

Solving the Two and Three Body Problems with Deep Learning Models

Machine Learning in Physics Course Exam

Julian Roddeman (s1080491)

July 26, 2024

Abstract

This study addresses the two-body and three-body gravitational problems using deep learning models, with a focus on Physics-Informed Neural Networks (PINNs). We cover data generation for various orbital scenarios, implement and compare different neural network architectures (MLP, LSTM, and PINN), and evaluate their performance in predicting orbital trajectories. Our approach demonstrates the potential of machine learning techniques in solving complex gravitational problems, particularly highlighting the advantages of physics-informed methods. The complete codebase for this project is available on GitHub at https://github.com/JulianRodd/Orbit_Prediction.

1 Introduction

The two-body and three-body problems are fundamental challenges in celestial mechanics. This study explores the application of deep learning models to solve these problems, addressing several key aspects of the assignment. We begin by simulating two-body and three-body problems with varying initial conditions and increased acceleration scenarios. We then implement and compare different neural network architectures, including Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Physics-Informed Neural Networks (PINNs). Our evaluation focuses on model performance in predicting orbital trajectories for different time horizons, specifically 10, 100, and 500 steps. We analyze the advantages of incorporating physical constraints in neural networks through PINNs, aiming to improve prediction accuracy and physical consistency. Finally, we assess model robustness

under different levels of measurement uncertainty, reflecting real-world challenges in astronomical observations. Through this comprehensive approach, we aim to demonstrate the potential of machine learning techniques in solving complex gravitational problems and provide insights into their applicability in celestial mechanics.

2 Literature Review

Neural networks have shown promise in orbital mechanics, with Liao et al. [3] successfully applying them to find and classify periodic orbits in the three-body problem. The introduction of Physics-Informed Neural Networks (PINNs) by Raissi et al. [6] has further advanced the field, offering a framework to incorporate physical laws directly into neural network architectures. In the context of gravitational problems, Martin and Schaub [4] demonstrated PINNs' efficacy in modeling gravity fields of celestial

bodies.

Deep learning approaches have also been applied to related challenges in astrodynamics. Cheng et al. [1] utilized neural networks for real-time optimal control in asteroid landings, while Gao and Liao [2] proposed an efficient gravity field modeling method based on Gaussian process regression. These advancements highlight the potential of machine learning in addressing complex gravitational dynamics.

However, challenges persist in ensuring physical constraints are properly enforced, avoiding overfitting, and generalizing to new scenarios. The physics-informed approach partially addresses these issues, but further research is needed to optimize network architectures, incorporate more comprehensive physical knowledge, and extend these methods to more complex N-body scenarios [5]. This literature review sets the stage for our study, which aims to contribute to this evolving field by exploring the application of neural networks and PINNs to two-body and three-body gravitational problems.

3 Methodology

3.1 Data Generation

Our data generation process was designed to create diverse, realistic orbital scenarios for two-body and three-body gravitational problems, while incorporating measurement uncertainties to reflect real-world challenges. We implemented three main simulation scenarios: the standard two-body problem, a two-body problem with increased acceleration, and a three-body problem. For the two-body problem, we simulated the motion of a light rotating object (spaceship) around a very heavy stationary object. The initial radius was set to 10,000 m, with the initial velocity calculated to achieve a stable circular orbit. In the increased acceleration scenario, we introduced

an additional sinusoidal force on the spaceship, simulating a propulsion system malfunction. The three-body problem involved simulating the motion of a light object (spaceship) under the gravitational influence of two heavy bodies. We generated varied initial conditions by randomizing the initial position and velocity of the spaceship within reasonable bounds. All simulations were based on Newton's law of gravitation:

$$F = -G \frac{Mm}{r^2} \hat{r}$$

where G is the gravitational constant, M and m are the masses of the objects, r is the distance between them, and \hat{r} is the unit vector in the direction of r . We used the Euler method for numerical integration to update the position and velocity of the objects:

$$v(t + \Delta t) = v(t) + a(t)\Delta t$$

$$r(t + \Delta t) = r(t) + v(t)\Delta t$$

To reflect real-world measurement challenges, we introduced a 1% relative uncertainty to the angle, radius, and velocity measurements of the orbiting object. These uncertainties were implemented using Gaussian noise, clipped to ensure the relative error remains within $\pm 1\%$. For each scenario, we simulated a total of 10 trajectories (spaceships) with different initial conditions, each for 100,000 time steps. We implemented rigorous checks to ensure the generated orbits are valid and realistic. For the two-body problem, we verified that the trajectory forms at least one complete orbit, spans a significant range in both x and y directions, and doesn't simply fly away. For the three-body problem, we additionally checked that the trajectory is bounded, shows sufficient complexity, and remains within a reasonable distance from the center of the system. The data was split into training, validation, and test sets.

Nine spaceships were used for training and validation, with their trajectories divided into sequences and split using 5-fold cross-validation. One spaceship was reserved exclusively for the test set, providing a truly unseen trajectory for final model evaluation. To verify the quality of our generated data, we implemented visualization functions to plot the trajectories and velocities for each scenario. These visualizations helped ensure the simulated orbits were realistic and exhibited the expected behaviors for each problem type. This comprehensive data generation approach provides a robust foundation for training and evaluating machine learning models on orbital mechanics problems, addressing the key requirements of simulating realistic scenarios with varying complexity and uncertainty levels.

