Atmospheric retrievals of exoplanets using learned parameterizations of pressure-temperature profiles

Timothy D. Gebhard
Max Planck Institute for Intelligent Systems, Tübingen, Germany
Institute for Particle Physics & Astrophysics, ETH Zurich, Switzerland
timothy.gebhard@tuebingen.mpg.de

Daniel Angerhausen
Institute for Particle Physics & Astrophysics, ETH Zurich, Switzerland

Björn Konrad
Institute for Particle Physics & Astrophysics, ETH Zurich, Switzerland

Eleonora Alei
Institute for Particle Physics & Astrophysics, ETH Zurich, Switzerland

Sascha P. Quanz
Institute for Particle Physics & Astrophysics, ETH Zurich, Switzerland

Bernhard Schölkopf
Max Planck Institute for Intelligent Systems, Tübingen, Germany
Department of Computer Science, ETH Zurich, Switzerland

Abstract

We describe a new, learning-based approach for parameterizing the relationship between pressure and temperature in the atmosphere of an exoplanet. Our method can be used, for example, when estimating the parameters characterizing a planet's atmosphere from an observation of its spectrum with Bayesian inference methods ("atmospheric retrieval"). On two data sets, we show that our method requires fewer parameters and achieves, on average, better reconstruction quality than existing methods, all while still integrating easily into existing retrieval frameworks. This may help the analysis of exoplanet observations as well as the design of future instruments by speeding up inference, freeing up resources to retrieve more parameters, and paving a way to using more realistic atmospheric models for retrievals.

1 Introduction

Atmospheric retrievals (ARs) are an important inference problem in the field of exoplanet science. The term refers to the task where, given an observed empirical spectrum of a planet (i.e., photon flux as a function of wavelength), one tries to infer a set of parameters θ that describes the atmosphere of the planet, such as the chemical composition, the presence of clouds, or the thermal structure [28]. This is a classic inverse problem: While the forward direction (i.e., going from parameters to a spectrum) can be solved approximately using analytical prescriptions and simulators, the backward direction is generally much more challenging. A standard approach to ARs is to combine a forward simulator

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

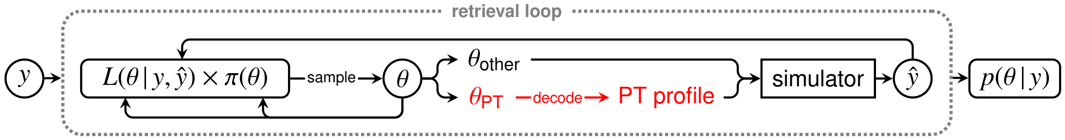


Figure 1: A typical retrieval workflow takes an observation y (a spectrum of an exoplanet) and, usually via nested sampling, infers from it a posterior distribution for the parameters θ that describe the planet's atmosphere. Marked in red is the part of the retrieval loop that we focus on in this paper.

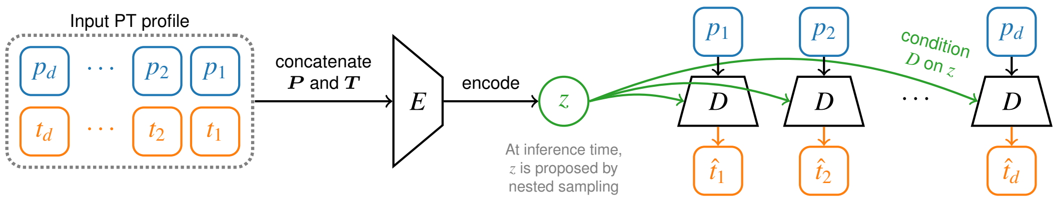


Figure 2: A PT profile is encoded into a latent variable $z \in \mathbb{R}^n$, which is used to condition the decoder D. Once conditioned, $D(\cdot \mid z)$ is a function that maps (log)-pressures onto temperatures.

with Bayesian inference methods such as nested sampling [37]: by repeatedly sampling a value for the parameters of interest (using a prior $\pi(\theta)$), simulating the corresponding spectrum $\hat{y}$, and comparing it to the observation y through some likelihood function L, one can compute a posterior $p(\theta \mid y)$. We illustrate this process (including the aspects described in the next paragraph) in fig. 1.

