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

Will Handley
<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

18th July 2024


UNIVERSITY OF CAMBRIDGE



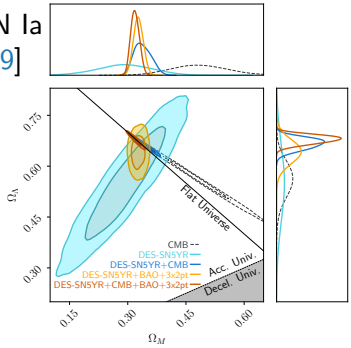
LBI: Likelihood-based inference

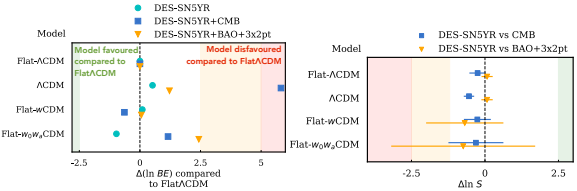
The standard approach if you are fortunate enough to have a likelihood function $P(D|\theta)$:

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

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





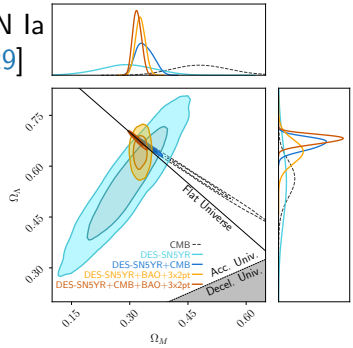
LBI: Likelihood-based inference

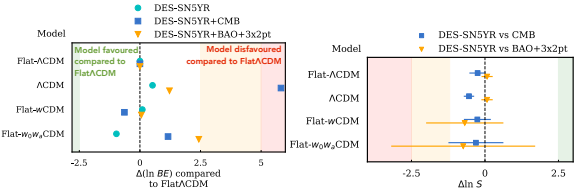
The standard approach if you are fortunate enough to have a likelihood function $P(D|\theta)$:

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

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

DES Y5 SN Ia [2401.02929]





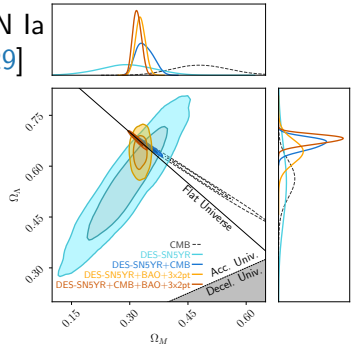
LBI: Likelihood-based inference

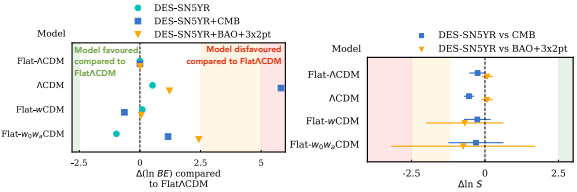
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

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

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]





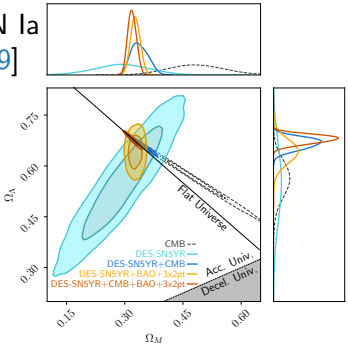
LBI: Likelihood-based inference

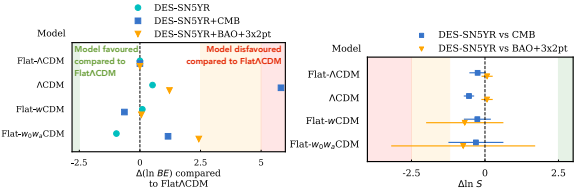
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

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

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





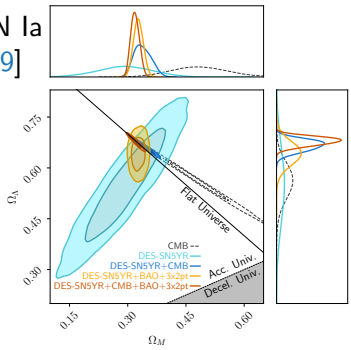
LBI: Likelihood-based inference

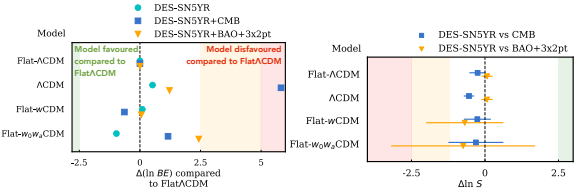
The standard approach if you are fortunate enough to have a likelihood function $\mathcal{L}(D|\theta)$:

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

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]



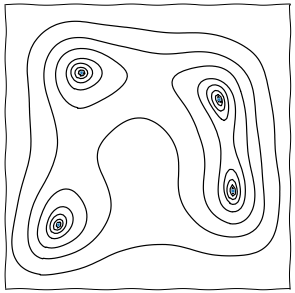


What is Nested Sampling?

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

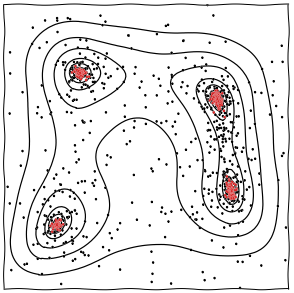
Optimisation

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



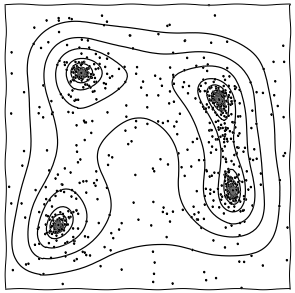
Exploration

draw/sample $\quad \theta \sim f$



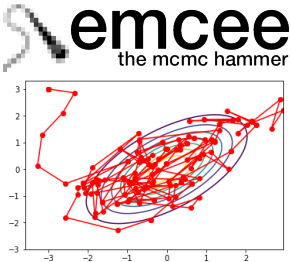
Integration

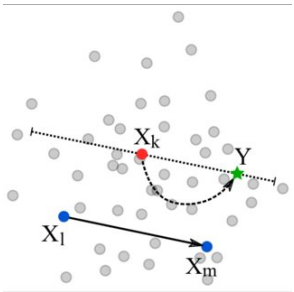
$$\int f(\theta) dV$$

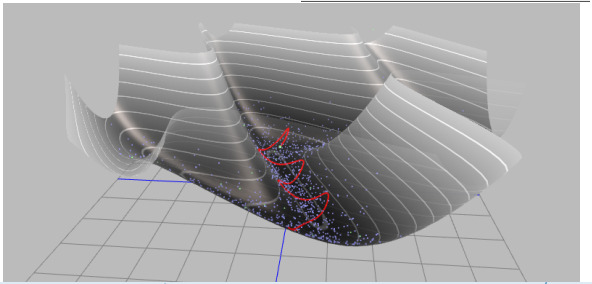


Where is Nested Sampling?

- 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

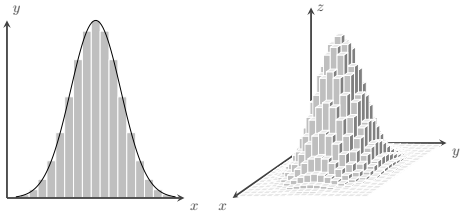
$$\Psi = \int e^{iS} \mathcal{D}x$$

Bayesian marginals

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

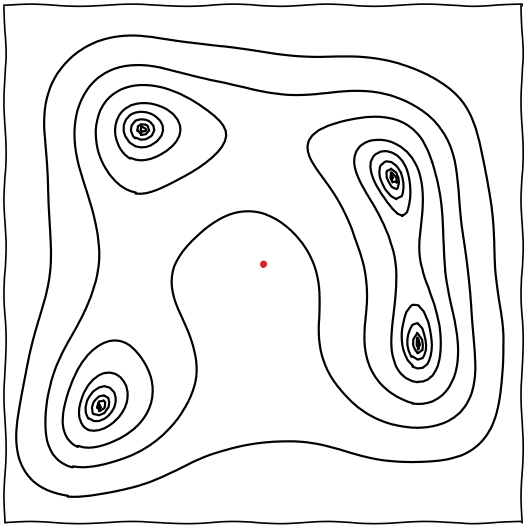
- Need numerical tools if analytic solution unavailable.
- High-dimensional numerical integration is hard.
- Riemannian strategy estimates volumes geometrically:

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

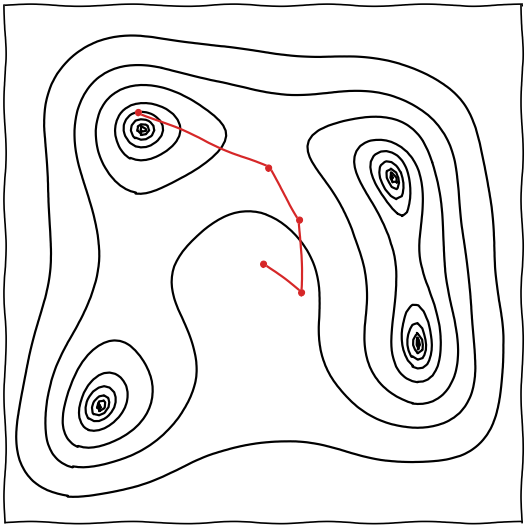


- Curse of dimensionality $\Rightarrow$ exponential scaling.

