

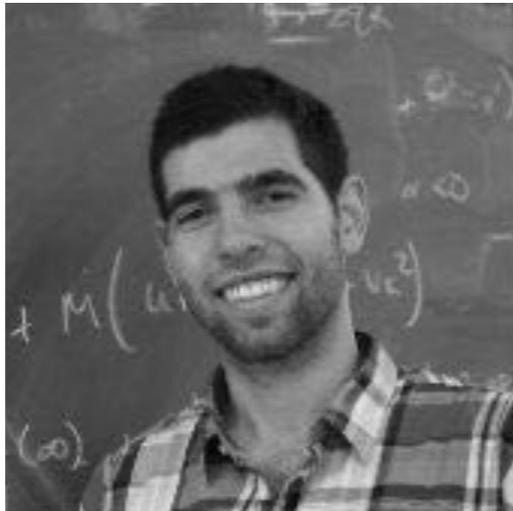


CENTRE FOR PLANETARY SCIENCES
UNIVERSITY OF TORONTO

Formation and stability of planetary systems

Hanno Rein

Collaborators



Ari Silburt

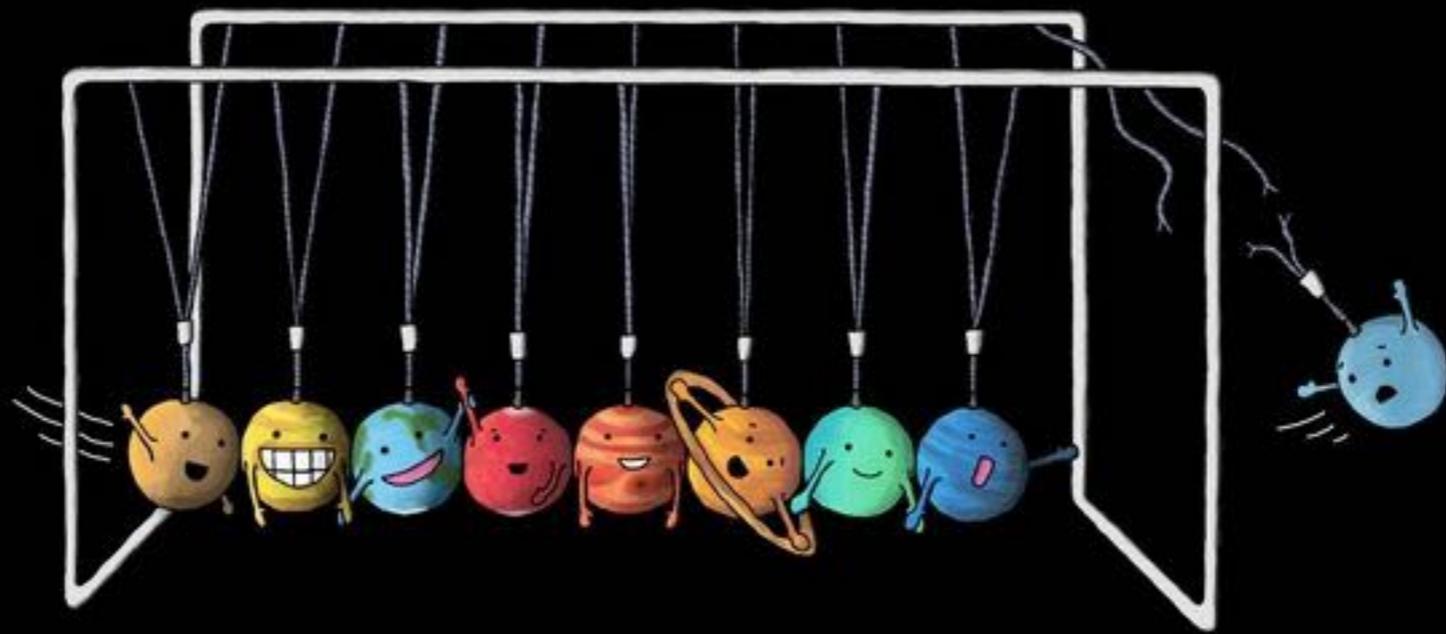
My first PhD student!
Now a postdoc at Penn State



Dan Tamayo

Postdoc
Centre for Planetary Sciences
University of Toronto

1. Solar System
2. WHFast
3. Reproducibility
4. JANUS
5. Machine Learning



1. The Solar System

Gravity

$$\ddot{\mathbf{r}}_i = \sum_{\substack{j=1 \\ j \neq i}}^N m_j \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^3}$$

History of secular perturbation theory

k	Lagrange (1774)	Laskar <i>et al.</i> , 2004
s_1	5.980	5.59
s_2	6.311	7.05
s_3	19.798	18.850
s_4	18.308	17.755
s_5	0	0
s_6	25.337	26.347

Previous direct numerical simulations

LONGSTOP (1982)

- Outer planets only
- No instability

Digital Orrery (1988)

- Outer planets only, 800 Myr
- Pluto is chaotic

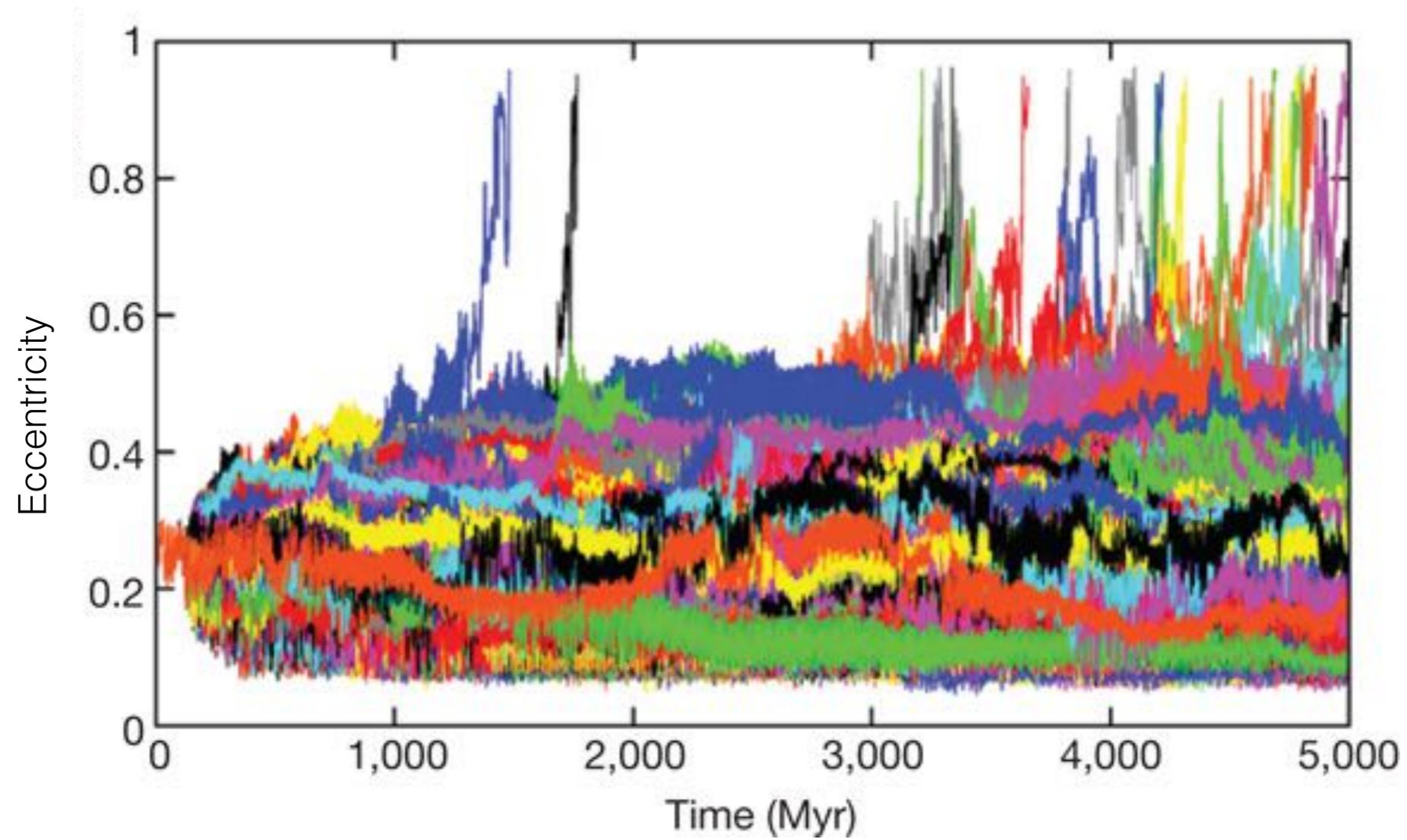
Laskar (1989)

