Draft version May 28, 2019
Typeset using LaTeX **twocolumn** style in AASTeX62

An Ensemble of Bayesian Neural Networks for Exoplanetary Atmospheric Retrieval

Adam D. Cobb[1,*] and Michael D. Himes[2,*]

Frank Soboczenski,[3] Simone Zorzan,[4] Molly D. O'Beirne,[5] Atilim Güneş Baydin,[1] Yarin Gal,[6]
Shawn D. Domagal-Goldman,[7] Giada N. Arney,[7] and Daniel Angerhausen[8,9]
2018 NASA FDL Astrobiology Team II

[1]*Department of Engineering Science, University of Oxford*
[2]*Planetary Sciences Group, Department of Physics, University of Central Florida*
[3]*SPHES, King's College London*
[4]*ERIN Department, Luxembourg Institute of Science and Technology*
[5]*Department of Geology and Environmental Science, University of Pittsburgh*
[6]*Department of Computer Science, University of Oxford*
[7]*NASA Goddard Space Flight Center, Greenbelt, MD*
[8]*CSH Fellow, Center for Space and Habitability, University of Bern, Switzerland*
[9]*Blue Marble Space Institute of Science, Seattle, United States*

ABSTRACT

Machine learning is now used in many areas of astrophysics, from detecting exoplanets in Kepler transit signals to removing telescope systematics. Recent work demonstrated the potential of using machine learning algorithms for atmospheric retrieval by implementing a random forest to perform retrievals in seconds that are consistent with the traditional, computationally-expensive nested-sampling retrieval method. We expand upon their approach by presenting a new machine learning model, `plan-net`, based on an ensemble of Bayesian neural networks that yields more accurate inferences than the random forest for the same data set of synthetic transmission spectra. We demonstrate that an ensemble provides greater accuracy and more robust uncertainties than a single model. In addition to being the first to use Bayesian neural networks for atmospheric retrieval, we also introduce a new loss function for Bayesian neural networks that learns correlations between the model outputs. Importantly, we show that designing machine learning models to explicitly incorporate domain-specific knowledge both improves performance and provides additional insight by inferring the covariance of the retrieved atmospheric parameters. We apply `plan-net` to the Hubble Space Telescope Wide Field Camera 3 transmission spectrum for WASP-12b and retrieve an isothermal temperature and water abundance consistent with the literature. We highlight that our method is flexible and can be expanded to higher-resolution spectra and a larger number of atmospheric parameters.

Keywords: methods: statistical — techniques: retrieval — techniques: machine learning — methods: Bayesian neural network — planetary systems — WASP-12b

1. INTRODUCTION

Over a decade ago, light emitted from an exoplanet was first measured, paving the way for the study of exoplanetary atmospheres (Charbonneau et al. 2005; Deming et al. 2005). In the years since, a diverse collection of worlds have been discovered, from rocky, Earth-like planets to massive gas giants that reach temperatures as hot as some stars (Hasegawa & Pudritz 2013; Batalha 2014). Edge-on planetary systems enable the measurement of transit (when the exoplanet passes in between the host star and the observer) and eclipse (when the exoplanet passes behind the host star as viewed by the observer) depths (Kreidberg 2017). Transit depths measure the effective radius of the planet as a function of wavelength; variations in measured radius arise from the

Corresponding authors:
Adam D. Cobb (machine learning questions)
Michael D. Himes (exoplanetary questions)
acobb@robots.ox.ac.uk, mhimes@knights.ucf.edu

* These two authors contributed equally

molecules in the atmosphere at the day-night terminator absorbing certain wavelengths of light, with more absorption corresponding to larger measured radii. Eclipse depths measure the ratio of the planet's and host star's emission as a function of wavelength. These depths provide insight into the composition and temperature structure of the planet's atmosphere.

Using measured transit or eclipse depths spanning a range of wavelengths, an atmospheric model for the planet can be determined with some uncertainty via atmospheric retrieval, an inverse modeling technique (Madhusudhan 2018). Early retrieval studies performed a parametric grid search over millions of pre-calculated forward models (Madhusudhan & Seager 2009). This method was later improved by Bayesian techniques employing Markov chain Monte Carlo (MCMC) and other sampling techniques (e.g., Skilling 2004; ter Braak 2006; ter Braak & Vrugt 2008) to explore a model parameter space by computing spectra for thousands to millions of atmospheric models (e.g., Madhusudhan & Seager 2010; Line et al. 2014; Waldmann et al. 2015; Oreshenko et al. 2017). Model parameters describe the temperature–pressure profile, $T(p)$; the vertical abundance profiles for each molecule in the atmospheric model; cloud parameters; and, for the transit case, the radius of the planet. These Bayesian techniques yield a posterior distribution which constrains the range of values that fit the data for each model parameter. For low-resolution data, some parameters may be only constrained to an upper/lower limit (or not at all) due to degeneracies among low-resolution spectra (e.g., a slightly cooler atmosphere with greater abundances of molecules will look the same as a slightly warmer atmosphere with lesser abundances). While high-resolution data allows for parameters to be more accurately determined, there is still some inherent uncertainty due to astrophysical and instrumental noise. Accurate quantification of this uncertainty informs the statistical significance of the results.

Data-driven machine learning (ML) approaches, which are able to learn complex relationships within large data sets, provide possible solutions to methods that can be computationally-expensive, such as atmospheric retrieval. Examples using ML can be seen across the field of astrophysics from applying Bayesian linear regression to remove common-mode systematics in Kepler data (Roberts et al. 2013), to automating the process of identifying exoplanets using deep learning (Shallue & Vanderburg 2018; Ansdell et al. 2018; Osborn et al. 2019). Furthermore, in an approach similar to our own, but in a different application domain, Perreault Levasseur et al. (2017) used Bayesian neural networks to map distant gravitationally-lensed galaxies.

