AAS 23-110

THE PHYSICS-INFORMED NEURAL NETWORK GRAVITY MODEL REVISITED: MODEL GENERATION III

J. R. Martin* **and H. Schaub†**

Scientific machine learning and the advent of the Physics-Informed Neural Network (PINN) show tremendous potential in their capacity to identify solutions to complex differential equations. Over the past two years, considerable work has gone into the development of PINNs capable of solving the gravity field modeling problem — i.e. learning a differentiable form of the gravitational potential from position and acceleration estimates. While the past PINN gravity model (PINN-GM) instances have demonstrated advantages in model compactness, robust to noise, and sample efficiency; there remain numerous potential failure points of these models which this paper aims to address. Specifically, this paper introduces the third generation of the Physics-Informed Neural Network Gravity Model (PINN-GM-III) which aims to solve problems of extrapolation error, numerical instability, bias towards low altitude samples, and compliant boundary conditions through numerous modifications to the model's design. Performance of PINN-GM-III is compared to its past two processors, looking specifically at test cases of the Earth and the asteroid 433 Eros.

INTRODUCTION

Recent developments in the field of scientific machine learning offer novel ways for dynamicists to learn high-fidelity solutions to complex differential equations using neural networks.[1–3] Thus far, much of the scientific machine learning literature focuses on the fields of fluid mechanics or structures for which there exist notoriously complex partial differential equations which lack analytic solutions. Less attention has be paid to the astrodynamics community, despite it offering an additional and fruitful use case for these tools.

In particular, there exist many physical processes within astrodynamics that abide by some known underlying differential equations, but whose solution is analytically complex or intractable. One ubiquitous problem facing the community is that of gravity field modeling. Given a collection of estimates of the gravitational accelerations produced by some celestial body, how can dynamicists regress a high-fidelity model of the body's gravitational potential before exploiting it for mission design?

Traditionally the reconstruction of these gravity fields is an effort left to popular analytic tools and strategies like fitting a spherical harmonic model to the data or assuming a constant density

*NSF Graduate Research Fellow, Ann and H. J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309.

†Professor and Department Chair, Schaden Leadership Chair, Ann and H. J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309, AAS Fellow, AIAA Fellow.

polyhedral as a reasonable first order solution.[4,5] While these approaches provide compelling analytics, they come with considerable overhead ranging from computational inefficiencies, limiting operational conditions, or assumptions about the body in question.

Given the ever-increasing precision required for future missions, these challenges have become problematic and dynamicists have reignited the search for novel gravity models capable of avoiding these difficulties. Among the most promising efforts in this search is the use of machine learning algorithms to learn new gravity models. These efforts include using extreme learning machines,[6] Gaussian processes,[7] neural networks,[8] and neural density fields[9] to learn mappings from position to acceleration. These efforts have been shown to offer advantages in computational efficiency; however, they remain prone to their own idiosyncratic disadvantages. Many of these machine learning models rely on the assumption that near infinite volumes of perfectly measured position and acceleration data already exists from which the models can be trained. In addition, most of these solutions make no guarantees that the learned models are satisfying relevant physics properties inherent to gravitational potentials. Furthermore, no studies investigate how the models extrapolate beyond the bounds of their training data, nor where within their training domain they preform sub-optimally.

In 2022, the physics-informed neural network gravity model (PINN-GM) was introduced, aiming to address many of these pitfalls.[10] Physics-informed neural networks are a class of machine learning algorithm which augment traditional cost functions with known differential constraints. For the gravity field modeling problem, this means that a neural network must not only learn a mapping from a position to an acceleration, but the network must also ensure that the acceleration produced is a direct byproduct of a more fundamental gravitational potential through the differential equation $-\nabla U = \boldsymbol{a}$. Additional properties like $\nabla^2 U = 0$ and $\nabla \times \nabla U = 0$ can also be added to provide additional constraints which guide the network towards an accurate and physically compliant solution. By adding these dynamical constraints to the cost functions of neural networks, these models maintain the computational efficiency of past machine learning models, but do so without compromising on the physics. Not only are these physics-informed constraints analytically satisfying, they also provide practical and quantitative improvements to the learned gravity field model in terms of model compactness, robustness to noise, and generalization capabilities.[10]

The initial PINN gravity model was trained and tested on the Earth and Moon systems, and despite its promising results, it was relatively difficult to train — requiring large training data sets and long training times before converging to a high-fidelity solution. Reference 11 proposes a variety of new design choices to improve network robustness, sample efficiency, and accuracy in the second generation of the PINN gravity model (PINN-GM-II), and demonstrates how these models can not only be useful for representing the fields of large planetary bodies, but also the more exotic celestial bodies like asteroids and comets.

Despite these improvements, there remain open challenges facing the PINN-GM. Specifically, both PINN-GM-I and II use a cost function which inadvertently leads models to prioritize low-altitude field points, where the accelerations are largest. This bias in altitude causes the networks to spend considerably less of their modeling capacity on the high-altitude regimes leading to poor performance in these domains. In addition, even when trained on data from exclusively high-altitudes, the PINN-GM can suffer numerical instabilities if the domain of the training data is sufficiently large.

This paper introduces a variety of design changes to the PINN-GM to address these challenges, and then revisits some of the original environments studied in the PINN-GM-I and PINN-GM-II

papers to quantify how these changes affect performance.

The organizational outline for the paper is as follows: First, a high-level overview of the spherical harmonic, polyhedral, and PINN gravity models is presented. Following this, a proposed set of modifications to the PINN-GM are introduced. A set of experiment are then performed to characterize the effect of these changes as compared to past PINN gravity models using the Earth's gravity field as a test case. Similar experiments are repeated on the asteroid 433-Eros to highlight the versatility of these changes to multiple astrodynamics settings. Finally a set of early-stage experimental modifications are discussed and their preliminary results assessed before offering concluding remarks on the current state of the PINN-GM-III model.

METHODS

There exists a plethora of gravity models, each with their own advantages and disadvantages. This section details some of the most common.

Spherical Harmonics Model

Early in the 1900s, it was proposed that spherical harmonic basis functions could be superimposed together to produce high-fidelity estimates of the Earth's gravitational potential.[12]

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

Equation 1 is referred to as the spherical harmonic gravity model where r is the radius to the field point, μ is the gravitational parameter of the body, R is the circumscribing radius of the body, l is the degree of the model, m is the order of the model, $C_{l,m}$ and $S_{l,m}$ are the spherical harmonic coefficients, λ is the longitude, ϕ is the latitude, and $P_{l,m}$ are the associated Legendre polynomials.[4]

One of the most compelling advantages of this spherical harmonic gravity model is how it can compactly capture one of the largest gravitational perturbations produced by large celestial bodies: planetary oblateness or J_2. Specifically, Equation 1 need only be expanded to a mere degree $l = 2$ and order $m = 0$ for the majority of Earth's oblateness to be represented within the model.

Despite its success in modeling planetary oblateness, spherical harmonics suffer when tasked to model the remaining and more discrete gravitational perturbations like mountain ranges, tectonic plate boundaries, and hotspots. These discontinuous features can require hundreds-of-thousands of harmonics superimposed together to be accurately captured with this basis set.[10] Not only does this result in memory inefficient models, but it also comes with a computational expense. Specifically, spherical harmonics require evaluating high-degree associated Legendre polynomials which can only be computed recursively. There exists no trivial way to parallelize these computations, leading to an unavoidable $\mathcal{O}(n^2)$ computational complexity which can quickly grow limiting for both ground-based simulation as well as processors flown on-board.[13]

Beyond the computational inefficiencies, spherical harmonics also has operational limits, diverging inside of the bounding (Brillouin) sphere of any celestial body.[5] While such effects are negligible for near-spherical planets or moons, they become problematic in the small-body settings. For asteroids or comets that exhibit highly non-spherical geometries, these effects can lead to large errors in the predicted accelerations and potentially risk the safety of a spacecraft.

Polyhedral Gravity Model

To combat spherical harmonics' divergence inside the Brillouin sphere, a polyhedral gravity model can be used instead.[5] If a shape model (a collection of triangular facets and vertices) of the body is available, then it can be used to analytically compute the potential of that body under the assumption of constant density 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 field point, L_e is an analog to the potential contribution by the edge, $\mathbf{F}_f$ is the face normal dyad, $\mathbf{r}_f$ is the distance between the face normal and the field point, and ω_f is analog to the potential contribution by the face.

While the polyhedral gravity model circumvents the numerical divergence within the Brillouin sphere, it comes with three distinct disadvantages. First, this gravity model is also computationally expensive. Many shape models have hundreds-of-thousands of facets and vertices which must be individually looped over to compute the acceleration at a single field point. While this model can be computed on parallelized hardware, such computational capabilities are not yet available on-board spacecraft — preventing this model from being effectively flown.

