Optics EXPRESS

Physics constraint Deep Learning based radiative transfer model

QUANHUA LIU[1,*] AND XINGMING LIANG[2]

[1] *Center for Satellite Applications and Research, National Environmental Satellite Data, and Information Service (NESDIS), National Oceanic and Atmospheric Administration, College Park, Maryland 20740, USA*
[2] *Earth System Science Interdisciplinary Center, University of Maryland, College Park, Maryland 20740, USA*
[*] *quanhua.liu@noaa.gov*

Abstract: Deep Learning (DL) open sources libraries such as TensorFlow, Keras, and PyTorch have been widely and successfully applied for many applications in a forward model. We have developed the DL radiative transfer model over Oceans under a clear-sky condition. However, the derived physical model from the DL forward model has difficulties in predicting physical properties such as the Jacobian, because multiple solutions can fit the forward model results during the deep learning training process. The Jacobian model in a radiative transfer can calculate radiance sensitivities on geophysical parameters, which are required by satellite radiance assimilation in support of weather forecasts and for retrieving environmental data records. In this study, we introduce a physics constraint into our deep learning training for deriving the forward model that retains right physics. With this physics constraint, the radiance sensitivities are well captured by this new DL radiative transfer.

1. Introduction

Environmental data from satellite measurements have dramatically increased as the need for accurate weather forecasts and climate studies [1]. Industries also deliver a large amount of valuable environmental data from their smallSATs and CubeSATs. To quantitatively utilize the environmental data, a very fast and accurate radiative transfer model (RTM) [2] is required to simulate and assimilate the environmental information. The Community Radiative Transfer Model (CRTM) [3], developed at the Joint Center for Satellite Data Assimilation in the United States, is a fast and accurate radiative transfer model. The CRTM is in operation for weather forecasts and the generation of environmental data records in the United States and other countries. The CRTM model is fast but not fast enough so that only less than 10% satellite remote sensing data are assimilated, for example a subset of 124 channels from 8461 IASI channels is used for numerical weather prediction [4]. In addition, we have to choose fewer stream members in degraded scattering calculations for mitigating the computational resources. The community requirement and recent deep learning innovation motivated this study.

Rapid innovations in deep learning from both software and hardware as well as society needs [5] promote the public release of open sources libraries such as TensorFlow and Keras libraries (https://www.tensorflow.org/tutorials/keras/keras_tuner). Open resource libraries play an important role in exploring big data applications. Their applications in remote sensing [6,7], image processing [8], and numerical weather prediction [9,10] are presently being explored [11], in which artificial intelligent (AI) based radiative transfer model development receives great attention [12,13]. There are two types of radiative transfer models used in weather forecasts and in the retrieval of environmental data records; broadband fluxes in downward and upward directions and narrowband radiance at any given direction. Broadband fluxes are used for calculating atmospheric cooling and heating rates as well as the energy budget at the surface and at the top of the atmosphere. An initial shallow neural network (SNN)-based model, NeuroFlux, has been

#493818
Journal © 2023
https://doi.org/10.1364/OE.493818
Received 20 Apr 2023; revised 13 Jun 2023; accepted 14 Jun 2023; published 11 Aug 2023

developed to estimate the longwave radiation budget from the top of the atmosphere to the earth's surface [14]. SNNs have been further expanded to more complex architectures, which include multiple hidden layers (Deep Neural Network, DNNs) and resulted in improved RTM calculation accuracy and efficiency, thereby demonstrating the potential to replace conventional RTMs in climate models [15,16]. Recently, Yao et al. [17] have incorporated a physical relationship between fluxes and heating rates into a layer of the network so that the energy conservation can be satisfied and the predicted accuracy was improved as well. Mishra and Molinaro [18] applied physics informed neural networks (PINNs) for simulating radiative transfer, which are trained by minimizing the residual of the underlying radiative transfer equations.

AI-based radiative transfer model used for radiance calculations are significantly more complicated than those used for flux calculations because radiances depend on the sensor observing zenith and azimuth angles at both upward and downward directions. We had developed an AI-based radiative transfer model using Keras libraries for forward radiance calculations (referred to AI_Keras_rt hereafter) [19]. We perform microwave radiative transfer calculations because microwave remote sensing data contribute the most in the direct radiance assimilation to weather forecasting. It is found that the forward radiance calculations are pretty accurate, whereas the Jacobian calculations are often not. For microwave window channels and water vapor channels, the AI-based Jacobian calculations are still questionable. This indicates that the AI-based forward model doesn't take up the complex physics used in the radiative transfer process, although the forward model can map the input features to the outputs very well. The radiative transfer model deals with layer and surface statistical properties. Physics in the radiative transfer model refers to macrophysics for example the radiance sensitivity (or Jacobian) on water vapor in the atmosphere, although the macrophysics foundation is microphysics such as molecular emission and absorption described by quantum mechanics.

The Jacobian in our derived model agrees with the finite difference between two forward model calculations very well. Both forward and the derived models use the same configuration (input, output and hidden layers as well as corresponding nodes) and the same weights/biases. Therefore, the Jacobian calculated from the derived model is fully consistent with the finite difference based on the AI forward model. Generally speaking, the weights and biases are updated by minimizing a loss function according to the forward model results in an iterative way through a backpropagation algorithm [6]. However, this minimization only provides a solution for the local minimum. There are multiple solutions for updating weights/biases. We also found that using the Adam optimization [20] can reach a lower minimum of the loss function. The improvement is more for the AI forward model and the accuracy of the Jacobian calculations is still unsatisfactory.

In this study, we introduce physics constraints to the forward model training for improved weights/biases that can better represent the sensitivities of atmospheric temperatures and water vapors on radiances or brightness temperatures. The physics constraints are neither a part of the AI forward model nor the derived model (herein the Jacobian model). The physics constraints help us to find physics representative weights/biases. This new methodology is relevant to other applications although this study focuses on an AI-based radiative transfer model. Section II details the new methodology used to introduce the physics constraints. The training data and the validation data are described in Section III. The comparison results of with and without the physics constraints are given in Section IV. CPU timing table for various algorithm is given in Section IV as well. Section V is for discussions and conclusions.

2. Deep learning algorithm

We use a supervised and fully connected artificial neural network (ANN) in this study. ANN is one of the methodologies in deep learning (DL). DL is a subfield or a branch of artificial intelligence (AI). The fully connected ANN here consists of a series of fully connected layers those link every neuron (node) in one layer to every neuron (node) in the adjacent layer. The first artificial neural network was invented by psychologist Frank Rosenblatt and called Perceptron [21].

