Inferring Atmospheric Properties of Exoplanets with Flow Matching and Neural Importance Sampling

Anonymous submission

Abstract

Atmospheric retrievals characterize exoplanets by estimating atmospheric parameters from observed light spectra, typically by framing the task as a Bayesian inference problem. However, traditional approaches such as nested sampling are computationally expensive, thus sparking an interest in solutions based on machine learning (ML). In this ongoing work, we first explore flow matching posterior estimation (FMPE) as a new ML-based method for AR and find that, in our case, it is more accurate than neural posterior estimation (NPE), but less accurate than nested sampling. We then combine both FMPE and NPE with importance sampling, in which case both methods outperform nested sampling in terms of accuracy and simulation efficiency. Going forward, our analysis suggests that simulation-based inference with likelihood-based importance sampling provides a framework for accurate and efficient AR that may become a valuable tool not only for the analysis of observational data from existing telescopes, but also for the development of new missions and instruments.

Introduction

"NASA Says Distant Exoplanet Could Have Rare Water Ocean" (Luscombe 2023) — Headlines like this have recently made it even into the mainstream news. But how do we know what happens on (and above) the surface of planets outside our solar system? In many cases, the answer is *atmospheric retrievals* (AR), that is, "the inference of atmospheric properties of an exoplanet given an observed spectrum" (Madhusudhan 2018). These properties include the abundances of chemical species (e.g., water or methane), the thermal structure, or the presence of clouds. In practice, performing an AR usually means combining a simulator for the forward direction (i.e., parameters $\rightarrow$ spectrum) with a Bayesian inference technique such as nested sampling (Skilling 2006; Ashton et al. 2022) to compute a posterior distribution over the atmospheric parameters of interest. Depending on the complexity of the simulator, the number of parameters, the spectral resolution of the observed data, and other factors, this can become very computationally expensive: A single AR can easily require on the order of tens of thousands of CPU hours, often resulting in wall times of days to weeks.

Reducing this computational burden has already attracted the attention of the machine learning (ML) community, including even a competition at NeurIPS 2022 (Changeat and Yip 2022). Previously proposed ML approaches to the problem of AR include the usage of GANs (Zingales and Waldmann 2018), random forests (Márquez-Neila et al. 2018; Fisher et al. 2020), Monte Carlo dropout (Soboczenski et al. 2018), Bayesian neural networks (Cobb et al. 2019), various deep learning architectures (Yip et al. 2021; Ardévol Martínez et al. 2022; Giobergia, Koudounas, and Baralis 2023; Unlu et al. 2023), variational inference (Yip et al. 2022), and neural posterior estimation (NPE) using discrete normalizing flows (Vasist et al. 2023). NPE was also used by the winning entry to the 2023 edition of the ARIEL data challenge (Aubin et al. 2023). Finally, a related but somewhat orthogonal direction are the approaches by Himes et al. (2022) and Hendrix, Louca, and Miguel (2023), who do not predict a posterior directly, but instead speed up ARs by replacing the computationally expensive simulator with a learned emulator.

In this workshop paper, we first introduce another ML approach to AR: flow matching posterior estimation (FMPE) using continuous normalizing flows. Focusing on one specific case study from the literature, we then compare FMPE to both a nested sampling approach and neural posterior estimation (NPE) as introduced by Vasist et al. (2023). Finally, we combine both FMPE and NPE with neural importance sampling and show that this improves the results significantly.

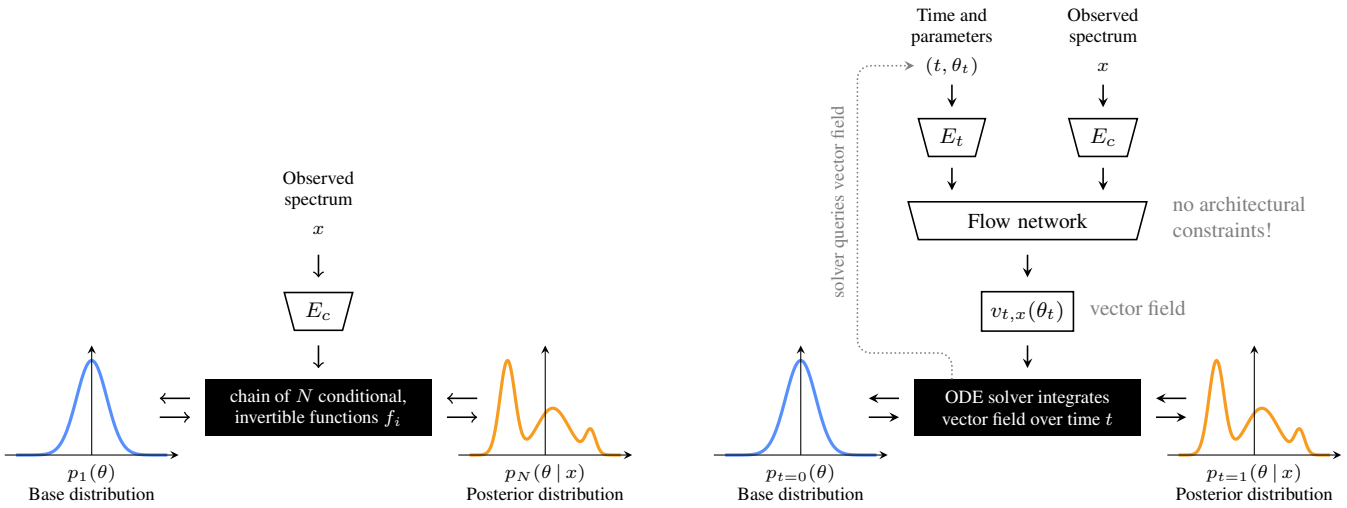
Method

We briefly recapitulate NPE, which will serve as another baseline besides nested sampling, and then introduce the FMPE method as well as the idea of neural importance sampling. A schematic comparison of NPE and FMPE is found in fig. 1.

