The Astrophysical Journal, 920:3 (12pp), 2021 October 10 https://doi.org/10.3847/1538-4357/ac1518

DPNNet-2.0. I. Finding Hidden Planets from Simulated Images of Protoplanetary Disk Gaps

Sayantan Auddy[1,2], Ramit Dey[3], Min-Kai Lin[1,4], and Cassandra Hall[5,6]
[1] Institute of Astronomy and Astrophysics, Academia Sinica, Taipei 10617, Taiwan; sauddy@iastate.edu, sayantanauddy21@gmail.com
[2] Department of Physics and Astronomy, Iowa State University, Ames, IA 50010, USA
[3] School of Mathematical and Computational Sciences, Indian Association for the Cultivation of Science, Kolkata-700032, India
[4] Physics Division, National Center for Theoretical Sciences, Taipei 10617, Taiwan
[5] Department of Physics and Astronomy, University of Georgia, Athens, GA 30602, USA
[6] Center for Simulational Physics, University of Georgia, Athens, GA 30602, USA
Received 2021 February 25; revised 2021 July 11; accepted 2021 July 14; published 2021 October 6

Abstract

The observed substructures, like annular gaps, in dust emissions from protoplanetary disks are often interpreted as signatures of embedded planets. Fitting a model of planetary gaps to these observed features using customized simulations or empirical relations can reveal the characteristics of the hidden planets. However, customized fitting is often impractical owing to the increasing sample size and the complexity of disk–planet interaction. In this paper we introduce the architecture of DPNNet-2.0, second in the series after DPNNet, designed using a convolutional neural network (CNN, specifically ResNet50 here) for predicting exoplanet masses directly from simulated images of protoplanetary disks hosting a single planet. DPNNet-2.0 additionally consists of a multi-input framework that uses both a CNN and multilayer perceptron (a class of artificial neural network) for processing image and disk parameters simultaneously. This enables DPNNet-2.0 to be trained using images directly, with the added option of considering disk parameters (disk viscosities, disk temperatures, disk surface-density profiles, dust abundances, and particle Stokes numbers) generated from disk–planet hydrodynamic simulations as inputs. This work provides the required framework and is the first step toward the use of computer vision (implementing CNNs) to directly extract the mass of an exoplanet from planetary gaps observed in dust surface-density maps by telescopes such as the Atacama Large Millimeter/submillimeter Array.

Unified Astronomy Thesaurus concepts: Exoplanet detection methods (489); Convolutional neural networks (1938); Neural networks (1933); Protoplanetary disks (1300)

1. Introduction

Exoplanet surveys using different techniques (Butler et al. 2017; Deeg & Alonso 2018; Hojjatpanah et al. 2019) have revealed a remarkable variety of planets and planetary systems throughout the Galaxy (Cassan et al. 2012; Batalha et al. 2013). With more than 4000 confirmed detections, and an even larger number of detected objects awaiting confirmation, we have a rich sample of data characterizing the demographics of exoplanets. However, most of these discovered exoplanets are much older ($\sim 10^3$ Myr). To get a complete picture of the planet formation processes and their formation environments (Raymond & Morbidelli 2020) it is imperative to constrain the distribution of sizes and diversity in radii, composition, and mass (e.g., Winn & Fabrycky 2015; Fulton et al. 2017) of young planets including those at their formation stage.

However, with current search techniques (Fischer et al. 2014; Lee 2018) it is particularly challenging to detect young planets embedded within their natal disk. The spectral characteristic and the light curves from such systems are too faint compared to the emission from the young host star and the protoplanetary disk (hereafter PPDs). Thus, direct detection in the visible and infrared spectrum from Hα (Cugno et al. 2019; Zurlo et al. 2020), or thermal emission from the gradually cooling planets embedded within the disks, are rare as well as difficult. These factors add up to severely limit the detection of young exoplanets during their formation epochs.

An alternative approach is to search for indirect influences/signatures of these unseen planets on the physical structure of the PPDs induced by disk–planet interaction. Theoretical models have long predicted disk–planet interactions (Goldreich & Tremaine 1980; Lin & Papaloizou 1993) resulting in spiral features and/or annular gaps in gas and dust density distribution (Lin & Papaloizou 1986). However, it is only recently that high-resolution observations in submillimeter interferometry using the Atacama Large Millimeter/submillimeter Array (ALMA) and other facilities (for example SPHERE, Gemini) have reported such complex—possibly planet-induced—features with unprecedented precision (e.g., Andrews et al. 2018; Clarke et al. 2018; Dipierro et al. 2018; Huang et al. 2018b, 2018c; Long et al. 2018, 2020; Pérez et al. 2018; Liu et al. 2019; van der Marel et al. 2019).

Near-infrared images from ALMA have revealed the ubiquity of concentric rings and gaps in many observed systems, like HL Tau (ALMA Partnership et al. 2015), TW Hya (Andrews et al. 2016; Huang et al. 2018a), HD 97048 (van der Plas et al. 2017), and HD 169141 (Momose et al. 2015). There are many possible explanations for the formation of these observed structures. Disk-specific mechanisms such as secular gravitational instability (Youdin 2011; Takahashi & Inutsuka 2014), magnetorotational instability (MRI) turbulence (e.g., Johansen et al. 2009; Simon & Armitage 2014), self-induced dust pileups (Gonzalez et al. 2015), gap opening in the dust spatial distribution due to large-scale vortices (Barge et al. 2017), and radially variable magnetic disk winds (Suriano et al. 2018) are some of the possible scenarios that can lead to the formation of these substructures in a dusty disk. However, perhaps the most popular interpretation is that these substructures are induced by gap-opening planets (Dipierro et al. 2016; Rosotti et al. 2016; Dong & Fung 2017).

Recent detection of planetary signatures (velocity kinks) using gas kinematic measurements (Pinte et al. 2018, 2019, 2020; Teague et al. 2018), particularly inside the observed dust gaps, provide strong evidence in support of planet-induced gaps. More so, direct detection of an accreting planet, using Hα emissions, inside the gap/cavity of the transition disk PDS-70 (Keppler et al. 2018; Wagner et al. 2018; Haffert et al. 2019) strengthens the idea of planetary gaps. This further suggests that planets may form much earlier than previously thought as most of these PPDs are relatively young ($<$10 Myr). This opens up a unique window to probe unseen, young, and fully grown exoplanets by characterizing substructures (particularly axisymmetric disk gaps) in PPDs from dust emission.

Disk–planet interactions are studied extensively using generic hydrodynamical simulations to reproduce dust distributions that are comparable with observations (Zhang et al. 2018). Planet-induced gap features, mostly width and depth, are modeled both using numerical and analytic approaches (e.g., Crida et al. 2006; Paardekooper & Papaloizou 2009; Duffell & Macfadyen 2013; Fung et al. 2014; Duffell 2015; Kanagawa et al. 2015; Ilee et al. 2020). In order to infer planet properties (such as planet mass), one typically needs to match observations with outputs from these customized dusty disk–planet simulations. Each simulation is characterized using multiple parameters, like local disk aspect ratio, disk density profile, dimensionless viscosity parameter, particle size, and dust abundance (or metallicity) along with the gap features of gap depth and width. Furthermore, as the disk models improve with the inclusion of additional physics, like migrating planets (Paardekooper et al. 2010; Meru et al. 2019), multiple gaps and rings (Wafflard-Fernandez & Baruteau 2020), and improved thermodynamics (Miranda & Rafikov 2019, 2020), more parameters are necessary to set up the simulation. Thus, it becomes impractical to perform disk–planet simulations spanning such a large ever-increasing parameter space to match each observed target in the current and future surveys of PPDs. It is therefore essential to seek a generic, future-proof, and accurate model for planet-induced gaps.

1.1. DPNNet-1.0

In Auddy & Lin (2020) (hereafter Paper I) we introduced Disk Planet Neural Network (DPNNet), a deep-learning model which predicts the planet mass from simulated/observed disk properties. It outperforms simple analytic models (for examples, see Kanagawa et al. 2016; Lodato et al. 2019) in its ability to encapsulate the multidimensional parameter space, which is always a challenge with analytic relations. DPNNet was trained with synthetic data, normalized using the standard scaling (z-score) by removing the mean and then scaling it by the standard deviation, generated from numerical simulations using the FARGO3D hydrodynamics code (Benítez-Llambay et al. 2019).

DPNNet's architecture consists of a fully connected multilayer perceptron (MLP), which takes inputs directly from the user to predict the planet mass (see Figure 1 in Paper I). For disks with observed gaps, one of the key input features is the dust gap width. The specific way in which the gap widths are measured (or the definition of gap width) varies somewhat across the literature. In general, the gap width is defined as the radial distance between the inner and the outer edge of the gap where the surface density reaches a predefined threshold, Σ_T. In Paper I, consistent with Kanagawa et al. (2016), we set Σ_T to half of the undepleted initial surface density. This differs from Dong & Fung (2017), where Σ_T is geometric mean of the undepleted and the minimum surface density, as well as Zhang et al. (2018), where Σ_T is the average of the peaks and the minimum (gap) surface density. Similarly, different functional forms were used by Clarke et al. (2018) (difference of two logistic functions) and Long et al. (2018) (symmetrical Gaussian) to fit the radial dust profile to measure the gap width. Such variations limit the application of DPNNet, requiring the network to be trained each time for different definitions of gap width.