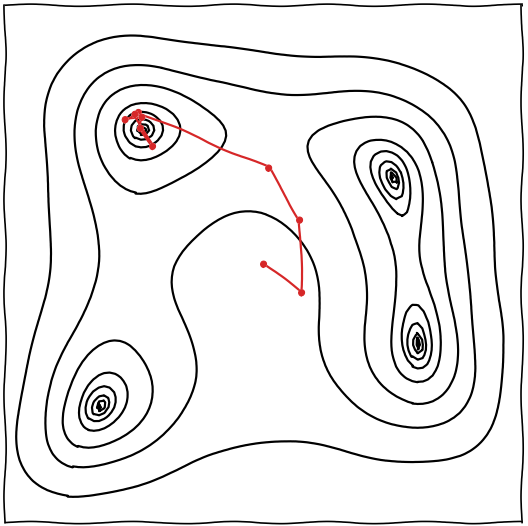
MCMC



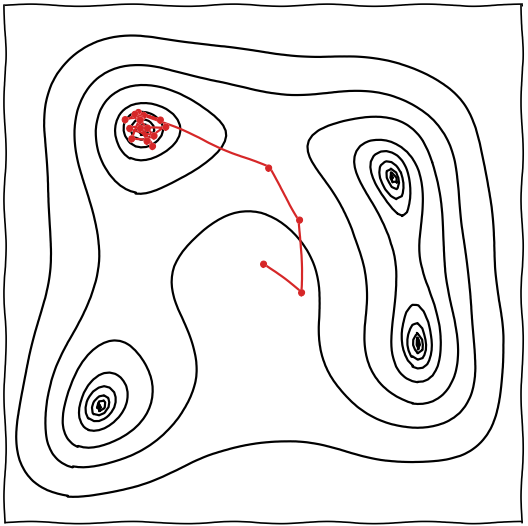
MCMC



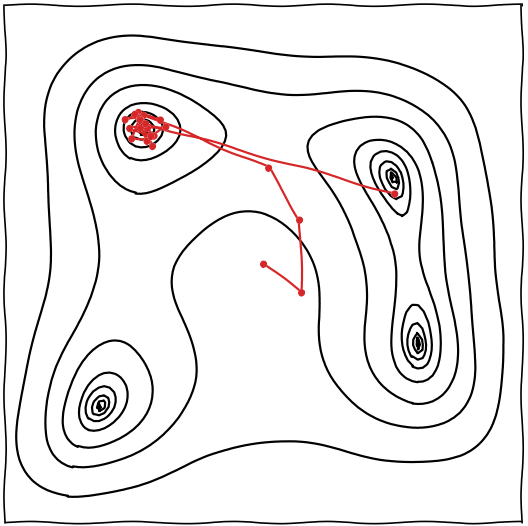
MCMC



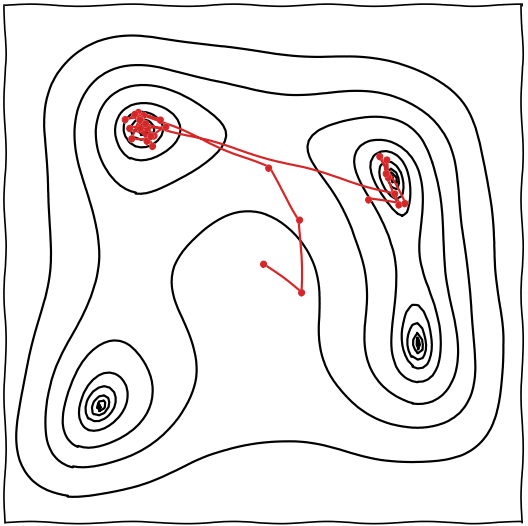
MCMC



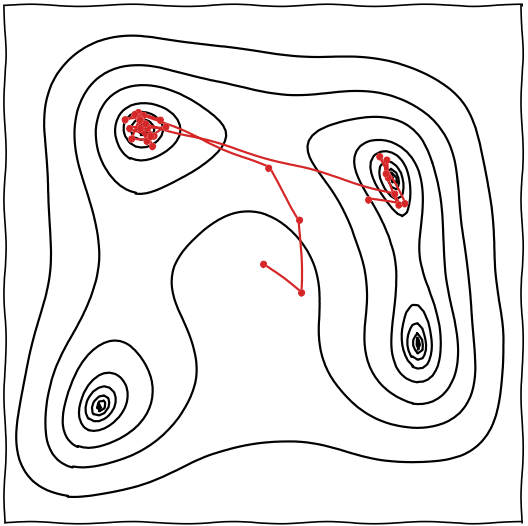
MCMC



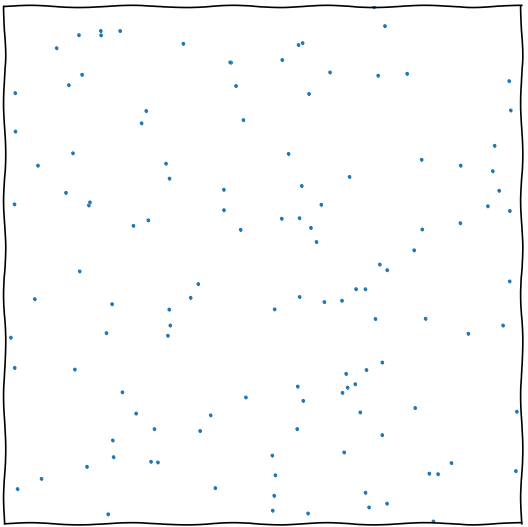
MCMC



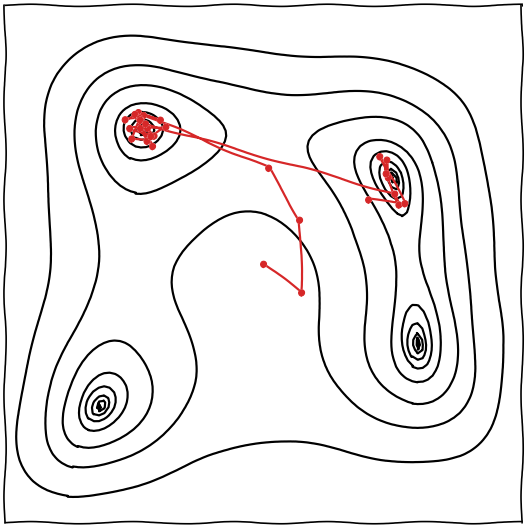
MCMC



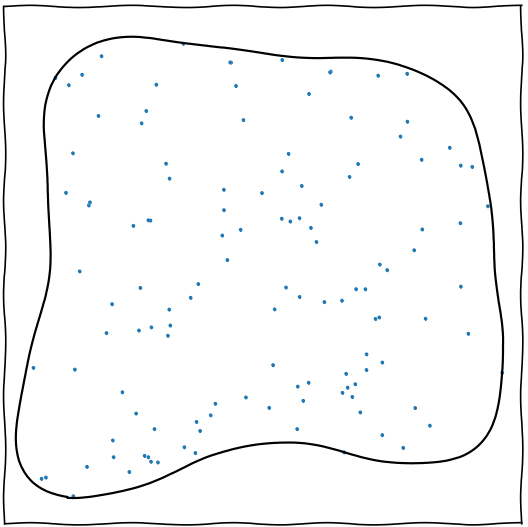
Nested sampling



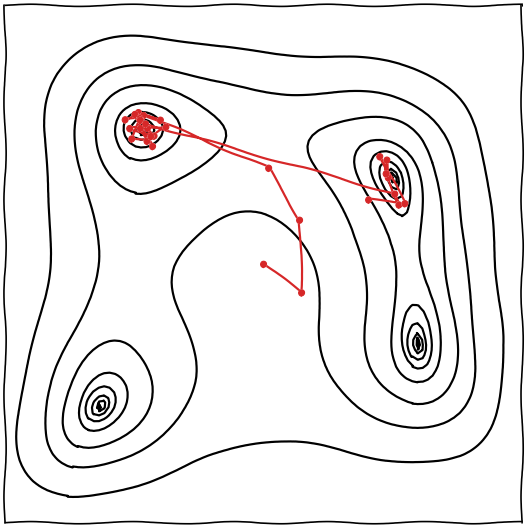
MCMC



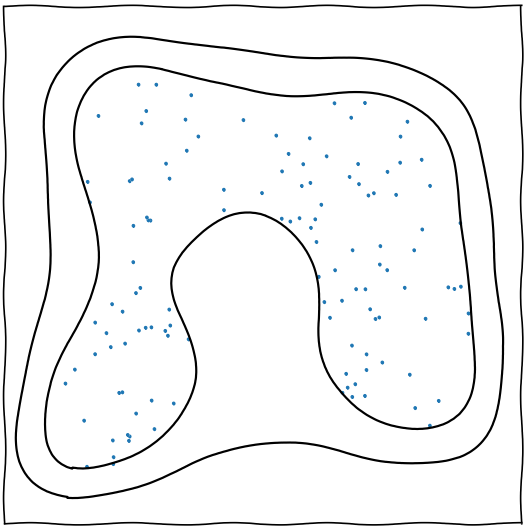
Nested sampling



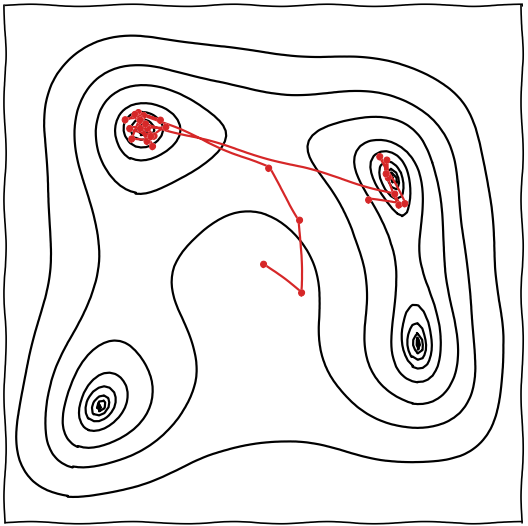
MCMC



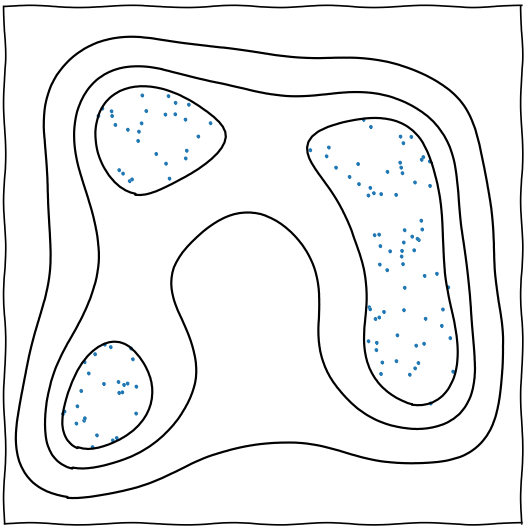
Nested sampling



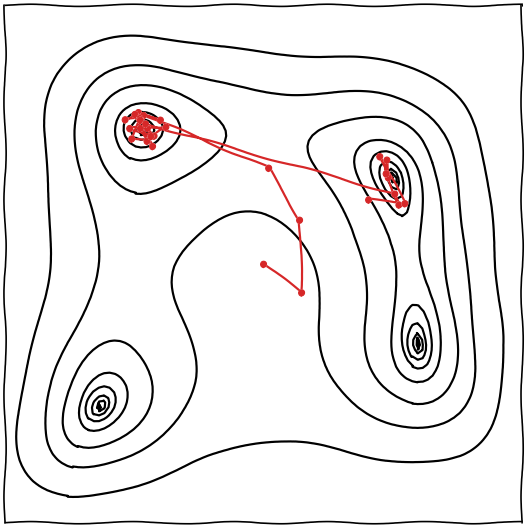
MCMC



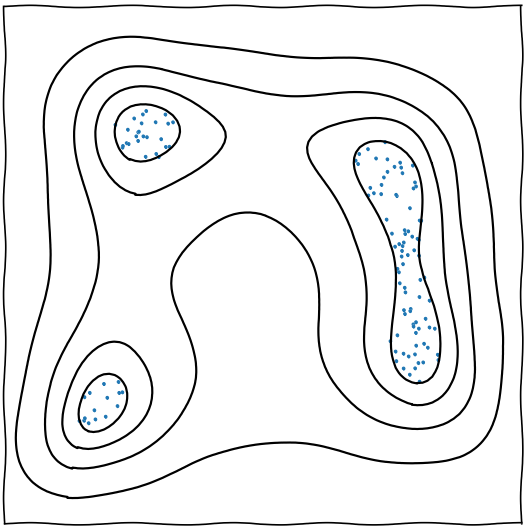
Nested sampling



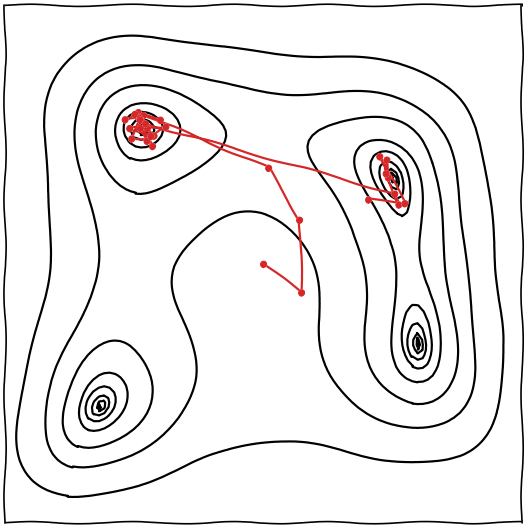
MCMC



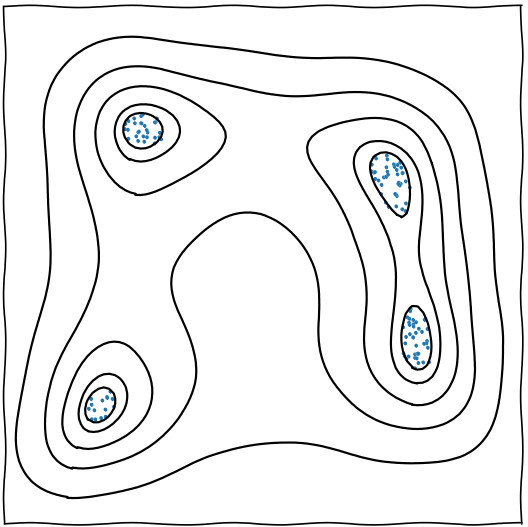
