

Random Feature Hamiltonian Networks for N-Body Problems

Bachelor's Thesis Talk

Sofia Koufogazou Loukianova

Advisor: Atamert Rahma

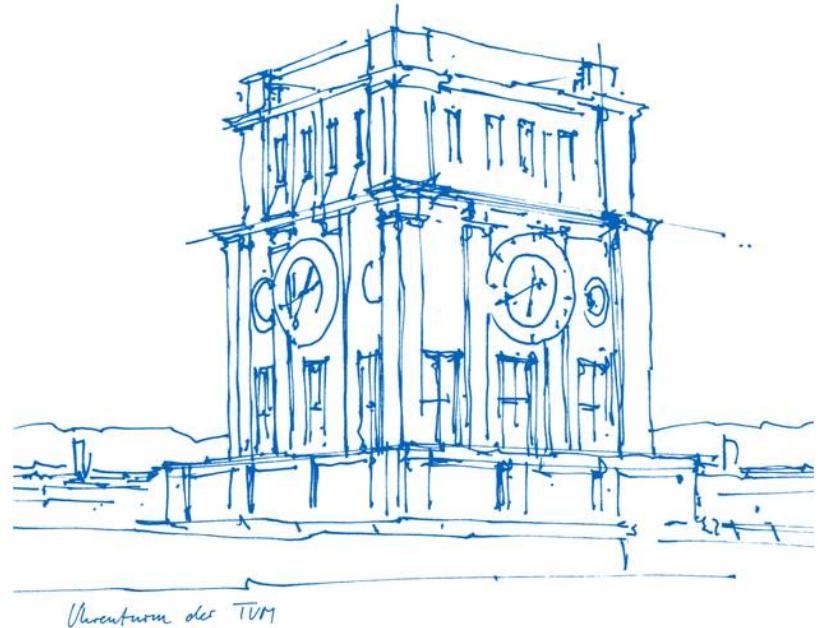
Examiner: Dr. Felix Dietrich

Technical University of Munich

TUM School of Computation, Information and Technology

Chair of Scientific Computing

January 28th, 2026



Outline

- 1 Introduction
- 2 State of the art
- 3 Random Feature Hamiltonian Networks for N-Body Systems
- 4 Conclusion and future work

Introduction

Newton, I. (1684–1685) on predicting the motions of $N > 2$ bodies:

“... to define these motions by exact laws admitting of easy calculation exceeds, if I am not mistaken, the force of any human mind”. [1]

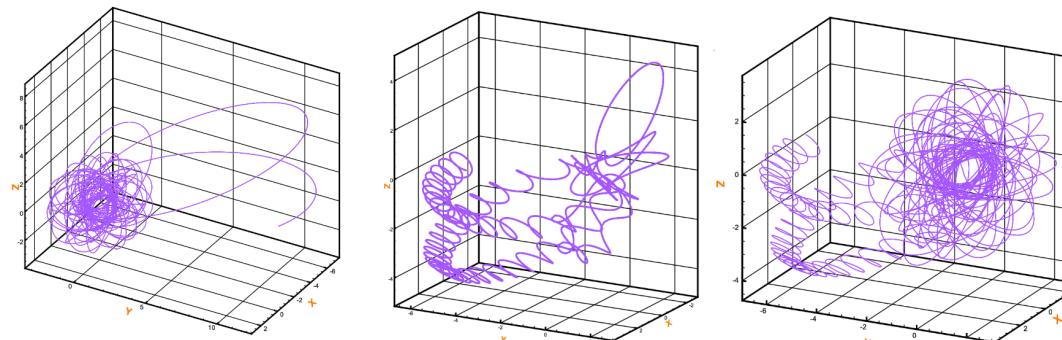


Figure 1 Chaotic orbits of a three-body system, between body 1 (left), body 2 (middle) and body 3 (right). [2]

Outline

1 Introduction

2 State of the art

3 Random Feature Hamiltonian Networks for N-Body Systems

4 Conclusion and future work

State of the art — Hamiltonian Systems

A **Hamiltonian system** [2] defines a first-order ODE system of the form

$$\dot{\mathbf{z}} = J \nabla \mathcal{H}(\mathbf{z}, t), \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

The vector $\mathbf{z} = (q, p)$ consists of positions q and momenta p , with

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p}, \quad \dot{p} = -\frac{\partial \mathcal{H}}{\partial q}$$

Separable Hamiltonian: $\mathcal{H}(q, p) = \mathcal{T}(p) + \mathcal{V}(q)$

Key properties:

- Hamiltonian systems conserve energy: $\frac{d\mathcal{H}}{dt} = \frac{\partial \mathcal{H}}{\partial q} \cdot \dot{q} + \frac{\partial \mathcal{H}}{\partial p} \cdot \dot{p} = 0$
- Symplecticity of the flow φ^t

State of the art — The N-Body Problem

The gravitational N -body problem describes the motion of N point masses under pairwise Newtonian gravity.

- The classical equations of motion:

$$\dot{q}_i = p_i, \quad \dot{p}_i = -G \sum_{\substack{j=1 \\ j \neq i}}^N m_j \frac{q_i - q_j}{\|q_i - q_j\|^3}.$$

- The N-body Hamiltonian:

$$\mathcal{H}(q, p) = \underbrace{\sum_{i=1}^N \frac{\|p_i\|^2}{2m_i}}_{\mathcal{T}(p)} + \underbrace{\left(-G \sum_{1 \leq i < j \leq N} \frac{m_i m_j}{\|q_i - q_j\|} \right)}_{\mathcal{V}(q)}$$

- Derivation of Hamilton's equations of motion:

$$\dot{q}_i = \frac{\partial H}{\partial p_i} = \frac{p_i}{m_i} \stackrel{m_i=1}{=} p_i, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i} = -G \sum_{\substack{j=1 \\ j \neq i}}^N \frac{m_i m_j (q_i - q_j)}{\|q_i - q_j\|^3}$$

State of the art — The N-Body Problem

The two-body problem

For $N = 2$ the problem admits a **closed-form solution**.

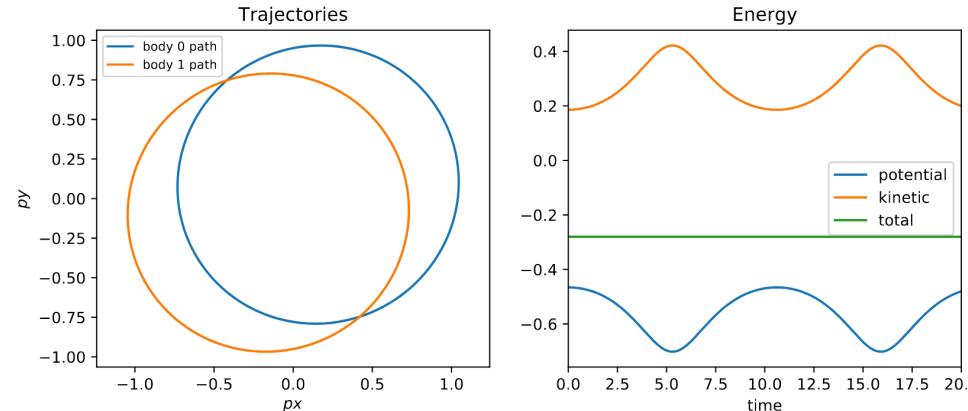


Figure 2 A representative two-body trajectory and its corresponding energy evolution. *Left:* stable motion about the center of mass. *Right:* kinetic, potential, and total energies, showing energy conservation. Taken from [3]

State of the art — The N-Body Problem

The three-body problem

For $N > 2$ the problem admits a **no closed-form solution**.
However, stable periodic trajectories exist for some initial conditions.

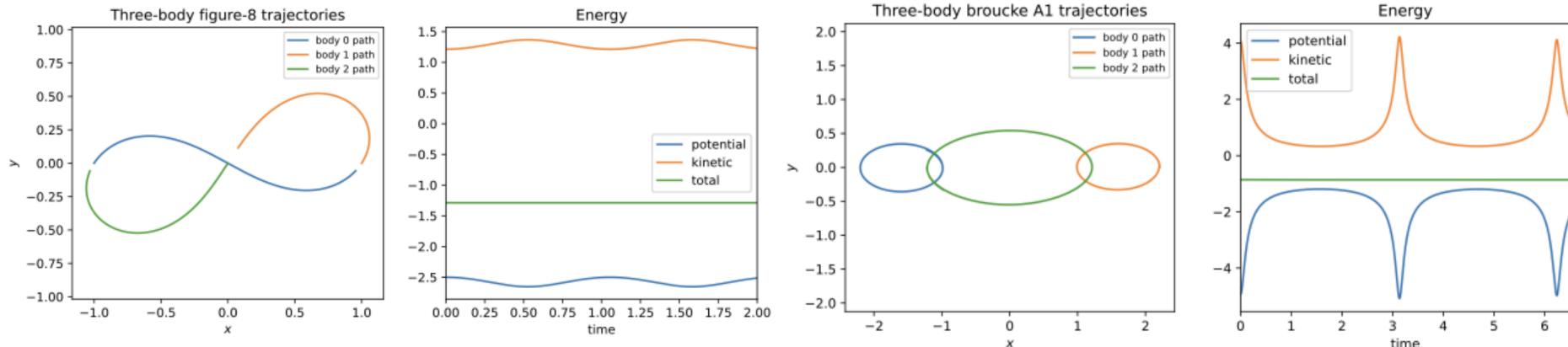


Figure 3 Stable periodic configurations of Figure-8 (*left*) and Broucke A1 (*right*) trajectories, where all three bodies follow a single closed curve while conserving total energy.