Additionally, DPNNet only considers the gap width, while ignoring other morphological features like gap depth (Fung et al. 2014), asymmetries such as vortices (de Val-Borro et al. 2007; Hammer et al. 2016, 2018), and spirals (Zhu et al. 2015). Quantifying these features using user-defined probes can be a complex task and further creates the scope for more uncertainties. An advanced approach is to use computer vision to initiate image-based characterization of the disk features. This facilitates the extraction of complex nonlinear features from images directly without explicit user interference, thereby minimizing the associated errors.

1.2. Introduction to DPNNet-2.0

In this paper we introduce DPNNet-2.0, a deep neural network model based on the convolutional neural network (CNN) architecture, to predict mass of super-Earth to Saturn-sized planets from simulated images of PPDs. Compared to DPNNet (with MLP architecture) that requires users to measure disk features from images, for instance obtaining the gap widths as defined by the user, DPNNet-2.0 is an end-to-end model that learns the linear and the nonlinear features directly from the disk images. This minimizes biases associated with different user-defined measurement probes. Additionally, DPNNet-2.0 includes a complementary module that allows it to accept disk parameters (feature variables) like disk aspect ratio, disk viscosity, disk surface-density profile, metallicity, and particle size in order to better constrain the disk properties. This gives DPNNet-2.0 the flexibility to accept both images and disk attributes simultaneously. As we will demonstrate in this paper, the use of mixed data inputs improves the model performance while trained on the same data set.

1.3. Paper Outline

The paper is outlined as follows. In Section 2 we describe the disk–planet hydrosimulations as well as the parameter space considered for the current study. We introduce the CNN architecture implemented in DPNNet-2.0 along with the multi-input module in Section 3 and discuss in detail the steps used to preprocess the data. In Section 4 we present the prediction from our trained DPNNet-2.0, based on simulated images, using both the single and the multi-input module. Finally, in Section 5 we discuss the limitations and future scopes.

2. Disk–Planet Simulations

We are interested in modeling the structural variations of the dust surface density in a dusty protoplanetary disk due to an embedded planet. We develop a machine-learning (ML) algorithm to identify and classify these variations. Our ML model is trained using sample images generated from hydrodynamic simulations of disk–planet systems. The details of the ML model and the training step are given in the following section. The physical disk model and the simulations used to generate the training data are discussed in detail in Paper I. However, for convenience of the reader we briefly review the relevant details.

2.1. Disk Setup

We consider a razor-thin 2D disk with a nonmigrating planet of mass $M_{\rm P}$ placed at $R = R_0$, on a circular Keplerian orbit around a central star of mass M_*. The planet's orbital period $P_0 = 2\pi/\Omega_{\rm K0}$ is used as the unit of time, where $\Omega_{\rm K} = \sqrt{GM_*/R^3}$ is the Keplerian frequency. The subscript 0 denotes evaluation at the planet's location R_0. We set $G = R_0 = M_* = 1$, where G is the gravitational constant, such that the units are all dimensionless. The disk is initialized with a gas surface-density profile given as

$$\Sigma_{\rm g}(R) = \Sigma_{\rm g0}\left(\frac{R}{R_0}\right)^{-\sigma}, \tag{1}$$

where σ is the exponent of the density profile and $\Sigma_{\rm g0} = 10^{-4}$ in code units. The disk's self-gravity is negligible as our models are gravitationally stable with a Toomre parameter $\mathcal{Q}_0 \gtrsim 80$. The disk is locally isothermal having a sound-speed profile $c_{\rm s}(R) = h_0 R\Omega_{\rm K}$ and a constant flaring index F, such that aspect ratio is

$$h = \frac{H}{R} = h_0\left(\frac{R}{R_0}\right)^{F}, \tag{2}$$

where H is the disk-scale height, and h_0 is the aspect ratio at the planet's location $R = R_0$. The disk turbulence is modeled using the standard α prescription of Shakura & Sunyaev (1973), where the kinematic viscosity $\nu = \alpha c_{\rm s}^2/\Omega_{\rm K}$ and α is a constant.

We consider gas and dust simulations where the dust particles are modeled as a pressureless fluid (Jacquet et al. 2011). The dust particles are parameterized using constant Stokes numbers $S \equiv t_{\rm s}\Omega_{\rm K}$, where the dust–gas coupling (Weidenschilling 1977) is characterized using the stopping time $t_{\rm s}$. The dust surface density is initialized as $\Sigma_{\rm d} = \epsilon\Sigma_{\rm g}$, where ϵ is the constant dust-to-gas ratio. We include the dust back reaction onto the gas.

2.2. Simulation Setup and Parameter Space

We use the FARGO3D hydrodynamic code (Benítez-Llambay et al. 2019), running in a 2D cylindrical geometry (R, ϕ), to simulate the dusty disk–planet system. The grid is uniformly spaced with 512×512 discretized cells. The computational domain spans from $R \in [0.4, 2.5]R_0$ and $\phi \in [0, 2\pi]$. The radial boundaries are set to their initial equilibrium solution while we apply periodic boundary conditions in ϕ. We damp waves generated by the planet to zero near the boundaries using the wave-killing module (de Val-Borro 2006) to avoid reflections and interference with the disk bulk/morphology. The planet is placed from the beginning of the simulation. We use a constant softening length of $r_s = 0.6h_0R_0$ for the planet's potential. The frame of reference rotates with the planet.

Table 1
Parameter Space for Hydrodynamic Simulations

Name	Notation	Max.	Min.	Mean	Std
Planet mass in Earth masses	$M_{\rm P}/M_\oplus$	120.0	8.0	64.0	32.3
Disk aspect ratio	h_0	0.100	0.025	0.063	0.022
Disk surface-density profile	σ	1.50	0.50	0.75	0.29
Disk viscosity parameter	α ($\times 10^{-3}$)	10	0.10	5.00	2.86
Global dust-to-gas ratio	ϵ	0.10	0.01	0.06	0.03
Particle Stokes numbers	$S_{\rm t}$	0.100	0.001	0.051	0.029

We consider the same parameter space, summarized in Table 1, as used in Paper I. This enable us to compare the relative performance between our previous ML model DPNNet and the current improved CNN model DPNNet-2.0. We follow each simulation for 3000 orbits which translates to about ~1 Myr at 45 au around a 1 $M_\odot$ star. The range of Stokes number considered here scale to particle radii ranging from ~1 mm to ~10 cm assuming gas surface density of 100 g cm^{-2} and an internal grain density of 1 g cm^{-3}. We target super-Earth to Saturn-sized planets, $M_{\rm P} \in [8, 120]M_\oplus$, as these are comparatively difficult to detect using conventional methods, thus making our deep-learning model widely applicable.

3. DPNNet-2.0

In this section we briefly introduce the CNN architecture and its implementation in designing the DPNNet-2.0 model. Further, we discuss the process of data acquisition, data preprocessing, and DPNNet-2.0 training.

3.1. Convolutional Neural Network

A CNN is a type of feed-forward neural network consisting of convolutional layers followed by fully connected layers (Rawat & Wang 2017) that can be used for classification or regression problems. Figure 1 shows a typical CNN architecture along with the various layers within it. The crucial part of a CNN architecture is the introduction of convolutional layers, consisting of a set of neurons having shared weights. The convolutional layer can be viewed as a layer consisting of several neurons that analyses a small overlapping section of the input image/data. This idea was first introduced by Fukushima & Miyake (1982) and LeCun et al. (1998), where the convolution operations were implemented in one or multiple layers in addition to the fully connected layers in the end.

CNNs have one major advantage over MLP-based models. Due to the weight sharing as well as the small kernels, the system requirements for CNN is significantly reduced compared to MLP (with fully connected layers) performing general matrix multiplication. This opened up the possibility of building faster, more cost effective, and deeper networks. They can perform exceptionally well in extracting features from an image (from low levels to higher levels) because of the hierarchical learning, enabled by stacking several convolutional filters having their own shared weights. These networks also come with the advantage that they can automatically deal with spatial translations (rotation and scaling as well with some modifications).

