TOBIAS BUCK

27.9.2024 TOBIAS.BUCK@IWR.UNI-HEIDELBERG.DE

GENERATIVE AI II

THE NOBEL PRIZE IN PHYSICS 2024

Ill. Niklas Elmehed © Nobel Prize Outreach

John J. Hopfield

Prize share: 1/2

Ill. Niklas Elmehed © Nobel Prize Outreach

Geoffrey E. Hinton

Prize share: 1/2

"for foundational discoveries and inventions that enable machine learning with artificial neural networks"

OUTLINE

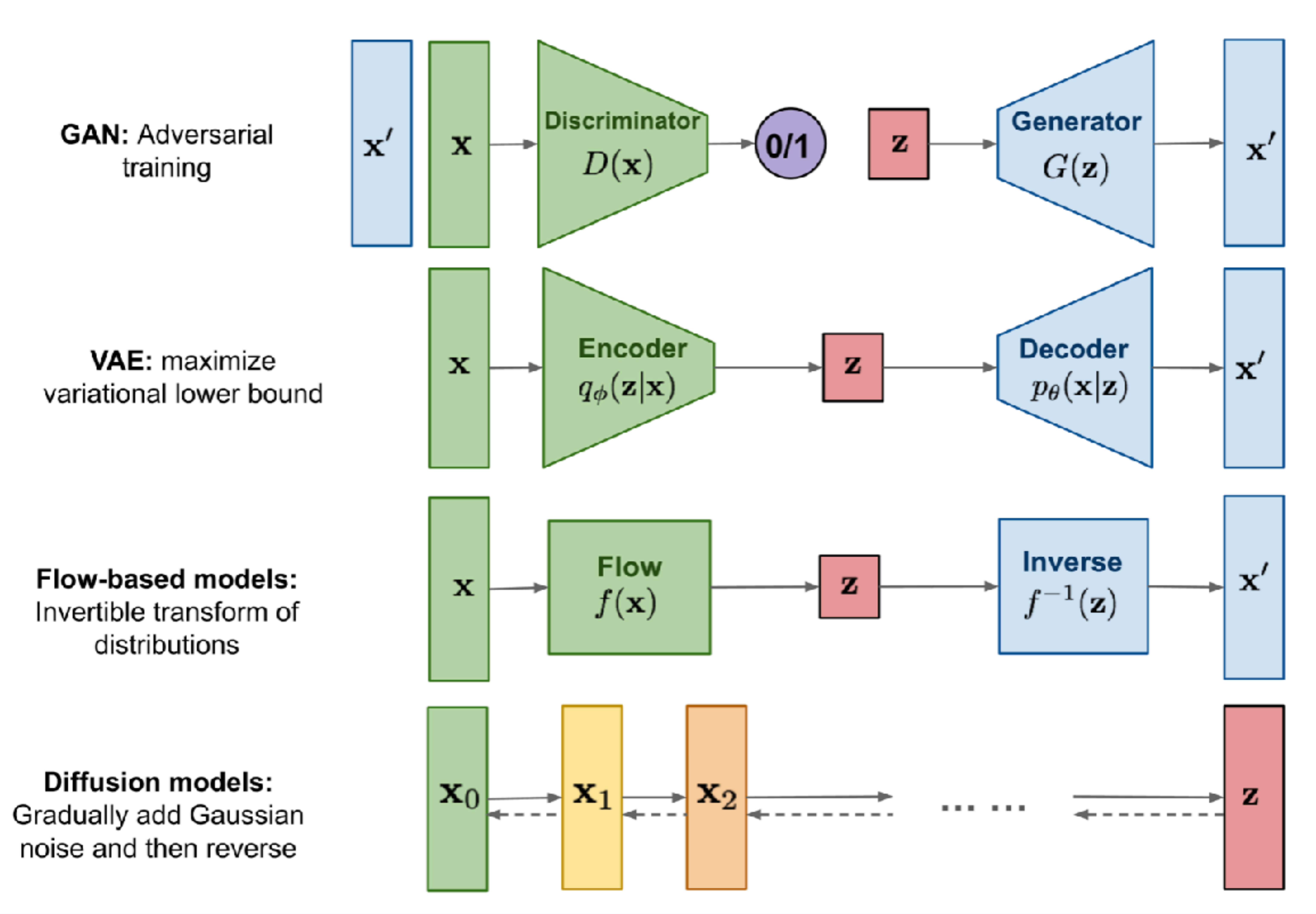
1. Day 3: Generative AI II
   A. Simulation-Based Inference
2. Day 3: Scientific ML
   B. Neural ODEs

WHERE TO FIND THE LECTURE MATERIAL

https://github.com/TobiBu/graddays

GENERATIVE AI FLAVOURS

"Creating noise from data is easy; creating data from noise is generative modeling.“ (Song+2020)



blog post by https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

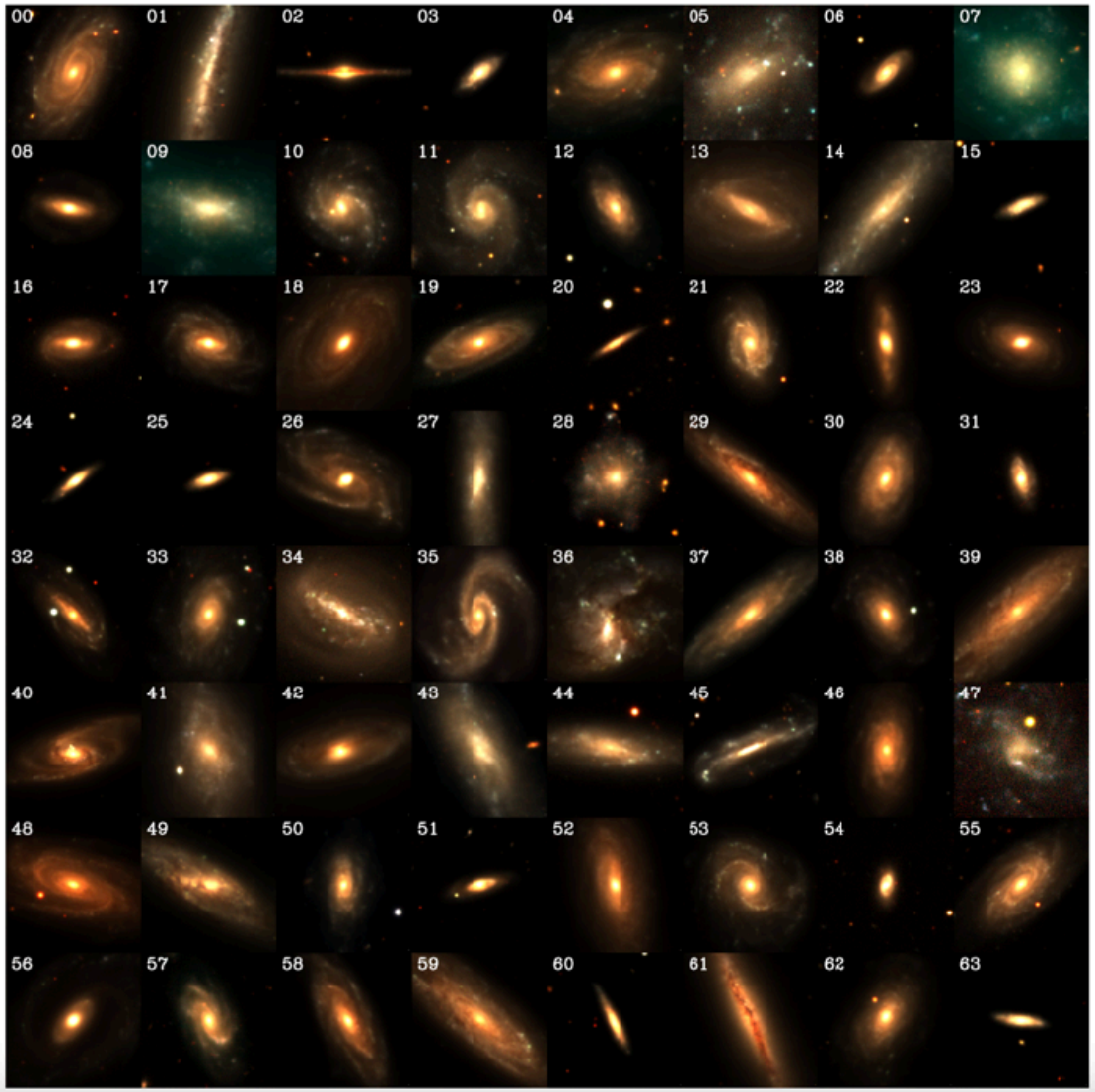
APPLICATION OF SCORE MATCHING AND DIFFUSION MODELS

Learn generative model purely from data!

Smith+2021

But:
In order to train my model, I had to have lots of data.
My model now is just giving more of that data.

How is that useful for science?

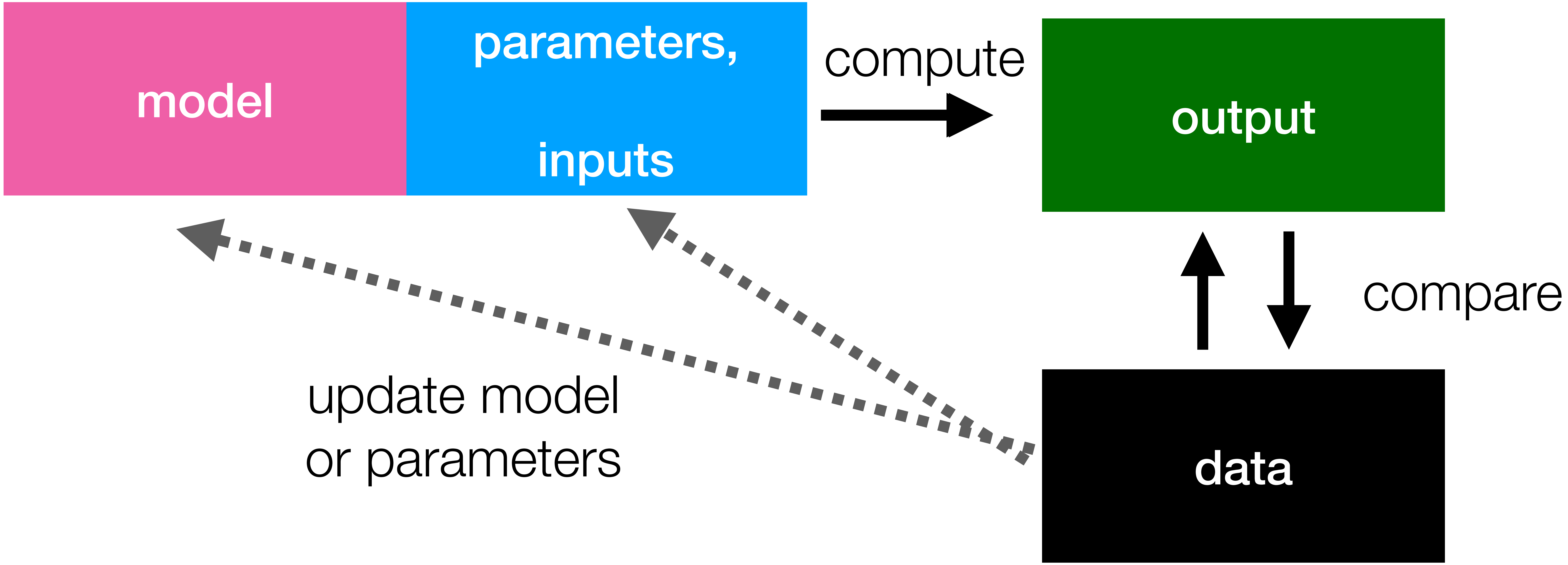


http://www.mjjsmith.com/thisisnotagalaxy/

Classical Inference

TOY SCIENTIFIC WORKFLOW

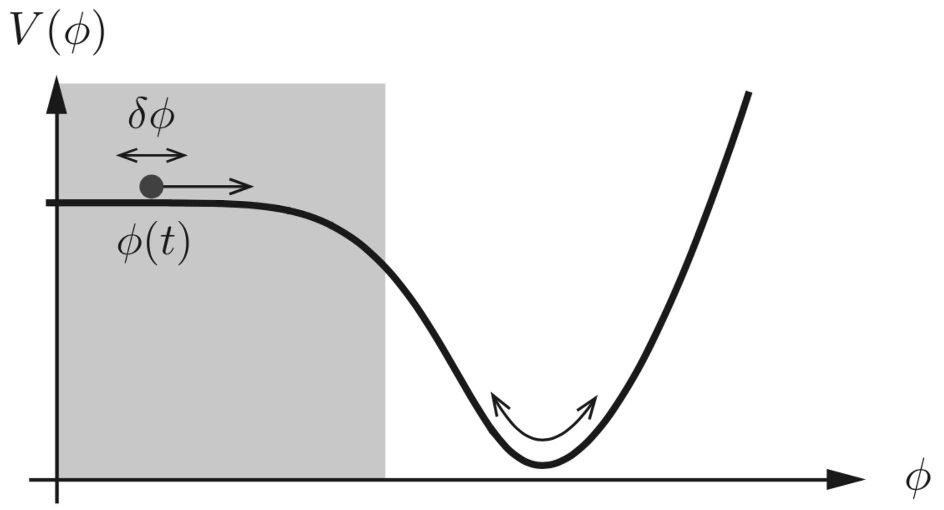
In science, we often perform some subdiagram of:

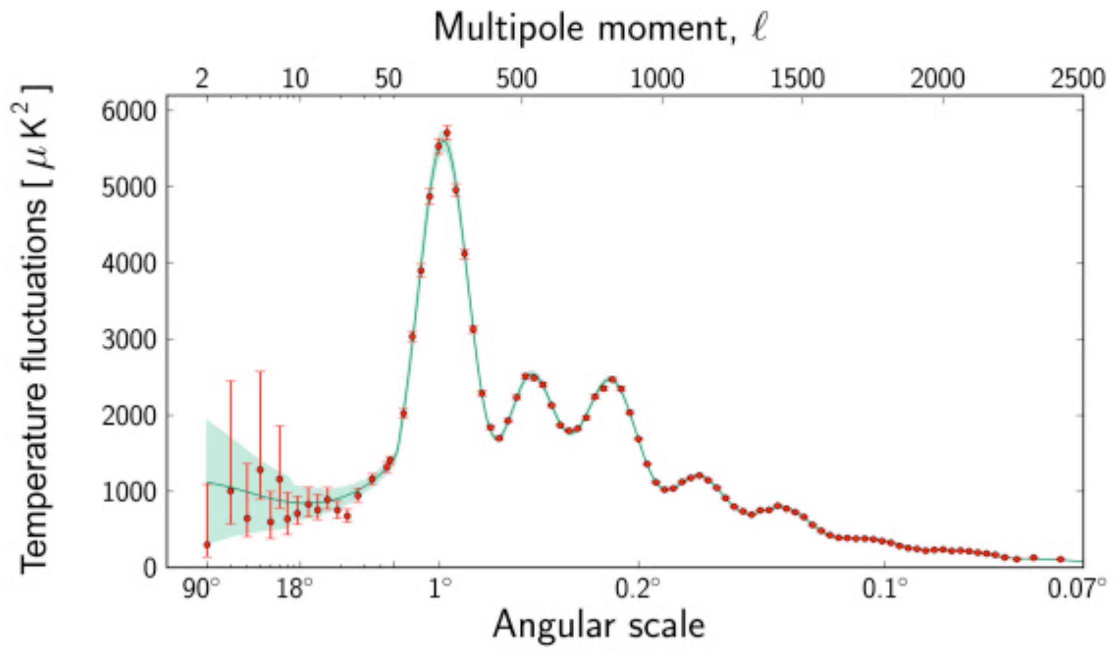


Slides adapted form Alex Cole

TOY SCIENTIFIC WORKFLOW

inflation





model

parameters,

inputs

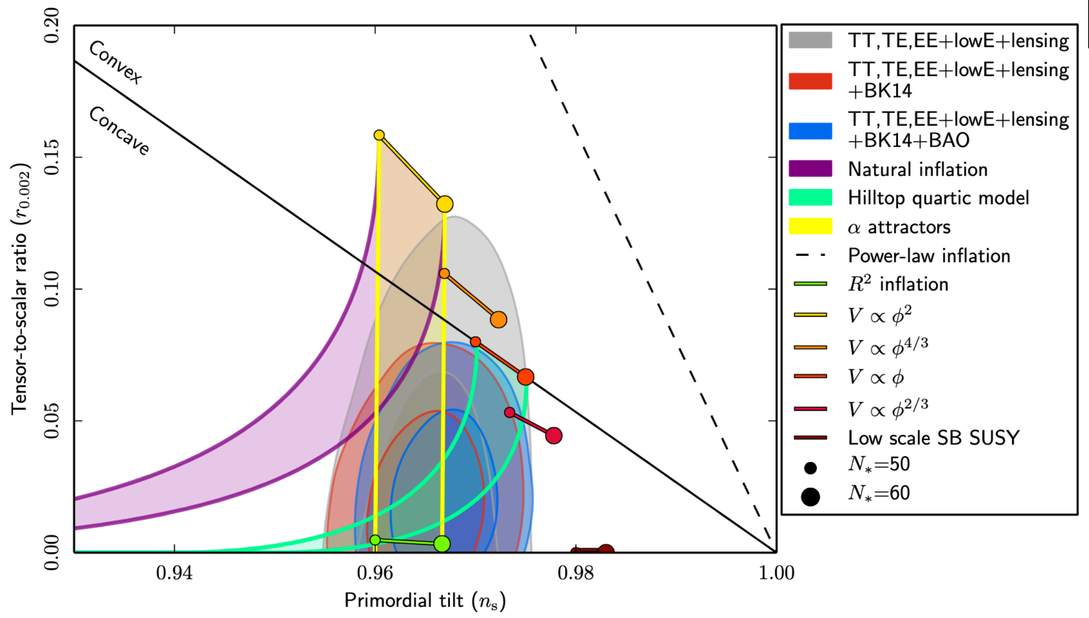
compute

output

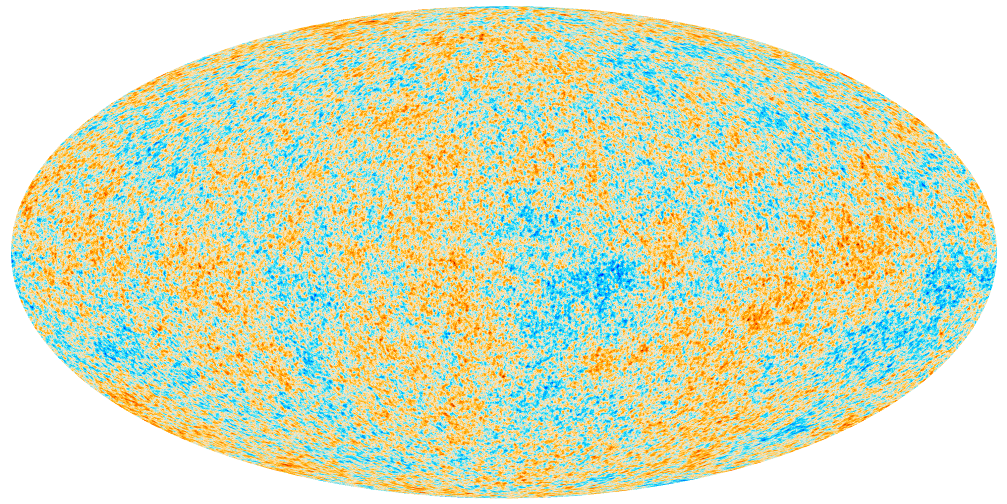
compare

constrain
inflationary models

data

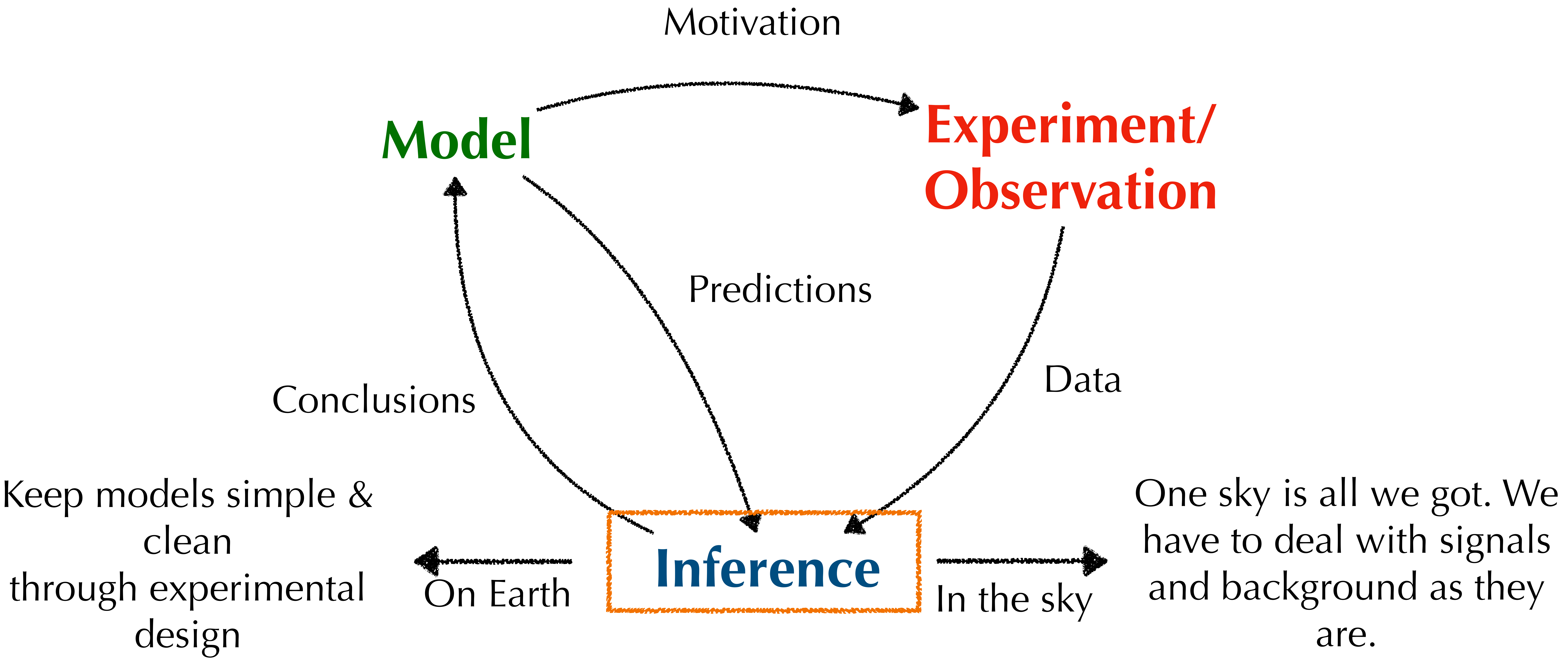


[Planck '18]



