# Hamiltonian Neural Network Exploration for Electron Particle Tracking

Mikaela Meitz

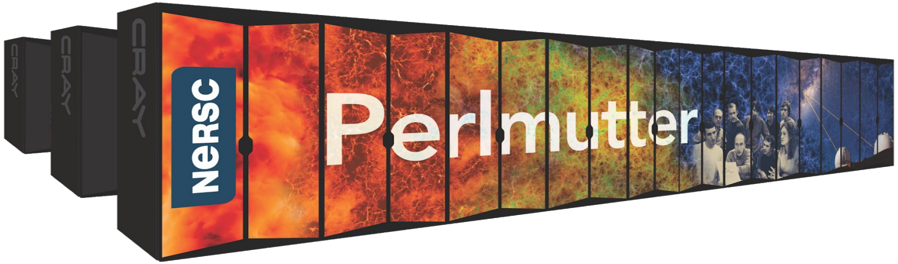
Thesis Project Defense
M.S. Applied Mathematics, CSULB
May 2023

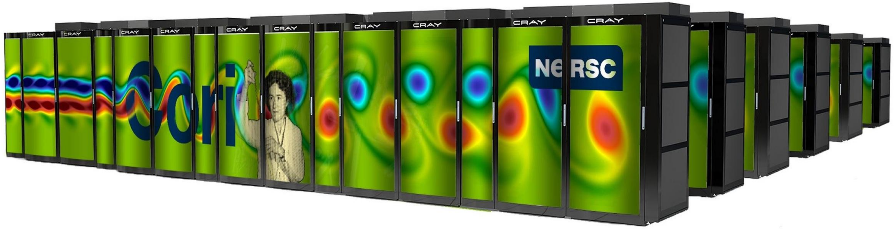
# National Energy Research Scientific Computing Center

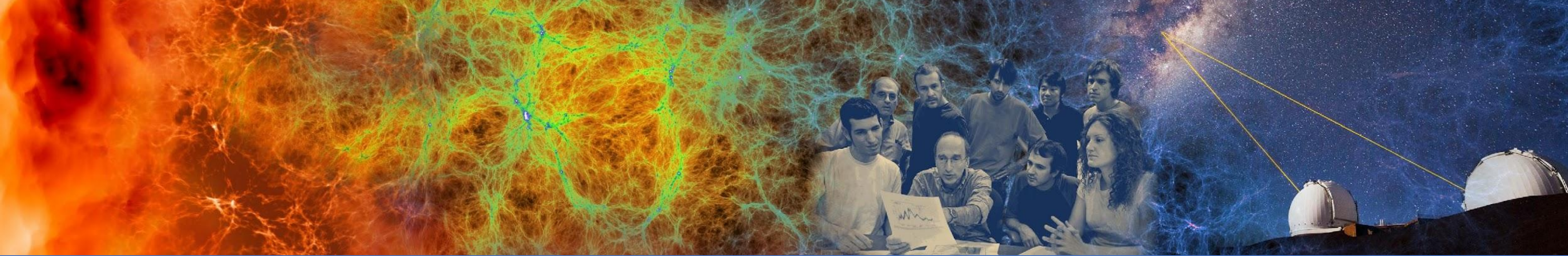
- NERSC is a national supercomputer center funded by the U.S. Department of Energy Office of Science (SC)
    - Part of Berkeley Lab

NERSC Supercomputer Systems as of Spring 2022:

Perlmutter and Cori



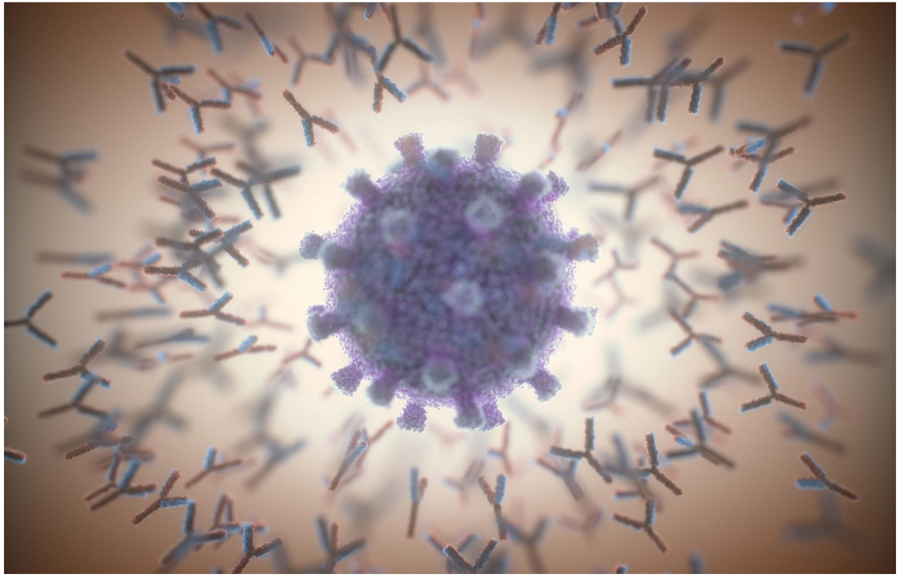





# Motivation

# Particle Accelerator

- machine that propels charged particles at high speeds and energies
- Create radiation for diffraction imaging used in chemistry, biology, material science, and history research

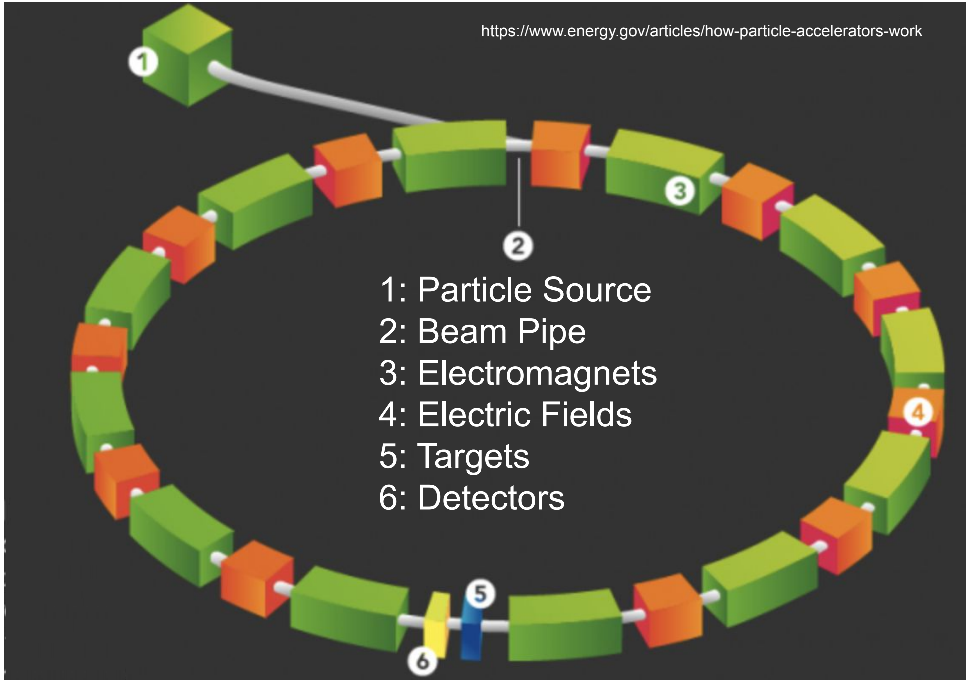


Advanced Light Source (ALS) particle accelerator at the Lawrence National Berkeley Laboratory

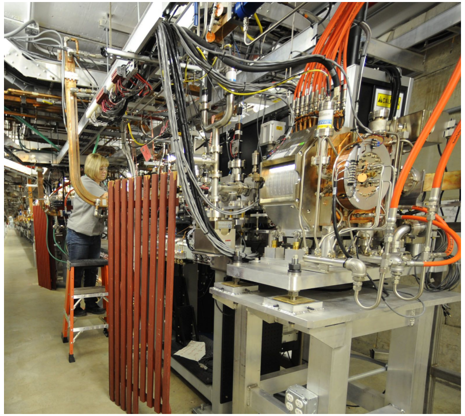
https://als.lbl.gov/new-insight-into-titans-hazy-atmospheric-chemistry/

# Particle Accelerator

- Facilities around the work require regular upgrades
- Need to provide new services to users




**Advanced Photon Source, Argonne National Lab** (Illinois)



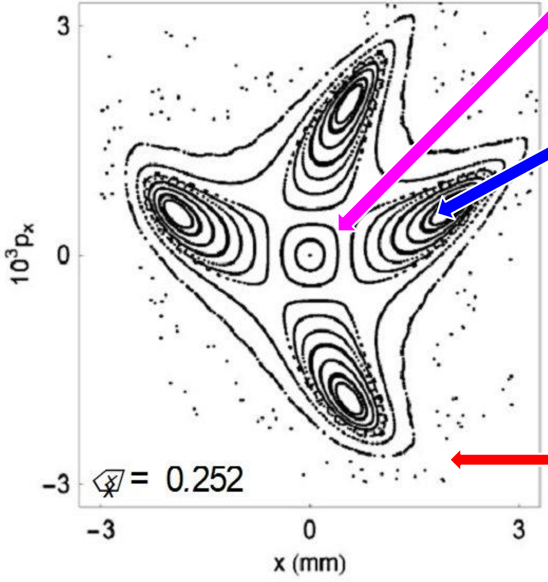
**LHC, CERN** (European Organization for Nuclear Research, Switzerland)

# Importance of Stability

Periodic accelerators are prone

to:

