2025 年 1 月 3 日

地球流体的数值模拟与 AI 预测
基于机器学习的行星大气光谱反演

姓名：王艺霖

学号：2200011456

1 问题介绍

近年来，随着 JWST 等空间望远镜的发射和运行，我们对系外行星的观测也日益丰富。对于一颗行星，观测可以告诉我们的是它的发射或者透射光谱，对于除了裸岩行星之外的大部分行星来说，这些光谱的特征主要是根据其大气廓线和成分决定的。那么，很自然的问题就是我们应该如何从这些观测所得到的光谱数据来推断出行星大气的性质，包括温度、压强廓线以及大气成分的含量和垂直分布等。

假如我们已知的是大气的廓线和成分，想要推断它的光谱，这件事情是很直接的，我们可以利用逐线积分或 correlated-k 等方法进行计算，虽然这个过程可能会由于计算量很大而面临优化问题，但是这件事情本身是没有原则上的困难的。然而正如简单的两个质数相乘的逆过程质因数分解格外困难一样，我们在对光谱反演的时候也会遇到非常大的困难。这个困难主要来自于大气性质与光谱的高度非线性关系，气体分子本身就具有非常复杂的谱线，再加上由于多普勒和碰撞造成的展宽，而这两个展宽效应本身又依赖于局地的温度、压强以及气体成分，这就导致光谱信号不可能被温度廓线、大气成分廓线等进行线性的表示。

目前，进行大气光谱反演问题的主流方法是马尔科夫链蒙特卡罗（Markov-Chain Monte Carlo, MCMC），这种方法的基本过程是在参数空间内进行随机游走，经过长时间的游走之后对结果取统计平均。由于每进行一步随机游走都需要根据当前的参数进行一次光谱计算来和观测数据比对，而进行一次完整的 MCMC 可能需要几十万步的随机游走，所以这种方法的显著缺点就是慢，即使考虑到我们可以通过同时维护多条马尔科夫链来进行并行计算，这种方法的消耗的时间和计算资源也是很可观的。

机器学习恰好是一种适合对非线性的数据关系进行预测的方法，因此我们希望利用机器学习的方法来对行星大气的光谱进行快速准确的反演。

2 方法

在我的实现过程中主要包括生成数据集和训练两大部分，接下来我将分别对这两个部分进行介绍。

2.1 生成数据集

机器学习的训练需要大量的数据，而我在互联网上很难找到质量较高、格式比较统一、参数覆盖范围比较广的观测数据集，因此我决定自己生成数据集。

在生成数据集的过程中，我使用的是 PYTHON Radiative-transfer in a Bayesian framework (PYRAT BAY) [1] 这个辐射传输模式。这个模式允许用户对行星的半径、顶部和底部压强、温度和成分廓线以及其主星的光谱等各种参数进行设置，从而计算出行星的透射或发射光谱。在我生成数据包括后续进行训练的时候，我最主要关心的是行星大气的温度和成分廓线，因此我选取了行星的发射光谱，并固定了行星的半径和压强范围，只对温度和成分廓线进行改变和训练。

PYRAT BAY 模式允许用户自己给定温度廓线，也提供了三种预设的温度廓线类型供用户选择，分别是等温廓线、Guillot 廓线和 Madhu 廓线。显然，在模拟中不能使用等温廓线，因为那样会导致得到的光谱就是理想黑体谱。为了方便起见，我选择直接使用模型中预设的 Madhu 廓线，这个廓线是 Madhusudhan 提出的一种用于拟合系外行星温度廓线的模型 [2]。如图1所示，这个模型将大气分为三层，最下面的一层是等温的。

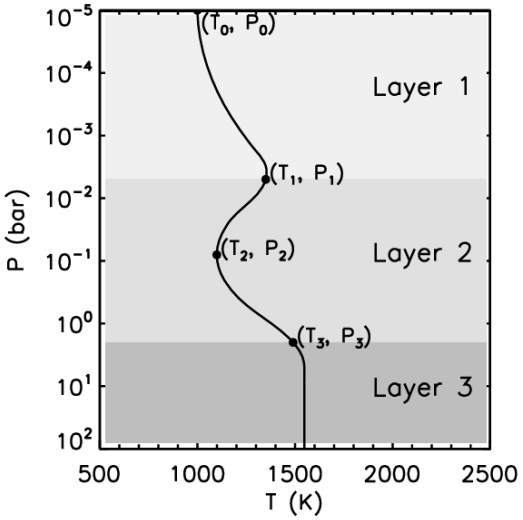

图 1: Madhu 温度廓线示意图

Madhu 模型的数学表达式如式 (1) 所示，一共有 6 个自由参数，分别是大气层顶的温度 T_0、三个压强参数 p_1, p_2, p_3，以及第一层和第二层温度廓线的形状参数 a_1, a_2。p_1 和 p_3 分别是三层大气的两个分界面对应的压强，所以必须满足 $p_1 < p_3$，而 p_2 则没有限制，如果 $p_2 > p_1$ 的话，就会在大气的第二层形成一个逆温层，反之则不会。

