species in the longwave, and in the reference code the contributions from minor gases are computed separately.

One factor here is the use of the smaller *k*-distributions combined with relatively complex NN models. For the reference code, the halving (roughly) of the number of *g*-points results to a corresponding decrease in the number of floating point operations, and the LW gas optics is roughly 75% faster compared to the old *k*-distribution (on GPU). With RTE+RRTMGP-NN, the new gas optics models are only $\approx$ 45% faster. These models were kept relatively complex in order to more easily produce accurate surface and TOA forcings with respect to minor greenhouse gases (Paper 3). It would be possible to predict heating rates accurately using less complex models. Finally, the Fortran inference code is not optimal on GPUs: despite using the efficient CUDA BLAS library for matrix-matrix computations, timings done in Paper 3 revealed that inference using an external runtime library (ONNX) was much faster than the Fortran NN code.

One optimization not featured in the current GPU implementation of RTE+RRTMGP is inlining the bias addition, the last step in every NN layer, with the matrix-matrix multiplication. (Such operations are included in the cuBLASLt library, for which Fortran bindings are unfortunately not available).



(a) Timing comparison between RTE+RRTMGP (left) and RTE+RRTMGP-NN (right) varying the number of OpenMP threads (x-axis). Timings were done using GNU Fortran compiler version 9.3, an AMD Ryzen 3900 CPU, and a block size of 8 columns in case of RTE+RRTMGP-NN and 72 in case of RTE+RRTMGP, which were roughly the fastest choices.

(b) Same as above, but using a GPU (NVIDIA GTX 1070), NVIDIA HPC SDK compiler version 21.5, and cuBLAS 11.3. Block size was set to the problem size (7200 columns) which gives the fastest results in this case.

Figure 3.1: Time to solution for computing fluxes for 7200 clear-sky profiles with 60 model layers using recent k-distributions with fewer g-points. Left subfigures are the timings for the reference RTE+RRTMGP code version 1.5 and right subfigures are using RTE+RRTMGP-NN, with the NN models developed in Paper 3. Timings do not include IO (such as loading netCDF files), but represent the total time spent in the gas optics and solvers.

References

Chevallier, F., Chéruy, F., Scott, N., and Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. *Journal of applied meteorology*, 37(11):1385–1397.

Hogan, R. (2018). Radiation in the next generation of weather forecast models: Workshop report. Technical report.

4

Paper 2: Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer

4.1 Motivation

The following paper was borne out of an idea to test some of the implicit assumptions made in Paper 1, that emulating an entire radiation scheme using a NN may be too difficult to do with the level of accuracy, interpretability and energy conservation that is required to use such models in operational weather forecasts and climate simulations. It was speculated that only emulating the gas optics scheme is a much more accurate approach. In this paper, this is made into a testable hypothesis by training NNs to emulate both the full radiation scheme and its components, and by comparing the tradeoff in accuracy and speed. Its more novel contribution is developing a recurrent NN method to emulate a radiation scheme, where previous work have only used feed-forward or convolutional networks for this problem.

JAMES | Journal of Advances in Modeling Earth Systems®

RESEARCH ARTICLE

10.1029/2021MS002875

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

Peter Ukkonen[1,2]

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

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

Correspondence to:

P. Ukkonen,
peter.ukkonen@nbi.ku.dk

Citation:

Ukkonen, P. (2022). Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer. *Journal of Advances in Modeling Earth Systems*, *14*, e2021MS002875. https://doi.org/10.1029/2021MS002875

Received 20 OCT 2021
Accepted 28 FEB 2022

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 larger 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 high 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 the 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 of 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 have 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 (Brenowitz & Bretherton, 2018; Gentine et al., 2018; Rasp 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 (Gentine et al., 2018; Rasp 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 is the radiation scheme, accounting for 50% of the runtime in one global climate model (Cotronei & Slawig, 2020). 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 $\mathrm{Wm^{-2}}$ 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 $\mathrm{day^{-1}}$ in heating rates and 20 $\mathrm{Wm^{-2}}$ in fluxes.

Although these differences seem large compared to parameterization errors for clear-sky radiation (Hogan & Matricardi, 2020, Figures 5 and 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 model's 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 O'Gorman (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 algorithmically 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

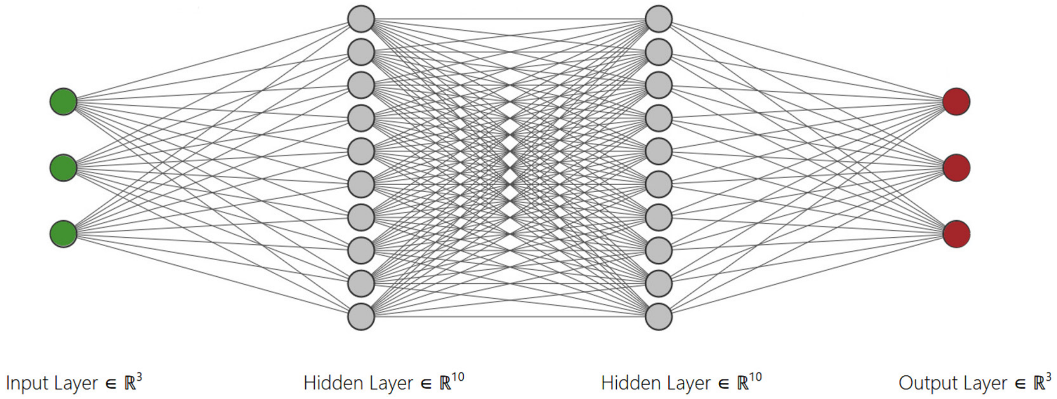

Input Layer ∈ $\mathbb{R}^3$
Hidden Layer ∈ $\mathbb{R}^{10}$
Hidden Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^3$

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 feed-forward neural network 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).

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.

The aim of this study is 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, that is, 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, neural networks are trained 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 the data and codes are introduced (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

Data from the global Copernicus Atmospheric Monitoring Service (CAMS) reanalysis (Inness et al., 2019) was acquired for 2009–2018, saving 4 days for each year (1.2, 1.5, 1.8 and 1.11), and 2 times for each day (03 and 15 UTC) 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-m temperature, and forecast albedo. True solar zenith angles were also computed 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 original vertical grid of 60 layers (top at 10 Pa). Each year consists of 5,120 × 8 = 40,960 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) subsets. Although having testing data from the middle of the period may not represent a realistic use case, the data was interleaved in this manner to avoid greenhouse gas concentrations in the evaluation that are higher than those in the training data. Testing the ability of NNs to extrapolate beyond the training data may be relevant for for example, climate modeling, but was not the aim here. Results from other studies suggest that NNs may struggle to extrapolate but can interpolate in between extremes (O'Gorman & Dwyer, 2018; Rasp et al., 2018). 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 40,960 × 8 = 327,680 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, random samples were extracted from the data, limiting the number of training samples to roughly 33 million (reflectance-transmittance) or 2 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 applications — Parallel (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 favor 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).

Table 1
Description of the Different Models

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	$2 + 9\ nlay = 542$	$2 + 9 = 11$	5	7
Output	vertical profiles of broadband fluxes $F_\uparrow$, $F_\downarrow$	vertical profiles of broadband fluxes $F_\uparrow$, $F_\downarrow$	absorption/Rayleigh cross-sections as a vector of *g*-points → τ, *ssa*	R_{dif}, T_{dif}, R_{dir}, T_{dir}
Output size	$2\ nlev = 122$	2	$ngpt = 224$	4
Required iterations	*ncol*	*ncol* (×3 *nlay* internallly)	$ncol \times nlay$ (×2 NN models)	$ncol \times nlay \times ngpt$
Hidden layers	Dense, Dense, Dense	RNN, Dense, RNN, RNN	Dense, Dense	Dense, Dense
Activation functions in hidden and output layers	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
Flexible with regards to vertical grid	No	Yes	Yes	Yes
Input scaling	$x = \log(x)$ for p; $x = x^{\frac{1}{4}}$ for H_2O and O_3; $x_i = \frac{x_i}{\max(x_i)}$	$x = \log(x)$ for p; $x = x^{\frac{1}{4}}$ for H_2O and O_3; $x_i = \frac{x_i}{\max(x_i)}$	$x = \log(x)$ 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	$y_i = \frac{y_i}{F_{\downarrow,0}}$	$y_i = \frac{y_i}{F_{\downarrow,0}}$	$y = y^{\frac{1}{8}}$; $y_i = \frac{y_i - \bar{y}_i}{\sigma_y}$	$y = y^{\frac{1}{4}}$

Abbreviations: *ncol* = Number of Columns, *nlay* = Number of Atmospheric Layers, $nlev = nlay + 1$ = Number of Atmospheric Levels, *ngpt* = Number of *g*-points, α = Surface Albedo, μ_0 = Cosine of Solar Zenith Angle, T = Temperature, p = Pressure, τ = Optical Depth, *ssa* = Single-Scattering Albedo, *g* = Asymmetry Parameter, *T* = Transmittance, *R* = Reflectance, H_2O = Mixing Ratio of Water Vapor, O_3 = Mixing Ratio of Ozone. Activation Functions: tanh = Hyperbolic Tangent, Sigmoid = Logistic Function, ReLU = REctified Linear Unit ($\max(x, 0)$), Softsign = $\frac{x}{1+|x|}$, Hard Sigmoid = $\max(0, \min(1, 0.2x + 0.5))$

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) was 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 radiation computations which account for gases and clouds to generate the input-output pairs for machine learning. The computations account for scattering, and cloud optical properties were generated with a cloud optics extension in RTE + RRTMGP that is based on Mie calculations. Clouds were assumed to fill each grid box horizontally (fractional cloud cover was not considered). 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 given its popularity in the research community, 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 $F_{\downarrow}$, $F_{\downarrow}$ (total flux, given by diffuse plus direct flux) and direct shortwave fluxes $F_{\downarrow dir}$, $F_{\downarrow dir}$ for each g-point and model level (also known as half-level). Finally, broadband fluxes are obtained $F_{\downarrow}$, $F_{\downarrow}$ 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; Pal et al., 2019; Roh & Song, 2020) 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 it was chosen to predict fluxes, while HR is given by the vertical divergence of net fluxes at each model layer i as in physical radiation codes:

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

where $F_{\mathrm{i}+1/2,\ \mathrm{SW}}$ 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 and devising two techniques to improve emulation of SW radiative transfer.

First, 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, which is beneficial for training. In addition, incoming flux is no longer needed as an input and model generalization should improve. 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 was combined with a sigmoid activation in the output layer to constrain outputs within the 0–1 range, which was found to reduce errors.

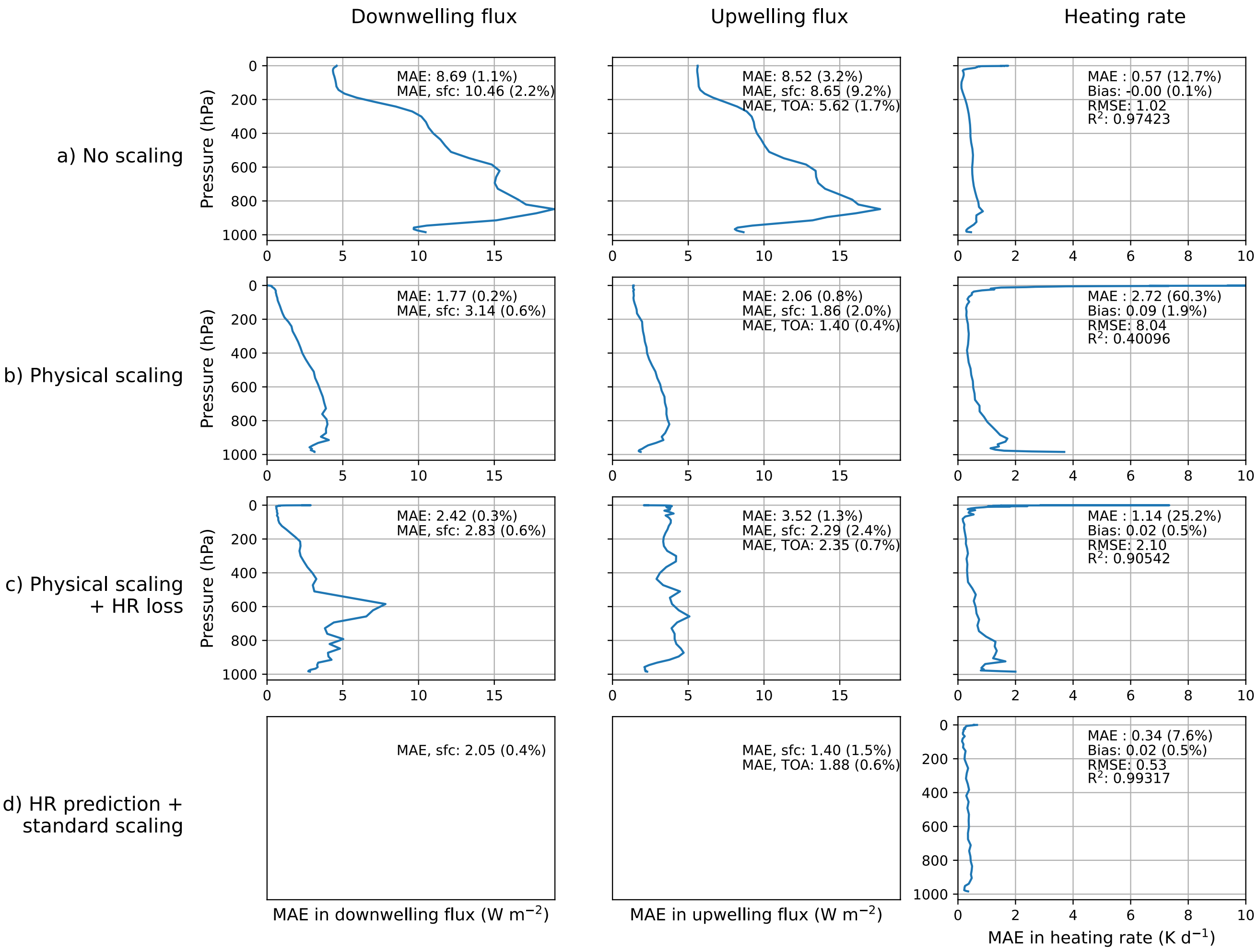


Figure 2. Impact of scaling, loss function and predictand on the vertical profiles of mean absolute error in downwelling flux (left column), upwelling flux (middle column) and heating rate (right column) for the validation data from 2014 with randomly sampled solar zenith angles. The outputs of the different feed-forward neural network 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}). The outputs were scaled by the incoming flux in (b), (c) and standardized in (d) to have a mean of zero and unit variance. 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 FNNs were trained to allow different random initializations of weights, and the results with smallest heating rate errors were saved and compared.

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, the method corresponding to Figure 2 (c) 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 hyperparameters of the FNN were tuned by hand, testing a few different activation functions and levels of complexity (64, 128, or 192 neurons in 1–3 hidden layers), and aiming to maximize validation accuracy as the primary goal and efficiency as secondary. To improve the latter, the same number of neurons are used in each hidden layer, and a simple RELU (see caption of Table 1) activation is used in all hidden layers but the last one. These choices did not seem to sacrifice accuracy. The FNNs have 128 neurons in three hidden layers. Two hidden layers was only slightly less accurate, but a shallow FNN 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 FNN 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 recurrent neural network (RNN; reviewed in Young et al., 2018), 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, that is, 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; Schuster & Paliwal, 1997). A BiRNN is comprised 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 level) as well as the upward flux at the top of the layer (the previous 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 ($nlev = nlay + 1$), 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

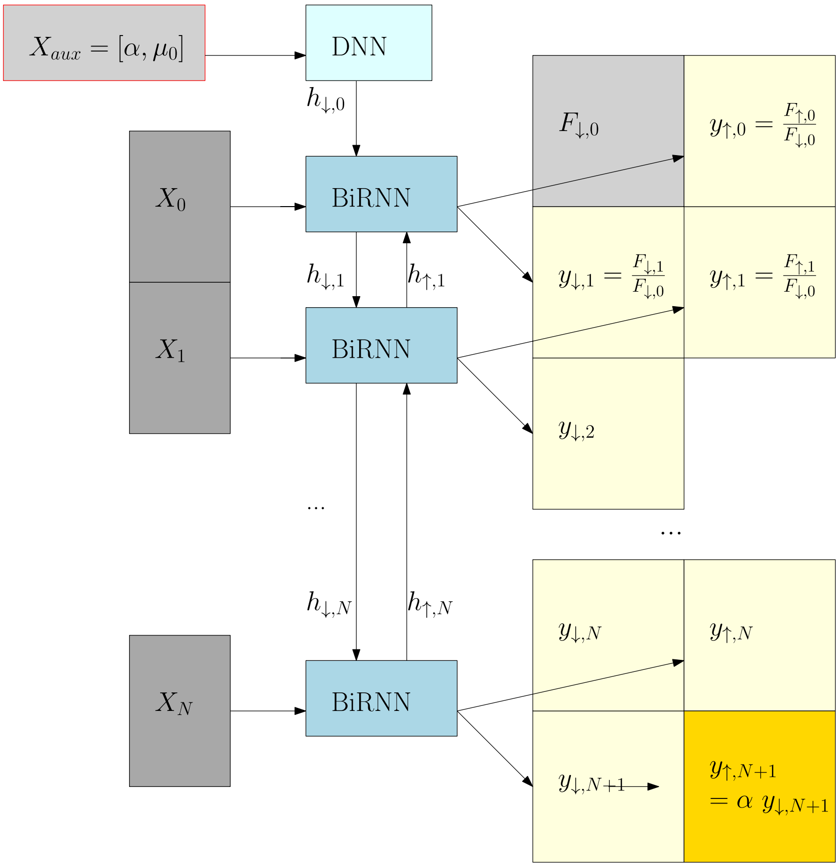

$X_{aux} = [\alpha, \mu_0]$
DNN
$h_{\downarrow,0}$
X_0
BiRNN
$h_{\downarrow,1}$
$h_{\uparrow,1}$
X_1
BiRNN
...
$h_{\downarrow,N}$
$h_{\uparrow,N}$
X_N
BiRNN
$F_{\downarrow,0}$
$y_{\uparrow,0} = \frac{F_{\uparrow,0}}{F_{\downarrow,0}}$
$y_{\downarrow,1} = \frac{F_{\downarrow,1}}{F_{\downarrow,0}}$
$y_{\uparrow,1} = \frac{F_{\uparrow,1}}{F_{\downarrow,0}}$
$y_{\downarrow,2}$
...
$y_{\downarrow,N}$
$y_{\uparrow,N}$
$y_{\downarrow,N+1}$
$y_{\uparrow,N+1}$
$= \alpha\ y_{\downarrow,N+1}$

Figure 3. A RNN-based approach to predicting radiative fluxes. Input variables defined at N model layers (X_0, $X_1 \ldots X_N$) form the sequential input to the bidirectional RNN (BiRNN), while the output consists 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 $h_\uparrow$, $h_\downarrow$ in the case of the BiRNN which internally consists 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 $h_{0,\uparrow}$, $h_{N,\uparrow}$. 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.

