Deep learning to predict the lifetime of compact planetary systems

Miles Cranmer[a,†], Daniel Tamayo[a], Hanno Rein[b,c], Peter Battaglia[d], Samuel Hadden[e], Philip J. Armitage[f,g], Shirley Ho[g,a,h], and David Spergel[g,a]

[a]Department of Astrophysical Sciences, Princeton University, Princeton, New Jersey 08544, USA; [b]Department of Physical and Environmental Sciences, University of Toronto at Scarborough, Toronto, Ontario M1C 1A4, Canada; [c]David A. Dunlap Department of Astronomy and Astrophysics, University of Toronto, Toronto, Ontario, M5S 3H4, Canada; [d]DeepMind, London, UK; [e]Center for Astrophysics | Harvard & Smithsonian, 60 Garden St., MS 51, Cambridge, MA 02138, USA; [f]Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11790, USA; [g]Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA; [h]Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15217, USA

This manuscript was compiled on December 9, 2020

We introduce a Bayesian neural network model that can accurately predict not only if, but also when a compact planetary system with three or more planets will go unstable. Our model, trained directly from short N-body time series of orbital elements, is more than two orders of magnitude more accurate at predicting instability times than analytical estimators, while also reducing the bias of existing machine learning algorithms by nearly a factor of three. Despite being trained on compact resonant and near-resonant three-planet configurations, the model demonstrates robust generalization to both non-resonant and higher multiplicity configurations, in the latter case outperforming models fit to that specific set of integrations. The model computes instability estimates up to 10^5 times faster than a numerical integrator, and unlike previous efforts provides confidence intervals on its predictions. Our model will be made publicly available in the `SPOCK*` package.

Deep Learning | Planetary Dynamics | Bayesian Analysis | Chaos

The final growth of terrestrial bodies in current theories of planet formation occurs in a phase of giant impacts (1). During this stage, the number of planets slowly declines as bodies collide and merge (2, 3). Close planetary encounters and the wide dynamic range exhibited by the times between consecutive collisions computationally limit current numerical efforts to model this process. Two theoretical roadblocks impede the development of a more efficient iterative map for modeling planet formation. First, one must predict a distribution of instability times from a given initial orbital configuration. Second, one must predict a distribution of post-collision orbital architectures (e.g., 4) subject to mass, energy and angular momentum constraints. Toward this end, we focus on the longstanding former question of instability time prediction.

In the compact dynamical configurations that characterize the planet formation process, the simpler two-planet case is understood analytically. In this limit, instabilities are driven by the interactions between nearby mean motion resonances (MMRs), i.e. integer commensurabilities between the orbital periods of the planets like the 3:2 MMR between Pluto and Neptune (5–8). While the general higher-multiplicity case is not yet understood, two important results guide our analysis and provide an important test for any model. First, when planets are initialized on circular orbits, chaos is driven by the overlap of 3-body MMRs between trios of adjacent planets (9), and theoretical estimates of the timescale required for the orbits to reach orbit-crossing configurations accurately match numerical integrations (10). As we show below, such analytical estimates perform poorly in the generic eccentric case where the effects of 2-body MMRs are dominant (10, 11). However, analytical and empirical studies agree that while the dynamical behavior changes strongly from the two to three-planet case (3, 12**?** –17), three-planet systems are the simplest prototype for predictions at higher multiplicities in compact systems (10, 11).

We recently presented a machine learning model, dubbed the Stability of Planetary Orbital Configurations Klassifier, or SPOCK, trained to classify the stability of compact planetary systems over timescales of 10^9 orbits (11). This represented a long-term effort to exploit the substantial but incomplete current analytical understanding (5, 6, 8, 18, 19) to engineer summary metrics that captured these systems' chaotic dynamics; these features were then used by the machine learning model to classify whether the input configuration would be stable over 10^9 orbits.

While simple binary stability classification is effective for constraining physical and orbital parameters consistent with long-term stability (20), other applications like modeling terrestrial planet formation require the prediction of continuous instability times. Additionally, several fields in which it is challenging to find effective hand-picked features—such as computer vision, speech recognition, and text translation—have been revolutionized by neural networks in the last decade (notable early breakthroughs include 21–23). Rather than relying on domain expert input, these flexible models learn data-driven features that can often significantly outperform human-engineered approaches. A key theme with deep learning models is that their

Significance Statement

Despite over three hundred years of effort, there are no known solutions for determining whether a general planetary configuration will remain long-term stable, or how long it will take for planets to eventually collide or be ejected from the system. We implement a new Bayesian neural network architecture which borrows its non-standard pooling operation from theoretical approaches to this problem, which helps it outperform traditional time series models. Our model can quickly and accurately predict instability timescales in compact multi-planet systems. This computationally opens up the development of fast terrestrial planet formation models, and enables the efficient exploration of stable regions of parameter space in multi-planet systems. While current machine learning (ML) relies on scientist-derived metrics, this algorithm uses raw orbital elements to learn its own transformed coordinate system and instability metrics from time series data.

Please provide details of author contributions here.

Please declare any competing interests here.

[†]To whom correspondence should be addressed. E-mail: mcranmer@princeton.edu

structure resembles the hand-designed algorithm, but with added flexibility parametrized by neural networks (for discussion, see 24). For example, modern computer vision models consist of learned convolutional filters which take the place of hand-designed filters in classic algorithms (25).

Pursuing a deep learning approach, we present a neural network that, trained only on short time series of the orbits in compact planetary systems, not only improves on long-term predictions of previous models based on engineered features (11, 26), but also significantly reduces the model bias and improves generalization beyond the training set. We design our model as a Bayesian neural network, which naturally incorporates confidence intervals into its instability time predictions, accounting both for model uncertainty as well as the intrinsic uncertainty due to the chaotic dynamics. Finally, unlike previous machine learning models based on decision trees (11, 26), our model is differentiable. That is, this provides estimates of the derivatives of the predicted instability times with respect to the parameters defining the orbital configuration in question. Such gradient information can significantly speed up parameter estimation using Hamiltonian Monte Carlo techniques (27).

Model

Dataset generation. We focus on the regime leading to typical compact multi-planet systems observed to date, with mass ratios with the central star ranging from 10^{-7} (roughly the ratio of the Moon-mass embryos thought to initially characterize the giant impact phase, relative to the Sun) to 10^{-4} (roughly Neptune's mass relative to the Sun). As detailed in *Materials and Methods*, we place planets on nearly co-planar orbits, with adjacent planets spaced within 30 mutual Hill radii† of one another (e.g., 28). Orbital eccentricities in observed systems are often poorly constrained, so we consider the range from initially circular to orbit-crossing values.