3.2 Model Architectures

To address the challenge of predicting orbital dynamics, we implemented and compared three neural network architectures: Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Physics-Informed Neural Network (PINN). Each architecture was designed to capture different aspects of the two-body and three-body problems, including scenarios with increased acceleration.

Table 1: Overview of Model Architectures

Feature	MLP	LSTM	PINN
Input Size	12	12	12
Hidden Layers	3	2	4
Hidden Size	256	128	256
Output Size	4	4	4
Activation	LeakyReLU	ReLU	Tanh
Dropout Rate	0.2	0.2	0.1
Normalization	Batch	Layer	Batch
Residual Conn.	Yes	Yes	Yes

The MLP serves as our baseline model, capable of learning non-linear relationships between input features and output predictions.

Its architecture includes three hidden layers with 256 neurons each, using LeakyReLU activation and batch normalization.

The LSTM architecture, designed to capture temporal dependencies, utilizes a bidirectional LSTM layer with 128 hidden units and 2 layers. This is followed by two fully connected layers, each with 128 neurons and ReLU activation. Layer normalization is applied after the LSTM layer to stabilize training.

The PINN architecture incorporates physical constraints into the learning process. It consists of four hidden layers with 256 neurons each, using Tanh activation and batch normalization. Crucially, it employs a custom loss function that combines mean squared error with physics-based constraints:

$$total = L_{MSE} + \lambda(L_{energy} + L_{momentum}) \quad (1)$$

Where L_{MSE} is the standard mean squared error, L_{energy} ensures conservation of total energy (kinetic + potential), and $L_{momentum}$ enforces conservation of linear momentum. The hyperparameter λ balances the influence of physical constraints.

All models use residual connections and dropout to facilitate training of deeper networks and prevent overfitting. They were trained using the Adam optimizer with a learning rate of 0.001 and early stopping (patience of 5 epochs). We employed 5-fold cross-validation to ensure robust evaluation of each model's performance.

The selection of these architectures was motivated by their unique strengths. The Multilayer Perceptron (MLP) provides a simple yet effective baseline for comparison, offering a straightforward approach to learning complex patterns. The Long Short-Term Memory (LSTM) network excels in capturing long-term dependencies in orbital trajectories, making it particularly suitable for time-series prediction tasks. The Physics-Informed Neural

Network (PINN) incorporates domain-specific knowledge, potentially leading to more physically plausible predictions by integrating fundamental physical laws into the learning process. By comparing these diverse approaches, we aim to determine the most effective method for predicting orbital dynamics across various scenarios and time horizons. This comparison allows us to balance between purely data-driven learning and physics-informed modeling, providing insights into the strengths and limitations of each approach in the context of celestial mechanics.

3.2.1 Training Configuration

We implemented a consistent training approach across all model architectures to ensure a fair comparison. The Adam optimizer was used with a learning rate of 0.001. For MLP and LSTM models, we employed Mean Squared Error (MSE) as the loss function, while PINN utilized a custom loss function (detailed elsewhere). All models were trained with a batch size of 32 for a maximum of 20 epochs, with early stopping implemented using a patience of 5 epochs. To ensure robust evaluation, we employed 5-fold cross-validation. Training was conducted for each model type (MLP, LSTM, PINN) on three dataset types: the two-body problem, the two-body problem with increased acceleration, and the three-body problem. In each training iteration, one fold was reserved for validation, allowing us to assess model performance on unseen data throughout the training process. This comprehensive approach allowed us to rigorously compare the performance of different model architectures across various gravitational scenarios while minimizing the risk of overfitting.

The `generic_train` function in our codebase handles the training loop, implementing early stopping and model checkpointing. Training progress and performance metrics

were logged using Weights & Biases (wandb) for comprehensive experiment tracking.

3.2.2 Physics-Informed Neural Networks

The PINN architecture incorporates physical laws into the learning process through a custom loss function:

$$L_{\text{total}} = L_{\text{MSE}} + 0.1 \cdot (L_{\text{energy}} + L_{\text{momentum}}) \quad (2)$$

In this equation, L_{MSE} represents the standard mean squared error between predicted and true values. L_{energy} enforces energy conservation and is defined as:

$$L_{\text{energy}} = \frac{1}{N} \sum_{i=1}^N (E_{\text{pred},i} - E_{\text{true},i})^2 \quad (3)$$

where $E = \frac{1}{2}v^2 - \frac{GM_1}{r}$. Here, v is velocity, G is the gravitational constant, M_1 is the mass of the central body, and r is the distance from the central body. The L_{momentum} term ensures momentum conservation:

$$L_{\text{momentum}} = \frac{1}{N} \sum_{i=1}^N (p_{\text{pred},i} - p_{\text{true},i})^2 \quad (4)$$

with $p = mv$ representing momentum. This loss function is implemented in the `pinn_loss` function, utilizing PyTorch's automatic differentiation capabilities. The physics-based terms are weighted by a factor of 0.1 to balance their influence with the data-driven MSE loss. By incorporating these physics-based constraints, the PINN architecture aims to learn solutions that not only fit the data but also respect the underlying physical principles of orbital mechanics. This approach directly addresses the assignment's focus on PINNs and their potential advantages in solving gravitational problems, particularly for complex scenarios like the three-body problem and cases with increased acceleration. Our implementation allows for a comprehensive comparison

between purely data-driven approaches (MLP and LSTM) and the physics-informed method (PINN) across different orbital dynamics scenarios, providing insights into their relative strengths and limitations.

