

lsbi: linear simulation based inference

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

11th September 2024



UNIVERSITY OF
CAMBRIDGE



Context from phystat

- ▶ From **Jesse Thaler**'s talk on Interpretable Machine Learning:
*"If asked what is the most under-used Machine Learning technique in physics...
... my answer is only half-jokingly linear regression."*
- ▶ sbi mentioned in (so far)
 - ▶ Monday
 - [Ben Wandelt](#) Cosmology and machine learning
 - [Maximilian Dax](#) Simulation-based machine learning for gravitational-wave analysis
 - [Andre Scaffidi](#) Anomaly aware machine learning for dark matter direct detection at the DARWIN experiment
 - [Joshua Villarrea](#) Feldman-Cousins' ML Cousin
 - ▶ Tuesday
 - [Aishik Ghosh](#) Simulation-based Inference (SBI)

Who?

Idea I've been working on/talking about for the better part of 18 months,

- ▶ Nicolas Mediatto Diaz (MSci project)
- ▶ David Yallup (Postdoc)
- ▶ Thomas Gessey Jones (Postdoc)

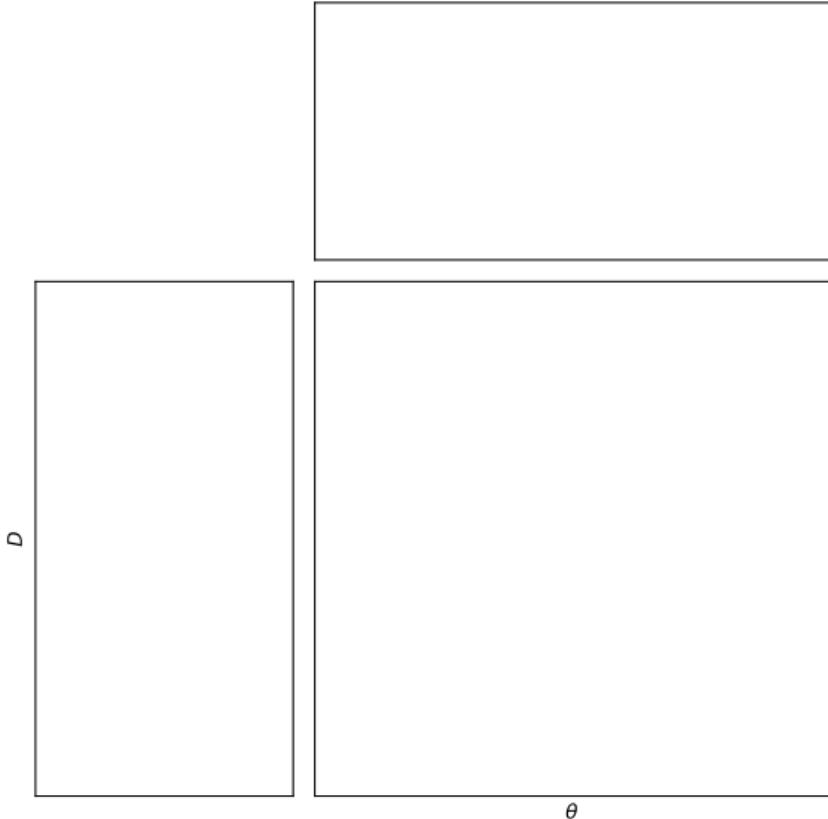
Many others have also presented this idea independently

- ▶ SELFI incorporates much of this idea: Leclercq [[1902.10149](#)]
- ▶ some of these ideas are in MOPED: Heavens [[astro-ph/9911102](#)]
- ▶ Also appears in Häggström [[2403.07454](#)]



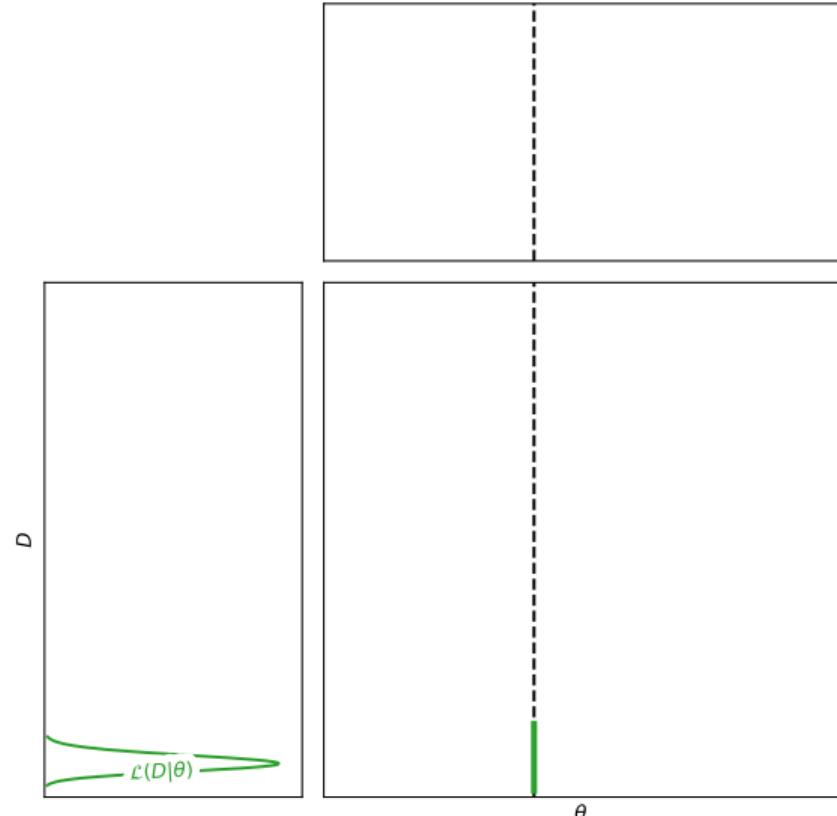
SBI: Simulation-based inference

- ▶ 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* $\mathcal{Z}(D)$ or even *likelihood* $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



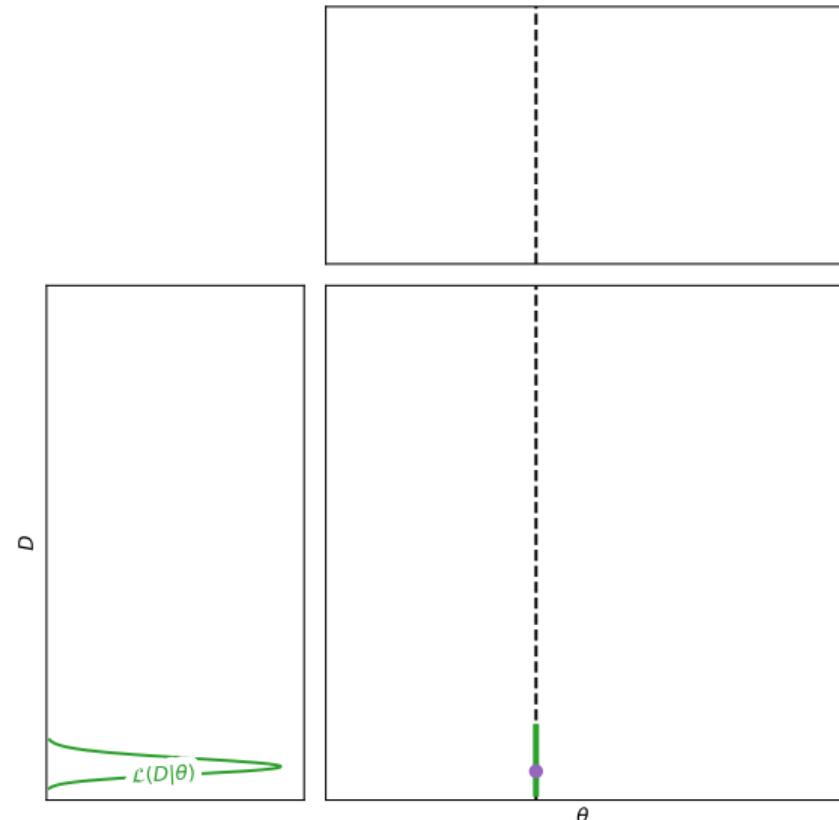
SBI: Simulation-based inference

- ▶ 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* $\mathcal{Z}(D)$ or even *likelihood* $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



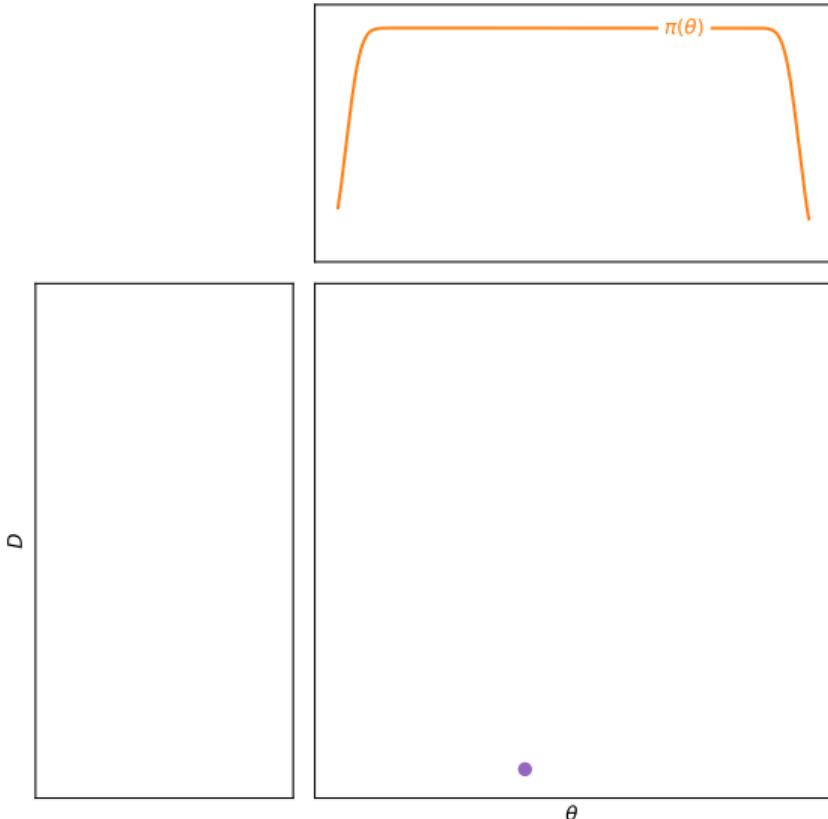
SBI: Simulation-based inference

- ▶ 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* $\mathcal{Z}(D)$ or even *likelihood* $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



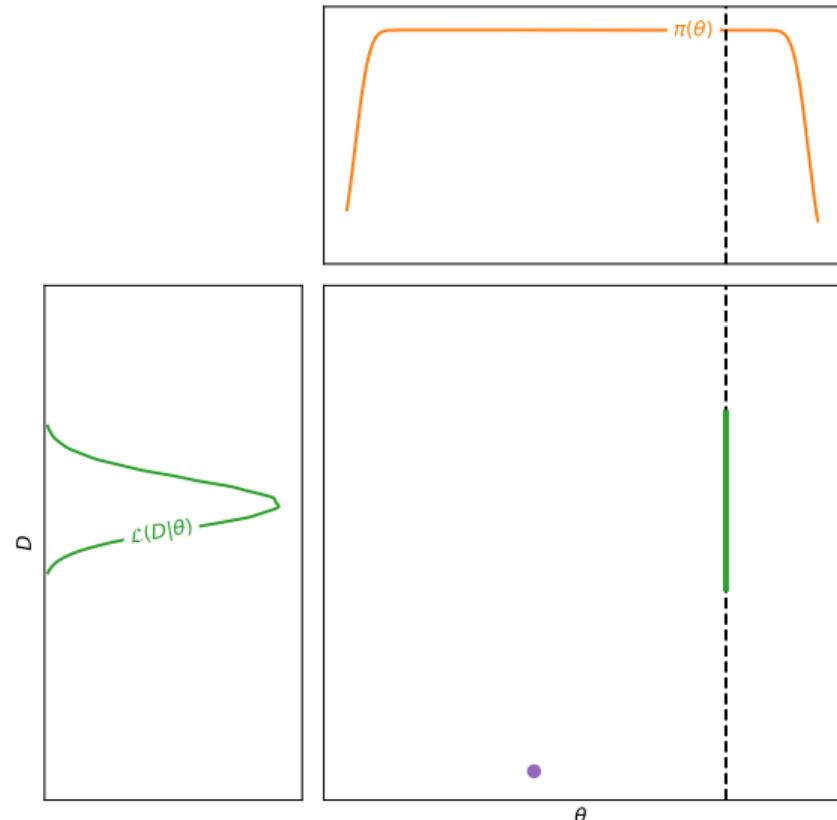
SBI: Simulation-based inference

- ▶ 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* $\mathcal{Z}(D)$ or even *likelihood* $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



