ORIGINAL ARTICLE

The Physics-Informed Neural Network Gravity Model Generation III

John Martin¹ [ID] **· Hanspeter Schaub²**

Abstract

Scientific machine learning and the advent of the Physics-Informed Neural Network (PINN) have shown high potential in their ability to solve complex differential equations. One example is the use of PINNs to solve the gravity field modeling problem—learning convenient representations of the gravitational potential from position and acceleration data. These PINN gravity models, or PINN-GMs, have demonstrated advantages in model compactness, robustness to noise, and sample efficiency when compared to popular alternatives; however, further investigation has revealed various failure modes for these and other machine learning gravity models which this manuscript aims to address. Specifically, this paper introduces the third generation Physics-Informed Neural Network Gravity Model (PINN-GM-III) which includes design changes that solve the problems of feature divergence, bias towards low-altitude samples, numerical instability, and extrapolation error. Six evaluation metrics are proposed to expose these past pitfalls and illustrate the PINN-GM-III's robustness to them. This study concludes by evaluating the PINN-GM-III modeling accuracy on a heterogeneous density asteroid, and comparing its performance to other analytic and machine learning gravity models.

Keywords Scientific machine learning · Physics informed neural networks · Astrodynamics · Gravity modeling

---

✉ John Martin
jrmartin@umd.edu

Hanspeter Schaub
hanspeter.schaub@colorado.edu

1   Department of Aerospace Engineering, University of Maryland, 4298 Campus Dr, College Park, MD 20742, USA

2   Ann and H. J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 431 UCB, Boulder, CO 80309, USA
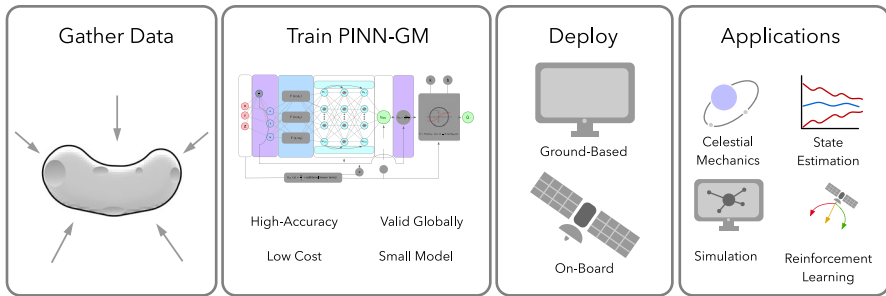
🍩 Springer

Fig. 1 PINN-GMs offer high-accuracy, low-cost gravity solutions to be deployed across a variety of applications

1 Introduction

Nearly all problems in astrodynamics involve the force of gravity. Be it in trajectory optimization, spacecraft rendezvous, orbit determination, or other problems, gravity often plays a significant—if not dominant —role in the system dynamics. The ubiquity of this force is a testament to its significance, yet despite this, there exist surprisingly few ways to represent this force to high accuracy. The construction of high-fidelity gravity models is henceforth referred to as the gravity modeling problem, and currently there exists no universally adopted solution. Some gravity models perform well for large planetary bodies, but break down when modeling objects with exotic geometries like asteroids and comets. Other models can handle irregular shapes but require assumptions and come with high computational cost. The one commonality is that all existing gravity models come with their own unique pitfalls that prevent standardization across the community. Consequently, researchers are continuing to design new solutions to the gravity modeling problem with hopes of one day finding a universal model. One encouraging vein of research explores the use of neural networks and scientific machine learning to learn gravity models free of these past difficulties (Fig. 1).

Scientific machine learning uses neural networks to generate high-fidelity models of, and solutions to, complex differential equations [1, 2]. Physics-informed neural networks (PINNs) are one class of model available to solve such problems. PINNs are neural networks trained in a manner that intrinsically respects relevant differential and physics-based constraints. This compliance, or "physics-informing", is achieved by augmenting the network cost function with said constraints, such that the learned model is penalized for violating any of the underlying physics. Through this design change, PINNs have been shown to achieve higher accuracy with less data than their traditional neural network counterparts [3].

Recently, researchers have proposed the use of PINNs as well as other machine learning models to solve the gravity modeling problem [4, 5]. These models have been shown to produce high-accuracy solutions in both large- and small-body settings under specific training conditions. While these models demonstrate early promise, closer inspection reveals that there remain regimes in which these machine

learning models perform unreliably. This paper exposes these shortcomings and proposes solutions to them through multiple design changes to the underlying machine learning architecture. Specifically, this paper introduces the third generation of the PINN gravity model, or PINN-GM, which includes design changes that solve the problems of feature divergence, extrapolation error, numerical instability, bias towards low-altitude samples, and incompliant boundary conditions.

2 Background

Gravity field models can be categorized into two groups: analytic and numerical. Analytic models are derived from first principles and yield closed-form equations for the gravitational potential, whereas numerical models are constructed in a data-driven manner that come without an explicit equation. Each model has its own set of advantages and drawbacks, and the choice of which model to use is often dictated by application. The following section briefly surveys the available gravity models and their corresponding pros and cons.

2.1 Analytic Models

2.1.1 Spherical Harmonics Model

In the 1900s, spherical harmonic basis functions were proposed to represent high-order perturbations in the Earth's gravity field [6]. These harmonics can be superimposed to produce the spherical harmonic gravity model through:

$$U(r) = \frac{\mu}{r} \sum_{l=0}^{l} \sum_{m=0}^{l} \left(\frac{R}{r}\right)^{l} P_{l,m}(\sin\phi)\left[C_{l,m}\cos(m\lambda) + S_{l,m}\sin(m\lambda)\right] \tag{1}$$

where r is the distance to the test point, μ is the gravitational parameter of the body, R is the circumscribing radius, l and m are the degree and order of the model, $C_{l,m}$ and $S_{l,m}$ are the Stokes coefficients, λ and ϕ are the longitude and latitude, and $P_{l,m}$ are the associated Legendre polynomials [7]. The spherical harmonic gravity model is commonly used to represent the fields of large planetary bodies like the Earth [8], the Moon [9], and Mars [10], as they are among the most efficient models for capturing planetary oblateness—the largest gravitational perturbation found on these large, rotating bodies. Using only a single coefficient, $C_{2,0}$, spherical harmonics can succinctly capture this important gravitational feature and its effects on spacecraft motion.

While these models are effective at representing planetary oblateness, they struggle to model the remaining gravitational perturbations like mountain ranges, tectonic plate boundaries, and craters. These discontinuous features are notoriously hard to represent using periodic basis functions—requiring the superposition of hundreds of thousands of high-frequency harmonics to overcome the 3D equivalent of Gibbs phenomenon [11]. These higher frequencies introduce an $\mathcal{O}(n^2)$ computational and

memory cost [12], and the regression of these harmonics is especially difficult due to their rapidly fading observability. High-order spherical harmonic models also can diverge when evaluated within the sphere that bounds all mass elements (Brillouin sphere). While these effects are negligible for near-spherical planets or moons, they can become problematic in small-body settings where objects can exhibit highly non-spherical geometries [13].

2.1.2 Polyhedral Gravity Model

The polyhedral gravity model is a popular alternative in these small-body settings, offering a solution that maintains validity down to the surface of any object regardless of shape. These models require a preexisting shape model of the asteroid or comet — a collection of triangular facets and vertices which captures its geometry—from which an analytic acceleration can be computed through:

$$\nabla U = -G\sigma \sum_{e \in \text{edges}} \mathbf{E}_e \cdot \mathbf{r}_e \cdot L_e + G\sigma \sum_{f \in \text{facets}} \mathbf{F}_f \cdot \mathbf{r}_f \cdot \omega_f \tag{2}$$

where G is the gravitational constant, σ is the density of the body, $\mathbf{E}_e$ is an edge dyad, $\mathbf{r}_e$ is the position vector between the center of the edge and the test point, L_e is an analog to the potential contribution of the edge, $\mathbf{F}_f$ is the face normal dyad, $\mathbf{r}_f$ is the distance between the face normal and the test point, and ω_f is an analog to the potential contribution of the face [13].

While the polyhedral gravity model avoids divergence within the Brillouin sphere, it comes with its own challenges. First, this gravity model is expensive when evaluated on high-resolution shape models—requiring intensive summation loops over all vertices and facets to compute the acceleration at a single test point. The model also assumes that the body's density is known a priori. While a constant density assumption is often used, literature has shown that such assumption is not necessarily valid [14, 15].

2.1.3 Mascon Gravity Model

In contrast to the analytically derived spherical harmonic and polyhedral models, the mascon gravity models instead approximates the gravitational potential using a collection of point mass elements, known as mascons [16]. These masses are distributed within the body and summed to form a global approximation of the gravity field through Equation 3:

$$\boldsymbol{a}(\boldsymbol{r}) = \sum_{k=0}^{N} \mu_k \frac{\boldsymbol{r} - \boldsymbol{r}_k}{\|\boldsymbol{r} - \boldsymbol{r}_k\|^3} \tag{3}$$

where μ_k and $\boldsymbol{r}_k$ are the estimated gravitational parameter and position for the k-th mascon.

While the mascon model offers an efficient alternative to the polyhedral model, it becomes inaccurate when evaluated near individual mascons [17]. Hybrid

mascon models offer a more accurate alternative — representing each mascon with a low fidelity spherical harmonic model — but this incorporates additional complexity in their regression and remains prone to the same challenges of the traditional mascon and spherical harmonic models [18].

2.1.4 Other Analytic Models

The ellipsoidal harmonic model follows a similar approach to that of spherical harmonics but uses ellipsoidal harmonic basis functions instead [19]. This yields a smaller region in which the model could diverge; however, it still struggles to model discontinuity with its periodic basis functions. The interior spherical harmonic model inverts the spherical harmonic formulation and can model a local region whose boundary intersects only one point on the surface of the body [20]. This model maintains stability down to that single point making it valuable for precise landing operations; however, the solution becomes invalid on any other point on the surface and outside of the corresponding local sphere. Finally, the interior spherical Bessel gravity model expands upon the interior model but uses Bessel functions rather than spherical harmonics to achieve a wider region of validity. This model comes with added analytic complexity and can also struggle to capture discontinuous features efficiently [21].

2.2 Machine Learning Models

As an alternative to analytic gravity models, recent efforts explore the use of machine learning to regress models of complex gravity fields in a data-driven manner. Unlike analytic approaches, machine learning models are generally free of assumptions about the body they are modeling and have no analytic limitations — e.g. divergence in the bounding sphere, required shape models, etc. These models have historically required large volumes of training data; however once trained, they can offer high-accuracy predictions at comparatively low computational cost. Table 1 summarizes key accuracy and training metrics for recently reported machine learning gravity models, and a specific discussion for each model is provided below.

Table 1 Machine Learning Gravity Model Statistics – See Appendix C

Model	Parameters	Training Data	Avg. Error [%]	Valid Globally
GP [22]	12,960,000	3600	1.5	✗
NNs [23]	1,300,000	800,000	0.35	✗
ELMs [24]	350,000	768,000	1–10	✗
GeodesyNet [25]	80,800	1,000,000	0.36	✓
PINN-GM-III	2211	4096	0.30	✓

2.2.1 Gaussian Processes

Gaussian processes, or GPs, are non-parametric models that are fit by specifying a prior distribution over functions, and updating that prior based on observed data. This requires the user to first specify some kernel function which measures similarity between data and then compute a covariance matrix between all data pairs using that function. Once computed, the covariance matrix is inverted and used to evaluate the mean and uncertainty of the learned function at a test point.

GPs were proposed to solve the gravity modeling problem in 2019 [22], regressing a mapping between position and acceleration training data. GPs are advantageous because they provide a probabilistic estimate of the uncertainty in the model's prediction; however these models do not scale well to large data sets. The GP's covariance matrix is built from the training data and scales as $\mathcal{O}(n^2)$, where n is the size of the training data set, and the computational complexity of the matrix inversion scales as $\mathcal{O}(n^3)$. This scaling makes it impractical to fit GPs using large quantities of data, intrinsically limiting its utility and performance.

2.2.2 Extreme Learning Machines

Extreme learning machines (ELMs) have also been proposed to predict gravitational accelerations [24]. ELMs are single layer neural networks fit by randomly initializing the weights from the inputs to the hidden layer, and then computing the optimal weights to the output layer using least-squares regression. Explicitly the ELM minimizes the mean-square error loss function:

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=0}^{N} \|\hat{\boldsymbol{y}}_i(\boldsymbol{x}_i|\boldsymbol{\theta}) - \boldsymbol{y}_i\|^2 \tag{4}$$

where $\hat{y}_i(\boldsymbol{x}_i|\boldsymbol{\theta})$ is the machine learning model prediction at input $\boldsymbol{x}_i$ with trainable model parameters $\boldsymbol{\theta}$ [26].

ELMs are advantageous because they can model non-linear functions, and they only require a single training iteration. However, depending on the width of the ELM and the amount of training data used, these models can be prone to memory and computational difficulties due to the large $\mathcal{O}(n^3)$ matrix inversion used in the least squares solution. Iterative chunking strategies have been proposed to remedy this issue, though this makes the relative advantage between ELMs and traditional neural networks less apparent. Moreover, these models are data intensive, such that past solutions have required hundreds of thousands of training data to produce high-accuracy models.

2.2.3 Neural Networks

Neural networks are similar to ELMs, but they are typically multi-layer and fit using gradient decent and backpropagation rather than least squares. These models iteratively update their parameters using small batches of training data

combined with various optimization algorithms [27–29]. Neural networks have also been proposed as a candidate gravity model [30]. When juxtaposed with ELMs, neural networks' primary benefit is their lack of required matrix inverse; however, this comes at the cost of iterative training which can take long periods of time. Neural networks share similar drawbacks to ELMs, requiring large quantities of data and are prone to extrapolation error when evaluated outside of the bounds of the training data.

2.2.4 Physics-Informed Neural Networks

Physics-informed neural networks, or PINNs, are yet another model proposed to solve the gravity modeling problem [4, 5]. These physics-informed models increase sample efficiency over traditional networks by incorporating differential constraints into the loss function of the neural network. These constraints limit the set of learnable functions to only those that comply with the underlying physics [3]. The first PINN gravity model, or PINN-GM, uses the known differential equation $-\nabla U = \boldsymbol{a}$ to define the loss function:

$$L(\theta) = \frac{1}{N} \sum_{i=0}^{N} \| - \nabla \hat{U}(x_i|\theta) - \boldsymbol{a}_i\|^2 \tag{5}$$

where $\hat{U}$ is the potential learned by the network, which can then be differentiated via automatic differentiation [31] to compute the corresponding acceleration. The inclusion of these physics constraints, combined with network design modifications, have shown that PINN-GMs can yield solutions that maintain similar accuracy to their predecessors while using orders of magnitude fewer parameters and data. These models have also demonstrated enhanced robustness to uncertainty in the training data as a result of their physics-informed constraints [5].

2.2.5 GeodesyNet

Finally, in 2022 Ref. [25] introduced GeodesyNets as a candidate solution to the gravity modeling problem. GeodesyNets are neural density fields—close relatives to the popular neural radiance fields [32]—which use neural networks to learn a density function for every point in a 3D volume. Once learned, these density fields can be numerically integrated to yield a gravitational acceleration or a value of the potential. This work is an exciting new class of machine learning model capable of achieving high accuracy given sufficient quantities of data. These models also behave more reliably at high-altitudes as the networks are not required to predict densities beyond a unit volume; albeit the cost required to perform the numerical integration can be relatively high—requiring evaluation of 30,000 to 300,000 quadrature points per prediction [25]—which can make these models cumbersome to evaluate and train. Finally, a key feature of these models is their ability to estimate an internal density profile of an asteroid—a capability presumed available to PINNs by evaluating Poisson's equation inside the body—but such capabilities are not the focus of this work.

2.3 Current Challenges

As these machine learning gravity models become more mainstream, they warrant further scrutiny. While these models can yield accurate solutions under ideal training and test cases, further investigation reveal that these models are not yet universally robust. Rather, there exist common problems that exist for the majority of these models that have yet to be identified or addressed. Specifically, these models are prone to extrapolation error and numerical instabilities, and they are also relatively brittle to sparse and noisy data conditions. If the community strives to develop a universally valid machine learning gravity model, these challenges need to be addressed.

In this manuscript, we aim to expose these various failure cases and introduce design modifications to the underlying machine learning architectures that can increase the their robustness and generalizability. To accomplish this, we introduce the third generation PINN gravity model, or PINN-GM-III, which includes a variety of design changes that improve modeling accuracy and robustness across a wide set of training and test cases. These modifications, and the failure modes they eliminate, are discussed at length in Secti. 3. While these modifications are applied specifically to the PINN-GM, it should be noted that some can be applied to other machine learning gravity models as well. Details of the specific PINN modifications and their impact on performance are supplied below.

3 PINN-GM-III

The training process for the original PINN-GM can be found in Ref. [5] and is briefly summarized here for convenience. To begin, the neural network requires a set of N position and acceleration vectors sampled around a celestial body:

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \bar{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{6}$$

This data can be collected in one of two ways: 1) It can be generated synthetically from pre-existing high-fidelity gravity models like EGM-2008 for Earth [33] or a high-fidelity polyhedral model for asteroids [34], or 2) the data can be estimated on-board without any prior knowledge of the field. The latter can be accomplished using estimation techniques like dynamic model compensation [35] to simultaneously estimate spacecraft position and accelerations in-situ as is shown in Ref. [36]. For this study, all data will be synthetically generated and evaluation on real data is left for future work.

