maxwellcai.com

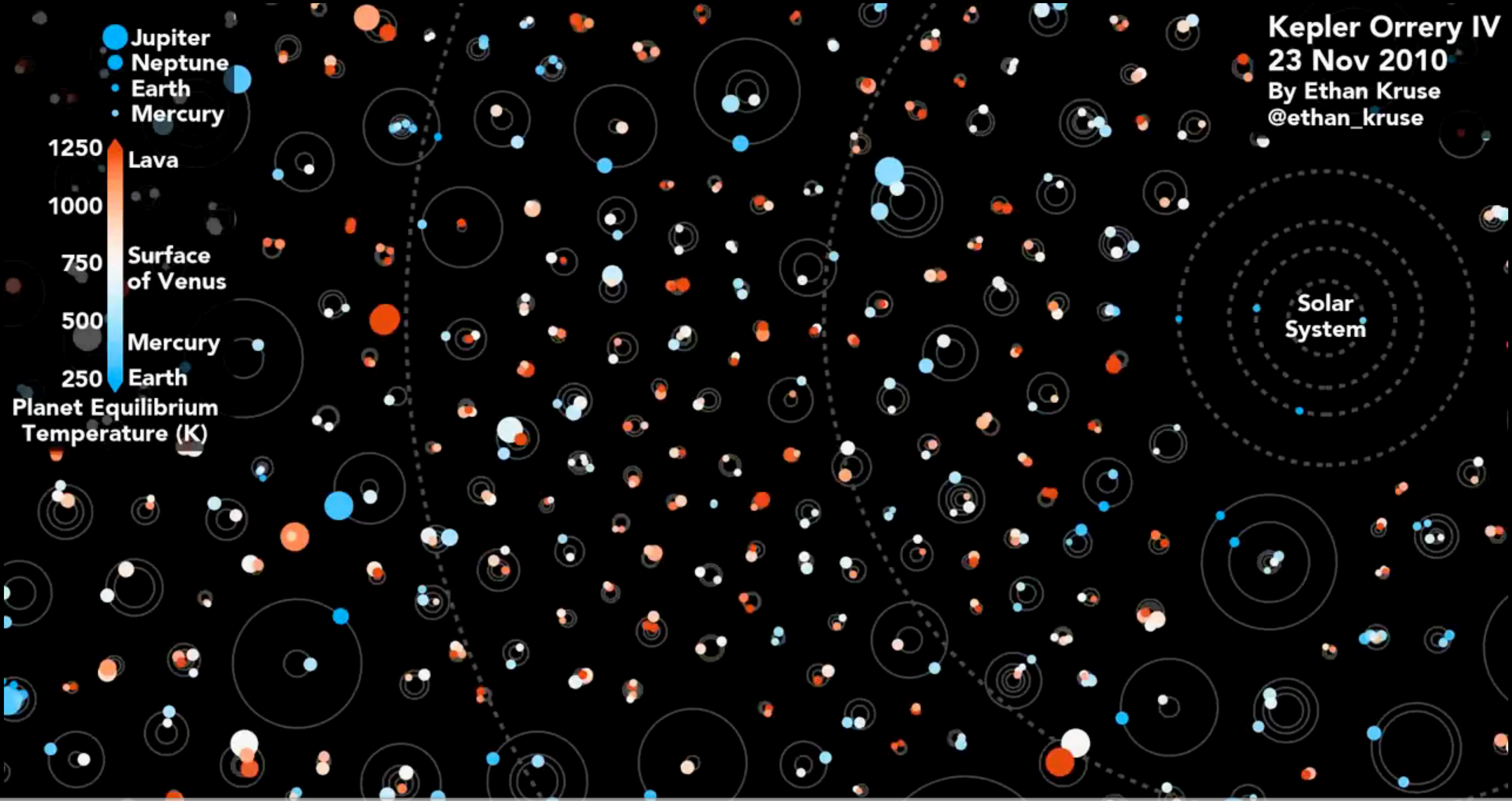
Image credit: NASA/JPL

Accelerating the simulations of nonlinear dynamical systems with deep learning

Maxwell Cai (Leiden U / SURF)
Simon Portages Zwart (Leiden U)
Damian Podareanu (SURF)
Valeriu Codreanu (SURF)
Caspar van Leewuen (SURF)


Mercury
Venus
Earth
Mars
Jupiter
Saturn
Uranus
Neptune
Source: Wikipedia

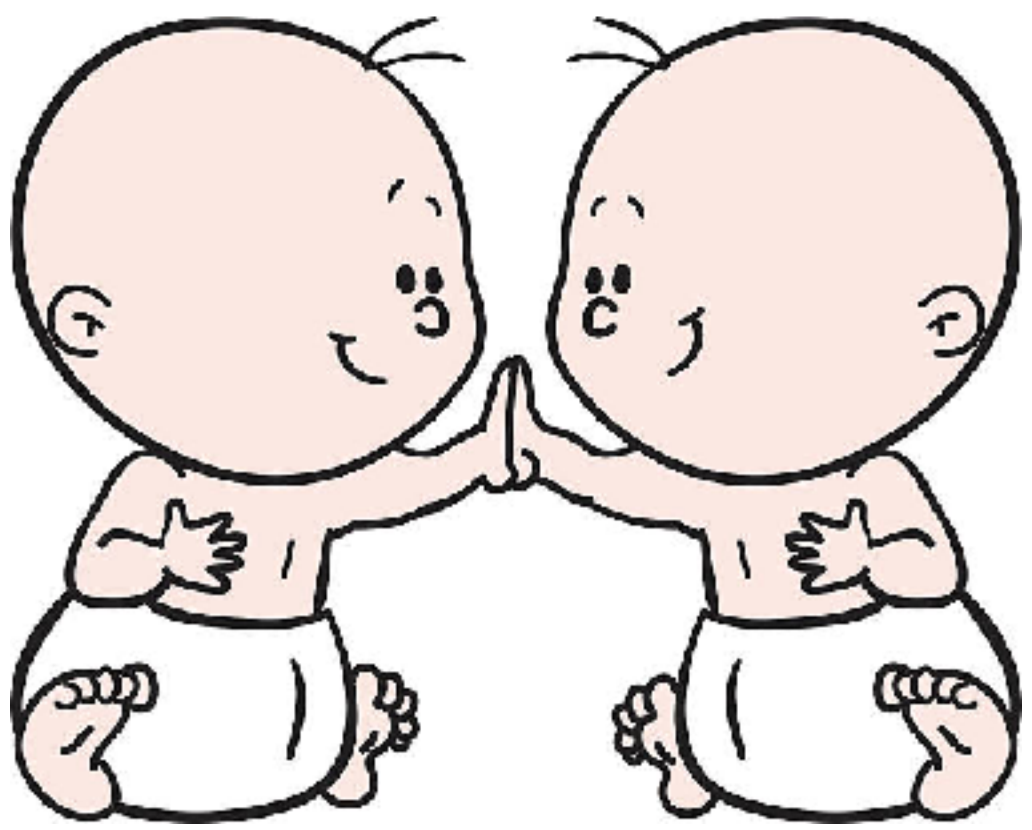
Credit: NASA



4,082 planets **3,046** planetary systems **660** multiple planetary systems (15 June 2019)