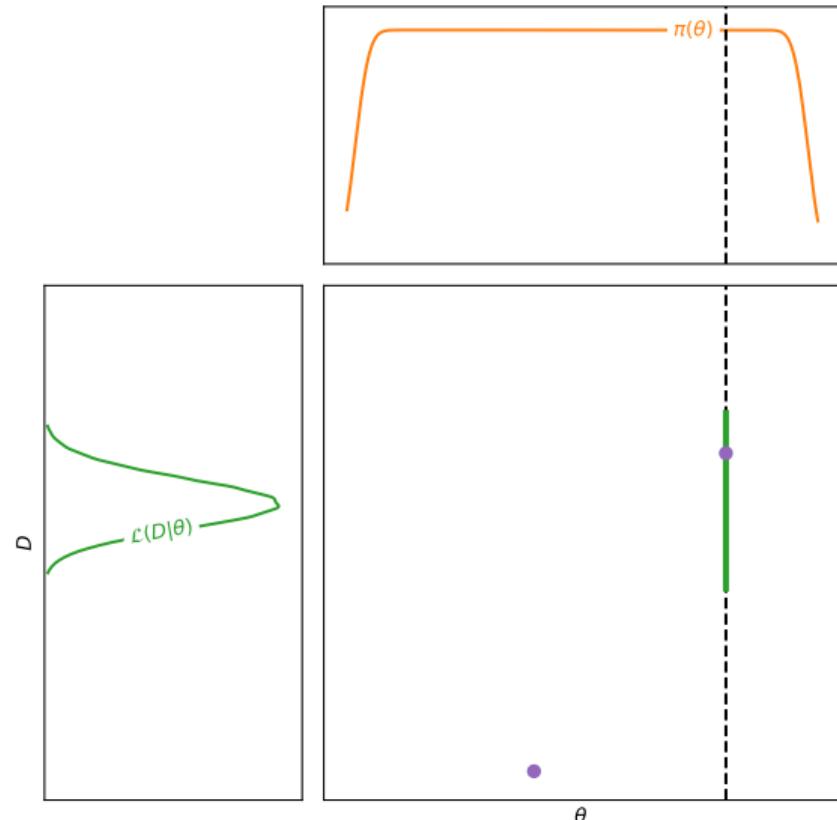
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



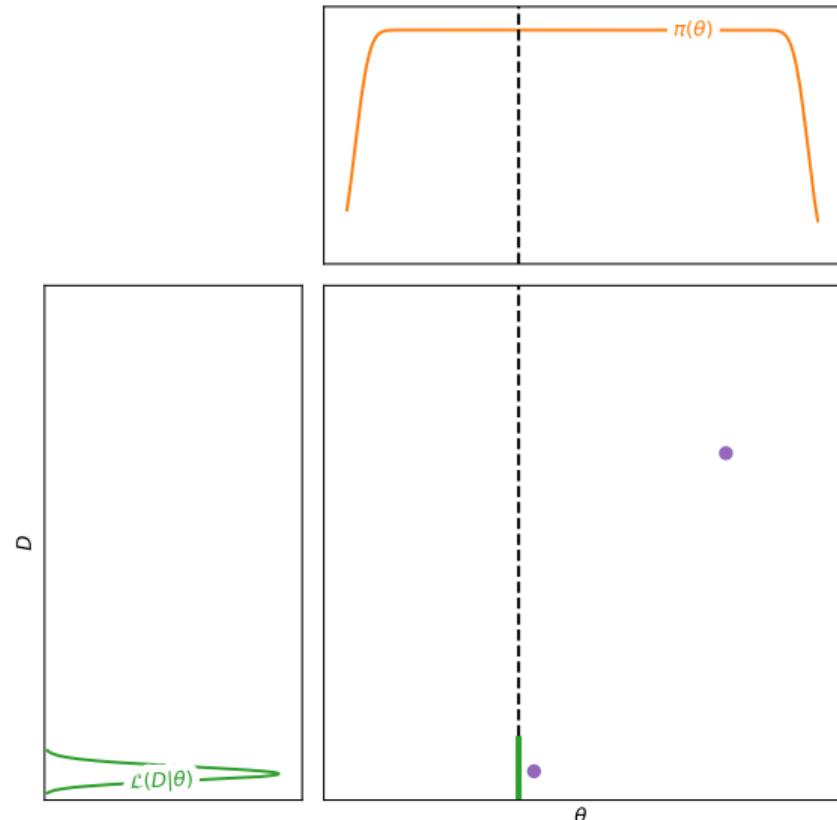
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



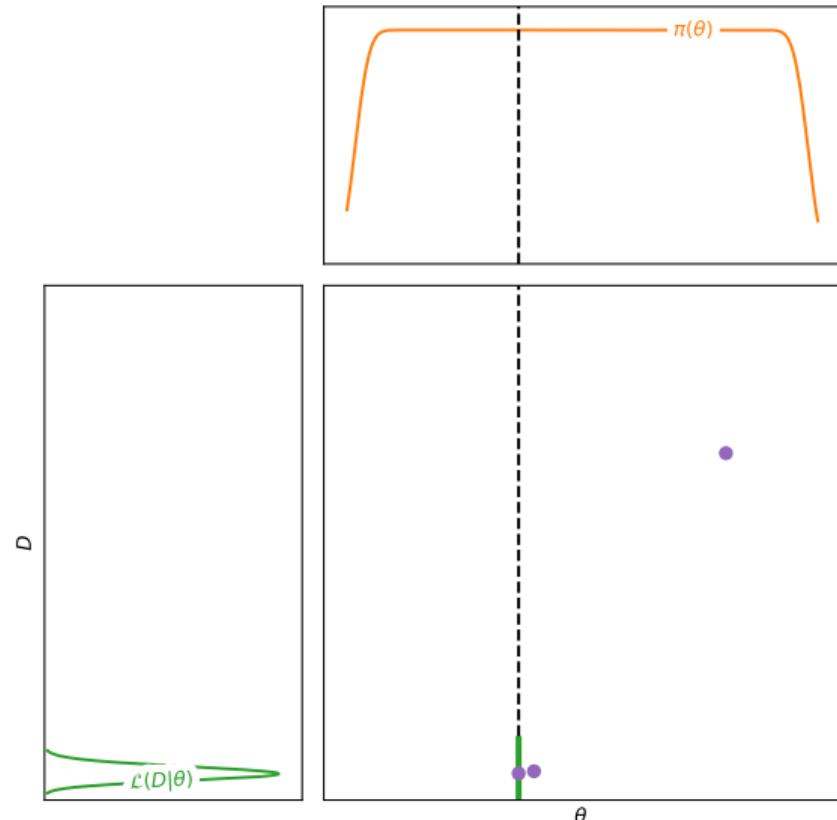
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



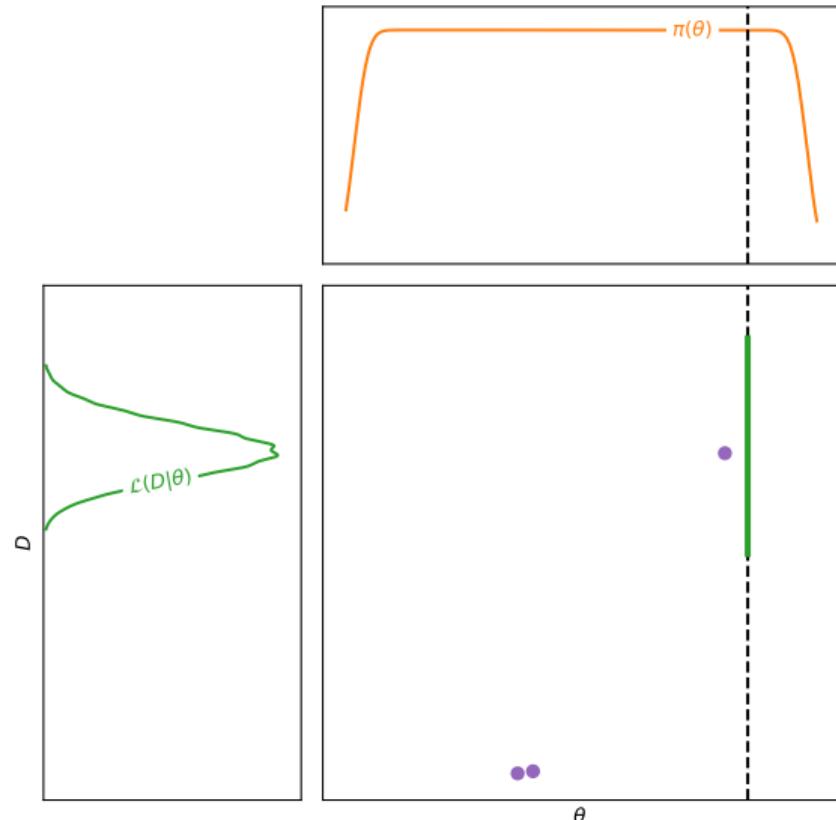
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



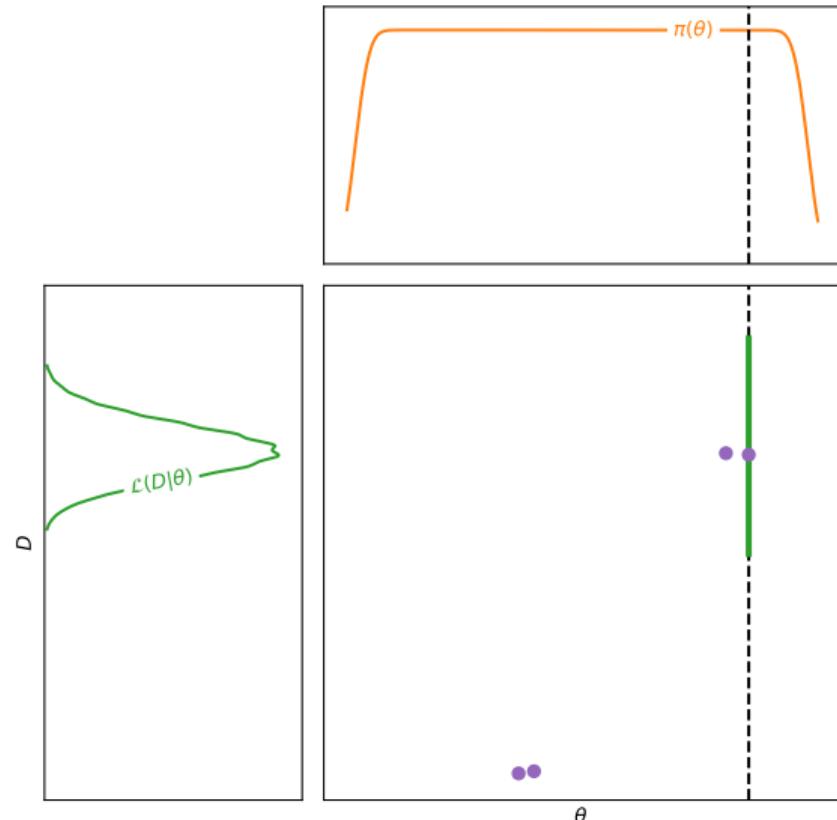
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



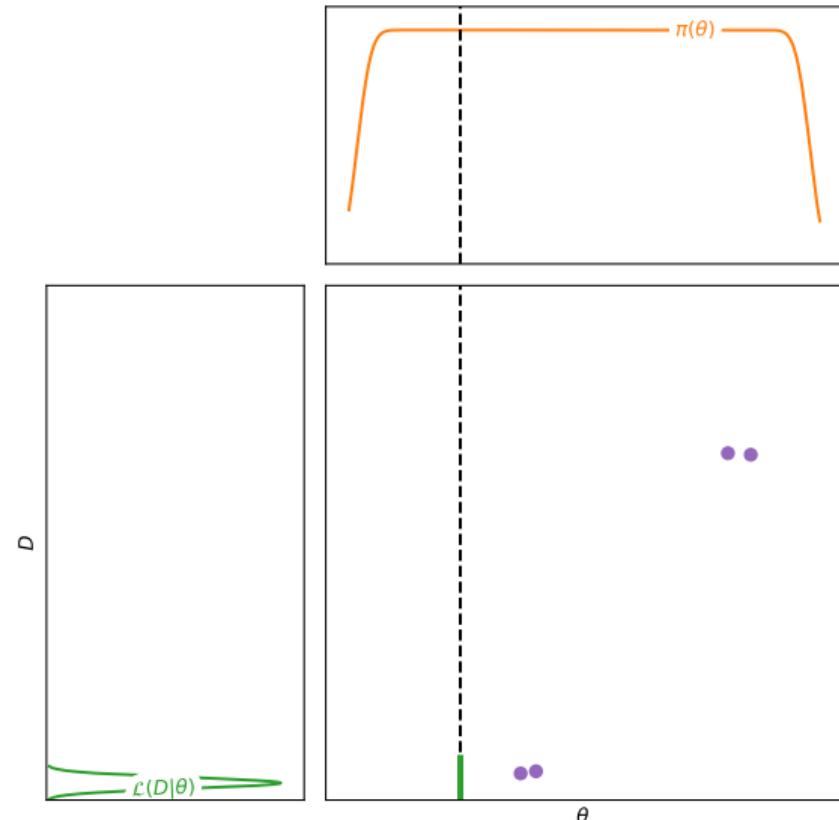
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



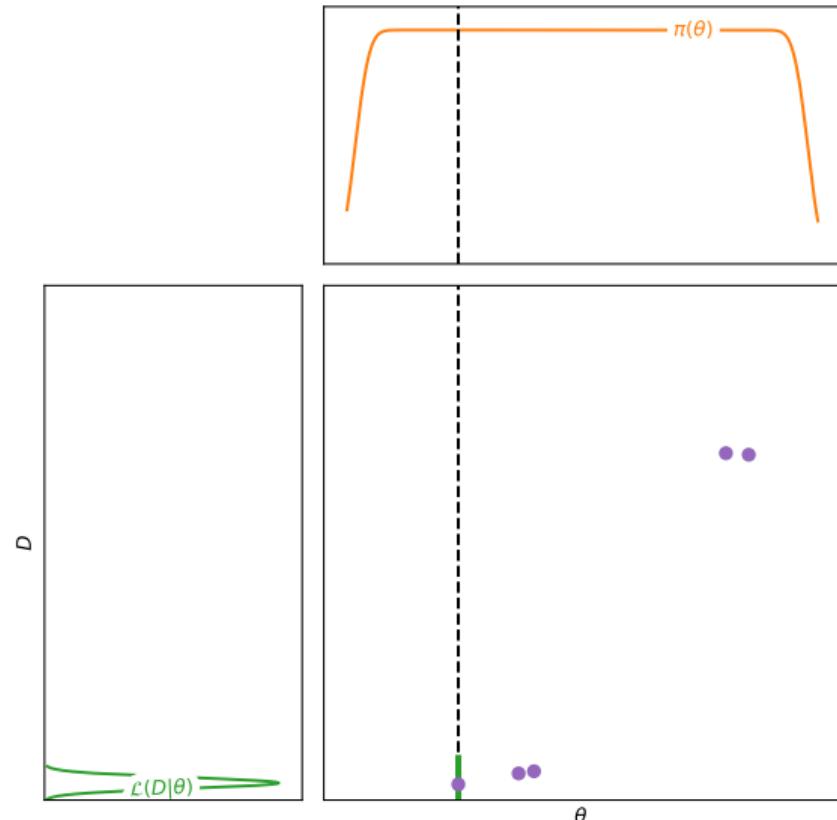
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