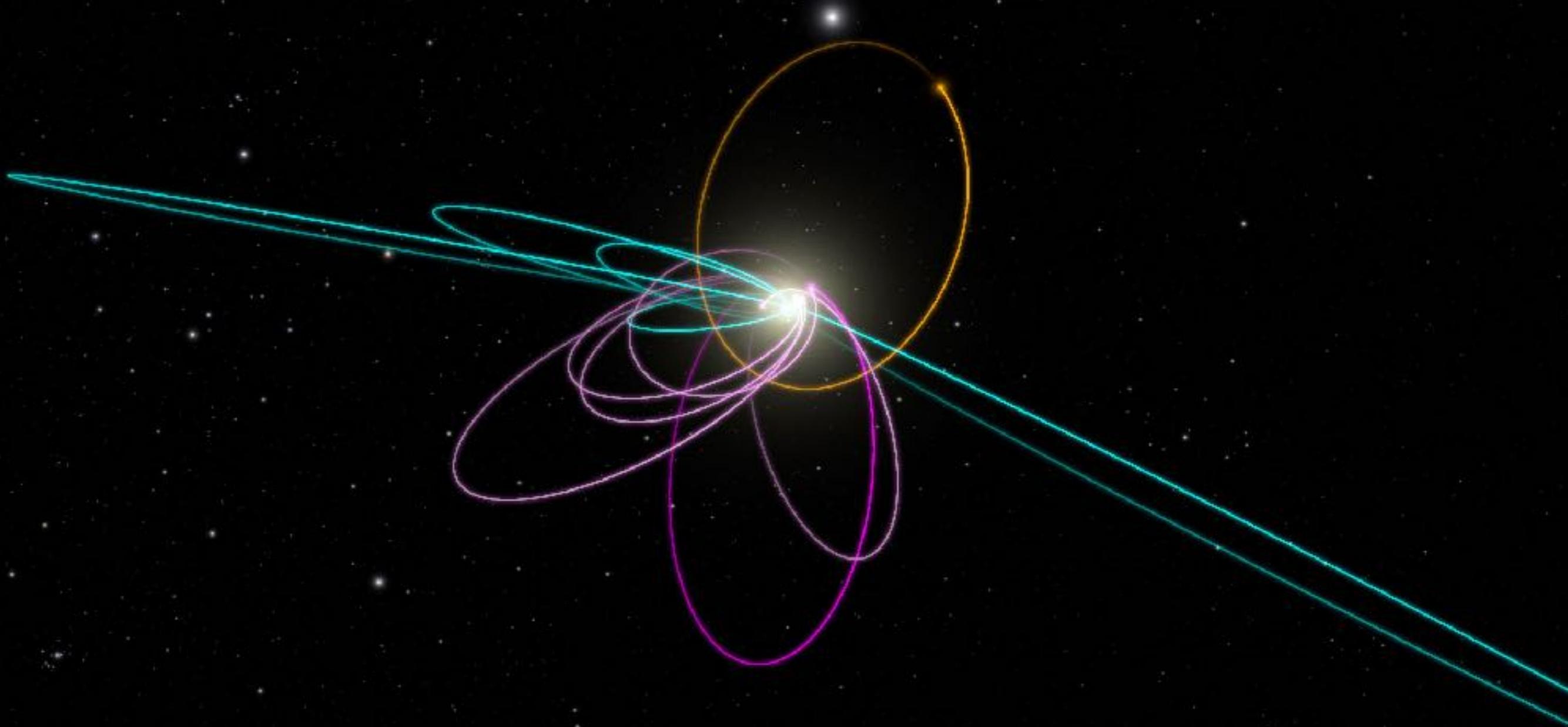
- All planets, averaged equations
- Earth is chaotic on a 100 Myr timescale

Laskar (2009)

- All planets, full equations
- Collisions between terrestrial planets possible

Stability of the Solar System





2. WHFast

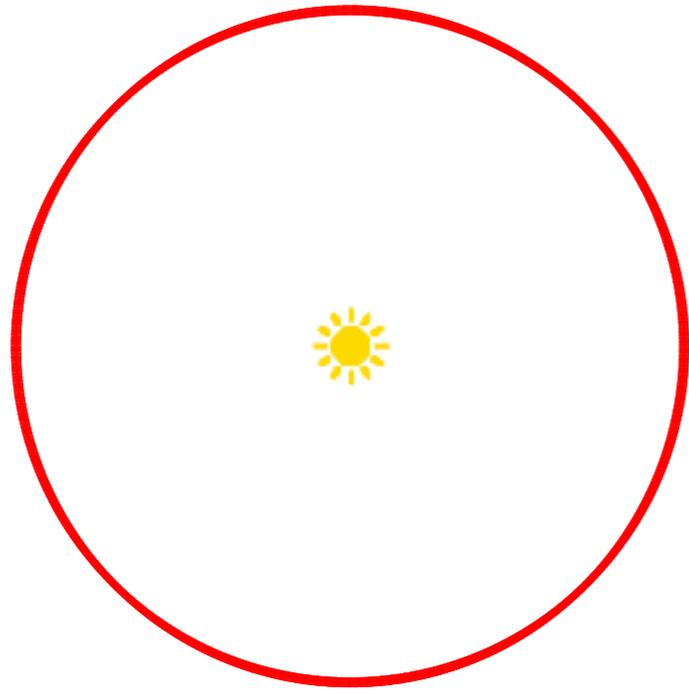
Mixed Variable Symplectic Integrator

$$H = \underbrace{\frac{1}{2}p^2 + \Phi_{\text{Sun}}(q)}_{\text{Drift}} + \underbrace{\Phi_{\text{Other}}(q)}_{\text{Kick}}$$