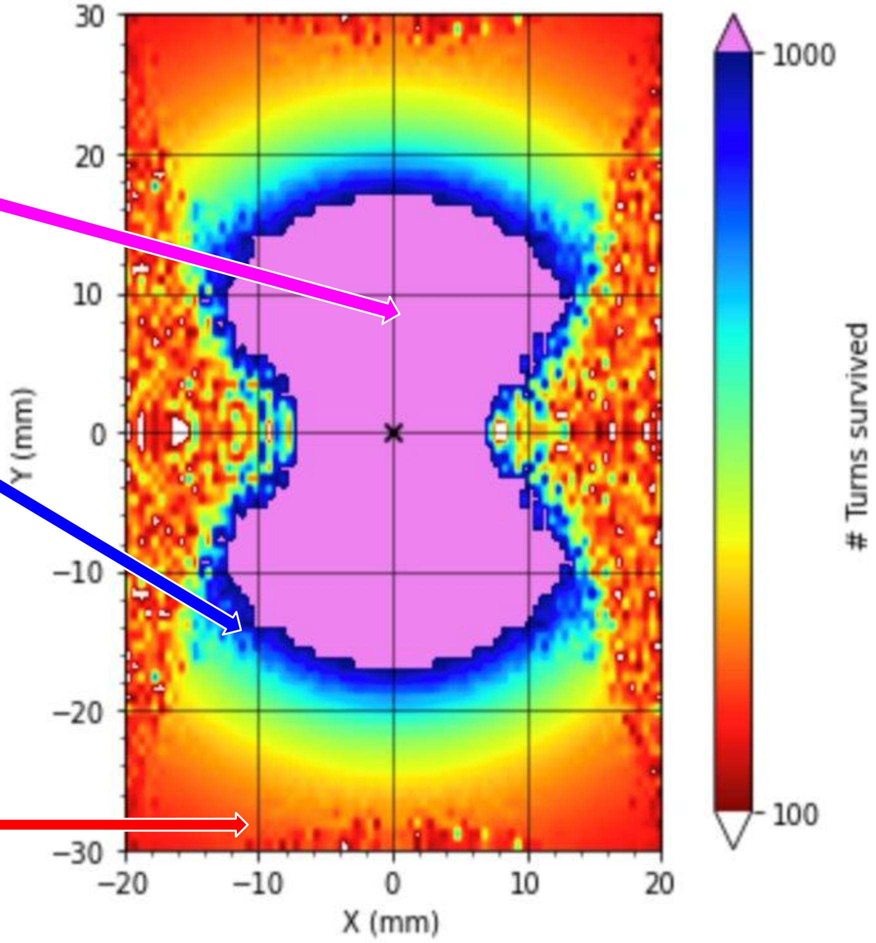
- Beam instability resulting in particle loss
- Less particles = less




Stable Region (infinite)

Limited stability

No stability, significant particle loss



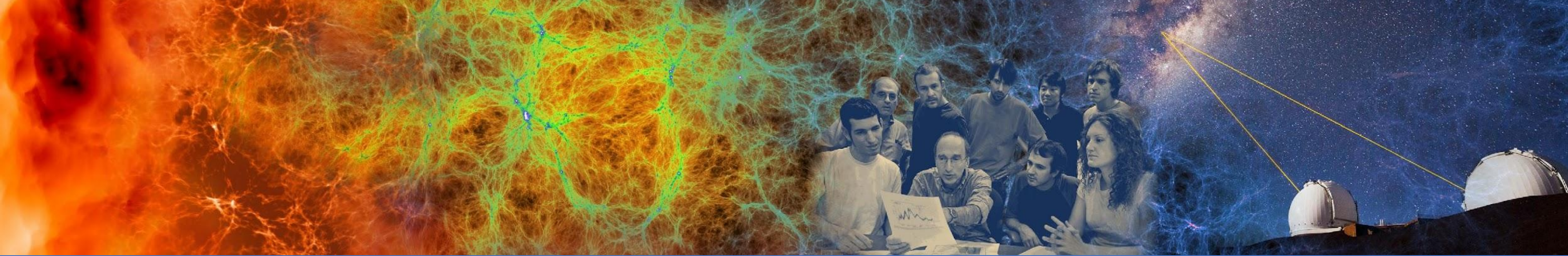
# Numerical Particle Tracking

- Numerical Particle tracking is computationally expensive but needs to be done
- Want to speed up this process without without sacrificing the quality of the results
- Physics relationships between components and the electrons is all encoded in the particle tracking

Can we utilize machine learning to replace part of the computationally expensive numerical particle tracking?

# A Hamiltonian Neural Network:

- Helps describe the motion/time evolution of a dynamical system through its energy
- Can help to solve iterative IVP by finding approximate solutions



# Theory

# Integrating a Hamiltonian

- Particle motion is governed by Hamiltonian (expression of total energy of system)

$$E_{total} = \mathcal{H}(q, p)$$

where Hamilton's Equations are,

$$\frac{dq}{dt} = \frac{\partial \mathcal{H}}{\partial p} \, , \, \frac{dp}{dt} = -\frac{\partial \mathcal{H}}{\partial q}$$

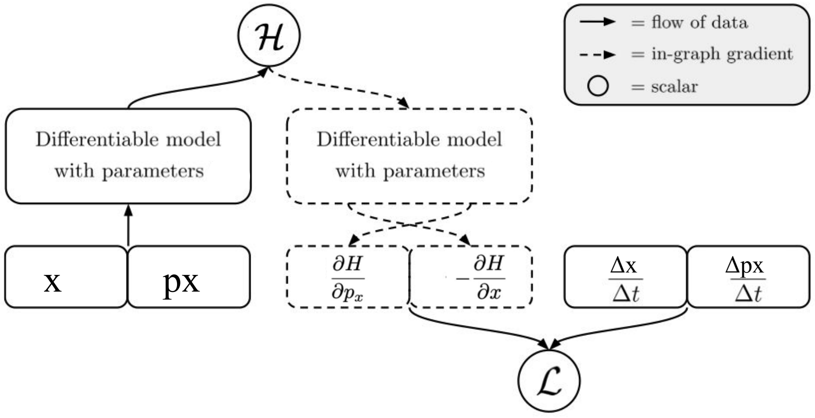
- To solve the IVP:

$$(q_1, p_1) = (q_0, p_0) + \int_{t_0}^{t1} \left(\frac{\partial \mathcal{H}}{\partial p}, -\frac{\partial \mathcal{H}}{\partial q}\right) dt$$

[2]

# General Idea

1. Receive a set of coordinates and send through differentiable model
2. Returns a single “energy-like” value
3. Sent through differentiable model again
4. Compute and return sum of gradients of outputs with respect to inputs
5. Split and swap data to create the gradient, $\mathbf{S}_{\mathcal{H}} = \left(\frac{\partial H}{\partial p_x}, \cdot -\frac{\partial H}{\partial x}\right)$
6. Map coordinates to their approximate time derivatives




**Hamiltonian NN**

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

[3]

# Method

# Exploration with a toy problem

- HNN learns the toy function below and maps coordinates to their approximate time derivatives

$$\mathcal{H}(x, p_x) = \frac{1}{2}(x^2 + p_x^2) + V(x)$$
$$= \frac{1}{2}(x^2 + p_x^2) - \frac{\alpha}{3}x^3$$

with $V(x)$ as the sextuple potential

- Hamiltonian Equations:

$$\frac{\partial \mathcal{H}}{\partial p_x} = \frac{\partial x}{\partial t} \; , \; -\frac{\partial \mathcal{H}}{\partial x} = \frac{\partial p_x}{\partial t}$$

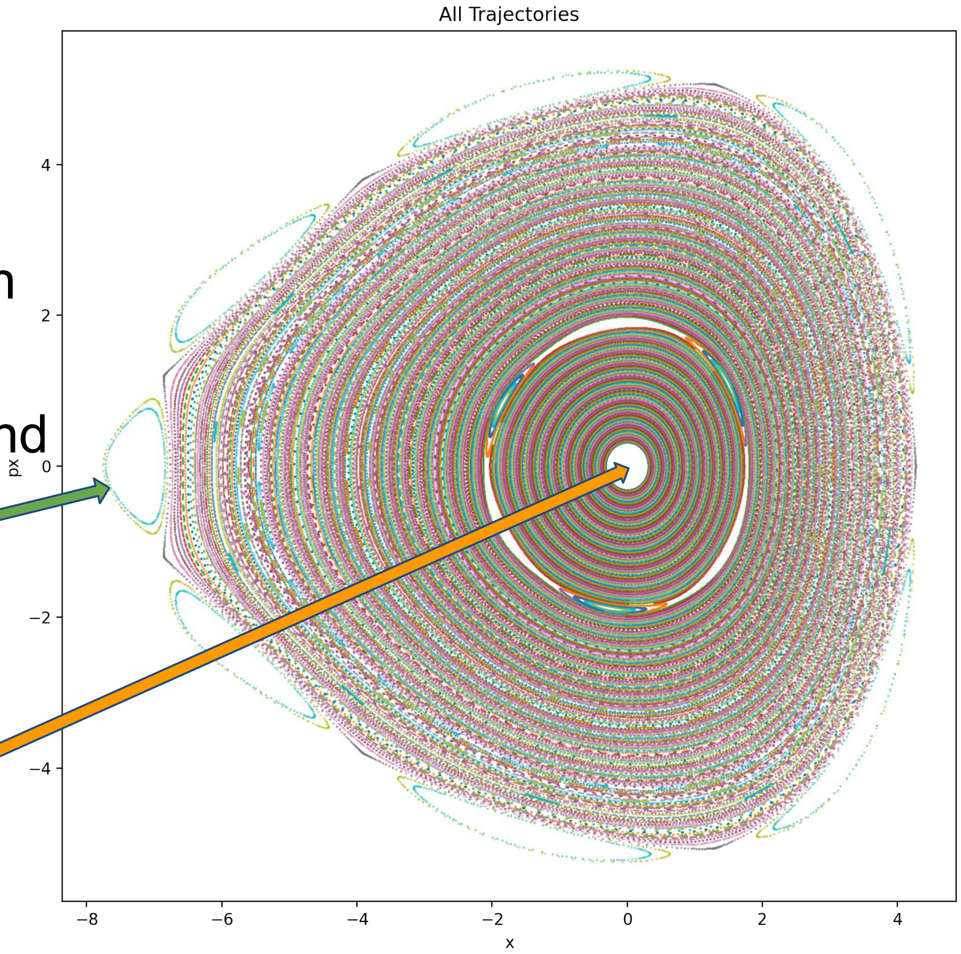
[1]

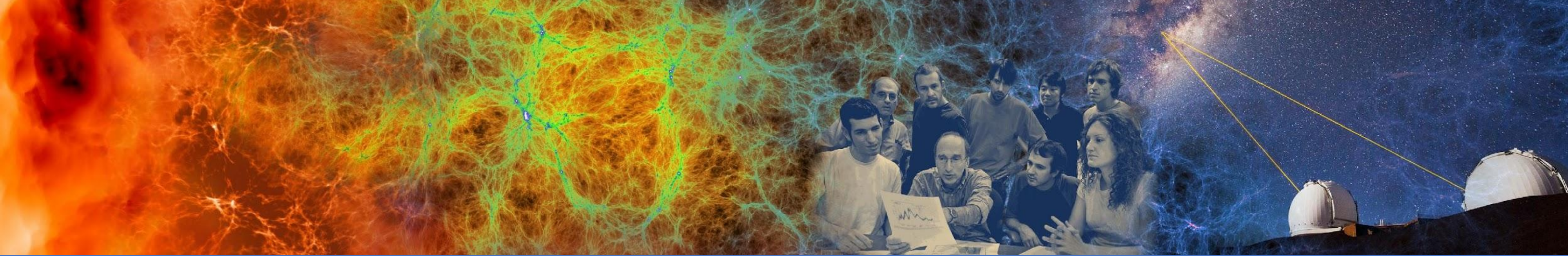
# Data Set

- Simulated particle trajectory data
- 300 particle trajectories, each with 1,000 coordinates
- 2 dimensions, $(x, p_x)$, position and momentum

