

# Nested sampling: powering next-generation inference and machine learning tools for astrophysics, cosmology, particle physics and beyond

Will Handley

[wh260@cam.ac.uk](mailto:wh260@cam.ac.uk)

Royal Society University Research Fellow  
Astrophysics Group, Cavendish Laboratory, University of Cambridge  
Kavli Institute for Cosmology, Cambridge  
Gonville & Caius College  
[willhandley.co.uk/talks](http://willhandley.co.uk/talks)

30<sup>th</sup> August 2024



UNIVERSITY OF  
CAMBRIDGE

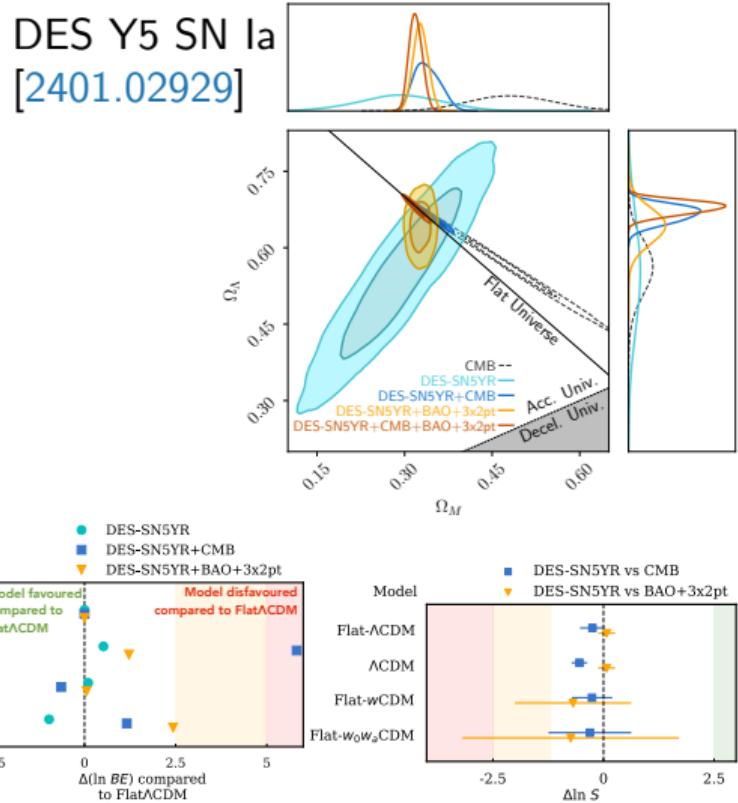


# LBI: Likelihood-based inference

The standard approach if you are fortunate enough to have a likelihood function  $P(D|\theta)$ :

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

1. Define prior  $\pi(\theta)$ 
  - ▶ spend some time being philosophical
2. Sample posterior  $P(\theta|D)$ 
  - ▶ use out-of-the-box MCMC tools such as emcee or MultiNest
  - ▶ make some triangle plots
3. Optionally compute evidence  $\mathcal{Z}(D)$ 
  - ▶ e.g. nested sampling or parallel tempering
  - ▶ do some model comparison (i.e. science)
  - ▶ talk about tensions



# LBI: Likelihood-based inference

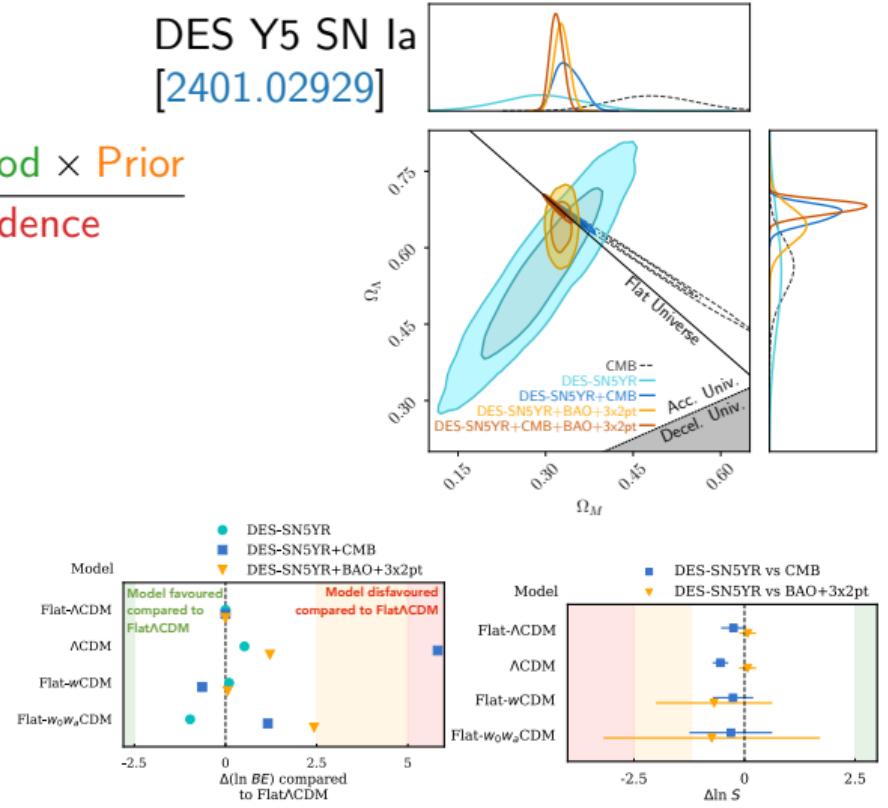
The standard approach if you are fortunate enough to have a likelihood function  $P(D|\theta)$ :

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

1. Define prior  $\pi(\theta)$ 
  - ▶ spend some time being philosophical
2. Sample posterior  $P(\theta|D)$ 
  - ▶ use out-of-the-box MCMC tools such as emcee or MultiNest
  - ▶ make some triangle plots
3. Optionally compute evidence  $\mathcal{Z}(D)$ 
  - ▶ e.g. nested sampling or parallel tempering
  - ▶ do some model comparison (i.e. science)
  - ▶ talk about tensions

DES Y5 SN Ia  
[2401.02929]



# LBI: Likelihood-based inference

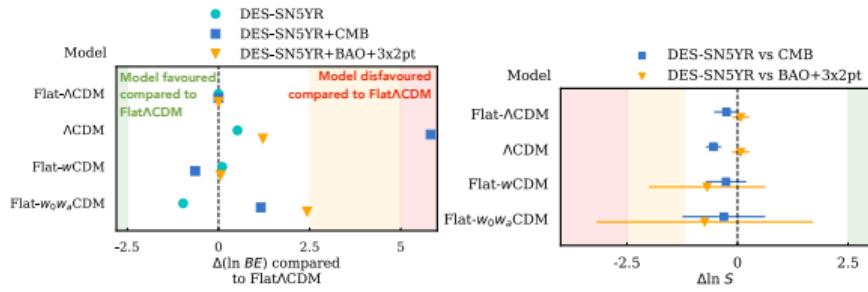
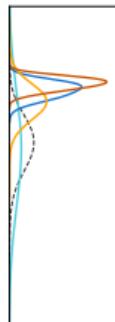
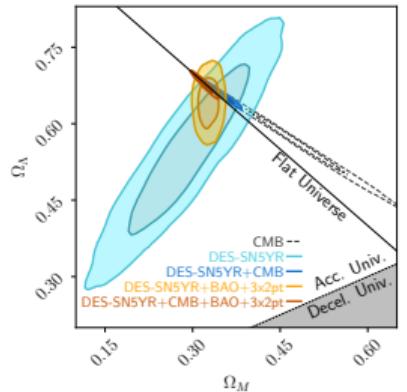
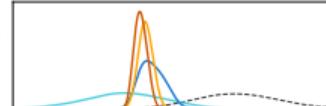
The standard approach if you are fortunate enough to have a likelihood function  $\mathcal{L}(D|\theta)$ :

$$\mathcal{P}(\theta|D) = \frac{\mathcal{L}(D|\theta)\pi(\theta)}{\mathcal{Z}(D)}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

1. Define prior  $\pi(\theta)$ 
  - ▶ spend some time being philosophical
2. Sample posterior  $\mathcal{P}(\theta|D)$ 
  - ▶ use out-of-the-box MCMC tools such as emcee or MultiNest
  - ▶ make some triangle plots
3. Optionally compute evidence  $\mathcal{Z}(D)$ 
  - ▶ e.g. nested sampling or parallel tempering
  - ▶ do some model comparison (i.e. science)
  - ▶ talk about tensions

DES Y5 SN Ia  
[2401.02929]

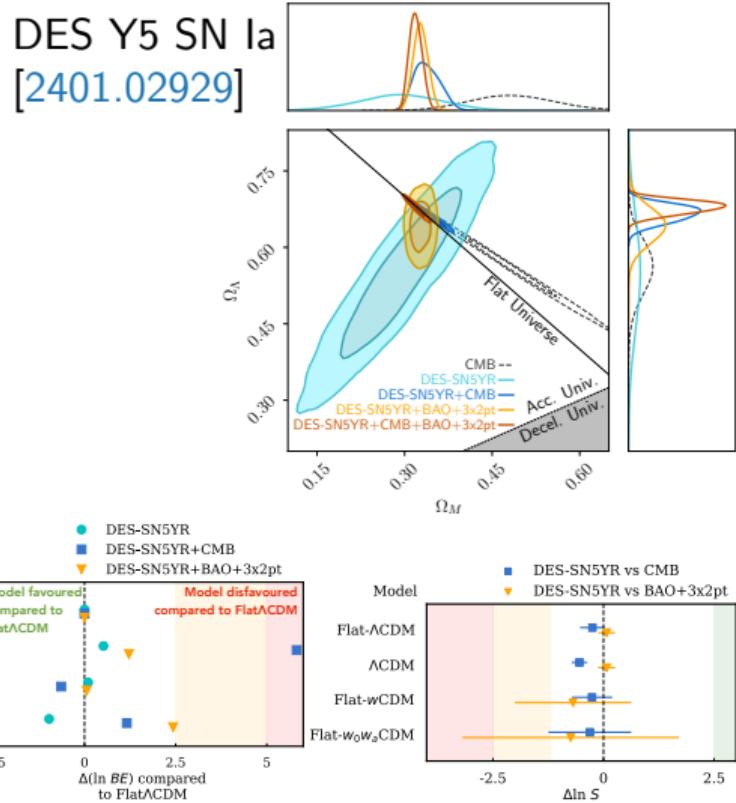


# LBI: Likelihood-based inference

The standard approach if you are fortunate enough to have a likelihood function  $\mathcal{L}(D|\theta)$ :

$$P(\theta|D)P(D) = P(\theta, D) = P(D|\theta)P(\theta),$$

1. Define prior  $\pi(\theta)$ 
  - ▶ spend some time being philosophical
2. Sample posterior  $P(\theta|D)$ 
  - ▶ use out-of-the-box MCMC tools such as emcee or MultiNest
  - ▶ make some triangle plots
3. Optionally compute evidence  $\mathcal{Z}(D)$ 
  - ▶ e.g. nested sampling or parallel tempering
  - ▶ do some model comparison (i.e. science)
  - ▶ talk about tensions

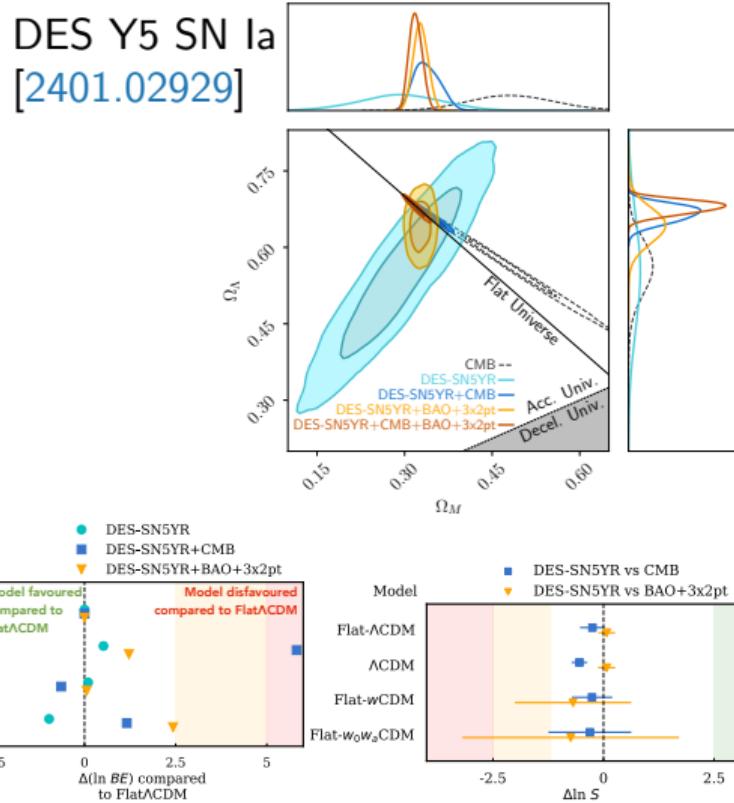


# LBI: Likelihood-based inference

The standard approach if you are fortunate enough to have a likelihood function  $\mathcal{L}(D|\theta)$ :

$$\mathcal{P} \times \mathcal{Z} = \mathcal{J} = \mathcal{L} \times \pi, \quad \text{Joint} = \mathcal{J} = P(\theta, D)$$

1. Define prior  $\pi(\theta)$ 
  - ▶ spend some time being philosophical
2. Sample posterior  $P(\theta|D)$ 
  - ▶ use out-of-the-box MCMC tools such as emcee or MultiNest
  - ▶ make some triangle plots
3. Optionally compute evidence  $\mathcal{Z}(D)$ 
  - ▶ e.g. nested sampling or parallel tempering
  - ▶ do some model comparison (i.e. science)
  - ▶ talk about tensions



# The three pillars of Bayesian inference

## Parameter estimation

What do the data tell us about the parameters of a model?  
e.g. the size or age of a  $\Lambda$ CDM universe

$$P(\theta|D, M) = \frac{P(D|\theta, M)P(\theta|M)}{P(D|M)}$$

$$\mathcal{P} = \frac{\mathcal{L} \times \pi}{\mathcal{Z}}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

## Model comparison

How much does the data support a particular model?  
e.g.  $\Lambda$ CDM vs a dynamic dark energy cosmology

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

$$\frac{\mathcal{Z}_M \Pi_M}{\sum_m \mathcal{Z}_m \Pi_m}$$

$$\text{Posterior} = \frac{\text{Evidence} \times \text{Prior}}{\text{Normalisation}}$$

## Tension quantification

Do different datasets make consistent predictions from the same model? e.g. CMB vs Type IA supernovae data

$$\mathcal{R} = \frac{\mathcal{Z}_{AB}}{\mathcal{Z}_A \mathcal{Z}_B}$$

$$\begin{aligned} \log \mathcal{S} &= \langle \log \mathcal{L}_{AB} \rangle_{\mathcal{P}_{AB}} \\ &\quad - \langle \log \mathcal{L}_A \rangle_{\mathcal{P}_A} \\ &\quad - \langle \log \mathcal{L}_B \rangle_{\mathcal{P}_B} \end{aligned}$$

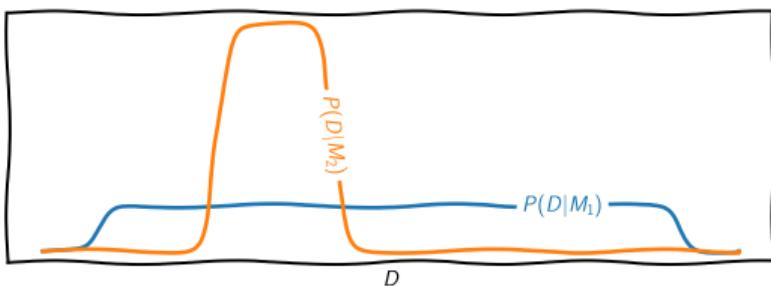
# Model comparison $\mathcal{Z} = P(D|M)$

- ▶ Bayesian model comparison allows mathematical derivation of key philosophical principles.

Viewed from data-space  $D$ :

## Popper's falsificationism

- ▶ Prefer models that make bold predictions.
- ▶ if proven true, model more likely correct.



- ▶ Falsificationism comes from normalisation

Viewed from parameter-space  $\theta$ :

## Occam's razor

- ▶ Models should be as simple as possible
- ▶ ... but no simpler

- ▶ Occam's razor equation:

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{KL}$$

- ▶ “Occam penalty”: KL divergence between prior  $\pi$  and posterior  $\mathcal{P}$ .

$$\mathcal{D}_{KL} \sim \log \frac{\text{Prior volume}}{\text{Posterior volume}}$$

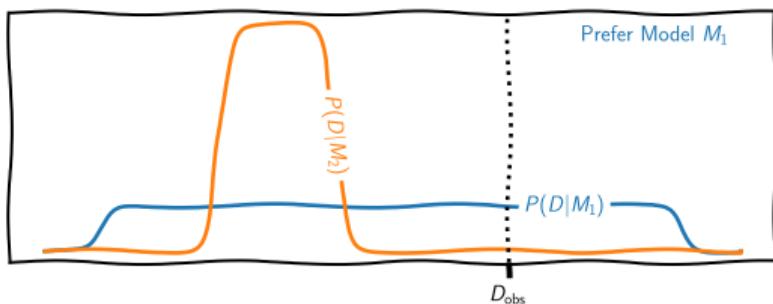
# Model comparison $\mathcal{Z} = P(D|M)$

- ▶ Bayesian model comparison allows mathematical derivation of key philosophical principles.

Viewed from data-space  $D$ :

## Popper's falsificationism

- ▶ Prefer models that make bold predictions.
- ▶ if proven true, model more likely correct.



- ▶ Falsificationism comes from normalisation

Viewed from parameter-space  $\theta$ :

## Occam's razor

- ▶ Models should be as simple as possible
- ▶ ... but no simpler

- ▶ Occam's razor equation:

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\text{KL}}$$

- ▶ “Occam penalty”: KL divergence between prior  $\pi$  and posterior  $\mathcal{P}$ .

$$\mathcal{D}_{\text{KL}} \sim \log \frac{\text{Prior volume}}{\text{Posterior volume}}$$

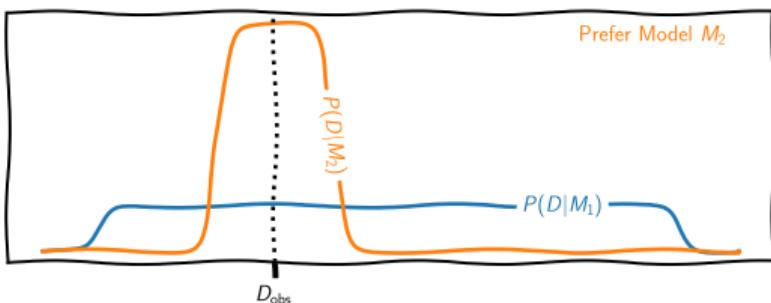
# Model comparison $\mathcal{Z} = P(D|M)$

- ▶ Bayesian model comparison allows mathematical derivation of key philosophical principles.

Viewed from data-space  $D$ :

## Popper's falsificationism

- ▶ Prefer models that make bold predictions.
- ▶ if proven true, model more likely correct.



- ▶ Falsificationism comes from normalisation

Viewed from parameter-space  $\theta$ :

## Occam's razor

- ▶ Models should be as simple as possible
- ▶ ... but no simpler

- ▶ Occam's razor equation:

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\text{KL}}$$

- ▶ “Occam penalty”: KL divergence between prior  $\pi$  and posterior  $\mathcal{P}$ .