State of the art — Numerical Integration

For a dynamical systems with flow map φ^t such that $z(t) = \varphi^t(z_0)$, numerical integration:

- Approximates the continuous time evolution with a **discrete sequence** $\{z_n\}_{n \geq 0}$
- Doesn't require analytical solutions of the equations of motion

Classical Integration Methods:

- **Explicit Euler**

$$\begin{cases} q_{n+1} = q_n + h \nabla_p H(q_n, p_n). \\ p_{n+1} = p_n - h \nabla_q H(q_n, p_n) \end{cases}$$

- **Runge-Kutta methods (RK4)**

State of the art — Numerical Integration

For a dynamical systems with flow map φ^t such that $z(t) = \varphi^t(z_0)$, numerical integration:

- Approximates the continuous time evolution with a **discrete sequence** $\{z_n\}_{n \geq 0}$
- Doesn't require analytical solutions of the equations of motion

Symplectic Integration Methods:

- **Symplectic Euler**

$$\begin{cases} q_{n+1} = q_n + h \nabla_p H(p_{n+1}, q_n) \\ p_{n+1} = p_n - h \nabla_q H(p_{n+1}, q_n) \end{cases}$$

- **Implicit Midpoint**

$$\begin{cases} q_{n+1} = q_n + h \nabla_p H\left(\frac{q_n + q_{n+1}}{2}, \frac{p_n + p_{n+1}}{2}\right) \\ p_{n+1} = p_n - h \nabla_q H\left(\frac{q_n + q_{n+1}}{2}, \frac{p_n + p_{n+1}}{2}\right) \end{cases}$$

- **Störmer Verlet**

State of the art — Feed Forward Neural Networks

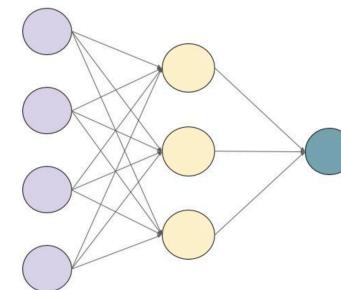
Goal: Learn system dynamics directly from data

- FFNN $\Phi : X \rightarrow Y$ in regression settings: approximating an unknown target function

$$f(x) \approx \Phi_\theta(x) \quad (\text{e.g. the vector field } F_\theta(z) \text{ or the Hamiltonian } \mathcal{H})$$

- Output of the l -th layer:

$$\Phi^{(l)}(x) = \begin{cases} x, & \text{for } l = 0, \\ \sigma\left(W_l \Phi^{(l-1)}(x) - b_l\right), & \text{for } 0 < l \leq L, \\ W_l \Phi^{(l-1)}(x) - b_l, & \text{for } l = L + 1, \end{cases}$$



- Parameters $\theta = \{W_l, b_l\}_{l=1}^{L+1}$ represent the weights and biases
- Loss $\mathcal{L}(\theta)$ measures prediction error on data
- Gradient descent iteratively updates parameters $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_{\theta} \mathcal{L}(\theta^{(t)})$

State of the art — Neural ODEs

Goal: Learn continuous-time system dynamics from data

- Neural ODEs (NODEs) learn the vector field $F_\theta(z)$ directly from data, where

$$\frac{dz}{dt} = F_\theta(z)$$

- Trajectory prediction: $\{\hat{z}_i(\theta)\}_{i=0}^T = \text{ODESolve}\left(F_\theta, z_0, \{t_i\}_{i=0}^T\right)$
- The parameters θ can be learned by minimizing: $\mathcal{L}_2(\theta) = \sum_{i=1}^T \|z_i - \hat{z}_i\|_2$
- Adjust sensitivity method is used instead of backpropagation
- **Limitation:** For physical systems with conserved quantities, this model doesn't learn the exact conservation laws

State of the art — Hamiltonian Neural Networks (HNNs)

Goal: Learn system dynamics from data while **preserving physical structure** (energy)

- HNNs learn a Hamiltonian $\mathcal{H}_\theta(q, p) \approx \mathcal{H}(q, p)$, and output $\hat{z}_\theta(q, p) = J\nabla\hat{\mathcal{H}}_\theta(q, p)$

- Loss function:

$$\mathcal{L}_{\text{HNN}} = \left\| \frac{\partial \mathcal{H}_\theta}{\partial p} - \frac{dq}{dt} \right\|_2 + \left\| \frac{\partial \mathcal{H}_\theta}{\partial q} + \frac{dp}{dt} \right\|_2$$

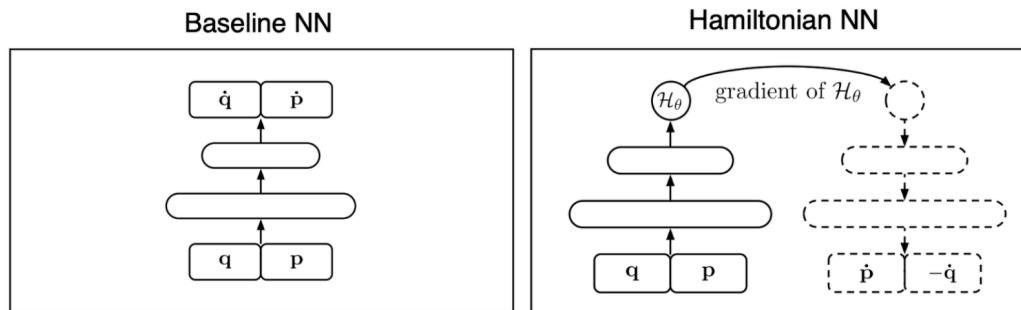


Figure 4 *Left:* NODE learns (\dot{q}, \dot{p}) directly from (q, p) . *Right:* HNN learns a scalar Hamiltonian \mathcal{H}_θ and recovers the dynamics via Hamilton's equations. Taken from [3]

State of the art — Random Feature Hamiltonian Neural Networks (RF-HNNs)

Goal: Reduce training cost of HNNs while preserving Hamiltonian structure

- Standard HNNs use gradient-descent training
 - Expensive, slow convergence
- RF-HNNs **sample hidden layer parameters** and only train last layer
 - Learning becomes a linear least squares problem
 - No backpropagation through hidden layers
 - Performance comparable to fully trained NNs
 - Orders-of-magnitude faster training!

State of the art — Random Feature Hamiltonian Neural Networks (RF-HNNs)

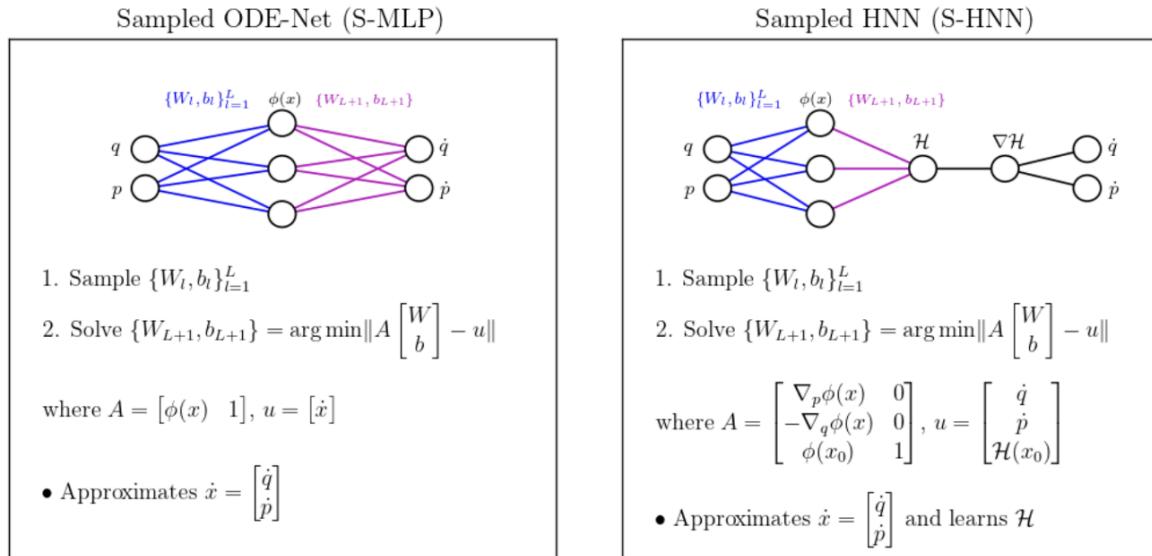


Figure 5: A comparison between RF-NODE (*left*) and RF-HNN (*right*), taken from [4]

