Solving the Gravitational N-body problem with Machine Learning

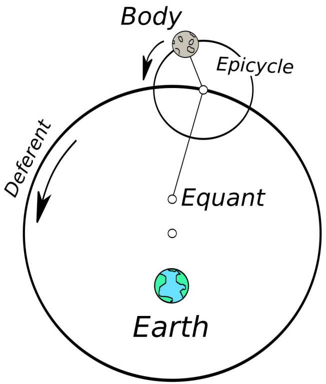
Verónica Saz Ulibarrena

Contents

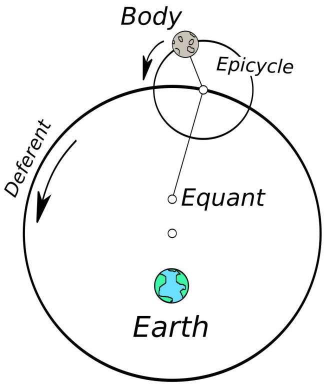
Contents

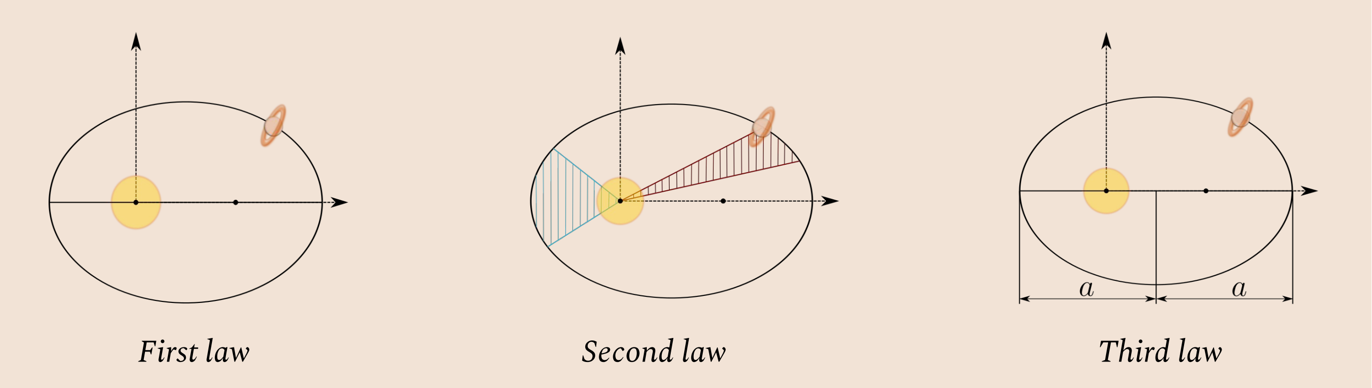
Contents

The beginning of the N-body problem

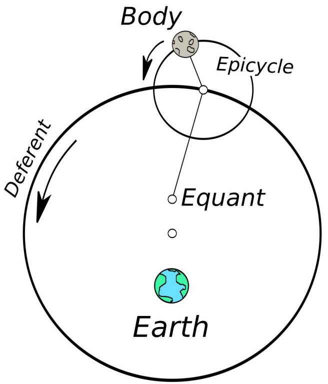


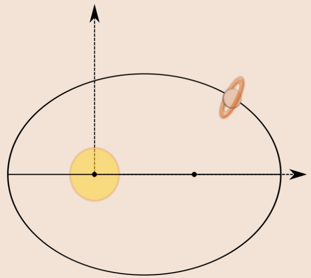
The beginning of the *N*-body problem

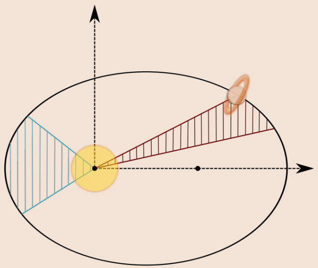


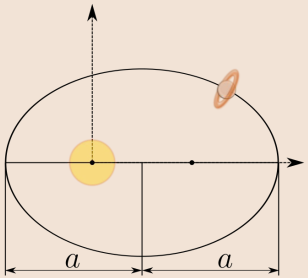


The beginning of the N-body problem

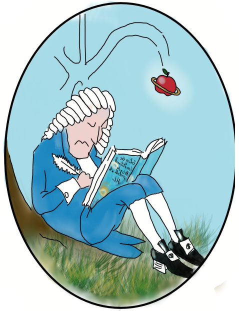



First law

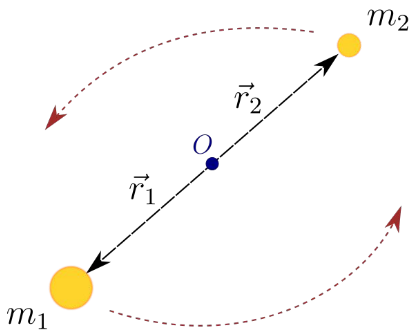

Second law



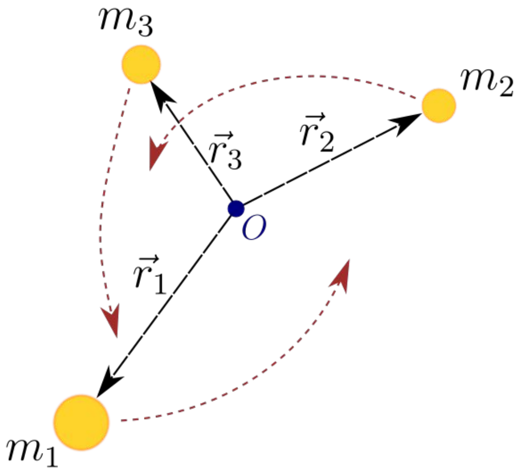
Third law



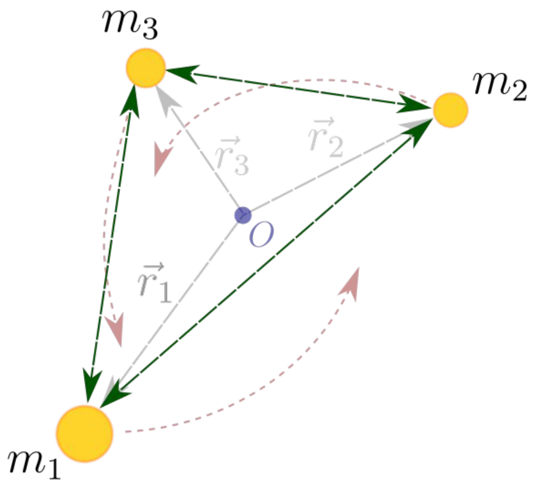
$$\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\,\vec{r}_{12}$$



