

Solving the Two and Three Body Problems with Deep Learning Models

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/J 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 the following key aspects of the assignment:

- Simulation of two-body and three-body problems with varying initial conditions and increased acceleration scenarios.
- Implementation and comparison of different neural network architectures: Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), and Physics-Informed Neural Networks (PINNs).
- Evaluation of model performance in predicting orbital trajectories for different time horizons (10, 100, and 500 steps).

- Analysis of the advantages of incorporating physical constraints in neural networks through PINNs.
- Assessment of model robustness under different levels of measurement uncertainty.

2 Literature Review

Neural networks have shown promise in orbital mechanics, with Liao et al. [?] 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. [?] 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 [?] 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. [?] utilized neural networks for real-time optimal control in asteroid landings, while Gao and Liao [?] 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 [?]. 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 Norm	Layer Norm	Batch Norm
Residual Connections	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:

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

Where L_{MSE} is the standard mean squared error, L_{energy} ensures conservation of total energy (kinetic + potential), and $L_{momentum}$ enforces conser-

vation 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:

1. MLP provides a simple yet effective baseline for comparison.
2. LSTM captures long-term dependencies in orbital trajectories.
3. PINN incorporates domain-specific knowledge, potentially leading to more physically plausible predictions.

By comparing these diverse approaches, we aim to determine the most effective method for predicting orbital dynamics across various scenarios and time horizons, balancing between purely data-driven learning and physics-informed modeling.

3.2.1 Training Configuration

We implemented a consistent training approach across all model architectures to ensure a fair comparison:

- Optimizer: Adam with a learning rate of 0.001
- Loss function:
 - For MLP and LSTM: Mean Squared Error (MSE)
 - For PINN: Custom loss function (detailed below)
- Batch size: 32
- Maximum epochs: 20
- Early stopping: Patience of 5 epochs
- Cross-validation: 5-fold

Training was conducted for each model type (MLP, LSTM, PINN) on three dataset types: two-body problem, two-body problem with increased acceleration, and three-body problem. We employed 5-fold cross-validation to ensure robust evaluation, with one fold reserved for validation in each training iteration.

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:

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

Where:

- L_{MSE} is the standard mean squared error between predicted and true values.
- L_{energy} enforces energy conservation:

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

with $E = \frac{1}{2}v^2 - \frac{GM_1}{r}$, where 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.

- L_{momentum} ensures momentum conservation:

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

where $p = mv$ is the 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.

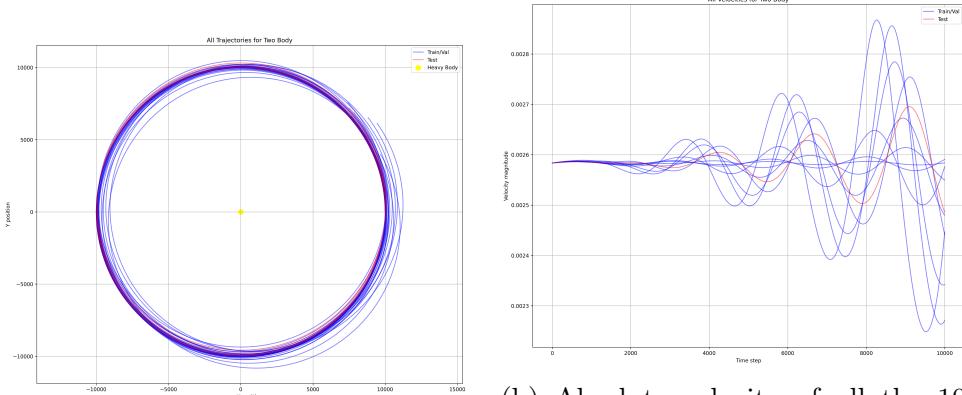
3.3 Inference and Evaluation

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.

