








Exoplanet characterization using conditional invertible neural networks

Jonas Haldemann^{1,2}, Victor Ksoll³, Daniel Walter³, Yann Alibert¹, Ralf S. Klessen^{3,4}, Willy Benz¹,
Ullrich Koethe⁵, Lynton Ardizzone⁵, and Carsten Rother⁵

¹ Department of Space Research & Planetary Sciences, University of Bern, Gesellschaftsstrasse 6, 3012 Bern, Switzerland
e-mail: jonas.haldemann@unibe.ch

² Abteilung Physik, Gymnasium Lerbermatt, Kirchstrasse 64, 3098 Köniz, Switzerland

³ Universität Heidelberg, Zentrum für Astronomie, Institut für Theoretische Astrophysik, Albert-Ueberle-Straße 2, 69120 Heidelberg, Germany
e-mail: v.ksoll@uni-heidelberg.de

⁴ Universität Heidelberg, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany

⁵ Computer Vision and Learning Lab (HCI, IWR), Universität Heidelberg Berliner Str. 43, 69120 Heidelberg, Germany

Received 31 January 2022 / Accepted 20 February 2023

ABSTRACT

Context. The characterization of the interior of an exoplanet is an inverse problem. The solution requires statistical methods such as Bayesian inference. Current methods employ Markov chain Monte Carlo (MCMC) sampling to infer the posterior probability of the planetary structure parameters for a given exoplanet. These methods are time-consuming because they require the evaluation of a planetary structure model $\sim 10^5$ times.

Aims. To speed up the inference process when characterizing an exoplanet, we propose to use conditional invertible neural networks to calculate the posterior probability of the planetary structure parameters.

Methods. Conditional invertible neural networks (cINNs) are a special type of neural network that excels at solving inverse problems. We constructed a cINN following the framework for easily invertible architectures (FreIA). This neural network was then trained on a database of 5.6×10^6 internal structure models to recover the inverse mapping between internal structure parameters and observable features (i.e., planetary mass, planetary radius, and elemental composition of the host star). We also show how observational uncertainties can be accounted for.

Results. The cINN method was compared to a commonly used Metropolis-Hastings MCMC. To do this, we repeated the characterization of the exoplanet K2-111 b, using both the MCMC method and the trained cINN. We show that the inferred posterior probability distributions of the internal structure parameters from both methods are very similar; the largest differences are seen in the exoplanet water content. Thus, cINNs are a possible alternative to the standard time-consuming sampling methods. cINNs allow inferring the composition of an exoplanet that is orders of magnitude faster than what is possible using an MCMC method. The computation of a large database of internal structures to train the neural network is still required, however. Because this database is only computed once, we found that using an invertible neural network is more efficient than an MCMC when more than ten exoplanets are characterized using the same neural network.

Key words. planets and satellites: interiors – methods: numerical – methods: data analysis

1. Introduction

More than a decade ago, exoplanetary science has entered the era of characterization, where new observations are used to infer physical and chemical properties of exoplanets. These properties can be related to the atmosphere (e.g., [Hoeijmakers et al. 2019](#); [Madhusudhan 2019](#)) or to the planetary composition (e.g., [Dorn et al. 2015](#)). In the latter case, mass and radius measurements of an exoplanet are used to derive its internal structure (e.g., size of the iron core, presence of water, or gas mass fraction). This problem is notoriously strongly degenerate ([Rogers & Seager 2010](#)), but part of this degeneracy can be removed by assuming that the bulk refractory composition of an exoplanet matches the composition of its parent star ([Dorn et al. 2017b](#)). This assumption is supported by numerical simulations (e.g., [Thiabaud et al. 2015](#)) as well as Solar System observations (e.g., [Sotin et al. 2007](#)), although studies of observed exoplanets are not yet conclusive ([Plotnykov & Valencia 2020](#); [Schulze et al. 2021](#); [Adibekyan et al. 2021](#)).

Even under the assumption that the bulk compositions of the exoplanet and parent star match, the problem of deriving the planetary composition from the mass, radius, and refractory composition remains degenerate. The traditional method is to use Bayesian inference where the posterior probability of the planetary structure parameters is derived from the set of observed parameters, given prior probability distributions on the planetary structure parameters. These Bayesian calculations are in general performed using a Markov chain Monte Carlo (MCMC) method ([Mosegaard & Tarantola 1995](#); [Dorn et al. 2015](#); [Dorn et al. 2017b](#); Haldemann et al., in prep.)

Markov chain Monte Carlo methods are an efficient and well-tested way of sampling probability distributions. No analytical description of the whole normalized probability density function (PDF) of the target distribution is required. Instead, only the ratios of the PDF at pairs of locations in the phase space need to be calculated ([Hogg & Foreman-Mackey 2018](#)). This means that in the case of Bayesian inference, only the prior probability and the likelihood function need to be computed, and the

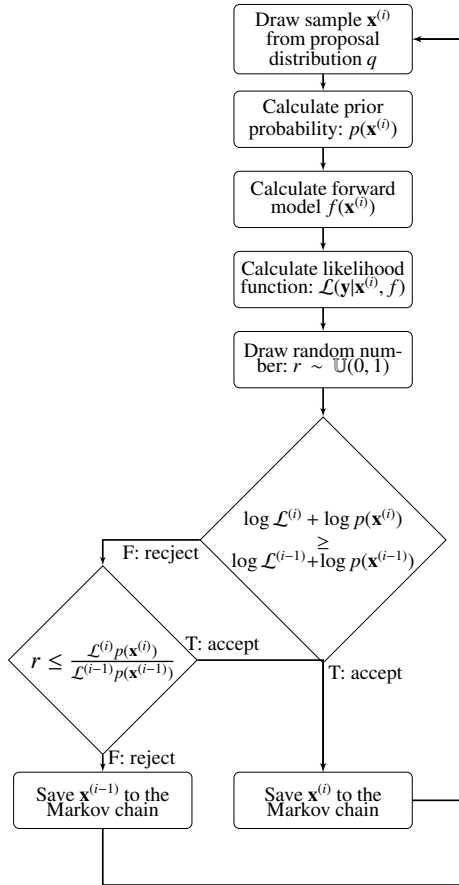


Fig. 1. Schematic overview of a Metropolis Hastings MCMC algorithm. The MCMC algorithm generates samples $\mathbf{x}^{(0)} \dots \mathbf{x}^{(n)} \sim \pi$, where the distribution π is proportional to the posterior probability $p(\mathbf{x}|\mathbf{y})$.

expensive calculation of Bayes' integral can be skipped, which acts as a normalization constant. A short conceptual summary of a common Metropolis Hastings MCMC method (Metropolis et al. 1953; Hastings 1970) is shown in Fig. 1. We refer to Trotta (2008) and Hogg & Foreman-Mackey (2018) for a more detailed introduction into MCMC methods and MCMC sampling.

In the past years, MCMC sampling together with a planetary structure model has led to a number of successful planetary characterizations (e.g., Dorn et al. 2017a; Agol et al. 2021), but using an MCMC suffers from at least two main difficulties. The first difficulty is that when planetary parameters are measured with small error bars, the likelihood function becomes very narrow. This is for instance the case for radius determination using high-precision telescopes such as CHEOPS (Benz et al. 2017, 2021) or PLATO in the future (Rauer & Heras 2018). Depending on the MCMC method that was used, small error bars can drastically increase the time required to converge to a solution. The second difficulty lies in the consideration of multiplanetary systems. In planetary systems, the observable data of the individual planets are often correlated. To benefit from these correlations, it is necessary to run an inference scheme for the whole system at once. When multiple exoplanets are characterized simultaneously, however, a planetary structure model needs to be calculated for every considered exoplanet. This increases the computational cost linearly. At the same time, the number of parameters that characterize the planetary compositions in a multiplanetary system is similarly scaled with the number of exoplanets. This increase in dimensionality of the parameter space

also generally implies an increase in the number of points needed to properly sample the posterior distribution, and therefore, an increase in the required computing time, which is more than linear overall.

In this paper, we propose a new method to derive the posterior distribution of planetary structure parameters. This method is based on conditional invertible neural networks (cINNs), which are a type of neural network architecture that is able to provide the posterior distribution of planetary structure parameters for any given choice of observed parameters (e.g., mass, radius, and refractory composition of the host star). The cINN has proven to be a robust method for calculating the posterior distribution in an inverse problem for any perfect observation, that is, without observational uncertainty, within the range of the training data (Ardizzone et al. 2019a,b). In this work, we expand the method to observations with observational uncertainties (see Sect. 2). Because the cINN was first proposed for cases without observational uncertainty, it is naturally suited for the case of high-precision measurements with very small uncertainties. Another key aspect is that once it is trained, the cINN provides the posterior distribution of planetary structure parameters in a few minutes, where modern MCMCs often require hours or even days to converge because the evaluation of the forward model is time-consuming (see Sect. 4).

In the past years, several authors used machine learning techniques to predict the output of a forward model (e.g., Alibert & Venturini 2019; Lin et al. 2022), or to solve inverse problems (de Wit et al. 2013; Atkins et al. 2016; Baumeister et al. 2020). The inverse problems were mostly modeled using mixture density neural networks (Bishop 1994), that is, a combination of a deep neural network with a probability mixture model. Likewise, cINNs have already seen successful applications in astronomy. Ksoll et al. (2020) were able to estimate stellar parameters from photometric observations of resolved star clusters using a cINN, and Kang et al. (2022) recently showed a cINN approach to recover physical parameters of star-forming clouds from spectral observations.

This paper is structured in the following way. In Sect. 2 we describe the basic concept of cINNs. We show how they can be set up in order to characterize exoplanets and how the forward model works that generates the training data for the neural networks. In Sect. 3, we first validate our proposed method using a simple toy model. In Sect. 4, we then apply the proposed method to characterize an observed exoplanet and compare its performance to a regular Metropolis-Hastings MCMC that was previously used for the same purpose. In Sect. 5, we discuss the current limitations of the approach and compare the time required to run either an MCMC or use the proposed method for cINNs. Finally, we summarize our findings in Sect. 6.

2. Methods

2.1. Invertible neural networks

The invertible neural network (INN) provides an architecture that excels in solving inverse problems (Ardizzone et al. 2019a). In these problems, access to a well-understood forward model is often given (e.g., a simulation) that describes the mapping between underlying physical parameters \mathbf{x} of an object and their corresponding observable quantities \mathbf{y} , for instance. At the same time, however, recovering the inverse mapping $\mathbf{y} \rightarrow \mathbf{x}$, which is of central interest in many applications, is a difficult task. The INN approach to these inverse problems makes the additional assumption that the known forward model always induces some

form of information loss in the mapping of $\mathbf{x} \rightarrow \mathbf{y}$, which can be encoded in some unobservable, latent variables \mathbf{z} . Leveraging a fully invertible architecture the INN is then trained to approximate the known forward model f , learning to associate \mathbf{x} values with unique pairs of $[\mathbf{y}, \mathbf{z}]$, that is, a bijective mapping. In doing so, it automatically provides a solution for the inverse mapping f^{-1} for free. For simplicity (as described in [Ardizzone et al. 2019a](#)), it is further assumed that the latent variables \mathbf{z} follow a Gaussian prior distribution, which is enforced during the training process. In principle, however, any desired distribution can be prescribed for the latent priors.

Given a new observation \mathbf{y} , this procedure allows us to predict the full posterior distribution $p(\mathbf{x}|\mathbf{y})$ by simply sampling from the known prior distribution of the latent space. The architecture of the INN consists of a series of reversible blocks, so-called affine coupling blocks, following a design proposed by [Dinh et al. \(2016\)](#). After splitting the input vector \mathbf{u} into two halves \mathbf{u}_1 and \mathbf{u}_2 , these blocks perform two complementary affine transformations,

$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp(s_2(\mathbf{u}_2)) + t_2(\mathbf{u}_2), \quad (1)$$

$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp(s_1(\mathbf{v}_1)) + t_1(\mathbf{v}_1), \quad (2)$$

using element-wise multiplication \odot and addition. Here, s_i and t_i denote arbitrarily complex mappings of \mathbf{u}_2 and \mathbf{v}_1 , for example, like small fully connected networks, which are not required to be invertible as they are only ever evaluated in the forward direction.

Inverting these affine transformations is trivial following

$$\mathbf{u}_2 = (\mathbf{v}_2 - t_1(\mathbf{v}_1)) \odot \exp(-s_1(\mathbf{v}_1)), \quad (3)$$

$$\mathbf{u}_1 = (\mathbf{v}_1 - t_2(\mathbf{u}_2)) \odot \exp(-s_2(\mathbf{u}_2)). \quad (4)$$

2.1.1. Conditional invertible neural networks

We employ a modification to this approach called conditional invertible neural network (cINN), as proposed in [Ardizzone et al. \(2019b\)](#) and as previously applied in [Ksoll et al. \(2020\)](#). Here, the affine coupling blocks are adapted to accept a conditioning input \mathbf{c} such that the mappings in Eqs. (1)–(4), that is, $s_2(\mathbf{u}_2)$, $t_2(\mathbf{u}_2)$, and so on, are replaced with $s_2([\mathbf{u}_2, \mathbf{c}])$ and $t_2([\mathbf{u}_2, \mathbf{c}])$, respectively. By concatenating conditions to the inputs of the subnetworks like this, the invertibility of the architecture is not affected. The forward $f(\mathbf{x}; \mathbf{c}) = \mathbf{z}$ and backward mapping $\mathbf{x} = g(\mathbf{z}; \mathbf{c})$ of the cINN both entail this conditioning, and the invertibility of the network is given for the fixed condition \mathbf{c} as $f(\cdot; \mathbf{c})^{-1} = g(\cdot; \mathbf{c})$.

