The Astrophysical Journal, 875:40 (14pp), 2019 April 10 https://doi.org/10.3847/1538-4357/ab0e8a

Realistic On-the-fly Outcomes of Planetary Collisions: Machine Learning Applied to Simulations of Giant Impacts

Saverio Cambioni[1], Erik Asphaug[1], Alexandre Emsenhuber[1], Travis S. J. Gabriel[2], Roberto Furfaro[3], and Stephen R. Schwartz[1]

[1] Lunar and Planetary Laboratory, University of Arizona, 1629 E. University Blvd., Tucson, AZ 85721, USA; cambioni@lpl.arizona.edu

[2] School of Earth and Space Exploration, Arizona State University, 781 E. Terrace Mall, Tempe, AZ 85287, USA

[3] Systems and Industrial Engineering Department, University of Arizona, 1127 E. James E. Rogers Way, Tucson, AZ 85721, USA

Received 2019 January 10; revised 2019 March 6; accepted 2019 March 9; published 2019 April 12

Abstract

Planet formation simulations are capable of directly integrating the evolution of hundreds to thousands of planetary embryos and planetesimals as they accrete pairwise to become planets. In principle, these investigations allow us to better understand the final configuration and geochemistry of the terrestrial planets, and also to place our solar system in the context of other exosolar systems. While these simulations classically prescribe collisions to result in perfect mergers, recent computational advances have begun to allow for more complex outcomes to be implemented. Here we apply machine learning to a large but sparse database of giant impact studies, which allows us to streamline the simulations into a classifier of collision outcomes and a regressor of accretion efficiency. The classifier maps a four-dimensional (4D) parameter space (target mass, projectile-to-target mass ratio, impact velocity, impact angle) into the four major collision types: merger, graze-and-merge, hit-and-run, and disruption. The definition of the four regimes and their boundary is fully data-driven. The results do not suffer from any model assumption in the fitting. The classifier maps the structure of the parameter space and it provides insights into the outcome regimes. The regressor is a neural network that is trained to closely mimic the functional relationship between the 4D space of collision parameters, and a real-variable outcome, the mass of the largest remnant. This work is a prototype of a more complete surrogate model, that will be based on extended sets of simulations (big data), that will quickly and reliably predict specific collision outcomes for use in realistic *N*-body dynamical studies of planetary formation.

Key words: methods: numerical – planetary systems – planets and satellites: terrestrial planets

Supporting material: machine-readable tables

1. Introduction

The idea of giant impacts has gone well beyond the formation of the Moon (e.g., Hartmann & Davis 1975; Stevenson 1987; Benz et al. 1989; Canup & Asphaug 2001) and it is now able to give a new understanding of planet formation during the late stage, where bodies that are similar in size collide at one to several times their mutual escape velocity v_{esc},

$$v_{\mathrm{esc}} = \sqrt{\frac{2G(M_{\mathrm{T}} + M_{\mathrm{P}})}{R_{\mathrm{coll}}}}, \tag{1}$$

where M_{T} is the mass of the target, M_{P} is the mass of the projectile and $R_{\mathrm{coll}} = R_{\mathrm{T}} + R_{\mathrm{P}}$ is the separation at initial contact (e.g., Wetherill 1985; Asphaug 2010). Supported by increasingly sophisticated models, hypotheses have emerged for the giant impact formation of planets, including Mercury (Benz et al. 2007; Asphaug & Reufer 2014; Chau et al. 2018), Pluto–Charon (e.g., Canup 2005, 2011), Haumea (Leinhardt et al. 2010), Titan (Asphaug & Reufer 2013), and the Moon. In a broad sense, giant impact events have had a significant role in determining the final physical properties of rocky/icy planets.

In *N*-body dynamical studies, planetary embryos orbit the Sun and each giant impact is typically assumed to be fully accretionary, so that *N* only decreases in time. However, perfect merging is known (Chambers 2013) to be a problematic oversimplification of more complex outcomes, which has been demonstrated by decades of detailed hydrocode simulations (e.g., Asphaug et al. 2006) using methods such as Smoothed-Particle Hydrodynamics (SPH)—as described later on. The most common collision events at the end-stage of terrestrial planet formation in the solar system involve similar-sized bodies and $v_{\mathrm{coll}}/v_{\mathrm{esc}} = 1$–4 (Agnor et al. 1999). Over this range of mass ratios and impact velocities, collision outcomes span all the regimes of accretion, erosion, and hit-and-run (Leinhardt & Stewart 2012). Some more advanced *N*-body approaches have implemented simple rules to limit accretion efficiency (e.g., Chambers 2013), but approximations such as perfect mergers are still the norm (e.g., O'Brien et al. 2006; Raymond et al. 2009).

One ultimate strategy is to model collisions on the fly, such as by using SPH to model a given impact event while the *N*-body evolution is in progress (e.g., Haghighipour et al. 2017). However, in practice this approach has limitations. For a giant impact simulation to run in less than an hour, which is a practical limit when the *N*-body evolution must wait, the hydrocode resolution is limited to a $\sim 10^4$ particles, which is only adequate to classify the most basic outcomes (Agnor & Asphaug 2004). In addition, data reduction is a concern. Well-resolved giant impact simulations often generate multiple debris products (including intact remnants), and these must be identified and characterized in each output file to be fed back into the *N*-body code. These include the projectile runner in the case of hit-and-run (Asphaug et al. 2006), and other self-gravitating clumps and debris. Graze-and-merge collisions can spin off escaping bodies up to a third the size of the progenitors

(Asphaug & Reufer 2013), and head-on impacts appear to make fields of sizable clumps (e.g., Sugiura et al. 2018). Keeping track of all this requires post-processing analysis of the collision outcome and this increases N, which can stall the evolution. Ensuring convergence of the debris field requires larger numbers of particles than a nominal simulation (e.g., Genda et al. 2015).

However, a detailed description of the debris field is neither needed nor desired. Instead, we would prefer a summary description of the two or three major bodies emerging from the giant impact, their thermodynamic and orbital dynamic states, and useful statistics regarding the remaining debris, such as their characteristic sizes and velocity distributions, and also the overall mass, momentum and composition. Lastly, knowledge of the specific impact properties (e.g., angle of impact relative to spin state of planet) is in fact completely unknown, so that running a superb 3D simulation of a specified giant impact is a misplaced effort unless the results can be generalized in some way. Our approach is to use high-fidelity SPH calculations as a training dataset, beginning with the impact simulations published by Reufer (2011), which is also the basis for Gabriel et al.'s (2019) development of a forward-functional model from the same dataset.

We use SPH to model giant impacts on planetary bodies such as the Moon, Mercury and Mars (e.g., Reufer et al. 2012; Asphaug & Reufer 2014; Asphaug et al. 2015). Each SPH outcome is a complex N-Dimensional state (consolidated planets, clumps, unconsolidated ejecta, and their thermodynamic states and other characteristics) that requires detailed analysis. Giant impacts cover a large range of input parameters and they are intrinsically three-dimensional events. For example, a colliding pair of planets is represented by masses M_1 and M_2, their impact velocity and angle, target and impactor spin rate and orientation, plus some assumptions on their composition and internal structure. Performing five realizations of each variable would require nearly 400,000 simulations, just to produce a coarse mapping of the parameter space. The necessity of a detailed coverage of the parameter space is coupled with the requirement for precise (high-resolution) simulations. Simulations with 10^6 particles have become standard (e.g., Canup et al. 2013; Hyodo et al. 2017; Emsenhuber et al. 2018), and runs are extended to many gravitational times $\tau_g = \sqrt{4\pi/3\rho G}$ (Jutzi & Benz 2017).

We apply machine learning (ML) to build an accurate data-driven model of giant impacts, which does not simply interpolate the available data but rather generalizes the underlying functional relationship between impact properties and collision outcomes. Our aim is to fit the available data, but not to over-fit it; that is, to be inclusive of the expectation of new data that is yet to be observed. The data described here is ideal for an initial study but is being superceded by much higher fidelity models. One of the advantages of this approach is that higher fidelity data can be added to lower fidelity data in a weighted manner as they become available.

We present two distinct machine-learned response functions for collisions in the gravity regime: a *classifier* of collision types and a *regressor* of accretion efficiency. These functional models—compact algorithms—map the outcome of a giant impact (post-collision end state) into a four-dimensional (4D) parameter space; i.e., mass of target, projectile-to-target mass ratio, impact velocity and impact angle. The training is performed on existing giant impact simulations between similar-size differentiated chondritic bodies. The resulting surrogate collision models give a reliable result to within a known degree of confidence and at a highly reduced computational time (with respect to full giant impact simulations; i.e., on the order of seconds on a single computing thread). Therefore, they are designed to apply especially well to N-body evolution calculations and to constrain pre-impact dynamical conditions from an hypothesized post-collision scenario (Jackson et al. 2018).

Table 1
Pairs of Target Masses and Projectile-to-target Mass Ratio Present in the Collisions Dataset from Reufer (2011) and Gabriel et al. (2019)

Target mass [$M_\oplus$]	Mass ratio (projectile/target)
1	0.20, 0.70
10^{-1}	0.10, 0.20, 0.35, 0.70
10^{-2}	0.20, 0.70

Note. For each pair, more than 100 runs with different impact velocity and angle are performed, ranging between 0° and 90°, and 1–4 times the mutual escape velocity, respectively (Figure 1).

The rest of this paper is organized as follows. In Section 2.1 we describe the available dataset. In Section 2.2 we provide an introduction to ML, with a focus on the two distinct algorithms used to train the classifier of collision types and the regressor of accretion efficiency: Support Vector Machine (SVM) (Section 2.2.1) and Neural Network (NN) (Section 2.2.2), respectively. In Sections 3 and 4, we present and discuss the predictions by the trained algorithms regarding the post-collision end states and the characterization of the parameter space. Finally, we discuss the potential of the methodology and we make several recommendations for future work/application in Section 5.

2. Materials and Methods

2.1. Dataset

In this study, we use SPH simulations from Reufer (2011). This dataset is completed at ~200,000-particle resolution and it spans a wide range of parameters, such as target mass, mass ratio (projectile/target), impact angle, and impact velocity. The bodies are similar in size and they are initially non-rotating. They are differentiated with a chondritic composition of 30% iron and 70% silicate. The values for the first two parameters that are present in the dataset are provided in Table 1. For each pair, more than 100 runs with different impact velocities and angles are performed, ranging between 0° and 90°, and 1–4 times the mutual escape velocity, respectively (see the top left-hand and top right-hand panels of Figure 1). These conditions are the most relevant to late-stage planet formation (Stewart & Leinhardt 2012; Chambers 2013). We do not include the initial spin rates among the impact parameters, which require three additional variables for each of the bodies (one for the magnitude and two for the orientation). However, the target and impactor spin rate have been found to be relevant for the overall impact outcome (e.g., Canup 2005, 2011) and we intend to include this parameter in our future ML applications.