Recently, the study of exoplanetary atmospheres has been aided by ML techniques. Waldmann (2016) makes use of deep belief networks to classify exoplanet emission spectra, importantly showing that ML approaches can identify molecular signatures in emission spectra. The first supervised ML retrieval algorithms, `HELA` (Márquez-Neila et al. 2018) and `ExoGAN` (Zingales & Waldmann 2018), have been developed and show promising results. `HELA` uses a random forest to classify observed spectra into some planetary model (see Section 3.1 for more details), while `ExoGAN` combines a generative adversarial network (GAN, Goodfellow et al. 2014) with a technique called semantic image inpainting (Yeh et al. 2017) to retrieve atmospheric parameters. These methods reduce retrieval times from hundreds of central processing unit hours to just seconds/minutes, highlighting the large reductions in computation times offered by ML.

Here, we introduce a new ML retrieval method, `plan-net`[1], which is based on an ensemble of Bayesian neural networks, and apply it to the benchmark data set of Márquez-Neila et al. (2018). BNNs are a good choice of model for atmospheric retrievals as they give the advantage of both providing probability distributions over their outputs and scaling to high-dimensional data. We directly compare our model with `HELA` over the same data set and demonstrate how incorporating domain-specific knowledge into machine learning models can improve results and offer insights into the covariance of the atmospheric parameters.

In this paper we first introduce the data set in Section 2 along with the notation. We then introduce both ML models in Section 3, where we start with the random forest followed by a detailed explanation of our model. In Section 4 we both display and discuss our results. Finally, in Section 5 we make conclusions about the implications of our results and suggest further avenues for research in this area.

2. DATA SET

2.1. *Description*

We use the spectral data set of Márquez-Neila et al. (2018) which consists of 100,000 synthetic Hubble Space Telescope Wide Field Camera 3 (WFC3) transmission spectra of hot Jupiters. These spectra were created using the formalism detailed in Heng & Kitzmann (2017), which makes use of line-by-line calculations for opacities (S. Grimm, priv. comm.). This is based on five atmospheric parameters: an isothermal temperature; abun-

[1] Our code is available at https://github.com/exoml/plan-net.

dances of H_2O, NH_3, and HCN gas; and a gray cloud opacity, κ_0. Each spectrum has 13 channels with bandpasses matching those used in Kreidberg et al. (2015) ($0.838 - 1.666$ μm). Each channel holds the transit depth within the corresponding bandpass. We refer the reader to their papers for more details, particularly the 'Methods' section of Márquez-Neila et al. (2018), as this is where the boundary conditions are described.

For each transit depth, we assume the same 50 parts per million uncertainty as Márquez-Neila et al. (2018). We similarly split the data set between training $(80,000)$ and testing $(20,000)$. We reserve $10,000$ spectra from the training set to be the validation set, which is used to optimize model hyperparameters and architectures. This ensures that inferences are made on the test data only one time. We use the same real-data test case: the WASP-12b WFC3 transit depths as analyzed by Kreidberg et al. (2015). Two sample input spectra can be seen in the Appendix, Figures 3c and 4c.

2.2. *Notation*

In this paper we use the following notation to describe our data set, $\mathcal{D}$. A single spectrum with 13 channels is denoted by the vector $\mathbf{s} \in \mathcal{R}^{13}$ and $\boldsymbol{\theta} \in \mathcal{R}^5$ defines the vector of five atmospheric parameters. Furthermore, we generalize our model by referring to the dimension of $\boldsymbol{\theta}$ as D. The training and testing data sets are denoted by $\mathcal{D}_{\text{tr}}$ and $\mathcal{D}_{\text{te}}$ respectively, where the test data is given by $\mathcal{D}_{\text{te}} = \{\mathbf{s}_n, \boldsymbol{\theta}_n\}_{n=1}^{N}$ for N total input-output pairs.

3. MACHINE LEARNING MODELS

In machine learning, the task of inferring a function from labeled data comes under the area of supervised learning. In our case, the task is a multivariate regression problem, where the objective is to model the relationship between the input-space, $\mathbf{s}$, and the output-space, $\boldsymbol{\theta}$. In addition to predicting the values of the outputs, it is vital that the ML model also provides an uncertainty estimation over these values. Astronomical observations inherently introduce uncertainty in measurements, and accurately accounting for and reporting these uncertainties is a critical part of retrieval results.

In this section we introduce the previously-used random forest along with our `plan-net` model. In each section we explain how each model aims to solve this multivariate regression task and how they each deal with uncertainty. We highlight that the `plan-net` model is specifically designed to deal with both the uncertainty and the correlations between the outputs, whereas the the random forest does not differ from those used in other multivariate regression tasks.

3.1. *Random Forest*

Here, we briefly summarize the random forest regression model used in Márquez-Neila et al. (2018), where the details of the model are available at https://github.com/exoclime/HELA. The core of their model comes from the `ensemble.RandomForestRegressor` method in `sklearn` (Pedregosa et al. 2011). A random forest (RF) consists of multiple decision trees (or regression trees, for the case of continuous data), whereby each tree makes a prediction given an input (see Criminisi et al. (2012)). Márquez-Neila et al. (2018) showed that no more than $1,000$ regression trees were required, which led to choosing that number for the model. They set the number of nodes in each tree via a variance threshold of 0.01. This is a metric that is related to the proportion of the remaining training data that is split at the current node.

To produce the posterior plots, as shown in Figure 2a, each prediction from a tree corresponds to a sample from an empirical distribution. The 1000 samples therefore correspond to the density estimation of the atmospheric parameters.

3.2. *Bayesian Neural Networks*

Our model is built from Bayesian neural networks, which inherit their structure from neural networks. Although we provide details of both techniques in the following section, we highlight their strong relationship with multivariate linear regression, where the objective is to learn a matrix of weights $\mathbf{W}$ that map an input $\mathbf{s}$ to an output $\boldsymbol{\theta}$. Fully connected deep neural networks extend upon this by combining layers of linear regression with non-linear functions to result in a more powerful function-approximating capability, despite still operating on the same supervised learning task as a linear regression model.

