

Physics-Enhanced Neural Networks for Chaotic Hamiltonian Dynamics

Aryan Shrivastava*

School of Physical Sciences, National Institute of Science Education and Research

(Dated: November 20, 2025)

Chaotic Hamiltonian systems present a major challenge to data-driven forecasting due to their extreme sensitivity to initial conditions. We demonstrate an idea of a Hamiltonian Neural Network which learns the Hamiltonian, enforces the conservation laws, and produces reasonably accurate trajectories far beyond the training horizon. This significantly outperforms the standard feed-forward Neural Network in terms stability and long-term predictive power. The results highlight the importance of embedding the physical structure into machine learning models for chaotic dynamical systems.

I. INTRODUCTION

Chaotic Hamiltonian systems occupy a central place in modern nonlinear physics, appearing in celestial mechanics and molecular vibrations. Their extreme sensitivity to initial conditions makes long-time prediction challenging, even when the underlying laws are deterministic. The Hamiltonian formulation provides a description of such systems, with phase-space trajectories generated by energy-conserving and volume-preserving flows [1].

Machine-learning methods have recently shown promise in modeling complex dynamical systems, but conventional artificial neural networks (ANNs) remain **chaos-blind** [2]. Hamiltonian Neural Networks (HNNs) [3] overcome these limitations by learning a scalar Hamiltonian from data and generating dynamics via Hamilton's equations, embedding the structure by construction.

In this project, we implement ANN and HNN models for the Hénon–Heiles system, generate high-precision training data using RK45 method, and compare the models using long-time predictions and energy drifts. By reproducing a few results from the paper [2], we highlight the importance of incorporating physical structure into data-driven models of chaotic Hamiltonian systems.

II. THEORY

A. Hamiltonian Dynamics

The Hamiltonian formulation efficiently describes and reveals essential symmetries of a physical system. A system with N degrees of freedom is represented by generalized coordinates and conjugate momenta

$$(q_1, \dots, q_N, p_1, \dots, p_N) \in \mathbb{R}^{2N},$$

and its evolution is governed by a single scalar Hamiltonian function $\mathcal{H}(q, p)$.

Hamilton's equations are

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i}, \quad (1)$$

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i}. \quad (2)$$

If the Hamiltonian does not have an explicit time dependence, an immediate consequence is conservation of energy:

$$\frac{d\mathcal{H}}{dt} = \sum_i \left(\frac{\partial \mathcal{H}}{\partial q_i} \dot{q}_i + \frac{\partial \mathcal{H}}{\partial p_i} \dot{p}_i \right) = 0,$$

so trajectories evolve on constant-energy hypersurfaces.

B. The Hénon–Heiles Hamiltonian

The Hénon–Heiles system is a standard benchmark for studying the transition from regularity to chaos. It models stellar motion in a galactic potential and vibrational modes of molecules. The Hamiltonian is

$$\mathcal{H} = \frac{1}{2}(p_x^2 + p_y^2) + \frac{1}{2}(x^2 + y^2) + x^2y - \frac{1}{3}y^3. \quad (3)$$

The extra term in the potential,

$$V_p(x, y) = (x^2y - \frac{1}{3}y^3)$$

perturbs the circular harmonic oscillator by cubic nonlinearities, breaking rotational symmetry and introducing triangular symmetry (see Figure 1).

Hamilton's equations of motion become

$$\dot{x} = p_x, \quad (4)$$

$$\dot{y} = p_y, \quad (5)$$

$$\dot{p}_x = -x - 2xy, \quad (6)$$

$$\dot{p}_y = -y - x^2 + y^2. \quad (7)$$

Bounded motion exists only for energies $0 < E < \frac{1}{6}$, inside a triangular region in the (x, y) plane. At low energy the motion is with smooth closed curves in phase