Second, the polyhedral gravity model must assume a density profile for the body in question. Often the density is taken to be constant, but literature shows that such assumption is not necessarily valid.[14,15] Third and finally, the polyhedral gravity model requires that dynamicists have a pre-computed shape model of the body which is non-trivial to acquire in practice.[16–18]

Traditional Machine Learning Models

Given these challenges and the increased rise in small-body exploration, there has been renewed interest in formulating novel solutions to the gravity field modeling problem. Given the success of deep learning to solve complex problems in multiple academic communities,[19] many gravity modeling researchers have begun turning to machine learning as a potentially fruitful avenue of exploration. The majority of machine learning gravity model efforts center on using tools like Gaussian processes,[7] extreme learning machines,[6] and neural networks[8] to learn a mapping from position to gravitational acceleration in a data-driven manner.

At a high level, many of these machine learning models seek minimize the following loss function:

$$L(\theta) = \frac{1}{N} \sum_{i=0}^{N} |\hat{\boldsymbol{a}}_i(x_i|\theta) - \boldsymbol{a}_i|^2 \tag{3}$$

where $\hat{\boldsymbol{a}}_i(\boldsymbol{x_i}|\theta)$ is the machine learning model prediction of the acceleration at field point $\boldsymbol{x}_i$ with trainable model parameters θ. These parameters can be iteratively updated using a form of stochastic gradient descent through

$$\theta^{m+1} = \theta^m - \eta \nabla_{\theta^m} L(\theta^m) \tag{4}$$

where η is a user-specified learning rate and m is the training iteration.

Once trained, these machine learning models demonstrate considerable promise in their ability to reduce the computational loads required to compute gravitational accelerations about small bodies.

Despite this, there remain multiple weaknesses of these approaches. Foremost, these models require prohibitively large sets of training data. For celestial systems in which a high-fidelity gravity model already exists, this challenge can be surmounted by generating artificial training data from said model. Unfortunately, the only systems for which this is true are that of the Earth and Moon. In most other astrodynamic settings, there does not exist a preexisting model, and these systems must be trained on far sparser and noisier estimates of the position and acceleration. Second, most machine learning models do not extrapolate well outside of the bounds of their training data – intrinsically limiting their operational domain. Third, these model offer no guarantees that they satisfy important physics properties like Laplace's Equation $\nabla^2 U = 0$ from which the spherical harmonic model and polyhedral models are derived.

Physics Informed Neural Network Gravity Model

Physics-Informed Neural Network (PINN) are a relatively new class of neural networks capable of addressing some of these challenges. PINNs are neural networks which explicitly incorporate known differential constraints into their cost function.[1] By penalizing violations of these constraints during training, the networks are forced to learn solutions which are naturally compliant with the underlying physics of the system.

The Physics-Informed Neural Network Gravity Model (PINN-GM) explores the application of this new class of neural network for the gravity modeling problem.[10] Rather than training traditional neural networks with a cost function like what is shown in Equation 3, the PINN-GM used a cost function of which leverages the known differential equation $-\nabla U = \boldsymbol{a}$ through

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

where ∇U is taken via automatic differentiation of the neural network.[20]

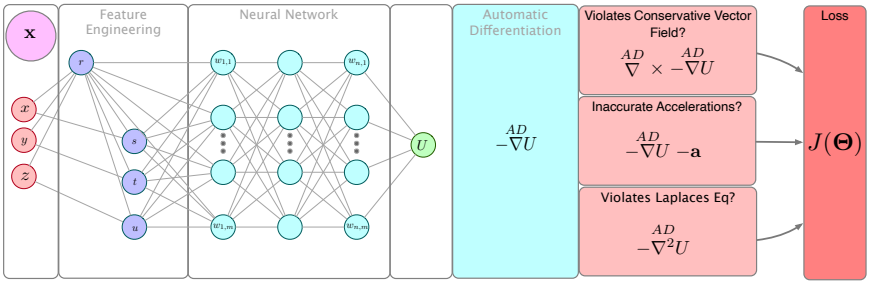


Figure 1: PINN-GM Generation II[11]

The first generation PINN-GM was trained on Earth and Moon systems at low altitude, and it was demonstrated that the PINN-GM is capable of learning more parametrically compact representations and computationally efficient gravity models than their spherical harmonic counterpart.[10] Reference 11 extends this work by investigating the effects of adding the additional dynamical constraints of $\nabla^2 U = 0$ and $\nabla \times \nabla U = 0$ into the cost function and suggests a collection of design modifications to the network architecture (see Figure 1) for improved performance around small-bodies.

While the PINN-GM-II offers welcomed improvements to its first generation predecessor, there remains opportunity for increased performance and robustness. Specifically, the past PINN-GMs

struggle to prioritize high altitude samples, suffer from numerical instabilities, and offer no solution to enforce gravity field boundary conditions. This paper attempts to solve these problems, among others, with the introduction of the next generation of the PINN-GM.

PINN-GM-III

The third generation PINN gravity model (PINN-GM-III) introduces a collection of new design choices aimed at addressing various failure points of past generations, all the while increasing model accuracy and robustness. A high-level architectural glance of these changes are shown in gray boxes in Figure 2, and can be juxtaposed with the previous PINN-GM-II in Figure 1. The details of each modification are detailed in the subsections below.

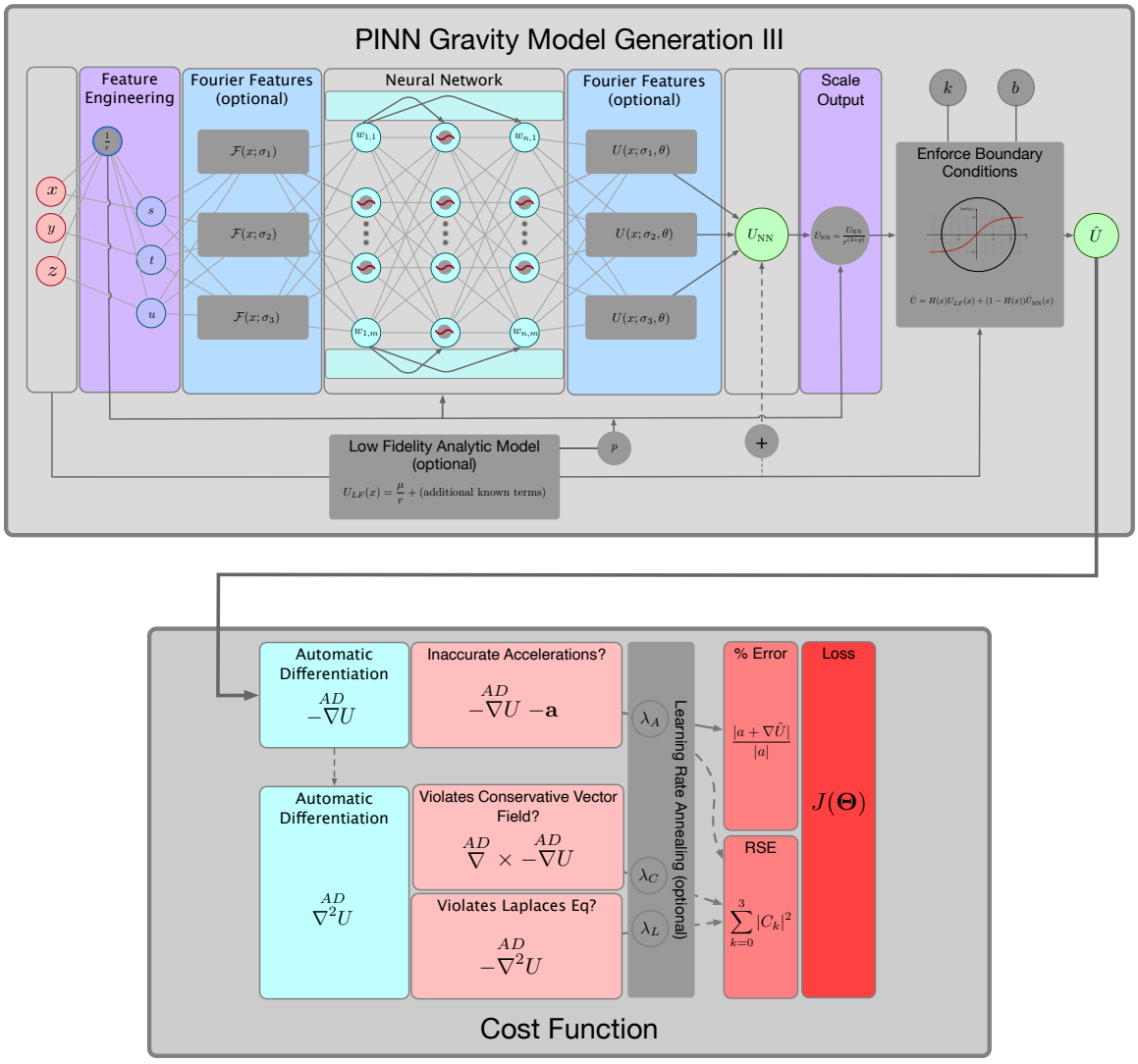


Figure 2: PINN-GM Generation III with new modifications contained in dark gray boxes