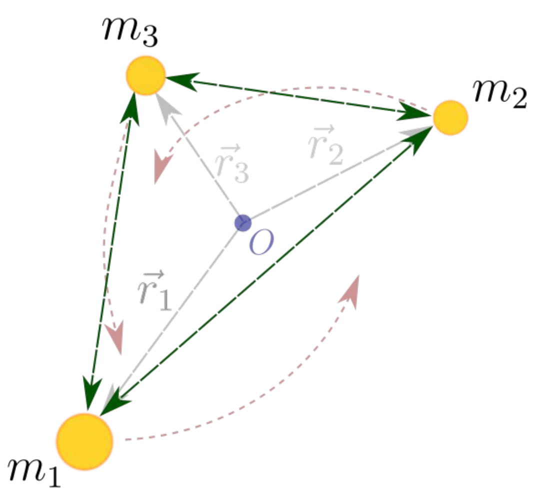
The three-body problem

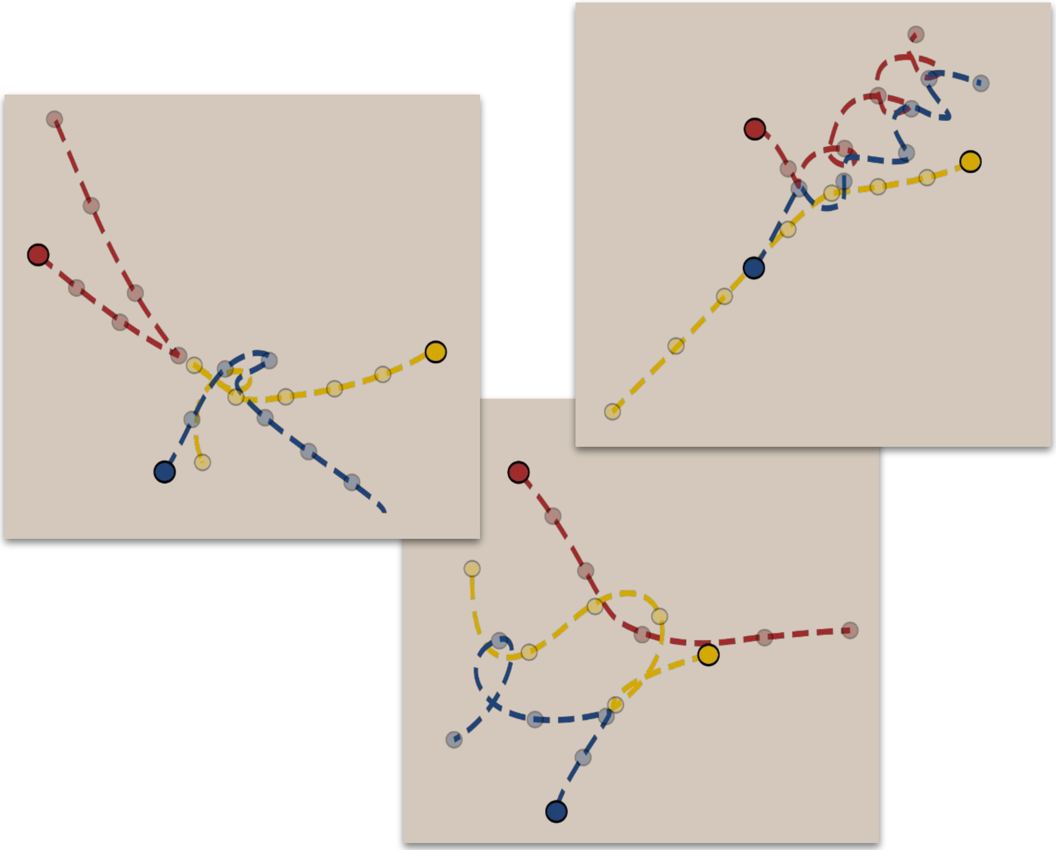


The three-body problem

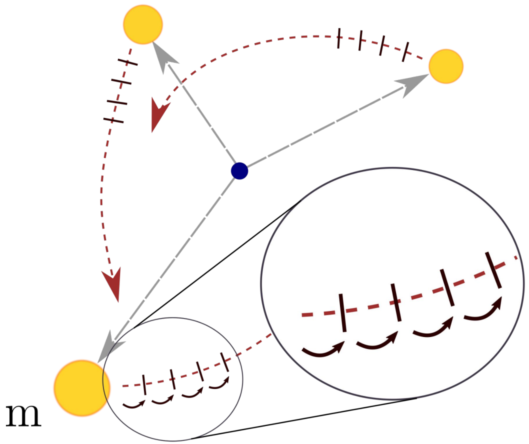


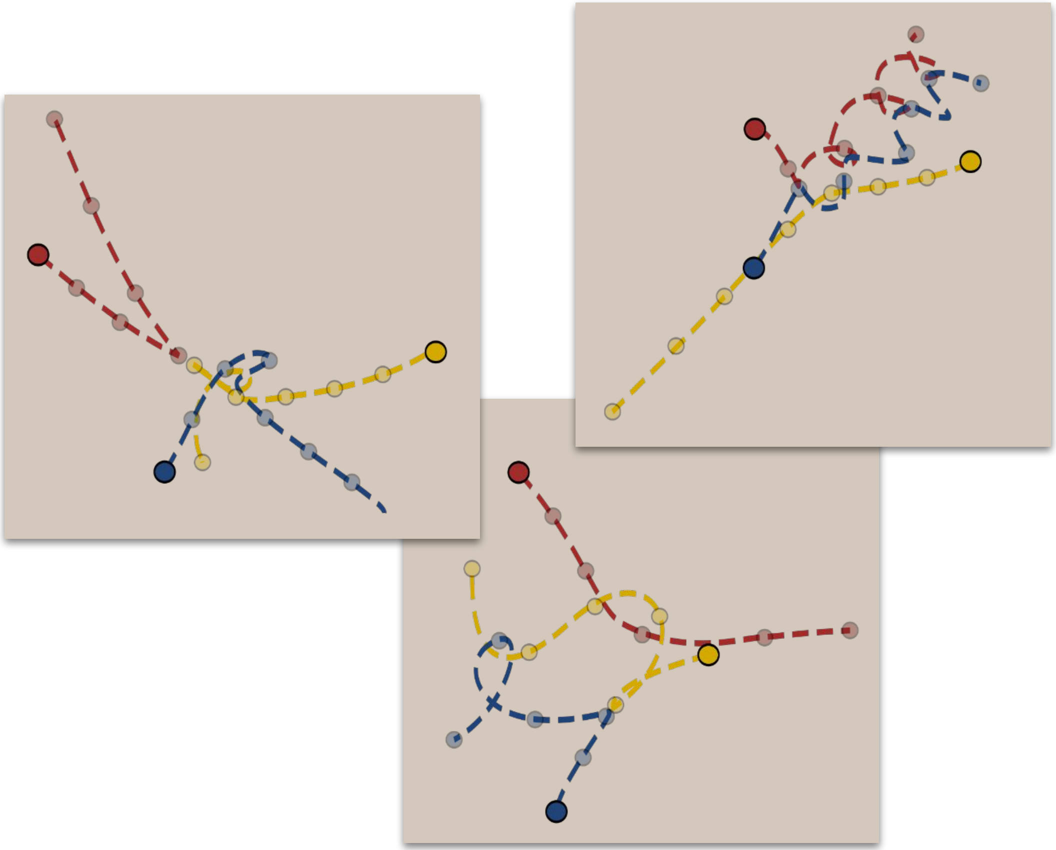
The three-body problem



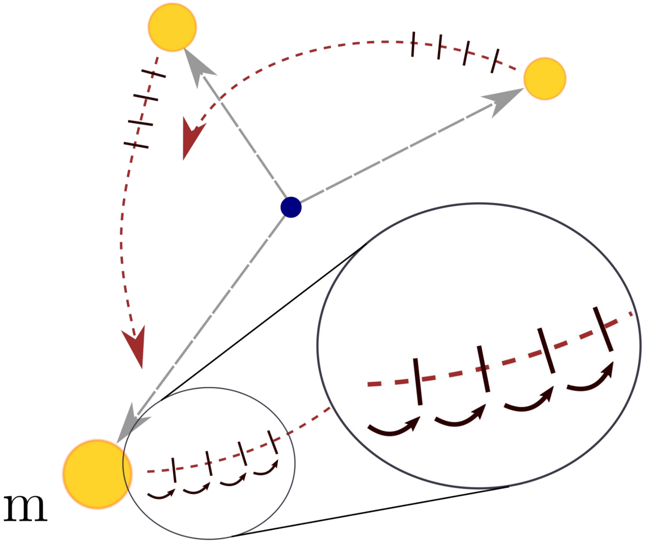


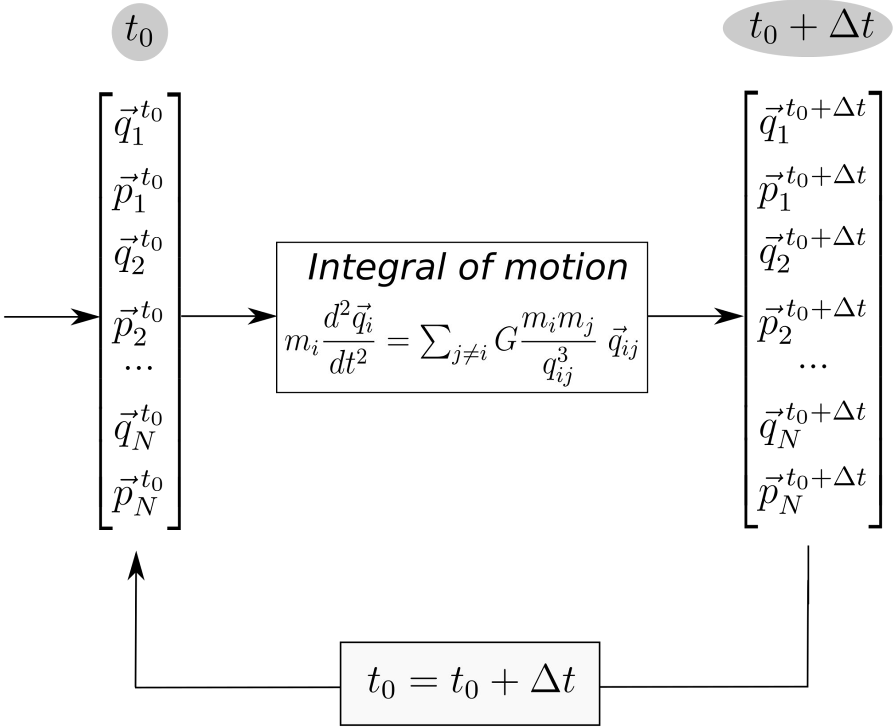
The three-body problem



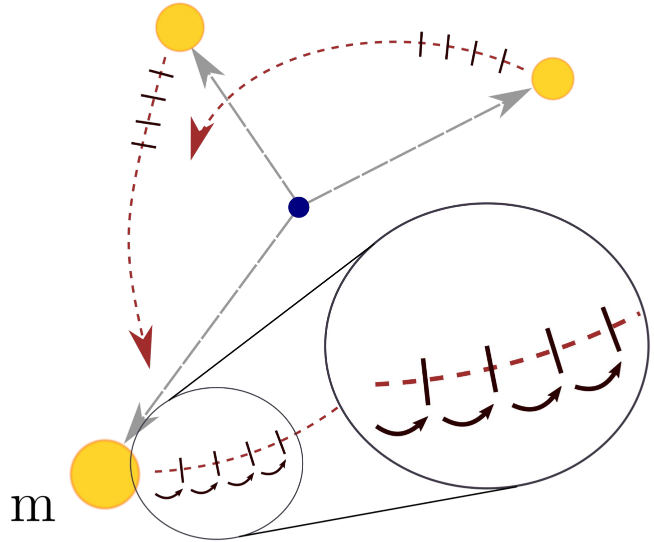


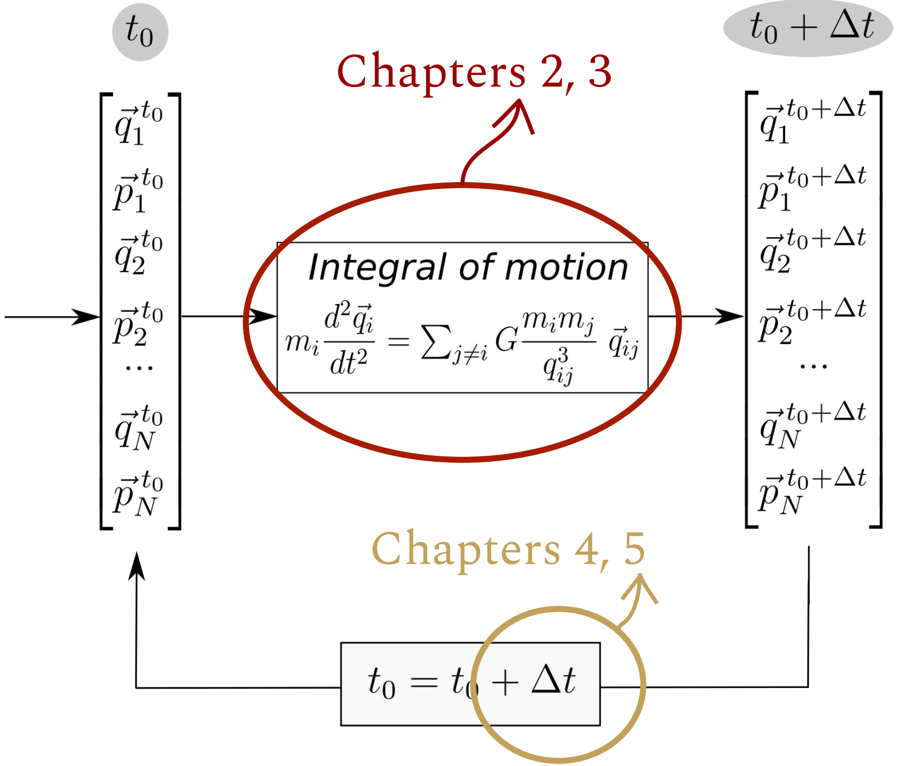
Numerical integration





Numerical integration



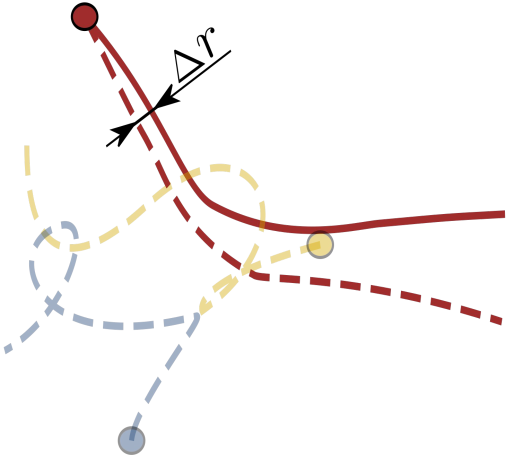


Numerical Error

$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} - \underbrace{G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

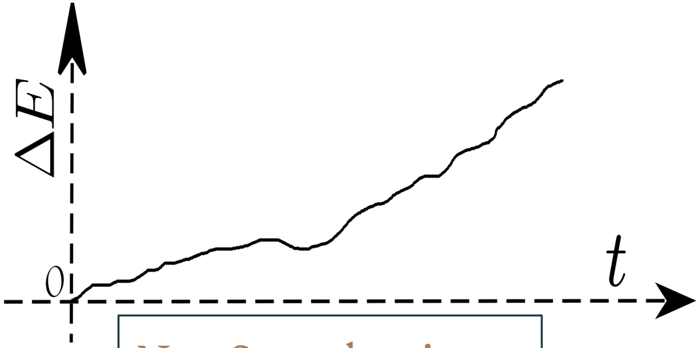
Numerical Error

$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} - \underbrace{G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

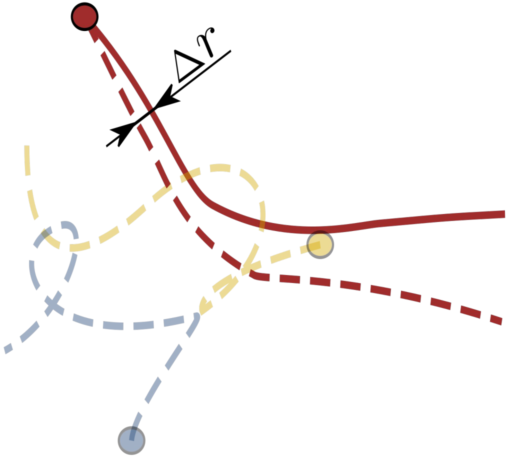


Numerical Error

$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} - \underbrace{G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

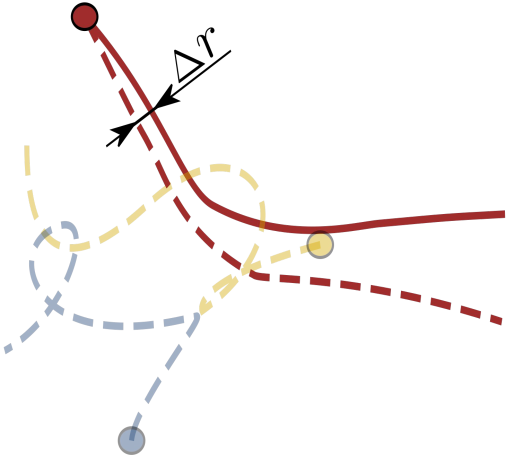


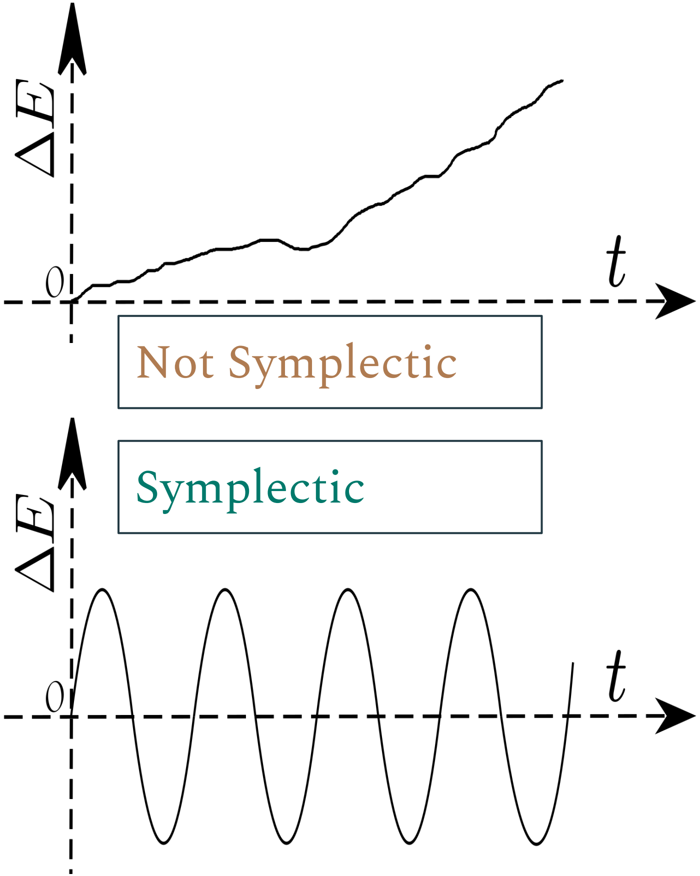
Not Symplectic



Numerical Error

$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} - \underbrace{G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

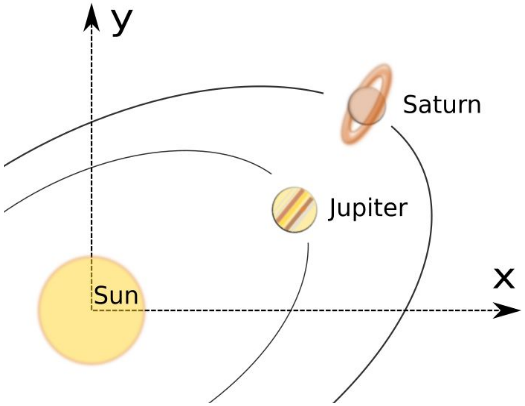


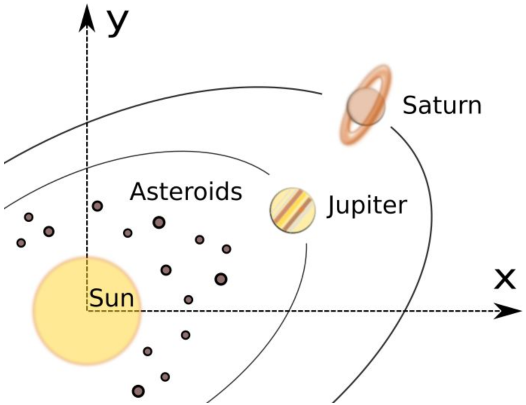


Chapter 2

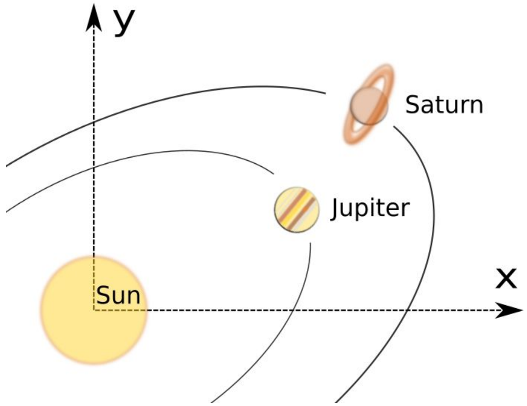
A hybrid approach for solving the gravitational N-body problem with Artificial Neural Networks

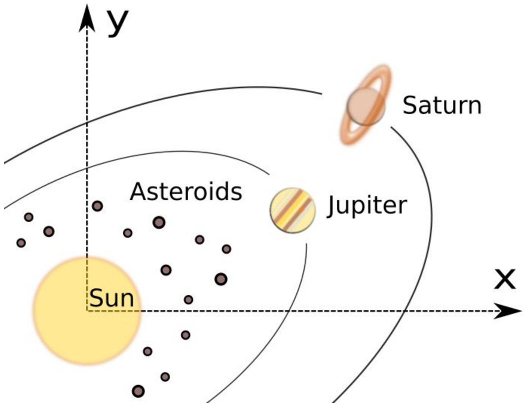
Planetary Systems





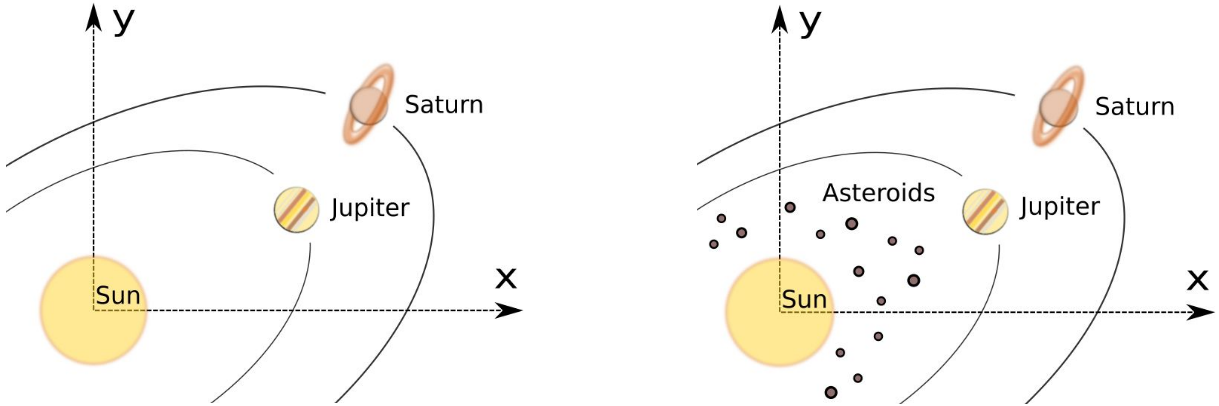
Planetary Systems





$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} \underbrace{- G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

Planetary Systems

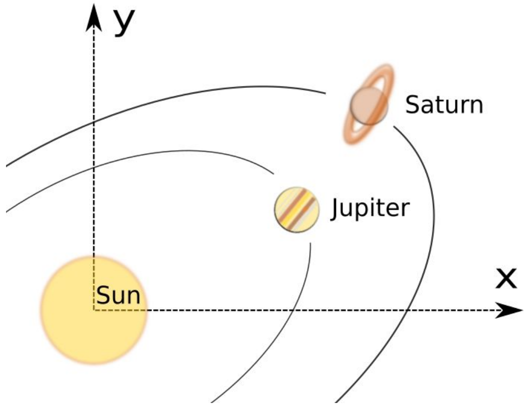


$$\mathcal{H} = \underbrace{\sum_{i=0}^{N-1} \frac{||\vec{p}_i||^2}{2m_i}}_{\text{Kinetic Energy}} \underbrace{- G \sum_{i=0}^{N-2} m_i \sum_{j=i+1}^{N-1} \frac{m_j}{||\vec{q}_j - \vec{q}_i||}}_{\text{Potential Energy}}$$

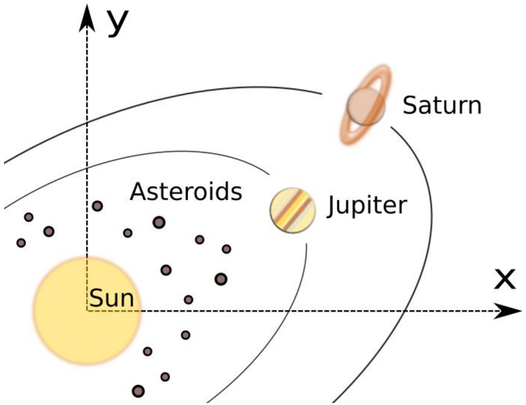
Wisdom-Holman Integrator

$$\mathcal{H} = \mathcal{H}_{Kepler} + \mathcal{H}_{perturbations}$$

Planetary Systems



a) **SJS**

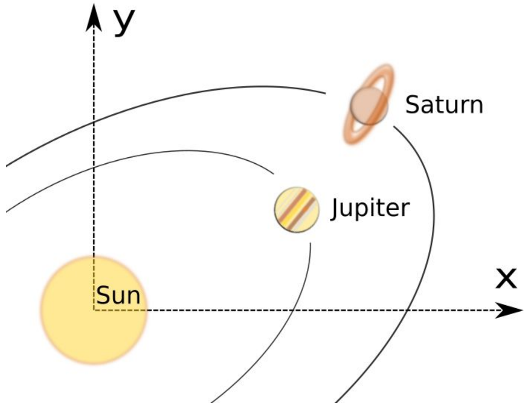


b) **SJSa**

Perturbations

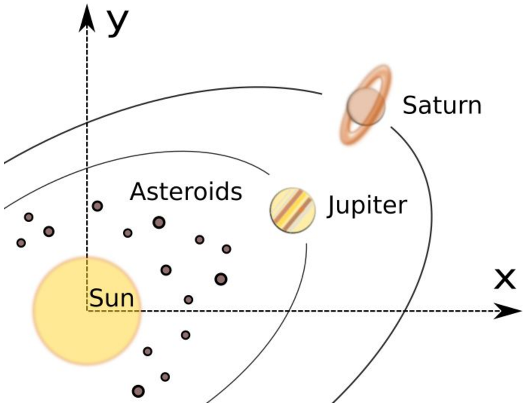
$$\boxed{\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\ \vec{r}_{12}} \quad \rightarrow \quad \mathcal{O}(N^2)$$