When using cINNs for inverse regression problems, such as characterizing the internal structure of an exoplanet, the observations \mathbf{y} (e.g., planetary mass, radius, and stellar refractory composition) serve as the conditioning input. Figure 2 shows a schematic representation of the cINN in this case. In doing so, the cINN, just like the INN, will learn to encode all the variance of the physical parameters \mathbf{x} that is not explained by the observations \mathbf{y} into the latent variables \mathbf{z} during training. In addition to usually delivering better results, the cINN approach has the additional advantage that no zero padding is needed if the dimensions of $[\mathbf{y}, \mathbf{z}]$ and \mathbf{x} do not match, as we can simply set $\dim(\mathbf{z}) = \dim(\mathbf{x})$, ([Ardizzone et al. 2019a,b](#)). Zero padding refers to a procedure in which a zero vector of dimension $k = |m - n|$ is appended to either the input \mathbf{x} or output $[\mathbf{y}, \mathbf{z}]$ if their dimensions $\dim(\mathbf{x}) = m$ and $\dim([\mathbf{y}, \mathbf{z}]) = n$ do not match, which is a requirement in the standard INN approach. The zero padding can also be used to embed the entire problem into a higher dimensional

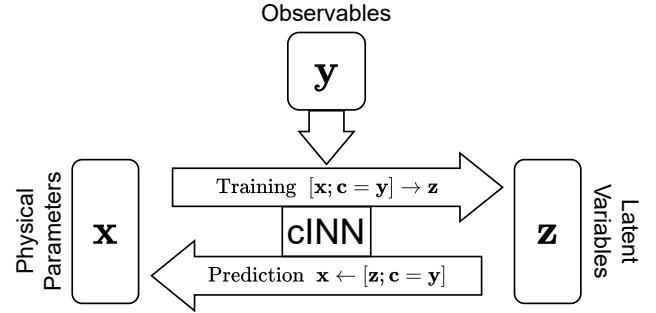


Fig. 2. Schematic overview of the cINN. During training, the cINN learns to encode all information about the physical parameters \mathbf{x} in the latent variables \mathbf{z} (while enforcing that they follow a Gaussian distribution) that is not contained in the observations \mathbf{y} . At prediction time, conditioned on the new observation \mathbf{y} , the cINN then transforms the known prior distribution $p(\mathbf{z})$ to \mathbf{x} -space to retrieve the posterior distribution $p(\mathbf{x}|\mathbf{y})$.

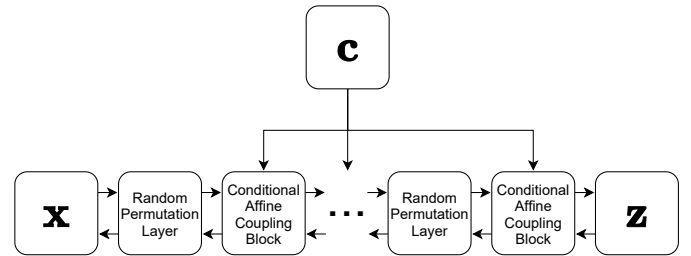


Fig. 3. Schematic overview over the cINN architecture.

space. The latter, however, introduces further hyperparameters and, thus, complicates the training procedure.

Given the condition \mathbf{c} of a new observation \mathbf{y} , the posterior distribution of the physical parameters is, as for the INN, determined by sampling the latent variables \mathbf{z} from their Gaussian prior,

$$p(\mathbf{x}|\mathbf{y}) \sim g(\mathbf{z}; \mathbf{c} = \mathbf{y}) \quad \text{with} \quad \mathbf{z} \sim p_z(\mathbf{z}) = \mathbb{N}(\mathbf{z}, \mathbf{0}, \mathbf{I}), \quad (5)$$

where \mathbf{I} is the $K \times K$ unity matrix with $K = \dim(\mathbf{z})$. In this framework, prior information on \mathbf{x} is learned by the network from the distribution of \mathbf{x} in the set of training data. This means that the distribution of the training data should follow the prior probability distribution $p(\mathbf{x})$.

2.1.2. Network architecture

For the work presented in this paper, we employed the framework for easily invertible architectures (FrEIA; [Ardizzone et al. 2019a,b](#)) and mostly followed the specific cINN architecture suggested in [Ardizzone et al. \(2019b\)](#). This means that we alternated the reversible affine coupling blocks in the generative flow (GLOW; [Kingma & Dhariwal 2018](#)) configuration with random permutation layers (see Fig. 3 for a schematic of the structure). The former is a computationally efficient variant, where the outputs of the mappings $s_i(\cdot)$ and $t_i(\cdot)$ are predicted jointly by a single subnetwork instead of one each. We used simple three-layer (width of 512 per layer) fully connected networks with rectified linear unit (ReLU) activation functions for these subnetworks. The random permutation layers use random orthogonal matrices that were fixed during training and are cheaply invertible, to better mix the information between the streams \mathbf{u}_1 and \mathbf{u}_2 . Together with the structure of the affine transformations,

this ensured that the cINN cannot just ignore the conditioning input during training. In our final architecture, we employed eight reversible blocks (an optimal value that we determined through extensive hyperparameter search for the given problem). In contrast to [Ardizzone et al. \(2019b\)](#), we did not apply a feature extraction network that transforms the input conditions into an intermediate representation because of the low dimensionality of the observable parameter space in our problem. Some early experiments have shown that a network like this did not benefit the predictive performance of the cINN on the given regression task. We trained the cINN by minimizing the maximum likelihood loss \mathcal{L} ,

$$\mathcal{L} = \mathbb{E}_i \left[\frac{\|f(\mathbf{x}_i; \mathbf{c}_i, \theta)\|_2^2}{2} - \log |J_i| \right], \quad (6)$$

where \mathbf{x}_i , \mathbf{c}_i denote the parameter-condition pair of training sample i , θ refers to the network weights, and $J_i = \det(\partial f / \partial \mathbf{x}_i)$ is the determinant of the Jacobian evaluated at training example i . To update the weights during training, we used the widely used adaptive moment estimation (ADAM) optimizer. For further details, we refer to [Ardizzone et al. \(2019b\)](#).

2.2. Forward model

The forward model f maps the physical input parameters \mathbf{x} to a prediction in the data space \mathbf{y}' , that is,

$$f(\mathbf{x}) = \mathbf{y}'. \quad (7)$$

It does so by calculating the interior structure of a 1D spherically symmetric sphere in hydrostatic equilibrium. Following [Kippenhahn et al. \(2012\)](#) and similar to the case of stellar structures, we solved the two point boundary value problem given by the equations

$$\frac{\partial r}{\partial m} = \frac{1}{4\pi r^2 \rho}, \quad (8)$$

$$\frac{\partial P}{\partial m} = -\frac{Gm}{4\pi r^4}, \quad (9)$$

$$\frac{\partial T}{\partial m} = \frac{\partial P}{\partial m} \frac{T}{P} \nabla, \quad (10)$$

where r is the radius, m is the mass within the radius r , P is the pressure, T is the temperature, ρ is the density, G is the gravitational constant, and ∇ is the dimensionless temperature gradient. The sphere was split into three layers of distinct composition similar to a differentiated planet (see Fig. 4). The thermodynamic properties of each layer are given by the set of equations of state (EoS) listed in Table 1.

From the EoS, we calculated ρ , the thermal expansion coefficient α , and the specific heat capacity c_P . We assumed that each layer is in a regime of vigorous convection. Therefore, the dimensionless temperature gradient is given by the adiabatic temperature gradient

$$\nabla = \left(\frac{\partial \ln T}{\partial \ln P} \right)_S = \frac{\alpha P}{\rho c_P}. \quad (11)$$

2.2.1. Core

We considered a solid iron core made out of hcp-Fe with possible inclusions of less dense FeS alloys. In the model, the composition of the core is given by the sulfur fraction $x_{\text{S}|_{\text{Core}}}$, that is,

$$x_{\text{Fe}|_{\text{Core}}} = \frac{1 - 2 x_{\text{S}|_{\text{Core}}}}{1 - x_{\text{S}|_{\text{Core}}}}, \quad (12)$$

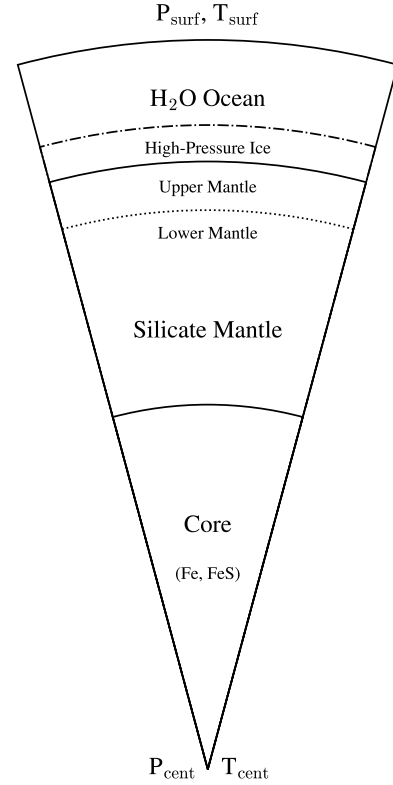


Fig. 4. Schematic representation of the layered planetary structure. Three main layers are present: the core, silicate mantle, and volatile layer. Depending on the size of the layers, an upper mantle can be present if the volatile layer above is not too massive. Conversely, if the volatile layer is massive enough, high-pressure ices might form on the bottom of the layer.

Table 1. List of EoSs used in the forward model.

Layer	Composition	EoS
Core	Fe, S	Hakim et al. (2018) , Fei et al. (2016) ^(a)
Mantle	Fe, Mg, Si, O	Sotin et al. (2007)
Volatile	H ₂ O	Haldemann et al. (2020)

Notes. ^(a)Only for pressures below 310 GPa.

$$x_{\text{FeS}|_{\text{Core}}} = \frac{x_{\text{S}|_{\text{Core}}}}{1 - x_{\text{S}|_{\text{Core}}}}. \quad (13)$$

The thermodynamic properties of Fe and FeS within the core were calculated using the EoS of [Hakim et al. \(2018\)](#). However, for Fe at pressures below 310 GPa, the EoS of [Fei et al. \(2016\)](#) was used, as advised by [Hakim et al. \(2018\)](#).

2.2.2. Mantle

The mantle structure was calculated following the model used in [Sotin et al. \(2007\)](#). It assumes a homogeneous elemental composition of Fe, Mg, Si, and O throughout the mantle, considering four different minerals. In the upper mantle, the model includes the iron and magnesium end members of the minerals olivine ([Mg,Fe]₂SiO₄) and ortho pyroxene ([Mg,Fe]₂Si₂O₆), while for the lower mantle, a composition of perovskite ([Mg,Fe]SiO₃) and wüstite ([Mg,Fe]O) is assumed. The transition between upper

and lower mantle was calculated as in [Sotin et al. \(2007\)](#),

$$P_{\text{transition}} = 25 \text{ GPa} - 0.0017 \frac{\text{GPa}}{\text{K}} \cdot (T - 800 \text{ K}). \quad (14)$$

The respective fractions of the upper and lower mantle mineral phases were calculated from the ratios of the $x_{\text{Mg}}/x_{\text{Si}}|_{\text{Mantle}}$ and $x_{\text{Fe}}/x_{\text{Si}}|_{\text{Mantle}}$ mole fractions as in [Sotin et al. \(2007\)](#), where

$$\frac{x_{\text{Mg}}}{x_{\text{Si}}}\Big|_{\text{Mantle}} = \frac{x_{\text{MgO}}}{x_{\text{SiO}_2}}\Big|_{\text{Mantle}} \quad (15)$$

and

$$\frac{x_{\text{Fe}}}{x_{\text{Si}}}\Big|_{\text{Mantle}} = \frac{x_{\text{FeO}}}{x_{\text{SiO}_2}}\Big|_{\text{Mantle}}. \quad (16)$$

Because a homogeneous elemental composition is assumed and because of the choice of minerals in the model of [Sotin et al. \(2007\)](#), the $x_{\text{Mg}}/x_{\text{Si}}|_{\text{Mantle}}$ and $x_{\text{Fe}}/x_{\text{Si}}|_{\text{Mantle}}$ ratios are limited by the possible spread in these minerals. Thus, only compositions that fulfil the relation

$$1 \leq \frac{x_{\text{Mg}}}{x_{\text{Si}}}\Big|_{\text{Mantle}} + \frac{x_{\text{Fe}}}{x_{\text{Si}}}\Big|_{\text{Mantle}} \leq 2 \quad (17)$$

can be calculated with this model. This ultimately also limits the possible Mg to Fe and Si to Fe ratios of the whole exoplanet. The resulting limits are further discussed in Sect. 2.3.

2.2.3. Volatiles

The outermost volatile layer was assumed to be entirely made up of H₂O. We forwent including an additional H/He layer in order to reduce the number of model parameters and hence the time needed to calculate the database of forward models used to train the cINN. More realistic volatile layers are planned to be added in the future, however. The EoS of H₂O is given by the AQUA-EoS of [Haldemann et al. \(2020\)](#), which combines the ab initio EoS of [Mazevet et al. \(2019\)](#) with the EoS of the high-pressure ices (VII and X) by [French & Redmer \(2015\)](#), the EoS for ice II-VI by [Journaux et al. \(2020\)](#), the EoS for ice Ih by [Feistel & Wagner \(2006\)](#), and the EoSs by [Wagner & Pruß \(2002\)](#), [Brown \(2018\)](#), [Gordon & McBride \(1994\)](#), and [McBride & Gordon \(1996\)](#) for the liquid and vapor regions where [Mazevet et al. \(2019\)](#) is not applicable.

