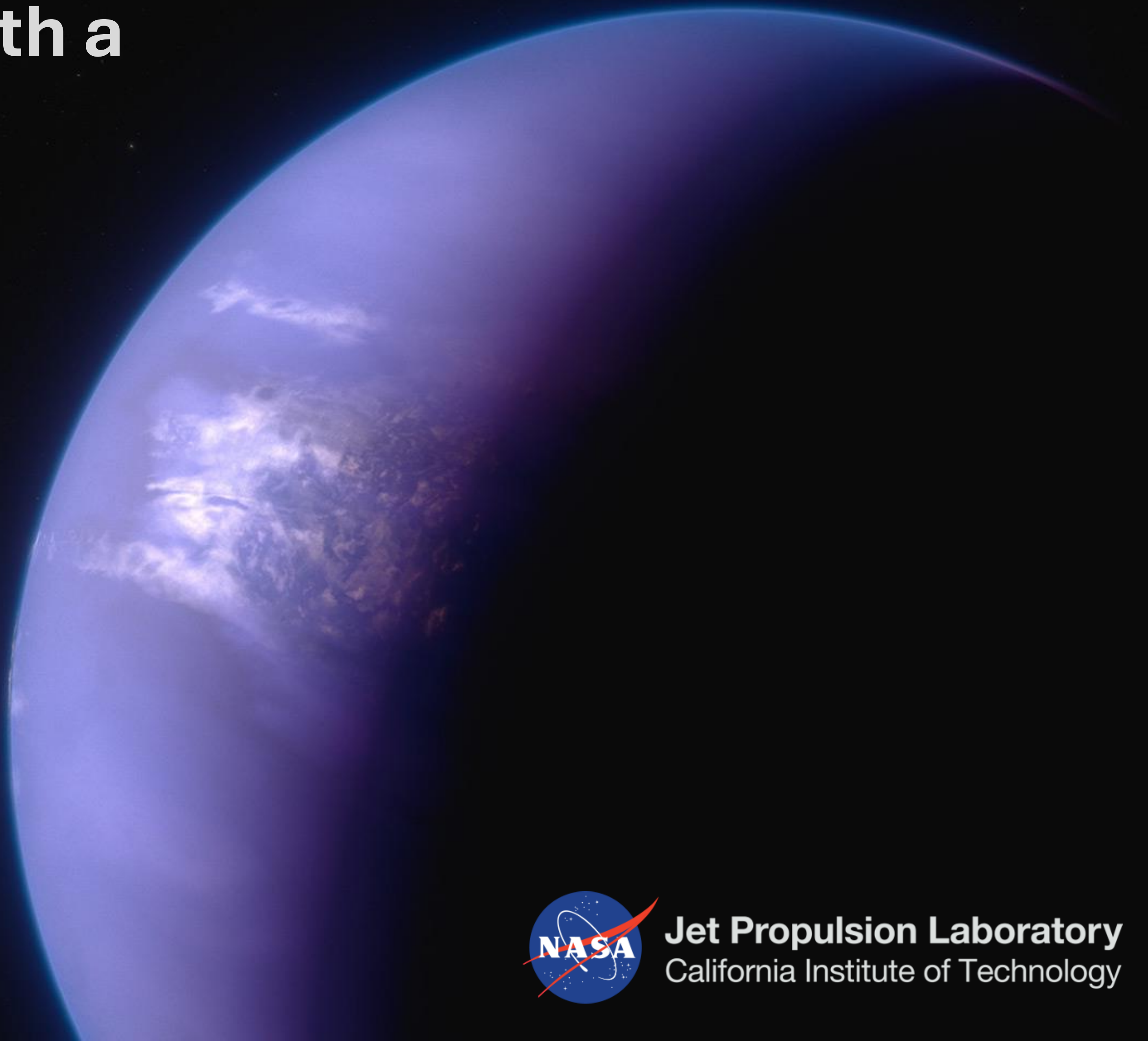
Picture Credit: ESA

# Accelerating Radiative Transfer for by Orders of Magnitude with a Machine Learning Model



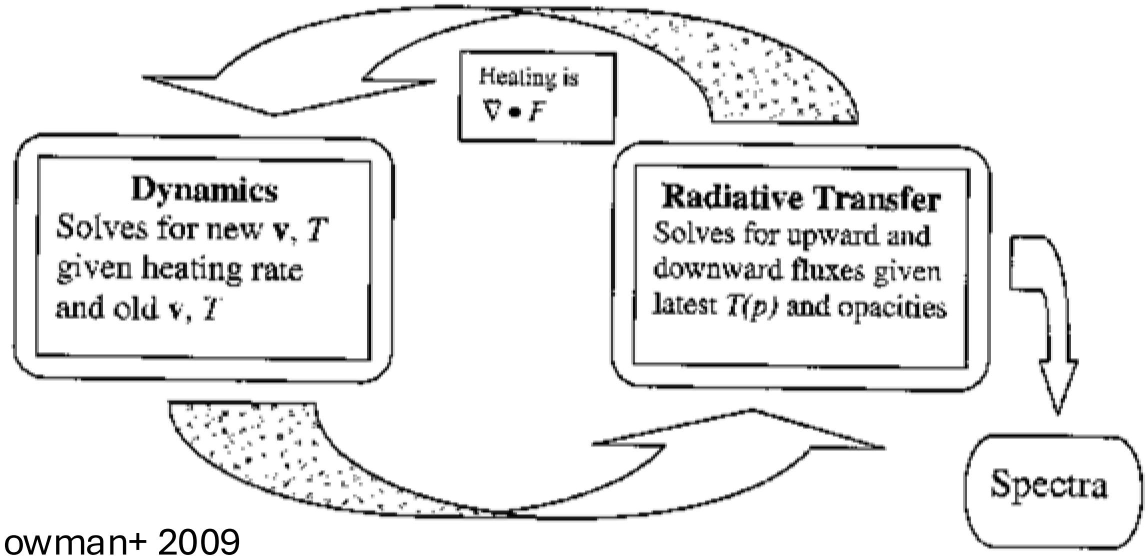

Isaac Malsky
NASA Jet Propulsion Laboratory,
California Institute of Technology
*NASA Postdoctoral Program*
*NPP Mentor: Tiffany Kataria*

*Collaborators:*
*Tiffany Kataria (NASA JPL)*
*Natasha Batalha (NASA AMES)*
*Matthew Graham (Caltech)*


Jet Propulsion Laboratory
California Institute of Technology

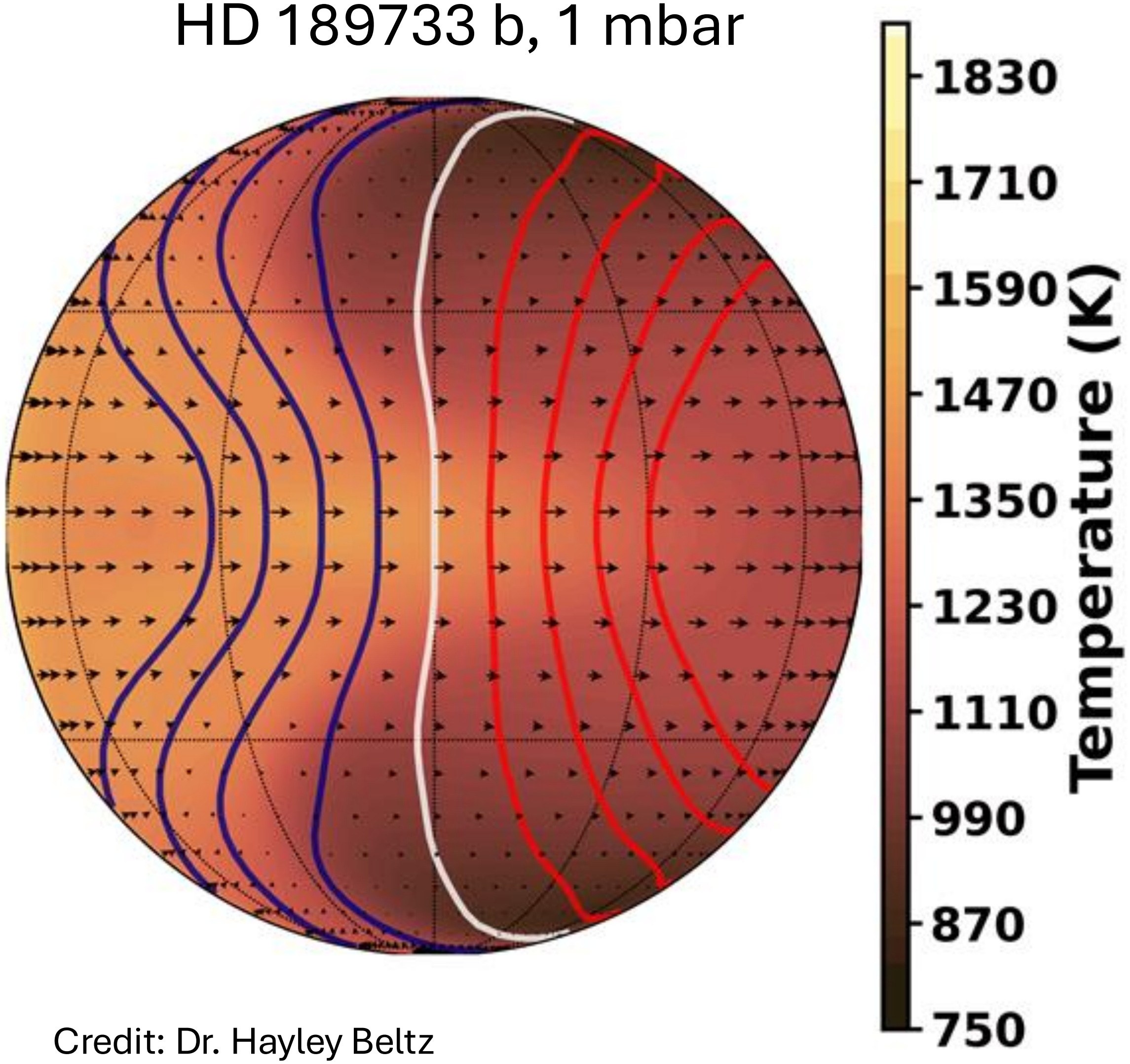
# The need for speed: current RT dominates model runtimes

