



Accelerating Giant-impact Simulations with Machine Learning

Caleb Lammers¹ , Miles Cranmer^{2,3,4} , Sam Hadden⁵ , Shirley Ho^{1,6,7} , Norman Murray^{5,8} , and Daniel Tamayo⁹ ¹Department of Astrophysical Sciences, Princeton University, 4 Ivy Lane, Princeton, NJ 08544, USA²Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, UK³Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge, CB3 0HA, UK⁴Kavli Institute for Cosmology, University of Cambridge, Madingley Road, Cambridge, CB3 0HA, UK⁵Canadian Institute for Theoretical Astrophysics, University of Toronto, 60 St. George Street, Toronto, ON M5S 3H8, Canada⁶Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, USA⁷Department of Physics & Center for Data Science, New York University, 726 Broadway, New York, NY 10003, USA⁸Department of Physics, University of Toronto, 60 St. George Street, Toronto, ON M5S 1A7, Canada⁹Department of Physics, Harvey Mudd College, Claremont, CA 91711, USA

Received 2024 April 19; revised 2024 September 14; accepted 2024 September 23; published 2024 November 6

Abstract

Constraining planet-formation models based on the observed exoplanet population requires generating large samples of synthetic planetary systems, which can be computationally prohibitive. A significant bottleneck is simulating the giant-impact phase, during which planetary embryos evolve gravitationally and combine to form planets, which may themselves experience later collisions. To accelerate giant-impact simulations, we present a machine learning (ML) approach to predicting collisional outcomes in multiplanet systems. Trained on more than 500,000 N -body simulations of three-planet systems, we develop an ML model that can accurately predict which two planets will experience a collision, along with the state of the postcollision planets, from a short integration of the system's initial conditions. Our model greatly improves on non-ML baselines that rely on metrics from dynamics theory, which struggle to accurately predict which pair of planets will experience a collision. By combining with a model for predicting long-term stability, we create an ML-based giant-impact emulator, which can predict the outcomes of giant-impact simulations with reasonable accuracy and a speedup of up to 4 orders of magnitude. We expect our model to enable analyses that would not otherwise be computationally feasible. As such, we release our training code, along with an easy-to-use user interface for our collision-outcome model and giant-impact emulator (<https://github.com/dtamayo/spock>).

Unified Astronomy Thesaurus concepts: Exoplanets (498); Extrasolar rocky planets (511); Planet formation (1241); Planetary dynamics (2173)

1. Introduction

The nebular hypothesis, initially proposed by I. Kant (1755) and later built upon by P.-S. Laplace (1796), remains the leading explanation for the formation of planetary systems. In the modern picture, terrestrial planets form out of a protoplanetary disk before experiencing a phase of giant impacts in which planets grow via collisions (see, e.g., J. J. Lissauer 1993). Giant impacts are therefore believed to play an important role in shaping the orbital configuration of planetary systems, and there has been significant effort to simulate this process (e.g., J. E. Chambers & G. W. Wetherill 1998; E. Kokubo & S. Ida 1998; C. B. Agnor et al. 1999; J. E. Chambers 2001; S. N. Raymond et al. 2005, 2008; E. Kokubo et al. 2006; B. M. S. Hansen & N. Murray 2012).

Interest in terrestrial planet formation has been reinvigorated by the discovery, due to NASA's Kepler mission, that compact multiplanet systems are remarkably common in the Galactic neighborhood (W. J. Borucki et al. 2011; J. J. Lissauer et al. 2011; F. Fressin et al. 2013; D. C. Fabrycky et al. 2014; W. Zhu et al. 2018). Planets in these systems typically possess short periods, sub-Neptune radii and masses, low eccentricities, and low inclinations. Motivated by the tight dynamical spacing of planets in observed systems (B. Pu & Y. Wu 2015; K. Volk

& B. Gladman 2015) and the inability of these systems to host additional intervening planets (J. Fang & J.-L. Margot 2013; A. Obertas et al. 2023), it is believed that observed systems are shaped by dynamical instabilities arising over their gigayear lifetimes. In general, close encounters between planets can result in planet–planet collisions, planet–star collisions, or ejections. However, planets in typical transiting multiplanet systems orbit with semimajor axes for which dynamical instabilities nearly always result in planet–planet collisions.

Numerical simulations of the giant-impact phase have been successful in reproducing many properties of the observed population of multiplanet systems (e.g., B. M. S. Hansen & N. Murray 2013; A. Izidoro et al. 2017; S. T. S. Poon et al. 2020; M. Goldberg & K. Batygin 2022; C. Lammers et al. 2023; T. Ghosh & S. Chatterjee 2024). These simulations typically begin with a collection of overly packed planets¹⁰ and evolve forward in time using a hybrid integrator (e.g., *mercury6*, J. E. Chambers 1999; or *MERCURIUS*, H. Rein et al. 2019). Hybrid integrators make long-term integrations computationally feasible by using a rapid Wisdom–Holman integrator (J. Wisdom & M. Holman 1991) when planets are widely separated, switching to a high-precision integrator only when it is necessary to resolve a close encounter. Before reaching the maximum integration time, systems in these simulations continually destabilize, resulting in collisions

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

¹⁰ Low-mass bodies in N -body simulations are sometimes referred to as “planetary embryos” or “protoplanets.” For simplicity, throughout this work, we will refer to the bodies in our N -body as “planets” regardless of their mass.

between planets (which are typically treated as perfect inelastic mergers). Due to the chaotic nature of multiplanet systems, outcomes of such numerical experiments are only meaningful in a statistical sense, necessitating simulations of many realizations with different initial conditions.

Efforts to numerically model this process are limited by the computational cost of running large samples of long-term N -body simulations. In recent years, machine learning (ML) techniques have been successfully applied to accelerate N -body simulations, both in the cosmological (S. He et al. 2019; D. Jamieson et al. 2023) and planetary (D. Tamayo et al. 2020; M. Cranmer et al. 2021) contexts. In particular, D. Tamayo et al. (2020) (hereafter, **SPOCKI**) presented an ML model to predict whether a compact planetary system is stable over 10^9 orbits (here, and throughout the paper, the “orbits” unit refers to the initial orbital period of the innermost planet), based on summary statistics calculated over a short 10^4 orbit integration. Extending this work, M. Cranmer et al. (2021) (hereafter, **SPOCKII**) developed a model that, when provided the raw orbital elements of the planets from a 10^4 orbit integration, can accurately predict the time at which a compact planetary system will destabilize. These tools have enabled investigations into the long-term stability of planetary systems that would not otherwise be computationally feasible (e.g., D. Tamayo et al. 2021; S. W. Yee et al. 2021; N. A. Bailey & D. C. Fabrycky 2022; R. Tejada Arevalo et al. 2022; A. Obertas et al. 2023; N. Sobski & S. C. Millholland 2023). The development of **SPOCKII** was also motivated by the possibility of accelerating giant-impact simulations. Toward this end, here we present an ML-based approach to predicting the outcomes of planet–planet collisions in N -body simulations.

This paper is organized as follows: Section 2 describes our training set and ML model. We evaluate the performance of our model and compare it with non-ML approaches in Section 3. Discussion and conclusions are presented in Section 4.

2. Machine Learning Model

2.1. Training Data Set

To train our ML model, we require a large collection of N -body simulations in which two planets experience a physical collision. Unfortunately, the **SPOCKI** training data set (which was also used in **SPOCKII**) is unsuitable for this purpose due to choices made to tailor the data set for predicting long-term stability (e.g., stopping N -body integrations when the Hill spheres of two planets intersect). As a result, for this work, we generate a new training set of N -body simulations. Motivated by the intention that our model will often be used in tandem with **SPOCKII**, we sample initial conditions that span a similar region of parameter space.

As in the **SPOCK** training set, we restrict our data set to systems of three planets. This simplification is motivated by the finding that there is a qualitative change in the instability behavior between two- and three-planet systems, but little change when additional planets are included in the system (e.g., J. E. Chambers et al. 1996; A. C. Petit et al. 2020). Theoretically, this is believed to be due to the importance of mean-motion resonances between adjacent planets, and the secondary resonances they generate, in driving multiplanet systems to destabilize (A. C. Quillen 2011; A. C. Petit et al. 2020; C. Lammers et al. 2024). Because three-body interactions are believed to drive the instability process, we expect

that restricting the training set to three-planet systems is also a reasonable assumption for the task of predicting collisional outcomes.

We draw planet-to-star mass ratios for each of the three planets log-uniformly from the range $[10^{-7}, 10^{-4}]$, spanning approximately from the mass ratio of small planetary embryos to larger than that of Neptune. Orbital inclinations were sampled log-uniformly from the narrow range of $[10^{-3}$ rad, 0.3 rad] to reflect that most observed multiplanet systems are nearly coplanar (J. Fang & J.-L. Margot 2012; D. C. Fabrycky et al. 2014; M. Y. He et al. 2019). Nonetheless, this choice allows for mutual inclinations of up to 34° in the training set. Initial orbital angles (i.e., longitude of the ascending node, longitude of pericenter, and true longitude) were drawn uniformly from $[0, 2\pi]$. Eccentricities for each planet were drawn log-uniformly from $[10^{-3}, e_{\text{cross}}]$, where e_{cross} is the eccentricity at which the orbit of the planet crosses that of its neighbor,

$$e_{\text{cross}} = \frac{a_{i+1} - a_i}{a_{i+1}}, \quad (1)$$

where $i=1$ for the innermost planet and $i=2$ for the outermost planet. For the middle planet, we take e_{cross} to be the minimum eccentricity required to cross either the orbit of the inner planet or the outer planet (i.e., $e_{\text{cross}} = \min\{(a_2 - a_1)/a_2, (a_3 - a_2)/a_3\}$).

