Master thesis : Exoplanet Orbital Characterization Using Simulation-Based Inference

Auteur : Ruth, Matteo
Promoteur(s) : Louppe, Gilles; Absil, Olivier
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil en science des données, à finalité spécialisée
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20393

University of Liège
School of Engineering and Computer Science

Exoplanet Orbital Characterization Using Simulation-Based Inference

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Data Science and Engineering

Author
Matteo Ruth

Supervisors
Pr. Olivier Absil
Pr. Gilles Louppe

Academic year 2023-2024

Abstract

This thesis aims at leveraging advances in deep learning, particularly simulation-based inference, to enhance the orbital parameter characterization of exoplanets. The current methods, like MCMC, are computationally expensive and slow to converge. Using Normalizing Flows and the expected forward Kullback-Leibler divergence as a loss function to train the model, we reproduced the results of the state-of-the-art method, α-DPI.

However, the non-amortized nature of this approach limited its generalizability, necessitating retraining for new datasets or additional observations of the exoplanet β-Pic b. To address these limitations, a generic model for exoplanet astrometry was developed using a ResMLP as an embedding network. Using different experiments, we showed that this generic model was able to infer the posterior of the orbital parameters of all four planets of the HR 8799 system, significantly reducing the computational effort compared to MCMC.

Despite these advancements, challenges remain, particularly in generalizing the model across exoplanets from different systems, as this generic model could not infer the posterior of the orbital parameters of β-Pic b.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my two supervisors, Olivier Absil and Gilles Louppe. Their unwavering guidance, insightful feedback, and continuous support throughout my thesis have been invaluable. Their expertise has significantly enhanced the quality of my work, and I am profoundly grateful for the opportunity to work on a project that aligns with my passion for astrophysics with my studies.

I extend my heartfelt thanks to my parents and Natalia for their support and encouragement throughout my studies. Their love, understanding, and belief in me have been a constant source of motivation, and I am immensely thankful for everything they have done to help me reach this point.

A special thanks to my friends, who have made my academic journey more enjoyable and memorable.

Contents

Chapter 1

Introduction

The detection and characterization of exoplanets through direct imaging are among the most exciting advancements in modern astronomy. By capturing a series of images of an exoplanet over time, astronomers can gain critical insights into the properties and dynamics of its planetary system. Analyzing an exoplanet's astrometric data enables us to understand its formation and evolution.

An exoplanet's orbit is defined by six Keplerian elements: semi-major axis, eccentricity, inclination angle, argument of periastron of the secondary's orbit, longitude of ascending node, and epoch of periastron passage. Additionally, parameters such as parallax and the total system mass are essential for fully characterizing the orbit.

The challenge in orbit fitting lies in estimating the posterior distribution of these parameters based on astrometric data of the exoplanet relative to its host star, derived from telescope images. This task is computationally intensive due to the high dimensionality of the parameters and the potential multi-modality of the posterior distributions. Traditional sampling-based approaches, such as Markov Chain Monte Carlo (MCMC) methods, are widely used for such inference problems. However, these methods can be prohibitively slow to converge without an optimal proposal distribution or a good random initialization.

Recent advances in deep learning, particularly using techniques like normalizing flows, offer promising alternatives to traditional methods. These approaches can significantly speed up the recovery of orbital parameters.

1.1 Problem Statement

This work aims to develop a novel approach for recovering the full posterior distribution of orbital parameters using the latest advancements in simulation-based inference. By leveraging modern techniques like normalizing flows, we seek to enhance the efficiency and accuracy of inferring these parameters from exoplanet imaging datasets.

The first part of this thesis focuses on reproducing the results of Sun *et al.* (2022) [1].

The second part aims to develop a more general and efficient model for exoplanet astrometry data. The goal is to create a model that can infer the full posterior distribution for any exoplanet, rather than requiring a separate model for each individual exoplanet. This involves designing and testing different neural network architectures, such as ResMLP[2]

and Deep Set[3] networks, to find an optimal solution for handling astrometric observations and inferring orbital parameters.

By achieving these objectives, this work contributes to the development of more practical and scalable tools for the astronomical community, potentially allowing for faster and more accurate characterization of exoplanets' orbital parameters.

Chapter 2

Exoplanet astrometry

The last thirty years have seen a revolution in the field of exoplanet detection. Since the first unambiguous detection of an exoplanet, 51 Pegasi b, discovered in 1995 [4], more than 5000 exoplanets have been discovered and confirmed. This number will continue to increase as new methods are developed and new telescopes are launched.

Several methods exist to detect exoplanets, with the most common being the radial velocity method, the transit method, the microlensing method, and the direct imaging method. All the exoplanets discovered have provided us with a better understanding of how common planetary systems like our own are and how they fit into the grand scheme of the universe. A wide diversity of exoplanetary systems has been found, exhibiting a range of masses, sizes, and orbits.

No single observational method can probe all of them. The radial velocity and transit methods are better suited to study close-in planets around mature stars, while the direct imaging method is more effective for studying planets in wide orbits around young stars. This is depicted in Figure 2.1, where exoplanets discovered by direct imaging are shown in the upper right corner of the plot, a region not well-probed by other methods.

Direct imaging stands out for its ability to capture actual photographs of exoplanets, allowing for direct measurement of their light and spectra. This method is used to study planets located far from their young host stars, and emits bright light in infrared wavelengths.

The characterization of exoplanet orbits is crucial for understanding their formation and evolutionary histories. By analyzing the orbits, scientists can infer the processes that shaped these planetary systems. The adaptative optics system and coronagraphic facility at the Very Large Telescope (VLT/SPHERE) [5] and the Roman mission that will be launched in 2027[1] are expected to revolutionize this field by providing more data using direct imaging. The GAIA mission[2] will also provide a long list of potential planets that will then need to be confirmed by direct imaging with ground-based telescopes. This will lead to a flood of high-quality data, necessitating the development of new methods to analyze this data quickly and efficiently.

[1] Roman website
[2] GAIA mission website

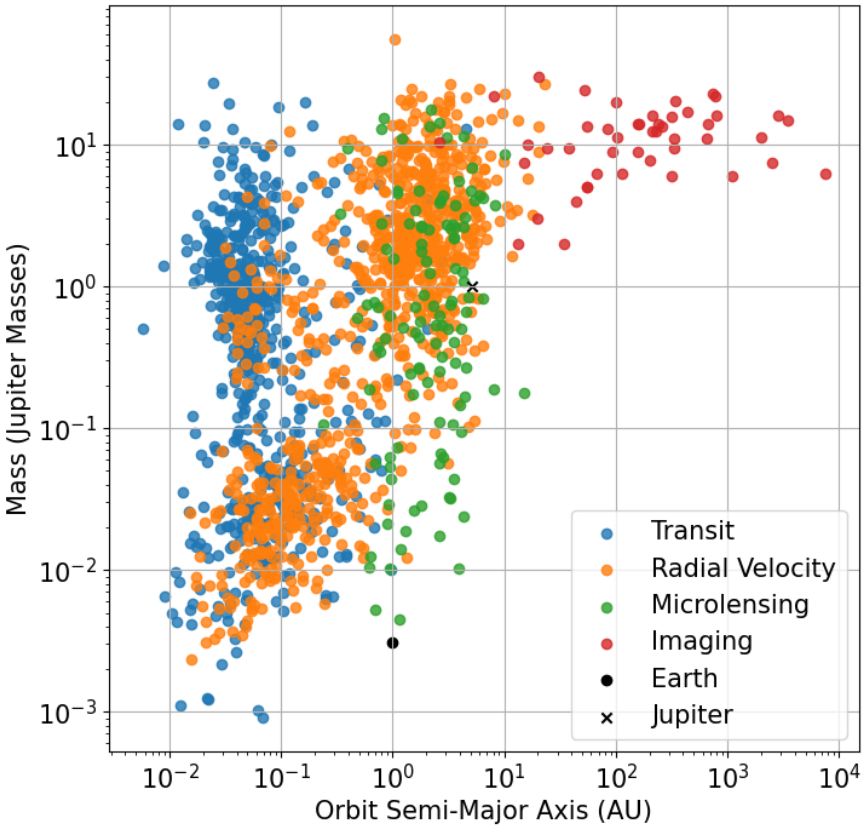

Figure 2.1. Scatter plot of exoplanets discovered with the different methods of detection. The x-axis represents the semi-major axis of the exoplanet's orbit and the y-axis represents the size of the exoplanet. The color of the points represents the method of detection. Plot adapted from Bowler *et al.* (2016) [6]. **Source :** NASA Exoplanet Archive

This thesis focuses on enhancing the methods for characterizing exoplanet orbits, particularly through direct imaging. With the anticipated influx of high-quality data from forthcoming missions, there is a pressing need to develop efficient analytical techniques. Traditional methods like Monte Carlo Markov Chain (MCMC) and Orbits for the Impatient (OFTI)[7] provide robust frameworks but often struggle with high-dimensional parameter spaces and multi-modal distributions or are computationally expensive. The newly proposed α-Deep Probabilistic Inference[1] aims to address these challenges by combining the strengths of variational inference and normalizing flows, offering a promising alternative for rapid and accurate orbit characterization. This approach leverages advanced computational techniques to streamline the analysis process, ensuring that we can keep pace with the ever-growing flood of exoplanetary data.

This chapter provides an overview of exoplanet astrometry, the current methods used to characterize exoplanet orbits and their limitations.

2.1 Direct Imaging

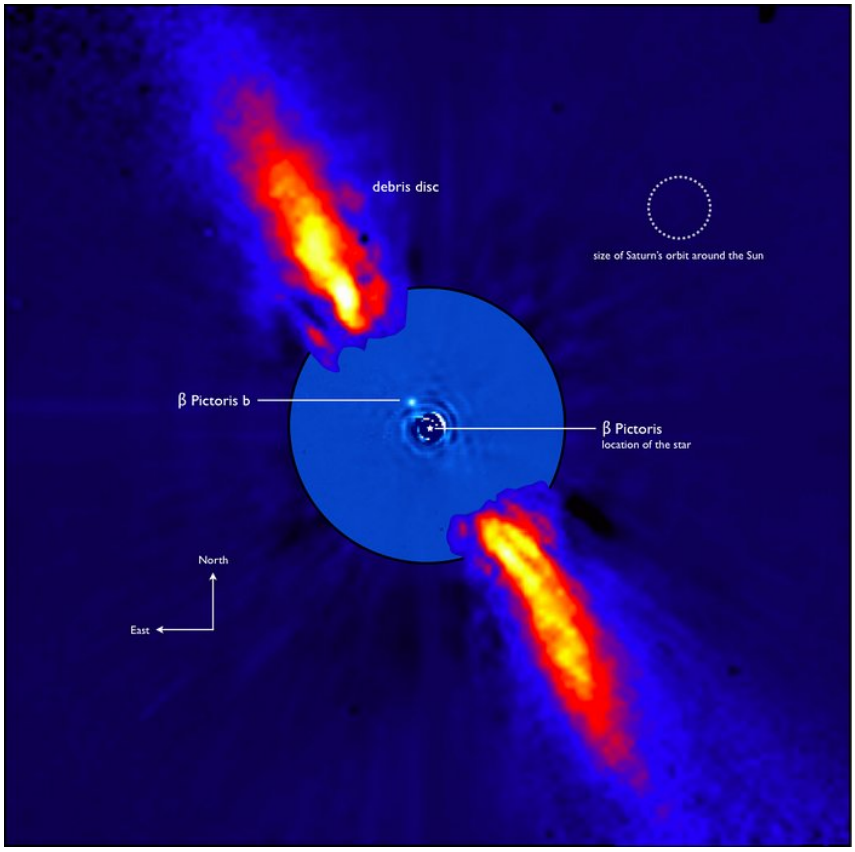

Figure 2.2. An example of direct imaging of an exoplanet. This image represents the stellar system β-Pictoris in near-infrared light. It is the composition of two images, both obtained by removing a large part of the halo of the star. The outer part of the image shows the reflected light on the dust disc, and in the inner part of the image, the infrared light of the exoplanet β-Pictoris b is visible. **credit :** ESO/A.-M. Lagrange *et al.*

Exoplanets reflect extremely little light from their host stars, making detection challenging due to the overwhelming brightness of the star's halo. The principle of direct imaging is to separate this halo from the exoplanet's light.

However, the actual light reflected by an Earth-like or even a Jupyter-like exoplanet is too faint to be detected directly even with the most advanced telescopes, ground-based or space-based.

Fortunately, young Jovian exoplanets radiate a significant amount of infrared light due to their high temperatures and large sizes. The infrared light emitted by the exoplanet is still faint, but much brighter than their reflected light. By using a coronagraph it is possible to suppress the light of the star and detect the infrared light of the exoplanet.

It works as follows, as the light of the star is detected by the telescope, small deformations are picked up as the light reflects off small imperfections in the telescope's mirror. A coronagraph is then used to block most of the light of the star. A lyot stop is then used to block the rest of the diffracted light of the star.

As the exoplanet is slightly offset from the star, the light of the exoplanet is not blocked

by the coronagraph which is centered on the star. However, even with all of that, the imperfections in the telescope's mirror still create a halo. This halo is then removed by using a deformable mirror that corrects the deformations of the telescope's mirror and the turbulence of the atmosphere picked up by the light of the star. After all of these steps and some post-processing, images like the one on the inner part of Figure 2.2 are produced.

This technique works thus better with space-based telescopes as there is no atmosphere to create turbulence. It also works better when the star is relatively close to the Sun, the exoplanet is large, hot and far enough from the star. [8]

The first generation of high-contrast imaging instruments did not provide sufficiently precise relative astrometry to derive accurate measurements. However, new instruments specifically designed for exoplanet imaging have been developed and now offer precise relative astrometry. As explained in the paper Bowler *et al.* [6] and illustrated in Figure 2.1, exoplanets detected by direct imaging typically have larger semi-major axes and longer periods compared to those detected by other methods. [9]

From these snapshots, we can derive the astrometry of the exoplanet, which is the position of the exoplanet in the sky.

One of the other advantages of direct imaging is that by passing the light through a prism, we can obtain the spectrum of the exoplanet and determine its composition.

2.2 Astrometry data

The astrometric data of the exoplanet can be expressed in two ways, the separation (SEP) and position angle (PA) or the right ascension (ΔRA) and declination (ΔDEC).

1. The separation and position angle are the distance in the sky between the exoplanet and its host star and the angle between the north and the exoplanet going towards the east, respectively. They are expressed in milliarcseconds (mas) and in degrees (°).
2. The right ascension and declination are the coordinates of the exoplanet in the sky relative to the star. It is expressed in milliarcseconds (mas) and this is the representation that is used in this thesis.

The ΔRA/ΔDEC data could be seen as cartesian coordinates and the SEP/PA data could be seen as polar coordinates. These representations are interchangeable and can be converted from one to the other using the following equations [10]:

$$\Delta\text{RA} = \text{SEP} \times \sin(\text{PA}), \tag{2.1}$$

$$\Delta\text{DEC} = \text{SEP} \times \cos(\text{PA}). \tag{2.2}$$

This conversion is useful when working with datasets that may have astrometric data represented in different formats from various sources.

The time used in the astrometric data is the Modified Julian Date (MJD) which is the number of days since the 17th of November 1858 at midnight. [11]

A typical dataset will contain the astrometric data of the exoplanet at different times and the uncertainties of the measurements. An example of such a dataset is shown in Annex A and Annex B for the exoplanet β-Pictoris b and HR 8799 bcde respectively.

2.3 Keplerian Elements

An orbit is parameterized by six Keplerian elements: the semi-major axis, the eccentricity, the inclination angle, the argument of periastron, the longitude of ascending node and the true anomaly at a certain time of observation. In addition to these six elements, we add the parallax and the total mass of the system as they influence the astrometric data we derived from the direct imaging, for example, a bigger mass of the system, which is mainly the mass of the star, will make the period of the orbit shorter. A schema of the keplerian elements is shown in Figure 2.3.

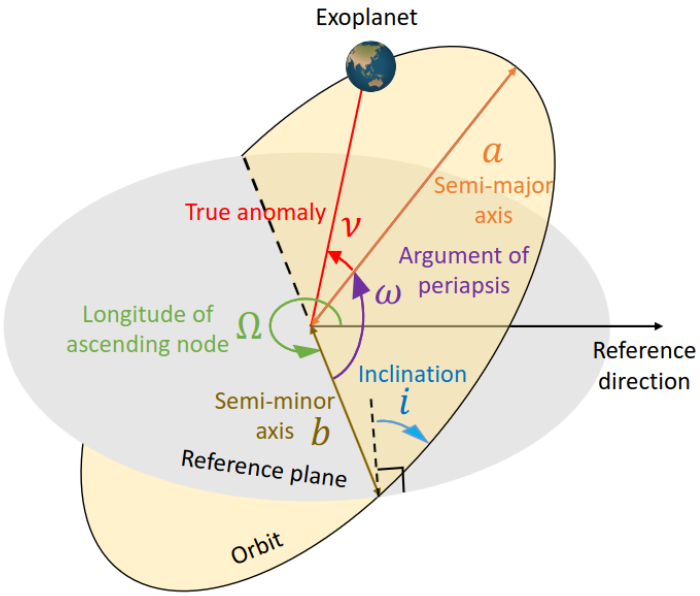

Figure 2.3. Keplerian elements of the orbit of an exoplanet around a star. The orbit is an ellipse characterized by the semi-major axis (a) and semi-minor axis (b). The eccentricity determines the shape of the ellipse, it depends on both the semi-major and semi-minor axes. The true anomaly (ν) is the angle between the periapsis (closest point of the orbit to the star) and the current position of the exoplanet at a given time. The ascending node is the point where the exoplanet crosses the reference plane from the south to the north. The argument of periapsis (ω), inclination (i), and the longitude of the ascending node (Ω) define the orientation of the orbit in space relative to the reference plane. This reference plane is the plane that is perpendicular to the line of sight from us to the star. The schema was taken from Sun *et al.* (2022) [1]

When looking at a frame centered on one of the two bodies, in this case, the star, the trajectory of the other body, the exoplanet, can be described by the eight orbital elements defined in Table 2.1.

Table 2.1. Explanation of the orbital elements of an exoplanet. [1, 7, 10, 12, 13]

Parameter	**Explanation**
Semi-major axis (a)	The sum of the *periapsis*, the maximum distance between the bodies, and *apoapsis*, the minimum distance between the bodies, distances divided by two. It is described in Astronomical Units (au), with 1 au being the distance between the Earth and the Sun.
Eccentricity (e)	Gives the shape of the ellipse, describing how much it is elongated compared to a circle with $e = 0$ being a circular orbit, $e < 1$ an elliptical orbit and $e = 1$ a parabolic trajectory. It depends on the semi-major axis and the semi-minor axis $e = \sqrt{1 - \frac{b^2}{a^2}}$
Inclination angle (i)	Vertical tilt of the ellipse with respect to the reference plane. It is described in degrees. An inclination of 0° would mean that the orbit plane is perpendicular to the line of sight and a 90° inclination would mean that the orbit plane is parallel to the line of sight.
Argument of periastron (ω)	The orientation of the ellipse in the orbital plane, as an angle measured from the ascending node, the point where the exoplanet crosses the reference plane from the south to the north, to the *periapsis*. It is described in degrees.
Longitude of ascending node (Ω)	The angle between the reference direction, the line of sight from us to the star, and the ascending node. It is described in degrees.
Epoch of periastron passage (τ)	Fraction of the orbital period past a reference epoch, bounded between 0 and 1. $\tau = \frac{t_p - t_{ref}}{P} \mod 1$, where t_p is the epoch of periastron, t_{ref} is the reference epoch, and P is the orbital period. In this work t_{ref} is set to 50.000 MJD, which corresponds to 10-10-1995. A value of 0 corresponds to the reference epoch, while 1 corresponds to the same position in the orbit as the reference epoch but one period later. This method of describing the epoch is useful since t_p can be difficult to constrain directly.
parallax (Π)	The apparent shift in the position of a star when observed from two different vantage points. This shift provides information about the distance of the star from an observer on Earth. It is described in milliarcseconds (mas).
Total mass (M_T)	The total mass of the system. It is the sum of the mass of the star and the exoplanet, and is described in solar masses $M_\odot$

2.4 Keplerian Orbits

With these parameters, we can describe the trajectory of the exoplanet using the Kepler equations[3], giving us [10, 14]

$$\Delta\text{RA} = \Pi a(1 - e\cos E)[\cos^2\frac{i}{2}\sin(\nu+\omega+\Omega) - \sin^2\frac{i}{2}\sin(\nu+\omega-\Omega)], \tag{2.3}$$

$$\Delta\text{DEC} = \Pi a(1 - e\cos E)[\cos^2\frac{i}{2}\cos(\nu+\omega+\Omega) + \sin^2\frac{i}{2}\cos(\nu+\omega-\Omega)], \tag{2.4}$$

with ν the true anomaly and E the eccentric anomaly:

$$E - e\sin E = 2\pi(\frac{t}{P} - (\tau - \tau_{ref})), \tag{2.5}$$

$$(\frac{P}{yr})^2 = (\frac{a}{au})^3(\frac{M_\odot}{M_{tot}}), \tag{2.6}$$

$$\nu = 2\tan^{-1}[\sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}]. \tag{2.7}$$

In Equation 2.5 the eccentric anomaly E is not retrievable analytically, so we need to use numerical methods to solve it. In this work, I will use the solver implemented in the `Orbitize!` library [10] which is actually two solvers, one for low eccentricity orbits and one for high eccentricity orbits. For eccentricities below 0.95, they use Newton's method with a tolerance of 10^{-9}. They explain that for eccentricities above 0.95, the number of iterations needed to converge increases significantly. They thus use the Mikkola solver [15].

2.5 Bayesian Inference

Astronomers have long been interested in characterizing the orbits of exoplanets. Accurate orbital parameters are crucial for several reasons: they help constrain the future positions of exoplanets, calculate the probability of transits, and could assess the climates and habitability of exo-planets that resemble Earth from future space imaging missions. [10]

Early methods focused on the orbital analyses of binary stars, which laid the groundwork for current exoplanet studies. [16] Some of these initial approaches involved grid searches over a limited number of parameters combined with linear least-squares fitting of the remaining parameters to map out χ^2 surfaces like in Hartkopf *et al.* (1989) [17] or used non-linear least-squares fitting to adjust all parameters simultaneously like in Forveille *et al.* (1999) [18]

Point estimates of orbital parameters are insufficient because they do not account for uncertainties or the full range of possible orbits. A more robust approach is to use Bayesian inference to estimate the posterior distribution of the orbital parameters.

[3]These equations to retrieve the relative Right Ascension and relative Declination come from the `Orbitize!` documentation

The goal of Bayesian inference is to update initial prior beliefs about the parameters θ given some observations $\mathbf{x}$ with the Bayes' theorem :

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}, \tag{2.8}$$

where $p(\theta)$ is the *prior* distribution of parameters, $p(\mathbf{x}|\theta)$ is the *likelihood* of the observed data given the parameters, $p(\mathbf{x})$, called the *evidence*, the distribution of the observed data marginalized over the parameters and $p(\theta|\mathbf{x})$ is the *posterior* distribution of the parameters given the observed data, which is the target distribution.

In the context of this work on exoplanet astrometry, the parameters θ are the Keplerian elements of the orbit of the exoplanet and the observations $\mathbf{x}$ are the astrometry data of the exoplanet relative to the star expressed in the right ascension and declination.

One of the major challenges in Bayesian inference is to compute the evidence $p(\mathbf{x})$ which is often intractable as it requires the integration over all the parameter space

$$p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta. \tag{2.9}$$

However, as this work is focused on parameter estimation, the evidence can be ignored since it is independent of the parameters. Bayes' theorem can then be rewritten as

$$p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta). \tag{2.10}$$

But even with this simplification, getting the posterior distribution remains a challenge when the likelihood is intractable.

To address this, statisticians and astronomers have developed methods to estimate the posterior distribution of these parameters, ensuring all potential scientific interpretations are considered and the uncertainty in the inference is properly quantified.[1]

The most common method used to estimate the posterior distribution of the orbital parameters is the Monte Carlo Markov Chain (MCMC) method.

2.6 State of The Art

2.6.1 Monte Carlo Markov Chain

The Monte Carlo Markov Chain (MCMC) method is used when direct sampling from the posterior distribution is not feasible. Instead, we sample from a Markov Chain whose stationary distribution approximates the target posterior distribution. This technique combines two fundamental concepts:

- **The Markov Chain :** A sequence of random variables where the probability of the next value depends only on the current value.
- **Monte Carlo Methods :** A group of algorithms that rely on repeated random sampling to estimate numerical results, particularly useful when deterministic solutions are too complex or time-consuming.

One of the simplest and most commonly used MCMC algorithms is the Metropolis-Hastings algorithm. The steps are as follows:

- let $f(\theta)$ be the target density function. In this case, it is the posterior distribution of the orbital parameters θ given the observed data $\mathbf{x}$. With Bayes' theorem we have

$$f(\theta) = p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}.$$

- let $q(\theta^{'}|\theta)$ be the proposed function used to generate $\theta^{'}$ when we are in θ. It needs to be easy to generate data from. It could be a Gaussian distribution centered around θ for example.
- let $a(\theta^{'}|\theta)$ be the acceptation function. It is the probability of accepting the next point $\theta^{'}$ and is defined as such

$$a(\theta^{'}|\theta) = \min(1, \frac{f(\theta')}{f(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}) \tag{2.11}$$

$$= \min(1, \frac{p(\mathbf{x}|\theta^{'})p(\theta^{'})}{p(\mathbf{x}|\theta)p(\theta)} \frac{q(\theta|\theta')}{q(\theta'|\theta)}). \tag{2.12}$$

In the case of exoplanet orbital characterization, the likelihood $\log p(\mathbf{x}|\theta)$ is chosen to be a Gaussian likelihood like in `Orbitize!`[10]:

$$\log p(\mathbf{x}|\theta) = -\frac{1}{2}\sum_{i}^{N} \frac{(\alpha_\theta(t_i) - \alpha_o(t_i))^2}{\sigma^2_{\alpha_o(t_i)}} - \frac{1}{2}\sum_{i}^{N} \frac{(\delta_\theta(t_i) - \delta_o(t_i))^2}{\sigma^2_{\delta_o(t_i)}}, \tag{2.13}$$

with α and δ being the offset in right ascension and declination of the exoplanet, N the number of observations, $\cdot_\theta(t_i)$ the model prediction at time t_i, $\cdot_o(t_i)$ the observed data at time t_i and $\sigma_{\cdot_o}$ the observational uncertainty of the observed data. α_θ and δ_θ are computed by solving the Kepler equations for the given parameters θ and the time t_i. This is done by solving Equations 2.3 and 2.4 using the solver from `Orbitize!` [10].

The algorithm then works as described in Algorithm 1.

