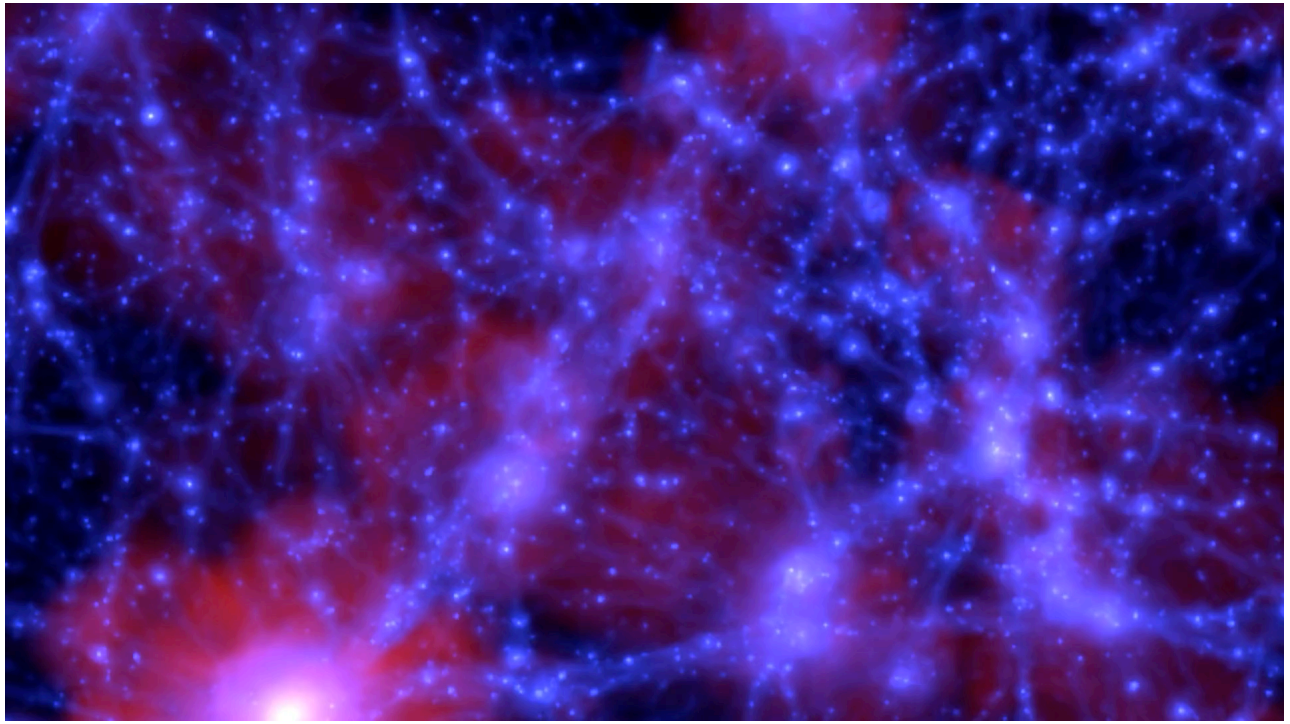
Rediscovering orbital mechanics with Machine Learning



Pablo Lemos

Learning to Discover - Representation Learning Workshop

https://arxiv.org/abs/2202.02306

Collaborators

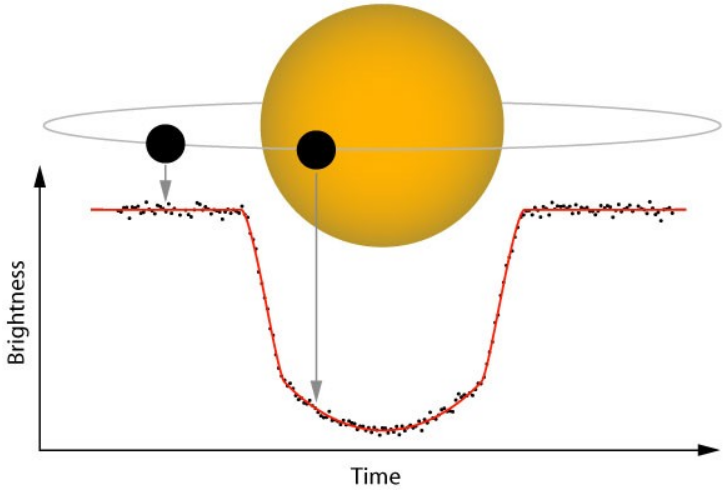
- Niall Jeffrey (UCL/ENS)
- Miles Cranmer (Princeton)
- Shirley Ho (Princeton/Flatiron Institute)
- Peter Battaglia (DeepMind)

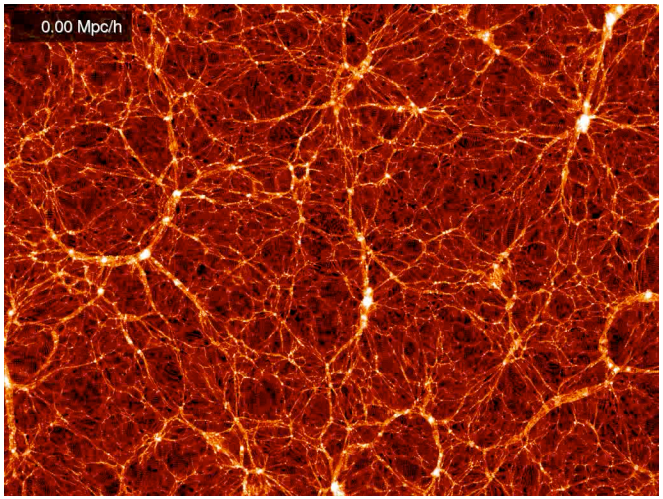
UNIVERSITY OF SUSSEX

UCL

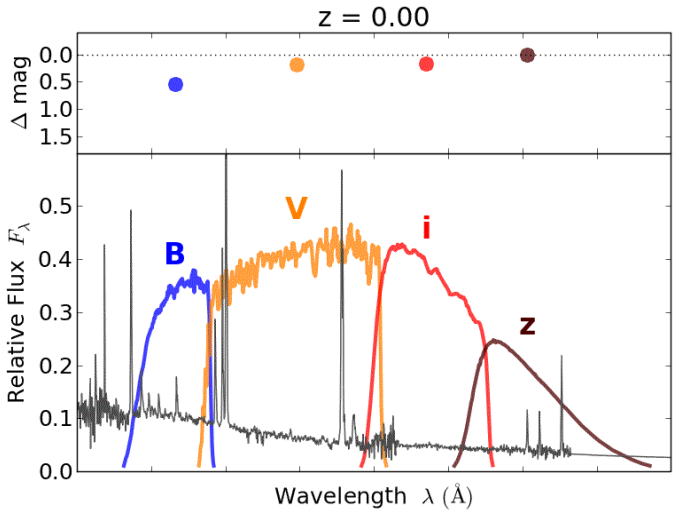
Introduction

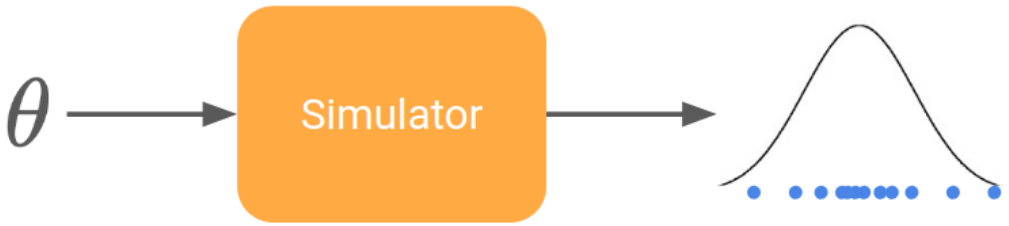
ML in astronomy





- Image classification
- Extrapolation
- Generate simulations
- Parameter estimation

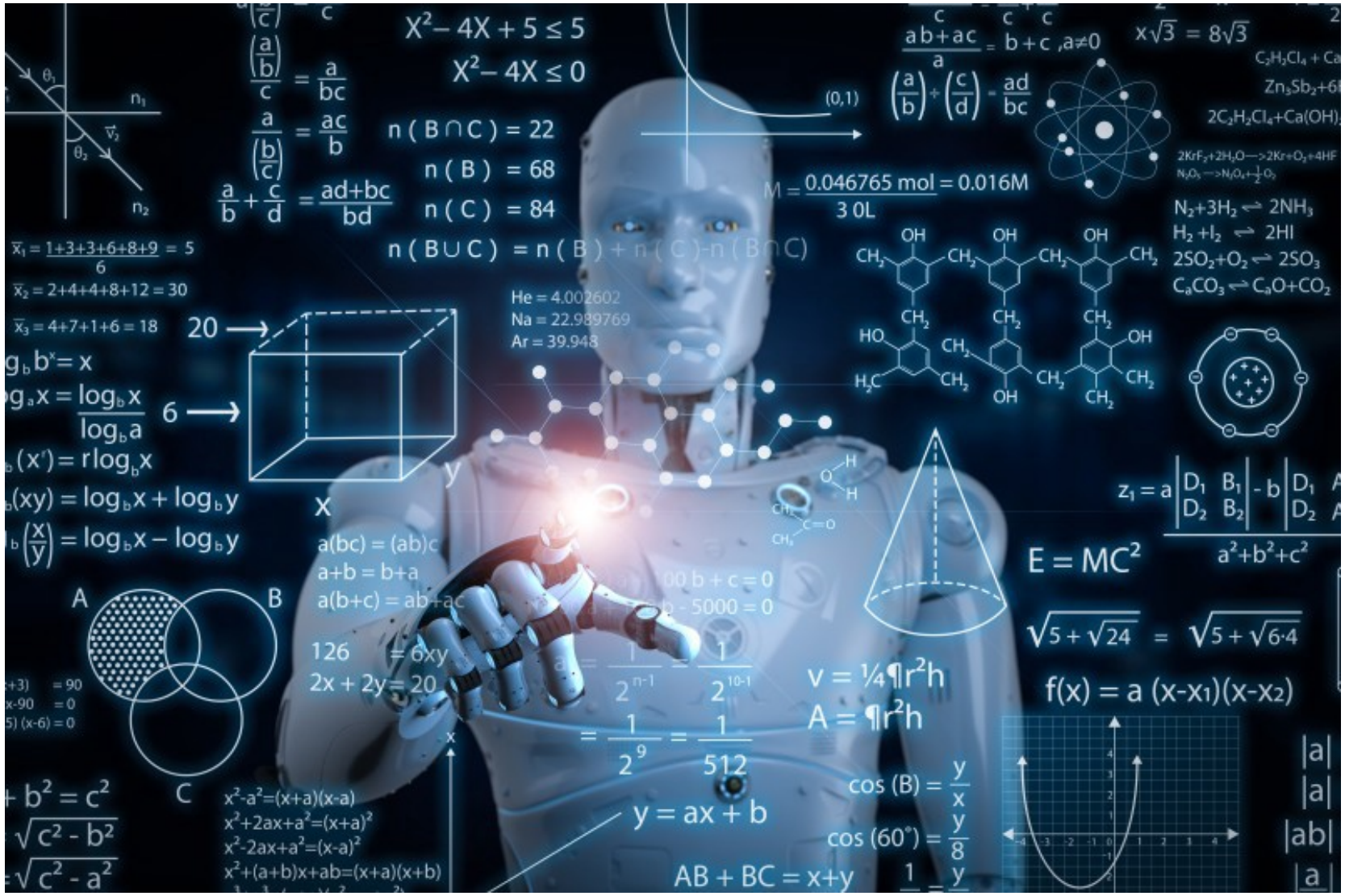




UNIVERSITY OF SUSSEX

UCL

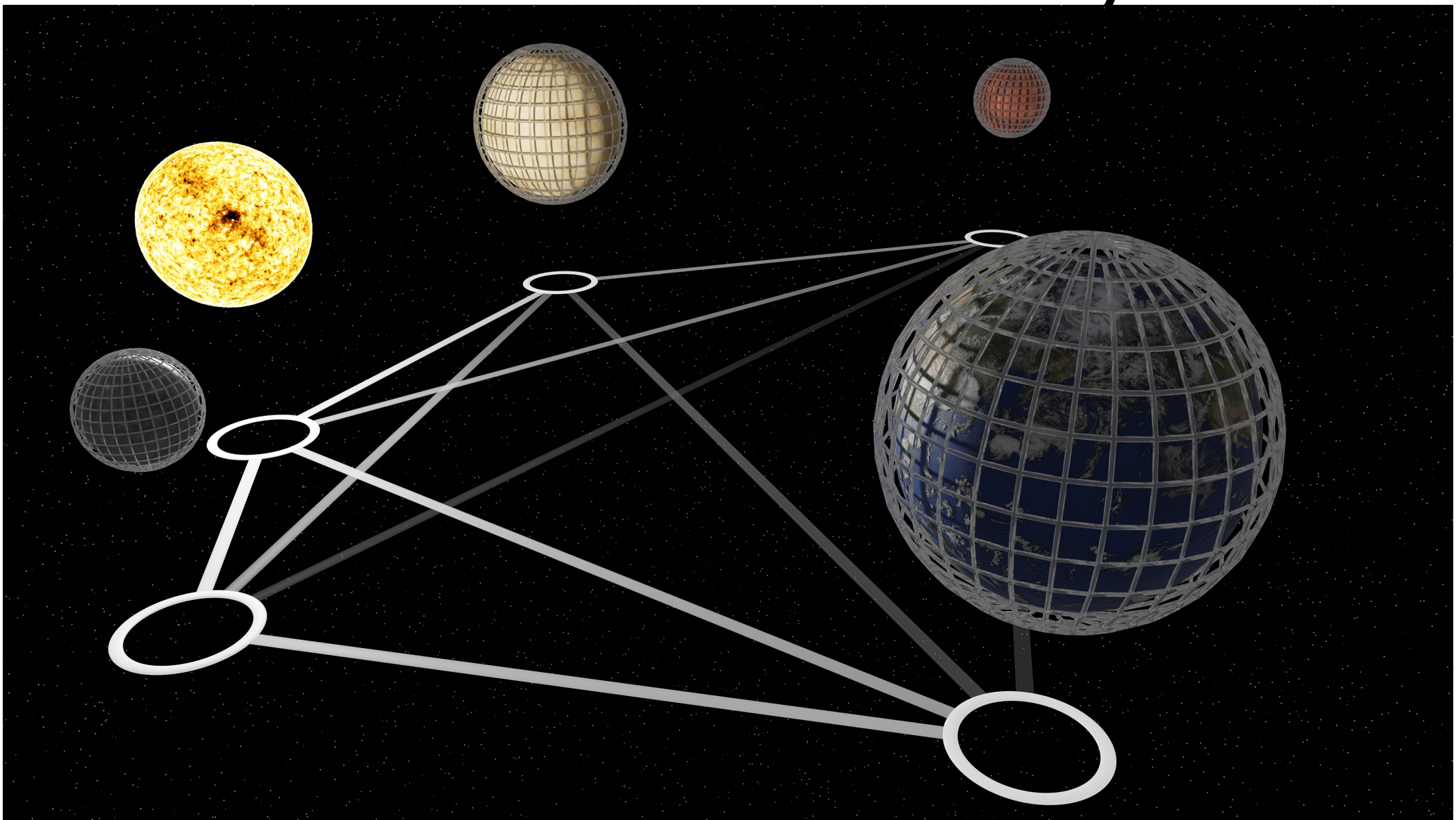
Scientific discovery



UNIVERSITY OF SUSSEX

UCL

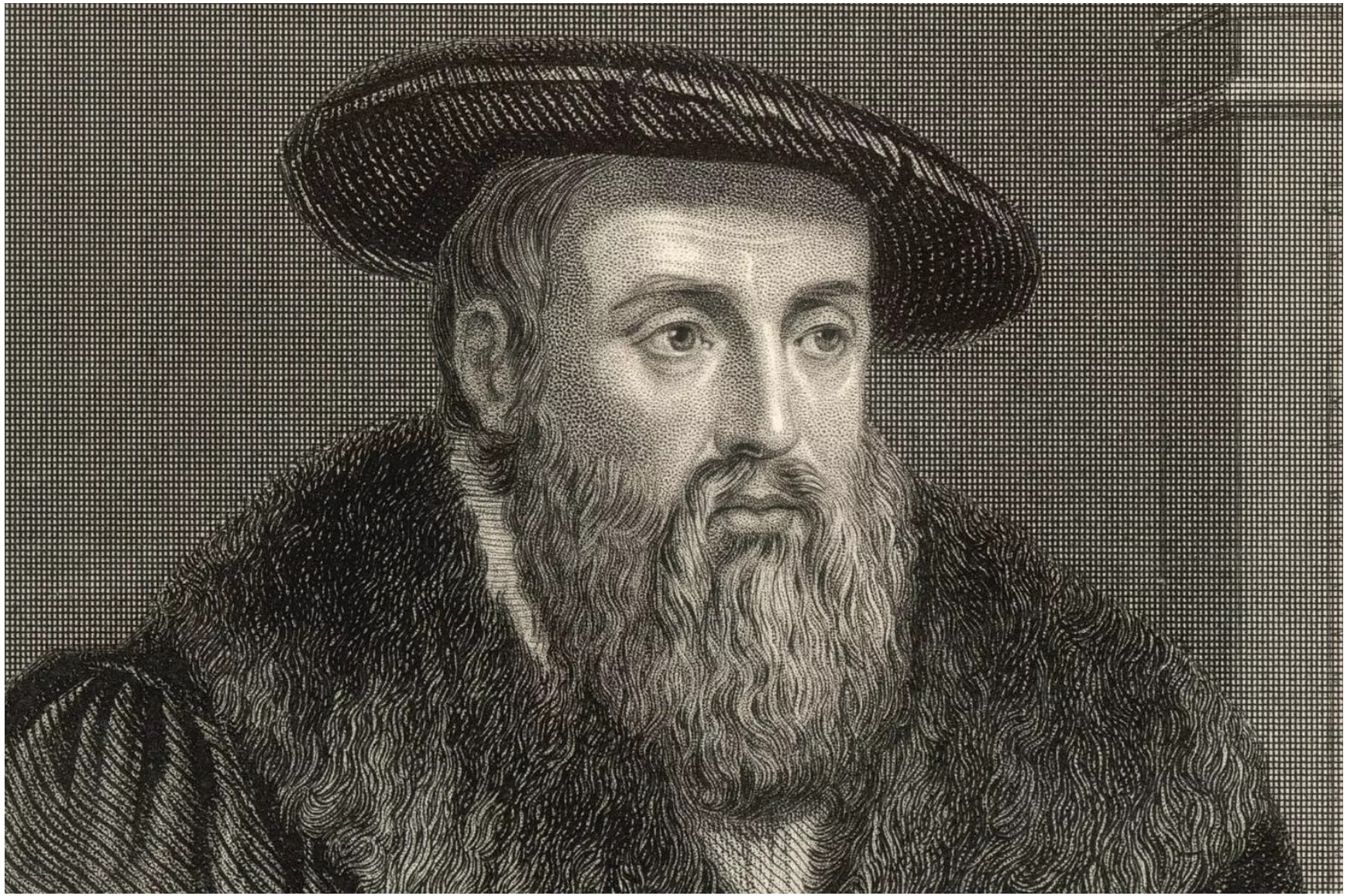
Scientific discovery



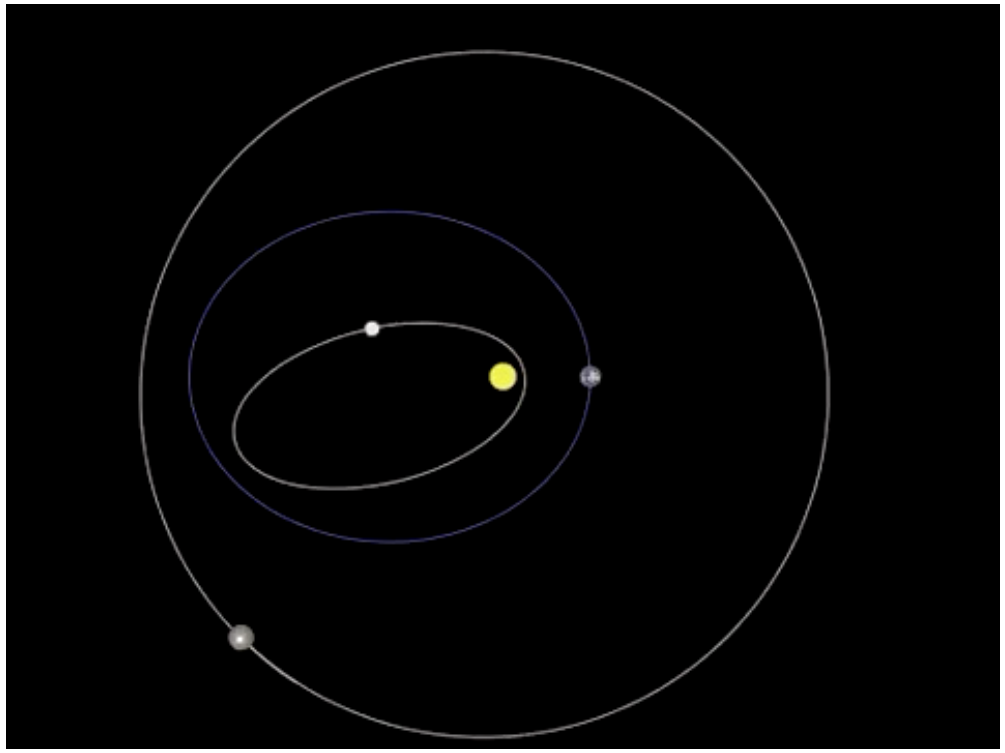
UNIVERSITY OF SUSSEX

UCL

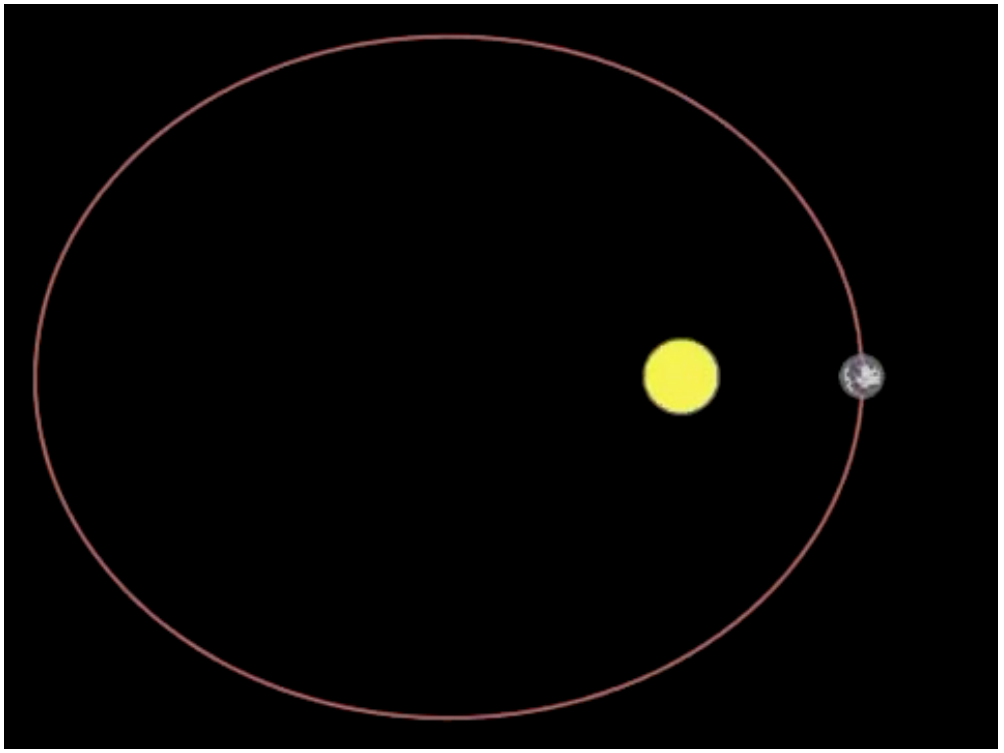
Kepler's Laws



Kepler's Laws



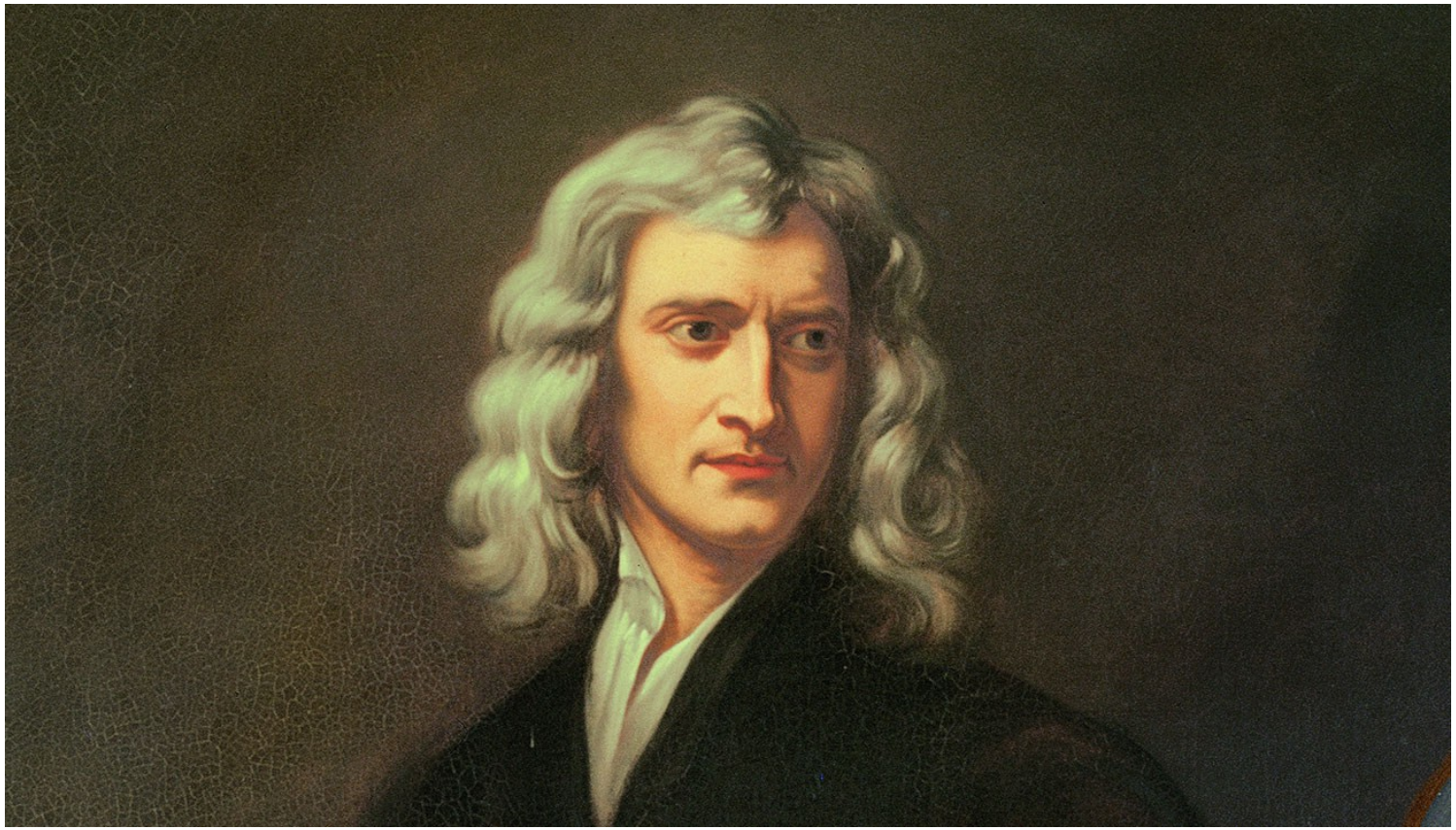
Orbits are ellipses

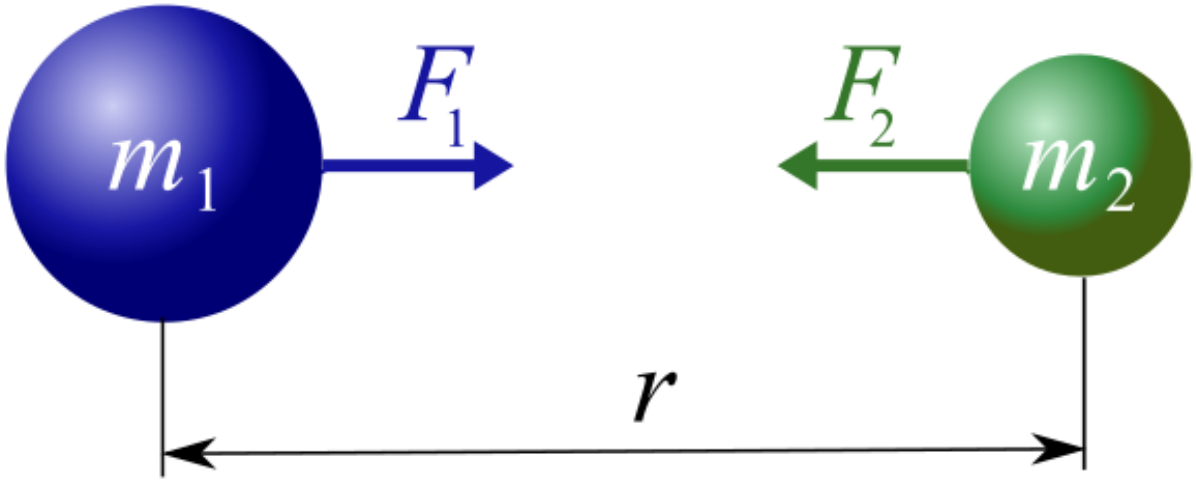


Equal areas in equal times

UNIVERSITY OF SUSSEX

UCL





UNIVERSITY OF SUSSEX

UCL

Samuel Brockington
@brockingtonian

1/8 We've just lost our "Newton's Apple Tree" to Storm Eunice (gravity is such a downer, arf arf). It was planted in 1954, so has stood at the Brookside entrance @CUBotanicGarden for 68 years. An iconic tree, and sad loss. But what does it mean to be "Newton's Apple Tree"? ... a🧵