3.3 Inference and Evaluation

Our evaluation process encompasses a comprehensive assessment of model performance across all folds and datasets. We employ 5-fold cross-validation, with models trained on each fold evaluated on their respective validation sets and the unseen test set. Performance metrics are calculated using mean squared error (MSE) for both position and velocity predictions, with results aggregated across all folds to provide robust estimates. We evaluate models on three distinct sets: training spaceships (to assess fitting), validation spaceships (for generalization within known scenarios), and test spaceships (for performance on entirely unseen trajectories). For each set, we compute the mean and standard deviation of MSE for both position and velocity predictions, considering prediction horizons of 10, 100, and 500 steps. This approach allows us to assess model performance across different time scales and problem complexities. Importantly, all predictions are made using normalized data and subsequently denormalized for evaluation, ensuring consistent scaling across different orbital scenarios.

4 Results and Discussion

4.1 Data Generation and Simulation Results

Our data generation process simulated orbital dynamics for three distinct scenarios: the two-body problem, the two-body problem with increased acceleration, and the three-body problem. For each scenario, we simulated trajectories for 10 spaceships, providing a diverse

dataset for our subsequent machine learning models. Figure 1 in appendix A presents a comprehensive overview of the trajectories and velocities for all spaceships across these scenarios.

4.1.1 Two-Body Problem

In the standard two-body problem (Figures 1a and 1b), we observe predominantly circular orbits. There is a little amount of variation between trajectories, which aligns with our expectations given the introduced measurement uncertainties but without the additional force and acceleration. The velocity profiles show consistent periodic patterns, reflecting the stable nature of these orbits.

4.1.2 Two-Body Problem with Increased Acceleration

The introduction of increased acceleration (Figures 1c and 1d) maintains the circular nature of the orbits but with some key differences. The trajectories show slightly more variation between spaceships compared to the standard two-body problem, suggesting a less constrained system. This is in line with the expectations that the additional acceleration and force introduce this extra variability. Notably, the velocity profiles exhibit higher overall amplitudes while maintaining similar wavelengths to the standard case. This indicates that the additional force increases the speed of the spaceships without significantly altering their orbital periods.

4.1.3 Three-Body Problem

The three-body problem (Figures 1e and 1f) introduces significant complexity and chaos into the orbital dynamics. The trajectories show extreme variability, with some spaceships maintaining bounded orbits while others are ejected from the system entirely. This chaotic

behavior is mirrored in the velocity profiles, which display a wide range of amplitudes and frequencies.

The velocity patterns in the three-body problem are particularly noteworthy. We observe a mix of very high and very low amplitude oscillations, often with extremely short wavelengths. These extreme variations likely correspond to close encounters with the heavy bodies, resulting in rapid accelerations and decelerations. The highest velocity peaks probably occur when a spaceship passes close to one of the heavy bodies, gaining substantial kinetic energy before being propelled back into space or into a rapid orbit around the heavy object.

This rich variety of behaviors in the three-body problem underscores the challenge it presents for predictive modeling. The sensitive dependence on initial conditions and the complex interplay of gravitational forces create a dataset that will rigorously test the capabilities of our machine learning models, particularly in terms of their ability to generalize and make accurate long-term predictions.

4.2 Model Performance Comparison

Table 2 presents Mean Squared Error (MSE) results for three different models (MLP, LSTM, and PINN) across three problem types (Two-Body, Two-Body with Acceleration, and Three-Body) for various prediction horizons (10, 100, and 500 steps). Several interesting patterns emerge. In the Two-Body Problem, MLP and PINN generally outperform LSTM in position prediction, with PINN showing the lowest standard deviation in short-term predictions, indicating more consistent performance. All models perform similarly in velocity prediction, with slight improvements for longer horizons. For the Two-Body Problem with Acceleration, MLP consistently outperforms both LSTM and PINN across all prediction horizons, while PINN shows the highest MSE

and standard deviation, suggesting it struggles with the additional complexity. In the Three-Body Problem, PINN performs best in position prediction for short-term predictions (10 steps), while LSTM shows superior performance for longer prediction horizons (100 and 500 steps) in both position and velocity. Notably, MLP exhibits extremely high MSE for 100-step predictions in this scenario, indicating potential instability. Overall, these results suggest that different models excel in different scenarios: MLP performs well in simpler problems and with added acceleration, LSTM shows strength in long-term predictions for complex systems, and PINN demonstrates advantages in short-term predictions for the most complex (three-body) problem. These findings highlight the importance of choosing the appropriate model based on the specific characteristics of the orbital dynamics problem at hand.

