



HAL
open science

A Machine Learning Framework to Predict Images of Edge-on Protoplanetary Disks

Zoie Telkamp, Jorge Martínez-Palomera, Gaspard Duchêne, Aishabibi Ashimbekova, Edward Wolfe, Isabel Angelo, Christophe Pinte

► **To cite this version:**

Zoie Telkamp, Jorge Martínez-Palomera, Gaspard Duchêne, Aishabibi Ashimbekova, Edward Wolfe, et al.. A Machine Learning Framework to Predict Images of Edge-on Protoplanetary Disks. *The Astrophysical Journal*, 2022, 939, 10.3847/1538-4357/ac96f1 . insu-03860260

HAL Id: insu-03860260

<https://insu.hal.science/insu-03860260>

Submitted on 19 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



A Machine Learning Framework to Predict Images of Edge-on Protoplanetary Disks

Zoie Telkamp^{1,2} , Jorge Martínez-Palomera^{3,4} , Gaspard Duchêne^{2,5} , Aishabibi Ashimbekova², Edward Wolfe²,
Isabel Angelo^{2,6} , and Christophe Pinte^{5,7} 

¹Dept. of Astronomy, University of Virginia, Charlottesville, Virginia 22904, USA

²Astronomy Department, University of California, Berkeley, CA 94720, USA

³Bay Area Environmental Research Institute, P.O. Box 25, Moffett Field, CA 94035, USA

⁴NASA Ames Research Center, Moffett Field, CA, USA

⁵Univ. Grenoble Alpes, CNRS, IPAG, F-38000 Grenoble, France

⁶Department of Physics & Astronomy, University of California Los Angeles, Los Angeles, CA 90095, USA

⁷School of Physics and Astronomy, Monash University, Clayton, Vic 3800, Australia

Received 2022 May 29; revised 2022 September 18; accepted 2022 September 29; published 2022 November 8

Abstract

The physical structure and properties of protoplanetary disks are typically derived from spatially resolved disk images. Edge-on disks in particular provide an important view point on the vertical structure and degree of settling of disks. Such analyses rely on radiative transfer (RT) calculations that are generally computationally intensive due to the high optical depth of disks. Here we present a machine learning framework that has the potential to dramatically speed up the forward modeling process by approximating the results of RT calculations. This framework, trained on an initial set of RT calculations, utilizes an autoencoder neural network to enable the generation of synthetic scattered light images of edge-on disks directly from a set of physical parameters. We demonstrate that this framework generates synthetic images 2–3 orders of magnitude faster than using RT calculations. These machine learning-generated images appear to approximate the RT images well, in particular preserving their size and shape. We also find a strong correlation between the latent space representations of the generated disk images and several of their associated physical parameters. Finally, we discuss potential changes to the framework, such as methods to further improve the image quality, extending the framework to multiple wavelengths, and inverting the process to infer physical parameters from observed images. Overall, these new tools have the potential to enable a more efficient and uniform analysis of edge-on disk properties and the initial conditions of planet formation.

Unified Astronomy Thesaurus concepts: [Pre-main sequence stars \(1290\)](#); [Radiative transfer \(1335\)](#)

1. Introduction

Understanding the diversity of planetary systems is a key effort in modern astronomy. The heterogeneity of masses, eccentricities, and orbital periods present within these populations of planets is thought to stem from the different mechanisms from which they form, and thus from the initial conditions of their birth sites (Winn & Fabrycky 2015, and references therein). Characterizing the structure and evolution of the protoplanetary disks (PPDs) in which these planets form is a key step toward understanding the vast variety of systems that have been observed in the last two decades. Spatially resolved imaging of disks over a range of wavelengths has become commonplace (e.g., Benisty et al. 2022; Miotello et al. 2022), providing the necessary data sets to achieve these goals.

A particularly informative case arises when these disks are viewed edge-on. For the purpose of this study, we classify PPDs as edge-on disks (EODs) when they are inclined highly enough to occult the central starlight. In this configuration, a disk is viewed as two parallel nebulae where the light from the star scatters off of the outer surfaces of the disk, separated by a dark midplane (Whitney & Hartmann 1992). This geometry naturally exposes the vertical structure of the disk. Using multiwavelength observations, quantities such as the scale height and degree of settling of the dust component of the disk

can be inferred, whereas these properties typically cannot be constrained for lower inclination disks (Watson et al. 2007). Over three dozen EODs are currently known, many of which have been observed at multiple wavelengths from the optical to the millimeter (e.g., Burrows et al. 1996; Stapelfeldt et al. 2003; Villenave et al. 2020).

The physical parameters of these disks are typically derived through forward modeling processes, where synthetic images are generated through radiative transfer (RT) calculations and fit to the observations (e.g., Duchene et al. 2010). However, the high computational cost of these RT calculations provides a severe limitation on this analysis. Because they are optically thick at most wavelengths, modeling PPDs requires proper treatment of multiple scattering during the RT calculations (Bastien & Menard 1988; Pinte et al. 2006). A further challenge is imposed by the complexity of the underlying physics of these objects. In order to avoid oversimplification of the models, PPDs are typically described using a high number of free parameters, which adds to the computational cost. All of these factors contribute to the global computing time problem for model fitting in this context. This becomes an issue when high volumes of synthetic images must be generated, such as during a Markov Chain Monte Carlo (MCMC) approach. As a result, only a few individual studies have been conducted using rigorous MCMC treatment to constrain EOD parameters (e.g., Wolff et al. 2017). While these can inform our perspectives of individual disks, a large-scale statistical approach is needed to properly assess EOD demographics in relation to planet-forming scenarios.



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

In order to enable such large-scale analyses, a more efficient means of modeling EODs is required. In this study, we explore the use of generative modeling with machine learning to achieve this goal. The term *generative modeling* describes the process of identifying the joint distribution of lower-dimensional variables—which define a so-called *latent space*—describing the data of interest. This latent space is an optimized decomposition of the input data into a minimum number of quantities that contain most of the information in the image. Sampling from this distribution enables the generation of new instances that realistically could have come from the same data set (Goodfellow et al. 2016). Deep generative models (DGMs) use neural networks to learn this distribution and create new samples. The main advantage of using neural networks for the purpose of this study is the computing time. DGMs such as autoencoders (AEs; Goodfellow et al. 2016), variational AEs (Kingma & Welling 2014), and generative adversarial networks (Goodfellow et al. 2014) have been used to efficiently generate realistic astronomical data. Examples include the use of generative adversarial networks by Dia et al. (2019) to generate realistic images of galaxies, image inpainting conducted by Yi et al. (2020) using a variational AE to restore missing data of microwave background maps, and the use of a generative adversarial network by Mustafa et al. (2019) to generate weak lensing convergence maps. Generative modeling has even been used in the context of PPDs. Ribas et al. (2020) used an artificial neural network to generate spectral energy distributions (SEDs) of non-edge-on PPDs to allow more efficient modeling of 23 disks. Convolutional neural networks have also been used to perform planet mass prediction directly on face-on images of PPD with gaps (Auddy et al. 2021; Zhang et al. 2022). Given this development, there is hope for significant gains in using DGMs to improve the efficiency and breadth of EOD image analysis.

Yet, a major limitation of using DGMs to generate new synthetic images derives from the fact that they do not connect predicted observables (disk image or SED) to physical parameters (such as disk mass, radius, or inclination). In other words, DGMs can produce plausible disk images but cannot predict the image corresponding to a specific set of disk properties. As a result, DGMs are generally not useful for interpretive purposes. Martínez-Palomera et al. (2020) demonstrated that this can be addressed by connecting the learned joint distribution of the latent variables of a generative model to the physical parameters that characterize the input data. This allows for the creation of data associated with specific values of the chosen physical parameters. Martínez-Palomera et al. (2020) explored the usage of a variational AE paired with a regressor to generate visually and physically realistic light curves for periodic variable stars. This concept can be extended to enable more efficient generation of 2D images directly from physical parameters, with the potential of drastically improving the computational cost of modeling processes typically dependent on RT calculations.

In this study, we demonstrate and assess the efficacy of using an AE neural network to enable generative modeling of Hubble Space Telescope (HST)-like scattered light images of EODs, building off of the machine learning framework described in Martínez-Palomera et al. (2020). We accomplish this by first training the AE to learn a compressed representation of our input data, as described in Section 2. Section 3 describes how we map the learned latent space representation of the data to the

physical parameters that characterize our EOD models using multiple regression techniques. We use the resulting framework to enable image generation directly from physical parameters without performing RT modeling, as demonstrated in Section 4. In Section 5, we discuss the successes and shortcomings of our current implementation, as well as the potential of using ML to efficiently model EODs in the future. We note that our choice of modeling visible light scattered light images is mostly a proof of concept in a challenging regime, and that the work conducted here could be generalized to other wavelengths and/or observing regimes.

2. Autoencoder Architecture and Training

2.1. Motivation

The machine learning algorithm we used is an AE deep neural network. AEs are trained to learn a condensed representation of the inputted observations in order to enable generative modeling of the data (Goodfellow et al. 2016). An AE consists of two parts, an encoder and a decoder. The encoder learns how to compress data into a lower-dimensional *latent space* representation, and the decoder then reconstructs the original data from this representation, prioritizing specific aspects of the data through this process. By nature, AEs do not provide perfect reconstructions of the data due to the fact that they discard some information in order to learn a condensed representation of the data. Nonetheless, this process represents the input data with a dramatically smaller number of dimensions while still retaining the key information that describes it.