Even though CNNs were initially designed to perform image classification (Krizhevsky et al. 2017) or object detection

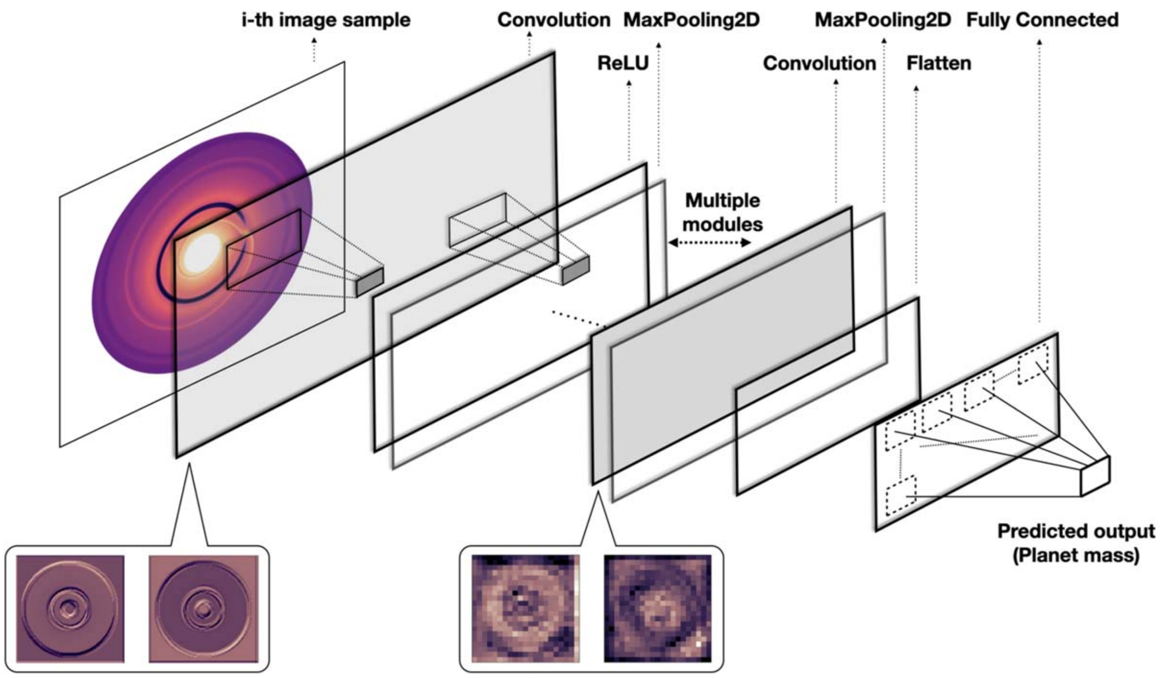

Figure 1. A schematic diagram showing a typical/vanilla CNN architecture along with the various hidden layers. It consists of multiple convolution and subsampling (MaxPooling2D) layers followed by a nonlinearity (ReLU in this case). The characteristic feature of this architecture is the presence of a fully connected layer with linear activation for performing regression. The feature maps, showing both high- and low-level features extracted by the convolutional filters, are shown in the extended panels. The final output is the planet mass. Note that this schematic diagram is for illustration purposes only and the convolutional layers do not show the ResNet50 architecture used in DPNNet-2.0. While the input and the output layers remain the same, the convolutional layers can be swapped with different CNN architectures. For additional details see Section 3.2.

(Girshick et al. 2014), in recent times these are also used for solving regression problems. This is implemented using a linear layer following the fully connected layer in the end. Using this approach, commonly known as *vanilla deep regression*, state-of-the-art results are obtained in computer vision regression problems (Toshev & Szegedy 2014; Belagiannis et al. 2015; Liu et al. 2016).

3.2. DPNNet-2.0 Architecture

DPNNet-2.0 uses a CNN architecture to predict planet mass using features extracted from input images directly. We tested the performance of DPNNet-2.0 with several well-established 2D CNNs such as AlexNet (Krizhevsky et al. 2017), VGG-16 (Simonyan & Zisserman 2014), ResNet (He et al. 2016), as well as a *vanilla* CNN consisting of multiple convolution and pooling (subsampling) layers followed by a nonlinearity (ReLU in this case). Our findings suggest that ResNet performs significantly better compared to the other CNN architectures and the training time is much shorter as well (see Figure 9). Based on this, the main analysis of this paper, as well as the benchmarking, is done using ResNet as the CNN block of the DPNNet-2.0 model.

The main reason for the superior performance of the ResNet architecture is the use of “identity shortcut connections” that enable the construction of deeper networks while reducing the number of parameters. While AlexNet has 5 convolutional layers, VGG-16 has 19 layers and ResNet50 has 50 layers. Simply increasing the number of layers to built a deeper network is burdened with the problem of a vanishing gradient, making them harder to train. For the case of ResNet, stacking several layers does not hinder the performance of the network because of the presence of the identity mappings (empty layers that do not do anything and just act as a connection between two layers). Except for the first convolutional layer of the ResNet architecture all other ones have a identity connection, thus making them *residual* convolutional blocks. These shortcuts or “skip connections” allow the direct back-propagation of the gradient to earlier layers. For DPNNet-2.0 we use an Adam optimizer, instead of RMSProp as used in Auddy & Lin (2020), as it performs better in incorporating the use of moving average of previous gradients in order to accelerate learning with inertia.

3.3. Multi-input DPNNet-2.0

Multi-input DPNNet-2.0 is designed as an end-to-end model which can accept both images and disk features as inputs to predict the planet mass. This is crucial, as in addition to images each observed disk–planet system has its own physical characteristics, like disk profile, temperature, viscosity, as well particle size and metallicity. Thus, a complete model for planet-mass prediction needs to be sensitive not only to the disk morphology (observed from images) but also to the disk properties which are measured separately.

This is achieved by developing an additional module that combines the CNN-based architecture with the MLP network implemented in Paper I. The MLP module is a fully connected feed-forward neural network with two hidden layers, having 256 and 128 neurons, and an input layer with five feature variables. Figure 2 gives the schematic of the network architecture of the multi-input DPNNet-2.0.

The image and the disk features are processed independently using the CNN and the MLP architecture in DPNNet-2.0, respectively. The concatenated output from the two branches serves as a input to a fully connected layers of neurons. Finally, it passes through a linear activation regression head to output the predicted planet mass. Note that we amend DPNNet (MLP

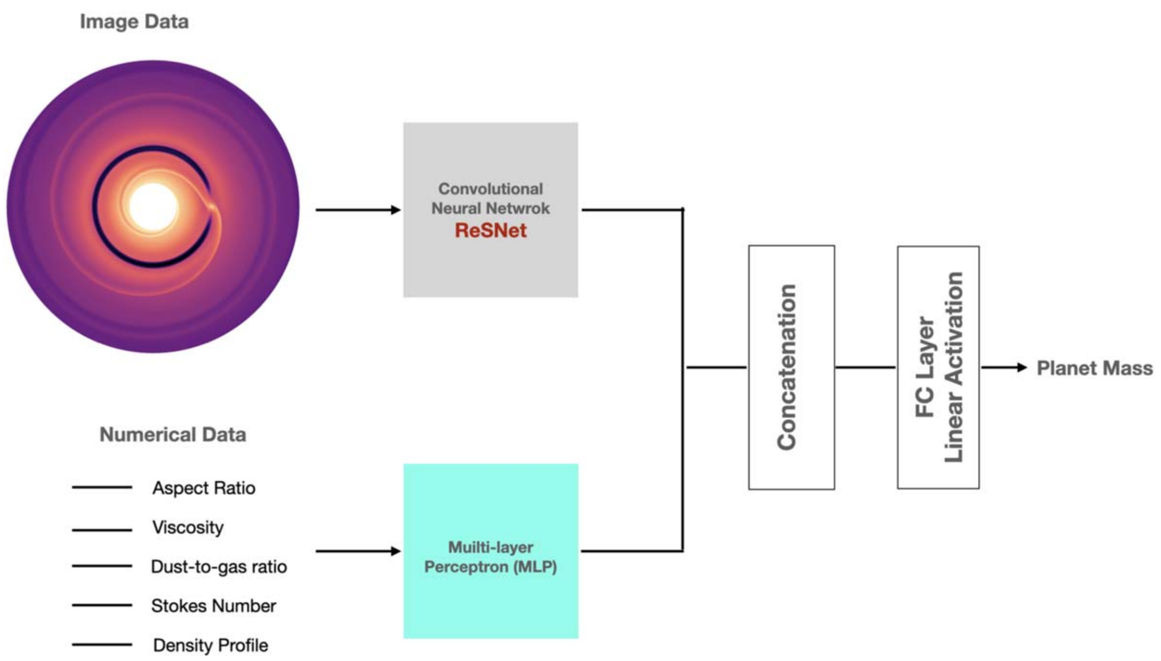

Figure 2. A schematic representation of the multi-input DPNNet-2.0 that can accept both the image and the disk characteristics as inputs. The image is fed to the CNN and the disk parameters propagate through the MLP model. The output from the CNN and the MLP is concatenated and passed through a fully connected (FC) layer and a linear activation layer. The final output is the planet mass. For more details refer to Section 3.3.

model), such that it no longer requires a gap width as input, as the gap features are characterized from the image directly by the CNN. Only the disk initial conditions (α, h_0, σ, S_t, ϵ) are considered. This allowed us to remove the uncertainties associated with DPNNet (user-defined gap-width measurement) while retaining its flexibility.