Once the data is gathered, a neural network is fit to learn a mapping between these data. The neural network takes the form:

$$\hat{y}_\theta(x) = W^{(L)} \cdot \sigma(W^{(L-1)} \cdot \ldots \cdot \sigma(W^{(1)} \cdot x + b^{(1)}) + b^{(L-1)}) + b^{(L)} \tag{7}$$
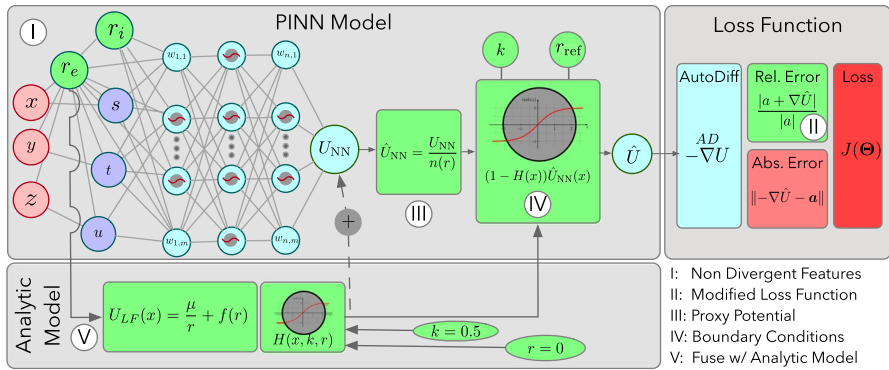
Fig. 2 PINN-GM Generation III with new modifications highlighted in green

where $\hat{y}_{\theta}$ is the output of the neural network parameterized by weights and biases $W, b \in \theta$. These parameters are found in the 0th to L-th hidden layers of the network. Successive non-linear transformations of the network input x are applied via the activation function, σ, to yield $\hat{y}_{\theta}(x)$.

For PINN-GMs, the output of the neural network corresponds to the predicted gravitational potential, $\hat{U}$, at position x. That potential is then differentiated to produce an acceleration vector which is used in the network's physics-informed loss function:

$$L(\theta) = \frac{1}{N} \sum_{i=0}^{N} \| - \nabla \hat{U}(x_i|\theta) - a_i \|^2 \tag{8}$$

The network is iteratively trained using stochastic gradient descent which estimates the gradient of the loss with respect the network parameters, $\nabla_{\theta} L$. For every iteration, the parameters are updated via:

$$\theta' \rightarrow \theta - \eta \nabla_{\theta} L \tag{9}$$

where η is a user-defined learning rate.

The second generation PINN-GM, or PINN-GM-II, enhances this training process by augmenting the loss function with additional constraints to increase robustness to noise. In addition, better input features are selected to increase sample efficiency. Despite these improvements, closer inspection has revealed that the PINN-GM-II has failure cases that remain unaddressed. While these pitfalls are explicitly exposed within the PINN-GM-II, they are general to other published machine learning gravity models to the best of the authors' knowledge. The subsequent sections outline these pitfalls and propose architectural changes to the underlying model architecture to address them. Collectively, these changes form the third generation PINN gravity model, or PINN-GM-III. The modifications are illustrated in Fig. 2 and discussed in detail in Sect. 3.1 through 3.5, with their combined effect demonstrated in Sect. 3.6.

Algorithm 1 PINN-GM-III algorithm

1: Collect training data $(\boldsymbol{x}, \boldsymbol{a})$ from:
   (a) a pre-existing model
   (b) online state estimates [36]
2: Non-dimensionalize the training data   ▷ App. A
3: Convert to 5D non-singular spherical coordinates (r_i, r_e, s, t, u) ▷ Sec. 3.1
4: Propagate through the neural network
5: Output proxy potential U_{NN} ▷ Sec. 3.3
6: Scale proxy potential into true potential $\hat{U}_{\mathrm{NN}}$ ▷ Sec. 3.3
7: *(Optional)* Fuse with weighted low-fidelity potential $\hat{U}_{\mathrm{LF}}$ ▷ Sec. 3.5
8: Enforce boundary conditions on the network potential $\hat{U}$ ▷ Sec. 3.4
9: Autodifferentiate (AD) potential to produce acceleration $\hat{\boldsymbol{a}}$
10: **if** Training **then**
11:      Sum acceleration percent error and absolute error to form loss   ▷ Sec. 3.2
12:      Compute gradients of loss function
13:      Update network parameters
14: **end if**

3.1 Feature Engineering

The first pitfall of past machine learning gravity models is the choice of inefficient and divergent input features. Cartesian position coordinates are most regularly used as inputs for these models; however, these coordinates are prone to two major drawbacks. First, they span the set of all real numbers, and the networks require data sampled across this entire domain to learn robust solutions for any possible value of $\boldsymbol{x}$. The PINN-GM-II first identified this problem and proposed a conversion to 4D non-singular spherical coordinates (r, s, t, u) where s, t, and u are the sine of the angle between the test point and the cartesian axes. While the domain for s, t, and u was reduced to $[-1, 1]$ which increases the model's sample efficiency, the radial coordinate, r, remains prone to the second major drawback: feature divergence.

When evaluating these numerical models at points far from the body, the radial coordinate remains unbounded and can introduce numerical instabilities that cause the model to diverge. While the radial coordinate can be converted to $1/r$, this instead introduces instability for test points near the surface of the body. In both cases, the radial feature can have magnitudes greater than one which will cause many activation functions to prematurely saturate and decrease the learning efficiency of the model [37].

This pitfall can be addressed through a simple design modification where the Cartesian position coordinates are converted into a 5D spherical coordinate description of (r_i, r_e, s, t, u) where r_e and r_i are two proxies of the test point radius, r, defined as

$$r_i = \begin{cases} r & r \in [0, R] \\ 1 & r \in [R, \infty) \end{cases} \quad \text{and} \quad r_e = \begin{cases} 1 & r \in [0, R] \\ \frac{1}{r} & r \in [R, \infty) \end{cases} \tag{10}$$

Using this convention, the network will maintain the desired sample efficiency of past approaches while also ensuring that its input features can never diverge regardless of the location of any training or test point. This modification constitutes the first design change of the PINN-GM-III.

3.2 Modified Loss Function to Account for High-Altitude Samples

The second pitfall of past machine learning gravity models arises from their default loss functions which inadvertently decrease modeling accuracy at high-altitudes. Specifically, most machine learning gravity models use an absolute error loss function—e.g. mean squared error (MSE), root mean squared error (RMS). For the original PINNs, this was captured through the norm of the differenced acceleration vectors:

$$\mathcal{L}_{\text{ABS}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=0}^{N} \left\| -\nabla \hat{U}(\boldsymbol{x}_i|\boldsymbol{\theta}) - \boldsymbol{a}_i \right\| \tag{11}$$

While these loss functions will successfully minimize the most flagrant residuals between the predicted and true acceleration vectors; they induce an undesirable effect at high-altitudes. Because accelerations produced near the surface of a celestial body have much larger magnitudes than the accelerations produced at high-altitudes, low-altitude errors will always appear disproportionately large compared to any errors at high-altitude. As a consequence, loss functions that minimize absolute errors will always prioritize low-altitudes samples, even if the high-altitude predictions are more erroneous in a relative sense.

Fortunately, this design flaw can be trivially remedied by augmenting the loss function with an term that captures relative error, like mean percent error:

$$\mathcal{L}_{\text{ABS}+\%}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=0}^{N} \left(\left\| -\nabla \hat{U}(\boldsymbol{x}_i|\boldsymbol{\theta}) - \boldsymbol{a}_i \right\| + \frac{\left\| -\nabla \hat{U}(\boldsymbol{x}_i|\boldsymbol{\theta}) - \boldsymbol{a}_i \right\|}{\|\boldsymbol{a}_i\|} \right) \tag{12}$$

The joining of these relative and absolute error loss terms eliminates this altitude sensitivity and constitutes the second design change of the PINN-GM-III.

3.3 Improve Numerics by Learning a Proxy to the Potential

Another common problem of past machine learning gravity models is numerical clipping. These models use the same matrix operations (i.e. the neural network) to predict both very large and very small values of the potential depending on the test point's altitude, which can pose difficult numerics. When the altitude exceeds some critical threshold, the predictions grow too small, and the network prematurely clips their value to zero. This is problematic for both model training and inference—prematurely capping the maximum altitude for which these models are usable.

The PINN-GM-III addresses this problem by instead learning a more numerically favorable proxy to the potential, U_{NN}, defined as:

$$U_{\mathrm{NN}} = U * n(r); \quad n(r) = \begin{cases} 1 & r < R \\ r & r > R \end{cases} \tag{13}$$

where U is the true potential and $n(r)$ is an altitude-dependent scaling function. This scaling function leverages the fact that potentials decay according to an inverse power-law outside the Brillouin sphere. By learning a potential normalized to $1/r$, the neural network predictions will not decay to unrecoverably small values but will instead remain bounded and centered about a non-dimensionalized value of μ. This scaling eliminates numerical clipping to first order, granting stable numerics out to infinity. Importantly, once the network produces its numerically stable value for U_{NN}, the model will then explicitly transform that value into the true potential $\hat{U}_{\mathrm{NN}}$ by dividing the output by the known scaling function through:

$$\hat{U}_{\mathrm{NN}} = \frac{U_{\mathrm{NN}}}{n(r)} \tag{14}$$

3.4 Enforcing Boundary Conditions via Transition Function to Avoid Extrapolation Error

The largest identified weakness of past machine learning gravity models is their extrapolation error. When tested beyond the bounds of their training data, machine learning models remain unconstrained and often diverge. The PINN-GM-III solves this problem by introducing a transition function that forces the model to seamlessly blend its network solution into a known analytic boundary condition outside the bounds of the training data through:

$$\hat{U}(r) = w_{\mathrm{NN}}(r)\hat{U}_{\mathrm{NN}}(r) + w_{\mathrm{BC}}(r)U_{\mathrm{BC}}(r) \tag{15}$$

where $\hat{U}_{\mathrm{NN}}$ is the predicted gravitational potential, U_{BC} is the potential at the known boundary condition, and w_{NN} and w_{BC} are the altitude dependent weights for each defined as:

$$w_{\mathrm{BC}}(r, k, r_{\mathrm{ref}}) = H(r, k, r_{\mathrm{ref}}) \tag{16}$$

$$w_{\mathrm{NN}}(r, k, r_{\mathrm{ref}}) = 1 - H(r, k, r_{\mathrm{ref}}) \tag{17}$$

$H(r)$ is a smoothing function defined as

$$H(r, k, r_{\mathrm{ref}}) = \frac{1 + \tanh(k(r - r_{\mathrm{ref}}))}{2} \tag{18}$$

where r is the distance to the test point, r_{ref} is a reference radius, and k is a smoothing parameter which controls the sharpness of the transition.

Equation (15) takes inspiration from Ref. [38] which introduced the idea that PINNs can enforce physics compliance through other mechanisms than terms in their loss function. Explicitly, Ref. [38] proposed the use of smooth, differentiable

forms of the Heaviside function placed at the boundary to forcibly transition neural network outputs towards the known values.

In the case of gravity modeling, the most obvious boundary condition exists in the limit as $r \to \infty$, where the potential decays to zero. Setting $U_{\text{BC}} = 0$ and $r_{\text{ref}} = \infty$ in Eq. (15), however, is not practical, as it demands that the neural network must learn a model of the potential for the entire domain $r \in [0, \infty)$. A more useful choice is to leverage insights from the spherical harmonic gravity model and recognize that high frequency components of the gravitational potential decay to zero more quickly than the point mass contribution at high altitudes—i.e.

$$U_{\text{BC}}(r) = U_{\text{LF}} = \frac{\mu}{r} + \sum_{l=0}^{n} \sum_{m=0}^{l} \frac{\mu}{r} \left(\frac{R}{r}\right)^{l} \overset{0}{(\dots)} \tag{19}$$

as $r \to \infty$. Therefore, $U_{\text{BC}}(r)$ can be set to $\frac{\mu}{r}$ assuming $r \gg R$. Therefore the PINN-GM-III sets $U_{\text{BC}} = \frac{\mu}{r} + f(r)$ in Eq. (15), where $f(r)$ are any higher order terms in the spherical harmonic gravity model that the user knows a priori and wishes to leave as part of the boundary condition.

For completeness, the recommended value for r_{ref} is the maximum altitude of the training data, and the recommended value for the transition coefficient is $k = 0.5$. These choices ensure that the model will quickly transition to the boundary condition outside the training data, but without applying the transition too rapidly such that it changes the gradient of the potential and accidentally induces acceleration errors.

3.5 Leveraging Preexisting Gravity Information into PINN-GM Solution

The final design change of the PINN-GM-III proposes a mechanism for incorporating past analytic models into the network solution. Explicitly, analytic models remain popular for good reason, as they are often very good at representing certain parts of the gravity field — e.g. a point mass model captures first-order dynamics with a single parameter and and spherical harmonics can represent planetary oblateness with only $C_{2,0}$. Rather than abandoning these models and requiring the network to relearn these prominent behaviors and features, it would be far more convenient to incorporate these models into the learned solutions.

To accomplish this, the PINN-GM-III proposes modifying Equation (15) to include an analytic model term in the prediction:

$$\hat{U}(r) = w_{\text{NN}}\left(U_{\text{LF}}(r) + U_{\text{NN}}(r)\right) + w_{\text{BC}}U_{\text{LF}}(r) \tag{20}$$

where U_{LF} refers to the known, low-fidelity analytic model such as $U_{\text{LF}}(r) = \frac{\mu}{r} + U_{J_2}(r)$. By incorporating these low-order models, the network can exclusively focus its modeling efforts on capturing high-order perturbations from these models.

It should be noted that the inclusion of analytic models should be performed carefully. In the case of small-body gravity modeling, where the geometries may

vastly differ from a point mass or low-fidelity spherical harmonic approximation, the incorporation of past analytic models can induce unnecessary error at low altitudes. To ensure that these analytic models are only used where appropriate, the hyperbolic tangent fusing function, $H(x, r, k)$, is reused to dynamically weight the analytic solution through:

$$\hat{U}_{\mathrm{LF}}(r) = w_{\mathrm{LF}} U_{\mathrm{LF}}(r) \tag{21}$$

where $\hat{U}_{\mathrm{LF}}$ is the weighted low fidelity analytic model with $w_{\mathrm{LF}} = H(r, R^*, k^*)$, $R^* = 0$, and $k = 0.5$. This ensures that at high altitude, where the analytic approximation is most accurate, it fully contributes to the final learned solution, but at lower altitudes, the analytic contribution has a reduced weight to the final solution. A visualization of the various weighting functions used in the PINN-GM-III is shown in Fig. 3, highlighting where the different parts of the model are fully activated and deactivated.

3.6 Visualizing Modifications' Effects on Model Performance

To visualize how the proposed modifications affect model accuracy, six PINN-GM are trained. The first model corresponds with the original PINN-GM-II, and the subsequent models sequentially add the proposed modifications. The PINN-GM are trained with a 5,000 point dataset uniformly distributed between the surface of the asteroid Eros to an altitude of 15R. The corresponding acceleration error of each model is reported within the training distribution in Fig. 4a and outside the training distribution in Fig. 4b.

Figure 4a illustrates that these design modifications consistently improve the modeling accuracy of the PINN within the bounds of the training distribution. The PINN-GM-II had errors approaching 1% at the high 15R altitudes; however this decreases to 0.2%, 0.05%, 0.01%, 0.03%, and 0.001% for each of the respective modifications. Notably the transition to the new loss function (II) does reduce
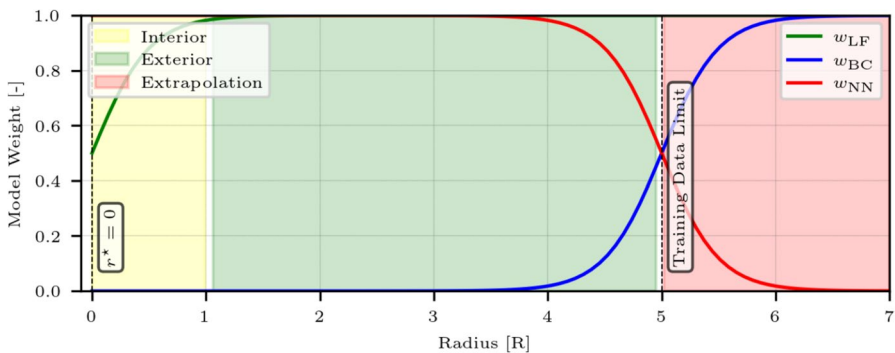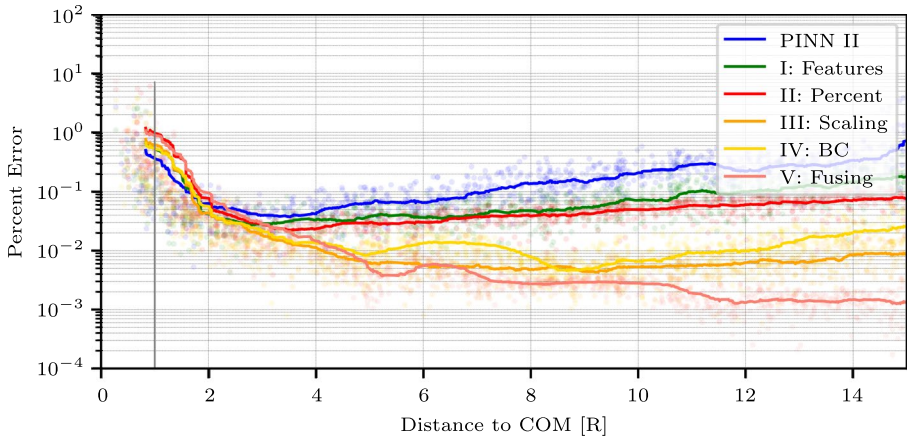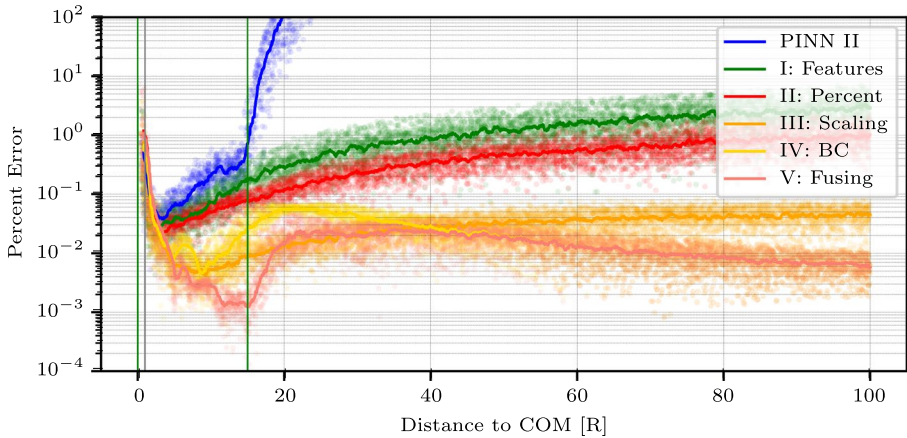

Fig. 3 Visualization of the various weighting factors applied to the neural network potential, boundary condition potential, and the low-fidelity potential

(a) Error **inside** the training bounds

(b) Error **outside** the training bounds

Fig. 4 Acceleration percent error as a function of altitude after sequentially applying each of the proposed PINN III modifications

accuracy near the surface, but this is to be expected given the redistribution of modeling priorities that now balance accuracy at both low- and high-altitudes.