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 a model using only 16 neurons in each of the three RNN layers is evaluated. Gated Recurrent Units (GRU), which are more complex than simple RNN layers, were used in each RNN layer. 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.

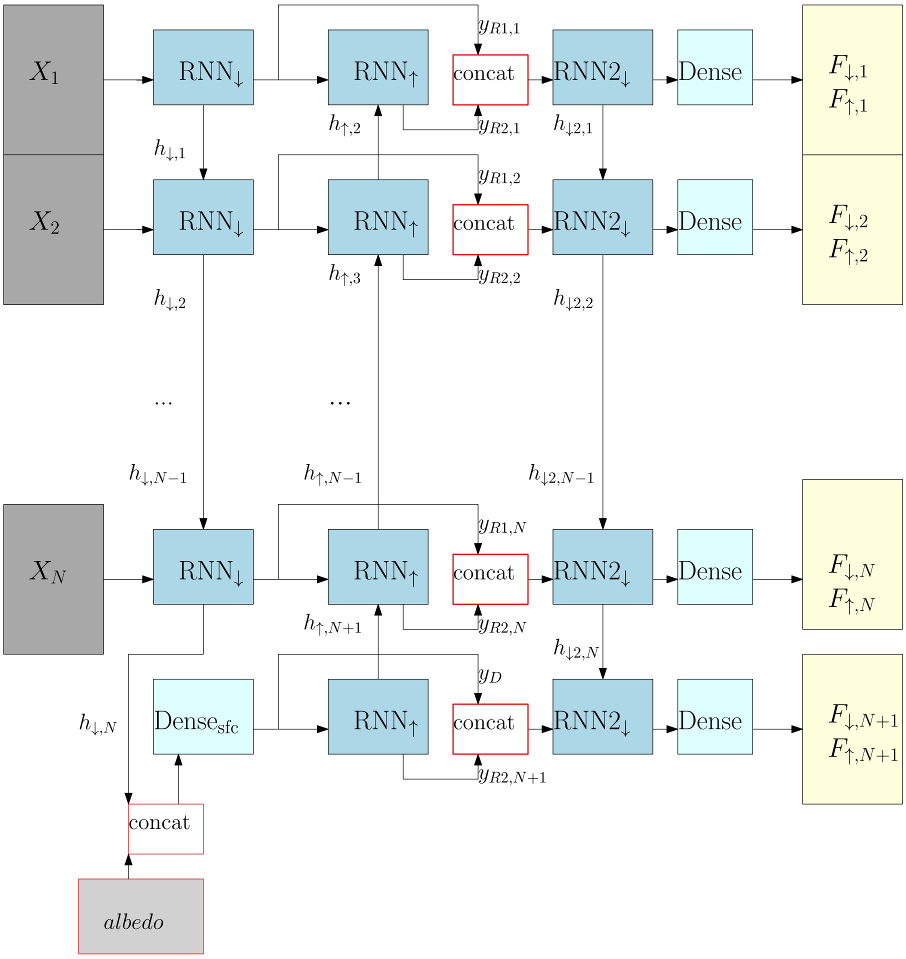


Figure 4. A recurrent NN (RNN)-based approach to predicting fluxes at layer interfaces ($N + 1$) from layer-wise inputs (N), consisting of three RNNs to mimic two-stream radiative transfer equations with scattering. The first RNN ($RNN_{\downarrow}$) has a forward (downward) direction, and when it reaches the end of the sequence, that is, the last vertical layer at N, its hidden state $h_{\downarrow N}$ is concatenated ("concat") with the surface albedo(s) 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

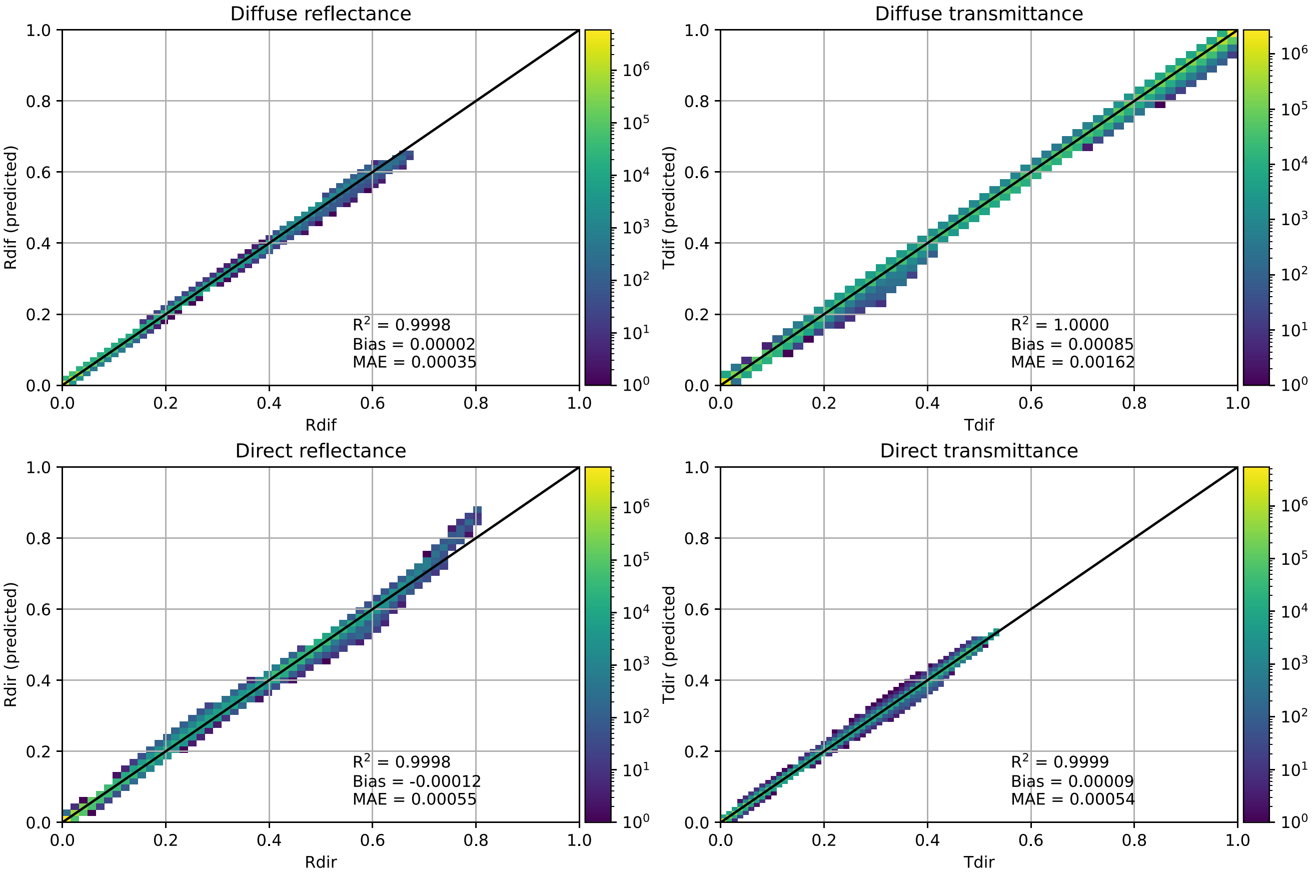


Figure 5. Comparison of the predicted (*y*-axis) and true (*x*-axis) reflectance and transmittance values using the validation data set and final REFTRANS model, which has 12 neurons in two hidden layers. These errors are for the immediate NN output, not implemented inside the radiation code. The colors on the scatter plot correspond to the occurrence on a log-scale.

layer-wise optical properties normalized by the path number of molecules N (cross-sections, from which optical depth τ is then computed by multiplying with N), 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 underrepresented 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). Custom loss functions were then tested, which give smaller weights to intermediate values of all four outputs, and/or a smaller weight to diffuse transmittance, but no clear improvement in the predicted fluxes were found. 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 reader is advised to refer to this table to keep track of the four different emulation methods. The model hyperparameters (number of hidden neurons, hidden layers and activation functions) as well as suitable pre-processing methods 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. Pre-processing was found to be at least as important as NN hyperparameters. Input variables spanning many orders of magnitude were first log-scaled or power-scaled for a more linear distribution, and then all inputs were scaled into a similar numerical range (Ukkonen et al., 2020).

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.

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 and independent testing data is shown in Figure 6. In this offline evaluation based on one year of global data which was not used for model training or tuning, 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 data set. 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

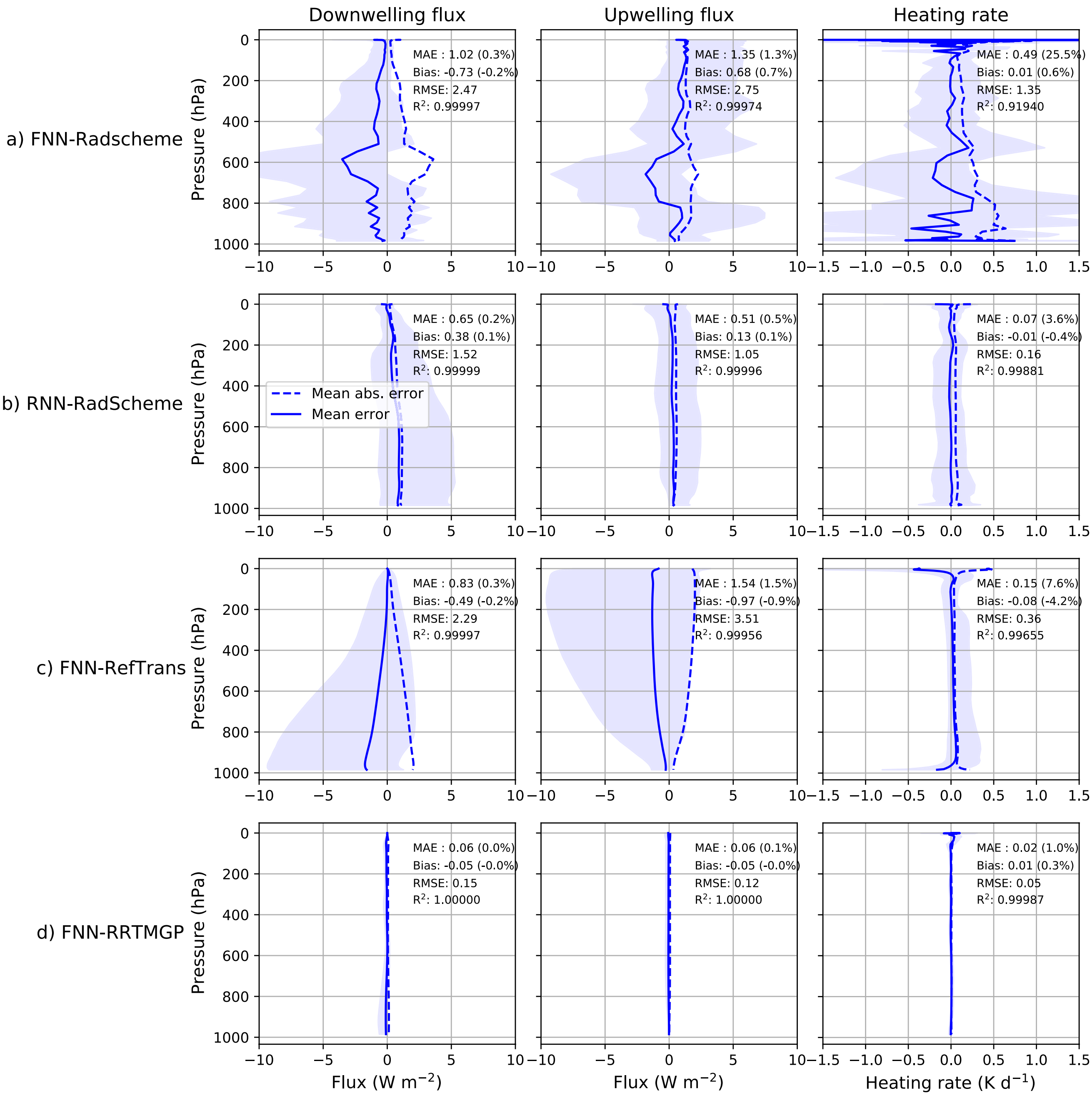


Figure 6. Vertical profiles of the error in shortwave downwelling flux (left column), upwelling flux (middle column) and heating rates (right column) for the test data (2015) using different emulation methods: replacing the radiation scheme with (a) a feed-forward neural network (NN) or (b) a bidirectional 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 differences (predicted flux - true flux) at each level. 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.

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 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).

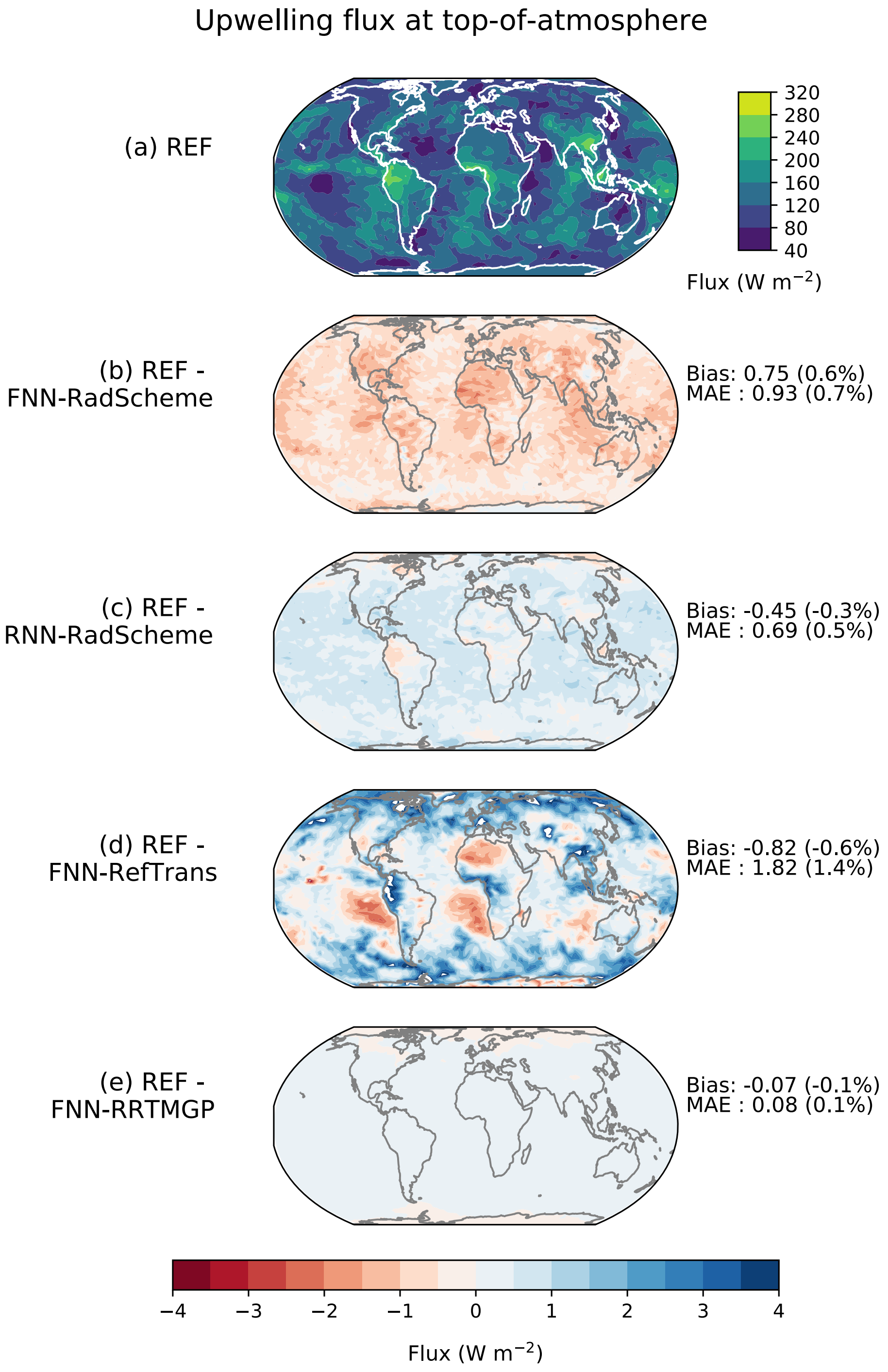


Figure 7. Global upwelling shortwave flux at top-of-atmosphere for the testing year 2015 as computed with RTE + RRTMGP (a) and the grid box mean differences in this quantity using different emulators (b)–(e). Bulk error statistics with respect to individual columns are displayed on the right hand side.

