# LEO Satellite Orbit Prediction with Physics Informed Machine Learning

**Francesco Alesiani** [*]
NEC Laboratories Europe,
Kurfürsten-Anlage 36, Heidelberg 69115 Germany

Makoto Takamoto
NEC Laboratories Europe,
Kurfürsten-Anlage 36, Heidelberg 69115 Germany

Toshio Kamiya
School of Science and Engineering,
Meisei University, 2-1-1 Hodokubo, Hino, Tokyo 191-8506 Japan

Daisuke Etou
Space System Department,
NEC Corporation, 1-10 Nisshin-cho, Fuchu, Tokyo 183-8501 Japan

## Abstract

In recent space missions, the more complicated the missions become, the more important autonomous spacecraft controllers are. In this study, we focus on a precise autonomous orbit prediction of LEO (Low-Earth Orbit) satellite. The Physics Informed Machine learning (PIML) enables us to predict the satellite orbit with compatible accuracy to the numerical simulation but more efficiently. The proposed physics-informed machine learning algorithm is based on modeling the orbital trajectory as a Partial Differential Equation (PDE) and using deep Neural Operator (NO) to model the system's dynamic. We analyze the prediction accuracy of the proposed model and compare it with a pure data-driven model. We also provide the evaluation of the computation and memory requirements of the proposed approach and compare it with a numerical orbit simulator.

## 1 Introduction

Recently, there has been a growing movement to utilize space using LEO (Low-Earth-orbit) satellites in various fields such as earth observation, communications, positioning, disaster prevention, and the environment. A LEO satellite orbits the Earth at an altitude of 400 km to 600 km, so it slows down due to the thin atmosphere and loses its altitude. To compensate for the altitude loss, orbit control is performed by firing the thrusters mounted on the satellite. To perform accurate and autonomous orbital control, it is necessary to perform long-span orbital prediction with high accuracy by the computer installed on the satellite. In the last decade, machine learning has gained significant attention as a vital tool for obtaining valuable insights and achieving precision in predictions across diverse domains. This has been driven by the accessibility of substantial datasets and enhancements in computational capabilities.

A highly accurate orbit prediction numerical model such as special perturbation considers the effects of atmosphere and solar radiation, but it is not suitable for practical use because some parameters such

[*] Francesco.Alesiani@neclab.eu

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

Table 1: Position error by each disturbance

| Term | Contribution | |
|---|---|---|
| | Ratio to 2-Body dynamics | Accumulated error for 1 week (km) |
| 2 Body(Earth) | 100% | - |
| J2 term(Earth) | 0.14% | 9.52 |
| J3 term(Earth) | 0.00081% | 0.05508 |
| J4 term(Earth) | 0.00101% | 0.06868 |
| Sun | 0.000006% | 0.00043996 |
| Moon | 0.000015% | 0.00102 |
| Aerodynamics | 0.00659% | 0.44812 |
| Solar Radiation | 0.00068% | 0.045968 |

as atmospheric drag coefficients must be uploaded frequently from the ground. Physics Informed Machine Learning (PIML) aims at extending machine learning (ML) methods to model physical systems. PIML neural networks are used to solve Ordinary Differentiable Equations (ODEs) and Partial Differentiable Equations (PDEs) complementing traditional numerical solvers. Numerical methods are based on exact equations, while ML methods are data-driven and can naturally work with noisy observations or when the underlying physical model is not fully known. Moreover, neural models are continuously differentiable in both their inputs and their parameters, a useful property in several applications from engineering to physics.

## 2 LEO Equation of motion

The equation of motion of LEO satellite is as follows,

$$M_v \frac{d^2 x}{dt^2} = M_v \nabla_U(x,t) + f_{\text{dist}}(x,t) \tag{1}$$

where $M_v$ is mass of satellite, $U(x,t)$ is gravitational potential by the earth, and $f_{\text{dist}}$ is the force by disturbance on orbit. Gravitational potential U(t) is written as