* aryan.shrivastava@niser.ac.in
Roll no. - 2311041

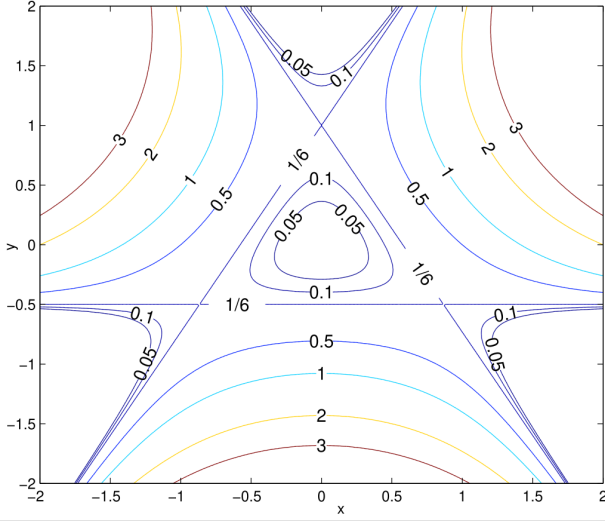


FIG. 1. Contour map for Henon-Heiles potential. (Source: Adapted from [4])

space. As $E \rightarrow 1/6$, the previously smooth and regular paths in phase space start to break apart, and the motion becomes partly chaotic. In this regime, some initial conditions lead to orderly motion while others lead to chaotic behavior, so both types of motion exist.

III. METHOD OF IMPLEMENTATION

A. Artificial Neural Networks (ANNs)

A conventional feed-forward artificial neural network approximates a nonlinear function parameterized by weights and biases θ (see Figure 2).

In this project, the ANN takes as input the state vector in the form

$$z = (q_x, q_y, \dot{q}_x, \dot{q}_y),$$

and is trained to predict the time derivatives

$$(\dot{q}, \ddot{q}) = f_\theta(z).$$

The loss function is the mean-squared error

$$\mathcal{L}_{\text{ANN}} = \|\dot{q}_t - \dot{q}\|^2 + \|\ddot{q}_t - \ddot{q}\|^2. \quad (8)$$

where \dot{q}_t and \ddot{q}_t are the true values in the dataset.

While ANNs are universal approximators, in chaotic systems, the amount of data and the number of neurons they would require to make accurate long-term predictions would be impractical.

The ANN's were observed to be very slow, so we had to resort to using Euler update for integration and generating the trajectory. Library RK4, standard RK45 methods were tried, but they were too time-consuming to produce any results. At the end we stick to the Euler update, as suggested in [2].

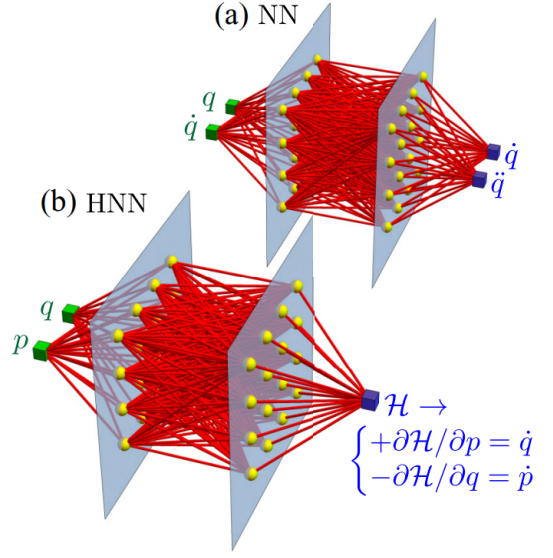


FIG. 2. Comparison of the model architectures: a conventional feed-forward ANN that learns time derivatives directly, and the Hamiltonian Neural Network (HNN) that learns a scalar Hamiltonian whose gradients generate the dynamics.

B. Hamiltonian Neural Networks (HNNs)

Hamiltonian Neural Networks incorporate physical structure directly into the learning process. Instead of predicting the time derivatives directly, the HNN learns the Hamiltonian:

$$\mathcal{H}_\theta = \mathcal{H}_\theta(q, p),$$

a scalar function of the canonical coordinates (see Figure 2).

To obtain the time derivatives, we use TensorFlow's automatic differentiation. Using `tf.GradientTape`, the network output $\mathcal{H}(q, p)$ is differentiated with respect to its input variables, which directly provides the partial derivatives,

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

allowing the Hamiltonian Neural Network to generate the time evolution of the system.