Figure 4b illustrates the stabilizing effect of these modifications on extrapolation error — quantifying prediction error outside of the training bounds out to 100R. As is shown in blue, the PINN-GM-II entirely diverges after leaving the bounds of the training data at 15R; however, the design modifications stabilize performance and prevent this divergence. When only the features (I) and percent loss (II) modifications are included, the error in the high-altitude limit hits a numerical plateau as expected; however, the inclusion of the proxy potential (III) lowers that plateau

by over an order-of-magnitude. The boundary condition modification (IV) eliminates this plateau — albeit inducing a small penalty near the transition point due to changes in the gradient. Finally, the addition of the low-fidelity point mass potential (V) eliminates this penalty and further reduces the error at higher within the bounds of the training data.

4 Benchmarking Suite

The lack of comprehensive and standard performance benchmarks for gravity models are part of the reason these failure cases had not yet been identified. To eliminate this possibility moving forward and to further scrutinize the proposed PINN-GM-III, six new evaluation metrics are proposed. These metrics characterize gravity model accuracy within and beyond the training distribution, aiming to provide both coarse and fine measures of performance in different orbital regimes. These metrics are explained in detail in Sect. 4.1 and are used to characterize the performance of a trained PINN-GM-III on a heterogeneous density asteroid in Sect. 4.2.

4.1 Metrics

4.1.1 Planes Metric

The first accuracy metric assesses the mean percent error of the predicted acceleration vector along the three cartesian planes (XY, XZ, YZ) extended between $[-5R, 5R]$ where R is the radius of the body. The field is evaluated on a 200×200 grid of points along each plane, and the average percent error is computed as

$$P = \frac{1}{N} \sum_{i=1}^{N} \frac{\|a_{\text{true}} - a_{\text{PINN}}\|}{\|a_{\text{true}}\|} \times 100 \tag{22}$$

This metric is intended to provide a coarse measure of model performance across a wide range of operational regimes.

4.1.2 Generalization Metrics

The second, third, and fourth metrics investigate the generalization of the model across a range of altitudes both within and beyond the training bounds. Explicitly, the mean acceleration error is evaluated as a function of altitude and divided into three testing regimes: interior, exterior, and extrapolation. The **interior** metric assesses error within the bounding sphere of radius R. The **exterior** metric investigates the error out to the maximum altitude of the training dataset. Finally, the **extrapolation** metric measures the error exclusively outside the training dataset—specifically reaching altitudes 10 times larger than the maximum altitude represented in the training set. For every unit of radius, 500 samples are distributed uniformly in altitude to produce the test set.

4.1.3 Surface Metric

The fifth metric evaluates the mean acceleration error across all facets on a shape model of a celestial body, if available. This metric is used to characterize model performance at the most complex region of the field.

4.1.4 Trajectory Metric

The sixth metric evaluates the time-averaged position error of a trajectory propagated by the regressed model and the true trajectory of a spacecraft in a 24-hour low-altitude polar orbit about a rotating celestial body. Time-averaged error is used as it ensures that the error is monotonically increasing. To compute this value, the instantaneous position error $\Delta x(t)$ must be computed through

$$\Delta x(t) = \| \underbrace{x(t)}_{\text{True Pos.}} - \underbrace{\hat{x}(t)}_{\text{Propagated Pos.}} \| \tag{23}$$

from which the time-averaged error, $\mathcal{S}$, can be computed using numerical integration via:

$$\mathcal{S} = \frac{1}{T} \int_0^T \Delta x(t) \mathrm{d}t \tag{24}$$

4.2 Experiment

These metrics are used to evaluate a PINN-GM-III trained on data generated from a synthetic, heterogeneous density asteroid modeled after 433-Eros. Heterogeneous asteroids provide an especially challenging scenario for gravity models, as their internal density distributions are not directly observable. Some asteroids contain over- and under-dense regions within their interior, or may have been formed by two asteroids merging together. While some heuristic methods have been proposed to estimate these asteroids' internal densities—which can then be used in heterogeneous forms of the polyhedral model [39]—the more common practice is to simply proceed with a constant density assumption.

   To induce this heterogeneous density body, two small mass inhomogeneities are placed inside the asteroid. In one hemisphere, a mass element is added, and in the other hemisphere, a mass element is removed. Each mass element contains 10% of the total mass of the asteroid, and they are symmetrically displaced along the x-axis by 0.5R (see Fig. 5a). The gravitational contributions of these mass elements are superimposed onto the gravity field of a constant density polyhedral model to form the simulated ground truth. This choice emulates the gravity field of a single body formed by two merged asteroids of different characteristic
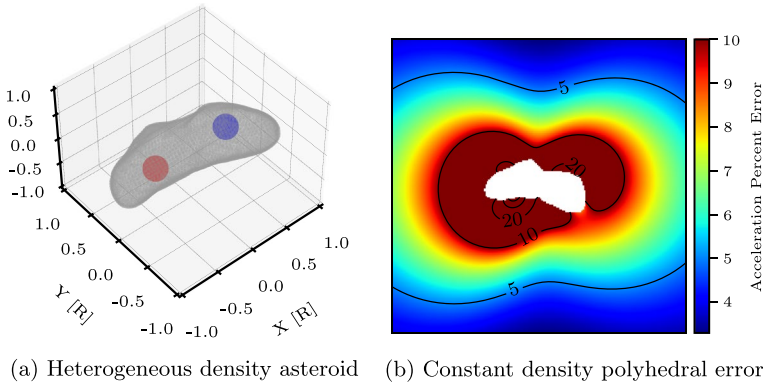
(a) Heterogeneous density asteroid   (b) Constant density polyhedral error

Fig. 5 Heterogeneous density distributions found within asteroids can cause standard models and assumptions to break down

densities. The choice to make each mass element ±10% of the total mass is motivated based on literature with similar candidate density distributions [40].

Using this heterogeneous density model, 90,000 position and acceleration data are sampled uniformly around the body from 0-10R. An additional 200,700 points are sampled on the surface of the asteroid—corresponding to a data point on every facet of the asteroid shape model. Together these approximately 300,000 data points constitute a "best case" training set for the PINN-GM-III used to evaluate the upper-bound performance of the model. Subsequent sections explore model performance under more realistic and stressful data conditions.

Using this data, a PINN-GM-II and a PINN-GM-III are trained. Both networks have six hidden layers with 32 nodes per layer which corresponds to approximately 5,300 learnable parameters. The default hyperparameters used to train these models can be found in Appendix A alongside the studies used to select them in Appendix B. Once trained, these PINN-GMs are evaluated using the aforementioned metrics and compared to a constant-density polyhedral model as shown in Fig. 6.

4.2.1 PINN-GM-III Performance

The generalization metrics are shown at the top of Fig. 6, highlighting model performance across the three altitude regimes. The constant density polyhedral model produces the highest error on the interior and exterior metrics, averaging 10% and 1% respectively. In the extrapolation metric, the polyhedral error decreases as the model begins to behave like a point mass approximation. In comparison, the PINN-GM-II produces lower average errors on the interior and exterior metrics—0.5% and 1%—however the candidate pitfalls described in Sect. 3 also become apparent. The former RMS loss function yields a monotonically increasing error as a function of altitude within the bounds of the training data, and the model diverges outside of it. In contrast, the PINN-GM-III maintains the lowest errors across all three regimes, averaging less than 0.5% and 0.005% error on the interior and exterior metrics, and
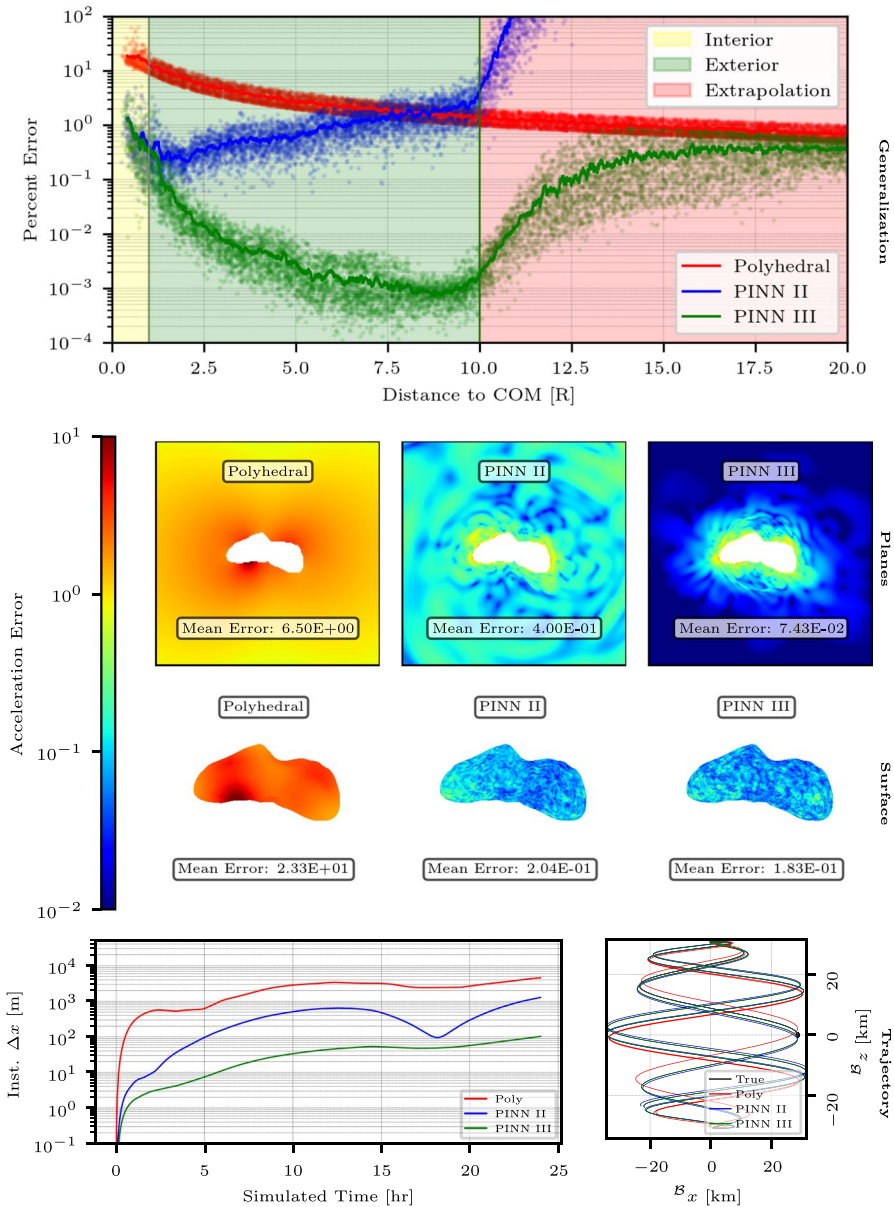
Fig. 6 All proposed metrics evaluated for the constant density polyhedral model, the PINN-GM-II, and the PINN-GM-III

maintains stability in the extrapolation regime due to the hyperbolic tangent function which enforces the point mass boundary conditions.

The planes and surface metrics are shown in the middle rows of Fig. 6. The polyhedral gravity model produced an average acceleration error of 6.5% on the planes,

the PINN-GM-II reduces this to 0.4%, and the PINN-GM-III to 0.07%. The errors are largest near the surface for all models, averaging at 23% for the constant density polyhedral model, 0.2% for the PINN-GM-II, and 0.18% for the PINN-GM-III.

Finally, the trajectory experiment is shown in the bottom row of Fig. 6. For this experiment, the orbit is defined by the initial conditions $\{a, e, i, \omega, \Omega, M\} = \{32 \text{ km}, 0.1, 90°, 0°, 0°, 0°\}$, and the asteroid is rotating at $\omega_0 = 0.00073$ degrees per second along the z-axis. The figure on the left shows the instantaneous position error over the 24 hour integration period which, when time-averaged over the entire trajectory, yields average errors of 2,270 meters of error for the polyhedral model, 363 m for the PINN-GM-II, and 38 m for the PINN-GM-III.

Under these ideal data conditions, the PINN-GM-III achieves better performance on each benchmark when compared to the constant density polyhedral model and the prior PINN-GM-II. While these results are encouraging, the robustness of these models still needs to be tested under more realistic and stressful data conditions. Moreover it remains important that the PINN-GM-III is compared to other popular gravity models beyond just previous generation PINN-GMs. To accomplish this, the next section explores what happens to these models and other models under less ideal data conditions.

5 Comparative Study

A comparative study is performed to evaluate the PINN-GM-III against other popular gravity models. Explicitly, a point mass (PM), spherical harmonic (SH), mascon, polyhedral, extreme learning machine (ELM), traditional neural network (TNN), PINN-GM-I, PINN-GM-II, and geodesyNet model are each regressed and then evaluated using the aforementioned metrics. Each model is fit eight times, permuting the data quantity, data uncertainty, and the size of each model.[1] These conditions purposefully test model robustness under more stressful data conditions and different model sizes.

The dataset used for the comparison is generated by sampling position and acceleration data between 0-10R about the heterogeneous density asteroid discussed in Sect. 4. For the data sparse case, $N = 500$ pairs are sampled. For the data rich case, $N = 50,000$ pairs are sampled. These datasets have two configurations: noisy or noiseless. The noiseless configuration assumes ideal conditions in which the acceleration vectors are perfectly observable and have no error, whereas the noisy configuration perturbs every acceleration vector in a random direction by 10% of the acceleration magnitude via

---

[1] Note: The point mass model only has four permutations, as it does not have a "large" option and the polyhedral model only has two permutations, large and small. This is because polyhedral models are regressed in an entirely unrelated fashion to the other models — relying primarily on image data — and their sensitivity to data quantity and quality can not be adequately compared through this proposed experiment. Therefore, only the effect of model size is considered, and the performance can be considered an upper-bound.

$$\tilde{\boldsymbol{a}}_j = \boldsymbol{a}_j + 0.1\|\boldsymbol{a}_j\|\hat{\boldsymbol{u}}_j$$

where $\hat{\boldsymbol{u}}_j$ is randomly sampled from the unit sphere. This error is chosen as a purposefully exaggerated test case to determine which of these models remain reliable under more stressful mission conditions.

In addition to training each model on these four datasets, this study also explores how the model size impacts performance. To do this, each model is tested at a small (S) and large (L) parametric capacity. These labels correspond to the total number of parameters used by each model—e.g. Stokes coefficients for a spherical harmonic model, facets and vertices in a shape model, or total weights and biases in a neural network. Each small model contain approximately 250 parameters, whereas the large models have approximately 30,000 parameters. Together, these varying data and parametric conditions provide insight into which models are capable of maintaining competitive performance in data sparse and low-memory regimes. Importantly, each model has slightly different regression procedures and model size calculations which are discussed in Appendix D.

5.1 Results

After the models are fit on the datasets, their performance is evaluated using the accuracy metrics from Sect. 4 and their values are reported in Table 2. Along every metric (column), each model is colored by rank with the best model colored purple and the worst model colored red. Cells colored red or black correspond to values that diverged and are automatically assigned last place (70th). In the case of the planes, interior, exterior, extrapolation, and surface metrics, this divergence corresponds with values that exceed 100% error and therefore are not usable. These individual ranks are then summed to produce a final model score which is used to sort the table and quantify relative performance.

Table 2 should be read as follows: The first column corresponds to the specific model that was regressed, and the subsequent two columns define the data conditions used for the regression. The N column is the number of training data and the error column is the magnitude of the acceleration vector error (i.e. either 0% or 10%). When these cells are colored light red, that means the condition is less desirable and is used to stress model performance, whereas white corresponds to the favorable data condition.

The fourth column is the aforementioned model score which sums the metric ranks for the listed model. For example, the small PINN-III (PINN-III S) is ranked 4th on the planes metric, 5th on the interior, 4th on exterior, 3rd on extrapolation, 8th on surface, and 5th on the trajectory metric. Added together, these sum to a model score of 29 which is the second best score among the 70 tested models. The individual ranks for every model in each metric are reported separately in Appendix D Table 6 for reference.