$$\begin{aligned} U(x,t) &= \frac{GM}{r}(U_0(x,t) + U_{J_2}(x,t) + U_{J_3}(x,t) + U_{J_4}(x,t)) \qquad (2) \\ U_0(x,t) &= -1 \\ U_{J_2}(x,t) &= \left(\frac{a}{r}\right)^2 J_2 \frac{1}{2}(3\cos^2\theta - 1) \\ U_{J_3}(x,t) &= \left(\frac{a}{r}\right)^3 J_3 \frac{5}{2}(\cos^3\theta - \frac{3}{5}\cos\theta) \\ U_{J_4}(x,t) &= \left(\frac{a}{r}\right)^4 J_4 \frac{35}{8}(\cos^4\theta - \frac{6}{7}\cos^2\theta + \frac{3}{35}) \end{aligned}$$

with $\theta$ latitude of the satellite, $a$ the diameter of the Earth at equator (6378.137 km), $GM = 3.986e5km^3/s^2$, $J_2 = 0.0010826$, $J_3 = -0.00000254$, $J_4 = -0.00000151$, where the values are based on experimental data. The force by disturbance on orbit $f_{\text{dist}}$ is written as

$$f_{\text{dist}}(x,t) = f_{\text{moon}}(x,t) + f_{\text{sun}}(x,t) + f_{\text{aero}}(x,t) + f_{\text{solar}}(x,t) \tag{3}$$

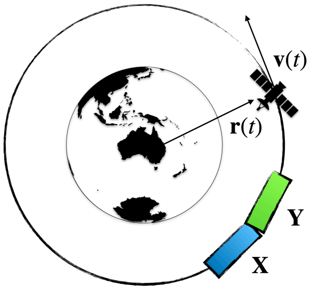
where $f_{\text{moon}}(x,t)$, $f_{\text{sun}}(x,t)$, $f_{\text{aero}}(x,t)$, $f_{\text{solar}}$ are the disturbance due to the gravitational force of the Moon, the Sun, the aerodynamic force, and solar radiation force.

### 2.1 Estimating the error by each disturbance

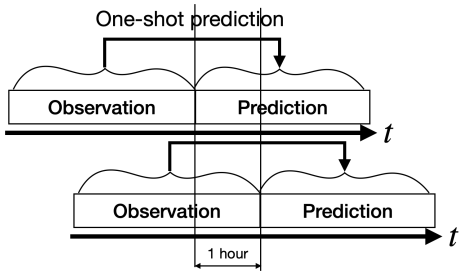
Table 1 shows the estimated error by each disturbance. In this table, the force in the two-body dynamics is taken as $100\%$, and the ratio with each disturbance force is shown. Next, the position error when the disturbance force has passed for one week is shown.

Table 2: Detail of orbital numerical simulation

| Element | Configuration |
|---|---|
| Algorithm | Special perturbation method |
| Gravity field order from the Earth | 12 |
| Gravity from the Moon | Considered |
| Gravity from the Sun | Considered |
| Atmosphere model | MSISE00 |
| Solar Radiation | Considered |
| Cross-sectional area of the satellite body | $12.5m^2$ |
| Cross-sectional area of the solar paddle | $10m^2$ |



(a) Data preparation for the PIML training, where $\mathbf{X}$ are the past 24-hour, while $\mathbf{Y}$ are the next 24-hours.



(b) PIML trajectory prediction, where at regular intervals, for example, 1 hour, the PIML predicts the next period. This approach is called one-shot.

## 3 LEO Orbit prediction PIML model

The satellite that we used for our experiments is an earth observation satellite, whose identity can not be disclosed. Its orbital altitude is about 500km. The orbit of the satellite is Dawn-Dusk, which means there will be a full sun path, i.e., the satellite is always affected by solar radiation. The orbit data used in the experiments have been computed based on the parameters in Table 2 using an in-house developed orbital numerical simulator. The numerical solver models the effects of the atmosphere, solar radiation pressure, and other disturbance effects described in section 2. The starting position of the satellite used to generate the data is based on the actual satellite position.

To train a PIML model, we first define the prediction task. The prediction task consists of observing past data up to 24 hours and predicting the orbit for the next 24 hours (seeFigure 1a). Thus, a single ML sample consists of a pair of input and output of the same size ($\mathbb{R}^{n\times 6} \times \mathbb{R}^{n\times 6}$), where each orbit sample has information on the position $(X, Y, Z)$ and velocity $(V_x, V_y, V_z)$ in the Earth-centered coordinate system. The length $n$ of the ML sample depends on the sampling frequency (24 hours × sampling frequency ). We generate multiple overlapping samples that overlap for 30 minutes. We then split the available data into train, validation, and test data ($10k$, $1k$, $1k$ samples). The train data is used for actual training of the PIML model, while the validation data is used for hyper-parameter selection, where hyper-parameters consist of different architecture choices, pre-processing for example. The test data is the data we use to evaluate the performance of the trained model.

