Theory meets experiment 2025

New frontiers in particle cosmology

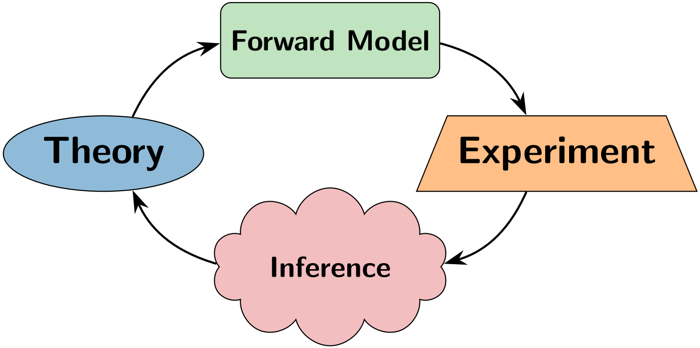
Will Handley
<wh260@cam.ac.uk>

Royal Society University Research Fellow
Institute of Astronomy, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
willhandley.co.uk/talks

7th January 2024

TMEX: Theory meets experiment

New frontiers in particle cosmology

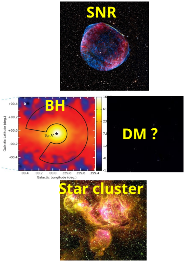


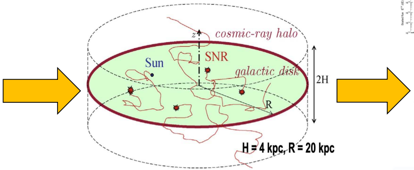
- Inference sits at the interface between theory and experiment
- Also called "inverse problems"
- Process is direct: "measurement"

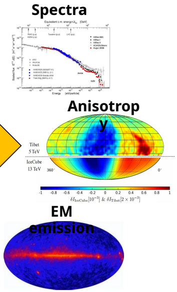
- This talk focuses on **frontiers**:

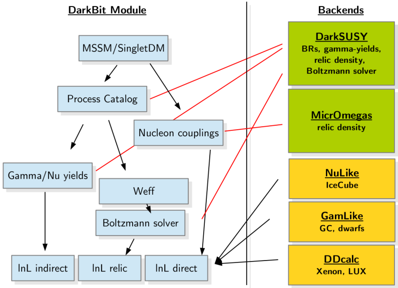
1. Simulation-based inference
2. GPU-accelerated inference

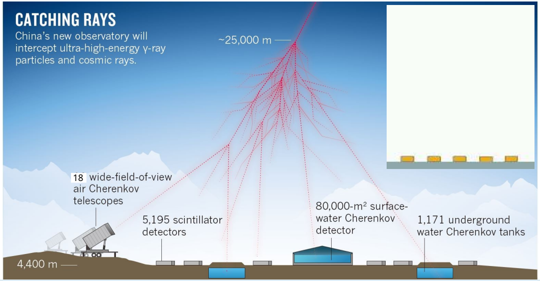
Examples of forward models from Monday

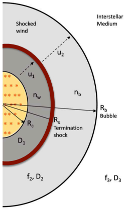












Morlino et al., MNRAS 504 (2021) 4

Time-stationary transport equation in spherical geometry:

$$\frac{\partial}{\partial r}\left[r^2 D(r,p)\frac{\partial f}{\partial r}\right] - r^2 u(r)\frac{\partial f}{\partial r} + \frac{d\left[r^2 u\right]}{dr}\frac{p}{3}\frac{\partial f}{\partial p} + r^2 Q(r,p) = 0$$

- Arbitrary diffusion coefficient $D(r,p)$
- Injection only at the termination shock $\quad Q(r,p) \propto \delta(p - p_{\rm inj})\,\delta(r - R_s)$
- Wind velocity profile: $u(r) = \begin{cases} u_1 = v_w & \text{for } r < R_s\,, \\ \frac{u_1}{\sigma}\left(\frac{R_s}{r}\right)^2 & \text{for } R_s < r < R_b\,, \\ 0 & \text{for } r > R_b\,; \end{cases}$

Boundary conditions:

1. No net flux at the cluster center: $\quad r^2[D\partial_r f - uf]_{r=R_c} = 0$
2. Matching the Galactic distribution: $\quad f(r \to \infty, p) = f_{\rm gal}(p)$

Bayesian & frequentist data combination

Multimessenger approaches

Frequentist

- Preferred by particle physicists & mathematicians
- Probability/stochasticity only in the data D

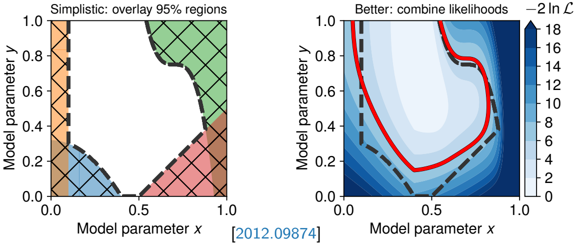
Bayesian

- Preferred by astronomers & machine learning/information theorists
- Quantifies all uncertainties in data & model (D, θ, M) using probability.

- Whether Bayesian or frequentist, If you have a model M with parameters θ, multiple datasets combine at the likelihood level:

$$P(D_1, D_2|\theta, M) = P(D_1|\theta, M)P(D_2|\theta, M)$$

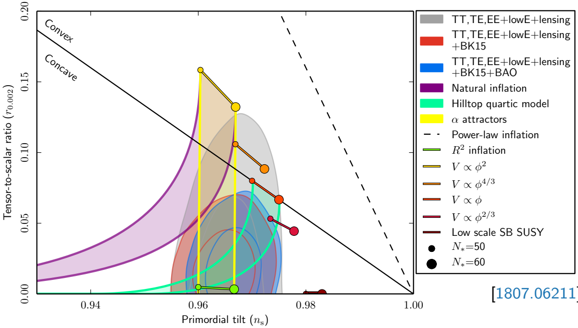
$$\mathcal{L}_{\text{joint}} = \mathcal{L}_1 \times \mathcal{L}_2 \times \ldots \times \mathcal{L}_n$$