The spacing of the planets was chosen so that systems span a range of initial separations, but are nonetheless likely to destabilize. It has been shown that instability times in multiplanet systems scale with the spacing unit $S_i = (a_{i+1} - a_i)/(a_{i+1}\mu^{1/4})$ where $\mu = (m_i + m_{i+1})/M_*$ (A. C. Petit et al. 2020; C. Lammers et al. 2024). Leveraging this, we arbitrarily place the innermost planet at $a_1 = 1.00$ and sample S_1 uniformly from $[0.0, 7.5]$ and then S_2 uniformly from $[0.0, 7.5]$, to determine the semimajor axes of the outer two planets. Planet radii were assigned using the mass–radius relationship $(M/M_\oplus) = 2.7(R/R_\oplus)^{1.3}$ from A. Wolfgang et al. (2016). Note, however, that we must assume a typical value of a_1 to convert planet radii from units of R_\oplus to units in which $a_1 = 1.0$; we adopt a nominal value of $a_1 = 0.1$ au, interior to which the prevalence of super-Earths declines. We caution that the predictions of the model may be inaccurate for systems of planets with densities that differ drastically from the adopted mass–radius relationship.

N -body integrations were performed using the **MERCURIUS** integrator (H. Rein et al. 2019) from the open-source **REBOUND** code. **MERCURIUS** is a hybrid integrator, which uses a fixed-time step **Wisdom–Holman** integrator (**WHFast**; J. Wisdom & M. Holman 1991; H. Rein & D. Tamayo 2015) when the planets are widely separated and switches smoothly to a high-order adaptive integrator (**IAS15**; H. Rein & D. S. Spiegel 2015) when planets come within three Hill radii of each other. As recommended by J. Wisdom (2015), we adopt a time step of $T_p/20$, where T_p is the minimum perihelion-passage timescale of the three planets. For consistency across systems, before integrating, we align the z -axis of our coordinate system with the angular-momentum vector of the three-planet system.

As in previous works, we treat collisions between planets as perfect inelastic collisions in which mass and momentum are conserved.¹¹ We integrate systems for 10^7 orbits of the

¹¹ Adopting a more sophisticated merger prescription does not meaningfully impact the resulting orbital configurations (S. T. S. Poon et al. 2020; L. Esteves et al. 2022).

innermost planet, stopping the simulation prematurely if two planets merge. Our choice of stopping time is motivated by the typical time required for a physical merger to occur (D. R. Rice et al. 2018) and makes it computationally feasible to generate a training set of 500,000+ systems. Note that **SPOCKI** and **SPOCKII** were trained to predict stability on timescales up to 10^9 orbits. However, whereas there is a clear advantage to increasing the integration time for stability predictions, there is less of an advantage for our task; it is unlikely that mergers that occur in systems that take longer to destabilize differ meaningfully from mergers experienced by systems that destabilize more quickly.

While constructing the training data set, we reject systems that experienced a merger in less than 10^4 orbits (collision-outcome predictions are not needed for such systems because short integrations are computationally inexpensive) or survive 10^7 orbits without a merger. Restricting the training set to systems that experience mergers allows us to train a model that focuses exclusively on predicting collisional outcomes, leaving aside the challenge of predicting stability, which was the focus of **SPOCKI** and **SPOCKII**. After removing 1540 systems that experience an ejection (i.e., systems in which a planet is scattered to $a > 50a_1$), the final training set consists of 517,016 systems. Note that the distributions of initial orbital elements in the training set are affected by the requirement that systems must experience a merger between 10^4 and 10^7 orbits (e.g., compact, eccentric systems typically experience a merger in less than 10^4 orbits and are thereby excluded).

2.2. Model Description

We break up the task of predicting planet–planet collision outcomes into two subproblems: (1) predicting which pair of planets will experience a collision and (2) predicting the orbital configuration of the postcollision system. The former is a classification problem with three possibilities: planet 1 collides with planet 2 (43.3% of systems in the training set), planet 2 collides with planet 3 (37.4% of systems), or planet 1 collides with planet 3 (19.3% of systems). Predicting the orbital elements of the postcollision system, on the other hand, is a regression problem with multiple outputs. We explored different ML model architectures for tackling these tasks, but we found a shallow multilayer perceptron (MLP) model to work well for both. A schematic of our ML model, encompassing both collision classifier and orbital-outcome regressor parts, is shown in Figure 1.

To address the classification problem, we construct a model that, when provided information about the system as input, outputs three numbers: the probability of a collision occurring between planets 1–2, planets 2–3, and planets 1–3. Motivated by previous works (D. Tamayo et al. 2016; **SPOCKI**; **SPOCKII**), as input, we provide the MLP model with the mean and standard deviation of the orbital elements of the three planets over a short 10^4 orbit integration of the system. The time required to perform a 10^4 orbit integration is approximately equal to the inference time of the model, so there is no computational gain to generating inputs with a shorter integration. The classification MLP was trained to minimize binary cross-entropy loss with the Adaptive Moment Estimation Optimizer (ADAM; D. P. Kingma & J. Ba 2014) for 1,000,000 optimizer steps. The hyperparameters of the model (network depth = 1 hidden layer, number of hidden nodes = 30, learning rate = 7×10^{-4} , weight decay = 1×10^{-4} , and

minibatch size = 1000) were determined prior to training with Bayesian optimization on a pared-down version of the training set, and the final model was trained on 80% of the full training set (413,612 systems). A full description of the model architecture and training process can be found in Appendix A.1.

To address the regression problem, we require a model that can predict the postcollision orbital configuration of the system, given a short integration of the system and a choice of which two planets are to collide. Although only two of the three planets are involved in the collision, we find that the state of the third planet can also change substantially during the instability phase. As a result, we require a model that can predict the orbital parameters of the newly merged planet, along with those of the surviving planet.

In practice, we found it challenging to predict the orbital angles (longitude of the ascending node, longitude of pericenter, and true longitude) of the two postcollision planets. Further investigation reveals that, in integrations of the synthetic systems where the initial conditions have been slightly perturbed (see Section 3.1), the orbital angles of the planets typically span from approximately $[0, 2\pi]$ across the set of perturbed systems. As a result, we trained the regression model to predict only semimajor axes, eccentricities, and inclinations, and when the orbital angles are needed for a particular application of the model (e.g., replacing three planets in a synthetic planetary system with two postcollision planets), we simply draw them randomly from $[0, 2\pi]$. It may prove valuable to work toward a model that can predict dynamically relevant combinations of the orbital angles (e.g., mutual inclination) in the future.

As input, we provide the regression model with the mean and standard deviation of the orbital elements of the three planets, ordered to specify which two planets are involved in the collision. The regression MLP was trained using a mean-squared-error loss function with a correction term to account for the upper limits on the orbital-element outputs (see Appendix A.2 for more details). The model was trained on 80% of the full training set using the ADAM optimizer for a total of 1,000,000 optimizer steps. Like the classification model, hyperparameters (network depth = 1 hidden layer, number of hidden nodes = 60, learning rate = 7×10^{-4} , weight decay = 1×10^{-4} , and minibatch size = 3000) were determined with Bayesian optimization performed on a subset of the training set.

2.3. Giant-impact Emulator

Combining our collision-outcome model with the stability predictions provided by **SPOCKII** allows us to create a rapid giant-impact-simulation emulator. We outline our iterative emulator model below.

