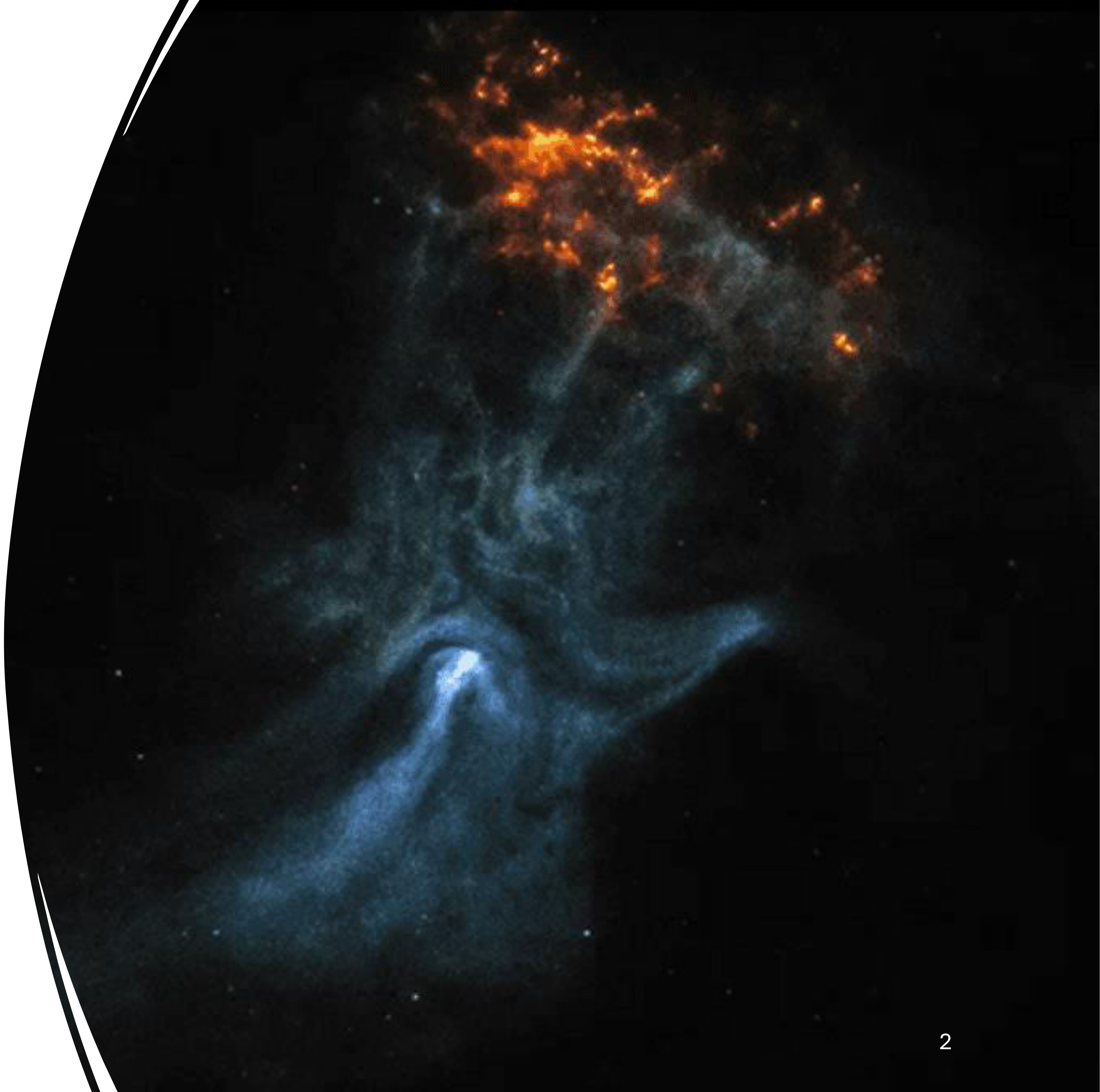
# Fast Emulation of Neutron Star Properties

Brendan Reed

# Overview

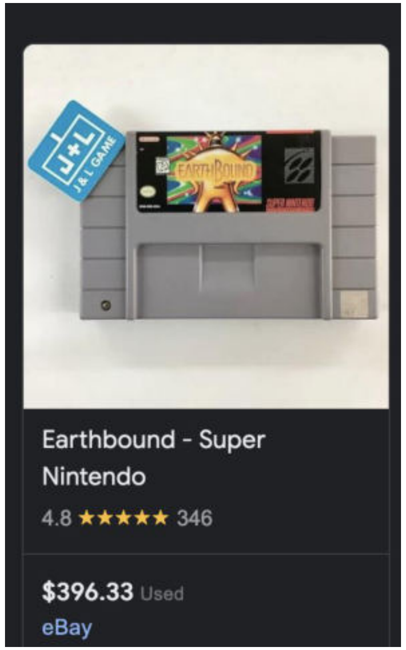
- What are emulators?
- Why do we want to emulate?
- Application to Neutron Stars



# What are emulators?

- Ansatz: Video game emulation


+




=

# Alternative

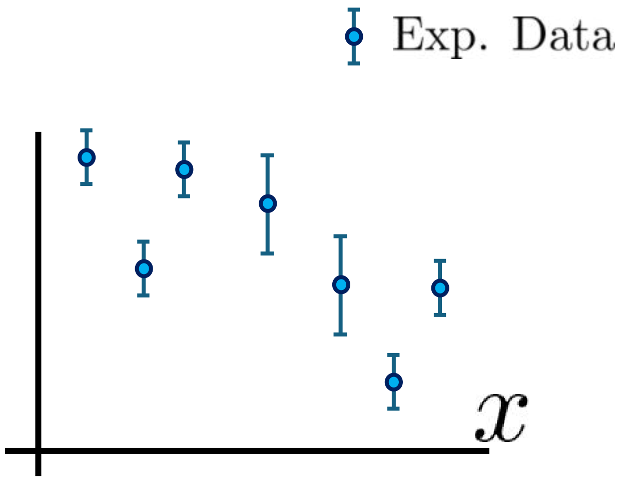
- Use emulators to play the game instead

# What Happened

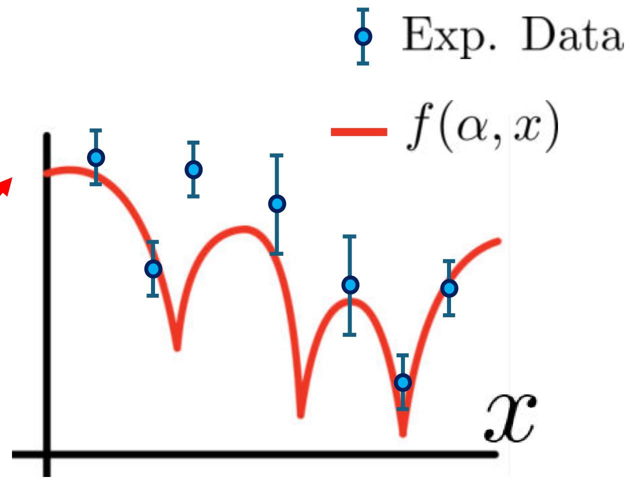
- Took the true thing (SNES)
- Created a program (or model) which needs input (game file)
- Provided input then outputs (emulates) playing the real game

- Provides much easier method of playing than using the real thing
- With less input, can still get desired result