Credit: NASA

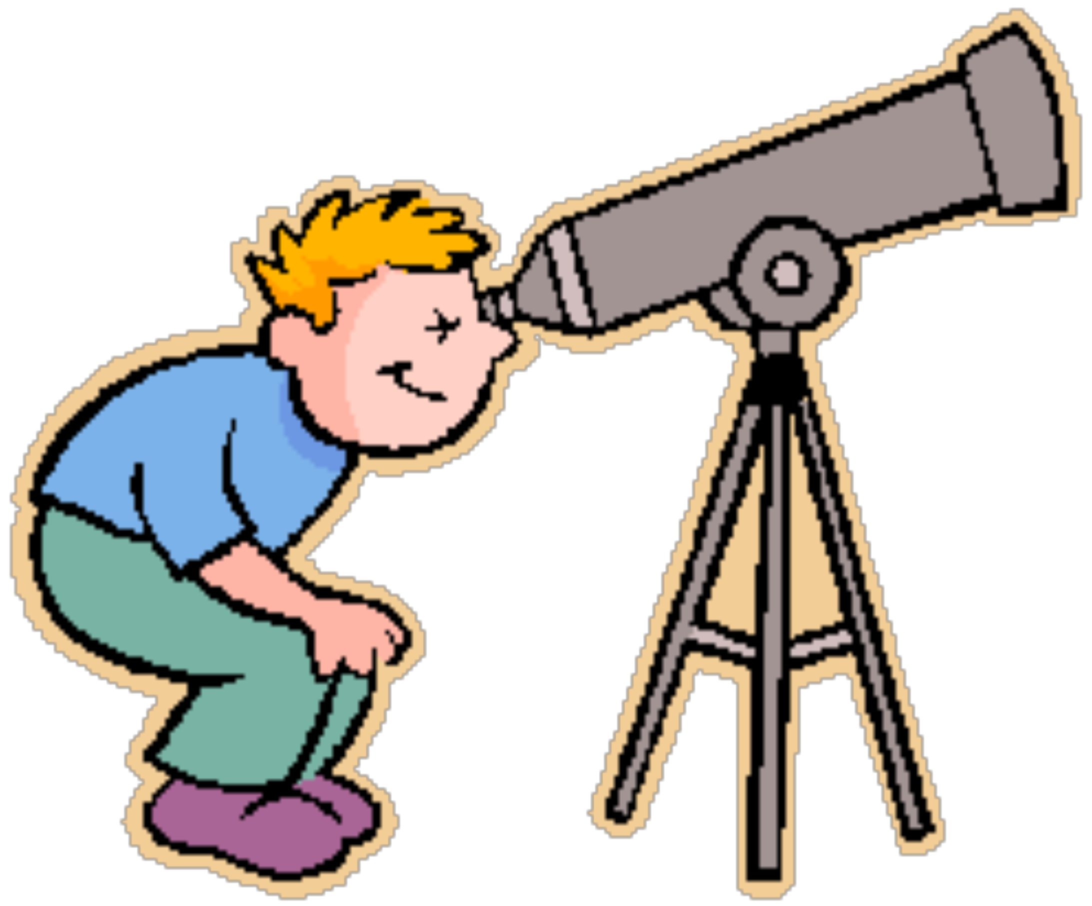
METHODOLOGY



Identical twins

Different education

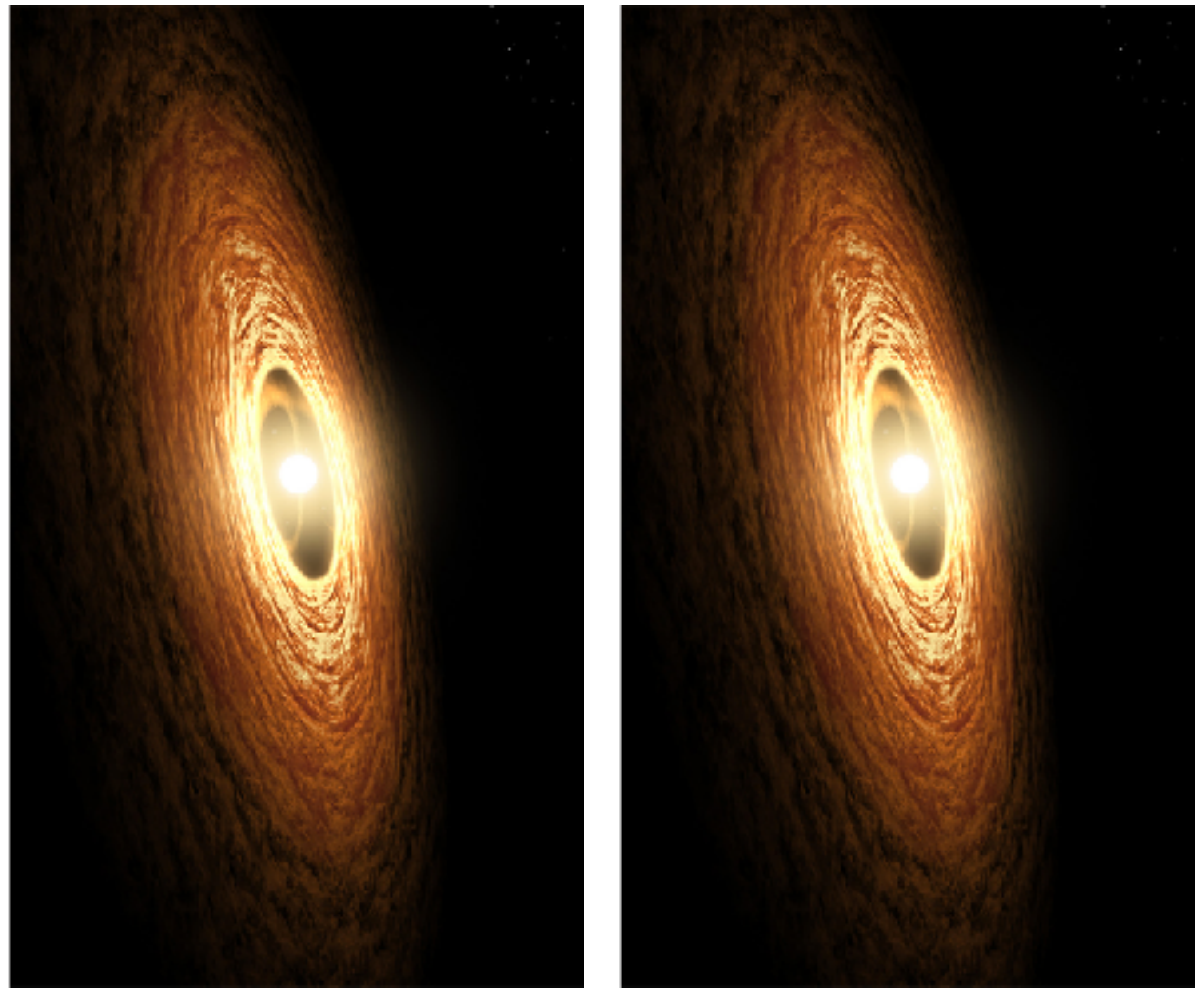
Different environments



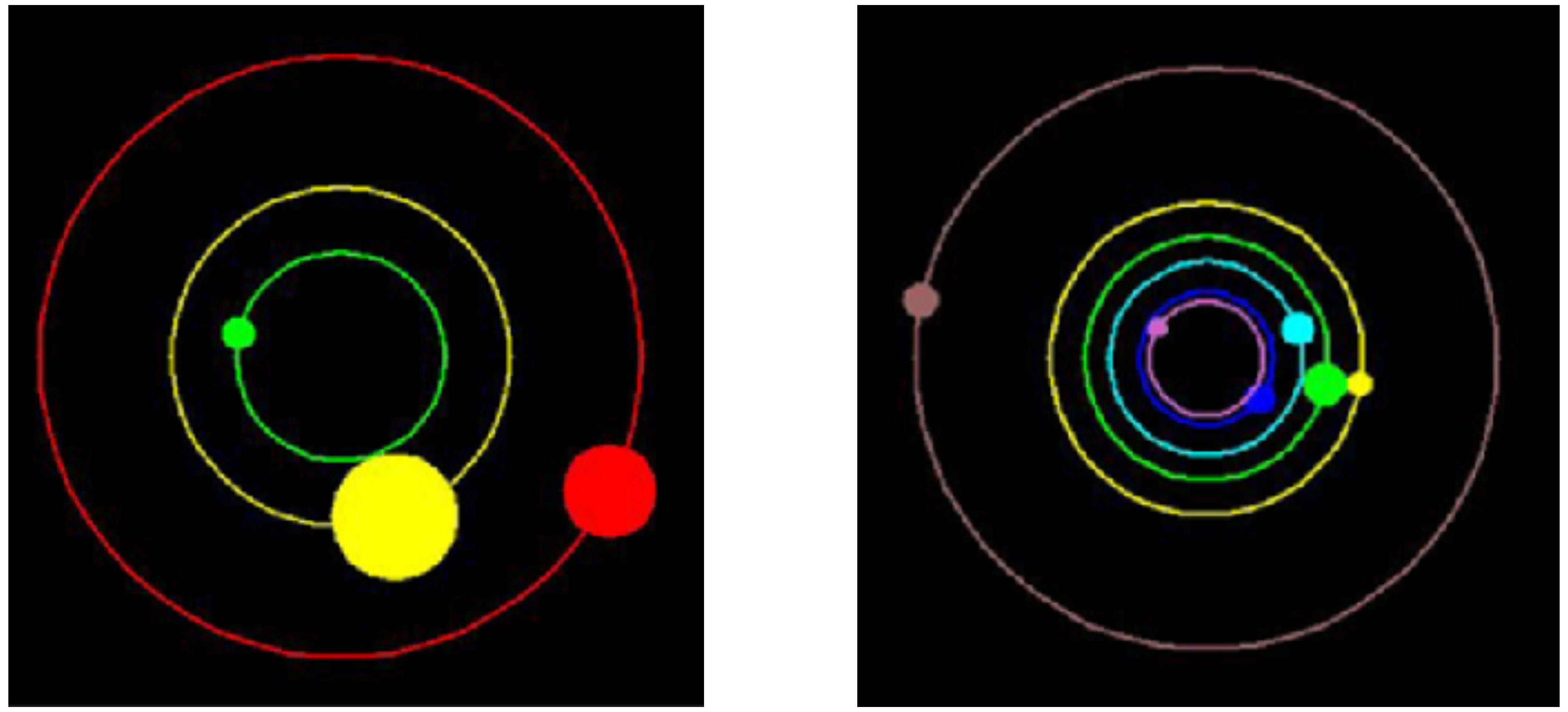
Astronomer

Musician

Less Diverse **TIME** ***More Diverse***



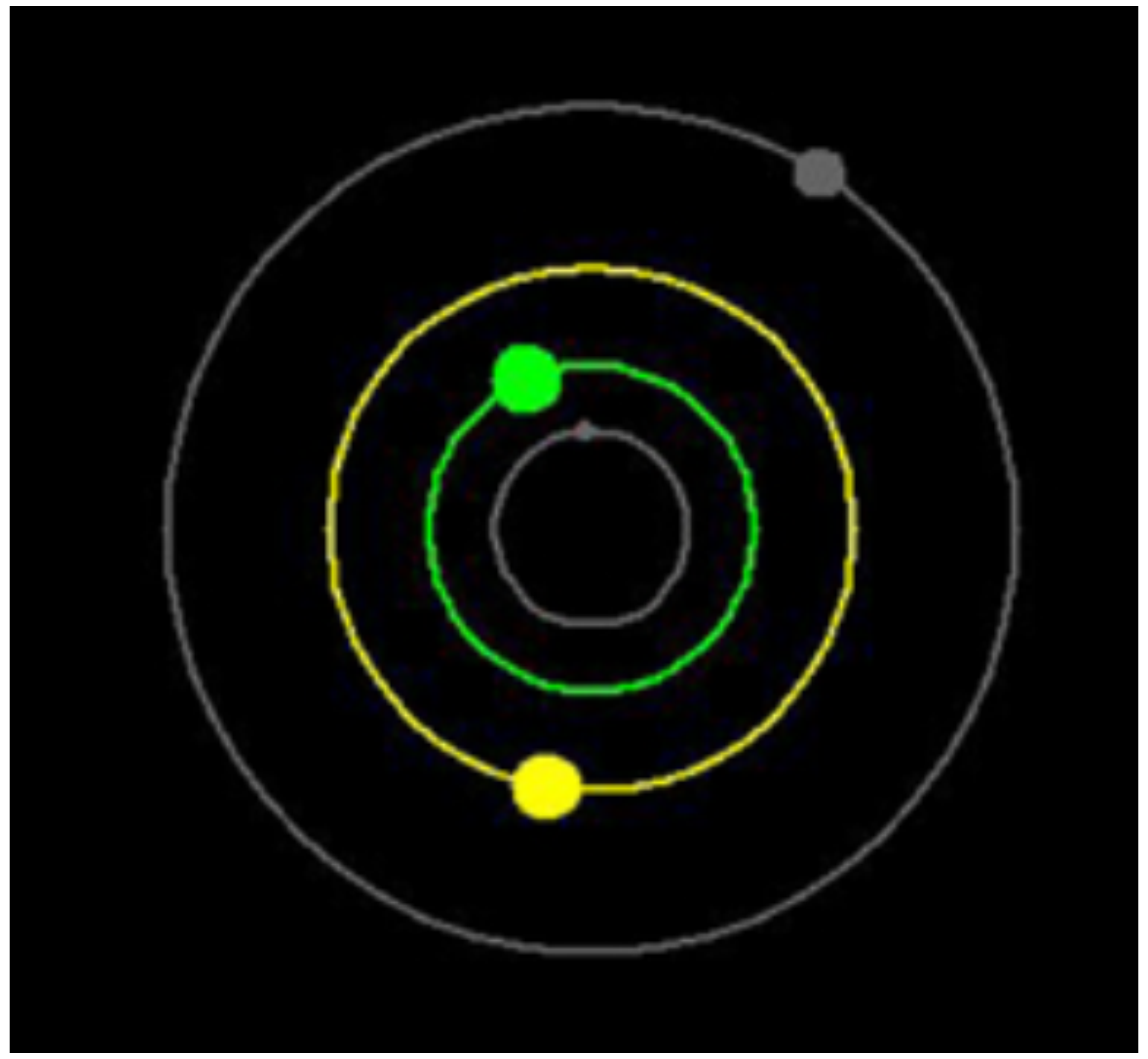
Different environments



Different orbital architectures

Multi-scale Modeling

Solar System



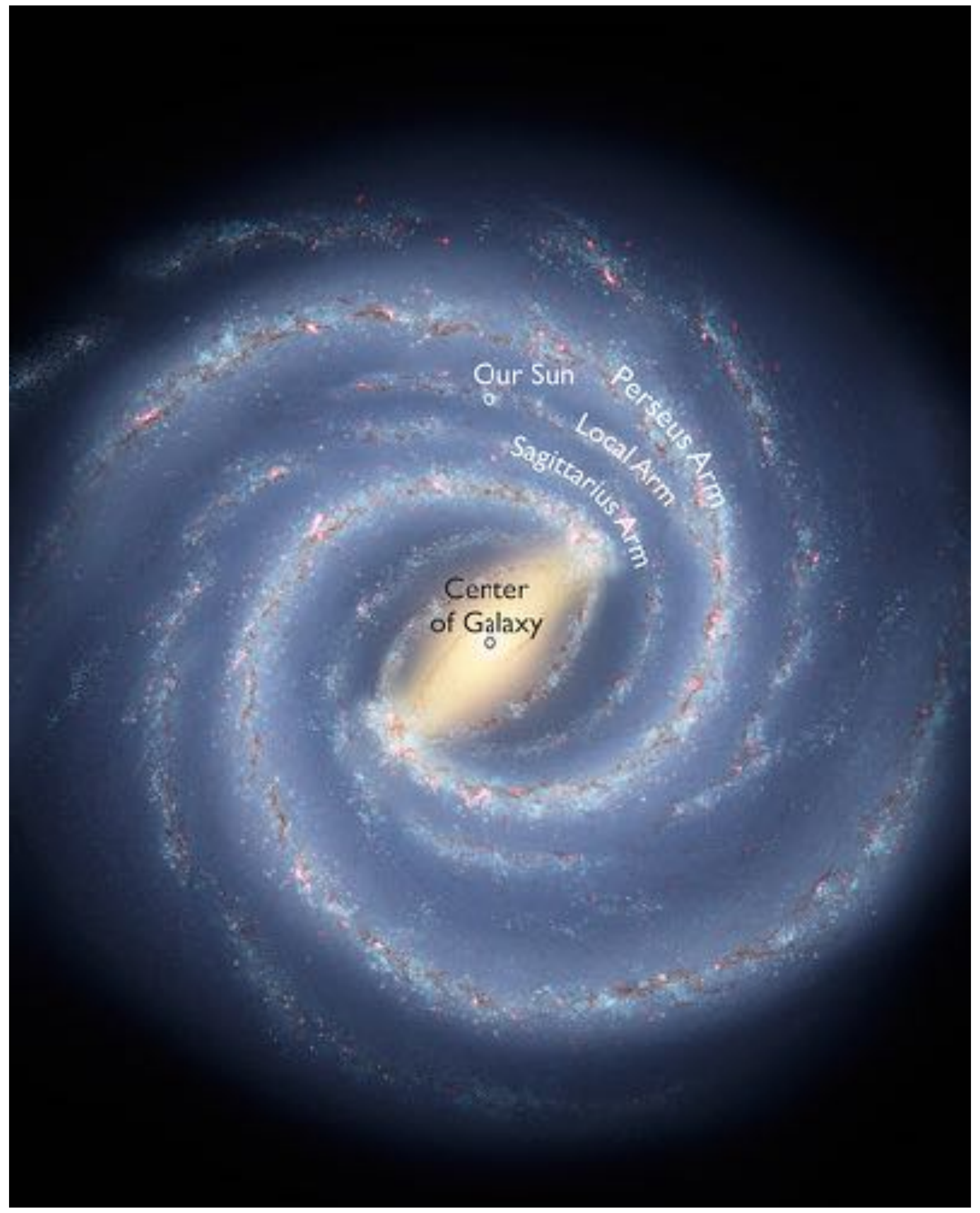
1-2 stars
A few planets
100 AU

Star cluster

1000 - 1000000 stars

100,000,000 AU

Milky Way



200,000,000,000 stars

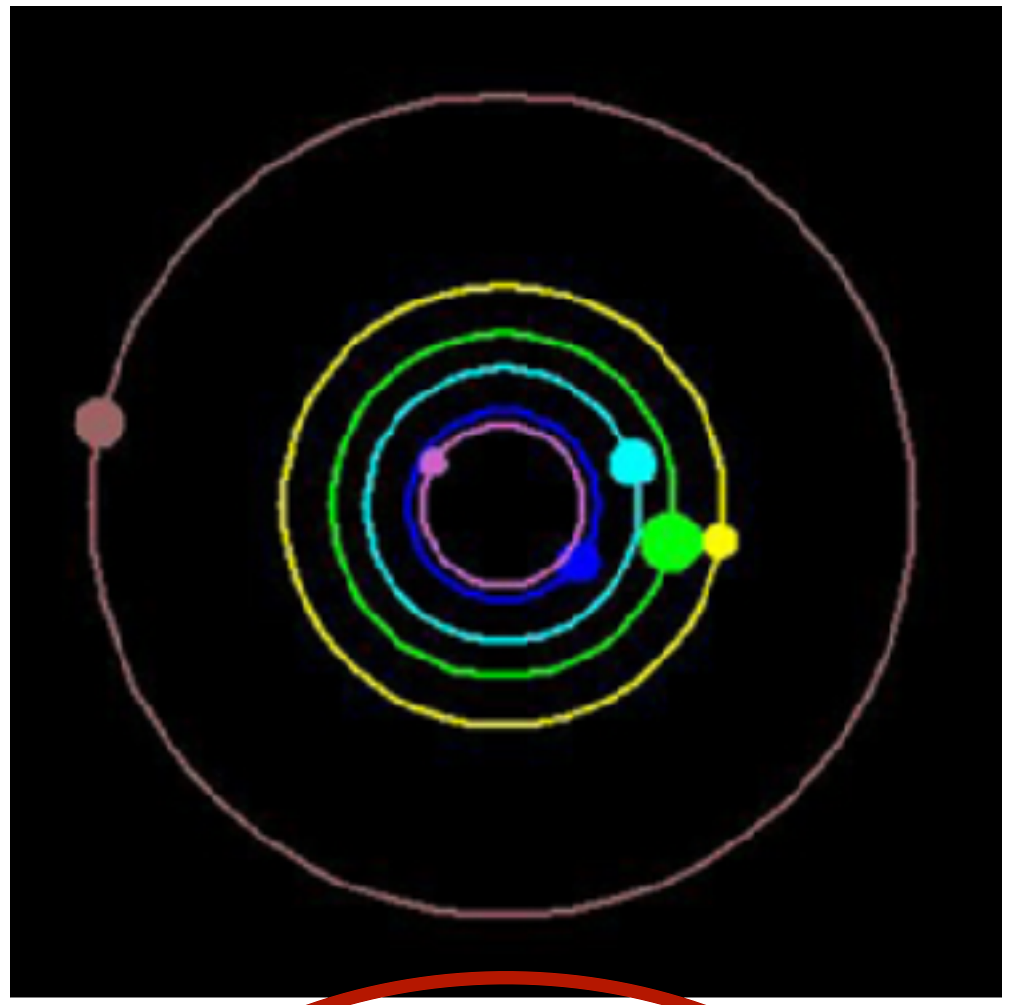
100,000,000,000,000 AU

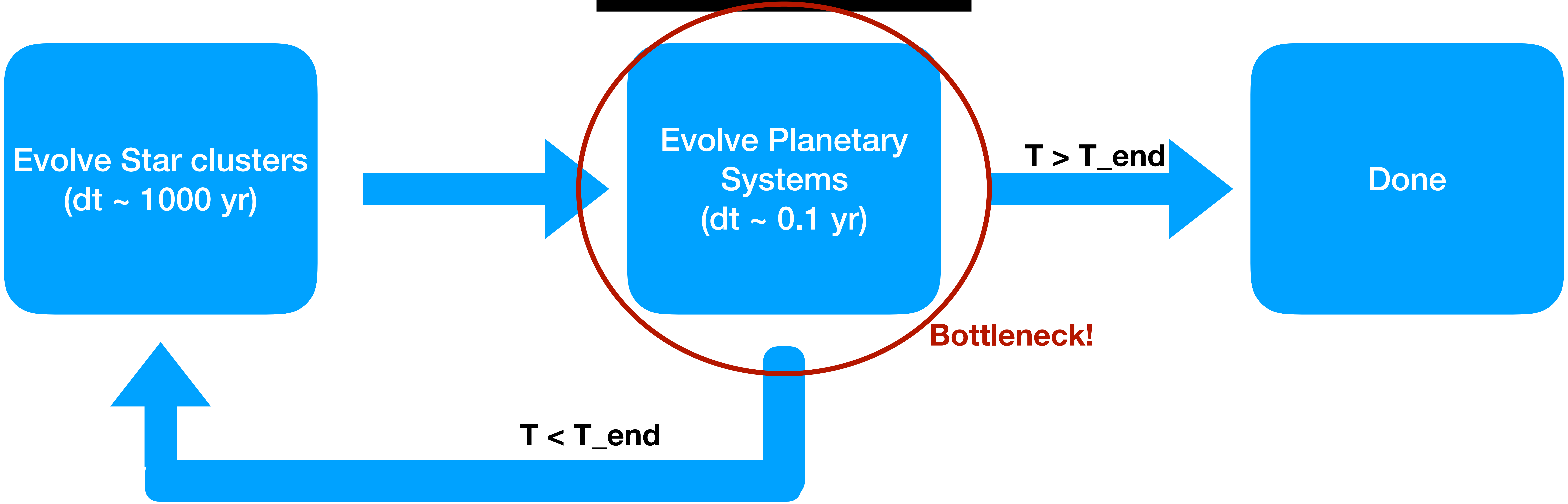
1 AU = distance from Sun to Earth = 150,000,000 km

149 975 Light Years

The Milky Way,
the galaxy in which our sun goes round every 200 million years.