Beginning with a system of overly packed planets, the giant-impact emulator (1) partitions the system into trios of adjacent planets, (2) predicts the instability time of each trio using **SPOCKII**, and (3) predicts the merger outcome of the trio with the shortest instability time and replaces the planets. This process is repeated until the system is deemed stable by **SPOCKII** (i.e., the full planetary system has an instability time larger than a user-chosen maximum). In step (2), we determine each trio’s instability time using the median of **SPOCKII**’s instability-time samples. We find that the emulator’s predictions compare much more accurately with N -body results (see Section 3.2) when using median instability-

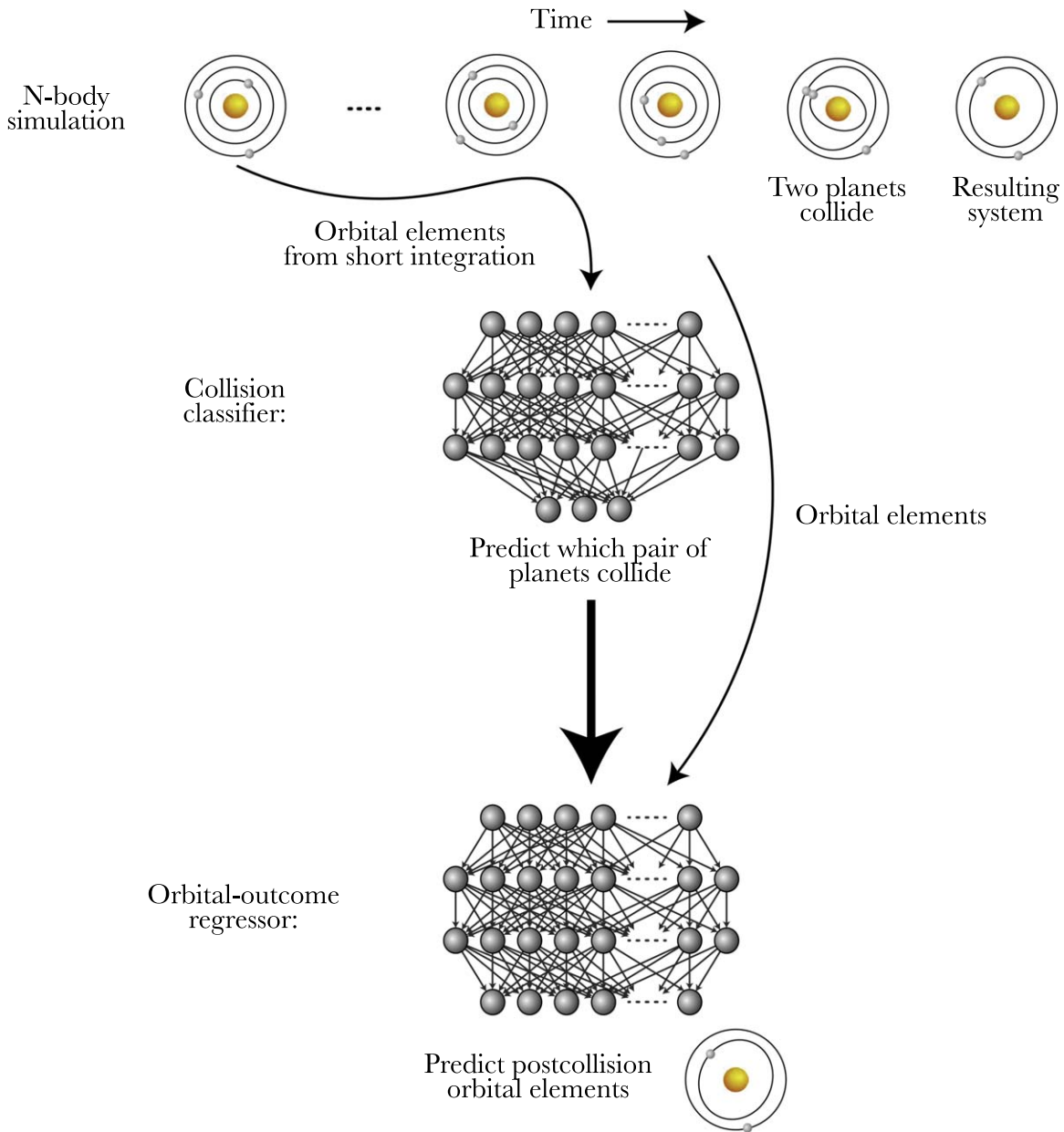


Figure 1. Schematic of our ML model. The collision classifier takes the mean and standard deviation of the three planets’s orbital elements from a short N -body integration as input and predicts which pair of planets will collide (in the form of three collision probabilities). The orbital-outcome regressor takes the orbital elements, as well as a choice for which two planets to combine, and predicts the orbital elements of the two resulting postcollision planets.

time predictions, as opposed to directly sampling **SPOCKII**’s predicted instability-time posterior. In step (3), we first predict the probabilities of a collision occurring between planets 1–2, 2–3, or 1–3 in the unstable planet trio using the collision-classifier model. Then, we sample the probabilities to determine which pair of planets to combine, and we predict the resulting orbital elements using the orbital-outcome regressor model. Results from our giant-impact emulator are presented in Section 3.2 and caveats are discussed in Section 4.

3. Results

3.1. Performance on Validation Set

In this section, we study the performance of our ML model, including both collision classifier and orbital-outcome regressor parts, on systems from the 103,404 system validation set.

Interpreting the performance of our model is complicated by the chaotic nature of multiplanet systems, which can cause systems with indistinguishable initial conditions to rapidly diverge, resulting in different late-time outcomes (e.g., D. R. Rice et al. 2018; N. Hussain & D. Tamayo 2020). Properly evaluating the performance of our model therefore requires quantifying the limits on prediction accuracy imposed by chaos.

To isolate the influence of chaos, we perform “shadow integrations” of 200 randomly selected systems from the validation set, in which we slightly perturb the initial conditions of the systems and reintegrate it. For each validation system, we construct 250 shadow realizations where we have perturbed the initial Cartesian positions and velocities of the three planets (in the barycentric frame) by random factors drawn from a Gaussian distribution $\mathcal{N}(\mu = 1.0, \sigma = 10^{-12})$.

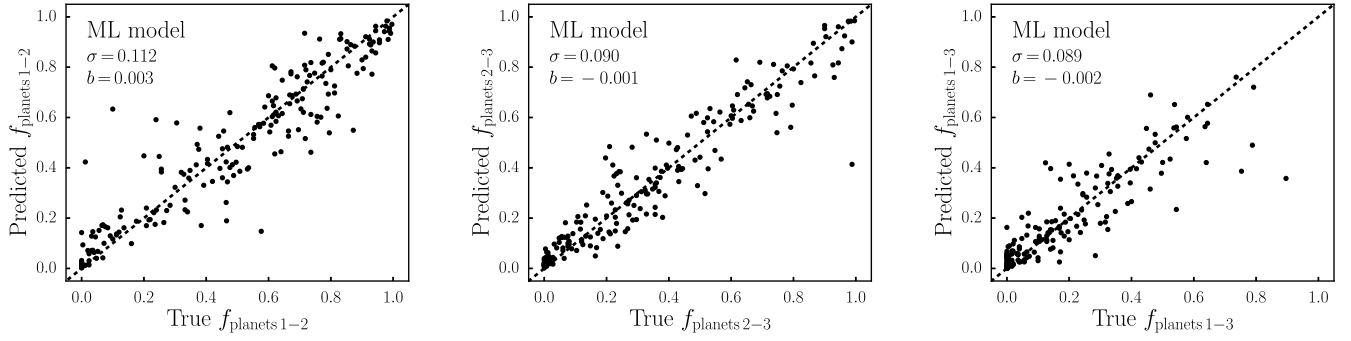


Figure 2. Performance of the collision-classifier model on 200 random validation systems. The probabilities of a collision occurring between planets 1–2, 2–3, and 1–3, as predicted by the ML model, are plotted against the true fraction of collisions that occur between the planet pairs, determined by performing 250 shadow integrations of each system. The model accurately predicts collision probabilities with a scatter of $\sim 10\%$ and very little bias about the true fractions (σ is the rms error and b is the mean of the residuals). For comparison, all baseline models we considered performed poorly (see Appendix B).

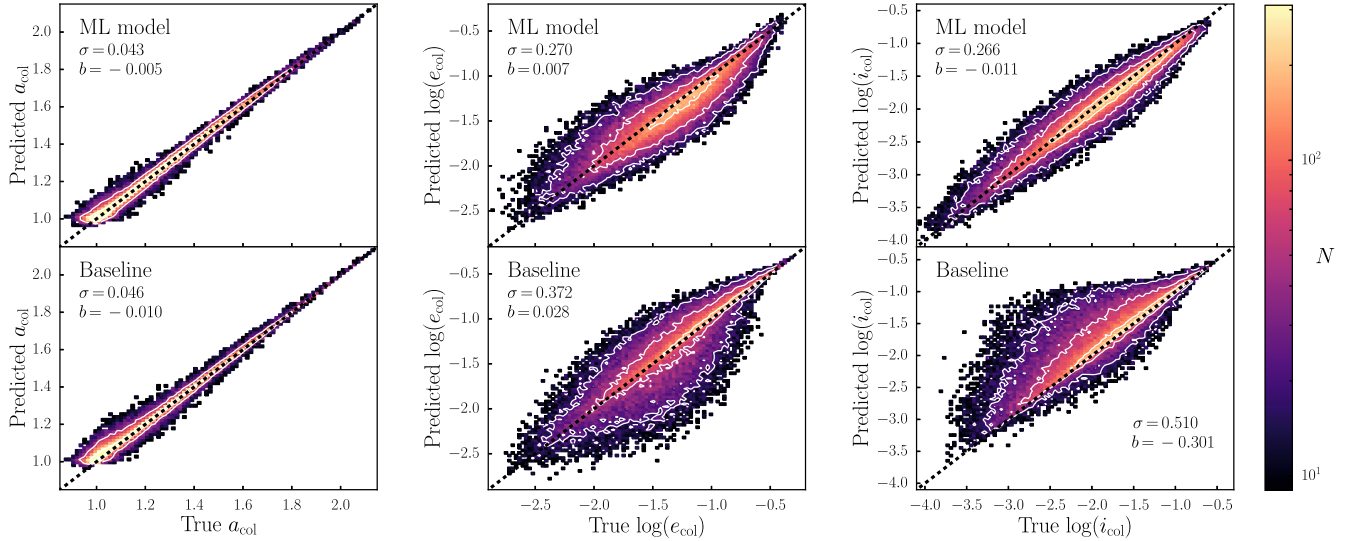


Figure 3. Comparison between the performance of the orbital-outcome regressor (top row) and a non-ML baseline model (bottom row) on the validation set of 103,404 three-planet systems. Predicted orbital elements (i.e., semimajor axis, eccentricity, and inclination) for the new, merged planets are plotted against their true orbital elements. The ML model predicts orbital elements with somewhat less scatter and bias about the true values than the baseline model, approaching the accuracy limits imposed by chaos (see Table 1).