The collisions in our dataset are modeled using the SPH technique. SPH is a physically based hydrodynamical model that uses a Lagragian description, which is suited for collision modeling, where a large range of densities is expected. Furthermore, no grid is required, which is in contrast

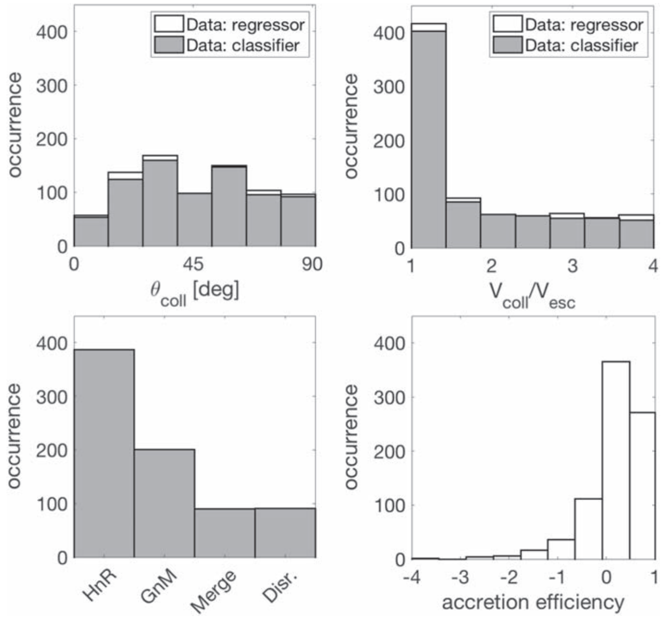

Figure 1. Top left-hand and top right-hand panels: frequency distributions of input impact angle $\theta_{\rm coll}$ and velocity $v_{\rm coll}/v_{\rm esc}$, respectively; the values for the other two input parameters that are present in the dataset—target mass and mass ratio (projectile/target)—are provided in Table 1. Bottom left-hand panel: frequency distribution of the collision classes as labeled in the classification task (Sections 2.2.1 and 3.1); on the x-axis, HnR refers to the simulations labeled as hit-and-run cases, GnM refers to the graze-and-merge cases, and Disr. refers to the disruption cases. Bottom right-hand panel: frequency distribution of accretion efficiency values—Equation (11)—which are used in the regression task (Sections 2.2.2 and 3.2). The simulations come from Reufer (2011). Further details on the database of simulations and detailed physical analysis are provided in Gabriel et al. (2019).

to the Eulerian methods. Quantities are obtained by interpolating over ∼50–100 neighbor particles using a kernel function—in this case a β-spline (Monaghan & Lattanzio 1985). Spatial derivatives are retrieved using the derivative of the kernel, so that no grid is required. Time evolution is provided by Euler's equations: mass conservation is used to obtain the density, energy conservation is used to obtain the internal energy, and momentum conservation is used to obtain the pressure gradient and self-gravity. An artificial viscosity is added to resolve shocks as is common in nearly all SPH implementations in planetary science (e.g., Monaghan 1992). An equation of state is required to obtain the pressure from the density and internal energy. Therefore, we use M-ANEOS for SiO_2 and ANEOS for iron (Thompson & Lauson 1972; Melosh 2007), which is a common choice for such studies. Self-gravity is based on a hierarchical spatial tree (Barnes & Hut 1986), where contributions from distant regions are estimated using a multi-pole approximation. The same tree is used to walk the nearest-neighbor search, which is a process that occurs throughout the simulation.

Each simulation begins with the bodies approaching from several radii away, which allows for tidal deformation to take place prior to the collision. The initial conditions are determined assuming a two-body problem, so that the velocity and angle at initial contact follow the prescribed values. The simulations are evolved for $50\tau_{\rm coll}$ past initial contact, with $\tau_{\rm coll}$ being the collision timescale defined as

$$\tau_{\rm coll} = \frac{2R_{\rm coll}}{v_{\rm coll}}, \tag{2}$$

where $v_{\rm coll}$ is the impact velocity and $R_{\rm coll} = R_{\rm T} + R_{\rm P}$ is the separation at initial contact. The indexes T and P refer to the target and projectile, respectively. Once the simulation has finished, the resulting bodies are found using the following iterative algorithm: particles pairs are iterated over, starting with the ones that have the lowest gravitational potential energy, and checked whether the pair is bound. If a pair is bound, then a new clump is started and the iteration continues checking particles against the new clump. For each particle added, the iteration is repeated until no further particle is found to be bound to the clump. This procedure is also used to compute the mass of the largest remnant of the collisions. Details on the simulation database and detailed physical analysis are provided in Gabriel et al. (2019). Snapshots of the movie rendering of these simulations are shown in Figure 2.

The simulations in the dataset use SPH in the original, fluid mode, while the equation of motion is derived only from the pressure gradient (e.g., Monaghan 1992) and self-gravity. This is appropriate when the stresses of gravity exceed the possible mechanical strengths, and for this reason the existing dataset has its lower limit at 1400 km diameters (the so-called gravity regime). For bodies 100–1000 km in diameter, it has been shown that friction (e.g., Jutzi 2015) and strength (Emsenhuber et al. 2018) are important, and have the potential to challenge our ideas of the origin of moons and embryos during the late stage of planet formation. For super-Earth and Neptune-mass bodies (10,000 km and larger), the dominant variables are thermodynamic processes, shocks and gravity (Marcus et al. 2009, 2010b; Liu et al. 2015; Kegerreis et al. 2018a). For this work, we limit ourselves to Earth-sized planets and smaller because a sufficiently large database for super-Earth and Neptune-sized collisions is not reported in the literature.

2.2. *Machine Learning*

ML is a subfield of data analysis that lies at the cornerstone between statistical methods and computer science, and is also at the core of artificial intelligence. ML was originally conceived to address the question of how to build computers that can autonomously improve through direct experience, and it enables machines to learn features and trends from the available data. Encouraged by the advances in parallel computing technologies (e.g., Graphic Processing Units, GPUs), the availability of massive labeled data and the breakthrough in understanding of deep NNs, over the past few years there has been an explosion of ML algorithms that can accurately process images for classification and regression tasks, such as image and video recognition (Krizhevsky et al. 2012), natural language processing (Socher et al. 2012), speech recognition (Hinton et al. 2012). State-of-the-art ML techniques have several advantages: first, they can streamline the generation of datasets to most efficiently explore regions of interest in a large parameter space; and second, they can perform accurate mappings of initial conditions and end states, with associated probabilities, while taking a high-dimensional parameter space into account. This is in contrast to human operators, who are often limited to a mostly 2D understanding of the data. ML schemes can take advantage of this big data problem to spot new and sometimes unexpected correlations.

ML techniques can be divided into supervised (or predictive) and unsupervised (or descriptive) methods. Supervised methods rely on a training set of data, with features/predictors and

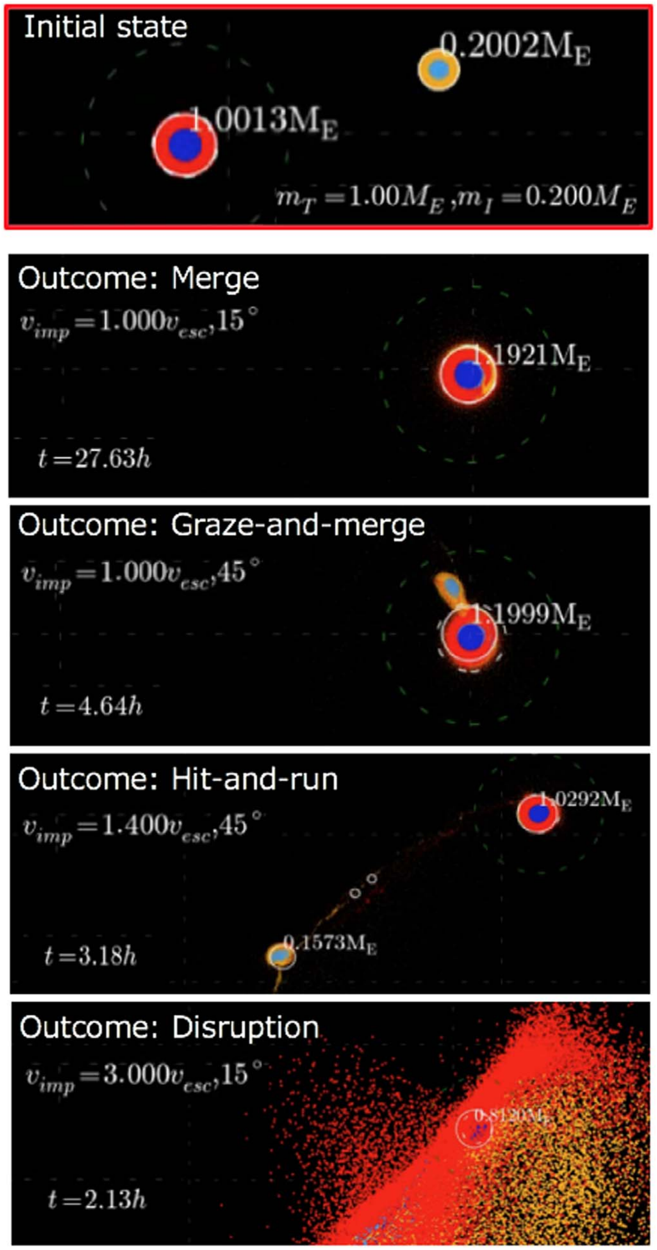

Figure 2. Different combinations of the four impact properties (predictors: mass of the target, projectile-to-target mass ratio, impact angle, impact velocity), which lead to different collision outcomes (responses). The SPH code allows easy visualization of the results in the form of short clips. For example, the top panel of the figure shows the initial state of the simulation (target and impactor before the collision). The other panels show the collision type for various impact velocities and angles. From top to the bottom: a merging event, resulting from a head-on collision at low-impact velocity; a graze-and-merge event, resulting from a collision at the most probable impact angle (45°, Shoemaker 1962) and low-impact velocity; a hit-and-run event, resulting from a collision at an impact angle of 45° and moderate impact velocity; a disruptive event, resulting from a head-on collision at high impact velocity. The time after the collision (in hours) is reported at the bottom left of each frame. The simulations come from Reufer (2011). Further details of the simulation database and detailed physical analysis are provided in Gabriel et al. (2019).

labels, that is known with some level of confidence. For example, in a giant impact, a set of predictors (e.g., impact angle, impact velocity, mass of the target) results in a collision outcome, such as merger or disruption (the label). In supervised learning, the dataset is split into training samples, validation samples (data used to measure generalization capability of the algorithm), and testing samples (data that do not affect training and are used as an independent measure of performance during and after training). In contrast, unsupervised methods do not label the data directly into classes but rather attempt to find patterns and trends underlying in the data. These algorithms (e.g., K-mean, Ahmad & Dey 2007) usually require an initial assumption on the data (e.g., number of clusters) and their results heavily depend on these initial assumptions.