We train our model on the set of 113,543 publicly available, compact three-planet configurations in and near strong resonances from (11). In particular, this "resonant" dataset initializes one pair of planets in or near a strong MMR, while the third planet's orbital parameters are chosen randomly. This choice focuses the training set on the narrow resonant regions of phase space where the dynamical behavior changes most strongly, and we later test the model's generalization to non-resonant systems in Results.

Each initial condition was integrated for 10^9 orbits of the innermost planet using the `WHFast` integrator (29) in the REBOUND N-body package (30). If at any point two planets came within a distance of one another given by the sum of their Hill radii, the simulation was stopped and the instability time was recorded. Because gravity is scale invariant, the instability time $t_{\rm inst}$ is most usefully non-dimensionalized by the innermost orbital period $P_{\rm orb}$. Given the large dynamic range in timescales over which instabilities can occur, we define the dimensionless log instability time $T \equiv \log_{10}(t_{\rm inst}/P_{\rm orb})$. Configurations with instability times longer than 10^9 orbits ($T > 9$) were labeled as stable, and integration was stopped early.

Network architecture. To predict systems' instability times, we perform a computationally cheap numerical integration of the first 10^4 orbits, and use this time series to make long-term predictions. Each of the three planets' 3D positions and velocities correspond to six standard orbital elements (*Materials and Methods*), which we record at $n_t = 100$ equally spaced outputs across the short integration along with the corresponding time. In addition we pass the three constant mass ratios for each planet relative to the star, for a combined input matrix of real values $X \in \mathbb{R}^{3+19\times n_t}$ for a given configuration.

Because the dynamics of compact multi-planet systems are chaotic, instability times for a given initial orbital configuration are practically not deterministic. Nevertheless, numerical experiments (31, 32) have shown that instability times for unstable, compact multi-planetary systems settle to well defined, approximately log-normal distributions. Thus, rather than predicting a single instability time for a given orbital configuration, our model maps from an input initial orbital configuration to a predicted log-normal distribution of instability times, i.e., a Gaussian distribution of T with mean μ and variance σ^2. This gives the network the flexibility to both model the fundamental uncertainties imposed by the chaotic dynamics, and to incorporate model uncertainty into its predictions by assigning larger widths to configurations it is less sure about.

In our initial efforts, we experimented with various combinations of convolutional neural networks (see reviews by 33–35), long short-term memory networks (36), 1D scattering transforms (37), regular multi-layer perceptrons (MLP, see 24) and gaussian processes (38). All of these models underperformed or tended to overfit the data.

The fundamental challenge for making such predictions is the sharp transitions in dynamical behavior at MMRs, where instability times can change by several orders of magnitude (39) over sub-percent changes in planetary orbital periods, i.e., in the original space of orbital elements. We found substantially improved performance by structurally splitting the problem into three components: 1) Find a transformation from the sharply punctuated space of orbital elements to new variables. 2) Calculate statistical summaries of the time series in these transformed variables. 3) Use these summary features to predict a mean instability time μ for the input orbital configuration, as well as a log-normal width of the distribution σ. This is illustrated in fig. 1.

We model steps 1 and 3 with neural networks f_1 and f_2, respectively. In step 2, we choose to simply calculate a mean and variance for each transformed time series (brown in fig. 1). More sophisticated aggregations could potentially improve performance. However, this structure facilitates conceptual comparisons to (11), who similarly calculated means and standard deviations of time series transformed according to a simplified analytical two-planet model (rather than f_1).

Likelihood. Our model is parametrized by m neural network weights $\theta \equiv (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2), \theta \in \mathbb{R}^m$ ($\boldsymbol{\theta}_1$ for f_1 and $\boldsymbol{\theta}_2$ for f_2). Defining the training set D as the collection of input orbital configurations and their associated N-body instability times, we seek the most likely set of model parameters given the data, i.e., we maximize $P(\theta|D)$, which is in turn proportional to $P(D|\theta)P(\theta)$.

Our model predicts a lognormal distribution of instability times for any input orbital configuration. For a given set of network weights θ, the likelihood $P(D|\theta)$ is then simply the product of the probabilities that each training set example's output T_i is drawn from the associated Gaussian $\mathcal{N}(\mu_i, \sigma_i^2)$ predicted by the model. As discussed above, this choice is motivated by the numerical result that the distribution in T is normal, for different configurations with a wide range of mean instability times (32).

Note that we have $4 < T \leq 9$ as a constraint for unstable simulations: $T < 4$ simulations are not included in the training set, and $T > 9$ integrations were terminated at $T = 9$ and have an unknown T. Thus, we build a truncated normal distribution with a cutoff at $T = 4$, and the cumulative probability of the Gaussian above $T = 9$ being counted towards a classification of stability. A mathematical derivation of this likelihood is given in *Materials and Methods*.

†The mutual Hill radius R_H is a relevant length scale within which the gravity of the planets dominates that of the star $R_H \approx a_1(\mu_1+\mu_2)^{1/3}$, where a_1 is the inner planet's semimajor axis, and μ_1 and μ_2 are each planet's mass ratio relative to the star.