Modified Loss Function to Account for High Altitude Samples

The loss function for the original PINN-GM is the root mean squared (RMS) metric:

$$\mathcal{L}_{\text{RMS}}(\theta) = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left|-\nabla\hat{U}(x_i|\theta) - \boldsymbol{a}_i\right|^2} \tag{6}$$

This loss function is among the most common in machine learning and attempts to minimize the most flagrant residuals between the true and predicted values. Unfortunately, this metric comes with a unique disadvantage in the gravity field modeling problem. Gravitational accelerations produced near the surface of the body have considerably larger magnitudes than the accelerations produced at high altitudes. As a consequence, a small relative error in the low-altitude prediction can obfuscate a large relative error in high-altitude prediction. This artificially encourages the network to dedicate the majority of its modeling capacity towards low-altitude regimes which comes at the expense of accurately modeling the gravity field at higher altitudes.

To combat this flaw, PINN-GM-III adopts a new loss function, mean percent error:

$$\mathcal{L}_{\%}(\theta) = \frac{1}{N}\sum_{i=0}^{N}\frac{|-\nabla\hat{U}(x_i|\theta) - \boldsymbol{a}_i|}{|\boldsymbol{a}_i|} \tag{7}$$

By using percent error as the loss function rather than the RMS, PINN-GM-III ensures that all regions of the gravity field, regardless of altitude, are prioritized by the network. If a user has applications which are especially sensitive to low-altitude accelerations, Equation 7 can optionally be augmented with Equation 6 as suggested in Figure 2.

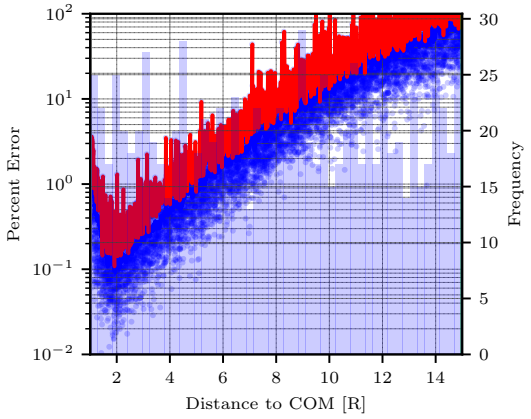
To demonstrate the effect of these changes on network performance, a simple test is proposed which trains two PINN-GMs on 1,000 position / acceleration training data pairs distributed from 0 - 15 Earth radii above the Earth's surface. One PINN-GM is trained using the original RMS loss function (Equation 6) and the other is trained on with the percent error loss function (Equation 7). Once trained, each network is evaluated on a set of 10,000 randomly distributed test points within the 0-15R domain, and their resulting RMS and percent error values are reported as a function of altitude in Figure 3.

Figure 3a confirms that the networks trained with the RMS loss function disproportionally favor low-altitude field points (0-2R) at the expense high altitude field points (5-15R). In contrast, the PINN-GM model trained with the percent error loss function prioritizes accurate modeling at all altitudes, and conveniently also produces lower RMS values for high altitude samples as shown in Figure 3d.

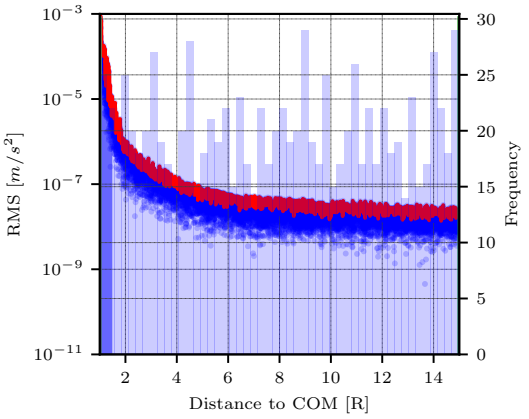
Enforcing Boundary Conditions through Model Design

While PINNs are most commonly designed to satisfy physics through their cost function, there are also other ways to enforce compliance through the design of the machine learning model itself. Reference 21 notes how machine learning models can be designed to seamlessly transition into known boundary conditions through the use of heaviside-like functions. For PINN-GM-III, the following design change is proposed:

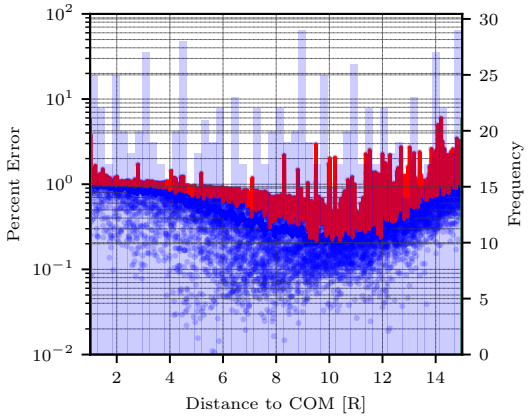
$$\hat{U}(r) = (1 - H(r))U_{NN}(r) + H(r)U_{BC}(r) \tag{8}$$



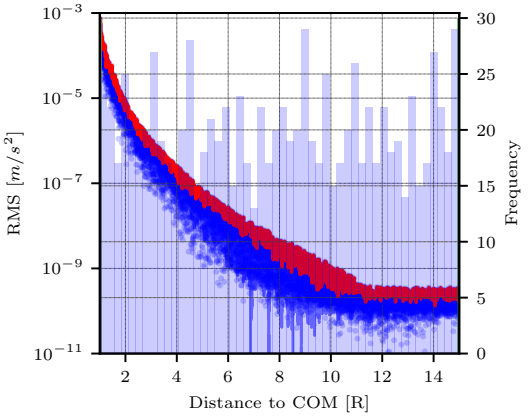
(a) Percent Error With $\mathcal{L}_{\text{RMS}}$ Function



(b) RMS Error With $\mathcal{L}_{\text{RMS}}$ Function



(c) Percent Error With $\mathcal{L}_{\%}$ Function



(d) RMS Error With $\mathcal{L}_{\%}$ Function

Figure 3: Different loss function change network performance at high- and low-altitudes. Blue points are the individual errors of the test data, the blue histogram is the distribution of training data, and the red line is the maximum error within a sliding window of 50 points.

where U_{NN} is the output of the neural network, U_{BC} is the potential at the boundary condition, and $H(r)$ is a heaviside-like function defined as

$$H(r) = \frac{1 + \tanh(k(r - r_{\text{ref}}))}{2} \tag{9}$$

where r is the radius of the field point, r_{ref} is a learned (via gradient descent) or user-prescribed radius, and k is a learned or user-prescribed smoothing parameter to control for a more continuous or discrete transition.

Equation 9 enforces that the PINN-GM must smoothly transition into a known boundary condition past a given altitude. In this way, the machine learning model can leverage the neural network solution in regions in which the boundary condition is irrelevant, but smoothly decrease the network's role as it approaches the boundary.

For the gravity modeling problem, there exists multiple ways in which this design choice can

manifest. In the limit of $r \to \infty$, it is clear that the potential decays to zero. However, setting $U_{BC} = 0$ and $r_{\text{ref}} = \infty$ in Equation 8 is not particularly practical as it demands the neural network must learn a model of the potential for the entire domain $r \in [0, \infty)$. An more useful choice is to leverage insights from the spherical harmonic gravity model and recognize that high frequency components of the gravity model decay to zero more quickly than the point mass contribution at high altitudes. I.e.

$$U_{BC}(r) = U_{LF} = \frac{\mu}{r} + \sum_{l=0}^{n} \sum_{m=0}^{l} \frac{\mu}{r} \left(\frac{R}{r}\right)^{l} (\ldots) \to 0 \tag{10}$$

$$\tag{11}$$

as $r \to \infty$.

This observation implies that $U_{BC}(r)$ can be set to $\frac{\mu}{r}$ assuming $r \gg R$. As such, the PINN-GM-III sets $U_{BC} = \frac{\mu}{r} + f(r)$ in Equation 8, where $f(r)$ are any higher order terms in the spherical gravity model that the user knows a-priori and wishes to leave as part of the boundary condition.

