INVESTIGATION OF THE ROBUSTNESS OF NEURAL DENSITY FIELDS

Jonas Schuhmacher[(1)], **Fabio Grat**[(1)], **Dario Izzo**[(2)], **Pablo Gómez**[(2, 3)]

[(1)] *Technische Universität München, Arcisstraße 21, 80333 München, Germany, {jonas.schuhmacher, f.gratl}@tum.de*
[(2)]*Advanced Concepts Team, European Space Agency, European Space Research and Technology Centre (ESTEC), Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, {pablo.gomez, dario.izzo}@esa.int*
[(3)]*AI Sweden, Lindholmspiren 11, 417 56 Göteborg, Sweden*

ABSTRACT

Recent advances in modeling density distributions, so-called neural density fields, can accurately describe the density distribution of celestial bodies without, e.g., requiring a shape model - properties of great advantage when designing trajectories in close proximity to these bodies. Previous work introduced this approach, but several open questions remained. This work investigates neural density fields in the context of robustness to external factors like noise or constraints during training, like the maximal available gravity signal strength due to a certain distance. Thus, this work studies the relative errors of 2824 trained neural networks for the two bodies 433 Eros and 67P/Churyumov-Gerasimenko with varying training conditions. Models trained with mascon or the polyhedral ground truth without any noise perform similarly in the experiments with relative errors below 5% until $100\,m$ distance for training in the near-range. The impact of solar radiation pressure on a typical probe affects training neglectably, with the relative error being of the same magnitude as without noise. Limiting the training point accuracy to $1\,mGal$ severely hurts the obtainable precision when only sampling above several kilometers of height. However, inducing a relative error of 10% stills leads to acceptable results, with errors below 5% in the sampling range used for training and more than 50 times away. Finally, pretraining is shown as practical in order to speed up network training. Hence, geodesyNet is a helpful tool extending the toolbox of assembling a navigation model for a small body.
All code and results are publicly available at `https://github.com/gomezzz/geodesyNets`

1 INTRODUCTION

In recent years small bodies in our solar system have been of increasing interest as mission targets. Various past, present, and future missions visited them and even collected samples allowing study of their composition and properties. For example, in the past, NEAR visited the asteroid 253 Mathilde and later soft landed on 433 Eros in February 2001, or Rosetta, with its lander Philae visiting the comet 67P/Churyumov–Gerasimenko in 2014. There are also several upcoming missions such as Hera, ZhenHe, and Psyche [1]. Hera, the complementary mission of ESA to NASA's DART mission, will analyze the latter's impact with the smaller moon Dimorphos, which revolves around the asteroid Didymos, with the goal of studying the process of asteroid deflection. It is scheduled to start in 2024 and rendezvous the binary system in 2026 together with two accompanying cubesats [2].

With these prospects, it is even more crucial for guidance, navigation and control to have accurate and precise models of these target bodies. However, this is especially difficult with small bodies since of their irregular shape. To complicate matters, the density distribution of these bodies is also rarely homogeneous but rather heterogeneous in nature [3, 4, 5].
These facts make modeling these bodies with the established three models difficult: spherical harmonics, mascon models and polyhedral gravity models. The former struggle with the convergence inside the Brillouin sphere, and the irregular shapes slow down convergence, making it unsuitable for asteroids and comets [1, 6, 7]. The minimum Brillouin sphere is the sphere centered around the origin of the body and still enveloping all of the body's mass. The latter two require the knowledge of the shape model of the target body, and their application comes with constraints like discretization [8] in the case of the mascon model or the assumption of homogeneous density in the case of polyhedral gravity models [9, 10].
With the recent advances in neural networks, a new approach has appeared, requiring no previous assumptions but only existing data originating from measurements or synthetic sources. With this data, one can train a neural network to solve the gravity inversion problem. Multiple approaches exist for creating such a network, as presented in Section 2. This paper focuses on geodesyNets —a network learning the density distribution of a body whose integration leads to gravity [1]. Previous work showed acceptable error boundaries for models trained with a synthetic ground truth based on a mascon model. Further, the approach was also successfully employed for describing the density of asteroid (101955) Bennu and validated using the data of the OSIRIS-REx mission [11].
This work aims to determine if this approach is imaginable in a practical onboard scenario, given that multiple sources of perturbations can pollute the gravity signal used for training, ranging from non-gravitational forces like solar radiation pressure to measurement inaccuracies. Further, the utilized gravity signal might be weaker due to safety distance when on a trajectory far from the celestial body, further hindering the network from learning correctly. Last, it is investigated if pretraining on a prior shape model could improve the network performance under these constraints and if it reduces the amount of training iterations needed, thus reducing computational cost in a hypothetical onboard scenario.
This work uses the polyhedral gravity model and the mascon model for conducting these experiments. It demonstrates the applicability of geodesyNets trained with a noisy input signal, as long as the induced error is reasonably sized compared to the gravity signal. This means that an error relative to the magnitude of the gravity signal in the training data set still leads to acceptable errors. On the other hand, a large absolute limit on measurement accuracy can render any training useless. This circumstance exemplifies in the conducted study about the sampling distance utilized for training the network. Finally, it is shown how pretraining can reduce the number of iterations required for training. All results and code are made publicly available via GitHub.

2 RELATED WORK

2.1 Polyhedral Gravity Models

Polyhedral gravity models can calculate the full gravity tensor, including potential, acceleration, and second derivatives for an arbitrary given point P around a polyhedral source. They provide an analytical solution given homogeneous density distributions given a shape model. One of the approaches to implementing a polyhedral gravity model is given by Tsoulis et al. [9, 10]. They use a line integral approach to convert the triple integral into a nested summation, which, thanks to the introduction of dedicated singularity avoids potential singularities that can affect these models [12, 13]. In this work, we use Tsoulis model to compare the performance of a geodesyNet trained with a mascon model to a

network trained with a polyhedral model, but also to determine how the mesh granularity affects the achievable precision. Furthermore, we perform a detailed study of the achievable precision at close range since the high accuracy, even within the Brillouin sphere, and the analytical formulation does not require numerical integration methods.

2.2 Mascon Models

Mascon models are the second approach for modeling gravity utilized in this work. Here, the body is represented as a combination of multiple point masses, so-called mascons (short for "mass concentrations" [14]) filling up its volume. The mascon elements do not need to be of uniform size. Instead, various approaches exist, combining mascons of different weights [8]. Its simplicity and ability to model irregular shapes and density distributions as they appear for small bodies make the model appealing. Regardless, high accuracy of the gravity field can only be achieved with an extensive number of mascons. Even then, the field's accuracy near the body's surface remains challenging due to the discretized mass distribution [8].