Table 2 shows that the three highest scoring gravity models are the Mascon L, PINN-III S, and PINN-III L models respectively. Each of these models are regressed under the favorable 50,000 and 0% error data conditions, and they all maintain low

Table 2 Gravity models ranked by evaluation metrics

Model	N	Error (%)	Score	Planes (%)	Interior (%)	Exterior (%)	Extrap. (%)	Surface (%)	Traj. (km)
MASCONS L	50000	0	14	1.2E-2	0.13	5.4E-4	3.4E-4	8.3	1.7E-3
PINN III S	50000	0	29	0.17	1.2	2.8E-2	0.065	11	4.2E-2
PINN III L	50000	0	34	1.3E-2	0.099	1.7E-3	1.3	3.2	2.7E-3
PINN II L	50000	0	84	4.7E-2	0.080	0.070	3.2E14	2.5	2.8E-2
PINN III S	500	0	88	2.2	18	0.51	0.13	35	0.89
PINN III L	50000	10	89	3.4	3.7	2.0	0.28	13	6.3
PINN II S	50000	0	97	0.22	0.57	0.31	2.1E7	6.2	0.11
PINN III L	500	0	98	1.6	17	0.23	0.41	52	0.67
MASCONS L	500	0	101	0.82	14	2.5E-4	9.8E-8	350	1.3E-7
POLY S			110	6.2	16	3.3	0.34	23	1.9
MASCONS L	50000	10	113	1.3	9.3	0.16	0.15	610	0.37
PINN II L	500	0	115	0.69	5.1	0.22	3.2E12	25	0.18
MASCONS S	50000	0	117	3.1	4.4	2.8	2.7	26	12
POLY L			119	6.5	16	3.5	0.35	23	2.3
PINN II L	50000	10	122	1.4	2.7	2.2	4.4E14	8.8	0.68
MASCONS S	50000	10	124	3.3	4.6	3.0	3.0	26	13
MASCONS S	500	0	135	2.7	12	1.9	1.8	71	6.7
PINN III L	500	10	140	6.2	23	3.1	0.36	45	6.0
PINN II S	50000	10	140	1.5	2.9	1.8	1.1E8	9.4	6.4
PINN II S	500	0	152	1.8	7.8	1.7	5.4E6	32	3.2
PINN III S	500	10	159	8.2	51	3.3	0.36	76	5.0
PINN III S	50000	10	160	8.1	51	2.3	0.32	79	17
PINN I L	50000	0	174	10	4.3	8.4	1.6E5	8.4	12
PINN II L	500	10	179	5.7	13	4.8	5.6E11	32	5.4
PINN II S	500	10	182	6.4	13	6.5	8.6E6	33	2.7
PINN I L	50000	10	183	13	5.4	16	1.8E5	14	7.0
SH S	50000	0	202	24	370	0.45	0.22	1900	1.1
PINN I S	50000	10	202	14	9.5	31	1.6E5	31	20
PINN I S	50000	0	203	16	8.6	38	1.2E5	33	17
SH S	50000	10	211	24	320	0.53	0.29	2200	1.4
PM -	500	0	211	50	43	50	50	70	110
PM -	500	10	215	50	43	51	51	70	110
SH S	500	0	217	17	340	1.2	0.86	950	0.97
PM -	50000	10	220	53	45	53	53	70	110
PINN I L	500	10	221	29	16	48	46000	34	22
PM -	50000	0	224	53	45	53	53	70	110
SH S	500	10	227	27	580	1.6	0.74	1500	2.4
PINN I S	500	10	235	33	21	69	46000	37	31
SH L	50000	0	239	inf	1.3E99	0.45	0.22	8.8E116	1.1
PINN I S	500	0	242	41	24	75	93000	41	42
GEONET L	500	0	245	83	84	83	83	87	160
SH L	50000	10	247	inf	1.8E99	0.53	0.29	1.2E117	1.4
MASCONS S	500	10	251	34	98	29	28	460	64
GEONET L	50000	0	252	87	87	87	87	87	160
GEONET L	500	10	255	87	90	86	86	93	160
GEONET L	50000	10	260	87	87	87	87	87	160
GEONET S	500	0	273	99	99	99	99	99	180
SH L	500	10	275	inf	2.5E97	5.0	0.35	1.0E115	4.4
SH L	500	0	278	inf	9.8E96	5.0	0.35	2.1E115	4.4
PINN I L	500	0	280	52	41	100	1.1E5	47	89
GEONET S	50000	10	281	99	99	99	99	99	180
GEONET S	500	10	288	99	99	99	99	99	180
GEONET S	50000	0	292	99	99	99	99	99	180
ELM L	500	0	308	76	97	300	8.0E7	99	65
ELM L	500	10	308	78	97	310	8.4E7	99	65
ELM S	500	0	315	74	98	250	2.0E7	99	66
ELM S	500	10	315	76	98	250	2.1E7	99	66
MASCONS L	500	10	369	3.6E11	5.5E12	1.7E8	0.63	1.4E14	
TNN L	500	10	408	570	1200	810	56000	1100	420
TNN S	500	0	409	1300	1100	2600	1.5E5	990	450
TNN S	50000	0	410	630	1100	1900	1.7E5	1100	570
TNN L	50000	0	411	550	1100	1800	3.4E5	1600	630
TNN S	500	10	412	770	340	1200	66000	240	630
TNN S	50000	10	413	640	1100	2300	2.2E5	1200	720
TNN L	50000	10	414	610	1100	2300	2.2E5	1200	800
TNN L	500	0	415	8900	1100	6300	3.2E5	810	930
ELM L	50000	10	416	9200	1000	54000	2.6E9	520	44000
ELM L	50000	0	417	9600	1100	56000	2.6E9	550	49000
ELM S	50000	10	418	9200	1000	54000	1.8E8	520	1.9E6
ELM S	50000	0	419	9700	1100	57000	1.9E8	550	2.0E6

error across the different metrics. The PINN-III L performs better near the surface of the asteroid, where the mascon model is known to struggle; however, the mascon model performs better in the high-altitude extrapolation regime where no training data was present.

Notably, five of the eight PINN-IIIs score in the top ten models, despite some of these models being trained under noisy or sparse data conditions. For example, the 5th best model is a PINN-III S, but this model is only trained with 500 data points. The next best model trained under these conditions is a Mascon L model which scores 9th and uses a two orders of magnitude more parameters. Similarly a PINN-III L performs the best among all models trained with 10% error on the acceleration vectors—scoring 6th across all models. Finally, a PINN-III L is also the best performer of all models trained on both sparse and noisy data, scoring in the upper 30th percentile and placing 18th overall. Table 3a sorts and ranks these models by their respective training conditions for a more compact summary of model performance and again highlights how PINN-III S and L remain the two highest scoring models across all tested conditions on a heterogenous density Eros.

Table 2 simultaneously illustrates the accuracy and robustness of the PINN-GM-III to these stressful data conditions while also exposing the brittleness of other gravity models—including past PINN-GM generations. For example, PINN-II L has the lowest error at the surface of any gravity model, averaging 2.5% across all 200,700 facets on the shape model. However, when tested in the extrapolation regime, PINN-II diverges. Similarly, the Mascon L has the highest score across all models when regressed on ideal training conditions, but when trained on sparse and noisy data sets, this model diverges in all but one metric, ranking 58th of 70 overall.

Past machine learning models exhibit greater concern, with the traditional neural networks, ELMs, PINN-I, and GeodesyNets consistently scoring in the lower 50th percentile. The candidate failure modes discussed in Sect. 3 are best captured by inspecting the interior, exterior, and extrapolation metrics. Note how these models exhibit deteriorating accuracy when evaluated on increasingly high altitude data — i.e. the interior values are generally lower than exterior values. Moreover, these vast majority of models—GeodesyNets excluded—meet the divergence criteria when tested beyond the bounds of the training data. These behaviors provide confirmation of the challenges many machine models face when tested across a more diverse set of experiments, and the PINN-GM-III avoids these same challenges due to the new suite of proposed modifications.
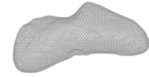
As a caveat, GeodesyNets are not prone to the exact same failure modes as the other machine models due to their fundamentally different architecture and evaluation procedure. Because the densities predicted are only computed within a unit volume, these models do not pose the same risk of feature or model divergence at high altitudes. That said, these models still struggle to converge as there does not exist sufficient amounts of data to reliably constrain the density function. Even with 50,000 noiseless data points, the GeodesyNets yield errors in excess of 80% across the different testing regimes. These results further highlight how sensitive and data intensive these alternative ML model implementations can be — illustrating the need for more nuanced discussion about how and when these models can break down. While the regression strategies used to fit these models sought to

**Table 3** Summarized rank for all models across four test asteroids
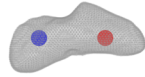
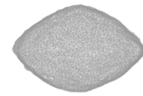| N	500		50000	
Error (%)	10	0	10	0
Model				
PINN III L	1	2	1	3
PINN III S	2	1	6	2
PINN II L	3	4	3	4
MASCONS L	15	3	2	1
PINN II S	4	6	5	5
MASCONS S	9	5	4	7
POLY S				6
PINN I L	6	13	7	9
PM -	5	7	10	12
POLY L				8
SH S	7	8	9	10
PINN I S	8	9	8	11
GEONET L	10	10	12	14
SH L	11	12	11	13
GEONET S	12	11	13	15
TNN S	17	16	14	16
ELM L	13	14	16	18
ELM S	14	15	17	19
TNN L	16	17	15	17

(a) Eros Heterogeneous

N	500		50000	
Error (%)	10	0	10	0
Model				
POLY L				1
PINN III S	2	2	1	4
PINN III L	2	2	2	3
PINN II L	3	4	5	6
POLY S				5
PINN II S	4	6	6	7
MASCONS L	15	3	3	2
MASCONS S	8	5	4	8
PM -	5	7	9	12
PINN I L	7	13	7	9
PINN I S	9	11	8	10
SH S	7	8	10	11
GEONET L	10	9	11	13
GEONET S	12	11	13	15
SH L	11	12	12	14
TNN S	16	16	14	17
ELM L	13	14	16	18
ELM S	14	15	17	19
TNN L	17	17	15	16

(b) Eros Homogeneous
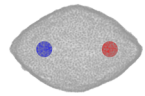
N	500		50000	
Error (%)	10	0	10	0
Model				
SH S	1	1	1	4
PINN III L	2	2	4	2
PINN III S	5	3	7	3
PINN II L	3	4	6	5
PINN II S	6	6	3	6
MASCONS L	15	5	5	1
PM -	4	7	9	11
MASCONS S	14	10	2	7
PINN I L	7	8	8	10
POLY L				8
PINN I S	9	12	10	12
POLY S				10
SH L	8	9	11	13
GEONET L	10	11	12	14
GEONET S	11	13	13	15
TNN L	16	17	14	16
ELM L	12	15	16	18
ELM S	14	14	17	19
TNN S	17	16	15	17

(c) Bennu Heterogeneous

N	500		50000	
Error (%)	10	0	10	0
Model				
POLY L				1
SH S	1	1	1	6
PINN III S	4	4	2	4
PINN III L	4	2	4	4
POLY S				5
PINN II L	5	5	6	7
PINN II S	6	6	5	8
MASCONS L	15	3	7	2
PM -	2	7	8	11
PINN I L	8	8	9	9
MASCONS S	13	10	3	10
PINN I S	9	12	10	12
SH L	7	9	12	13
GEONET L	10	11	11	14
GEONET S	11	13	13	15
ELM L	12	14	16	18
ELM S	14	15	17	19
TNN L	17	16	14	17
TNN S	16	17	15	16

(d) Bennu Homogeneous

remain close to the original implementations — as is discussed in Appendix D — it remains possible that different training configurations and practices could yield better performance.

To ensure the conclusions drawn through this study are generalizable, three additional gravity fields are tested. These analyses regress the same 70 models on a homogeneous density version of Eros, and a homo- and a heterogenous density version of the asteroid Bennu. The imposed heterogenous density profiles again remain $\pm 10\%$ mass at locations $\pm R/2$ along the x-axis of the respective asteroid. These four environments attempt to span common and exotic gravity fields: from nearly spherical to irregular geometries, and from homogeneous to heterogeneous densities profiles. By studying the performance across these different cases, the general behaviors of these models can be reasonably inferred for other candidate small bodies. The exact metric values for these additional cases are reported in Appendix D.7, and their score summaries across the different data conditions are reported in Table 3.

Across these three additional experiments, the PINN-GM-III remains competitive — consistently reporting the highest score among the machine learning models. Notably, the new scenarios do show the strength of the analytic models, such as the spherical harmonic model for the asteroid Bennu, and the polyhedral model for the constant density cases. Importantly, the results for the polyhedral model are not directly comparable to the other models reported, as the polyhedral shape was not constructed / regressed from the position and acceleration data. Its inclusion can only be used to infer the strength of the model in a parametric sense, where the polyhedral models of approximately 30,000 parameters achieves better performance than other models of the same size. Regardless, these results imply that the PINN-GM-III exhibits its greatest utility when modeling asteroids with irregular shapes and unknown density.

5.1.1 Inference Time

While Tables 2 and 3a highlight the relative accuracy of the available gravity models, another important comparison point is each model's evaluation speed. To characterize this, 1,000 randomly distributed test points are evaluated using each gravity model. The total evaluation time is measured, and an average inference time per sample is reported in Fig. 7.

For the large models, Fig. 7 shows that the large GeodesyNet (33,601 parameters) and the large polyhedral model (30,006 parameters/approximately 10,000 facets) are the slowest gravity models reporting approximately 40 and 10 ms for each evaluation respectively. The neural network models (PINN I-III and TNN) execute an order of magnitude faster, executing between 0.4 - 0.7 ms, followed by the mascon, spherical harmonic, and ELM models at 0.3 ms, 0.09 ms, and 0.08 ms respectively. For small models, the GeodesyNet is the slowest at 3 ms, followed by the remaining machine learning gravity models averaging between 0.3 and 0.6 ms. The small polyhedral model (204 parameters or 66 faces) is faster at 0.09 ms, and the mascon, spherical harmonic, and point mass models are the fastest, executing in less than 0.01 ms per evaluation.
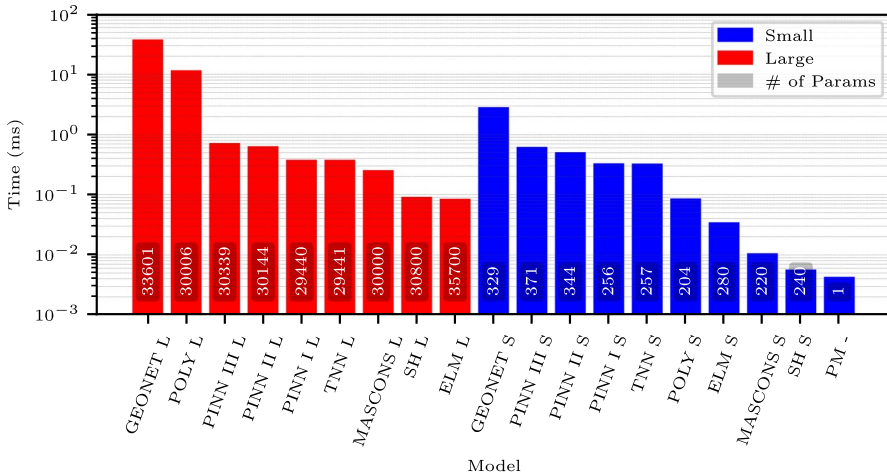
Fig. 7 Inference time per sample with the unique number of parameters in each model overlaid on the individual bars

These results highlight that speed is among the largest trade-offs for the PINN-GM-III, particularly in the small model regime. That said, the PINN-III models are not prohibitively slow, and their computational cost remains nearly constant irrespective of model size. For example, PINN-III L is two orders of magnitude larger than PINN-III S, but its computational cost is only twice as large. In contrast, the large polyhedral models cost appears to scale linearly, taking approximately 100 times longer for a model that is 100 times bigger. This suggests that the PINN-III remains a strong option if robustness and accuracy are the primary goal, and users can increase the size and performance of these models with relatively little added overhead.

6 Conclusions

Scientific machine learning and physics informed neural networks offer a compelling set of tools to address the gravity field modeling problem. Rather than using prescriptive analytic gravity models prone to various limitations, PINNs can learn convenient representations of the gravitational potential while maintaining desirable physics properties and assurances. While past machine learning gravity models have offered early glimpses into the potential advantages of these numerical solutions, this greater class of model are often susceptible to various pitfalls that had yet to be exposed or addressed. This paper highlights these failure cases for a variety of past models, and introduces design modifications within the machine learning architecture that can overcome these challenges. Taken together, these modifications form the third generation PINN gravity model, or PINN-GM-III. This model is designed to solve the problems of feature divergence, bias towards low-altitude samples, numerical instability, and extrapolation error, while also proposing a framework for

fusing analytic and numerical gravity models together for enhanced modeling accuracy. While these modifications are studied exclusively on the PINN gravity model, it should be noted that many of these modifications can likely be applied to other machine learning solutions to enhance their own robustness and generalizability.

Beyond introducing the PINN-GM-III, this manuscript also proposes new evaluation metrics to more comprehensively assess the modeling accuracy of various analytic and numerical gravity models in different orbital regimes. When trained on data from both homogeneous and heterogeneous density asteroids, the PINN-GM-III is shown to achieve competitive performance over past generations and other numerical and analytic models, while also demonstrating robustness to data sparse and noisy conditions. Future work will continue to investigate design modifications that can improve model performance, with a particular emphasis on returning to large celestial bodies and investigating ways in which the high-frequency components can be learned and represented more efficiently.

A PINN-GM-III Training Details

Like the PINN-GM-II, the PINN-GM-III is composed of a feed-forward multi-layer perception, preceded by a feature engineering/embedding layer. Skip connections are attached between the embedding layer and each of the hidden layers of the network, and the final layer uses linear activation functions to produce the network's prediction of the proxy potential. Unlike the PINN-GM-II, all of the experiments tested in this paper do not make use of the multi-constraint loss function due to findings presented in Appendix B.

The default hyperparameters used to train PINN-GM-III are listed in Table 4. The network is trained using the Adam optimizer with a learning rate of 2^{-8}. The learning rate is decayed when the validation loss plateaus for 1500 epochs. The default batch size is set to 2^{11} although many of the training data sizes are less than this value, so the batch size is automatically reduced to the size of the training data set when appropriate. The networks are trained for 8,192 epochs unless otherwise specified.

Table 4 Default Set of Hyperparameters for Neural Network Training

Hyperparameter	Value
learning_rate	2^{-8}
batch_size	2^{11}
num_epochs	2^{13}
optimizer	Adam
activation_function	GELU
lr_scheduler	Plateau
lr_patience	1,500
decay_rate	0.5
min_delta	0.001
min_lr	1e-6

The network is initialized using the Xavier uniform initialization scheme [41], and the network activation function is GELU [42]. The final layer weights are initialized to zero, which heuristically led to faster convergence and better performance.

PINN-GM-III preprocesses its training data differently than PINN-GM-II. Explicitly, the position and potential are normalized by the characteristic length $x^\star$ and maximum potential $U^\star$ respectively. Using these characteristic scalars, a time constant can be computed and used in conjunction with $x^\star$ to non-dimensionalize the accelerations. This manifests through:

$$x = \frac{\bar{x}}{x^\star}, \quad U = \frac{\bar{U}}{U^\star}, \quad a = \frac{\bar{a}}{a^\star} \tag{25}$$

where $x^\star, U^\star$, and $a^\star$ are the non-dimensionalization constants defined as:

$$x^\star = R, \quad U^\star = \max_i(\bar{U}_i - \bar{U}_{\text{LF},i}), \quad t^\star = \sqrt{\frac{x^{\star 2}}{U^\star}}, \quad a^\star = \frac{x^\star}{t^{\star 2}} \tag{26}$$

where R is the maximum radius of the celestial body, $\bar{U}_i$ is the true gravitational potential at the training datum at x_i, and U_{LF} is any low-fidelity potential contributions already accounted for within the PINN-GM (Sect. 3.5).