The purpose of this study is to train an AE to learn a condensed representation of synthetic scattered light EOD images, translating the (187×187 pixel in our case) images into latent vectors of only a few values. We then aim to expand these results beyond the typical usage of AEs and connect the physical parameters that describe PPDs to the learned latent space. In order to accomplish this, we use a regressor to learn the relationship between the disk physical parameters and the latent space representation of its corresponding image. This allows us to input any combination of physical parameters, obtain the corresponding latent space representation, and use the decoder side of the AE to expand this representation into the predicted full-sized image. This method of going from physical parameters to disk image is much more efficient than computing the images through RT calculations, allowing us to generate an image corresponding to any combination of parameters within the range considered during the training phase in a fraction of the time.

We first tried a simple forward model (FM) as our machine learning framework, but found that this performed rather poorly (see Appendix), which led us to use an AE instead. In addition, the mirrored architecture of AE enables the possibility of inverting the generative modeling process. This would allow us to pass an image into our machine learning framework and receive the predicted physical parameters of the EOD, without the need for forward modeling. While we leave the inverse problem for a future work, using an AE leaves this option open. This choice was further motivated by the work of Martínez-Palomera et al. (2020), who demonstrated that a (variational) AE could be used in combination with a regression function to enable generative modeling of the light curves of variable stars given a set of physical parameters. The need for a variational AE in that work arose from the sparse nature of the observational

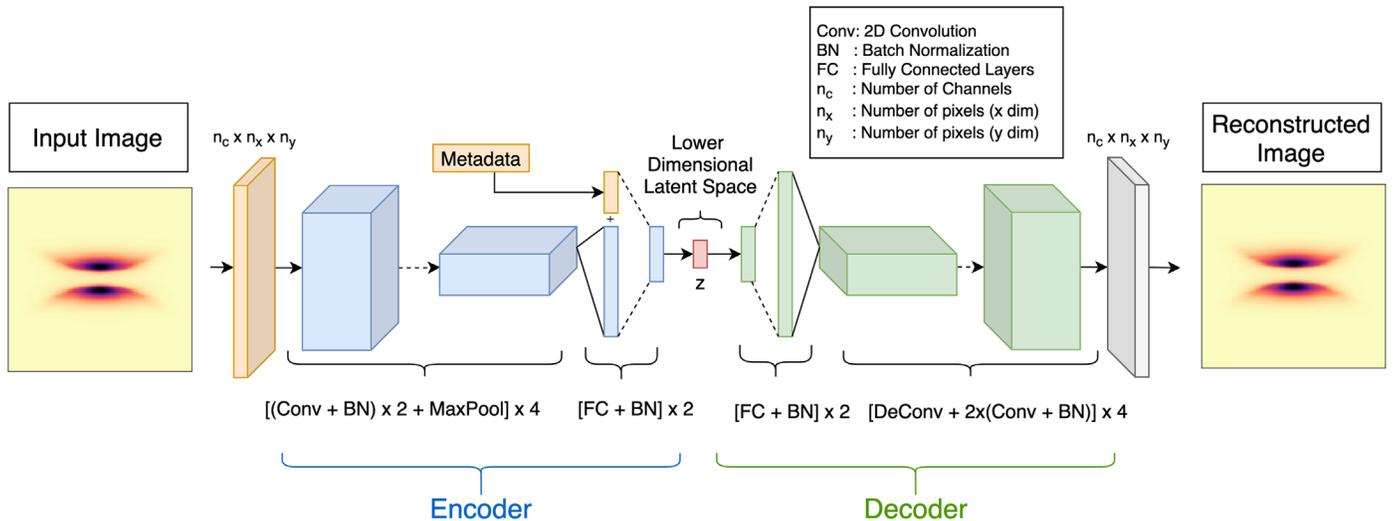


Figure 1. Adopted architecture for the autoencoder neural network. The autoencoder takes the inputted images, breaks them down into a lower-dimensional latent space through a series of layers, and reconstructs them through a symmetric process. Brackets denote a block unit, while each element inside is a layer. In this study, we adopt $n_c = 1$ (single wavelength image), but the architecture can be readily expanded to multiple images of a given disk. Metadata is the set of physical parameters that describes each disk.

data set, which made it difficult to smoothly interpolate between points in that space. In the present work, we are using synthetic data and can virtually fill up the entire range of physical parameters to generate training data, so that a standard AE is sufficient for our needs. Here, we build off of the AE architecture presented in Martínez-Palomera et al. (2020), extending it to a 2D problem and tailoring it to handle our own data.

2.2. Autoencoder Architecture

Our AE consists of two types of layers: convolutional layers that extract key features from the data, and fully connected layers that combine and compress these features into the lower-dimensional latent space (see Figure 1).

The convolutional layers follow the standard structure outlined in Goodfellow et al. (2016), beginning with several instances of convolution followed by an activation function (AF) and pooling operation. The convolutional layers operate by performing 2D convolutions of each image with a kernel, which is a 2D array of weights that are learned during the training process. Next, the outputs from the convolution stage pass through a rectified linear unit (ReLU) AF, i.e., a piecewise linear function that takes in the output from the previous layer and directly returns numbers greater or equal to 0, while outputting 0 for negative numbers. This acts as a nonlinear transformation of the data to allow for greater complexity in the learned model. These blocks are concluded by a max pooling layer that replaces a 2D data window of a given size with the maximum value of the whole window. Each block extracts higher-order features, and combining many of these blocks helps with covering different scales of features in the image (Goodfellow et al. 2016). The AE uses batch normalization within these layers, i.e., the absolute lowest and highest pixel values in each batch are set to 0 and 1, respectively, to standardize the input data and allow for faster training times and quicker convergence (Ioffe & Szegedy 2015). Following standard procedures, we create each convolutional block as follows: [(Conv + BN) \times 2 + MaxPool] (see Figure 1). We stack 4 of these blocks to reduce the image size by a factor of 45, taking into account the channels in the output, from (187 \times 187) pixels to 16 channels \times (7 \times 7) pixels.

Table 1
Values of Hyperparameters Tested in the Grid Search

Parameter	Grid Search
Batch size	[64, 128, 256, 512]
Learning rate	[10^{-5} , 10^{-4} , 5×10^{-4} , 10^{-3} , 5×10^{-3} , 10^{-2}]
Learning rate scheduler ^a	[Step, Exponential, Cosine, Plateau]
Latent space dimension	[4, 8, 12, 16, 24, 32, 48, 64]
Dropout probability ^b	[0.01, 0.1, 0.2, 0.3, 0.4, 0.5]

Notes. The parameters of the best-performing model are in bold.

^a Function that defines how the learning rate changes over training time (steps) to improve convergence.

^b Probability that the neurons in a given layer will be turned off to prevent over fitting (Srivastava et al. 2014).

Note that, since the original images are in a single filter, the number of input channels is 1.

The fully connected layers consist of linear combinations whose weights are also learned during the training process. These layers combine the features outputted by the convolutional layers and further compress them into the latent space. This part of the AE also includes batch normalization layers (BN) and ReLU activations.

Given this structure, the flow of data through the AE is as follows. A batch of images enters the encoder, where a stack of 4 convolutional layers (as defined above) reduces the dimensions of each image into a vector of hundreds of features. Next, we have a series of fully connected layers that combine the image features with metadata (which in our case are the physical parameters that describe the EODs; see Section 2.4 for more details) and compress the resulting vectors into the latent space, which acts as the bottleneck of the neural network. After testing the latent space dimensionalities shown in Table 1, we found that many yielded similar reconstruction loss values. We found a latent space dimensionality of 8 to be a good compromise between capturing relevant features to reconstruct the data and keeping a low dimensionality. This remained true in both the case where we embedded the metadata into the architecture and the case where we did not. This is also convenient because we

have 8 physical parameters, so choosing this dimensionality simplifies the regression process, especially if we want to invert it.

The latent space vectors then pass through the decoder of the neural network, which uses the same hyperparameters (e.g., number and type of layers) as the encoder. The decoder mirrors the architecture of the encoder, starting with linear layers and passing the results into a series of deconvolutional layers (see the decoder side in green of Figure 1). These layers use transpose convolutions (Zeiler et al. 2010) to expand the images back to their original shape, as well as convolutional layers to smooth out the high-frequency noise characteristic of deconvolved data. A sigmoid AF then rescales the data to values between 0 and 1 to match the range of the input images that were previously normalized to this range. Finally, the AE outputs the rescaled, reconstructed images.

2.3. Training Methodology

The inputs of the AE are synthetic EOD images that are described in detail in the next subsection; here we focus on the methodology of the AE training itself. Before training the AE, we randomly split our data set in separate training (80%) and test (20%) sets. Since we are focused on the morphology of the disks and not their absolute brightness, we only need to preserve the relative fluxes of the pixels. This allows us to perform image-by-image min–max normalization to speed up the training process and allow for quicker model convergence. In a single training epoch, batches of the training images enter the architecture, where the encoder compresses them into their latent space representation vectors. The decoder then expands these vectors into full-sized images. The AE compares the reconstructed images to their original counterparts using a loss function, i.e., a function that measures the degree of similarity between the two images (akin to a goodness-of-fit metric in standard model fitting problems). The model parameters are learned by minimizing the loss function via the backpropagation algorithm (Rumelhart et al. 1986). Each epoch uses the entire training set split up into batches. When a new epoch begins, the AE is trained with the same data but in a different order and with slight variations, such as vertical flipping and rotations to allow for a rotationally invariant model (Cabrera-Vives et al. 2017). We trained our neural network using 150 epochs to prevent underfitting and ensure convergence. At the end of each epoch, the loss function is evaluated for the test models as an independent check on the quality of the training.