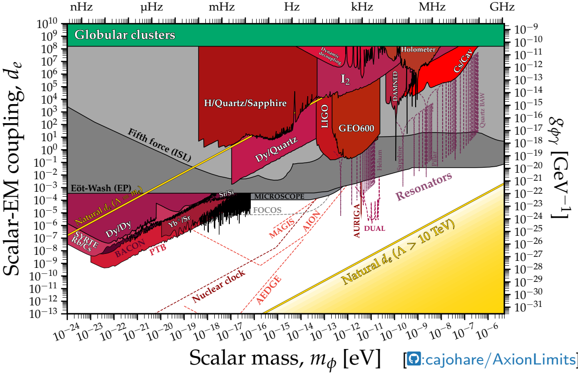
[2012.09874]

An aside: difference in plotting

Exclusion vs. posterior plots



[1807.06211]



[:cajohare/AxionLimits]

- Contours indicate **allowed** regions
- Preferred in astro/cosmology

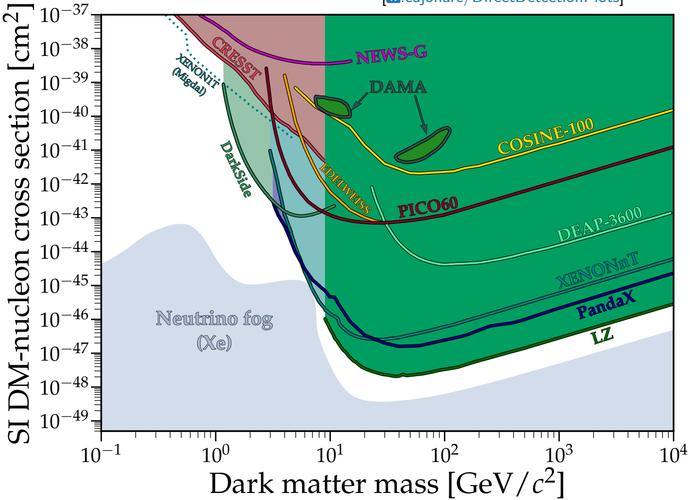
- Contours indicate **excluded** regions
- Preferred in particle physics

An aside: difference in plotting

Exclusion vs. posterior plots

- Beware this kind of particularly confusing plot, which uses both!
- Here almost all of these are 2σ exclusion plots
- But 'DAMA' are (controversial & conflicting) superimposed constraints/allowed regions.

[:cajohare/DirectDetectionPlots]



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 ΛCDM 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. ΛCDM vs a dynamic dark energy cosmology*

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

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

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

Tension quantification

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

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

$$\log\mathcal{S} = \langle\log\mathcal{L}_{AB}\rangle_{\mathcal{P}_{AB}} - \langle\log\mathcal{L}_{A}\rangle_{\mathcal{P}_{A}} - \langle\log\mathcal{L}_{B}\rangle_{\mathcal{P}_{B}}$$

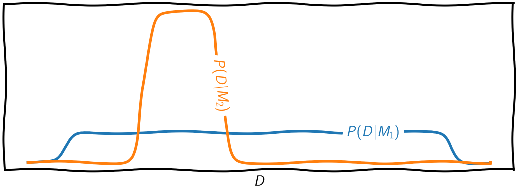
Model comparison $\mathcal{Z} = P(D|M)$

- 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 θ:

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 π and posterior $\mathcal{P}$.

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

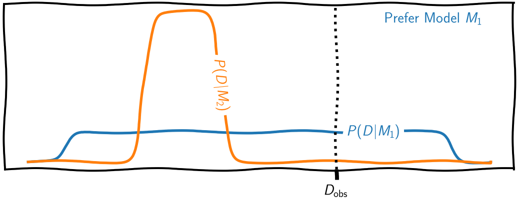
Model comparison $\mathcal{Z} = P(D|M)$

- 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 θ:

Occam's razor

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

- Occam's razor equation:

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

- "Occam penalty": KL divergence between prior π and posterior $\mathcal{P}$.

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

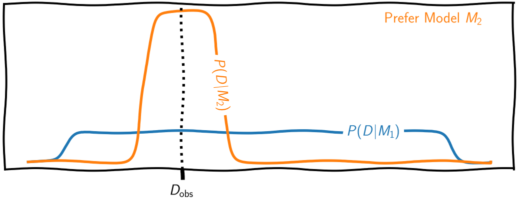
Model comparison $\mathcal{Z} = P(D|M)$

- 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 θ:

Occam's razor

- Models should be as simple as possible
- …but no simpler

- Occam's razor equation:

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

- "Occam penalty": KL divergence between prior π and posterior $\mathcal{P}$.

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

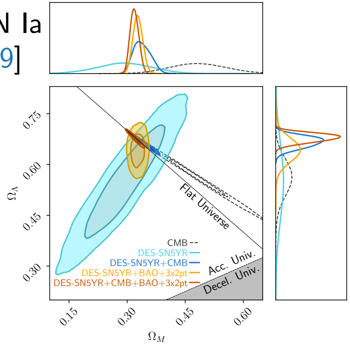
LBI: Likelihood-based inference

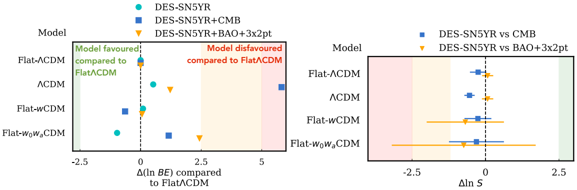
The standard approach if you are fortunate enough to have a likelihood function $P(D|\theta)$:

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

1. Define prior $\pi(\theta)$
   - spend some time being philosophical
2. Sample posterior $\mathcal{P}(\theta|D)$
   - 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]

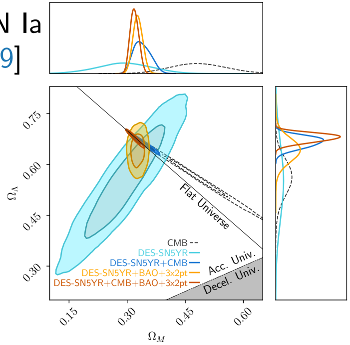




LBI: Likelihood-based inference

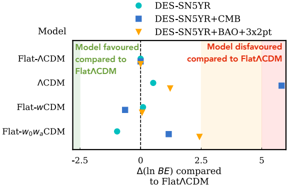
The standard approach if you are fortunate enough to have a likelihood function $P(D|\theta)$:

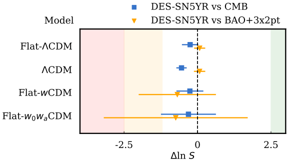
DES Y5 SN Ia [2401.02929]



$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \qquad \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

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





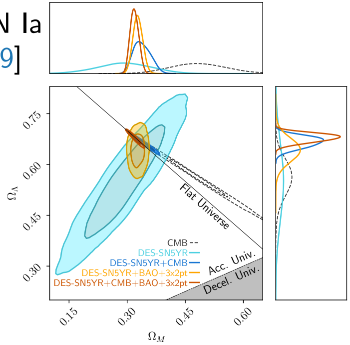
LBI: Likelihood-based inference

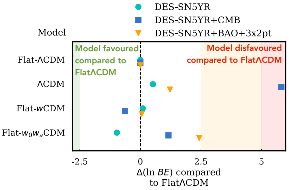
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

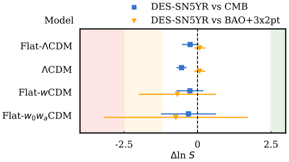
$$\mathcal{P}(\theta|D) = \frac{\mathcal{L}(D|\theta)\pi(\theta)}{\mathcal{Z}(D)} \qquad \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

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]





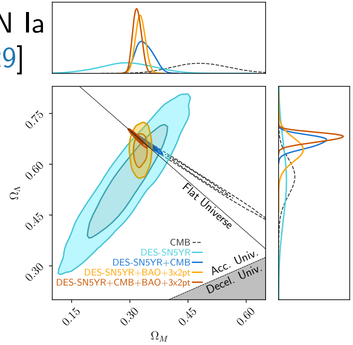


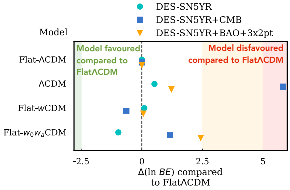
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

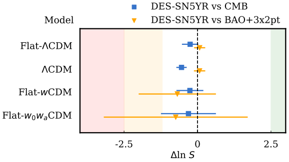
$$P(\theta|D)P(D) = P(\theta, D) = P(D|\theta)P(\theta),$$

1. Define prior $\pi(\theta)$
   - spend some time being philosophical
2. Sample posterior $\mathcal{P}(\theta|D)$
   - 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]







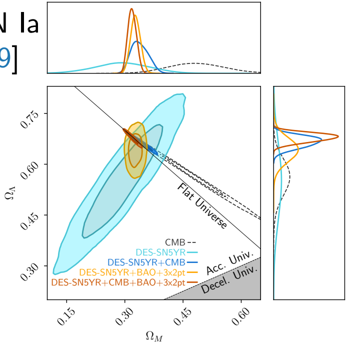
LBI: Likelihood-based inference

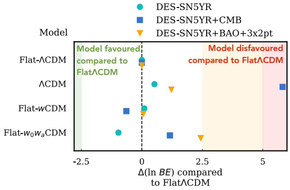
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

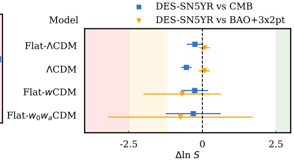
$$\mathcal{P} \times \mathcal{Z} = \mathcal{J} = \mathcal{L} \times \pi, \qquad \text{Joint} = \mathcal{J} = P(\theta, D)$$

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]

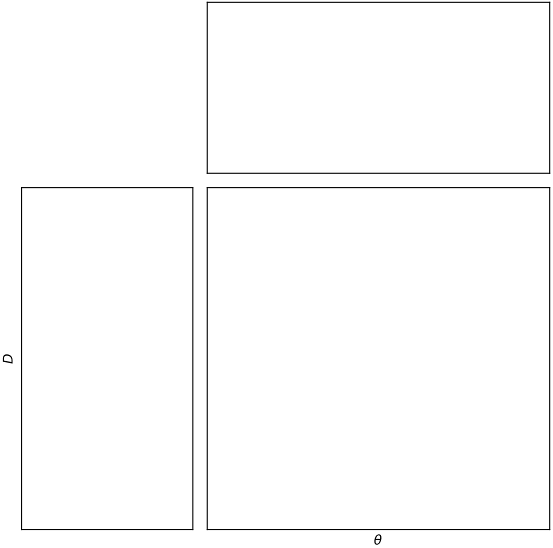






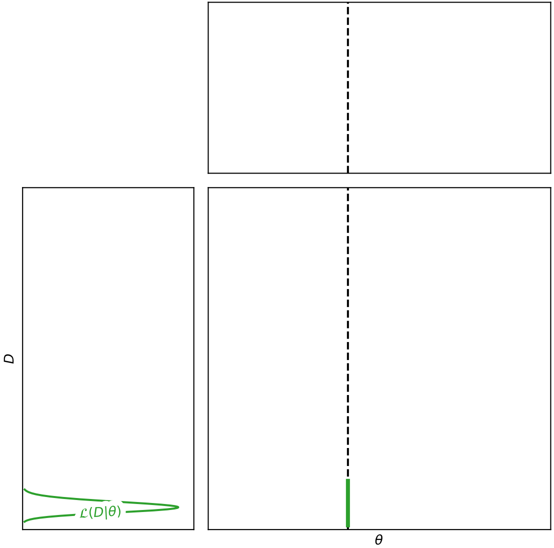
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



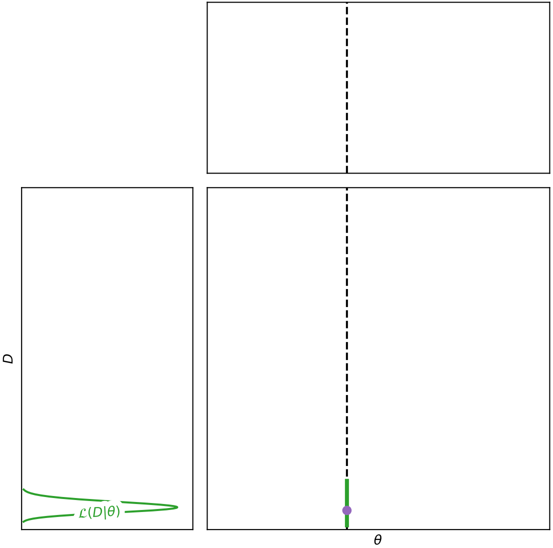
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