Slides adapted form Alex Cole

HOW SCIENTIFIC PROGRESS GOES

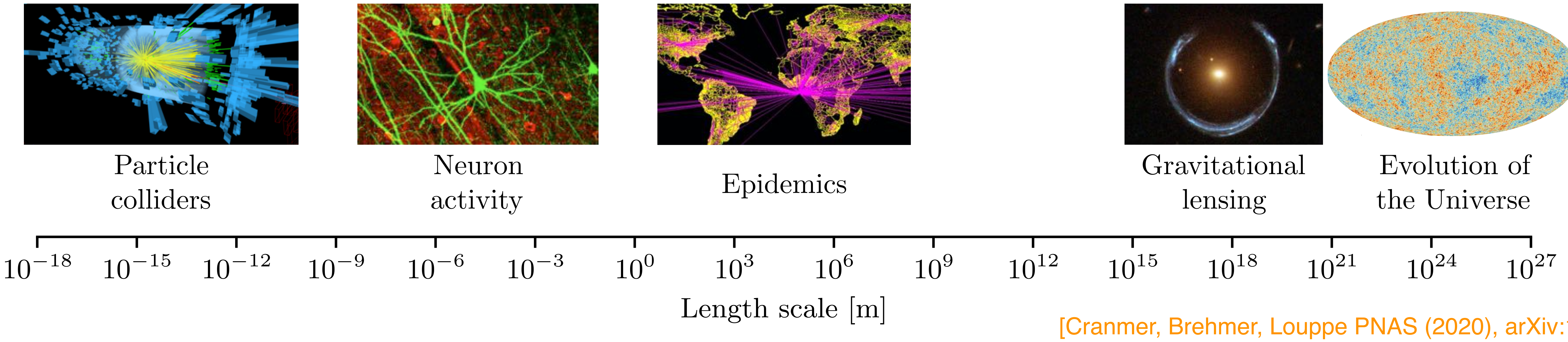


Slide adapted form C. Weniger

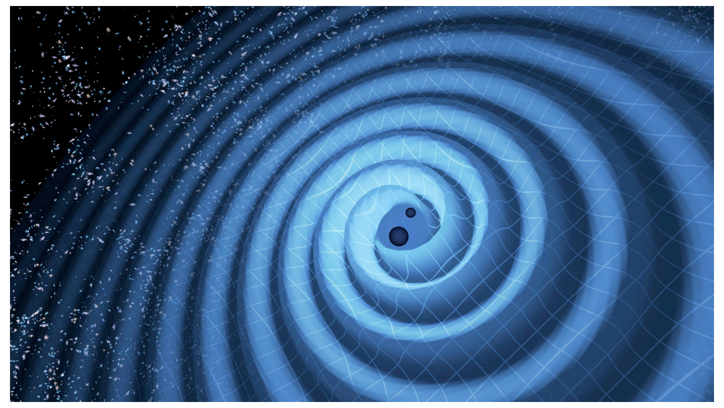
MODELS ARE OFTEN COMPUTATIONAL MODELS

model $\xrightarrow{\text{compute}}$ output

- Advanced computational models allow us to simulate data across length scales:



[Cranmer, Brehmer, Louppe PNAS (2020), arXiv:1911.01429]



gravitational waves

- However, forward models are not well-suited for statistical inference.

Slide adapted form A. Coles

PARAMETER INFERENCE

We go from data to constraints using Bayes' formula

$$p(\theta \,|\, \mathbf{x}) = \frac{p(\mathbf{x} \,|\, \theta)}{p(\mathbf{x})} p(\theta)$$

Classical techniques (MCMC or nested sampling) use **evaluations of the likelihood** to accept/reject proposed steps, resulting in (weighted) samples of the **joint posterior** $p(\theta \,|\, x)$

INTRACTABILITY

- The word **intractable** often shows up when discussing Bayesian inference.

- What is typically meant is there is a *high-dimensional integral* we don't have the resources to perform numerically, e.g.

$$p(x) = \int d\theta p(x \mid \theta) p(\theta)$$ (with θ high-dimensional).

$\boxed{\text{The evidence is typically intractable}}$

$\implies$ MCMC, …

- Note that the likelihood can even be intractable,

$$p(x \mid \theta) = \int d\eta p(x, z \mid \theta)$$ with z latent variables.

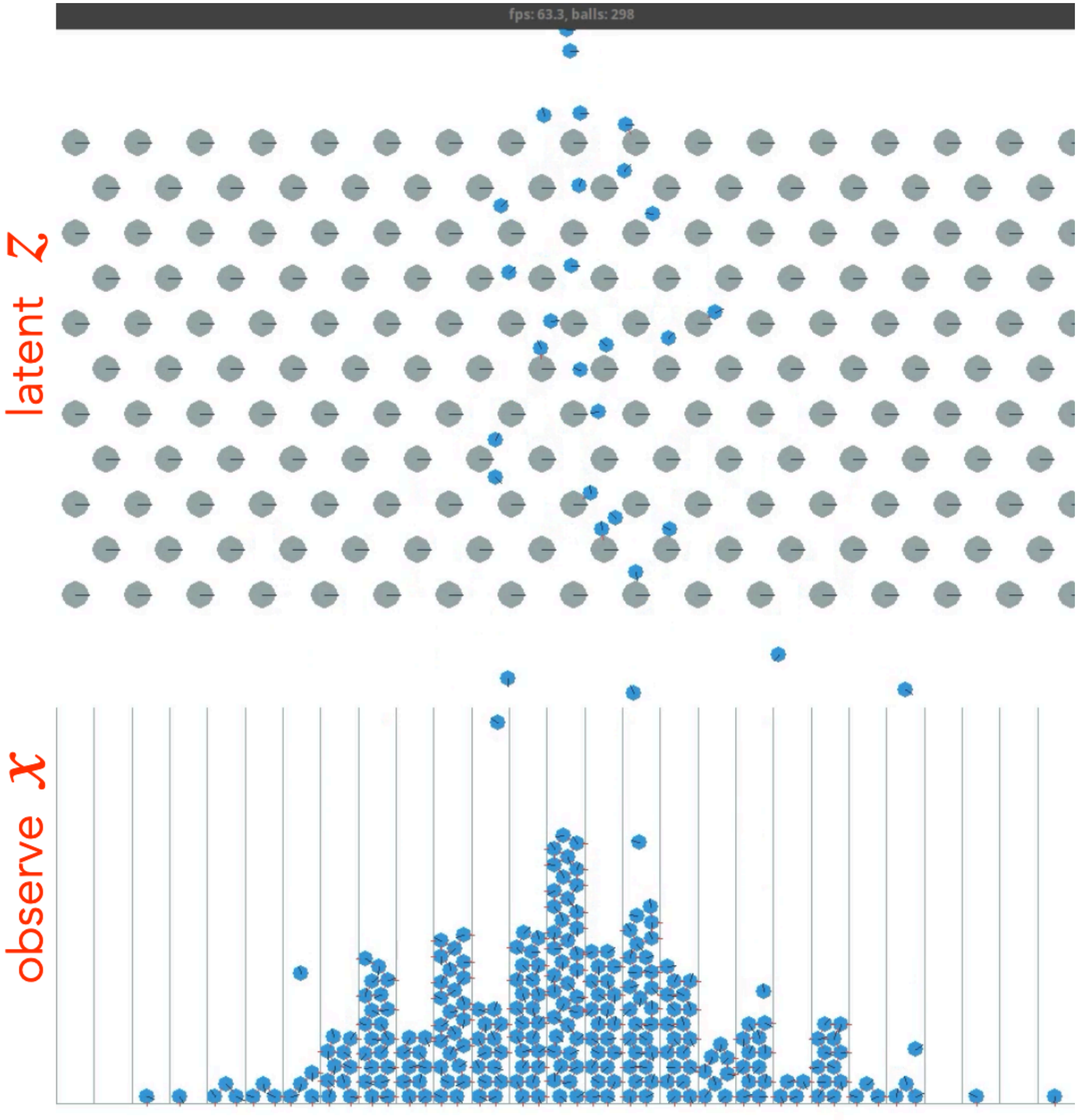
$\boxed{\text{The likelihood can also be intractable}}$

$\implies$???

Slide adapted form A. Coles

SIMULATORS

- **Deterministic evolution of initial state**
  - e.g. differential equations, fluid dynamics, N-body simulations…
- **Stochastic evolution**
  - e.g. Markov processes, molecular dynamics, stochastic differential equations…
- Integral over latent variables is typically intractable $p(x\,|\,\theta) = \int p(x, z\,|\,\theta)dz$



Slide K. Cranmer

LATENT VS. NUISANCE VARIABLE

- Laten variable: unobserved „data“ $p(x, z \mid \theta)$
- Nuisance variable: calibration, etc. $p(x \mid \theta, \eta)$
- Practically, the same consequence — need to integrate/marginalize to get correct answer! This is often intractable.

THE TWO PROBLEMS OF „CLASSICAL“ INFERENCE

PROBLEM I: INTRACTABLE LIKELIHOOD

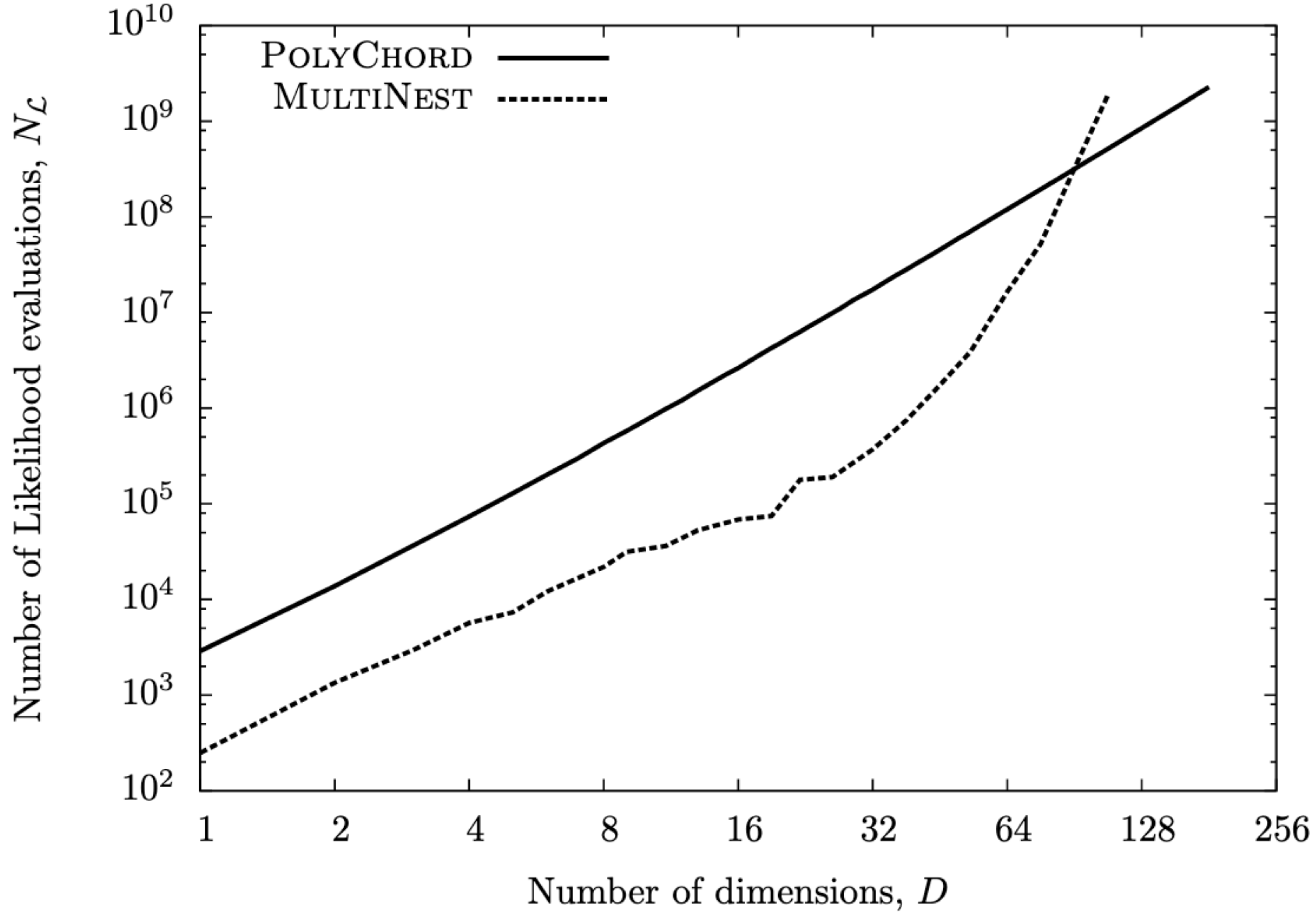
- For most simulators we **cannot evaluate the full likelihood**.
  - In cosmology: large-scale structure, 21cm field, most late-time observations…
- Practitioners often **restrict** to theoretically controlled summary statistics such as the power spectrum large scales
  - Are we using the „right“ / most informative summary statistics???
- Such problems clearly **demand more refined summary statistics**. One option is hand-crafted summaries, e.g. persistent homology for large-scale structure, whose **likelihoods can be approximated**. But we clearly would prefer more knobs to optimize, theoretical guarantees about saturating information content.

PROBLEM II: SCALING

Even for known/tractable likelihood

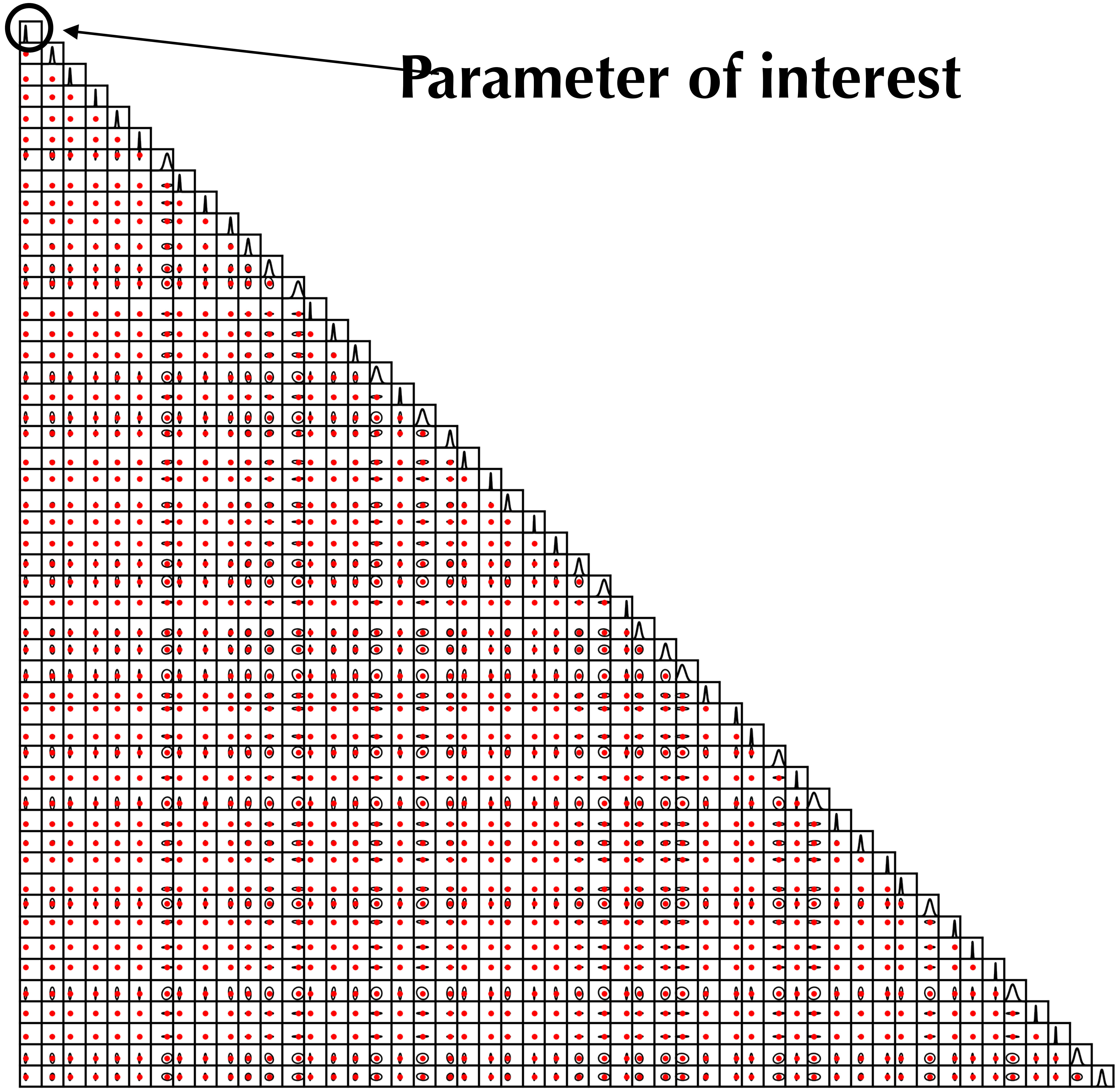
- For realistic inference, one must vary over instrumental calibration parameters, foreground residuals, latent variables…
- **Sampling the joint posterior** scales poorly with parameter space dimension.