4.2. Speed-Up

Speedup of the radiation codes was 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.

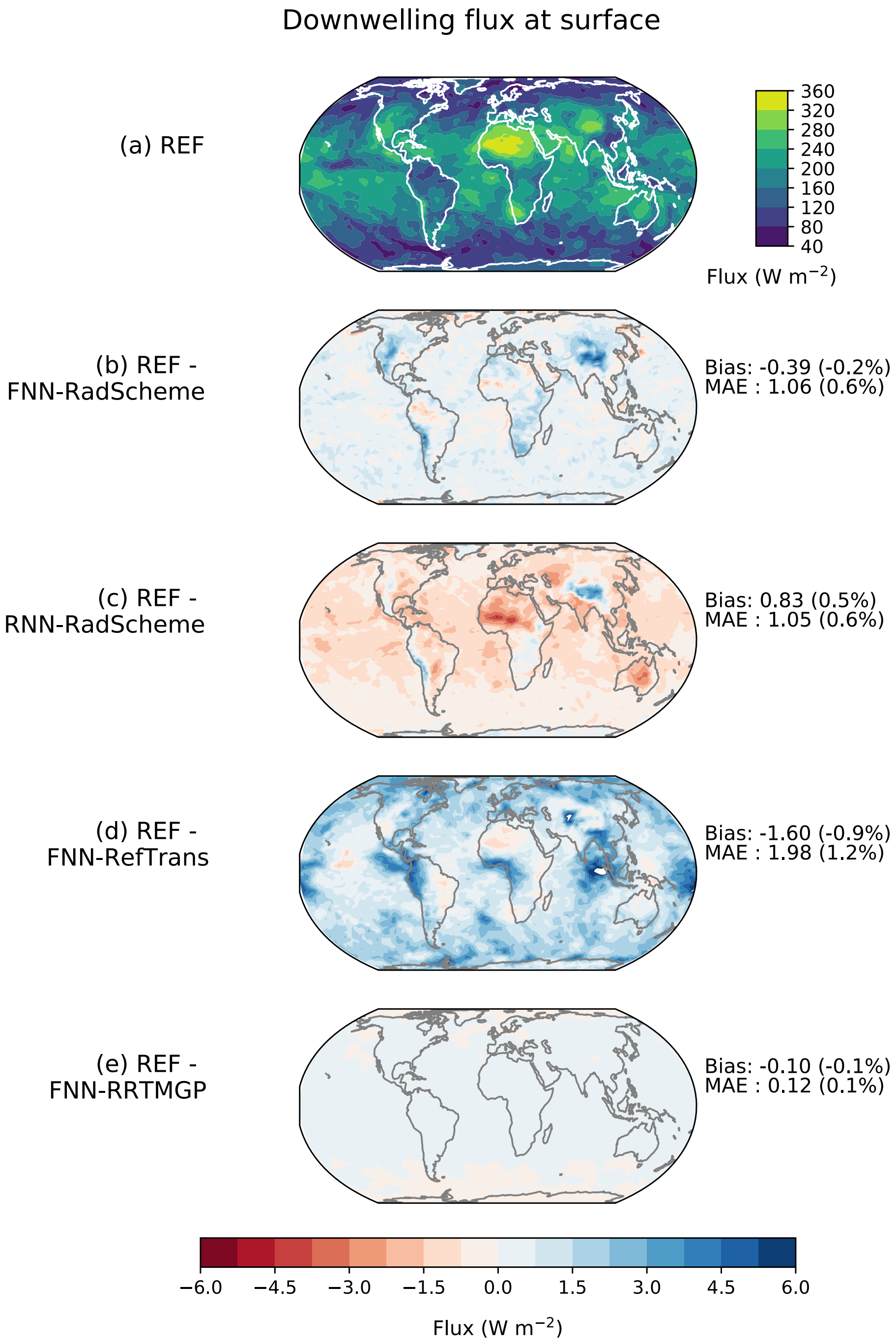


Figure 8. As in Figure 7, but for downwelling flux at surface.

Principal timings were done using the AMD Ryzen 7 5800H processor and GNU compiler version 11 (compiler options *-march = native -O$_3$*). 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 81,920 test columns took roughly 18.5 s using the reference code and a single core on the CPU (Central Processing Unit). By comparison, the FNN-RadScheme computed fluxes in just 0.35 s, 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 were 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 s 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 s. 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 RTX 3060 Mobile GPU (Graphics Processing Unit) was then briefly evaluted using ORT. The RNN inference time is reduced to 0.34 s on the GPU, while the FNN inference took a mere 0.046 s. 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 large heating rate errors and noisy training losses with flux-predicting FNN may be caused by the fact that the NN outputs at different atmospheric levels are not structurally correlated with outputs at adjacent levels, and that heating rate is given by the vertical divergence in flux. The RNN, which does 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 and 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 speed-up of 54 times was obtained relative to running the original code on a single CPU core. (Since modern CPUs have many cores, the effective speed-up will be considerably lower than this. The comparison is also hindered by the use of commodity hardware).

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. While it is very difficult to know how the emulators would perform as an online parameterization based on offline metrics, it may be useful to compare the errors obtained here to studies were both offline and online errors were evaluated. These include the ones mentioned above with similar or 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 of 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, but a model trained on one vertical grid is not guaranteed or even likely to perform well when applied to another, due to optical properties being vertically integrated quantities. (Training a single model on different data sets with varying resolutions may be possible, but was not investigated here.) 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.

Data Availability Statement

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.

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 thoroughly reviewing the manuscript. The author also thanks the two anonymous reviewers, whose feedback helped to improve the paper.

References

Aldakheel, F., Satari, R., & Wriggers, P. (2021). Feed-forward neural networks for failure mechanics problems. *Applied Sciences*, *11*(14), 6483. https://doi.org/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. https://doi.org/10.1103/physrevlett.126.098302

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. https://doi.org/10.1029/2018gl078510

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

Fuhrer, O., Chadha, T., Hoefler, T., Kwasniewski, G., Lapillonne, X., Leutwyler, D., et al. (2018). Near-global climate simulation at 1 km resolution: Establishing a performance baseline on 4888 GPUs with COSMO 5.0. *Geoscientific Model Development*, *11*(4), 1665–1681. https://doi.org/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. 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. https://doi.org/10.1016/0022-4073(89)90044-7

Hansen, J. E. (1971). Multiple scattering of polarized light in planetary atmospheres part I. The doubling method. *Journal of the Atmospheric Sciences*, *28*(1), 120–125. https://doi.org/10.1175/1520-0469(1971)028<0120:msopli>2.0.co;2

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. https://doi.org/10.1175/2010jas3202.1

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. https://doi.org/10.5194/gmd-13-6501-2020

Inness, A., Ades, M., Agusti-Panareda, A., Barré, J., Benedictow, A., & Blechschmidt, A.-M. (2019). othersThe 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. https://doi.org/10.1364/ao.44.002613

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*, *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. https://doi.org/10.1175/2009mwr3149.1

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 the Atmospheric Sciences*, *37*(3), 630–643. https://doi.org/10.1175/1520-0469(1980)037<0630:tsatrt>2.0.co;2

Odyssee-Mure (2021). *Sectoral profile households - electricity consumption per dwelling*. Retrieved from https://www.odyssee-mure.eu/publications/efficiency-by-sector/households/electricity-consumption-dwelling.html

O'Gorman, P. A., & Dwyer, J. G. (2018). Using machine learning to parameterize moist convection: Potential for modeling of climate, climate change, and extreme events. *Journal of Advances in Modeling Earth Systems*, *10*(10), 2548–2563. https://doi.org/10.1029/2018MS001351

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. https://doi.org/10.1029/2018gl081646

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

Räisänen, P., Barker, H. W., & Cole, J. (2005). The Monte Carlo independent column approximation's conditional random noise: Impact on simulated climate. *Journal of Climate*, *18*(22), 4715–4730. https://doi.org/10.1175/jcli3556.1

Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent subgrid processes in climate models. *115*(39), 9684–9689. https://doi.org/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. https://doi.org/10.1029/2020gl089444

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Song, H.-J., & Roh, S. (2021). Improved weather forecasting using neural network emulation for radiation parameterization. *Journal of Advances in Modeling Earth Systems*, e2021MS002609. https://doi.org/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. https://doi.org/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. https://doi.org/10.1098/rsta.2020.0095

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/mci.2018.2840738

Yuval, J., & O'Gorman, P. A. (2020). Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, *11*(1), 1–10. https://doi.org/10.1038/s41467-020-17142-3

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. https://doi.org/10.1029/2020gl091363

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. https://doi.org/10.1002/qj.2378

5

Paper 3: Implementation of machine-learned gas optics parameterization in the ECMWF Integrated Forecasting System)

Implementation of a machine-learned gas optics parameterization in the ECMWF Integrated Forecasting System

Peter Ukkonen[1], **Robin J. Hogan**[2]

[1]Danish Meteorological Institute
[2]European Centre for Medium-Range Weather Forecasts

Key Points:

- A machine-learned gas optics scheme, RRTMGP-NN, was implemented in a global weather model
- Compared to the original RRTMGP gas optics, RRTMGP-NN speeds up the ecRAD radiation scheme by roughly 30%
- Using RRTMGP-NN instead of RRTMGP does not seem to have a significant impact on model climate

Corresponding author: Peter Ukkonen, `puk@dmi.dk`

Abstract

Radiative transfer parameterizations are physically important but computationally expensive components of weather and climate models. In previous work, it was demonstrated that the gas optics module of a radiation scheme, which traditionally rely on look-up-tables, can be replaced with neural networks (NN) to improve speed while retaining a high degree of accuracy. However, the evaluation of the NN version of the RRTMGP gas optics scheme (RRTMGP-NN) was based only on offline radiation computations.

In this paper, we describe the implementation and prognostic evaluation of RRTMGP-NN in the Integrated Forecasting System (IFS) of the European Centre for Medium-Range Weather Forecasts (ECMWF). This was carried out by incorporating the gas optics scheme in ecRAD, the modular radiation scheme used in the IFS. Year-long coupled ocean-atmosphere simulations show that the impact on model climate from using RRTMGP-NN is small compared to the differences between existing gas optics schemes. The use of RRTMGP-NN speeds up the radiation scheme by roughly a third compared to RRTMGP, and is also faster than the older and less accurate RRTMG which is used in the current operational cycle of the IFS.

1 Introduction

Although atmospheric radiation is well understood and very accurate solutions are available, atmospheric models need to settle for a trade-off in the accuracy and cost of radiation computations. This trade-off can be controlled via many factors like the temporal and spatial frequency of computations, simplifying assumptions for radiative transfer problem (e.g. neglecting 3D effects), and spectral resolution. Most modern radiation schemes use the correlated-k-distribution method which allows computing broadband fluxes with high accuracy using only O(2-3) quadrature points, compared with O(6) for line-by-line methods.

Despite this, computations remain expensive enough that many other of the aforementioned approximations need to be made, and still large-scale climate simulations especially may spend a large share of the total model runtime on radiation computations (Cotronei & Slawig, 2020). To make better use of computer resources in an era where computer hardware is becoming more heterogenous, and the gap between the theoretical peak performance and the performance of typical physics codes is probably increasing, the use of machine learning (ML) for physics parameterizations is promising. Indeed, interest in the use of ML for radiative transfer in NWP and climate models has been growing but has a long history as a research topic (Chevallier et al., 1998; V. M. Krasnopolsky et al., 2008; V. Krasnopolsky et al., 2010; Pal et al., 2019; Liu et al., 2020; Roh & Song, 2020; Song & Roh, 2021). These studies have attempted to replace the entire radiation scheme with a feed-forward neural network (FNN). An alternative approach, which offers better accuracy at the cost of a smaller speed-up, is to keep the radiative transfer equations but replace the computation of gas optical properties with NNs. Since gas optics rely on look-up-tables and empiricism, it's a very suitable problem for ML. FNNs were developed to emulate the RRTMGP gas optics scheme (Pincus et al., 2019) in two different studies, which found speed-ups of 2-6x compared to the original code (Ukkonen et al., 2020; Veerman et al., 2021). The NN gas optics was combined with a refactored radiative transfer solver to speed up the entire radiation scheme (without clouds or aerosols) by a factor of 1.8 - 3.5 in Ukkonen et al. (2020). Recently, Ukkonen (2021) compared different emulation strategies for shortwave radiation, and found that using NNs for gas optics did not sacrifice almost any accuracy, whereas replacing the entire scheme with FNNs was the fastest but also least accurate approach, with heating rates (computed from predicted broadband fluxes) having a root-mean-square-error (RMSE) of 1.35 K day^{-}1. An interesting alternative for emulating the full radiation scheme was found in recurrent NNs, which produced far more accurate fluxes and heating rates (RMSE 0.16 K day^{-}1) than FNNs while offering a smaller but still significant speedup.

While these results indicate that gas optics emulation is more accurate and more ready for operational implementation than emulating the entire radiation code, for instance due to inherently better generalization (e.g. to various vertical grids), the evaluations were based on offline radiation computations (Ukkonen et al., 2020; Veerman et al., 2021; Ukkonen, 2021). In this study, the NN version of the RRTMGP gas optics scheme (Ukkonen et al., 2020) is implemented in the ecRAD radiation scheme used in the Integrated Forecasting System (IFS), which is a global numerical weather prediction model developed at the European Centre for Medium-Range Weather Forecasts (ECMWF). New NN models are trained on RRTMGP k-distributions that recently became available, which have around the same number of k-terms as the older RRTMG scheme that is used operationally in the IFS.

The structure of the paper is as follows: Section 2 briefly describes the ecRAD and RRTMGP-NN codes, and the implementation of RRTMGP-NN in ecRAD. Section 3 provides an overview of the machine learning methodology, which has been refined to capture radiative forcings with respect to individual gases more accurately. The results are then presented in Section 4, consisting of an offline evaluation, and a prognostic evaluation to evaluate the impact of the new gas optics schemes (RRTMGP and RRTMGP-NN) on model climate using ”climate runs” with the IFS.

2 Codes

2.1 RRTMGP-NN and implementation in ecRAD

RRTMGP-NN previously loaded models from ASCII files like the Neural-Fortran code it is based on. We have refined the code so that models are loaded from netCDF files, which contain not only the weights and activation functions, but also input and output scaling coefficients, as well as metadata about the training data. These files could in the future be expanded to replace the k-distribution files in their entirety, keeping relevant metadata and the look-up-table coefficients used to compute Planck sources from Planck fraction and temperature.

We now briefly describe the integration of RRTMGP into ecRad. The goal was to avoid larger changes in ecRad. However, since (RTE+)RTTMGP makes heavy use of Fortran derived types to specify e.g. gas concentrations and optical properties, use of existing RRTMGP interfaces would imply a significant amount of array copying to communicate between ecRad and RRTMGP derived types. Larger changes in RRTMGP are not desirable either, because they reduce maintainability of RRTMGP itself, which continues to evolve.

With these conflicting goals in mind, a balance was sought with non-intrusive changes in both codes, but prioritizing minimal changes in ecRad. Firstly, the refactored radiation scheme with neural networks, RTE+RRTMGP-NN, was implemented instead of the reference gas optics code to make direct use of existing NN code. This has the advantage that RTE+RRTMGP-NN uses the same dimension order as ecRad with optical properties having g-points innermost and columns as the outermost dimension, removing the need for expensive array transposes (Ukkonen et al., 2020). While the NN fork of RTE+RRTMGP is currently only maintained by one person, the code is very similar to RTE+RRTMGP. The underlying k-distributions are loaded from netCDF files which can be copied over as new ones are made available in the main repository.