We divide the algorithms into metric and non-metric, depending on their specific operating principles. Metric algorithms employ measures of similarities and distances to the predictors, whereas non-metric algorithms do not. Among the metric-based algorithms, we consider: SVMs (Hearst et al. 1998), which use a kernel to compute the inner product of all pairs of data in the feature space and implicitly projects the data in a higher-dimensional space where the data are linearly separable; and *K*-Nearest Neighbors (KNN, Duda et al. 2012), which uses a suitable similarity function/distance to evaluate the closeness of a new sample to samples stored in memory. Among the non-metric algorithms, we consider: Decision Trees (DT, Safavian & Landgrebe 1991), which construct a tree structure and explore nodes and leaves for both classification and regression; and Random Forest (Breiman 2001), which is an ensemble of multiple DT, where each tree is constructed by sampling a random set of attributes from the data. Each tree performs regression via a mean prediction and classification via majority voting. Ensemble methods (e.g., Bootstrap Aggregation or Bagging, Breiman 1996), where an ensemble of weak learners are combined to produce a stronger learner, are considered for both regression and classification tasks.

2.2.1. Classification Task: SVM

In SPH, the continuous fluid is represented as a Lagrangian set of particles that move with the flow. This allows easy visualization and supports analytical deductions (Asphaug et al. 2015, and references therein). We digest the dataset for the classifier by defining the qualitative outcome of each giant impact simulation according to four distinct classes of responses: merging, disruption, graze-and-merge, hit-and-run (e.g., Asphaug et al. 2006, 2015; Stewart & Leinhardt 2009), see Figure 2. In our classification, we distinguish between merging and graze-and-merge scenarios. The latter is a transient evolution that would eventually lead the projectile to merge with the target, but escaping bodies up to a third of the size of the progenitors can spin off during the collision (Asphaug & Reufer 2013). The outcome of the simulations (response, or class) is associated to four impact parameters (predictors): mass of the target, projectile-to-target mass ratio, impact angle, and impact velocity. The dataset has entries:

$$\{(M_{\mathrm{T}}, \gamma, \theta_{\mathrm{coll}}, v_{\mathrm{coll}}/v_{\mathrm{esc}}); \text{class}\} \quad (3)$$

where M_{T} is the mass of the target, $\gamma = M_{\mathrm{P}}/M_{\mathrm{T}}$ is the projectile-to-target mass ratio (M_{P} being the mass of the projectile), θ_{coll} is the impact angle and v_{coll} is the collision velocity normalized to the mutual escape velocity v_{esc} (Equation (1)). Matching between predictors and response is done during one of our movie days: four co-authors watched

Table 2
Excerpt of the Labeled Data for the Classification Task

Target Mass [$M_\oplus$]	Mass Ratio (projectile/target)	Impact Angle	Impact Velocity [$v_{\rm esc}$]	Collision Class
1	0.70	89.5	1.30	hit-and-run (flag: 1)
1	0.70	89.5	1.05	graze-and-merge (flag: 2)
1	0.70	22.5	1.00	merging (flag: 3)
1	0.70	22.5	4.00	disruption (flag: 4)
10^{-1}	0.70	30.0	1.50	hit-and-run (flag: 1)
10^{-1}	0.70	30.0	1.40	graze-and-merge (flag: 2)
10^{-1}	0.70	22.5	1.00	merging (flag: 3)
10^{-1}	0.70	22.5	4.00	disruption (flag: 4)

Note. The elements in columns first to fourth are the predictors (pre-impact conditions): $M_T \in [10^{-2}, 1]\mathrm{M}_\oplus$; $\gamma = M_{\rm P}/M_{\rm T} \in [0.2, 0.7]$; $\theta_{\rm coll} \in [0, 90]$; $v_{\rm coll}/v_{\rm esc} \in [1, 4]$. The elements in the fifth column are the responses (type of collision outcome). among the responses: hit-and-run cases are coded as #1; graze-and-merge cases are coded as #2; merging cases are coded as #3; and disruptive cases are coded as #4.

(This table is available in its entirety in machine-readable form.)

short movie clips of the simulations and, based on this visualization, agreed on the outcome of the simulations (class). When dealing with a complicated problem, a group of experts with varied experience in the same area have a higher probability of reaching a satisfactory solution than a single expert (Baruque & Corchado 2010). However, labeling error can still occur, due to subjectivity, data-entry error, or inadequacy of the information used to label each entry. Domains in which experts disagree are natural places for subjective labeling errors (Brodley & Friedl 1999, and references therein). To mitigate mislabeling and its negative effect on the performance of the classifier, the labeling of the dataset is performed by the domain experts with a majority vote. Taking a majority over many hypotheses, all of which proposed by different experts, reduces the random variability of the labels (Baruque & Corchado 2010). More advanced approaches to labeling (e.g., crowd-sourcing or weighted-voting, Rodrigues et al. 2013) or to labeling error mitigation (e.g., ensemble learning, Zhang & Ma 2012) are also possible, but they are beyond the scope of this pilot study.

An excerpt of the labeled data is reported in Table 2. The dataset for the classification task is published in its entirety in the machine-readable format. Among the available schemes, we selected a multi-class SVM (Hearst et al. 1998) as the algorithm that achieves the highest validation for the classification task (see Section 3.1). SVMs were introduced by Boser, Guyon & Vapnik (Boser et al. 1992), and they have become very popular because of their success in the handwritten digit recognition task. SVMs are ML algorithms that can discriminate between different classes given input data. They are considered primary examples of the so-called kernel methods.

Consider a set of given training vectors $\boldsymbol{x}_i \in \mathbb{R}^n$, $i = 1, \ldots., l$ that belong to two classes, and a class indicator vector $\boldsymbol{y} \in \mathbb{R}^l$ such that $y_i \in [-1, 1]$. The basic SVM algorithm solves the following primal optimization problem:

$$\min_{\boldsymbol{w},b,\eta} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\eta_i \tag{4}$$

subject to the following constraints:

$$y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geqslant 1 - \eta_i,\ \ \eta_i \geqslant 0. \tag{5}$$

Here, $\phi(\boldsymbol{x}_i)$ maps the training vectors $\boldsymbol{x}_i$ into a higher-dimensional space. $C \geqslant 0$ is the Tikhonov regularization parameter. Generally, the vector variable $\boldsymbol{w}$ lives in a high-dimensional space. Thus, one equivalently solves the following dual problem:

$$\min_\alpha \frac{1}{2}\alpha^T Q\alpha - \boldsymbol{e}^T\alpha \tag{6}$$

subject to:

$$\boldsymbol{y}^T\alpha = 0,\ \ 0 \leqslant \alpha_i \leqslant 0,\ i = 1, \ldots, l. \tag{7}$$

Here, $\boldsymbol{e} = [1, \ldots.,1]^T$ is a vector comprising all ones, Q is an $l \times l$ positive semi-definite matrix where $Q_{i,j} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_i)$. The kernel function $K(\cdot,\ \cdot)$ is defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_i) = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_i)$. After the optimization problem is solved via the primal-dual relationship, the optimal vector $\boldsymbol{w}$ satisfies the following relationship:

$$\boldsymbol{w} = \sum_{i=1}^{l} y_i\alpha_i\phi(\boldsymbol{x}_i). \tag{8}$$

Importantly, the decision (discriminative) function for the binary classification problem is mathematically described as:

$$\mathrm{sgn}(\boldsymbol{w}^T\phi(\boldsymbol{x}) + b) = \mathrm{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \tag{9}$$

This formulation holds when the problem has nonlinear decision surfaces because the input vector $\boldsymbol{x}$ is substituted by a properly selected mapping function ϕ that projects the training data into a suitable feature space (Shashua 2009). The choice of the function ϕ is done using k-fold cross-validation, which subdivides the training set in k subsets and trains the classifier (i.e., solve the primal-dual optimization problem) using only $(k - 1)$ subsets. The validation accuracy (i.e., percentage of correct classification) is computed—after training—on the kth subset. The procedure is repeated several times and the average validation accuracy is used to compare different schemes with different hyperparameters (i.e., the value of k and the function ϕ). The model with the highest validation accuracy is adopted.

Once the SVM is trained and validated, its performance is assessed by means of a confusion matrix computed on a testing set. The confusion matrix shows the degree to which the classifier is confused when it makes predictions. Each row represents the instances in a predicted class while each column

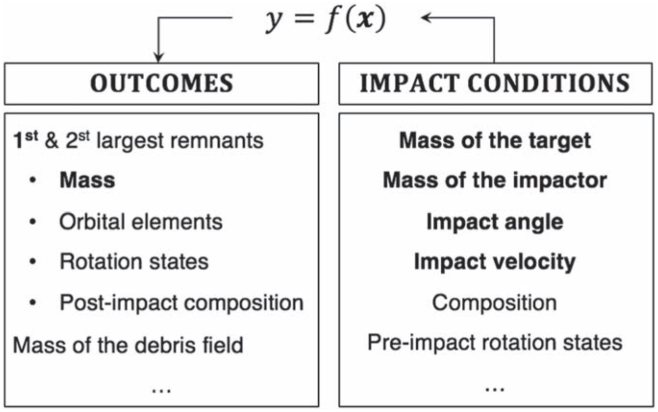

Figure 3. A surrogate model (e.g., neural network) is able to generalize the functional relationship $y = f(\boldsymbol{x})$ between real-variable input $\boldsymbol{x}$ (impact conditions, right column) and outputs y (collision outcomes, left column). Training occurs on N data of the type: $\{\boldsymbol{x}; y\}_i$ = {predictor; label}$_i$, $i = 1, \ldots, N$. Examples of impact conditions (predictors) and outcomes (labels) are shown in the right-hand and left-hand columns, respectively. In this pilot study, we train a neural network to associate four impact conditions (mass of the target, projectile-to-target mass ratio, impact angle, impact velocity) to accretion efficiency (or mass of the largest remnant, Equation (11)).

represents the instances in an actual class (Ting 2010). For example, for the binary subproblem of classification between graze-and-merge (GnM) and hit-and-run (HnR), the confusion matrix has the form:

	Actual: HnR	Actual: GnM
Predicted: HnR	a	b
Predicted: GnM	c	d

The diagonal elements are the instances of correct classifications, while the off-diagonal values account for misclassifications. In this example, the total number of actual hit-and-run events is "$a + c$"; after training, the SVM classifies correctly "a" events and misclassifies "c" events as graze-and-merge. The accuracy of the classifier is computed as the percentage of true positives (correct predictions) over total number of sample:

$$\mathrm{AC}\,[\%] = \frac{a + d}{a + b + c + d} \times 100. \tag{10}$$

Classification problems with a number of classes greater than 2 are decomposed into multiple binary classification problems, according to different transformation techniques (e.g., one versus one and one versus rest strategies, Bishop 2006). The choice of a specific technique is also part of hyperparameter optimization.