B Hyperparameter Optimization

Neural networks can be particularly sensitive to the correct choice of hyperparameters. This section seeks to characterize the sensitivity of the PINN-GM to these core hyperparameters, as well as determine the effect of different network sizes and quantity/quality of training data. Explicitly, network depth, width, batch size, learning rate, epochs, and loss function are varied, as well as the total amount data and its quality. As before, the heterogeneous density asteroid Eros detailed in Sect. 4 is used to provide training data. 32,768 training data are sampled uniformly between 0-3R, and the network performance is evaluated using a mean percent error averaged across 4096 separate validation samples for each of the proposed tests.

B.1 Network Size

The first test investigates the PINN-GM's sensitivity to network size, varying both network depth and width. The network depth is varied between 2, 4, 6, 8 hidden layers, and the width is varied between 8, 16, 32, 64 nodes per hidden layer. The remaining hyperparameters are kept fixed at the defaults provided in Table 4. The mean percent error of the models are reported in Fig. 8a with the corresponding model size, or total trainable parameters, overlaid on the bars.

Figure 8a demonstrates that PINN-GM-III performs well across a variety of different network sizes. The smallest networks will perform worse than the largest networks given their limited modeling capacity, yet despite this, all models remain below 5% error. The smallest model of 227 trainable parameters averages at 0.9%
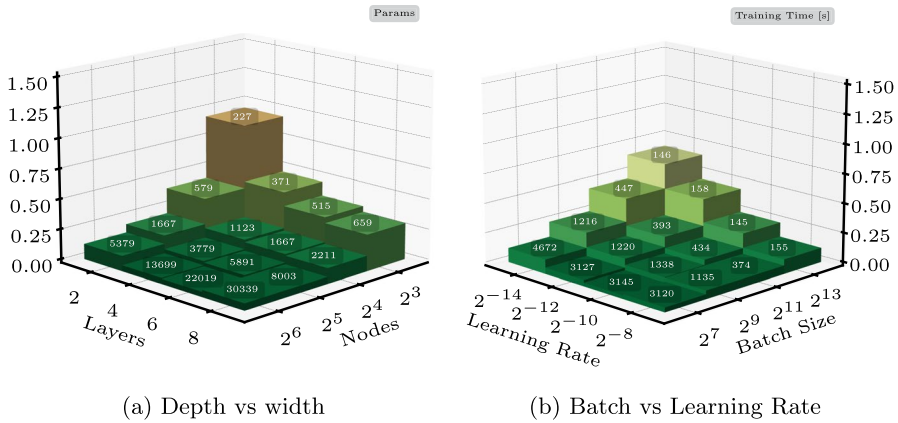
(a) Depth vs width   (b) Batch vs Learning Rate

Fig. 8 Core Hyperparameters

error, whereas the largest model with 30,339 parameters achieves 0.03% error. This figure illustrates that optimal models require a minimum of 16 nodes with four hidden layers. The performance continues to improve with larger models, but only marginally. Therefore, a model of six hidden layers and 16 nodes per layer (1667 parameters) or six hidden layers and 32 nodes (5891 parameters) are recommended.

B.2 Batch Size and Learning Rate

The next experiment studies the effect of batch size and learning rate on the PINN-GM. The network size is kept fixed at six hidden layers and 32 nodes per layer, but the batch size is varied between $\{2^7, 2^9, 2^{11}, 2^{13}\}$ and the learning rate is varied between $\{2^{-14}, 2^{-12}, 2^{-10}, 2^{-8}\}$. Again, the mean percent error of 4,096 samples are evaluated and shown in Fig. 8b with the total training time overlaid on the bars rather than parameter count.

Figure 8b demonstrates that there are a variety of learning rates and batch sizes that are acceptable for the PINN-GM. In general, smaller batch sizes produce more accurate models, albeit this comes with much longer training times. Regarding learning rate, higher is better. This is most likely coupled with the chosen learning rate scheduler, which decreases the learning rate by a factor of 0.5 every 1,500 epochs that the validation loss does not improve. Given this learning rate scheduler configuration, a learning rate of at least 2^{-8} is recommended for most applications.

B.3 Data Quantity and Epochs

Fixing the learning rate to 2^{-8} and the batch size to 2^{11}, a third experiment investigates the sensitivity of the PINN-GM to quantity of training data and length of training time. A PINN-III S and PINN-III L are prepared for this test. The PINN-III S uses the two layer, eight nodes per layer network configuration (227 parameters),
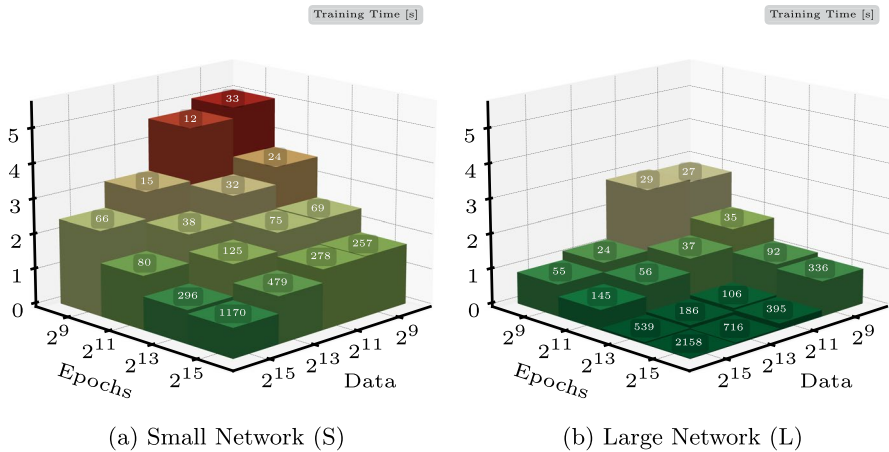
(a) Small Network (S)   (b) Large Network (L)

Fig. 9 Data vs Epochs

whereas PINN-III L uses the eight layer, 64 node configuration (30,339 parameters). Both the PINN-III S and L are trained with increasing amounts of training data ranging from $\{2^9, 2^{11}, 2^{13}, 2^{15}\}$ samples and increasing the total number of training epochs from $\{2^9, 2^{11}, 2^{13}, 2^{15}\}$. As before, the mean percent error is evaluated and presented with the length of training time in Fig. 9a and b.

Figure 9a shows that the PINN-III small is capable of achieving errors as low as 1% given sufficient quantities of training data and training time. Performance does decrease as the number of samples and epochs decrease; however, all models converge to solutions with less than 5% error. Similarly, Fig. 9b demonstrates that PINN-III L consistently benefits from additional training data and epochs, with the most accurate models reaching 0.04% error. Unlike the PINN-III S, the L models always remained below 3% error even in the low data and epoch regimes. Taken together, the results suggest that the PINN-GM-III should be trained for at minimum 8,192 epochs but can benefit from longer training if resources allow.

B.4 Data Quality and Physics Constraints

The final experiment investigates the effect of additional physics constraints on model performance. Explicitly, an added Laplacian constraint in the network loss function improved model performance for the PINN-GM-II when trained on noisy data [5]. This added constraint, however, also added considerable computational overhead to compute the second order derivative of the potential via automatic differentiation. This experiment seeks to determine if this term remains necessary given the new design modifications of the PINN-GM-III.

This experiment begins by corrupting every acceleration vector by adding 10% of their magnitude in a random direction to the truth vector as discussed in Sect. 5. The PINN-GM-III S and L are then trained on increasing amounts of this data ranging from $N = \{2^9, 2^{11}, 2^{13}, 2^{15}\}$. These models are trained once with the
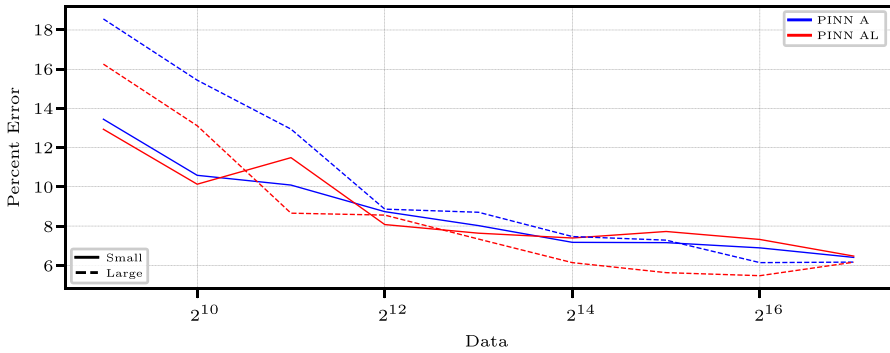
Fig. 10 Error as a function of training data and loss function

proposed loss function of Sect. 3.2 which only penalizes errors in the acceleration vector (PINN A)

$$\mathcal{L}_{\text{PINN A}} = \frac{1}{N} \sum_{i=0}^{N} \left(\left\| -\nabla \hat{U}(x_i|\theta) - a_i \right\| + \frac{\| -\nabla \hat{U}(x_i|\theta) - a_i \|}{\|a_i\|} \right) \tag{27}$$

and again with the Laplacian term added (PINN AL) or

$$\mathcal{L}_{\text{PINN AL}} = \frac{1}{N} \sum_{i=0}^{N} \left(\left\| -\nabla \hat{U}(x_i|\theta) - a_i \right\| + \frac{\| -\nabla \hat{U}(x_i|\theta) - a_i \|}{\|a_i\|} + \left\| \nabla^2 U(x_i) \right\| \right) \tag{28}$$

The batch size for this experiment is increased to 2^{15} to complete the experiment in a reasonable amount of training time, and the performance of the PINN-III L and S models as a function of training data are provided in Fig. 10.

Figure 10 shows that the noise in the training data does deteriorate model performance; however even with low quantities of training data, the PINN-GM S is able to achieve errors as low as 13%, just slightly above the noise floor. As the amount of training data increases, the S model reduces to approximately 7% error. The fact that these models are capable of regressing solutions beneath the noise floor is a testament to the physics-informed nature of these models. By leveraging the known dynamics of the system, the models can better ignore parts of the training data that are fundamentally inconsistent with the physics. That said, the additional Laplacian component added to the loss function (PINN AL) does not yield a compelling advantage to warrant its added training time. Therefore, it is advised that the PINN-GM does not incorporate the additional Laplacian constraint in its loss function.

C Comments on Past Machine Learning Performance

Table 1 highlights the general performance of past and present machine learning gravity models. All values are taken from their corresponding reference, but further context is warranted as each model assessed accuracy in different ways and on different asteroids. This section aims to provide relevant details regarding these metrics for completeness.

For the Gaussian process gravity model reported in Ref. [22], the number of model parameters are not explicitly reported but can be deduced. Gaussian processes are defined by their covariance matrix and kernel function. The covariance matrix scales as $\mathcal{O}(N^2)$ and the maximum number of data points used were $N = 3,600$. This suggests that the minimum number of parameters used in the model is 12,960,000. The accuracy for these models are also reported at fixed radii from the center of mass for each asteroid rather than across the full domain, so the values can be considered upper-bounds. Moreover, the model is shown to diverge at high altitudes and therefore is not valid globally.

For the extreme learning machine gravity model reported in Ref. [24], the model size is determined by the fact that there are 50,000 hidden nodes in the ELM. The random weights connecting the three inputs to the 50,000 hidden nodes constitutes the first 150,000 parameters, and the weights connecting the hidden layer to the acceleration output correspond to the next 150,000 parameters. Each node in the hidden layer can also have a bias, adding another 50,000 points summing to total of 350,000 total model parameters. The asteroid modeled is 25143 Itokawa, and the error in Ref. [24] is reported in terms of absolute terms rather than relative terms. Using Fig. 9, it can be approximated that the relative error varies between 1% and 10%.

In Ref. [30], the neural network gravity model is reported to use 512 nodes per hidden layer for 6 hidden layers. The approximate number of weights and biases can be approximated by squaring the number of nodes per hidden layer and multiplying by the total number of layers minus one, and adding the biases for each node ($\approx 512^2 * (6-1) + 512 * 6 = 1,313,792$ parameters). The paper reports 1,000,000 training data were generated and divided into an 8:2 ratio between training and testing data, yielding 800,000 training data. The asteroid investigated is also 433-Eros and the average relative error of the test set is reported as 0.35% in their Table 3.

For GeodesyNets [25], SIRENs of 9 hidden layers with 100 nodes each are used ($\approx 100^2 * (9-1) + 8 * 100 = 80,800$). Four asteroids are studied: Bennu, Churyumov-Gerasimenko, Eros, and Itokawa. In their supplementary materials (Table S4), the relative error about Eros is reported at three characteristic altitudes. At their lowest altitude, the average error is 0.571% and their highest altitude is 0.146%. In attempts to quantify error across the entire high and low altitude regime, these values are averaged for the reported value of 0.359%. The number of training data referenced is a result of the original paper sampling 1,000 data points every 10 iterations where the models were trained for 10,000 iterations, thereby equating to 1,000,000 training data.

Finally for the PINN-GM-III, the average error reported for a network trained with 8 hidden layers with 16 nodes per layer. The model was trained on 4,096 data points distributed between 0-10R using the same hyperparameters specified in Appendix B, and achieved an average error on the validation set of 0.3%.

D. Regression Details

In Sect. 5, multiple gravity models are regressed and tested on a variety of data conditions. The regression procedure for the neural networks is consistent with prior explanations; however, the remaining models require further context. This section aims to provide that context, explaining how each of the remaining gravity models are regressed on the data discussed in Sect. 5.

D.1 Spherical Harmonics

Spherical harmonic gravity models are regressed by solving a linear system for their $p = N(N + 1)$ total Stokes coefficients, where N is the degree of the model. This system takes the following form:

$$a = Hc \tag{29}$$

$$H^T a = H^T H c \tag{30}$$

$$(H^T H)^{-1} H^T a = c \tag{31}$$

where c are the vector of spherical harmonic coefficients C_{lm} and S_{lm}, a are the vector of accelerations, and H is the jacobian $\frac{\partial a}{\partial c}|_r$, and r is the position vector for each test point. Direct least square solutions may be appropriate for small systems, but when many harmonics need to be regressed and/or the amount of data used to regress them increases, least squares becomes computationally infeasible. Moreover, high-degree spherical harmonic models pose numerical instabilities as the observability of the harmonics decays rapidly as $(R/r)^l$, often making $H^T H$ ill-conditioned.

Kaula's rule — a form of ridge regression that is used to regularize the spherical harmonic coefficients — can help eliminate the ill-conditioned nature of the high-degree regression. Rather than seeking to minimize the mean squared error solution

$$L(c) = \|a - Hc\|^2 \tag{32}$$

ridge regression adds a penalty term to ensure the coefficients decay in magnitude for higher degree harmonics through

$$L(c) = \|a - Hc\|^2 + \|\Gamma c\|^2 \tag{33}$$

where Γ are the regularization matrix defined through Kaula's rule

$$\Gamma_{ll} = \begin{cases} \frac{\alpha}{l^2} & l > 0 \\ 1 & l = 0 \end{cases} \tag{34}$$

where α is a user specified constant typically chosen through cross validation. The corresponding solution to the ridge regression then becomes

$$(H^T H + \Gamma)^{-1} H^T \boldsymbol{a} = \boldsymbol{c} \tag{35}$$

which yields a solution with increasingly small spherical harmonic coefficients at high-degree. This simultaneously removes the ill-posed nature of the original regression and also provides a framework for supporting low-data regression.

While using Kaula's rule mitigates the ill-posedness of the original regression, it remains computationally expensive to invert $(H^T H + \Gamma)$ for large datasets. This is solved with two strategies: (1) recursive least squares and (2) iterative coefficient regression. Recursive least square sequentially feeds in small batches of data to maintain computational tractability, while iterative coefficient regression regresses low-degree harmonics before the high-degree harmonics.

For the regression used in Sect. 5, recursive least squares is performed in batches of 100 position / acceleration pairs using the following recursion relationships

$$K_{i+1}^{-1} = K_i^{-1} - K_i^{-1} H_k^T (I + H_k K_i^{-1} H_k^T)^{-1} H_k K_i^{-1} \tag{36}$$

$$\boldsymbol{c}_{k+1} = \boldsymbol{c}_k + K_{i+1}^{-1} H_k^T (\boldsymbol{a}_k - H_k \boldsymbol{c}_k) \tag{37}$$

where $K_0^{-1} = (H_0^T H_0 + \Gamma)^{-1}$, and the iterative coefficient regression is performed by only regressing 5,000 coefficients at a time. In addition, samples beneath the Brillouin surface $r < R$ are purposefully omitted from the regression due to the $(R/r)^l$ scaling in the harmonic model. For very high degree models, this term diverges when evaluated on sub-Brillouin samples, which breaks the regression.

D.2 Mascons

Like spherical harmonics, mascons regression is also a form of least squares regression. Specifically, a set of N mascons are uniformly distributed through the volume of the shape model, and their corresponding masses are regressed. Ideally, the mascon regression should also include a non-linear constraint applied to all mass elements that ensures they always have a value greater than zero to ensure compliance with physics — i.e. there are no such things as negative masses. This physical compliance, however, often comes at a major loss of modeling accuracy. Therefore, Sect. 5 allows for negative masses to be regressed to give the model the strongest cases against the PINN-GM.

In addition to allowing negative masses, the mascon regression also requires iterative fitting when the model sizes are large and there exists much data. To do this, the mascons are distributed and fit in batches of 500 using all available data. After each batch is fit, their contribution to the acceleration vectors are removed, the next batch is randomly distributed throughout the volume, and the model fits to

the acceleration residuals. This process is repeated until the total mass elements are reached.

The total number of parameters for a mascon model correspond with four parameters per mascon, corresponding to the three-component position of the mass, r_k, and it's associated gravitational parameter μ_k. Therefore the total parameter count for the mascon model is $p = 4N$.

D.3 Extreme Learning Machines

Extreme learning machines regression also closely resembles that of the spherical harmonic regression in that it also uses ridge regression and recursive least squares optimization. Explicitly, the ELM models are regressed by applying a fixed ridge regression matrix $\Gamma_{\text{ELM}} = \alpha \mathbb{I}$ and breaking the dataset into batches to be applied recursively using Eqs. 37. Notably, the ELM also relies on a random non-linear projection into a higher-dimensional space before the linear regression as described in detail in [24]. For the comparison study in this paper, all input and output data are preprocessed using a min-max transformation to [0, 1] and applying a sigmoid activation function at the hidden nodes. The total parameter count for these models can be computed by summing the weights from the three inputs to the hidden layer, the biases of the hidden layer, and the weights to the three outputs, totaling to: $3N + N + 3N = 7N$.