The entirety of the RTE+RRTMGP-NN package was then added as an ecRad subdirectory (this was necessary because RRTMGP and RTE are intertwined). The source code of RTE+RRTMGP(-NN) is kept separate: it does not use any of the ecRad modules. Instead, new interfaces were written for RTE+RRTMGP-NN for easy interoperability with ecRad while avoiding having to copies over larger arrays. For instance, the new interface for the longwave (*gas_optics_int_ecRad*) replaces the derived type arguments containing optical properties and Planck sources with explicit shape arrays (used in ecRad). The same RTTGMP(-NN) kernels can then called as they do not use derived types. In ecRad, another interface is then used which prepares the RRTMGP-NN gas concentrations (columns outermost) by transposing the ecRad gases (columns innermost) and calls *gas_optics_int_ecRad* (longwave) and *gas_optics_ext_ecRad* (shortwave). The overhead from transposing the gases and thermodynamic arrays is not significant. ecRad has corresponding interfaces for RRTMG and ECCKD gas optics.

3 Machine learning

In training NNs to emulate RRTMGP, we use a similar methodology as in Ukkonen et al. (2020), where detailed offline evaluation against line-by-line computations suggested a similar level of accuracy in overall fluxes and heating rates as the original scheme, despite using fairly simple NN models with two hidden layers and 16-48 neurons in each hidden layer. The choice of outputs, loss function, model optimization, and NN complexity are changed slightly as described in the next sections.

3.1 Data

We use similar training data as in Ukkonen et al. (2020), in which a diverse and extensive data set was prepared from several sources, including atmospheric profiles used in previous radiation studies, as well as data from future climate experiments and a reanalysis. These initial data sets were synthetically supplemented, or extended, by varying greenhouse gas concentrations both manually and by using Hypercube sampling. The data in this study differs from Ukkonen et al. (2020) in that: 1) data provided by the Radiative Forcing Model Intercomparison Project (RFMIP, Pincus et al., 2016), comprising of 100 profiles and 18 perturbation experiments now serves as an independent validation dataset used for early-stopping (section 3.3) instead of training, and 2) a different CAMS reanalysis data set is used. The new CAMS data uses the same approach as in ECMWF and the Correlated k-distribution Model Intercomparison Project (CKDMIP, Hogan & Matricardi, 2020), where only nine gases are considered, but the radiative forcing of many minor greenhouse gases is represented by artificially increasing the concentration of CFC-11. The height dependence of these gases is represented, and other RRTMGP gases are set to zero. (Neither of these generally applies to the data from other sources, where all minor RRTMGP gases are included, but as scalar concentrations).

The reanalysis profiles are designed to encompass the variability in present-day atmospheric conditions, with the following steps taken to increase variance and capture extremes. Starting from an initial pool of roughly 164 000 profiles spanning global reanalysis data from 2008 and 2017 and interpolated to a 320 km resolution equal-area grid Ukkonen (2021), 1000 profiles were drawn. Of these, 17 were selected to contain the minimum and maximum of temperature, humidity and ozone at different pressure levels (a total of 9 variables) in the whole dataset, similarly to Hogan and Matricardi (2020). Another 486 profiles were selected by constructing $k = 81$ k-means clusters which are clustered in the 9 dimensions represented by the variables in the previous step. From each cluster, which the k-means algorithm ensures are as different to other clusters as possible, 6 random profiles were selected. The remaining roughly 500 profiles were randomly drawn from the entire dataset minus ones already chosen. Vertical profiles selected by the minimum-maximum, semi-random and random method are depicted in Fig. 1.

The 1000 CAMS profiles were then expanded into 42 experiments or scenarios where CH_4, N_2O, $CFC_{11}-$eq and CFC_{12} are varied similarly to Hogan and Matricardi (2020). The 1000 × 42 × 60 (layers) ≈ 2.5 million samples make up roughly 47% of the 5.42 mil-

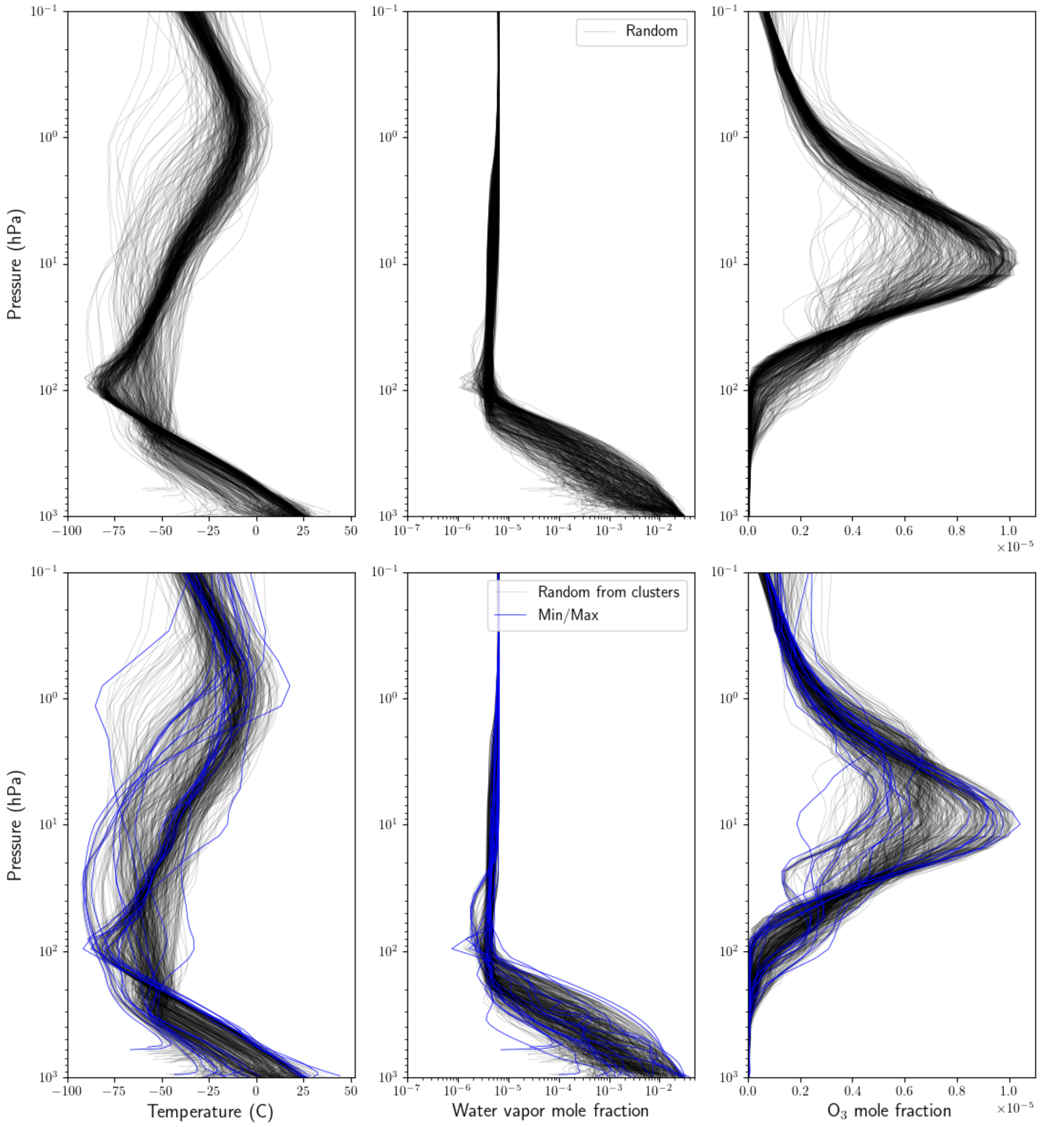


Figure 1: Vertical profiles of temperature, water vapor and ozone selected from the CAMS data as described in Sect. 3.1. The top panel shows 486 random profiles (black), and the bottom panel shows 486 profiles drawn from k-means clusters (black) and 17 that were selected to sample minimum and maximum values (blue).

lion training samples in total. The remaining part comprises of 1) CMIP6 data corresponding to a high-emissions scenario experiment, 2) profiles from CKDMIP, and 3) 42 profiles used for tuning RRTMGP; all of which were expanded into up to hundreds of experiments as described in Ukkonen et al. (2020).

3.2 Choice of inputs and outputs

Our RRTMGP emulator predicts layer-wise optical properties from an input vector which contains gas mixing ratios, temperature, and log-pressure. The NNs take as input all the RRTMGP gases and output all g-points, which results in better computational intensity and efficiency than computing one band at a time, and the contributions from minor gases one gas at a time, as is done in the look-up-table kernels in RRTMGP (Ukkonen et al., 2020). In the shortwave, the NN outputs are absorption and Rayleigh cross-sections, while the longwave predictands are absorption cross-section and Planck fraction. Here, cross-sections refers to optical depth divided by the number of dry air molecules in a layer *N*. This allows generalization to arbitrary vertical grids, since optical depths are obtained in a separate step by multiplying the cross-sections with *N*. *Planck fraction* is the fraction of a band's total Planck function that is associated with each *g*-point, obtained by 3D interpolation in the original code. Like in RRTMGP, this is multiplied with the band-wise Planck function at a level or layer (interpolated from a look-up-table using the temperature of that level/layer) to get the Planck function for each longwave *g*-point. This retains a small look-up-table interpolation, but simplifies the NN model by requiring only ng outputs, instead of $3 \times ng$ to directly predict the Planck functions used in reference RRTMGP, or $2 \times ng$ to get the Planck functions in RRTMGP-NN. (The original code has one Planck variable for each layer and two for each layer interface, the upward and downward emission, whereas RTE+RRTMGP-NN has one for each layer and layer interface. ecRad only uses one Planck function, defined at layer interfaces). Reducing the number of NN outputs can substantially reduce NN complexity and runtimes, since most of the floating point operations occur in the final NN layer given $n_{gpt} = 112$ (SW) or 128 (LW) $> N_{neurons} = 16 - 48$. In this work, a single longwave model is used which predicts both absorption cross-sections and Planck fractions. This may not be the fastest approach but has the benefit of easing the optimization procedure described in the next section, and is also physically justified due to emission and absorption being inverse processes (monochromatic, directional emissivity and absorptivity are equal according to Kirchoff's law, although these characteristics do not apply to radiation parameterized by correlated-k methods).

In addition to predicting cross-sections instead of optical depths, to obtain good results with less complex NN models it is useful to preprocess both inputs and outputs to a high degree. Specifically, square root transformations are used for all outputs and some inputs to make their distributions more uniform, and afterwards the inputs are scaled to the 0-1 range and outputs are scaled to have roughly zero mean and unit variance using a variant of standardization that preserves correlations between different outputs (Ukkonen et al., 2020).

3.3 Can we optimize for fluxes or heating rates?

Using NNs only for gas optics presents a potential tuning challenge, as the variables we ultimately care about are radiative fluxes and heating rates - the output from the solver. We previously found it relatively easy to develop gas optics NNs which upon implementation in the radiation code result in low mean errors in fluxes and heating rates, but difficult to obtain accurate radiative forcings at the top-of-atmosphere or surface with respect to changes in the concentration of individual gases, especially minor gases. (Ukkonen et al., 2020). The problem is likely to stem from predicting aggregated optical properties, instead of computing the contribution from minor gases separately (as is done in reference RRTMGP), which is more efficient but leads to major gases dominating the loss function. Mostly accurate radiative forcings for CKDMIP gases were ultimately obtained via a time-consuming, iterative process where new models were continuously trained, evaluated, and the training data expanded. In this work we have attempted to automate the optimization with regards to fluxes, heating rates and forcings to at least some extent by adding two new techniques to the training methodology.

Firstly, errors in fluxes and heating rates were monitored during training. While these accuracy metrics can not be easily be used for optimizing the NN weights, they can be used as a criteria to know when to stop training (*early-stopping*), or to optimize NN hyperparameters. Therefore, a Python training program was written where the end of every epoch, the NN models are saved to a file, and the Fortran radiation program is called with the new model, passing the location as a command-line-argument. The Fortran program runs RTE+RRTMGP-NN on a validation dataset, and writes some error metrics to standard output, which are finally read by the training program. For validation we used the RFMIP dataset consisting of 100 profiles and 18 different perturbation experiments, since this allowed computing radiative forcing errors with respect to CH_4, N_2O, and errors in total forcing with respect to all RRTMGP gases. In addition, a benchmark line-by-line solution was available for this data, which allows computing the total error and not only NN error. Our goal was to develop NNs that have a similar level of accuracy as RRTMGP; that is, emulation errors should be smaller than parameterization error. The error metrics were thus normalized by the RRTMGP values, so that a value of one indicates the same level of performance as RRTMGP and larger values indicate worse performance. An overall ”radiation error” was computed by taking the root-mean-square value of a total of 8 metrics which differ slightly for the longwave and shortwave (Table 1). This overall metric was used in the early stopping criteria and the model weights from the best epoch (a minimum in the metric) were saved.

Metric	Longwave	Shortwave
MAE Heating rate	X	X
MAE Heating rate (present-day)	X	X
MAE Heating rate (preindustrial)		X
MAE Heating rate (”future-all”)		X
Bias surface downwelling flux		X
Bias TOA upwelling flux	X	
Bias TOA IRF (present-day - preindustrial)	X	
Bias TOA IRF (future - present-day)	X	
Bias TOA IRF (future - preindustrial)		X
Bias surface IRF (future - preindustrial)	X	X
Bias surface IRF CH4 (present-day - preindustrial)	X	X
Bias surface IRF N2O (present-day - preindustrial)	X	

Table 1: Metrics that comprise the overall ”radiation error”.

Second, a custom loss function was devised to minimize the error in the difference in y associated with different perturbation experiments, in addition to mean-squared-error of y, where y are the scaled NN outputs. The new loss function indirectly measures radiative forcing errors (albeit weakly due to a non-linear dependence between optical properties and broadband fluxes) and has the form:

$$loss = \alpha \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + (1-\alpha) \sum_{\substack{i=1 \\ i \text{ odd}}}^{N} \Big((y_{i+1} - y_i) - (\hat{y}_{i+1} - \hat{y}_i) \Big)^2,$$

where y and $\hat{y}$ are the target and NN output vectors, respectively. The second term measures the error in the difference in y between different perturbation experiments if the data is organized so that adjacent samples (of a total N training samples) correspond to different experiments but the same columns and vertical layers, which was achieved by transposing the data so that the experiment dimension is innermost. In addition, the experiments should be designed so that every odd element and its neighbour relate to the goal, which was minimizing the TOA and surface forcing errors of individual gases. Therefore, RFMIP-style experiments such as present-day versus future concentrations of all greenhouse gases, or 8X CO_2 versus preindustrial CO_2, should be avoided, as they can easily dominate the error compared to varying the concentration of minor greenhouse gases (which was the challenge to begin with). This requirement was only partially fulfilled since we wanted to make use of existing training data. Though rather convoluted, and requiring bespoke data, the approach does seem to reduce the forcing errors in practice as is illustrated in Figure 2.

In the end, there was still a substantial random element in results obtained, and several models were trained before settling on the final models (based on errors with respect to training data, and not the independent offline evaluation, which was only performed once). To obtain a satisfactory LW model the early-stopping criteria was furthermore loosened (to 60 epochs); it might be possible to minimize forcing errors by simply training a very large number of epochs, at the risk of overfitting if the training data is not very extensive. In addition, increasing the number of hidden neurons compared to Ukkonen et al. (2020) seemed to improve results slightly. The final LW model has 64 neurons in two hidden layers, and the SW models have 32 neurons in two hidden layers. All models use the ”softsign” activation function.

Future studies could explore directly minimizing flux and forcing errors when training NN-based gas optics models. Doing this via gradient descent optimization would require differentiating the radiative transfer solver to obtain the derivative of fluxes with respect to changes in optical properties (and NN weights), which should be possible using automatic differentiation tools like Autograd if the radiative transfer code was reimplemented in Python (Autograd, for instance, can differentiate Numpy code).