# Back to Physics



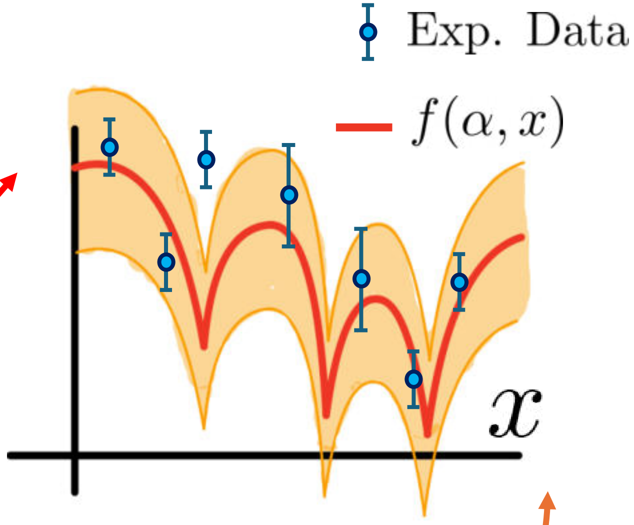
# Back to Physics




Parameters

Physical model $f(\alpha, x)$

# Back to Physics




Parameters

Physical model $f(\alpha, x)$

Posterior distribution $P(\alpha|\mathbf{Y}) = \dfrac{P(\mathbf{Y}|\alpha)P(\alpha)}{P(\mathbf{Y})}$

# Back to Physics




**Posterior distribution**

(what we want)

$$\boxed{P(\alpha|\mathbf{Y})} = \frac{\boxed{P(\mathbf{Y}|\alpha)}\,\boxed{P(\alpha)}}{\boxed{P(\mathbf{Y})}}$$

**Likelihood**

(what we think happened)

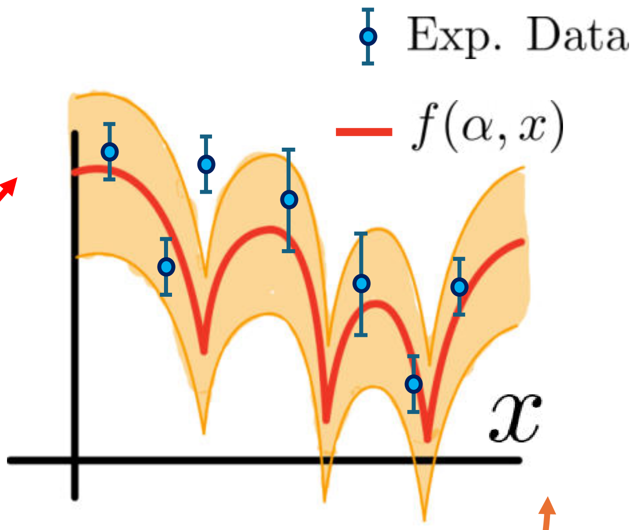
**Prior**

(what we think we know)

**Evidence**

(how much sense all makes)

# Back to Physics




Parameters

Physical model $f(\alpha, x)$

Posterior distribution $$P(\alpha|\mathbf{Y}) = \frac{P(\mathbf{Y}|\alpha)P(\alpha)}{P(\mathbf{Y})}$$

# Back to Physics

“High Fidelity”

→

“Emulated”


Physical model $f(\alpha, x) \longrightarrow \hat{f}(\alpha, x)$

Posterior distribution $P(\alpha|\boldsymbol{Y}) \longrightarrow \hat{P}(\alpha|\boldsymbol{Y})$

# Back to Physics

“High Fidelity” → “Emulated”


Physical model: $f(\alpha, x) \rightarrow \hat{f}(\alpha, x)$

Posterior distribution: $P(\alpha|\mathbf{Y}) \rightarrow \hat{P}(\alpha|\mathbf{Y})$

Model Order Reduction:

- Reduced basis methods (~eigenvector continuation)
- Dynamic Mode Decomposition
- SINDy and NIF
- ...

Gaussian Process

Neural Networks

...

Normalizing flows

Chaos expansion

Variational Bayes

...

# Back to Physics

"High Fidelity"

"Emulated"


Physical model $f(\alpha, x) \rightarrow \hat{f}(\alpha, x)$

Posterior distribution $P(\alpha|\mathbf{Y}) \rightarrow \hat{P}(\alpha|\mathbf{Y})$

Model Order Reduction:

Reduced basis methods (~eigenvector continuation)

Dynamic Mode Decomposition

SINDy and NIF

...

Gaussian Process

Neural Networks

...

Normalizing flows

Chaos expansion

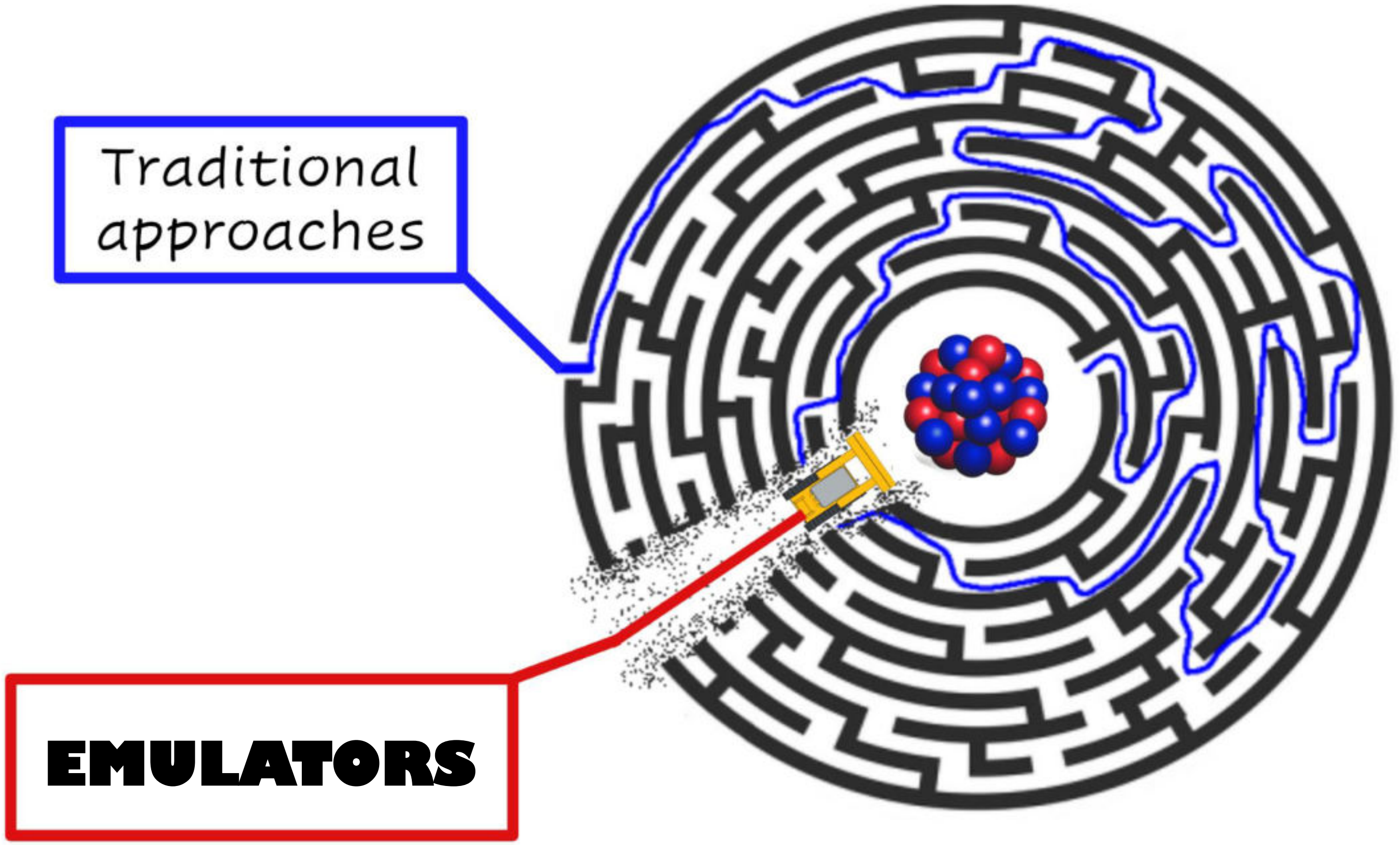
Variational Bayes

...