2.2.4. Numerical method

To solve the two-point boundary value problem of Eqs. (8)–(10), we used a so-called bidirectional shooting method. This means that given the set of input parameters listed in Table 2, Eqs. (8)–(10) were integrated using a fifth-order Cash-Karp Runge-Kutta method with adaptive step size control ([Press et al. 1996](#)). This integration yields as output the total radius of the planet. The two remaining output variables, the planetary Mg/Fe and Si/Fe ratios, can be calculated from the core and mantle composition and the respective layer mass fractions.

2.3. Limits of the forward model

As mentioned in Sect. 2.4.4, in this method, the limits of the forward model need to be considered and the sampling of observable features \mathbf{y} needs to be restricted to the domain of the training set. Otherwise, we find that when an observation is close to the

Table 2. List of forward model parameters.

Symbol	Parameter
Input	
M_{tot}	The exoplanet's total mass
w_{core}	Core mass fraction
w_{vol}	Volatile mass fraction
$x_{\text{SiO}_2} _{\text{mantle}}$	Molar fraction of SiO ₂ in the mantle
$x_{\text{MgO}} _{\text{mantle}}$	Molar fraction of MgO in the mantle
$x_{\text{S}} _{\text{core}}$	Molar fraction of S in the core
Constants	
$T_{\text{surf}} = 300 \text{ K}$	Surface temperature
$P_{\text{surf}} = 1 \text{ atm}$	Surface pressure
Output	
R_{tot}	The exoplanet's total radius
$x_{\text{Mg}}/x_{\text{Fe}} _{\text{Planet}}$	Planetary ratio of Mg to Fe mole fractions
$x_{\text{Si}}/x_{\text{Fe}} _{\text{Planet}}$	Planetary ratio of Si to Fe mole fractions
R_{core}	The exoplanet's iron core radius
R_{mantle}	The exoplanet's mantle radius
R_{vol}	The exoplanet's volatile layer radius

domain boundary of the training set, then the quality of the sampling suffers greatly because the cINN cannot properly learn the inverse mapping for these regions. From the planetary structure model we used, two sets of limits can be constructed for the parameters in the space of observable features.

2.3.1. Limits of Mg/Fe and Si/Fe

The mantle model of [Sotin et al. \(2007\)](#) allows only for mantle compositions that fulfil Eq. (17). This range in possible mantle compositions can be translated into a limit for the possible bulk composition of the modeled exoplanets. The limits for the bulk composition were derived in the following way. In our structure model, iron can occur both in the core and the mantle, whereas Mg and Si are only included in the mantle. Thus, the Mg to Si ratio of the mantle always represents the Mg to Si ratio of the whole exoplanet. The upper limit of the Mg to Si ratio therefore occurs when there is no iron in the mantle, that is, when

$$\frac{x_{\text{Mg}}}{x_{\text{Si}}}\Big|_{\text{Mantle}} = \frac{x_{\text{Mg}}}{x_{\text{Si}}}\Big|_{\text{Planet}} = 2. \quad (18)$$

Multiplying Eq. (18) with $x_{\text{Si}}/x_{\text{Fe}}|_{\text{Planet}}$ returns the upper limit on the planetary Mg to Fe ratio,

$$\frac{x_{\text{Mg}}}{x_{\text{Fe}}}\Big|_{\text{Planet}} = 2 \frac{x_{\text{Si}}}{x_{\text{Fe}}}\Big|_{\text{Planet}}. \quad (19)$$

The lower bound of the $x_{\text{Mg}}/x_{\text{Si}}|_{\text{Planet}}$ occurs when the maximum possible amount of iron is in the mantle, that is, in exoplanets without a core, where $x_{\text{Fe}}/x_{\text{Si}}|_{\text{Mantle}} = x_{\text{Fe}}/x_{\text{Si}}|_{\text{Planet}}$. In this case, we can write similarly to Eq. (17)

$$1 \leq \frac{x_{\text{Mg}}}{x_{\text{Si}}}\Big|_{\text{Planet}} + \frac{x_{\text{Fe}}}{x_{\text{Si}}}\Big|_{\text{Planet}}, \quad (20)$$

or in terms of iron abundance ratios,

$$\frac{x_{\text{Si}}}{x_{\text{Fe}}}\Big|_{\text{Planet}} \leq \frac{x_{\text{Mg}}}{x_{\text{Fe}}}\Big|_{\text{Planet}} + 1. \quad (21)$$

Values of $x_{\text{Si}}/x_{\text{Fe}}|_{\text{Planet}}$ and $x_{\text{Mg}}/x_{\text{Fe}}|_{\text{Planet}}$ that do not fulfil Eqs. (19) and (21) cannot be modeled by the forward model. The prior probability of any such value is therefore zero. This does not mean that in nature, these values cannot occur. It is simply a limitation of the current model.

2.3.2. Limits of M_{tot} and R_{tot}

Similar to the compositional output parameters, we can determine the limits of the forward model for M_{tot} and R_{tot} . The limiting relations in this case are given by the mass radius relation of the most and least dense composition. The densest composition of this forward model is given by a pure iron sphere. The corresponding mass radius relation is

$$\frac{R_{\text{tot}}}{R_{\text{E}}} = 0.796 \cdot \left(\frac{M_{\text{tot}}}{M_{\text{E}}} \right)^{0.2485}. \quad (22)$$

In contrast, the least dense composition we considered is an exoplanet consisting of 70 wt.% of water and 30 wt.% mantle, with a composition given by $x_{\text{Mg}}/x_{\text{Si}}|_{\text{Mantle}} = 2$ and $x_{\text{Fe}}/x_{\text{Si}}|_{\text{Mantle}} = 0$. The corresponding mass radius relation is then

$$\frac{R_{\text{tot}}}{R_{\text{E}}} = 1.341 \cdot \left(\frac{M_{\text{tot}}}{M_{\text{E}}} \right)^{0.2564}. \quad (23)$$

Regarding the limits of the refractory elements, any combinations of total mass and total radius that does not fulfil Eqs. (22) and (23) cannot be modeled with the used forward model. We show in Sect. 2.4.1 that these relations indeed bracket the generated training data of this forward model.

2.4. Training of the cINN

2.4.1. Generation of the training data

In order to train the cINN, we computed 5.9×10^6 forward models. We also experimented with a smaller training set size that comprised only 70% of the total computed models, and found no significant change in the performance of our cINN compared to using the entire dataset (bar the held-out test set). From this experiment, we concluded that this training set size appears to be sufficient for the task, and we did not generate any additional models. The forward model input parameters were drawn at random from the distributions summarized in Table 3. The total mass of the planet was drawn from the uniform distribution $\mathbb{U}(0.5 M_{\text{E}}, 15 M_{\text{E}})$. Since the layer mass fractions ($w_{\text{core}}, w_{\text{mantle}}, w_{\text{vol}}$) sum up to one per definition, they were drawn uniformly from the 3D probability simplex, with the restriction that the maximum water mass fraction cannot exceed a value of 0.7. The mantle Si/Fe and Mg/Fe ratios were calculated from the mantle SiO₂, MgO, and FeO mole fractions, which are the assumed sole constituents of the mantle model of Sotin et al. (2007) and hence sum up to one. Similar to the layer mass fractions, we drew the SiO₂, MgO, and FeO mole fractions uniformly from the 3D probability simplex. Because we used the mantle model of Sotin et al. (2007), however, an additional rejection sampling was performed without any combination that did not fulfill Eq. (17). The resulting distribution on the probability simplex is shown in Fig. 5. The accepted values were then used to calculate the mantle Si/Fe and Mg/Fe ratios, which are the forward model input parameters. In the forward model, the core is made of a mixture of Fe and FeS. Hence, we drew $x_{\text{S}}|_{\text{Core}}$ from $\mathbb{U}(0, 0.5)$. The resulting distribution of all input and output parameters within the training set is shown in Figs. 6 and 7.

Table 3. Distribution of the forward model input parameters within the training set.

Parameters	Distribution in training set
M_{tot}	$M_{\text{tot}} \sim \mathbb{U}(0.5 M_{\text{E}}, 15 M_{\text{E}})$
$w_{\text{core}}, w_{\text{vol}}$	Uniform on Δ^2 , $w_{\text{vol}} = \min(w_{\text{vol}}, 0.7)$
$x_{\text{SiO}_2} _{\text{Mantle}}, x_{\text{MgO}} _{\text{Mantle}}$	Uniform on Δ^2 and Eq. (17)
$x_{\text{S}} _{\text{Core}}$	$x_{\text{S}} _{\text{Core}} \sim \mathbb{U}(0, 0.5)$

Notes. Here, Δ^2 denotes the 3D probability simplex.

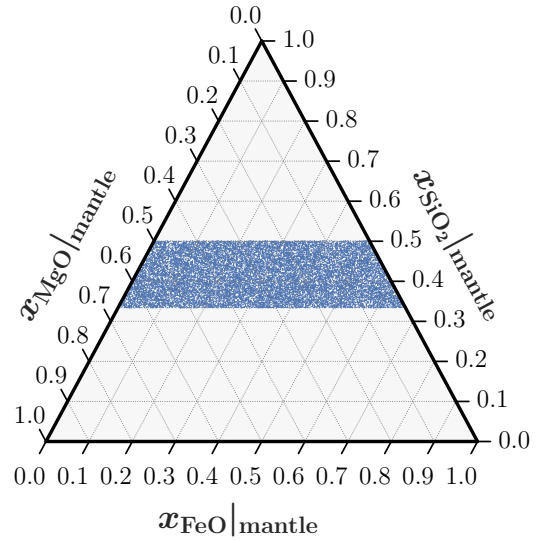


Fig. 5. Distribution of the SiO₂, MgO, and FeO mole fractions of the training set.

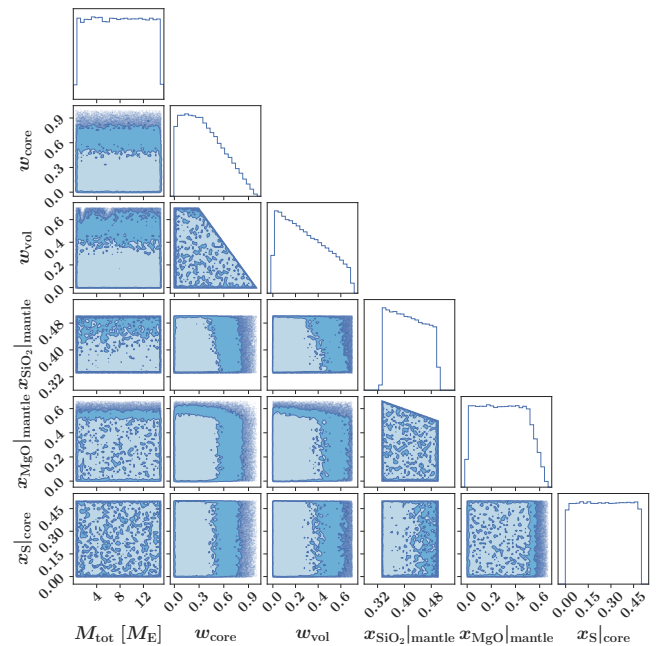


Fig. 6. Distribution of the forward model input parameters as generated for the training set. The underlying distributions from which the parameters were generated are listed in Table 3.

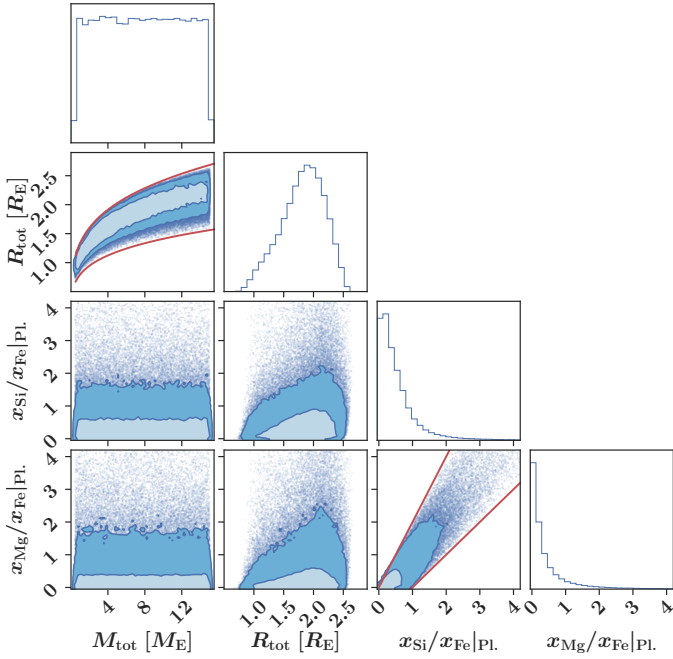


Fig. 7. Distribution of the forward model output parameters as generated for the training set. The underlying distributions from which the parameters were generated are listed in Table 3. The solid red lines indicate the limits of the forward model, as described in Sect. 2.3. The light shaded areas in the 2D diagrams indicate the 68% HDR, and the dark shaded areas are the 89%-HDR.

2.4.2. Data preprocessing

Before any training and prediction was performed, the data were preprocessed. First, we transformed the physical parameters \mathbf{x} and observables \mathbf{y} into log-space. This ensures that the physical parameters stay strictly positive, while it also reduces magnitude differences between different physical quantities. This is important because vastly different magnitudes between parameters can cause training instabilities, for instance, because a single parameter might dominate the target function (loss) that is minimized during the training procedure. To further address this issue, we also centered both \mathbf{x} and \mathbf{y} and rescaled them, such that their standard deviations became unity by subtracting and then dividing by the per parameter and observation means and standard deviations, respectively. These linear scaling transformations are easily inverted at prediction time. The scaling transformation parameters were derived from the training set, and the same transformations were applied to new data at prediction time.