3.2.1. *A Summary*

Bayesian neural networks (BNNs) offer the powerful function-approximating capability of deep neural networks with the additional advantage of being able to provide distributions over their outputs (MacKay 1992; Neal 1995). Therefore, these characteristics are well-suited to the task of atmospheric retrieval. To enable BNNs to scale to large architectures we employ the Monte Carlo dropout approximation to BNNs (Gal & Ghahramani 2016). This is a stochastic variational inference approach (Hoffman et al. 2013) that allows BNN inference to be performed for both large architectures and large data sets. The alternative approach would be to implement a form of MCMC such as Hamiltonian Monte Carlo (HMC, Neal 1995) to perform inference.

Although HMC has been shown to be successful at small scale, it currently cannot be scaled in the same way as stochastic variational inference approaches.

Deep neural networks consist of a hierarchy of layers, where each layer applies a non-linear weighted transformation of its input. We define each layer l, to have its own matrix of weights $\mathbf{W}_l$ and biases $\mathbf{b}_l$. If $\mathbf{h}(\cdot)$ is a non-linear function then we can define a fully connected dense neural network with L layers and input $\mathbf{s}$ as:

$$\mathbf{f}^{\boldsymbol{\omega}}(\mathbf{s}) = \mathbf{W}_L \mathbf{h}\left(\dots \mathbf{h}(\mathbf{W}_0 \mathbf{s} + \mathbf{b}_l) \dots\right) + \mathbf{b}_L,$$

where $\boldsymbol{\omega} = \{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{L}$ and refers to all the network weights. A BNN takes this formulation and adds a prior $p(\boldsymbol{\omega})$ over the weights, often taking the form of a multivariate normal distribution. Bayesian inference in BNNs requires computing an intractable integral to infer $p(\boldsymbol{\omega} \mid \mathcal{D}_{\rm tr})$. The Monte Carlo dropout approximation provides a (variational) approximation to this distribution and comes under the wider area of variational inference (Jordan et al. 1998). Practical implementation of MC dropout requires drawing dropout masks (Srivastava et al. 2014) from Bernoulli-distributed random variables to set a certain proportion of weights to zero. Applying this during the training of the network acts as a regularizer to prevent overfitting. Dropping these weights whilst making predictions at test-time results in the test-time approximation for predictions over the outputs. For a given input $\mathbf{s}_n$, we can sample the network T times to result in an empirical distribution $p(\boldsymbol{\theta} \mid \mathbf{s}_n, \mathcal{D}_{\rm tr})$.

Determining the proportion of weights to be dropped in each layer p_l often requires tuning over a validation set. However, we use concrete dropout layers to automatically optimize for these values in the training process (Gal et al. 2017).

3.2.2. *The Model*

Our model, `plan-net`, shown in Figure 1, is a deep neural network with four dense concrete dropout layers (Gal et al. 2017). The model is implemented in `Keras` (Chollet et al. 2015) with a `TensorFlow` backend (Abadi et al. 2016). Each layer consists of 1024 units, and we use a batch size of 512. For training the model, we use the Adam optimization algorithm (Kingma & Ba 2014). For deciding on the architecture, we implemented a grid search over the number of layers and the number of units per layer.

Our task is to accurately predict the atmospheric parameters and provide posterior[2] distributions over their values. These parameters are expected to covary and we directly use this domain knowledge to design our model, such that we can represent the atmospheric parameters to be jointly distributed by a multivariate normal distribution. Therefore we design the output of the BNN to consist of the a lower triangular matrix $\mathbf{L}$ of dimensions $D \times D$ and a mean vector $\boldsymbol{\mu}$ of dimension D. We can then represent the precision matrix of a multivariate normal via its Cholesky decomposition $\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^\top$.

Figure 1 demonstrates the atmospheric retrieval process after the model is trained. We implement T forward passes through the network for a given observed spectrum $\mathbf{s}_n$, resulting in the samples $\{\boldsymbol{\mu}(\mathbf{s}_n)_t, \mathbf{L}(\mathbf{s}_n)_t\}_{t=1}^{T}$. In the next step, we take the mean over these network samples to give the expected $\mathbf{L}(\mathbf{s}_n)$ and $\boldsymbol{\mu}(\mathbf{s}_n)$ for a given spectrum:

$$\mathbf{L}(\mathbf{s}_n) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{L}(\mathbf{s}_n)_t, \tag{1}$$

$$\boldsymbol{\mu}(\mathbf{s}_n) = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\mu}(\mathbf{s}_n)_t. \tag{2}$$

The final step is to sample from the multivariate normal distribution,

$$\boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{s}_n), (\mathbf{L}(\mathbf{s}_n)\mathbf{L}^\top(\mathbf{s}_n))^{-1}\right) \tag{3}$$

to retrieve samples from the inferred atmospheric parameters, where this distribution is parameterized by the expectation BNN output.

3.2.3. *Training*

In order to train this model, we must design a loss that ensures the network learns the correlations between the atmospheric parameters. In order to estimate the covariance, our loss is the negative log-likelihood of the multivariate normal, as defined by $\boldsymbol{\mu}$ and $\mathbf{L}$. The loss,

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\mu}, \mathbf{L}) = -2\sum_{d=1}^{D} \log(l_{dd}) + (\mathbf{y}-\boldsymbol{\mu})^\top \mathbf{L}\mathbf{L}^\top (\mathbf{y}-\boldsymbol{\mu}), \tag{4}$$

is defined to be implicitly dependent on the network weights $\boldsymbol{\omega}$ through the lower triangular matrix $\mathbf{L}$ and the inferred mean $\boldsymbol{\mu}$ (see Figure 1). As also mentioned in Dorta et al. (2018), we must be careful to ensure that the diagonal elements, l_{ii}, of $\mathbf{L}$ are positive such that $\boldsymbol{\Lambda}$ is positive-definite; we therefore take the exponential of the diagonal terms to ensure this. In comparison to previous loss functions that have been used for BNNs, such

[2] In the machine learning literature, the output distribution would normally be called the predictive distribution as we are inferring the posterior over the weights of the network and then working with this posterior to infer a predictive distribution. However, to remain consistent with the exoplanet literature, we avoid that here.