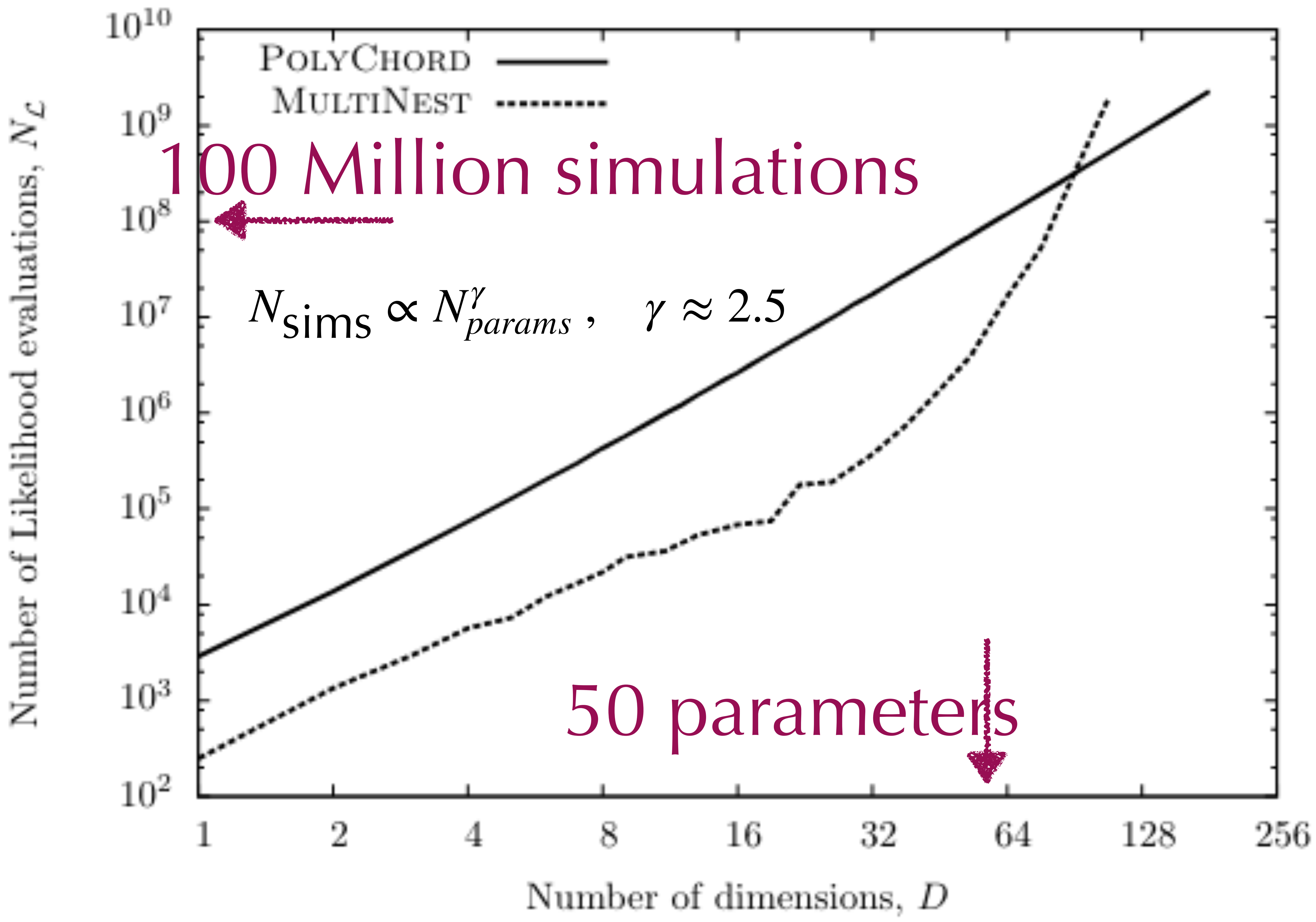
classical inference cost
w/ dimension



[Handley et al. 1506.00171]

THE CURSE OF DIMENSIONALITY





Feroz+ 0809.3437 (MultiNest)
Handley+ 1502.01856 (PolyChord)

Slide C. Weniger

THERE HAS TO BE A BETTER WAY…

THERE HAS TO BE A BETTER WAY…

1. High fidelity physics simulators

3. Deep Learning

5. ???

7.Profit


Simulation-based Inference

SIMULATION-BASED INFERENCE - SBI

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \propto p(\mathbf{x}|\theta)p(\theta)$$

- Insight: running a **stochastic simulator** with input θ gives an output $\mathbf{x}$ that is drawn from an implicit likelihood $p(\mathbf{x}|\theta)$
- „simulation-based inference“ or „likelihood-free inference“ or „implicit likelihood inference“ or … (review: Cranmer+2020)
- recent progress thanks to deep learning algorithms, e.g. conditional normalizing flows (Papamarkios+2019, Greenberg+2019, Hermans+2020, …)

SIMULATION-BASED INFERENCE

Gonçalves+2020

SBI: NEURAL X ESTIMATION

- Use neural networks to approximate some quantities in Bayes' formula

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)}{p(\mathbf{x})} p(\theta) = \frac{p(\mathbf{x} \mid \theta)}{p(\mathbf{x})} p(\theta)$$

- Neural Posterior Estimation (NPE)
- Neural Likelihood Estimation (NLE)
- Neural Ratio Estimation (NRE)

SBI: (CONDITIONAL) DENSITY ESTIMATION

- NLE and NPE both estimate normalised probability densities, hence:
  - restricted network architectures, e.g. normalizing flows or mixture density models. potentially difficult to train (Papamarkios+2021)
  - for high-dimensional data, compression/embedding network needed.
- but: restriction can be a good inductive bias, especially if posterior or likelihood is “perturbation around Gaussian distribution”
- automatic marginalization possible

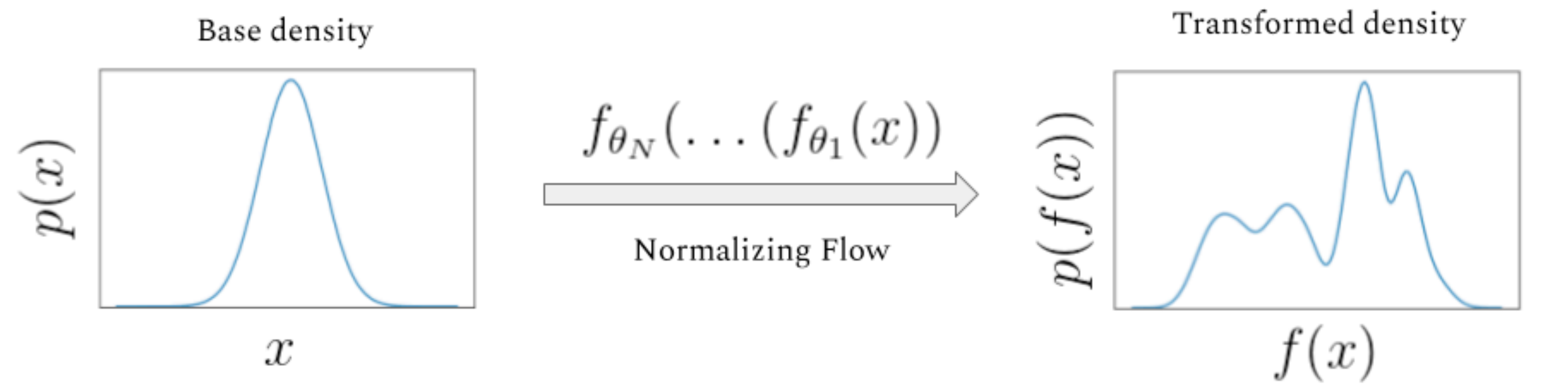


figure: https://gebob19.github.io/normalizing-flows/

c.f. pydelfi Alsing+2018,2019;
moment networks Jeffrey+Wandelt 2020,
SBI Jakob Macke, ItU-ili Ho+2024,
Bayesflow Radev+2020,2023, swyft Miller+2021,2022

https://simulation-based-inference.org/
https://github.com/smsharma/awesome-neural-sbi for references to software and applications

SBI: RATIO ESTIMATION

- Ratio estimation is fundamentally different to NLE and NPE
  - Train a classifier to distinguish data- parameter pairs (x, θ) jointly drawn from $p(x, \theta)$ (label y=1) from marginally drawn $p(x)p(\theta)$ (label y=0)

likelihood-to-evidence ratio

classifier

$$\boxed{r(x, \theta)} \equiv \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{x})p(\boldsymbol{\theta})} = \frac{\tilde{p}(\boldsymbol{x}, \boldsymbol{\theta} \mid y=1)}{\tilde{p}(\boldsymbol{x}, \boldsymbol{\theta} \mid y=0)} = \frac{\boxed{\tilde{p}(y=1 \mid \boldsymbol{x}, \boldsymbol{\theta})}}{1 - \tilde{p}(y=1 \mid \boldsymbol{x}, \boldsymbol{\theta})}.$$

[Hastie et al., 2001; Sugiyama et al., 2012; Cranmer et. al., 2015]

SBI: RATIO ESTIMATION

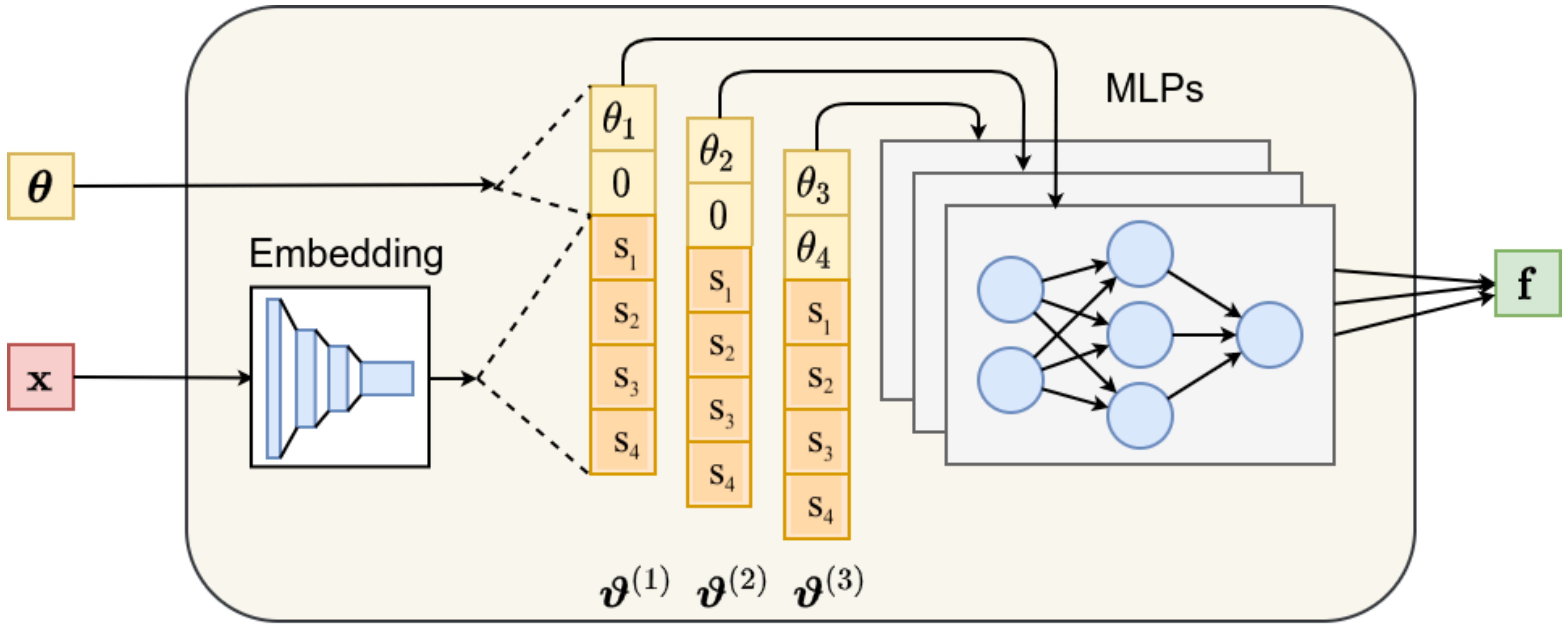
- Intuitive picture: given two probability distributions $q_1(x), q_2(x)$, the best guess for whether x came from q_1 or q_2 is closely related to the probability ratio:

$$\frac{q_1(x)}{q_2(x)}$$

- Now, if we take $q_1 = p(x, \theta),\ \ q_2 = p(x)p(\theta)$ the classifier will learn the likelihood-to-evidence ratio!

SBI: RATIO ESTIMATION IN PRACTISE

- Classifiers are very flexible in network architecture. Training is also simple.
- It is still useful to use a “compression” or “embedding” network, which turns complex data x into features s.



NB: this picture actually shows *marginal* ratio estimation, wait for next slide

MARGINAL RATIO ESTIMATION

- With neural methods, **automatic marginalization** is possible.

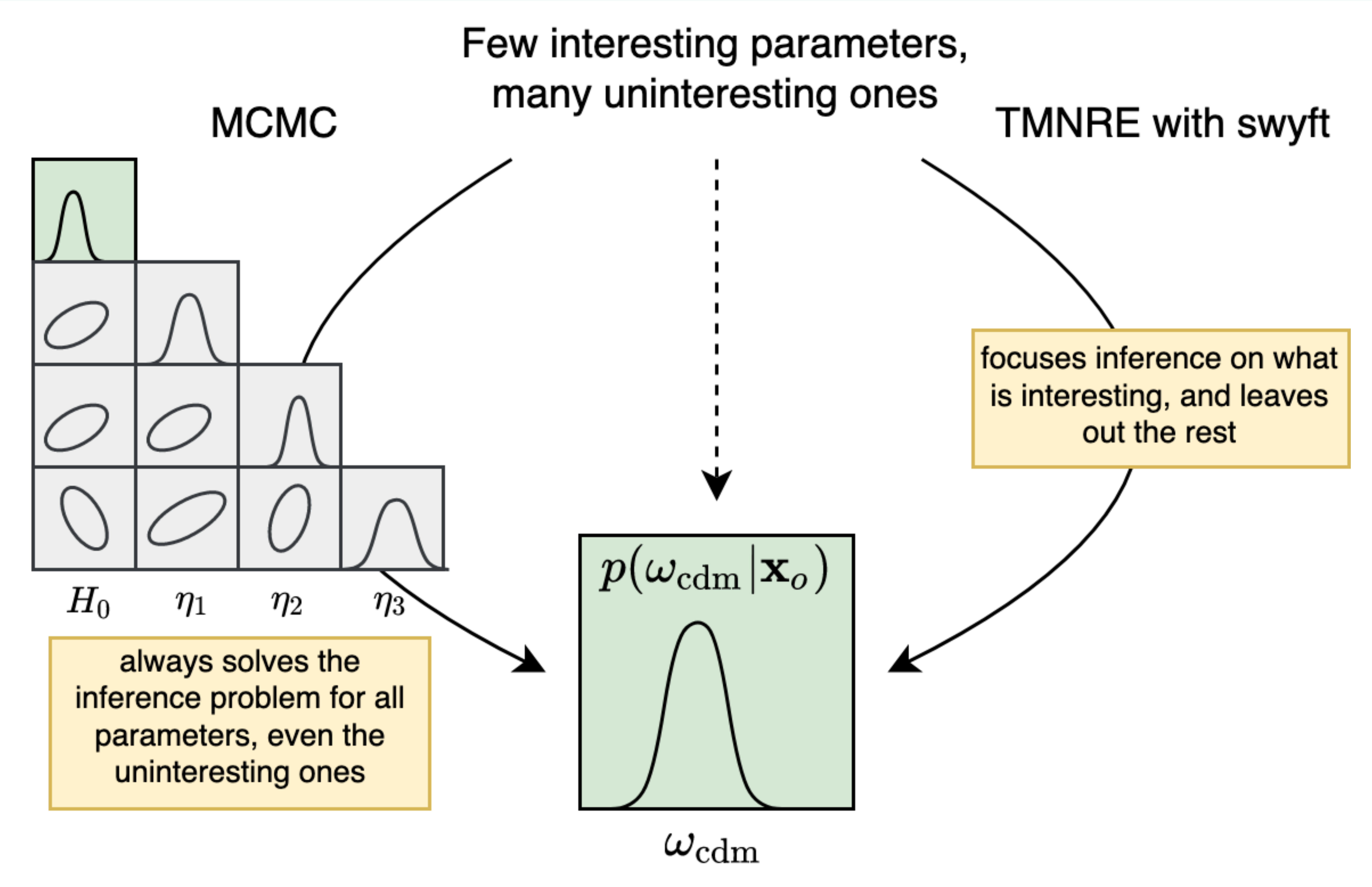
- For example, we **define the marginal ratio**

$$r(\theta, x) \equiv \frac{p(x\,|\,\theta)}{p(x)} = \frac{\int \mathrm{d}\eta\, p(x\,|\,\theta, \eta)p(\eta)}{p(x)}$$

which can be directly trained by omitting η from the information given to the classifier.
We train an individual network for each marginal ratio.

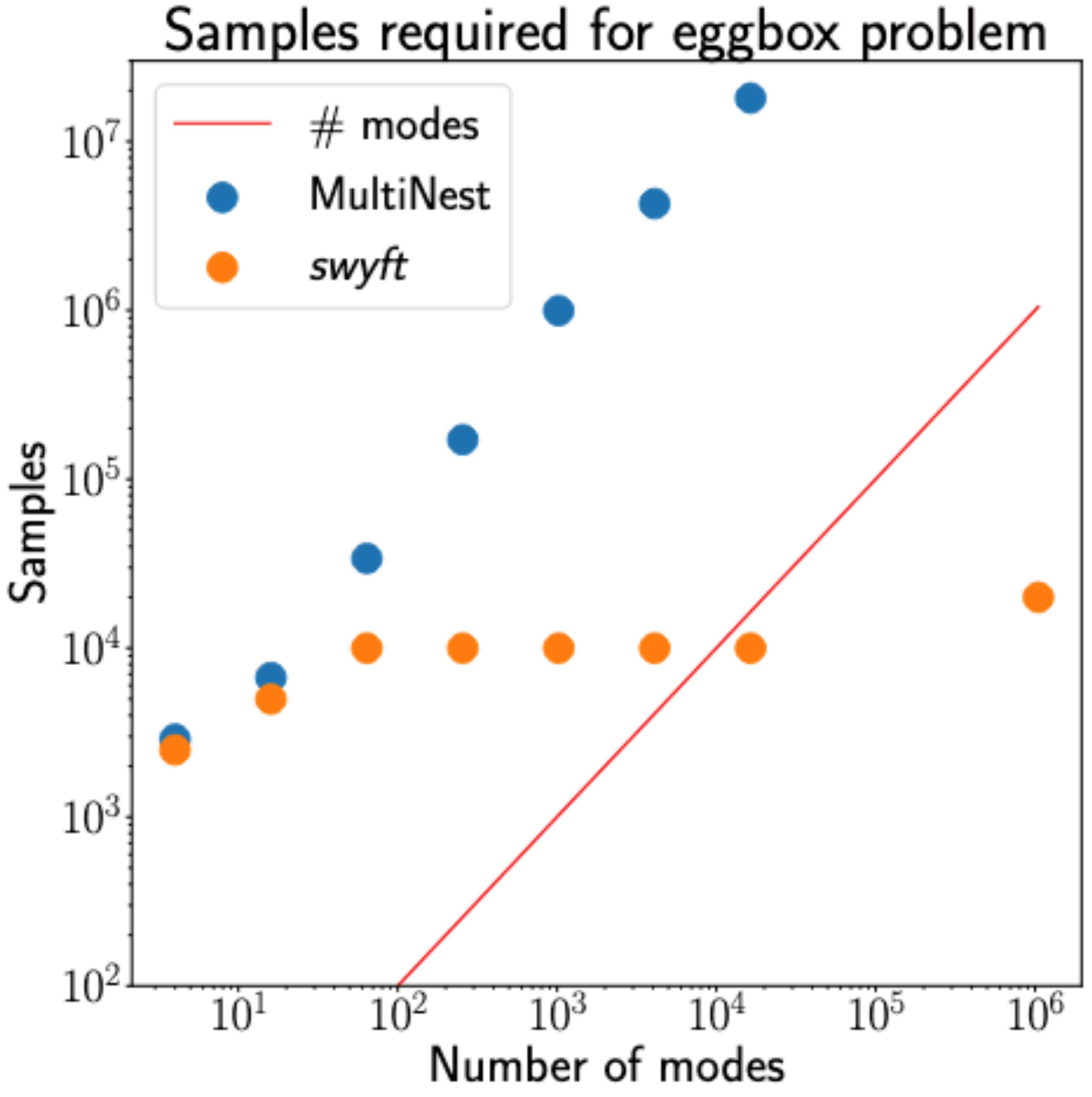
Alsing&Wandelt 2019; Hermans+ 2019; Miller+ 2020; Jeffrey&Wandelt 2020; Miller+ 2021