5:54 PM · Feb 19, 2022

Read the full conversation on Twitter

742 Reply Share

Read 34 replies

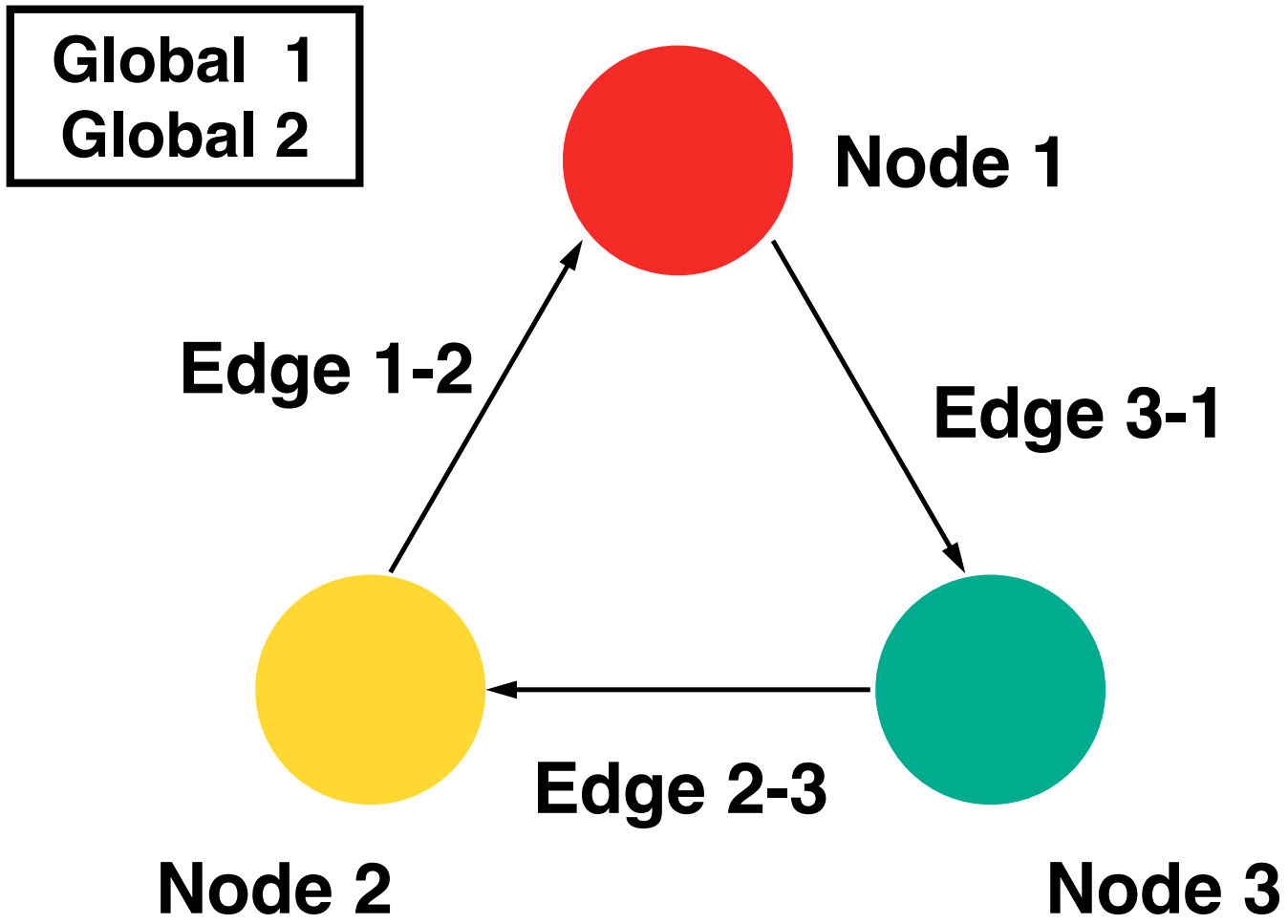
How do we do this?

1. Train a GN to learn the law of physics.
2. Extract the equation from that GN.

How do we do this?

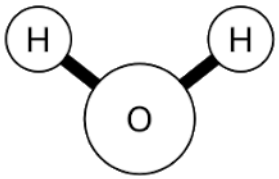
1. **Train a GN to learn the law of physics.**
2. Extract the equation from that GN.

Graph Neural Networks



Many Complex Systems are Structured

Molecule



Mass-Spring System

n-body System

Rigid Body System

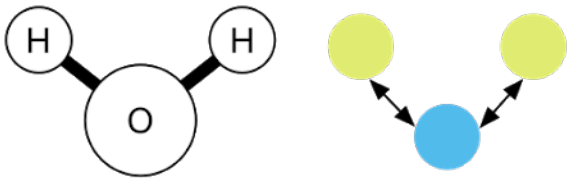
Sentence

The brown
dog jumped.

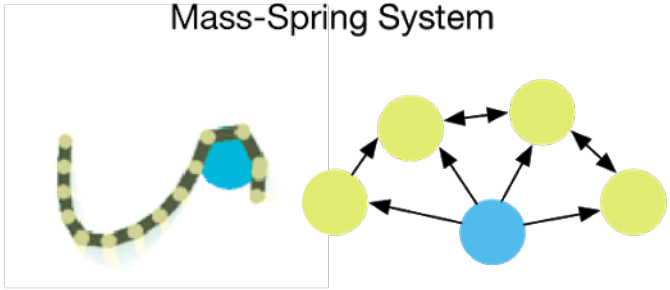
Image

Many Complex Systems are Structured

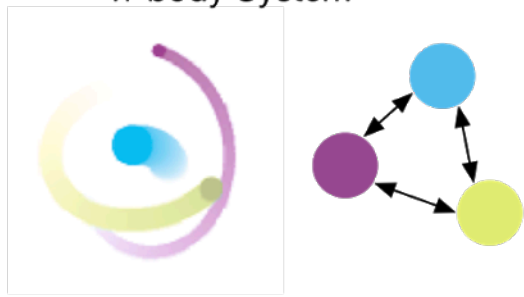
Molecule



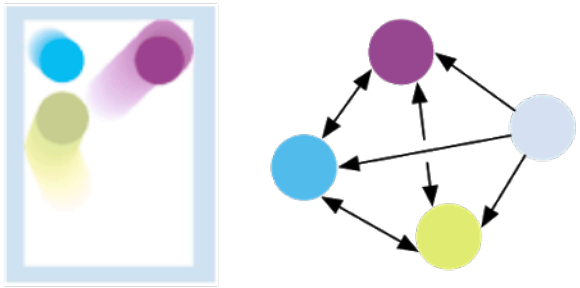
Mass-Spring System



n-body System



Rigid Body System



Sentence and Parse Tree

The brown
dog jumped.

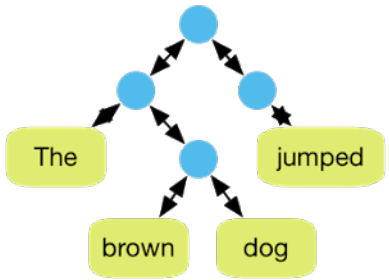
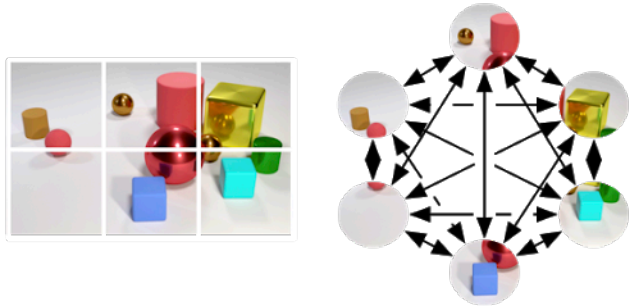
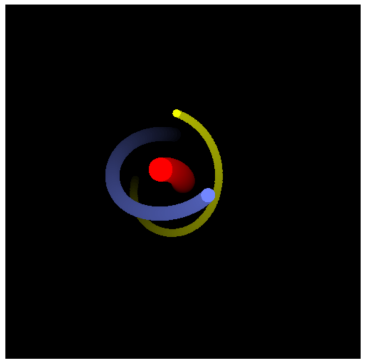


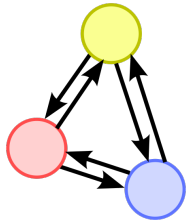
Image and Fully-Connected Scene Graph



Physical systems as graphs

n-body

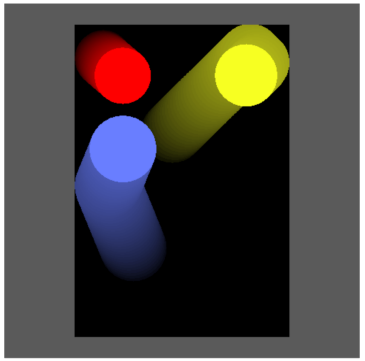


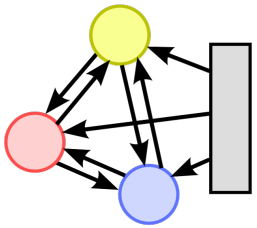


Nodes: bodies

Edges: gravitational forces

Balls

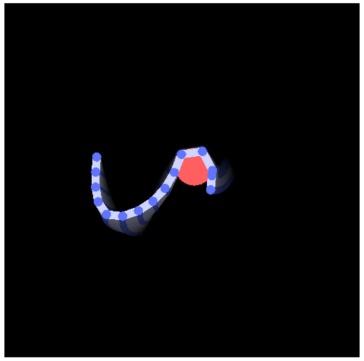


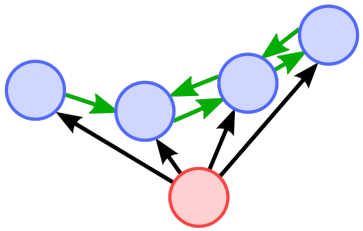


Nodes: balls

Edges: rigid collisions between balls/ walls

String

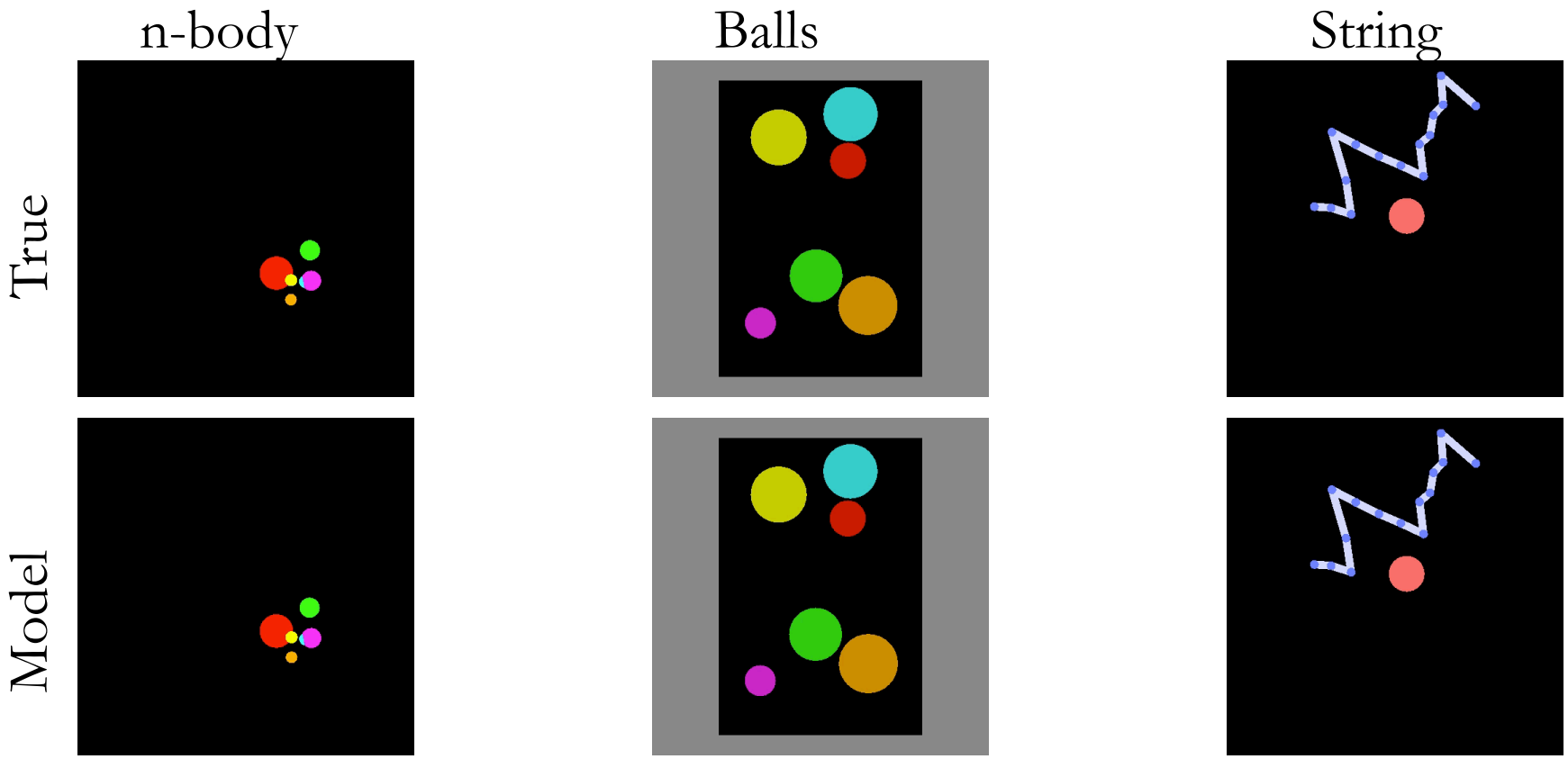




Nodes: masses

Edges: springs and rigid collisions

Battaglia et al., 2016, NeurIPS



Battaglia et al., 2016, NeurIPS

How do we do this?

1. Train a GN to learn the law of physics.
2. **Extract the equation from that GN.**

Symbolic Regression

Heuristics for Empirical Discovery

Pat Langley
Herbert A. Simon
Gary L. Bradshaw

CMU-RI-TR-84-14

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213 USA

June 25 1984

This research was supported by Contract N00014-84-K-0345 from the Information Sciences Division, Office of Naval Research. Approved for public release; distribution unlimited. Reproduction in whole or part is permitted for any purpose of the United States Government.

To appear in L. Bolc (Ed.), *Knowledge Based Learning Systems.* Berlin: Springer-Verlag, 1984.

"Used for finding regularities (constancies and trends) in data"

Data-Driven Approaches to Empirical Discovery

Pat Langley
Irvine Computational Intelligence Project
Department of Information & Computer Science
University of California, Irvine, CA 92717

Jan M. Zytkow†
Computer Science Department
Wichita State University
Wichita, Kansas 67208

Technical Report 88–24

October 31, 1988

† Currently on leave of absence at Department of Computer Science, George Mason University, Fairfax, VA 22030.

This research was supported in part by Contract N00014–84–K–0345 from the Information Sciences Division, Office of Naval Research.

We would like to thank Herbert Simon, Gary Bradshaw, Bruce Koehn, and Bernd Nordhausen, who have contributed significantly to the development of BACON, FAHRENHEIT, and IDS.

"Used for discovering bivariate equations only"

Symbolic Regression

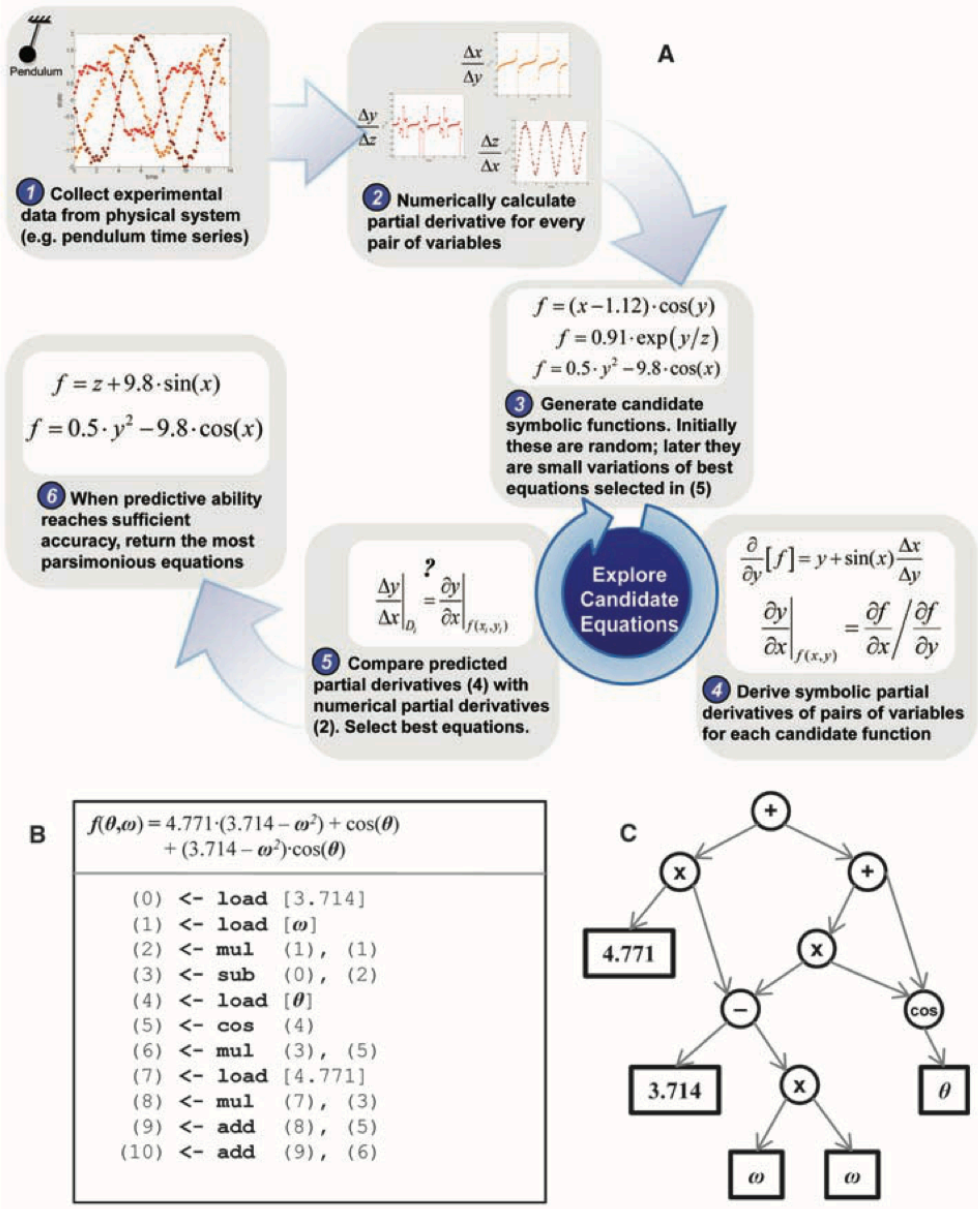
UNIVERSITY OF SUSSEX

UCL

Distilling Free-Form Natural Laws from Experimental Data

Michael Schmidt[1] and Hod Lipson[2,3]*

For centuries, scientists have attempted to identify and document analytical laws that underlie physical phenomena in nature. Despite the prevalence of computing power, the process of finding natural laws and their corresponding equations has resisted automation. A key challenge to finding analytic relations automatically is defining algorithmically what makes a correlation in observed data important and insightful. We propose a principle for the identification of nontriviality. We demonstrated this approach by automatically searching motion-tracking data captured from various physical systems, ranging from simple harmonic oscillators to chaotic double-pendula. Without any prior knowledge about physics, kinematics, or geometry, the algorithm discovered Hamiltonians, Lagrangians, and other laws of geometric and momentum conservation. The discovery rate accelerated as laws found for simpler systems were used to bootstrap explanations for more complex systems, gradually uncovering the "alphabet" used to describe those systems.



Symbolic Regression

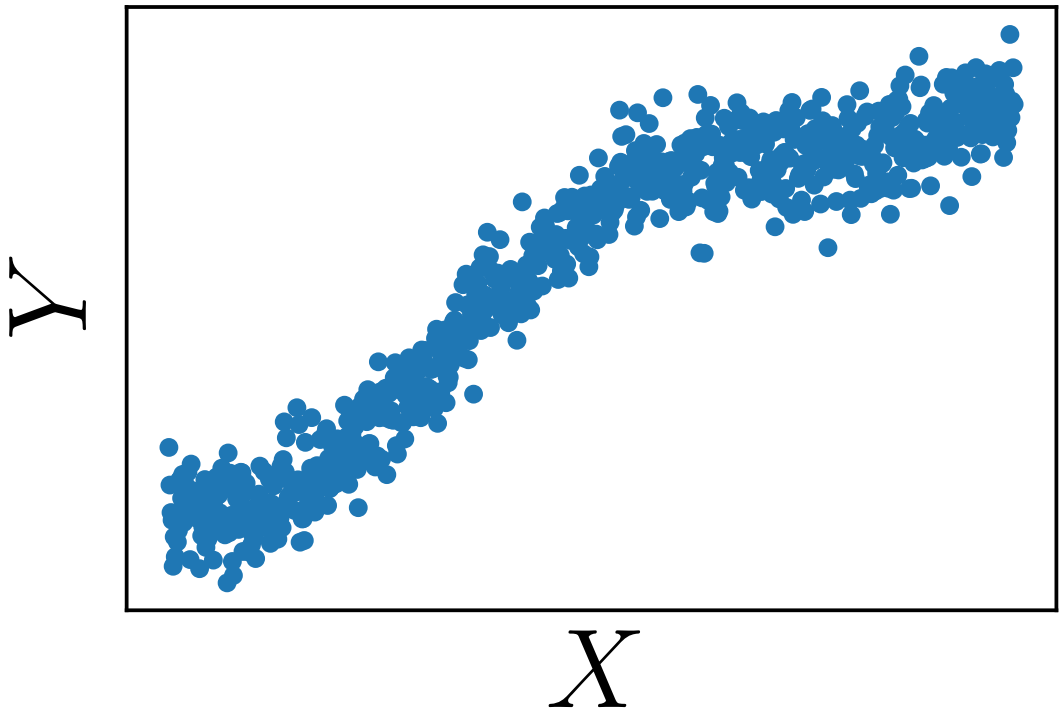
https://github.com/MilesCranmer/PySR

PySR: High-Performance Symbolic Regression in Python

PySR is built on an extremely optimized pure-Julia backend, and uses regularized evolution, simulated annealing, and gradient-free optimization to search for equations that fit your data.

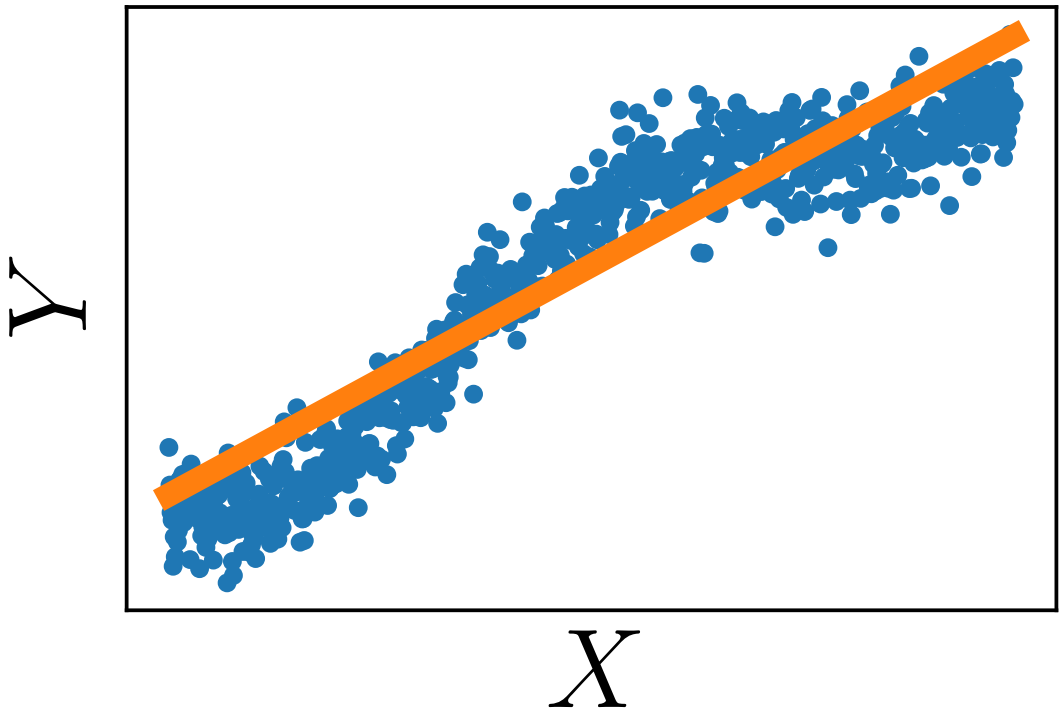
Docs	pip	conda	Stats
docs passing	pypi package 0.7.9	conda-forge v0.7.9	downloads 110k