2.3 Gravity Modeling with Neural Networks

Artificial intelligence emerging in almost any area in recent years provides new opportunities in modeling traditionally expensive computational processes in physics [15, 16]. One such domain is modeling the gravity of irregularly shaped bodies and mapping positions to accelerations. In this context, one can distinguish between neural networks representing the actual body, enabling an indirect mapping to gravitational accelerations, and those directly mapping positions around a given body to gravitation. Thus, the former can also be referred to as neural fields since they parameterize the bodies [16]. A geodesyNet initially described by [1] represents a body through its neural density field.
The inspiration for geodesyNets originates from Neural Radiance Fields (NeRF) introduced by [17]. Their network learns to represent a three-dimensional object from two-dimensional pictures. Thus, their network maps a 5D input vector for position and view angle to a 2D color and volume density vector. Similarly how they solve the inverse problem of image rendering, a geodesyNet solves the inverse problem of gravity inversion mapping a cartesian vector to a candidate body density [1]. The neural field of a geodesyNet represents the body's density distribution.
The subsequent Subsection 3.2 will describe it in detail since it is the object of study, whereas this section provides a brief look over alternatives from the first category of networks.
Generally, all of the following presented networks have in common that they directly map coordinates to potentials or accelerations and are trained on polyhedral ground truth. However, the concrete approaches vary a lot. Furfaro et al. [7] use Single-Layer Feedforward Networks using Extreme Learning, reducing the number of required iterations to fine-tune their model to map the relationship between spacecraft points and gravitational acceleration. They report relative errors (relRMSE) below 5% for asteroid 25143 Itokawa and only slightly above 5% for comet 67/P Churyumov-Gerasimenko. They conducted two experiments. One time the global gravity field was learned using target points sampled from a near-range sphere around the body. The second time a local field was learned using target points from a cylinder above a given landing zone. This approach contrasts this work as here the robustness is examined when also only sampling from far away.
Cheng et al. [18] use deep neural networks (DNN) also with the aim of generating trajectories in a computationally cheap way for a soft landing on small bodies. They show that the median error of their model is 0.33% when taking the mean over the three axial directions. Given that 1% deviations are usually acceptable in traditional gravity models [18], they conclude that their DNN is a practical approach. Similarly to the later employed geodesyNet, they normalize the input and outputs to the

range $(-1, 1)$. Further, they do not train the network with sample points from the near field but instead with points some kilometers away from the body. In their case study for 433 Eros, points are collected from the sphere $3\,km$ to $60\,km$.
Finally, Gao et al. [19] model the gravity field of multiple sample bodies with a Gaussian Process Regression. They report a relative mean error of 1.27% for their bodies when validating inside the sampling area. They only train their networks with samples including the near range of the bodies. However, in contrast to the study presented later on geodesyNet, their approach shows strong generalization difficulties when further away from the body. Thus, the relative error increases beyond 60% in the case of 433 Eros at $40\,km$ distance. For comparison: Training was done here with points up to $20\,km$ distance.
Martin et al. [20, 21, 22] also present a model that maps Cartesian coordinates to gravity. However, their approach differs from the usual training in the sense that the network is bound to additional constraints beyond the usual loss to guarantee physical correctness. These constraints are integrated into the loss function and include, for example, the satisfaction of Laplace's equation. The training thus penalizes not only inaccuracy but also physical violations of the underlying differential equations. They also train on 433 Eros with sampling up to a distance of three times the radius. The network generalizes well to distance, but has difficulties predicting closer to the asteroid.

3 METHODS

3.1 Ground Truth Models

Equation 1 shows the derived formalism for computing the gravitational acceleration around a polyhedral source at the origin with G as the gravitational constant, ρ as the constant density using the polyhedral gravity model by [9, 10]. It consists of two summations with the outer summing over the polyhedral faces p and the inner one iterating over the segments q of each face. For a detailed description, refer to [9, 10].

$$\overrightarrow{a} = G\rho \cdot \sum_{p=1}^{n} \overrightarrow{N_p} \cdot \left[\sum_{q=1}^{m} \sigma_{pq} h_{pq} L N_{pq} + h_p \sum_{q=1}^{m} \sigma_{pq} A N_{pq} + \text{sing}_{\mathcal{A}_p} \right] \tag{1}$$

This polyhedral gravity model was recently implemented in a work preceding this one. For implementation details, refer to [23].
This model will be contrasted with the mascon model, whose formula is displayed in Equation 2 for a generic point $\overrightarrow{r_i}$ and a set of mascons $\mathcal{M} = \{(x_i, y_i, z_i)\ i = 1..n\}$, in the following experiments. The aim is to determine the achievable precision with the knowledge that the Mascon model, used in the original paper, has difficulties in the immediate proximity to the surface due to discretization. Next to the calculation of the ground truth is the evaluation of the model at the given points of the batch. For this purpose, the neuronal density field is integrated using Equation 4. After the models were always trained with vectorial accelerations and not scalar potentials, demonstratively, only these formulas are displayed.

$$\overrightarrow{a}(\overrightarrow{r_i}) = -G \sum_{j=1}^{n} \frac{m_j}{r_{ij}^3} r_{ij} \tag{2}$$

3.2 GeodesyNets - Neural Density Fields

A neural density field, also referred to as geodesyNet [1, 15], is a fully-connected neural network trained with a gravity signal from an arbitrarily shaped body with the aim of learning the body's density distribution. Thus, it solves the gravity inversion problem and provides a fully differentiable expression for mapping a cartesian point to a candidate body density (see Equation 3) compatible with the observed gravity signal.

$$f(x, y, z) \rightarrow \rho \tag{3}$$

The appeal of this method is that it does not require a shape model, as is the case with traditional gravitational models, but that the gravitational signal can come from any source.
Using the trained model, a numerical integration over the neuronal density field can be performed to calculate the potential or the acceleration. This procedure is displayed in Equation 4.

$$\overrightarrow{a} = G \int_{x \in V} \frac{\rho(x)}{|\overrightarrow{r} - \overrightarrow{x}|^3} (\overrightarrow{r} - \overrightarrow{x}) \, dV \tag{4}$$

Previous studies utilized this training approach in combination with the mascon model as ground truth, demonstrating its theoretical applicability [1]. The measured relative error was always less than 1% even near bodies. There, however, it was not considered how robust the network would be if the sample points were only sampled with some distance like a spacecraft would do due to safety measurements.
GeodesyNets have also been looked at from a practical point of view with measured gravity signals obtained from the OSIRIS-REx mission, demonstrating the practical applicability of the given approach while also giving first insights above limitations of the approach given the noise in a measured gravity signal [11].
This work now provides a detailed study of the imaginable disruptive factors which could lead to wrong model behavior, including noise in the ground-truth measurements or errors in the numerical integration. As well as the influence that the distance has on the possible precision.