3.4. Data Acquisition and Preprocessing

We use the Latin hypercube sampling (LHS; McKay et al. 1979; Iman et al. 1981) method to sample the parameter space for initializing the disk–planet simulation. The LHS was implemented using the PYDOE package to generate a uniform random distribution of the parameters. Each parameter value was centered within the sampling intervals. We ran 1200 simulations in total initialized using the above parameter space. For each simulation we generate dust surface-density maps after the system has evolved substantially and reached a quasi-steady state. This corresponds to $\approx$2000 orbits of evolution for most of our disk–planet simulation. Thus we generate dust density maps from 2400, 2600, 2800, and 3000 orbits of evolution. Thereby, for each simulation we have images from four time instances resulting in a total of 4800 unique images for 1200 distinct simulations. This gives us a much larger sample size, which is necessary for better optimization of the CNN.

Figure 3 shows a normalized dust surface-density distribution for a set of sample simulations that are used to train the DPNNet-2.0. The most distinct and visible features are the presence of dust gaps and rings. These annular structure can be broadly categorized as (a) a single deep gap of increasing width (left to right) as seen in the top row, (b) two gaps adjacent to a planet (first two panels from the left in the bottom row), and (c) a deep cavity as evident in the rightmost plot in the bottom row in Figure 3. For details about the characteristics of the gaps refer to Section 3.2 in Paper I.

The images are initially filtered to eliminate runs that have undetectable gaps or more than two gaps. After the initial screening we are left with 2945 images combined from all the four different time instances. We crop the images to remove the axis and the color bars. Then they are resized to exactly match the required resolution, which is this case was 512×512. The cropped images are further preprocessed before feeding them as inputs to the network. The pixel values of the images ranges between 0 and 255. We normalize each pixel value within the range 0–1, as inputs with larger values can slow down/disrupt the training process. Furthermore, we implement a standardization technique where the pixel values are distributed normally with a mean of 0 and standard deviation of 1.

Each image is associated with a set of input parameters M_P, α, h_0, σ, S_t, ϵ. We assign each image a target variable planet mass M_P. For the single-input DPNNet-2.0 the images serve as the sole input and the planet mass is the targeted output. However, for the multi-input DPNNet-2.0, we assign each simulation a set of feature variables (α, h_0, σ, S_t, ϵ) and the corresponding image as the input. The target variable is the planet mass M_P. The data for each feature variable is normalized using standard (z-score) scaling by subtracting the mean and scaling it by the standard deviation.

3.5. DPNNet-2.0 Training

The CNN and the MLP architectures are implemented using the deep-learning API, Keras, which acts as an interface for TensorFlow (Abadi et al. 2015). For the purpose of training and testing the model we split the data set into two blocks, a training set consisting of 85% and a testing set consisting of 15% of the simulation data. The training set is further split to keep 15% of the data aside for model validation. There are two separate training processes for the two modules of the DPNNet-2.0. For the single-input CNN-based module only the preprocessed simulated images from the training set are fed into the network as the input. The multi-input module accepts

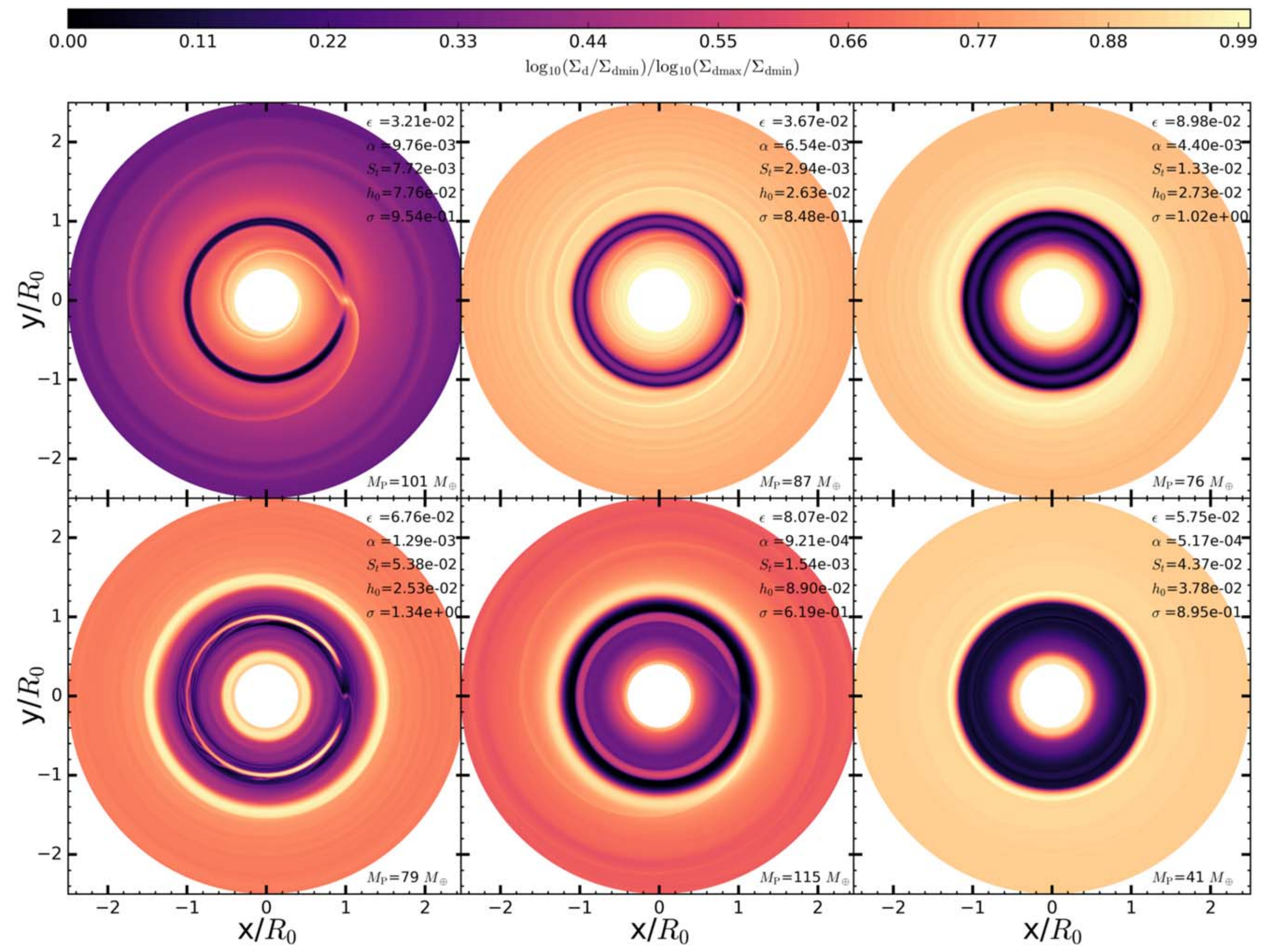

Figure 3. The normalized dust surface-density distribution after 3×10^3 orbits for different planet masses and different disk initial conditions indicated on the top right corner of each image. The top row shows systems with a single gap with increasing width from left to right. The first two panels on the bottom row have two gaps adjacent to a planet, and the rightmost plot in the bottom row has a deep cavity. The mass of the embedded planet is indicated in each panel.

both the preprocessed simulated images and the corresponding normalized feature variables (α, h_0, σ, S_t, ϵ). In both the cases the target variable, planet mass M_P, is the same. The resolution of the input images are fixed to 512×512.

The training process works iteratively where the weights are adjusted after each step until they are optimized. For each iteration a subset of the training data set is fed into the network (batch size = 20) to minimize the computational cost during the optimization process. We use an adaptive learning rate 10^{-5} with a small decay of $1/2000$. The single- and multi-input modules are trained independently for several epochs while monitoring the validation loss, mean square error (MSE), after each iteration. We implement early stopping and use the callback function to stop the training when the model is stable and reaches optimal performance (minimum MSE). This is done to avoid overfitting or underfitting of the model. A small number of epochs would result in an underfitted model whereas a large number of epochs might lead to overfitting. Figure 7 shows the behavior of the loss function (in terms of MSE) as a function of epochs during the training process until it stabilizes for both the modules. Note that the training process is almost independent of the batch size due to the use of ResNet architecture, which is shown to be insensitive to the specific choice of batch size (Lathuilière et al. 2018). The training is accomplished with GPU machines RTX 8000 and TESLA P100 available in the supercomputing clusters at ASIAA. For the initial code development and testing we used the M1-ARM-based chip from Apple and the nonpaid version of Google Colab. However, both Google Colab and M1 had limited GPU memory and could only be used for low-resolution images. The performance benchmarking of the different hardware against the use of various CNN architecture is shown in the Appendix.