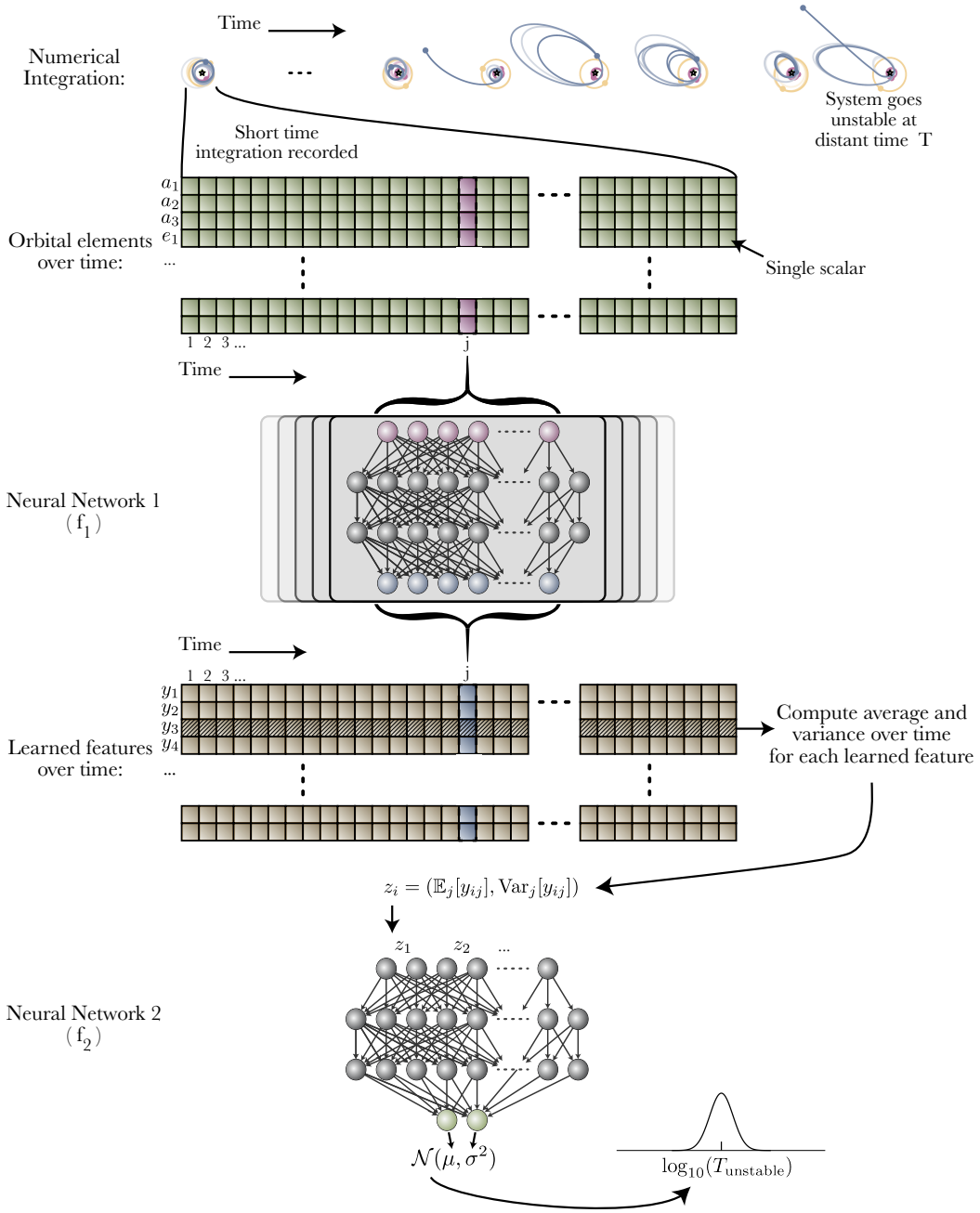

Fig. 1. Schematic of our model. The model numerically integrates 10,000 orbits for a compact 3-planet system (top) and records orbital elements at 100 times. Neural network f_1 creates learned summary features from these elements at each time. Neural network f_2 takes the average and variance of these features as input and estimates a distribution over possible instability times.

$P(\theta)$ is a prior on the neural network's parameters. The per-parameter prior here is unimportant: what matters is the prior *induced on the output of the model*, and we use an uninformative Gaussian prior on parameters to induce an uninformative prior on the output. See (40) for a detailed discussion of priors in Bayesian deep learning.

Bayesian neural network implementation. By having our model predict a distribution of instability times with a finite width, we account for intrinsic uncertainty (sometimes referred to as "aleatoric" uncertainty). However, we also wish to include extrapolation uncertainty (or "epistemic" uncertainty) for systems that differ from those found in the training set. To do this we marginalize over potential model parameters, with what is referred to as a "Bayesian Neural Network" (BNN). This is a neural network whose parameters are distributions rather than point values, and is trained with Bayesian inference. These models naturally fold in extrapolation uncertainty via marginalization.

This concept is familiar in traditional statistical inference, where one can marginalize out the internal nuisance parameters of a model using Markov Chain Monte Carlo (MCMC) techniques. The fact that neural networks typically have millions of parameters renders MCMC computationally prohibitive, and various practical simplifications are adopted for implementing a BNN. The most common strategy is Monte Carlo dropout (41, 42) which treats the neural network's parameters as independent Bernoulli random variables, and has been used in several astronomical applications (43–46). Other techniques include Bayes by Backprop (47), and Bayes by Hypernet (48, 49). One recently-proposed strategy, named "MultiSWAG" (50, 51) learns a distribution over the posterior of weights that best fit the training set, without a diagonal covariance assumption, and is much closer to standard MCMC inference. We experimented with these various techniques, and found that "MultiSWAG" produced the best accuracy and uncertainty estimations in a 5-fold cross-validation.

To move beyond a single best-fit set of parameters θ, SWAG, or "Stochastic Weight Averaging Gaussian" (50, 52), instead fits a Gaussian to a mode of the posterior over θ, assuming a low-rank covariance matrix. This was extended in (51) to "MultiSWAG," which repeats this process for several modes of the weight posterior, to help fill out the highly degenerate parameter space. This technique is summarized below:

1. Train f_1 and f_2 simultaneously via stochastic gradient descent until the parameters settle into a minimum of the weight posterior.

2. Increase the learning rate and continue training. This causes the optimizer to take a random walk in parameter space near the minima, which is assumed to look like a high-dimensional Gaussian.

3. Accumulate the average parameters along this random walk as well as a low-rank approximation of the covariance matrix.

4. The average parameters not only provide better generalization performance (stochastic weight averaging or SWA), but we have additionally fit a Gaussian to a mode of the parameter posterior. We can thus sample weights from this Gaussian to marginalize over parameters. This is SWAG (50).

5. The next step is to repeat this process from a different random initialization of the parameters. This will find another mode of the parameter posterior.

6. Fit ~30 different modes of the parameter space. We can then sample weights from multiple modes of the parameter posterior, which gives us a more rigorous uncertainty estimates. This is MultiSWAG (51).

Training a neural network through stochastic gradient descent is approximately the same as Bayesian sampling of the weights (53, 54), so this aforementioned process allows one to learn a Bayesian posterior over the weights of a neural network $P_{\text{MultiSWAG}}(\theta)$.