NPE with normalizing flows NPE (Papamakarios and Murray 2016) is a technique for simulation-based inference (SBI; Cranmer, Brehmer, and Louppe 2020) that trains a density estimator $q(\theta \mid x)$ to approximate the posterior $p(\theta \mid x)$ by minimizing the following loss:

$$\mathcal{L}_{\text{NPE}} = -\mathbb{E}_{\theta\sim\pi(\theta)}\, \mathbb{E}_{x\sim p(x\mid\theta)} \log q(\theta \mid x)\,. \tag{1}$$

Here, $\pi(\theta)$ denotes the prior and sampling from the likelihood corresponds to a call of the forward simulator. Once trained, $q(\theta \mid x)$ serves as a surrogate for the posterior, enabling cheap sampling and density evaluation. The density $q(\theta \mid x)$ is often parameterized with a conditional discrete normalizing flow (DNF; Tabak and Vanden-Eijnden 2010; Rezende and



(a) NPE (b) FMPE

Figure 1: Schematic comparison between neural posterior estimation (NPE) and flow matching posterior estimation (FMPE).

Mohamed 2015). DNFs construct the distribution

$$q(\theta \mid x) = p_0\left(\psi_x^{-1}(\theta)\right) \cdot \det \left| \frac{\mathrm{d}\psi_x^{-1}(\theta)}{\mathrm{d}\theta} \right| \tag{2}$$

by applying a chain of invertible functions $\psi_x : \mathbb{R}^n \to \mathbb{R}^n$, $\psi_x = f_N \circ \cdots \circ f_2 \circ f_1$ to a simple base distribution p_0 (e.g., an n-dimensional Gaussian). To ensure invertibility and efficient computation of the Jacobian in eq. (2), most DNFs impose strong constraints on the architecture.

FMPE Continuous normalizing flows (CNF; Chen et al. 2018) also transform a simple base distribution to a more complex one, but describe this transformation continuously. Specifically, the sample trajectories $\psi_{t,x}$ are parameterized in terms of a "time" parameter $t \in [0, 1]$, and a vector field $v : [0, 1] \times \mathbb{R}^{m+n} \to \mathbb{R}^n$, where $m = \dim(x)$ and $n = \dim(\theta)$, defined by the ordinary differential equation (ODE):

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi_{t,x}(\theta) = v_{t,x}\left(\psi_{t,x}(\theta)\right) , \quad \psi_{0,x}(\theta) = \theta . \tag{3}$$

Conversion between the base ($t = 0$) and target ($t = 1$) distribution is then achieved by integration,

$$q_1(\theta \mid x) = q_0(\theta) \cdot \exp\left\{ - \int_0^1 \operatorname{div} v_{t,x}(\theta_t)\, \mathrm{d}t \right\} . \tag{4}$$

CNFs offer great flexibility, as they are parameterized by unconstrained vector fields and thus do not impose architectural constraints. However, likelihood maximization can be prohibitively expensive due to the cost of the ODE integration.

Flow matching (Lipman et al. 2022) provides an alternative training objective that directly regresses v onto a target vector field u by minimizing $\mathbb{E}\left[||v - u||^2\right]$. It has been shown that the target can be carefully designed as a sample-conditional vector field $u_t(\theta \mid \theta_{t=1})$, resulting in a tractable and efficient training objective. In the context of SBI, flow matching has been explored by Dax et al. (2023b), and the resulting method (flow matching posterior estimation; FMPE) maintains the desirable properties of NPE (expressiveness of the distribution, tractable density, simulation-based training) without requiring constrained neural architectures. In comparison to NPE, FMPE training is typically faster (due to simpler architectures) and inference is slower (due to the ODE integration).

Neural importance sampling In practice, both NPE and FMPE results may deviate from the exact posterior due to insufficient training data or network capacity, or when confronted with out-of-distribution (OOD) data. Further, it is typically difficult to assess whether an inferred posterior is accurate without comparing to results from another (trusted) inference method, which may not always be available: While nested sampling has been hailed as the gold standard for AR, some implementations have already been found to produce overly confident results (e.g., Ardévol Martínez et al. 2022).

One way to address these challenges is to combine SBI methods with likelihood-based importance sampling (Dax et al. 2023a). In this case, the inferred estimate $q(\theta \mid x)$ is used as a proposal distribution for importance sampling (IS; Kloek and van Dijk 1978) by attaching importance weights

$$w_i = p(\theta_i \mid x) \cdot p(\theta_i) / q(\theta_i \mid x) \tag{5}$$

to each sample $\theta_i \sim q(\theta \mid x)$. This transforms N samples from $q(\theta \mid x)$ into weighted samples from the true posterior $p(\theta \mid x)$. Dax et al. (2023a) showed that this results in asymptotic recovery of the exact posterior, and that failures (e.g., due to OOD data) are marked by a low sampling efficiency

$$\epsilon = \frac{1}{N} \cdot \left(\sum_{i=1}^{N} w_i \right)^2 \Bigg/ \sum_{i=1}^{N} w_i^2 , \quad \epsilon \in [0, 1] . \tag{6}$$

The sampling efficiency also provides a direct performance measure: the better $q(\theta \mid x)$ matches $p(\theta \mid x)$, the higher the sampling efficiency. In practice, however, ϵ is susceptible to slight mismatches in even just one dimension of θ, resulting in a high variance of the w_i and thus low ϵ.

Experiments and results

We empirically evaluate both FMPE and NPE on the benchmark retrieval case from Vasist et al. (2023), which is based on a study of the planet HR 8799 e by Mollière et al. (2020).