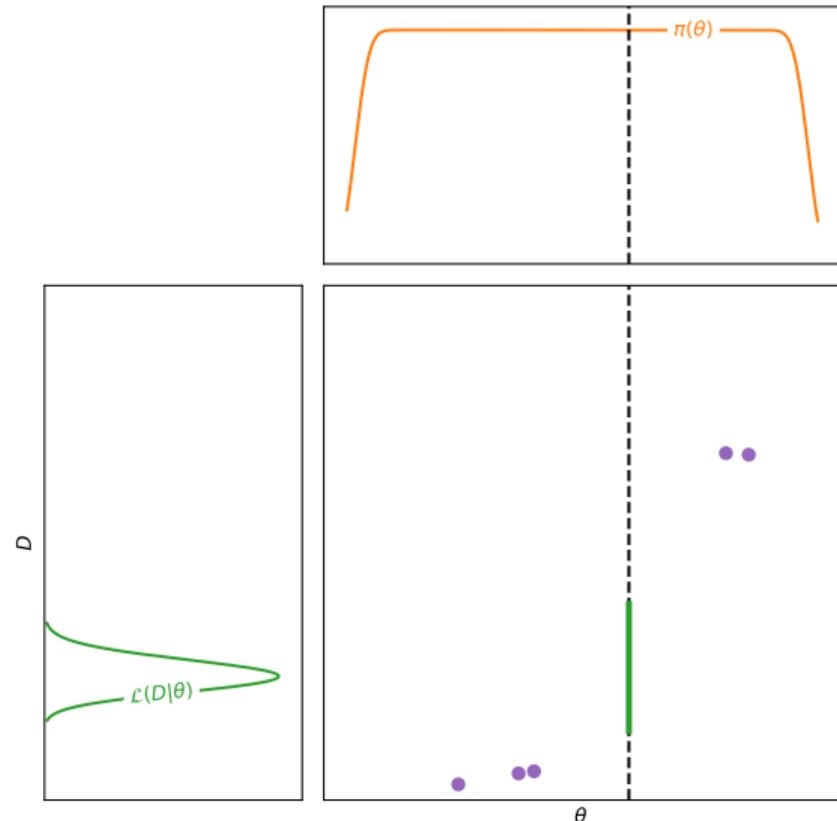
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



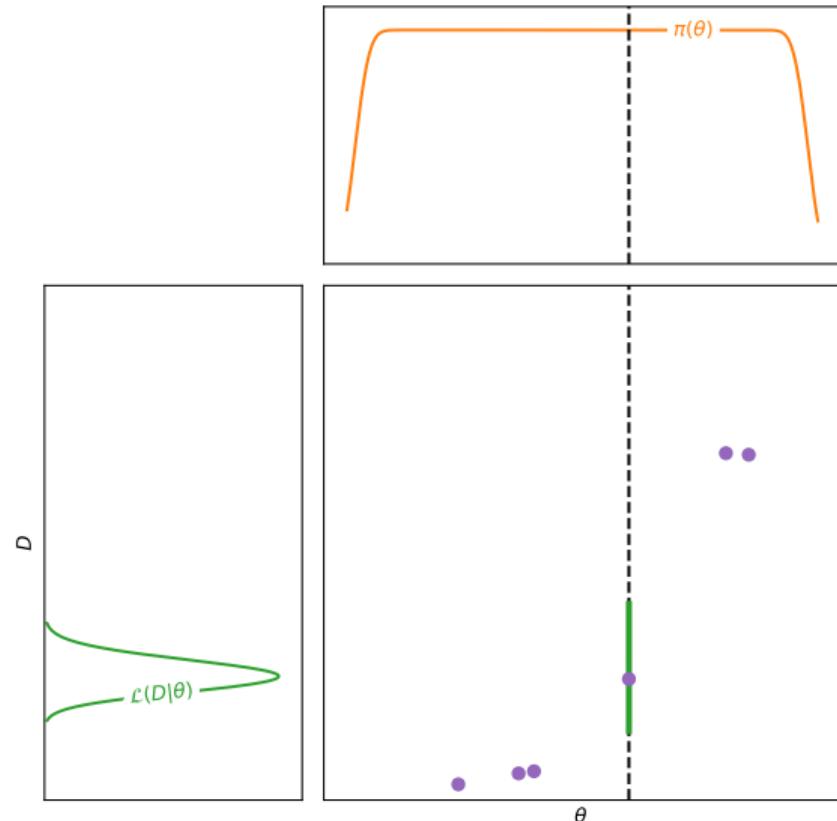
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



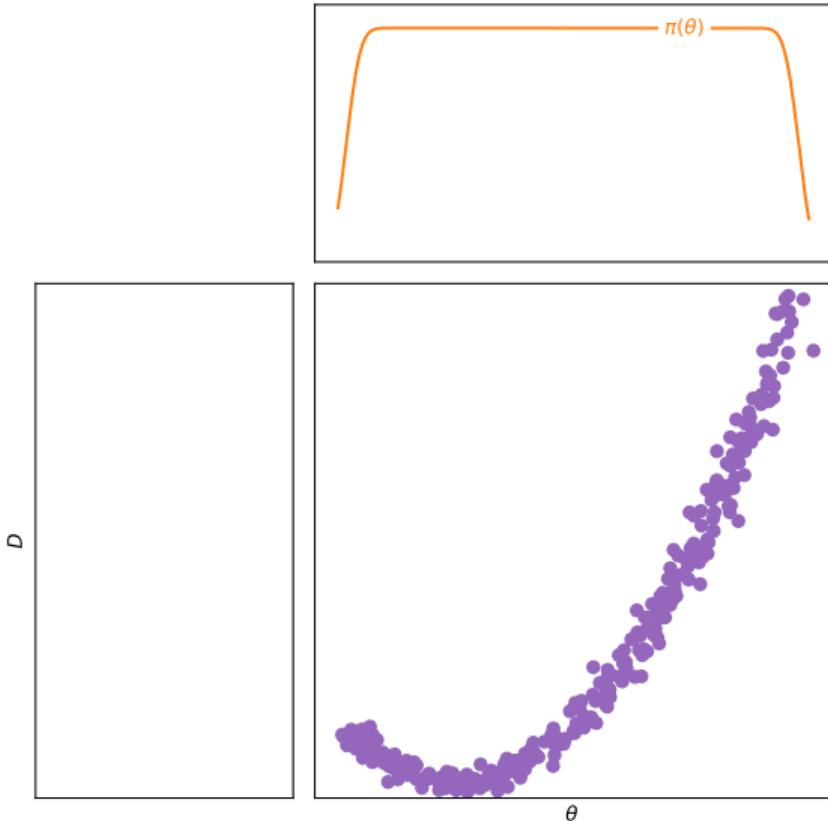
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



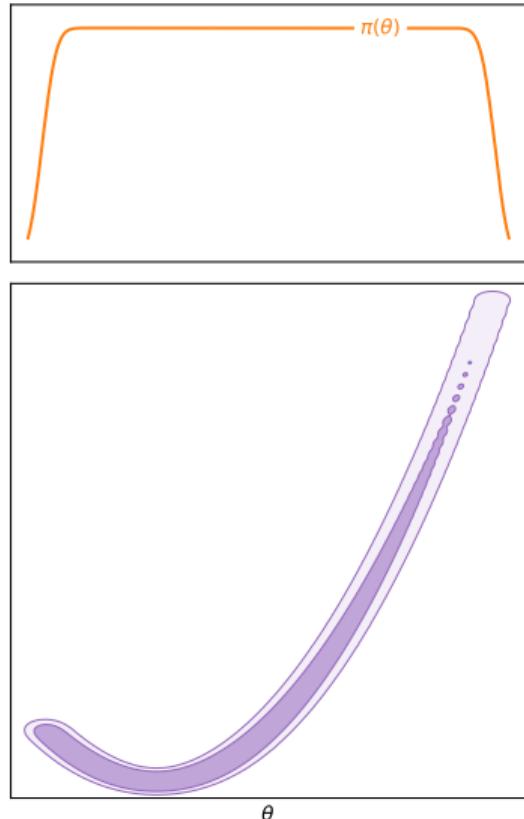
SBI: Simulation-based inference

- ▶ 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* $\mathcal{Z}(D)$ or even *likelihood* $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



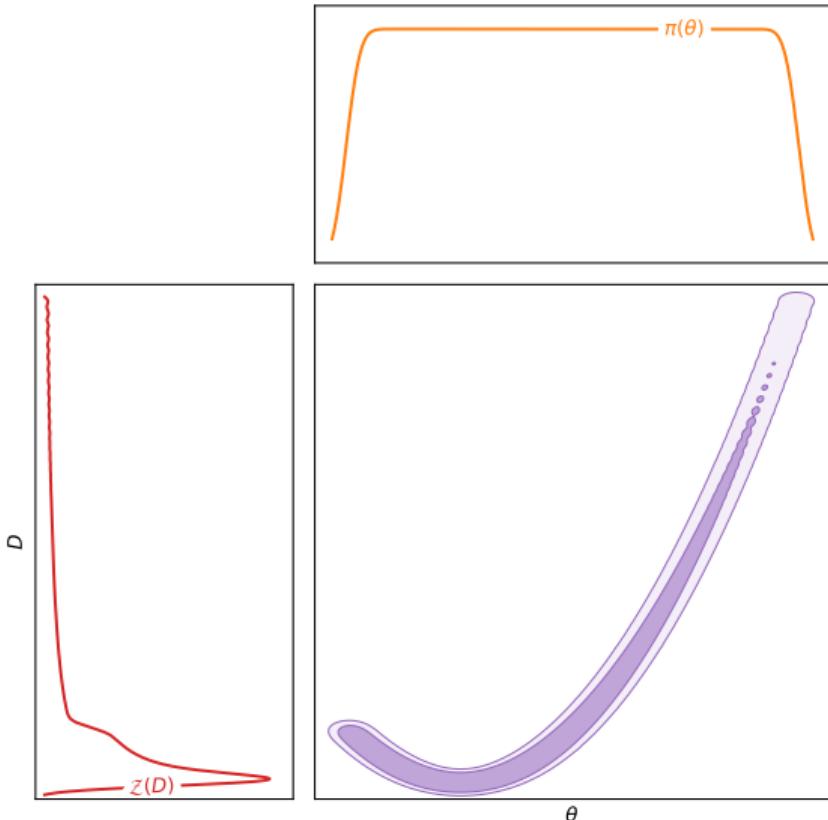
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



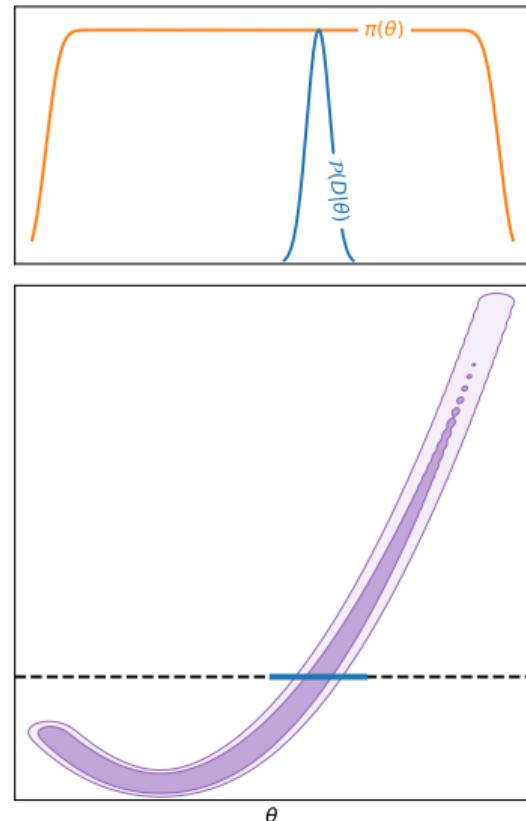
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



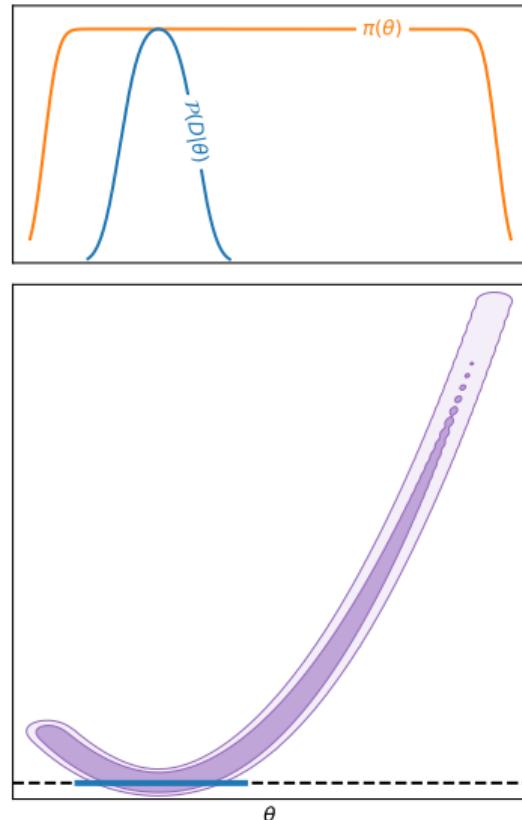
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