2.4.3. Evaluating the training performance

To quantify the success of the cINN training procedure, we proceeded as in Ksoll et al. (2020). We measured its performance on a test data set, a held-out subset of the training data consisting of 20 000 randomly selected synthetic observations. On this test set, we then first confirmed whether the distribution of latent variables had converged to match the target multivariate normal distribution with unit covariance matrix. We settled for this small test set for our final performance evaluation because we found negligible performance differences in comparison to using a larger held-out test set (e.g., 30%), and we wished to provide the exact performance for the best-informed cINN that was later applied to the real observational data.

Afterward, we evaluated the shape of the predicted posterior distributions by computing the median calibration error s , as proposed in Ardizzone et al. (2019a), for each of the target parameters x . Given an uncertainty interval q , the calibration error $e_{\text{cal},q}$ for a collection of N observations is defined as the difference

$$e_{\text{cal},q} = q_{\text{inliers}} - q, \quad (24)$$

where $q_{\text{inliers}} = N_{\text{inliers}}/N$ denotes the fraction of observations, where the true value \tilde{x} lies within the q -confidence interval of the predicted posterior PDF. Values of $e_{\text{cal},q} < 0$ signify that the predicted PDFs are too narrow, whereas positive values suggest the opposite, that is, that the PDFs are too broad. The median calibration error s was derived as the median of the absolute calibration errors over the range of confidences from 0 to 1.

Next, we quantified the cINNs predictive capability for maximum a posteriori (MAP) point estimates \hat{x} . To do this, we derived an accuracy for the individual target parameters x over the entire test set as given by the root mean squared error (RMSE) and normalized RMSE (NRMSE). They are defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (\hat{x}_i - \tilde{x}_i)^2}{N}}, \quad (25)$$

where \tilde{x}_i is the ground-truth value of the target parameter for the i th observation, and

$$\text{NRMSE} = \frac{\text{RMSE}}{\bar{x}}, \quad (26)$$

where $\bar{x} = x_{\text{max}}^{\text{ts}} - x_{\text{min}}^{\text{ts}}$ denotes the range of parameter x within the training data. To determine the MAP estimates \hat{x} from the predicted samples of the posterior distribution, we performed a kernel density estimation (KDE) to model the PDF and find its maximum. This KDE employed a Gaussian kernel function and was computed on an evenly spaced grid of 1024 points, covering the full range of the given posterior samples. The kernel bandwidth h was derived using Silverman's rule (Silverman 1986),

$$h = 1.06 \cdot \min\left(\sigma, \frac{\text{IQR}}{1.34}\right) \cdot n^{-\frac{1}{5}}, \quad (27)$$

where IQR, σ , and n denote the interquartile range, standard deviation, and number of the posterior samples, respectively.

2.4.4. Predicting posteriors for noisy observations

As the method was outlined so far, the cINN did not include the possibility that a given input observation can be uncertain. Instead, the described method assumed perfect observations as an input. However, in many real-world applications, all observed quantities usually have measurement uncertainties. In order to predict the posterior probability distribution of \mathbf{x} given a noisy observation using the cINN, we devised the following strategy.

Let the noisy observation be represented by \mathbf{y}^* and the true observable properties of the target be denoted as \mathbf{y} . For this paper, we assumed that the distribution of \mathbf{y}^* follows a multivariate normal distribution of dimension k with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$, that is,

$$\mathbf{y}^* \sim \mathbb{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (28)$$

Given the law of total probability, the posterior probability distribution $p(\mathbf{x} | \mathbf{y}^*)$ can be written as

$$p(\mathbf{x} | \mathbf{y}^*) = \int_Y p(\mathbf{x} | \mathbf{y}^* \cap \mathbf{y} = \mathbf{y}') \Phi_{\mathbf{y}' | \mathbf{y}^*}(\mathbf{y}') d\mathbf{y}', \quad (29)$$

where \mathbf{y}' is a point in the space of observational parameters, and $\Phi_{\mathbf{y}' | \mathbf{y}^*}$ is the probability density function of \mathbf{y}' given \mathbf{y}^* . Because we assumed that \mathbf{y}^* follows a multivariate normal distribution, $\Phi_{\mathbf{y}' | \mathbf{y}^*}$ is given by

$$\Phi_{\mathbf{y}' | \mathbf{y}^*}(\mathbf{y}') = \frac{1}{(2\pi)^{k/2} \sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{y}' - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{y}' - \boldsymbol{\mu})\right). \quad (30)$$

Next we used that \mathbf{x} is conditionally independent of \mathbf{y}^* given \mathbf{y} , that is, $((\mathbf{x} \perp \mathbf{y}^*) | \mathbf{y})$. It follows that $p(\mathbf{x} | \mathbf{y}^* \cap \mathbf{y}) = p(\mathbf{x} | \mathbf{y})$ and hence

$$p(\mathbf{x} | \mathbf{y}^*) = \int_Y p(\mathbf{x} | \mathbf{y} = \mathbf{y}') \Phi_{\mathbf{y}' | \mathbf{y}^*}(\mathbf{y}') d\mathbf{y}', \quad (31)$$

where $p(\mathbf{x} | \mathbf{y} = \mathbf{y}')$ can be calculated for a given \mathbf{y}' from Eq. (5).

The posterior probability distribution can now be calculated using simple Monte Carlo integration. Here the Monte Carlo samples of \mathbf{x} were generated by first drawing N times a sample \mathbf{y}'_i from the multivariate normal distribution given in Eq. (28). For each sample \mathbf{y}'_i , we then calculated the point estimate of $p(\mathbf{x} | \mathbf{y}'_i)$ using the cINN as outlined in Sect. 2.1.1. For each \mathbf{y}'_i , we therefore sampled another M times from the latent variables \mathbf{z} and evaluated for each \mathbf{z}_i the backward mapping of the cINN, that is, $g(\mathbf{z}_i, \mathbf{c} = \mathbf{y}'_i)$.

This resulted in $N \times M$ samples of \mathbf{x} drawn from the posterior probability distribution $p(\mathbf{x} | \mathbf{y}^*)$. By definition, the conditional probability $p(\mathbf{x} | \mathbf{y})$ is zero if $p(\mathbf{x}) = 0$ or $p(\mathbf{y}) = 0$. Hence, if the prior distribution of \mathbf{x} has a compact support, then $p(\mathbf{x} | \mathbf{y})$ is automatically zero for any \mathbf{x} outside of the domain of \mathbf{x} . While the extent of the domain is in principle learned by the cINN, it is still possible that the cINN maps some \mathbf{z}_i to an \mathbf{x} for which $p(\mathbf{x}) = 0$. During the sampling of \mathbf{z} , all such samples should thus be rejected. Additionally, the compact support of $p(\mathbf{x})$ simultaneously limits the possible output values of the forward model and therefore also induces limits on \mathbf{y} . We can hence forgo evaluating the cINN for any \mathbf{y}'_i for which $p(\mathbf{y}'_i) = 0$. The kind of limits introduced for \mathbf{y} depend on the forward model. We discuss this aspect in more detail in Sect. 2.3.

3. Method validation

In order to validate the sampling scheme outlined in Sect. 2, we used a simple toy model to benchmark the proposed scheme against a common Metropolis-Hastings MCMC sampler. We wished to test here in particular how the sampling performs when i) the posterior distribution of the model parameters has nonzero probability along the boundary of the prior domain, and ii) the observation is close to a region in which the forward model can no longer be applied. This will in particular demonstrate how the method performs when an observation is close to the border of the set of training data.

To mimic situations (i) and (ii), we set up the following toy model. We defined the space of model parameters $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$, as well as the space of observable features $\mathbf{y} = (y_1, y_2) \in \mathbb{R}^2$. The forward model $f(\mathbf{x})$ is given by the linear model

$$f\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \quad (32)$$

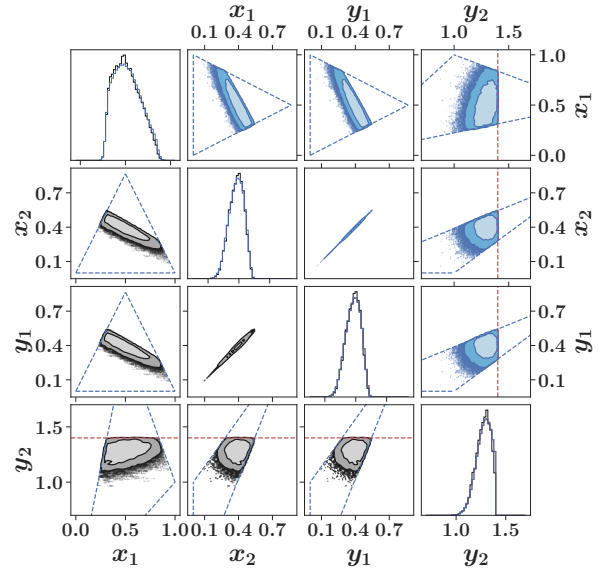


Fig. 8. Comparison of the cINN and an MCMC method when applied to the toy model. The data in the lower triangle (black) were predicted by the cINN method, and the data in the upper triangle (blue) were generated with an MCMC sampler. The dashed blue lines indicate the boundaries of the prior domain, outside of which the prior probability is zero. The dashed red line indicates the upper limit on y_2 as in Eq. (36). The histograms of the analytical solution are omitted since they overlap with the MCMC data. The light shaded areas in the 2D scatter plots indicate the 68% HDR, and the dark shaded areas are the 89%-HDR.

The prior domain is given by an equilateral triangle in the space of model parameters, defined by its corners,

$$\mathbf{c}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{c}_3 = \begin{bmatrix} 0.5 \\ 0.5\sqrt{3} \end{bmatrix}. \quad (33)$$

For simplicity, we chose a uniform prior probability distribution within the prior domain. $p(\mathbf{x})$ is therefore written as

$$p(\mathbf{x}) = \begin{cases} \frac{4}{\sqrt{3}} & \text{if } \mathbf{x} \text{ within triangle}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \\ 0, & \text{if } \mathbf{x} \text{ outside triangle}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \end{cases}, \quad (34)$$

where the probability to be within the triangle is the inverse of the area of the triangle. For a noisy observation \mathbf{y}^* defined as in Eq. (28), that is, as a multivariate normal distribution,

$$\mathbf{y}^* \sim \mathbb{N}_2\left(\boldsymbol{\mu} = \begin{bmatrix} 0.4 \\ 1.3 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}\right), \quad (35)$$

the posterior distribution on \mathbf{x} will become heavily truncated by the prior domain. To mimic case (ii), we further added a restriction on the observable parameter y_2 and arbitrarily set an upper limit of

$$y_2 \leq 1.4. \quad (36)$$

The cINN was then trained on a dataset containing 10^6 samples of \mathbf{x} and corresponding $f(\mathbf{x})$ values. This number of training samples is larger than necessary. As the training samples are cheap to generate, we forwent finding an optimal number of training samples. Given \mathbf{y}^* as in Eq. (35), we then followed the method outlined in Sect. 2.4.4 to sample from the posterior distribution. The resulting distributions from the cINN and the MCMC sampling are shown in Fig. 8, with the summary

statistics of the marginalized distributions listed in Table 4. The summary statistics of the marginalized distributions show that the median of each variable does not vary between methods. Moreover, the centered 1σ interval (containing 68.3% of all samples) and the centered 2σ interval (containing 95.4% of all samples) are almost identical (except for a 0.01 deviation of the lower bound of the 1σ interval of y_2 and the upper bound of the 2σ interval of x_1). To compare the shape of the resulting distributions, we also computed the Hellinger distance h between the marginalized probability distributions of the two methods.

The Hellinger distance $h(r, q)$ between two discrete probability distributions r and q is given by

$$h(r, q) = \sqrt{1 - b(r, q)}, \quad (37)$$

where $b(r, q)$ is the Bhattacharyya coefficient (see Hellinger 1909),

$$b(r, q) = \sum_x \sqrt{r(x)q(x)}. \quad (38)$$

The Hellinger distance is a proper distance metric. It is zero if the distributions r and q are identical and one if they are disjoint. Here, the Hellinger distance was determined from the histograms of the 1D marginalized distributions generated using the cINN or MCMC method. The optimal number of bins was estimated using the method of Doane (1976).

We report that the Hellinger distance between the distributions for x_1 is $h = 0.026$, whereas for x_2 , it is $h = 0.022$. For the observable features, the Hellinger distance has similarly low values of $h = 0.023$ for y_1 and $h = 0.046$ for y_2 . For comparison, the Hellinger distance of two normal distributions, where the median of the two distributions differs by 10^{-1} or 10^{-2} , is $h = 0.035$ or $h = 0.004$, respectively. For more details see Table 5.

The 2D marginalized posterior densities in Fig. 8 show that the posterior distribution on the input parameters x_1 and x_2 is strongly truncated, as expected. Moreover, the upper limit of y_2 has a notable effect on the distribution of the input parameters. No major differences between the two methods are visible, however. We conclude that the cINN method can be successfully used for this simple model, even when the observation is close to the boundary of the training data.

4. Results

4.1. Training performance

For the planet characterization task, we trained a cINN to predict the physical parameters R_{core} , R_{mantle} , R_{vol} (i.e., the radii of the core, mantle, and surface layer, respectively), w_{core} , w_{vol} , $x_{\text{SiO}_2}|_{\text{mantle}}$, $x_{\text{MgO}}|_{\text{mantle}}$ and $x_{\text{S}}|_{\text{core}}$ from the observables M_{tot} , R_{tot} , $x_{\text{Si}}/x_{\text{Fe}}|_{\text{planet}}$, and $x_{\text{Mg}}/x_{\text{Fe}}|_{\text{planet}}$, using the database described in Sect. 2.4.1. To fully assess the cINN performance, we would need to generate posterior distributions for a large sample of mock observations spread over the set of training data. To be statistically significant, the number of observations would need to be on the order of $\sim 10^3$ – 10^4 cases. Computing the posterior distribution of such a large number of cases using an MCMC method is computationally unfeasible. Instead, we used two complementary methods to assess the performance of the cINN.