Simulator We use the simulation code from Vasist et al. (2023), which itself is based on the `petitRADTRANS` simulator (Mollière et al. 2019; we used v2.6.7). This maps a $\dim(\theta) = 16$ dimensional parameter space (see table 3 for descriptions and priors) to simulated emission spectra for a gas giant-type planet (cf. fig. 3). We work at a spectral resolution of $R = \Delta\lambda/\lambda = 1000$ (compared to $R = 400$ in Vasist et al. 2023) with a wavelength range of $0.95\,\mu\text{m}$–$2.45\,\mu\text{m}$, corresponding to $\dim(x) = 947$ bins. Following Vasist et al. (2023), we apply independent Gaussian noise with $\mu = 0, \sigma = 0.1257$ for each bin of the spectrum.

Nested sampling baseline We use `nautilus` (Lange 2023) as a baseline, which implements importance nested sampling enhanced with deep learning. More conventional samplers such as `PyMultiNest` (Buchner et al. 2014) or `dynesty` (Speagle 2020) did not converge after several weeks. Using the Gaussian likelihood implied by our simulator, we run with $10\,000$ live points, $0.1\,\%$ remaining live points as convergence criterion, a target effective sample size of $50\,000$, and default values for all other settings.

NPE and FMPE models We train both an NPE and an FMPE model. The configurations reported here are the respective best ones from our preliminary experiments.

As illustrated in fig. 1, the NPE model consists of two parts: (1) a context embedding network E_c for the flux values x in the form of 15 residual blocks of decreasing size (from 4096 to 256) that outputs a representation $z \in \mathbb{R}^{256}$, and (2) a neural spline flow (Durkan et al. 2019) with 20 piecewise rational quadratic coupling transforms (hidden size 1024, 4 blocks, 16 bins) which are conditioned on z. For FMPE, the model has three parts: (1) a context embedding network E_c for the flux x with two residual blocks, (2) a residual network E_t with positional encodings applied to t, mapping t and θ_t to a 512-dimensional embedding, and (3) the flow network with 40 residual blocks of decreasing size (from 8192 to 16) that receives the embedded spectrum and (t, θ_t)-tuple and predicts the vector field $v_{t,x}(\theta_t)$. The total number of trainable parameters is 318 M for NPE and 501 M for FMPE.

We train our models for up to 1000 epochs on a dataset of 16.8 M simulations with batch size 16 384, using the AdamW optimizer (Loshchilov and Hutter 2017) with a ReduceLROnPlateau scheduler (initial learning rate 10^{-4}, patience 30 epochs, factor 0.5), early stopping (patience 100 epochs), and gradient clipping (L_2 norm ≤ 1.0). Both models use dropout; for FMPE, batch normalization also proved beneficial. We use a 98% / 2% split between training and validation, and only store the models that achieve the lowest validation loss. Following standard ML practices, the atmospheric parameters θ are standardized by subtracting the mean and dividing by the standard deviation. Like in Vasist et al. (2023), the flux values x are rescaled as $x \mapsto x/(1+|x/100|)$. For FMPE, we train with automatic mixed precision (AMP) as it speeds up training significantly, while for NPE, we find that AMP has almost no effect. On a single NVIDIA H100 GPU, training to convergence takes approximately 54 h for the FMPE model (747 epochs, 261 s per epoch), and about 148 h (1000 epochs, 533 s per epoch) for NPE.

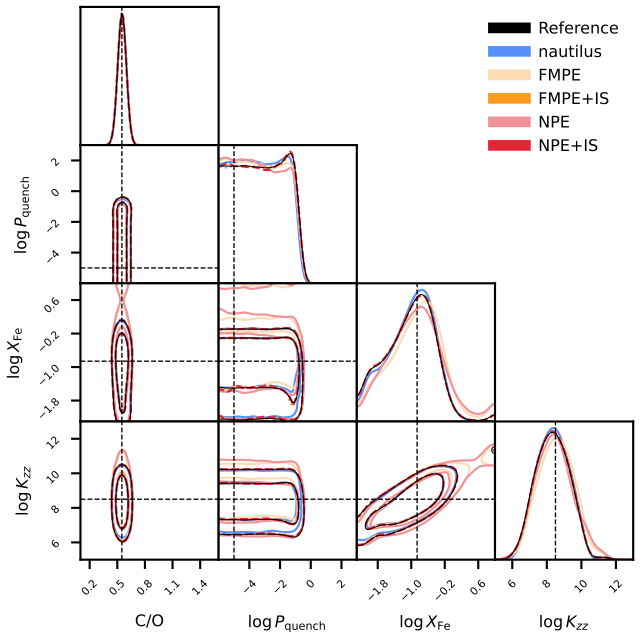


Figure 2: Comparison of the posterior estimates for four parameters; see fig. 4 in the Appendix for the full version.

Reference posterior Thorough quantitative evaluation of inference results requires comparison to reference posteriors. Even nested sampling may produce inaccurate results, so we here combine all three methods to produce a reference posterior with the approach proposed in Dax et al. (2023a). For our given benchmark spectrum, we first generate a large number of approximate posterior samples (7 M samples, equally distributed between `nautilus` / NPE / FMPE).[1] Then, we train an unconditional DNF $q(\theta)$ to estimate the distribution of these samples using a maximum log-likelihood objective. Finally, we generate weighted posterior samples with importance sampling, $\theta_i \sim q(\theta)$, $w_i = \pi(\theta_i) \cdot p(x \mid \theta_i)/q(\theta_i)$. With this method, we generate $n_{\text{eff}} = 616\,\text{k}$ samples ($\epsilon = 6.2\,\%$), which represent our reference posterior. Importance sampling is asymptotically exact if the proposal covers the entire target. For us, this is the case if the initial 7 M samples cover the posterior support because density recovery with the DNF is performed using a probability mass covering training objective; see the discussion in Dax et al. (2023a) for details.

