

Using Bayesian Deep Learning to Infer Planet Mass from Gaps in Protoplanetary Disks

Sayantan Auddy¹ , Ramit Dey^{2,3} , Min-Kai Lin^{4,5} , Daniel Carrera¹ , and Jacob B. Simon¹ 

¹ Department of Physics and Astronomy, Iowa State University, Ames, IA 50010, USA; sayantanauddy21@gmail.com

² School of Mathematical and Computational Sciences, Indian Association for the Cultivation of Science, Kolkata-700032, India

³ Department of Physics and Astronomy, The University of Western Ontario, London, ON N6A 3K7, Canada

⁴ Institute of Astronomy and Astrophysics, Academia Sinica, Taipei 10617, Taiwan

⁵ Physics Division, National Center for Theoretical Sciences, Taipei 10617, Taiwan

Received 2022 February 22; revised 2022 May 31; accepted 2022 June 17; published 2022 September 5

Abstract

Planet-induced substructures, like annular gaps, observed in dust emission from protoplanetary disks, provide a unique probe for characterizing unseen young planets. While deep-learning-based models have an edge in characterizing a planet’s properties over traditional methods, such as customized simulations and empirical relations, they lack the ability to quantify the uncertainties associated with their predictions. In this paper, we introduce a Bayesian deep-learning network, “DPNNet-Bayesian,” which can predict planet mass from disk gaps and also provides the uncertainties associated with the prediction. A unique feature of our approach is that it is able to distinguish between the uncertainty associated with the deep-learning architecture and the uncertainty inherent in the input data due to measurement noise. The model is trained on a data set generated from disk–planet simulations using the FARGO3D hydrodynamics code, with a newly implemented fixed grain size module and improved initial conditions. The Bayesian framework enables the estimation of a gauge/confidence interval over the validity of the prediction, when applied to unknown observations. As a proof of concept, we apply DPNNet-Bayesian to the dust gaps observed in HL Tau. The network predicts masses of $86.0 \pm 5.5 M_{\oplus}$, $43.8 \pm 3.3 M_{\oplus}$, and $92.2 \pm 5.1 M_{\oplus}$, respectively, which are comparable to those from other studies based on specialized simulations.

Unified Astronomy Thesaurus concepts: Neural networks (1933); Exoplanet detection methods (489); Protoplanetary disks (1300); Bayesian statistics (1900)

1. Introduction

Observed substructures, like concentric rings and gaps, from near-infrared images of protoplanetary disks (PPDs) in dust emission (ALMA Partnership et al. 2015; Andrews et al. 2016; Huang et al. 2018a) are often interpreted as the signatures of an embedded planet. However, owing to limitations in the current planet search techniques (Fischer et al. 2014; Lee 2018), the direct detection of planets in disks is still a challenge.

Only recently have there been measurements using gas kinematics (Pinte et al. 2018; Teague et al. 2018; Pinte et al. 2019, 2020) that have found planetary signatures, such as “kinks” due to velocity perturbations, inside the observed dust gaps. Additionally, the direct detection using $H\alpha$ emission of an accreting planet (Keppler et al. 2018; Wagner et al. 2018; Haffert et al. 2019) inside the cavity of the transition disk PDS-70 has provided compelling evidence in favor of planetary gaps.

Thus, characterizing the large-scale planet-induced distortions observed in PPDs offers a unique opportunity to probe unseen young planets during the epoch of their formation. This is achieved by identifying and comparing the features with theoretical models of planetary gaps generated using customized simulations (Crida et al. 2006; Paardekooper & Papaloizou 2009; Duffell & Macfadyen 2013; Fung et al. 2014; Duffell 2015; Kanagawa et al. 2015a; Zhang et al. 2018; Ilee et al. 2020) or empirical relations (Kanagawa et al. 2016; Lodato et al. 2019). However, owing to the increasing sample

size of the observed disks and the complexity of disk–planet simulations, one needs to run hundreds of customized simulations to analyze each observed system. This is too expensive and inefficient to be of practical use as a detection tool. The alternative is to use deep-learning models, trained with simulation data (generated only once), to detect/characterize exoplanets from the observed data in a more efficient and accurate way.

In Auddy & Lin (2020) and Auddy et al. (2021), hereafter Papers I and II, respectively, we successfully designed deep-learning models using a multilayer perceptron (MLP) and/or a Convolutional Neural Network (CNN) to predict planet mass from the planetary gaps in PPDs. Both models (DPNNet-1.0 and DPNNet-2.0) are end-to-end pipelines that are capable of predicting the mass of an exoplanet from an observed PPD harboring a gap-inducing planet. Recently, Zhang et al. (2022) have also developed a CNN-based model to directly infer the planet mass from radio dust continuum images. However, one of the main challenges with traditional deep-learning models is quantifying the uncertainties associated with their predictions. This has direct implications for how a deep-learning model performs when trained on limited and/or noisy data. Furthermore, the inability of the traditional deep-learning model to answer “I do not know” or “I am not confident” is concerning, particularly when treating *out-of-training* distribution points (i.e., input data points that are outside the parameter space explored in the training data set). Thus, it is crucial to understand the limitations of a deep-learning model by quantifying the errors associated with its predictions and identifying the source of the uncertainty.

Recently, some of these shortcomings have been addressed by introducing probabilistic methods, such as Bayesian



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

techniques, to deep-learning architectures (Kendall & Gal 2017; Jospin et al. 2020; Wilson & Izmailov 2020). A Bayesian Neural Network (BNN) improves on traditional deep-learning models by providing uncertainty estimates associated with the predictions of these models. BNNs are truly unique in their ability to distinguish between the uncertainty that is caused by the limitations of the model and the uncertainty that is inherent to the input data. More so, these architectures can even respond to pathological out-of-training cases with a large error bar.

For a classical neural network, one would have deterministic point estimates from a single optimized network as the output of the network (e.g., see Paper I). A BNN considers an aggregation of predictions (also known as marginalization) from multiple independent predictor networks and performs a model average to obtain the output as a distribution. A BNN is a type of stochastic neural network, in which the weights are given as a distribution. It enables us to understand the uncertainty that is associated with the network architecture. This is particularly powerful in constraining the error when performing a regression task to estimate parameters (such as planet mass, in our case) using deep-learning models. Thus, any prediction comes with a confidence interval, meeting the requirement of a scientific experiment, and one that can be compared with other methods of parameter estimation.

In this paper, we develop and introduce “DPNNNet-Bayesian,” a standard MLP-based model designed using Bayesian formalism to predict planet mass from disk gaps as well as to quantify the uncertainty associated with the prediction. Our network is trained with synthetic data from explicit planet–disk simulations using the FARGO3D hydrodynamics code, with a newly implemented fixed grain size module and improved initial conditions.

The paper is organized as follows. In Section 2, we give an overview of Bayesian deep learning, its implementation, and its advantages over traditional deep-learning approaches. In Section 3, we describe the disk–planet hydrosimulations as well as the parameter space considered for the current study. We introduce the BBN architecture implemented in DPNNNet-Bayesian in Section 4, and discuss in detail the steps used to preprocess the data. In Section 5, we give the predictions of, along with the error estimates from, our trained DPNNNet-Bayesian, based on simulated data. In Section 6, as a proof of concept, we deploy DPNNNet-Bayesian to observed PPDs around HL Tau and AS 209 to infer the planet masses and the associated errors. Finally, in Section 7, we discuss the results and draw conclusions.

2. Bayesian Deep Learning: A General Overview

A neural network architecture consists of a few thousands (or even millions) of nodes/neurons that are densely interconnected. For this architecture, let

1. θ represent all the internal model parameters, and
2. D represent the training data, which consists of a specified set of observed features D_x (e.g., gap size) and their corresponding output labels D_y (e.g., planet mass).

Deep learning is a technique for optimizing the model parameters θ based on a training data set D (Goodfellow et al. 2016; Shrestha & Mahmood 2019). This is done by computing the minimum “cost” using some method of backpropagation.

Usually, the cost function is defined in terms of the log likelihood of the training data set, and thus the training process can be defined as a maximum-likelihood estimation. This approach might suffer from overconfidence, as it is deterministic in nature. An alternative to this is to implement the use of stochastic components in the neural network. This can account for the uncertainties in the determination of the network parameters θ , which finally shows up as an estimated error in the predictions made by a deep-learning architecture (Jospin et al. 2020). For these types of neural nets, the stochastic elements can be introduced in terms of stochastic model weights or activation. Thus, instead of a single deterministic output from the traditional deep-learning model, for the stochastic networks a marginalization of the output is considered, based on some statistical principle (Wilson & Izmailov 2020).

In order to design a predictive model based on BNN, one starts with a suitable choice of deep-learning architecture. The next step is to choose a systematic prescription, in order to introduce the stochastic elements into the network architecture. This can be modeled in terms of some possible prior distribution of the model parameters, $p(\theta)$.

