Revisiting Machine Learning Approaches for Short- and Longwave Radiation Inference in Weather and Climate Models, Part I: Offline Performance

Guillaume Bertoli[1], Firat Ozdemir[2], Fernando Perez-Cruz[2,3], and Sebastian Schemm[1]

[1]Institute for Atmospheric and Climate Science, ETH Zurich, Zurich, Switzerland
[2]Swiss Data Science Center, ETH Zurich and EPFL, Zurich, Switzerland
[3]Computer Science Department, ETH Zurich, Zurich, Switzerland

Key Points:

- Physics-informed normalization and height-depending physics-informed penalization during training improve all tested ML architectures.
- Combining the above with a recurrent neural network outperforms U-Net,multilayer perceptron and random forest architectures.
- Atmospheric model top and day-night boundaries continue to challenge all tested architectures with the exception of a random forest.

Corresponding author: Guillaume Bertoli, guillaume.bertoli@env.ethz.ch

Abstract

As climate modellers prepare their code for kilometre-scale global simulations, the computationally demanding radiative transfer parameterization is a prime candidate for machine learning (ML) emulation. Because of the computational demands, many weather centres use a reduced spatial grid and reduced temporal frequency for radiative transfer calculations in their forecast models. This strategy is known to affect forecast quality, which further motivates the use of ML-based radiative transfer parameterizations. This paper contributes to the discussion on how to incorporate physical constraints into an ML-based radiative parameterization, and how different neural network (NN) designs and output normalisation affect prediction performance. A random forest (RF) is used as a baseline method, with the European Centre for Medium-Range Weather Forecasts (ECMWF) model ecRad, the operational radiation scheme in the Icosahedral Nonhydrostatic Weather and Climate Model (ICON), used for training. Surprisingly, the RF is not affected by the top-of-atmosphere (TOA) bias found in all NNs tested (e.g., MLP, CNN, UNet, RNN) in this and previously published studies. At lower atmospheric levels, the RF is able to compete with all NNs tested, but its memory requirements quickly become prohibitive. For a fixed memory size, most NNs outperform the RF except at TOA. For the best emulator, we use a recurrent neural network architecture which closely imitates the physical process it emulates. We additionally normalize the shortwave and longwave fluxes to reduce their dependence from the solar angle and surface temperature respectively. Finally, we train the model with an additional heating rates penalty in the loss function.

Plain Language Summary

Atmospheric radiation is an essential component of atmospheric modelling, which describes the amount of solar energy absorbed by the atmosphere and surface, and the thermal energy emitted as a response. The current radiation solver in the climate model named ICON is accurate but the complexity of the radiation process makes it computationally slow. Therefore the radiation solver cannot be called frequently in space and time by the model, which reduces the quality of the climate prediction. A possible approach to accelerate the computation of the radiation is to use machine learning methods. Machine learning methods can speed up the computation of the radiation substantially. However they are known to cause the climate predictions to drive away from a physically correct solution since they do not necessarily satisfy essential physical properties. In this paper we study neural networks, an increasingly popular deep learning approach. We explore various architectures, loss functions and output normalizations. We compare the results with a random forest emulation of radiation, which is easier to train than the neural network but as a prohibitive memory cost.

1 Introduction

The computation of atmospheric radiation is a central part of each Earth System Model (ESM). It models the solar energy absorbed by the Earth, the complex interactions between radiation and greenhouse gases, clouds and aerosols, scattering, and the energy radiated back as thermal (longwave) radiation. The operational radiation solver in the Icosahedral Nonhydrostatic Weather and Climate model (ICON) (Prill et al., 2020) is ecRad (Hogan & Bozzo, 2018), which is the new operational weather forecasting model of the Swiss (MeteoSwiss) and German weather services. EcRad is actively developed at European Centre for Medium-Range Weather Forecasts (ECMWF) where a GPU port is under development. The general outline of ecRad is that it first computes the gas, aerosols and clouds optics and passes those to a solver which predicts the atmospheric radiation fluxes based on which the driving model computes the fluxes convergence to obtain the corresponding heating rates. In ICON, the atmospheric radiation is operationally not

solved on the same spatial grid as the rest of the model. For computational reasons, the radiation fluxes are only computed on a coarser horizontal grid. Furthermore, the time interval between two calls of ecRad is large to further reduce the computational time. This is known to reduce the quality of the prediction (Hogan & Bozzo, 2018). Reducing the computational time required to predict the radiation fluxes would allow to solve the radiation with a smaller time step and on a finer spatial grid, which has the potential to improve the accuracy of the weather forecast. A promising approach to accelerate the computation of the radiation fluxes and to improve its energy efficiency is to use machine learning (ML) methods. There has been a wealth of research in recent years to replace physical parameterizations in weather and climate models with data-driven parameterizations (Brenowitz & Bretherton, 2019, 2018; Gentine et al., 2018; O'Gorman & Dwyer, 2018; Yuval et al., 2021; Kashinath et al., 2021) and in the following, we review recently published radiation emulating strategies before we outline the contribution by this study.

1.1 State of research in ML-based radiation parameterizations

The two central questions for data-driven radiative transfer parameterizations are which ML architecture to use and how to account for known physical relationships. In short, how to get the physics into the statistics? Two influential papers on machine learning-based parameterizations of atmospheric radiation, which are preludes to the above formulated questions, are Chevallier et al. (1998) and Krasnopolsky et al. (2005).

The prelude: Chevallier et al. (1998) and Chevallier et al. (2000), who extend the research started in Chéruy et al. (1996), emulate the ECMWF wideband scheme described in Morcrette (1991) and the line-by-line model described in Scott and Chedin (1981). They only consider the longwave fluxes. To increase the generalization capability of the emulator, the authors add several steps to the ML pipeline to enforce known physical relations. First, the emulator predicts (longwave) radiation fluxes but not the corresponding heating rates. The latter are instead computed based on the predicted fluxes. This strategy preserves the physical relation between the emulated fluxes and the heating rates. Then, to enforce cloud-radiation interactions, the emulator does not predict directly the fluxes. Instead it first predicts with one NN the radiation for a cloud-free atmosphere. Next the scheme computes the radiation for an atmosphere with a single blackbody cloud at a given height level. This computation is performed one time per atmospheric level, by varying the position of the blackbody cloud. The net fluxes are then a combination of the clear sky radiation and the radiation fluxes obtained for an atmosphere with a single blackbody cloud. The cost of these intermediate steps is a lower speedup of the machine learning parameterization.

Krasnopolsky et al. (2005), whose work is extended in Krasnopolsky et al. (2008) and Krasnopolsky et al. (2010), emulate radiation through purely data-driven parameterization. They do not decompose the problem into smaller subproblems but instead compute directly the final outputs, which allows a maximal speed up. Furthermore, the proposed method directly computes the heating rates and skips the emulation of the radiation fluxes. From a numerical point of view, this is attractive because such an approach does not require any additional derivation to calculate the heating rates from the radiation fluxes. However, when emulating the heating rates, they can only be compared against heating rates derived from the observed radiation fluxes (e.g., satellite data), making them a more suboptimal metric for validation. Further, as already stated, computing heating rates from the radiative fluxes guarantees physical consistency and radiative fluxes are required as inputs, for example, to the land model in an ESM.

A key question is thus to whether emulate fluxes, heating rates or both and how to ensure their consistency. The radiative fluxes can be observed by instruments, they serve as input to the land component of an ESM and are also relevant for impact mod-

elers, for example, to compute electricity production by solar panels. The disadvantage of emulating the radiative fluxes is the additional computational cost and numerical error that results from the required vertical derivative needed to obtain the corresponding heating rates that drive the evolution of atmospheric temperature. Even if the fluxes are predicted accurately, the heating rate error may be large if the vertical profiles of the fluxes are not smooth. In Krasnopolsky et al. (2005) the surface and top of atmosphere (TOA) fluxes are predicted by the ML emulation in addition to the heating rates. From the heating rates and net fluxes at the top or surface, one can recover the net fluxes at each atmospheric level. However, the individual contribution of upward and downward longwave and shortwave radiation fluxes cannot be recovered. In the next two sections, we first provide an overview of the various ML model architectures that were recently explored in the field of radiation emulation:

Fully-connected feedforward NNs: Fully-connected feedforward NNs are studied in Pal et al. (2019), Roh and Song (2020) and Belochitski and Krasnopolsky (2021). Pal et al. (2019) propose a radiation emulator based on fully connected feedforward NNs composed of three hidden layers for the Super-Parameterized Energy Exascale Earth System Model (SP-E3SM) and reports an error smaller than the internal variability of the climate model. Roh and Song (2020) emulate the radiation fluxes and the corresponding heating rates of the Korea Local Analysis and Prediction System (KLAPS) based on the single-layer feedforward NN following the scheme provided by Krasnopolsky (2014). They assess the quality of the emulation by comparing simulations where the radiation is computed at every time step using the machine learning emulation, against simulations where the original solver is used at larger time interval. Testing a similar computational burden by running emulator more frequent; the prediction of heating rates, cloud fraction, radiation fluxes, surface temperature and precipitation was shown to be more accurate for simulations where the emulation is run every time step (every 3 seconds) compared to simulations where the original parameterization is called every 20 time steps (every 60 seconds). In Meyer et al. (2022), the authors use feedforward NNs to emulate the 3D effects of clouds for the radiative transfer. They take as input the radiation fluxes computed by ecRad with a one dimensional cloud solver and as training target the difference between the fluxes computed by ecRad with a one dimensional cloud solver and a three dimensional cloud solver. This strategy substantially increases the speed at which fluxes are computed for the three dimensional cloud solver at the cost of an acceptable reduction in accuracy.

Convolutional and recurrent NNs: More complex deep learning architectures, such as convolutional NNs (CNNs) (LeCun et al., 1998) or recurrent NNs (RNNs) (Rumelhart et al., 1986), have also been recently explored for radiation parameterizations. In a feedforward CNN, fixed length kernel(s) are convolved over activations at a given layer as opposed to densely connecting each neuron with each neuron of the subsequent layer as in fully-connected feedforward NNs. RNNs on the other hand consist of an inner loop that reuses a set of neurons over a given dimension of input vectors, e.g., typically time-axis. In Liu et al. (2020), numerical experiments with CNNs exploiting the correlation between horizontally adjacent atmospheric columns are performed, but the authors report that CNNs reduce the computational speed substantially for a marginal increase in accuracy. In Lagerquist et al. (2021), the authors experiment with the UNet++ architecture developped in Zhou et al. (2020). The authors observ that the UNet++ architecture allows them to outperform existing fully-connected feedforward network parameterization, in particular the model developed in Krasnopolsky et al. (2010). Ukkonen (2022) employs RNNs to exploit the correlation between vertically stacked atmospheric levels. The design of this strategy is justified by the observation that the radiation fluxes at one height level result of the interaction of the radiation fluxes with, for example humidity, in the atmospheric levels above and below. An RNN approach, which can learn prediction as a function of previous atmospheric levels appears as a natural choice. In their work, the RNN predicts shortwave fluxes and derived heating rates more accurately

than the fully connected NNs at the cost of a smaller speed-up. The RNN experiences however large heating rate errors near the surface and model top. To avoid this issue, the authors suggest to normalize the output by dividing the shortwave fluxes at each height level by the TOA incoming radiation flux.

Decision trees: Finally, random forests (RF), and more generally tree approximation methods to predict the radiation fluxes, are - to our knowledge - rarely explored for radiation emulation. Belochitski et al. (2011) compare NNs, nearest neighbors approximation, regression trees, RFs and sparse occupancy trees. They conclude that although the tree approximations provide accurate results that compete with NNs, they require a large amount of memory compared to NN which make them difficult to use for parallel computing. Nevertheless, as observed in O'Gorman and Dwyer (2018), their stability and energy conservation properties make them good candidate ML methods within weather forecasting, where the need to generalisation is much less pressing than in longterm climate simulations where the ML model will receive data far outside its training space.

Including the physics into the statistics: In addition to the choice and design of the network architectures, another key strategy to build reliable and accurate weather and climate emulators is to incorporate physical knowledge into the data-driven radiation emulator. One way to do so is to design custom loss functions which penalize the NNs if they do not satisfy relevant physical relations. For example, in Lagerquist et al. (2021), the authors modify the loss function by increasing the penalty if large heating rates are not well predicted. In a similar spirit, Ukkonen (2022) adds a constraint to the objective function that penalizes errors in heating rates. Thus, both the radiation fluxes and the heating rates are incorporated in the loss function to ensure physical consistency at each pressure level. A second way is to build hybrid models which continue to use part of the original parameterization. Veerman et al. (2021) and Ukkonen et al. (2020) do not emulate the full radiation parameterization scheme but only the gas optics, i.e., the most expensive part of the physics-based radiation parameterization ecRad (Hogan & Bozzo, 2018), is emulated. Since the gas optics is less understood than the radiative transfer equation, its emulation is particularly well-suited for a data-driven parameterization while the remaining parts are computed by the physics-based radiative transfer model. It remains to be shown if hybrid models generalize better than loss-function constrained models, which makes them a relevant research topic.

1.2 Contributions of this paper

Based on the above review of the state of the art, we aim to first deliver a systematic review of the performance of different classes of ML methods (e.g. fully-connected, convolutional, recurrent networks and RFs) and discuss how physical knowledge can be incorporated in their training and change their performance. We investigate and discuss specific data preprocessing approaches and architectural design choices. For the systematic review we choose an idealized aquaplanet simulation for the training as it appears reasonable for such a comparison to perform it in a controlled and simple environment. In part one of this study, main focus is on the *offline* accuracy of the different methods, which refers to performance independent of a driving numerical model. In part two of this study, we will then investigate the *online* performance using the seamless weather and climate prediction model ICON.

2 Methods to emulate radiation

2.1 Framework and notations

In this paper, we study machine learning methods to emulate the radiation solver ecRad. The solver ecRad takes as inputs the temperature, the pressure, the cloud cover, the specific humidity, the specific cloud ice and liquid water content and the mixing ra-

tio of other gases and aerosols, at each atmospheric level of the model, in addition to the cosine of solar zenith angle, the surface pressure and temperature, the longwave emissivity and the albedos for chosen spectral bands. It then predicts the longwave and shortwave upward and downward fluxes at each atmospheric level. The ecRad solver has a modular architecture which allows one to change the gas, aerosol and cloud optics computation. We focus our research on the default optics computation used in the ICON climate model. In ICON, the usual plane parallel approximation is chosen for the computation of the radiation. When predicting the radiation for a given atmospheric column, we therefore omit the contribution of the features in neighboring columns. Mathematically, we represent ecRad as a function $f : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, where d_1 is the number of inputs and d_2 is the number of outputs. We construct a machine learning approximation $f_{ML} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ of f. Note that in practice, the machine learning approximation f_{ML} could use less or more inputs than the function f.