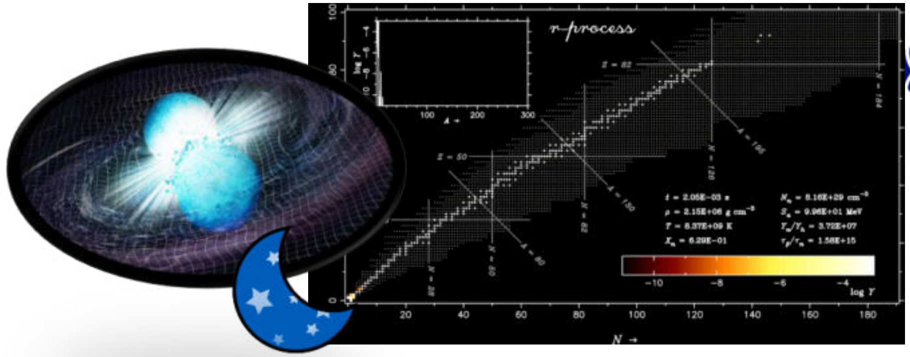
These methods can be data hungry

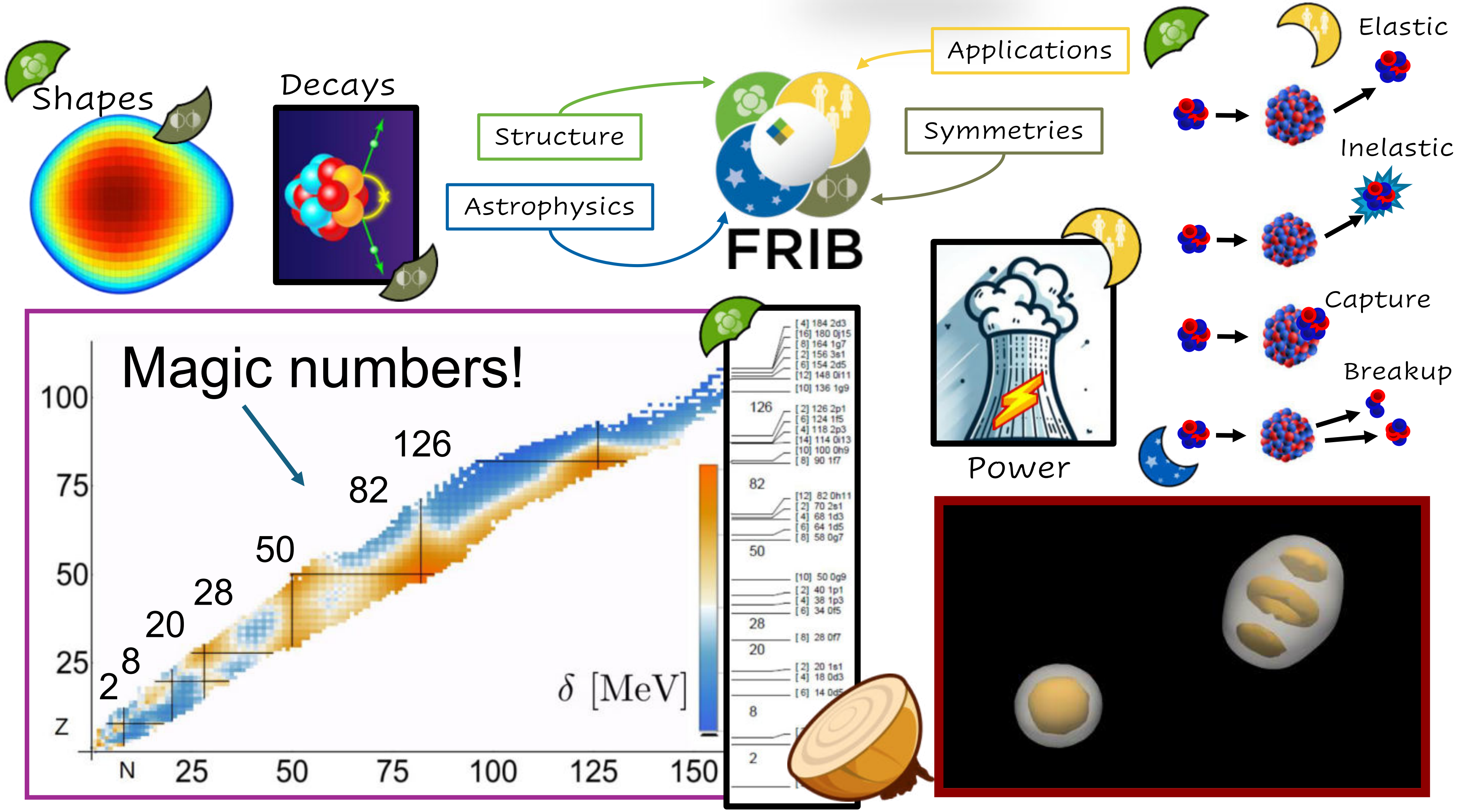
Opportunity for high performance computing to shine

Why do we want to emulate things?



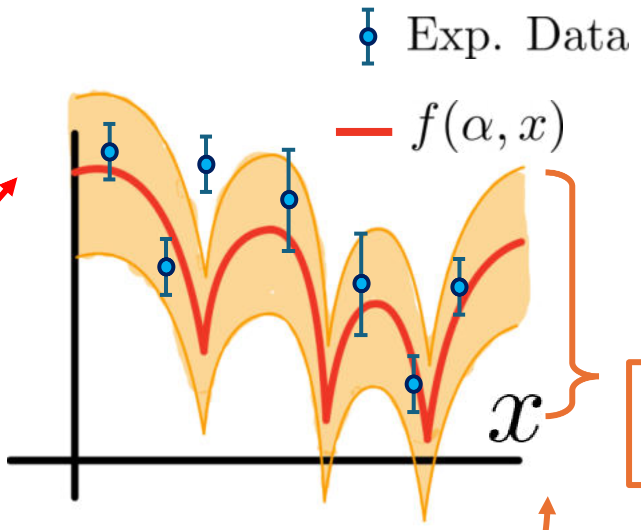
# Big questions



Los Alamos NATIONAL LABORATORY

# Dimensionality Reduction




5,000,000 samples

Parameters

Physical model $f(\alpha, x)$

Posterior distribution $$P(\alpha|\mathbf{Y}) = \frac{P(\mathbf{Y}|\alpha)P(\alpha)}{P(\mathbf{Y})}$$