One key ingredient for this process is the pressure-temperature (PT) profile of the atmosphere, which is a function that maps pressures onto temperatures and thus provides a one-dimensional approximation of the thermal structure.[1] To find the PT profile of an atmosphere during a retrieval, one may either solve a set of radiative transfer (RT) equations, or use an ad hoc fitting function that provides the PT profile in a parameterized form [22, 25, 35]. Due to the available spectral resolution (i.e., the size of the wavelength bins in which the photon flux is sampled) and computational concerns, in practice, the latter approach is often preferred, and different parameterizations have been proposed in the literature. Some approaches [9, 20, 25, 27] assume particular analytical descriptions for the relationship between P and T, while others use splines [41] or low-order polynomials [24]. These approaches typically use 4–6 parameters, which is a substantial fraction of $\dim(\theta)$ (usually in the range 10 to 15) and assume a fixed functional form. Nevertheless, they still provide only a simplified approximation of the PT profile that one would find using RT and, of course, they also allow to parameterize many functions that are not valid PT profiles, thus reducing sampling efficiency.

In this work, we present a fundamentally new approach to parameterizing PT profiles, which does not assume a fixed functional form for the mapping $P \mapsto T$, but instead uses a conditional latent variable model to learn efficient neural representations of realistic, physically consistent PT profiles from data.

2 Method

Our goal, in principle, is to learn a distribution over functions $P \mapsto T$ that can be sampled efficiently during an AR to find the PT profile that best matches the observational data. The method that we propose for this is essentially a reinterpretation of a neural process [14, 15], although additional inspiration came from works such as Dupont *et al.* [11]. The general idea, illustrated in fig. 2, is to use two neural networks; an encoder E and a (conditional) decoder D. At training time, E takes in a PT profile, consisting of a vector $\boldsymbol{P} = (p_1, \ldots, p_d)$ of pressure values[2] and a vector $\boldsymbol{T} = (t_1, \ldots, t_d)$ of corresponding temperatures, and outputs a latent variable $z \in \mathbb{R}^n$, that is, $z = E(\boldsymbol{P}, \boldsymbol{T})$. We can think of z as an abstract representation of the PT profile $(\boldsymbol{P}, \boldsymbol{T})$. The decoder D is now conditioned on z, and $D(\cdot \mid z)$ is evaluated on $\boldsymbol{P}$ to produce $\hat{\boldsymbol{T}} = (D(p_1 \mid z), \ldots, D(p_p \mid z))$. We train E and D jointly to minimize the distance between $\boldsymbol{T}$ and $\hat{\boldsymbol{T}}$, subject to some constraints on z: During an AR, the

[1] In reality, of course, the thermal structure is also a function of the location—the North Pole is cooler than the equator. However, atmospheric retrievals with full 3D climate models are currently computationally infeasible.

[2] For simplicity, we always talk about the pressure P; in practice, our implementation actually uses $\log_{10}(P)$.

nested sampling procedure will propose values for z and use D to generate PT profiles. For this to work properly, we want to make sure that similar PT profiles are assigned similar values of z, and that the latent space does not have any "holes" where z cannot be decoded into a sensible PT profile.

To place such a prior on z, we do not use the common evidence lower-bound (ELBO) objective, but instead take inspiration from the idea of an MMD-VAE [42, 43]. MMD-VAEs are a variant of variational auto-encoders that replace the usual ELBO loss with a kernel-based estimate of the distance between two distributions, namely the maximum mean discrepancy (MMD) [18, 19]. The main reason for this choice is that, for sufficiently powerful decoders, training with the ELBO objective can result in a model that ignores the latent variable [5]. This is not desirable for our application; if we want to use the trained decoder in a retrieval loop, we want to be sure that the mutual information between z and $\hat{\boldsymbol{T}}$ is maximal. Ultimately, we are minimizing the following objective function:

$$\mathcal{L}(z, \boldsymbol{T}, \hat{\boldsymbol{T}}) = \text{mean}\left((\boldsymbol{T} - \hat{\boldsymbol{T}})^2\right) + \beta \cdot \text{MMD}(z, S)\,, \tag{1}$$

which corresponds to the assumption of a Gaussian likelihood function. S denotes a sample from a n-dimensional standard normal distribution; the size of S matches the batch size. The factor $\beta \in \mathbb{R}^+$ controls the trade-off between the reconstruction error and the prior on z. We chose $\beta = 1000$ here.

3 Data sets, model specification, implementation, and training details

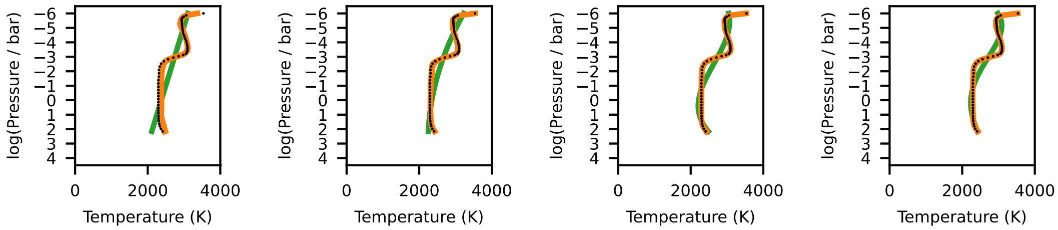
Data sets We use two different data sets (for two different types of exoplanets) to experimentally validate our proposed method: (1) The PyAtmos data set [3, 6, 7] consists of 124 314 physically and chemically self-consistent atmospheres of Earth-like planets around a solar-type star, which were obtained with Atmos [2, 30], a one-dimensional coupled photochemistry-climate model. (2) The Goyal-2020 data set [17] contains 11 290 atmospheres of hot Jupiter-like planets simulated with Atmo [1, 10]. We split both data sets randomly into a training and a test set: For PyAtmos, we use 100k PT profiles for training and validation; for Goyal-2020, we use 10k. The rest of the data are held out for evaluation.

Model specification For both for the encoder and the decoder, we use a simple multi-layer perceptron (MLP) with LeakyReLU activation functions. Encoders use 5 layers with 512 neurons; decoders have 6 layers with 256 neurons. We have also briefly experimented with encoders akin to the original NP work [14, 15] (i.e., encode each point of a PT profile independently as $z_i = E(p_i, t_i)$ and then aggregate them as $z = \text{mean}(z_i)$), as well as SIREN-like decoders [36], and found that they give very similar performance. Future work may study this in more detail or perform proper hyper-parameter optimization; however, for this first proof-of-concept, even simple MLPs give very good results.

Implementation and training details We implement all of our experiments in PyTorch [32] in combination with the Lightning wrapper [12]. Models are trained with `AdamW` [26] for up to 1000 epochs using a random 90%/10% split for training and validation. The initial learning rate is set to 3×10^{-4}, and we use a `ReduceLROnPlateau(patience=20, factor=0.5)` scheduler to decrease it throughout training. The batch size is set to 1024 (PyAtmos) or 256 (Goyal-2020). If the validation loss does not decrease for 100 epochs, we stop early, and we only save the model with the lowest validation loss. Training is fast: Models usually converge in < 1 hour when using a modern GPU (e.g., NVIDA V100).