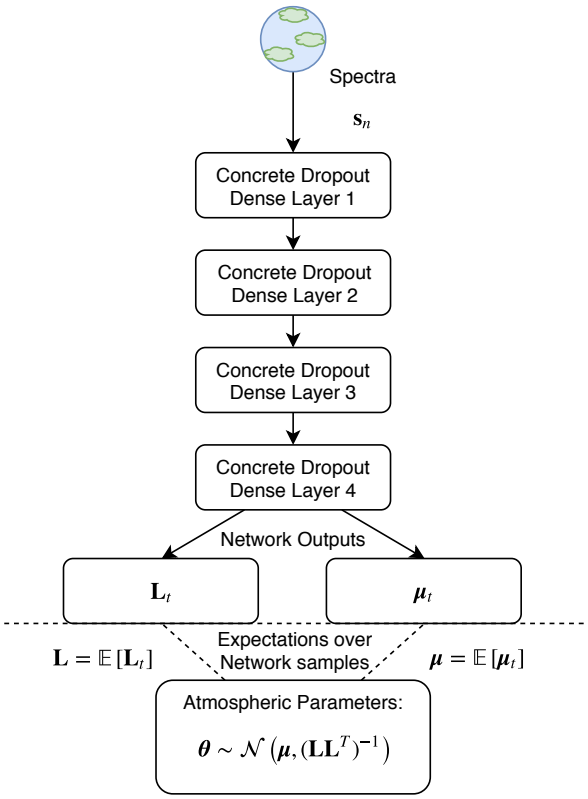


Figure 1. `plan-net` model procedure at test time for a given spectrum $\mathbf{S}_n$. T samples are taken from the BNN and the expectations over the lower triangular matrix and the mean are then used to parameterize the multivariate normal distribution. $\boldsymbol{\theta}$ can then be drawn from this distribution to retrieve the atmospheric parameters. Each concrete dropout layer consists of 1024 units.

as the squared loss and the heteroscedastic squared loss (see Gal (2016, Chapter 4)), our new loss in Equation (4) is able to model correlations between atmospheric parameters. These inferred correlations lead to better uncertainty estimates for the retrieved atmospheric parameters than the previous losses.

In addition to using the Adam optimizer, we employ early stopping, with a patience of 30 epochs, according to the validation loss. Furthermore, we use model checkpointing to save the model that has the best performance on the validation set.

3.3. *Ensemble*

It has been shown that an ensemble of neural networks can offer more accurate estimations of the predictive uncertainty than a single network (Lakshminarayanan et al. 2017; Gal & Smith 2018). The additional benefit is that an ensemble is more robust to changes in weight initialization and the path taken during stochastic optimization.

In this paper we use an ensemble of five `plan-net` models and provide comparison to a single model. Five models were chosen due to the empirical performance in Table 1, as larger ensembles result in increasingly marginal improvements.

The challenge in using an ensemble is in how the outputs from the individual models are combined. In our case, each output is the mean and covariance of a multivariate normal distribution. Therefore in combining these distributions together, we can treat the overall output from the ensemble as a Gaussian mixture model, whereby the each component weight corresponds to $^1/_M$, where M is the number of models in the ensemble.

To calculate the expectation of this mixture model, $\boldsymbol{\mu}_{\text{ens}}$, we take the average of the individual component means such that

$$\boldsymbol{\mu}_{\text{ens}} = \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{\mu}_m.$$

The variance of the mixture model $\boldsymbol{\Sigma}_{\text{ens}}$ can be calculated by employing the law of total variance:

$$\boldsymbol{\Sigma}_{\text{ens}} = \frac{1}{M}\sum_{m=1}^{M}(\boldsymbol{\mu}_{\text{m}} - \boldsymbol{\mu}_{\text{ens}})^2 + \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{\Sigma}_m,$$

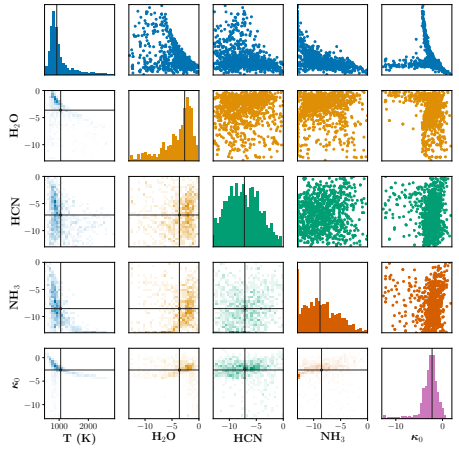
where the inferred covariance matrix of a single model is given by $\boldsymbol{\Sigma}_m = \boldsymbol{\Lambda}_m^{-1} = (\mathbf{L}_m\mathbf{L}_m^\top)^{-1}$. This combines the variance in the component means with the expectation of the variance of the individual models, thus taking into account how unsure each model is and how far each model's mean lies from the ensemble mean. Therefore the atmospheric parameters retrieved via the ensemble $\boldsymbol{\theta}_{\text{ens}}$ are distributed according to $\boldsymbol{\theta}_{\text{ens}} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{ens}}, \boldsymbol{\Sigma}_{\text{ens}})$.

4. RESULTS AND DISCUSSION

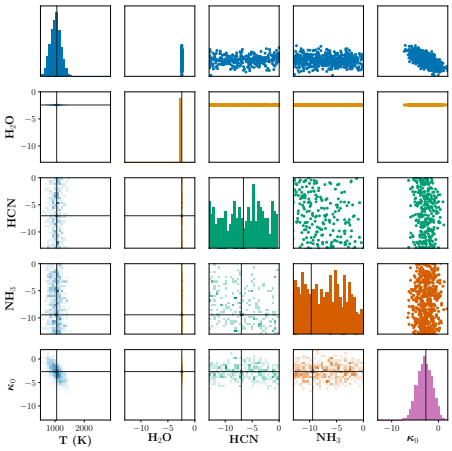
Table 1 displays a comparison of R^2 values across the models, where R^2 corresponds to the coefficient of determination

$$R^2 = 1 - \frac{\sum_{n=1}^{N}\sum_{d=1}^{D}\left(\theta_n^{(d)} - \mu_{\text{ens}}^{(d)}(\mathbf{s}_n)\right)}{\sum_{n=1}^{N}\sum_{d=1}^{D}(\theta_n^{(d)} - \tilde{\theta}^{(d)})} \tag{5}$$