Trajectory 300

Origin

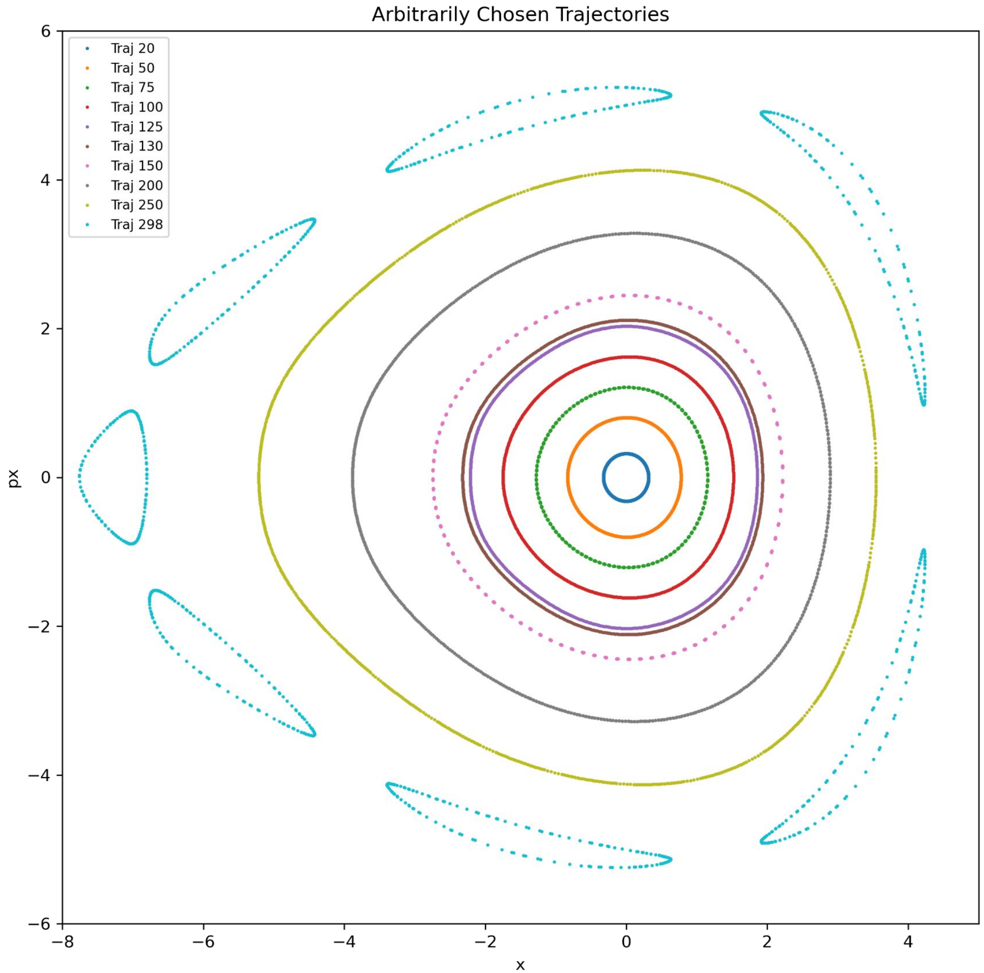


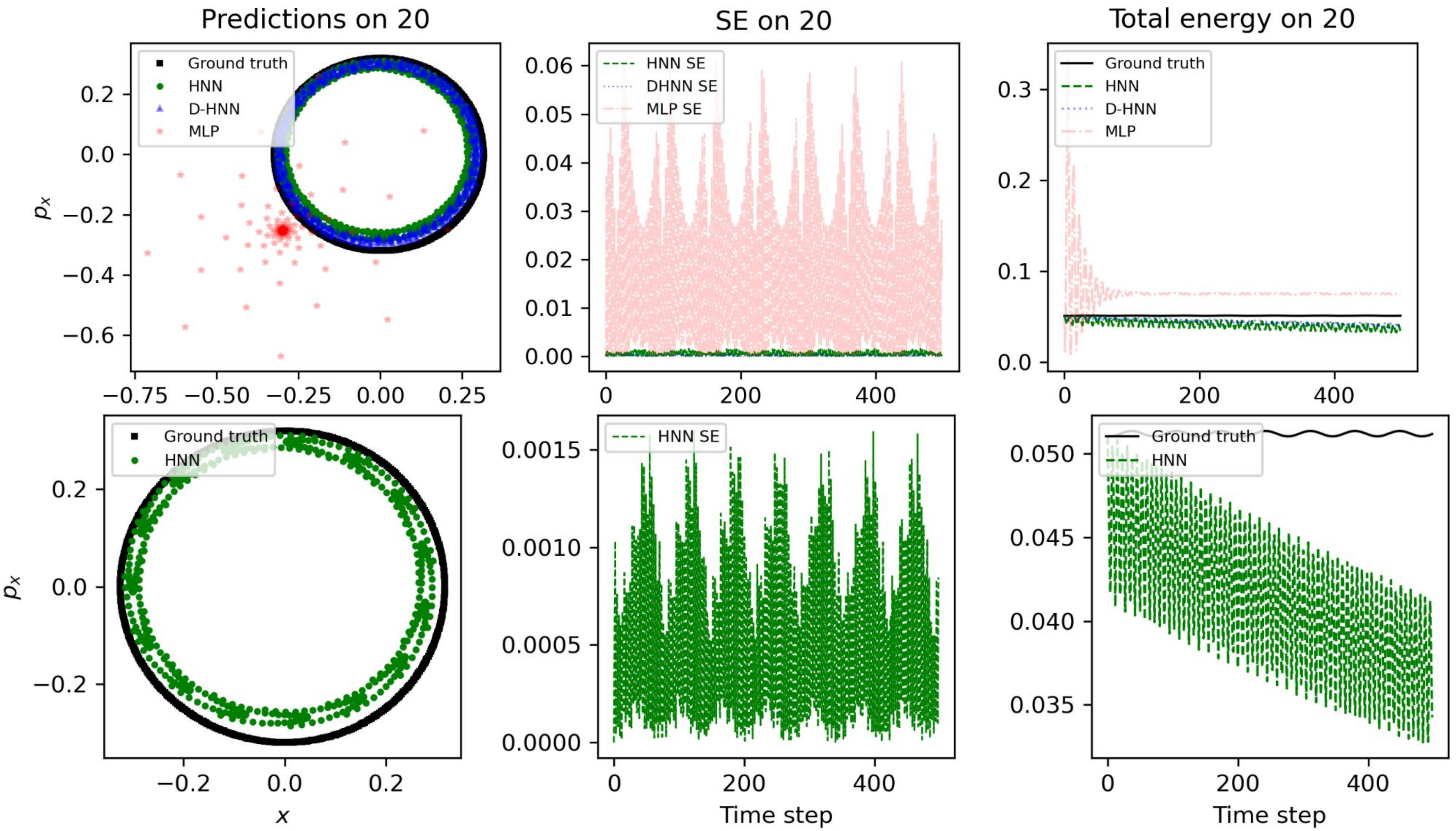


# Analysis and Results

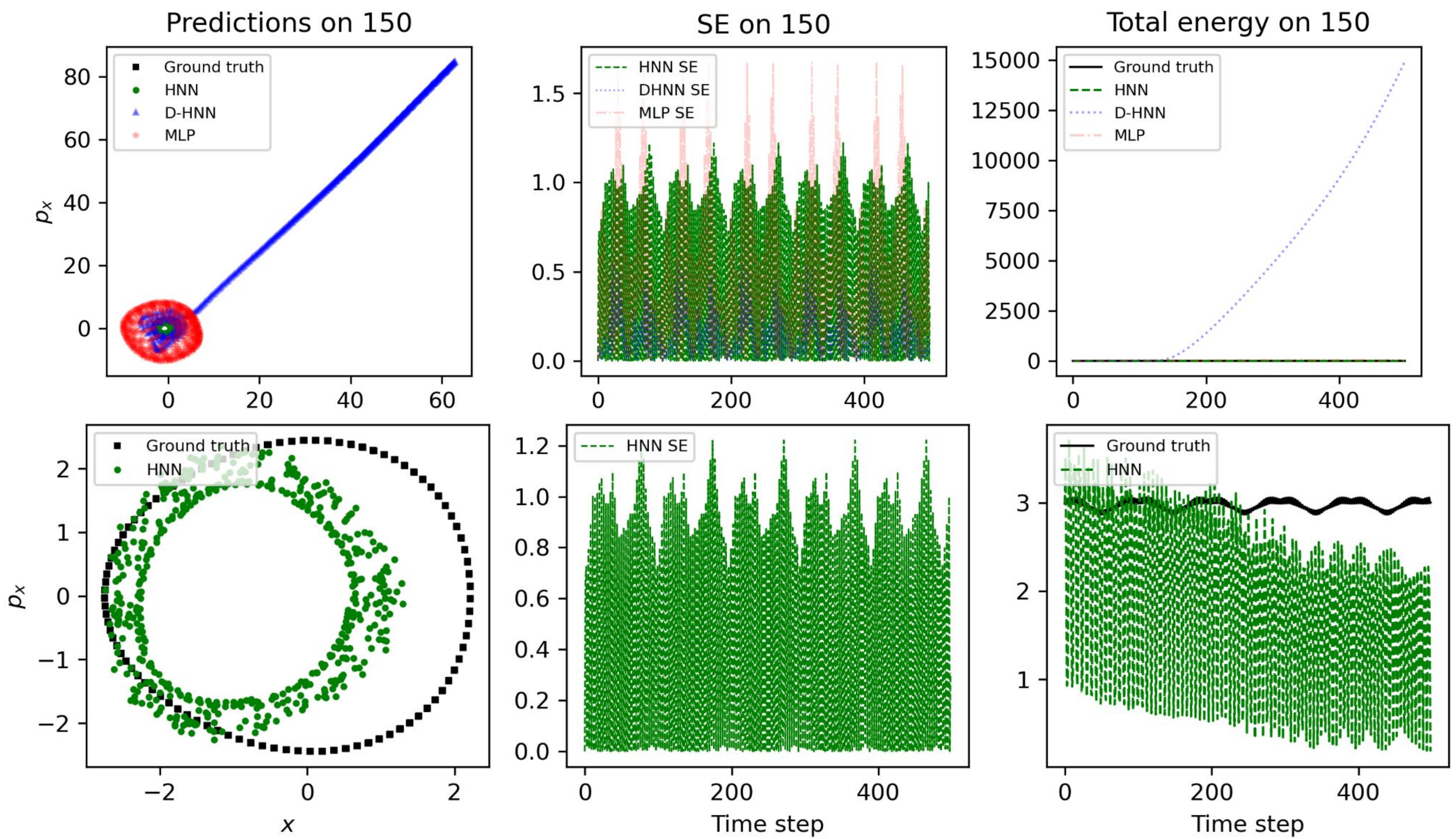
# Arbitrary Data Set

- Trajectories 20, 50, 75, 100, 125, 130, 150, 200, 250, and 298
- Various particle trajectory behaviors
- All trajectories are in the desired stability region
- Compare HNN against MLP and DHNN