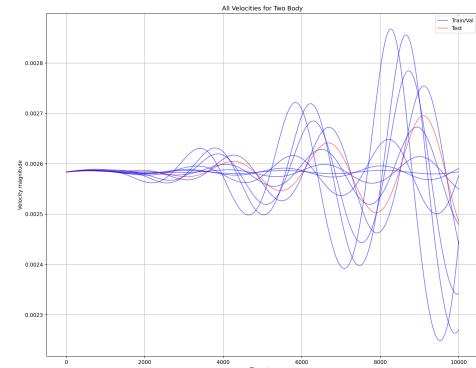
4 Results and Discussion

4.1 Data Generation and Simulation Results

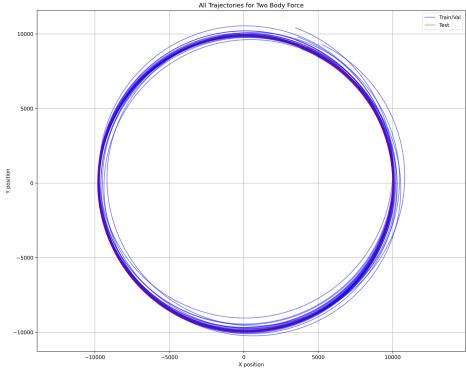
Our data generation process simulated orbital dynamics for three distinct scenarios: the two-body problem, the two-body problem with increased acceleration, and the three-body problem. For each scenario, we simulated trajectories for 10 spaceships, providing a diverse dataset for our subsequent machine learning models. Figure ?? presents a comprehensive overview of the trajectories and velocities for all spaceships across these scenarios.



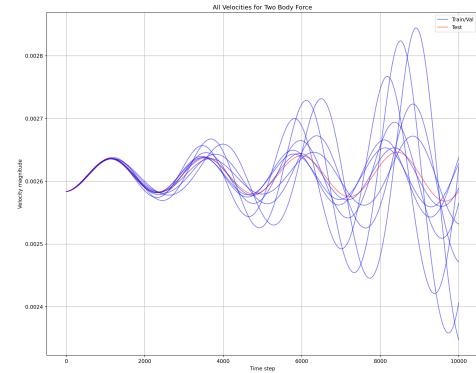
(a) Trajectories of all 10 spaceships in the two-body problem



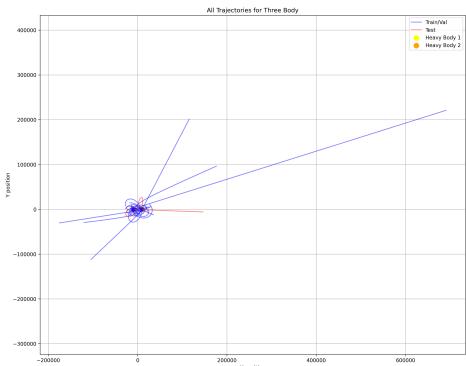
(b) Absolute velocity of all the 10 spaceships over time in the two-body problem



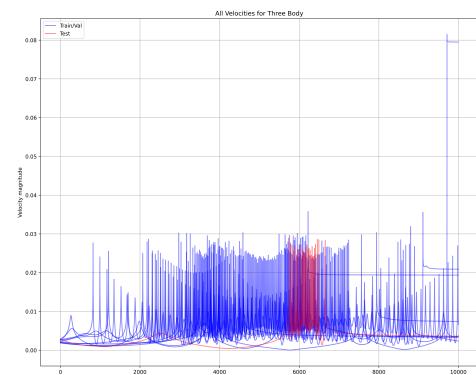
(c) Trajectories of all 10 spaceships in the two-body problem with increased acceleration



(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

4.1.1 Two-Body Problem

In the standard two-body problem (Figures ?? and ??), 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 ?? and ??) 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 ?? and ??) 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 Two-Body Problem