$f(\alpha, x)$

1 second

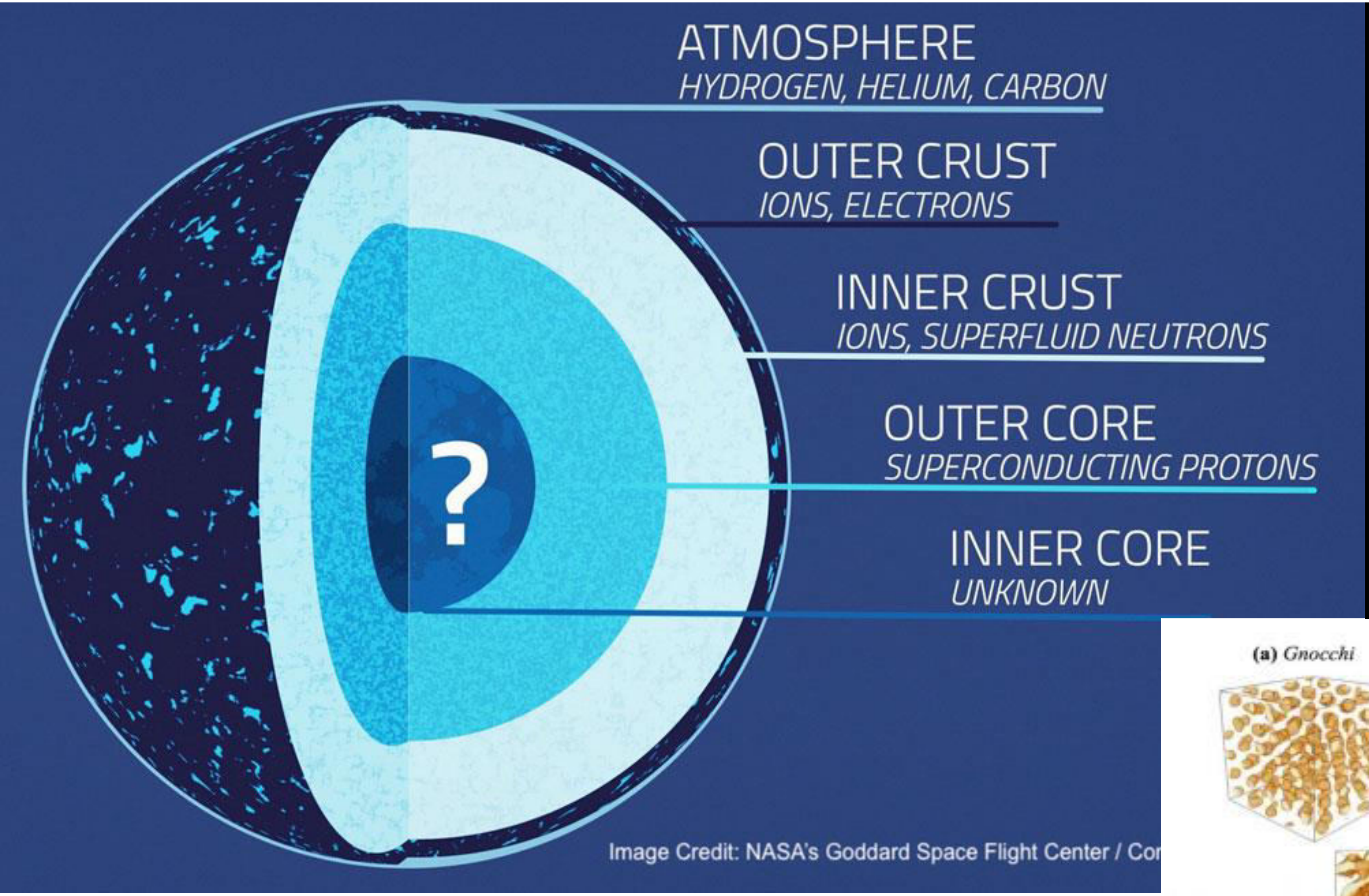
2 months

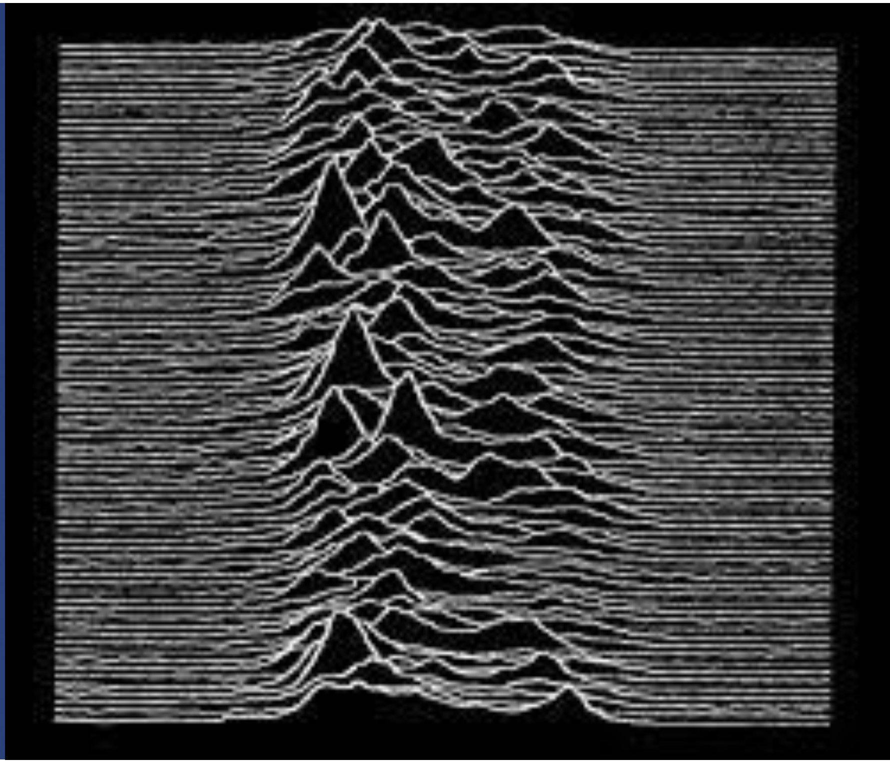
1 hour

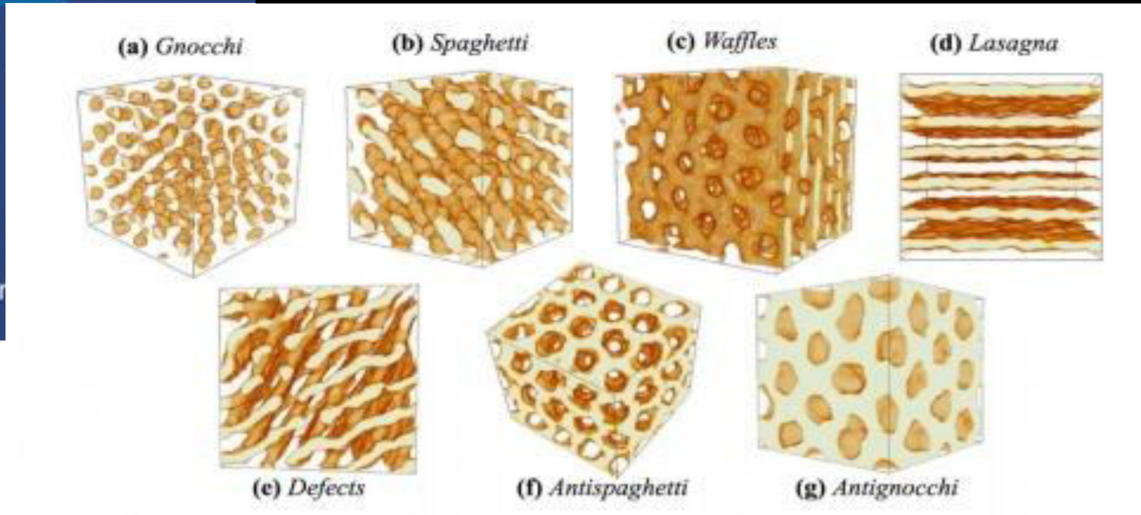
1/2 century

# One Application: Neutron Stars

Los Alamos
NATIONAL LABORATORY

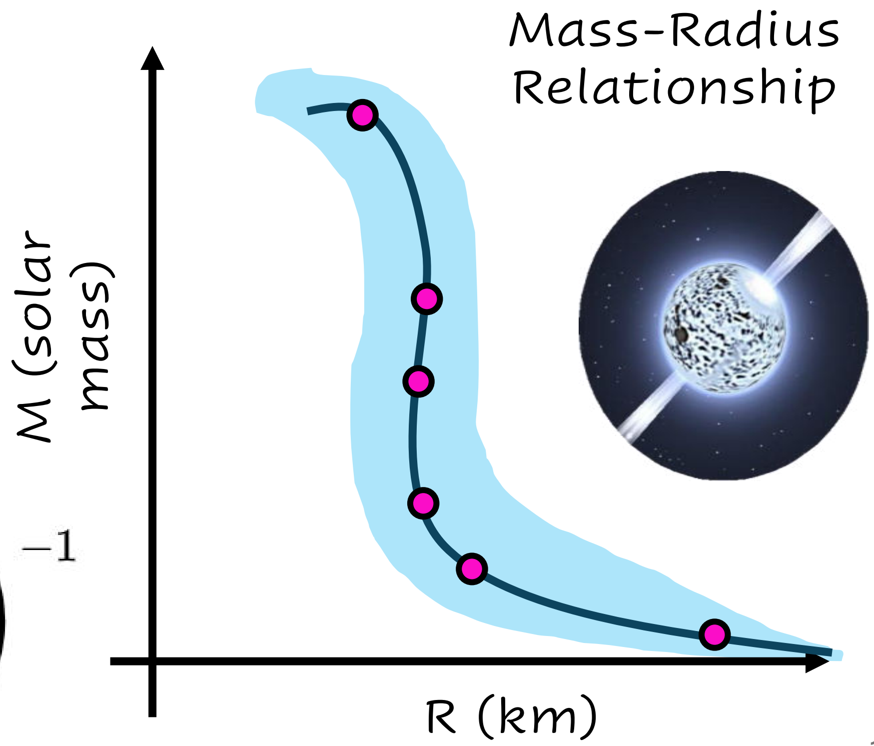




# Math

- Structure is governed by GR
  - Tolman-Oppenheimer-Volkoff equations
- Provide an equation of state → solve on a grid