- 3D models are expensive, reducing accuracy and limiting what we can explore




Showman+ 2009

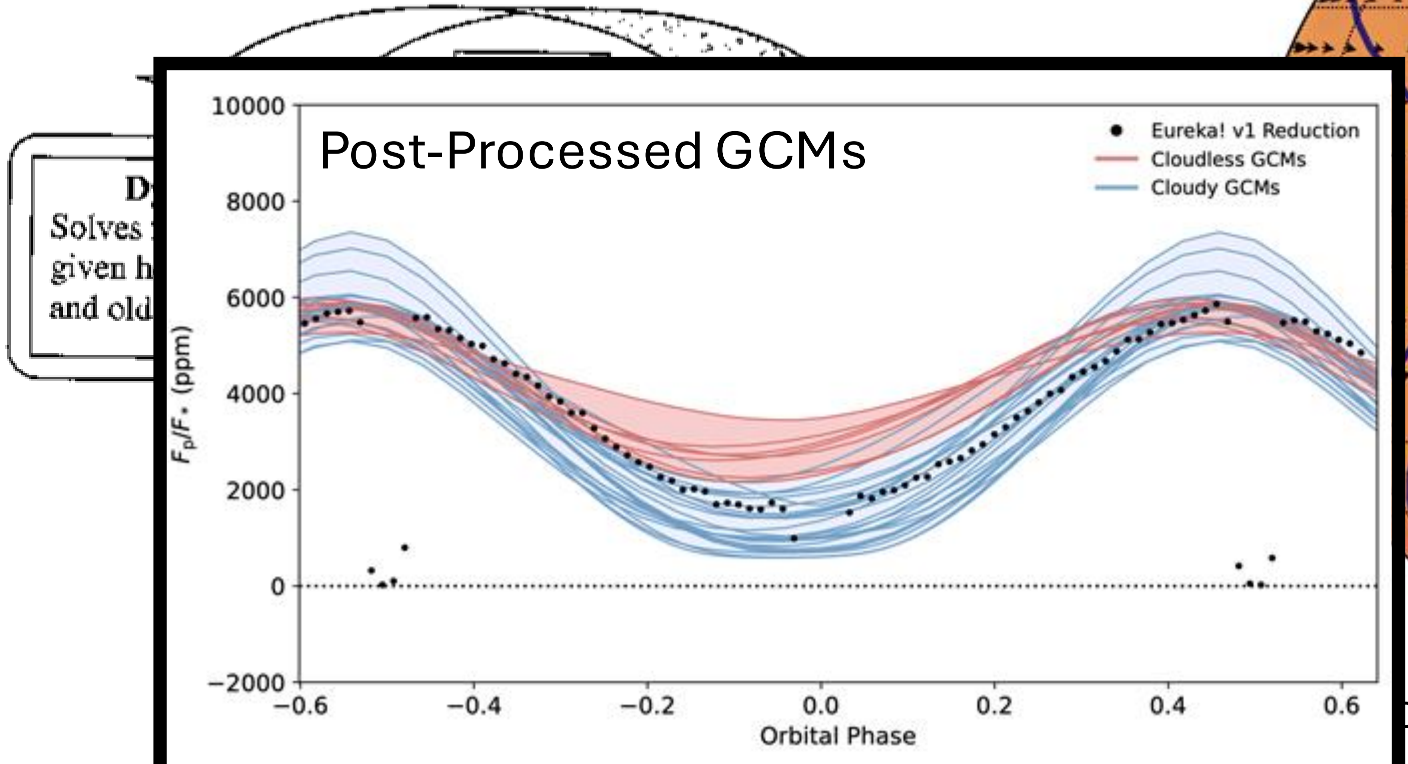
HD 189733 b, 1 mbar




Credit: Dr. Hayley Beltz

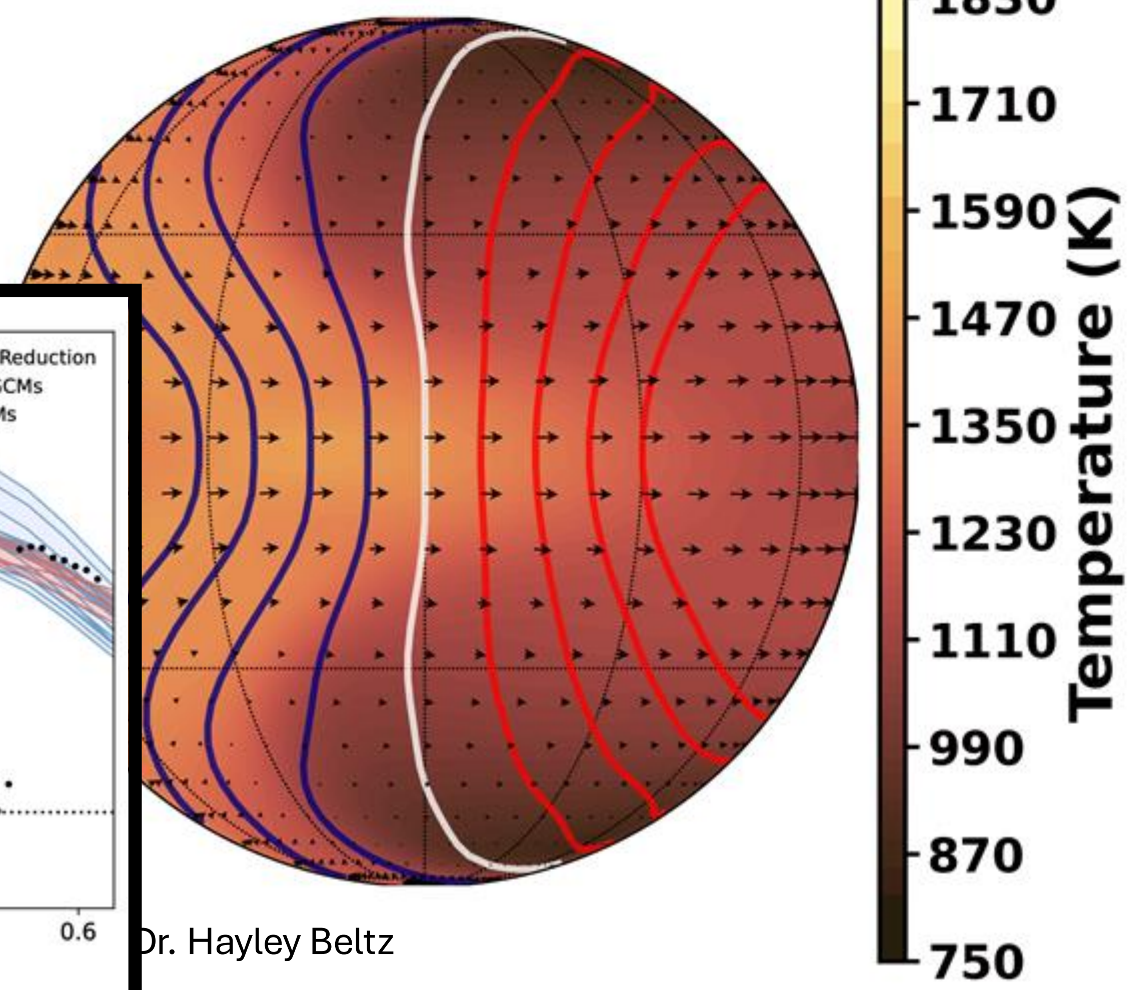
# The need for speed: current RT dominates model runtimes

- 3D models are expensive, reducing accuracy and limiting what we can explore




Bell+ 24 (inc. Malsky)

HD 189733 b, 1 mbar




Dr. Hayley Beltz

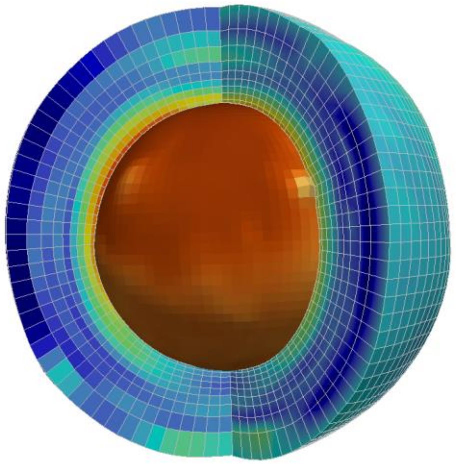
# The need for speed: current RT dominates model runtimes

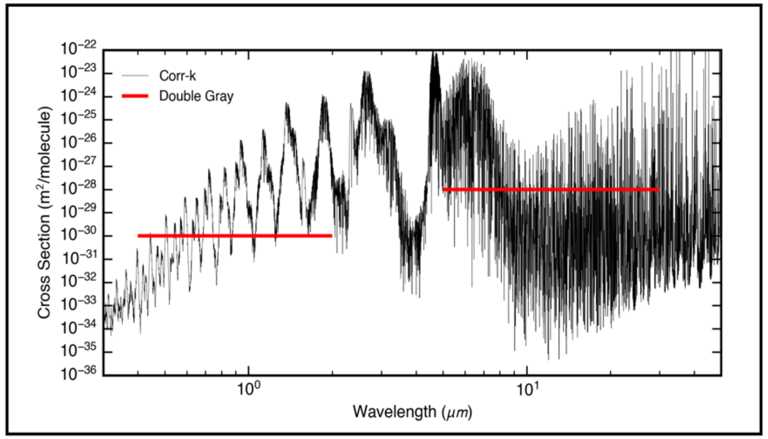
- Accurate solutions are known (and easy to generate), but extremely computationally demanding

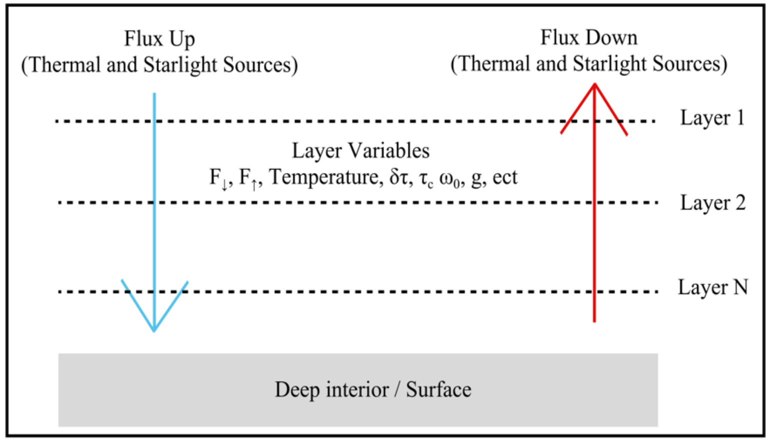
For every latitude + longitude + 1D profile