The training loss is

$$L_{\text{HNN}} = \left[\left(\dot{q}_t - \frac{\partial \mathcal{H}_\theta}{\partial p} \right)^2 + \left(\dot{p}_t + \frac{\partial \mathcal{H}_\theta}{\partial q} \right)^2 \right].$$

where \dot{q}_t and \dot{p}_t are the true values in the dataset.

This network was found to be much faster than ANN in terms of prediction, so we can use a more stable method of integration, which is the Library RK4 function (code provided in the Github repository [5]).

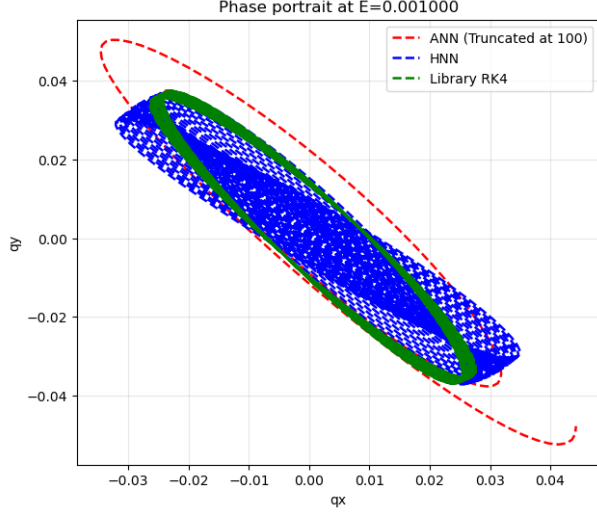


FIG. 3. Phase-space trajectories (q_x, q_y) at energy E_1 obtained using three methods: Artificial Neural Network (ANN, red), Hamiltonian Neural Network (HNN, blue), and the Library RK4 (green). The ANN prediction is truncated after 100 time steps, beyond which it diverges, while the HNN remains consistent with the true orbit.

IV. OBSERVATIONS AND ANALYSIS

We trained both the Artificial Neural Network (ANN) and the Hamiltonian Neural Network (HNN) on numerically integrated trajectories for energies $0 < E < 1/6$, using 20 energies and 20 orbits per energy. The dataset was generated by numerically integrating the Henon-Heiles equations of motion using `Scipy.integrate.solve_ivp`, which uses RK45 method.

The models were trained on very short trajectories (orbit time = 100) due to the large computation time, but predictions were evaluated over much longer durations (up to orbit time = 1800).

TABLE I. Python neural network parameters. We tested all values but generated our results with the **bold** ones.

| Description | Values |
|---|---------------------------|
| Number of layers | 2 , 4 |
| Neurons per layer | 100, 200 |
| Optimizer | Adam , SGD |
| Training loss threshold | 10^{-4} , 10^{-5} |
| Batch size | 256 |
| Epochs | 10, 20, 50 , |
| Activations | Tanh , ReLU |
| Orbit time (in dataset) | 100 |
| Orbit time (in evaluation) | 500, 1500, 1800 |
| Energy samples | 20 |
| Orbits per energy | 20 , 15 |
| Integration scheme (for dataset) | RK45 |
| Integration scheme (for ANN Trajectory) | Euler , RK4 |
| Integration scheme (for HNN Trajectory) | Library RK4 , RK45 |
| Data sampling rate | 0.01, 0.1 , 1, 10 |