Evaluation We now compare all methods: `nautilus`, FMPE, FMPE-IS (i.e., FMPE augmented with importance sampling), NPE and NPE-IS. For each method, we generate 50 k effective samples. With importance sampling, we find efficiencies of $\epsilon = 13.0\,\%$ for FMPE-IS and $\epsilon = 2.5\,\%$ for NPE-IS. Qualitative results are shown in fig. 2 for se-

[1] Generation with NPE / FMPE is cheap; for `nautilus`, we use the (correlated) intermediate samples to save computational cost.

Table 1: JS divergence (in mnat) between the marginals of our reference posterior and the different methods (lower is better).

Method	C/O	[Fe/H]	$\log P_{\text{quench}}$	$\log X_{\text{Fe}}$	$\log X_{\text{MgSiO}_3}$	f_{sed}	$\log K_{zz}$	σ_g	$\log g$	R_P	T_0	$\frac{T_3}{T_{\text{connect}}}$	T_2/T_3	T_1/T_2	α	$\frac{\log \delta}{\alpha}$	Mean
nautilus	1.7	1.5	37.9	53.3	20.3	20.8	56.7	9.6	7.1	10.5	12.5	4.3	**5.6**	4.3	7.3	9.1	16.4
FMPE	6.6	11.1	20.0	107.1	106.1	69.2	86.0	17.9	34.4	17.1	44.3	21.6	15.4	19.3	51.2	46.8	42.1
FMPE+IS	0.6	**0.4**	**3.2**	6.7	**4.9**	**4.3**	8.1	**4.6**	**0.7**	3.7	1.9	**3.7**	5.6	**3.5**	5.4	2.1	3.7
NPE	14.8	29.1	22.9	114.7	110.7	160.0	83.4	13.3	28.6	25.4	65.1	33.2	16.1	11.1	72.5	59.0	53.8
NPE+IS	**0.2**	0.4	4.1	**4.5**	8.7	4.4	**7.0**	8.0	1.6	**1.6**	**1.6**	5.6	6.3	5.7	**4.8**	**1.8**	4.1

lected and in fig. 4 for all parameters. For a quantitative evaluation, we compare each result to the reference in terms of the Jensen-Shannon divergence (JSD) between the 1D marginal distributions (table 1). As an additional accuracy measure, we report upper bounds on the linear optimal transport distance in the Appendix (fig. 5), which also captures high-dimensional distributional differences.

First, we see that nested sampling clearly outperforms standard FMPE and NPE in terms of accuracy. However, when FMPE and NPE are augmented with IS, their performance improves by an order of magnitude and their accuracies even exceed the `nautilus` baseline. The deviations between FMPE-IS, NPE-IS and the reference are negligibly small, which is expected as all three of these estimates are asymptotically exact. In practice, we expect that even the deviation of nested sampling from the reference will be scientifically irrelevant, and that the main advantage of FMPE-IS and NPE-IS is not improved accuracy, but reduced computational cost, especially in an amortized setting (see below). Further, we note that FMPE consistently produces slightly more accurate results than NPE. Future work needs to investigate if this only holds for our specific case, or if this is a general trend. Lastly, we observe that both FMPE and NPE struggle with similar parameters (e.g., $\log X_{\text{Fe}}$), which could indicate that the main challenge for the model lies in extraction of the relevant information from the spectrum, and not in the density estimation.

Runtime considerations Besides the training time (2 days for FMPE, 6 days for NPE), the total runtime of our methods is additionally composed of the time to generate the dataset and to do inference. On a single core of an AMD EPYC 7662 CPU, simulating one spectrum (with random parameters drawn from the prior) takes about $3.2 \pm 0.7\,\text{s}$, implying a total of about 15 000 CPU hours that can be arbitrarily parallelized. For example, with 16 AMD EPYC 7662 CPUs (with 64 cores each), simulating our training set would take about $15\,\text{h}$.

At inference, sampling and evaluating the log-probabilities from the trained model is almost negligible in terms of computational cost compared to the `nautilus` baseline: On a single GPU, this takes about $12\,\text{s}$ for NPE, and about $12\,\text{min}$ for FMPE (using a tolerance of 10^{-3} for the ODE solver). However, this is also arbitrarily parallelizable. For IS, we additionally have to consider the cost for simulating spectra: Assuming a sampling efficiency of $\epsilon = 5\%$, generating $50\,\text{k}$ effective samples requires another 900 single core hours, or less than one hour when assuming 16 CPUs. This means that if we can make proper use of the parallelization capabilities of NPE and FMPE, it seems plausible that we can beat the wall time of nested sampling even in a non-amortized setting (i.e., when running only a single retrieval), which in our case was 8.5 days for `nautilus`, and would be much higher for traditional samplers. Of course, this computational advantage becomes much more significant once we consider multiple retrievals. Table 2 summarizes the expected inference costs based on our experiments.

Table 2: Comparison of the computational costs and sampling efficiencies at inference time. This does not include the time required for generating data and training models.

	# simulations	efficiency	CPU hours	wall time
nautilus	4 462 500	1.12 %	13 450	8.5 d
FMPE	—	—	—[1]	12 min
FMPE+IS	384 691	13.00 %	342	3.5 h[2]
NPE	—	—	—[1]	12 s
NPE+IS	1 999 354	2.50 %	1777	18.5 h[2]

[1] Sampling standard FMPE / NPE uses a GPU. [2] Assuming 96 CPUs (which is what we used for running `nautilus`); in practice, this is arbitrarily parallelizable.