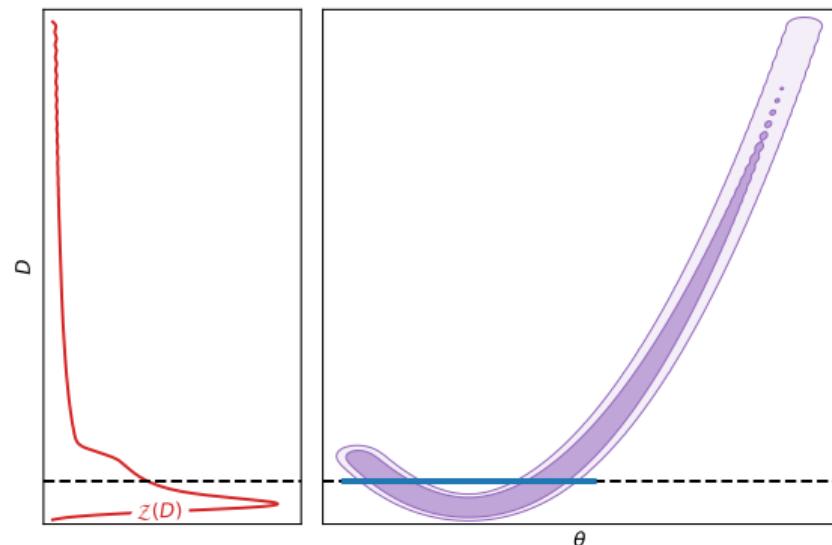
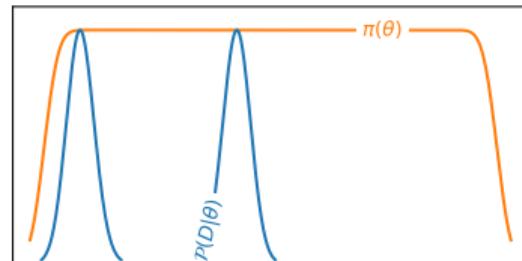
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



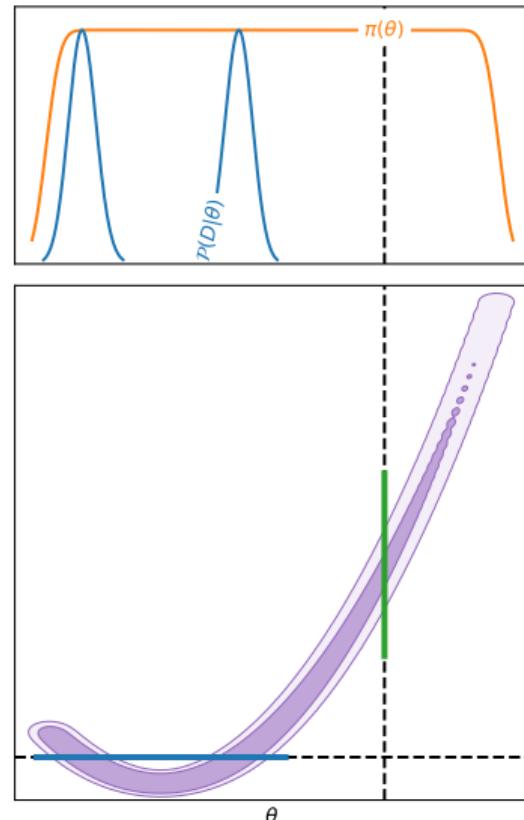
SBI: Simulation-based inference

- ▶ 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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



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 $\mathcal{Z}(D)$ or even likelihood $\mathcal{L}(D|\theta)$ or joint $\mathcal{J}(\theta, D)$.
- ▶ Present state of the art achieves this using *machine learning* (neural networks).



Why linear SBI?

If neural networks are all that, why should we consider the regressive step of going back to linear versions of this problem?

- ▶ It is pedagogically helpful
 - ▶ separates general principles of SBI from the details of neural networks
 - ▶ (particularly for ML skeptics)
- ▶ It is practically useful
 - ▶ for producing expressive examples with known ground truths.
- ▶ It is pragmatically useful
 - ▶ competitive with neural approaches in terms of accuracy,
 - ▶ faster and more interpretable.

Linear Simulation Based Inference

Mathematical setup

- ▶ Linear generative model (m, M, C)

$$D = m + M\theta \pm \sqrt{C}$$

where:

θ : n dimensional parameters

D : d dimensional data

M : $d \times n$ transfer matrix

m : d -dimensional shift

C : $d \times d$ data covariance

¹N.B. using matrix variate notation where primes denote transposes $M' = M^T$

Linear Simulation Based Inference

Mathematical setup

- ▶ Linear generative model (m, M, C)

$$D \sim \mathcal{N}(m + M\theta, C)$$

where:

- θ : n dimensional parameters
- D : d dimensional data
- M : $d \times n$ transfer matrix
- m : d -dimensional shift
- C : $d \times d$ data covariance

- ▶ k Simulations

$$S = \{(\theta_i, D_i) : i = 1, \dots, k\}$$

- ▶ Define simulation statistics¹:

$$\begin{aligned}\bar{\theta} &= \frac{1}{k} \sum_k \theta_i \\ \bar{D} &= \frac{1}{k} \sum_k D_i \\ \Theta &= \frac{1}{k-1} \sum_i (\theta_i - \bar{\theta})(\theta_i - \bar{\theta})' \\ \Delta &= \frac{1}{k-1} \sum_i (D_i - \bar{D})(D_i - \bar{D})' \\ \Psi &= \frac{1}{k-1} \sum_i (D_i - \bar{D})(\theta_i - \bar{\theta})'\end{aligned}$$

¹N.B. using matrix variate notation where primes denote transposes $M' = M^T$

Linear Simulation Based Inference: gory mathematical details

- ▶ We now wish to infer the parameters of the linear model (m, M, C) from simulations S (which define $\bar{\theta}, \bar{D}, \Theta, \Delta, \Psi$)
- ▶ The likelihood for this problem is:

$$P(\{D_i\}|\{\theta_i\}|m, M, C) = \prod_i \mathcal{N}(D_i|m + M\theta_i, C)$$

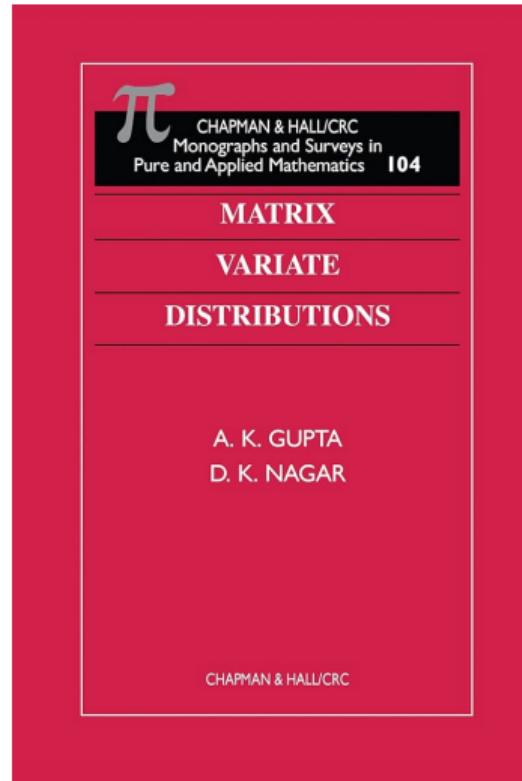
- ▶ It can be shown the posterior \mathcal{P} is...

Linear Simulation Based Inference: gory mathematical details

- ▶ We now wish to infer the parameters of the linear model (m, M, C) from simulations S (which define $\bar{\theta}, \bar{D}, \Theta, \Delta, \Psi$)
- ▶ The likelihood for this problem is:

$$P(\{D_i\}|\{\theta_i\}|m, M, C) = \prod_i \mathcal{N}(D_i|m + M\theta_i, C)$$

- ▶ It can be shown the posterior \mathcal{P} is...



Linear Simulation Based Inference: gory mathematical details

- ▶ We now wish to infer the parameters of the linear model (m, M, C) from simulations S (which define $\bar{\theta}, \bar{D}, \Theta, \Delta, \Psi$)
- ▶ The likelihood for this problem is:

$$P(\{D_i\}|\{\theta_i\}|m, M, C) = \prod_i \mathcal{N}(D_i|m + M\theta_i, C)$$

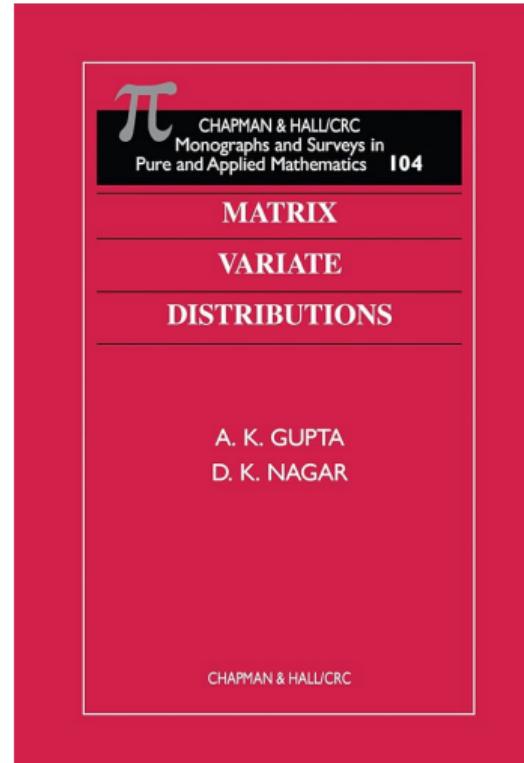
- ▶ It can be shown the posterior \mathcal{P} is...

$$m|M, C, S \sim \mathcal{N}\left(\frac{k}{k+1}(\bar{D} - M\bar{\theta}), \frac{C}{k+1}\right),$$

$$M|C, S \sim \mathcal{MN}\left(\Psi\Theta_*^{-1}, \frac{C}{k-1}, \Theta_*^{-1}\right),$$

$$C|S \sim \mathcal{W}_\nu^{-1}(C_0 + (k-1)(\Delta - \Psi\Theta^{-1}\Psi')),$$

where $\Theta_* = \frac{1}{k-1}\Theta_0 + \Theta$, $\nu = \nu_0 + k$, and C_0 define conjugate prior π on m, M, C

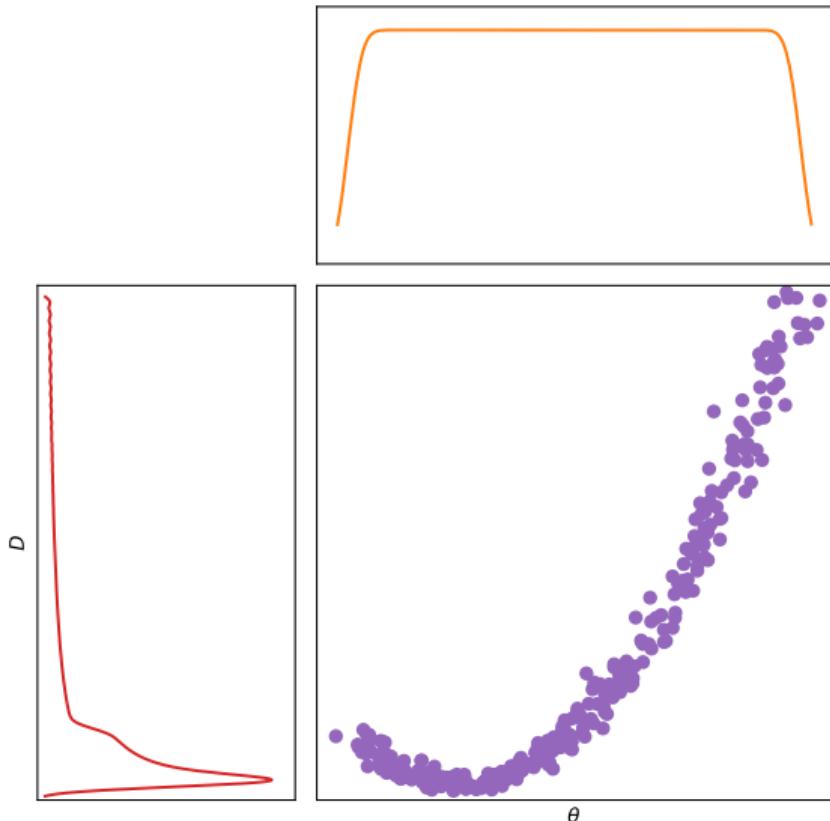


Sequential LSBI

- ▶ As we shall see, for non-linear problems, a linear approximation is unlikely to be a good one.
- ▶ Sequential methods iteratively improve by focussing effort around observed data D_{obs} .
 - ▶ This is orthogonal to amortised approaches
 - ▶ More appropriate to cosmology, where there is only one dataset
 - ▶ Less appropriate to particle physics/GW
- ▶ We are free to choose where to place simulation parameters $\{\theta_i\}$, so it makes sense to choose these so that they generate simulations close to the observed data
- ▶ Our current approximation to the posterior is a natural choice.