For a visual representation of the models' performance, refer to Appendices B, C, and D, which contain trajectory plots comparing predicted and actual paths for the first spaceship in each scenario, providing a tangible illustration of the models' outputs across different problem types and prediction horizons.

Problem Model Set	10 Steps			100 Steps			500 Steps		
	Pos M (SD)	Vel M (SD)	Pos M (SD)	Vel M (SD)	Pos M (SD)	Vel M (SD)	Pos M (SD)	Vel M (SD)	Pos M (SD)
Two-Body	MLP	Tr 5.01e+7 (3.27e+6)	0.62 (1.78e-4)	5.15e+7 (3.80e+6)	0.59 (4.82e-4)	5.19e+7 (3.71e+6)	0.43 (7.64e-4)		
		Va 4.99e+7 (3.46e+6)	0.62 (1.91e-4)	5.08e+7 (3.59e+6)	0.59 (5.07e-4)	5.12e+7 (3.43e+6)	0.43 (8.21e-4)		
		Te 5.01e+7 (3.31e+6)	0.62 (1.80e-4)	5.14e+7 (3.77e+6)	0.59 (4.87e-4)	5.17e+7 (3.67e+6)	0.43 (7.76e-4)		
	LSTM	Tr 7.18e+7 (2.25e+7)	0.62 (5.33e-4)	7.74e+7 (2.43e+7)	0.59 (8.63e-4)	7.73e+7 (2.46e+7)	0.43 (8.41e-4)		
		Va 7.03e+7 (2.37e+7)	0.62 (5.71e-4)	7.64e+7 (2.59e+7)	0.59 (9.06e-4)	7.65e+7 (2.62e+7)	0.43 (9.00e-4)		
		Te 7.15e+7 (2.28e+7)	0.62 (5.41e-4)	7.72e+7 (2.46e+7)	0.59 (8.72e-4)	7.72e+7 (2.49e+7)	0.43 (8.53e-4)		
	PINN	Tr 5.03e+7 (2.16e+6) 0.62 (1.03e-4)	5.27e+7 (3.95e+6)	0.59 (5.13e-4)	5.06e+7 (5.47e+6)	0.43 (8.49e-4)			
		Va 5.00e+7 (2.11e+6) 0.62 (1.02e-4)	5.18e+7 (3.37e+6)	0.59 (4.99e-4)	4.92e+7 (4.16e+6)	0.43 (7.29e-4)			
		Te 5.03e+7 (2.15e+6) 0.62 (1.03e-4)	5.25e+7 (3.86e+6)	0.59 (5.11e-4)	5.03e+7 (5.27e+6)	0.43 (8.30e-4)			
Two-Body Accel.	MLP	Tr 4.29e+7 (4.64e+6) 0.58 (7.61e-5)	4.24e+7 (4.78e+6)	0.55 (1.78e-4)	4.31e+7 (3.58e+6)	0.40 (6.90e-4)			
		Va 4.32e+7 (4.93e+6) 0.58 (7.22e-5)	4.25e+7 (5.10e+6)	0.55 (1.57e-4)	4.33e+7 (3.78e+6)	0.40 (7.03e-4)			
		Te 4.30e+7 (4.70e+6) 0.58 (7.54e-5)	4.24e+7 (4.84e+6)	0.55 (1.70e-4)	4.31e+7 (3.62e+6)	0.40 (6.92e-4)			
	LSTM	Tr 4.92e+7 (2.72e+6)	0.58 (9.30e-5)	4.80e+7 (3.00e+6)	0.55 (1.81e-4)	4.82e+7 (2.85e+6)	0.40 (3.67e-4)		
		Va 4.89e+7 (2.78e+6)	0.58 (7.65e-5)	4.84e+7 (2.93e+6)	0.55 (1.82e-4)	4.85e+7 (2.87e+6)	0.40 (2.78e-4)		
		Te 4.91e+7 (2.73e+6)	0.58 (9.06e-5)	4.81e+7 (2.99e+6)	0.55 (1.79e-4)	4.82e+7 (2.86e+6)	0.40 (3.52e-4)		
	PINN	Tr 7.76e+7 (3.16e+7)	0.58 (4.57e-4)	7.32e+7 (2.30e+7)	0.55 (1.48e-3)	7.00e+7 (2.71e+7)	0.40 (1.13e-3)		
		Va 6.98e+7 (2.56e+7)	0.58 (4.53e-4)	6.85e+7 (2.07e+7)	0.55 (1.50e-3)	6.52e+7 (2.56e+7)	0.40 (1.05e-3)		
		Te 7.61e+7 (3.07e+7)	0.58 (4.57e-4)	7.23e+7 (2.27e+7)	0.55 (1.48e-3)	6.90e+7 (2.69e+7)	0.40 (1.12e-3)		
Three-Body	MLP	Tr 6.72e+8 (5.59e+8)	0.16 (9.30e-3)	4.03e+26 (1.04e+27)	4.90e+14 (1.26e+15)	9.26e+9 (1.54e+10)	0.15 (9.86e-3)		
		Va 4.26e+8 (3.63e+8)	0.16 (1.01e-2)	9.30e+25 (2.63e+26)	1.13e+14 (3.20e+14)	9.21e+9 (1.54e+10)	0.15 (9.79e-3)		
		Te 5.43e+8 (4.93e+8)	0.13 (2.22e-3)	1.52e+26 (3.04e+26)	1.86e+14 (3.71e+14)	9.27e+9 (1.53e+10)	0.14 (3.22e-3)		
	LSTM	Tr 6.63e+9 (6.26e+9)	0.15 (9.99e-3)	2.18e+10 (1.29e+10)	0.14 (1.07e-2)	2.47e+10 (1.46e+10)	0.14 (1.06e-2)		
		Va 7.23e+9 (7.67e+9)	0.15 (1.03e-2)	2.33e+10 (1.41e+10)	0.14 (1.04e-2)	2.64e+10 (1.60e+10)	0.14 (1.27e-2)		
		Te 3.72e+9 (1.31e+9)	0.13 (6.07e-3)	2.06e+10 (1.04e+10)	0.13 (6.82e-3)	2.40e+10 (1.28e+10)	0.13 (6.67e-3)		
	PINN	Tr 1.41e+9 (2.25e+9)	0.15 (9.40e-3)	2.38e+10 (4.51e+10)	0.15 (1.03e-2)	3.30e+10 (6.33e+10)	0.15 (1.06e-2)		
		Va 1.02e+9 (1.62e+9)	0.16 (1.05e-2)	2.65e+10 (4.66e+10)	0.15 (1.08e-2)	3.74e+10 (6.63e+10)	0.15 (1.30e-2)		
		Te 4.90e+8 (3.95e+8)	0.13 (2.41e-3)	2.29e+10 (4.26e+10)	0.13 (4.80e-3)	3.36e+10 (6.34e+10)	0.14 (6.14e-3)		