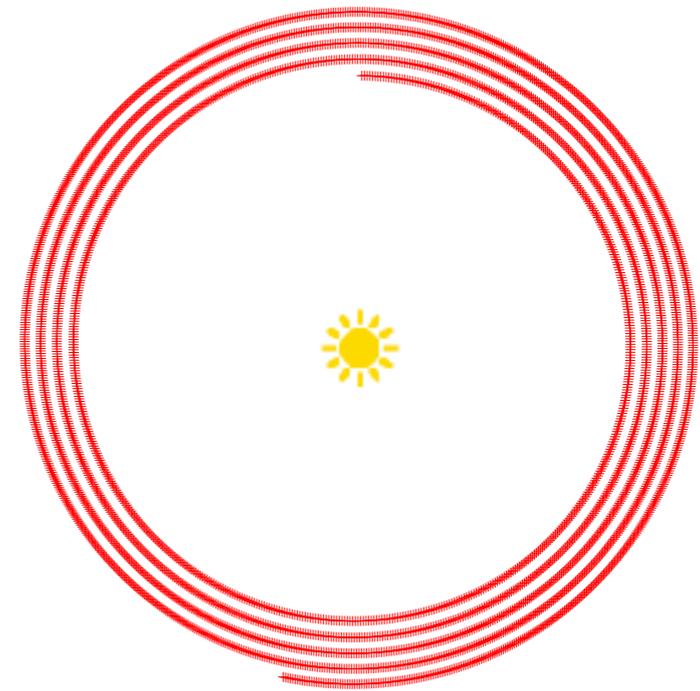
Particularly good if

$$\frac{1}{2}p^2 + \Phi_{\text{Sun}}(q) \gg \Phi_{\text{Other}}(q)$$

Symplectic integrators



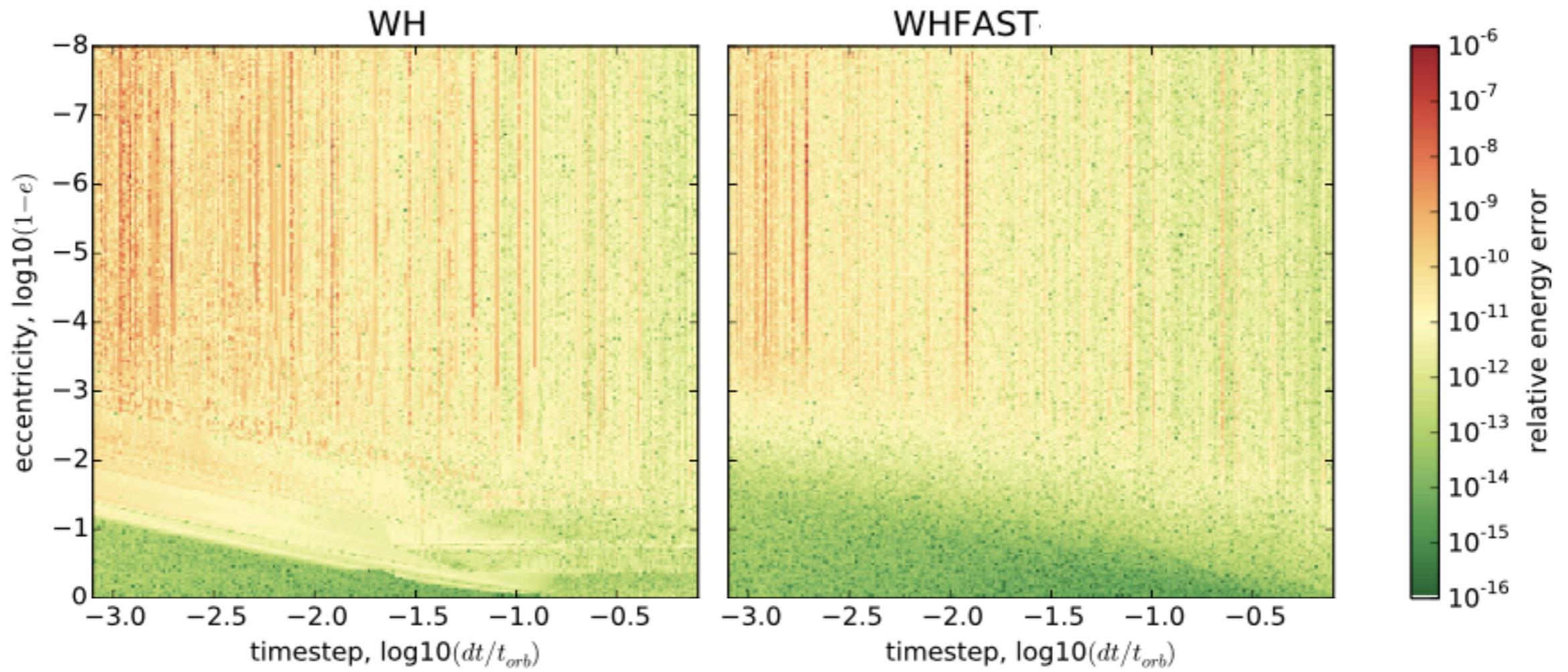
Symplectic integrator



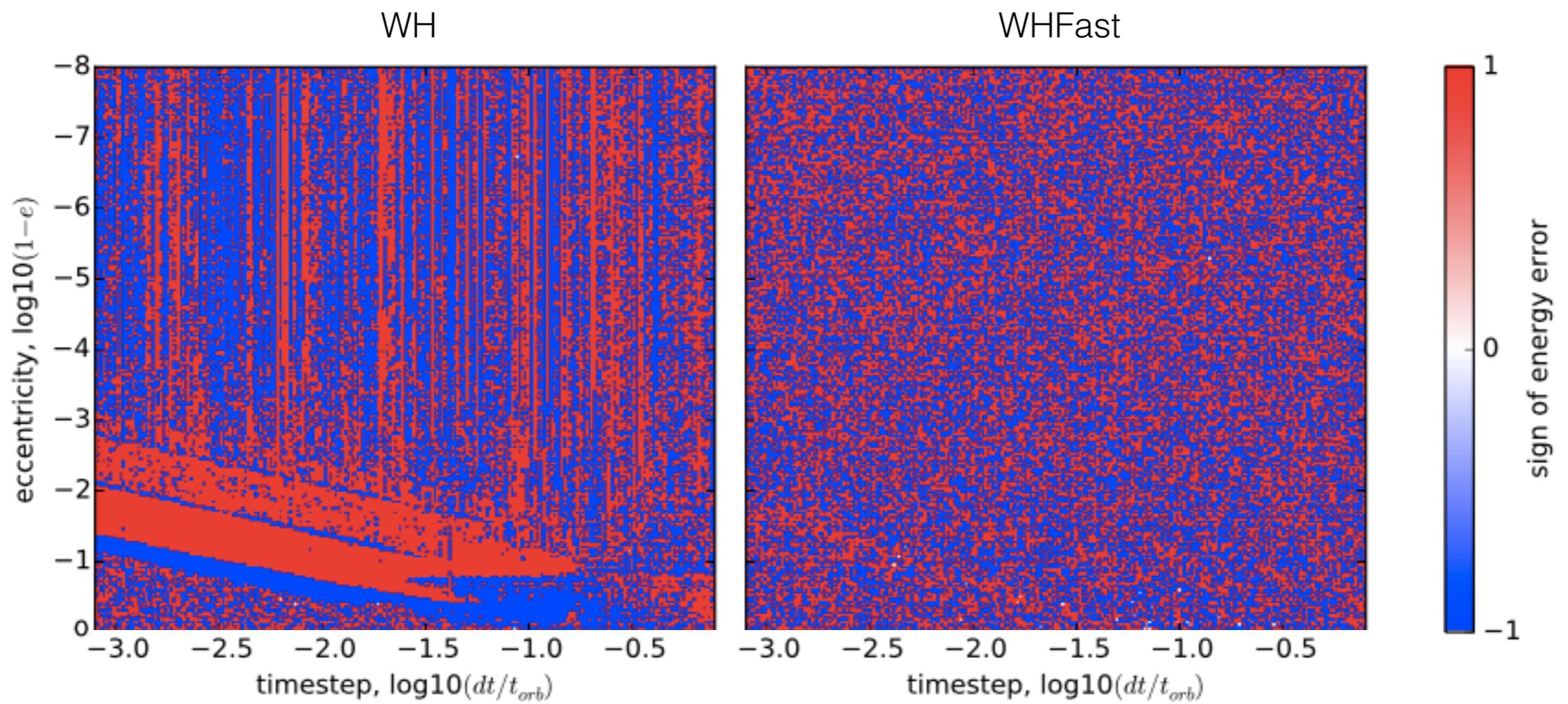
Non-symplectic
integrator

- 2nd order symplectic integrator with up to 11th order symplectic correctors
- Extremely efficient implementation of a Wisdom-Holman Map (WHM)
- WHFast can integrate variational equations
- Bias free implementation

2-body results



2-body results



3. Exact Reproducibility

Reproducible Experiments

Exact reproducibility

- N-body simulations are experiments on a computer
- They do not represent the real physical system
- Simplifications lead to a controllable experiment
- Yet none of the published results are reproducible

Why does it matter?

- Reproducibility is the hallmark of good science
- Non-reproducibility raises bar for follow-up investigations
- Wasted resources, e.g. 6.2 million CPU hours
by Laskar 2009

Reproducible Experiments

Reasons for non-reproducible results

- Source code not available
- Initial conditions not available
- Machine dependent software

REBOUND + Simulation Archive

- Open Source
- Machine independent
- Exactly reproducible
- Easy to share simulations with the SimulationArchive