In this work, we consider two machine learning models: RFs and NNs. RFs are ensembles of decision trees. Each tree provides a rough estimate of the function f. The RF approximation is then given by the average of the different trees. A NN is a composition of simple non linear functions. Both methods are described in more details in section 2.2 and section 2.3. The neural networks optimization methodology is as follows. We consider a set of inputs x_i, $i = 1, \dots, N$ for which we compute the target $f(x_i)$ with ecRad. The NN model is then optimized to achieve these targets through an iterative process in order to minimize a given loss function. Typically, the loss function is defined as the mean squared error between the target $f(x_i)$ and predicted $f_{ML}(x_i)$. More terms can be added to the loss function to penalize the NN model for violating physical properties. Through minimizing for empirical risk, the goal is to achieve an approximation model f_{ML} that has a small error for all x in a sufficiently large subspace of the input space.

In this paper, our data are generated by an aquaplanet simulation performed by the ICON climate model, where the radiation fluxes are computed by the ecRad solver. We simulate one year of data with a physics time step interval of 3 minutes (and a dynamical core time step interval of 36 seconds) on a 80km spatial grid (ICON grid R02B05). We store samples with a frequency of three hours. For each stored atmospheric column, we therefore have access to input (in $\mathbb{R}^{d_1}$) and output variables (in $\mathbb{R}^{d_2}$) to optimize our ML emulator of ecRad. More details on the data set is given is section 3.

2.2 Neural networks

In this section, we describe the NN architectures and various loss functions we investigate in this paper.

Neural Networks Architectures

In this paper, we consider multilayers perceptrons (MLP), one dimensional convolutional neural networks (CNN), in particular UNet, and recurrent neural networks (RNN). We describe here the different architectures considered in this paper.

An MLP is a feedforward and fully connected neural network. An MLP f_{NN} is a composition of simple nonlinear functions $g_m : \mathbb{R}^{c_m} \to \mathbb{R}^{c_{m+1}}$

$$f_{NN}(x) = \left(\prod_{m=0}^{P} g_m \right)(x), \tag{1}$$

where $\prod$ represents composition of functions. The functions g_m are of the form

$$g_m(x) = \sigma_k(A_m x + B_m),$$

where $A_m \in \mathbb{R}^{c_{m+1} \times c_m}$ is a matrix, $B_m \in \mathbb{R}^{c_{m+1}}$ is a vector and $\sigma_m : \mathbb{R}^{c_{m+1}} \to \mathbb{R}^{c_{m+1}}$ is a typically nonlinear function, also called activation function. The number P is the number of hidden layers and the dimensions c_m for $m = 1, \ldots, P$ are the number of neurons in each hidden layer. The dimensions $c_0 = d_1$ and $c_{P+1} = d_2$ are the input and output dimensions of the NN. A standard choice for the activation functions σ_k is the rectified linear unit (ReLU) function:

$$\sigma(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

In this paper, all activation functions are ReLU functions. Note, that the standard choice for the last activation function σ_P is the identity, $\sigma_P(x) = x$ for all x. While our NNs also adopt this, we include a post-processing step via an additional ReLU function (unless mentioned otherwise) since the radiation fluxes are always positive.

CNN were developed in the context of image recognition. The idea is to replace fully connected layers with discrete convolution layers where only neighboring pixels are connected to a given layer. In our one dimensional context, this means that in (1), $g_m : \mathbb{R}^{H_m \times c_m} \to \mathbb{R}^{H_{m+1} \times c_{m+1}}$ is defined as

$$g_m(x) = \sigma_m(A_m * x + B_m),$$

where $A_m \in \mathbb{R}^{s \times c_m \times c_{m+1}}$ are matrices and $B_m \in \mathbb{R}^{c_{m+1}}$ a vectors. The dimension c_k is, in the CNN context, called the number of channels while H_m is the dimension of the mth latent space. The constant s is the size of the convolution. For $s = 3$, the discrete convolution is defined for all $j = 0, \ldots, c_{m+1}$ and $h = 1, \ldots, H_m$ by

$$(A_m * x + B_m)_{h,j} = \sum_{i=0}^{c_m} (a_{1,i,j} x_{h-1,i} + a_{2,i,j} x_{h,i} + a_{3,i,j} x_{h+1,j}) + b_j.$$

where, $x_{0,i} = 0$ and $x_{H_m+1,i} = 0$. Note that other options exist for the boundary points instead of zero padding like only applying the convolution for outputs at $h = 2, \ldots, H_m - 1$ and thus allowing the latent space dimension to diminish. In this work, we pad boundary values of the input vector to achieve smoother outputs, i.e., $x_{0,i} = x_{1,i}$ and $x_{H_m+1,i} = x_{H_m,i}$. The discrete convolution is defined similarly for higher values of s. To control the dimension of the latent space, average pooling layers are used. The average pooling reduces the latent space dimension by replacing pairs of neighboring levels by their average.

In this paper, we consider the Unet architecture. It is a specific kind of NN using convolutional layers developed initially for medical imagery. A UNet is composed of two parts. The UNet starts with the encoding part, where a succession of convolutional, pooling and fully connected layers are used to reduce progressively the latent space dimension. Then starts the decoding part where the encoding process is reversed by increasing progressively the latent space dimension to recover the output y. At each stage of a UNet decoder, latent features from the encoder with corresponding space dimension are stacked with the decoder input. This allows exploiting finer features extracted at the encoding stages, allowing for higher resolution predictions.

RNN is a neural network architecture developed for natural language processing. Assuming the input and the output have the same dimension, an RNN layer $g_m : \mathbb{R}^{d_0} \to \mathbb{R}^{d_0}$ is defined as follows. First, given the first element x_1 of the input vector x, a hidden state $g_m(x_1)$ for the first output element y_1 is computed. Depending on the exact RNN type, this can already be the approximation for $\hat{y_1}$ or there can be additional pathways within the RNN layer that estimate $\hat{y_1}$, e.g., long short term memory (LSTM) networks. At a next recurrent step, RNN approximates $\hat{y}_2$ given $g_m(x_1)$ and x_2. The process is iterated to predict $\hat{y}_{h+1}$ from $g_m(x_h)$ and x_{h+1}. It is worth noting that $g_m(x_h)$ can embed information from all inputs x_i for $i = 1, ..., h$. We hence obtain a vector $\hat{y}$

constructed from the vector x. In this work we use long short-term memory (LSTM) layers. Note that an RNN layer can also iterate the input vector in reverse. By stacking two independent LSTM layers, one starting from the TOA and the second one starting from the surface, we construct a bidirectional LSTM layer (BiLSTM) which allows the network to make predictions at each height level based on observations from the levels above and below.

Physics-informed normalization strategy for neural networks

Due to the nature of different units of observed features, we normalize all inputs for each height level to have zero mean and uni-variance, calculated based on the observations used for training. We refer to this as statistical normalization strategy and is common in ML training. Although this is the standard pre-processing also for the target features, recent works suggest feature specific means to normalize fluxes, which we refer to as physics-informed normalization strategy. In particular, Ukkonen (2022) normalizes each column of shortwave flux values using the value at the TOA. Since, shortwave fluxes can be roughly decomposed as the product between incoming flux, cosine of solar zenith angle ($\cos(\theta)$) and interaction with the atmosphere and surface, this corresponds to dividing shortwave flux values by $\cos(\theta)\cdot 1400$, where 1400 Wm^{-2} is an upper bound for the approximated incoming shortwave radiation. We apply the same strategy, which scales all shortwave flux values into the range of $[0, 1]$ and make them invariant to their horizontal positions. For values of $\cos(\theta)$ smaller than 10^{-4}, the predictions are swapped with 0 at each height level for both shortwave up and down.

For the longwave fluxes there exists no simple decomposition because the atmosphere itself emits in the longwave at each height level. However from the Stefan-Boltzmann law for the emission of a black body, we know that the surface emission in the longwave is bounded by $T_s^4 \cdot \sigma$, where T_s is the surface temperature, σ is the Stefan-Boltzmann constant ($\approx 5.67 \cdot 10^{-8} Wm^{-2}K^{-4}$). We therefore scale the target longwave fluxes by $T_s^4 \cdot \sigma$. Note that for simulations with topography, it could be advantageous to divide by $T_s^4 \cdot \sigma \cdot \epsilon_s$ instead where ϵ_s is the surface emissivity. After normalization, all target features are scaled to the range of $[0, 1]$. Accordingly, all NNs trained with this normalization strategy have sigmoid layer as their final activation function as opposed to ReLU.

Physics-constrained loss function

We describe here the loss functions that we consider in this paper. A paired training set $X_{\text{tr}} = \{x_k, f(x_k)\}$ is first created. A loss function $\mathcal{L}$ of the form

$$\mathcal{L}(X_{\text{tr}}) = \frac{1}{K} \sum_{k=1}^{K} \left\| f_{NN}(x_k) - f(x_k) \right\|_2^2 \tag{2}$$

is then computed iteratively for mini-batches of size K for a random subset drawn from the training set. The parameters of the NN are updated using a gradient-based optimizer for minimizing $\mathcal{L}$. This process is repeated until $\mathcal{L}$ is sufficiently small, e.g. ML model has converged.

In climate simulations, there may be trends and shifts of the data, as is the case for climate warming. Those trends and shifts could make ML models less accurate over time as the new data move away from the training set. To mitigate the reduction in accuracy of the NN over time, additional terms can be added to the loss function (2) to account for scientific prior knowledge about the observation space. For example, the radiation fluxes play a central role in the energy balance for atmospheric columns. One can thus add a new term in the loss function to better guide the optimization of the NN parameters by penalizing flux predictions that do not respect the energy balance equation.

The time evolution of the energy in an atmospheric column is described by the following equation (Kato et al., 2016):

$$\begin{aligned} &\frac{1}{g}\frac{\partial}{\partial t}\int_0^{p_s}(c_pT+\Phi_s+k+Lq)\,\mathrm{d}p \\ &+\frac{1}{g}\nabla_p\cdot\int_0^{p_s}\mathbf{U}\,(c_pT+\Phi+k+Lq)\,\mathrm{d}p \\ &\quad=(R_t-R_s)-F_{sh}-F_{lh}, \end{aligned} \tag{3}$$

with the following variables: gravitational acceleration g, pressure p, pressure at surface p_s, specific heat of air at constant pressure c_p, temperature T, geopotential Φ, geopotential at the surface Φ_s, kinetic energy k, horizontal wind vector $\mathbf{U}$, the net radiative flux at the top of atmosphere R_t, the net radiative flux at the surface R_s (both shortwave and longwave fluxes contribute to R_t and R_s), latent heat of vaporization L, specific humidity q, and surface sensible and latent heat fluxes F_{sh} and F_{lh}, respectively. From (3), we observe that in addition to exchanges with neighbouring columns, the energy in a column depends on precipitation, the heat exchange with the surface and the air above, and on the amount of shortwave and longwave fluxes absorbed by the atmosphere. The net irradiance, that is the amount of energy per square meter absorbed by the atmospheric column, $I := R_t - R_s$, is thus of particular importance since it plays a central role in the energy balance of an atmospheric column. If the net irradiance I is not predicted correctly, the climate model may, for example, compensate with an increase or decrease in precipitation, which could lead to a significant climate drift and hence a poor climate prediction.

A first idea would be to add an additional penalty term to the loss function (2) of the NN to increase the accuracy of the net irradiance I_{net} prediction:

$$\mathcal{L}_I(X_{\text{tr}}) = \frac{1}{K}\sum_{k=1}^{K}\|f_{NN}(x_k)-f(x_k)\|_2^2+\lambda\frac{1}{K}\sum_{k=1}^{K}\left(I_k-\hat{I}_k\right)^2, \tag{4}$$

where $\lambda \geq 0$ is the weight of the new irradiance penalty, where K denotes the number of data samples in the mini-batch, and where $I_k \in \mathrm{R}$ and $\hat{I}_k \in \mathrm{R}$ are the exact and approximated net irradiance for the k-th training sample. The net irradiance term in (4) only affects the surface and top height levels, and in the adverse case the NN minimizes the penalty by adding at the surface and top levels radiative fluxes to overcompensate for potentially inaccurate predictions in the middle of the atmosphere. This results in large heating rates at the top and bottom for a given column.

An alternative to the loss function (4) is to penalize the NN if the energy absorbed at each height level is not well predicted. For example, the shortwave energy absorbed at height level h, where $h = 0$ is the top of atmosphere, is given by

$$E_h^{sw} = f_{h-1}^{sw} - f_h^{sw},$$

where f^{sw} is the net shortwave radiation at height level h. The absorbed energy term E_h^{sw} is directly related to the shortwave heating rates. Indeed, the heating rate equation for shortwave at height level h is defined by,

$$\mathrm{HR}_h^{sw} = -\frac{g}{c_p}\frac{f_{h-1}^{sw}-f_h^{sw}}{p_{h-1}-p_h} \approx -\frac{g}{c_p}\frac{\partial f^{sw}(p_h)}{\partial h}. \tag{5}$$

The longwave energy absorbed by level h and longwave heating rates are defined similarly. We hence consider the following loss function for $\lambda \geq 0$:

$$\mathcal{L}_{HR}(X_{\text{tr}}) = \frac{1}{K}\sum_{k=1}^{K}\|f_{NN}(x_k)-f(x_k)\|_2^2+\frac{1}{K}\sum_{k=1}^{K}\frac{1}{H}\sum_{h=1}^{H}\lambda(h)\left\|\mathrm{E}_{k,h}-\hat{\mathrm{E}}_{k,h}\right\|_2^2, \tag{6}$$

where H is the number of height levels per columns and $E_{k,h}$, $\hat{E}_{k,h}$ are the exact and approximated energy absorbed by the sample k at height level h, computed for both shortwave and longwave. Note that we allow here the weight $\lambda(h)$ to depend on the height level h.

2.3 Random forest

In this section, we discuss the emulation of ecRad using RF. The RF model will serve as the baseline emulator. An RF is an ensemble method based on decision trees. Each tree is constructed as follows. For a given tree, we construct a specific training set constructed by bootstrapping the main training set, i.e. random elements of the training set are picked with possible repetitions. A random subset of the input features of size $\sqrt{d_0}$ is then picked, where d_0 is the input space dimension. Amongst this feature subset, the feature n_1 and the associated scalar α_1 are picked such that n_1 and α_1 give the best way to separate the input space into the two parts $HS_{1,<} = \{x \in \mathbb{R}^{d_0}; x_{n_1} \leq \alpha_1\}$ and $HS_{1,>} = \{x \in \mathbb{R}^{d_0}; x_{n_1} > \alpha_1\}$. To evaluate the quality of the cut, the output average of all vectors from the bootstrapped training set belonging to $HS_{1,<}$ and $HS_{1,>}$ is computed. This average value is the output prediction for all vector in $HS_{1,<}$ and $HS_{1,>}$ respectively. From there, the MAE of the predictions is computed. The division of the input space continues as follows. A random subset of the input features space of size $\sqrt{d_0}$ is picked. Then the feature n_2, the scalar α_2 and the subspace amongst $HS_{1,<}$ and $HS_{2,>}$ that reduces the MAE the most amongst all possible way of cutting $HS_{1,*}$ along the hyperplane $\{x \in HS_{1,*} | x_{n_2} = \alpha_2\}$ is picked. The procedure continues until all subspaces contain sufficiently few elements, in this case at most 0.01% of the training set size. Note that subspaces which contain sufficiently few elements are no longer eligible for a cut. The process is repeated until 10 different trees are constructed. The random forest prediction is given by the average prediction of all trees in the forest. The random forest is hence a piecewise constant function. Another distinctive property of RFs is that they never predict values larger or smaller than what was observed in the training set. This will prove to be an advantage for the prediction of the fluxes at the upper levels of the atmosphere where the fluxes vary less due to the absence of clouds and humidity. At the same time, this property of the RF prevents it from generalizing well if larger or smaller values of the fluxes appear in the test set due for example to an increase in the global temperature. The same output normalization as the one introduced in Section 2.2 for the neural networks is used. The inputs are not normalized since RF are invariant by linear transformations of the input features.