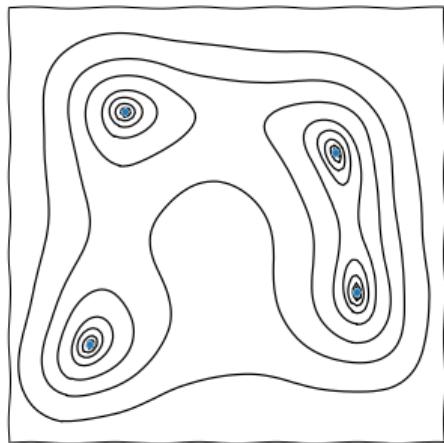
$$\mathcal{D}_{\text{KL}} \sim \log \frac{\text{Prior volume}}{\text{Posterior volume}}$$

# What is Nested Sampling?

- ▶ Nested sampling is a radical, multi-purpose numerical tool.
- ▶ Given a (scalar) function  $f$  with a vector of parameters  $\theta$ , it can be used for:

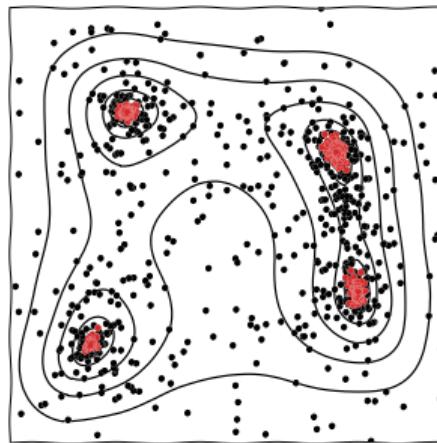
## Optimisation

$$\theta_{\max} = \max_{\theta} f(\theta)$$



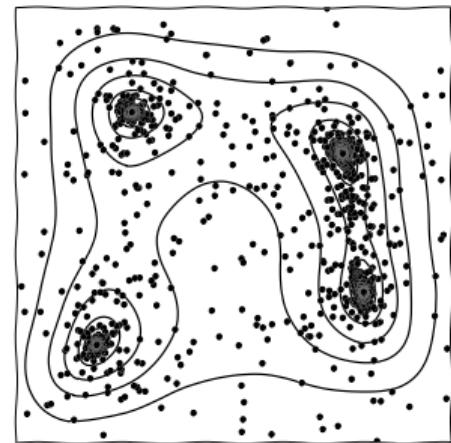
## Exploration

draw/sample  $\theta \sim f$



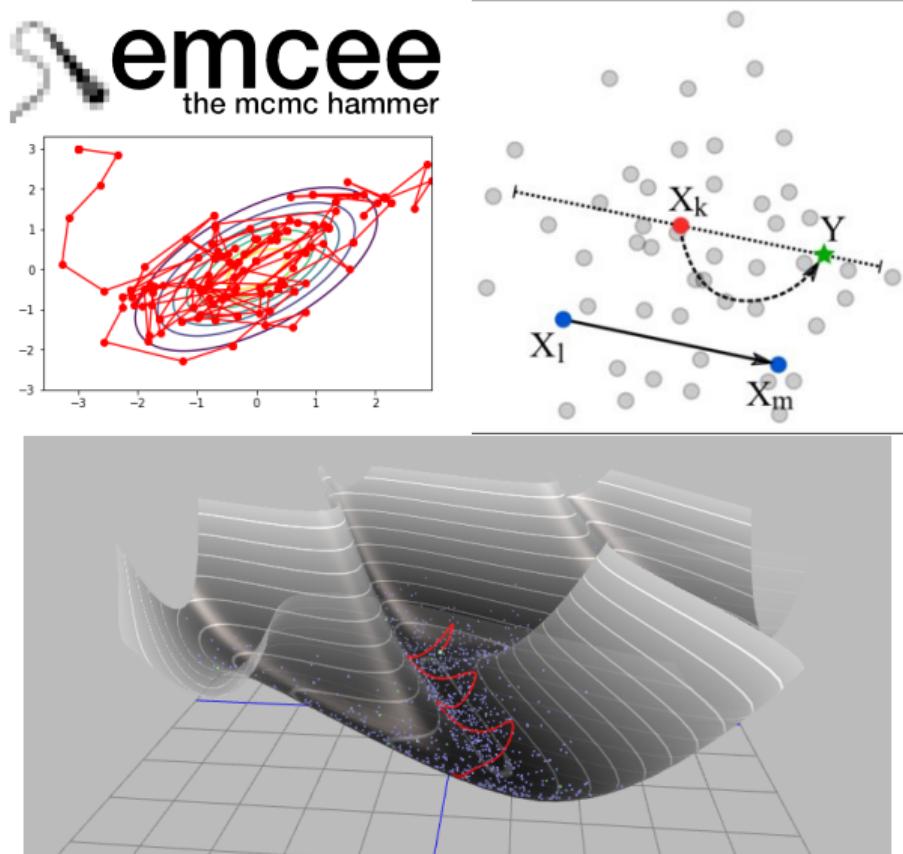
## Integration

$$\int f(\theta) dV$$



# Where is Nested Sampling?

- ▶ For many purposes, in your Neural Net you should group Nested Sampling with (MCMC) techniques such as:
  - ▶ Metropolis-Hastings (PyMC, MontePython)
  - ▶ Hamiltonian Monte Carlo (Stan, blackjax)
  - ▶ Ensemble sampling (emcee, zeus).
  - ▶ Variational Inference (Pyro)
  - ▶ Sequential Monte Carlo
  - ▶ Thermodynamic integration
  - ▶ Genetic algorithms
- ▶ You may have heard of it branded form:
  - ▶ MultiNest
  - ▶ PolyChord
  - ▶ dynesty
  - ▶ ultranest



# Integration in Physics

- ▶ Integration is a fundamental concept in physics, statistics and data science:

## Partition functions

$$Z(\beta) = \int e^{-\beta H(q,p)} dq dp$$

## Path integrals

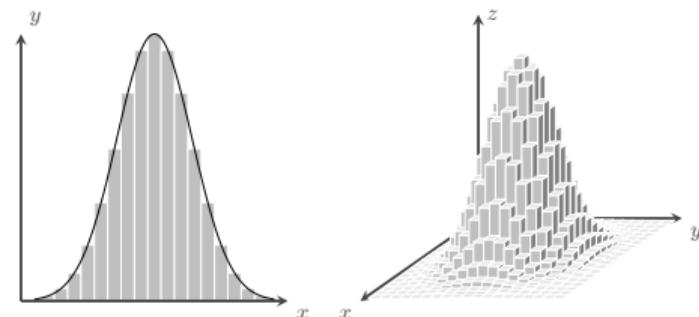
$$\Psi = \int e^{iS} \mathcal{D}x$$

## Bayesian marginals

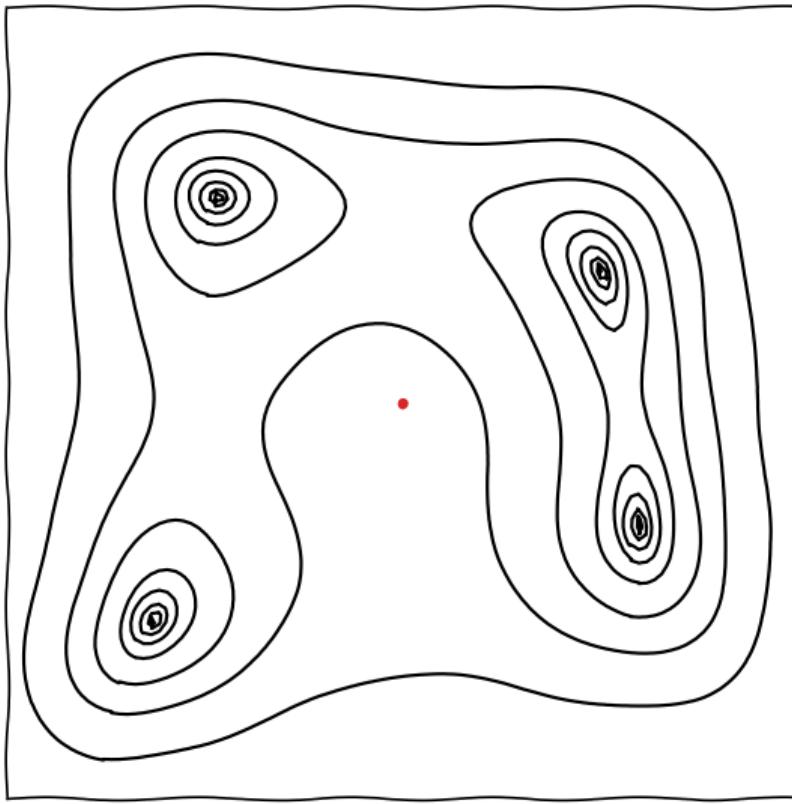
$$\mathcal{Z}(D) = \int \mathcal{L}(D|\theta) \pi(\theta) d\theta$$

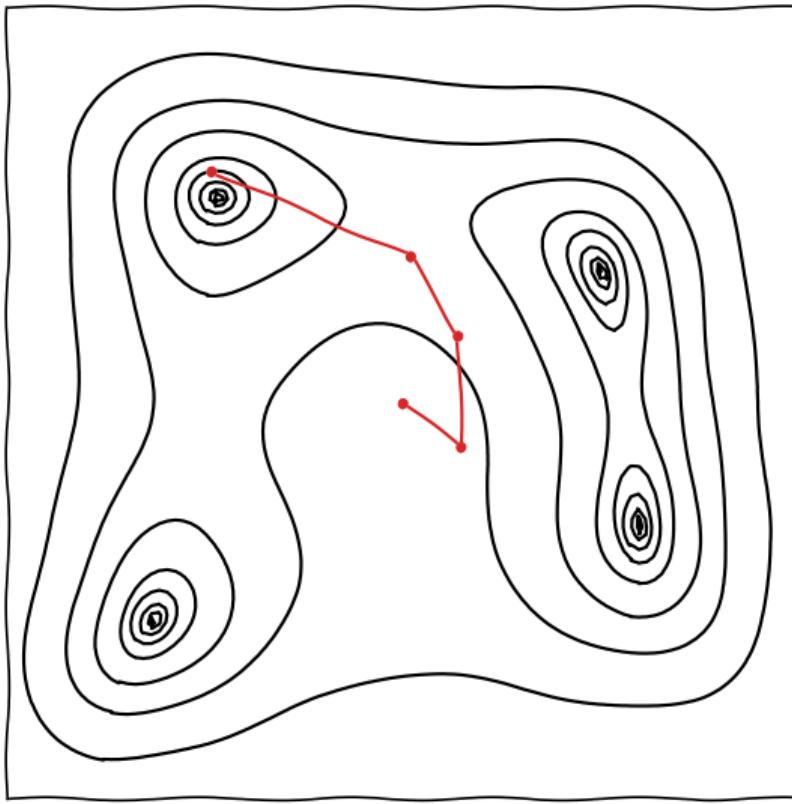
- ▶ Need numerical tools if analytic solution unavailable.
- ▶ High-dimensional numerical integration is hard.
- ▶ Riemannian strategy estimates volumes geometrically:

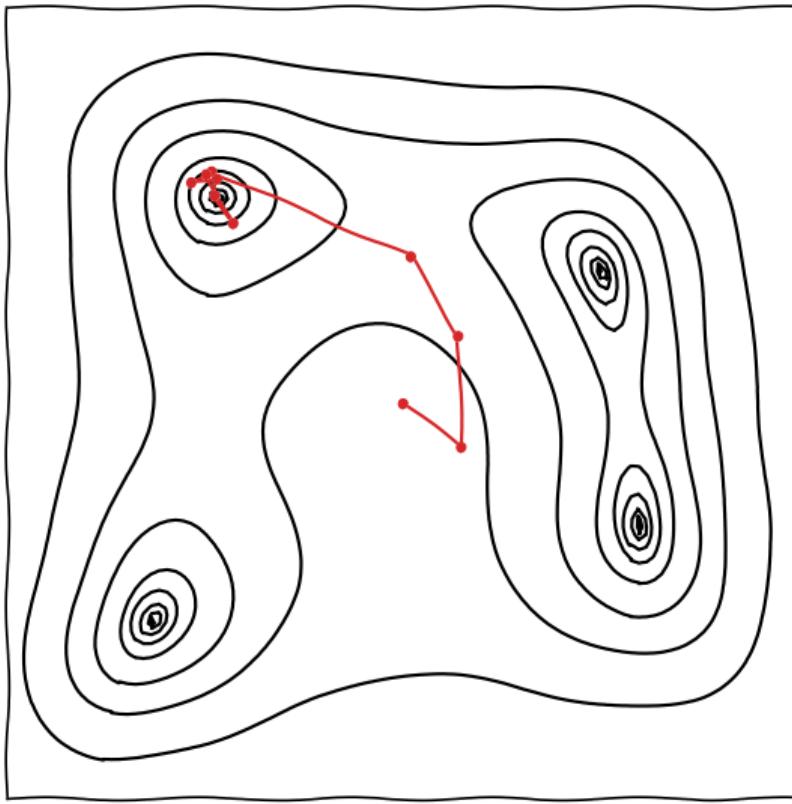
$$\int f(x) d^n x \approx \sum_i f(x_i) \Delta V_i \sim \mathcal{O}(e^n)$$

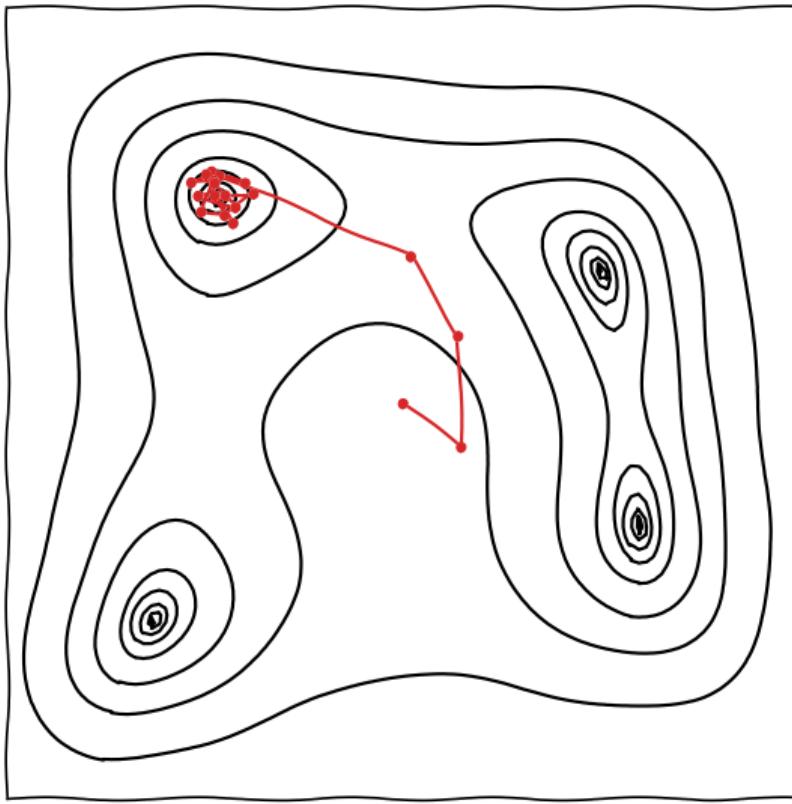


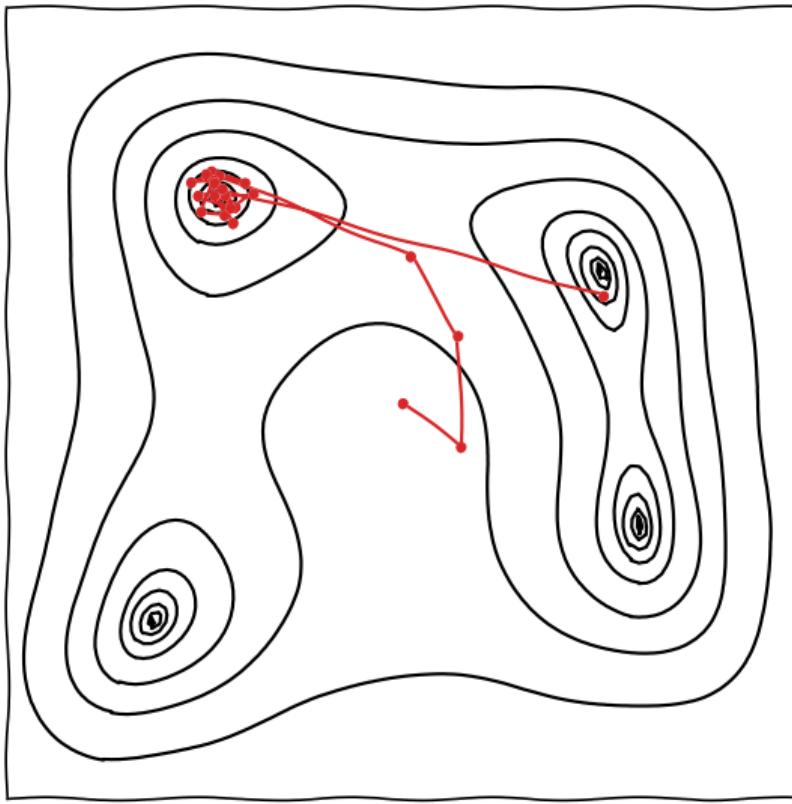
- ▶ Curse of dimensionality  $\Rightarrow$  exponential scaling.