D.4 Polyhedral Models

The regression of polyhedral models follows an entirely unrelated process to the other gravity models, requiring image data to construct the shape model of the body using stereophotoclinometry. Given that these shape models cannot be computed directly from position and acceleration data, Sect. 5 instead only includes the performance of polyhedral models that are of similar parameter counts to the other models and assumes these models are perfectly regressed. The model sizes of the polyhedral models are computed by summing the positions of each vertex in the model, and the indices (stored as long integers) which identify the vertices that comprise each face. Therefore the total model size can be computed via $p = 3V + \frac{3F}{2}$ where F corresponds to the number of facets and V corresponds to the number of vertices. The $3F/2$ captures the fact that long integers are half the size of 32-bit floats.

D.5 Neural Network Models

The regression of the neural network models is detailed at the beginning of Sect. 3. The parameter count for these models corresponds with the weights connecting the $l - 1$ hidden layers, the weights connecting the inputs to the first hidden layer, the weights connecting the last hidden layer to the outputs, and the biases for all nodes. Therefore, the total parameters can be estimated with $3N + N^2(l - 1) + N + lN$ respectively where N is the width of the network and l is the number of hidden

layers. Note that the traditional neural networks will have three outputs, whereas PINNs only have one.

D.6 GeodesyNet

The regression of the GeodesyNets follows the direct training method outlined in Ref. [25]. The direct training method is chosen over the differential training method because the former does not assume knowledge of the asteroid shape — therefore matching the data conditions for the other machine learning gravity models.

For their regression, the GeodesyNets use 300,000 quadrature points to numerically evaluate the acceleration at a field point. In the original paper, the models are trained for 10,000 iterations, where 1,000 new training data are randomly generated in the unit volume — i.e. near the body — once every 10 iterations. In this manuscript, the data is generated a priori — either 500 or 50,000 points spanning 0-10R — and 1,000 data points are randomly sampled from this generated data once every 10 iterations to mimic the original training configuration. The architecture of the GeodesyNet matched that of the other machine learning models. The small GeodesyNet had four hidden layers and eight nodes per hidden layer, and the large GeodesyNet had eight hidden layers with 64 nodes per hidden layer. All training data were non-dimensionalized to ensure that all of the asteroid density remains within the unit cube.

The performance of the GeodesyNets in the comparative study is believed to be the result of the small dataset size and its distribution across the training volume. As discussed, the original GeodesyNet training paper achieves high accuracy by training on 1,000,000 data that exist within the unit cube or sphere; however, the data used in the comparative study experiments span a spherical volume out to 10R. This yields a volume that is approximately 1,000 times larger (volume scales cubically) with considerably less data occupying that volume. As such, the GeodesyNets do not have enough data to regress an accurate density field across the entire unit volume.

D.7 Supplemental Comparison Study Results

The following tables report the exact metric values used for the comparison study, as well as the associated rank across all models.

D.7.1 Eros Heterogeneous Density

See Tables 5 and 6.

Table 5 Metric values for each gravity model trained on heterogeneous Eros gravity field data

Model	N	Error (%)	Score	Planes (%)	Interior (%)	Exterior (%)	Extrap. (%)	Surface (%)	Traj. (km)
MASCONS L	50000	0	14	1.2E-2	0.13	5.4E-4	3.4E-4	8.3	1.7E-3
PINN III S	50000	0	29	0.17	1.2	2.8E-2	0.065	11	4.2E-2
PINN III L	50000	0	34	1.3E-2	0.099	1.7E-3	1.3	3.2	2.7E-3
PINN II L	50000	0	84	4.7E-2	0.080	0.070	3.2E14	2.5	2.8E-2
PINN III S	500	0	88	2.2	18	0.51	0.13	35	0.89
PINN III L	50000	10	89	3.4	3.7	2.0	0.28	13	6.3
PINN II S	50000	0	97	0.22	0.57	0.31	2.1E7	6.2	0.11
PINN III L	500	0	98	1.6	17	0.23	0.41	52	0.67
MASCONS L	500	0	101	0.82	14	2.5E-4	9.8E-8	350	1.3E-7
POLY S			110	6.2	16	3.3	0.34	23	1.9
MASCONS L	50000	10	113	1.3	9.3	0.16	0.15	610	0.37
PINN II L	500	0	115	0.69	5.1	0.22	3.2E12	25	0.18
MASCONS S	50000	0	117	3.1	4.4	2.8	2.7	26	12
POLY L			119	6.5	16	3.5	0.35	23	2.3
PINN II L	50000	10	122	1.4	2.7	2.2	4.4E14	8.8	0.68
MASCONS S	50000	10	124	3.3	4.6	3.0	3.0	26	13
MASCONS S	500	0	135	2.7	12	1.9	1.8	71	6.7
PINN III L	500	10	140	6.2	23	3.1	0.36	45	6.0
PINN III S	50000	10	140	1.5	2.9	1.8	1.1E8	9.4	6.4
PINN II S	500	0	152	1.8	7.8	1.7	5.4E6	32	3.2
PINN III S	500	10	159	8.2	51	3.3	0.36	76	5.0
PINN III S	50000	10	160	8.1	51	2.3	0.32	79	17
PINN I L	50000	0	174	10	4.3	8.4	1.6E5	8.4	12
PINN II L	500	10	179	5.7	13	4.8	5.6E11	32	5.4
PINN II S	500	10	182	6.4	13	6.5	8.6E6	33	2.7
PINN I L	50000	10	183	13	5.4	16	1.8E5	14	7.0
SH S	50000	0	202	24	370	0.45	0.22	1900	1.1
PINN I S	50000	10	202	14	9.5	31	1.6E5	31	20
PINN I S	50000	0	203	16	8.6	38	1.2E5	33	17
SH S	50000	10	211	24	320	0.53	0.29	2200	1.4
PM -	500	0	211	50	43	50	50	70	110
PM -	500	10	215	50	43	51	51	70	110
SH S	500	0	217	17	340	1.2	0.86	950	0.97
PM -	50000	10	220	53	45	53	53	70	110
PINN I L	500	10	221	29	16	48	46000	34	22
PM -	50000	0	224	53	45	53	53	70	110
SH S	500	10	227	27	580	1.6	0.74	1500	2.4
PINN I S	500	10	235	33	21	69	46000	37	31
SH L	50000	0	239	inf	1.3E99	0.45	0.22	8.8E116	1.1
PINN I S	500	0	242	41	24	75	93000	41	42
GEONET L	500	0	245	83	84	83	83	87	160
SH L	50000	10	247	inf	1.8E99	0.53	0.29	1.2E117	1.4
MASCONS S	500	10	251	34	98	29	28	460	64
GEONET L	50000	0	252	87	87	87	87	87	160
GEONET L	500	10	255	87	90	86	86	93	160
GEONET L	50000	10	260	87	87	87	87	87	160
GEONET S	500	0	273	99	99	99	99	99	180
SH L	500	10	275	inf	2.5E97	5.0	0.35	1.0E115	4.4
SH L	500	0	278	inf	9.8E96	5.0	0.35	2.1E115	4.4
PINN I L	500	0	280	52	41	100	1.1E5	47	89
GEONET S	50000	0	281	99	99	99	99	99	180
GEONET S	500	10	288	99	99	99	99	99	180
GEONET S	50000	0	292	99	99	99	99	99	180
ELM L	500	0	308	76	97	300	8.0E7	99	65
ELM L	500	10	308	78	97	310	8.4E7	99	65
ELM S	500	0	315	74	98	250	2.0E7	99	66
ELM S	500	10	315	76	98	250	2.1E7	99	66
MASCONS L	500	10	369	3.6E11	5.5E12	1.7E8	0.63	1.4E14	
TNN L	500	10	408	570	1200	810	56000	1100	420
TNN S	500	0	409	1300	1100	2600	1.5E5	990	450
TNN S	50000	0	410	630	1100	1900	1.7E5	1100	570
TNN L	50000	0	411	550	1100	1800	3.4E5	1600	630
TNN S	500	10	412	770	340	1200	66000	240	630
TNN S	50000	10	413	640	1100	2300	2.2E5	1200	720
TNN L	50000	10	414	610	1100	2300	2.2E5	1200	800
TNN L	500	0	415	8900	1100	6300	3.2E5	810	930
ELM L	50000	10	416	9200	1000	54000	2.6E9	520	44000
ELM L	50000	0	417	9600	1100	56000	2.6E9	550	49000
ELM S	50000	10	418	9200	1000	54000	1.8E8	520	1.9E6
ELM S	50000	0	419	9700	1100	57000	1.9E8	550	2.0E6

Table 6 Rank values for each gravity model trained on heterogeneous Eros gravity field data

Model	N	Error (%)	Score (Rank)	Planes (Rank)	Interior (Rank)	Exterior (Rank)	Extrap. (Rank)	Surface (Rank)	Traj. (Rank)
MASCONS L	50000	0	1	1.0	3.0	2.0	2.0	4.0	2.0
PINN III S	50000	0	2	4.0	5.0	4.0	3.0	8.0	5.0
PINN III L	50000	0	3	2.0	2.0	3.0	22	2.0	3.0
PINN II L	50000	0	4	3.0	1.0	5.0	70	1.0	4.0
PINN III S	500	0	5	13	26	12	4.0	22	11
PINN III L	50000	10	6	17	8.0	20	8.0	9.0	27
PINN II S	50000	0	7	5.0	4.0	9.0	70	3.0	6.0
PINN III L	500	0	8	11	25	8.0	18	27	9.0
MASCONS L	500	0	9	7.0	21	1.0	1.0	70	1.0
POLY S			10	20	23	27	12	11	17
MASCONS L	50000	10	11	8.0	16	6.0	5.0	70	8.0
PINN II L	500	0	12	6.0	12	7.0	70	13	7.0
MASCONS S	50000	0	13	15	10	23	24	14	31
POLY L			14	22	24	28	15	12	18
PINN II L	50000	10	15	9.0	6.0	21	70	6.0	10
MASCONS S	50000	10	16	16	11	24	25	15	33
MASCONS S	500	0	17	14	18	19	23	32	29
PINN III L	500	10	19	19	28	25	17	25	26
PINN II S	50000	10	19	10	7.0	18	70	7.0	28
PINN II S	500	0	20	12	14	17	70	18	21
PINN III S	500	10	21	24	36	26	16	33	24
PINN III S	50000	10	22	23	35	22	11	34	35
PINN I L	50000	0	23	25	9.0	33	70	5.0	32
PINN II L	500	10	24	18	20	29	70	17	25
PINN II S	500	10	25	21	19	32	70	20	20
PINN I L	50000	10	26	26	13	34	70	10	30
SH S	50000	0	28	30	70	11	7.0	70	14
PINN I S	50000	10	28	27	17	36	70	16	36
PINN I S	50000	0	29	28	15	37	70	19	34
SH S	50000	10	31	31	70	14	10	70	16
PM -	500	0	31	37	31	39	27	31	46
PM -	500	10	32	38	32	40	28	30	47
SH S	500	0	33	29	70	15	21	70	12
PM -	50000	10	34	40	33	41	29	29	48
PINN I L	500	10	35	33	22	38	70	21	37
PM -	50000	0	36	41	34	42	30	28	49
SH S	500	10	37	32	70	16	20	70	19
PINN I S	500	10	38	34	27	43	70	23	38
SH L	50000	0	39	70	70	10	6.0	70	13
PINN I S	500	0	40	36	29	44	70	24	39
GEONET L	500	0	41	46	37	45	31	36	50
SH L	50000	10	42	70	70	13	9.0	70	15
MASCONS S	500	10	43	35	45	35	26	70	40
GEONET L	50000	0	44	47	38	47	33	35	52
GEONET L	500	10	45	48	40	46	32	38	51
GEONET L	50000	10	46	49	39	48	34	37	53
GEONET S	500	0	47	50	46	49	35	39	54
SH L	500	10	48	70	70	30	13	70	22
SH L	500	0	49	70	70	31	14	70	23
PINN I L	500	0	50	39	30	70	70	26	45
GEONET S	50000	10	51	51	47	50	36	42	55
GEONET S	500	10	52	52	48	51	37	44	56
GEONET S	50000	0	53	53	49	52	38	43	57
ELM L	500	0	55	44	42	70	70	41	41
ELM L	500	10	55	45	41	70	70	40	42
ELM S	500	0	57	42	44	70	70	46	43
ELM S	500	10	57	43	43	70	70	45	44
MASCONS L	500	10	58	70	70	70	19	70	
TNN L	500	10	59	70	70	70	70	70	58
TNN S	500	0	60	70	70	70	70	70	59
TNN S	50000	0	61	70	70	70	70	70	60
TNN L	50000	0	62	70	70	70	70	70	61
TNN S	500	10	63	70	70	70	70	70	62
TNN S	50000	10	64	70	70	70	70	70	63
TNN L	50000	10	65	70	70	70	70	70	64
TNN L	500	0	66	70	70	70	70	70	65
ELM L	50000	10	67	70	70	70	70	70	66
ELM L	50000	0	68	70	70	79	70	70	67
ELM S	50000	10	69	70	70	79	70	70	68
ELM S	50000	0	70	70	70	70	70	70	69

D.7.2 Eros Homogeneous Density

See Tables 7 and 8.

Table 7 Metric values for each gravity model trained on homogeneous Eros gravity field data

Model	N	Error (%)	Score	Planes (%)	Interior (%)	Exterior (%)	Extrap. (%)	Surface (%)	Traj. (km)
POLY L			11	2.7E-3	1.2E-2	8.9E-4	3.1E-5	0.090	2.2E-3
MASCONS L	50000	0	20	1.2E-2	0.11	5.2E-4	3.4E-4	8.0	1.5E-3
PINN III L	50000	0	37	2.1E-2	0.20	3.5E-3	0.52	3.7	3.2E-2
PINN III S	50000	0	38	0.22	1.5	3.8E-2	1.3E-2	11	0.14
POLY S			47	0.69	2.7	0.26	1.7E-2	5.7	0.36
PINN III S	50000	10	85	2.2	3.3	1.5	3.8E-2	11	4.1
PINN II L	50000	0	89	4.4E-2	0.090	0.066	1.2E15	2.4	3.3E-2
PINN III L	500	0	91	1.4	16	0.20	0.061	44	0.073
PINN III S	500	0	91	2.0	19	0.34	1.6E-2	41	0.31
MASCONS L	500	0	105	0.82	13	2.4E-4	9.8E-8	330	6.2E-8
PINN III L	50000	10	105	3.3	3.5	1.9	4.1E-2	13	9.1
PINN II S	50000	0	113	0.29	0.71	0.53	1.9E6	7.9	0.44
MASCONS S	50000	0	125	3.0	4.3	2.7	2.7	25	11
MASCONS L	50000	10	129	1.3	9.3	0.14	0.12	610	0.30
MASCONS S	50000	10	132	3.2	4.4	2.9	2.8	25	11
PINN II L	500	0	132	0.68	5.2	0.20	8.8E12	26	0.57
PINN II L	50000	10	133	1.4	2.7	2.7	3.4E14	7.8	1.0
MASCONS S	500	0	139	2.7	12	2.0	2.0	62	7.0
PINN II S	50000	10	151	1.9	2.8	3.0	4.5E6	12	1.3
PINN I L	50000	0	157	6.6	2.3	5.1	31000	6.3	2.4
PINN II S	500	0	158	1.7	6.4	1.7	3.7E5	27	1.2
PINN I L	50000	10	179	9.7	4.7	12	64000	9.7	6.6
PINN III S	500	10	183	28	63	9.9	4.3E-2	89	52
PINN III L	500	10	183	26	64	9.3	4.9E-2	89	51
PINN II L	500	10	186	5.6	13	5.0	3.5E11	31	4.3
PINN I S	50000	0	197	11	5.1	25	24000	23	22
PINN I S	50000	10	197	11	5.9	18	62000	20	16
PINN II S	500	10	199	6.3	13	6.0	3.3E6	34	11
PM -	500	0	210	51	43	52	52	71	110
SH S	50000	0	212	25	370	0.48	0.24	2100	1.0
PM -	500	10	214	51	43	52	52	71	110
PM -	50000	10	218	53	44	53	53	70	110
PM -	50000	0	222	53	44	54	53	70	110
SH S	500	0	222	16	330	1.3	0.94	910	1.1
SH S	500	10	223	25	330	0.57	0.33	2200	1.4
PINN I L	500	10	228	38	17	69	1.1E5	37	13
SH S	500	0	228	26	550	1.6	0.82	1500	1.3
GEONET L	50000	10	241	58	88	55	54	96	120
MASCONS S	500	10	243	32	88	27	27	390	59
GEONET L	500	0	243	81	82	81	81	86	160
GEONET L	50000	0	249	87	87	87	87	87	160
SH L	50000	0	250	inf	1.4E99	0.48	0.24	9.8E116	1.0
PINN I S	500	10	252	44	21	95	1.2E5	45	69
GEONET L	500	10	260	88	92	88	88	94	170
SH L	50000	10	262	inf	2.2E99	0.57	0.33	1.3E117	1.4
GEONET S	500	0	268	97	97	97	97	97	180
PINN I S	500	0	268	47	21	110	1.2E5	37	66
SH L	500	10	270	inf	2.2E97	3.7	4.0E-2	1.1E115	6.5
SH L	500	0	273	inf	1.3E97	3.7	4.0E-2	2.1E115	6.5
GEONET S	50000	0	278	99	99	99	99	99	180
GEONET S	500	10	284	99	99	99	99	99	180
PINN I L	500	0	293	88	41	160	1.4E5	43	120
GEONET S	50000	10	294	100	100	100	100	100	180
ELM L	500	0	310	77	98	300	7.8E7	99	64
ELM L	500	10	310	78	98	300	8.0E7	99	65
ELM S	500	0	315	75	98	250	2.0E7	99	66
ELM S	500	10	315	76	98	250	2.0E7	99	66
MASCONS L	500	10	370	3.5E11	5.1E12	1.7E8	0.64	1.3E14	
TNN S	50000	10	408	470	1100	970	90000	1000	290
TNN L	50000	10	409	470	1100	920	81000	1100	300
TNN S	500	0	410	900	1100	2500	1.9E5	1000	390
TNN S	500	10	411	730	1100	2100	1.5E5	1000	490
TNN L	50000	0	412	420	1100	450	2.7E5	1400	490
TNN S	50000	0	413	430	1000	490	33000	1100	550
TNN L	500	10	414	680	1000	1700	1.4E5	930	560
TNN L	500	0	415	610	1100	1800	1.7E5	860	570
ELM L	50000	0	416	8500	920	50000	2.5E9	460	3100
ELM L	50000	10	417	8200	890	48000	2.5E9	450	5600
ELM S	50000	10	418	8300	900	48000	1.7E8	450	1.6E6
ELM S	50000	0	419	8500	930	50000	1.7E8	470	1.7E6

