This will be a board talk, so the slides are not comprehensive

(Caveat emptor)

Intro to Machine Learning in Astrophysics & Cosmology

(with inescapable personal bias)

Niall Jeffrey
he/him, n.jeffrey@ucl.ac.uk

Making the impossible, possible

Astrophysics and Cosmology: questions

Astrophysics and Cosmology: questions

AGN feedback

Λ?

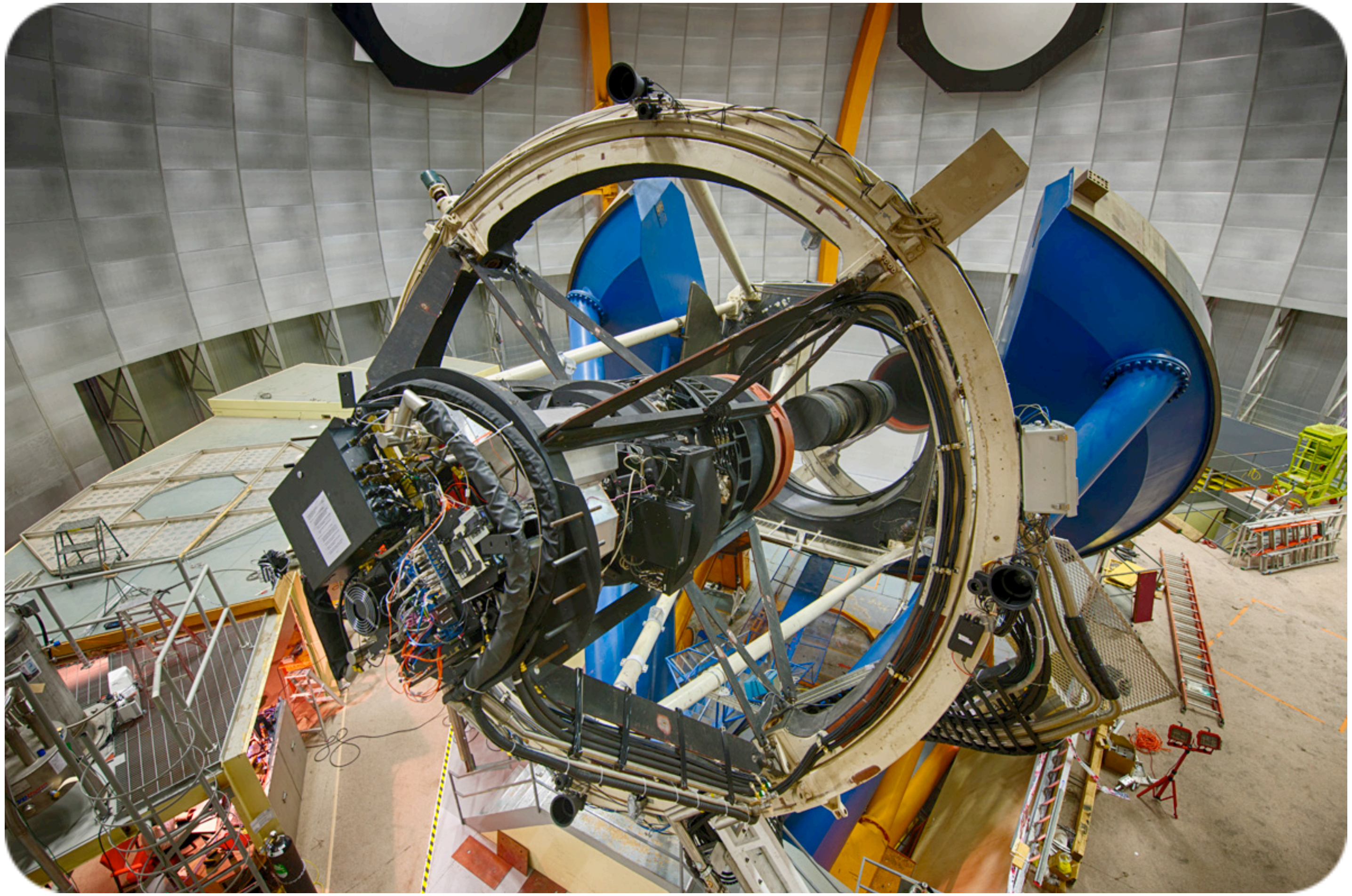
H_0?