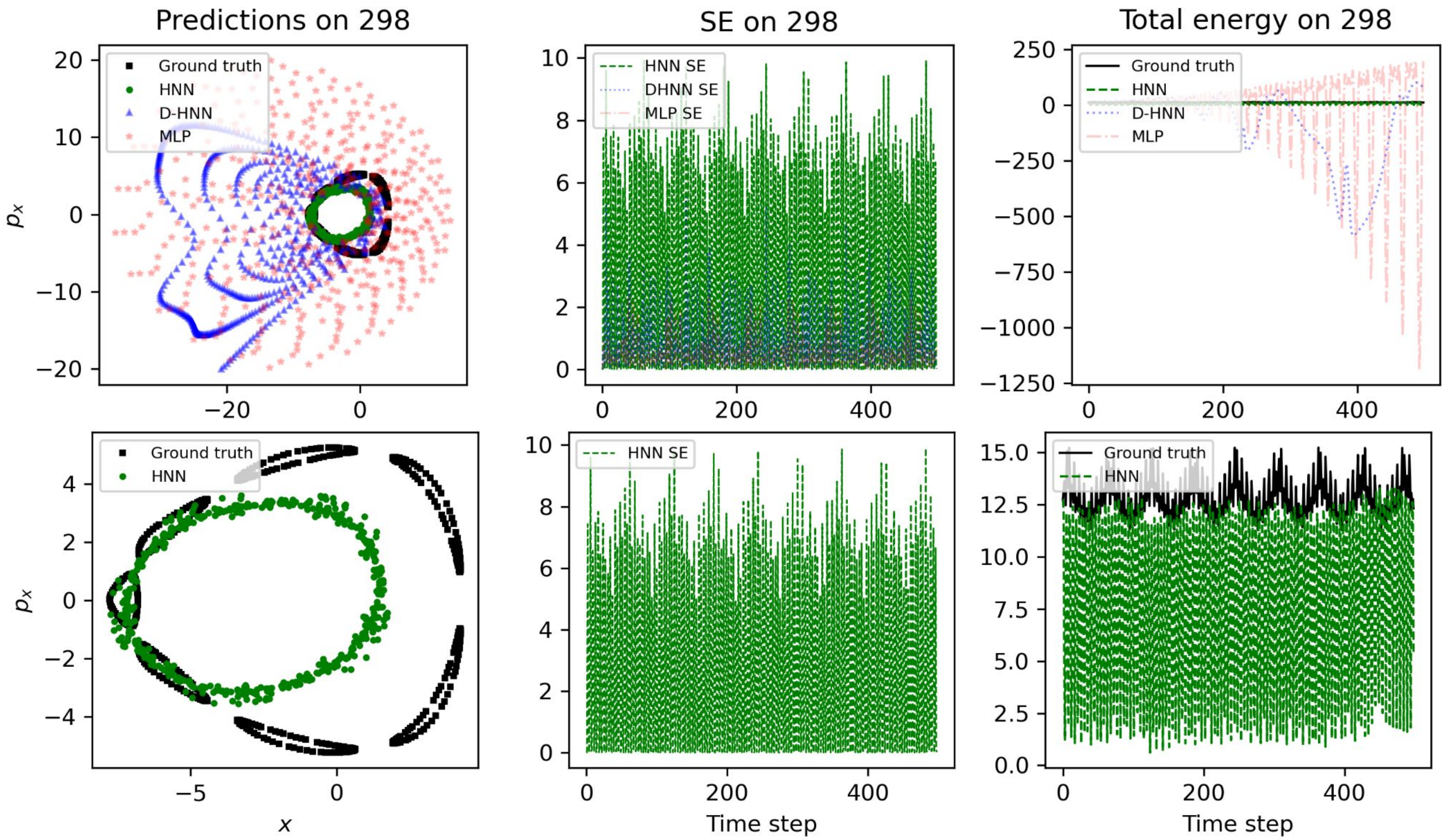




Predictions on 20
SE on 20
Total energy on 20
Ground truth
HNN
D-HNN
MLP
HNN SE
DHNN SE
MLP SE
$p_x$
$x$
Time step



Predictions on 150
SE on 150
Total energy on 150
Ground truth
HNN
D-HNN
MLP
HNN SE
DHNN SE
MLP SE
$p_x$
$x$
Time step
Time step


Predictions on 298
SE on 298
Total energy on 298
Ground truth
HNN
D-HNN
MLP
HNN SE
DHNN SE
MLP SE
$p_x$
$x$
Time step

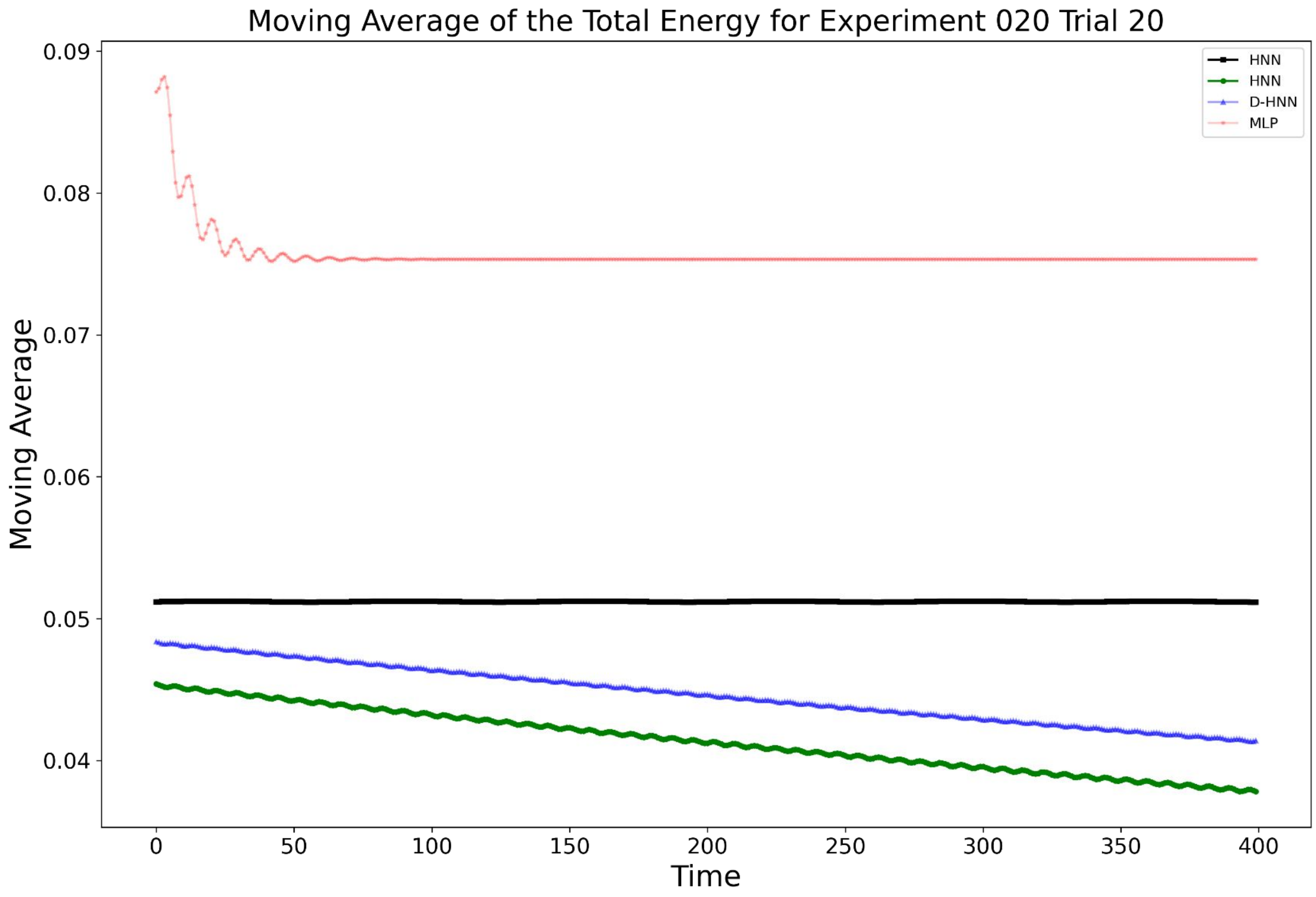
# Moving Average of the Total Energy

- Moving average of the total energy is given by,

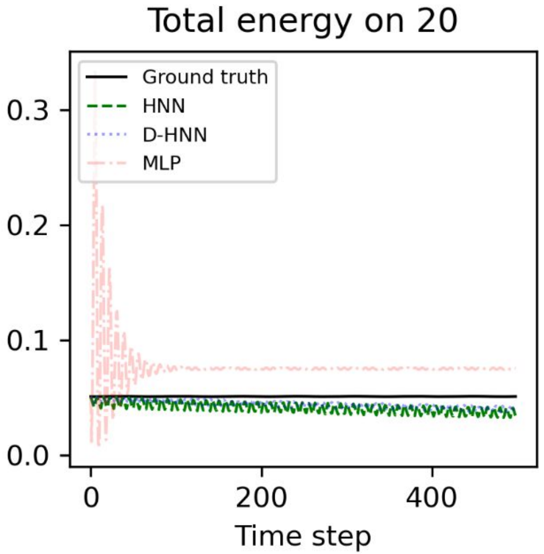
$$MA_w = \frac{p_{n-w+1} + p_{n-w+2} + \ldots + p_n}{w}$$
$$= \frac{1}{w} \sum_{i=n-w+1}^{n} p_i$$

