The Astrophysical Journal, 961:30 (16pp), 2024 January 20 https://doi.org/10.3847/1538-4357/ad063f

© 2024. The Author(s). Published by the American Astronomical Society.

OPEN ACCESS

To Sample or Not to Sample: Retrieving Exoplanetary Spectra with Variational Inference and Normalizing Flows

Kai Hou Yip[1], Quentin Changeat[1,2], Ahmed Al-Refaie[1], and Ingo P. Waldmann[1]

[1] Department of Physics and Astronomy University College London Gower Street, London WC1E 6BT, UK; kai.hou.yip@ucl.ac.uk

[2] European Space Agency (ESA), ESA Office, Space Telescope Science Institute (STScI), 3700 San Martin Drive, Baltimore, MD 21218, USA

Received 2022 May 13; revised 2023 September 17; accepted 2023 October 18; published 2024 January 11

Abstract

Current endeavours in exoplanet characterization rely on atmospheric retrieval to quantify crucial physical properties of remote exoplanets from observations. However, the scalability and efficiency of said technique are under strain with increasing spectroscopic resolution and forward model complexity. The situation has become more acute with the recent launch of the James Webb Space Telescope and other upcoming missions. Recent advances in machine learning provide optimization-based variational inference as an alternative approach to perform approximate Bayesian posterior inference. In this investigation we developed a normalizing-flow-based neural network, combined with our newly developed differentiable forward model, `Diff-`τ, to perform Bayesian inference in the context of atmospheric retrievals. Using examples from real and simulated spectroscopic data, we demonstrate the advantages of our proposed framework: (1) training our neural network does not require a large precomputed training set and can be trained with only a single observation; (2) it produces high-fidelity posterior distributions in excellent agreement with sampling-based retrievals; (3) it requires up to 75% fewer forward model calls to converge to the same result; and (4) this approach allows formal Bayesian model selection. We discuss the computational efficiencies of `Diff-`τ in relation to `TauREx3`'s nominal forward model and provide a "lessons learned" account of developing radiative transfer models in differentiable languages. Our proposed framework contributes toward the latest development of neural network–powered atmospheric retrieval. Its flexibility and significant reduction in forward model calls required for convergence holds the potential to be an important addition to the retrieval tool box for large and complex data sets along with sampling-based approaches.

Unified Astronomy Thesaurus concepts: Exoplanet atmospheres (487); Neural networks (1933); Bayesian statistics (1900); Posterior distribution (1926); Exoplanets (498); Observational astronomy (1145); Transmission spectroscopy (2133)

1. Introduction

Atmospheric retrieval has become an indispensable tool for astronomers to explain individual observations from transit, eclipse, and phase-curve spectroscopy at both low (e.g., Tinetti et al. 2007; Line et al. 2013, 2014, 2016; Kreidberg et al. 2014; Lee et al. 2014; Haynes et al. 2015; Tsiaras et al. 2016b, 2016a; Evans et al. 2016; MacDonald & Madhusudhan 2017, 2019; Sheppard et al. 2017; Stevenson et al. 2017; Kreidberg et al. 2018; Mikal-Evans et al. 2019; Tsiaras et al. 2019; Pluriel et al. 2020a; Alam et al. 2020; Anisman et al. 2020; Changeat & Al-Refaie 2020; Chubb et al. 2020; Skaf et al. 2020; von Essen et al. 2020; Zhang et al. 2020; Alam et al. 2021; Carone et al. 2021; Changeat & Edwards 2021; Changeat et al. 2021; Edwards et al. 2021; Mugnai et al. 2021; Saba et al. 2022; Sheppard et al. 2021; Swain et al. 2021; Yip et al. 2021; Changeat & Yip 2023; Foote et al. 2022; Mansfield et al. 2022; Mikal-Evans et al. 2022, and references therein) and high resolution (e.g., Brogi & Line 2019; Gibson et al. 2020; Mollière et al. 2020; Seidel et al. 2020; Boucher et al. 2021; Challener & Rauscher 2022; Harrington et al. 2022; MacDonald & Lewis 2022; Meech et al. 2022; Rasmussen et al. 2022). Over the years, the community has come up with a variety of retrieval frameworks, each coupled with different modeling assumptions and sampling techniques (e.g., Irwin et al. 2008; Madhusudhan & Seager 2009; Line et al. 2013; Lavie et al. 2017; Gandhi et al. 2019; Zhang et al. 2019; Lothringer & Barman 2020; Min et al. 2020; Al-Refaie et al. 2021; Cubillos & Blecic 2021). As the number of spectroscopic observations increases with the advent of new space and ground-based observatories, the community has started to look at planetary characterization on a population level (Sing et al. 2016; Barstow et al. 2017; Pinhas et al. 2019; Tsiaras et al. 2019; Mansfield et al. 2021; Roudier et al. 2021; Changeat et al. 2022; Edwards et al. 2023).

At its core, atmospheric retrieval strives to find an atmospheric model that can best explain a given observation. Most contemporary retrieval frameworks formulated the inverse problem in terms of Bayesian statistics, where the free parameters of the physical model are framed as random variables. The probability densities of these random variables (θ) given the observed data (D) are collectively referred to as the posterior distribution, $p(\theta|D)$. Bayes' theorem provides a way to calculate the posterior distribution via the following relation:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}, \tag{1}$$

where $p(D|\theta)$ and $p(\theta)$ represent the likelihood and the prior distributions, respectively. However, the denominator, $p(D)$, or the evidence is intractable in most cases. The community has thus far relied on sampling techniques such as Markov Chain

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Monte Carlo (MCMC) or nested sampling to compute the approximate posterior distribution (see Madhusudhan 2018 for a recent review).

Nevertheless, sampling-based techniques are prohibitively slow when the dimensionality of the problem and quantity of the observed data are large (Zhang et al. 2019). This issue will become increasingly pressing with the increased data volume originating from the recently launched James Webb Space Telescope (JWST; Greene et al. 2016) and other future missions such as Ariel (Tinetti et al. 2021), Twinkle (Edwards et al. 2019), and ELTs from the ground (e.g., Udry et al. 2014). These telescopes are designed to provide hundreds of higher-resolution spectroscopic measurements with wider wavelength coverage. On the other hand, recent investigations have highlighted potential biases associated with some commonly used modeling assumptions, such as isothermal atmospheres, constant with altitude chemistry, or the 1D plane–parallel approximation (e.g., Rocchetto et al. 2016; Changeat et al. 2019; Pluriel et al. 2020b; Changeat et al. 2020; MacDonald et al. 2020; Ih & Kempton 2021), prompting the need for more realistic treatment of the atmosphere (e.g., Changeat & Al-Refaie 2020; Feng et al. 2020; Irwin et al. 2020). This will inevitably increase both the computational cost and complexity (Changeat et al. 2021) of the forward model. The increase in both quantity of data and model complexity signals the need for alternative approaches to computing the posterior distribution.

Recent years have seen a surge in machine learning (ML)-based techniques being applied to many areas within exoplanetary science, from data detrending (e.g., Morvan et al. 2020; Gebhard et al. 2020, 2022; Krick et al. 2020; Nikolaou et al. 2020; Morvan et al. 2021), to planet detection (e.g., Shallue & Vanderburg 2018; Yip et al. 2019; Yu et al. 2019; Valizadegan et al. 2021), and to planet characterization (e.g., Márquez-Neila et al. 2018; Zingales & Waldmann 2018; Cobb et al. 2019; Waldmann & Griffith 2019; Hayes et al. 2020; Oreshenko et al. 2020; Yip et al. 2020; Ardevol Martinez et al. 2022; Haldemann et al. 2023; Himes et al. 2022). In 2022, the topic of planet characterization has also been featured as a competition at the Neural Information Processing Systems (NeurIPS; Changeat & Yip 2023; Yip et al. 2022) conference.

Variational inference (VI) is a widely studied approach in the field of ML used to provide approximate posterior distributions for a large and high-dimensional data set with reduced computational demand compared to Markovian sampling approaches (e.g., Blei et al. 2016; Buchholz et al. 2018; Fellows et al. 2018; Shu et al. 2018; Zhang et al. 2019, 2021; Argelaguet et al. 2020; Fortuin et al. 2020; Friston et al. 2020; Lopez-Alvis et al. 2021; Karchev et al. 2022; Lopez-Alvis et al. 2022). However, variational methods require models that can provide their gradient with respect to some (input) parameters. The field has recently explored different applications of differentiable physical models. Differentiable models open up the possibility to construct "physics-aware" neural networks, a type of network that is explicitly constrained by physical laws (e.g., Raissi et al. 2019; Chen et al. 2020; Morvan et al. 2021; Amini Niaki et al. 2021; Cai et al. 2021; Haghighat et al. 2021; Viana & Subramaniyan 2021; Cuomo et al. 2022). For instance, Kawahara et al. (2022) used Hamiltonian Monte Carlo (HMC), a gradient-informed Monte Carlo sampling algorithm (Duane et al. 1987; Hoffman & Gelman 2011), to perform atmospheric retrieval of exoplanets on high-resolution spectroscopic data. Others have also applied HMC to speed up light curve fitting (e.g., Agol et al. 2021; Foreman-Mackey et al. 2021).

Here we present the following contributions:

1. We present `Diff-`τ, a `Tensorflow`-based fully differentiable atmospheric forward model, based on the implementation of `TauREx3` (Al-Refaie et al. 2021, 2022).
2. For the first time, we introduced VI as a more efficient alternative to perform atmospheric retrieval.
3. As our framework only requires a single data instance during training time, there is no need for a large library of spectra for pretraining.
4. We show that our framework formally takes into account the uncertainties associated with the observations and is able to reproduce physically motivated correlations between atmospheric parameters.
5. Our Bayesian neural network is capable of producing posterior distributions on par with distributions produced from sampling-based approaches.

2. Overview

In this investigation our core aim is to explore an alternative approach to the conventional, sampling approach with the use of modern deep learning techniques. For simplicity we will denote the conventional, sampling-based approach simply as `NS-retrieval` and our proposed approach as `VI-retrieval`. Our approach involves three core components: a differentiable physical model, a formulation of VI, and a normalizing flow (NF)-based neural network. Here we will provide a top-level overview of how the three components interact with each other, and see Figure 1 for a schematic overview of the `VI-retrieval`.

Instead of relying on sampling to map the unknown posterior distribution (as one normally does with, e.g., MCMC-based approaches), `VI-retrieval` relies on finding a best-fit surrogate distribution to the actual posterior distribution through optimization. The use of optimization-based techniques means that `VI-retrieval` can be orders of magnitudes faster than sampling-based approaches, especially on high-dimensional problems. However, there are two implementation difficulties that prevented the widespread use of variational methods in the field of exoplanetary atmospheres: (1) Gradients. Many optimization procedures demand complete knowledge of the gradient flow within a computational graph (i.e., a neural network), contemporary atmospheric forward models break the flow as they are undifferentiable; and (2) surrogate distributions: the chosen distribution (usually a multinomial Gaussian distribution) is often too simplistic to represent the actual, underlying posterior distribution.

To circumvent the above difficulties, we built `Diff-`τ with the `Tensorflow` library, and utilize its automatic differentiation capabilities to compute gradients of the forward model. At the same time, we implemented an NF-based neural network, a deep learning approach that can transform a simple, "seed" distribution (such as a multinomial Gaussian) to arbitrarily complex distributions.