2.2.2. Regression Task: NNs

Whereas classifiers are able to handle discrete, qualitative responses, a regressor is a surrogate model that is able to mimic the parent SPH input-output function to predict continuous (real-variable) outputs given the input parameters (predictors), see Figure 3. This scheme provides a synthesis of the collision outcome in terms of a set of output properties of interests (e.g., mass of the largest remnants, their post-collision orbital elements, etc.) by learning from large planetary formation datasets of collision. Running the surrogate model drastically reduces the computational time with respect to full SPH simulations (from hours to seconds). We design a surrogate model for the prediction of accretion efficiency (i.e., the mass of the largest remnant of the collision) at several times the collision timescale, see Equation (2). After this time, pressure and temperature gradient forces are no longer acting and the resulting scenario (largest remnants and their orbital properties) can be treated using N-body integrator rather than hydrocodes. We use the definition of accretion efficiency by Asphaug (2010):

Table 3
Excerpt of the Data for the Regression Task

Target Mass [$M_\oplus$]	Mass Ratio (Projectile/ Target)	Impact Angle	Impact Velocity [$v_{\rm esc}$]	Accretion Efficiency (Equation (11))
1	0.70	52.5	1.15	0.02
1	0.70	22.5	3.00	−0.58
1	0.70	45.0	1.30	0.02
10^{-1}	0.70	15.0	1.40	0.90
10^{-1}	0.20	15.0	3.50	−1.52
10^{-1}	0.35	15.0	3.50	−1.25
10^{-2}	0.70	60.0	1.70	0.00

Note. The elements in columns first to fourth are the predictors (pre-impact conditions): $M_T \in [10^{-2},\ 1]\,M_\oplus$; $\gamma = M_{\rm P}/M_{\rm T} \in [0.2,\ 0.7]$; $\theta_{\rm coll} \in [0,\ 90]$; $v_{\rm coll}/v_{\rm esc} \in [1, 4]$. The elements in the fifth column are the responses (accretion efficiency ξ) as post-processed by Gabriel et al. (2019).

(This table is available in its entirety in machine-readable form.)

$$\xi = \frac{(M_{\rm LR} - M_{\rm T})}{M_{\rm P}} \tag{11}$$

where $M_{\rm LR}$ is the mass of the largest remnant, $M_{\rm T}$ is the mass of the target body and $M_{\rm P}$ is the mass of the projectile. For each simulation in our dataset, the largest remnants are identified as discussed in Section 2.1. A summary of the data is reported in Table 3. The dataset for the regression task is published in its entirety in machine-readable format.

This effort of the work is entirely independent from the classification of Section 3.1. For this task, a NN is trained, validated and tested to replace the more computationally expensive parent numerical models (e.g., the full SPH simulation) in the prediction of accretion efficiency. NNs are able to learn (i.e., improve the performance of a specific tasks) from data by modeling the functional relationship between inputs and outputs, which is exemplified by labeled data. NNs consist of many mathematical units called neurons, which communicate in a parallel fashion through weights that represent the strength of the corresponding synapses. Neurons are the basic processing units for the network and are characterized by an activation function $h(\cdot)$. Additive nodes with activation functions have the following structure:

$$G(\boldsymbol{a}_i, b_i, \boldsymbol{x}) = h(\boldsymbol{a}_i^T \boldsymbol{x} + b_i) \tag{12}$$

where $\boldsymbol{a}_i \in \mathbb{R}^m$ and $b_i \in \mathbb{R}$. The tanh-sigmoid function is a common activation function for shallow neural networks

$$h(s) = \frac{2}{1 + \exp(-2s)} - 1. \tag{13}$$

For deeper architectures, such as Convolutional Neural Network (CNN, Krizhevsky et al. 2012), other activation functions (e.g., ReLu, Rectified Linear Unit) are more commonly used. Neurons are organized by layers. For this application, we adopt a shallow network comprising one input layer, one hidden layer and an output layer. The hidden layer is assumed to have a specified number of neurons S. The overall process begins with a summation of each input with the correspondent weights (synapses) and then further processing by an activation function. In regression problems, the overall NN output function is typically represented as follows:

$$f_S(\boldsymbol{x}) = \sum_{i=1}^{S} \beta_i h_i(\boldsymbol{x}) = \sum_{i=1}^{S} \beta_i G(\boldsymbol{a}_i, b_i, \boldsymbol{x}) \tag{14}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\beta}_i \in \mathbb{R}^m$. The weights $\boldsymbol{a}_i$ an biases b_i are determined during the training process which implies minimization of a loss function. For regression problems, the typical loss function is the Mean Square Error (MSE), i.e.:

$$\mathrm{MSE}(\boldsymbol{a}_i, b_i) = \frac{1}{N} \sum_{i=1}^{N} (f_S(\boldsymbol{x}_i) - y_i)^2 \tag{15}$$

where $\{\boldsymbol{x}, y\}_i$ is the associated training set of size N.

In this paper, the regressor is trained, validated and tested on data of the type:

$$\{(M_{\mathrm{T}}, \gamma, \theta_{\mathrm{coll}}, v_{\mathrm{coll}}/v_{\mathrm{esc}}); \xi\} \tag{16}$$

while the composition is kept as a parameter. The dataset is subdivided in training, validation and testing subsets, typically in proportion 70%–15%–15%, respectively. Network training is performed using the training set and involves the fitting of the network parameters (weight $\boldsymbol{a}_i$ and biases b_i) via minimization of the loss function (Equation (15)). A common approach to training involves backpropagation or Stochastic Gradient Descent (Schmidhuber 2015). A training step (or epoch) consists in a round of predictions for the predictors $\boldsymbol{x}_i$ in the training set, followed by backpropagation of the residuals between targets and prediction and update of weights $\boldsymbol{a}_i$ that reduce the MSE. At each epoch, the performance of the network (and progress toward a successful training) is evaluated in terms of MSE, Equation (15), which is expected to decrease as the number of epochs increases, thus indicating progressive improvement in the performance (i.e., learning).

Both the validation and testing sets are employed to protect against overfitting of the training set (Bishop 1995). The training process is not a simple interpolation of the training set, but it rather involves the search for families of parametric functions (i.e., the neural network) that globally fit the data (generalization). The validation procedure consists in the search of those network hyperparameters (e.g., the learning rate or numbers of hidden neurons, which are not learned during training) that minimize the MSE on the validation set. In addition to helping to protect against overfitting, the testing set is used for an independent assessment of the generalization capabilities of the network; i.e., the behavior of the MSE on an unseen ensemble of data. Properly trained networks ensure that the data in the validation and the testing sets follow the same probability distribution as the data in the training set. At every training epoch, the MSE for validation and testing is computed. The training is completed when the MSE on the validation set does not further decrease for six consecutive training epochs.

In addition to the MSE, the overall process is also evaluated also in terms of regression value R, which measures the degree of correlation between outputs and targets. This quantity is analogous to the SVM classification accuracy for real-variable data. The regression value R is a non-dimensional quantity and it allows us to compare the performance of different approaches to the problem (e.g., data-driven approach versus data interpolation) with respect to the data at testing. An optimal result shows low MSE values (i.e., close to zero) and a high degree of correlation between predictions and targets (i.e., a R value close to 100%) on the testing set.

3. Results

The trained response functions (classifier of collision types and regressor of accretion efficiency) are presented in the following two sections. We also discuss their prediction performance with respect to the labels of the entries in the datasets.

3.1. Classifier of Collision Outcomes

The classifier of collision outcomes maps the four impact properties (mass of the target, projectile-to-target mass ratio, impact angle, impact velocity) into one of the following types of collisions: merging, disruption, hit-and-run, graze-and-merge. The classifier is trained, cross-validated and tested as discussed in Section 2.2.1. The ensemble of 769 labeled SPH simulations in Table 2 is split in a training dataset (90%) and a testing dataset (10%) via random sampling without replacement. The training set is used for training the network with 10-fold cross-validation, which allows us to perform hyperparameter optimization for what concerns the best kernel feature parametrization. We find that a quadratic kernel ($K = \phi^T\phi = (k^Tx + m)^2$) achieves the best cross-validation accuracy (91.0%).

The performance of the classifier, in terms of its confusion matrix, is shown in Figure 4, left-hand panel. The performance is evaluated on the testing set, corresponding to 77 entries, which was not used for training and cross-validation. Testing the algorithm on this separate dataset provides an independent, additional assessment of the performance of the classifier on unseen data. We achieve an overall accuracy above 93% at testing. However, certain regimes are characterized by more misclassifications (e.g., disruption versus merging) than others (e.g., hit-and-run). Those classes that are characterized by high false negative rates prevent the classifier from achieving 100% accuracy at testing (i.e., a fully diagonal confusion matrix), which is found to be indicative of confusion along the decision boundaries between regimes; we will address this point in more detail in Section 4.2 (Figure 7, left-hand panel).

The classifier is intrinsically a 4D scheme, with as many dimensions as the number of predictors (impact properties). The algorithm describes the outcome in parameter space by means of decision hyper-surfaces, which mark the transition between different regimes. To better appreciate these features, the parameter space can be sectioned in 2D slices; an example of this map is given in Figure 4, right-hand panel, for a mass of the target $M_{\mathrm{T}} = 0.1\,M_{\oplus}$ and similar-mass projectile ($\gamma = M_{\mathrm{P}}/M_{\mathrm{T}} = 0.7$). The collision type is mapped into a space of collision velocity (in units of mutual escape velocity) and impact angle. We recognize four distinct collision regimes, whose decision boundaries are the traces of the decision hyper-surfaces suggested by the classifier. Each regime is a phase, in which the collision outcome is qualitative similar; i.e., a scaling law is expected to apply.