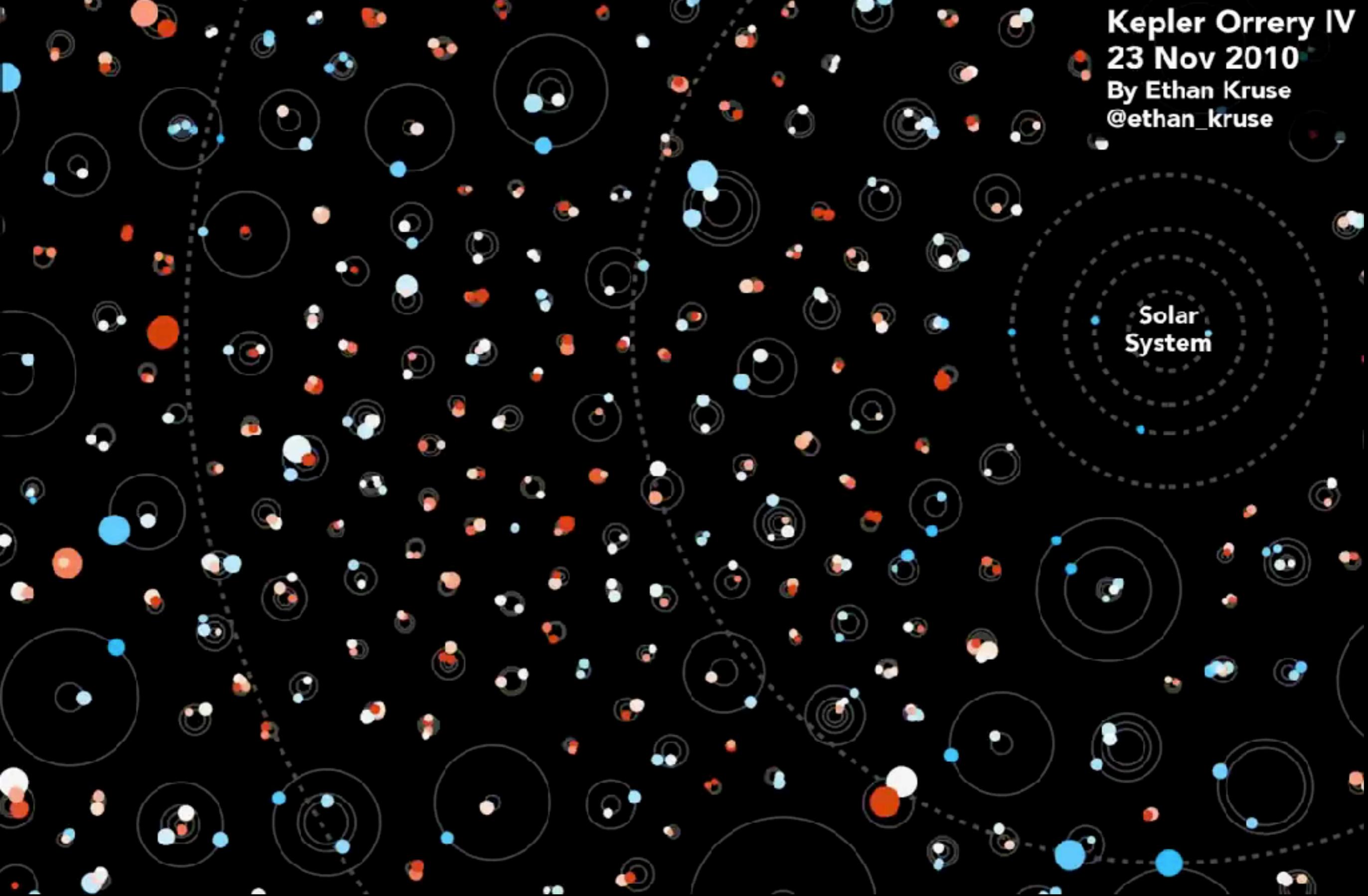
demo / rebound

rein010.utsc.utoronto.ca



4. Janus

Kepler Orrery IV
23 Nov 2010
By Ethan Kruse
@ethan_kruse