### 3.1 Orbit1d Architecture, Training, Testing, and Hyper-parameter search

In this paper, we extend two machine learning models, Fourier Neural Operator (FNO) as a PIML model [LKA+21], and Transformer [VSP+17] as a pure ML model. FNO is a Neural Operator model [KLL+21] that uses the expressive power of neural networks for learning mesh-free, infinite-dimensional operators. Although it is mainly developed for simulating partial differential equations [TPL+22], we extended it to predict the satellite trajectory; FNO accepts now the past 24h trajectory data as one- dimensional field data and predicts the next 24h trajectory also as one-dimensional field data. Indeed, contrary to the original FNO architecture, the input of the model is over time as opposed to space, while the channels are the position and velocities as opposed to the field values (e.g. density, flow), we thus name the architecture Orbit1d. We experimented with auto-regressive

Table 3: Performance of the evaluated architectures: Orbit1d (FNO) and Transformer (T5). Position error is computed in $km$, while speed in $km/s$.

| Measure | Dataset | Model | Value |
|---|---|---|---|
| MSE | Test | Orbit1d | 333.45 |
| | | Transformer | 790.88 |
| MAE | Test | Orbit1d | 21.55 |
| | | Transformer | 225.62 |
| Position MAE | Test | Orbit1d | 43.05 |
| | | Transformer | 450.74 |
| MSE | Validation | Orbit1d | 435.05 |
| MAE | Validation | Orbit1d | 23.44 |
| Position MAE | Validation | Orbit1d | 46.81 |

Table 4: Computational and Memory evaluation of the PIML model, where samples are 30 seconds frequency, and comparison with NS.

| Evaluation metric | Value | Unit |
|---|---|---|
| Inference time (prediction of 24 hours, based on past 24 hours) @ GPU @30s | $1.754 \pm 0.366$ | ms |
| GPU Memory use @30s | 3.152 | MB |
| Number of parameters @30s | 220’638 | |
| Numerical Solver (24 hours prediction) @ CPU @10s | 25.6 | s |
| Computation advantage of PIML vs NS | 14’630x | - |

architecture, but the error quickly accumulated as shown in the experimental section, so we adopted the one-shot prediction (Figure 1b), where the full trajectory is predicted. We also trained a T5-type Transformer model [RSR+20] as a pure data-driven baseline model. T5 accepts past 15-minute trajectories and predicts the next 6-minute trajectories with 3-minute intervals; T5 continues this process autoregressively until predicting the next 24h trajectories. The training set-up and procedures are provided in Appendix A.

## 4 Results

We computed the error on a test and a validation set. The validation set is used to select the optimal hyper-parameter configuration, reported in Table 3. The loss used during training is the MSE loss between the predicted and the true 24-hour trajectory. We also compute the error in terms of mean absolute error (MAE) of the position, and the position and the velocities whose values are listed in the following Table. The results show that our PIML model (FNO) has a much better performance than the traditional data-driven method (T5). One of the reasons for the larger error of the T5 model is the use of autoregressive prediction, that is, iteratively predicts the next time-step trajectory by utilizing the prediction for the next-step input data. As a result, the error accumulates in the temporal direction, resulting in a larger error for the next time-step prediction. On the contrary, the FNO-based Orbit1d model performs the prediction in one shot (see Figure 1b) and thus is able to model the consistency of the full trajectory. Further, the Orbit1d model used spectral convolution that has a better frequency resolution inspired by the transitional numerical solver. A key limitation we observe is the scale of the error and main trajectory. The model, after observing the past 24 hours, needs to perform two tasks, for the next 24 hours: 1) learning to predict the main trajectory and 2) learning the disturbance and correcting the prediction with the disturbance. The current method that presents better performance, only performs the second task, while querying a traditional solver for the first term. We evaluated the model for the number of parameters, memory requirement, and computation time. For this, we experimented with a GPU card (GeForce GTX 1080 Ti) with GPU RAM of 11264MiB. From Table 4, we observe that the ML model can predict around 500 trajectories per second, this speed can be used for applications that require a high refreshing frequency. The proposed approach provides accurate trajectory prediction for non-critical applications and has a speedup of around $14,000$ times, as shown in Table 4, in comparison to accurate numerical simulators, further, it adapts to the availability of new observatory data and thus could model hidden dynamics.