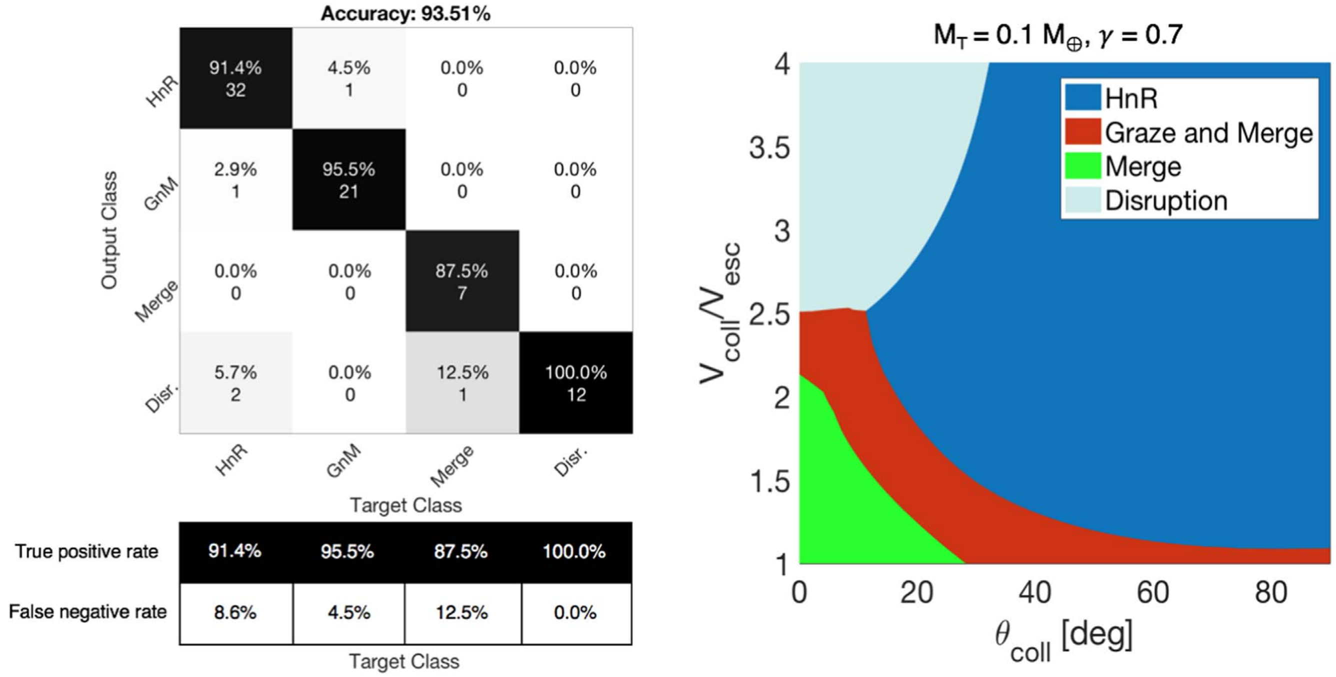

Figure 4. Left-hand panel: confusion matrix of the 4D classifier, quantifying the degree of accuracy of the classification on the testing set. The elements on the diagonal of the confusion matrix represent those instances that have been correctly classified by the SVM (true positives). Conversely, each extra-diagonal element represents the number of misclassifications with respect the SPH data (i.e., the labels). The number of misclassifications is added along each column to compute the false negative rates. Overall, we achieve a true positive rate of 91.4% on the hit-and-run (HnR) class, 95.5% for the graze-and-merge (GnM) class, 87.5% for the merge class and 100.0% for the disruption class. The confusion matrix is close to be fully diagonal; the accuracy—which is computed as the mean value of the true positives over the whole population, Equation (10)—is above 93%. Right-hand panel: decision boundaries for the collision type, as predicted by the classifier for a mass of the target $M_{\rm T} = 0.1\ M_\oplus$ and a mass ratio between the projectile and the target $\gamma = 0.7$. The impact velocity spans a range between 1 and 4 times the mutual escape velocity (Equation (1)) while the impact angle ranges from head-on to grazing configurations.

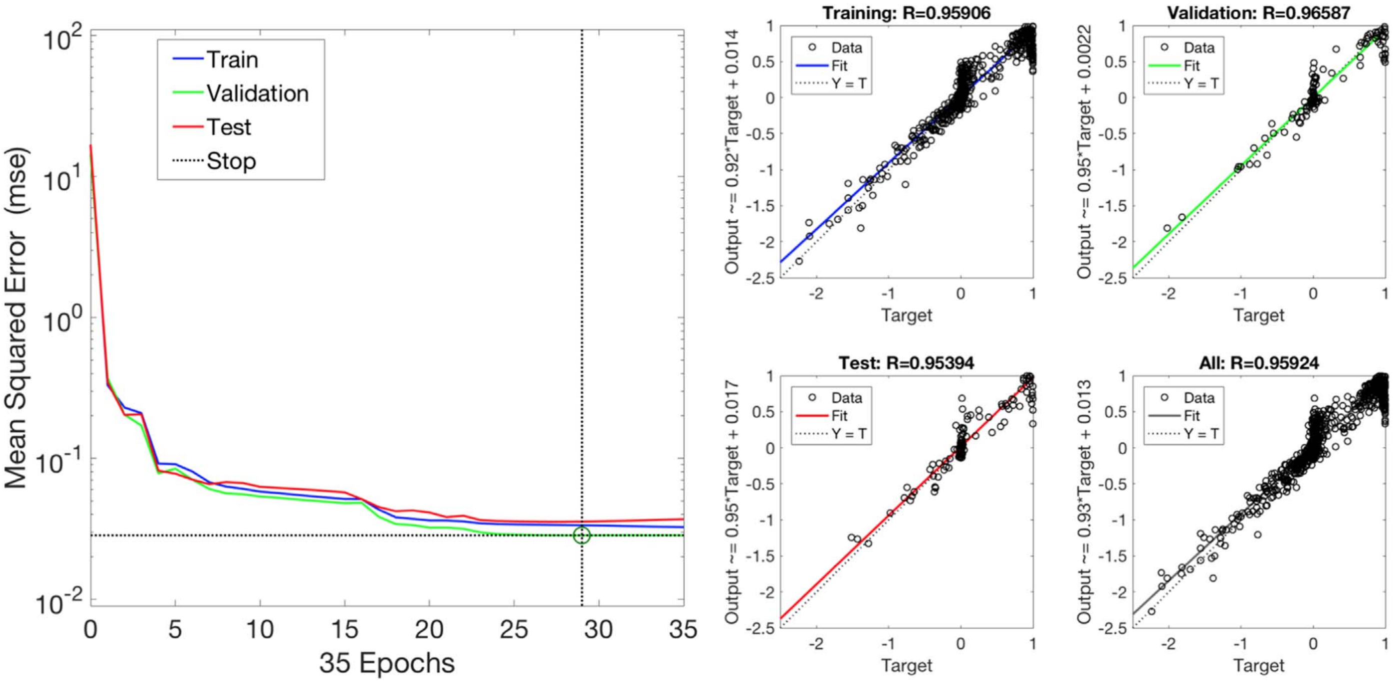

Figure 5. Left-hand panel: evolution of the Mean Square Error (MSE) for training, testing and validation, for increasing epochs of training. When validation is concluded, the average plateau value of the testing MSE is 0.04. This quantifies the global uncertainty of the surrogate model in mimicking the parent numerical model; i.e., the SPH simulations. Right-hand panel: correlation between predictions and target, and overall fitting with respect to an expected 1:1 line. The regression index R is about 96% (average), close to the optimal value of 100%.

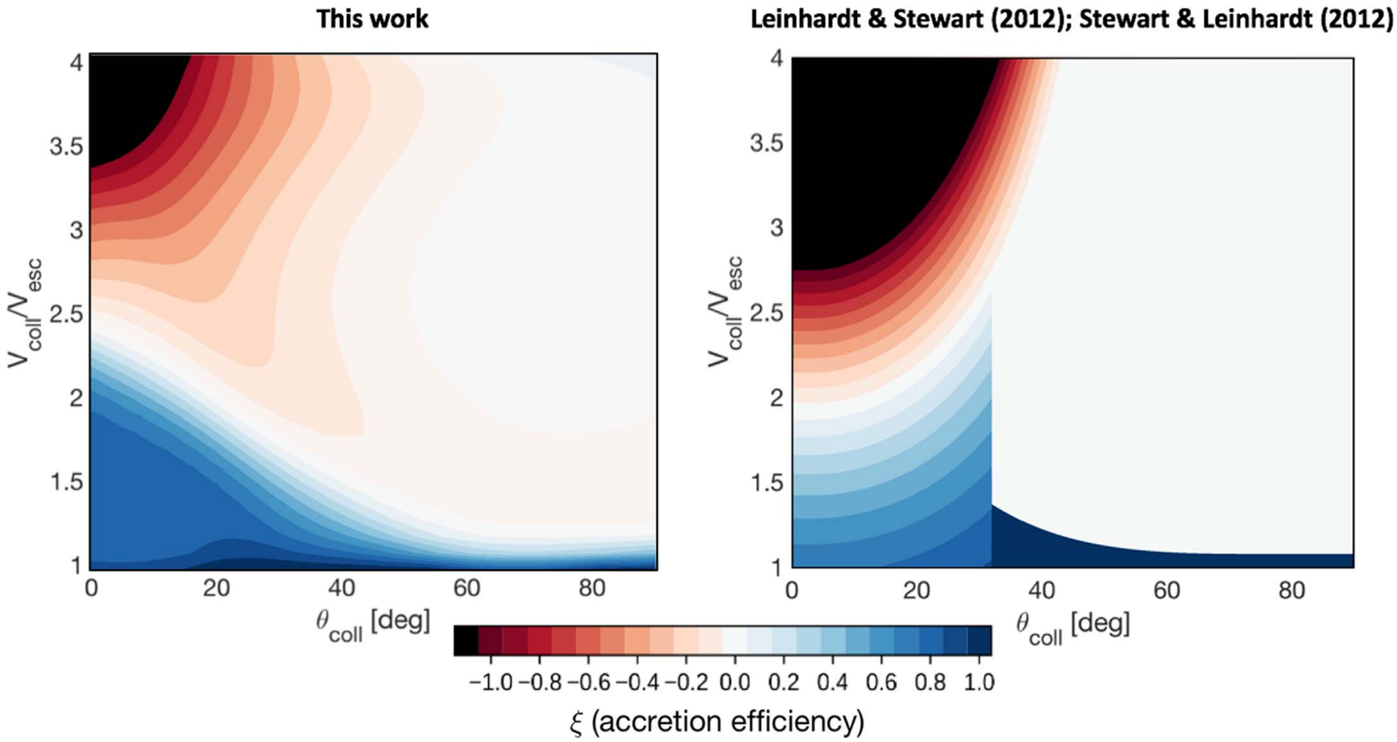

Figure 6. Left-hand panel: map of accretion efficiency—Equation (11)—as predicted by the neural network (Section 3.2). Right-hand panel: map of collision outcome and accretion efficiency generated using the scaling laws proposed by Leinhardt & Stewart (2012), for the same combination of mass of the target and mass of the projectile, using the values $c^\star = 1.9$ and $\bar{\mu} = 0.36$, which were fit to hydrodynamic planets. Impact velocity (y axis) ranges between 1 to 4 $v_{\rm esc}$, impact angle (x axis) ranges from head-on to grazing, $M_{\rm T} = 0.1\ M_\oplus$, and $\gamma = M_{\rm P}/M_{\rm T} = 0.7$. The grid was sampled in steps of $0^\circ\!.01$ and $0.01 v_{\rm esc}$; the color for each mesh face is dictated by the vertex with the smallest index. Accretion efficiency shows a rich range of outcomes, which includes transitions from accretion (cooler colors) to disruption (warmer/black colors), to hit-and-run (almost net-zero accretion; white colors).

3.2. *Regressor of Accretion Efficiency*

The neural network has four input neurons (as many as the impact properties), one hidden layer, and one output layer which predicts accretion efficiency. The dataset of Table 3 is composed by 810 simulations, whose predictors (i.e., the impact properties) are internally scaled in a min–max procedure. The training is performed using the Levenberg–Marquardt algorithm (Demuth et al. 2014) on 70% of the overall dataset. The rest of the data is split between a validation set (15%) and a testing set (15%). The dataset is split via random sampling without replacement to assure that the data in the three sets follow the same probability distribution. Figure 5, left-hand panel, shows learning dynamics in terms of the evolution of the MSE for training, validation and testing, at different epochs of training procedure. For the hidden layer, the choice of 10 hidden neurons gives the lowest MSE at validation. The testing MSE converges to an error level of about 0.04. This value is an estimate of the global accretion efficiency error, as it quantifies the (squared) residual between the values predicted by the regressor and the values of accretion efficiency of the SPH data in the testing set. The training error is also 0.04, while the validation error is about 0.03. Figure 5, right-hand panel, shows the correlation index at the end of the training procedure, whose value is above 95% on testing.