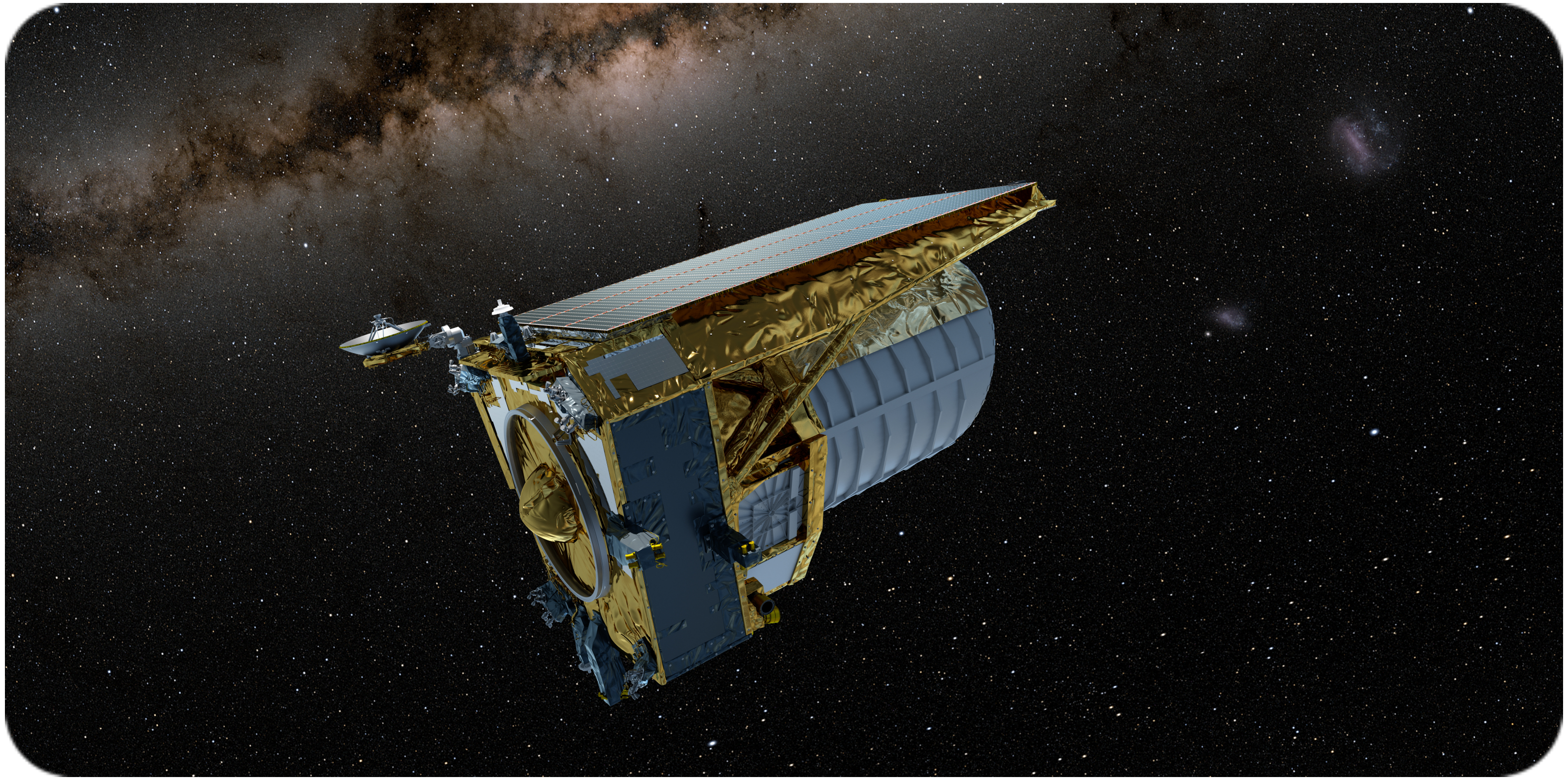
Dust modelling

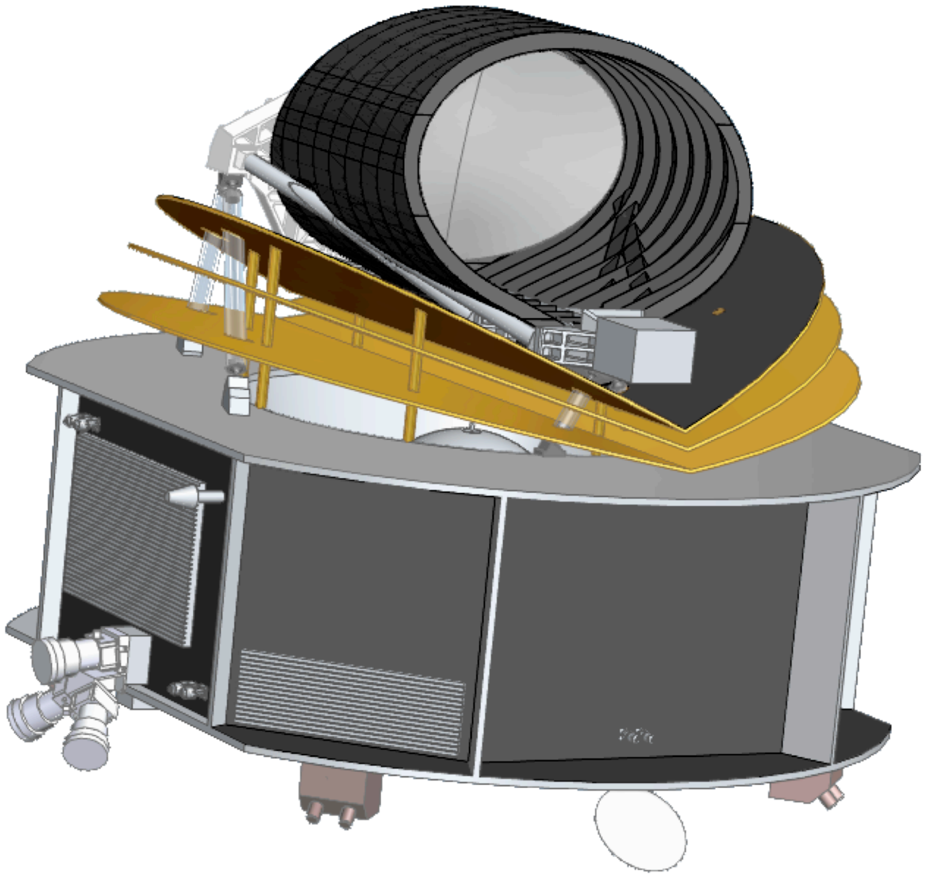
Galaxy-halo connection

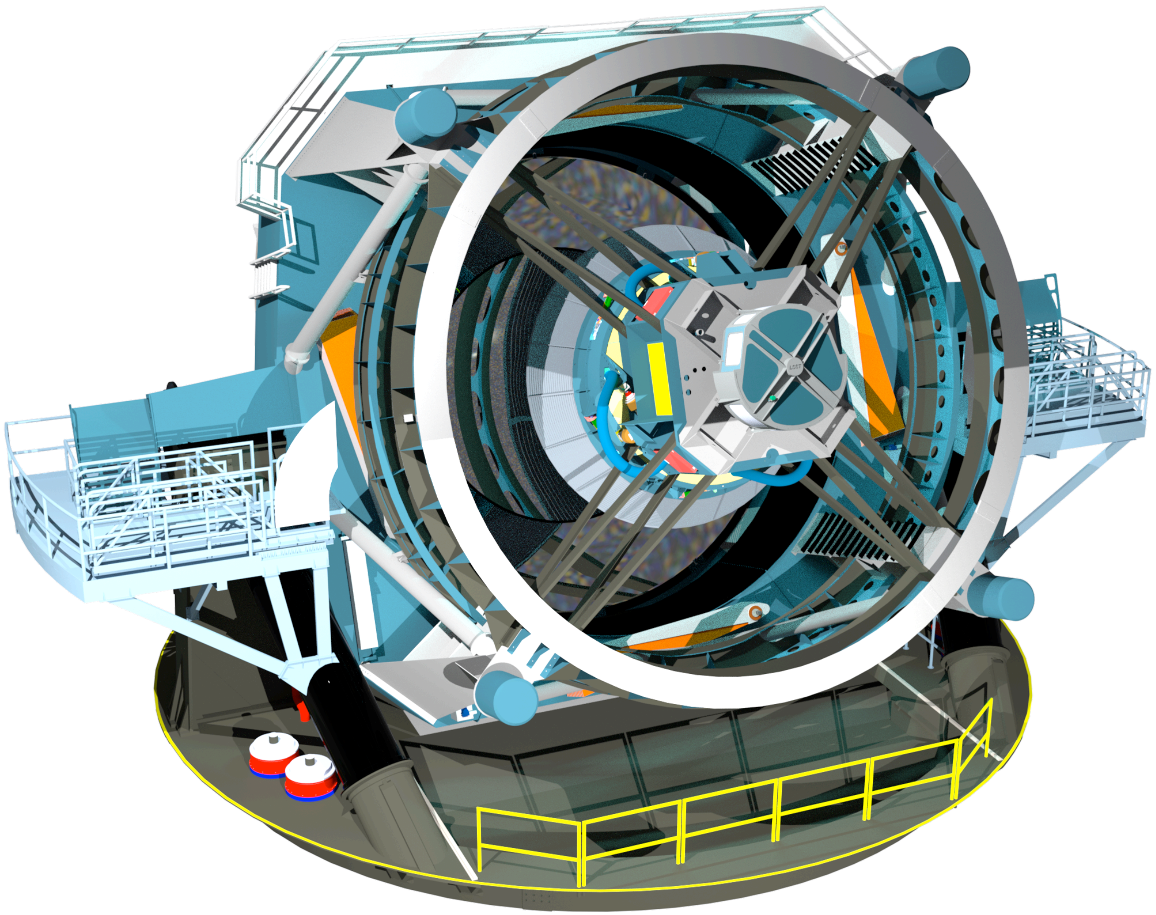
Astrophysics: data

DECam









Data x, Model M

Model parameters

$$p(\theta \mid x, M)$$

Data x, Model M

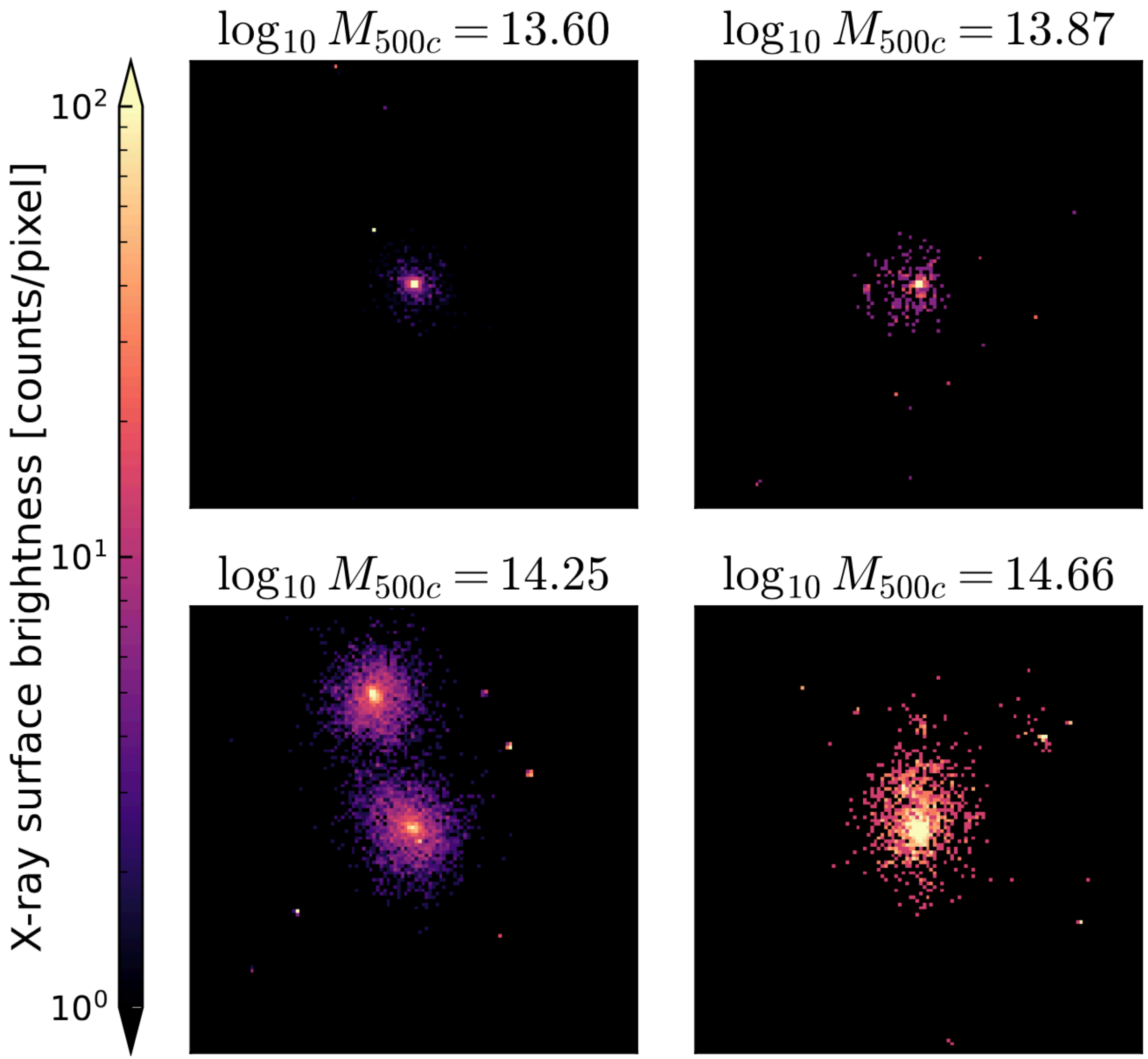
Model parameters

$$p(\theta | x, M)$$

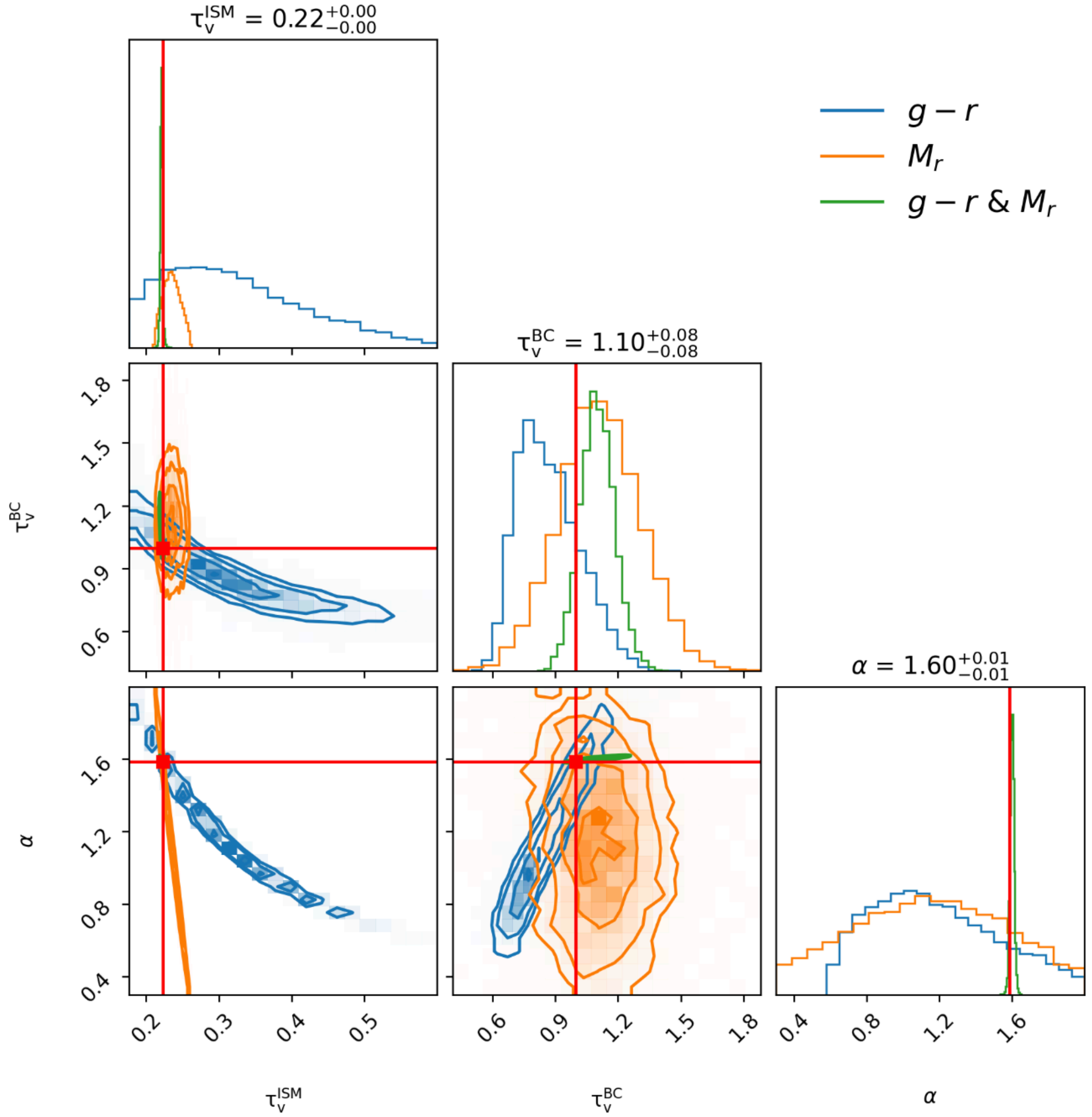
Data x, Model M

$$p(M_1 | x) \text{ vs } p(M_0 | x)$$

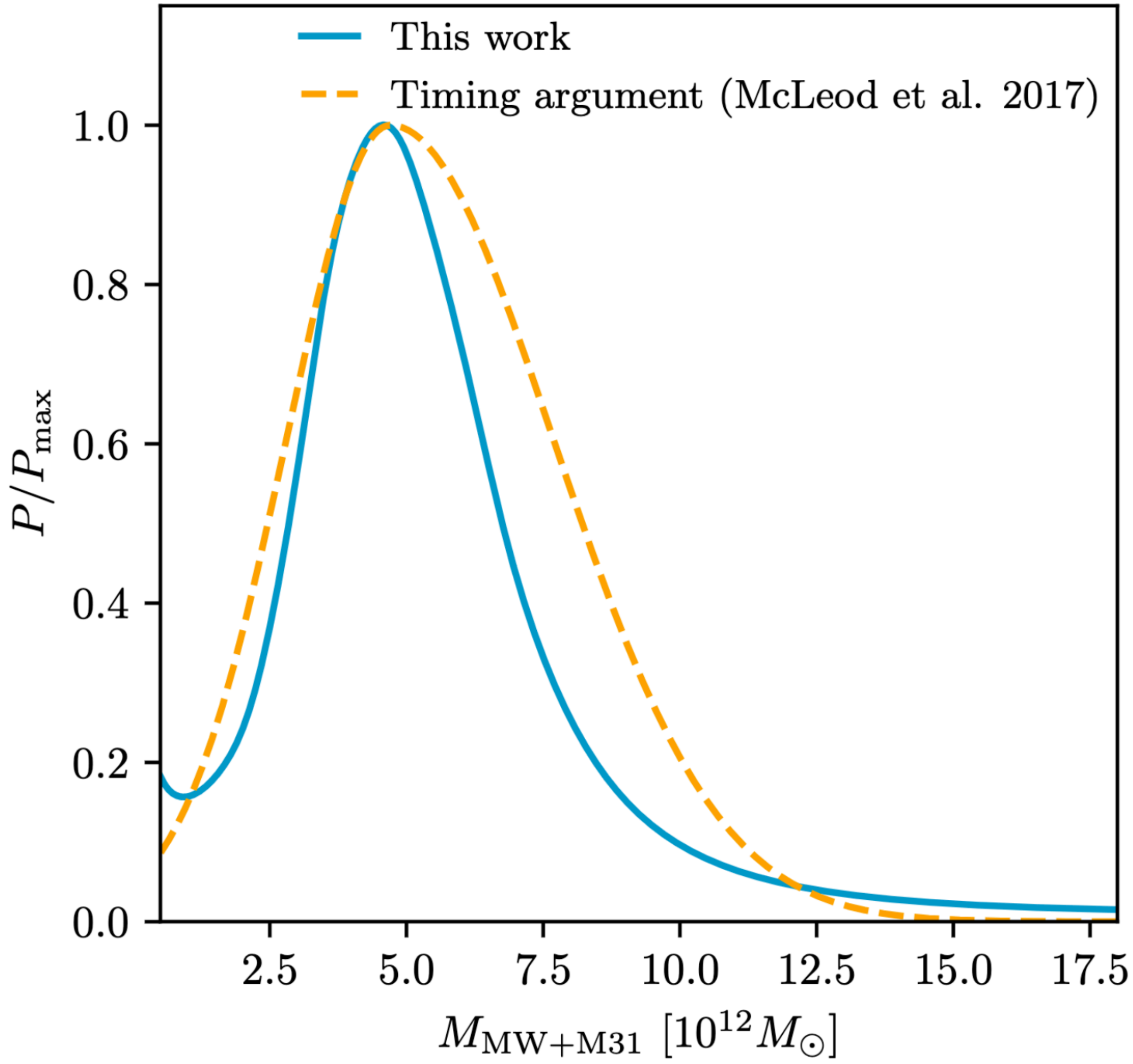
Some examples....