$$\frac{dP}{dr} = -\frac{Gm}{r^2}\rho\left(1+\frac{P}{\rho c^2}\right)\left(1+\frac{4\pi r^3 P}{mc^2}\right)\left(1-\frac{2Gm}{rc^2}\right)^{-1}$$

$$\frac{dM}{dr} = 4\pi\rho r^2$$

- Define some step size and central pressure (density)
- Integrate outward until P=0

# More Math

- Often also want to solve for Tidal Deformability
  - Constrained from GWs
  - Describes quadrupole gravitational radiation
- Important for nuclear structure due t
  large R dependence

$$\Lambda = \frac{2}{3} k_2 \left(\frac{Rc^2}{GM}\right)^5$$

$$\begin{aligned}
k_2 = &\frac{1}{20}\xi^5(1-\xi)^2\Big[(2-y_R)+(y_R-1)\xi\Big] \\
&\times\Big\{\Big[(6-3y_R)+\frac{3}{2}(5y_R-8)\xi\Big]\xi \\
&+\frac{1}{2}\Big[(13-11y_R)+\frac{1}{2}(3y_R-2)\xi+\frac{1}{2}(1+y_R)\xi^2\Big]\xi^2 \\
&+3\Big[(2-y_R)+(y_R-1)\xi\Big](1-\xi)^2\ln(1-\xi)\Big\}^{-1}
\end{aligned}$$

$$r\frac{dy}{dr} + y^2 + F(r)y + r^2 Q(r) = 0$$

$$F(r) = \frac{r - 4\pi G r^3(\rho(r) - p(r))}{r - 2GM(r)}$$

$$\begin{aligned}
Q(r) = &\frac{4\pi r}{r-2GM(r)}\left[G\left(5\rho(r)+9p(r)+\frac{\rho(r)+p(r)}{c_s^2(r)}\right)\right. \\
&\left.-\frac{6}{4\pi r^2}\right] - 4\left[\frac{G(M(r)+4\pi r^3 p(r))}{r(r-2GM(r))}\right]^2
\end{aligned}$$

# How it works

*Reducing your model in two easy steps:*

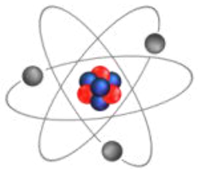
1) Find good reduced coordinates

2) Find equations for them

# How it works

*Reducing your model in two easy steps:*

1) Find good reduced coordinates

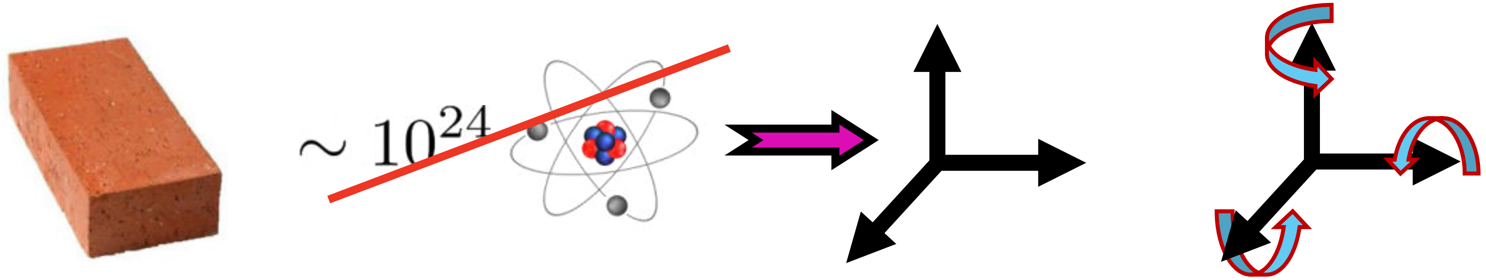
 $\sim 10^{24}$ 

2) Find equations for them

# How it works

*Reducing your model in two easy steps:*

1) Find good reduced coordinates



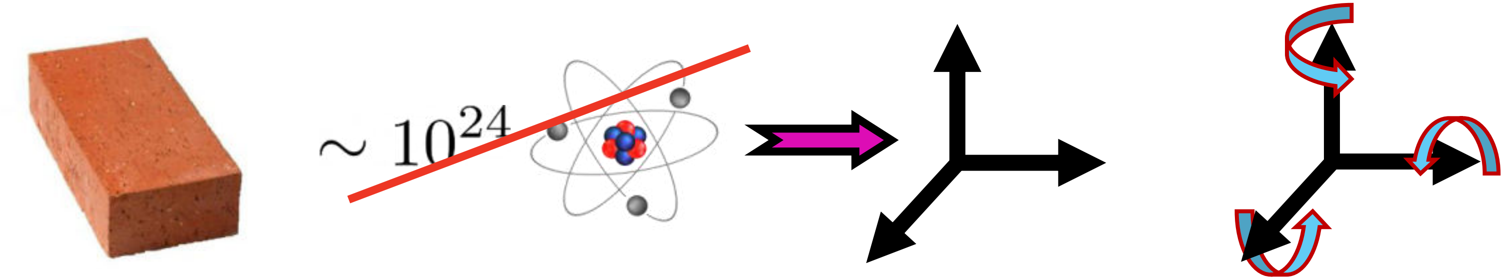

3 translations + 3 rotations

2) Find equations for them

# How it works

*Reducing your model in two easy steps:*

1) Find good reduced coordinates




3 translations + 3 rotations

2) Find equations for them

$$\frac{d}{dt}\left(\frac{d\mathcal{L}}{d\dot{q}_k}\right) - \frac{d\mathcal{L}}{dq_k} = 0$$

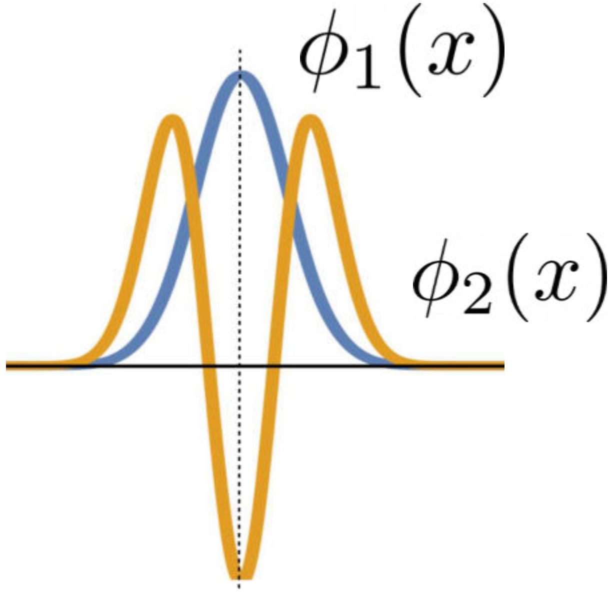
# How it works

*Reducing your model in two easy steps:*

1) Find good reduced coordinates

$$\hat{\phi}(x) = \phi_0 + \sum_{k}^{n} a_k \phi_k(x)$$

Principal Component Analysis




2) Find equations for them

- Neural Network mapping
- Gaussian Processes
- Galerkin Projection
- Black Box Techniques (ask me about this)