State of the art — Random Feature Hamiltonian Neural Networks (RF-HNNs)

Sampling strategies

- **Data-agnostic** (ELM): $W_l \sim \mathcal{N}(0, I), b_l \sim \mathcal{U}[-1, 1]$
- **Data-driven** (SWIM): $w_{l,i} = s_1 \frac{x_{l-1,i}^{(2)} - x_{l-1,i}^{(1)}}{\|x_{l-1,i}^{(2)} - x_{l-1,i}^{(1)}\|^2}, \quad b_{l,i} = \langle w_{l,i}, x_{l-1,i}^{(1)} \rangle + s_2.$

Variants:

- SWIM: $(x^{(1)}, x^{(2)})$ sampled according to gradient magnitude of target function
- U-SWIM: SWIM algorithm with point pairs uniformly sampled from input space
- A-SWIM: SWIM with point pairs sampled according to U-SWIM target function predictions

Outline

- 1 Introduction
- 2 State of the art
- 3 Random Feature Hamiltonian Networks for N-Body Systems
- 4 Conclusion and future work

RF-HNNs for N-Body Systems — Motivation

Prior Work:

- HNNs introduced for 2- and 3-body systems in previous works [3]
- However, non-symplectic numerical integration used (RK4)
- RF-HNN studies [4] focused on simpler systems:
(spring-mass, Hénon-Heiles, Lotka-Voltera)

In this work we aim to:

- Evaluate RF-HNNs on 2-body and 3-body systems
- Perform forward simulation using symplectic integrators
- Compare against NODEs and standard HNNs
- Study the effect of different sampling strategies

RF-HNNs for N-Body Systems — Two-body experiments

Orbit data generation

System state: $z = (q_{x,1}, q_{y,1}, q_{x,2}, q_{y,2}, p_{x,1}, p_{y,1}, p_{x,2}, p_{y,2}) \in \mathbb{R}^8$ with unit masses.

Initial conditions (near circular orbits):

- Positions: $q_1(0) \sim \text{Uniform}([0.5, 1.5]^2)$

$$q_2(0) = -q_1(0)$$

- Momenta: $p_1(0) \propto \begin{pmatrix} -y_1(0) \\ x_1(0) \end{pmatrix} (1 + \text{noise}), \quad \text{noise} \sim \text{Uniform}(-1, 1) \sigma_{\text{orbit}}$

$$p_2(0) = -p_1(0)$$

- Numerical integrator evolves the state in 2D for each sampled initial condition
- Final dataset contains $(z_i, \dot{z}_i, H(z_i), \nabla H(z_i))$

RF-HNNs for N-Body Systems — Two-body experiments

Training results

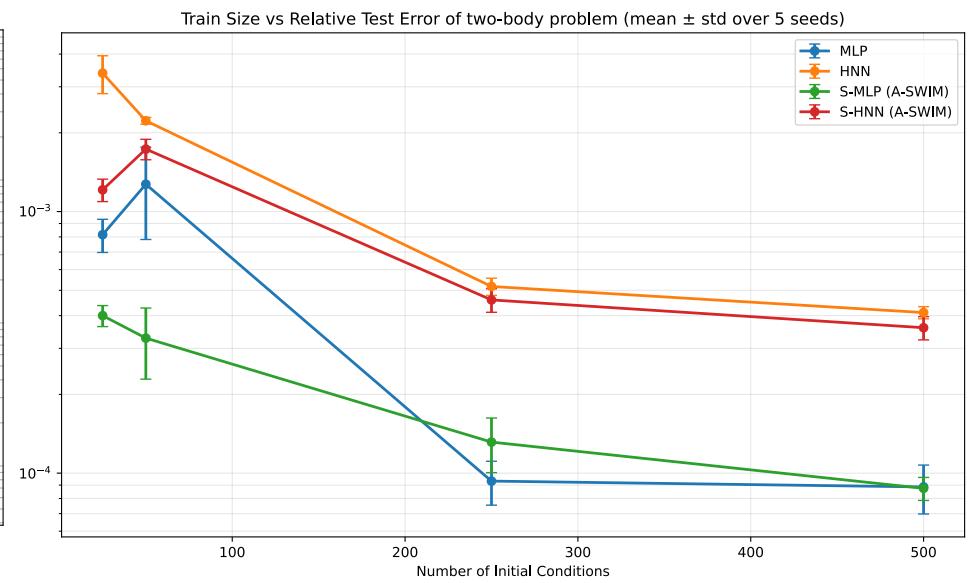
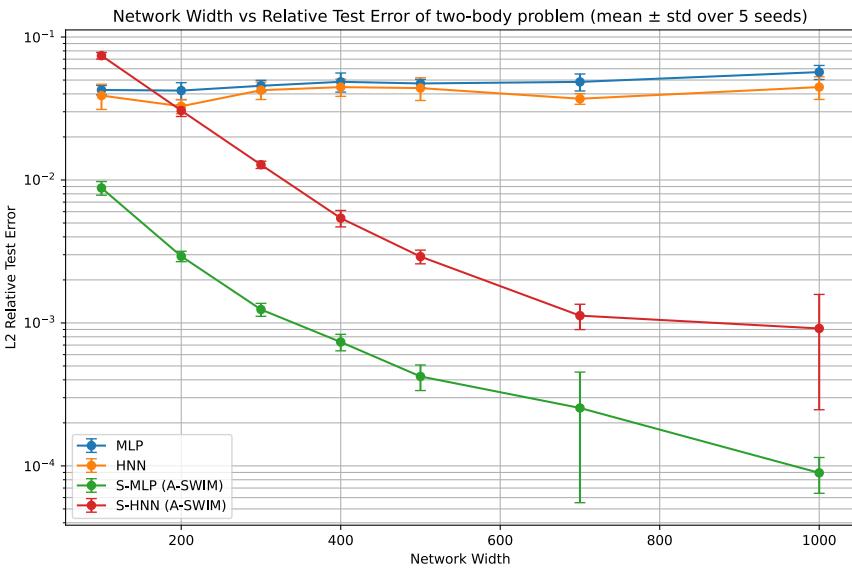
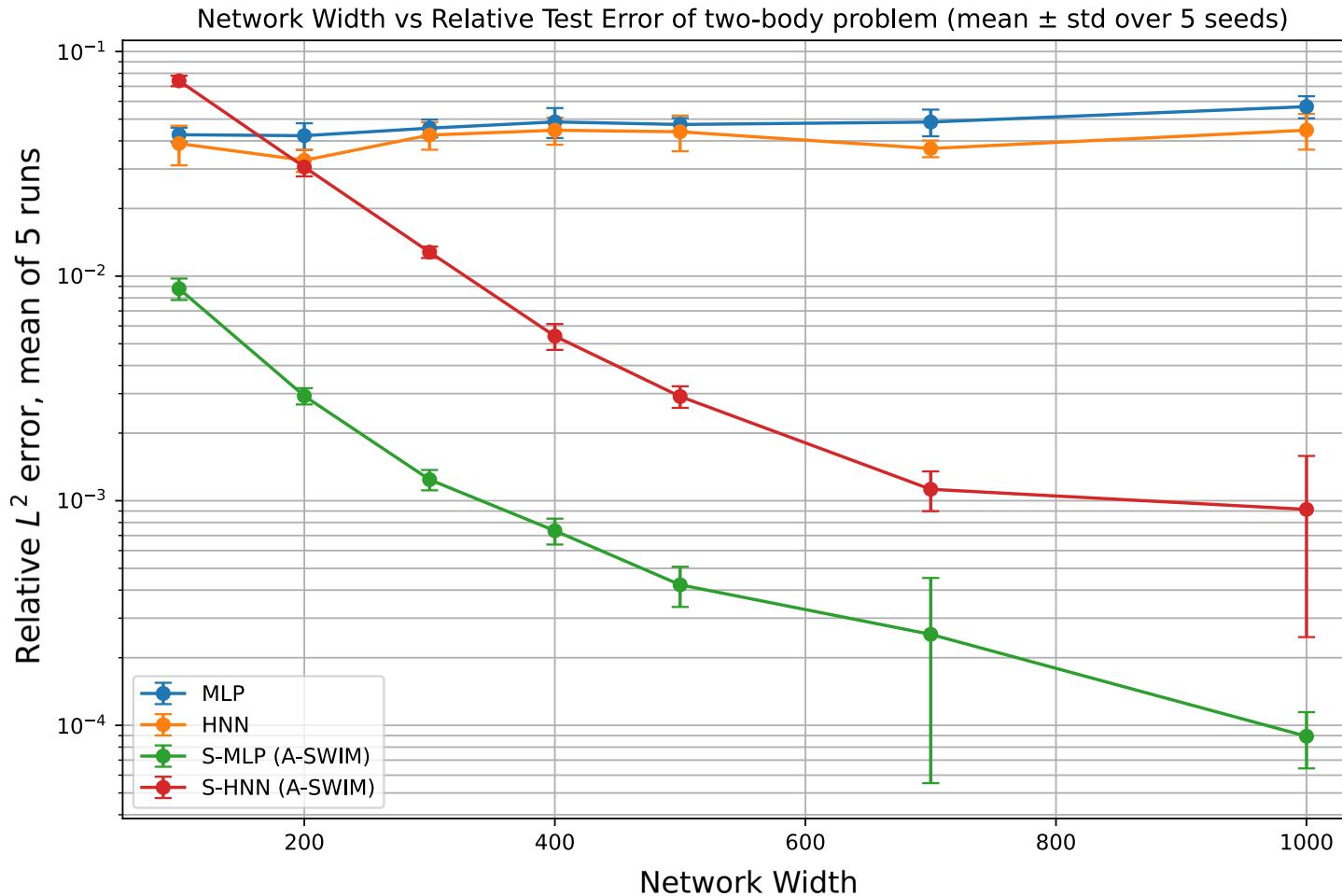
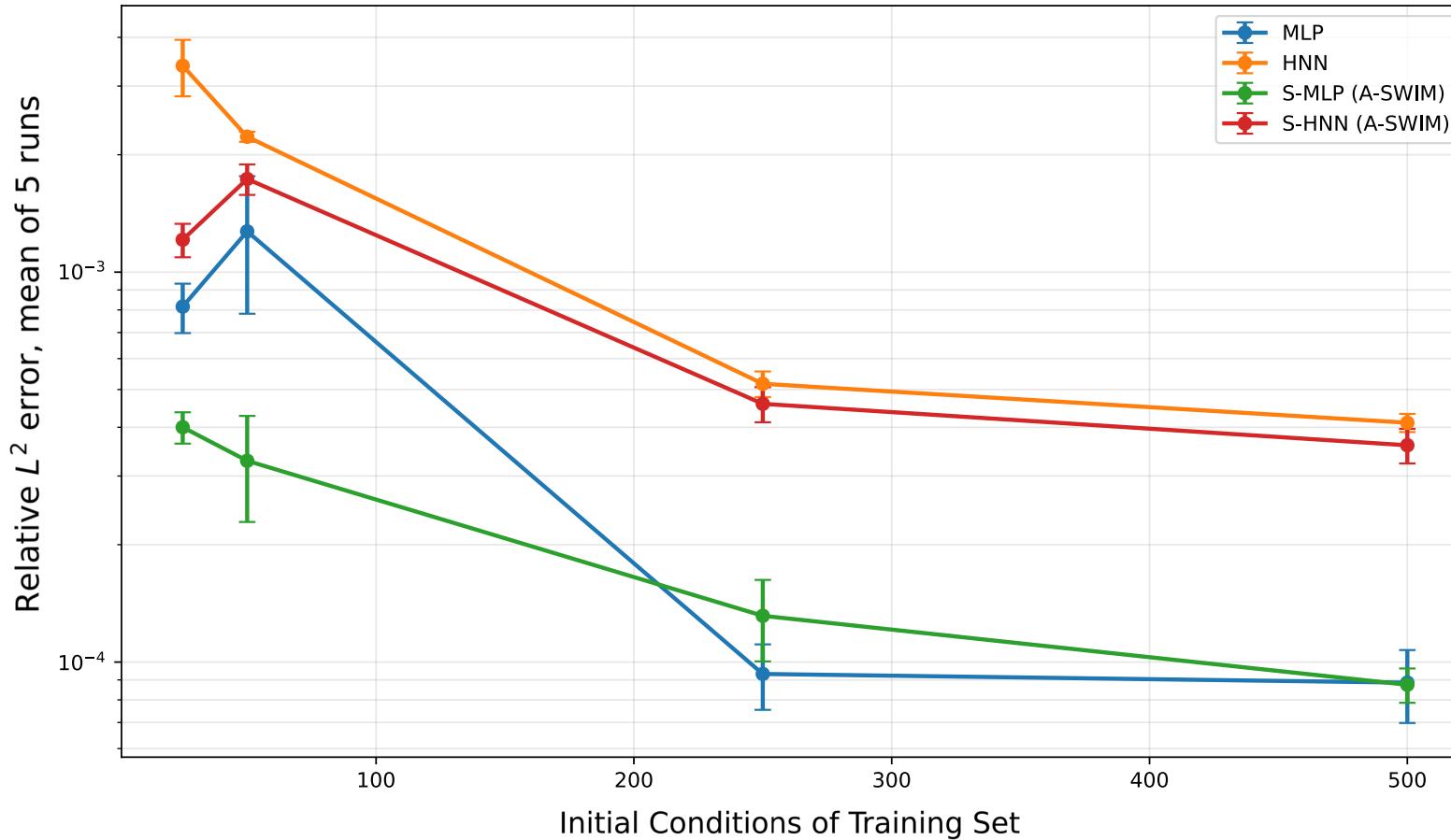