Determine atmospheric opacities

Calculate radiative transfer






Feed back in layer fluxes and repeat

# Machine learning methods are being applied in astronomy

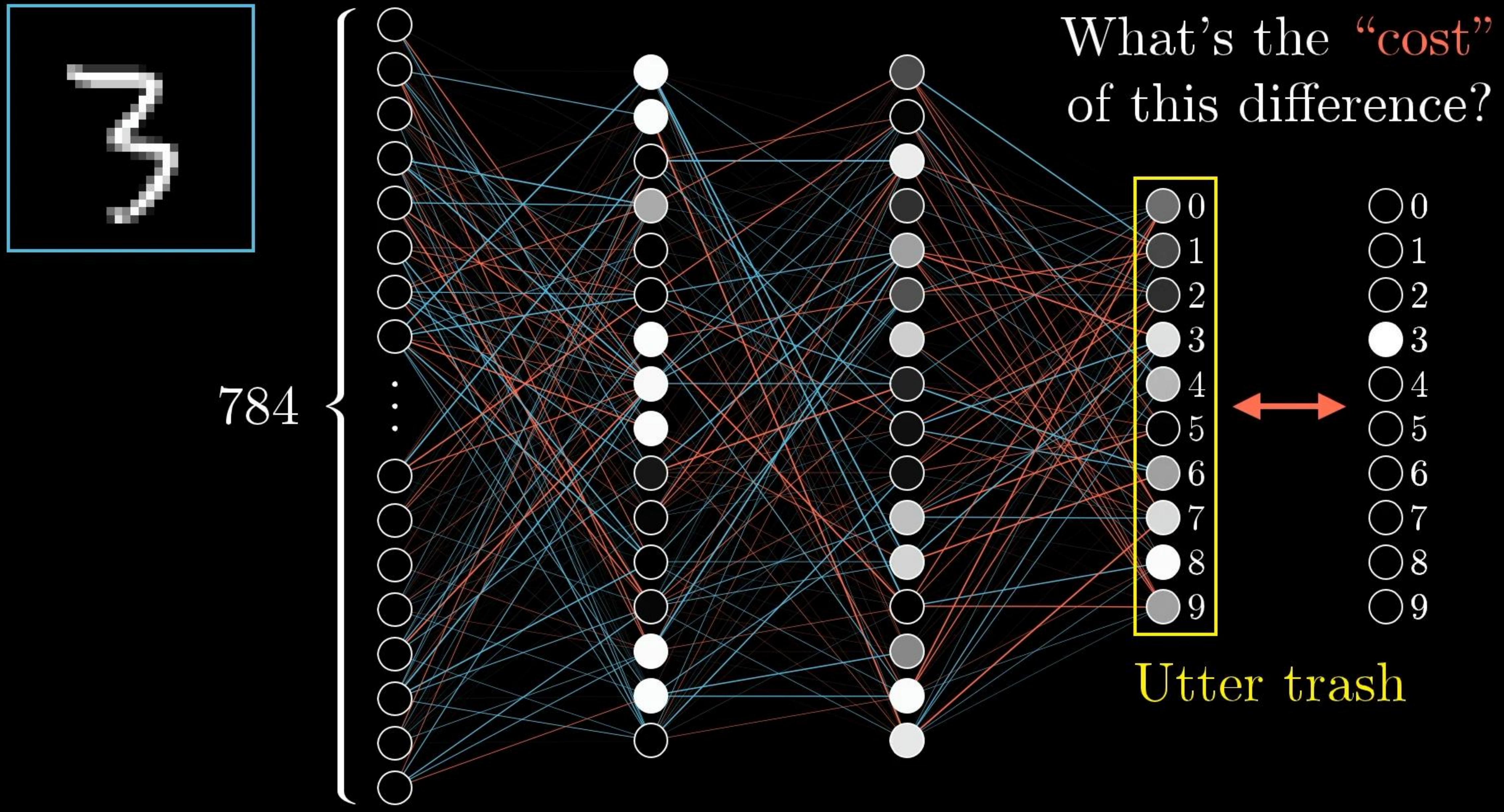



Image: Three Blue One Brown

# Language translation = radiative transfer (kind of)

- Goal: Determine from sequence A, the corresponding sequence B
- Need to capture the interdependencies of the sequence(s)

**Input**

A fluffy blue creature roamed the forest → Una criatura azul y esponjosa vagaba por el bosque.

**Output**

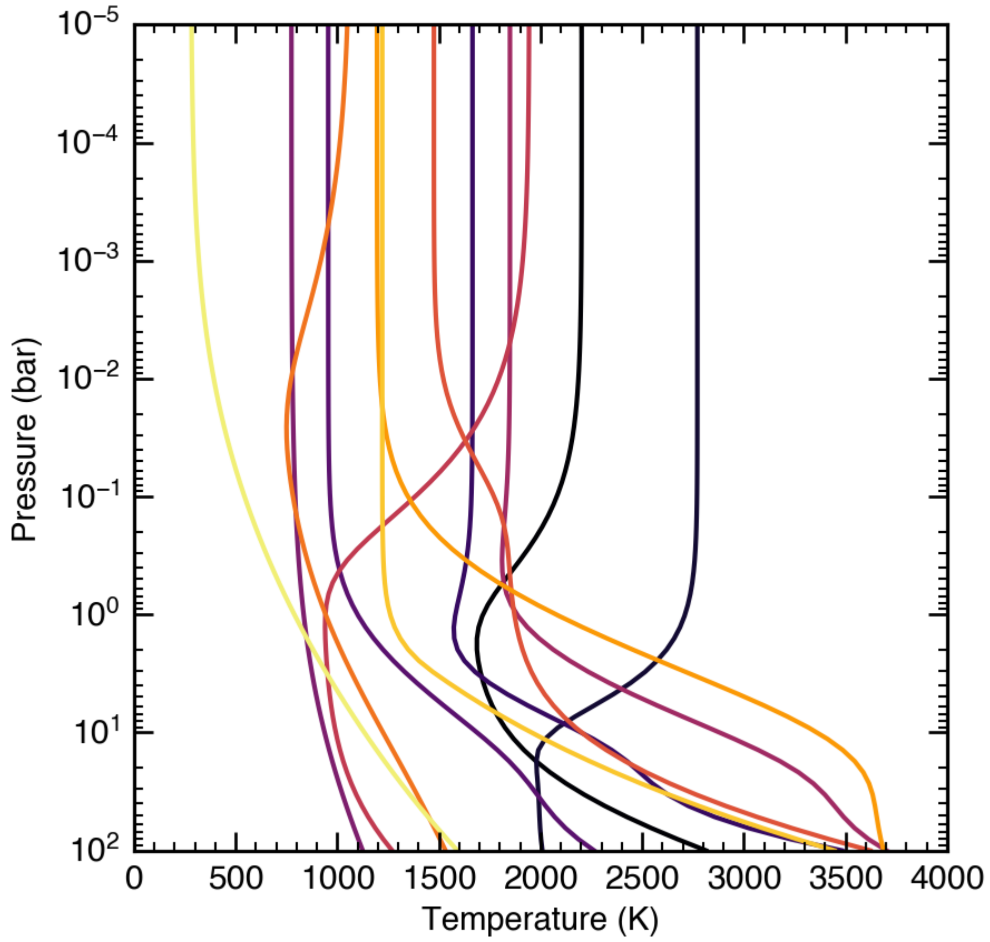
# Language translation = radiative transfer (kind of)

- Goal: Determine from sequence A, the corresponding sequence B
- Need to capture the interdependencies of the sequence(s)

| **Input** | | **Output** |
|---|---|---|
| A fluffy blue creature roamed the forest | $\rightarrow$ | Una criatura azul y esponjosa vagaba por el bosque |
| $(T_1, P_1), (T_2, P_2), (T_3, P_3)\ldots(T_N, P_{1N})$ | $\rightarrow$ | $(Flux_1), (Flux_2), (Flux_3)\ldots(Flux_N)$ |

# We need a training dataset that covers the given parameter space

- Randomly generate ~2 million parameterized 1D profiles
- Simple (no zenith angle, single star type, etc) but flexible framework