5. Machine Learning

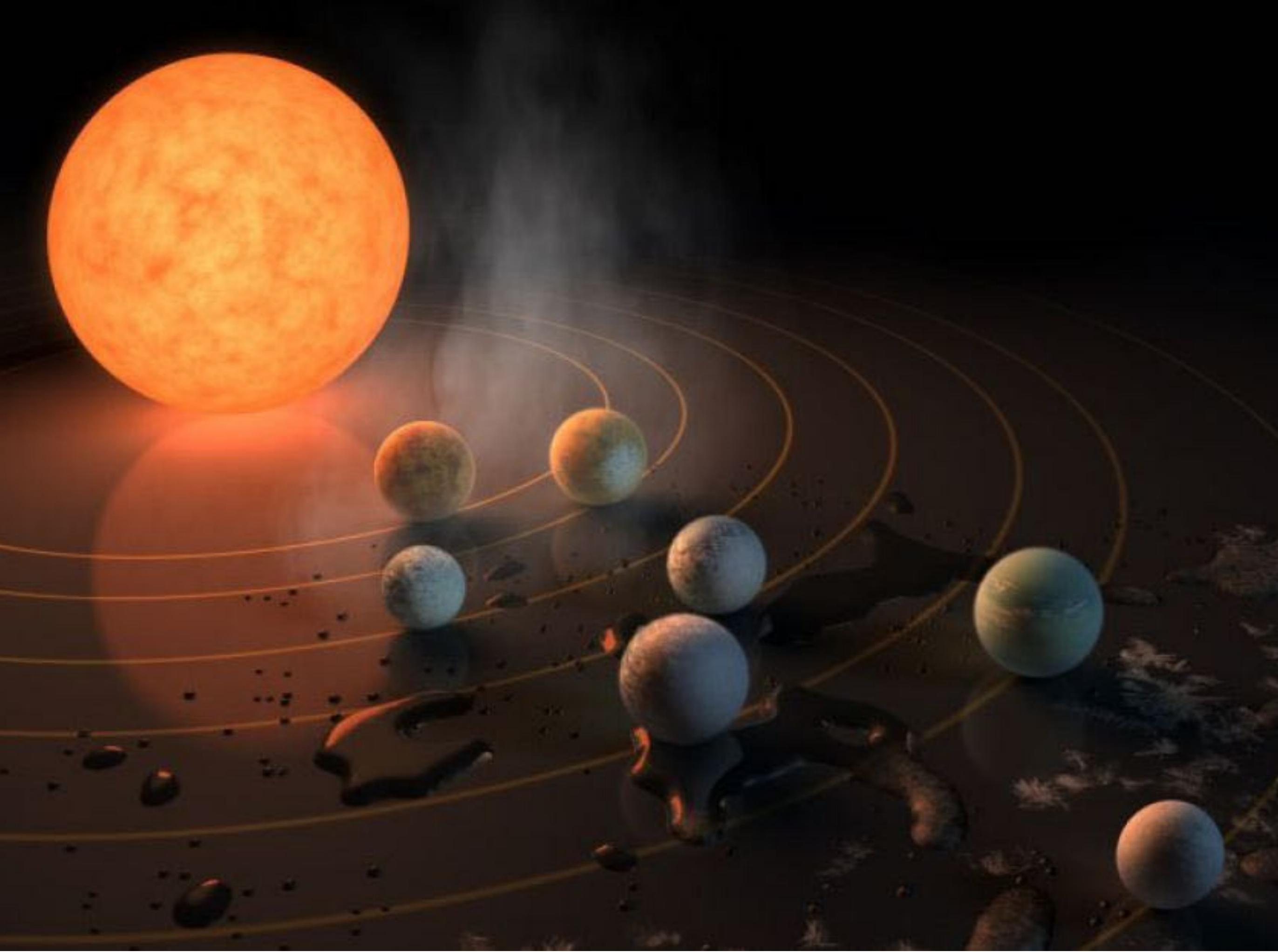
3668

x

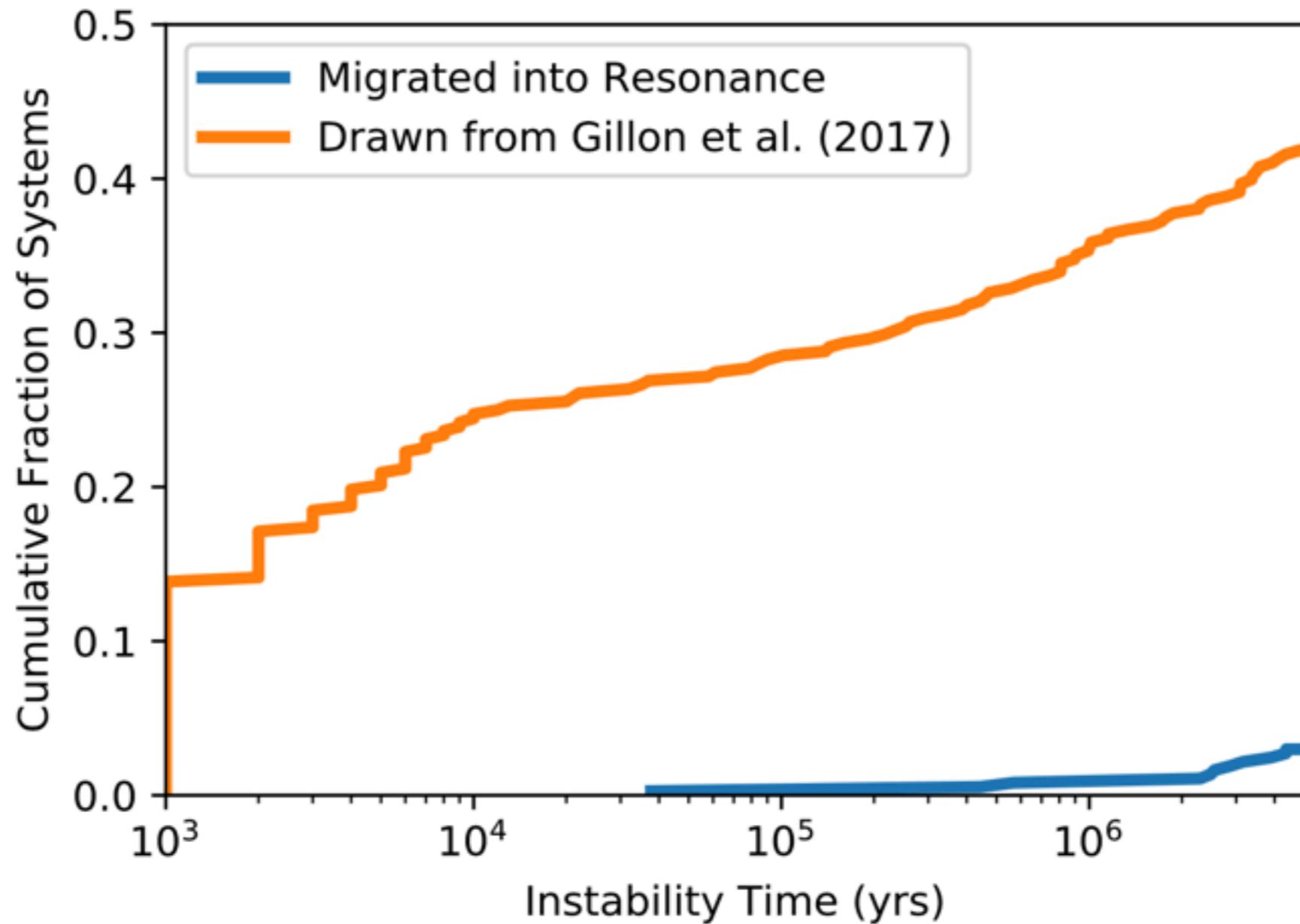
1 000 000 000 000

x