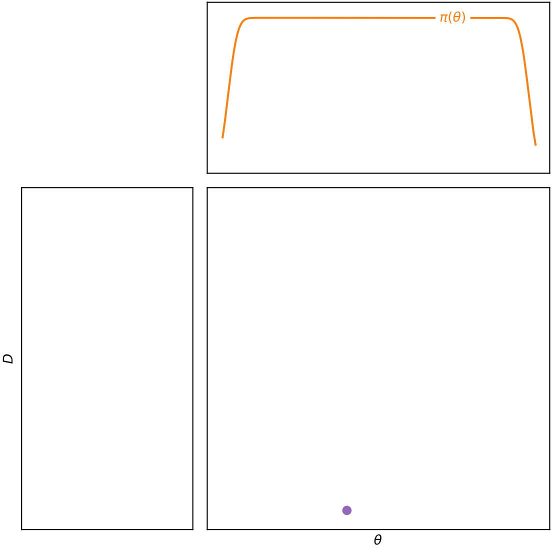
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



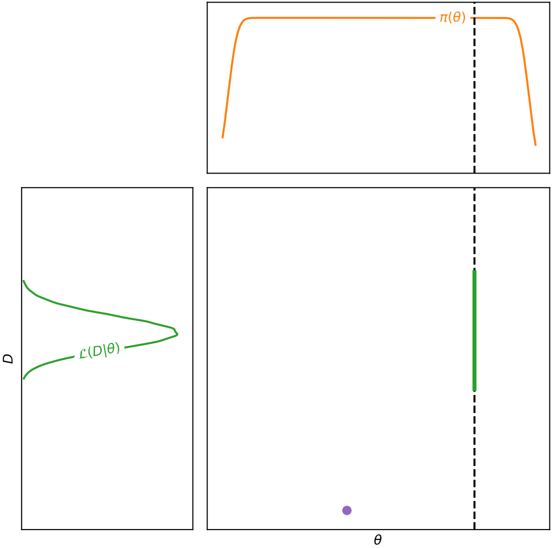
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].

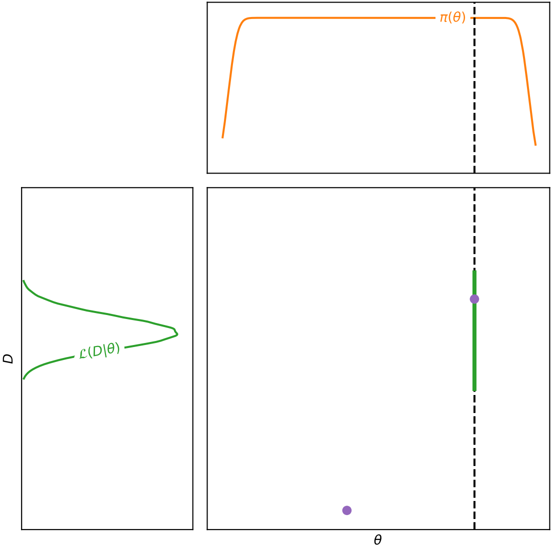


SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].

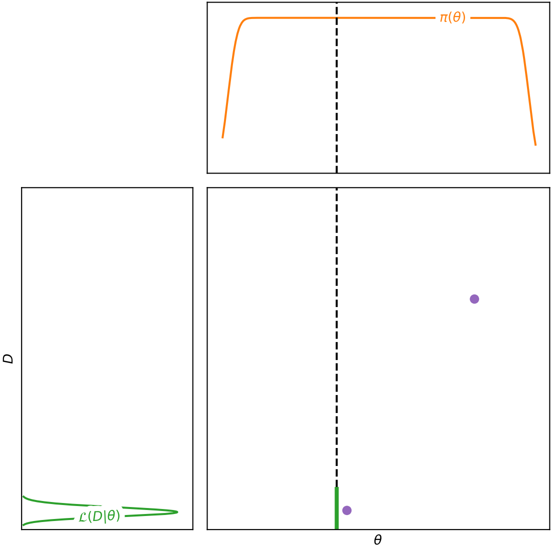


- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \rightarrow D$ defines an *implicit* likelihood $\mathcal{L}$.
- Simulator generates samples from $\mathcal{L}(\cdot|\theta)$.
- With a prior $\pi(\theta)$ can generate samples from joint distribution $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$ *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 [:handley-lab/lsbi].



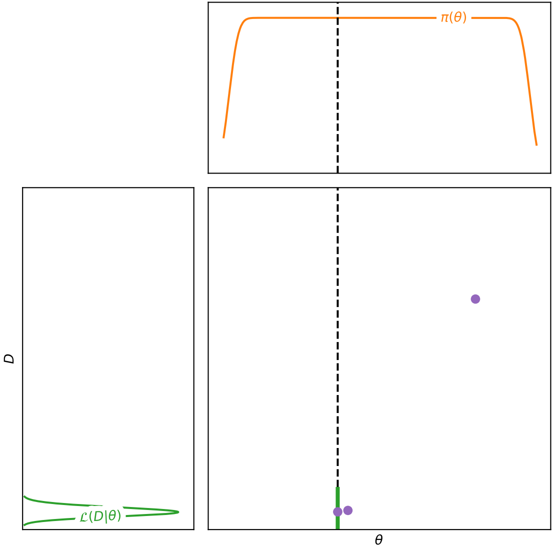
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



