Celestial Mechanics and Dynamical Astronomy (2022) 134:13
https://doi.org/10.1007/s10569-022-10069-5

ORIGINAL ARTICLE

Physics-informed neural networks for gravity field modeling of the Earth and Moon

John Martin[1] · **Hanspeter Schaub**[1]

Received: 22 June 2021 / Revised: 19 February 2022 / Accepted: 23 February 2022

Abstract
High-fidelity representations of the gravity field underlie all applications in astrodynamics. Traditionally these gravity models are constructed analytically through a potential function represented in spherical harmonics, mascons, or polyhedrons. Such representations are often convenient for theory, but they come with unique disadvantages in application. Broadly speaking, analytic gravity models are often not compact, requiring thousands or millions of parameters to adequately model high-order features in the environment. In some cases these analytic models can also be operationally limiting—diverging near the surface of a body or requiring assumptions about its mass distribution or density profile. Moreover, these representations can be expensive to regress, requiring large amounts of carefully distributed data which may not be readily available in new environments. To combat these challenges, this paper aims to shift the discussion of gravity field modeling away from purely analytic formulations and toward machine learning representations. Within the past decade there have been substantial advances in the field of deep learning which help bypass some of the limitations inherent to the existing analytic gravity models. Specifically, this paper investigates the use of physics-informed neural networks (PINNs) to represent the gravitational potential of two planetary bodies—the Earth and Moon. PINNs combine the flexibility of deep learning models with centuries of analytic insight to learn new basis functions that are uniquely suited to represent these complex environments. The results show that the learned basis set generated by the PINN gravity model can offer advantages over its analytic counterparts in model compactness and computational efficiency.

Keywords Gravity field modeling · Machine learning · Physics-informed neural network · Spherical harmonics · Planetary bodies · Astrodynamics

✉ John Martin
john.r.martin@colorado.edu

[1] Ann and H. J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Dr., Boulder, CO 80303, USA

Published online: 24 March 2022

1 Introduction

One of the earliest high-fidelity representations of Earth's gravity field was formed using the spherical harmonic basis (Kaula 1966). This representation is particularly efficient at capturing Earth's largest gravitational perturbation—planetary oblateness or J_2—making it the de facto standard within the astrodynamics community. The popularity of the spherical harmonic representation has helped motivate multiple missions like GRACE and GRACE-FO to resolve increasingly high-degree spherical harmonic estimates of the Earth's potential (Tapley 2008). Similar missions like GRAIL sought to accomplish the same for the Moon (Lemoine et al. 2014). As of the time of writing, the highest fidelity spherical harmonic representation of the Earth is the 2008 Earth Gravitational Model (EGM-2008) which extends Earth's static gravity field to an impressive degree 2, 159—a model containing over 4 million parameters (Pavlis et al. 2012). Likewise, the highest fidelity static gravity field for the Moon reaches degree and order 1200 or approximately 1.5 million parameters (Goossens et al. 2016).

While spherical harmonic gravity models are popular for many spacecraft operations, they can become unreliable in certain circumstances. Specifically, the spherical harmonic gravity model assumes that spacecraft operations occur outside of a sphere which circumscribes all mass elements of the body: the Brillouin sphere. If a spacecraft enters the Brillouin sphere, the representation of the potential begins to diverge which can generate numerical difficulties when simulating trajectories near the surface of a body—a problem especially apparent in small-body exploration (Werner and Scheeres 1997). Seeing as many asteroids are irregularly shaped, the majority of an asteroid's surface may exist within the Brillouin sphere which makes landing or touch-and-go mission phases particularly risky when using this representation. The ellipsoidal harmonic gravity model presents a slightly more amenable solution for this problem which allows for a tighter circumscribing ellipsoid about the body, but it also remains prone to the same diverging representation once inside of the ellipsoid (Romain and Jean-Pierre 2001).

Alternative gravity models exist that attempt to combat this representation divergence (Fig. 1). One such alternative is to use mascons to represent the potential (Koch and Morrison 1970). By distributing discrete mass elements across the surface of the body and making use of the linearity of the Laplace operator, a potential function can be approximated as the summation of the potentials generated by distributed point-mass elements. This approach yields stable dynamics inside the circumscribing sphere; however, the representation generates inconsistent dynamics near the surface where the discrete nature of the mascons becomes increasingly apparent (Tardivel 2016). In 1996, Werner and Scheeres introduced the polyhedral gravity model to solve this problem. The polyhedral model forms a representation of the potential directly from a shape model of the body. The representation can generate stable dynamics all the way down to the surface of the body; however, it comes at the cost of assuming a density profile for the object which is often difficult to uniquely estimate (Takahashi and Scheeres 2014). In addition, both the mascon and polyhedral models come with a large computational cost if left unparallelized which can be limiting both in simulation and on-board spacecraft (Russell and Arora 2012; Wittick and Russell 2019).

Beyond the divergence of the representation inside of the Brillouin sphere, spherical harmonics also suffers from the same ailments as other periodic bases. Just as Fourier series struggle to converge over discontinuities in 1D (as best demonstrated through Gibbs phenomenon), so too do spherical harmonics struggle to capture discontinuity in three dimensions (Gottlieb and Shu 1997; Hewitt and Hewitt 1979). This becomes problematic when modeling

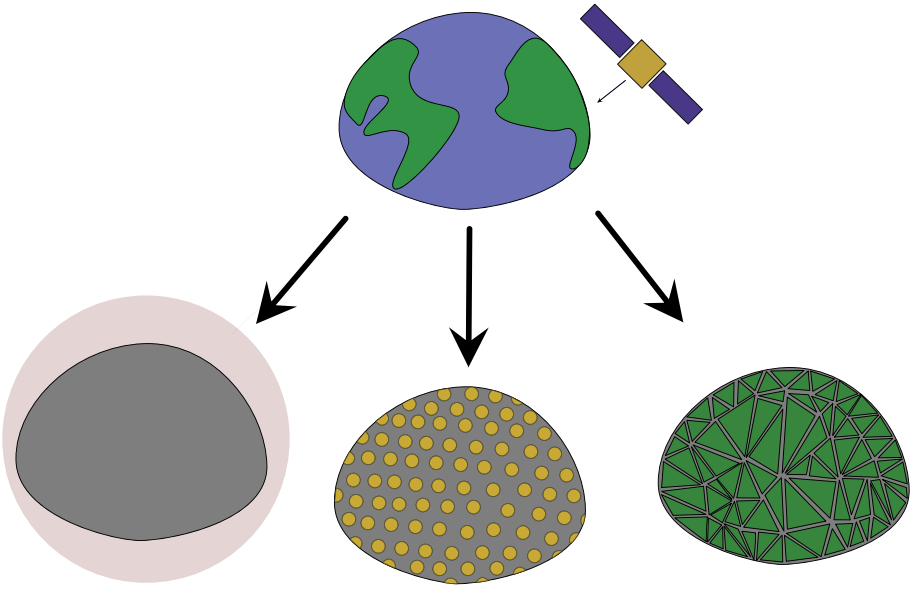


Fig. 1 Popular gravity modeling options include spherical harmonics (left), mascons (center), or polyhedral (right) representations of the potential

the Earth's gravity field where the gravitational perturbations aside from J_2 are generated from discontinuous features—e.g., mountain ranges, tectonic subduction zones, hotspots, etc. Consequently spherical harmonic gravity models can require tens-of-thousands of coefficients/parameters to begin representing these features and hundreds-of-thousands more to suppress the oscillations around them.

The size of spherical harmonic gravity models can thereby grow to be a challenge as well—particularly in regard to efficient computation. The algorithm required to evaluate a spherical harmonic model and produce a corresponding acceleration scales as $\mathcal{O}(n^2)$ where n is proportional to the largest degree in the model. Such scaling makes the highest fidelity spherical harmonic models prohibitively expensive to evaluate. In addition, spherical harmonics are one of the only gravity models that cannot be easily parallelized. The inherent recursion of the associated Legendre polynomials makes it particularly challenging to evaluate these harmonics on a GPU or with a multiprocessed algorithm (Martin and Schaub 2020). The spherical harmonic computational constraints therefore leave dynamicists to trade between computational efficiency and accuracy.

This research aims to directly address each of these challenges. While the aforementioned gravity field representations are powerful in certain conditions, they all come with limitations or assumptions as a result of the analytics. Ideally there can exist a representation that is flexible enough to capture the most dominant perturbations, robust enough to work across all operational conditions without making assumptions, and efficient enough to compute quickly both on-board spacecraft and in simulation. This research posits that machine learning and artificial neural networks are capable of navigating through each of these constraints.

Machine learning is a field markedly successful at regressing accurate models from data observed in complex environments, typically through the use of artificial neural networks (LeCun et al. 2015). These neural networks are trained to learn an optimal mapping between some desired input variable and output variable, while making no inherent assumptions

about the problem formulation. This is immediately advantageous for the gravity modeling problem, as the prescribed analytics are precisely what cause divergence of the spherical harmonic representation within the circumscribing sphere, or require prior knowledge about the shape and density of the body for the polyhedral model. By training an artificial neural network to learn the mapping between position and acceleration from the data alone, there is not an a priori expectation of how the gravitational potential must be represented. Instead the network's only goal is to resolve a basis that is maximally efficient at mapping between these two spaces.

While the focus of this paper is on the use of neural networks to construct such bases, there exist other approaches to discovering convenient representations of unknown problem dynamics. Recently Manzi and Vasile (2020) introduced deep symbolic regression which incrementally constructs a tree of elementary functions which can be used to represent unknown dynamics using a combination of sparse regression and symbolic regression. Their approach offers a compelling method to reconstructing unmodeled dynamics particularly when those dynamics are sufficiently well behaved such that they can be reconstructed using a relatively sparse tree. In the case of more complex dynamics with frequent discontinuity and highly nonlinear behavior, such approach may suffer as the tree would need sufficient depth to adequately capture these ill-behaved regions. Neural networks bypass these challenges by effectively preconstructing the tree. This approach eliminates the chance of a minimal and analytically expressible set of basis functions, but instead offers greater modeling flexibility capable of quickly capturing discontinuous behaviors. As will be discussed later, gravity fields can have highly discontinuous features which is why neural networks are the primary machine learning model studied in this paper.

Using machine learning models to represent gravity fields has been demonstrated with some level of success in the literature (Cheng et al. 2020; Gao and Liao 2019; Furfaro et al. 2021). These works center on small-body applications using neural networks, extreme learning machines, and Gaussian processes to predict gravitational accelerations produced by these bodies in lieu of the computationally expensive polyhedral representation. The literature shows that these alternative gravity models are often efficient to evaluate, but the analyses do not fully compare the advantages and disadvantages of these representations with their analytic predecessors. For instance, it is unclear what data conditions are required to resolve an accurate representation of the gravity field. Are these models capable of generating a more efficient basis set than their analytic counterparts? Under what conditions do these models begin to overfit or diverge? Are these representations universally convenient for all planetary bodies? This paper aims to address these questions and broaden the discussion about the robustness of the neural network gravity model particularly applied to large celestial bodies. While similar questions exist for modeling small-body gravity fields, such studies are purposefully left to a future article as there exist sufficiently different analyses necessary to characterize the robustness of the model in these more exotic environments.