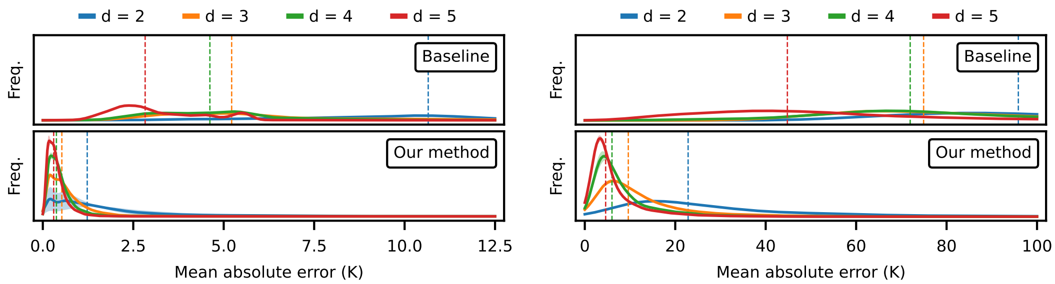
4 Experiments and results

Reconstruction quality for different dim(z) As a first experiment, we train models on both of our data sets for $\dim(z) \in \{2, 3, 4, 5\}$. Then, we use the encoder E to compute an initial encoding z_{initial} for each PT profile in the test set. This initial value is then refined using an AdamW optimizer to find $z_{\text{refined}} = \text{argmin}_z(\boldsymbol{T} - D(\boldsymbol{P}\,|\,z))^2$. After all, we cannot use the encoder during a retrieval, and we are only interested how well the decoder—using the "correct" z—can approximate a given PT profile. We repeat all these steps three times for every model using different random seeds that control the initialization and the train / validation split. Exemplary results are shown in fig. 3, where we compare the fit using our trained decoder with a polynomial with the same number of free parameters. We find that, for all values of $\dim(z)$, our model gives a much better approximation of the PT profile than the baseline; in particular, sharp kinks are reproduced much better than what is possible with a low-order polynomial. In fig. 4, we provide more systematic evidence for this. For all PT profiles in the test set, we have computed the mean absolute difference between the true and the best fit profile. Looking at the distributions, we find that even for $\dim(z) = 2$, our model has (on average) a lower



(a) $d = 2$ (b) $d = 3$ (c) $d = 4$ (d) $d = 5$

Figure 3: A random PT profile from the GOYAL-2020 test set (black dots), and the best fit with both our trained model (orange) and a polynomial (green). By d, we denote the number of fitting parameters, that is, $\dim(z)$ for our model, or the number of coefficients of of the polynomial. The uncertainties on our model (due to different random seeds) are too small to be visible in this plot.



(a) Results for PYATMOS data set. (b) Results for GOYAL-2020 data set.

Figure 4: Distribution of the mean absolute error (per profile on the test set) for our method and the baseline. By d, we again denote the number of fitting parameters. The dashed lines mark the median of the respective distribution; see table 1 for exact values. Error bands are due to multiple random seeds.

error than the 5-parameter baseline. Additionally, unlike for the baseline, the utility of additional parameters decreases for our method, making it easier to trade off accuracy against $\dim(z)$.

Atmospheric retrieval with a learned PT profile parameterization To show that our method is readily integrated into existing AR frameworks and yields reliable results, we add our model to the AR routine introduced in Konrad *et al.* [24], which is based on petitRADTRANS [31] and PyMultiNest [4]. We run an AR on the spectrum of an Earth-twin at 10 pc orbiting a Sun-like star with a photon noise signal-to-noise ratio of 10. We compare the performance of a version of our model with $\dim(z) = 2$ to a fourth-order polynomial PT model (i.e., five fitting parameters). The results of this experiment are shown in fig. 5. We find that the posterior on the thermal structure obtained with our model matches the ground truth well, except at the lowest pressures. However, as shown by the emission contribution function, these layers contribute virtually nothing to the observed spectrum, making them hard to constrain. Despite using only 2 instead of 5 fitting parameters, our model gives a (visually) better posterior than the baseline, in particular for the surface pressure and temperature.

5 Discussion and outlook

Exoplanet science is still a fast-growing field, and with now 5000 confirmed detections [8], the characterization of planets is becoming a main focus of observations and the development of future instruments. Our method may be helpful for both of these aspects: Speeding up ARs, or allowing to retrieve additional parameters of interest (e.g., chemical species in the atmosphere) does not only benefit the analysis of actual observational data (from, e.g., the James Webb Space Telescope), but is also valuable for the design phase of new generation exoplanet instruments and large international space missions such as LIFE [34] or LUVOIR/HabEx [16, 39], which require thousands of simulated retrievals.