The ANN, we used here, has one input layer, several hidden layers, and one output layer. The input data are used to predict the outputs and the ANN weights/biases are updated by minimizing the difference between the model predicted and truths or reference outputs in supervision. We found that using ReLu activation at hidden layers and a linear function at an output layer can deliver promising results for microwave radiative transfer calculations in the atmosphere.

2.1. Forward model

The microwave radiative transfer computes the brightness temperatures under clear sky conditions by considering surface and atmospheric emission governed by the Planck function at a local temperature and atmospheric absorption such as water vapor and oxygen molecular absorptions. The water surface microwave emissivity and reflectivity, which are a function of the surface wind and the surface temperature as well as sensor zenith and azimuth angles, also play an important role in the radiative transfer calculations. The predicted brightness temperature (Y^{pred}) under the ANN umbrella will be a function of the weights, the biases, activation functions and the vector X_0 containing input variables, that is:

$$Y^{pred} = F(w, b, A, X_0). \tag{1}$$

F here is an ANN or AI forward model. w, b, A represent weights, biases, and activations, respectively. In this study, we detail the AI model for simulating Joint Polar Satellite System (JPSS) Advanced Technology Microwave Sounder (ATMS) observations [22]. We introduced physic constraints, which is new to an AI radiative transfer model and the technique can also be applied to develop ANN models for other applications such as radiation budget at the top of the atmosphere in climate studies. The microwave sounder ATMS play an important role in weather forecasts and climate studies. The ANN used in this study is configured with one input layer, three hidden layers, and one output layer. As shown in Fig. 1, the input layer values are multiplied by weights in a matrix and adding biases are evaluated at the next layer through the activation function. These activation function values are the inputs to the next layer. The output layer contains the brightness temperatures for 22 ATMS channels using a linear activation function.

For the given ANN in Fig. 1, Eq. (1) can be written explicitly:

$$Y^{pred} = b_4 + w_4 A_3(b_3 + w_3 A_2(b_2 + w_2 A_1(b_1 + w_1 X_0))). \tag{2}$$

In a standard process, the weights and biases are updated in a backward propagation by minimizing the loss function in these iterative processes. We choose commonly used mean square error (MSE) as the loss function for K channels:

$$loss = \frac{1}{S \times K} \sum_{j=1}^{S} \sum_{i=1}^{K} (Y^{pred}(i,j) - Y^{true}(i,j))^2 \tag{3}$$

where S is the sample number or batch size. Dynamic range and variability can be quite different among input features and outputs. Normalizing the input and output data results in a Gaussian distribution with a mean close to zero. Normalizing the data generally speeds up learning and leads to faster convergence and a better accuracy. We normalized input data and the reference data for output as follows:

$$\overline{X_0} = \frac{X_0 - \mu_x}{\sigma_x} \tag{4a}$$

$$\overline{Y^{true}} = \frac{Y^{true} - \mu_y}{\sigma_y} \tag{4b}$$

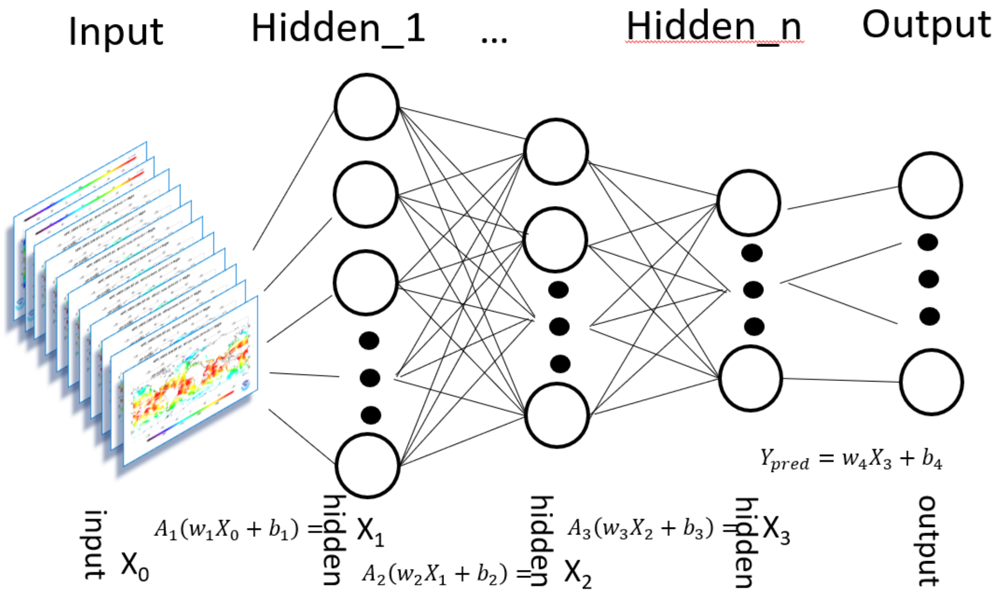

Fig. 1. A schematic process of the ANN for simulating ATMS brightness temperatures. A_k represents the activation function. w_k and b_k are the weights and biases, respectively.

where μ and σ are the mean value and the standard deviation, respectively. X_0 (Y^{true}) and $\overline{X_0}$ ($\overline{Y^{true}}$) are for the model input (reference output) data before and after standardization, respectively. In the application, the prediction (output) needs to be converted back to the original value as:

$$Y^{pred} = \sigma_y \times \overline{Y^{pred}} + \mu_y. \tag{5}$$

The normalization helps ANN training in both accuracy and computational efficiency (less iterations). Using Adam optimization algorithm further improves the forward prediction accuracy. The Adam optimization is an extension to stochastic gradient descent [20].

We can skip using Eq. (5) if we incorporate the mean value and the standard deviation of the input features (outputs) into weights/biases for the first layer (last layer). The incorporation can simplify the calculation of forward radiance and Jacobian calculations for the users.

2.2. Jacobian model

The ANN model used in this study is configured for the forward model to simulate satellite observations. The Jacobian model is a derived model from the forward model rather than a part of the ANN configuration for the forward model. Jacobian is a derivative of the radiative transfer output variable with respect to each of the input variables. The Jacobian model contains the most important sensitivities (physics) such as how the satellite measured radiances change with the surface temperature, water vapor and carbon dioxide in the atmosphere. At numerical weather prediction centers, operational radiative transfer models [3] use physics based and fast algorithms in the forward model. The Jacobian model is also derived from its forward model through tangent linear and adjoint programming. The forward and Jacobian models must be consistent, which is very important for stability in weather forecasting.

It is neither practical nor accurate to design one ANN for both forward and Jacobian models because unknown parameters in the Jacobian model are considerably more than those in the forward model. In this study, the predicted variables are 22 ATMS brightness temperatures in the forward model whereas there are 4136 (22 channels by 188 input parameters) predicted variables in the Jacobian model even under clear conditions. It would be hard for ANN to accurately predict so many variables efficiently.