3.3 Sampling Points for Training

Sampling is also one of the variables in the following experiments. The spherical envelopes chosen for training were $(0, 1)$, $(1, 3)$, and $(3, 5)$. The former covers the close range around the bodies, which the original study used for sampling. Here now, two more sampling distances are included. The mid-range $(1, 3)$ represents the distance a lander would reach during the approach. The far-range $(3, 5)$ represents a realistic distance an orbiter could reach around the body. The latter distances are the more realistic scenarios with respect to onboard training, as they maintain a certain safety distance for the spacecraft. In contrast, the close distance $(0, 1)$ would only be practical in a gravity-model-based (pre-)training. In the following, it is refered to $(0, 1)$, $(1, 3)$, and $(3, 5)$ as close-, mid- and far-range.
The utilized meshes for Eros with a resolution of 100% comes from [24], and the one for Churyumov-Gerasimenko from [25]. Eros consists of 7374 vertices and 14744 triangular faces, whereas Churyumov-Gerasimenko comprises 9149 vertices and 18294 triangular faces. Both are based on the measurements of the probes that visited the bodies. The Macon models used for both models are from [1] and are derived from the above 100% resolution meshes. For this purpose, a centroid with mass m_j is placed in each tetrahedron in the delaunay tetrahedralized version. In the course of this study, additional meshes were constructed. These are downsampled versions with 10%, 1%, or 0.1% of the vertices and faces. The purpose of this is to investigate to what extent a model can be (pre-)trained, even with possibly insufficient ground truth. All meshes are normalized versions so that the bodies fit

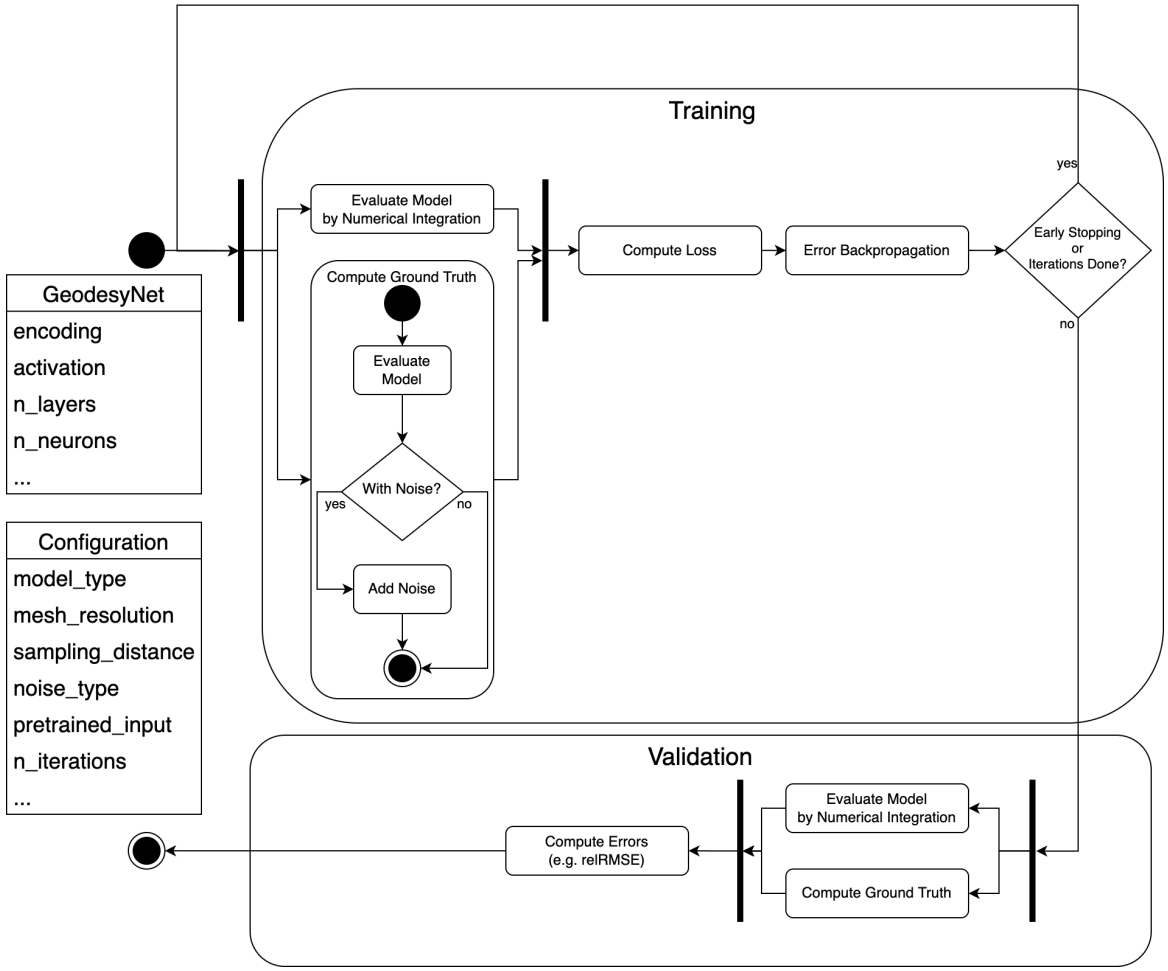
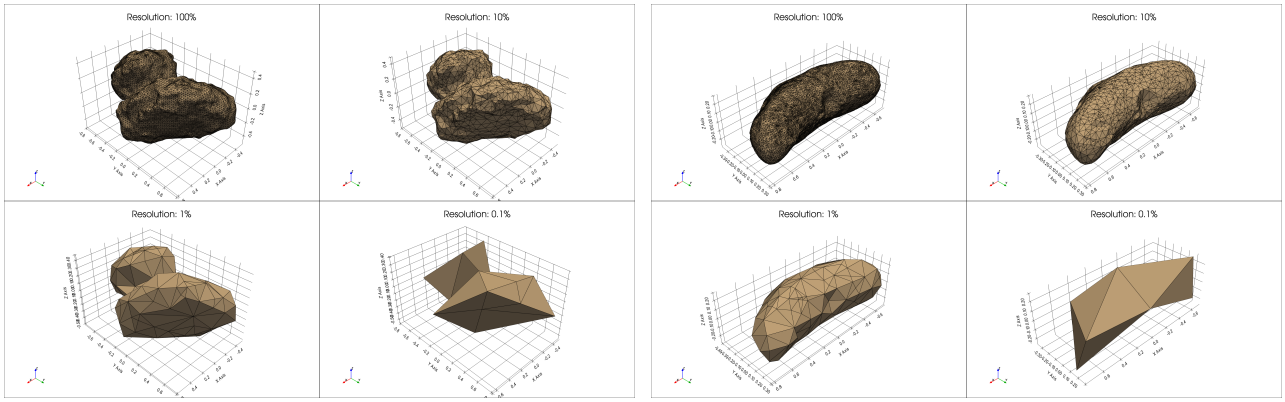


Figure 1: UML Activity Diagram of the training and validation toolchain. Further, the two main objects involved are shown: The geodesyNet itself and the parameterized configuration for the experiments. Both of them just show the relevant aspects utilized in this work.

in the hypercube $[-1, 1]^3$. Figure 2 illustrates the employed input meshes for the polyhedral gravity model.



(a) Churyumov-Gerasimenko (b) Eros

Figure 2: The original high resolution mesh denoted with 100% and downscaled to meshes with respectively only 10%, 1% or 0.1% of vertices and faces.