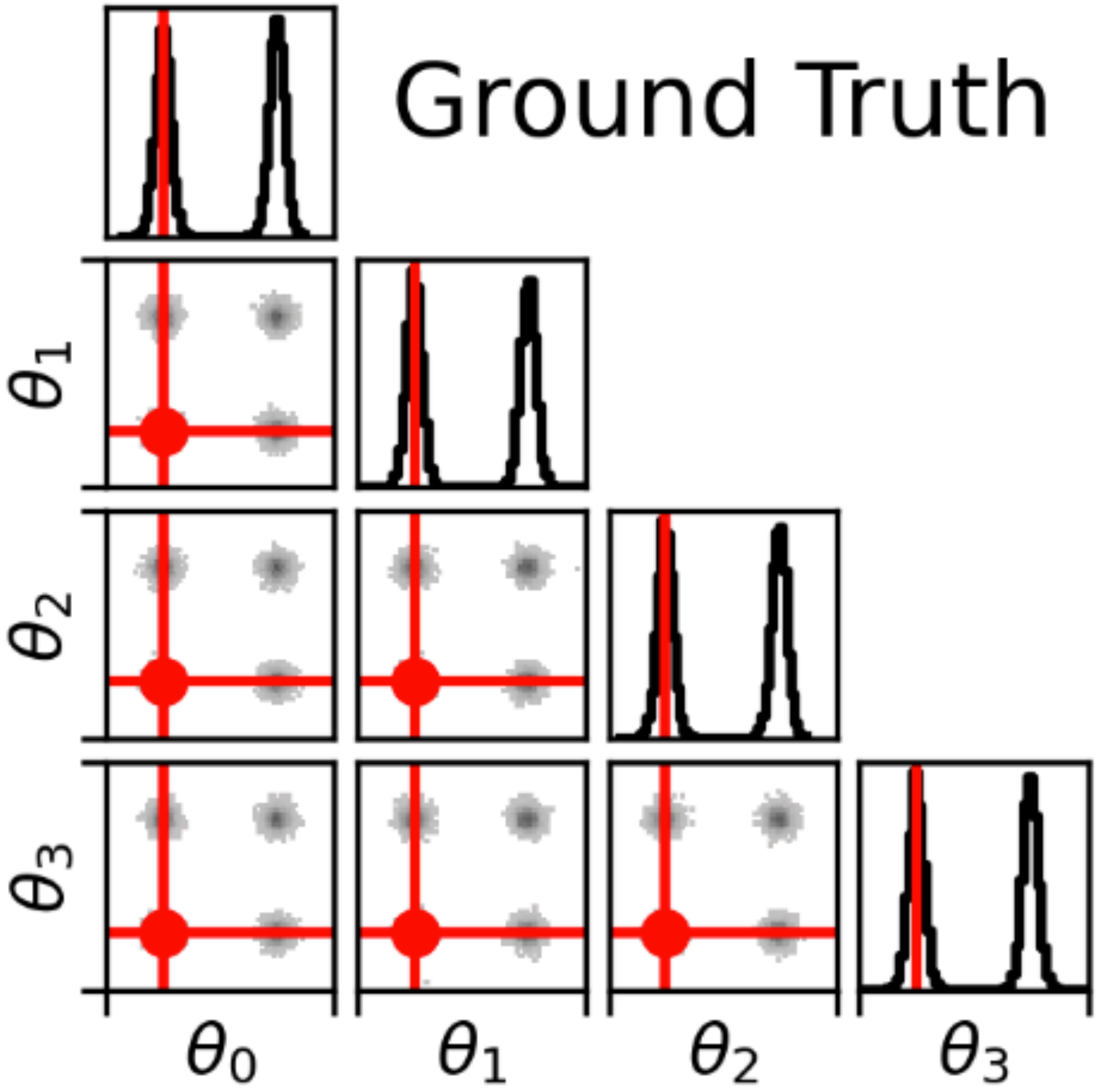
MARGINAL RATIO ESTIMATION



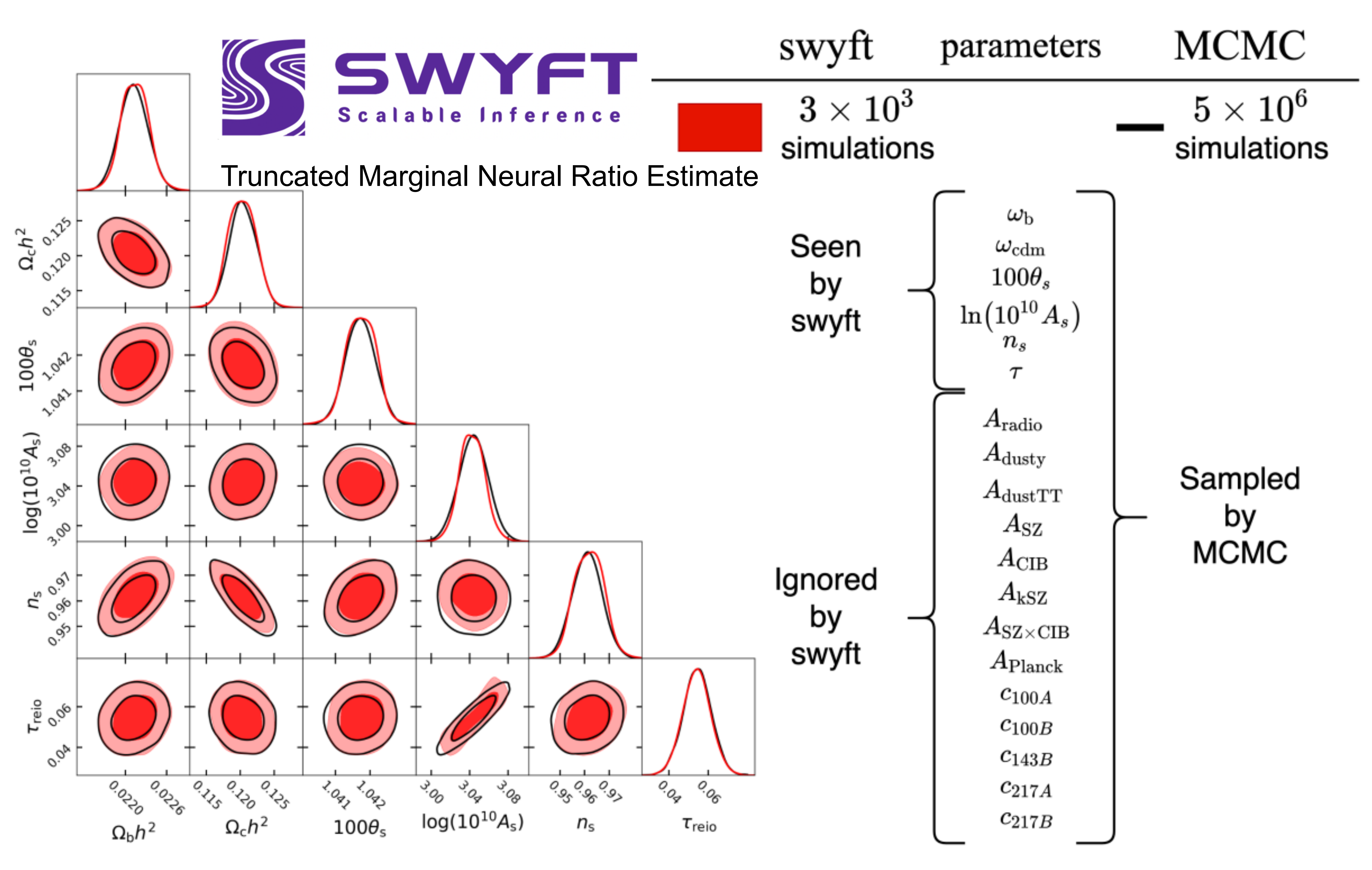
SOME BENEFITS OF AUTOMATIC MARGINALISATION

[Miller et al. ’20; Miller et al. ’21]





SBI: APPLICATION IN COSMOLOGY



Cole+2022

SEQUENTIAL METHODS / ACTIVE LEARNING

SEQUENTIAL METHODS

- Sequential Neural X Estimation: Use proposal density to select relevant simulation for training
- Note: definition of marginal X means nuisance parameters must be sampled from prior!

Sequential Neural Likelihood: Fast Likelihood-free Inference with Autoregressive Flows

George Papamakarios, University of Edinburgh
David C. Sterratt, University of Edinburgh
Iain Murray, University of Edinburgh

Automatic Posterior Transformation for Likelihood-free Inference

David S. Greenberg [1] Marcel Nonnenmacher [1] Jakob H. Macke [1]

Likelihood-free MCMC with Amortized Approximate Ratio Estimators

Joeri Hermans [1] Volodimir Begy [2] Gilles Louppe [1]

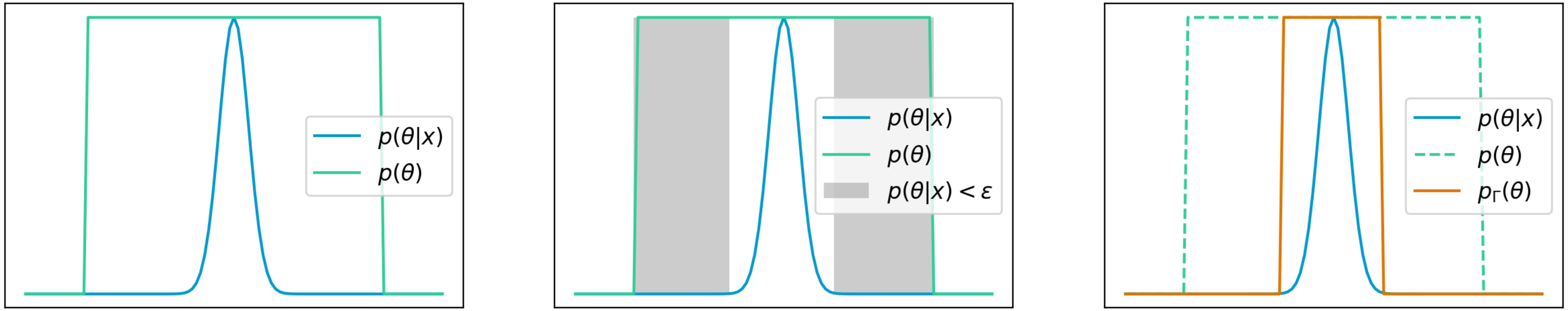
On Contrastive Learning for Likelihood-free Inference

Conor Durkan [1] Iain Murray [1] George Papamakarios [2]

Sequential Neural Posterior Estimation (SNPE, Papamakarios&Murray 2016; Lueckmann+2017; Greenberg+2019), “Truncated Marginal Neural Ratio Estimation” (TMNRE)

Truncation

[Miller et al. ’20; Miller et al. ’21]



- Sometimes priors are much wider than posteriors. Let’s call the relevant region of parameter space Γ.

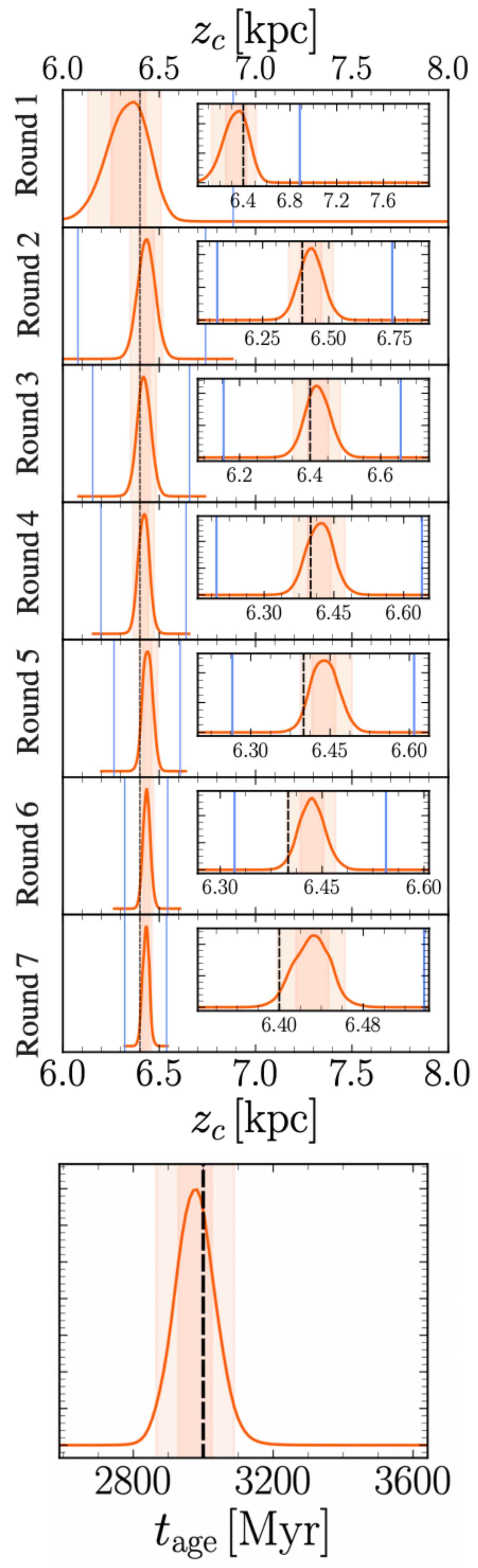
- We **zoom into the relevant region** by approximating Γ (requiring $\hat{p}(\theta\,|\,x) > \epsilon$) in a series of rounds.

- With marginal posteriors, Γ is approximated via a product of low-dimensional projections. These can reflect expected correlations.

Slides adapted from A. Cole

SBI: APPLICATION FOR STELLAR STREAMS

TRUNCATED MARGINAL NEURAL RATIO ESTIMATION (TMNRE) FOR STELLAR STREAMS



STEP 1: (RE-) SIMULATE

- Sample parameters θ from (truncated) prior $p(\theta)$
- Simulate data $x \sim p(x\,|\,\theta)$

$$\theta \equiv (t_{\mathrm{age}}, \sigma_v, \ldots) \rightarrow x = \text{stream} + \text{bkg}\,.$$

STEP 2: RATIO ESTIMATION

- Train ratio estimators $r(x;\theta_i)$ on simulated data to approximate the posterior-to-prior ratio $r(x;\theta_i) \sim p(\theta_i\,|\,x)/p(\theta_i)$ for each parameter of interest θ_i

sstrax

FAST STELLAR STREAM MODELLING IN JAX

ALBATROSS

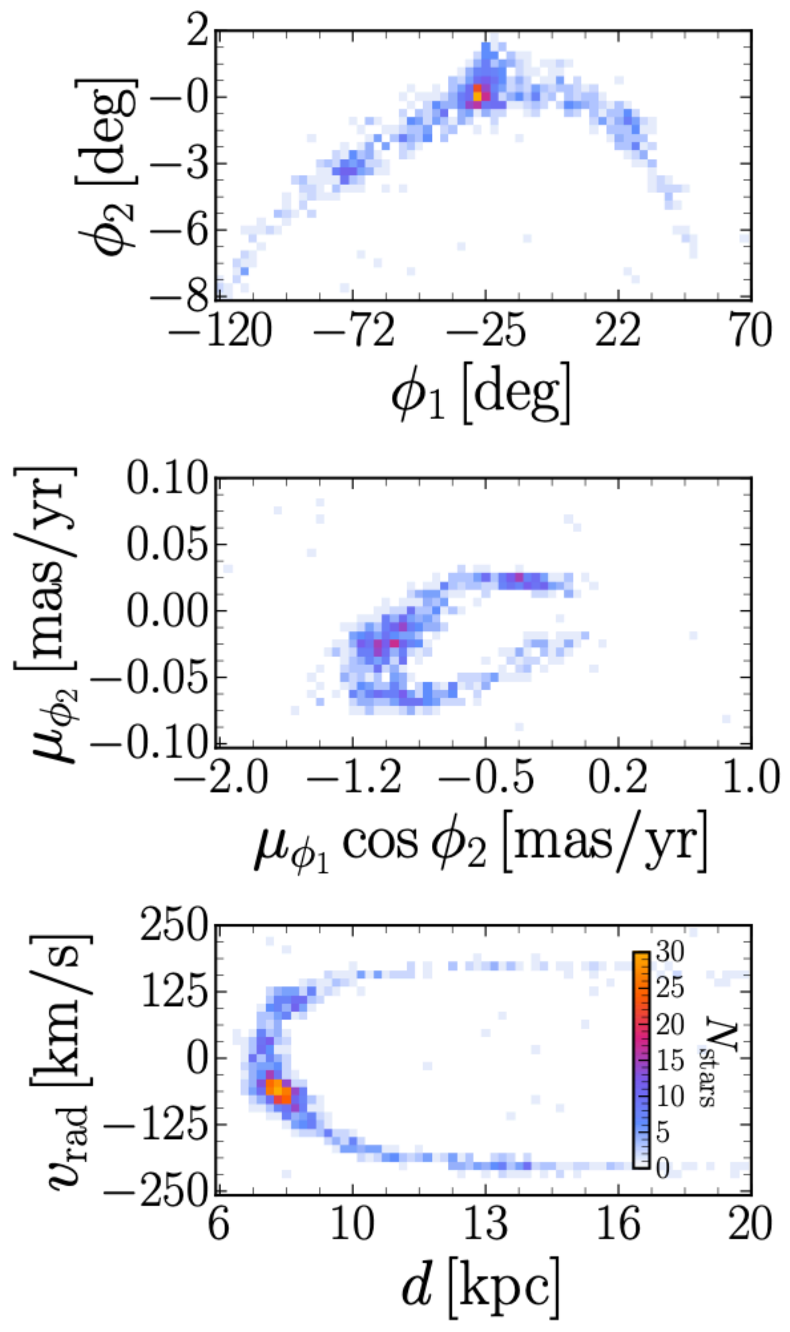
STELLAR STREAM PARAMETER INFERENCE WITH NEURAL RATIO ESTIMATION

STEP 4: TRUNCATION

- Use ratios $r(x_0;\theta_i)$ to remove regions with extremely low posterior density
- If this leads to a reduction in prior volume, re-simulate from **Step 1** with truncated prior
- Repeat until converged across all parameters, then obtain resulting posterior

STEP 3: INFERENCE

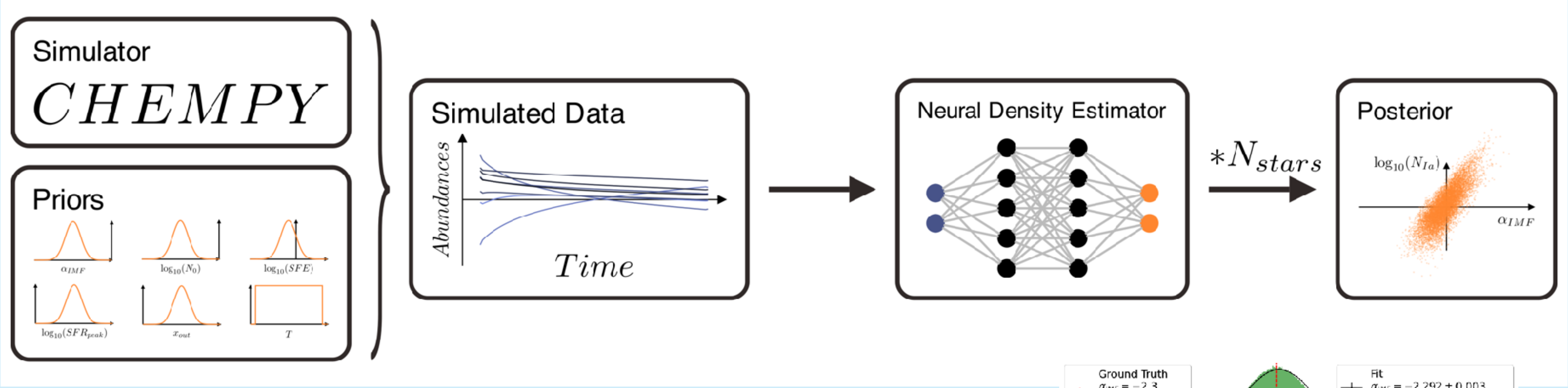
- Obtain a prior sample from $p(\theta_i)$
- Target a specific observation x_0 and compute the ratios $r(x_0;\theta_i)$ across the prior sample
- Weight the samples according to this ratio