From Bayes’ theorem, we get

$$p(\theta|D) \propto p(D_y|D_x, \theta)p(\theta), \quad (1)$$

where D_y denotes the training labels and D_x are the training features, as mentioned earlier. The Bayesian posterior distribution, $p(\theta|D)$, depicts the model parameters as a probabilistic distribution, rather than some deterministic value.

The Bayesian approach allows us to compute the model averages by essentially integrating over the models, parameterized by the model parameters θ . The distribution of the possible values of y (e.g., planet mass) given some input x (e.g., dust gap width) is given by

$$p(y|x, D) = \int_{\theta} p(y|x, \theta')p(\theta'|D)d\theta'. \quad (2)$$

In Equation (2), the variance of $p(y|x, \theta')$ captures the uncertainty due to the measurement noise of the input data, while the variance of $p(\theta'|D)$ corresponds to the uncertainty associated with the neural network architecture.

The classical training would correspond to $p(\theta|D)$ being replaced by a constant value of θ , but in the Bayesian case this would typically be a probability distribution with some finite variance (Wilson & Izmailov 2020; Abdar et al. 2021). Also, computing $p(\theta|D)$ from the training data is one of the most challenging and difficult aspects of the BNN architectures, due to the large size of the parameter space. This is further discussed in the following sections.

2.1. The Variational Inference Approach to Bayesian Deep Learning

Due to the large dimensionality of the sampling space, it can be quite challenging to merge the concepts of a neural network with the Bayesian approach. As the posterior distribution is extremely complex, directly estimating $p(\theta|D)$ to perform the integral in Equation (2) is not always possible. Different methods, involving various approximations, have been implemented to do this. One of the most popular and direct methods for sampling the posterior distribution, $p(\theta|D)$, is using a Markov Chain Monte Carlo (MCMC) method (Bardenet et al. 2017). However, MCMC often has issues with scalability,

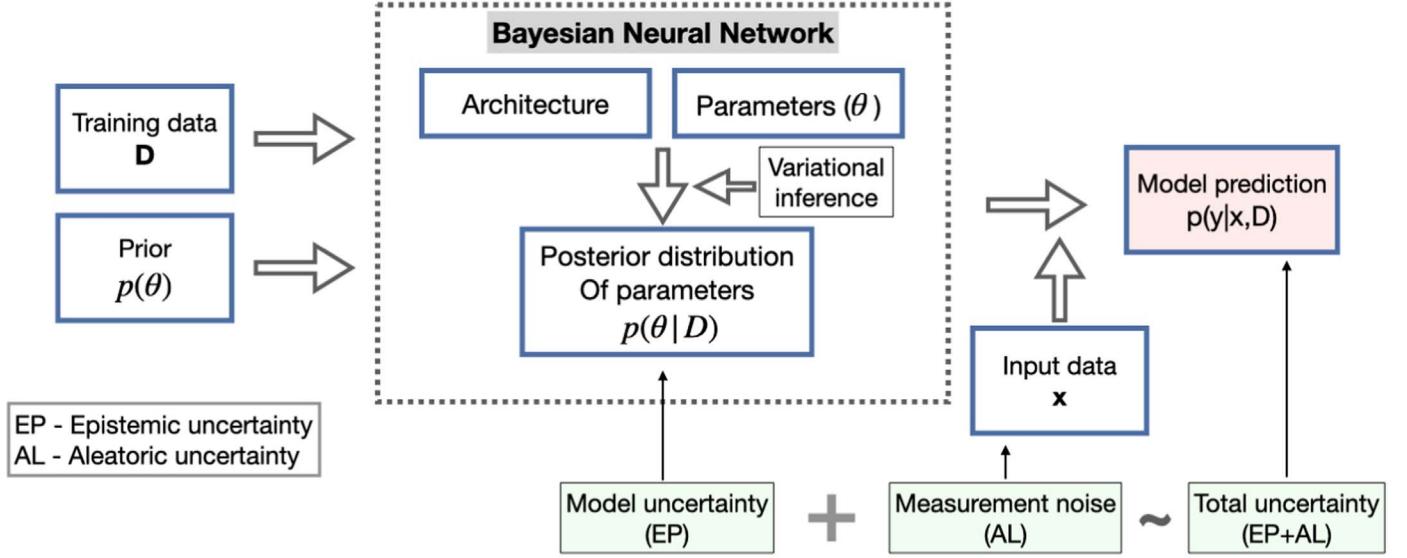


Figure 1. This schematic diagram shows the workflow of a typical BNN, along with how the uncertainty/errors propagate.

owing to the large model sizes, and it can be computationally expensive as well.

Instead of sampling from the posterior directly, the method of variational inference (Blei et al. 2017; Mohan et al. 2022) can be used as an alternative. A distribution (referred to as the variational distribution), parameterized by an arbitrary set of parameters ϕ , is used to approximate the posterior. The parameter ϕ is learned in a way that the distribution $q_\phi(\theta)$ becomes close to the exact posterior $p(\theta|D)$. The closeness between the posterior and $q_\phi(\theta)$ is quantified in terms of the Kullback–Leibler (KL) divergence (Zhang et al. 2019; Jospin et al. 2020). More precisely, the formalism of the *evidence lower bound* (ELBO) is used for this purpose:

$$\text{ELBO} = \int_{\theta} q_\phi(\theta') \log \left(\frac{p(\theta', D)}{q_\phi(\theta')} \right) d\theta'. \quad (3)$$

Minimizing the ELBO is equivalent to minimizing the KL divergence, as $\text{ELBO} = \log(p(D)) - D_{\text{KL}}(q_\phi||p)$, where $\log(p(D))$ depends on the prior and D_{KL} is the KL divergence. Such a variational approach has been used in a variety of other astronomical applications, such as in Roberts et al. (2013), Perreault Levasseur et al. (2017), Leung & Bovy (2019), and Spindler et al. (2021).

2.2. Features and Advantages of BNN

Once the ELBO function is optimized, and the variational distribution is thus learned/trained, we can use it to predict the labels of new (unseen) observations. The predictive posterior distribution $p(y^*|x^*, D)$ is given by integrating out the model/variational parameters (trained using the data set D):

$$p(y^*|x^*, D) = \int_{\theta} p(y^*|x^*, \theta') p(\theta'|D) d\theta', \quad (4)$$

where (x^*, y^*) represent the features and labels, respectively, and the superscript “*” denotes unseen data. In order to obtain this predictive distribution, we use Monte Carlo sampling. First, we sample from the distribution of model parameters

N times:

$$\theta^j \sim p(\theta|D) \quad j \in \{1, 2, \dots, N\}. \quad (5)$$

Then x_i is sampled n times from the distribution of the input features, whose width is determined by the errors associated with the feature itself, to estimate the predictive distribution $p(y_i^*|x_i^*, \theta^j)$. For $N \times n$ iterations, we get

$$p(y^*|x^*, D) \approx \sum_{i=1}^n \sum_{j=1}^N p(y_i^*|x_i^*, \theta^j) p(\theta^j|D). \quad (6)$$

The error/uncertainty in the predictions of the BNN can be broadly classified into two categories: epistemic and aleatoric uncertainty (Abdar et al. 2021; Hüllermeier & Waegeman 2021). “Aleatoric” uncertainty (AL) represents the uncertainty inherent in the data that is given as the input for the model (e.g., the measurement noise). In contrast, “Epistemic” uncertainty (EP) represents the limitations of the machine-learning (ML)–based model that usually stem from a lack of information about the system. Taking a larger data set for the training of the model (spanning a larger parameter space) or improving the ML architecture can typically reduce the EP. On the other hand, as the AL is intrinsic to the measurement noise of the input data, it cannot be reduced by solely improving the ML model. Both these uncertainties propagate to the output to give the net predictive uncertainty (Abdar et al. 2021), as shown in Figure 1.

We can measure the EP, , from the variance of the posterior distribution of the model parameters, $p(\theta^j|D)$, given as:

$$\sigma_{\text{EP}}^2 \sim \text{Variance of } p(\theta^j|D). \quad (7)$$

σ_{EP} quantifies the intrinsic uncertainty associated with the model. It enables us to understand the correlation between the model’s performance, the quantity/quality of the data, and the parameters of the ML architecture. Thus, monitoring σ_{EP} can give insights into improving the overall ML model.

The AL, is obtained from the variance of $p(y_i^*|x_i^*, \theta^j)$, given as:

$$\sigma_{\text{AL}}^2 \sim \text{Variance of } p(y|x, \theta). \quad (8)$$

σ_{AL} quantifies the statistical uncertainty/error associated with the various measurements that go as input features into the neural net at the training stage, as well as when the model is deployed for real-world application. Knowing σ_{AL} , one can monitor the errors in the predictions of the model as the measurement noise of the input features changes. The advantage of knowing both σ_{EP} and σ_{AL} is that when testing the model, one can trace the source of the uncertainty and improve the model/data set accordingly.