In the following sections we will explain the theoretical background behind each technique, and in the latter part of the paper we will demonstrate how these concepts are linked to

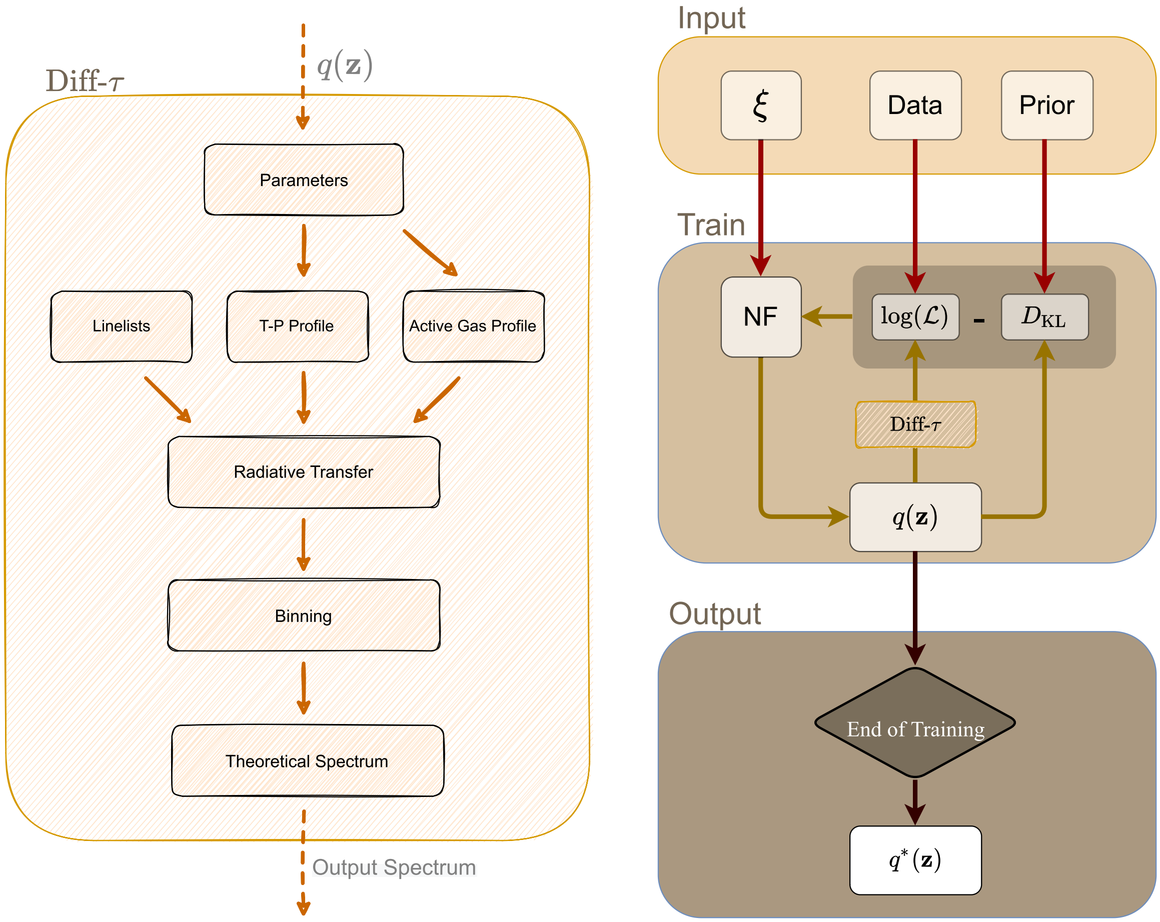

Figure 1. Overview of our proposed method. The left panel shows the structure of Diff-τ. It is almost identical to a conventional forward model, in the sense that it takes in a set of physical parameters and outputs the corresponding theoretical spectrum, but with the additional ability to provide gradients. The right panel shows a schematic of our proposed VI-retrieval. We provide a simple (user-defined) seed distribution ξ, observed data, and our prior bounds as inputs to the optimization loop (Train). The NF-based neural network, i.e., NF, is tasked to transform ξ to a surrogate distribution $q(z)$. We employ VI to approximate the typically intractable posterior distributions, where the network (NF) must learn the best transformation to optimize the ELBO (controlled by $\log(\mathcal{L})$ and $D_{\rm KL}$). To calculate log ($\mathcal{L}$), we used Diff-τ to transform samples from $q(z)$ into spectra and compared with the observed data. Once the training is completed, the trained model is able to provide a $q * (z)$ that best approximates $p(z|\boldsymbol{x})$.

each other in practice by providing retrieval examples for three different scenarios.

2.1. VI

The mathematical theory of VI and its application in the field of ML have been extensively discussed in Blei et al. (2016). There are ongoing efforts to investigate the statistical implication of using VI for parameter estimation (e.g., Chérief-Abdellatif & Alquier 2018; Pati et al. 2018). Here we provide a brief overview of the methodology.

Given an observed spectrum $\boldsymbol{x}$ defined by the transit depths (x_i) and associate uncertainties (σ_i) in each spectral bin i, the goal of atmospheric retrieval is to find the posterior distribution $p(z|\boldsymbol{x})$ of the set of latent variables (z) that can best describe the observation under a specific atmospheric model assumption $\mathcal{M}$. Instead of approximating the unnormalized $p(z|\boldsymbol{x})$ via sampling, variational methods approximate the distribution by finding a best-matching surrogate distribution via optimization.

Suppose we have a family of probability distributions $\mathbb{Q}$ parameterized by some latent variables z. The optimal distribution (best-matching surrogate distribution to $p(z|\boldsymbol{x})$) is the one that minimizes the statistical distance to $p(z|\boldsymbol{x})$. A common choice is to compute the Kullback–Leibler (K-L) divergence between the two probability density functions (PDFs), i.e.:

$$q^*(z) = \underset{q(z)\in\mathbb{Q}}{\arg\min}\ D_{\rm KL}[p(z|\boldsymbol{x})||q(z)], \tag{2}$$

$$= \mathbb{E}\left[\log\left(\frac{p(z|\boldsymbol{x})}{q(z)}\right)\right]. \tag{3}$$

K-L divergence measures the relative entropy between two PDFs with range [0, ∞]. A score of 0 means that the two distributions contain identical information, and any (positive) deviation from 0 means the two distributions become increasingly different from one another (Kullback & Leibler 1951). K-L divergence can be computed analytically if one knows the functional form of both PDFs. In cases when the functional forms of one or both PDFs are unknown, as we will see below, numerical approaches must be sought to approximate the divergence.

However, Equation (3) itself cannot be the objective function for our optimization task, as we do not have any knowledge of $p(z|\boldsymbol{x})$ with which to begin. To negate the dependence on the unknown true posterior distribution, VI provides an alternative

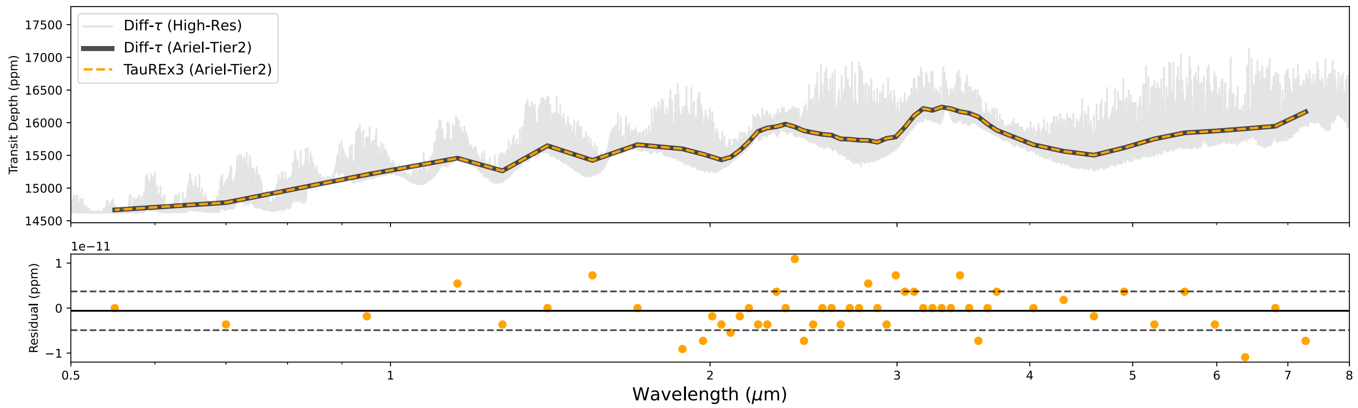

Figure 2. Top panel: comparing the outputs from `Diff-τ` in native (light gray) and binned (Ariel) resolution (orange) and `TauREx3` binned to the same resolution (black dots and dashed line). The lower panel shows the residuals between the two binned spectra. The gray lines represent the mean (solid line) and 1σ standard deviation (dashed lines).

formulation, the Evidence Lower BOund (ELBO):

$$\mathrm{ELBO} = \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{z})] - D_{\mathrm{KL}}[q(\boldsymbol{z})\|p(\boldsymbol{z})], \quad (4)$$

where the first term is the expected value of the log-likelihood $\log(\mathcal{L})$ and the second term is the K-L divergence between $q(\boldsymbol{z})$ and the prior distribution $p(\boldsymbol{z})$. This formulation can be understood, in terms of Bayesian statistics, as a "tug of war" between the likelihood function and our prior belief on the distribution. We included a detailed discussion on ELBO and its link to Equation (3) in the Appendix.

Unlike Equation (3), the ELBO does not require any knowledge of the intractable evidence $p(\boldsymbol{x})$ and therefore it can be computed analytically. We can hence find the optimal distribution $q*(\boldsymbol{z})$ by minimizing the ELBO. This optimization technique is well studied in the field of deep learning (Kingma & Welling 2013; Kingma et al. 2016). Most contemporary neural networks are trained to minimize a given loss function (e.g., ELBO in our case) by relying on a combination of gradient descent and back-propagation algorithms. Modern deep learning libraries such as `Tensorflow` (Abadi et al. 2015), `PyTorch` (Paszke et al. 2019), and `JAX` (Bradbury et al. 2018) provide easy access to model training and evaluation. In this study, our implementation will be solely based on the `Tensorflow` framework, but other deep learning frameworks can similarly be used (Kawahara et al. 2022).

2.2. Differentiable Forward Model, `Diff-τ`

`Diff-τ` is an atmospheric forward model built entirely within the `Tensorflow` framework (Abadi et al. 2015). We followed the forward model formulation as specified in Al-Refaie et al. (2021) to construct `Diff-τ`, with minor modifications to comply with the `Tensorflow` framework. As the code is largely based on `TauREx3`, it provides excellent agreement between the two forward models; see Figure 2 for an empirical comparison between the two. We leverage the built-in automatic differentiation functionality (Baydin et al. 2018), which has the ability to differentiate (almost) any functions automatically without the need to specify the corresponding derivative form explicitly. This is immensely helpful as atmospheric models are a mixture of different physical processes, and deriving the respective derivative forms analytically can be a time consuming and nontrivial task.

2.3. NFs

The stringent requirement of a predefined family of distributions, $\mathbb{Q}$, presents a major limitation in using VI. For many real-life scenarios, the desired posterior distributions are rarely Gaussian or well defined. We implemented an NF-based neural network to break the limitations of Gaussianity in $q^*(\boldsymbol{z})$ in Equation (4) by transforming it into an arbitrarily complex probability distribution.

NF describes a mechanism to "craft" a complex, multimodel distribution from a simple, "seed" distribution (Rippel & Adams 2013; Rezende & Mohamed 2015; Kobyzev et al. 2021). This can usually be a distribution in the exponential family (e.g., a Gaussian) or a uniform distribution.

Suppose we have an invertible function g, such that we can transform a random variable $\boldsymbol{\xi} \sim p_{\xi}$ into another random variable $\boldsymbol{y} \sim p_{y}$ using $\boldsymbol{y} = g(\boldsymbol{\xi})$. The probability density p_{y} of the random variable $\boldsymbol{y}$ can be computed using the change of variable formula, i.e.:

$$p_{y}(\boldsymbol{y}) = p_{\xi}(f(\boldsymbol{y}))|\det \mathrm{D}(f(\boldsymbol{y}))|, \quad (5)$$

where f is the inverse of g, i.e., $f \equiv g^{-1}$, $\mathrm{D}f(\boldsymbol{y})$ is the Jacobian of f, i.e., $\mathrm{D}f(\boldsymbol{y}) = \frac{\partial f}{\partial y}$, and $\mathrm{D}g(\boldsymbol{\xi})$ is the Jacobian of g, $\mathrm{D}g(\boldsymbol{\xi}) = \frac{\partial g}{\partial \xi}$. In terms of generative models, the invertible function $g(.)$ is a generator that "pushes" forward the seed distribution to a more complex distribution function (the generative direction). On the other hand, the function $f(.)$ moves in the opposite direction, transforming it back to a simple, "normalized" distribution (normalizing direction). It has been shown that one can generate or craft any form of distribution p_{y} from any base distribution p_{ξ}, given that the generator g can be arbitrarily complex (Bogachev et al. 2005; Medvedev 2008).

Up until now we have only shifted the problem from crafting an arbitrarily complex density function to an arbitrarily complex generator function. Fortunately, invertible functions (or bijections) have a nice property—the composition of invertible functions is itself invertible, meaning that one can build a successively more complicated function by chaining nonlinear invertible functions together, i.e.:

$$g = g_{N} \circ g_{N-1} \cdots \circ g_{1}, \quad (6)$$

where $g_1 ... g_N$ is a set of N bijective function. Similarly, g has an inverse:

$$f = f_1 \circ f_2 ... \circ f_N , \tag{7}$$

and conveniently, the determinant of the Jacobian D$f(\boldsymbol{y})$ is the product of individual determinant of the Jacobians D$f_i(\boldsymbol{y})$, i.e.:

$$\mathrm{D}f(\boldsymbol{y}) = \prod_{i}^{N} \mathrm{D}f_i(\varphi_i). \tag{8}$$

We denote φ_i as the resultant vector of the ith intermediate flow, i.e., $\varphi_i = g_i \circ \cdots \circ g_1(\boldsymbol{\xi}) = f_i \circ \cdots \circ f_N(\boldsymbol{y})$, where $\varphi_N = \boldsymbol{y}$.