In addition to studying these questions, this paper investigates how network performance can be improved by adding physics constraints into the training process. Specifically, artificial neural networks are powerful tools capable of solving high-dimensional problems; however, their flexibility often comes at the expense of interpretability. Because machine learning models are data driven, there are no explicit analytic insights embedded into the learning process—that is until recently. In 2019, Raissi et. al. introduced the physics-informed neural network (PINN) which injects underlying physics equations into the cost function of a traditional neural network (Raissi et al. 2019). Using automatic differentiation, the PINNs are trained not only to prioritize an accurate mapping from an input space to an output space, but also to enforce that the solution/function resolved by the network satisfies pertinent dif-

ferential equations and boundary conditions. PINNs are therefore able to unify the flexibility of neural networks with centuries of analytic insights derived from physics. This makes the PINN a natural candidate to apply to the gravity modeling problem. Rather than forcing a solution to be of a preconstructed analytic form (e.g., spherical harmonics), a PINN can learn a convenient basis to represent the problem and simply ensure that the final function represented by the network satisfies the equations of Newton and Laplace.

The PINN gravity models in this paper are trained to represent the gravity fields of two representative celestial bodies: the Earth and Moon. Both bodies are commonly modeled using high-degree spherical harmonic representations, but their gravitational perturbations are appreciably different. When looking beyond the oblateness term, Earth's gravitational perturbations are relatively sparse and infrequent—formed by large geologic discontinuities at tectonic boundaries and near mountain ranges. In contrast, the Moon's perturbations beyond degree and order 2 form an almost random surface topology due to the excessive cratering and mascons. Contrasting the performance of the neural network gravity model on these large bodies provides a more complete picture of the conditions necessary for the networks to outperform their analytic predecessor.

The paper is divided into the following sections: Sect. 2 discusses the current spherical harmonic model used for the Earth and investigates the performance of this model at varying levels of fidelity. These results serve as the baseline metrics to which the traditional and physics-informed neural network gravity models are compared. Following this characterization, Sect. 3 provides a discussion of the neural networks used for this paper along with information about the data and hyperparameters used to train them. The paper then compares the performance of the neural network gravity models to that of the spherical harmonic representation focusing specifically on arguments of representational compactness (Sect. 4), generalization ability (Sect. 5), and evaluation speed (Sect. 6). The analyses are then repeated for the Moon in Sect. 7, before finally concluding in Sect. 8.

2 Spherical Harmonic Representation

The spherical harmonic expansion of the gravitational potential, U, is defined as:

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

where r is the distance to a point defined with respect to the gravitational body's center-of-mass, μ is the gravitational parameter of the body, R is the circumscribing radius of the body, l is the degree of the spherical harmonic model, m is the order of the spherical harmonic model, $C_{l,m}$ and $S_{l,m}$ are the regressed spherical harmonic coefficients, λ is the longitude, ϕ is the latitude, and $P_{l,m}$ are the associated Legendre polynomials (Kaula 1966).

This representation of the potential is particularly advantageous when applied to near-spherical celestial bodies, as it is captures one of the most common and prominent gravitationally perturbing features (planetary oblateness) with the first nonzero term in the expansion beyond the point mass. As such, spherical harmonics remains a strong candidate solution to represent planetary gravity fields to 1st order. This initial efficiency, however, does not persist at higher degrees.

To demonstrate, consider the prominent gravitational features that remain after removing the point mass and planetary oblateness and obliquity accelerations (i.e., the features beyond degree and order 2). These features will henceforth be referred to as *perturbations*. To view these perturbations, one can define:

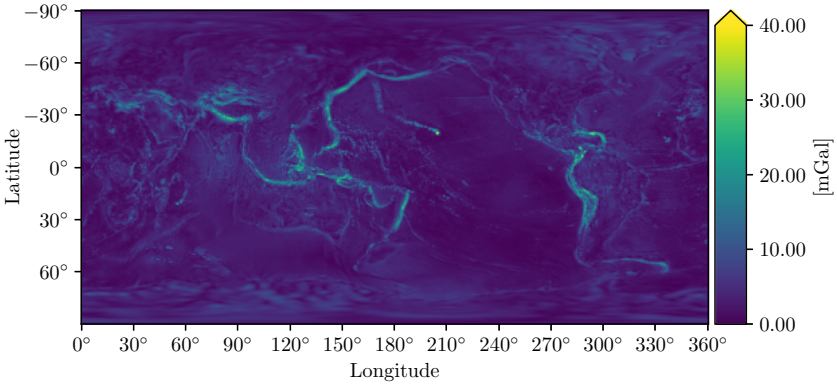


Fig. 2 Map of δa at the Earth's Brillouin sphere

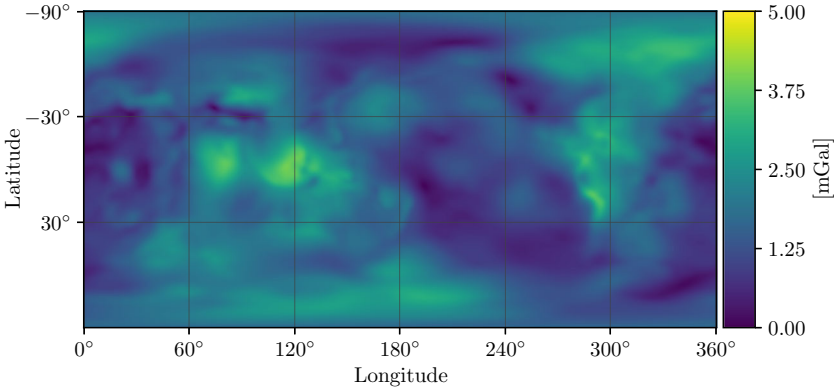


Fig. 3 Map of δa of the Earth at a LEO altitude (approximately 420 km)

$$\delta a(\boldsymbol{r}) = \left| -\nabla U^{\mathrm{SH}}_{\mathrm{Truth}}(\boldsymbol{r}) - \left(-\nabla U^{\mathrm{SH}}_{2}(\boldsymbol{r}) \right) \right|, \tag{2}$$

where $U^{\mathrm{SH}}_{\mathrm{Truth}}$ is constructed using EGM-2008 expanded to degree $l = 1,000$, and the gradient of the potential is taken using Pines' algorithm to avoid singularities at the poles (Pines 1973; Martin and Schaub 2020). Figure 2 shows Eq. (2) applied across the Brillouin sphere of Earth and verifies that the perturbations are typically discontinuous features in the crust like mountain ranges such as the Himalayas and Andes, tectonic subduction zones as best seen in the Pacific, and hotspots scattered across the globe. Such findings are intuitive as the accelerations are directly proportional to the gradient of the potential ($\boldsymbol{a} = -\nabla U$) and large displacements in landmass generate large changes in the potential.

The perturbations shown in Fig. 2 are important signals to capture with a gravity model, so it is reasonable to ask how efficiently do spherical harmonic models represent these perturbations? If a lower-degree model is used, how much error remains in these important parts of the gravity field? To investigate, a mean root-squared error (MRSE) metric is introduced:

$$\mathrm{MRSE}(\mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N_f} \delta a_p(\boldsymbol{r}_i) \quad \boldsymbol{r_i} \in \mathcal{A}, \tag{3}$$

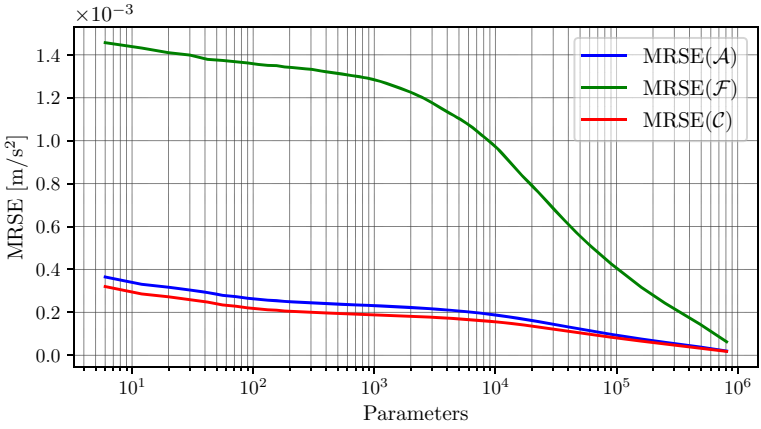


Fig. 4 Plot of MRSE($\mathcal{A}$), MRSE($\mathcal{F}$), and MRSE($\mathcal{C}$) as a function of total parameters, p, used in the spherical harmonic gravity model where $p = l(l+1)$

where $\mathcal{A}$ is the set of positions for which the gravity field is evaluated, i.e., field points, N_f is the total number of field points in set $\mathcal{A}$, and δa_p is a generalization of Eq. (2) such that:

$$\delta a_p(\boldsymbol{r_i}) = |-\nabla U^{\text{SH}}_{\text{Truth}}(\boldsymbol{r}_i) + \nabla U^{\text{model}}_p(\boldsymbol{r}_i)|, \tag{4}$$

where p represents the maximum number of parameters/coefficients used in the gravity field model being evaluated.

The MRSE metric of Eq. (4) is applied to three sets: The first set includes $N_f = 250,000$ field points distributed in a Fibonacci grid at the Brillouin sphere of Earth ($\mathcal{A} : \{\boldsymbol{r}_i \quad \forall i < N_f\}$). The Fibonacci grid is chosen to ensure a near-isotropic distribution of data about the Brillouin sphere, thereby avoiding the clustering of data at the poles where the set to be distributed uniformly in latitude and longitude (Swinbank and Purser 2006). The second set, $\mathcal{F}$, is a subset of $\mathcal{A}$ that only includes the prominent perturbations. Specifically, $\mathcal{F}$ is generated by selecting the field points within $\mathcal{A}$ whose acceleration exceeds 2 standard deviation of the mean acceleration of $\mathcal{A}$, $\delta\bar{a}$, such that $\mathcal{F} : \{|\delta a(\boldsymbol{r}_i) - \delta\bar{a}| > 2\sigma_{\boldsymbol{a}}(\mathcal{A})\}$. The third and final set, $\mathcal{C}$, is the compliment of set $\mathcal{F}$ ($\mathcal{C} : \mathcal{A}/\mathcal{F}$) representing the "background" of the gravity field. The MRSE metric is applied to sets $\mathcal{A}$, $\mathcal{F}$ and $\mathcal{C}$ and presented in Fig. 4 as a function of total parameters used in the spherical harmonic representation.