Table 2: Mean Squared Error (MSE) for different models across problem types, showing mean (M) and standard deviation (SD) for Position (Pos) and Velocity (Vel) at 10, 100, and 500 prediction steps on train (Tr), validation (Va), and test (Te) sets

5 Discussion

Our study on applying deep learning models to the two-body and three-body gravitational problems has yielded several noteworthy insights:

Model Performance Across Problem Types:

In the two-body problem, MLP and PINN outperformed LSTM in position prediction, with PINN showing the most consistent performance (lowest standard deviation) for short-term predictions. This suggests that for simpler orbital scenarios, the physical constraints incorporated in PINNs provide a significant advantage. For the two-body problem with increased acceleration, MLP consistently outperformed both LSTM and PINN. This unexpected result indicates that the additional complexity introduced by the sinusoidal force may be more easily captured by the flexible MLP architecture. In the three-body problem, PINN excelled in short-term predictions (10 steps), while LSTM showed superior performance for longer horizons (100 and 500 steps). This highlights the LSTM’s strength in capturing complex temporal dependencies in chaotic systems.

Impact of Physical Constraints in PINNs: The incorporation of energy and momentum conservation in the PINN loss function led to improved short-term predictions, especially in the complex three-body scenario. However, its performance degraded for longer prediction horizons, suggesting that the physical constraints might be overly restrictive for long-term predictions in chaotic systems. **Short-term vs. Long-term Predictions:** All models showed degrading performance as the prediction horizon increased from 10 to 500 steps. This degradation was most pronounced in the three-body problem, underscoring the challenge of long-term predictions in chaotic systems. **Model Robustness to Measurement Uncertainty:** The 1% relative uncertainty intro-

duced in our data generation process affected all models. However, PINNs showed more consistent performance (lower standard deviations) in the presence of this uncertainty, particularly for the two-body problem. **Effectiveness of Input Sequence Length:** Using the last 3 positions and velocities as input proved sufficient for short-term predictions but may be inadequate for longer horizons, especially in the three-body problem. This suggests that increasing the input sequence length could potentially improve long-term predictions, particularly for LSTM models. **Computational Efficiency:** While not explicitly measured, we observed that MLP models were generally faster to train compared to LSTM and PINN models. This efficiency could be a crucial factor in real-time applications or when computational resources are limited.

6 Conclusion

Our investigation into applying deep learning models to gravitational problems has yielded significant insights into the strengths and limitations of different architectures for predicting orbital dynamics. The MLP model demonstrated superior performance in the two-body problem with increased acceleration, achieving a mean squared error (MSE) of $4.30e+7$ for position and 0.40 for velocity in 500-step predictions on the test set. This outperformed both LSTM and PINN models, suggesting that MLPs are particularly effective for scenarios with additional forces beyond standard gravitation.

In the standard two-body problem, PINNs showed the most consistent performance, with the lowest standard deviation in short-term predictions ($2.15e+6$ for 10-step position predictions on the test set). This consistency underscores the value of incorporating physical constraints in the model architecture, particu-

larly for well-understood gravitational scenarios.