Table 8 Rank values for each gravity model trained on homogeneous Eros gravity field data

Model	N	Error (%)	Score (Rank)	Planes (Rank)	Interior (Rank)	Exterior (Rank)	Extrap. (Rank)	Surface (Rank)	Traj. (Rank)
POLY L			1	1.0	1.0	3.0	2.0	1.0	3.0
MASCONS L	50000	0	2	2.0	3.0	2.0	3.0	8.0	2.0
PINN III L	50000	0	3	3.0	4.0	4.0	19	3.0	4.0
PINN III S	50000	0	4	5.0	6.0	5.0	4.0	11	7.0
POLY S			5	8.0	9.0	10	6.0	4.0	10
PINN III S	50000	10	6	16	11	18	7.0	10	23
PINN II L	50000	0	7	4.0	2.0	6.0	70	2.0	5.0
PINN III L	500	0	9	12	25	9.0	13	26	6.0
PINN III S	500	0	9	15	27	11	5.0	24	9.0
MASCONS L	500	0	11	9.0	23	1.0	1.0	70	1.0
PINN III L	50000	10	11	20	12	21	10	13	29
PINN II S	50000	0	12	6.0	5.0	14	70	7.0	11
MASCONS S	50000	0	13	18	13	24	24	16	30
MASCONS L	50000	10	14	10	20	7.0	14	70	8.0
MASCONS S	50000	10	16	19	14	25	25	17	32
PINN II L	500	0	16	7.0	17	8.0	70	18	12
PINN II L	50000	10	17	11	8.0	23	70	6.0	15
MASCONS S	500	0	18	17	21	22	23	28	28
PINN II S	50000	10	19	14	10	26	70	12	19
PINN I L	50000	0	20	23	7.0	30	70	5.0	22
PINN II S	500	0	21	13	19	20	70	19	17
PINN I L	50000	10	22	24	15	34	70	9.0	27
PINN III S	500	10	24	32	35	33	11	35	37
PINN III L	500	10	24	31	36	32	12	36	36
PINN II L	500	10	25	21	22	29	70	20	24
PINN I S	50000	0	27	25	16	36	70	15	35
PINN I S	50000	10	27	26	18	35	70	14	34
PINN II S	500	10	28	22	24	31	70	21	31
PM -	500	0	29	37	31	38	27	32	45
SH S	50000	0	30	29	70	13	16	70	14
PM -	500	10	31	38	32	39	28	31	46
PM -	50000	10	32	39	33	40	29	30	47
PM -	50000	0	34	40	34	41	30	29	48
SH S	500	0	34	27	70	17	22	70	16
SH S	50000	10	35	28	70	16	18	70	21
PINN I L	500	10	37	34	26	43	70	22	33
SH S	500	10	37	30	70	19	21	70	18
GEONET L	50000	10	38	41	40	42	31	38	49
MASCONS S	500	10	40	33	39	37	26	70	38
GEONET L	500	0	40	46	37	44	32	33	51
GEONET L	50000	0	41	47	38	45	33	34	52
SH L	50000	0	42	70	70	12	15	70	13
PINN I S	500	10	43	35	29	47	70	27	44
GEONET L	500	10	44	49	41	46	34	37	53
SH L	50000	10	45	70	70	15	17	70	20
GEONET S	500	0	47	50	42	48	35	39	54
PINN I S	500	0	47	36	28	70	70	23	41
SH L	500	10	48	70	70	27	8.0	70	25
SH L	500	0	49	70	70	28	9.0	70	26
GEONET S	50000	0	50	51	47	49	36	40	55
GEONET S	500	10	51	52	48	50	37	41	56
PINN I L	500	0	52	48	30	70	70	25	50
GEONET S	50000	10	53	53	49	51	38	46	57
ELM L	500	0	55	44	44	70	70	43	39
ELM L	500	10	55	45	43	70	70	42	40
ELM S	500	0	57	42	46	70	70	45	42
ELM S	500	10	57	43	45	70	70	44	43
MASCONS L	500	10	58	70	70	70	20	70	
TNN S	50000	10	59	70	70	70	70	70	58
TNN L	50000	10	60	70	70	70	70	70	59
TNN S	500	0	61	70	70	70	70	70	60
TNN S	500	10	62	70	70	70	70	70	61
TNN L	50000	0	63	70	70	70	70	70	62
TNN S	50000	0	64	70	70	70	70	70	63
TNN L	500	10	65	70	70	70	70	70	64
TNN L	500	0	66	70	70	70	70	70	65
ELM L	50000	0	67	70	70	70	70	70	66
ELM L	50000	10	68	70	70	70	70	70	67
ELM S	50000	10	69	70	70	79	70	70	68
ELM S	50000	0	70	70	70	70	70	70	69

D.7.2 Bennu Heterogeneous Density

See Tables 9 and 10.

Table 9 Metric values for each gravity model trained on heterogeneous Bennu gravity field data

Model	N	Error (%)	Score	Planes (%)	Interior (%)	Exterior (%)	Extrap. (%)	Surface (%)	Traj. (km)
MASCONS L	50000	0	13	1.0E-2	0.37	1.4E-3	7.1E-4	3.3	5.4E-5
PINN III L	50000	0	26	1.4E-2	0.62	1.6E-3	0.31	3.0	4.4E-5
PINN III S	50000	0	31	0.11	1.6	3.1E-2	3.2E-2	3.3	1.9E-4
SH S	50000	0	39	0.22	3.9	0.072	1.7E-3	6.1	1.2E-4
SH S	50000	10	66	0.35	3.8	0.21	0.16	6.0	1.0E-2
SH S	500	0	79	0.37	4.1	0.23	0.16	6.2	1.3E-2
PINN II L	50000	0	93	4.5E-2	0.39	3.9E-2	3.5E16	1.4	4.0E-3
PINN III L	500	0	94	0.49	17	0.12	0.053	28	1.8E-4
PINN II S	50000	0	102	0.14	1.2	0.16	2.8E5	2.6	2.3E-3
MASCONS S	50000	10	111	1.1	5.7	0.92	0.83	10	0.066
PINN III S	500	0	113	0.72	19	0.20	0.14	32	1.9E-3
MASCONS S	50000	0	116	1.3	5.6	1.0	0.97	10	0.075
PINN II L	500	0	122	0.28	5.3	0.13	4.1E12	8.4	9.1E-4
MASCONS L	500	0	123	0.42	25	1.7E-2	5.0E-8	140	2.6E-11
PINN II S	500	0	129	0.48	4.1	0.62	2.6E7	6.2	1.8E-3
SH S	500	10	136	2.4	15	1.3	0.81	19	0.065
POLY L			137	6.3	14	3.5	0.35	15	9.4E-3
PINN II S	50000	10	147	1.1	2.6	1.4	4.7E7	4.4	0.084
POLY S			148	6.4	15	3.5	0.36	16	1.1E-2
PINN I L	50000	0	148	1.6	1.7	1.4	1.0E5	2.8	0.12
PINN III L	50000	10	151	2.3	14	1.6	0.33	20	0.24
MASCONS L	50000	10	154	0.79	27	0.17	0.10	160	7.5E-3
PINN II L	50000	10	157	1.2	4.1	1.7	3.2E13	6.1	0.072
PINN III S	50000	10	174	4.5	19	1.9	0.33	30	0.31
PINN I L	50000	10	180	3.8	4.0	10	83000	6.2	0.11
PM -	50000	10	184	6.9	17	4.1	2.2	20	0.17
PINN III L	500	10	184	5.7	22	2.7	0.35	34	0.27
PM -	50000	0	186	6.9	17	4.2	2.3	20	0.18
PINN II L	500	10	188	3.9	9.5	4.0	3.2E11	12	0.067
PM -	500	0	188	6.9	17	4.2	2.4	20	0.18
PM -	500	10	191	6.9	17	4.4	2.6	20	0.20
PINN III S	500	10	192	7.3	20	3.0	0.34	31	0.34
PINN I S	50000	0	194	5.2	7.7	18	83000	13	0.058
PINN II S	500	10	200	4.5	8.7	5.4	1.3E7	11	0.14
PINN I S	500	10	210	6.7	8.5	17	69000	15	0.11
PINN I L	500	0	216	5.9	7.8	11	53000	11	0.44
SH L	50000	0	224	2300	2.0E7	0.072	1.7E-3	5.4E8	1.2E-4
PINN I L	500	10	228	8.1	9.9	21	58000	13	0.19
SH L	500	0	242	7.4	5100	3.7	0.37	1.8E5	9.8E-3
MASCONS S	500	0	244	25	46	24	24	67	1.1
SH L	500	10	244	7.8	9900	3.7	0.37	2.9E5	9.8E-3
GEONET L	500	0	246	39	45	38	38	50	2.0
SH L	50000	10	249	9500	3.1E7	0.21	0.16	1.9E9	1.0E-2
GEONET L	50000	0	250	44	44	44	44	44	2.2
PINN I S	500	10	250	12	13	43	59000	20	0.56
PINN I S	500	0	253	13	14	47	51000	21	0.24
GEONET L	50000	0	260	51	52	51	51	52	2.5
GEONET L	500	10	276	85	94	84	84	95	3.3
GEONET S	500	10	281	91	95	91	91	95	3.5
GEONET S	500	0	289	95	96	95	95	97	3.6
GEONET S	50000	0	301	98	98	98	98	98	3.6
GEONET S	50000	10	309	98	98	98	98	98	3.6
ELM S	500	0	322	62	97	220	1.8E7	98	0.21
ELM L	500	0	324	59	97	240	4.9E7	97	0.45
ELM L	500	10	325	59	97	240	4.9E7	97	0.44
MASCONS S	500	10	326	87	150	84	81	210	2.7
ELM S	500	10	326	62	97	220	1.8E7	98	0.21
MASCONS L	500	10	358	1.9E11	1.2E13	5.8E9	0.25	5.1E13	10
TNN S	500	0	399	400	490	840	71000	470	1.9
TNN L	500	0	401	380	620	770	65000	610	2.2
TNN L	50000	10	405	450	730	1200	1.8E5	670	3.3
TNN S	50000	10	408	450	720	1200	1.1E5	730	3.5
TNN L	50000	0	412	450	840	1300	2.3E5	870	5.5
TNN S	50000	0	413	450	780	1300	1.2E5	770	5.5
TNN L	500	10	414	390	540	850	73000	520	6.5
TNN S	500	10	415	400	510	950	79000	500	6.8
ELM L	50000	10	417	4300	600	25000	1.9E9	520	560
ELM L	50000	0	418	4200	580	24000	1.9E9	500	670
ELM S	50000	10	419	4500	620	26000	8.3E7	530	18000
ELM S	50000	0	420	4300	600	25000	8.2E7	520	18000

Table 10 Rank values for each gravity model trained on heterogeneous Bennu gravity field data

Model	N	Error (%)	Score (Rank)	Planes (Rank)	Interior (Rank)	Exterior (Rank)	Extrap. (Rank)	Surface (Rank)	Traj. (Rank)
MASCONS L	50000	0	1	1.0	1.0	1.0	2.0	5.0	3.0
PINN III L	50000	0	2	2.0	3.0	2.0	13	4.0	2.0
PINN III S	50000	0	3	4.0	5.0	4.0	5.0	6.0	7.0
SH S	50000	0	4	6.0	9.0	6.0	4.0	9.0	5.0
SH S	50000	10	5	8.0	8.0	14	10	8.0	18
SH S	500	0	6	9.0	13	15	11	11	20
PINN II L	50000	0	7	3.0	2.0	5.0	70	1.0	12
PINN III L	500	0	8	12	29	8.0	6.0	33	6.0
PINN II S	50000	0	9	5.0	4.0	10	70	2.0	11
MASCONS S	50000	10	10	16	16	17	23	16	23
PINN III S	500	0	11	13	34	12	8.0	36	10
MASCONS S	50000	0	12	18	15	18	24	15	26
PINN II L	500	0	13	7.0	14	9.0	70	14	8.0
MASCONS L	500	0	14	10	38	3.0	1.0	70	1.0
PINN II S	500	0	15	11	11	16	70	12	9.0
SH S	500	10	16	21	27	19	22	25	22
POLY L			17	29	26	27	18	23	14
PINN II S	50000	10	18	15	7.0	21	70	7.0	27
POLY S			20	30	28	28	19	24	19
PINN I L	50000	0	20	19	6.0	20	70	3.0	30
PINN III L	50000	10	21	20	25	22	14	31	39
MASCONS L	50000	10	22	14	39	11	7.0	70	13
PINN II L	50000	10	23	17	12	23	70	10	25
PINN III S	50000	10	24	24	35	24	15	34	42
PINN I L	50000	10	25	22	10	37	70	13	28
PM -	50000	10	27	32	33	32	25	30	32
PINN III L	500	10	27	27	37	25	17	37	41
PM -	50000	0	28	33	32	33	26	29	33
PINN II L	500	10	30	23	21	31	70	19	24
PM -	500	0	30	34	31	34	27	28	34
PM -	500	10	31	35	30	35	28	27	36
PINN III S	500	10	32	36	36	26	16	35	43
PINN I S	50000	0	33	26	17	40	70	20	21
PINN II S	500	10	34	25	20	36	70	18	31
PINN I S	50000	10	35	31	19	39	70	22	29
PINN I L	500	0	36	28	18	38	70	17	45
SH L	50000	0	37	70	70	7.0	3.0	70	4.0
PINN I L	500	10	38	39	22	41	70	21	35
SH L	500	0	39	37	70	29	20	70	16
MASCONS S	500	0	41	42	42	42	29	41	48
SH L	500	10	41	38	70	30	21	70	15
GEONET L	500	0	42	43	41	43	30	39	50
SH L	50000	10	43	70	70	13	9.0	70	17
GEONET L	50000	0	45	44	40	45	31	38	52
PINN I S	500	10	45	40	23	44	70	26	47
PINN I S	500	0	46	41	24	46	70	32	40
GEONET L	50000	10	47	45	43	47	32	40	53
GEONET L	500	10	48	50	44	49	34	43	56
GEONET S	500	10	49	52	45	50	35	42	57
GEONET S	500	0	50	53	46	51	36	44	59
GEONET S	50000	0	51	54	51	52	37	47	60
GEONET S	50000	10	52	55	52	53	38	50	61
ELM S	500	0	53	48	49	70	70	48	37
ELM L	500	0	54	46	47	70	70	45	46
ELM L	500	10	55	47	48	70	70	46	44
MASCONS S	500	10	57	51	70	48	33	70	54
ELM S	500	10	57	49	50	70	70	49	38
MASCONS L	500	10	58	70	70	70	12	70	66
TNN S	500	0	59	70	70	70	70	70	49
TNN L	500	0	60	70	70	70	70	70	51
TNN L	50000	10	61	70	70	70	70	70	55
TNN S	50000	10	62	70	70	70	70	70	58
TNN L	50000	0	63	70	70	70	70	70	62
TNN S	50000	0	64	70	70	70	70	70	63
TNN L	500	10	65	70	70	70	70	70	64
TNN S	500	10	66	70	70	70	70	70	65
ELM L	50000	10	67	70	70	70	70	70	67
ELM L	50000	0	68	70	70	70	70	70	68
ELM S	50000	10	69	70	70	79	70	70	69
ELM S	50000	0	70	70	70	70	70	70	70

D.7.4 Bennu Homogeneous Density

See Tables 11 and 12.