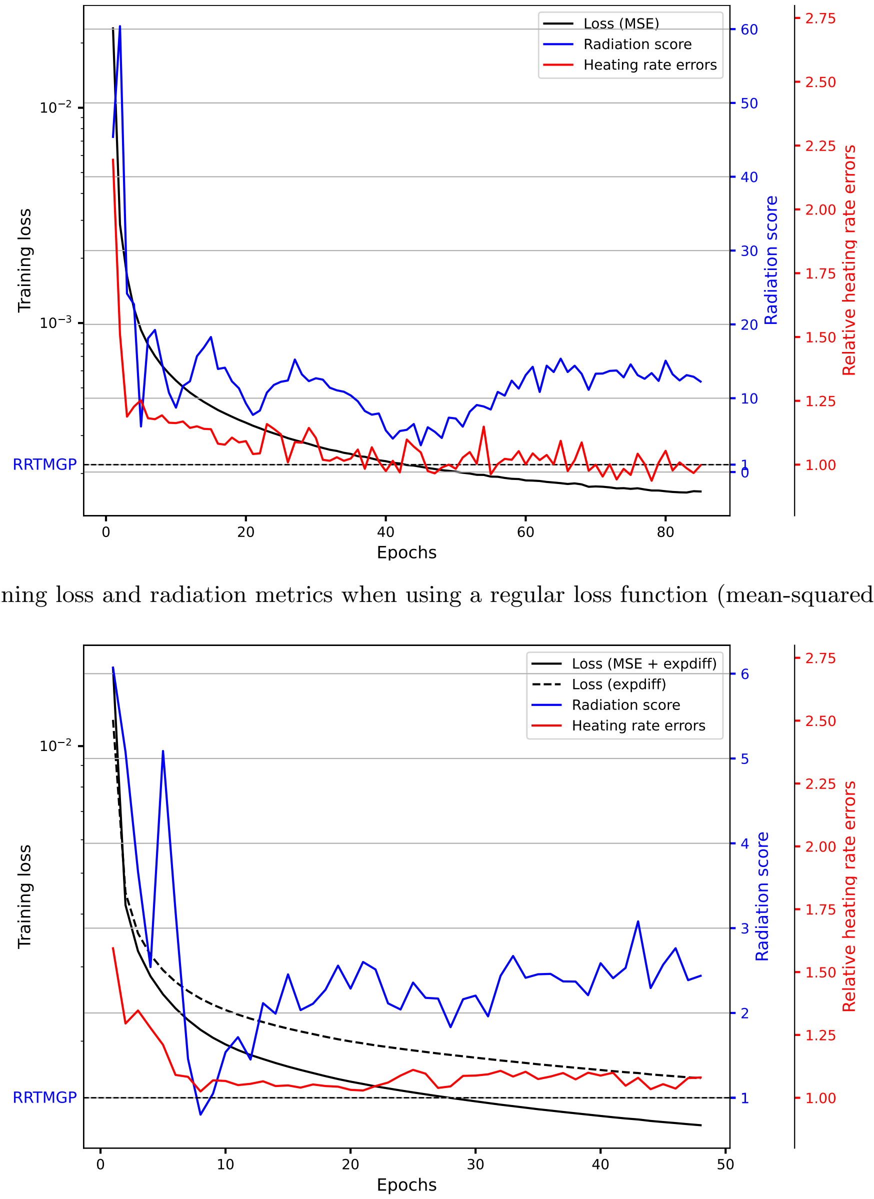
4 Results

Below we evaluate the accuracy and speed of ecRAD with different gas optics schemes (RRTMGP, RRTMGP-NN, and the older RRTMG scheme) in both an offline and online setting. The results were obtained using an optimized development version of ecRAD which refactors the TripleClouds and SPARTACUS solvers for better efficiency and includes the new RRTMGP(-NN) gas optics. Another optimization is that reflectances and transmittances are computed in the same numerical precision as the rest of the model (in the current operational version of ecRAD, these two-stream computations are always performed in double precision), which improves the single-precision performance of all solvers in ecRAD. The optimizations, which are described in a forthcoming paper, have a negligible impact on fluxes and heating rates while making TripleClouds significantly cheaper, and thus increase the share of the gas optics in the total runtime of ecRAD.

For both the offline and online evaluation, we configured ecRAD similarly, using a configuration planned for a future IFS cycle. The settings correspond to the operational configuration of ecRAD as described in Table 1 of Hogan and Bozzo (2018), except the solver, for which we use TripleClouds instead of McICA, and the cloud overlap assumption, where *exponential-random* (exponential decorrelation length for adjacent cloud layers but random overlap for layers separated by clear sky) is used instead of purely exponential overlap.

4.1 Speed-up

The runtime of ecRAD with different gas optics schemes was evaluated offline using 10,000 input profiles that were saved from a benchmark forecast run in the IFS, and a block size of 8 columns (equal to the block size ”NPROMA” in the IFS). Figure 3 shows timing results obtained on a single node of the new ECMWF AMD-based supercomputer in Bologna, to which the migration of ECMWF’s operational forecast is expected later in 2022.



(a) Training loss and radiation metrics when using a regular loss function (mean-squared-error).

(b) Training loss and radiation metrics when using the hybrid loss function.

Figure 2: Monitoring of heating rate error (solid red line, given by the mean of the heating rate metrics in Table 1), and the total radiation error (solid blue line, given by the RMS of the metrics listed in Table 1) when training the longwave gas optics model. The metrics are computed with respect to line-by-line data and normalized by the RRTMGP value. Also shown is the training loss (black lines). The larger radiation error when not using the hybrid loss function (a) was mostly due to a single metric, the surface radiative forcing of N2O (not shown).

With RRTMGP, the runtime of ecRAD is increased slightly due to the new gas optics being more expensive than the older RRTMG scheme (which is faster by a factor of 1.67). The relatively poor performance of RRTMGP is explained by short inner loops in the LUT code, where inner loops are over g-points in a band, which is only 12-16 for

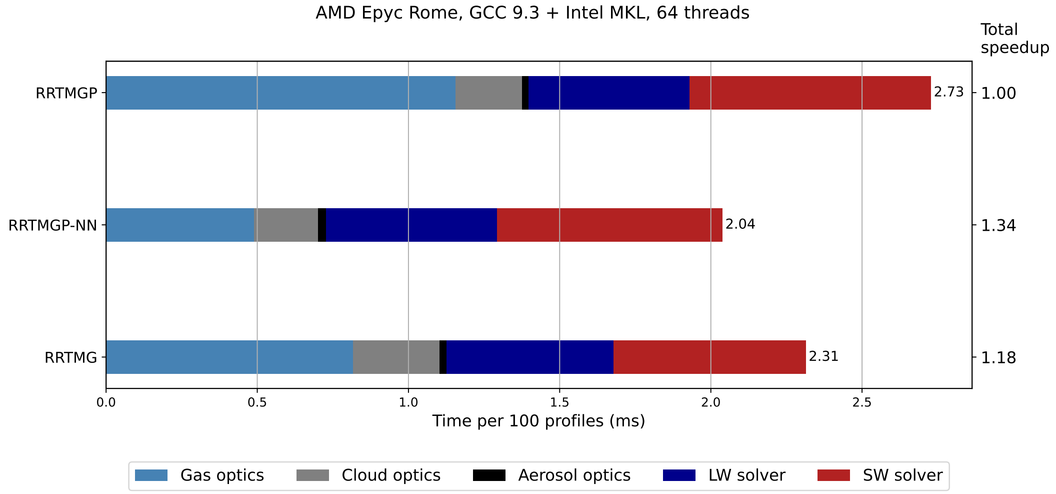


Figure 3: Runtime of ecRAD in single precision per 100 atmospheric profiles, broken down by component. Three gas optics schemes are compared: RRTMGP scheme with reduced spectral resolution (112 SW and 128 LW g-points, its neural network version (RRTMGP-NN), and the older RRTM-G scheme with 112 (SW) and 140 (LW) g-points. The values for each component were computed by taking the average of the per-thread values reported by the General Purpose Timing Library. CPU: 64-core AMD Epyc Rome. Software: GNU Fortran compiler version 9.3 and Intel MKL library 19.0.5 (used for general matrix-matrix multiplication (GEMM) in RRTMGP-NN)

the smaller k-distributions. However, the NN version of RRTMGP is faster than the look-up-table version by a factor of 2.36, and also faster than the old RRMTG scheme, leading to a total speedup of the radiation code by a factor of 1.13 compared to operationally used RRTMG.

4.2 Offline evaluation

Independent validation of the NN gas optics models was carried out by using data and tools from CKDMIP (”Evaluation 1” data). The accuracy of the new RRTMGP-NN longwave model, relative to a line-by-line benchmark, is first shown in Figure 4 for the present-day scenario. The fluxes and heating rates have very similar accuracy to the RRTMGP look-up-table code (Fig. 5), with the NN actually showing a smaller bias in upwelling LW flux. The results were similar for the pre-industrial and future scenarios, with the NN achieving the same level of accuracy as RRTMGP (not shown). It should be noted that the RRTMGP LW results were not produced using the original RTE+RRTMGP package, which uses two Planck source functions for half-levels which are then combined into one, and the LW results may be slightly impacted by the simpler computation of Planck source. For simplicity the RRTMGP-NN scheme without look-up-tables is hereafter be referred to as ”reference RRTMGP”.

In the shortwave, the CKDMIP results are similarly encouraging, with the NN having almost identical accuracy to RRTMGP across different scenarios (not shown). For instance, the present-day RMSE in surface downwelling flux was 0.80 W m^{-2} for RRTMGP-NN and 0.78 W m^{-2} for the original scheme, and heating rate RMS errors were 0.256-0.257 K $^{-d}$ above 4 hPa and 0.056-0.057 K $^{-d}$ below this for both schemes. In general, the close emulation of RRTMGP was already demonstrated in Ukkonen et al. (2020) and the remaining results are not discussed (the full results will be made available on the CK-

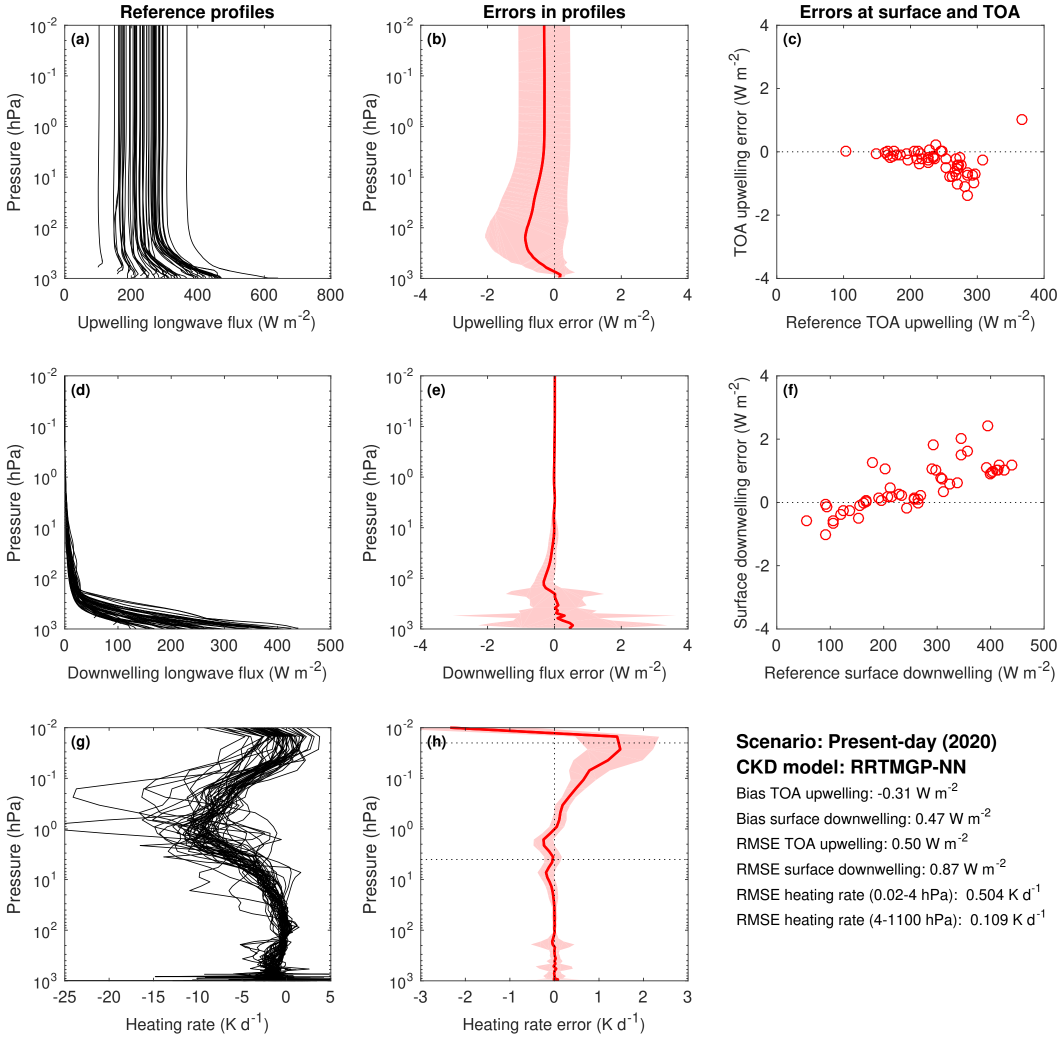


Figure 4: Evaluation of RRTMGP-NN longwave fluxes and heating rates using the 50 independent profilesof the CKDMIP Evaluation-1 dataset with present-day concentrations of greenhouse gases. The left column shows the reference profiles from LBL calculations, the middle column shows biases (solid lines) and 95th percentile of errors (shaded area), and the right column shows errors in upwelling TOA and downwelling surface fluxes. the middle and right columns show errors. The RRTMGP-NN calculations use an identical radiative transfer solver as the reference calculations, with four angles per hemisphere.

DMIP website at `https://confluence.ecmwf.int/display/CKDMIP`). One notable difference is that the top-of-atmosphere and surface forcings with respect to N_2O, CFC_{11} and CFC_{12} have been improved and are now almost perfect (Fig. 6). Finally, we note that the new k-distributions with 112 (SW) and 124 (LW) g-points seem to trade only a little accuracy for a lot of speed: except for the LW heating rates in the mesosphere, the results are overall quite similar to the RTE+RRTMGP results obtained with the original k-distributions with almost double as many g-points.

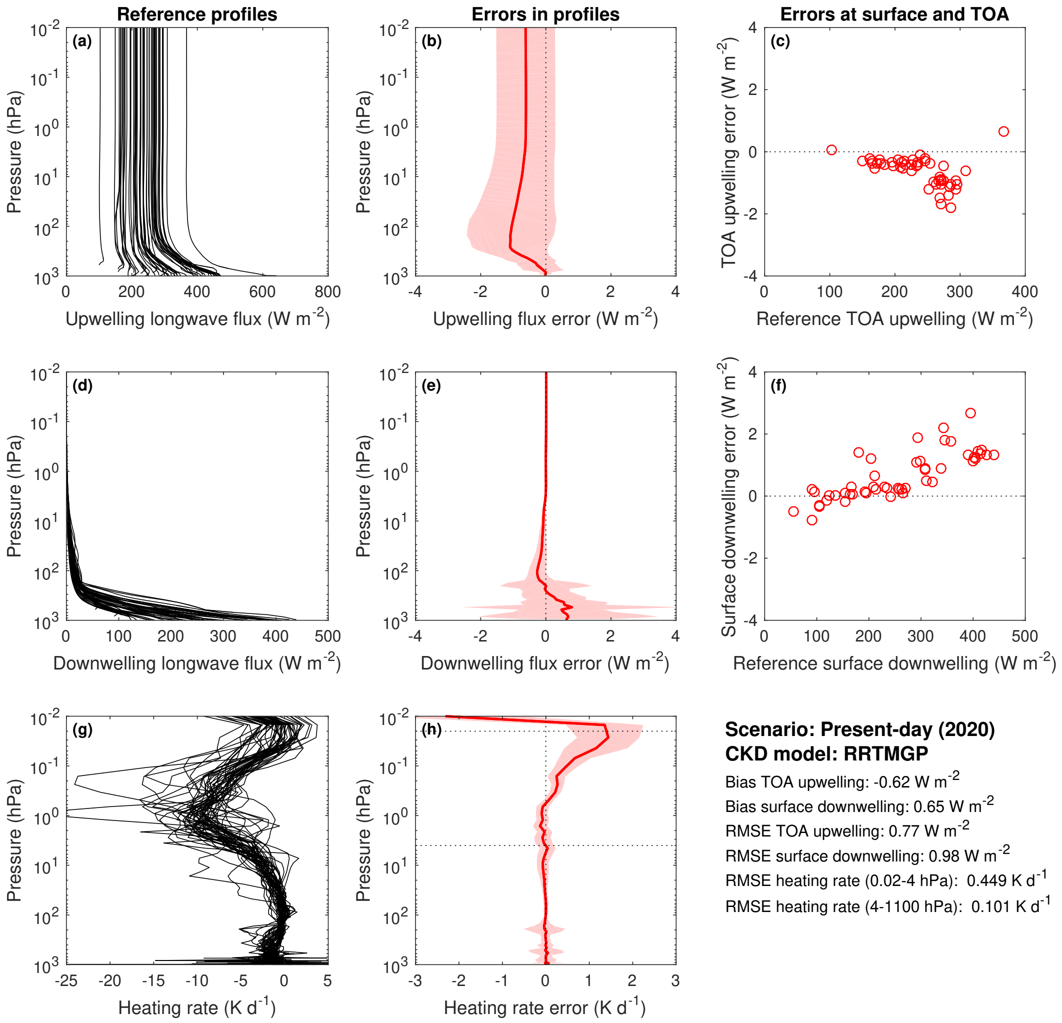


Figure 5: As in Fig. 4 but for the original RRTMGP scheme used for training RRTMGP-NN.