Leveraging Preexisting Gravity Information into PINN-GM Solution

In addition to enforcing the boundary condition through the model design, another design choice enabled by the PINN-GM-III is the ability to fuse an a-priori gravity model with the neural network solution. For example, most large celestial bodies exhibit planetary oblateness which is succinctly captured with the $C_{2,0}$ spherical harmonic coefficient as discussed in the methods section. Rather than requiring the network to relearn this prominent and easily observable perturbation, this information can be directly embedded into the network model. In this way, the PINN-GM-III can predict accelerations by leveraging a low-fidelity, first-order analytic model with a network responsible for capturing high-order perturbations through:

$$\hat{U}(r) = (1 - H(r)) \left(U_{LF}(r) + U_{NN}(r)\right) + H(r) \left(U_{LF}(r)\right) \tag{12}$$

where U_{LF} refers to the known, low-fidelity analytic model such as $U_{LF} = \frac{\mu}{r} + U_{J_2}$. Note that this choice must be carefully applied in the context of small bodies where harmonics models can break down inside the Brillouin sphere and produce instability in the model if the geometry is sufficiently non-spherical.

Preprocessing Efforts

It is generally considered good practice for the inputs and outputs of any neural network to be normalized.[22] This is because a) most neural network activation functions exhibit their greatest non-linearity between the bounds of -1 and 1 and b) this choice often leads to greater numerical stability during training. In traditional deep learning, the inputs and outputs of the network are typically normalized in a manner agnostic of one another; however, in physics-informed neural networks, where the inputs and outputs may share units, this decoupled normalization can produce non-compliant physics.

For the gravity field modeling problem, it is known that the positions share the distance coordinate (unit $[m]$) with the gravitational potential (unit $[m^2/s^2]$). Thus, by non-dimensionalizing or normalizing the position, one must simultaneously non-dimensionalized the potential accordingly.

To ensure that both the position inputs and potential outputs of the network exist within the numerically desireable domain of [-1,1], the following non-dimensionalization is proposed:

$$x^\star = R \tag{13}$$
$$U^\star = \max(U - U_{\text{LF}}) \tag{14}$$
$$t^\star = \sqrt{\frac{x^{\star 2}}{U^\star}} \tag{15}$$
$$a^\star = \frac{x^\star}{t^{\star 2}} \tag{16}$$
$$\tag{17}$$

where R is the maximum radius of the celestial body. Therefore to non-dimensionalize the position, potential, and accelerations $(\bar{x}, \bar{U}, \bar{a})$ the trivial equations can be used:

$$x = \frac{\bar{x}}{x^\star} \tag{18}$$
$$U = \frac{\bar{U}}{U^\star} \tag{19}$$
$$a = \frac{\bar{a}}{a^\star} \tag{20}$$
$$\tag{21}$$

Before the positions pass through the network, they undergo one additional preprocessing step. Like with PINN-GM-II, the cartesian coordinates are first converted to a 4D spherical coordinate description of (r, s, t, u) where r is the radius, and s, t, and u are the $\sin()$ of the angle between the field point and each cartesian axes.[23] This conversion ensures s, t, and u remain in the numerically desirable range of $[-1, 1]$. The radial coordinate, however, is not guaranteed these same conveniences. For field points that exist at infinite radii from the body, the r coordinate can scale from $[0, \infty]$ which can pose numerical challenges to the network. To circumvent these difficulties, the radial coordinate is inverted as $\frac{1}{r}$ to ensure that the bounds never exceed the domain of $[0, 1]$ and remain numerically stable for learning.

Rescaling the Output of the Neural Network

All gravitational potentials decay at high altitudes. As discussed, this decay can make it challenging to leverage RMS loss functions, but it also can pose other numerical difficulties for neural networks. For example, if one assumes that the largest potential of a gravity field is non-dimensionalized to magnitude 1, then field points at extremely high altitudes may have values that extend all the way down to machine precision i.e. U =1E-14. These stark differences in magnitude can lead to poor conditioning and learning for the neural network in its attempts to produce high-fidelity acceleration predictions in both high and low altitude regimes. To combat this, a new approach is taken for which the network learns a scale invariant form of the potential, U_{NN} which is later rescaled within the model to the proper order-of-magnitude through

$$\hat{U}_{\text{NN}} = \frac{U_{\text{NN}}}{\bar{r}^p} \tag{22}$$

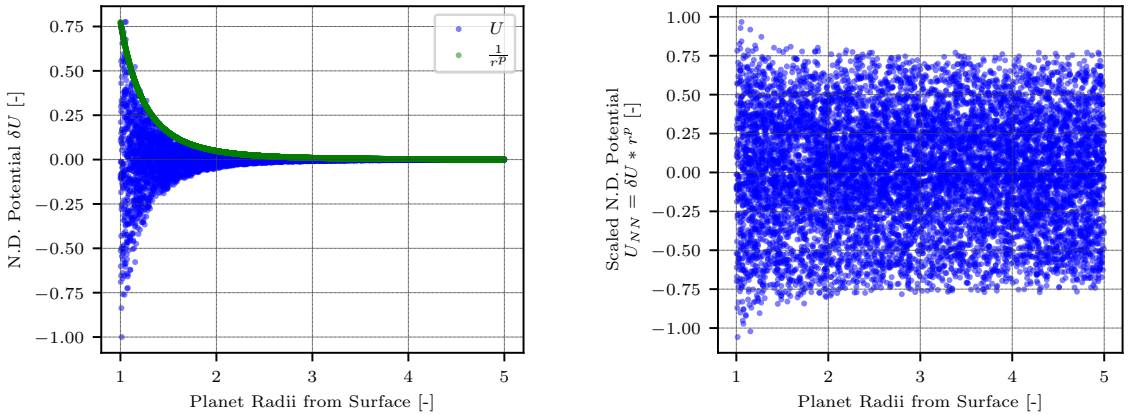
where $\hat{U}_{\text{NN}}$ is the potential produced by the entire PINN-GM-III model, U_{NN} is the scale invariant output of the neural network, $\bar{r}$ is the radius of the field point input into the network and normalized by the radius of the celestial body in question. Finally p is defined through

$$p = \begin{cases} l_r + 0 & l_r = -1 \\ l_r + 1 & l_r = 0 \vee 1 \\ l_r + 2 & l_r \geq 2 \end{cases} \tag{23}$$

where l_r is the maximum spherical harmonic degree included within the optional analytic part of the model, U_{LF}.

To visualize this natural decay and proposed scaling, Figure 4 shows the high-order potential perturbations ($\delta U = U - U_{\text{LF}}$ where U_{LF} is a degree $l = 2$ spherical harmonic model) of 10,000 test points within 0 - 5 radii from the Earth. The perturbations can be seen decaying at a rate of $\frac{1}{r^4}$. With Equation 22 applied to these potentials, they transition into a distribution that exists in a more numerically favorable domain of $[-1, 1]$ as seen in Figure 4b.

The utility of this scaling is shown with a simple test where two PINN-GM are trained. Both models are trained with 100 samples randomly distributed from 0 - $15R$ where R is the radius of the Earth. One network is trained without the scaling proposed in Equation 22 (Figure 5a) whereas the other is rescaled (Figure 5b). The RMS error is evaluated on 10,000 test points that span from 0-$150R$. Figure 5a clearly shows that the PINN-GM without scaling hits a numerical barrier at approximately $r = 15R$ where the magnitude of the predicted acceleration remains fixed and causes a constant error to be present in the gravity model. In contrast, the PINN-GM trained with the numerical scaling does not hit this barrier.



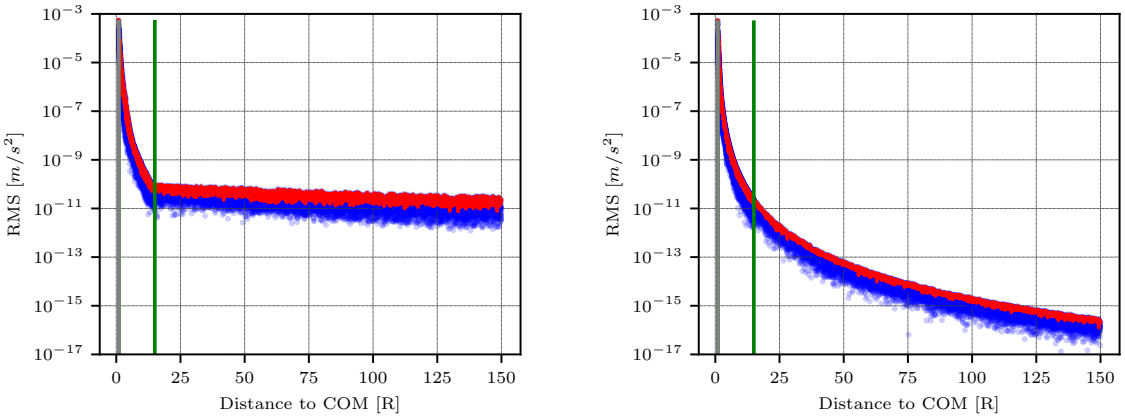
(a) Potential without Radial Scaling

(b) Potential with Radial Scaling

Figure 4: Rescaling the potential to exist within numerically well conditioned domain of $[-1, 1]$ assists with network convergence.