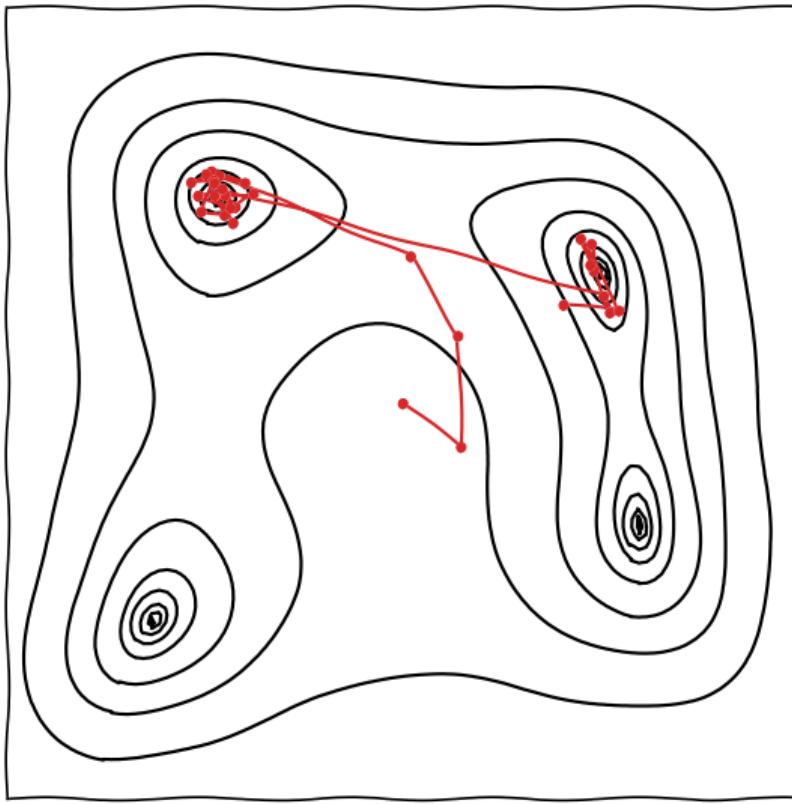




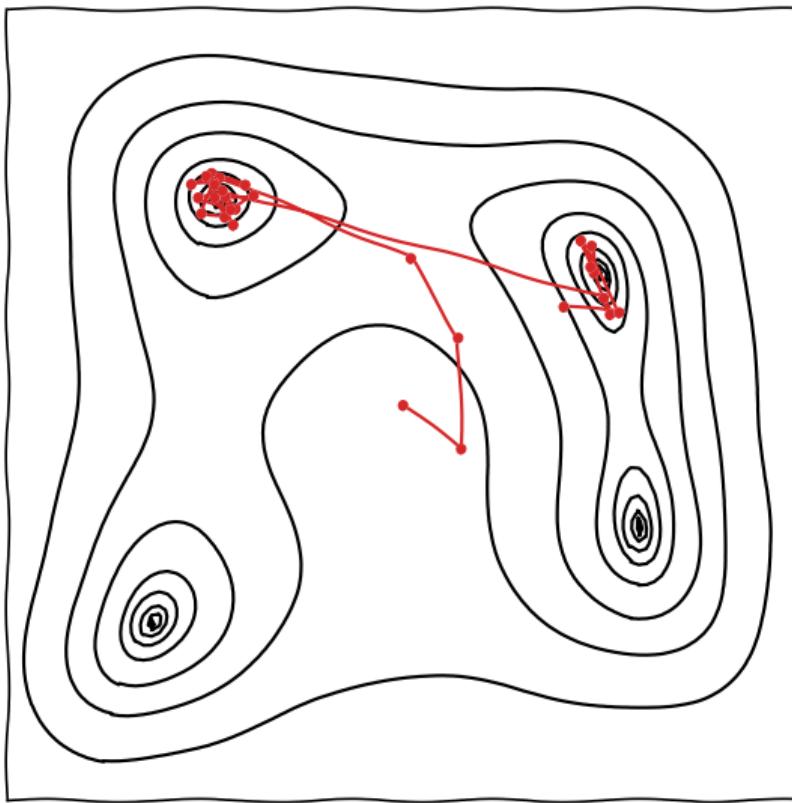




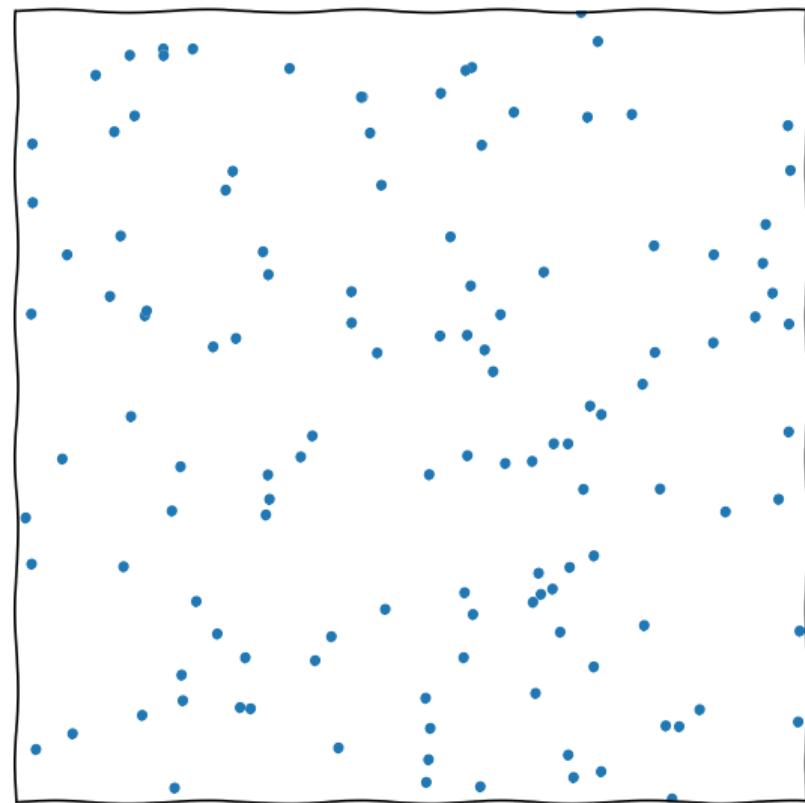




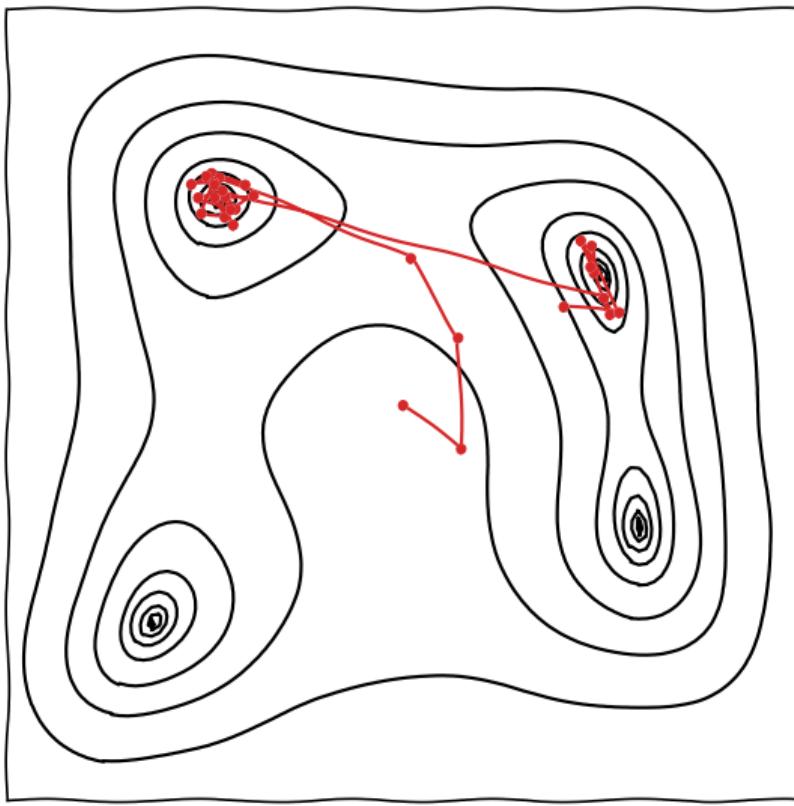
## MCMC



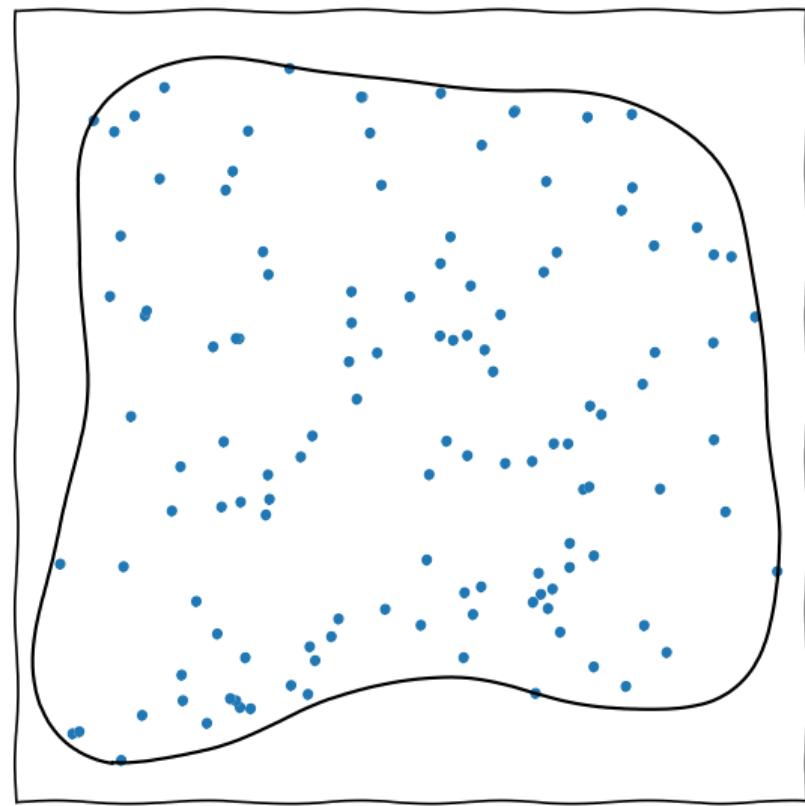
## Nested sampling



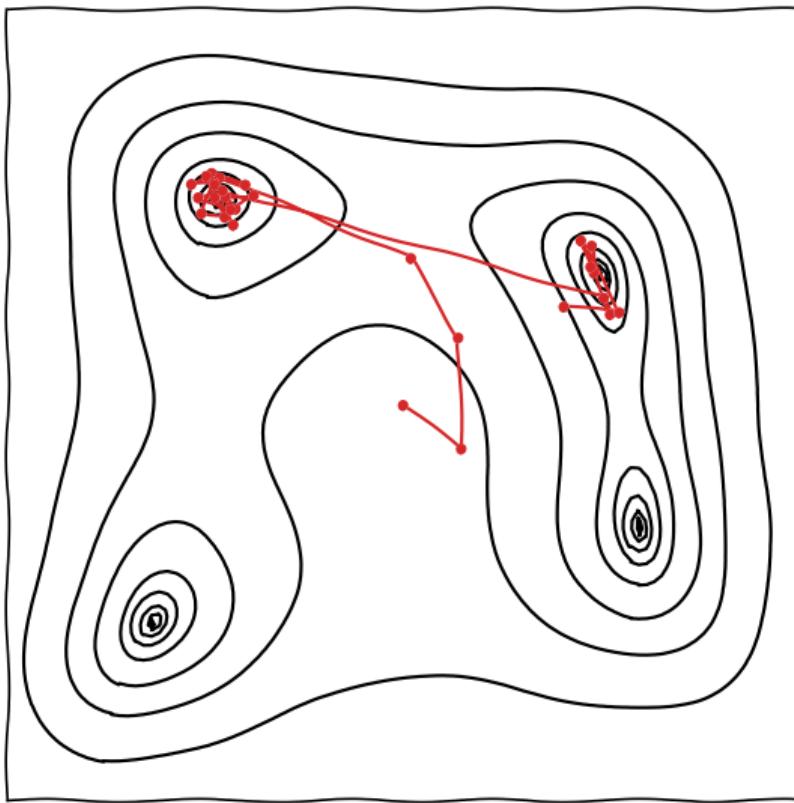
## MCMC



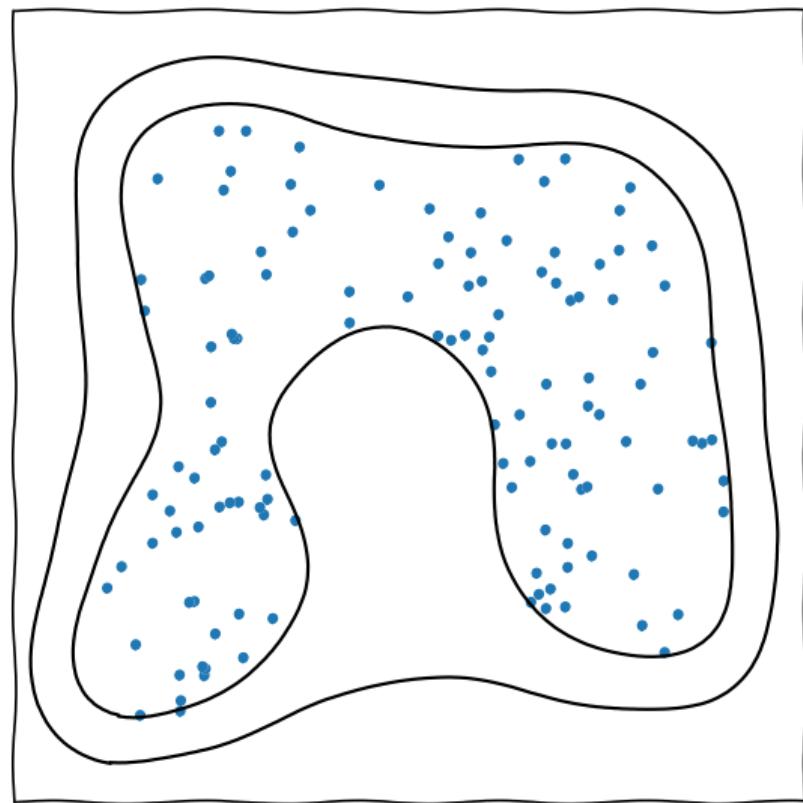
## Nested sampling



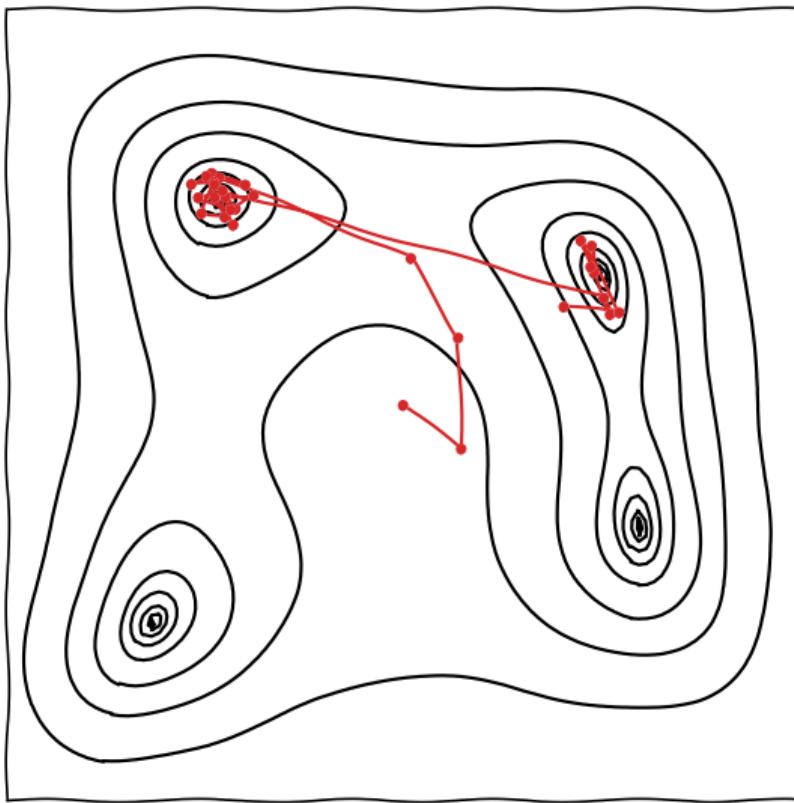
## MCMC



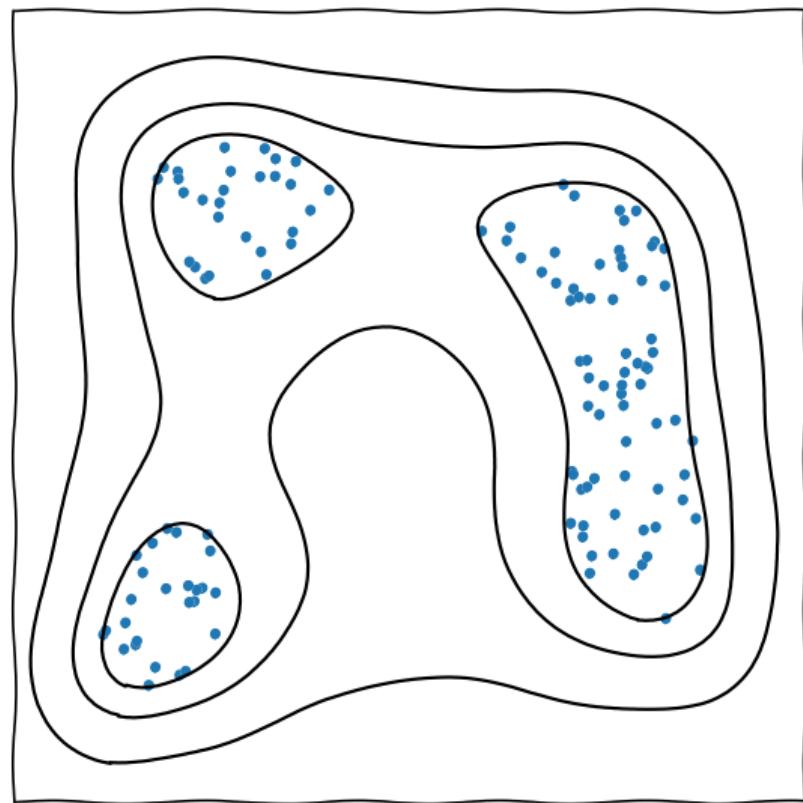
## Nested sampling



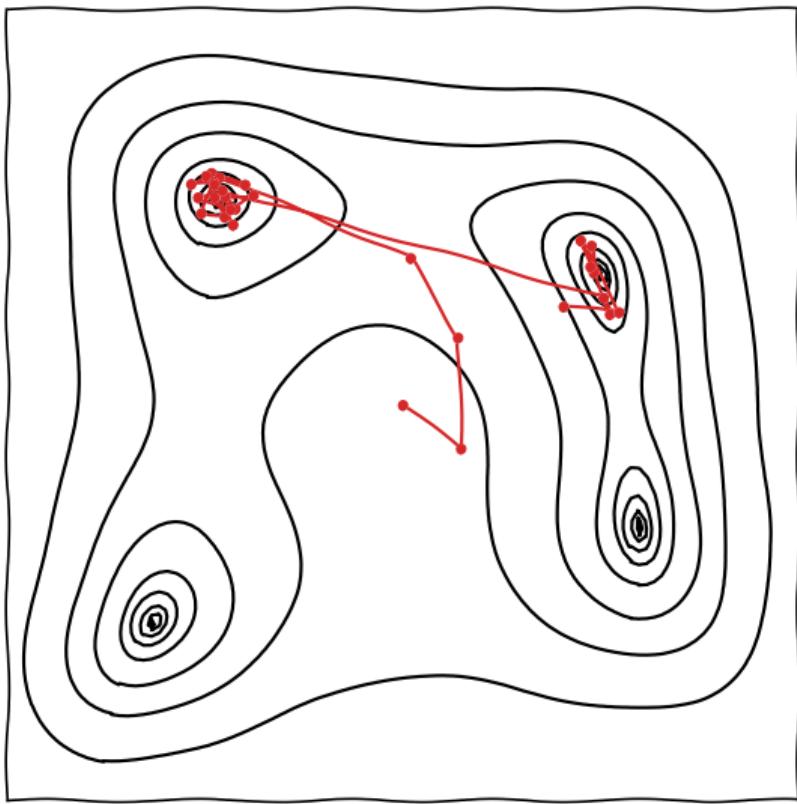
## MCMC



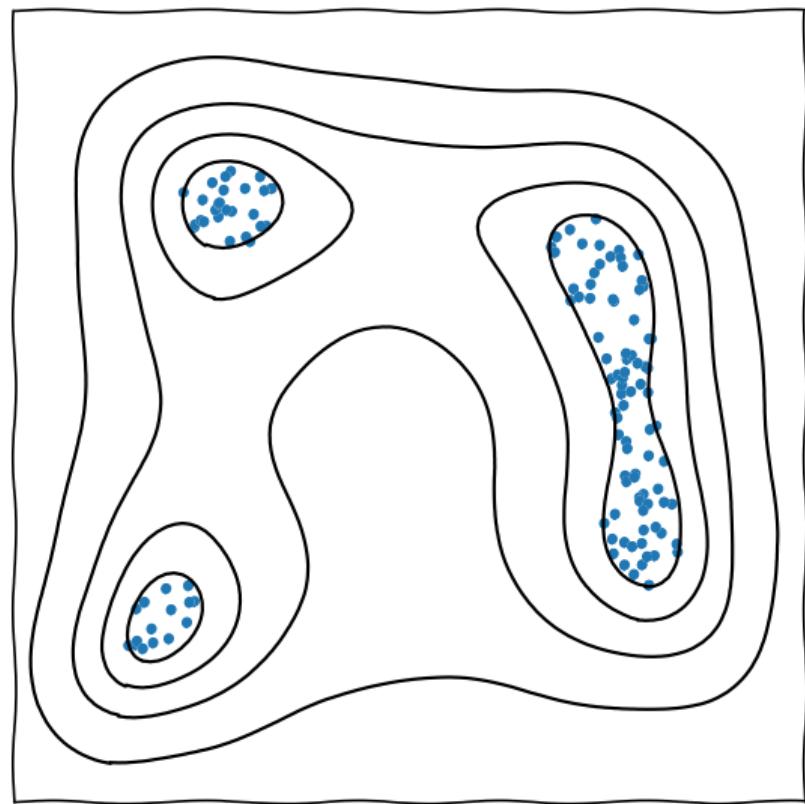
## Nested sampling



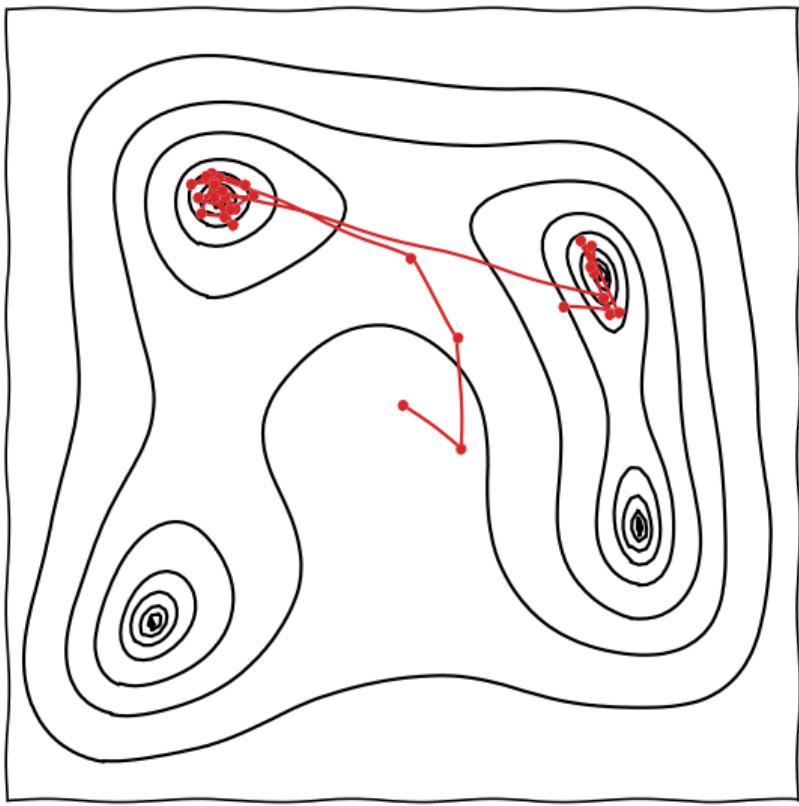
## MCMC



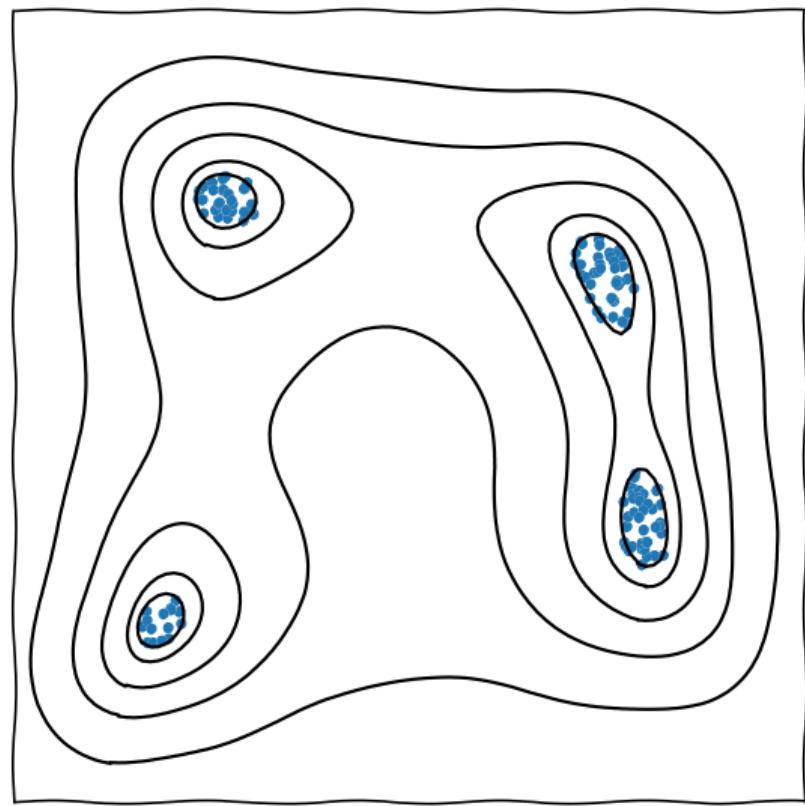
## Nested sampling



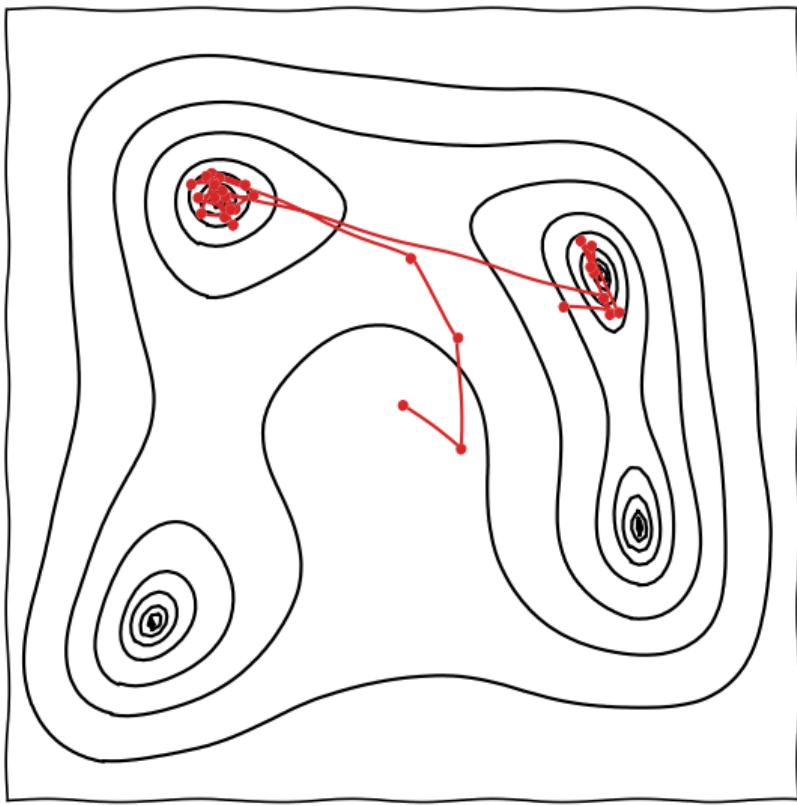
## MCMC



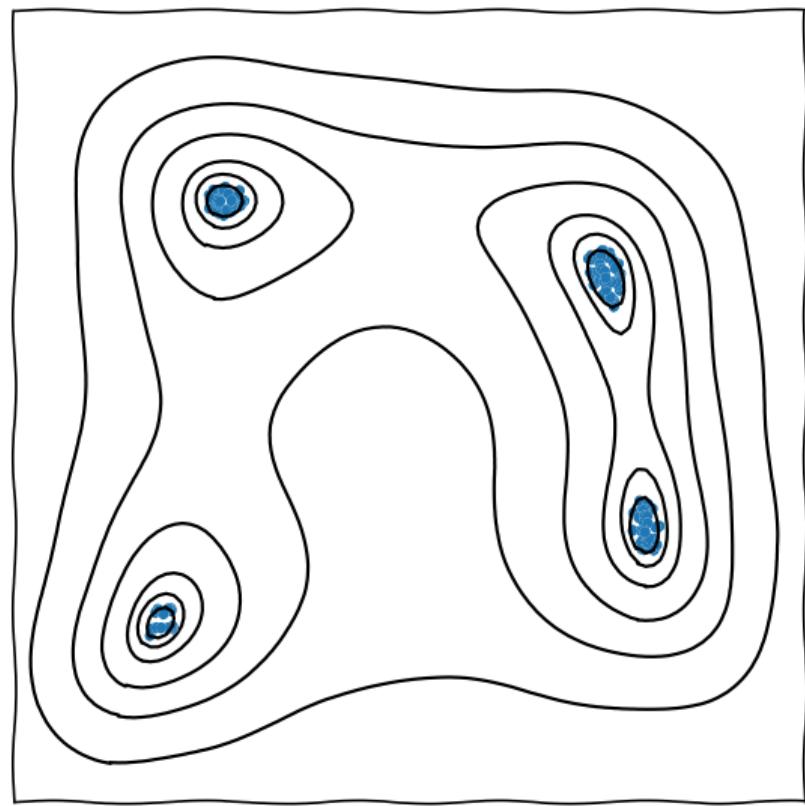
## Nested sampling



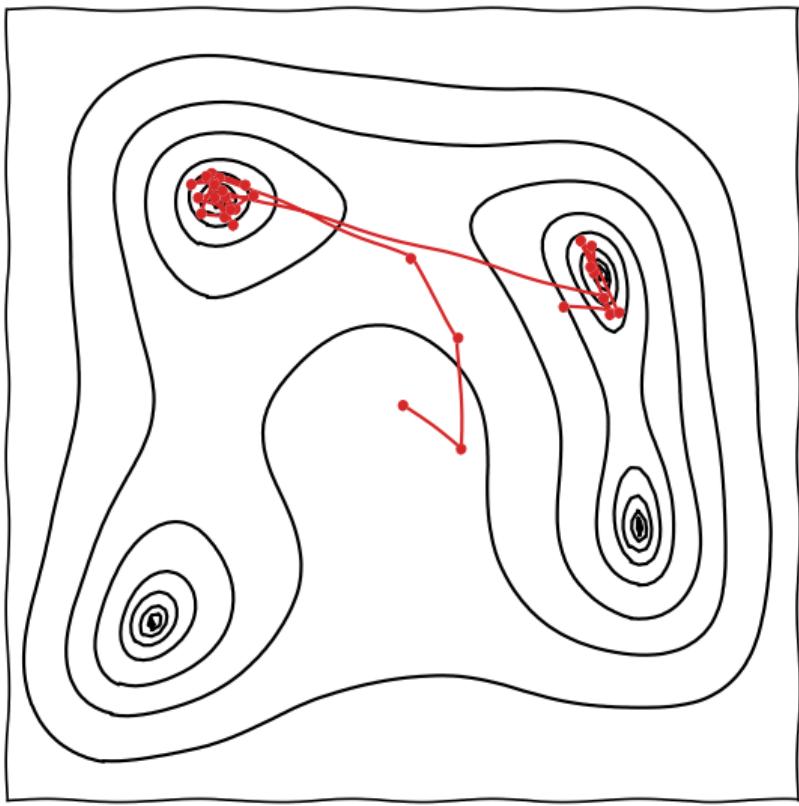
## MCMC



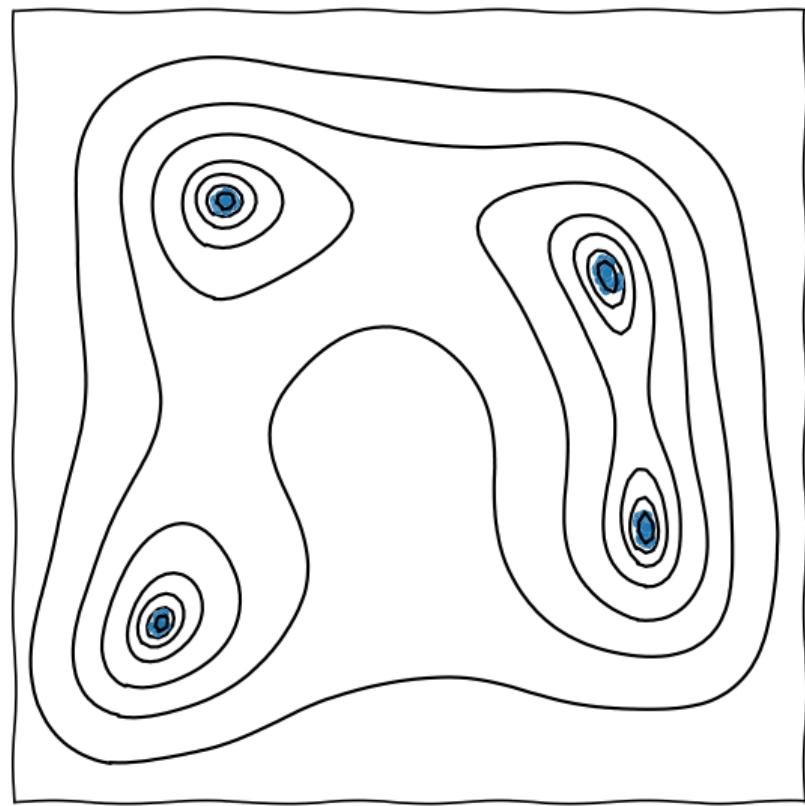
## Nested sampling



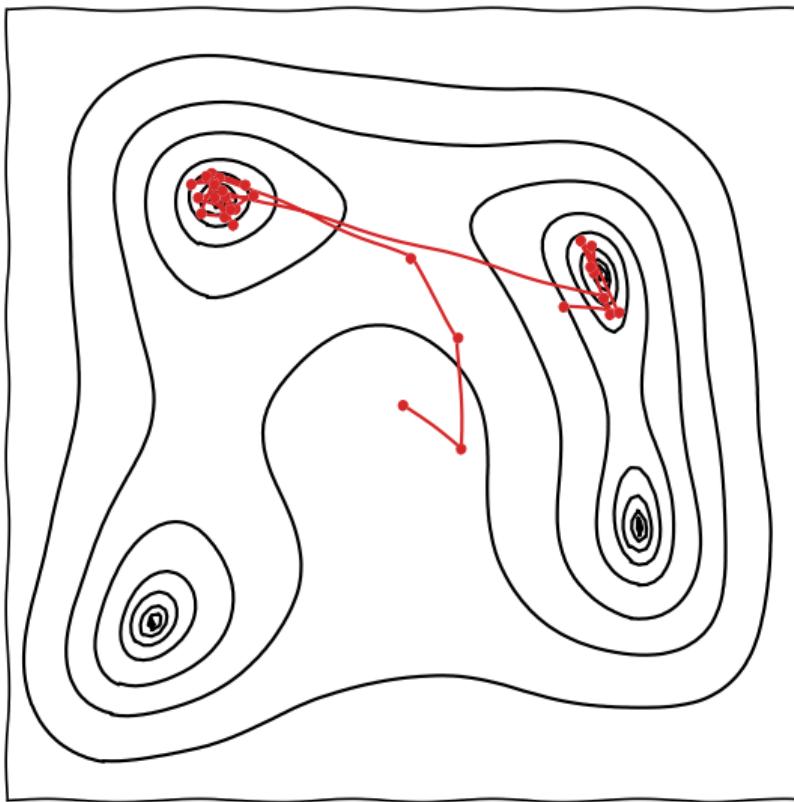
## MCMC



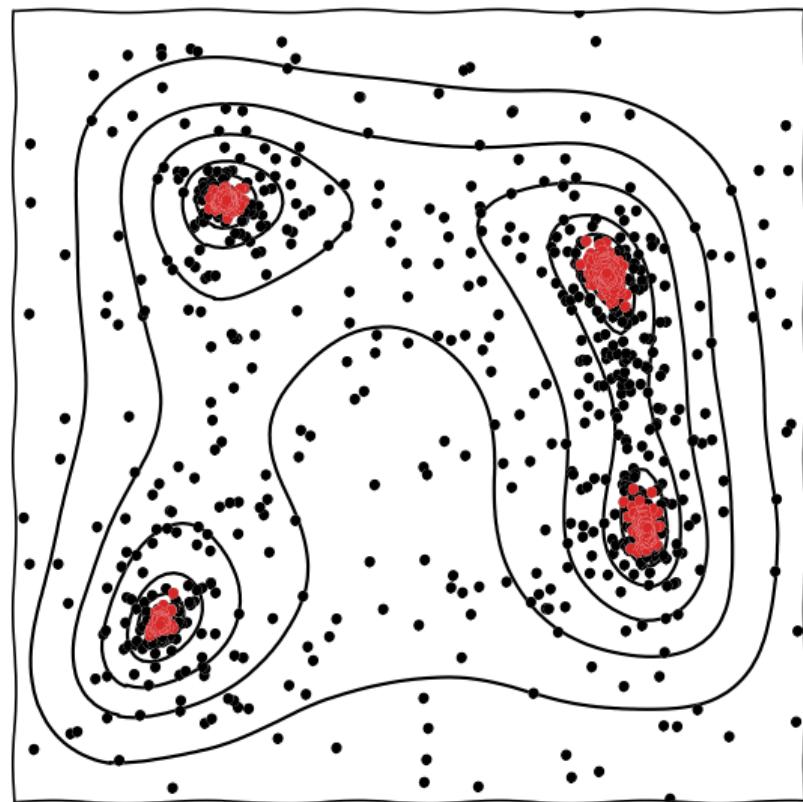
## Nested sampling



## MCMC

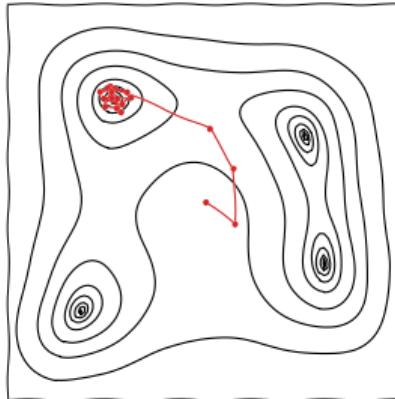


## Nested sampling



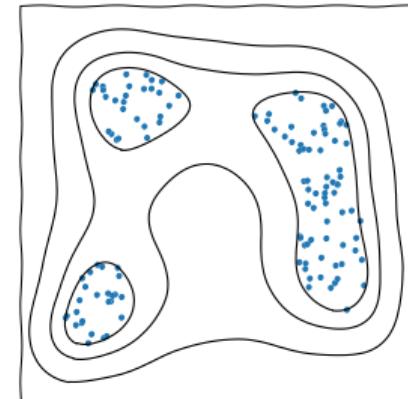
## MCMC

- ▶ Single “walker”
- ▶ Explores posterior
- ▶ Fast, if proposal matrix is tuned
- ▶ Parameter estimation, suspiciousness calculation
- ▶ Channel capacity optimised for generating posterior samples



## Nested sampling

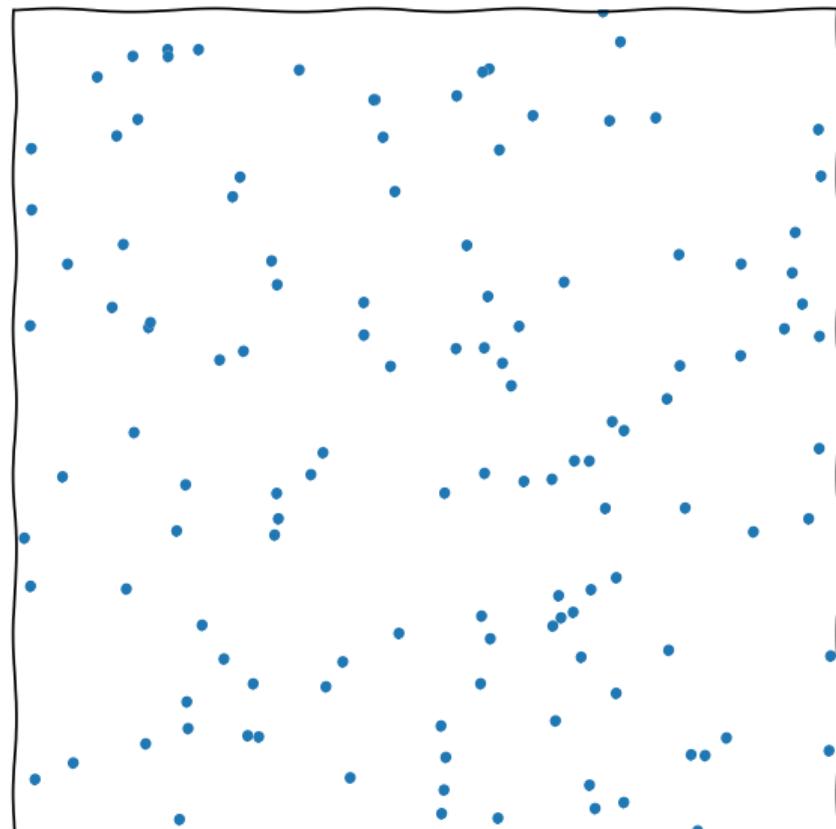
- ▶ Ensemble of “live points”
- ▶ Scans from prior to peak of likelihood
- ▶ Slower, no tuning required
- ▶ Parameter estimation, model comparison, tension quantification
- ▶ Channel capacity optimised for computing partition function



## The nested sampling meta-algorithm: live points

- ▶ Start with  $n$  random samples over the space.
- ▶ Delete outermost sample, and replace with a new random one at higher integrand value.
- ▶ The “live points” steadily contract around the peak(s) of the function.