4. Results

Once the training process is complete, we have two trained modules of DPNNet-2.0 that can be deployed to predict planet masses from disk images. The single-input module accepts simulated images as input, while the multi-input one additionally considers the feature variables. We deploy both the modules on the test data set to quantify the network's performance when applied to unseen data. We compare the network's accuracy in terms of the $r2$ score as well as the rms error (RMSE) for the single- and the multi-input modules.

The test data set consists of 429 randomly chosen sample simulated images along with the corresponding feature variables and the true values of the planet mass (as used in the underlying hydrodynamic simulation). These images are fed into the single-input DPNNet-2.0 to predict the planet mass. Figure 4(A) illustrates the correlation between the predicted planet mass (in Earth-mass units, $M_\oplus$) and the actual values of

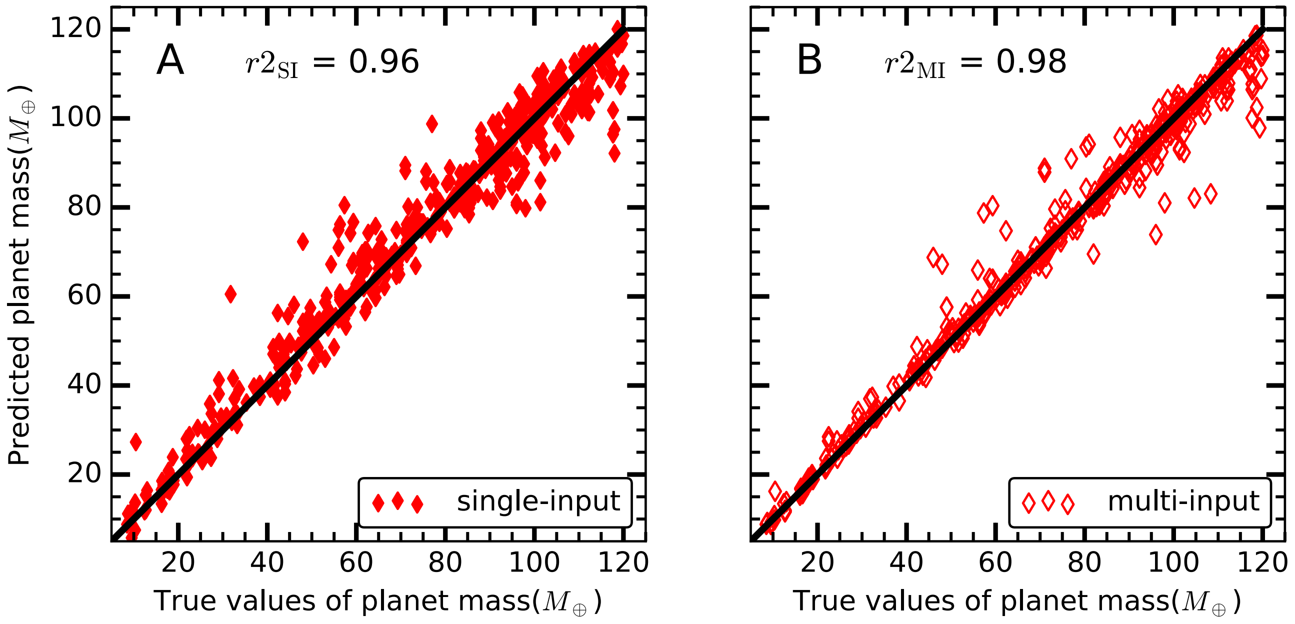

Figure 4. Correlation between the simulated planet mass and the predicted planet mass (in units of $M_\oplus$) obtained from the single-input (only images) and multi-input (images and disk parameters) DPNNet-2.0 model. The $r2$ scores indicate the goodness of fit.

the planet mass used in the simulations. The predicted planet masses mostly lie along the black line indicating a strong correlation with an $r2$ score of $r2_{\mathrm{SI}} = 0.96$. We further estimate the RMSE and the mean absolute error (MAE) to quantify the model's performance. DPNNet-2.0 with a single input has an RMSE of 6.2 $M_\oplus$ and an MAE of 4.3$M_\oplus$ when applied to the test data set.

On implementing the multi-input module we have an improved model performance. Figure 4(B) shows the correlation between the predicted and the simulated planet mass. The correlation is much tighter with $r2_{\mathrm{MI}} = 0.98$, which is an improvement compared to the single-input module. The RMSE and the MAE are 4.6$M_\oplus$ and 2.5$M_\oplus$, respectively, which are comparatively lower.

To further reduce statistical fluctuations due to data sampling, we apply k-fold cross validation, which is a resampling process, using the Python SCIKIT-LEARN library. The data is shuffled randomly and divided into $k = 7$ folds or groups of equal size. A single fold is used for testing while the remaining $k - 1$ folds are used to train the model. The process is repeated k number of times, each time selecting a different group to test and the remaining groups to train. The mean of the RMSE values obtained from each iteration is used as a measure of the model performance. Figure 8 shows the variation of the mean RMSE (left panel) and the mean $r2$ score (right panel) as function of image resolution for both the single- and multi-input DPNNet-2.0. At the maximum image resolution of 512 × 512 we measure a mean RMSE of (7.5$M_\oplus$, 4.9$M_\oplus$) and mean $r2$ score of (0.94, 0.97) for the single- and multi-input models, respectively. These values can be considered as the prediction uncertainty of DPNNet-2.0. Thus the use of image-based characterization enables DPNNet-2.0 to perform significantly better compared to the MLP-based model (DPNNet) in Paper I, which reported an RMSE of 12.5$M_\oplus$ and an MAE of 7.9$M_\oplus$ when applied to the test data set.

This is further illustrated in Figure 5, where we plot the prediction error $M_{\mathrm{P,simulated}} - M_{\mathrm{P,predicted}}$ for each of the models. The error follows a normal distribution. The mean for DPNNet-2.0 ($|\mu|_{\mathrm{SI}} = 0.8$, $|\mu|_{\mathrm{MI}} = 0.3$) is more centered toward zero and the standard deviation ($\sigma_{\mathrm{SI}} = 6.2$, $\sigma_{\mathrm{MI}} = 4.6$) is also lower compared to DPNNet ($|\mu| = 1.2$, $\sigma = 13.3$), indicating a tighter constraint for the predicted planet mass. Thus image-based characterization of disk images using CNN significantly brings down the prediction uncertainly/error. DPNNet-2.0 outperforms its predecessor DPNNet in term of both predictive capability as well as the flexibility to accept images as a direct input.

5. Discussion and Conclusion

In this paper we introduce part I of the CNN-based DPNNet-2.0 architecture to predict the mass of unseen exoplanets from disk images. Currently this model is trained and tested on images generated from hydrodynamic simulations of disk–planet interaction. We demonstrate the high accuracy and the efficiency of the network to predict the planet mass from the simulated images. This provides the required framework for extracting the mass of exoplanets from observed images, which is a work in progress, where we need to train the model with synthetic images (DPNNet-2.0 part II; S. Auddy et al. 2021, in preparation). In part I we primarily focus on building the CNN-based architecture and demonstrate its functionality based on simulated images.

In Auddy & Lin (2020) we developed DPNNet, an MLP-based model that accepts a set of disk parameters and predicts the planet mass. This required measuring the gap width explicitly from each of the observed images, which is often prone to additional uncertainties (see Section 1.1). In this work we exploit the power of computer vision, implementing a CNN architecture, to learn in a more direct way the underlying relationship between the features observed in the simulated images of PPDs hosting an embedded planet. A CNN architecture can extract features from a given image using the various convolutional filters present within the network. This makes it possible to map the connection between the various disk structures, most prominent being the dust gap, and the complex disk–planet interaction.

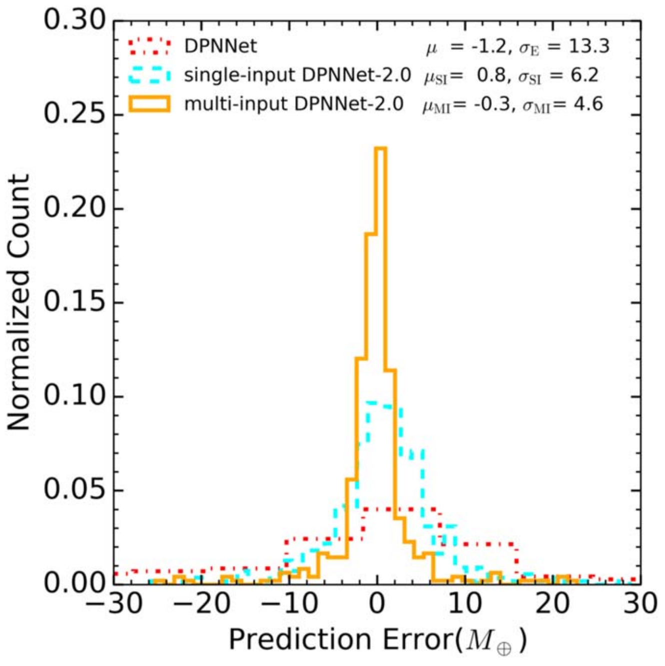

Figure 5. Distribution of the prediction error $M_{P,\text{simulated}} - M_{P,\text{predicted}}$ for DPNNet, single-input, and multi-input DPNNet-2.0.