Figure 4 quantifies the relationship between spherical harmonic model size and model accuracy. While MRSE($\mathcal{A}$) and MRSE($\mathcal{C}$) have relatively low error independent of the number of spherical harmonic coefficients, the same cannot be said about MRSE($\mathcal{F}$). When the spherical harmonic model includes less than 10,000 coefficients (degree $l = 110$), MRSE($\mathcal{F}$) has significantly higher error than MRSE($\mathcal{A}$) implying that low-degree spherical harmonics models struggle to efficiently capture the prominent perturbations. Only after a spherical harmonics model exceeds degree $l = 110$ do these perturbations begin to get represented accurately. The modeling inefficiency of the low-degree spherical harmonic models suggests that the majority of these features can only be represented with harmonics wavelengths smaller than those generated by an $l = 110$ spherical harmonic field. To a dynamicist, this implies that the next most important perturbations beyond J_2 require, at the minimum, a spherical harmonic model that exceeds degree $l = 110$ if they aspire to incorporate the dynamic effects of the Earth's high-order perturbations into their application.

The discrepancy between the modeling error of $\mathcal{A}$ and $\mathcal{F}$ showcases how the spherical harmonic representation struggles to capture perturbations in order of dynamical significance. While the initial efficiency of representing Earth's oblateness is undeniable, the convenience does not extend into the next most important perturbations. This is because spherical harmonics prioritize fitting prescribed geometries onto a system in which those geometries are not naturally present. As a consequence, the spherical harmonic model must superimpose many high-order frequencies/harmonics before capturing these perturbations.

In some circumstances this may not be a problem. When a sufficiently high-fidelity model exists and the researcher is not computationally limited, spherical harmonics will eventually converge even over the discontinuous features. On-board spacecraft, however, computational resources may be limited or a high-fidelity spherical harmonic model may not exist for the body in question. In these conditions, operations over short timescales and near large surface features could be negatively affected by spherical harmonics inability to efficiently represent these perturbations.

The results shown in Fig. 4 motivate why this research turns to learned neural network gravity representations as an alternative to spherical harmonics. The spherical harmonics basis is inherently limited in resolving discontinuous perturbations while retaining a compact model size. The perturbations present on the Earth require small-wavelength harmonics which are only present in high-degree expansions. Neural networks, in contrast, do not have prescribed basis functions and corresponding characteristic wavelengths. There is no inherent minimum of 10,000 parameters needed to represent a specific mountain range or other discontinuity. Rather, the neural networks learn a convenient basis that represents the most important perturbations of the field independent of their geometry or scale. In principle, neural network gravity models can therefore yield more compact representations that achieve comparable, if not greater, accuracy than traditional spherical harmonics.

3 Machine learning representations

3.1 Traditional neural network

Artificial neural networks are a series of learned, nonlinear transformations that map data from an input space to a desired output space by minimizing a prescribed loss function such as mean squared error:

$$\mathcal{J}(\boldsymbol{\Theta}) = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| (y_i - \hat{y}(x_i|\boldsymbol{\Theta}))^2 \right|, \tag{5}$$

where y_i is the true output, $\hat{y}_i(x_i|\boldsymbol{\Theta})$ is predicted output by the artificial neural network given the vector of trainable parameters $\boldsymbol{\Theta}$ which includes the weights, $\mathbf{w}$, and biases, $\mathbf{b}$, of the network, and N_f is the total number of points used to train the network.

The networks for this paper are constructed as a series of densely connected hidden layers with N nodes per layer:

$$h_i^{(k)} = \sigma\left(w_{ij}^{(k-1)} h_j^{(k-1)} + b_i\right) \quad k \in \{1, \ldots, k_{\max}\}, \tag{6}$$

where $h^{(k)}$ is the k-th hidden layer, i is the node in the layer, w_{ij} are the weights connecting the hidden layers, b_i are the biases attached to the nodes in the layer, and σ is the nonlinear

transformation (typically sigmoid, hyperbolic tangent, or rectified linear unit). Note that $h^{(0)} = x$, and $h^{(k_{\max})} = \hat{y}$.

The neural network is trained by iteratively updating the weights and biases to minimize Eq. (5) such that:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\boldsymbol{\Theta}}(\mathcal{J}(\mathbf{w})); \quad \mathbf{b}^* = \arg\min_{\mathbf{b}\in\boldsymbol{\Theta}}(\mathcal{J}(\mathbf{b})), \tag{7}$$

which can be solved using a gradient descent algorithm like Adam or SGD (Kingma and Ba 2014; Bottou 2012):

$$\boldsymbol{\Theta}^{m+1} = \boldsymbol{\Theta}^m - \eta\nabla_{\boldsymbol{\Theta}^m}\mathcal{J}^m(\boldsymbol{\Theta}), \tag{8}$$

where η is the learning rate and m is the training iteration.

3.2 Physics-informed neural networks

One of the disadvantages of using traditional neural networks to represent a function found in physics is that the learned representation may not inherently satisfy the fundamental properties of said function. For example, in the gravity field modeling problem, physics implies that gravitational accelerations are really by-products of a more fundamental scalar potential:

$$\mathbf{a} = -\nabla U, \tag{9}$$

In addition that scalar potential must be a solution to Laplace's equation for all field points that exist outside of the body:

$$\nabla^2 U = 0, \tag{10}$$

Eq. (10) is precisely why spherical harmonics is a popular basis to represent the potential—spherical harmonics are one of the general solutions to Laplace's equation (Courant and Hilbert 1989).

Traditional neural networks are not trained with these physics properties in mind. Instead they prioritize predicting an accurate acceleration from a position vector, irrespective of the more fundamental properties. In this sense, the network will be trained agnostic to the fact that the gravity field it represents produces conservative forces, and the underlying potential must be sufficiently smooth and continuous for sensible dynamics.

In 2019, Raissi et. al. recognized this problem and suggested that neural networks models do not need to be agnostic of the physics which govern the function they are attempting to represent (Raissi et al. 2019). Instead, networks can be trained specifically to ensure that learned representations obey some underlying differential equations. To this end, Raissi et. al. introduced the physics-informed neural network (PINN). PINNs inject differential equations into the cost function of a traditional neural network and use automatic differentiation to ensure that these equations are respected by the function learned by the network.

For example, consider the following arbitrary differential equation:

$$f''(x) + f'(x) + f(x) = 0, \tag{11}$$

Assume there exist measurements of x and the corresponding values of $f(x)$. A traditional neural network can use these observations as training data to learn a mapping from $x \rightarrow \hat{f}(x|\boldsymbol{\Theta})$ by minimizing the cost function $J(x|\boldsymbol{\Theta}) = |f(x) - \hat{f}(x|\boldsymbol{\Theta})|^2$. The risk of training the network with this particular cost is that the network does not know that the mapping,

$\hat{f}(x|\boldsymbol{\Theta})$, must also satisfy Eq. (11). PINNs change this paradigm by inserting the original differential equation into the cost function:

$$J(\boldsymbol{\Theta}) = \frac{1}{N_f}\left(\sum_{i=1}^{N_f}\left|f(x_i) - \hat{f}(x_i|\boldsymbol{\Theta})\right|^2 + \left|\hat{f}''(x_i|\boldsymbol{\Theta}) + \hat{f}'(x_i|\boldsymbol{\Theta}) + \hat{f}(x_i|\boldsymbol{\Theta})\right|^2\right), \quad (12)$$

where the derivatives of the network $\hat{f}(x_i|\boldsymbol{\Theta})$ are taken with automatic differentiation. This cost function, while similar to Eq. (5), not only penalizes erroneous values of $\hat{f}(x|\boldsymbol{\Theta})$, but also penalizes when the learned function violates the differential form of the problem. This extra term serves as a form of regularization in the training process which can lead to improved solutions that conveniently also satisfy important physics properties.

Applying this formulation to the gravity modeling problem yields the following cost function:

$$J(\boldsymbol{\Theta}) = \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\mathbf{a}_i + \overset{\text{AD}}{\nabla}\hat{U}(r_i|\boldsymbol{\Theta})\right|^2, \quad (13)$$

where $\mathbf{a}_i$ is the measured acceleration at position $\mathbf{r}_i$, and $\hat{U}$ is the learned potential function. Note how the neural network is not directly representing the acceleration. Instead, it is learning a model of the scalar potential, $\hat{U}$, and enforcing that the negative gradient of the learned potential (taken with automatic differentiation) matches the measured acceleration. In this sense, the network is providing a model capable of generating accelerations, but doing so in a way which requires that those accelerations are a function of a more fundamental scalar potential.

A few important notes on applying physics constraints to neural network gravity models: First, $\boldsymbol{a} = -\nabla U$ is the only physics-informed constraint applied to the PINNs trained in this paper though additional constraints can be used. For example, the cost function could also include a penalty for (i) violating the boundary condition $U(\mathbf{r}) = 0$ as $|r| \to \infty$, (ii) for mis-modeling the potential function itself, $U(r) - \hat{U}(r|\boldsymbol{\Theta}) = 0$ (this assumes an accurate representation of the potential already exists), and (iii) not satisfying conservative vector field properties like $\nabla^2\hat{U} = 0$ and $\nabla \times \hat{U} = 0$.

This paper purposefully omits these additional physics-informed constraints for two reasons. In regard to omitting optional constraint (ii), the domain of the potential is order-of-magnitudes larger than the domain of the corresponding accelerations. This discrepancy can be disruptive to training neural networks as machine learning models prefer data that is normalized (via min-max transform, normal transform, or other). This normalization serves to avoid excessively large gradients during backpropagation and to minimize the effect of round-off error. Given that most machine learning frameworks train with 32-bit floats rather than 64-bit for rapid evaluation, calculations with particularly large or small numbers can stunt efficient and accurate network training. These scaling issues can likely be addressed with proper nondimensionalization of the physics equations—however this is left to future work. This choice to exclude the potential in the cost function also conveniently reflects the more common data circumstances—i.e., a high-fidelity potential of the body in question does not exist.

The reason optional constraints (i) and (iii) are not applied is because, in the authors' experience, they tend to drive the PINN solution for $\hat{U}$ toward zero or slow convergence. This is attributed to the multi-objective optimization that gradient descent algorithms struggle to accommodate (Mertikopoulos et al. 2018). Explicitly, the constraints of a conservative vector field can be satisfied if the network identifies $U = 0$ as the optimal solution. This can force

the network away from the nonzero solution guided by the constraint $\boldsymbol{a} + \nabla U = 0$. In theory these competing objectives can be better balanced through dynamically adjusting learning rates for each of the objective costs as demonstrated in Wang et al. (2020) or by using extreme learning machines which use least-squares rather than gradient descent to solve for network weights (Huang et al. 2006). It is also possible that by modifying the network structure such constraints may perform better. For the sake of brevity, these studies are left for future work.

Another observation of the physics-informed constraint applied to the gravity field modeling problem is the inherent efficacy of training the network to represent the potential rather than accelerations directly. When training a traditional neural network to learn the mapping from position to acceleration, the three components of the acceleration vector are effectively treated as distinct and individual features for the network to learn. In fact, these components are, by construction, orthogonal such that there is no observable relationship between them. As such, a traditional neural network must learn a basis set that can solve three distinct problems simultaneously.