2.4 Specific model architectures

Random forest

Each RF is composed of 10 trees. The size of the RF is constrained by imposing a minimum leaf equal to 10^{-2}% of the training set size. This results in an RF with memory footprint comparable to the NNs we consider. Such a constraint is necessary to prevent computationally prohibitive RF parameterizations, despite their improved predictive performance. From a memory consumption viewpoint, NN are more efficient compared to RFs – more details are provided in the result section (see Figure 5). Two separate RFs are constructed; one to predict the shortwave fluxes and one for the longwave fluxes. We normalize the outputs as described in Section 2.2.

Neural networks

For predicting both the shortwave and longwave upward and downward fluxes, we consider several NN architectures. The trained models predict all four target variables at all height levels and the models are trained for shortwave and longwave radiation independently. We adopt a notation to depict models with loss components consisting of

(i) only squared error as $()^2$, (ii) squared error in addition with height independent heating rate constraints as $()^{\partial T}$; (iii) squared error in addition with height dependent heating rate constraints as $()^{\partial T(h)}$ (iv) models with physics-informed output normalization $()_{norm}$:

- *MLP*^{2}: MLP emulating radiative fluxes with standard squared loss function:
  The loss function of this NN is given by Eq. (2). We provide a scheme of our MLP architecture in Figure 1. First a different set of embeddings for both surface features as well as each height of height-dependent features (e.g., humidity) are extracted using different MLPs, each with two hidden layers of 128 and 256 nodes. Subsequently, the embeddings computed at each height level ($H = 60$) are flattened to have a size of $256 \times 60 = 15360$, which are later concatenated with the embeddings of the surface variables, creating a $15360+256 = 15616$ dimensional vector. Then another MLP with three hidden layers of 1024 nodes each is applied, finalized by another fully connected layer of size 240 which is then reshaped to 60×4 (full column of each target variable).
- *MLP*$^{\partial T}$: MLP with additional level-wise heating rate penalty:
  The loss function of this NN is given by Eq. (6) with $\lambda_h = 1$ for each height level h. Other details are identical to MLP^2.
- *MLP*$^2_{norm}$: MLP with output normalization and squared loss:
  This MLP is identical to MLP^2 except that the output are normalized. Employed normalization approach is explained in Section 2 *Physics-informed normalization strategy for neural networks.*
- *UNet*^{2}: UNet with squared loss:
  We adopt the architectural scheme of UNet, shown in Figure 2. Namely, we first broadcast surface features to match the same height axis of height dependent features and concatenate them with the height dependent features. We then apply a 1D UNet along height axis, starting with 64 feature channels and convolutional kernels of size 3. We use border value padding to preserve height length following convolutional operators. To account for the number of height levels ($H = 60$), we coarsen the height axis 4 times using maxpooling with sizes of 2, 3, 5, 2, respectively. We use attention gates (Oktay et al., 2018) at skip connections. The loss function of this NN is given by Eq. (2).
- *UNet*$^{\partial T(h)}$: UNet with additional level-wise heating rate penalty:
  The loss function of this NN is given by Eq. (6) with $\lambda(h)$ equal to

$$\lambda(h) = \exp\left(\frac{\ln(1000) - 1}{H - 1} \cdot (H - 1 - h) + 1\right), \tag{7}$$

  where $h = 0$ is the TOA and $h = H - 1$ is the height level closest to the surface. The weight λ is then equal to 1 at the surface and smoothly increases to 1000 at the TOA. The motivation for a height dependent weight of the heating rates penalty stems from the observation that the NNs perform weaker near the TOA.
- *UNet*$^2_{norm}$: UNet with squared loss and output normalization:
  This UNet is identical to UNet^2 except that the outputs are normalized similarly to MLP$^2_{norm}$.
- *RNN*$^2_{norm}$*: RNN with standard squared loss and output normalization:*
  The loss function of this NN is given in Eq. (2). As shown in Figure 3, we use bidirectional (Bi-) LSTMs as the RNN cell type. Similar to UNet, we first broadcast surface features to match height axis of height dependent features and concatenate them. This is followed by an independent MLP at each height level with two hidden layers of 128 and 256 nodes. MLP outputs are then concatenated along

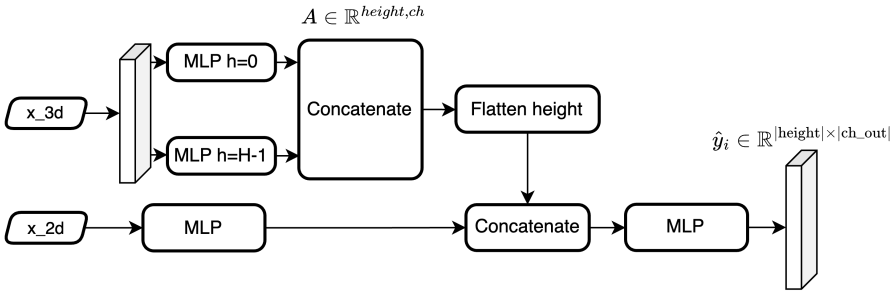

Figure 1: Schematic of the MLP used in this work. x_3d and x_2d correspond to 3d and 2d inputs described in Table 1.

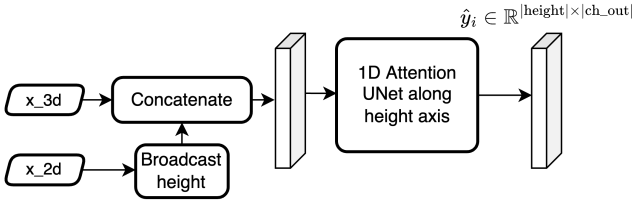

Figure 2: Schematic of the UNet used in this work. x_3d and x_2d correspond to 3d and 2d inputs described in Table 1.

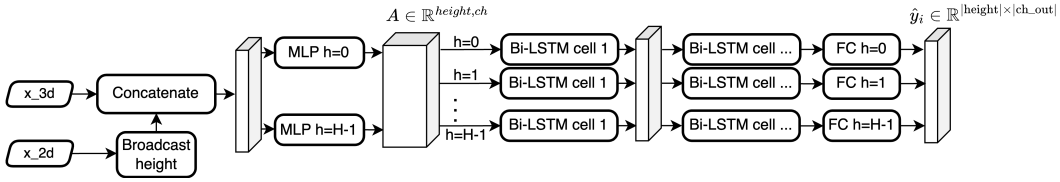

Figure 3: Schematic of the RNN used in this work. x_3d and x_2d correspond to 3d and 2d inputs described in Table 1.

height axis once again. We then apply three Bi-LSTM cells, each with 1024 channels, along the height axis. A fully connected layer at each height then maps the embeddings onto 4 channels.

- *$RNN^{\partial T(h)}$: RNN with additional level-wise heating rates penalty:*
  The loss function of this NN is given by Eq. (6) with λ_h given by Equation 7. All other details remain identical to RNN^2.
- *$RNN^{\partial T(h)}_{norm}$: RNN with additional level-wise heating rates penalty and output normalization:*
  This RNN is similar to $RNN^{\partial T(h)}$, however with output normalization similar to MLP^2_{norm}.

3 Data

In this work, we focus on aquaplanet simulations. We assume the mixing ratio of all gases to be constant except for the water vapor. Furthermore, we do not consider any aerosols. There are neither topography nor seasonality in our simulations. The sun al-

Inputs		Outputs
2d	3d	3d
surface temperature	temperature	shortwave down
surface pressure	pressure	shortwave up
specific humidity at surface	specific humidity	longwave down
cosine of solar zenith angle	cloud cover	longwave up
direct albedo, near infrared	water content	
diffuse albedo, near infrared	ice content	
direct albedo, UV-visible		
diffuse albedo, UV-visible		

Table 1: Inputs and outputs for the machine learning emulation. The 3d variables are stored for 60 atmospheric levels.

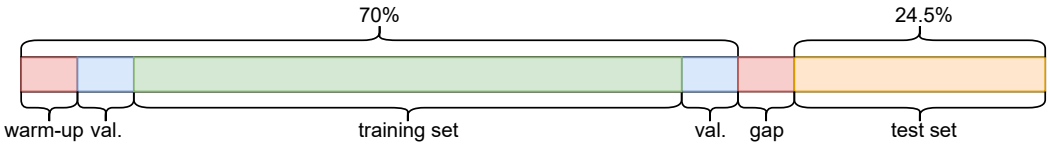

Figure 4: Data split for the 12 month aquaplanet. Warm-up, gap, and each block of validation sets (val.) are 20 days. Warm-up and gap are not used.

ways faces the equator. The simulation is run on the ICON grid R02B05 with a grid spacing of approximately 80 km. The ICON grid is constructed as follows. The sphere is first approximated with an icosahedron. Each vertex of each twenty triangle is divided into 2 such that we obtain in total 120 triangles. Finally, the procedure iteratively divides each vertex in two 5 times and we obtain finally 81′920 triangles. The NN and RF are trained on this icosahedrical grid. We run the ICON simulation with 60 atmospheric levels. The model time step is 180 seconds and we store the data every 3 hours. The simulation runs for one year with a 360 days calendar. We hence have 2′880 stored time steps.

The stored input and output features are given in Table 1. We have in total $8+6\times 60 = 368$ input variables and $4\times 60 = 240$ output variables. We dedicate the first 70% of the data to be used throughout training of the emulator and the last 30% to test and report the accuracy of the emulator. The first 20 days of the training set are removed to account for warming up period of ICON at the start of the simulation. The first 20 days of the test set are removed to ensure a gap between the train and test data. This ensures that the test data set is slightly out of distribution. The days 20 to 39 and the last 20 days of the training set are omitted from training and are used as a validation set. The aforementioned data split is summarized in Figure 4. After training NNs for a fixed number of steps, the validation set score is used to pick the training step with optimal NN parameters (e.g., early stopping criteria). In total, this yields a training set with 1'534 time-steps (∼192 days) and a validation set with 321 time-steps (∼40 days).

In ICON, the fluxes are given at half levels $(\frac{1}{2},\dots,60+\frac{1}{2})$ and the heating rates at full levels $(1,\dots,60)$. The flux f_h at atmospheric level h is at the interface between the level h and the level $h-1$. There is one more half level than full levels because each full level needs to be enclosed by two half levels. The half level $60+\frac{1}{2}$ corresponding to $h=60$ is the surface and the half level $\frac{1}{2}$ corresponding to $h=1$ is the model top of atmosphere.

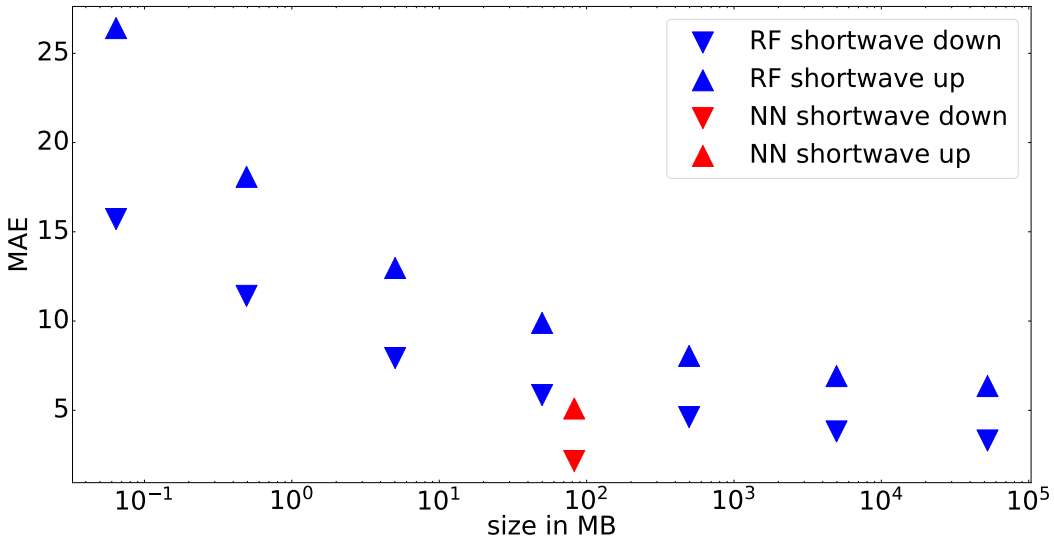

Figure 5: Size of the random forest in megabytes versus its MAE.

4 Results: Radiation emulation

Evaluation metrics: We evaluate the machine learning emulators on the test set using mean absolute error (MAE). At each time point $t \in \{1, \ldots, 321\}$, for each atmospheric column $c \in \{1, \ldots, 81920\}$ and at each height level $h \in \{1, \ldots, 60\}$, we have ground truth flux values computed by ecRad and predicted flux values computed by our proposed methods. Aggregating MAE over different pairs of variables allows us to observe different performance properties such as over time, horizontal space, and vertical space.

4.1 Random Forest

In general, RF model achieves the worst performance among the compared models for fluxes prediction (see Figures 6, 7 and 8). It outperforms, however, all compared NNs for the shortwave downward prediction near the top levels. The superior performance of RF near the TOA can be also observed for calculated shortwave heating rates. The success of RF near the TOA could be attributed to (i) the fact that RFs have a desirable property of being invariant to different scales of target variables as well as (ii) their property of averaging multiple decision trees that overfit to training data for their predictions. This implies that the smoothly varying vertical profile observed in training data directly reflects to predictions of the RF for the test data.

The random forest error: As our baseline RF model, we construct two RFs, one to predict the shortwave fluxes and one for the longwave fluxes. The RF model is constrained to a minimum leaf size of 0.01% of the training set. In our experiments, this resulted in an RF with a memory footprint of about 142MB. In Figure 5, we compare the MAE against the memory size of the RF responsible of computing the shortwave fluxes. As a reference, we also include MLP^2 in the plot. We observe that the accuracy of the RF can get close to the accuracy of NNs when its complexity increases. However the size of the RF quickly becomes too large to be of practical use. We observe that even for an RF of size close to 100GB, the MLP^2 remains more accurate. The random forest outputs are normalized as explained in 2.3 This improves the accuracy at no additional cost (see Table 2).