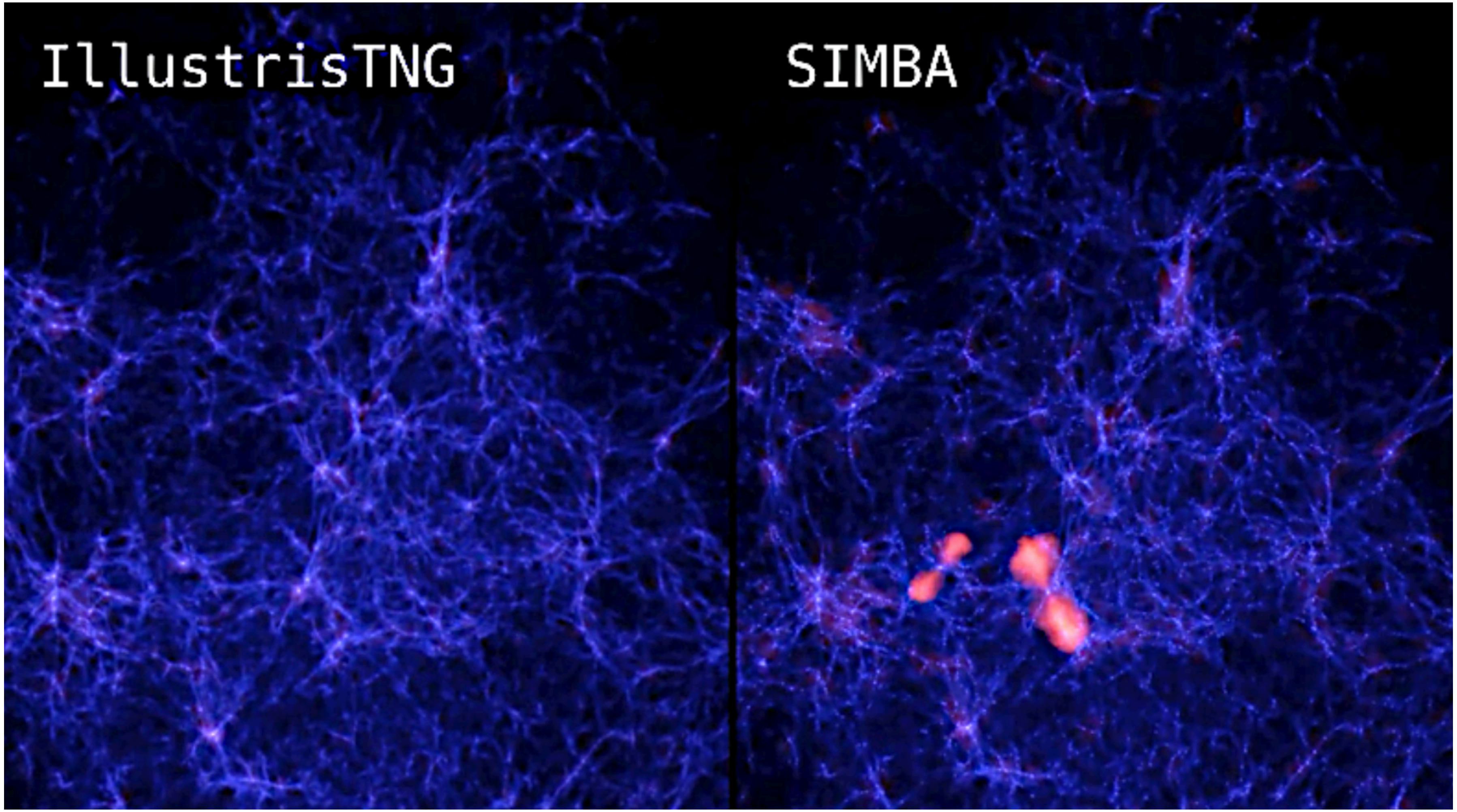
Galaxy cluster mass inference - Ho et al. 2402.05137



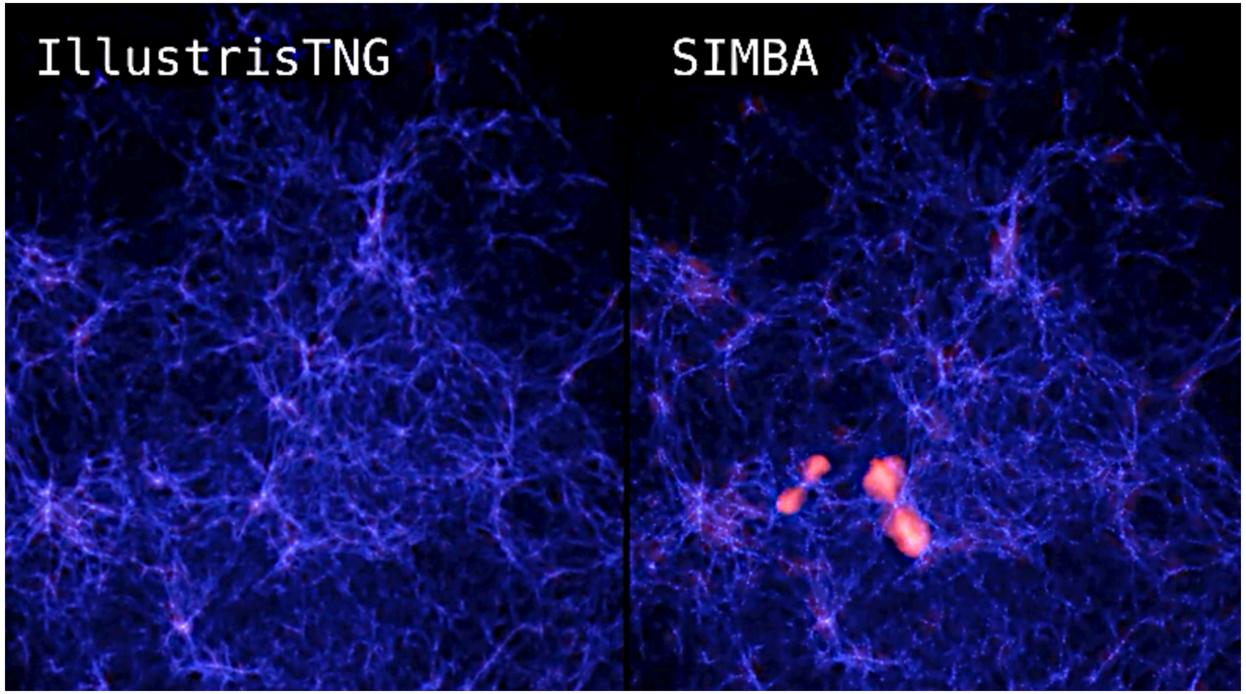
Dust model inference using simulations (Lovell et al. in prep.)



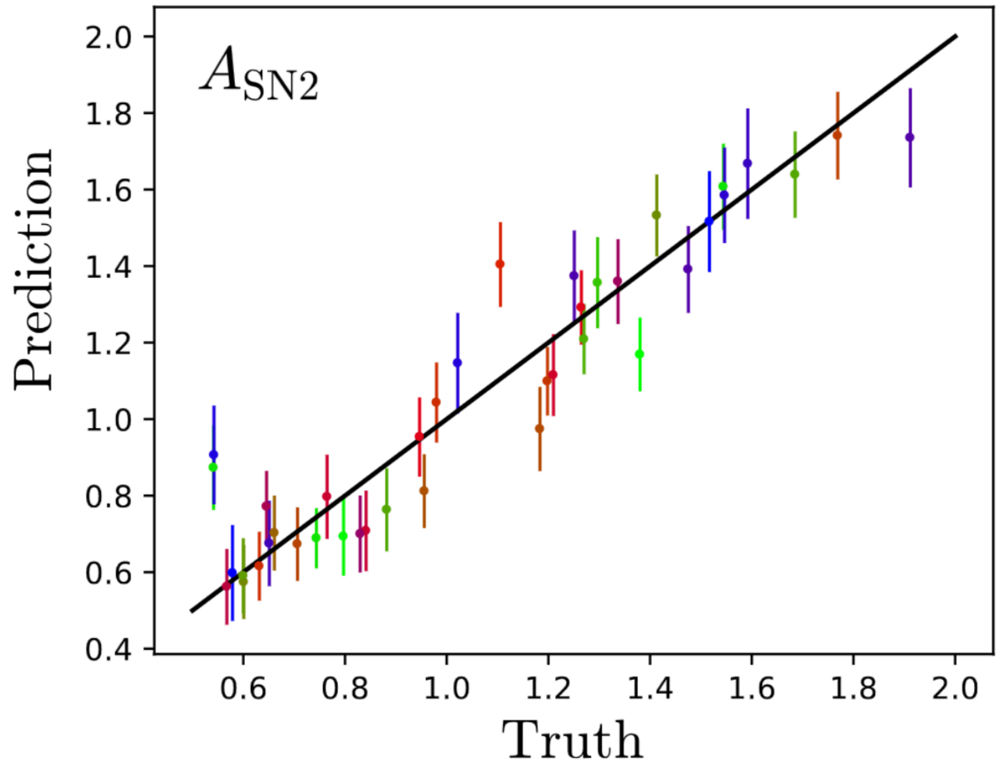
Mass of the local group (Lemos, NJ, et al. 2010.08537)



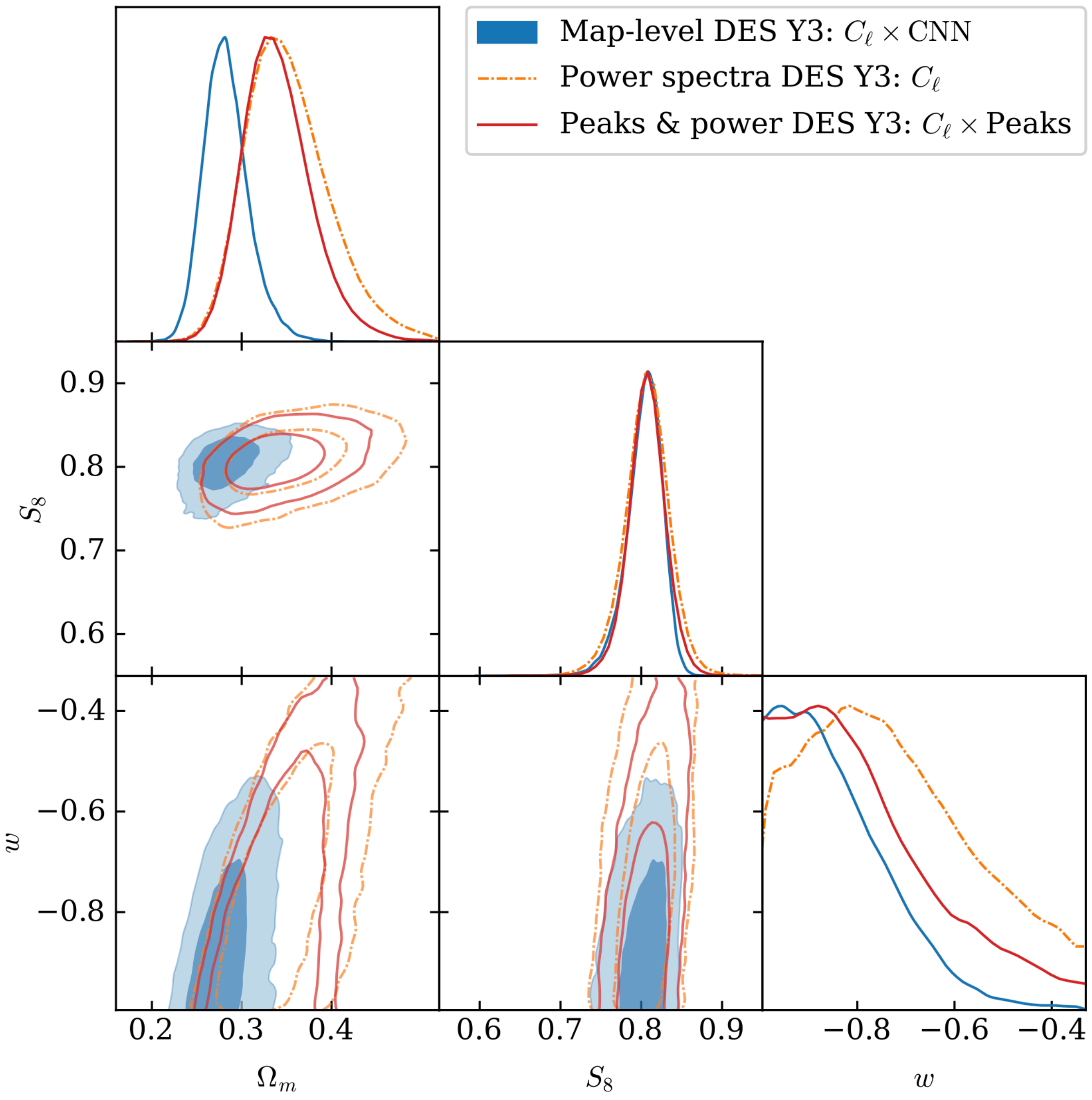
CAMELS: V*illaescusa-Navarro et al. 2109.10915*