This is an inefficient construction. By instead training the network to represent the scalar potential, the network only has to learn a basis to represent one feature. In this sense, a single training datum uses all three components of the acceleration vector to constrain just the potential, rather than distributing that knowledge between three learned features. As a consequence, PINNs can make more efficient use of the same amount of training data—decreasing training times while increasing sample efficiency.

Finally, it is worth stressing that the PINN gravity model does not require that a preexisting analytic gravity model exists from which training data can be generated. PINNs can be trained in situ, relying on estimates of accelerations either via finite differencing of relative velocities or more advanced filtering techniques. This paper chooses to focus primarily on the gravity modeling problem, not the gravity estimation problem. As such, a high-fidelity spherical harmonic model is used in the following studies to generate accurate and representative training data. Future work will investigate how these PINNs can be trained online to regress a field in situ, without a pre-existing gravity model.

3.3 Networks and hyperparameters

The networks tested for the remainder of the paper are divided equally between traditional neural networks and physics-informed neural networks. Each of these networks is densely connected with N nodes per layer and eight hidden layers. The choice of eight hidden layers offered a desirable balance between network capacity and reasonable training times. All networks share near-identical hyperparameters as expressed in Table 1.

For this study, the hyperparameters that have the largest effect on performance are the learning rate and mini-batch size. If the batch size is too small, the gradient descent algorithm can move in directions other than the local minimum resulting in longer training times. Unfortunately small batches are often unavoidable when the available GPU does not have sufficient VRAM to store the entire dataset on the device. In this case, smaller and more cautious learning rates should be used to ensure the optimizer does not quickly lead the network to a suboptimal minimum that later becomes challenging to escape. Conversely, if a large batch size can be used, larger learning rates are encouraged as they yield shorter training durations that are more likely to descend in the direction of the true gradient of the cost function.

Independent of initial learning rate magnitude, it can be advantageous to slowly decrease the learning rate toward the end of training to prevent the weights from oscillating above the

cost function's local minimum (Ruder 2016). For this study, an exponential decay is applied to the learning rate:

$$\eta_i = \begin{cases} \eta_0 & i < i_0 \\ \eta_0 * \text{pow}\left(\alpha, -\frac{i-i_0}{\sigma}\right) & i \geq i_0 \end{cases}, \tag{14}$$

where η_i is the learning rate at epoch i, i_0 is the reference epoch after which the decay begins, σ is the scale factor, α is the decay rate, and η_0 is the initial learning rate.

The other critical hyperparameter is the activation function. When training a physics-informed network, it is important that the activation function selected has a sufficiently high order of continuity. If the activation function does not have smooth high-order derivatives and if gradients of the network are taken using automatic differentiation to enforce the physics constraints, the cost function will no longer be well-behaved for gradient descent. As such, readers are encouraged to avoid using the popular rectified linear unit (ReLU) or leaky ReLU and instead opt for functions with infinite orders of continuity like hyperbolic tangent or the Gaussian exponential linear unit (GELU).

Each network is trained for 100,000 epochs using the Adam optimizer in TensorFlow 2.4[1] on a Nvidia RTX 2060 graphics card. The decision of setting the training length to 100,000 epochs was based on preliminary results which for 100,000 epochs was the maximum training length needed for the validation loss to plateau. This hyperparameter was kept fixed for all networks to minimize the number of confounding variables. All network weights are initialized according to the definition provided in Glorot and Bengio (2010). The traditional neural networks preprocess the input data using a min-max transformation fit to each component of $\boldsymbol{r}$. The outputs of the network are the acceleration vectors at the corresponding field points with the point mass and degree and order 2 contribution of the potential removed. The accelerations that remain are then also preprocessed using a min-max transformation on each component.

The PINNs preprocess the training data slightly differently. Only $\boldsymbol{r}$ is preprocessed using the min-max transformation along each component. The outputs, $\boldsymbol{a}$, also have the point mass and degree and order 2 contributions removed, but they are instead preprocessed using a uniform min-max transformation such that the data are scaled by the minimum and maximum across all components rather than each component individually. This guarantees that the learned potential function is also scaled in a manner consistent with the accelerations. Together the hyperparameters shared across all networks are expressed in Table 1, and the unique hyperparameters for each network are listed in Table 2.

4 Representational compactness

The first analysis comparing spherical harmonics with the neural network gravity model seeks to identify how many free parameters, p, are necessary to obtain a certain level of accuracy for the Earth's gravity field. Free parameters refer to the coefficients used in a spherical harmonic model or the trainable weights and biases of a network model. This analysis assumes that a sufficiently high-fidelity gravity field representation already exists (spherical harmonic, polyhedral, mascon, or otherwise) to provide perfect acceleration training data for the network, though this is not necessarily a requirement as discussed in Sect. 3.2.

[1] https://www.tensorflow.org/.

Table 1 Shared hyperparameters for the traditional and physics-informed neural networks trained in this paper

Hyperparameter	Value
η_0	0.005
i_0	25,000
α	0.5
σ	25,000
Optimizer	Adam
Initializer	Glorot uniform
Epochs	100,000
$\boldsymbol{x}$ transform	MinMax
Activation	GELU
Number of layers	8

Table 2 Unique hyperparameters for the traditional and physics-informed neural networks trained in this paper

Network type	Nodes per layer (N)	Model parameters	Batch size	$\mathbf{a}$ Transform
Traditional	10	843	262144	MinMax
Traditional	20	3083	262144	MinMax
Traditional	40	11763	262144	MinMax
Traditional	80	45923	262144	MinMax
PINN	10	820	262144	Uniform
PINN	20	3040	262144	Uniform
PINN	40	11680	262144	Uniform
PINN	80	45760	131072	Uniform

As shown earlier in Fig. 4, spherical harmonics takes nearly 10,000 coefficients before beginning to capture the Earth's high-order perturbations. This section aims to quantify how many parameters traditional and physics-informed neural networks require to achieve comparable performance. This is tested by training each of the networks presented in Table 2 on 5,000,000 position/acceleration vector pairs which are drawn randomly from a uniform distribution in altitude (0–420 km), latitude, and longitude. Once trained, MRSE metric (Eq. (3)) is used to evaluate the performance of each network using the same Fibonacci grid data as was used to generate Fig. 4—$\boldsymbol{r}_i \in \{\mathcal{A}, \mathcal{F}, \mathcal{C}\}$. The Fibonacci test samples provide a entirely decoupled dataset from the training data to ensure a fair evaluation of the network performance. The MRSE for the networks is juxtaposed with the MRSE of the spherical harmonic model and presented in Fig. 5.

Figure 5 demonstrates that there exists a wide domain in which the neural networks generate a more compact representation of the gravity field compared to their spherical harmonic counterparts. This range spans between 1,000 and 50,000 parameters (between spherical harmonic degree $l = 30$ and $l = 225$). While the network performance across the sets $\mathcal{A}$ and $\mathcal{C}$ remained comparable to that of the spherical harmonic representation, the performance in $\mathcal{F}$ demonstrates that the PINNs can offer a model that is an order-of-magnitude more compact than the equivalent spherical harmonic model.

There are conditions, however, where the compactness advantage becomes less apparent. Specifically, note the $N = \{10, 80\}$ traditional networks and PINNs. These particular net-

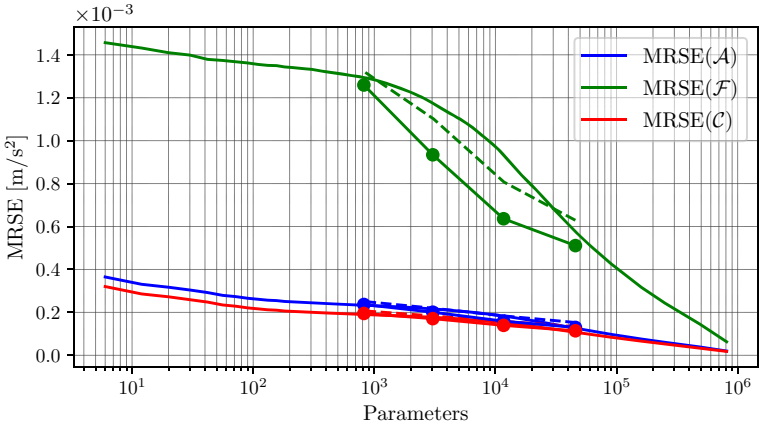


Fig. 5 Plot of MRSE as a function of total model parameters, p. Solid lines represent the spherical harmonic representation. Dashed lines represent traditional neural networks. The lines with circle markers represent the physics-informed neural networks

works have a less pronounced compactness advantage which is theorized to be a result of two factors. In the case of $N = 10$, the networks do not have a sufficiently high modeling capacity to represent the nonlinear perturbations of the gravity field. With a mere $p \approx 1,000$ trainable parameters, these small networks do not have the parametric flexibility necessary to capture the discontinuous high-order perturbations. The $N = 80$ case, in contrast, has such a large network capacity that it manages to model these features quite well—so well in fact that the network begins to overfit to the training data and suffer when tested on new datasets.

This overfitting can be combatted with additional training data. Not only will the overfitting in the $N = \{80\}$ network be resolved by using a larger training dataset, but the performance for each network is expected to improve. Networks that are exposed to more data will observe a more representative cost landscape and larger batches assist stochastic gradient descent in traversing that landscape toward a local minimum. It is therefore reasonable to assume the compactness advantage of these networks would grow even more apparent given a larger training dataset.

The compactness advantages demonstrated in the other networks suggest that the machine learning models are able to learn a set of basis functions that are substantially more efficient at representing the high-order perturbations of Earth's gravity field. In the case of the traditional neural networks, these basis functions and the intermediate nonlinear transformations used to generate them can be directly plotted as shown in Fig. 6.

Figure 6 can be interpreted as follows: The top row represents the first layer in the network—i.e., the normalized Cartesian position vectors inputs. Because these images are generated at the Brillouin sphere, the x and y components of the position vectors grow small at the poles and large at the equator, whereas the z component grows linearly from the south to north pole. The second row represents weighted, linear combinations of the first three inputs that are then passed through the Gaussian exponential linear unit (GELU) activation function. This row corresponds to the outputs of the first hidden layer of the network. This nonlinear transformation is repeated for each intermediate hidden layer until the 9th row of the figure (the 8th hidden layer) which represents the learned basis functions. These penultimate functions are then combined linearly without the GELU transformation to produce the three predicted acceleration components at the Brillouin sphere of Earth (Fig. 7).