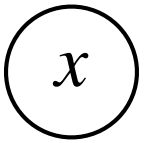
Symbolic Regression



Symbolic Regression

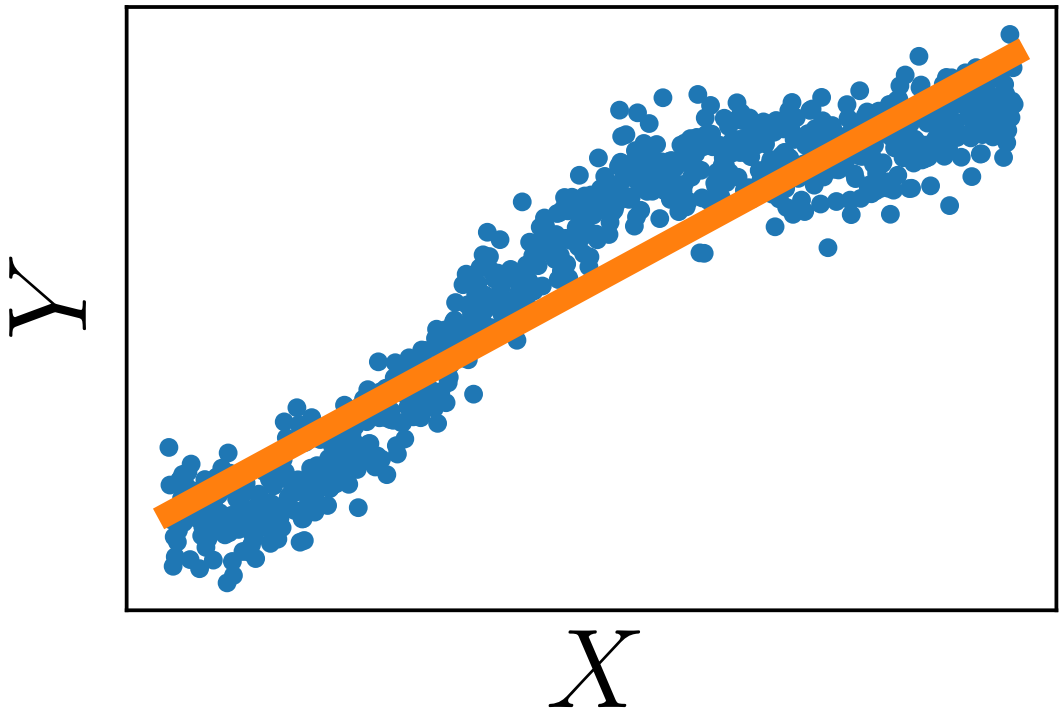
$$f(x) = x$$

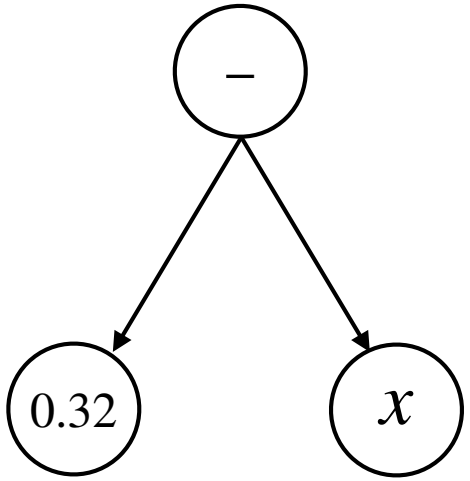




Symbolic Regression

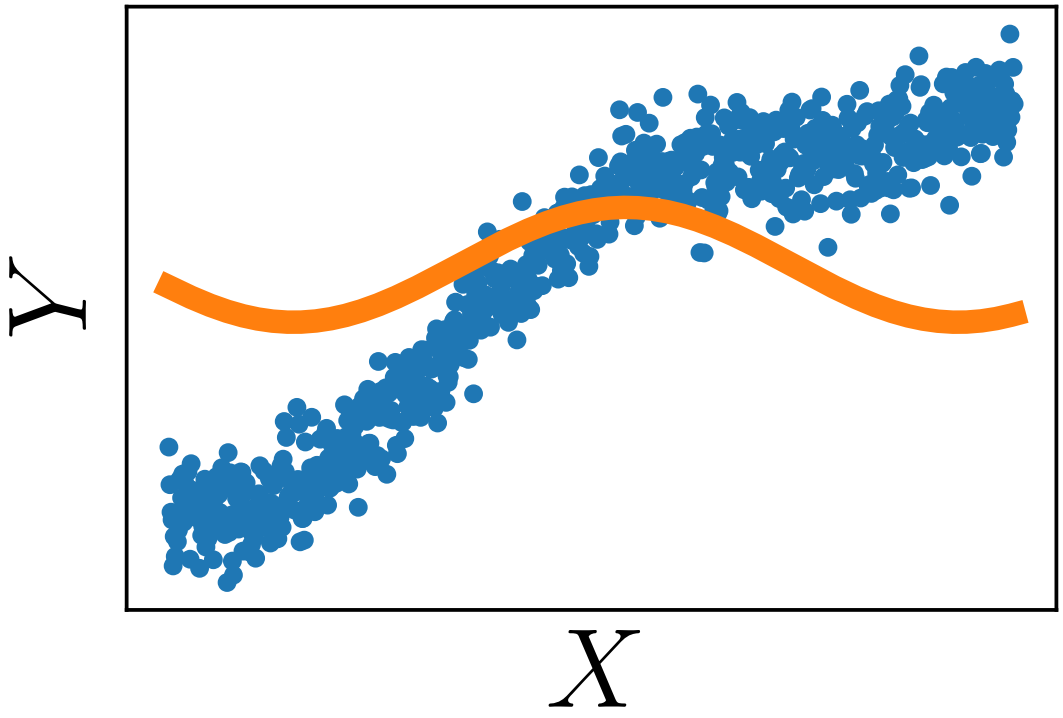
$$f(x) = x - 0.32$$

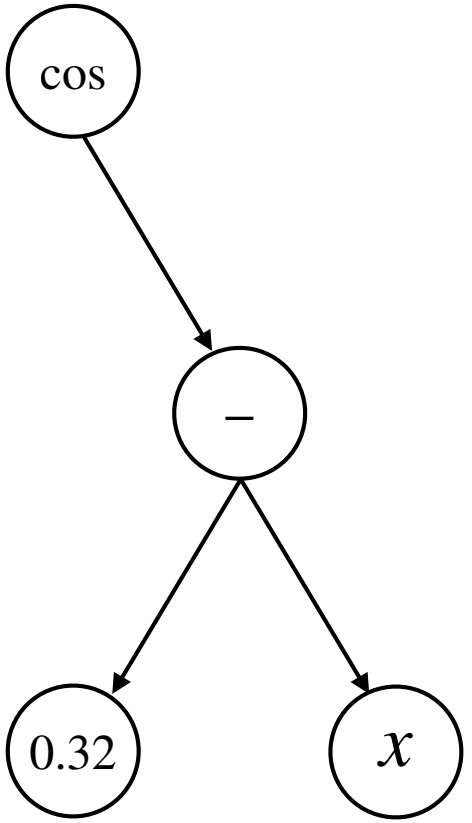




Symbolic Regression

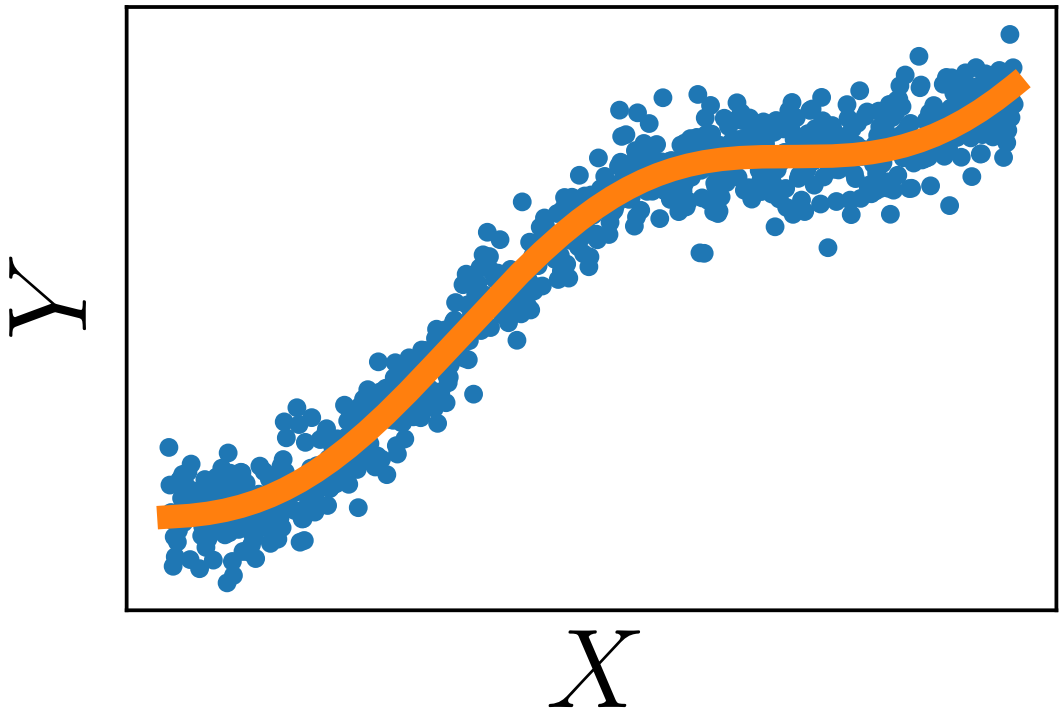
$$f(x) = \cos(x - 0.32)$$

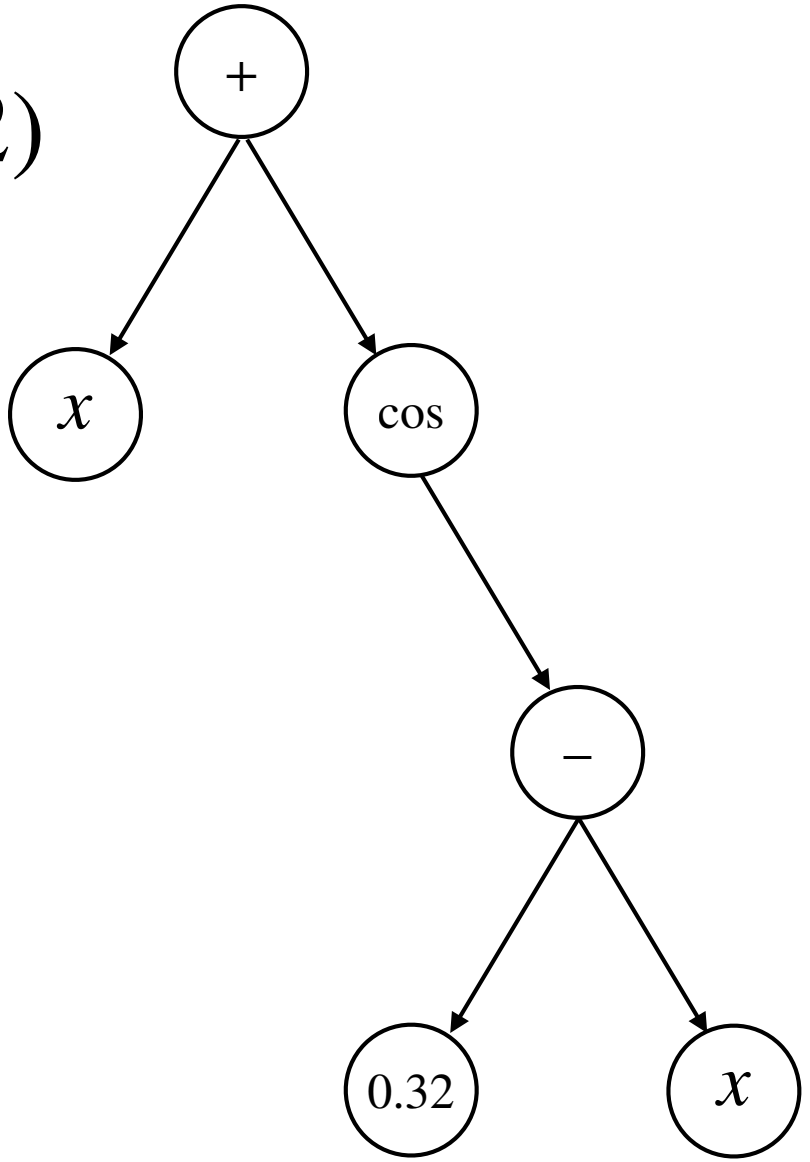




Symbolic Regression

$$f(x) = x + \cos(x - 0.32)$$



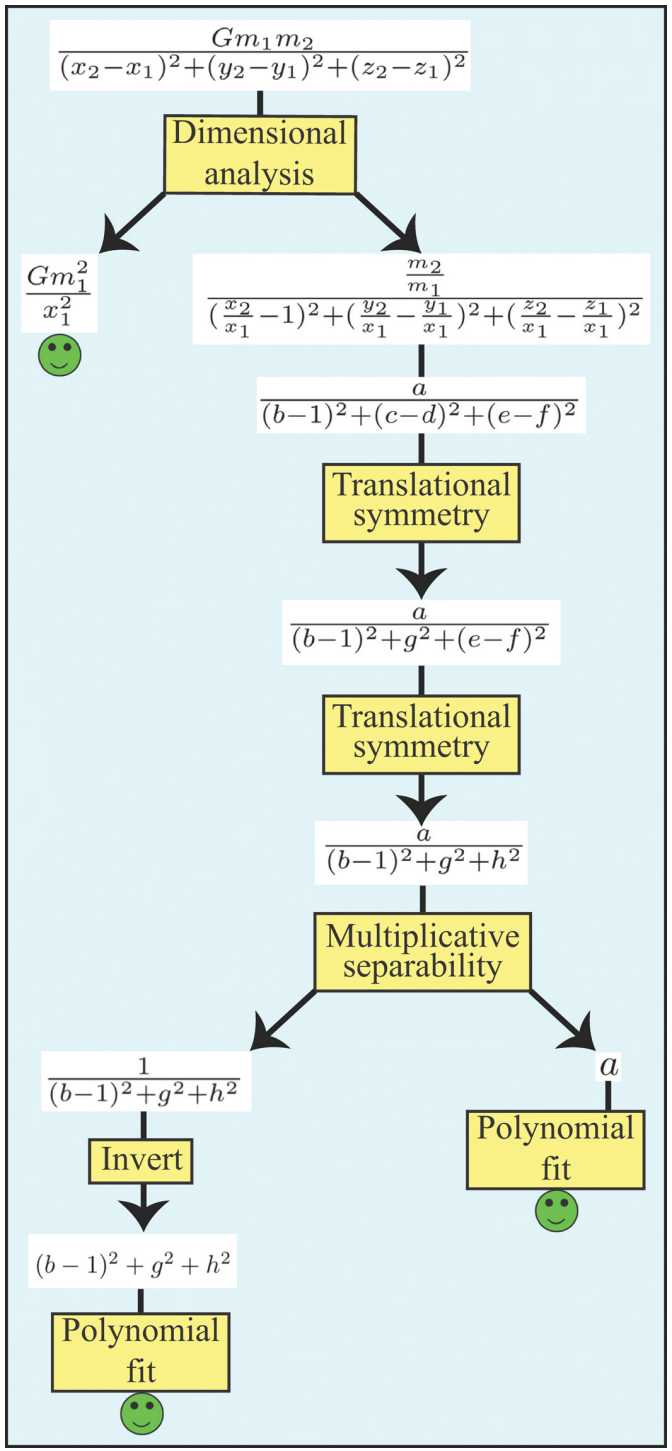


UNIVERSITY OF SUSSEX

UCL

Symbolic regression

- PySR repeats this process iteratively
- It comes up with a set of “candidate equations”



UNIVERSITY OF SUSSEX

UCL

Symbolic Regression

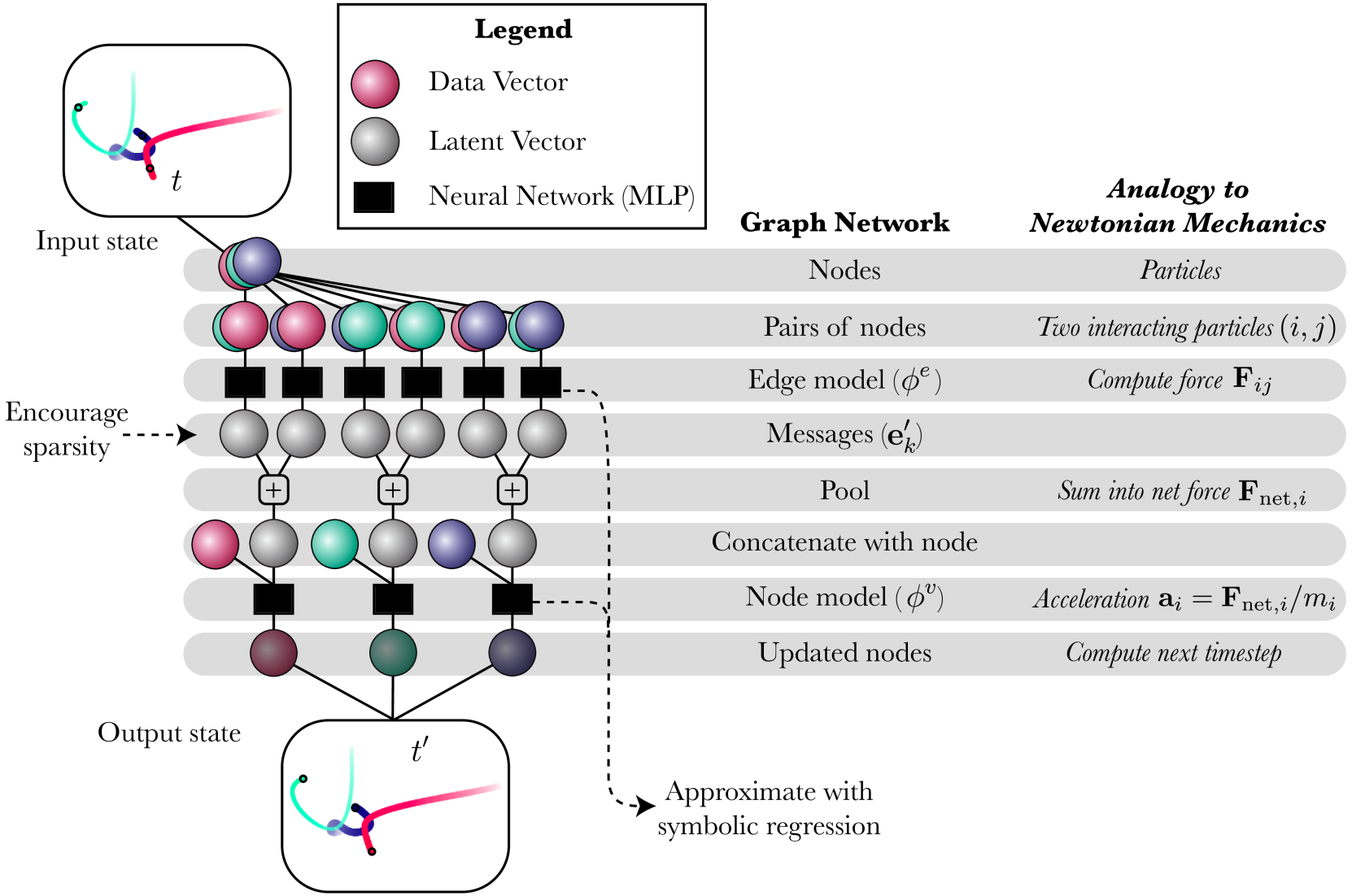
AI Feynman: a Physics-Inspired Method for Symbolic Regression

Silviu-Marian Udrescu, Max Tegmark*
Dept. of Physics & Center for Brains, Minds & Machines, Massachusetts Institute of Technology, Cambridge, MA 02139; sudrescu@mit.edu and Theiss Research, La Jolla, CA 92037, USA

(Dated: Published in *Science Advances*, 6:eaay2631, April 15, 2020)

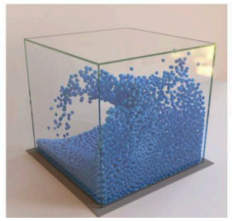
A core challenge for both physics and artificial intelligence (AI) is *symbolic regression*: finding a symbolic expression that matches data from an unknown function. Although this problem is likely to be NP-hard in principle, functions of practical interest often exhibit symmetries, separability, compositionality and other simplifying properties. In this spirit, we develop a recursive multidimensional symbolic regression algorithm that combines neural network fitting with a suite of physics-inspired techniques. We apply it to 100 equations from the Feynman Lectures on Physics, and it discovers all of them, while previous publicly available software cracks only 71; for a more difficult physics-based test set, we improve the state of the art success rate from 15% to 90%.

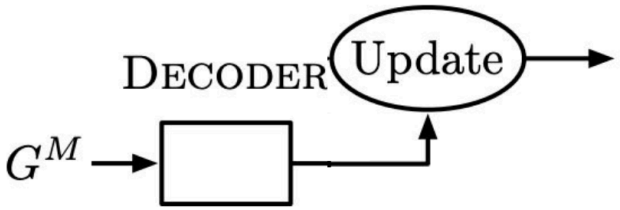
Summary of the algorithm



Cranmer et al., NeurIPS 2020

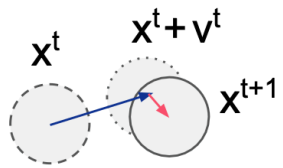
Decoder and update





- predict acceleration !
  - $v^{t+1} = v^t + \text{decode}(G)$
  - $x^{t+1} = x^t + v^{t+1}$

"Acceleration" (Euler integrator with dt=1)

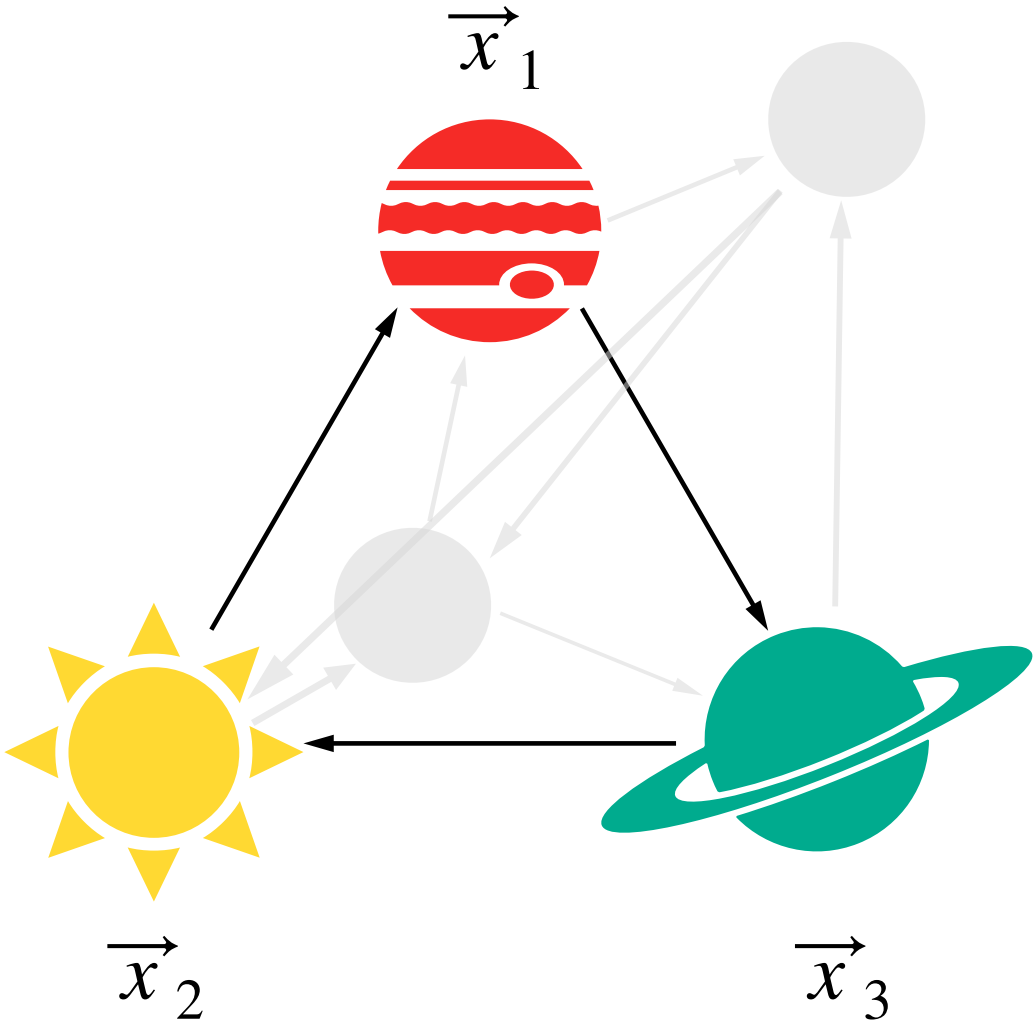


Easy to predict static dynamics

Easy to predict inertial dynamics → Good prior

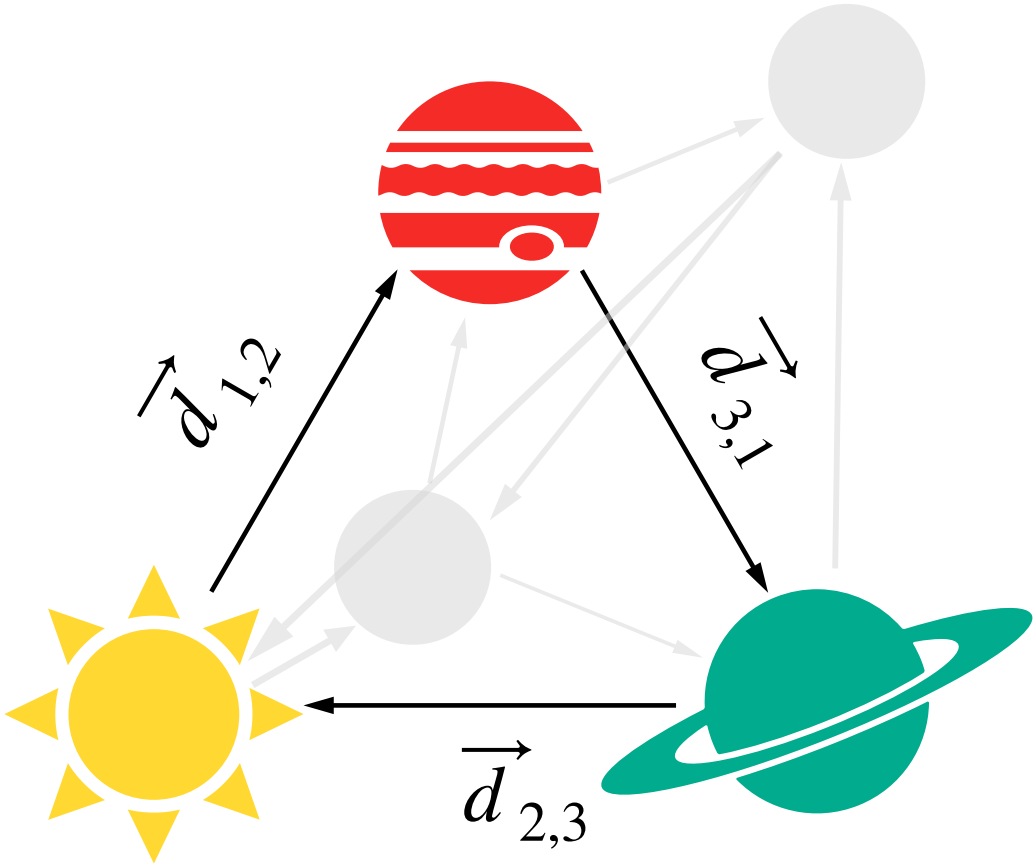
See Alvaro's talk yesterday!

1. **Our inputs are the positions of the bodies**



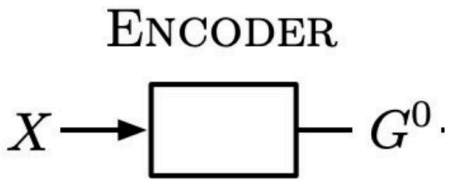
1. Our inputs are the positions of then bodies

2. **They are converted into pairwise distances**



Inductive biases

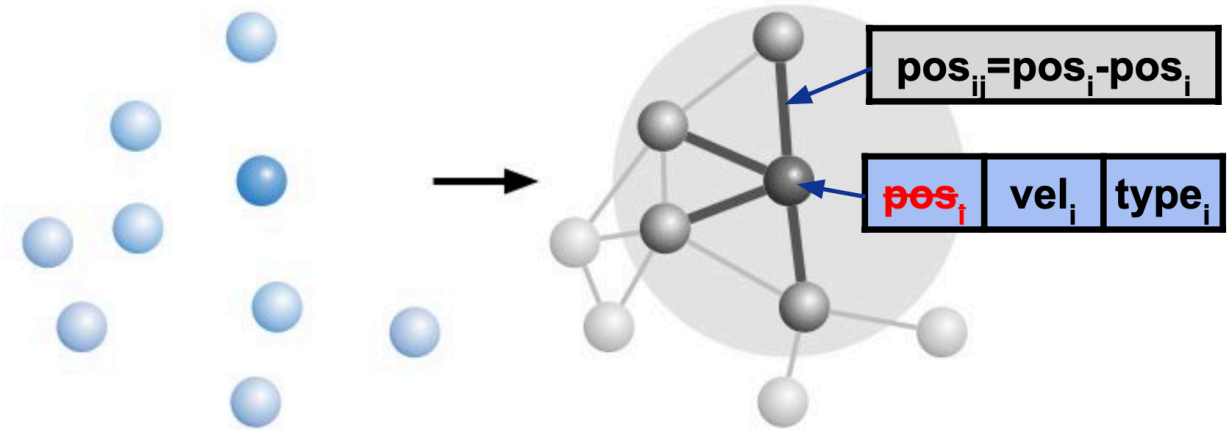
Encoding Graphs



Transform the inputs into a graph

- Edges: Mesh edges (for meshes) or proximity-based (for particles)

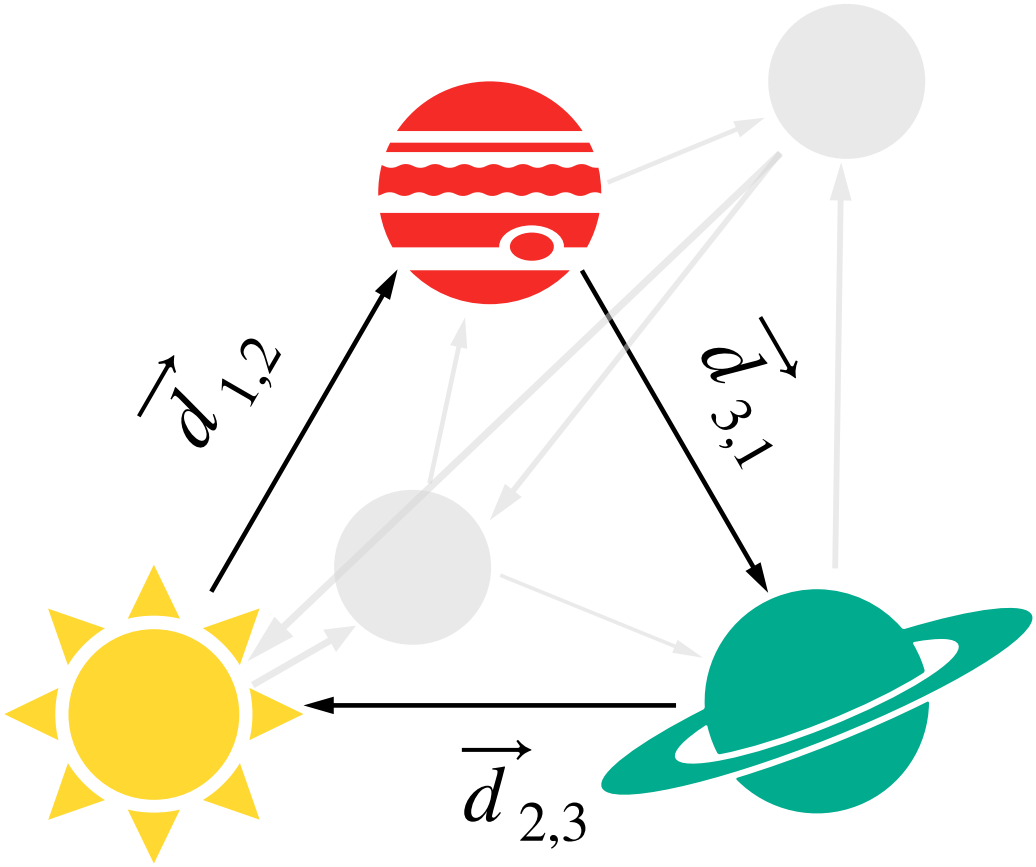
Spatial equivariance



See Alvaro's talk yesterday!