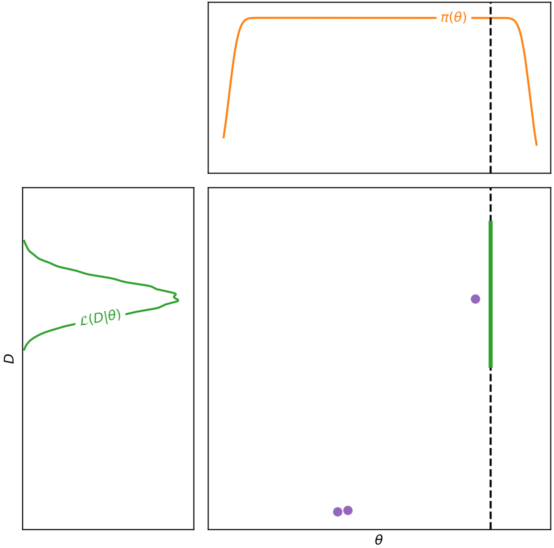
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



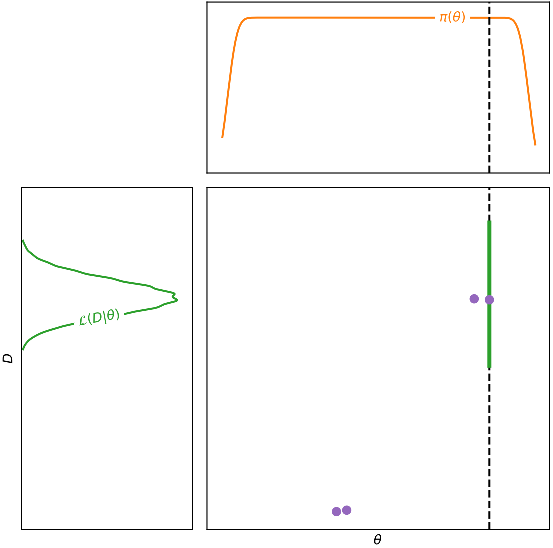
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



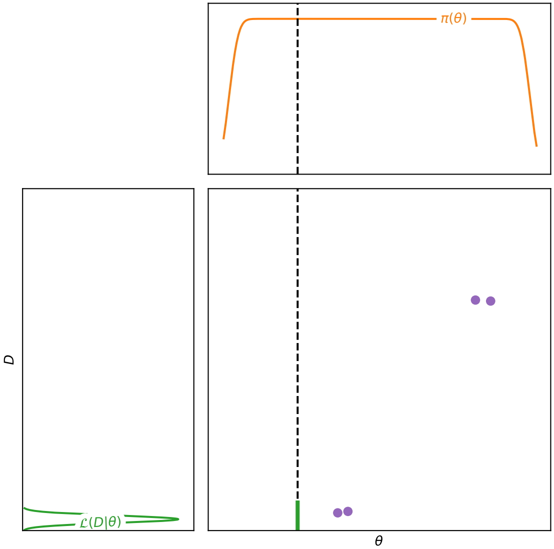
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



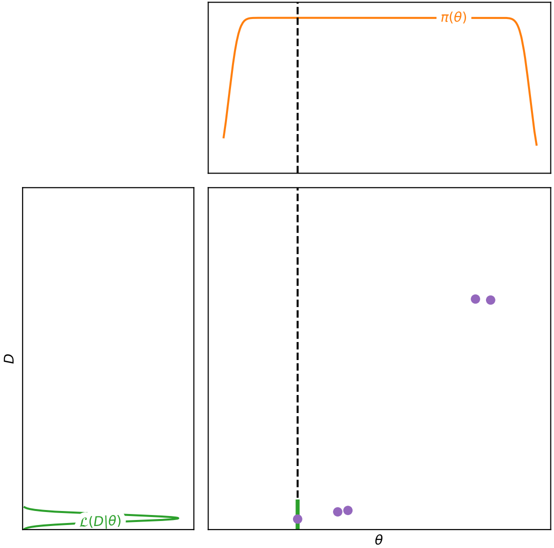
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



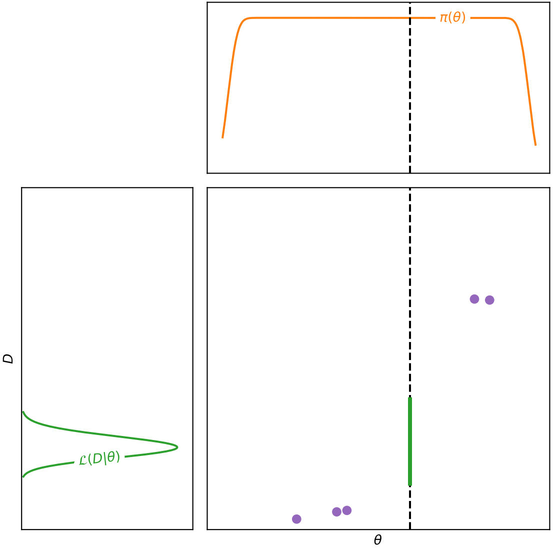
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



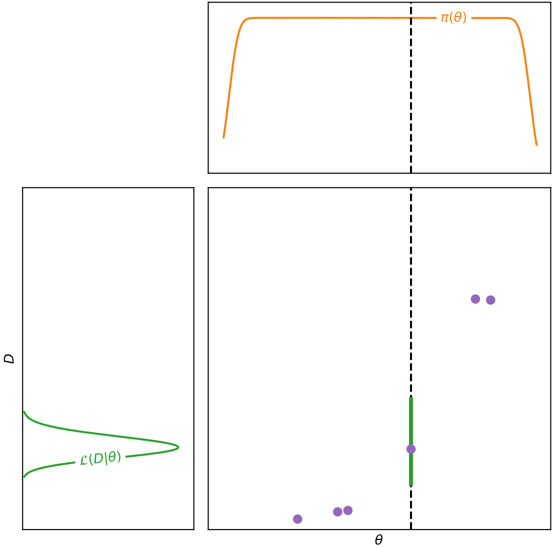
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



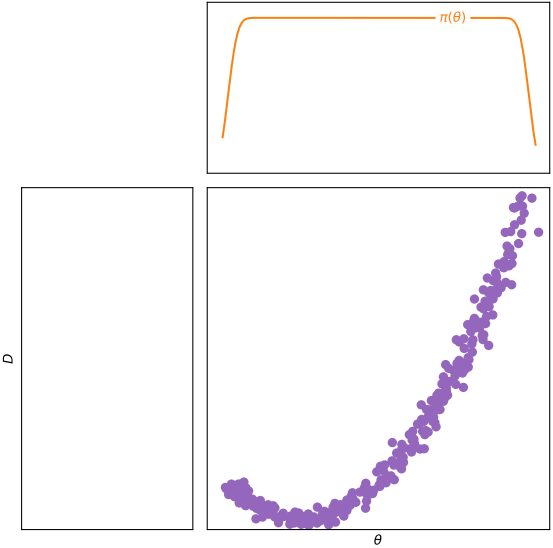
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