Future Optimization Efforts

Additional modifications and design improvements for PINN-GM-III are currently under development but are not yet considered stable. Later sections will explore the current status of these



(a) Potential without Radial Scaling

(b) Potential with Radial Scaling

Figure 5: Effect of rescaling the network output on the extrapolation of the network.

changes, but results should be taken as preliminary and further work must be performed before the conclusions can be taken at face value.

PINN-GM-III ON EARTH

The performance of the PINN-GM-III is tested using the Earth's static gravity field. The Earth's gravity field is typically represented with spherical harmonics, and the highest fidelity spherical harmonic model available is the Earth Gravity Model 2008 (EGM2008) which extends to degree and order 2160 (>4 million parameters).[24] Beyond the point mass and planetary oblateness terms, the largest remaining gravitational perturbations on Earth are features like mountain ranges and tectonic plate boundaries as can be seen in Figure 6. These discontinuous, small scale / high-frequency perturbations are one of the foundational motivations behind the PINN-GM. Spherical harmonics are ill-suited to capture these perturbations, requiring the superposition of thousands of harmonics before converging over these features due to effects like Gibbs phenomenon.[25] In contrast, neural networks are not bound by the prescribed, analytic scales of known harmonics and thus can learn more efficient basis function and ultimately require fewer parameters in their model.

Nominal Performance

The first generation PINN-GM has already been shown to offer advantages in modeling the Earth's gravity field over spherical harmonics in terms model compactness. The following tests investigate how much better the PINN-GM-III performs as compared to the original PINN-GM-I under different amounts of training data, training epochs, and model size.

Explicitly, for both model generation I and III, a set of 100 PINN-GMs are trained. The 100 networks are divided into four groups of 25. Each group of 25 is distinguished by number of nodes per hidden layer, N, drawn from the set $\{10, 20, 40, 80\}$. While each network maintains the same number of nodes, each of the 25 networks use a different combination of epochs drawn from the set $\{2^{10}, 2^{11}, 2^{12}, 2^{13}, 2^{14}\}$ and total training data from $\{2^{12}, 2^{13}, 2^{14}, 2^{15}, 2^{16}\}$. This experiment aims to highlight how much data is needed to converge on a solution and how long the networks must be trained to reach convergence for different model capacities.

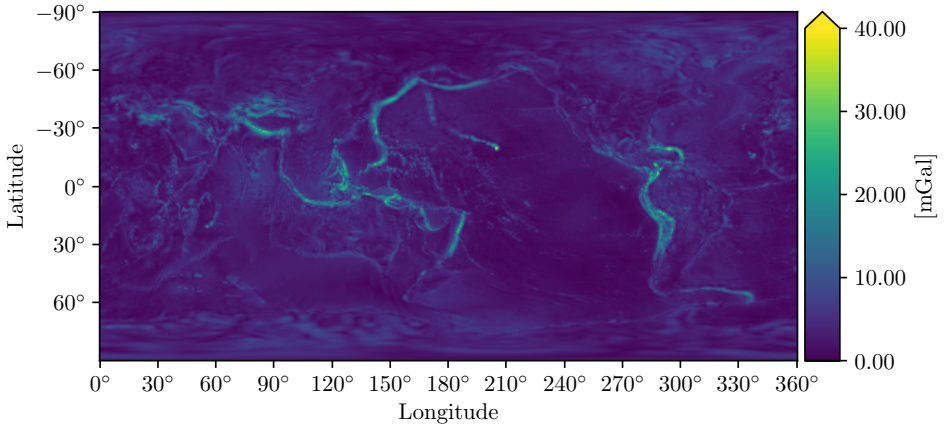


Figure 6: Remaining gravitational perturbations of Earth once point mass and planetary oblateness contributions are removed.

The training data for each network are distributed by selecting a random unit vector and multiplying it by a radius varied uniformly from an altitude of zero (the Earth's radius) to an approximately LEO altitude of 420 km. Once the field points are selected, their accelerations are computed using EGM2008 expanded to degree and order 1000 (approximately 1 million parameters). The default hyperparameters for the PINN-GM-III network are included in Table 1 and the default hyperparameters for PINN-GM-I can be found in Reference 10.

To evaluate the performance of each network, a mean percent error is computed on a separate test set of 50,000 randomly selected field points within the original training domain. Because the Earth's point mass and planetary oblateness contributions obscure the modeling accuracy of the higher-order perturbations, their contributions are removed before computing the percent error metric through

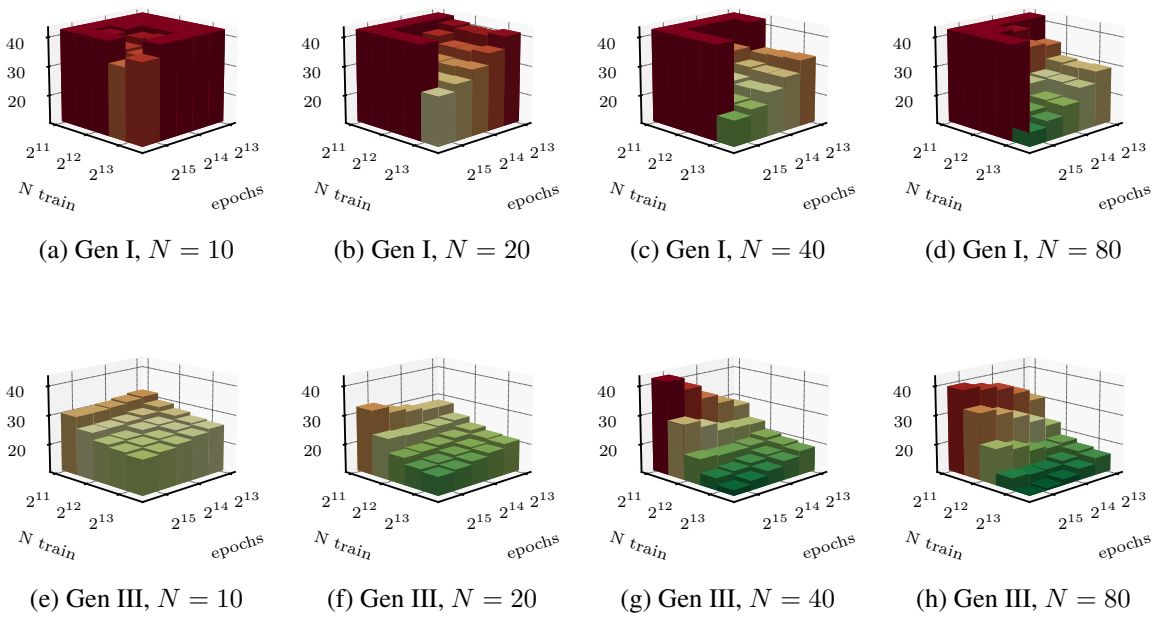
$$\mathcal{P} = \sum_{i=0}^{N} \frac{\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i}{\boldsymbol{a}_i - \boldsymbol{a}_{i,2}} \tag{24}$$

where $\hat{\boldsymbol{a}}$ are the network accelerations produced via $-\nabla U$ and $\boldsymbol{a}_{i,2}$ are the accelerations produced by the point mass and planetary oblateness terms of the spherical harmonic model.