The resulting 250 shadow systems represent equally valid possible long-term evolutions of the system, allowing us to quantify the variety of potential outcomes.

3.1.1. Collision Classifier

Applying our collision-classifier model to the 20% validation set, we find that the model correctly predicts the planet pair that experiences a collision for 66.5% of validation systems. This is an improvement over chance (33.3% accuracy) and always guessing the most common outcome (collision between planet 1 and 2; 43.3% accuracy), but further interpreting the model’s performance requires quantifying the limits on prediction accuracy imposed by chaos.

In many instances, shadow realizations of the same system ultimately experience collisions between a different planet pair. Using our set of shadow integrations, we measure the fractions $f_{\text{planets } i-j}$ of shadow systems in which a collision occurs between planets i and j , which vary widely depending on the system. In Figure 2, we compare the three collision probabilities predicted by the ML model with the three true collision fractions for each of the 200 validation systems (we report σ , the rms error, and b , the mean of the residuals). Based

on a single, short integration of the validation systems, the ML model successfully predicts the probabilities of a collision occurring between the different planet pairs, with a scatter of $\sim 10\%$ and very little bias about the true probabilities. Increasing the number of shadow runs does not decrease the scatter, indicating that it is dominated by the error in the ML model’s predictions.

To help interpret the performance of the ML model, we considered a variety of non-ML approaches to predicting the collision probabilities. All baseline models we constructed performed poorly, even when tuned to fit the true $f_{\text{planets } i-j}$ values (see Appendix B).

3.1.2. Orbital-outcome Regressor

Next, we analyze the performance of the orbital-outcome regression model. We compare the orbital elements of the newly merged planets, as predicted by the orbital regression model, with the true values for the 20% validation set in Figure 3. The ML model predictions exhibit little bias about the true values but are otherwise difficult to interpret without a baseline to compare against. The best baseline model we found for predicting the postcollision orbital elements of the merged

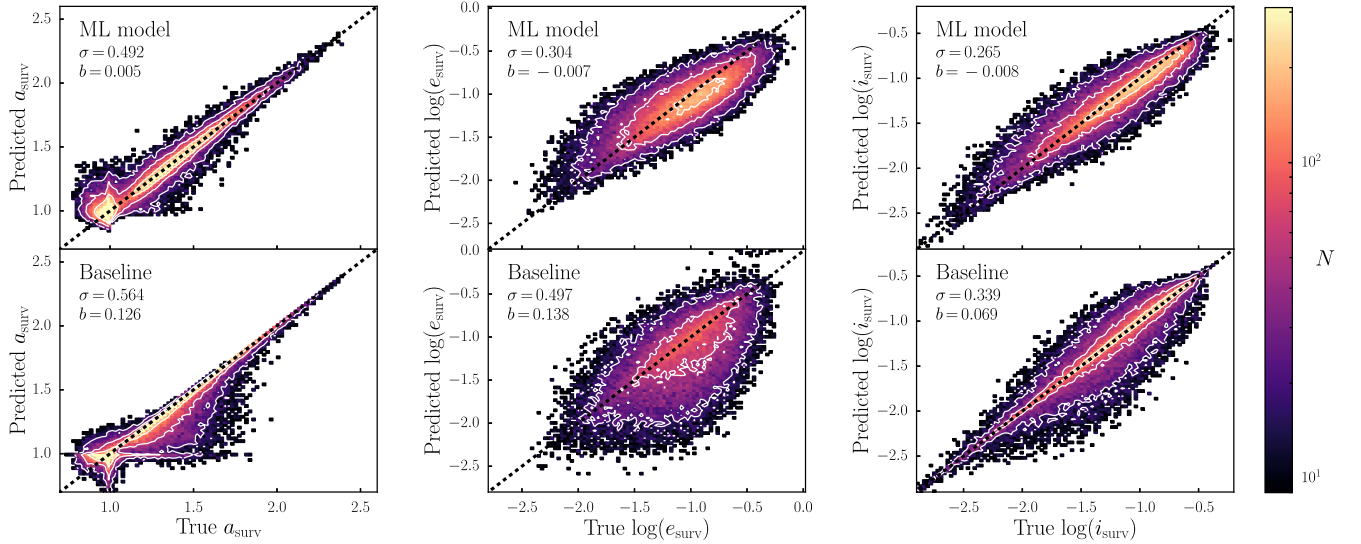


Figure 4. Same comparison as Figure 3, now for the three postcollision orbital elements of the surviving planet that is not involved in the merger. The orbital-outcome regressor predicts semimajor axis, eccentricity, and inclination with less scatter and bias than the non-ML baseline model, approaching the limit imposed by chaos (see Table 1).

planet relied on a mass-weighted average of the orbital elements of the two premerger planets over the 10^4 orbit integration. In contrast with the task of predicting collision probabilities, we find that the baseline model predicts postcollision orbital elements relatively well (see bottom panel of Figure 3). Nonetheless, our ML model improves moderately on the scatter and bias of the baseline model, particularly for the inclination of the newly merged planet.

Figure 4 shows the orbital elements of the planet that is not involved in the collision (the “surviving planet”), as predicted by our ML-based model and a baseline model, for the 20% validation set. We create a non-ML baseline model, in this case, by taking the predicted inclination to be the average inclination of the surviving planet over the 10^4 orbit integration, then we determine the planet’s semimajor axis and eccentricity by requiring the conservation of energy E and the z -component of angular momentum L_z . Note that E is not perfectly conserved during collisions, but the loss of energy during inelastic collisions in our training set simulations is small ($\sim 0.05\%$), so we may use the approximate conservation of E (and the baseline prediction of a_{col}) to predict a_{surv} . On the other hand, all components of L are conserved during collisions, but L_x and L_y depend on Ω_{col} and Ω_{surv} , which are unknown, so we leverage only the conservation of L_z . Similarly to the orbital elements of the new, merged planet (Figure 3), the ML model predicts the orbital elements of the surviving planet with less scatter and bias than the baseline model.

We may also use the conservation of L_z and the near conservation of E as a test of the performance of the orbital-outcome model. For the set of validation systems, the predicted postcollision states have a median percent error in L_z of 0.3% and a median percent error in E of 0.6%. The near-perfect conservation of L_z and E by the model on systems from outside the training set provides additional support for its performance.

Lastly, using our set of shadow integrations, we quantify the limits on orbital-element prediction accuracy imposed by chaos. To do so, for each of the 200 validation systems, we first bin the shadow integrations according to which two planets experienced a collision. Then, we measure the scatter in the orbital elements about the mean values across the shadow

Table 1
Root-mean-squared Scatter σ and Bias b (Defined as the Mean of Residuals) of the Orbital-outcome Regressor and a Baseline Model on the 103,404 Validation Systems

Target	Method	Scatter σ	Bias b
a_{col}	Baseline model	0.046	-0.010
a_{col}	ML model	0.043	-0.005
a_{col}	Limit from chaos	0.040	0.000
$\log(e_{\text{col}})$	Baseline model	0.372	0.028
$\log(e_{\text{col}})$	ML model	0.270	0.007
$\log(e_{\text{col}})$	Limit from chaos	0.247	0.000
$\log(i_{\text{col}})$	Baseline model	0.510	-0.301
$\log(i_{\text{col}})$	ML model	0.266	-0.011
$\log(i_{\text{col}})$	Limit from chaos	0.242	0.000
a_{surv}	Baseline model	0.564	0.126
a_{surv}	ML model	0.492	0.005
a_{surv}	Limit from chaos	0.494	0.000
$\log(e_{\text{surv}})$	Baseline model	0.497	0.138
$\log(e_{\text{surv}})$	ML model	0.304	-0.007
$\log(e_{\text{surv}})$	Limit from chaos	0.282	0.000
$\log(i_{\text{surv}})$	Baseline model	0.339	0.069
$\log(i_{\text{surv}})$	ML model	0.265	-0.008
$\log(i_{\text{surv}})$	Limit from chaos	0.243	0.000
E	Baseline model	0.000	0.000
E	ML model	2×10^{-5}	-3×10^{-6}
L_z	Baseline model	0.000	0.000
L_z	ML model	5×10^{-6}	-9×10^{-7}

Note. For comparison, we report the accuracy limits imposed by chaos (see Section 3.1 for more details).

system realizations (which we find to be distributed approximately log-normally), thereby isolating the unavoidable scatter due to the chaotic nature of these simulations. Limits on prediction accuracy due to chaos, along with the performance of the ML and baseline models are reported in Table 1. For all