3.4 Addition of Noise to the Ground Truth

In addition to the ground-truth model, mesh resolution, and sampling distance, this study has one more parameter: noise. This characteristic is applied to the ground truth during the training. It thus represents that the measurements, the accelerations used for training, are subject to distortion factors. Three different types are investigated in the course of this study: a constant bias, additive Gaussian noise, and multiplicative Gaussian noise.
The constant bias represents the possible effects of solar radiation pressure (SPR) on the measurement results. In a previous study [3, 11], this was a significant factor influencing the quality of the results. Although the SPR can be subtracted out, this only works if the mass and size of the object are known. Here, we will determine to what extent SRP influences the training. Equation 5 [26] shows the employed approach for calculating the acceleration affecting the spacecraft with P as the solar radiation pressure, c as the speed of light and G_{SC} as the solar constant, and R as the distance from the sun in astronomical units $[AU]$. Equation 6 calculates the acceleration a given the area-to-mass ration of the spacecraft A/m.

$$P = \frac{G_{SC}}{c \cdot R^2} \tag{5}$$

$$a = P \cdot \frac{A}{m} \tag{6}$$

These equations are a simplified version of the calculation. Theoretically, the reflectivity, as given in [27], could also be included in the calculation. For this work, however, having only a baseline of the magnitude is sufficient. The calculated acceleration a is then applied in one cartesian direction, assuming that the sun always shines from this direction. In the following experiments, a value was used for the SRP as it could realistically affect a probe analogous to Rosetta. The mass was assumed to be $1422\,kg$ (roughly Rosetta Dry Mass and Payload) and the area $69.88\,m^2$. For the distance, the semi-major axis of the bodies was used in each case. It should be added that the SRP for a cubesat would have a value of the same magnitude due to its smaller mass but equally smaller area.
Further, an additive Gaussian noise $X \sim \mathcal{N}(0, \sigma^2)$ is used during the training to simulate an absolute error in the measurement of the gravitational signal. The standard deviation σ is chosen in a way that it simulates an absolute error of $10^{-4}\frac{m}{s^2}$ or $10^{-5}\frac{m}{s^2}$. After the meshes are normalized, this standard deviation has also been normalized. The value 10^{-5} is the accuracy of GOCE (Gravity field and steady-state ocean circulation explorer) [28]. In addition to this value, 10^{-4} was chosen as the absolute precision to get an insight into how strongly one additional error magnitude affects the result.
Further, a multiplicative Gaussian noise $X \sim \mathcal{N}(1, \sigma^2)$ is used during the training to simulate a relative error in the measurement of the gravitational signal. The assumption is that the magnitude of the gravitational signal is known, but the value can only be determined to a certain point. For example that the precision is limited relatively to the magnitude of the accelerations to 10^{-x} with $x \in \{1, 2, 3\}$.

3.5 Training

The toolchain used for training and performance evaluation is shown in Figure 1.
This summarizes the characteristics of the training iterations. Overall, the same configuration is used for all other parameters as in the original paper [1]. In particular, this includes the Mean Absolute Error calculated from the ground-truth y and the model predictions $\hat{y}$ shown in Equation 7. Here, κ (see Equation 8) is a scaling parameter to normalize the mass and restrict training to learning the volume rather than finding the absolute mass [1].

$$\mathcal{L}_{\kappa\,MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \kappa\hat{y}_i| \tag{7}$$

$$\text{with } \kappa = \frac{\sum_{i=1}^{n}\hat{y}_i y_i}{\sum_{i=1}^{n} y_i^2} \tag{8}$$

3.6 Validation

For validation, the relative root mean squared error is used, which is calculated as shown in Equation 9. It is calculated for each model at the end of training for a range of distances from the body's surface. Comparisons are made against the polyhedral ground-truth using the 100% mesh resolution. The altitude sampler employed for this purpose uses the outward normal of the mesh faces to sample points at the appropriate altitude. So the distance in the validation plot's x-axis is scaled by the altitudes from the surface, not the altitudes from the mathematical origin of the bodies.

$$relRMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \kappa\hat{y}_i|}{||\hat{y}_i||_2}\right)^2} \tag{9}$$

4 RESULTS

All experiments follow Section 3 presented toolchain for training and validation except for the last one involving pretraining. One can imagine this last experiment as chaining of Figure 1 twice. So the models' weights are not randomly initialized but with the weights of a trained model. For each scenario presented here, ten training runs have been performed. The graphs show the mean value of the validation results and the standard deviation in a slightly transparent way. The batch size has been set to 1000 points, and every ten training iterations, a new batch has been generated. For this purpose, a spherical point sampler randomly generates points in a spherical shell that do not lie within the sample body.

4.1 Polyhedral vs. Mascon Model

Figure 3 compares the relative errors of the models trained with a mascon model and a polyhedral model for the two studied bodies. Spherical sampling was performed in a shell $\in (0, 1)$, but the results are similar for sampling with more distant shells. No qualitative difference can be found between models trained with the mascon or polyhedral models, not even in the close range around the body.

For Churyumov-Gerasimenko, the relative errors between the models only differ in the first significant digit of the results if they differ. E.g. for distance 0.01, the model's errors are equal at 2.2%, whereas for distance 0.001, the mascon models has a mean error of 6.1% compared to the polyhedral trained models' relative mean error of 5.8%.

For Eros, although the average relative error is slightly lower for the polyhedral model, it is not by order of a full magnitude. The model's relRMSE trained with the polyhedral model is only around 50% to 70% compared to the relRMSE of the mascon trained models through all validation distances.

Figure 4 shows the achievable precision for the polyhedral model depending on the precision of the mesh used. A model trained with a mesh with only one-tenth of the vertices and faces achieves a comparable performance to the entire 100% resolution mesh. Also interesting is that a geodesyNet trained