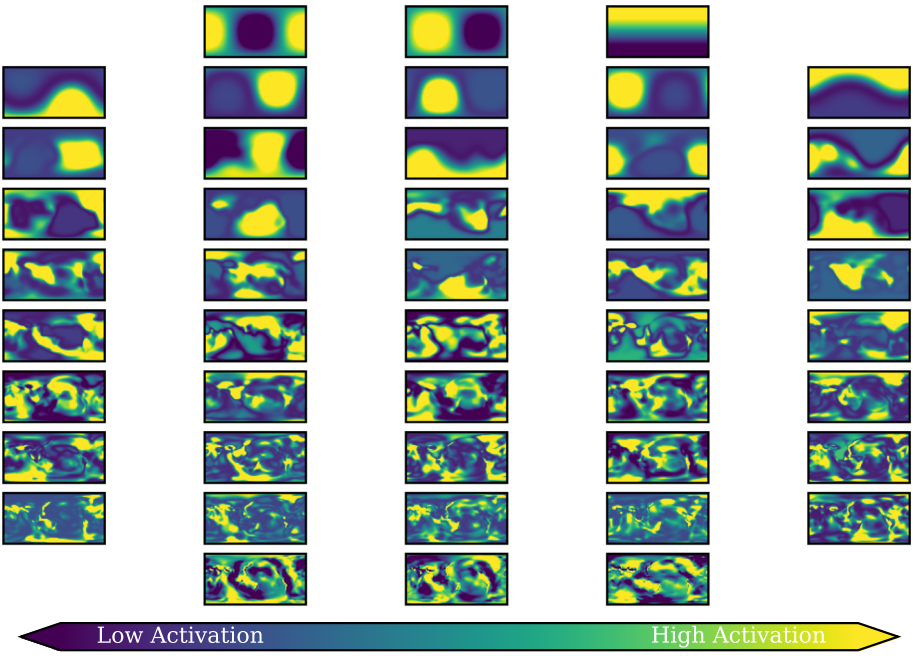


Fig. 6 Subset of the intermediate transformations and resulting basis function of the $N = 40$ traditional neural network. Each row corresponds with a single layer of the network in order of input (top) to output (bottom). The individual plots represent the normalized and dimensionless output of a particular node's activation function when evaluated across the Earth's Brillouin sphere

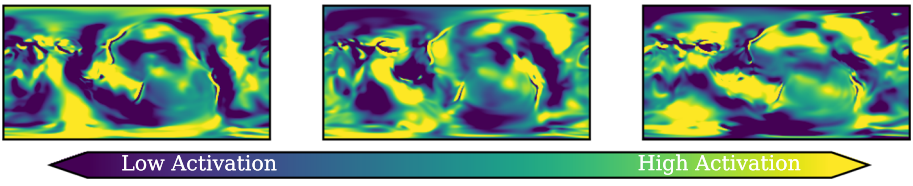
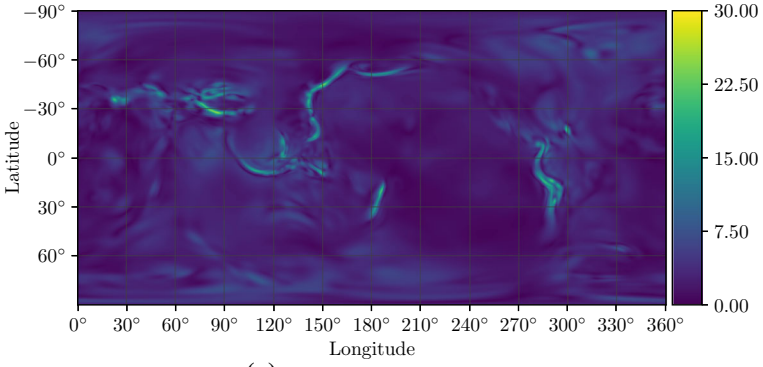


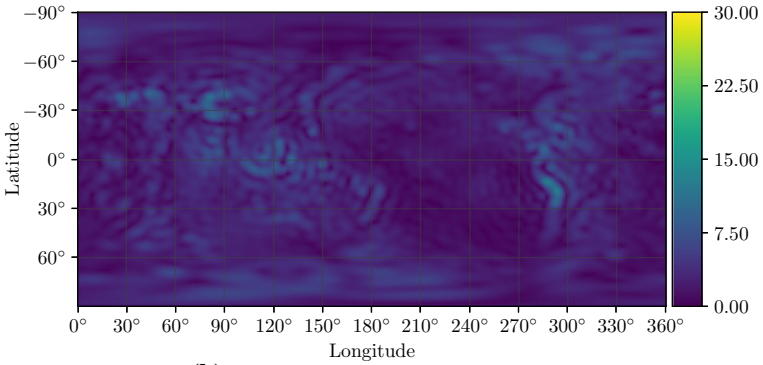
Fig. 7 Zoomed in final layer of Fig. 6 representing the predicted Cartesian components of the acceleration vectors plotted at the Brillouin sphere

Notably, as the inputs propagate deeper into the network, the corresponding outputs of the hidden layers grow increasingly complex. The earlier layers activate over broad regions across the entire Brillouin sphere, whereas the deeper layers activate over more localized features. This highlights how the shallower layers in networks tend to resolve high-level, abstracted feature spaces, while the deeper layers begin conforming to the specific perturbations of the body in question.

Figure 8 visually demonstrates the difference between modeling Earth's discontinuous perturbations using low parameter spherical harmonic model and the more flexible neural network representation. Figure 8a shows how the network representation is able to generate a sensible and accurate basis set capable of representing the most prominent perturbations. Conversely, spherical harmonics prescribe oscillatory basis functions which are not amenable to modeling discontinuous mountain ranges and subduction zones as shown in 8b. In fact, the spherical harmonic basis can leave unintended wave patterns that obfuscate the important



(a) PINN $N = 20 \Leftrightarrow p = 3,040$



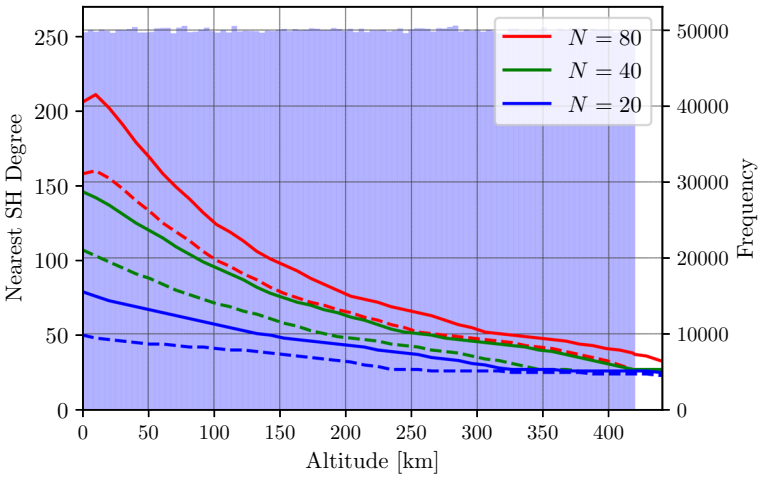
(b) Spherical harmonics $l = 55 \Leftrightarrow p = 3,080$

Fig. 8 Gravity model of Earth using (**a**) the neural network representation and (**b**) the spherical harmonic representation given approximately the same number of free parameters ($p \approx 3000$)

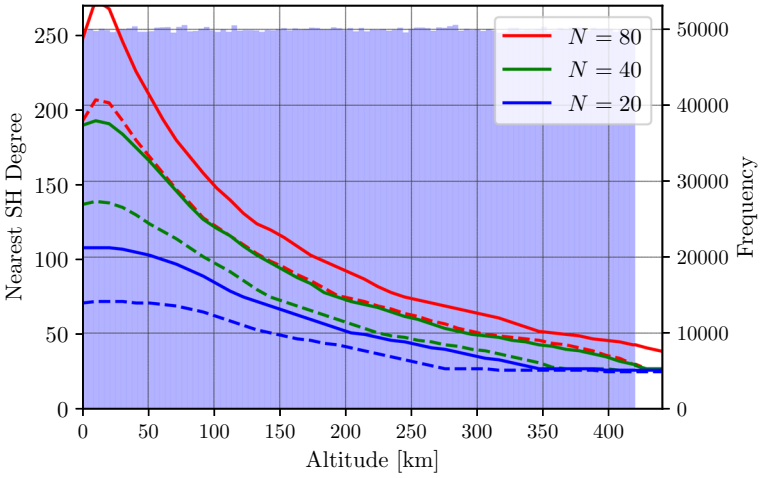
perturbations. Figure 8 thereby highlights one of the important takeaways of this research: Astrodynamicists do not need to apply a "one-size-fits-all" basis (i.e., spherical harmonics) to every gravity field; instead, neural networks provide astrodynamicists with the choice of generating unique basis functions that are maximally efficient for their specific gravity modeling problem.

5 Generalization

The prior analysis focuses on the network gravity models' accuracy at the Brillouin sphere where the perturbations are most prominent. While the surface of the body offers the most complex dynamics due to the unattenuated perturbations, most spacecrafts operate at higher altitudes where dynamics tend to simplify. As shown in Fig. 3, the perturbations tend to decrease in magnitude and span larger spacial scales as altitude increases. Such is a function of the $(R/r)^l$ term within Eq. (1) which rapidly reduces the contribution of high-degree/small wavelength harmonics at larger radii. This second analysis investigates how well the network models generalize to these higher altitudes.



(a) MRSE($\mathcal{A}_m$)



(b) MRSE($\mathcal{F}_m$)

Fig. 9 MRSE of $\mathcal{A}_m$ (top) and $\mathcal{F}_m$ (bottom) for the traditional (dashed) and physic-informed (solid) neural networks converted into the equivalent spherical harmonic degree as function of altitude. The blue histogram represents the training data distribution

5.1 Uniform distribution

Beginning with the 250,000 data Fibonacci grid used in Sect. 4, the same latitude and longitude of sets $\mathcal{A}$ and $\mathcal{F}$ are reused, but their altitudes are varied between 0 and 500 km in 10 km increments to generate sets $\mathcal{A}_m$ and $\mathcal{F}_m$, where $m \in [0, 500]$. The MRSE metric is then applied to these altitude-specific sets, converted into an equivalent spherical harmonic model degree, and plotted in Fig. 9.

Figure 9 demonstrates that the networks' performance tends to decay as a function of altitude. On average the networks outperform their corresponding spherical harmonic equiv-

alent for the first 100 km, but beyond 100 km spherical harmonics become the more compact representation. Across all tested altitudes, however, the networks remain more accurate than degree and order 25 spherical harmonic model suggesting the networks are well-behaved within the domain of the training data and remain a viable option for gravity field modeling up to a LEO altitude.

The altitude-dependent accuracy of the networks is attributed to the attenuation of the high-order perturbations at higher altitudes. Neural networks are trained specifically to minimize error, and the largest errors come from mis-modeling low-altitude samples where the perturbations are the largest. At higher altitudes, the features decay resulting in a smaller cost penalty during training. The features also spread over larger wavelengths such that spherical harmonics can often capture the smoothed perturbations with relatively low-parameter models in a manner that was not possible at the Brillouin sphere.