4.1.1. Performance on test data

First, we tested the performance of the trained cINN model on synthetic, held-out test data for all the target parameters. As

Table 4. Summary statistics (i.e., median centered 1σ interval and centered 2σ interval) of the marginalized posterior distributions of the toy model.

Method: cINN			
Parameter	Median	$1-\sigma$	$2-\sigma$
x_1	0.51	[0.36, 0.68]	[0.29, 0.81]
x_2	0.38	[0.30, 0.45]	[0.23, 0.50]
y_1	0.38	[0.30, 0.45]	[0.23, 0.50]
y_2	1.28	[1.20, 1.35]	[1.11, 1.39]
Method: MCMC			
Parameter	Median	$1-\sigma$	$2-\sigma$
x_1	0.51	[0.36, 0.68]	[0.29, 0.82]
x_2	0.38	[0.30, 0.45]	[0.23, 0.50]
y_1	0.38	[0.30, 0.45]	[0.23, 0.50]
y_2	1.28	[1.19, 1.35]	[1.11, 1.39]
Method: analytical			
Parameter	Median	$1-\sigma$	$2-\sigma$
x_1	0.51	[0.37, 0.69]	[0.29, 0.82]
x_2	0.38	[0.30, 0.45]	[0.23, 0.50]
y_1	0.38	[0.30, 0.45]	[0.23, 0.50]
y_2	1.28	[1.20, 1.36]	[1.11, 1.39]

Table 5. Hellinger distances comparing the marginalized posterior distributions for the toy model.

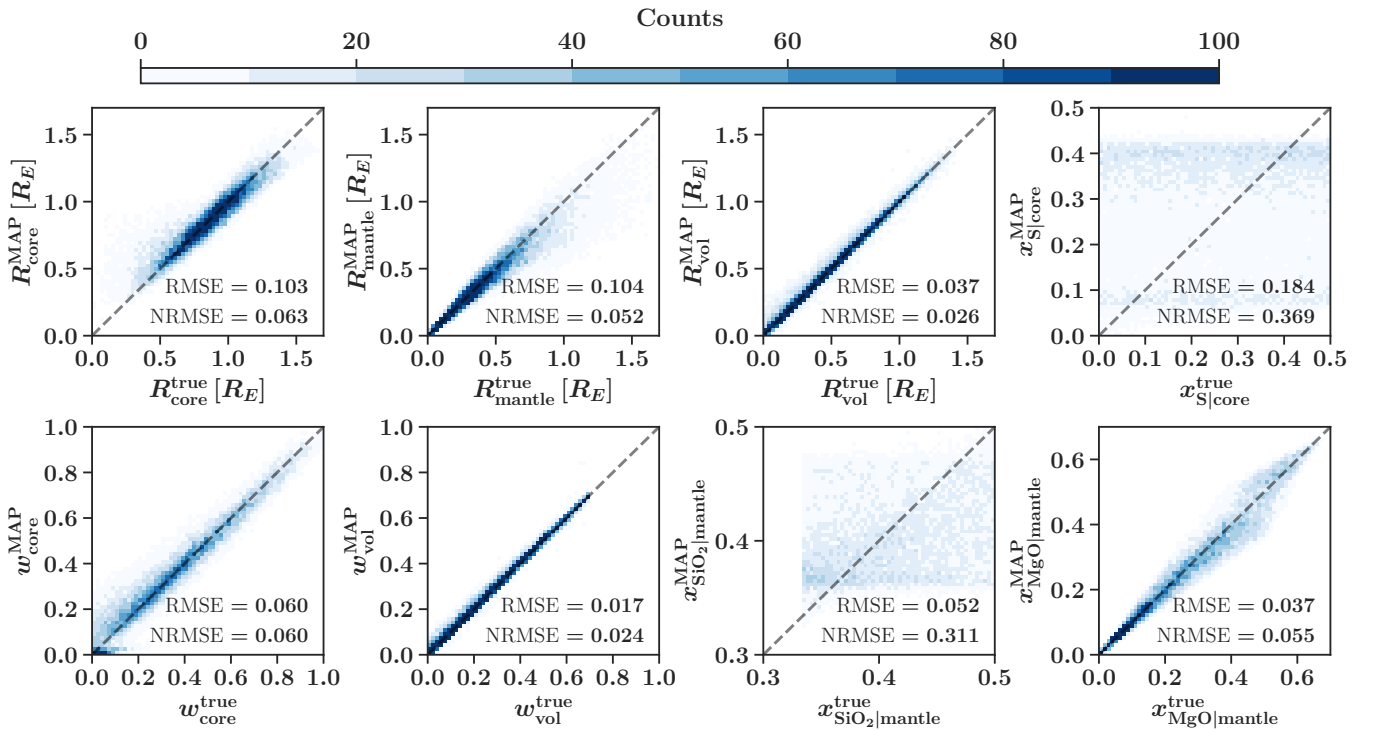
	$h(p_{\text{analytical}}, p_{\text{cINN}})$	$h(p_{\text{analytical}}, p_{\text{MCMC}})$	$h(p_{\text{MCMC}}, p_{\text{cINN}})$
x_1	0.026	0.006	0.026
x_2	0.020	0.008	0.022
y_1	0.023	0.008	0.023
y_2	0.048	0.006	0.046

described in Sect. 2.4.3, we computed the RMSE and NRMSE for the MAP point estimates for the test data, as well as the median calibration errors s and median uncertainty at 68% confidence u_{68} (i.e., the width of the 68% confidence interval) of the posterior distributions. This method allowed us to estimate how well the cINN learned the shape of the posterior distribution for a single-point estimate. The computed values are summarized in Table 6. Figure 9 also provides 2D histograms comparing the MAP estimates against the corresponding ground-truth values. The performance on the synthetic test set was evaluated without error resampling, that is, assuming perfect observations. As the histograms and the RMSEs/NRMSEs demonstrate, the cINN can recover R_{core} , R_{mantle} , R_{vol} , w_{core} , w_{vol} , and $x_{\text{MgO}}|_{\text{mantle}}$ quite well overall with MAP estimates that fall very close or directly on to the ideal one-to-one correlation in comparison to the ground truth. Examining R_{core} and R_{mantle} more closely, we realized some systematic divergences from the one-to-one correlation toward small cores and large mantles, however, indicating that these properties appear to be harder to constrain within these regimes. Likely related to this, we also observed a systematic underestimation of the core mass fraction for very low-mass cores between 0 and 0.1, showing that the cINN also slightly struggles in this range.

For the remaining two properties, $x_{\text{S}}|_{\text{core}}$ and $x_{\text{SiO}_2}|_{\text{mantle}}$, we find that the MAP point estimates cannot match the ground-truth values at all. The results are scattered across the entire parameter

Table 6. Overview of cINN test performance for the planet characterization task.

Parameter	RMSE _{MAP}	NRMSE _{MAP}	s	u_{68}	u_{68}^{training}
R_{core}	0.1026	0.0630	0.005	0.160	0.155
R_{mantle}	0.1042	0.0520	0.007	0.133	0.123
R_{vol}	0.0365	0.0256	0.002	0.043	0.041
w_{core}	0.0598	0.0601	0.005	0.095	0.091
w_{vol}	0.0170	0.0243	0.002	0.022	0.021
$x_{\text{SiO}_2 \text{mantle}}$	0.0518	0.3109	0.001	0.107	0.104
$x_{\text{MgO} \text{mantle}}$	0.0369	0.0555	0.001	0.063	0.060
$x_{\text{S} \text{core}}$	0.1843	0.3687	0.002	0.340	0.155


Fig. 9. Distribution of $N=20,000$ MAP estimates of the trained cINN plotted against the ground truth from the training data set, shown for the model input parameters.

ranges with no discernible overdensity at the one-to-one correlation. The median calibration errors s of the underlying predicted posterior distributions show that the cINN finds very well calibrated solutions (i.e., posteriors that are neither too broad nor too narrow) with values below 0.7% for all target parameters, including $x_{\text{S}|\text{core}}$ and $x_{\text{SiO}_2|\text{mantle}}$. From the median widths of the 68% confidence intervals, which are on the order of ≈ 0.1 on average, we find, however, that the posterior distributions tend to be rather broad in general (taking the target parameter ranges into account).

For the posterior distributions themselves, the issues with the $x_{\text{S}|\text{core}}$ and $x_{\text{SiO}_2|\text{mantle}}$ MAP estimates result from the fact that the cINN consistently predicts almost perfectly uniform distributions across the parameter ranges for these two parameters for all examples in the test set. In this case, performing an MAP estimate simply becomes unfeasible as it merely picks up on minor random fluctuations in these almost uniform distributions rather than identifying distinct peaks in the posteriors. As we show below in our direct comparison of the cINN and an MCMC approach in Sect. 4.2.2, these almost uniform posterior

distributions of $x_{\text{S}|\text{core}}$ and $x_{\text{SiO}_2|\text{mantle}}$ are not a flaw of our cINN model, but are also recovered by the MCMC. Because both cINN and MCMC therefore return rather uninformative posterior distributions for $x_{\text{S}|\text{core}}$ and $x_{\text{SiO}_2|\text{mantle}}$, we have to conclude that these two physical parameters cannot be constrained from the observables M_{tot} , R_{tot} , $x_{\text{Si}}/x_{\text{Fe}}|_{\text{Planet}}$, and $x_{\text{Mg}}/x_{\text{Fe}}|_{\text{Planet}}$.

4.1.2. Recalculation error

Complementary to the analysis of the cINN performance on the test data, we assessed whether the generated posterior samples correctly map back to their corresponding input observations. We performed this by computing the recalculation error ε of the cINN for all four input observables for each posterior sample of 5000 randomly selected synthetic examples from the held-out test set (covering the entire domain of the training data). This means that for each sample $\mathbf{x}^{(i)}$ (i.e., predicted set of the eight target parameters) of the 1024 samples generated by the cINN per posterior, we ran the forward model $f(\cdot)$ again and determined the relative difference between the forward model output $f(\mathbf{x}^{(i)})$

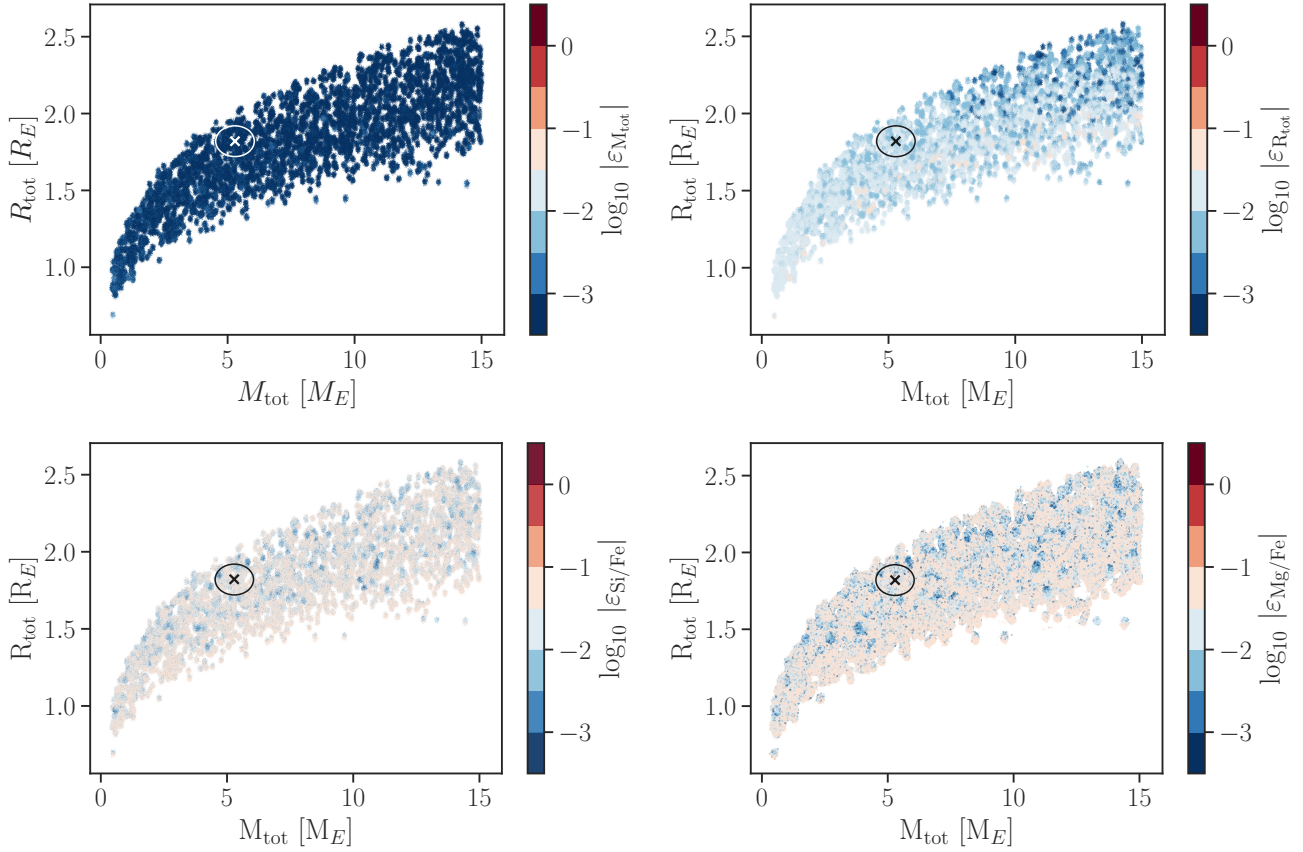


Fig. 10. Recalculatation error ε of the four observables M_{tot} , R_{tot} , $x_{\text{Si}}/x_{\text{Fe}}|_{\text{Planet}}$, and $x_{\text{Mg}}/x_{\text{Fe}}|_{\text{Planet}}$ as a function of true mass and radius. The black and white encircled cross marks the observed mass and radius of K2-111 b and its corresponding 1σ uncertainties.