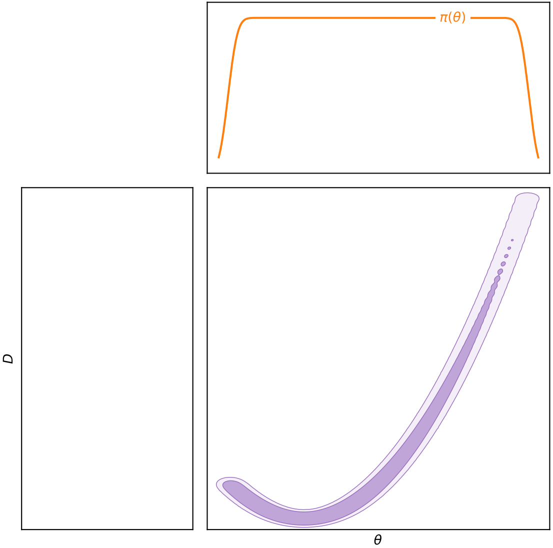
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



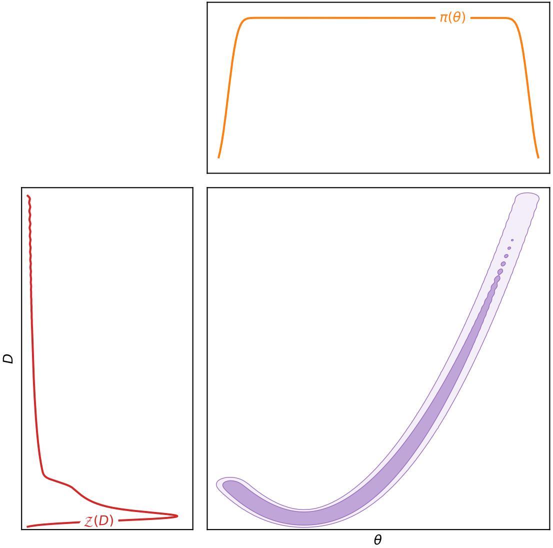
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



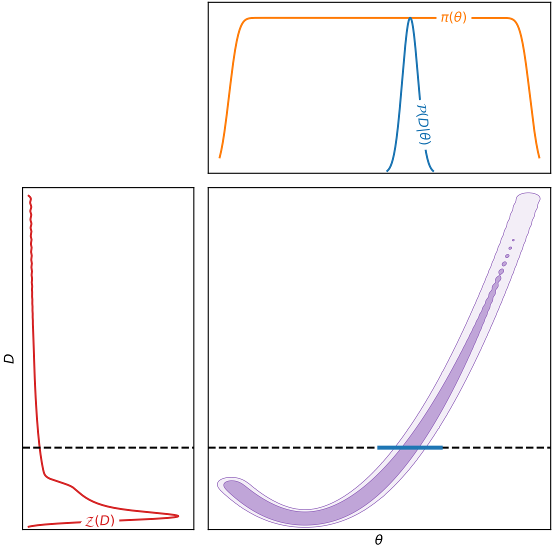
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



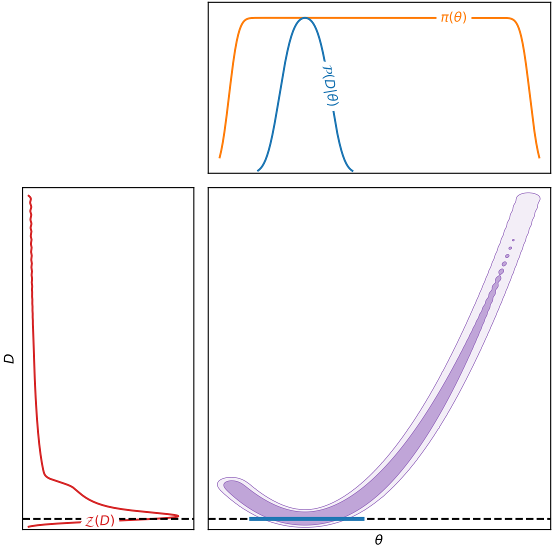
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



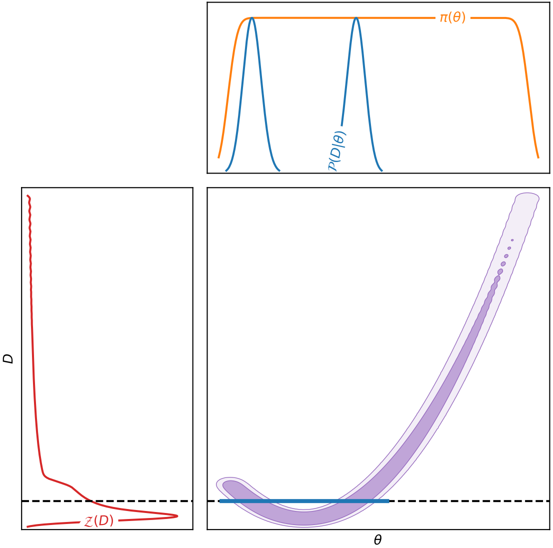
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



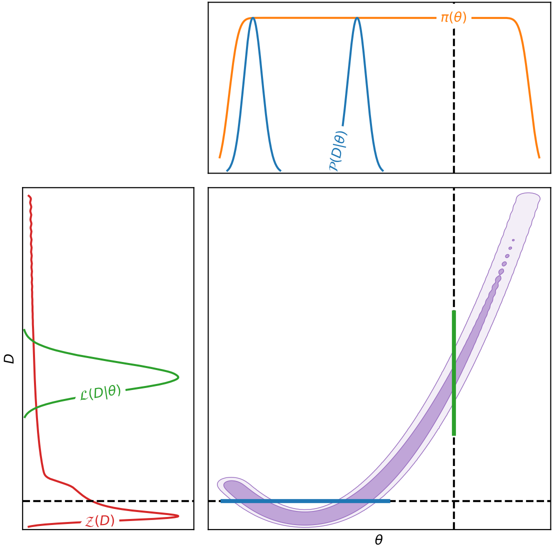
SBI: Simulation-based inference

- 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 [:handley-lab/lsbi].