For the loss function, we tested mean squared error (MSE) and binary cross entropy (BCE) functions. We tested MSE because it is a standard loss function for neural networks but found that it did not lead to quality image reconstructions. We were motivated to try BCE next because our EOD images contain a sharp contrast between bright and dim pixels, and this loss function rewards accurate relative pixel brightness levels. The BCE loss is calculated pixel by pixel using the following functional form:

$$\ell_n = -[y \log \hat{y} + (1 - y) \log (1 - \hat{y})],$$

where y is the pixel value of the input image, and \hat{y} is the pixel value predicted by the AE. We then take the average across all of the pixels in an image as our loss value.

After training the AE with this loss function, we found that BCE yielded visually convincing image reconstructions and plausible loss values at that point in the model development

process. As with any machine learning model, AEs require the user to fix certain *hyperparameters* that are external to the model and must be set before training. These hyperparameters include values such as the dimensionality of the latent space, the number of each type of layer in our architecture, and the size of the kernel used for convolution. In order to choose the hyperparameters of our model, we conducted a grid search while altering five hyperparameters between commonly used values (see Table 1), using the sweep method provided by Weights & Biases (Biewald 2020). In this sweep, we trained a model for each hyperparameter combination in the grid and recorded their training and test loss values. Once this was finished, we selected the model architecture with the lowest loss values. The loss values for the selected model are shown in Figure 2, where we can see that the loss decreases throughout the training process until it converges. The fact that the test loss value has converged is a sign that enough epochs have been completed.

We visually inspected the image reconstructions and the residual maps between the input and predicted images to make sure they were of sufficient quality. While we selected the model that performed the best in our sweep, this is not necessarily the optimal combination of hyperparameters. After training many AEs with different hyperparameter combinations and comparing their loss values, we can see that multiple combinations of hyperparameters provide similarly good results, meaning the results presented here are not overly sensitive to this selection.

2.4. Training Data Set

We computed the training and testing images using the RT modeling code MCFOST (Pinte et al. 2006, 2009). In order to account for the diversity in morphologies that we observe, we built our training data set around the grid of HST-like $0.6 \mu\text{m}$ scattered light PPD images detailed in Angelo et al. (in prep.). Specifically, the seven key physical parameters listed in Table 2 are varied, and for each model, scattered light images are computed using MCFOST at 15 inclinations ranging from 45° to 90° . The resulting images are convolved with an HST point-spread function (PSF) produced using the TinyTim package (Krist 1995) and resampled to $0''.04$ pixel scale. A total of 23,040 models at 15 inclinations each were computed as part of this grid.

The range of parameter values is informed by empirical distributions and designed to cover the parameter space sparsely but as broadly as possible (Angelo et al., in prep.). The models assume a tapered power law surface density profile (Williams & Cieza 2011),

$$\Sigma(R) = \Sigma_c (R/R_c)^\gamma e^{-(R/R_c)^{2+\gamma}},$$

where R_c is the critical radius of the disk, and γ is the surface density exponent, which defines how the dust is radially distributed at radii $R \ll R_c$. A vertically isothermal gas density profile, $\rho(z) \propto e^{-\frac{z}{2H(R)}^2}$, along with a flared structure, $H(R) = H_0 (R/R_0)^\beta$ where we arbitrarily set $R_0 = 100$ au, is adopted. Dust settling is implemented following the model of Fromang & Nelson (2009), which depends on the degree of turbulence in the disk measured by the α viscosity parameter and grain size.

Each model is evaluated as part of a single star system 140 pc away, with a radius of $2 R_\odot$ and a blackbody temperature of 4000 K. More detail about the choice of variable and fixed parameters

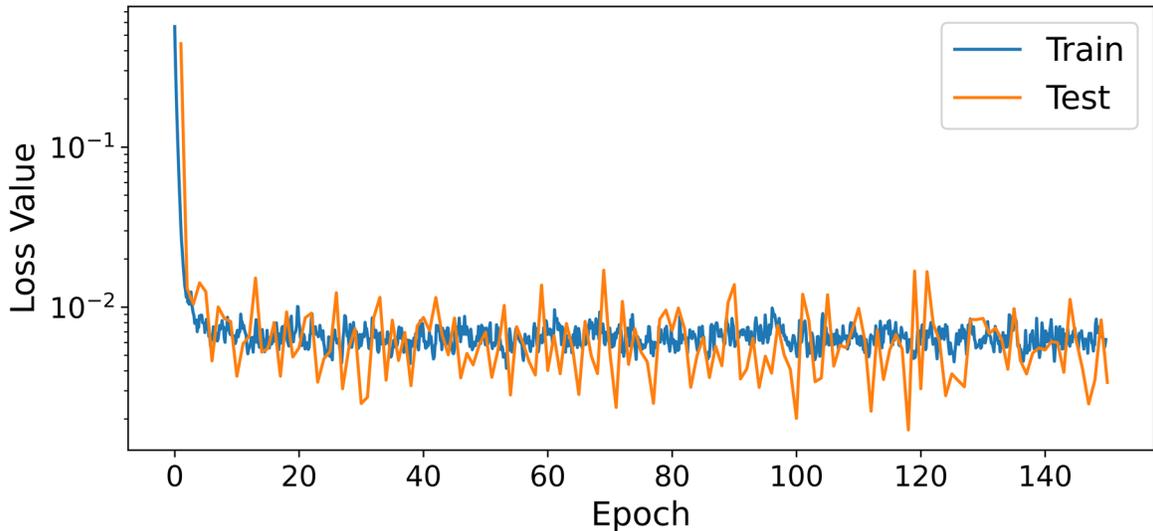


Figure 2. Training (blue) and test (orange) loss values over epoch number. The loss values decrease rapidly during the first few epochs of the training process before converging. The test losses are more erratic because of the much smaller number of models involved.

Table 2

Disk Physical Parameters Varied in Creating the Training and Test Images

Parameter	Sampling	Ranges
Dust Mass (M_{\odot})	log-uniform	$6 \times (10^{-7} - 10^{-4})$
Critical Radius R_c (au)	log-uniform	10–300
Flaring Exponent β (-)	uniform	1.0–1.3
Scale Height H_0 (au)	uniform	5–25
Inner Radius (au)	log-uniform	0.1–10
Surface Density γ (-)	uniform	-1.5–0
Dust Settling α (-)	log-uniform	$3 \times (10^{-6} - 10^{-3})$

can be found in I. Angelo et al. (2022, in preparation), but we stress that the detailed disk parameterization is unimportant for the purposes of image reconstruction using machine learning. This exercise could be repeated for any combination of varied parameters, or other parameters altogether. So long as the resulting disk images provide a large diversity of apparent morphologies, the basic methods and results would remain qualitatively unchanged.

Using this grid allowed us to cover a broad parameter space while making use of preexisting resources to cut computational costs. However, the coarse nature of the resulting training set resulted in preliminary poor image reconstruction for areas of the parameter space that lay between the grid points. In order to achieve better interpolation between the grid points, we also computed 15,000 models with parameter values (linearly or logarithmically) uniformly randomly sampled between the grid boundaries, at the same 15 inclinations sampled in the grid.

For both the initial grid and the additional randomly sampled models, some combinations of parameters lead to non-edge-on configurations. Since these are outside the scope of this study, we applied criteria designed in Angelo et al. (in prep.) to filter out the non-EODs. In short, these tests ensure that the central star is not seen directly in the image so that all photons received have scattered off the disk. Altogether, this led to a final library of 73,764 distinct EOD images to be used in the AE training (Telkamp et al. 2022). The top row of Figure 3 shows a sample of RT images with different morphologies, demonstrating the variety of disk shapes and sizes within this library. In this

figure, we can see that the disks span broad ranges in radius, thickness, and flaring.

2.5. Autoencoder Results

After training the AE on the RT images, we examined the reconstructed images, examples of which are shown in Figure 3. The remarkable similarity between the input and reconstructed images indicates that the AE successfully learned how to encapsulate most of the information contained within a disk image in an 8D space. The images in Figure 3 demonstrate the ability of the AE to preserve features such as the size and shape of the disks, the separation between the two nebulae, and the brightness ratio of the two nebulae. The residuals are structured but faint, indicating that the reconstructions are close to the input images but not perfect, as we expected from the dimensionality reduction taking place before the image reconstruction. The most significant shortcoming of the AE-produced images is that they are blurrier than the original RT images. Coupled with the min-max normalization of the image, which enforces the peak pixel to have the same values in both the RT and AE images, this leads to residuals, which are predominantly negative (see Figure 3) but of modest amplitude. Given the broad diversity of disk images that we trained it on and the inherent imperfection of such reconstructions, this level of performance was not guaranteed, although it is a necessary outcome to achieve our goal of generating new disk images to be used for forward modeling problems.

3. Connecting the Physical and Latent Space

So far, we have discussed the training of an AE neural network to reconstruct synthetic images of EODs without any explicit information about the physical parameters. However, our ultimate goal is to produce such images for any combination of physical parameters that are not part of our limited training set without having to generate computationally expensive RT model images. Achieving this goal entails tackling two distinct challenges toward this goal: ensuring that the AE framework can determine whether a specific combination of parameters leads to an EOD configuration, and translating the learned latent space of the AE to the physical parameters that are astrophysically compelling.