Planetary Systems



a) SJS

Fast (cheap)



b) SJSa

Expensive as N increases

Perturbations

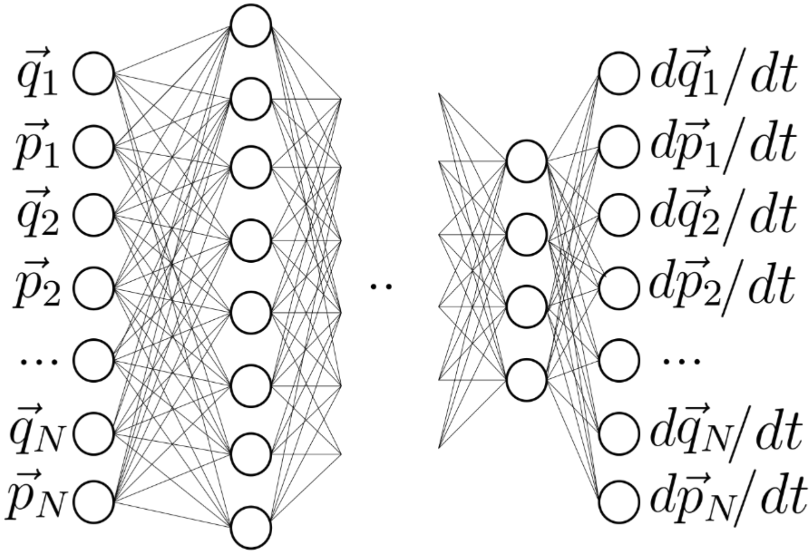
$$\boxed{\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\ \vec{r}_{12}} \quad \rightarrow \quad \mathcal{O}(N^2)$$

Neural Networks

Perturbations

$$\boxed{\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\,\vec{r}_{12}}$$

Deep Neural Networks

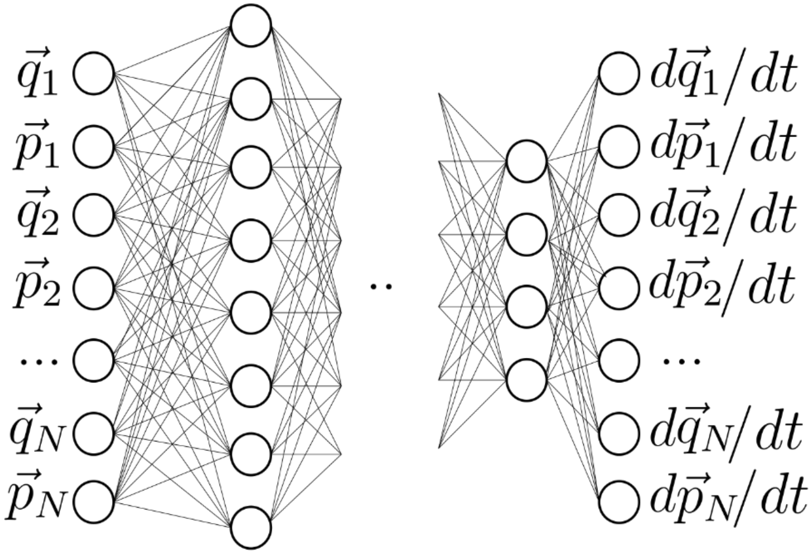


Neural Networks

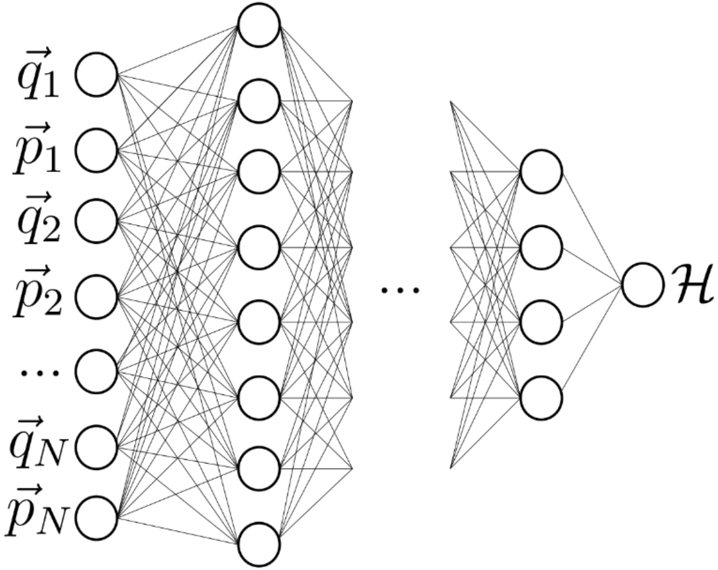
Perturbations

$$\boxed{\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\,\vec{r}_{12}}$$

Deep Neural Networks



Hamiltonian Neural Networks



Neural Networks

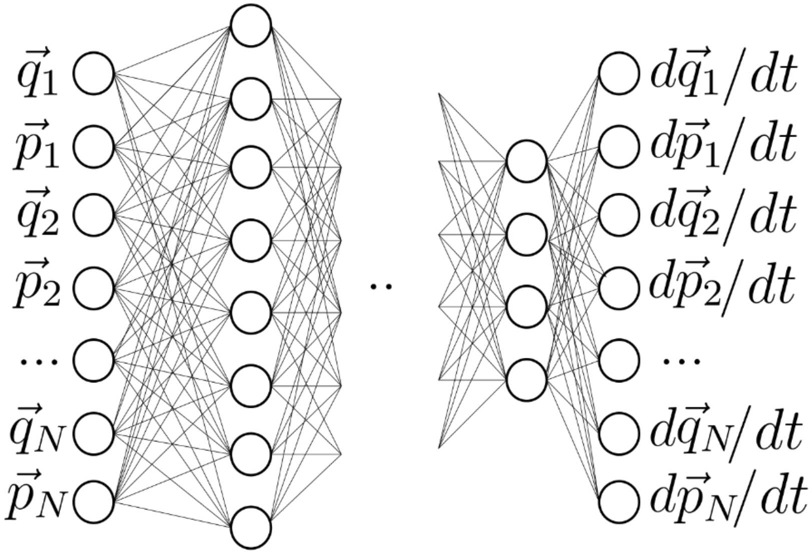
Math Alert

$$\mathcal{H} \to \boxed{\begin{array}{c} -\dfrac{\partial \mathcal{H}}{\partial \vec{q}} = \dfrac{d\vec{p}}{dt} \\[2ex] \dfrac{\partial \mathcal{H}}{\partial \vec{p}} = \dfrac{d\vec{q}}{dt} \end{array}}$$