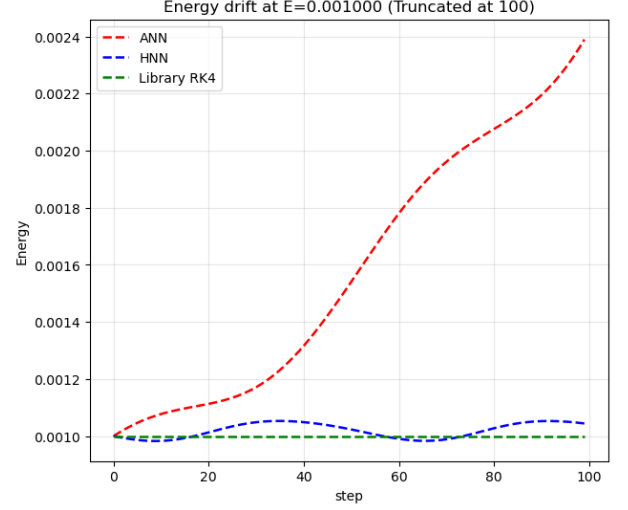


FIG. 4. Energy drift at E_1 for ANN and HNN. The ANN shows rapid deviation from the initial energy, while the HNN maintains near-constant energy.

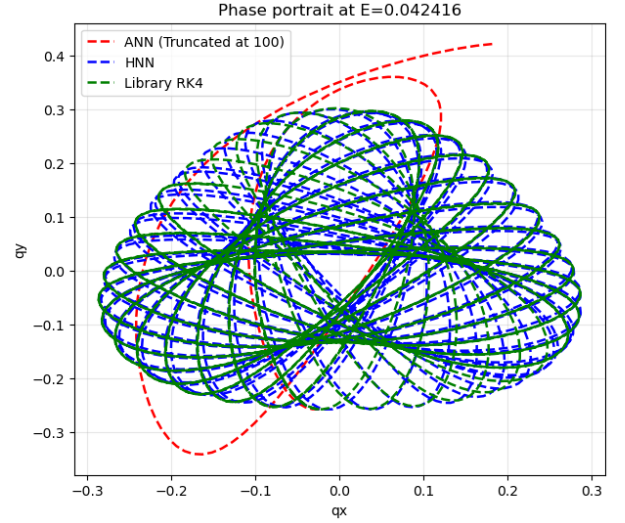


FIG. 5. Phase-space trajectories (q_x, q_y) at energy E_2 obtained similarly for the 3 methods. The ANN (red) prediction is truncated after 100 time steps, beyond which it diverges. The HNN (blue) maintains long-term stability and accurately reproduces the correct closed orbit.

A. Trajectory Comparison

For low energies and very small orbit times, both ANN and HNN reproduced dynamics reasonably well. However, clear divergences were observed as the prediction time increased (see Figure 3,5,7). The ANN trajectories deviated significantly from the true (RK45/RK4) orbits after approximately 60–100 integration steps, depending on the energy.

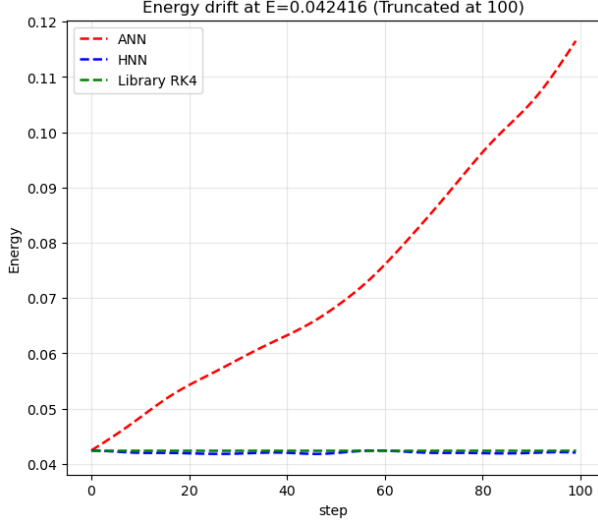


FIG. 6. Energy drift at E_2 for ANN and HNN. A similar trend is observed, that HNN obeys the conservation law.