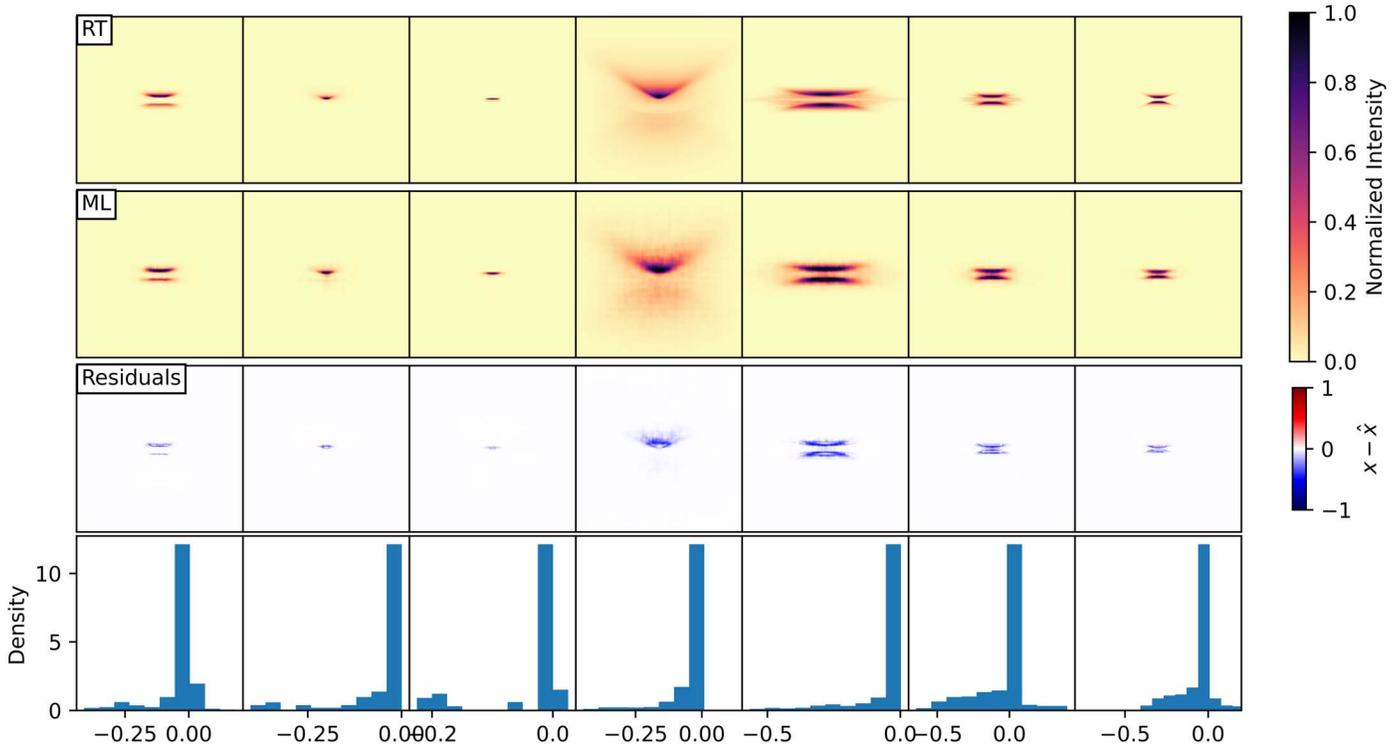


Figure 3. Comparison between machine learning (ML) and RT images during the testing phase. The first two rows represent the RT and AE-reconstructed images, respectively, while the third row shows the difference between the two. The fourth row provides a histogram plot of the residual map evaluated at all pixels with a brightness of at least 3% of the peak, in either the ML or RT image. For cosmetic purposes, all images are shown with the same orientation, corrected for the flips and rotations introduced during the training phase.

3.1. Automated Edge-on Classification

The AE described so far has only been trained to predict images of EODs, as our training data set only consists of such systems. This is because training an AE simultaneously on edge-on and non-edge-on images represents a much greater challenge due to the large variation in morphologies the AE would need to learn to predict. Furthermore, an additional complication would arise from systems where the star is in direct view, introducing a bright point source and a high contrast between the star and disk. This could degrade the quality of the EODs reconstructions, which are the focus of our analysis.

Since non-EODs are outside of the AE’s learned parameter space, the AE always predicts EOD-like images, even if the combination of physical parameters leads to a non-EOD configuration. Therefore, without determining a priori which combination of parameters leads to an EOD configuration, the output of the AE could not be trusted.

To solve this problem, we trained a random forest classifier (RFC; Breiman 2001) to predict, from a set of physical parameters, whether a disk would be considered an EOD based on the criteria defined by Angelo et al. (in prep.). RFCs train a set of decision trees on subsets of the data and then average the results of all of the trees. We trained and tested the RFC using all of the EOD and non-EOD image parameters in both the grid of models and the batches generated to fill in the gaps. We used a 90%–10% split between training and test images. We used 800 trees in the random forest with a binary classification threshold of 0.61, which we fine-tuned using precision-recall curves. We performed a grid search to select the number of trees, number of features considered at each split, maximum number of levels in a tree, minimum number of samples needed to split a node,

Table 3

Random Forest Classifier Precision-recall Probabilities for the Test Images			
True Category	N_{images}	Predicted	Category
Non-EOD	48,174	Non-EOD	EOD
EOD	7380	0.99	0.01
		0.12	0.88

minimum number of samples needed at each leaf node, and whether or not bootstrapping is used (Breiman 2001).

From the testing results (see Table 3), we found the RFC to correctly predict the class of a disk in the vast majority of cases. While its false negative probability is larger than one could hope, we note that the RFC-determined category is correct in 97.5% of cases. In other words, we have successfully trained the RFC to have a similar (and small) fraction of both false positives and false negatives. In addition, the false predictions showed no clear correlation with any of the physical parameters. The main features that drive the classifier were inclination, dust mass, and critical radius, which are the same features that are most correlated with the latent space (see Section 3.3).

3.2. Correlation with Physical Parameters

We have shown that the AE has learned how to compress the data into latent vectors while retaining the necessary information to produce accurate image reconstructions. Given this framework, we could generate a random sample of latent space vectors and pass them through the decoder side of the trained model to create new EOD images. However, these images would be untethered by any physical constraints and would have limited usefulness in a forward modeling context. Instead,

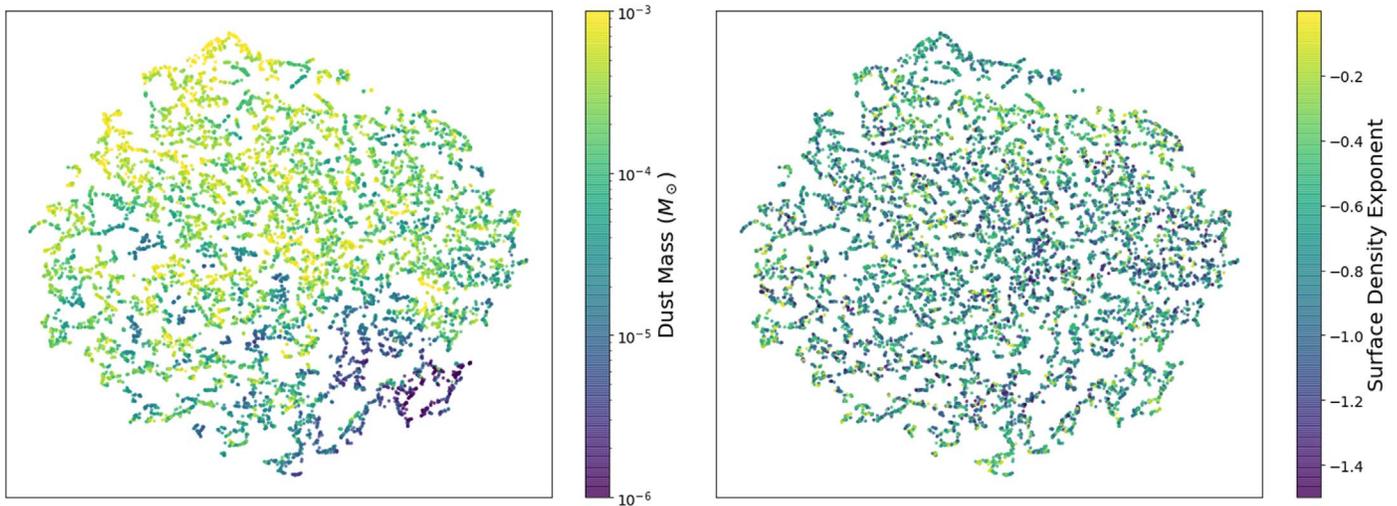


Figure 4. T-SNE projection color-coded by dust mass (left) and surface density exponent (right). The left panel provides an example of a parameter that is well correlated with the latent space, whereas the right panel is an example of a parameter with weak or no correlation with the latent space.

we aim to connect the learned latent space to the physical parameters that define our images, allowing the generation of new images directly from physical parameters. In order to connect the two parameter spaces, however, a sufficiently strong correlation must already exist between them. In other words, the AE must encode enough information about the physical parameters into the latent space to allow us to identify a strong relationship between the latent and physical parameter spaces. To ensure that this type of connection can be made, we embed the physical parameters that describe the input images into the latent space along with their compressed representations. We accomplish this by injecting the physical parameters as extra features after the convolutional layers (see the encoder side of Figure 1). In this way, the spatial features extracted by the convolutional layers are combined with the physical parameters when passing through the fully connected layers.

The insertion of the physical parameters into the latent vector is intended to strengthen the correlation between the embedded data and the physical parameters themselves. In Figure 4, we visualize a series of t-distributed stochastic neighbor embedding (t-SNE; van der Maaten & Hinton 2008) projections of the latent space, color-coded by physical parameter value. T-SNE projections map a high-dimensional data space to 2D representations that preserve the distances between data points. A visual examination of these projections reveals clear groupings by disk parameter values. This indicates that disks of similar physical parameter values are close together in the latent space, meaning the AE has embedded a correlation between the multidimensional latent and physical parameter spaces. In these projections, we observe strong clustering by dust mass (see left projection in Figure 4), critical radius, flaring exponent, scale height, and inclination. Not all of the physical parameters, however, appear to be correlated with the latent space. As shown in the right plot of Figure 4, the surface density exponent values appear to be randomly distributed in the t-SNE projection, which suggests that the AE has not learned a connection between this parameter and the embedded data. We observe a similarly random scatter in the t-SNE projections for the inner radius and the dust settling parameter. Rather than pointing at a shortcoming of the neural network, this confirms the expectation that these parameters have a much smaller, if any, impact on the morphology of the disks observed

in scattered light (e.g., Burrows et al. 1996). This supports the idea that the AE has only learned physically significant connections between the disk morphology and physical parameters.