I used the `Orbitize!` library, which implements MCMC using the `emcee` [19] and `ptemcee` [20] libraries. These libraries use variants of the Metropolis-Hastings algorithm that run multiple chains simultaneously, known as *walkers*. The proposal distribution $q(\theta'|\theta)$ of one chain depends on the current position of all other chains, increasing efficiency compared to the original Metropolis-Hastings algorithm. Convergence is determined by examining the autocorrelation time [19] and trace plots of the chains. If those chains are in different parts of the parameter space, it means that they did not converge.

A major drawback of MCMC is its slow convergence, especially when the posterior distribution is multi-modal, which is common in exoplanet orbital characterization. This challenge has prompted the development of new methods.

2.6.2 Orbit for the impatient

One of these new methods is the Orbit For The Impatient (OFTI) [7]. It is an Approximate Bayesian Computation (ABC) method as described in Cranmer et al. (2020) [21] and as

Algorithm 1 The Metropolis-Hastings algorithm

$\mathbf{x_{obs}}$: the observed data
for $i = 1, ..., \mathcal{T}$ **do**
    Draw a proposal for step t : $\theta_t^{'} \sim q(\theta_t^{'}|\theta_{t-1})$
    $a_t \leftarrow \min(1, \frac{p(\mathbf{x_{obs}}|\theta_t^{'})p(\theta_t^{'})}{p(\mathbf{x_{obs}}|\theta_{t-1})p(\theta)} \frac{q(\theta_{t-1}|\theta_t^{'})}{q(\theta_t^{'}|\theta_{t-1})})$
    Draw r_t : $r_t \sim \mathcal{U}[0, 1]$
    if $r_t < a_t$ **then**
        $\theta_t \leftarrow \theta_t^{'}$
    else
        $\theta_t \leftarrow \theta_{t-1}$
$\Theta = \{\theta_{T+1}, ..., \theta_{\mathcal{T}}\}$ — Discard the first T samples to ensure the Markov Chain has converged
return Θ

described in that paper there are some limitations to this method. Mainly, in our case, two of them are that this method would not scale with a lot of data points, and with new data points, the whole algorithm needs to be rerun which is not efficient.

So this method works well when the parameter space is relatively unconstrained, which is typically the case when the observed data only cover a relatively small arc of the total orbit. This makes sense as most exoplanets detected through direct imaging have larger orbits [6] meaning that the observed data over the last twenty years only cover a small fraction of the total orbit. But, for exoplanets where the measurements cover a larger fraction of the orbit, the OFTI algorithm becomes less efficient and the MCMC algorithm is more suited, which is once again highly time-consuming. This can be seen in Figure 6. of the OFTI paper [7] where the OFTI algorithm is compared to the MCMC algorithm with an increasing number of observations of β-Pictoris b.

The parameters are the orbital elements semi-major axis(a), eccentricity(e), inclination angle(i), argument of periastron(ω), longitude of ascending node(Ω), the epoch of periastron passage(τ) which are explained in Table 2.1.

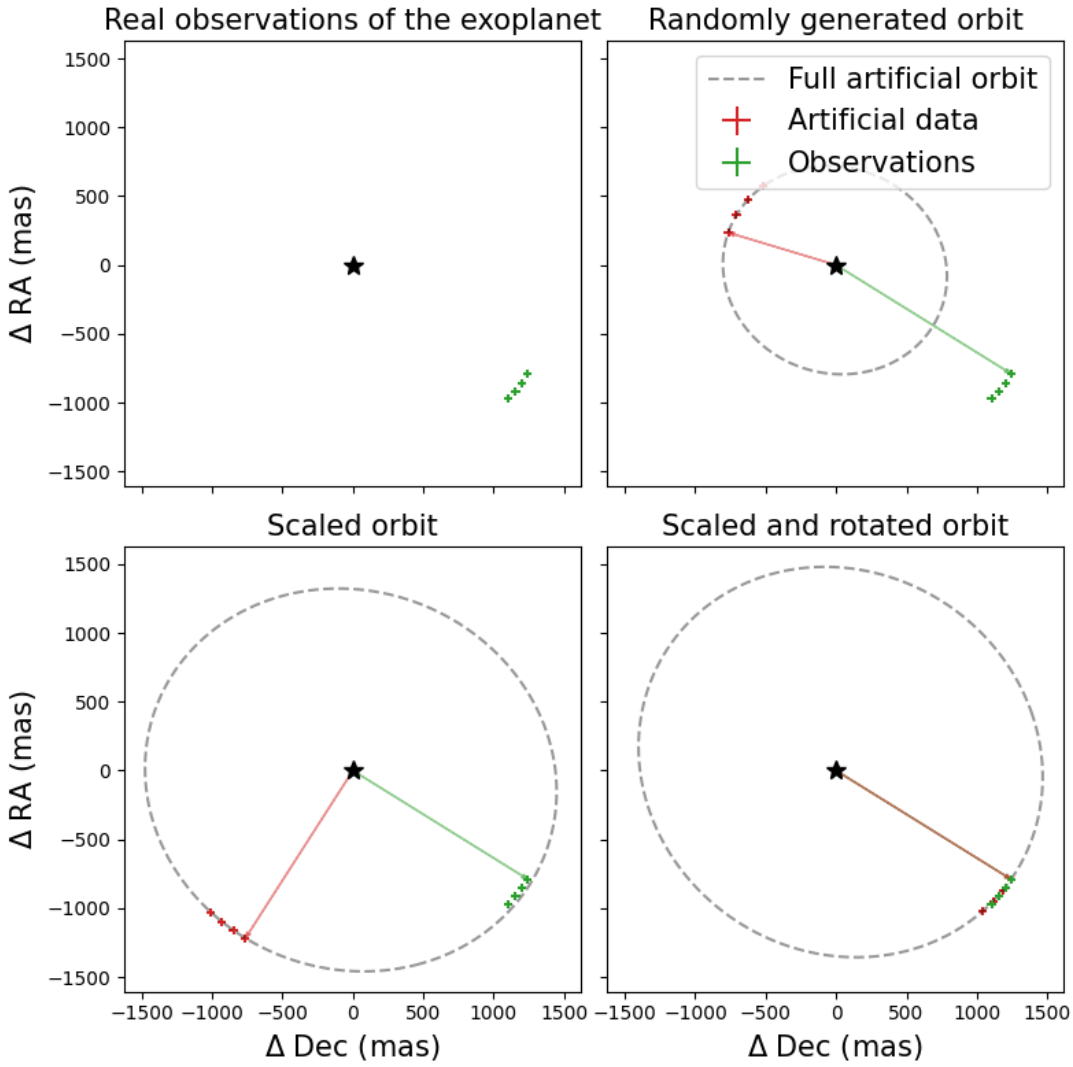

Figure 2.4. The first two steps of the OFTI algorithm. On the first subplot, 4 observations can be seen in green. The black star at the origin represents the star of the system. The second subplot shows the first step of the algorithm, the generation of a random orbit by sampling the priors for each parameter. The red points correspond to the observations at the same time as the green observations, the red and green arrows point to the first observation which is taken as a reference point for the algorithm. The third subplot shows how the semi-major axis is scaled and the fourth subplot shows the rotation of this orbit so that the two observations at the chosen reference timestep are aligned. After that, the chi-squared probability is calculated on the other observations to reject the orbits where the probability is too low. This whole process is repeated until a certain number of orbits are kept.

The OFTI algorithm works in three steps. Figure 2.4 shows the first two steps.

Monte Carlo Orbit Generation from Priors

The first step is to generate random samples of parameters from the prior distribution

$$\theta \sim p(\theta) = p(a, e, i, \omega, \Omega, \tau). \tag{2.14}$$

Even though the prior used in the paper was described as uniform for all parameters except for the eccentricity and the inclination angle, which had respectively a linearly descending prior and a sine prior, the implementation of the OFTI algorithm in `Orbitize!` permits the user to choose the prior distribution of the parameters. An orbit is then generated using these parameters.

Scale-and-Rotate

To restrict the wide parameter space of all possible orbits, the generated semi-major axis a is scaled and the position angle of nodes is rotated so that the produced orbit goes through a single astrometric data point. The choice of this astrometric data point is arbitrarily, OFTI uses an initial round where they find the data point that will result in the highest acceptance rate of orbits during the last phase.

They do not explain in the paper how they perform this initial round to choose the astrometric data nor how they scale and rotate the generated orbit. However, by looking at the source code, they choose the data point that corresponds to the smallest astrometric error. They then scale the semi-major axis by multiplying it by the ratio of the generated separation to the observed separation at the chosen data point and they rotate the position angle of nodes by the difference between the observed position angle and the generated position angle at the chosen data point.

Rejection sampling

Using the scaled and rotated parameters, the algorithm then calculates the other astrometric data points for all the other epochs and calculates the chi-squared probability of the predicted astrometry given the measured astrometry and uncertainties. The orbits where this probability is larger than a number sampled from a uniform distribution are kept. This process is repeated until a certain number of orbits are kept. This number is arbitrary chosen.

2.6.3 Alpha-Deep Probabilistic Inference

In the paper of Sun *et al.* α-Deep Probabilistic Inference [1], the authors propose a new method to estimate the posterior distribution of the parameters. They explain that sampling methods such as MCMC and OFTI are slow for the context of exoplanet orbital characterization because of the curse of dimensionality and they explain that variational inference methods may not be well suited for this task as they may lack estimation accuracy. They decide to use a method that tries to combine the best of both worlds.

The method is composed of two steps :

α-divergence Variational Inference with Normalizing flows

The goal of Variational Inference is to solve an optimization problem to estimate a posterior distribution. We try to find the parameters ϕ^* that best match the variational density function to the target posterior distribution.

$$\phi^* = \arg\min_{\phi} D[q_\phi(\theta|\mathbf{x})||p(\theta|\mathbf{x})], \tag{2.15}$$

where D is a divergence measure between the two density functions. The Kullback-Leibler divergence [22] is often used but they claim that it may not be well suited for this task. They explain that in theory, the KL-divergence should give a density function similar to the target distribution, but in practice it often pushes the result too much towards certain areas, ignoring the less likely ones. This is known as the mode-seeking effort of the reverse KL-divergence [23]. To solve this problem they use the Renyi's α-divergence instead [24] :

$$\phi^* = \arg\min_{\phi} D_\alpha[q_\phi(\theta)||p(\theta|\mathbf{x})] \tag{2.16}$$

$$= \arg\min_{\phi} \frac{1}{\alpha - 1} \log \mathbb{E}_{\theta \sim q_\phi(\theta)}\{\exp[(1-\alpha)(\log p(\mathbf{x}|\theta) + \log p(\theta) - \log q_\phi(\theta))]\}. \tag{2.17}$$

When $\alpha \to 1$, it approaches the Kullback-Leibler divergence, and when $\alpha = 0$ it corresponds to the Maximum Likelihood Estimation (MLE) of the parameters ϕ. The MLE is computationally efficient but may not capture the true posterior distribution accurately. In contrast, the KL-divergence is more precise in capturing the posterior distribution but at a higher computational cost. Therefore, they choose to tune α to find an optimal balance between the two.

Like for OFTI and MCMC, the likelihood function is assumed to be a Gaussian likelihood written in Equation 2.13.

For the prior, the paper is misleading as they claim to use uniform priors for all parameters except on the semi-major axis where they use a log-uniform prior, and the parallax and total mass where they use a Gaussian prior. Except for the parallax and total mass, the interval of the prior distributions are huge and make it seem as if this method is able to find the posterior distribution quickly in an hour on a GTX 1080 Ti. This appears to be a mistake in the paper as the values they give for the parallax and total mass do not correspond to the known values of the system they are studying, β-Pictoris. After looking at the source code, the priors are much more constrained. These priors are explained later in this work, in Table 4.1.

For $q(\theta)$ they use a normalizing flow. They decided to go with a Real-NVP network [25]. It uses simple affine transformations to transform a base normal distribution into a more complex distribution. These transformations are simple and computationally efficient, but they lack expressiveness and a lot of transformations need to be stacked to represent multi-modal or discontinuous densities [26, 27]. For example, in the paper, they used 32 transformations, each composed of a neural network with 3 fully connected layers. Each layer has a size of 128 neurons. Normalizing flows will be discussed more thoroughly in Section 3.1

They also implemented an annealed version of the α-divergence to help the optimization process. The log-likelihood and log prior in Equation 2.17 gives values that are a lot higher than $\log q_\phi(\theta)$ at the beginning of the training process which makes the optimization process difficult. They thus add a temperature parameter β_i for epoch i that is decreased over time. β_i is given by :

$$\beta_i = \max(1, \beta_0 e^{-i/\tau}). \tag{2.18}$$

They chose $\beta_0 = 10^4$ and $\tau = 3000$.

The annealed α-divergence at epoch i is then given by :

$$\arg\min_{\phi} \frac{1}{\alpha - 1} \log \mathbb{E}_{\theta \sim q_\phi(\theta)} \{\exp[(1-\alpha)(\frac{1}{\beta_i} \log p(\mathbf{x}|\theta) + \frac{1}{\beta_i} \log p(\theta) - \log q_\phi(\theta))]\}. \quad (2.19)$$

Importance Sampling

This is used in Monte Carlo methods to estimate the expected value of a function. The idea is to sample from a distribution that is easy to sample from, like a Gaussian distribution :

$$\theta \sim k(\theta), \quad (2.20)$$

and then reweight the samples to approximate the distribution we are interested in. The weights are the ratio of the target distribution to the proposal distribution.

$$w_i = \frac{p(\theta_i|\mathbf{x})}{k(\theta_i)} \propto \frac{p(\mathbf{x}|\theta_i)p(\mathbf{x})}{k(\theta_i)}. \quad (2.21)$$

They explain that because a normalizing flow is a bijective function, to generate disconnected nodes in the posterior distribution they need to include low-probability regions that interconnects the disconnected nodes.

By choosing $k(\theta)$ to be the trained normalizing flow q_ϕ, most of the weights would be close to 1. They then remove the samples with the lowest weights. They claim that this produces a cleaner posterior distribution.

Chapter 3

Simulation-based Inference

Simulation-based inference is a class of algorithms that overcomes the challenges of traditional inference methods by using simulators and deep neural networks to parameterize density estimators to estimate the target posterior.

For example, one of these algorithms, called Neural Posterior Estimation, trains a conditional density estimator $q_\phi(\theta|\mathbf{x})$ to approximate the posterior distribution $p(\theta|\mathbf{x})$ by finding the parameters ϕ^* that minimizes the expected Kullback-Leibler divergence[22] over all possible data points, meaning that the estimator matches the target posterior distribution as much as possible:

$$\phi^* = \arg\min_\phi \mathbb{E}_{p(\mathbf{x})} D_{KL}[p(\theta|\mathbf{x})||q_\phi(\theta|\mathbf{x})]. \tag{3.1}$$

Note that we use the expected **forward** Kullback-Leibler divergence and not the **reverse** Kullback-Leibler divergence like in the α-DPI algorithm (Equation 2.15). The paper of Papamakarios *et al.* (2021) [28] explain how the two are equivalent. The forward Kullback-Leibler divergence is more interesting in our case as it allows us to apply a simple trick to rewrite the optimization problem in a more tractable form. [29]

Rewriting the Kullback-Leibler divergence, we get

$$\phi^* = \arg\min_\phi \mathbb{E}_{p(\mathbf{x})} \mathbb{E}_{p(\theta|\mathbf{x})} [\log p(\theta|\mathbf{x}) - \log q_\phi(\theta|\mathbf{x})]. \tag{3.2}$$

The double expectation over the data distribution and over the posterior distribution can be rewritten as $\mathbb{E}_{p(\mathbf{x},\theta)}$, the expectation over the joint distribution of the data and the parameters. Because for any given data point $\mathbf{x}$, there corresponds one specific value of θ from the joint distribution, Equation 3.2 can then be rewritten as

$$\phi^* = \arg\min_\phi \mathbb{E}_{p(\mathbf{x},\theta)} [-\log q_\phi(\theta|\mathbf{x})]. \tag{3.3}$$

With Equation 3.3, the neural network that parameterizes the conditional density estimator can be trained by minimizing the negative log-likelihood of the parameters given the data.

To model this conditional density estimator, the first approach could be to train a neural network taking the data as input and outputting the parameters of a Gaussian distribution or of a mixture of Gaussians. However, such a model may struggle to capture the complexity of the true posterior distribution. The hypothesis space of the neural network choosing to represent the posterior as a mixture of Gaussians may be too restrictive. Normalizing flows are actually more suited for this task of approximating complex posterior distributions. [28]

3.1 Normalizing Flows

The idea behind normalizing flows is to build complex probability distributions by modifying a simple one sequentially.

This function $f : Z \to X$ maps a simple distribution $\mathbf{z} \sim \pi(\mathbf{z})$, for example a multivariate normal distribution $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, to a complex distribution $q(\mathbf{x})$.

This f is composed of a total of K transformations $f_1, ..., f_K$ that are all invertible and differentiable functions. Each of these transformations f_k are implemented as invertible neural networks. The whole transformation is denoted as

$$\mathbf{x} = f(\mathbf{z}) = f_1 \circ ... \circ f_K(\mathbf{z}), \tag{3.4}$$

because a composition of invertible functions is also invertible, the inverse of the transformation can be computed as

$$\mathbf{z} = f^{-1}(\mathbf{x}) = f_K^{-1} \circ ... \circ f_1^{-1}(\mathbf{x}). \tag{3.5}$$

Using a change of variable formula, the probability density function of the complex distribution can be expressed as

$$q(\mathbf{x}) = \pi(f^{-1}(\mathbf{x}))|\det(\frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}})|. \tag{3.6}$$

The function f can be seen as compressing and expanding the density of the simple distribution $\pi(\mathbf{z})$ and the Jacobian determinant of the function ensures that the total probability mass is conserved.

The Jacobian determinant of this inverse transformation is then

$$|\det(\frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}})| = \prod_{k=1}^{K} |\det(\frac{\partial f_k^{-1}(\mathbf{z}_k)}{\partial \mathbf{x}})|, \tag{3.7}$$

with $\mathbf{z}_k = f_k^{-1}(\mathbf{z}_{k-1})$ and $\mathbf{z}_0 = \mathbf{x}$.

Normalizing flows can also be used to estimate conditional densities by taking the data as additional input, meaning we can use them to estimate the posterior distribution of the parameters given the data and train the model minimizing Equation 3.3. [25–28]

Coupling layer

There is a sequence of potentially high dimensional Jacobian determinants to be computed which can be computationally expensive. One way to solve this issue is to ensure that this Jacobian is lower triangular, as its determinant is then the product of its diagonal elements. A coupling layer produces a lower triangular Jacobian by following these steps:

1. The input of the k^{th} transformation $\mathbf{z}^k$ is split into two parts $\mathbf{z}^k = [\mathbf{z}^k_{1:d-1}, \mathbf{z}^k_{d:D}]$.
2. The first part $\mathbf{z}^k_{1:d-1}$ is used as input of a neural network. It outputs the parameter(s) ϕ.
3. The second part $\mathbf{z}^i_{d:D}$ is transformed by a function g^i_ϕ depending on the parameters ϕ.
4. Return the concatenation of the first part $\mathbf{z}^k_{1:d-1}$ that was unchanged and the transformed second part $g^k_\phi(\mathbf{z}^k_{d:D})$

The transformation is then :

$$\mathbf{z}^{k+1}_{1:d-1} = \mathbf{z}^k_{1:d-1}, \tag{3.8}$$
$$\mathbf{z}^{k+1}_{d:D} = g^i_\phi(\mathbf{z}^k_{d:D}). \tag{3.9}$$

The Jacobian of this transformation is then

$$\frac{\partial \mathbf{z}^{k+1}}{\partial \mathbf{z}^k} = \begin{bmatrix} I_{d-1} & 0 \\ \frac{\partial \mathbf{z}^{k+1}_{d:D}}{\partial \mathbf{z}^k_{1:d-1}} & \frac{\partial \mathbf{z}^{k+1}_{d:D}}{\partial \mathbf{z}^k_{d:D}} \end{bmatrix}. \tag{3.10}$$

This needs to be a lower triangular matrix, $\frac{\partial \mathbf{z}^{k+1}_{d:D}}{\partial \mathbf{z}^k_{d:D}}$ needs to also be lower triangular, which is the case as $\mathbf{z}^{k+1}_{d:D}$ is depending only on $\mathbf{z}^k_{1:d-1}$. At each transformation, the inputs are permuted to ensure that all the dimensions are transformed at least once.

For example g^k_ϕ could be an affine transformation :

$$g^i_\phi(x) = \alpha x + \beta, \tag{3.11}$$

where α and β are the output of the neural network taking $\mathbf{z}^i_{1:d-1}$ as input.

Such coupling layers are easy to invert but may not be expressive enough to model complex, discontinuous, multi-modal distributions. [27]

Neural Spline Flow

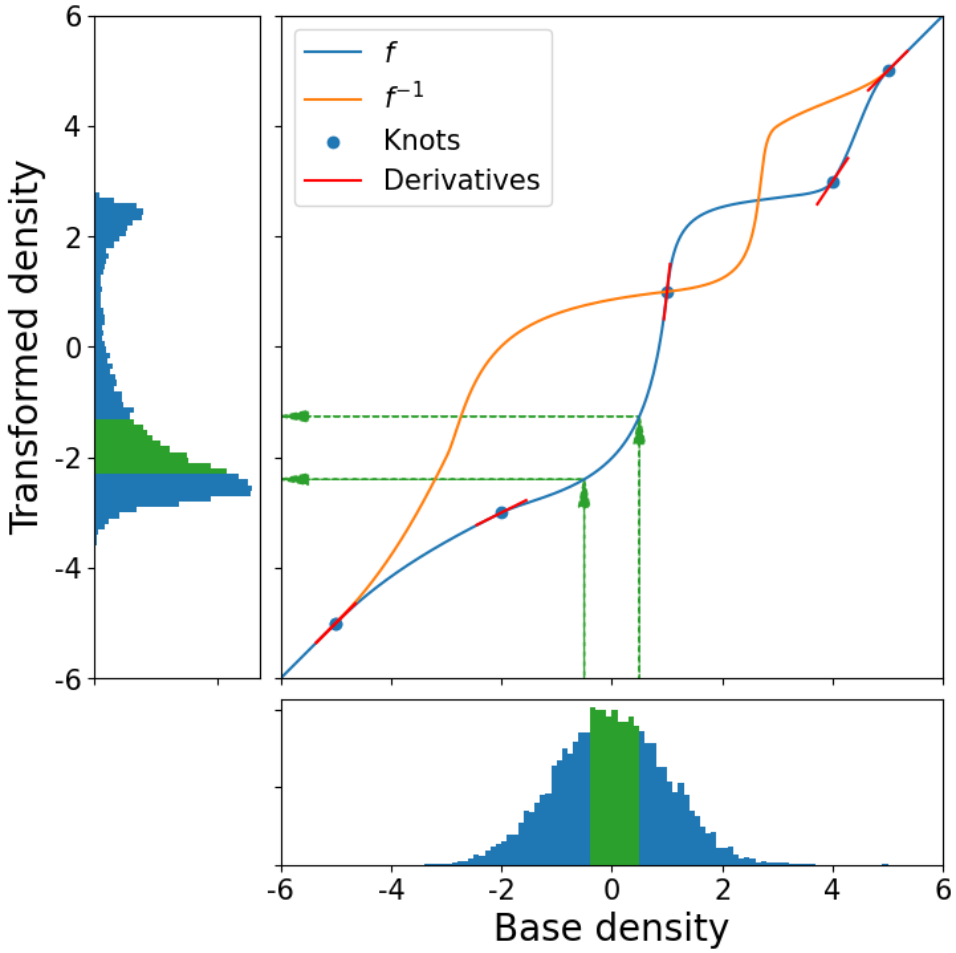

Figure 3.1. Example of a transformation of a normal distribution to a more complex distribution using a Neural Spline Flow. In this example, there are 5 knots in total. The first and last knots are always fixed at position $[-B, -B]$ and $[B, B]$, where B is 5, with a derivative of 1. The positions of the other knots and their derivatives are given by the neural network given the context. Outside of the interval $[-B, B]$, the transformation is the identity, meaning that the distribution will not be modified.

In the context of this work, it was decided to use Neural Spline Flow [27] as it is more flexible while still being differentiable and easy to invert. They model g_ϕ as a monotonic rational-quadratic spline on an interval and the identity function otherwise.

One transformation can be seen in Figure 3.1. The transformation is defined by a set of K knots $[a_k, b_k]$ and $K - 2$ derivatives . The first and last knots are always fixed at position $[-B, -B]$ and $[B, B]$, with their derivative set to 1. To ensure that the transformation is monotonic, the neural network produces the $K - 1$ widths and heights of bins. Those are then passed through a softmax function and multiplied by $2 \times B$ to ensure that the bins are in the interval $[-B, B]$. The cumulative sum of the widths and heights of the bins are then the positions of the knots.

To sample from the complex distribution, it is enough to sample from the simple tractable distribution, like the Gaussian depicted below the x-axis on Figure 3.1, see in which

interval the sample falls and apply the corresponding spline transformation to it, giving a sample from the complex distribution depicted to the left of the y-axis.

Because this transformation is invertible, going from the other way around is also possible and this is how the training is done. By taking Equation 3.3, we know we want to maximize the log-likelihood. What we maximize is the log-likelihood of the data points from the complex distribution passing through the inverse transformation meaning the neural network learns to transform the complex distribution into a normal distribution, hence the name *normalizing* flow.

Multiple of these transformations can be stacked to build a more complex distribution, where each output of one transformation is the input of the next one. [27]

3.2 Diagnosis

One of the challenges of Bayesian inference is diagnosing the quality of the posterior distribution approximation. A useful method for this is the calibration test proposed by Hermans *et al.* (2022) [30].

This calibration test is based on the idea of expected coverage. Let's take a normalizing flow that approximates the posterior distribution $q_\phi(\theta|\mathbf{x})$. We generate N samples $[\theta_1, ..., \theta_N]$ and their corresponding data points $[\mathbf{x}_1, ..., \mathbf{x}_N]$.

As defined in Hermans *et al.* (2002), the expected coverage probability is

$$\mathbb{E}_{p(\theta,\mathbf{x})}[\mathcal{I}(\theta \in \Theta_{p_\phi(\theta|\mathbf{x})})(1-\alpha)], \tag{3.12}$$

with $\mathcal{I}$ the indicator function that is equal to 1 if the condition is true and 0 otherwise, and where the function $\Theta_{p_\phi(\theta|\mathbf{x})})(1-\alpha)$ yields the $1-\alpha$ highest posterior density region of $p_\phi(\theta|\mathbf{x})$.

If our estimation of the posterior distribution using the Neural Spline Flow is well-calibrated, the parameters $[\theta_1, ..., \theta_N]$ should lie within the $1-\alpha$ credible region exactly $1-\alpha\%$ of the time.

We can visualize this by plotting the corner plot of the posterior distribution of the generated artificial data points, as shown in Figure 3.2. This plot is for one set of parameters $\theta \in [\theta_1, ..., \theta_N]$ and its corresponding data point $\mathbf{x}^*$. While the figure shows only three credible regions, theoretically, there are infinitely many.