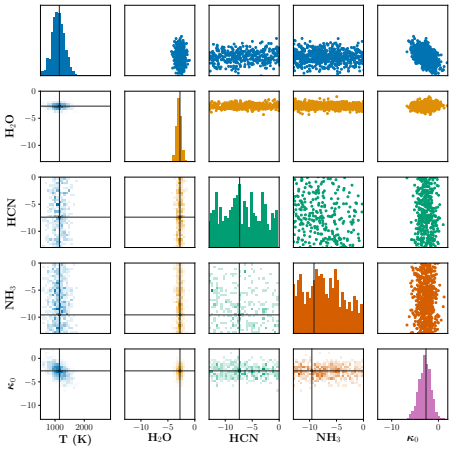
as defined in the `sklearn.metrics` Python package, where the summation is over both the size of the data set N and the output dimension D. $\tilde{\theta}^{(d)}$ is the data mean for each atmospheric parameter and the prediction for each data point is given by $\mu_{\text{ens}}^{(d)}(\mathbf{s}_n)$. This can be viewed as a ratio between the residuals for the model prediction and the total sum of squares. Values close to



(a) Random Forest



(b) `plan-net`



(c) `plan-net` Ensemble

Figure 2. Retrieval analysis of the WFC3 transmission spectrum of WASP-12b, where we compare the random forest with both a single `plan-net` and a `plan-net` ensemble. The black cross denotes the mean over the samples, where we report the results in Table 3. We note consistent results across all models, and highlight the broader posteriors of the ensemble when comparing to the single `plan-net`.

Table 1. Table reports R^2 values for each atmospheric parameter. Values near 1 indicate high correlation between model prediction and the known atmospheric parameters. `plan-net` achieves a higher overall mean R^2 as well as being higher for each individual parameter. Bold indicates the best R^2 value for each parameter.

	T(K)	$\log X_{H_2O}$	$\log X_{HCN}$	$\log X_{NH_3}$	κ_0	Mean
plan-net R^2	**0.770**	0.623	0.487	0.721	0.750	**0.673**
Ens. 5 plan-net R^2	**0.770**	**0.629**	**0.491**	**0.723**	**0.751**	**0.673**
Our Ran. Forest R^2	0.746	0.608	0.466	0.700	0.736	0.651
Ran. Forest[a] R^2	0.746	0.608	0.467	0.700	0.737	0.652

[a]Reported from Márquez-Neila et al. (2018).

1.0 are desirable as they are related to the correlation coefficient between the predicted and true atmospheric parameters.

Therefore, the results in Table 1 show that both of our models, the ensemble and the individual `plan-net` model, outperform the random forest. Furthermore, we note the slight performance boost that is gained from the ensemble. In order to show that the results are reproducible, we list both our implementation of the random forest and their reported results, which closely agree.

In addition to reporting the R^2 values, Table 2 contains the average covariance matrix over the test data. This table shows the average inferred correlations, where the diagonal corresponds to the variance in each atmospheric parameter and the off-diagonals indicate correlations between the parameters. As this is the average correlation matrix for all 20,000 test planets, not too many conclusions can be drawn from this matrix. However, we note the average negative correlation that appears between T(K) and κ_0 as well as T(K) and H_2O. This is consistent with intuition due to the known degeneracies in the data. More specifically, as the observed spectral features are caused by the temperature–pressure profile and the molecular abundances, increasing either whilst keeping all other parameters constant leads to stronger spectral features. Consequently, a simultaneous increase in temperature and a decrease in molecular abundances (or vice versa) could lead to the same observed spectrum. Finally, an increase in cloud opacity decreases the intensity of the observed spectral features and could therefore look similar to a decrease in temperature, hence the degeneracy and the expected negative correlation between T(K) and κ_0.

By designing our model to learn these correlations, we are able to interpret the results in a way that is not always available when using deep learning models. Specifically, we identify cases where both our model and the random forest approach do not recover the true values, but where our model includes the true values in its wider

Table 2. Mean inferred normalized correlation matrix, $(\mathbf{L}\mathbf{L}^\top)^{-1}$, across all test set atmospheric retrievals. The diagonal values are the mean marginalized variances for each parameter. The off-diagonals indicate correlations between these parameters; note the expected negative correlation between T(K) and κ_0 as well as T(K) and H_2O.

	T(K)	H_2O	HCN	NH_3	κ_0
T(K)	5.43	−7.50	−3.30	−4.88	−0.498
H_2O	−7.50	32.6	0.454	4.47	0.566
HCN	−3.30	0.454	56.7	1.37	1.95
NH_3	−4.88	4.47	1.37	12.1	0.965
κ_0	−0.498	0.566	1.95	0.965	3.74

posterior distributions. Figure 3 shows a case where the random forest infers narrow (highly confident) posterior distributions that fall far from the true values, whereas our `plan-net` ensemble model is (appropriately) less confident, leading to posterior distributions that cover the true values for the atmospheric parameters (shown by the red stars).

Given the performance over the synthetic test data set, we further test our models on the WFC3 transmission spectrum of WASP-12b. Figure 2 shows the posterior plots for the random forest, the single `plan-net` model and the `plan-net` ensemble. In the case of WASP-12b, both `plan-net`-based models find marginalized posteriors similar to the random forest for the cloud opacity (κ_0) and the abundances of HCN and NH_3. For temperature, both `plan-net`-based models have a distribution that is consistent with the retrieval performed in Kreidberg et al. (2015), while the random forest favors cooler temperatures. All models favor low ($\leq 10^{-7}$) abundances for HCN and NH_3, indicating a non-detection of these molecules. The H_2O abundance predicted by the individual `plan-net` model and the ensemble are more tightly constrained than the results of Márquez-Neila et al. (2018) or Kreidberg et al. (2015); see Table 3 for numerical comparisons[3]. Fisher & Heng (2018) found that, in general, WFC3 transmission spectra are adequately explained by an isothermal atmosphere (in the regions probed by transit observation), gray clouds, and H_2O only. Based on our ensemble's confidence in H_2O abundance (and lack of confidence in HCN and NH_3 abundances), it is likely that the model similarly learned this.