Nested sampling



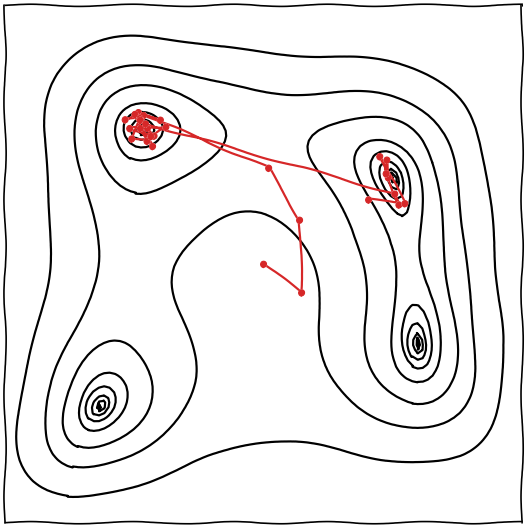
MCMC



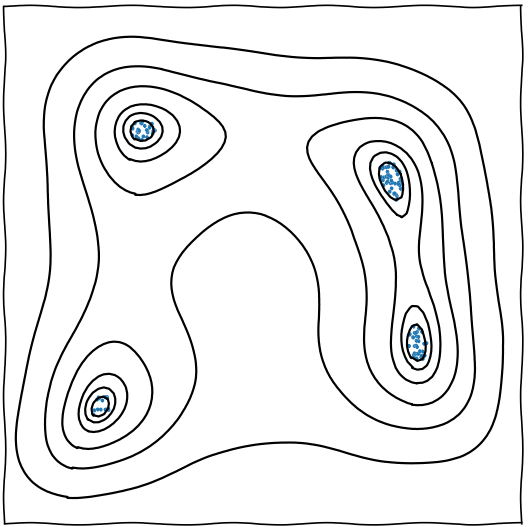
Nested sampling



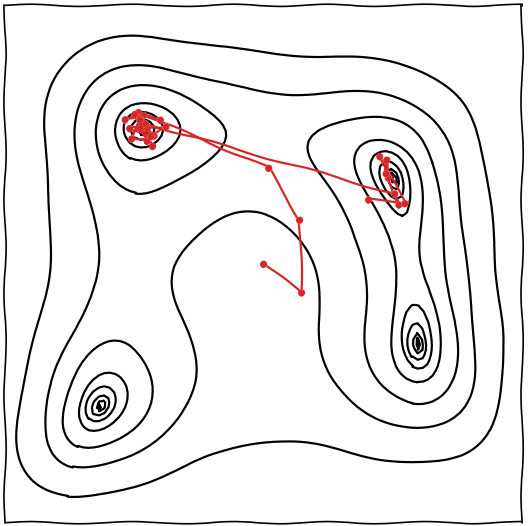
MCMC



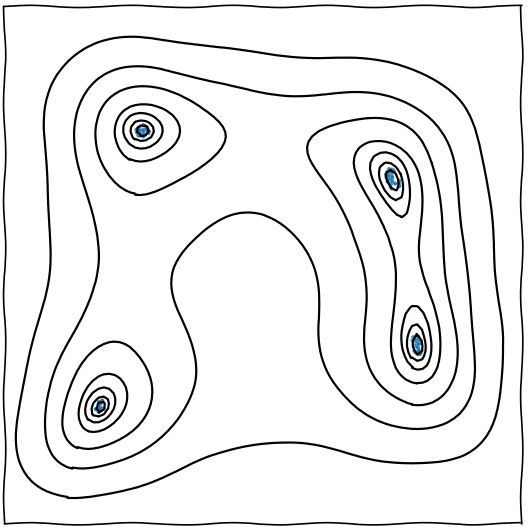
Nested sampling



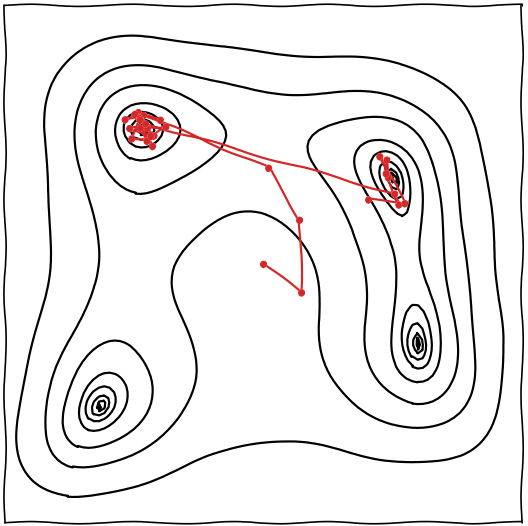
MCMC



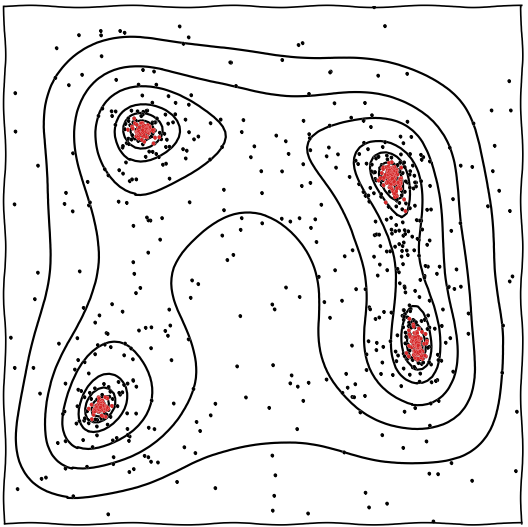
Nested sampling



MCMC

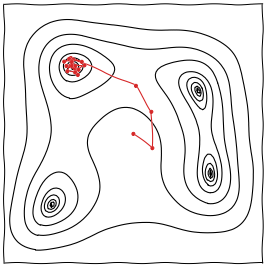


Nested sampling



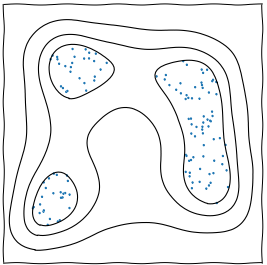
MCMC

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



Nested sampling

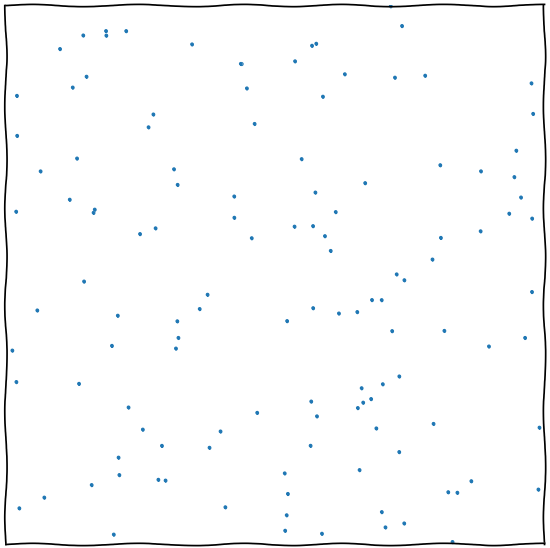
- Ensemble of "live points"
- Scans from prior to peak of likelihood
- Slower, no tuning required
- Parameter estimation, model comparison, tension quantification
- Channel capacity optimised for computing partition function