## 5 Discussion and Conclusions

In this paper, we applied physics physics-informed machine-learning technique to the satellite orbit prediction task for the first time.

We evaluate the performance of our PIML models in predicting the full trajectory generated using a highly accurate numerical simulator and using real data as a starting point, and not only the disturbances with respect to the nominal trajectory as in recent works [CS23]. Our preliminary results showed that our PIML model can provide us with a promising result. Although the trajectory error is still larger than the recent work that only learns disturbances, we consider this could be solved simply by increasing training data because of the data-driven manner. Importantly, we showed that our PIML can predict 24h trajectory approximately $10^4$ faster than the conventional numerical methods, which is very promising for the future use of on-the-fly satellite trajectory prediction.

## References

[CS23] Francisco Caldas and Cláudia Soares. Machine learning in orbit estimation: a survey, 2023.

[FKA+18] Zhen-Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao-Jun Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2235–2245, 2018.

[JBBH07] Adrian Jäggi, Gerhard Beutler, Heike Bock, and Urs Hugentobler. Kinematic and highly reduced-dynamic leo orbit determination for gravity field estimation. In *Dynamic Planet: Monitoring and Understanding a Dynamic Planet with Geodetic and Oceanographic Tools IAG Symposium Cairns, Australia 22–26 August, 2005*, pages 354–361. Springer, 2007.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[KLL+21] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

[LKA+21] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[ŠR03] D Švehla and M Rothacher. Kinematic and reduced-dynamic precise orbit determination of low earth orbiters. *Advances in Geosciences*, 1:47–56, 2003.

[TPL+22] Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[WLS+22] Kan Wang, Jiawei Liu, Hang Su, Ahmed El-Mowafy, and Xuhai Yang. Real-time leo satellite orbits based on batch least-squares orbit determination with short-term orbit prediction. *Remote Sensing*, 15(1):133, 2022.

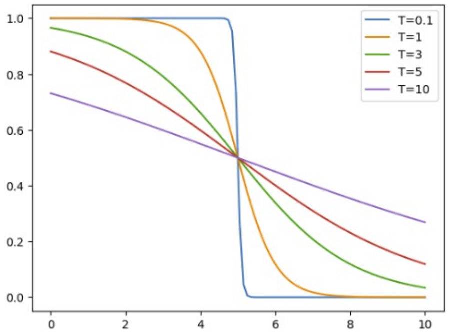


Figure 2: Noise scheduling function as a function of the training epoch.

## A The detailed training procedure for T5 model

For our T5 model, we utilized the vanilla Transformer encoder-decoder structure [VSP$^+$17]. The past trajectories are given as input data for both of encoder and decoder modules with 4 layers. The number of attention head is set as unity. The dense layer and layer normalization are substituted into the ones introduced in T5 paper. We utilized two one-layer networks before and after the Transformer model to change the channel number from 6 (x, y, z, vx, vy, vz) to the hidden-channel number of the Transformer model: 64. During the training, we calculated the loss function using the Huber loss and Wing loss [FKA$^+$18] functions. The weight parameters were optimized using Adam optimizer [KB17].

## B Training procedure

To train the network, we used data batches with random shuffling. The batch is composed of 100 sample trajectories. Each batch is propagated through the model and the prediction interval is compared with the true future trajectory. The gradient is computed with respect to the parameters of the network and the parameters of the network are updated based on the selected optimization algorithm. As shown in the next sections, we used different optimization algorithms. We repeated this procedure for all batches of the training dataset and repeated it for 200 epochs, reshuffling the dataset at every new epoch. To improve the robustness of training, we add noise whose strength over training epochs is defined by the scheduling function. The function is scaled with the variance of the error and the shape is shown in Figure 2.

During hyper-parameter search, we implemented some early-stop if we see no progress in the training loss.