3. Disk–Planet Simulations

We model the structural variations of a dusty PPD due to an embedded gap-opening planet using the FARGO3D hydrodynamic simulation code (Benítez-Llambay & Masset 2016). We update the public version of the code by parameterizing the dust–gas interaction with a constant particle size, instead of a fixed Stokes number. This includes a generalization of the steady-state profiles derived by Benítez-Llambay et al. (2019), which are presented in Appendix. We generate the data set with the updated simulations to train the BBN model. The simulation setup and the parameter space are discussed in the next section.

3.1. PPD Setup

We model a simple 2D razor-thin disk with an embedded planet of mass M_p located at $R = R_0$. The planet is on a Keplerian orbit around a central star of mass M_* and it is not migrating. The unit of time is the planet’s orbital period $P_0 = 2\pi/\Omega_{\text{K0}}$, where Ω_{K0} is the Keplerian frequency. The subscript “0” denotes evaluation at the planet’s location R_0 .

The disk’s initial gas surface density profile is given as

$$\Sigma_g(R) = \Sigma_{g0} \left(\frac{R}{R_0} \right)^{-\sigma_0}, \quad (9)$$

where σ_0 is the exponent of the surface density profile and $\Sigma_{g0} = 10^{-4}$ is a constant in the code unit, which is arbitrary for non-self-gravitating simulations without orbital migration. For reference, however, this surface density scale corresponds to Toomre parameters $Q \geq 80$ for all of our disk models, which are sufficient for the disk to be gravitationally stable (Toomre 1964). The disk is flared with a constant flaring index F , such that the aspect ratio varies as

$$h = \frac{H}{R} = h_0 \left(\frac{R}{R_0} \right)^F, \quad (10)$$

where H is the disk scale height and h_0 is the aspect ratio at $R = R_0$. The disk is locally isothermal, with a sound-speed profile of $c_s(R) = h_0 R \Omega_{\text{K}}$. To mimic the disk’s turbulence, we include viscous forces in the gas disk using the standard α -prescription (Shakura & Sunyaev 1973), where the kinematic viscosity $\nu = \alpha c_s^2 / \Omega_{\text{K}}$ and α is set by Equation (A21).

We consider a single population of dust particles modeled as a pressureless fluid (Jacquet et al. 2011). The dust particle size is parameterized using the Stokes number $S \equiv t_s \Omega_{\text{K}}$, where t_s is the stopping time characterizing the strength of the dust–gas coupling (Weidenschilling 1977). The initial dust surface density is $\Sigma_d = \epsilon \Sigma_g$, where ϵ is the dust-to-gas ratio set by Equation (A19). The dust’s backreaction onto the gas is included in the simulations.

3.2. Simulation Setup and Parameter Space

We run FARGO3D planet–disk simulations on graphical procession units. Our computational domains are $R \in [0.4, 3.0]R_0$ and $\phi \in [0, 2\pi]$, which are discretized with 512×512 logarithmically spaced cells. The radial and azimuthal boundaries are set to their initial equilibrium solutions, given in Equations (A10)–(A13). The planet mass is allowed to grow slowly over first $100P_0$. We implement the wave-killing (de Val-Borro 2006) module to damp the planet-generated waves to zero near the boundaries, to minimize the reflection and interference with disk morphology.

We target planets from super-Earth to Saturn size with masses $M_p \in [8, 120] M_{\oplus}$ around a $1 M_{\odot}$ central star. We model the disk for a range of values of the disk aspect ratio $h_0 \in [0.025, 0.1]$, viscosity parameters $\alpha_0 \in [10^{-4}, 10^{-2}]$, flaring index $F \in [0, 0.25]$, and power-law slope of the surface density profile $\sigma_0 \in [0.05, 1.2]$. In each disk, we consider a single dust species of fixed size characterized by the Stokes number $S_t \in [10^{-3}, 10^{-1}]$ and abundance $\epsilon_0 \in [0.01, 0.1]$. This range of S_t corresponds to particle radii ranging from ~ 1 mm to ~ 10 cm, for a typical gas surface density of 100 g cm^{-2} and an internal grain density of 1 g cm^{-3} . More details of the sampling of the parameter space in order to generate the data set are given in Section 4.2.

4. DPNNet-Bayesian

In this section, we introduce the Bayesian implementation of the DPNNet model.

4.1. Network Architecture

The DPNNet model, as presented in Paper I, was based on artificial neural networks, with a linear layer at the end for performing regression. In the Bayesian adaption of DPNNet, the front end of the model includes an MLP layer with 256 units of neurons having a ReLu activation function (this part of the network is the same as DPNNet-1.0). This is followed by a fully connected variational dense layer having 128 neurons, in which the BNN algorithms are implemented to probe the model uncertainty. We use the method of variational inference, as described in Section 2.1, to account for the stochasticity of the model parameters. Each model parameter is treated as a distribution having some variance. The stochastic sampling from this variational distribution gives an estimate of the model uncertainty. Furthermore, we use a trainable prior distribution for the parameter space, $p(\theta)$, where θ is drawn from a normal distribution. This is used to determine the approximate posterior distribution of the model weights using variational inference.

4.2. Data Acquisition and Preprocessing

For the training and testing of the model, we produce the new data set following the same procedure as used in Paper I. We generate a uniform random distribution of the parameters, each centered within the sampling interval, using the Latin hypercube sampling method (McKay et al. 1979; Iman et al. 1981). We run a total of 1800 simulations with a range of input parameters, as described in Section 3.2. From each simulation, we obtain the dust and gas surface density maps of the PPD with the embedded planet. Figure 2 shows sample images of the dust surface density distributions and the corresponding

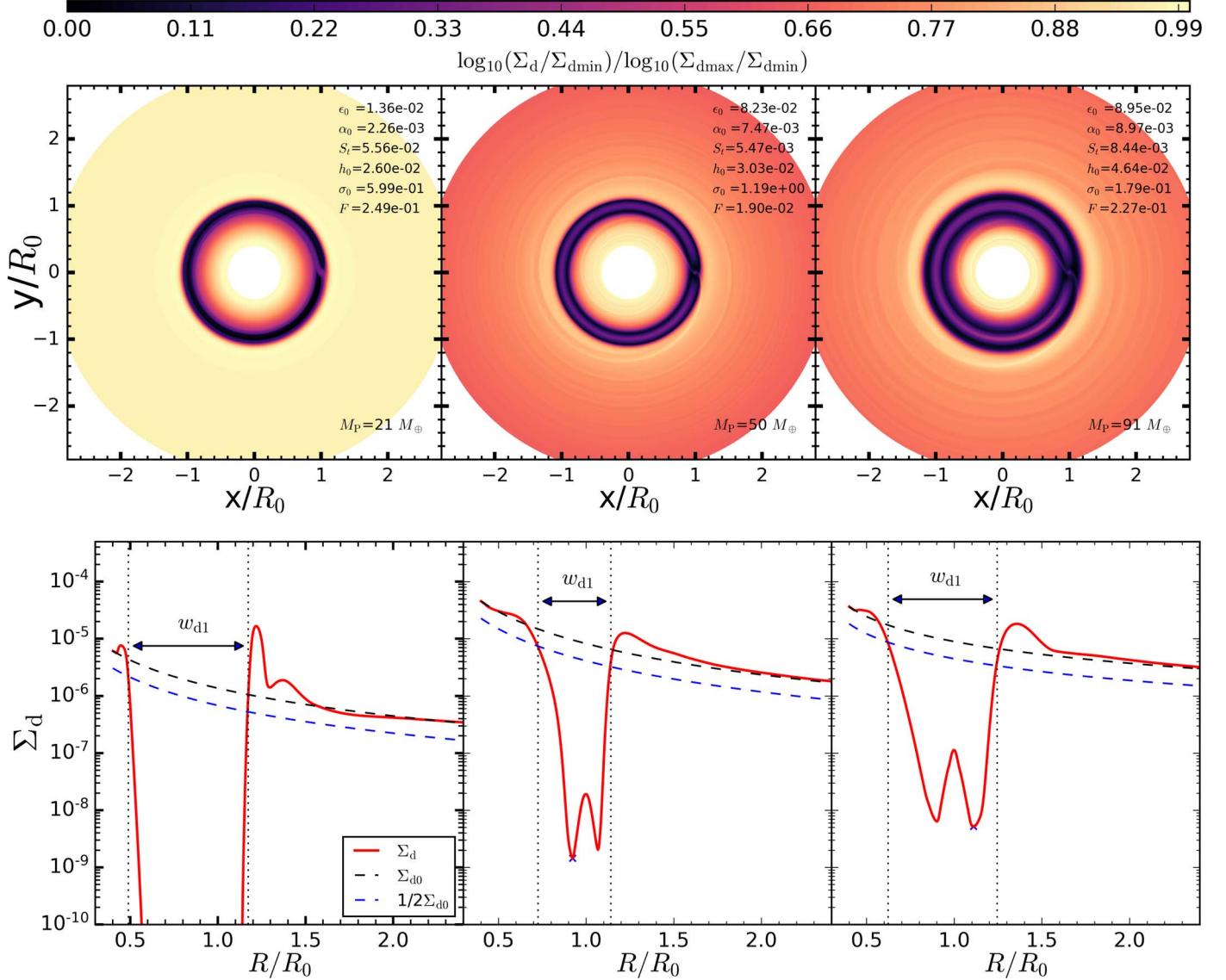


Figure 2. The upper panel shows the normalized dust surface density distributions of disks with different initial conditions and for different planet masses at $3000P_0$. The disk parameters are indicated at the top right of each plot, together with the planet mass at the bottom right. The lower panel shows the radial profiles of the azimuthally averaged surface densities at the beginning of the simulation (Σ_{d0} ; the dashed black lines) and at the end of the simulation after the gap is formed (Σ_d ; the solid red lines). The horizontal arrow marks the dust gap width. The gap width is the distance between the inner and outer edge of the gap (the vertical dotted lines) where $\Sigma_d(R)$ reaches 50% of the initial surface density Σ_{d0} . The crosses indicate the minimum surface density $\Sigma_{d\min}$.

radial profiles of the azimuthally averaged surface densities from some of the simulations. These improved disk–planet simulations (with fixed grain sizes and initial equilibrium disk setups) result in a much more physical model for dust–gas interaction compared to Paper I and Paper II.