SBI: Simulation-based inference

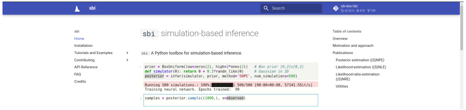
- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \rightarrow D$ defines an *implicit* likelihood $\mathcal{L}$.
- Simulator generates samples from $\mathcal{L}(\cdot|\theta)$.
- With a prior $\pi(\theta)$ can generate samples from joint distribution $\mathcal{J}(\theta, D) = \mathcal{L}(D|\theta)\pi(\theta)$ *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 [:handley-lab/lsbi].



Why SBI?

SBI is useful because:

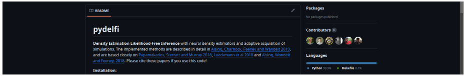
1. If you don't have a likelihood, you can still do inference
   - This is the usual case beyond CMB cosmology
2. Faster than LBI
   - emulation – also applies to LBI in principle
3. No need to pragmatically encode fiducial cosmologies
   - Covariance computation implicitly encoded in simulations
   - 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



[:sbi-dev]

[:undark-lab/swyft]

[:florent-leclercq/pyselfi]



[:justinalsing/pydelfi]

SBI in astrophysics

- 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`).
- [:smsharma/awesome-neural-sbi]

Investigating the turbulent hot gas in X-COP galaxy clusters

S. Dupourqué[1], N. Clerc[1], E. Pointecouteau[1], D. Eckert[2], S. Ettori[3], and F. Vazza[4,5,6]

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,[1,*] G. Campailla,[2] N. Jeffrey,[3] L. Whiteway,[3] A. Porredon,[4] J. Prat,[5] J. Williamson,[3] M. Raveri,[2] B.

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

Marco Crisostomi[1,2], Kallol Dey[3], Enrico Barausse[1,2], Roberto Trotta[1,2,4,5]

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

Katharena Christy,[a] Eric J. Baxter,[b] Jason Kumar[a]

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

Joachim Harnois-Déraps[1★], Sven Heydenreich[2], Benjamin Giblin[3], Nicolas Martinet[4], Tilman Tröster[5], Marika Asgari[1,6,7], Pierre Burger[8,9,10],Tiago Castro[11,12,13,14], Klaus Dolag[15], Catherine Heymans[3,16], Hendrik Hildebrandt[16], Benjamin Joachimi[17] & Angus H. Wright[16]

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

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

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

Beatrice Moser,[a,1] Tomasz Kacprzak,[a,b] Silvan Fischbacher,[a] Alexandre Refregier,[a] Dominic Grimm,[a] Luca Tortorelli[c]

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

Elena Massara,[1,2,*] ChangHoon Hahn,[3] Michael Eickenberg,[4] Shirley Ho,[5] Jiamin Hou,[6,7] Pablo Lemos,[8,9,5] Chirag Modi,[4,5] Azadeh Moradinezhad Dizgah,[10,11] Liam Parker,[5,12] and Bruno Régaldo-Saint Blancard[4]

Cosmology from HSC Y1 Weak Lensing with Combined Higher-Order Statistics and Simulation-based Inference

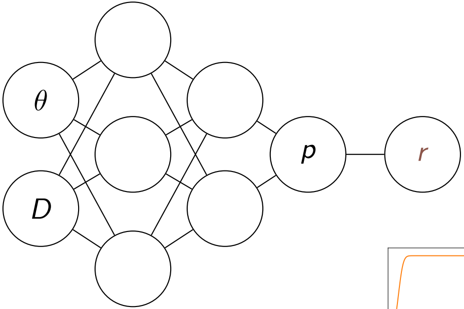
Camila P. Novaes[1,2,3,*] Leander Thiele[2,3,†] Joaquin Armijo[2,3], Sihao Cheng[4,5], Jessica A. Cowell[2,3,6], Gabriela A. Marques[7,8], Elisa G. M. Ferreira[2,3], Masato Shirasaki[9,10], Ken Osato[11,12,2], and Jia Liu[2,3]

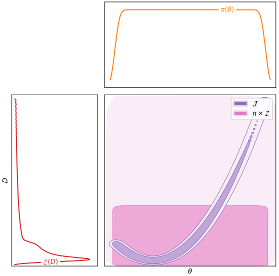
Neural Ratio Estimation

- SBI flavours: github.com/sbi-dev/sbi

  NPE Neural posterior estimation
  NLE Neural likelihood estimation
  NJE Neural joint estimation
  NRE Neural ratio estimation

- NRE recap:
  1. Generate joint samples $(\theta, D) \sim \mathcal{J}$
     - *straightforward if you have a simulator:* $\theta \sim \pi(\cdot)$, $D \sim \mathcal{L}(\cdot|\theta)$
  2. Generate separated samples $\theta \sim \pi$, $D \sim \mathcal{Z}$
     - *aside: can shortcut step 2 by scrambling the* (θ, D) *pairings from step 1*
  3. Train probabilistic classifier p to distinguish whether (θ, D) came from $\mathcal{J}$ or $\pi \times \mathcal{Z}$.
  4. $\frac{p}{1-p} = r = \frac{P(\theta,D)}{P(\theta)P(D)} = \frac{\mathcal{J}}{\pi\times\mathcal{Z}} = \frac{\mathcal{L}}{\mathcal{Z}} = \frac{\mathcal{P}}{\pi}$.
  5. Use ratio r for parameter estimation $\mathcal{P} = r\times\pi$





Neural Ratio Estimation

- SBI flavours: github.com/sbi-dev/sbi

  NPE Neural posterior estimation
  NLE Neural likelihood estimation
  NJE Neural joint estimation
  NRE Neural ratio estimation