and the original network input observation $\mathbf{y}^{(i)}$. ε was calculated for each component k of $\mathbf{y}^{(i)}$ as

$$\varepsilon(k) = 100 \cdot \frac{f(g(\mathbf{z}, \mathbf{c} = \mathbf{y}^{(i)}))_k - y_k^{(i)}}{y_k^{(i)}} = 100 \cdot \frac{f(\mathbf{x}^{(i)})_k - y_k^{(i)}}{y_k^{(i)}}. \quad (39)$$

The color-coding in Fig. 10 shows the recalculatation error for our four observables as a function of the ground-truth mass and radius. For comparison, we also indicate the position of our real-world test case K2-111 b. With average recalculatation errors of $-0.01^{+0.06}_{-0.06}\%$, $1.19^{+0.96}_{-0.71}\%$, $3.68^{+2.80}_{-2.62}\%$, and $3.68^{+2.79}_{-2.61}\%$ for M_{tot} , R_{tot} , $x_{\text{Si}}/x_{\text{Fe}}|_{\text{Planet}}$, and $x_{\text{Mg}}/x_{\text{Fe}}|_{\text{Planet}}$, respectively, we find an overall excellent agreement with the input observations. We can conclude that the cINN does indeed return valid posterior distributions on our synthetic test set. Together with the good posterior peak recovery indicated by the low MAP estimate NRMSE for all parameters except $x_{\text{Si}}|_{\text{core}}$ and $x_{\text{SiO}_2}|_{\text{mantle}}$, the cINN has therefore demonstrated a highly satisfactory predictive performance on the synthetic test data.

4.2. Comparison between cINN and MCMC for K2-111 b

In order to demonstrate that our cINN provides accurate posterior distributions of planetary parameters, we considered the case of K2-111 b (Mortier et al. 2020), which was observed to be a planet of radius $1.82^{+0.11}_{-0.09} R_E$ with a mass of $5.29^{+0.76}_{-0.77} M_E$. Its high mean density of $4.81^{+1.25}_{-1.01} \text{ g cm}^{-3}$ implies that the planet is very likely to have only a tiny gas envelope, making it a well-suited example for our purpose. We also assumed that the composition

of the photosphere of K2-111 is identical to that of the planet in terms of the elemental ratios of the refractory elements Mg, Fe, and Si. The elemental ratios given in Mortier et al. (2020) are $1.82^{+0.48}_{-0.38}$ for Si/Fe and $2.51^{+0.85}_{-0.63}$ for Mg/Fe. With this observation, we sampled the posterior distribution of the forward model parameters using the cINN method as described in Sect. 2.4.4.

4.2.1. MCMC setup

For this comparison, we also sampled the posterior probability distribution using an MCMC method, as is often employed to infer planetary interiors (see, e.g., Haldemann et al., in prep.; Dorn et al. 2017a). In particular, we used the adaptive Metropolis-Hastings MCMC algorithm (Haario et al. 2001), sampling the posterior distribution of

$$p(\mathbf{x}|\mathbf{y}_{\text{obs}}) \propto \mathbb{L}(\mathbf{y}_{\text{obs}}|\mathbf{x}) \mathbb{P}(\mathbf{x}) \quad (40)$$

given the same forward model $f(\cdot)$ as described in Sect. 2.2. We considered the same prior $\mathbb{P}(x)$ as we used to construct the training data (see Table 3). The likelihood $\mathbb{L}(\mathbf{y}_{\text{obs}}|\mathbf{x})$ was calculated using

$$\mathbb{L}(\mathbf{y}_{\text{obs}}|\mathbf{x}) = \frac{1}{(2\pi)^{N/2} \left(\prod_{i=1}^N \sigma_i\right)^{1/2}} \exp\left(-\frac{1}{2} \sum_{i=1}^N \frac{(f(\mathbf{x})_i - \mu_i)^2}{\sigma_i^2}\right), \quad (41)$$

where μ and σ^2 are the mean and variance of \mathbf{y}_{obs} and N is the number of dimensions or parameters of \mathbf{y}_{obs} .

The MCMC was initialized at a random location in the sample space. We then ran the MCMC for $\sim 5 \times 10^5$ steps. The

Table 7. Various statistics of the marginalized posterior distribution of the forward model parameters.

Parameter	Method: cINN			Method: MCMC			Difference: $100 \cdot (x_{\text{cINN}} - x_{\text{MCMC}}) / x_{\text{MCMC}}$		
	Median	1- σ	2- σ	Median	1- σ	2- σ	Δ Median	Δ 1- σ	Δ 2- σ
w_{core}	0.09	[0.03, 0.17]	[0.00, 0.24]	0.09	[0.03, 0.16]	[0.01, 0.23]	5.84	[-4.30, 6.60]	[-23.19, 6.95]
w_{rock}	0.64	[0.45, 0.81]	[0.30, 0.92]	0.64	[0.47, 0.78]	[0.33, 0.89]	0.69	[-4.01, 3.51]	[-8.66, 3.32]
w_{vol}	0.25	[0.07, 0.46]	[0.00, 0.64]	0.26	[0.11, 0.44]	[0.03, 0.61]	-3.85	[-34.35, 4.37]	[-89.56, 4.35]
$x_{\text{S}} _{\text{core}}$	0.26	[0.08, 0.42]	[0.01, 0.48]	0.27	[0.09, 0.43]	[0.01, 0.49]	-5.10	[-8.83, -3.23]	[-16.19, -1.32]
$x_{\text{FeO}} _{\text{mantle}}$	0.09	[0.03, 0.17]	[0.01, 0.24]	0.10	[0.03, 0.17]	[0.00, 0.23]	-4.56	[-8.95, 3.25]	[24.75, 5.10]
$x_{\text{SiO}_2} _{\text{mantle}}$	0.41	[0.36, 0.47]	[0.34, 0.49]	0.40	[0.35, 0.46]	[0.34, 0.49]	1.47	[0.79, 1.49]	[0.14, 0.33]
$x_{\text{MgO}} _{\text{mantle}}$	0.50	[0.39, 0.58]	[0.30, 0.63]	0.50	[0.42, 0.57]	[0.33, 0.63]	-0.19	[-6.02, 1.89]	[-8.79, 1.26]
$R_{\text{core}} [R_E]$	0.55	[0.37, 0.68]	[0.19, 0.78]	0.54	[0.37, 0.67]	[0.21, 0.76]	2.22	[-0.51, 2.10]	[-7.53, 2.08]
$R_{\text{rock}} [R_E]$	0.88	[0.70, 1.08]	[0.56, 1.31]	0.86	[0.71, 1.04]	[0.57, 1.23]	1.89	[-0.39, 4.67]	[-2.29, 6.33]
$R_{\text{vol}} [R_E]$	0.38	[0.15, 0.66]	[0.01, 0.88]	0.40	[0.21, 0.64]	[0.07, 0.85]	-4.89	[-29.47, 2.84]	[-85.79, 4.07]

Notes. Comparing the cINN method with a Metropolis-Hastings MCMC scheme when applied to K2-111 b. $\Delta 1\sigma$ and $\Delta 2\sigma$ are the relative differences of the interval boundaries of the 1 σ and 2 σ intervals. The w_{rock} and $x_{\text{FeO}}|_{\text{mantle}}$ were calculated from the other layer mass fractions and other mantle oxide fractions.

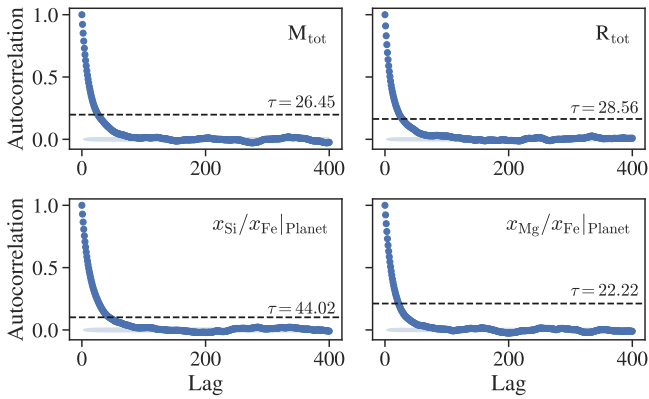


Fig. 11. Autocorrelation as a function of lag calculated from the Markov chain of the four output parameters of the forward model. The autocorrelation time τ of each parameter was calculated following Hogg & Foreman-Mackey (2018). The dashed lines indicate the autocorrelation when the lag is equal to the autocorrelation time.

resulting Markov chain had an autocorrelation time τ between 28 and 44 steps for the four output parameters (see Fig. 11). The autocorrelation time was calculated following Hogg & Foreman-Mackey (2018).

To generate independent samples from the Markov chain, we began by discarding the first 2000 steps along the Markov chain due to burn in, and then, accounting for the maximum autocorrelation time, add every 50th step along the chain to the set of independent samples. This resulted in a total of 10^4 independent samples from the Markov chain, which is sufficient given the shape and number of dimensions of the posterior probability distribution.

4.2.2. Comparison of marginalized posterior distributions

In order to compare the performance of the cINN with the MCMC method, we show in Table 7 a summary of the key statistics of the 1D marginalized posterior PDF. For each model parameter, the median as well as the centered 1 σ and 2 σ intervals (containing 68% and 95% of all samples, respectively) are shown. The median values show that both methods return almost identical results with a maximum difference of $\sim 5.8\%$. For the boundaries of the centered 1 σ and 2 σ regions, the differences

Table 8. Hellinger distance metric between the marginalized posterior distributions of the cINN and MCMC method.

Parameter	Hellinger distance h
M_{tot}	0.024
R_{tot}	0.071
$x_{\text{Si}}/x_{\text{Fe}} _{\text{Planet}}$	0.031
$x_{\text{Mg}}/x_{\text{Fe}} _{\text{Planet}}$	0.044
w_{core}	0.052
w_{vol}	0.107
$x_{\text{S}} _{\text{core}}$	0.042
$x_{\text{SiO}_2} _{\text{mantle}}$	0.040
$x_{\text{MgO}} _{\text{mantle}}$	0.092
R_{core}	0.045
R_{rock}	0.072
R_{vol}	0.115

are larger, notably for the lower boundaries of R_{vol} , w_{vol} , $x_{\text{S}}|_{\text{core}}$, $x_{\text{FeO}}|_{\text{mantle}}$, and $x_{\text{SiO}_2}|_{\text{mantle}}$. However, the Hellinger distances h between the marginalized posteriors predicted by the two methods are very low ($h < 0.05$) for all parameters, except for R_{vol} , w_{vol} , and R_{tot} (see Table 8). The largest Hellinger distance of 0.103 for R_{vol} is equivalent to the distance of two standard normal distributions whose median is shifted by 0.25. The fact that R_{vol} has the largest error here is an odd result at first glance because R_{vol} appeared to be one of the best-constrained parameters in our test on the synthetic data. There are multiple possible explanations for this result. First, this might simply be an effect of the uncertainty of the observables, which are taken into account here, but not in our previous synthetic test. Second, although the MAP estimates for R_{vol} are indeed overall quite excellent, there are a few statistical outliers with slightly larger discrepancies as well, so that the result for K2-111 b may just be one such outlier. Last, it is also important to note that the MAP RMSE and the Hellinger distance measure two different properties of the predicted posterior distributions. The former quantifies the point estimation capability, that is, the ability of the cINN to recover the true value as the most likely peak in the posterior distribution. The latter, on the other hand, quantifies a difference in the shape of the posterior with respect to the MCMC ground truth. It might therefore be that the shape of the R_{vol} posteriors in our synthetic test also tends to deviate slightly more from the