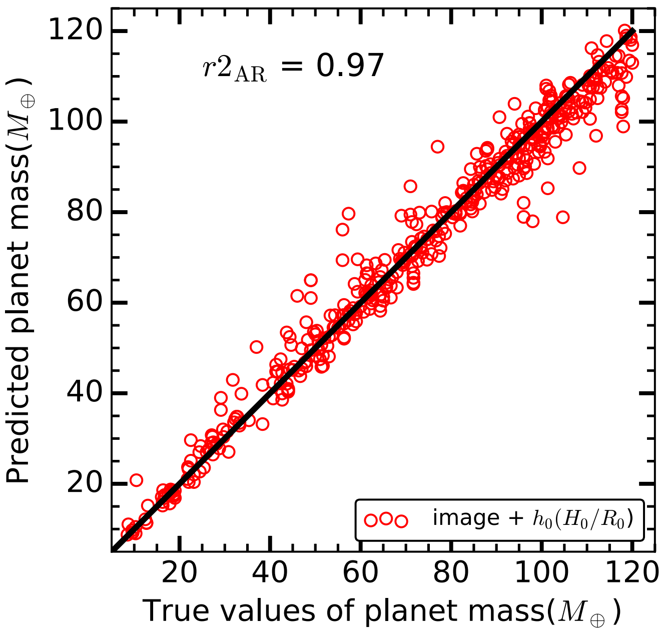

Figure 6. Correlation between the actual and the predicted mass when using a simulated image as well as the disk aspect ratio h_0 as inputs.

Equipped with a CNN at its core, DPNNet-2.0 predicts the mass of the unseen planet from the disk morphology. It extracts the features directly from images, thereby minimizing the errors associated with different user-defined measurement probes. Compared to the results of Paper I having a mean RMSE of $\pm 12.5 M_\oplus$, the single-input DPNNet-2.0 yields a much tighter constraint to the predicted planet mass with a reduced mean RMSE of $\pm 7.4 M_\oplus$. This can also be attributed to the fact that in addition to considering the visible gap width, DPNNet-2.0 takes in account other morphological characteristics (like gap depth and asymmetries such as vortices and spirals) which are the imprint of disk–planet interaction as observed in simulated disk images. This makes DPNNet-2.0 much more effective in constraining the properties of unseen planets, like planet mass.

The multi-input module in DPNNet-2.0 is designed to accept disk features in addition to dust surface-density-simulated images. It inherits the MLP module from DPNNet (Paper I), which gives it the flexibility to accept disk features, like α, h_0, σ, S_t, and ϵ along with images. This allows the network to further reduce the uncertainty in the predicted planet mass. It yields a mean RMSE of $\pm 4.9 M_\oplus$, which is an improvement over the single-input model. Thereby, multi-input DPNNet-2.0 retains the beneficial part of DPNNet while improving its capacity to better extract features from the simulated images.

However, many of the disk parameters that are required for the multi-input DPNNet-2.0 are often not well constrained from observation. One usually considers canonical values when precise measurement is not possible. For example, it is common to consider a disk to be weakly turbulent, $\alpha \approx 10^{-3}$, with a particle abundance (dust-to-gas ratio) of $\sim 1\%$, comprised of millimeter-sized dust grains. On the other hand, disk temperatures can be constrained with more confidence (Y.-W. Tang 2021, private communication), which is used to estimate the disk aspect ratio h_0. Thus, as a test we train an additional multi-input model where we drop all the feature variables except the disk aspect ratio h_0. Once trained we test the network on the same test data, but this time the network only accepts the simulated disk image and the corresponding h_0. Figure 6 gives the correlation between the predicted planet mass and the simulated values. The $r2$ score is $r2_{SI} < r2_{AR} < r2_{MI}$ implying an increase in accuracy compared to single-input models but a decrease compared to multi-input models. This is indicative of the fact that with the addition of more disk parameters as inputs, along with a simulated image, the prediction uncertainty gets reduced.

In the next section we will highlight the limitations associated with limited training data and CNN architecture, and discuss the scope for improvement in the future.

6. Network Limitations and Future Scope

The key to an effective well-trained model is a generous data set that includes a broad parameter space and captures a wide diversity of disk morphology induced by disk–planet interaction. A trained network is only as good as its training data set, which ideally must encompass all possible scenarios. However, it is not only challenging to design simulations which do that, but it is equally expensive to generate the data. In this paper we only use simulated images and explore a narrow parameter space due to finite computational resources.

The ultimate objective of DPNNet-2.0 is to predict planet mass from observed images. However, due to the limited availability of observed PPD images the model cannot be solely trained with observed data. Thus, DPNNet-2.0 is trained and tested on simulated images to verify the capability of the framework. Part I of the DPNNet-2.0 model, this paper, mainly focuses on the model build and the network architecture. Though in its current state this model has its limitations, this novel approach opens a unique direction of using computer vision for parameter estimation (for, e.g., planet mass) from observed disk images. Part II of this work will focus on developing more realistic training data sets (e.g., synthetic images, wider parameter space) to improve the robustness and applicability of the model on observed images.

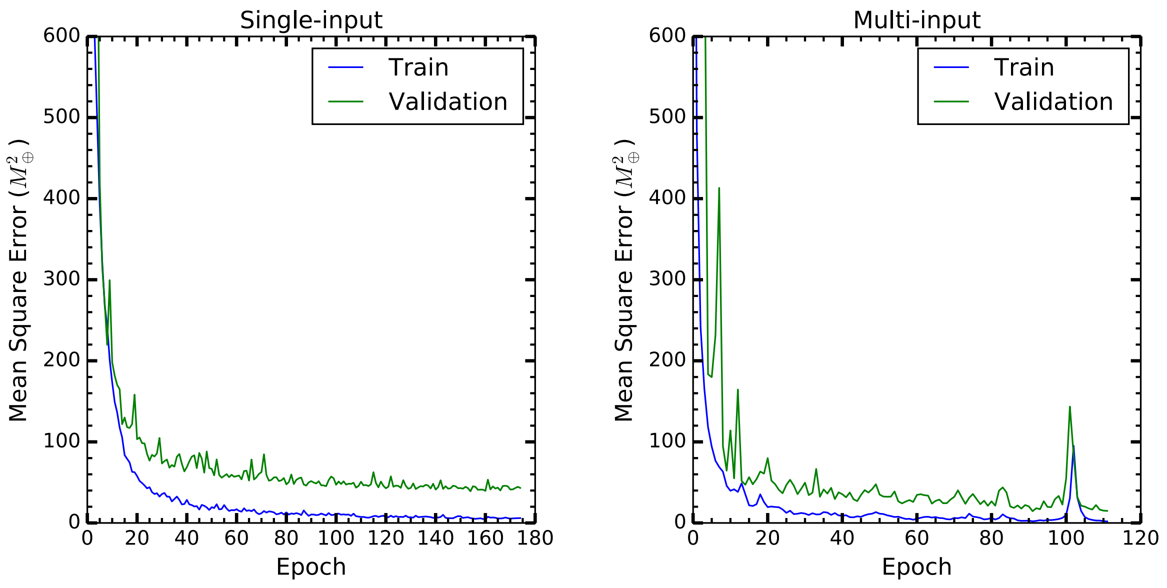

Figure 7. The training and validation loss is shown as a function of training epoch for the single- as well as the multi-input DPNNet-2.0 model.

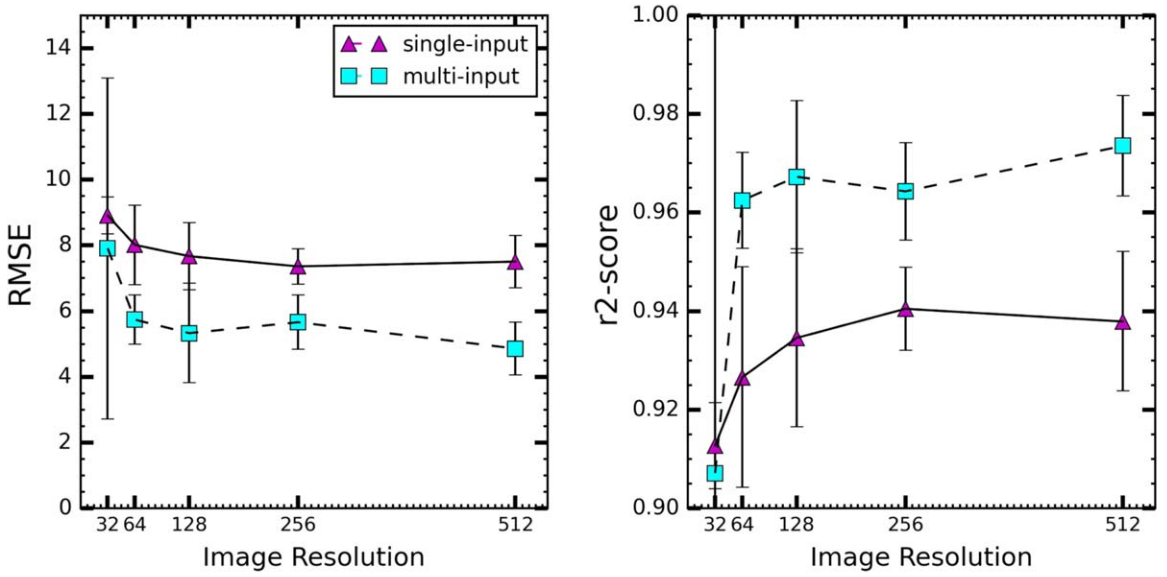

Figure 8. Left: the variation of the mean value of RMSE with the increasing resolution of images obtained from *k*-fold cross validation. Right: The corresponding mean value of the *r*2 score indicating the goodness of fit between the predicted and the simulated planet mass. The error bar indicates the standard deviation incurred due to sampling.