Of course, crucially, the validity of our approach hinges on the availability of appropriate training data: there is no guarantee for our method to perform well if the true PT profile is not covered by the training distribution. Generating suitable grids of self-consistent atmospheres covering a broad range of exo-

Table 1: Median of the distribution of the mean absolute error on the test set (in Kelvin), for both our datasets, and for both our model and the polynomial baseline. We denote the number of fitting parameters as d. The values here correspond directly to the dashed lines in fig. 4.

	PyAtmos				Goyal-2020			
	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
Baseline	10.65	5.22	4.62	2.83	95.87	74.92	71.95	44.79
Our method	1.23	0.53	0.38	0.30	22.85	9.62	6.01	4.61

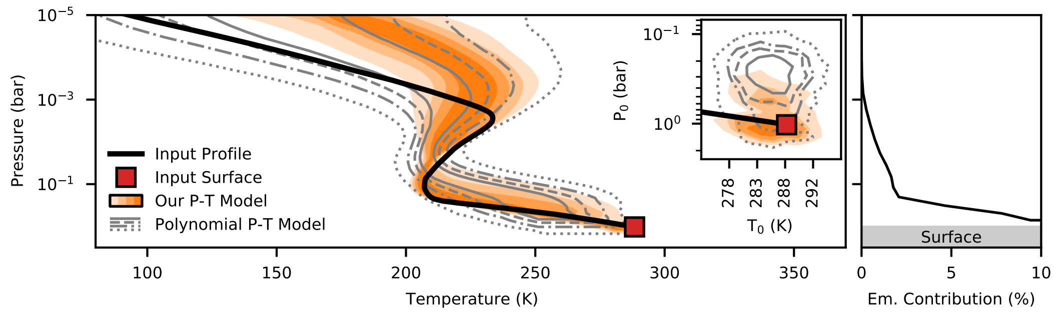


Figure 5: Results of our retrieval experiment: The left panel shows the input PT profile that was used to simulate the "observed" spectrum, as well as the posteriors obtained using our model and the polynomial baseline. The different shades (or line types) indicate percentiles: 5%-95%, 15%-85%, 25%-75%, and 35%-65%. The inset in the top right corner shows the posterior for the surface pressure P_0 and temperature T_0. The panel on the right displays the emission contribution function, that is, a relative measure of how much the atmosphere at a given pressure contributes to the overall spectrum.

planets types and parameters likely requires a joint community effort. However, as more such data sets become available, we may think about extending our method further and use it, for instance, not only for PT profiles, but also to parameterize abundances of chemical species as a function of pressure.

Acknowledgements

TDG acknowledges partial funding through the Max Planck ETH Center for Learning Systems. Part of this work has been carried out within the framework of the NCCR PlanetS supported by the Swiss National Science Foundation. We gratefully acknowledge the funding we have received through the European Space Agency's Open Space Innovation Platform (ESA OSIP) program.

This research has made use of the following Python packages: corner.py [13], matplotlib [23], numpy [21], pandas [29, 38], PyTorch [32], PyTorch Lightning [12], scikit-learn [33], and scipy [40].

Broader impact statement

The authors are not aware of any immediate ethical or societal implications of this work. Thinking more broadly, methods for characterizing atmospheres of exoplanets may contribute to our understanding of the conditions supporting life, and thus ultimately help communicate the public the rather unique conditions that our own planet's atmosphere currently offers. This may contribute to a broader understanding of the value of our ecosphere, and the importance of measures to mitigate climate change.

References

[1] D. S. Amundsen *et al.*, "Accuracy tests of radiation schemes used in hot Jupiter global circulation models," *Astronomy & Astrophysics*, volume 564: A59, 2014. DOI: 10.1051/0004-6361/201323169.

[2] G. Arney *et al.*, "The Pale Orange Dot: The Spectrum and Habitability of Hazy Archean Earth," *Astrobiology*, volume 16(11): 873–899, 2016. DOI: 10.1089/ast.2015.1422.