## C Hyper-parameters

The PIML model is parametrized by different hyper-parameters that are presented in Table 5. We present in the following meaning: 1) the architecture, we considered two architectures, one inspired by the FNO operator that we called Orbit1d and another one inspired by Visual Transformer (ViT) in a configuration to as Auto-Encoder (AE). 2) The prediction modality, we consider the one-shot prediction where we use the full data (X) and predict the full future trajectory (y). Auto-regressive approach (see Figure 4) generates prediction using a moving window. It first considers the samples from the last 15 minutes and then predicts the next period (in our case 6 minutes). The window is then moved in time and the prediction is included in the new input. The next hour is then predicted. The process is repeated until the full 24-hour period is covered. 3) the coordinate system: we considered the original cartesian coordinate system (x,y,z) or a polar transformation in $(\rho, \theta, \phi)$, since this transformation creates a jump in the angles, we considered the representation there the angles are mapped into the sine and cosine function, thus the coordinate system is $(\rho, sin\theta, cos\theta, sin\phi, cos\phi)$. 4) Data processing: we consider the original input or a time difference between consecutive samples or transform the input data to data of unitary norm and zero mean. The transformation is then applied

Table 5: Hyper-parameter description for the PIML

| Hyper-parameter | Possible configurations | Optimal parameter |
|---|---|---|
| Architecture | Orbit1d, ViT-AE | Orbit1d |
| Prediction | One-shot, autocorrelation | One-shot |
| Coordinate system | Cartesian, Polar, Polar with cos/sin representation | Cos/Sin Polar |
| Data pre-processing | Original space, Time sample difference, scaling to N(0,1) | Scaling |
| Time Resolution | 3 minutes, 1 minute, 30 seconds, 10 seconds | 30 seconds |
| ML optimizer | Adam, RMS, SGD | Adam |
| ML Scheduler | None, LinearLR, ExponentialLR | ExponentialLR |
| Orbit1d Modes, Width, hidden dimensions and number of layers | Range of values: [8,128], [8,64], [8, 512], [1,5] | 37, 51, 387, 2 |
| Learning rate | Range of values $[1e-5, 1e-1]$ | 0.9e-4 |
| Noise Temperature and level | Range of values $[.1, 10]$, $[1e-5, 1e-1]$ | $T_N = 6.4371$,<br>$\sigma_N = 0.0003$ |

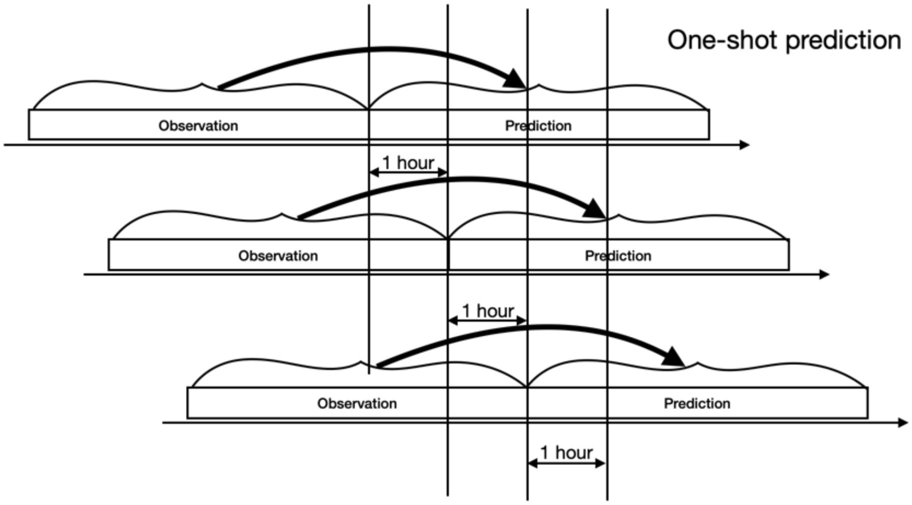


Figure 3: PIML trajectory prediction, where at regular interval, for example 1 hour, the PIML predicts the next period. This approach is called one-shot.