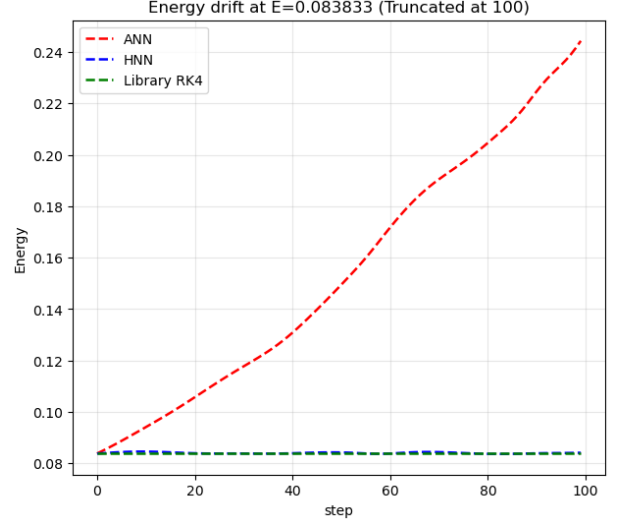


FIG. 8. Energy drift at E_3 . The ANN becomes unstable and accumulates large energy errors, while the HNN keeps the energy nearly conserved even in the chaotic regime.

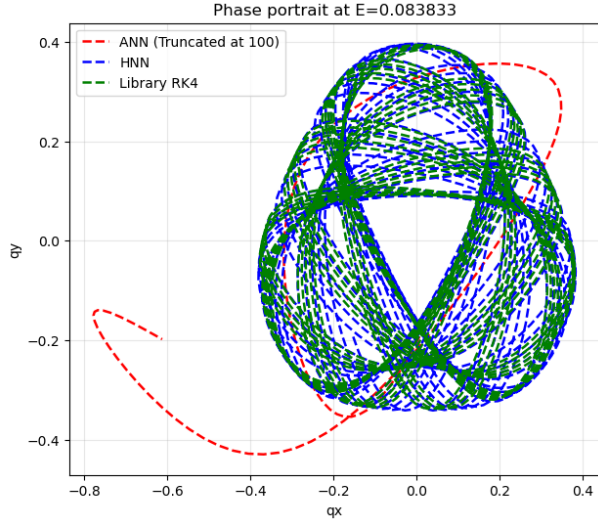


FIG. 7. Phase-space trajectories (q_x, q_y) at higher energy E_3 , where chaotic features become more prominent. The ANN prediction (red) is truncated after 100 time steps again, whereas the HNN (blue) continues to produce bounded trajectories closely following the Library RK4 solution (green).

In contrast, the HNN trajectories remained bounded for all initial conditions tested, even at long evolution times (much longer than the training input provided). The qualitative structure of the orbits closely matched the true trajectories. The long-time stability indicates that the learned Hamiltonian preserves the constant energy hyper-surface sufficiently well.

B. Energy Drift

Energy conservation provided a clear distinction between the models (see Figure 4,6,8). The ANN predictions showed rapid energy drift:

$$\Delta E(t) = |E(t) - E(0)|,$$

which grew with time. The drift increased more rapidly for higher energies, where the phase space is predominantly chaotic. At the highest bounded energy $E = 1/6$, the ANN became unstable in around 60 steps.

The HNN predictions showed minimal drift. Since $\dot{\mathcal{H}} = 0$ is built into Hamilton's equations, the model naturally constrained trajectories to remain close to constant-energy contours.

C. Comparison with Library RK4

We used the Library RK4 as a benchmark to compare the model performances. Side-by-side comparisons showed that ANN trajectories deform and spiral outwards, while HNN trajectories overlap closely with the numerical reference even after 1800 time steps (see Figure 9, 10).