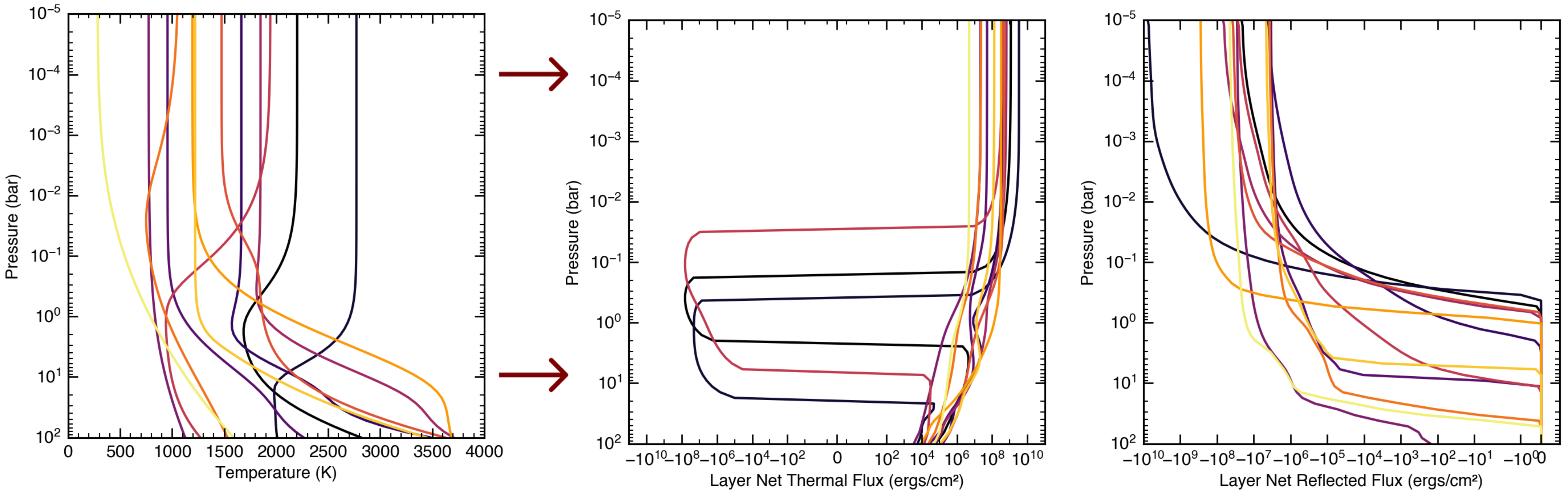



| **Uniform Distribution** | | | |
|---|---|---|---|
| **Parameter** | **min** | **max** | **Unit** |
| $\kappa_{IR}$ pressure power-law exponent | -0.5 | 1 | – |
| $\log_{10}(\kappa_{\rm IR})$ | -2.5 | 2.5 | $\rm m^2\,kg^{-1}$ |
| $\log_{10}(\gamma_1)$ | -2 | 2 | – |
| $\log_{10}(\gamma_2)$ | -2 | 2 | – |
| $\alpha$ | 0.0 | 1.0 | – |
| Temperature shift | -600 | 600 | K |
| **Normal Distribution** | | | |
| **Parameter** | $\boldsymbol{\mu}$ | $\boldsymbol{\sigma}$ | **Unit** |
| $T_{\rm int}$ | 300 | 700 | K |
| $T_{\rm irr}$ | 1800 | 500 | K |
| Orbital separation modifier | 1.0 | 0.5 | – |
| **Other / fixed** | | | |
| Pressure range (bounds) | $10^{-5}$ | $10^{2}$ | bar |
| Convective adjustment | On: 1/3 | Off: 2/3 | – |

Malsky+ (submitted)

# We need a training dataset that covers the given parameter space

- Take the 1D profiles and process them with PICASO to get net layer fluxes
- The many wavelength bin calculations would be too expensive in a GCM



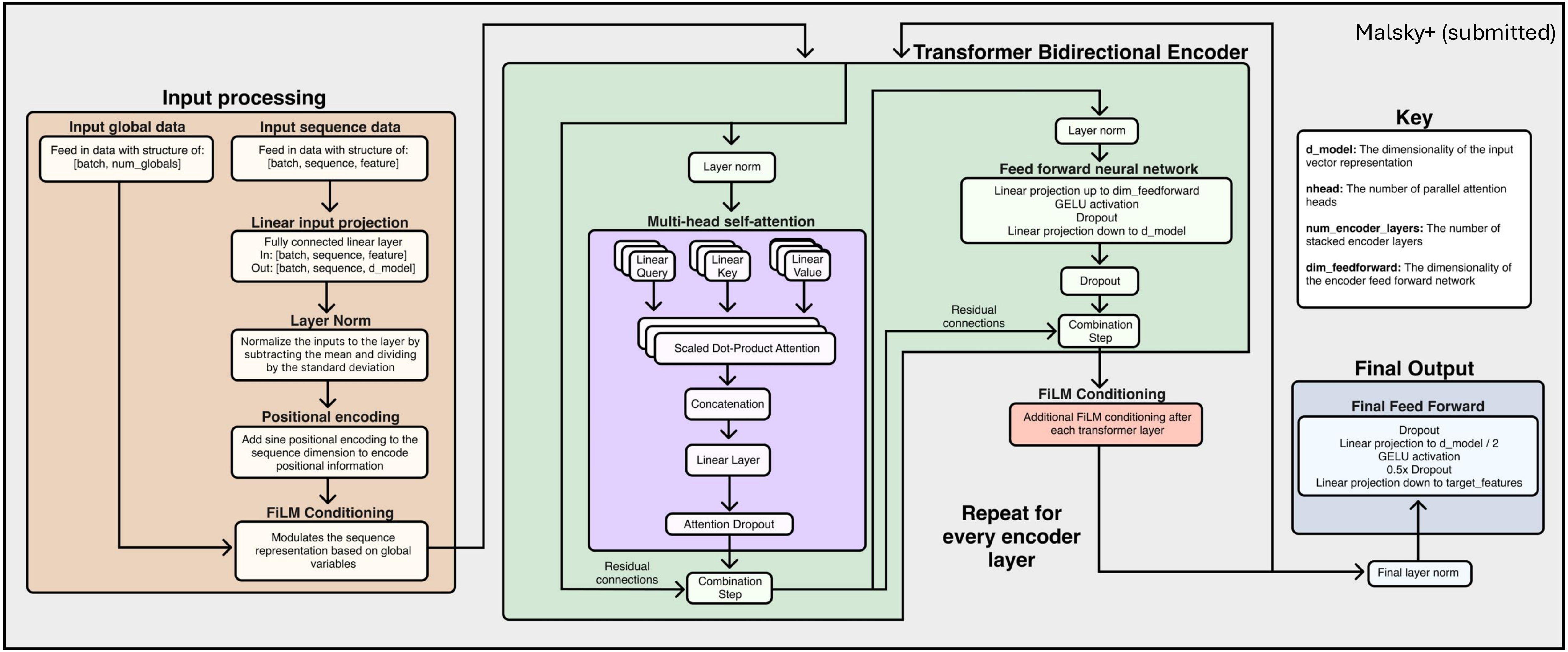

Malsky+ (submitted)

# A bidirectional encoder-only transformer proved to be the best model

No time masking

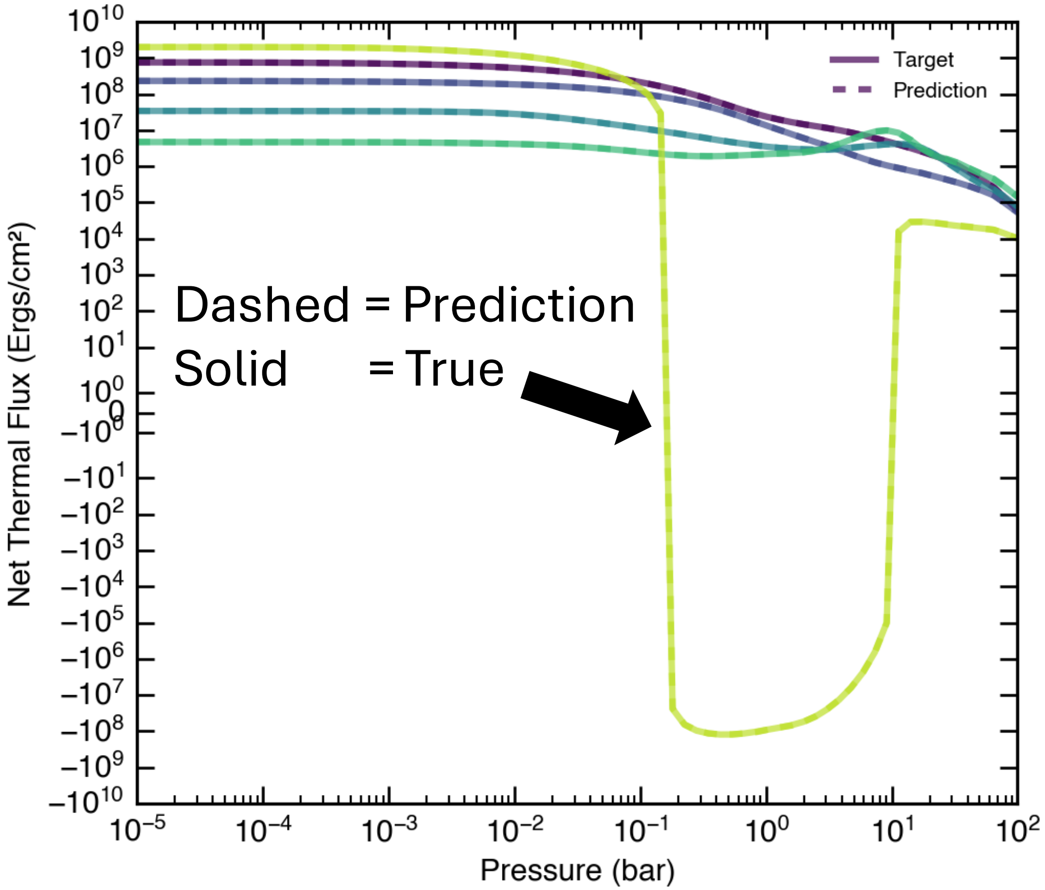
Creates representations, not generative

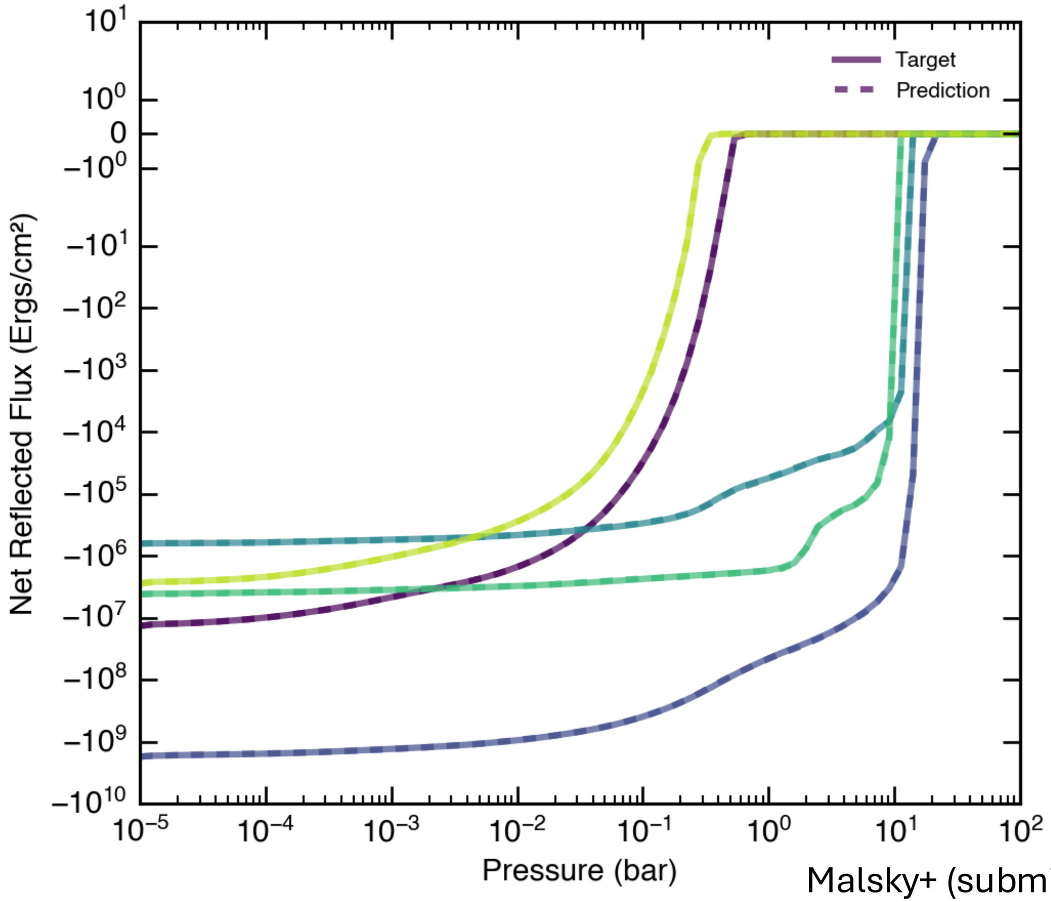
Uses attention mechanisms



# The model reproduces PICASO radiative transfer

- The model is accurate to within about 1% (compared to PICASO) for both thermal and scattered starlight fluxes