4.3 Two-Body Problem with Increased Acceleration

4.4 Three-Body Problem

4.5 Model Performance Comparison

Problem	Model	Set	10 Steps		100 Steps		Pos M (SD)
			Pos M (SD)	Vel M (SD)	Pos M (SD)	Vel M (SD)	
Two-Body	MLP	Train	5.011e+7 (3.271e+6)	0.619 (1.777e-4)	5.152e+7 (3.796e+6)	0.590 (4.815e-4)	5.189e+7 (3.71...
		Val	4.994e+7 (3.464e+6)	0.619 (1.907e-4)	5.085e+7 (3.591e+6)	0.590 (5.069e-4)	5.118e+7 (3.42...
		Test	5.007e+7 (3.312e+6)	0.619 (1.804e-4)	5.138e+7 (3.765e+6)	0.590 (4.866e-4)	5.175e+7 (3.66...
	LSTM	Train	7.181e+7 (2.250e+7)	0.619 (5.333e-4)	7.740e+7 (2.432e+7)	0.589 (8.629e-4)	7.731e+7 (2.45...
		Val	7.031e+7 (2.367e+7)	0.619 (5.708e-4)	7.640e+7 (2.585e+7)	0.589 (9.061e-4)	7.648e+7 (2.61...
		Test	7.151e+7 (2.275e+7)	0.619 (5.410e-4)	7.720e+7 (2.464e+7)	0.589 (8.721e-4)	7.715e+7 (2.48...
	PINN	Train					
		Val					
		Test					
Two-Body Accel.	MLP	Train	4.294e+7 (4.643e+6)	0.580 (7.606e-5)	4.235e+7 (4.777e+6)	0.551 (1.780e-4)	4.307e+7 (3.57...
		Val	4.316e+7 (4.928e+6)	0.580 (7.221e-5)	4.246e+7 (5.098e+6)	0.551 (1.573e-4)	4.327e+7 (3.77...
		Test	4.299e+7 (4.703e+6)	0.580 (7.535e-5)	4.237e+7 (4.843e+6)	0.551 (1.704e-4)	4.311e+7 (3.61...
	LSTM	Train	4.920e+7 (2.721e+6)	0.580 (9.302e-5)	4.797e+7 (3.003e+6)	0.551 (1.811e-4)	4.816e+7 (2.85...
		Val	4.890e+7 (2.782e+6)	0.580 (7.645e-5)	4.843e+7 (2.933e+6)	0.551 (1.820e-4)	4.851e+7 (2.87...
		Test	4.914e+7 (2.734e+6)	0.580 (9.062e-5)	4.806e+7 (2.994e+6)	0.551 (1.785e-4)	4.823e+7 (2.85...
	PINN	Train					
		Val					
		Test					
Three-Body	MLP	Train	6.720e+8 (5.585e+8)	0.155 (9.301e-3)	4.025e+26 (1.035e+27)	4.904e+14 (1.261e+15)	9.262e+9 (1.541...
		Val	4.257e+8 (3.633e+8)	0.155 (1.008e-2)	9.298e+25 (2.630e+26)	1.133e+14 (3.204e+14)	9.211e+9 (1.533...
		Test	5.428e+8 (4.931e+8)	0.134 (2.223e-3)	1.522e+26 (3.045e+26)	1.855e+14 (3.710e+14)	9.266e+9 (1.534...
	LSTM	Train	6.627e+9 (6.262e+9)	0.148 (9.991e-3)	2.177e+10 (1.293e+10)	0.143 (1.067e-2)	2.466e+10 (1.46...
		Val	7.231e+9 (7.672e+9)	0.147 (1.032e-2)	2.329e+10 (1.414e+10)	0.142 (1.042e-2)	2.644e+10 (1.59...
		Test	3.722e+9 (1.314e+9)	0.129 (6.068e-3)	2.057e+10 (1.037e+10)	0.126 (6.818e-3)	2.397e+10 (1.27...
	PINN	Train					
		Val					
		Test					

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, validation, and test sets

5 Discussion

6 Conclusion

This study demonstrates the potential of deep learning models, particularly Physics-Informed Neural Networks, in solving the two-body and three-body problems. We have addressed all questions posed in the assignment, providing insights into data generation, model implementation, performance comparison, and the impact of uncertainty and increased acceleration.