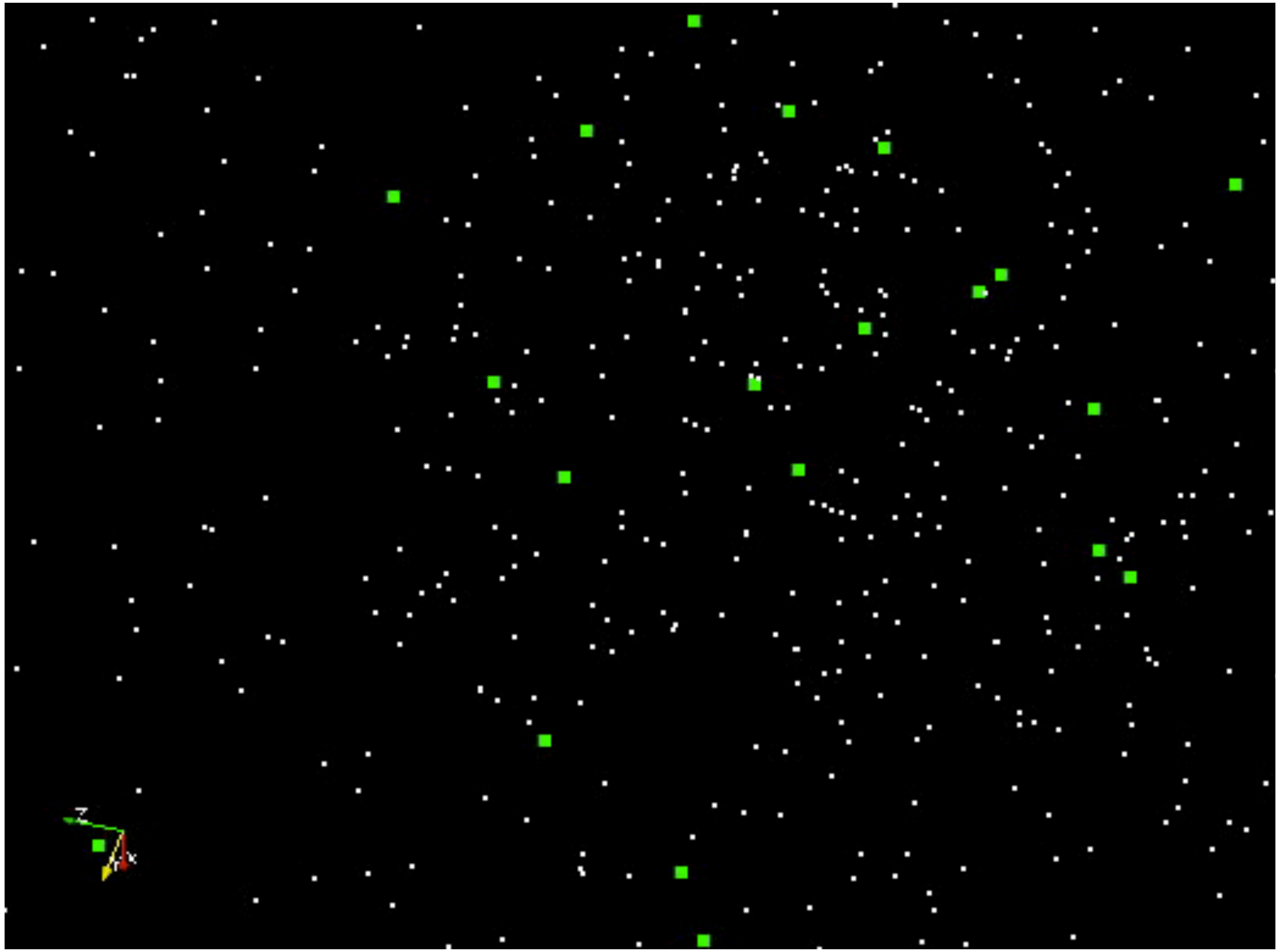
Multi-scale Modeling





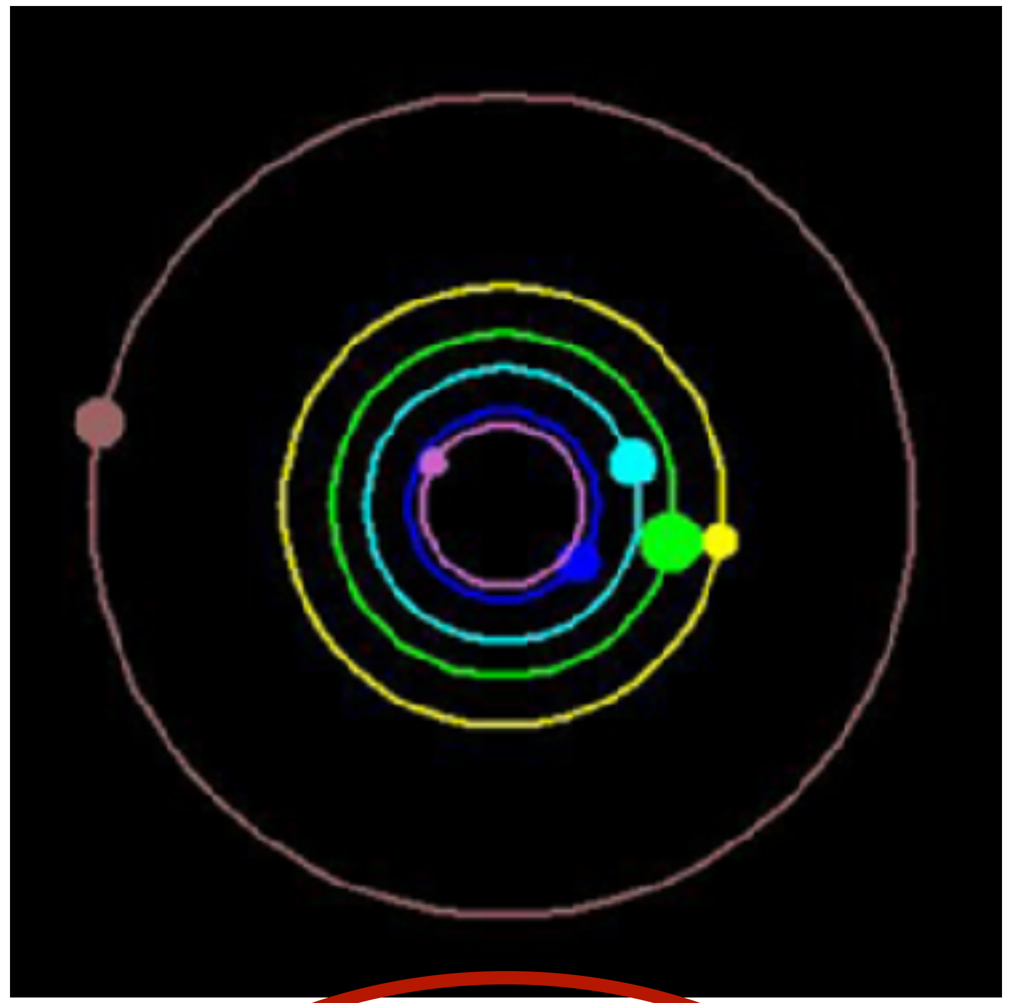

Credit: University of Bristol

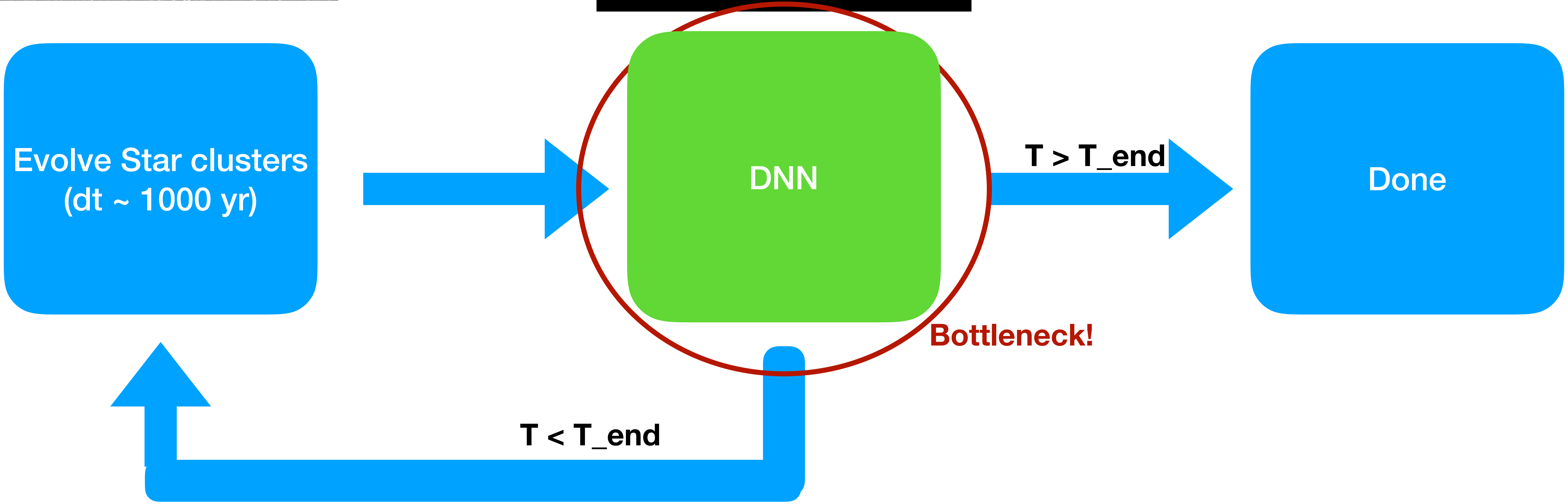
Coevolution of Planetary Systems and the Host Cluster



Cai et al. 2015 (ApJS, 219, 31), Cai et al. 2017 (MNRAS, 470, 4337), Cai et al. 2018 (MNRAS, 474, 5114), Cai et al. 2019 (arXiv: 1903.02316)

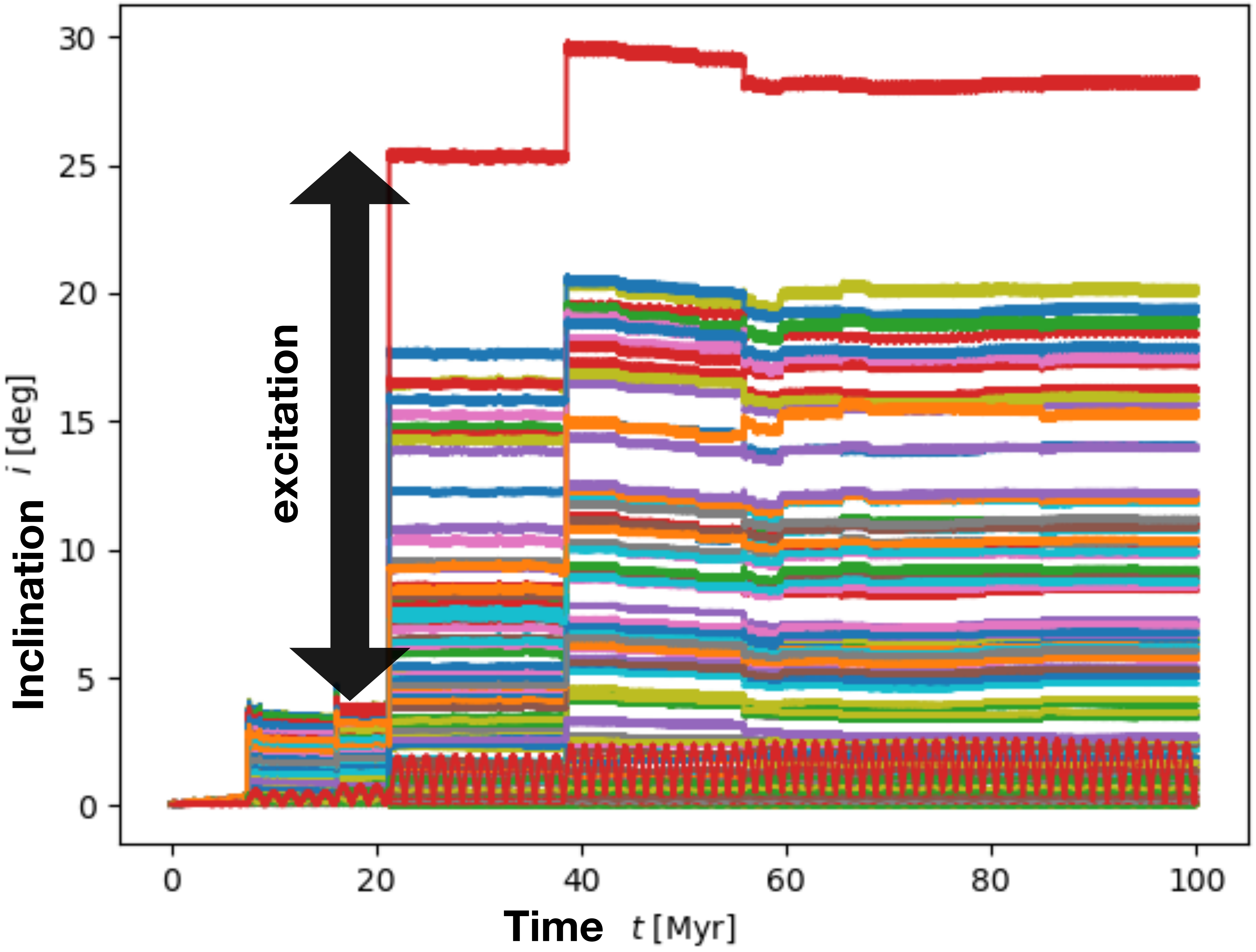
Multi-scale Modeling





Can a DNN learn to predict orbits?