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

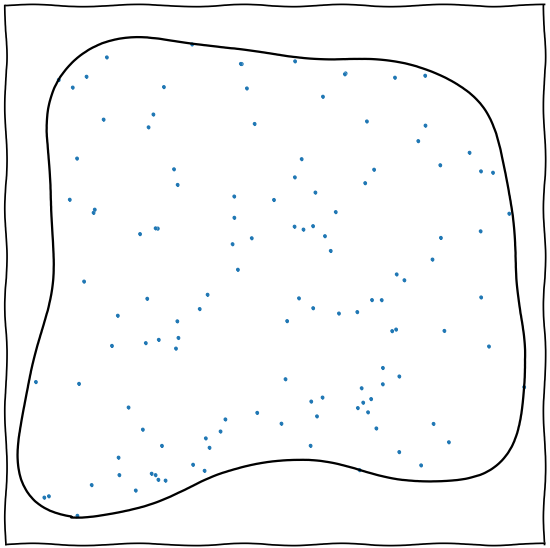
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

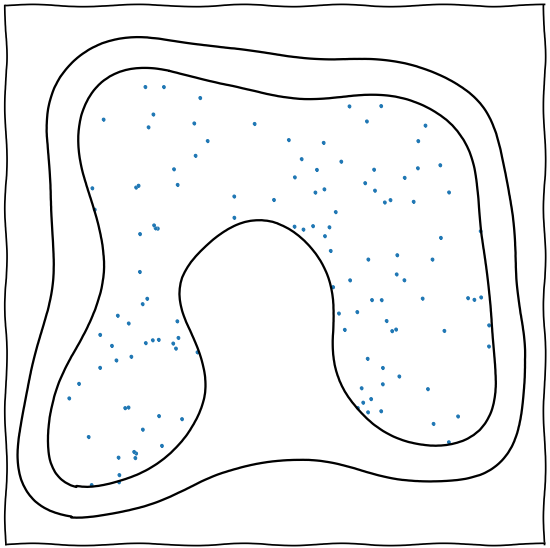
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

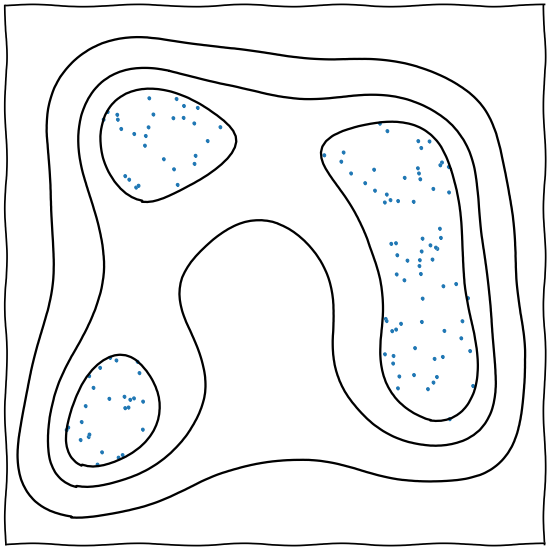
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

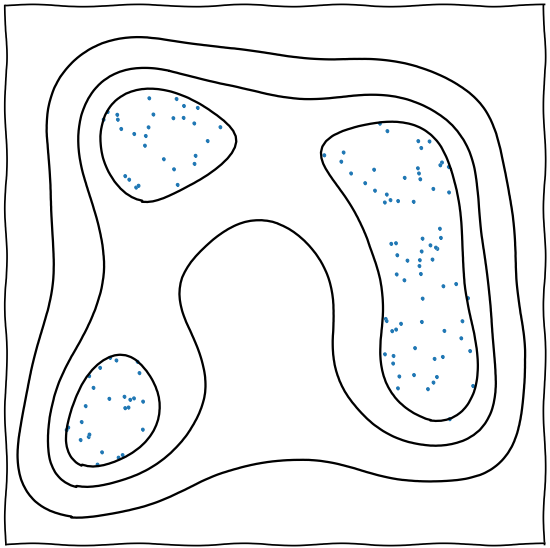
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



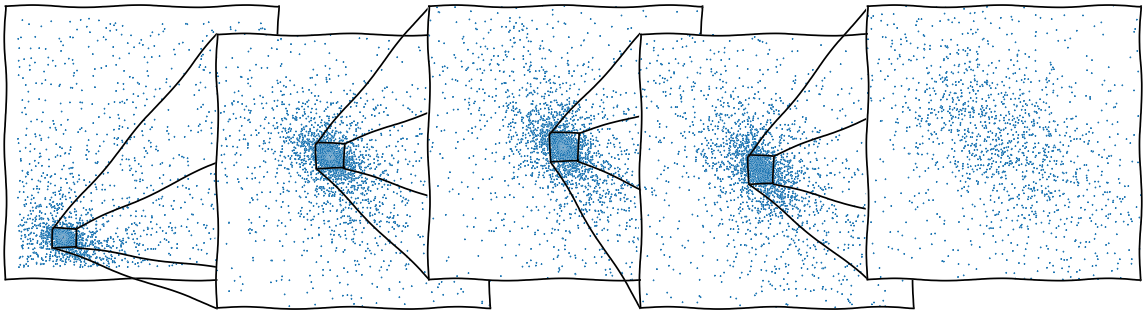
The nested sampling meta-algorithm: live points

- 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



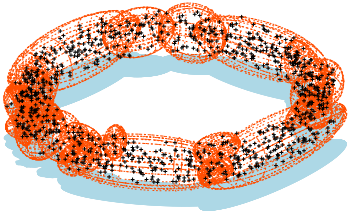
- At the end, one is left with a set of discarded "dead" points.
- Dead points have a unique scale-invariant distribution $\propto \frac{dV}{V}$.
- Uniform over original region, exponentially concentrating on region of interest (until termination volume).
- Good for training emulators (HERA [2108.07282]).

Applications

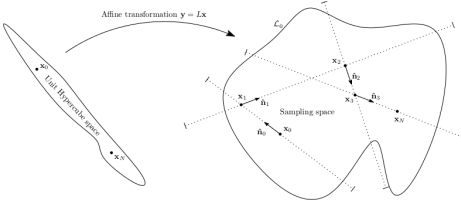
- training emulators.
- gridding simulations
- beta flows
- "dead measure"

Implementations of Nested Sampling [2205.15570](NatReview)

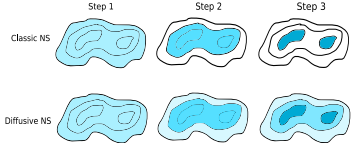
MultiNest [0809.3437]



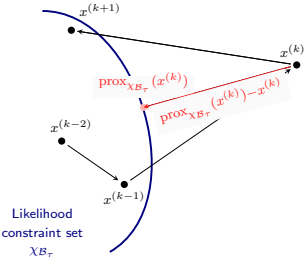
PolyChord [1506.00171]



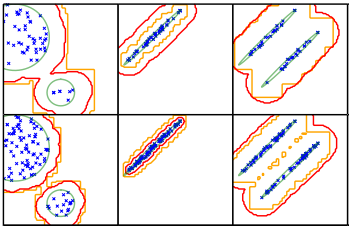
DNest [1606.03757]



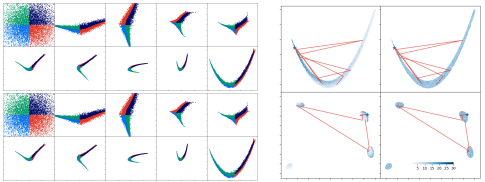
ProxNest [2106.03646]



UltraNest [2101.09604]



NeuralNest [1903.10860]



nessai [2102.11056]

nora [2305.19267]

jaxnest [2012.15286]

nautilus [2306.16923]

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 $d > d_0 \sim 10$.

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.

- Nested samplers usually come with:
  - *resolution* parameter n_{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 `eff` or `PolyChord` chain length n_{repeats}.

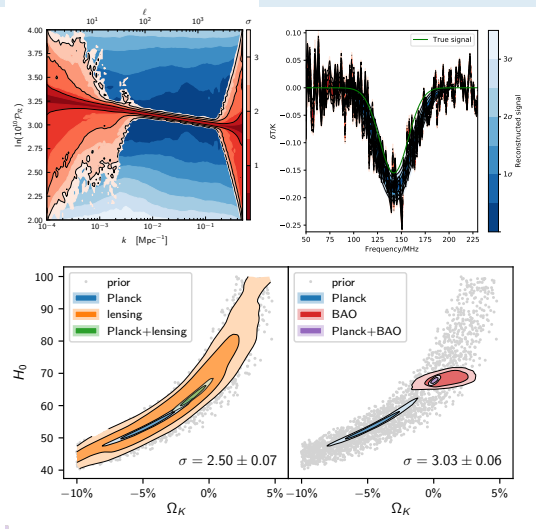
Applications of nested sampling

Cosmology

Adam Ormondroyd

PhD

- 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 ΛCDM 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.

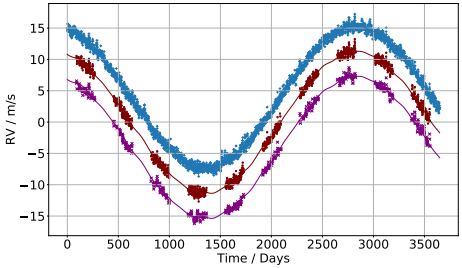


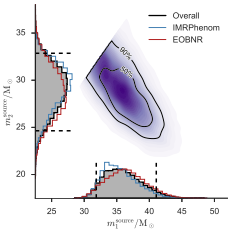
Applications of nested sampling

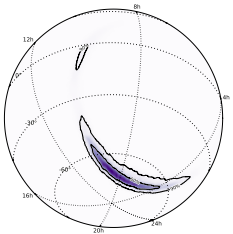
Astrophysics

Metha Prathaban
PhD

- 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





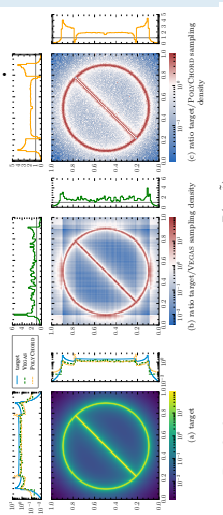


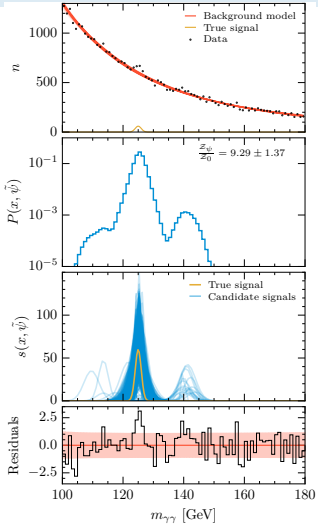
Applications of nested sampling

Particle physics

David Yallup
PDRA

- Nested sampling for cross section computation/event generation $\sigma = \int_\Omega d\Phi |\mathcal{M}|^2$.
- Nested sampling can explore the phase space Ω 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: $X \leftrightarrow p$, $\mathcal{L} \leftrightarrow \lambda$, $\theta \leftrightarrow x$.