We can derive the Jacobian model from the forward model. The explicit expression of the Jacobian model was given in our previous paper (see Eq. (A12) of Liang et al. [19]. Based on Fig. 1, we use slightly different notation here for better understanding with explanations.

As illustrated in Fig. 1, the Jacobian model (J) can be directly derived from Eq. (2):

$$J(X_0) = \frac{\partial Y^{pred}}{\partial X_0} = w_4 \frac{\partial Y^{pred}}{\partial A_3} \frac{\partial A_3}{\partial A_2} \frac{\partial A_2}{\partial A_1} \frac{\partial A_1}{\partial X_0}. \tag{6}$$

If n is the number of hidden layers and the linear activation function is used for the output layer, the Jacobian of the ANN can be expressed as:

$$J = w_{n+1} \prod\nolimits_{k=n}^{1} A_k^{'}(X_{k-1}) w_k\,. \tag{7}$$

J is the Jacobian which is derived from the AI forward model. It is worth to be mentioned that the activation function A_k is a vector in the forward model but the derivative of activation function A_k' is a square and diagonal matrix in the Jacobian model with the derivative of the elements of the vector A_k. The Jacobian model demands considerably more computational time than the forward model because of the expansion of the activation function from a vector in the forward model to a matrix in the Jacobian model. The computational CPU time in the Jacobian model is roughly proportional to the number of channels of sensors. Therefore, the AI Jacobian model calculation can be very expensive for hyperspectral sensors with thousands of channels.

On other hand, the AI adjoint model is computationally efficient and it only takes about $2 \sim 3$ times more CPU time as compared to the forward model calculation. An adjoint model calculates the total sensitivity of observations on geophysical parameters. A Jacobian model calculates the sensitivity on geophysical parameters for each channel of observations. In a physics-based radiative transfer model like the CRTM, optical properties, Planck functions, and the radiative transfer solution need to be evaluated for each channel regardless of Adjoint or Jacobian model. The computational times between Adjoint and Jacobian models are nearly the same. However, AI radiative transfer model is a fitting or mapping model. Fitting total sensitivity is much faster than fitting channel-dependent sensitivity. For example, JPSS Cross-track Infrared Sounder (CrIS) has 2211 channels. The AI adjoint calculations will be about 700 times faster than the Jacobian calculations. The adjoint value is the sum of Jacobian values over all channels. In this study, the adjoint values are a vector (188 elements) corresponding to 188 input variables. The Jacoban values are represented by a matrix of size 188 by 22 ATMS channels. The adjoint model has also been used in direct radiance assimilation at the European Centre for Medium-Range Weather Forecasts (ECMWF). It would be very efficient to use AI adjoint model than using the AI Jacobian model.

2.3. Add physics constraints to ANN training

For a given ANN and the training data including both inputs and true outputs, the weights and biases updates may depend on some parameters setting or model tuning, for example the batch size, the number of epochs, activation functions, and base learning rate. The ReLU is the most used activation function in the world right now. We found that the ReLU action function has good performance for AI based radiative transfer model. The normalization of inputs and true outputs can be important as well. We found that it is very helpful to normalize the true outputs and inputs except for atmospheric water vapor. Atmospheric water vapor changes rapidly and the water vapor values in stratosphere and upper troposphere are very small, which cause the instability when we reconstruct (divided by its standard deviation) water vapor Jacobian values. Therefore, the AI training model uses water vapor profiles (in a unit of g/kg) from ECMWF analysis data without any normalization.

The loss function is also important in the training of AI forward model. The loss function is a method of evaluating how well your algorithm can predict the forward model results. Using the backpropagation method, the gradient of the loss is used to update weights/biases of the AI forward model. The mean squared error (MSE) between the truth and the prediction may be the simplest and most common loss function.

Our previous study, AI_Keras_rt, showed quite accurate in the forward radiance calculations. However, the accuracy of Jacobian derived from the forward model is unsatisfactory to applications such as direct radiance assimilations and physical retrievals. We know that for a given configuration the AI forward model is purely determined by weights/biases coefficients. We also know that the weights/biases are updated from a local minimum of the loss function which contains referenced (CRTM physics model) and the AI predicted forward radiance. We have referenced (CRTM physics model) Jacobian and we can also derive the Jacobian from the forward model without changing the AI configuration. Therefore, we propose a new AI RTM forward model (hereafter AI_Phy_rt) to extend the loss function Eq. (3) to:

$$loss_{new} = loss + \gamma \times loss_{phy} \tag{8}$$

where γ is an empirical parameter and determined by visualizing the performance result, which may balance the two loss functions on the right side of Eq. (8). We use $\gamma = 0.01$ in this study. The physics constraint is the departure of Jacobian:

$$loss_{phy} = \frac{1}{S \times K \times M} \sum\nolimits_{j=1}^{S} \sum\nolimits_{k} c_k \sum\nolimits_{m} (J^{cal}(m,k,j) - J^{true}(m,k,j))^2. \tag{9}$$

As defined in Eq. (3), K is number of Channels and S is batch size. c_k are constant weights depending on channels. The constants c_k are also determined by visualizing the performance result. M represents number of input features. J^{cal} is the analytically calculated Jacobian based on AI_Phy_rt forward model (see Eq. (7)). The calculated Jacobian is fully consistent with the AI_Phy_rt forward model. J^{true} is the CRTM calculation. It may be worth mentioning again that the AI Jacobian is not the output of the forward model AI_Phy_rt.

Since the ANN system can provide multiple solutions that satisfy the accuracy requirement for the radiance output (e.g. 0.2 K), the addition of an extra loss function term related to the Jacobian accuracy forces the ANN training system to choose a solution that satisfies both the accuracy of the forward model and the Jacobian.

We select the sensitivities of input features on radiances for all channels. One can also choose partial channels and partial input features. Same as the forward radiance model, the physics constraint loss function is only a function of input features and weights/biases for the forward model. Therefore, physics constraint helps only in the training process which minimize both forward prediction and the derived model errors without changing the AI model configuration. The approach is analog to the constraint in the cost function used for retrieving geophysical parameters from satellite measurements and direct radiance assimilation for weather forecasting. Their cost function includes the background constraint that the solution needs close to the background value. The advantage of using the background constraint is that the solution is stable and consistent with the background. The disadvantage is that the solution is off from the true value. This off value can be severe if the background is not accurate. Using physics constraints in AI model training does not have the problem. When the sensitivities (physics) are right, the forward model will be more accurate than the forward model training without the constraint.