To investigate the strength of the connection between the physical and latent spaces, we calculate the Spearman correlation coefficient between each physical-latent vector pair. As shown in Figure 5, each latent parameter exhibits significant correlation with one or more physical parameters. The presence of correlation between each latent space variable and multiple physical parameters indicates the presence of a nonbijective relationship between the spaces. We also observe that certain physical parameters, such as the disk inclination, exhibit a much stronger connection with the latent space and will thus control the generative process more. Other parameters, such as the surface density exponent, show a much weaker relationship with the latent space. These physical parameters also are not associated with any grouping in the t-SNE plots, as shown in Figure 4. Both of these facts indicate that these parameters will remain poorly constrained, as expected from the minimal impact that these parameters have on the morphology of the EODs observed in scattered light (e.g., Burrows et al. 1996). We note that the Spearman correlation coefficients are very similar in both the case where we embed the physical parameter information into the latent space and the case where we do not, with slightly larger correlations in the former case. This indicates that the AE gains most of its knowledge from the images, not the parameters themselves. Since the inclusion of the physical parameters does not improve or diminish the reconstructed images, the motivation to include them is mostly to strengthen the correlation with the latent space.

We also calculated the Pearson correlation coefficients between the two sets of variables as a test of their linear relationship and found that there is an almost nonexistent linear correlation between most of the physical parameters and the latent variables. This indicates that the mapping between the physical and latent space parameters that our machine learning framework has learned is nonlinear. This is in line with the known degeneracies in fitting scattered light images of PPDs (e.g., Burrows et al. 1996; Wolff et al. 2021).

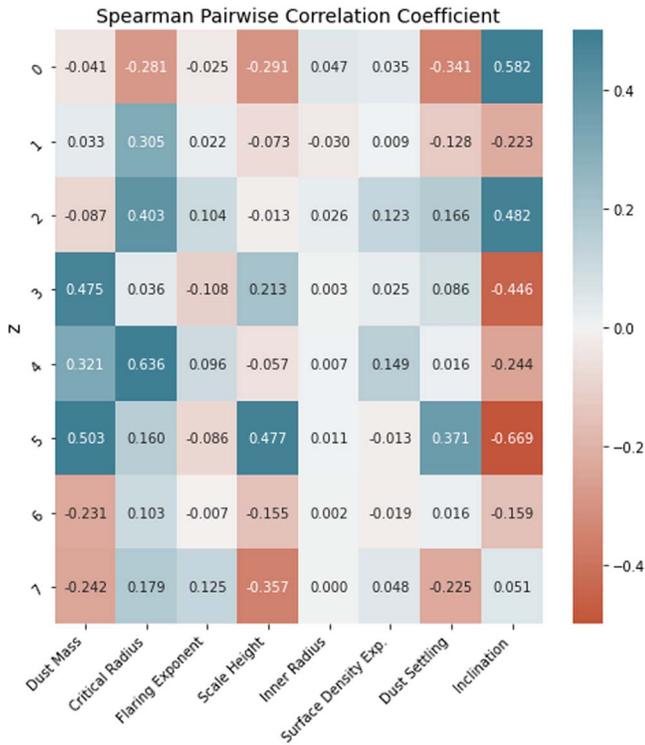


Figure 5. Spearman pairwise correlation coefficients between latent and physical parameters (vertical and horizontal axes, respectively). Greater color saturation indicates a stronger connection between parameters.

3.3. Identifying the Relationship between Physical and Latent Space Parameters

To leverage the correlation between the latent and physical parameter spaces, we aimed to identify an explicit mapping between them. Such mapping would take in a particular combination of physical parameters and predict its corresponding position in the latent space. Due to degeneracies and the intrinsically imperfect nature of the latent space representation of the data, we expected to need a nonlinear mapping between the two spaces. To confirm this assumption, we first tried using an ordinary least squares linear regression model, using `scikit`'s implementation (Pedregosa et al. 2011) to produce an invertible, multivariate linear model mapping the physical parameters of our data to their latent space vectors. As predicted, this produced poor results, with a coefficient of determination (R^2) score of 0.56 (compared to the ideal score of 1 for a perfect prediction).

To produce a nonlinear mapping between the two spaces, we tried random forest regression, where we used the random forest regressor (RFR) implementation of Pedregosa et al. (2011) to predict the latent space vectors from a given physical vector. This technique is similar to the RFC discussed in Section 3.1 but returns values of continuous variables instead of a class. We trained an RFR with 100 trees. This method consistently attained an R^2 score of 0.91 in the case where we embedded the physical parameters in the latent space and 0.90 when we did not. As with the correlation coefficients, the minimal difference between these two scores indicates that the AE embeds sufficient information about the physical parameters into the latent space regardless of whether or not we provide the parameters during training. Since the RFR performed significantly better than the linear regressor, we adopted this method in our image generation process. Once we trained the RFR to map physical parameters to latent space

values, we were equipped to generate new disk images directly from physical parameters.

4. End-to-end Framework

Figure 6 shows a schematic of the overall process of translating a set of physical parameters into an EOD image. This process begins with a set of physical parameter values sampled between the boundaries described in Table 2. Next, the parameters are either (a) input into the (MCFOST) RT code, which outputs the corresponding scattered light image directly at a comparatively high computational cost, or (b) inserted into the machine learning framework. If the latter route is taken, the RFC first uses these parameters to predict whether or not the disk is viewed as edge-on. For non-EODs, the algorithm outputs the PSF as the image to represent the fact that the image is dominated by a central star. For EODs, the RFR function translates the physical parameter combination into a latent vector. This latent vector then passes through the decoder side of the AE. The result is a full-sized image that our machine learning framework has predicted from the inputted physical parameters.

To assess the performance and efficiency of this end-to-end algorithm in approximating the results of RT calculations, we defined a group of 300 test disks with randomly sampled parameter values and computed both the machine learning and RT images that correspond to these points in parameter space. On average, the RT process took 5.1 ± 2.3 s to generate an image at a single inclination, whereas the machine learning image generation took 18.9 ± 4.4 ms. Typically, computing an ML image using a CPU takes around 2 to 3 orders of magnitude less time than computing the corresponding RT image. Using a single graphics processing unit (GPU), another order of magnitude in efficiency is gained. Furthermore, when using a GPU, the majority of the computing time stems from porting information between the CPU and GPU. Thus, it takes roughly the same amount of time to generate an arbitrarily large batch of images (only constrained by the GPU memory) from a set of physical parameter combinations as it does to generate a single image. All in all, this work confirms that the computing time gain from the machine learning process has the potential to speed up the modeling of any disk image by several orders of magnitude. This is particularly true for image modeling in an MCMC framework, as a single batch of many walkers can be computed at once on a GPU.

After we obtained the RT and machine learning images, we computed the residual maps that correspond to each ML-RT pair. These images are shown in Figure 7. Similar to the reconstructions in Figure 3, the machine learning-generated images typically preserve features of the original disk image, such as the disk size and shape and the brightness ratio and distance between the two nebulae. The median fraction of pixels which are within 5% of the RT intensity is 72%. It is important to stress that, unlike the reconstructions shown in Figure 3, these new disk images were generated directly from the physical parameters corresponding to a given disk. From this figure, we can see that the AE is able to produce images reasonably close to the corresponding RT images without any RT calculations or previous knowledge about the disk morphology. While the generated images are able to preserve the disk size and shape without this information, on a more detailed level, the images are somewhat noisy and blurry. The same shortcoming was observed in the reconstructed images

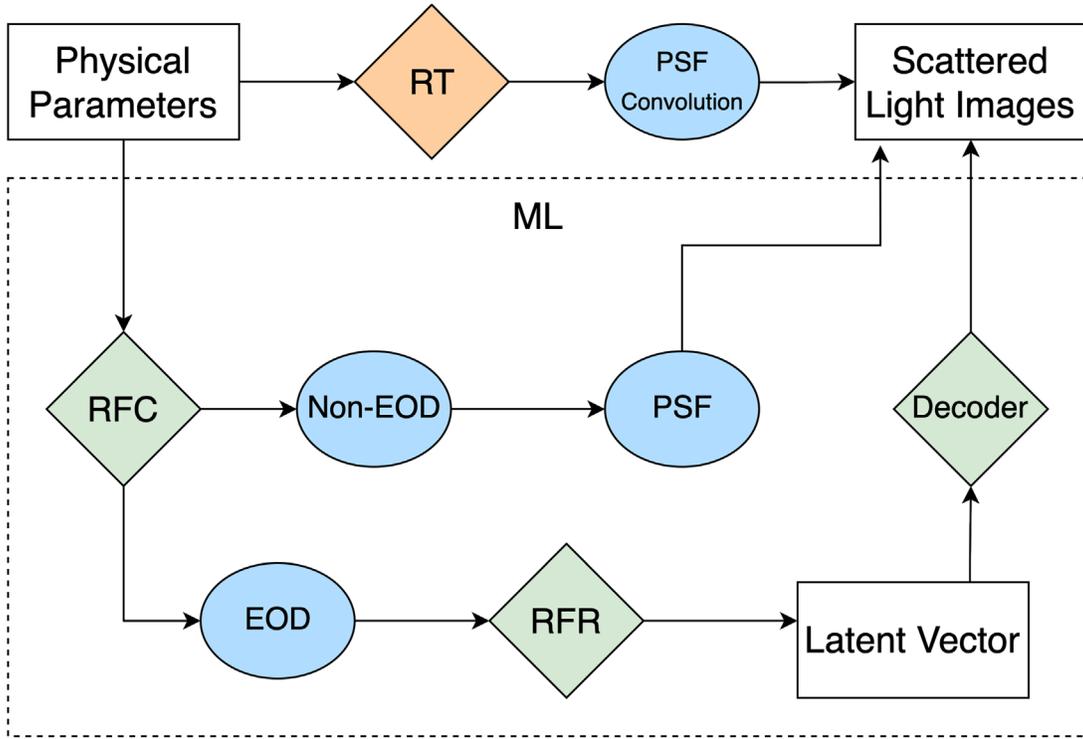


Figure 6. Schematic showing the image generation process from physical parameters, either through the radiative transfer code or through the machine learning pipeline.