- ▶ We can use this evolution to estimate volume *probabilistically*.
- ▶ At each iteration, the contours contract by  $\sim \frac{1}{n}$  of their volume.
- ▶ This is an exponential contraction, so

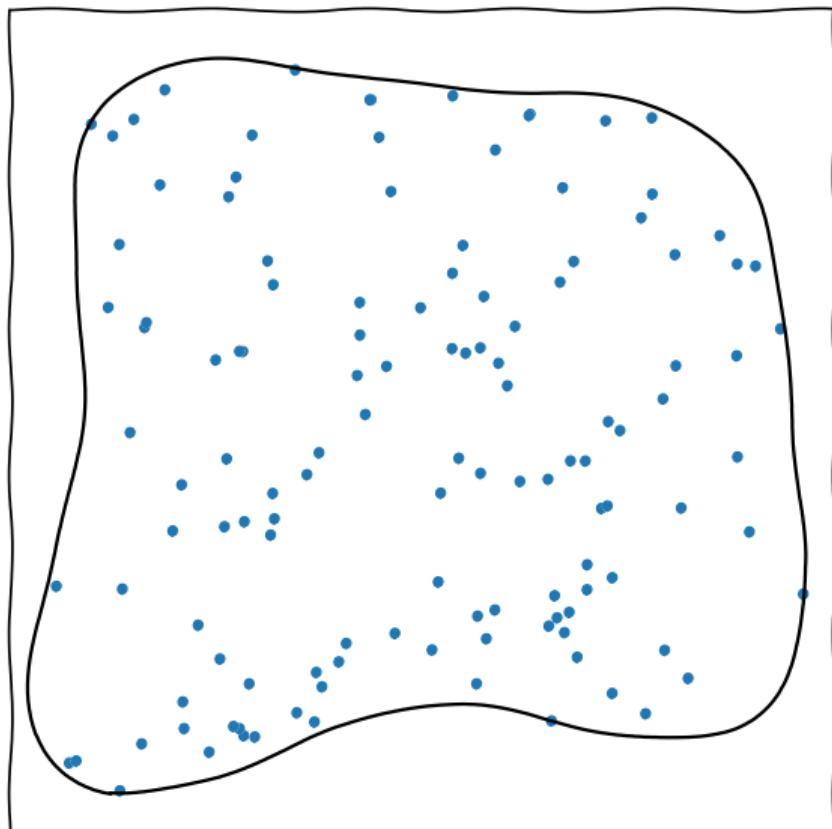
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



## The nested sampling meta-algorithm: live points

- ▶ Start with  $n$  random samples over the space.
- ▶ Delete outermost sample, and replace with a new random one at higher integrand value.
- ▶ The “live points” steadily contract around the peak(s) of the function.
- ▶ We can use this evolution to estimate volume *probabilistically*.
- ▶ At each iteration, the contours contract by  $\sim \frac{1}{n}$  of their volume.
- ▶ This is an exponential contraction, so

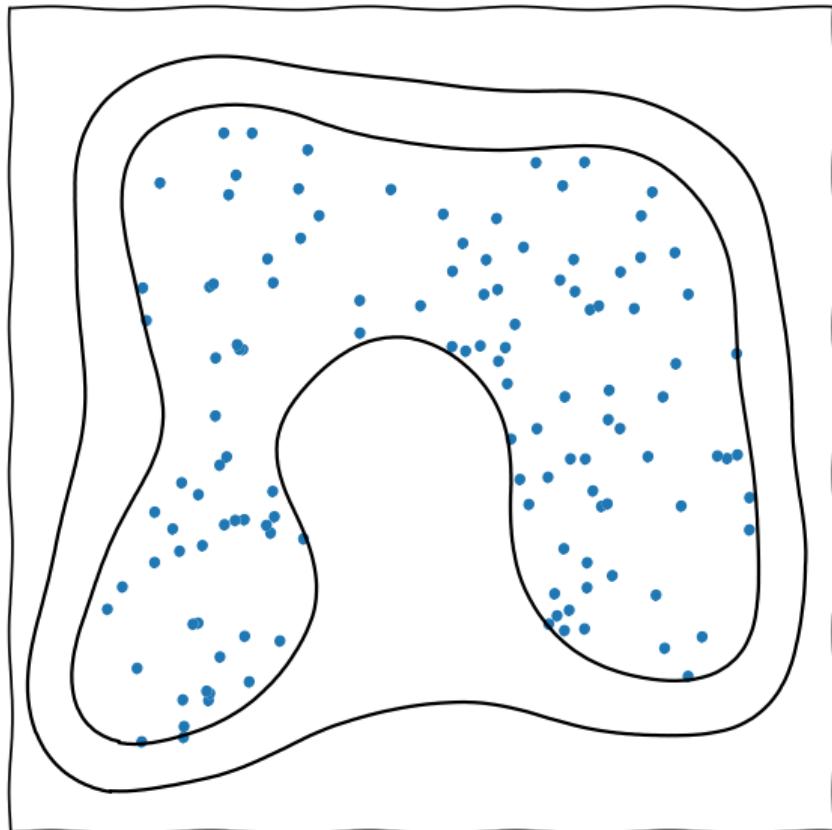
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



## The nested sampling meta-algorithm: live points

- ▶ Start with  $n$  random samples over the space.
- ▶ Delete outermost sample, and replace with a new random one at higher integrand value.
- ▶ The “live points” steadily contract around the peak(s) of the function.
- ▶ We can use this evolution to estimate volume *probabilistically*.
- ▶ At each iteration, the contours contract by  $\sim \frac{1}{n}$  of their volume.
- ▶ This is an exponential contraction, so

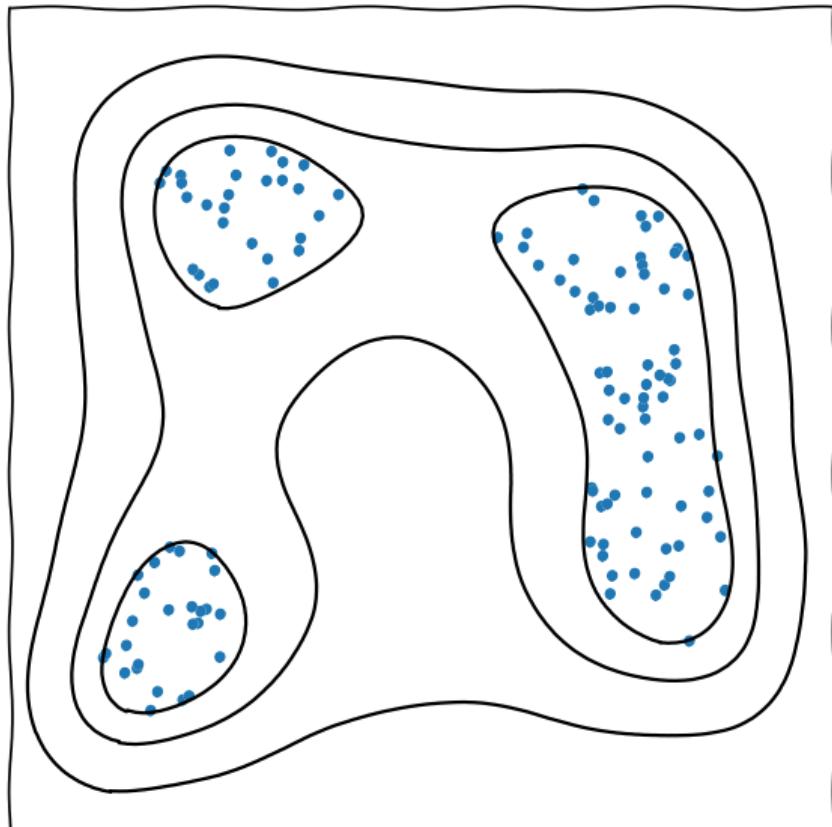
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



## The nested sampling meta-algorithm: live points

- ▶ Start with  $n$  random samples over the space.
- ▶ Delete outermost sample, and replace with a new random one at higher integrand value.
- ▶ The “live points” steadily contract around the peak(s) of the function.
- ▶ We can use this evolution to estimate volume *probabilistically*.
- ▶ At each iteration, the contours contract by  $\sim \frac{1}{n}$  of their volume.
- ▶ This is an exponential contraction, so