CAMELS: V*illaescusa-Navarro et al. 2109.10915*



Using Moment Networks (NJ & Wandelt 2011.05991)



NJ et al. [DES Collab.] 2403.0231

Model 1: galaxies are intrinsically aligned

Simple "Evidence Network" result:

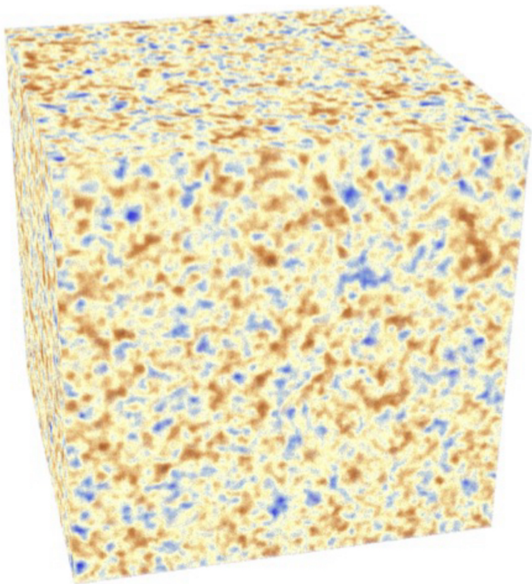
$$\log_{10} K = -0.8 \ (\pm 0.3)$$

Galaxy alignment model comparison (NJ & Wandelt 2403.0231)

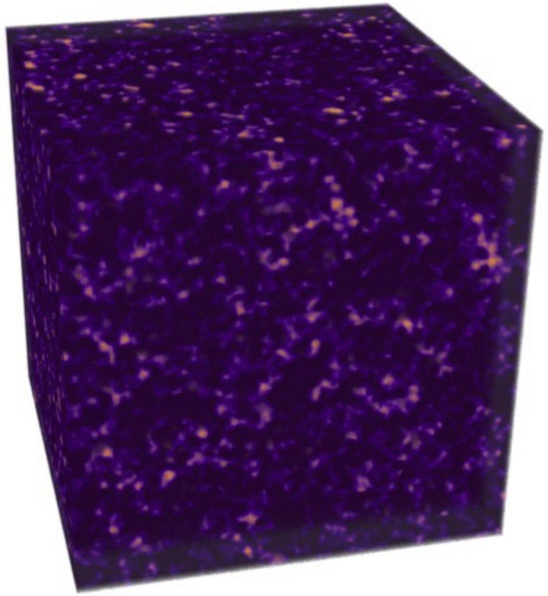
GPT4 - “Can you draw me a scientist looking at a galaxy from London?”

Generative AI: Diffusion models

Initial conditions (density field):

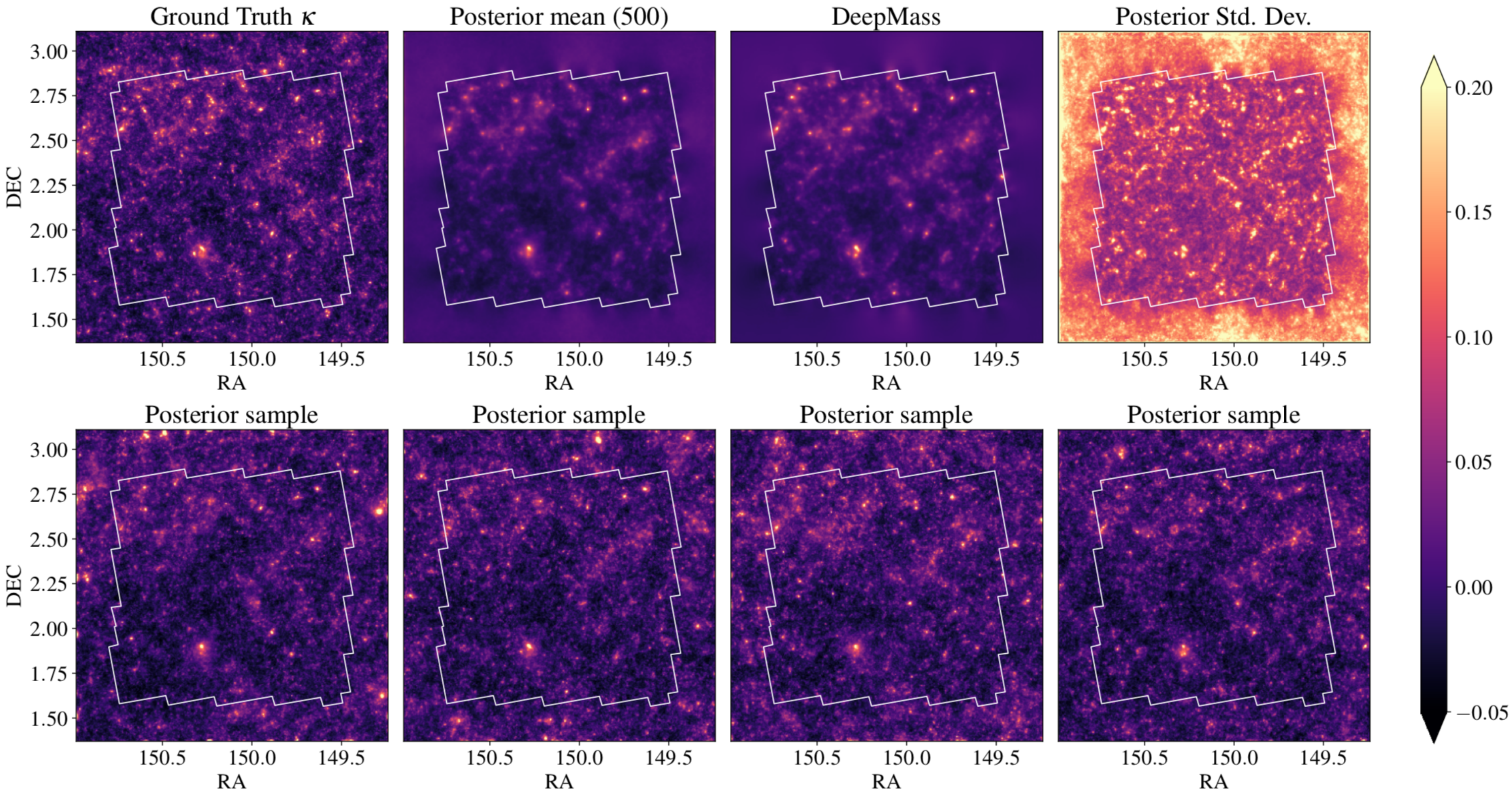


Late Universe, i.e. now (density field):