1. Our inputs are the positions of then bodies

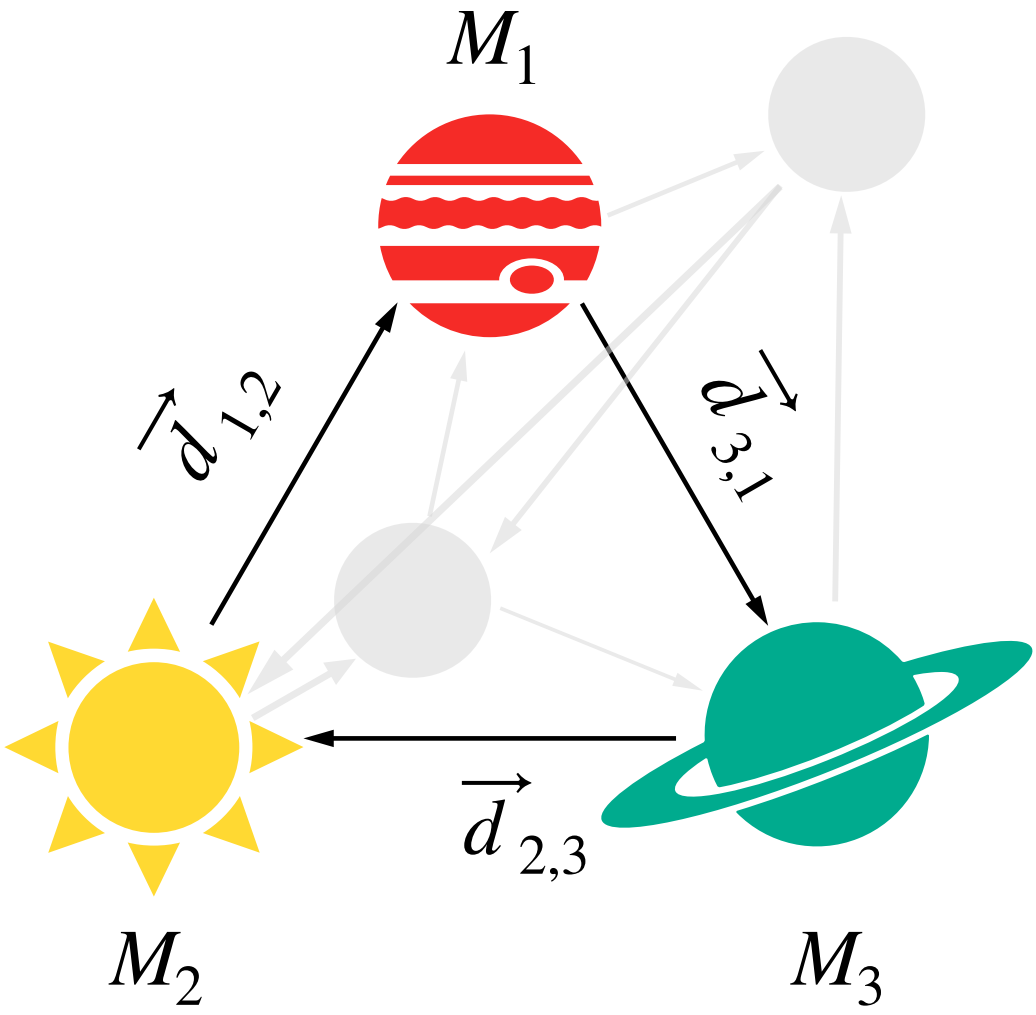
2. **They are converted into pairwise distances**



1. Our inputs are the positions of the bodies

2. They are converted into pairwise distances

3. **Our model tries to guess a mass for each body**

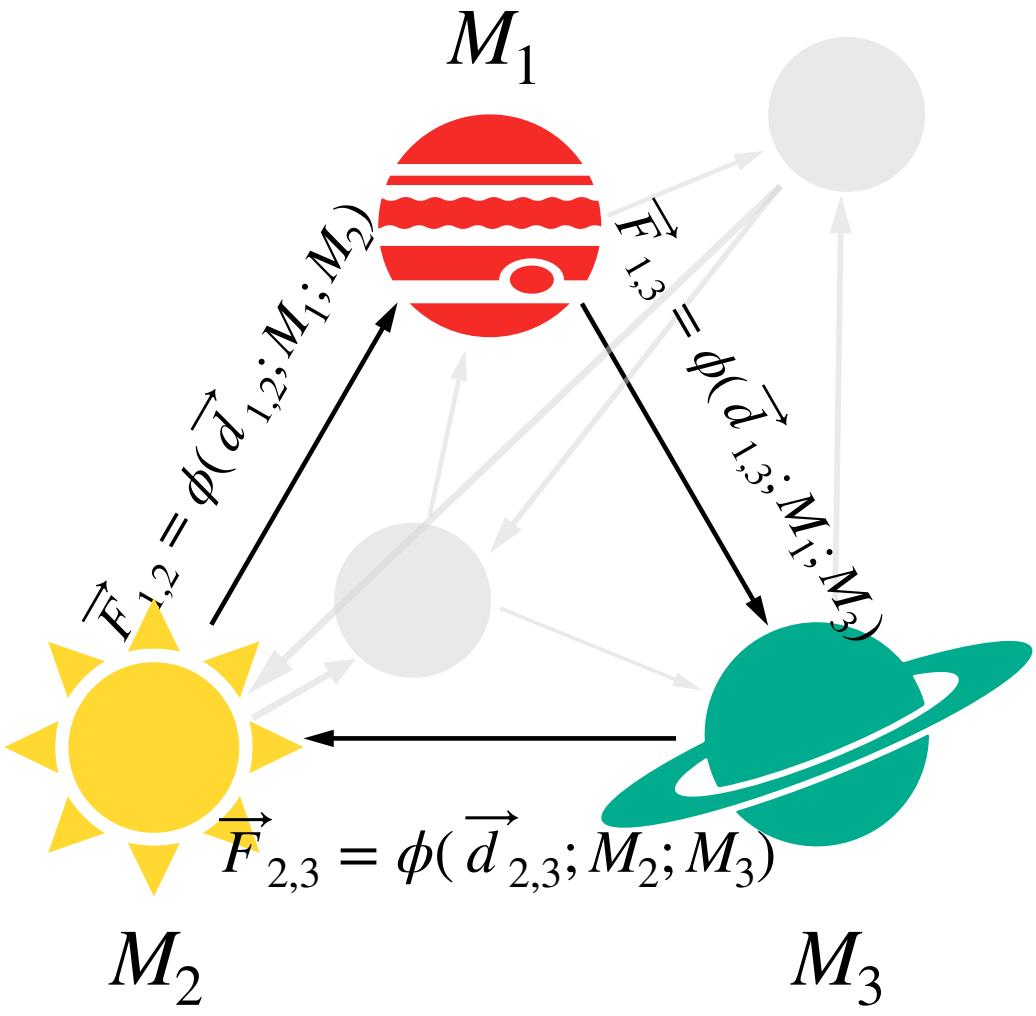


1. Our inputs are the positions of the bodies

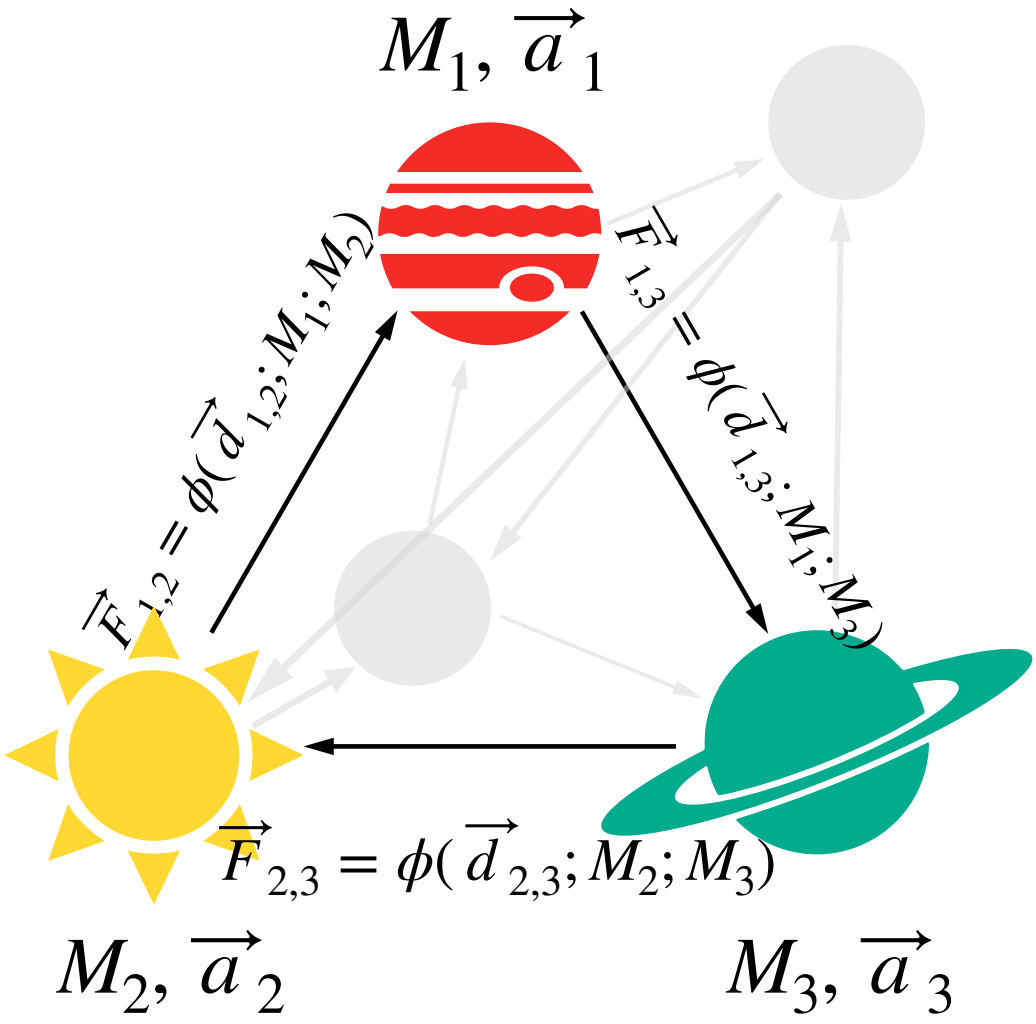
2. They are converted into pairwise distances

3. Our model tries to guess a mass for each body

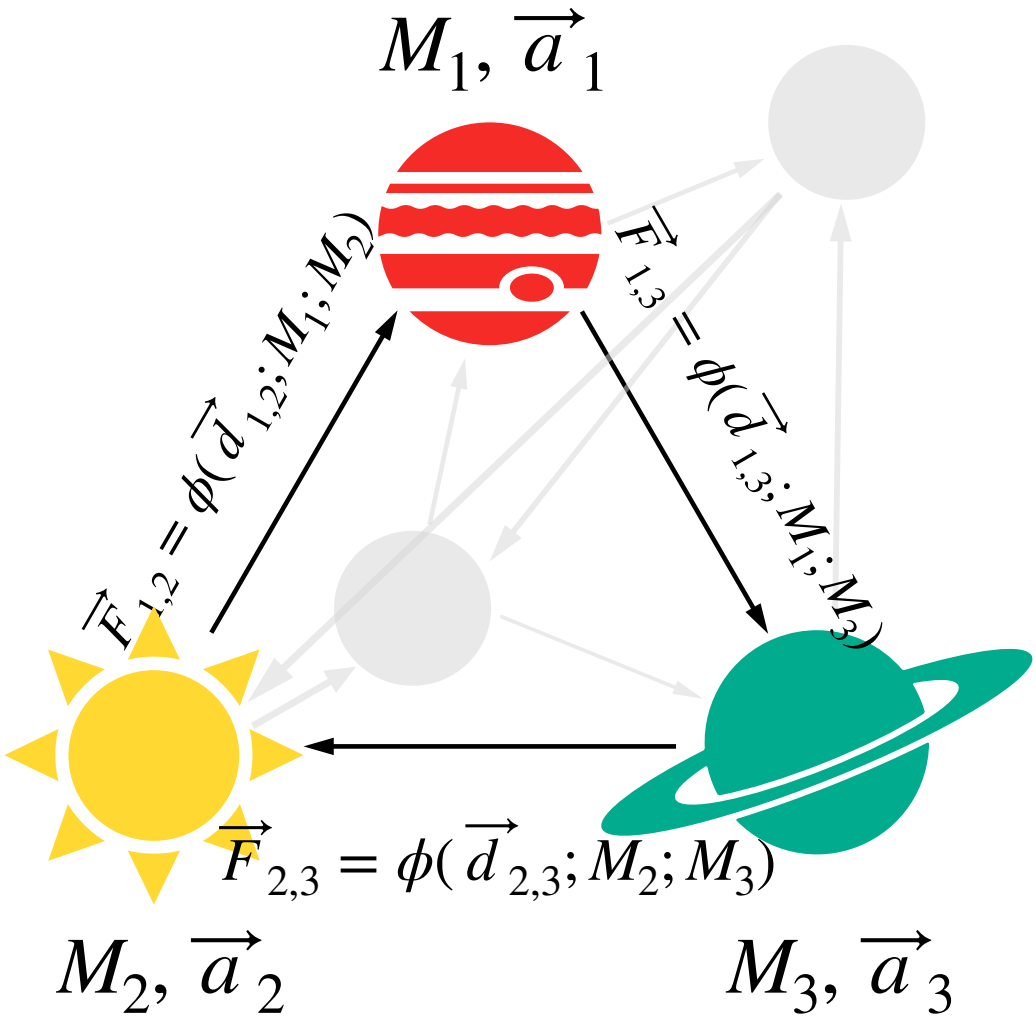
4. **It then also guesses a force, that is a function of distance and masses**



1. Positions
2. Distances
3. Mass guess
4. It then also guesses a force, that is a function of distance and masses
5. **Using Newton's laws of motion ($\sum \vec{F} = M\vec{a}$) it converts the forces into accelerations**

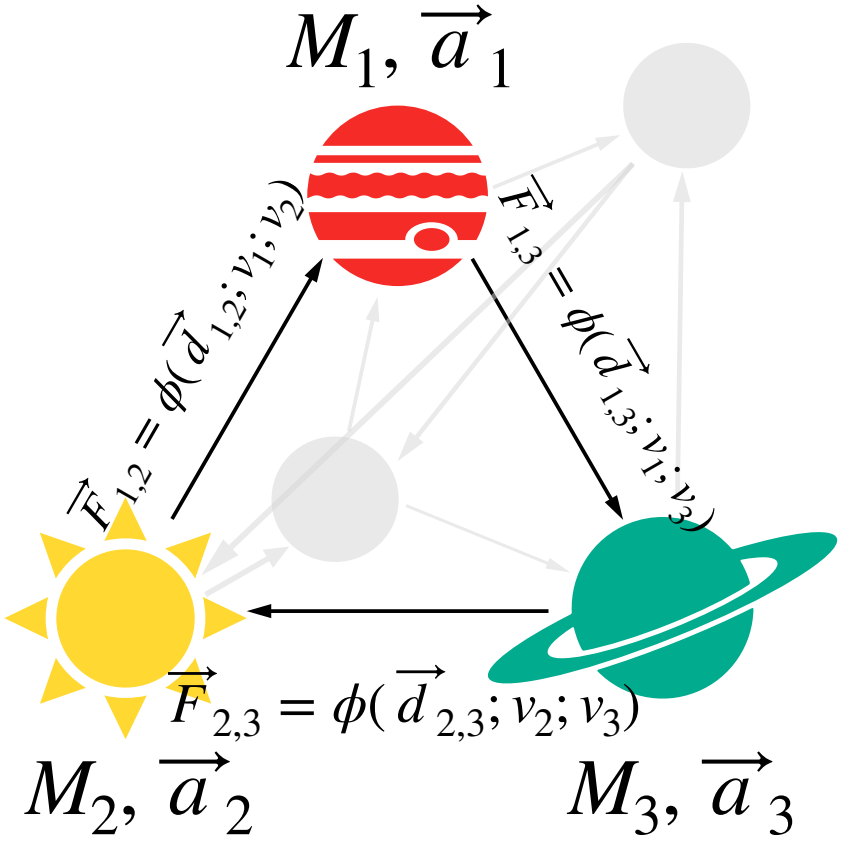


1. Positions
2. Distances
3. Mass guess
4. Force guess
5. Acceleration
6. **Finally, it compares this predicted acceleration, with the true acceleration from the data**



1. Positions
2. Distances
3. Mass guess
4. Force guess
5. Predicted acceleration
6. Minimize

$$\left| \vec{a}(\text{pred}) - \vec{a}(\text{true}) \right|^2$$



UNIVERSITY OF SUSSEX

UCL

Learned Simulator

UNIVERSITY OF SUSSEX

UCL

Data

- We use data from NASA's Ephymeris system.
- We extract orbits for bodies with $M > 10^{18}$ kg, leaving 31 bodies.
- We use data from Jan 1980 - Jan 2010 for training, and Jan 2010 - Jan 2013 for validation.
- We extract positions and velocities in Cartesian coordinates, in the solar system barycentre frame.

UNIVERSITY OF SUSSEX

UCL

Graph Network

- Our GN uses 3 hidden layers of 128 neurons each.
- It has an extra trainable scalar per node.
- We use “tanh” activation functions.
- Our loss function is a normalised MSE.

$$\text{Loss} = \sum \frac{(A - g(V, E; \theta))^2}{A^2}$$

UNIVERSITY OF SUSSEX

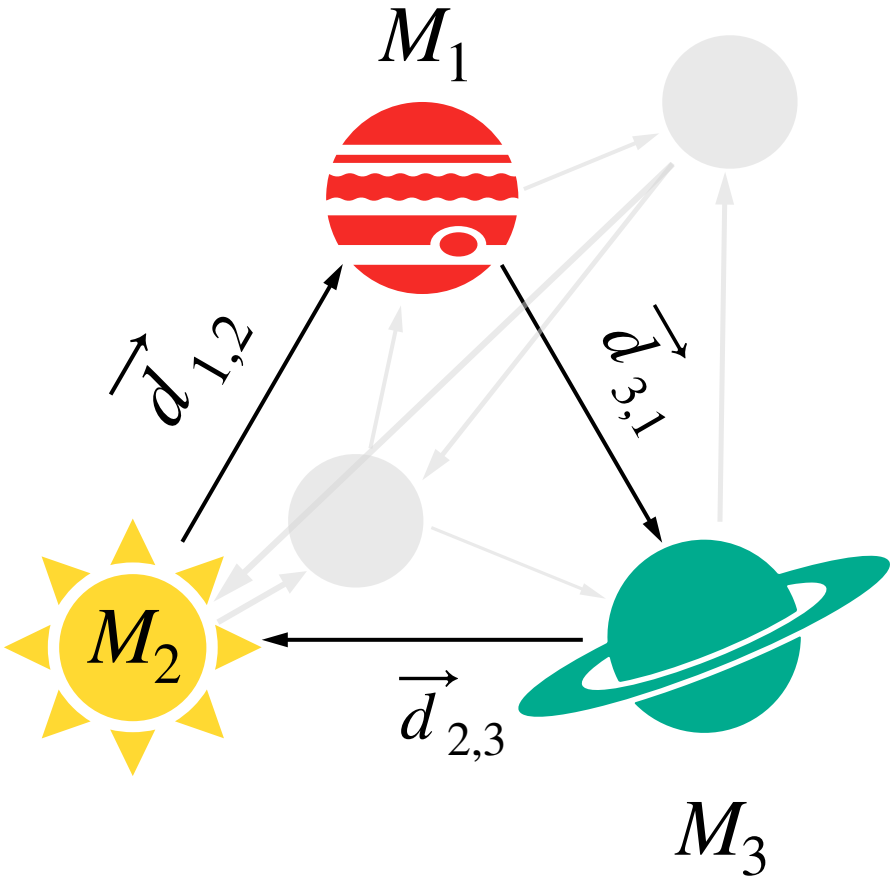
UCL

Graph Network

- We use data in log scale to deal with large dynamic ranges.
- We use 3D rotations of the system for data augmentation.
- We add random noise during training.

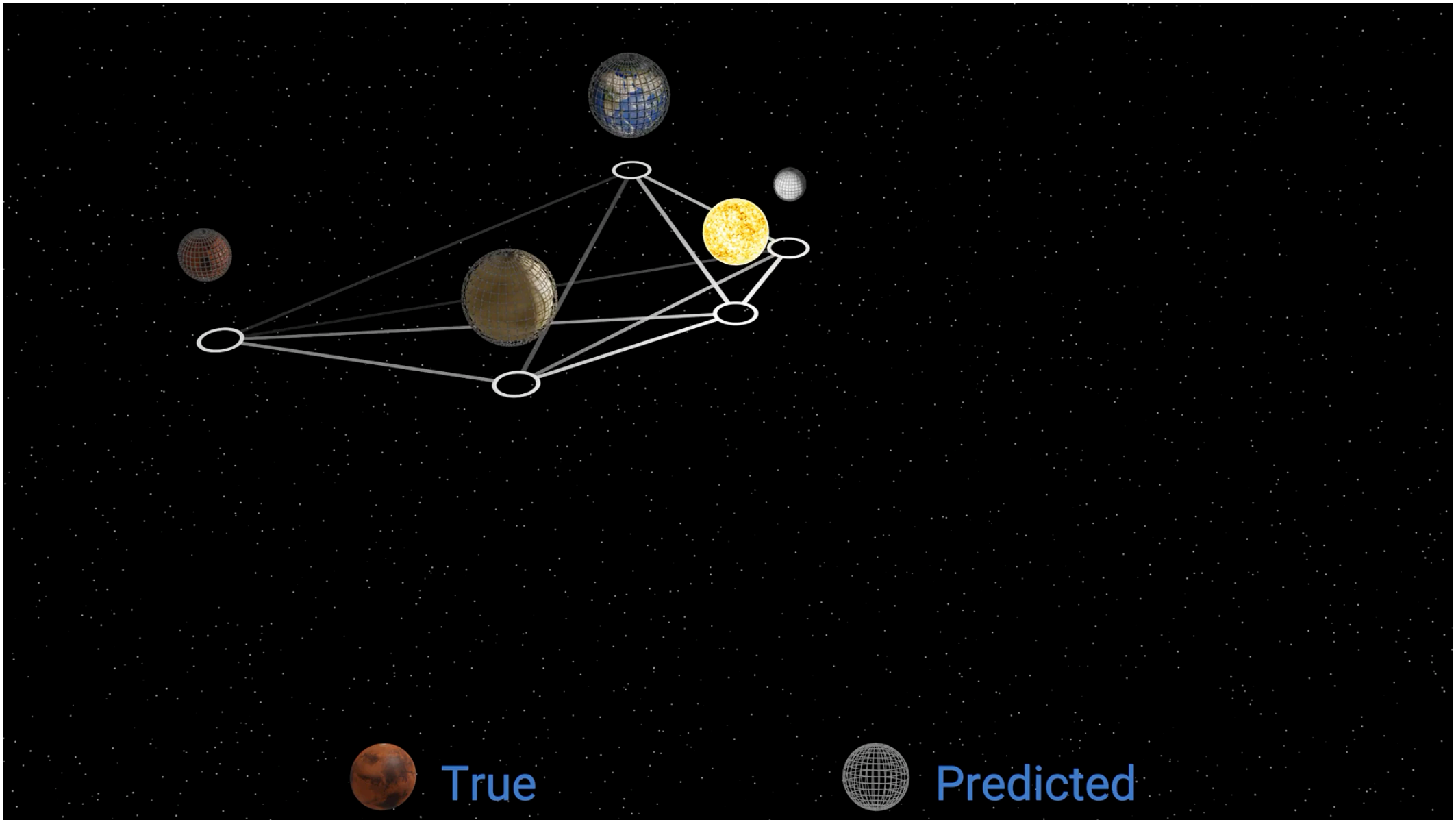
Inductive biases

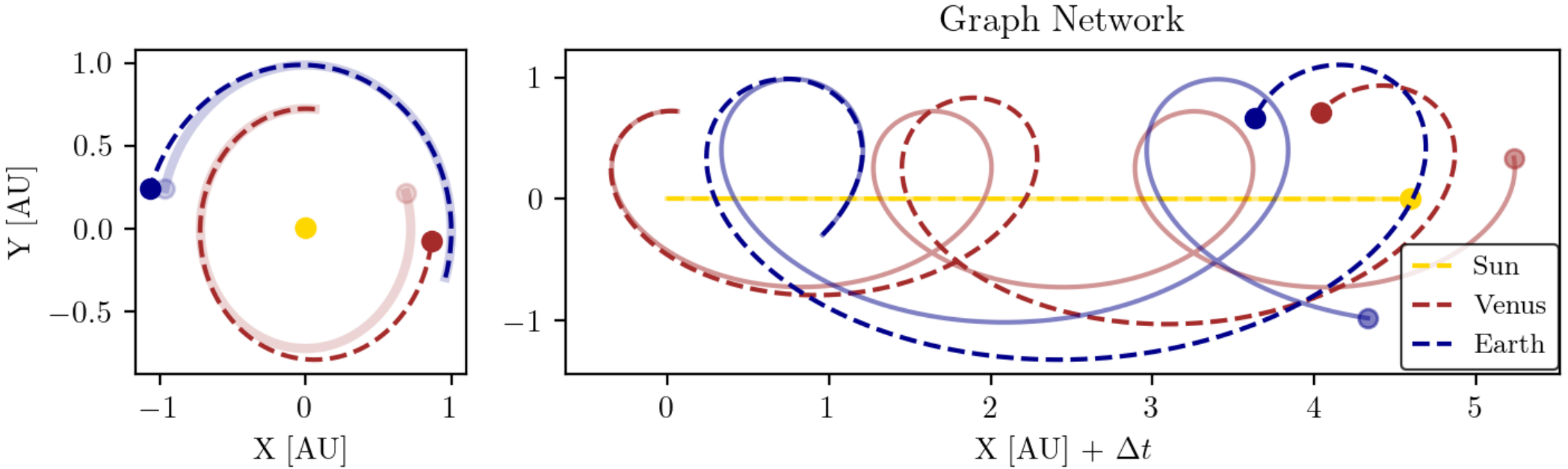
- Translational symmetry
- Rotational symmetry
- Newton's second law
$$\sum \vec{F} = M\vec{a}$$
- Newton's third law
$$\vec{F}_{ij} = -\vec{F}_{ji}$$
- Choice of reference frame, units, etc.