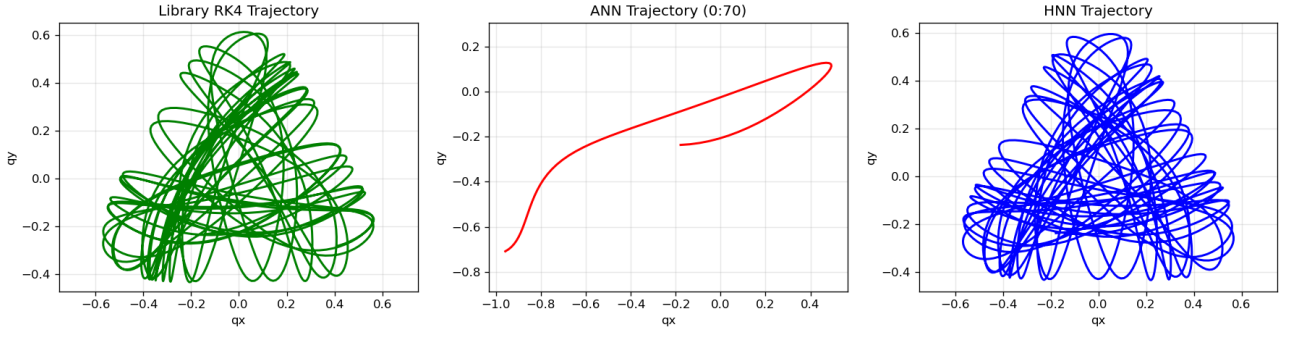


FIG. 9. Side-by-side comparison of trajectories at $E_4 = 0.125249$. The ANN rapidly deforms and leaves the bounded region, while the HNN remains stable and closely matches the RK4 reference.

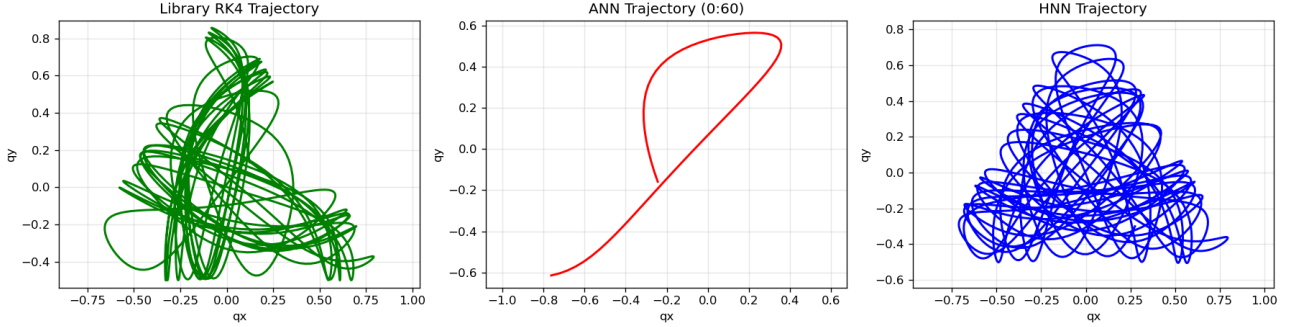


FIG. 10. Side-by-side comparison of trajectories at $E_5 = 1/6$ (highest energy for bounded orbits). The ANN diverges faster at higher energy.

V. CONCLUSION

In this project, we demonstrate advantages of embedding the physical structure in the form of the Hamiltonian into the machine-learning models, especially in the chaotic regime. The conventional ANN fails to conserve energy leading to divergence.

However, the HNN, learns the vector field and conservation laws **without invoking any details of its form**. It produced reasonably accurate trajectories even at longer orbit times. This work shows that Physics-Informed Neural Networks offer a powerful framework for accurate predictions of chaotic Hamiltonian systems.

VI. FUTURE WORK

Future extensions of this work can explore more quantitative measures, such as Lyapunov exponents, computed from HNN trajectories. Applying this framework to other systems, such as the double pendulum or dynamical billiards. Additionally, Lagrangian Neural Networks could be used to study non-conservative or damping-dominated systems, where energy is not preserved. Data-driven forecasting for chaotic systems is a promising direction for real-world applications.

[1] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Chapman and Hall/CRC, 2024.

[2] Anshul Choudhary, John F. Lindner, Elliott G. Holliday, Scott T. Miller, William L. Ditto, and Sudeshna Sinha. Physics-enhanced neural networks learn order and chaos. *Physical Review E*, 101(6):062207, 2020.

[3] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks, 2019.

[4] Debenben. Contour plot of the h  non  heiles potential. Wikimedia Commons, 2012. Creative Commons CC0 1.0 Universal Public Domain Dedication.

[5] crimsonpane23. Physics-enhanced neural networks for chaotic hamiltonian dynamics. GitHub repository, 2025. <https://github.com/crimsonpane23/Physics-Enhanced-Neural-Networks-for-Chaotic-Hamiltonian-Dynamics>