Following Paper I and Kanagawa et al. (2016), we define the dust gap width as

$$w = \frac{R_{d,\text{out}} - R_{d,\text{in}}}{R_0}, \quad (11)$$

where $R_{d,\text{in}}$ and $R_{d,\text{out}}$ are the inner and outer edge of the gap where the dust surface density Σ_d equals a predefined threshold fraction of Σ_{d0} . For consistency with Paper I, we adopt a threshold fraction of half the initial surface. We acknowledge that this definition of w_d has its limitations, as it fails for shallow gaps (for low-mass planets) when the gap depth is less than the threshold function. However, this does not limit the

model’s application, as DPNNet-Bayesian can be retrained for any threshold function or other definition of the gap width.

In the current paper, we only consider simulations with either one dust gap or two dust gaps. We remove models with >2 dust gaps, as the number of samples is not adequate to effectively train the model. To build our feature space, we simply define two dust gap widths for each simulation, w_{d1} and w_{d2} , but set $w_{d2} = 0$ if only a single dust gap is formed.

The initial data set consists of 1800 simulations. We preprocess the data by eliminating runs that do not open up detectable axisymmetric gaps or that have more than two gaps. After the initial filtering, we have data from 1098 simulations. Each simulation is characterized by its input parameters (M_p , α_0 , h_0 , σ_0 , F , S_t , and ϵ_0) and the associated gap width(s) in dust (w_{d1} , w_{d2}) and gas (w_g). In this paper, we will not use the gas gap width as an input parameter for training the model, since this is not directly measured from the observations.

Once the gap widths have been measured, we assign each simulation a set of feature variables (w_{d1} , w_{d2} , ϵ_0 , α_0 , S_t , h_0 , σ_0 , and F) that will go as the inputs for our model, and we label the planet mass, M_p , as the target variable. The data for each feature variable are then normalized using the standard (Z-score) scaling, by subtracting the mean and scaling it by the standard deviation.

4.3. Training DPNNet-Bayesian

DPNNet-Bayesian is implemented using Google TensorFlow Probability (Dillon et al. 2017), which is an open source platform that integrates probabilistic methods with deep neural networks. We randomly split the data into two blocks, for a training set consisting of 80% of the simulation data and a testing set consisting of 20% of the simulation data. Furthermore, for model validation, we use 20% of the training data set. We use the Adam optimizer with a learning rate of 0.001. The model is trained for ~ 2200 epochs, and the early-stopping callback algorithm is used to prevent the model from overfitting.

5. Results

Once the model has been trained, it is ready to be deployed to predict the planet mass from observed disk features. We test the network’s performance on the test data set that the model has not been exposed to. This enables us to quantify the network’s accuracy in predicting the planet mass from unseen data. The test data set comprises all the feature variables along with the target variable (the true value of the planet mass). The feature variables (w_{d1} , w_{d2} , ϵ_0 , α_0 , S_t , h_0 , σ_0 , and F) are normalized with the same scaling that the network was trained on. The trained DPNNet-Bayesian network takes the feature variables as inputs and predicts the corresponding planet mass. Unlike the standard MLP model, for the same set of input feature variables, DPNNet-Bayesian gives the predicted output as a distribution.

Figure 3 shows the correlation between the simulated and predicted planet mass in Earth-mass (M_\oplus) units for all the test samples. For each set of input variables, we conduct the experiment (i.e., deploy the trained network) 200 times, to obtain the marginalization of the predicted output. Thus, we obtain a distribution for the predicted planet mass (as demonstrated in Figure 4) capturing the uncertainties associated with the parameters of the trained model. The error bars give the standard deviation associated with each predicted value. The predicted values lie along the black line, indicating a strong correlation with the true planet mass. This is reflected by the r^2 score of 0.82 computed over the average prediction.

5.1. Error Analysis Using BNN

As mentioned before, the Bayesian approach offers a way of quantifying and understanding the uncertainties associated with the predictions from deep neural networks. As an example, in Figure 4, we demonstrate the different uncertainty estimates associated with predictions from DPNNet-Bayesian for sample disk features selected randomly from the test data set. We pick a disk–planet simulation with parameters/features $\epsilon_0 = 0.04$, $\alpha_0 = 0.008$, $S_t = 0.08$, $h_0 = 0.03$, $\sigma_0 = 1.11$, and $F = 0.21$ and measured dust gap widths $w_{d1} = 0.74$, $w_{d2} = 0$. Since these are simulated data, the true value of the planet mass, $M_p = 46 M_\oplus$, is known. To capture the EP, we deploy the trained model N

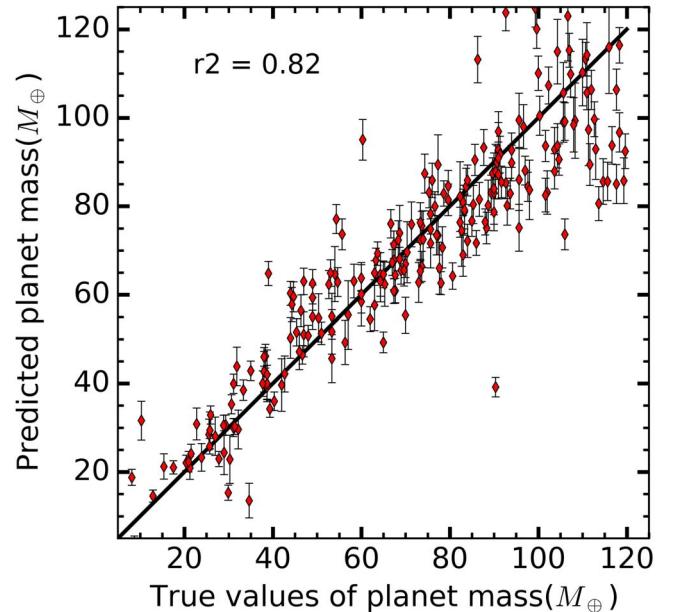


Figure 3. Correlation between the simulated and predicted mass in units of M_\oplus . The vertical error bars give the standard deviation associated with each predicted value, obtained using the BNN. The r^2 score indicates the goodness of the fit.

times for this fixed set of disk features. We set $N = 200$, with the i index fixed, in Equation (6), and for each iteration the model predicts a unique and independent value. This results in the distribution of the predicted planet mass displayed in the left panel of Figure 4. The mean ($\mu_{EP} = 47.6 M_\oplus$), shown with the red dotted line, is taken as the final predicted value, while the standard deviation ($\sigma_{EP} = 5.8 M_\oplus$) represents the uncertainty associated with the trained model.

The measured disk features are also prone to statistical errors, due to the measurement techniques and the limitations in the observed data. While the parameters in simulated models are well constrained, that is not the case for observed disk data. In order to simulate the uncertainties associated with the measured disk parameters, we use a Monte Carlo analysis. Instead of using fixed true values for the disk parameters, we randomly sample them n times from their respective standard normal distributions, with the mean set to the true value and the standard deviation equal to a 1% spread for each of the parameters. We set $n = 1000$, with j fixed, in Equation (6), such that for each set of sampled input parameters the model predicts a planet mass. The center panel of Figure 4 demonstrates $\sigma_{AL} = 8.3 M_\oplus$ corresponding to the predicted planet mass due to the measurement noise of the input disk parameters.

The rightmost panel in Figure 4 captures the total uncertainty in the predicted planet mass due to both EP and AL. We randomly sample the disk parameters, $n = 1000$ times, from their standard normal distribution, and for each draw we deploy the model $N = 200$ times. This results in a distribution of the predicted planet mass with mean $\mu = 49.8 M_\oplus$ and standard deviation $\sigma = 8.1 M_\oplus$.

6. Application to Observations

As a first test of our newly improved DPNNet-Bayesian model, we deploy it to estimate the planet masses from the observed dust gaps in the PPDs around HL Tau and AS 209. The results are summarized in Table 1 and discussed below.