Once we have learned $P_{\text{MultiSWAG}}$, we can draw from it to sample a set of network weights θ. This model then predicts a lognormal distribution of instability times with mean μ and variance σ^2 for the given input orbital configuration, from which we can finally sample a log instability time T. We can write a forward model for this prediction as follows:

$$(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \sim P_{\text{MultiSWAG}}(\theta), \quad [1]$$

$$\mathbf{y}_t = f_1(\mathbf{x}_t; \boldsymbol{\theta}_1) \quad \text{for all} \quad \mathbf{x}_t \equiv X_{:,t}, \quad [2]$$

$$\mathbf{z} \sim (\mathbb{E}_t[\mathbf{y}_t], \text{Var}_t[\mathbf{y}_t]), \quad [3]$$

$$(\mu, \sigma^2) = f_2(\mathbf{z}; \boldsymbol{\theta}_2), \quad [4]$$

$$T_{\text{instability}} = 10^T \quad \text{for} \quad T \sim \mathcal{N}(\mu, \sigma^2), \quad [5]$$

where t is a timestep from 1 to 100. Here, we have labeled $\mathbf{y}_t$ as the learned transformed variables for a single timestep of the system (brown cells in fig. 1), and $\mathbf{z}$ as the average and variance of these transformed variables over time. To account for statistical errors due to our finite number of time series samples, we sample the $\mathbf{z}$ from normal distributions with frequentist estimates of the variance in the sample mean and variance: $\frac{\text{Var}_t[\mathbf{y}_t]}{n_t}$ and $\frac{2\text{Var}_t[\mathbf{y}_t]^2}{(n_t-1)}$, respectively. A Bayesian graphical model for this is shown in *Materials and Methods*. Repeatedly sampling in this way provides a predicted distribution of T given the input orbital configuration, marginalized over the posterior distribution of network weights θ.

We train our model on 60% of our $\approx 100,000$ training examples of resonant and near resonant systems, and validate it on half of the remaining data to tune the hyperparameters. Hyperparameters for our model are given in the supplementary material, and we also release the code to train and evaluate our model.

With this trained model, we then explore its performance on the remaining 20% holdout data from the resonant dataset, as well as other datasets described below.

Results

Resonant test dataset. For a given orbital configuration, our probabilistic model produces one sample of T. If a given sample is above $T = 9$, we treat the sample as a "stable" prediction. Since we are unable to make specific time predictions above the maximum integration time in our training dataset of $T = 9$, we re-sample from a user-defined prior $P(T|T \geq 9)$ for each occurrence. For the purposes of this study, we assume a simple analytic form for this prior, though followup work on this prior is ongoing (see supplementary).

For all results, we sample 10,000 predicted values of the posterior over T per planetary system. We compare our predictions against several alternatives which are explained below. Since the models we compare against can only produce point estimates while our model predicts a distribution, we take the median of our model's predicted posterior over T. This is used for plotting points, as well as for computing root-mean-square prediction errors.

We first compute the N-body versus predicted (median) T value over the holdout test dataset of $\approx 20,000$ examples not seen during training, which can be seen in the bottom middle panel of fig. 2. We reiterate that the N-body instability times measured for the various

orbital configurations in our training set are not 'true' instability times, but rather represent single draws from the different planetary systems' respective instability time distributions, established by their chaotic dynamics. To estimate a theoretical limit (bottom right panel of fig. 2) we use the results from (32), who find that the T values measured by N-body integrations (x-axis of fig. 2) should be approximately normally distributed around the mean instability time predicted by an ideal model. We use a random standard deviation drawn from the values measured empirically for compact systems by (32), which they find are sharply peaked around ≈ 0.43 dex, independent of whether or not the system is near MMRs, and valid across a wide range of mean instability times. We plot this representative intrinsic width of 0.43 dex as dotted lines on all panels for comparison.

While we defer a detailed comparison to previous work to the following section, we measure a root mean square error (RMSE) of 1.02 dex for our model on the holdout test set. We note that while the RMSE is an intuitive metric for comparing models, it does not provide a full picture for a model that is trained on a different loss function to predict both μ and σ^2, A model that can predict its own σ^2 will sacrifice worse μ accuracy in challenging regions of parameter space to better predict it on more easily predictable configurations. For comparison, if we weight the RMSE by the predicted signal-to-noise ratio (SNR), μ^2/σ^2, the model achieves 0.87 dex, within a factor of ≈ 2 of the theoretical limit. These uncertainties provide confidence estimates in the predicted values, and can indicate to a user when to invest in a computationally costly direct integration. We apply transparency to our predictions in fig. 2 according to the model-predicted SNR, highlighting that the poorest predictions were typically deemed uncertain by the model.

Finally we quantitatively test whether the model-predicted uncertainties σ accurately capture the spread of N-body times around the predicted mean values μ. For each test configuration, we predict μ, subtract it from its respective T measured by N-body integration, and divide by the predicted σ. If this distribution approximates a Gaussian distribution of zero mean and unit variance, the model's uncertainty estimates are accurate. We find that a Komolgorov-Smirnov test cannot confidently distinguish our predictions from this ideal Gaussian (p-value of 0.056), and plot the two distributions in **??** of the supplementary material.

Comparison to Previous Work. Guided by the dynamical intuition that short-timescale instabilities are driven by the interaction of MMRs (5, 8, 11), we chose to train our model on systems with particular period ratios and orbital elements in the narrow ranges near such resonances where the dynamical behavior changes sharply (39). It is therefore important to test how well such a model generalizes to a more uniform coverage of parameter space, given that most observed orbital architectures are not in MMRs (possibly because such configurations typically have short lifetimes and have been eliminated). Additionally, previous work has typically ignored the sharp variations near MMRs to fit overall trends in instability times (39), so a test on resonant systems would not provide a fair comparison.

For this generalization test and comparison, we use the 'random' dataset of (11), 25,000 three-planet systems with the same mass ratio and inclination distributions as above, and eccentricities drawn log-uniformly from $\approx 10^{-3}$ to orbit-crossing. Rather than drawing near-integer period ratios as in our resonant training set, the spacing between adjacent planets is drawn uniformly between [3.5, 30] mutual Hill radii (see 11).

We find that our model exhibits only a minor loss in performance (1.20 vs 1.02 dex RMSE) generalizing to this uniform distribution of orbital configurations table 1. This supports the assertion that instabilities in compact systems within 10^9 orbits are dominantly driven by the MMRs we focused on in our training sample (11). To compare our results to the extensive set of past efforts, we divide previous approaches into three broad groups.

Resonant				
Model	RMSE	Classif. accur.	Bias† (4, 5)	Bias (8, 9)
Obertas et al. (2017)	2.12	0.628	1.04	-1.71
Petit et al. (2020)	3.22	0.530	3.99	0.54
Tamayo et al. (2020)	1.48	0.946	2.07	-0.62
Modified* Tamayo+20	0.99	0.946	0.65	-0.60
Ours	1.02	0.952	0.29	-0.38
Ours, SNR-weighted	0.87	0.971	0.18	-0.25
Theoretical limit	*0.43*	*0.992*	*0.05*	*-0.04*
Random				
Obertas et al. (2017)	2.41	0.721	2.15	-0.93
Petit et al. (2020)	3.09	0.517	4.17	0.50
Tamayo et al. (2020)	1.24	0.949	1.16	-0.59
Modified* Tamayo+20	1.14	0.945	0.79	-0.70
Ours	1.20	0.939	0.40	-0.51
Ours, SNR-weighted	1.09	0.959	0.23	-0.49
Theoretical limit	*0.44*	*0.989*	*0.06*	*-0.04*
	(dex)	(AUC)	(dex)	(dex)

†Average difference between predicted minus true T in given range
*Modified and re-trained for regression.