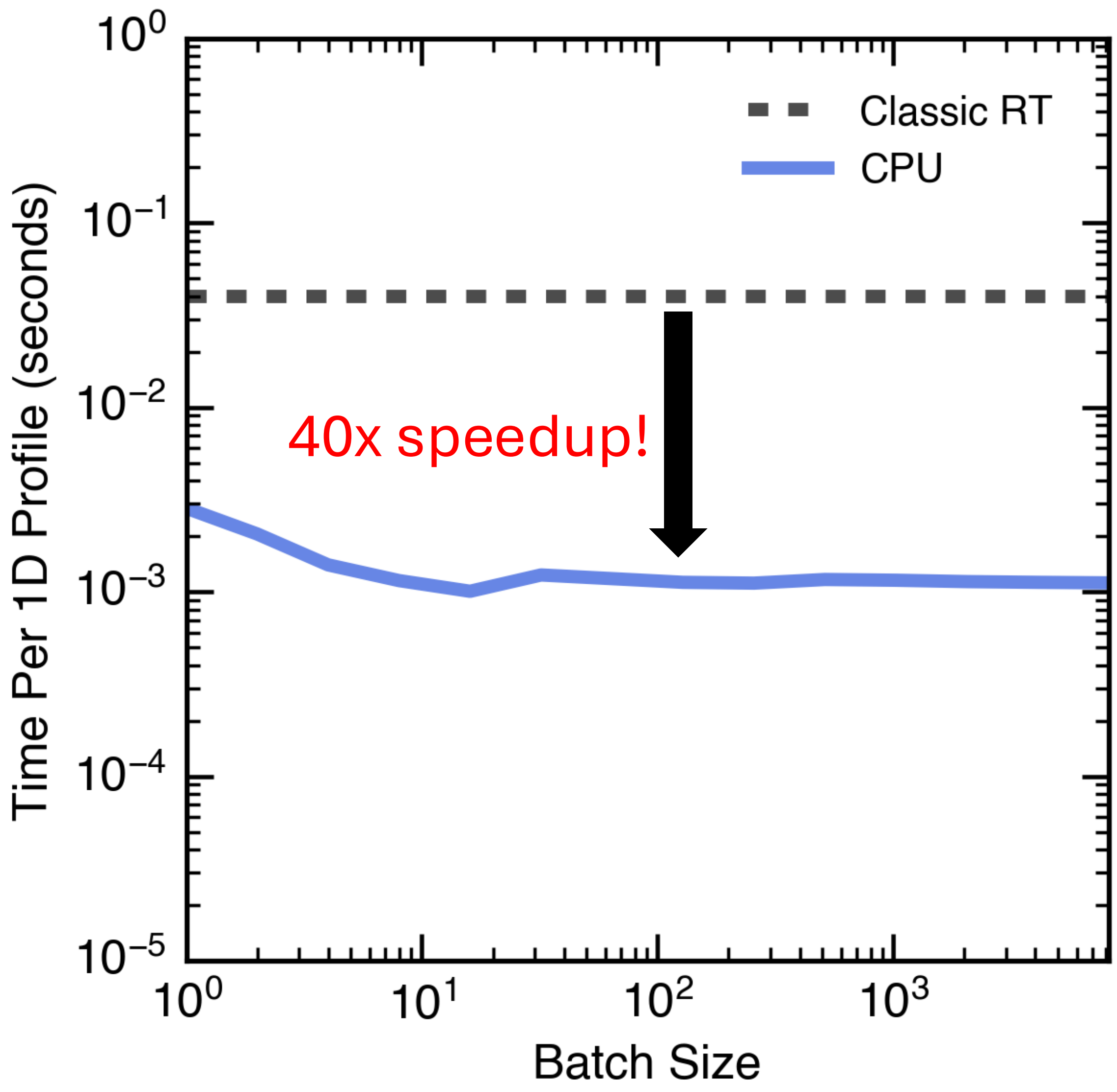



Malsky+ (submitted)

# Fast, accurate radiative transfer

- When run on a CPU, the model is quite fast

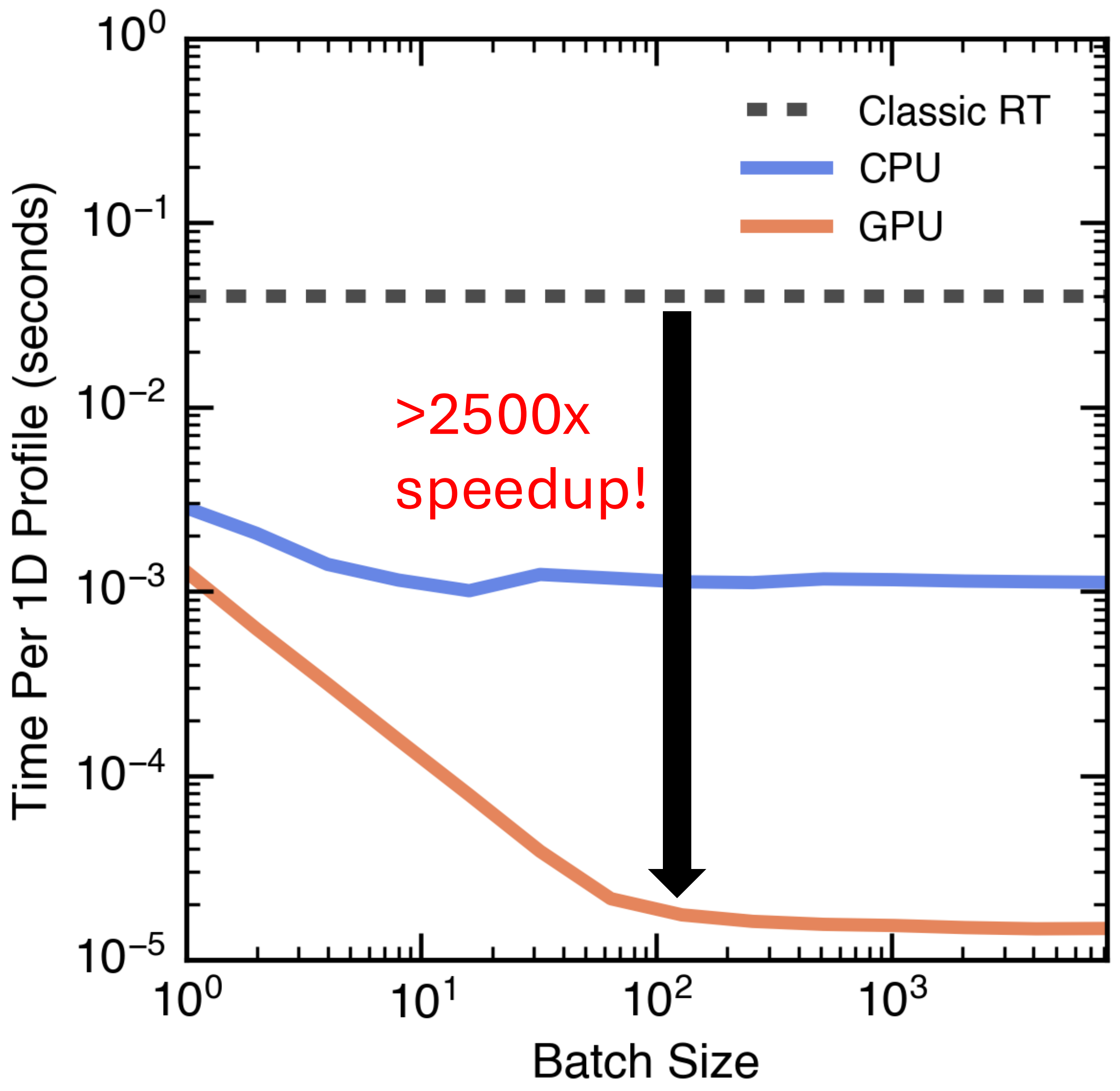



Malsky+ (submitted)

# Fast, accurate radiative transfer

- When run on a CPU, the model is quite fast

- On a GPU, the model is very fast




Malsky+ (submitted)

# Using the model

- The model is publicly available on github
  https://github.com/imalsky/Problemulator
- Several projects implementing this framework have started
  - Kevin Shao/Cheng Li are working on a DISORT/Uranus model
  - John Allen is working on a SPARC/MITgcm implementation
- Retraining and inference are designed to be as simple as possible