These results suggest that the network gravity models are most advantageous when used near the surface of a body. For Earth-based spacecraft operations, this is not a common operating regime as, at these altitudes, atmospheric drag alone would produce greater dynamic uncertainty than the high-order gravity perturbations. For bodies with very thin or nonexistent atmospheres however (e.g., the Moon or small-bodies), lower orbit altitudes are feasible and present a viable use case for the network representations.

5.2 Nonuniform distributions

The generalization analysis in Fig. 9 assumes that the training data for the networks are uniformly distributed in altitude as shown by the histogram in blue. The uniform training data distribution, however, is a luxury that is afforded only if a sufficiently high-resolution gravity field model exists from which artificial training data can be generated. In more realistic mission circumstances, a high-fidelity representation of the field may not exist and the network can only train using samples collected from orbit. To reflect these circumstances, two new datasets of 1,000,000 position/acceleration pairs are drawn from the following exponential distribution:

$$\mathbb{E}(x, x_0, \beta) = \exp\left\{-\frac{|x - x_0|}{\beta}\right\}, \tag{15}$$

where x_0 is the reference altitude of 420 km, and β is the scaling parameter. The first dataset sets $\beta = 10$ km and simulates a data distribution collected by a spacecraft that begins in a high-altitude orbit before gradually deorbiting. As such, the majority of the data would come from an operational orbit regime with sparser measurements closer to the surface of the body. The second distribution sets $\beta = 3$ km and represents a satellite in an eccentric orbit that remains at—and collects data from—an operational orbit altitude. There may exist infrequent measurements near the surface, but virtually all of the data are collected at altitudes greater than 200 km. Note that to prevent sampling from inside the Brillouin sphere, the distribution for both datasets is restricted to $x \in [0, 420]$ km. The results of the $N = \{20, 40, 80\}$ traditional networks and PINNs are shown in Fig. 10.

Figure 10 demonstrates that both the traditional neural network and PINN gravity models are most productive when exposed to low-altitude samples, even if infrequently. Figure 10b, d shows that the networks struggle to predict accurate accelerations at the Brillouin sphere given no data, but with a mere 2,000 data collected near the surface (Fig. 10a, c) the networks achieve substantially better performance at lower altitudes. This again highlights how the

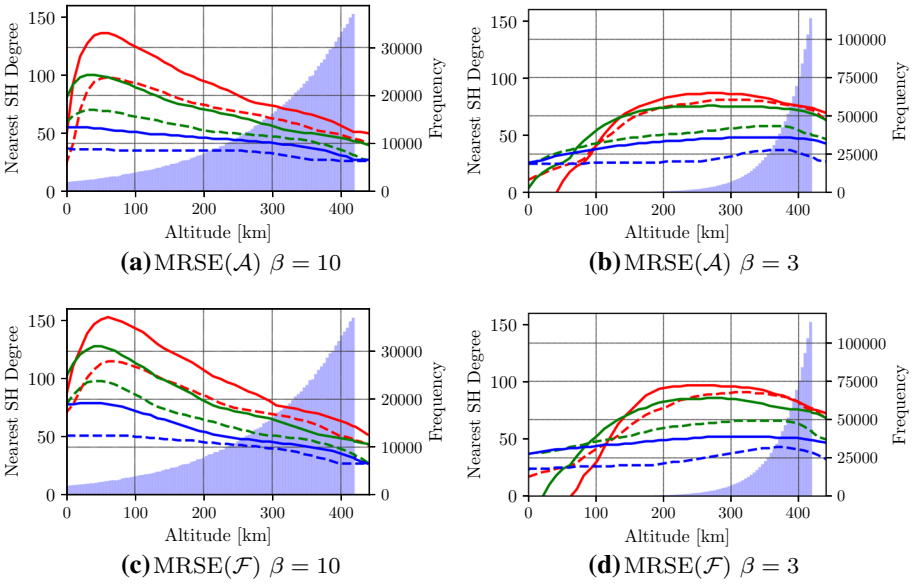


Fig. 10 Training data distribution and equivalent spherical harmonic degree at varying altitudes for the $\beta = 3$ and $\beta = 10$ datasets. Solid lines represent the PINNs and dashed lines represent the traditional neural networks

networks are often biased to focus their modeling effort on low-altitude samples because they quickly dominate the training cost when poorly modeled.

In general, the physics-informed networks consistently outperform the traditional networks with the exception the $N = \{40, 80\}$ PINNs in the low-altitude regime of the $\beta = 3$ distribution. While sub-optimal performance of the more powerful PINN seems counter-intuitive, this ultimately matches expectation. As discussed in Sect. 3 physics-informed networks are typically better at extracting more information from training data than traditional networks. Unfortunately this can lead to overfitting. Given that the majority of data in the $\beta = 3$ distribution comes from samples that exist at an altitude greater than 200 km, the physics-informed networks generate a better model than their traditional counterparts in this regime, but when it comes time to extrapolate into the lower altitudes, the PINNs begin to diverge from truth. In principle this can be improved using traditional L_2 regularization techniques on the network weights (Loshchilov and Hutter 2017) or by supplementing the PINN cost function with additional constraints. The latter of these two approaches should narrow the set of permissible basis functions found by the network, but such exploration is left for future work.

6 Computational speed

The last analysis investigates how quickly the trained network models can be executed as compared to other popular gravity representations. The neural network gravity models are written in TensorFlow 2.4 and executed on an NVIDIA RTX 2060 GPU and on the Ryzen 3400G for the GPU and CPU cases, respectively. Figure 11 shows the execution time required to evaluate the accelerations of 10,000 randomly distributed position data for models of vary-

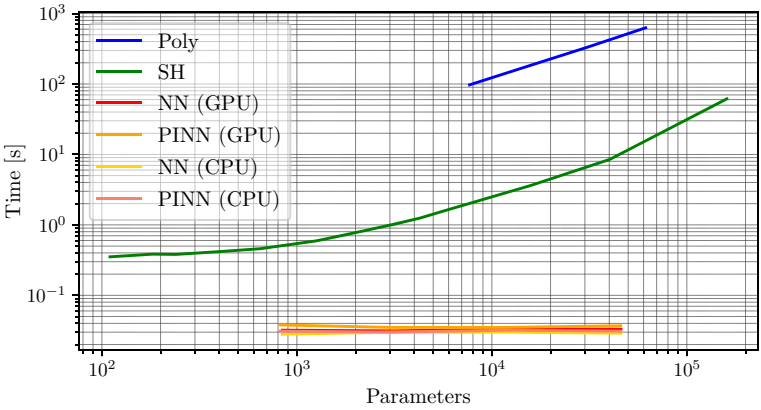


Fig. 11 Total evaluation time to evaluate 10,000 random data using the various gravity models

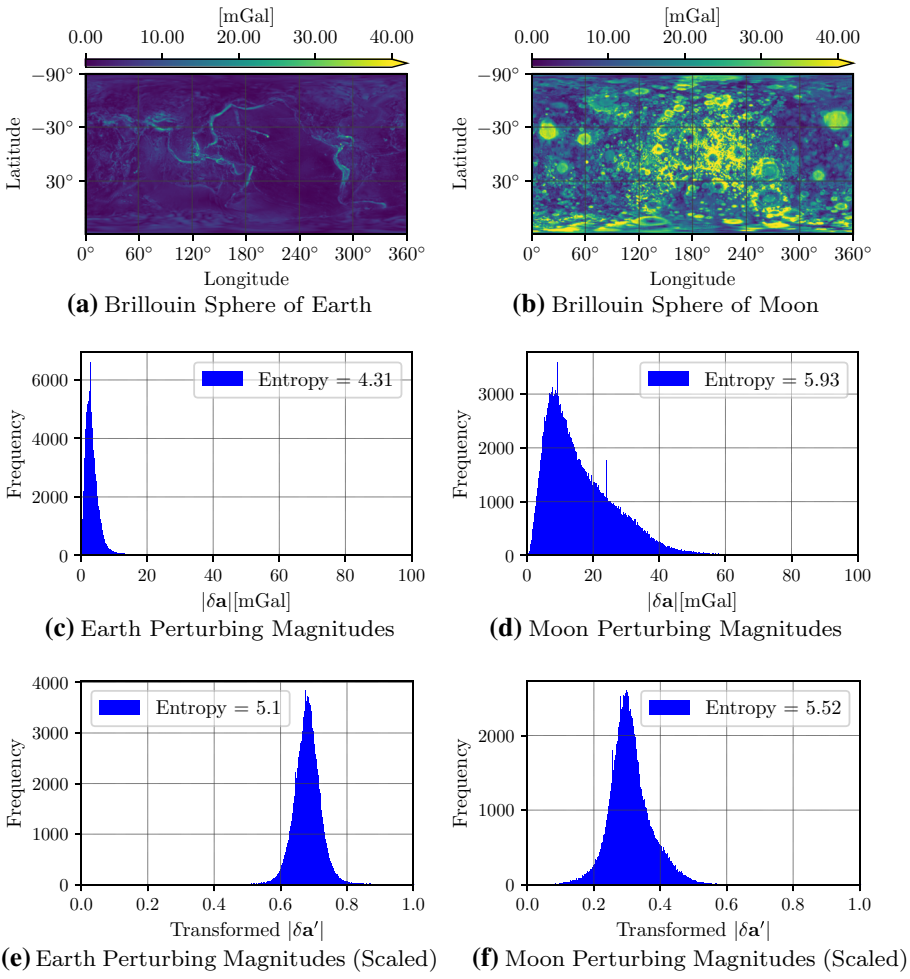
ing levels of fidelity/parameters. The PINN performance is compared first to the spherical harmonic representation, and then to a second analytic gravity representation often used to model small-bodies—the polyhedral gravity model. The spherical harmonics representations are each generated from the EGM-2008 model with different truncation degrees. The polyhedral models tested use increasingly degraded shape models of 433-Eros generated using Blender.[2] The two analytic representations are written in Python, just-in-time compiled using Numba,[3] and executed on a Ryzen 3400G CPU.

Figure 11 demonstrates that the spherical harmonics representation is relatively efficient to evaluate at low degree, but it also has the steepest gradient as the truncation degree increases which verifies its $\mathcal{O}(n^2)$ complexity. The polyhedral representation is by far the most time consuming to evaluate—taking nearly two orders-of-magnitude longer than that of a spherical harmonic model equipped with the same number of parameters. The neural network representation run either on the GPU or CPU is considerably more efficient than both of these representations—with performance that is nearly an order-of-magnitude more efficient than the lowest spherical harmonic model tested ($l = 10$), independent to the number of parameters by these networks.

These results are encouraging for two reasons. First, the PINN gravity model has clear advantage for use in simulation—especially when generating trajectories using high-order integration schemes. In such use cases, these gravity models would need to be evaluated multiple times to evolve the simulation forward a single time-step. As such, the results presented here can be considered a lower bound on performance and greater speedups may exist depending on the specific application. The second application of PINNs quick runtimes comes in the form of on-board control purposes. As recently demonstrated in Blacker et al. (2019) neural networks can be executed on radiation hardened processors. Consequently, PINNs could be considered for use within spacecraft control solutions—allowing for feed-forward terms that efficiently account for high-order gravitational perturbations.