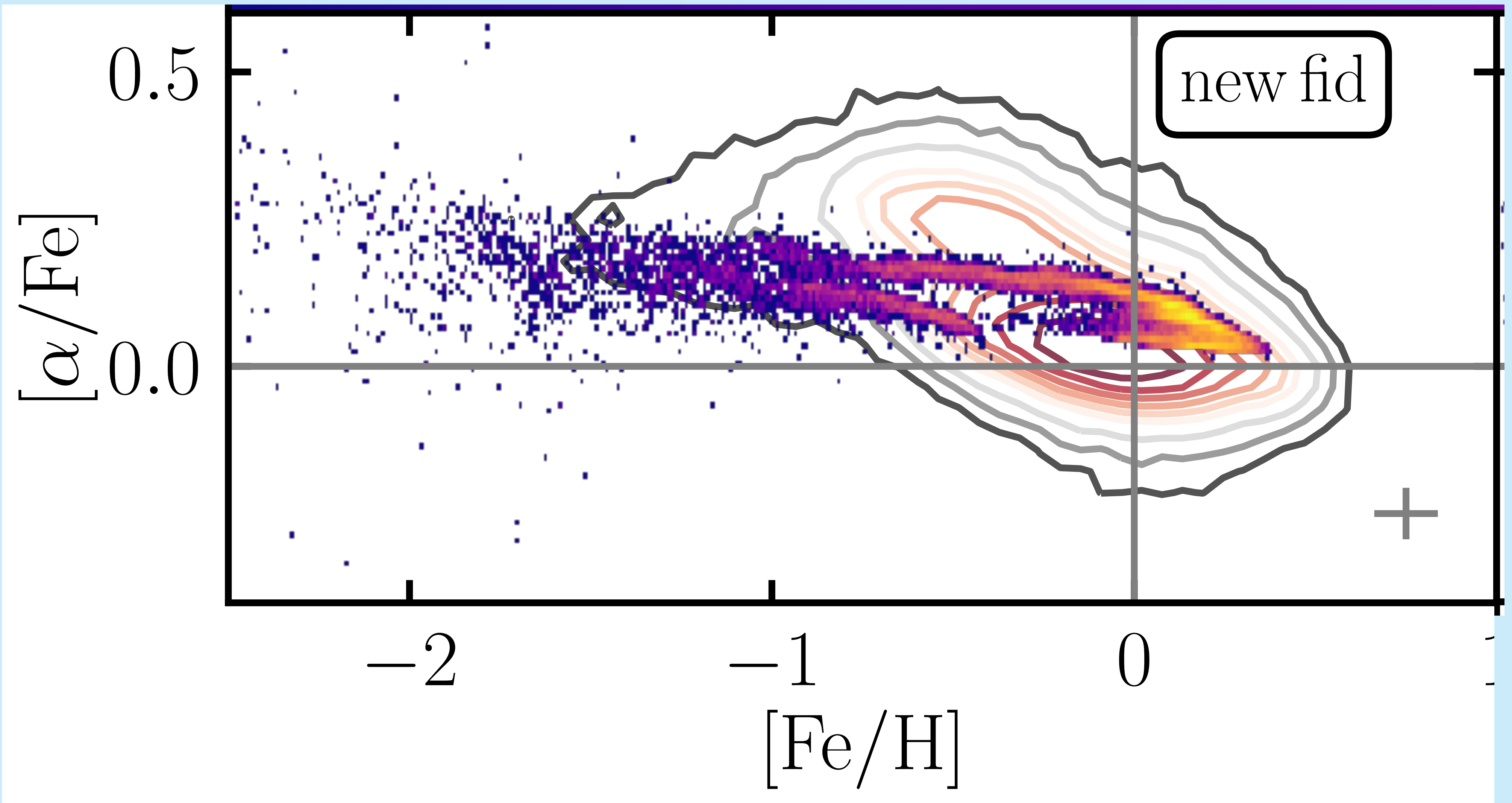


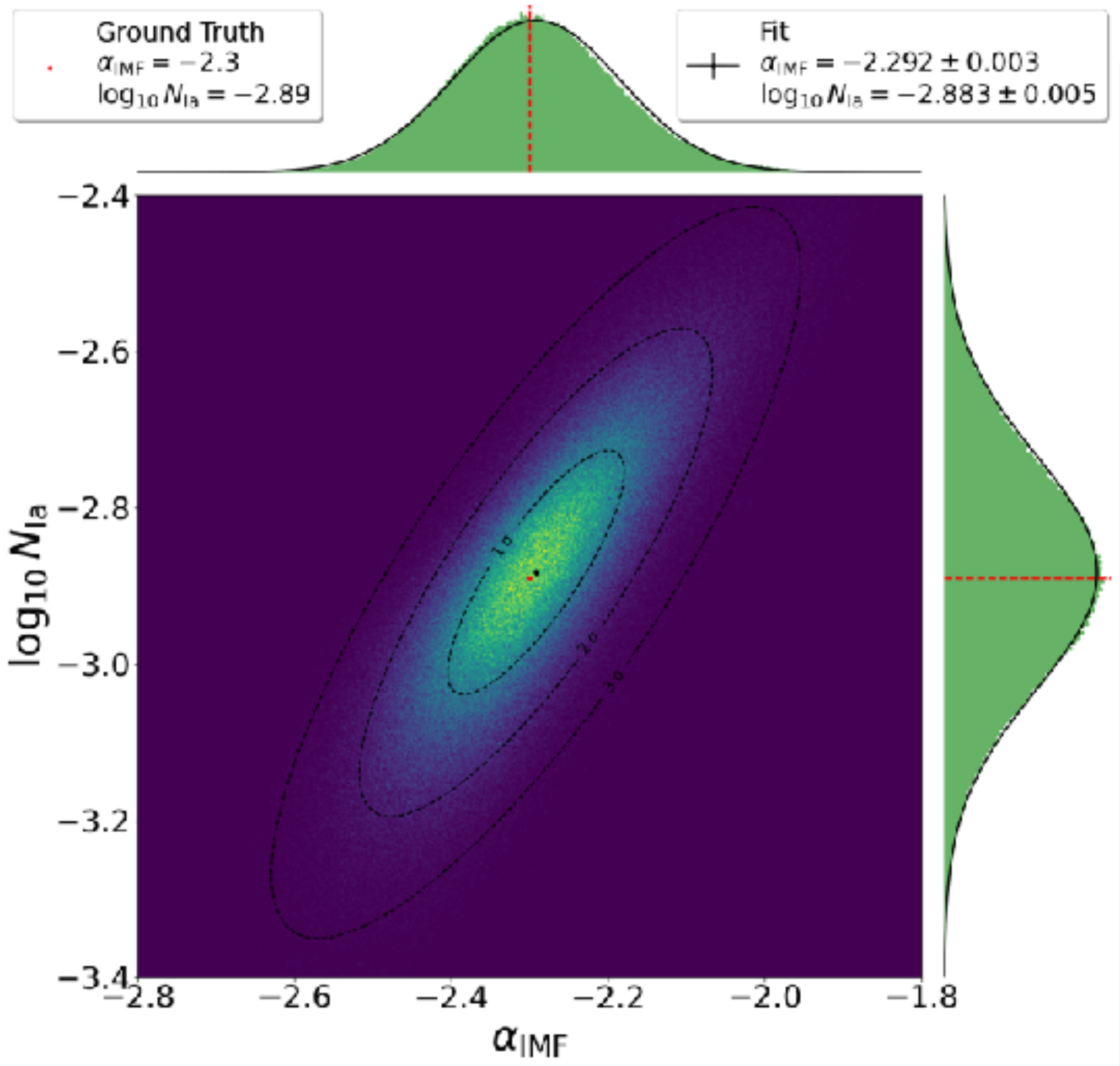
Observation, x_0

Alvey+2024

SBI: APPLICATION FOR GALACTIC CHEMICAL ENRICHMENT



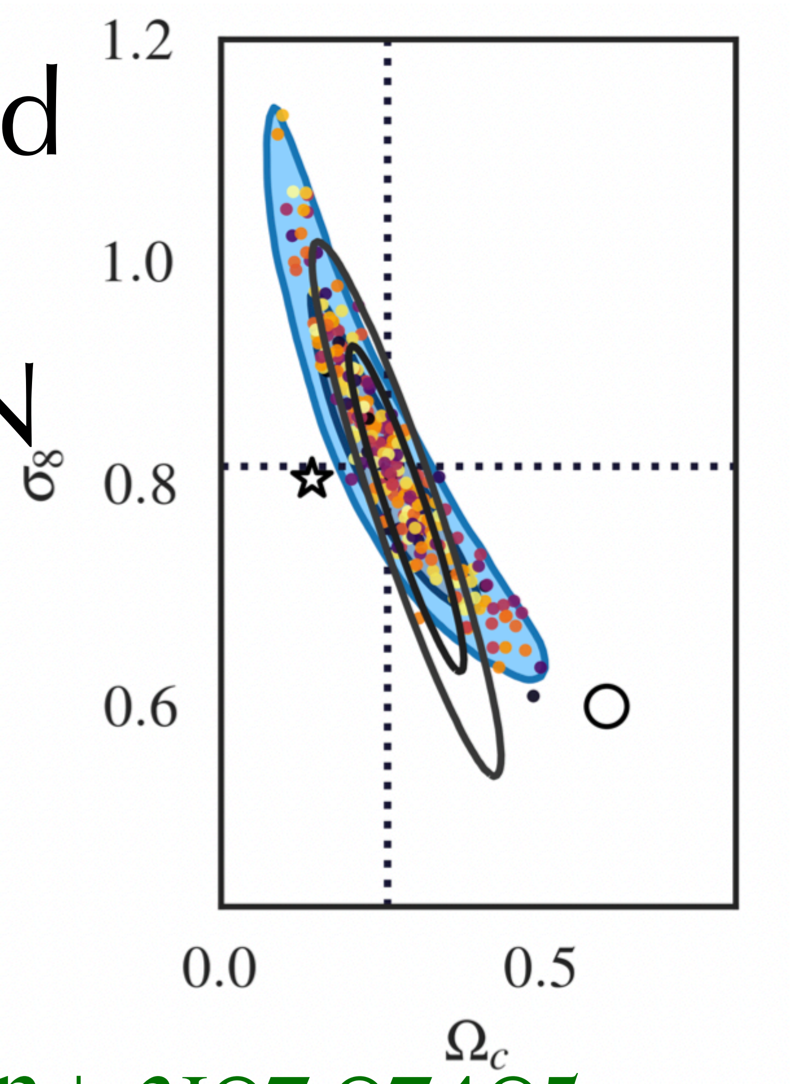




Günes+Buck subm.