Table 1. Statistical summaries of each estimator applied to a holdout test portion of the resonant dataset, and all of the random dataset.

First, many authors have run N-body integrations along low-dimensional cuts through the parameter space of compact orbital configurations, and fit simple functional forms to the resulting trends in instability times. For example, several studies have highlighted the steep dependence on interplanetary separation, while fixing orbits to be coplanar and initially circular, and planets to be equal mass and equally separated from one another (12, 14, 16, 17, 39). We compare the performance of the fit from such a study (39), using five equally spaced Earth-mass planets (mass ratio $\approx 3 \times 10^{-6}$) on our random test set in the top left panel of fig. 2, with a resulting RMSE of 2.41. Follow-up studies have incorporated the effect of finite inclinations and eccentricities (13, 15, 55, 56), but consider equal initial eccentricities, planetary masses etc. in order to fit simple models. We conclude that while such controlled experiments yield insight into the underlying dynamics (9, 15, 57), instability times depend sensitively on masses and several orbital parameters, rendering empirical fits to low-dimensional cuts in the parameter space of limited applicability. In the following section we perform the converse generalization test, where we ask our model to instead predict on the N-body simulations used for the fit by (39), and find good agreement.

Second, previous authors have developed analytical instability time estimates from first principles. These have been most successful in the limit of initially circular orbits, where three-body MMRs have been identified as the dominant driver of chaos (9). Recent work (10) has extended this theory to provide accurate instability time estimates. We will compare the predictions of our model to this limit of initially circular orbits in the next section. Here we simply emphasize the point by (10) that such predictions perform poorly at finite eccentricities (top middle panel of fig. 2), likely due to the dominant effects of stronger two-body MMRs. The fact that the analytic model predicts

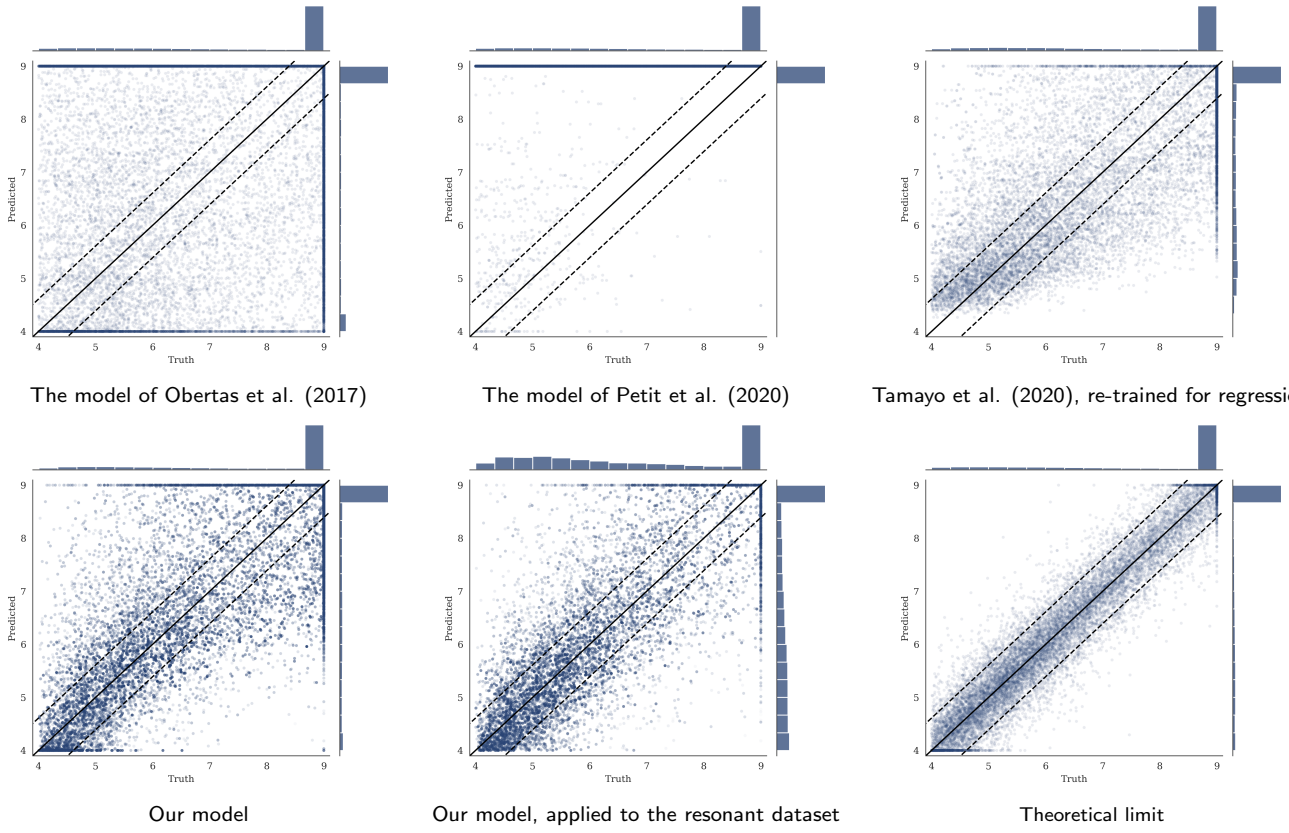

Fig. 2. Density plots showing the predicted versus true instability times for various models on the random dataset. All predictions outside of $T \in [4, 9]$ are moved to the edge of the range. For the plot showing the predictions of our model, transparency shows relative model-predicted SNR. The theoretical limit, using the numerical experiments of (32), is given in the lower-right plot. The 0.43 dex RMSE average from this is used to give the dotted contours in all plots.

the vast majority of systems to be stable implies that most of our test configurations would be stable on circular orbits, but that finite orbital eccentricities strongly modulate instability times.

The final approach is to make predictions across the high-dimensional space of orbital configurations using machine learning (11, 26). We consider two variants of (11) adapted for regression. The first, labeled 'Tamayo et al. (2020)', is to simply use an identical model as (11), but map the probability estimates of stability past 10^9 orbits through an inverse cumulative distribution of a log-normal with an optimized constant standard deviation. The second, labeled 'Modified Tamayo+20', the model is an XGBoost regression model (rather than classification) re-trained on the same features as used in (11).