For the classifier of collision outcome, the regressor maps accretion efficiency in a 4D parameter space. For a mass of the target $M_{\rm T} = 0.1\,M_\oplus$ and similar-mass projectile ($\gamma = M_{\rm P}/M_{\rm T} = 0.7$), Figure 6, left-hand panel, shows a 2D map (slice of the parameter space) of accretion efficiency in a plane of impact velocity (in units of mutual escape velocity) and impact angle. The grid has a step of $0^\circ\!.01$ along the impact angle axis ($\theta_{\rm coll}$) and 0.01 along the velocity axis ($v_{\rm coll}/v_{\rm esc}$). Accretion efficiency is color-coded such that the outcome varies from perfect merging (dark blue) to partial accretion (light blue) to partial erosion to disruption (redder colors and black for $\xi \leqslant -1$). The corner of the face that has the smallest indices determines the constant color of each mesh face. Catastrophic disruption is achieved when the mass of the largest remnant is less or equal to the half of the total mass of the system ($M_{\rm T} + M_{\rm P}$). Given that $M_{\rm P} = \gamma M_{\rm T}$, catastrophic disruption is characterized by an accretion efficiency equal or less than $\xi_D = 0.5$–0.5γ, with disruption threshold ($\xi_D = -0.21$ for $\gamma = 0.7$).

4. Discussion

High-resolution SPH simulations have been used to train, validate and test a classifier of collision type (Section 3.1) and a regressor of accretion efficiency (Section 3.2). Together with the prediction of the type of collision (e.g., merging versus disruption), real-variable collision outcomes (e.g., mass of the larges remnants, their post-collision orbits) are needed to realistically simulate collisions in an N-body dynamical evolution. The regression of these quantities can be done by means of a neural network that is able to map pre-impact conditions into outcomes (Figure 3). In this work, we present a first machine-learned regressor that predicts the accretion efficiency at many times the collision timescale—Equation (2).

The two surrogate collision models (classifier and regressor) describe a 4D parameter space in terms of mass of the target, projectile-to-target mass ratio, impact velocity and impact angle. However, interpretation by the human operator is preferably done on a 2D section (slice) of the 4D parameter space. For example, SVM decision boundaries and accretion efficiency are predicted for $M_{\rm T} = 0.1\,M_{\oplus}$ and $\gamma = M_{\rm P}/M_{\rm T} = 0.7$ in Figure 4, right-hand panel, and Figure 6, left-hand panel, respectively. The map of the accretion efficiency unveils a richer background scenario in regions were the collision outcome seemed homogeneous according to the classifier. Graze-and-merge and merging are somewhat inefficient in delivering mass to the target because a portion of the projectile as high as 50% can escape accretion. Because of its typical grazing nature, the hit-and-run regime is characterized by accretion efficiency close to 0, which is within the error associated with the training. However, at the most probable impact angle (i.e., 45°, Shoemaker 1962), lower-energy hit-and-run cases are indistinguishable from partially accreting graze-and-merge events, while partial erosion starts to dominate above $v_{\rm coll}/v_{\rm esc} \sim 2$. Overall, the target is likely to be slightly eroded in the hit-and-run regime, but the second largest remnant (i.e., the surviving projectile) underwent the highest collision and tidal stresses because the energy of the impact is partitioned equally in the two bodies.

4.1. Comparison with Scaling Laws

Predicting the outcome of a giant impact without performing a full hydrodynamics simulation has already been the subject of multiple studies, leading to the formulation of scaling laws (e.g., Davis & Ryan 1990; Benz & Asphaug 1999; Leinhardt & Stewart 2012). A scaling law is an analytic relationship between impact properties (e.g., mass ratio, impact angle, and impact velocity) and its outcome for any collision in a physical regime (e.g., between gravity-dominated bodies), assuming invariance with respect to one property, usually the mass of the target. Hydrodynamical simulations are used to fit the parameters of the relationship and, ideally, account for the transition between the different regimes.

Here, we compare our results with one such law by Leinhardt & Stewart (2012), who proposed scaling the collisions according to the ratio between the specific impact energy and the catastrophic disruption threshold Q^{*}_{RD}—which is the specific energy required to disperse half the total colliding mass (for non-grazing collision). The reference specific energy is first computed for head-on collisions between equal-mass bodies and then corrected for the mass ratio and impact angle.

The left-hand panel of Figure 6 shows the map of accretion efficiency (predicted using our regressor), again for $M_{\rm T} = 0.1\,M_{\oplus}$ and $\gamma = 0.7$. On the right-hand panel of Figure 6 is the analogous map generated using the scaling laws for hydrodynamic bodies proposed by Leinhardt & Stewart (2012) and Stewart & Leinhardt (2012). The fit parameters in their model that are most relevant to our results are $c^{\star} = 1.9 \pm 0.3$ and $\bar{\mu} = 0.36 \pm 0.01$ and were thus used to generate the right-hand panel of Figure 6. However, our data-driven approach and the empirical, physics-based energy scaling by Leinhardt & Stewart (2012) are different in two fundamental aspects: (1) the underlying dataset of simulations that were used for fitting procedures; and (2) the fitting methodology. Because of these differences, we keep the comparison between the two results shown in Figure 6 qualitative, and we aim to highlight the similarities and differences between them.

Leinhardt & Stewart (2012) segregate collisions into grazing and non-grazing according to the critical impact parameter $b_{\rm crit} = \sin\theta_{\rm crit} = R_{\rm T}/(R_{\rm T} + R_{\rm P})$ (Asphaug 2010) (see vertical line in the right-hand panel of Figure 6). However, this relationship was introduced by Asphaug (2010) as a geometrical guideline and it was not intended for the purpose of accurately predicting hit-and-run events. The description of the parameter space by our surrogate models does not show a hard transition between grazing and non-grazing scenarios based on the critical impact parameter value. We unveil the occurrence of hit-and-run events at angles lower than the critical value, as discussed further in Gabriel et al. (2019).

In the grazing domain (on the right-hand side of the critical impact angle), Leinhardt & Stewart (2012) assume that all collisions are hit-and-run in nature for sufficiently high impact velocities and accretion efficiency is assumed to be zero; i.e., the largest and second largest remnant masses are equal to the target and projectile mass respectively. In the hit-and-run regime, we confirm that the accretion efficiency is consistently close to zero (within the accuracy of the regressor) in the majority of the parameter space, but partial accretion or erosion scenarios are recorded close to transition with other regimes (Figure 6, left-hand panel).

Our surrogate models show that perfect merging is rare—it may happen for low-impact velocities and mid-impact angles (about 15°–50°, again, within the accuracy of the regressor). Grazing events need to eject some material to release angular momentum, which would otherwise lead to unphysical spin (Asphaug & Reufer 2013). Most of the regions categorized by the classifier as merging or graze-and-merge are actually partial accretions rather than perfectly merging. The underlying events were categorized as such because the lost mass is in the form of debris.

At the boundary between the hit-and-run and graze-and-merge regimes (low-impact velocity and high impact angle), the transition curve by our classifier of collision outcome (decision boundary in Figure 4, right-hand panel) is found to be similar to that by Stewart & Leinhardt (2012), who use the hit-and-run velocity criterion from Kokubo & Genda (2010) to mark the transition. However, across this region we also observe a rapid decrease in accretion efficiency—from merging to hit-and-run values—as the impact velocity increases (Figure 6, left-hand panel).

We also point to the similarity between the transition curves from our classifier (Figure 4, right-hand panel) and those of Leinhardt & Stewart (2012) (Figure 6, right-hand panel) at the boundary between the hit-and-run and the partial erosion regimes. For non-grazing scenarios, Leinhardt & Stewart (2012) determine the outcome by specific impact energy and γ solely. Accretion efficiency ranges from partial accretion (cool colors in Figure 6) to catastrophic disruption (black color); catastrophic disruption for this combination of parameters is $\xi \leqslant -0.21$. In addition to the differences in the assumed boundaries between regimes, our simulations are based on different underlying datasets. Our data-driven model is based on simulations from Reufer (2011), whereas the hydrocode simulations used in Leinhardt & Stewart (2012) are from diverse source models (e.g., Benz et al. 2007; Marcus et al. 2009, 2010b). Gabriel et al. (2019) demonstrate that the range of disruption thresholds exhibited by our dataset are close