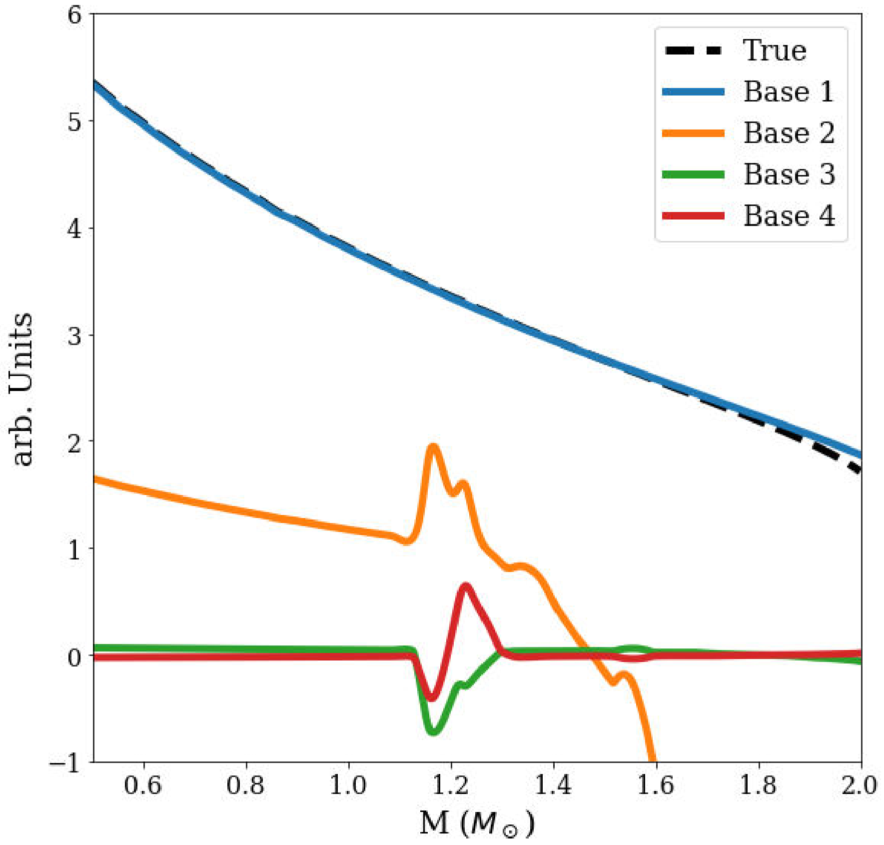
# How it works

Reducing your model in two easy steps:

1) Find good reduced coordinates

$$\hat{\phi}(x) = \phi_0 + \sum_k^n a_k \phi_k(x)$$

Principal Component Analysis



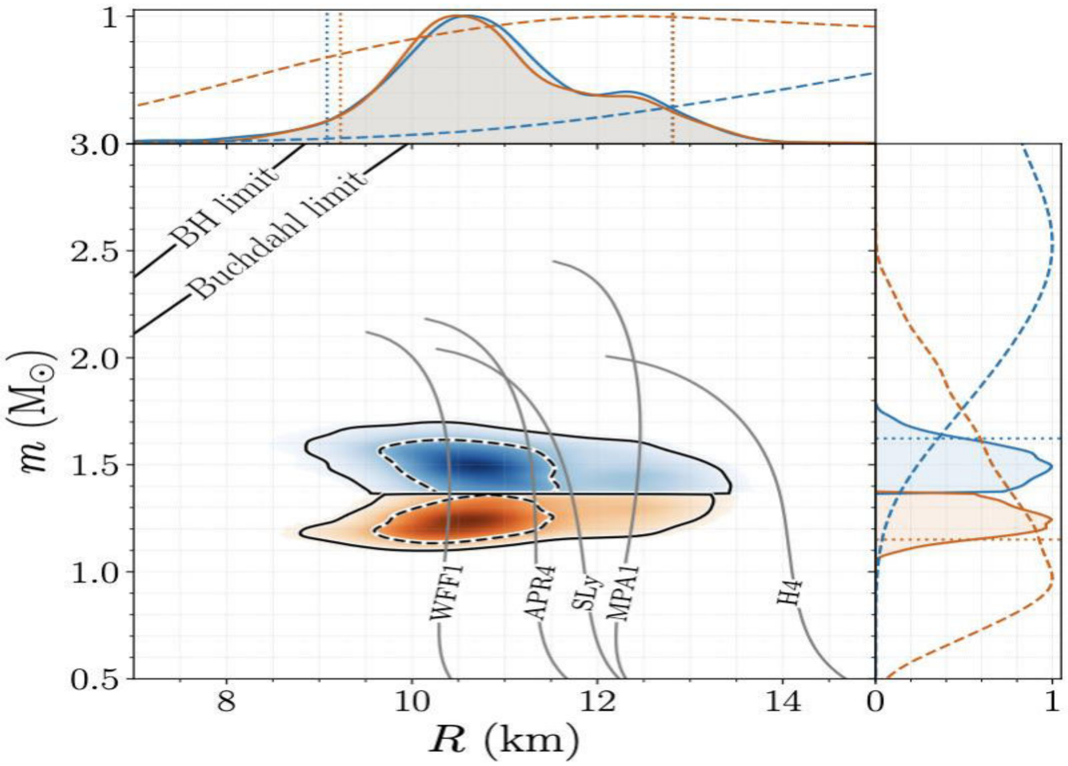

2) Find equations for them

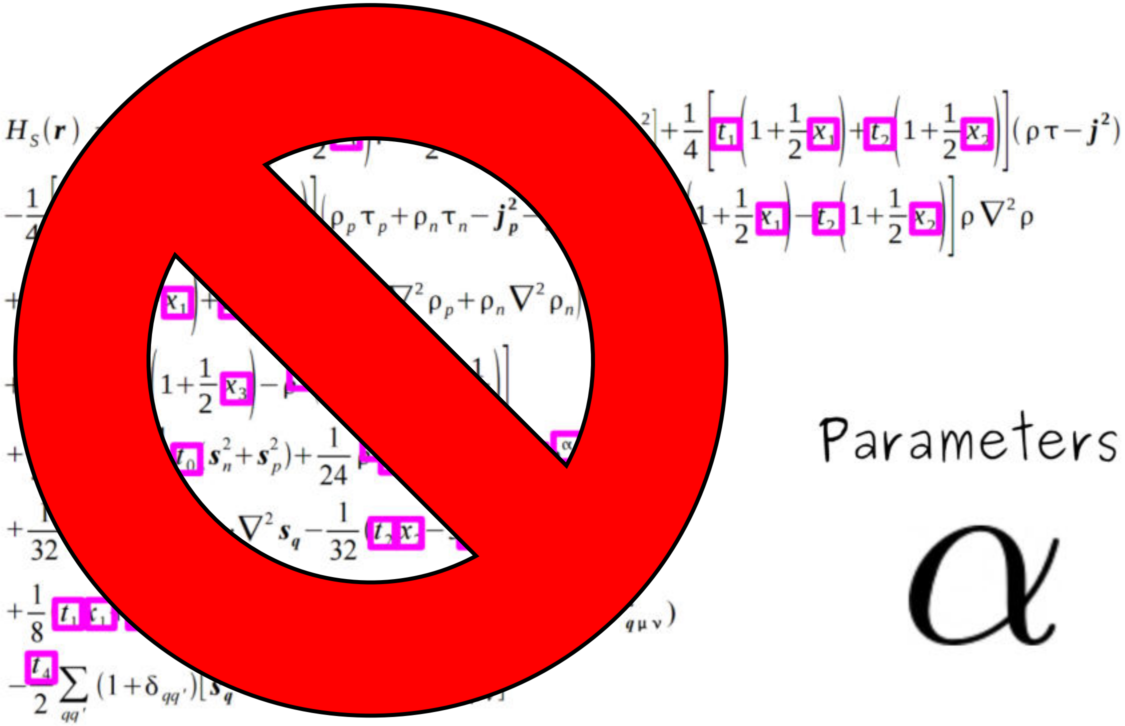
$$\hat{H}(\alpha)\,\hat{\phi}(x) = \lambda\hat{\phi}(x)$$

$$\vec{a} = \vec{\lambda}(\alpha)\lambda$$

# Finding Alphas

- Consider your data
  - What are the inputs?
  - Is there any preprocessing that can be done?
- How complex is the system?
  - Are the models stochastic or smooth?