We find that our model achieves similar performance to the Modified Tamayo+20 variant (top right panel of fig. 2, table 1), though the latter exhibits significant bias. We quantify this bias for each model in the range $T \in (4, 5)$ and $T \in (8, 9)$. As is evident in table 1 as well as fig. 2, the model introduced in this work exhibits significantly reduced bias compared to other models. Including SNR weighting further reduces bias. Bias is a measure of the generalizability of a model to out-of-distribution data (see chapter 7 of 58), so is an important metric for understanding how these predictive models will extrapolate to new data. Our model achieves predictions that that are more than two orders of magnitude more accurate than the analytic models in each case: e.g., $10^{2.41/1.09} \approx 162\times$ when comparing our SNR-weighted model with (39) on the random dataset.

Finally, we can make a comparison to the original classification model of (11) by using our regression model as a form of classifier. We count the the fraction of samples above $T = 9$ as the probability a given system is stable, and measure the performance of the classifier with the area under the receiver operating characteristic curve (AUC ROC) for a range of threshold probabilities for stability (table 1).

5-planet generalization with comparison. As a second generalization test of our model, we compare its performance on the limiting case considered by (39). This case of five equally spaced, Earth-mass planets on initially circular and coplanar orbits differs significantly from our training set of resonant and near resonant, eccentric and inclined configurations of three planets with unequal masses. This dataset contains 17500 simulations numerically integrated for 10^{10} orbits (39). This generalization to a limiting set of higher multiplicity configurations provides a stringent test of whether the model has learned features of the dynamics or whether it is naively interpolating across the distribution of orbital configurations present in our training dataset.

To extend our three-planet predictions to higher multiplicity systems, we perform the same short integration for all planets, but pass time series for each adjacent trio of planets to the model separately. The model samples a single instability time for each adjacent trio, and the minimum across this set is adopted as the instability time for the system, as an estimate of the time for the first trio to go unstable. This procedure is then repeated, and we record the median and confidence intervals of the resultant distribution in T. Such a reduction of compact multi-planet systems to sets of adjacent trios has been proposed on theoretical (9, 10) as well as empirical (11) grounds. This is motivated by the fact that the perturbative effects of planets on one another fall off exponentially with separation (9, 10), so non-adjacent interactions

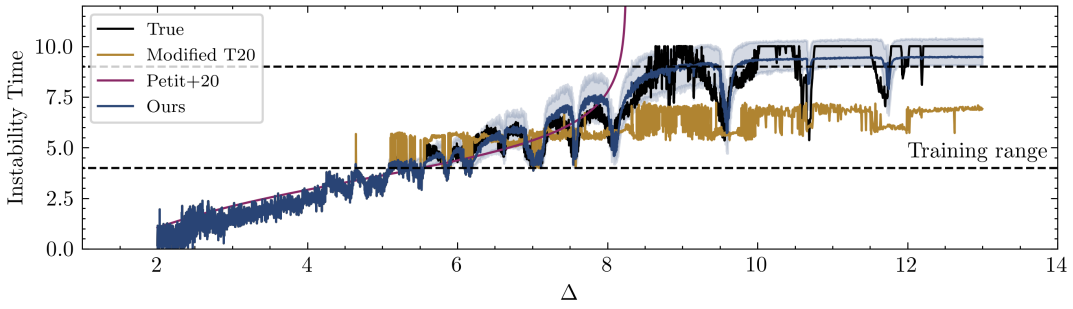

Fig. 3. The median instability time predictions of our model for the five-planet systems used in (39). These systems have fixed interplanetary separation between each body, which is labeled on the x-axis. Errorbars fill out the 68% confidence interval. The predictions from (10) and (11) are overplotted. Residuals are shown in the supplementary material.

can largely be ignored.

The predictions can be seen in fig. 3, and are remarkably accurate despite our model never seeing a system with five planets during training. We overplot the analytical result of (10) in magenta, developed from first principles for such cases with initially circular orbits, including a manual correction for five-planet systems. Our model captures the same overall trend, but additionally identifies the various dips, which correspond to locations of different MMRs (39). We emphasize that our model was trained on the general eccentric case where the magenta model of (10) does not apply (fig. 2), yet the generalization to this limiting case is excellent. In addition to matching the overall trend of (10), our model captures the additional instability time modulations at MMRs, as can be seen more clearly in the residuals in **??** of *Materials and Methods*. Additionally, our new model generalizes much better than the predictions of the modified regression model based on (11) based on manually engineered features (gold).

Interpretation. In industry machine learning, one is focused on making predictions as accurate as possible, even at the expense of a more interpretable model. However, in physics, we are fundamentally interested in understanding problems from first principles.

Obtaining such an explicit interpretation of our model will be difficult. However, as a first step we consider the feature importances of our model: what orbital elements is it using to make predictions, and does this align with expectations? To do this feature analysis, we exploit the differentiability of our model with respect to its input. We calculate the gradient of the predicted μ value our model with respect to the input instantaneous orbital elements; this is also referred to as a "saliency map" (24). This gives us a multi-dimensional array over feature, simulation, time step, and model, representing how much the predicted μ value will change should that feature be infinitesimally increased. We compute the variance of the gradients over time and each simulation, and then average these variances over sampled network weights θ. This gives us a coarse representation of the importance of each feature. We show this visually in fig. 4.

To compare these importances to other work, (11) argue empirically that the short-timescale instabilities we probe here in compact systems are driven specifically by the interactions between MMRs. A classical result of celestial mechanics is that in the absence of such MMRs, the long-term dynamics keeps the semimajor axes fixed. Variations in the semimajor axes during the short integrations thus act as a proxy for the importance of nearby MMRs (26), and we see that indeed, the semimajor axes exhibit the highest feature importance in our model fig. 4. The fact that the model ascribes comparable feature importances to any given orbital element for each of the three planets also suggests

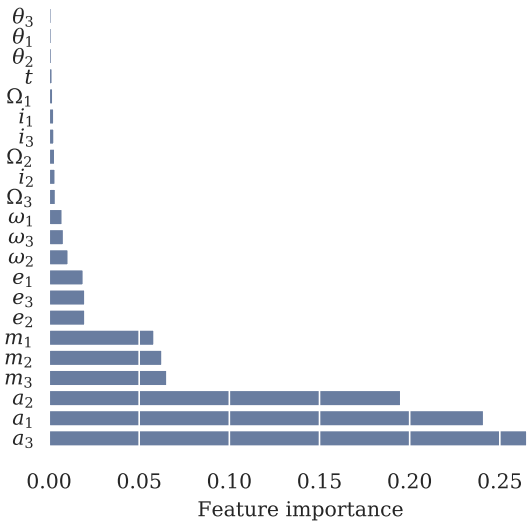

Fig. 4. Feature importances in the model, calculated as the root-mean-square of the gradients of the model's output with respect to the input features, and normalized.

a physically reasonable model.