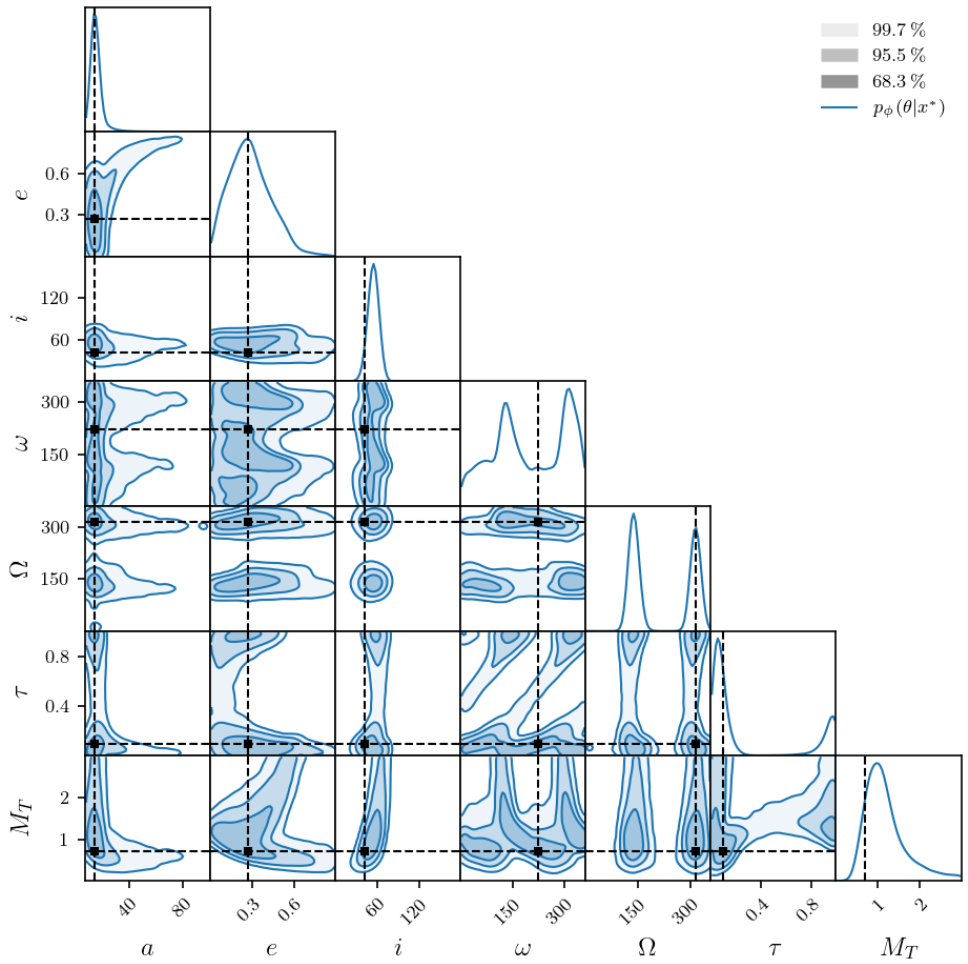

Figure 3.2. Corner plot of the posterior distribution of the artificial data points. It shows the 1D marginal distribution of each parameter on the diagonal and the 2D marginal distribution of each pair of parameters on the off-diagonal. Three different credible regions are shown, the 68.3%, 95.5%, and 99.7% credible regions. The true values of the parameters are shown as the black lines.

For each of the N samples, we compute the credible region of the parameters. The true parameters should fall within each of the $1-\alpha$ credible regions $1-\alpha$% of the time. Thus, if we plotted the corner plot for each of the N samples, the true parameter values would appear in the 68.3% credible region 68.3% of the time if the model is well-calibrated. We can plot the $1-\alpha$ line and observe how the coverage of the credible regions evolves. If the curve is below the $1-\alpha$ line, the model is overconfident, meaning the credible regions are too narrow, excluding some true parameters. If the curve is above the $1-\alpha$ line, the model is underconfident, with overly wide credible regions. This is illustrated in Figure 3.3.

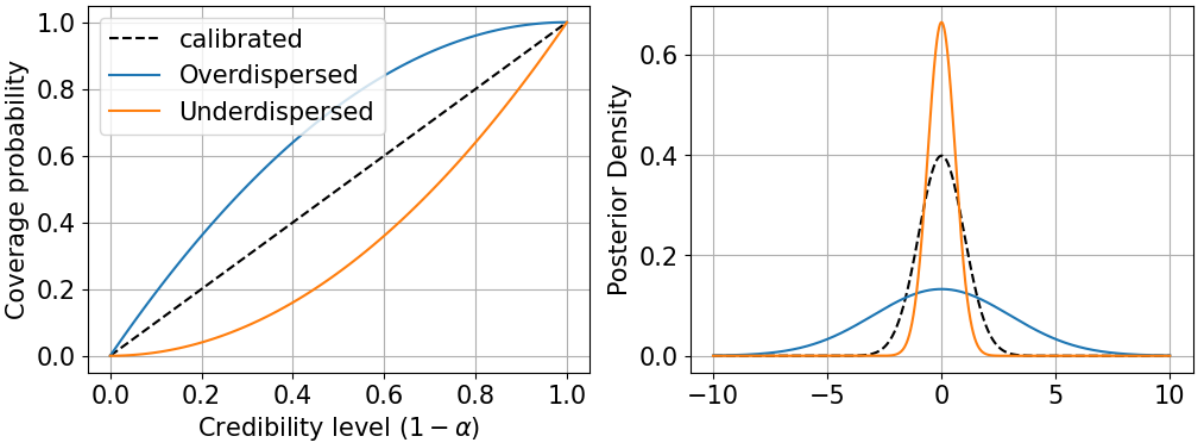

Figure 3.3. Example of a calibration test. On the left side, the perfect calibration is depicted by the black dotted line. The orange line represents an overdispersed posterior, which is undesirable as it may wrongly reject plausible parameter values. The blue line represents an underdispersed posterior, meaning it will be conservative in its estimation. This issue is mitigated by the fact that the ground truth will still be present, although with reduced precision. This phenomenon is illustrated in the plot on the right side.

It is important to note that this calibration test assesses the consistency of the approximation with the prior, not its accuracy. Therefore, while it cannot confirm the correctness of our approximation, it can indicate if the approximation is flawed.

Chapter 4

Orbital Characterization of β-pic b

The first part of this thesis was to reproduce the results of the α-DPI paper [1] on the exoplanet β-pic b using simulation-based inference and comparing the different methods.

4.1 Prior

The first step is to define the prior of the parameters of the orbit of the exoplanet $p(\theta)$. Table 4.1 shows the prior distribution used for the first part of this work. The parameters are the same as the ones used in the work of the α-DPI paper to better compare the results. Note that there is an error in the paper as the parallax and the total mass shown are from the planet GJ 504 and the prior for the other parameters are also not the ones they used in the implementation.[1]

Table 4.1. Prior distribution used for the different Keplerian parameters to characterize the orbit of β-pic b.

Parameter	**Unit**	**Prior Distribution**
Semi-major axis (a)	astronomical unit(au)	$\log\mathcal{U}(4, 40)$
Eccentricity (e)	-	$\mathcal{U}(10^{-5}, 0.99)$
Inclination angle (i)	degree (°)	Sine$(81, 99)$
Argument of periastron (ω)	degree (°)	$\mathcal{U}(0, 360)$
Longitude of ascending node (Ω)	degree (°)	$\mathcal{U}(25, 85)$
Epoch of periastron passage (τ)	-	$\mathcal{U}(0, 1)$
Parallax (π)	milliarcsecond(mas)	$\mathcal{N}(51.44, 0.12)$
Total mass (M_T)	solar mass($M_\odot$)	$\mathcal{N}(1.75, 0.05)$

A sine prior is used for the inclination angle because the inclination i and the longitude of ascending node Ω correspond to the two angles in a spherical coordinate system. The inclination is the polar angle and the longitude of the ascending node is the azimuthal

[1]The correct prior distribution used in the implementation of the α-DPI paper can be found in the GitHub repository of the main author: https://github.com/HeSunPU/DPI/blob/main/DPItorch/DPIx_orbit.py

angle. Using two uniform priors for these two angles would result in a non-uniform prior on the sphere where the poles would have a higher probability than the equator. Using a sine prior for the inclination angle solves that and gives an isotropic uniform distribution. This can be seen in Fig. 4.1.

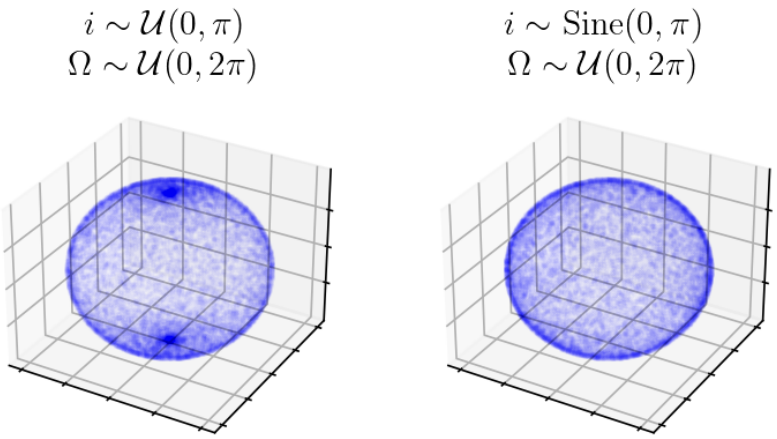

Figure 4.1. Comparaison of the uniform and sine prior for the inclination angle. Each point can be seen as an exoplanet at periastron with a fixed semi-major axis, fixed eccentricity, and fixed argument of periapsis. We can see that using two uniform priors generates clusters at the poles of the sphere. The sine prior gives a uniform isotropic distribution on the sphere. This plot is an adaption of the one found in the `orbitize!` Documentation. [10]

The choice of defining the time of the periastron passage τ between 0 and 1 explained in Table 2.1 is now more clear. If we did not have any information on the time of the periastron passage, we would have a uniform prior to the time, which is difficult to bound as it could be infinite. With the way it is defined in Table 2.1, we can just have a uniform prior between 0 and 1.

4.2 Simulator

The second step is to define the simulator that has to generate artificial data that has to be as close as possible to the real data :

$$(\mathbf{x}, \theta) \sim p(\theta)p(\mathbf{x}|\theta)$$

To reproduce the results of the α-DPI paper, I designed a simulator that generates an astrometric data point in relative Right Ascension (Δ RA) and relative Declination (Δ DEC) for each epoch with an observation of the exoplanet. These observations use sampled parameters θ from the prior defined in Table 4.1.

I used the `calc_orbit` function from the `orbitize!` package [10], which takes the parameters, epochs, and a reference epoch as input. It returns Δ RA and Δ DEC for each epoch by solving Equations 2.3 and 2.4. The function also returns the radial velocity (RV), which was not used in this work but could be incorporated in future work to enhance results. In

the paper of Maire *et al.* (2023) [9] it is explained how using data from different sources is actually the way to go to improve the results because each source can constrain different parameters in their own ways.

To account for observational uncertainties, I added random noise to each generated data point. This noise is sampled from a normal distribution with a standard deviation based on the observational error, as provided in the β-pic b dataset in the `orbitize!` package. Details of this dataset are available in Appendix A.

To ensure compatibility with the neural network, I standardized the data and parameters to the range $[-1, 1]$. This standardization removes scale impact and improves neural network convergence.

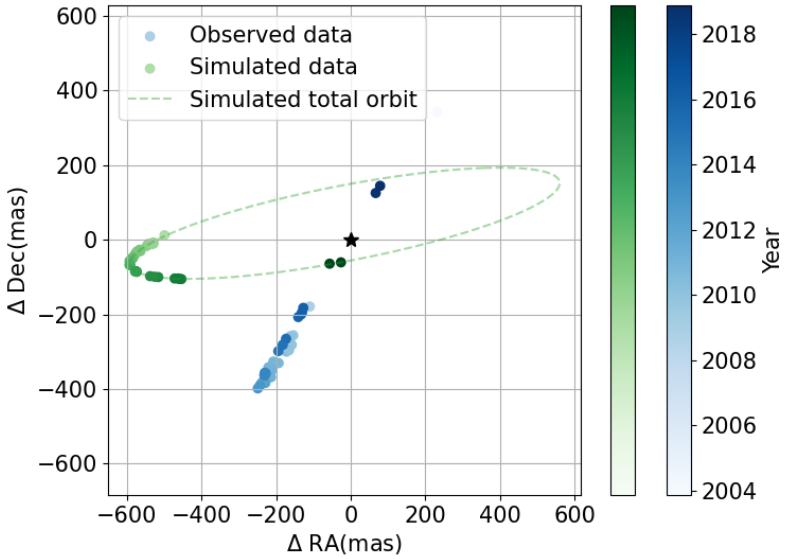

Figure 4.2. Right Ascension and Declination plot of the β-pic b exoplanet observations and one simulated orbit with dots at each epochs where there was an observation of the exoplanet. A dataset with 8 million of these simulated orbits was generated to train the neural network. The black star represents the position of the star β-pic. These observations are divided by a factor of 10^6 to be more suitable for neural network training. The parameters sampled for this orbit are : $a = 12.85$, $e = 0.46$, $i = 98.82$, $\omega = 271.35$, $\Omega = 79.01$, $\tau = 0.67$, $\pi = 51.29$, $M_T = 1.75$.

With this simulator, I generated a training set of size of 2^{23} pairs of $(\mathbf{x}, \theta)$ where $\mathbf{x}$ are the simulated observations as can be seen in Figure 4.2. I also generated a validation set of size 2^{20} to monitor the loss during the training and detect overfitting.

4.3 Architecture

The Neural Posterior Estimator (NPE) is implemented as an autoregressive neural spline flow. The concept is similar to the one presented in Section 3.1, but instead of using coupling layers, autoregressive layers are employed. This means that instead of splitting the input into two parts and transforming one based on the other, each parameter depends

on the preceding ones [31]. This approach was also used in the Neural Spline Flows paper [27], which demonstrates how both implementations can be used with the same overall performance.

The NPE consists of a neural spline flow with three transformations. Each transformation is a multi-layer perceptron (MLP) with 5 layers, each containing 512 neurons. The activation functions between the layers are Exponential Linear Units (ELU) [32]. The splines have 9 knots, with the first and last knots fixed at $[-5, -5]$ and $[5, 5]$ respectively, each with a derivative set to 1.

Each neural network outputs eight times the vector $[w, h, d]$, corresponding to the eight orbital parameters. The parameters w and h represent the width and height of the bins, each of size 8, and d represents the derivative of the spline at the knots and has a size of 7 because the derivative at the last knot is fixed.

A schematic of this architecture is shown in Figure 4.3. This implementation was achieved using the `Lampe` [33] and `Zuko` [34] libraries. The architectural choices were inspired by the work of Vasist *et al.* (2023) [29], who also used a Neural Posterior Estimator to retrieve the posterior distribution of exoplanet atmospheric parameters from spectroscopic observations.

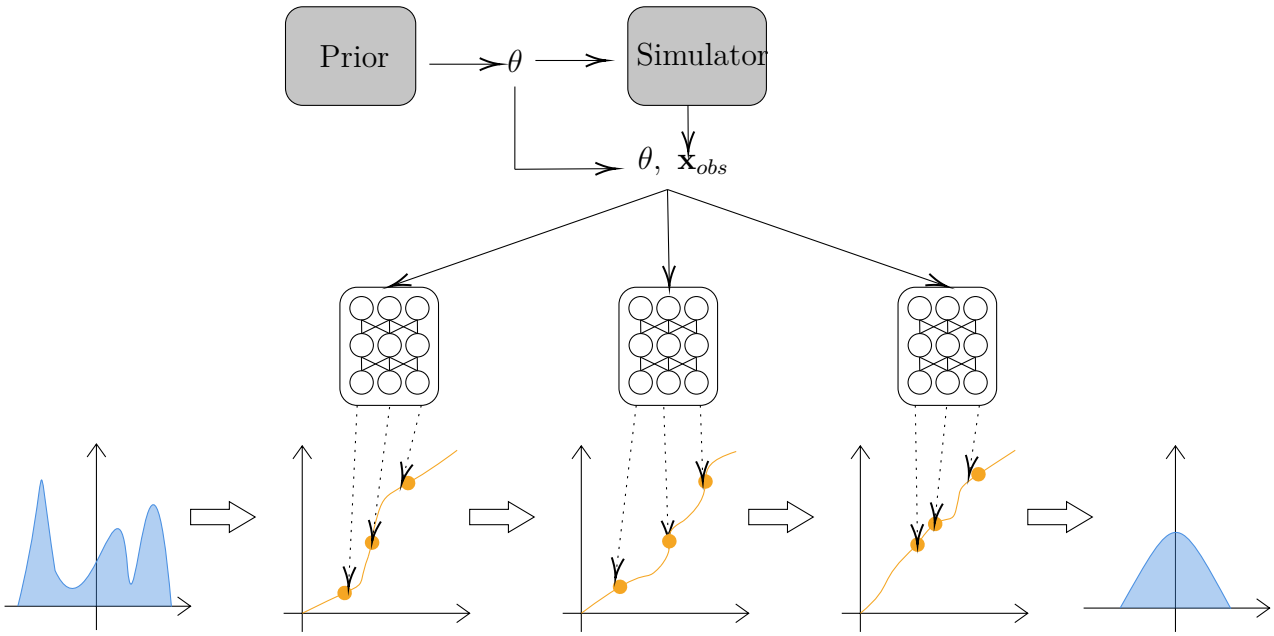

Figure 4.3. The architecture of the Neural Posterior estimation network used. At training time, θ are sampled from the prior which are used through the simulator to generate artificial observations $\mathbf{x}_{obs}$. Those are used with the corresponding θ as input to neural networks which outputs the positions of the knots and the derivatives at each knot for each transformation. There is one neural network for each transformation. The distribution to the left is unknown and represents the posterior distribution of one parameter, the whole network is trained to transform this distribution to a normal distribution, hence the name *Normalizing* Flows. It does so by minimizing the negative log-likelihood of the samples transformed to the normal distribution. At inference time, as explained the transformation is invertible, we can sample from the normal distribution and go through the inverse transformation to get a sample from the posterior distribution.

4.4 Training

This flow is trained by minimizing the expected negative posterior log density from Equation 3.3 over the training set. Each epoch of the training consists of taking 1024 samples of batch size 2048 from the training set and computing the loss. I also compute the validation loss at each epoch by taking a slice of 256 samples of batch size 2048 from the validation set and calculating the loss. To train I used a variant of stochastic gradient descent, the AdamW optimizer[35] implemented in pytorch with a starting learning rate of 10^{-3}. I used a factor scheduler to decay the learning rate by a factor of 0.5 if there is no improvement in the validation loss for 32 epochs. The training is stopped if the learning rate is under 10^{-6}. Weight decay is also used and set to 10^{-2}. The training is done on a single NVIDIA GTX 1080 Ti GPU like in the paper of Sun *et al.* (2022) [1] to ensure a legitimate comparison between the two methods. The idea behind this training procedure also came from the paper of Vasist *et al.* (2023) [29].

4.5 Results

4.5.1 Reproducing the results of the α-DPI paper

The first part of this work aims to reproduce the results of Sun *et al.* (2022) [1] on the exoplanet β-pic b. I used the same observations of β-pictoris b as they did to compare the corner plot of the posterior distributions. Although Sun *et al.* did not mention it explicitly, they used only 18 observations for their study. They employed 16 affine transformations using the RealNVP architecture [25], where each transformation consists of 2 fully connected layers with 64 units and leaky ReLU activation functions (negative slope of 0.01). They trained the model for 24,000 epochs with a batch size of 8,192, using the Adam optimizer [36] with a learning rate of 2×10^{-3}. They identified an optimal α value of 0.6. The same priors described in Table 4.1 were used.

Training on a single NVIDIA GTX 1080 Ti GPU took approximately 1.5 hours. To match this training time, I trained the NPE for 256 epochs.

As I only consider observations in the Right Ascension and Declination format in my simulator and not in the Separation/Position Angle format, like `Orbitize!` did when inferring the orbital parameters of real exoplanets that may have been observed in both formats, I transformed those observations into RA/DEC using Equations 2.1 and 2.2. This is the case for β-pic b.

Loss plot

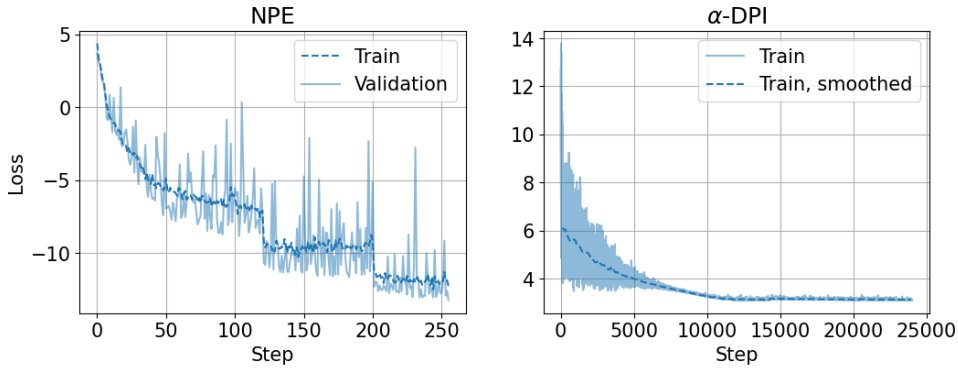

Figure 4.4. Loss plot showing the training and validation loss of the NPE and the training loss for the α-DPI method. Note that the validation loss is not available for the α-DPI method as it was not implemented. The two losses represent different metrics: the α-DPI loss is the annealed α-divergence loss from Equation 2.19, while the NPE loss is described in Equation 3.3, making direct comparison challenging. Despite the annealing of the α-DPI method, its loss still diverges to infinite values in the early stages of training. The number of steps is also different: for the NPE, one step corresponds to 1024 samples of batch size 2048, whereas for the α-DPI method, one step corresponds to 1 sample of batch size 8192. For clarity, a smoothed version of the loss, which takes the mean of every 100 steps, is also plotted on the right.

Plotting the loss is always a good way to diagnose the training of a neural network. In Figure 4.4, I plotted the training and validation loss of the NPE and the training loss of the α-DPI method, as they did not implement a validation loss. We can see that no overfitting is observed, as the validation loss is not increasing while the training loss is decreasing.

The higher variance in the validation loss of the NPE is due to the fact that the batch size per step is smaller, 256 samples of batch size 2048 compared to the 1024 samples of batch size 2048 for the train loss.

Overall, no overfitting is observed and the loss is decreasing, which shows that the model is learning the underlying distribution of the data and we can expect that it will be able to predict the posterior distribution of the parameters of β-pic b. One can also see that the loss of the α-DPI is plateauing, while the loss of the NPE is still decreasing. I stopped the training of the NPE at 256 epochs so that the overall training time would be comparable to the α-DPI method but we could have continued the training. Results from a longer run are available in the Appendix C, on Figure C.1 and C.2. No significant improvement is observed in the longer run. It was able to restrict the posterior distribution of the mass, the inclination, and the angle of the ascending node slightly more, but the overall shape of the posterior distributions stayed the same.

Corner plot

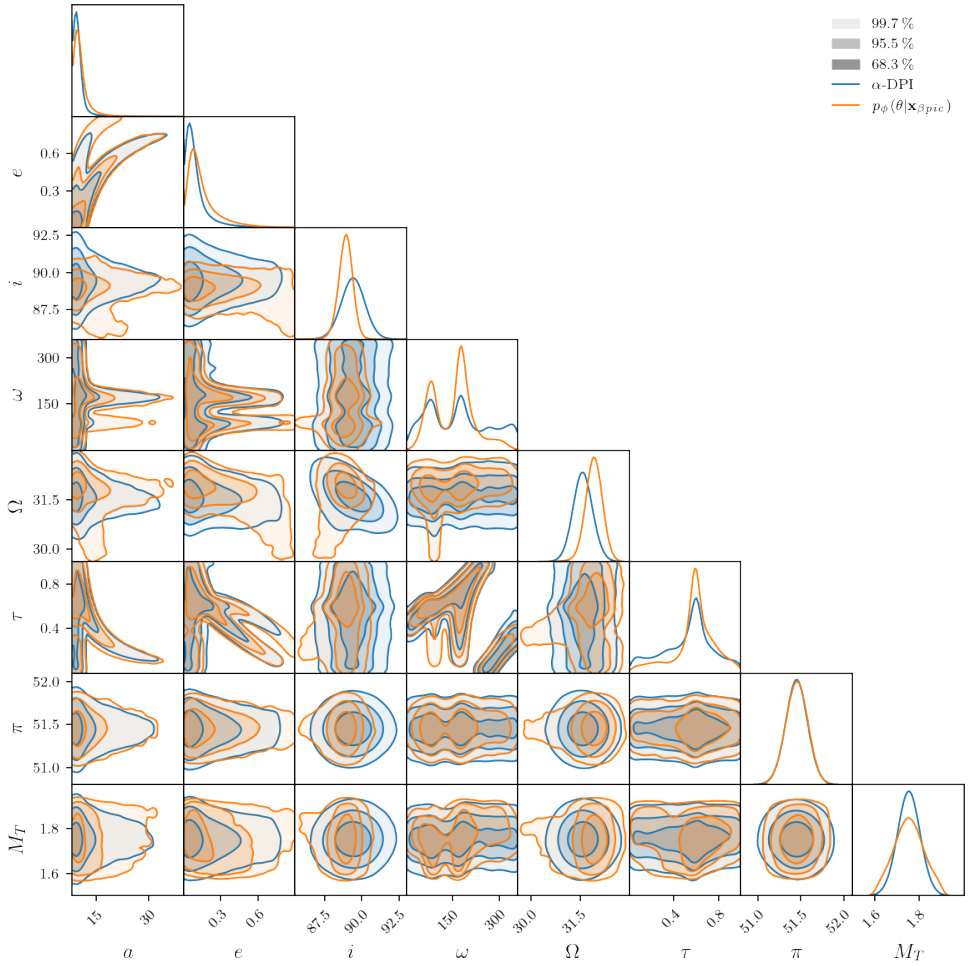

Figure 4.5. Corner plot of the posterior distribution of the parameters of β-pic b using the α-DPI and NPE methods. The contours represent the 68.3%, 95.5%, and 99.7% confidence intervals. The diagonal represents the marginal posterior distribution of each parameter and the off-diagonal represents the 2D marginal posterior distribution of each pair of parameters. In blue are the posterior predicted by the α-DPI method and in orange are the posterior predicted by the NPE. The MCMC procedure from the paper could not be exactly reproduced as the necessary details were not provided.

The corner plot in Figure 4.5 demonstrates that the NPE is capable of reproducing the results of the α-DPI. The paper by Sun *et al.* (2022) [1] claimed that the Kullback-Leibler divergence fails to capture disconnected modes in the posterior because it does not include samples from low-probability regions.