The LSTM model excelled in the three-body problem for longer prediction horizons, achieving an MSE of 2.40e+10 for position and 0.13 for velocity in 500-step predictions on the test set. This performance notably surpassed both MLP and PINN models, highlighting LSTM’s capability to capture complex temporal dependencies in chaotic systems.

However, all models showed degrading performance as the prediction horizon increased from 10 to 500 steps, particularly in the three-body problem. This degradation was most pronounced in the MLP model, which exhibited instability in 100-step predictions for the three-body problem with an MSE of 1.52e+26 for position.

The introduction of 1% measurement uncertainty affected all models, but PINNs demonstrated more robustness to this noise, particularly in the two-body problem. This suggests that physics-informed approaches may offer advantages in scenarios with noisy or imperfect data.

Our study has also revealed limitations in the current approach. The fixed input sequence of 3 time steps may be inadequate for longer-term predictions, especially in complex scenarios. Future work should explore adaptive input lengths and more sophisticated physical constraints in PINN models. Additionally, investigating model performance on real astronomical data would provide valuable insights into practical applicability.

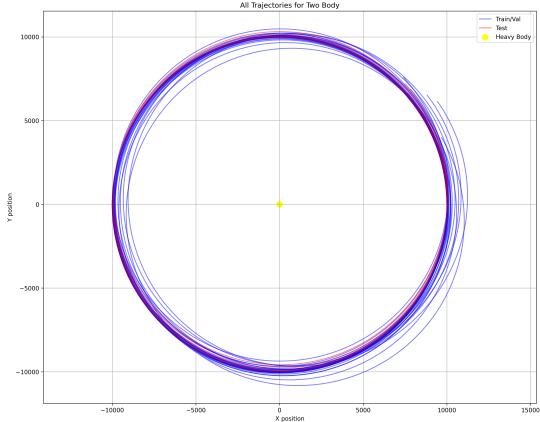
In conclusion, while our deep learning models show significant promise in solving gravitational problems, there remains substantial room for improvement, particularly in long-term predictions of chaotic systems. The MLP’s unexpected success in scenarios with additional forces, LSTM’s strength in complex temporal dynamics, and PINN’s consistency in physically constrained scenarios all point to-

wards the potential of hybrid models that combine these strengths. This study contributes to the ongoing effort to bridge the gap between traditional physics-based methods and modern machine learning techniques in celestial mechanics, paving the way for more accurate and physically consistent orbital predictions.

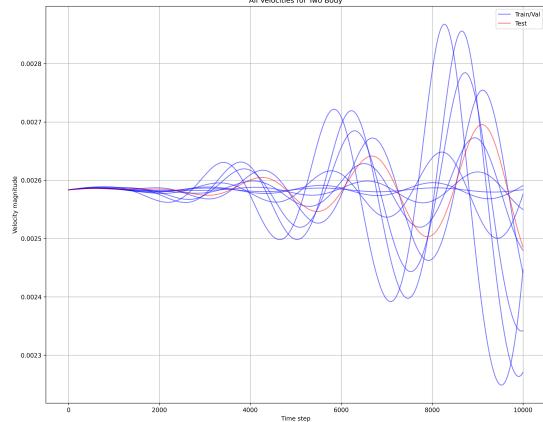
References

- [1] Lin Cheng, Zhenbo Wang, Yu Song, and Fanghua Jiang. Real-time optimal control for irregular asteroid landings using deep neural networks. *Acta Astronautica*, 170:66–79, 2020.
- [2] Ai Gao and Wentao Liao. Efficient gravity field modeling method for small bodies based on gaussian process regression. *Acta Astronautica*, 157:73–91, 2019.
- [3] Shijun Liao, Xiaoming Li, and Yu Yang. Three-body problem—from newton to supercomputer plus machine learning. *New Astronomy*, 96:101850, 2022.
- [4] John Martin and Hanspeter Schaub. Physics-informed neural networks for gravity field modeling of the earth and moon. *Celestial Mechanics and Dynamical Astronomy*, 134(2):13, 2022.
- [5] Chuizheng Meng, Sungyong Seo, Defu Cao, Sam Griesemer, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *arXiv preprint arXiv:2203.16797*, 2022.
- [6] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

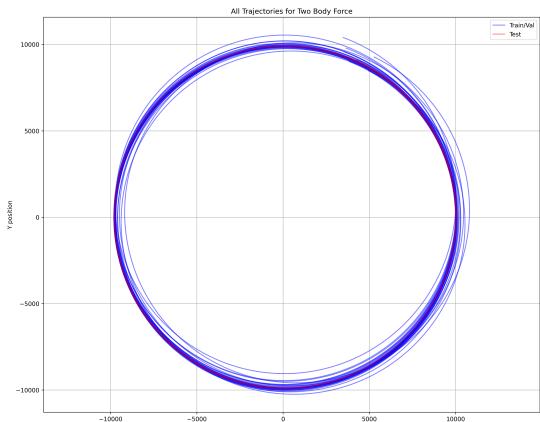
A Generated Trajectories and Velocities for Different Orbital Scenarios