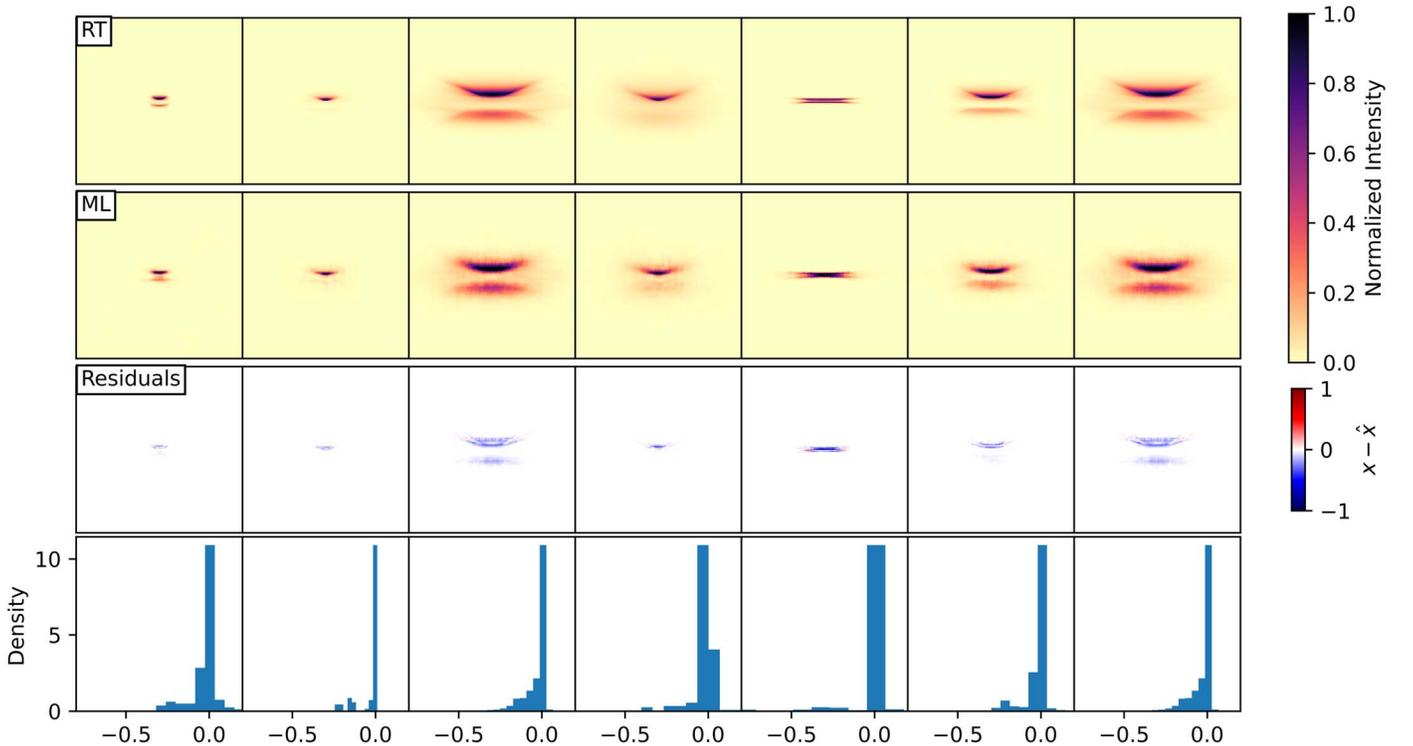


Figure 7. Comparison of EOD images computed through the standard RT approach (top row) and through the end-to-end machine learning pipeline (second row from top). As in Figure 3, the bottom two rows show the difference between the two and a histogram of the residuals for all pixels with a brightness of at least 3% of the peak, in either the ML or RT image. Unlike the reconstructions shown in Figure 3, these images were generated directly from physical parameters.

used in the training process (see Figure 3) and thus is an intrinsic limitation of the current AE implementation.

To further evaluate the performance of the AE, we also considered how its predictions compared to images of known

EODs. To this end, we selected two objects that have previously been modeled using the same disk parameterization as we adopted here. These disks are ESO H α 569 and STSC2D J163131.2-242627 (Wolff et al. 2017, 2021, respectively). For both systems,

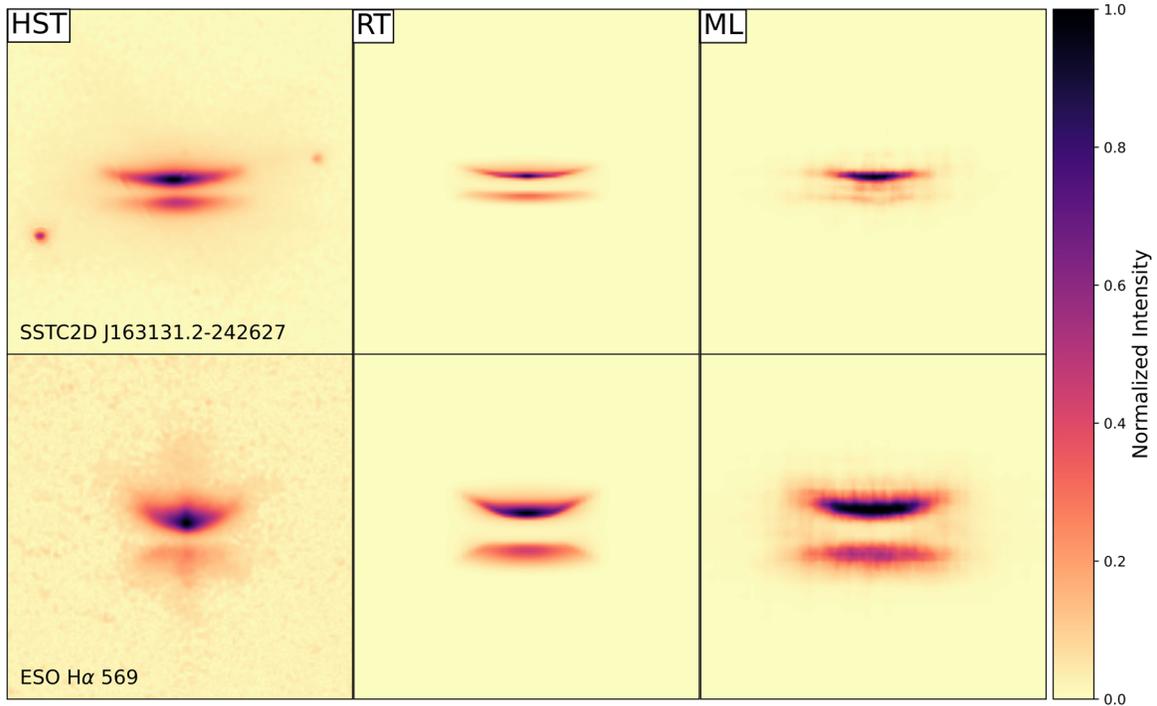


Figure 8. Observed $0.6 \mu\text{m}$ images of two known EODs (left column; Wolff et al. 2017, 2021) and corresponding RT- and ML-predicted model images. All images are shown on the same angular scale and stretch. The two faint point sources seen in the image of STSC2D J163131.2–242627 are unrelated background stars. Note that the ESO H α 569 hosts a jet that is visible in the HST $0.6 \mu\text{m}$ and increases the disk’s peak brightness toward its center. The image normalization applied here thus artificially reduces the contrast in the observed image compared to the RT and ML images.

we extracted the physical parameters from the best-fitting RT model to the HST image of the disk. We adjusted a handful of parameters to remain within the scope of the AE (see Table 2): the flaring of the STSC2D J163131.2–242627 disk was reduced from the original (unphysical) value of $\beta = 1.5$ to 1.3, the maximum value considered here. Similarly, the surface density exponent for ESO H α 569 was modified from $\gamma = -1.77$ to -1.5 . Since the model for ESO H α 569 did not include settling, we set the turbulence parameter to its highest value ($\alpha = 3 \cdot 10^{-3}$), which effectively leads to negligible settling. Finally, we adjusted the characteristic disk radius of ESO H α 569 from $R_c = 125$ au down to 100 au to compensate for the fact that, in the original model, the disk surface density was artificially truncated at that radius. We also note that the total dust mass for both disks is close to, if not equal to, the maximum value considered in our study. As a general rule, using ML predictive models close to the edge of the parameter space included in the training set increases the risk of incorrect predictions. Nonetheless we proceed to compute RT and ML images corresponding to both sets of physical parameters (see Figure 8). Interestingly, the two disks represent markedly different morphologies, thus providing complementary tests of the performance of the AE. The shortcomings of the ML prediction discussed above are also apparent for these models. The aspect that is least well reproduced is the lower nebula in the STSC2D J163131.2–242627 disk. This is most likely a consequence of the AE difficulty in reproducing sharp structures, further enhanced by the large flux ratio between both nebulae.