SBI: APPLICATION IN COSMOLOGY

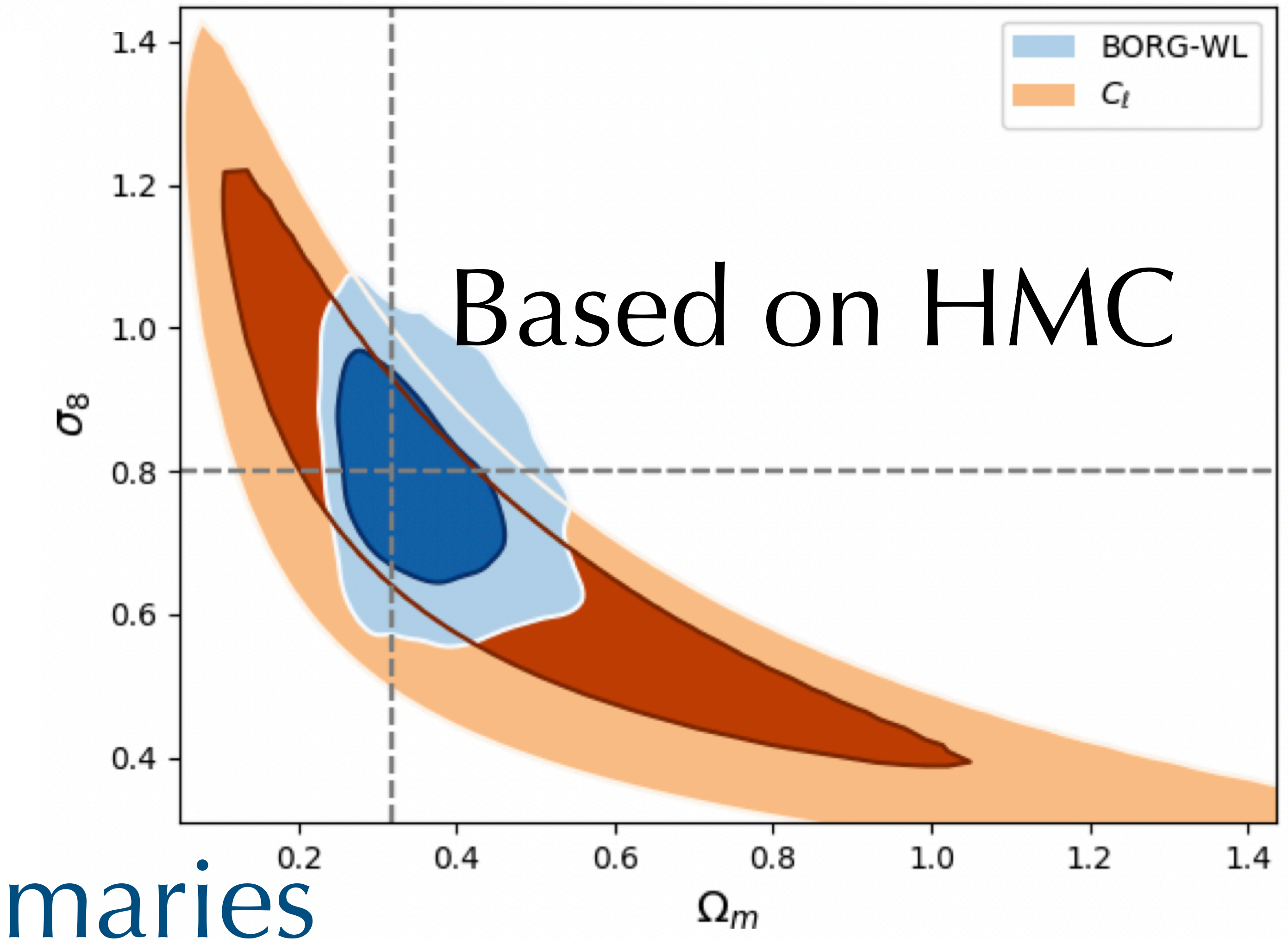
Based on CNN



Breaking degeneracy between DM density and power-spectrum amplitude

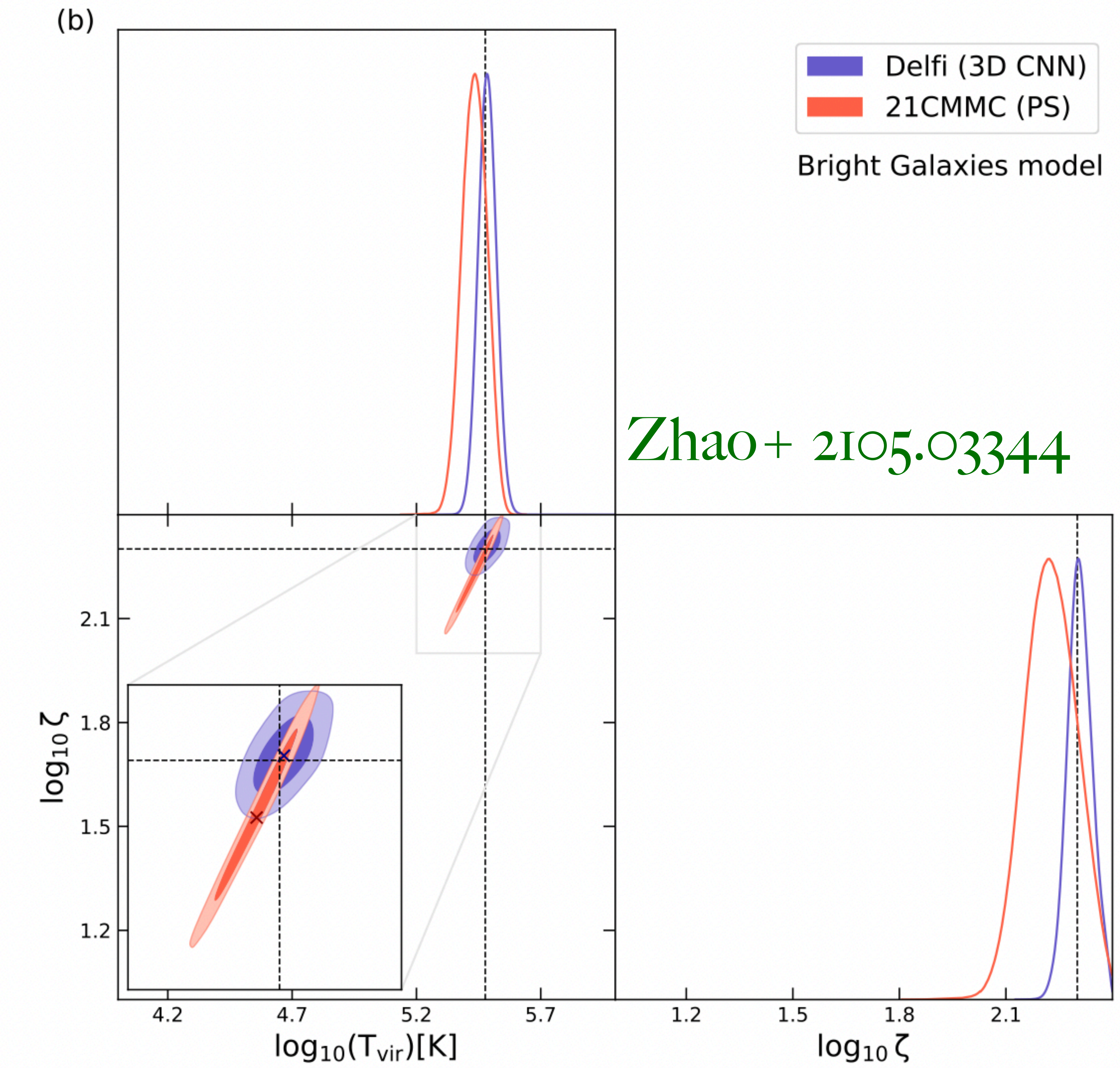
Makinen+ 2107.07405

Porqueres+ 2108.04825



Based on HMC

Alternative to:
Hand-crafted summaries



Zhao+ 2105.03344

Breaking degeneracy between ionisation parameters T_{vir} and ζ

slide from Cole

SBI: APPLICATION IN STRONG LENSING

Searching light DM halos

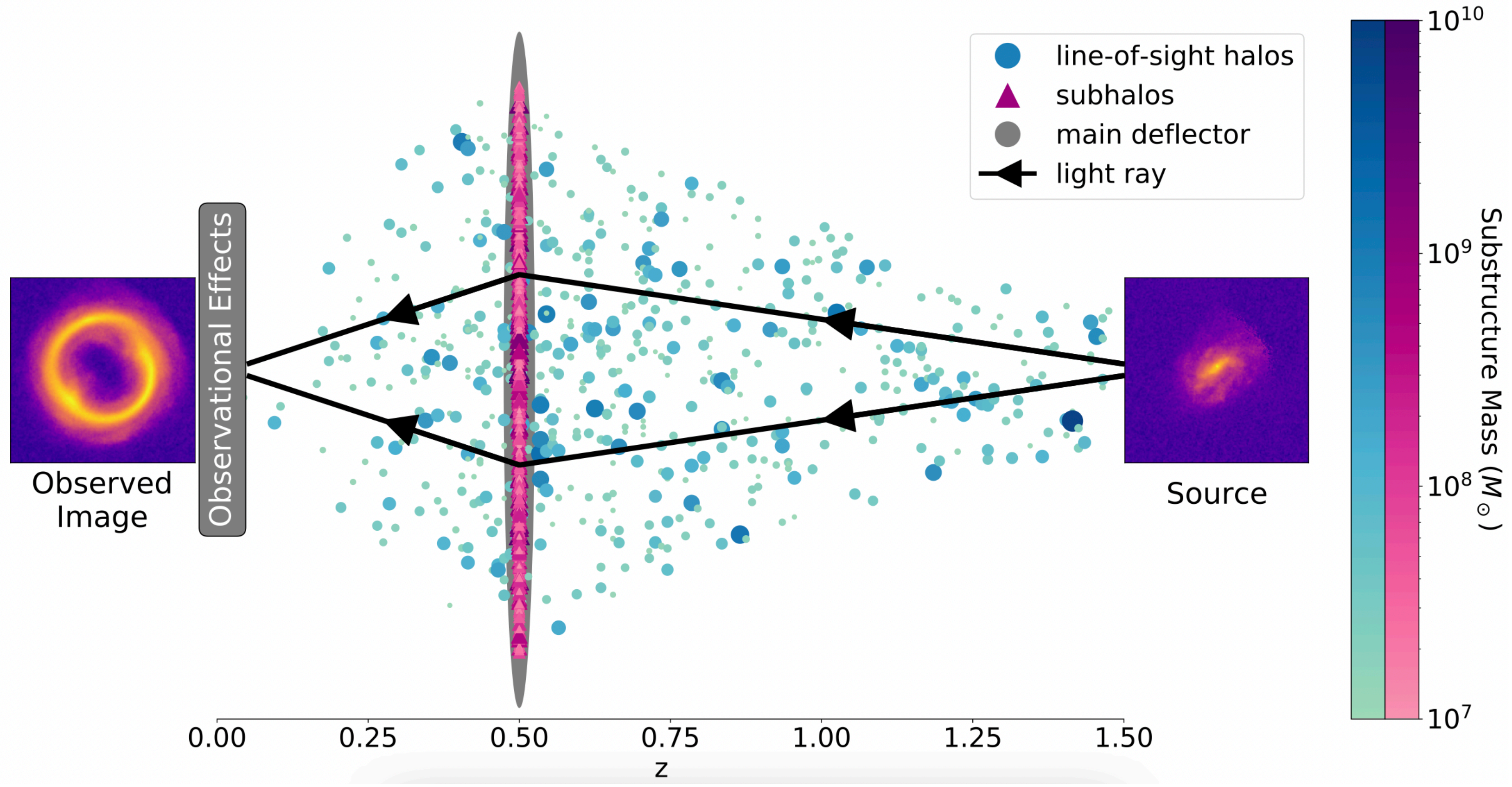
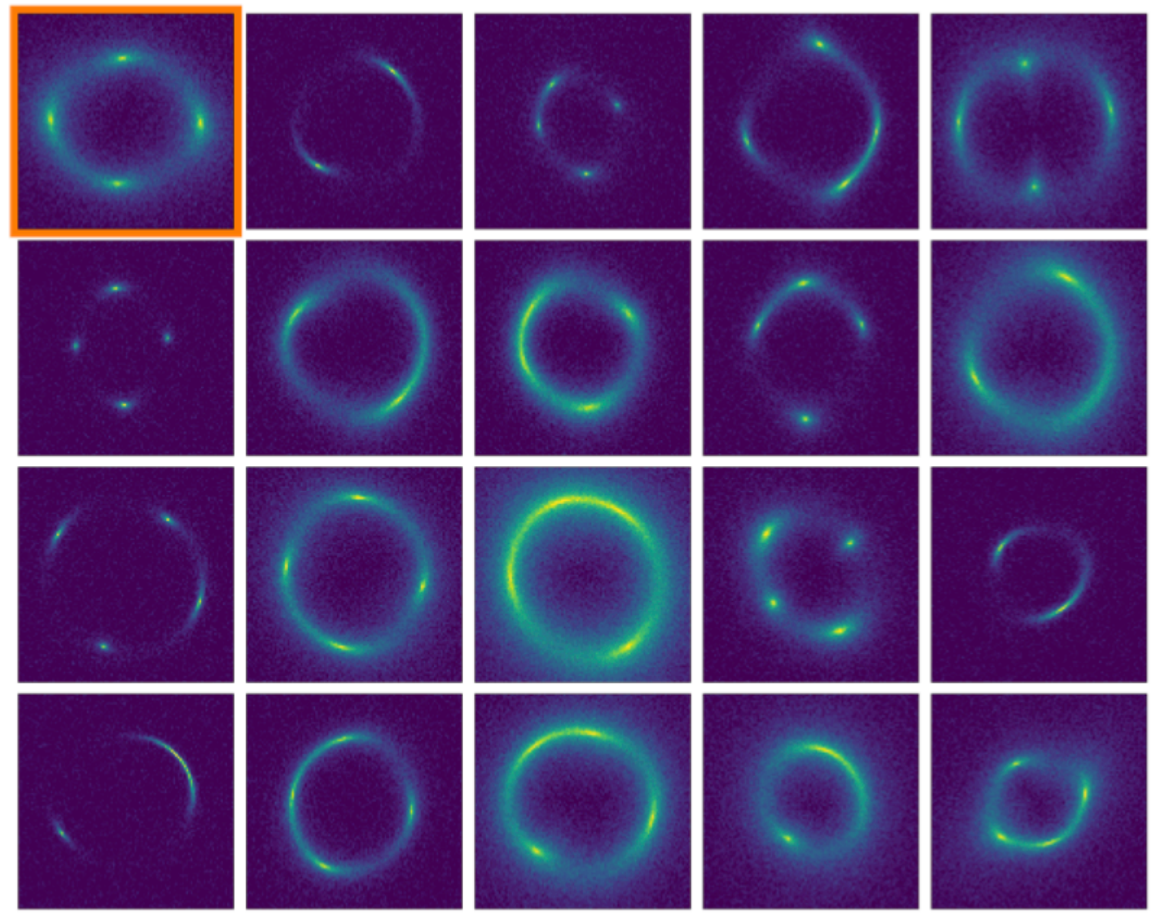
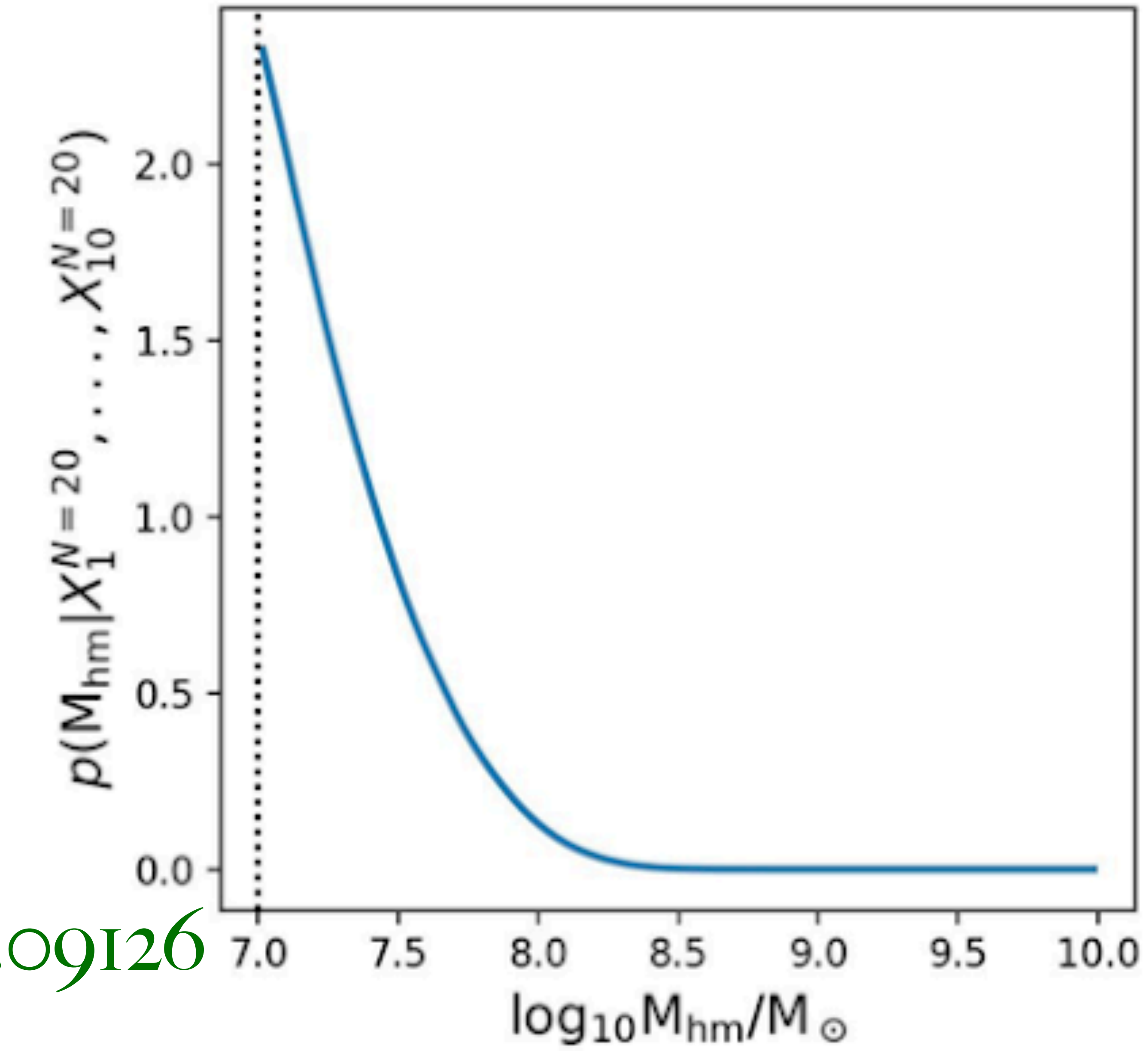


Image credit: Wagner-Carena+ 2203.00690

Halo mass function cutoff

Probing **population effects of light dark matter halos** rather than individual detections





Anau Montel+ 2205.09126

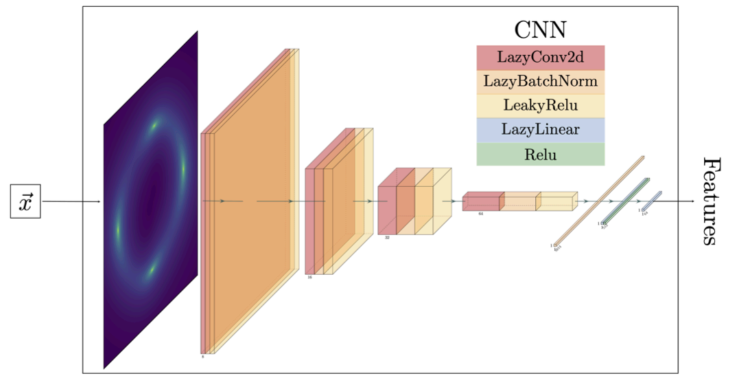
Alternative to:
HMC, parameter reduction, ABC, …

Related work: He+ 2010.13221 (similar in spirit, using ABC)

Wagner-Carena+ 2203.00690 (constraining subhalo mass function normalization)

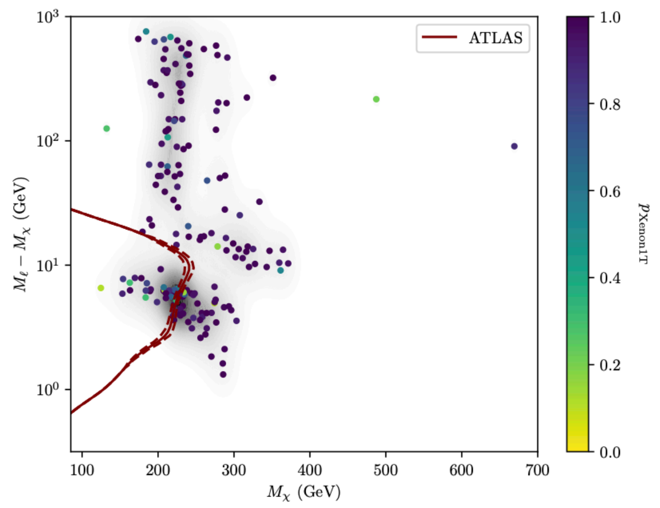
slide from Cole

SBI: APPLICATION IN STRONG LENSING



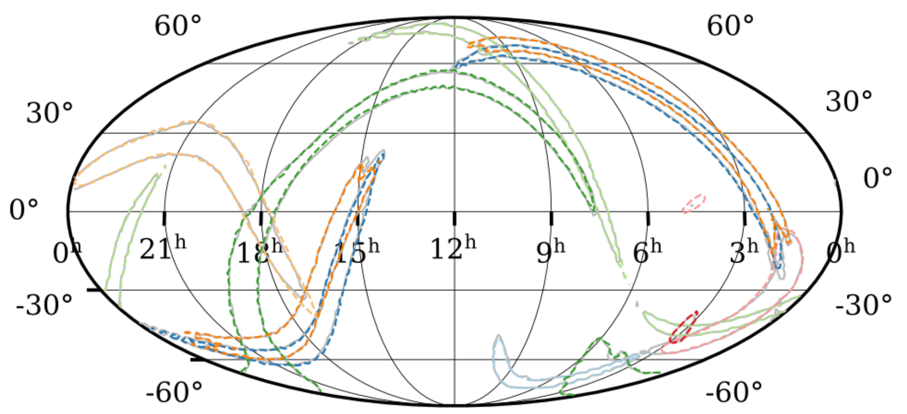
Strong lensing

Brehmer+ 1909.02005, Coogan+ 2010.07032, Legin+ 2112.05278, Wagner-Carena+ 2203.00690, Anau Montel+ 2205.09126, Coogan+ 2207.xxxxx



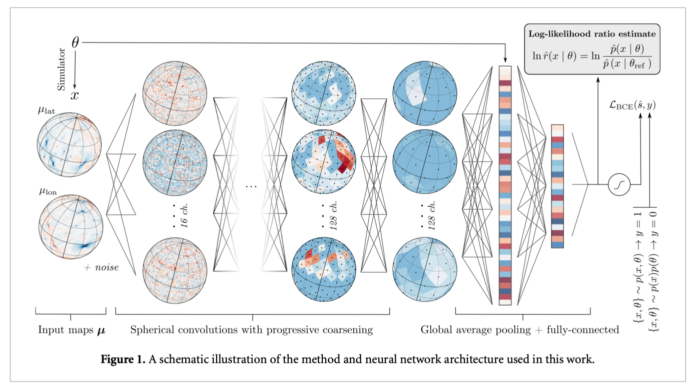
Effective field theory

Morrison+ 2203.13403



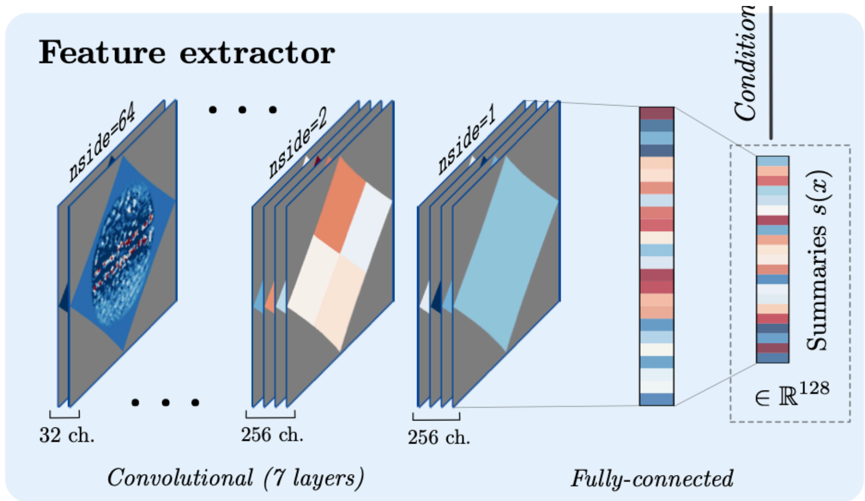
GW parameters

Delaunoy+ 2010.12931, Dax+ 2106.12594, ...



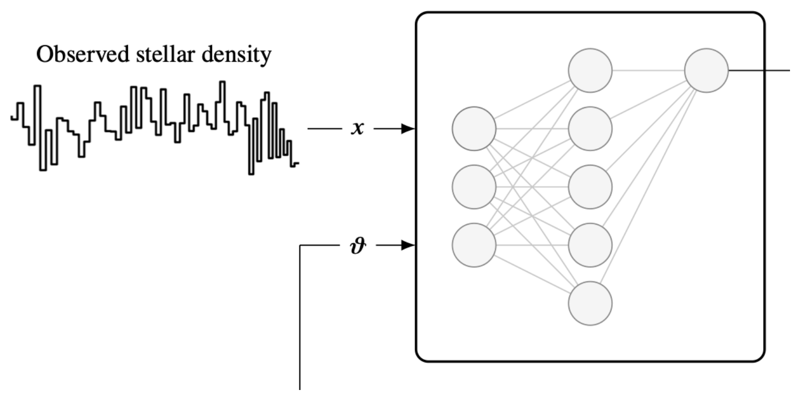
Astrometry

Mishra-Sharma+ 2110.01620



Fermi GeV excess

Mishra-Sharma+ 2110.06931



Stellar streams

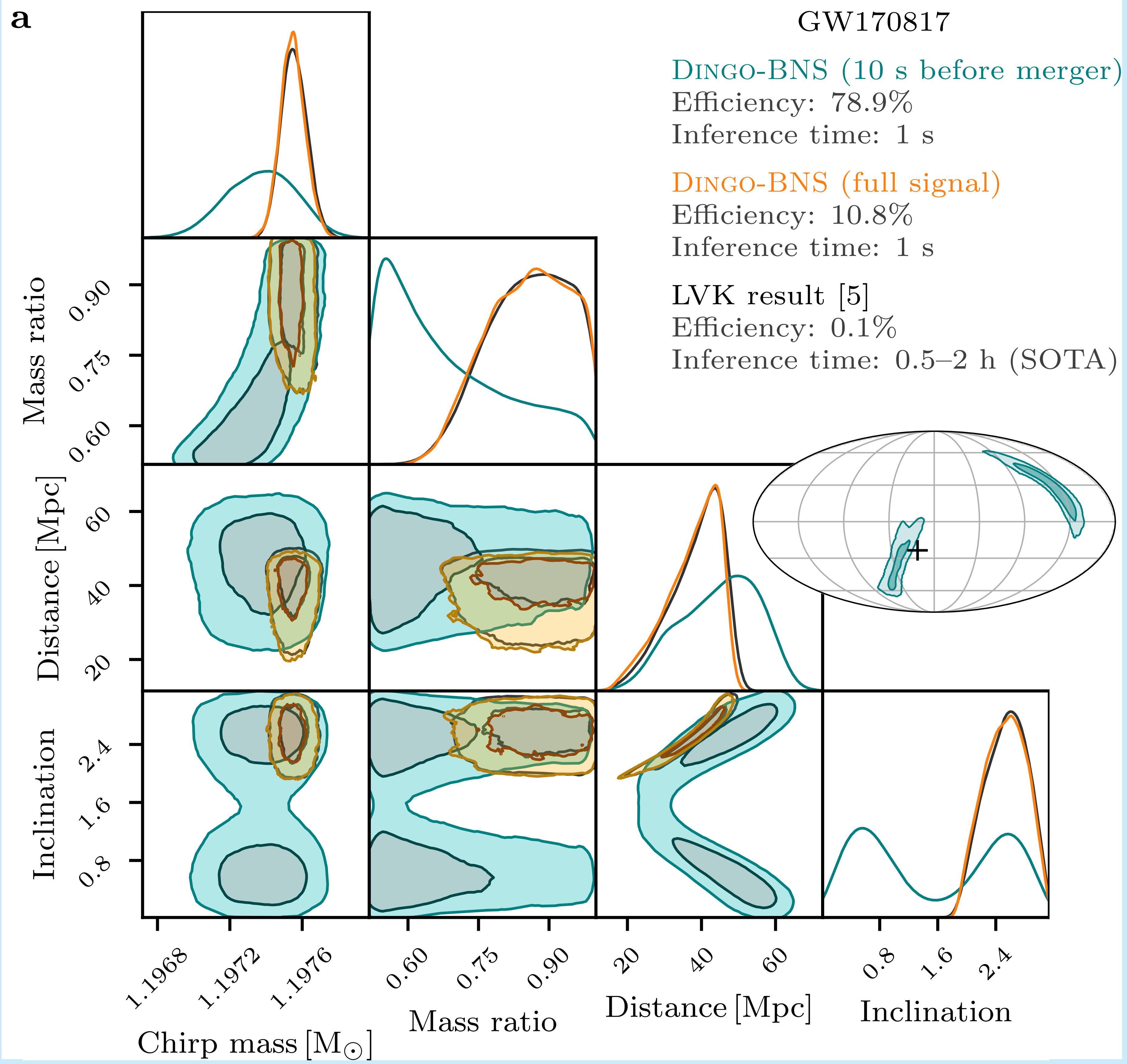
Hermans+ 2011.14923

Single frequency CMB B-mode inference with realistic foregrounds from a single training image

Niall Jeffrey,[1,2*] François Boulanger,[1] Benjamin D. Wandelt,[3,4] Bruno Regaldo-Saint Blancard,[1,5] Erwan Allys,[1] François Levrier[1]

slide from Cole

Gravitational wave parameter estimation



Dax+2021,2023,2024

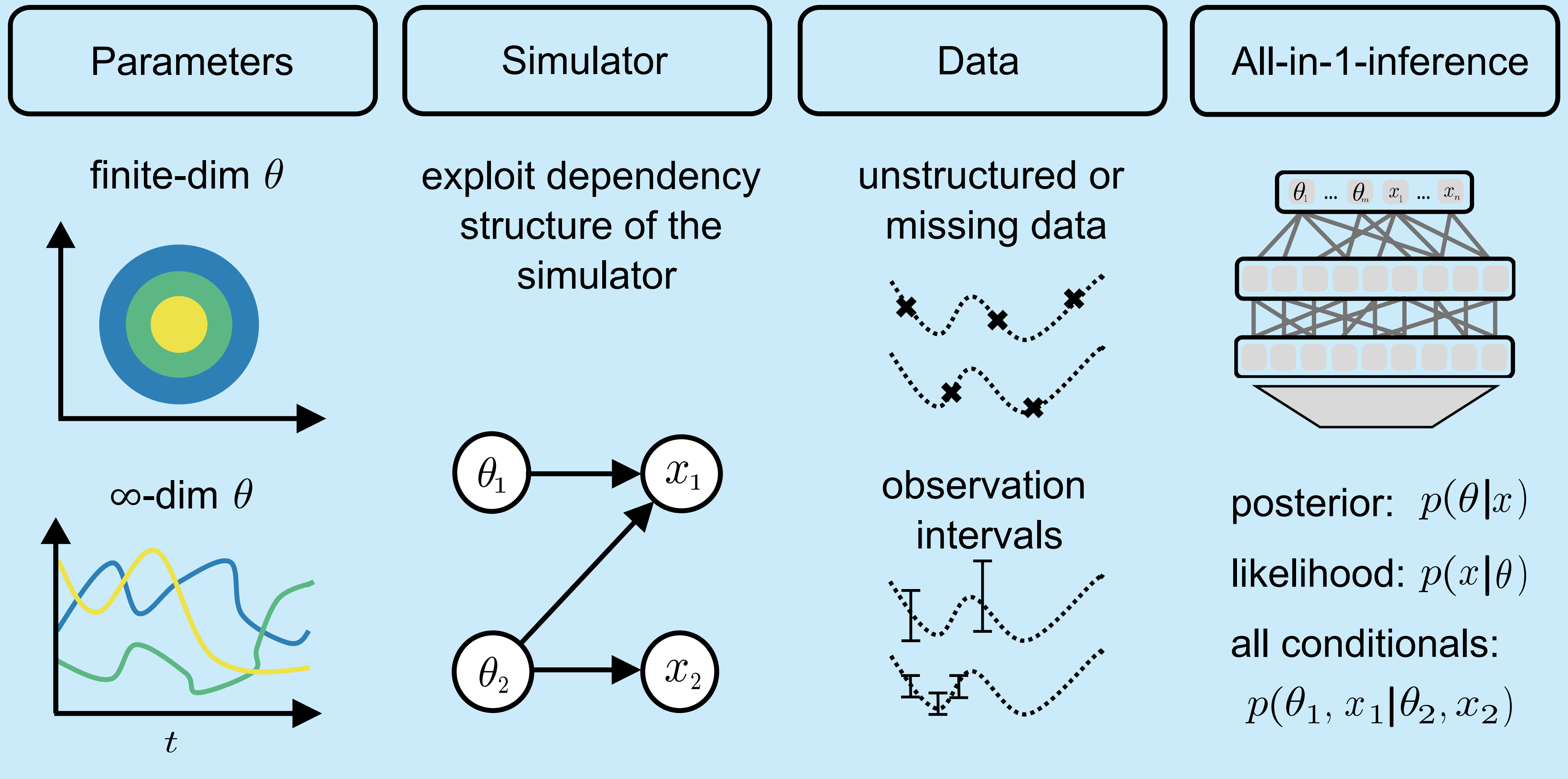
SBI WITH TRANSFORMER ARCHITECTURE

All-in-one simulation-based inference

Manuel Gloeckler [1] **Michael Deistler** [1] **Christian Weilbach** [2] **Frank Wood** [2] **Jakob H. Macke** [1 3]