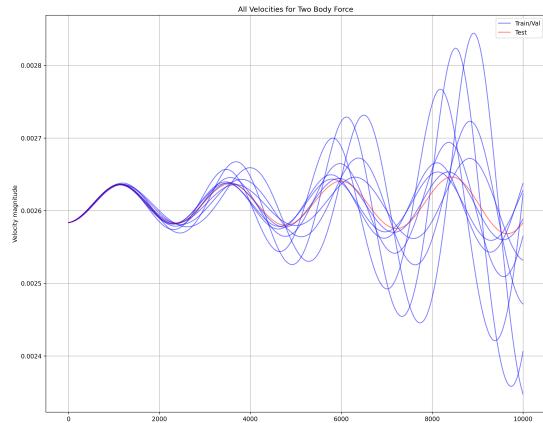
(a) Trajectories of all 10 spaceships in the two-body problem



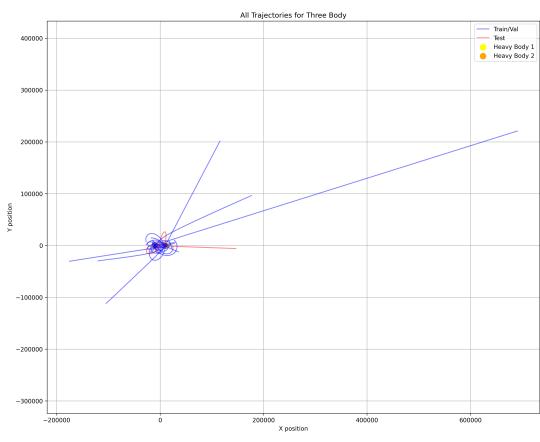
(b) Absolute velocity of all the 10 spaceships over time in the two-body problem



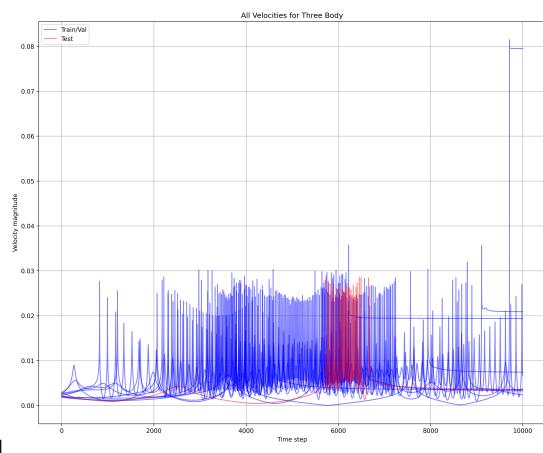
(c) Trajectories of all 10 spaceships in the two-body problem with increased acceleration



(d) Absolute velocity of all the 10 spaceships over time in the two-body problem with increased acceleration