The training process is only based on the forward model and the physics constraint helps the update of weights/biases toward a physics-based solution. Once the forward training is satisfied, for example the prediction error is comparable to the measurement error, the forward model is ready for applications. In radiative transfer calculations, the Jacobian model, which can easily and analytically be derived from the forward model, is one of the most important components in radiance assimilation in support of weather forecasting and retrieving environmental data records. The Jacobian model is a physical model that explicitly incorporates sensitivities of radiance on geophysical parameters.

3. Data

The data used for training ANN are the foundation of the AI model. We select global data on September 26 of 2021, and January 1, April 1, and July 1 of 2022, which can cover regional and seasonal variations. The selected European Centre for Medium-Range Weather Forecasts (ECMWF) analysis data contain the profiles of temperature, water vapor, and cloud water content as well as the surface variables. We also choose global ECMWF data on October 5, 2022 for the validation test to see if the AI radiative transfer model is still promising even though the training data are three months ahead. These five-days global data are matched up with the ATMS measurements. The criterion for this match-up is that data are within 3 h and 10 km between the ECMWF data and the ATMS measurements. The Joint Polar Satellite System-2 (JPSS-2) satellite was successfully launched on November 10, 2022 and was renamed NOAA-21 after reaching its polar orbit. Like its predecessors on NOAA-20 and Suomi NPP, the cross-track scanning radiometer ATMS onboard NOAA-21 also has 22 channels at frequencies ranging from 23 to 183 GHz, permitting the measurements of atmospheric temperature and moisture profiles under all weather conditions. This paper focuses on the simulation of ATMS brightness temperatures over oceans under clear-sky conditions. Additionally, this research is more about how the new method (physics-based constraint) can overcome the issue in Jacobian calculations derived from the ANN forward model.

The radiative transfer under clear-sky conditions represented in this research describes atmospheric emission and surface emission captured by sensors onboard the satellites. To exclude scatterings under cloud conditions, the ATMS channel 3 brightness temperature differences between the ATMS measurements and the CRTM simulations using the ECMWF data are required to be less than 5 K. About 800,000 data samples over Oceans from the first 4 days are used in the training. About 200,00 data samples on October 5, 2022 are used for the validation.

Table 1 lists the input parameters used in the ANN forward model. We don't use the pressure profile because it is purely determined by the surface pressure in the pressure sigma coordinate. We don't include the CRTM Jacobian values in the table since the Jacobian is not the outputs of this AI based radiative transfer model, although the Jacobian values can be used to improve the training of the AI-based radiative transfer model. The radiance reference values or truths are derived from the CRTM model simulations. CRTM is a physical model that can calculate the forward radiance, tangent linear radiance, adjoint radiance and Jacobian. Even though line-by-line model is more accurate [23] we couldn't use it as it demands enormous computational resources. In addition, CRTM modeling accuracy meets the operational requirements and the model is used operationally for direct radiance assimilation for weather forecasting and in retrieving environmental data records.

Radiative transfer model for calculating satellite measured microwave radiance or radiance at the top of the atmosphere is a widely known methodology. It describes how Planck radiance at a local temperature transmitted and absorbed by molecules in the atmosphere. The CRTM, developed at the joint center for satellite data assimilation in the United States, is an operational radiative transfer model. Its tangent-linear and adjoint as well as Jacobian models are derived from the forward model.

Radiative transfer calculations can be expansive. It takes a day to simulate five-days global data for the ATMS if using the CRTM. It would take more than 100 days to simulate four-days global data if using line-by-line radiative transfer calculations. We use the CRTM model simulations as references to train DL-RT model. The CRTM model is very fast in comparison to line-by-line

Table 1. The input and output parameters used in deep learning radiative model.

Input Variables		Output Variables
Names	Number	Number
Secant of sensor scan angle	1	22 ATMS brightness temperatures
Secant of sensor zenith angle	1	
Cosine of relative azimuth angle	1	
Surface wind speed	1	
Surface temperature	1	
Surface pressure	1	
Atmospheric temperatures	91	
Atmospheric water vapor	91	
Total	188	22

radiative model. But, the CRTM is still not fast enough to process huge satellite data. Only less than 10% of satellite data are assimilated for weather forecasting.

4. Results

We first used the open-source software library Keras. Keras acts as an interface for the open-source TensorFlow library (https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article). The AI training code for the radiative transfer model is written in Python and called AI_Keras_rt hereafter. Keras is user-friendly, modular, and extensible. As mentioned previously, the ANN for ATMS simulations is composed of input layer (188 features) and output layer (22 variables) as well 3 hidden layers with 256, 128, 64 nodes, respectively.

For the training, we need to define some hyperparameters: the number of epochs, batch size, and base learning rate. We use 7,000 epochs, a batch size of 512 and 0.01 for the base learning rate. It was found that more than 7,000 epochs for the training data used in this study improved the accuracy neither for the training data (dependent data) nor for the independent validation data. Using Keras the training for 7,000 epochs and 800,000 samples can be completed within 15 wall hours on a virtual machine equipped with 4 processors Intel Xeon CPU E5-2697A v4 @ 2.60 GHz and 24 GB Mem.

For the dependent data sets, the prediction standard errors for ATMS brightness temperatures are less than 0.2 K and the biases are close to zero. The errors are smaller than ATMT measurement errors (https://www.star.nesdis.noaa.gov/icvs/status_NPP_ATMS.php). For NOAA-20 and NOAA-21 ATMS, the specification value for noise equivalent differential temperature (NEDT) varies between 0.5 K and 3.6 K depending on the channels. The on-orbit performance is better than the specification. For the independent data on October 5, 2022, the prediction standard deviation error is still smaller than 0.2 K although the error is a little larger than the results for dependent data. The standard deviation errors are less than 0.1 K for most of the sounding channels. The biases are pretty small. The results are consistent with that in our previous study [19].

We use the Keras libraries to train the AI_Keras_rt forward model. After the forward model is established, we also use Keras TensorFlow Jacobian model based on the weights/biases from the forward model. TensorFlow provides the module tf.GradientTape for users to calculate Jacobian.

We examine the Keras' Jacobian model for the ATMS instrument. The ATMS is the microwave sounder mainly for acquiring atmospheric profiles of temperatures and water vapors. Its channels 2, 3, 6 and 9 are sensitive to the temperature at the surface, surface boundary layer, 500 hPa, and tropopause. ATMS channels 1, 17, 18, and 20 are sensitive to total column water vapor, water vapor in the surface boundary layer, water vapor at 500 hPa, and water vapor at 200 hPa, respectively. Atmospheric parameters at 500 hPa are very important in global atmospheric circulations. The 500 hPa geopotential is usually called the level of non-divergence. Beneath this level there is a level of convergence and above that level there is a level of divergence so that 500 hPa is the level to look for vertical motions. The geopotential anomaly at 500 hPa is the most important variable for judging the global forecast score.