$$T(p) = \begin{cases} T_0 + \left[\frac{1}{a_1}\ln(p/p_0)\right]^2 & \text{if } p < p_1 \\ T_2 + \left[\frac{1}{a_2}\ln(p/p_2)\right]^2 & \text{if } p_1 \le p < p_3 \\ T_3 & \text{if } p \ge p_3 \end{cases} \tag{1}$$

对于大气成分廓线，PYRAT BAY 提供了两种选择，第一种是指定各种气体成分的含量，然后模式会根据先前给定的温度廓线计算出达到热平衡之后的成分廓线，第二种就是简单的均匀分布。为了简单起见，我选择了均匀分布，并且选定了总共 18 种气体成分，包括 $H_2O, CO_2, N_2, O_2, CH_4, O3, He, H2, NH3, PH3, CO, SO_2, HCN, H_2S, NO, N_2O, HCl, C_2H_2$。其中有 8 种气体被我定义为主要气体，分别是 $H_2O, CO_2, N_2, O_2, CH_4, He, H_2, NH_3$，这些气体在我的数据中可能会含量很多，也可能会含量很少；另外的 10 种气体是微量气体，在所有的数据点中都含量很少。

在生成数据和后续训练的过程中，我总共选取了 24 个参数，分别是 Madhu 温度廓线的 6 个参数，以及 18 种气体的含量。在生成数据的过程中，我总共运行了 10800 个 case，每个 case 都对这 24 个参数进行了随机的指定。

在输出数据时，我选择的是行星发射光谱的 0.5μm $\sim 10\mu$m 波段，分辨率是 0.5nm，所以每一条输出光谱都包含 19001 个点。

2.2 训练

训练用的数据集包括 10800 个样本，每个样本包括一个长为 19001 的光谱数据，以及 24 个待学习的参数。

我使用了 6 种模型进行学习，分别是线性回归、随机森林、MLP、1D CNN、2D CNN、Transformer，其中 MLP 是多层感知机，1D CNN 和 2D CNN 分别是一维和二维的卷积神经

网络。为了使得模型之间得效果可以相互比较，我对数据的输入和输出标准进行了统一，并且统一采用 train:validation:test=8:1:1 的比例对数据进行随机划分。

在搭建并优化这些模型的过程中，ChatGPT 给我提供了很大的帮助，接下来我将对这几个模型进行一定的简述。

线性回归、随机森林和 CNN 都是直接对原始的长度为 19001 的光谱数组进行处理，而 MLP 和 Transformer 则是先利用主成分分析（PCA）将数据降维，对降维后的数据进行学习。

MLP 包括了输入层、两个隐藏层和输出层，总共 1536 个神经元，激活函数采用的是 SiLU，并采用了 AdamW 优化器和 Cosine Annealing 学习率调度器。训练过程中采用分批训练，每个 batch 包括 128 个样本，训练过程持续 330 个 epoch。

1D CNN 就是直接使用一维的卷积核对信号进行处理。在进行 1D CNN 的训练的时候，我想到能不能通过将一维数组折叠成二维的方式进行 2D CNN 学习，我进行了一些简单的检索之后发现这个想法是可行的，于是我就将光谱数据折叠成 190×100 的形状，并用 2D CNN 进行了训练。根据我的想法，这样的做法允许我们使用二维卷积核进行处理，可以捕捉到一些一维序列体现不出来的信息——当然这种信息未必有物理意义；但是这种做法的缺点也显而易见，原本连接处的序列信息可能会丢失，所以我想尝试一下，看看这样的处理方法总体上对结果有没有提升。

Transformer 的架构包括 Embedding 层、2 层的 Transformer 编码器和输出层，在训练过程中同样使用了 AdamW 优化器和 Cosine Annealing 学习率调度器。同时，在训练过程中还采用了数据增强，通过加入噪声、随机缩放等方法扩展训练集。训练过程总共持续 300 个 epoch。