Random forest MAE	Without normalization	With normalization
Shortwave down	6.81 Wm^{-2}	4.61 Wm^{-2}
Shortwave up	9.09 Wm^{-2}	8.06 Wm^{-2}
Longwave down	5.22 Wm^{-2}	5.11 Wm^{-2}
Longwave up	5.52 Wm^{-2}	5.32 Wm^{-2}

Table 2: Effect of normalization on the random forest error.

4.2 Neural networks

We discuss the performance of three NN architectures, MLP, UNet and RNN described in Section 2.4. For each architecture, we investigate the effect of the output normalization described in Section 2.4 and the effect of the physics informed loss function (6) on the accuracy.

4.2.1 MLP

In Figure 6, we show the error of the MLPs described in Section 2 for the fluxes and heating rates predictions. For downward directed fluxes, the error of all the MLPs (and also UNets and RNNs, see Figures 7 and 8) tends to increase towards the surface with peak error values at the cloud bottom height level typically located at around 1 km altitude. For upward directed fluxes, the MAE tends to increase with altitude and peak values are reached at the TOA, although the error exhibits its strongest increase in the 1–4 km levels, while it remains constant above. The error hence increases in the direction of the fluxes. Because prediction from one height level do not affect the next height level, the increase is not an accumulation of errors into the fluxes direction. The error increases in the fluxes direction because as the fluxes cross height levels, they interact with atmospheric constituents which thus increases the complexity of the prediction.

For the downward longwave fluxes and the longwave heating rates prediction, the MLP has an error jump around 18km (MLP^2, green dashed line in Figure 6). For the heating rates, the error jump is one order of magnitude large. It may be caused by a numerical discontinuity in the longwave downward prediction at that height. At the TOA, the MLP^2 is significantly less accurate than the RF for the shortwave downward fluxes prediction.

When trained with an additional heating rates penalty ($\text{MLP}^{\partial T}$, blue dotted line in Figure 6), an error jump appears for the shortwave downward fluxes, the longwave upward fluxes and shortwave heating rates around 10km height. The longwave error jump already present for the MLP^2 appears at 10km height instead of 18km. Overall, the loss function (6) does not improve the accuracy of the MLP except for the shortwave heating rates above 15km. Furthermore, it adds sudden error jump that are absent for the square loss function (2). We've tested two additional loss functions that are not shown in Figure 6. We first considered a height dependent heating rates penalty similar to $\text{UNet}^{\partial T(h)}$. With this loss functions, the MLP becomes inaccurate at all heights for both fluxes and heating rates (see Appendix Appendix B). We also considered the loss function 4. For this loss, the MLP learns to add energy at the top and bottom to satisfy the new penalty which significantly degrades the accuracy of the solution at those heights (see Appendix Appendix B). For those reasons, we do not discuss those loss functions further.

The output normalization increases the accuracy of the model at all heights except for the shortwave heating rates below 4km height where the accuracy is slightly reduced (MLP^2_{norm}, red line in Figure 6). Furthermore the error jumps that we observe for MLP^2

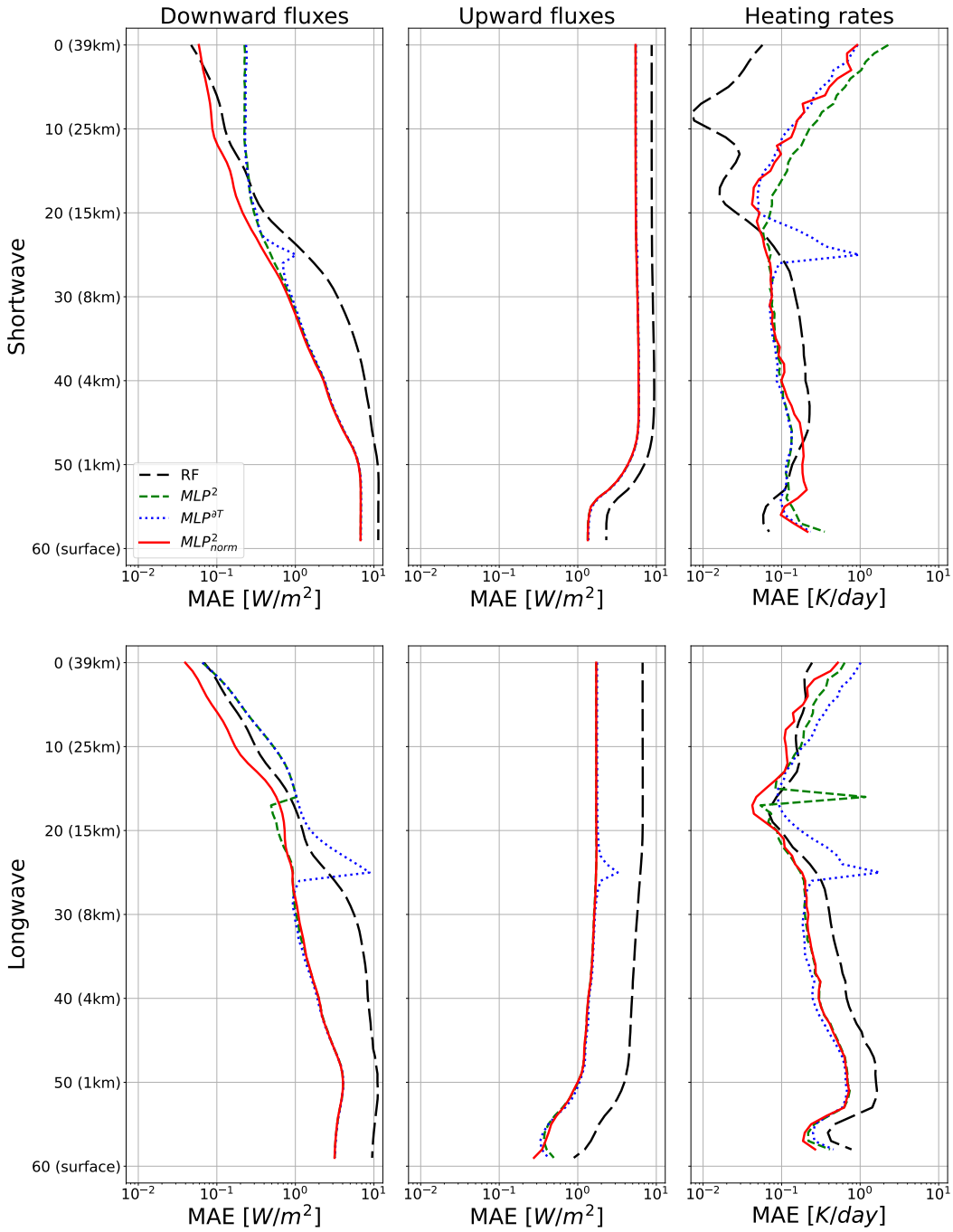

Figure 6: *MAE of the MLPs and of the RF emulator for the shortwave and longwave downward fluxes, upward fluxes and heating rates. Legend: RF; random forest, MLP^2; MLP trained with squared error loss, MLP^2_{norm}; MLP^2 with normalized output, $MLP^{\partial T}$; MLP^2 with an additional penalty for the inferred heating rates. The models are described in Section 2.4.*

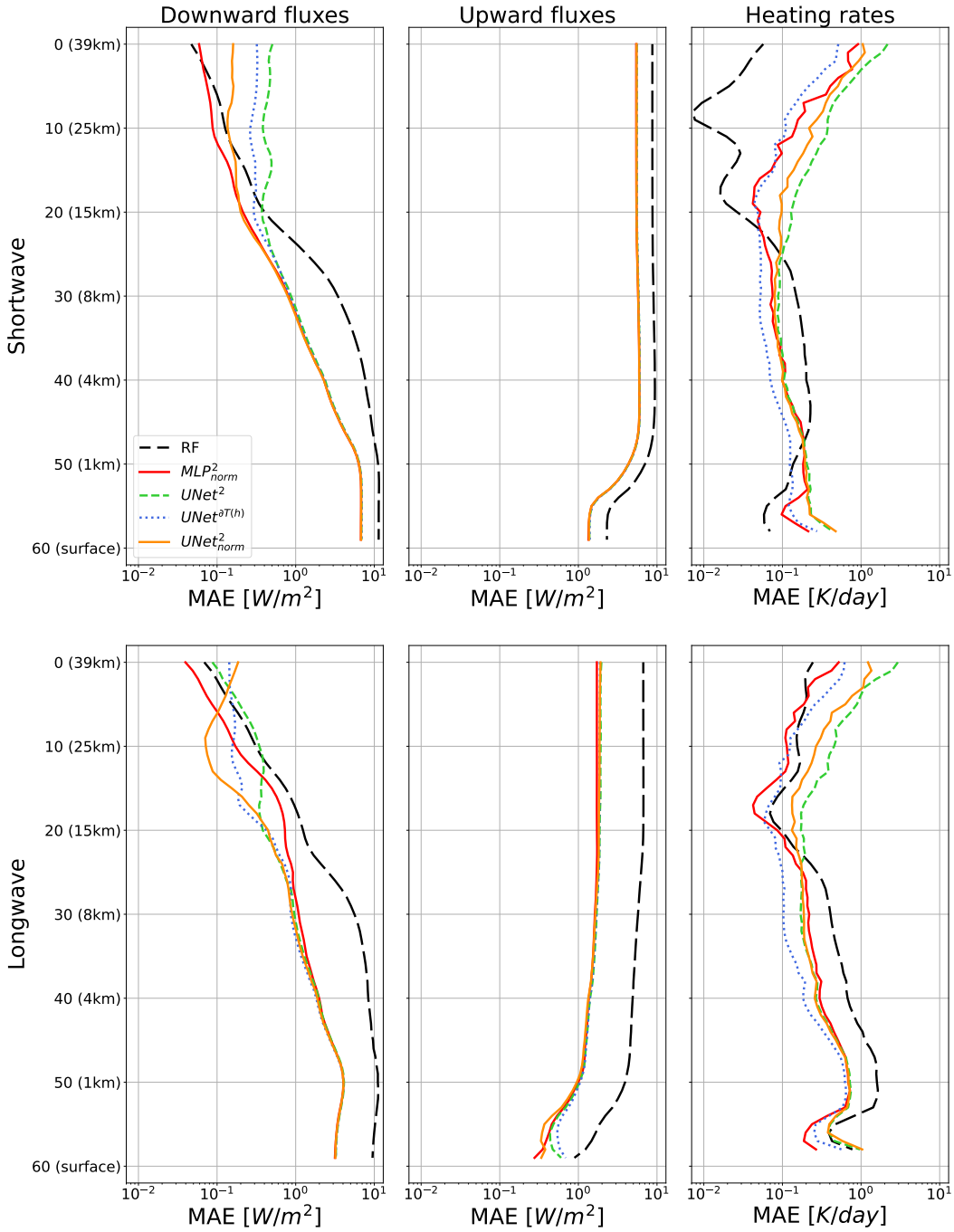

Figure 7: *MAE of the UNets and of the RF emulator for the shortwave and longwave downward fluxes, upward fluxes and heating rates.* MLP^2_{norm} *is included as a reference. Legend: RF; random forest,* MLP^2_{norm}*; MLP trained with squared error loss and normalized output,* $UNet^2$*; UNet trained with squared error loss,* $UNet^2_{norm}$*;* $UNet^2$ *with normalized output and* $UNet^{\partial T(h)}$*;* $UNet^2$ *trained with an additional height dependent heating rates penalty. The models are described in Section 2.4.*

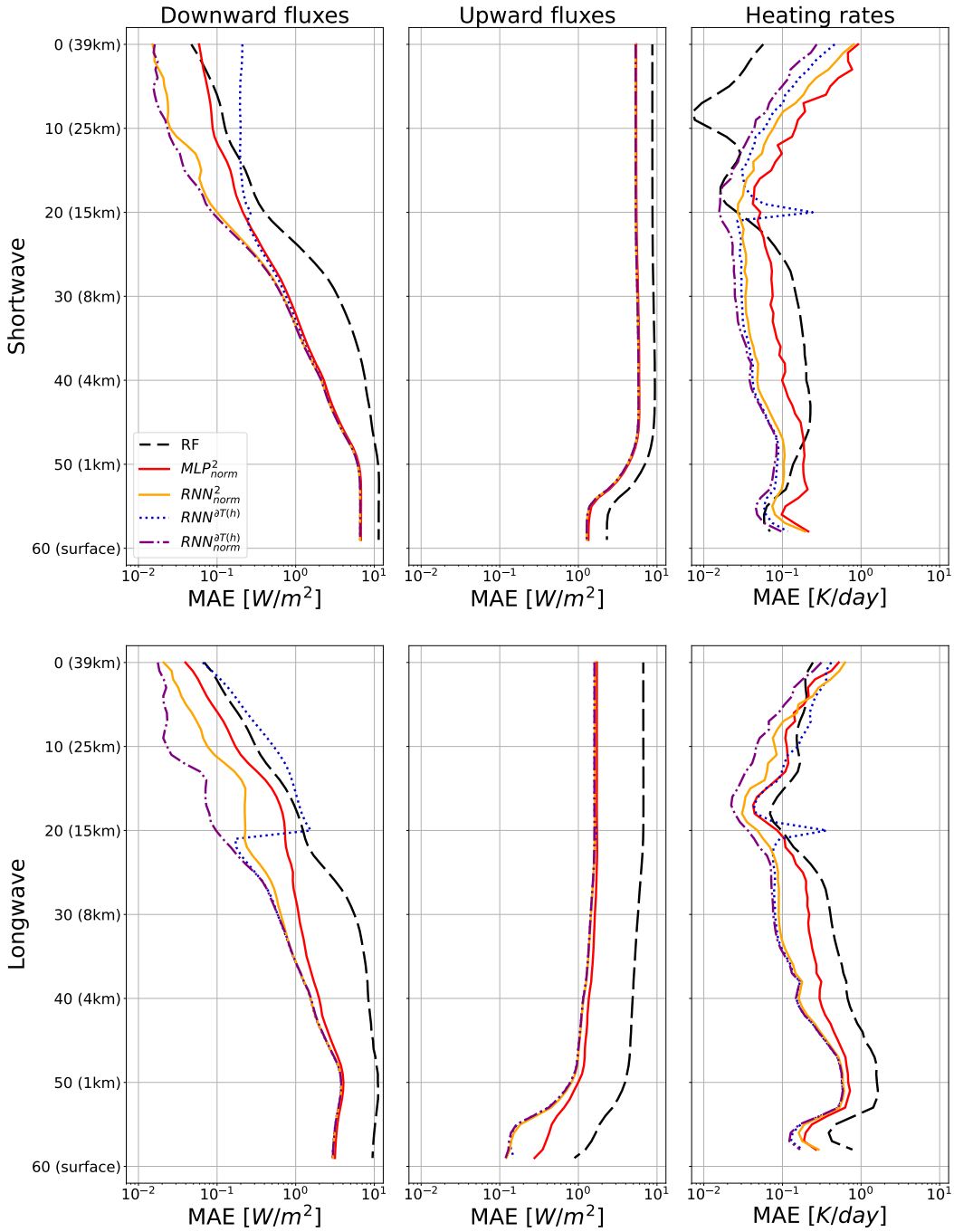

Figure 8: *MAE of the RNNs and of the RF emulator for the shortwave and longwave downward fluxes, upward fluxes and heating rates. MLP* MLP^2_{norm} *is included as a reference. Legend: RF; random forest,* MLP^2_{norm}*; MLP trained with squared error loss and normalized output,* RNN^2_{norm}*; RNN trained with squared error loss and output normalization,* $RNN^{\partial T(h)}$*; RNN trained with an additional height dependent heating rates penalty,* $RNN^{\partial T(h)}_{norm}$*;* $RNN^{\partial T(h)}$ *with output normalization. The models are described in Section 2.4.)*

around 18km disappears. For the shortwave downward fluxes, the MLP^2_{norm} becomes close to the RF error at the TOA.