UNIVERSITY OF SUSSEX

UCL

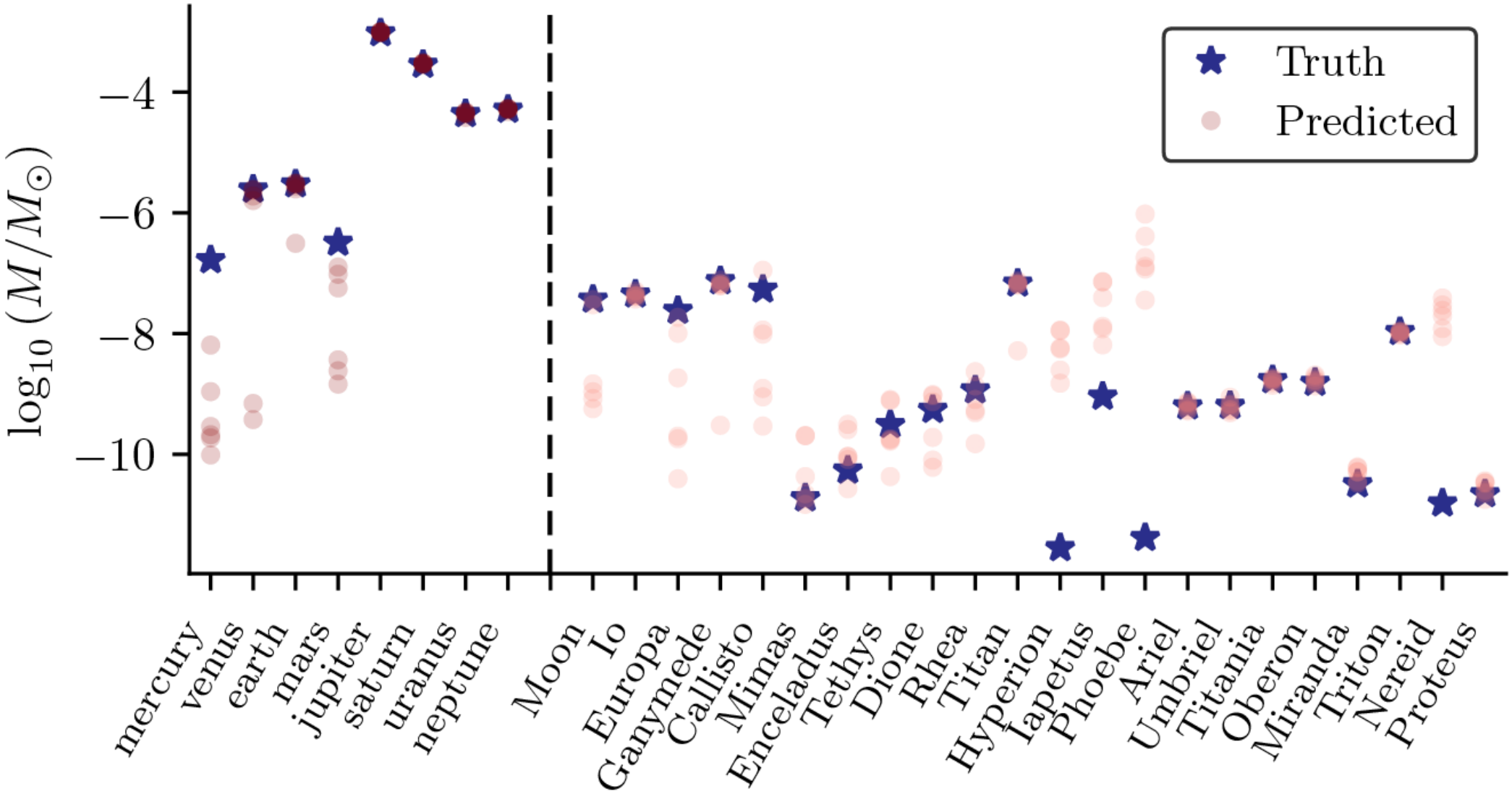



Graph Network
1.0
0.5
0.0
−0.5
Y [AU]
−1
0
1
X [AU]
1
0
−1
0
1
2
3
4
5
X [AU] + Δt
Sun
Venus
Earth

UNIVERSITY OF SUSSEX

UCL

Predicted masses



UNIVERSITY OF SUSSEX

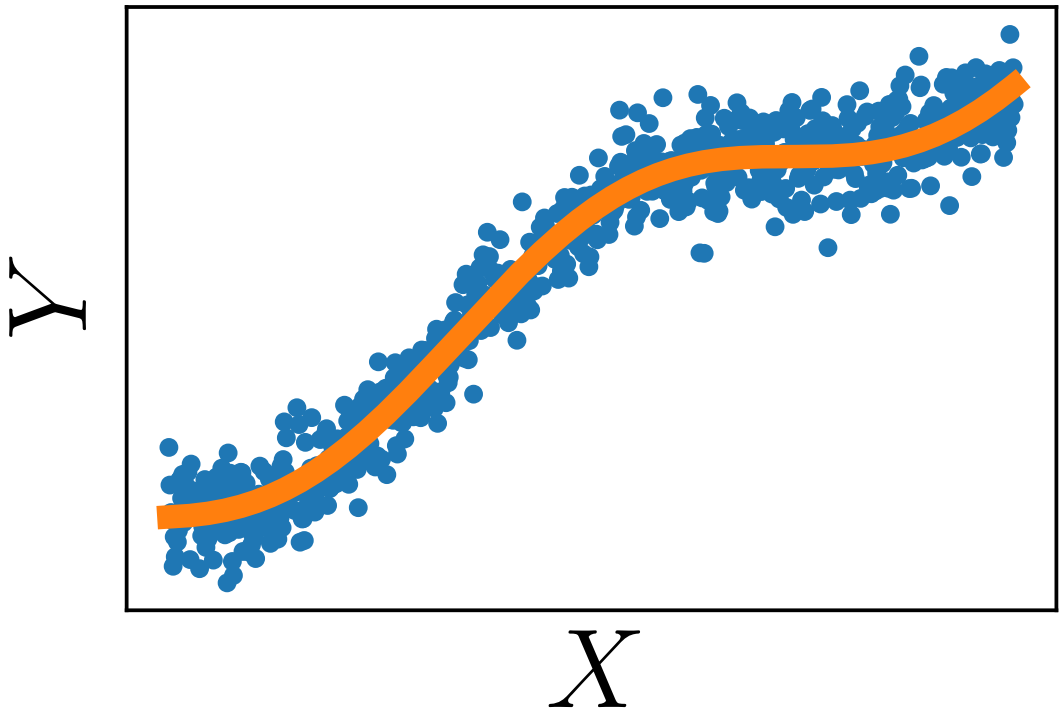
UCL

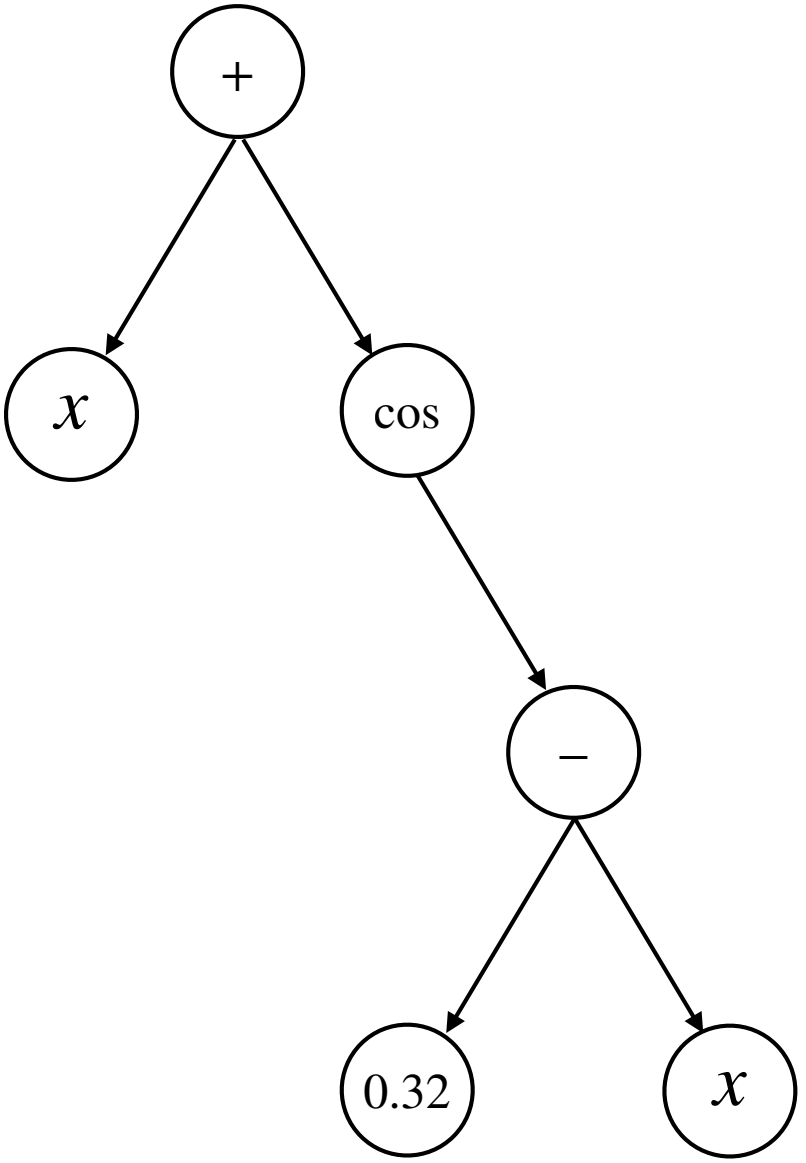
Symbolic regression

Symbolic regression

- We select a dataset of 500 data points, not used during training.
- Each point contains $x = (v_{r_k}, v_{s_k}, \overrightarrow{e}_k)$ (masses and distance for a pair of points) as **inputs**.
- It also contains $y = \vec{f}_{\mathrm{GN}}(x)$ the learned interaction as **output**.

Symbolic Regression





Symbolic regression

- The algorithm applies Occam's Razor by balancing accuracy and equation complexity.
- It does so by calculating a "score"

$$\text{Score} = \frac{\delta\ \text{Accuracy}}{\delta\ \text{Complexity}}$$

- We are currently working in a more sophisticated Occam's razor method

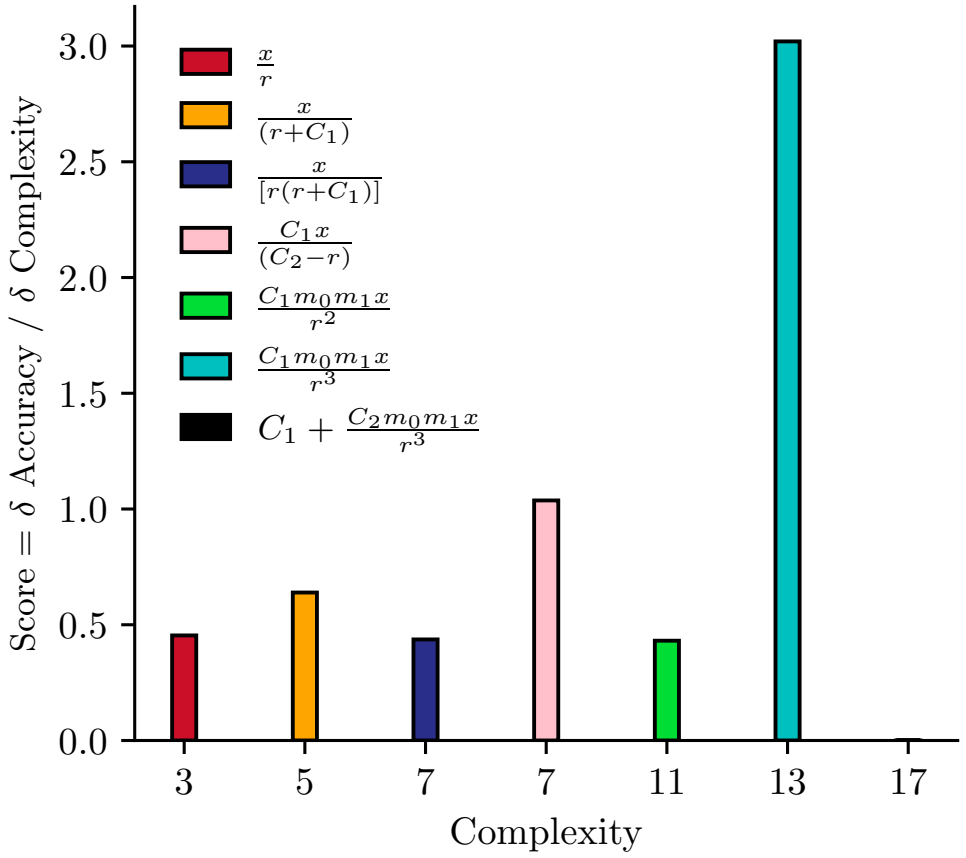
Symbolic regression

- Out loss is an MSE between $\vec{f}_{\mathrm{GN}}(x)$ and the proposed equation $\vec{f}_{\mathrm{SR}}(x, \theta)$.
- The allowed operators are $\{+, -, /, *, \mathrm{pow}, \log, \exp\}$.
- The maximum complexity is 40.