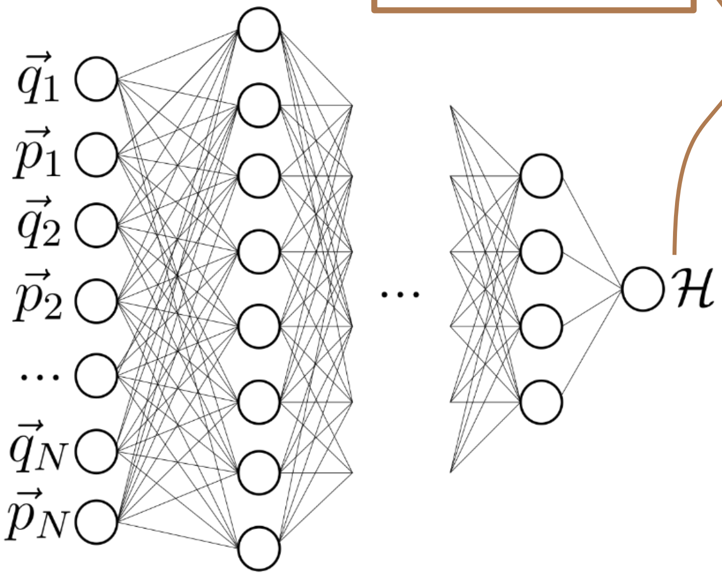
Perturbations

$$\boxed{\vec{F} = G\frac{m_1 m_2}{|\vec{r}_{12}|^3}\,\vec{r}_{12}}$$

Deep Neural Networks

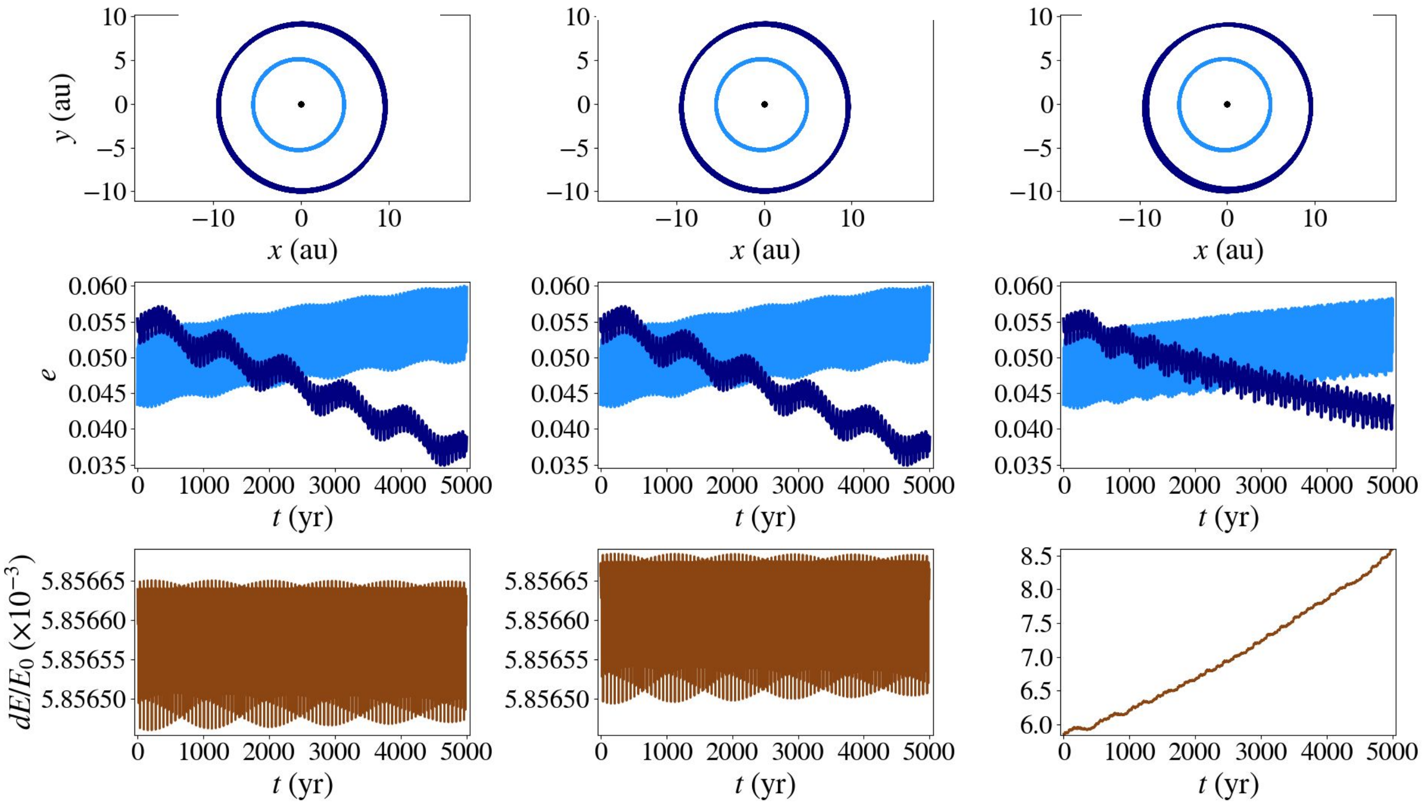


Hamiltonian Neural



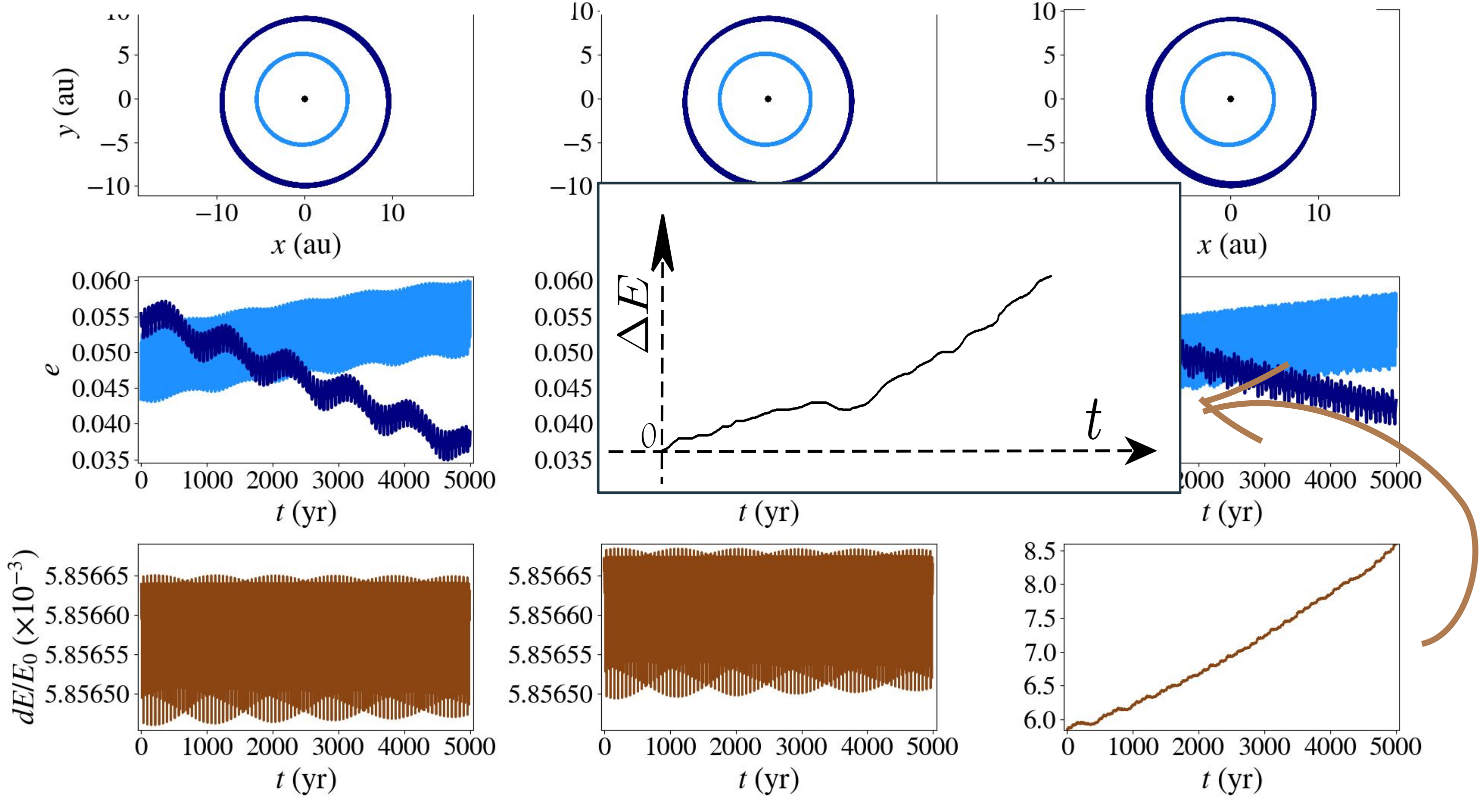
Results

No NN | HNN | DNN



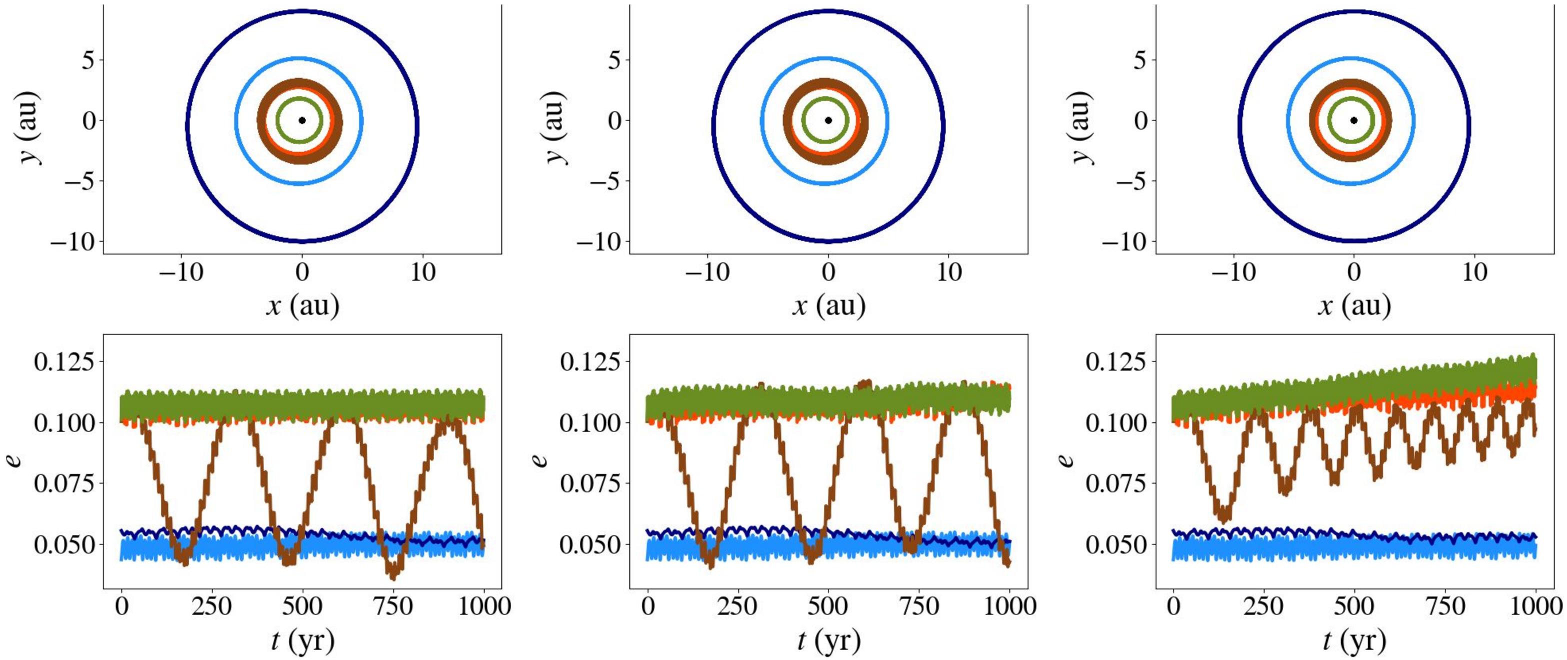
Results

No NN HNN DNN

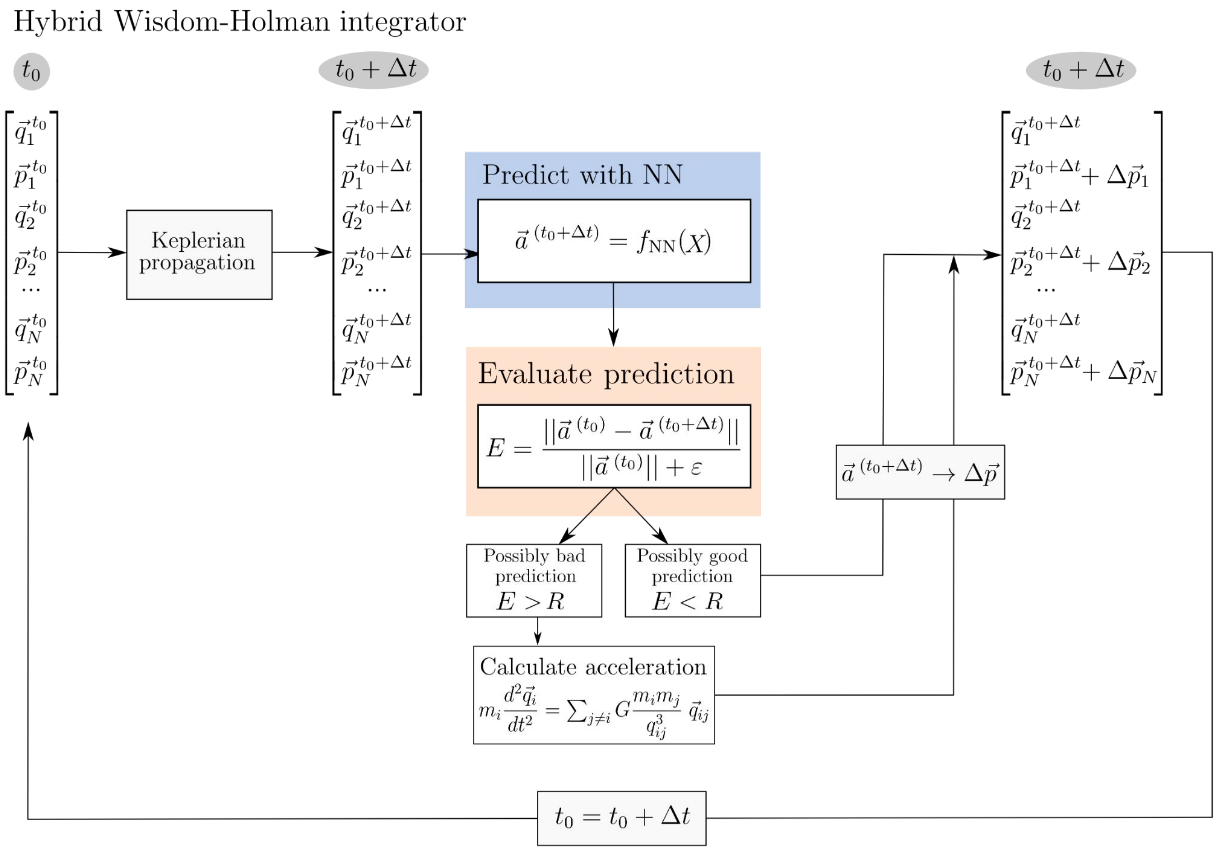


Results

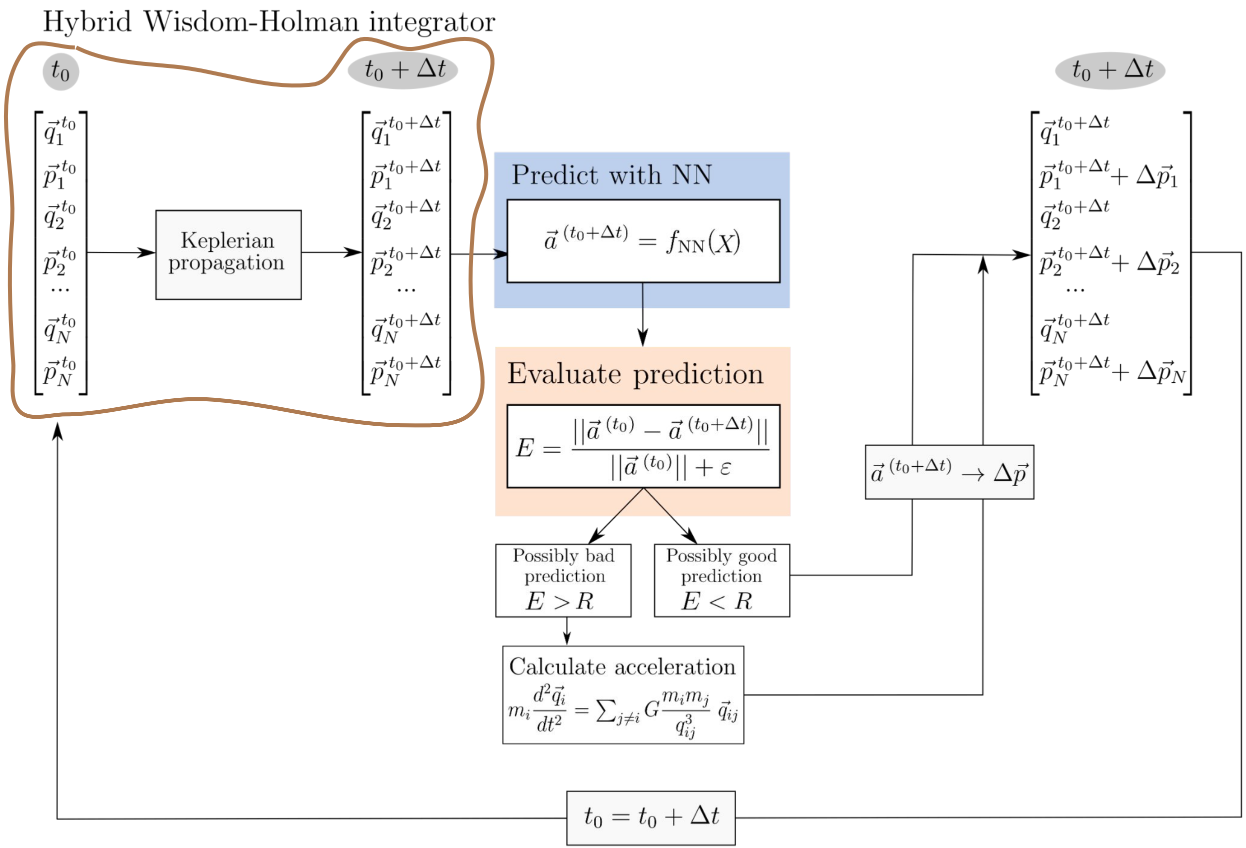
No NN | **HNN** | **DNN**



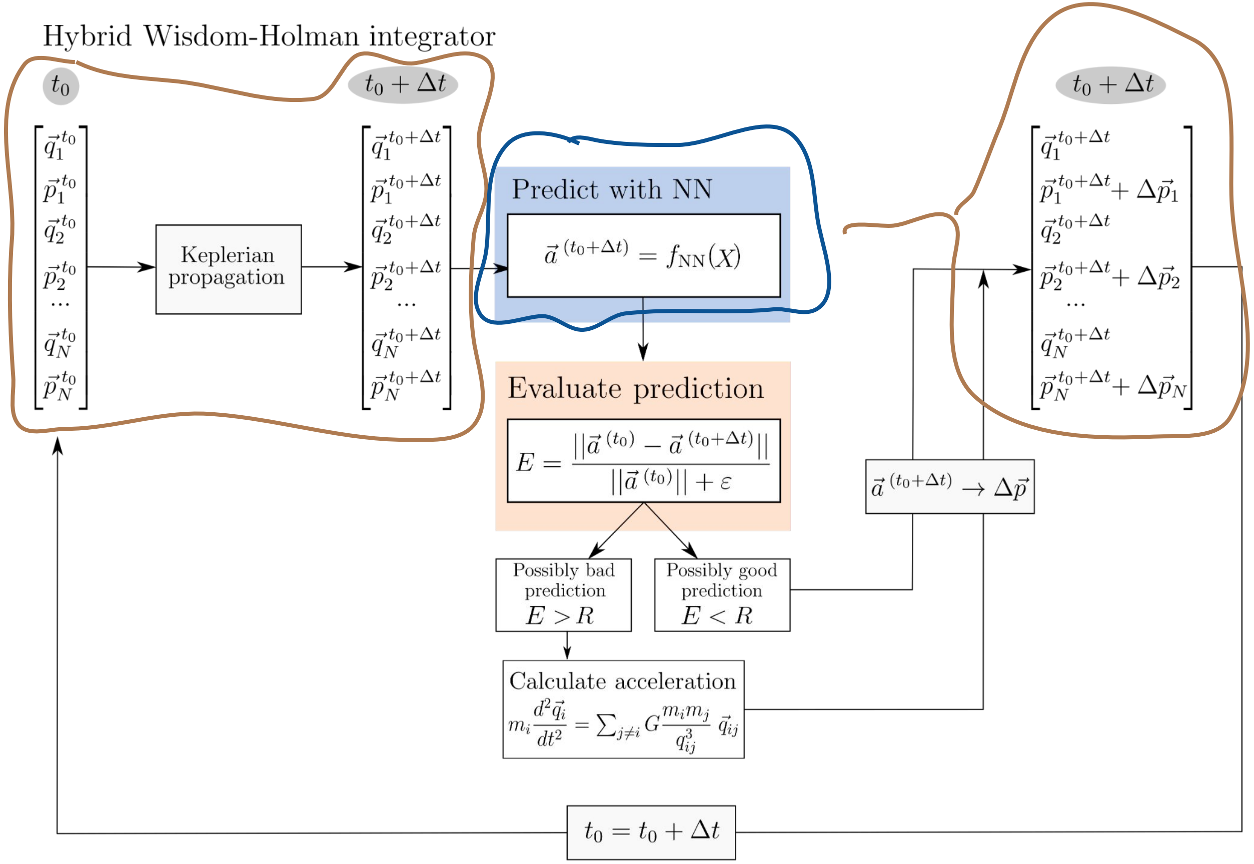
Hybrid method



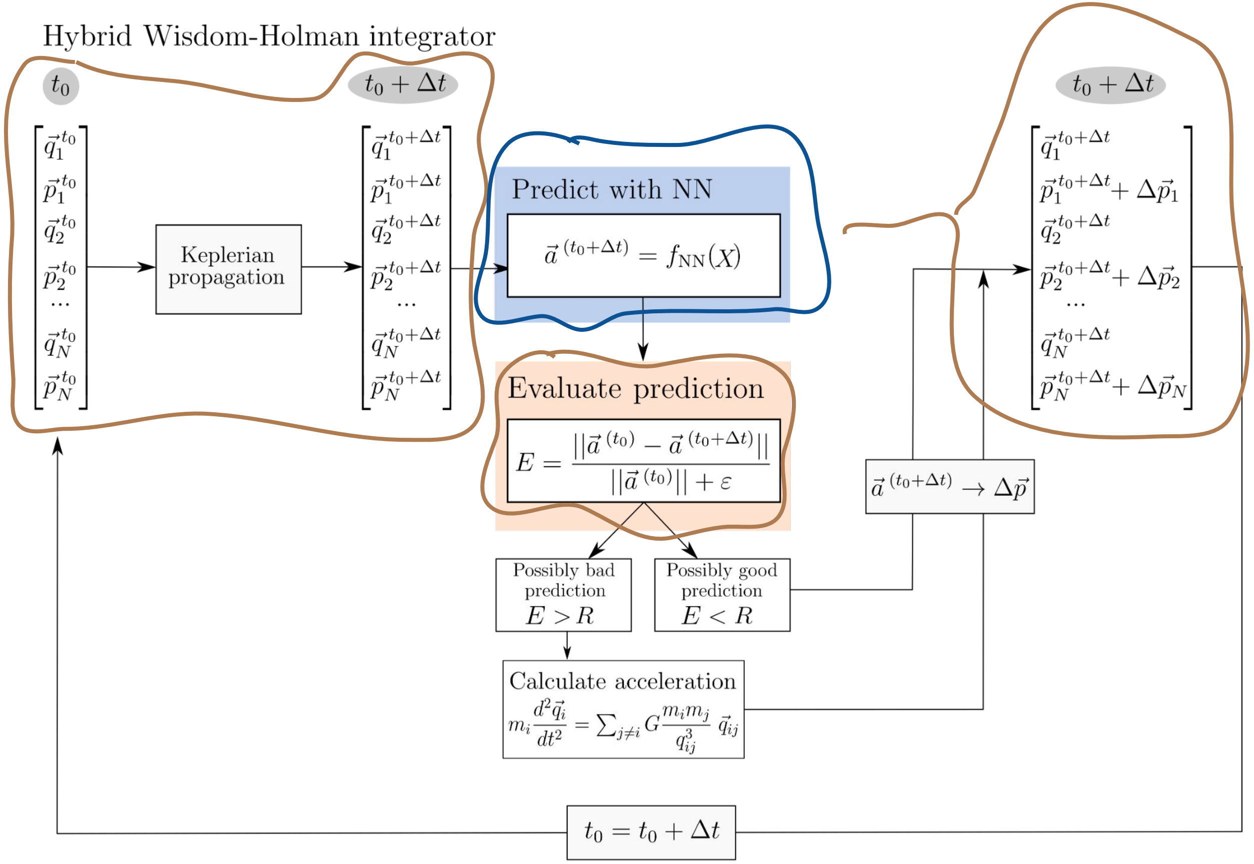
Hybrid method



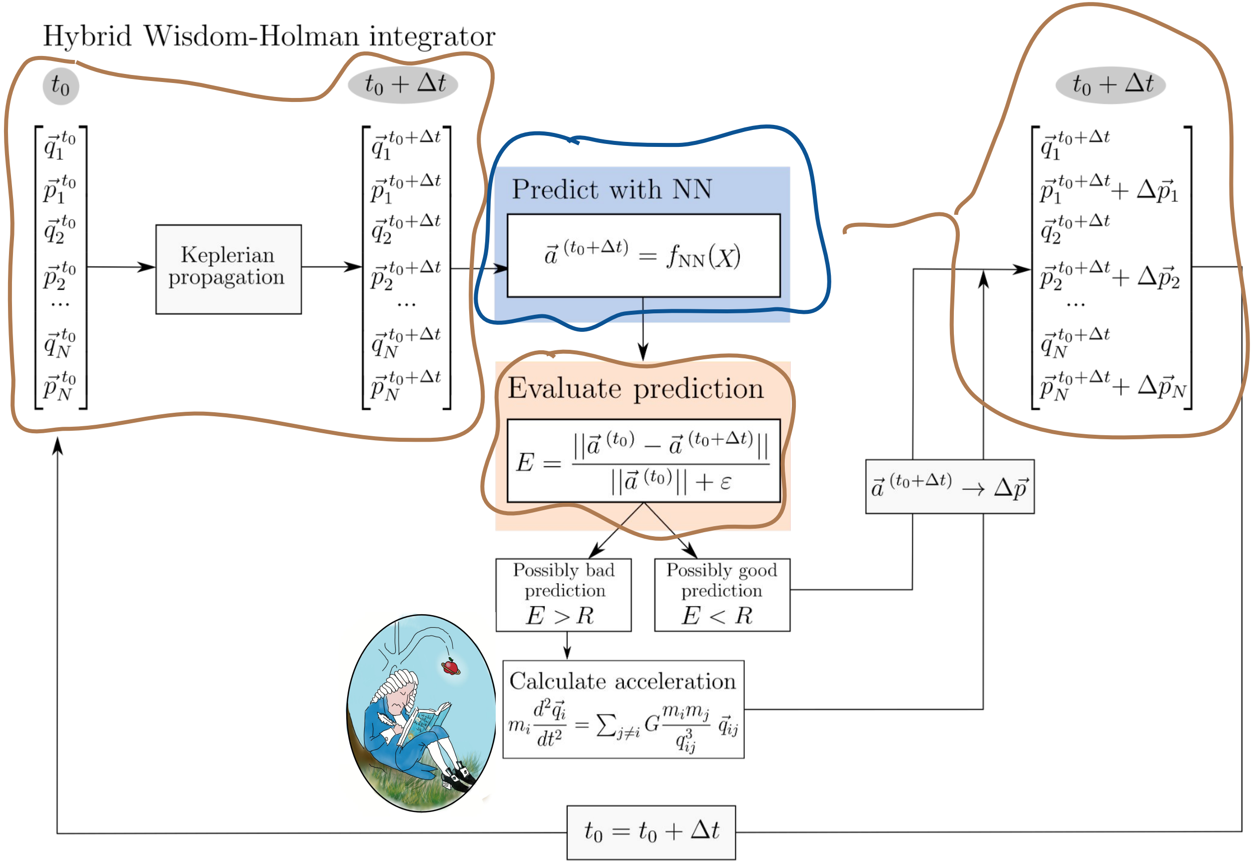
Hybrid method

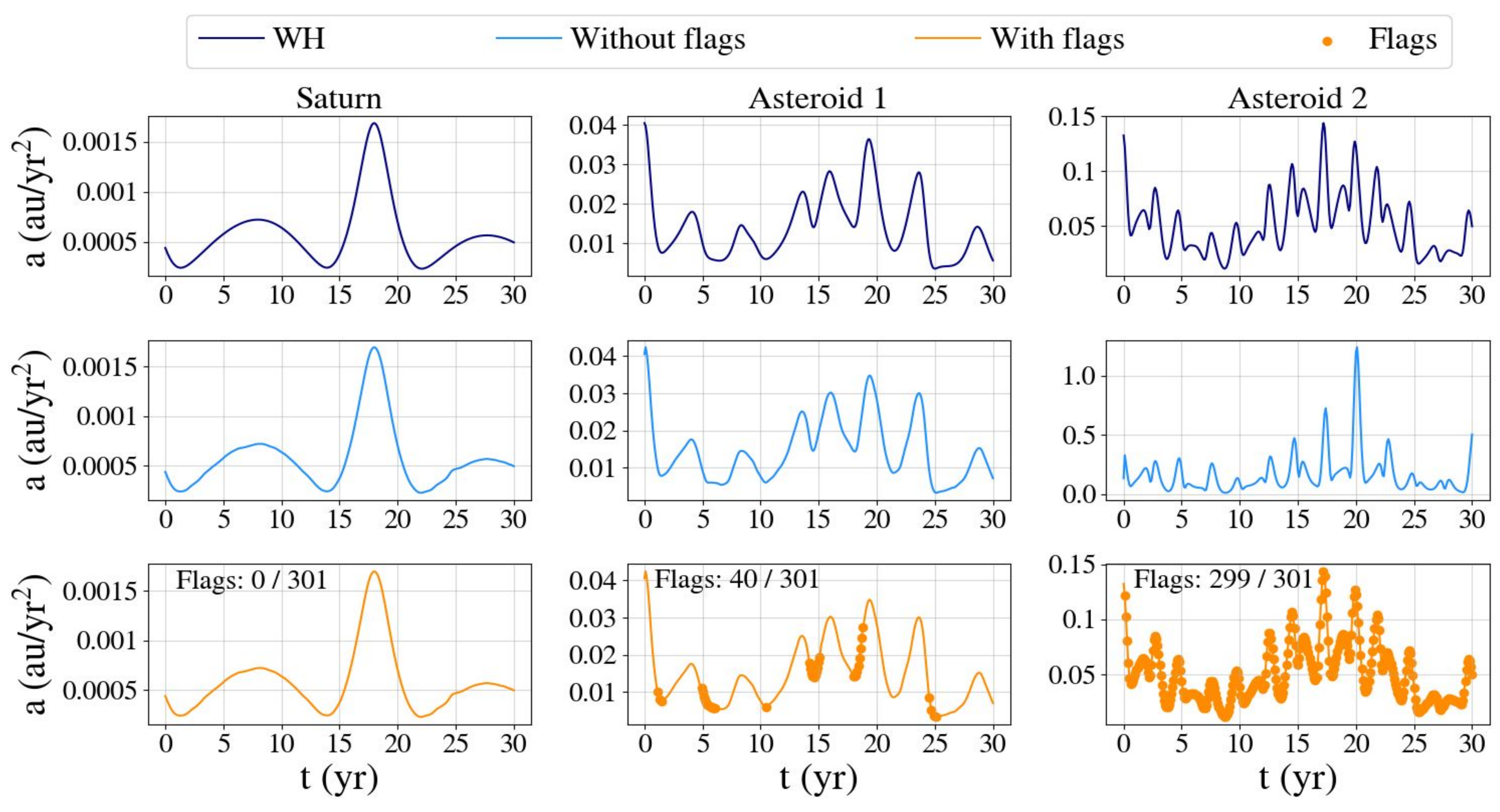