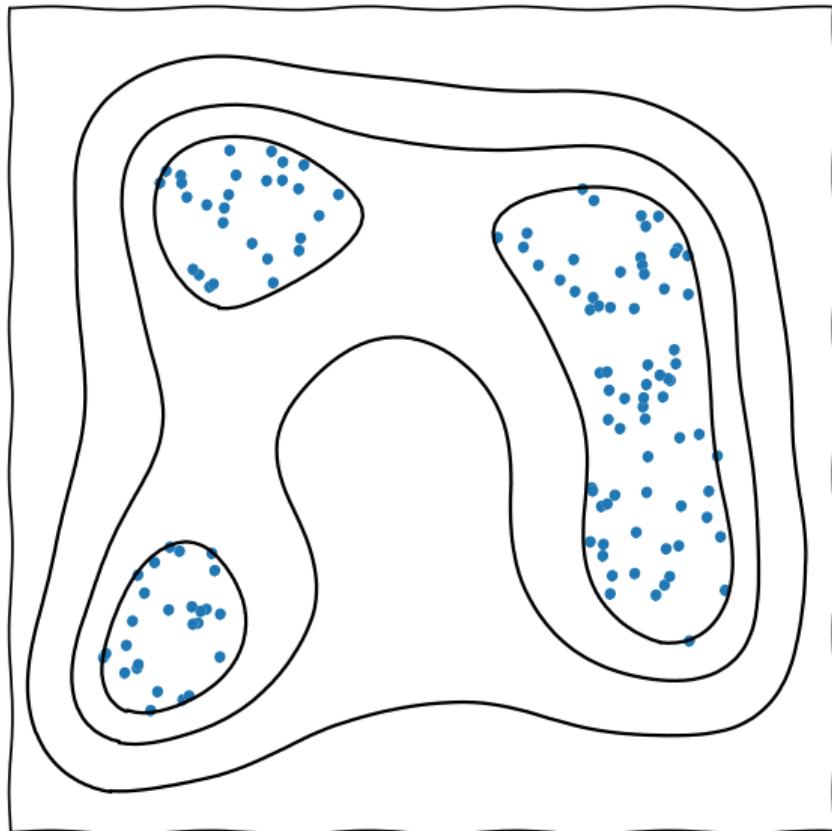
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



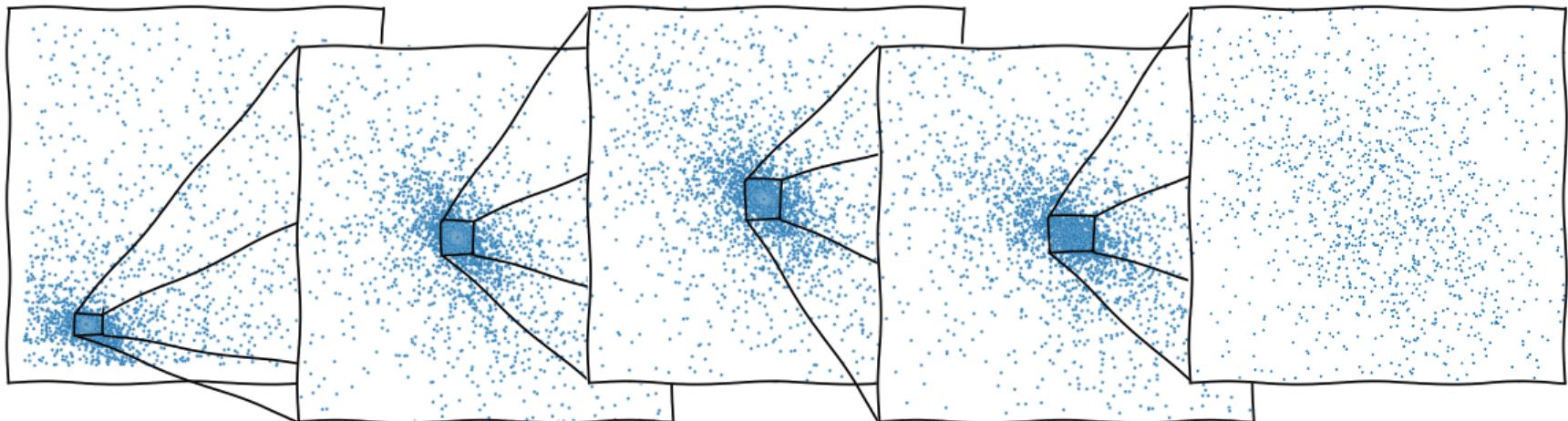
## The nested sampling meta-algorithm: live points

- ▶ Start with  $n$  random samples over the space.
- ▶ Delete outermost sample, and replace with a new random one at higher integrand value.
- ▶ The “live points” steadily contract around the peak(s) of the function.
- ▶ We can use this evolution to estimate volume *probabilistically*.
- ▶ At each iteration, the contours contract by  $\sim \frac{1}{n} \pm \frac{1}{n}$  of their volume.
- ▶ This is an exponential contraction, so

$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-(i \pm \sqrt{i})/n}$$



# The nested sampling meta-algorithm: dead points



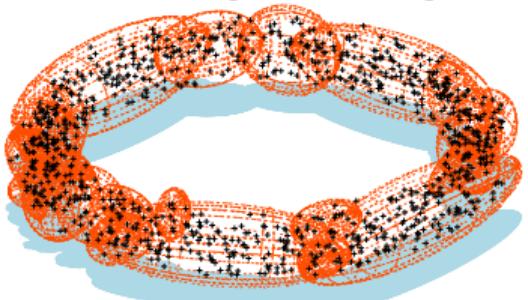
- ▶ At the end, one is left with a set of discarded “dead” points.
- ▶ Dead points have a unique scale-invariant distribution  $\propto \frac{dV}{V}$ .
- ▶ Uniform over original region, exponentially concentrating on region of interest (until termination volume).
- ▶ Good for training emulators (HERA [[2108.07282](#)]).

## Applications

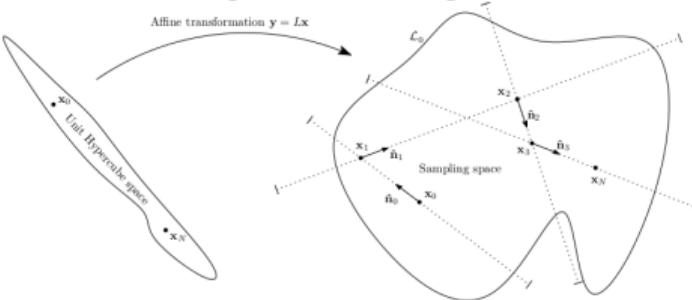
- ▶ training emulators.
- ▶ gridding simulations
- ▶ beta flows
- ▶ “dead measure”

# Implementations of Nested Sampling [2205.15570](NatReview)

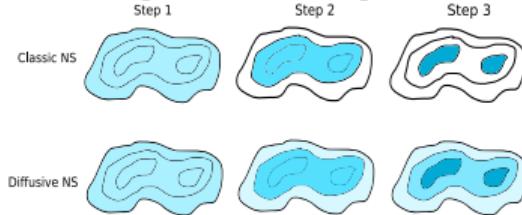
MultiNest [0809.3437]



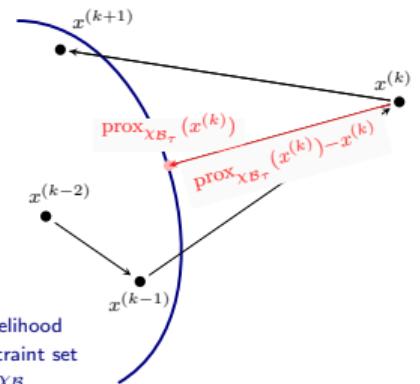
PolyChord [1506.00171]



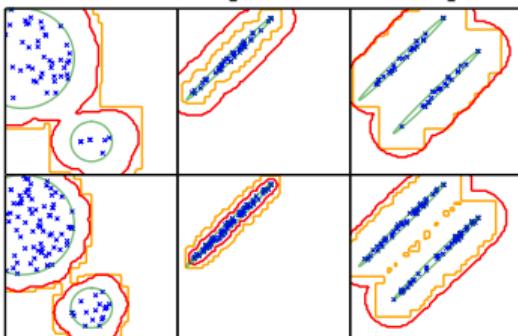
DNest [1606.03757]



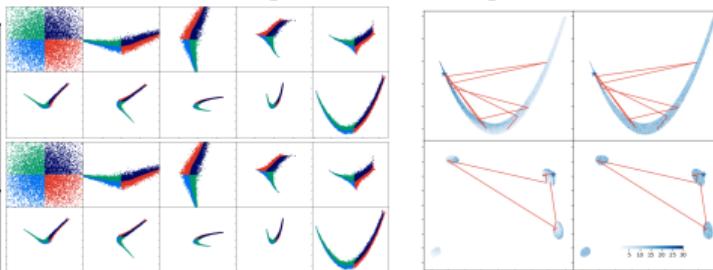
ProxNest [2106.03646]



UltraNest [2101.09604]



NeuralNest [1903.10860]



nessai [2102.11056]

nora [2305.19267]

jaxnest [2012.15286]

nautilus [2306.16923]

<wh260@cam.ac.uk>

willhandley.co.uk/talks

dynesty [1904.02180]

# Types of nested sampler

- ▶ Broadly, most nested samplers can be split into how they create new live points.
- ▶ i.e. how they sample from the hard likelihood constraint  $\{\theta \sim \pi : \mathcal{L}(\theta) > \mathcal{L}_*\}$ .

## Rejection samplers

- ▶ e.g. MultiNest, UltraNest.
- ▶ Constructs bounding region and draws many invalid points until  $\mathcal{L}(\theta) > \mathcal{L}_*$ .
- ▶ Efficient in low dimensions, exponentially inefficient  $\sim \mathcal{O}(e^{d/d_0})$  in high  $d > d_0 \sim 10$ .

- ▶ Nested samplers usually come with:

- ▶ *resolution* parameter  $n_{\text{live}}$  (which improve results as  $\sim \mathcal{O}(n_{\text{live}}^{-1/2})$ ).
- ▶ set of *reliability* parameters [2101.04525], which don't improve results if set arbitrarily high, but introduce systematic errors if set too low.
- ▶ e.g. Multinest efficiency  $\text{eff}$  or PolyChord chain length  $n_{\text{repeats}}$ .

## Chain-based samplers

- ▶ e.g. PolyChord, ProxNest.
- ▶ Run Markov chain starting at a live point, generating many valid (correlated) points.
- ▶ Linear  $\sim \mathcal{O}(d)$  penalty in decorrelating new live point from the original seed point.

# Applications of nested sampling

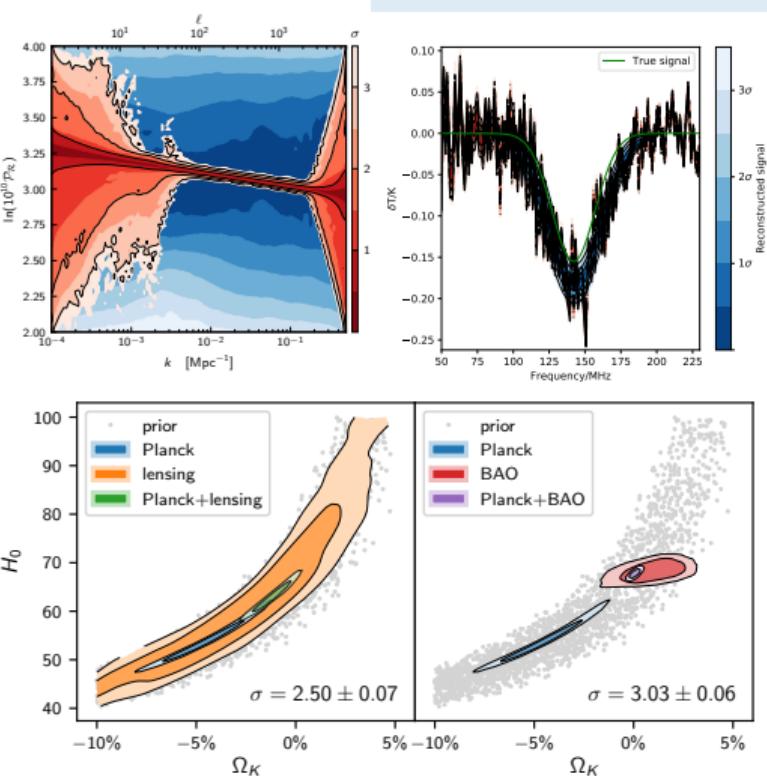
Adam Ormondroyd



PhD

## Cosmology

- ▶ Battle-tested in Bayesian cosmology on
  - ▶ Parameter estimation: multimodal alternative to MCMC samplers.
  - ▶ Model comparison: using integration to compute the Bayesian evidence
  - ▶ Tension quantification: using deep tail sampling and suspiciousness computations.
- ▶ Plays a critical role in major cosmology pipelines: Planck, DES, KiDS, BAO, SNe.
- ▶ The default  $\Lambda$ CDM cosmology is well-tuned to have Gaussian-like posteriors for CMB data.
- ▶ Less true for alternative cosmologies/models and orthogonal datasets, so nested sampling crucial.



1

# Applications of nested sampling

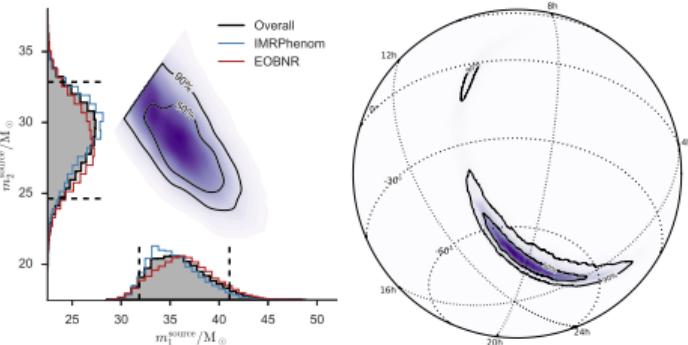
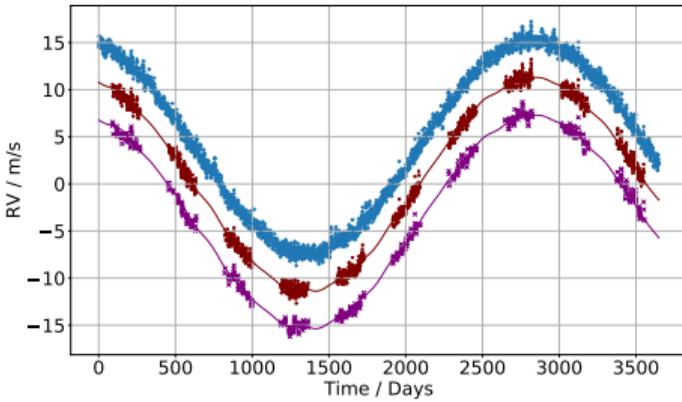
Metha Prathaban

PhD



## Astrophysics

- ▶ In exoplanets [1806.00518]
  - ▶ Parameter estimation: determining properties of planets.
  - ▶ Model comparison: how many planets? Stellar modelling [2007.07278].
  - ▶ exoplanet problems regularly have posterior phase transitions [2102.03387]
- ▶ In gravitational waves
  - ▶ Parameter estimation: Binary merger properties
  - ▶ Model comparison: Modified theories of gravity, selecting phenomenological parameterisations [1803.10210]
  - ▶ Likelihood reweighting: fast slow properties



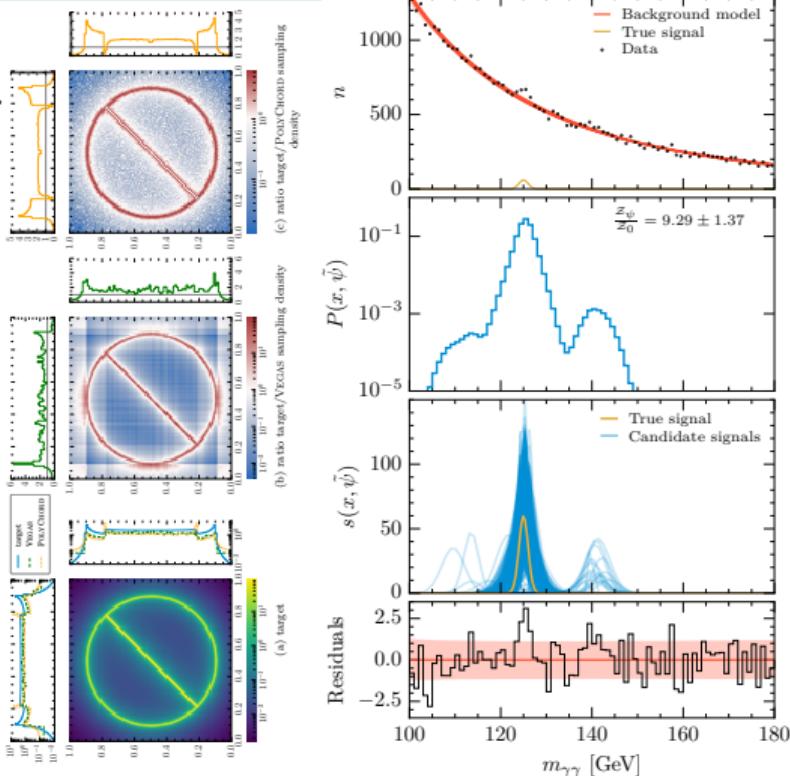
# Applications of nested sampling

## Particle physics

- ▶ Nested sampling for cross section computation/event generation  $\sigma = \int_{\Omega} d\Phi |\mathcal{M}|^2$ .
- ▶ Nested sampling can explore the phase space  $\Omega$  and compute integral blind with comparable efficiency to HAAG/RAMBO [2205.02030].
- ▶ Bayesian sparse reconstruction [1809.04598] applied to bump hunting allows evidence-based detection of signals in phenomenological backgrounds [2211.10391].
- ▶ Fine tuning quantification
- ▶ Fast estimation of small  $p$ -values [2106.02056](PRL), just make switch:  
 $X \leftrightarrow p, \mathcal{L} \leftrightarrow \lambda, \theta \leftrightarrow x.$

David Yallup

PDRA



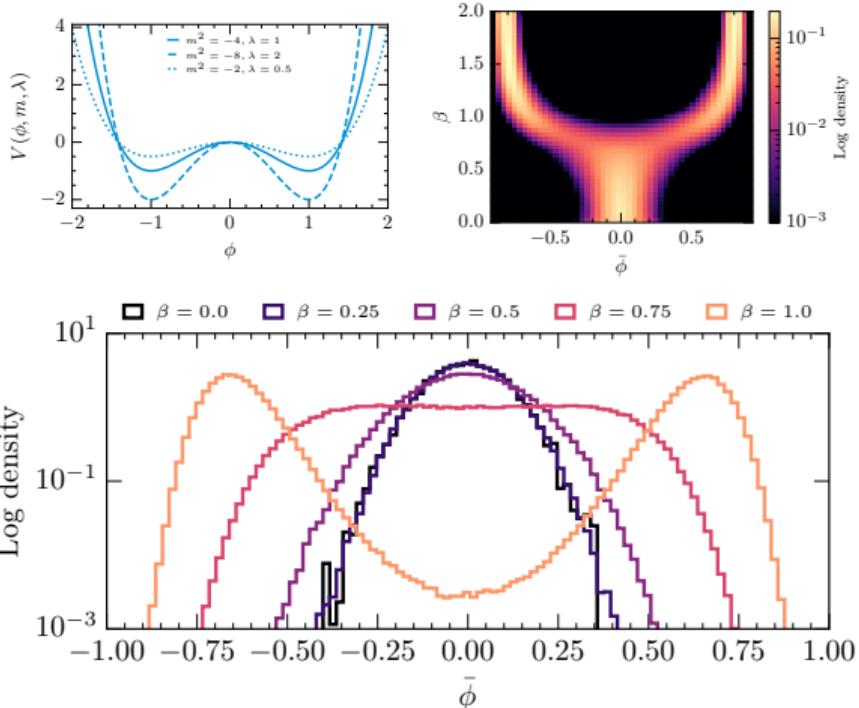
# Applications of nested sampling

## Lattice field theory

- Consider standard field theory Lagrangian:

$$Z(\beta) = \int D\phi e^{-\beta S(\phi)}, \quad S(\phi) = \int dx^\mu \mathcal{L}(\phi)$$

- Discretize onto spacetime grid.
- Compute partition function
- NS unique traits:
  - Get full partition function for free
  - allows for critical tuning
  - avoids critical slowing down
- Applications in lattice gravity, QCD, condensed matter physics

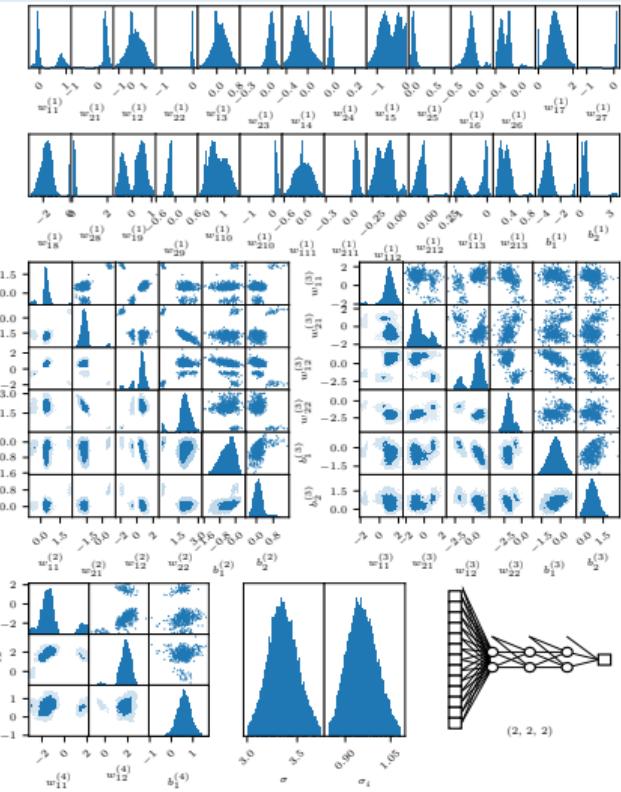




# Applications of nested sampling

## Machine learning

- ▶ Machine learning requires:
  - ▶ Training to find weights
  - ▶ Choice of architecture/topology/hyperparameters
- ▶ Bayesian NNs treat training as a model fitting problem
- ▶ Compute posterior of weights (parameter estimation), rather than optimisation (gradient descent)
- ▶ Use evidence to determine best architecture (model comparison), correlates with out-of-sample performance!
- ▶ Solving the full “shallow learning” problem without compromise [2004.12211][2211.10391].
  - ▶ Promising work ongoing to extend this to transfer learning and deep nets.
- ▶ More generally, dead points are optimally spaced for training traditional ML approaches e.g. [2309.05697]



# Applications of nested sampling

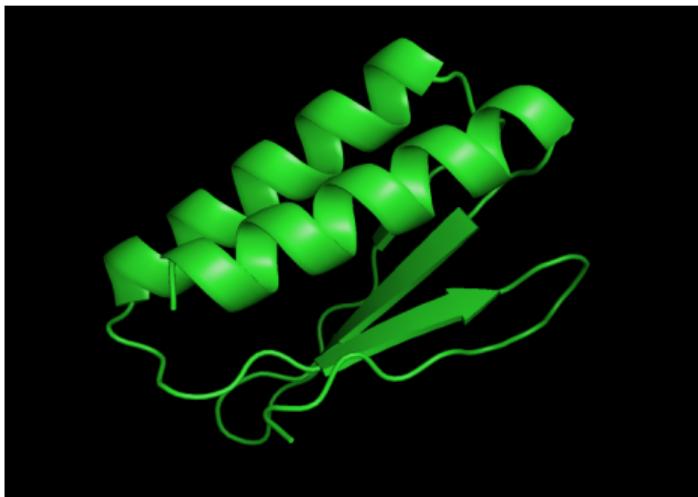
And beyond...