Figure 6 Left: Relative test error is plotted against network width, using train size 15000.
Right: Relative test error is plotted against train size, using width 1000.



Train Size vs Relative Test Error of two-body problem (mean \pm std over 5 seeds)

RF-HNNs for N-Body Systems — Two-body experiments

S-HNN Model: Trajectory simulation

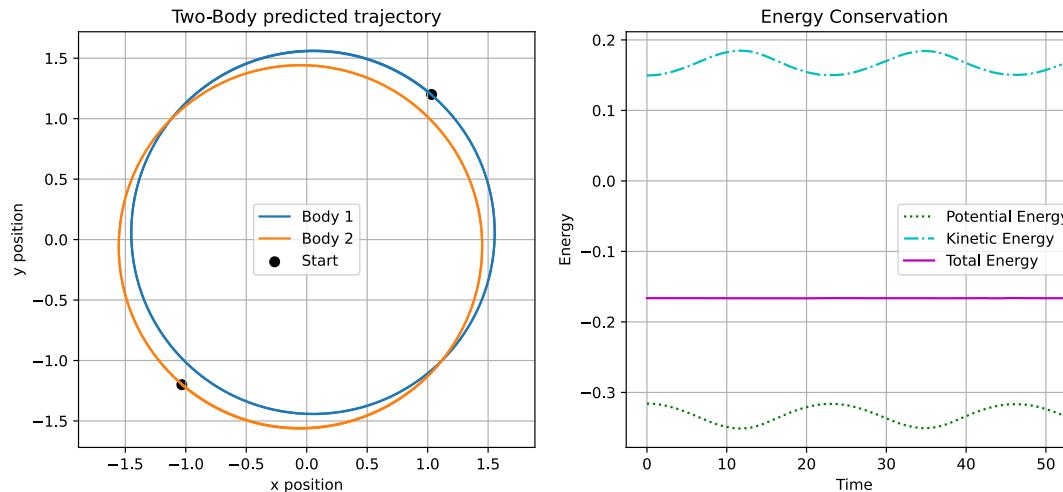


Figure 7 Predicted two-body trajectories (*left*) generated by S-HNN (A-SWIM) trained with train size 15000 and width 1000 and integrated via implicit midpoint with $\Delta t = 0.001$ for 50000 time steps. Kinetic, potential energy and total energy are plotted over time (*right*).

RF-HNNs for N-Body Systems — Two-body experiments

S-HNN Model: Trajectory simulation

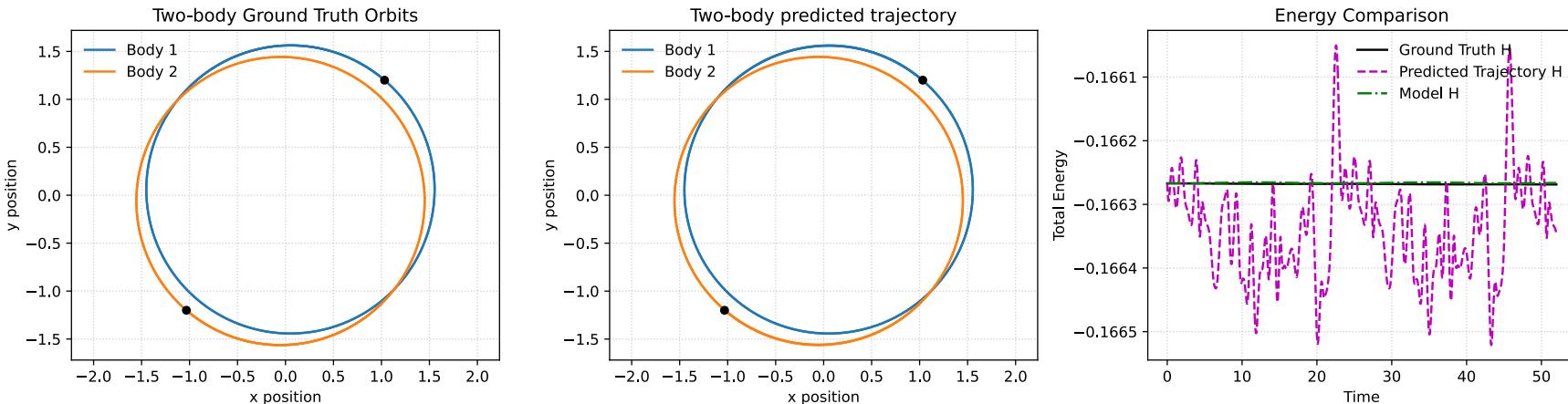


Figure 8 Comparison of ground-truth (*left*) and predicted (*middle*) two-body trajectories is shown. We also compare the ground truth Hamiltonian with the predicted trajectory's and model's learned Hamiltonian (*right*).