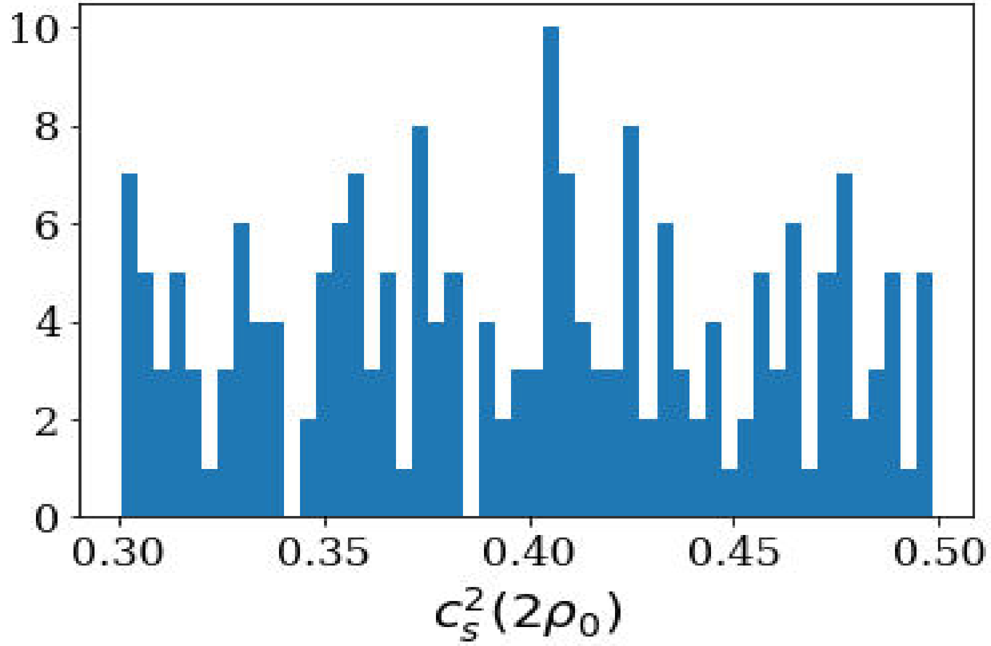



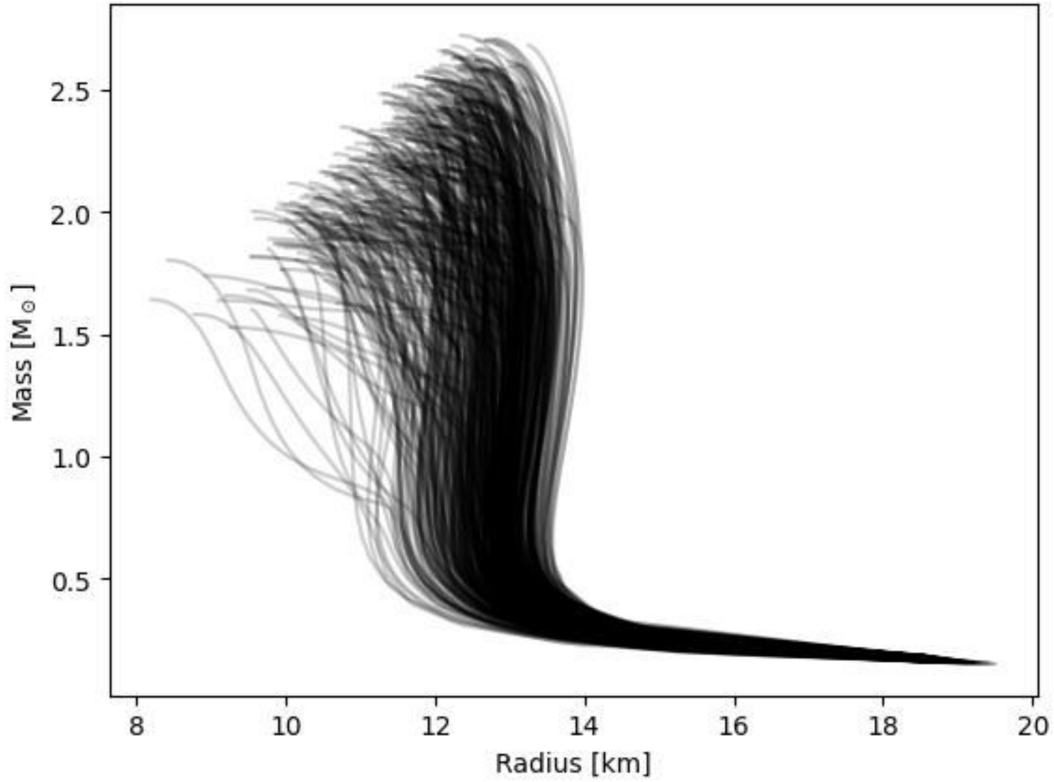
Parameters

$\alpha$

# Our Parameters

- Nuclear EOS
  - $\rho_0, \frac{E}{A}, K_{sat}, J, L, K_{sym}, c_s^2(2\rho_0), c_s^2(3\rho_0)$ …
- Use a large sample (~100,000)
- Keep some for validation (~10%)



# Building the Emulator

$$\hat{R}(x) = R_0 + \sum_{k=1} a_k^R(\alpha) R_k(x)$$

$$\hat{\Lambda}(x) = \Lambda_0 + \sum_{k=1} a_k^\Lambda(\alpha) \Lambda_k(x)$$

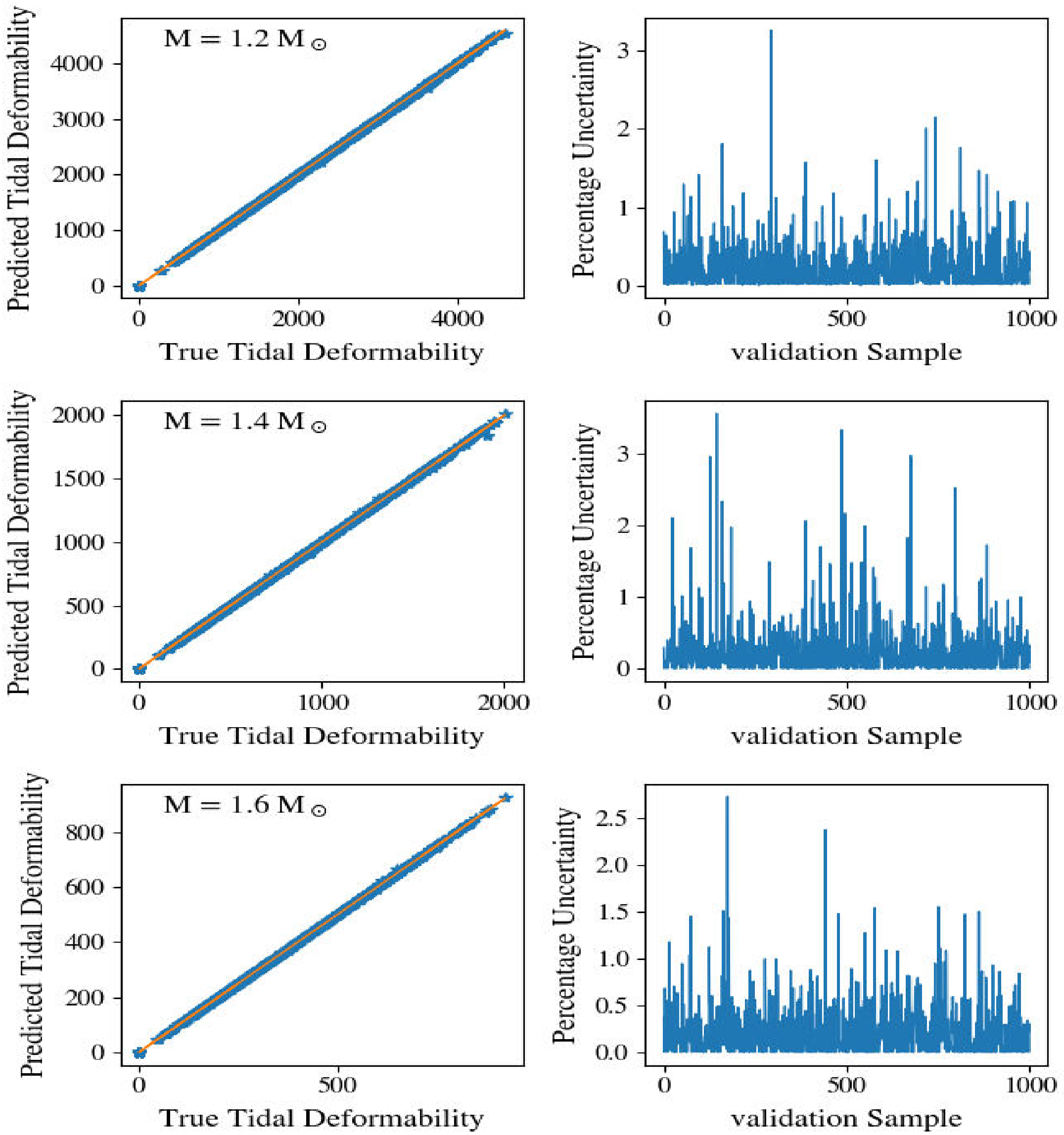
- Functions of model parameters
- Must be determined from data