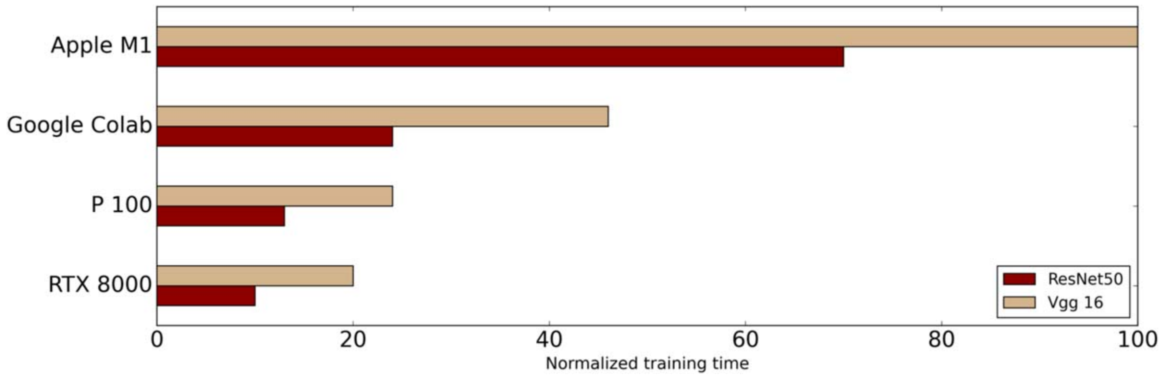

Figure 9. Comparison of various GPU hardware in terms of training time for the ResNet50/Vgg-16 architecture used in the DPNNet-2.0 model.

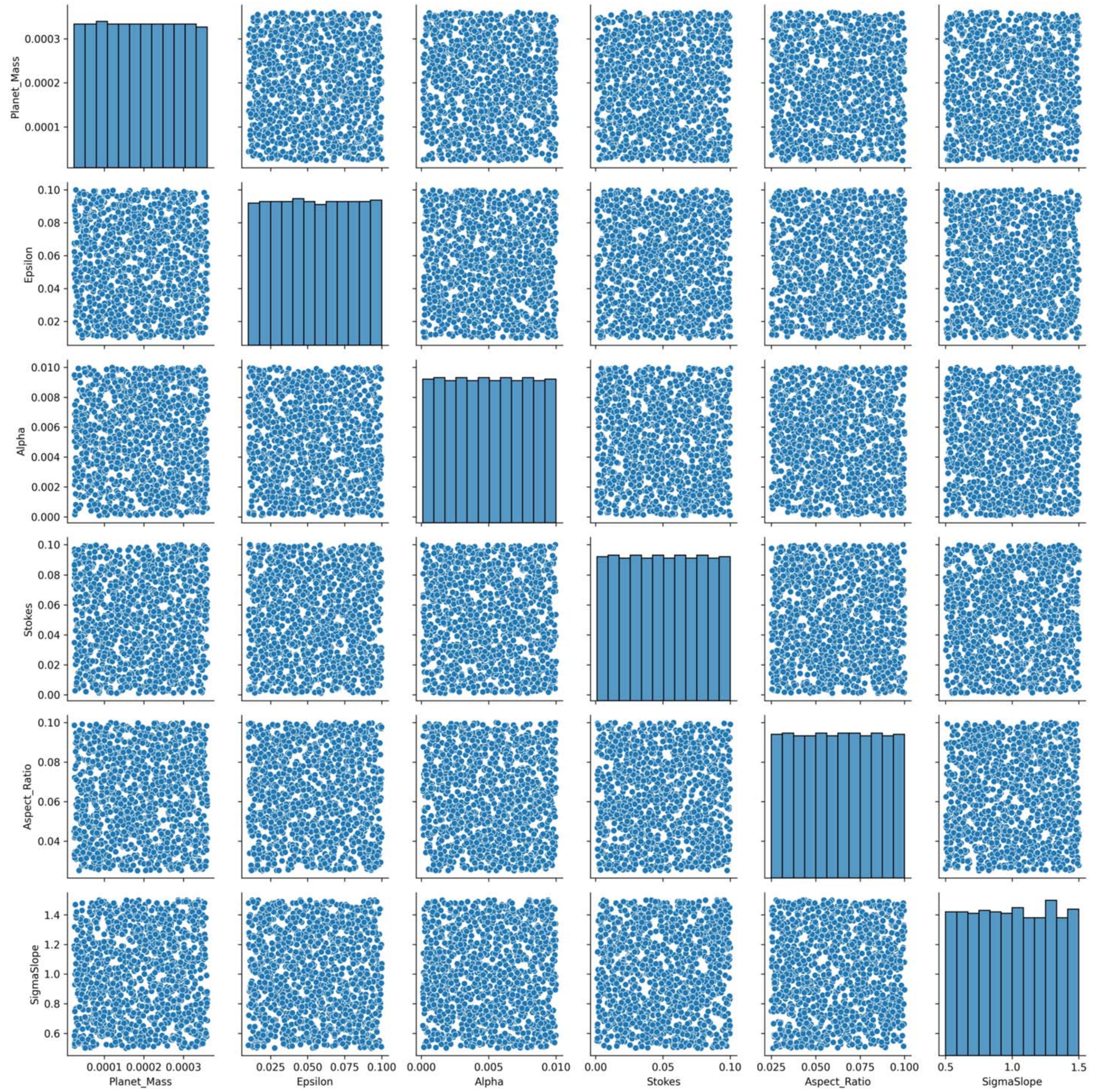

Figure 10. Pair plot representing the distribution of the input parameters (initial condition) sampled using LHS. The correlation between each quantity is demonstrated in each panel. The histograms of the different quantities are also represented in the diagonal. The LHS was implemented using the pyDOE package to generate a uniform random distribution of the parameters. Each parameter value was centered within the sampling intervals.

In DPNNet-2.0 part II (S. Auddy et al. 2021, in preparation) we use synthetic observations as the training data set. This requires 3D disk models obtained by running 3D hydrodynamic simulations or by "puffing up" 2D disk models. In either case, we follow up with radiative transfer calculations using a variety of dust opacities and protostellar properties, and put them through a synthetic image generator to mimic interferometric observations (Dong et al. 2015). Once the data set is generated we will simply retrain the DPNNet-2.0 model with these synthetic observations without changing the overall architecture of the network.

For deploying DPNNet-2.0 to observations, one should also account for additional factors such as the viewing angle, dust opacity, observing wavelength, multiple gaps, and stellar properties, to name a few. These dependencies will generally increase the training data set. However, some of these issues can be addressed without any change to the DPNNet-2.0 architecture. For example, although DPNNet-2.0 is trained on face-on images, in principle it can be applied to observations at other viewing angles by deprojecting it. It is, in fact, a standard procedure to deproject observed ALMA disk images (for example, see Huang et al. 2018a, 2018c; Pinte et al. 2020) with varying viewing angles (i.e., inclined and/or rotated with respect to the observer) to obtain face-on images. One can directly feed these deprojected images to DPNNet-2.0 as an input without altering the training process. In DPNNet-2.0 part II and future follow-ups we will address these issues in more detail, overcoming the limitations of computational resources, and work toward improving the efficiency of the model.

We thank the anonymous referee for the constructive comments and feedback. 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-, 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 (NCHC). This study was supported in part by resources and technical expertise from the Georgia Advanced Computing Resource Center, a partnership between the University of Georgia's Office of the Vice President for Research and Office of the Vice President for Information Technology.

Code Availability

The codes used for developing DPNNet-2.0 are available on the GitHub software repository at https://github.com/sauddy/DPNNet-2.0. The codes for the DPNNet in Auddy & Lin (2020) are also available at https://github.com/sauddy/DPNNet.