We note that there is a small but nonzero significance of the "Time" instantaneous feature. This can be interpreted as being important because the model takes the first 10^4 orbits as input, and can predict instability for the system as low as $10^{4.1}$. Thus, the orbital parameters given at 10^4 orbits may be more important than the orbital parameters at 10^0 orbits for predicting such unstable systems, and thus the time feature is used. The time feature would be less important for a system that goes unstable near 10^9 orbits, as the relative importance of the system's parameters at 10^4 orbits is comparable to that at 10^0 orbit.

Because we chose to structure our model to take means and variances of the times series of the learned transformed variables **??**, it may be possible to extract explicit interpretations of our model via symbolic regression. Given that our approach is structurally similar to that of a graph neural network (59), the frameworks of (60–62) would be particularly applicable. This would be done by finding analytic forms for f_1, representing each of the transformed variables, and then finding an analytic form for f_2, to compute the instability time given the transformed variables. This type of explicit analysis of the model will be considered in future work.

Conclusion

We have described a probabilistic machine learning model—a Bayesian neural network—that can accurately predict a distribution over possible instability times for a given compact multi-planet exoplanet system. Our model is trained on the raw orbital parameters of a multi-planet system over a short integration, and learns its own instability metrics. This is contrasted by previous machine learning approaches which have given their models hand-designed instability metrics based on specialized domain knowledge.

Our model is more than two orders of magnitude more accurate at predicting instability times than analytical estimators, while also reducing the bias of existing learned models by nearly a factor of three. We also demonstrate that our model generalizes robustly to five-planet configurations effectively drawn from a one-dimensional cut through the broad parameter space used to train the model. This improves on the estimates of analytic and other learned models, despite our model only being trained on compact three-planet systems.

Our model's computational speedup over N-body integrations by a factor of up to 10^5 enables a broad range of applications, such as using stability constraints to rule out unphysical configurations and constrain orbital parameters (20), and to develop efficient terrestrial planet formation models. Toward this end, our model will be made publicly available through the `SPOCK`‡ package.

Acknowledgements.

Miles Cranmer would like to thank Andrew Gordon Wilson and Dan Foreman-Mackey for advice on Bayesian deep learning techniques, and Andrew Gordon Wilson, Dan Foreman-Mackey, and Sebastian Wagner-Carena for comments on a draft of this paper. Philip Armitage, Shirley Ho, and David Spergel's work is supported by the Simons Foundation. This work made use of several Python software packages: `numpy` (63), `scipy` (64), `sklearn` (65), `jupyter` (66), `matplotlib` (67), `pandas` (68), `torch` (69), and `tensorflow` (70).