RF-HNNs for N-Body Systems — Two-body experiments

S-MLP Model: Trajectory simulation

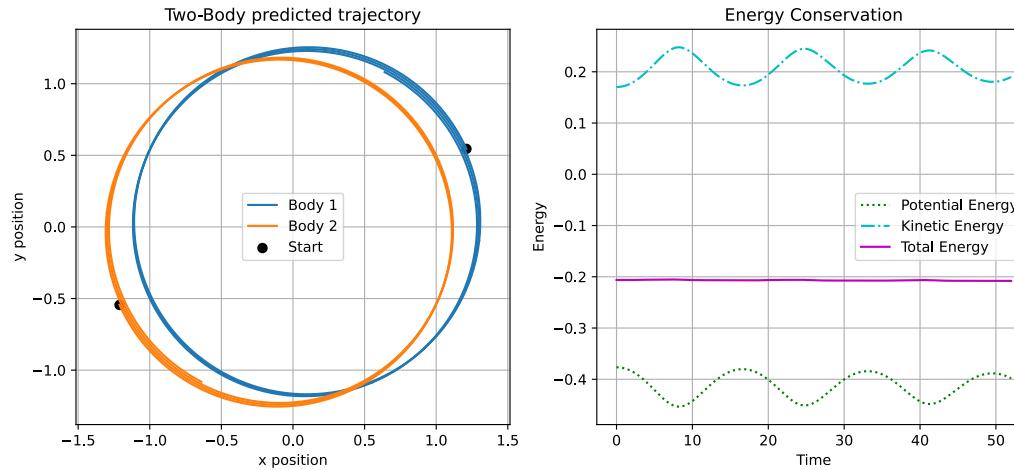


Figure 9 Predicted two-body trajectories (*left*) trained on S-MLP (A-SWIM) using train size 15000 and width 1000 are shown. Kinetic, potential and total energy over time is visualised on the *right*.

RF-HNNs for N-Body Systems — Two-body experiments

S-MLP Model: Trajectory simulation

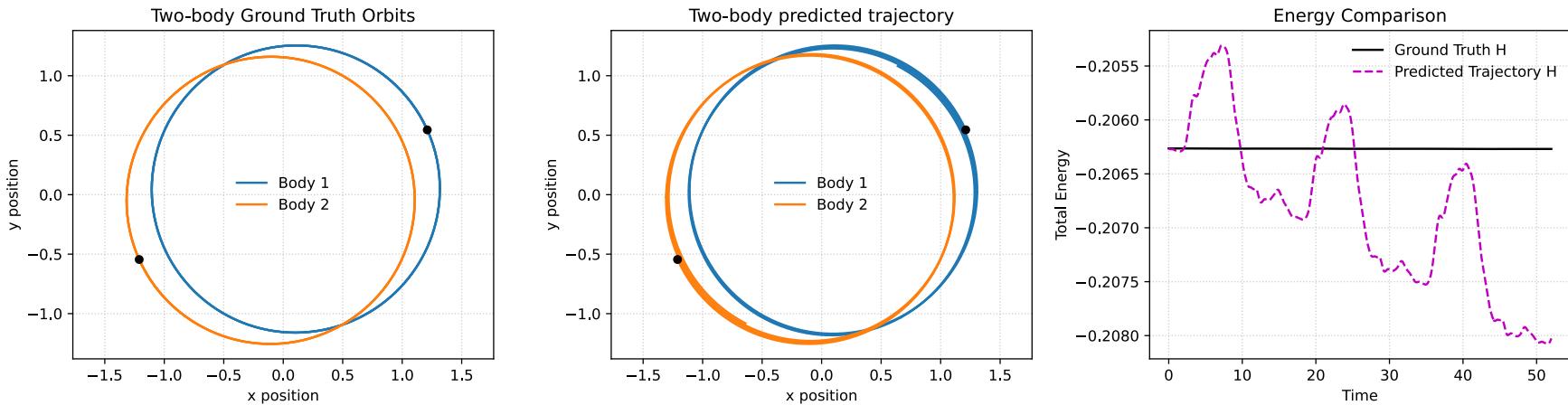


Figure 10 Ground-truth (*left*) and predicted (*middle*) two-body trajectories, with total energy comparison (*right*). The total energy of the predicted trajectory exhibits noticeable drift over time.

RF-HNNs for N-Body Systems — Two-body experiments

Sampled vs Traditionally Trained HNN: Trajectory simulation

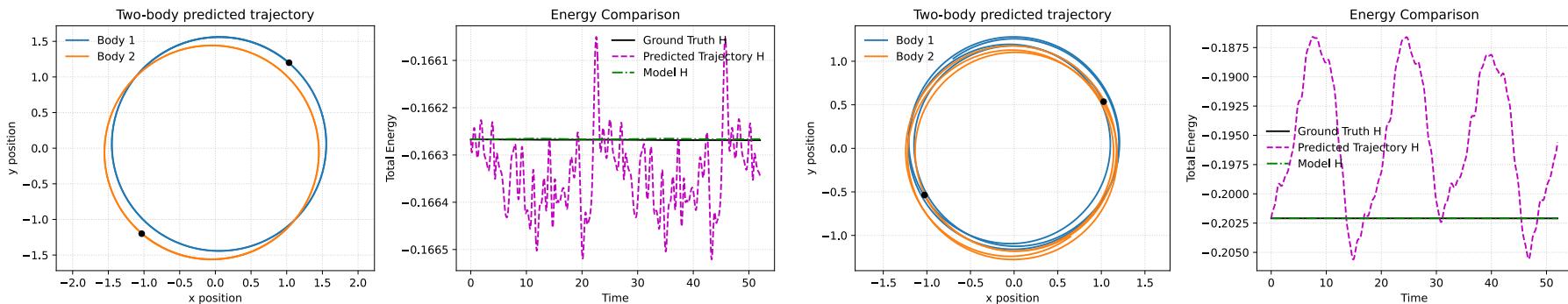


Figure 11 Comparing S-HNNs (*left*) with HNNs (*right*) on the two-body trajectories, shows the S-HNN preserves near-circular orbits and maintains more stable energy behaviour than the traditionally trained HNN.

RF-HNNs for N-Body Systems — 3-body experiments

Orbit data generation

Base initial conditions:

Use reference values of known periodic solutions:

- Chenciner–Montgomery Figure-8
- Broucke A1 orbit

Sampling initial conditions

- Integrate base initial conditions forward in time
- Sample a set of initial conditions uniformly spaced along the orbit as in fig
- Add small uniform noise to positions η

Trajectory generation:

- Integrate perturbed initial states forward in time

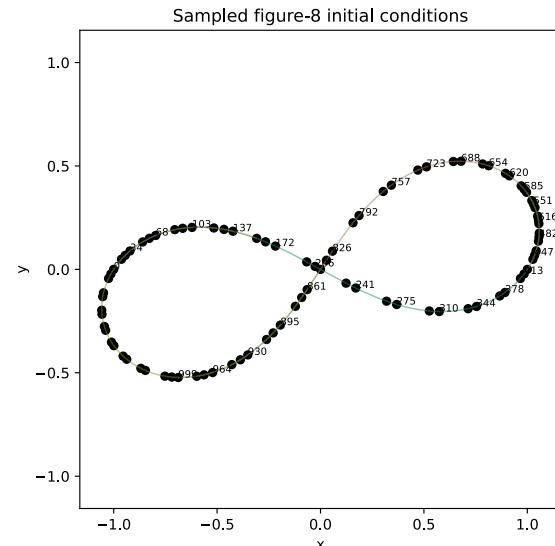


Figure 12 Sampled figure-8 initial conditions

RF-HNNs for N-Body Systems — 3-body experiments

Training results: Width vs Relative Test Error (Figure-8)