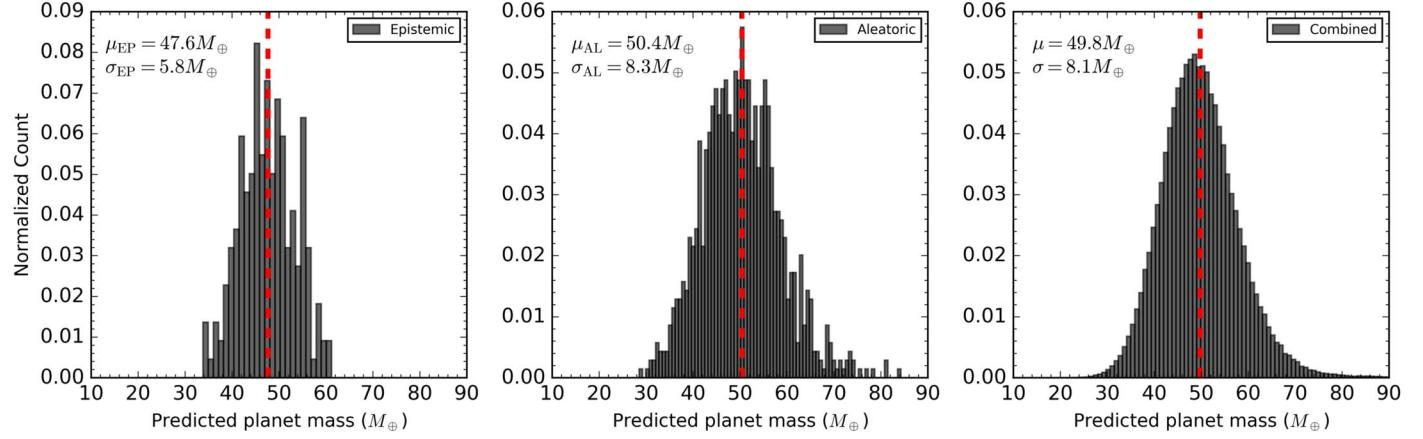


Figure 4. The distributions of the predicted planet masses for sample disk features selected randomly from the test data set. Left panel: the distribution captures the EP associated with DPNNNet due to lack of data and/or variations of the model parameters. Middle panel: the distribution represents the AL that arises due to errors/ variations inherent in the data. Right panel: the distribution captures the combined uncertainty in the predicted planet mass due to both EP and AL. The red dashed line represents the mean planet mass in each of the plots.

Table 1
Inferred Planet Masses from the Gaps in HL Tau and AS 209

Name	Observed Features							DPNNNet-Bayesian		Other Models					
	M_* (M_\odot)	R_{gap} (au)	w_{d1}	h_0	α_0	S_t	ϵ_0	σ_0	M_{p0} (M_J)	M_{p1} (M_J)	M_{p2} (M_J)	M_{p3} (M_J)	M_{p4} (M_J)	M_{p5} (M_J)	M_{p6} (M_J)
HL Tau	1.00	10	0.81	0.05	10^{-3}	0.005	0.01	1.0	0.27 ± 0.02	1.40	0.20	0.35	0.20
	1.00	30	0.23	0.07	10^{-3}	0.005	0.01	1.0	0.14 ± 0.01	0.20	0.20	0.17	0.27
	1.00	80	0.29	0.10	10^{-3}	0.005	0.01	1.0	0.30 ± 0.02	0.50	0.20	0.26	0.55
AS 209	0.83	9	0.42	0.04	10^{-4}	0.016	0.01	1.0	0.06 ± 0.01	0.37	...
	0.83	99	0.31	0.08	10^{-4}	0.016	0.02	1.0	0.12 ± 0.01	0.18	0.21

Note. M_{p0} is the mass (in the Jupiter-mass unit M_J) predicted by the DPNNNet-Bayesian model using the input features w_{d1} , h_0 , α_0 , S_t , ϵ_0 , and σ_0 , as given in columns 4–9, and setting $w_{d2} = 0$. M_{p1} is the mass predicted by Kanagawa et al. (2016), implementing the empirical relation. M_{p2} , M_{p3} , and M_{p4} are the masses inferred using customized simulations of HL Tau by Dong et al. (2015), Jin et al. (2016), and Dipierro et al. (2015), respectively. M_{p5} is the predicted planet mass of AS 209 by Zhang et al. (2018). M_{p6} is the predicted planet mass of AS 209 by Fedele et al. (2018), using specialized simulations.

6.1. HL Tau

HL Tau is a well-studied system, which shows distinct axisymmetric—possibly planet-induced—gaps in the dust emission (ALMA Partnership et al. 2015). Several studies involving hydrodynamic simulations (Dipierro et al. 2015; Dong et al. 2015; Jin et al. 2016) have suggested that each of the gaps are induced by a single planet. With the assumption that the observed gaps are all planet-induced, we use the properties of the observed gaps as input features for our BNN model to predict the planet mass. DPNNNet-Bayesian predicts a distribution for the planet masses for each of the gaps. For the three identified gaps at $R = 10$ au, 30 au, and 80 au, we obtain the gap widths ($w_{d1} = 0.81$, 0.23, and 0.29) and adopt the disk aspect ratios of $h_0 = 0.05$, 0.07, and 0.10, respectively, from Kanagawa et al. (2015b, 2016). HL Tau is mostly laminar, with viscosity $\alpha_0 = 10^{-3}$ (Dipierro et al. 2015) and a central star mass of $M_* = 1 M_\odot$. We set a canonical value of the dust-to-gas ratio $\epsilon_0 = 10^{-2}$ (ALMA Partnership et al. 2015; Dong et al. 2015) and adopt a mean Stokes number $S_t = 0.005$ (Dipierro et al. 2015) for all three gaps. Since the uncertainty associated with the input parameters is not well constrained across the literature, we take a conservative approximation and consider a 1%–5% variation in the gap

widths only. The top panel of Figure 5 shows the distribution of the predicted masses in the three identified gaps for the given disk parameters and gap widths. The mean values along with the standard deviations of the predicted planet masses at 10 au, 30 au, and 80 au are $86.0 \pm 5.5 M_\oplus$, $43.8 \pm 3.3 M_\oplus$, and $92.2 \pm 5.1 M_\oplus$, respectively. The standard deviation captures the total uncertainty, which includes the uncertainty associated with the BNN architecture (σ_{EP} is estimated as $4.0 M_\oplus$, $3.3 M_\oplus$, and $5.0 M_\oplus$, respectively, for the three gaps) along with the AL. For the assumed variation of 1%–5% in the gap widths, σ_{AL} is comparable to σ_{EP} . However, σ_{AL} can vary, depending on the noise in the input data. These estimates are mostly consistent with the inferred planet masses from direct numerical simulations like those of Dong et al. (2015), Jin et al. (2016), Dipierro et al. (2015), and DPNNNet-1.0 in Paper I.

6.2. AS 209

AS 209 is another well-studied system, hosting multiple gaps at 9, 24, 35, 61, 90, 105, and 137 au (Huang et al. 2018b). However, for simplicity, we only consider the two most prominent gaps, at $R = 9$ and 99 au. We adopt the disk parameters and dust gap widths from Zhang et al. (2018) and