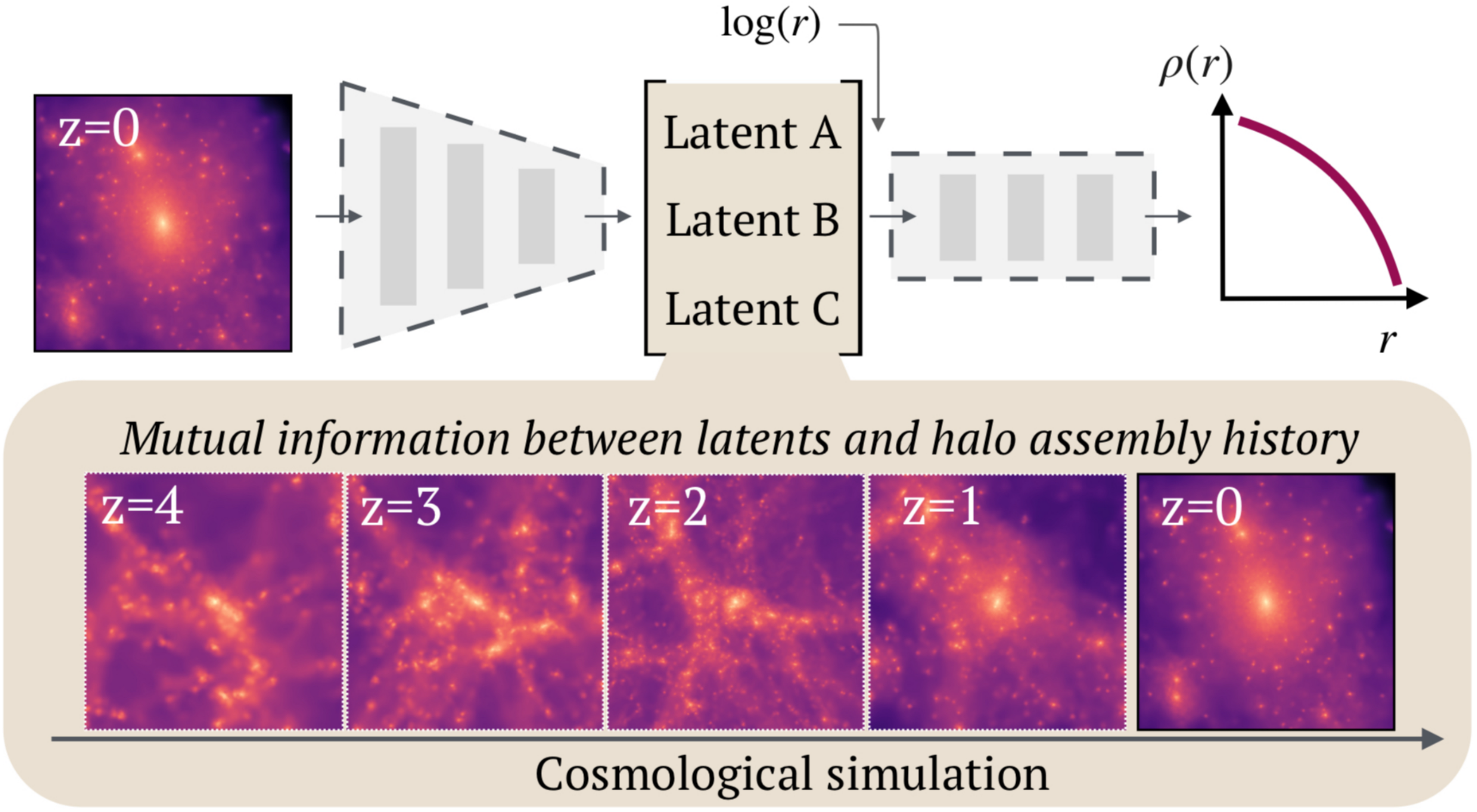
Legin et al. 2304.03788

Generative AI: Diffusion models for inverse problems



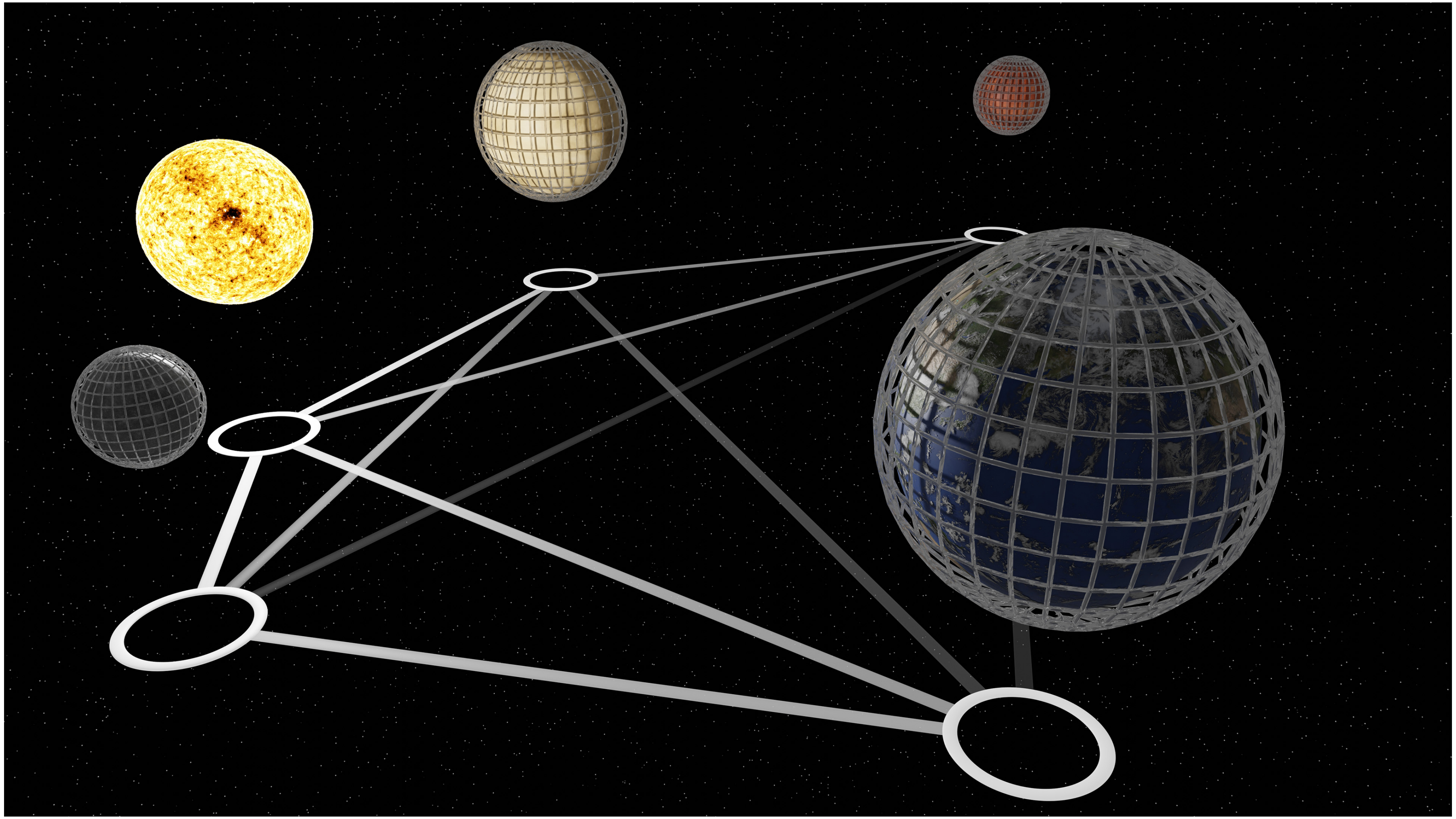
Remy et al. 2201.05561

Interpretation



Lucie-Smith et al. 2305.03077

Interpretation...



“Rediscovering” Newton’s gravity equation from data with graph networks:
Lemos, NJ, Cranmer, Ho, Battaglia 2202.02306

Back-up slides...

$$p(x\,|\,\theta)\ ?$$

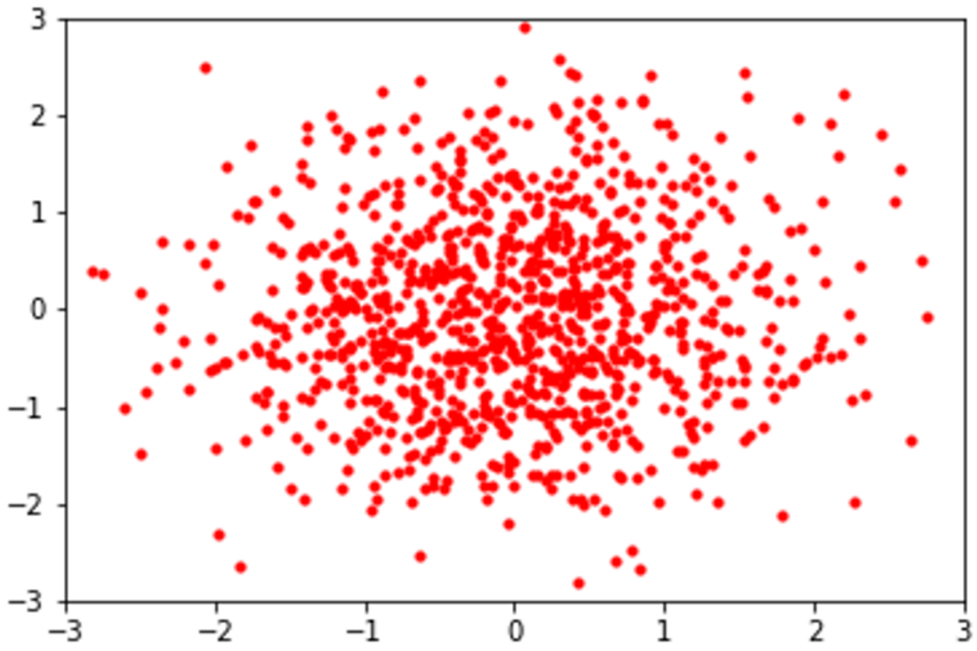
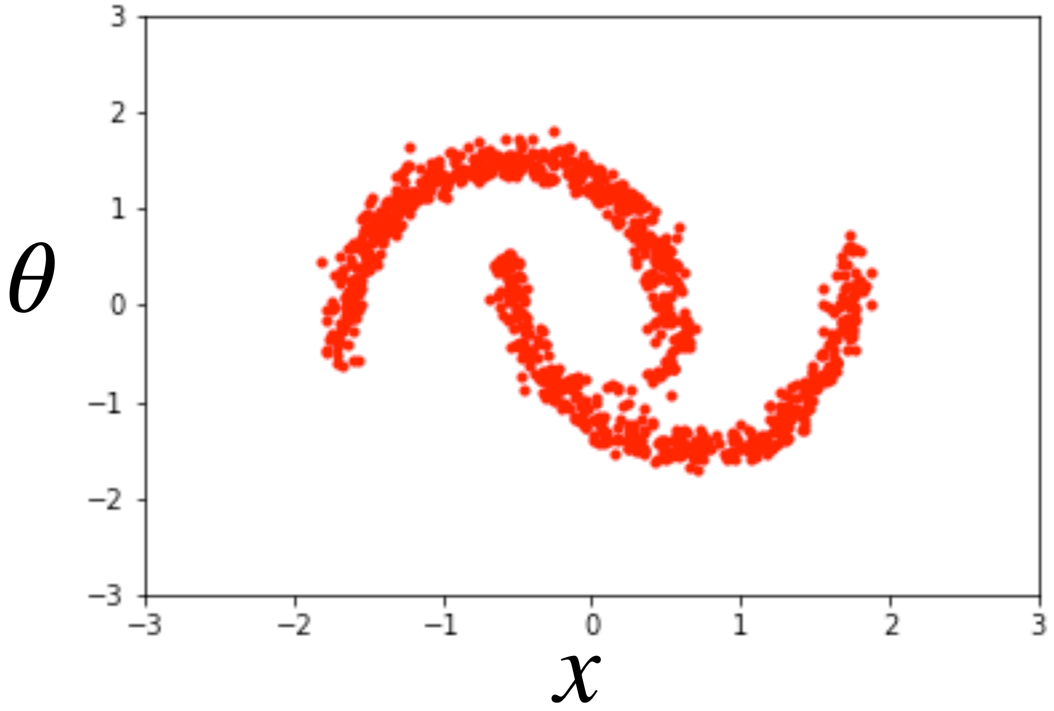


Image credit: Eric Jang

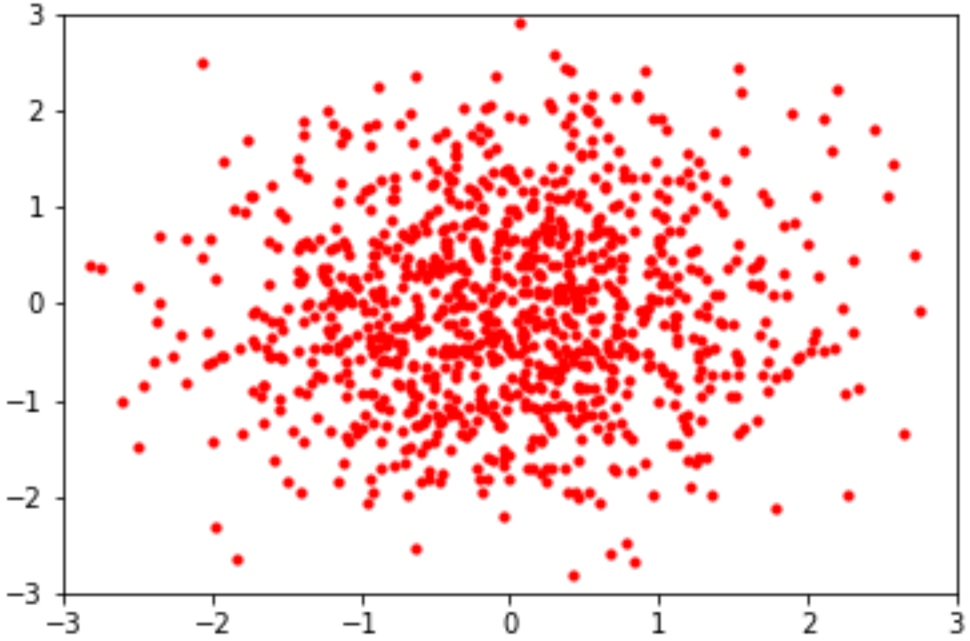


Normalising Flow

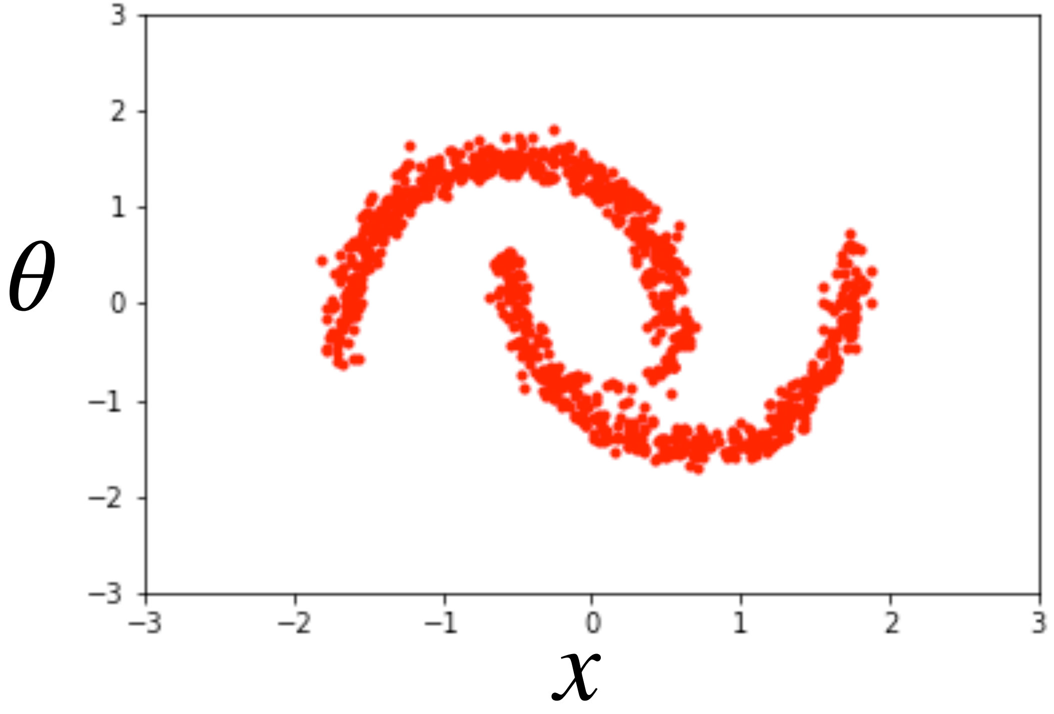
Simulated data

$p(x\,|\,\theta)$?