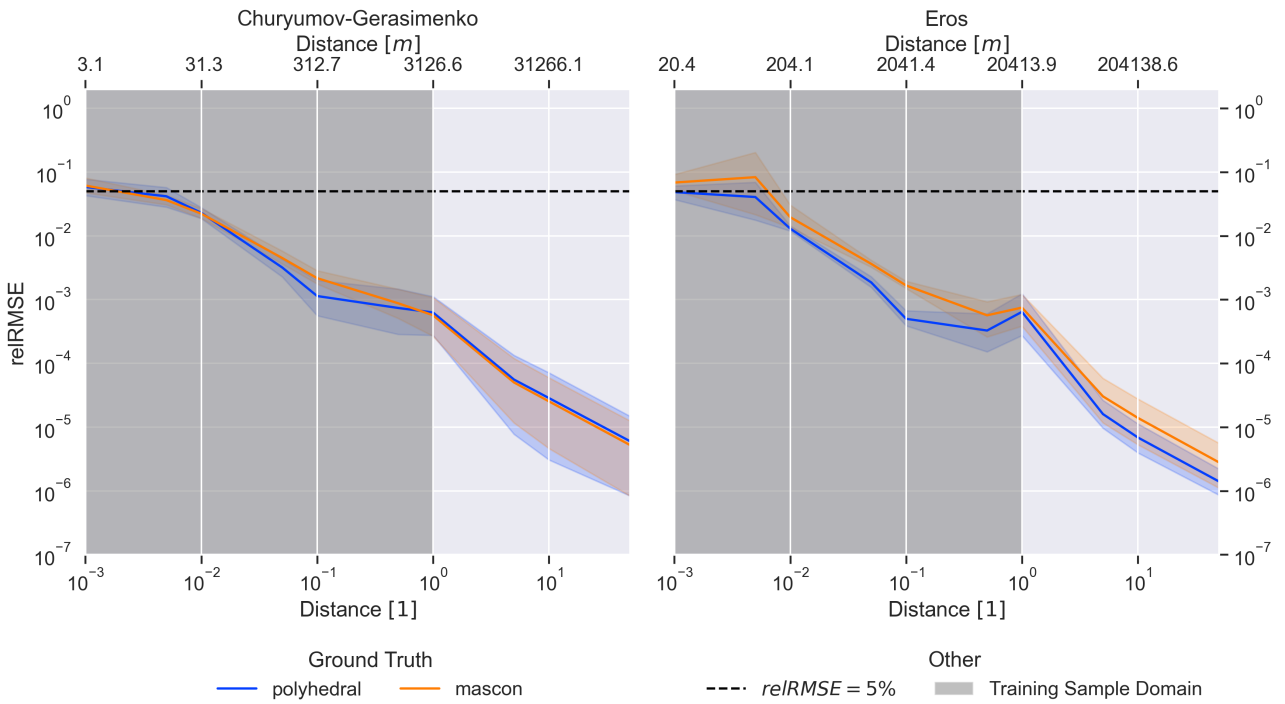


Figure 3: This figure compares the relative errors of mascon-trained models and polyhedral-trained models, always with 100% resolution and for 10000 iterations with early stopping.

with the 0.1% resolution mesh still shows a solid relRMSE of only 6.3% (Churyumov-Gerasimenko) and 2.5% (Eros) for a normalized range of 1.

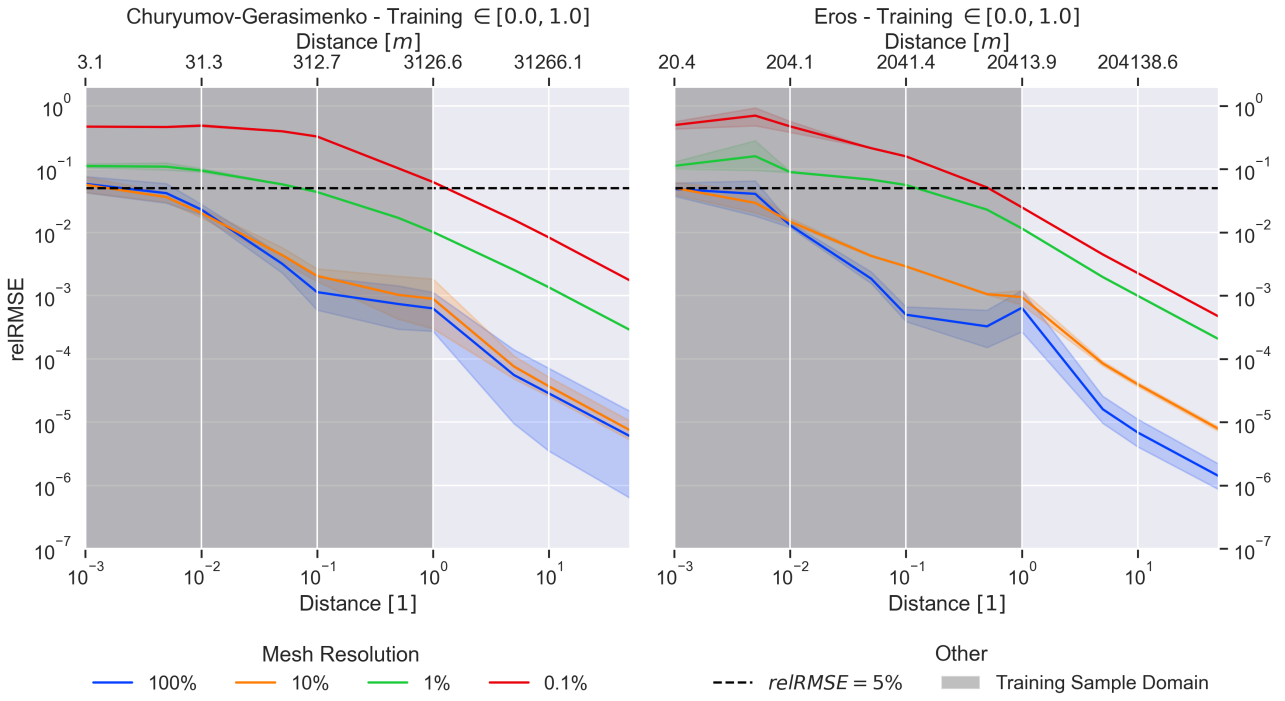


Figure 4: The performance of the trained model depending on the utilized mesh resolution. The validation was done based on the 100% polyhedral ground truth.

4.2 Sampling Distance

Note that the blue lines in Figure 3 and Figure 4 are the same as in Figure 5 in the left column. They show the average relative error of ten trained models with the polyhedral ground truth, respectively, based on the 100% resolution mesh and with sampling in the range $(0, 1)$ during training.
This subsection focuses on the other blue graphs in Figure 5. They show how the relative error changes for a given height above the target body when sampling only in the normalized range $(1, 3)$ or $(3, 5)$ during training.
Immediately, it is noticeable that if the network only gets points from mid or far away during the training, the relative error in the close range increases to 100% and above. However, the network performs well for the mid to far range, and the mean relative error remains below 4.3% until a normalized distance of 1 or further away for models trained $\in \{(1, 3), (3, 5)\}$. Hence, the network always performs well in the sampling range or further away in the observed cases. It generalizes to the ranges more distant than used during training. It is also interesting to note that in the range where geometry does not play such a large role, the network is able to generalize and scale the predictions when getting closer to the body. This is reflected in the plots on the right side of Figure 5. The network has been trained with points from the region $(3, 5)$, but the average relative error always remains below the 4.3% mark, even for the $(1, 3)$ region.

4.3 Robustness to Noise

This section refers again to Figure 5, which also displays the results when some noise has been applied to the ground truth for learning.
Starting with the constant bias, models trained with a constant bias approximately equal to the magnitude of the solar radiation pressure that would act on a probe such as Rosetta or a CubSat show an almost identical relative error in validation as models trained with the original ground truth.
In the case of additive Gaussian noise, which corresponds to an absolute error of $10^{-5}\ \frac{m}{s^2}$ or $10^{-4}\ \frac{m}{s^2}$ in non-normalized units, respectively, the network is most disturbed during learning. The mean relative errors are far beyond 5% for an absolute error of $10^{-4}\ \frac{m}{s^2}$ even within the training range. However, the results differ for Churyumov-Gerasimenko and Eros. The network shows smaller relative errors for Eros than for Churyumov-Gerasimenko.
The multiplicative Gaussian noise that distorts the ground truth by $+/-\ 0.1\%, 1\%, 10\%$ respectively settles in the middle. Although the relRMSE is usually larger than when no noise has been applied, it is not drastically worse. Training in the close range with a relative Gaussian noise of 1% still results in relative errors below 4.7% even until a range of 0.005 for training in $(0, 1)$. However, with $\sigma = 10\%$, the relative error maximally increases to 11.0% for models trained in $(0, 1)$.