In the context of our investigation, we will transform our seed distribution $\boldsymbol{\xi} \sim p_{\boldsymbol{\xi}}$ into $\boldsymbol{y} \sim p_{\boldsymbol{y}}$ and treat $\boldsymbol{y} \sim p_{\boldsymbol{y}}$ as our surrogate distribution $z \sim q(z)$, i.e., $\boldsymbol{y} \sim p_{\boldsymbol{y}} \equiv z \sim q(z)$.

However, these intermediate functions must—by definition—be diffeomorphic, meaning they must be bijective and differentiable (including their inverses). In recent years the field has put significant effort in constructing bijectors that conform with these restrictions but remain sufficiently expressive and computationally efficient even in high-dimensional problems such as images (Louizos & Welling 2017; Rothfuss et al. 2019; Nielsen et al. 2020; Wu et al. 2020; Zhang & Chen 2021) and notable bijector architectures including MADE (Germain et al. 2015), Masked Autoregressive Flow (Papamakarios et al. 2017), NICE (Dinh et al. 2014), RealNVP (Dinh et al. 2016), Sylvester NF (Berg et al. 2018), FFJORD (Grathwohl et al. 2018), Glow (Kingma & Dhariwal 2018), and NSF (Durkan et al. 2019). NF has proven to be a highly successful approach in a wide range of applications, including audio synthesis (Oord et al. 2018; Prenger et al. 2019; Aggarwal et al. 2020), text translation (Jin et al. 2019; Izmailov et al. 2020), anomaly detection (Rudolph et al. 2021; Gudovskiy et al. 2022), time series forcasting (Schmidt & Simic 2019; Rasul et al. 2020; Feng et al. 2022), and image generation (Grathwohl et al. 2018; Kingma & Dhariwal 2018; Lugmayr et al. 2020).

3. Implementation

3.1. Flow-based Model Setup

We implemented the inverse autoregressive flow (IAF; Kingma et al. 2016) as a default bijector unit in our NF-based neural network. To perform the transformation, we chained $N = 10$ bijector units together, each controlled by a two-layer, densely connected neural network with 64 hidden units and ReLU activation (He et al. 2015) in each layer. To stabilize the network, we followed Kingma et al. (2016) and added a batch normalization layer (Ioffe & Szegedy 2015) after each bijector unit. We used Adam (Kingma & Ba 2014) as our optimizer with a scheduled learning rate (see below), and the rest of the settings are kept as default from `Tensorflow`. We used a multidimensional uniform distribution as our seed distribution, $\boldsymbol{\xi}$.

3.2. ELBO Formulation

We begin by describing our implementation to the general ELBO formulation in Equation (4). The ELBO consists of two terms—the log-likelihood term and the prior term.

We defined the log-likelihood as an additive log-Gaussian PDF, i.e.:

$$\mathbb{E}[\log p(\boldsymbol{x}|z)] = \mathbb{E}[\log(\mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\mu}}, \boldsymbol{\sigma})], \tag{9}$$

$$= \mathbb{E}\left[\log\left(\frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}}\exp\left(-\frac{1}{2}\frac{(\boldsymbol{x} - \hat{\boldsymbol{\mu}})^2}{\boldsymbol{\sigma}^2}\right)\right)\right], \tag{10}$$

where $\hat{\boldsymbol{\mu}} = \texttt{Diff} - \tau(z)$ is a forward model generated by `Diff`-τ binned to the spectral resolution of the data and $\boldsymbol{\sigma}$ is the observed uncertainty. This formulation is the same as the likelihood formulation currently employed by many retrieval frameworks.

The second term, $D_{\mathrm{KL}}[q(z)||p(z)$, computes the K-L divergence between the surrogate distribution and the prior distribution (as defined by the user), it can be expanded in a similar fashion as Equation (3):

$$D_{\mathrm{KL}}[q(z)||p(z)] = \mathbb{E}\left[\log\left(\frac{q(z)}{p(z)}\right)\right]. \tag{11}$$

The functional form of $p(z)$ can be easily determined (as it is the user-defined priori distribution). The surrogate distribution, $q(z)$ has undergone multiple transformations by the flow-based neural network. Instead of computing the K-L divergence analytically, which requires knowledge of the full analytical form of the distribution, we can approximate the K-L divergence via Monte Carlo sampling, as it is not costly to sample repeatedly from the transformed surrogate distribution (Kingma et al. 2016). For this investigation, we sample both the prior and surrogate distributions 10,000 times to approximate the value of the K-L divergence. Consistent with existing literature on exoplanetary retrieval, we imposed a uniform prior on all physical parameters, but we note that any proper prior probability distribution can be used.

Instead of optimizing the ELBO objective in its actual formulation, we adopted the weight-annealing approach from Sun et al. (2022) to optimize a modified ELBO objective, i.e.:

$$\mathrm{loss} = \frac{1}{\beta}\mathbb{E}[\log p(\boldsymbol{x}|z)] - D_{\mathrm{KL}}[q(z)||p(z)], \tag{12}$$

where $\beta = \max(1, \beta_0^{\frac{\mathrm{epoch}}{\tau}})$ and β_0 and τ represent the weight constant and decay constant, respectively. This formulation prevents the neural network from converging to bad local minima at the start of the training, and encourages it to explore different solutions before converging. As training progresses, the objective function will slowly converge back to the original ELBO formulation.

3.3. The Role of `Diff`-τ

The formulation of the log-likelihood function (Equation (10)) prompts the need to compare our input observation with theoretical spectra. The role of `Diff`-τ can be seen simply as a deterministic transformation, i.e., from physical parameters to spectra. As part of the optimization process, the network must learn to adhere to physical laws imposed by the forward model. In the end, the network will produce outputs and correlations that are physically constrained in order to achieve better scores. In other words, the network's behavior becomes physically motivated and explainable.

3.4. Training Procedure

At training time, we define the seed (user-defined) distribution $\boldsymbol{\xi} = U\,[\text{min,max}]$,[3] and our flow-based neural network (the generator) will then transform it into a surrogate distribution $q(z)$ through our chain of bijections. We then sample from $q(z)$ and generate the atmospheric forward model using `Diff-τ`. At each iteration, we will sample five times from the surrogate distribution and compute the average (modified) ELBO objective, Equation (12). The entire process is repeated until the optimization has converged or is terminated. We have implemented an `Early Stopping` procedure to stop the optimization process if the loss value (ELBO) does not improve over 50 epochs. As we have adopted a weight-annealing objective, the stopping criteria will only become effective in the later stages of the optimization, when the modified objective converges back to the original ELBO formulation (weights stay unity).[4] The best model is used to produce the results in Section 4. As for the learning rate, we have implemented a cyclic learning rate as suggested by Smith (2015) and Himes et al. (2022). Our experiments are consistent with their claims of improved training performance and we show that our method outperforms the constant learning rate and step-wise decaying learning rate[5] in terms of loss value as well as speed of convergence.

Once trained, the generator network is decoupled from the framework. At inference time, we initiate a seed distribution and pass it to transform the seed distribution into our best-matching surrogate distribution.

4. Application

In this section we will perform atmospheric retrieval with `NS-retrieval` and `VI-retrieval`. We will demonstrate the technique through three cases:

1. A real Hubble Space Telescope/Wide Field Camera 3 (HST/WFC3) observation of the hot Jupiter HD 209458 b.
2. A simulated Ariel Tier 2 observation of the hot Jupiter HD 209458 b.
3. A simulated observation (0.5–15 μm) with an N-point temperature profile (Waldmann et al. 2015; Changeat et al. 2021) of the hot Jupiter WASP-43 b.

4.1. Atmospheric Model Setup for Cases I and II

We have taken values from Tsiaras et al. (2016b) as our reference values for the HD 209458 system. In both cases, we assumed blackbody emission for the host star of $T_{\text{eff}} = 6065$ K and stellar radius $R_* = 1.155\ R_\odot$. We also assumed a primary atmosphere (dominated by H and He, ratio = 0.175) at solar abundance. We divided the atmosphere into 70 atmospheric layers over the 10^{-5} and 10^6 Pa pressure range (evenly spaced in logarithmic scale) and furthermore assumed an isothermal *T–P* profile and an iso-abundance chemical profile. Table 1 shows the fitted input parameters for each case and their corresponding ground truths, prior bounds, and formulation/line list references. The atmospheric setup of Case III can be found in Section 4.4.

4.2. Case I: HST/WFC3 Observation of HD 209458 b

Case I aims to demonstrate our method's applicability to actual data. The transmission spectrum is observed with the HST/WFC3 G141 grism and processed by `Iraclis` (Tsiaras et al. 2016b, 2019). The detrending process is described in details in Tsiaras et al. (2016b). The basic atmospheric setup follows Section 4.1. For this case we are assuming a hydrogen/helium-dominated atmosphere, with opacities from trace gas absorption and Mie scattering clouds[6] (Lee et al. 2013). We ran the optimization procedure for 2000 epochs, with the convergence parameters, β_0 and τ, set to 100 and 200, respectively. These values are determined with a coarse hyperparameter search between $\beta_0 = [100,\ 1000]$ and $\tau = [100,\ 1000]$.

4.3. Case II: Ariel Tier 2 Observation of HD 209458 b

In the second case we would like to understand the ability of our proposed framework to retrieve the ground truth values and showcase the flexibility of our framework to switch to different atmospheric assumptions and spectral resolutions. We used `ArielRad` (Mugnai et al. 2020) to simulate the expected noise level for each wavelength channel for HD 209458 b at Ariel Tier 2 resolution. In terms of atmospheric chemistry, we adhered to the same setup as described in Section 4.1 and include five trace gases: H_2O, CH_4, CO, CO_2, and NH_3. We chose this set of molecules due to their expected contribution in the wavelength range considered, and because they have been successfully detected in hot Jupiter atmospheres. We ran the optimization procedure for 2000 epochs, with β_0 and τ set to 100 and 300, respectively.

4.4. Case III: N-Point T–P Profile Retrieval with WASP-43 b

We simulated the atmosphere of a WASP-43 b-like planet observed at a customized wavelength range, i.e., from 0.5 to 15 μm. The atmospheric model setup largely followed Section 4.1, but with 50 atmospheric layers instead of 70 layers. As for the stellar and planetary parameters, we followed Hellier et al. (2011) and set $T_{\text{eff}} = 4400$ K, $R_* = 0.6\ R_\oplus$, $M_p = 1.78\ M_J$, and $R_p = 0.93\ R_J$. We included $\log_{10}(H_2O)$ and $\log_{10}(CH_4)$ as trace gases in the atmosphere, with their abundances set at −4 and −3, respectively. As for the temperature–pressure profile, we adopted the dayside temperature profile of WASP-43 b retrieved by Changeat et al. (2021). We then generated a (binned) atmospheric model with the above settings, and added 100 ppm Gaussian white noise to produce an observation. The prior bounds of the parameters follow Table 1 apart from planet temperature, where we expanded the bounds to [100, 3000]. The observation remains the same for all retrievals performed in Case III.

As for the retrieval settings, we performed an N-point retrieval with each temperature point located at a fixed, predefined pressure point. The top and bottom pressure points are fixed at 10^{-5} and 10^6, respectively throughout the

[3] All bounded between [−1, 1].

[4] The weight-annealing strategy reduces the contribution from the much larger likelihood function initially and slowly returns it to its original value as training progresses. The loss value will almost always increase as the weights slowly increases back to unity).

[5] The learning rate will reduce if the loss value did not decrease after a certain number of epochs.

[6] We note that the alternative and commonly used flat cloud model (i.e., a constant pressure opacity cut-off) is inherently not differentiable and not physically viable. We therefore chose to include the differentiable Mie scattering formulation in `Diff-τ`.