We will discuss possible solutions to the AE shortcomings in Section 5. Nonetheless, these results shown here demonstrate that our machine learning framework can approximate, albeit imperfectly, RT calculations of EOD scattered light images using a computing time that is several orders of magnitude smaller.

5. Discussion and Conclusion

In its current state, the machine learning algorithm we presented here is capable of producing realistic and reasonably accurate synthetic EOD images directly from a set of physical parameters. These images typically take 2–3 orders of magnitude less time to generate than the corresponding RT images, significantly reducing the computation time of any model fitting technique (such as MCMC runs) that requires the generation of large volumes of images. While the AE produces images that are qualitatively very similar to the ground truth RT calculations, the algorithm has several limitations that currently restrict its quantitative accuracy. In particular, the quality of the produced images is degraded by extraneous noise as well as excessive blurriness, which we have not been able to suppress despite a large sweep of the hyperparameters of the model. Nonetheless, future modifications of the AE architecture could alleviate some of these issues.

For example, extending the AE architecture to a variational AE would help produce smoother latent spaces. This would improve the sampling process and could also lead to a better quality of reconstructed images. A second advantage of variational AEs is that their loss function can be used to impose orthogonality between the dimensions of the latent space, which could help to isolate different aspects of the generative process. Furthermore, Böhm & Seljak (2020) recently showed that coupling a normalizing flow with an AE (making it a probabilistic AE) improves the sample quality without requiring special tuning parameters with a longer training time, unlike variational AEs. In that approach, a normalizing flow is used to probabilistically interpret the latent space after training the AE. This allows for the generation of new data sets by sampling from the normalizing flow distribution and using the normalizing flow to obtain a point in the latent space. The latent space coordinate is then passed through the decoder to generate the corresponding image. Other

issues, such as the limited sensitivity of the algorithm to small changes in parameter values, could represent a fundamental limitation to the accuracy of results derived from using machine learning-generated images, but the work presented here has not reached that point, and it could be that these limitations exist at a level that is negligible in practice. It is therefore valuable to continue to improve the architecture and performance of the AE presented here.

While the current implementation of the AE is designed to generate $0.6 \mu\text{m}$ scattered light images, it can be easily extended to make simultaneous predictions at multiple wavelengths. To do this, the AE needs to be trained on images at multiple wavelengths. The architecture we have adopted can facilitate this, since the various images of each model can be injected as associated multiple channels of a single data set. This extension would enable the use of the AE in multiwavelength studies of PPD, which is desirable as different wavelengths are generally sensitive to physical parameters in different ways. In particular, thermal continuum emission maps probe millimeter-sized grains in the disk, which have decoupled from the gas component. As a result, disks are typically more compact and vertically thinner in the millimeter regime (e.g., Louvet et al. 2018; Villenave et al. 2020). Extending the image generation process presented here to the millimeter regime would allow us to better analyze the degree of dust settling in these disks, a key process in the formation of planets. Crucially, the physical distinction between thermal emission and scattered light images is irrelevant to the machine learning process, so long as it is trained on an adequately defined set of RT images.

While this process could be extended to non-EODs, this would present new challenges. By limiting our algorithm to EODs, we sidestepped the high-contrast problem introduced by lower inclination disks. Indeed, in these cases, the bright central point source produces a few pixels of high intensity, and the disk is distributed over many pixels of very low intensity, a contrast that is difficult to encode accurately. Besides, the instrumental PSF results in most pixels containing a complex (and generally hard to predict) combination of star and disk light. For both of these reasons, it is likely that an AE such as the one presented here would produce less accurate predictions.

Ultimately, one of the key goals of this work is to provide a more efficient means of generating model PPD images. This would enable uniform analyses of the entire sample of EODs that have been imaged in scattered light. Furthermore, the machine learning architecture could provide a solution to the *inversion problem*, i.e., a direct path from a disk image to the latent space and, from there, to the disk’s physical parameters. To enable this, we would need to identify an inverted regressor between the latent and physical parameter spaces. We would then use the encoder to map images into the latent space, and then use the inverted regressor to get the physical parameters. The RFR currently implemented in our ML framework is not invertible. One possible option is the use of normalizing flows, which would provide a probabilistic estimator of the physical parameters from the latent space vector. Achieving this inversion would allow us to constrain the physical parameters of a disk directly from the observations, without the need for RT or machine learning image generations and without computationally expensive MCMC runs. Either way, machine learning image generation of EODs offers a massive computational gain over RT calculation and, provided that the limitations of our current AE architecture can be overcome, could enable large-scale, systematic, and uniform analyses of large

samples of PPDs. In turn, this would inform the role that disk structure plays in planet formation.

This research used the Exalearn computational cluster resource provided by the IT Division at the Lawrence Berkeley National Laboratory (supported by the Director, Office of Science, Office of Basic Energy Sciences, the U.S. Department of Energy under contract No. DE-AC02-05CH11231).

We would like to thank the GPI team at The University of California, Berkeley (UC Berkeley) for insights on modeling, data processing, and analysis. This work was supported in part by the NASA NNX15AD95G/ NEXSS and 80NSSC18K0442 grants as well as by the UC Berkeley Undergraduate Research Apprenticeship Program.

Software: MCFOST (Pinte et al. 2006), numpy (van der Walt et al. 2011), matplotlib (Hunter 2007), pytorch (Paszke et al. 2019), jupyter (Kluyver et al. 2007), scikit-learn (Pedregosa et al. 2011), pandas (McKinney 2010), and Weights & Biases.

Appendix Forward Model

A.1. Forward Model Architecture

While there were incentives to build an AE model for EOD image generation, we also wanted to test the limitations of a simpler model. We aimed to demonstrate the need for the inherent complexity of the AE by testing the performance of a model that does not make use of an encoder and therefore does not break down an image into latent space. To this end, we built an FM that takes in a list of physical parameters as input, instead of an image like the AE, and aims to directly infer an EOD image. That FM is trained using the same data set as the AE.

Our FM mirrors the functionality of the decoder structure of the AE (only swapping the latent space for the physical parameter space), and their architectures are thus similar, consisting of the same two types of layers: convolutional and fully connected layers. In order to allow our model to perform well, a variety of structures and hyperparameters were tested. We varied kernel size on the convolutional layers, AFs (i.e., hyperbolic tangent, ReLU, sigmoid), number of neurons in the fully connected layers, as well as number of convolutional and fully connected layers, among others. We found that a LeakyReLU AF combined with a higher number of fully connected layers and slightly fewer convolutional layers than the AE worked best for our structure.

Our fully connected blocks consist of fully connected linear layers (FC), BN, and LeakyReLU AFs, connected to dropout layers (DO). Our fully connected blocks are structured as follows: $[\text{FC} + \text{BN} + \text{AF} + \text{DO}] \times 7 + [\text{FC} + \text{AF}]$. The complete FM contains a total of 8 blocks with the last block connected to an activation layer and directly to the convolutional layers. The convolutional layers’ architecture is exactly the same as that of the AE: $[\text{DeConv} + 2 \times (\text{Conv} + \text{BN})] \times 4$, where DeConv refers to a deconvolutional layer, and Conv to a convolutional layer, and each block is structured as previously defined for a total of 4 blocks. However, we have an extra block that consists of a deconvolution to convolution layer with a sigmoid AF at the end to rescale the data values accordingly. Due to the restricted similarity in structure of the FM to the AE, the flow of data is similar with the defining difference that we do not feed the model images but rather the physical parameters.