However, with the technique explained in Equation 3.2 with the double expectation, the KL-divergence can be used effectively without needing Renyi's α-divergence. With this providing the new loss function written in Equation 3.3, there is no need for the annealing required for the α-DPI method. The low-density regions of ω and τ are well captured.

The NPE appears slighlty less confident than the α-DPI for the parameters semi-major axis a and the eccentricity e and the total mass M_T but more confident for the inclination i, the argument of periastron ω, the longitude of the ascending node Ω and the time of

periastron passage τ. However, there are also minor discrepancies in the inclination i and Ω. Despite these differences, the posterior distributions are consistent with those obtained using more observations, as shown in the corner plot in Figure 4.8.

Posterior Predictive Check

The posterior predictive check is a method used to validate the model by comparing the observations to the predictions. In Figure 4.6, the 68.3% confidence interval of the orbit of β-pic b is shown for both methods. The blue-shaded region represents the predictions from the α-DPI method, while the orange-shaded region corresponds to the Neural Posterior Estimator (NPE) predictions.

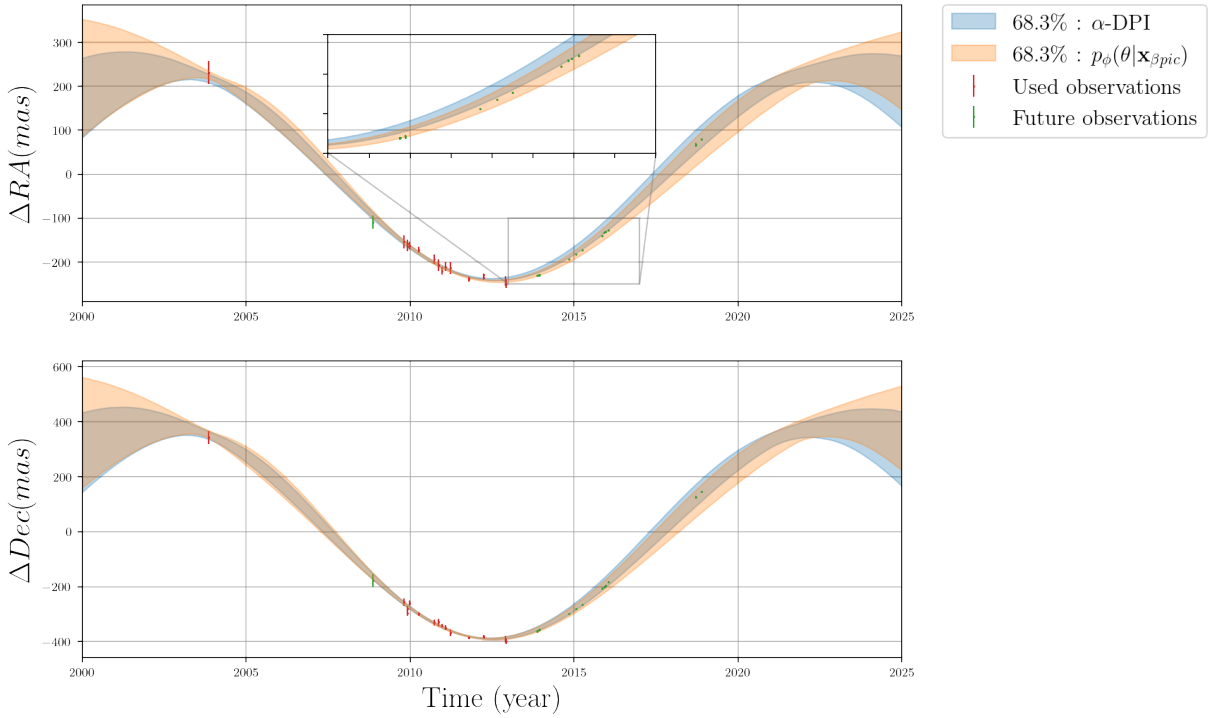

Figure 4.6. Right Ascension and Declination plot depending on the year of the observations of β-pic b using the α-DPI method and NPE denoted as $p_\phi(\theta|\mathbf{x}_{\beta pic})$. In red are the observations used for the training of both methods and in green are other observations of β-pic b. This plot was made by sampling 1000 samples from the posterior distribution of the parameters and to generate the corresponding orbit of β-pic b. The 68.3% confidence interval was then plotted by taking the 15.85% and 84.15% quantiles of these samples. The zoom on the upper plot shows how the NPE is more consistent with later observations of the exoplanets that were not used to train the two models, as the 68.3% confidence interval of the NPE encapsulates them.

We can observe several key points from this figure, first, the NPE produces a more realistic orbit compared to the α-DPI method, particularly in alignment with later observations. This suggests that the NPE model has effectively learned the underlying distribution of the exoplanet's orbital parameters. This could be due to the flexibility of the Neural Spline Flows used in the NPE compared to the RealNVP architecture used in the α-DPI method. This flexibility allows the NPE to capture more complex relationships between the observations and the parameters, leading to more accurate predictions. The slightly

wider confidence intervals of the NPE model also indicate a more cautious approach, capturing the parameter variability better than the α-DPI method which has narrower intervals.

Coverage plot

As explained in Section 3.2, the coverage plot evaluates the faithfulness of the Neural Posterior Estimation by assessing the expected coverage. In Figure 4.7, the plot demonstrates that the model's coverage is almost calibrated, but it is slightly larger than the credibility level $1 - \alpha$. This indicates that the posterior is somewhat less confident than it should be, which is preferable to being overly confident and potentially leading to incorrect predictions.

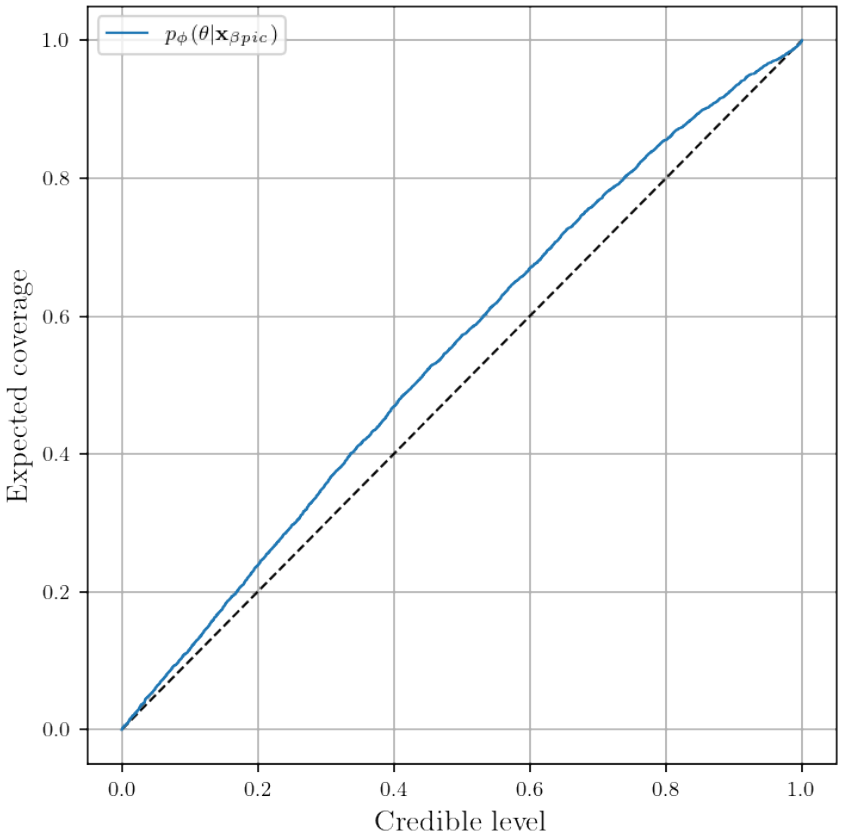

Figure 4.7. Calibration test of the NPE. We can see that the model is almost calibrated but still a bit conservative as the coverage is slightly larger than the credibility level $1 - \alpha$. 1000 pairs of parameters/observations were used to generate this plot.

We can see that the model is almost calibrated but a slight conservatism is present, as evidenced by the larger coverage compared to the credibility level (1-α). This conservatism is beneficial as it reduces the risk of overconfident wrong predictions.

Generating a direct coverage plot for α-DPI is not feasible due to computational constraints. Specifically, obtaining a plot comparable to Figure 4.7 would require running the method 1000 times. This necessity arises because α-DPI is not amortized, necessitating multiple runs to account for the likelihood calculation.

Even trying to infer its performance from the corner plot in Figure 4.5 is difficult as the marginal posterior distributions are narrower for some parameters and broader for others. If they were all broader, we could infer that the coverage would be above the credibility level of the NPE and if they were all narrower, we could infer that the coverage would be below the credibility level of the NPE. Here nothing can be inferred.

However, I did try to train a NPE with affine transformations instead of spline transformations to see if the results would be similar to the α-DPI method. I tried with the NICE normalizing flow [37] but could not achieve similar results as with NSF or α-DPI. The results can be found in the Appendix C on Figure C.3 and C.4.

The conservatism of the NPE could be attributed to the high errors in the observational data of β-pic b, especially prior to 2014. These high errors lead to broader posterior distributions, reflecting the increased uncertainty in the parameter estimates. This caution in the model's predictions ensures that it does not make overly precise claims that are not supported by the data quality.

4.5.2 Comparaison with MCMC using all the observations

To further evaluate the performance of the NPE, I decided to train it using all the available observations of β-pic b. The goal was to determine if the NPE could match the accuracy of MCMC while offering a computationally efficient alternative. The OFTI method was excluded as it cannot handle such a large dataset effectively.

I also ran the α-DPI method using the full set of observations to facilitate a comprehensive comparison among the three methods. Consistent priors, detailed in Table 4.1, were applied across all methods to ensure a fair comparison. The dataset creation and NPE training required approximately 1.5 hours, encompassing 512 epochs using the same training procedure as previously described. Similarly, α-DPI training also took around 1.5 hours. For the MCMC, 45.000 iterations were run across 1,000 chains with a burn-in of 40,000, resulting in a total runtime of 51:03:19 on 10 CPUs. This setup is the same procedure used in the original α-DPI paper. The converged MCMC chains can be found in the Appendix C on Figure C.7.

Corner plot

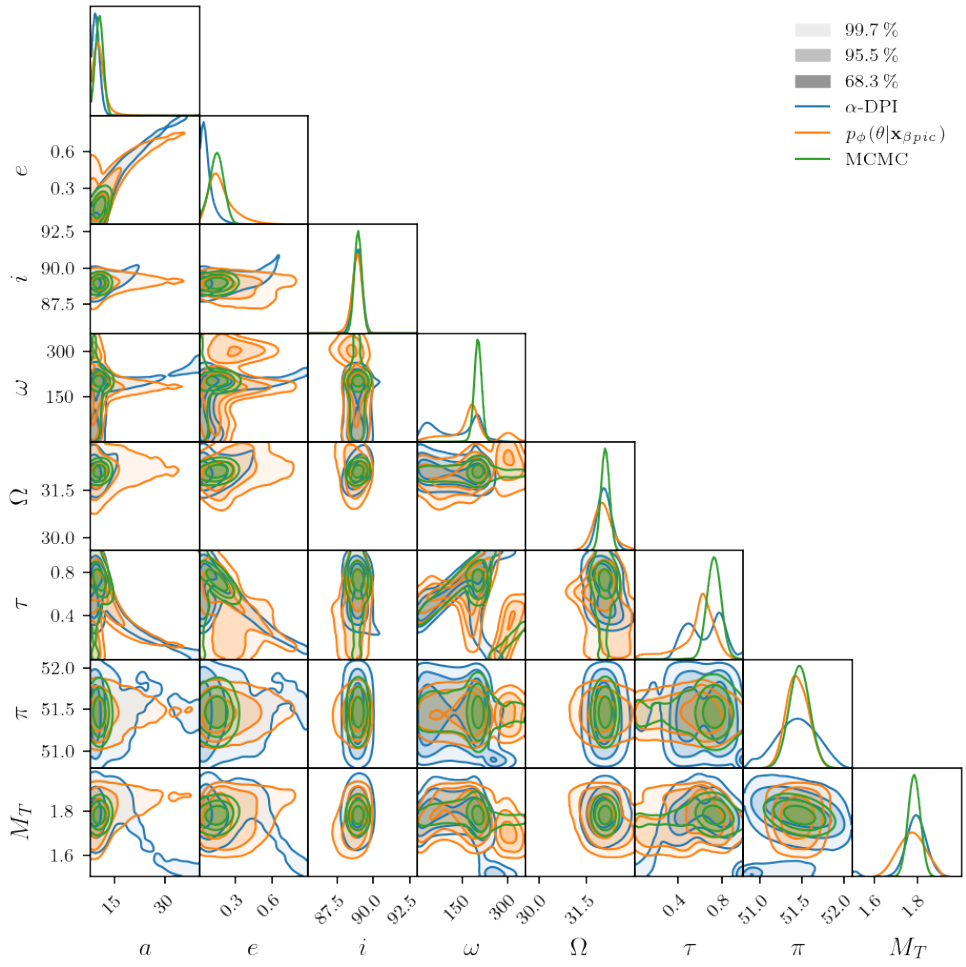

Figure 4.8. Corner plot of the posterior distribution of the parameters of β-pic b using the α-DPI method, NPE, and MCMC. The contours represent the 68.3%, 95.5%, and 99.7% confidence intervals. We can see that the NPE method, in orange, is able to reproduce the results of MCMC, in green, better than α-DPI, in blue.

The corner plot in Figure 4.8 provides a comparative visualization of the posterior distributions obtained from the three methods. It is evident that the marginal posterior distributions derived from the NPE resemble more those produced by MCMC, whereas the α-DPI method shows discrepancies, especially for the semi-major axis a, the eccentricity e, the argument of periastron ω and the time of periastron passage τ. The posterior of the parallax is also more spread out in the α-DPI method. This similarity in the NPE and MCMC results is particularly notable given the significant difference in computational efficiency. The NPE required an order of magnitude less time to compute compared to MCMC, making it a highly efficient alternative while maintaining a reasonable level of accuracy, although the posterior distributions produced by NPE are generally more dispersed compared to those obtained through MCMC.

Posterior Predictive Check

As with the previous analysis, I did a posterior predictive check to assess the NPE's ability to encapsulate the observed data. This is illustrated in Figure 4.6 where it shows the

68.3% confidence intervals for the Right Ascension (RA) and Declination (Dec) of β-pic b over time for the NPE, MCMC, and α-DPI methods.

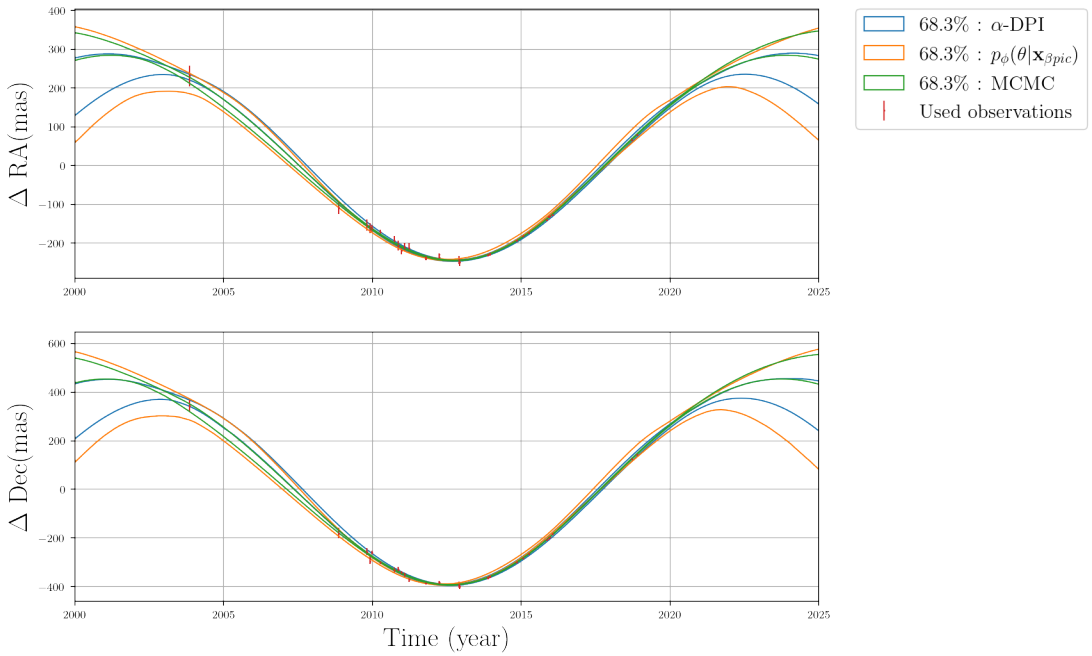

Figure 4.9. Right Ascension and Declination plot depending on the year of the observations of β-pic b using the α-DPI method and NPE denoted as $p_\phi(\theta|\mathbf{x}_{\beta pic})$. All three methods are consistent with the observations, but the NPE exhibits a broader confidence interval compared to the two other methods. 1000 samples were used for each method and the 15.85% and 84.15% quantiles were used to plot these confidence intervals.

All three methods are consistent with the observations, but the NPE exhibits a broader confidence interval compared to the two other methods. Notably, the NPE's confidence interval encapsulates the intervals predicted by the other two methods, particularly before 2002 and after 2020.

α-DPI displays noticeable discrepancies with MCMC both before 2002 and after 2020. This inconsistency arises from the α-DPI's misestimation of the semi-major axis (a) and eccentricity (e) parameters. Specifically, α-DPI predicts smaller values for both a and e compared to MCMC and NPE, leading to deviations in the predicted orbital paths.

The broader confidence intervals observed in the NPE predictions could reflect its flexibility and robustness in capturing the underlying uncertainties in the orbital parameters. This attribute is particularly valuable for ensuring that the predicted orbital paths remain reliable over a wide range of observations, even though it may result in less precise but more inclusive predictions. This can be confirmed by the coverage plot in the next section.

Coverage plot

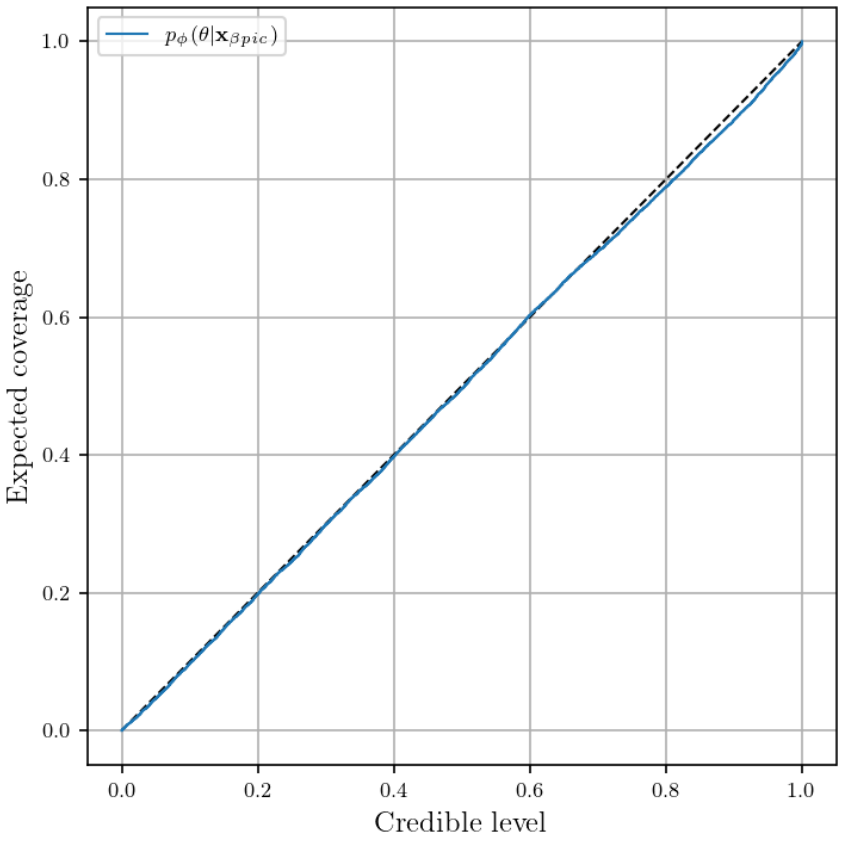

Figure 4.10. Calibration test of the NPE. We can see it is perfectly calibrated as the coverage is equal to the credibility level $1-\alpha$. 1000 pairs of parameters/observations were used to generate this plot. Once again, the α-DPI method could not be tested due to computational constraints, as it would require running the method 1000 times. The same is true for the MCMC method.

The coverage plot in Figure 4.10 demonstrates that the NPE is well calibrated, with the coverage closely matching the credibility level $1-\alpha$. This means that the predicted posterior marginal distribution from Figure 4.8 and the credibility intervals accurately reflect the true uncertainty in the model's predictions.

A possible explanation for the improved calibration compared to Figure 4.7 is that the newer observations had significantly lower error margins due to advancements in telescope precision. This reduction in observational error constrains the posterior distribution to a smaller region of the parameter space, leading to more precise and accurate predictions.

Conducting such a coverage diagnosis with α-DPI and MCMC is not feasible due to the high computational costs involved. However, by comparing the results of the NPE on the corner plot with those from MCMC and α-DPI, we can infer the likely performance of MCMC. It is reasonable to assume that MCMC would exhibit slightly lower coverage than the credibility level as it tends to be more confident in its predictions. while for α-DPI it is difficult to infer its performance as posterior distributions are sometimes narrower like for the eccentricity and sometimes broader like for the parallax.

The good calibration of the NPE is further illustrated by the posterior predictive check on a

generated observation from the test set, shown in Figure 4.11. In this figure, the confidence intervals effectively encapsulate the true orbit given the observations, demonstrating the NPE's ability to accurately predict the orbital parameters within the expected uncertainty ranges. Also, the amortization of the NPE allowed us to generate the results for this figure almost instantaneously. It just needed a single forward pass through the network. If I wanted to generate the same results with MCMC, it would have taken 50 hours, like for the original data of β-pic b.

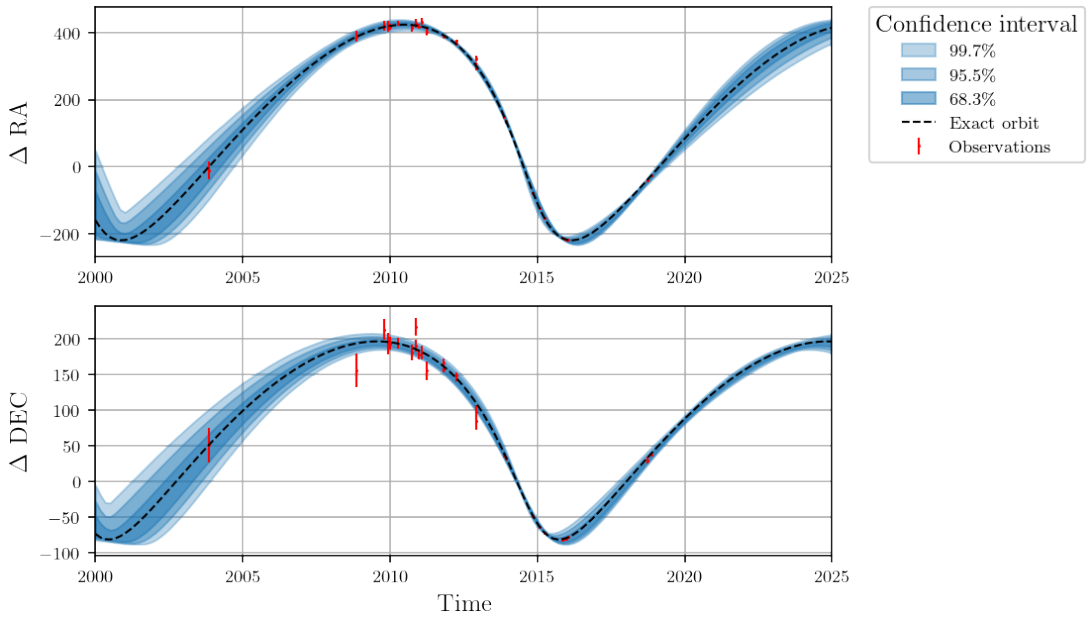

Figure 4.11. Posterior predictive check of the NPE on one of the samples from the test set. The 68.3%, 95.5% and 99.7% confidence intervals are shown along the true orbit

4.5.3 Conclusion

In this section, I demonstrated the ability to reproduce the results of the α-DPI method for the exoplanet β-pic b using NPE. While this method offers significant speed advantages, generating results almost instantaneously, it is important to note that this efficiency is constrained by certain conditions. Specifically, the model can only generate results for exoplanets with the same number of observations as β-pic b, taken at the same times. Additionally, if new observations of β-pic b were to be made, retraining the model would be necessary, meaning that the NPE is not yet capable of generating amortized results. Nonetheless, even with these constraints, the NPE remains an order of magnitude faster than traditional MCMC methods, though it does not fully exploit the potential of Simulation-Based Inference.

The true strength of SBI lies in its flexibility and scalability. With the appropriate simulator and model architecture, it is conceivable to develop an NPE capable of producing results comparable to MCMC, but almost instantaneously, for any exoplanet, following a single training phase. This would represent a major advancement in the field of exoplanet astrometry, providing researchers with a powerful tool for rapid and accurate orbital characterization.

The development of such a versatile and efficient network is the focus of the next part of this work. In the following chapter, we will explore the design and implementation of a

more general NPE model, capable of handling a diverse range of observational data and delivering high-fidelity results across various exoplanetary systems.

Chapter 5

Orbital Characterization of any Exoplanet

In this chapter, I present the development of a generic model capable of instantaneously characterizing the orbit of any exoplanet given a set of astrometric observations.

5.1 Prior

The priors used in Chapter 4 were used specifically to β-pic b and are not suitable for a more general model. To address this, the priors need to be wider to encapsulate the diversity of orbital parameters across different exoplanets. The following priors were selected:

Table 5.1. Prior distribution used for the different Keplerian parameters for a generic exoplanet.

Parameter	Unit	Prior Distribution
Semi-major axis (a)	astronomical unit(au)	$\log\mathcal{U}(4, 100)$
Eccentricity (e)	-	$\mathcal{U}(10^{-5}, 0.99)$
Inclination angle (i)	degree (°)	Sine(0, 180)
Argument of periastron (ω)	degree (°)	$\mathcal{U}(0, 360)$
Longitude of ascending node (Ω)	degree (°)	$\mathcal{U}(0, 360)$
Epoch of periastron passage (τ)	-	$\mathcal{U}(0, 1)$
parallax (π)	milliarcsecond(mas)	Not used
Total mass (M_T)	solar mass($M_\odot$)	$\mathcal{U}(0.2, 3)$

To reduce the parameter space and provide a proof of concept, I limited the upper bound of the semi-major axis to 100 au. If the model successfully represents accurate posterior distributions for known planets within this range, such as β-pic b or the four planets of HR 8779 [38], this bound could be extended, with additional training of the normalizing flow.