Table 1
Fitted Parameters for Cases I and II, along with Their Corresponding Ground Truths, Prior Bounds, and Molecular Line List References for Each Trace Gas Species

Parameters	Ground Truth*	Case I	Case II	Priors	Reference
T_p (K)	1449	✓	✓	$\mathcal{U}(100, 2500)$	N/A
R_p (R_J)	1.359	✓	✓	$\mathcal{U}(0.5, 2.5)$	N/A
$\log_{10}(H_2O)$	−5	✓	✓	$\mathcal{U}(-12, -2)$	Polyansky et al. (2018)
$\log_{10}(CH_4)$	−5		✓	$\mathcal{U}(-12, -2)$	Yurchenko et al. (2017)
$\log_{10}(CO_2)$	−5		✓	$\mathcal{U}(-12, -2)$	Yurchenko et al. (2020)
$\log_{10}(NH_3)$	−8		✓	$\mathcal{U}(-12, -2)$	Yurchenko et al. (2011)
$\log_{10}(CO)$	−8		✓	$\mathcal{U}(-12, -2)$	Li et al. (2015)
$\log\ _{10}(\chi_{\rm mie}^{\rm lee})$	N/A	✓		$\mathcal{U}(-40, -4)$	Lee et al. (2013)
q_{mie}^{lee}	N/A	✓		$\mathcal{U}(1, 99)$	Lee et al. (2013)
$\log\ _{10}(a_{\rm mie}^{\rm lee})$	N/A	✓		$\mathcal{U}(-3, 1)$	Lee et al. (2013)

Note. The check mark (✓) indicates whether the parameter is included in each case. The ground truths are only available for Case II.

investigation, while the (pressure) points in between them vary in accordance to the number of retrieved temperature (pressure) points. In all cases the pressure points are separated equally (in log-pressure space). Other retrieved parameters included R_p, $\log_{10}(H_2O)$, and $\log_{10}(CH_4)$. In summary, the total number of free parameters is N (temperature points) + 3. The network setup mostly follows Section 3.4. We ran the optimization procedure for 5000 epochs, with β_0 and τ set as 100 and 300, respectively.

5. Results

5.1. Cases I and II

In both cases (Figures 3 and 4), `VI-retrieval` (in yellow) is able to converge to very similar results to `NS-retrieval` (in red). We observed excellent agreement between both methods and the percentile values[7] are within 1σ of each other.

As for the shape of the posterior distribution, the surrogate distribution is able to reproduce peaked, Gaussian-like distributions whenever there are sufficient constrains from the observations (i.e., the radius of the planet at 1 bar pressure), and, at the same time, produces a uniform-like distribution (with upper limit) when the free parameters are unconstrained by the observation (e.g., CO in Case II), conforming with our uniform prior. In addition to this, the surrogate distribution provides a faithful reproduction of the covariances between the free parameters. These correlations result from the interactions between the free parameters through the formulation of the radiative transfer equations. They are most notable in Case II, when the high signal-to-noise ratio observation of Ariel allows better constraints on the free model parameters.

Apart from the similarities, there are also noticeable differences between the two retrievals. For instance, `VI-retrieval` is less adept at capturing “cliff-like” distributions (e.g., $\log\ _{10}(\chi_{\rm mie}^{\rm lee})$), when there is a sharp upper or lower bound on a parameter. There are also instances (such as T_p in Case I) when the surrogate distribution is only able to capture “part” of the conditional distribution obtained via `NS-retrieval`. The difference could be due to the imperfect optimization process. The loss function (ELBO) is a balance between the log-likelihood term and the prior term. Minimizing one term will always come at the expense of maximizing the other. This situation is aggravated by the fact that the variability of the log-likelihood term is orders of magnitude higher than that of the prior term, which means that the training will always be driven by the former term and produce sharply peaked distributions. Our modified objective function (Equation (12)) aims to alleviate the imbalance between the two terms by explicitly lowering the contribution from the log-likelihood term during the start of the training. However, the underdispersed situation will likely return as the formulation slowly converges back to the original ELBO form, as seen in both cases. An alternative remedy is to increase the contribution from the prior term permanently, which will inevitably increase the estimated uncertainties and makes it harder to compare to the conventional approach (i.e., `NS-retrieval`).

5.2. Case III

Figure 5 shows the outcome of the simulated WASP-43 b observation. Figure 6 shows the results of performing N-point retrievals from six free parameters to 12 free parameters. The first row compares the retrieved *T–P* profiles from both approaches (blue: `NS-retrieval`, yellow: `VI-retrieval`). The second row compares the number of forward models calls (Ncalls) required by each approach. In all cases `VI-retrieval` (in yellow) requires six times fewer Ncalls compared to `NS-retrieval` (blue). The third row compares the log-evidence obtained by each approaches. In all cases, the log-evidence obtained by `VI-retrieval` never goes higher than the ones obtained via `NS-retrieval`, which is consistent with the formulation of ELBO. Both approaches show a declining trend in log-evidence as the number of retrieved points increases, with `VI-retrieval` declining more rapidly than `NS-retrieval`. The faster decline in log-evidence may be affected by the increase in dimensionality (free parameters), which makes the accurate estimation of the log-evidence harder.

6. Discussion

6.1. Limitations of Grid-based Learning

Most contemporary ML-based atmospheric retrievals are trained in a supervised fashion with a large grid of simulated spectra produced by a forward model. We broadly refer these models as grid-based models here. This kind of training procedure takes away the computational burden of having to generate thousands to millions of forward models on the fly during model deployment and makes these models

[7] Defined as the 16th and 84th percentiles, indicated by the dashed line in Figures 3 and 4.

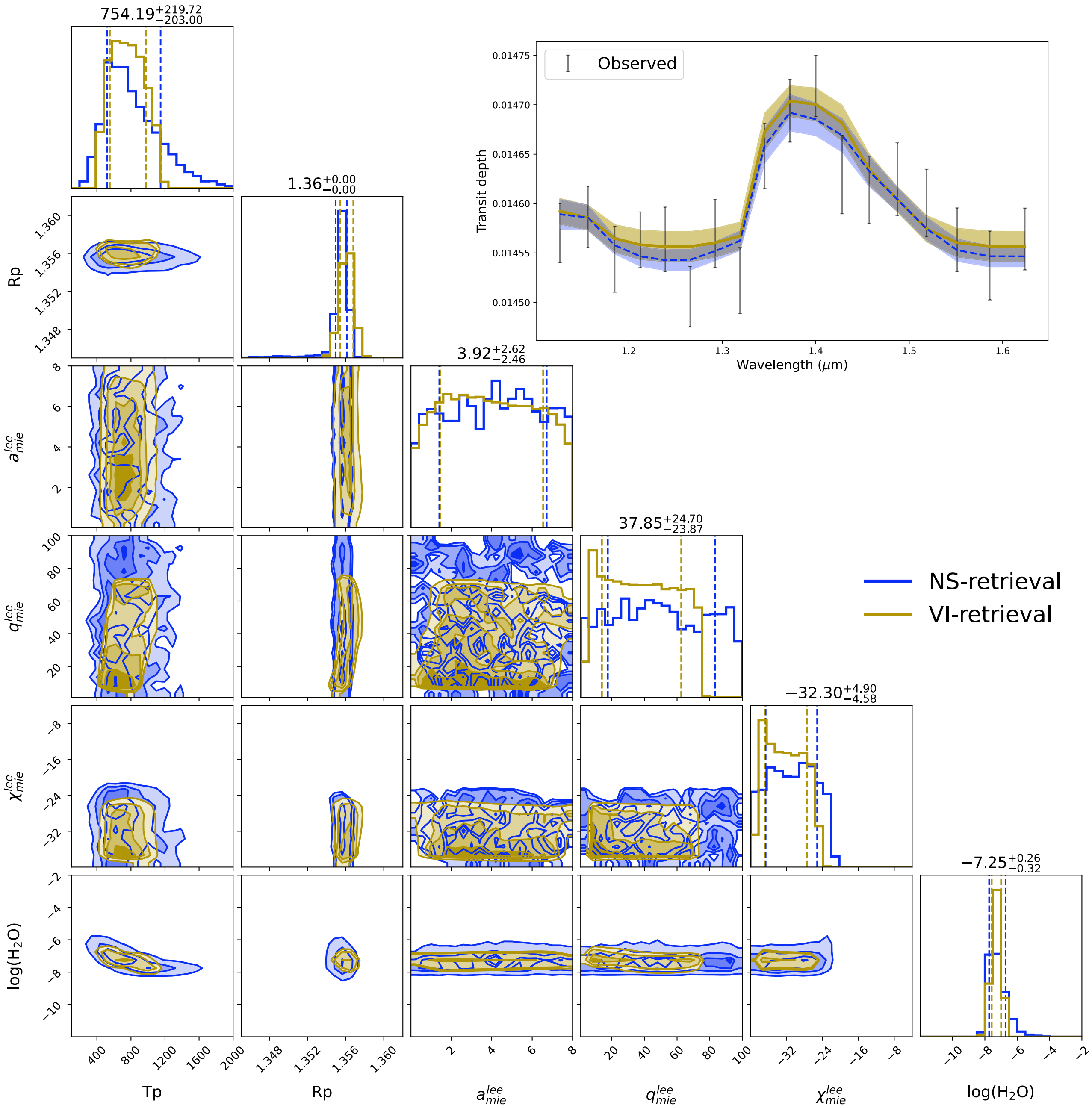

Figure 3. Posterior distributions of the HD 209458 b WFC3/G141 observation obtained from `NS-retrieval` (blue) and `VI-retrieval` (yellow). The top right corner is an empirical comparison between the two best-fitted spectra as obtained by the two methods. The shaded area shows the 1σ spread of the respective retrieval approaches.

computationally fast for applications within their trained domain. In other words, these models offload the computational burden of atmospheric retrievals to the model training stage, with the fully trained model being quick to run at inference. However, this approach comes with three limitations: (1) model generalizability, (2) lack of a Bayesian framework, and (3) a lack of model interpretability.

Generalizability. Any changes to the underlying model, spectral range, or resolution, will hinder the model's performance, and in some cases, will require a full recomputation of the training data from scratch and retraining the model (Márquez-Neila et al. 2018; Zingales & Waldmann 2018; Cobb et al. 2019; Yip et al. 2020; Ardevol Martinez et al. 2022; Haldemann et al. 2023). Such a scenario can be triggered by anything as simple as adding an extra molecule that is previously not present in the training data. These limitations can be alleviated to a certain extent by training a surrogate forward model as done in Himes et al. (2022). Their model is tasked to produce synthetic spectra at very high resolution, and the output can subsequently be down-sampled to any appropriate spectral range and resolution when required. Nevertheless, the model is not immune to changes to the underlying atmospheric assumptions and will likely need to be retrained in those cases.

Lack of a Bayesian framework. Most contemporary retrievals aim to map the Bayesian posterior distribution. In contrast, most ML models applied in the field of atmospheric characterization are formulated to perform maximum

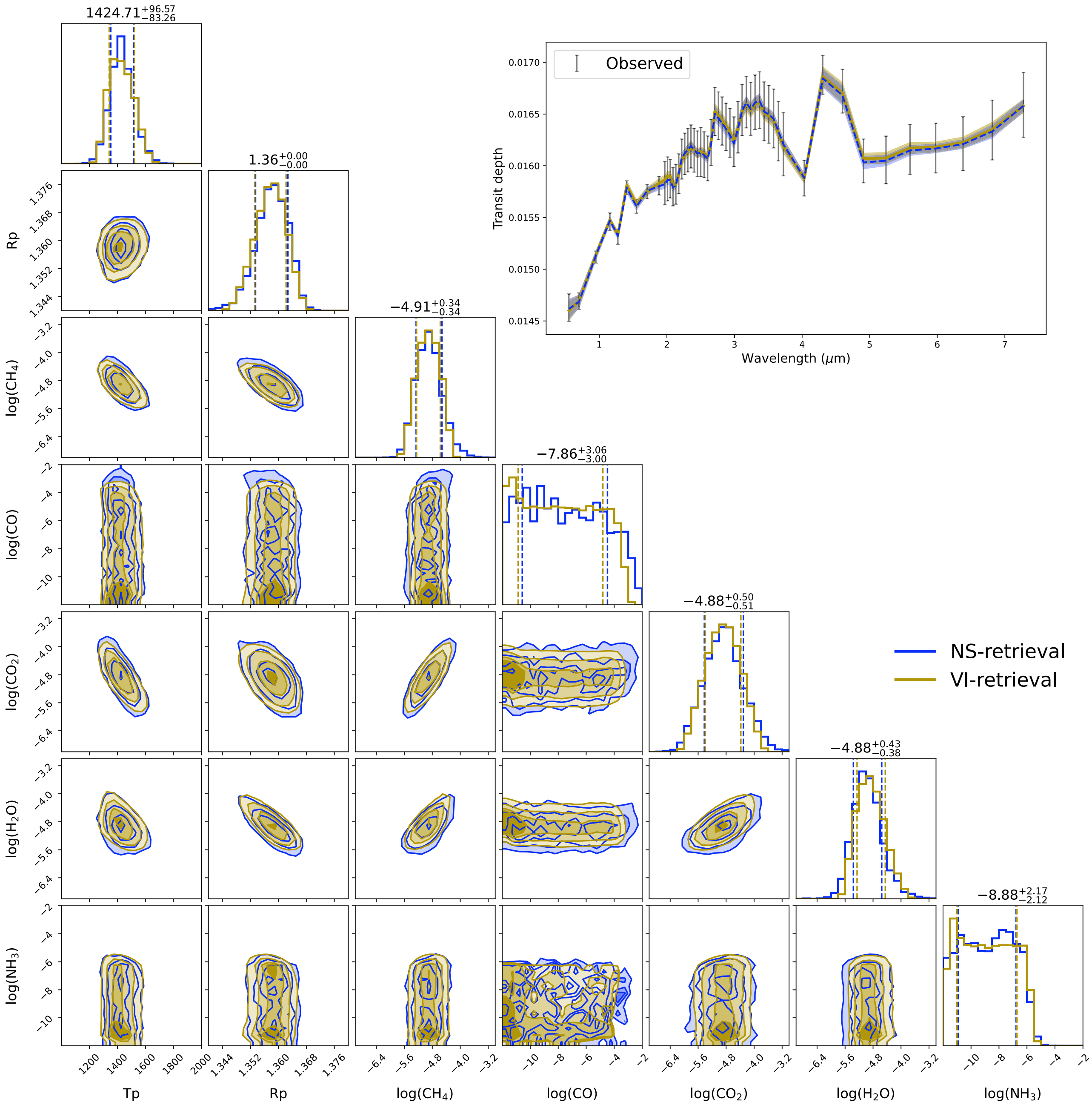

Figure 4. Posterior distributions of HD 209458 b at Ariel Tier 2 resolution obtained via `NS-retrieval` (blue) and `VI-retrieval` (yellow). The top right corner is an empirical comparison between the two best-fitted spectra as obtained by the two methods. The shaded area shows the 1σ uncertainty of the respective retrieval methods.