Conclusions

We compared flow matching posterior estimation (FMPE), a new approach to exoplanet atmospheric retrieval based on CNFs, with both neural posterior estimation (NPE) using DNFs and (ML-enhanced) nested sampling as implemented by `nautilus`. Both FMPE and NPE yielded good agreement with the reference posterior, while reducing inference times by orders of magnitude compared to nested sampling. Notably, FMPE demonstrated slightly higher accuracy and significantly shorter training times than NPE. Combining both approaches with neural importance sampling, we matched the accuracy of an established nested sampling algorithm, `nautilus`, while retaining a runtime advantage, in particular assuming an amortized setting. One limitation of this ongoing work is that we have only considered a single benchmark retrieval. In future work, we will study the properties of FMPE (with and without importance sampling) more systematically. Our results encourage a broader adoption of SBI approaches for AR to combine high accuracy and diminishing retrieval costs not only in the analysis of real observational data (e.g., from JWST), but also during the design phase of new instruments and missions for exoplanet science, such as HWO (Clery 2023) or LIFE (Quanz et al. 2022).

References

Ardévol Martínez, F.; Min, M.; Kamp, I.; and Palmer, P. I. 2022. Convolutional neural networks as an alternative to Bayesian retrievals for interpreting exoplanet transmission spectra. *A&A*, 662: A108.

Ashton, G.; Bernstein, N.; Buchner, J.; et al. 2022. Nested sampling for physical scientists. *Nature Reviews Methods Primers*, 2(1).

Aubin, M.; Cuesta-Lazaro, C.; Tregidga, E.; et al. 2023. Simulation-based Inference for Exoplanet Atmospheric Retrieval: Insights from winning the Ariel Data Challenge 2023 using Normalizing Flows. *arXiv preprints*.

Buchner, J.; Georgakakis, A.; Nandra, K.; et al. 2014. X-ray spectral modelling of the AGN obscuring region in the CDFS: Bayesian model selection and catalogue. *A&A*, 564: A125.

Changeat, Q.; and Yip, K. H. 2022. ESA-Ariel Data Challenge NeurIPS 2022: Introduction to exo-atmospheric studies and presentation of the Atmospheric Big Challenge (ABC) Database. *arXiv preprints*.

Chen, R. T. Q.; Rubanova, Y.; Bettencourt, J.; et al. 2018. Neural Ordinary Differential Equations. *arXiv preprints*.

Clery, D. 2023. Future NASA scope would find life on alien worlds. *Science*, 379(6628): 123–124.

Cobb, A. D.; Himes, M. D.; Soboczenski, F.; Zorzan, S.; O'Beirne, M. D.; Baydin, A. G.; Gal, Y.; Domagal-Goldman, S. D.; Arney, G. N.; and and, D. A. 2019. An Ensemble of Bayesian Neural Networks for Exoplanetary Atmospheric Retrieval. *AJ*, 158(1): 33.

Cranmer, K.; Brehmer, J.; and Louppe, G. 2020. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48): 30055–30062.

Dax, M.; Green, S. R.; Gair, J.; et al. 2023a. Neural Importance Sampling for Rapid and Reliable Gravitational-Wave Inference. *Phys. Rev. Lett.*, 130(17).

Dax, M.; Wildberger, J.; Buchholz, S.; Green, S. R.; Macke, J. H.; and Schölkopf, B. 2023b. Flow Matching for Scalable Simulation-Based Inference. *arXiv preprints*.

Durkan, C.; Bekasov, A.; Murray, I.; and Papamakarios, G. 2019. Neural Spline Flows. *arXiv preprints*.

Fisher, C.; Hoeijmakers, H. J.; Kitzmann, D.; Márquez-Neila, P.; Grimm, S. L.; Sznitman, R.; and Heng, K. 2020. Interpreting High-resolution Spectroscopy of Exoplanets using Cross-correlations and Supervised Machine Learning. *AJ*, 159(5): 192.

Giobergia, F.; Koudounas, A.; and Baralis, E. 2023. Reconstructing Atmospheric Parameters of Exoplanets Using Deep Learning. *arXiv preprints*.

Guillot, T. 2010. On the radiative equilibrium of irradiated planetary atmospheres. *Astronomy and Astrophysics*, 520: A27.

Hendrix, J. L. A. M.; Louca, A. J.; and Miguel, Y. 2023. Using a neural network approach to accelerate disequilibrium chemistry calculations in exoplanet atmospheres. *MNRAS*, 524(1): 643–655.

Himes, M. D.; Harrington, J.; Cobb, A. D.; Baydin, A. G.; Soboczenski, F.; O'Beirne, M. D.; Zorzan, S.; Wright, D. C.; Scheffer, Z.; Domagal-Goldman, S. D.; and Arney, G. N. 2022. Accurate Machine-learning Atmospheric Retrieval via a Neural-network Surrogate Model for Radiative Transfer. *The Planetary Science Journal*, 3(4): 91.

Kloek, T.; and van Dijk, H. K. 1978. Bayesian Estimates of Equation System Parameters: An Application of Integration by Monte Carlo. *Econometrica*, 46(1): 1.

Lange, J. U. 2023. nautilus: boosting Bayesian importance nested sampling with deep learning. *MNRAS*, 525(2): 3181–3194.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; et al. 2022. Flow Matching for Generative Modeling. *arXiv preprints*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. *arXiv preprints*.

Luscombe, R. 2023. Nasa says distant exoplanet could have rare water ocean and possible hint of life. *The Guardian*.

Madhusudhan, N. 2018. Atmospheric Retrieval of Exoplanets. In *Handbook of Exoplanets*, 1–30. Springer International Publishing.