[3] A. Bell *et al.*, "NASA Frontier Development Lab 2018: Using machine learning to study E.T. biospheres," in *5th Workshop on Challenges in Machine Learning (NeurIPS)*, 2018. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2018cml..confE...1B.

[4] J. Buchner *et al.*, "X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue," *Astronomy & Astrophysics*, volume 564: A125, 2014. DOI: 10.1051/0004-6361/201322971.

[5] X. Chen *et al.*, "Variational Lossy Autoencoder," in *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1611.02731.

[6] A. Chopra *et al.* "About the FDL PyATMOS dataset." (2018), [Online]. Available: https://exoplanetarchive.ipac.caltech.edu/docs/fdl_landing.html.

[7] A. Chopra *et al.*, "PyATMOS: A Scalable Grid of Hypothetical Planetary Atmospheres," Under Review.

[8] J. L. Christiansen, "Five thousand exoplanets at the NASA Exoplanet Archive," *Nature Astronomy*, volume 6(5): 516–519, 2022. DOI: 10.1038/s41550-022-01661-8.

[9] I. Dobbs-Dixon and J. Blecic, "GCM-motivated Multidimensional Temperature Parametrization Scheme for Phase-curve Retrieval," *The Astrophysical Journal*, volume 929(1): 46, 2022. DOI: 10.3847/1538-4357/ac5898.

[10] B. Drummond *et al.*, "The effects of consistent chemical kinetics calculations on the pressure-temperature profiles and emission spectra of hot Jupiters," *Astronomy & Astrophysics*, volume 594: A69, 2016. DOI: 10.1051/0004-6361/201628799.

[11] E. Dupont, Y. W. Teh, and A. Doucet. "Generative Models as Distributions of Functions." arXiv:2102.04776. (2021).

[12] W. Falcon and The PyTorch Lightning team, *PyTorch Lightning*, 2019. [Online]. Available: https://github.com/Lightning-AI/lightning/.

[13] D. Foreman-Mackey, "corner.py: Scatterplot matrices in Python," *The Journal of Open Source Software*, volume 1(2): 24, 2016. DOI: 10.21105/joss.00024.

[14] M. Garnelo *et al.* "Conditional Neural Processes." arXiv:1807.01613. (2018).

[15] M. Garnelo *et al.* "Neural Processes." arXiv:1807.01622. (2018).

[16] B. S. Gaudi *et al.* "The Habitable Exoplanet Observatory (HabEx) Mission Concept Study Final Report." arXiv:2001.06683. (2020).

[17] J. M. Goyal *et al.*, "A library of self-consistent simulated exoplanet atmospheres," *Monthly Notices of the Royal Astronomical Society*, volume 498(4): 4680–4704, 2020. DOI: 10.1093/mnras/staa2300.

[18] A. Gretton *et al.*, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., volume 19, MIT Press, 2006. [Online]. Available: https://proceedings.neurips.cc/paper/2006/file/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Paper.pdf.

[19] A. Gretton *et al.*, "A kernel two-sample test," *Journal of Machine Learning Research*, volume 13(25): 723–773, 2012. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html.

[20] T. Guillot, "On the radiative equilibrium of irradiated planetary atmospheres," *Astronomy & Astrophysics*, volume 520: A27, 2010. DOI: 10.1051/0004-6361/200913396.

[21] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, volume 585(7825): 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

[22] K. Heng, *Exoplanetary Atmospheres*. Princeton University Press, 2017. DOI: 10.1515/9781400883073.

[23] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, volume 9(3): 90–95, 2007. DOI: 10.1109/mcse.2007.55.

[24] B. S. Konrad *et al.*, "Large Interferometer For Exoplanets (LIFE) — III. Spectral resolution, wavelength range, and sensitivity requirements based on atmospheric retrieval analyses of an exo-Earth," *Astronomy & Astrophysics*, volume 664: A23, 2022. DOI: 10.1051/0004-6361/202141964.