likelihood estimation (e.g., Márquez-Neila et al. 2018; Yip et al. 2020; Ardevol Martinez et al. 2022; Haldemann et al. 2023). The difference between these two objectives presents an obstacle when trying to compare the outputs from the two methodologies. Other non-Bayesian approaches includes `ExoGAN`, in which case the generator is trained to optimize the adversarial loss (Zingales & Waldmann 2018), but this loss function does not guarantee a robust computation of $P(\theta|D)$. There are other ways to bypass this limitation. For instance, Cobb et al. (2019) and Ardevol Martinez et al. (2022) use an ensemble of neural networks to approximate the conditional distribution. Himes et al. (2022) bypassed this constraint by running a traditional retrieval with an ML-based surrogate forward model.

Interpretability. Interpretability varies from one model to another. Learning algorithms such as linear regression and decision trees are some of the most transparent algorithms, but with limited modeling capability. Deep learning algorithms sits at the opposite end of the spectrum, with extensive learning capabilities, but at the cost of very limited interpretability. Most grid-based learning models (Márquez-Neila et al. 2018; Zingales & Waldmann 2018; Cobb et al. 2019; Yip et al. 2020; Ardevol Martinez et al. 2022; Haldemann et al. 2023) use more complex models to learn the covariance between parameters from a large grid of spectral examples. However, the loss of interpretability means that it becomes hard for users to understand when, where, and how the algorithm may break. With no explicit control on the learning process (remember

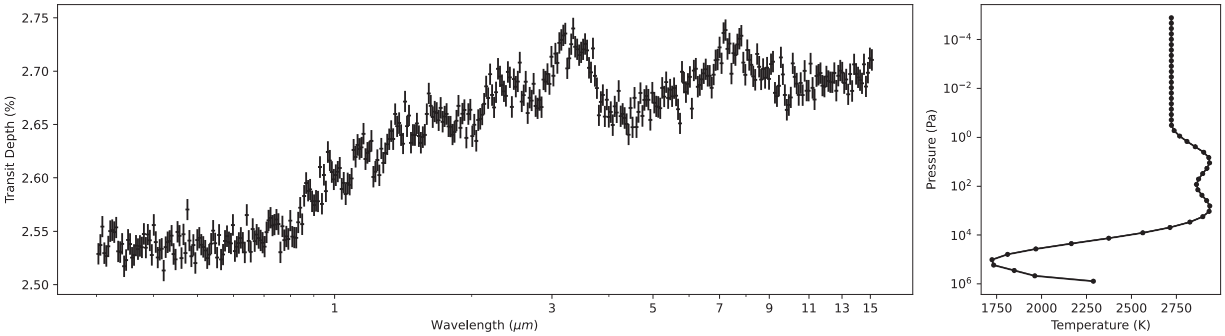

Figure 5. Left: simulated observation of WASP-43 b from 0.5 to 15 μm; Right: dayside temperature profile used to generated this observation (adopted from Changeat et al. 2021). There are 50 pressure points separated evenly (in logarithmic scale) between 10^{-5} and 10^{6}.

they are only asked to optimize the learning objective), it is uncertain how the model may interpret the training data. By relying only on the training data, it becomes important to make sure the training data can adequately represent the underlying forward model, which may be computational expensive or in some cases difficult to ascertain (Fisher & Heng 2022).

6.2. A Flexible and Interpretable Bayesian Framework

Motivated by the above limitations, we presented an alternative approach to train an end-to-end deep learning model for atmospheric retrieval. Instead of focusing on a generalizable model (one that works on a wide range of spectra), our framework is specific (or in other words, overfitted) to the observed data, and similar to conventional Markovian sampling algorithms, it is not generalizable and must be rerun for any changes in observed data and/or model assumptions. By dropping the goal of training a generalizable model, we forgo the need to create a vast training set as well as the need to pretrain our model before any retrieval could take place. It furthermore affords us with the flexibility of easily changing our input data and forward models. This flexibility has allowed us to explore the performance of our model at different spectral resolutions, wavelength ranges, observational uncertainties, and model assumptions with relative ease (as demonstrated above). However, this alternative approach also comes with disadvantages, see Section 6.4.1.

To allow direct comparison with conventional retrievals, we formulated the objective function to optimize the ELBO function, an alternative formulation of Bayes' theorem. This modification constrains the behavior of our trained model, and allows us to obtain results comparable to conventional retrievals (as demonstrated above). Another advantage of this approach is the drastic reduction of the amount of forward model computation to a fraction of its retrieval counterpart. Table 2 shows a comparison of the number of forward model calls between `NS-` and `VI-retrieval` in Cases I and II.

In the grid-based approaches, models have to learn implicitly the physical relationships between atmospheric parameters from the training data. There is no guarantee that the physical laws are preserved or correctly represented in the trained model. Our framework is purposely built to impose physical laws explicitly. One can view our NF-based network as a generator and Diff-τ as a corresponding decoder that transforms parameters into spectra. The static[8] decoder acts as a physical regularizer to the generator, which explicitly constrains the output from the generator to align with physical laws (any misalignment will be reflected in the transformed spectra). This is demonstrated through our examples, where the surrogate distributions are able to provide physically plausible correlations and are aligned with correlations produced from standard Bayesian sampling retrievals (`NS-retrieval`, blue). Having a fully analytic model allows us to impose physical laws directly, without going through a training data proxy.

6.3. Objective Function

The objective (loss) function is a crucial factor that governs the learning behavior of a deep learning model. Deep learning models in the literature are usually trained to best match the respective ground truth values of the physical parameters[9] (i.e., fitting for parameters). Here we opt to align our objective to that of our retrieval counterpart; in other words, we are explicitly asking our model to look for solutions that can best explain our observation (i.e., fitting for observations).

Adopting an observation-based likelihood not only allows us to align with the objective function of conventional retrievals, it also has the added advantage of properly accounting for observational uncertainties. ML-based retrieval methods often incorporate observational uncertainties through noise augmentation (Yip et al. 2020; Ardevol Martinez et al. 2022); this has resulted in overestimation of the error bound as compared to nested sampling–based retrievals (Ardevol Martinez et al. 2022).

In an ideal world, both approaches (fitting for parameters and fitting for observations) will agree with each other. However, it is not the case for inverse problems, where our observations are inherently corrupted,[10] and we may never be able to recover the ground truths in some cases due to a loss of information and the inverse processing being ill defined. In such cases, differences in the objective function may lead to different results. On one hand, the fitting-for-parameter approach is asking the neural network to pursue parameter values that may no longer be possible to retrieve (due to corruption of the observed data), which will cause the neural network to exhibit fictitious behavior if it results in the lowering of the loss function values (Yip et al. 2020). On the other hand, the fitting-for-observation approach explicitly asks for spectra that can explain the observation and not the underlying ground truth. Of course, that will also mean that our approach is not immune from the

[8] Static in the sense that it remains invariant throughout the training.

[9] Parameter values used to generate the forward model.

[10] Possible sources include instrument and astrophysical noise sources as well as information loss from binning data and/or by the forward model.

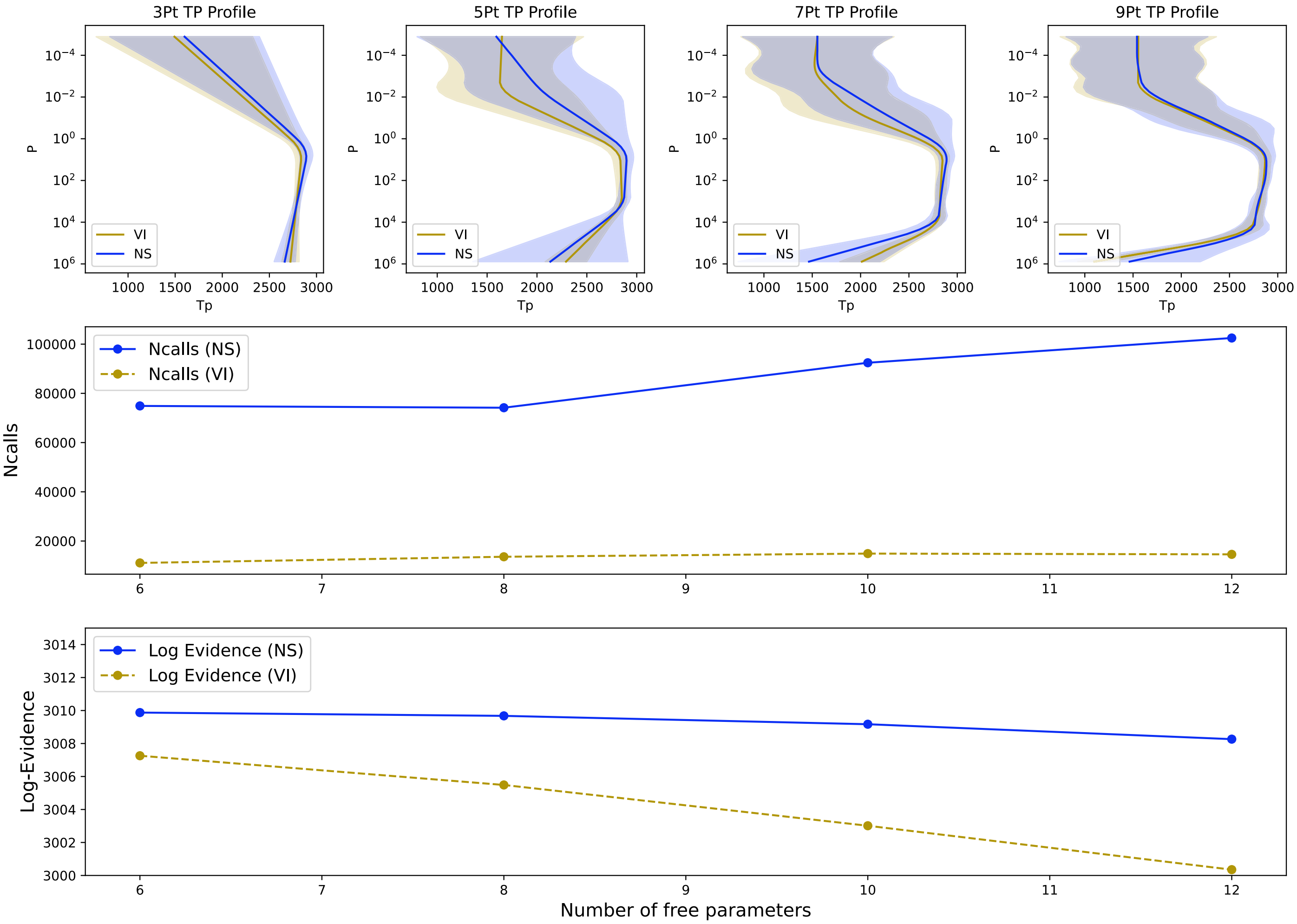

Figure 6. Comparing the performance of `VI-retrieval` (yellow) and `NS-retrieval`(blue). First row: *T–P* profiles retrieved by each approach. Each subplot shows a different fixed-point *T–P* profile, going from three points to nine points in steps of two. Second row: the number of forward model calls requires for each N-point retrieval. Third row: the log-evidence values retrieved by each approach.