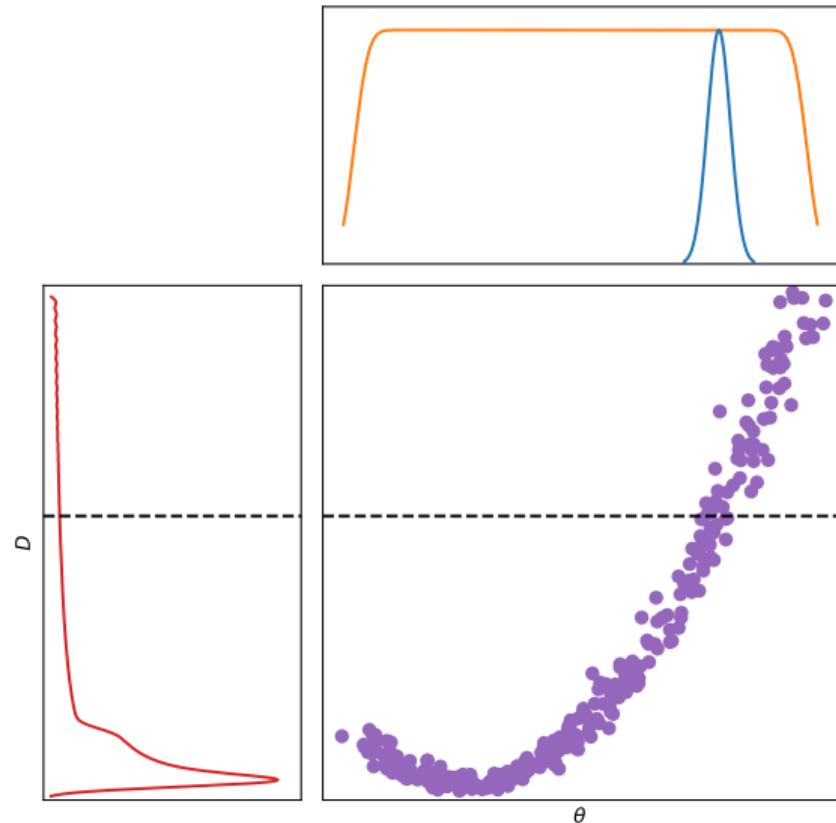
Example of this on our toy model

- ▶ Same model as before



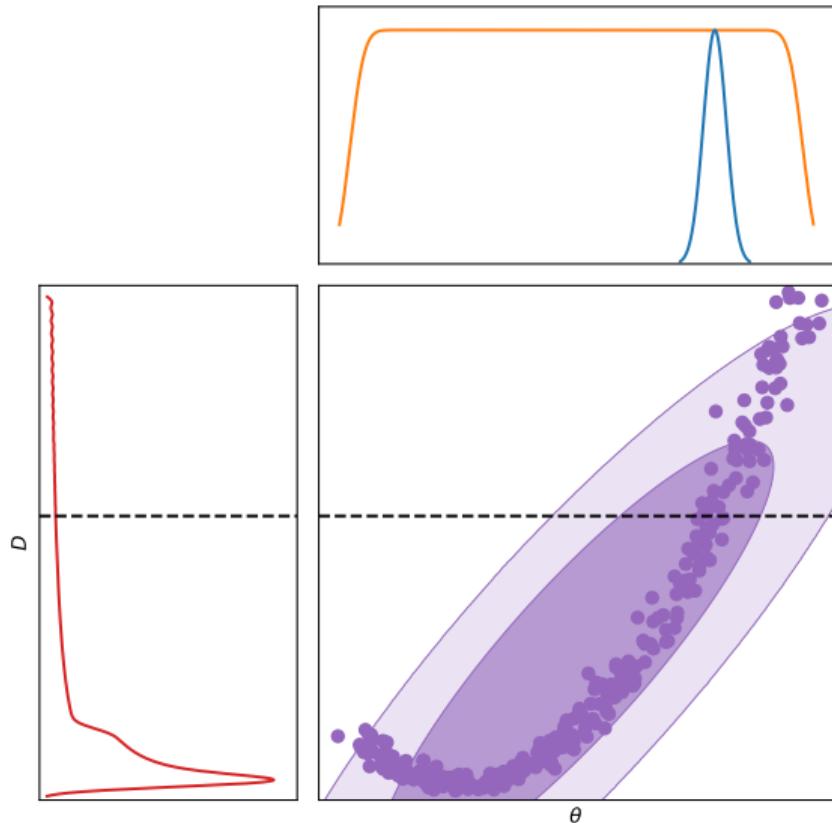
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}



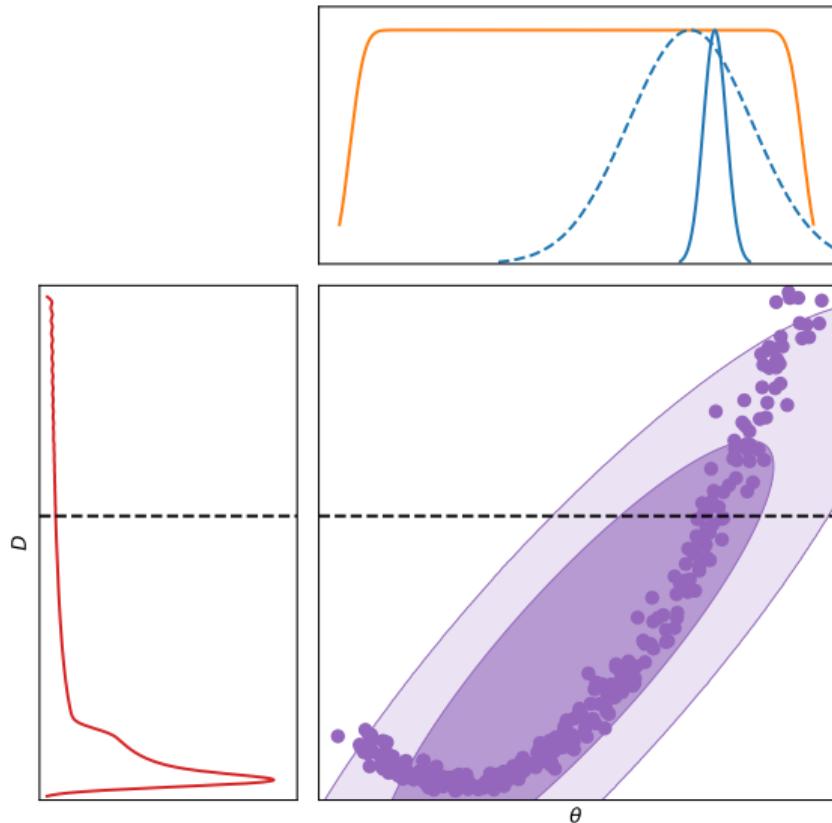
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi



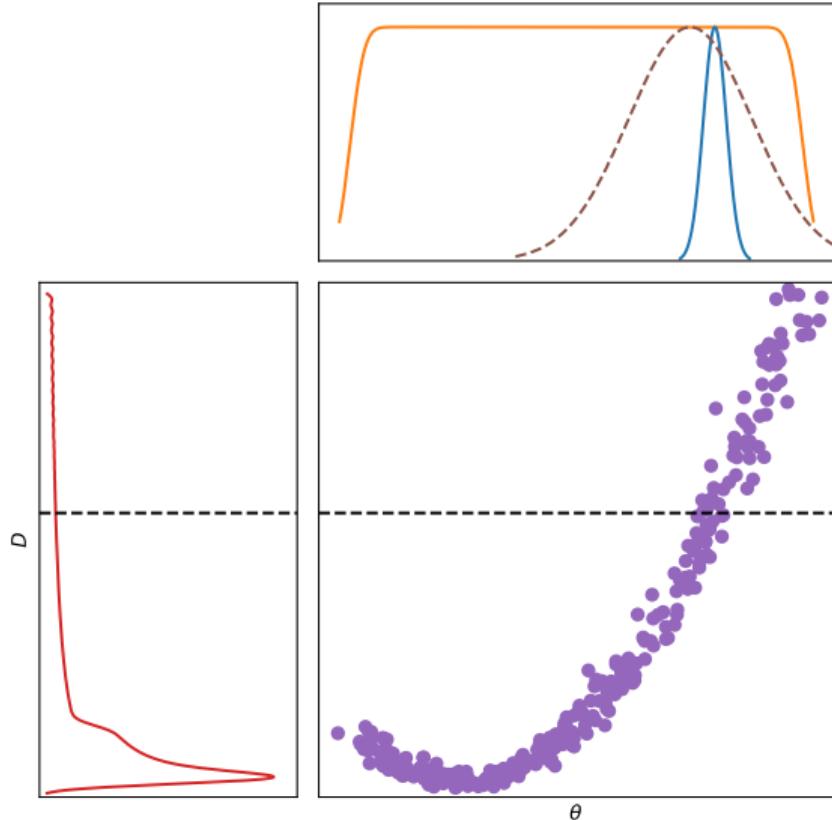
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)



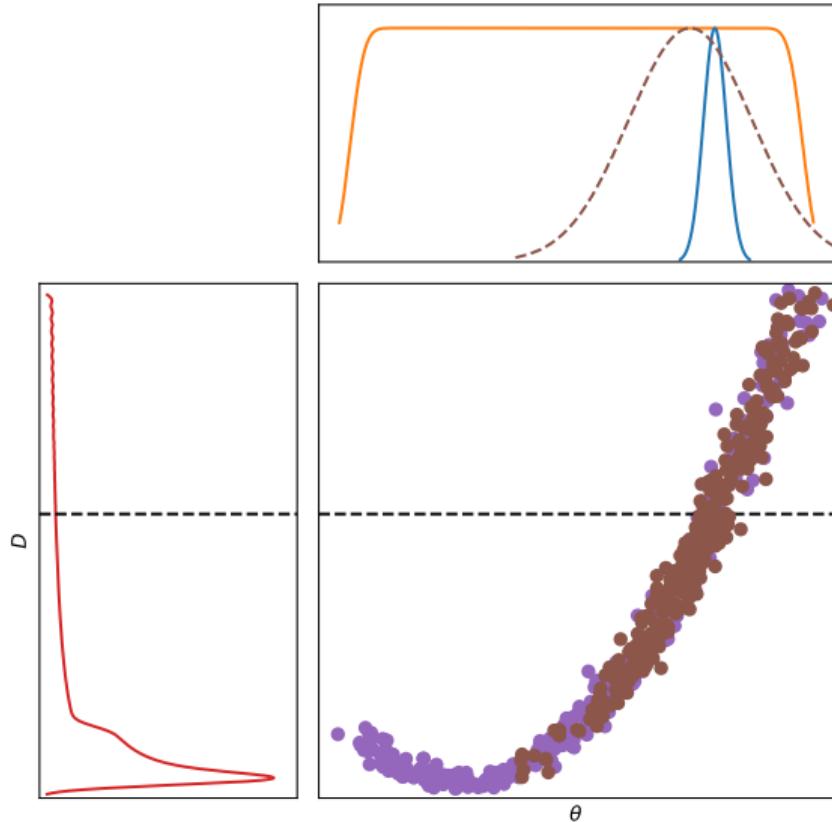
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator



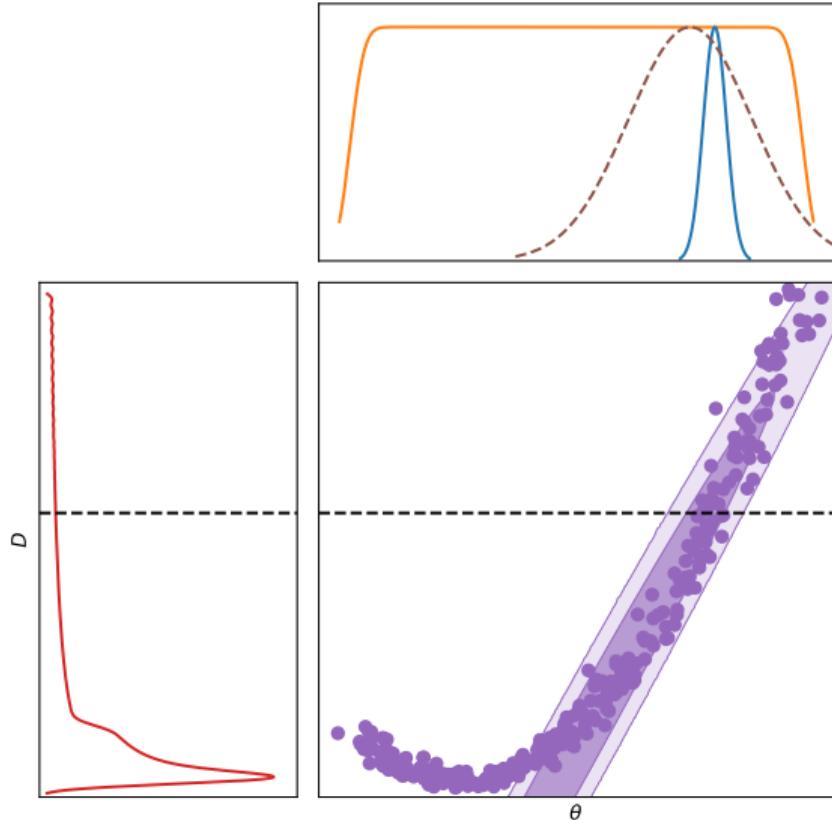
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these