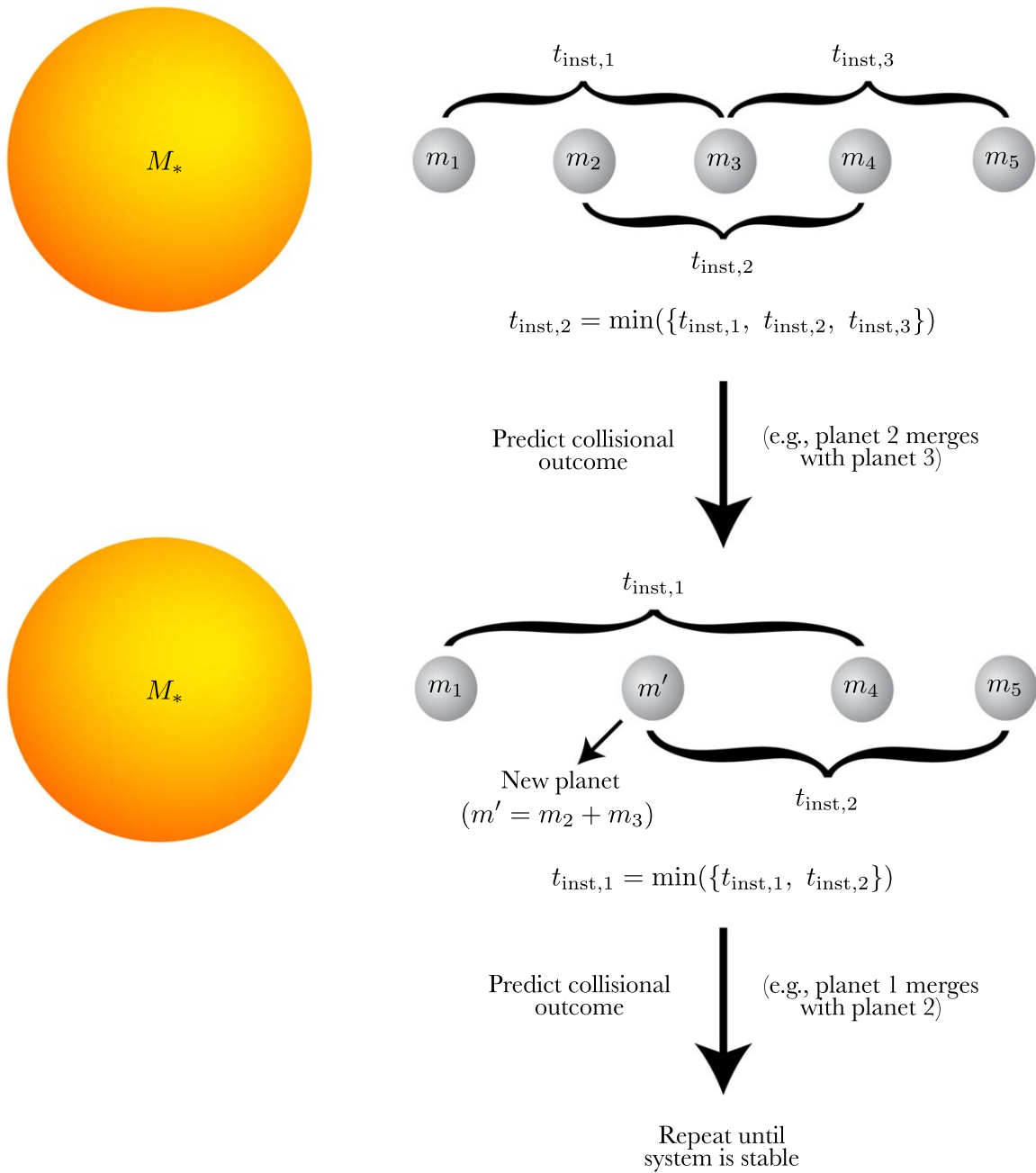


Figure 5. Schematic of the iterative giant-impact emulator. At each step, instability times for subtrios of planets are predicted using [SPOCKII](#) (M. Cranmer et al. 2021). Then, with the ML model presented in this work, we predict the collisional outcome of the planet trio with the shortest instability time, replace the trio with the two new planets, and repeat until [SPOCKII](#) identifies the system as long-term stable.

six predicted orbital elements, the accuracy of our ML model is approaching the limit imposed by chaos, indicating that there is little room for improvement. Note, however, that there are likely complex correlations between the outputs of the model.

3.2. Predicting Giant-impact Outcomes

By combining our ML model with [SPOCKII](#), which can rapidly predict instability times, we create an iterative giant-impact emulator for modeling the planet-formation process (see Figure 5 for a schematic that illustrates this model). Below, we evaluate the ability of the emulator to reproduce the results of computationally expensive N -body simulations.

For comparison with our giant-impact emulator, we carry out 500 long-term N -body simulations of initially overly packed, ten-planet systems. We initialize each system around a solar-mass star, with planet masses drawn randomly from $\mathcal{N}(3M_{\oplus}, 3M_{\oplus})$ and planet radii determined based on the mass–radius relationship from A. Wolfgang et al. (2016). Planets were spaced by iteratively drawing initial adjacent-planet period ratios P_{i+1}/P_i from the uniform distribution spanning [1.10, 1.75]. Initial orbital eccentricities and inclinations were drawn from Rayleigh distributions with scale parameters $\sigma = 0.01$ and $\sigma = 0.5^\circ$, respectively. Initial true longitudes and longitudes of pericenter were drawn uniformly over the range $[0, 2\pi]$. These initial conditions were chosen to reflect the expected state of multiplanet systems when the gas

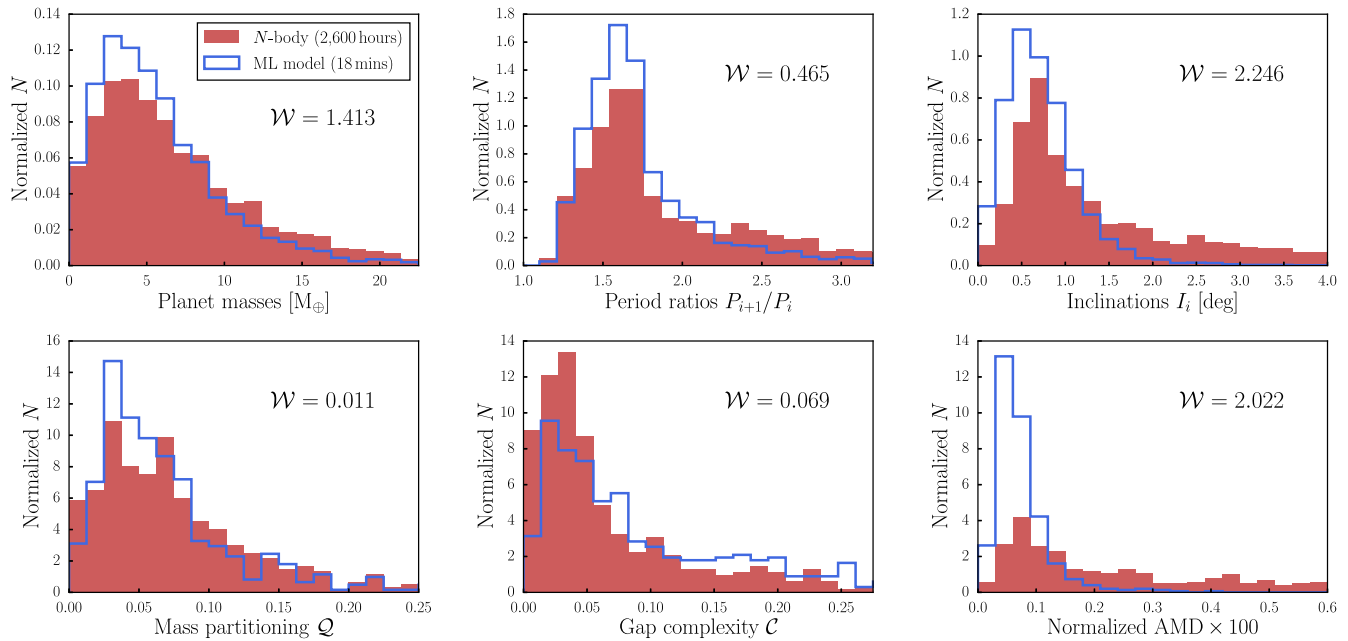


Figure 6. Comparison between the properties of 500 synthetic planetary systems from long-term N -body simulations and our ML-based giant-impact emulator (which is $\sim 10^4$ faster than full N -body simulations). The masses, spacings, and orbital inclinations of the planets in the ML-predicted systems agree well with those from N -body simulations (top row). System-level properties of the ML-predicted systems also agree with the N -body systems (bottom row), although we find the ML-predicted systems to be somewhat less dynamically excited. The 1D Wasserstein distance \mathcal{W} between the two distributions are reported in each panel. Our ML-based model outperforms a non-ML baseline model, whose predictions agree less well with the true N -body distributions (for the quantities shown in the top-left to the bottom-right panels, $\mathcal{W} = 0.281, 0.507, 2.293, 0.038, 0.096, 2.048$).

disk dissipates, and they resemble those adopted when numerically modeling the giant-impact process (see, e.g., S. T. S. Poon et al. 2020; M. Goldberg & K. Batygin 2022; C. Lammers et al. 2023). Systems were integrated for 10^9 orbits using the MERCURIUS integrator with a time step of $1/20$ th of the innermost planet’s initial orbital period, and collisions were treated as perfect inelastic mergers. After $10^9 P_1$ of dynamical evolution, the systems contain a mean of 5.5 planets (factor ~ 2 reduction), whose orbital properties have been set by the giant-impact process.