For shortwave heating rates, the MLPs are outperformed by the RF above 15km by a large margin. This is likely because the RF predicts fluxes profiles that are smooth with height, while the NNs do not. The notable increase of the prediction error at the TOA is observed for all NNs and also reported in previous studies (Lagerquist et al., 2021; Ukkonen, 2022). For the derived longwave heating rates, the MLP is more accurate than the RF at most levels and especially in the troposphere. At the TOA however, the prediction error increases and the MLP is less accurate compared to the RF. As a comparison with the next NN architecture, we draw the MLP^2_{norm} error in Figures 7 and 8.

4.2.2 UNet

In Figure 7, we investigate the UNet architecture. We observe that the MLP^2_{norm} outperforms the $UNet^2$ (dashed green line in Figure 7) for the fluxes and heating rates predictions except for the longwave downward fluxes between 4km and 20km. The error difference is particularly large at the upper layers for the downward fluxes and heating rates. The $UNet^2$ doesn't have error peaks similar to the ones observed for the MLP^2 and $MLP^{\partial T}$.

When training the UNet with an additional heating rates penalty ($UNet^{\partial T(h)}$, blue dotted line in Figure 7), the model performance increases substantially for the heating rates prediction. Note that we consider here a heating rates penalty with height dependent weights (larger weights towards TOA). With this new penalty, $UNet^{\partial T(h)}$ outperforms MLP^2_{norm} at most heights for the heating rates predictions except at the top for the longwave. For the fluxes, the additional penalty also improves the accuracy for the downward fluxes at the upper layers except near the TOA for the longwave. Furthermore, contrary to what was observed for the $MLP^{\partial T}$, the additional penalty does not introduce error jumps.

The output normalization also increases the accuracy of the UNet ($UNet^2_{norm}$, orange line in Figure 7). In particular, between 15km and 25 km, the $UNet^2_{norm}$ is significantly more accurate than the $UNet^2$. Above 25km longwave downward flux error of the $UNet^2_{norm}$ starts to increase and it becomes the least accurate among other compared UNets at the TOA. The accuracy improvement from the output normalization is less important than the one obtained when adding a heating rates term in the loss function.

4.2.3 RNN

In Figure 8, we investigate the RNNs described in Section 2. The model RNN^2_{norm} (orange line in Figure 8) is everywhere more accurate than the MLP^2_{norm} except near the TOA for the longwave heating rates prediction.

If the RNN is trained with an additional heating rates penalty ($RNN^{\partial T(h)}$, blue dotted line in Figure 8) but no output normalization, error peaks appear at 15km height for the downward fluxes and heating rates prediction. Note that these error jumps are not at the same height as the ones observed for MLP^2 and $MLP^{\partial T}$

If we both normalize the outputs and trained the RNN with height dependent heating rates ($RNN^{\partial T(h)}_{norm}$, purple dashed-dotted line in Figure 8), the error peak disappear and the model we obtain becomes the best model at all heights for both the fluxes and heating rates prediction. We therefore investigate the model $RNN^{\partial T(h)}_{norm}$ further by looking at the zonal climatology (Figure 9), the zonal MAE (Figure 10), the top climatology (Figure 11), the top MAE (Figure 12), the surface climatology (Figure 13), the sur-

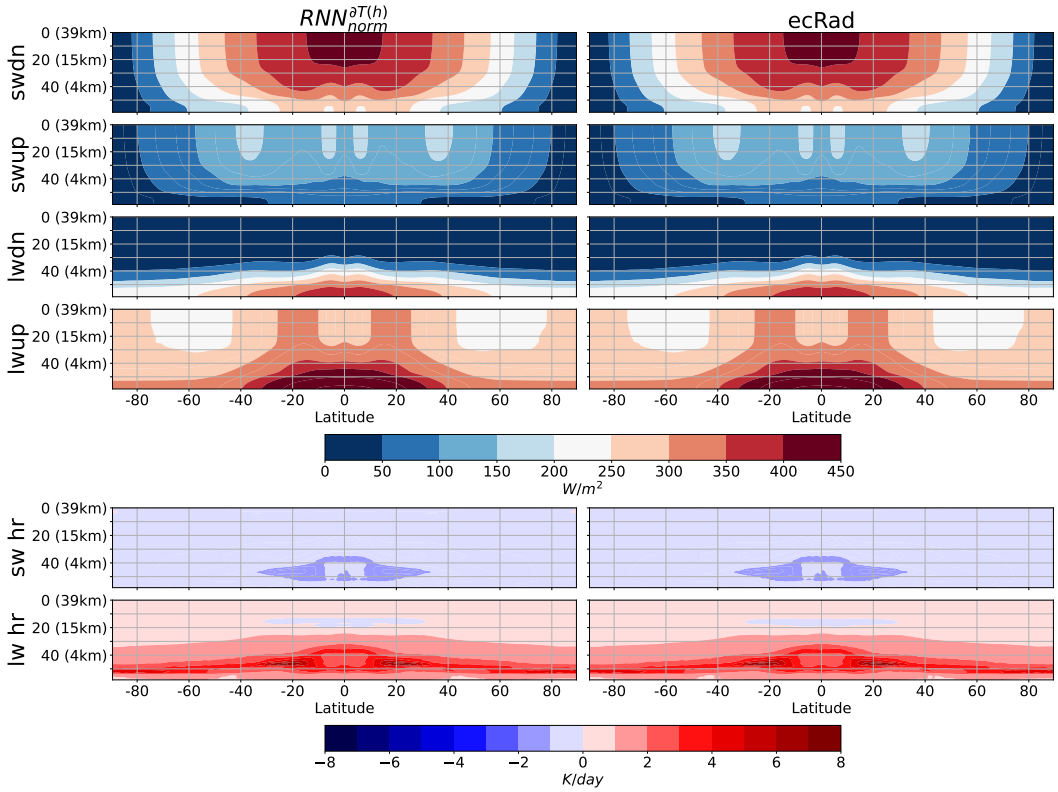

Figure 9: *Zonal climatology of the model* $RNN^{\partial T(h)}_{norm}$ *and of the solver ecRad. The mean is taken over all time steps and all columns in one degree latitude intervals.*

face MAE (Figure 14) and a pointwise comparison of ecRad and $RNN^{\partial T(h)}_{norm}$ predictions (Figure 15).

Zonal MAE and climatology: In Figure 9, we compare the zonal mean of $RNN^{\partial T(h)}_{norm}$ and ecRad's prediction. The mean is taken over all time steps and all columns in one degree latitude intervals. The zonal mean of the emulator $RNN^{\partial T(h)}_{norm}$ is similar, for both fluxes and heating rates, to the zonal mean of ecRad prediction.

In Figure 10, we plot the zonal MAE of $RNN^{\partial T(h)}_{norm}$. Similar to Figure 9, the mean is taken over all time steps and all columns in one degree latitude intervals. We observe that the shortwave error is concentrated at the lower height levels for the downward fluxes and on the upper levels for upward fluxes. This corroborates findings previously in Figure 8. Most of the flux prediction error appears in the tropical region. It is particularly large for the shortwave fluxes were the error reaches 10 W/m^2. In contrast, the zonal MAE for the longwave fluxes never exceeds 4.5 W/m^2. We can observe the error related to the clouds at 1km height where large errors occur below that height for the downward fluxes and above that height for the upward fluxes.

The error for longwave heating rates is significantly larger than the shortwave error. The most significant longwave heating rates errors are located between 500m and 3km height where the error reaches 0.9 K/day. We observe that the large errors in the longwave heating rates prediction corresponds to the height where the mean longwave heating rates is the highest.

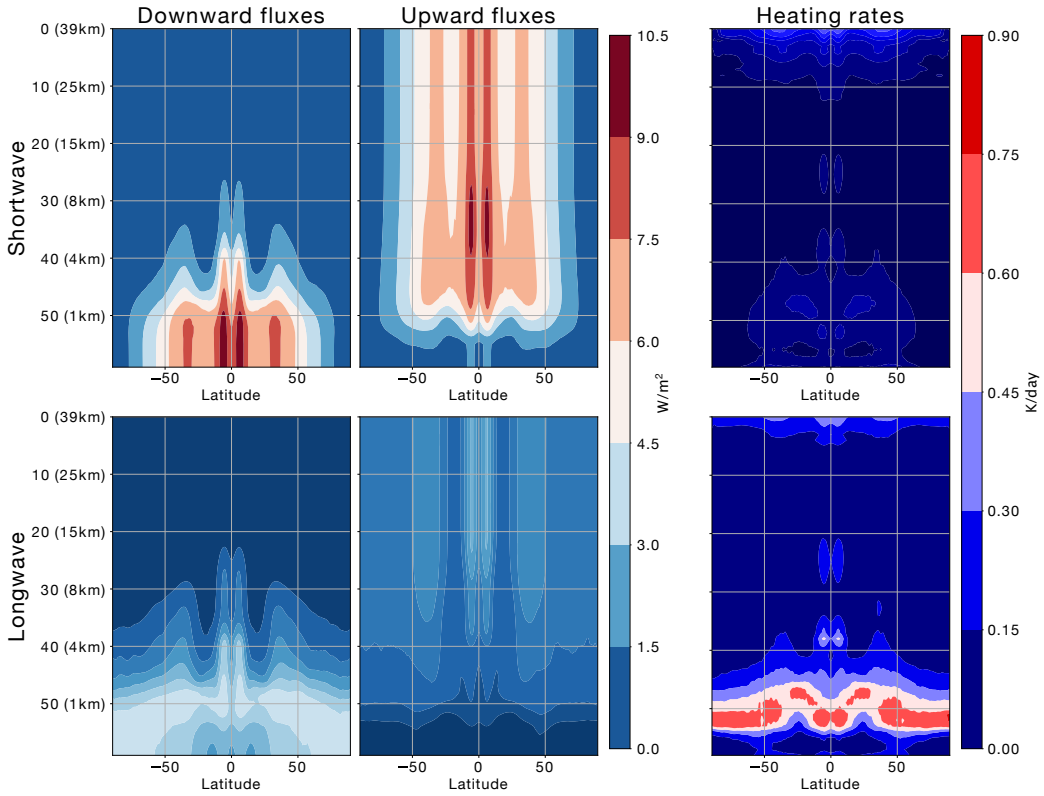

Figure 10: *Zonal MAE of the model* $RNN_{norm}^{\partial T(h)}$. *The mean is taken over all time steps and all columns in one degree latitude intervals.*

Top MAE and climatology: In Figure 11, we plot the time average prediction of $RNN_{norm}^{\partial T(h)}$ and of ecRad at the TOA. For the fluxes, $RNN_{norm}^{\partial T(h)}$ time average prediction is close to ecRad's.

For the heating rates, $RNN_{norm}^{\partial T(h)}$ and ecRad produce two different climatology. In particular $RNN_{norm}^{\partial T(h)}$ heating rates are too large (in absolute value) almost everywhere, except around -50, 50 degrees latitude where the heating rates are underestimated (in absolute value). For the shortwave heating rates, $RNN_{norm}^{\partial T(h)}$ underestimates the heating rates near the 8 positions which can face the sun in our dataset (recall that the data are stored every 3 hours), and overestimates the 9 positions in-between (observe that the 9 positions where the $RNN_{norm}^{\partial T(h)}$ heating rates are large are shifted compared to the 8 positions where ecRad predicts large heating rates.)

In Figure 12, we show the MAE of the $RNN_{norm}^{\partial T(h)}$ at the TOA. The mean is taken over time. The error is large for the upward fluxes and small for the downward fluxes. This is to be expected because the shortwave downward flux is straighforward to compute at the TOA and the longwave downward flux is essentially zero at the TOA. Most of the upward fluxes error is concentrated in two bands near the equator. Note that we also observe these error bands in the zonal MAE (Figure 10). We remark that the two bands we observe for the longwave upward flux in the climatology (Figure 11) are further away from the equator compared to the two error bands in Figure 12. This suggests that the $RNN_{norm}^{\partial T(h)}$ predicts the poleward side of the bands accurately but has large error on the equatorward side. For the heating rates, large error bands also appear around

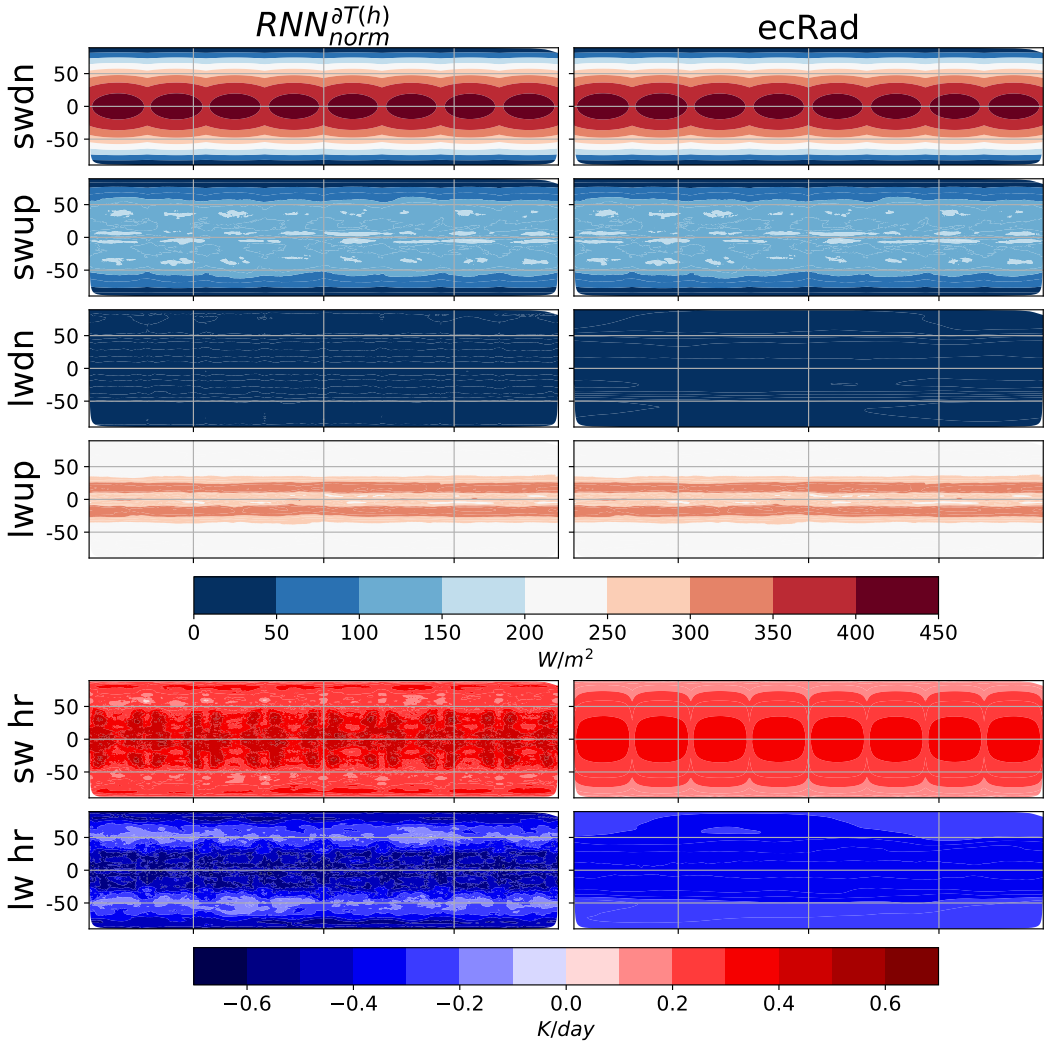

Figure 11: *TOA climatology of the model* $RNN^{\partial T(h)}_{norm}$ *and of the solver ecRad. The mean is taken over all time steps.*

-50 and 50 degree latitude. For the heating rates, the error is larger for the longwave and for the fluxes the error is largely dominated by the shortwave upward fluxes.