Mollière, P.; Stolker, T.; Lacour, S.; Otten, G. P. P. L.; Shangguan, J.; Charnay, B.; Molyarova, T.; Nowak, M.; Henning, T.; Marleau, G.-D.; Semenov, D. A.; van Dishoeck, E.; Eisenhauer, F.; Garcia, P.; Garcia Lopez, R.; Girard, J. H.; Greenbaum, A. Z.; Hinkley, S.; Kervella, P.; Kreidberg, L.; Maire, A.-L.; Nasedkin, E.; Pueyo, L.; Snellen, I. A. G.; Vigan, A.; Wang, J.; de Zeeuw, P. T.; and Zurlo, A. 2020. Retrieving scattering clouds and disequilibrium chemistry in the atmosphere of HR 8799e. *A&A*, 640: A131.

Mollière, P.; Wardenier, J. P.; van Boekel, R.; Henning, T.; Molaverdikhani, K.; and Snellen, I. A. G. 2019. petitRADTRANS: a Python radiative transfer package for exoplanet characterization and retrieval. *A&A*, 627: A67.

Márquez-Neila, P.; Fisher, C.; Sznitman, R.; and Heng, K. 2018. Supervised machine learning for analysing spectra of exoplanetary atmospheres. *Nature Astronomy*, 2(9): 719–724.

Papamakarios, G.; and Murray, I. 2016. Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation. *arXiv preprints*.

Quanz, S. P.; Ottiger, M.; Fontanet, E.; et al. 2022. Large Interferometer For Exoplanets (LIFE): I. Improved exoplanet detection yield estimates for a large mid-infrared space-interferometer mission. *A&A*, 664: A21.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. *arXiv preprints*.

Skilling, J. 2006. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4).

Soboczenski, F.; Himes, M. D.; O'Beirne, M. D.; Zorzan, S.; Baydin, A. G.; Cobb, A. D.; Gal, Y.; Angerhausen, D.; Mascaro, M.; Arney, G. N.; and Domagal-Goldman, S. D. 2018. Bayesian Deep Learning for Exoplanet Atmospheric Retrieval. *arXiv preprints*.

Speagle, J. S. 2020. dynesty: a dynamic nested sampling package for estimating Bayesian posteriors and evidences. *MNRAS*, 493(3): 3132–3158.

Tabak, E. G.; and Vanden-Eijnden, E. 2010. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1): 217 – 233.

Unlu, E. B.; Forestano, R. T.; Matchev, K. T.; and Matcheva, K. 2023. Reproducing Bayesian Posterior Distributions for Exoplanet Atmospheric Parameter Retrievals with a Machine Learning Surrogate Model. *arXiv preprints*.

Vasist, M.; Rozet, F.; Absil, O.; Mollière, P.; Nasedkin, E.; and Louppe, G. 2023. Neural posterior estimation for exoplanetary atmospheric retrieval. *A&A*, 672: A147.

Yip, K. H.; Changeat, Q.; Al-Refaie, A.; and Waldmann, I. 2022. To Sample or Not To Sample: Retrieving Exoplanetary Spectra with Variational Inference and Normalising Flows. *arXiv preprints*.

Yip, K. H.; Changeat, Q.; Nikolaou, N.; Morvan, M.; Edwards, B.; Waldmann, I. P.; and Tinetti, G. 2021. Peeking inside the Black Box: Interpreting Deep-learning Models for Exoplanet Atmospheric Retrievals. *AJ*, 162(5): 195.

Zingales, T.; and Waldmann, I. P. 2018. ExoGAN: Retrieving Exoplanetary Atmospheres Using Deep Convolutional Generative Adversarial Networks. *AJ*, 156(6): 268.

Appendix

This appendix contains table 3 and figs. 3 to 5 that were referenced in the main text.

Table 3: The 16 atmospheric parameters of interest that we consider in this work, including the priors used for the data generation and the parameter values θ_0 of the spectrum used as the benchmark case. See also tables 1 and 2 in Vasist et al. (2023).

Parameter	**Prior**	θ_0 **value**	**Meaning**
$\mathrm{C/O}$	$\mathcal{U}(0.1, 1.6)$	0.55	Carbon-to-oxygen ratio
$[\mathrm{Fe/H}]$	$\mathcal{U}(-1.5, 1.5)$	0.00	Metallicity
$\log P_{\mathrm{quench}}$	$\mathcal{U}(-6.0, 3.0)$	-5.00	Pressure at which CO, CH_4 and H_2O abundances become vertically constant
$\log X_{\mathrm{Fe}}$	$\mathcal{U}(-2.3, 1.0)$	-0.86	Scaling factor for equilibrium cloud abundances (Fe)
$\log X_{\mathrm{MgSiO_3}}$	$\mathcal{U}(-2.3, 1.0)$	-0.65	Scaling factor for equilibrium cloud abundances ($MgSiO_3$)
f_{sed}	$\mathcal{U}(0.0, 10.0)$	3.00	Sedimentation parameter
$\log K_{zz}$	$\mathcal{U}(5.0, 13.0)$	8.50	Vertical mixing parameter
σ_g	$\mathcal{U}(1.05, 3.0)$	2.00	Width of cloud particle size distribution (log-normal)
$\log g$	$\mathcal{U}(2.0, 5.5)$	3.75	(Logarithm of) surface gravity
R_P	$\mathcal{U}(0.9, 2.0)$	1.00	Planet radius (in Jupiter radii)
T_0	$\mathcal{U}(300, 2300)$	1063.60	Interior temperature of the planet (in Kelvin)
T_3/T_{connect}	$\mathcal{U}(0.0, 1.0)$	0.26	
T_2/T_3	$\mathcal{U}(0.0, 1.0)$	0.29	Parameters that describe the pressure-temperature profile (i.e., temperature as a
T_1/T_2	$\mathcal{U}(0.0, 1.0)$	0.32	function of pressure). The forward simulator uses a spline-based version of the
α	$\mathcal{U}(1.0, 2.0)$	1.39	parameterization scheme proposed in Guillot (2010).
$\log \delta/\alpha$	$\mathcal{U}(0.0, 1.0)$	0.48	