Table 2
Comparing the Number of Forward Model Calls by Each Method for the Same Atmospheric Models (as in Case I) at WFC3/G141 and Ariel Spectral Resolution

Ncalls	`NS-retrieval`	`VI-retrieval`
WFC3/G141	105,622	8781
Ariel	65,414	9570

Note. In both cases, the number of forward model calls by `VI-retrieval` is significantly lower than `NS-retrieval`.

intrinsic retrieval biases induced by the atmospheric forward model itself (Rocchetto et al. 2016; Feng et al. 2020; MacDonald et al. 2020). These biases can only be alleviated through increasing the complexity of the atmospheric forward model to represent better the physical/chemical processes leading to the observed spectra.

In terms of model development, our framework bypasses the need to train the network with a large library of synthetic spectra before applying to actual data, as we are training the network directly on actual observations. This move avoids the problem of data shift (Quionero-Candela et al. 2009), where our training distribution is different from our test distribution.

6.3.1. *Computational Cost of* `VI-retrieval`

In this section we will discuss the computational cost associated with running a retrieval using `VI-retrieval` compared to `NS-retrieval`. Under the same forward model, `VI-retrieval` will always beat `NS-retrieval`, but the matter becomes slightly more complicated as the two methods do not always reuse the same forward model. We provide an initial assessment of the two methodologies with two different forward models. Note that the following figures are based on running both methods using a Macbook Pro with an Apple M1 chip (eight CPU cores in total). These figures should be taken with caution as the two forward models are at different developmental stages: TauREx3 is a highly optimized code while `Diff-`τ is proof-of-concept forward model written from scratch entirely in Tensorflow.

A single forward model call (`FMcall`) for Case I takes about 0.89 ± 0.02 s on `Diff-`τ and 0.26 ± 0.10 s with TauREx3's forward model. For a single retrieval, `VI-retrieval` takes 8781 `FMcall` and `NS-retrieval` takes 105,622 `FMcall`. Based on these numbers, `VI-retrieval` takes 7815 s and `NS-retrieval` takes 94,003 s. As for Case II, a single forward model call takes about 5.08 ± 0.38 s on `Diff-`τ and 0.51 ± 0.34 s with TauREx3's highly optimized forward model. For a single retrieval, `VI-retrieval` takes

Table 3
Retrieved ELBO from VI-retrieval and Bayesian (log-) Evidence from NS-retrieval (Ref) for Each Scenario

Model	ELBO	Ref	log $_{10}(\mathcal{B})$
Flat line	62.74	62.83	315.66
No methane	345.37	347.18	33.03
Complete	378.40	380.20	N/A
Overspecified Model	374.00	377.74	4.4

Note. $\log_{10}(\mathcal{B})$ shows the log-difference between the ELBO of our complete model and other competing models.

9570 FMcall and NS-retrieval takes 65,414 FMcall. Based on these numbers, VI-retrieval takes 48,615.2 s and NS-retrieval takes 33,361 s.

From our simple analysis above we can see that VI-retrieval's computation time is dependent on the speed of the forward model. It is empirically faster than NS-retrieval if the computational times of the forward models are similar to each other (Case I), but the advantage will elapse as the forward model takes longer to compute. This difference in performance can be easily narrowed down with model optimization and the maturity of the framework itself. In this vein, VI-retrieval will play an important role in retrieval scenarios with complex, and subsequently slow-to-run, forward models. In these situations the optimization objective becomes the number of forward model calls with VI-retrieval being an order of magnitude more efficient.

6.3.2. Model Selection

Model selection is a key part of the model evaluation cycle. So far, the ML-retrieval literature has largely ignored the issue by training networks under one or several fixed atmospheric assumptions. Our flexible framework and similar objective function to NS-retrieval means that we can, for the first time, utilize some of the tools frequently used by sampling-based retrievals to compare the retrieval results from different models.

Given that our surrogate distribution is a good approximation of the underlying posterior distribution, our ELBO, despite being a lower bound, should closely approximate the Bayesian evidence. We can therefore use the ELBO as a proxy of the evidence, and compare our models by estimating the Bayes' factor:

$$\mathcal{B}_{10} = \frac{P(\boldsymbol{x}|\mathcal{M}_1)P(\mathcal{M}_1)}{P(\boldsymbol{x}|\mathcal{M}_0)P(\mathcal{M}_0)}, \tag{13}$$

where $P(\boldsymbol{x}|\mathcal{M}_k)$ represents the Bayesian evidence attained from model $\mathcal{M}_k$and $P(\mathcal{M}_k)$ represents our prior belief on a particular model.

As an empirical example, we took our example from Case II and performed VI-retrieval and NS-retrieval with different atmospheric assumptions, including a flat line model, an incomplete model (without methane), a complete model (as specified in Case II), and an overspecified model (Case II plus TiO and VO). Table 3 compares the corresponding ELBO from VI-retrieval and the Bayesian evidence from NS-retrieval. The ELBO retrieved from each model closely follows, but is always smaller than, the corresponding Bayesian evidence, as set out from the definition of ELBO. The Bayes' factor displayed in Table 3 allows us to differentiate between the different models. Following Jeffrey's guideline scale (Jeffreys 1998; Hobson et al. 2002; Padilla et al. 2019), the Bayes' factor strongly favors our complete model over the other competing models. For more discussion on using Bayesian evidence as a model selection tool, please refer to the appendix in Changeat et al. (2021).

6.4. Limitations

6.4.1. Limitations on VI-retrieval

In this section we will focus on the framework of VI-retrieval.

1. *Generalizability*. As set out from the design goal, VI-retrieval is targeted to work on a single observation. This approach has earned us flexibility, where we can freely change our model assumptions, spectral range, and resolution with relative ease. However, the same advantage has also limited the generalizability of our framework. As opposed to other deep learning approaches, which can be rapidly deployed to a data set within its training set range, it will have to be retrained for each observation.
2. *Convergence*. Gradient descent helps to converge to a global optimum if the function is convex. However in the presence of model degeneracy, the function becomes nonconvex and convergence to global minima is not guaranteed.
3. *Flow bijections*. The expressiveness of the surrogate distribution depends highly on the architecture of neural network and the type of flow bijectors (IAF in our case). While our investigation has demonstrated the flexibility if these bijectors, it does not mean that any distribution (Durkan et al. 2019) can be mimicked. It remains an ongoing effort to design bijectors that are both flexible and computational efficient.
4. *Approximation*. The formulation of VI, especially the use of ELBO in the objective function, means that the retrieved surrogate distribution will always be an approximation to the ground truth. The use of flow-based neural networks has of course improved the fidelity of the approximation.
5. *Tendency to produce an underdispersed solution*. The implementation of ELBO and its modified form in Section 3.2 hints at the tendency for the network to produce an underdispersed solution. The negative log-likelihood term (first term) is almost always going to be larger than the prior term (second term), meaning that the network will tend to produce a sharp distribution first before widening itself to comply with the prior. This effect is also related to the expressiveness of the bijections, as their flexibility is ultimately bounded by the range of "action" that they can perform to transform the distribution. We note here that this effect is minor in our case as the shapes of the target distributions are relatively simple.
6. *Hyperparameter tuning*. Similar to conventional neural networks, hyperparameters do not always stay optimal from one setup to another, which means some degree of hyperparameter search should be performed to identify a good setup However, this is alleviated by identifying the key hyperparameters that may influence the optimization procedure. In our case we realized the annealing weights

plays a role in the optimization procedure, owing to their close relationship with the objective function. Other hyperparameters are generally fixed and do not differ much.

6.4.2. Limitations on Differentiable Frameworks

The ability to differentiate and provide gradients with respect to some quantities is central to modern deep learning algorithms. The growing popularity of ML in recent years has accelerated the development of differentiable frameworks, among them `Tensorflow`, `PyTorch`, and `JAX` have amassed substantial user bases.

There have been several recent attempts within the field of exoplanets to develop differentiable physical models within these frameworks. Initial results show that these differentiable models may hold the keys to overcome the curse of dimensionality brought by our increasingly complex models (Morvan et al. 2021; Kawahara et al. 2022), and may one day enable us to perform population studies without placing significant demand on computational resources.

However, these frameworks are not without issues. Here we would like to provide our "user experience" for readers who are interested and/or would like to implement their own models.

1. *Significant overheads*. Current development in differentiable programming mandates that any implementation must be written entirely in terms of a chosen framework. While it has become relatively straightforward to translate most operations from one framework to another, it is not a trivial and error-free process. These frameworks come with their own programming restrictions and conventions. Users are expected to adhere to these conventions or otherwise they might risk losing the ability to differentiate.
2. *Differentiability is not guaranteed*. Not all operations are differentiable. Some operations may be mathematically nondifferentiable[11] or they are not designed to have a gradient, such as a `LookUpTable` or `interpolation`. Depending on the algorithmic structure of the forward (atmospheric) model, this obstacle may present difficulties in obtaining a valid gradient. One good example is the gray cloud model. This model assumes the planet becomes completely opaque below a certain pressure level/altitude. This model, while easy to implement in any computational language, is not differentiable. A possible way around is for the user to implement their own gradient like in Kawahara et al. (2022), but this approach is only possible if one knows the differentiable form.
3. *Suboptimal performance*. Many modern differentiable languages are built and designed around deep learning–based applications. They are not designed to handle complex computational models. In other words, a differentiable model may suffer from reduced performance compared to its undifferentiable counterpart (Hu et al. 2019, 2020).
4. *Rapidly evolving language*. Differentiable languages are under constant development and rapid release cycles in response to the latest research. Some of these developments may not be backward compatible[12] and may impact the long term sustainability of the developed model.

7. Conclusion

In this paper we introduced the differentiable forward model (`Diff-`τ) based on `TauREx3` and implemented in `Tensorflow`. We combined our newly developed model with our density-alternating neural network and showed that it is possible to compute an approximate Bayesian posterior distribution that is in excellent agreement with the ones produced from computationally more expensive sampling-based techniques. Through our examples we have demonstrated three advantages of our framework:

1. *Fewer forward model calls*. Our `VI-retrieval` requires 25% or fewer forward model calls compared to `NS-retrieval` to converge, which opens up opportunities for more rapid retrieval with more complex (i.e., slower) forward models.
2. *Flexibility*. Unlike many deep learning–based frameworks, which rely on large precomputed libraries of spectra, our proposed framework resembles more closely a traditional retrieval set up by explicitly including the physical forward model in the deep learning architecture. Consequently, it retains many of the advantages of traditional retrieval codes, such as the freedom to choose the model assumptions, number of free parameters, spectral wavelength range, and observed uncertainties, without having to produce a separate training data set each time.
3. *Error propagation*. By incorporating observational uncertainties directly into the likelihood function, we demonstrated the capability of our network to produce uncertainty bounds that are on par with conventional atmospheric retrievals.
4. *Model selection*. We demonstrated, for the first time, that we can compare the adequacy of our neural network model to a given observation by computing the Bayesian evidence and Bayes factor.

Our proposed framework presents a major step toward the wider adoption of neural network–powered atmospheric retrieval. With significantly higher spectral resolutions and signal-to-noise ratios afforded by JWST and Ariel data, atmospheric forward models must consequently increase in complexity to model these data accurately. Such an increase in the dimensionality of the problem will result in significant strain on traditional sample-based retrieval approaches. While the results shine light to rapid AI-assisted Bayesian inference, we must stress that our framework is not meant to replace the sampling-based framework, but to complement existing frameworks. `VI-retrieval`, similar to most MCMC-based approaches, are susceptible to nonglobal minima. `NS-retrieval` on the other hand, tends to allow a more thorough exploration of the parameter space. This difference becomes important when multiple solutions are equally plausible for a given observation. The approximate retrieval performed by `VI-retrieval` can act as a precursor before the more computationally heavy nested sampling retrievals are required. Its rapid convergence means that one can scan through multiple candidate atmospheric models before converging to a few promising models for further, more detailed evaluations. The parameter bounds extracted by our framework can also act as an informative prior to speed up the sampling process of conventional retrievals.

[11] Such as a discontinuity in the function, e.g., logarithmic function.

[12] One good example is the switch from `Tensorflow` 1 to `Tensorflow` 2.

Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 758892, ExoAI) and the European Union's Horizon 2020 COMPET program (grant agreement No 776403, ExoplANETS A). Furthermore, we acknowledge funding by the Science and Technology Funding Council (STFC) grants: ST/K502406/1, ST/P000282/1, ST/P002153/1, and ST/S002634/1, and by the United Kingdom Space Agency (UKSA) grant No. ST/X002616-1.

Software: TauREx3 (Al-Refaie et al. 2021, 2022), Tensorflow 2 (Abadi et al. 2015), tensorflow-probability (Dillon et al. 2017), scipy (Virtanen et al. 2020), NumPy (Harris et al. 2020), Pandas (pandas development team 2020), nestle (Barbary 2021), corner (Foreman-Mackey 2016), matplotlib (Hunter 2007), and h5py (Collette 2013).

Appendix
ELBO Derivation

In this section we will describe the mathematical derivation that led to the formulation of the ELBO. We will start from Equation (3), i.e.:

$$q^*(z) = \underset{q(z)\in\mathbb{Q}}{\arg\min}\, D_{\mathrm{KL}}[p(z|\boldsymbol{x})||q(z)]. \tag{A1}$$

We start by expanding the left-hand side of the above expression:

$$D_{\mathrm{KL}}[p(z|\boldsymbol{x})||q(z)] = \mathbb{E}[\log q(z)] - \mathbb{E}[\log p(z|\boldsymbol{x})], \tag{A2}$$

and expand the latter term with the multiplication rule and chain rule of probability:

$$D_{\mathrm{KL}}[p(z|\boldsymbol{x})||q(z)] = \mathbb{E}[\log(q(z))] \\ -\mathbb{E}[\log p(z, \boldsymbol{x})] + \log p(\boldsymbol{x}), \tag{A3}$$

$$= \mathbb{E}[\log(q(z))] - \mathbb{E}[\log p(\boldsymbol{x}|z)] + \mathbb{E}[\log p(z)] + \log p(\boldsymbol{x}), \tag{A4}$$

$$= -(\mathbb{E}[\log p(\boldsymbol{x}|z)] - (\mathbb{E}[\log(q(z))] \\ -\mathbb{E}[\log p(z)]) + \log p(\boldsymbol{x}), \tag{A5}$$

$$= -(\mathbb{E}[\log p(\boldsymbol{x}|z)] - D_{\mathrm{KL}}[q(z)||p(z)]) + \log p(\boldsymbol{x}). \tag{A6}$$

Here we will substitute the expression for the ELBO as defined in Equation (4):

$$D_{\mathrm{KL}}[p(z|\boldsymbol{x})||q(z)] = -\mathrm{ELBO} + \log p(\boldsymbol{x}). \tag{A7}$$

From the above expression we can see that the K-L divergence is determined by the interaction between the ELBO and the Bayesian evidence term. While the former term depends on the quality of the surrogate distribution $q(z)$, the latter term is a constant as it depends only on the data $\boldsymbol{x}$. This, combined with the fact that $D_{\mathrm{KL}} \geqslant 0$, the ELBO term is inherently constrained by $\log p(\boldsymbol{x})$, i.e.:

$$\mathrm{ELBO} \leqslant \log p(\boldsymbol{x}). \tag{A8}$$

Hence the term is known as the ELBO. We can therefore use the ELBO as our objective function, since maximizing the ELBO is equivalent to minimizing the K-L divergence. In our implementation, the objective function is set to minimize the negative ELBO.

ORCID iDs

Kai Hou Yip https://orcid.org/0000-0002-9616-1524
Quentin Changeat https://orcid.org/0000-0001-6516-4493
Ahmed Al-Refaie https://orcid.org/0000-0003-2241-5330
Ingo P. Waldmann https://orcid.org/0000-0002-4205-5267

References

Aggarwal, V., Cotescu, M., Prateek, N., Lorenzo-Trueba, J., & Barra-Chicote, R. 2020, in IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2020) (Piscataway, NY: IEEE), 6179
Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/
Agol, E., Dorn, C., Grimm, S. L., et al. 2021, PSJ, 2, 1
Alam, M. K., López-Morales, M., MacDonald, R. J., et al. 2021, ApJL, 906, L10
Alam, M. K., López-Morales, M., Nikolov, N., et al. 2020, AJ, 160, 51
Al-Refaie, A. F., Changeat, Q., Venot, O., Waldmann, I. P., & Tinetti, G. 2022, ApJ, 932, 123
Al-Refaie, A. F., Changeat, Q., Waldmann, I. P., & Tinetti, G. 2021, ApJ, 917, 37
Amini Niaki, S., Haghighat, E., Campbell, T., Poursartip, A., & Vaziri, R. 2021, CMAME, 384, 113959
Anisman, L. O., Edwards, B., Changeat, Q., et al. 2020, AJ, 160, 233
Ardevol Martinez, F., Min, M., Kamp, I., & Palmer, P. I. 2022, arXiv:2203.01236
Argelaguet, R., Arnol, D., Bredikhin, D., et al. 2020, Genome Biol., 21, 111
Barbary, K. 2021, nestle: Nested sampling algorithms for evaluating Bayesian evidence, Astrophysics Source Code Library, ascl:2103.022
Barstow, J. K., Aigrain, S., Irwin, P. G. J., & Sing, D. K. 2017, ApJ, 834, 50
Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. 2018, J. Mach. Learn. Res., 18, 1
Berg, R. v. d., Hasenclever, L., Tomczak, J. M., & Welling, M. 2018, arXiv:1803.05649
Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. 2016, arXiv:1601.00670
Bogachev, V. I., Kolesnikov, A. V., & Medvedev, K. V. 2005, SbMat, 196, 309
Boucher, A., Darveau-Bernier, A., Pelletier, S., et al. 2021, AJ, 162, 233
Bradbury, J., Frostig, R., Hawkins, P., et al. 2018, JAX: composable transformations of Python+NumPy programs, 0.2.5, http://github.com/google/jax
Brogi, M., & Line, M. R. 2019, AJ, 157, 114
Buchholz, A., Wenzel, F., & Mandt, S. 2018, in Proc. 35th Int. Conf. on Machine Learning, Vol. 80, ed. J. Dy & A. Krause (PMLR), 668, https://proceedings.mlr.press/v80/buchholz18a.html
Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. 2021, AcMSn, 37, 1727
Carone, L., Mollière, P., Zhou, Y., et al. 2021, A&A, 646, A168
Challener, R. C., & Rauscher, E. 2022, AJ, 163, 117
Changeat, Q., & Al-Refaie, A. 2020, ApJ, 898, 155
Changeat, Q., Al-Refaie, A., Mugnai, L. V., et al. 2020, AJ, 160, 80
Changeat, Q., Al-Refaie, A. F., Edwards, B., Waldmann, I. P., & Tinetti, G. 2021, ApJ, 913, 73
Changeat, Q., & Edwards, B. 2021, ApJL, 907, L22
Changeat, Q., Edwards, B., Al-Refaie, A. F., et al. 2022, ApJS, 260, 3
Changeat, Q., Edwards, B., Waldmann, I. P., & Tinetti, G. 2019, ApJ, 886, 39
Changeat, Q., & Yip, K. H. 2023, RASTI, 2, 45
Chen, Y., Lu, L., Karniadakis, G. E., & Dal Negro, L. 2020, OExpr, 28 11618
Chérief-Abdellatif, B.-E., & Alquier, P. 2018, Electron. J. Statist., 12, 2995
Chubb, K. L., Min, M., Kawashima, Y., Helling, C., & Waldmann, I. 2020, A&A, 639, A3
Cobb, A. D., Himes, M. D., Soboczenski, F., et al. 2019, AJ, 158, 33
Collette, A. 2013, Python and HDF5 (O'Reilly), https://www.oreilly.com/library/view/python-and-hdf5/9781491944981
Cubillos, P. E., & Blecic, J. 2021, MNRAS, 505, 2675
Cuomo, S., Schiano di Cola, V., Giampaolo, F., et al. 2022, arXiv:2201.05624
Dillon, J. V., Langmore, I., Tran, D., et al. 2017, arXiv:1711.10604
Dinh, L., Krueger, D., & Bengio, Y. 2014, arXiv:1410.8516
Dinh, L., Sohl-Dickstein, J., & Bengio, S. 2016, arXiv:1605.08803
Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. 1987, PhLB, 195, 216