Table 1: Default Hyperparameters for PINN-GM-III

Hyperparameter	Value	Hyperparameter	Value
Batch Size	2^{20}	Activation	GeLU
Learning Rate	0.001	Network Architecture	Transformer Inspired
Optimizer	Adam	Data Non-Dimensionalization	[Planet Radius, maximum U]
LR Scheduler	On Plateau	Preprocessing Layers	Pines $\rightarrow \frac{1}{r}$
Patience	2500 Epochs	Hidden Layer Weight Initializer	Glorot Normal
Loss Function	Percent	Final Layer Weight Initializer	Zeros

Figure 7 shows that there is a considerable jump in performance between the PINN-GM-I and the PINN-GM-III when modeling the Earth's gravity field. Foremost, PINN-GM-III models are capable



(a) Gen I, $N = 10$ (b) Gen I, $N = 20$ (c) Gen I, $N = 40$ (d) Gen I, $N = 80$

(e) Gen III, $N = 10$ (f) Gen III, $N = 20$ (g) Gen III, $N = 40$ (h) Gen III, $N = 80$

Figure 7: Average Percent Error of PINN-GM-I (top) and PINN-GM-III (bottom) trained on Earth.

of converging to a reliable, high accuracy solution with extremely small sets of data and relatively little training time – conditions for which PINN-GM-I is entirely unsuited to handle as seen in the $N = 10$ nodes per hidden layer case. Even once the PINN-GM-I has enough modeling capacity to converge on a solution ($N = 80$), the PINN-GM-III is able to produce a stronger model across all conditions.

One interesting result from Figure 7 is that for the high capacity models ($N = \{40, 80\}$), when there is insufficient data, the PINN-GM-III has an increased propensity for overfitting as can be seen by the increase in error given an increase in training epochs. While the overfitting of PINN-GM-III remains considerably less problematic than what is seen with the PINN-GM-I, this can be trivially remedied by adding a small amount of dropout to each layer, or incorporating the laplacian and curl physics constraints into the cost function.

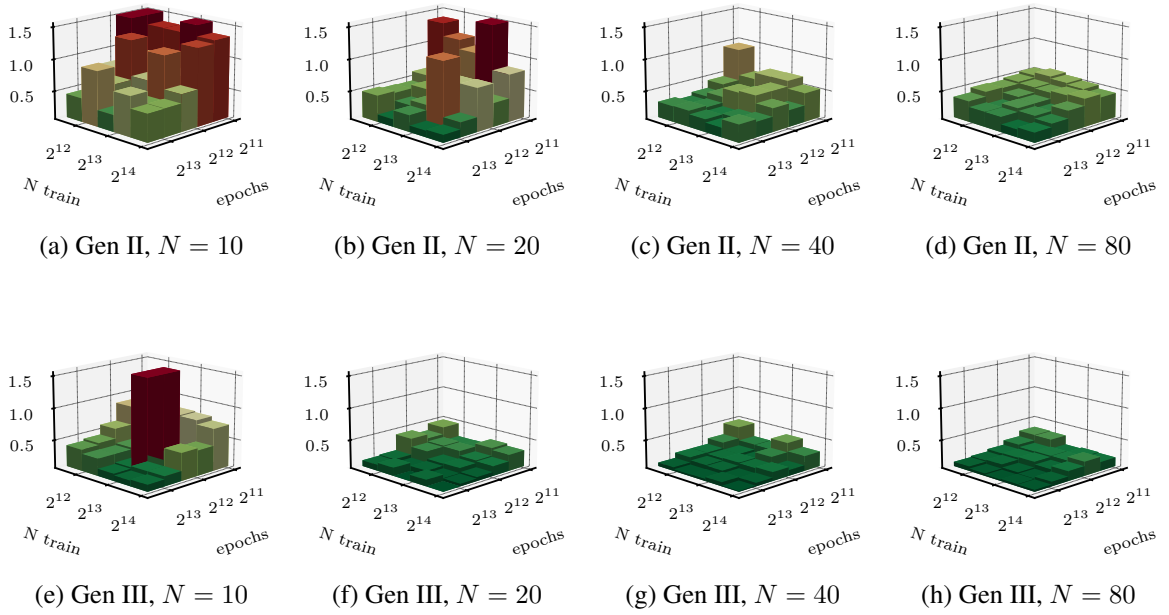
PINN-GM-III ON EROS

A second test is performed to investigate the improvements offered by the PINN-GM-III over its more recent predecessor PINN-GM-II. Specifically, this section investigates if the PINN-GM-III's benefits translate to modeling gravity fields produced by small-bodies, looking specifically at the performance when modeling the gravity field of the asteroid 433-Eros.

Nominal Performance

To investigate this, the same analysis performed on the Earth is repeated for Eros using PINN-GM-II in place of PINN-GM-I. Also, instead of using a spherical harmonic model, the truth accelerations are computed using a polyhedral model comprised of approximately 11,000 total facets and vertices, and the training data are distributed uniformly between the surface of the asteroid and a

maximum radius equal to 3 radii from the center of mass. The results of the trained PINN-GM-II and PINN-GM-III models are shown in Figure 8.



(a) Gen II, $N = 10$ (b) Gen II, $N = 20$ (c) Gen II, $N = 40$ (d) Gen II, $N = 80$

(e) Gen III, $N = 10$ (f) Gen III, $N = 20$ (g) Gen III, $N = 40$ (h) Gen III, $N = 80$

Figure 8: Average Percent Error of PINN-GM-II (top) and PINN-GM-III (bottom) trained on Eros.

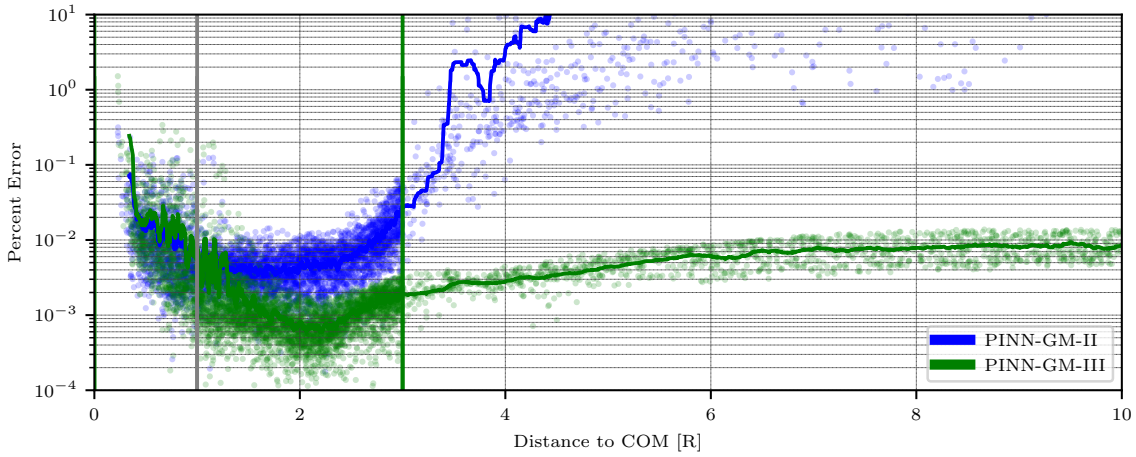
Again, the PINN-GM-III demonstrates consistent modeling superiority over its second generation predecessor. PINN-GM-II had considerably less stable and less accurate models in the low capacity regimes of $N = \{10, 20\}$ than PINN-GM-III. Even when training stabilized in the higher capacity PINN-GM-II instances, PINN-GM-III consistently achieved greater accuracy.

Beyond the accuracy within the training distribution, another second experiment showcases the difference in extrapolation error of the two different models. In this test, two networks are trained, a PINN-GM-II and PINN-GM-III each with 10 nodes per hidden layer and eight hidden layers. Each network is trained on 1,000 data points spanning 0-3R about the asteroid Eros. The average errors of both models are reported as a function of altitude from 0-10R in Figure 9a, and the XY cartesian plane cross sections are show in Figures 9b and 9c.

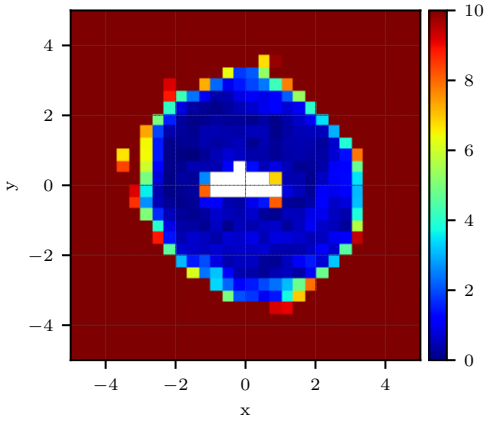
Note how only the PINN-GM-III is able to maintain low errors at high altitudes; whereas the PINN-GM-II quickly diverges beyond the bounds of the training data. This showcases the advantages of enforcing boundary conditions and scaling the network potential to ensure numerical accuracy, and offers the first demonstration that machine learning gravity models can be used globally and beyond the bounds of their training data.