4.4 A posteriori: Using Pretraining

After the previous sections have shown that noise and a long sampling distance increase the relative error, we will now investigate whether these results can be improved by pretraining. The calculation is to perform the pretraining on an imprecise low-resolution model in the near range $(0, 1)$ and the fine-tuning on the 100% resolution mesh in the range $(3, 5)$ for a limited number of iterations. In the experiments, the fine-tuning was also, in parts, the subject of noise. However, here, the focus lies only on fine-tuning without any noise, as the results are comparable.
Figure 6 shows the results. Pretraining was performed for 10000 iterations, and fine-tuning (or training for the non-pretrained models) took 10 or 100 iterations. The pretrained models correspond to those in Figure 3. The blue curves are not to be confused with the preceding plots. After all, before, 10000 iterations were trained. Here, however, significantly less.

It should also be added that the fine-tuning with the distant sample points from $(3, 5)$ partially worsened the result, as a comparison of Figure 4 and Figure 6 shows. For example, the pretrained networks $\in (0, 1)$ for Eros using the 0.1% resolution mesh achieved an relRMSE of 2.5% for a normalized distance of 1, whereas the 10 iterations fine-tuning $\in (3, 5)$ worsened the relRMSE to 14.3%.
Another example of this is comparing 10 and 100 iterations for fine-tuning. The error for the far range $(3, 5)$ becomes smaller (e.g. 0.1% pretrained-resolution, distance 5.0: $1.3\% \rightarrow 0.09\%$), but it becomes larger for the close range $(0, 1)$ (e.g. 0.1% pretrained-resolution, distance 0.1: $28.1\% \rightarrow 43.2\%$). As the previous sections have shown, the model adapts to the $(3, 5)$ range. From there, it cannot generalize well to the near geometry. In the context of this experiment, however, the assumption was that the pretraining was done on an inaccurate model, and the fine-tuning contains the actual relevant onboard measurements.

5 DISCUSSION

5.1 Polyhedral vs. Mascon Model

No clear qualitative difference could be found in the comparison between the mascon and polyhedral models in Subsection 4.1. In both cases, the relative error is similar even for close distances. A possible reason could be that the mascon model for both bodies is derived from the polyhedral mesh, and therefore, there is a dependency between the two ground truths used. Furthermore, the numerical accuracy could be limited during training. Especially the numerical integration could be crucial and limit the maximum achievable precision. In addition, only single precision is used, but the polyhedral model provides a more precise result, i.e., precision that cannot be learned in this sense - since it cannot be represented in single precision. Mao et al. [29], for example, could improve accuracy using double precision in their physics-informed neural network. In the context of this work, only GPUs of type RTX 2080 Ti were available. Thus, testing double precision was not practical for this reason.
A pleasing result is that even with a ground truth of poor quality, as was the case here during training with the 0.1% resolution mesh, a satisfactorily low error can be achieved up to the mid-range distance - an essential requirement for pretraining on Earth before the mission as discussed in Subsection 5.4.

5.2 Sampling Distance

Regarding the results presented in Subsection 4.2 on sampling, one could expect that the network could not predict the geometry and associated accelerations only by training in the mid-range to far-range. However, a geodesyNet is able to generalize and scale measurements as a function of distance successfully. This is clearly shown in the right plots in Figure 5. Conversely, a network is able to generalize successfully from the conducted sampling ranges up to 50 times the distance. A possible reason for that could be the employed normalization and loss strategy involving the continuously recalculated mass normalization factor. The experiments also showed that the models could generalize from the far to the mid-range as long as the geometry is not of influence. This feature could also be beneficial in a hypothetical onboard scenario, as it allows a probe to get closer step by step and improve the model accordingly. As here observed, the model makes a solid prediction a little closer than the actual sampling range during training.

5.3 Noise

Subsection 4.3 shows that a slight constant bias is calculable. In other words, future missions using a proper spacecraft will not have to worry about accounting for it.

It also shows that training a geodesyNet when expecting a large absolute error in the measurements utilized for training does not make sense. Especially if the error is notable large compared to the actual gravity; this is also why models trained for Eros perform better with additive Gaussian noise. Churyumov-Gerasimenko is less massive and smaller. Thus, the gravity signal is smaller than it would be in the case of Eros, and an absolute error dominates the input signal rather than merely perturbing it.
If, on the other hand, as in the case of multiplicative errors, the magnitude of the gravity signal is known and only the number of determinable digits is a problem, then even with no exact digit after the decimal point, as in the 10% deviation case, a robust network can be constructed.

5.4 Pretraining

Subsection 4.4 analyzed pretraining. The results show that pretraining reduces the model error and allows for obtaining a better performance with fewer gravity measurements. In an onboard scenario, where energy is of critical interest, pretraining on Earth, albeit with a lower resolution ground truth, is thus advantageous.
Moreover, it must be added in this context that the scenario, pretraining, and fine-tuning on range $(0, 1)$ has not been considered here. Instead, the mixed range sampling scenario was considered, as it would be conceivable for a mission: training at home and slowly improving the model when approaching the target.

6 CONCLUSION AND FUTURE WORK

In summary, this work studied the robustness of geodesyNets. The variables of interest were: the underlying training ground truth, the sampling distance for the training points, and the effects of different types of noise and whether pretraining can reduce the number of training iterations.
There were only minor differences when the geodesyNet was trained with the mascon or the polyhedral gravity model, probably due to limitations by the numerical precision and the relation of the mascon ground truth and the polyhedral mesh.
The geodesyNet cannot learn the geometry of an irregular body properly with only distant measurements. However, the neural density field generalizes well and yields robust results even in areas where no training has been performed, as long as they are not in the close range. In order to be able to predict the acceleration in the close-range, near the surface, training in this region needs to be conducted.
Noise negatively impacts the training's results. However, this depends strongly on the magnitude compared to the input gravity signal. If there exists an absolute boundary for the measurable precision of the acceleration and thus the actual gravity signal is no longer distinguishable from the measurement error, the training is unproductive. However, if the magnitude of the gravity signal is known, even with relative measurement deviations of up to 10%, solid training results are achievable. From the point of view that the magnitude of the gravity signal is known, the training is successful.
Pretraining allows more precise results in an onboard scenario with less sampling and is preferable, even if the ground truth is of low resolution, to a random Xavier initialization.
Future work could, for example, consider other forms of sampling such as sampling from a trajectory. The utilization of double precision for training could also be of interest in order to decrease the error further.