The total mass prior ranges from 0.2 to 3 solar masses, encompassing spectral types from approximately M5V to A0V. This range represents the types of stars around which planets

have been directly imaged.

The parallax parameter is not used directly in the model thanks to the precise parallax measurements provided systematically by the Gaia[1] mission for all stars in the galactic neighborhood. This precision encompasses all stars suitable for direct imaging of exoplanets, offering significantly better accuracy than what can be deduced from planetary orbit adjustments alone. Including it would unnecessarily widen the parameter space. In comparison to the prior used in Chapter 4, where the prior of the parallax was taken as a normal distribution, a more general prior would be too wide and would need longer training time for the normalizing flow to really understand the underlying link between the different parameters. Instead, parallax is treated as a scaling factor (see Equations 2.3 and 2.4), fixed to an arbitrary value (e.g., 100 mas) during model inference, and rescaled to the known parallax of the star afterward.

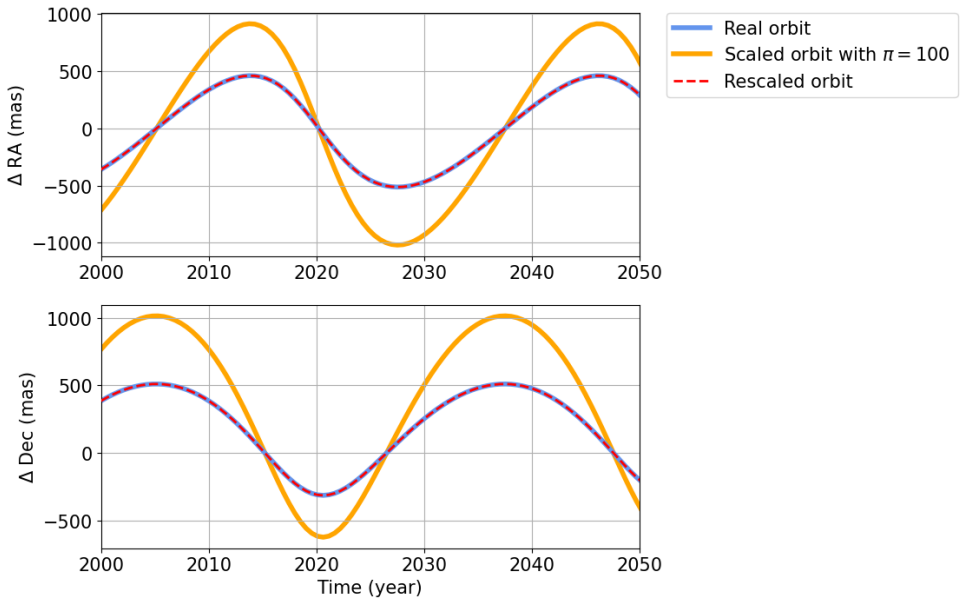

Figure 5.1. Effect on rescaling the parallax. The blue line represents an orbit generated by sampling orbital parameters from the prior distribution and calculating the 1000 astrometric observations over a certain amount of time. The orange line represents the orbit generated by the same parameters but with the parallax set to 100 mas, also by calculating the 1000 astrometric observations. The red dotted line is this orange orbit times the actual parallax of the star divided by 100. This demonstrates how the parallax can be put to the side during inference and rescaled afterward.

Figure 5.1 illustrates the effect of rescaling the parallax. During inference, the astrometric observations are adjusted to correspond to a parallax of 100 mas. The results are then scaled back to reflect the actual parallax of the star.

5.2 Simulator

With the appropriate priors in place, the simulator must be capable of generating data in a form that is invariant to the number of observations. The initial approach involved

[1] Gaia mission website

generating n observations over a span of twenty years, from 01/01/2002 to 01/01/2022, as most direct imaging observations of exoplanets fall within this period. (A list of examples of observations using direct imaging can be found in the paper of Do Ó et al. 2023 [39]).

Noise is added to the observations to simulate observational error. The noise is sampled from a normal distribution with a standard deviation of σ_{err}, which is chosen based on typical error values in different astrometric datasets. This error model reflects the uncertainties present in real observational data.

$$\epsilon = \mathcal{N}(0, \sigma_{\text{err}})$$

To handle varying numbers of observations, a mask is applied to the n values to randomly select between three and thirty observations. Three observations are necessary because any three distinct points on an ellipse define the ellipse's orbital plane. Thirty is chosen as the upper limit based on the maximum typical number of directly imaged observations of exoplanets.

This simulator design ensures that the generated dataset is diverse and representative of different observation scenarios, providing a robust basis for training the normalizing flow model. The next steps will involve refining the simulator to better match observational conditions and further tuning the model to improve its accuracy and efficiency.

5.3 Validation

To validate the model, I decided to apply it to all four planets orbiting the star HR 8799. Typically, astronomers would run separate MCMC simulations for each planet, a process that can be very time-consuming [40]. For this validation, I specifically chose to focus on HR 8799e.

The observations of all four planets are available in the Annex B. They were taken from Zurlo et al. [41].

For HR 8799e, I conducted an MCMC run lasting 14:21:32 hours on 10 CPUs using 1000 chains with 10000 iterations each. The converged chains, which are presented in the Annex C in Figure C.8, will serve as a benchmark for comparing the results produced by the normalizing flow model.

5.4 Residual Multi-layer Perceptron

For the first part of this work explained in Chapter 4, it was easier to define the dataset as we just had to produce one observation for each epoch. Here, the dataset is more complex as we have to produce a variable number of observations and at different epochs.

The first idea was to use a Residual Multi-layer Perceptron (ResMLP) [2] using the idea of zero padding. The first step is to divide the span of eighteen years into n intervals of the same length. In each of those intervals, a timestep is randomly chosen. The simulator would then produce observations at these timesteps given parameters sampled from the

prior. Observations would randomly be removed and changed to zero, meaning that there was no observation in this interval of time.

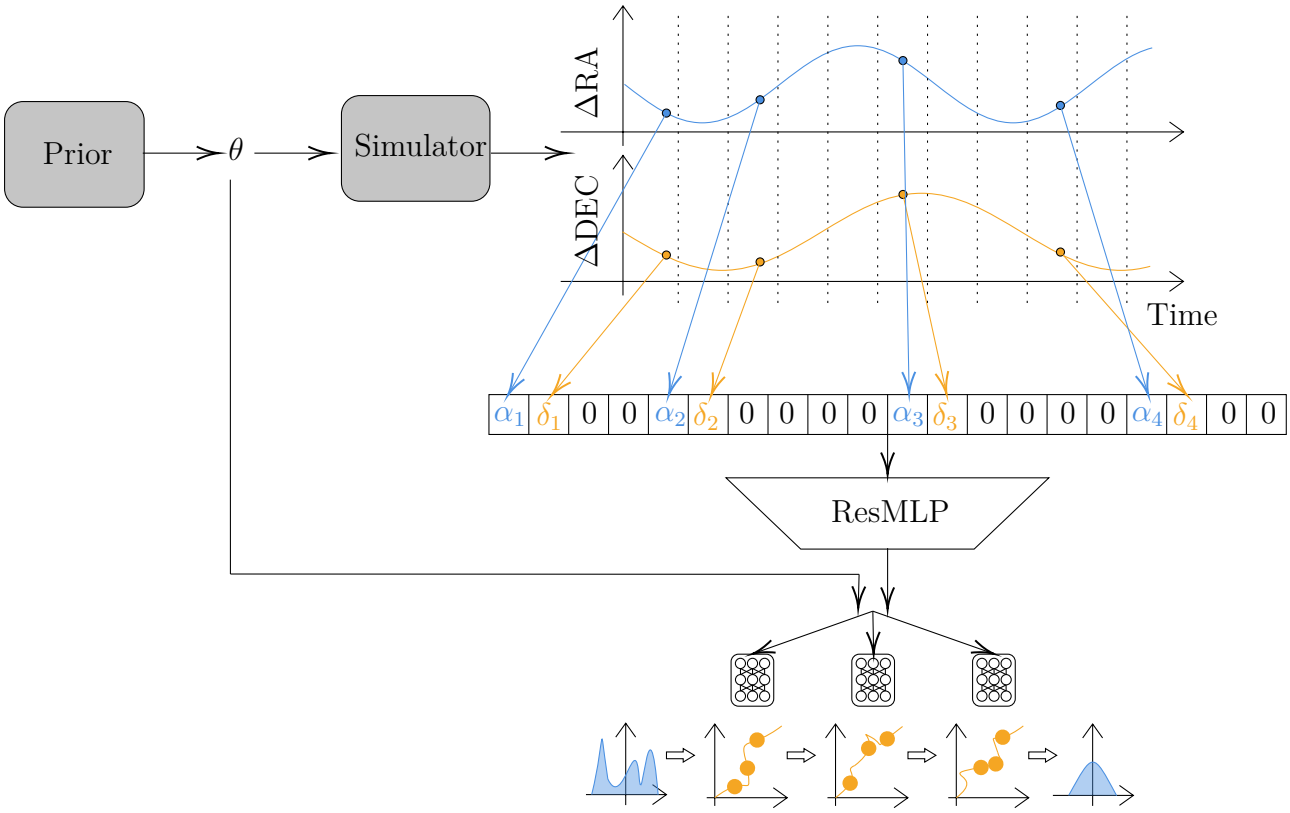

Figure 5.2. Architecture of the generic model using a ResMLP as an embedding network. Orbital parameters are drawn from the prior and used by the simulator to produce the observations, in this case, 4 observations are produced and are shown as dots on the Δ RA and Δ DEC plots. The two lines on which the dots are placed represent the orbit of the exoplanet. The time is discretized into 10 intervals of the same length, represented by the vertical dotted lines. The vector that is send to the ResMLP is of size 2× the discretization. Each even index represents the Δ RA and each odd index represents the Δ DEC. When there is no observation for a given interval, the value is set to zero. The embedding vector produced by the ResMLP is then used to train the Neural Posterior Estimator as in the previous chapter.

The ResMLP would then take this input and produce an embedding vector of a fixed size that would be used by the normalizing flow. This embedding network should be able to learn the underlying features of the data. The overall architecture is depicted in Figure 5.2. There is not much change in the NPE architecture, it is the same as the one used in Chapter 4 and shown in Figure 4.3.

The ResMLP architecture was the same as in the paper of Vasist et al. 2022 [29], meaning it is composed of 10 residual blocks, the first two of size 512, the next 3 of size 256, and the last 5 of size 128.

5.4.1 Effect of the discretization and the error

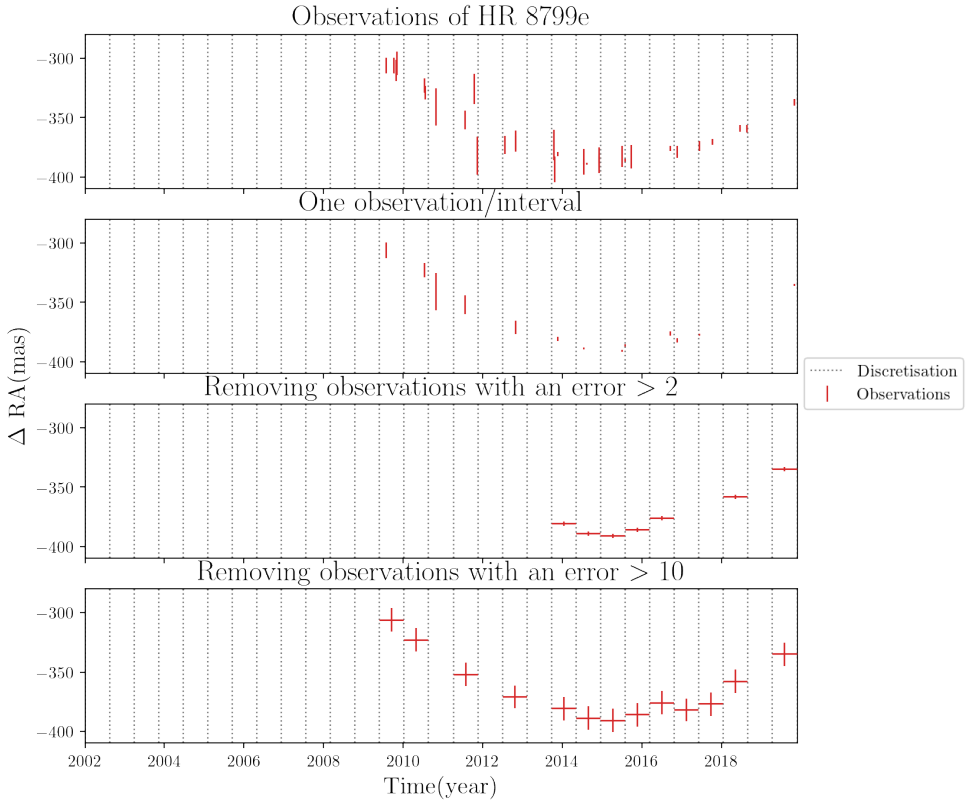

Figure 5.3. Processing the astrometric data of the exoplanet HR 8799e with different. The discretization, here set to 30 intervals of equal length between 2002 and 2020, are shown as vertical dotted lines. The first plot shows the observations of the exoplanet with their error bars. The second plot is after taking one observation per interval. If there are multiple observations in the same interval, the observation with the smallest error is taken. The two last plots are the observations that are left after the error threshold is applied. This threshold depends on the error that was applied in the training set. A trade-off can be seen, taking more accurate observations means less data is available, and taking less accurate observations means making the error on observations with a small error larger as the same error is applied to all observations in the training set.

The discretization of the time is an important parameter to set, as it adds a uniform error on the time of the observations in addition to the normal error on the observations. Setting it too high would mean having input vectors that are really large and sparse which will make the training of the ResMLP a lot harder as the weights of the network will not be updated often. Setting it too low will make the error on the time of the observations too high and the network will not be able to learn the underlying features of the data.

The error in the observations is also an important parameter to set. It is used to add observational noise to the data. If I choose a too-small error, it would be able to only take data with low noise, reducing the amount of data usable at inference time. If I choose a too-large error, observations that are known to be accurate will be made less accurate, this will increase the uncertainty on the observations and thus the uncertainty on the predictions.

Both the effect of the discretization and the error on the processing of real astrometric

data are shown in Figure 5.3.

To test the effect of the discretization, three different discretizations were tested, 30, 180, and 1800 intervals. The error in the observations was set to 2 mas.

The time for generating the dataset was about 1 hour for all three methods and the training took 6h 27m 11s for the discretization in 30 intervals, 7h 28m 42s for the discretization in 180 intervals and 16h 3m 38s for the discretization in 1800 intervals.

Corner plot

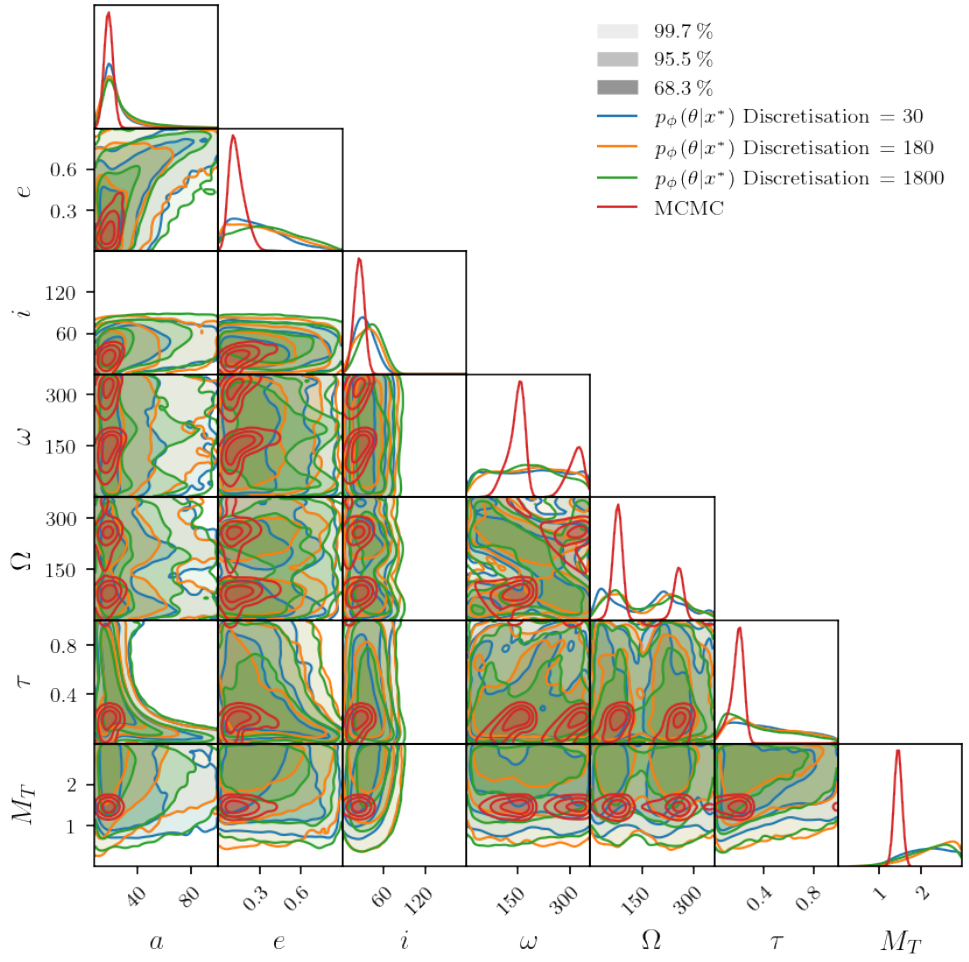

Figure 5.4. Corner plot of the posterior distribution of the orbital parameters of the exoplanet HR 8799e with the effect of discretisation

The three different discretization levels (30, 180, 1800) are compared against the MCMC results. Interestingly, the results shown in the corner plot in Figure 5.4 are all very close to each other. The marginal distributions for the discretization level of 30 seem to align more closely with the MCMC results for parameters such as the semi-major axis, eccentricity, and inclination. However, it is challenging to definitively determine which discretization level is the best, as all three levels seem to have high levels of inaccuracies.

One notable observation is that all methods struggle to accurately estimate the total mass of the system. This could be because, over the short fraction of the orbit observed, the effect of the system's mass on the astrometric data is not readily noticeable. While the

mass is an important parameter for characterizing the entire orbit, its influence may not be significant in the limited time span of our observations, making it a difficult parameter to constrain accurately. This parameter is still included because it becomes more relevant when considering longer observational periods or when combined with other data.

Posterior predictive check

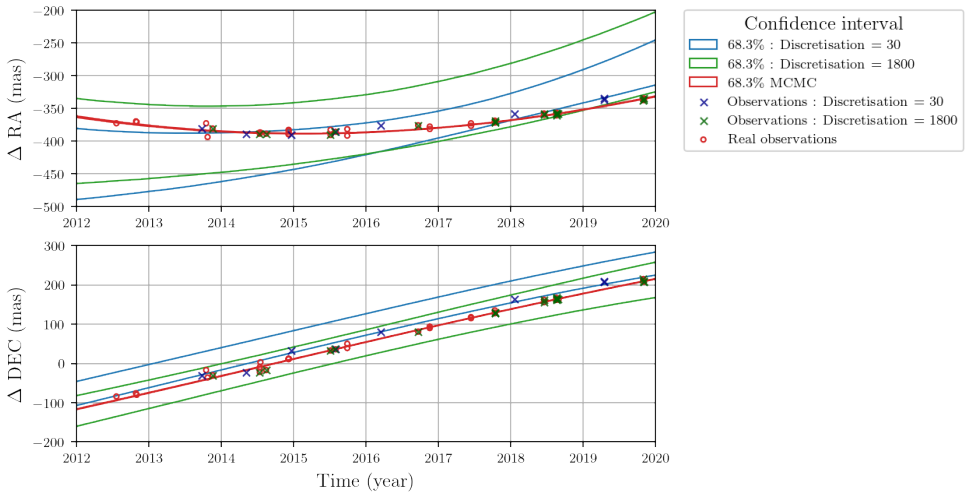

Figure 5.5. Posterior predictive check of the exoplanet HR 8799e with the effect of discretization. To make the plot more readable, only the two extremes of the discretization are shown. For the sack of clarity, only the two extremes are shown, the 30 and 1800 intervals discretization. The two methods seem to not get the general trend of the Right Ascension of the orbit. The 30 intervals discretization seems to be off while the 1800 intervals discretization seems to be more centered on the actual observations.

In Figure 5.5, we can see that both methods shown are not able to correctly predict the observations. The effect of the discretization is visible on the Δ DEC plot with the discretization in 30 intervals. The discretization sets a bias in the 68.3 % credible interval which is not centered on the actual observations. The discretization in 1800 intervals seems to be well centered on the actual observations for Δ DEC but is a lot more uncertain than MCMC while taking the same time to train.

For both discretization levels, the model fails to capture the overall shape of the orbit in terms of Δ RA. This indicates a fundamental limitation in the model's ability to accurately predict the Right Ascension component of the orbit, regardless of the discretization level.

Coverage plot

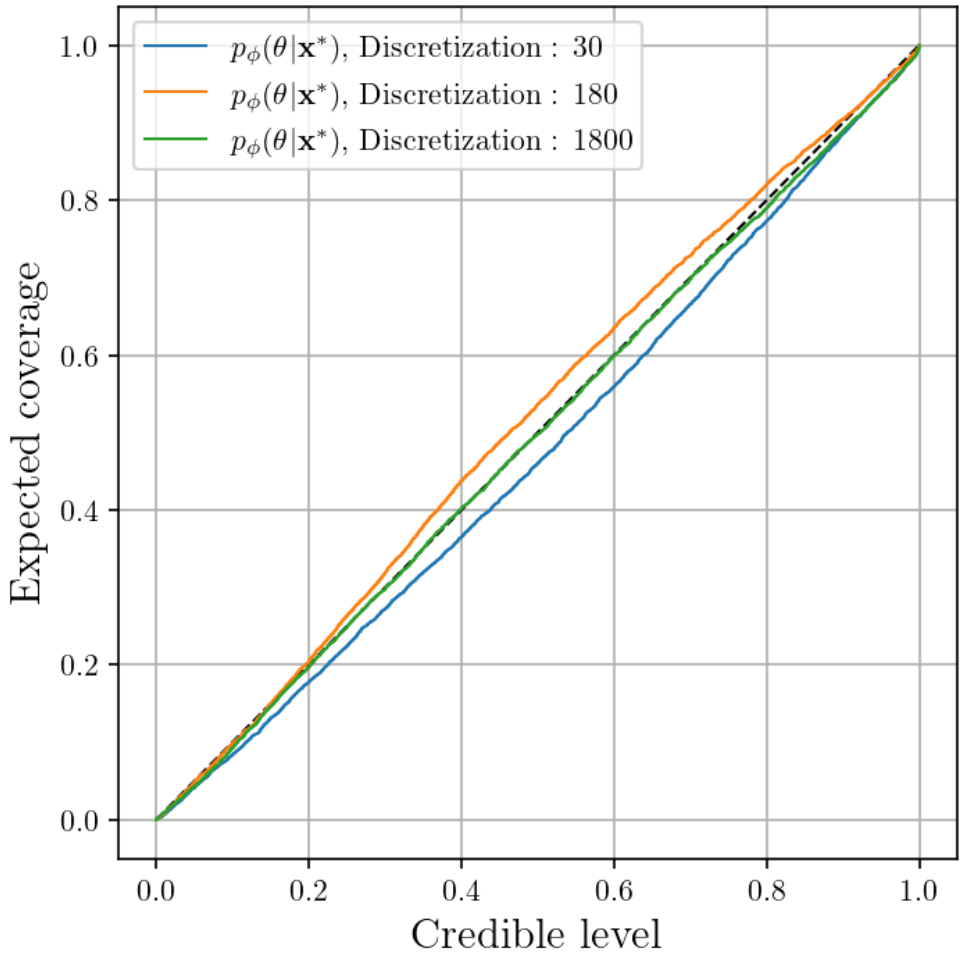

Figure 5.6. Coverage plot of the exoplanet HR 8799e with the effect of discretisation. The 1800 discretization is well calibrated while the 30 intervals discretization is underdispersed and the 180 intervals discretization is a bit overdispersed.

The coverage plot in Figure 5.6 provides valuable insights into the model's behavior across different discretization levels, revealing the trade-offs between computational efficiency and predictive accuracy. Firstly, at a low discretization level of 30 intervals, the plot indicates overdispersion. This overconfidence is also evident in the posterior predictive check plot (Figure 5.5) where the 68.3% credible interval is smaller but misaligned with the actual data. Consequently, the low discretization level fails to capture the underlying features of the data accurately, making the model unreliable.

Secondly, the coverage plot for 180 intervals shows a conservative approximation. The model slightly overestimates the uncertainty, which, while better than overconfidence, suggests that the model could be more precise. This level of discretization allows the model to learn the shape of the orbit better, although it still results in higher uncertainty due to the uniform error in the time of observations, leading to a conservative model.

In contrast, the highest level of discretization, 1800 intervals, shows well-calibrated results. This level of detail reduces the error in the time, leading to more reliable predictions. However, despite the improved accuracy, this discretization level comes with significant

computational and memory costs, requiring around 120 GB of memory for the training set.

5.4.2 Effect of the mass

As discussed in the previous section with the corner plot, the mass of the system is not well constrained by either of the three models purely from the data. However, the mass of the system is well known for stars in the solar neighborhood. While Gaia provides precise parallax measurements, the masses of these stars are typically determined through color measurements or spectroscopic observations by fitting stellar models to the data. Generally, stellar masses in the solar neighborhood are known to approximately 10-20% accuracy, depending on the spectral type considered. This means I may not have to use uninformative priors. We can thus use the mass of the system as input to the model and not as a parameter to be inferred. This would reduce the overall parameter space and make the training of the model easier. The mass of the system is still sampled from the prior for each data point, but it is used as input to the model by adding it to the input vector and not as a parameter to be inferred. It is also standardized to the interval $[-1, 1]$. This still is a large approximation as there is still some uncertainty on the mass of the system that is not kept into account and has to be kept in mind when interpreting the results.

Corner

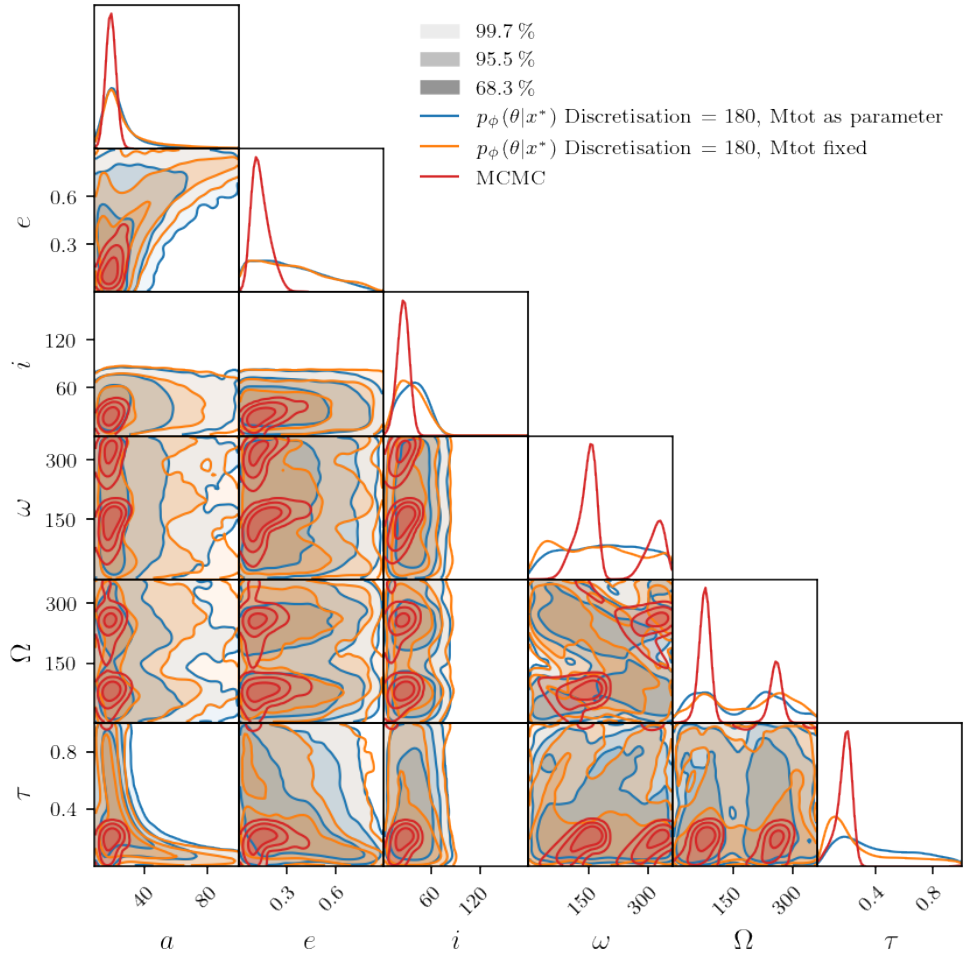

Figure 5.7. Corner plot of the posterior distribution of the orbital parameters of the exoplanet HR 8799e with M_{tot} fixed and as a parameter. A difference can be seen with the epoch of periastron passage τ and the inclination i. However, for the other parameters, no real difference can be seen.

The corner plot in Figure 5.7 shows the impact of taking the mass of the system as input and not as a parameter on the posterior distributions of the orbital parameters. Notably, this adjustment leads to the epoch of the periastron aligning more closely with the MCMC results. This outcome was expected, as the mass of the system directly influences the orbital period and, consequently, the epoch of the periastron. By fixing the mass, the variability in the period is reduced, resulting in a more accurate prediction of the periastron timing.

Additionally, the posterior distributions for the semi-major axis and eccentricity remain consistent, showing no significant deviation from the results obtained without fixing the mass. This consistency indicates that these parameters are less sensitive to the mass of the system and more robustly estimated by the model.

Posterior predictive check

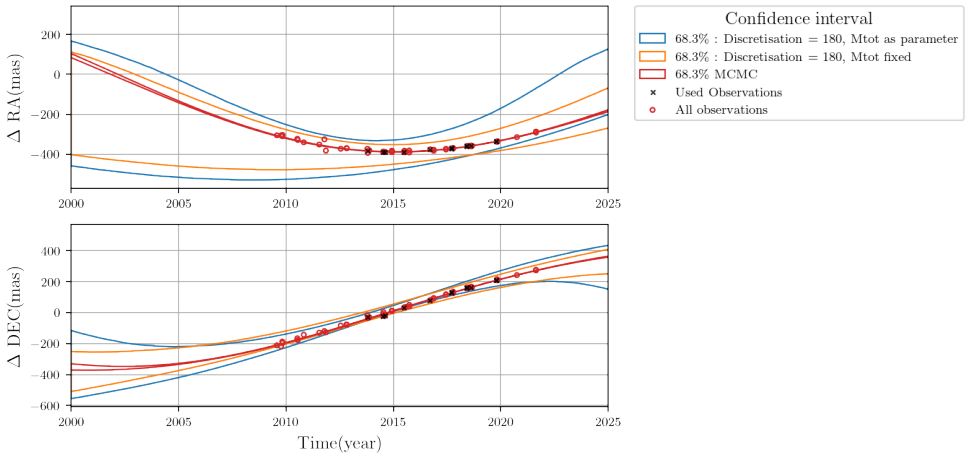

Figure 5.8. Posterior predictive check of the exoplanet HR 8799e with M_{tot} fixed and used as a parameter. The model is able to better predict the observations by fixing M_{tot} and is closer to the baseline MCMC results. The uncertainty is also reduced. Note that the observations are different from Figure 5.5 as in that Figure, the observations were changed due to the discretization. Indeed, they were set at the beginning of each interval.

In Figure 5.8, the model demonstrates an improved ability to predict the observations more accurately, centering closer to the MCMC orbit prediction. Fixing the mass leads to predictions that better align with the observed data, reducing the uncertainty in the predicted orbits.

This improvement is evident in the tighter credible intervals and the alignment of the predicted path with the actual observational data. By removing the mass of the system as a variable parameter and fixing it instead, the model gains a significant boost in predictive accuracy and reliability. This result confirms that incorporating known quantities, such as the total mass, enhances the model's performance.

Coverage plot

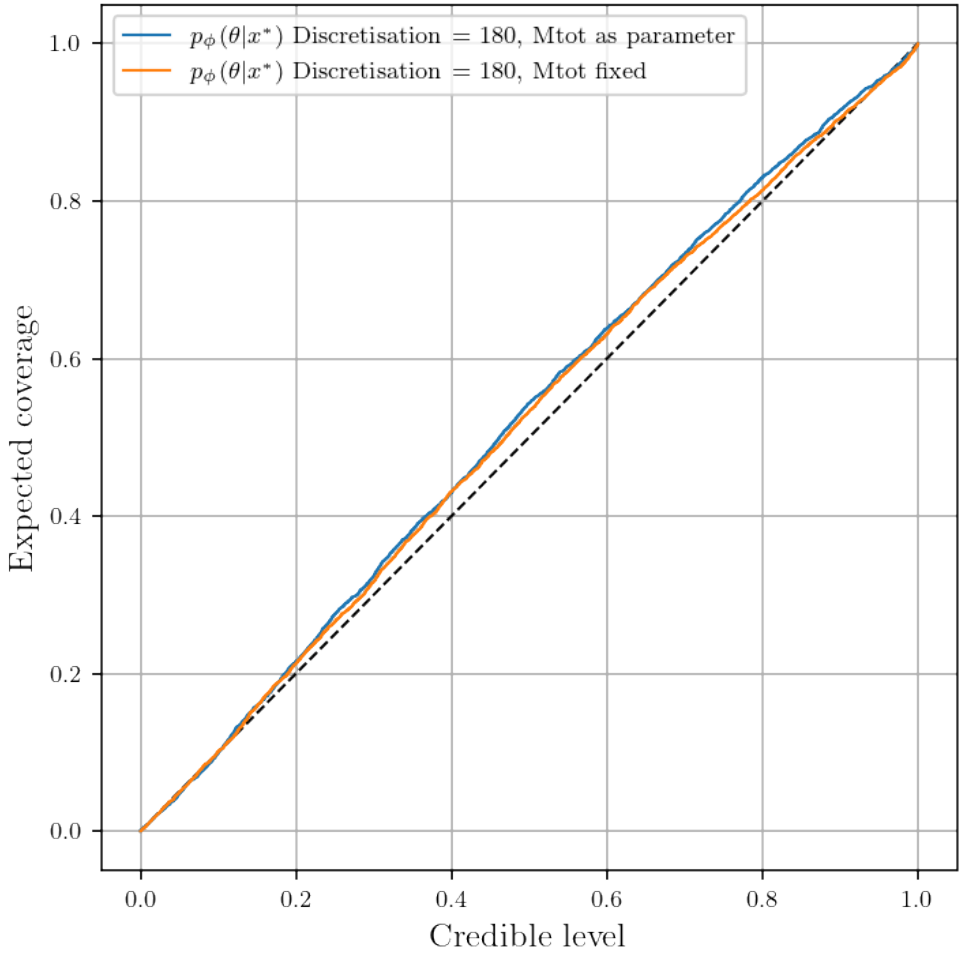

Figure 5.9. Coverage plot of the exoplanet HR 8799e with Mtot fixed

The coverage plot in Figure 5.9 shows that both methods exhibit good calibration, though the model with the fixed mass appears to be slightly less conservative.

The reduction in the 68.3% credible interval, as seen in Figure 5.8, is thus not due to overconfidence, but rather to a more accurate and constrained model. This outcome strengthens the case for using fixed known parameters to enhance the precision and reliability of exoplanet orbital predictions.

By validating the model on real data of HR 8799e and demonstrating improved predictive performance with fixed mass, we can conclude that this approach is beneficial. This method of fixing the total mass will be adopted in the next sections, ensuring that the model remains robust and accurate across different datasets and observational scenarios.

5.4.3 Reduced time period

One of the main problems with the model is that generating data with lower errors means using less available data at inference time. All data prior to 2014 were excluded due to their high error margins. By focusing on the period from 2014 to 2020, we can create a training set with an observational error of 2 milliarcseconds (mas), as this period coincides

with the use of more precise telescopes. This approach is applicable to both the HR 8799 exoplanets and the β Pictoris b. As more data become available in the future, the time period can be extended, such as from 2014 to 2024, but for this thesis, the focus remains on the 2014 to 2020 period.

The discretization is set to 100 intervals of equal length.

Corner plot

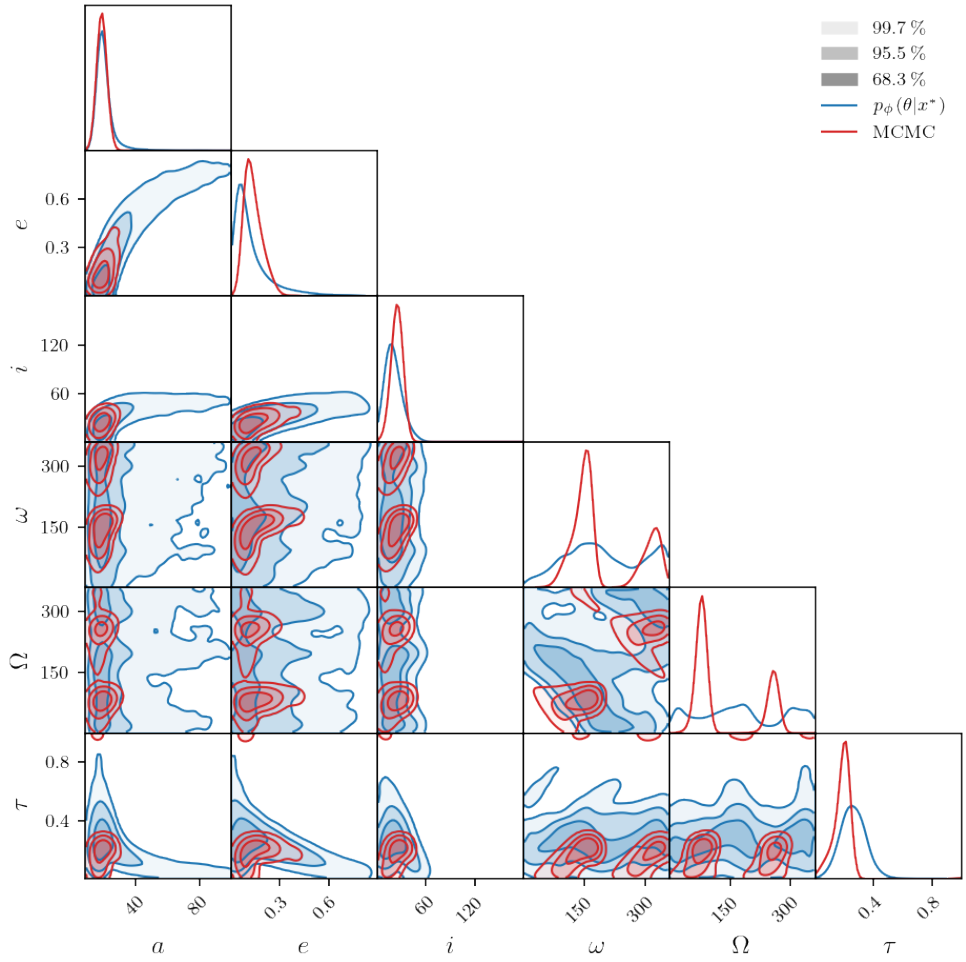

Figure 5.10. Corner plot of the posterior distribution of the orbital parameters of the exoplanet HR 8799e using MCMC and using a NPE trained on a training set with the time period reduced to 2014-2020. We can see that the NPE is able to better match the MCMC results than previously.

Figure 5.10 shows that the model can more closely predict the MCMC results than when the time period was from 2002 to 2020. However, there are still small discrepancies in the eccentricity, the inclination, and the epoch of the periastron compared to MCMC. This could be due to the fact that MCMC utilized all available observations, whereas my method relied on fewer data points based on specific assumptions. This comparison highlights the strength of MCMC in fully leveraging all observations, whereas my method demonstrates robustness with fewer data points. To ensure a fair comparison, MCMC should indeed use all observations as it is designed to handle them, unlike my method

which is constrained by its underlying assumptions. This allows MCMC to use a larger fraction of the orbit thereby allowing it to better understand the underlying features of the data.

I could not compare the results with OFTI as it is not able to handle the large amount of data that is available for the exoplanet HR 8799e. The algorithm stops when it has accepted a certain number of samples but with such a large amount of data, the percentage of accepted samples is really low and would take a lot of time to get a good approximation of the posterior distribution.

Posterior predictive check

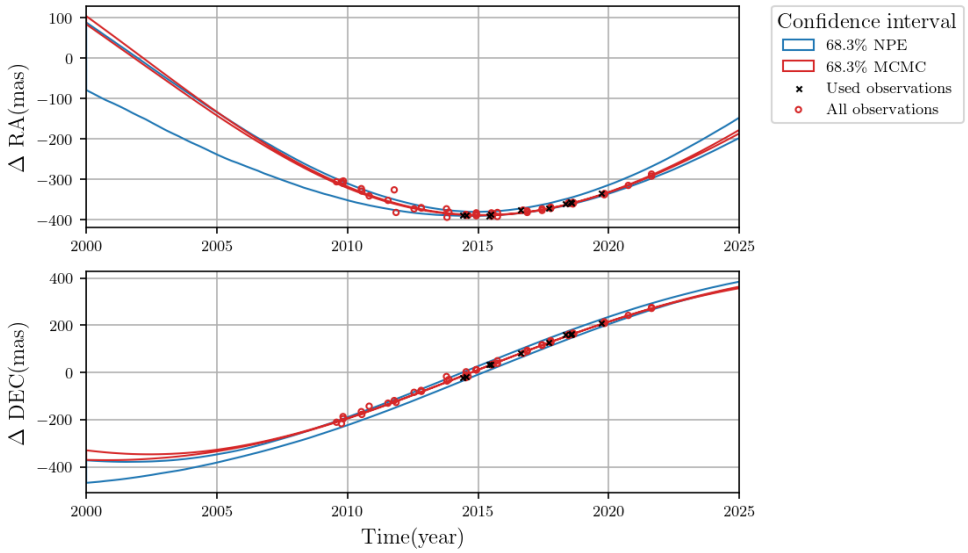

Figure 5.11. Posterior predictive check of the exoplanet HR 8799e. The NPE model is able to better predict the observations a lot better than previously. The uncertainty is greatly reduced but still not as good as for MCMC.

As was expected from the corner plot, the model is able to better predict the observations than when the time period was larger. This improvement is expected as the training set is more similar to the real observational data used.

The amortization of the inference procedure of the neural posterior estimator makes it possible to produce a posterior distribution of the orbital parameters of all the other exoplanets of the HR 8799 system instantaneously. The posterior predictive check of the 4 exoplanets of HR 8799 is shown in Figure 5.12. As we saw that the NPE was able to produce a good approximation of the orbit of HR 8799e, we can expect a similar result for the other exoplanets.

By comparing the results with those in the paper of Sepulveda and Bowler (2022) [40], the results are close to the ones they obtained by using MCMC for the other planets. The plot is shown in the Annex C.4.

However, we can see in Figure 5.12 that there are still some impossible orbits that are generated by the model.

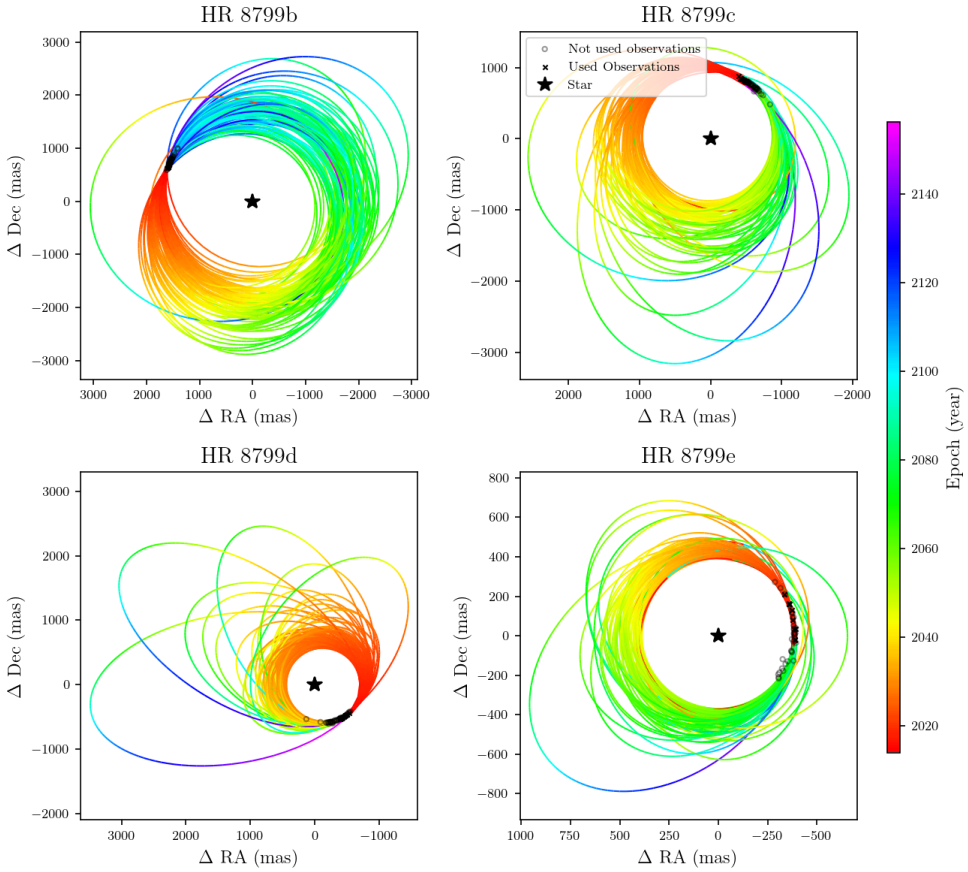

Figure 5.12. Posterior predictive check of the 4 exoplanets of HR 8799, 1000 orbits are generated for each exoplanet. The NPE produces a good approximation of those orbits, however, some impossible orbits are still generated like on HR8799b, HR8799c or HR8799e where there are orbits that are not passing through the observations. The x-axis has been inverted to make the plot comparable with the one in the paper of Sepulveda and Bowler (2022) [40].

However, the model is not able to correctly predict all the orbital parameters of β Pictoris b as can be seen in the Annex C.3 on Figure C.5.

This is because, by design, the idea of discretization reduces the amount of data available and may not closely resemble the actual observations of exoplanets.

Coverage plot

The coverage plot is also showing that we can trust the model but that it is still conservative.

With all of this we can conclude that even if the model was able to predict the orbit of the four exoplanets of HR 8799, it failed to predict the orbit of β Pictoris b. The problem may not be the choice in the hyperparameters but the model itself. The way of discretizing the time may not be the optimal way to make a generic model.

This is why I decided to try a different approach, the Deep Set network.

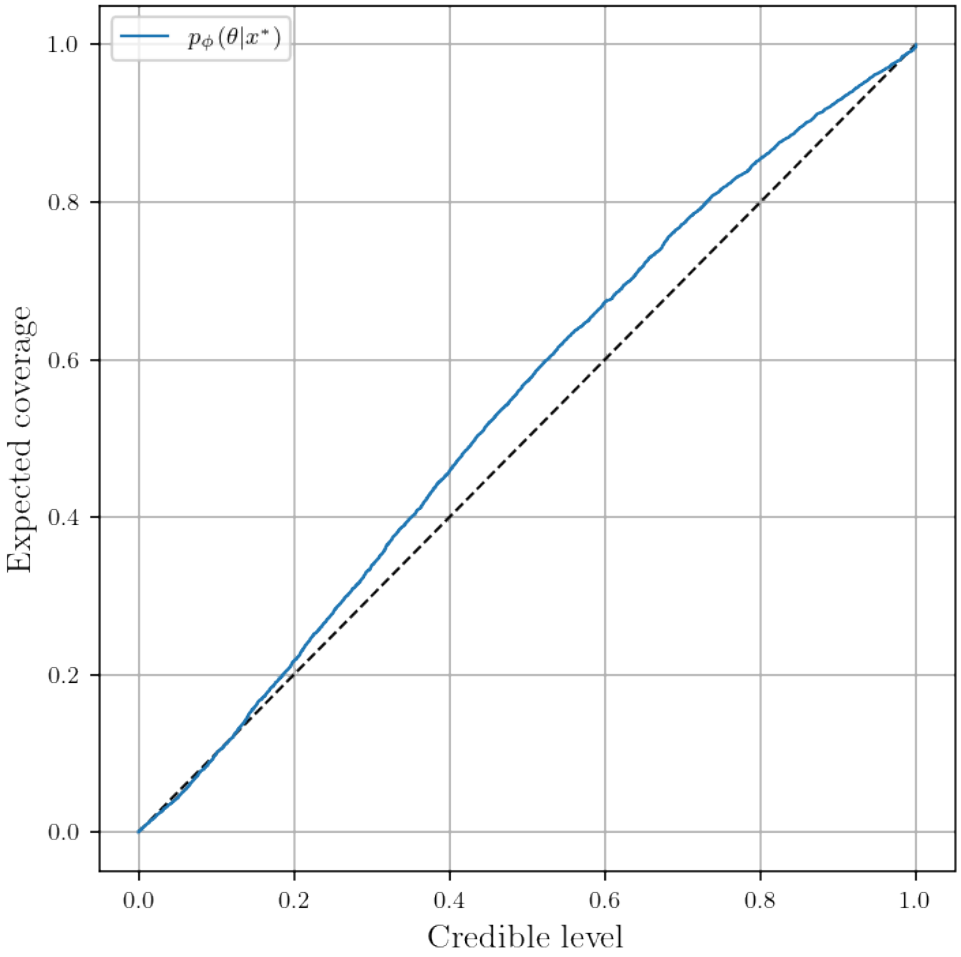

Figure 5.13. Coverage plot of the exoplanet HR 8799e with the time period reduced to 2014-2020. The model is almost calibrated but still a bit conservative.

5.5 Deep Set

The second architecture explored was the Deep Set network [3]. This network is designed to handle sets of inputs. Each input in the set is independently processed by a feature extractor network, and the resulting features are then aggregated to produce a single output. In our implementation, the aggregation is performed using a summation operation. This aggregated feature vector is subsequently passed to a second neural network, which generates the embedding vector. The overall architecture is illustrated in Figure 5.14.

One significant advantage of this approach is that the time does not need to be discretized. Instead, the timesteps are transformed into a vector of size 16 using the positional encoding scheme inspired by the Transformer model [42]. This positional encoding is designed to provide information about the relative positions of observations in time, utilizing sine and cosine functions of different frequencies. This should enable the model to understand the order of the input data.

The positional encoding is defined as follows:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right) \tag{5.1}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{5.2}$$

where pos is the position of the time in the sequence. To determine pos, we set the first element of the sequence to 52,275, which corresponds to January 1, 2002. For each time input, we calculate the difference between the given time and 52,275, and use this as the position. The dimension d of the embedding is set to 16 in this case.

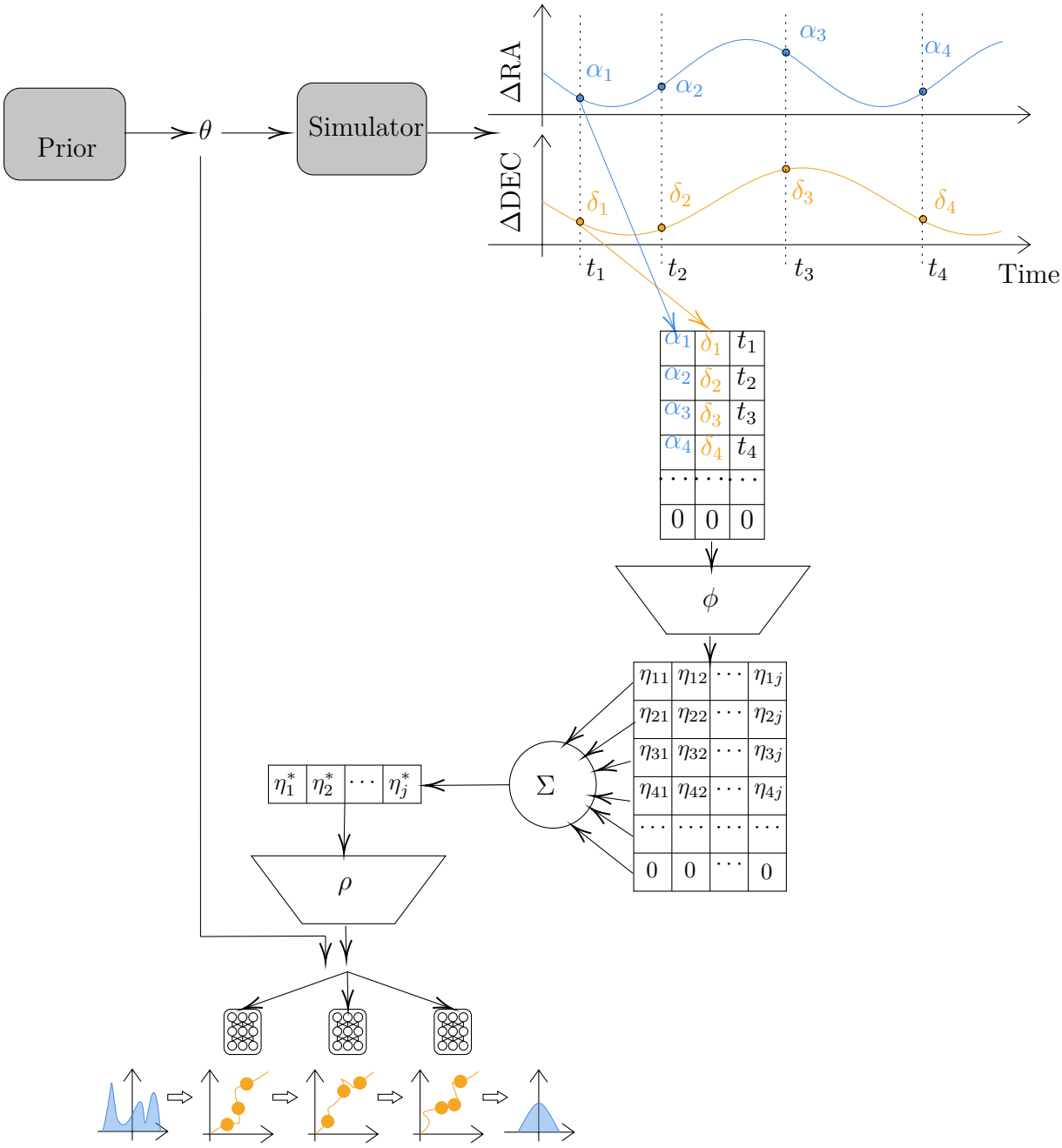

Figure 5.14. Architecture of the generic model using a Deepset as an embedding network. The model takes a set of inputs, each being the RA and DEC position and the time of the observation. Each of these inputs is embedded using the feature extractor network ϕ producing the set of features η. A maximum of 30 observations are taken, if there are less, the last embeddings are set to 0 to match the number of observations. For example if there are 20 observations, the last 10 embeddings are set to 0. This set of aggregated features is then aggregated using a sum operation. The aggregated features are then passed to the regressor network ρ. The output is then used to train the normalizing flow. All of the t_i are vectors obtained using the positional encoding from the *Attention is all you need* paper[42].

5.5.1 Results

Different architectures were tested, varying the sizes and shapes of the two networks used as the embedding network. We also experimented with different methods of encoding time. Unfortunately, the best model, which utilized a feature extractor with layers of 16, 32, and 64 neurons, and a regressor with layers of 128, 256, 128, and 64 neurons, did not perform as well as the previous ResMLP model. Deeper or wider layers resulted in significantly longer training times compared to the ResMLP model, with only marginal improvements in performance.

The results of the best model are shown in the corner plot in Figure 5.15. This model was trained for 512 epochs using the same training procedure as described in Section 5.4. The corner plot reveals that the model fails to accurately predict the true values of the parameters, particularly missing the posterior distributions of the eccentricity and inclination, rendering the model ineffective.

This shortcoming may be attributed to the nature of the data, which is not inherently a set but a sequence. Despite using a time vector, the model struggles to grasp the sequential order of observations. In contrast, the ResMLP model retains sequence information, as each position of the input vector corresponds to a distinct observation time, thereby maintaining the temporal order.

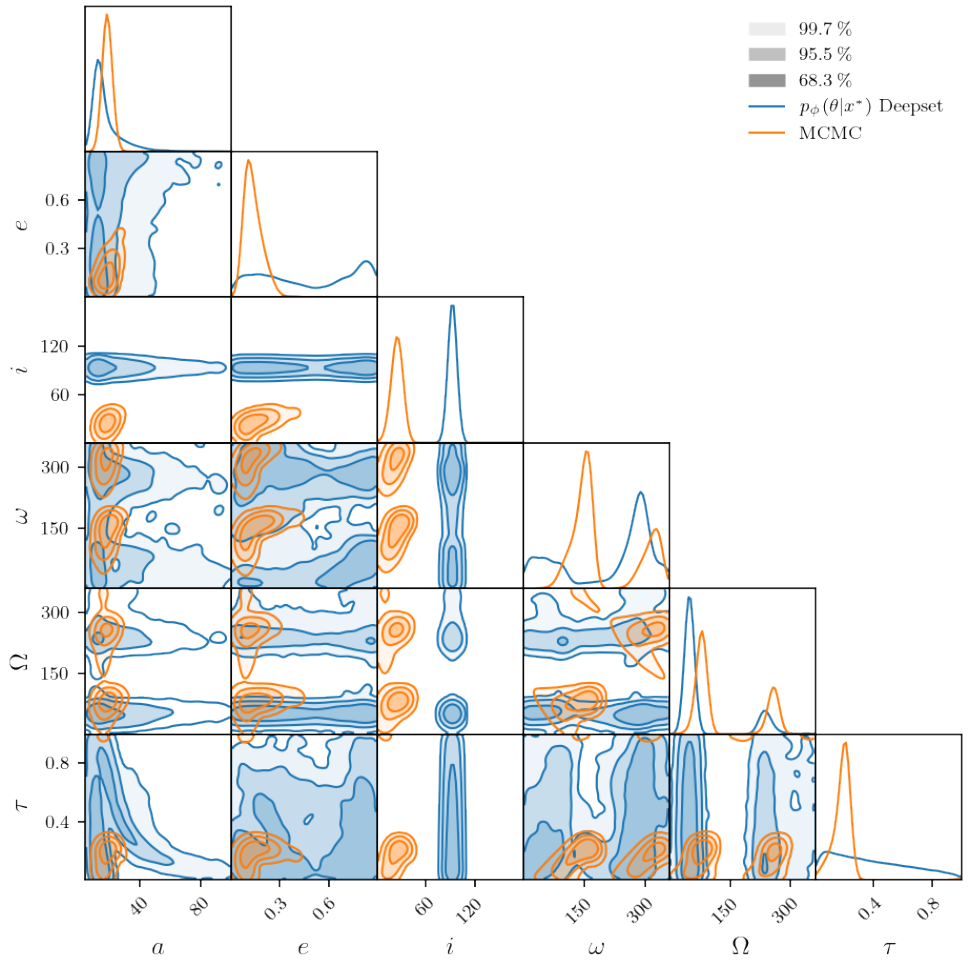

Figure 5.15. Corner plot of the generic model using a Deep Set as the embedding network. The model is not able to correctly predict the posterior distributions of the parameters.

Chapter 6

Conclusion

The objective of this thesis was to explore and enhance the characterization of orbital parameters of exoplanets using advancements in deep learning, specifically through simulation-based inference.

Characterizing the orbital parameters of exoplanets solely from astrometric data is a challenging task but can provide valuable insights into the dynamics of exoplanetary systems. Current methods, implemented in libraries such as `orbitize!` or `Orvara`, rely on Markov Chain Monte Carlo (MCMC) methods or Approximate Bayesian Computation to infer the posterior distributions of these parameters given the data. These methods are computationally expensive and can be slow to converge.

The first part of this work involved reproducing the results of the state-of-the-art method, the α-DPI, which employs variational inference and a normalizing flow, RealNVP. In this thesis, Neural Spline Flows are used along with the expected forward Kullback-Leibler divergence allowed us to avoid computing the likelihood of the data for the loss function used to train the model.

The results are slightly more uncertain than those obtained with α-DPI and MCMC using all available observations of the exoplanet β-Pic b. However, by examining the calibration of our NPE, which is perfectly calibrated, we can infer that the results of MCMC are actually slightly overconfident.

Like α-DPI, our NPE model was not fully amortized. Although inference is possible on different datasets, the datasets need to have astrometric observations at exactly the same times as the observations for β-Pic b because the model is designed this way. Additionally, the priors chosen did not encompass a sufficiently large parameter space, which would be necessary for a more general model. This lack of amortization implies that if new observations of β-Pic b were added or if we wanted to use the model for another exoplanet, a new model would have to be retrained from scratch. This is where the second part of this thesis comes into play: designing an architecture that can be used for any exoplanet.

The first architecture developed for this second part used a ResMLP as the embedding network, which takes the observations of the exoplanet and outputs a vector that is used by the normalizing flow to infer the posterior of the orbital parameters. The input vector of the embedding network is a period of time that is discretized into a certain number of bins. If an observation falls into a bin, the value of the bin is set to the values of the

observation; otherwise, it is set to zero.

Several experiments were conducted to enhance the model, such as testing the number of bins, setting the mass of the exoplanet as an input to the embedding network instead of as a parameter of the normalizing flow, and reducing the period of time to use only recent observations from telescopes with better precision and thus better data.

The model was able to infer the posterior of the orbital parameters of all four planets of the HR 8799 system, which would have required running MCMC for each planet separately. However, the experiments may have been too tailored to the HR 8799 system and not sufficiently general, as the model could not infer the posterior of the orbital parameters of β-Pic b.

A second architecture was designed using a Deep Set network as the embedding network. However, this architecture performed worse than the ResMLP. This could be due to the fact that the observations are not independent of each other and cannot be treated as a set.

In conclusion, it is possible to create a program that trains a model based on the observations of an exoplanet and can infer the posterior of the orbital parameters, significantly reducing the time required and enabling the use of large datasets, which OFTI cannot. Developing a truly generic model is more complex, as many assumptions are made in the design of the model, potentially rendering the results less reliable.

6.1 Future Work

Future research could focus on utilizing more advanced simulators, such as the one provided by `Orvara`, to generate data more efficiently. The `Orvara` simulator is known for its high precision and speed, which would enable the creation of more accurate training datasets in a shorter amount of time. By improving the quality and quantity of the training data, the performance of the models could be significantly enhanced. This would be particularly beneficial for developing models that can generalize across a wide range of exoplanetary systems.

Incorporating a physically motivated distribution for eccentricity could yield more realistic results. The eccentricity distribution of exoplanets is known to follow a beta distribution, as demonstrated by Kipping *et al.* (2013)[43] and verified for directly imaged planets by Bowler *et al.* (2020)[44]. This could potentially improve the precision.

Revisiting the prior ranges for certain parameters could enhance model efficiency. Specifically, the priors for the argument of periastron (ω) and the longitude of the ascending node (Ω) are currently set between 0° and 360°. However, for reasons of symmetry, one of these parameters could potentially be reduced to a range of 0° to 180°. Clarifying and adjusting these priors could reduce computational overhead.

Another promising direction is to explore the transformer architecture as an embedding network. The self-attention mechanism inherent in transformers could effectively capture the dependencies between observations. Unlike traditional architectures that may struggle with long-range dependencies, transformers can process all input data simultaneously, considering the relationships between all pairs of observations. This capability is crucial for

accurately modeling the sequential nature of astrometric data and could lead to substantial improvements in the inference of orbital parameters.

Incorporating additional data points, such as radial velocity measurements, could also enhance the model's ability to constrain the parameter space. Radial velocity data provides complementary information about the motion of exoplanets, offering insights into their masses and orbital characteristics that are not captured by astrometric data alone. By integrating both astrometric and radial velocity observations, the model could achieve a more comprehensive understanding of the exoplanetary systems, leading to more accurate and reliable parameter estimates. This multi-modal approach would leverage the strengths of different types of observations to produce a more robust inference framework.

Bibliography

[1] He Sun et al. "alpha-Deep Probabilistic Inference (alpha-DPI): efficient uncertainty quantification from exoplanet astrometry to black hole feature extraction". In: *arXiv preprint arXiv:2201.08506* (2022) (pages 1, 4, 7, 8, 10, 14, 24, 28, 30).

[2] Hugo Touvron et al. "ResMLP: Feedforward networks for image classification with data-efficient training". 2021. arXiv: 2105.03404 [cs.CV] (pages 1, 41).

[3] Manzil Zaheer et al. "Deep Sets". 2018. arXiv: 1703.06114 [cs.LG] (pages 2, 55).

[4] Michel Mayor and Didier Queloz. "A Jupiter-mass companion to a solar-type star". In: *Nature* 378.6555 (Nov. 1995), pp. 355–359. DOI: 10.1038/378355a0 (page 3).

[5] J.-L. Beuzit et al. "SPHERE: the exoplanet imager for the Very Large Telescope". In: *Astronomy amp; Astrophysics* 631 (Nov. 2019), A155. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201935251. URL: http://dx.doi.org/10.1051/0004-6361/201935251 (page 3).

[6] Brendan P. Bowler. "Imaging Extrasolar Giant Planets". In: *Publications of the Astronomical Society of the Pacific* 128.968 (Aug. 2016), p. 102001. ISSN: 1538-3873. DOI: 10.1088/1538-3873/128/968/102001. URL: http://dx.doi.org/10.1088/1538-3873/128/968/102001 (pages 4, 6, 12).

[7] Sarah Blunt et al. "Orbits for the Impatient: A Bayesian Rejection-sampling Method for Quickly Fitting the Orbits of Long-period Exoplanets". In: *The Astronomical Journal* 153.5 (Apr. 2017), p. 229. ISSN: 1538-3881. DOI: 10.3847/1538-3881/aa6930. URL: http://dx.doi.org/10.3847/1538-3881/aa6930 (pages 4, 8, 11, 12).

[8] Thayne Currie et al. "Direct Imaging and Spectroscopy of Extrasolar Planets". 2023. arXiv: 2205.05696 [astro-ph.EP] (page 6).

[9] Anne-Lise Maire et al. "Workshop Summary: Exoplanet Orbits and Dynamics". In: *Publications of the Astronomical Society of the Pacific* 135.1052 (Nov. 2023), p. 106001. DOI: 10.1088/1538-3873/acff88. URL: https://dx.doi.org/10.1088/1538-3873/acff88 (pages 6, 26).

[10] Sarah Blunt et al. "orbitize!: A Comprehensive Orbit-fitting Software Package for the High-contrast Imaging Community". In: *The Astronomical Journal* 159.3 (Feb. 2020), p. 89. ISSN: 1538-3881. DOI: 10.3847/1538-3881/ab6663. URL: http://dx.doi.org/10.3847/1538-3881/ab6663 (pages 6, 8, 9, 11, 25).

[11] "Modified Julian Dates". https://core2.gsfc.nasa.gov/time/. [Accessed 28-05-2024] (page 6).

[12] "Keplerian Elements". https://www.amsat.org/keplerian-elements-tutorial/. [Accessed 28-03-2024] (page 8).

[13] "Orbital Elements". https://taitussoftware.com/kb/orbital-elements/. [Accessed 28-03-2024] (page 8).

[14] Michael Perryman. "The Exoplanet Handbook". 2nd ed. Cambridge University Press, 2018 (page 9).

[15] S. Mikkola. "A cubic approximation for Kepler's equation". In: *Celestial Mechanics 40* (1987). DOI: 10.1007/BF01235850. URL: https://doi.org/10.1007/BF01235850 (page 9).

[16] Timothy D. Brandt et al. "orvara: An Efficient Code to Fit Orbits Using Radial Velocity, Absolute, and/or Relative Astrometry". In: *The Astronomical Journal* 162.5 (Oct. 2021), p. 186. ISSN: 1538-3881. DOI: 10.3847/1538-3881/ac042e. URL: http://dx.doi.org/10.3847/1538-3881/ac042e (page 9).

[17] William I. Hartkopf, Harold A. McAlister, and Otto G. Franz. "Binary Star Orbits from Speckle Interferometry. II. Combined Visual/Speckle Orbits of 28 Close Systems". In: *aj* 98 (Sept. 1989), p. 1014. DOI: 10.1086/115193 (page 9).

[18] Thierry Forveille et al. "Accurate masses of very low mass stars: I Gl 570BC (0.6+0.4 Msol)". 1999. arXiv: astro-ph/9909342 [astro-ph] (page 9).

[19] Daniel Foreman-Mackey et al. "emcee: The MCMC Hammer". In: *Publications of the Astronomical Society of the Pacific* 125.925 (Mar. 2013), pp. 306–312. ISSN: 1538-3873. DOI: 10.1086/670067. URL: http://dx.doi.org/10.1086/670067 (page 11).

[20] W. D. Vousden, W. M. Farr, and I. Mandel. "Dynamic temperature selection for parallel tempering in Markov chain Monte Carlo simulations". In: *Monthly Notices of the Royal Astronomical Society* 455.2 (Nov. 2015), pp. 1919–1937. ISSN: 1365-2966. DOI: 10.1093/mnras/stv2422. URL: http://dx.doi.org/10.1093/mnras/stv2422 (page 11).

[21] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117.48 (May 2020), pp. 30055–30062. ISSN: 1091-6490. DOI: 10.1073/pnas.1912789117. URL: http://dx.doi.org/10.1073/pnas.1912789117 (page 11).

[22] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. ISSN: 00034851. URL: http://www.jstor.org/stable/2236703 (visited on 04/25/2024) (pages 15, 17).

[23] Andy Jones. "KL(q||p) is mode-seeking". https://andrewcharlesjones.github.io/journal/klqp.html. [Accessed 07-06-2024] (page 15).

[24] Yingzhen Li and Richard E. Turner. "Rényi Divergence Variational Inference". 2016. arXiv: 1602.02311 [stat.ML] (page 15).

[25] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. "Density estimation using Real NVP". 2017. arXiv: 1605.08803 [cs.LG] (pages 15, 18, 28).

[26] Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. "Normalizing Flows: An Introduction and Review of Current Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.11 (Nov. 2021), pp. 3964–3979. ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2992934. URL: http://dx.doi.org/10.1109/TPAMI.2020.2992934 (pages 15, 18).

[27] Conor Durkan et al. “Neural Spline Flows”. 2019. arXiv: 1906.04032 [stat.ML] (pages 15, 18–21, 27).

[28] George Papamakarios et al. “Normalizing Flows for Probabilistic Modeling and Inference”. 2021. arXiv: 1912.02762 [stat.ML] (pages 17, 18).

[29] Malavika Vasist et al. “Neural posterior estimation for exoplanetary atmospheric retrieval”. In: *Astronomy & Astrophysics* 672 (Apr. 2023), A147. ISSN: 1432-0746. DOI: 10.1051/0004-6361/202245263. URL: http://dx.doi.org/10.1051/0004-6361/202245263 (pages 17, 27, 28, 42).

[30] Joeri Hermans et al. “A Trust Crisis In Simulation-Based Inference? Your Posterior Approximations Can Be Unfaithful”. 2022. arXiv: 2110.06581 [stat.ML] (page 21).

[31] George Papamakarios, Theo Pavlakou, and Iain Murray. “Masked Autoregressive Flow for Density Estimation”. 2018. arXiv: 1705.07057 [stat.ML] (page 27).

[32] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)”. 2016. arXiv: 1511.07289 [cs.LG] (page 27).

[33] François Rozet, Arnaud Delaunoy, Benjamin Miller, et al. “LAMPE: Likelihood-free Amortized Posterior Estimation”. 2021. DOI: 10.5281/zenodo.8405782. URL: https://pypi.org/project/lampe (page 27).

[34] François Rozet et al. “Zuko: Normalizing flows in PyTorch”. 2022. DOI: 10.5281/zenodo.7625672. URL: https://pypi.org/project/zuko (page 27).

[35] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. 2019. arXiv: 1711.05101 [cs.LG] (page 28).

[36] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. 2017. arXiv: 1412.6980 [cs.LG] (page 28).

[37] Laurent Dinh, David Krueger, and Yoshua Bengio. “NICE: Non-linear Independent Components Estimation”. 2015. arXiv: 1410.8516 [cs.LG] (page 33).

[38] Christian Marois et al. “Images of a fourth planet orbiting HR 8799”. In: *Nature* 468.7327 (Dec. 2010), pp. 1080–1083. ISSN: 1476-4687. DOI: 10.1038/nature09684. URL: http://dx.doi.org/10.1038/nature09684 (page 39).

[39] Clarissa R. Do Ó et al. “The Orbital Eccentricities of Directly Imaged Companions Using Observable-based Priors: Implications for Population-level Distributions”. In: *The Astronomical Journal* 166.2 (July 2023), p. 48. ISSN: 1538-3881. DOI: 10.3847/1538-3881/acdc9a. URL: http://dx.doi.org/10.3847/1538-3881/acdc9a (page 41).

[40] Aldo G. Sepulveda and Brendan P. Bowler. “Dynamical Mass of the Exoplanet Host Star HR 8799”. In: *The Astronomical Journal* 163.2 (Jan. 2022), p. 52. ISSN: 1538-3881. DOI: 10.3847/1538-3881/ac3bb5. URL: http://dx.doi.org/10.3847/1538-3881/ac3bb5 (pages 41, 52, 53, 75, 77).

[41] A. Zurlo et al. “Orbital and dynamical analysis of the system around HR 8799: New astrometric epochs from VLT/SPHERE and LBT/LUCI”. In: *Astronomy & Astrophysics* 666 (Oct. 2022), A133. ISSN: 1432-0746. DOI: 10.1051/0004-6361/202243862. URL: http://dx.doi.org/10.1051/0004-6361/202243862 (pages 41, 67–69).

[42] Ashish Vaswani et al. “Attention Is All You Need”. 2023. arXiv: 1706.03762 [cs.CL] (pages 55, 56).

[43] David M. Kipping. “Parametrizing the exoplanet eccentricity distribution with the Beta distribution”. In: *Monthly Notices of the Royal Astronomical Society: Letters* 434.1 (June 2013), pp. L51–L55. ISSN: 1745-3925. DOI: 10.1093/mnrasl/slt075. eprint: https://academic.oup.com/mnrasl/article-pdf/434/1/L51/54658174/mnrasl_434_1_l51.pdf. URL: https://doi.org/10.1093/mnrasl/slt075 (page 59).

[44] Brendan P. Bowler, Sarah C. Blunt, and Eric L. Nielsen. “Population-level Eccentricity Distributions of Imaged Exoplanets and Brown Dwarf Companions: Dynamical Evidence for Distinct Formation Channels*”. In: *The Astronomical Journal* 159.2 (Jan. 2020), p. 63. DOI: 10.3847/1538-3881/ab5b11. URL: https://dx.doi.org/10.3847/1538-3881/ab5b11 (page 59).

Appendix A

β-pic observations

Table A.1: Observations of the separation and position angle of β-pic b relative to the primary star β-pic

Epoch	**Sep. (mas)**	**Sep. Error**	**PA**	**PA Error (°)**
52953	413	22	34	4
54781	210	27	211.49	1.9
55129	299	14	211	3
55168	339	10	209.2	1.7
55168	323	10	209.3	1.8
55194	306	9	212.1	1.7
55296	346	7	209.9	1.2
55467	383	11	210.3	1.7
55516	387	8	212.4	1.4
55517	390	13	212	2
55555	407	5	212.8	1.4
55593	408	9	211.1	1.5
55646	426	13	210.1	1.8
55854	452	3	211.6	0.4
55854	455	5	211.9	0.6
56015	447	3	210.8	0.4
56015	448	5	211.8	0.6
56263	461	14	211.9	1.2
56265	470	10	212	1.2
56612	430.8	1.5	212.43	0.17
58440	164.5	1.8	28.64	0.7

Continued on next page

Table A.1: Observations of the separation and position angle of β-pic b relative to the primary star β-pic (Continued)

Epoch	**Sep. (mas)**	**Sep. Error**	**PA**	**PA Error (°)**
56612	429.1	1	212.58	0.15
56614	430.2	1	212.46	0.15
56636	425.5	1	212.51	0.15
56636	424.4	1	212.85	0.15
56637	425.3	1	212.47	0.16
56969	356.2	1	213.02	0.19
57046	335.5	0.9	212.88	0.2
57114	317.3	0.9	213.13	0.2
57332	250.5	1.5	214.14	0.34
57361	240.2	1.1	213.58	0.34
57378	234.5	1	213.81	0.3
57408	222.6	2.1	214.84	0.44
58382	141.9	5.3	28.16	1.82
58440	164.5	1.8	28.64	0.7

Appendix B

HR 8799 Observations

Table B.1: Observations of the Right Ascension and Declination of HR8799 bcde relative to the primary star HR8799 . The values are in mas and the error of each observation is given in the next column. The observations comes from Zurlo *et al.* (2022) [41]

	Planet b				Planet c				Planet d				Planet e			
Epoch	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err
51117	1411	9	986	9	-	-	-	-	-	-	-	-	-	-	-	-
51117	1418	22	1004	20	-837	26	483	23	133	35	-533	34	-	-	-	-
52472	1481	23	919	17	-	-	-	-	-	-	-	-	-	-	-	-
53199	1471	6	884	6	-739	6	612	6	-	-	-	-	-	-	-	-
53568	1496	5	856	5	-713	5	630	5	-87	10	-578	10	-	-	-	-
54313	1504	3	837	3	-683	4	671	4	-179	5	-588	5	-	-	-	-
54397	1500	7	836	7	-678	7	676	7	-175	10	-589	10	-	-	-	-
54656	1527	4	799	4	-658	4	701	4	-208	4	-582	4	-	-	-	-
54689	1527	2	801	2	-657	2	706	2	-216	2	-582	2	-	-	-	-
54726	1516	4	818	4	-663	3	693	3	-202	4	-588	4	-	-	-	-
54792	1532	20	796	20	-654	20	700	20	-217	20	-608	20	-	-	-	-
54839	-	-	-	-	-612	30	665	30	-	-	-	-	-	-	-	-
55044	1526	4	797	4	-639	4	712	4	-237	3	-577	3	-306	7	-211	7
55058	1536	10	785	10	-	-	-	-	-	-	-	-	-	-	-	-
55088	1538	30	777	30	-634	30	697	30	-282	30	-590	30	-	-	-	-
55109	1535	20	816	20	-636	40	692	40	-270	70	-600	70	-	-	-	-
55113	1532	7	783	7	-627	7	716	7	-241	7	-586	7	-306	7	-217	7
55135	1524	10	795	10	-636	9	720	9	-251	7	-573	7	-310	9	-187	9
59449	1626	1	578	2	-339	2	890	2	-563	2	-391	2	-287	4	272	2

Continued on next page

Table B.1: Observations of the Right Ascension and Declination of HR8799 bcde relative to the primary star HR8799 . The values are in mas and the error of each observation is given in the next column. The observations comes from Zurlo *et al.* (2022) [41] (Continued)

	Planet b				Planet c				Planet d				Planet e			
Epoch	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err
55139	1540	19	800	19	-630	13	720	13	-240	14	-580	14	-304	10	-196	10
55390	1532	5	783	5	-619	4	728	4	-265	4	-576	4	-323	6	-166	6
55398	1547	6	757	9	-606	6	725	6	-269	6	-580	6	-329	6	-178	6
55500	1535	15	766	15	-607	12	744	12	-296	13	-561	13	-341	16	-143	16
55763	1541	5	762	5	-595	4	747	4	-303	5	-562	5	-352	8	-130	8
55850	1579	11	734	11	-561	13	752	13	-299	13	-563	13	-326	13	-119	13
55876	1546	11	725	11	-578	13	767	13	-320	13	-549	13	-382	16	-127	16
56128	1545	5	747	5	-578	5	761	5	-339	5	-555	5	-373	8	-84	8
56227	1549	4	743	4	-572	3	768	3	-346	4	-548	4	-370	9	-76	9
56231	1558	6	729	9	-557	6	763	6	-343	6	-555	6	-371	6	-80	6
56581	1545	22	724	22	-542	22	784	22	-382	16	-522	16	-373	13	-17	13
56589	1562	8	713	13	-538	6	784	13	-377	7	-538	11	-394	11	-36	17
56614	-	-	-	-	-537	1	782	2	-370	1	-539	1	-381	2	-30	0
56851	-	-	-	-	-	-	-	-	-400	4	-512	4	-389	1	-22	2
56851	1570	3	704	3	-521	3	790	9	-391	2	-530	2	-387	2	-10	3
56855	1560	13	725	13	-540	13	799	13	-400	11	-534	11	-387	11	3	11
56884	-	-	-	-	-	-	-	-	-396	1	-524	2	-389	1	-17	2
56914	1569	4	707	2	-519	1	794	2	-397	1	-530	2	-	-	-	-
56997	1575	2	702	4	-511	2	799	2	-400	2	-523	2	-385	3	12	2
56997	1574	3	701	2	-514	3	798	4	-399	4	-525	4	-389	8	11	4
56997	1574	4	701	3	-512	3	798	4	-400	4	-523	4	-390	7	12	4
56997	1573	3	701	3	-512	3	797	4	-403	4	-524	4	-383	8	11	4
57209	-	-	-	-	-	-	-	-	-424	4	-509	3	-391	1	33	2
57209	1579	1	694	1	-498	1	806	1	-417	1	-517	1	-383	9	33	5
57235	1580	5	689	3	-495	2	806	2	-419	2	-516	1	-386	1	36	1
57260	1569	11	666	7	-482	5	813	6	-436	11	-510	12	-	-	-	-
57293	-	-	-	-	-	-	-	-	-420	4	-513	4	-392	1	39	3
57293	1580	1	688	1	-494	1	811	1	-426	1	-512	1	-382	9	50	5
57652	-	-	-	-	-	-	-	-	-466	1	821	2	-453	1	-376	2
57710	-	-	-	-	-	-	-	-	-464	1	-486	2	-382	2	94	6
59449	1626	1	578	2	-339	2	890	2	-563	2	-391	2	-287	4	272	2

Continued on next page

Table B.1: Observations of the Right Ascension and Declination of HR8799 bcde relative to the primary star HR8799 . The values are in mas and the error of each observation is given in the next column. The observations comes from Zurlo *et al.* (2022) [41] (Continued)

	Planet b				**Planet c**				**Planet d**				**Planet e**			
Epoch	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err	ΔRA	Err	ΔDEC	Err
57710	1589	2	666	1	-464	1	824	2	-454	2	-489	2	-378	4	90	2
57918	-	-	-	-	-	-	-	-	-473	2	-476	2	-377	1	115	3
57918	1591	1	653	1	-449	1	835	1	-472	2	-482	2	-373	3	118	2
58039	-	-	-	-	-	-	-	-	-480	2	-478	2	-372	1	129	2
58039	1595	1	647	1	-441	1	839	1	-480	1	-477	1	-369	1	128	1
58039	-	-	-	-	-	-	-	-	-492	5	-463	6	-370	2	135	3
58039	1595	1	647	1	-441	1	839	1	-480	1	-477	1	-371	2	128	2
58287	-	-	-	-	-	-	-	-	-495	1	-460	2	-360	2	162	2
58287	1601	1	635	1	-424	1	848	1	-497	2	-463	2	-358	2	156	2
58349	-	-	-	-	-	-	-	-	-509	2	-452	3	-361	2	166	2
58349	1601	2	632	3	-421	1	850	1	-502	1	-461	1	-357	1	162	2
58349	-	-	-	-	-	-	-	-	-503	2	-456	2	-359	1	167	2
58349	1600	1	632	1	-421	1	851	1	-502	2	-458	1	-358	2	163	1
58360	-	-	-	-	-	-	-	-	-	-	-	-	-358	0	163	0
58787	-	-	-	-	-	-	-	-	-527	3	-432	3	-337	3	210	3
58787	1606	2	615	2	-392	2	875	2	-532	3	-425	2	-338	2	215	2
58791	-	-	-	-	-	-	-	-	-528	1	-435	2	-337	2	208	2
58791	1611	1	611	1	-388	2	870	2	-530	2	-430	1	-335	1	210	1
59124	1620	1	591	3	-364	2	883	1	-551	1	-415	4	-315	3	242	5
59449	-	-	-	-	-	-	-	-	-569	1	-390	2	-292	2	276	2
59449	1626	1	578	2	-339	2	890	2	-563	2	-391	2	-287	4	272	2

Appendix C

Additional plots

C.1 NPE with a longer training

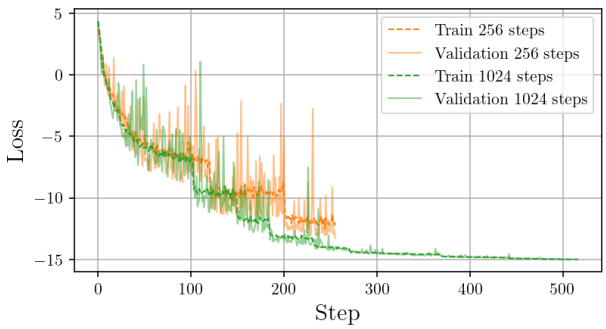

Figure C.1. Loss plot of the NPE trained for 256 steps and 1024 steps maximum. We can see that the loss is stopping around the 500^{th} step, this is due to the fact that the learning rate has decreased to less than 10^{-6}, stopping the training. The total training time for this longer training was around 4 hours. We also see that the variance on the validation loss is decreasing, this is due to the decreased learning rate.

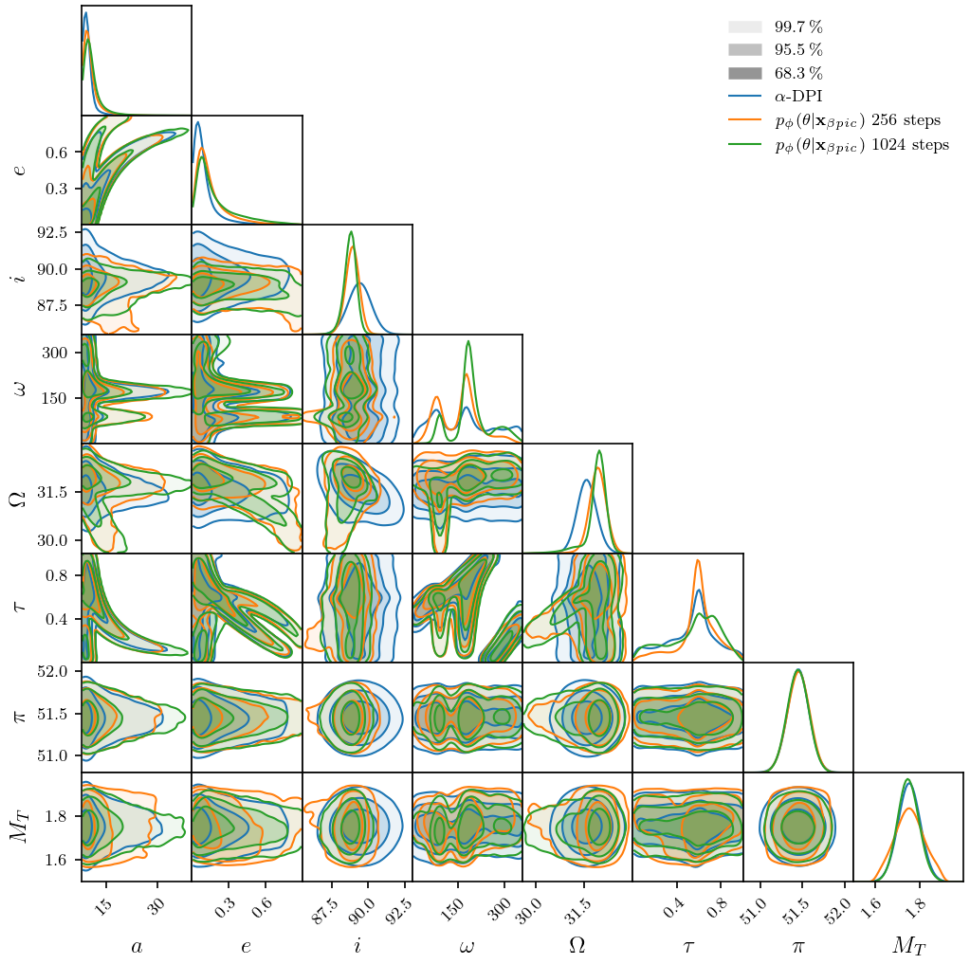

Figure C.2. Corner plot of the posterior distribution of the parameters of β-pic b using the NPE with a longer training. No significant difference can be seen between the corner plot of the NPE with a longer training and the NPE with a shorter training except for the total system mass, which is slightly more constrained and perfectly match the prediction of α-DPI.

C.2 Using NICE normalizing flows

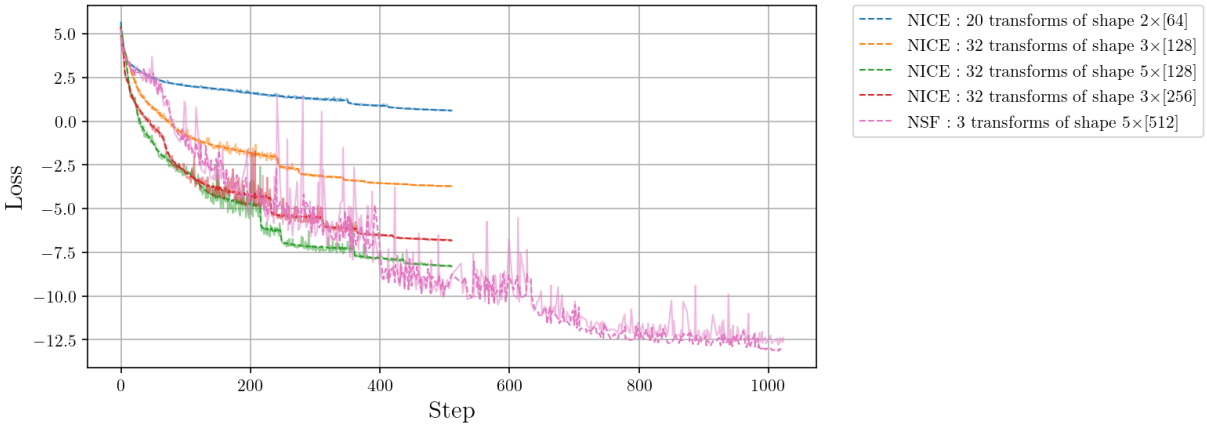

Figure C.3. This loss plot shows the performance of the NPE trained with NICE normalizing flows using different architecture sizes and a NSF normalizing flow. They were all trained on the same dataset for the same number of steps except for the NSF. The latter was trained for 1024 steps instead of 512 as it is faster to train due to having fewer transforms. It also took less time to train even with the additional steps, 1 hour compared to approximatively 1.5 hours for all the other architectures. We observe that the complexity of the architecture significantly impacts the loss for the NICE architecture. Since NICE uses affine transformations, it requires a large number of them to effectively model the data, which slows down the training process. In contrast, the NSF architecture outperforms NICE with almost ten times fewer transforms. This efficiency allows NSF transforms to be larger and more complex without increasing training time. The plot also indicates that all NICE losses are nearly converging, whereas the NSF loss is still decreasing, suggesting that additional training could further improve its performance. The dotted lines represent the training loss, and the solid lines represent the validation loss for each architecture.

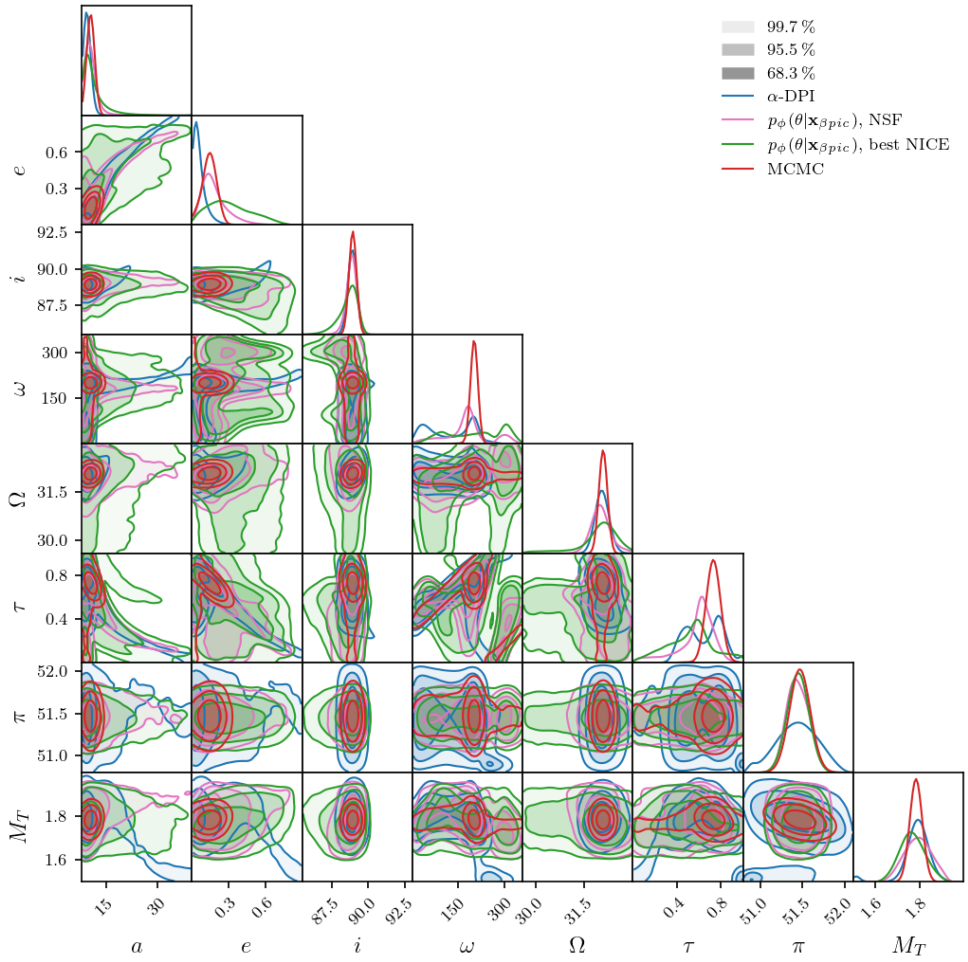

Figure C.4. Corner plot of the posterior distribution of the parameters of β-pic b, using α-DPI, NPE with NSF normalizing flow, MCMC and NPE with NICE normalizing flow, the one that achieved the lowest validation loss on Figure C.3. We can see the NICE architecture is not able to constrain the parameters as well as the NSF architecture.

C.3 β-pic b using the ResMLP model

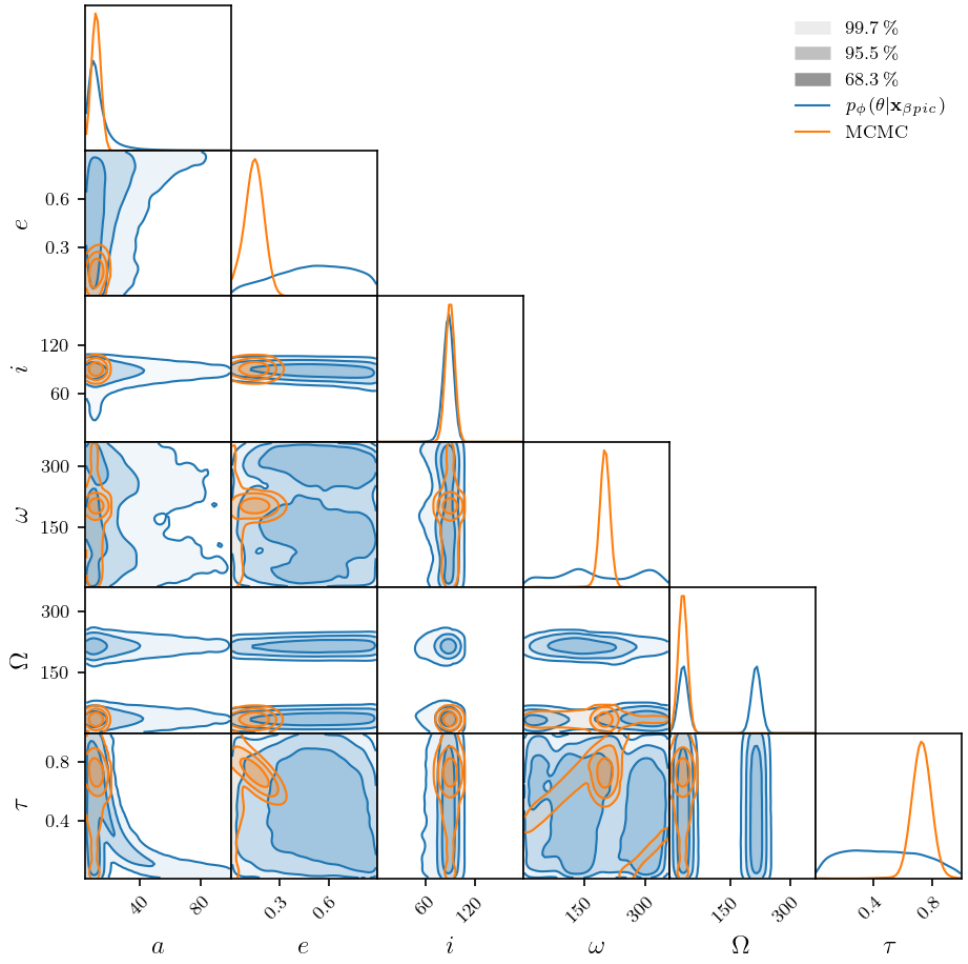

Figure C.5. Corner plot of the posterior distribution of the parameters of β-pic b using the ResMLP model. We can see that the model is not able to constrain the parameters as well as the model described in Chapter 4.

C.4 HR8799 bcde using MCMC

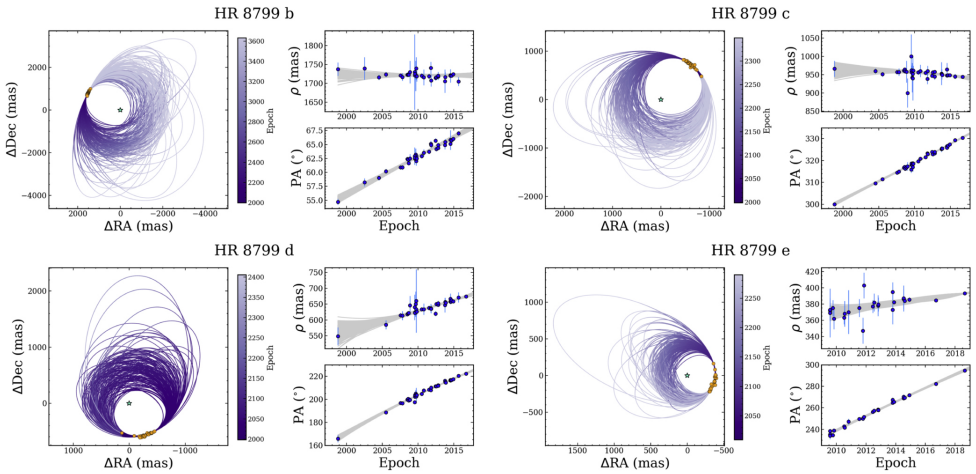

Figure C.6. Plot taken from the paper of Sepulveda et al. [40] showing the orbits of HR8799 bcde obtained using MCMC.

C.5 MCMC chains

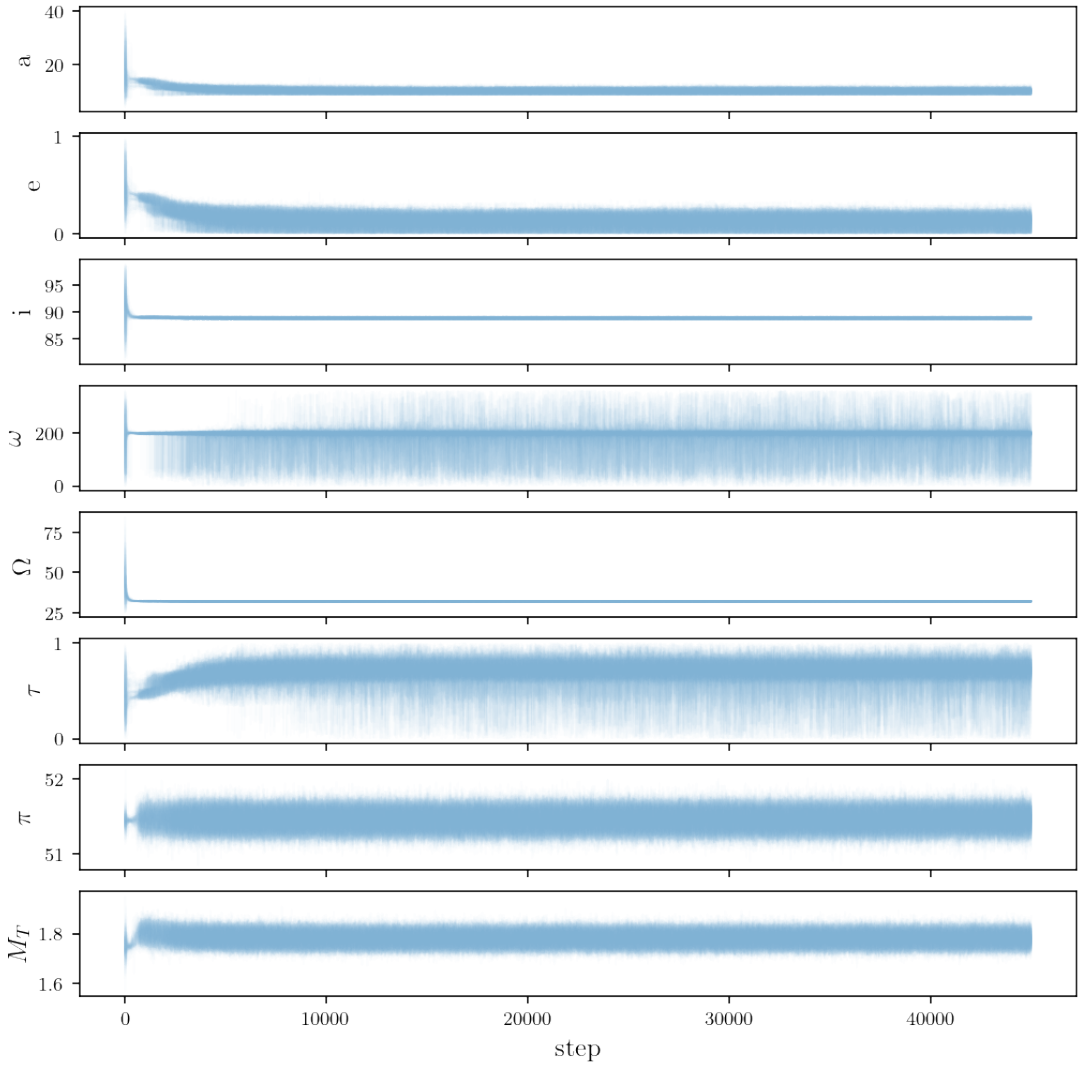

Figure C.7. MCMC chains for the parameters of β-pic b. We can see that the chains have converged.

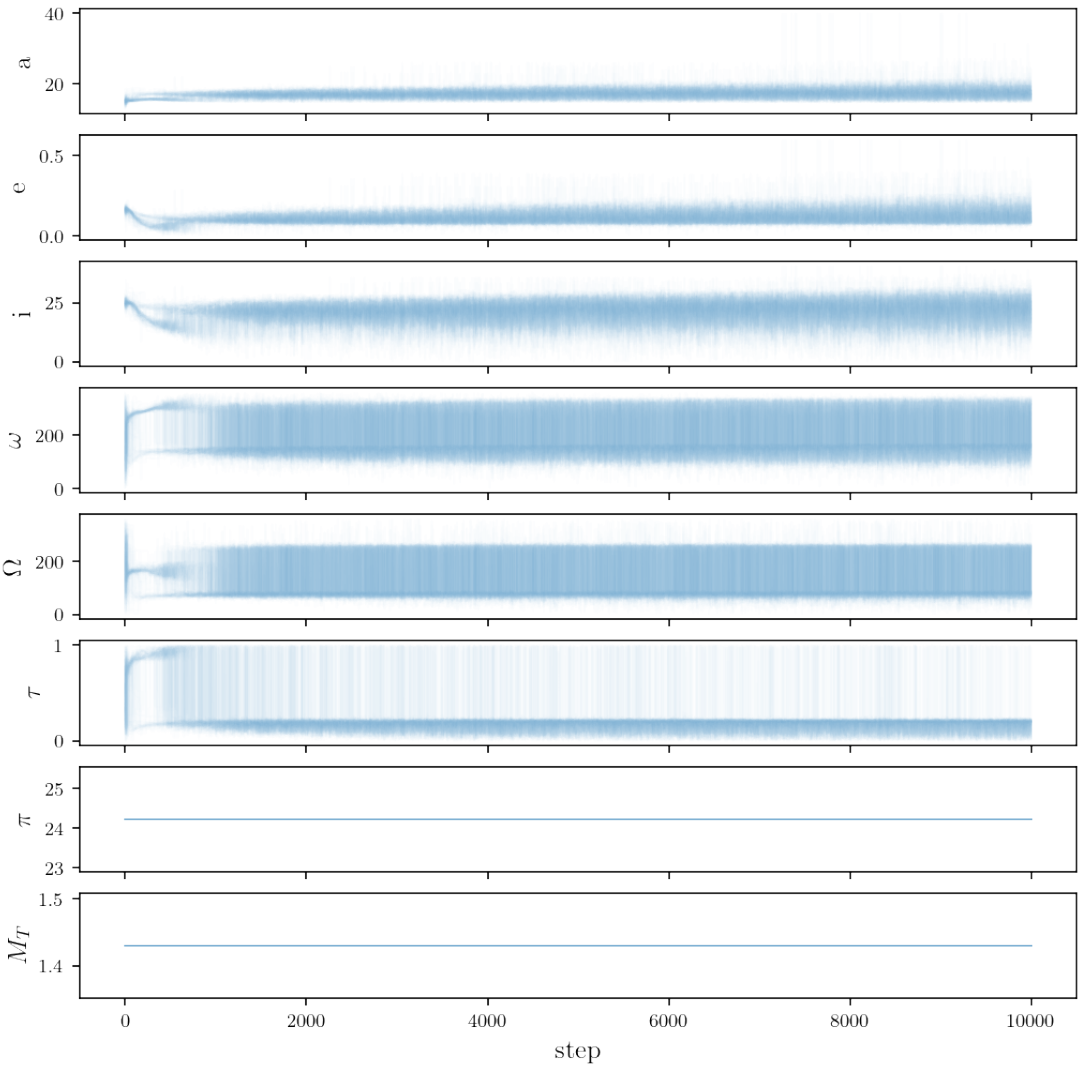

Figure C.8. MCMC chains for the parameters of HR8799 e. Here also the chains have converged. The convergence was quicker as I started the chains from prior close to the values computed by Sepulveda et al. [40]