[3] Márquez-Neila et al. (2018) utilize a constant-opacity cloud parameterization, while Kreidberg et al. (2015) use a cloud and haze model that assumes an opaque gray cloud deck, which introduces degeneracies between the cloud and haze parameters. Consequently, a direct comparison between the two models cannot be made in Table 3.

4.1. *Limitations*

We highlight that employing variational approximate inference in BNNs is known to have problems, particularly underestimating uncertainty (Blei et al. 2017). Unlike the RF, our ensemble BNN model favors large uncertainties when the data cannot constrain a parameter, as shown in Figures 3 and 4. Though an ensemble of models helps to improve the uncertainty estimation, we emphasize that accurate uncertainty estimation requires using MCMC, nested sampling, or another Bayesian sampling algorithm proven to obtain accurate posterior distributions and therefore uncertainty estimations (e.g., ter Braak & Vrugt 2008).

Nevertheless, BNNs are presently an important tool for retrievals. They provide a reasonable estimation of parameters orders of magnitude faster than traditional methods that require hundreds of hours of CPU time, helping to constrain parameter spaces. As an example, a single `plan-net` prediction over a test planet takes 29.4 ms, when $T = 30$ samples, and an ensemble of five takes 1.5 s if they are run sequentially.[4]

As long as the data set used to train the model contains all relevant molecules, BNNs can inform which molecules should be considered in a traditional retrieval analysis based on retrieved abundances and their uncertainties. A single `plan-net` must be trained once for a certain class of planets, e.g., WFC3 transmission spectra of hot Jupiters. Once the model has been trained, all inferences with that model are fast and repeatable, for the class of planets represented in the training set. Therefore, although training the model can be computationally expensive, this only needs to be done once. In our example, each `plan-net` model takes 20 minutes to train over the WFC3 transmission spectra. Thus, despite the limitations of BNNs, their results are valuable and help save compute time spent on retrieval analyses.

Our approach is a generalizable technique that is not limited to any specific type of planet. In addition, important parameters such as the radius of the planet should be included in future models, as in this paper we make use of the data set of Márquez-Neila et al. (2018) which does not include it in the parameter space. Therefore the challenge in using BNNs comes from ensuring that the data set contains both the parameter space and planet types of interest.

[4] Hardware: Ubuntu 18.04, 32GB memory, CPU: Intel Core i7-8700K, GPU: TITAN Xp

Table 3. Retrieved atmospheric parameters for WASP-12b. All retrievals are consistent, with our ensemble `plan-net` model achieving closer agreement with the temperature and H_2O abundance retrieved by Kreidberg et al. (2015). We note that Kreidberg et al. (2015) did not retrieve for $\log X_{\rm HCN}$ and $\log X_{\rm NH_3}$. They also used a different cloud parameterization that makes κ_0 not applicable to their model. Errors are reported for one standard deviation, where we report the median and equivalent asymmetric posterior percentiles for the random forest and for Kreidberg et al. (2015).

	T(K)	$\log X_{\rm H_2O}$	$\log X_{\rm HCN}$	$\log X_{\rm NH_3}$	κ_0
Kreidberg et al. (2015)	1371^{+466}_{-343}	$-2.7^{+1.0}_{-1.1}$	-	-	-
Márquez-Neila et al. (2018) nested sampling	1105^{+545}_{-287}	$-3.0^{+2.0}_{-1.9}$	$-8.5^{+3.8}_{-2.9}$	$-8.4^{+3.1}_{-2.9}$	-2.8 ± 0.9
Our Rand. Forest	937^{+410}_{-146}	$-2.835^{+1.51}_{-3.37}$	$-7.484^{+3.43}_{-2.89}$	$-9.202^{+4.12}_{-2.74}$	$-2.281^{+1.09}_{-1.57}$
Ens. 5 plan-net	1142 ± 412	-2.781 ± 0.429	-8.210 ± 12.7	-9.605 ± 6.7	-2.601 ± 1.23

5. CONCLUSIONS

In this paper, we have demonstrated how domain-knowledge can be used to design a machine learning model that both outperforms the previous approach and provides inferred correlations between its outputs. Furthermore, we have introduced a novel likelihood function for BNNs which captures correlations between output dimensions. This extends on the diagonal Gaussian likelihood often used in the literature that does not capture these correlations. We highlight that this is extremely easy to do with BNNs and stochastic approximate inference, when comparing to traditional ML techniques (e.g. Gaussian processes), where it would involve many more approximations.

Using the data set of Márquez-Neila et al. (2018), we independently reproduced the results of their RF. For the first time, we have shown that ML retrieval results are reproducible and consistent across implementations.

In addition to comparing our approach to the random forest using 20,000 test planet models, we also analyzed the inferred posteriors for WASP-12b, where we take the results of Kreidberg et al. (2015) to be the ground truth. Our ensemble of five `plan-net` models gives results consistent with the RF of Márquez-Neila et al. (2018) and achieved distributions for H_2O abundance and temperature that agree more closely with Kreidberg et al. (2015) and the nested sampling retrieval of Márquez-Neila et al. (2018) than the RF. The low ($< 10^{-7}$) retrieved abundances and large uncertainties of HCN and NH_3 indicate a non-detection of these molecules.

Furthermore, we have found that an ensemble of BNNs provides posterior distributions that better represent those of traditional Bayesian atmospheric retrieval methods, compared to both a single BNN model and the RF model. A single `plan-net` model can underestimate the size of the posterior distributions due to overconfidence in their predictions, while the RF can be overconfident in a wrong answer.