Figure 6 compares the properties of the 500 planetary systems from N -body simulations with the outcomes predicted by our ML-based giant-impact emulator, which is 8700 times faster in this case. The distribution of planet masses, adjacent-planet spacings, and orbital inclinations across the ML-predicted systems agree well with that of the N -body systems (for quantitative comparisons, we report the 1D Wasserstein distance between the ML and N -body distributions in Figure 6). In addition to the aggregate properties of the planets, the system-level properties of the ML-predicted and N -body systems largely agree. To illustrate, we compare the two sets of systems using three metrics: the mass partitioning \mathcal{Q} , which describes how mass is distributed among the planets ($\mathcal{Q} = 0$ when planets have equal masses and $\mathcal{Q} = 1$ when only a single planet is massive; \mathcal{Q} is defined in G. J. Gilbert & D. C. Fabrycky 2020); the gap complexity \mathcal{C} , which quantifies the spacing of the planets ($\mathcal{C} = 0$ when planets are equally spaced in log-period space; defined in G. J. Gilbert & D. C. Fabrycky 2020); and normalized angular-momentum deficit (AMD), which describes the dynamical excitation of the system (see J. E. Chambers 2001; D. Turrini et al. 2020). Comparing such metrics is a more stringent test of the ML-based emulator because they depend sensitively on which planets are combined (see, e.g., C. Lammers et al. 2023).

The mass partitioning and gap complexity of the ML-predicted systems agree well with that of the N -body systems (Figure 6). However, the normalized AMDs of the ML-predicted systems are typically smaller than that of the N -body systems, indicating that the ML systems are less dynamically excited (this is also reflected in the distribution of orbital inclinations). This discrepancy is likely caused by the fact that the ML-predicted systems have undergone fewer collisions than the N -body systems. Even though SPOCKII predicts the final ML-predicted systems to be stable, they contain a mean of 6.6 planets, 1.1 planets above the mean of the true N -body systems. We find this trend to hold for other sets of initial conditions as well, so users of the emulator model should be aware that systems tend to be somewhat less dynamically evolved than N -body simulations run for the same maximum time. More accurate instability-time predictions would help resolve this issue. We note, additionally, that predicting instability times with SPOCKII is the current computational bottleneck of the model—it is possible to speed up the emulator model (by up to a factor of ~ 3) with more rapid instability-time predictions.

To help interpret the performance of the ML-based giant-impact emulator, and quantify to what extent the strong agreement with N -body simulations is due to the tendency of collisions to “average out,” we also predict giant-impact outcomes for the 500 initial conditions with a non-ML baseline model. Our baseline model, in this case, relies on the same iterative process as the emulator (Figure 5), except that at each step, the unstable trio is chosen based on a stability heuristic, the colliding planet pair is determined using our best non-ML baseline model (Appendix B), and mergers are handled with our best non-ML baseline (Figures 3 and 4). We adopt the stability criterion (Equation (3)) from S. Tremaine (2015), which attempted to predict the outcomes of the giant-impact

process analytically. Systems are identified as stable, and the iterative process is stopped, once no planet pairs violate the 10^9 orbit stability criterion (i.e., all adjacent-planet pairs are separated by more than 10 mutual Hill radii).

To compare the accuracy of the baseline model with the ML-based emulator, we calculate the 1D Wasserstein distance between the properties of the baseline systems and the N -body systems. For the properties reported in the top-left to bottom-right panels of Figure 6, the baseline systems have $\mathcal{W} = 0.281, 0.507, 2.293, 0.038, 0.096,$ and 2.048 , indicating that the ML-predicted systems more closely match the N -body systems for all considered properties except the planet masses. The baseline planet masses agree better with the N -body planet masses largely because the multiplicities of the baseline systems (mean of 5.3 planets) are closer to that of the true N -body systems (mean of 5.5 planets) than the ML-predicted system multiplicities are (mean of 6.6 planets). Note, however, that the mean number of planets in the baseline systems depends sensitively on the adopted stability heuristic, and despite the similar multiplicities, the other properties of the baseline systems are all less accurate than the ML-predicted systems.

Comparing with the analytical predictions of S. Tremaine (2015) is complicated by the presence of the free parameter τ in their predicted distributions, which depends nontrivially on the set of initial conditions. For normalized eccentricity (Equation (32) in S. Tremaine 2015), which does not depend on τ , we find the distribution predicted by S. Tremaine (2015) to significantly overestimate the normalized eccentricities of the N -body systems in Figure 6. Additionally, the absolute eccentricities of the N -body systems have a high-eccentricity tail that is not well described by the functional form predicted by S. Tremaine (2015), nor a Rayleigh distribution.

As an additional test of the emulator model, we compare the angular momentum L_z and energy E of the final systems with that of the initial conditions. We find that the predicted systems have a median percent error in L_z of just 0.8% and a median percent error in E of 1.8%. In practice, the conservation of L_z and E can be used as model diagnostics, allowing the user to reject unphysical systems in which L_z or E are poorly conserved due to hallucinations of the model, which would otherwise be challenging to identify.

4. Discussion and Conclusions

In this work, we present an ML-based approach to predicting the collisional outcomes of multiplanet systems. Breaking this task into two subproblems, we develop two models: a “collision classifier” that predicts which pair of planets will collide, and an “orbital-outcome regressor” that predicts the orbital elements of the two postcollision planets (see Figure 1).

Using shadow integrations in which the initial conditions of our synthetic planetary systems are slightly perturbed, we show that our collision-classifier model can accurately predict the probabilities of a collision occurring between different planet pairs. With a scatter of $\sim 10\%$ and little bias about the true collision probabilities, our model significantly outperforms baseline models constructed based on metrics from dynamics theory. Similarly, we find that the orbital-outcome regressor outperforms non-ML baselines at the task of predicting the planets’s postcollision states. The prediction accuracy of the outcome regressor is only narrowly below the accuracy limits imposed by chaos, and the model’s predictions conserve energy and angular momentum to within $\lesssim 1\%$.

By combining our model with SPOCKII (M. Cranmer et al. 2021), which provides rapid instability-time predictions, we create an efficient giant-impact emulator (see the schematic in Figure 5). We find that our ML-based emulator can predict the outcomes of N -body giant-impact simulations with reasonable accuracy and a speedup of up to 4 orders of magnitude. Although powerful, this approach requires caution, and we list some notable caveats below.

First, because our ML model was restricted to three-planet systems, instabilities and collisions cannot occur between planet pairs separated by more than one intervening planet (e.g., planets 1 and 4). This could present an issue for systems containing bodies that span many orders of magnitude in mass, in which case nonconsecutive trios may cause the system to destabilize. Second, our model does not account for the scattering of planets that are not included in the unstable planet trio (e.g., if planets 1, 2, and 3 form an unstable trio, a collision between planets 2 and 3 could scatter planet 4). Third, in some instances, the time required for planets to merge after the system has destabilized can be comparable to the system’s instability time (D. R. Rice et al. 2018), which could result in planets being merged in an incorrect order. Comparisons with N -body simulations revealed that there is room for improvement in the accuracy of the giant-impact emulator’s predictions. Progress will likely require more accurate instability-time predictions or changes to the iterative-prediction process on which the emulator is based, rather than improvements to the collision-classifier and orbital-outcome regressor models.

The substantial computational speedup provided by our collision-outcome model and giant-impact emulator enables analyses that would not otherwise be feasible, including the ability to carry out more complete explorations of the parameter space of initial conditions. For this reason, we release our model with an easy-to-use API through the SPOCK package,¹² alongside our full training code.¹³

Acknowledgments

We thank the anonymous referee for a thorough review that significantly improved the quality of the paper. We would also like to thank Christian Kragh Jespersen, Amir Siraj, Yubo Su, Scott Tremaine, and Joshua Winn for useful discussions. N.M. acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [RGPIN-2023-04901].

Appendix A Neural Network Architectures

Here, we describe the architecture of our two MLP models, constructed using the open-source PyTorch (A. Paszke et al. 2019) package. Before training the models presented in the main text, the hyperparameters for both models were determined (independently) with Bayesian optimization, as implemented in the hyperopt code. Specifically, for both models, we found the combination of hyperparameters (i.e., the learning rate, weight decay, minibatch size, number of hidden layers, and number of nodes in the hidden layers) that minimized the loss on a validation set after 10,000 optimization steps. For this, we used a reduced training set of 51,702

¹² <https://github.com/dtamayo/spock>

¹³ https://github.com/CalebLammers/ML_for_collisions

systems randomly selected from the full training set, of which 20% were reserved for validation.