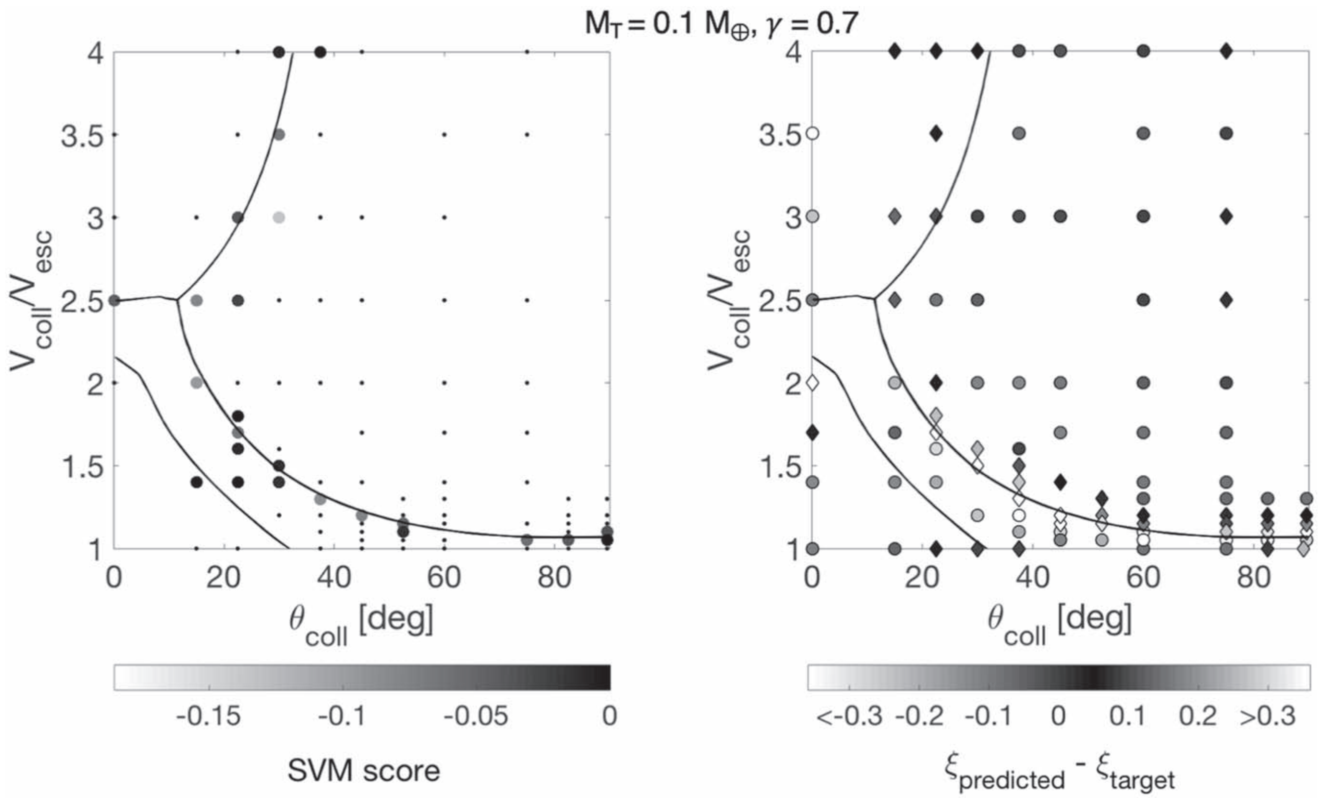

Figure 7. Left-hand panel: local SVM score of the SPH simulations by the classifier of collision outcomes (small dots: correct predictions; color-coded data points: misclassifications). Right-hand panel: residuals between the predictions by the regressor and the SPH data, for the same combination of mass of the target and mass of the projectile (diamonds: positive residuals; dots: negative residuals). Impact velocity ranges between 1 and 4 times the mutual escape velocity, impact angle ranges from head-on to grazing, $M_T = 0.1\ M_\oplus$ and $\gamma = M_P/M_T = 0.7$. High uncertainty is recorded along the decision boundaries (black curves), where misclassifications and inaccurate predictions tend to cluster. In these regions, additional SPH simulations are required to further reduce the confusion of the ML algorithms.

to the uncertainty of the disruption threshold of Leinhardt & Stewart (2012). Thus, we do not consider the difference in disruptive behavior observed in Figure 6 to be significant.

4.2. A Guide to Parameter Space Exploration

The designed classifier and regressor have high global accuracies (Figure 4, left-hand panel and Figure 5), but misclassifications and inaccurate predictions can still occur locally in the parameter space. For a classifier, the local degree of confusion is quantified by the SVM classification score, which is the signed distance to the decision boundary. If the classifier is asked to predict the class for a labeled data, then a positive, large score on the correct label means that the prediction is correct (the data are within the subspace of the correct class), while a negative score indicates misclassification; the more negative the value, the higher the signed distance from the decision hyper-surface. The decision boundaries—transition curve from a collision regime to another—are in regions where the score tends to be negative, as the outcome is more sensitive to slight variations in the pre-impact conditions and mislabeling is likely to occur. This is evident in Figure 7, left-hand panel, which shows the classification scores for the data with $M_T = 0.1\,M_\oplus$ and similar-mass projectile ($\gamma = M_P/M_T = 0.7$), in a plane of impact velocity and impact angle. Correct predictions are represented using small dots, while misclassified data points are color-coded according to their score (i.e., signed distance from the true classification boundary). The larger the absolute value of the score, the more severe the misclassification. The decision boundaries from the classifier are also reported (black curves). As expected, misclassification occurs more often along the boundaries.

A similar trend is observed in Figure 7, right-hand panel, where the predicted values for accretion efficiency are locally compared directly to the SPH data, again for $M_T = 0.1\,M_\oplus$ and similar-mass projectile ($\gamma = M_P/M_T = 0.7$), in a plane of impact velocity and impact angle. For the regressor, the local accuracy is quantified in terms of the residuals between predictions and targets (geometric distance). Inaccurate predictions are more distant with respect to their corresponding SPH data than accurate predictions. A positive (negative) residual indicates that the regressor is overestimating (underestimating) accretion efficiency with respect to the target value. In the right-hand panel Figure 7, predictions with positive residuals are represented using diamonds, while predictions with negative residuals are represented using dots. For the whole datasets (810 entries), 49% of the predictions have positive residuals and the remaining 51% cases have negative values. Therefore, the regressor is found to not systematically overestimate or underestimate accretion efficiency. We note that inaccurate predictions occur near the decision boundaries (black curves), which is to be expected. Local accuracies are thus expected to vary depending on location in the parameter space. The residual distribution is well approximated by a Gaussian fit centered at zero with 1σ value equal to 0.18. Large areas are characterized by residuals <0.1, and few cases (less than 1%) have residuals up to 0.66 near transition regimes (absolute value, accretion efficiency units).

The distributions of SVM scores (local uncertainties for the classifier) and residuals (local uncertainties for the regressor) provide a guideline toward the completeness of the dataset by indicating those regions of the parameter space that require additional simulations. The decision boundaries must not be intended as binary hard boundaries between different regimes of giant impact outcome, but rather as indicators that the outcome is gradually transitioning from a regime to another. Examples are the transition regions between graze-and-merge and merging (low-impact angle and low-impact velocity, left-hand bottom corner of the panels in Figure 7), and hit-and-run and graze-and-merge (high impact angle and low-impact velocity, right-hand bottom corner of the panels in Figure 7).

The extent of the transition regions is given by the size of the clusters of inaccurate classifications and predictions (uncertainty band). For the classifier, the uncertainty band quantifies the degree of confusion of the field experts during the labeling process. This confusion arises because, near and along the decision boundaries, the outcomes of collision events seem alike or are unclear to the experts performing the labeling. These cases include the distinction between impactor disruption (e.g., Leinhardt & Stewart 2012) and hit-and-run. Furthermore, in proximity of certain decision boundaries, the outcome of a collision is highly sensitive to small changes in the impact parameters. For this reason, misclassifications correlate with inaccurate predictions by the regressor in the transition regions. Accretion efficiency is a real-number physical quantity and its transitions are smooth due to the occurrence of runner disruption at the boundary between erosive and hit-and-run collisions. However, the local gradient can be large and more simulations may be needed for the regressor to resolve the region; i.e., to accurately learn the functional relationship between pre-impact conditions and accretion efficiency along the decision boundaries.

Meanwhile, in regions where classification is exact and regression is accurate, one can avoid running a full SPH simulation to figure out the outcome because the classifier is certain in the prediction of the type of collision and the regressor is able to mimic the parent model at high fidelity.

5. Conclusion and Future Work

We have applied machine learning (ML) to explore a large dataset of SPH simulations for giant impacts (Reufer 2011; Gabriel et al. 2019). The relationship between beginning state (e.g., target mass, projectile mass, impact velocity and impact angle) and end state (impact outcome) has been mapped using two approaches. The result is a prototype of a full surrogate model of planet-forming giant impacts, which does not suffer from assumed physical models and which runs in a fraction of a second compared to days of simulation effort. This enables a fine—and fast—mapping of the parameter space to a known level of accuracy.

First, we train, validate and test a SVM (Hearst et al. 1998) to predict the type of the collision among four classes: merger, graze-and-merge, hit-and-run, and disruption. The classifier has global accuracy above 93% at testing (Figure 4, left-hand panel), but local misclassifications are found to occur in proximity of the decision boundaries (Figure 7, left-hand panel). Second, we train a neural network to predict the accretion efficiency; i.e., mass of the largest remnant of the collision. The network has a global error level of 0.04 (MSE between predictions and the dataset of SPH accretion efficiencies) and regression index above 95% at testing (left-hand and right-hand panel in Figure 5, respectively). Locally in the parameter space, residuals can reach 0.66 in accretion efficiency units (absolute value) but are generally lower, depending on the parameter region (Figure 7, right-hand panel). These functions—classifier of collision outcome and regressor of accretion efficiency—are called surrogate models because they provide a synthesis of the collision outcomes without the need to run a full hydrodynamical simulation. They are derived by generalizing the functional relationship between impact properties and outcomes, which are derived from the SPH simulations, to the whole parameter space within the ranges of the dataset (Table 1 and Figure 1). The use of surrogate models avoids the need to perform additional simulations over the entirety of the parameter space, which would be computationally inefficient given the large number of parameters and the requirement for high-resolution simulations to produce reliable outcomes.

The present training has been done using a dataset that is sparse in many regions of importance. One feature of ML is that the surrogate models can be easily updated if the training landscape is expanded as new simulations become available. Future collision surrogate models will benefit from the publication of datasets available to researchers in the community. A proposed list of impact conditions and correspondent collision outcomes for use in realistic *N*-body dynamical studies of planetary formation can be found in Figure 3. Additional interesting outcomes include the thermodynamic history of the hydro-particles (pressure, temperature, and density) which provides insights into the composition and size distribution of the debris field.

For the present work, we have trained on giant impacts in the gravity regime, where material strength plays a negligible role in the mass of post-collision remnants. In our future work, we will extend the parameter space to small giant impacts that involve bodies hundreds to thousands of kilometers diameter, colliding at around their mutual escape velocities, at hundreds to thousands of meters per second. In this regime, friction plays a non-negligible role (e.g., Jutzi 2015; Elkins-Tanton & Weiss 2017). New inroads have been made into SPH modeling of friction-governed planetary collisions (Jutzi 2015; Emsenhuber et al. 2018; Sugiura et al. 2018), which have revealed its importance in thousand-kilometer-scale (embryo–embryo) collisions.Furthermore, collisions in the friction regime have also been studied using soft-sphere discrete element (DEM) contacts in code PKDGRAV (Schwartz et al. 2012), which has been applied to asteroid family formation (Michel et al. 2001, 2004), ejecta cloud evolution (Schwartz et al. 2016), and comet formation through catastrophic disruption (Schwartz et al. 2018). The angle of internal friction and material composition (e.g., icy versus chondritic, Schwartz et al. 2018) are found to have a significant effect on the mass of the largest remnant (Ballouz et al. 2014, 2015). On asteroids, intergranular cohesion (Scheeres et al. 2010) becomes a sizeable source of tensile strength, which may affect the impact outcome. Resolving these complex physics requires higher numerical resolution and much more computational overhead per timestep of evolution. At the larger extreme, there are few sets of data regarding giant impacts for planets larger than the Earth (see Marcus et al. 2009, 2010a, 2010b; Liu et al. 2015; Kegerreis et al. 2018b). A primary challenge here is the reliable treatment of massive atmospheres. The same techniques of surrogate model development can be applied to these simulations, ultimately forming a general surrogate model for similar-sized planetary collisions at every scale; however, to date, no data table has been published at every scale.

The surrogate model is only as good as the post-processing of the physical simulations that gives us the derived outcomes for each run. The masses of the final bound remnants, and their velocities, rotations and compositions, must be reliably determined. In this study, the final masses have been computed using a friends-of-friends analysis and a calculation of binding energy. However, this is an approximation compared to running the simulation out many days longer in time to get the final bound objects, which is increasingly effected by inaccuracies in the integrator. The application of CNNs (Krizhevsky et al. 2012) could also improve the reliability of clump detection, allowing for a more accurate identification and classification of second- and third-mass planets or planetesimals emerging from accretion-regime giant impacts. If it is possible to reliably identify bound clumps much earlier in a calculation, then emphasis could be placed on higher numerical resolution rather than longer runtime.