We choose a case that has a sea surface temperature (290.0 K) and total precipitable water (TPW, 26.2 millimeter) very close to a global mean TPW over oceans [24] [25]. As shown in Fig. 2, the standard (using Keras library) AI_Keras_rt Jacobian values for ATMS channels 2 and 3 display large errors and large oscillations. The two channels are window channels. Their brightness temperatures are mainly contributed by surface emission and reflection. The atmospheric signals at the microwave window channels are relatively weak, and therefore it may be difficult for AI to capture this weak sensitivity. ATMS channels 6 and 9 are sounding channels, rarely affected by the surface. The AI_Keras_rt Jacobian agrees with the CRTM calculated except for the small oscillations. The water vapor Jacobian values above 200 hPa are substantially away from the true values. The AI_Keras_rt water vapor Jacobian calculations have too large uncertainties to applications such as direct radiance assimilations and retrievals of environmental data records. We also examined the Jacobian for very dry and very moist profiles. The behaviors are very similar to the case for Fig. 2.

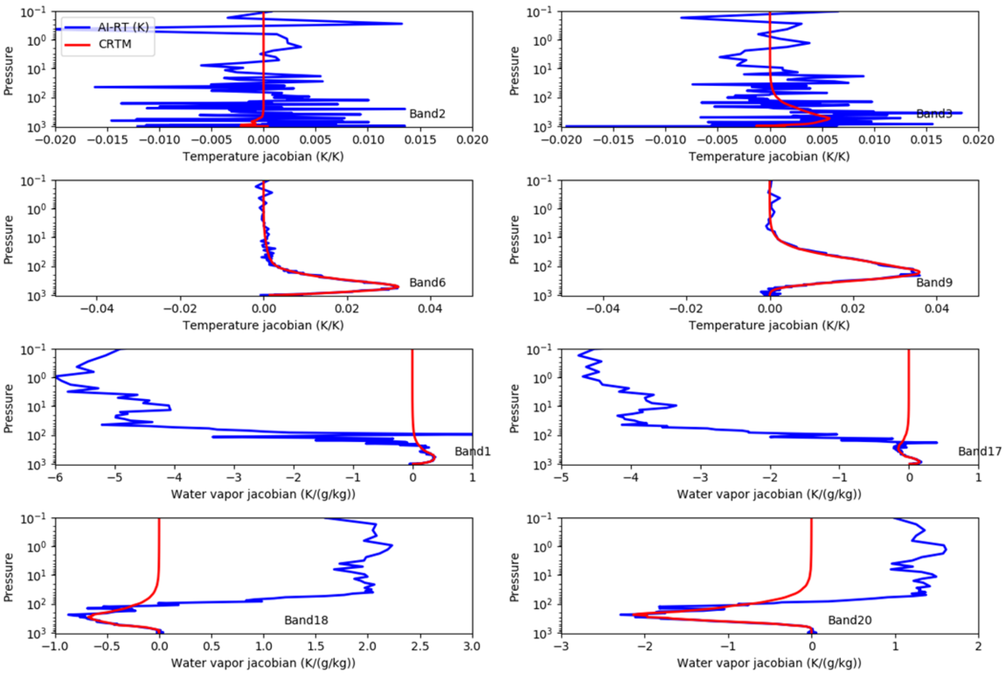

Fig. 2. The upper four panels are for ATMS channels 2, 3, 6, and 9 temperature Jacobian. The lower four panels are for ATMS channels 1, 17, 18, and 20 water vapor Jacobian. The red lines are the CRTM physical model calculations and referred as “true” or reference values. The blue lines are the AI_Keras_rt Jacobian values.

Keras libraries provide fast and optimal training for the forward model. The interface is also friendly for many applications. The libraries can be used in parallel computations with and without GPUs. One can also train Keras models with TPUs. But, it is unable to add a physics constraint to Keras training processes, because customers can build a loss function that only takes the true values and predicted values from a forward model, not the predicted values from a derived model.

We can add the physics constraint to the source codes of the artificial neural network we had developed [26]. Our training libraries were written in FORTRAN and were successfully applied to a non-linear problem in calculating downward long-wave radiation from Special Sensor Microwave/Imager of Defense Meteorological Satellite Program F08. The FORTRAN AI libraries have basic functions that Keras/TensorFlow has. We adopt the AI model for radiative transfer (RT) calculations, therefore it is called AI_Phy_rt model in this study.

We first check the radiative transfer model accuracy using AI_Phy_rt without any physics constraint. The ANN configuration and the training data as well as the validation data are identical to what we used for AI_Keras_rt. Figure 3 shows the comparisons between CRTM truths and the AI_Phy_rt predictions. AI_Phy_rt doesn’t have GPU capability and only partial codes use openMP parallel computations now. Although the AI_Phy_rt training is about four times slower than AI_Keras_rt, AI_Phy_rt prediction accuracy for both training and validation data is comparable to the prediction accuracy what AI_Keras_rt obtained. As one can see from Fig. 3, the AI_Phy_rt prediction accuracy for training data is similar to the accuracy for the validation data which are 3 months after the training data. Our results suggest that AI_Phy_rt model is stable and comparable to AI_Keras_rt.