A.1. Collision Classifier

To predict whether a collision will occur between planets 1–2, planets 2–3, or planets 1–3, we use a fully connected MLP with three output nodes. The model includes one hidden layer with 30 hidden nodes and takes 45 inputs, corresponding to the three planet masses $\log(m_i)$, along with the mean and standard deviation of a_i , $\log(e_i)$, $\log(I_i)$, $\sin(\varpi_i)$, $\cos(\varpi_i)$, $\sin(\Omega_i)$, and $\cos(\Omega_i)$, where $i = 1, 2, 3$. We found that providing the model as input the logarithm of the masses, eccentricities, and inclinations improved the performance of both the classification and regression models, likely because these inputs can span several orders of magnitude.

We use a Rectified Linear Unit (ReLU) activation function on the input and hidden layers and a softmax activation function on the output layer so that the three outputs sum to unity and can be interpreted as probabilities. We train the model using the ADAM optimizer, with a learning rate of 7×10^{-4} and a weight decay of 1×10^{-4} . The model was trained for 1,000,000 optimizer steps using a binary cross-entropy loss function, random minibatches of size 1000, and a 20% validation set. We explored training the model with other optimizers, including stochastic gradient descent, but found that models trained with ADAM performed best on the validation set.

Before training, we rescale the input masses and orbital elements so that the input features have a mean of zero and a standard deviation of unity across all training systems (during inference, inputs are similarly normalized before being passed to the model). We found that normalizing the input features helped to speed up the training of the model. During training, the model was provided with a random subset of the 100 orbit samples recorded across the 10^4 orbit short integration. In detail, we drew n_t uniformly from [5, 100] and calculated the mean and standard deviation of each orbital element from n_t samples selected randomly (without replacement) from the 100 recorded states. To reflect the varying number of orbit samples and add a small amount of noise to the input features, we draw the means and standard deviations that are passed as input to the MLP model, $\hat{\mu}$ and $\hat{\sigma}$, from their frequentist distributions $\hat{\mu} = \mathcal{N}(\mu, \sigma/\sqrt{n_t})$ and $\hat{\sigma} = \mathcal{N}(\sigma, \sigma/\sqrt{2(n_t - 1)})$, where μ and σ are the mean and standard deviation measured based on the n_t orbit samples. Varying the number of orbit samples during training acts as a form of data augmentation, which we found improved generalization to the validation set. During inference, the mean and standard deviation are calculated from all 100 orbital samples with no added statistical noise.

A.2. Orbital-outcome Regressor

To predict the orbital elements of the two postcollision planets, we use a fully connected MLP with six output nodes, corresponding to a_{col} , $\log(e_{\text{col}})$, $\log(i_{\text{col}})$, a_{surv} , $\log(e_{\text{surv}})$, and $\log(i_{\text{surv}})$. The model takes 45 inputs and consists of a single hidden layer with 60 hidden nodes. As input, we provide the regression model with the same 45 values required by the classification model ($\log(m_i)$, a_i , $\log(e_i)$, $\log(I_i)$, $\sin(\varpi_i)$, $\cos(\varpi_i)$, $\sin(\Omega_i)$, and $\cos(\Omega_i)$, where $i = 1, 2, 3$), except that the inputs are ordered in order to encode which two planets are to be combined. The orbital properties of the two colliding

planets (ordered by semimajor axis) are provided first, followed by those of the surviving planet. We found that reordering the inputs to tell the model which two planets are to be combined worked significantly better than passing a separate input value that specified the two colliding planets.

For the regression MLP, we use a ReLU activation function on the input and hidden layer, and a linear activation function on the output layer. Similarly to the classification model, we train the model with ADAM (which is, again, the best-performing optimizer) for 1,000,000 optimizer steps, in this case with a batch size of 3000, a learning rate of 7×10^{-4} , and a weight decay of 1×10^{-4} , which were determined based on the hyperparameter optimization process. As with the collision-classifier model, during training, we varied the number of orbit samples used to calculate the mean and standard deviation of the orbital elements (n_t was drawn uniformly from [5, 100]), and drew the inputs passed to the model from their corresponding frequentist distributions.

A complication of the orbital-element regression problem is that some of the outputs have strict upper limits (e.g., $\log(e_{\text{col}}) < 0.0$) and $\log(e_{\text{surv}}) < 0.0$). For data points in the training set that are near their upper limit, it is no longer reasonable to assume that their error terms are well described by symmetric Gaussian distributions, and as a result, the model with minimum mean-squared error no longer corresponds to the maximum likelihood model. As a result, when trained with a mean-squared-error loss function, we found that the eccentricities predicted by the regression model were systematically biased toward low values. To resolve this, we adopt the loss function $L(y_{\text{pred}} | y_{\text{true}}) = \sum_{i=0}^5 ((y_{\text{pred},i} - y_{\text{true},i})^2 + \log(1 + \text{erf}(y_{\text{max},i} - y_{\text{pred},i})))$, where y_{pred} represents the six outputs of the model, y_{true} represents the true values, and y_{max} represents the maximum allowed value of each output. The first term in $L(y_{\text{pred}} | y_{\text{true}})$ corresponds to the standard mean-squared error and the second term accounts for the penalization of predictions near the upper limits (see the Appendix of SPOCKII for a mathematical derivation of the likelihood for data with lower/upper limits). With this loss function, the model predicts postcollision eccentricities with little bias (see the middle panel of Figures 3 and 4). For the interested reader, our full training code is available at https://github.com/CalebLammers/ML_for_collisions.

Appendix B Baseline Classification Model

Despite extensive theoretical investigation into the destabilization of compact multiplanet systems, there have been relatively few attempts to predict which planets will be involved in the collision. Any binary criterion based on the configuration of the system (e.g., nearest planet pair) inevitably predicts a collision fraction of unity for one planet pair and a fraction of zero for the other two, which is a poor prediction (see Figure 2). Randomly selecting a pair of planets (resulting in $f_{\text{planets } i-j} \approx 0.33$) is a similarly inadequate prediction. Instead, we approach this task by predicting three logits (logit_{1-2} , logit_{2-3} , and logit_{1-3}) based on the initial conditions of the systems, which are converted to collision fractions according to $f_{\text{planets } i-j} = \text{softmax}(\text{logit}_{i-j})$.

We attempted to predict collision probabilities using several different physical properties of the systems. First, motivated by