Cai et al. in prep.



Challenges

- Predict on extremely long timescales
- The systems exhibit chaotic behaviors
- High dynamic range
- Huge parameter space
- Imbalance training samples — interesting events are rare

Can a DNN learn to predict orbits?

Cai et al. in prep.

Predict individual systems accurately

Very challenging on long timescales

Predict overall statistics accurately

Possible, but simple ML might be enough

Predict both individual systems and overall statistics accurately

Very challenging on long timescales

Challenges

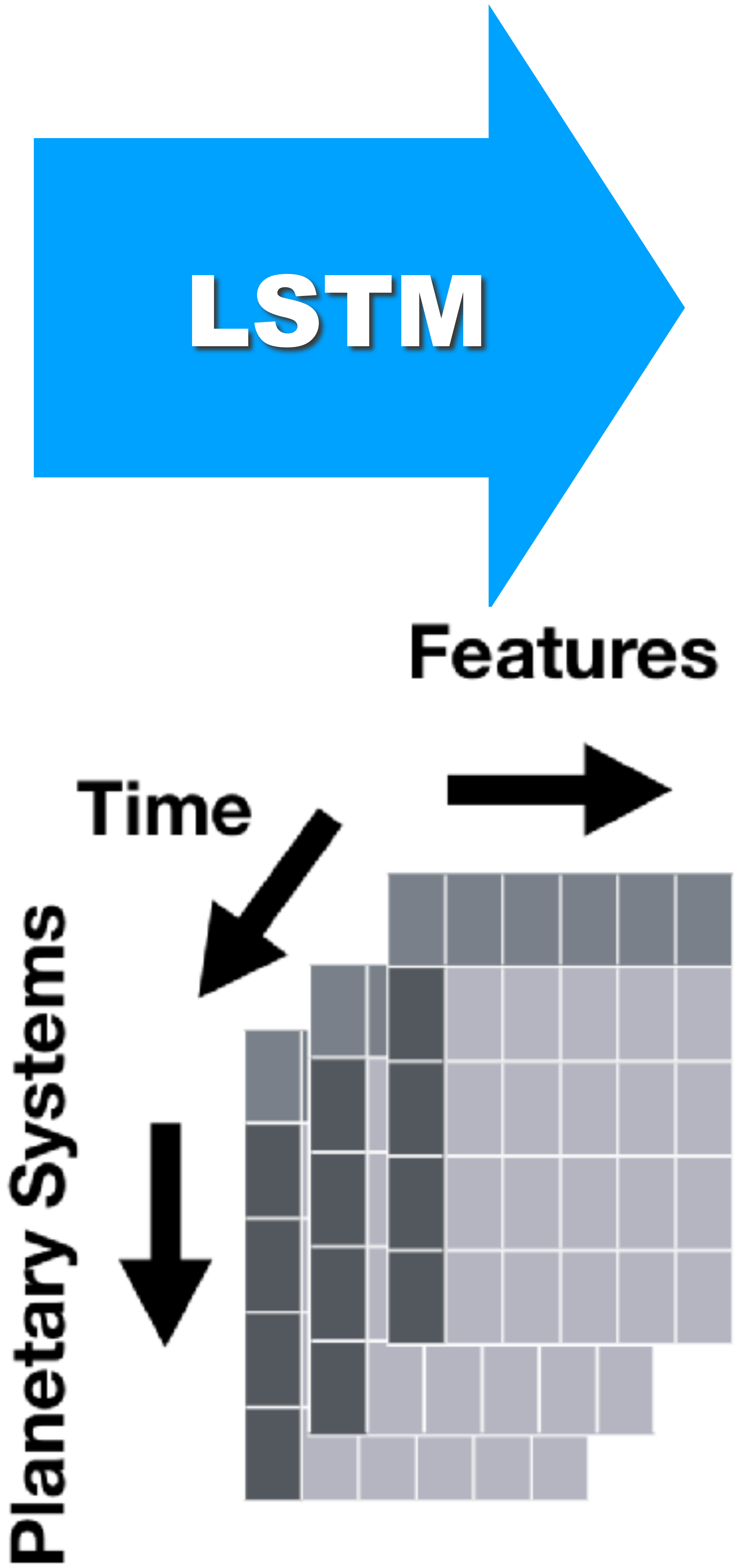
- Predict on extremely long timescales
- The systems exhibit chaotic behaviors
- High dynamic range
- Huge parameter space
- Imbalance training samples — interesting events are rare

Multivariate Time Series Prediction

LSTM: Long Short-term Memory (Hochreiter & Schmidhuber 1997)

Multiple Features

- Eccentricities
- Inclinations
- Semi-major axis
- Mass of perturber
- distance of the perturber
- velocity of the perturber
- position of the perturber
- ....

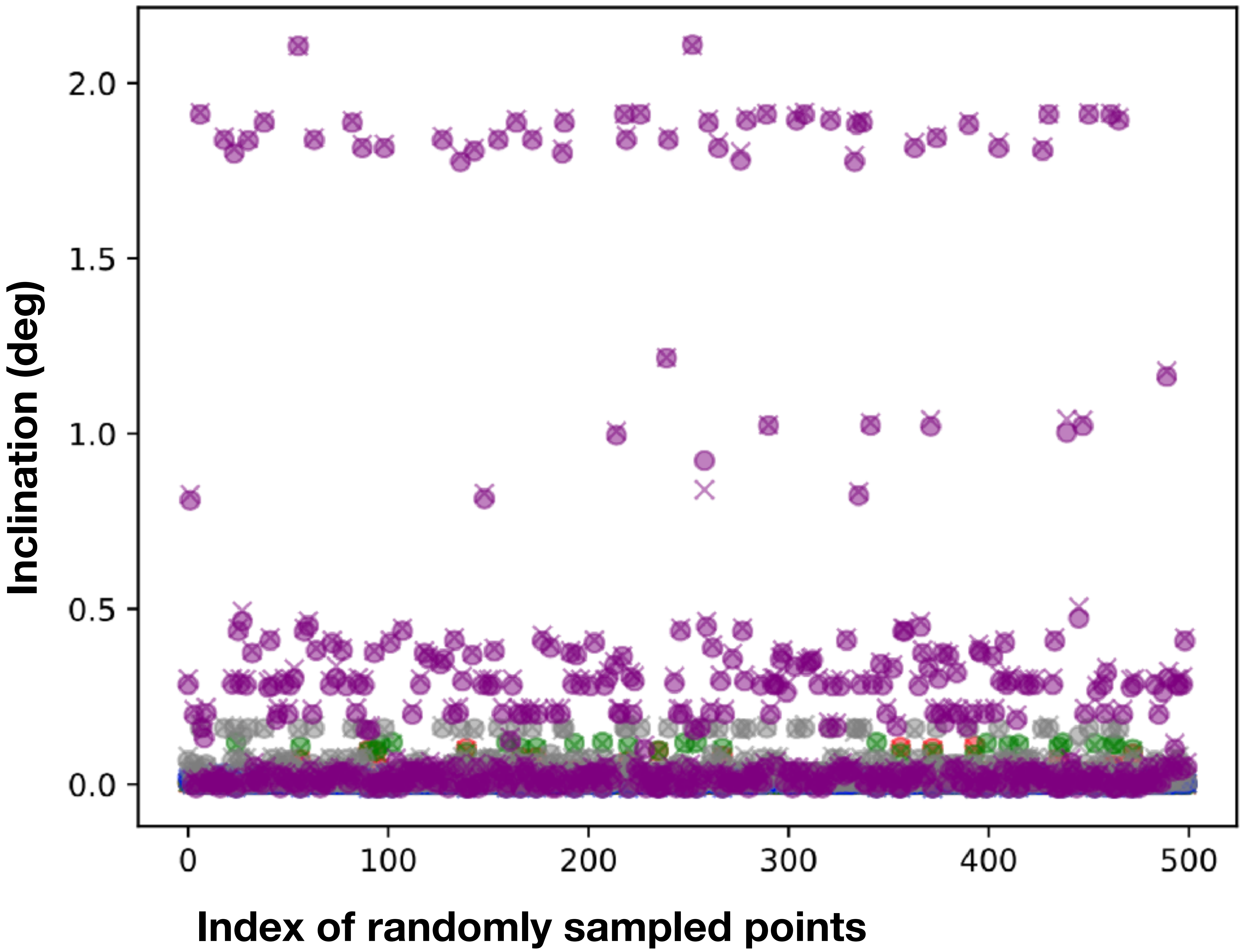


Next *n* steps

- Eccentricities
- Inclinations
- Semi-major axis

Limitation of Time Series Prediction

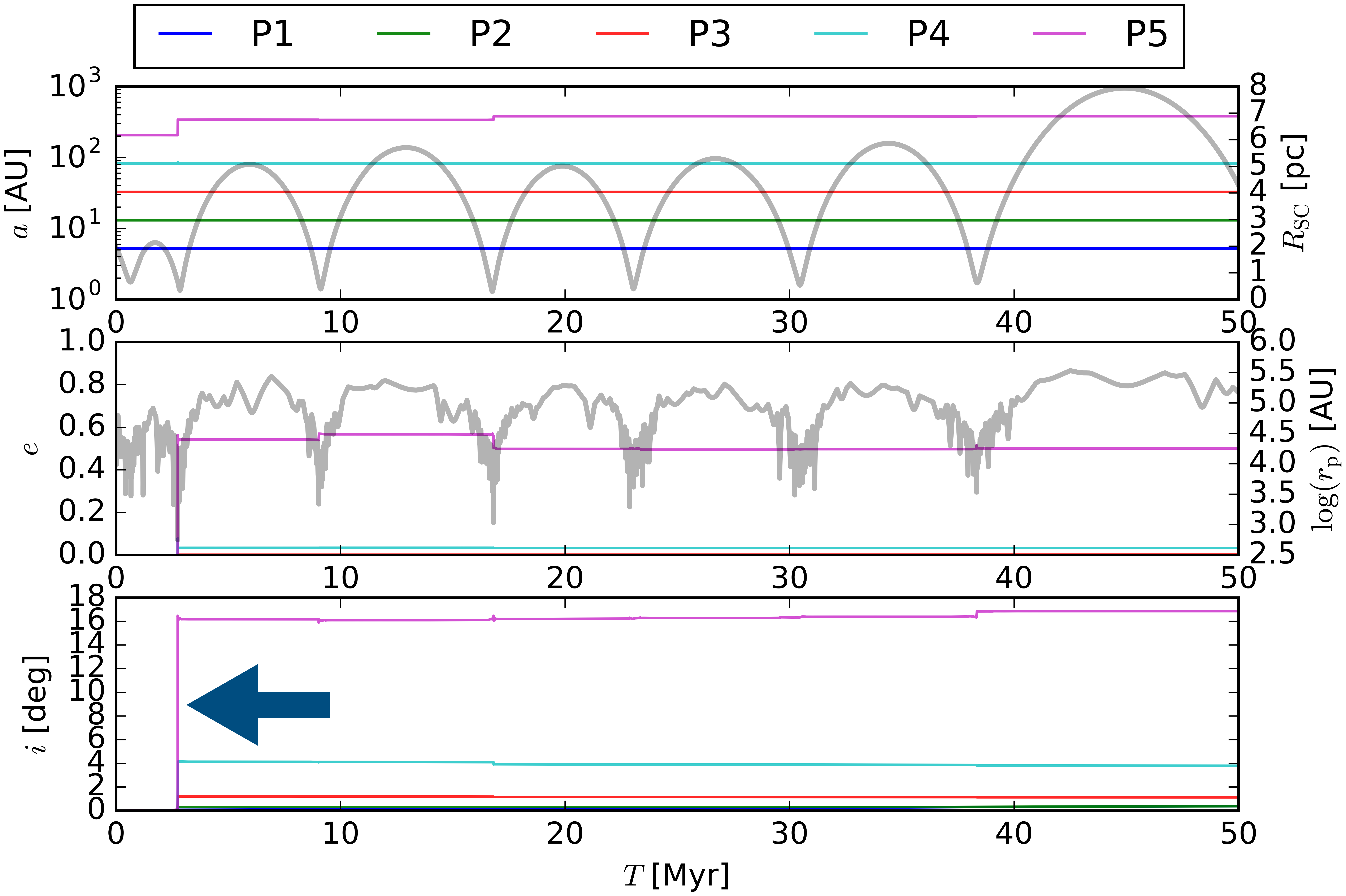
- Reasonably accurate for short timescales
- Errors accumulate over long timescales



Supervised learning

Supervised learning requires:

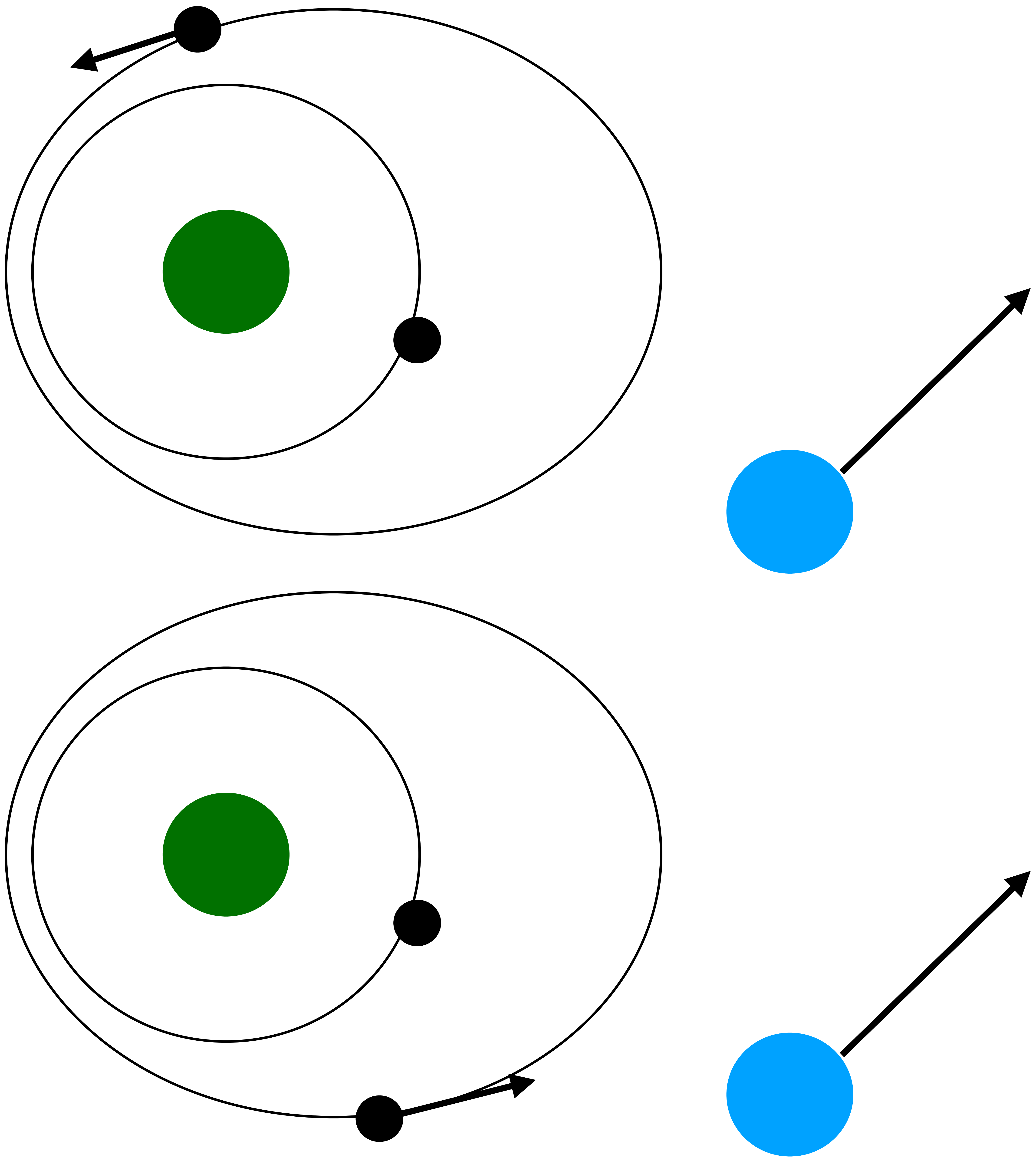
- ➡ Samples are randomized among batches
- ➡ Each batch has the same or similar distribution
- ➡ Samples are independent of each other in the same batch