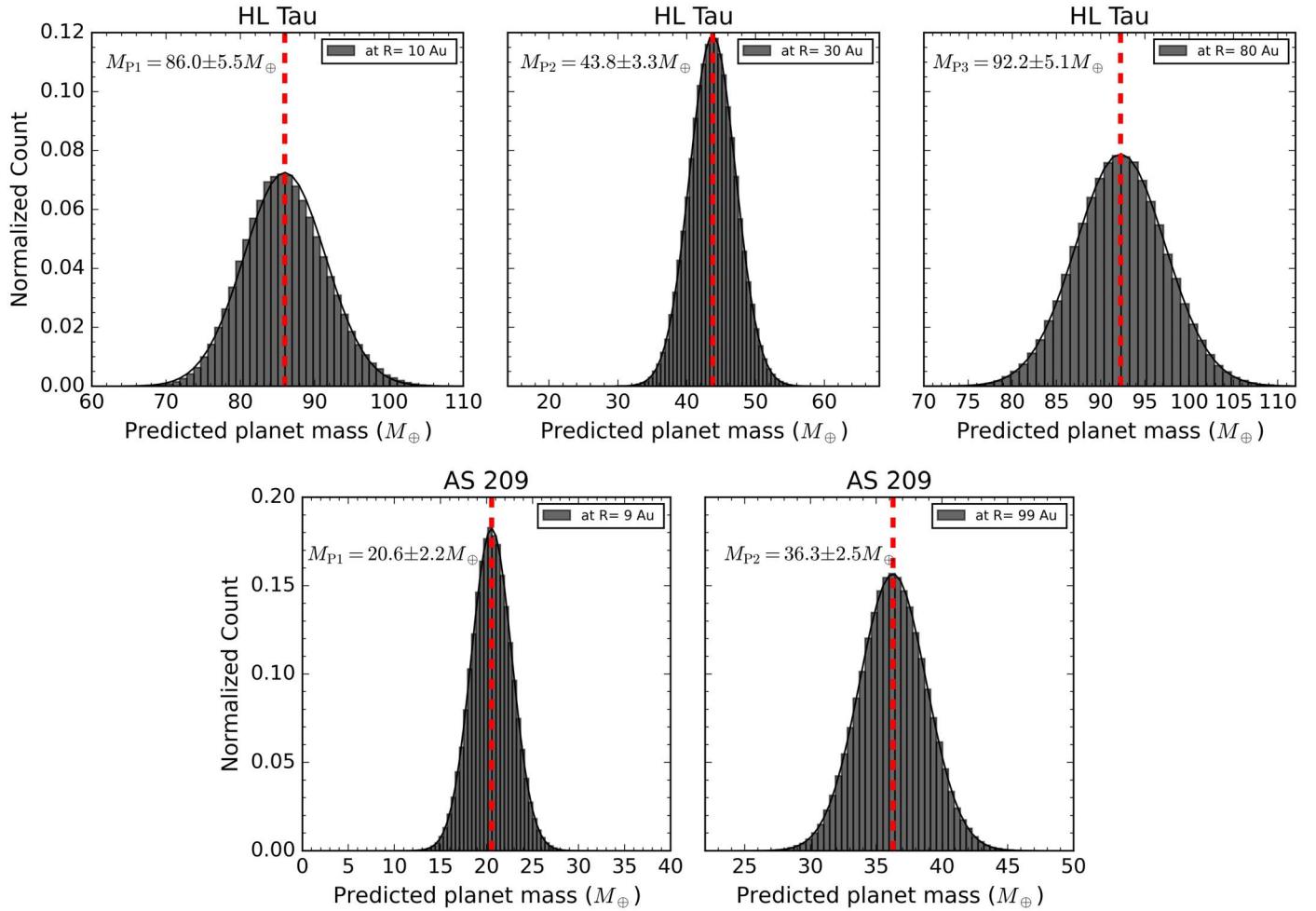


Figure 5. Top panels: the distributions of the predicted planet masses at the three identified gaps in HL Tau, obtained using DPNNNet-Bayesian. The mean values along with the standard deviations of the predicted planet masses at 10 au, 30 au, and 80 au are $86.0 \pm 5.5 M_{\oplus}$, $43.8 \pm 3.3 M_{\oplus}$, and $92.2 \pm 5.1 M_{\oplus}$, respectively. Bottom panel: the distributions of the predicted masses for the two identified gaps in AS 209 for the given disk parameters and gap widths. The mean predicted planet masses along with the standard deviations are $20.6 \pm 2.2 M_{\oplus}$ and $36.3 \pm 2.5 M_{\oplus}$ at 9 au and 99 au, respectively.

use a range of values for dust abundance, particle size, and disk viscosity that are within our training parameter space. For example, we set $\alpha_0 = 10^{-4}$, $S_t = 156.9 \times 10^{-4}$, and the surface density profile $\sigma_0 = 1$ (Fedele et al. 2018) for both the gaps. Next, for the innermost gap at $R = 9$ au, we assign values to the other parameters: $w_{d1} = 0.42$, $h_0 = 0.04$, and $\epsilon_0 = 0.012$. For the outer gaps at $R = 99$ au, we similarly select $w_{d1} = 0.31$, $h_0 = 0.08$, and $\epsilon_0 = 0.017$. Once again, we consider a 1%–5% variation in the gap widths as a conservative approximation for capturing the uncertainty associated with their measurement. The bottom panel of Figure 5 shows the distributions of the predicted masses for the two identified gaps in AS 209, for the given disk parameters and gap widths. The mean predicted planet masses along with the standard deviations are $20.6 \pm 2.2 M_{\oplus}$ and $36.3 \pm 2.5 M_{\oplus}$ at 9 au and 99 au, respectively. The standard deviation captures the total uncertainty, which includes the EP (σ_{EP} is $2.0 M_{\oplus}$ and $2.1 M_{\oplus}$ for the two gaps, respectively) along with the AL. σ_{AL} is comparable to σ_{EP} for the assumed variation of 1%–5% in gap widths. However, σ_{AL} can vary, depending on the noise in the input data. These estimates, particularly at $R = 9$ au, are lower compared to Zhang et al. (2018). The discrepancy is partially due to differences in the ways that the gap widths are measured,

the uncertainty in the adopted disk parameters, and the simulation setups.

7. Discussion and Conclusion

In this paper, we have introduced DPNNNet-Bayesian, a BNN-based model that characterizes the masses of unseen exoplanets from observed PPDs. Compared to its predecessor, DPNNNet-1.0, the Bayesian approach has the following advantages. First, it quantifies the errors associated with the predictions of the network and traces the source of the uncertainty. Second, it gives a framework to understand various regularization techniques that are used in traditional deep-learning approaches (Jospin et al. 2020). Finally, compared to traditional deep-learning models, where the weights are initialized by some implicit prior, the Bayesian framework accounts for this prior distribution explicitly.

For a BNN, one trains a distribution/ensemble of deep-learning models and then, based on the Bayesian formalism, performs model averaging/marginalization of the output. It is to be noted that the BNN approach is fundamentally different from “deep ensembles,” where different initializations are used to train a set of networks, with the outputs being combined/optimized. More so for the BNN approach, a weighted

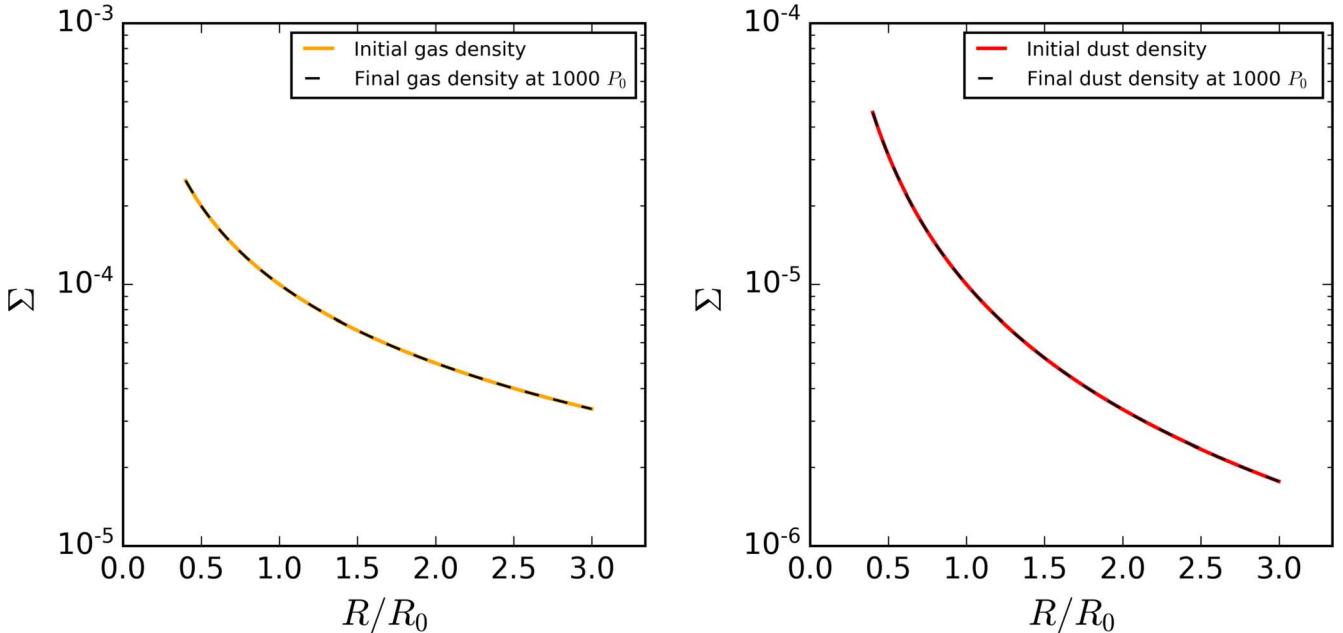


Figure 6. The radial profiles of the azimuthally averaged gas and dust surface densities at the beginning of the simulation and after $1000 P_0$. The steady-state profile is maintained with little evolution, in the absence of an embedded planet.

averaging is achieved by taking into account the posteriors of the network parameters as weights. This is different from deep ensembles, where the averaging is directly performed with the output of the network.

Furthermore, the error/uncertainty associated with the output of a BNN model is different from the standard metrics, like the rms error (RMSE) and/or the mean absolute error (MAE) that are used in traditional deep-learning models. Both the MAE and the RMSE are measures of the error ($|M_{P,\text{simulation}} - M_{P,\text{predicted}}|$) when the model is deployed on a validation/testing data set for which the “true” target values are known. However, they do not necessarily quantify the errors accurately (or provide a confidence interval) when applied to an unknown data set for which the target values are not known. This can be problematic, as there is no way of knowing how accurate the prediction is, so one needs to trust the validation/testing error as an indication of the accuracy of the prediction. Since a BBN model works on the principle of marginalization, rather than optimization, the output is obtained by sampling from a posterior distribution, thus providing the necessary confidence interval.