[25] M. R. Line *et al.*, "A Systematic Retrieval Analysis Of Secondary Eclipse Spectra. I. A Comparison Of Atmospheric Retrieval Techniques," *The Astrophysical Journal*, volume 775(2): 137, 2013. DOI: 10.1088/0004-637x/775/2/137.

[26] I. Loshchilov and F. Hutter. "Decoupled Weight Decay Regularization." arXiv:1711.05101. (2017).

[27] N. Madhusudhan and S. Seager, "A Temperature And Abundance Retrieval Method For Exoplanet Atmospheres," *The Astrophysical Journal*, volume 707(1): 24–39, 2009. DOI: 10.1088/0004-637x/707/1/24.

[28] N. Madhusudhan, "Atmospheric Retrieval of Exoplanets," in *Handbook of Exoplanets*, Springer International Publishing, 2018, pages 1–30. DOI: 10.1007/978-3-319-30648-3_104-1.

[29] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, 2010, pages 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[30] V. S. Meadows *et al.*, "The Habitability of Proxima Centauri b: Environmental States and Observational Discriminants," *Astrobiology*, volume 18(2): 133–189, 2018. DOI: 10.1089/ast.2016.1589.

[31] P. Mollière *et al.*, "petitRADTRANS: A Python radiative transfer package for exoplanet characterization and retrieval," *Astronomy & Astrophysics*, volume 627: A67, 2019. DOI: 10.1051/0004-6361/201935470.

[32] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, 2019, pages 8024–8035.

[33] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, volume 12: 2825–2830, 2011.

[34] Quanz, S. P. *et al.*, "Large Interferometer For Exoplanets (LIFE) — I. Improved exoplanet detection yield estimates for a large mid-infrared space-interferometer mission," *Astronomy & Astrophysics*, volume 664: A21, 2022. DOI: 10.1051/0004-6361/202140366.

[35] S. Seager, *Exoplanet Atmospheres*. Princeton University Press, 2010. DOI: 10.1515/9781400835300.

[36] V. Sitzmann *et al.* "Implicit Neural Representations with Periodic Activation Functions." arXiv:2006.09661. (2020).

[37] J. Skilling, "Nested sampling for general Bayesian computation," *Bayesian Analysis*, volume 1(4), 2006. DOI: 10.1214/06-ba127.

[38] T. pandas development team, *pandas-dev/pandas: Pandas*, version latest, 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: https://doi.org/10.5281/zenodo.3509134.

[39] The LUVOIR Team. "The LUVOIR Mission Concept Study Final Report." arXiv:1912.06219. (2019).

[40] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, volume 17: 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[41] Y. Zhang *et al.*, "The ^{13}CO-rich atmosphere of a young accreting super-Jupiter," *Nature*, volume 595(7867): 370–372, 2021. DOI: 10.1038/s41586-021-03616-x.

[42] S. Zhao. "A Tutorial on Information Maximizing Variational Autoencoders (InfoVAE)." (2017), [Online]. Available: https://szhao.me/a-tutorial-on-mmd-variational-autoencoders/.

[43] S. Zhao, J. Song, and S. Ermon. "InfoVAE: Information Maximizing Variational Autoencoders." arXiv:1706.02262. (2017).

NeurIPS paper checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Yes
   (b) Have you read the ethics review guidelines and ensured that your paper conforms to them?
   Yes
   (c) Did you discuss any potential negative societal impacts of your work?
   Yes
   (d) Did you describe the limitations of your work?
   Yes,
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?
   Not applicable
   (b) Did you include complete proofs of all theoretical results?
   Not applicable
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   No (but we plan to release the code in a journal version that is currently in preparation).
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
   Yes
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   Yes
   (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?
   Yes
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators?
   Yes (data sets and Python packages)
   (b) Did you mention the license of the assets?
   No (the data sets do not explicitly state a license)
   (c) Did you include any new assets either in the supplemental material or as a URL?
   No
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?
   Not applicable
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?
   Not applicable
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable?
   Not applicable
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?
   Not applicable
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?
   Not applicable