Surface MAE and climatology: In Figure 13, we plot the time average prediction of $RNN^{\partial T(h)}_{norm}$ and of ecRad at the surface. The averaged fluxes of $RNN^{\partial T(h)}_{norm}$ and ecRad as well as the heating rates appear fairly similar. Therefore a more detailed analysis of the MAE is necessary.

The heating rates time average prediction of $RNN^{\partial T(h)}_{norm}$ is close to ecRad prediction in contrast to what was observed at the TOA. For the longwave heating rates, we observe in the climatology several locations where the mean longwave heating rates is positive. Those locations probably correspond to stationary weather events. For a longer dataset, the heating rates climatology should tend to become zonally uniform, while for a one year training data set zonal asymmetries are to be expected.

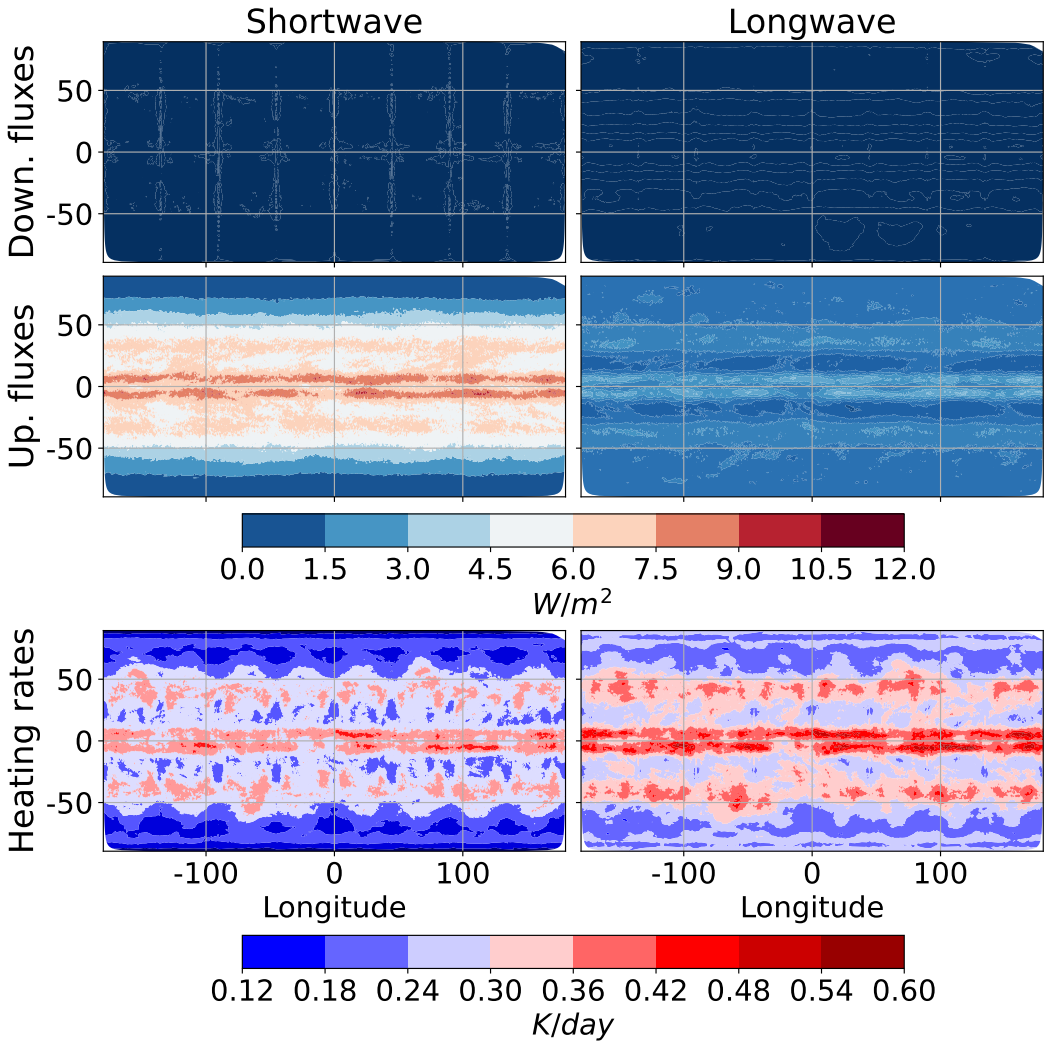

Figure 12: *Top MAE of the model* $RNN_{norm}^{\partial T(h)}$. *The mean is taken over all time steps.*

In Figure 14, we show the MAE of $RNN_{norm}^{\partial T(h)}$ at the surface. We observe that the fluxes error is largely dominated by the shortwave downward fluxes error. It is surprising that the upward shortwave flux error is so small compared to the downward flux error. Indeed the shortwave upward flux should be more complex to compute since it result from the interaction of the shortwave downward flux with the surface and the atmospheric layer closest to the surface.

In contrast to the fluxes error, the heating rates error is largely dominated by the longwave heating rates. The longwave heating rates error is mostly concentrated in the subtropics. Contrary to the TOA, the error near the equator is small. The error is concentrated in several locations at -50 and 50 degree latitude. At the same latitudes, we observed in the surface climatology positive longwave heating rates. As already discussed, for a larger test set, uniform error bands located at -50,50 degree latitude should appear instead.

Scatter plot: In Figure 15, for each flux and heating rate, we choose an interval that contains all predicted values (e.g. $[0, 1400]$ for shortwave down). We then divide the

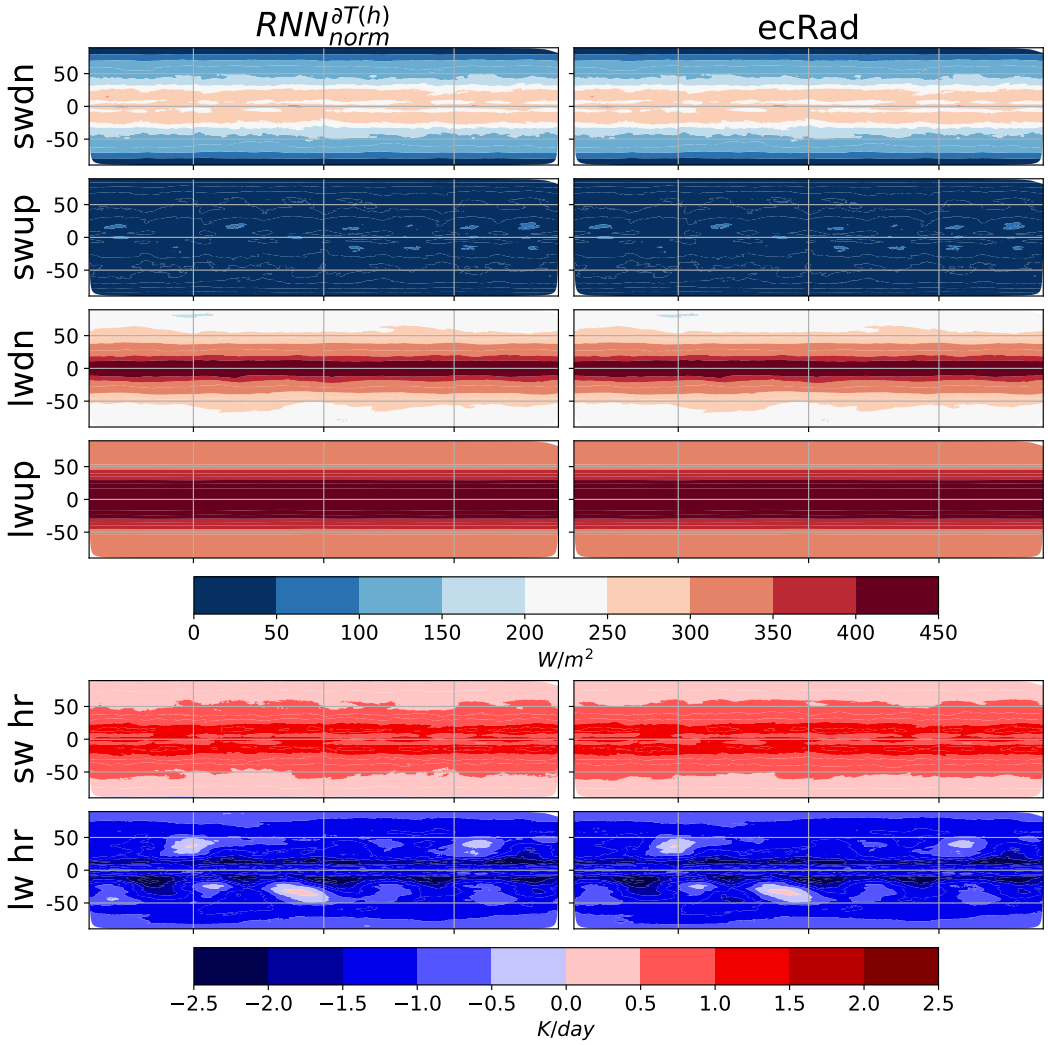

Figure 13: *Surface climatology of the model* $RNN^{\partial T(h)}_{norm}$ *and of the solver ecRad. The mean is taken over all time steps.*

interval into 100 smaller intervals (e.g. $[14 \cdot k, 14 \cdot (k+1)]$, $k = 0, \ldots, 99$ for shortwave down). Each prediction of ecRad and of $RNN^{\partial T(h)}_{norm}$ falls into one of the 100 intervals. Comparing ecRad and $RNN^{\partial T(h)}_{norm}$ predictions, we can assign each point of our test set (time, column and height) to one of the 100×100 squares. We then count the number of predicted values falling into each square. Ideally, the only squares with a nonzero count would be the one on the diagonal (i.e. ecRad and $RNN^{\partial T(h)}_{norm}$ predictions are close). The size of the squares is 14 W/m^2, 11.1 W/m^2, 4.4 W/m^2, 4.1 W/m^2 for respectively the shortwave downward and upward fluxes and for the longwave downward and upward fluxes. The size of the squares is 1.5 K/day and 2 K/day for respectively the shortwave and longwave heating rates.

The fluxes scatter plots are roughly symmetrical to the $x = y$ line with highest deviation from the $x = y$ line happening at different x coordinates ($\approx 700 W/m^2$ for shortwave down, $\approx 500 W/m^2$ for shortwave up, $\approx 200 W/m^2$ for longwave down and $\approx 300 W/m^2$ for longwave up.) For the shortwave heating rates, we observe that some predictions are negative when the exact solution is always positive. Furthermore for both

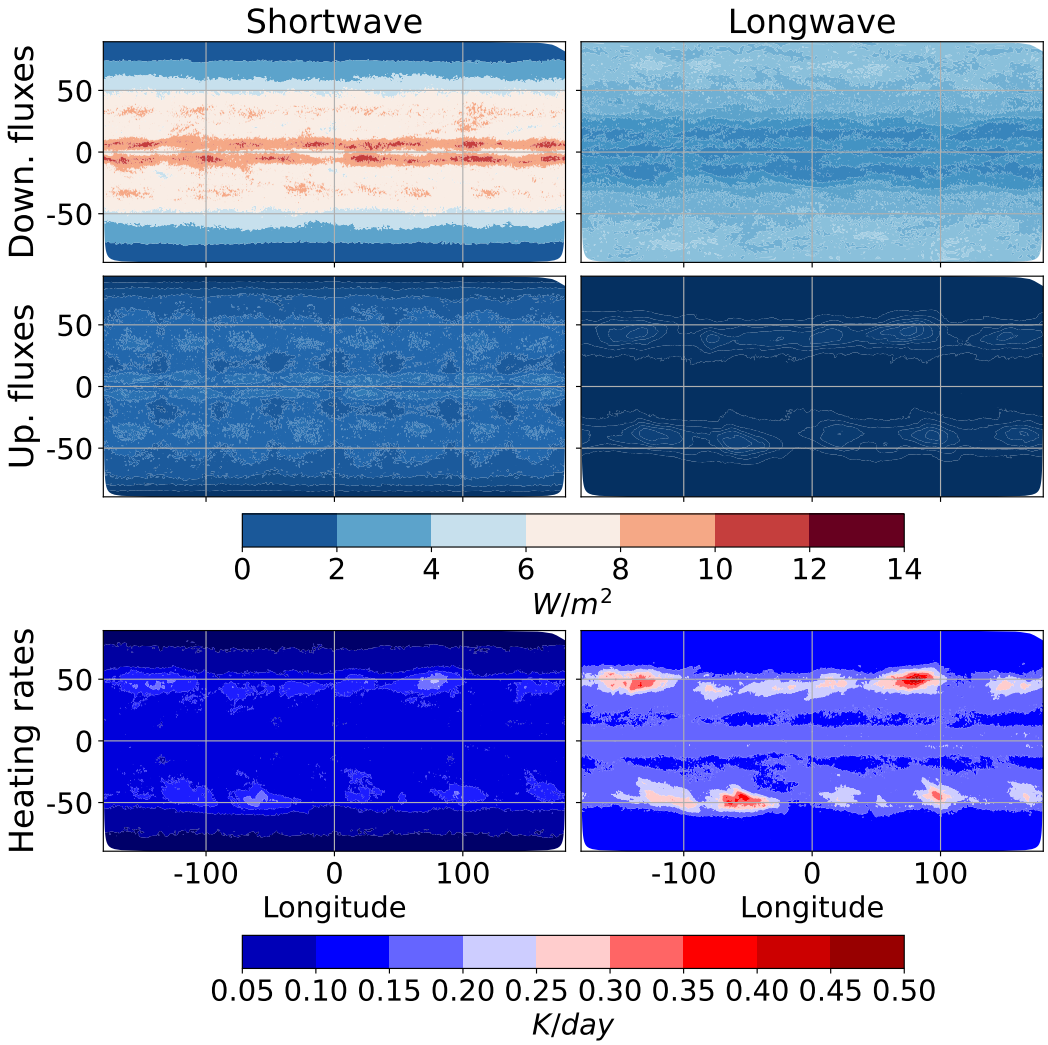

Figure 14: *Surface MAE of the model* $RNN_{norm}^{\partial T(h)}$. *The mean is taken over all time steps.*

longwave and shortwave heating, there are deviation of the prediction when the exact solution is zero, which points to some difficulties of the NNs predicting the rate of change of the corresponding flux along the day time or near the TOA where the heating rates drop to zero from one level to the next. Here, some fine tuning to the specifics of the underlying Numerical Weather Prediction (i.e., ICON) model might solve this issue. We also observe a few significant outliers for the shortwave heating rates, where the NN prediction reached 60 K/day while ecRad predicted 0 K/day.