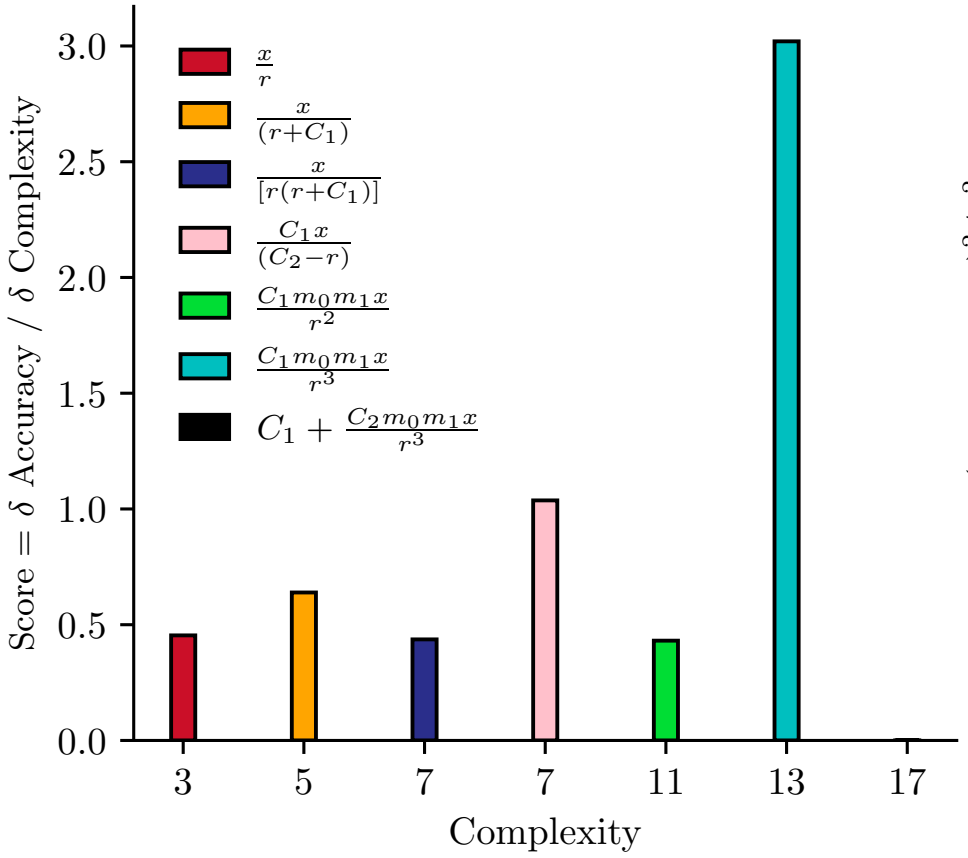
UNIVERSITY OF SUSSEX

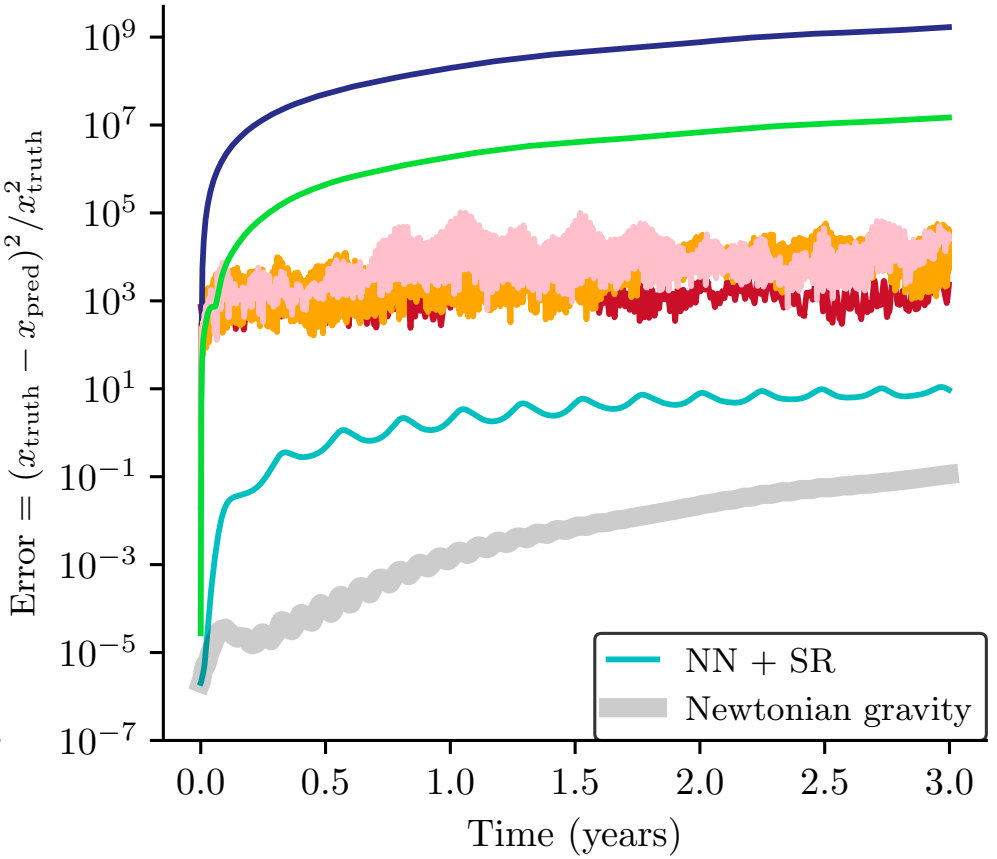
UCL

Predicted equations



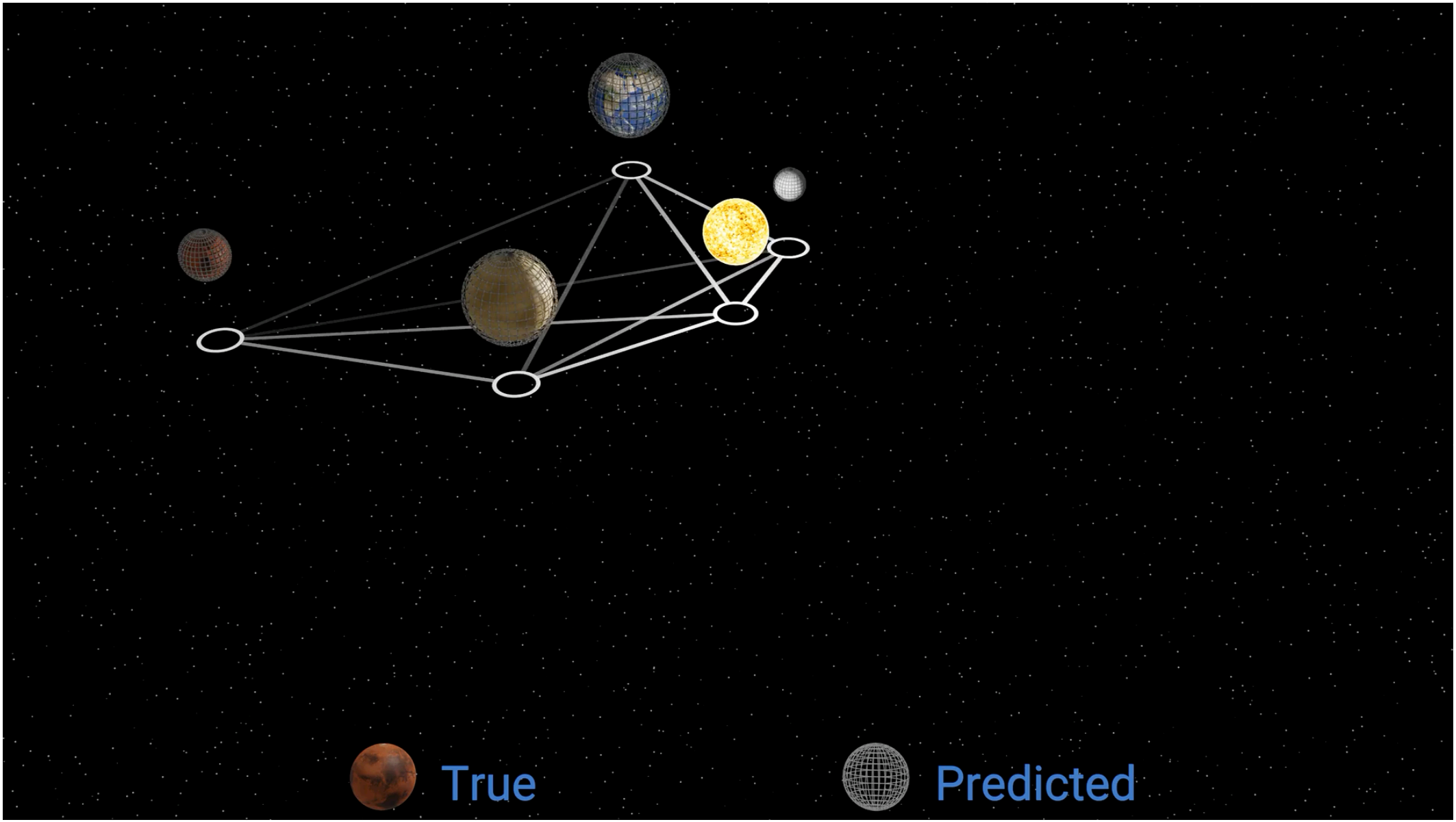
Predicted equations

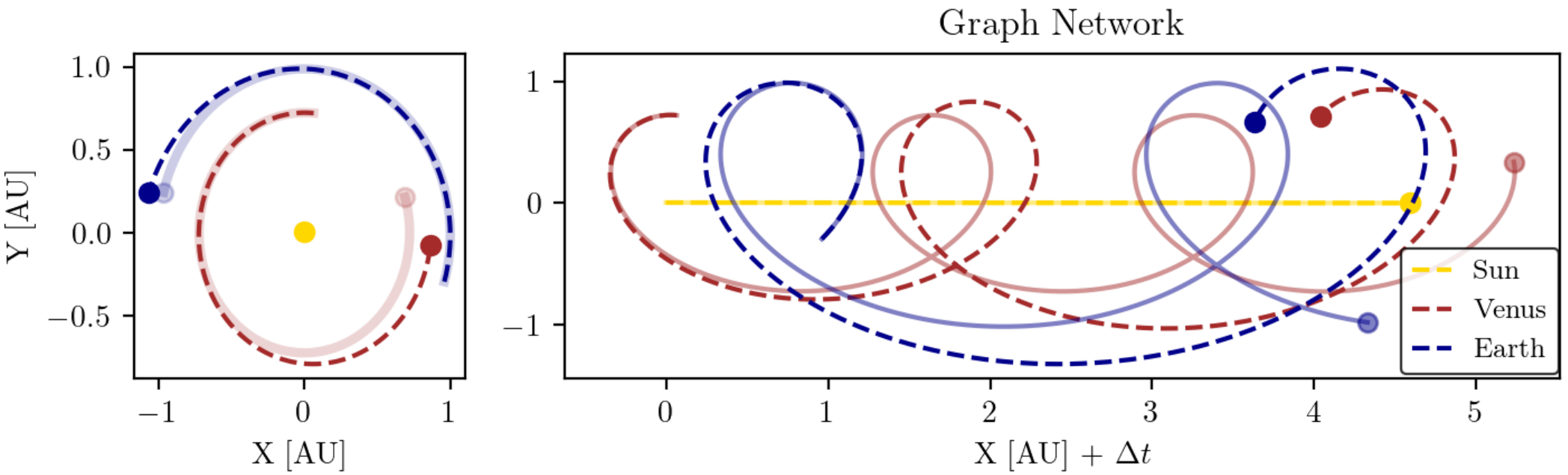


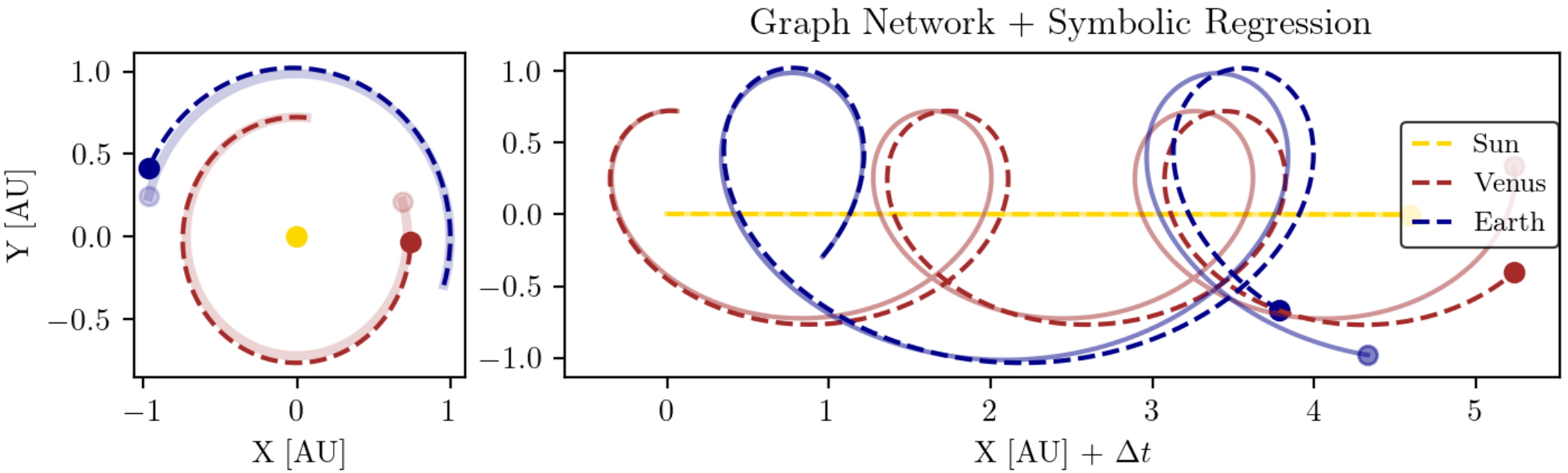


UNIVERSITY OF SUSSEX

UCL




Graph Network
Y [AU]
X [AU]
X [AU] + Δt
Sun
Venus
Earth


Graph Network + Symbolic Regression
Y [AU]
X [AU]
X [AU] + Δt
Sun
Venus
Earth

Final results

1. Positions
2. Distances
3. Mass guess
4. ~~Force guess~~ **Learned equation**
5. Predicted acceleration
6. Minimize

$$\left| \vec{a}(\text{pred}) - \vec{a}(\text{true}) \right|^2$$

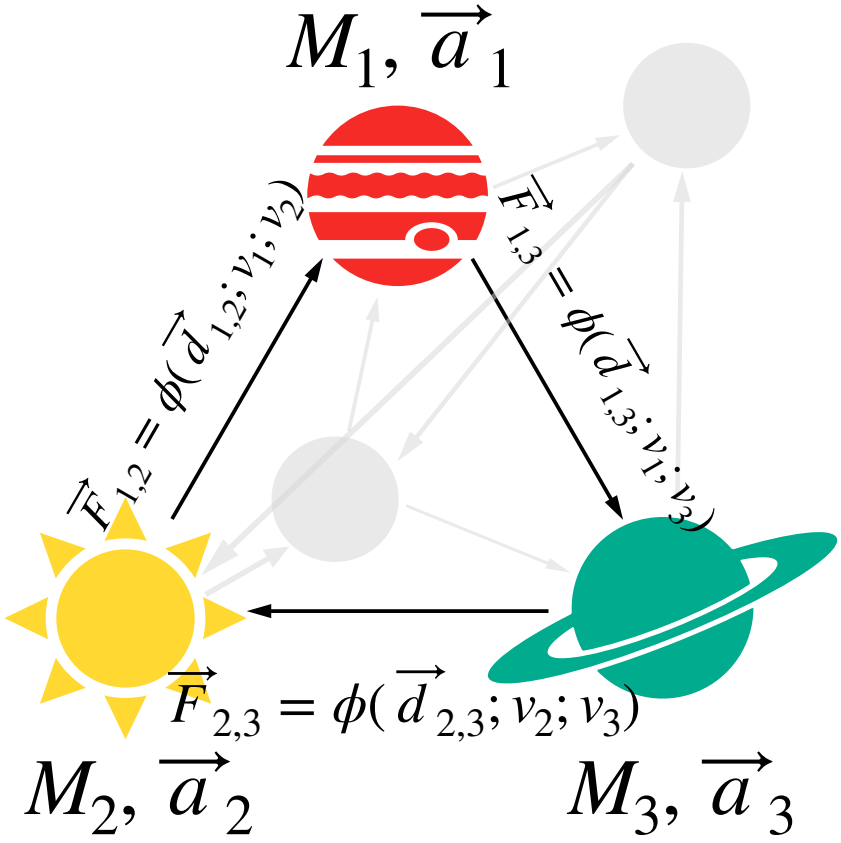


Fig. 4A: Graph network + symbolic regression

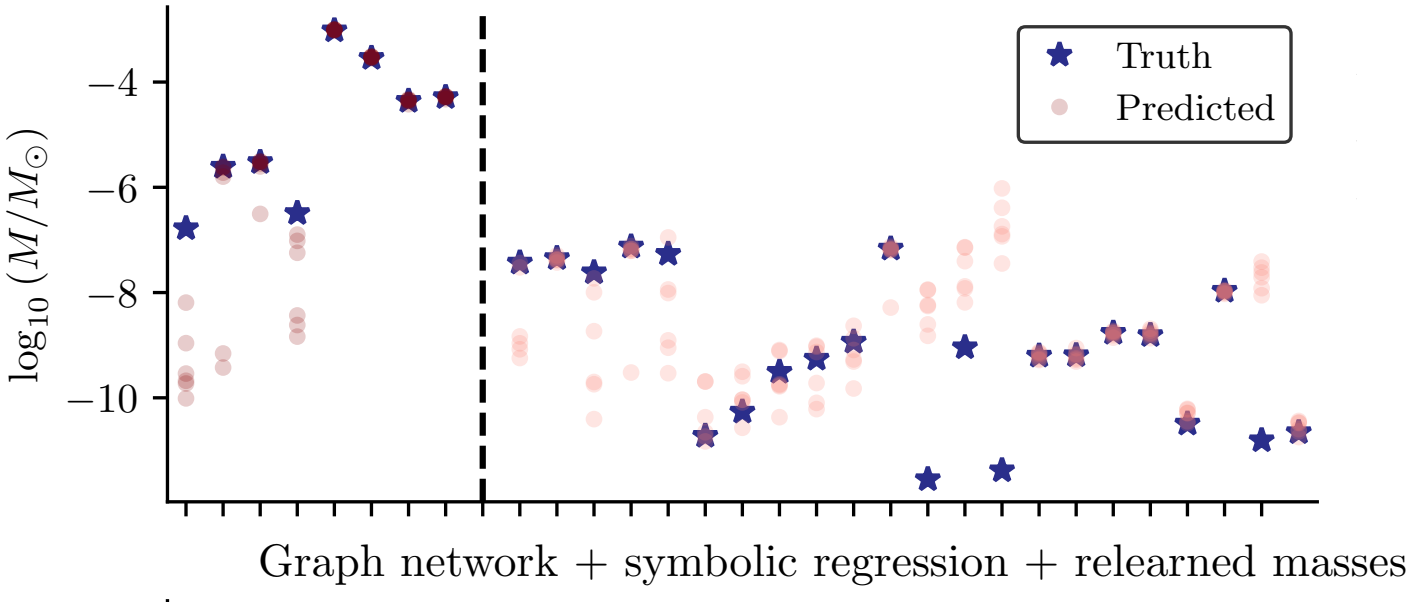
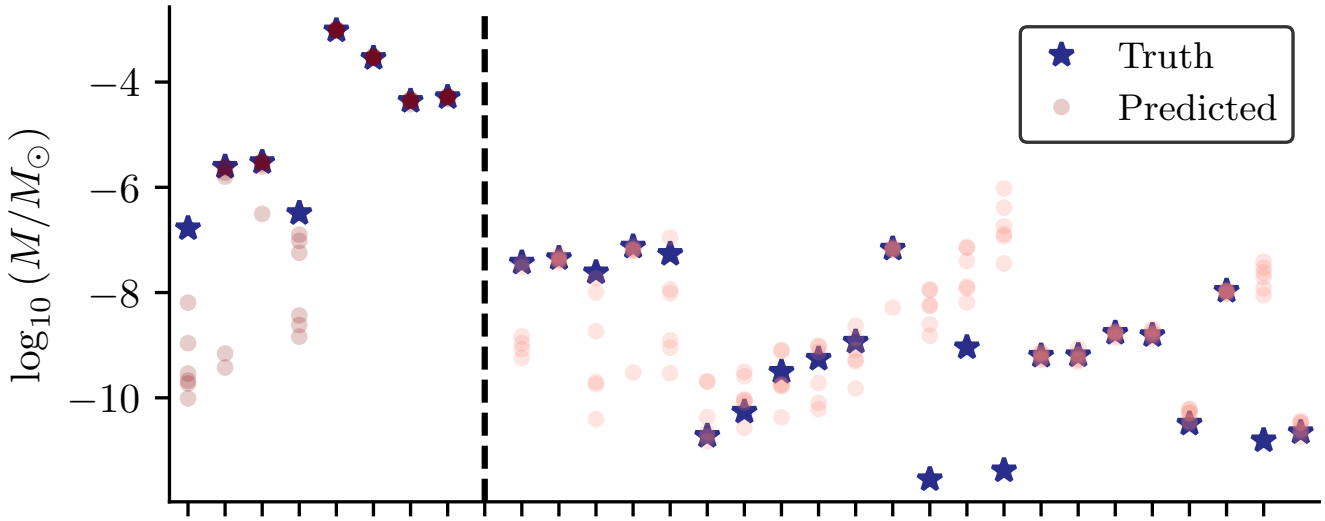
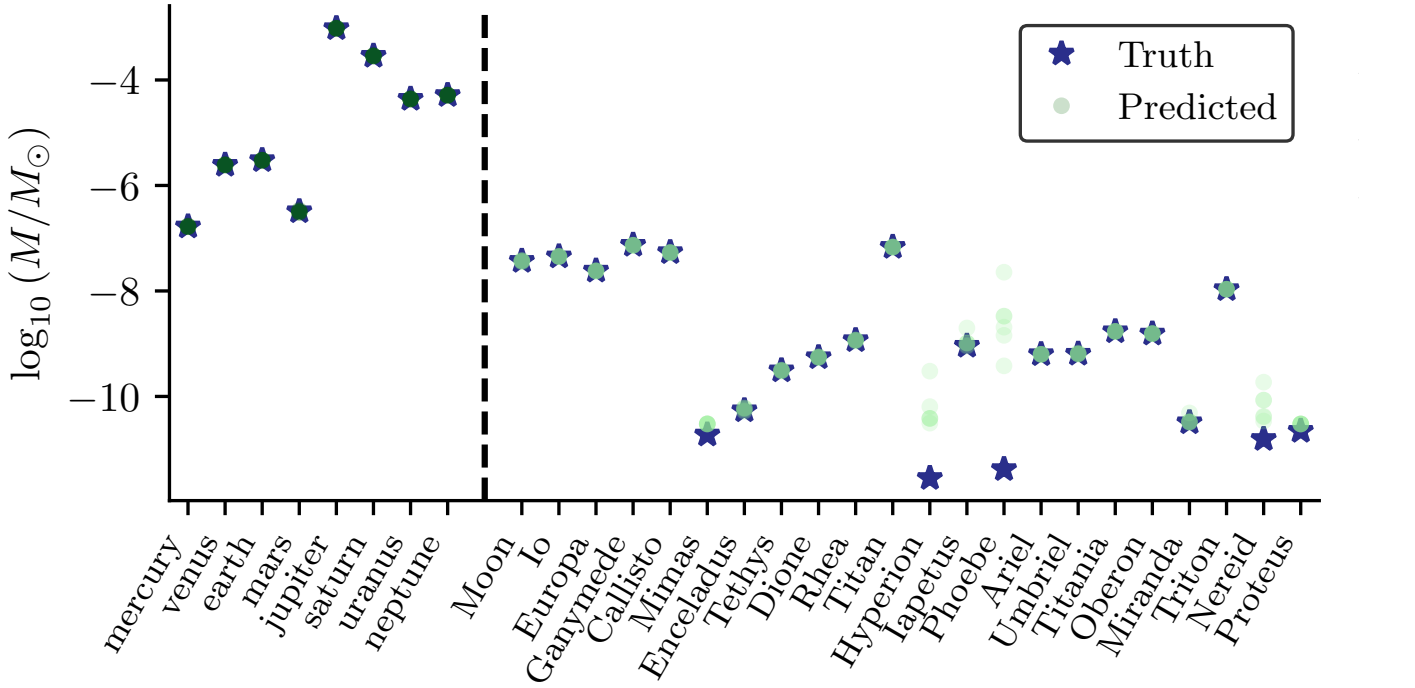


Fig. 4A: Graph network + symbolic regression

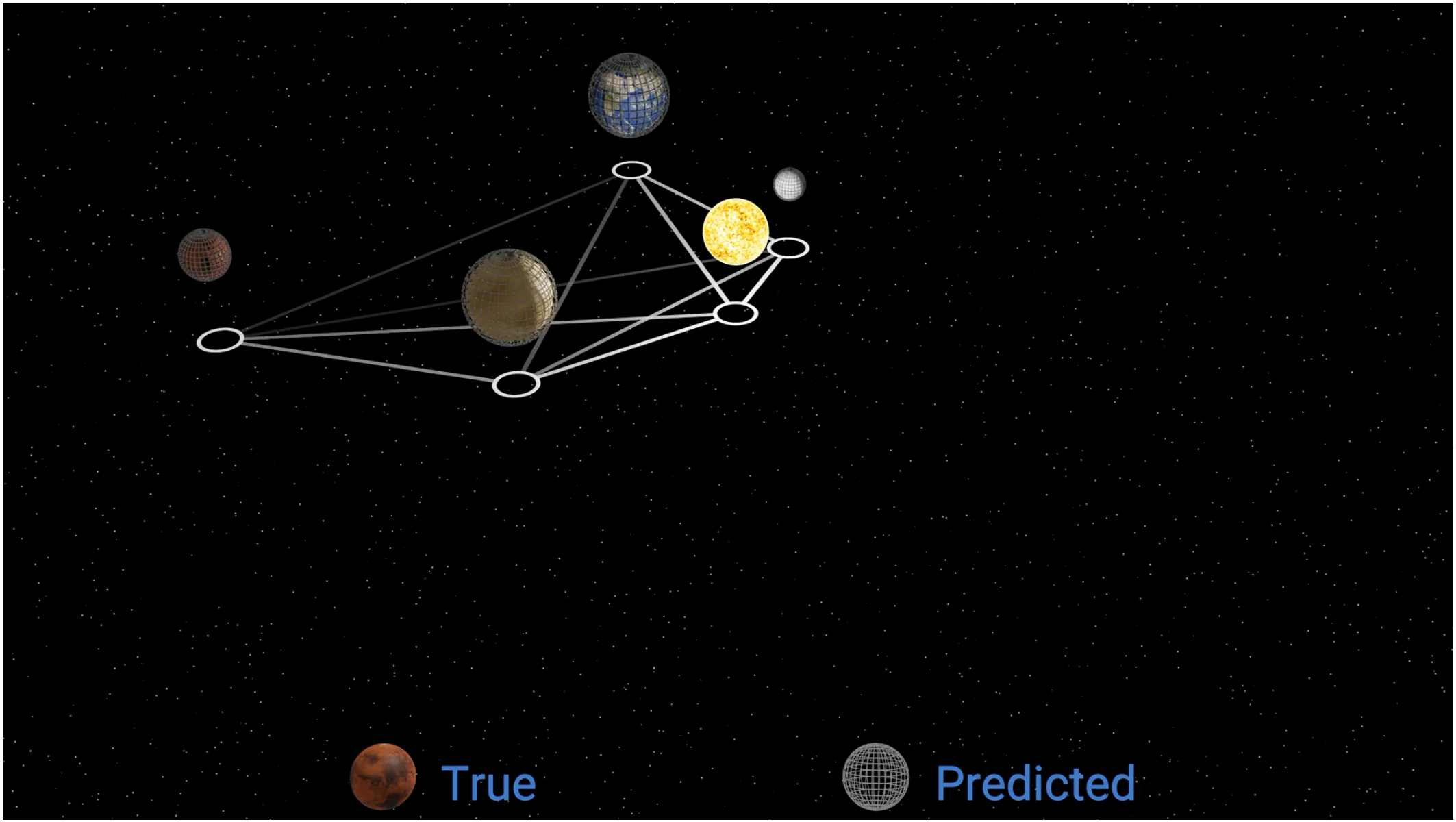


Graph network + symbolic regression + relearned masses



UNIVERSITY OF SUSSEX

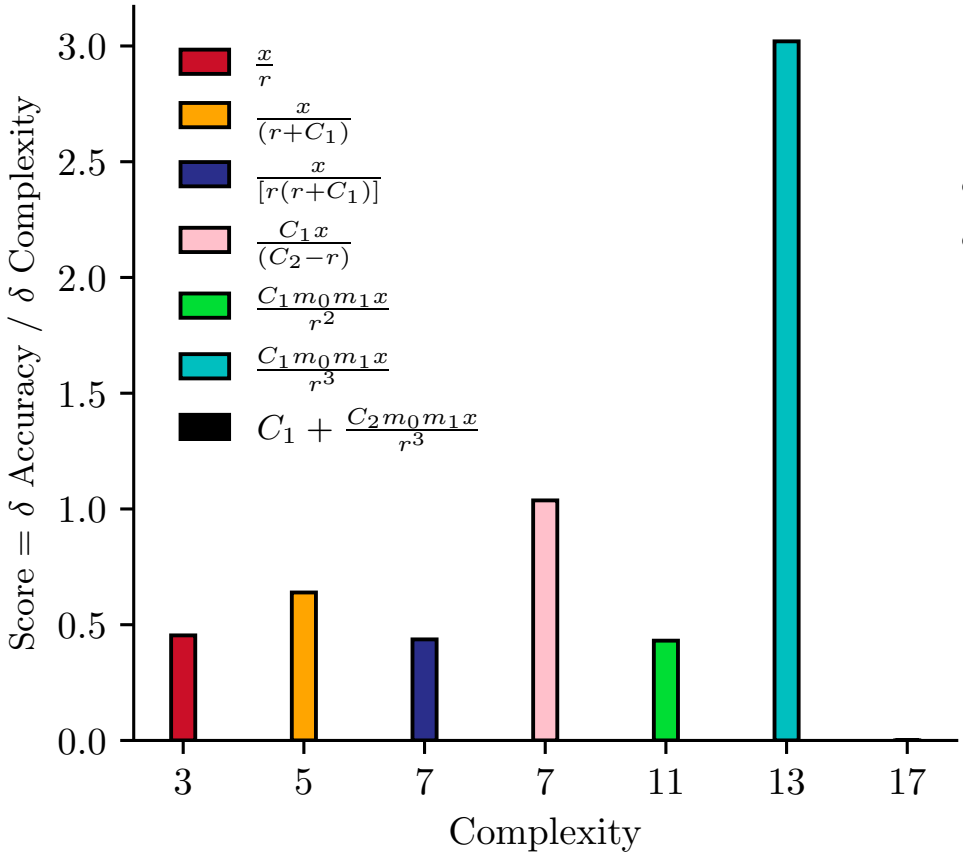
UCL



UNIVERSITY OF SUSSEX

UCL

Predicted equations



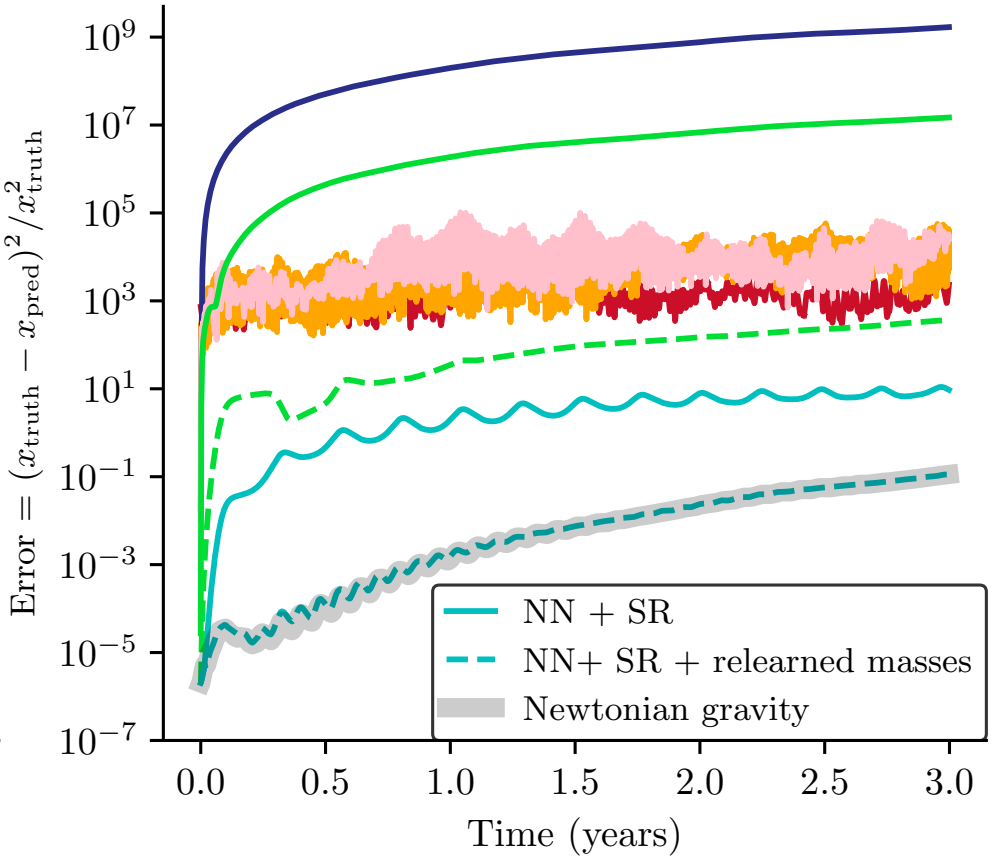


Fig. 2A: Graph Network

Fig. 2B: Rollout

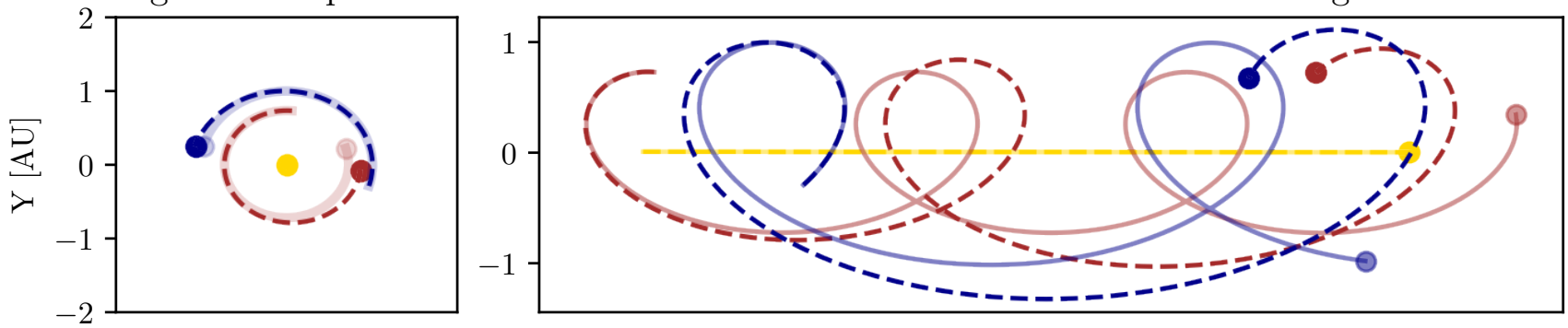


Fig. 2C: Graph Network + Symbolic Regression

Fig. 2D: Rollout

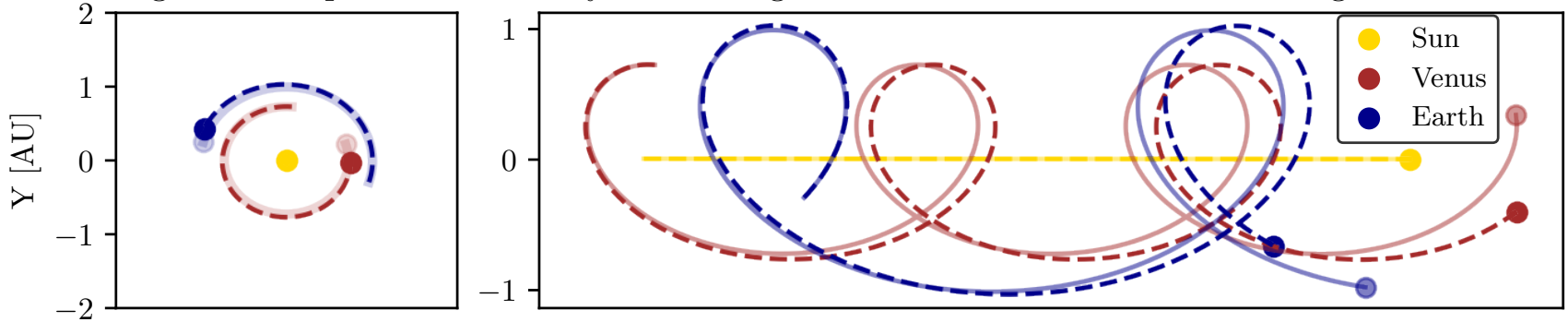
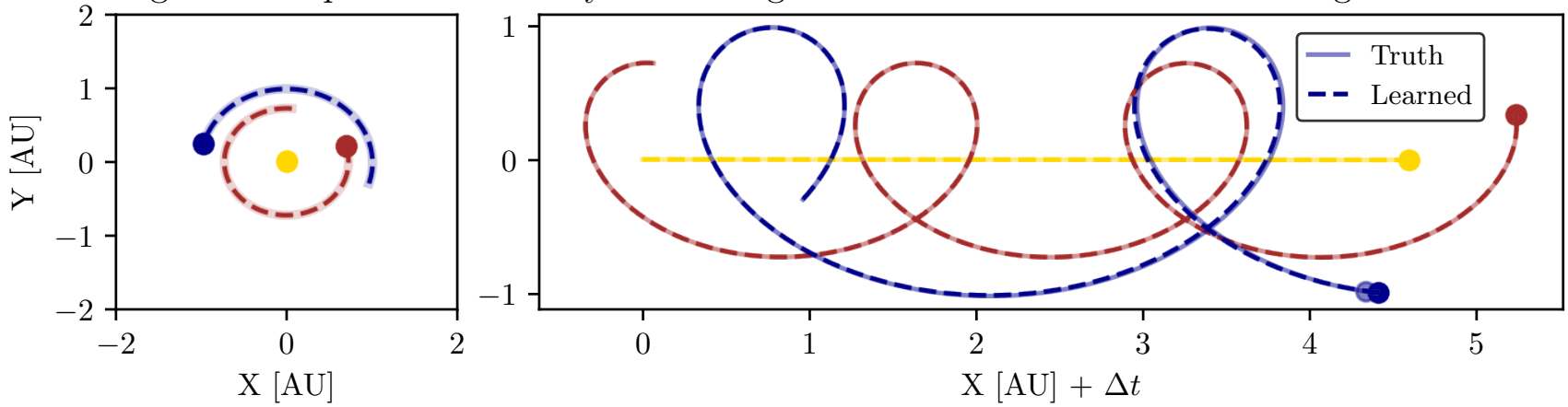
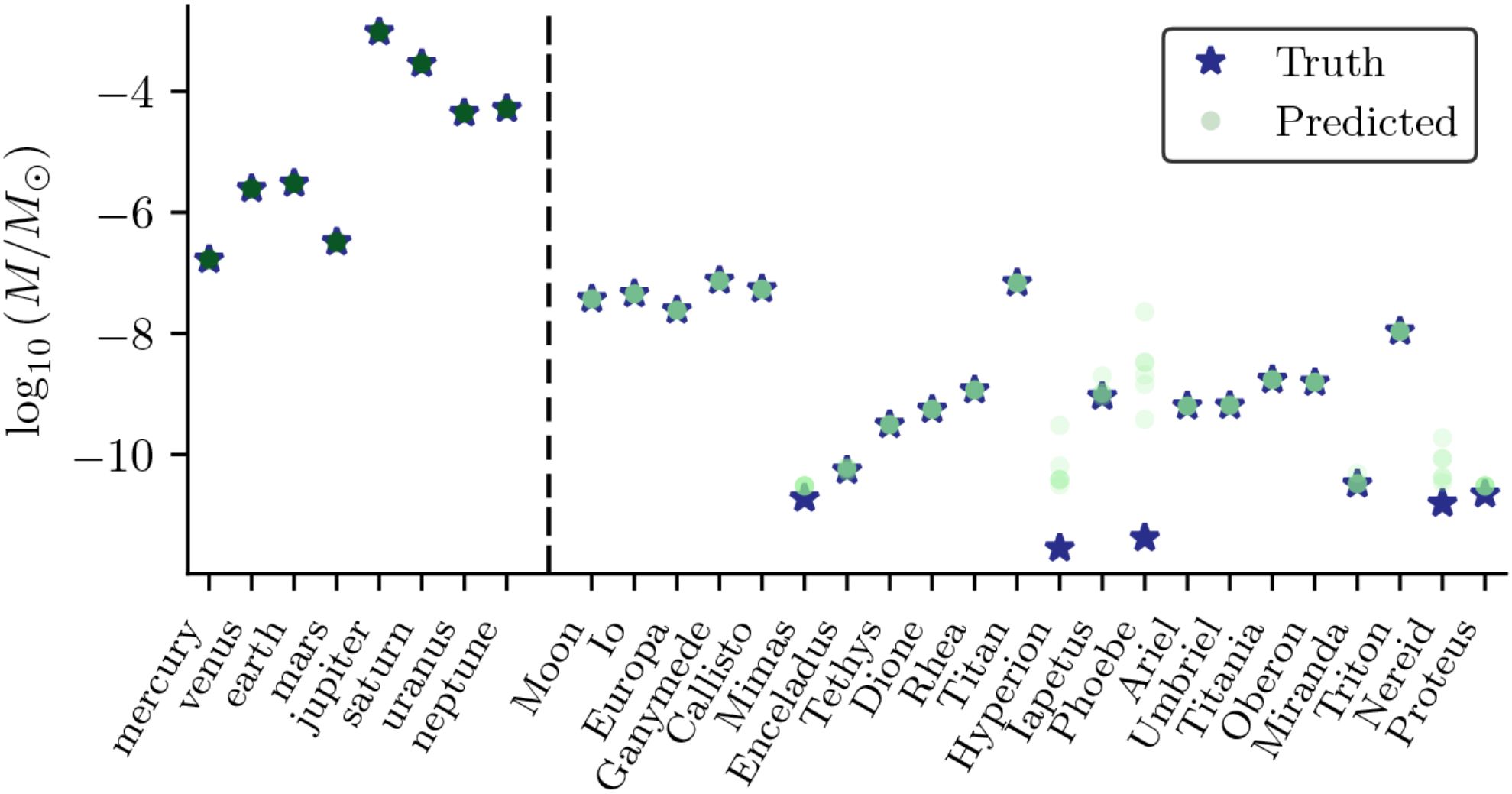