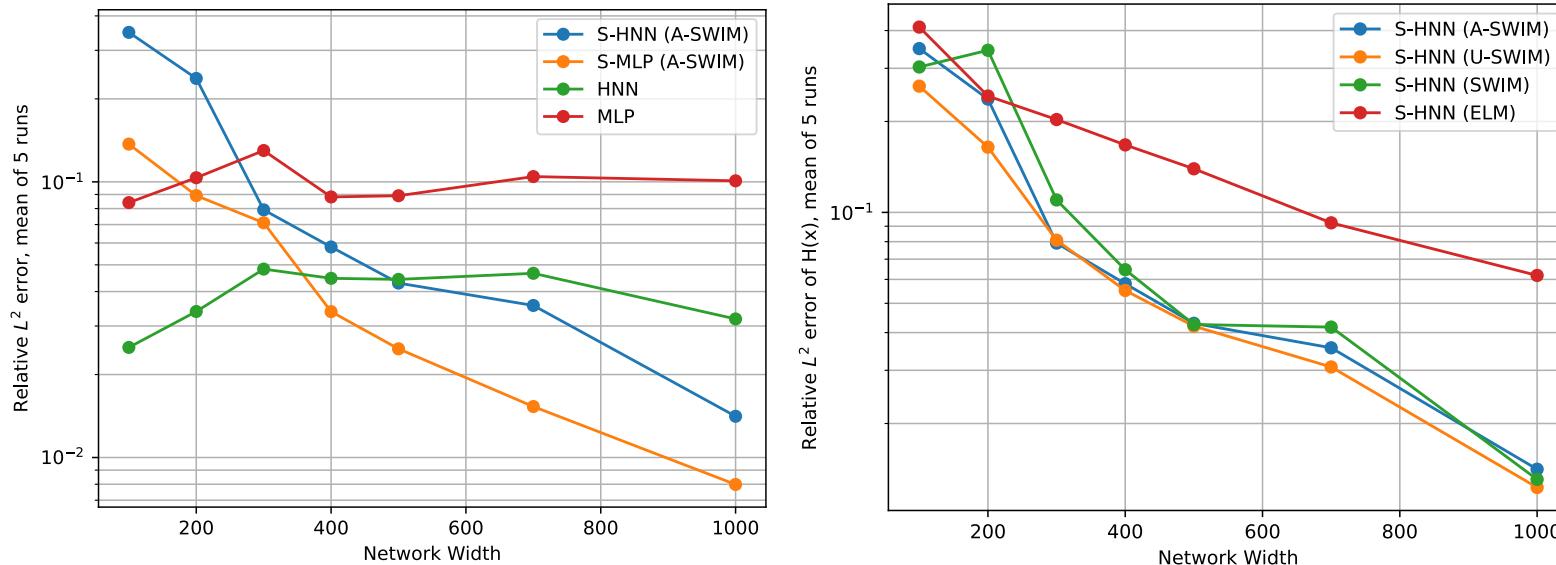


Figure 12 Relative test errors for the three-body figure-8 system are plotted against network width. *Left:* comparison between traditionally trained and sampled models. *Right:* comparison of different sampling strategies for S-HNN.

RF-HNNs for N-Body Systems — 3-body experiments

Training results: Train Size vs Relative Test Error

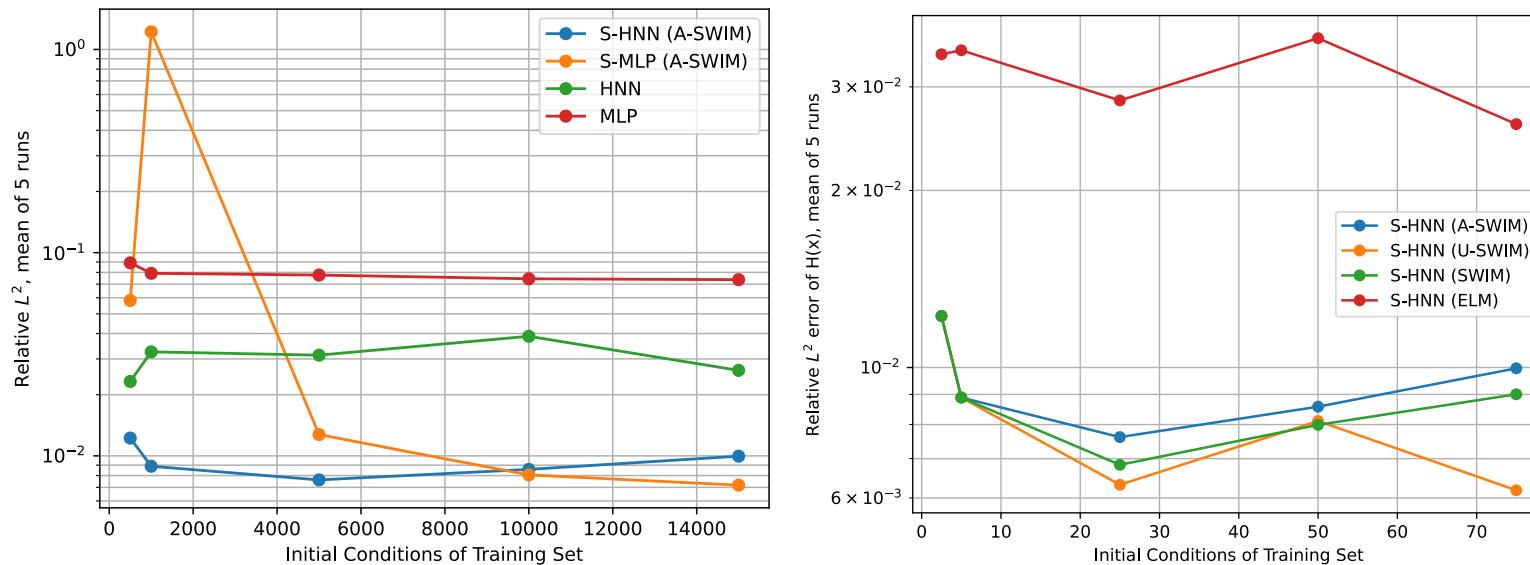


Figure 13 Relative test errors for the three-body figure-8 system are plotted against training dataset size. *Left:* comparison between traditionally trained and sampled models. *Right:* comparison of different sampling strategies for S-HNN.

RF-HNNs for N-Body Systems — 3-body experiments

S-HNN Model: Trajectory simulation via symplectic euler

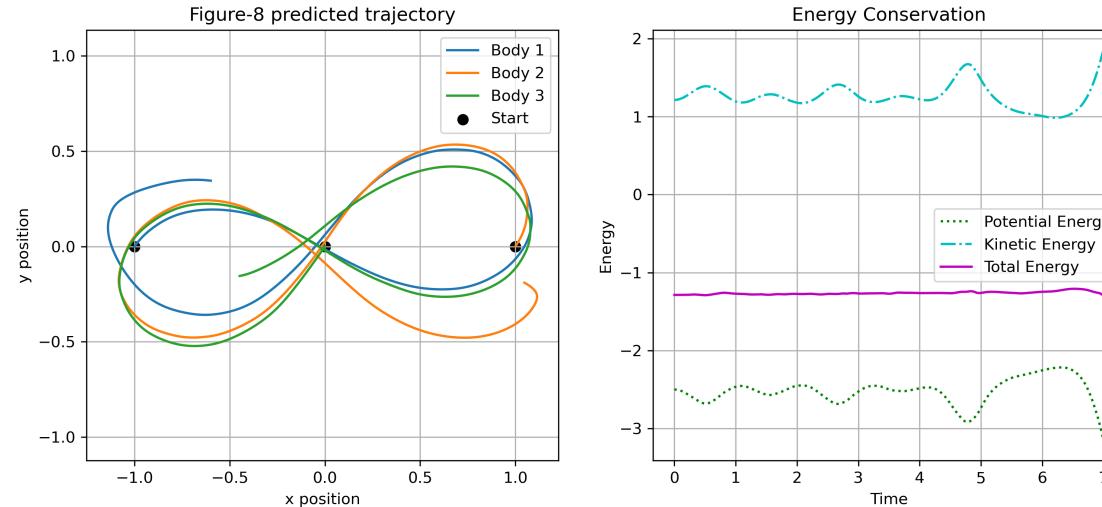


Figure 15: Predicted Figure-8 three-body trajectory generated by the S-HNN model using **symplectic Euler** integration (*left*) using $\Delta t = 0.001$ for 7000 steps, and corresponding energy evolution over time (*right*).