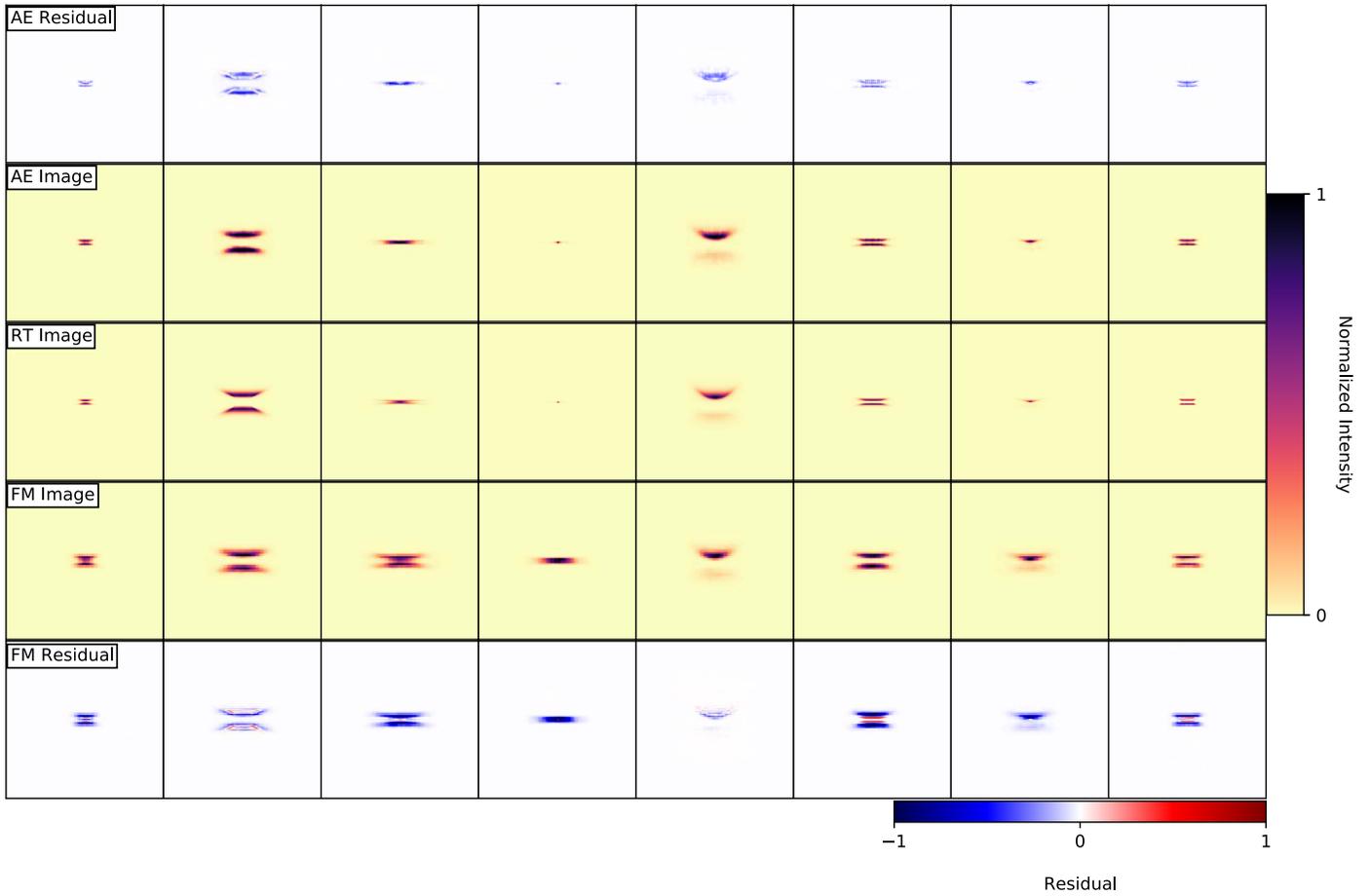


Figure 9. Comparison of predicted images from both machine learning models. The middle row represents the true images obtained from the RT model from random combinations of physical parameters. The second and fourth rows are the images recreated by the AE and the FM, respectively. The first and last rows show the residuals from the RT image (i.e., $RT - ML$, where ML is a model either FM or AE).

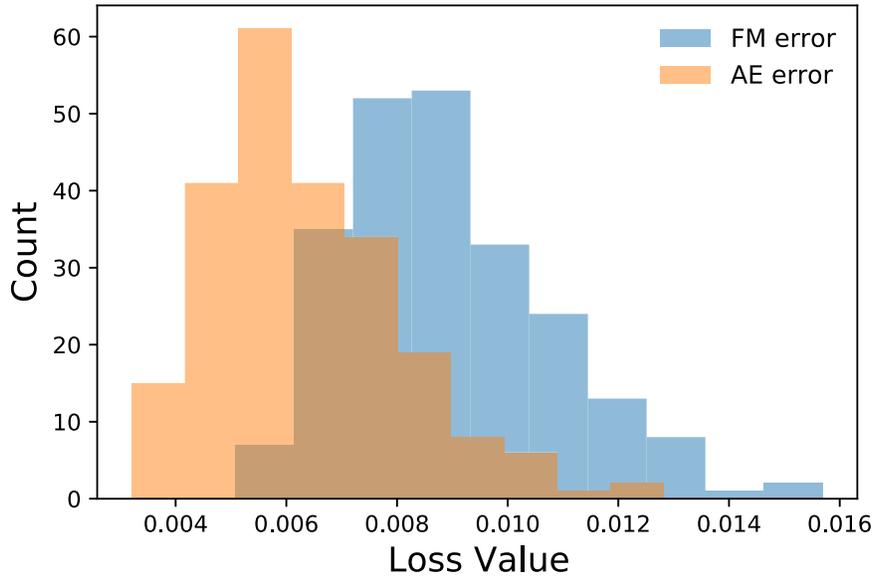


Figure 10. Distribution of loss values for roughly 15,000 individual images computed with the FM and AE models (blue and orange histograms, respectively), indicating that the latter generally performs significantly better than the former.

A.2. Forward Model Performance

After testing our FM and comparing its performance to the AE model, we have found that, on average, the training time of the FM is about 4 times faster than the AE model due to its

simpler architecture. On the downside, the FM generates lower quality images with often very high residuals (see Figure 9). By examining the loss distribution of the two models (Figure 10), we found that the FM has a higher loss function than the AE,

i.e., the AE model produces more accurate images with a higher quality. An important general trend is that the FM performs poorly when the disks are the most compact (in either direction) but generates a roughly similar image to the AE-generated image for the larger and thicker disks. After testing the limitations of this FM, we conclude the AE model to perform better in general, producing higher quality images, lower residuals, and a lower loss function. Therefore, we have elected to use an AE model over the simpler FM for the purposes of this work.

ORCID iDs

Zoie Telkamp  <https://orcid.org/0000-0001-6465-9590>

Jorge Martínez-Palomera  <https://orcid.org/0000-0002-7395-4935>

Gaspard Duchêne  <https://orcid.org/0000-0002-5092-6464>

Isabel Angelo  <https://orcid.org/0000-0002-9751-2664>

Christophe Pinte  <https://orcid.org/0000-0001-5907-5179>

References

- Auddy, S., Dey, R., Lin, M.-K., & Hall, C. 2021, *ApJ*, 920, 3
- Bastien, P., & Menard, F. 1988, *ApJ*, 326, 334
- Benisty, M., Dominik, C., Follette, K., et al. 2022, arXiv:2203.09991
- Biewald, L. 2020, Experiment Tracking with Weights and Biases, <https://wandb.ai/site>
- Böhm, V., & Seljak, U. 2020, CoRR, arXiv:2006.05479
- Breiman, L. 2001, *Mach. Learn.*, 45, 5
- Burrows, C. J., Stapelfeldt, K. R., Watson, A. M., et al. 1996, *ApJ*, 473, 437
- Cabrera-Vives, G., Reyes, I., Forster, F., Estevez, P. A., & Maureira, J.-C. 2017, *ApJ*, 836, 97
- Dia, M., Savary, E., Melchior, M., & Courbin, F. 2019, in ASP Conf. Ser. 527, *Astronomical Data Analysis Software and Systems XXIX* (San Francisco, CA: ASP)
- Duchene, G., McCabe, C., Pinte, C., et al. 2010, *ApJ*, 712, 112
- Fromang, S., & Nelson, R. P. 2009, *A&A*, 496, 597
- Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (Cambridge, MA: MIT Press)
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. 2014, *Generative Adversarial Networks*, arXiv:1406.2661
- Hunter, J. D. 2007, *CSE*, 9, 90
- Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
- Kingma, D. P., & Welling, M. 2014, arXiv:1312.6114
- Kluyver, T., Ragan-Kelley, B., & Perez, F. 2007, *CSE*, 9, 90
- Krist, J. 1995, in ASP Conf. Ser., 77, *Astronomical Data Analysis Software and Systems IV*, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco, CA: ASP), 349
- Louvet, F., Dougados, C., Cabrit, S., et al. 2018, *A&A*, 618, A120
- Martínez-Palomera, J., Bloom, J. S., & Abrahams, E. S. 2020, arXiv:2005.07773
- McKinney, W. 2010, in Proc. of the 9th Python in Science Conf., ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56
- Miotello, A., Kamp, I., Birnstiel, T., Cleevess, L. I., & Kataoka, A. 2022, arXiv:2203.09818
- Mustafa, M., Bard, D., Bhimji, W., et al. 2019, *ComAC*, 6, 1
- Paszke, A., Gross, S., Massa, F., et al. 2019, in *Advances in Neural Information Processing Systems 32*, ed. H. Wallach et al. (Red Hook, NY: Curran Associates Inc.), 8024
- Pedregosa, F., Varoquaux, G., & Gramfort, A. 2011, *JMLR*, 12, 2825
- Pinte, C., Harries, T. J., Min, M., et al. 2009, *A&A*, 498, 967
- Pinte, C., Ménard, F., Duchêne, G., & Bastien, P. 2006, *A&A*, 459, 797
- Ribas, A., Espaillat, C. C., Macías, E., & Sarro, L. M. 2006, *A&A*, 459, 797
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, *Natur*, 323, 533
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *JMLR*, 15, 1929
- Stapelfeldt, K. R., Menard, F., Watson, A. M., et al. 2003, *ApJ*, 589, 410
- Telkamp, Z., Martínez-Palomera, J., & Duchêne, G. 2022, *Radiative Transfer Edge-on Protoplanetary Disk Images, v1.0*, Zenodo, doi: 10.5281/zenodo.7032727
- van der Maaten, L., & Hinton, G. 2008, *JMLR*, 9, 2579
- van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, *CSE*, 13, 22
- Villenave, M., Ménard, F., Dent, W. R. F., et al. 2020, *A&A*, 642, A164
- Watson, A. M., Stapelfeldt, K. R., Wood, K., & Ménard, F. 2007, in *Protostars and Planets V*, ed. B. Reipurth, D. Jewitt, & K. Keil, 523
- Whitney, B. A., & Hartmann, L. 1992, *ApJ*, 395, 529
- Williams, J. P., & Cieza, L. A. 2011, *ARA&A*, 49, 67
- Winn, J. N., & Fabrycky, D. C. 2015, *ARA&A*, 53, 409
- Wolff, S. G., Duchene, G., Stapelfeldt, K. R., et al. 2021, *AJ*, 161, 238
- Wolff, S. G., Perrin, M. D., Stapelfeldt, K., et al. 2017, *ApJ*, 851, 56
- Yi, K., Guo, Y., Fan, Y., Hamann, J., & Wang, Y. G. 2020, arXiv:2001.11651
- Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. 2010, in 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 2528
- Zhang, S., Zhu, Z., & Kang, M. 2022, *MNRAS*, 510, 4473