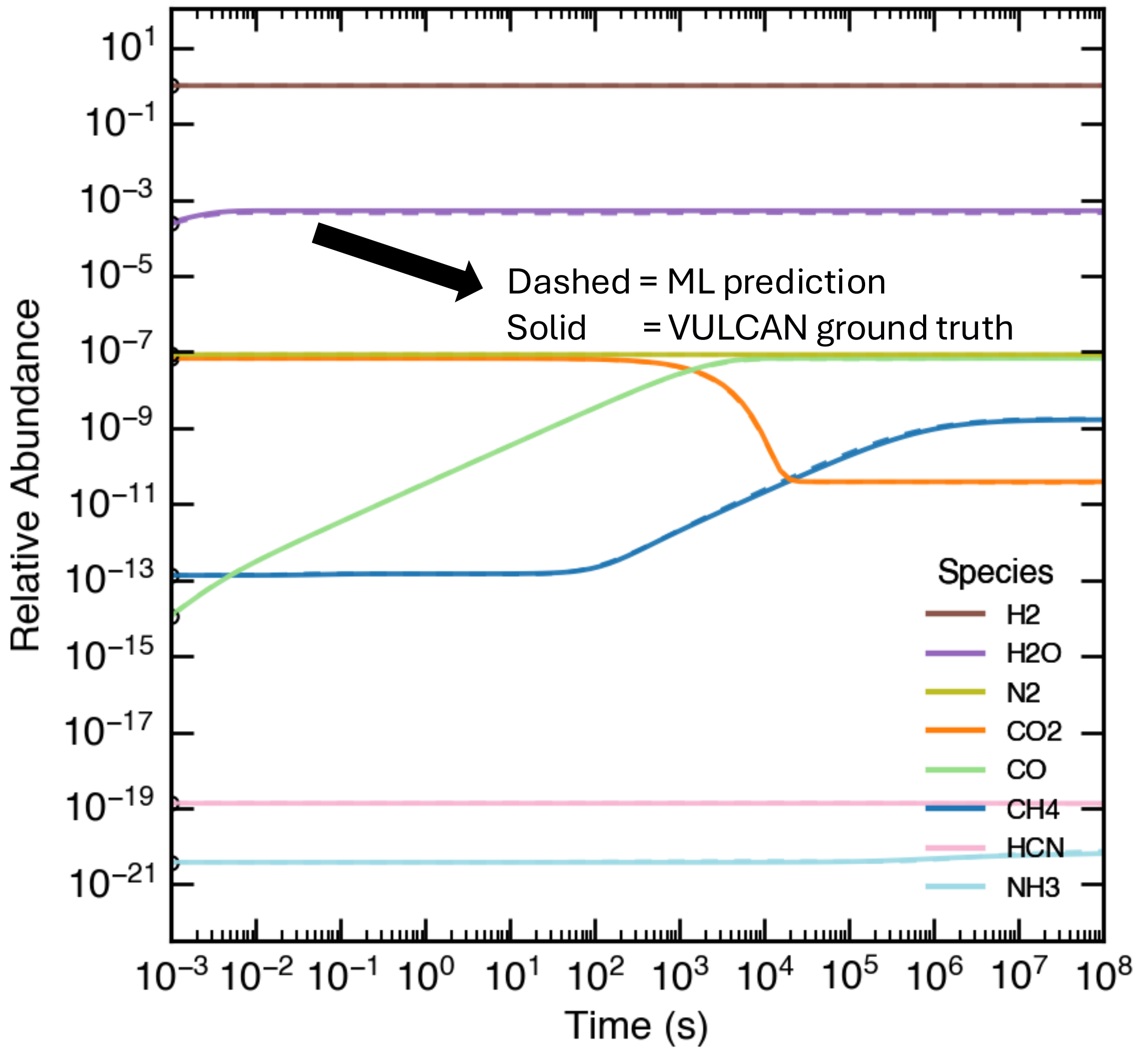
# A broader application of ML emulation of physical processes

In many ways, physical processes are the perfect domain for ML:

1. The desired solutions are known
2. Easy to generate training data
3. Easy to verify the solution
4. Original solutions are computationally expensive

- Replace chemical kinetics with an ML emulator?

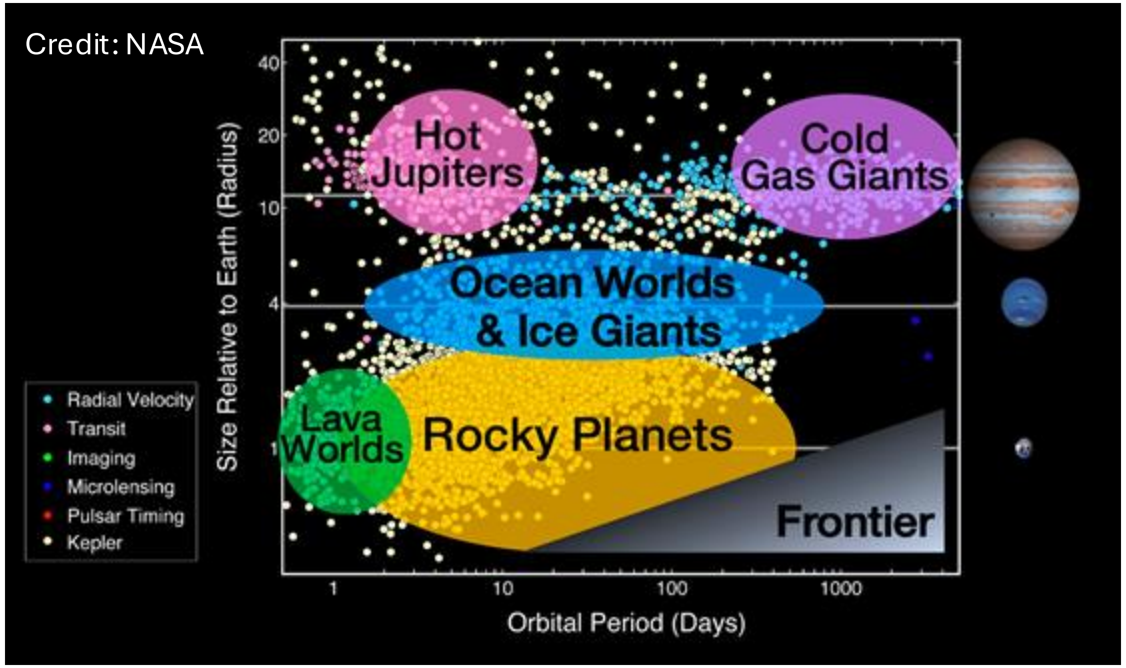
**Chemical Kinetics Emulator**

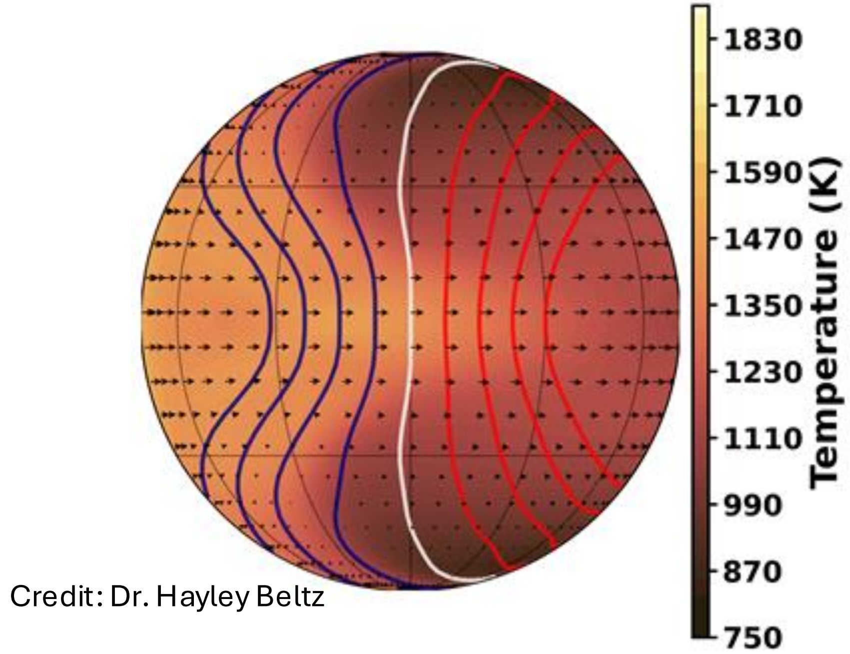


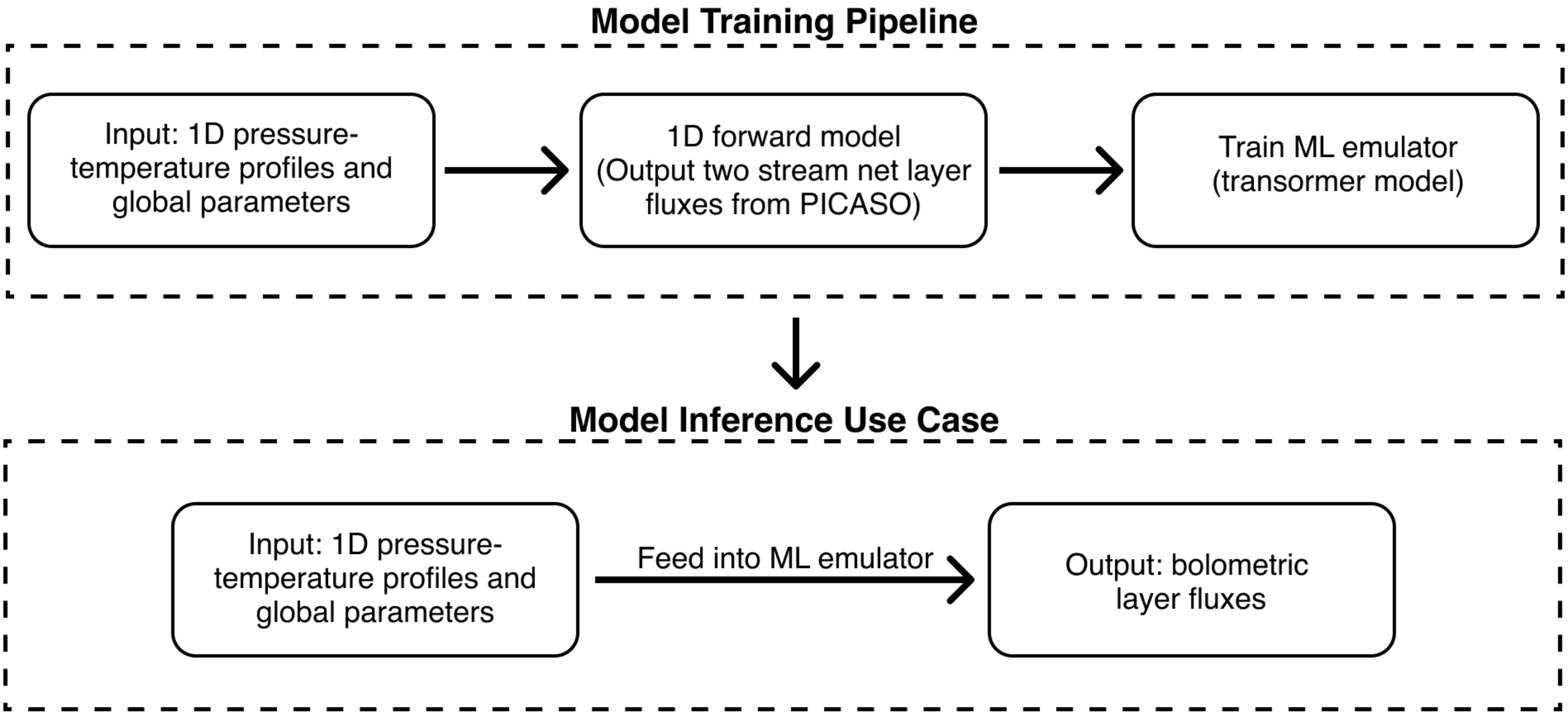

Malsky+ (in prep)