Appendix

Figure 7 shows the training and the validation loss (MSE) as a function of training epoch for the single- and multi-input models, respectively. In Figure 8 we show the variation of the mean RMSE and the $r2$ score for different image resolutions. Figure 9 gives the relative computation cost for both ResNet50 and Vgg-16 using different GPU hardware. Figure 10 is a pair plot representing the distribution of the input parameters sampled using LHS.

ORCID iDs

Sayantan Auddy https://orcid.org/0000-0003-3784-8913
Ramit Dey https://orcid.org/0000-0002-0786-7307
Min-Kai Lin https://orcid.org/0000-0002-8597-4386
Cassandra Hall https://orcid.org/0000-0002-8138-0425

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/
ALMA Partnership, Brogan, C. L., Pérez, L. M., et al. 2015, ApJL, 808, L3
Andrews, S. M., Huang, J., Pérez, L. M., et al. 2018, ApJL, 869, L41
Andrews, S. M., Wilner, D. J., Zhu, Z., et al. 2016, ApJL, 820, L40
Auddy, S., & Lin, M.-K. 2020, ApJ, 900, 62
Barge, P., Ricci, L., Carilli, C. L., & Previn-Ratnasingam, R. 2017, A&A, 605, A122
Batalha, N. M., Rowe, J. F., Bryson, S. T., et al. 2013, ApJS, 204, 24
Belagiannis, V., Rupprecht, C., Carneiro, G., & Navab, N. 2015, in Proc. IEEE Int. Conf. on Computer Vision (Piscataway, NJ: IEEE), 2830
Benítez-Llambay, P., Krapp, L., & Pessah, M. E. 2019, ApJS, 241, 25
Butler, R. P., Vogt, S. S., Laughlin, G., et al. 2017, AJ, 153, 208
Cassan, A., Kubas, D., Beaulieu, J. P., et al. 2012, Natur, 481, 167
Clarke, C. J., Tazzari, M., Juhasz, A., et al. 2018, ApJL, 866, L6
Crida, A., Morbidelli, A., & Masset, F. 2006, Icar, 181, 587
Cugno, G., Quanz, S. P., Hunziker, S., et al. 2019, A&A, 622, A156
de Val-Borro, M., Artymowicz, P., D'Angelo, G., & Peplinski, A. 2007, A&A, 471, 1043
de Val-Borro, M. E. 2006, MNRAS, 370, 529
Deeg, H. J., & Alonso, R. 2018, in Handbook of Exoplanets, ed. H. J. Deeg & J. A. Belmonte (Cham: Springer), 117
Dipierro, G., Laibe, G., Price, D. J., & Lodato, G. 2016, MNRAS, 459, L1
Dipierro, G., Ricci, L., Pérez, L., et al. 2018, MNRAS, 475, 5296
Dong, R., & Fung, J. 2017, ApJ, 835, 146
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
Fischer, D. A., Howard, A. W., Laughlin, G. P., et al. 2014, in Protostars and Planets VI, ed. H. Beuther et al. (Tucson, AZ: Univ. Arizona Press), 715
Fukushima, K., & Miyake, S. 1982, Competition and Cooperation in Neural Nets (Berlin: Springer), 267
Fulton, B. J., Petigura, E. A., Howard, A. W., et al. 2017, AJ, 154, 109
Fung, J., Shi, J.-M., & Chiang, E. 2014, ApJ, 782, 88
Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2014, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 580
Goldreich, P., & Tremaine, S. 1980, ApJ, 241, 425
Gonzalez, J. F., Laibe, G., Maddison, S. T., Pinte, C., & Ménard, F. 2015, MNRAS, 454, L36
Haffert, S. Y., Bohn, A. J., de Boer, J., et al. 2019, NatAs, 3, 749
Hammer, M., Kratter, K. M., & Lin, M.-K. 2016, MNRAS, 466, 3533
Hammer, M., Pinilla, P., Kratter, K. M., & Lin, M.-K. 2018, MNRAS, 482, 3609
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 770
Hojjatpanah, S., Figueira, P., Santos, N. C., et al. 2019, A&A, 629, A80
Huang, J., Andrews, S. M., Cleeves, L. I., et al. 2018a, ApJ, 852, 122
Huang, J., Andrews, S. M., Dullemond, C. P., et al. 2018b, ApJL, 869, L42
Huang, J., Andrews, S. M., Pérez, L. M., et al. 2018c, ApJL, 869, L43
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
Johansen, A., Youdin, A., & Klahr, H. 2009, ApJ, 697, 1269
Kanagawa, K. D., Muto, T., Tanaka, H., et al. 2015, ApJL, 806, L15
Kanagawa, K. D., Muto, T., Tanaka, H., et al. 2016, PASJ, 68, 43
Keppler, M., Benisty, M., Müller, A., et al. 2018, A&A, 617, A44
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2017, Commun. ACM, 60, 84
Lathuilière, S., Mesejo, P., Alameda-Pineda, X., & Horaud, R. 2018, in IEEE PAMI (Piscataway, NJ: IEEE), 2065
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proc. IEEE, 86, 2278
Lee, C.-H. 2018, Galax, 6, 51
Lin, D. N. C., & Papaloizou, J. 1986, ApJ, 309, 846
Lin, D. N. C., & Papaloizou, J. C. B. 1993, in Protostars and Planets III, ed. E. H. Levy & J. I. Lunine (Tucson, AZ: Univ. Arizona Press), 749
Liu, X., Liang, W., Wang, Y., Li, S., & Pei, M. 2016, in 2016 IEEE Int. Conf. on Image Processing (ICIP), IEEE (Piscataway, NJ: IEEE), 1289
Liu, Y., Dipierro, G., Ragusa, E., et al. 2019, A&A, 622, A75
Lodato, G., Dipierro, G., Ragusa, E., et al. 2019, MNRAS, 486, 453
Long, F., Pinilla, P., Herczeg, G. J., et al. 2018, ApJ, 869, 17
Long, F., Pinilla, P., Herczeg, G. J., et al. 2020, ApJ, 898, 36
McKay, M. D., Beckman, R. J., & Conover, W. J. 1979, Technometrics, 21, 239
Meru, F., Rosotti, G. P., Booth, R. A., Nazari, P., & Clarke, C. J. 2019, MNRAS, 482, 3678
Miranda, R., & Rafikov, R. R. 2019, ApJL, 878, L9
Miranda, R., & Rafikov, R. R. 2020, ApJ, 892, 65
Momose, M., Morita, A., Fukagawa, M., et al. 2015, PASJ, 67, 83
Paardekooper, S. J., Baruteau, C., Crida, A., & Kley, W. 2010, MNRAS, 401, 1950
Paardekooper, S.-J., & Papaloizou, J. C. B. 2009, MNRAS, 394, 2297
Pérez, L. M., Benisty, M., Andrews, S. M., et al. 2018, ApJL, 869, L50
Pinte, C., Price, D. J., Ménard, F., et al. 2018, ApJL, 860, L13
Pinte, C., Price, D. J., Ménard, F., et al. 2020, ApJL, 890, L9
Pinte, C., van der Plas, G., Ménard, F., et al. 2019, NatAs, 3, 1109
Rawat, W., & Wang, Z. 2017, Neural Comput., 29, 2352
Raymond, S. N., & Morbidelli, A. 2020, arXiv:2002.05756
Rosotti, G. P., Juhasz, A., Booth, R. A., & Clarke, C. J. 2016, MNRAS, 459, 2790
Shakura, N. I., & Sunyaev, R. A. 1973, A&A, 24, 337

Simon, J. B., & Armitage, P. J. 2014, ApJ, 784, 15
Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556
Suriano, S. S., Li, Z.-Y., Krasnopolsky, R., & Shang, H. 2018, MNRAS, 477, 1239
Takahashi, S. Z., & Inutsuka, S.-i. 2014, ApJ, 794, 55
Teague, R., Bae, J., Bergin, E. A., Birnstiel, T., & Foreman-Mackey, D. 2018, ApJL, 860, L12
Toshev, A., & Szegedy, C. 2014, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 1653
van der Marel, N., Dong, R., di Francesco, J., Williams, J. P., & Tobin, J. 2019, ApJ, 872, 112
van der Plas, G., Wright, C. M., Ménard, F., et al. 2017, A&A, 597, A32
Wafflard-Fernandez, G., & Baruteau, C. 2020, MNRAS, 493, 5892
Wagner, K., Follete, K. B., Close, L. M., et al. 2018, ApJL, 863, L8
Weidenschilling, S. J. 1977, MNRAS, 180, 57
Winn, J. N., & Fabrycky, D. C. 2015, ARA&A, 53, 409
Youdin, A. N. 2011, ApJ, 731, 99
Zhang, S., Zhu, Z., Huang, J., et al. 2018, ApJL, 869, L47
Zhu, Z., Dong, R., Stone, J. M., & Rafikov, R. R. 2015, ApJ, 813, 88
Zurlo, A., Cugno, G., Montesinos, M., et al. 2020, A&A, 633, A119