REFERENCES

[1] D. Izzo and P. Gómez, "Geodesy of irregular small bodies via neural density fields: geodesynets," *arXiv preprint arXiv:2105.13031*, 2021.

[2] M. Trisolini, C. Colombo, and Y. Tsuda, "Target selection for near-earth asteroids in-orbit sample collection missions," *Acta Astronautica*, vol. 203, pp. 407–420, 2023.

[3] J. M. Leonard, J. L. Geeraert, B. R. Page, A. S. French, P. G. Antreasian, C. D. Adam, D. R. Wibben, M. C. Moreau, and D. S. Lauretta, "Osiris-rex orbit determination performance during the navigation campaign," in *2019 AAS/AIAA Astrodynamics Specialist Conference*, no. AAS 19-714, 2019.

[4] D. Scheeres, A. French, P. Tricarico, S. Chesley, Y. Takahashi, D. Farnocchia, J. McMahon, D. Brack, A. Davis, R.-L. Ballouz *et al.*, "Heterogeneous mass distribution of the rubble-pile asteroid (101955) bennu," *Science advances*, vol. 6, no. 41, p. eabc3350, 2020.

[5] A. Fujiwara, J. Kawaguchi, D. Yeomans, M. Abe, T. Mukai, T. Okada, J. Saito, H. Yano, M. Yoshikawa, D. Scheeres *et al.*, "The rubble-pile asteroid itokawa as observed by hayabusa," *Science*, vol. 312, no. 5778, pp. 1330–1334, 2006.

[6] M. Šprlák and S.-C. Han, "On the use of spherical harmonic series inside the minimum brillouin sphere: Theoretical review and evaluation by grail and lola satellite data," *Earth-Science Reviews*, vol. 222, p. 103739, 2021.

[7] 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*, vol. 67, no. 1, pp. 617–638, 2021.

[8] P. T. Wittick and R. P. Russell, "Mascon models for small body gravity fields," in *AAS/AIAA astrodynamics specialist conference*, vol. 162, 2017, pp. 17–162.

[9] D. Tsoulis, "Analytical computation of the full gravity tensor of a homogeneous arbitrarily shaped polyhedral source using line integrals," *Geophysics*, vol. 77, no. 2, pp. F1–F11, 2012.

[10] D. Tsoulis and G. Gavriilidou, "A computational review of the line integral analytical formulation of the polyhedral gravity signal," *Geophysical Prospecting*, vol. 69, no. 8-9, pp. 1745–1760, 2021.

[11] M. von Looz, P. Gomez, and D. Izzo, "Study of the asteroid bennu using geodesyanns and osiris-rex data," *arXiv preprint arXiv:2109.14427*, 2021.

[12] S. Petrović, "Determination of the potential of homogeneous polyhedral bodies using line integrals," *Journal of Geodesy*, vol. 71, no. 1, pp. 44–52, 1996.

[13] D. Tsoulis and S. Petrović, "On the singularities of the gravity field of a homogeneous polyhedral body," *Geophysics*, vol. 66, no. 2, pp. 535–539, 2001.

[14] A. Konopliv, S. Asmar, E. Carranza, W. Sjogren, and D. Yuan, "Recent gravity models as a result of the lunar prospector mission," *Icarus*, vol. 150, no. 1, pp. 1–18, 2001.

[15] D. Izzo, G. Meoni, P. Gómez, D. Dold, and A. Zoechbauer, "Selected trends in artificial intelligence for space applications," *arXiv preprint arXiv:2212.06662*, 2022.

[16] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar, "Neural fields in visual computing and beyond," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 641–676.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[18] 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, pp. 66–79, 2020.

[19] A. Gao and W. Liao, "Efficient gravity field modeling method for small bodies based on gaussian process regression," *Acta Astronautica*, vol. 157, pp. 73–91, 2019.

[20] 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, no. 2, p. 13, 2022. [Online]. Available: https://doi.org/10.1007/s10569-022-10069-5

[21] ——, "Physics-informed neural networks for gravity field modeling of small bodies," *Celestial Mechanics and Dynamical Astronomy*, vol. 134, no. 5, p. 46, 2022. [Online]. Available: https://doi.org/10.1007/s10569-022-10101-8

[22] ——, "THE PHYSICS-INFORMED NEURAL NETWORK GRAVITY MODEL REVISITED: MODEL GENERATION III," 2023, https://hanspeterschaub.info/Papers/Martin2023.pdf, last accessed: 15.04.2023.

[23] J. Schuhmacher, "Efficient Polyhedral Gravity Modeling in Modern C++," Technische Universität München, Tech. Rep., 2022.

[24] R. W. Gaskell, "Eros polyhedral model," https://arcnav.psi.edu/urn:nasa:pds:gaskell.ast-eros.shape-model, 2008, last accessed: 15.04.2023.

[25] ESA/Rosetta/MPS, "Churyumov-Gerasimenko polyhedral model," https://sci.esa.int/web/rosetta/-/54728-shape-model-of-comet-67p, 2014, last accessed: 15.04.2023.

[26] Y. Kubo and T. Chujo, "Optimization of body configuration and joint-driven attitude stabilization for transformable spacecrafts under solar radiation pressure," *arXiv preprint arXiv:2301.08435*, 2023.

[27] M. Yousef, M. El-Saftawy, and A. Mostafa, "Balancing the effects of solar radiation pressure on the orbital elements of a spacecraft using lorentz force," *Scientific Reports*, vol. 12, no. 1, p. 15819, 2022.

[28] R. Rummel, T. Gruber, and R. Koop, "High level processing facility for goce: products and processing strategy," in *Proceedings of the 2nd International GOCE User Workshop GOCE, The Geoid and Oceanography, ESA SP-569*, 2004.

[29] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112789, 2020.