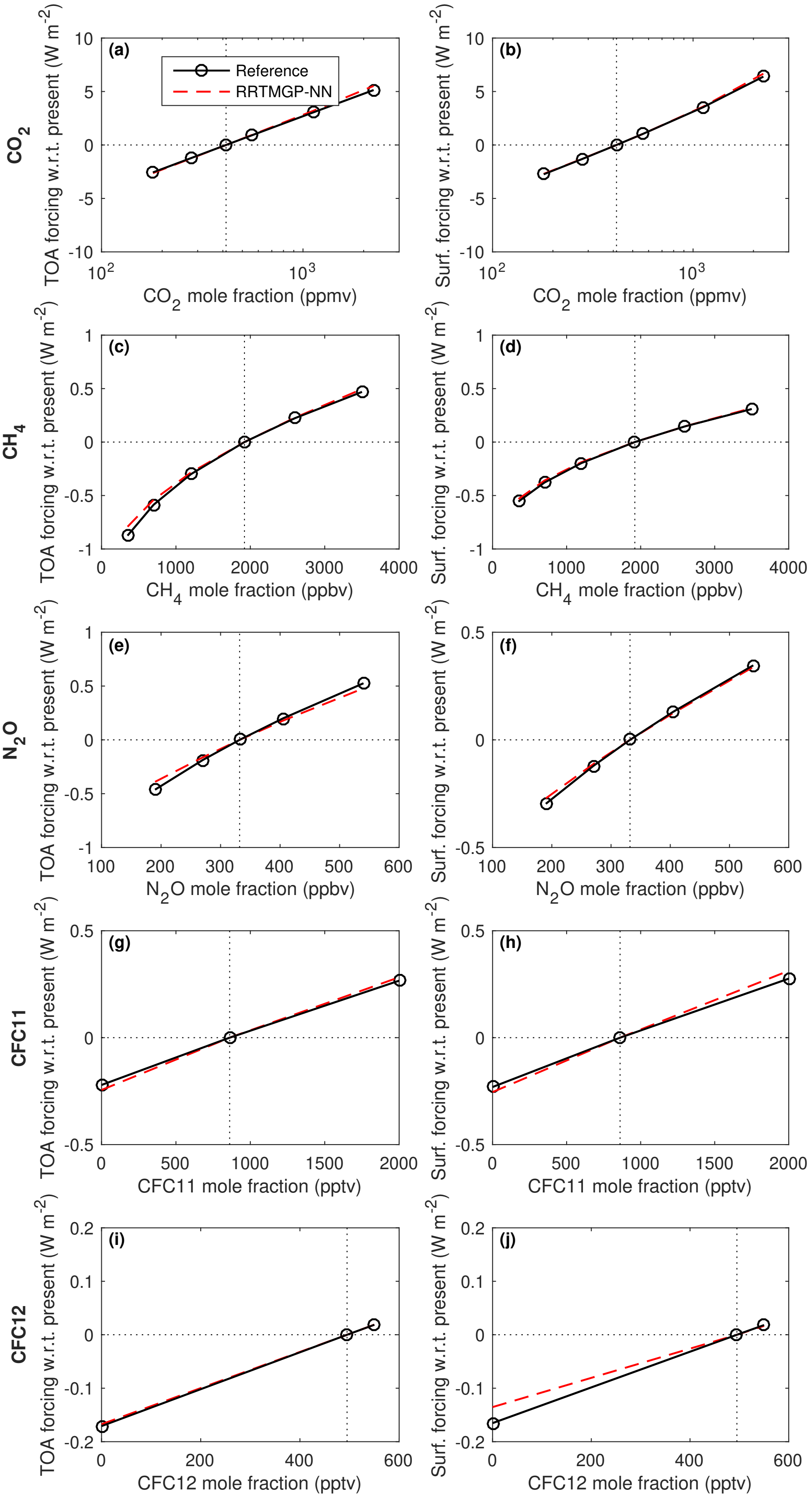


Figure 6: Comparison of RRTMGP-NN and reference LBL calculations of instantaneous longwave clear-sky radiative forcing at top of atmosphere (left column) and surface (right column) when perturbing different greenhouse gases (rows), averaged over the 50 profiles in the CKDMIP Evaluation 1 dataset.

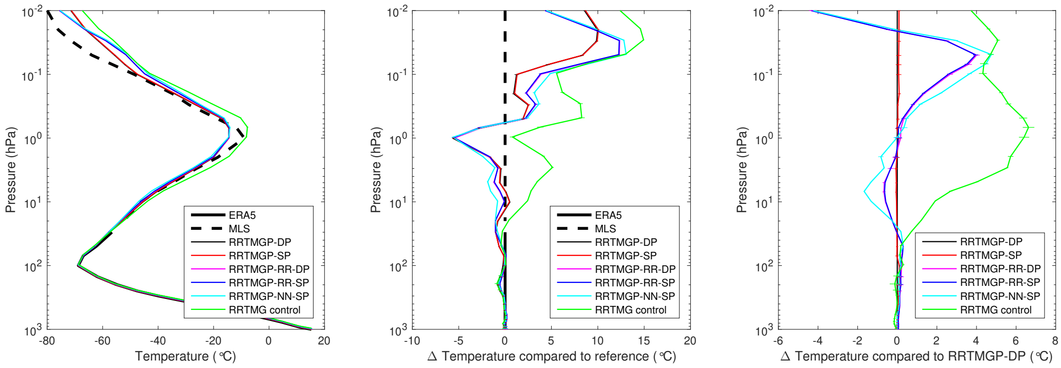


Figure 7: (a) Mean temperature, (b) difference against a reference dataset consisting of the MLS climatology above the 20-hPa height level and ERA5 below this level, and (c) difference against RRTMGP-DP. RRTMGP-SP = single precision run (SP) using RRTMGP (original k-distributions with higher spectral resolution), RRTMGP-DP = double precision (DP) using RRTMGP (original k-distributions), RRTMGP-RR-DP = newer RRTMGP k-distributions with Reduced spectral Resolution (RR), RRTMGP-NN-SP = Neural Network version of RRTMGP trained on the RR distributions, RRTMG control = control run using the current operational configuration of ecRAD (older RRTMG gas optics and McICA solver).

4.3 Prognostic evaluation

In this section we describe results from a prognostic evaluation of RRTMGP-NN and the original RRTMGP scheme, which was carried out by performing ”climate runs” with the IFS model. The motivation for performing longer simulations was that changes in the radiation scheme tend to have a larger impact on the climate of the model than, for instance, short-term forecasts of surface temperature.

The model simulations consisted of four atmosphere-ocean coupled simulations 13 months long initialized on 1 August of the years 2000, 2001, 2002 and 2003. After a 1-month spin-up for each simulation, the remaining 12 months were averaged over each simulation. This configuration is very similar to that used in section 5 of Hogan and Bozzo (2018) to evaluate the impact of changes to the radiation scheme; the simulations are long enough to capture fast atmospheric and land-surface processes that respond to changes in the treatment of radiative transfer, but short enough that the response is not significantly affected by the longer-term changes to ocean circulation. The one-year forecast length also matches the longest operational forecast length used in ECMWF’s seasonal forecasts. The control model configuration was as in operational IFS model cycle 47r3 but with a horizontal resolution of $\mathrm{T_{Co}}199$ (around 60 km) and 137 vertical levels. The radiation scheme was called every hour.

The impact of different gas optics schemes on annual-mean temperature from the surface to the lower mesosphere is shown in Figure 7. Because RRTMGP has to our knowledge not been properly tested in single precision yet, both single precision (SP) and double precision (DP) runs were done with the reference RRTMGP scheme. The original RRTMGP k-distributions (”RRMTGP”) were tested in addition to the newer distributions with reduced spectral resolution (”RRTMGP-RR”). The NN version of the RRTMGP-RR scheme is only evaluated in single precision (internally, the RRTMGP-NN code always uses SP, as higher numerical precision does not benefit NNs). In general, larger differences between the runs are only seen in the lower stratosphere and upper mesosphere. Comparison against a reference dataset based on the Microwave Limb Sounder (MLS) instrument above 20 hPa, and ERA5 reanalysis data below this level depicted in Fig. 7 (b) shows that the newer gas optics schemes are all in closer agreement with the MLS compared to the older RRTMG scheme, thanks to much more recent solar spectrum that reduces the UV radiation by around 8%. In general, the stratosphere and mesosphere are very sensitive to heating-rate differences.

A height-latitude cross section of temperature likewise shows larger differences between the old RRTMG scheme and RRTMGP than between different RRTMGP configurations and the NN version (Fig. 8). A strong warm bias in the stratosphere is evident for RRTMG but not any versions of RRTMGP, although the RRTMGP(-NN) runs do show weak stratospheric warm bias over high latitudes and a clearer cold bias in the tropical stratosphere. The differences between RRTMGP-(NN) runs look like noise as opposed to anything consistent: for instance, the RRTMGP-NN run in single precision seems closer to the double precision run using the scheme it is trained on (RRTMGP-RR) than the RRTMGP-RR single precision run, while the latter resembles the SP run using the original RRTMGP with high spectral resolution.

2-metre temperature compared against the double precision run with the high-spectral resolution version of RRTMGP suggests that the signal in the surface temperature is also small compared to natural variability (Fig. 9), as the differences between SP and DP runs are at least as big as between different gas optics schemes. No strong signal can be made out the clear-sky net longwave flux at top-of-atmosphere either (Fig. 10).

Averaging only 4 years of data means there is inevitably noise present, but this variability hiding the impact on model climate from using the NN gas optics, combined with a detailed offline evaluation of the scheme which shows practically identical results to the original scheme, demonstrates that there are is no clear disadvantage to using the NN version of RRTMGP.

References

Chevallier, F., Chéruy, F., Scott, N., & Chédin, A. (1998). A neural network approach for a fast and accurate computation of a longwave radiative budget. *Journal of applied meteorology*, *37*(11), 1385–1397.

Cotronei, A., & Slawig, T. (2020). Single precision arithmetic in echam radiation reduces runtime and energy consumption. *arXiv preprint arXiv:2001.01214*.

Hogan, R. J., & Bozzo, A. (2018). A flexible and efficient radiation scheme for the ecmwf model. *Journal of Advances in Modeling Earth Systems*, *10*(8), 1990–2008.

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. Retrieved from `https://gmd.copernicus.org/preprints/gmd-2020-99/` doi: 10.5194/gmd-2020-99

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.

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2008). Decadal climate simulations using accurate and fast neural network emulation of full, longwave and shortwave, radiation. *Monthly Weather Review*, *136*(10), 3683–3695.

Liu, Y., Caballero, R., & Monteiro, J. M. (2020). Radnet 1.0: Exploring deep learning architectures for longwave radiative transfer. *Geoscientific Model Development*, *13*(9), 4399–4412.

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., Forster, P. M., & Stevens, B. (2016). The radiative forcing model intercomparison project (rfmip): experimental protocol for cmip6. *Geoscientific Model Development*, *9*(9), 3447–3460. Retrieved from `https://www.geosci-model-dev.net/9/3447/2016/` doi: 10.5194/gmd-9-3447-2016

Pincus, R., Mlawer, E. J., & Delamere, J. S. (2019). Balancing accuracy, efficiency, and flexibility in radiation calculations for dynamical models. *Journal of Advances in Modeling Earth Systems*, *11*(10), 3074–3089.

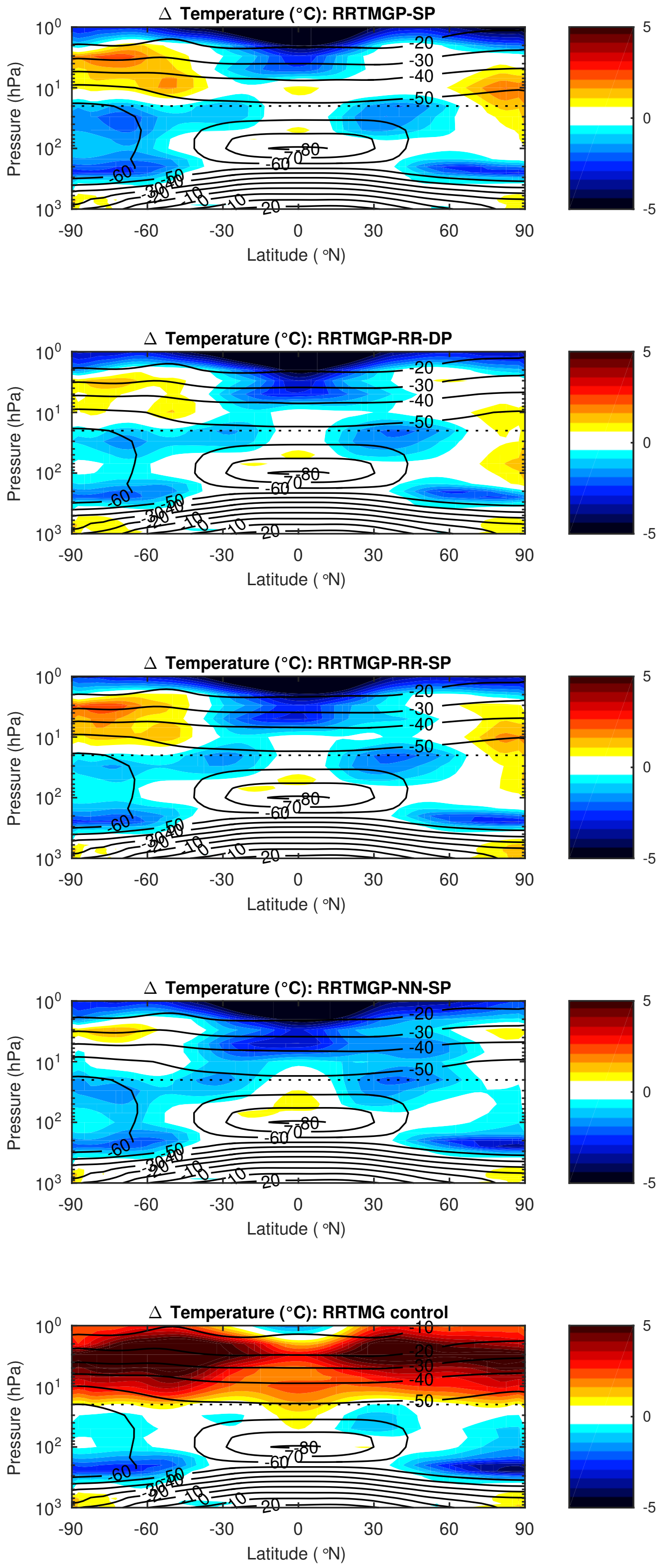
Roh, S., & Song, H.-J. (2020). Evaluation of neural network emulations for radiation parameterization in cloud resolving model. *Geophysical Research Letters*, *47*(21), e2020GL089444.

Song, H.-J., & Roh, S. (2021). Improved weather forecasting using neural network emulation for radiation parameterization. *Journal of Advances in Modeling Earth Systems*, *13*(10), e2021MS002609.

Ukkonen, P. (2021). Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer. *Journal of Advances in Modeling Earth Systems*, e2021MS002875. doi: 10.1029/2021MS002875

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

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.



∆ Temperature (°C): RRTMGP-SP

∆ Temperature (°C): RRTMGP-RR-DP

∆ Temperature (°C): RRTMGP-RR-SP

∆ Temperature (°C): RRTMGP-NN-SP

∆ Temperature (°C): RRTMG control

Figure 8: Similar to Fig. 7 but showing the the height-latitude cross section of mean temperature (black contours) and temperature difference (colors) against the reference datasets, and only until 1 hPa.

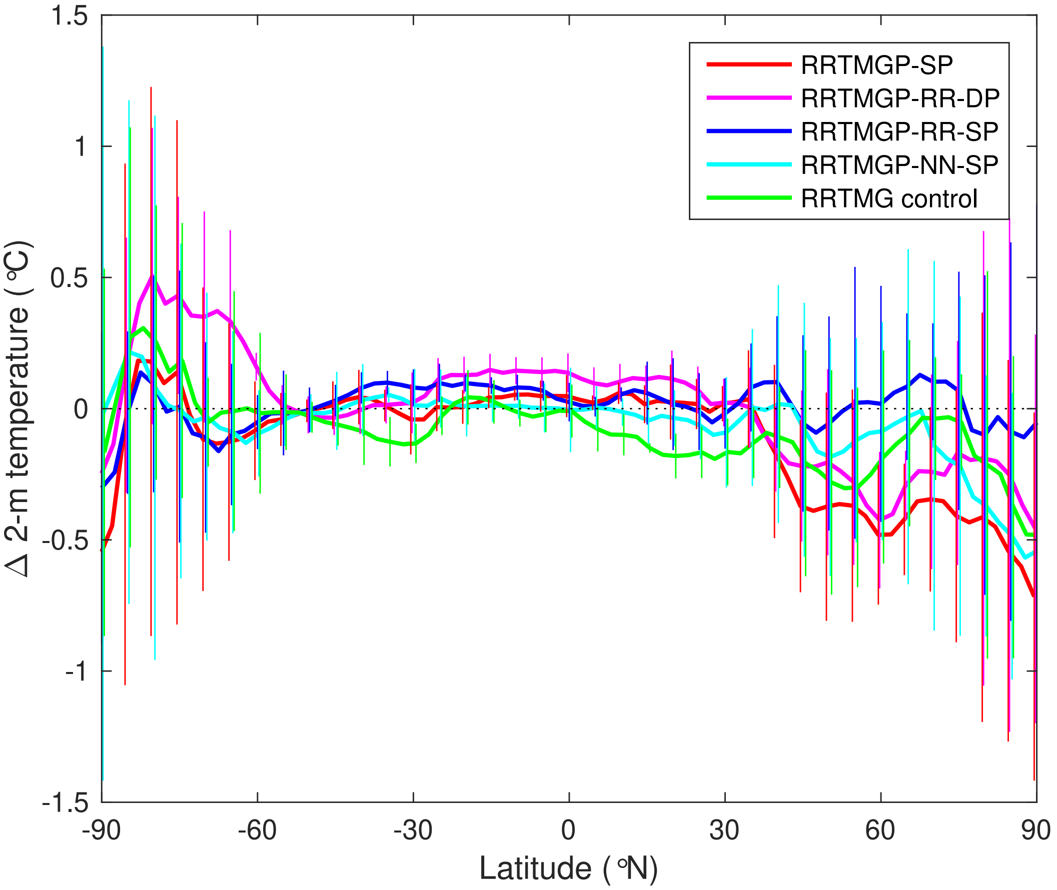


Figure 9: Mean 2-metre temperature difference against RRTMGP-DP (higher spectral resolution).