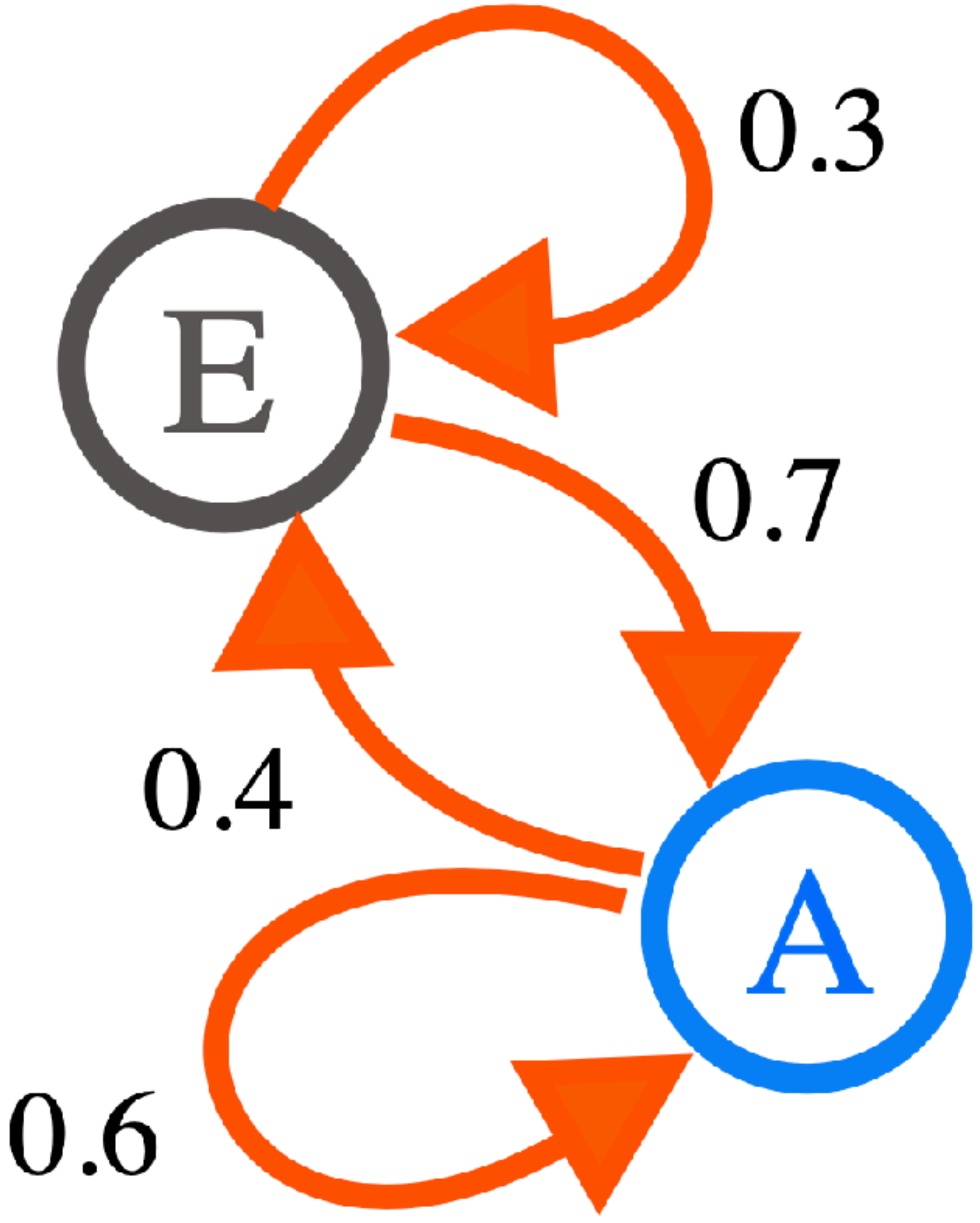


Better solution?

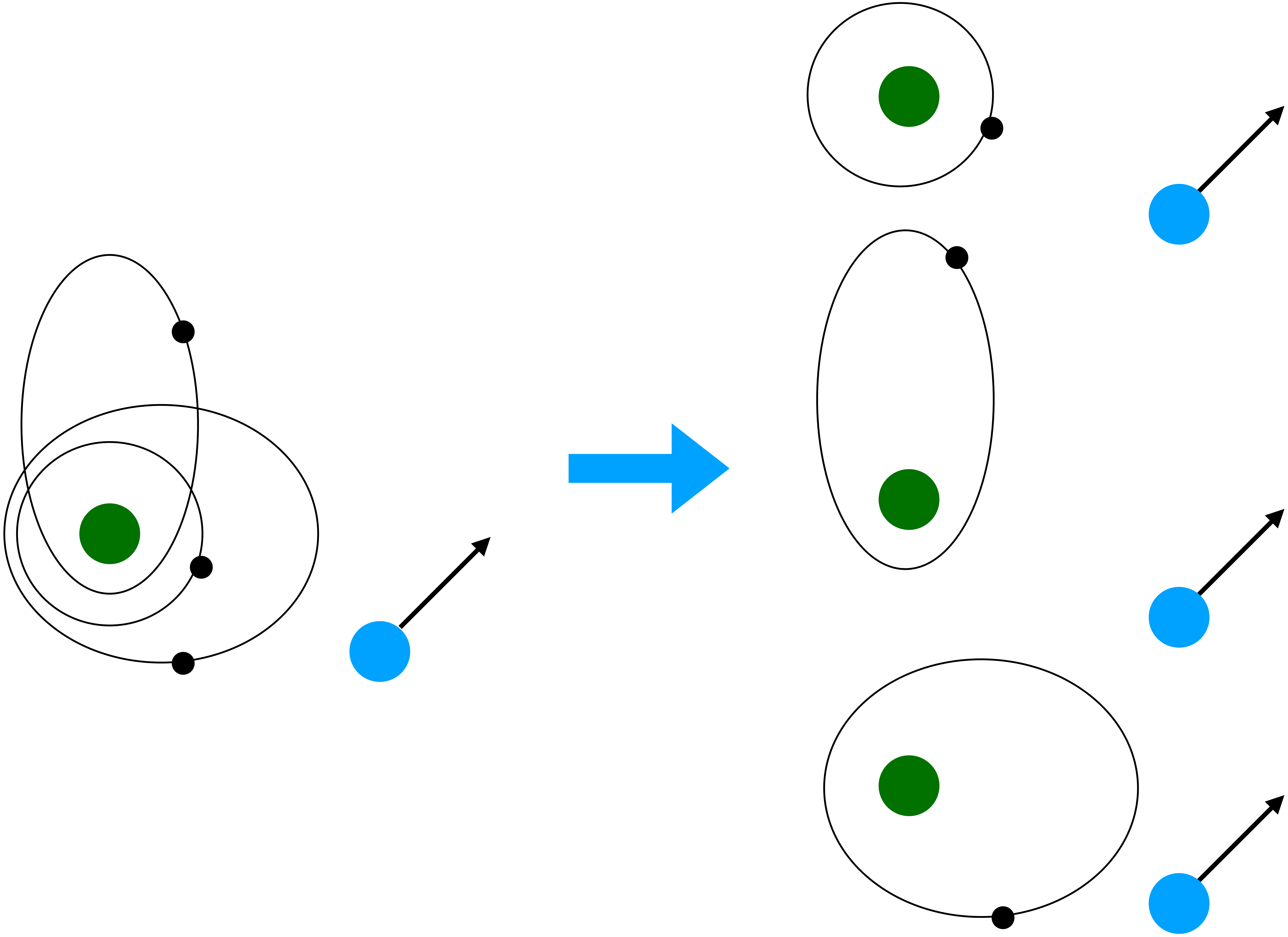
Can a neural network learn the physics by itself?

Stocastic Orbital Changes

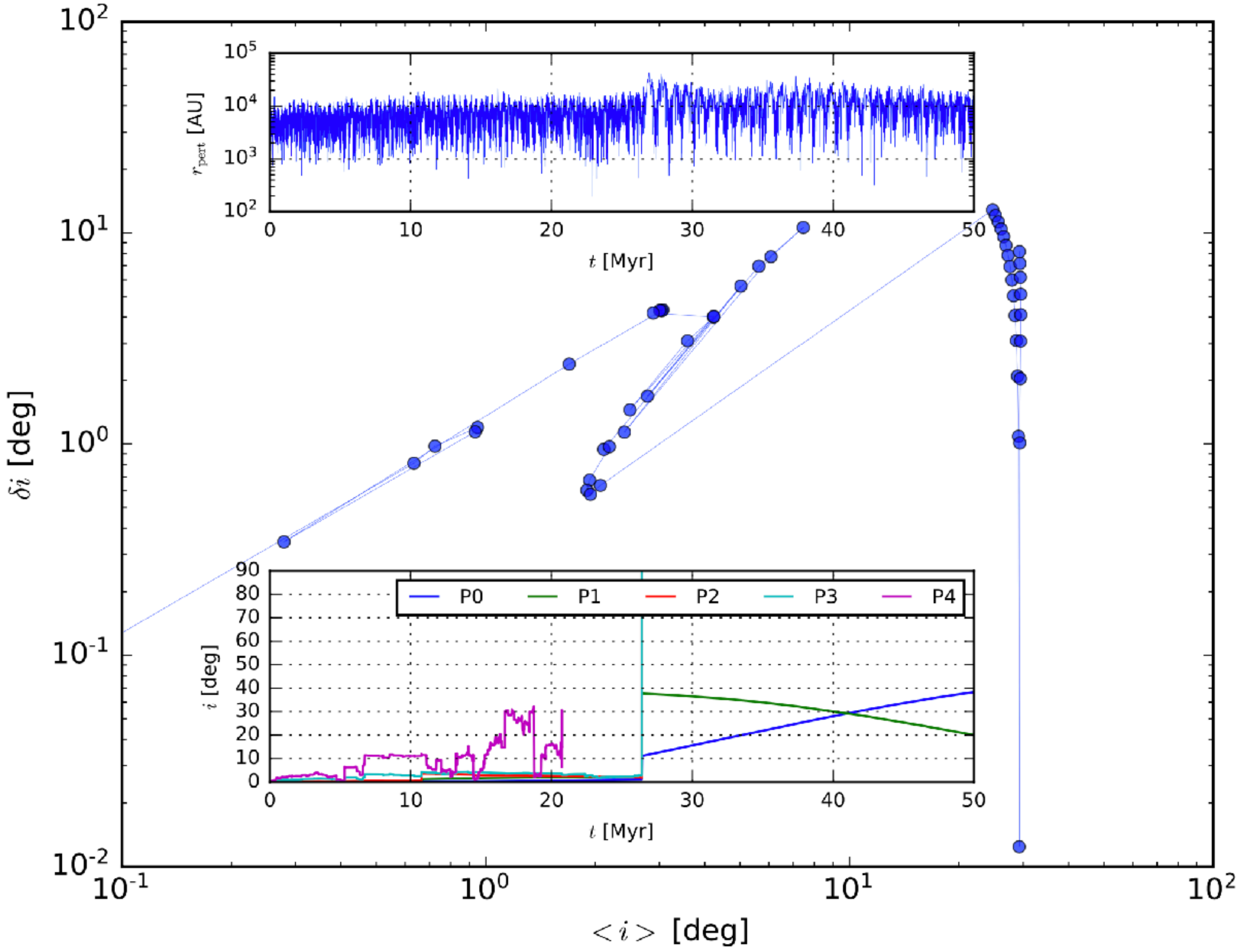




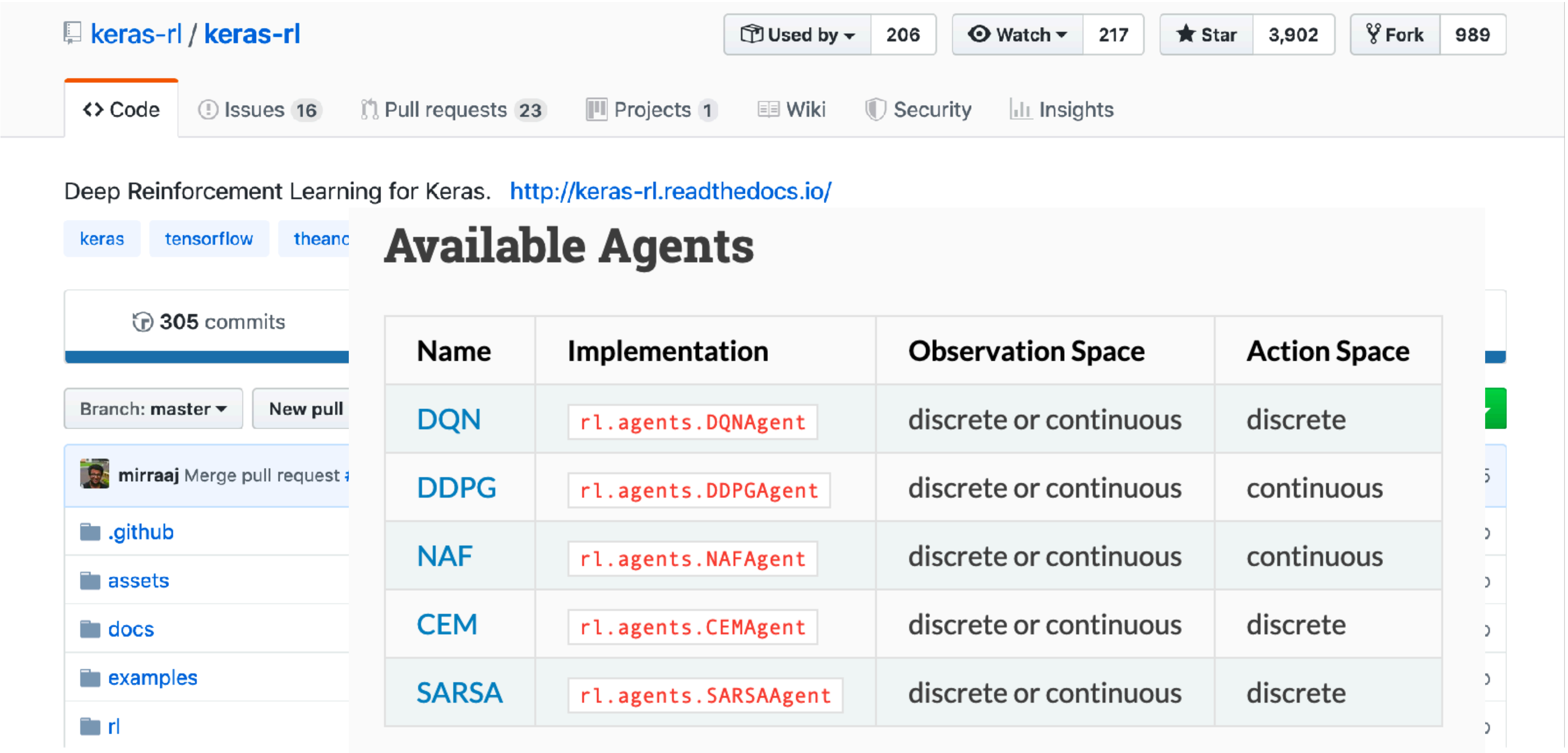
Markov chain. *Source: Wikipedia*



Can a DNN predict phase-space trajectories?