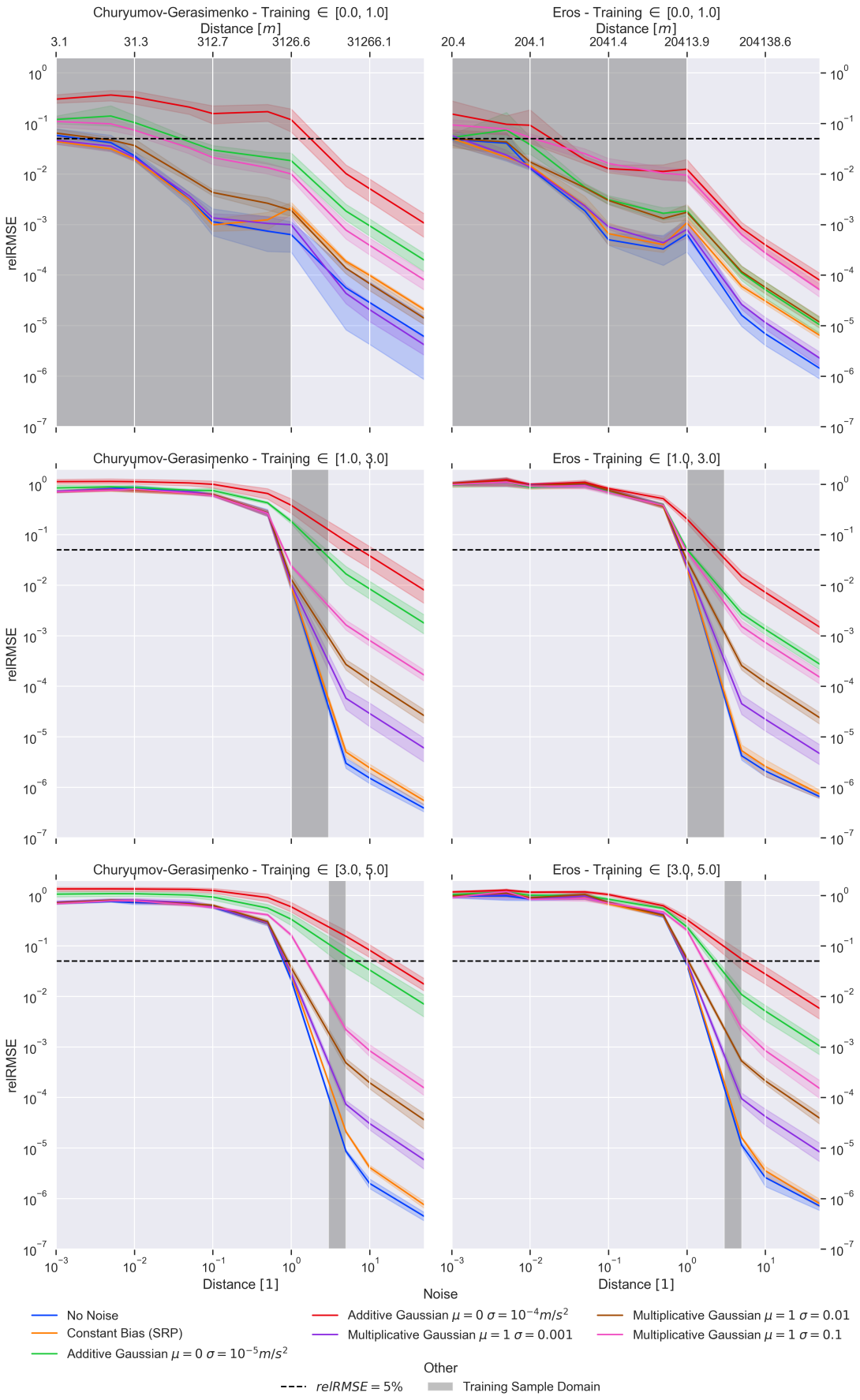


Figure 5: Effects of Noise and Sampling Shell on obtainable precision.

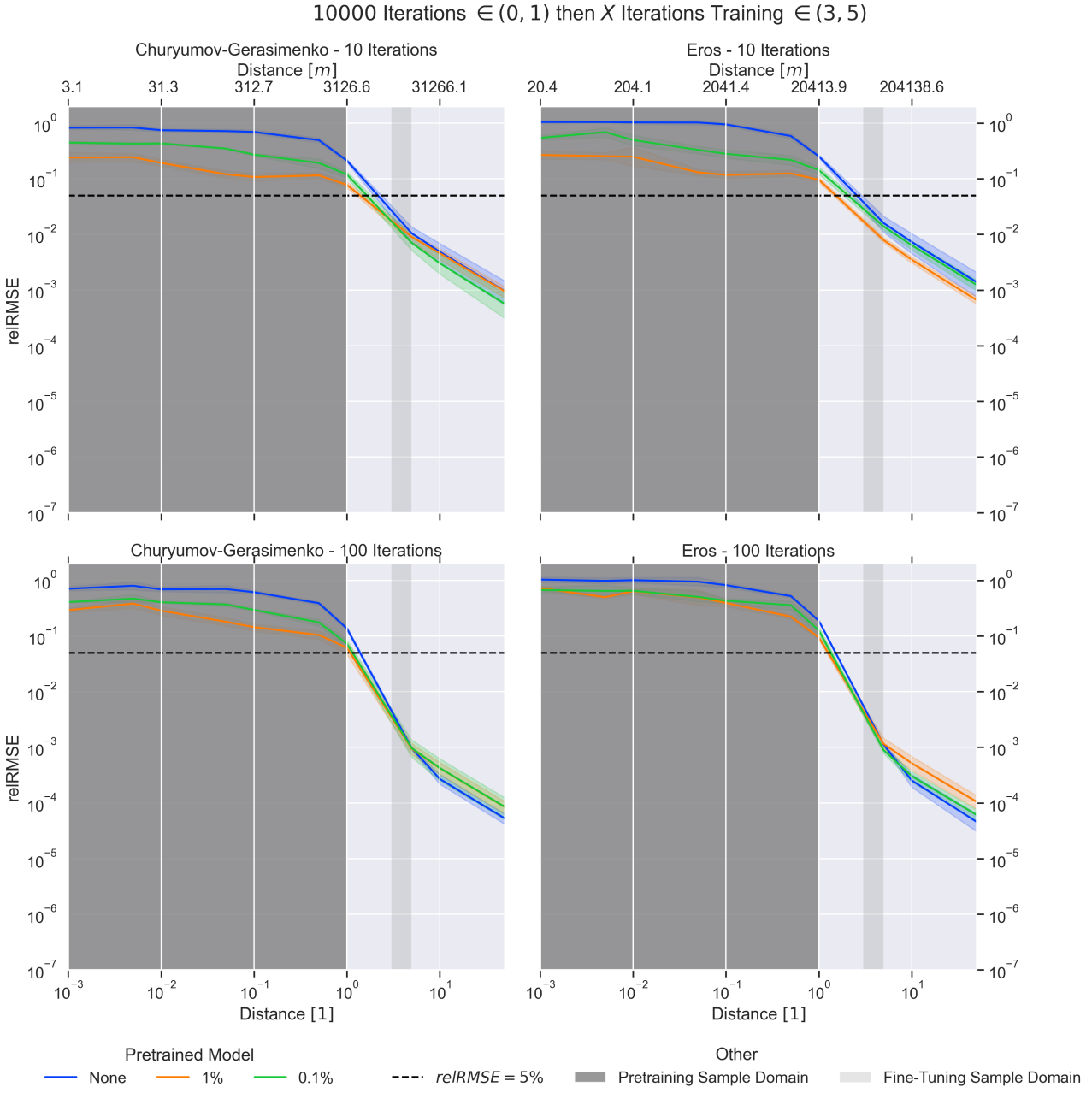


Figure 6: This figure compares the relative errors of pretrained models versus non-pretrained models. The pretraining was always conducted in the range $(0, 1)$ marked in dark gray for 10000 iterations, and the fine-tuning (training for none) was conducted in the range $(3, 5)$ for 10 or 100 iterations. Here, no noise was applied.