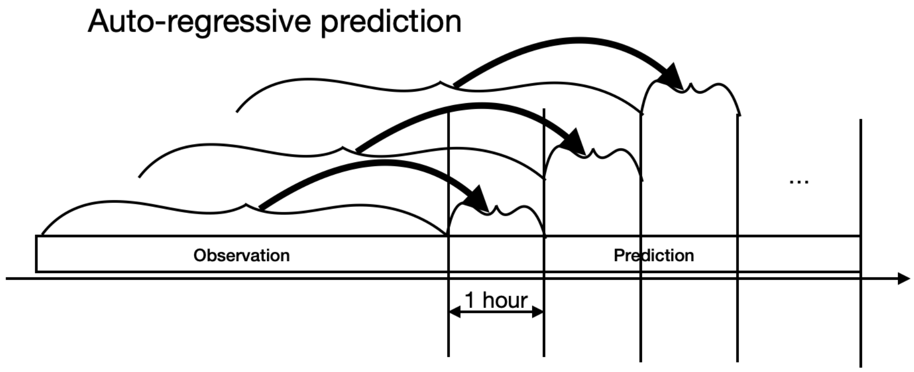


Figure 4: Auto-regressive prediction using PIML, contrary to the one-shot approach the prediction is repeated inside a loop.

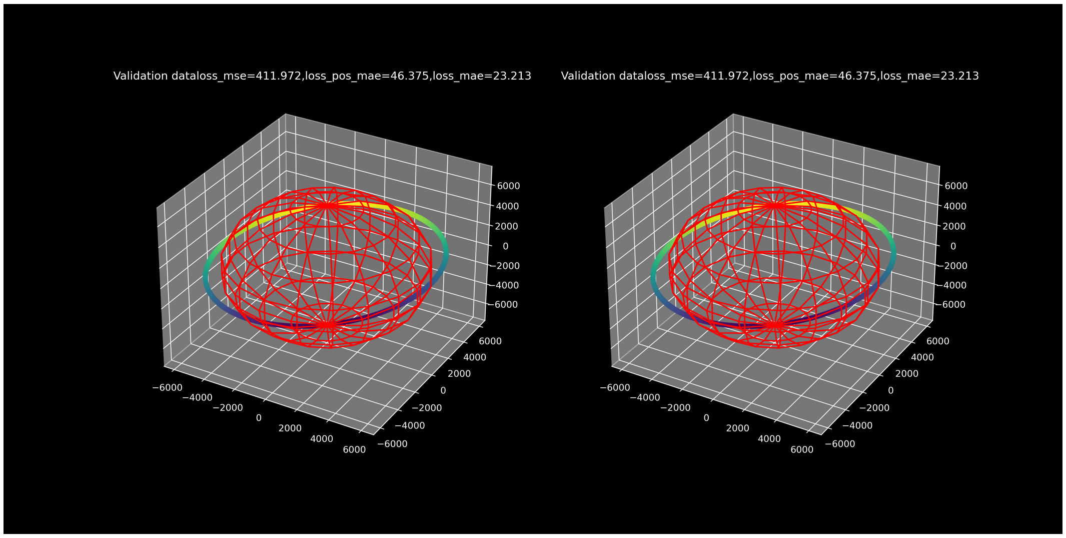


Figure 5: Visualization of orbit from the validation dataset (predicted, true).

back during prediction. 5) time-frequency: we consider the data at 3 minutes, 1 minute, 30 seconds, or 10 seconds. In Figure 3 and Figure 3 we show the two prediction modalities.

## D 24-hours prediction (3d & 1d plots)

Figure 5 shows an example of prediction along with the true trajectory for the 24-hour period. The period of the satellite is about 90 minutes, so we expect to observe around 16 orbits.

In Figure 6 we show the same information but on the three major cartesian coordinates with respect to Earth: X, Y, Z. Both predicted and true trajectories are visualized. The error is barely seen and seems concentrated at the turning point of the sinusoidal shape. In other experiments, we have seen the model introducing error at the very end of the trajectory, showing that the extremities of the prediction are the most critical points for the ML model to predict.

In Figure 7 and Figure 8 similar to the previous figures, we show the predicted and true trajectory for the test datasets. While the ML model is optimized for the validation dataset, performance is similar to the test dataset.