Table 11 Metric values for each gravity model trained on homogeneous Bennu gravity field data

Model	N	Error (%)	Score	Planes (%)	Interior (%)	Exterior (%)	Extrap. (%)	Surface (%)	Traj. (km)
POLY L			8	2.9E-3	3.6E-2	1.0E-3	6.9E-5	0.21	1.1E-5
MASCONS L	50000	0	19	9.8E-3	0.34	1.5E-3	7.1E-4	3.2	5.3E-5
PINN III S	50000	0	35	0.11	1.4	3.1E-2	1.1E-2	2.8	3.7E-4
PINN III L	50000	0	35	1.3E-2	0.52	1.6E-3	2.2E-2	3.1	6.7E-5
POLY S			56	0.31	2.7	0.13	9.2E-3	3.6	1.2E-3
SH S	50000	0	57	0.18	3.4	0.061	1.1E-2	5.5	7.3E-4
SH S	500	0	83	0.30	3.4	0.18	0.12	5.4	9.0E-3
SH S	50000	10	89	0.34	3.4	0.21	0.16	5.4	1.1E-2
PINN II L	50000	0	99	0.050	0.39	0.050	1.6E17	2.6	8.8E-4
PINN II S	50000	0	101	0.11	1.0	0.11	95000	2.3	3.5E-4
PINN III L	500	0	104	0.42	17	0.073	1.6E-2	29	3.2E-4
MASCONS L	500	0	126	0.43	25	1.8E-2	5.0E-8	140	2.6E-11
PINN I L	50000	0	127	0.86	1.0	0.74	20000	2.0	5.2E-3
PINN III S	50000	10	129	1.5	3.9	1.0	2.6E-2	6.7	0.19
PINN III S	500	0	130	1.0	41	0.18	1.0E-2	50	8.7E-4
PINN II L	500	0	131	0.23	3.8	0.12	8.7E11	6.2	8.3E-4
MASCONS S	50000	10	136	1.4	5.6	1.1	1.1	10	0.084
MASCONS S	50000	0	140	1.5	5.5	1.3	1.2	10	0.094
PINN III L	50000	10	141	2.2	6.5	1.1	1.9E-2	11	0.21
PINN II S	500	0	142	0.50	3.0	0.60	7.2E6	5.1	1.6E-2
PINN II S	50000	10	149	0.78	2.3	1.2	7.5E6	4.0	4.5E-2
SH S	500	10	160	2.3	15	1.3	0.84	19	0.066
PINN II L	50000	10	168	1.2	3.5	1.6	2.1E14	4.9	0.074
MASCONS L	50000	10	169	0.79	27	0.17	0.11	160	7.7E-3
PM -	500	0	169	2.6	9.8	1.8	1.4	13	0.12
PM -	500	10	172	2.7	9.7	2.0	1.7	13	0.14
PM -	50000	10	174	2.7	9.7	2.1	1.7	13	0.14
PM -	50000	0	176	2.8	9.7	2.1	1.8	13	0.15
PINN I L	50000	10	195	3.0	3.4	5.1	6500	5.5	0.10
PINN I L	500	0	205	3.7	6.4	9.6	67000	9.3	0.070
PINN III S	500	10	209	26	18	9.4	2.2E-2	29	0.80
PINN III L	500	10	209	25	18	9.2	2.2E-2	29	0.79
PINN II L	500	10	211	4.0	9.6	3.9	2.0E9	12	0.067
PINN II S	500	10	212	3.9	9.5	4.6	7.2E6	12	0.083
SH L	500	0	217	2.5	6800	0.82	2.0E-2	1.2E5	1.7E-2
PINN I S	50000	10	220	4.4	5.2	8.9	5600	9.5	0.22
SH L	500	10	223	2.9	11000	0.82	2.0E-2	2.2E5	1.7E-2
PINN I S	50000	0	226	4.4	6.4	17	12000	11	0.21
SH L	50000	0	234	2500	1.4E7	0.061	1.1E-2	4.7E8	7.3E-4
MASCONS S	500	0	236	20	33	20	19	47	1.0
PINN I L	500	10	241	9.6	10	27	81000	15	0.13
GEONET L	500	0	244	32	40	32	32	46	1.8
GEONET L	50000	0	246	38	38	38	38	38	2.0
GEONET L	50000	10	259	59	59	59	59	59	2.7
PINN I S	500	0	260	11	12	46	62000	18	0.77
SH L	50000	10	262	8700	4.1E7	0.21	0.16	1.9E9	1.1E-2
PINN I S	500	10	264	15	18	48	74000	25	0.37
GEONET L	500	10	273	83	94	83	83	95	3.3
GEONET S	500	0	285	97	98	97	97	98	3.6
GEONET S	500	10	293	97	98	97	97	98	3.6
GEONET S	50000	10	299	98	98	98	98	98	3.6
GEONET S	50000	0	305	99	99	99	99	99	3.6
ELM L	500	0	318	59	97	240	4.9E7	98	0.43
ELM S	500	0	319	62	98	220	1.8E7	98	0.21
ELM L	500	10	320	60	97	240	5.0E7	98	0.42
MASCONS S	500	10	323	82	130	80	78	180	3.0
ELM S	500	10	324	63	98	220	1.8E7	98	0.21
MASCONS L	50000	10	363	1.9E11	1.1E13	6.1E9	0.25	5.1E13	10
TNN L	50000	10	408	450	720	360	8800	660	4.5
TNN S	50000	10	409	380	660	280	16000	650	4.7
TNN S	50000	0	410	380	690	340	34000	680	4.8
TNN L	50000	0	411	380	810	320	1.0E5	820	5.2
TNN L	500	0	413	700	560	3000	1.1E5	510	14
TNN S	500	0	414	430	490	1000	90000	490	15
TNN S	500	10	415	480	510	1200	1.1E5	520	15
TNN L	500	10	416	440	550	1100	1.0E5	550	17
ELM L	50000	0	417	4000	560	23000	1.9E9	480	36
ELM L	50000	10	418	4200	580	24000	1.9E9	490	120
ELM S	50000	0	419	4200	580	24000	8.3E7	490	18000
ELM S	50000	10	420	4300	600	25000	8.4E7	510	19000

Table 12 Rank values for each gravity model trained on homogeneous Bennu gravity field data

Model	N	Error (%)	Score (Rank)	Planes (Rank)	Interior (Rank)	Exterior (Rank)	Extrap. (Rank)	Surface (Rank)	Traj. (Rank)
POLY L			1	1.0	1.0	1.0	2.0	1.0	2.0
MASCONS L	50000	0	2	2.0	2.0	2.0	3.0	7.0	3.0
PINN III S	50000	0	4	5.0	7.0	5.0	6.0	5.0	7.0
PINN III L	50000	0	4	3.0	4.0	3.0	15	6.0	4.0
POLY S			5	10	9.0	12	4.0	8.0	13
SH S	50000	0	6	7.0	13	7.0	7.0	14	9.0
SH S	500	0	7	9.0	12	15	18	13	16
SH S	50000	10	8	11	11	17	20	12	18
PINN II L	50000	0	9	4.0	3.0	6.0	70	4.0	12
PINN II S	50000	0	10	6.0	6.0	10	70	3.0	6.0
PINN III L	500	0	11	12	33	9.0	9.0	36	5.0
MASCONS L	500	0	12	13	37	4.0	1.0	70	1.0
PINN I L	50000	0	13	17	5.0	19	70	2.0	14
PINN III S	50000	10	14	21	17	22	16	17	36
PINN III S	500	0	15	18	42	14	5.0	40	11
PINN II L	500	0	16	8.0	16	11	70	16	10
MASCONS S	50000	10	17	20	20	24	23	21	28
MASCONS S	50000	0	18	22	19	26	24	20	29
PINN III L	50000	10	19	23	23	23	10	23	39
PINN II S	500	0	20	14	10	18	70	11	19
PINN II S	50000	10	21	15	8.0	25	70	9.0	22
SH S	500	10	22	24	32	27	22	32	23
PINN II L	50000	10	23	19	15	28	70	10	26
MASCONS L	50000	10	25	16	38	13	17	70	15
PM -	500	0	25	26	29	29	25	29	31
PM -	500	10	26	27	28	30	26	28	33
PM -	50000	10	27	28	27	31	27	27	34
PM -	50000	0	28	29	26	32	28	26	35
PINN I L	50000	10	29	31	14	35	70	15	30
PINN I L	500	0	30	32	21	39	70	18	25
PINN III S	500	10	32	42	35	38	13	34	47
PINN III L	500	10	32	41	36	37	14	35	46
PINN II L	500	10	33	34	25	33	70	25	24
PINN II S	500	10	34	33	24	34	70	24	27
SH L	500	0	35	25	70	20	11	70	21
PINN I S	50000	10	36	36	18	36	70	19	41
SH L	500	10	37	30	70	21	12	70	20
PINN I S	50000	0	38	35	22	40	70	22	37
SH L	50000	0	39	70	70	8.0	8.0	70	8.0
MASCONS S	500	0	40	40	39	41	29	39	48
PINN I L	500	10	41	37	30	42	70	30	32
GEONET L	500	0	42	43	41	43	30	38	49
GEONET L	50000	0	43	44	40	44	31	37	50
GEONET L	50000	10	44	45	43	47	32	41	51
PINN I S	500	0	45	38	31	45	70	31	45
SH L	50000	10	46	70	70	16	19	70	17
PINN I S	500	10	47	39	34	46	70	33	42
GEONET L	500	10	48	51	44	49	34	42	53
GEONET S	500	0	49	52	49	50	35	45	54
GEONET S	500	10	50	53	50	51	36	48	55
GEONET S	50000	10	51	54	51	52	37	49	56
GEONET S	50000	0	52	55	52	53	38	50	57
ELM L	500	0	53	46	45	70	70	43	44
ELM S	500	0	54	48	47	70	70	46	38
ELM L	500	10	55	47	46	70	70	44	43
MASCONS S	500	10	56	50	70	48	33	70	52
ELM S	500	10	57	49	48	70	70	47	40
MASCONS L	500	10	58	70	70	70	21	70	62
TNN L	50000	10	59	70	70	70	70	70	58
TNN S	50000	10	60	70	70	70	70	70	59
TNN S	50000	0	61	70	70	70	70	70	60
TNN L	50000	0	62	70	70	70	70	70	61
TNN L	500	0	63	70	70	70	70	70	63
TNN S	500	0	64	70	70	70	70	70	64
TNN S	500	10	65	70	70	70	70	70	65
TNN L	500	10	66	70	70	70	70	70	66
ELM L	50000	0	67	70	70	70	70	70	67
ELM L	50000	10	68	70	70	70	70	70	68
ELM S	50000	0	69	70	70	79	70	70	69
ELM S	50000	10	70	70	70	70	70	70	70

Declarations

Conflicts of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Cuomo, S., Di Cola, V.S., Giampaolo, F. et al. Scientific machine learning through physics-informed neural networks: Where we are and what's next. J. Sci. Comput. **92**, 88 (2022). https://doi.org/10.1007/s10915-022-01939-z
2. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nat. Rev. Phys. **3**(6), 422–440 (2021). https://doi.org/10.1038/s42254-021-00314-5
3. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019). https://doi.org/10.1016/j.jcp.2018.10.045
4. Martin, J., Schaub, H.: Physics-informed neural networks for gravity field modeling of the Earth and Moon. Celest Mech Dyn Astron **134**, 13 (2022). https://doi.org/10.1007/s10569-022-10069-5
5. Martin, J., Schaub, H.: Physics-informed neural networks for gravity field modeling of small bodies. Celest Mech Dyn Astron **134**, 46 (2022). https://doi.org/10.1007/s10569-022-10101-8
6. Brillouin, M.: Équations aux dérivées partielles du 2e ordre. Domaines à connexion multiple. Fonctions sphériques non antipodes **4**, 173–206 (1933)
7. Kaula, W.M.: Theory of Satellite Geodesy: Applications of Satellites to Geodesy. Blaisdell Publishing Co, Waltham (1966)
8. Pavlis, N.K., Holmes, S.A., Kenyon, S.C., Schmidt, D., Trimmer, R.: A Preliminary Gravitational Model to Degree 2160. In: Jekeli, C., Bastos, L., Fernandes, J. (eds) Gravity, Geoid and Space Missions. International Association of Geodesy Symposia, vol 129. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-26932-0_4
9. Goossens, S., Lemoine, F., Sabaka, T., Nicholas, J., Mazarico, E., Rowlands, D., Loomis, B., Chinn, D., Neumann, G., Smith, D., Zuber, M.: A Global Degree and Order 1200 Model of the Lunar Gravity Field Using GRAIL Mission Data. In: 47th Annual Lunar and Planetary Science Conference. Lunar and Planetary Science Conference, pp. 1484–1484 (2016)
10. Genova, A., Goossens, S., Lemoine, F.G., Mazarico, E., Neumann, G.A., Smith, D.E., Zuber, M.T.: Seasonal and static gravity field of Mars from MGS, Mars Odyssey and MRO radio science. Icarus **272**, 228–245 (2016). https://doi.org/10.1016/j.icarus.2016.02.050
11. Hewitt, E., Hewitt, R.E.: The Gibbs-Wilbraham phenomenon: an episode in Fourier analysis. Arch. Histo. Exact Sci. **21**(2), 129–160 (1979). https://doi.org/10.1007/BF00330404
12. Martin, J.R., Schaub, H.: GPGPU Implementation of Pines' Spherical Harmonic Gravity Model. In: Wilson, R.S., Shan, J., Howell, K.C., Hoots, F.R. (eds.) AAS/AIAA Astrodynamics Specialist Conference. Univelt Inc., Virtual Event (2020)
13. Werner, R., Scheeres, D.: Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia. Celest. Mech. Dyn. Astron. **65**(3), 313–344 (1997). https://doi.org/10.1007/BF00053511
14. Scheeres, D.J., Khushalani, B., Werner, R.A.: Estimating asteroid density distributions from shape and gravity information. Planet. Space Sci. **48**(10), 965–971 (2000). https://doi.org/10.1016/s0032-0633(00)00064-7
15. Scheeres, D.J., French, A.S., Tricarico, P., Chesley, S.R., Takahashi, Y., Farnocchia, D., McMahon, J.W., Brack, D.N., Davis, A.B., Ballouz, R.L., Jawin, E.R., Rozitis, B., Emery, J.P., Ryan, A.J., Park, R.S., Rush, B.P., Mastrodemos, N., Kennedy, B.M., Bellerose, J., Lubey, D.P., Velez, D., Vaughan, A.T., Leonard, J.M., Geeraert, J., Page, B., Antreasian, P., Mazarico, E., Getzandanner, K., Rowlands, D., Moreau, M.C., Small, J., Highsmith, D.E., Goossens, S., Palmer, E.E., Weirich, J.R., Gaskell, R.W., Barnouin, O.S., Daly, M.G., Seabrook, J.A., Al Asad, M.M., Philpott, L.C., Johnson, C.L., Hartzell, C.M., Hamilton, V.E., Michel, P., Walsh, K.J., Nolan, M.C., Lauretta, D.S.: Heterogeneous mass distribution of the rubble-pile asteroid (101955) Bennu. Sci. Adv. 6(41) (2020). https://doi.org/10.1126/sciadv.abc3350

16. Muller, A.P.M., Sjogren, W.L.: Mascons: lunar mass concentrations. Science **161**(3842), 680–684 (1968)

17. Tardivel, S.: The Limits of the Mascons Approximation of the Homogeneous Polyhedron. In: AIAA/AAS Astrodynamics Specialist Conference, pp. 1–13. American Institute of Aeronautics and Astronautics, Reston, Virginia (2016). https://doi.org/10.2514/6.2016-5261

18. Wittick, P.T., Russell, R.P.: Mixed-model gravity representations for small celestial bodies using mascons and spherical harmonics. Celest. Mech. Dyn. Astron. **131**(7), 31–31 (2019). https://doi.org/10.1007/s10569-019-9904-6

19. Romain, G., Jean-Pierre, B.: Ellipsoidal harmonic expansions of the gravitational potential: theory and application. Celest. Mech. Dyn. Astron. **79**(4), 235–275 (2001). https://doi.org/10.1023/A:1017555515763

20. Takahashi, Y., Scheeres, D.J., Werner, R.A.: Surface gravity fields for asteroids and comets. J. Guid. Control Dyn. **36**(2), 362–374 (2013). https://doi.org/10.2514/1.59144

21. Takahashi, Y., Scheeres, D.J.: Small body surface gravity fields via spherical harmonic expansions. Celest. Mech. Dyn. Astron. **119**(2), 169–206 (2014). https://doi.org/10.1007/s10569-014-9552-9

22. Gao, A., Liao, W.: Efficient gravity field modeling method for small bodies based on Gaussian process regression. Acta Astronautica **157**, 73–91 (2019). https://doi.org/10.1016/j.actaastro.2018.12.020

23. Cheng, L., Wang, Z., Jiang, F.: Real-time control for fuel-optimal Moon landing based on an interactive deep reinforcement learning algorithm. Astrodynamics **3**(4), 375–386 (2019). https://doi.org/10.1007/s42064-018-0052-2

24. Furfaro, R., Barocco, R., Linares, R., Topputo, F., Reddy, V., Simo, J., Le Corre, L.: Modeling irregular small bodies gravity field via extreme learning machines and Bayesian optimization. Adv. Space Res. (2020). https://doi.org/10.1016/j.asr.2020.06.021

25. Izzo, D., Gómez, P.: Geodesy of irregular small bodies via neural density fields. Commun. Eng. **1**(1), 48 (2022). https://doi.org/10.1038/s44172-022-00050-3

26. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1–3), 489–501 (2006). https://doi.org/10.1016/j.neucom.2005.12.126

27. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15 (2014)

28. Dozat, T.: Incorporating Nesterov momentum into Adam. ICLR 2016: International Conference on Learning Representations (2016). https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ

29. Nocedal, J., Wright, S.J.: Numerical Optimization. 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5

30. Cheng, L., Wang, Z., Song, Y., Jiang, F.: Real-time optimal control for irregular asteroid landings using deep neural networks. Acta Astronautica **170**, 66–79 (2020). https://doi.org/10.1016/j.actaastro.2019.11.039

31. Baydin, A.G., Pearlmutter, B.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. J. Machine Learn. Res. **18**, 1–43 (2018)

32. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: representing scenes as neural radiance fields for view synthesis. Commun. ACM **65**(1), 99–106 (2022). https://doi.org/10.1145/3503250

33. Pavlis, N.K., Holmes, S.A., Kenyon, S.C., Factor, J.K.: The development and evaluation of the earth gravitational model 2008 (EGM2008). J. Geophys. Res. Solid Earth (2012). https://doi.org/10.1029/2011JB008916

34. Gaskell, R.W.: Gaskell Eros Shape Model V1.1. (2021). https://doi.org/10.26033/d0gq-9427

35. Leonard, J.M., Nievinski, F.G., Born, G.H.: Gravity error compensation using second-order gauss-markov processes. J. Spacecraft Rockets **50**(1), 217–229 (2013). https://doi.org/10.2514/1.A32262

36. Martin, J.R., Schaub, H.: Preliminary Analysis of Small-Body Gravity Field Estimation using Physics-Informed Neural Networks and Kalman Filters. 73rd International Astronautical Congress, Paris, France, IAC-22,C1,8,2,x70686 (2022)

37. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)

38. Zhu, Q., Liu, Z., Yan, J.: Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks. Comput. Mech. **67**, 619–635 (2021). https://doi.org/10.1007/s00466-020-01952-9

39. Takahashi, Y., Scheeres, D.J.: Morphology driven density distribution estimation for small bodies. Icarus **233**, 179–193 (2014). https://doi.org/10.1016/j.icarus.2014.02.004

40. Kanamaru, M., Sasaki, S.: Estimation of Interior Density Distribution for Small Bodies: The Case of Asteroid Itokawa. Trans. JSASS Aerospace Tech. Japan **17**(3), 270–275 (2019). https://doi.org/10.2322/tastj.17.270

41. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. **9**, 249–256 (2010)

42. Hendrycks, D., Gimpel, K.: Gaussian Error Linear Units (GELUs). arXiv e-prints arXiv:1606.08415 [cs.LG] (2023). https://doi.org/10.48550/arXiv.1606.08415