Keras-rl library



Available Agents

Name	Implementation	Observation Space	Action Space
DQN	`rl.agents.DQNAgent`	discrete or continuous	discrete
DDPG	`rl.agents.DDPGAgent`	discrete or continuous	continuous
NAF	`rl.agents.NAFAgent`	discrete or continuous	continuous
CEM	`rl.agents.CEMAgent`	discrete or continuous	discrete
SARSA	`rl.agents.SARSAAgent`	discrete or continuous	discrete

OpenAI Gym

openai / gym

Used by 5,592 | Watch 939 | Star 17,111 | Fork 4,610

Code | Issues 148 | Pull requests 17 | Projects 0 | Wiki | Security | Insights

A toolkit for developing and comparing reinforcement learning algorithms. https://gym.openai.com/

1,036 commits | 78 branches | 21 releases | 180 contributors | View license

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

mor-katz and pzhokhov Modify MultiDiscrete comment regarding NOOP (#1537) — Latest commit c03ec69 2 days ago

bin	fix mujoco-related build failure	8 months ago
docs	remove instructions for adding new environments to gym (#1458)	2 months ago
examples	cleanup examples/scripts/sim_env, make it python3 compatible	4 months ago
gym	Modify MultiDiscrete comment regarding NOOP (#1537)	2 days ago
scripts	Cleanup, removal of unmaintained code (#836)	last year
tests/gym/wrappers	Implement FilterObservationWrapper (#1500)	26 days ago

Modeling with Reinforcement Learning

Trajectory $\tau = (s_0, a_0, s_1, a_1, s_2, a_2, \ldots, s_H, a_H, s_{H+1})$

Reward $R(\tau) = r_1 + r_2 + r_3 + \ldots + r_H + r_{H+1}$

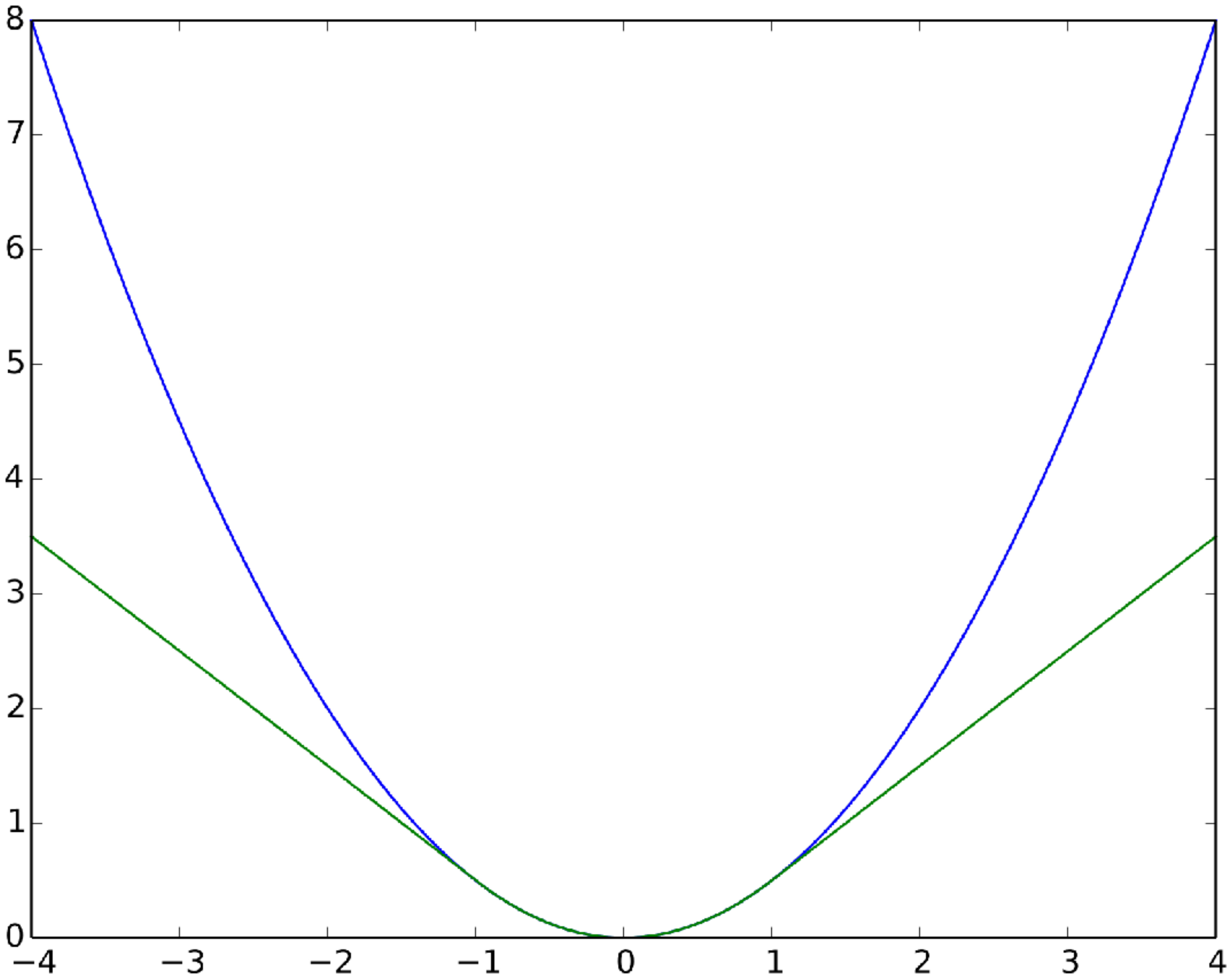
Expectation $U(\theta) = \sum_{\tau} P(\tau, \theta) R(\tau)$

Gradient $\nabla_\theta \approx \hat{g} := \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H} \nabla_\theta \log \pi_\theta \left(a_t^{(i)} \mid s_t^{(i)} \right) R(\tau^{(i)})$

Update $\theta \leftarrow \theta + \alpha \hat{g}$

Huber Loss

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \textbf{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \textbf{otherwise.} \end{cases}$$



arXiv:1509.02971

CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING

Timothy P. Lillicrap,[*] Jonathan J. Hunt,[*] Alexander Pritzel, Nicolas Heess,
Tom Erez, Yuval Tassa, David Silver & Daan Wierstra
Google Deepmind
London, UK
{countzero, jjhunt, apritzel, heess,
etom, tassa, davidsilver, wierstra} @ google.com

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

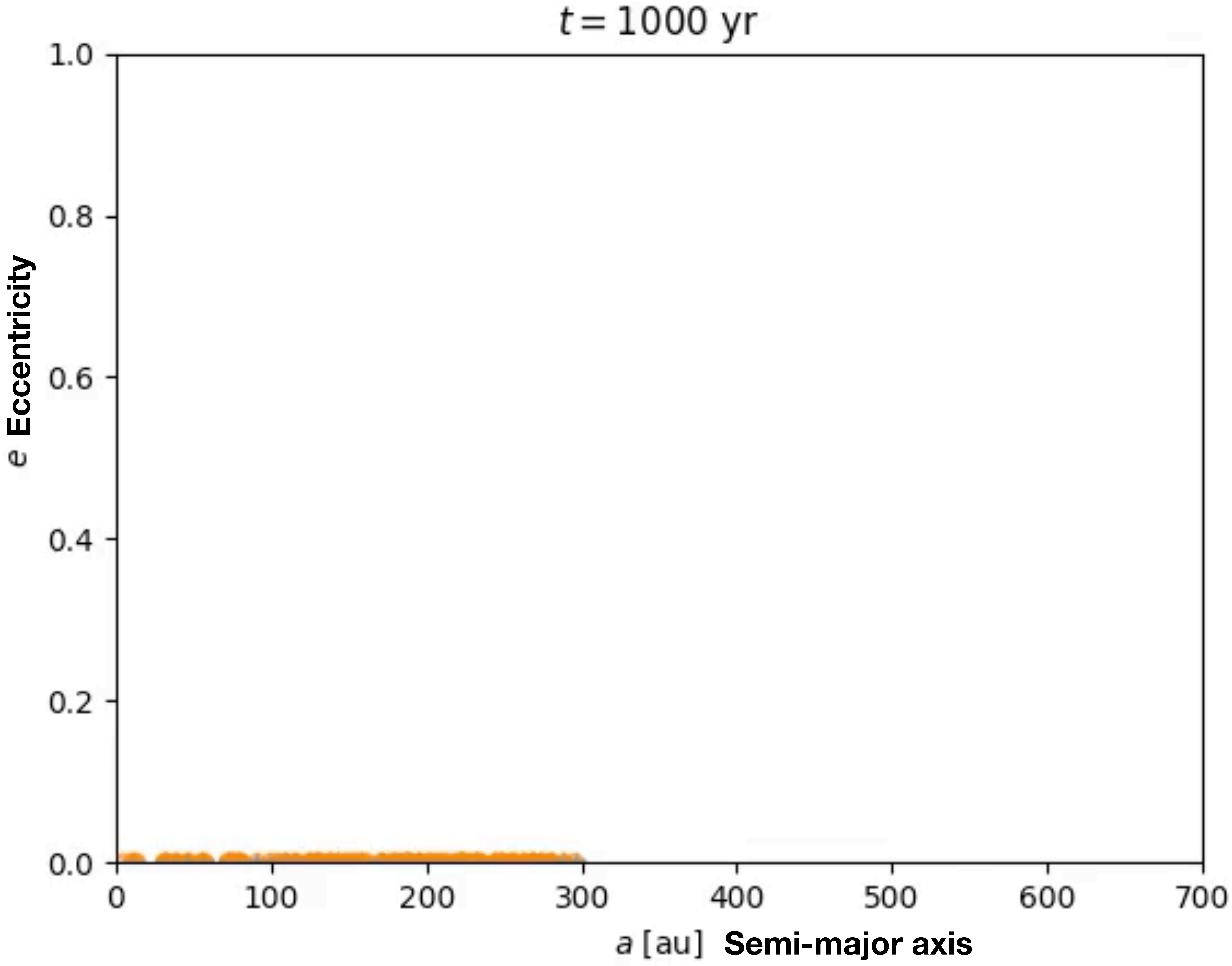
$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

Lillicrap et al. (2015)