The performance of any ML-based model depends on the quality and/or quantity of the training data set. The Bayesian approach helps us to understand the correlation between the quality/quantity of the data set and the model performance by monitoring the change in model uncertainty. The knowledge of the EP and the AL enables the model to learn from a smaller data set, by regulating the overfitting or underfitting. Thus, the use of BNN architecture plays a pivotal role in improving the training process, as well as determining the optimal deep-learning architecture.

We conclude by highlighting the key results:

- DPNNet-Bayesian estimates the planet mass from the gap width observed in the dust emission. It takes as its input dust properties (abundance, Stokes numbers) and gas disk properties (aspect ratio, surface density profile, viscosity),

along with dust gap widths from observations, and predicts the embedded planet mass.

- For the training of DPNNet-Bayesian, we generated a data set from disk–planet simulations using the FARGO3D hydrodynamics code, with a newly implemented fixed grain size module and improved initial conditions. The trained model was tested on the simulated data set in order to illustrate its applicability. The predicted mass closely correlated with the “true” value of the simulated planet mass.
- Compared to DPNNet-1.0 (Paper I), the DPNNet-Bayesian architecture additionally estimates the error associated with the prediction of the model. It further distinguishes between the uncertainty pertaining to the deep-learning architecture (EP) and the errors associated with the input variables due to measurement noise (AL).
- We deploy DPNNet-Bayesian to dust gaps observed in the PPDs around HL Tau and AS 209. Our network predicts masses of $86.0 \pm 5.5 M_{\oplus}$, $43.8 \pm 3.3 M_{\oplus}$, and $92.2 \pm 5.1 M_{\oplus}$ for the gaps at 10 au, 30 au, and 80 au, respectively, at HL Tau. Similarly, for the AS 209 disk, we find planet masses of $20.6 \pm 2.2 M_{\oplus}$ and $36.3 \pm 2.5 M_{\oplus}$ at the 9 au and 99 au gaps, respectively. These estimates are in close agreement with the results from other studies based on specialized simulations (Dong et al. 2015; Zhang et al. 2018).

We thank the anonymous referee for the constructive comments. S.A. and J.B.S. acknowledge support from NASA under *Emerging Worlds* through grant 80NSSC20K0702. D.C. acknowledges support from NASA under *Emerging Worlds* through grant 80NSSC21K0037. M.K.L. is supported by the Ministry of Science and Technology of Taiwan (grants 107-2112-M-001-043-MY3, 110-2112-M-001-034-, and 110-2124-M-002-012-) and an Academia Sinica Career Development Award (AS-CDA-110-M06). Numerical simulations were

performed on the TIARA cluster at ASIAA, as well as the TWCC cluster at the National Center for High-performance Computing in Taiwan.

Appendix

Steady-state Drift Solution for Fixed Particle Size

The current public version of the FARGO3D code has a multifluid setup, with pressureless dust species interacting with the gas by means of a drag force parameterized by a constant Stokes number. We update this version of the code to incorporate a steady background solution for the radial drift of dust of fixed particle sizes instead of a fixed Stokes number. Following the generalized steady-state solutions from Benítez-Llambay et al. (2019) for a vertically integrated disk with an isothermal equation of state, with pressure $P = c_s^2 \Sigma$, we rewrite the exact background solutions for the gas and dust (i.e., in the limit of zero coupling) as

$$\mathbf{v}_g = \beta(R) \mathbf{v}_K, \quad (\text{A1})$$

$$\mathbf{v}_d = \mathbf{v}_K, \quad (\text{A2})$$

respectively, where $\mathbf{v}_K = R\Omega_K(r)\hat{\phi}$ is the Keplerian velocity, $\hat{\phi}$ is the unit vector in the azimuth, and

$$\beta = \sqrt{1 + 2\eta(R)}, \quad \eta = \frac{h^2}{2} \frac{d \log P}{d \log R}. \quad (\text{A3})$$

When the dust and gas are coupled, we write their velocities as $\mathbf{v}_{d,g} \rightarrow \mathbf{v}_{d,g} + \delta\mathbf{v}_{d,g}$. Considering the steady-state axisymmetric Navier–Stokes equations for perturbed velocities (see Equations (63)–(66) in Benítez-Llambay et al. 2019), the gas and dust velocity perturbations in the radial direction are given as

$$\delta v_{gR}(R) = -2\beta Q_N \Psi(\beta - 1) v_K, \quad (\text{A4})$$

$$\delta v_{dR}(R) = \frac{2S_t}{1 + S_t^2}(\beta - 1)v_K + \frac{\delta v_{gR} + 2S_t \delta v_{g\phi}}{1 + S_t^2}, \quad (\text{A5})$$

and the azimuthal counterpart is

$$\delta v_{g\phi}(R) = -[(\mathcal{S}_N + 2\xi)\mathcal{S}_N + Q_N^2 \Psi(\beta - 1)v_K], \quad (\text{A6})$$

$$\delta v_{d\phi}(R) = \frac{1}{1 + S_t^2}(\beta - 1)v_K + \frac{2\delta v_{g\phi} - S_t \delta v_{gR}}{2(1 + S_t^2)}, \quad (\text{A7})$$

where

$$\mathcal{S}_N \equiv \frac{\epsilon}{1 + S_t^2}, \quad Q_N \equiv \frac{\epsilon S_t}{1 + S_t^2}, \quad (\text{A8})$$

$$\Psi = [(\mathcal{S}_N + \beta)(\mathcal{S}_N + 2\xi) + Q_N^2] \times \text{and} \quad \xi \equiv \beta \left(\frac{1}{2} + \frac{d \log P}{d \log R} \right). \quad (\text{A9})$$

Expanding Equations (A4) to (A7) using Equations (A8) and (A9), we get

$$\delta v_{gR}(R) = \frac{-2\beta(\beta - 1)v_K S_t \epsilon}{\epsilon^2 + \epsilon\beta(2\gamma + 3) + 2\beta^2(\gamma + 1) + 2\beta^2 S_t^2(\gamma + 1)} \quad (\text{A10})$$

$$\delta v_{g\phi}(R) = -\frac{(\beta - 1)\epsilon v_K (2\beta\gamma + \epsilon + 2\beta)}{\epsilon^2 + 2\beta^2 S_t^2(\gamma + 1) + \beta\epsilon(2\gamma + 3) + 2\beta^2(\gamma + 1)}, \quad (\text{A11})$$

$$\delta v_{dR} = \frac{4\beta^2(\beta + 1)S_t v_K(\gamma + 1)}{\epsilon^2 + \epsilon\beta(2\gamma + 3) + 2\beta^2(\gamma + 1) + 2\beta^2 S_t^2(\gamma + 1)}, \quad (\text{A12})$$

$$\delta v_{d\phi} = \frac{v_K(\beta - 1)(\epsilon\beta + 2\beta^2(\gamma + 1))}{\epsilon^2 + \epsilon\beta(2\gamma + 3) + 2\beta^2(\gamma + 1) + 2\beta^2 S_t^2(\gamma + 1)}. \quad (\text{A13})$$

The above solutions, Equations (A10)–(A13), are solutions to the steady-state drift solution only if they satisfy the following continuity equations:

$$\partial_R(R\Sigma_g \delta v_{gR}) = \partial_R(R\Sigma_d \delta v_d) = 0. \quad (\text{A14})$$

The particle size is parameterized by the Stokes number, $S_t = t_s \Omega_K$, where the stopping time characterizing the strength of the dust–gas coupling is given as

$$t_s = \rho_p s / (\rho_g c_s) = \frac{\pi s \rho_p}{2\Sigma_g}. \quad (\text{A15})$$

Here, ρ_p is the density of the dust particles, s is the radius of the dust particles, and Σ_g is the gas surface density. For a fixed particle size, $s = \text{constant}$ while the Stokes number $S_t \propto \Sigma_g^{-1}$ is a function of disk radius, since $\Sigma_g \propto R^{-\sigma_0}$, unlike the default setup, where S_t is set to constant.