Hybrid method

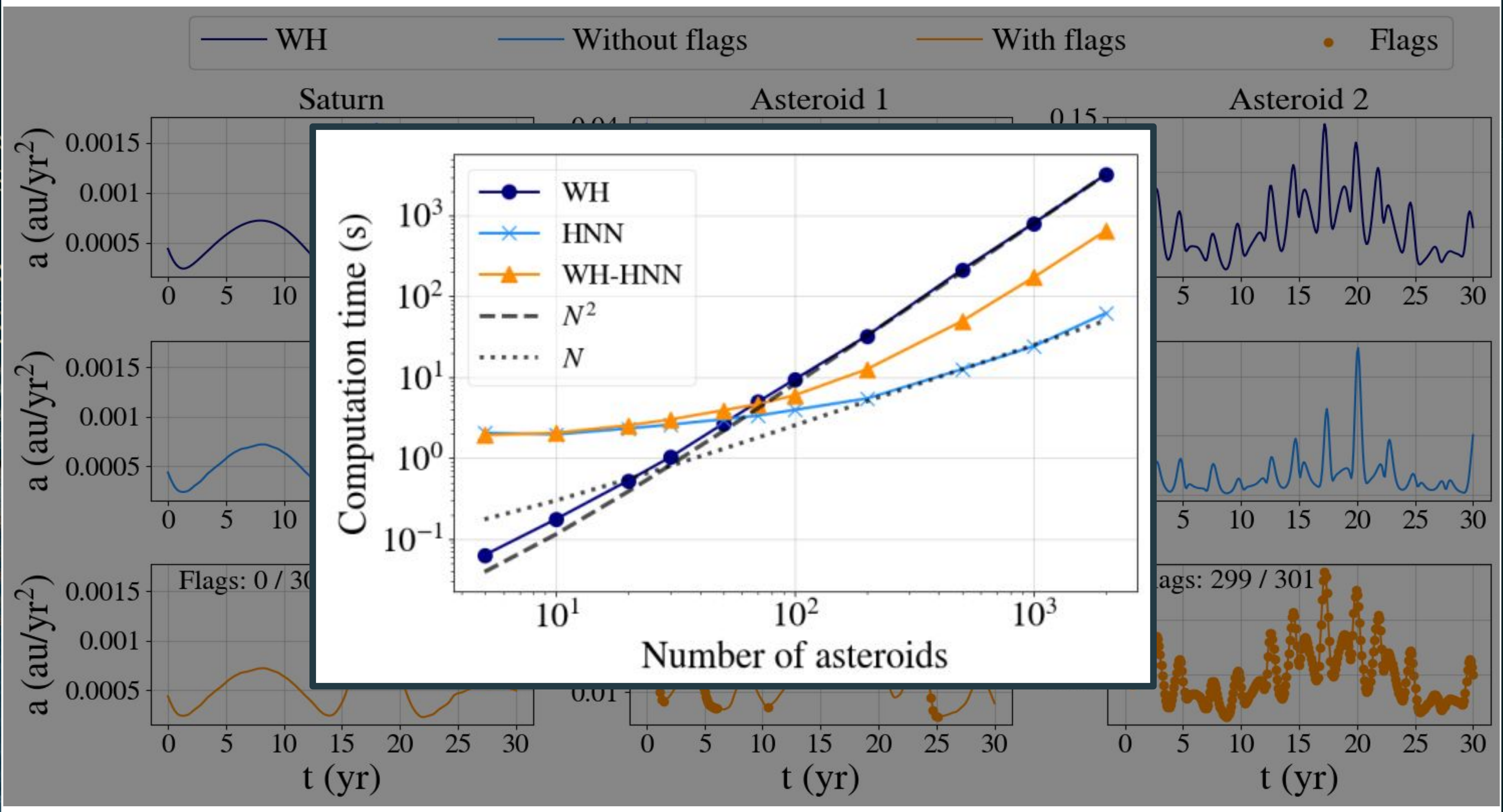


Hybrid method





WH
Without flags
With flags
Flags
Saturn
Asteroid 1
Asteroid 2
a (au/yr^2)
Flags: 0 / 301
Flags: 40 / 301
Flags: 299 / 301
t (yr)


WH
Without flags
With flags
Flags
Saturn
Asteroid 1
Asteroid 2
a (au/yr^2)
Flags: 299 / 301
t (yr)
Computation time (s)
Number of asteroids
WH
HNN
WH-HNN
N^2
N

Conclusions and disclaimers

HNNs can conserve energy error better — DNNs are easier to train

Errors accumulate — Hybrid method

Increasing N implies retraining

Accuracy of NNs lower than integrators

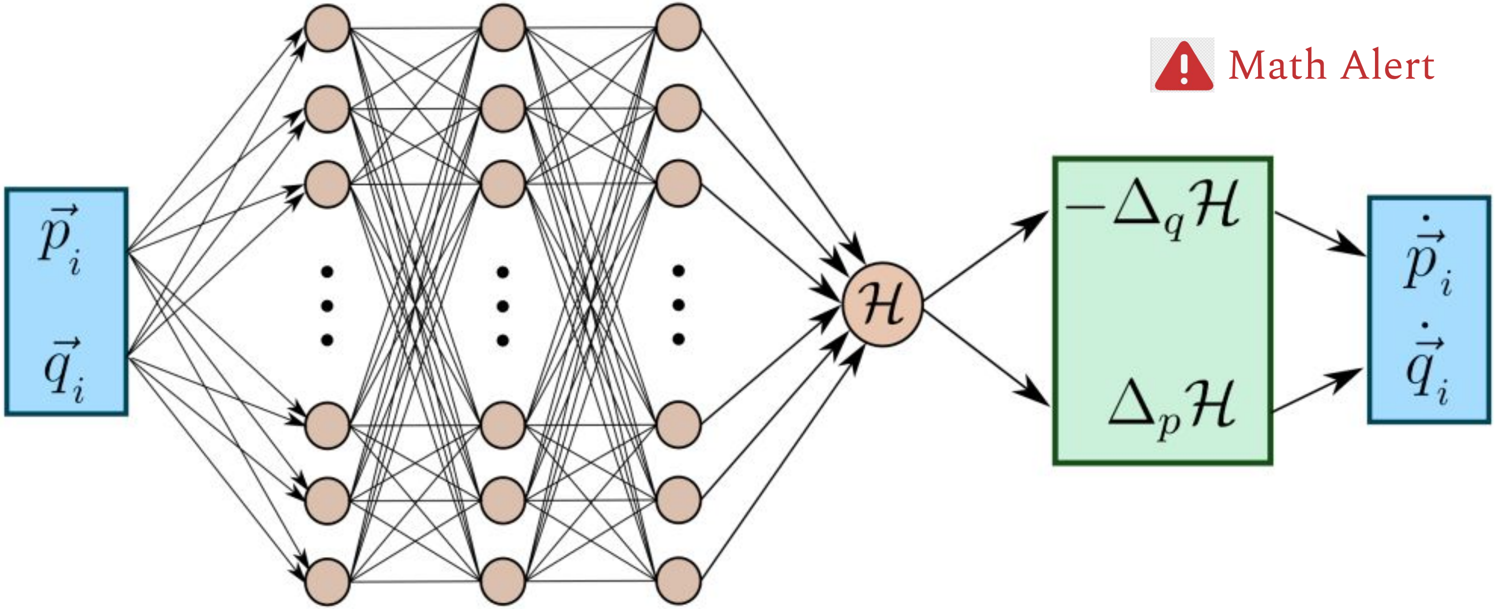
We need better NNs for complex physics problems