  with window size, $w = 100$

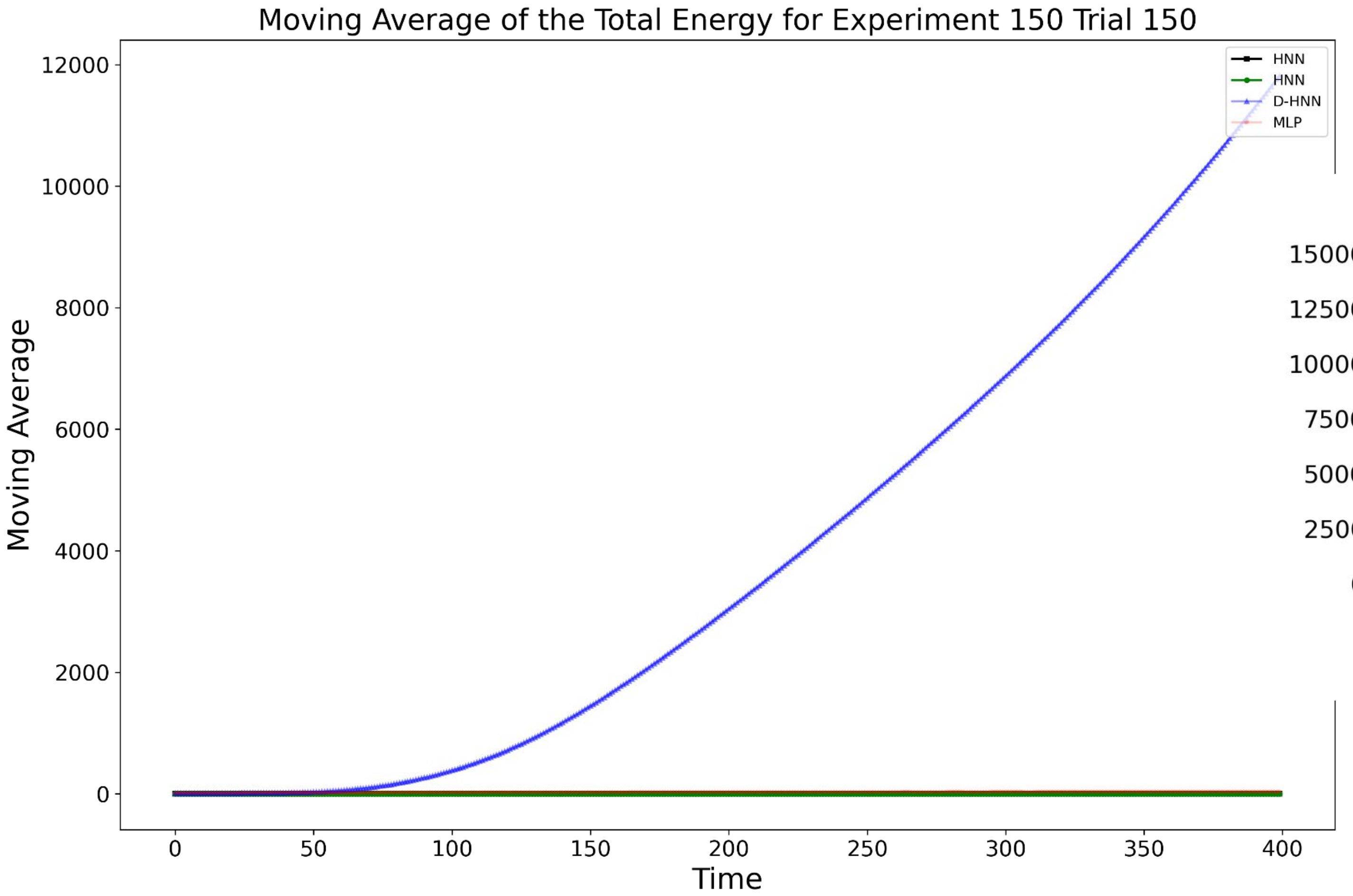
- Moving average reduces high frequencies in the dataset to provide smoother representations of the data

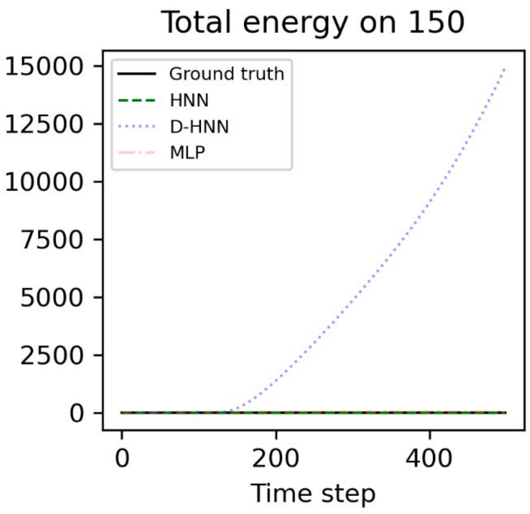


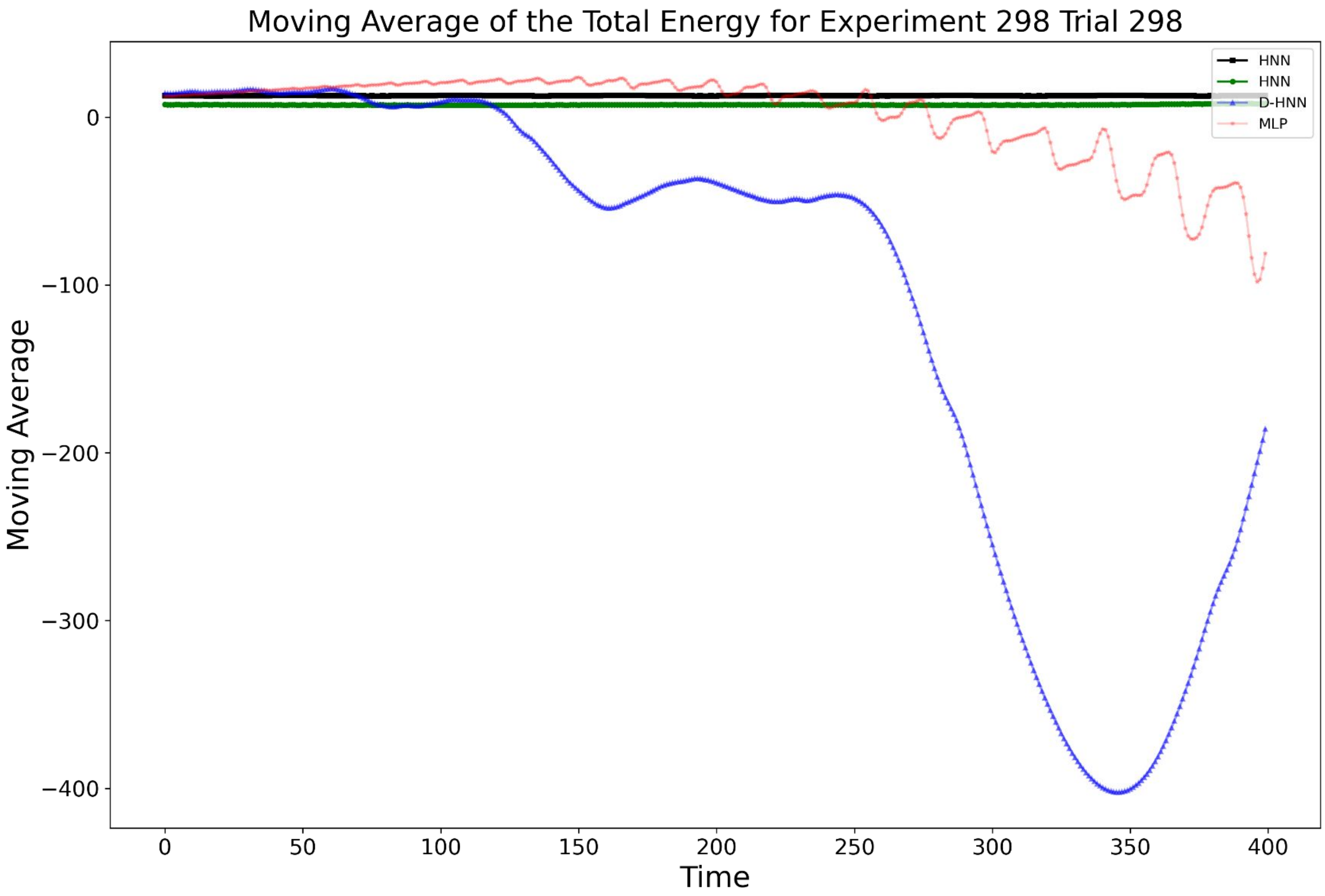
Moving Average of the Total Energy for Experiment 020 Trial 20
HNN
HNN
D-HNN
MLP
Moving Average
0.09
0.08
0.07
0.06
0.05
0.04
0
50
100
150
200
250
300
350
400
Time

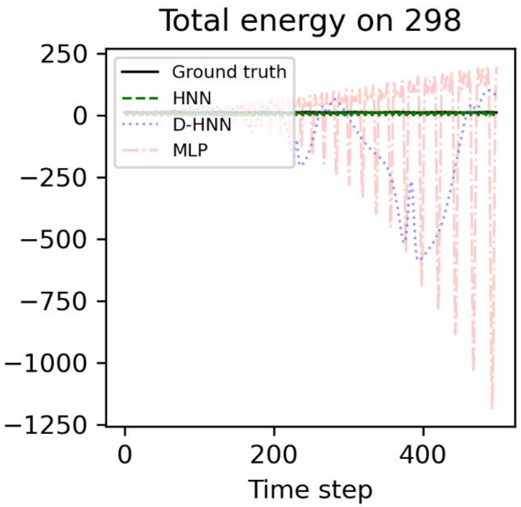



Total energy on 20
Ground truth
HNN
D-HNN
MLP
0.3
0.2
0.1
0.0
0
200
400
Time step


Moving Average of the Total Energy for Experiment 150 Trial 150
Moving Average
Time
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300
350
400
HNN
HNN
D-HNN
MLP