EXPERIMENTAL MODIFICATIONS FOR PINN-GM-III

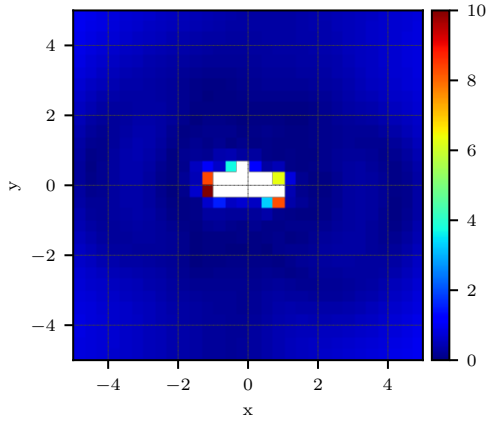
The experiments shown thus far quantify the performance of the stable version of the PINN-GM-III. Ongoing work explores additional modifications to the PINN-GM motivated by recent developments in the scientific machine literature. Experiments for these additional modifications are still ongoing, but their current status and motivation are detailed below.



(a) PINN-GM-II and III tested inside and outside (3R-10R) the training domain



(b) PINN-GM-II % Error



(c) PINN-GM-III % Error

Figure 9: Top Row: Percent error of PINN-GM inside (0-3R) and outside (3R-10R) of the training domain. Gray vertical line is Brillouin radius. Green vertical line is maximum bounds of training data. Bottom Row: % Error of PINN-GM in XY plane.

Avoiding Spectral Bias with Fourier Feature Mapping

A well-documented challenge faced by many neural networks is an effect called spectral bias.[26] The gradient flow dynamics of traditional and physics-informed neural networks are known to preferentially learn low-frequencies before high-frequencies during training. This bias towards lower frequencies can lead to excessively slow convergence rates when attempting to model signals with high-frequencies like those found in many gravity fields. One need only consider the discontinuous perturbations on the Earth or the many craters on the surface of the Moon to recognize such bias can be detrimental to training PINN-GMs efficiently.

Multiple research efforts aim to address this bias. One effort proposes the introduction of a fourier feature mapping layer to aid in making the neural tangent kernel stationary.[27] By randomly projecting the inputs of a neural network into fourier space, it is shown that the spectral bias can be

eliminated if the inputs are projected into the correct range of frequencies. Another effort extends these findings to demonstrate how multi-scale problems in scientific machine learning may require multiple projections into fourier space to efficiently learn both high and low frequency functions.[28] Separately, the Sinusoidal Representation Networks (SIRENs) was introduced which exclusively leverage sine functions for the network activations along with a principled initialization scheme to construct more compelling implicit neural representations of complex signals.[29] SIRENs are shown to produce continuously differentiable and higher accuracy representations of complex environments using less data than networks with other non-linear activation functions.

Tests for each of these modifications on the PINN-GM-III are still ongoing and are excluded for brevity. Of the analyses conducted thus far, the SIREN approach appears to offer the most benefit for the gravity problem but the magnitude is only marginal.

To demonstrate this, an experiment is proposed which seeks to produce a PINN-GM for the Moon's gravity field. The Moon's gravity field is of considerably higher complexity than that of the Earth,[10] and exhibits a wide range of both high and low frequency signals (see Figure 10a) making it an appropriate testbed to investigate the utility of the SIRENs approach.

This experiment trains two networks: first a stable PINN-GM-III model and second a PINN-GM-III with the proposed SIRENs activation / initialization scheme. Each network is trained on 100,000 data points distributed between the surface of the Moon and an altitude of 50 kilometers. Each PINN has 8 hidden layers and 20 nodes per hidden layer and was trained for 5,500 epochs. The resulting models are presented in Figure 10.

Both the vanilla PINN-GM-III and the PINN-GM-III with SIREN appear to produce models of comparable accuracy. The average percent error of the PINN-GM-III SIREN is 1% more accurate at the surface, and 7% more accurate across the entire training domain. Future work must investigate if these values are representative of optimal performance and if the SIRENS solution should be included in the final PINN-GM-III design.

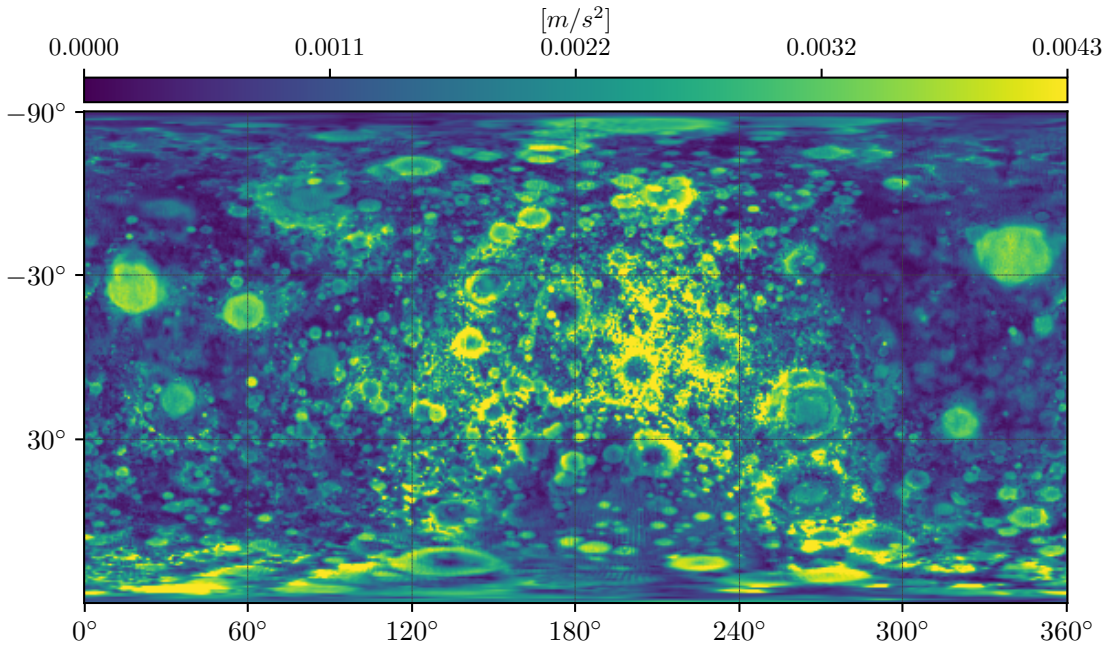
Learning Rate Annealing Algorithm

Another potential modification for the PINN-GM investigates how to better leverage a multi objective loss function. For PINN-GM-II, the physics informed loss function, Equation 5, was augmented with additional physics constraints as:

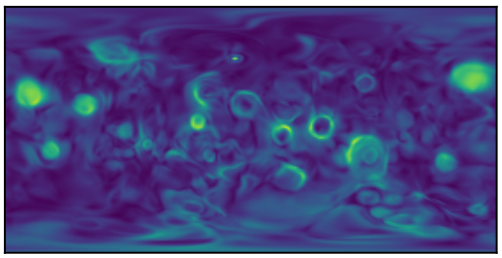
$$L(\theta) = \frac{1}{N}\sum_{i=0}^{N}\left|-\nabla\hat{U}(x_i|\theta) - \boldsymbol{a}\right|^2 + \left|\nabla^2\hat{U}(x_i|\theta)\right|^2 + \left|\nabla\times\nabla\hat{U}(x_i|\theta)\right|^2 \tag{25}$$

These additional physics constraints assist the networks in finding performant gravity models in the presence of noisy training data.[11] However, despite this additional robustness, these constraints also act as a form of regularization and can degrade performance of PINN-GMs slightly when trained on perfect position and acceleration measurements. Moreover, these constraints increase the amount of training time necessary for the PINN-GM to converge given the computational complexity of computing the second order derivatives of the potential. As such, for the tests conducted in this paper, these additional constraints are omitted from the cost function.

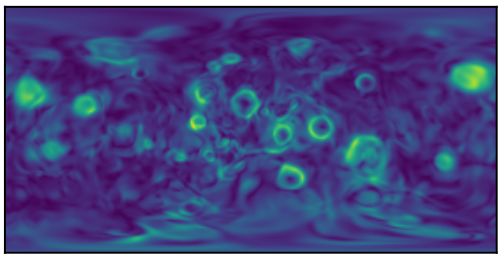
Despite this, there likely exists a way in which these additional constraints can return to the cost function and assist network performance even in the presence of perfectly measured training data. Specifically, a recent paper highlights that PINNs with multiple physics objectives in their loss



(a) True field



(b) PINN-GM-III



(c) PINN-GM-III SIREN

Figure 10: The resulting PINN-GM-III and III-SIREN networks trained on Moon's gravity field expanded to degree $l = 1,000$

function can have competing gradient flow dynamics – i.e. the different objectives have different learning behaviors which may prevent some objectives from being leveraged during training.[30] To combat this, a learning rate annealing algorithm is proposed which adaptively reweights the different components in the loss function to circumvent these difficulties. Future work for the PINN-GM will investigate the effect of this learning rate annealing algorithm on network performance and determine if its incorporation should be included in the final PINN-GM-III model.