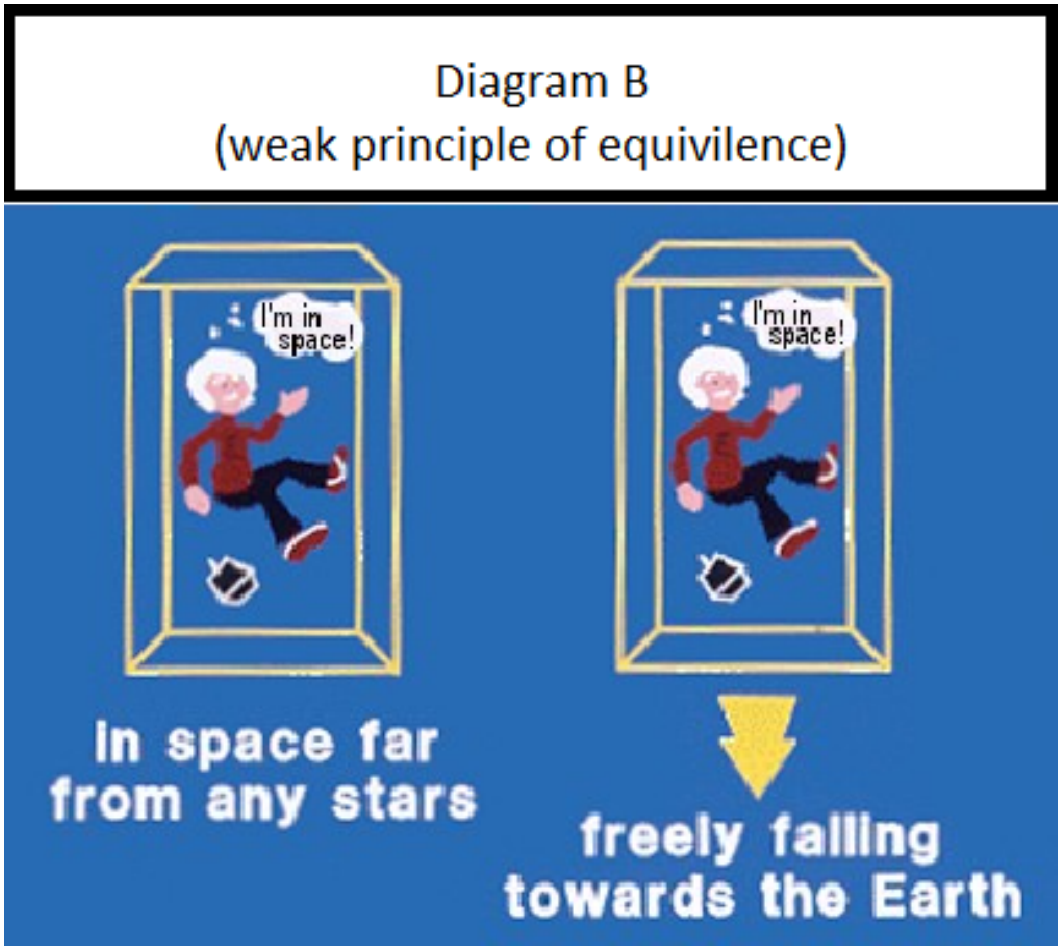
Fig. 2E: Graph Network + Symbolic Regression + relearned masses

Fig. 2F: Rollout

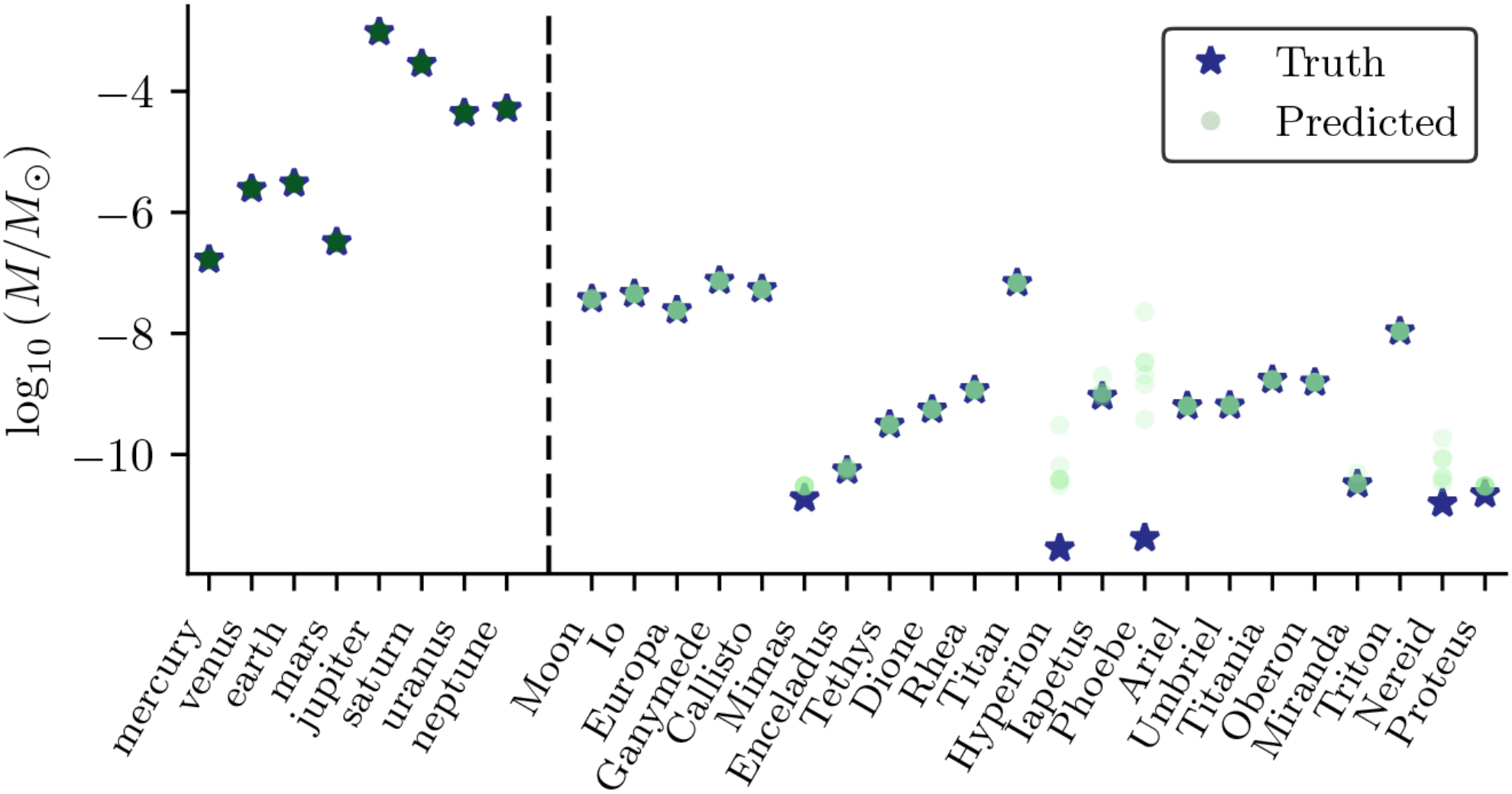


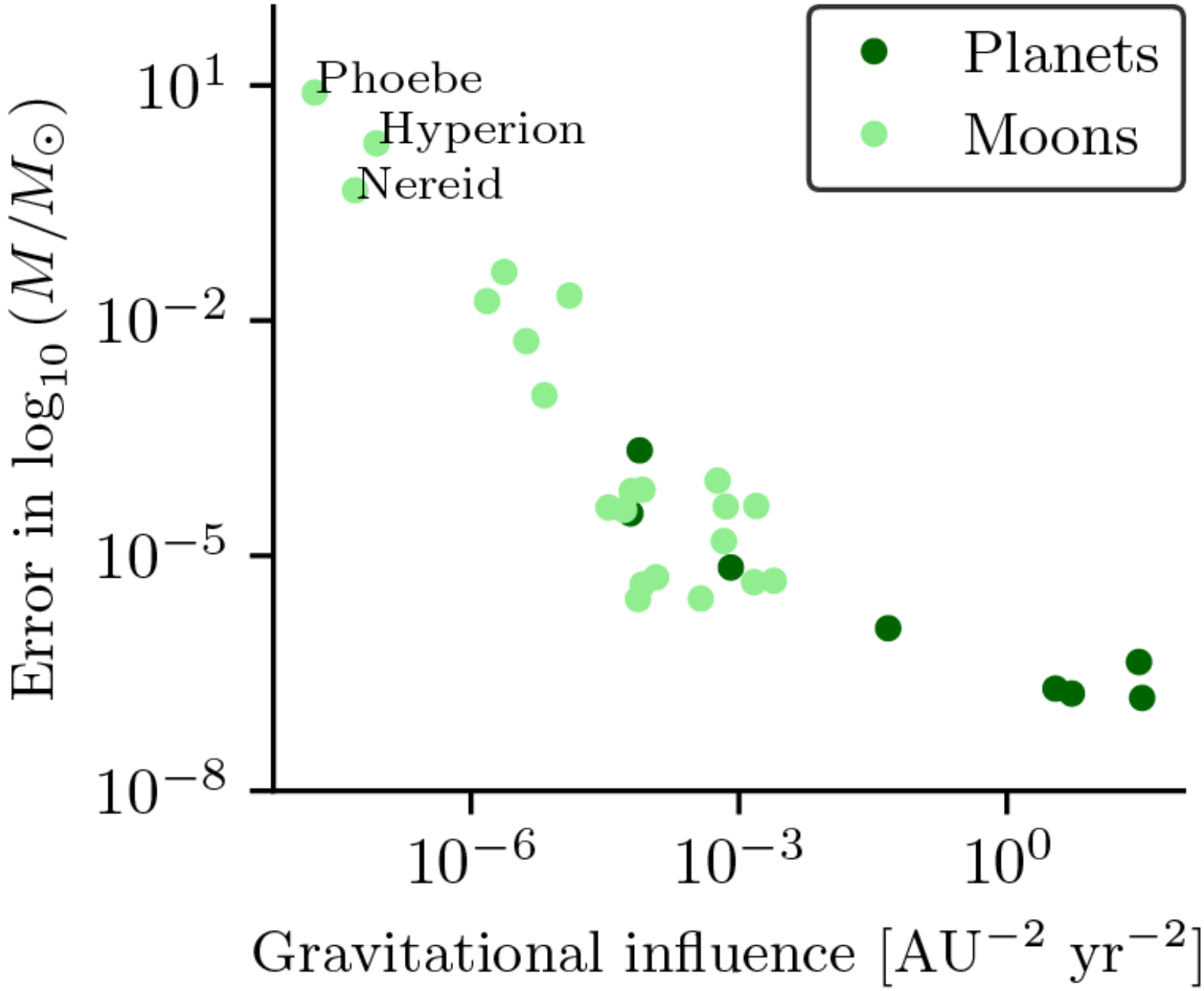

Truth
Predicted
$\log_{10}(M/M_\odot)$
−4
−6
−8
−10
mercury
venus
earth
mars
jupiter
saturn
uranus
neptune
Moon
Io
Europa
Ganymede
Callisto
Mimas
Enceladus
Tethys
Dione
Rhea
Titan
Hyperion
Iapetus
Phoebe
Ariel
Umbriel
Titania
Oberon
Miranda
Triton
Nereid
Proteus

Equivalence principle: Your movement does not depend on your own mass



= Masses can only be measured from their influence on other bodies


Truth
Predicted
$\log_{10}(M/M_\odot)$
−4
−6
−8
−10
mercury
venus
earth
mars
jupiter
saturn
uranus
neptune
Moon
Io
Europa
Ganymede
Callisto
Mimas
Enceladus
Tethys
Dione
Rhea
Titan
Hyperion
Iapetus
Phoebe
Ariel
Umbriel
Titania
Oberon
Miranda
Triton
Nereid
Proteus

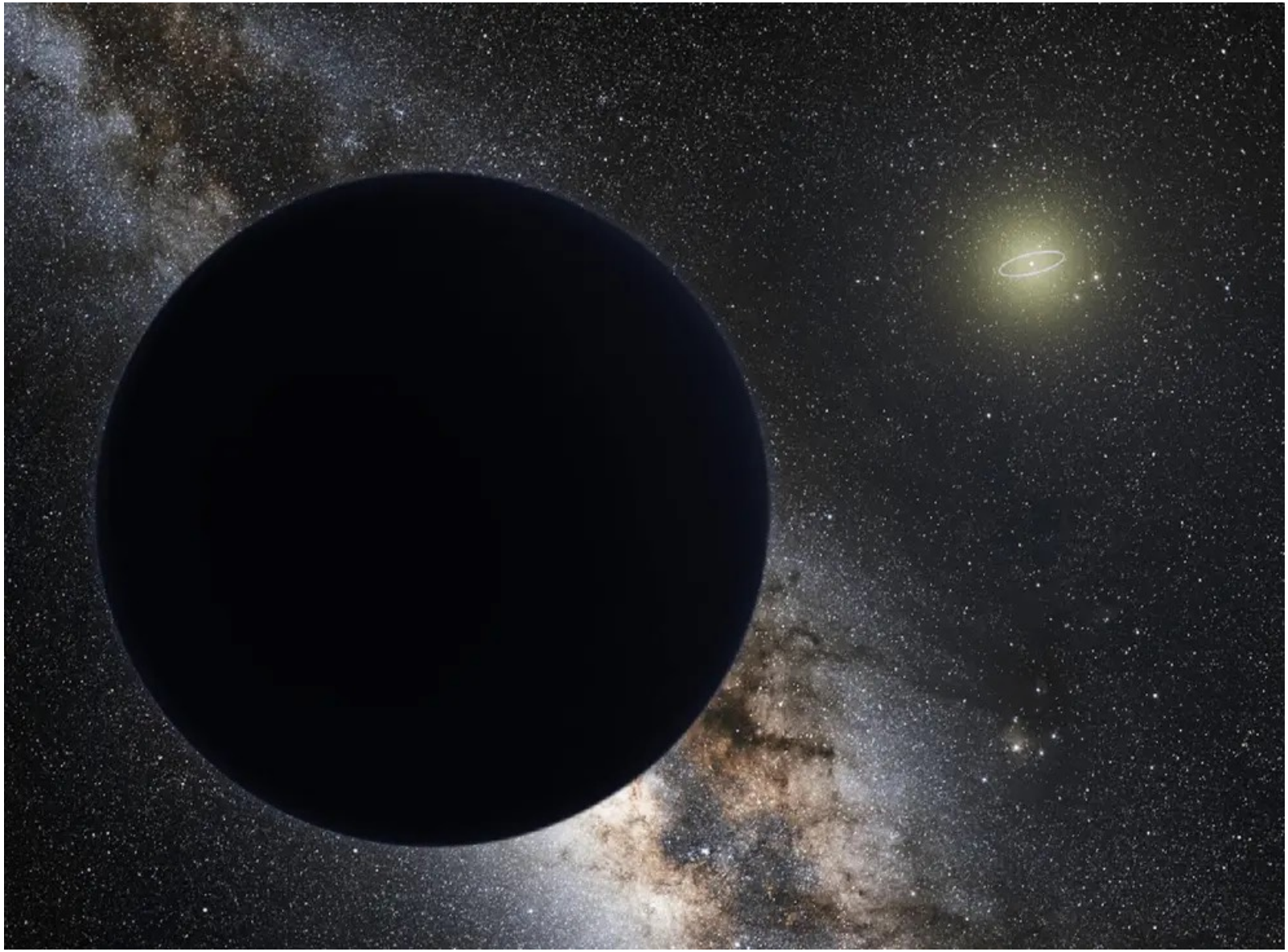

Planets
Moons
Phoebe
Hyperion
Nereid
10^{1}
10^{-2}
10^{-5}
10^{-8}
Error in $\log_{10}(M/M_\odot)$
10^{-6}
10^{-3}
10^{0}
Gravitational influence [AU^{-2} yr^{-2}]

Future work

UNIVERSITY OF SUSSEX

UCL

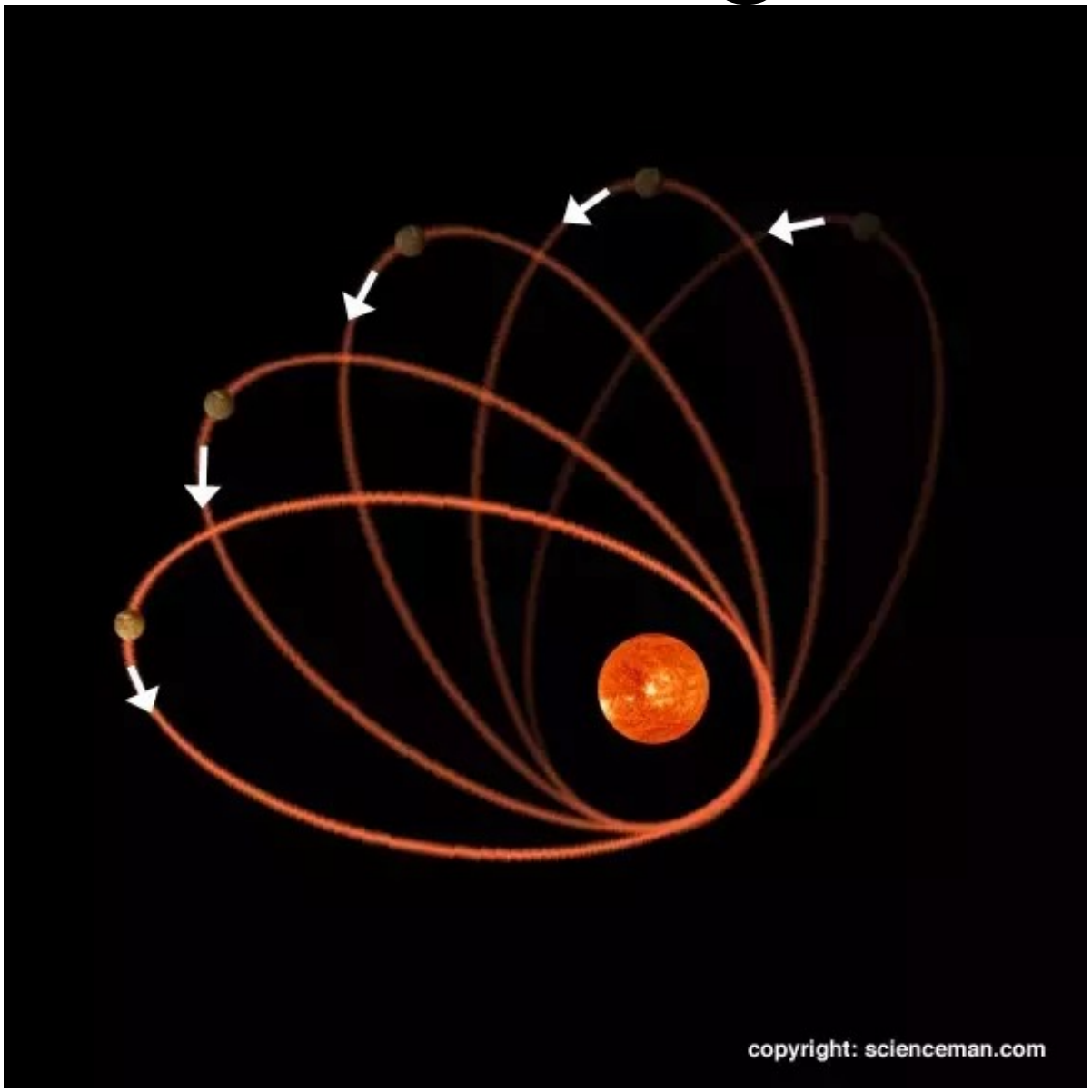
Find missing/hidden bodies



UNIVERSITY OF SUSSEX

UCL

Recovering GR

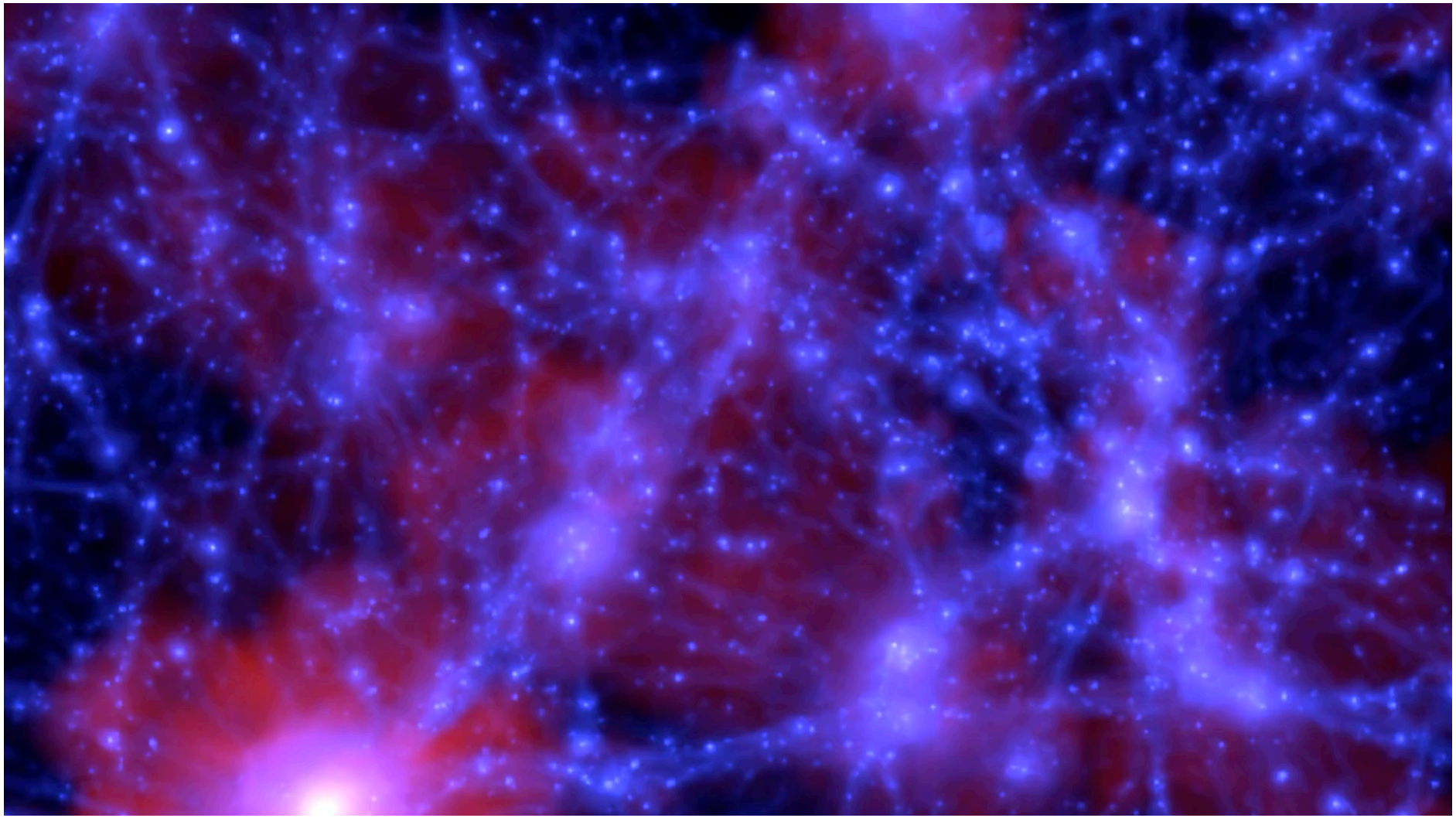


SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

Learning the Universe



SIMONS FOUNDATION

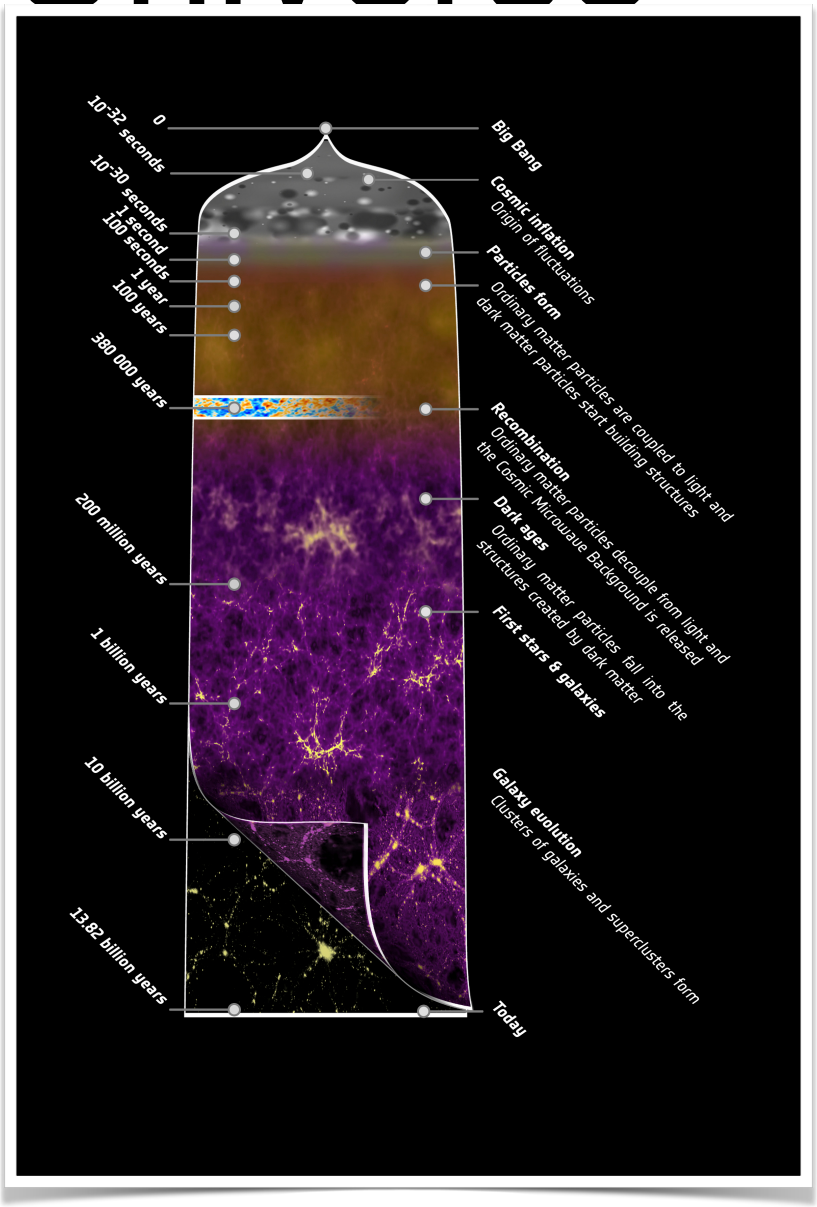
UNIVERSITY OF SUSSEX

UCL

Learning the Universe

The entire history of the Universe, from Inflation until now, is fully described by 6 parameters:

- Baryon Density: Ω_b
- Matter Density: Ω_m
- Hubble Constant: H_0
- Two inflation Parameters: n_s, A_s
- Optical Depth: τ

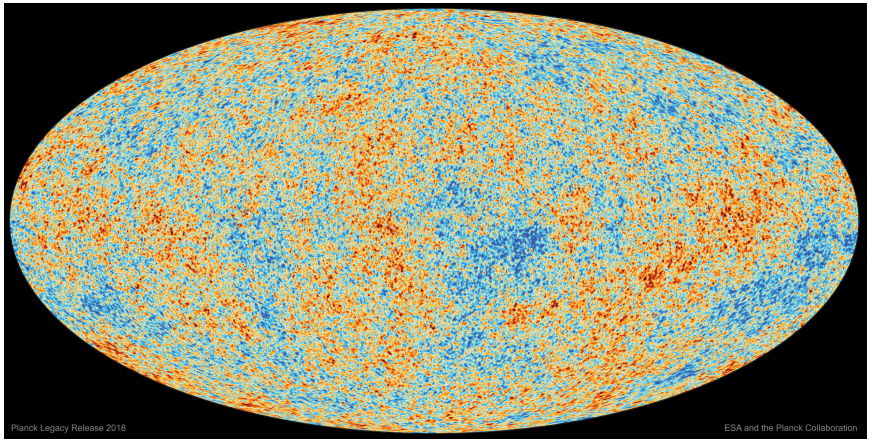


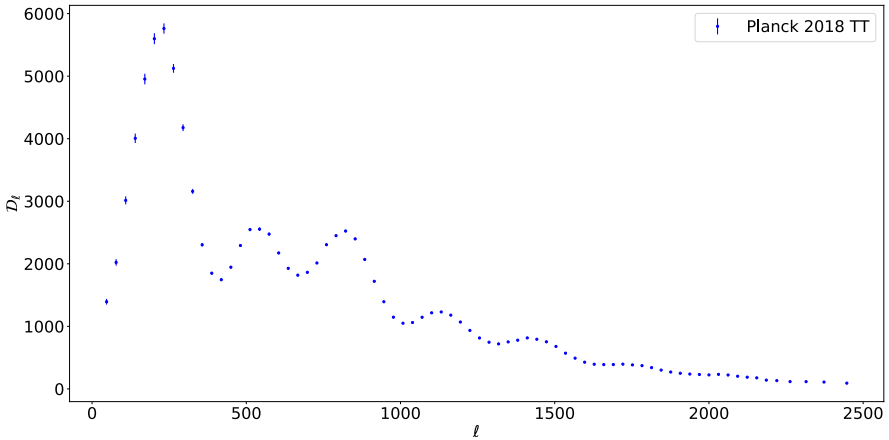
SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