[2] https://www.blender.org/.

[3] https://numba.pydata.org/.



(a) Brillouin Sphere of Earth **(b)** Brillouin Sphere of Moon

(c) Earth Perturbing Magnitudes **(d)** Moon Perturbing Magnitudes

(e) Earth Perturbing Magnitudes (Scaled) **(f)** Moon Perturbing Magnitudes (Scaled)

Fig. 12 Contrasting gravity field and acceleration distributions of the Earth and Moon

7 Network performance applied to the Moon's gravity field

The analyses presented thus far are focused specifically on the Earth, but the conclusions on model compactness and generalization are not necessarily universal to any celestial body whose gravity field is traditionally represented by spherical harmonics. To demonstrate this, the prior experiments are repeated for a body with characteristically different perturbations: the Moon.

The Moon offers an interesting point of comparison to the Earth, as the gravitational perturbations of the Moon are substantially more frequent and of larger magnitude. The Earth's perturbations are typically generated by large, infrequent, and localized geologic structures (mountains, tectonic plate boundaries, etc). The Moon's perturbations, in contrast, are generated by craters and associated mascons which cover most of its surface. As will be shown, the complexity of the Moon's gravity field makes efficient modeling of such perturbations more challenging for the networks.

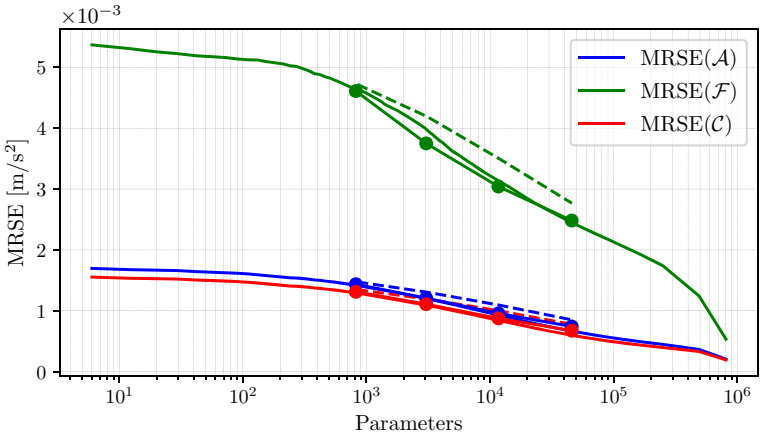


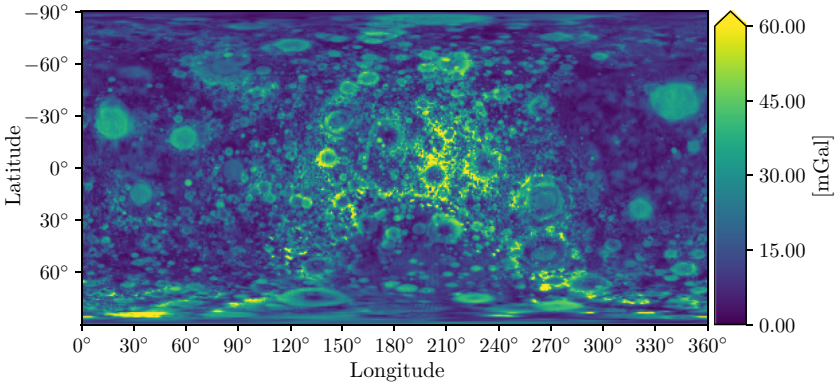
Fig. 13 Plot of MRSE as a function of total model parameters for the Moon. Dashed lines represent traditional neural networks. The lines with circle markers represent the physics-informed neural networks

Figure 12 aims to contrast the differences between the two bodies and their fields. Qualitatively, note how much simpler the Earth's gravity field (Fig. 12a) is compared to that of the Moon (Fig. 12b). The craters on the Moon not only form a near-random surface topology, but also they intermingle some of the strongest perturbations with the weakest. The Earth's perturbations, in comparison, are much more well-behaved. The complexity of the two fields is quantified by estimating the Shannon entropy of the acceleration distributions shown in Fig. 12c, d. For the Earth, the field entropy is 4.31 nats (unit of information expressed in base e), whereas the entropy for the Moon is 5.93 nats. This suggests that the Moon's gravity field contains approximately five times more information on average than that of Earth's gravity field—a significantly more challenging modeling task for the neural networks. These distributions are preprocessed before using them as training data for networks, so Fig. 12e, f shows how the distributions change once having applied the uniform min-max transformation to the acceleration vectors as detailed in Sect. 3. After the preprocessing, the Moon's gravity field still contains more information compared to the Earth, albeit by only 50%. This suggests in both the raw and preprocessed form, the Moon's gravity field is the more challenging modeling task.

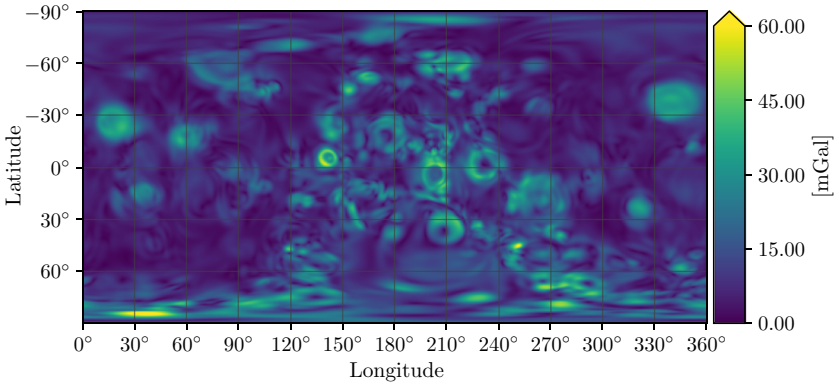
7.1 Compactness

The modeling efficiency of the networks applied to the Moon's gravity field is shown in Fig. 13. The figure replicates the same experiment of Sect. 4, but instead of training the networks from field points drawn uniformly between 0 and 420 km altitude, the training data span between a 0 and 50 km altitude. This narrower training domain attempts to reflect the fact that spacecraft can orbit at lower altitudes around bodies like the Moon given its very thin atmosphere. In addition, the true potential, $U^{\text{SH}}_{\text{Truth}}$, is generated using the lunar GRGM1200A gravity field model (Goossens et al. 2016).

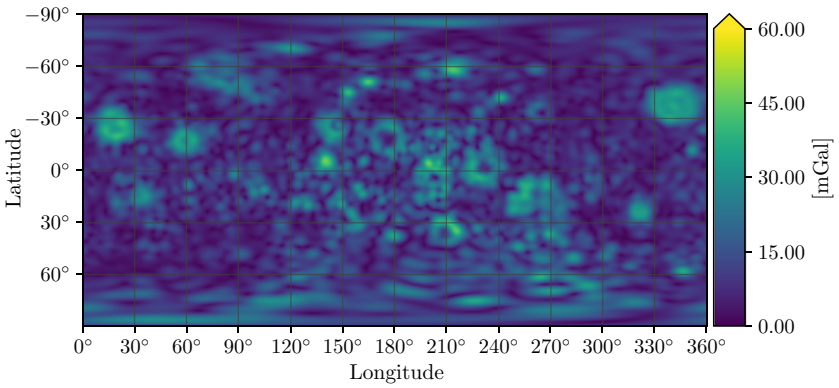
In Fig. 13, note how the spherical harmonic curve differs substantially from the Earth's curve (Fig. 5). In the case of the Earth's gravity field, nearly 10,000 spherical harmonic coefficients are required to begin converging on the feature set, $\mathcal{F}$. Such requirement implies that only the small-wavelength harmonics are useful when modeling the localized perturbations



(a) True Field



(b) PINN $N = 20 \Leftrightarrow p = 3,040$



(c) SH $l = 55 \Leftrightarrow p = 3,080$

Fig. 14 Gravity model of Moon using (**a**) the full $l = 1,000$ spherical harmonic model, (**b**) the PINN representation with $p = 3040$, and (**c**) the low fidelity $l = 55 \Leftrightarrow p = 3080$ spherical harmonic representation

found on the Earth. A similar high-parameter requirement is not exhibited for the Moon's gravity field. Rather, a mere 300 spherical harmonic coefficients are needed to begin converg-

ing on the Moon's perturbations. This suggests that long-wavelength harmonics provide a meaningful contribution to the modeling effort. This is attributed to the near-isotropic nature of the perturbations. Even if the low-degree harmonics cannot efficiently reduce the error of a single crater, they do contribute a small amount to all of the craters covering the entire surface. This collective contribution generates a measurable reduction in error seen within the feature set, $\mathcal{F}$. To this end, spherical harmonics are substantially better suited to model the Moon's gravity field features than was the case for the Earth.

Turning to the neural network performance, Fig. 13 shows that only the PINN representation offers a representational advantage over spherical harmonics and by a small margin ($\approx$ 30% instead of the 1000% seen for the Earth). The relatively small performance gain is an interesting result attributed to two factors: First is the aforementioned efficiency of the long wavelengths in the spherical harmonic representation. The analytic model is simply more effective for the Moon than it was for the Earth, so the networks are competing with a more productive representation.

The second contribution to the performance discrepancy is the greater complexity/higher entropy of the Moon's gravity field. The networks are being tasked to generate a distribution that contains more information than was present in the Earth's gravity field—i.e., the networks have to work harder. It remains possible that greater network performance exists, but to witness that performance the networks would likely require additional training data, additional feature engineering, and longer training times to compensate for this more complex problem.

Future work is required to determine whether there is a more fundamental relationship between the entropy of a gravity field and the capacity for a successful neural network gravity model. As is shown in Fig. 14, the learned, custom basis functions provided by the PINNs do not seem to offer a major advantage to modeling the Moon's gravity environment. Because environments with high gravitational entropy will generate accelerations that appear unstructured or random, it is possible that all basis sets (learned or analytic) will struggle equally to model these types of fields.

7.2 Generalization

Repeating the methodology presented in Sect. 5, the trained network model performance is tested at varying altitudes to determine how these networks generalize to different orbits. The altitudes tested vary only between 0 and 55 km instead of the 0–500 km domain to remain consistent with Sect. 7.1. Otherwise the same analysis is repeated and presented in Fig. 15.

The general performance trend of the Moon networks is similar to that of the Earth networks. Both the traditional and physics-informed neural networks tend to have degraded accuracy at higher altitudes. There is, however, a difference in the magnitude of the performance peak in the low-altitude regime. In the original Earth networks (Fig. 9b), the performance peaked around 15 km above the Brillouin sphere. This performance peak for the Earth networks demonstrates that the learned representation is about 15% less efficient at the Brillouin sphere than at a 15 km altitude. This sizeable dip in performance is attributed to the fact that neural networks often struggle when tested at the boundaries of their original training data as there was nothing to constrain the fit from the "other side." Interestingly, the performance peak for the Moon networks is less pronounced (the fit at the Brillouin sphere is $\approx$ 5% less compact) and occurs at a lower altitude of 5 km.