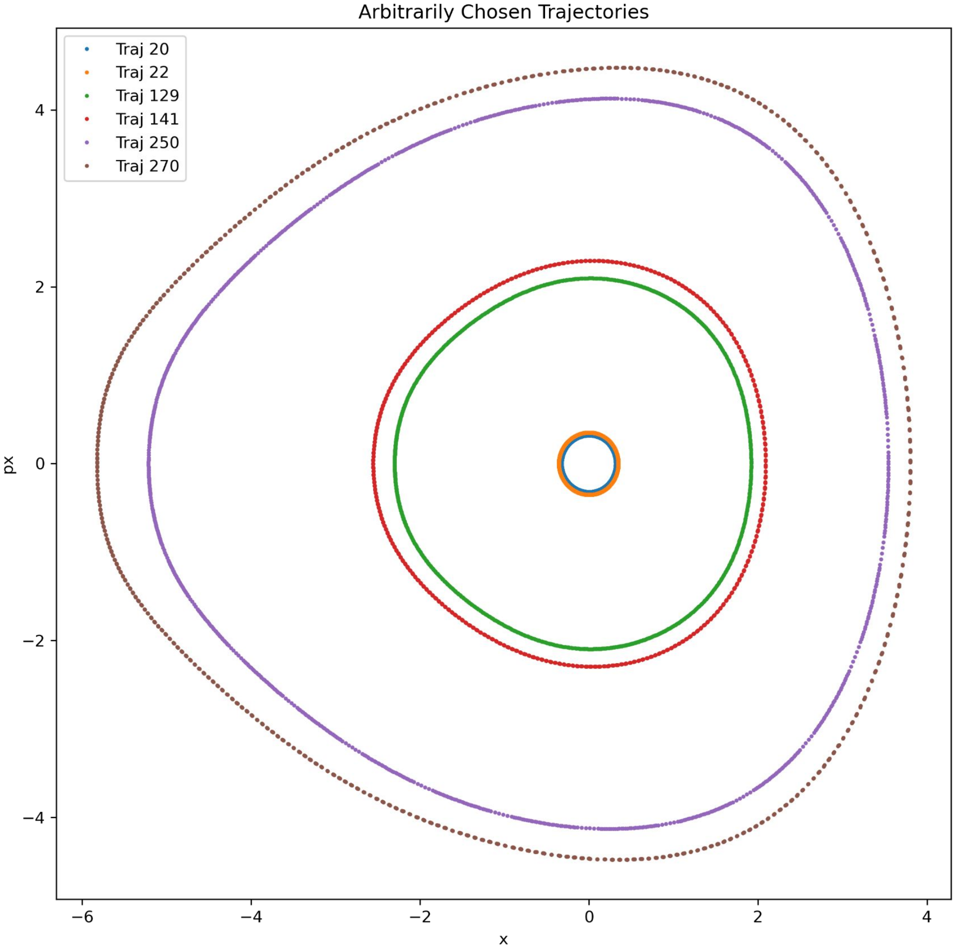

Total energy on 150
15000
12500
10000
7500
5000
2500
0
0
200
400
Time step
Ground truth
HNN
D-HNN
MLP

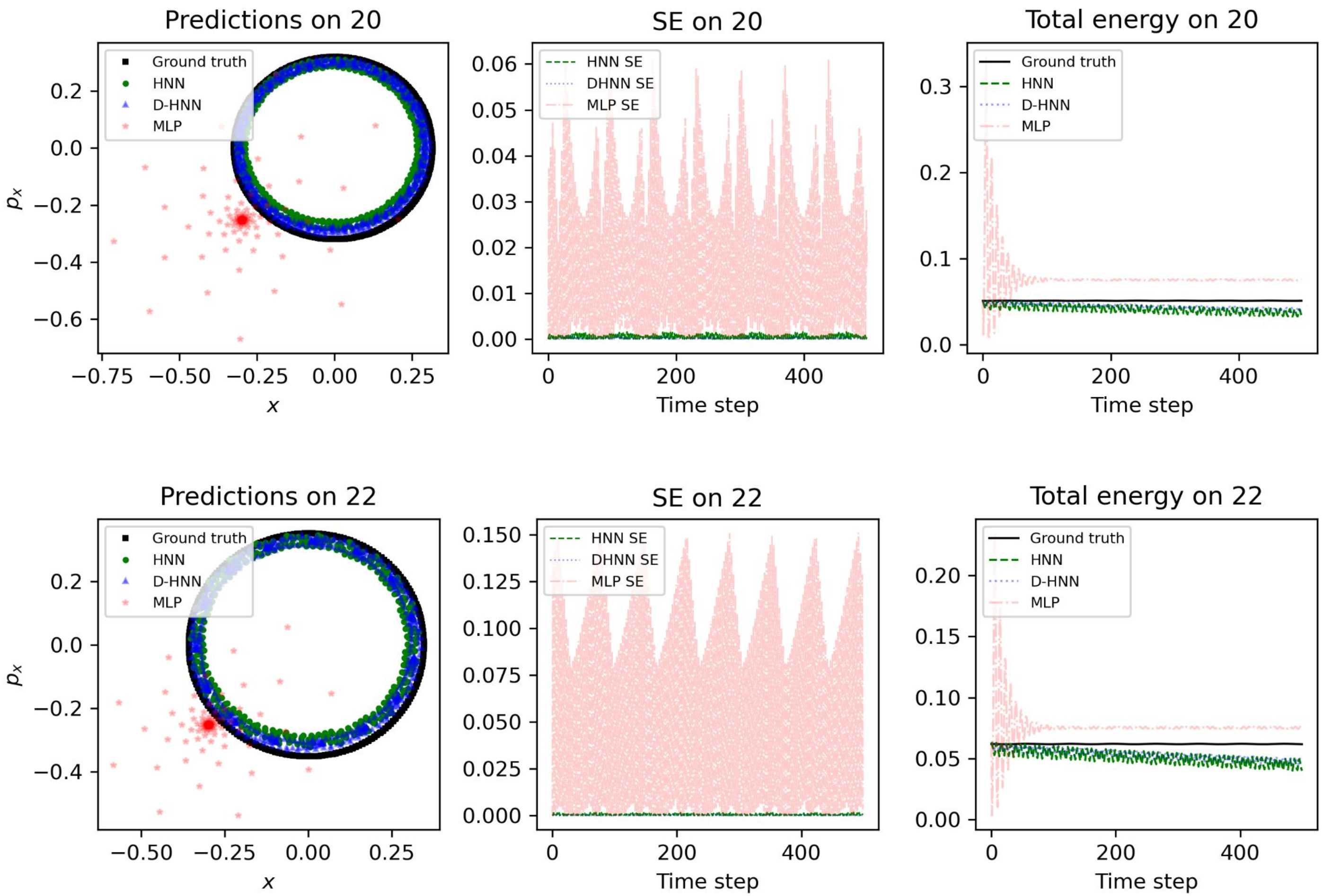

Moving Average of the Total Energy for Experiment 298 Trial 298
HNN
HNN
D-HNN
MLP
Moving Average
0
−100
−200
−300
−400
0
50
100
150
200
250
300
350
400
Time


Total energy on 298
Ground truth
HNN
D-HNN
MLP
250
0
−250
−500
−750
−1000
−1250
0
200
400
Time step

# Model Generalization

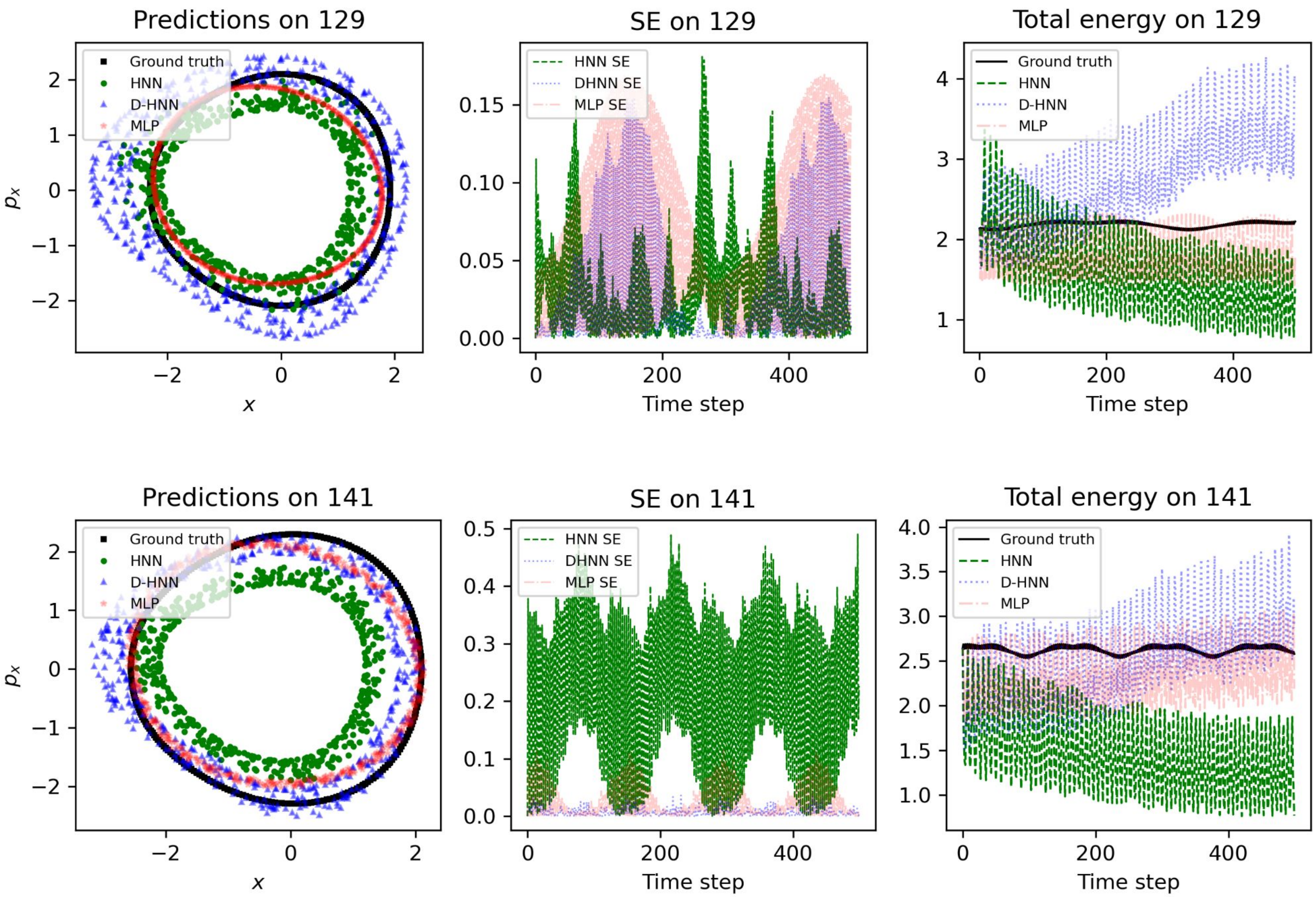
- Get a better idea how well the models do at predicting trajectories with training from a different trajectory
- Traj. Number is general
- Want to analyze inner, middle, and outer trajectories