We have presented the first study that employs BNNs for atmospheric retrievals, setting the foundation for further research in this area. As the data available for atmospheric retrievals expands, it will become increasingly important to combine domain-knowledge with machine learning models. It is equally important that it remains possible to interpret the outputs of these models so that inferences can be physically understood. Our method easily scales to higher dimensionality; in future work, we will expand our model to higher resolution spectra and a larger number of atmospheric parameters.

We thank Chloe Fisher for making the data set from Márquez-Neila et al. (2018) publicly available on GitHub upon request. Adam D. Cobb is sponsored by the AIMS CDT (http://aims.robots.ox.ac.uk) and the EPSRC (https://www.epsrc.ac.uk). Frank Soboczenski gratefully acknowledges the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research (GPU No 900-1G611-2530-000). A.G. Baydin is funded by Lawrence Berkeley National Lab and EPSRC/MURI grant EP/N019474/1. We thank NASA FDL (http://www.frontierdevelopmentlab.org/) and SETI (https://www.seti.org) for making this collaboration possible.

REFERENCES

Abadi, M., Barham, P., Chen, J., et al. 2016, in OSDI, Vol. 16, 265–283

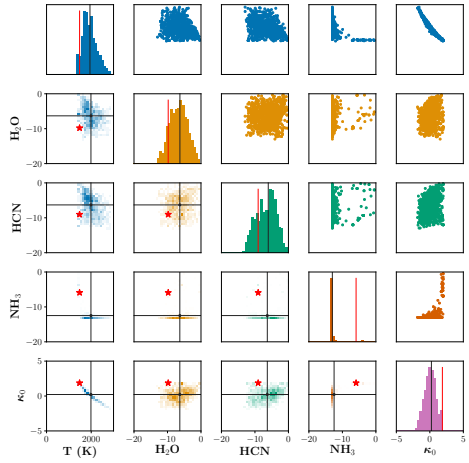
Ansdell, M., Ioannou, Y., Osborn, H. P., et al. 2018, ApJL, 869, L7

Batalha, N. M. 2014, Proceedings of the National Academy of Science, 111, 12647

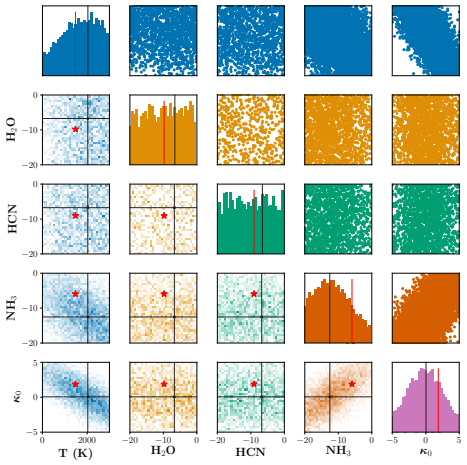
Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. 2017, Journal of the American Statistical Association, 112, 859

Charbonneau, D., Allen, L. E., Megeath, S. T., et al. 2005, ApJ, 626, 523
Chollet, F., et al. 2015, Keras, ,
Criminisi, A., Shotton, J., Konukoglu, E., et al. 2012, Foundations and Trends® in Computer Graphics and Vision, 7, 81
Deming, D., Seager, S., Richardson, L. J., & Harrington, J. 2005, Nature, 434, 740
Dorta, G., Vicente, S., Agapito, L., Campbell, N. D., & Simpson, I. 2018, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5477–5485
Fisher, C., & Heng, K. 2018, MNRAS, 481, 4698
Gal, Y. 2016, PhD thesis, PhD thesis, University of Cambridge
Gal, Y., & Ghahramani, Z. 2016, in International Conference on Machine Learning, 1050–1059
Gal, Y., Hron, J., & Kendall, A. 2017, in Advances in Neural Information Processing Systems, 3581–3590
Gal, Y., & Smith, L. 2018, arXiv preprint arXiv:1806.00667
Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. 2014, in Advances in Neural Information Processing Systems, 2672–2680
Hasegawa, Y., & Pudritz, R. E. 2013, ApJ, 778, 78
Heng, K., & Kitzmann, D. 2017, MNRAS, 470, 2972
Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. 2013, The Journal of Machine Learning Research, 14, 1303
Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. 1998, in Learning in graphical models (Springer), 105–161
Kingma, D. P., & Ba, J. 2014, arXiv preprint arXiv:1412.6980
Kreidberg, L. 2017, Exoplanet Atmosphere Measurements from Transmission Spectroscopy and Other Planet Star Combined Light Observations (Springer International Publishing), 100
Kreidberg, L., Line, M. R., Bean, J. L., et al. 2015, ApJ, 814, 66
Lakshminarayanan, B., Pritzel, A., & Blundell, C. 2017, in Advances in Neural Information Processing Systems, 6402–6413
Line, M. R., Knutson, H., Wolf, A. S., & Yung, Y. L. 2014, ApJ, 783, 70
MacDonald, R. J., & Madhusudhan, N. 2017, ApJL, 850, L15
MacKay, D. J. 1992, Neural computation, 4, 448
Madhusudhan, N. 2018, ArXiv e-prints, arXiv:1808.04824
Madhusudhan, N., & Seager, S. 2009, ApJ, 707, 24
—. 2010, ApJ, 725, 261
Márquez-Neila, P., Fisher, C., Sznitman, R., & Heng, K. 2018, Nature Astronomy, arXiv:1806.03944
Neal, R. M. 1995, PhD thesis, University of Toronto
Oreshenko, M., Lavie, B., Grimm, S. L., et al. 2017, ApJL, 847, L3
Osborn, H. P., Ansdell, M., Ioannou, Y., et al. 2019, arXiv preprint arXiv:1902.08544
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Perreault Levasseur, L., Hezaveh, Y. D., & Wechsler, R. H. 2017, ApJ, 850, L7
Roberts, S., McQuillan, A., Reece, S., & Aigrain, S. 2013, MNRAS, 435, 3639
Shallue, C. J., & Vanderburg, A. 2018, AJ, 155, 94
Skilling, J. 2004, in American Institute of Physics Conference Series, Vol. 735, American Institute of Physics Conference Series, ed. R. Fischer, R. Preuss, & U. V. Toussaint, 395–405
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, The Journal of Machine Learning Research, 15, 1929
ter Braak, C. 2006, Statistics and Computing, 16, 239. http://dx.doi.org/10.1007/s11222-006-8769-1
ter Braak, C. J. F., & Vrugt, J. A. 2008, Statistics and Computing, 18, 435. http://dx.doi.org/10.1007/s11222-008-9104-9
Waldmann, I. P. 2016, ApJ, 820, 107
Waldmann, I. P., Rocchetto, M., Tinetti, G., et al. 2015, ApJ, 813, 13
Yeh, R. A., Chen, C., Yian Lim, T., et al. 2017, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5485–5493
Zingales, T., & Waldmann, I. P. 2018, ArXiv e-prints, arXiv:1806.02906