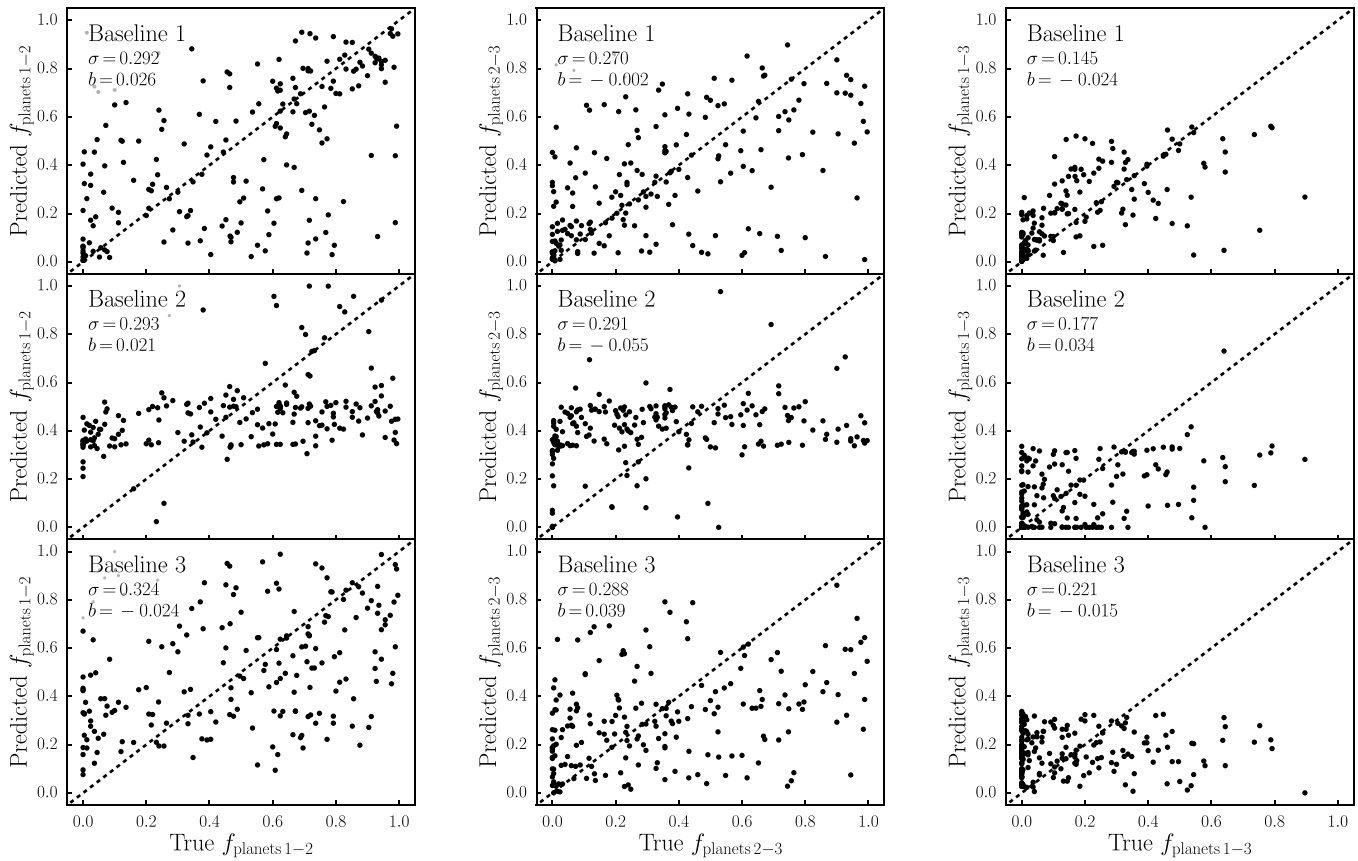


Figure 7. Performance of three non-ML baseline classification models on the 200 validation systems, for comparison with our ML model (Figure 2). From top to bottom, the baseline model predictions rely on the mass ratios of the planets, the amount of angular momentum required for the planets’s orbits to cross, and the normalized eccentricity of the planets. The baseline models predict three logits, which are converted to probabilities using a softmax function (see Appendix B). Despite leveraging dynamics-inspired metrics, the predictions of all three baseline models are poor.

the finding that small planets are preferentially involved in collisions (C. Lammers et al. 2023), we considered a model that calculates logits using the ratio of the planet masses: $\text{logit}_i = c_i \log(\max(m_{i+1}/m_i, m_i/m_{i+1}))$ where c_i is an arbitrary constant. We tune the three constants c_i to minimize the bias b (defined as the mean of the residuals) with respect to the true collision fractions $f_{\text{planets } i-j}$. Second, we construct a model in which $\text{logit}_i = c_i L_{i-j, \text{cross}}$, where $L_{i-j, \text{cross}}$ is the minimum orbital momentum that, when transferred to planet i or planet j , increases the planets’s eccentricity enough to allow for crossing orbits. This criteria thereby takes into account the planets’s spacings, masses, and eccentricities. Lastly, we consider a model that calculates logits using the planets’s normalized eccentricities (as defined in D. Tamayo et al. 2021), which depend on the planets’s spacings, eccentricities, and orbital orientations ($\text{logit}_i = c_i |\tilde{e}_{ij}|$).

The collision probabilities predicted by the three baseline models are compared with the true collision fractions for 200 validation systems in Figure 7. All models perform poorly, with a scatter about the true collision fractions of $\sigma \approx 0.3$, 3 times larger than that of our ML model (Figure 2). Predictions are similarly poor when logits are predicted based on the mean orbital elements over 10^4 orbits or the state of the system at the end of the 10^4 orbits, or when using other common stability metrics, like the spacing of the planets in mutual Hill radii. The inability of all baseline models we considered to accurately predict collision probabilities highlights the difficulty of this task.

ORCID iDs

Caleb Lammers <https://orcid.org/0000-0001-9985-0643>
Miles Cranmer <https://orcid.org/0000-0002-6458-3423>
Sam Hadden <https://orcid.org/0000-0002-1032-0783>
Shirley Ho <https://orcid.org/0000-0002-1068-160X>
Norman Murray <https://orcid.org/0000-0002-8659-3729>
Daniel Tamayo <https://orcid.org/0000-0002-9908-8705>

References

- Agnor, C. B., Canup, R. M., & Levison, H. F. 1999, *Icar*, 142, 219
Bailey, N. A., & Fabrycky, D. C. 2022, *MNRAS*, 514, 4765
Borucki, W. J., Koch, D. G., Basri, G., et al. 2011, *ApJ*, 736, 19
Chambers, J. E. 1999, *MNRAS*, 304, 793
Chambers, J. E. 2001, *Icar*, 152, 205
Chambers, J. E., & Wetherill, G. W. 1998, *Icar*, 136, 304
Chambers, J. E., Wetherill, G. W., & Boss, A. P. 1996, *Icar*, 119, 261
Cranmer, M., Tamayo, D., Rein, H., et al. 2021, *PNAS*, 118, e2026053118
Esteves, L., Izidoro, A., Bitsch, B., et al. 2022, *MNRAS*, 509, 2856
Fabrycky, D. C., Lissauer, J. J., Ragozzine, D., et al. 2014, *ApJ*, 790, 146
Fang, J., & Margot, J.-L. 2012, *ApJ*, 761, 92
Fang, J., & Margot, J.-L. 2013, *ApJ*, 767, 115
Fressin, F., Torres, G., Charbonneau, D., et al. 2013, *ApJ*, 766, 81
Ghosh, T., & Chatterjee, S. 2024, *MNRAS*, 527, 79
Gilbert, G. J., & Fabrycky, D. C. 2020, *AJ*, 159, 281
Goldberg, M., & Batygin, K. 2022, *AJ*, 163, 201
Hansen, B. M. S., & Murray, N. 2012, *ApJ*, 751, 158
Hansen, B. M. S., & Murray, N. 2013, *ApJ*, 775, 53
He, M. Y., Ford, E. B., & Ragozzine, D. 2019, *MNRAS*, 490, 4575
He, S., Li, Y., Feng, Y., et al. 2019, *PNAS*, 116, 13825
Hussain, N., & Tamayo, D. 2020, *MNRAS*, 491, 5258

- Izidoro, A., Ogihara, M., Raymond, S. N., et al. 2017, *MNRAS*, **470**, 1750
- Jamieson, D., Li, Y., de Oliveira, R. A., et al. 2023, *ApJ*, **952**, 145
- Kant, I. 1755, *Allgemeine Naturgeschichte und Theorie des Himmels* (Lausanne: Univ. of Lausanne)
- Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
- Kokubo, E., & Ida, S. 1998, *Icar*, **131**, 171
- Kokubo, E., Kominami, J., & Ida, S. 2006, *ApJ*, **642**, 1131
- Lammers, C., Hadden, S., & Murray, N. 2023, *MNRAS*, **525**, L66
- Lammers, C., Hadden, S., & Murray, N. 2024, *ApJ*, **972**, 53
- Laplace, P.-S. 1796, *Exposition du Système du Monde* (Paris: Cerie-Social)
- Lissauer, J. J. 1993, *ARA&A*, **31**, 129
- Lissauer, J. J., Ragozzine, D., Fabrycky, D. C., et al. 2011, *ApJS*, **197**, 8
- Obertas, A., Tamayo, D., & Murray, N. 2023, *MNRAS*, **526**, 2118
- Paszke, A., Gross, S., Massa, F., et al. 2019, in *Advances in Neural Information Processing Systems 32* (Red Hook, NY: Curran Associates, Inc.), 8024
- Petit, A. C., Pichierri, G., Davies, M. B., & Johansen, A. 2020, *A&A*, **641**, A176
- Poon, S. T. S., Nelson, R. P., Jacobson, S. A., & Morbidelli, A. 2020, *MNRAS*, **491**, 5595
- Pu, B., & Wu, Y. 2015, *ApJ*, **807**, 44
- Quillen, A. C. 2011, *MNRAS*, **418**, 1043
- Raymond, S. N., Barnes, R., & Mandell, A. M. 2008, *MNRAS*, **384**, 663
- Raymond, S. N., Quinn, T., & Lunine, J. I. 2005, *ApJ*, **632**, 670
- Rein, H., Hernandez, D. M., Tamayo, D., et al. 2019, *MNRAS*, **485**, 5490
- Rein, H., & Spiegel, D. S. 2015, *MNRAS*, **446**, 1424
- Rein, H., & Tamayo, D. 2015, *MNRAS*, **452**, 376
- Rice, D. R., Rasio, F. A., & Steffen, J. H. 2018, *MNRAS*, **481**, 2205
- Sobski, N., & Millholland, S. C. 2023, *ApJ*, **954**, 137
- Tamayo, D., Cranmer, M., Hadden, S., et al. 2020, *PNAS*, **117**, 18194
- Tamayo, D., Murray, N., Tremaine, S., & Winn, J. 2021, *AJ*, **162**, 220
- Tamayo, D., Silburt, A., Valencia, D., et al. 2016, *ApJL*, **832**, L22
- Tejada Arevalo, R., Tamayo, D., & Cranmer, M. 2022, *ApJL*, **932**, L12
- Tremaine, S. 2015, *ApJ*, **807**, 157
- Turrini, D., Zinzi, A., & Belinchon, J. A. 2020, *A&A*, **636**, A53
- Volk, K., & Gladman, B. 2015, *ApJL*, **806**, L26
- Wisdom, J. 2015, *AJ*, **150**, 127
- Wisdom, J., & Holman, M. 1991, *AJ*, **102**, 1528
- Wolfgang, A., Rogers, L. A., & Ford, E. B. 2016, *ApJ*, **825**, 19
- Yee, S. W., Tamayo, D., Hadden, S., & Winn, J. N. 2021, *AJ*, **162**, 55
- Zhu, W., Petrovich, C., Wu, Y., Dong, S., & Xie, J. 2018, *ApJ*, **860**, 101