Catherine Watkinson



Senior Data Scientist

- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

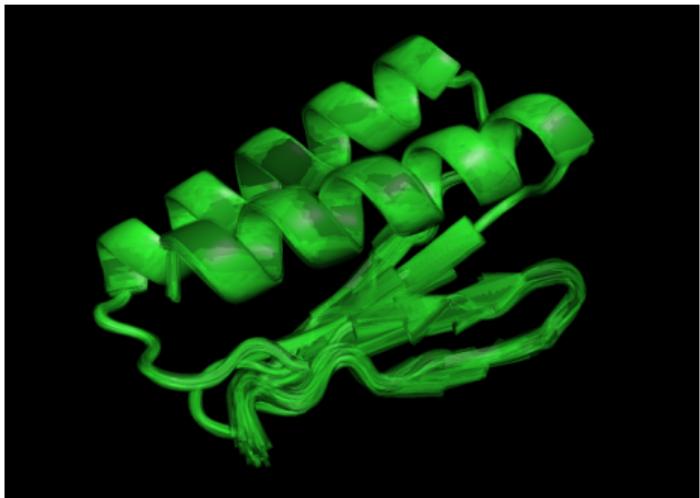
And beyond...

Catherine Watkinson



Senior Data Scientist

- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

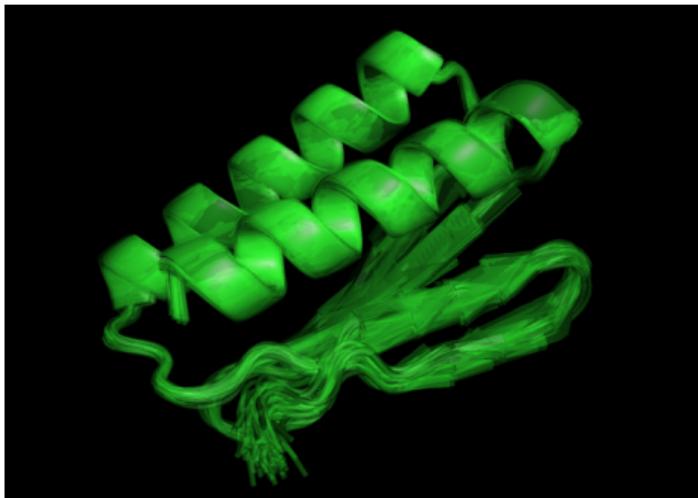
And beyond...

Catherine Watkinson

Senior Data Scientist



- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

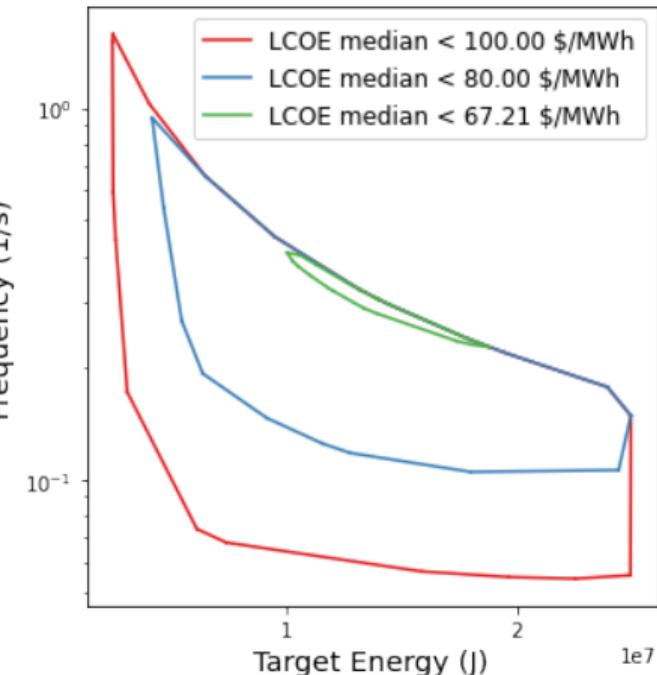
And beyond...

- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



Catherine Watkinson

Senior Data Scientist



# Applications of nested sampling

And beyond...

Thomas Mcaloone

PhD → Data Scientist



- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

And beyond...

Thomas Mcaloone

PhD → Data Scientist



- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

And beyond...

Thomas Mcaloone

PhD → Data Scientist



- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



# Applications of nested sampling

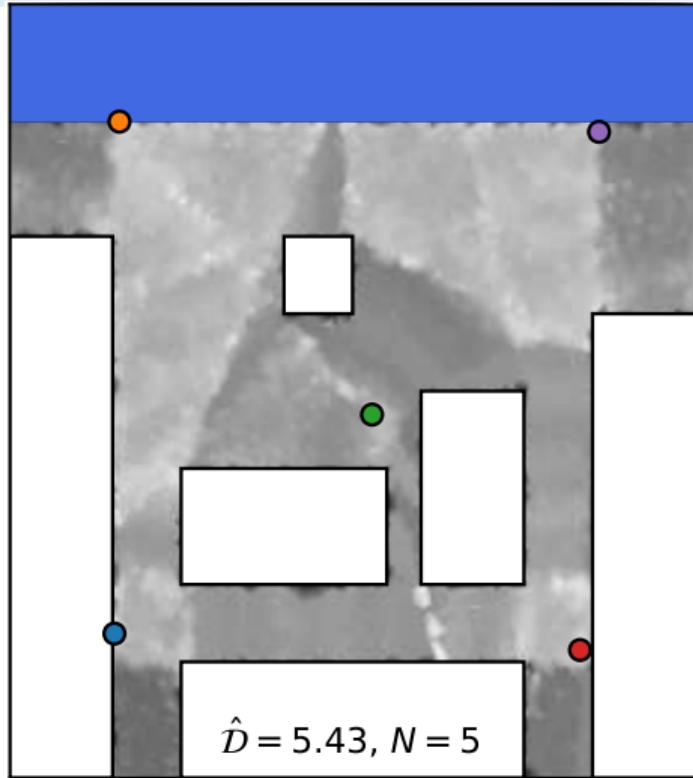
And beyond...

Thomas Mcaloone

PhD → Data Scientist



- ▶ Techniques have been spun-out (PolyChord Ltd) to:
- ▶ Protein folding
  - ▶ Navigating free energy surface.
  - ▶ Computing misfolds.
  - ▶ Thermal motion.
- ▶ Nuclear fusion reactor optimisation
  - ▶ multi-objective.
  - ▶ uncertainty propagation.
- ▶ Telecoms & DSTL research (MIDAS)
  - ▶ Optimising placement of transmitters/sensors.
  - ▶ Maximum information data acquisition strategies.



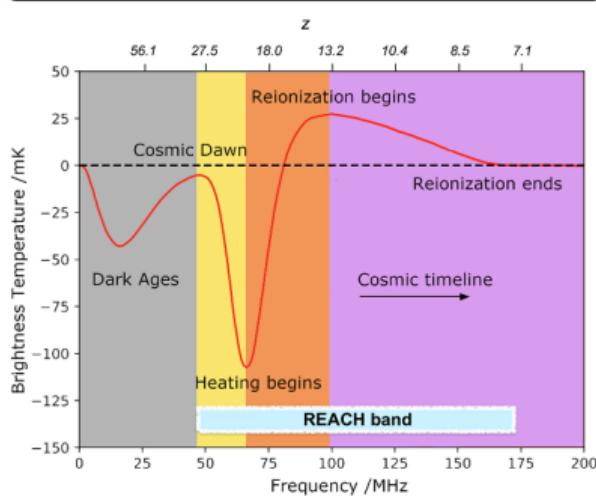
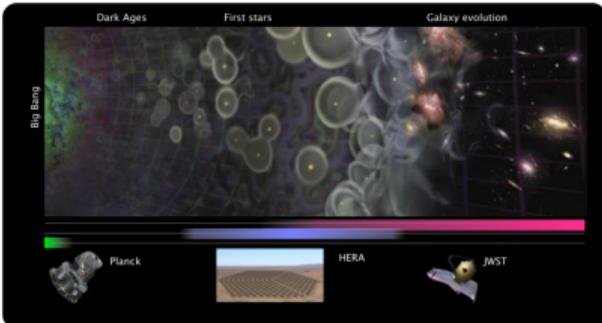
# REACH: Global 21cm cosmology [2210.07409](NatAstro)

Ian Roque

PhD



- ▶ Imaging the universal dark ages using CMB backlight.
- ▶ 21cm hyperfine line emission from neutral hydrogen.
- ▶ Global experiments measure monopole across frequency.
- ▶ Challenge: science hidden in foregrounds  $\sim 10^4 \times$  signal.
- ▶ Lead data analysis team (REACH first light in January)
- ▶ Nested sampling woven in from the ground up (calibrator, beam modelling, signal fitting, likelihood selection).
- ▶ All treated as parameterised model comparison problems.



# GAMBIT

## Interdisciplinary case studies

- ▶ GAMBIT is an interdisciplinary community and software framework.
- ▶ Like CosmoMC/Cobaya/Bilby, an organiser of data, likelihoods & theory, including:
  - ▶ Collider data (e.g. LHC)
  - ▶ Direct detections (e.g. XENON1T)
  - ▶ Cosmology (MontePython)
  - ▶ Astrophysics (e.g. Bullet Cluster, Supernovae)
  - ▶ Pulsar timing
  - ▶ ... & much more
- ▶ GravBit and LowEnergyBit arising from GAMBIT@KICC workshop



# GAMBIT: sub-GeV Dark matter constraints

## Interdisciplinary case studies

Felix Kahlhoefer et al

GAMBIT cosmo/DM working group



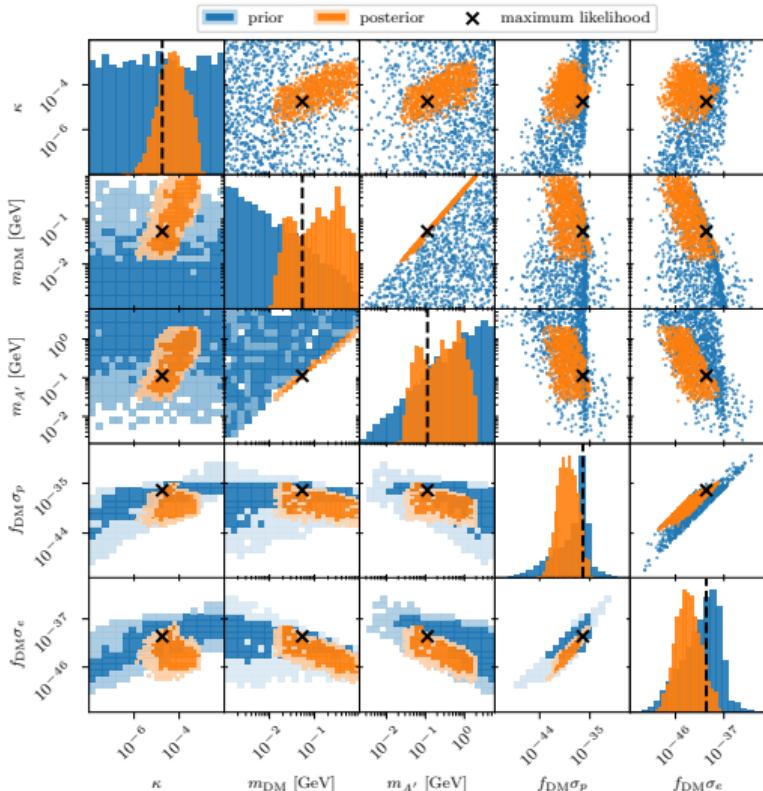
- ▶ Physical model of sub-GeV thermal dark matter with a dark photon mediator  $A$ :

$$\mathcal{L}_{\text{int}} = -\frac{1}{2} m_{A'}^2 A'^{\mu} A'_{\mu} - \frac{1}{4} A'^{\mu\nu} A'_{\mu\nu} - \kappa e A'^{\mu} \sum_f q_f \bar{f} \gamma_{\mu} f ,$$

- ▶ Constrain using cosmological, astrophysical, accelerator & direct detection data.
- ▶ Bayesian Model comparison of Fermion  $\psi$  vs scalar  $\Phi$  models (scalar preferred).

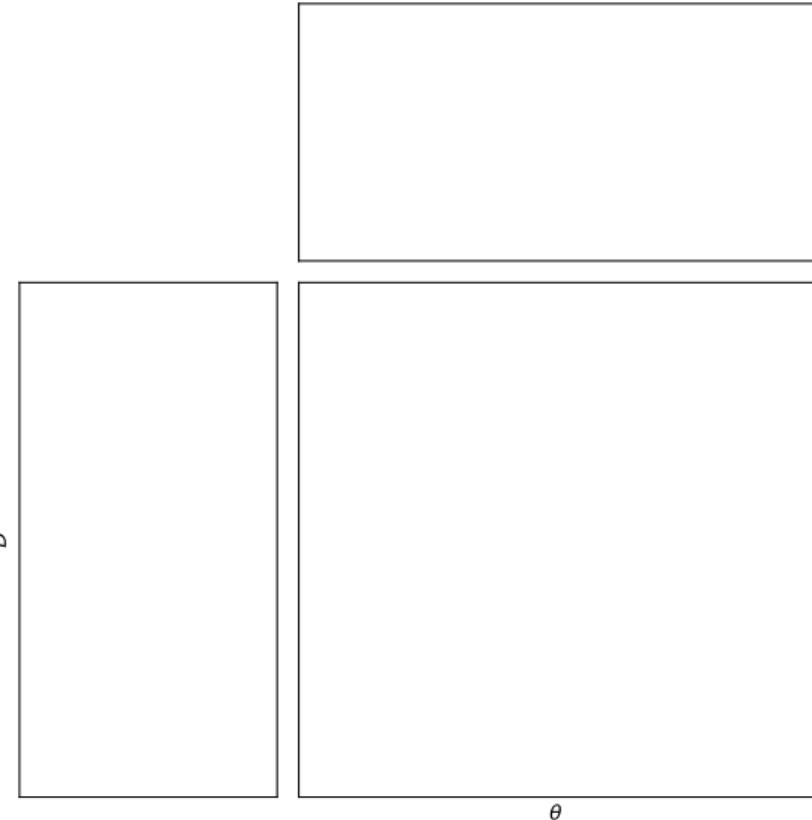
$$\mathcal{L}_{\psi} = \bar{\psi} (i\cancel{\partial} - m_{\text{DM}}) \psi + g_{\text{DM}} A'^{\mu} \bar{\psi} \gamma_{\mu} \psi ,$$

$$\begin{aligned} \mathcal{L}_{\Phi} = & |\partial_{\mu} \Phi|^2 - m_{\text{DM}}^2 |\Phi|^2 - g_{\text{DM}}^2 A'_{\mu} A'^{\mu} |\Phi|^2 \\ & + i g_{\text{DM}} A'^{\mu} [\Phi^* (\partial_{\mu} \Phi) - (\partial_{\mu} \Phi^*) \Phi] , \end{aligned}$$



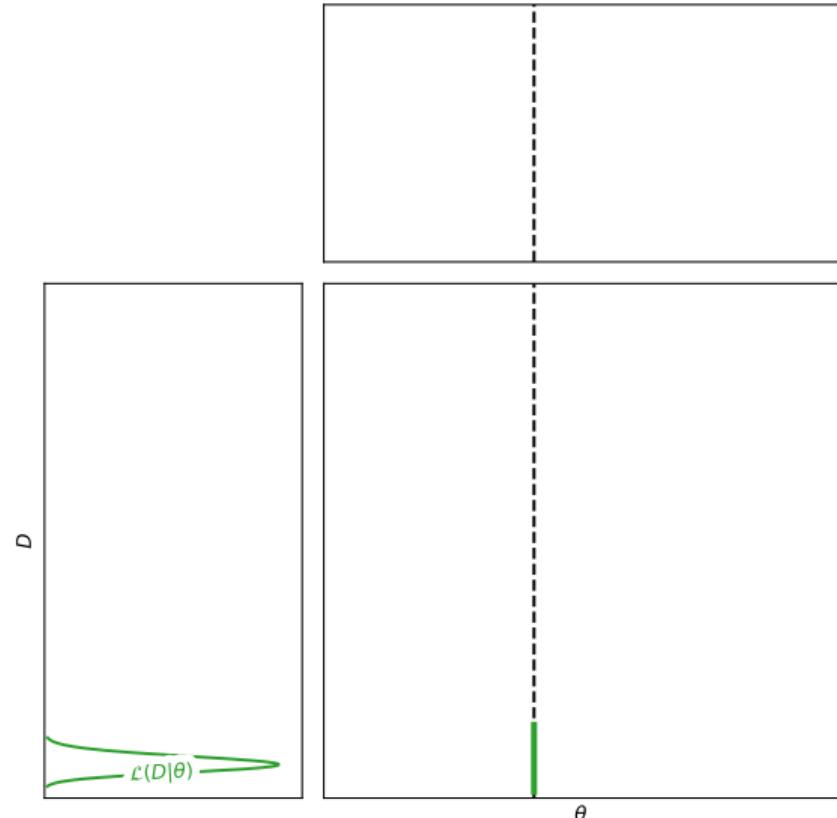
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from  
joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and  
learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using  
*machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



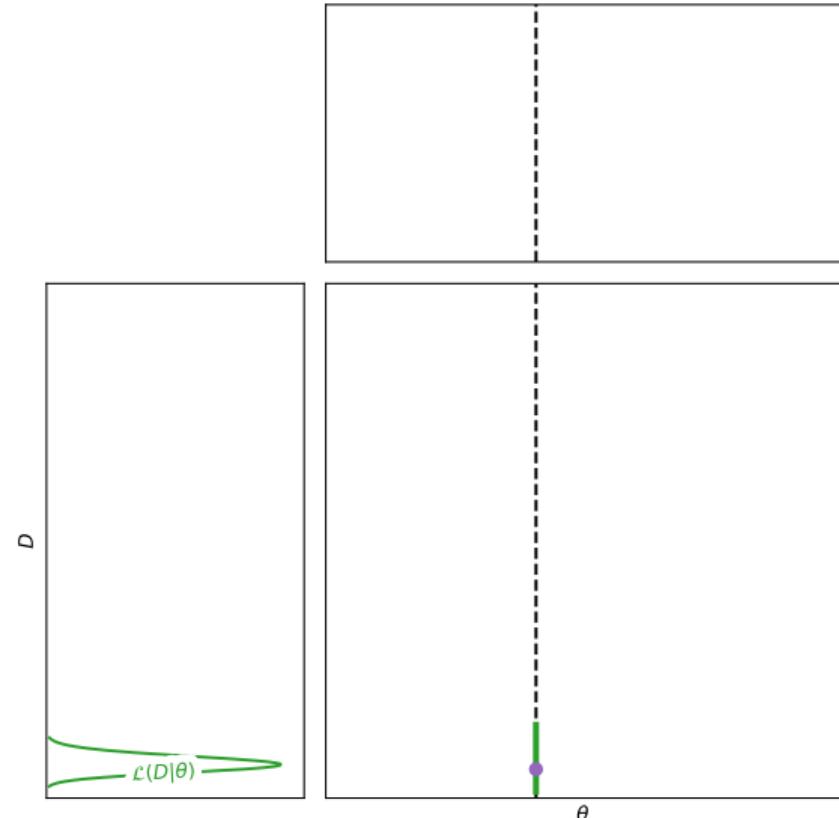
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



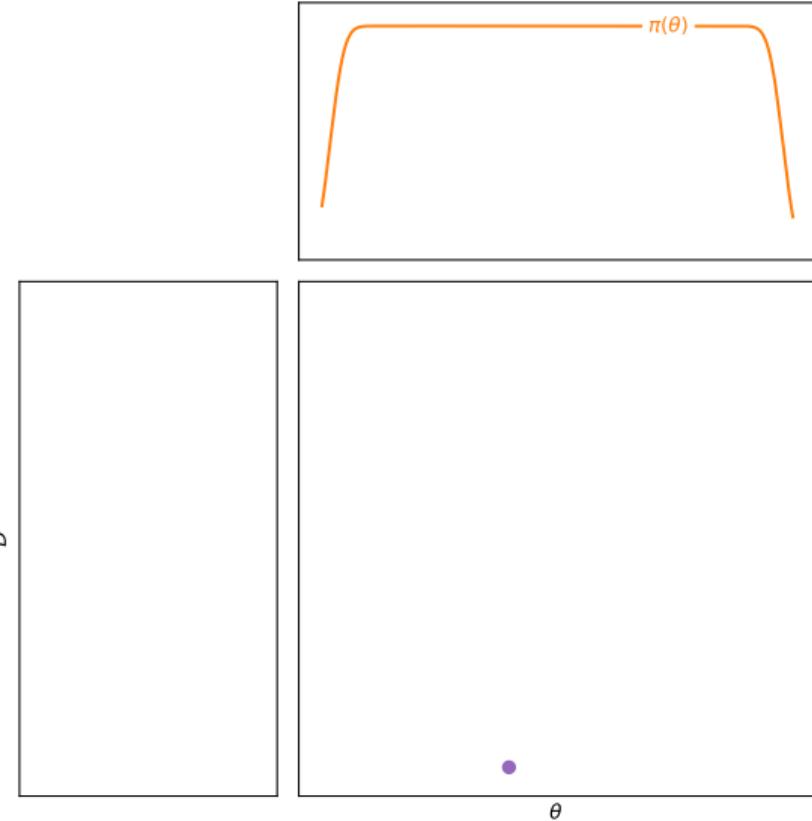
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