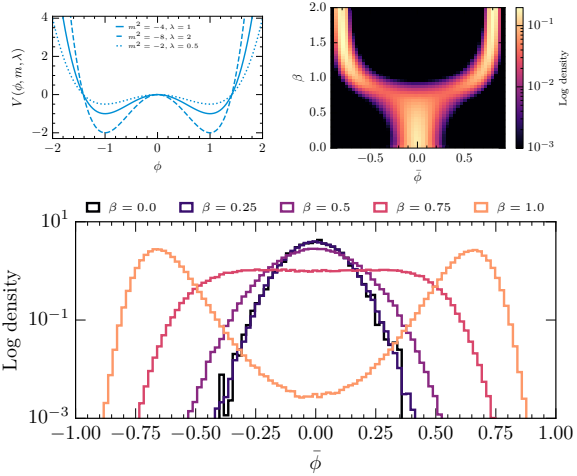
David Yallup

PDRA

- 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

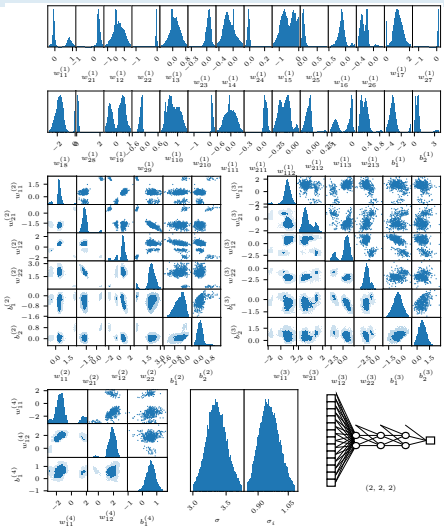


Applications of nested sampling

Machine learning

Kamran Javid
PDRA

- 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]

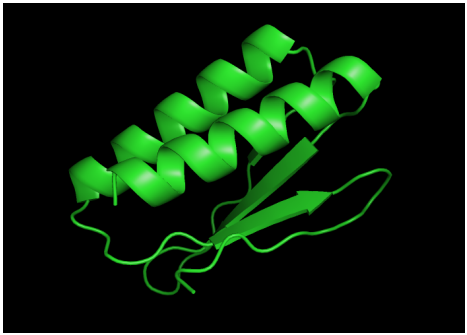


Applications of nested sampling

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.

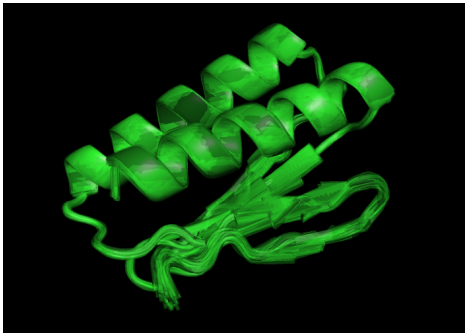


Applications of nested sampling

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.

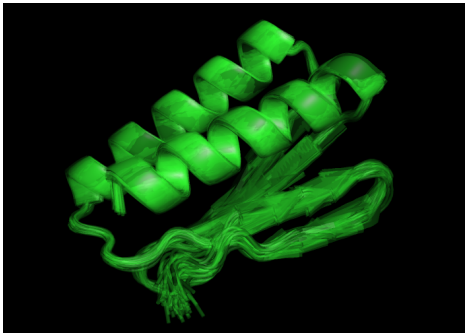


Applications of nested sampling

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.

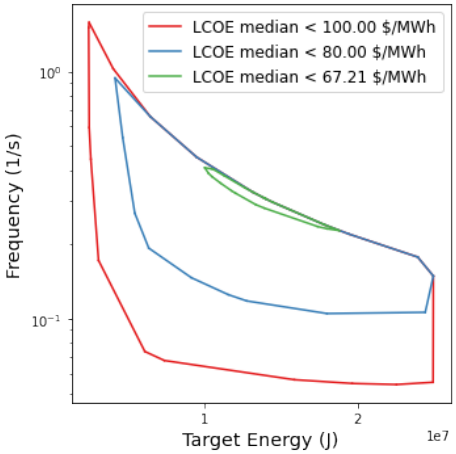


Applications of nested sampling

And beyond...

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.

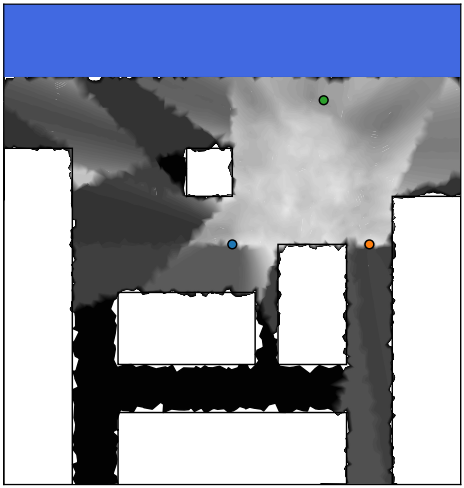


Applications of nested sampling

And beyond...

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.

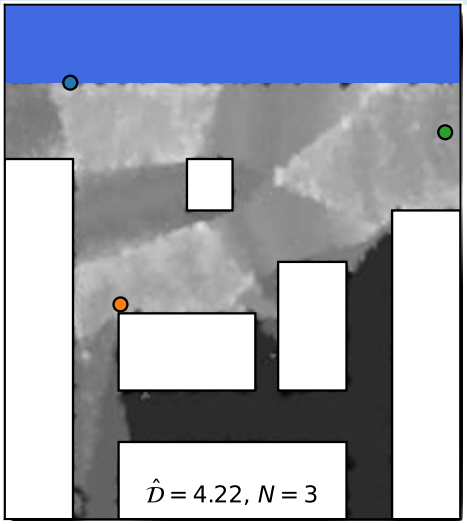


Applications of nested sampling

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.

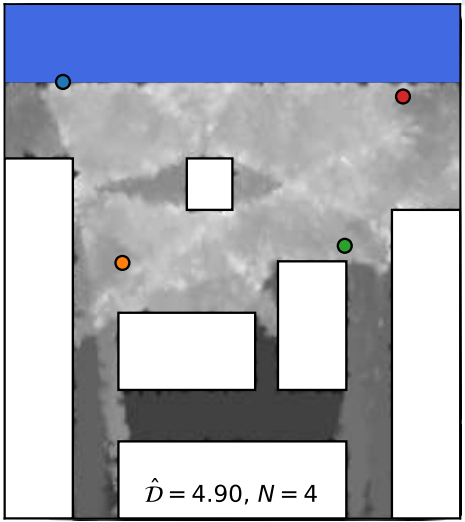


Applications of nested sampling

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.

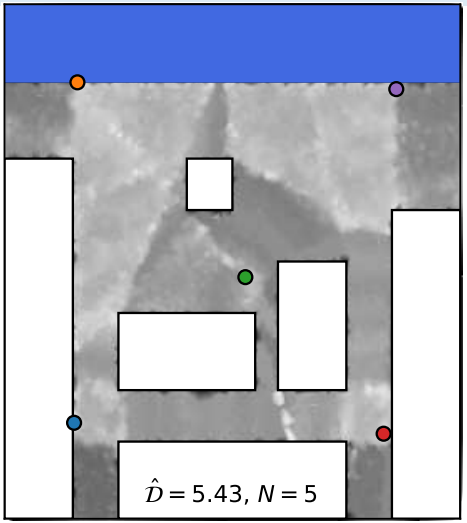


Applications of nested sampling

And beyond...

Thomas Mcaloone
PhD → Data Scientist

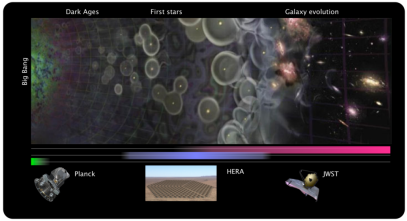
- Techniques have been spun-out (PolyChord Ltd) to:
- Protein folding
  - Navigating free energy surface.
  - Computing misfolds.
  - Thermal motion.
- Nuclear fusion reactor optimisation
  - multi-objective.
  - uncertainty propagation.
- Telecoms & DSTL research (MIDAS)
  - Optimising placement of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

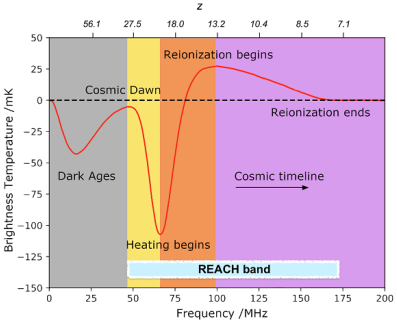


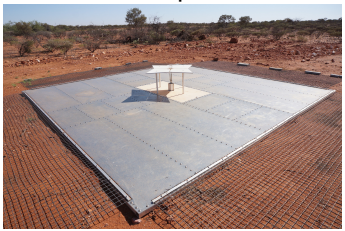
REACH: Global 21cm cosmology [2210.07409](NatAstro)

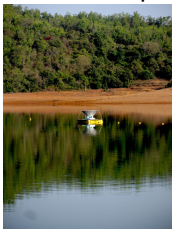
Ian Roque
PhD

- 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.