3 结果

为了一开始能对结果的准确度有一个整体的把握，我们先不去看 24 个参数各自的 RMSE，而是将数据分成两组，Madhu 参数和气体成分含量。六种模型对于这两组参数预测的均方误差（RMSE）如图2所示。

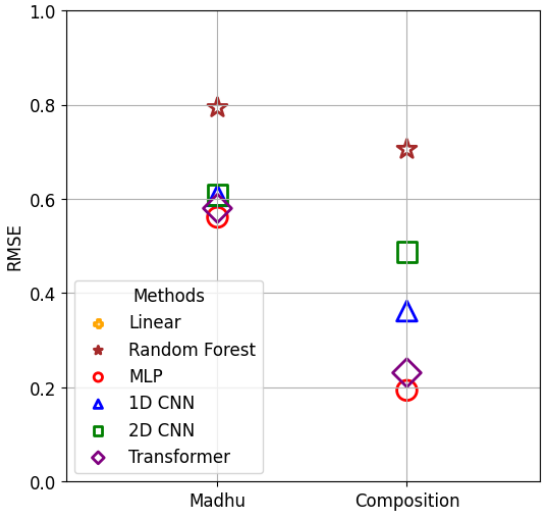

图 2: 六种模型的分组训练结果

我们在图例中看到了线性回归，但是在图中却没有对应的图标显示，这是因为线性回归给出的预测结果偏差过大，对 Madhu 参数和气体成分含量的 RMSE 分别达到了 76.13 和

43.74，对于经过归一化后的数据来说，这个误差在数量级上就无法接受，这也同时反映出光谱和大气廓线的关系的确是高度非线性的，不能够使用线性回归的模型进行处理。

在其他五种方法中，随机森林的效果相较于其他四个模型明显较差，但是其他四个模型的效果也并不令人满意，因此我们需要仔细查看这 24 个参数各自的 RMSE，来分析误差这么大的原因。

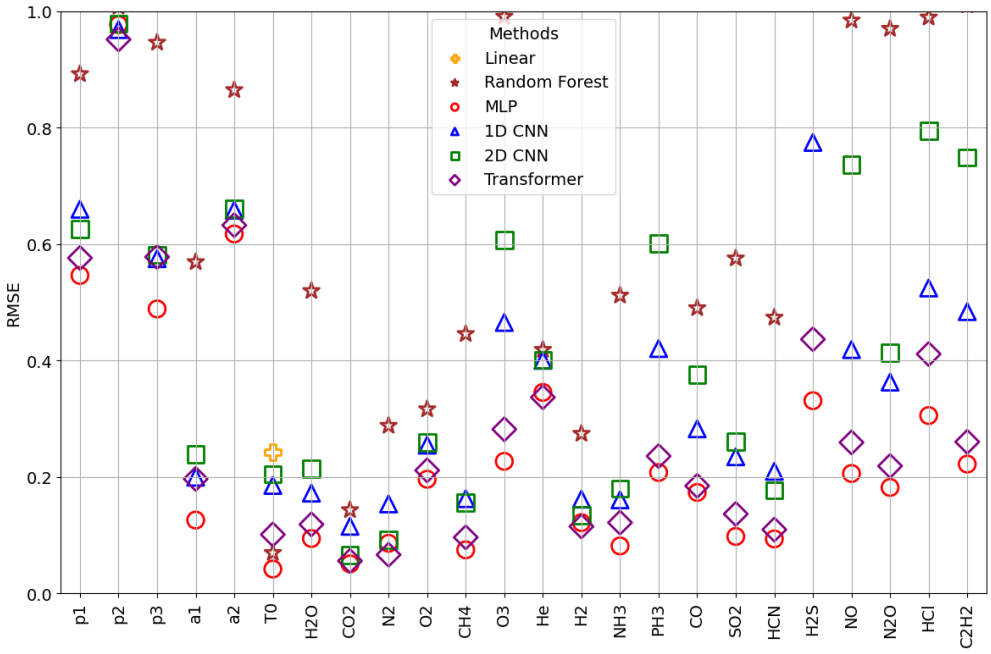

图 3: 六种模型的 24 参数训练结果

从图中我们可以看到，在 6 个温度参数中，a_1, T_0 的学习效果还可以，其余的四个参数几乎没有被学习到；在 18 个气体成分参数中，我们之前定义的 8 种主要气体的学习效果相对较好，而 10 种微量气体的学习效果较差。基于这个观察，我们重新定义两个指标来评估模型的预测效果，T_0 以及主要气体成分含量。