Chapter 3

A Generalized Framework of Neural Networks for Hamiltonian Systems

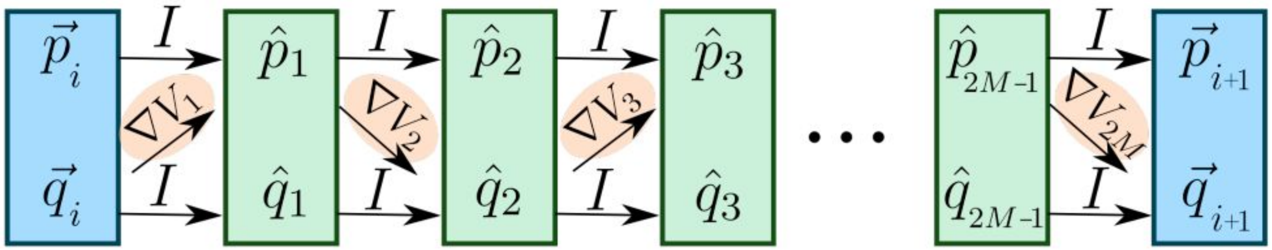
Hamiltonian Neural Network

Math Alert

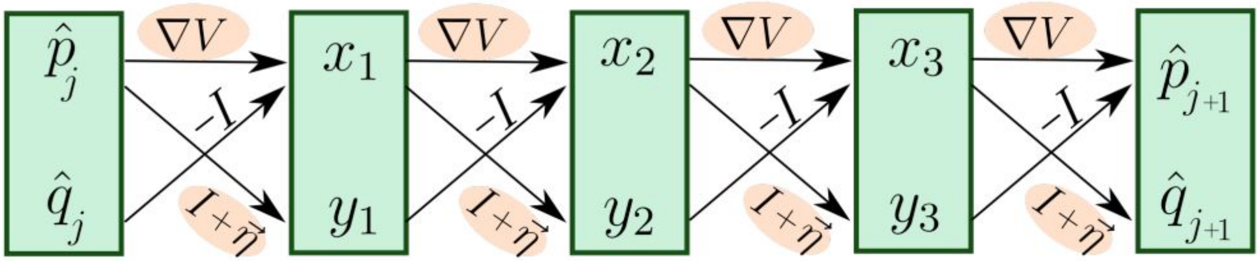


Other physics-aware Neural Networks

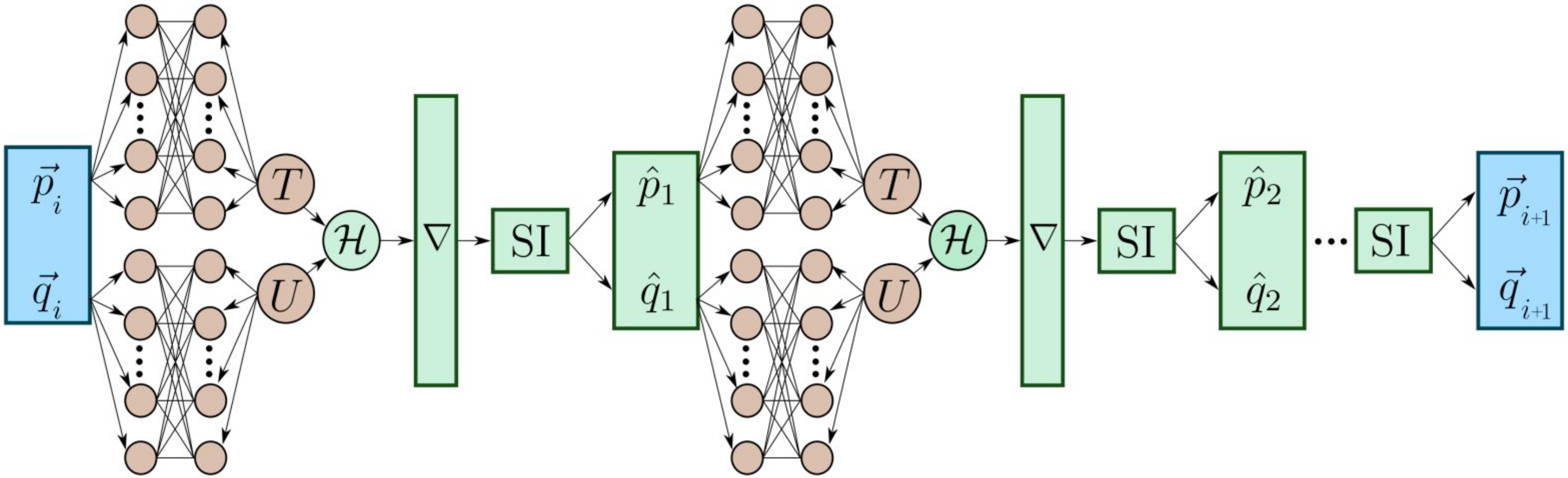
SympNets



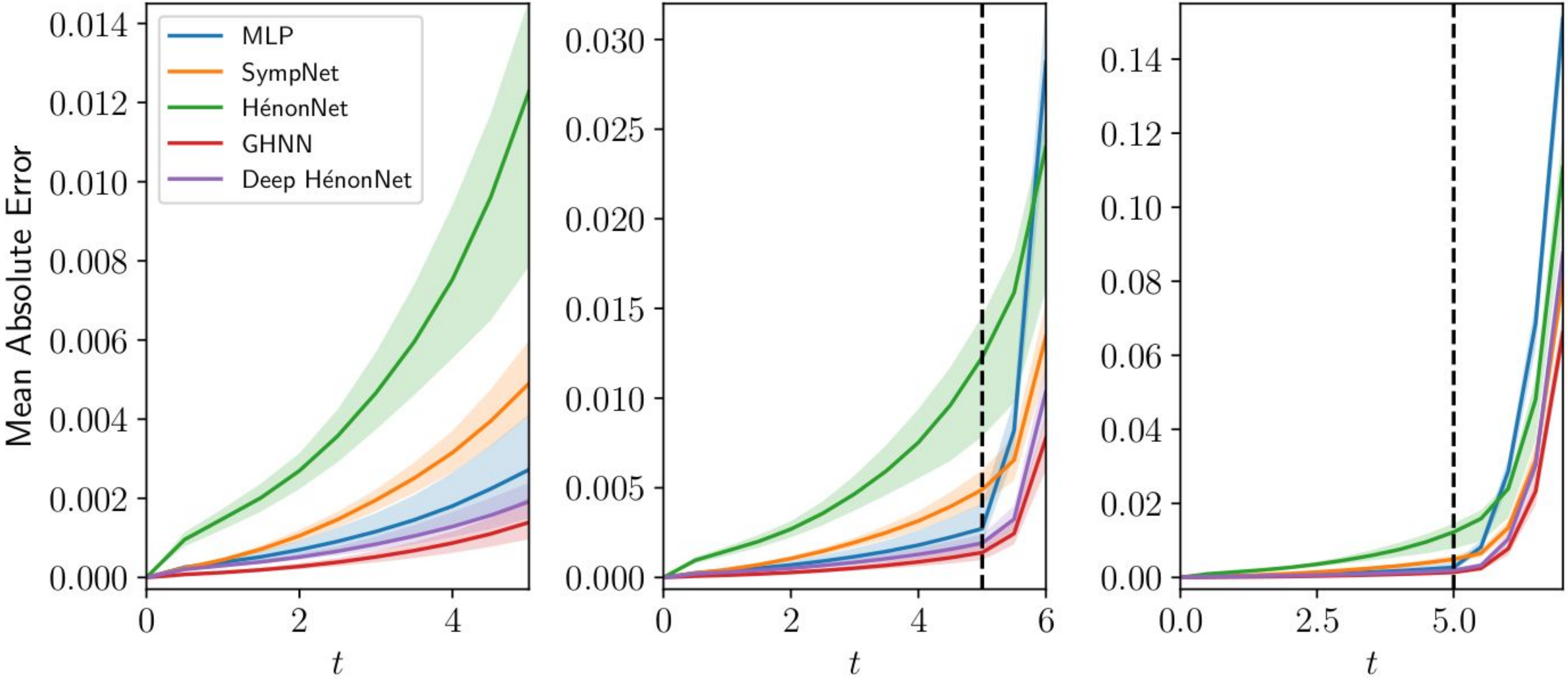
Hénon Nets



Generalized Framework



Results for the 3-body Problem



Conclusions and disclaimers

Physics structure — Better energy conservation — Less flexible

Different methods work for different problems

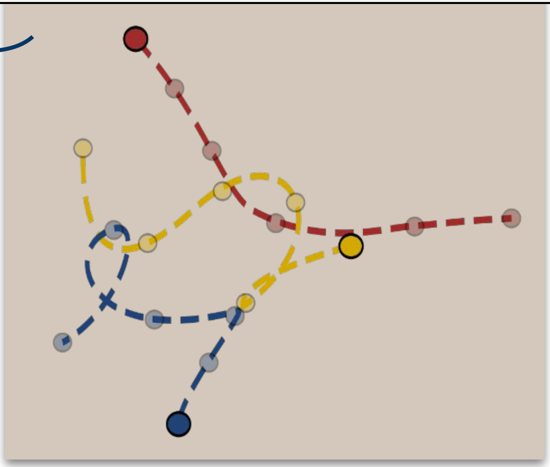
Current algorithms are not yet ready for their application on complex Astrophysics cases

Chapter 4

Reinforcement Learning for Adaptive Time-Stepping in the Chaotic Gravitational Three-Body Problem