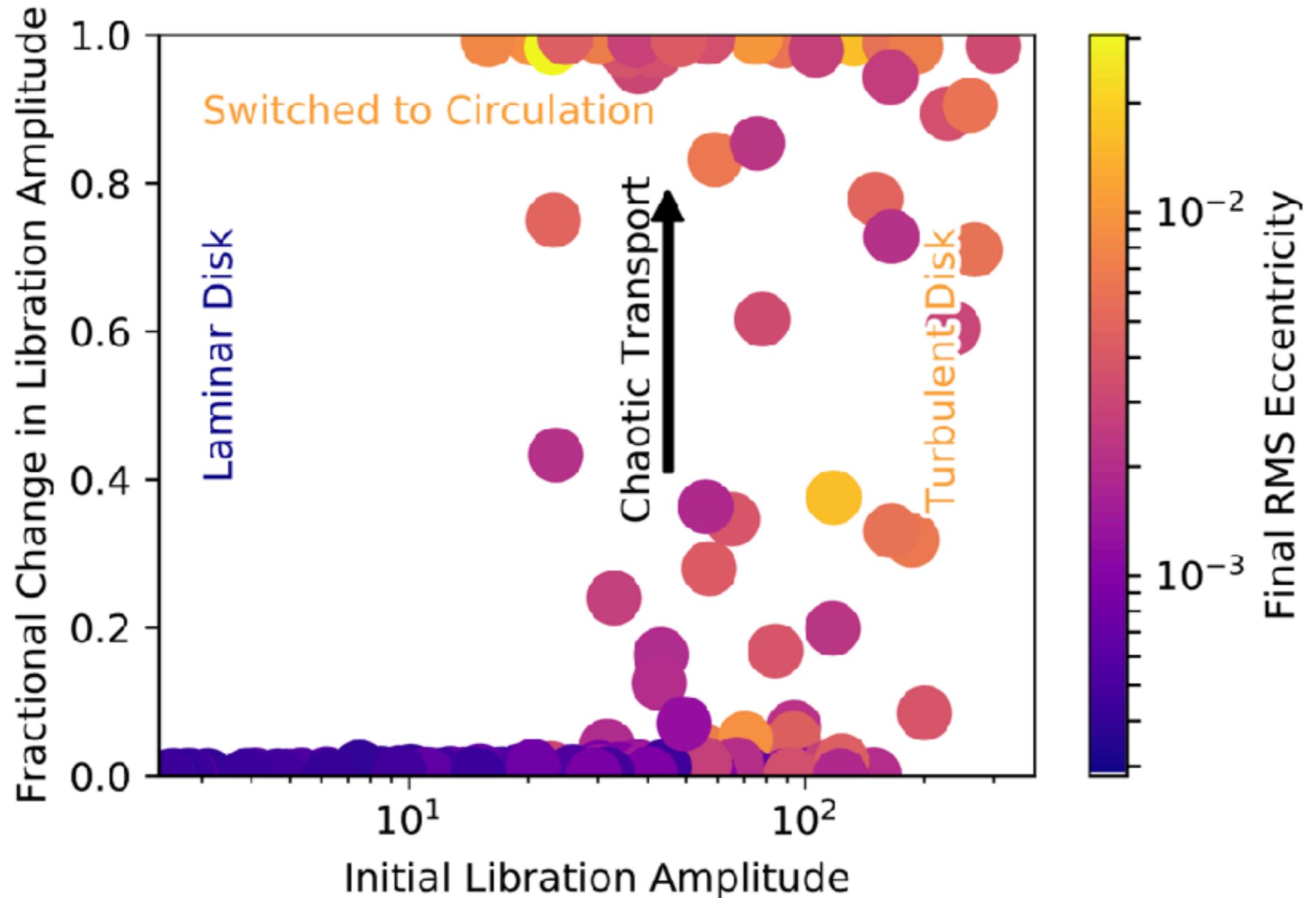
10 000



Migration saves Trappist-1



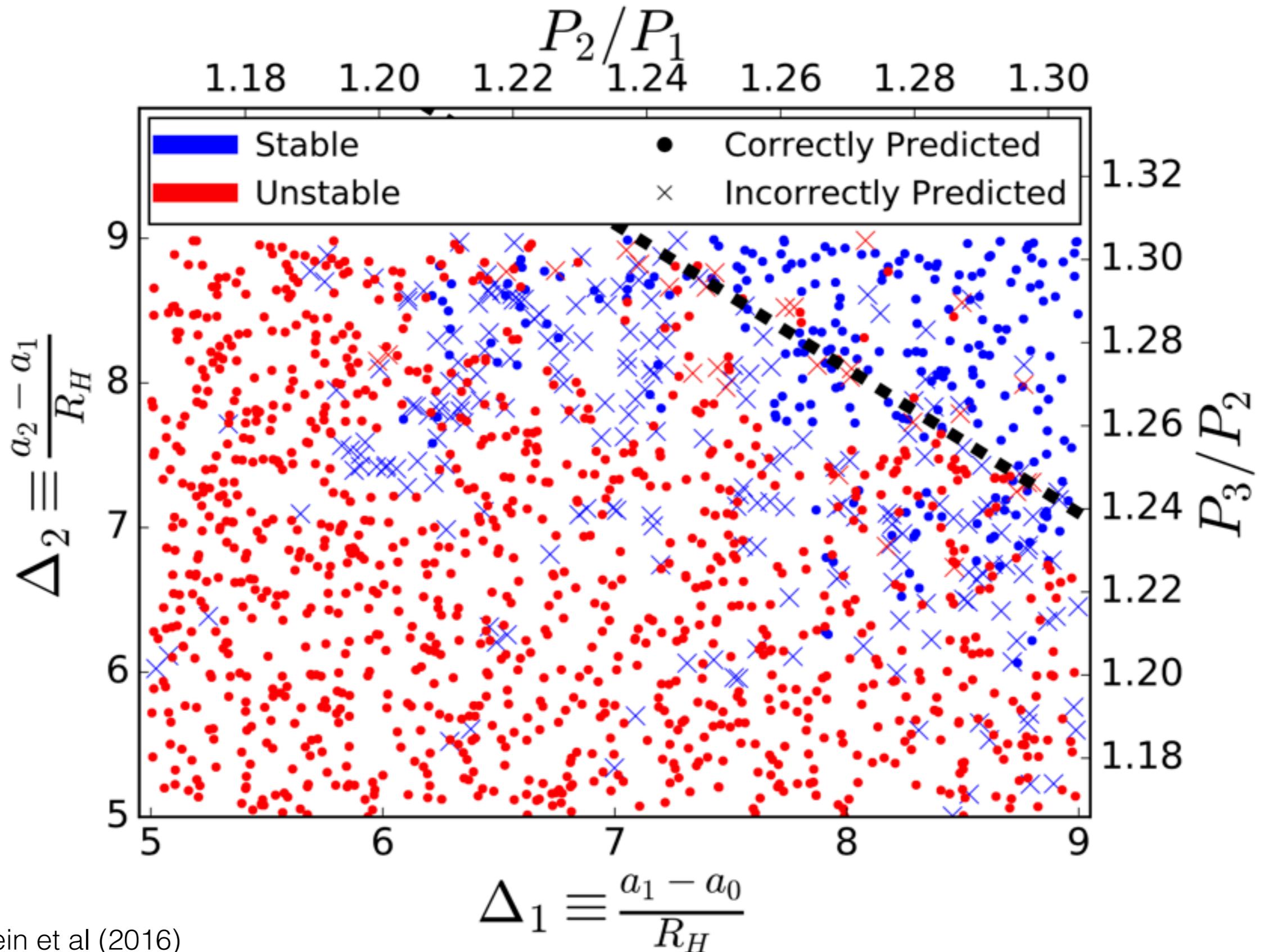
Stability implies small amount of turbulence