RF-HNNs for N-Body Systems — 3-body experiments

S-HNN Model: Trajectory simulation via implicit midpoint

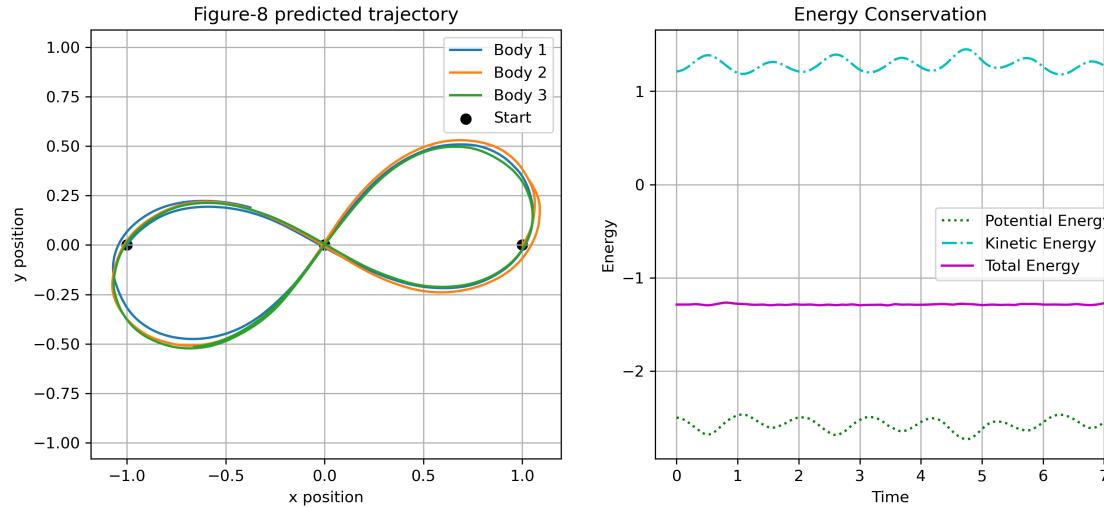


Figure 16: Predicted Figure-8 three-body trajectory generated by the S-HNN model using **implicit midpoint** integration (*left*) using $\Delta t = 0.001$ for 7000 steps, and corresponding energy evolution over time (*right*).

RF-HNNs for N-Body Systems — 3-body experiments

S-HNN Model: Trajectory simulation

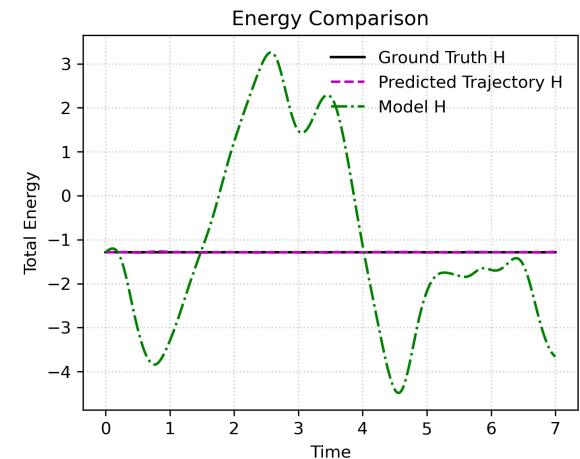
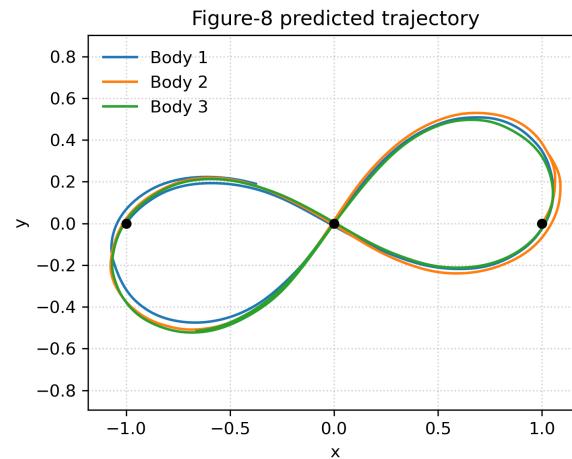
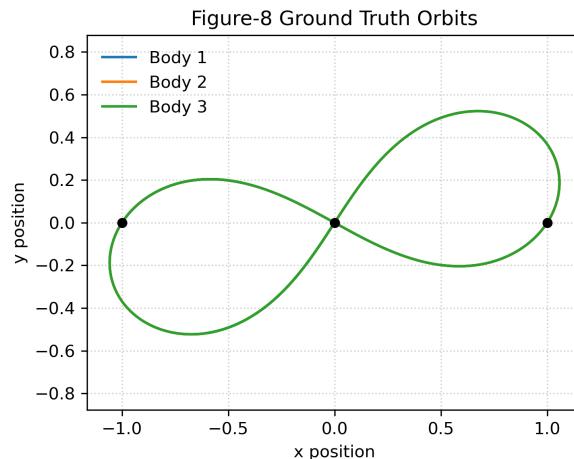


Figure 17: *Left:* ground-truth periodic orbit. *Middle:* predicted trajectory obtained by implicit midpoint integration. *Right:* energy comparison between the ground truth, predicted orbit's and S-HNN model's Hamiltonian over 7000 time steps.

RF-HNNs for N-Body Systems — 3-body experiments

S-MLP Model: Trajectory simulation

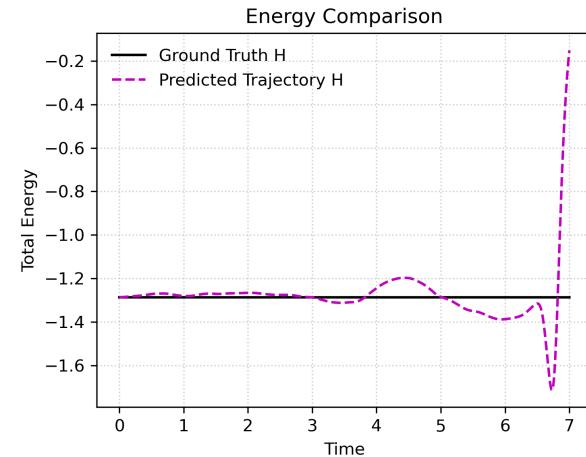
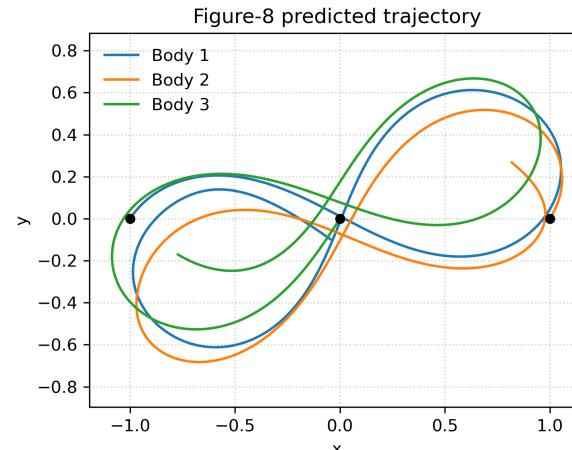
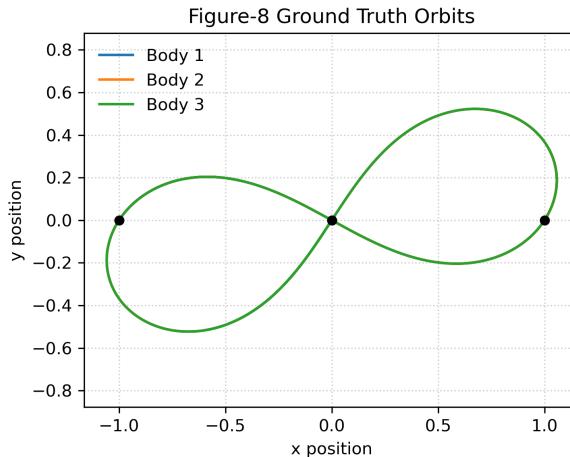


Figure 18: *Left:* ground-truth periodic orbit. *Middle:* predicted trajectory obtained by implicit midpoint integration. *Right:* energy comparison between the ground truth, predicted orbit's and S-MLP model's Hamiltonian over 7000 time steps.

RF-HNNs for N-Body Systems — 3-body experiments

HNN and MLP Models: Trajectory simulation

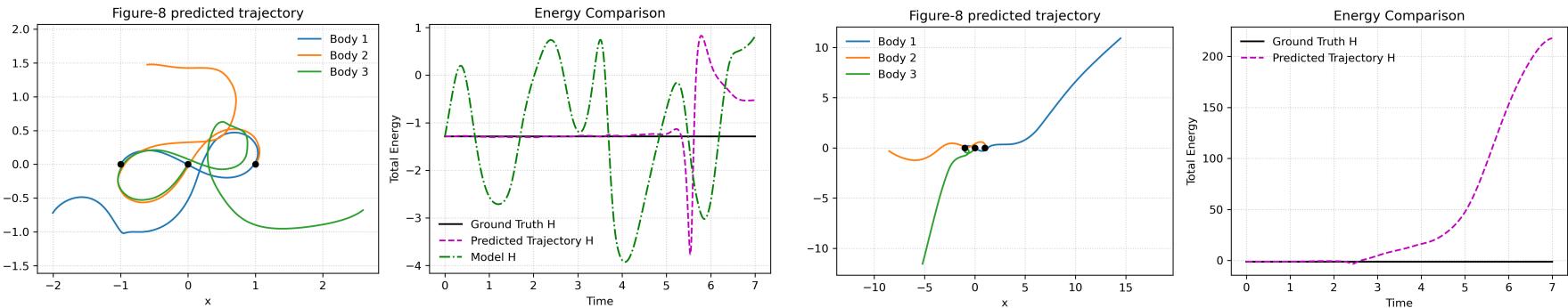


Figure 19: Predicted three-body trajectories and comparison of corresponding ground truth total energy evolution versus the predicted trajectory's energy evolution for HNN (left) and MLP models (right) on the figure-8 orbit are modelled.