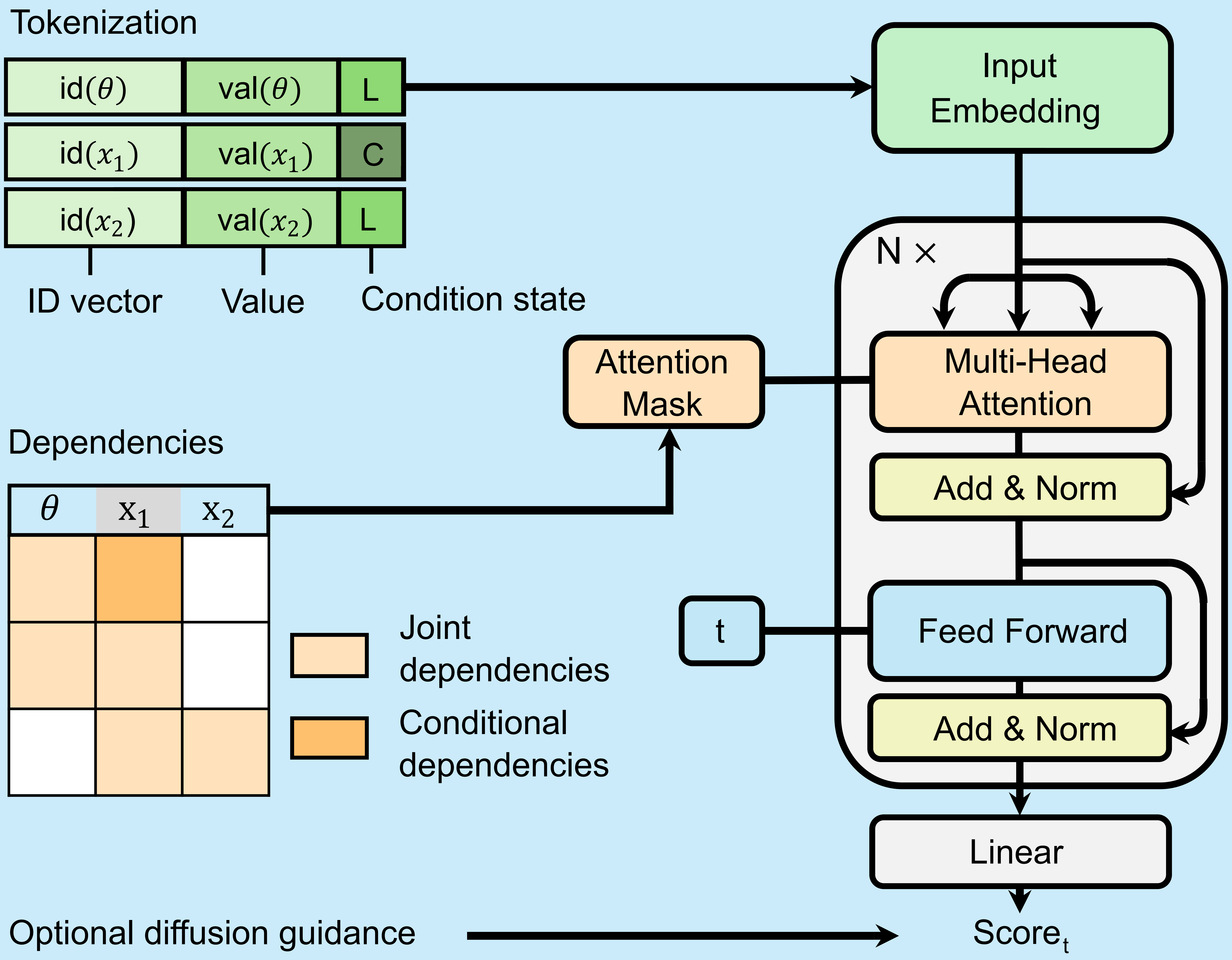
Abstract

Amortized Bayesian inference trains neural networks to solve stochastic inference problems using model simulations, thereby making it possible to rapidly perform Bayesian inference for any newly observed data. However, current simulation-based amortized inference methods are simulation-hungry and inflexible: They require the specification of a fixed parametric prior, simulator, and inference tasks ahead of time. Here, we present a new amortized inference method—the Simformer—which overcomes these limita-



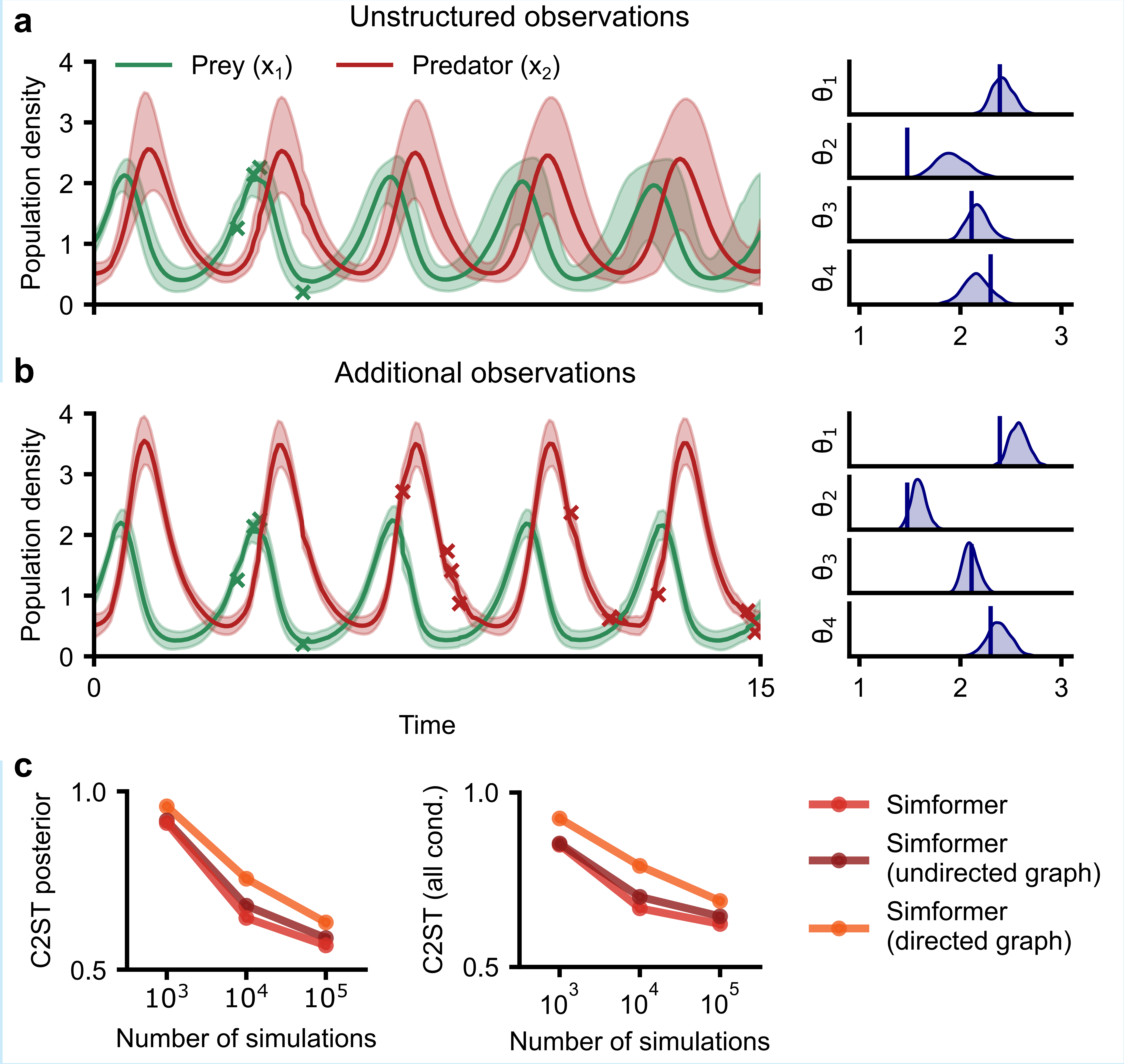
Gloeckler+2024

SBI WITH TRANSFORMER ARCHITECTURE



Gloeckler+2024

SBI WITH TRANSFORMER ARCHITECTURE



Gloeckler+2024

SOME FINAL WORDS…

AMORTIZATION AND CONSISTENCY

AMORTIZATION & CONSISTENCY

- Once trained, the network can rapidly generate posteriors for any data drawn from $p(x\,|\,\theta)p_\Gamma(\theta)$. This is called **“amortization”**.

- This enables **rapid tests of statistical consistency** that are not possible with sampling-based methods.

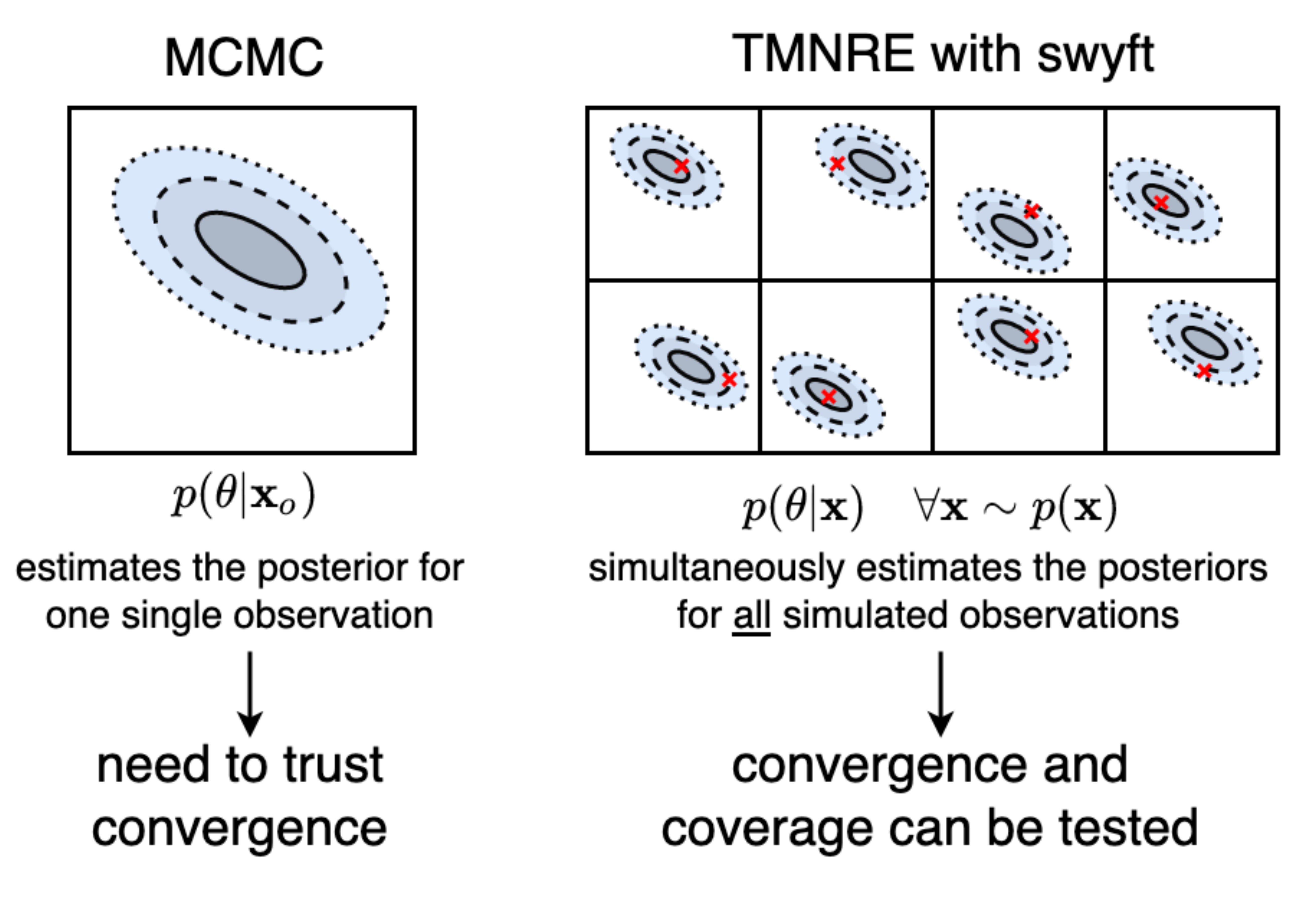


image credit A. Cole

AMORTIZATION & CONSISTENCY

- We can therefore draw many samples from our simulation bank, generate posteriors, and see **how often the true parameters lie within the N % credible region**.

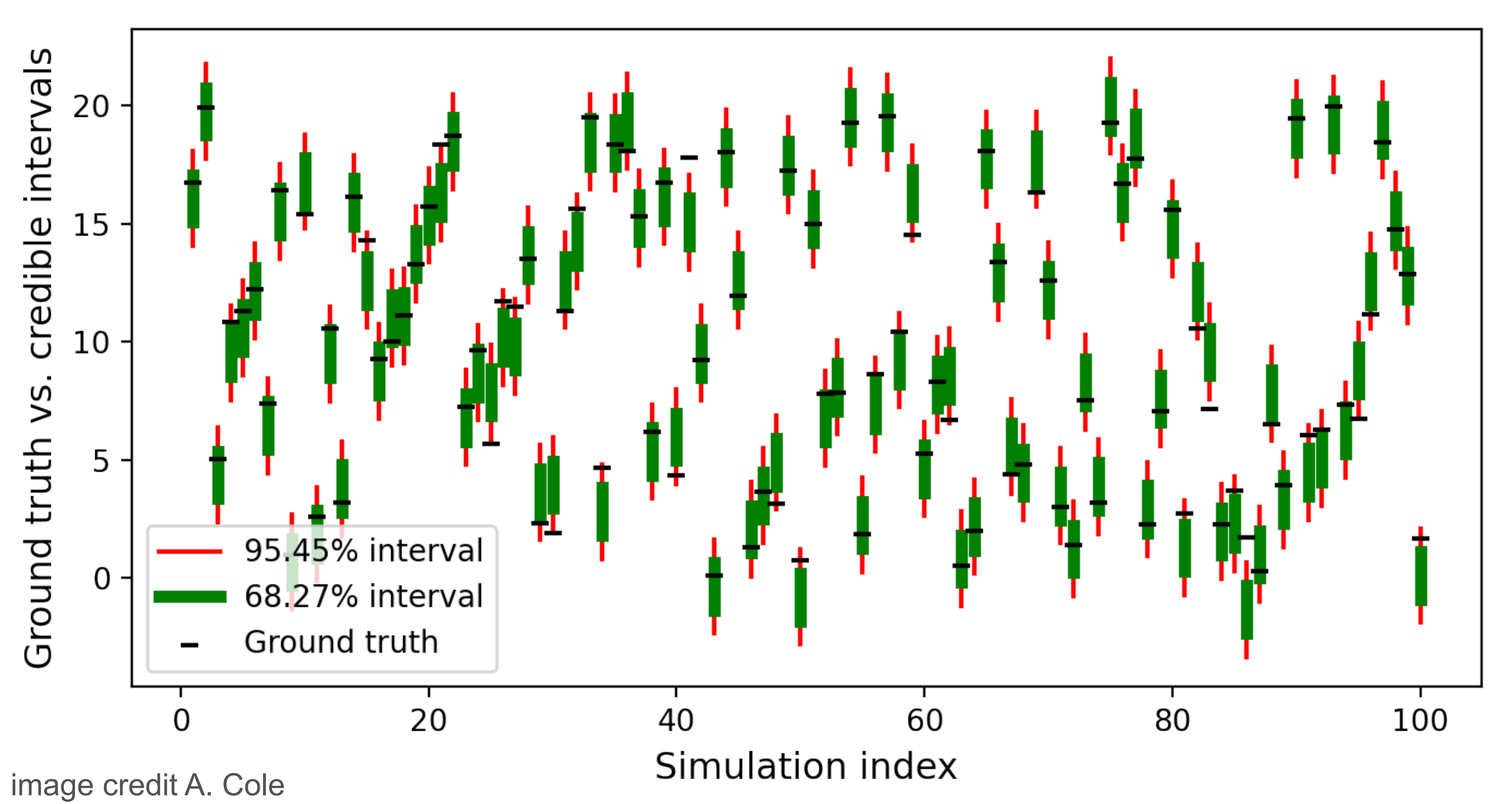
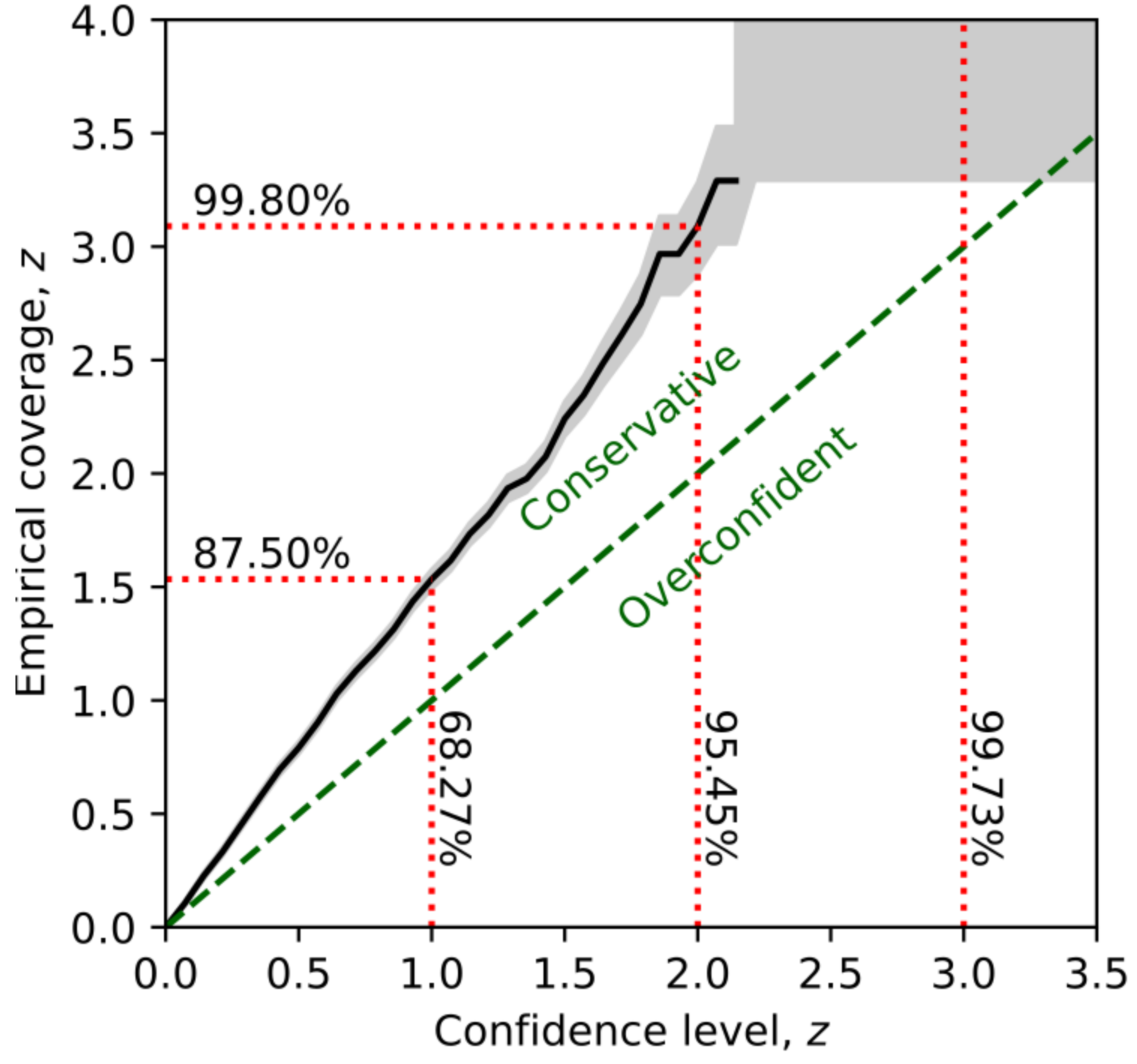