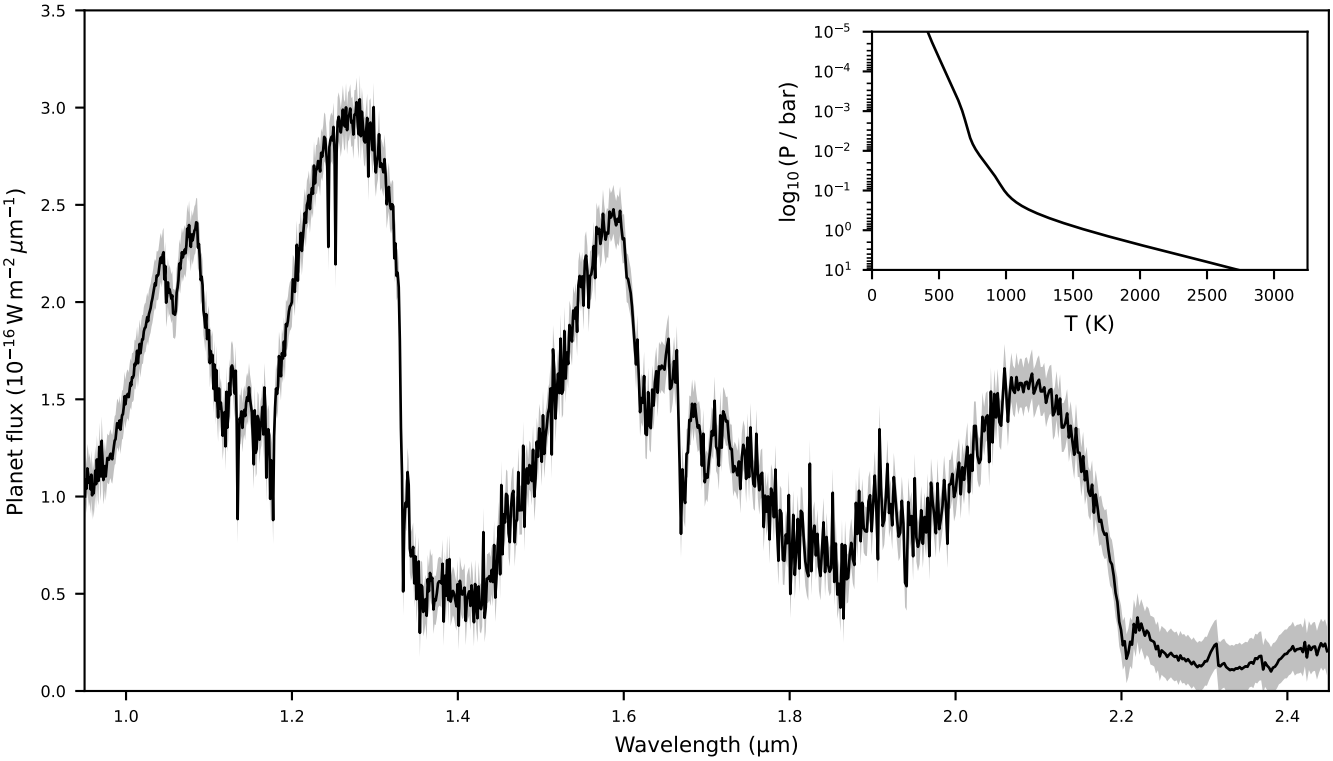


Figure 3: Simulated emission spectrum and pressure-temperature profile corresponding to θ_0 (i.e., our benchmark case). The shaded area marks the 1σ error bars we assumed in the likelihood, both for nested sampling and for the training data generation.