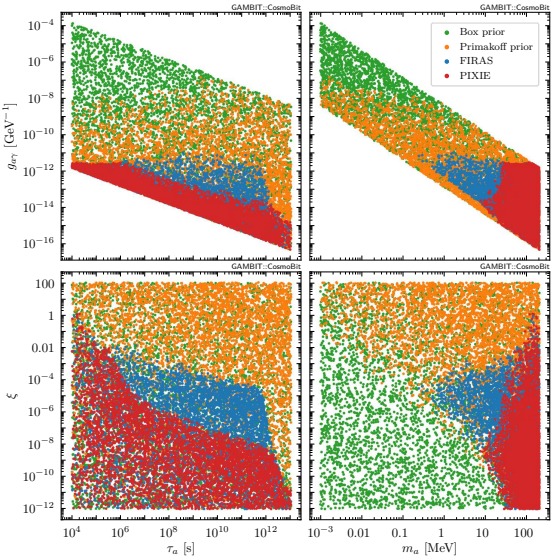






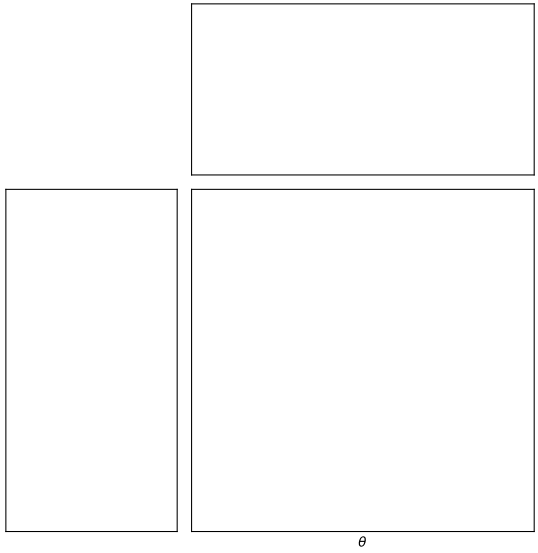
GAMBIT: combining particle physics & cosmological data

- Multinational team of particle physicists, cosmologists and statisticians.
- Combine cosmological data, particle colliders, direct detection, & neutrino detectors in a statistically principled manner [2205.13549].
- Lead Cosmo/Dark Matter working group [2009.03286].
- Nested sampling used for global fitting, and fine-tuning quantification [2101.00428]



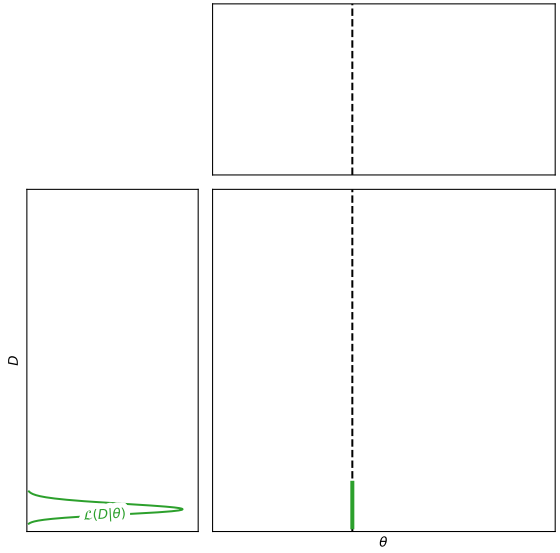
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



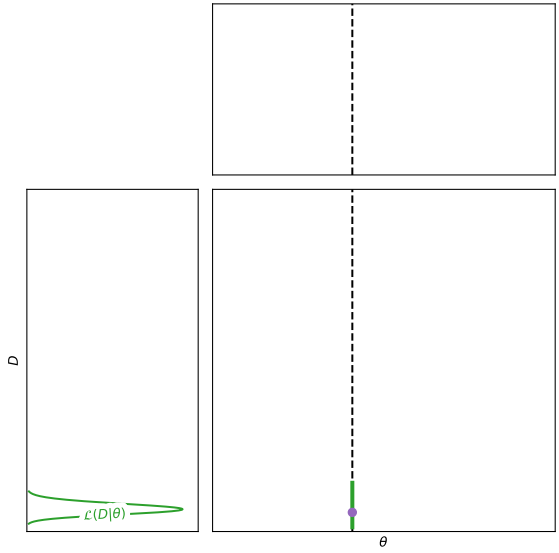
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



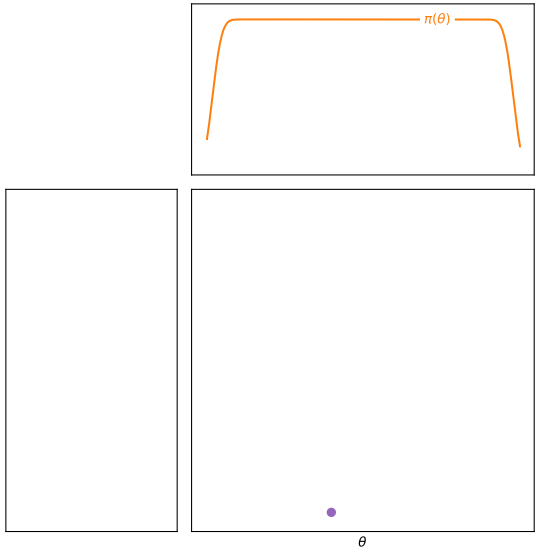
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



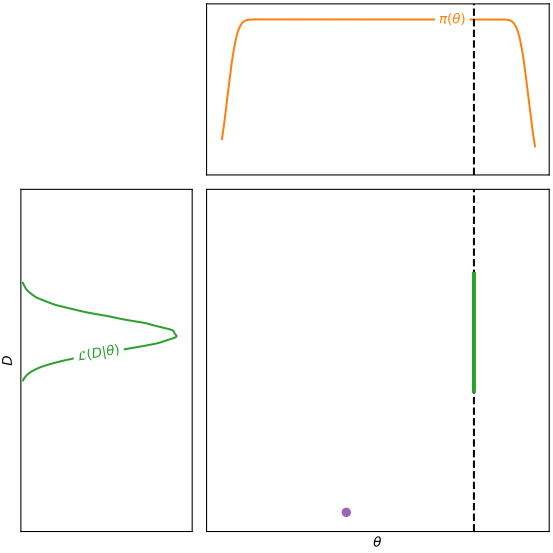
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



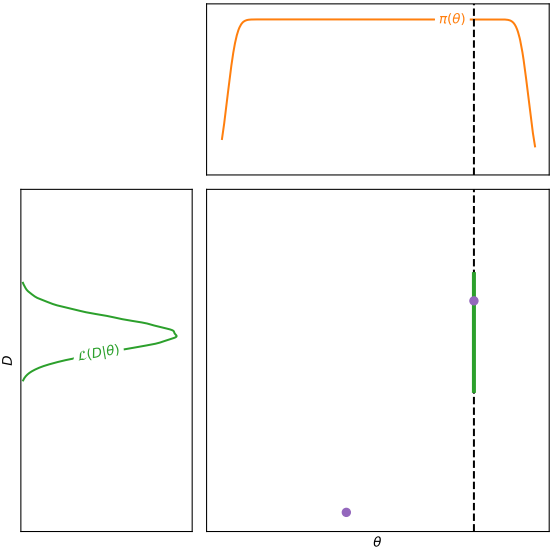
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



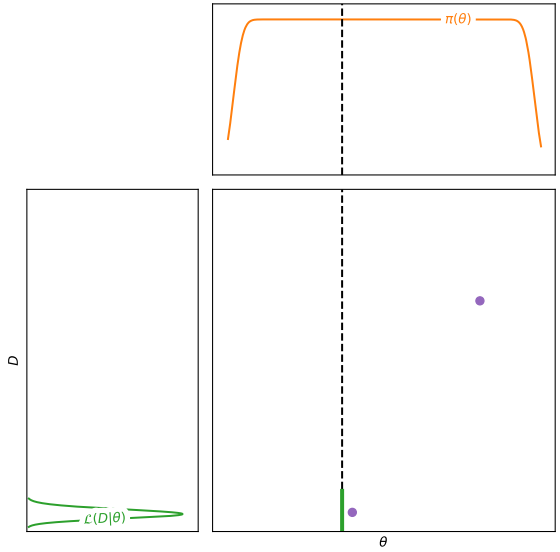
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



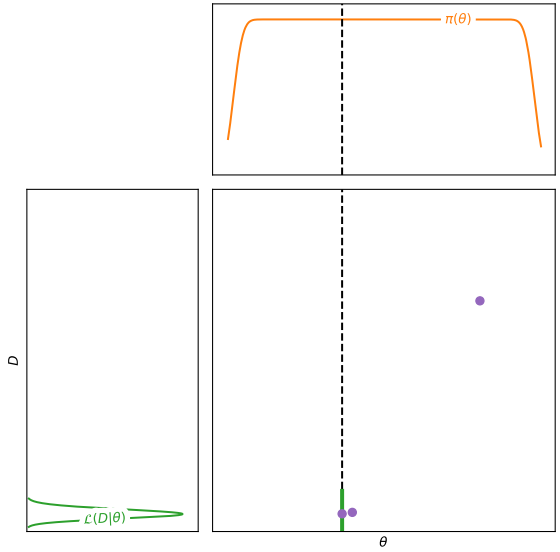
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



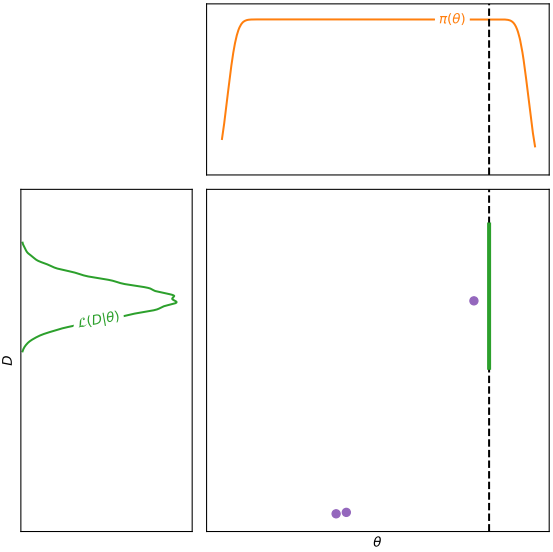
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



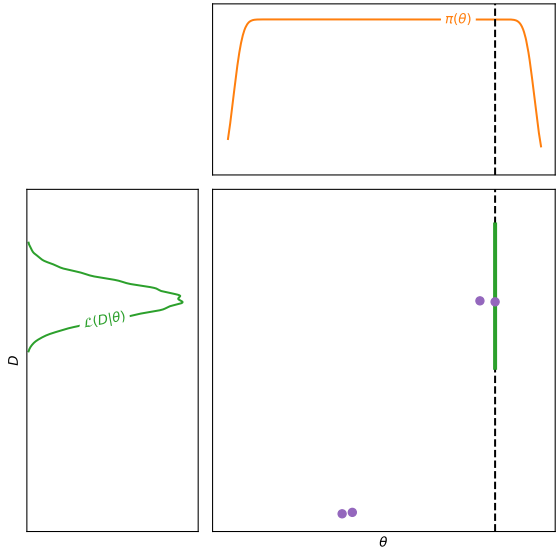
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