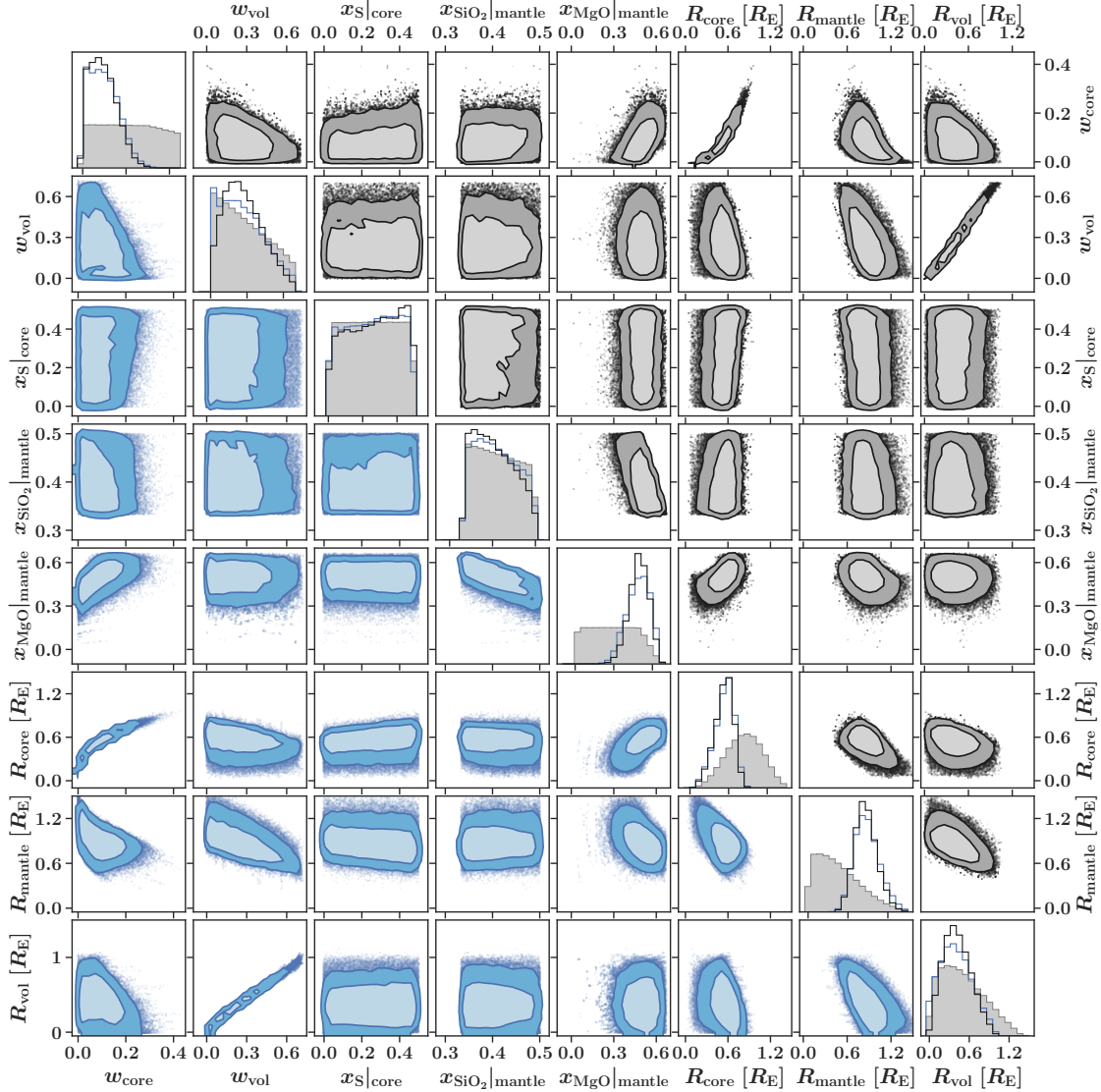


Fig. 12. Comparison of the cINN and an MCMC method when applied to K2-111 b. The data in the lower triangle of each panel (blue points) were generated with the cINN method, and the data in the upper triangle (black points) were generated with the MCMC sampler. In the diagonal panels, we also show the marginalized prior probability (gray). The light shaded areas in the 2D diagrams indicate the 68% HDR, and the dark shaded areas are the 89% HDR.

ground-truth shape, even though the low MAP NRMSE indicates that they have the correct peak. Unfortunately, due to the prohibitive computational cost of running the MCMC for a statistically large enough sample of the synthetic test data, we were unable to quantify this effect in our previous test.

In Figs. 12 and 13, we show the pairwise marginalized 2D posterior PDF of all parameters. In the diagonal of these figures, we also show the 1D histograms of the marginalized PDF, as well as the prior distribution of the points in the training data. Overall, the shape of the pairwise marginalized 2D posterior PDF is very similar between the two methods. Figure 13 also shows that the cINN method can be used close to the boundary of the training data (red lines). For the 1D histograms, the largest difference is again seen for R_{vol} and w_{vol} , especially toward dry compositions.

The layer mass fractions and the mantle composition are each a set of compositional variables (hence they sum up to one). Their distribution on the ternary diagram is therefore shown in Fig. 14. For the layer mass fractions, the agreement in the region above 0.1 w_{vol} is very good. Because this is also present in the 1D marginalized distributions, there are fewer samples with low

volatile content in the Markov chain than in the set generated using the cINN. For the mantle composition, the posterior distribution of both methods is centered around compositions with 60% MgO and less than 10% FeO. Compared to the MCMC method, the cINN predicts slightly more compositions with larger amounts of SiO₂ than MgO, which results in the two kinks in the contours in the ternary diagram.

4.2.3. Recalculation error of trained cINN for K2-111 b

To assess the quality of the inverse mapping from \mathbf{y} to \mathbf{x} , we calculated the recalculation error ϵ for each sample $\mathbf{x}^{(i)}$ generated with the cINN according to Eq. (39). In Fig. 15, we show the distribution of the recalculation errors of all data variables together with w_{vol} , given the set of $\mathbf{x}^{(i)}$ generated for the case of K2-111 b. The cINN learned the mapping of the total mass best, and the median recalculation error of the other variables is between 1.3% and 2.5%. Figure 15 also indicates that the recalculation errors increase toward low values of w_{vol} , especially below 0.1. This indicates that the quality of the cINN mapping in this region is

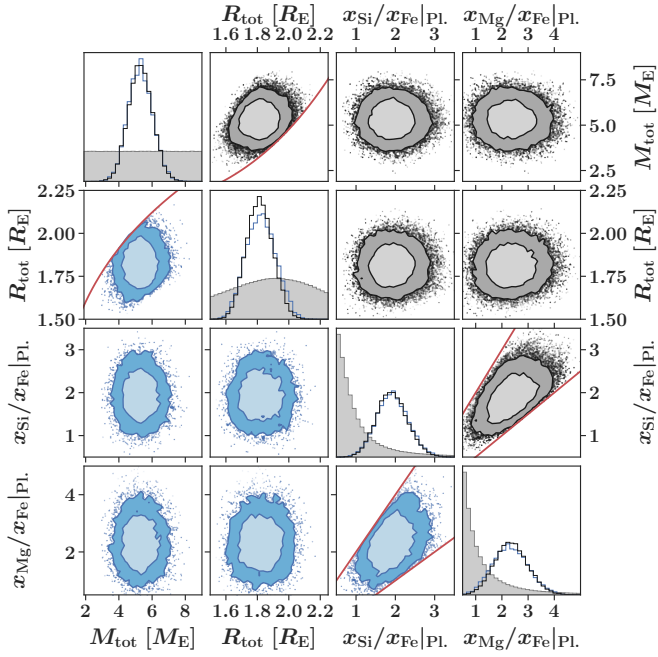


Fig. 13. Comparison of cINN and MCMC posteriors when applied to K2-111 b. The colors and shadings are the same as in Fig. 12. The solid red line indicates the limits of the forward model, as discussed in Sect. 2.3.

not yet optimal. This likely explains the observed differences in the posterior distribution of w_{vol} and R_{vol} between the MCMC and the cINN method.

5. Discussion

5.1. Comparison to the initial characterization of K2-111 b

The exoplanet K2-111 b was also characterized in [Mortier et al. \(2020\)](#). In their work, two different internal structure models were used for characterization, one considering four layers, that is, an iron core, a silicate mantle, a water layer, and an H/He envelope. The other model only considered two layers, that is, an iron core with a surrounding mantle. In the first model, the inferred bulk composition of K2-111 b was $w_{\text{core}} = 0.10^{+0.07}_{-0.07}$, $w_{\text{mantle}} = 0.68^{+0.13}_{-0.14}$, $w_{\text{water}} = 0.20^{+0.16}_{-0.13}$, and $\log_{10} w_{\text{H/He}} = -8.76^{+2.20}_{-2.21}$. The inferred small amount of H/He by [Mortier et al. \(2020\)](#) is one reason we chose this exoplanet for our work because we trained the cINN only on planetary structures without H/He layers so far.

When we compare this to the results obtained using the cINN, which are given in Table 7, we see that the inferred core mass fraction is almost the same, while the mantle mass fraction is slightly smaller and the water mass fraction is slightly larger than in [Mortier et al. \(2020\)](#). They also inferred a mantle composition of $x_{\text{FeO}}|_{\text{Mantle}} = 0.09^{+0.07}_{-0.06}$, $x_{\text{SiO}_2}|_{\text{Mantle}} = 0.39^{+0.05}_{-0.04}$ and $x_{\text{MgO}}|_{\text{Mantle}} = 0.51^{+0.06}_{-0.06}$, together with a core composition of $x_{\text{S}}|_{\text{Core}} = 0.27^{+0.16}_{-0.18}$. This also agrees well with the prediction results of the cINN. We note, however, that the good agreement in $x_{\text{S}}|_{\text{Core}}$ and $x_{\text{SiO}_2}|_{\text{Mantle}}$ may only be by chance here, as both the cINN and MCMC regard these two parameters as largely unidentifiable from the available observations (see Sect. 4.1). At the same time, the inference method used in [Mortier et al. \(2020\)](#) can merely constrain $x_{\text{S}}|_{\text{Core}}$ and $x_{\text{SiO}_2}|_{\text{Mantle}}$ beyond the constraints given by the used structure model.

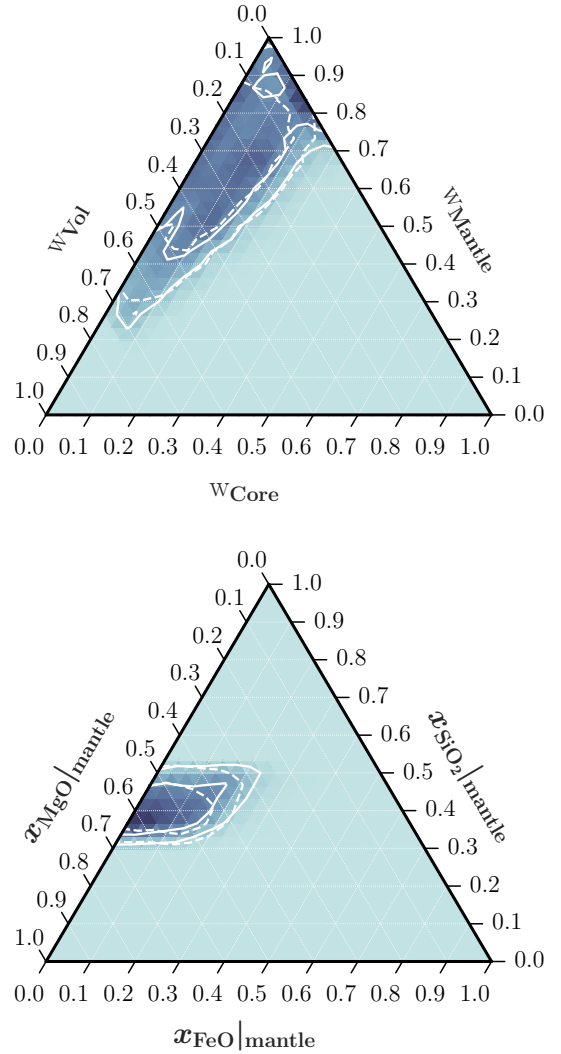


Fig. 14. Ternary diagrams of the cINN prediction. Top panel: kernel density estimate of the layer mass fractions as determined with the cINN. Bottom panel: kernel density estimate of the mantle composition. The kernel density in both panels was estimated using a Gaussian kernel with a standard deviation of 0.2. The white lines indicate the contours of the 68% HDR and 95% HDR. For comparison, the HDR from the posterior calculated with the MCMC is also shown (dashed lines).

The reason for the small difference in the mantle mass fraction and water mass fraction is likely that they used a four-layer model, including a H/He layer. Although K2-111 b has a very small H/He content in mass ($\sim 10^{-8} M_{\text{E}}$ as found in [Mortier et al. \(2020\)](#)), this small H/He layer can still contribute to radius by $\sim 0.1 R_{\text{E}}$, given the high equilibrium temperature of the planet ($T_{\text{eq}} = 1309 \text{ K}$). Thus, our results are expected to differ slightly from their study. Taking the difference in model setup into account, we conclude that our results agree well with the characterization performed by [Mortier et al. \(2020\)](#).

5.2. Comparison of the computational cost

One main motivation for using cINNs to infer planetary compositions is to reduce the time needed to perform a single inference. We provide here an overview over the encountered computational cost when using the two methods shown in this work, that is, an adaptive Metropolis Hastings MCMC method and the cINN method.

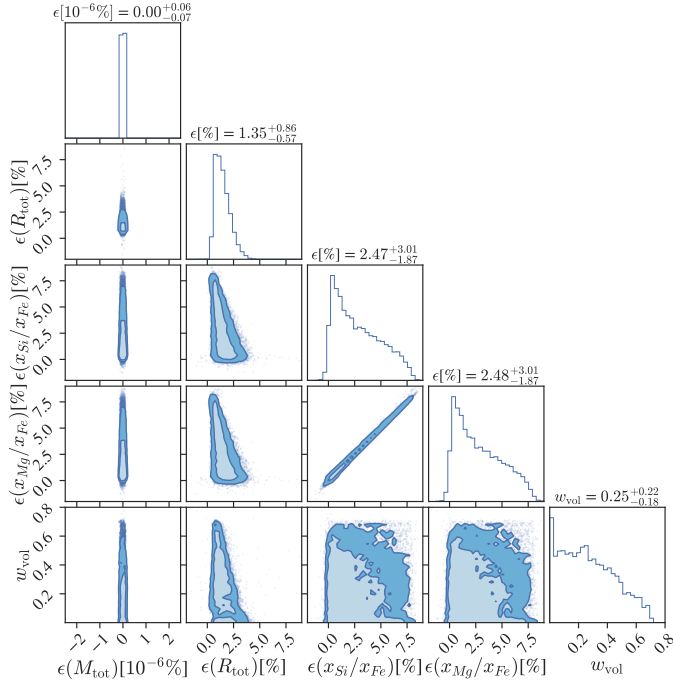


Fig. 15. Recalculations errors of the model output variables for the set of samples generated with the cINN for the case of K2-111 b.