1. E Kokubo, S Ida, Dynamics and accretion of planetesimals. *Prog. Theor. Exp. Phys.* **2012**, 01A308 (2012).
2. K Volk, B Gladman, Consolidating and crushing exoplanets: Did it happen here? *The Astrophys. J. Lett.* **806**, L26 (2015).
3. B Pu, Y Wu, Spacing of kepler planets: sculpting by dynamical instability. *The Astrophys. J.* **807**, 44 (2015).
4. S Tremaine, The Statistical Mechanics of Planet Orbits. *Astrophys. J.* **807**, 157 (2015).
5. J Wisdom, The resonance overlap criterion and the onset of stochastic behavior in the restricted three-body problem. *The Astron. J.* **85**, 1122–1133 (1980).
6. KM Deck, M Payne, MJ Holman, First-order Resonance Overlap and the Stability of Close Two-planet Systems. *Astrophys. J.* **774**, 129 (2013).
7. AC Petit, J Laskar, G Boué, Hill stability in the AMD framework. *Astron. & Astrophys.* **617**, A93 (2018).
8. S Hadden, Y Lithwick, A criterion for the onset of chaos in systems of two eccentric planets. *The Astron. J.* **156**, 95 (2018).
9. AC Quillen, Three-body resonance overlap in closely spaced multiple-planet systems. *Mon. Notices Royal Astron. Soc.* **418**, 1043–1054 (2011).
10. AC Petit, G Pichierri, MB Davies, A Johansen, The path to instability in compact multi-planetary systems. *Astron. & Astrophys.* **641**, A176 (2020).
11. D Tamayo, et al., Predicting the long-term stability of compact multiplanet systems. *Proc. Natl. Acad. Sci.* **117**, 18194–18205 (2020).
12. JE Chambers, GW Wetherill, AP Boss, The Stability of Multi-Planet Systems. *Icarus* **119**, 261–268 (1996).
13. K Yoshinaga, E Kokubo, J Makino, The stability of protoplanet systems. *Icarus* **139**, 328–335 (1999).
14. F Marzari, SJ Weidenschilling, Eccentric Extrasolar Planets: The Jumping Jupiter Model. *Icarus* **156**, 570–579 (2002).
15. JL Zhou, DN Lin, YS Sun, Post-oligarchic evolution of protoplanetary embryos and the stability of planetary systems. *The Astrophys. J.* **666**, 423 (2007).
16. P Faber, AC Quillen, The total number of giant planets in debris discs with central clearings. *Mon. Notices Royal Astron. Soc.* **382**, 1823–1828 (2007).
17. AW Smith, JJ Lissauer, Orbital stability of systems of closely-spaced planets. *Icarus* **201**, 381–394 (2009).
18. BV Chirikov, A universal instability of many-dimensional oscillator systems. *Phys. reports* **52**, 263–379 (1979).
19. S Hadden, An Integrable Model for the Dynamics of Planetary Mean-motion Resonances. *Astron. J.* **158**, 238 (2019).
20. D Tamayo, C Gilbertson, D Foreman-Mackey, Stability constrained characterization of multiplanet systems (2020).
21. A Krizhevsky, I Sutskever, GE Hinton, Imagenet classification with deep convolutional neural networks in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12. (Curran Associates Inc., Red Hook, NY, USA), p. 1097–1105 (2012).
22. A Graves, Ar Mohamed, G Hinton, Speech recognition with deep recurrent neural networks in *2013 IEEE international conference on acoustics, speech and signal processing*. (IEEE), pp. 6645–6649 (2013).
23. I Sutskever, O Vinyals, QV Le, Sequence to sequence learning with neural networks in *Advances in neural information processing systems*. pp. 3104–3112 (2014).
24. I Goodfellow, Y Bengio, A Courville, *Deep Learning*. (MIT Press), (2016).
25. DG Lowe, Object recognition from local scale-invariant features in *Proceedings of the seventh IEEE international conference on computer vision*. (Ieee), Vol. 2, pp. 1150–1157 (1999).
26. D Tamayo, et al., A machine learns to predict the stability of tightly packed planetary systems. *The Astrophys. J. Lett.* **832**, L22 (2016).
27. E Agol, et al., Refining the transit timing and photometric analysis of trappist-1: Masses, radii, densities, dynamics, and ephemerides (2020).
28. LM Weiss, et al., The california-kepler survey. v. peas in a pod: planets in a kepler multi-planet system are similar in size and regularly spaced. *The Astron. J.* **155**, 48 (2018).
29. H Rein, D Tamayo, whfast: a fast and unbiased implementation of a symplectic wisdom–holman integrator for long-term gravitational simulations. *Mon. Notices Royal Astron. Soc.* **452**, 376–388 (2015).
30. H Rein, SF Liu, Rebound: an open-source multi-purpose n-body code for collisional dynamics. *Astron. & Astrophys.* **537**, A128 (2012).
31. DR Rice, FA Rasio, JH Steffen, Survival of non-coplanar, closely packed planetary systems after a close encounter. *Mon. Notices Royal Astron. Soc.* **481**, 2205–2212 (2018).
32. N Hussain, D Tamayo, Fundamental limits from chaos on instability time predictions in compact planetary systems. *Mon. Notices Royal Astron. Soc.* **491**, 5258–5267 (2019).
33. A Waibel, T Hanazawa, G Hinton, K Shikano, KJ Lang, Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, signal processing* **37**, 328–339 (1989).
34. Y LeCun, et al., Backpropagation applied to handwritten zip code recognition. *Neural computation* **1**, 541–551 (1989).
35. W Rawat, Z Wang, Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* **29**, 2352–2449 (2017).
36. S Hochreiter, J Schmidhuber, Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).
37. M Andreux, et al., Kymatio: Scattering Transforms in Python (2018).
38. CE Rasmussen, CKI Williams, *Gaussian Processes for Machine Learning*. (2006).
39. A Obertas, C Van Laerhoven, D Tamayo, The stability of tightly-packed, evenly-spaced systems of earth-mass planets orbiting a sun-like star. *Icarus* **293**, 52–58 (2017).
40. AG Wilson, The case for bayesian deep learning (2020).
41. Y Gal, Z Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (2015).
42. Y Gal, J Hron, A Kendall, Concrete Dropout (2017).
43. YD Hezaveh, L Perreault Levasseur, PJ Marshall, Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature* **548**, 555–557 (2017).
44. L Perreault Levasseur, YD Hezaveh, RH Wechsler, Uncertainties in Parameters Estimated with Neural Networks: Application to Strong Gravitational Lensing. *Astrophys. Journal, Lett.* **850**, L7 (2017).
45. HW Leung, J Bovy, Deep learning of multi-element abundances from high-resolution spectroscopic data. *Mon. Notices RAS* **483**, 3255–3277 (2019).
46. S Wagner-Carena, et al., Hierarchical Inference With Bayesian Neural Networks: An Application to Strong Gravitational Lensing (2020).
47. C Blundell, J Cornebise, K Kavukcuoglu, D Wierstra, Weight Uncertainty in Neural Networks (2015).
48. N Pawlowski, A Brock, MCH Lee, M Rajchl, B Glocker, Implicit Weight Uncertainty in Neural Networks (2017).
49. D Krueger, et al., Bayesian Hypernetworks (2017).
50. W Maddox, T Garipov, P Izmailov, D Vetrov, AG Wilson, A Simple Baseline for Bayesian Uncertainty in Deep Learning (2019).
51. AG Wilson, P Izmailov, Bayesian Deep Learning and a Probabilistic Perspective of Generalization (2020).
52. P Izmailov, D Podoprikhin, T Garipov, D Vetrov, AG Wilson, Averaging Weights Leads to Wider Optima and Better Generalization (2018).
53. S Mandt, MD Hoffman, DM Blei, Stochastic gradient descent as approximate bayesian inference. *The J. Mach. Learn. Res.* **18**, 4873–4907 (2017).
54. C Mingard, G Valle-Pérez, J Skalse, AA Louis, Is SGD a Bayesian sampler? Well, almost (2020).
55. B Funk, G Wuchterl, R Schwarz, E Pilat-Lohinger, S Eggl, The stability of ultra-compact planetary systems. *Astron. & Astrophys.* **516**, A82 (2010).
56. DH Wu, RC Zhang, JL Zhou, JH Steffen, Dynamical instability and its implications for planetary system architecture. *Mon. Notices Royal Astron. Soc.* **484**, 1538–1548 (2019).
57. A Yalinewich, C Petrovich, Nekhoroshev estimates for the survival time of tightly packed planetary systems (2019).
58. T Hastie, R Tibshirani, J Friedman, *The elements of statistical learning: data mining, inference, and prediction*. (Springer Science & Business Media), (2009).
59. PW Battaglia, et al., Relational inductive biases, deep learning, and graph networks (2018).
60. MD Cranmer, R Xu, P Battaglia, S Ho, Learning Symbolic Physics with Graph Networks (2019).

‡https://github.com/dtamayo/spock

61. M Cranmer, et al., Discovering Symbolic Models from Deep Learning with Inductive Biases (2020).
62. M Cranmer, PySR: Fast & Parallelized Symbolic Regression in Python/Julia (2020).
63. CR Harris, et al., Array programming with NumPy. *Nature* **585**, 357–362 (2020).
64. P Virtanen, et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272 (2020).
65. F Pedregosa, et al., Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
66. T Kluyver, et al., Jupyter Notebooks – a publishing format for reproducible computational workflows in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, eds. F Loizides, B Schmidt. (IOS Press), pp. 87 – 90 (2016).
67. M Droettboom, et al., matplotlib: matplotlib v1.5.1 (2016).
68. Wes McKinney, Data Structures for Statistical Computing in Python in *Proceedings of the 9th Python in Science Conference*, eds. Stéfan van der Walt, Jarrod Millman. pp. 56 – 61 (2010).
69. A Paszke, et al., PyTorch: An Imperative Style, High-Performance Deep Learning Library in *Advances in Neural Information Processing Systems 32*, eds. H Wallach, et al. (Curran Associates, Inc.), pp. 8024–8035 (2019).
70. M Abadi, et al., Tensorflow: A system for large-scale machine learning in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*. pp. 265–283 (2016).

Materials and Methods