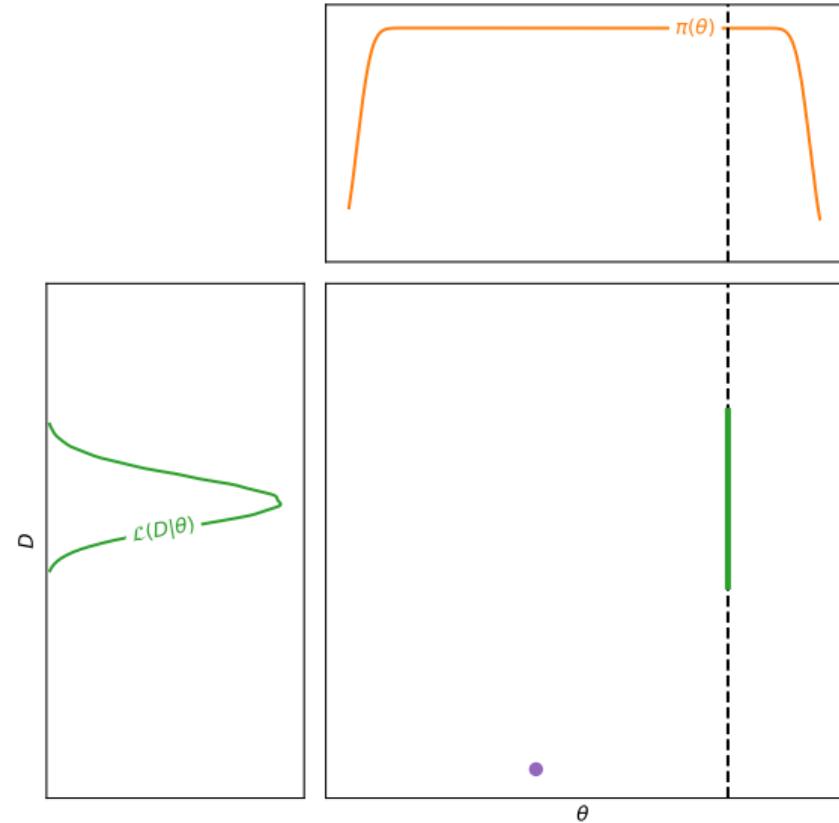
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



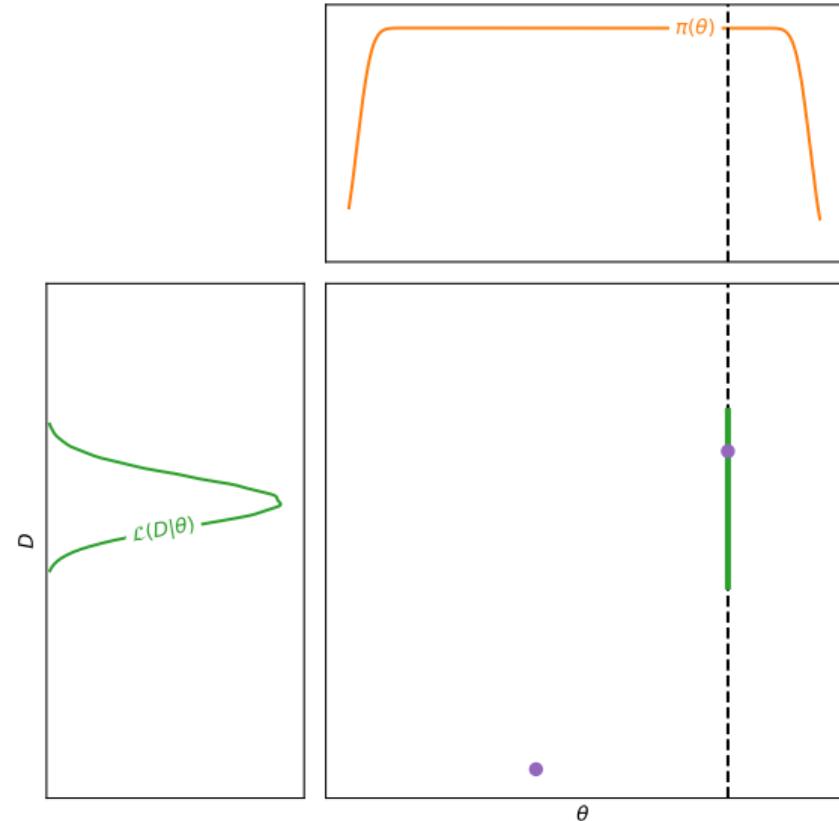
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to remove machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



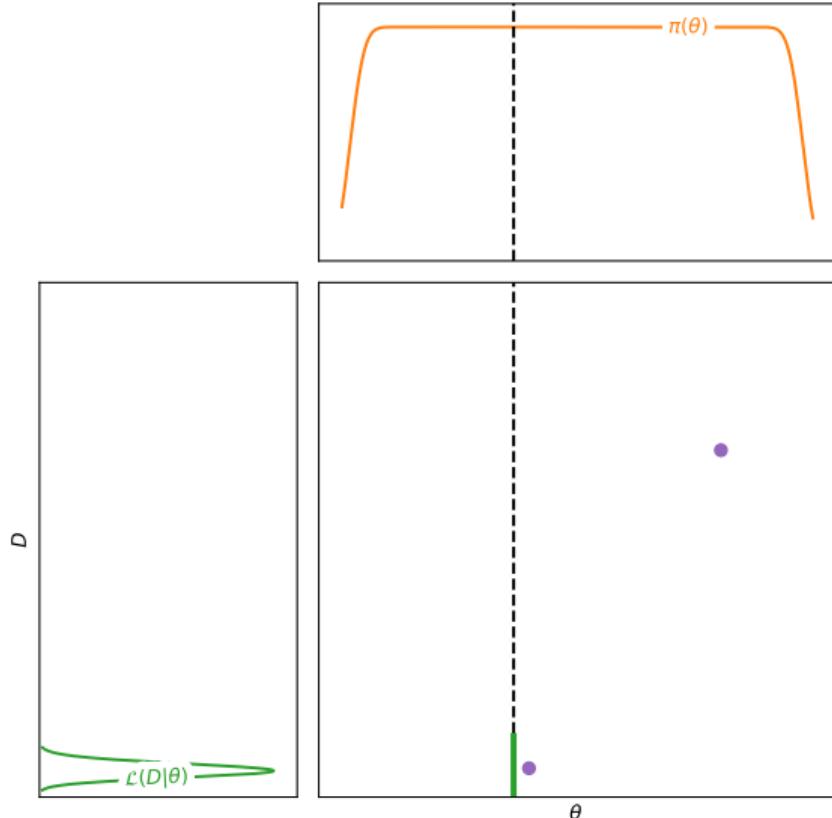
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



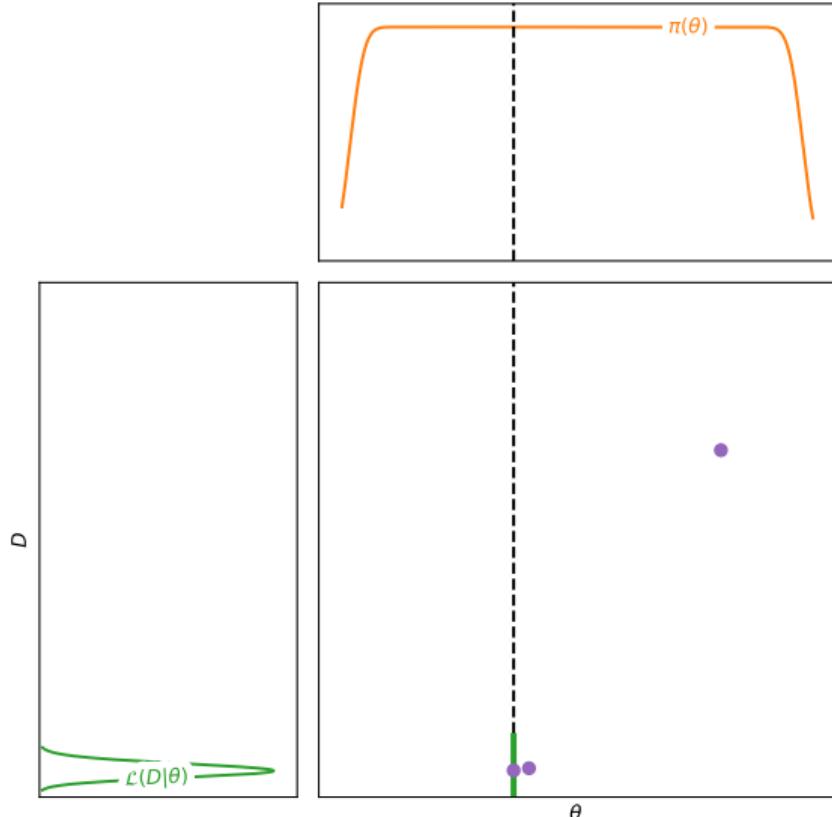
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



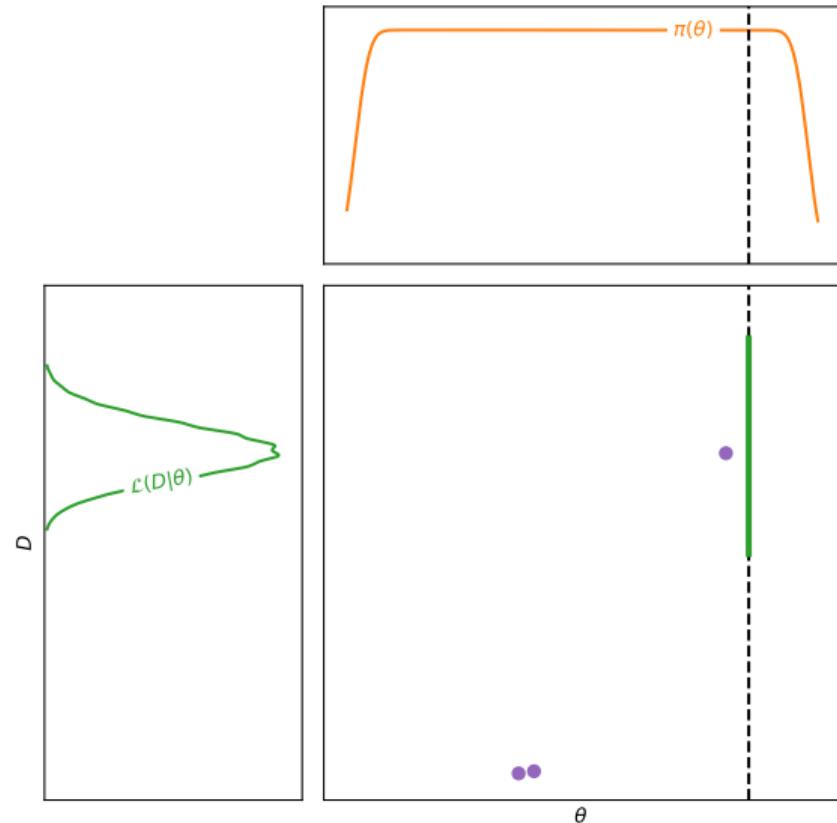
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



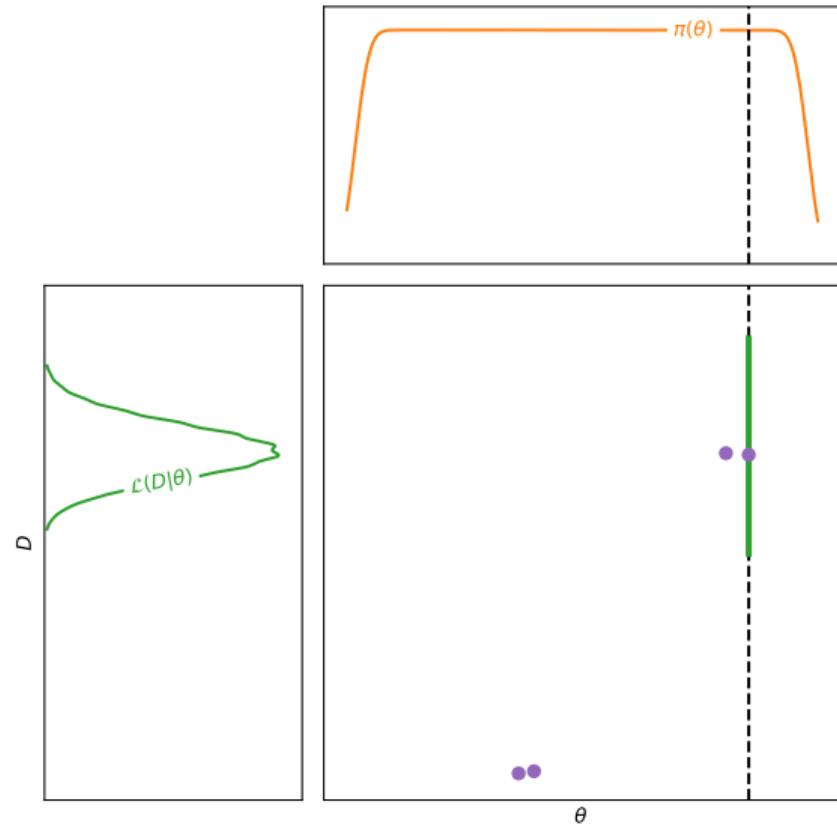
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



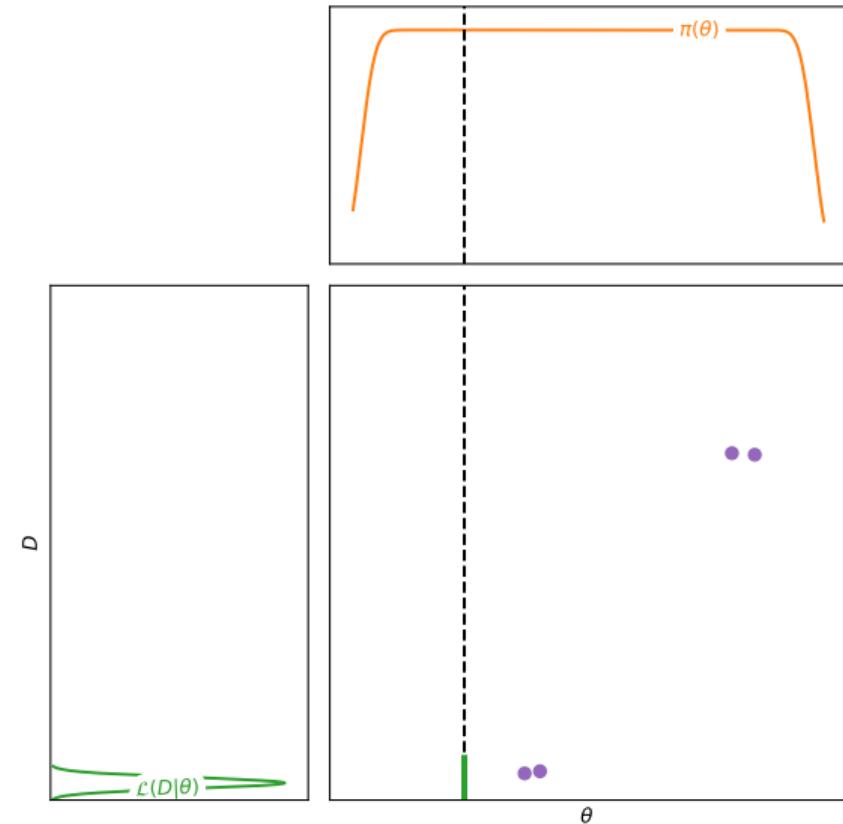
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



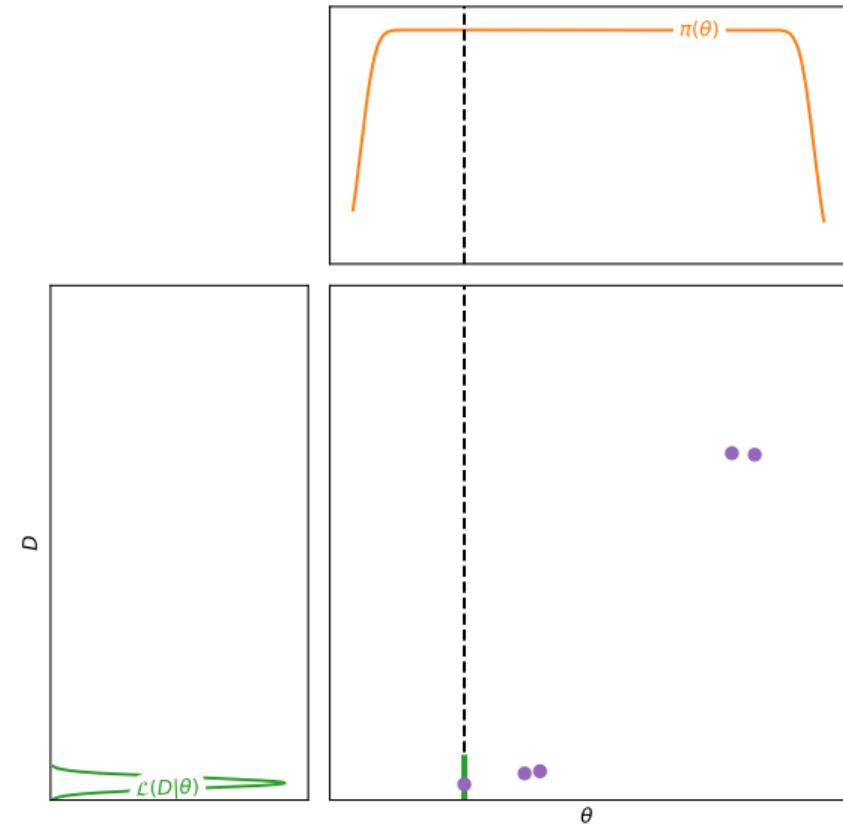
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



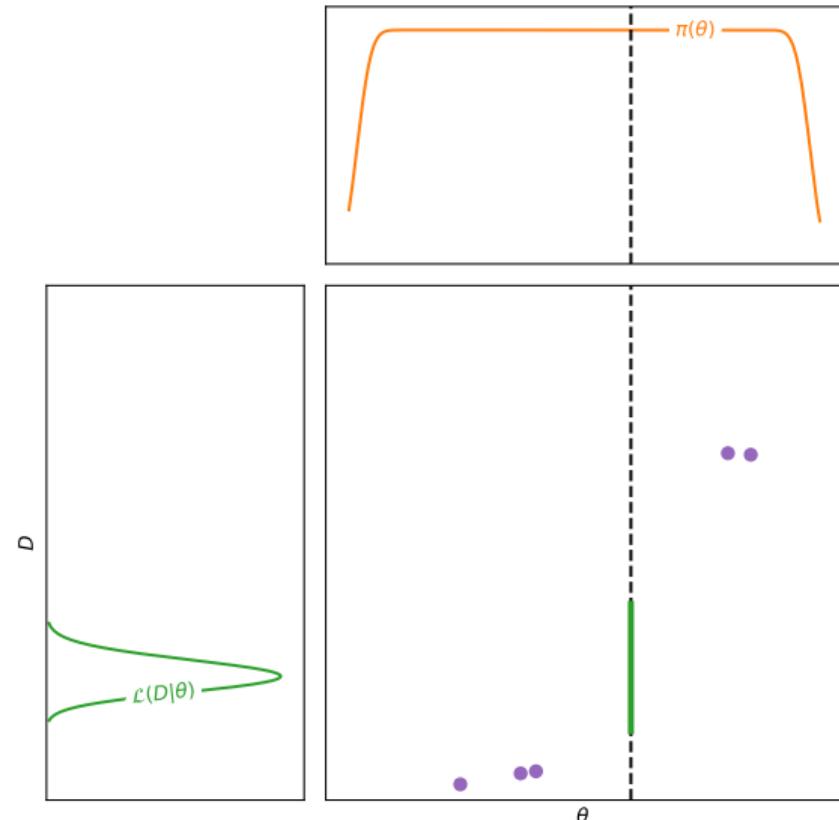
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



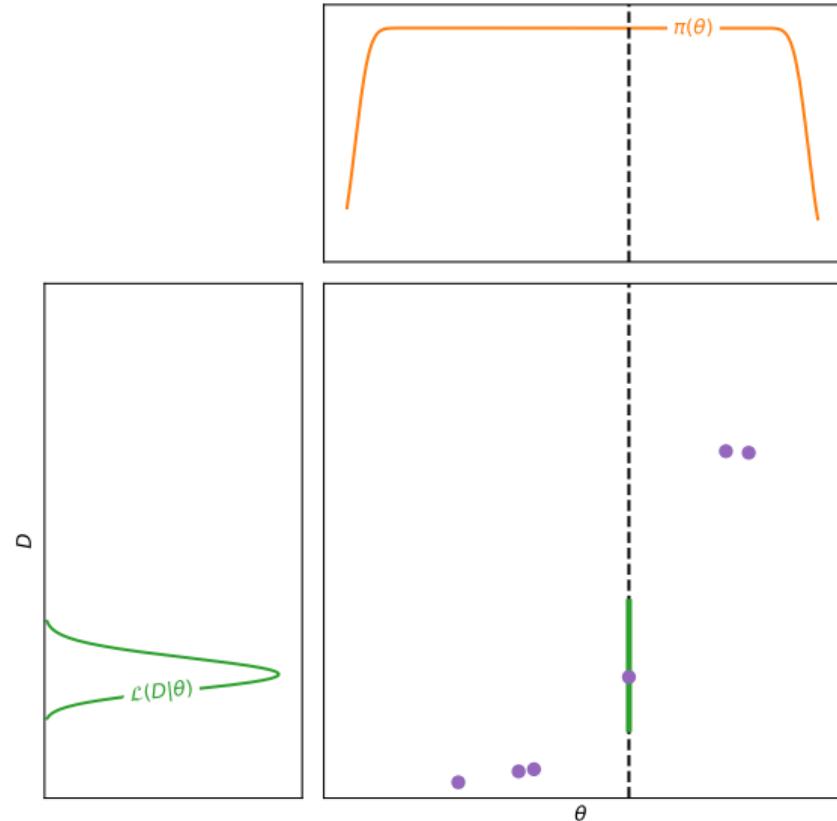
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