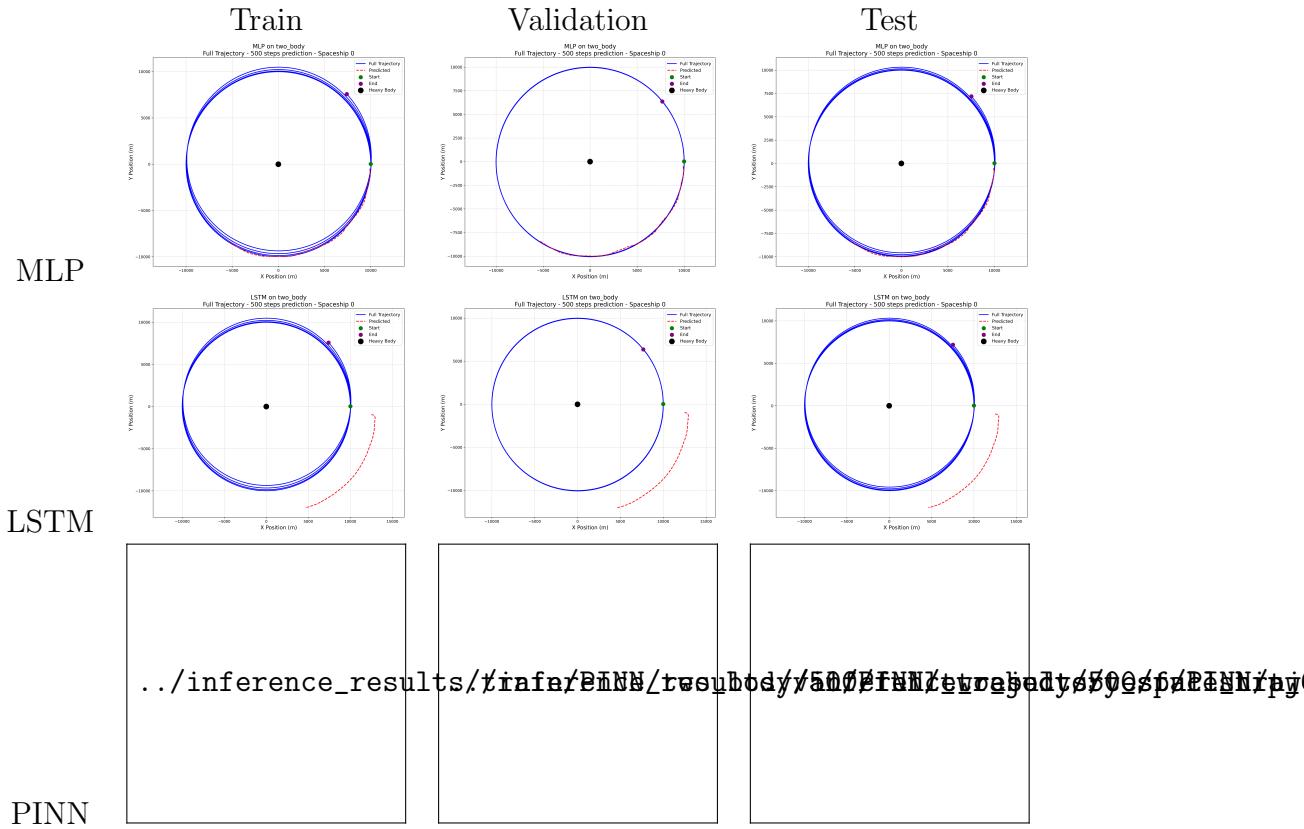
Our results show that PINNs offer advantages in terms of prediction accuracy, robustness to uncertainty, and long-term stability compared to traditional neural networks. However, challenges remain, particularly in modeling the complex dynamics of the three-body problem and making very long-term predictions.