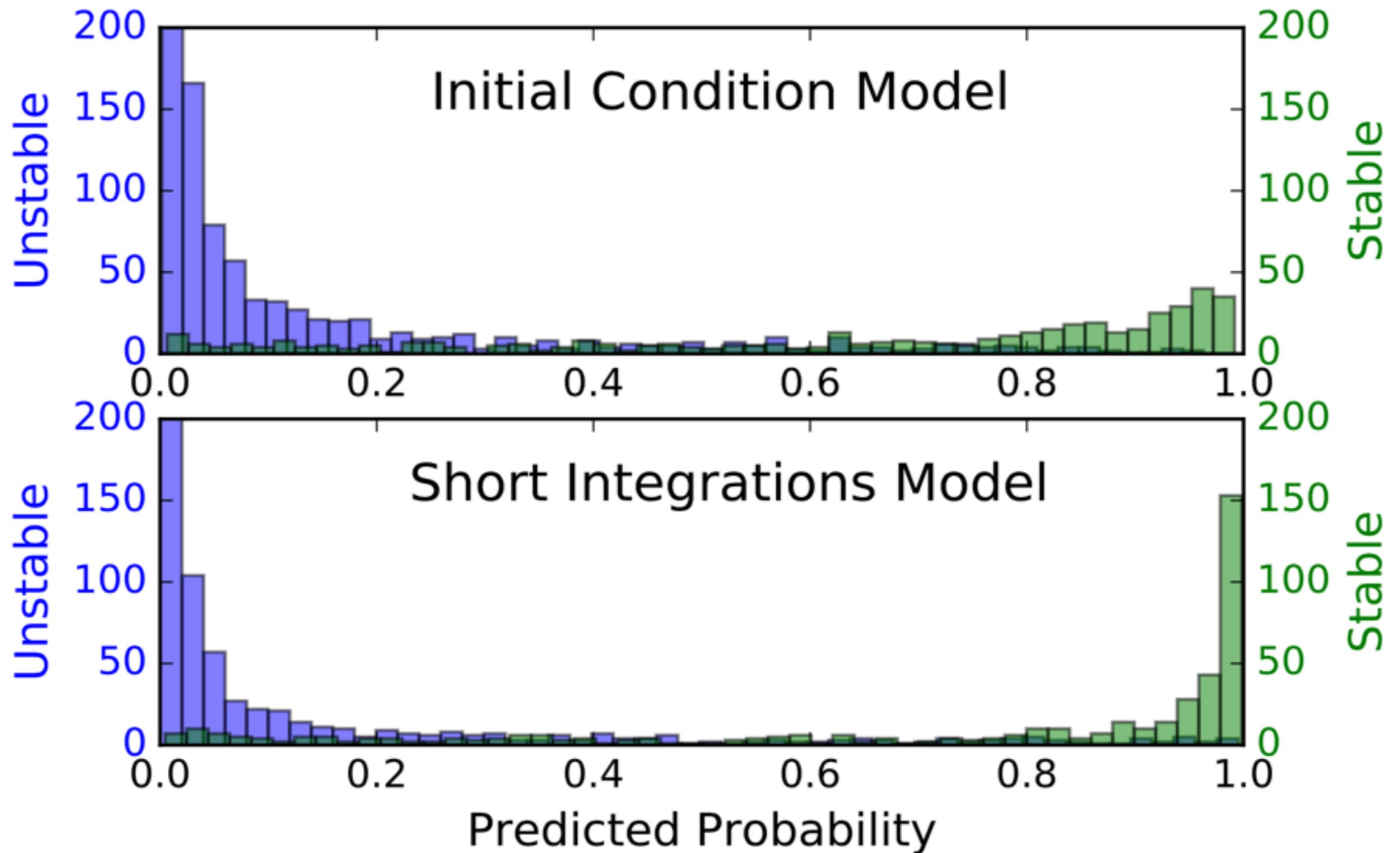
Idea

- Long term integrations of many systems still hard to do or even impossible
- Train a ML algorithm on short integrations to predict long term outcome