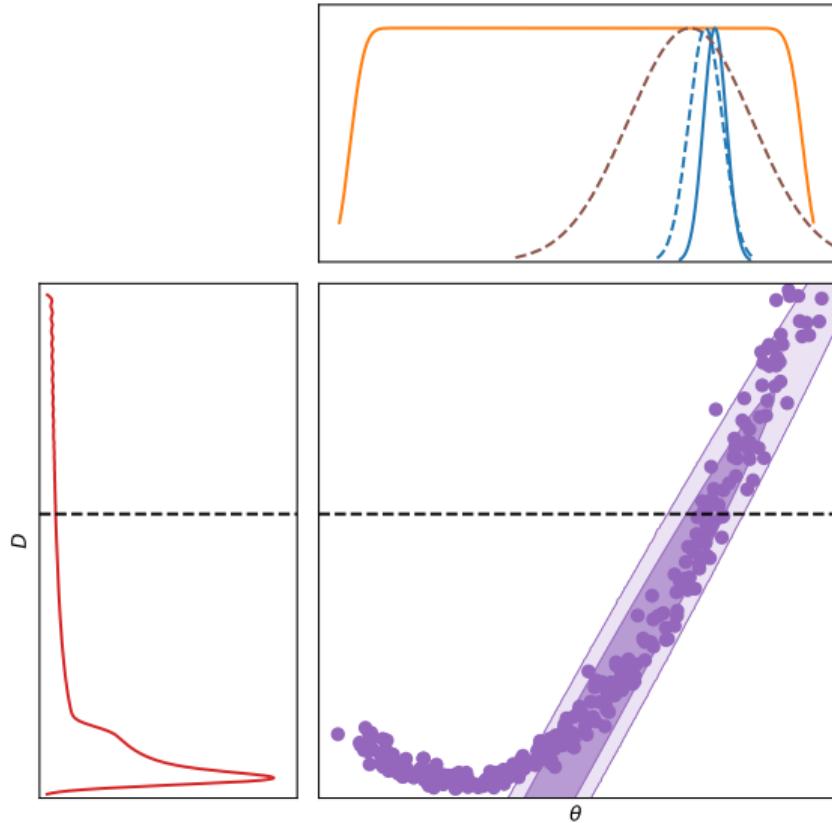
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior



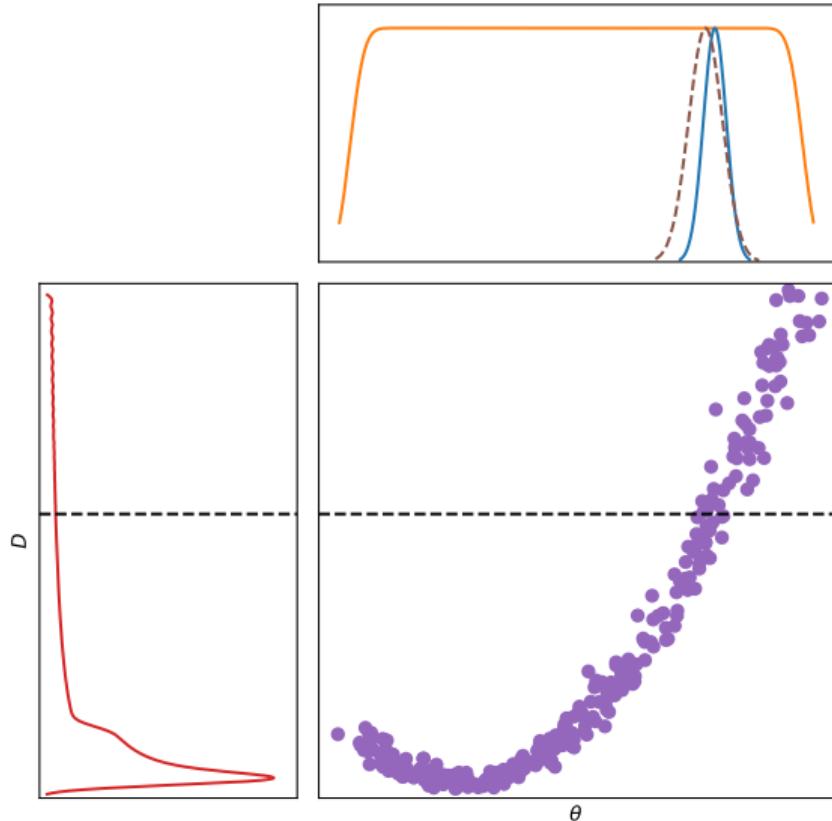
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



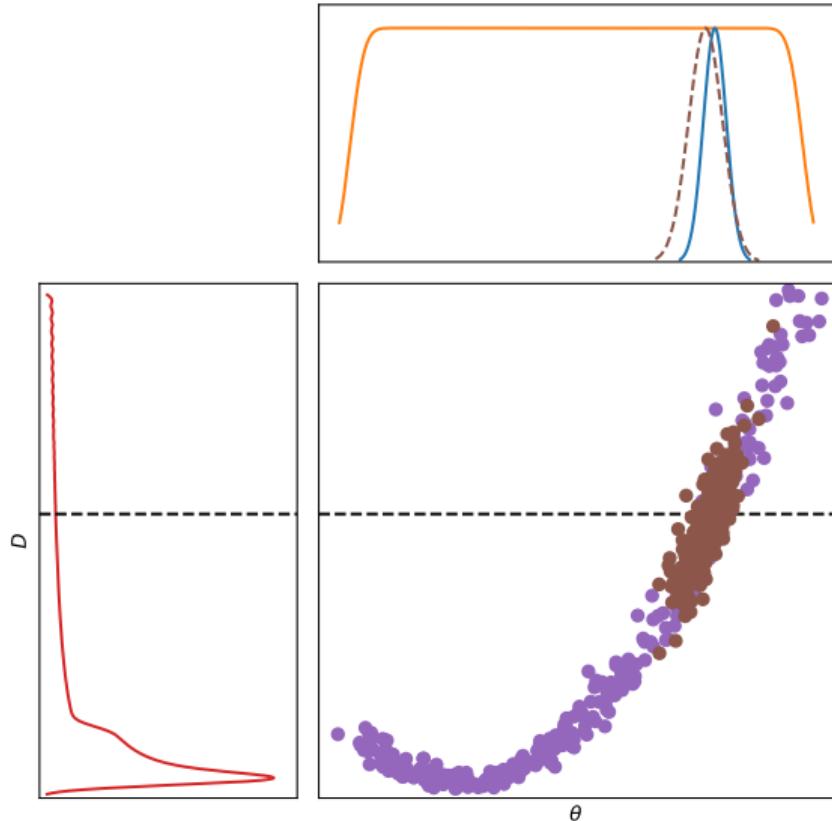
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



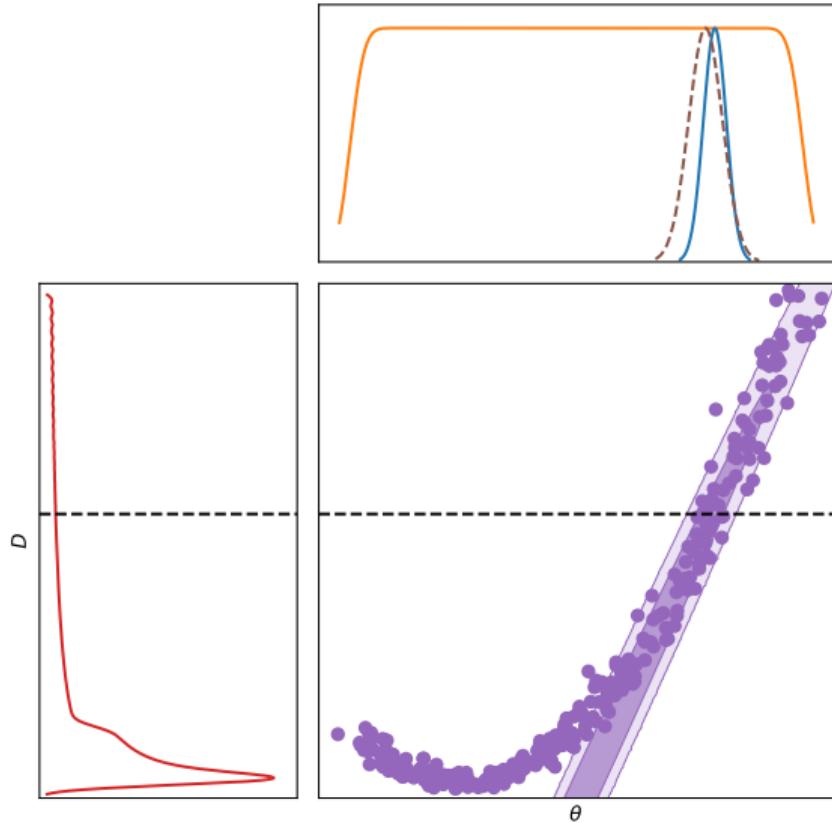
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



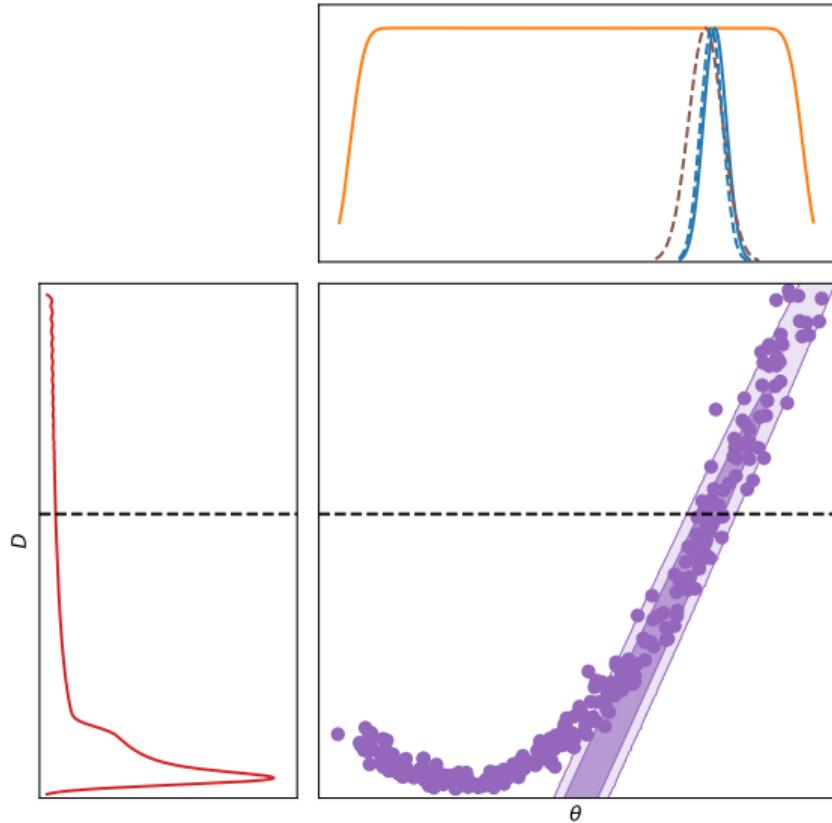
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



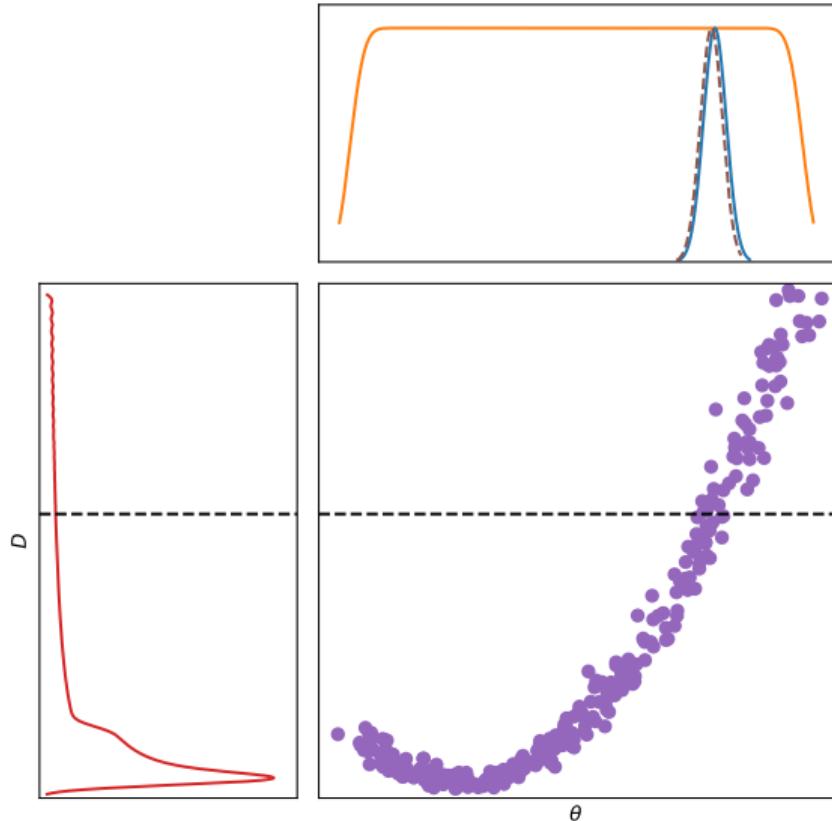
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



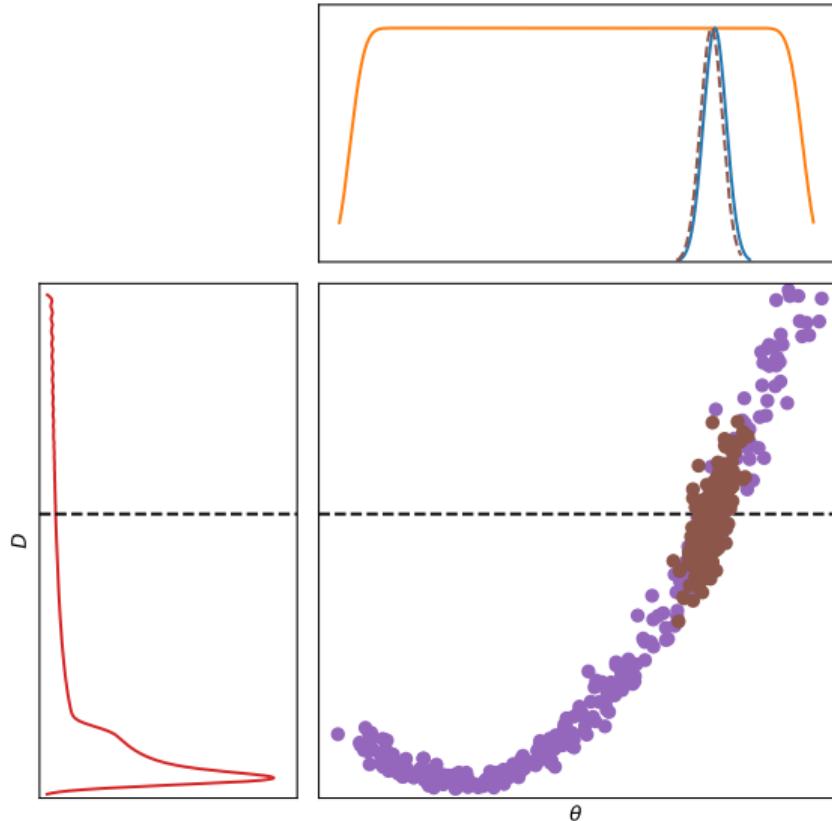
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