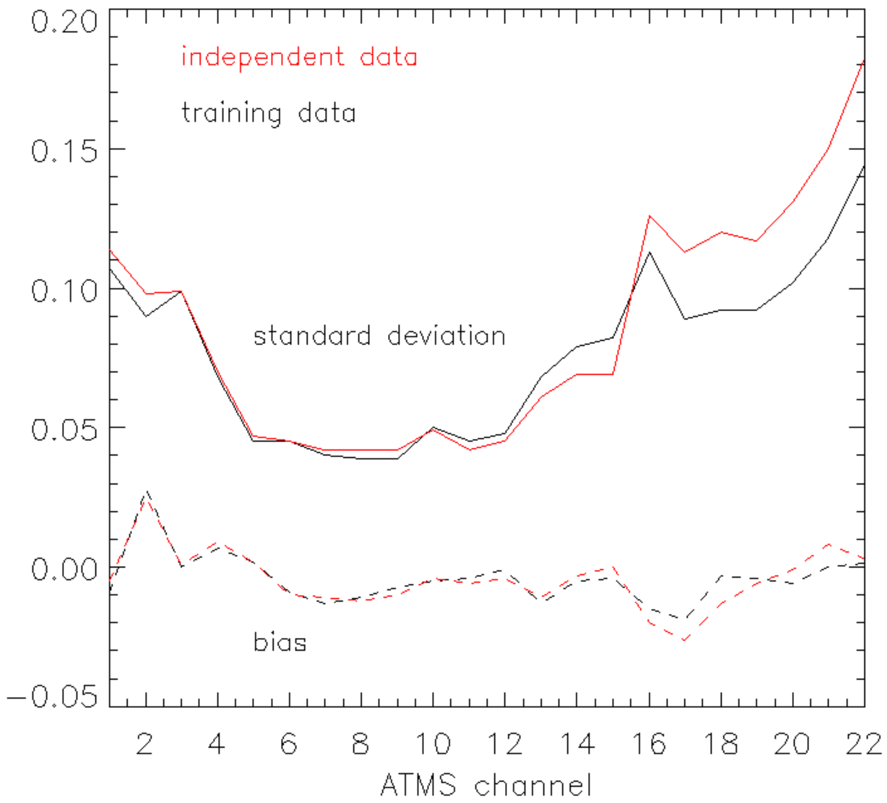

Fig. 3. Standard deviations (solid lines) and biases (dashed lines) of AI_rt brightness temperatures for ATMS 22 channels. The black lines represent the difference between the CRTM truths and the AI_Phy_rt predictions for the training data between September 26, 2021 and July 1, 2022. The red lines are for the validation data on October 8, 2022.

With the physics constraint in AI_Phy_rt, the weights/biases are updated by fitting the forward predictions with forward truths and keeping the right sensitivities such as temperature and water vapor Jacobian profiles as the CRTM true Jacobian in the training process. The inline prediction model is the same regardless of whether using AI_Keras_rt or using AI_Phy_rt training, because the format and the size of the AI model weights/biases are the same.

Figure 4 plots the loss function for the forward model and for the physics constraint part, respectively. During our artificial neural network (AI_Phy_rt) training, we only save the weights which have better performance for forward radiance predictions and Jacobian predictions in the derived model. Since the outputs are normalized (see Eq. (4b)), the loss function for the forward part here is not a root mean square error between true and AI predicted brightness temperatures. However, the loss function values represent the convergence of the AI training since the smaller of the loss function, the smaller of the root mean square error. The loss function for the forward part decreases smoothly.

In Table 2, we compared the CPU timing in seconds for 100,000 profiles and 22 ATMS channels' brightness temperatures among the CRTM model, AI_Keras_rt (Python), and AI_Phy_rt (FORTRAN). Keras libraries don't have the function for calculating the adjoint radiance. We compared the CPU timing in two computational modes: all profiles in one calculation and each calculation for one profile since both modes are used in the current applications. AI_Keras_rt (Python) is very efficient for all profiles in one calculation. The result indicates that AI_Keras_rt

Optics EXPRESS

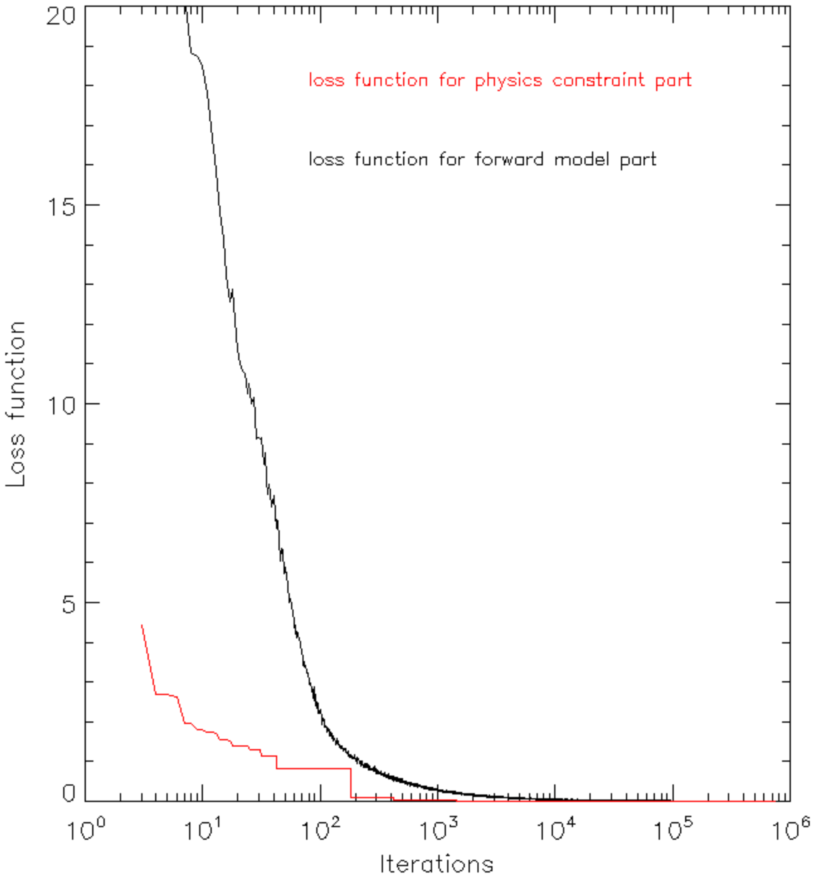

Fig. 4. The loss function contains both the part for the forward model and the part for physics constraint. The back line is for the forward part and the red line is for the physics constraint part.

(Python) achieves high performance in parallel computations. However, AI_Kreas_rt is extremely slow for each profile per calculation. AI_Phy_rt (FORTRAN) is also efficient for all profiles in one calculation. For the one profile in each calculation, AI_Phy_rt is considerably faster than AI_Keras_rt. One may notice from Table 2 that the CPU time for the CRTM Jacobian calculation is about 2.5 times that for the CRTM forward calculation. However, the AI-based Jacobian calculation is 30 times slower than the AI-based forward calculation. In AI forward calculation of one profile, we have to deal with the computation of a matrix multiplication by a vector. On the other hand, AI Jacobian calculation involves a matrix multiplication by another matrix. The CPU time in AI Jacobian calculation is proportional to the number of channels.

The AI_Phy_rt training is very slow since the AI Jacobian model calculations are very CPU time consuming. To use Jacobian to improve the forward model weights/biases update, the Jacobian prediction is needed. AI_Keras_rt training took about 15 hours of wall time for 7,000 epochs. AI_Phy_rt training would demand 1050 hours of wall time for 7,000 epochs. Fortunately, AI_Phy_rt training learns faster than AI_Keras_rt. Using 400 epochs, AI_Phy_rt can achieve comparable accuracy in the forward radiance prediction as AI_Keras_rt uses 7,000 epochs. Using AI_Phy_rt in the forward prediction, the biases are less than 0.03 K and the standard deviations are smaller than 0.2 K. The standard deviation for the most sounding is also less than 0.1 K.

Table 2. CPU timing (seconds) for 100,000 profiles and 22 ATMS channels in forward and Jacobian calculations. The CPU timing for the CRTM model and AI_Phy_rt are listed as well. Keras libraries don't have the function for calculating the adjoint radiance yet.

	Forward		Jacobian		Adjoint	
	Profile Set	Per Profile	Profile Set	Per Profile	Profile Set	Per Profile
CRTM	96.12	102.61	227.31	242.67	227.11	242.32
AI_Keras_rt	0.63	135.20	18.40	17,521		
AI_Phy_rt	0.79	1.87	32.36	32.73	1.88	4.36

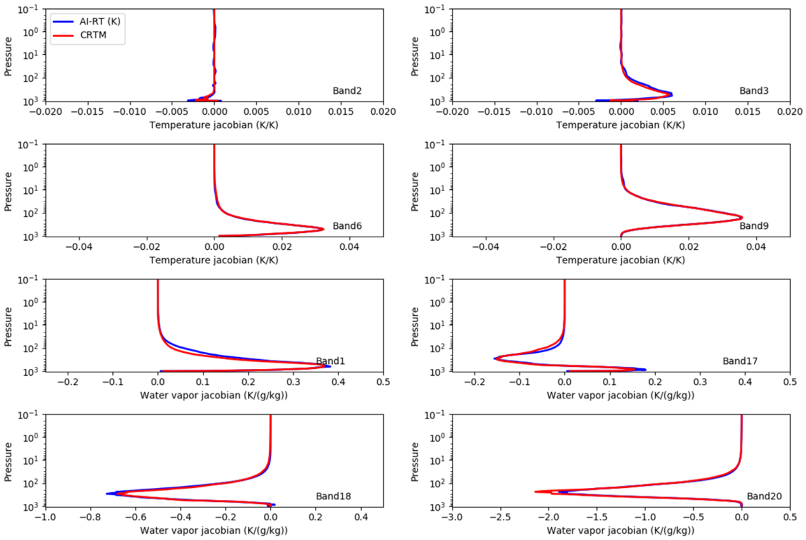

Fig. 5. The upper four panels are for ATMS channels 2, 3, 6, and 9 temperature Jacobian. The lower four panels are for ATMS channels 1, 17, 18, and 20 water vapor Jacobian. The red lines are the CRTM physical model calculations and referred as "true" or reference values. The blue lines are the AI_Phy_rt Jacobian.

For this study, AI_Phy_rt training uses 400 epochs. On the other side, the AI_Phy_rt Jacobian accuracy is dramatically improved.

Figure 5 is similar to Fig. 2 except for using AI_Phy_rt. AI_Phy_rt forward and Jacobian prediction models are the same as AI_Keras_rt except for using different weights/biases. The red line represents the CRTM calculation. The blue line represents AI_Phy_rt calculation. AI_Phy_rt training follows the radiative transfer physics so that the derived Jacobian achieve better accuracy to the temperature and water vapor sensitivities on the ATMS brightness temperatures.

We also examined the AI forward and Jacobian calculations for very dry and very humid cases. The results and conclusions are very similar to the global mean case above.

5. Discussion and conclusion

We have studied two types of AI-based radiative transfer models: AI_Keras_rt a python code embedded with Keras libraries and AI_Phy_rt a FORTRAN code embedded with Artificial Neural Network [26]. The two algorithms achieve high accuracy for the forward radiance calculations. However, AI_Keras_rt encounters difficulty in delivering accurate Jacobians those are required by direct radiance assimilation in support of weather forecasting and retrieving environmental data records from satellite measurements. The AI_Keras_rt displays oscillation for the ATMS window (surface) channels and very large uncertainties for water vapor channels close to the strong water vapor absorption line at 183.31 GHz (see Fig. 2).

The forward and Jacobian need to be very consistent to avoid unstable solutions because direct radiance assimilation and retrievals involve multiple iterations. The consistency can be evaluated by comparing finite difference (partial derivative) from two forward calculations against the Jacobian. The partial derivative from AI forward model won't be consistent with the Jacobian from another AI model because of the uncertainty in AI prediction model. One may build one AI model containing both forward radiance and Jacobian rin its outputs, but the finite difference of the forward radiance won't be consistent with the Jacobian prediction either, because the prediction error for the forward radiance doesn't correspond to the prediction error for the Jacobian due to random error in the prediction model. Therefore, we can only build a single AI forward model and use a derived model for the Jacobian.

Using the physics constraint in AI_Phy_rt training, we update the forward model weights/biases with meaningful physics (sensitivities of input features on radiances) and overcome the problems in the derived Jacobian model. The AI_Phy_rt Jacobian calculation is pretty accurate for applications (see Fig. 5, please note the different x-axis scale).

We realized that AI_Keras_rt training is considerably faster than AI_Phy_rt for a given number of epochs. However, AI_Phy_rt converges faster. In total effect, AI_Phy_rt training still demands four times more CPU time to achieve a comparable accuracy as AI_Keras_rt in the forward radiance model. Keras libraries show high computational efficiency in model training. AI_Phy_rt can deliver accurate Jacobian. We plan to add our physics constraint to AI_Keras_rt training.

It is noticed that one calculation for multiple profiles is very efficient than one calculation for one profile. In particular for using AI_Keras_rt Jacobian prediction model for 100,000 profiles and 22 ATMS channels, one calculation for one profile is about 1,000 times slower than one calculation for multiple profiles. It would be a good time to change to one calculation for multiple profiles in applications if one can.

We also noticed that AI Jacobian calculations demand much more CPU time than the AI forward radiance calculations. In the AI Jacobian calculations, the CPU time is proportional to the number of channels (see Table 2). For hyperspectral sensors having thousands of channels, the AI Jacobian computational time would be more than that of the CRTM model. AI_Phy_rt adjoint radiance calculation is very fast. The CPU time ratio of the adjoint to forward radiance computation is about 2.5. The AI adjoint radiance calculation is much faster than the AI Jacobian calculation. We also plan to study whether we can use AI adjoint radiance in the NOAA Microwave Integrated Retrieval System (MiRS) [27] [28].

Funding. National Oceanic and Atmospheric Administration (NA20OAR4600287).

Acknowledgments. We thank our internal reviewer very much for the edit and valuable comments. We thank two anonymous reviewers for their very valuable comments and suggestions. The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official NOAA or U.S. Government position, policy, or decision.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. X. Liu, W. Wu, B. A. Wielicki, Q. Yang, S. H. Kizer, X. Huang, X. Chen, S. Kato, Y. L. Shea, and M. G. Mlynczak, "Spectrally Dependent CLARREO Infrared Spectrometer Calibration Requirement for Climate Change Detection," J. Clim. **30**(11), 3979–3998 (2017).
2. R. Saunders, P. Rayer, P. Brunel, A. von Engeln, N. Bormann, L. Strow, S. Hannon, S. Heilliette, X. Liu, F. Miskolczi, Y. Han, G. Masiello, J.-L. Moncet, G. Uymin, V. Sherlock, and D. S. Turner, "A comparison of radiative transfer models for simulating Atmospheric Infrared Sounder (AIRS) radiance," J. Geophys. Res. **112**(D1), D01S90 (2007).