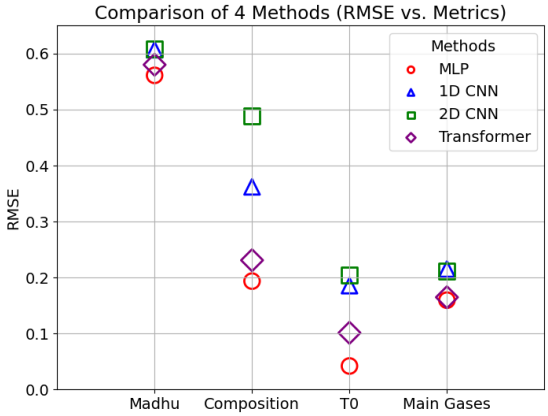

图 4: 四种模型的 T_0 和主要气体成分含量训练结果

对于我们重新定义的两个指标，结果可以参考图4，我们发现这四种模型的预测结果都基本令人满意，RMSE 都降到了 0.2 及以下，尤其是效果最好的 MLP，对 T_0 和主要气体成分含量的 RMSE 分别达到了 0.05 和 0.15。如果我们随机选取 100 个数据点，将 MLP 的预测

结果画出来，我们可以更清晰地看到其预测效果的确非常出色（如图5）。

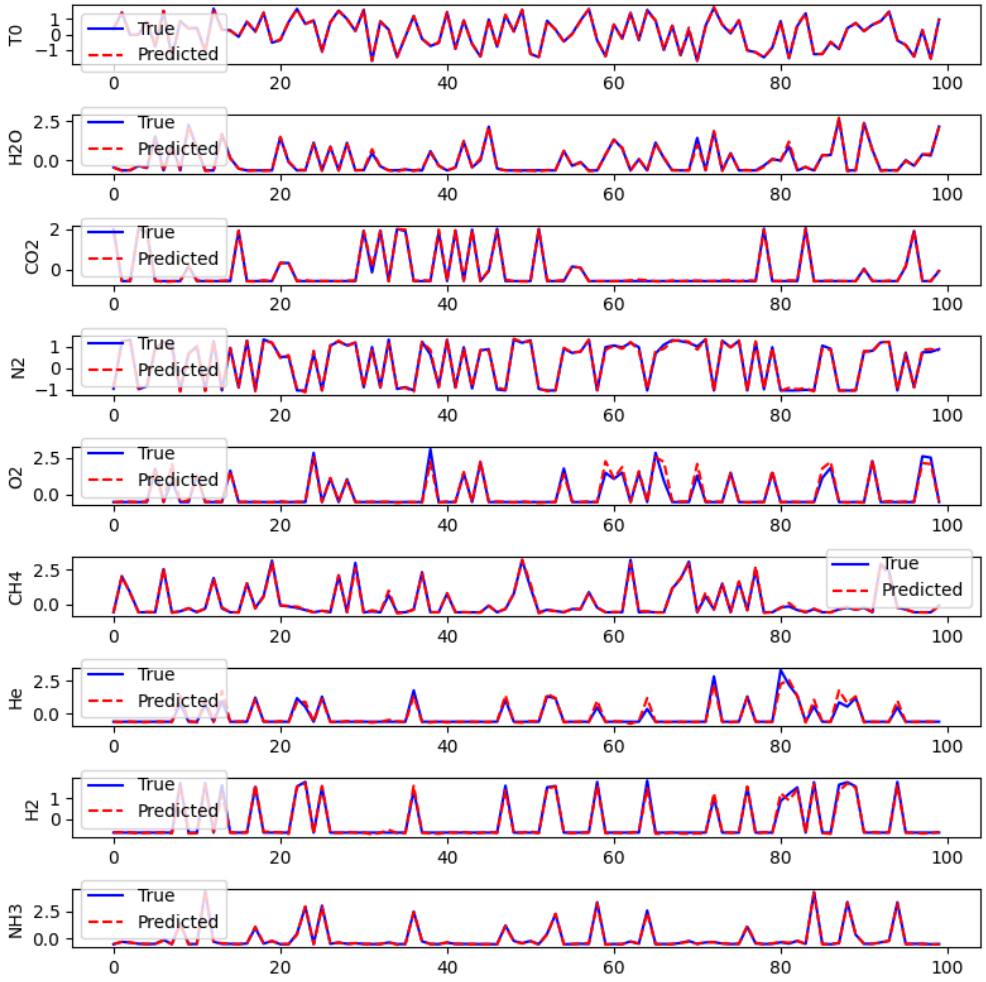

图 5: 四种模型的 T_0 和主要气体成分含量训练结果

4 讨论

我们首先要讨论的是为什么有些参数的学习效果较好，而其他的较差。

对于气体成分含量，8 种主要气体的预测效果较好，而 10 种微量气体的预测效果较差，这一点是很容易理解的。因为如果一种气体的含量较少，那么它对于光谱的影响也很小，如果这种气体成分在数据集中的所有样本中都含量很少，那么就很难在有限的样本数量中学习到。