- Functions determined from SVD
- Stay fixed regardless of model parameters

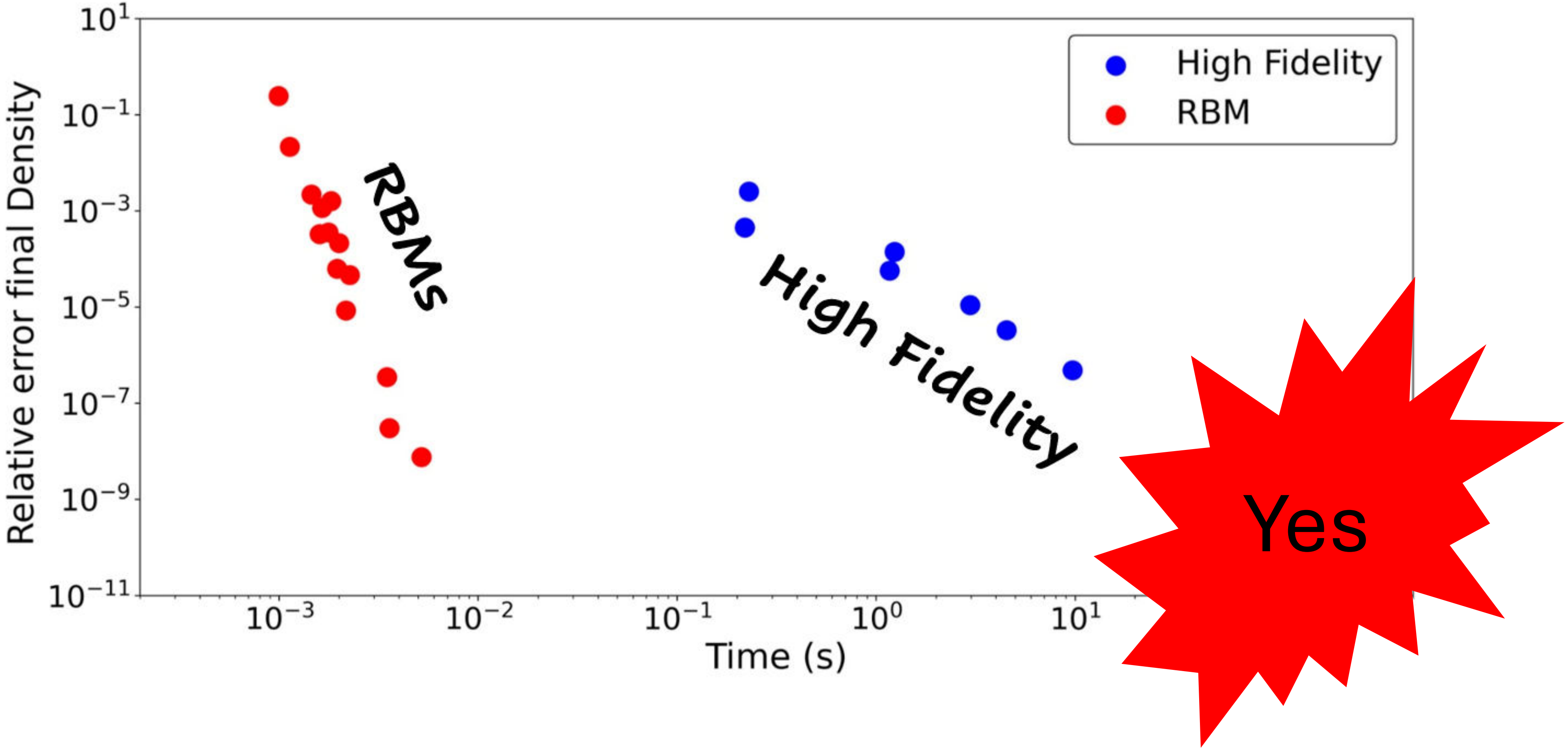
# How To Determine Coefficients

- Neural Networks
- Gaussian Processes
- Galerkin Projection
- Black Box Techniques (ask me about this)

$$\alpha \to a_k$$



# Does this Improve Things?



# Thank-yous

Los Alamos NATIONAL LABORATORY

NSF

BAND Bayesian Analysis of Nuclear Dynamics

## ASCSN Team

Kyle Godbey
Edgard Bonilla

## Friends & collaborators

Diogenes Figueroa
Frederi Viens
Jorge Piekarewicz
Paul Gueye

Dean Lee
Daniel Odell
Moses Chan
Kyle Beyer
Manuel Catacora-Rios
Filomena Nunes
Dick Furnstahl
Yukari Yamauchi
Landon Buskirk
Megan Campbell
Ruchi Garg
Aaron Phillip
Amy Anderson
Andrew Y. S.
Yanlai Chen
Eric Flynn

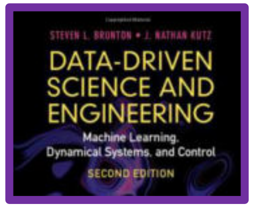
Daniel Lay
Ruoyu Cheng
Michelle Kuchera
Ágnes Mócsy
Morten Hjorth-Jensen
Alexandra Semposki
Sudhanva Lalit
Magda Kuich
Ingo Twes
Amy Lovell
Ana Posada
Witek Nazarewicz
Daniel Phillips
Dirac ...

And thank you, for listening

# Further resources

Watch on YouTube

giulianp@
frib.msu.edu

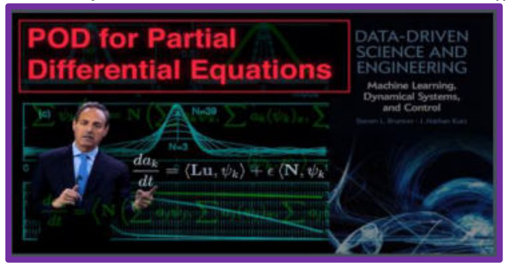



http://databookuw.com/


https://www.youtube.com/watch?v=n9Kgb72Ozgk

Dimensionality Reduction in Nuclear Physics
Presented by ASCSN

https://dr.ascsn.net




FRIB-TA Summer School: Practical Uncertainty Quantification and Emulator Development in Nuclear Physics

https://forum.ascsn.net/t/about-the-2023-frib-ta-summer-school/42

Link to slides