CONCLUSIONS

Scientific machine learning and physics informed neural networks offer a compelling set of tools to address the gravity field modeling problem. While past generations of the PINN-GM have offered early glimpses into the potential advantages of this machine learning approach, considerable work remained to address the pitfalls prone to these machine learning systems. This paper demonstrates that with additional design considerations, many of these aforementioned challenges can be addressed and offer global improvements in performance in a broad range of training and network conditions. Future work will investigate the performance in the limit of large quantities of training data, data with uncertainty, and the effects of additional physics constraints include in the loss function.

REFERENCES

[1] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 378, Feb. 2019, pp. 686–707, 10.1016/j.jcp.2018.10.045.

[2] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning through Physics-Informed Neural Networks: Where We Are and What's Next," June 2022.

[3] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-Informed Machine Learning," *Nature Reviews Physics*, Vol. 3, May 2021, pp. 422–440, 10.1038/s42254-021-00314-5.

[4] W. M. Kaula, *Theory of Satellite Geodesy: Applications of Satellites to Geodesy*. Waltham, Mass.: Blaisdell Publishing Co, 1966.

[5] R. Werner and D. Scheeres, "Exterior Gravitation of a Polyhedron Derived and Compared with Harmonic and Mascon Gravitation Representations of Asteroid 4769 Castalia," *Celestial Mechanics and Dynamical Astronomy*, Vol. 65, No. 3, 1997, pp. 313–344, 10.1007/BF00053511.

[6] R. Furfaro, R. Barocco, R. Linares, F. Topputo, V. Reddy, J. Simo, and L. Le Corre, "Modeling Irregular Small Bodies Gravity Field via Extreme Learning Machines and Bayesian Optimization," *Advances in Space Research*, No. June, 2020, 10.1016/j.asr.2020.06.021.

[7] A. Gao and W. Liao, "Efficient Gravity Field Modeling Method for Small Bodies Based on Gaussian Process Regression," *Acta Astronautica*, Vol. 157, Apr. 2019, pp. 73–91, 10.1016/j.actaastro.2018.12.020.

[8] L. Cheng, Z. Wang, Y. Song, and F. Jiang, "Real-Time Optimal Control for Irregular Asteroid Landings Using Deep Neural Networks," *Acta Astronautica*, Vol. 170, May 2020, pp. 66–79, 10.1016/j.actaastro.2019.11.039.

[9] D. Izzo and P. Gómez, "Geodesy of Irregular Small Bodies via Neural Density Fields: geodesyNets," May 2021.

[10] J. Martin and H. Schaub, "Physics-Informed Neural Networks for Gravity Field Modeling of the Earth and Moon," *Celestial Mechanics and Dynamical Astronomy*, Vol. 134, Apr. 2022, 10.1007/s10569-022-10069-5.

[11] J. Martin and H. Schaub, "Physics-Informed Neural Networks for Gravity Field Modeling of Small Bodies," *Celestial Mechanics and Dynamical Astronomy*, Sept. 2022, p. 28.

[12] M. Brillouin, "Équations Aux Dérivées Partielles Du 2e Ordre. Domaines à Connexion Multiple. Fonctions Sphériques Non Antipodes," Vol. 4, 1933, pp. 173–206.

[13] J. R. Martin and H. Schaub, "GPGPU Implementation of Pines' Spherical Harmonic Gravity Model," *AAS/AIAA Astrodynamics Specialist Conference* (R. S. Wilson, J. Shan, K. C. Howell, and F. R. Hoots, eds.), Virtual Event, Univelt Inc., 2020.

[14] D. J. Scheeres, B. Khushalani, and R. A. Werner, "Estimating Asteroid Density Distributions from Shape and Gravity Information," *Planetary and Space Science*, Vol. 48, No. 10, 2000, pp. 965–971, 10.1016/s0032-0633(00)00064-7.

[15] D. J. Scheeres, A. S. French, P. Tricarico, S. R. Chesley, Y. Takahashi, D. Farnocchia, J. W. McMahon, D. N. Brack, A. B. Davis, R. L. Ballouz, E. R. Jawin, B. Rozitis, J. P. Emery, A. J. Ryan, R. S. Park, B. P. Rush, N. Mastrodemos, B. M. Kennedy, J. Bellerose, D. P. Lubey, D. Velez, A. T. Vaughan, J. M. Leonard, J. Geeraert, B. Page, P. Antreasian, E. Mazarico, K. Getzandanner, D. Rowlands, M. C. Moreau, J. Small, D. E. Highsmith, S. Goossens, E. E. Palmer, J. R. Weirich, R. W. Gaskell, O. S.

Barnouin, M. G. Daly, J. A. Seabrook, M. M. Al Asad, L. C. Philpott, C. L. Johnson, C. M. Hartzell, V. E. Hamilton, P. Michel, K. J. Walsh, M. C. Nolan, and D. S. Lauretta, "Heterogeneous Mass Distribution of the Rubble-Pile Asteroid (101955) Bennu," *Science Advances*, Vol. 6, No. 41, 2020, 10.1126/sciadv.abc3350.

[16] M. T. Zuber, D. E. Smith, A. F. Cheng, J. B. Garvin, O. Aharonson, T. D. Cole, P. J. Dunn, Y. Guo, F. G. Lemoine, G. A. Neumann, D. D. Rowlands, and M. H. Torrence, "The Shape of 433 Eros from the NEAR-Shoemaker Laser Rangefinder," *Science*, Vol. 289, No. 5487, 2000, pp. 2097–2101, 10.1126/science.289.5487.2097.

[17] M. K. Shepard, B. Timerson, D. J. Scheeres, L. A. Benner, J. D. Giorgini, E. S. Howell, C. Magri, M. C. Nolan, A. Springmann, P. A. Taylor, and A. Virkki, "A Revised Shape Model of Asteroid (216) Kleopatra," *Icarus*, Vol. 311, 2018, pp. 197–209, 10.1016/j.icarus.2018.04.002.

[18] J. K. Miller, A. S. Konopliv, P. G. Antreasian, J. J. Bordi, S. Chesley, C. E. Helfrich, W. M. Owen, T. C. Wang, B. G. Williams, D. K. Yeomans, and D. J. Scheeres, "Determination of Shape, Gravity, and Rotational State of Asteroid 433 Eros," *Icarus*, Vol. 155, No. 1, 2002, pp. 3–17, 10.1006/icar.2001.6753.

[19] Y. Lecun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444, 10.1038/nature14539.

[20] A. G. Baydin, B. A. Pearlmutter, and J. M. Siskind, "Automatic Differentiation in Machine Learning: A Survey," *Journal of Machine Learning Research*, Vol. 18, 2018, pp. 1–43.

[21] Q. Zhu, Z. Liu, and J. Yan, "Machine Learning for Metal Additive Manufacturing: Predicting Temperature and Melt Pool Fluid Dynamics Using Physics-Informed Neural Networks," Sept. 2020.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[23] S. Pines, "Uniform Representation of the Gravitational Potential and Its Derivatives," *AIAA Journal*, Vol. 11, Nov. 1973, pp. 1508–1511, 10.2514/3.50619.

[24] N. K. Pavlis, S. A. Holmes, S. C. Kenyon, and J. K. Factor, "The Development and Evaluation of the Earth Gravitational Model 2008 (EGM2008)," *Journal of Geophysical Research: Solid Earth*, Vol. 117, Apr. 2012, pp. n/a–n/a, 10.1029/2011JB008916.

[25] E. Hewitt and R. E. Hewitt, "The Gibbs-Wilbraham Phenomenon: An Episode in Fourier Analysis," *Archive for History of Exact Sciences*, Vol. 21, No. 2, 1979, pp. 129–160, 10.1007/BF00330404.

[26] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, "On the Spectral Bias of Neural Networks,"

[27] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, "Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains,"

[28] S. Wang, H. Wang, and P. Perdikaris, "On the Eigenvector Bias of Fourier Feature Networks: From Regression to Solving Multi-Scale PDEs with Physics-Informed Neural Networks," *Computer Methods in Applied Mechanics and Engineering*, Vol. 384, Oct. 2021, p. 113938, 10.1016/j.cma.2021.113938.

[29] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit Neural Representations with Periodic Activation Functions," June 2020.

[30] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and Mitigating Gradient Pathologies in Physics-Informed Neural Networks," *arXiv*, Jan. 2020, pp. 1–28.