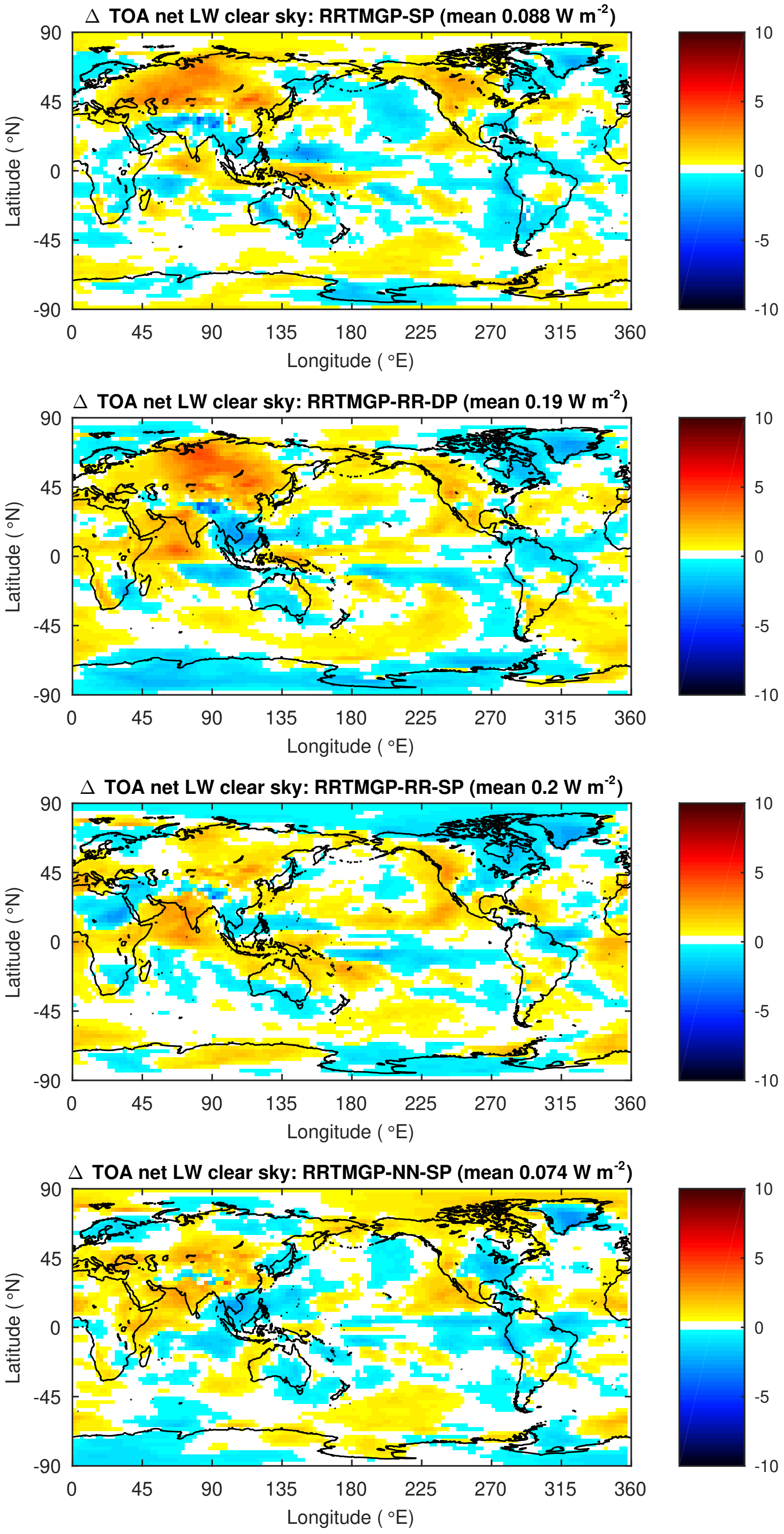

△ TOA net LW clear sky: RRTMGP-SP (mean 0.088 W m-2)
△ TOA net LW clear sky: RRTMGP-RR-DP (mean 0.19 W m-2)
△ TOA net LW clear sky: RRTMGP-RR-SP (mean 0.2 W m-2)
△ TOA net LW clear sky: RRTMGP-NN-SP (mean 0.074 W m-2)
Latitude (°N)
Longitude (°E)
90
45
0
-45
-90
0
45
90
135
180
225
270
315
360
10
5
0
-5
-10

Figure 10: Difference in net longwave clear-sky radiation at top of atmosphere against RRTMGP-DP (higher spectral resolution).

6

Paper 4: Optimizing the ecRAD radiation scheme with a new gas optics scheme results in affordable computations of 3D cloud radiative effects

6.1 Abstract

Radiative transfer codes are some of the most expensive components of large-scale dynamical models. In this work we attempt to improve the performance of ecRAD, a state-of-the-art modular radiation code. A major focus was to improve the performance with ECCKD, a new gas optics model based on the correlated-k method that uses only 16-32 pseudo-monochromatic spectral intervals (k-terms). The small spectral dimension reduces the number of computations performed in the whole radiation scheme, but leads to inefficiency in the reference code due to short loop lengths. A combination of higher-level code restructuring and kernel-level optimizations are performed, targeting the TripleClouds and SPARTACUS radiative transfer solvers, where the latter can represent cloud 3D radiative effects.

We find that for computations without vertical loop dependencies (calculations of layer-wise reflectance and transmittance), the vertical and the innermost spectral dimension can be collapsed together to increase the vector length and reduce the number of procedure calls. This can be done for clear-sky as well as cloudy-sky computations, but the latter requires batching together adjacent cloudy layers.

The refactored Tripleclouds and SPARTACUS solvers make ecRAD roughly 2.3 and 2.8 times faster than before when using ECCKD, respectively. As a result

of the optimizations and a smaller spectral resolution, SPARTACUS combined with ECCKD is computationally cheaper than the operational configuration of ecRAD in the IFS.

6.2 Introduction

This paper describes various optimizations to ecRAD with a focus on higher-level code restructuring techniques to expose more parallelism, but also changes in individual kernels (including minor changes to underlying equations), for instance to avoid the use of double precision in numerically sensitive calculations. Many of these optimizations therefore require domain knowledge, in addition to an intermediate knowledge of computer hardware. This approach may be contrasted with a more detailed performance analysis (such as looking at assembly code) and algorithmically less invasive optimizations that might be carried out by a computing expert striving for numerically identical results. To achieve the best performance, these two approaches should probably be combined in a close collaboration between domain scientists and computer scientists.

Instead, we follow a simple optimization strategy where we use the GPTL timing library to manually instrument ecRAD code and get a profile of the runtimes as well as FLOPS (Floating point operations per second) counts of different sections of the code. Although FLOPS is not always a useful metric, radiation codes are generally computationally intensive (as opposed to other parts in NWP models which tend to limited primarily by memory bandwidth), and ecRAD code sections with significant runtimes and low FLOPS indicated room for improvement in the overall time-to-solution by optimizing these sections. The targeted radiative transfer solvers were the longwave (LW) and shortwave (SW) versions of TripleClouds and SPARTACUS Hogan et al. (2016).

We note that thorough performance refactoring of SPARTACUS is a laboursome undertaking: in addition to a fairly flat computational profile, it is a more sophisticated radiative transfer solver than most operationally used schemes, and contains over 1000 lines of code, excluding subroutines, in both the SW and LW solvers. In total, many person months were spent on the refactoring. However, this effort should we well-placed as SPARTACUS is the only radiation scheme implemented in a global weather model that is capable of representing 3D radiative effects at a relatively low computational cost, being only 5 times more expensive than the operational configuration of ecRAD Hogan and Bozzo (2018). (Full 3-D radiative transfer codes based on Monte Carlo methods, by contrast, are several orders of magnitude more expensive). This gap is further reduced by the use of ECCKD. ECCKD is new gas optics scheme that uses the full-spectrum correlated-k (FSCK) method in the LW, and a carefully designed partioning of k-terms, to reduce the number of k-terms drastically compared to current gas optics schemes using 100-200 k-terms. The candidate ECCKD LW and SW models used here have 32 k-terms (also known as g-points). The main motivation for the present work is to eliminate the remaining performance gap, and make SPARTACUS computationally efficient enough (when combined with ECCKD) to be considered for operational NWP and climate applications.

Although the code refactoring effort carried out was significant, many of the most effective changes had to do with exposing more parallelism in a manner which may be applicable to other radiation codes or even other physical parameterizations.

6.3 Higher-level refactoring to expose more parallelism

Beginning with a trivial change in the code that improves efficiency, the computation of clear-sky reflectance and transmittance, which is done also for cloudy layers, can be made much faster by simply moving the kernel call outside the vertical loop and instead collapsing the vertical dimension with the innermost dimension. The performance of the optimized shortwave reflectance-transmittance kernel (which includes lower-level optimizations described later in Sections 6.4.1 - 6.4.2) as a function of the vectorized dimension N is shown in Figure 6.1. When the vertical dimension is fully collapsed with the spectral dimension, the performance with ECCKD is roughly doubled compared to the previous code layout which in this case results in a loop length of only $ng = 32$.

The lack of loop dependencies in the vertical dimension can also be exploited to increase loop lengths when computing the reflectance and transmittance of cloudy layers and regions, but this requires batching together the two cloudy regions and/or adjacent cloudy layers. The best way to do this depends on the particular solver.

6.3.1 SPARTACUS-SW

SPARTACUS represents cloud 3-D radiative effects by adding extra terms to the two-stream equations to represent lateral transport between clear and cloudy regions. The coupled system of equations can be solved by a method based on the matrix exponential. In both LW and SW SPARTACUS, these matrix exponentials are a computational hotspot, with the $expm$ kernel accounting for roughly 40% of the combined runtime of the two solvers. The matrix exponential is performed for each "3D" g-point in each cloudy layers, where 3D effects are not considered for *g*-points which have very large optical depths. Because the individual matrices for which the matrix exponential is computed have small sizes corresponding to the total number of clear and cloudy regions, $(nreg \times 3, nreg \times 3) = (9, 9)$, they are placed non-contiguously in memory and the *g*-point dimension is vectorized instead.

For SPARTACUS, the matrix exponential computations for adjacent cloudy layers can be grouped together. Recognizing that $ng3D$ in cloudy layers is typically close to ng, 3D computations can be performed for all *g*-points without much redundancy, and the *g*-point dimension collapsed with the vertical dimension by grouping together adjacent cloudy layers. This was implemented with a *do while* loop which checks if any cloudy layers still exists and finds the top and bottom of this "extended" cloudy layer. The result is a much longer typical vector length for the matrix exponential computations ($ng \times nlev_{cloud-depth}$ instead of $ng3D \approx ng$), where the individual matrices have sizes of $(nreg \times 3, nreg \times 3) = (9, 9)$ and are placed non-contiguously in memory.

6.3.2 SPARTACUS-LW

In the longwave the fraction of *g*-points which have optical depths small enough for 3D computations to matter is much lower than in the shortwave, and doing them in all *g*-points would result in a great deal of redundancy. Therefore, the code was restructured to collect all the "3D" *g*-points from adjacent cloudy layers, with variable $ng3D$, into larger arrays with inner dimension $ng3D_{tot}$. This increases the code complexity and overhead somewhat but is worth it as the time spent in $expm$ is more than halved when using ECCKD due to avoiding very inefficient calls with small loop lengths. This change made the longwave solver faster by roughly a third.

6.3.3 TripleClouds-SW

In shortwave TripleClouds, reflectance-transmittance computations are batched similarly to SPARTACUS-SW, by grouping together adjacent cloudy layers. This leads to a vectorized dimension of $2 \times ng \times nlev_{cloud-depth}$ due to TC having two cloudy regions).

6.3.4 TripleClouds-LW

Computations are batched only over the 2 cloudy regions (and *g*-points) and not layers as this was slightly faster on tested hardware and software platforms, but to achieve better performance on platforms with longer vector lengths (such as GPUs or CPUs with AVX-512 instructions) it is likely worth the slight increase in memory use to batch over the vertical dimension also. For now, this was not implemented to avoid sacrificing performance on current hardware or having to write more complex code.

6.4 Lower-level optimizations

6.4.1 Single precision computation of reflectance and transmittance

To compute reflectance and transmittance using the two-stream approximation Meador and Weaver (1980), ecRAD previously always did these calculations in double precision to ensure correctness of results, as the underlying equations are numerically sensitive. We find that the code can be made mostly accurate in single precision by using a different threshold value for the variable *k* in the single precision case, but that rare combinations of the input variables (single-scattering albedo, optical depth and asymmetry factor) could still cause unphysical results and subsequent crashes. This issue was solved by constraining the output variables to ensure $Tdir <= 1 - Tnoscat - Rdir$. Here, $Tdir$ is the direct transmittance (fraction of incident direct radiation that is scattered in the forward direction), $Rdir$ is the direct reflectance and $Tnoscat$ is the transmittance of the direct beam with no scattering. The combined effect of the adjusted threshold and the physical security is that the mean absolute differences in SW and LW net fluxes between a reference double precision computation using ecRAD with the TripleClouds solver, and the equivalent computation performed fully in single precision, was around 0.001 Wm^{-2} for 10000 columns saved from a high-resolution IFS simulation. The biases in heating rates were close to zero.

6.4.2 Pipelining and vectorization

Similarly to a car assembly line which can produce cars at a rate that is significantly faster than the time taken to produce a single car, microprocessors have a level of parallelism that comes from *pipelined* instructions. Because pipelined instructions include a wind-up and wind-down phase where microprocessor units are idling for a given number of cycles - known as latency or *depth* - the throughput (number of operations per cycle) when executing *N* independent operations with a pipeline of depth of *m* is given by (Hager and Wellein, 2010):

$$p = \frac{1}{1 + \frac{m-1}{N}}$$

Prior to restructuring the code, the reflectance-transmittance kernels were called inside a vertical loop and N was equal to the number of g-points. With ECCKD, $N = 32$, and to obtain a decent efficiency of $p = 0.64$ results per cycle, we arrive at $m = 19$. However, complex calculations can have much longer latencies than this, with the exponential function alone having a longer latency. The computations of reflectance and transmittance are very involved and include many high-latency operations such as divide. This can easily lead to the instruction stream being stalled ("pipeline bubble"). Vector or superscalar parallelism makes the situation even worse as multiple identical pipelines operating in parallel decreases the loop length of each pipe.

Knowing that the exponential computation alone has a long latency, simply moving it outside of the long SIMD-vectorized loop with other complex arithmetic significantly improves performance by alleviating such a pipeline stall.

Even after the separately vectorized exponential, increasing N by collapsing the vertical and g-point dimension together is still highly beneficial for the two-stream reflectance-transmittance computation (Figure 6.1), even more so than for SPARTACUS due to the more complex instructions in the two-stream equations than the simple multiply adds which dominate $expm$.

6.4.3 Hand-optimized matrix operations: loop unrolling and avoiding redundancy

Compilers can in some cases unroll loops automatically automatically, but if the loop bounds are not known at compile time the compiler may not know it is advantageous. More involved code patterns may also prevent automatic loop unrolling. In SPARTACUS, the individual matrices to be multiplied are small and loop unrolling is beneficial, but the individual matrices are stacked non-contiguosly in memory as the inner dimension is over g-points. The tested compilers did not unroll the loops automatically even when the outer dimensions (shape of individual matrices) was known at compile time.

Some redundant computations can be identified and removed, e.g. matrix-matrix multiplication kernels can take advantage of sparsity and repeated elements in the matrices that are used in the shortwave SPARTACUS computations.