Image credit: Eric Jang

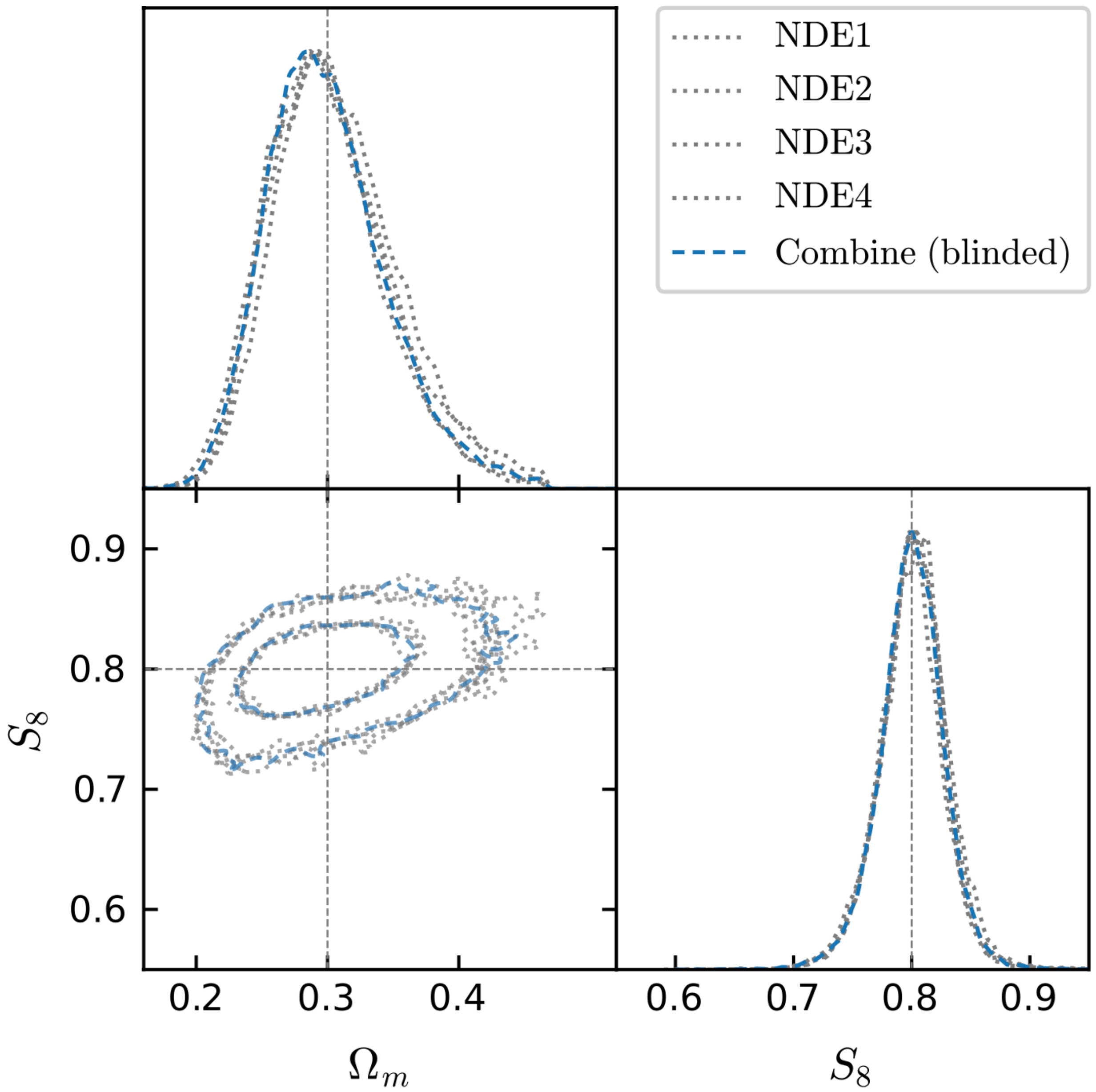


Normalising Flow

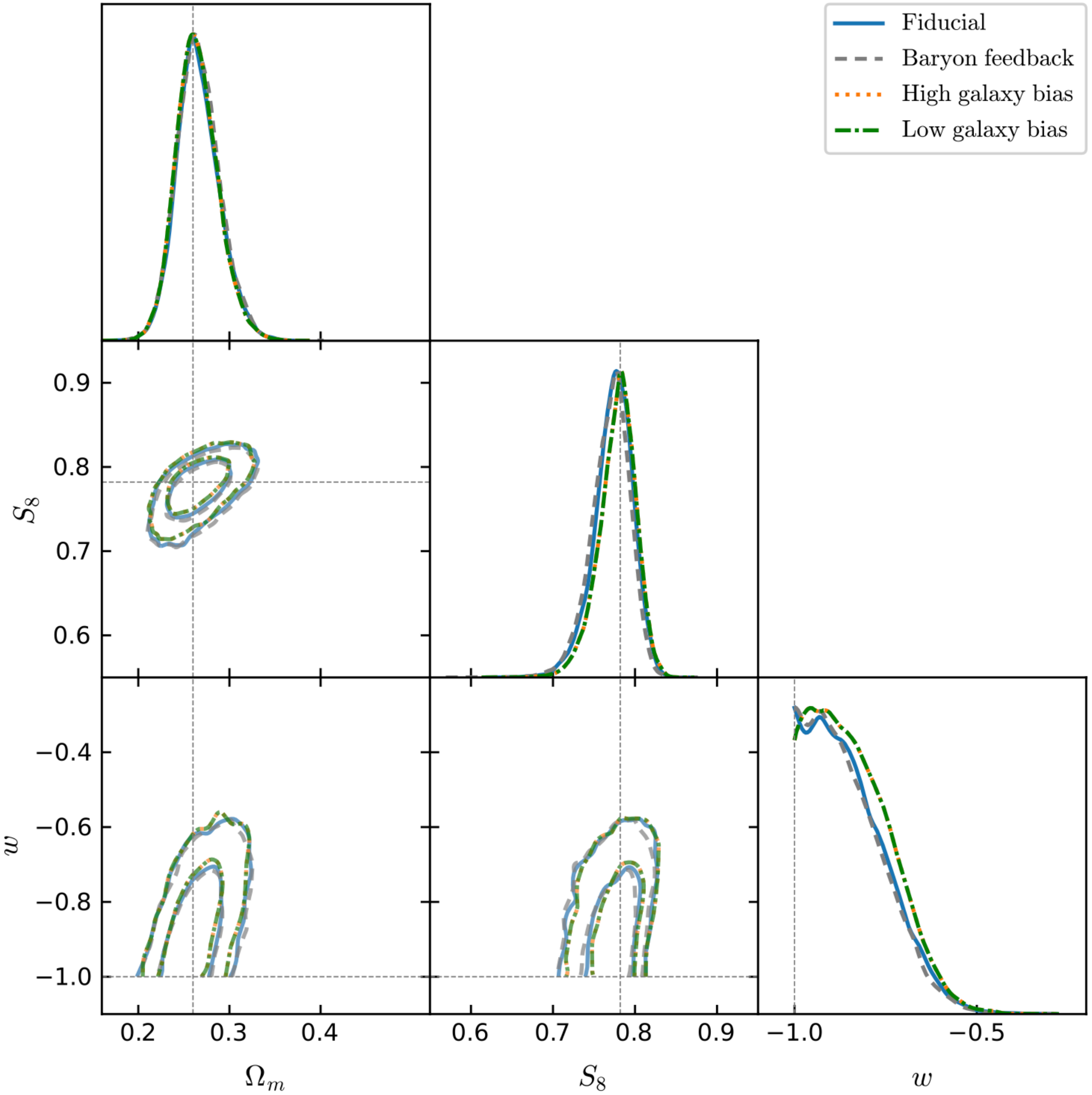


Simulated data

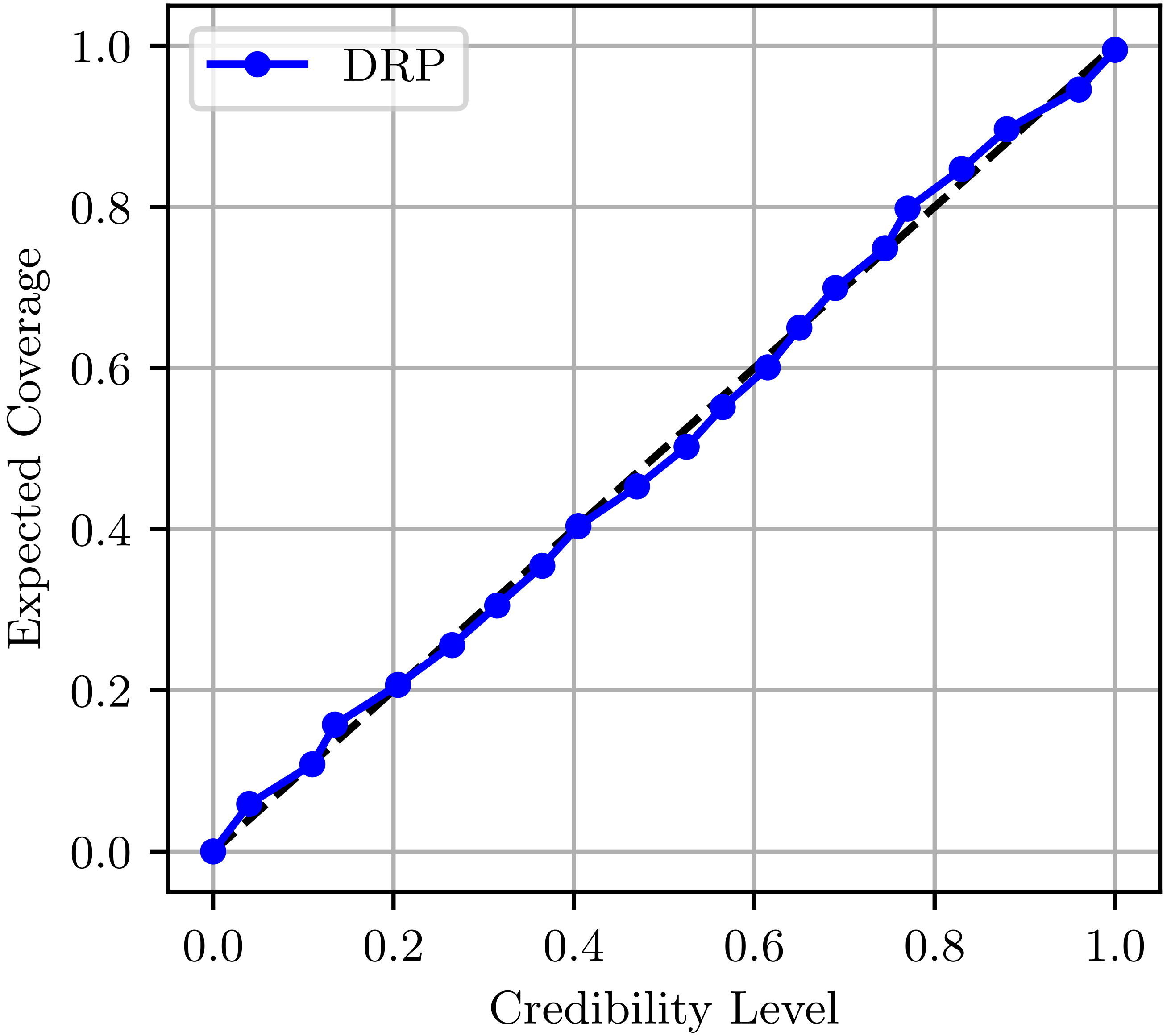
How do I know this is right?



How do I know this is right?



How do I know this is right?



The challenge of model comparison

$$p(M_1 | x) \text{ vs } p(M_0 | x)$$

Bayes factor: $$K = \frac{p(x_O|M_1)}{p(x_O|M_0)}$$

What do people usually do?

Marginal likelihood:

$$p(x_O|M_1) = \int p(x_O|\theta, M_1)\ p(\theta|M_1)\ \mathrm{d}\theta$$

Evidence Networks

1. Generate/collect data for each model: $x_i \sim p(x \mid M_1)$

2. Bespoke loss function: $\mathscr{V}(f(x), m)$

3. Train networks to estimate Bayes factor: $f^*(x_O) = \log K$

What does training a neural network do?

What does training a neural network do?

$$I[f] = \sum_{m\in\{0,1\}} \int \mathcal{V}(f(x), m)\; p(x, m)\; \mathrm{d}x$$

neural network

UCL

What does training a neural network do?

$$I[f] = \sum_{m\in\{0,1\}} \int \mathcal{V}(f(x), m)\; p(x, m)\; \mathrm{d}x$$

model label

loss

training data distribution

How does this work?

$$I[f] = \sum_{m\in\{0,1\}} \int \mathcal{V}(f(x), m)\; p(x, m)\; \mathrm{d}x$$

$$\mathcal{V}(f(x), m) = e^{(\frac{1}{2} - m) f(x)}$$

Exponential loss

How does this work?

$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right) f(x)}$$

$m \in \{0,1\}$

↑

model label

How does this work?