The MCMC method is calculated sequentially, that is, for each step of the Markov chain, a forward model is run until a sufficient number of steps are generated. Hence, its computational cost scales linearly with the number of steps of the generated Markov chain. For the forward model we used, it was proven sufficient to generate on the order of 5×10^5 forward models. On a single core of an Intel Xeon Gold 6132 processor running at 2.6 GHz, one forward model takes 1.5 s to compute on average. Therefore, computing one planetary structure inference takes approximately 8.7 days in total on a single core.

Instead of computing a single Markov chain, it is also possible to initialize multiple chains in parallel or use an ensemble method such as in *emcee* (Foreman-Mackey et al. 2013). This parallelized approach allows leveraging the availability of multicore CPU architectures. For an efficient tuning of the proposal distribution, a minimum length of the Markov chain on the order of $\sim 10^4$ samples is required, however. Using 28 cores of an Intel Xeon Gold 6132 CPU, we had to compute 10^4 steps for each of the 28 Markov chains to account for burn-in and tuning of the proposal distribution. This results in a total of 7.8×10^5 forward models to be calculated by 28 parallel MCMC chains, which takes approximately 12 h. While the time for a single inference is only a fraction compared to a sequential MCMC run, the total computational cost is equivalent to 13.5 days of single threaded run time. The reason are the larger number of samples, which cannot be used for inference due to burn-in and tuning of the proposal distribution.

The total computational cost of the cINN method in contrary is split into three parts. The computation time needed to generate the training data, the time needed to train the cINN including the search for the hyperparameters for optimal training (i.e., determining learning rate, network architecture, etc.), and the time needed to sample the posterior using the cINN. For this project, the training data were generated calculating 5.9×10^6 forward models. With an average run time of 1.5 s, this would take 102 days to run on a single core, whereas using a compute node

Table 9. Overview over the computational cost for the two inference methods.

Method	Computing time	
	gen. TD & training	single inference
cINN ([‡])	105 days	5 min
MCMC ([†])	–	8.7 days
parallel MCMC ([†])	–	12 h

Notes. ([†])Run on a compute node containing 2×14 -Core Intel Xeon Gold 6132 CPUs at 2.6 GHz. ([‡])Run on a compute node containing 2×14 -Core Intel Xeon Gold 6132 CPUs at 2.6 GHz and $10 \times$ Nvidia Titan Xp GPUs at 1.6 GHz.

with 28 cores, the data set can be generated within 3.7 days. With the training data at hand, training the cINN itself takes between 2 and 3 h on a single GPU. The training and inference were performed on a compute node at the Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR) in Heidelberg, which consists of 2×14 -Core Intel Xeon Gold 6132 at 2.6 GHz and $10 \times$ Nvidia Titan Xp at 1.6 GHz, but only one GPU was used. Then a hyperparameter search is necessary to find the parameters for optimal training (i.e., learning rate, number of the neural network layers, network layer widths, etc.). For this study, we performed 23 trials to find the optimal parameters, thus repeating the training of the cINN 23 times. A single inference of the composition of an exoplanet using the trained cINN on the same GPU as mentioned above can be performed in 5 min. Performing all preparatory steps, that is, generating the training data, training, and hyperparameter search, takes approximately 105 days of sequential computing time. Leveraging multithreaded CPUs can reduce the runtime of all preparatory steps to 6 days.

When comparing the computational cost of the two methods, it is clear that for a single inference, the MCMC method is far cheaper given the large number of training data needed to train the cINN. When multiple inferences using the same forward model are to be performed, however, then generating the training data will contribute less to the total computational cost the more inferences are performed. Taking into account that the hyperparameter search in both does cases not need to be repeated (except when the MCMC method is used for very different data), then the cINN already becomes the more efficient method if the same forward model is used for more than ten planetary structure inferences. In Table 9 we show a summary of the computational cost of the two methods.

So far, we used a forward model without an additional atmosphere layer. Including an atmosphere in the forward model would add another two to three more input parameters and also make the forward model computationally more expensive. From experience with running structure inference models that include atmosphere layers, a number of 5×10^5 – 10^6 samples would be necessary to be generated for the Markov chain. We did not yet create a database of forward models including an atmosphere, hence we cannot conclude how this change would affect the computational cost. The time needed for inference should remain on the order of minutes for the cINN approach, however.

6. Conclusions

We discussed how INNs, in particular, the cINN, can be used to characterize the interior structure of exoplanets. So far, mainly MCMC methods were used to do this.

Compared to the cINN version initially proposed by [Ardizzone et al. \(2019b\)](#) for point estimates, we showed how the method can be adapted for noisy data. We validated this approach using a toy model, for which we compared the cINN performance against a regular Metropolis Hastings MCMC.

Then we applied the method to the exoplanet K2-111 b, inferring its composition. To do this, we trained a cINN on a simplified internal structure model for exoplanets and showed that in this case, cINNs also offer a computationally efficient alternative to the MCMC sampler that is commonly used for Bayesian inference.

In the benchmark of K2-111 b, only minor differences can be seen between the MCMC methods and the cINN method. The largest differences appeared in the marginalized posterior distribution of R_{vol} and w_{vol} . Computing the recalculation error of the benchmark case showed that the largest errors in total radius appeared for low values of w_{vol} . This agrees with the observed differences in the marginalized posterior distributions of w_{vol} and R_{vol} . Hence, it is likely that the difference between the two methods will become smaller if the training of the cINN can be further improved. Nevertheless, the two methods return very similar posterior distributions of the model parameters.

A key benefit of using cINNs over an MCMC method is that most of the computational cost of the method occurs during the generation of the training data and training, but not during the inference. This allows reducing the computational time spent for inference by almost four orders of magnitude compared to a regular MCMC method. In order to have an overall benefit in computational cost against the MCMC method used in this work, the cINN needs to be used to infer more than approximately ten planetary structures. While other authors successfully used neural networks to predict the output of their forward models (e.g., [Alibert & Venturini 2019](#); [Baumeister et al. 2020](#)), this work showed that it is also possible to train a neural network that encapsulates the full inverse problem.

Acknowledgements. We thank the referee Philipp Baumeister for the insightful comments which helped to improve the manuscript. J.H. and Y.A. acknowledge the support from the Swiss National Science Foundation under grant 200020_172746. V.K. and R.S.K. acknowledge financial support from the European Research Council via the ERC Synergy Grant “ECOGAL” (project ID 855130), from the Deutsche Forschungsgemeinschaft (DFG) via the Collaborative Research Center “The Milky Way System” (SFB 881 – funding ID 138713538 – subprojects A1, B1, B2 and B8), from the Heidelberg Cluster of Excellence (EXC 2181 – 390900948) “STRUCTURES”, funded by the German Excellence Strategy, and from the German Ministry for Economic Affairs and Climate Action in project “MAINN” (funding ID 50002206). They also thank for computing resources provided by the Ministry of Science, Research and the Arts of the State of Baden-Württemberg through bwHPC and DFG through grant INST 35/1134-1 FUGG and for data storage at SDS@hd through grant INST 35/1314-1 FUGG. For this publication, the following software packages have been used: [Python-matplotlib](#) by [Hunter \(2007\)](#), [Python-seaborn](#) by [Waskom et al. \(2021\)](#), [Python-corner](#) by [Foreman-Mackey \(2016\)](#), [Python-ternary](#) by [Harper et al. \(2019\)](#), [Python-numpy](#), [Python-pandas](#). The cINN implementation is based on the FrEIA framework available at <https://github.com/VLL-HD/FrEIA>.

References

Adibekyan, V., Dorn, C., Sousa, S. G., et al. 2021, *Science*, **374**, 330
 Agol, E., Dorn, C., Grimm, S. L., et al. 2021, *Planetary Science Journal*, **2**, 1
 Alibert, Y., & Venturini, J. 2019, *A&A*, **626**, A21
 Ardizzone, L., Kruse, J., Rother, C., & Köthe, U. 2019a, in *International Conference on Learning Representations*

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., & Köthe, U. 2019b, ArXiv [arXiv:1907.02392]
 Atkins, S., Valentine, A. P., Tackley, P. J., & Trampert, J. 2016, *Phys. Earth Planet. Interiors*, **257**, 171
 Baumeister, P., Padovan, S., Tosi, N., et al. 2020, *ApJ*, **889**, 42
 Benz, W., Ehrenreich, D., & Isaak, K. 2017, in *Handbook of Exoplanets*, eds. H. J. Deeg, & J. A. Belmonte (Cham: Springer International Publishing), 1
 Benz, W., Broeg, C., Fortier, A., et al. 2021, *Exp. Astron.*, **51**, 109
 Bishop, C. M. 1994, *Mixture Density Networks* (Birmingham: Aston University)
 Brown, J. M. 2018, *Fluid Phase Equilibria*, **463**, 18
 de Wit, R. W. L., Valentine, A. P., & Trampert, J. 2013, *Geophys. J. Int.*, **195**, 408
 Dinh, L., Sohl-Dickstein, J., & Bengio, S. 2016, ArXiv e-prints [arXiv:1605.08803]
 Doane, D. P. 1976, *Am. Statist.*, **30**, 181
 Dorn, C., Khan, A., Heng, K., et al. 2015, *A&A*, **577**, A83
 Dorn, C., Hinkel, N. R., & Venturini, J. 2017a, *A&A*, **597**, A38
 Dorn, C., Venturini, J., Khan, A., et al. 2017b, *A&A*, **597**, A37
 Fei, Y., Murphy, C., Shibasaki, Y., Shahar, A., & Huang, H. 2016, *Geophys. Res. Lett.*, **43**, 6837
 Feistel, R., & Wagner, W. 2006, *J. Phys. Chem. Ref. Data*, **35**, 1021
 Foreman-Mackey, D. 2016, *J. Open Source Softw.*, **1**, 24
 Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, *PASP*, **125**, 306
 French, M., & Redmer, R. 2015, *Phys. Rev. B*, **91**, 014308
 Gordon, S., & McBride, B. J. 1994, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. Part 1: Analysis*, Tech. rep., NASA Lewis Research Center
 Haario, H., Saksman, E., & Tamminen, J. 2001, *Bernoulli*, **7**, 223
 Hakim, K., Rivoldini, A., Van Hoolst, T., et al. 2018, *Icarus*, **313**, 61
 Haldemann, J., Alibert, Y., Mordasini, C., & Benz, W. 2020, *A&A*, **643**, A105
 Harper, M., Weinstein, B., Simon, C., et al. 2019, <https://doi.org/10.5281/zenodo.2628066>
 Hastings, W. K. 1970, *Biometrika*, **57**, 97
 Hellinger, E. 1909, *J. Reine Angew. Math.*, **1909**, 210
 Hoeijmakers, H. J., Ehrenreich, D., Kitzmann, D., et al. 2019, *A&A*, **627**, A165
 Hogg, D. W., & Foreman-Mackey, D. 2018, *ApJS*, **236**, 11
 Hunter, J. D. 2007, *Comput. Sci. Eng.*, **9**, 90
 Journaux, B., Brown, J. M., Pakhomova, A., et al. 2020, *J. Geophys. Res.: Planets*, **125**, e2019JE006176
 Kang, D. E., Pellegrini, E. W., Ardizzone, L., et al. 2022, *MNRAS*, **512**, 617
 Kingma, D. P., & Dhariwal, P. 2018, ArXiv e-prints [arXiv:1807.03039]
 Kippenhahn, R., Weigert, A., & Weiss, A. 2012, *Stellar Structure and Evolution*, 2nd edn., Astronomy and Astrophysics Library (Berlin Heidelberg: Springer-Verlag)
 Kroll, V. F., Ardizzone, L., Klessen, R., et al. 2020, *MNRAS*, **499**, 5447
 Lin, Q., Fouchez, D., Pasquet, J., et al. 2022, *A&A*, **662**, A36
 Madhusudhan, N. 2019, *ARA&A*, **57**, 617
 Mazevet, S., Licari, A., Chabrier, G., & Potekhin, A. Y. 2019, *A&A*, **621**, A128
 McBride, B. J., & Gordon, S. 1996, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications II. Users Manual and Program Description*, Tech. rep., NASA Lewis Research Center
 Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953, *J. Chem. Phys.*, **21**, 1087
 Mortier, A., Zapatero Osorio, M. R., Malavolta, L., et al. 2020, *MNRAS*, **499**, 5004
 Mosegaard, K., & Tarantola, A. 1995, *J. Geophys. Res.: Solid Earth*, **100**, 12431
 Plotnykov, M., & Valencia, D. 2020, *MNRAS*, **499**, 932
 Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1996, *Numerical Recipes in Fortran 90: Vol. 2, Volume 2 of Fortran Numerical Recipes: The Art of Parallel Scientific Computing* (Cambridge University Press)
 Rauer, H., & Heras, A. M. 2018, in *Handbook of Exoplanets*, eds. H. J. Deeg, & J. A. Belmonte (Cham: Springer International Publishing), 1309
 Rogers, L. A., & Seager, S. 2010, *ApJ*, **712**, 974
 Schulze, J. G., Wang, J., Johnson, J. A., et al. 2021, *Planet. Sci. J.*, **2**, 113
 Silverman, B. W. 1986, *Density estimation for Statistics and Data Analysis* (Chapman and Hall)
 Sotin, C., Gasset, O., & Mocquet, A. 2007, *Icarus*, **191**, 337
 Thiabaud, A., Marboeuf, U., Alibert, Y., Leya, I., & Mezger, K. 2015, *A&A*, **580**, A30
 Trotta, R. 2008, *Contemp. Phys.*, **49**, 71
 Wagner, W., & Pruß, A. 2002, *YJARS58457 J. Phys. Chem. Ref. Data*, **31**, 387
 Waskom, M., Gelbart, M., Botvinnik, O., et al. 2021, [mwaskom/seaborn/10.5281/zenodo.592845](https://doi.org/10.5281/zenodo.592845)