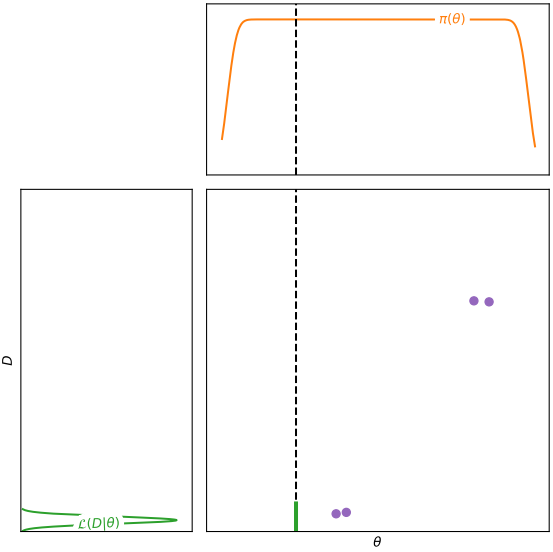
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



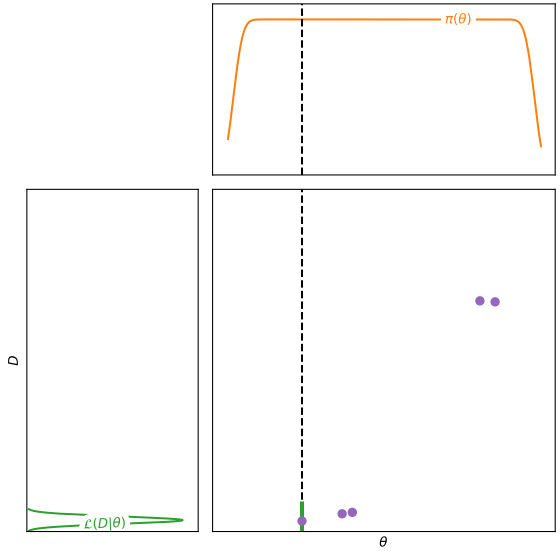
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



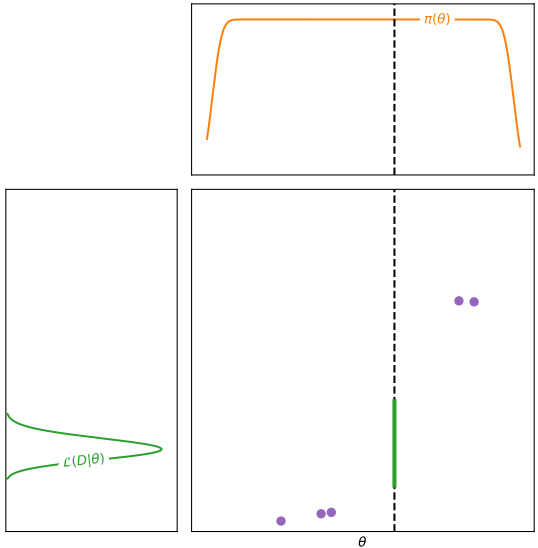
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



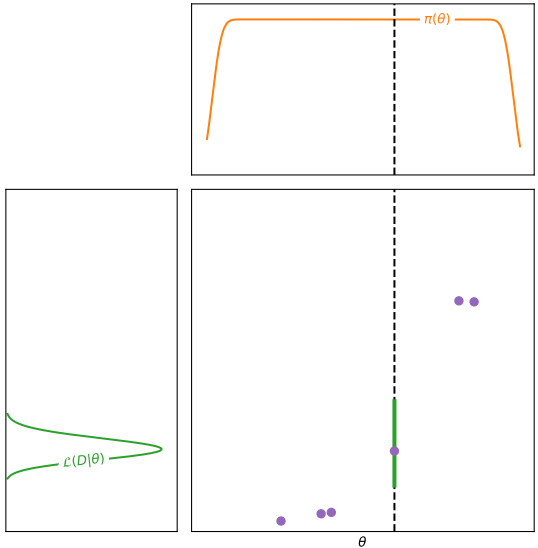
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



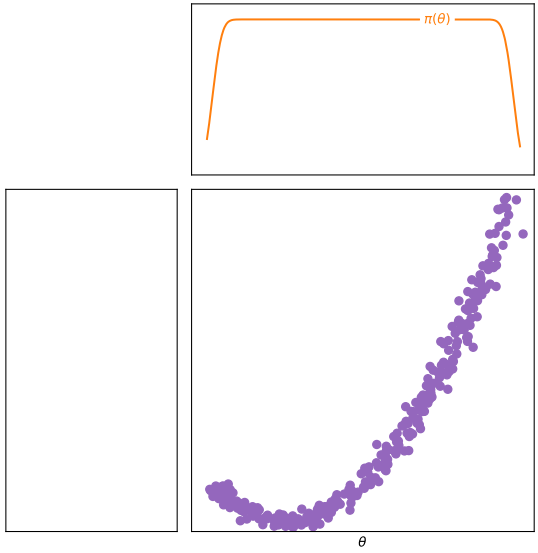
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



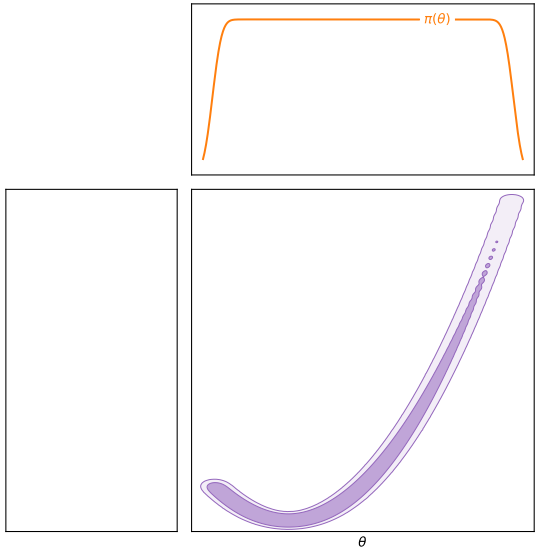
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



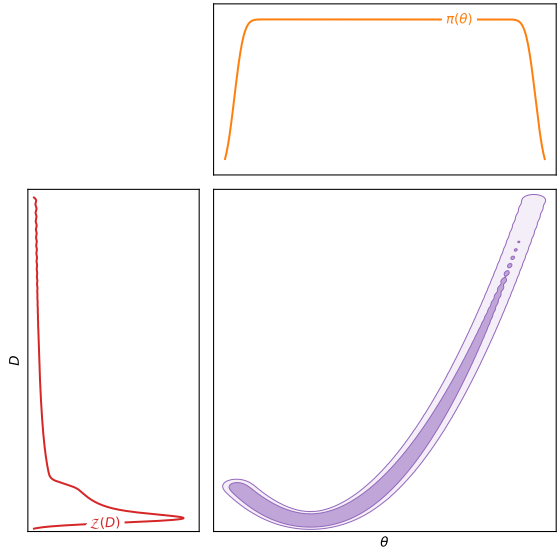
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



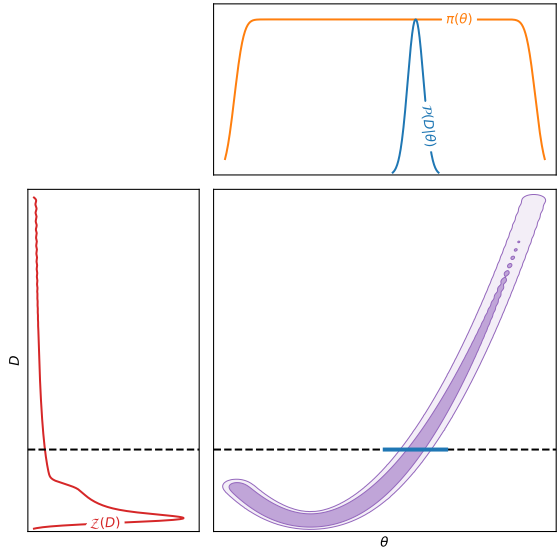
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



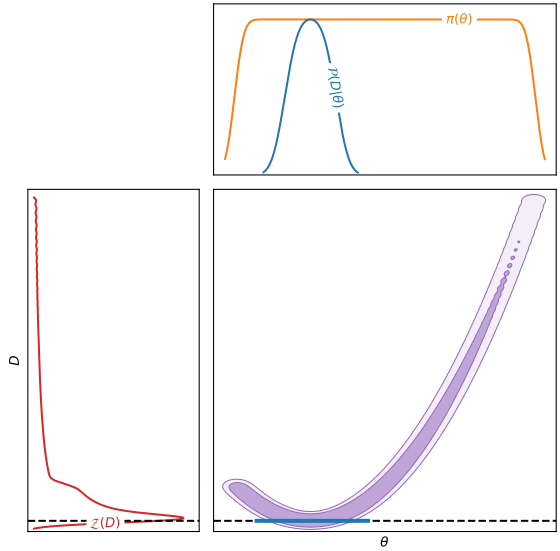
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



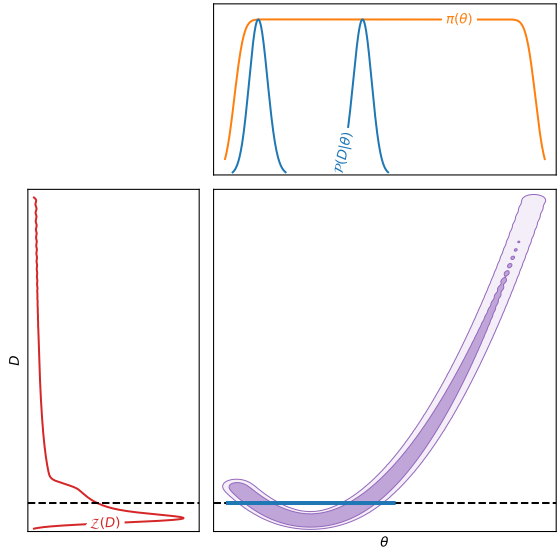
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



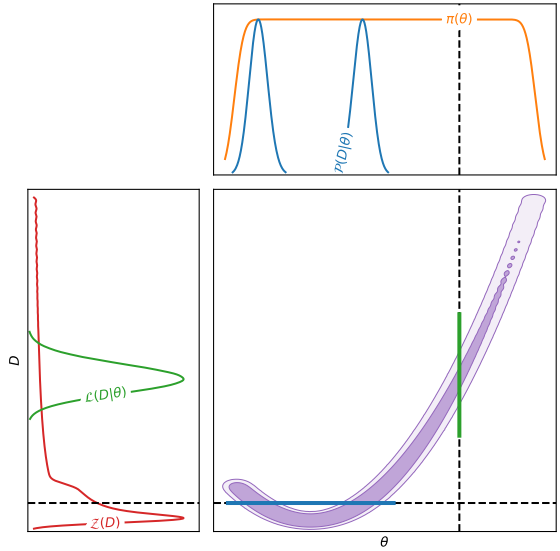
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



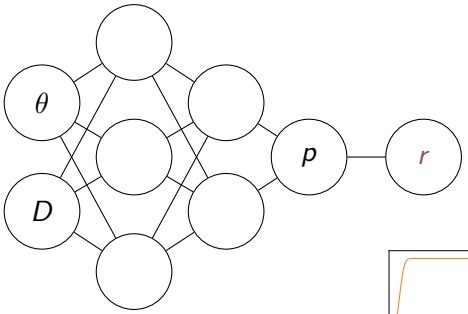
SBI: Simulation-based inference

- What do you do if you don't know $\mathcal{L}(D|\theta)$?
- If you have a simulator/forward model $\theta \to D$ 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`.