RF-HNNs for N-Body Systems — 3-body experiments

Training results: Width vs Relative Test Error (Broucke A1)

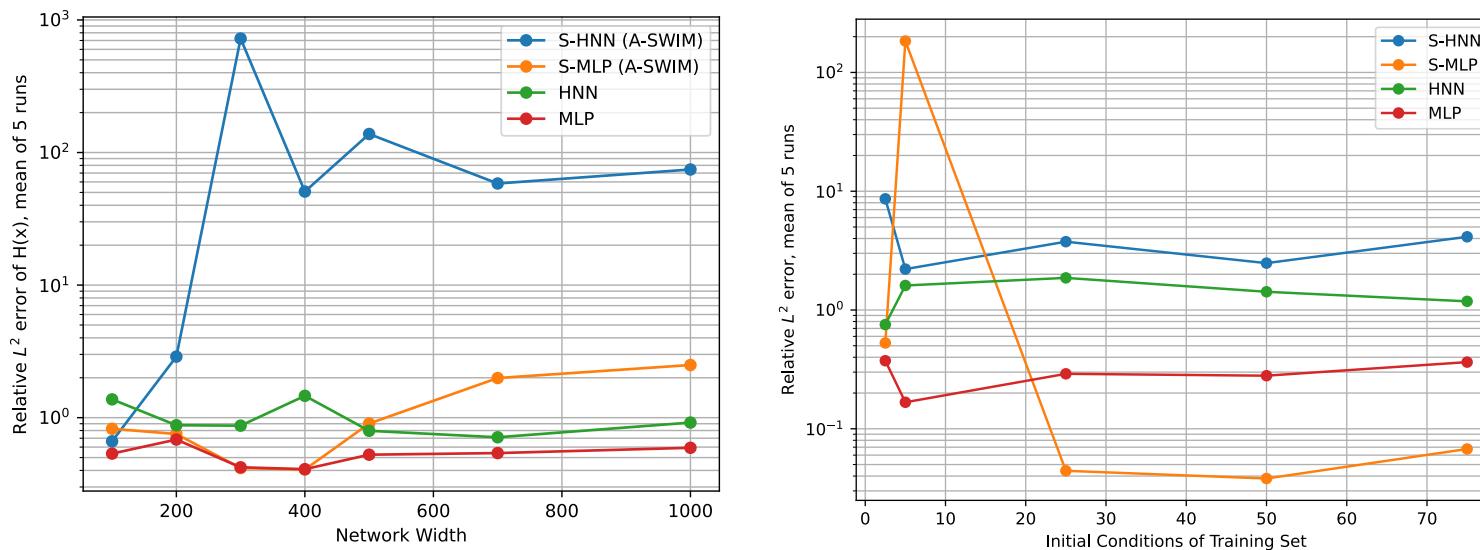


Figure 20 Relative test errors for the three-body Broucke A1 system plotted against network width (*left*) and training dataset size (*right*) for both sampled and traditionally trained models.

RF-HNNs for N-Body Systems — 3-body experiments

S-HNN Model: Trajectory simulation

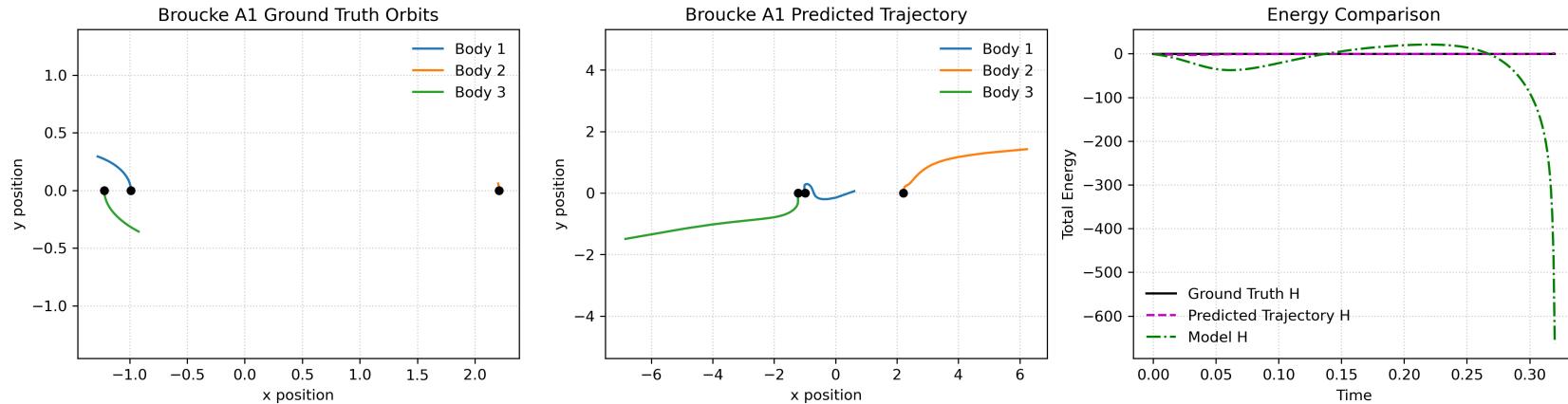


Figure 21 Ground-truth periodic orbit shown on the *left* compared to predicted trajectory of Broucke A1 in the *middle*. The right plot shows energy comparison between the ground truth, predicted orbit's and S-HNN model's Hamiltonian over 300 time steps.

RF-HNNs for N-Body Systems — 3-body experiments

HNN Model: Trajectory simulation

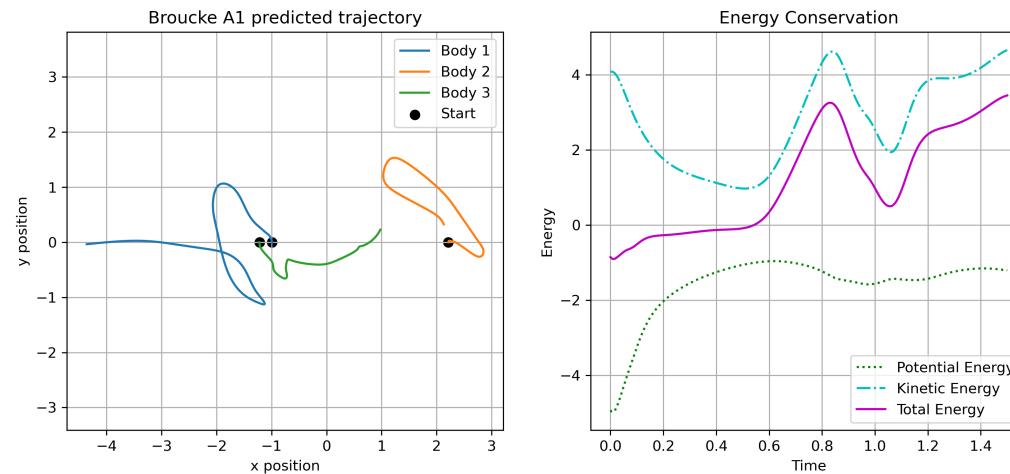


Figure 21 Predicted trajectory of Broucke A1 shown on the right compared to the kinetic, potential and total energy progression on the left for 1400 time steps.

Outline

- 1** Introduction
- 2** State of the art
- 3** Random Feature Hamiltonian Networks for N-Body Systems
- 4** Conclusion and future work

Conclusion

- RF-HNNs successful for simulating two-body system
- RF-HNNs reproduced the qualitative figure-8 dynamics but failed to capture the Broucke-A1 orbit.

Future Work

- RF-HNNs that learn the Hamiltonian separably.
 - This would allow modelling the model's kinetic and potential energies
 - Other integration schemes could be used such as Störmer verlet.
- Compare RF-HNN to other models: RF-HGNs, SRNNs

Literature

- [1] Newton, I. (1989). *The preliminary manuscripts for Isaac Newton's 1687 Principia, 1684–1685* (D. T. Whiteside, Ed.). Cambridge University Press.
- [2] K. R. Meyer and D. C. Offin. Introduction to Hamiltonian Dynamical Systems and the N-Body Problem. Applied Mathematical Sciences. Springer Cham, 2017.
- [3] S. Greydanus, M. Dzamba, and J. Yosinski. "Hamiltonian neural networks". In: Advances in neural information processing systems 32 (2019).
- [4] A. Rahma, C. Datar, and F. Dietrich. "Training Hamiltonian neural networks without backpropagation". In: arXiv preprint arXiv:2411.17511 (2024).