5 Discussions

In the previous section, we investigated the performance of three NN architectures (MLP, UNet, RNN) with and without output normalization trained with the usual squared loss (Eq. 2), or with an additional heating rates penalty (Eq. 6), inspired by the column-integrated energy equation in an atmospheric column. Output normalization greatly improved our results. It is beneficial for each tested architecture and lead to improved accuracy for both fluxes and heating rates. Adding a heating rates penalty to the train-

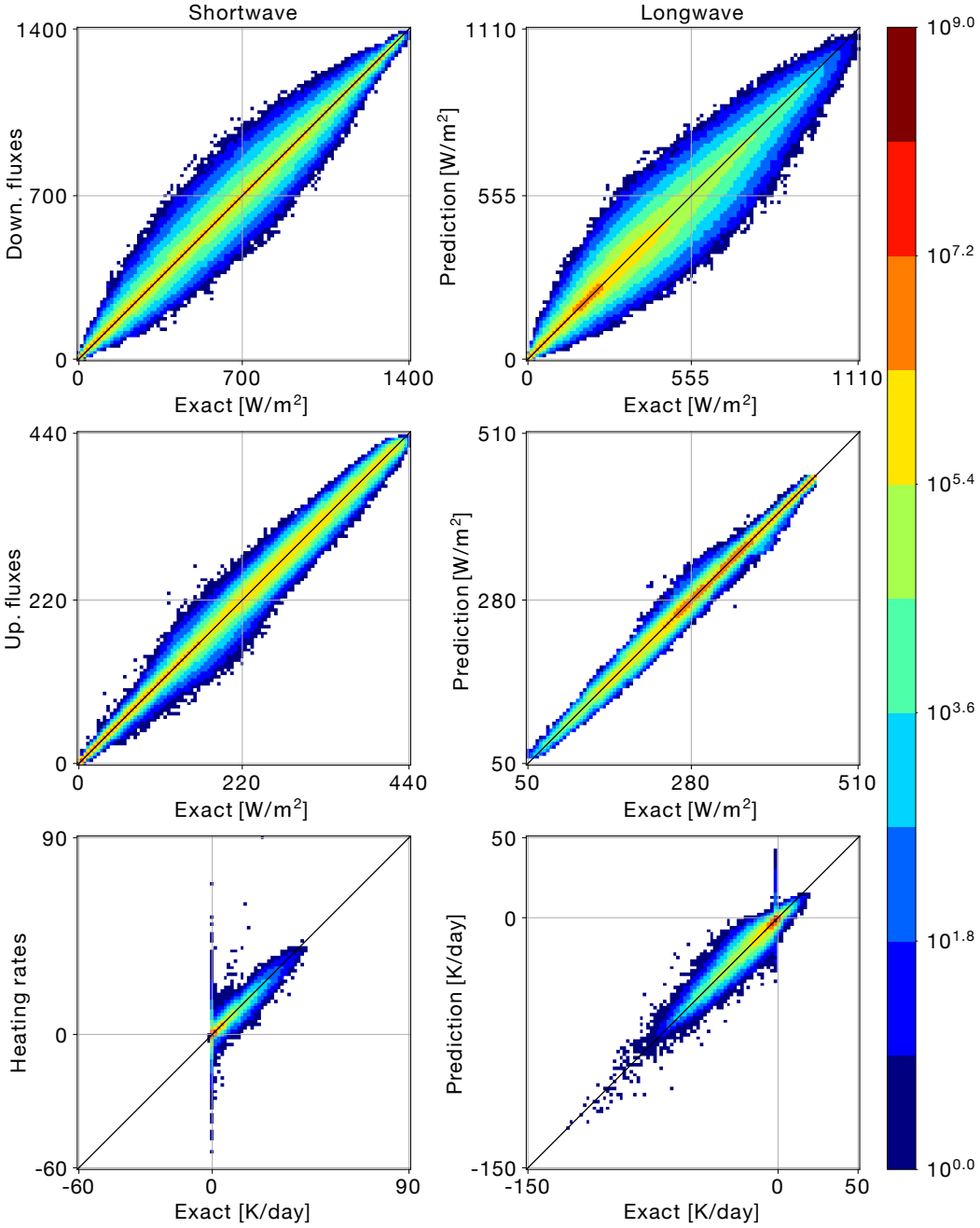

Figure 15: *For each column, time step and height level, we compare* $RNN_{norm}^{\partial T(h)}$ *(y-axis) and ecRad prediction (x-axis) and assign the result to one of the* 100×100 *squares.*

ing loss allowed us to improve the performance of RNN and UNet substantially. However, for MLPs, the additional heating rates penalty accentuated the error discontinuities already present in the MLP trained with squared loss, MLP^2. Similarly, we observed discontinuities in the error profile for the RNN without output normalization, $RNN^{\partial T(h)}$. However, together with the output normalization, the additional penalty term gives the most accurate RNN. For the UNet, the additional penalty, even without normalization, was highly beneficial. Note that amongst the models tested, the UNet is the only one for which we did not encounter discontinuities in the error profile. For both the UNet and RNN, height dependent weight for the heating rates penalty improved the results. For the MLPs it was reducing the accuracy and we only considered a height independent heating rates penalty.

Our best model is the RNN with physics-informed input and output normalization and heating rate loss (Eq. 6). From a physical point of view, it is not surprising that the RNN outperforms the other models. Indeed, physically the fluxes are crossing the atmospheric levels one after the other in the direction of the fluxes. The fluxes at a given height level h are then function of the fluxes in the height level $h-1$ above (downward fluxes), $h+1$ below (upward fluxes) and of the atmospheric composition in the given level h. This justifies the adopted bidirectional architecture. Although the $RNN_{norm}^{\partial T(h)}$ outperforms the other NNs at all heights, it does not outperform the RF for the heating rates prediction at the TOA, particularly for the shortwave. As already discussed, this may be due to the smoother profiles produced by the RF.

6 Summary

In this first of two studies, we provide a systematic overview of different ML methods to emulate the radiative transfer in the atmosphere. We tested ML architectures of varying complexity used in previous studies, including MLP (Chevallier et al., 1998; Ukkonen et al., 2020), UNet (Lagerquist et al., 2021), RNN (Ukkonen, 2022), and RF (Belochitski et al., 2011) and different variants of physics-constraints in the loss function to obtain a holistic picture of the performance of these ML methods before testing them online in a state-of-the-art weather and climate model.

We can conclude that achieving higher accuracy near TOA is more trivial through RFs without the cost of fine engineering needed with NNs. At TOA, the increase in MAE can be reduced by making the heating rates penalty term in the loss function height dependent. In general, however, it seems to be challenging for all tested architectures except for the RF to fit smoothly to near-zero values at the TOA. For the best performing NN model, the MAE is larger for shortwave than for longwave radiation fluxes but longwave heating rates exhibit larger errors compared to shortwave heating rates. Shortwave downward fluxes errors increase towards the surface as humidity content increases and is in particular pronounced around the equator where surface precipitation indicates the existence of deep convective clouds. Shortwave upward fluxes error increases towards the TOA with a local maximum at tropical cloud tops. For longwave fluxes, the error patterns are fairly similar but smaller in magnitude everywhere. In general, the error patterns point to cloud top and cloud bottom regions as the main source of error. While shortwave heating rates are well predicted, the derived longwave heating rates exhibit larger MAEs around 1 km height at most latitudes. The error hence seems to be associated with the top of the planetary boundary layer (PBL) and its strong humidity gradient and shallow clouds on its top. A way forward could be to train different models for different heights in the atmosphere or make the importance of input features during training height dependent.

For the design of ML-based radiation emulators, we propose to predict the corresponding fluxes and penalize training with the associated heating rates with height-depending weights. TOA and surface fluxes are important to predict because these are observabal

and hence used to constrain the energy balance of a climate model. The latter also serves as input to other model components in an ESM, such as the land model. Within the atmosphere however, the heating rates are of relevance to move the temperature state forward in time. In theory one could directly predict the heating rates and derive the flux through integration albeit losing information on its direction. Nevertheless, we opt for the presented compromise to predict the fluxes and penalize by the heating rates.

We recommend normalizing target features with respect to the largest value, e.g., found at the model top (proportional to the solar constant) and surface (according to Boltzmann's law) for shortwave and longwave radiation respectively. A recurrent network architecture running in both directions along height levels, suggested also by Ukkonen (2022), seems to be a natural choice because of the direction of radiative fluxes, however it remains to be seen how emerging ML architectures, such as transformers, will perform. Our preliminary experiments with transformers (not shown in this work) achieved good performance, yet far from the level of the RNN. Additional work required to make the transformer architecture competitive is left for future work.

In other preliminary studies, we also trained an RF to predict the Fourier coefficients of the radiation fluxes field using similar input variables as described above. Based on the predicted coefficients, the emulated radiation field can be reconstructed by Fourier synthesis. While that experiment produced reasonable results for the clear-sky flux, it proved to be more challenging to predict Fourier coefficients of the total flux field due to the high-frequency components associated with cloud-radiation interactions.

In an upcoming study, we will report on the online performance of the various models discussed here. To this end, the offline trained ML models will be coupled to ICON. This will also allow for alternating between ecRad and ML-based emulator(s) in a closed loop during runtime forming a potential hybrid model, which potential could be an attractive possibility for simulation beyond the weather scale.

Open Research Section

The data were generated using the ICON climate model described in Prill et al. (2023). The software is available for individuals on request at `https://code.mpimet.mpg.de/projects/iconpublic/wiki/How_to_obtain_the_model_code`. The codes to reproduce the results of this paper will be made available in `https://renkulab.io/gitlab/deepcloud/rfe`. Data to reproduce results of this work will be hosted at ETH Research Collection `https://www.research-collection.ethz.ch/` (with a DOI) together with the ICON runscript used to generate the full dataset. ETH Zurich's Research-Collection adheres to the FAIR principles and data is stored for at least 10 years.

Acknowledgments

This work was supported by Swiss Data Science Center (SDSC grant C20-03). We thank Eniko Székely for helpful discussions on decision trees.

References

Belochitski, A., Binev, P., DeVore, R., Fox-Rabinovitz, M., Krasnopolsky, V., & Lamby, P. (2011, September). Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. *Journal of Computational and Applied Mathematics*, *236*(4), 447–460. Retrieved from `https://doi.org/10.1016/j.cam.2011.07.013` doi: 10.1016/j.cam.2011.07.013

Belochitski, A., & Krasnopolsky, V. (2021, December). Robustness of neural network emulations of radiative transfer parameterizations in a state-of-the-art

general circulation model. *Geoscientific Model Development*, *14*(12), 7425–7437. Retrieved from https://doi.org/10.5194/gmd-14-7425-2021 doi: 10.5194/gmd-14-7425-2021

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

Brenowitz, N. D., & Bretherton, C. S. (2019, August). Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, *11*(8), 2728–2744. Retrieved from https://doi.org/10.1029/2019ms001711 doi: 10.1029/2019ms001711

Chéruy, F., Chevallier, F., Morcrette, J.-J., Scott, N. A., & Chédin, A. (1996). Une méthode utilisant les techniques neuronales pour le calcul rapide de la distribution verticale du bilan radiatif thermique terrestre. *Comptes Rendus de l'Académie des Sciences*, *322*, 665—672. Retrieved from https://hal.archives-ouvertes.fr/hal-02954375

Chevallier, F., Chéruy, F., Scott, N. A., & Chédin, A. (1998, November). A neural network approach for a fast and accurate computation of a longwave radiative budget. *Journal of Applied Meteorology*, *37*(11), 1385–1397. Retrieved from https://doi.org/10.1175/1520-0450(1998)037<1385:annafa>2.0.co;2 doi: 10.1175/1520-0450(1998)037⟨1385:annafa⟩2.0.co;2

Chevallier, F., Morcrette, J.-J., Chéruy, F., & Scott, N. A. (2000, January). Use of a neural-network-based long-wave radiative-transfer scheme in the ECMWF atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, *126*(563), 761–776. Retrieved from https://doi.org/10.1002/qj.49712656318 doi: 10.1002/qj.49712656318

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., & Yacalis, G. (2018, June). Could machine learning break the convection parameterization deadlock? *Geophysical Research Letters*, *45*(11), 5742–5751. Retrieved from https://doi.org/10.1029/2018gl078202 doi: 10.1029/2018gl078202

Hogan, R. J., & Bozzo, A. (2018, August). A flexible and efficient radiation scheme for the ECMWF model. *Journal of Advances in Modeling Earth Systems*, *10*(8), 1990–2008. Retrieved from https://doi.org/10.1029/2018ms001364 doi: 10.1029/2018ms001364

Kashinath, K., Mustafa, M., Albert, A., Wu, J.-L., Jiang, C., Esmaeilzadeh, S., ... Prabhat (2021, February). Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *379*(2194), 20200093. Retrieved from https://doi.org/10.1098/rsta.2020.0093 doi: 10.1098/rsta.2020.0093

Kato, S., Xu, K.-M., Wong, T., Loeb, N. G., Rose, F. G., Trenberth, K. E., & Thorsen, T. J. (2016, September). Investigation of the residual in column-integrated atmospheric energy balance using cloud objects. *Journal of Climate*, *29*(20), 7435–7452. Retrieved from https://doi.org/10.1175/jcli-d-15-0782.1 doi: 10.1175/jcli-d-15-0782.1

Krasnopolsky, V. M. (2014). Nn-tsv, ncep neural network training and validation system. *National Oceanic and Atmospheric Administration*. Retrieved from https://repository.library.noaa.gov/view/noaa/6945 doi: 10.7289/V5QR4V2Z

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Belochitski, A. A. (2008, October). Decadal climate simulations using accurate and fast neural network emulation of full, longwave and shortwave, radiation. *Monthly Weather Review*, *136*(10), 3683–3695. Retrieved from https://doi.org/10.1175/2008mwr2385.1 doi: 10.1175/2008mwr2385.1

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005, May). New

approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Monthly Weather Review*, *133*(5), 1370–1383. Retrieved from https://doi.org/10.1175/mwr2923.1 doi: 10.1175/mwr2923.1