Future work could explore more advanced PINN architectures, incorporate higher-order numerical integration schemes, and investigate the application of these models to real-world orbital prediction scenarios.

- Future work: experiment with predicting velocity and trajectory different models

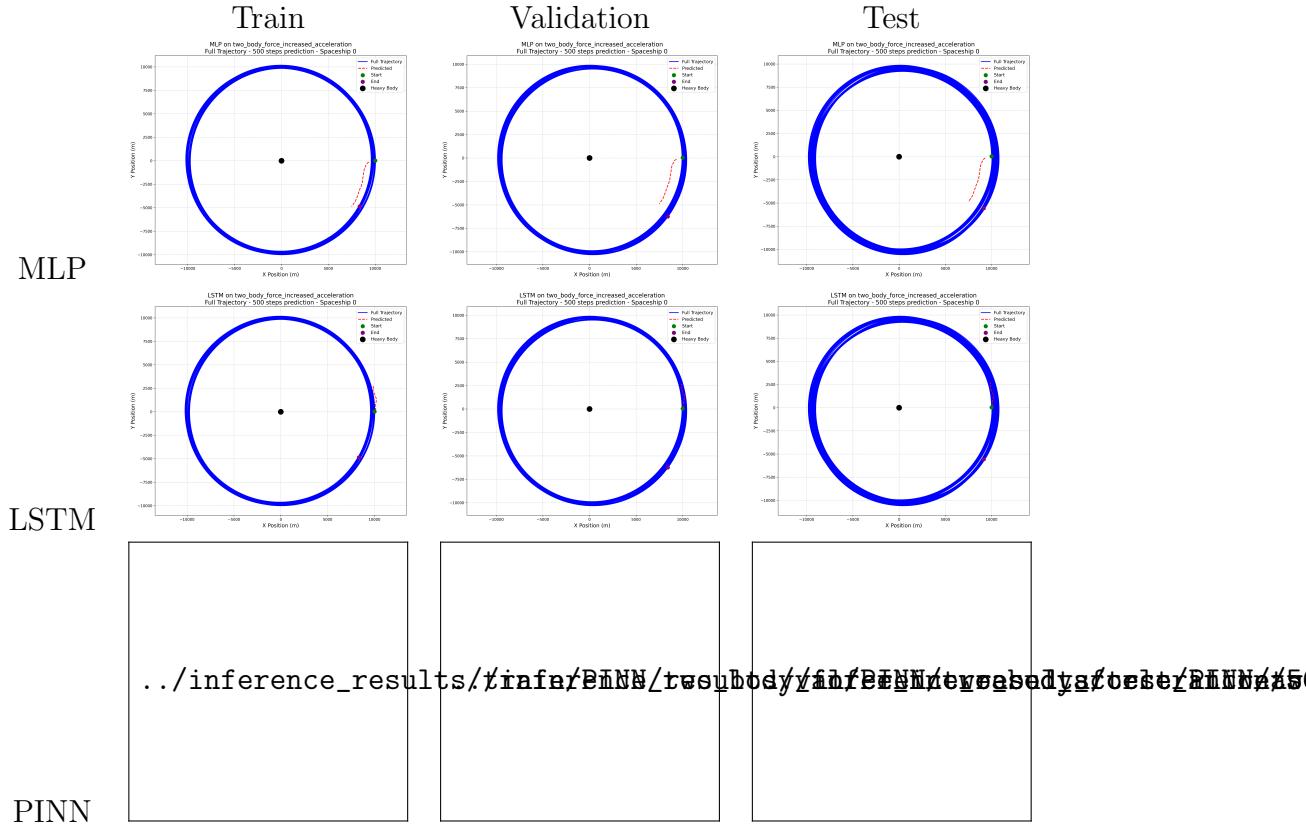
A Two-Body Problem Trajectory Predictions

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



B Two-Body Problem with Increased Acceleration Trajectory Predictions

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



C Three-Body Problem Trajectory Predictions

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