$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right) f(x)}$$

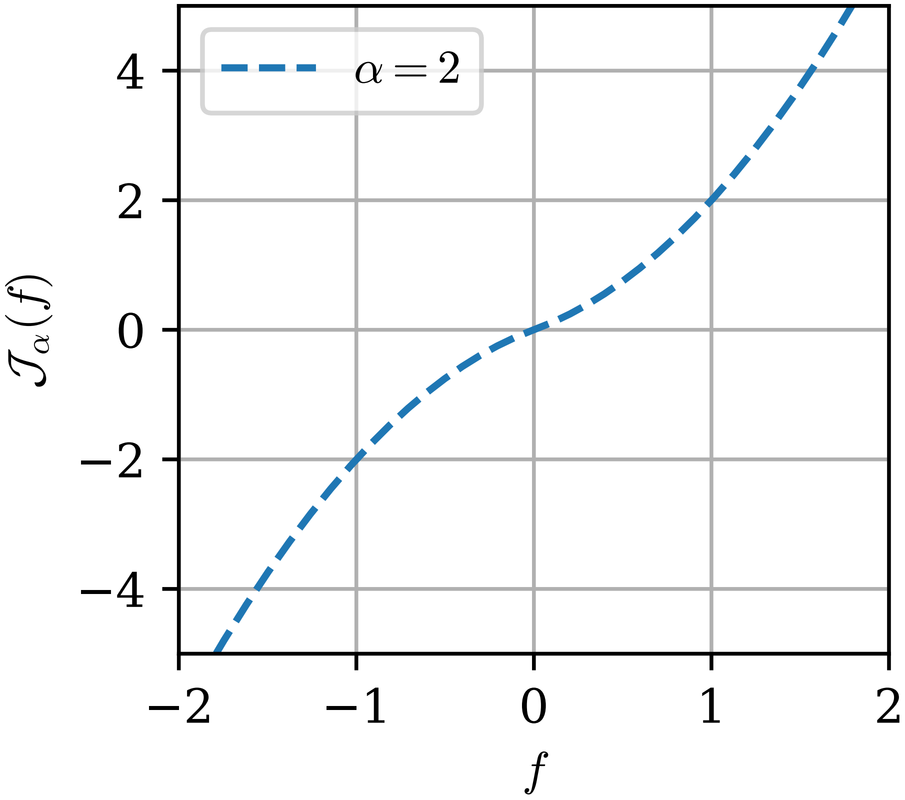
↓

$$f^*(x_O) = \log K$$

How does this work?

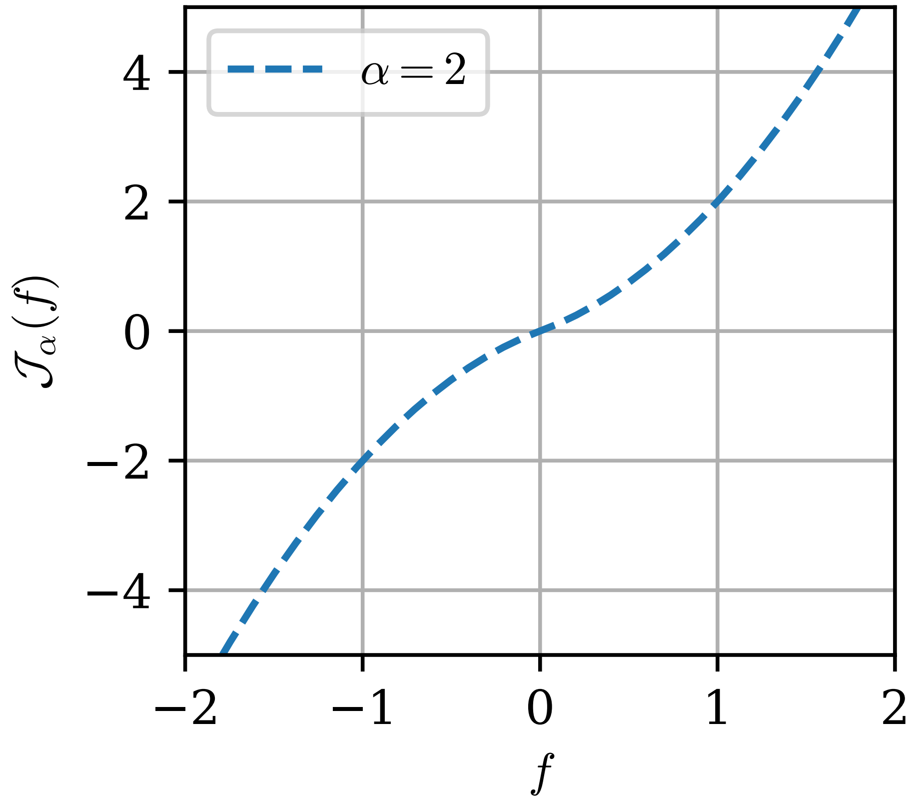
$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right)} \mathcal{J}_\alpha(f(x))$$

leaky parity odd power
(l-POP) transform



$$\mathcal{V}(f(x), m) = e^{\left(\frac{1}{2} - m\right)\mathcal{J}_\alpha(f(x))}$$

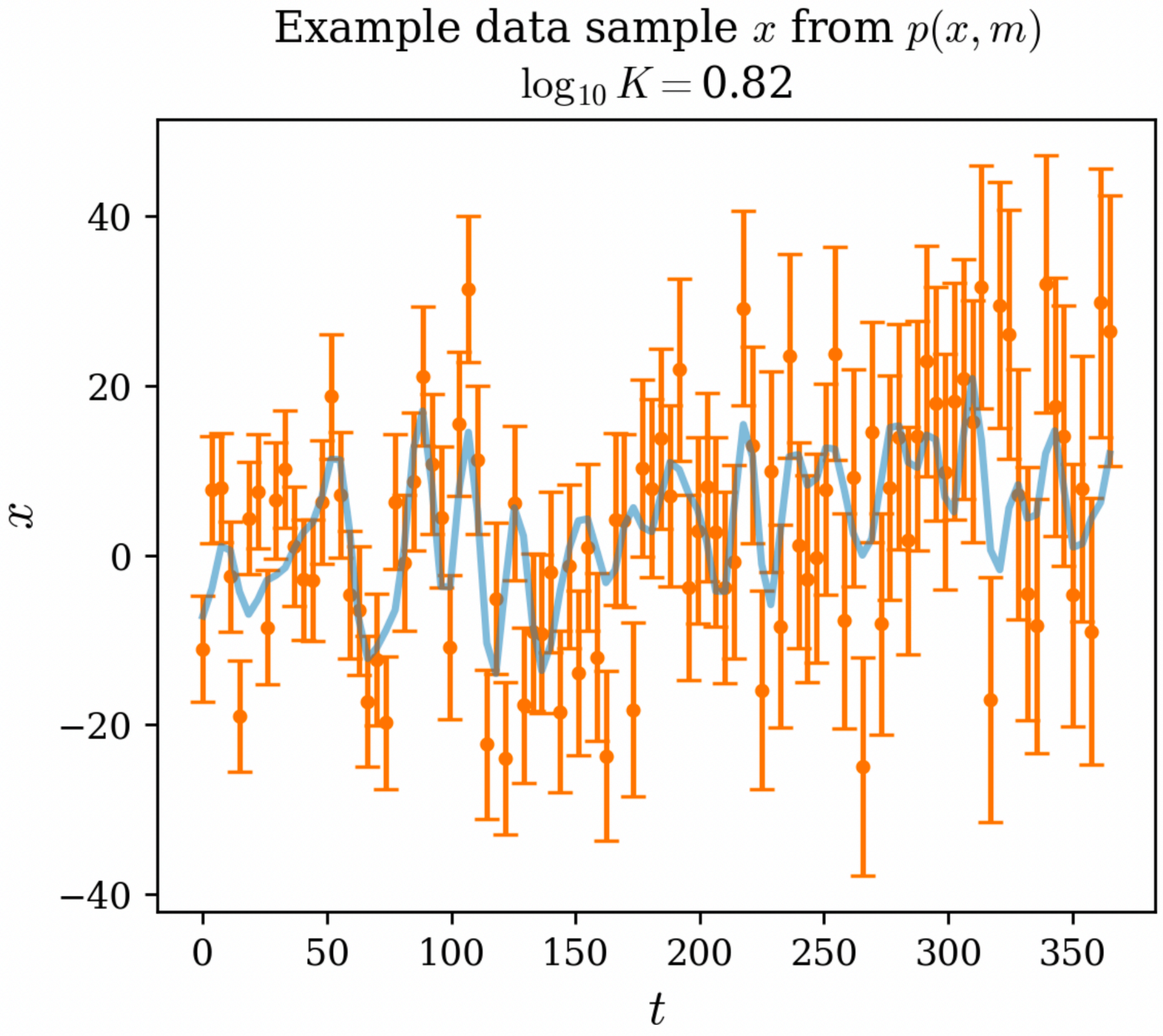
leaky parity odd power
(l-POP) transform



$$\tilde{\mathcal{J}}_\alpha(x) = \text{sgn}(x)\ |x|^\alpha$$

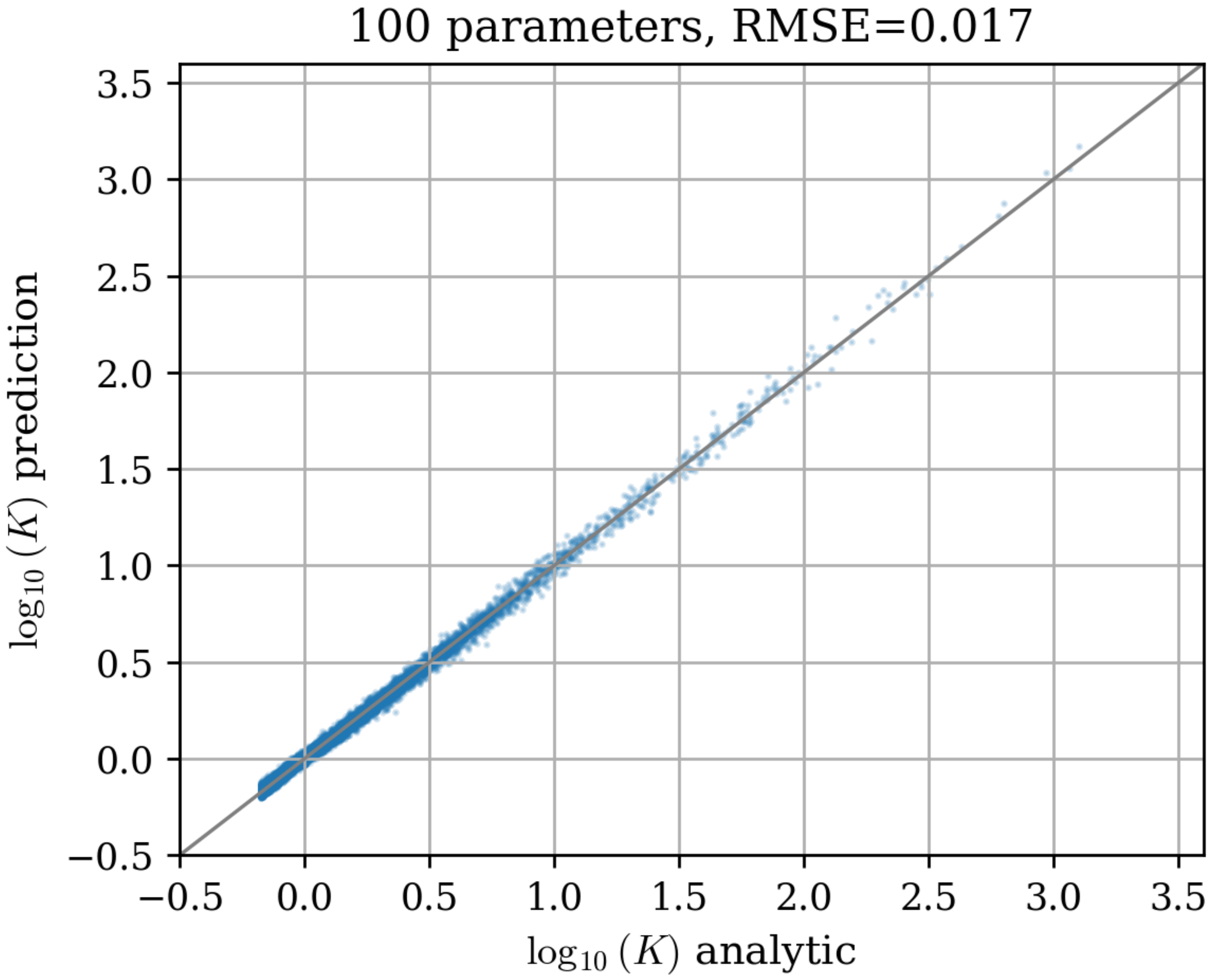
$$\tilde{\mathcal{J}}_\alpha(x) \coloneqq x\ |x|^{\alpha-1}\ \ \forall\ x \in \mathbb{R} \text{ where } \alpha \in \mathbb{R} \text{ and } \alpha \geq 1$$

Demonstration: does this work with data?