6.4.4 Declaring ng at compile time

6.4.5 Other optimizations

- Removing conditionals. Conditionals within a vectorized loop to check for positive values in code sections where optical properties from gases, clouds

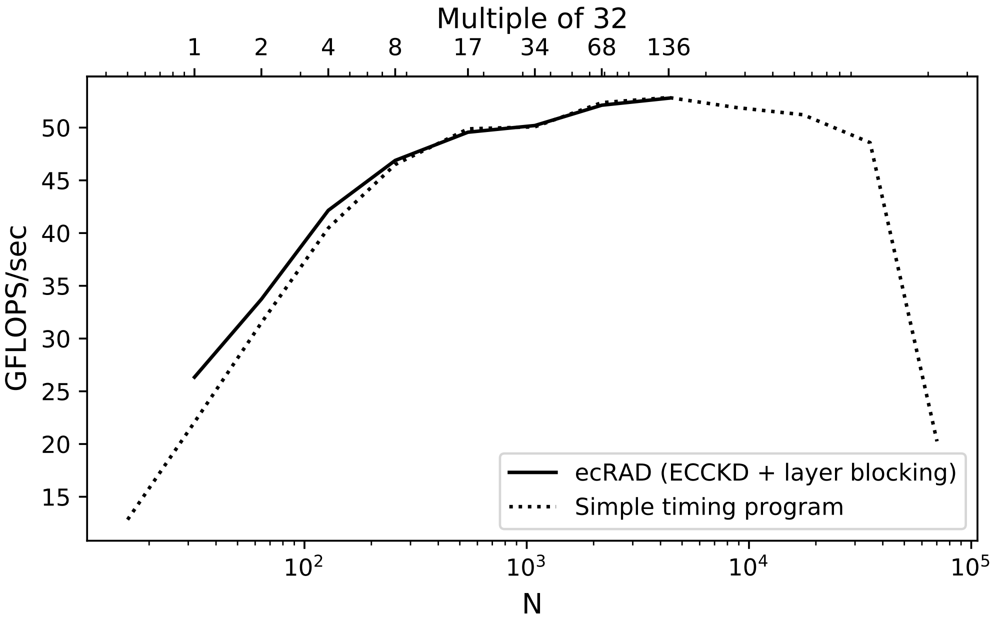


Figure 6.1: Serial single precision performance of the the optimized shortwave reflectance-transmittance kernel (y-axis) versus loop length (x-axis). The solid black line shows the performance as measured within a realistic program running the full radiation code for 7320 columns with a column block size of 8, ECCKD gas optics, the TripleClouds solver, and blocking also in the vertical dimension with different block sizes (top x-axis) to test the impact of different N. Conveniently, the performance peaks around N corresponding to the number of *g*-points in ECCKD (32) times vertical levels in the IFS (137), meaning that collapsing the *g*-point dimension with the full vertical dimension results in optimal performance on this platform (AMD Ryzen 9 3900, GNU Fortran 9.3). A simple timing program which tests a wider range of N (dotted black line) shows that considerably larger spectral and/or vertical dimensions can also be accommodated before an inevitable performance drop-off when the arrays can no longer fit in faster cache.

and aerosols were combined were replaced with the use of max(*value*, *some number*) in the denominator to protect against division by zero, recognizing that when the denominator was zero, the numerator was also zero and so the second argument can be almost any non-zero number. In other cases with "true" conditionals, placing them in a separate preparation loop improved performance by ensuring vectorization of the computationally heavier parts of the loop.

- Merged broadband flux computations. The last step in the solver is to compute broadband fluxes by summing the fluxes defined at *g*-points and the three regions. In the shortwave, there are three variables for which this reduction over two dimensions is performed: upwelling flux, downwelling flux, and direct downwelling flux. These reductions can be made more efficient by doing them all in a single loop over *g*-points with the SIMD REDUCTION clause in OpenMP, and unrolling the sum over regions, which increases the computional intensity compared to having three separate calls to the sum intrinsic.

- Faster computation of longwave derivatives. This final component in the longwave solver was relatively expensive owing to tiny matrix-vector multiplications (m=nreg) repeated over *ng g*-points, followed by a multiplication with transmittance(ng,nreg) and finally a sum over ng and nreg, within each level and column. In the case of nreg=3, the matrix-vector computations, multiplication and a partial sum operation (over regions) can all be combined in a single vectorized loop over *g*-points by inlining and manually unrolling the region dimension, roughly doubling performance.

- Avoiding unnecessary temporary arrays. In many code sections one or more several temporary arrays could be avoided, for instance by using the output array(s) of a subroutine for intermediate computations as well. This could lead to small performance gains particularly when the temporary was used within the assignment of the output array. Code clarity can be retained by the use of $associate$.

6.5 Results

Figure 6.2 shows the runtimes achieved using ecRAD with different gas optics and solvers and both the reference ("OPT") and optimized ("OPT") versions of the code, with a breakdown of the different components in ecRAD. Here, "OPT"

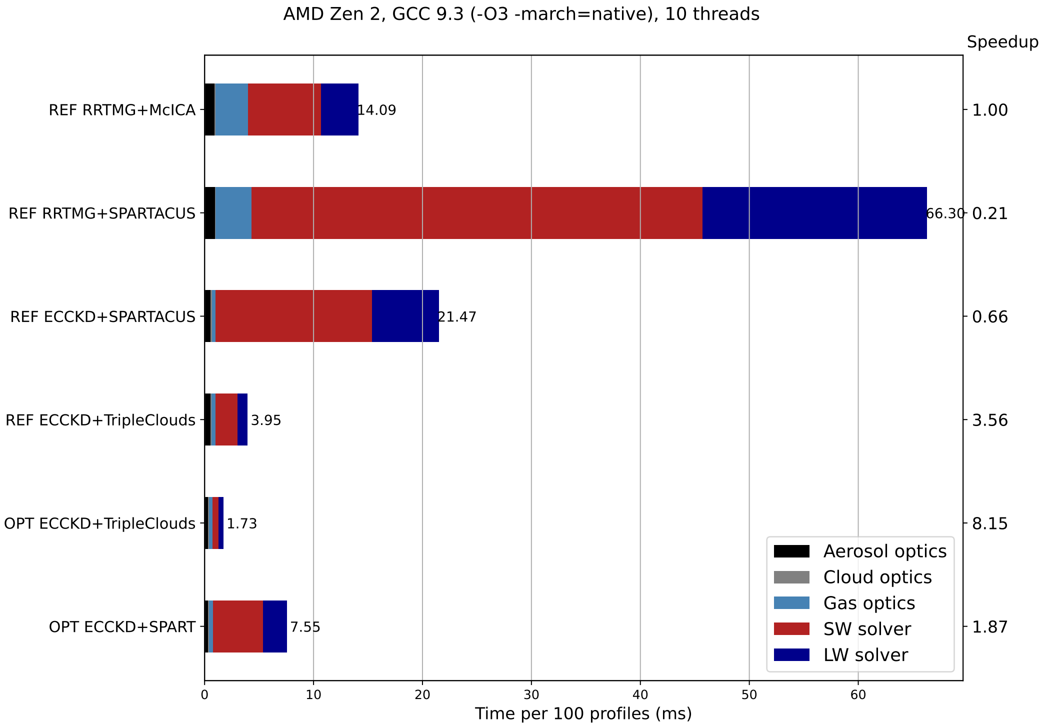


Figure 6.2: Time per profile for different configurations and code versions of ecRAD. "REF" = reference code, "OPT" = optimized code.

includes the cumulative impact of all optimizations described in the previous section. The values were obtained using a test case of 10,000 profiles saved from a high-resolution IFS run (where the vertical grid contains 137 levels), and a block size of 8 columns. On the tested platform, the SPARTACUS solver makes reference ecRAD roughly 5 times more expensive when using the RRTMG gas optics. When combining SPARTACUS with the new ECCKD gas optics, the number of computations is drastically reduced, and the time per profile is only about 40% more expensive than RRTMG+McICA, which is current the operational configuration in the IFS.

Comparing the optimized and reference versions of the code for the same configuration, ECCKD and SPARTACUS, the total runtime is decreased by nearly threefold. The most important aspect of these results is that the computational expense of optimized ECCKD+SPARTACUS is smaller than reference RRTMG+McICA: on this particular platform, using the more sophisticated solver and new ECCKD gas optics is faster than the operational ecRAD code by a factor of 1.87.

On the same platform, we also compared runtimes using a data set with a lower vertical resolution (60 levels), as similarly course vertical grids are likely

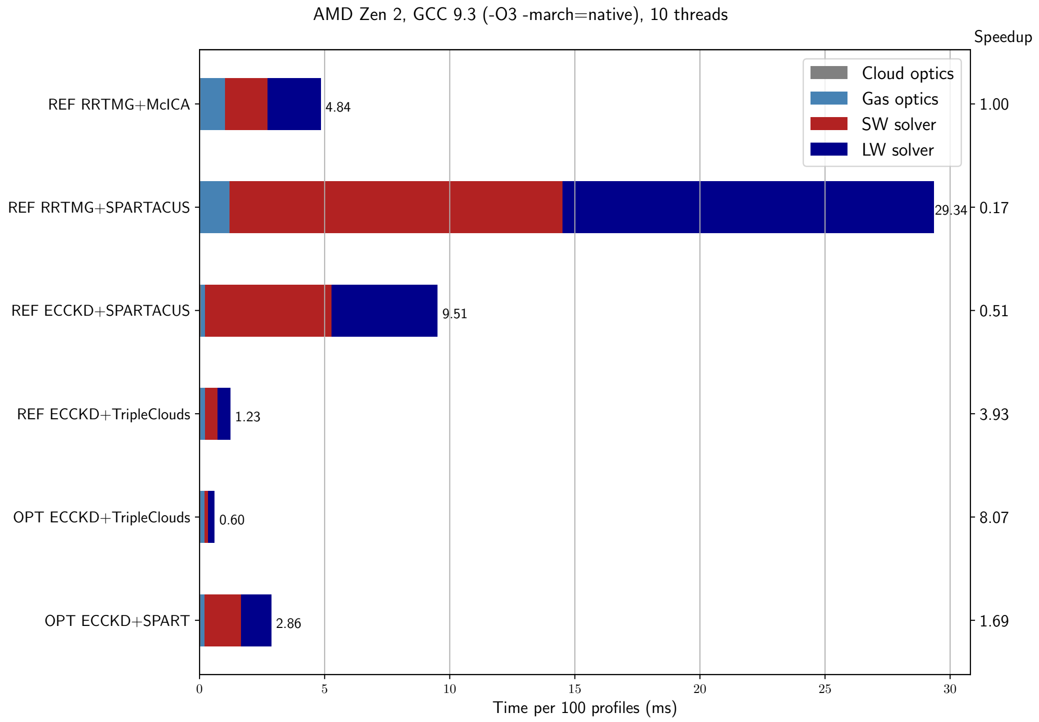


Figure 6.3: As in Figure 6.2, but using a different data set with lower vertical resolution (60 model levels) and no aerosols included.

to be used in global climate simulations. The impact of the optimization is similar, but the runtimes are reduced across the board as is expected. In many cases, the speed-up slightly exceeds the proportional decrease in the vertical dimensions; for instance, the time per profiles using optimized ECCKD+SPARTACUS is reduced by a factor of 2.6. This is expected, as increasing the vertical dimension can deteriorate cache use by increasing the sizes of many arrays, and the solver also includes many computations with loop dependencies in the vertical dimension. Using optimized ECCKD+TripleClouds, the fastest option in the refactored radiation scheme, the fluxes for 100 profiles can be computed in just 0.6 ms per 100 profiles.

References

Hager, G. and Wellein, G. (2010). *Introduction to high performance computing for scientists and engineers.* CRC Press.

Hogan, R. J. and Bozzo, A. (2018). A flexible and efficient radiation scheme for the ecmwf model. *Journal of Advances in Modeling Earth Systems*, 10(8):1990–2008.

Hogan, R. J., Schäfer, S. A., Klinger, C., Chiu, J. C., and Mayer, B. (2016). Representing 3-d cloud radiation effects in two-stream schemes: 2. matrix formulation and broadband evaluation. *Journal of Geophysical Research: Atmospheres*, 121(14):8583–8599.

Meador, W. and 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.

7

Discussion and conclusion

The research presented here has focused on the goal of improving the efficiency of radiative transfer computations in NWP and climate models. The initial plan of the PhD study was rooted firmly in the use of machine learning to emulate physical radiative transfer schemes. A concern early on was the suitability of NNs to essentially replace a well-understood set of radiative transfer equations: although NNs are known as universal approximators, radiation schemes have particular characteristics that make them less attractive to be replaced with an entirely statistical model. Particularly, the "transfer" side of radiative transfer, like fluid dynamics, is a problem that intuitively should perhaps not be solved using a purely statistical approach. One issue is the conservation of energy, which is of fundamental importance in NWP and particularly climate models. Another is that that feed-forward NNs do not structurally seem to capture the sequential aspect of radiative transfer computations.

With this in mind, the research was redirected towards solving a less complex, and more empirical problem with neural networks, which in radiation schemes is the computation of optical properties. This immediately avoids issues with energy conservation, for example, as even wrongly predicted atmospheric optical properties do not violate any conservation laws. The recently developed gas optics scheme RRTMGP was a promising target for emulation, as it is state-of-the-art, and because it represents so many minor greenhouse gases, also a relatively expensive component of the RTE+RRTMGP radiation scheme. The results obtained by emulating RRTMGP with NNs (Paper 1) achieved a high degree of accuracy and speed-up, especially when it was combined with a rewriting the dimension order of the radiative transfer solver to avoid a computational bottleneck, which would otherwise have reduced the relative speedup given by NNs. The total speedup for the computation of clear-sky fluxes was a factor of 2-3. In Paper 3, newly trained RRTMGP-NN models were implemented in the IFS weather model. It was demonstrated that the differences between the original RRTMGP scheme and its NN version (in terms of model climate) were small in comparison to the differences between the older RRTMG scheme and RRTMGP, and the former differences could be attributable to noise alone (in the CKDMIP evaluation, RRTMGP-NN and RRTMGP produce extremely similar results).

The refactoring done to achieve larger efficiency improvements in Paper 1 highlighted that for accelerating radiation computations, the use of ML is one among many possible approaches. In general, the performance of traditional physics code on CPUs can in many cases be radically improved using code refactoring techniques targeting better memory use and higher vectorization.

Newer HPC architectures such as GPU's and other accelerators represent another promising way of increasing the speed and reducing the energy cost of computations that are highly parallel and have a high throughput. Radiative transfer computations performed in thousands or even millions of columns in ever higher-resolution dynamical models typically meet both of these criteria, but this parallelism needs to be exposed by the programmer. The use of GPU's and machine learning as ways to reduce the runtimes or energy costs of sub-grid physics computations are of course not independent: because NNs run efficiently on GPUs, combining the two could offer the highest efficiency for radiative transfer computations, especially in terms of energy-to-solution metrics. However, the devil is in the details. For instance, the NNs used to parameterize gas optics in Papers 1-3 are small enough that they have relatively good performance on CPU's compared to GPU (Paper 1, supplementary results).

In addition, developing better physical parameterizations through e.g. new physical insight or carefully redesigned implementations of existing methods can in some cases achieve the clearest improvements in efficiency, but requires a high level of expertise and was not the subject of this research. A great example of this is ECCKD, which drastically reduces the number of spectral integrations, but based the evaluation done in CKDMIP, offers a similar level of accuracy as other correlated-*k* distribution schemes. The motivation for Paper 4, which is still in preparation, was to further improve the efficiency of the ecRAD scheme with ECCKD.

7.1 Which emulators for radiative transfer?

In Paper 3, different ways of emulating a shortwave radiation scheme were compared. The objective was to evaluate the trade off between speed and accuracy for different NN approaches. The paper made a novel contribution to the growing literature on NN emulation of radiative transfer by developing a method based on recurrent neural networks that structurally resemble physical radiative transfer computations. The RNN was compared to an feed-forward NN, were both were trained to predict fluxes to ensure energy conservation. Heating rates derived from the RNN predictions were shown to be very accurate, with errors similar in magnitude to clear-sky parameterization errors of correlated-k schemes. Errors produced with the FNN were an order of magnitude higher. On the other hand, the sequential nature of RNNs, which makes them more accurate for emulating radiative transfer than FNNs, also makes them slower. This demonstrates that machine learning methods offer no free lunch, and also suffer from a trade-off between accuracy and speed when used for radiation computations. Finally, the gas optics emulation is the most accurate approach, but the potential speedup is limited by the gas optics' share of the runtime of the total radiation scheme, which may not always be high.

7.2 Future perspectives

Many interesting directions for future research have emerged from this PhD study. The most obvious is the use of RNNs for emulating radiation schemes. Perhaps the most interesting use for ML is not only as a code acceleration tool, but to represent more complex physics than current parameterizations do. For instance, RNNs could be trained to emulate the SPARTACUS scheme, or other radiative transfer models that represent 3D radiative effects.

It could be argued that code optimization is an unexploited potential in current NWP and climate models. Refactoring existing codes for performance can take significant effort, but is likely to pay itself back in reduced runtimes and energy use. Future Earth System scientists should also be trained in high-performance computing at university. For accelerating radiation computations, the use of machine learning is perhaps a trendier topic than refactoring existing code, but in the present work it was shown that code refactorings can give meaningful efficiency improvements and do not sacrifice accuracy. It would also be interesting to compare the accuracy and speed offered by more accurate ML methods (RNN) to optimized, state-of-the-art physical radiation schemes. Comparing the speed achieved for optimized ecRAD scheme with ECCKD+TripleClouds (Paper 4) to the inference speed when using RNNs on the CPU in Paper 3, one arrives at similar values, although the comparison is hindered by many factors. To really start measuring the potential of ML to improve the efficiency of radiation computations in a more meaningful sense, it would be useful for the community to adopt common metrics and data sets, and ensure that comparisons are made to the state-of-the art in physical parameterizations.