(e) Trajectories of all 10 spaceships in the three-body problem

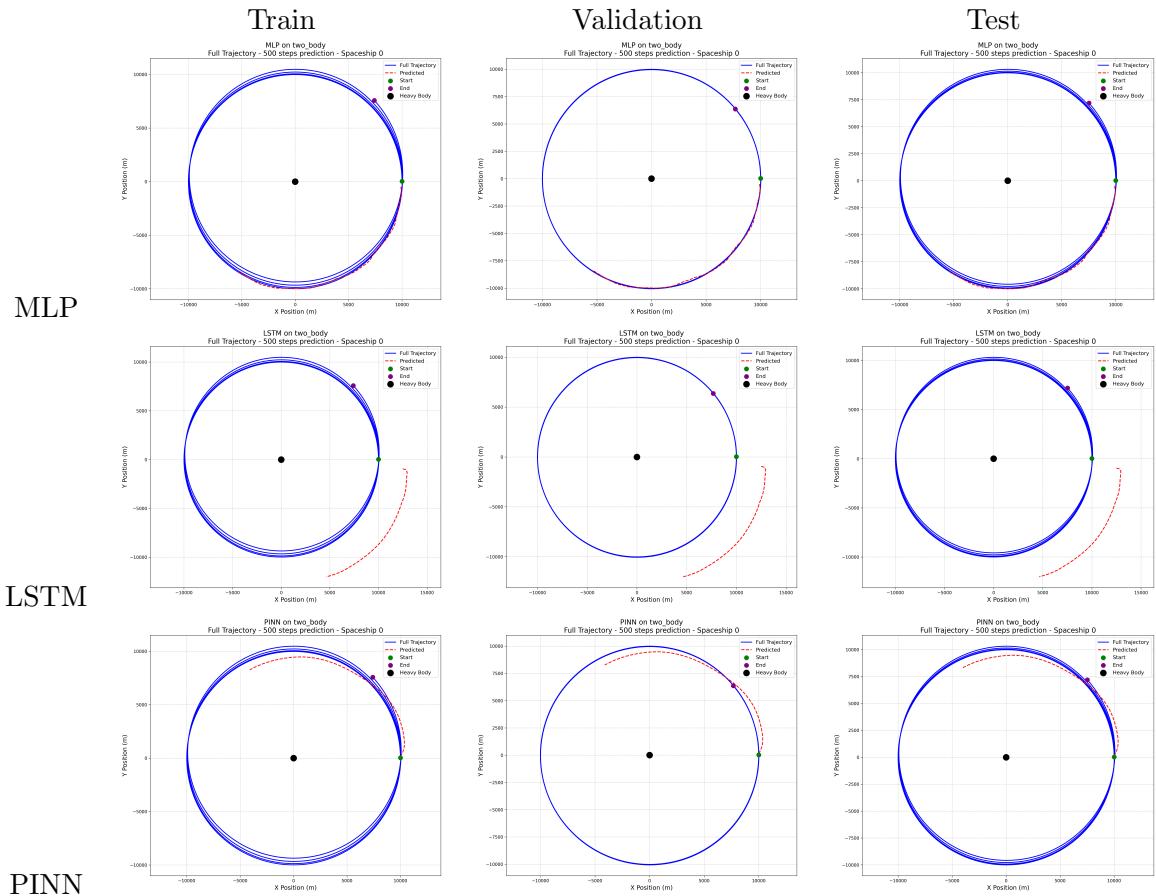


(f) Absolute velocity of all the 10 spaceships over time in the three-body problem

Figure 1: Trajectories and velocities of spaceships in different orbital scenarios

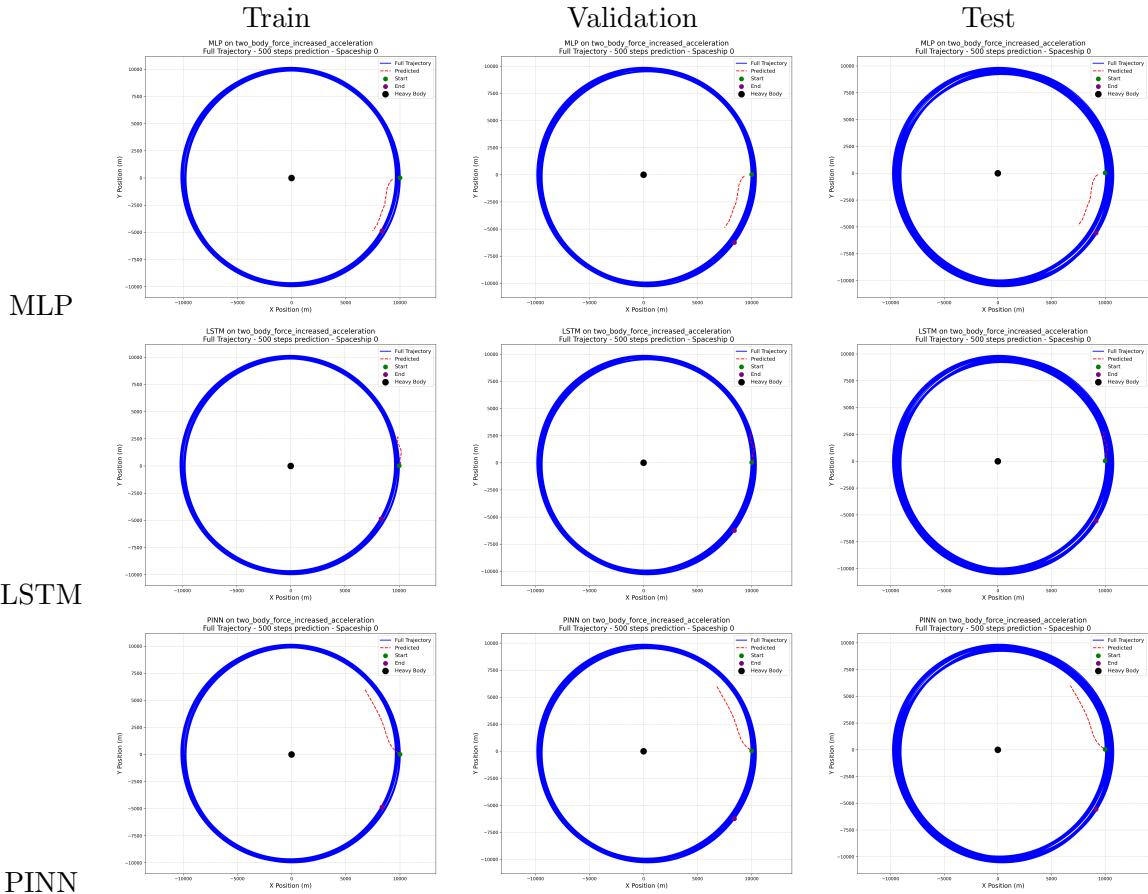
B Two-Body Problem Trajectory Predictions

Figure 2: Two-Body Problem: Model Predictions vs. Actual Trajectories for first spaceship



C Two-Body Problem with Increased Acceleration Trajectory Predictions

Figure 3: Two-Body Problem with Increased Acceleration: Model Predictions vs. Actual Trajectories for first spaceship



D Three-Body Problem Trajectory Predictions

Figure 4: Three-Body Problem: Model Predictions vs. Actual Trajectories for first spaceship