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].

TMNRE: Truncated Marginal Neural Ratio Estimation

swyft: github.com/undark-lab/swyft

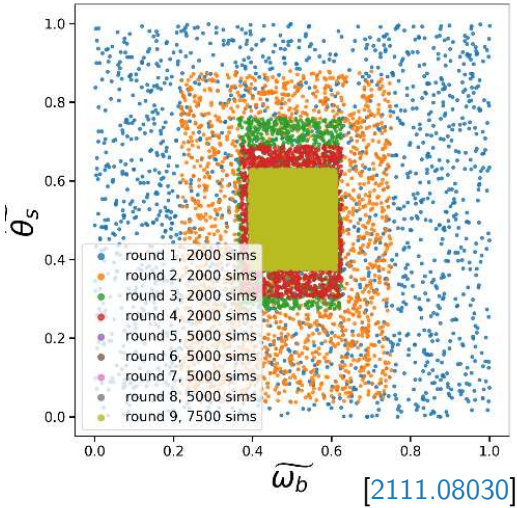
- Two tricks for practical NRE:

Marginalisation

- Only consider one or two parameters at a time.
- Fine if your goal is to produce triangle plots.
- Problematic if information is contained jointly in more than two parameters.

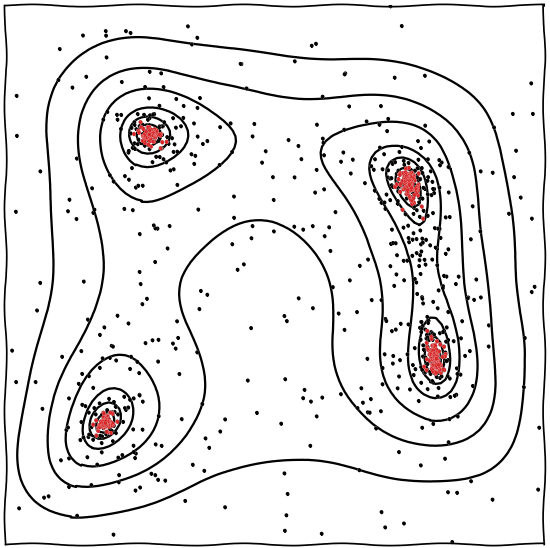
Truncation

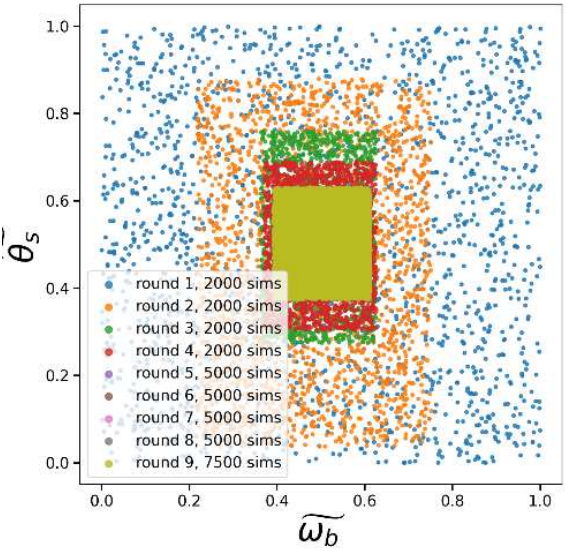
- focus parameters θ on a subset of the prior which reproduces observed data D_{obs}
- region is somewhat arbitrary (usually a box)
- not amortised, sounds a bit like ABC



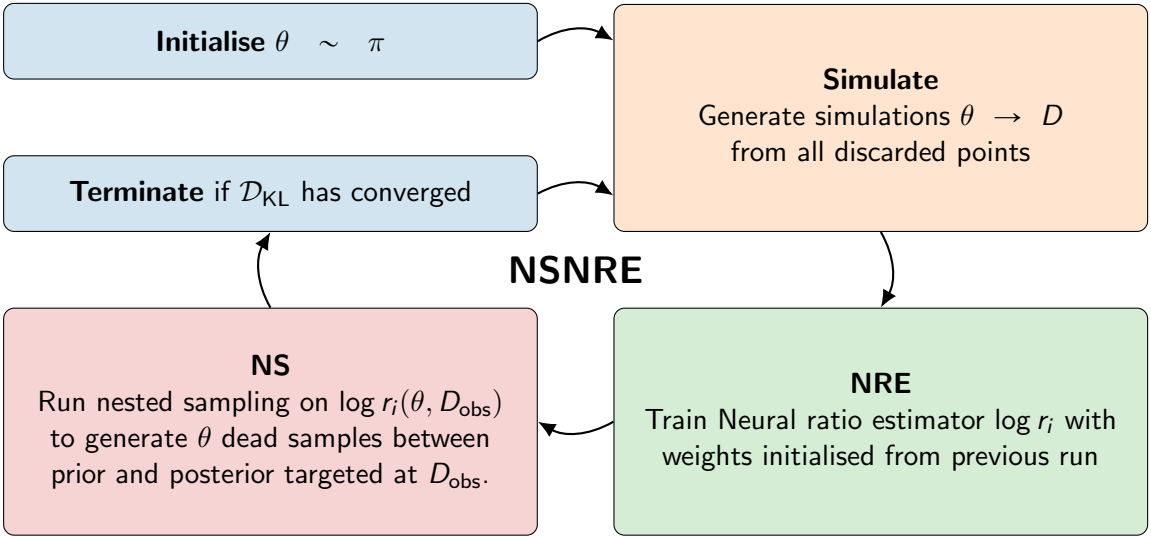
[2111.08030]

Similarities





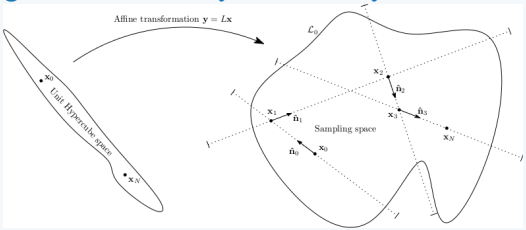
Sequential NRE with nested sampling



PolySwyft

PolyChord

github.com/PolyChord/PolyChordLite



- Widely used high-performance nested sampling tool (implementing slice sampling & clustering in MPI Fortran)

Swyft

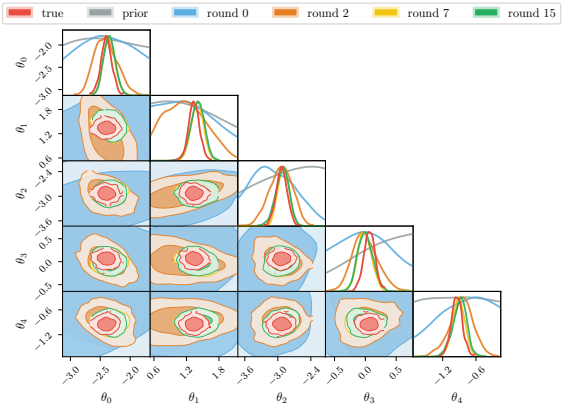
github.com/undark-lab/swyft

- Widely used TMNRE tool in cosmology/astrophysics.

However, NSNRE is general, and not specific to these choices.

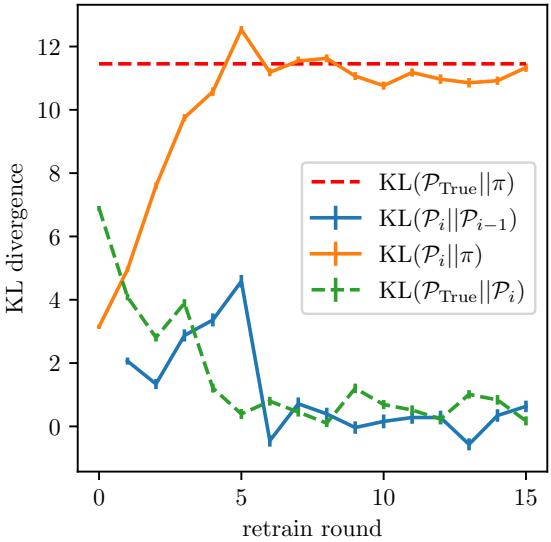
Convergence diagnostics

- Example for a $n = 5$ dimensional parameter space, with $d = 100$ data points, (`lsbi` gaussian mixture model).
- This is the regime for cosmological scale problems.
- To determine convergence we track:
  - The change in KL divergence between rounds (blue), and check when this goes to zero.
  - The total KL divergence between prior and posterior estimate (orange), and check when this levels off (ground truth in red).
  - Also shown is the KL divergence between the estimate and the ground truth (green).



Convergence diagnostics

- Example for a $n = 5$ dimensional parameter space, with $d = 100$ data points, (`1sbi` gaussian mixture model).
- This is the regime for cosmological scale problems.
- To determine convergence we track:
  - The change in KL divergence between rounds (blue), and check when this goes to zero.
  - The total KL divergence between prior and posterior estimate (orange), and check when this levels off (ground truth in red).
  - Also shown is the KL divergence between the estimate and the ground truth (green).



Conclusions

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 & machine learning.
- It's unique traits as the only numerical Lebesgue integrator mean with compute it will continue to grow in importance.