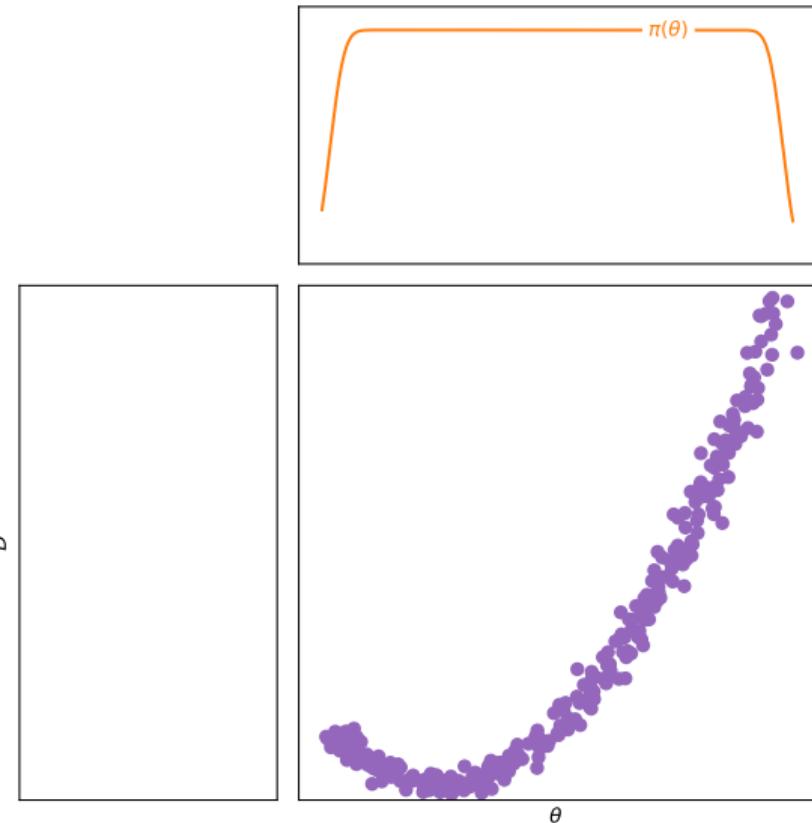
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



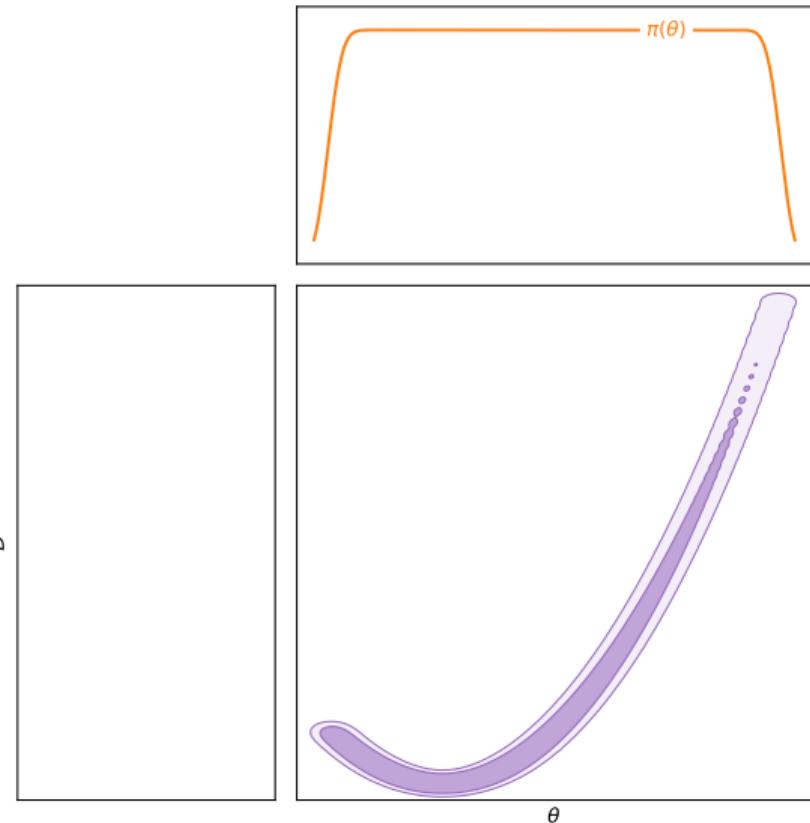
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



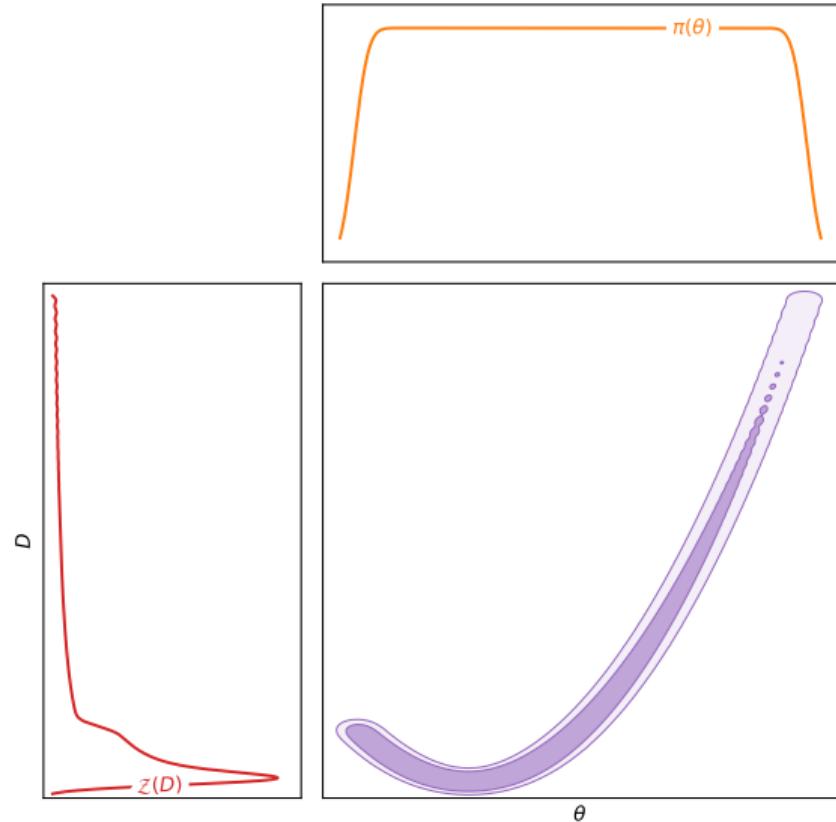
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$   
and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to remove machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



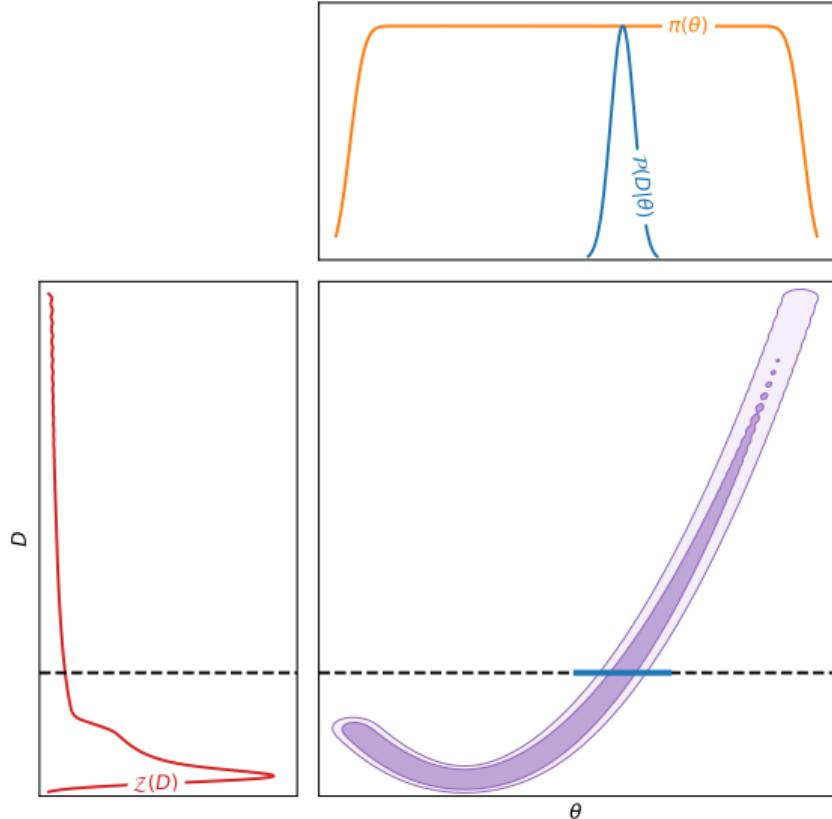
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



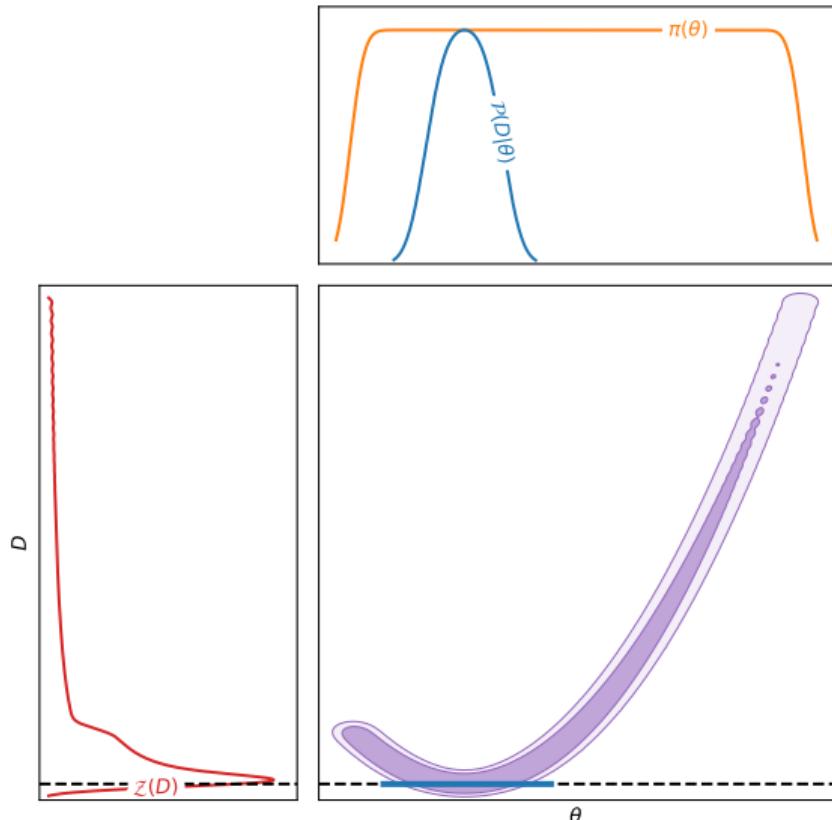
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “*probability of everything*”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to removes machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



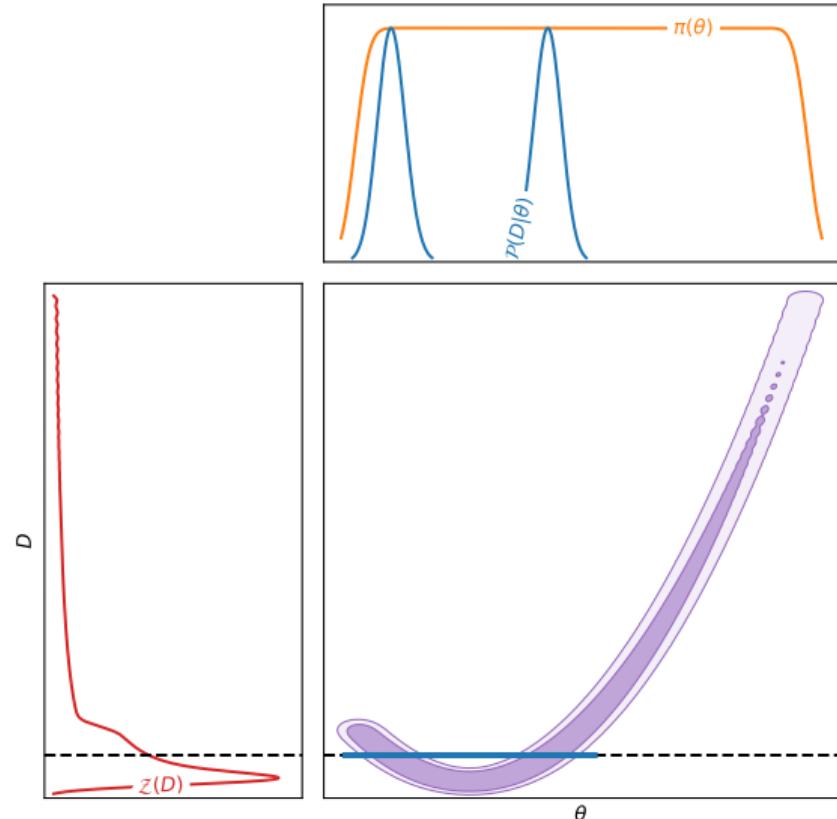
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to remove machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



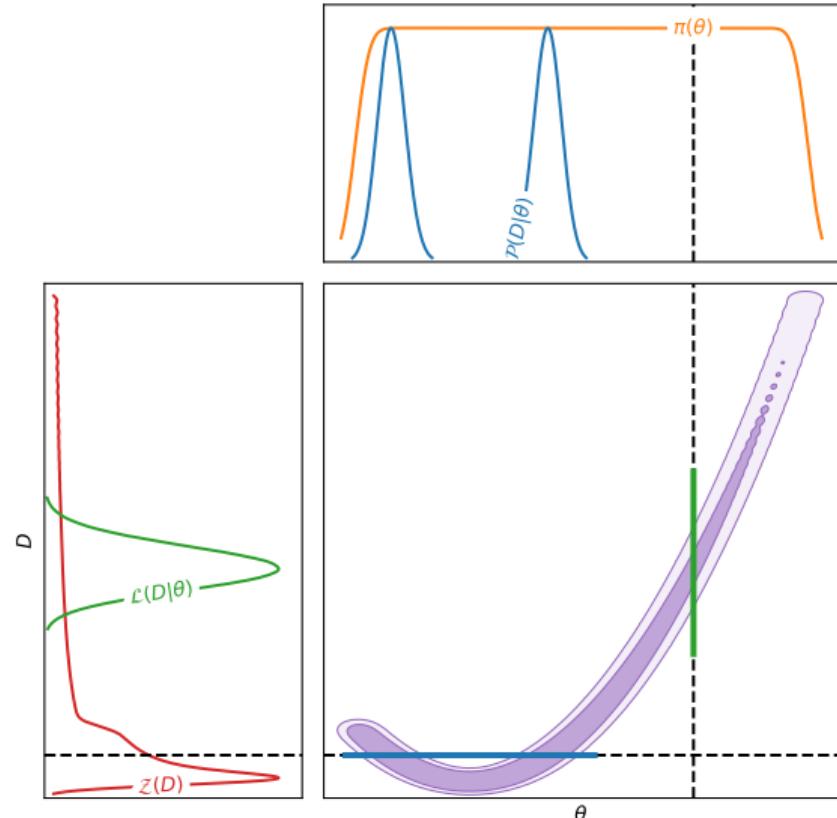
# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to remove machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



# SBI: Simulation-based inference

- ▶ What do you do if you don't know  $\mathcal{L}(D|\theta)$ ?
- ▶ If you have a simulator/forward model  
 $\theta \rightarrow D$  defines an *implicit likelihood*  $\mathcal{L}$ .
- ▶ Simulator generates samples from  $\mathcal{L}(\cdot|\theta)$ .
- ▶ With a prior  $\pi(\theta)$  can generate samples from joint distribution  $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$   
the “probability of everything”.
- ▶ Task of SBI is take joint  $\mathcal{J}$  samples and learn posterior  $\mathcal{P}(\theta|D)$  and evidence  $\mathcal{Z}(D)$  and possibly likelihood  $\mathcal{L}(D|\theta)$ .
- ▶ Present state of the art achieves this using *machine learning* (neural networks).
  - ▶ My group's research tries to remove machine learning [github.com/handley-lab/lsbi](https://github.com/handley-lab/lsbi).



# Why SBI?

SBI is useful because:

1. If you don't have a likelihood, you can still do inference
  - ▶ This is the usual case beyond CMB cosmology
2. Faster than LBI
  - ▶ emulation – also applies to LBI in principle
3. No need to pragmatically encode fiducial cosmologies
  - ▶ Covariance computation implicitly encoded in simulations
  - ▶ Highly relevant for disentangling tensions & systematics
4. Equips AI/ML with Bayesian interpretability
5. Lower barrier to entry than LBI
  - ▶ Much easier to forward model a systematic
  - ▶ Emerging set of plug-and-play packages
  - ▶ For this reason alone, it will come to dominate scientific inference



[github.com/sbi-dev](https://github.com/sbi-dev)



[github.com/undark-lab/swyft](https://github.com/undark-lab/swyft)



[github.com/florent-leclercq/pyselfi](https://github.com/florent-leclercq/pyselfi)



[github.com/justinalsing/pydelfi](https://github.com/justinalsing/pydelfi)

# SBI in cosmology

- ▶ 2024 has been the year it has started to be applied to real data.
- ▶ Mostly for weak lensing
- ▶ However: SBI requires mock data generation code
- ▶ Most data analysis codes were built before the generative paradigm.
- ▶ It's still a lot of work to upgrade cosmological likelihoods to be able to do this (e.g. plik & camspec).

## Investigating the turbulent hot gas in X-COP galaxy clusters

S. Dupourqué<sup>1</sup>, N. Clerc<sup>1</sup>, E. Pointecouteau<sup>1</sup>, D. Eckert<sup>2</sup>, S. Ettori<sup>3</sup>, and F. Vazza<sup>4,5,6</sup>

Dark Energy Survey Year 3 results: simulation-based cosmological inference with wavelet harmonics, scattering transforms, and moments of weak lensing mass maps II. Cosmological results

M. Gatti,<sup>1,\*</sup> G. Campailla,<sup>2</sup> N. Jeffrey,<sup>3</sup> L. Whitney,<sup>3</sup> A. Paredes,<sup>4</sup> J. Prat,<sup>5</sup> J. Williamson,<sup>3</sup> M. Ravera,<sup>2</sup> B.

## Neural Posterior Estimation with guaranteed exact coverage: the ringdown of GW150914

Marco Crisostomi<sup>1,2</sup>, Kallol Dey<sup>3</sup>, Enrico Barausse<sup>1,2</sup>, Roberto Trotta<sup>1,2,4,5</sup>

## Applying Simulation-Based Inference to Spectral and Spatial Information from the Galactic Center Gamma-Ray Excess

Katharena Christy,<sup>a</sup> Eric J. Baxter,<sup>b</sup> Jason Kumar<sup>a</sup>

## KIDS-1000 and DES-Y1 combined: Cosmology from peak count statistics

Joachim Harnois-Déraps<sup>1\*</sup>, Sven Heydenreich<sup>2</sup>, Benjamin Giblin<sup>3</sup>, Nicolas Martinet<sup>4</sup>,  
Tilman Tröster<sup>5</sup>, Marika Asgari<sup>1,6,7</sup>, Pierre Burger<sup>8,9,10</sup>, Tiago Castro<sup>1,12,13,14</sup>,  
Klaus Dolag<sup>15</sup>, Catherine Heymans<sup>3,16</sup>, Hendrik Hildebrandt<sup>16</sup>, Benjamin Joachimi<sup>17</sup> &  
Angus H. Wright<sup>16</sup>

## KiDS-SBI: Simulation-Based Inference Analysis of KiDS-1000 Cosmic Shear

Maximilian von Wietersheim-Kramsta<sup>1,2,3</sup>, Kiyam Lin<sup>4</sup>, Nicolas Tessore<sup>1</sup>, Benjamin Joachimi<sup>1</sup>, Arthur Loureiro<sup>4,5</sup>,  
Robert Reischke<sup>6,7</sup>, and Angus H. Wright<sup>1</sup>

## Simulation-based inference of deep fields: galaxy population model and redshift distributions

Beatrice Moser,<sup>a,1</sup> Tomasz Kacprzak,<sup>a,b</sup> Silvan Fischbacher,<sup>a</sup>  
Alexandre Refregier,<sup>a</sup> Dominic Grimm,<sup>a</sup> Luca Tortorelli<sup>c</sup>

## SmBiG: Cosmological Constraints using Simulation-Based Inference of Galaxy Clustering with Marked Power Spectra

ELENA MASSARA  <sup>1,2,\*</sup>, CHANGHOON HAN  <sup>2</sup>, MICHAEL EICKENBERG <sup>2</sup>, SHERELY HO <sup>3</sup>, JIAMIN HOU <sup>2</sup>,  
PABLO LEMOS <sup>4,5</sup>, CHIRAG MODI <sup>4,6</sup>, AZADEH MORADNEZHAD DEHGHAN  <sup>7,8,11</sup>, LIAM PARKER <sup>3,12</sup> AND  
BENOÎT RÉGALDO-SAINT BLANCARD 

# Conclusions

[github.com/handley-lab](https://github.com/handley-lab)



- ▶ Nested sampling is a multi-purpose numerical tool for:
  - ▶ Numerical integration  $\int f(x) dV$ ,
  - ▶ Exploring/scanning/optimising *a priori* unknown functions,
  - ▶ Performing Bayesian inference and model comparison.
- ▶ It is applied widely across cosmology & particle physics.
- ▶ It's unique traits as the only numerical Lebesgue integrator mean with compute it will continue to grow in importance.
- ▶ SBI represents the future of inference beyond LBI.