To maintain a constant mass flux, i.e., $\partial_R(R\Sigma_g \delta v_{gR}) = \partial_R(R\Sigma_d \delta v_d) = 0$, we set

$$R\Sigma_g \delta v_{gR} = \text{constant}. \quad (\text{A16})$$

Here, we only consider the gas component, but same result can be arrived at for dust as well. Furthermore, since $R\Sigma_g \propto \left(\frac{R}{R_0}\right)^{1-\sigma_0}$, δv_{gR} must scale as

$$\delta v_{gR} \propto \left(\frac{R}{R_0}\right)^{\sigma_0-1}, \quad (\text{A17})$$

such that the net mass flux is constant. Thus, from Equations (A10) and (A17) we can establish

$$\delta v_{gR}(R) = \delta v_{gR0} \left(\frac{R}{R_0}\right)^{\sigma_0-1} = \frac{a\epsilon}{\epsilon^2 + b\epsilon + c}, \quad (\text{A18})$$

where $a = -2\beta(\beta - 1)v_K S_t \epsilon$, $b = \beta(2\gamma + 1)$, $c = 2\beta^2(\gamma + 1) + 2\beta^2 S_t^2(\gamma + 1)$, and $\delta v_{gR0} = \frac{a_0 \epsilon_0}{\epsilon_0^2 + b_0 \epsilon_0 + c_0}$, with the subscript “0” referring to the values at $R = R_0$. Rearranging Equation (A18), we solve for the dust-to-gas ratio ϵ and arrive at

$$\epsilon = \frac{-(bd_0 - a) \pm \sqrt{(bd_0 - a)^2 - 4d_0^2 c}}{2d_0}, \quad (\text{A19})$$

where $d_0 = \frac{a_0 \epsilon_0}{\epsilon_0^2 + b_0 \epsilon_0 + c_0} \left(\frac{R}{R_0}\right)^{\sigma_0-1}$ and the negative root gives a physical solution. Thus, in order to satisfy the condition for constant mass flux for a fixed particle size, we adjust the dust-to-gas ratio ϵ accordingly.

Furthermore, the viscous velocity of the gas V_{vis}^{2D} without dust grains is given as (Lynden-Bell & Pringle 1974)

$$V_{\text{vis}}^{2D} = -\frac{3\nu}{R} \frac{d \ln(\nu \Sigma_g R^{1/2})}{d \ln R}, \quad (\text{A20})$$

where $\nu = \alpha c_s h$ is the kinematic viscosity. For the steady-state background solution, i.e., $V_{\text{vis}}^2 = 0$, we set $\nu \Sigma_g R^{1/2} = \text{constant}$ in Equation (A20), which is equivalent to

$$\alpha = \alpha_0 (R/R_0)^{\sigma_0 - 2F - 1}, \quad (\text{A21})$$

where the subscript “0” refers to the values at $R = R_0$.

We initialize the disk–planet FARGO3D simulation with the updated dust-to-gas ratio in Equation (A19) and fix the particle size to a constant value using the fixed particle module. Figure 6 shows the azimuthally averaged initial dust and gas radial surface densities and the final surface density profiles after the system has evolved $1000P_0$. As demonstrated, the steady-state profile is maintained with little evolution, in the absence of an embedded planet.

ORCID iDs

Sayantan Auddy  <https://orcid.org/0000-0003-3784-8913>
 Ramit Dey  <https://orcid.org/0000-0001-5717-1589>
 Min-Kai Lin  <https://orcid.org/0000-0002-8597-4386>
 Daniel Carrera  <https://orcid.org/0000-0001-6259-3575>
 Jacob B. Simon  <https://orcid.org/0000-0002-3771-8054>

References

Abdar, M., Pourpanah, F., Hussain, S., et al. 2021, *Inf. Fusion*, 76, 243
 ALMA Partnership, Brogan, C. L., Pérez, L. M., et al. 2015, *ApJL*, 808, L3
 Andrews, S. M., Wilner, D. J., Zhu, Z., et al. 2016, *ApJL*, 820, L40
 Auddy, S., Dey, R., Lin, M.-K., & Hall, C. 2021, *ApJ*, 920, 3
 Auddy, S., & Lin, M.-K. 2020, *ApJ*, 900, 62
 Bardeau, R., Doucet, A., & Holmes, C. 2017, *JMLR*, 18, 1515
 Benítez-Llambay, P., Krapp, L., & Pessah, M. E. 2019, *ApJS*, 241, 25
 Benítez-Llambay, P., & Masset, F. S. 2016, *ApJS*, 223, 11
 Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. 2017, *J. Am. Stat. Assoc.*, 112, 859
 Crida, A., Morbidelli, A., & Masset, F. 2006, *Icar*, 181, 587
 de Val-Borro, M. E. 2006, *MNRAS*, 370, 529
 Dillon, J. V., Langmore, I., Tran, D., et al. 2017, arXiv:1711.10604
 Dipierro, G., Price, D., Laibe, G., et al. 2015, *MNRAS*, 453, 73
 Dong, R., Zhu, Z., & Whitney, B. 2015, *ApJ*, 809, 93
 Duffell, P. C. 2015, *ApJL*, 807, L11
 Duffell, P. C., & Macfadyen, A. I. 2013, *ApJ*, 769, 41
 Fedele, D., Tazzari, M., Booth, R., et al. 2018, *A&A*, 610, 24
 Fischer, D. A., Howard, A. W., Laughlin, G. P., et al. 2014, in *Protostars and Planets VI*, ed. H. Beuther et al., Vol. 715 (Tucson, AZ: Univ. Arizona Press)
 Fung, J., Shi, J.-M., & Chiang, E. 2014, *ApJ*, 782, 88
 Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (Cambridge, MA: MIT Press)
 Haffert, S. Y., Bohn, A. J., de Boer, J., et al. 2019, *NatAs*, 3, 749
 Huang, J., Andrews, S. M., Cleeves, L. I., et al. 2018a, *ApJ*, 852, 122
 Huang, J., Andrews, S. M., Pérez, L. M., et al. 2018b, *ApJL*, 869, L43
 Hüllemeier, E., & Waegeman, W. 2021, *Mach. Learn.*, 110, 457
 Ilee, J. D., Hall, C., Walsh, C., et al. 2020, *MNRAS*, 498, 5116
 Iman, R. L., Helton, J. C., & Campbell, J. E. 1981, *JQT*, 13, 174
 Jacquet, E., Balbus, S., & Latter, H. 2011, *MNRAS*, 415, 3591
 Jin, S., Li, S., Isella, A., Li, H., & Ji, J. 2016, *ApJ*, 818, 76
 Jospin, L. V., Buntine, W., Boussaid, F., Laga, H., & Bennamoun, M. 2020, arXiv:2007.06823
 Kanagawa, K. D., Muto, T., Tanaka, H., et al. 2015a, *ApJL*, 806, L15
 Kanagawa, K. D., Muto, T., Tanaka, H., et al. 2016, *PASJ*, 68, 43
 Kanagawa, K. D., Tanaka, H., Muto, T., Tanigawa, T., & Takeuchi, T. 2015b, *MNRAS*, 448, 994
 Kendall, A., & Gal, Y. 2017, arXiv:1703.04977
 Keppler, M., Benisty, M., Müller, A., et al. 2018, *A&A*, 617, A44
 Lee, C.-H. 2018, *Galax*, 6, 51
 Leung, H. W., & Bovy, J. 2019, *MNRAS*, 483, 3255
 Lodato, G., Dipierro, G., Ragusa, E., et al. 2019, *MNRAS*, 486, 453
 Lynden-Bell, D., & Pringle, J. E. 1974, *MNRAS*, 168, 603
 McKay, M. D., Beckman, R. J., & Conover, W. J. 1979, *Technometrics*, 21, 239
 Mohan, D., Scaife, A. M. M., Porter, F., Walmsley, M., & Bowles, M. 2022, *MNRAS*, 511, 3722
 Paardekooper, S.-J., & Papaloizou, J. C. B. 2009, *MNRAS*, 394, 2297
 Perreault Levasseur, L., Hezaveh, Y. D., & Wechsler, R. H. 2017, *ApJL*, 850, L7
 Pinte, C., Price, D. J., Ménard, F., et al. 2018, *ApJL*, 860, L13
 Pinte, C., van der Plas, G., Ménard, F., et al. 2019, *NatAs*, 3, 1109
 Pinte, C., Price, D. J., Ménard, F., et al. 2020, *ApJL*, 890, L9
 Roberts, S., McQuillan, A., Reece, S., & Aigrain, S. 2013, *MNRAS*, 435, 3639
 Shakura, N. I., & Sunyaev, R. A. 1973, *A&A*, 24, 337
 Shrestha, A., & Mahmood, A. 2019, *IEEE Access*, 7, 53040
 Spindler, A., Geach, J. E., & Smith, M. J. 2021, *MNRAS*, 502, 985
 Teague, R., Bae, J., Bergin, E. A., Birnstiel, T., & Foreman-Mackey, D. 2018, *ApJL*, 860, L12
 Toomre, A. 1964, *ApJ*, 139, 1217
 Wagner, K., Follette, K. B., Close, L. M., et al. 2018, *ApJL*, 863, L8
 Weidenschilling, S. J. 1977, *MNRAS*, 180, 57
 Wilson, A. G., & Izmailov, P. 2020, arXiv:2002.08791
 Zhang, C., Bütepage, J., Kjellström, H., & Mandt, S. 2019, *ITPAM*, 41, 2008
 Zhang, S., Zhu, Z., & Kang, M. 2022, *MNRAS*, 510, 4473
 Zhang, S., Zhu, Z., Huang, J., et al. 2018, *ApJL*, 869, L47