- NRE recap:
  1. Generate joint samples $(\theta, D) \sim \mathcal{J}$
     - *straightforward if you have a simulator:* $\theta \sim \pi(\cdot)$, $D \sim \mathcal{L}(\cdot|\theta)$
  2. Generate separated samples $\theta \sim \pi$, $D \sim \mathcal{Z}$
     - *aside: can shortcut step 2 by scrambling the* (θ, D) *pairings from step 1*
  3. Train probabilistic classifier p to distinguish whether (θ, D) came from $\mathcal{J}$ or $\pi \times \mathcal{Z}$.
  4. $\frac{p}{1-p} = r = \frac{P(\theta,D)}{P(\theta)P(D)} = \frac{\mathcal{J}}{\pi\times\mathcal{Z}} = \frac{\mathcal{L}}{\mathcal{Z}} = \frac{\mathcal{P}}{\pi}$.
  5. Use ratio r for parameter estimation $\mathcal{P} = r \times \pi$

Bayesian proof

- Let $M_\mathcal{J}$: $(\theta, D) \sim \mathcal{J}$, $M_{\pi\mathcal{Z}}$: $(\theta, D) \sim \pi \times \mathcal{Z}$
- Classifier gives $p(\theta, D) = P(M_\mathcal{J}|\theta, D) = 1 - P(M_{\pi\mathcal{Z}}|\theta, D)$
- Bayes theorem then shows
  $$\frac{p}{1-p} = \frac{P(M_\mathcal{J}|\theta,D)}{P(M_{\pi\mathcal{Z}}|\theta,D)} = \frac{P(\theta,D|M_\mathcal{J})P(M_\mathcal{J})}{P(\theta,D|M_{\pi\mathcal{Z}})P(M_{\pi\mathcal{Z}})} = \frac{\mathcal{J}}{\pi\mathcal{Z}},$$
  where we have assumed
  - $P(M_\mathcal{J}) = P(M_{\pi\mathcal{Z}})$,

  and by definition
  - $\mathcal{J}(\theta, D) = P(\theta, D|M_\mathcal{J})$
  - $\pi(\theta)\mathcal{Z}(D) = P(\theta, D|M_{\pi\mathcal{Z}})$.

Neural Ratio Estimation

- SBI flavours: github.com/sbi-dev/sbi

  NPE Neural posterior estimation
  NLE Neural likelihood estimation
  NJE Neural joint estimation
  NRE Neural ratio estimation

- NRE recap:
  1. Generate joint samples $(\theta, D) \sim \mathcal{J}$
     - *straightforward if you have a simulator:* $\theta \sim \pi(\cdot)$, $D \sim \mathcal{L}(\cdot|\theta)$
  2. Generate separated samples $\theta \sim \pi$, $D \sim \mathcal{Z}$
     - *aside: can shortcut step 2 by scrambling the* (θ, D) *pairings from step 1*
  3. Train probabilistic classifier p to distinguish whether (θ, D) came from $\mathcal{J}$ or $\pi \times \mathcal{Z}$.
  4. $\frac{p}{1-p} = r = \frac{P(\theta,D)}{P(\theta)P(D)} = \frac{\mathcal{J}}{\pi\times\mathcal{Z}} = \frac{\mathcal{L}}{\mathcal{Z}} = \frac{\mathcal{P}}{\pi}$.
  5. Use ratio r for parameter estimation $\mathcal{P} = r \times \pi$

Why I like NRE

- The link between classification and inference is profound.
- Density estimation is hard – Dimensionless r divides out the hard-to-calculate parts.

Why I don't like NRE

- Practical implementations require marginalisation [2107.01214], or autoregression [2308.08597].
- Model comparison and parameter estimation are separate [2305.11241].

I want (my student) to get started with SBI...

...where should I send them?

the swyft package

- Ratio estimation
- astro/cosmology specific examples

swyft.readthedocs.io/en/stable

the pydelfi package

- Neural density estimation
- astro/cosmology specific examples

justinalsing.github.io/pydelfi

the sbi package

- General package
- Not domain specific
- A lot of (opaquely named) methods

sbi-dev.github.io/sbi/latest/tutorials

All methods generally require:

- A forward simulator
- A data compressor

All methods either:

- "Amortized" over data D
- "Sequential" tuning to D_{obs}

GPU-accelerated inference

CMB `cosmopower` [2106.03846]
CMB `candl` [2401.13433]
SNe `BayesSN` [2401.08755]
SGW `Eryn` [2303.02164]
GW `redback` [2308.12806]
GW `ripple` [2302.05329]
EP `ExoJAX` [2105.14782]
X `jaxspec` [2409.05757]

[:JAXtronomy]

- Increase in the number of cosmological codes written for GPUs (particularly jax).
- Over the next few years, more and more analyses will be done on GPUs.
- Several trends trigger this
  - the rise of machine learning, whose linear algebra is well-suited to GPUs
  - the creation of usable languages for GPU programming (e.g. `jax`, `pytorch`, `tensorflow`)
  - the rise of large language models, which ease writing codes for GPUs
- Prediction: low-power GPUs (likely ARM-based) will become the norm for scientific computing.

Jax-based nested samplers

David Yallup
PDRA

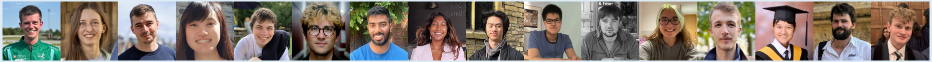
- **very** recent work over the past month
- Have implemented a nested slice sampler in `blackjax` [:blackjax-devs/blackjax/pull/755].

```
pip install git+https://github.com/handley-lab/blackjax@nested_sampling
import blackjax.ns.adaptive
```

- Think `MultiNest` for jax.
- Plugs into `jim` [:kazewong/jim] and `ripple` [2302.05329]

Conclusions

[:handley-lab]



- **Inference** bridges theory and experiment, crucial for extracting information from data.
- **Simulation-Based Inference (SBI)** enables inference when the likelihood is intractable, using simulations and machine learning. SBI is becoming increasingly popular for complex astrophysical analyses.
- **GPU-accelerated inference** is transforming the field, allowing faster and more complex computations. Tools like `jax` are empowering a new generation of GPU-ready inference codes.

Frontiers of simulation based inference [1911.01429]