Krasnopolsky, V. M., Fox-Rabinovitz, M. S., Hou, Y. T., Lord, S. J., & Belochitski, A. A. (2010, May). 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. Retrieved from https://doi.org/10.1175/2009mwr3149.1 doi: 10.1175/2009mwr3149.1

Lagerquist, R., Turner, D., Ebert-Uphoff, I., Stewart, J., & Hagerty, V. (2021, July). Using deep learning to emulate and accelerate a radiative-transfer model. *Journal of Atmospheric and Oceanic Technology*. Retrieved from https://doi.org/10.1175/jtech-d-21-0007.1 doi: 10.1175/jtech-d-21-0007.1

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. Retrieved from https://doi.org/10.1109/5.726791 doi: 10.1109/5.726791

Liu, Y., Caballero, R., & Monteiro, J. M. (2020, September). RadNet 1.0: exploring deep learning architectures for longwave radiative transfer. *Geoscientific Model Development*, *13*(9), 4399–4412. Retrieved from https://doi.org/10.5194/gmd-13-4399-2020 doi: 10.5194/gmd-13-4399-2020

Meyer, D., Hogan, R. J., Dueben, P. D., & Mason, S. L. (2022, mar). Machine learning emulation of 3d cloud radiative effects. *Journal of Advances in Modeling Earth Systems*, *14*(3). doi: 10.1029/2021ms002550

Morcrette, J.-J. (1991). Radiation and cloud radiative properties in the european centre for medium range weather forecasts forecasting system. *Journal of Geophysical Research*, *96*(D5), 9121. Retrieved from https://doi.org/10.1029/89jd01597 doi: 10.1029/89jd01597

O'Gorman, P. A., & Dwyer, J. G. (2018, October). 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. Retrieved from https://doi.org/10.1029/2018ms001351 doi: 10.1029/2018ms001351

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... others (2018). Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Pal, A., Mahajan, S., & Norman, M. R. (2019, June). Using deep neural networks as cost-effective surrogate models for super-parameterized e3sm radiative transfer. *Geophysical Research Letters*, *46*(11), 6069–6079. Retrieved from https://doi.org/10.1029/2018gl081646 doi: 10.1029/2018gl081646

Prill, F., Reinert, D., Rieger, D., & Zängl, G. (2020). Icon tutorial 2020: Working with the icon model. *Deutscher Wetterdienst*. Retrieved from https://www.dwd.de/EN/ourservices/nwv_icon_tutorial/pdf_volume/icon_tutorial2020_en.pdf doi: 10.5676/DWD_PUB/NWV/ICON_TUTORIAL2020

Prill, F., Reinert, D., Rieger, D., & Zängl, G. (2023). Icon tutorial 2023: Working with the icon model. Retrieved from https://www.dwd.de/EN/ourservices/nwv_icon_tutorial/pdf_volume/icon_tutorial2023_en.pdf?__blob=publicationFile&v=3 doi: 10.5676/DWD_PUB/NWV/ICON_TUTORIAL2023

Roh, S., & Song, H.-J. (2020, November). Evaluation of neural network emulations for radiation parameterization in cloud resolving model. *Geophysical Research Letters*, *47*(21). Retrieved from https://doi.org/10.1029/2020gl089444 doi: 10.1029/2020gl089444

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986, October). Learning repre-

sentations by back-propagating errors. *Nature*, *323*(6088), 533–536. Retrieved from https://doi.org/10.1038/323533a0 doi: 10.1038/323533a0

Scott, N. A., & Chedin, A. (1981, July). A fast line-by-line method for atmospheric absorption computations: The automatized atmospheric absorption atlas. *Journal of Applied Meteorology*, *20*(7), 802–812. Retrieved from https://doi.org/10.1175/1520-0450(1981)020<0802:aflblm>2.0.co;2 doi: 10.1175/1520-0450(1981)020⟨0802:aflblm⟩2.0.co;2

Ukkonen, P. (2022, April). Exploring pathways to more accurate machine learning emulation of atmospheric radiative transfer. *Journal of Advances in Modeling Earth Systems*, *14*(4). Retrieved from https://doi.org/10.1029/2021ms002875 doi: 10.1029/2021ms002875

Ukkonen, P., Pincus, R., Hogan, R. J., Nielsen, K. P., & Kaas, E. (2020, December). Accelerating radiation computations for dynamical models with targeted machine learning and code optimization. *Journal of Advances in Modeling Earth Systems*, *12*(12). Retrieved from https://doi.org/10.1029/2020ms002226 doi: 10.1029/2020ms002226

Veerman, M. A., Pincus, R., Stoffer, R., van Leeuwen, C. M., Podareanu, D., & van Heerwaarden, C. C. (2021, February). Predicting atmospheric optical properties for radiative transfer computations using neural networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *379*(2194), 20200095. Retrieved from https://doi.org/10.1098/rsta.2020.0095 doi: 10.1098/rsta.2020.0095

Yuval, J., O'Gorman, P. A., & Hill, C. N. (2021, March). 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). Retrieved from https://doi.org/10.1029/2020gl091363 doi: 10.1029/2020gl091363

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2020, June). UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, *39*(6), 1856–1867. Retrieved from https://doi.org/10.1109/tmi.2019.2959609 doi: 10.1109/tmi.2019.2959609

Appendix A Random forest output normalization

In Figure A1, we compare the random forest MAE on the test set with and without normalization of the outputs presented in Section 2.2. The normalization procedure increases significantly the accuracy of the random forest for the shortwave fluxes prediction. For the longwave downward flux, the normalization has essentially no effect on the error. For the longwave upward flux, the normalization increases the accuracy below 1 km. Between 1 km and 10 km, the accuracy is slightly reduced and above 10km the normalization has no effect on the accuracy. We still recommend the longwave output normalization as it increases the longwave upward flux significantly near the surface.

Appendix B MLP additional loss functions

We discuss the following MLPs:

1. *$MLP^{\int E}$: MLP with additional column-integrated energy penalty*
   The loss function of this NN is given by Eq. (4). All architectural details remain identical to MLP^2.
2. *$MLP^{\partial T(h)}$ MLP with height dependent heating rates penalty*
   The loss function of this NN is similar to $UNet^{\partial T(h)}$. All architectural details remain identical to MLP^2.

$\text{MLP}^{\int E}$ is penalized if column integrated energy, defined as the difference between the net radiation at the top and surface without distinction between shortwave and longwave, is not accurately predicted Eq. (4). The idea is, that this MLP preserves energy in the climate model. The MLP tries to satisfy the new penalty by modifying the TOA and surface fluxes. This completely breaks the models at those heights. Furthermore it adds oscillation in the longwave fluxes and heating rates.

$MLP^{\partial T(h)}$ has a height dependent heating rates penalty. With the penalty, the MLP becomes inaccurate at all heights for both the fluxes and heating rates.

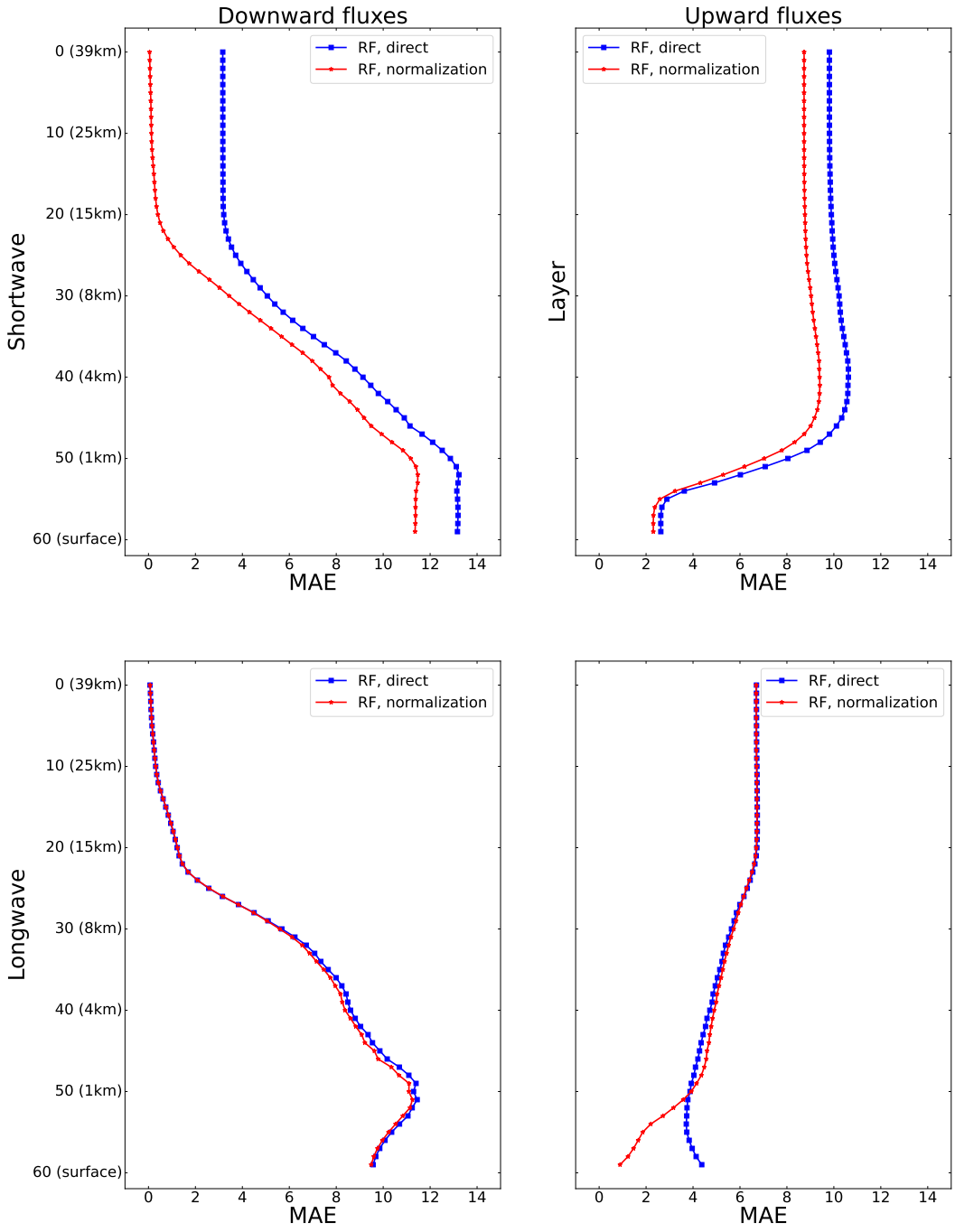

Figure A1: *Effect of the normalization described in Section 2.2 for the random forest. The outputs are not normalized for the RF error drawn in blue and they are normalized for the RF drawn in red.*

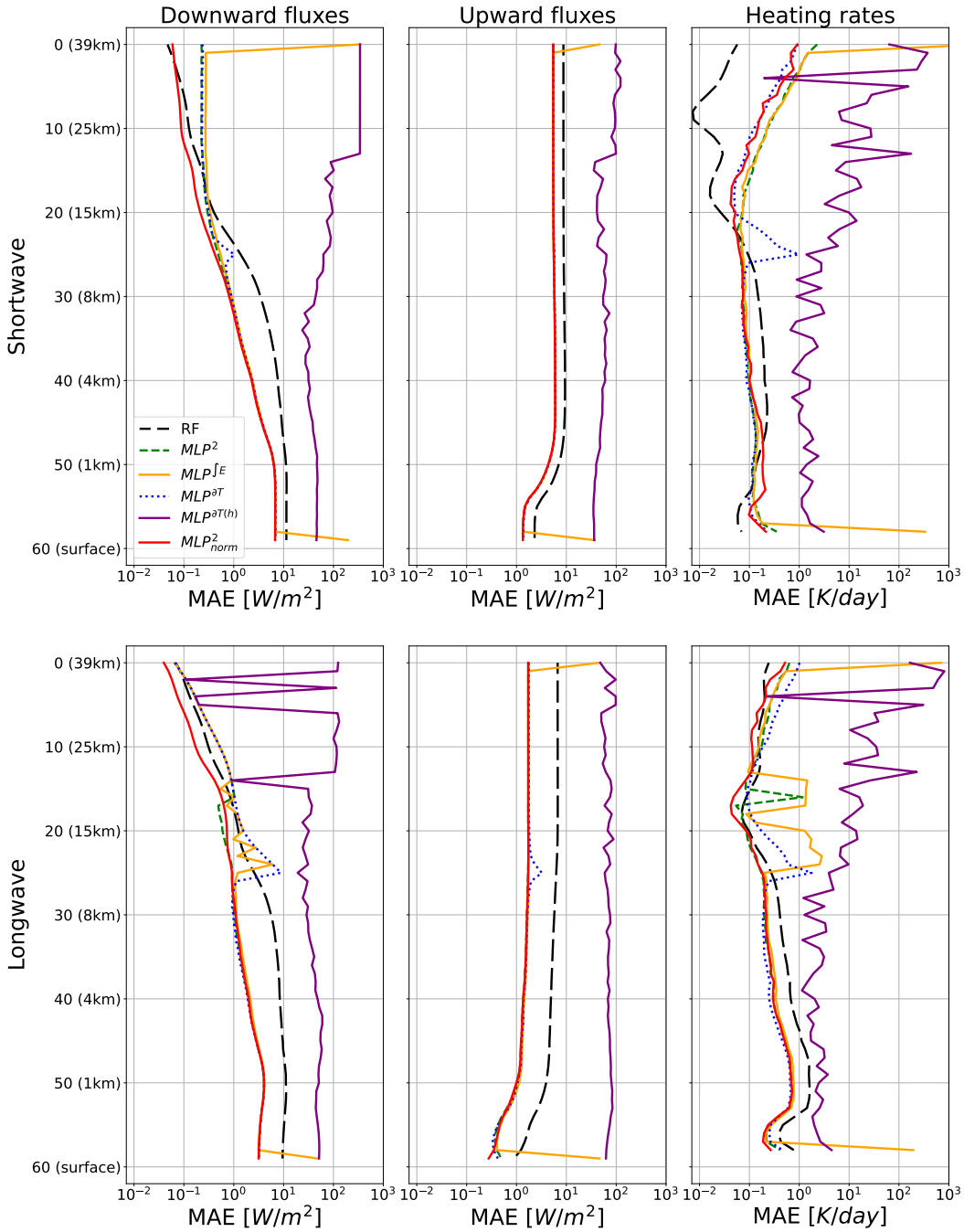

Figure B1: *MAE of the MLPs and of the RF emulator for the shortwave and longwave downward fluxes, upward fluxes and heating rates. Legend: RF; random forest,* MLP^2*; MLP trained with squared error loss,* MLP^2_{norm}*;* MLP^2 *with normalized output,* $MLP^{\partial T}$*;* MLP^2 *with an additional penalty for the inferred heating rates,* $MLP^{\int E}$*;* MLP^2 *with loss function top and bottom energy penalty,* $MLP^{\partial T(h)}$*;* $MLP^{\partial T}$ *with height dependent penalty. The models are described in Section 2.4.*