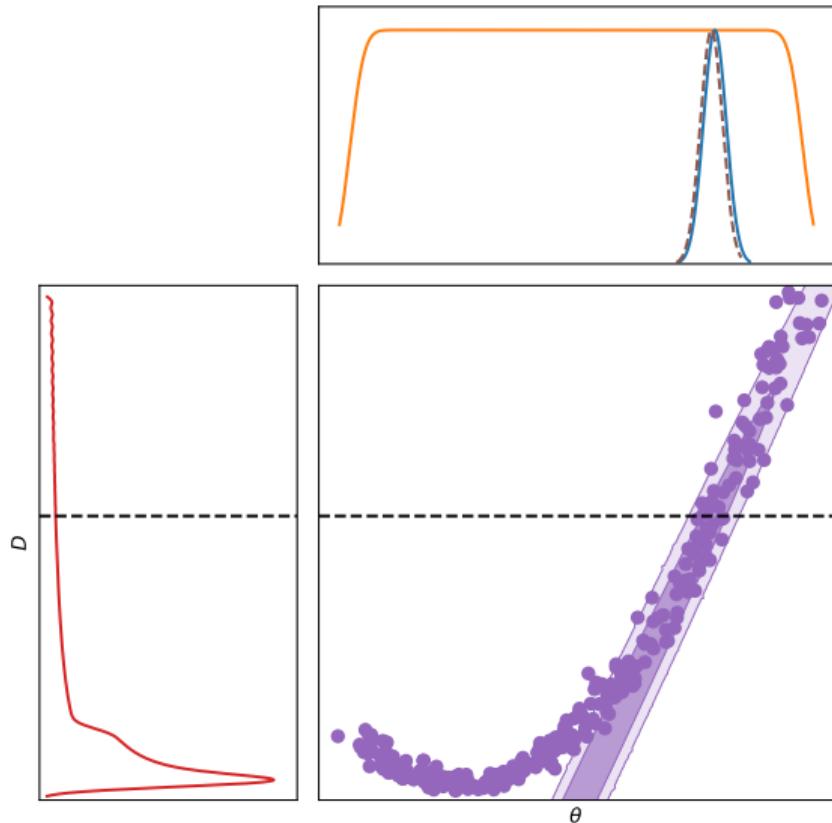
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



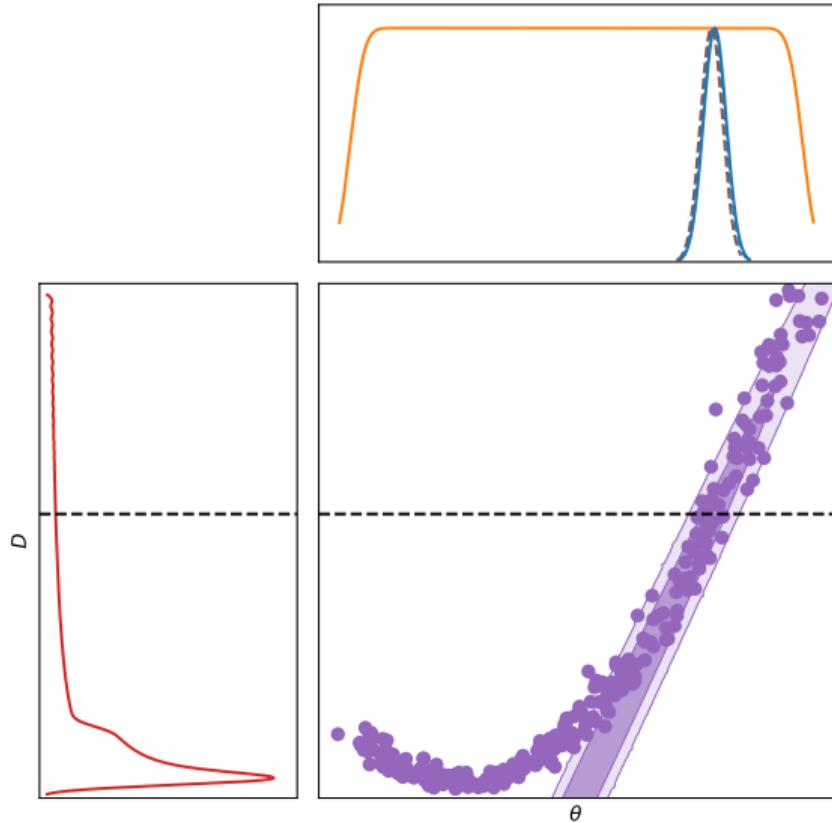
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



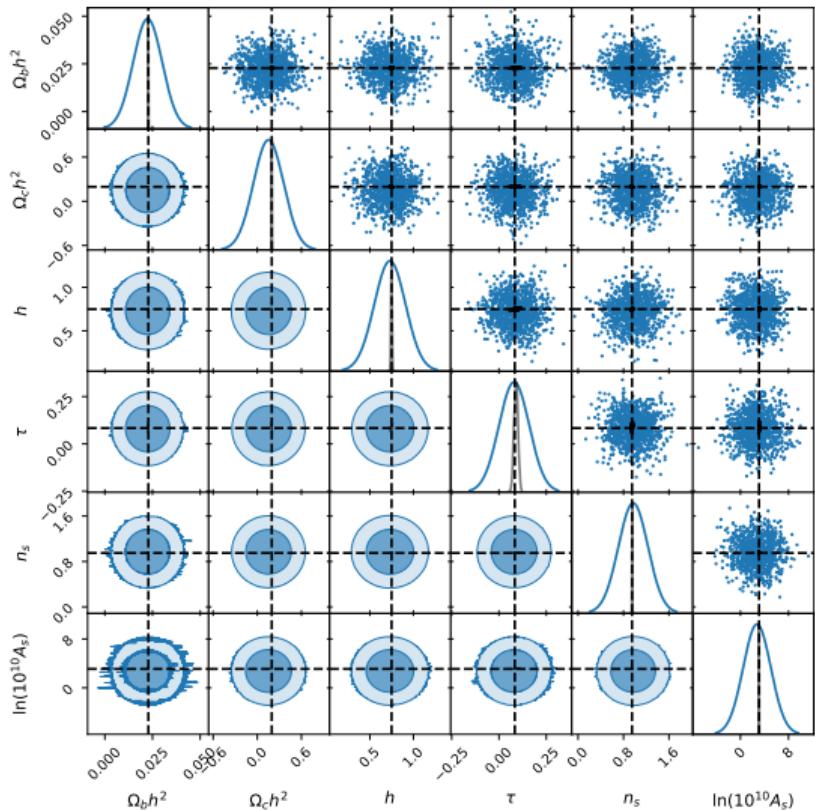
Example of this on our toy model

- ▶ Same model as before
- ▶ Mark the observed data D_{obs}
- ▶ Fit a model using lsbi
- ▶ Evaluate the posterior (cheap as linear)
- ▶ Now use this posterior to pick $\{\theta_i\}$
- ▶ Generate $\{D_i\}$ from original simulator
- ▶ Fit lsbi to these
- ▶ Evaluate the new posterior
- ▶ Iterate



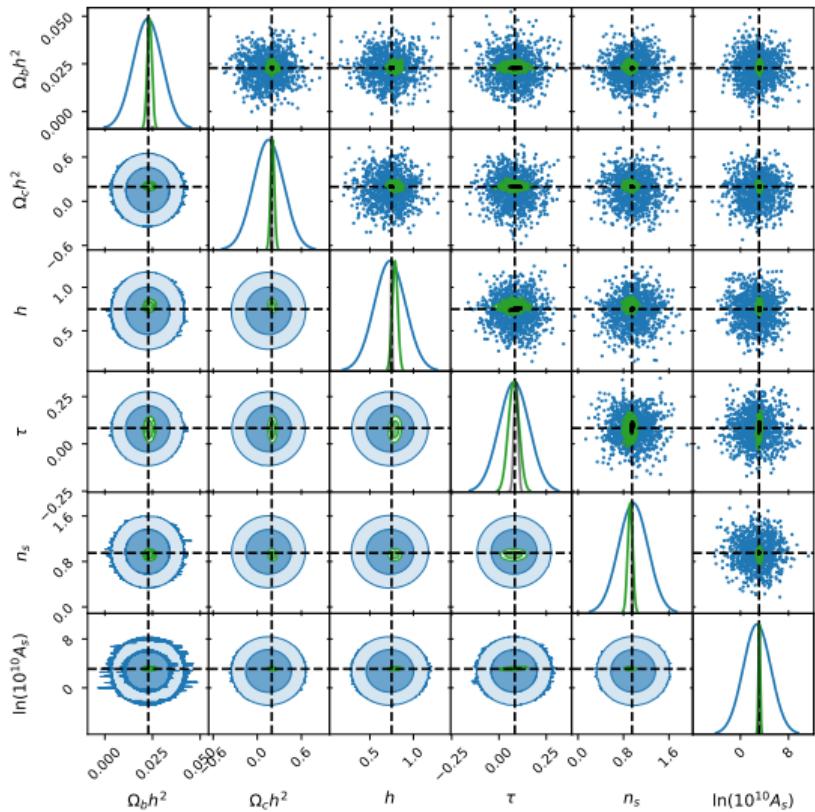
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



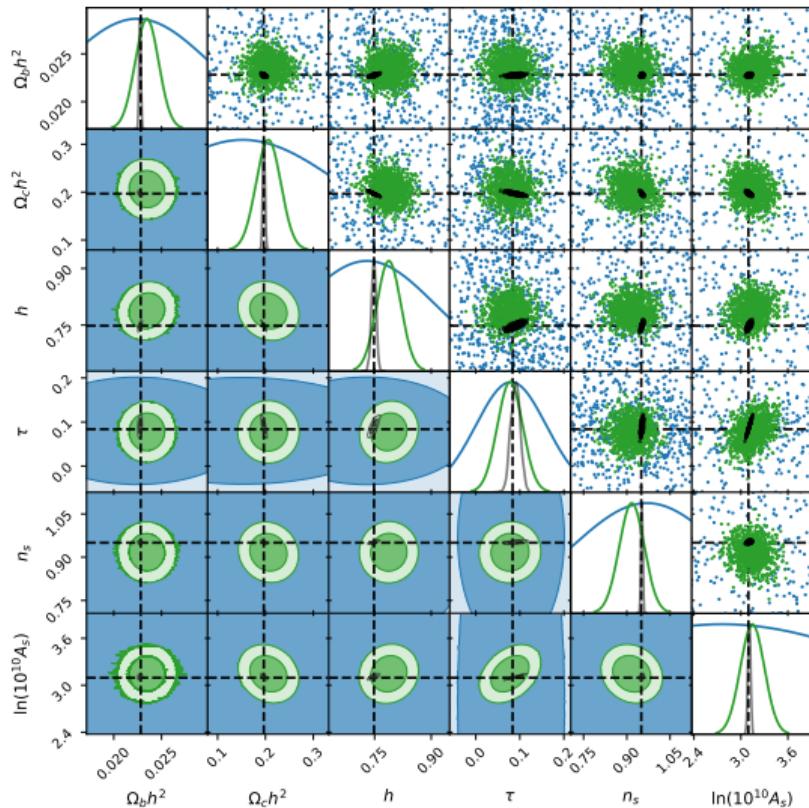
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



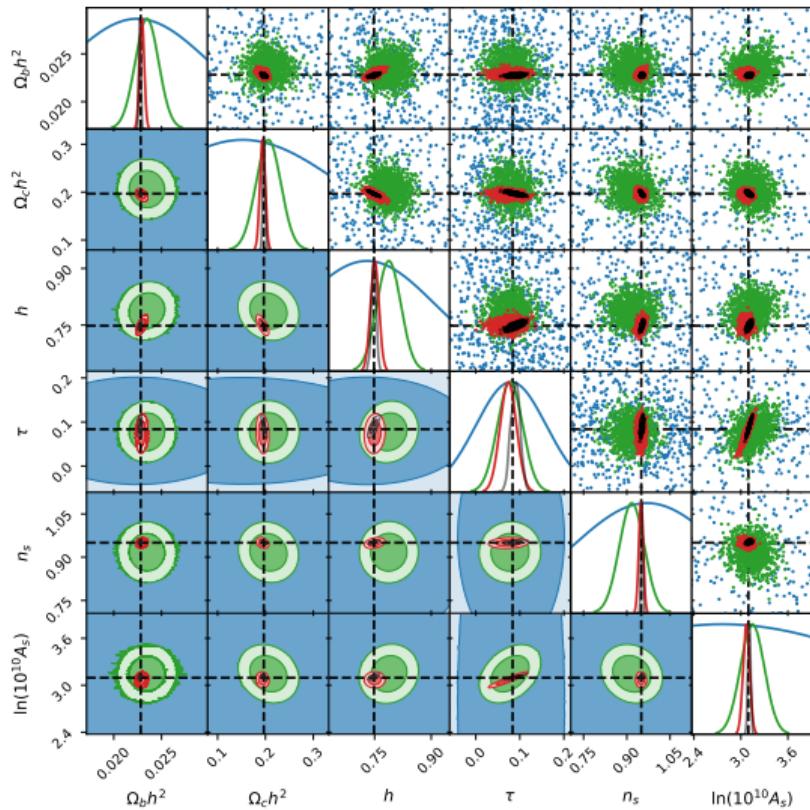
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