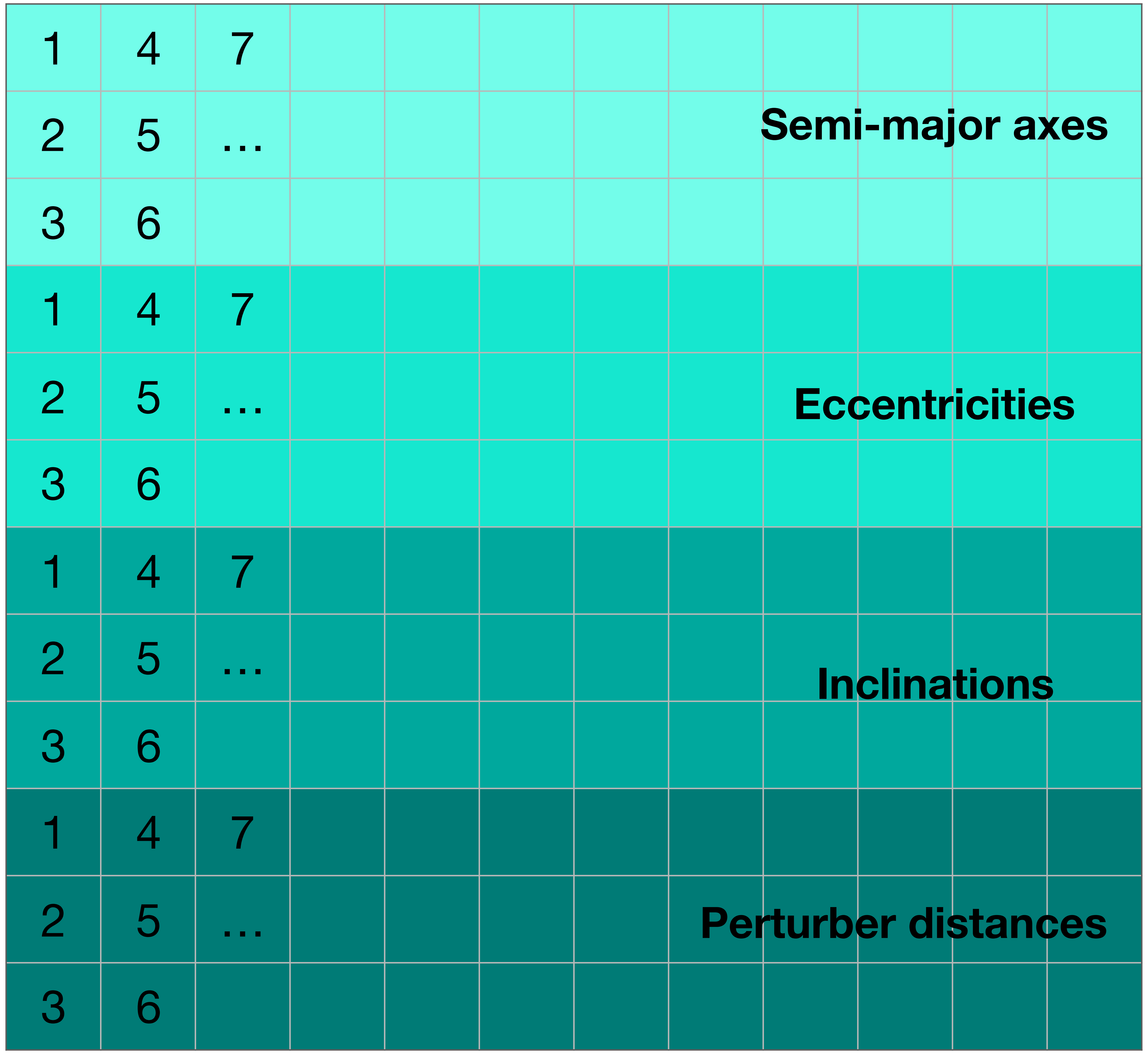
Comparing with N-body simulations



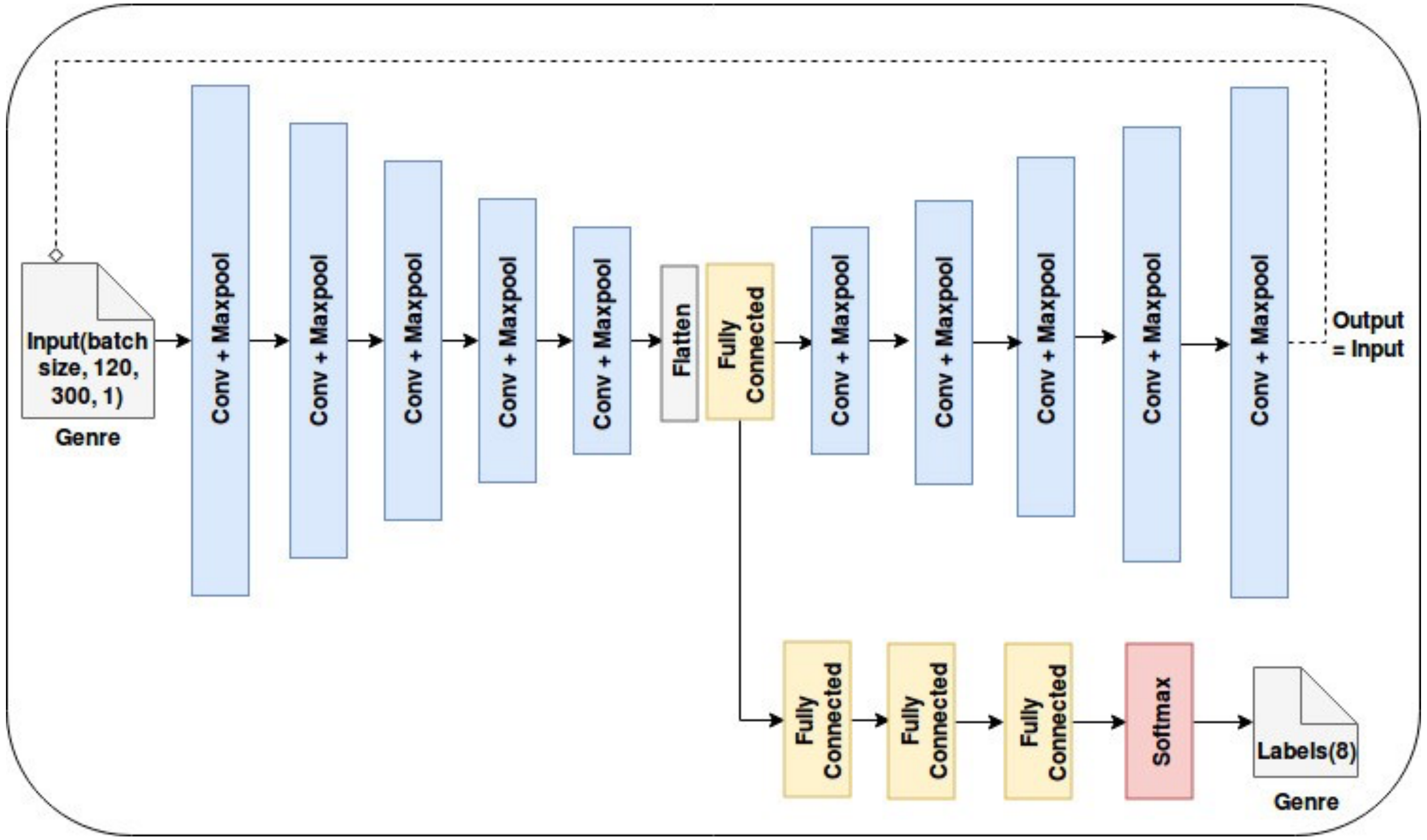
DQN managed to capture the basic physics
Speed up by a factor of 10^2 - 10^6

Predict accuracy depends on the resolution of the training data

Predict the future according to the past: pattern recognition

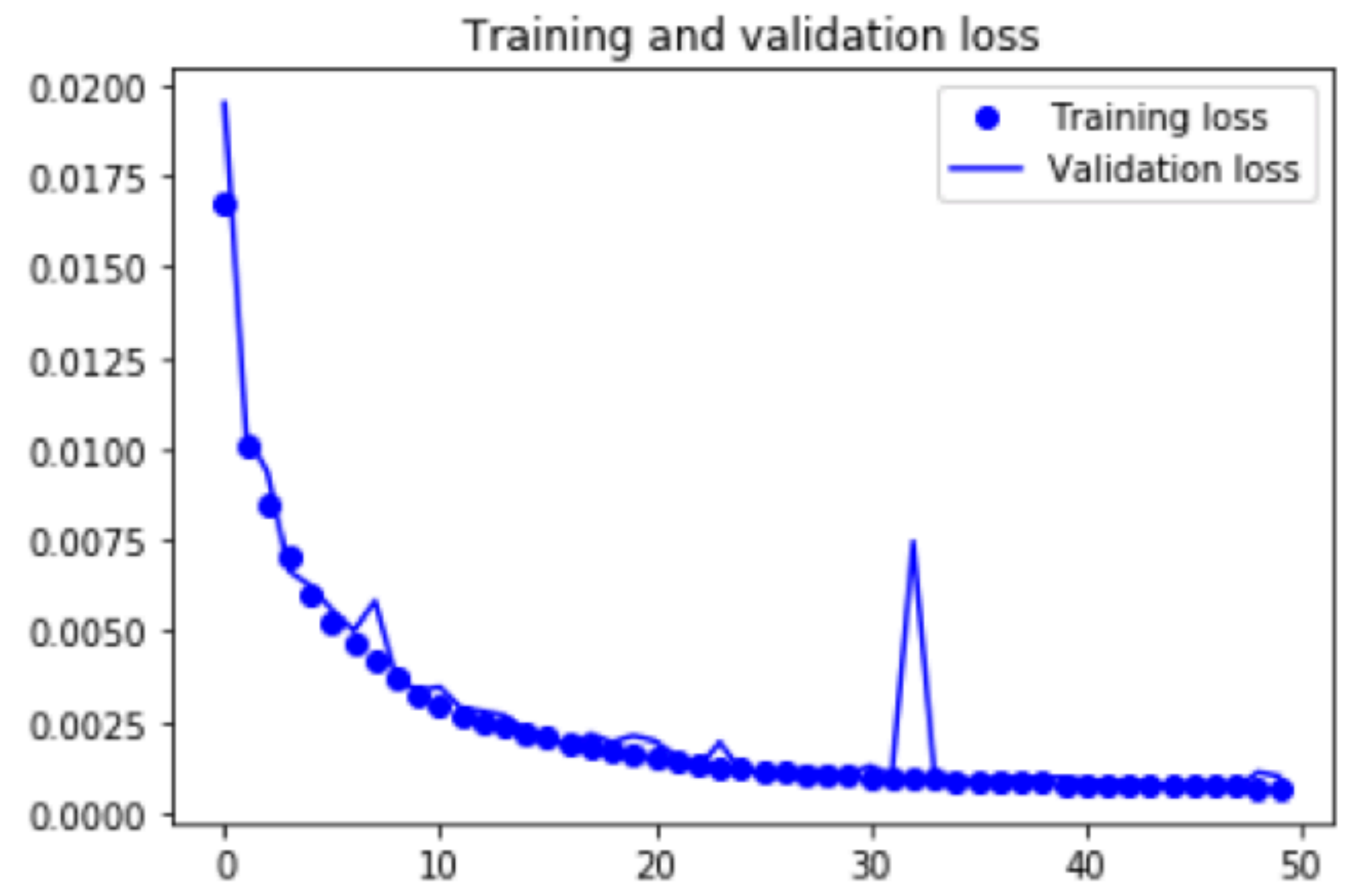


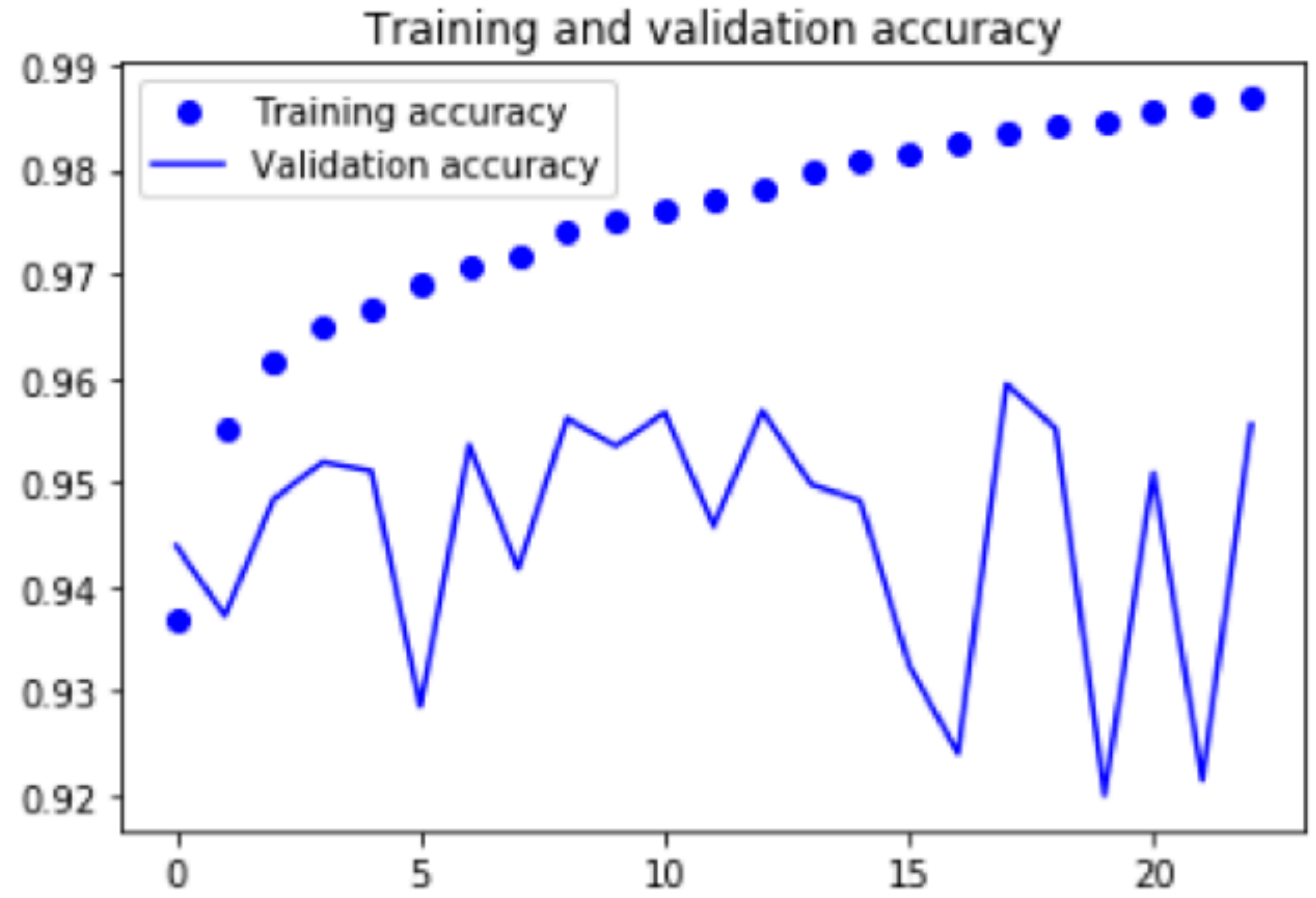
Variational Autoencoder as classifier

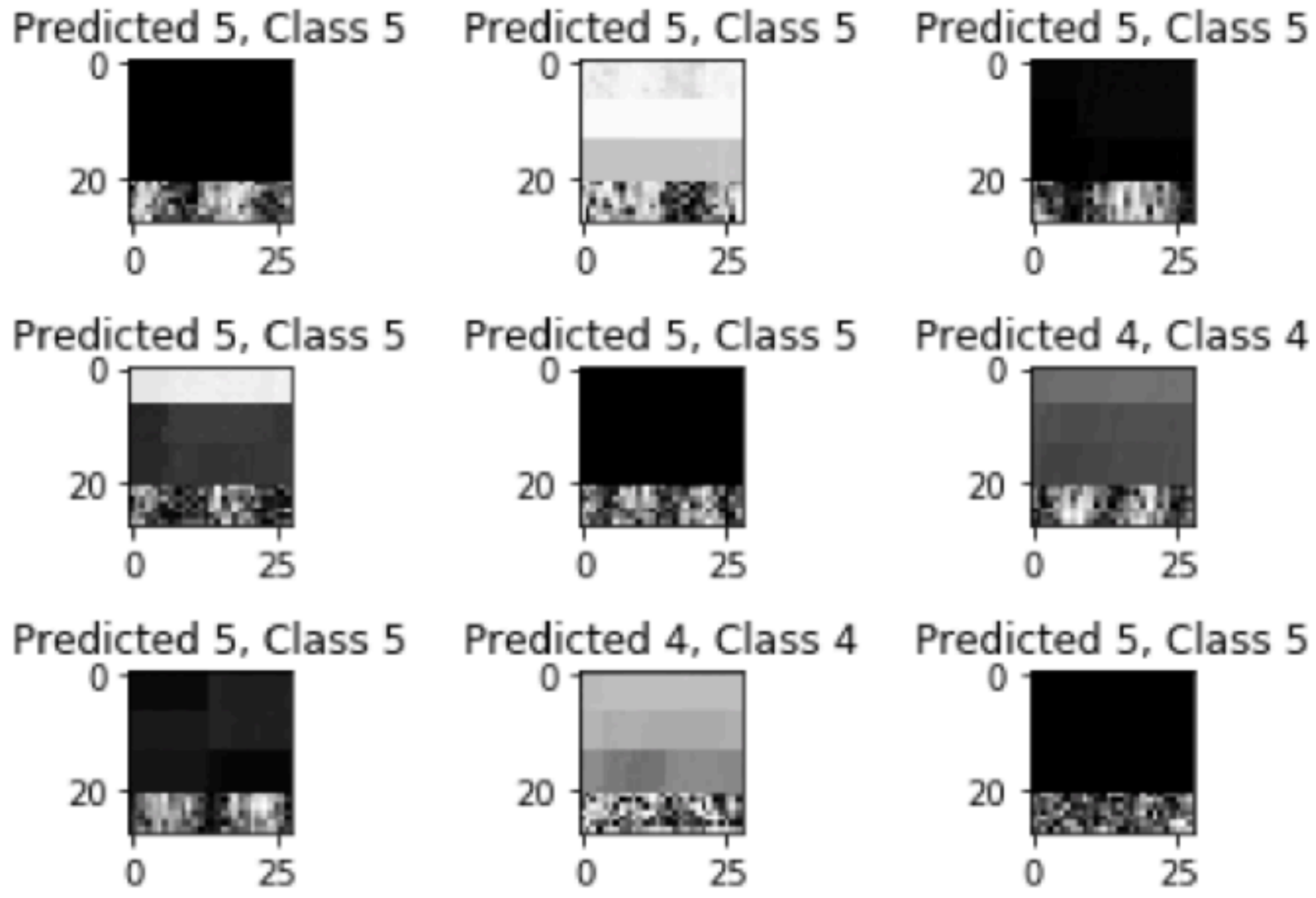


Label: the change of orbital eccentricity in the next 1 Myr

Predict the future according to the past: pattern recognition







	precision	recall	f1-score	support
Class 0	0.97	0.98	0.98	918
Class 1	0.74	0.76	0.75	141
Class 2	0.63	0.83	0.72	252
Class 3	0.76	0.71	0.73	897
Class 4	0.93	0.92	0.93	24992
Class 5	0.97	0.98	0.97	71386
Class 6	0.83	0.80	0.81	1471
Class 7	0.74	0.80	0.77	440
Class 8	0.85	0.74	0.79	221
Class 9	0.89	0.89	0.89	162
micro avg	0.96	0.96	0.96	100880
macro avg	0.83	0.84	0.83	100880
weighted avg	0.96	0.96	0.96	100880

Summary

W	+	X	+	Y	+	Z	=	S
Past state(s)		**Current state**		**Perturbation**		**Future state(s)**		**New state**

Challenges:
- Underlying systems chaotic
- High dynamic range
- Extremely imbalance training samples
- Extremely long term prediction needed
- System not deterministic

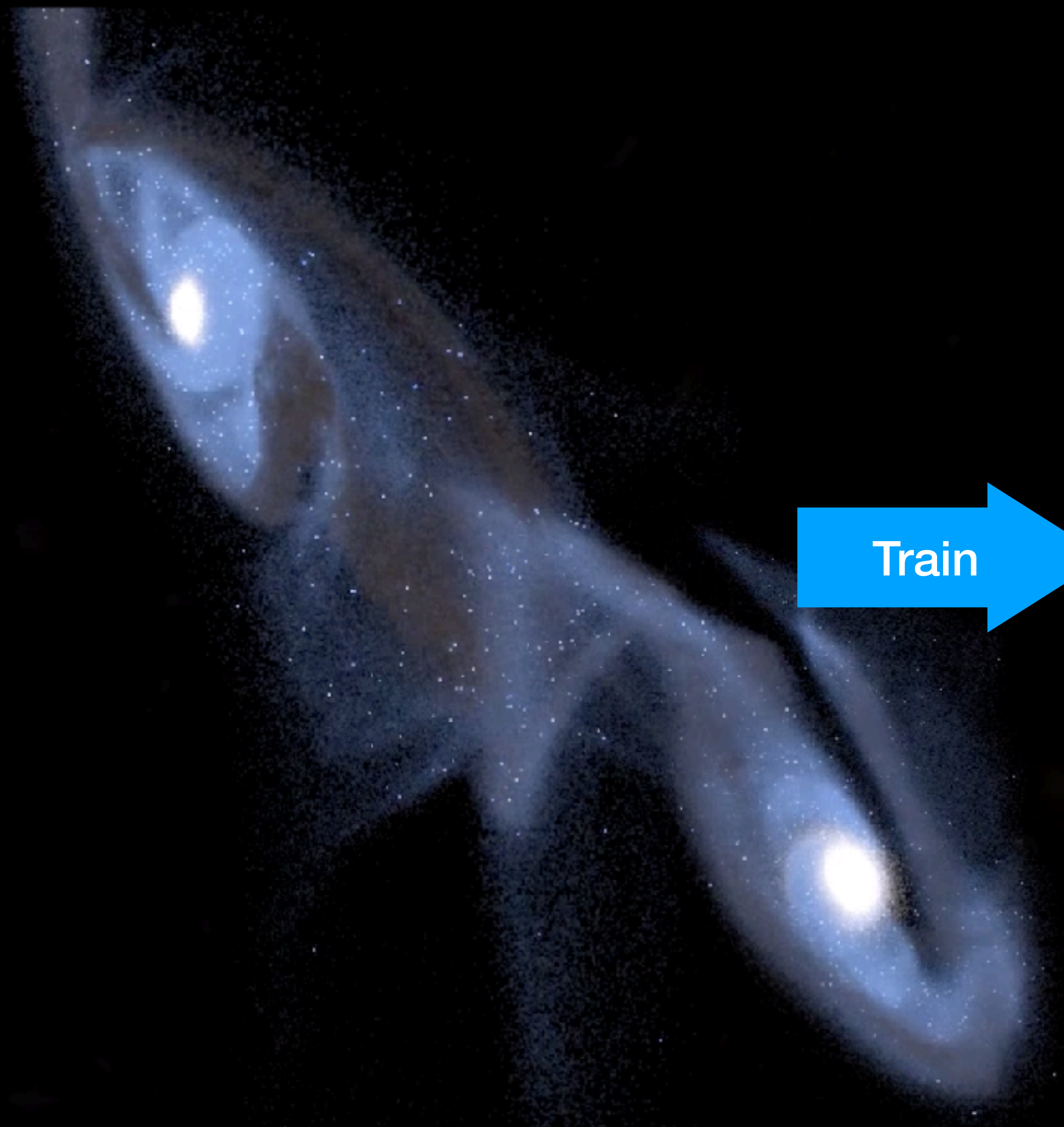
Conclusions
- Supervised learning is useful, but only for short-term prediction
- It is unusual to use RL for time series prediction, but it seems that RL can indeed learn physical laws
- Long term error inevitable, because we can’t change the chaotic nature of the systems
- Multiple neural network architectures needed to collectively tackle the problem
- DL/RL can be useful for multi-scale modeling in physics

Bonus slides

Galaxy merger Simulations

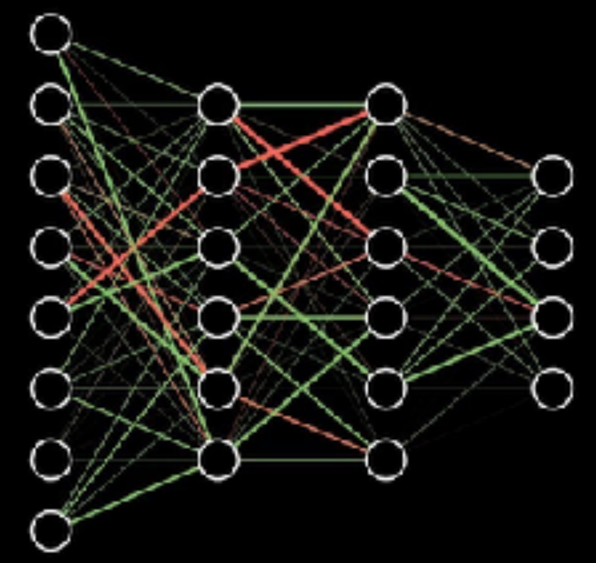
Credit: Jeroen Bédorf

Simulation



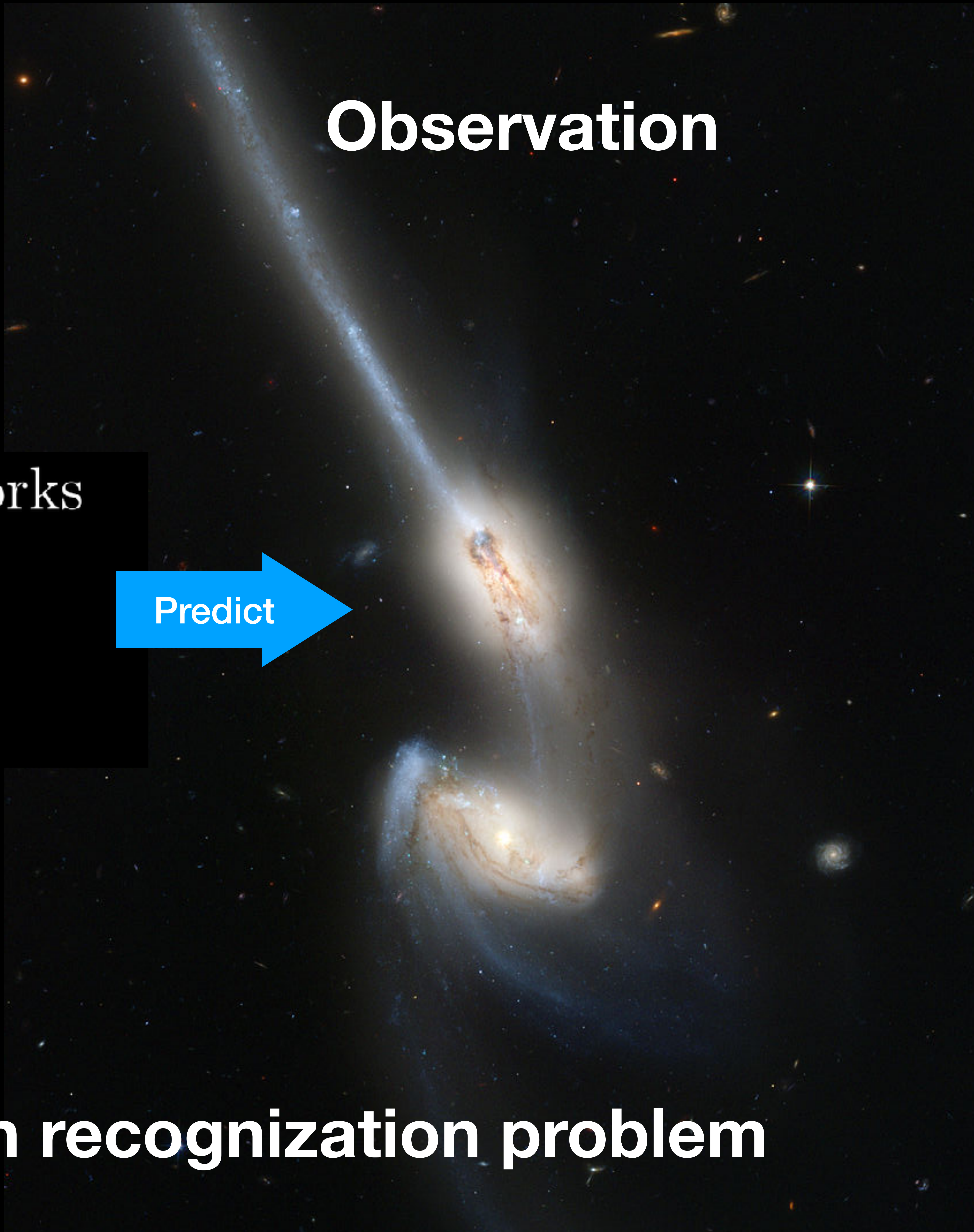
Train

Neural Networks



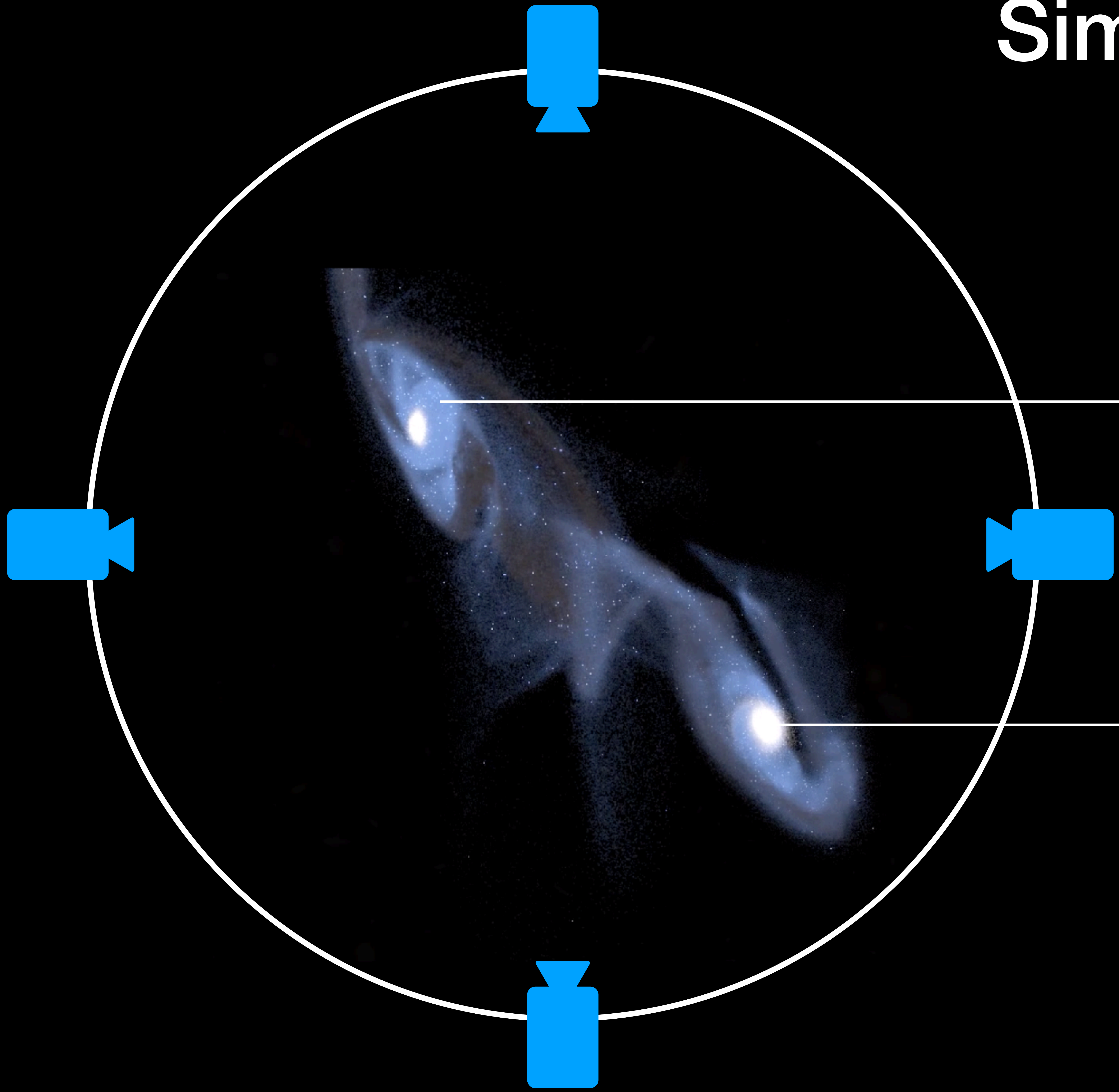
Predict

Observation



Astrophysical problem → Pattern recognization problem

Simulations & Visualization



M_1

$= 1:1, 1:2, 1:3, 2:3, \ldots$

M_2

~60 GB of image data

A lot more simulation data