# Summary and next steps

- Fast accurate RT, easy to implement in GCMs/retrievals
- There are broader applications for ML emulators, for carefully chosen problems
- Enable more accurate simulations, broader grids, new physical processes in models




Model Training Pipeline
Input: 1D pressure-temperature profiles and global parameters
1D forward model (Output two stream net layer fluxes from PICASO)
Train ML emulator (transormer model)
Model Inference Use Case
Input: 1D pressure-temperature profiles and global parameters
Feed into ML emulator
Output: bolometric layer fluxes

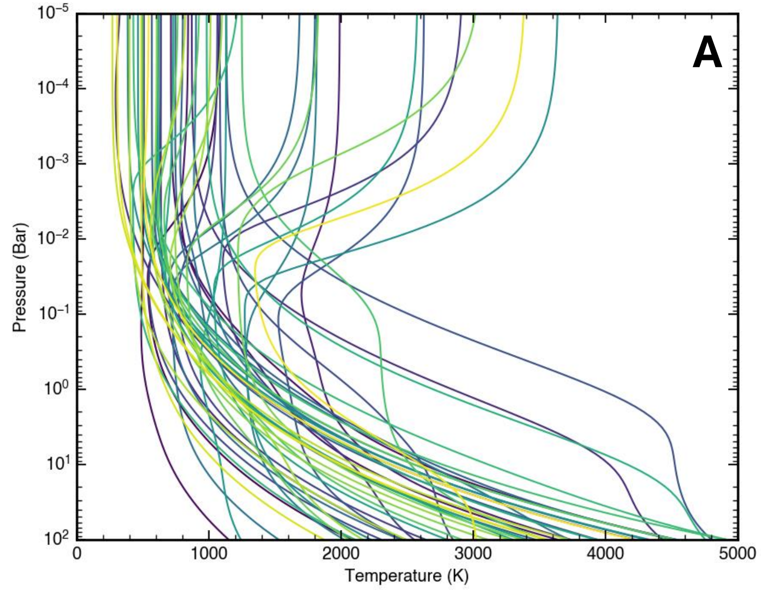
# How to make a ML radiative transfer emulator

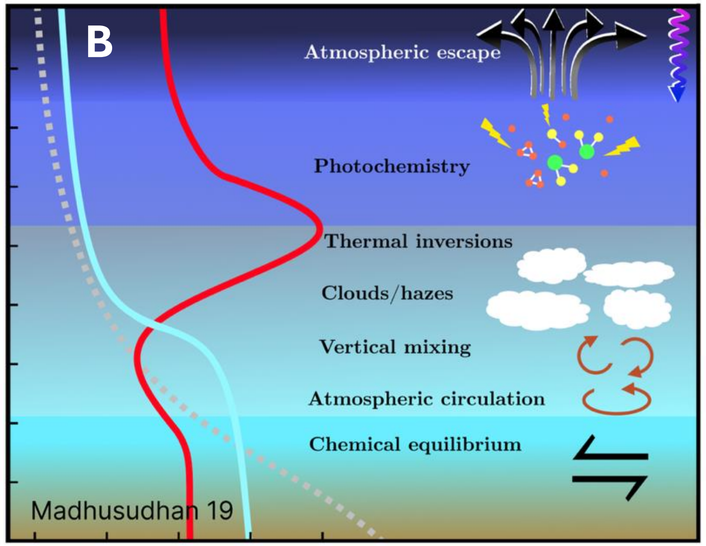
**A)** Create parameterized PT profiles

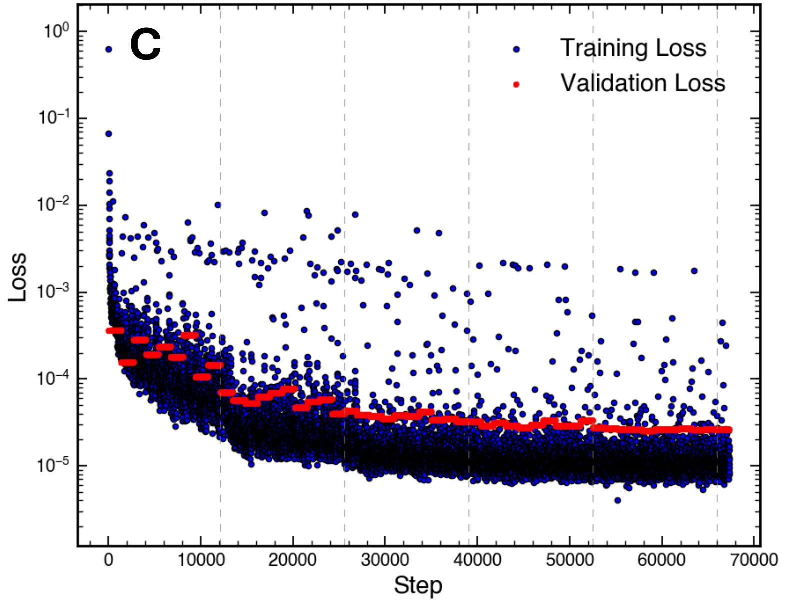
**B)** Feed profiles into 1D forward model

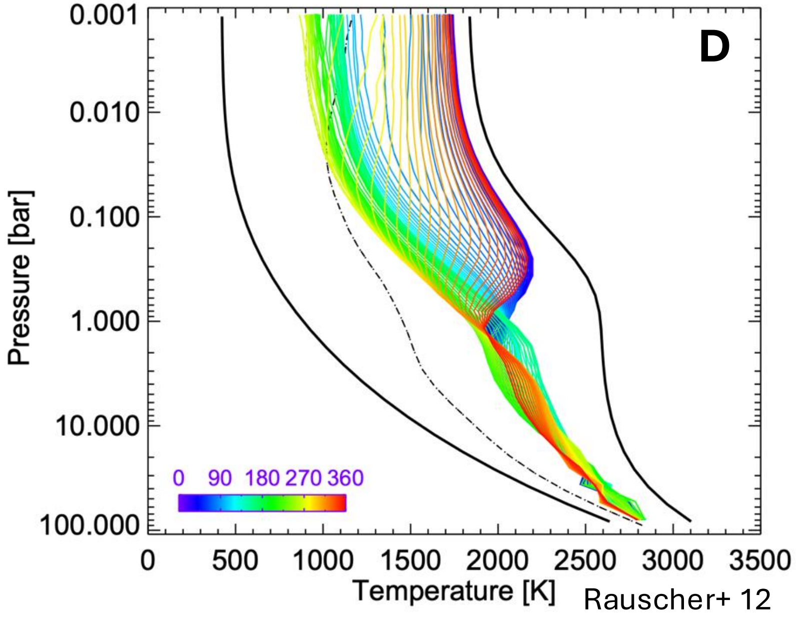
**C)** Train model to predict Flux from P, T, etc

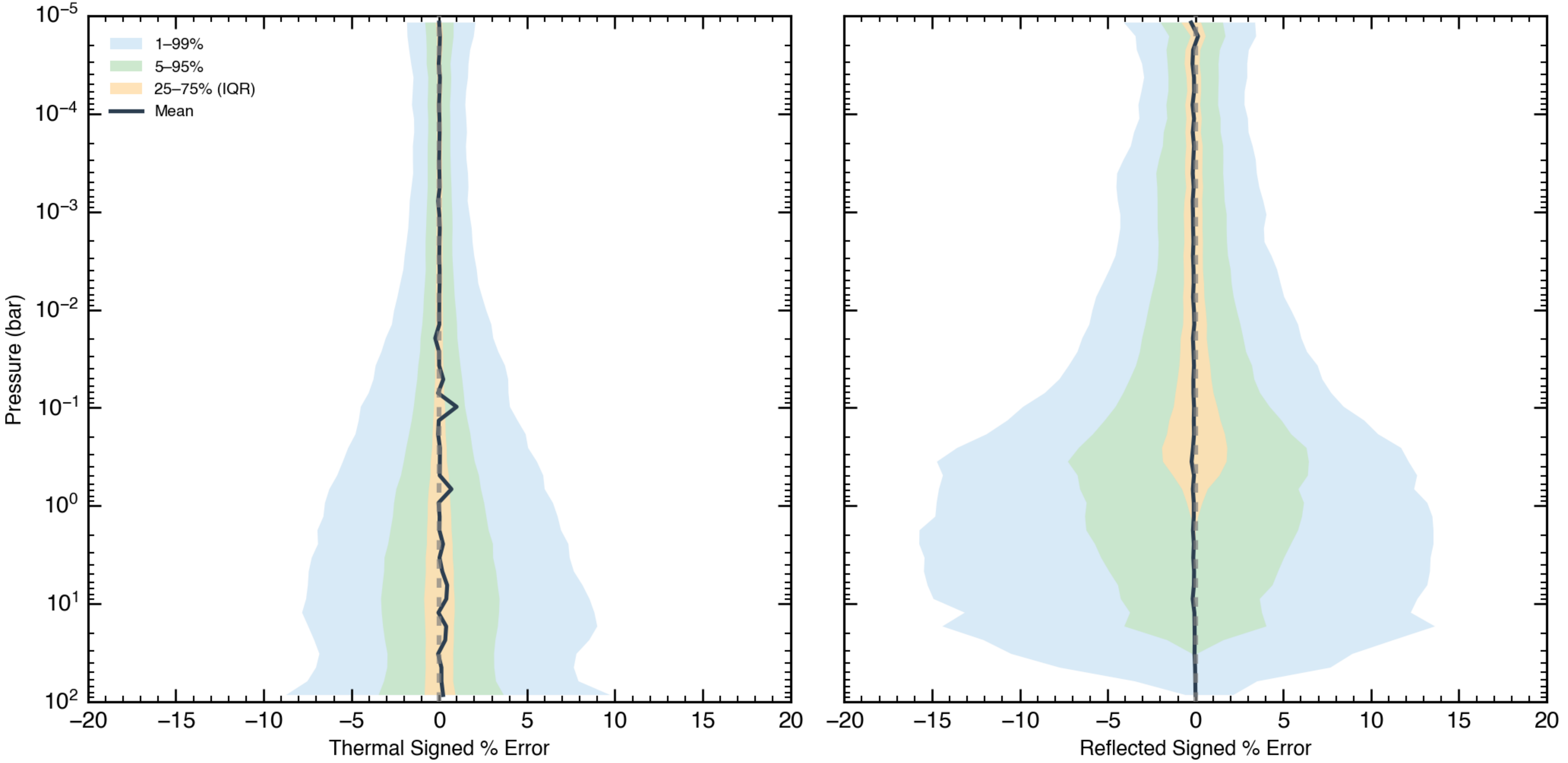
**D)** Incorporate into GCMs, Retrievals