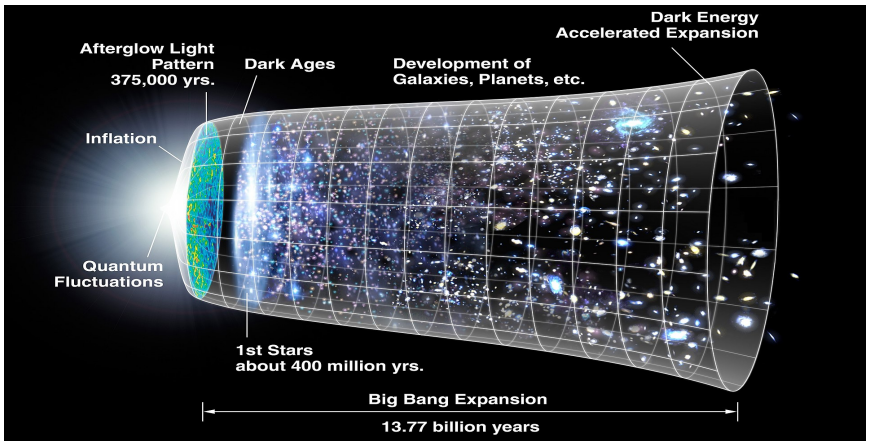
UCL

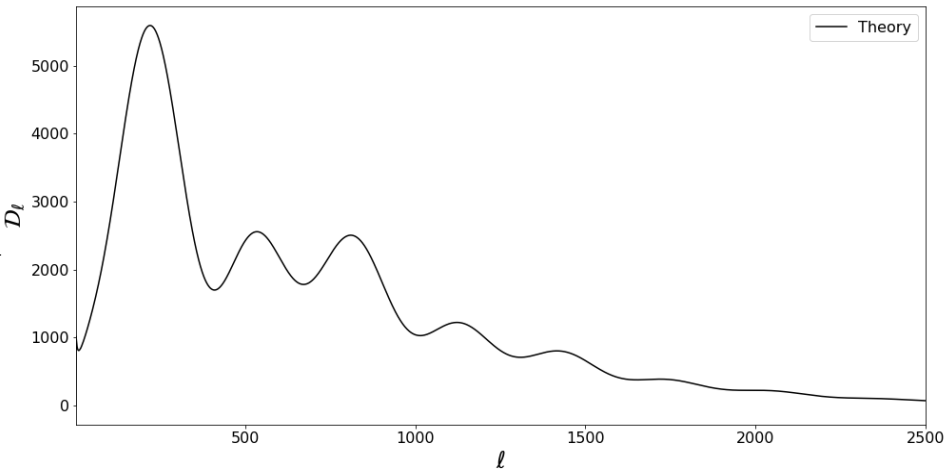
Learning the Universe





$$\mathscr{L} \equiv P(D \,|\, \theta, M)$$





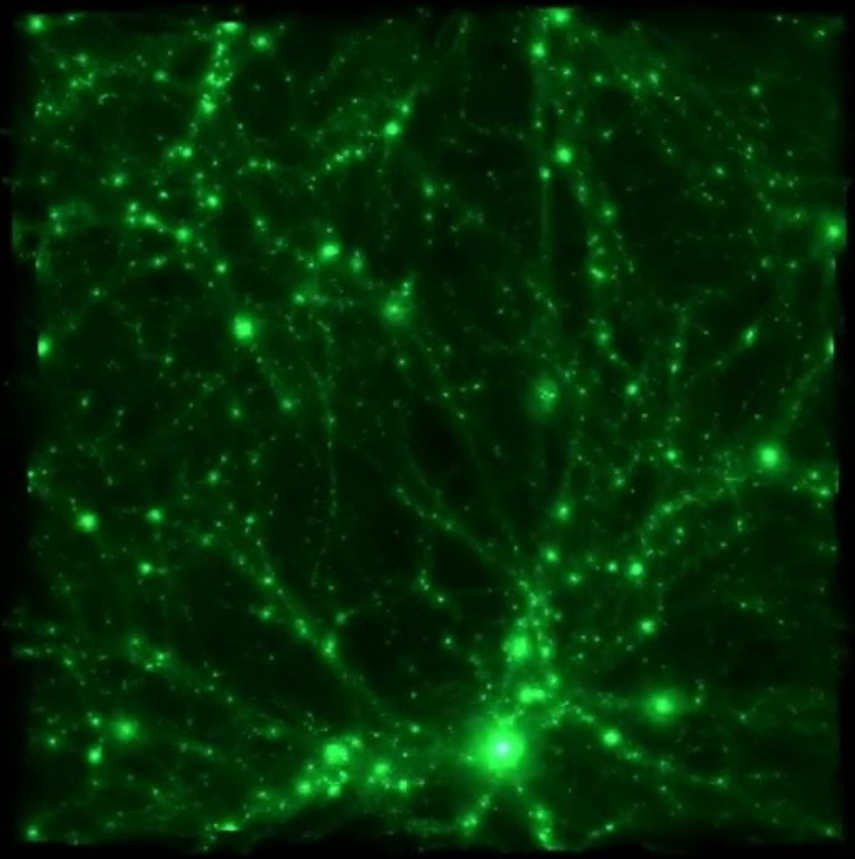
SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

Learning the Universe

Ω_m	0.3
σ_8	0.8
A_{SN1}	4.00
A_{AGN1}	1.0
A_{SN2}	1.0
A_{AGN2}	1.0



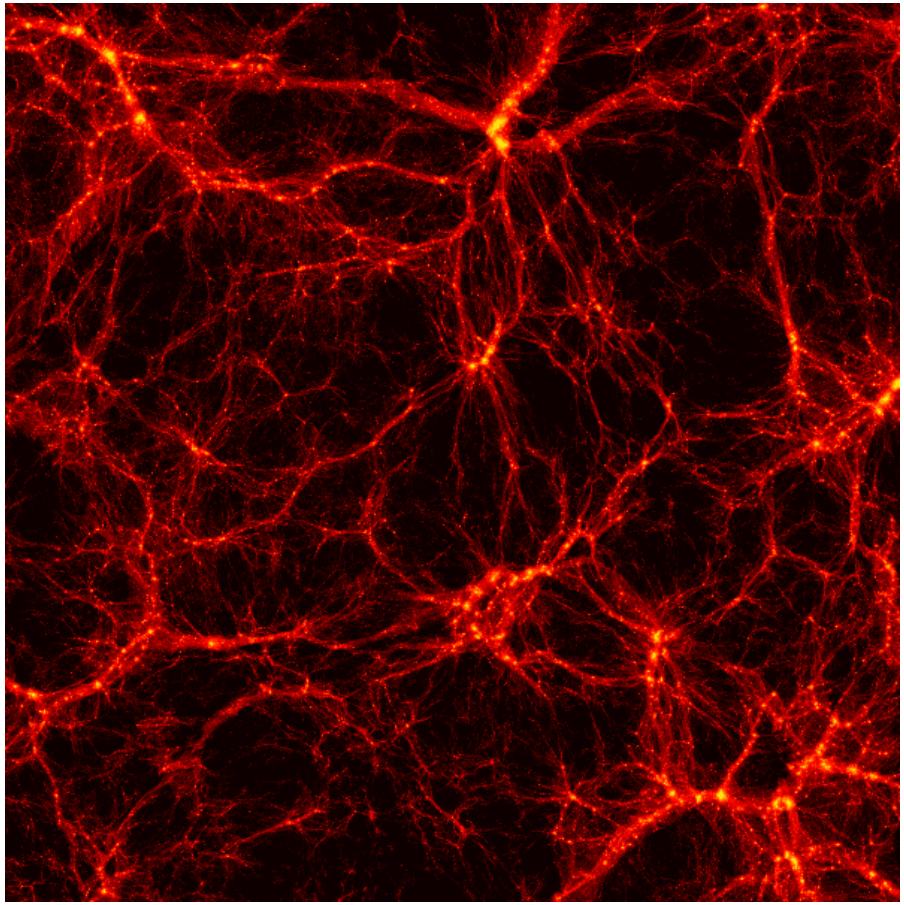
Villaescusa-Navarro et al.
CAMELS simulations

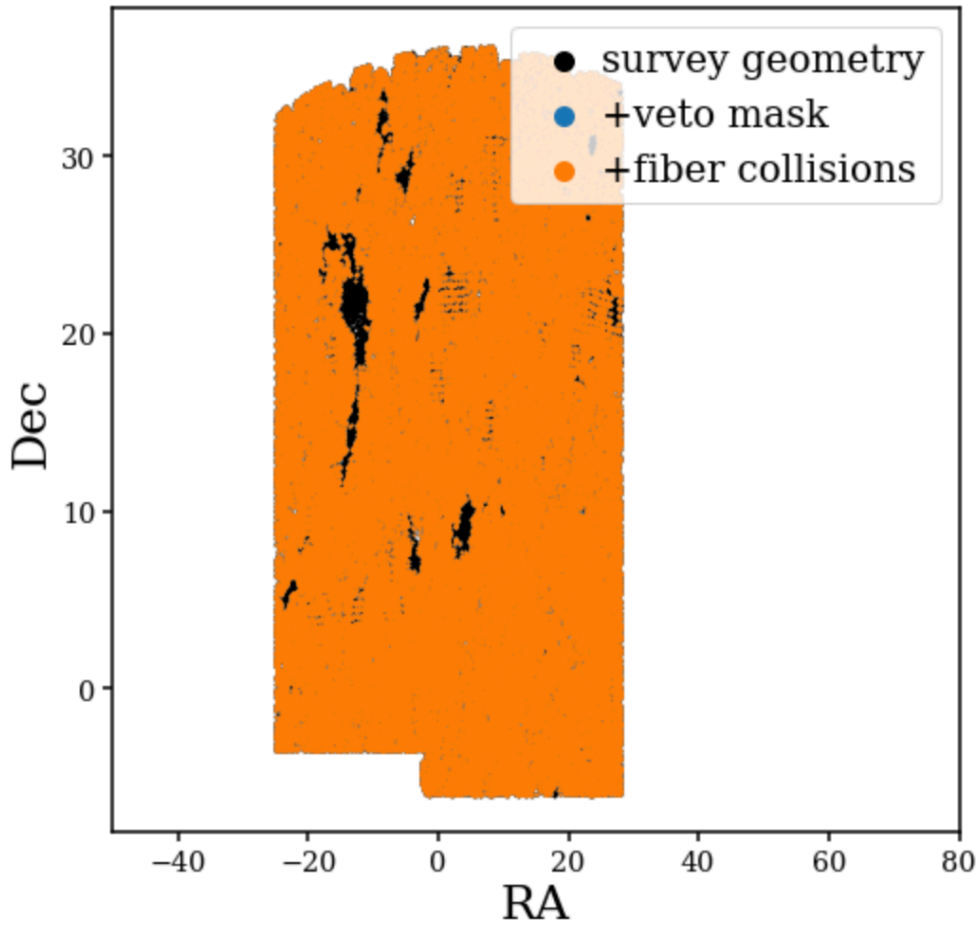
SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

SIMulation Based Inference of Galaxies (SIMBIG)



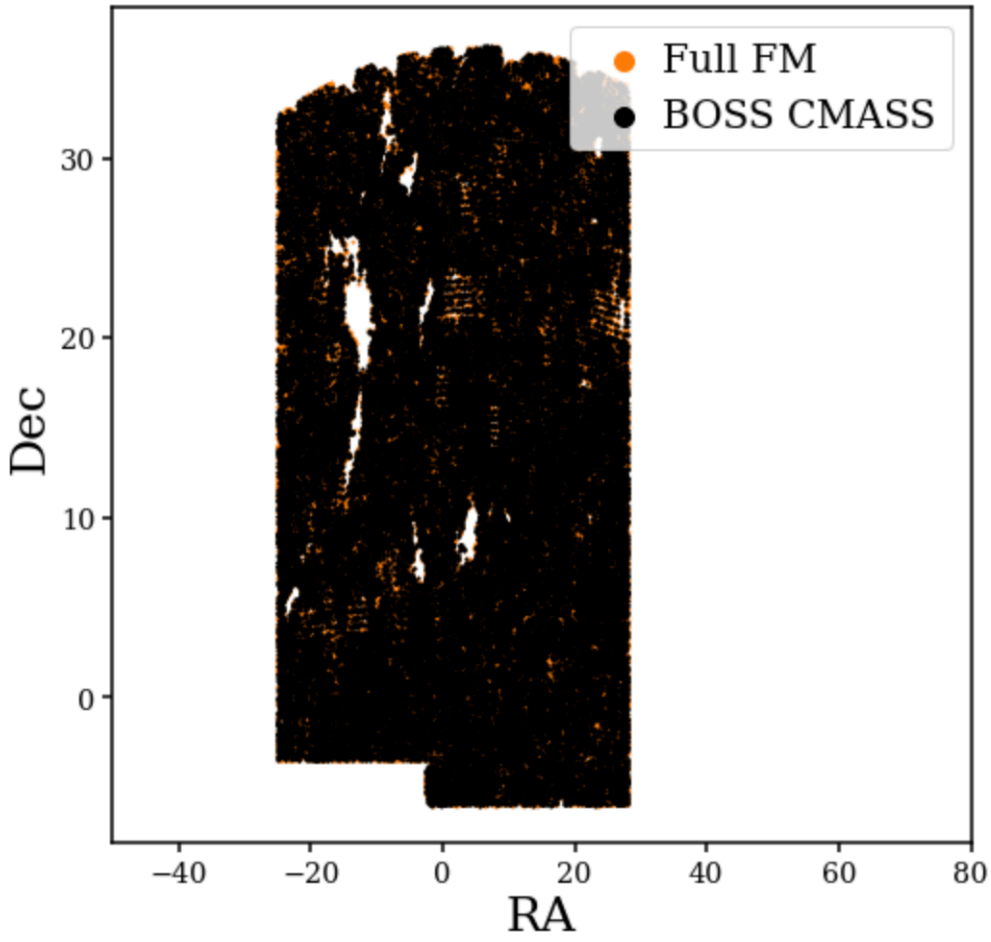


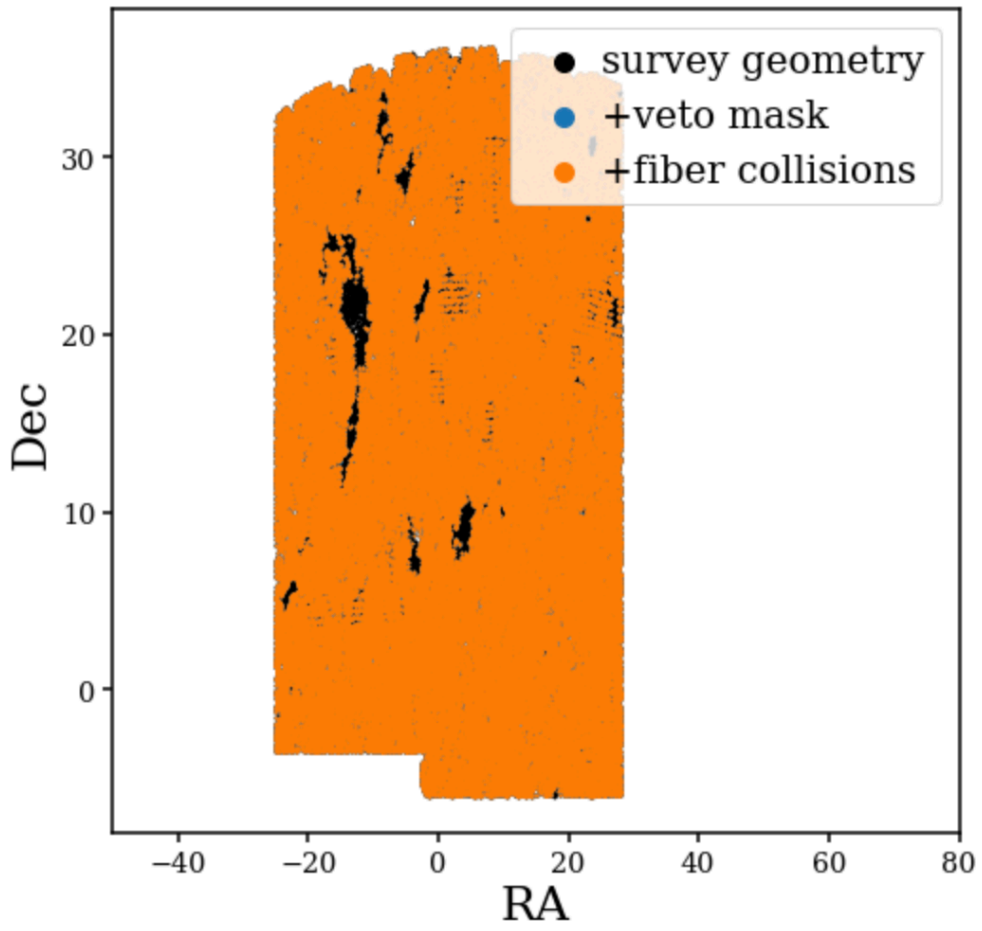
SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

SIMulation Based Inference of Galaxies (SIMBIG)



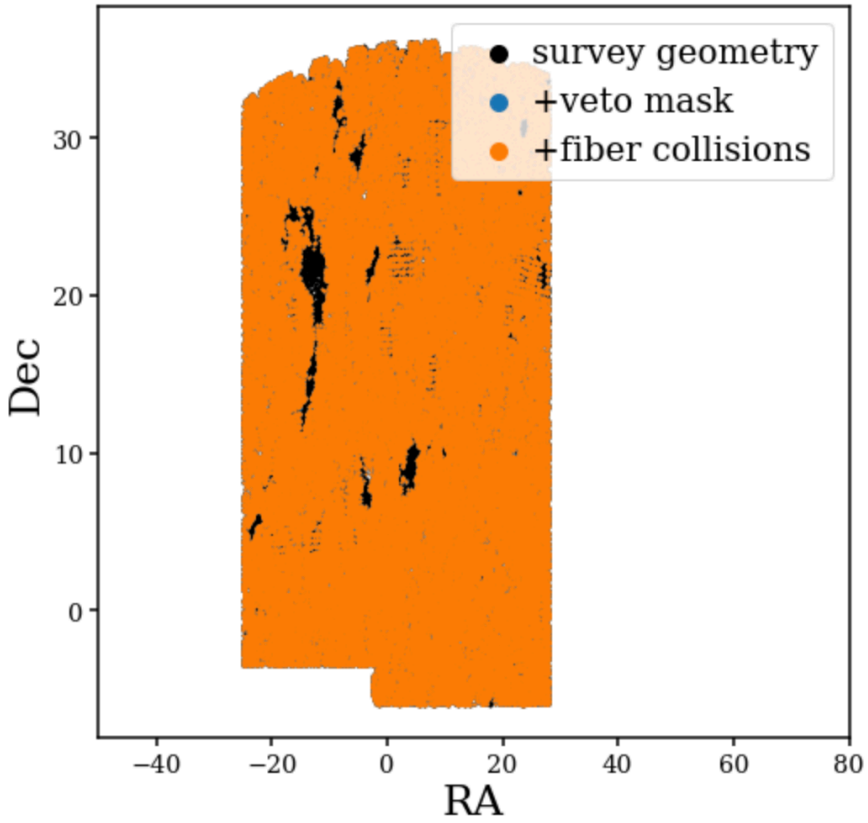


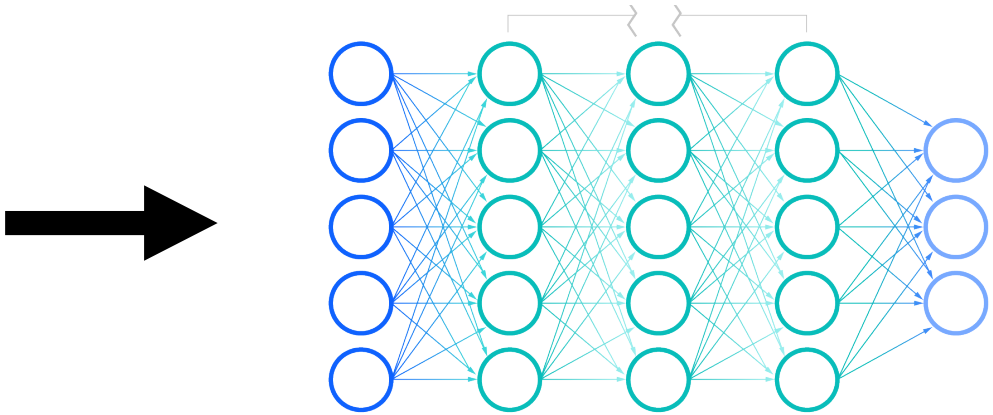
SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

Learning the Universe





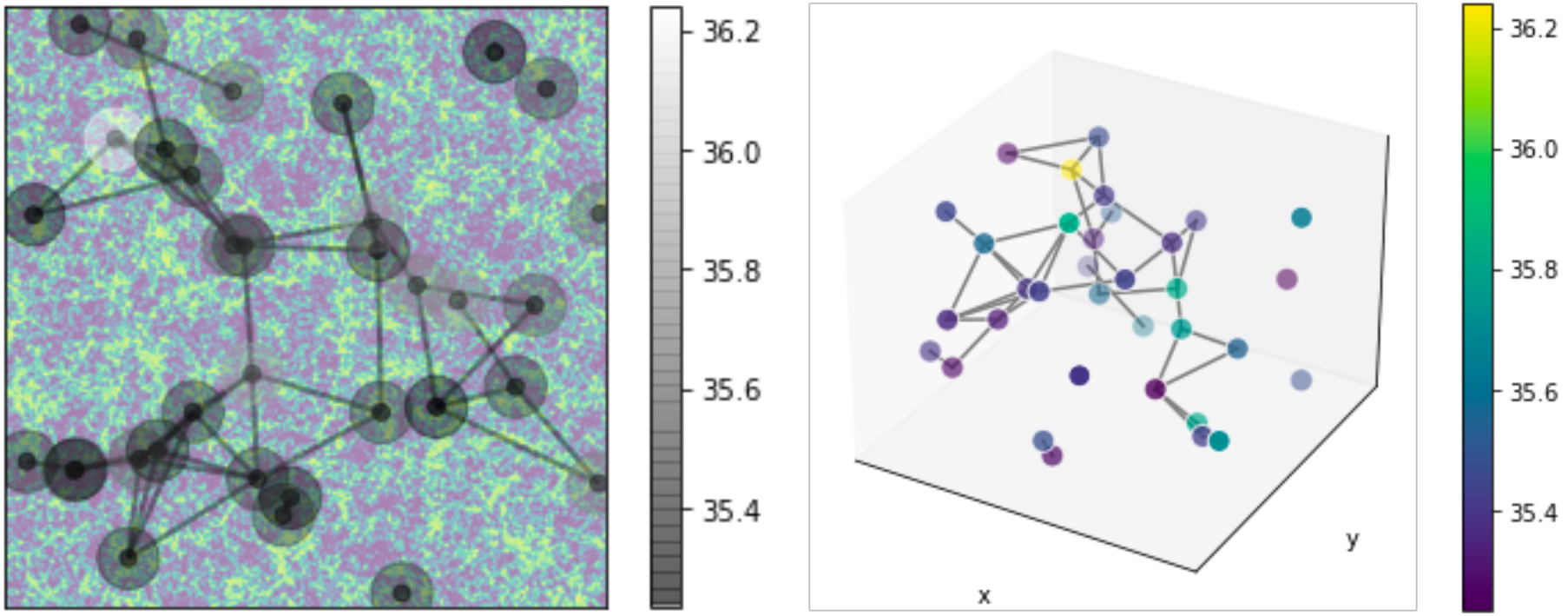
$$\theta = \{\Omega_m, \Omega_b, H_0, A_s, n_s, \tau\}$$

SIMONS FOUNDATION

UNIVERSITY OF SUSSEX

UCL

Learning the Universe

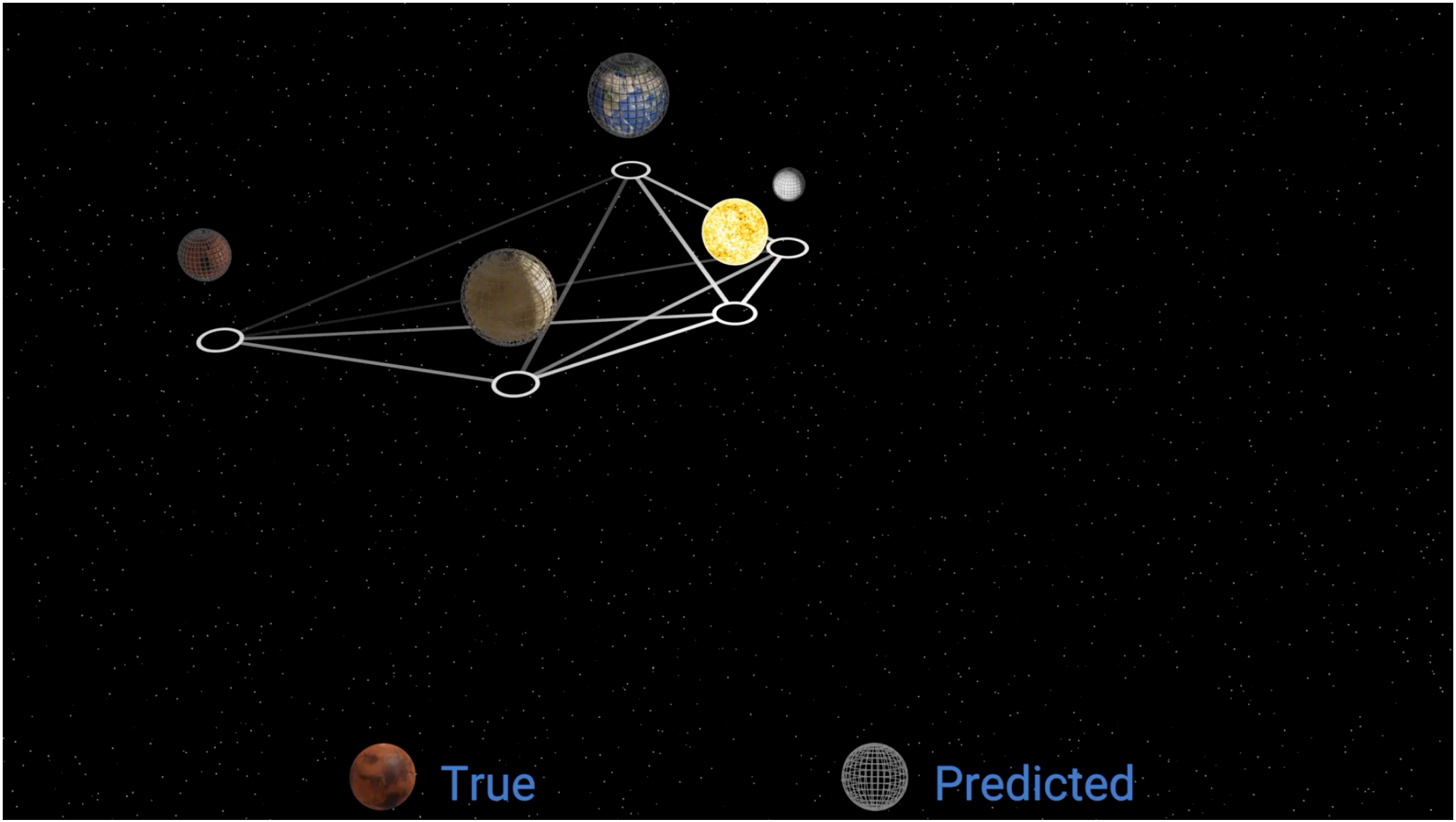


Work in progress, led by **Lucas Makinen** (Imperial) in collaboration with **PL, Alan Heavens** (Imperial), **Natalia Porqueres** (Imperial) & **Benjamin Wandelt** (IAP, Paris).

Summary

- We use graph neural networks and symbolic regression on 30 years of Solar System data.
- Our algorithm learns the correct equation for Newtonian gravity, and the masses of the planets and moons, directly from the data.
- This is only possible thanks to using inductive biases, and other prior knowledge.
- This shows that AI can be used to automate scientific discovery.

DeepMind graph nets library


True
Predicted