对于 Madhu 温度廓线参数，最重要的影响因素应该是光学厚度。我们知道，在发射光谱中，不同深度的气体具有不同的权重函数，这种权重函数在光学厚度为 1 的地方最大，而深层（光学厚度远大于 1）的气体发射的光谱很难穿透。而在模拟中我使用的是一个相对深厚的大气（$10^{-6} \sim 10$bar），因此我们可以预期我们的模型将对表层的大气预测效果较好，而对深层的预测效果较差。我们回顾一下 Madhu 廓线的定义式1就很容易发现，描述最上面一层大气的参数，正是 a_1 和 T_0，其他的四个参数相对来说描述的都是更深层的大气，所以学习效果不好也是可以理解的。

对于大气成分含量的预测，我认为我们的模型的能力是可以大概判断出行星大气的主

要成分及含量，也就是以地球为例，我们可以判断出地球大气主要是由约 80% 的氮气和约 20% 的氧气组成的。但是对于更加微量的气体成分，我们的模型就几乎无能为力了，因为对于微量的成分，一个非常重要的任务是判断其在大气中是否存在，比如我们为了寻找 biosignature 可能会去寻找 PH_3，但是我们的模型只能在假定这种成分存在的基础上给出一个含量，不能进行大气中是否有这种成分的判断。

作为对机器学习应用于光谱分析的进一步展望，我认为未来有如下的几个改进方向：

1. 使用更浅的大气进行训练和预测。首先是因为我们期待寻找的宜居行星是类地的，在之前训练中使用的 10bar 大气过于深厚了，另外就是由于光学厚度的限制，我们的模型本身也只能对相对较浅的大气进行分析，很难将其应用于拥有深厚大气的行星；

2. 使用每一层的温度而不是 Madhu 参数作为温度廓线的输入变量。最开始我选用 Madhu 参数进行数据生成和预测主要是因为其简便性，我只需要控制 6 个参数就可以完成整个流程，但是后来我逐渐发现，Madhu 参数有两个缺点，一是它起初是用于描述热木星的温度廓线，描述各种行星的泛化能力有限，二是它作为一个拟合模型，对温度廓线的影响比较不直观；

3. 使用达到热平衡的气体成分分布进行训练。这是很自然的要求，之前使用均匀分布这样的 toy model 主要是出于简便，希望尽快验证模型的可行性；

4. 和主流的方法的预测效果、资源消耗进行对比。正如第一部分介绍中所说，主流方法 MCMC 的缺点就是消耗的时间和计算资源较多，机器学习模型在准确度上很可能不如 MCMC，但是一定会更加节省时间和计算资源。未来或许还可以结合二者，首先使用机器学习模型给出一个粗略的结果，将其作为初始状态输入 MCMC，应当可以大大减少 MCMC 达到平衡所需要的步数。

5 总结

在这个项目中，我们尝试使用机器学习对行星大气发射光谱进行反演。利用 PYRAT BAY 辐射传输模式生成了训练所需要的数据集，然后利用了 6 种不同的机器学习模型进行训练和预测，并将得到的结果进行了分析和对比，最后对结果进行了讨论。

6 致谢

感谢杨邱老师，他帮助我上手了我的第一个机器学习项目；感谢 ChatGPT，在模型构建和优化方面给了我很大帮助；感谢 PYRAT BAY 模式的开发者，这个模式简便的封装接口和丰富的 documentation 给我提供了很大的便利；感谢我自己，经过努力把从数据集生成到模型训练的全流程走通。

参考文献

[1] Patricio E Cubillos and Jasmina Blecic. The pyrat bay framework for exoplanet atmospheric modelling: a population study of hubble/wfc3 transmission spectra. Monthly Notices of the Royal Astronomical Society, 505(2):2675–2702, 05 2021.

[2] N. Madhusudhan and S. Seager. A temperature and abundance retrieval method for exoplanet atmospheres. The Astrophysical Journal, 707(1):24, nov 2009.