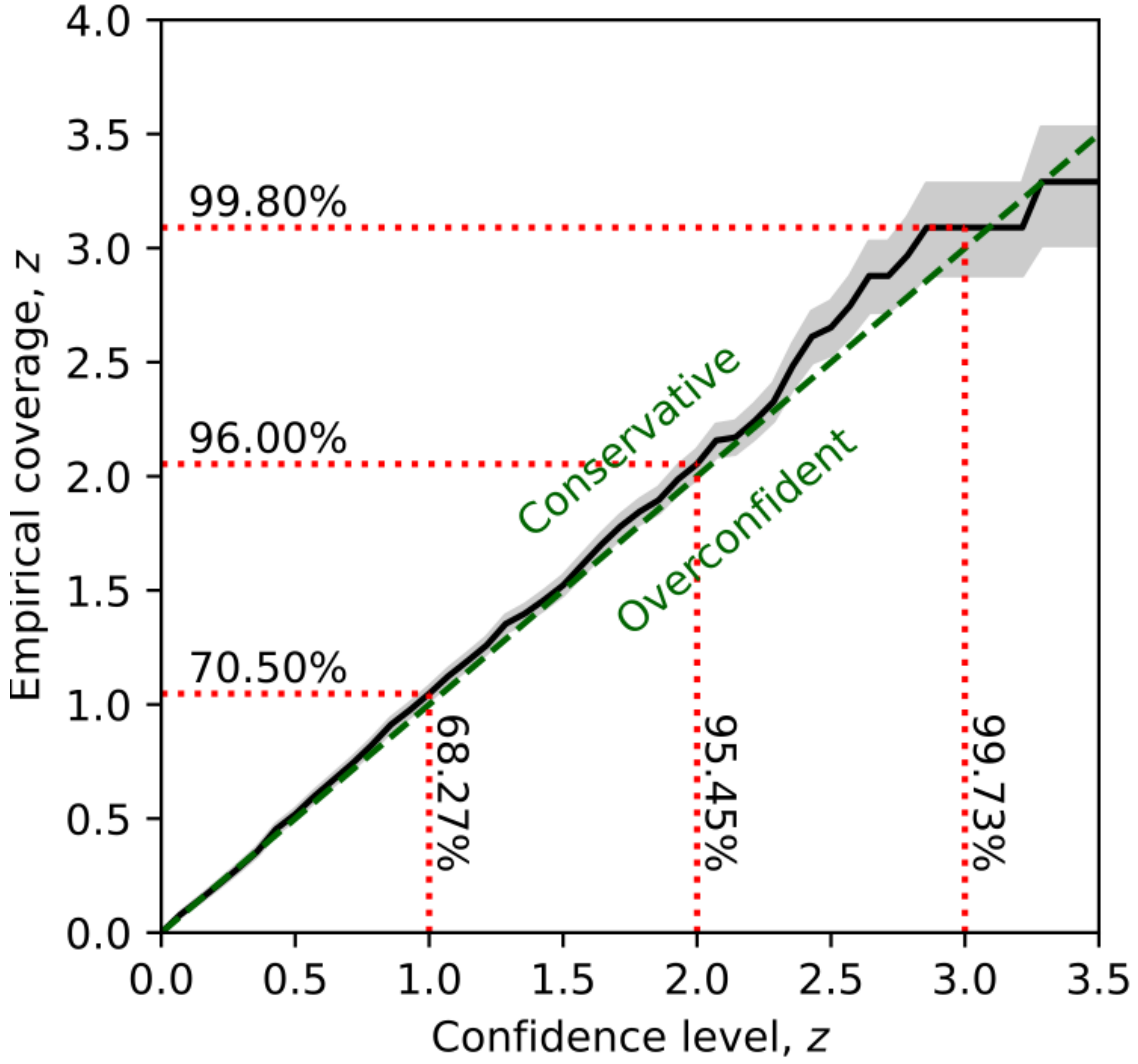
image credit A. Cole

AMORTIZATION & CONSISTENCY

- We compare the network's predictions to the empirical coverage to **assess convergence** and ensure our network is not overconfident.
- This consistency test makes no reference to likelihoods or the true parameters of observed data.



still converging…



converged!

image credit A. Cole

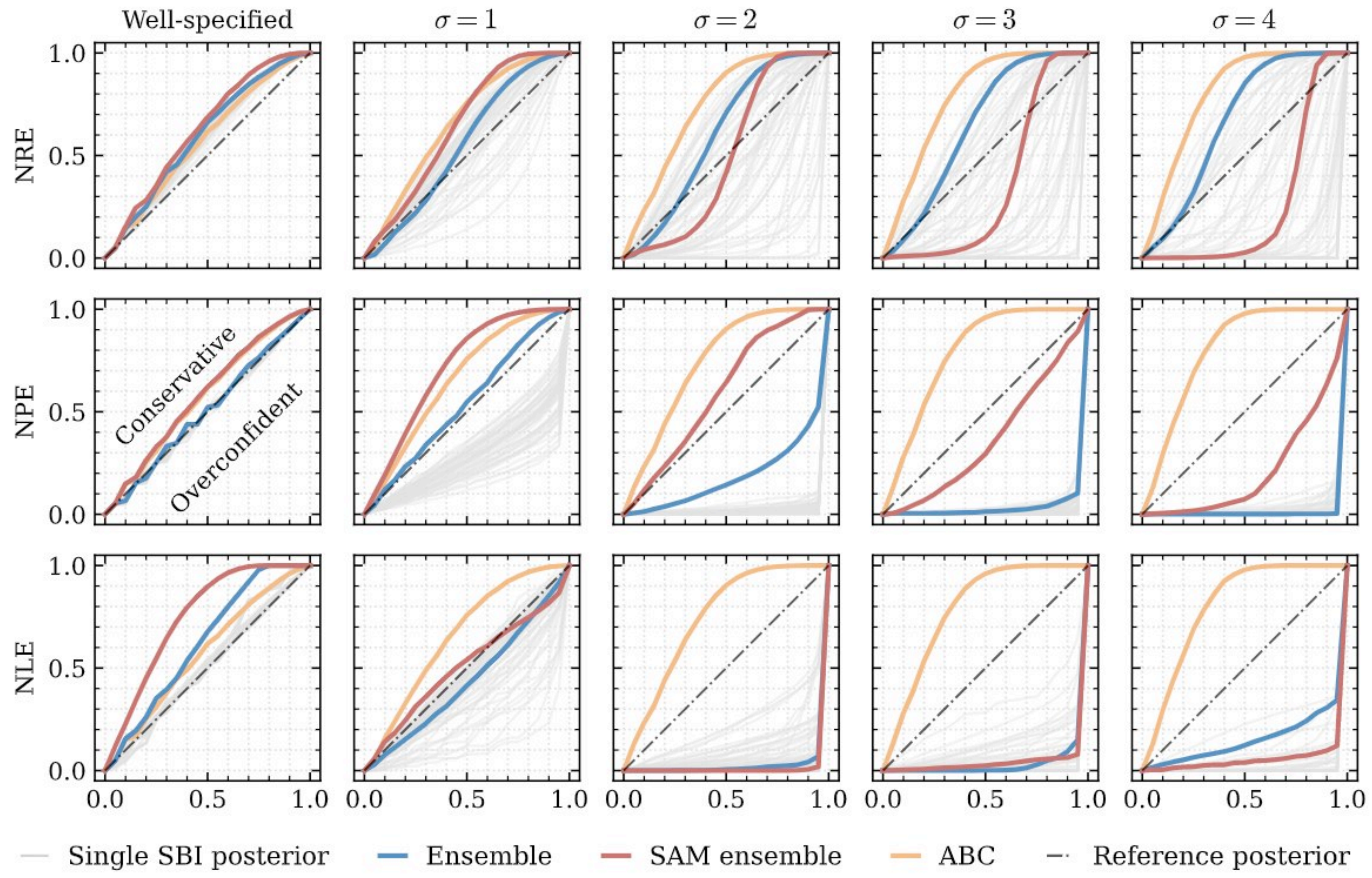
ONE LAST NOTE OF CAUTION!

Investigating the Impact of Model Misspecification in Neural Simulation-based Inference

Patrick Cannon*[1], Daniel Ward[2], and Sebastian M. Schmon[1,3]

[1] *Improbable, UK*
[2] *School of Mathematics, Bristol University, UK*
[3] *Department of Mathematical Sciences, Durham University, UK*

Gaussian model with wrong variance



Sebastian Schmon
@SeBayesian

What's the takeaway? SBI methods can perform very well when real data looks like simulated data. If not there is a danger of wild inaccuracy. Future work should look for methods to 1) identify and 2) counter misspecification.

12:31 PM · Sep 8, 2022 · Twitter Web App

ONE LAST NOTE OF CAUTION!

MODEL MISSPECIFICATION!

- Remember: SBI works by having a reliable simulator.

→ What if there is a distribution shift between simulated data and real data.

SOFTWARE & BENCHMARKS

SOFTWARE & BENCHMARKS

sbi

Search

mackelab/sbi
v0.19.2 325 74

sbi
- Home
- Installation
- Tutorials and Examples
- Contribute
- API Reference
- FAQ
- Credits

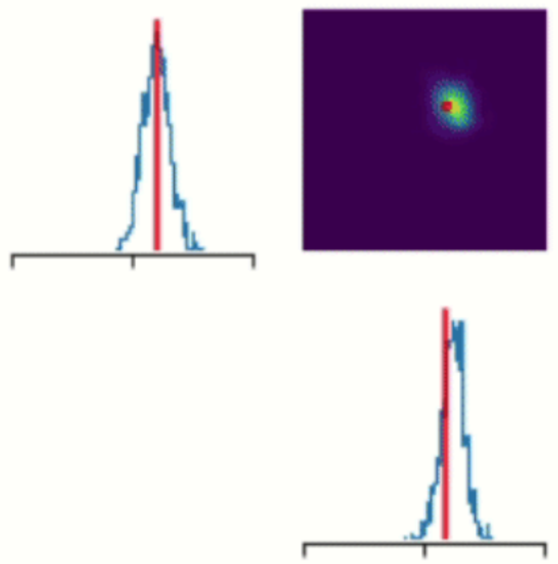
`sbi` : simulation-based inference

`sbi` : A Python toolbox for simulation-based inference.

```
prior = BoxUniform(low=zeros(2), high=2*ones(2))   # Box prior [0,2]x[0,2]
def simulator(θ): return θ + 0.1*randn_like(θ)     # Gaussian in 2D
posterior = infer(simulator, prior, method='SNPE', num_simulations=500)
```

```
Running 500 simulations.: 100%|██████████| 500/500 [00:00<00:00, 57141.55it/s]
Neural network successfully converged after 109 epochs.
```

```
samples = posterior.sample((1000,), x=observed)
pairplot(samples, points=ground_truth, **plot_style);
```



Inference can be run in a single line of code:

```
posterior = infer(simulator, prior, method='SNPE', num_simulations=1000)
```


https://www.mackelab.org/sbi

SOFTWARE & BENCHMARKS

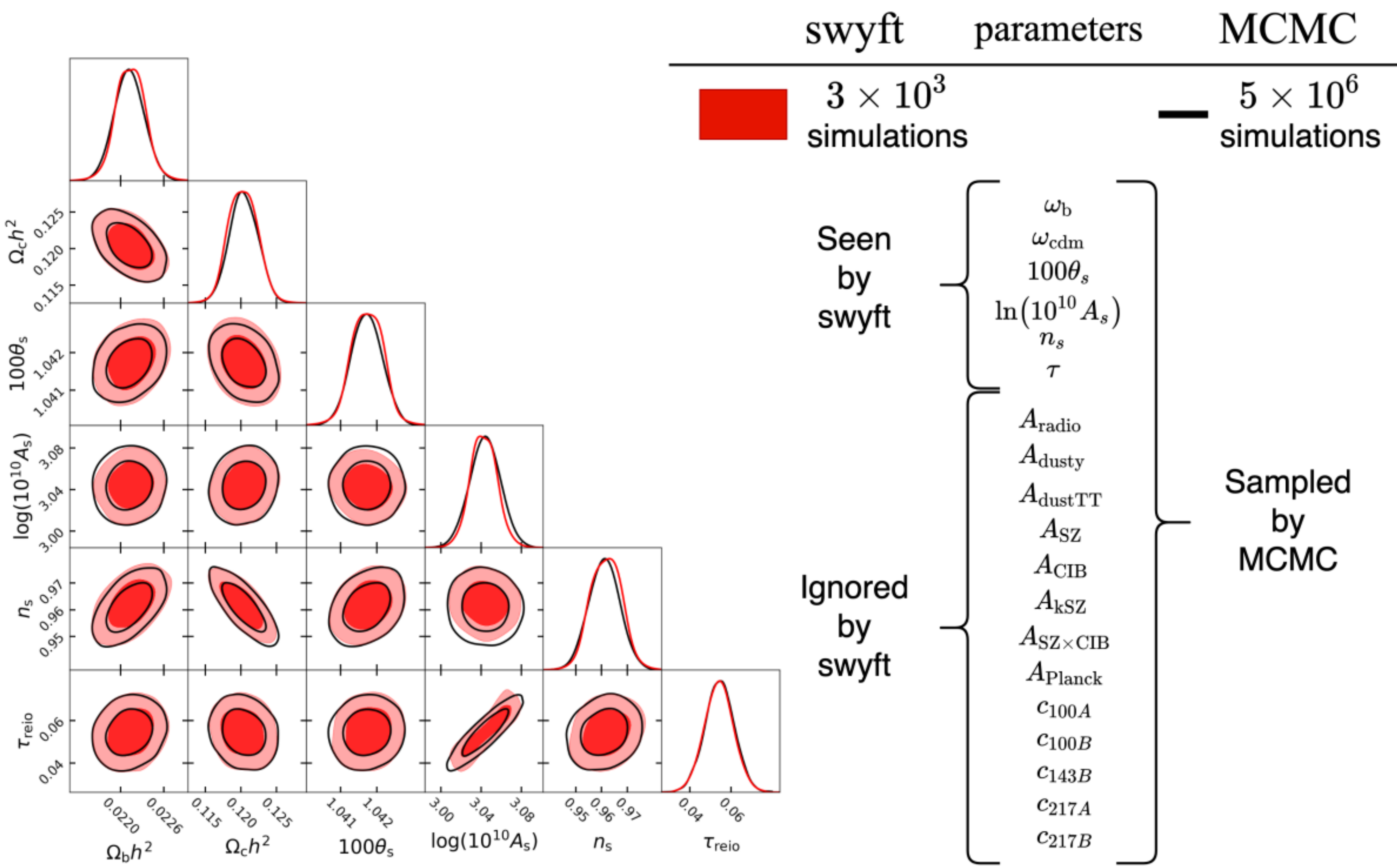
JOSS The Journal of Open Source Software

SWYFT Scalable Inference

swyft: Truncated Marginal Neural Ratio Estimation in Python

Benjamin Kurt Miller[1,2,3], **Alex Cole**[1], **Christoph Weniger**[1], **Francesco Nattino**[4], **Ou Ku**[4], **and Meiert W. Grootes**[4]

1 Gravitation Astroparticle Physics Amsterdam (GRAPPA), University of Amsterdam, Science Park 904, 1098 XH Amsterdam **2** Amsterdam Machine Learning Lab (AMLab), University of Amsterdam, Science Park 904, 1098 XH Amsterdam **3** AI4Science Lab, University of Amsterdam, Science Park 904, 1098 XH Amsterdam **4** Netherlands eScience Center, Science Park 140, 1098 XG Amsterdam, The Netherlands



- A python library built on pytorch/lightning
- "Official" implementation of **Truncated Marginal Neural Ratio Estimation (TMNRE)** algorithm
- Makes it simple to estimate marginal posteriors for very high dimensional models
- https://github.com/undark-lab/swyft

SOFTWARE & BENCHMARKS

LtU ILI

latest

Search docs

CONTENTS:

- Home
- Installation
- Contributing
- API

Welcome to LtU ILI's documentation!

Edit on GitHub

Welcome to LtU ILI's documentation!

Contents:

- Home
- Installation
- Contributing
- API

LtU-ILI

all contributors 12 | unit-tests passing | codecov 97% | docs passing

The **Learning the Universe Implicit Likelihood Inference (LtU-ILI)** pipeline is an all-in-one framework for performing machine learning parameter inference in astrophysics and cosmology. Given labeled training data ${(x_i,\theta_i)}_{i=1}^N$ or a stochastic simulator $x(\theta)$, LtU-ILI is designed to automatically train state-of-the-art neural networks to learn the data-parameter relationship and produce robust, well-calibrated posterior inference.

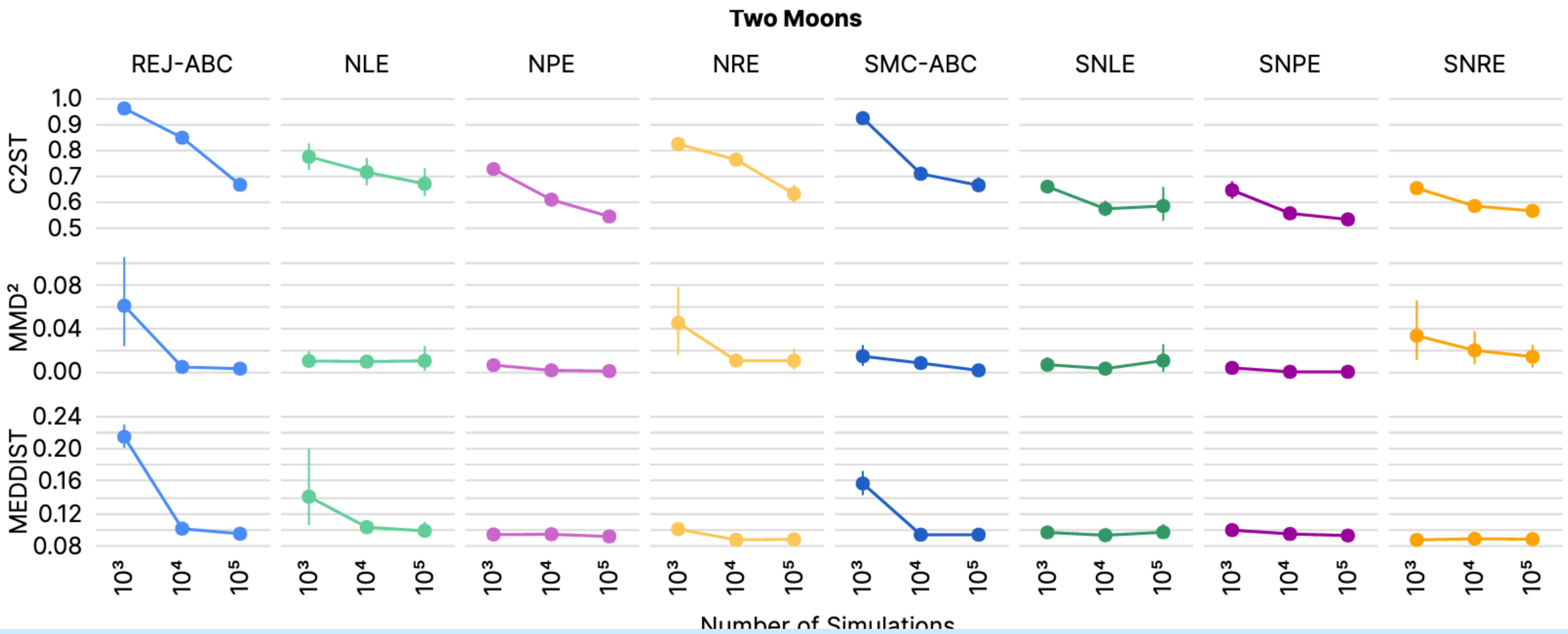
The pipeline is quick and easy to set up; here's an example of training a Masked Autoregressive Flow (MAF) network to predict a posterior over parameters y, given input data x:

SOFTWARE & BENCHMARKS

Benchmarking Simulation-Based Inference

Jan-Matthis Lueckmann[1,2] Jan Boelts[2] David S. Greenberg[2,3]
Pedro J. Gonçalves[4] Jakob H. Macke[1,2,5]

[1]University of Tübingen [2]Technical University of Munich [3]Helmholtz Centre Geesthacht
[4]Research Center caesar [5]Max Planck Institute for Intelligent Systems, Tübingen



SUMMARY & CONCLUSION

Simulation-based inference is making rapid progress with new deep learning algorithms.

Several routes: NPE, NLE, NRE, sequential/active methods....

Already available software implementations.

SUMMARY & CONCLUSION

Many cool applications of SBI haven’t been mentioned:
neuroscience, epidemiology, particle physics, ...

Ongoing work examines consistency, how modifications to vanilla algorithms can avoid mistakes, improving efficiency.

Together we can unlock the full scientific content of the data we measure!

WHERE TO FIND THE LECTURE MATERIAL

https://github.com/TobiBu/graddays


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS