An Open-Box Physics-Based Neural Network for Modeling Shortwave Radiative Transfer

Henry Schneiderman, 2026

Schedule

4/2
- Presentation: Will
- Pizza: Andy

4/16
- Presentation: Zilu
- Pizza:

4/30
- Presentation: Celeste
- Pizza:

5/14
- Presentation:
- Pizza: Eliot

5/28
- Presentation: Dominik
- Pizza:

Motivation

- Traditional radiative transfer models can be slow and take up a lot of time in NWP models
- ML NWP models typically have one model learn everything from dynamics to microphysics (e.g., GraphCast)
  - Desire to separate radiation from other processes for climate change simulations
- ML models are usually “black boxes”

Paper overview

- Designed a neural network to predict shortwave (as opposed to longwave) radiative transfer
- Meant to be a drop in replacement for existing radiative transfer models
- Explicitly designed to be “open box”: neural network weights have some physical meaning rather than just being free, uninterpretable parameters

Design of a shortwave radiative transfer model

- **Transmissivity**: what fraction of incoming direct radiation is directly transmitted through an atmospheric layer
- **Scattering**: what happens to the fraction of incoming direct radiation that is NOT directly transmitted through an atmospheric layer (absorbed, reflected, diffused)
- **Multi-reflection (up) + radiation propagation (down)**: interreflection of radiation between multiple atmospheric layers

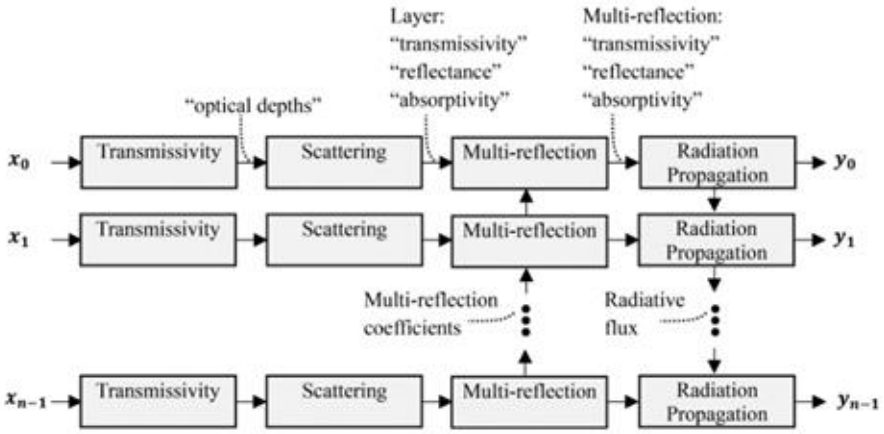
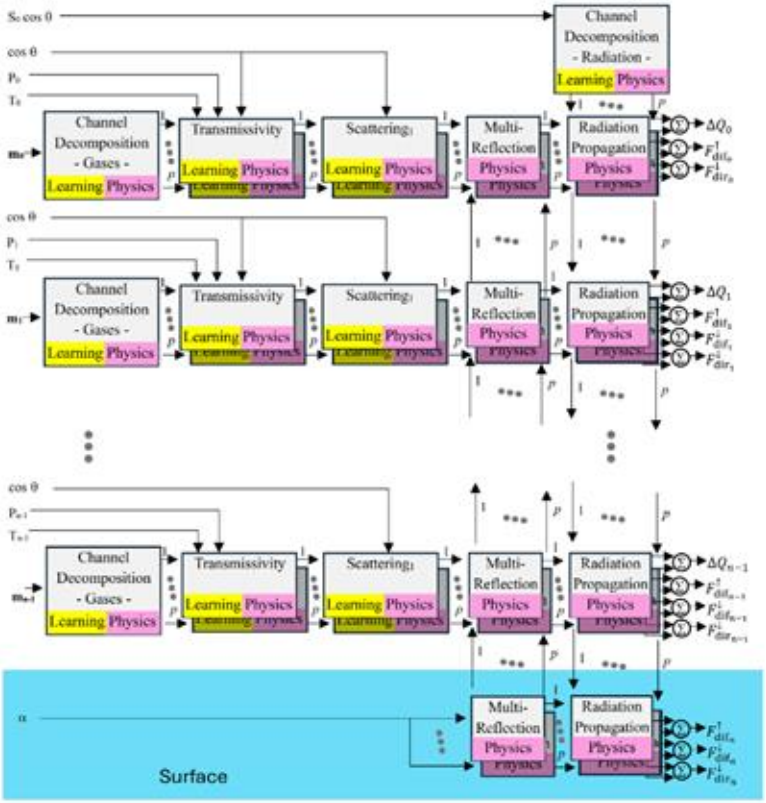


FIG. 1. A hypothetical open-box neural network for shortwave RT. Each rectangle represents a physical subprocess (e.g., transmissivity), modeled by hardcoded physics-based equations and/or standard neural network elements. Arrows represent each component's physically defined input and output variables (e.g., optical depth).

Absolute monster of a figure



Taking a step back: inputs and outputs

Inputs:

- Each vertical atmospheric column receives: θ (solar zenith angle), α (surface albedo), and S_0 (solar constant of 1361 W/m^2)
- Each atmospheric layer k receives: T_k (temperature), p_k (pressure), m_k (mass of eight constituents: vapor, liquid, and ice H_2O, O_3, CO_2, N_2O, CH_4, O_2
- Source: ECMWF's Copernicus Atmosphere Monitoring Service (CAMS)

Outputs:

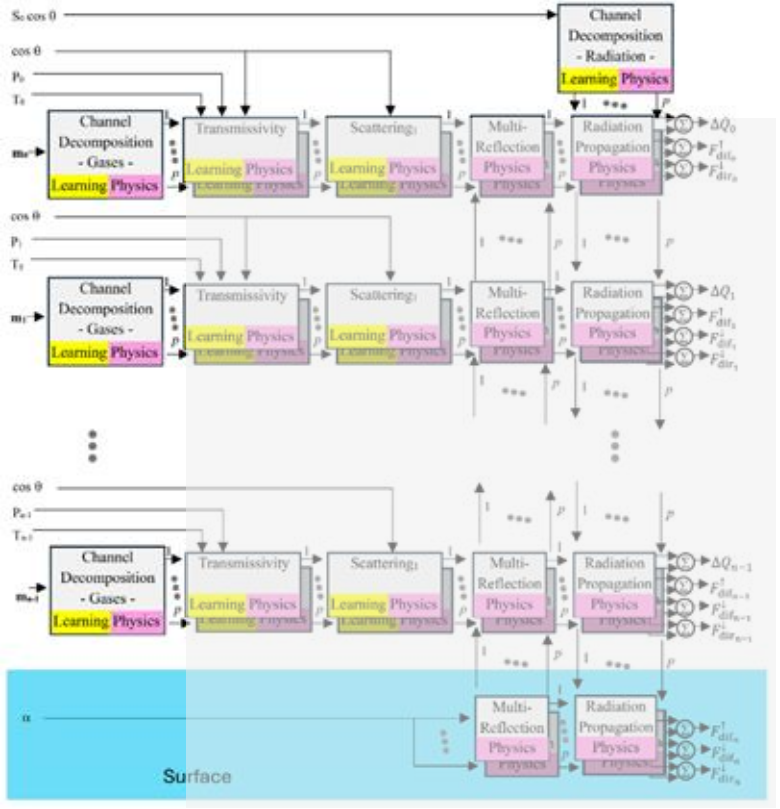
Given an input atmospheric column, we predict the following values for each atmospheric layer k:

- ΔQ_k—The heating rate (K day^{-1})
- $F^{\downarrow}_{\mathrm{dir}_k}$—The direct radiative flux entering the layer from above (W m^{-2})
- $F^{\downarrow}_{\mathrm{dir}_k}$—The diffuse radiative flux entering the layer from above (W m^{-2})
- $F^{\downarrow}_{\mathrm{dir}_k}$—The diffuse radiative flux exiting the layer from above (W m^{-2})

We also predict the fluxes flowing into and out of the surface: $F^{\downarrow}_{\mathrm{dir}_{k=n}}$, $F^{\downarrow}_{\mathrm{dif}_{k=n}}$, $F^{\uparrow}_{\mathrm{dif}_{k=n}}$.

Source: ecRAD (ECMWF's radiative transfer model) with some assumptions

Channel decomposition of radiation and gases



Channel decomposition of radiation and gases

Split radiation in p components: learned weights are c_i

$$\mathbf{F}^{\downarrow}_{\mathrm{dir}_0} = S_0 \cos\theta \ \mathbf{z}, \qquad z_i = \frac{\exp(c_i)}{\sum_{j=1}^{p} \exp(c_j)}.$$

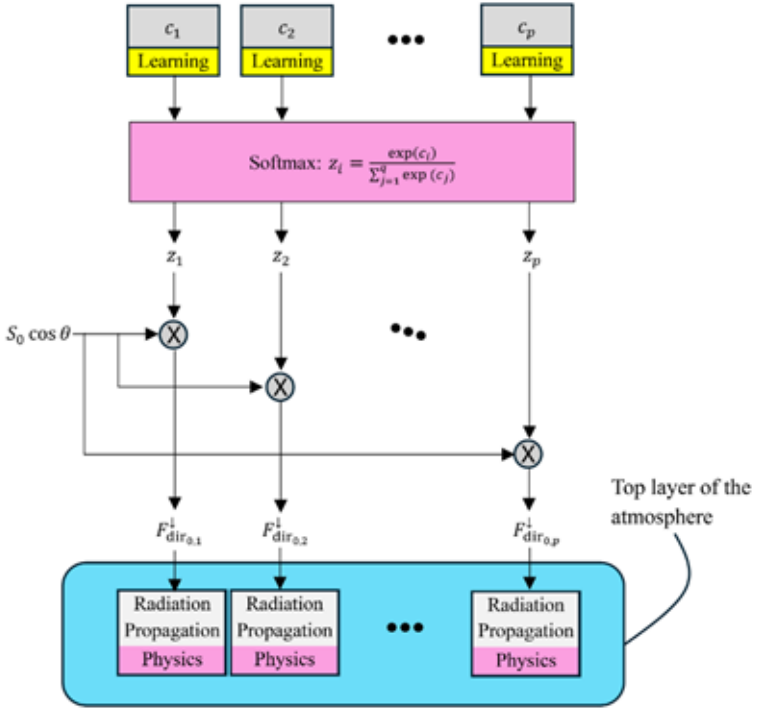


FIG. 4. Channel decomposition of radiation. This component learns a set of weights, $c_1, \ldots, c_p$, that it transforms into $z_1, \ldots, z_p$ through a softmax function. This structure then hardcodes the multiplication of each z_i with the total incoming solar flux $S_0 \cos\theta$, creating the split of incoming solar flux $F^{\downarrow}_{\mathrm{dir}_{0,1}}, \cdots, F^{\downarrow}_{\mathrm{dir}_{0,p}}$.

Channel decomposition of radiation and gases

Split radiation in p components: learned weights are c_i

$$\mathbf{F}^{\downarrow}_{\mathrm{dir}_0} = S_0 \cos\theta \ \mathbf{z}, \qquad z_i = \frac{\exp(c_i)}{\sum_{j=1}^{p} \exp(c_j)}.$$

Learn the contribution of each gas: learned weights are $w_{i,j}$

$$\eta_{k,i,j} = m_{k,j} \exp(w_{i,j}),$$

k is the atmospheric layer and j is the constituent

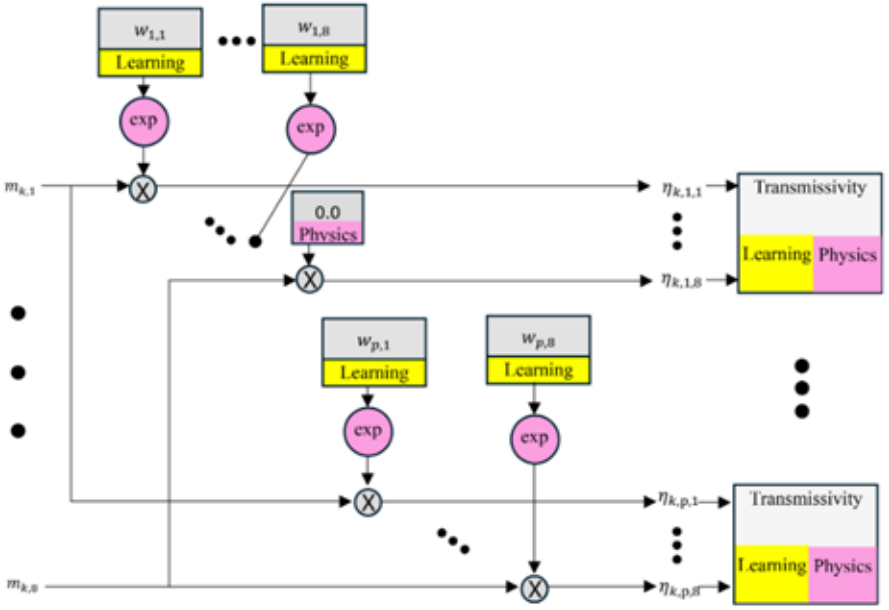
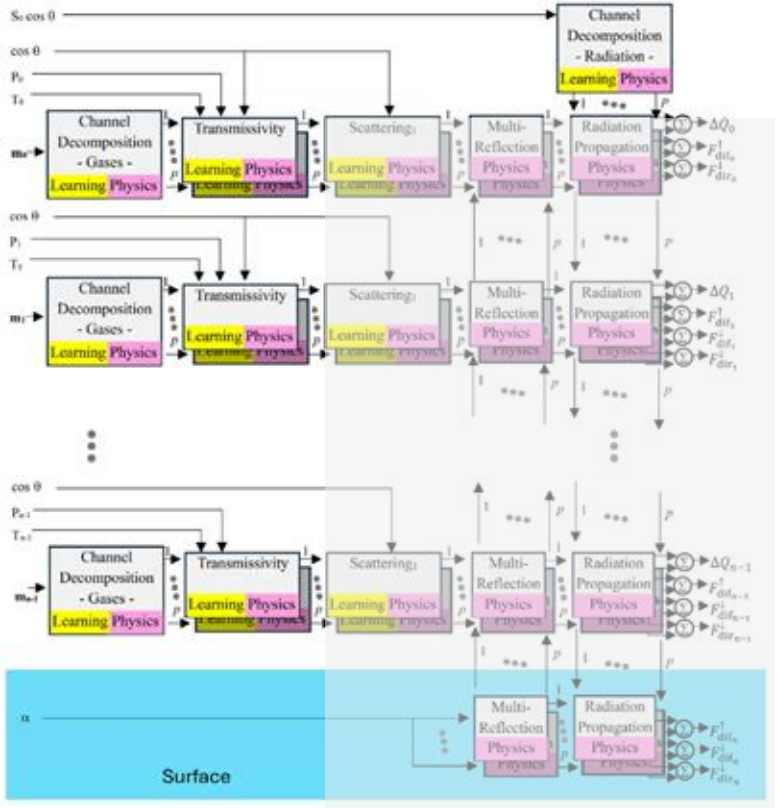


FIG. 5. Channel decomposition of gases. The network learns a set of weights, $w_{1,1}, \ldots, w_{p,8}$, that pass through exponential activation functions. This structure then hardcodes the product of the exponential of each weight with the appropriate input mass, $m_{k,j}$, giving a pseudo mass $\eta_{k,i,j}$. The physics box with 0.0 illustrates an atmospheric constituent, $j = 8$, that is hardcoded to have no influence on a particular channel, $i = 1$.

Transmissivity: how much radiation makes it through?



Transmissivity: how much radiation makes it through?

Inputs: pseudo-masses from channel decomposition of gases

Learned weights: k (mass extinction coefficient) → related to optical depth via Beer's law

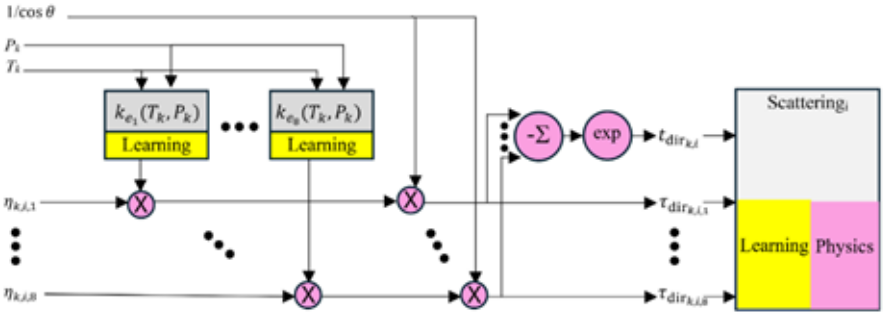
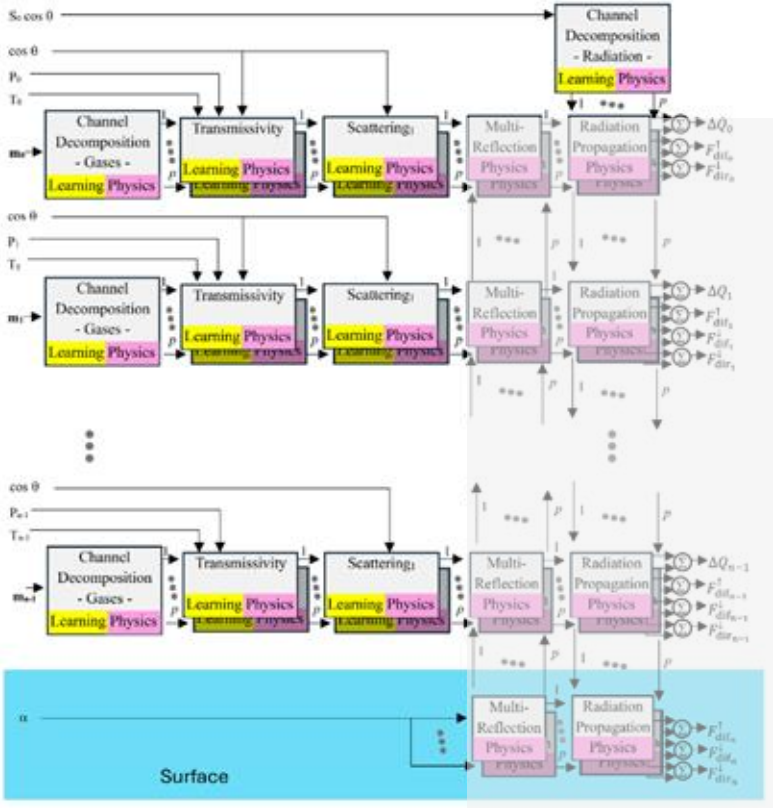


FIG. 6. Transmissivity component. Each horizontal path computes an optical depth by hardcoding the product of a pseudo mass with its corresponding mass extinction coefficient and with the inverse of the cosine of the sun's zenith angle. Neural network modules represent each mass extinction coefficient's dependence on temperature and pressure $k_{e_j}(T_k, P_k)$. The structure also hardcodes the computation of the direct transmittance coefficient, $t_{\mathrm{dir}_{k,i}}$, by taking the exponential of the negative summation of the individual optical depths.

Scattering: what happens to radiation that didn't make it through?



Scattering: what happens to radiation that didn't make it through?

Three possibilities: absorbed, reflected (upward), or transmitted (downward)

Inputs: optical depths of the atmospheric constituents + zenith angle

Output: fraction that is absorbed, reflected, or transmitted

Note: this component is fully machine learned

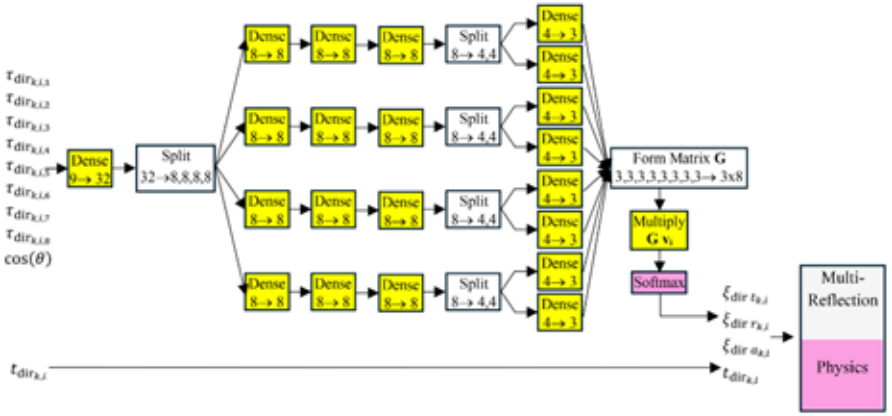


FIG. 7. Scattering component. Each dense unit is a single fully connected neural network layer with a ReLU activation function, where $m \rightarrow n$ indicates m inputs and n outputs, giving $m\,n$ weights. A matrix multiplication unit multiplies the 3×8 matrix, $\mathbf{G}$, by an 8×1 column vector, $\mathbf{v}_i$, of weights, which is uniquely learned for each channel i. The three coefficients resulting from the matrix multiplication then pass through a softmax function. The columns of $\mathbf{G}$ could be thought of as basis vectors for forming these output coefficients. These basis vectors do not need to be explicitly coupled, which allows their computation to be split into the separate horizontal paths. Doing so avoids the learning of unnecessary weights, reducing susceptibility to overfitting.

Multi-reflection and radiation propogation

Purely physical model

Inputs: scattering fractions (absorbed, reflected, transmitted) from scattering module

Outputs: Upward and downward fluxes → used to compute heating rates

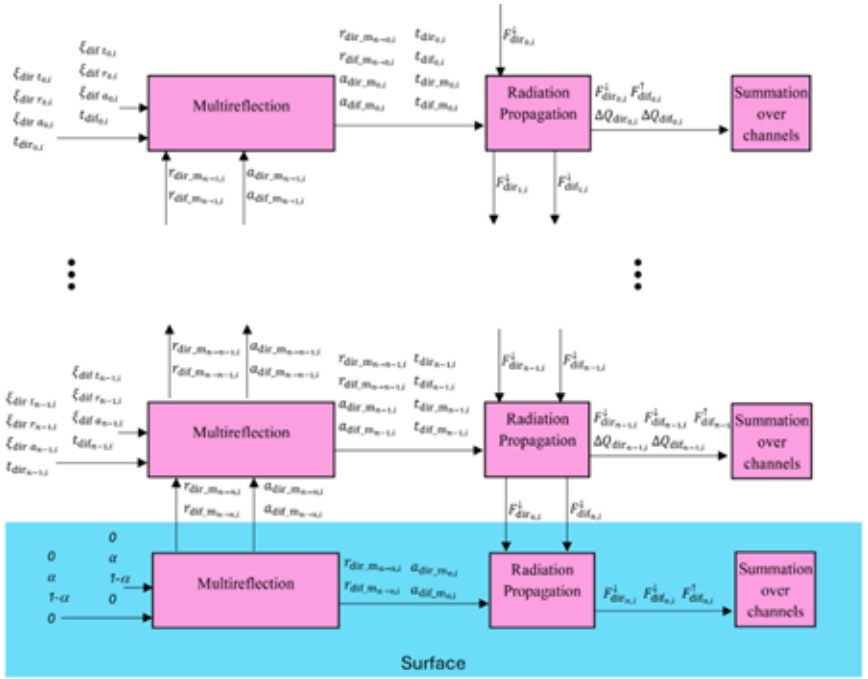


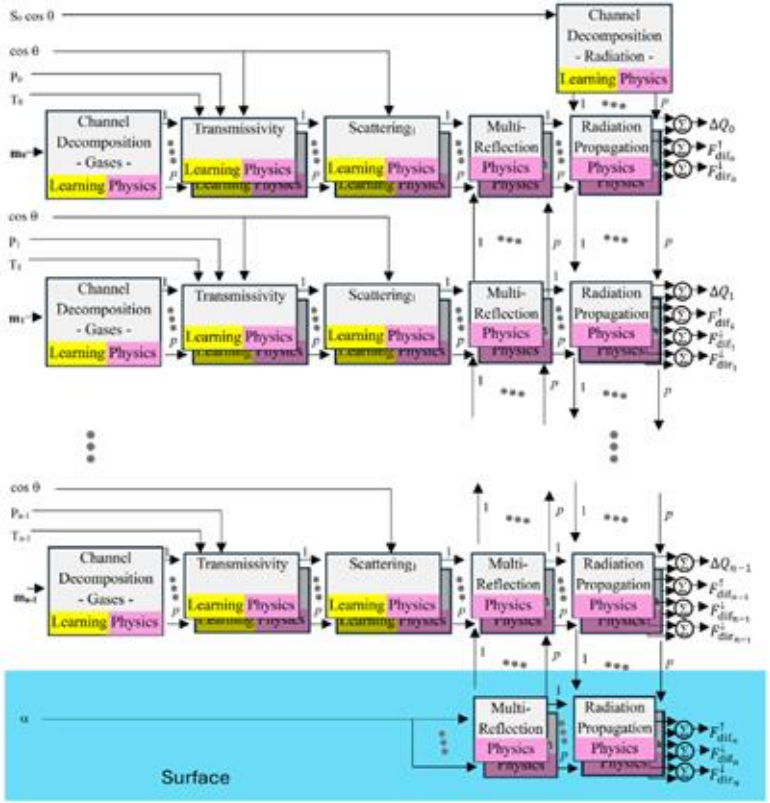
FIG. 8. Dataflow through the multireflection and radiation propagation components for a single channel i.

Loss function

Normal neural network loss function would penalize output variables equally

Proposed loss function which breaks error into 4 terms: direct flux, diffuse flux, direct extinction, diffuse heating rate

$$J_{\text{open\_box}} = d_1 J_{\text{dir\_flux}} + d_2 J_{\text{dif\_flux}} + d_3 J_{\text{dir\_ext}} + d_4 J_{\text{dif\_heat}}, \quad (17)$$



Discussion

Key assumptions of model:

- Independent column approximation, fixed solar constant, homogenous clouds

Longwave radiation next?