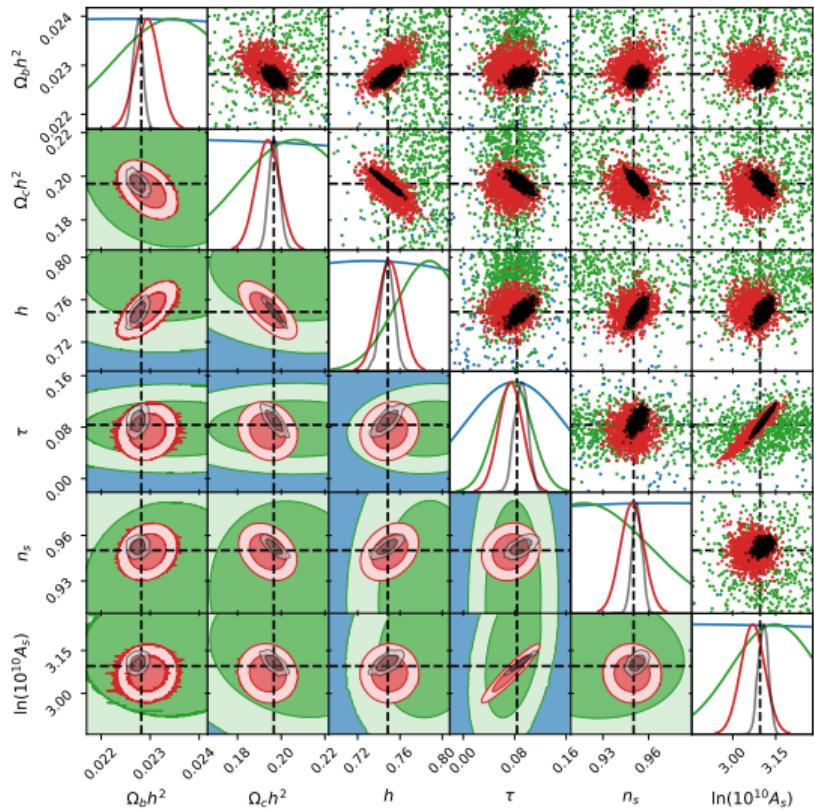
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



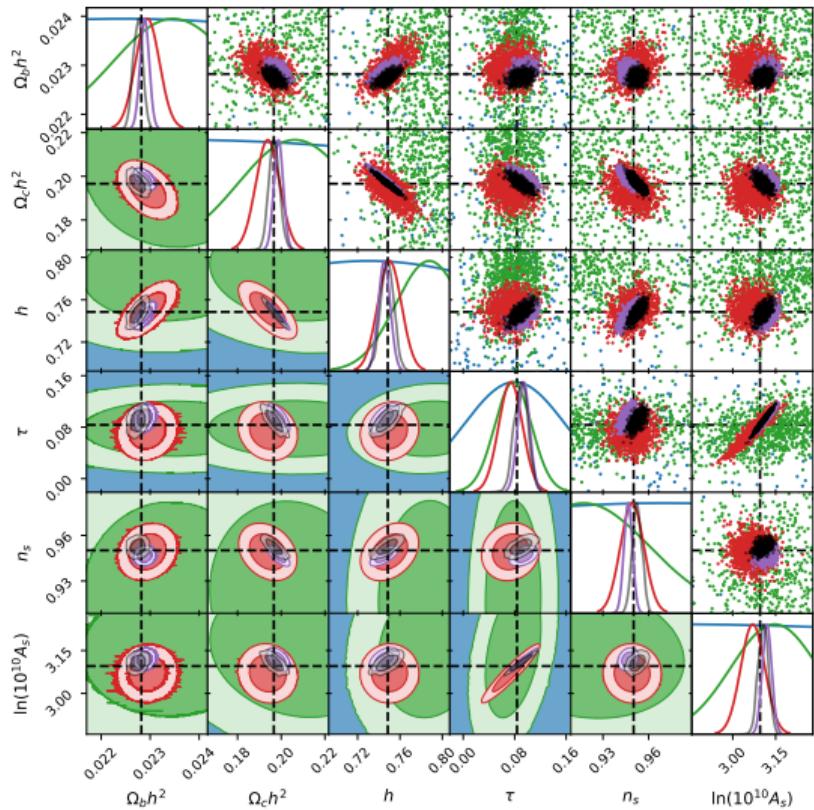
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



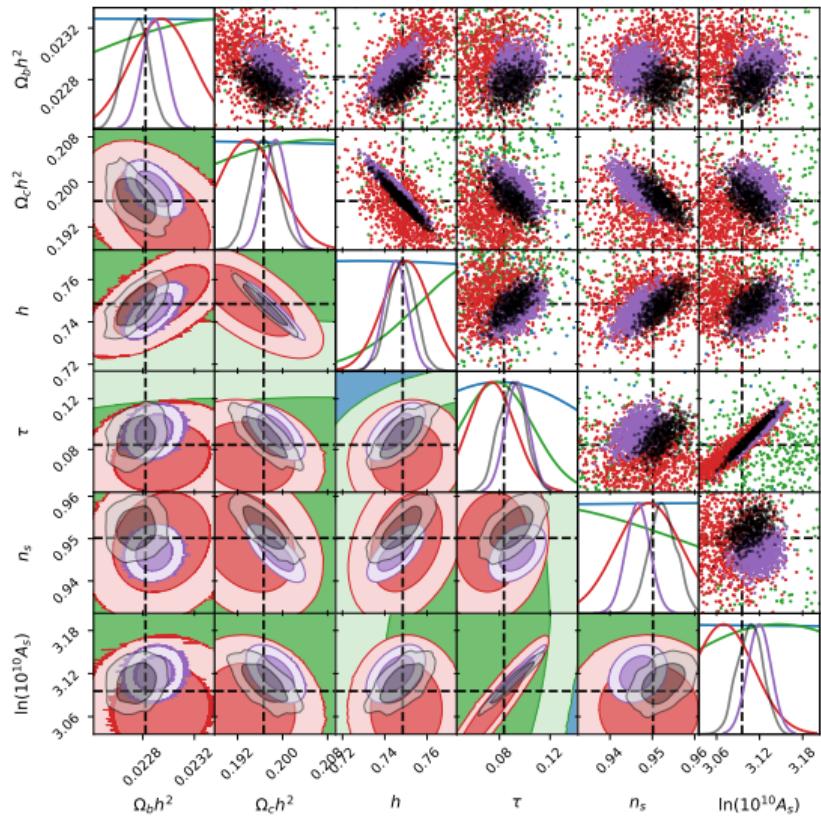
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



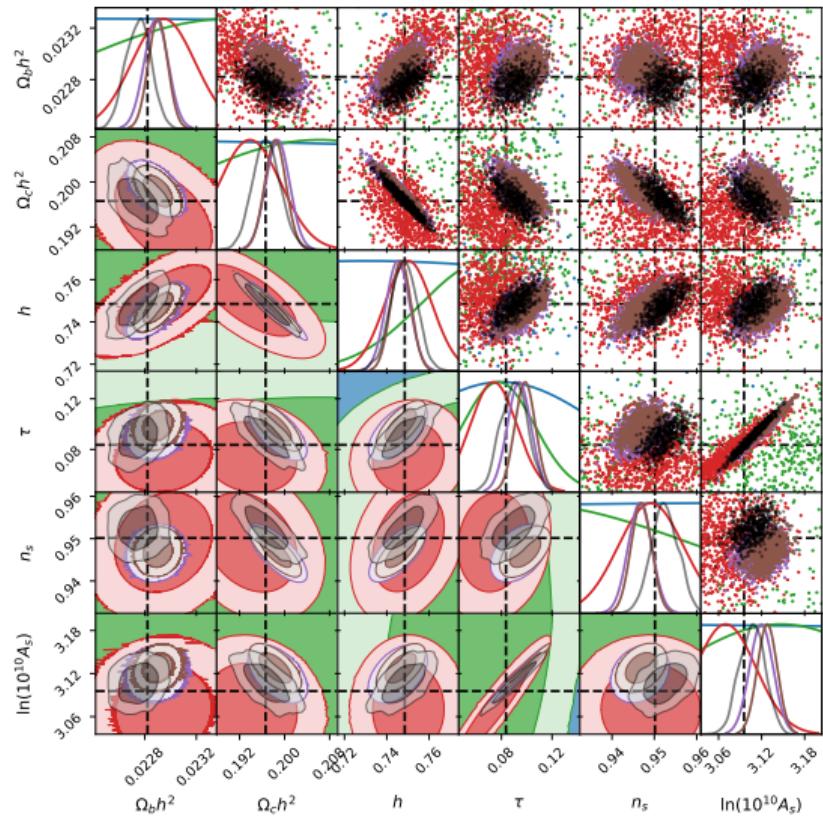
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



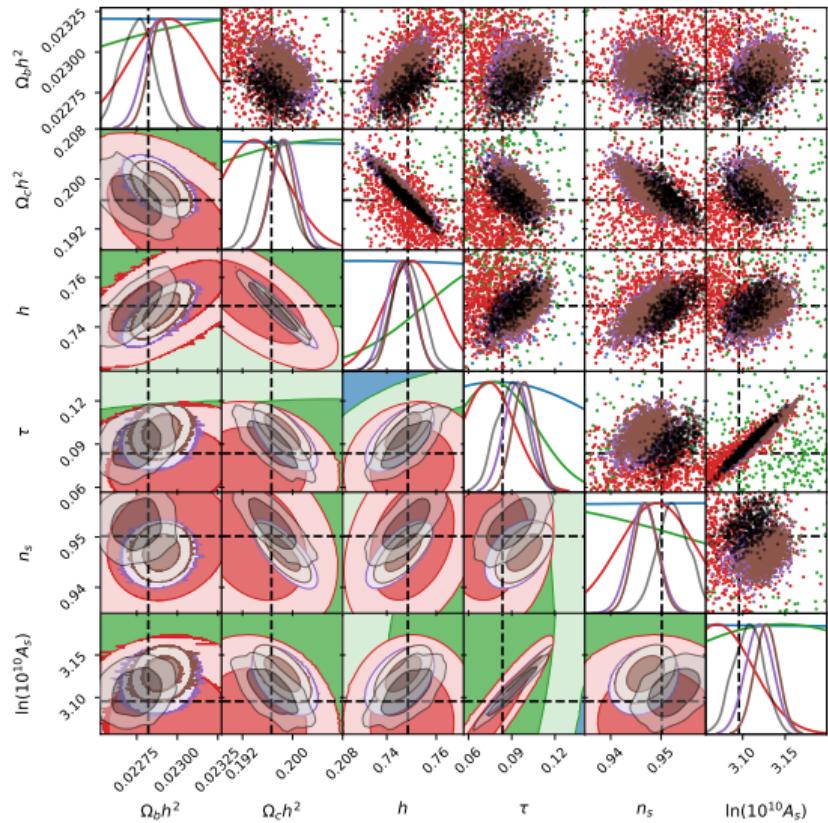
Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



Example of this on the CMB

- ▶ Now apply this to a “real” cosmology example, inferring Λ CDM from the CMB
- ▶ Unfortunately generative planck likelihoods do not exist yet
- ▶ Consider a cosmic-variance limited, temperature-only, full sky CMB experiment with no foregrounds
- ▶ This is a $n = 6$, $d = 2500$ non-linear problem
 - ▶ No compression needed
- ▶ Apply the above procedure
- ▶ Slight bias these results, but this can be fixed by marginalising over m, M, C , rather than taking point estimates.



lsbi: linear simulation based inference

Code details

- ▶ lsbi is a pip-installable python package
- ▶ it extends `scipy.stats.multivariate_normal`
 - ▶ vectorised distributions with (broadcastable) arrays of `mean` and `cov`
 - ▶ `.marginalise(...)` and `.condition(...)` methods
 - ▶ Plotting functionality
- ▶ Implements `LinearModel` class with `.prior()`, `.likelihood(theta)`, `.posterior(D)` & `.evidence()` methods which return distributions
- ▶ Also implement `MixtureModel`
- ▶ Under active development
 - ▶ Open source
 - ▶ Continuous integration
- ▶ github.com/handley-lab/lsbi

Where next?

- ▶ Paper being written up
 - ▶ soft deadline for Nicolas' MPhil start in October
 - ▶ hard deadline for PhD applications
- ▶ Include realistic CMB simulation effects (foregrounds)
- ▶ Extend to more examples (BAO, SNe)
- ▶ How does LSBI contribute to the question of compression
- ▶ Explore limits of d and n
- ▶ Explore mixture modelling for real nonlinear effects
- ▶ If the posterior is the answer, what is the question?
- ▶ Importance sampling?
- ▶ Model comparison?

Conclusions

github.com/handley-lab



- ▶ **Introduction to lsbi:** A linear simulation-based inference method developed over 18 months by the speaker and collaborators.
- ▶ **Benefits of Linear SBI:** Pedagogical value, practical examples with known ground truths, competitive accuracy, speed, and interpretability compared to neural networks.
- ▶ **Mathematical Setup:** Uses a linear generative model to fit simulation data and iteratively refine posterior estimations, demonstrated through toy and cosmology examples.
- ▶ **lsbi Python Package:** Extends `scipy.stats.multivariate_normal` with functionalities for marginalization, conditioning, and plotting; under active development and open source.
- ▶ **Future Directions:** Include realistic CMB simulations, extend to other examples (BAO, SNe), explore parameter limits, mixture modeling, and integrate importance sampling and model comparison.