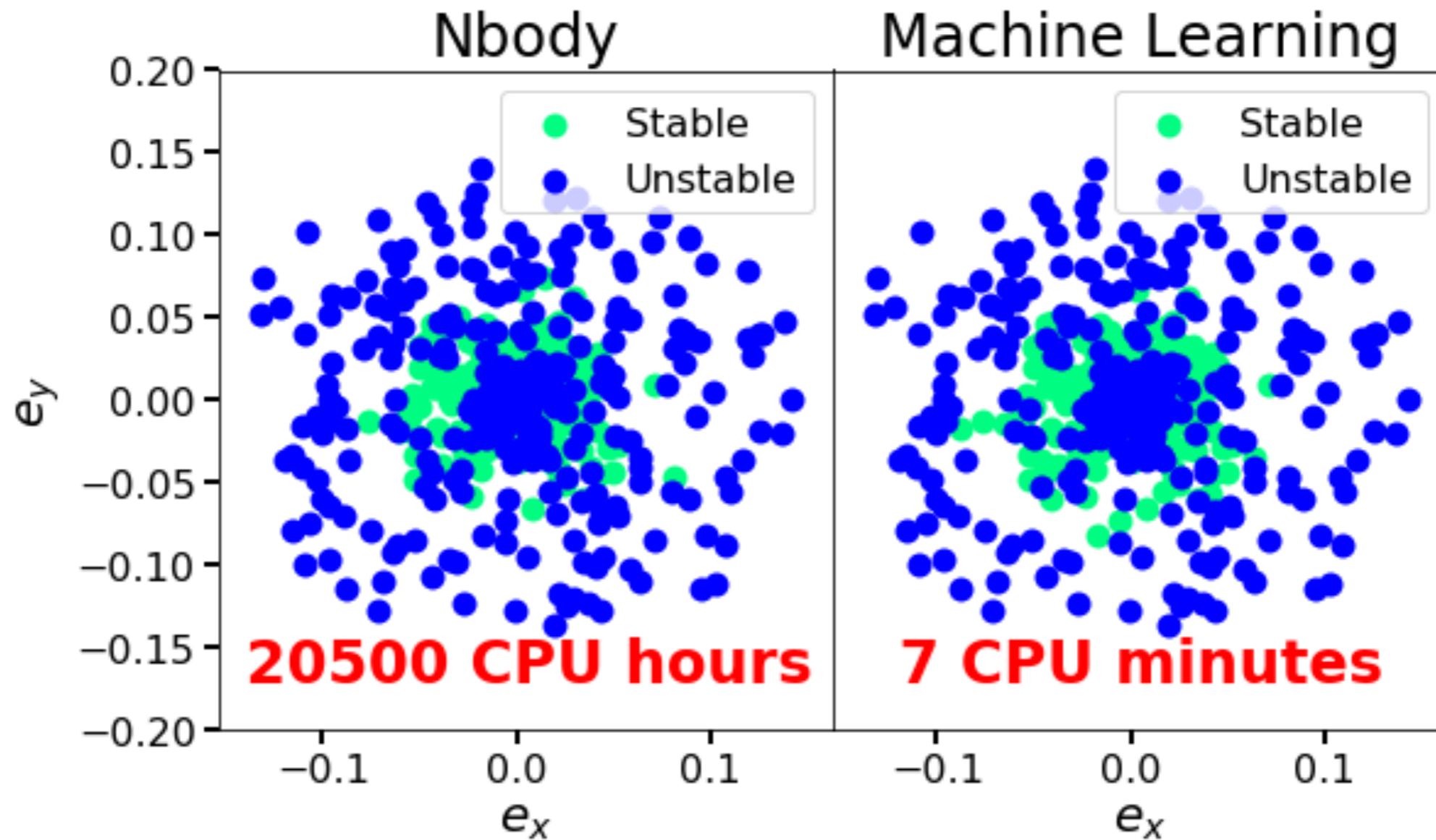
Machine Learning to Predict Stability



Accuracy of Machine Learning Algorithm

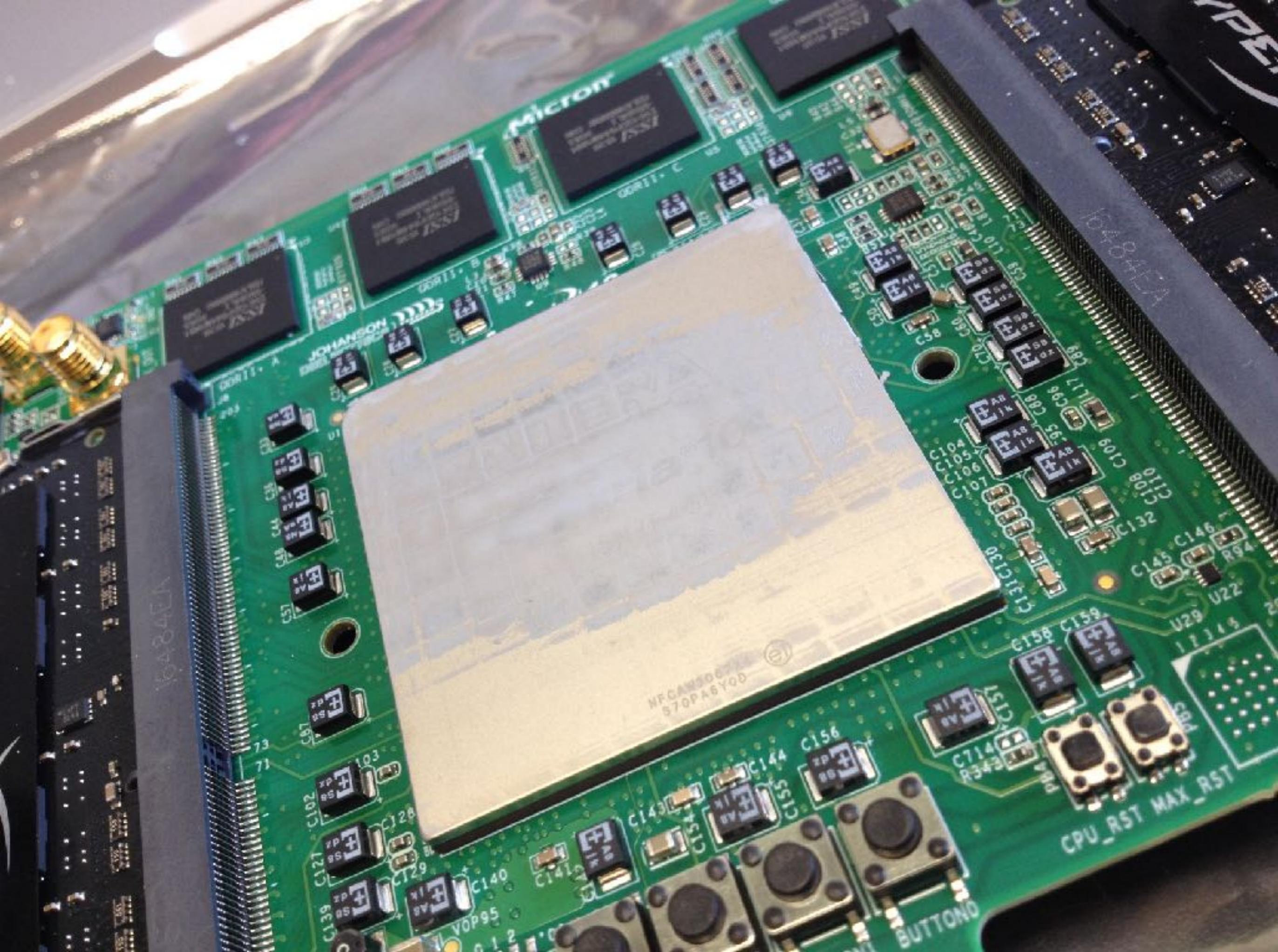


Kepler-431



Work by Dan Tamayo. Watch out for a new paper!

How to generate
training data



Micron

DDR11 - A
DDR11 - B
DDR11 - C

JOHANSON

NFGAN506720
S70PAGY0D

CPU_RST MAX_RST

BUTTON0

64834E1

64834E1

Thank you!

Try out REBOUND!

[github.com/
hannorein/rebound](https://github.com/hannorein/rebound)

REBOUND - An open-source multi-purpose N-body code

rebound v3.5.7 pypi package 3.5.7 license GPL build canceled coverage 90% arXiv 1110.4876 arXiv 1409.4779 arXiv 1506.0103

arXiv 1603.03424 docs failing launch binder

FEATURES

REBOUND is an N-body integrator, i.e. a software package that can integrate the motion of particles under the influence of gravity. The particles can represent stars, planets, moons, ring or dust particles. REBOUND is very flexible and can be customized to accurately and efficiently solve many problems in astrophysics. An incomplete feature list of REBOUND includes:

- Symplectic integrators (WHFast, WHFastHelio, SEI, LEAPFROG)
- High accuracy non-symplectic integrator with adaptive timestepping (IAS15)
- Support for collisional/granular dynamics, various collision detection routines
- The code is written entirely in C, conforms to the ISO standard C99 and can be used as a thread-safe shared library
- Easy-to-use Python module, installation in 3 words: `pip install rebound`
- Extensive set of example problems in both C and Python
- Real-time, 3D OpenGL visualization (C version)
- Parallelized with OpenMP (for shared memory systems)
- Distributed memory system using an essential tree for gravity and collisions (for distributed memory system)
- Visualization using `reboundviz` for visualization is optional

More information from [http://github.com/hannorein/rebound](https://github.com/hannorein/rebound)