This discrepancy is again a by-product of the Moon's high-entropy gravity field. The field's greater complexity ultimately acts as a type of regularization that helps prevent the networks

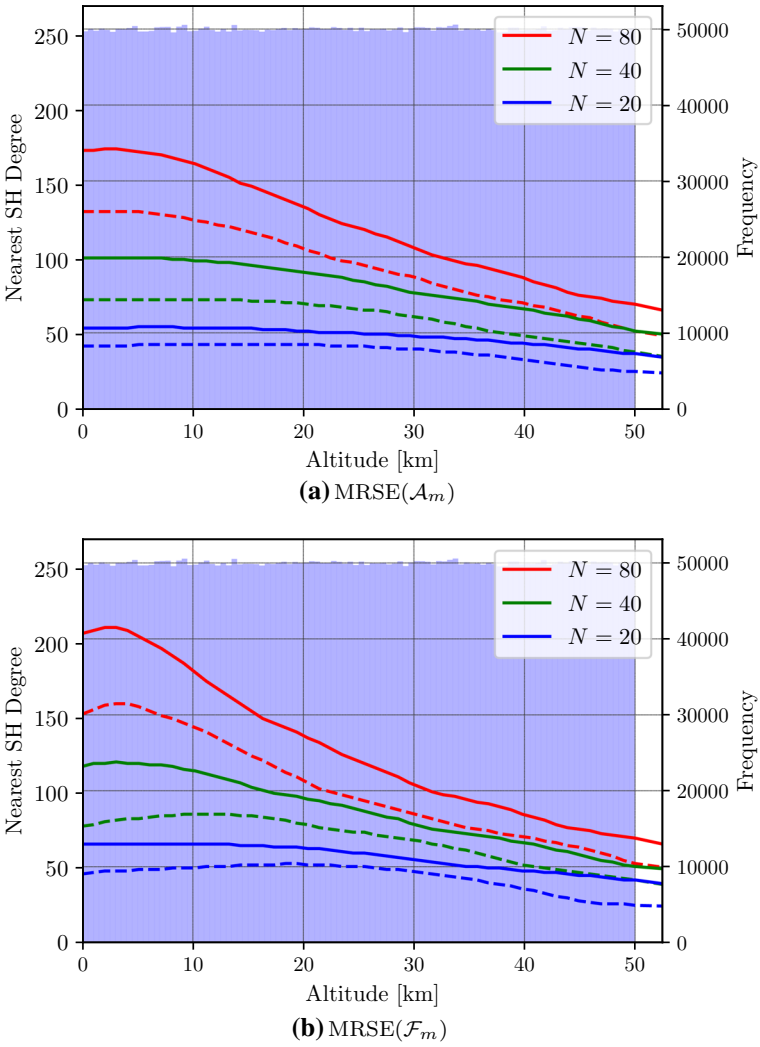


Fig. 15 MRSE of $\mathcal{A}_m$ (top) and $\mathcal{F}_m$ (bottom) for the traditional (dashed) and physic-informed (solid) neural networks converted into the equivalent spherical harmonic degree as function of altitude for the Moon. The blue histogram represents the training data distribution

from overfitting to the training data. The perturbations are so diverse and ever-present that the networks can not develop an overconfidence in their representation. Analogous effects are seen when using small mini-batch sizes to train neural networks. These small batches can cause a noisier gradient descent which takes longer to converge but assists the optimizer in exploring the cost landscape (Goodfellow et al. 2016). The difference here is instead of decreasing the batch size, the complexity of the cost landscape is increased to achieve a similar effect. This prevents large extrapolation error near the bounds of the training data domain.

8 Conclusions

Combining new machine learning techniques and artificial neural networks to solve the gravity field modeling problem offers an interesting alternative to more traditional and analytic approaches. For the case of the Earth, the neural network representation of the gravity field offers strong advantages to modeling highly discontinuous and perturbing features without the need for excessive parameters. While the accuracy of such representations tends to decay as a function of altitude, they ultimately remain stable up to a LEO orbit—achieving accuracy greater than a degree $l = 25$ field at their weakest. Finally, the neural network representation also has the advantage of fast execution times which could make it a viable modeling alternative both on-board spacecraft and within simulation.

There remain conditions, however, for which the network representation's appeal grows less prominent. Early results suggest that gravity fields with high information content like the heavily cratered surface of on the Moon tend to limit neural network performance. Modeling such complex environments using neural networks may require additional physics informed constraints, training data, and feature engineering to achieve appreciably superior performance.

Altogether the physics-informed neural network gravity model is a novel and powerful way to represent the gravity field of large celestial bodies and offers a number of encouraging prospects for future research. In particular, all PINNs in this paper included only one physics constraint ($\boldsymbol{a} = -\nabla U$). There exists many more constraints that can be applied to the gravity modeling problem. Such efforts will likely require adaptive scaling of the differing physics constraints as demonstrated recently in Wang et al. (2020) to improve performance. Moreover, there remains many opportunities for exploration of more advanced architectures and more carefully selected hyperparameters.

Acknowledgements This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 2040434.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Blacker, P., Bridges, CP., Hadfield, S.: Rapid Prototyping of Deep Learning Models on Radiation Hardened CPUs. IEEE, pp 25–32, https://doi.org/10.1109/AHS.2019.000-4, https://ieeexplore.ieee.org/document/8792934/ (2019)

Bottou, L.: Stochastic Gradient Descent Tricks, vol 7700. Springer (2012), https://doi.org/10.1007/978-3-642-35289-8_25, http://link.springer.com/10.1007/978-3-642-35289-8_25

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

Courant, R., Hilbert, D.: Methods of Mathematical Physics. Wiley, New York. https://doi.org/10.1002/9783527617234 (1989)

Furfaro, R., Barocco, R., Linares, R., Topputo, F., Reddy, V., Simo, J., et al.: Modeling irregular small bodies gravity field via extreme learning machines and Bayesian optimization. Adv. Space Res. **67**, 617–638 (2021). https://doi.org/10.1016/j.asr.2020.06.021

Gao, A., Liao, W.: Efficient gravity field modeling method for small bodies based on Gaussian process regression. Acta Astronaut. **157**, 73–91 (2019). https://doi.org/10.1016/j.actaastro.2018.12.020
Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. J. Mach. Learn. Res. **9**, 249–256 (2010)
Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
Goossens, S., Lemoine, F.G., Sabaka, T.J., Nicholas, J.B., Mazarico, E., Rowlands, D.D., et al.: A Global Degree and Order 1200 Model of the Lunar Gravity Field using GRAIL Mission Data. https://ui.adsabs.harvard.edu/abs/2016LPI....47.1484G (2016)
Gottlieb, D., Shu, C.W.: On the Gibbs phenomenon and its resolution. SIAM Rev. **39**, 644–668 (1997). https://doi.org/10.1137/S0036144596301390
Hewitt, E., Hewitt, R.E.: The Gibbs–Wilbraham phenomenon: an episode in Fourier analysis. Arch. Hist. Exact Sci. **21**, 129–160 (1979). https://doi.org/10.1007/BF00330404
Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**, 489–501 (2006). https://doi.org/10.1016/j.neucom.2005.12.126
Kaula, W.M.: Theory of Satellite Geodesy: Applications of Satellites to Geodesy. Blaisdell Publishing Co, New York (1966)
Kingma, DP., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings pp 1–15, http://arxiv.org/abs/1412.6980 (2014)
Koch, K.R., Morrison, F.: A simple layer model of the geopotential from a combination of satellite and gravity data. J. Geophys. Res. **75**, 1483–1492 (1970). https://doi.org/10.1029/JB075i008p01483
LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015). https://doi.org/10.1038/nature14539
Lemoine, F.G., Goossens, S., Sabaka, T.J., Nicholas, J.B., Mazarico, E., Rowlands, D.D., et al.: GRGM900C: a degree 900 lunar gravity model from GRAIL primary and extended mission data. Geophys. Res. Lett. **41**, 3382–3389 (2014). https://doi.org/10.1002/2014GL060027
Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019 http://arxiv.org/abs/1711.05101 (2017)
Manzi, M., Vasile, M.: Discovering Unmodeled Components in Astrodynamics with Symbolic Regression. IEEE, pp. 1–7, https://doi.org/10.1109/CEC48606.2020.9185534 (2020)
Martin, J.R., Schaub, H.: GPGPU Implementation of Pines' Spherical Harmonic Gravity Model. Univelt Inc., Escondido (2020)
Mertikopoulos, P., Papadimitriou, C., Piliouras, G.: Cycles in adversarial regularized learning. Soc. Ind. Appl. Math. (2018). https://doi.org/10.1137/1.9781611975031.172
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
Pines, S.: Uniform representation of the gravitational potential and its derivatives. AIAA J. **11**, 1508–1511 (1973). https://doi.org/10.2514/3.50619
Raissi, M., Perdikaris, P., Karniadakis, G.: 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
Romain, G., Jean-Pierre, B.: Ellipsoidal harmonic expansions of the gravitational potential: theory and application. Celest. Mech. Dyn. Astron. **79**, 235–275 (2001). https://doi.org/10.1023/A:1017555515763
Ruder, S.: An Overview of Gradient Descent Optimization Algorithms, pp. 1–14, arXiv arXiv:1609.04747 (2016)
Russell, R.P., Arora, N.: Global point mascon models for simple, accurate, and parallel geopotential computation. J. Guid. Control Dyn. **35**, 1568–1581 (2012). https://doi.org/10.2514/1.54533
Swinbank, R., Purser, R.J.: Fibonacci grids: a novel approach to global modelling. Q. J. R. Meteorol. Soc. **132**, 1769–1793 (2006). https://doi.org/10.1256/qj.05.227
Takahashi, Y., Scheeres, D.: Morphology driven density distribution estimation for small bodies. Icarus 233, 179–193 (2014), doi: 10.1016/j.icarus.2014.02.004
Tapley, B.D.: Gravity model determination from the GRACE mission. J. Astronaut. Sci. **56**, 273–285 (2008). https://doi.org/10.1007/BF03256553
Tardivel, S.: The Limits of the Mascons Approximation of the Homogeneous Polyhedron. Am. Inst. Aeronaut. Astronaut. (2016) https://doi.org/10.2514/6.2016-5261
Wang, S., Teng, Y., Perdikaris, P.: Understanding and mitigating gradient pathologies in physics-informed neural networks, pp. 1–28, arXiv arXiv:2001.04536 (2020)

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**, 313–344 (1997). https://doi.org/10.1007/BF00053511

Wittick, P.T., Russell, R.P.: Mixed-model gravity representations for small celestial bodies using mascons and spherical harmonics. Celestial Mechanics and Dynamical Astronomy **131**, 31 (2019). https://doi.org/10.1007/s10569-019-9904-6