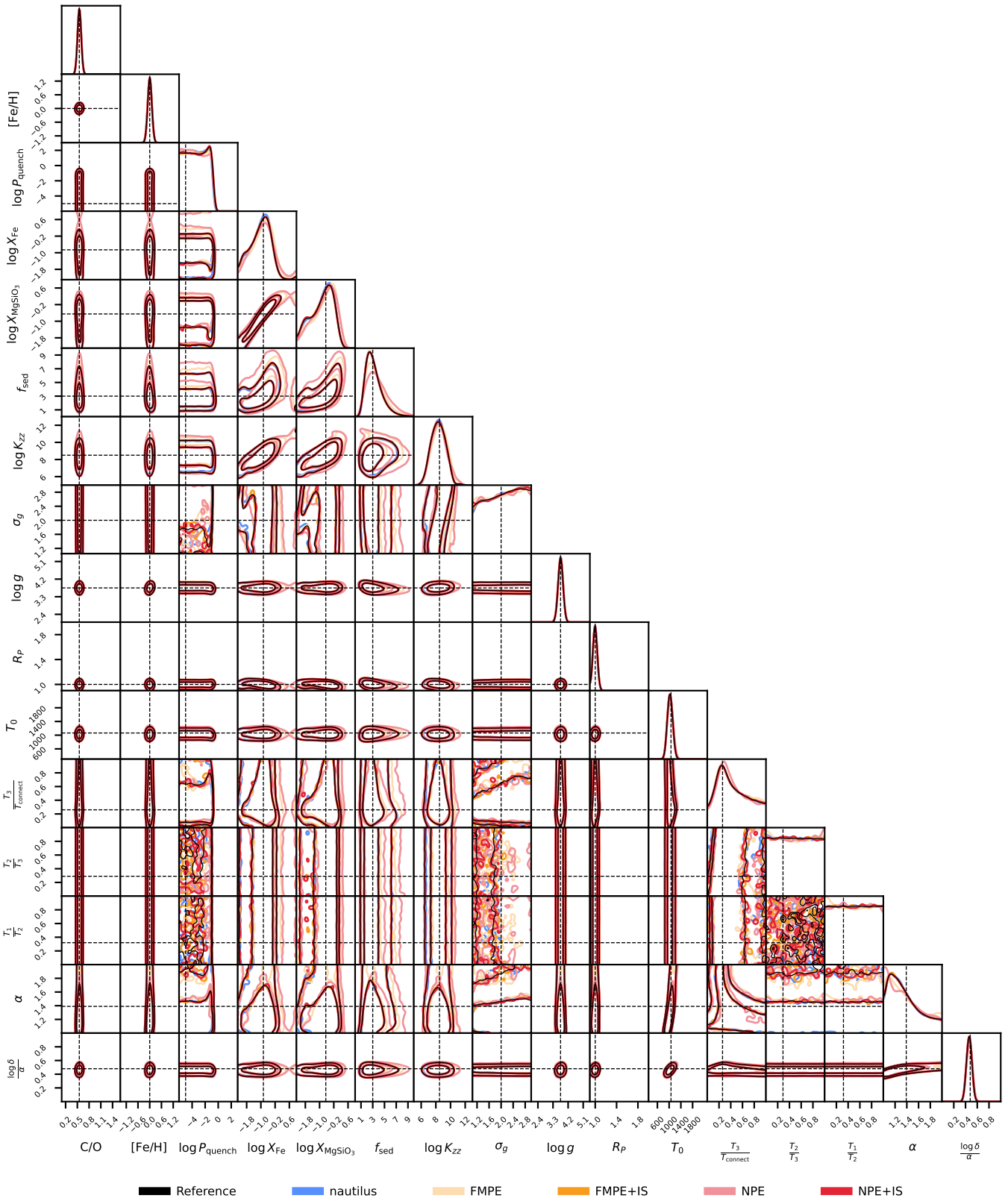


Figure 4: Results of our benchmark atmospheric retrieval: This corner plot shows a comparison of the 1D and 2D marginal posterior distributions for the different inference methods (`nautilus` is our nested sampling baseline). The true value θ_0 is marked by the dashed lines. The axes limits are set to the ranges of the respective priors. This figure is best viewed digitally.

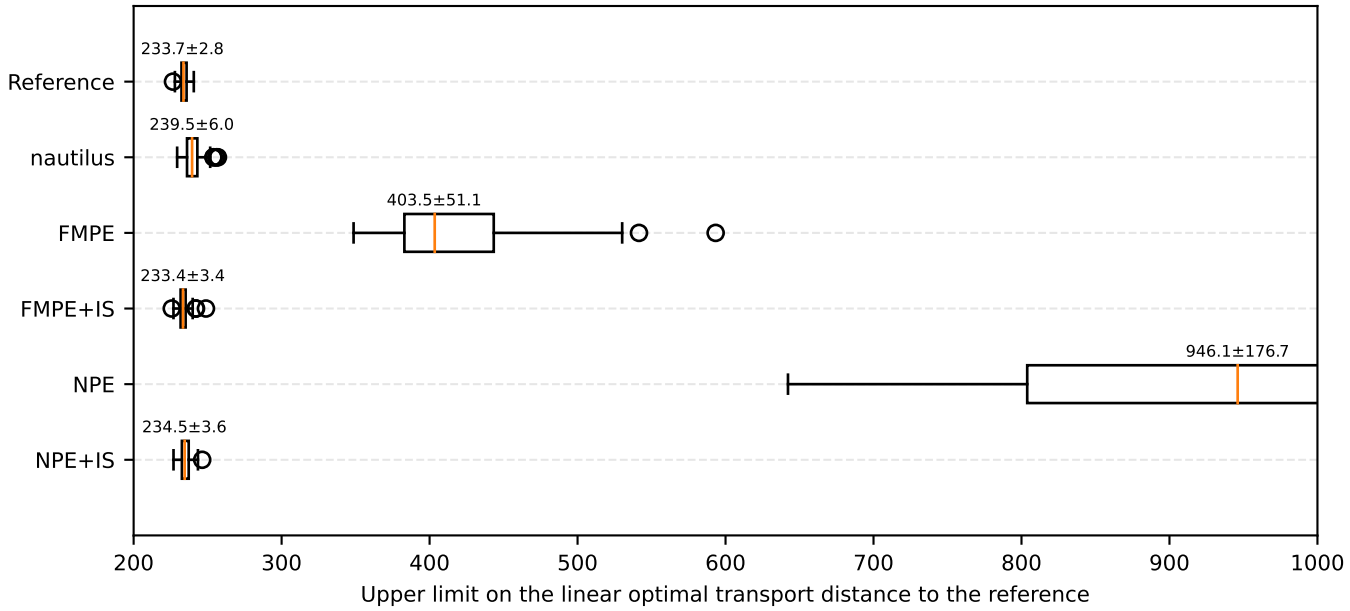


Figure 5: Upper bounds on the linear optimal transport distance between our reference posterior and the different estimators: We use the `ott-jax` package to estimate the linear optimal transport (OT) distance between the reference posterior and the different estimates (including the reference itself, to establish the general scale) by treating them as point clouds: For each method, we randomly choose 10 k posterior samples and compare them against an equal-sized random subset of the reference posterior by computing an upper bound on the OT distance using a Sinkhorn solver. We repeat this 100 times for each method and compute the median, which we take as an estimate of the distance to the reference. The resulting pattern matches the one from table 1.