Small Δt → Accurate - Expensive

Large Δt → Inaccurate - Cheap

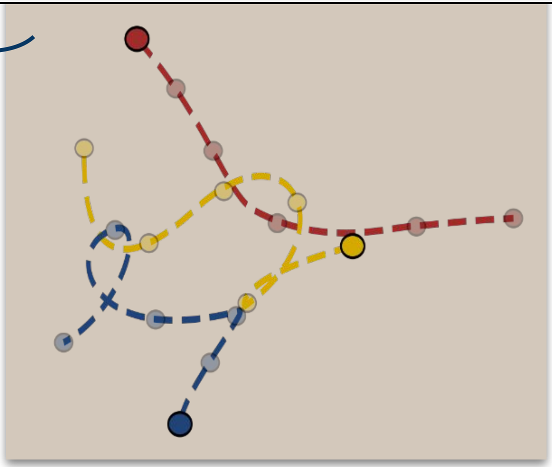


Integrator types

1. Fixed time-step integrators - *Symple*
2. Variable time-step integrators - *Huayno, Hermite, Brutus*

Small Δt → Accurate - Expensive

Large Δt → Inaccurate - Cheap

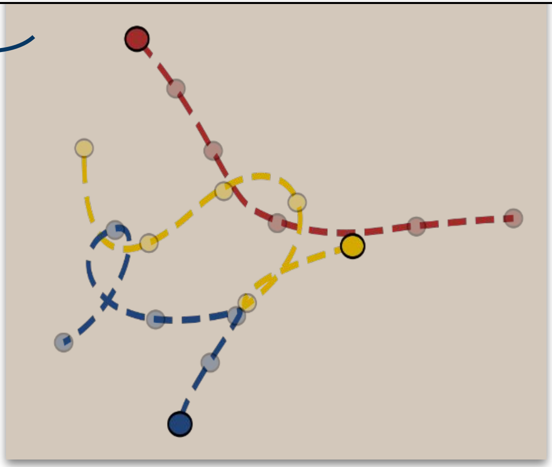


Integrator types

1. Fixed time-step integrators - *Symple*
2. Variable time-step integrators - *Huayno, Hermite, Brutus* - μ

Small Δt → Accurate - Expensive

Large Δt → Inaccurate - Cheap

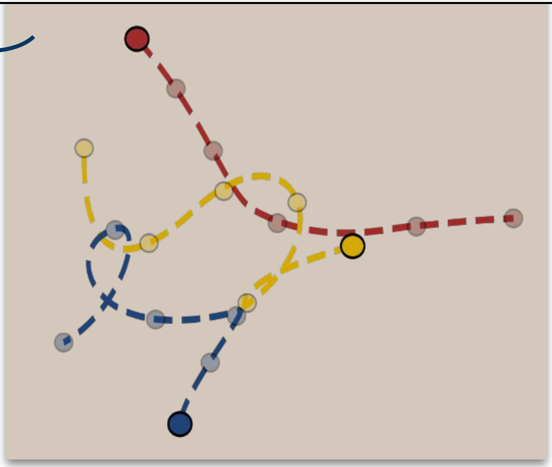


Integrator types

1. Fixed time-step integrators - *Symple*
2. Variable time-step integrators - *Huayno, Hermite, Brutus* - μ

Small Δt → Accurate - Expensive

Large Δt → Inaccurate - Cheap



Dynamic environment

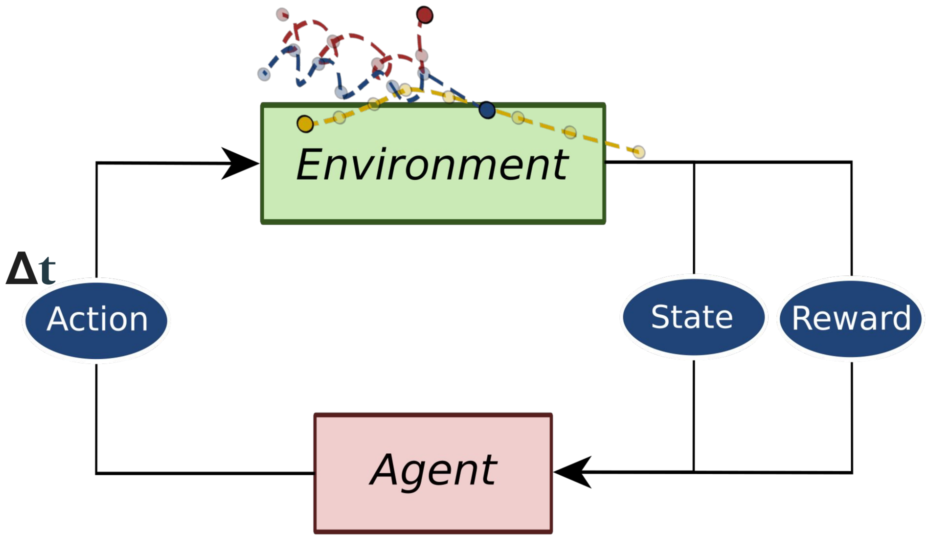
Keep it accurate

Close encounters → *Smaller Δt*

Far away particles → *Larger Δt*

Keep it efficient

Reinforcement Learning

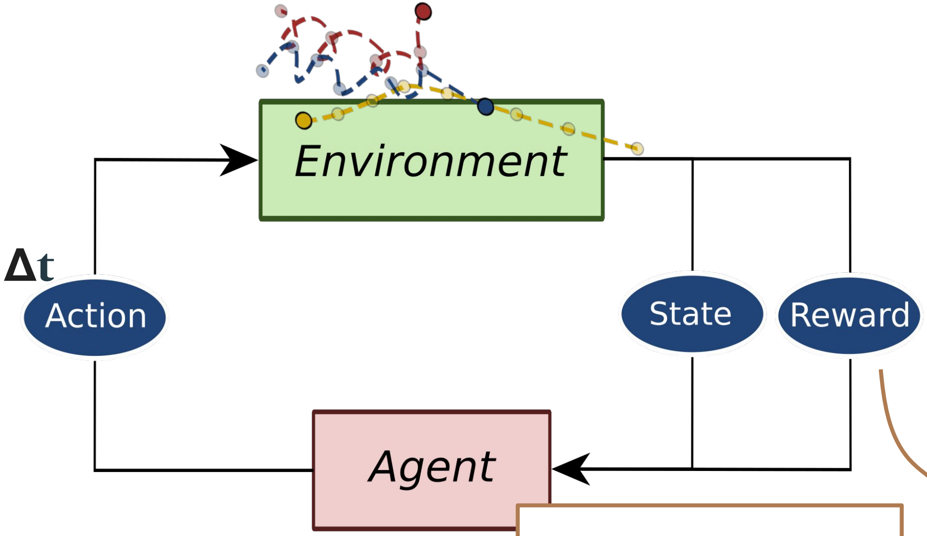


Goal: choose time-step size

Information: state of the system

Optimize: accuracy & speed

Reinforcement Learning



Goal: choose time-step size

Information: state of the system

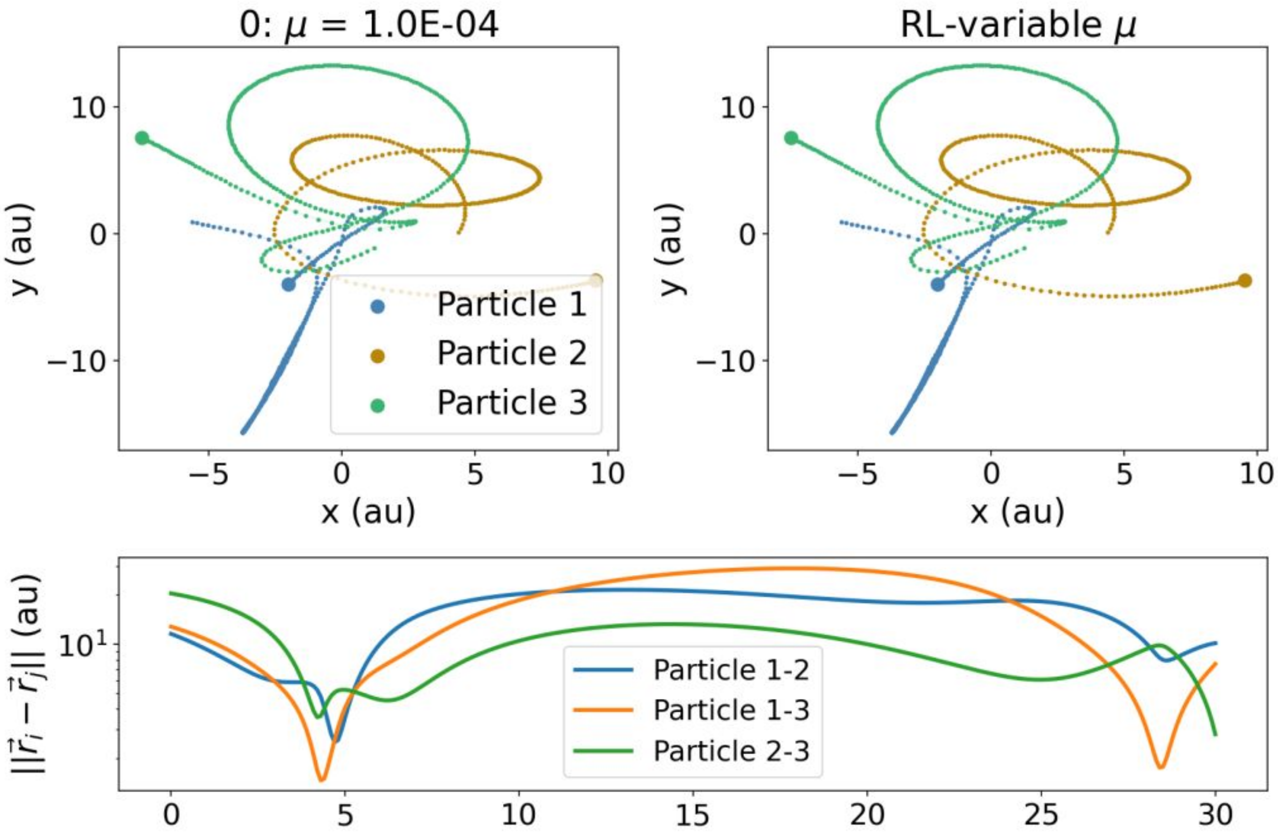
Optimize: accuracy & speed

Accuracy

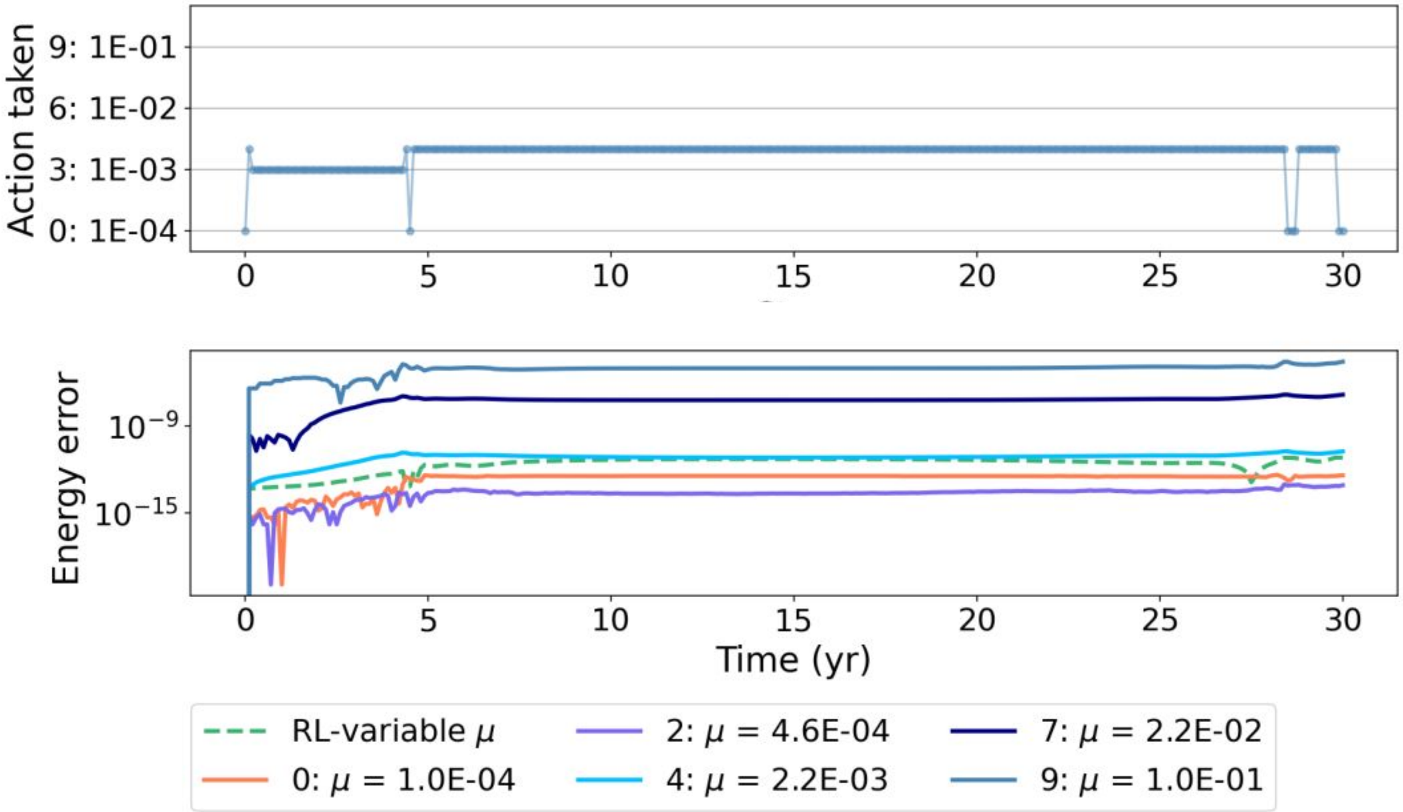
Comp. Time

$$R = -W_0 \frac{1}{\text{step}} \frac{\log_{10}\left(\left|\Delta E_i\right| / 10^{-10}\right)}{\left|\log_{10}(\Delta E_i)\right|^3} + W_1 \frac{1}{\left|\log_{10}(\mu)\right|}$$

Results

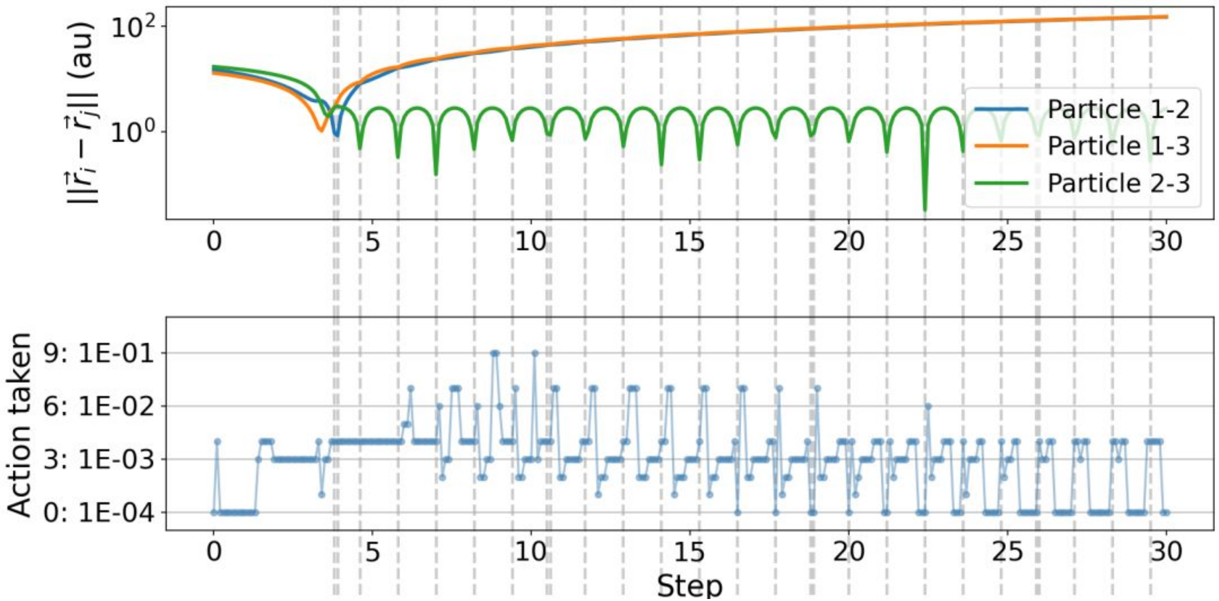

0: μ = 1.0E-04
RL-variable μ
y (au)
x (au)
Particle 1
Particle 2
Particle 3
$||\vec{r}_i - \vec{r}_j||$ (au)
Particle 1-2
Particle 1-3
Particle 2-3

Results



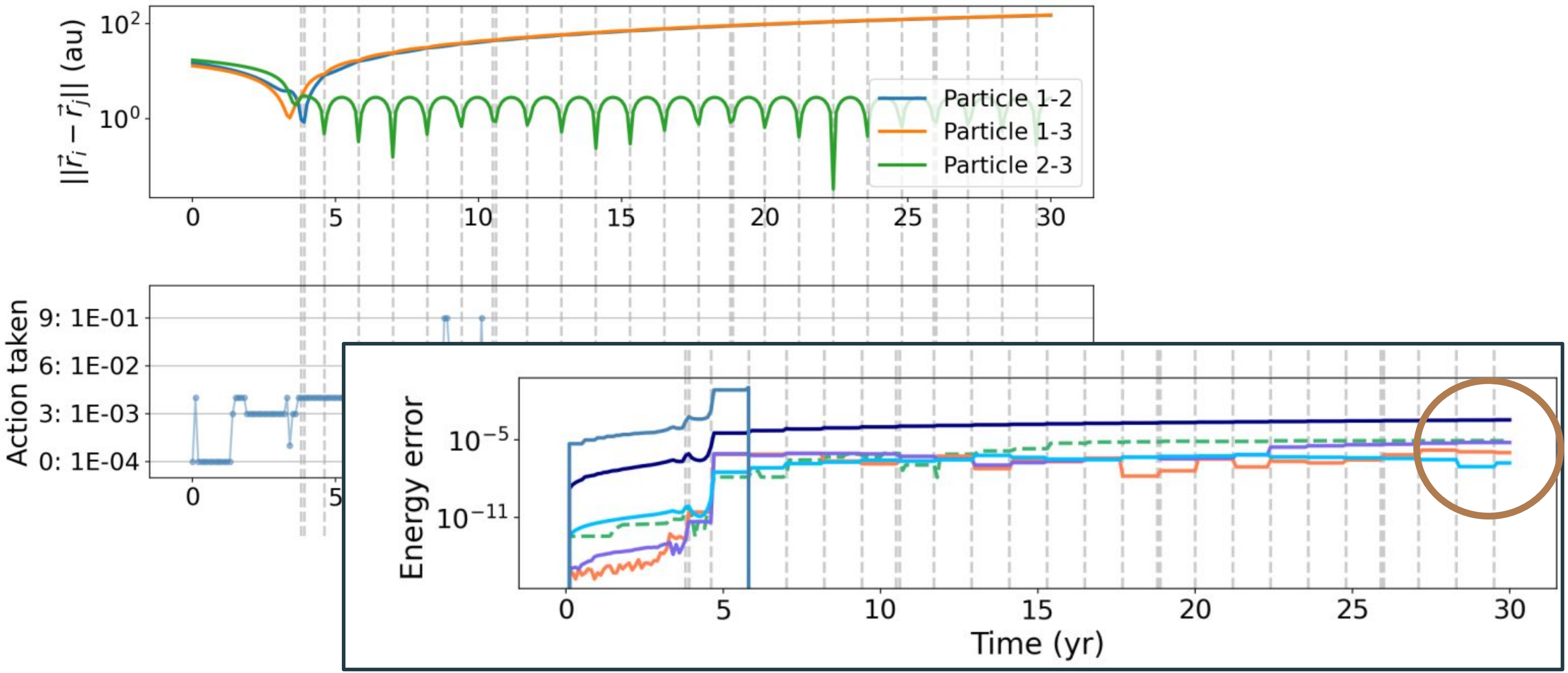
Action taken
9: 1E-01
6: 1E-02
3: 1E-03
0: 1E-04
0
5
10
15
20
25
30
Energy error
10^{-9}
10^{-15}
0
5
10
15
20
25
30
Time (yr)
RL-variable μ
0: μ = 1.0E-04
2: μ = 4.6E-04
4: μ = 2.2E-03
7: μ = 2.2E-02
9: μ = 1.0E-01

Results



$||\vec{r}_i - \vec{r}_j||$ (au)
10^2
10^0
Particle 1-2
Particle 1-3
Particle 2-3
0
5
10
15
20
25
30
Action taken
9: 1E-01
6: 1E-02
3: 1E-03
0: 1E-04
Step

Results



$||\vec{r}_i - \vec{r}_j||$ (au)
10^2
10^0
Particle 1-2
Particle 1-3
Particle 2-3
0
5
10
15
20
25
30
Action taken
9: 1E-01
6: 1E-02
3: 1E-03
0: 1E-04
Energy error
10^{-5}
10^{-11}
Time (yr)

Conclusions and disclaimers

Chapter 5

Easy to extrapolate to other problems

Good results with a preliminary training

Errors in prediction accumulate

Increasing N means retraining

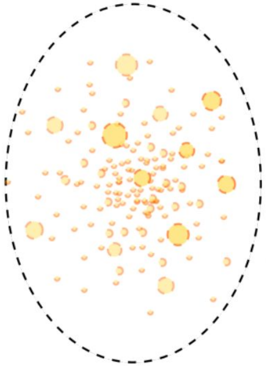
Let's try to apply it to a more complex case

Chapter 5

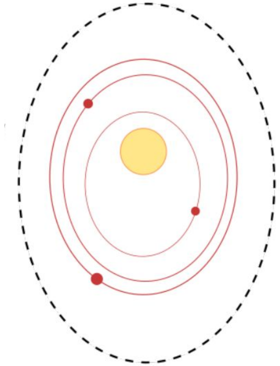
Reinforcement Learning for the Determination of the Bridge Time Step in Cluster Dynamics Simulations

Bridged Cluster with Planets

Cluster

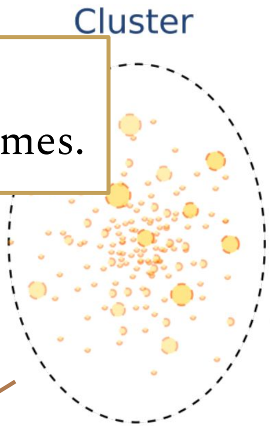


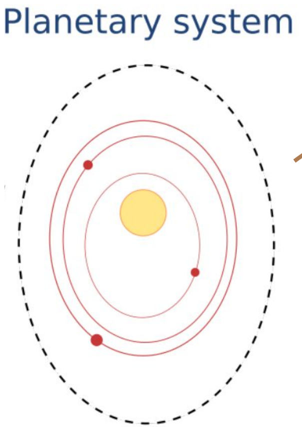
Planetary system



Bridged Cluster with Planets

- Specific Integrator.
- Large integration times.



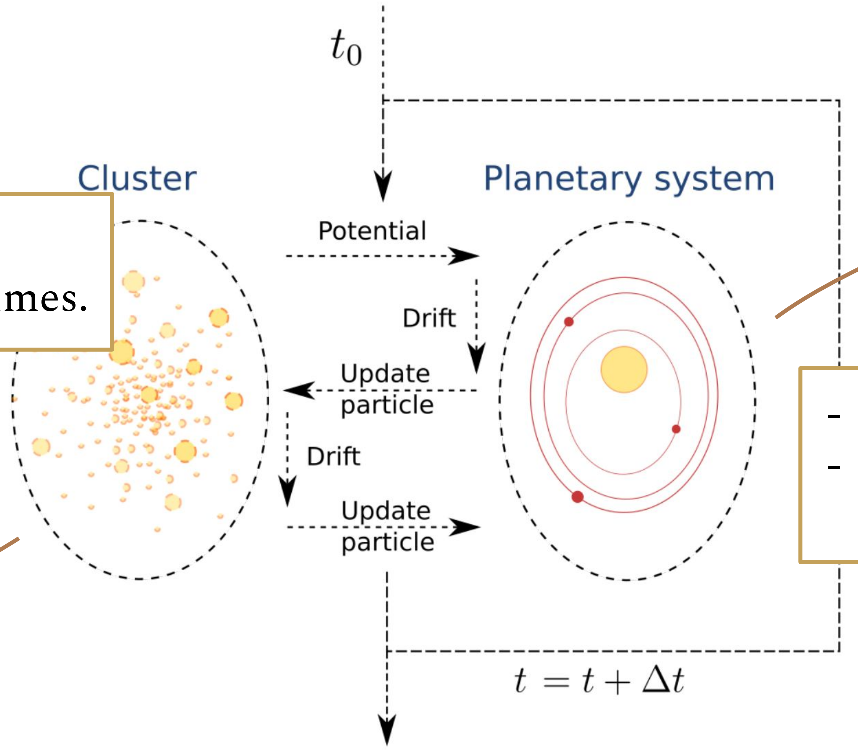


- Specific Integrator.
- Very small integration times.

Cluster Δt | | Planetary System Δt

Bridged Cluster with Planets

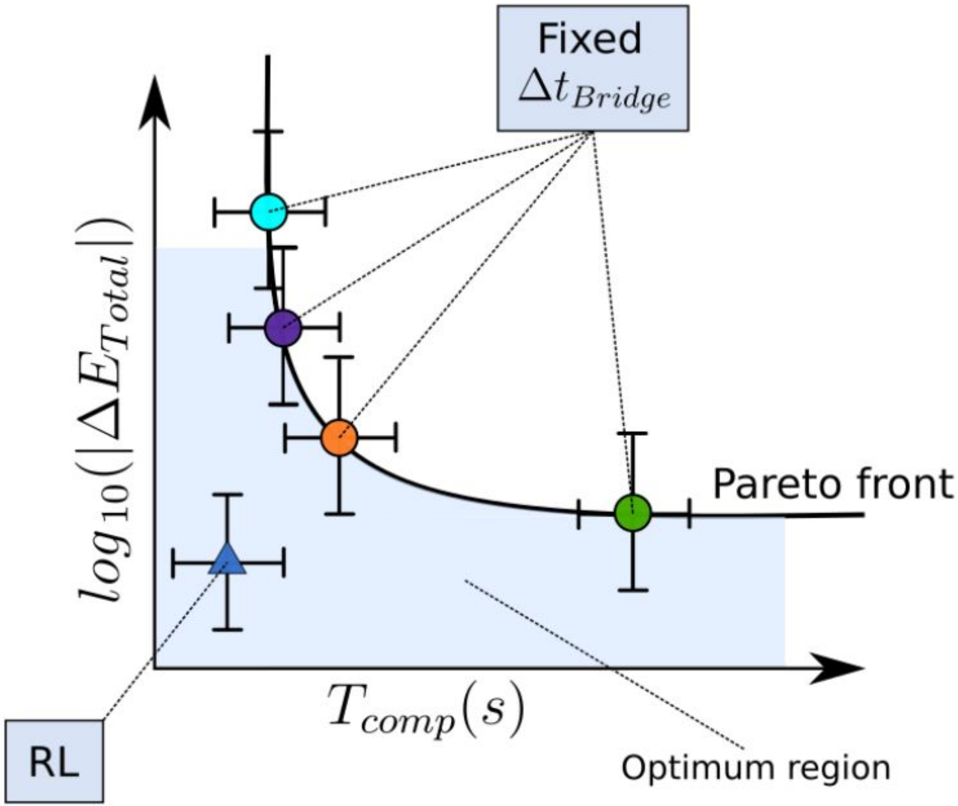
- Specific Integrator.
- Large integration times.



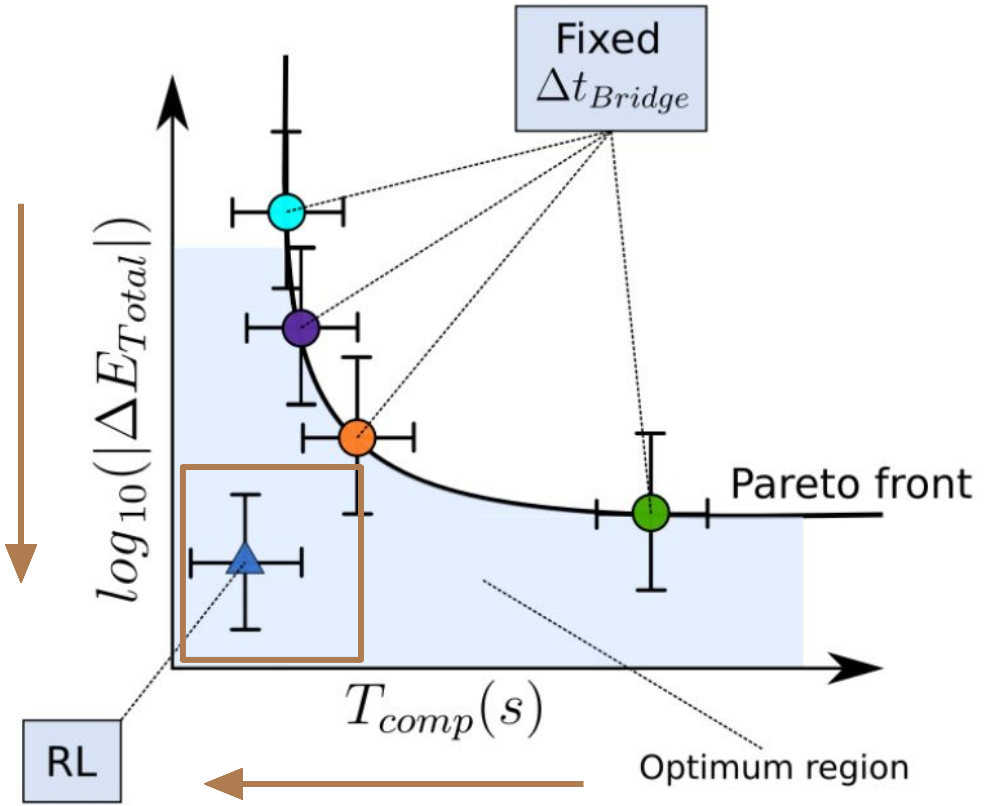
- Specific Integrator.
- Very small integration times.

Cluster Δt | **Bridge** Δt | Planetary System Δt

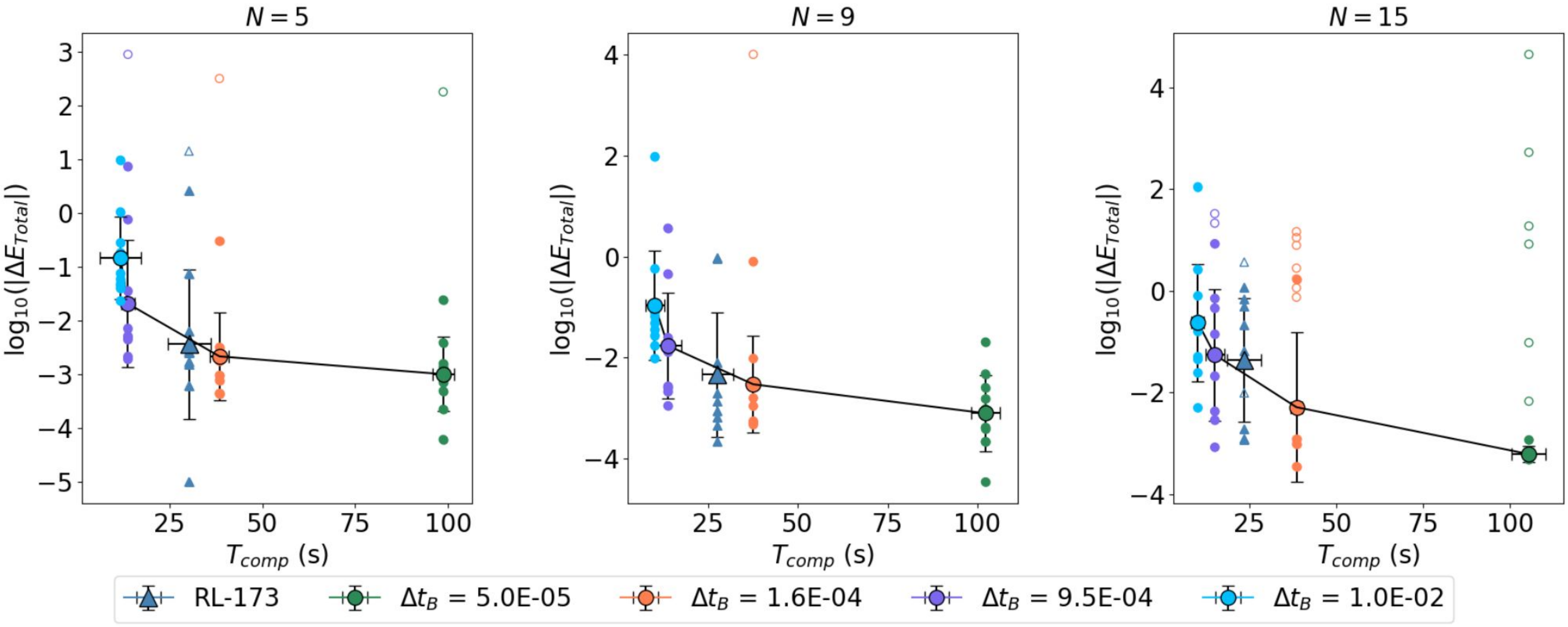
Finding an optimum solution



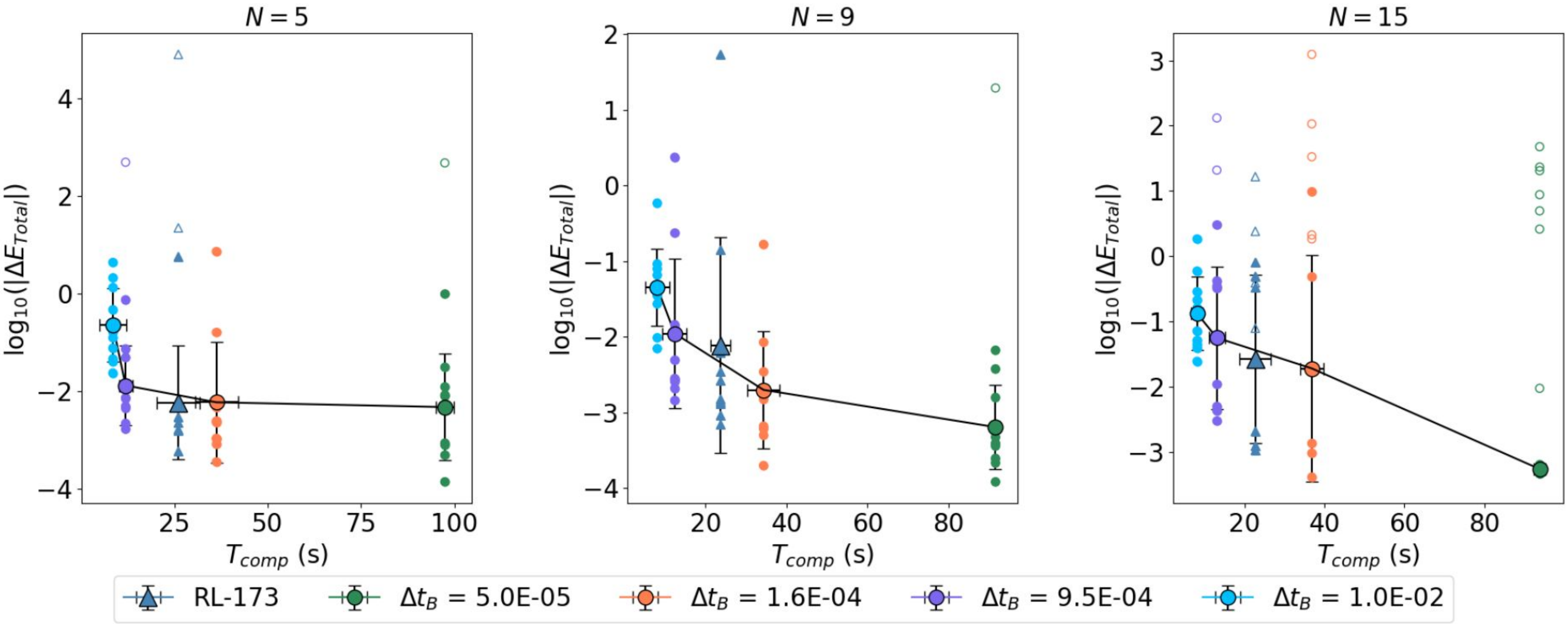
Finding an optimum solution



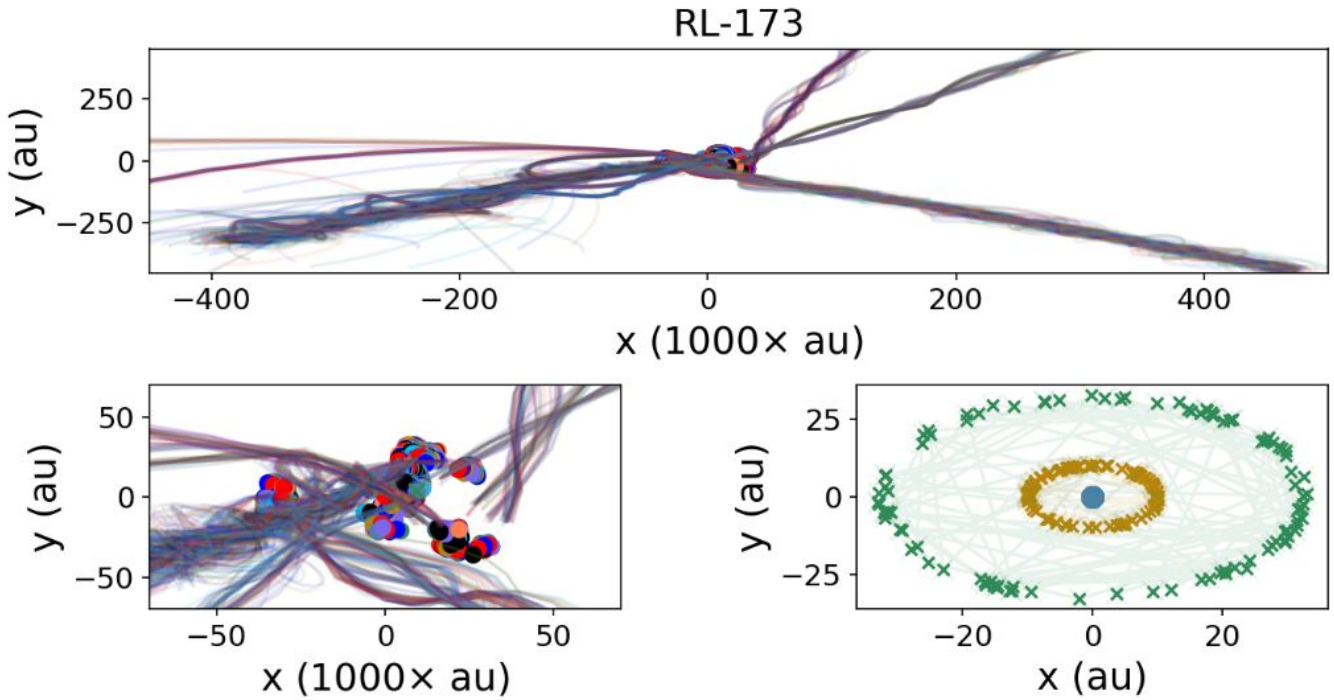
Finding an optimum solution



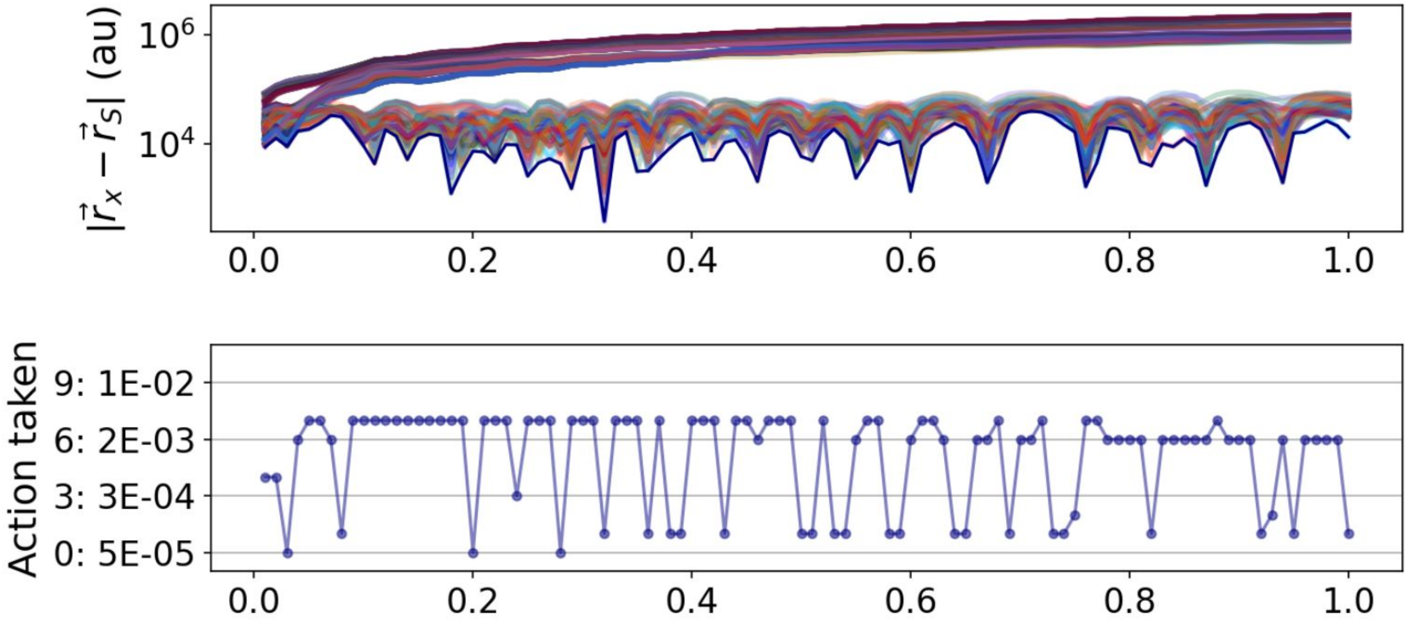
What if we use different integrators?



Application to a larger system
$N = 1{,}000$



Application to a larger system
N = 1,000



Conclusions and disclaimers

Input does not depend on N

Energy Error is unreliable for large N

Valid for many problems with small amount of re-training

Hybrid method improves robustness

Final thoughts

1. *In a simulation of the N-body problem* ***prediction errors accumulate****.*

2. *Complex network topologies help* ***energy conservation*** *- but can be hard to adapt to Astrophysics problems.*

3. *We need a better* ***state representation*** *for ML than Cartesian coordinates.*

4. *Many times,* ***integration settings*** *are left as default without studying the effects of this choice. It is better to* ***automatize*** *in a smart way.*

Thank you for your attention