Durkan, C., Bekasov, A., Murray, I., & Papamakarios, G. 2019, Advances in Neural Information Processing Systems, 32 (Red Hook, NY: Curran Associates, Inc.)
Edwards, B., Changeat, Q., Mori, M., et al. 2021, AJ, 161, 44
Edwards, B., Changeat, Q., Tsiaras, A., et al. 2023, ApJS, 269, 31
Edwards, B., Rice, M., Zingales, T., et al. 2019, ExA, 47, 29
Evans, T. M., Sing, D. K., Wakeford, H. R., et al. 2016, ApJL, 822, L4
Fellows, M., Mahajan, A., Rudner, T. G. J., & Whiteson, S. 2018, arXiv:1811.01132
Feng, S., Xu, K., Wu, J., et al. 2022, arXiv:2205.07493
Feng, Y. K., Line, M. R., & Fortney, J. J. 2020, AJ, 160, 137
Fisher, C., & Heng, K. 2022, ApJ, 934, 31
Foote, T. O., Lewis, N. K., Kilpatrick, B. M., et al. 2022, AJ, 163, 7
Foreman-Mackey, D. 2016, JOSS, 1, 24
Foreman-Mackey, D., Luger, R., Agol, E., et al. 2021, JOSS, 6, 3285
Fortuin, V., Baranchuk, D., Raetsch, G., & Mandt, S. 2020, in Proc. 23rd Int. Conf. on Artificial Intelligence and Statistics, Vol. 108, ed. S. Chiappa & R. Calandra (PMLR), 1651, https://proceedings.mlr.press/v108/fortuin20a.html
Friston, Karl J., Parr, Thomas, Zeidman, Peter, et al. 2020, Wellcome Open Res., 5, 89
Gandhi, S., Madhusudhan, N., Hawker, G., & Piette, A. 2019, AJ, 158, 228
Gebhard, T. D., Bonse, M. J., Quanz, S. P., & Schölkopf, B. 2020, arXiv:2010.05591
Gebhard, T. D., Bonse, M. J., Quanz, S. P., & Schölkopf, B. 2022, A&A, 666, A9
Germain, M., Gregor, K., Murray, I., & Larochelle, H. 2015, in Proc. 32nd Int. Conf. on Machine Learning (PMLR), 881, https://proceedings.mlr.press/v37/germain15.html
Gibson, N. P., Merritt, S., Nugroho, S. K., et al. 2020, MNRAS, 493, 2215
Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., & Duvenaud, D. 2018, arXiv:1810.01367
Greene, T. P., Line, M. R., Montero, C., et al. 2016, ApJ, 817, 17
Gudovskiy, D., Ishizaka, S., & Kozuka, K. 2022, in Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (Piscataway, NJ: IEEE), 98
Haghighat, E., Raissi, M., Moure, A., Gomez, H., & Juanes, R. 2021, CMAME, 379, 113741
Haldemann, J., Ksoll, V., Walter, D., et al. 2023, A&A, 672, A180
Harrington, J., Himes, M. D., Cubillos, P. E., et al. 2022, PSJ, 3, 80
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357
Hayes, J. J. C., Kerins, E., Awiphan, S., et al. 2020, MNRAS, 494, 4492
Haynes, K., Mandell, A. M., Madhusudhan, N., Deming, D., & Knutson, H. 2015, ApJ, 806, 146
He, K., Zhang, X., Ren, S., & Sun, J. 2015, arXiv:1502.01852
Hellier, C., Anderson, D. R., Collier Cameron, A., et al. 2011, A&A, 535, L7
Himes, M. D., Harrington, J., Cobb, A. D., et al. 2022, PSJ, 3, 91
Hobson, M. P., Bridle, S. L., & Lahav, O. 2002, MNRAS, 335, 377
Hoffman, M. D., & Gelman, A. 2011, arXiv:1111.4246
Hu, Y., Anderson, L., Li, T.-M., et al. 2020, in 8th Int. Conf. on Learning Representations (ICLR 2020)
Hu, Y., Li, T.-M., Anderson, L., Ragan-Kelley, J., & Durand, F. 2019, ACM Transactions on Graphics (TOG), 38, 201
Hunter, J. D. 2007, CSE, 9, 90
Ih, J., & Kempton, E. M. R. 2021, AJ, 162, 237
Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
Irwin, P. G. J., Parmentier, V., Taylor, J., et al. 2020, MNRAS, 493, 106
Irwin, P. G. J., Teanby, N. A., de Kok, R., et al. 2008, J. Quant. Spec. Radiat. Transf., 109, 1136
Izmailov, P., Kirichenko, P., Finzi, M., & Wilson, A. G. 2020, in Proc. 37th Int. Conf. on Machine Learning, Vol. 119, ed. H. Daumé, III & A. Singh (PMLR), 4615, https://proceedings.mlr.press/v119/izmailov20a.html
Jeffreys, H. 1998, The Theory of Probability, Oxford Classic Texts in the Physical Sciences (Oxford: Oxford Univ. Press)
Jin, L., Doshi-Velez, F., Miller, T., Schwartz, L., & Schuler, W. 2019, in Proc. 57th Annual Meeting of the Association for Computational Linguistics (Stroudsburg, PA: ACL), 2442
Karchev, K., Coogan, A., & Weniger, C. 2022, MNRAS, 512, 661
Kawahara, H., Kawashima, Y., Masuda, K., et al. 2022, ApJS, 258, 31
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kingma, D. P., & Dhariwal, P. 2018, arXiv:1807.03039
Kingma, D. P., Salimans, T., Jozefowicz, R., et al. 2016, arXiv:1606.04934
Kingma, D. P., & Welling, M. 2013, arXiv:1312.6114
Kobyzev, I., Prince, S. J., & Brubaker, M. A. 2021, ITPAM, 43, 3964
Kreidberg, L., Bean, J. L., Désert, J.-M., et al. 2014, ApJL, 793, L27
Kreidberg, L., Line, M. R., Parmentier, V., et al. 2018, AJ, 156, 17
Krick, J. E., Fraine, J., Ingalls, J., & Deger, S. 2020, AJ, 160, 99
Kullback, S., & Leibler, R. A. 1951, Ann. Math. Stat., 22, 79
Lavie, B., Mendonça, J. M., Mordasini, C., et al. 2017, AJ, 154, 91
Lee, J.-M., Heng, K., & Irwin, P. G. J. 2013, ApJ, 778, 97
Lee, J.-M., Irwin, P. G. J., Fletcher, L. N., Heng, K., & Barstow, J. K. 2014, ApJ, 789, 14
Li, G., Gordon, I. E., Rothman, L. S., et al. 2015, ApJS, 216, 15
Line, M. R., Knutson, H., Wolf, A. S., & Yung, Y. L. 2014, ApJ, 783, 70
Line, M. R., Stevenson, K. B., Bean, J., et al. 2016, AJ, 152, 203
Line, M. R., Wolf, A. S., Zhang, X., et al. 2013, ApJ, 775, 137
Lopez-Alvis, J., Laloy, E., Nguyen, F., & Hermans, T. 2021, CG, 152, 104762
Lopez-Alvis, J., Nguyen, F., Looms, M. C., & Hermans, T. 2022, JGRB, 127, e22581
Lothringer, J. D., & Barman, T. S. 2020, AJ, 159, 289
Louizos, C., & Welling, M. 2017, in Proc. 34th Int. Conf. on Machine Learning, Vol. 70, (PMLR), 2218, https://proceedings.mlr.press/v70/louizos17a.html
Lugmayr, A., Danelljan, M., Gool, L. V., & Timofte, R. 2020, European Conference on Computer Vision (Berlin: Springer), 715
MacDonald, R. J., Goyal, J. M., & Lewis, N. K. 2020, ApJL, 893, L43
MacDonald, R. J., & Lewis, N. K. 2022, ApJ, 929, 20
MacDonald, R. J., & Madhusudhan, N. 2017, MNRAS, 469, 1979
MacDonald, R. J., & Madhusudhan, N. 2019, MNRAS, 486, 1292
Madhusudhan, N. 2018, in Atmospheric Retrieval of Exoplanets, ed. H. J. Deeg & J. A. Belmonte (Berlin: Springer), 104
Madhusudhan, N., & Seager, S. 2009, ApJ, 707, 24
Mansfield, M., Line, M. R., Bean, J. L., et al. 2021, NatAs, 5, 1224
Mansfield, M., Wiser, L., Stevenson, K. B., et al. 2022, AJ, 163, 261
Márquez-Neila, P., Fisher, C., Sznitman, R., & Heng, K. 2018, NatAs, 2, 719
Medvedev, K. V. 2008, Theory Stoch. Process., 14, 95
Meech, A., Aigrain, S., Brogi, M., & Birkby, J. L. 2022, MNRAS, 512, 2604
Mikal-Evans, T., Sing, D. K., Barstow, J. K., et al. 2022, NatAs, 6, 471
Mikal-Evans, T., Sing, D. K., Goyal, J. M., et al. 2019, MNRAS, 488, 2222
Min, M., Ormel, C. W., Chubb, K., Helling, C., & Kawashima, Y. 2020, A&A, 642, A28
Mollière, P., Stolker, T., Lacour, S., et al. 2020, A&A, 640, A131
Morvan, M., Nikolaou, N., Tsiaras, A., & Waldmann, I. 2020, AJ, 159, 109
Morvan, M., Tsiaras, A., Nikolaou, N., & Waldmann, I. P. 2021, PASP, 133, 034505
Mugnai, L. V., Modirrousta-Galian, D., Edwards, B., et al. 2021, AJ, 161, 284
Mugnai, L. V., Pascale, E., Edwards, B., Papageorgiou, A., & Sarkar, S. 2020, ExA, 50, 303
Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., & Welling, M. 2020, Advances in Neural Information Processing Systems, 33 (Red Hook, NY: Curran Associates, Inc.), 12685
Nikolaou, N., Waldmann, I. P., Tsiaras, A., et al. 2020, arXiv:2010.15996
Oord, A., Li, Y., Babuschkin, I., et al. 2018, in Proc. 35th Int. Conf. on Machine Learning, Vol. 80, (PMLR), 3918, https://proceedings.mlr.press/v80/oord18a.html
Oreshenko, M., Kitzmann, D., Márquez-Neila, P., et al. 2020, AJ, 159, 6
Padilla, L. E., Tellez, L. O., Escamilla, L. A., & Vazquez, J. A. 2019, arXiv:1903.11127
pandas development team, T, 2020 pandas-dev/pandas: Pandas v2.2, Zenodo, 10.5281/zenodo.3509134
Papamakarios, G., Pavlakou, T., & Murray, I. 2017, Advances in Neural Information Processing Systems, 30 (Red Hook, NY: Curran Associates, Inc.)
Paszke, A., Gross, S., Massa, F., et al. 2019, in Advances in Neural Information Processing Systems, ed. H. Wallach et al., 32 (Red Hook, NY: Curran Associates, Inc.), 8024
Pati, D., Bhattacharya, A., & Yang, Y. 2018, in Proc. 21st Int. Conf. on Artificial Intelligence and Statistics, Vol. 84, ed. A. Storkey & F. Perez-Cruz (PMLR), 1579, https://proceedings.mlr.press/v84/pati18a.html
Pinhas, A., Madhusudhan, N., Gandhi, S., & MacDonald, R. 2019, MNRAS, 482, 1485
Pluriel, W., Whiteford, N., Edwards, B., et al. 2020a, AJ, 160, 112
Pluriel, W., Zingales, T., Leconte, J., & Parmentier, V. 2020b, A&A, 636, A66
Polyansky, O. L., Kyuberis, A. A., Zobov, N. F., et al. 2018, MNRAS, 480, 2597
Prenger, R., Valle, R., & Catanzaro, B. 2019, in IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2019) (Piscataway, NJ: IEEE), 3617
Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. 2009, Dataset Shift in Machine Learning (Cambridge, MA: The MIT Press)
Raissi, M., Perdikaris, P., & Karniadakis, G. 2019, JCoPh, 378, 686
Rasmussen, K. C., Brogi, M., Rahman, F., et al. 2022, AJ, 164, 35

Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U., & Vollgraf, R. 2020, arXiv:2002.06103

Rezende, D. J., & Mohamed, S. 2015, arXiv:1505.05770

Rippel, O., & Adams, R. P. 2013, arXiv:1302.5125

Rocchetto, M., Waldmann, I. P., Venot, O., Lagage, P. O., & Tinetti, G. 2016, ApJ, 833, 120

Rothfuss, J., Ferreira, F., Boehm, S., et al. 2019, arXiv:1907.08982

Roudier, G. M., Swain, M. R., Gudipati, M. S., et al. 2021, AJ, 162, 37

Rudolph, M., Wandt, B., & Rosenhahn, B. 2021, in Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision (Piscataway, NJ: IEEE), 1907

Saba, A., Tsiaras, A., Morvan, M., et al. 2022, AJ, 164, 2

Schmidt, M., & Simic, M. 2019, arXiv:1906.06904

Seidel, J. V., Ehrenreich, D., Pino, L., et al. 2020, A&A, 633, A86

Shallue, C. J., & Vanderburg, A. 2018, AJ, 155, 94

Sheppard, K. B., Mandell, A. M., Tamburo, P., et al. 2017, ApJL, 850, L32

Sheppard, K. B., Welbanks, L., Mandell, A. M., et al. 2021, AJ, 161, 51

Shu, R., Bui, H. H., Zhao, S., Kochenderfer, M. J., & Ermon, S. 2018, Advances in Neural Information Processing Systems, 31 (Red Hook, NY: Curran Associates, Inc.)

Sing, D. K., Fortney, J. J., Nikolov, N., et al. 2016, Natur, 529, 59

Skaf, N., Bieger, M. F., Edwards, B., et al. 2020, AJ, 160, 109

Smith, L. N. 2015, arXiv:1506.01186

Stevenson, K. B., Line, M. R., Bean, J. L., et al. 2017, AJ, 153, 68

Sun, H., Bouman, K. L., Tiede, P., et al. 2022, ApJ, 932, 99

Swain, M. R., Estrela, R., Roudier, G. M., et al. 2021, AJ, 161, 213

Tinetti, G., Eccleston, P., Haswell, C., et al. 2021, arXiv:2104.04824

Tinetti, G., Vidal-Madjar, A., Liang, M.-C., et al. 2007, Natur, 448, 169

Tsiaras, A., Rocchetto, M., Waldmann, I. P., et al. 2016a, ApJ, 820, 99

Tsiaras, A., Waldmann, I. P., Rocchetto, M., et al. 2016b, ApJ, 832, 202

Tsiaras, A., Waldmann, I. P., Tinetti, G., Tennyson, J., & Yurchenko, S. N. 2019, NatAs, 3, 1086

Udry, S., Lovis, C., Bouchy, F., et al. 2014, arXiv:1412.1048

Valizadegan, H., Martinho, M., Wilkens, L. S., et al. 2021, ApJ, 926, 120

Viana, F. A., & Subramaniyan, A. K. 2021, Arch. Comput. Methods in Eng., 28, 3801

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261

von Essen, C., Mallonn, M., Hermansen, S., et al. 2020, A&A, 637, A76

Waldmann, I. P., & Griffith, C. A. 2019, NatAs, 3, 620

Waldmann, I. P., Rocchetto, M., Tinetti, G., et al. 2015, ApJ, 813, 13

Wu, H., Köhler, J., & Noé, F. 2020, Advances in Neural Information Processing Systems, 33 (Red Hook, NY: Curran Associates, Inc.), 5933

Yip, K. H., Changeat, Q., Edwards, B., et al. 2021, AJ, 161, 4

Yip, K. H., Changeat, Q., Nikolaou, N., et al. 2020, arXiv:2011.11284

Yip, K. H., Nikolaou, N., Coronica, P., et al. 2019, arXiv:1904.06155

Yip, K. H., Waldmann, I. P., Changeat, Q., et al. 2022, arXiv:2206.14642

Yu, L., Vanderburg, A., Huang, C., et al. 2019, AJ, 158, 25

Yurchenko, S. N., Amundsen, D. S., Tennyson, J., & Waldmann, I. P. 2017, A&A, 605, A95

Yurchenko, S. N., Barber, R. J., & Tennyson, J. 2011, MNRAS, 413, 1828

Yurchenko, S. N., Mellor, T. M., Freedman, R. S., & Tennyson, J. 2020, MNRAS, 496, 5282

Zhang, C., Butepage, J., Kjellstrom, H., & Mandt, S. 2019, ITPAM, 41, 2008

Zhang, M., Chachan, Y., Kempton, E. M. R., & Knutson, H. A. 2019, PASP, 131, 034501

Zhang, M., Chachan, Y., Kempton, E. M. R., Knutson, H. A., & Chang, W. H. 2020, ApJ, 899, 27

Zhang, Q., & Chen, Y. 2021, Advances in Neural Information Processing Systems, 34 (Red Hook, NY: Curran Associates, Inc.), 16280

Zhang, X., Nawaz, M. A., Zhao, X., & Curtis, A. 2021, AdGeo, 62, 73

Zingales, T., & Waldmann, I. P. 2018, AJ, 156, 268