3. Q. Liu and S. Boukabara, "Community Radiation Transfer Model (CRTM) Applications in Supporting the Suomi National Polar-Orbiting Partnership (SNPP) Mission validation and Verification," Remote Sen. Environ. **140**, 744–754 (2014).
4. O. Coopmann, V. Guidard, N. Fourrié, B. Josse, and V. Marécal, "Update of Infrared Atmospheric Sounding Interferometer (IASI) channel selection with correlated observation errors for numerical weather prediction (NWP)," Atmos. Meas. Tech. **13**(5), 2659–2680 (2020).
5. K. Arai, "Data fusion between microwave and thermal infrared radiometer data and its application to skin sea surface temperature, wind speed and salinity retrievals," Int. J. Adv. Comput. Sci. Appl. **4**(2), 239–244 (2013).
6. K. Arai, "Visualization of learning process for back propagation Neural Network clustering," Int. J. Adv. Comput. Sci. Appl. **4**(2), 234–238 (2013).
7. X. Liang and Q. Liu, "Applying Deep Learning to Clear-Sky Radiance Simulation for VIIRS with Community Radiative Transfer Model—Part 1: Develop AI-Based Clear-Sky Mask," Remote Sens. **13**(2), 222 (2021).
8. Z. Zhang, H. Wang, F. Xu, and Y. Q. Jin, "Complex-Valued Convolutional Neural Network and Its Application in Polarimetric SAR Image Classification," IEEE Trans. Geosci. Remote Sensing **55**(12), 7177–7188 (2017).
9. M. Maskey, R. Ramachandran, I. Gurung, B. Freitag, A. Kaulfus, D. Bollinger, D. J. Cecil, and J. Miller, "Deepti: Deep-Learning-Based Tropical Cyclone Intensity Estimation System," IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing **13**, 4271–4281 (2020).
10. C. K. Sønderby, L. Espeholt, J. Heek, M. Dehghani, A. Oliver, T. Salimans, S. Agrawal, J. Hickey, and N. Kalchbrenner, "Metnet: A neural weather model for precipitation forecasting," arXiv, arXiv:2003.12140 (2021).
11. J. A. Weyn, D. R. Durran, R. Caruana, and N. Cresswell-Clay, "Subseasonal forecasting with a large ensemble of deep-learning weather prediction models," J. Adv. Model. Earth Syst. **13**(7), 1 (2021).
12. B. D. Bue, D. R. Thompson, S. Deshpande, M. Eastwood, R.O. Green, V. Natraj, T. Mullen, and M. Parente, "Neural network radiative transfer for imaging spectroscopy," Atmos. Meas. Tech. **12**(4), 2567–2578 (2019).
13. F. Aires, C. Prigent, and W. B. Rossow, "Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3 Network Jacobians," J Geophys. Res. **109**(D10), D10305 (2004).
14. V. M. Krasnopolsky, M. S. Fox-Rabinovitz, Y. T. Hou, S. J. Lord, and A. A. Belochitski, "Accurate and Fast Neural Network Emulations 395 of Model Radiation for the NCEP Coupled Climate Forecast System: Climate Simulations and Seasonal Predictions," Mon. Weather Rev. **138**(5), 1822–1842 (2009).
15. F. Chevallier, F. Chéruy, N. A. Scott, and A. Chédin, "A neural network approach for a fast and accurate computation of a longwave radiative budget," J. Appl. Meteorol. **37**(11), 1385–1397 (1998).
16. F. Chevallier and J. F. Mahfouf, "Evaluation of the Jacobians of infrared radiation models for variational data assimilation," J. Appl. Meteorol. **40**(8), 1445–1461 (2001).
17. Y. Yao, X. Zhong, Y. Zheng, and Z. Wang, "A physics-incorporated deep learning framework for parameterization of atmospheric radiative transfer," J. Adv. Model. Earth. Syst. **15**(5), e2022MS003445 (2023).
18. S. Mishra and R. Molinaro, "Physics informed neural networks for simulating radiative transfer," J. Quant. Spectrosc. Radiat. Transfer **270**, 1 (2021).
19. X. Liang, K. Garrett, Q. Liu, E. S. Maddy, K. Ide, and S. Boukabara, "A Deep-Learning-Based Microwave Radiative Transfer Emulator for Data Assimilation and Remote Sensing," IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing **15**, 8819–8833 (2022).
20. D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv, arXiv:1412.6980v9 (2014).
21. F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," Psychological Review **65**(6), 386–408 (1958).
22. E. Kim, C.-H. J. Lyu, K. Anderson, R. Vincent Leslie, and W. J. Blackwell, "S-NPP ATMS instrument prelaunch and on-orbit performance evaluation," J. Geophys. Res. **119**(9), 5653–5670 (2014).
23. S. A. Clough, M. J. Iacono, and J.-L. Moncet, "Line-by-line calculation of atmospheric fluxes and cooling rates: Application to water vapor," J. Geophys. Res. **97**(D14), 15761–15785 (1992).
24. Y. K. Lee, C. Grassotti, Q. Liu, S. Y. Liu, and Y. Zhou, "In-depth evaluation of MiRS total precipitable water from NOAA-20 ATMS using multiple reference data sets," Earth and Space Science **9**(2), e2021EA002042 (2022).
25. M. Menne, "Global Long-term Mean Land and Sea Surface Temperatures," *National Climate Data Center* (2000).
26. Q. Liu, C. Simmer, and E. Ruprect, "Estimating longwave net radiation at sea surface from the Special Sensor Microwave/Imager (SSM/I)," J. Appl. Meteorol. **36**(7), 919–930 (1997).
27. S. Boukabara, K. Garrett, W. Chen, F. Iturbide-Sanchez, C. Grassotti, C. Kongoli, R. Chen, Q. Liu, B. Yan, F. Weng, R. Ferraro, T.J. Kleespies, and H. Meng, "MiRS: An All-Weather 1DVAR Satellite Data Assimilation and Retrieval System," IEEE Trans. Geosci. Remote Sensing **49**(9), 3249–3272 (2011).
28. E. S. Maddy and S. A. Boukabara, "MIIDAPS-AI: An Explainable Machine-Learning Algorithm for Infrared and Microwave Remote Sensing and Data Assimilation Preprocessing - Application to LEO and GEO Sensors," IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing **14**, 8566–8576 (2021).