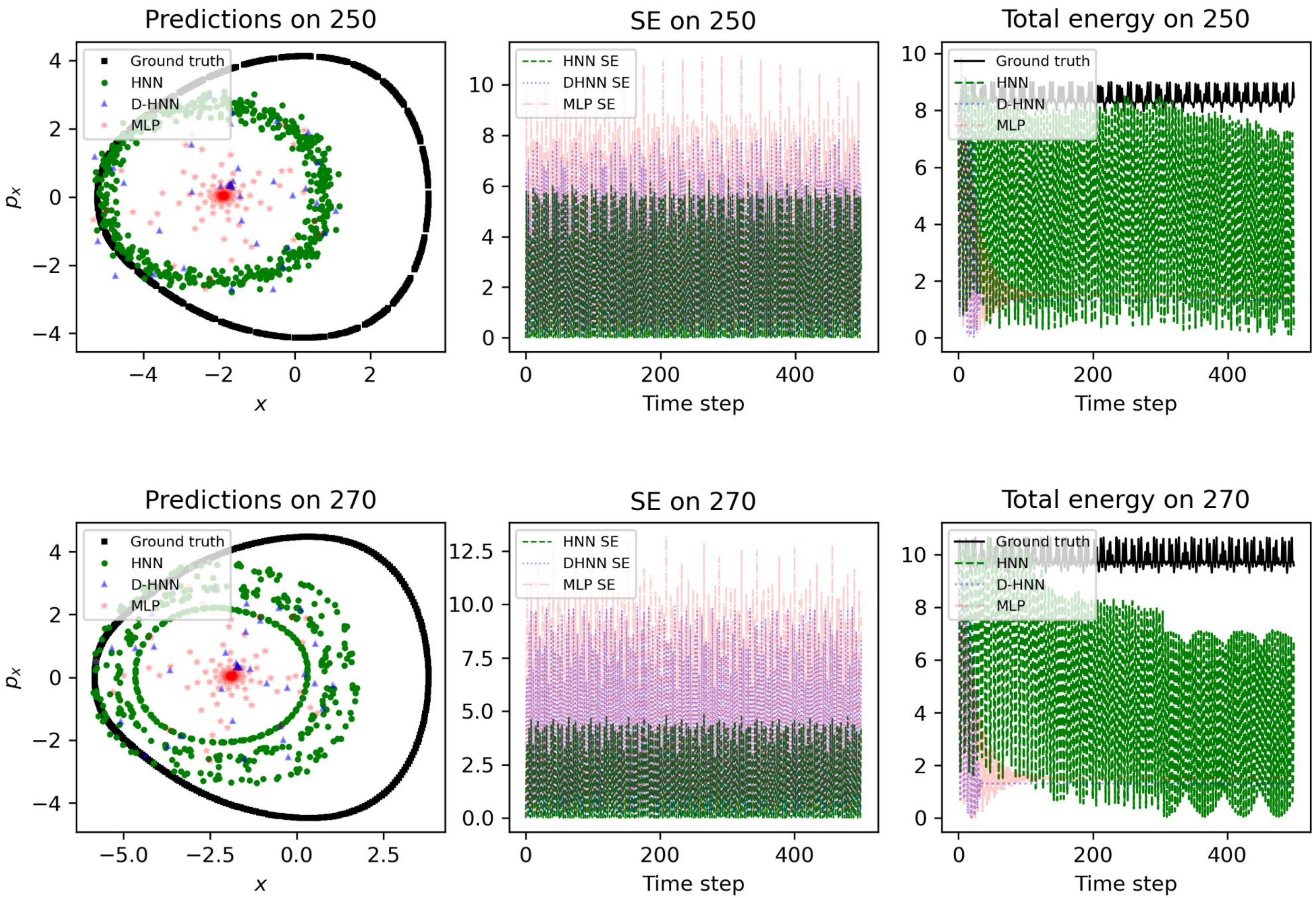

Trained on Trajectory 20

Tested on 20 and 22 (10% larger than 20)



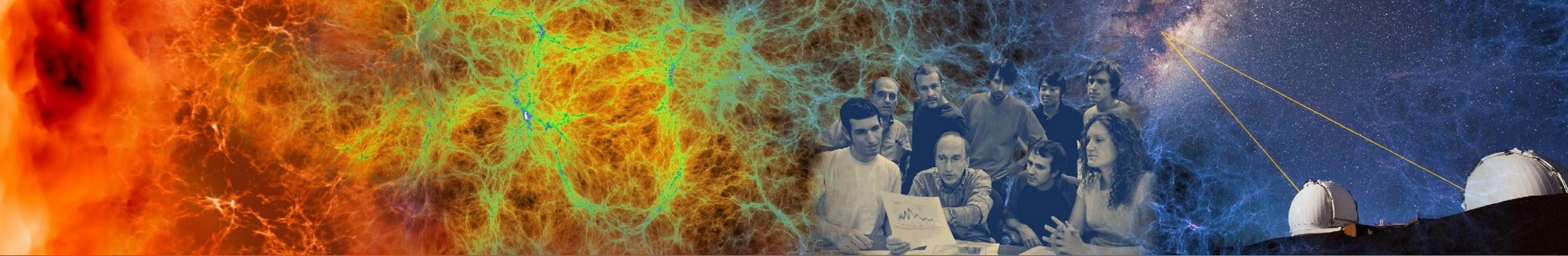

Trained on Trajectory 129

Tested on 129 and 141 (10% larger than 129)




Trained on Trajectory 250

Tested on 250 and 270 (10% larger than 250)



# Conclusion

# Future Endeavors:

- Addition of real world particle accelerator parameters
- Use real particle accelerator data

# In Conclusion,

- Hamiltonian Neural Networks (HNN) for predicting particle trajectories in phase space shows significant potential for efficiency
- HNN has demonstrated the ability to track particles within the desired region of phase space
- Further investigations could establish HNN as a reliable and efficient approach in this domain
- Compelling area of study for the future development of efficient particle tracking methods

# Acknowledgments:

Special thank you to,

Mentor: Dr. Lipi Gupta, LBNL

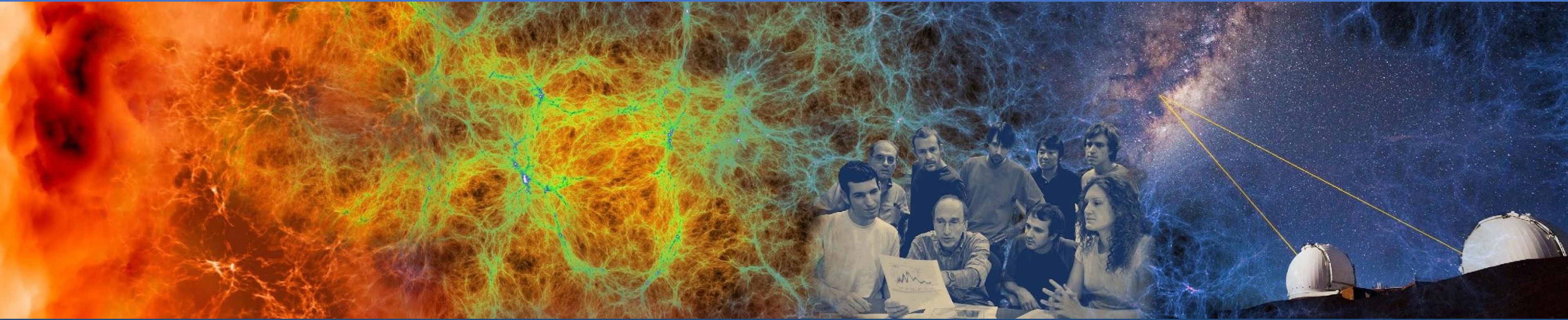
Thesis Committee:
Dr. Jen Mei Chang, Dr. Xiyue Liao, and Dr. Paul Sun


## References:


[1] Gupta, Lipi. “Analytic and Machine Learning Methods for Controlling Nonlinearities in Particle Accelerators.” PhD diss. University of Chicago, 2021.

[2] S. Greydanus, A. Sosanya. Dissipative Hamiltonian Neural Networks: Learning Dissipative and Conservative Dynamics Separately. arXiv preprint arXiv:2201.10085, 2022.

[3] S. Greydanus, M. Dzamba, J. Yosinski. Hamiltonian Neural Networks. arXiv preprint arXiv:1906.01563, 2019.

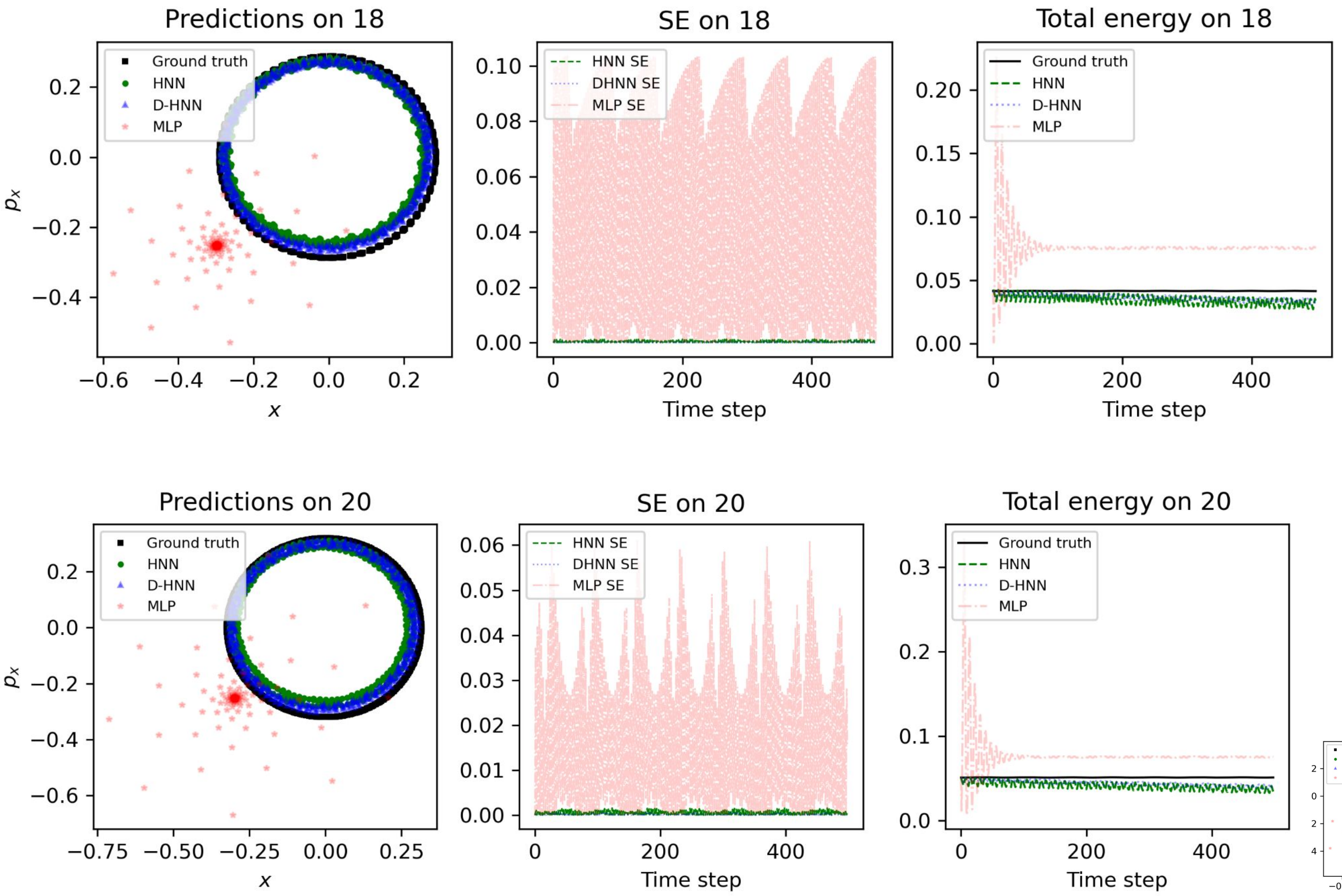



# Efficiency

- In one study, traditional numerical tracking methods required approximately 17 hours to track 500 turns of 61 particle trajectories

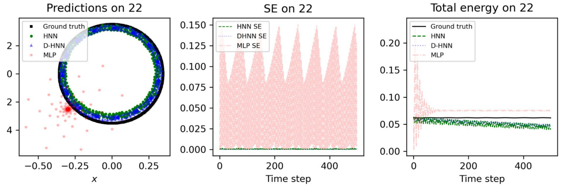
Neural networks significantly reduce the time required for particle tracking

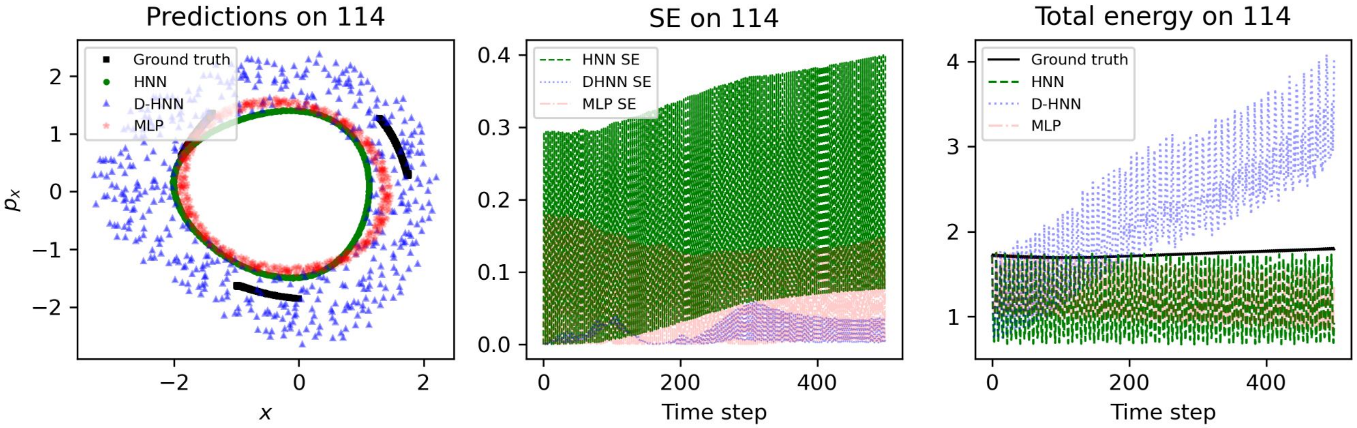
- 24 minutes on average required to train, test (on HNN, MLP, and DHNN), and analyze the data for each experiment
- Ability to run multiple jobs simultaneously

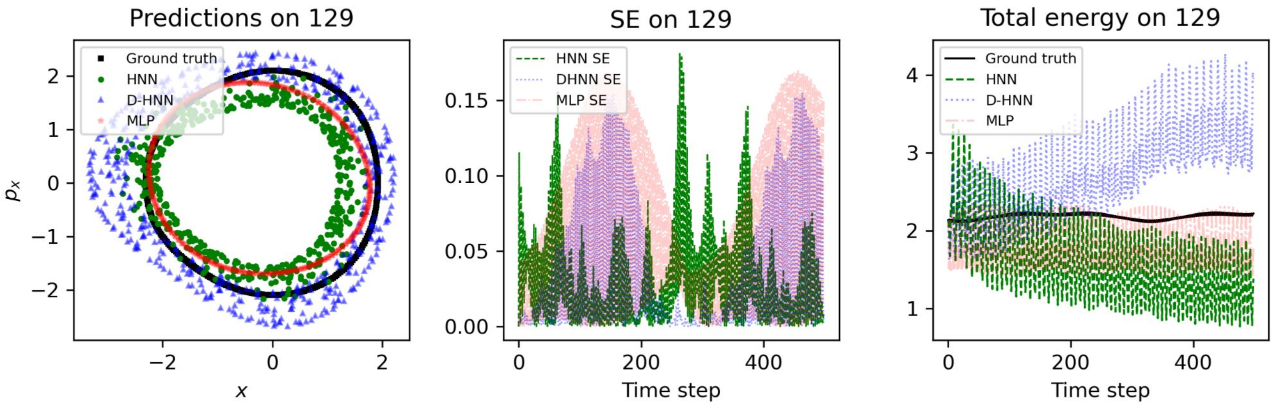



Trained on Trajectory 20

Tested on 20 and 18 (10% smaller than 20)

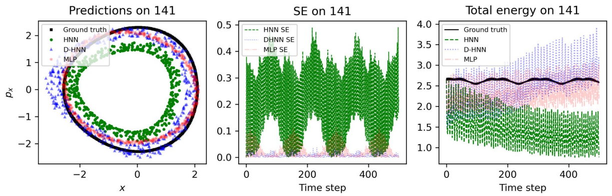


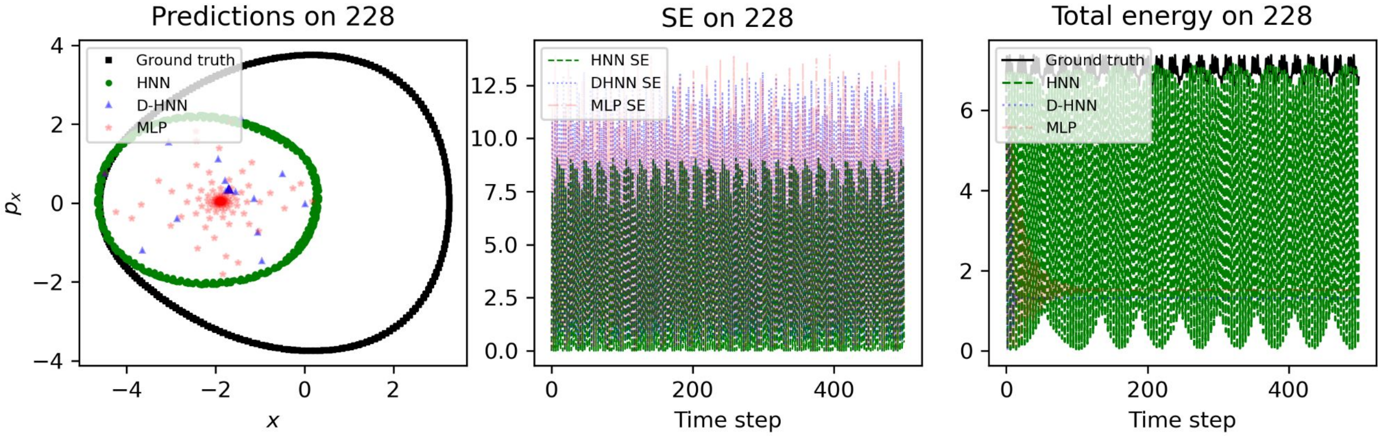



Trained on Trajectory 129

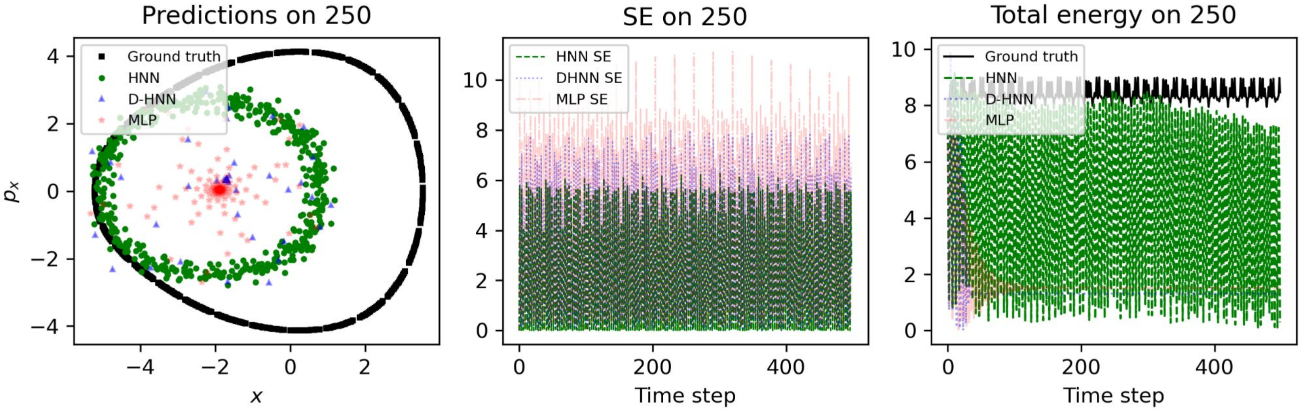
Tested on 129 and 114 (10% smaller than 129)

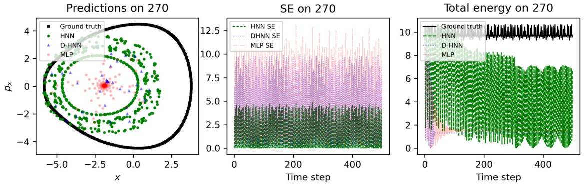





Trained on Trajectory 250

Tested on 250 and 228 (10% smaller than 250)



# More about HNN algorithm :

- 3 hidden layers, 256 nodes per layer
- Gradient calculation involves taking the partial derivatives with respect to both position and momentum, which creates a challenge for backpropagation because of the mathematical properties of Hamiltonian dynamics
- Split the gradient into two parts:
    - Gradient with respect to position
    - Gradient with respect to momentum
- Swap the data to ensure that neural network learns the symplectic structure of the Hamiltonian, ideally want to satisfy Hamilton's Equations:

$$\frac{\partial \mathcal{H}}{\partial p_x} = \frac{\partial x}{\partial t}\ , \quad -\frac{\partial \mathcal{H}}{\partial x} = \frac{\partial p_x}{\partial t}$$