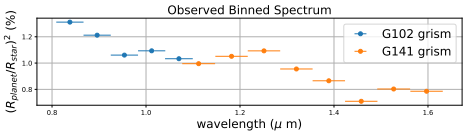
APPENDIX



(a) Random Forest

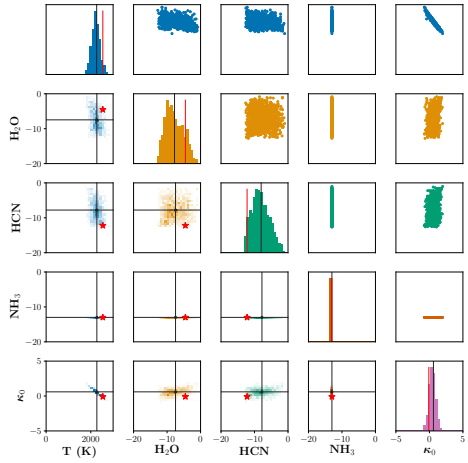


(b) `plan-net` Ensemble

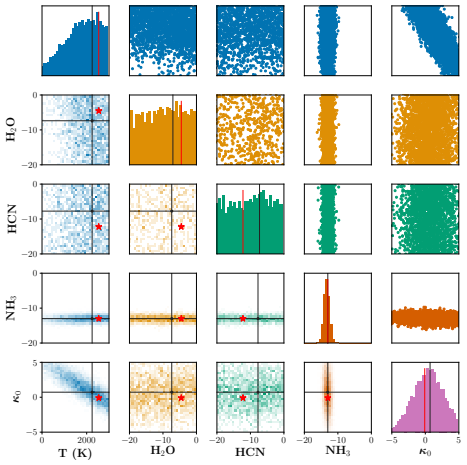


(c) Test Spectrum

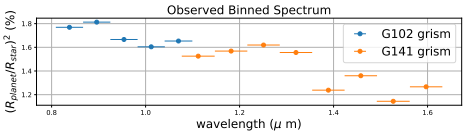
Figure 3. Test planet 1: an example taken from the test set, where the random forest is overconfident and far from the true parameter values, denoted by the red star. In comparison, the `plan-net` ensemble demonstrates its uncertainty in its predicted values by inferring broader posterior distributions that cover the true parameters. Figure 3c gives the observed input spectrum, where the binning is given by Table 3 in Kreidberg et al. (2015). Each spectral coverage of the wavelengths is given by two grisms indicated in the legend.



(a) Random Forest

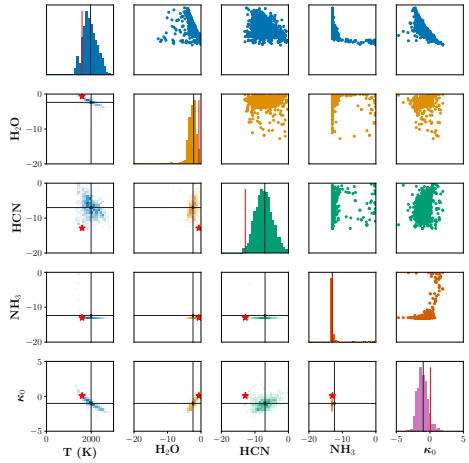


(b) `plan-net` Ensemble

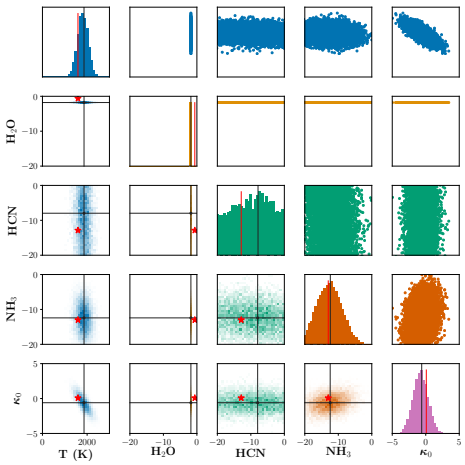


(c) Test Spectrum

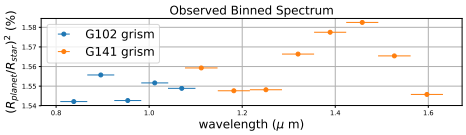
Figure 4. Test planet 2: an example taken from the test set, where both models retrieve parameters close to the true labels, as denoted by the red stars. However, like in Figure 3, the random forest demonstrates highly confident posteriors, where it may not be appropriate. Figure 4c gives the observed input spectrum, where the binning is given by Table 3 in Kreidberg et al. (2015). Each spectral coverage of the wavelengths is given by two grisms indicated in the legend.



(a) Random Forest



(b) `plan-net` Ensemble



(c) Test Spectrum

Figure 5. Test planet 3: an example taken from the test set where the H_2O abundance is high, allowing it to be tightly constrained. Note that `plan-net` is unable to constrain the HCN abundance, whereas the RF makes a confident over-prediction; with an abundance of $< 10^{-10}$, HCN would be difficult to constrain for traditional methods (MacDonald & Madhusudhan 2017). Figure 5c gives the observed input spectrum, where the binning is given by Table 3 in Kreidberg et al. (2015). Each spectral coverage of the wavelengths is given by two grisms indicated in the legend.