The combination of giant impact studies and ML is new research and we anticipate that many future studies will follow (e.g., Valencia et al. 2019). Machine classification is able to corral the herd of thousands of high-resolution simulations to identify the underlying structure of the parameter space. Machine regression is able to produce a quick and efficient algorithm for accretion efficiency, which can be used in dynamical models, such as *N*-body codes studying the growth of planets. These constitute the prototype of a surrogate model that will reliably map inputs to outcomes and will effectively be equivalent to running an SPH simulation as an intermediate step during *N*-body studies of planet formation. In fact, the surrogate models may become preferred because they run on an expedient functional call, yet are trained on high-resolution simulations instead of low-resolution simulations that would be run on-the-fly.

Because it represents simulation-derived data as a *function*, a surrogate model can be inverted to formally understand the likelihood of specified scenarios of planet formation, such as Theia deriving from nearby the Earth or Mercury forming in a couple of hit-and-run collisions (Chau et al. 2018). Such inversion can be performed by means of Markov Chain Monte Carlo Bayesian inference (Stuart 2010) of observed post-collision scenarios, in which the surrogate models are used to sample the (unknown) posterior distribution of pre-impact conditions. Recent uses of this approach in planetary science include a new technique for constraining the thermal inertias of rock and regolith, and relative rock abundance, on asteroids from observed infrared fluxes (Cambioni et al. 2019). Rather than a boutique of scenarios that can solve for the origin of a given planet, there can be an inversion of outcomes. Lastly, there is an unknown future significance of ML in studies of planet formation, where *unsupervised* classification of these datasets can reveal new and unforeseen trends and relationships in the data, leading to the development of better scientific models. Humans are excellent at looking for patterns in 2D and 3D datasets, but *N*-dimensional trends can often be performed better by a computer, leading to accurate data-driven models and scaling laws that help us to explain why collisions happen the way that they do.

S.C., E.A., A.E. and S.R.S. acknowledge support from NASA Planetary Science Division (PSD)—grant title “Application of Machine Learning to Giant Impact Studies of Planet Formation”—and the University of Arizona. The authors thank the anonymous referees for their precious comments and suggestions, which have improved this manuscript.

ORCID iDs

Saverio Cambioni https://orcid.org/0000-0001-6294-4523
Erik Asphaug https://orcid.org/0000-0003-1002-2038
Alexandre Emsenhuber https://orcid.org/0000-0002-8811-1914
Travis S. J. Gabriel https://orcid.org/0000-0002-9767-4153
Stephen R. Schwartz https://orcid.org/0000-0001-5475-9379

References

Agnor, C., & Asphaug, E. 2004, ApJL, 613, L157
Agnor, C. B., Canup, R. M., & Levison, H. F. 1999, Icar, 142, 219
Ahmad, A., & Dey, L. 2007, Data & Knowledge Engineering, 63, 503
Asphaug, E. 2010, ChEG, 70, 199
Asphaug, E., Agnor, C. B., & Williams, Q. 2006, Natur, 439, 155
Asphaug, E., Collins, G., & Jutzi, M. 2015, in Asteroids IV, ed. P. Michel, F. E. DeMeo, & W. F. Bottke (Tucson, AZ: Univ. Arizona Press), 661
Asphaug, E., & Reufer, A. 2013, Icar, 223, 544
Asphaug, E., & Reufer, A. 2014, NatGe, 7, 564
Ballouz, R.-L., Richardson, D. C., Michel, P., & Schwartz, S. R. 2014, ApJ, 789, 158
Ballouz, R.-L., Richardson, D. C., Michel, P., Schwartz, S. R., & Yu, Y. 2015, P&SS, 107, 29
Barnes, J., & Hut, P. 1986, Natur, 324, 446
Baruque, B., & Corchado, E. 2010, Fusion Methods for Unsupervised Learning Ensembles (Berlin: Springer)
Benz, W., Anic, A., Horner, J., & Whitby, J. A. 2007, SSRv, 132, 189
Benz, W., & Asphaug, E. 1999, Icar, 142, 5
Benz, W., Cameron, A. G. W., & Melosh, H. J. 1989, Icar, 81, 113
Bishop, C. M. 1995, Neural Networks for Pattern Recognition (Oxford: Oxford Univ. Press)
Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Berlin: Springer)
Boser, B. E., Guyon, I. M., & Vapnik, V. N. 1992, in Proc. Fifth Annual Workshop on Computational Learning Theory (New York: ACM), 144
Breiman, L. 1996, Machine Learning, 45, 261
Breiman, L. 2001, Machine Learning, 45, 5
Brodley, C. E., & Friedl, M. A. 1999, Journal of Artificial Intelligence Research, 11, 131
Cambioni, S., Delbo, M., Ryan, A. J., Furfaro, R., & Asphaug, E. 2019, Icar, 325, 16
Canup, R. M. 2005, Sci, 307, 546
Canup, R. M. 2011, AJ, 141, 35
Canup, R. M., & Asphaug, E. 2001, Natur, 412, 708
Canup, R. M., Barr, A. C., & Crawford, D. A. 2013, Icar, 222, 200
Chambers, J. E. 2013, Icar, 224, 43
Chau, A., Reinhardt, C., Helled, R., & Stadel, J. 2018, ApJ, 865, 35
Davis, D. R., & Ryan, E. V. 1990, Icar, 83, 156
Demuth, H. B., Beale, M. H., De Jess, O., & Hagan, M. T. 2014, Neural Network Design, 2, 9
Duda, R. O., Hart, P. E., & Stork, D. G. 2012, Pattern Classification (New York: Wiley)
Elkins-Tanton, L. T., & Weiss, B. P. 2017, Planetesimals: Early Differentiation and Consequences for Planets (Cambridge: Cambridge Univ. Press)
Emsenhuber, A., Jutzi, M., & Benz, W. 2018, Icar, 301, 247
Gabriel, T. S. J., Jackson, A., Asphaug, E., Jutzi, M., & Benz, W. 2019, ApJ, in press
Genda, H., Kobayashi, H., & Kokubo, E. 2015, ApJ, 810, 136
Haghighipour, N., Maindl, T., & Schaefer, C. 2017, AAS/DPS Meeting, 49, 508.02
Hartmann, W. K., & Davis, D. R. 1975, Icar, 24, 504
Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. 1998, IEEE Intelligent Systems and their Applications, 13, 18
Hinton, G. E., Deng, L., Yu, D., et al. 2012, ISPM, 29, 1

Hyodo, R., Genda, H., Charnoz, S., & Rosenblatt, P. 2017, ApJ, 845, 125
Jackson, A. P., Gabriel, T. S. J., & Asphaug, E. I. 2018, MNRAS, 474, 2924
Jutzi, M. 2015, P&SS, 107, 3
Jutzi, M., & Benz, W. 2017, A&A, 597, A62
Kegerreis, J. A., Eke, V. R., Massey, R. J., et al. 2018a, LPSC, 49, 1886
Kegerreis, J. A., Teodoro, L. F. A., Eke, V. R., et al. 2018b, ApJ, 861, 52
Kokubo, E., & Genda, H. 2010, ApJL, 714, L21
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, CACM, 60, 84
Leinhardt, Z. M., Marcus, R. A., & Stewart, S. T. 2010, ApJ, 714, 1789
Leinhardt, Z. M., & Stewart, S. T. 2012, ApJ, 745, 79
Liu, S.-F., Hori, Y., Lin, D. N. C., & Asphaug, E. 2015, ApJ, 812, 164
Marcus, R. A., Sasselov, D., Hernquist, L., & Stewart, S. T. 2010a, ApJL, 712, L73
Marcus, R. A., Sasselov, D., Stewart, S. T., & Hernquist, L. 2010b, ApJL, 719, L45
Marcus, R. A., Stewart, S. T., Sasselov, D., & Hernquist, L. 2009, ApJL, 700, L118
Melosh, H. J. 2007, M&PS, 42, 2079
Michel, P., Benz, W., & Richardson, D. C. 2004, P&SS, 52, 1109
Michel, P., Benz, W., Tanga, P., & Richardson, D. C. 2001, Sci, 294, 1696
Monaghan, J. J. 1992, ARA&A, 30, 543
Monaghan, J. J., & Lattanzio, J. C. 1985, A&A, 149, 135
O'Brien, D. P., Greenberg, R., & Richardson, J. E. 2006, Icar, 183, 79
Raymond, S. N., O'Brien, D. P., Morbidelli, A., & Kaib, N. A. 2009, Icar, 203, 644
Reufer, A. 2011, PhD thesis, Univ. Bern
Reufer, A., Meier, M. M. M., Benz, W., & Wieler, R. 2012, Icar, 221, 296
Rodrigues, F., Pereira, F., & Ribeiro, B. 2013, PaReL, 34, 1428
Safavian, S. R., & Landgrebe, D. 1991, ITSMC, 21, 660
Scheeres, D. J., Hartzell, C. M., Sánchez, P., & Swift, M. 2010, Icar, 210, 968
Schmidhuber, J. 2015, NN, 61, 85
Schwartz, S. R., Michel, P., Jutzi, M., et al. 2018, NatAs, 2, 379
Schwartz, S. R., Richardson, D. C., & Michel, P. 2012, Granul. Matter, 14, 363
Schwartz, S. R., Yu, Y., Michel, P., & Jutzi, M. 2016, AdSpR, 57, 1832
Shashua, A. 2009, arXiv:0904.3664
Shoemaker, E. M. 1962, Physics and Astronomy of the Moon (New York: Academic)
Socher, R., Bengio, Y., & Manning, C. D. 2012, in Tutorial Abstracts of ACL 2012, Association for Computational Linguistics, 5
Stevenson, D. J. 1987, AREPS, 15, 271
Stewart, S. T., & Leinhardt, Z. M. 2009, ApJL, 691, L133
Stewart, S. T., & Leinhardt, Z. M. 2012, ApJ, 751, 32
Stuart, A. M. 2010, AcNum, 19, 451
Sugiura, K., Kobayashi, H., & Inutsuka, S. 2018, A&A, 620, A167
Thompson, S. L., & Lauson, H. S. 1972, Improvements in the CHART-D Radiation-hydrodynamic code III: Revised analytic equations of state, Tech. Rep. SC-RR-71 0714 (Albuquerque, NM: Sandia National Laboratories)
Ting, K. M. 2010, Encyclopedia of Machine Learning (Boston, MA: Springer)
Valencia, D., Paracha, E., & Jackson, A. P. 2019, arXiv:1902.04052
Wetherill, G. W. 1985, Sci, 228, 877
Zhang, C., & Ma, Y. 2012, Ensemble Machine Learning: Methods and Applications (Berlin: Springer)