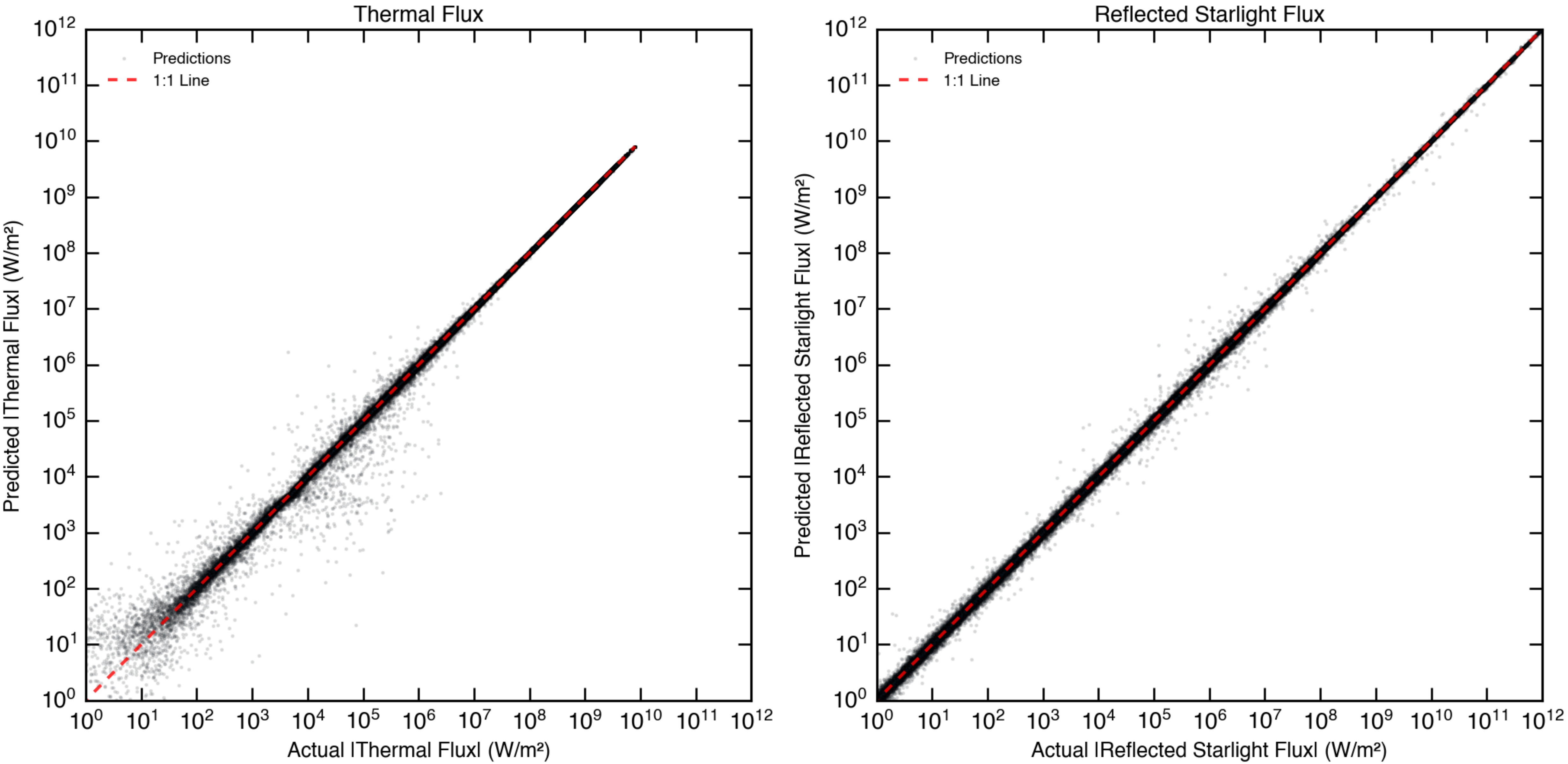



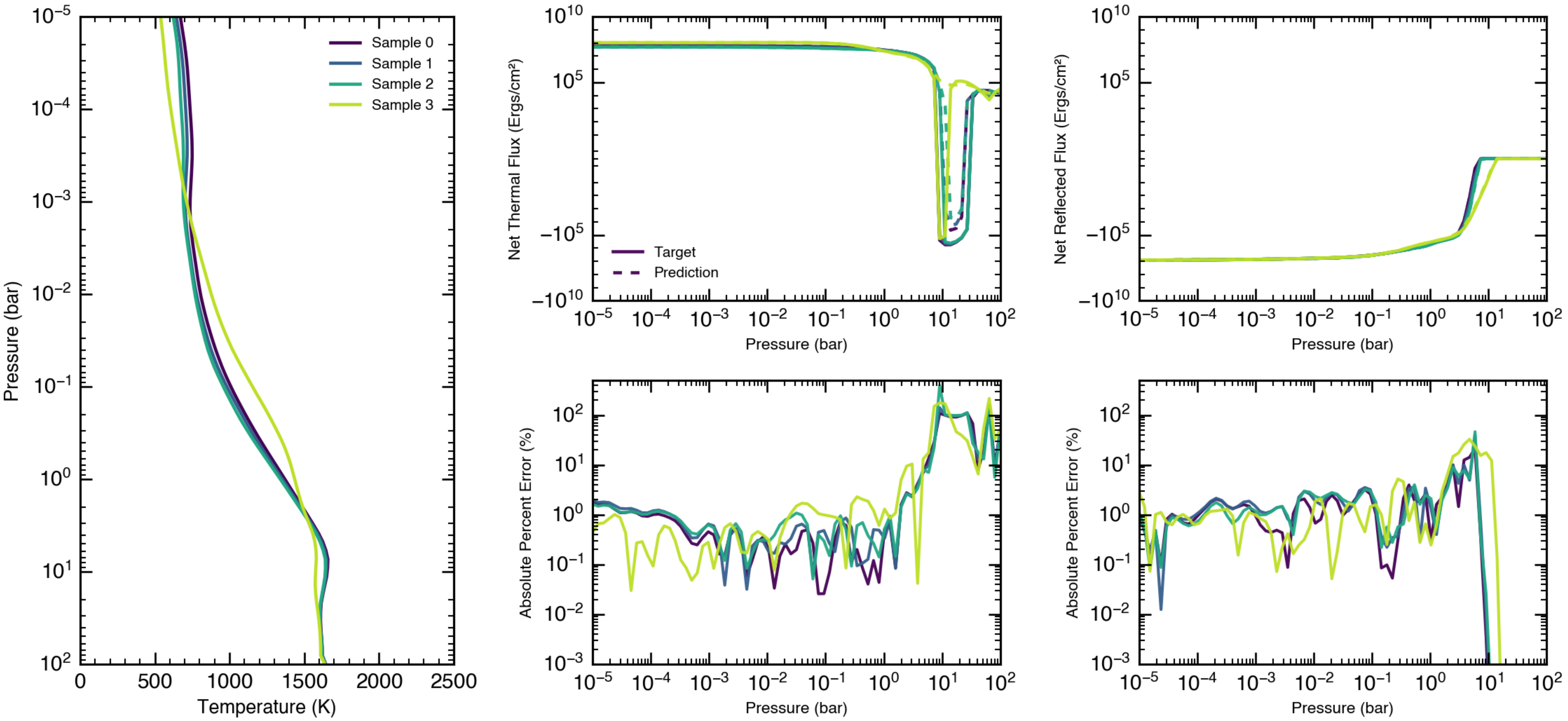





1–99%
5–95%
25–75% (IQR)
Mean
Pressure (bar)
Thermal Signed % Error
Reflected Signed % Error


Thermal Flux
Predictions
1:1 Line
Predicted |Thermal Flux| (W/m²)
Actual |Thermal Flux| (W/m²)
Reflected Starlight Flux
Predictions
1:1 Line
Predicted |Reflected Starlight Flux| (W/m²)
Actual |Reflected Starlight Flux| (W/m²)
$10^0$ $10^1$ $10^2$ $10^3$ $10^4$ $10^5$ $10^6$ $10^7$ $10^8$ $10^9$ $10^{10}$ $10^{11}$ $10^{12}$



Sample 0
Sample 1
Sample 2
Sample 3
Pressure (bar)
Temperature (K)
Net Thermal Flux (Ergs/cm²)
Target
Prediction
Pressure (bar)
Net Reflected Flux (Ergs/cm²)
Pressure (bar)
Absolute Percent Error (%)
Pressure (bar)
Absolute Percent Error (%)
Pressure (bar)