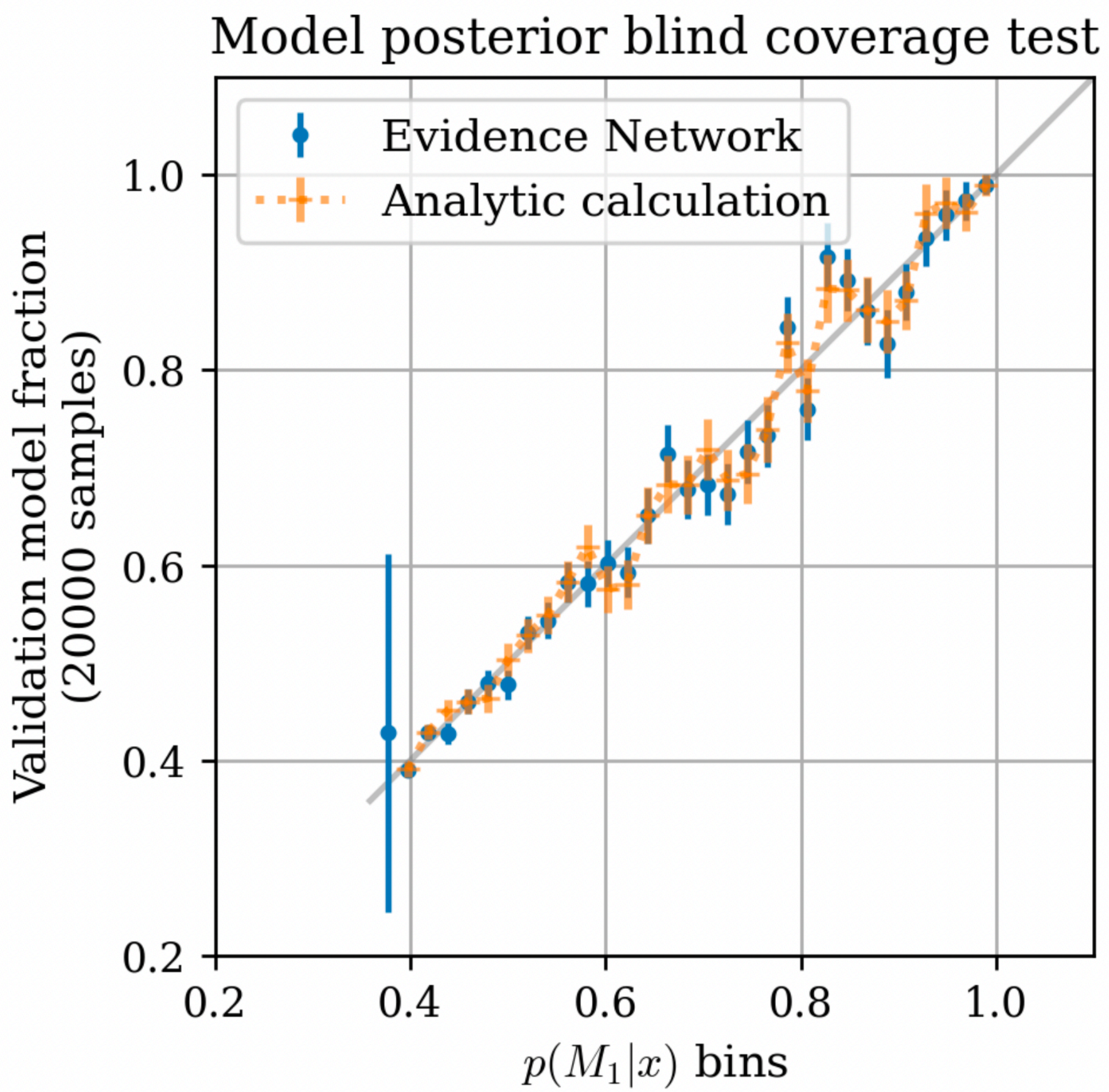


Model 1: Linear growth term

Model 2: No linear growth term


100 parameters, RMSE=0.017
$\log_{10}(K)$ prediction
$\log_{10}(K)$ analytic
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5

How do we know it worked?


Model posterior blind coverage test
Evidence Network
Analytic calculation
Validation model fraction
(20000 samples)
0.2
0.4
0.6
0.8
1.0
$p(M_1|x)$ bins

Considering alternative methods...

"Likelihood-free"?

$$p(x_O|M_1) = \int p(x_O|\theta, M_1)\; p(\theta|M_1)\; \mathrm{d}\theta$$

Considering alternative methods...

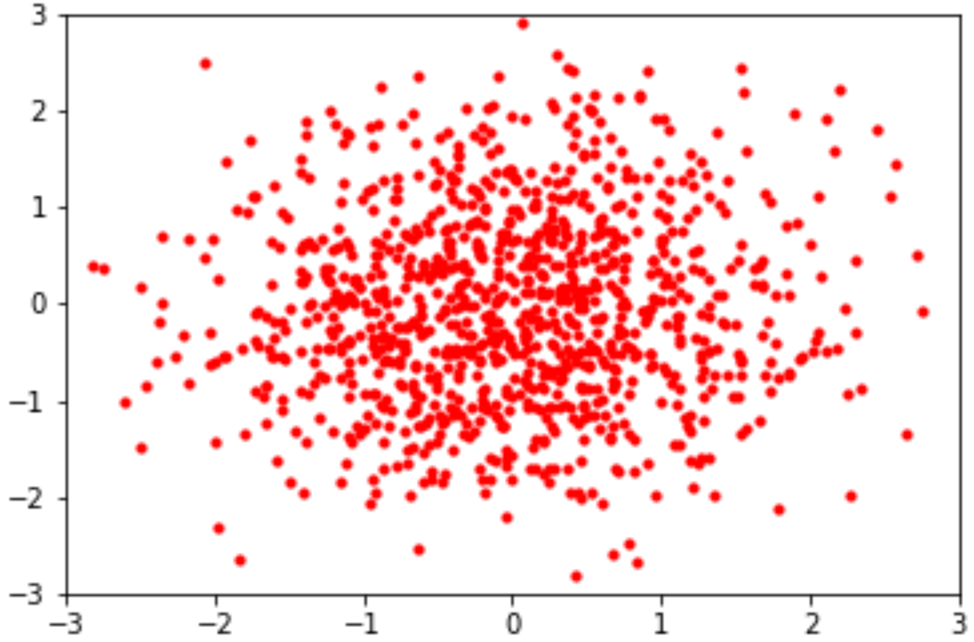
"Likelihood-free"?

$$p(x_O|M_1)$$

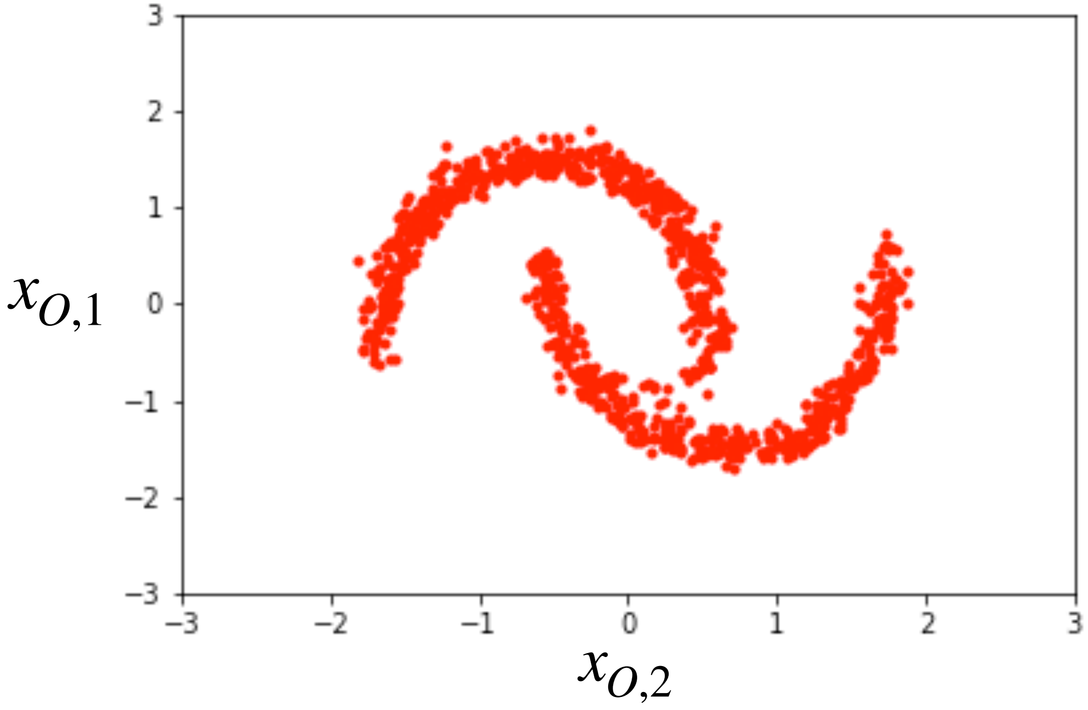
Considering alternative methods…

"Likelihood-free"?

$$p(x_O|M_1)$$

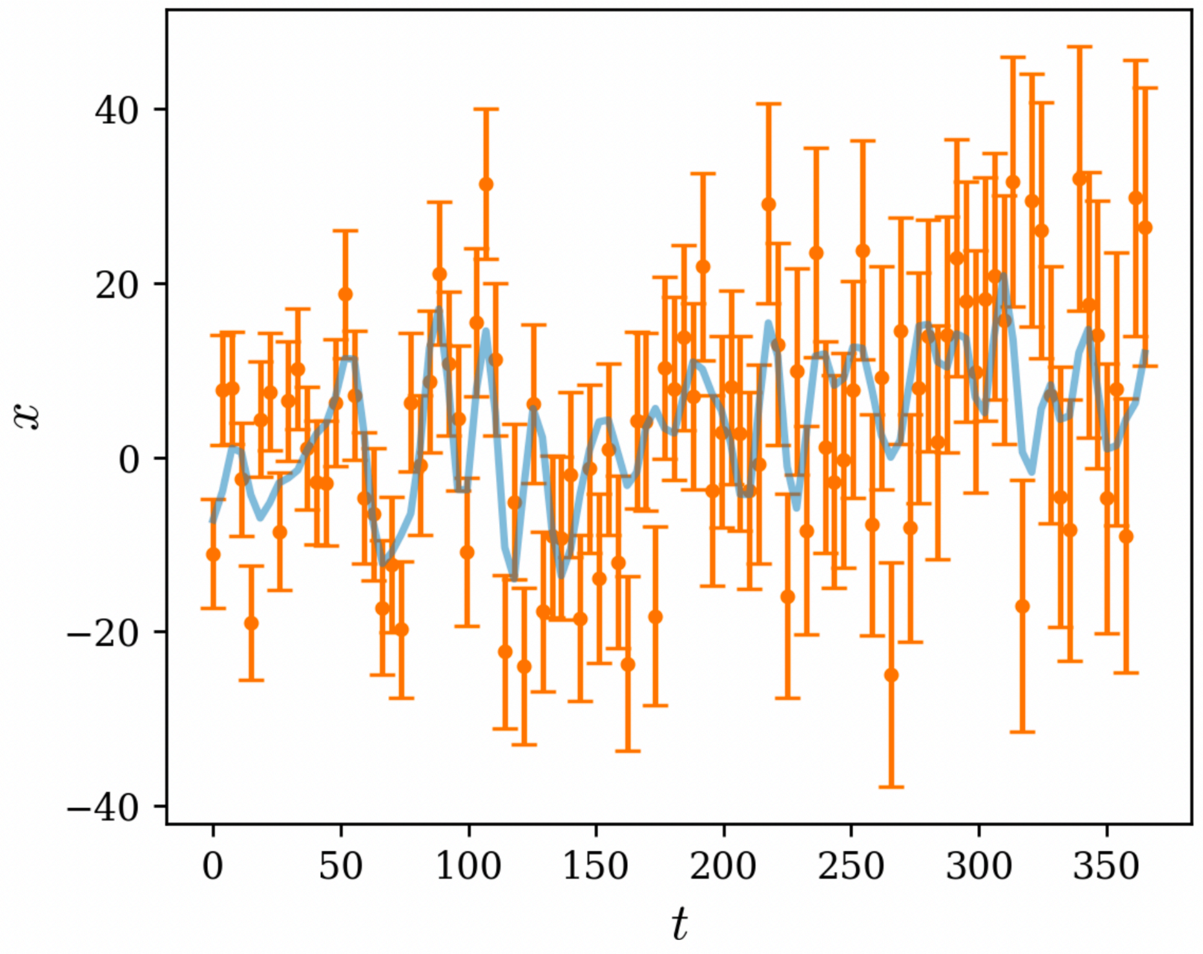


Normalising Flow



Simulated data

Neural density estimation: $p(M_1 | x)$ & $p(M_0 | x)$



PolyChord (assumed likelihood)