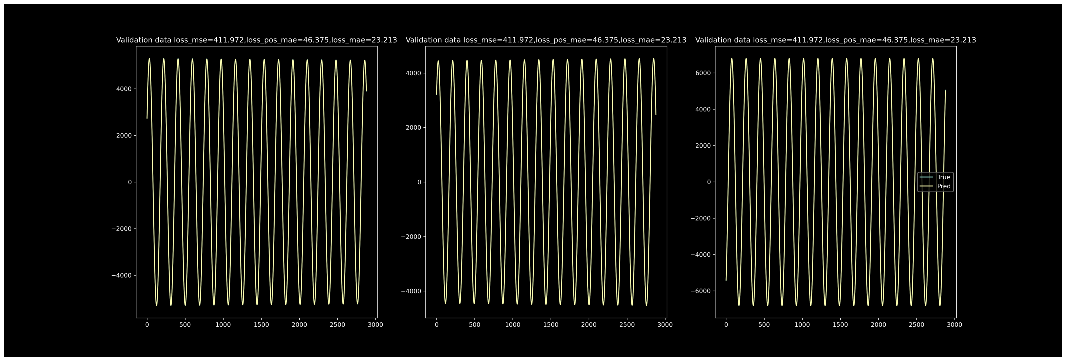


Figure 6: predicted trajectory of a sample trajectory from the validation dataset presented in the x, y, and z axes. The plot shows the comparison of the predicted and the true trajectories.

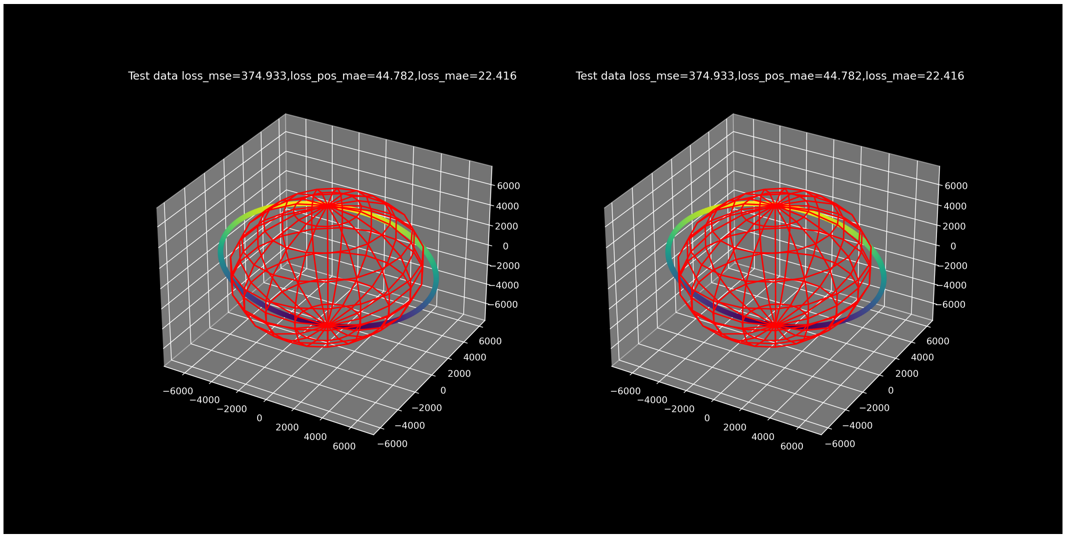


Figure 7: Visualization of the predicted and true trajectories from the test dataset

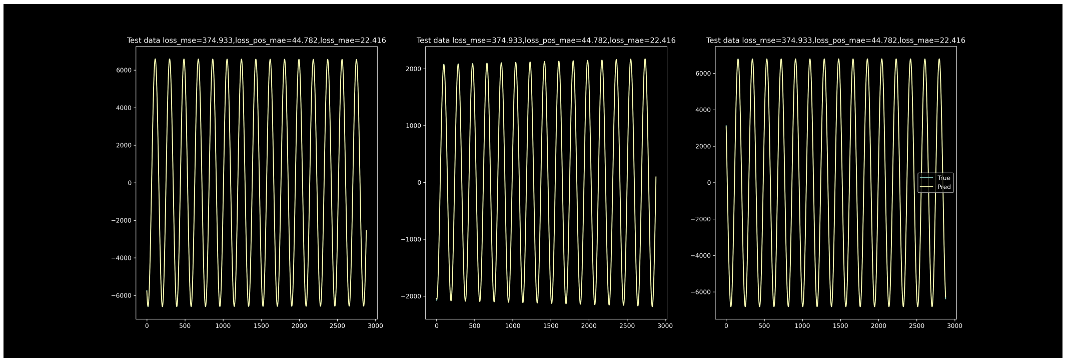


Figure 8: Visualization of the predicted and true trajectories in the cartesian coordinate system from the test dataset.