GPU Accelerated Nested Sampling

Will Handley
<wh260@cam.ac.uk>

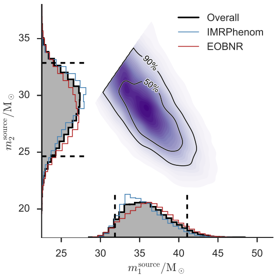
Royal Society University Research Fellow
Institute of Astronomy, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
willhandley.co.uk/talks

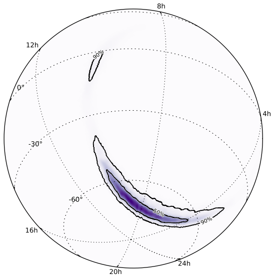
27th June 2025

A Case Study in Astrostatistics

The Challenge: GW170817

- **Gravitational wave detected**: Binary neutron star merger
- **Real-time parameter estimation**: 15+ dimensional space
- **Sky localization**: ~ 30 deg^2 uncertainty
- **EM counterpart follow-up**: Telescopes need targets within seconds to minutes





Statistical Requirements

- **Parameter estimation**: Masses, spins, distance, sky position
- **Model comparison**: Signal vs noise, waveform models

The Broader Astrostatistics Context

- **High-dimensional**: $d \sim 10^2$–10^3
- **Multimodal**: Competing physical models
- **Expensive likelihoods**: Complex sims
- **Model selection critical**: Which physics?

The Bayesian Inference Challenge

Parameter Estimation $P(\theta|D, M)$

- Posterior: $\mathcal{P}(\theta|D) \propto \mathcal{L}(D|\theta)\pi(\theta)$
- Need samples from $\mathcal{P}(\theta|D)$
- Standard approach: MCMC methods
- Well-solved problem in many cases

Model Comparison $P(M|D)$

- $\mathcal{Z} = \mathcal{P}(D|M) = \int \mathcal{L}(D|\theta)\pi(\theta)d\theta$
- Evidence/marginal likelihood/Bayes factor
- **Much harder to compute**
- MCMC doesn't estimate $\mathcal{Z}$ directly

Challenges for Modern Science

- **High dimensions**: $d \sim 10^2$–10^3
- **Natural, relevant multimodality**: Multiple acceptable answers need investigation
- **Computational cost**: Complex forward models
- **Model selection**: Which physics to include?

Key Insight:
Need methods that compute *both* posterior samples *and* evidence

Sampling Methods for Bayesian Inference

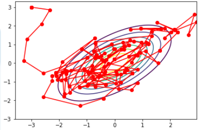
Single-Chain MCMC

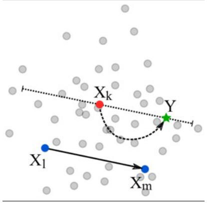
- **Metropolis-Hastings**: Simple, widely used (PyMC)
- **HMC/NUTS**: Gradient-based, efficient (Stan, BlackJAX)
- Fast for unimodal well-conditioned problems, no evidence

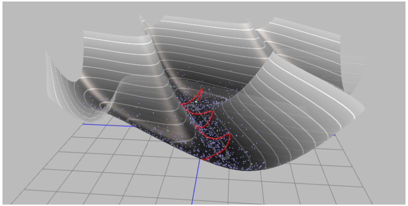
Ensemble Methods

- **Affine-invariant**: emcee, zeus
- **Sequential Monte Carlo**: Tempering, annealing
- can struggle with multimodality, some estimate evidence

emcee
the mcmc hammer

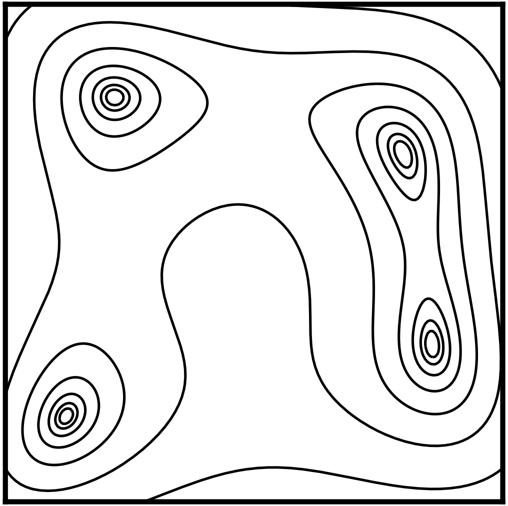




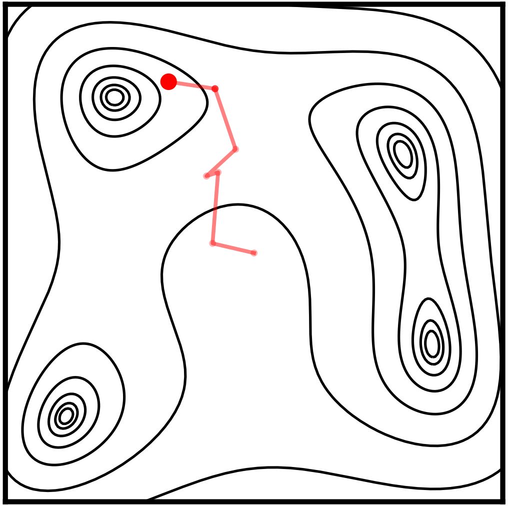


Nested sampling:
Unique in targeting evidence computation directly

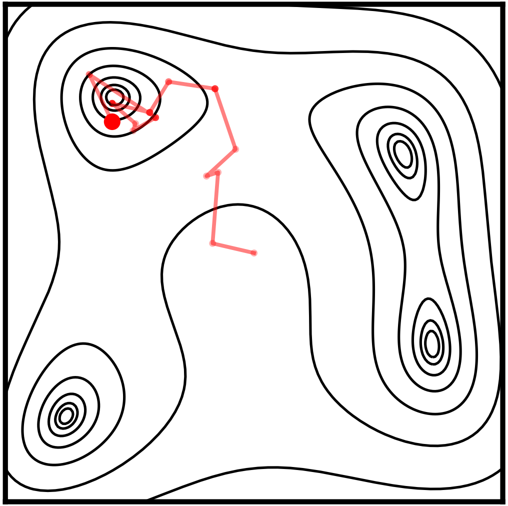
MCMC



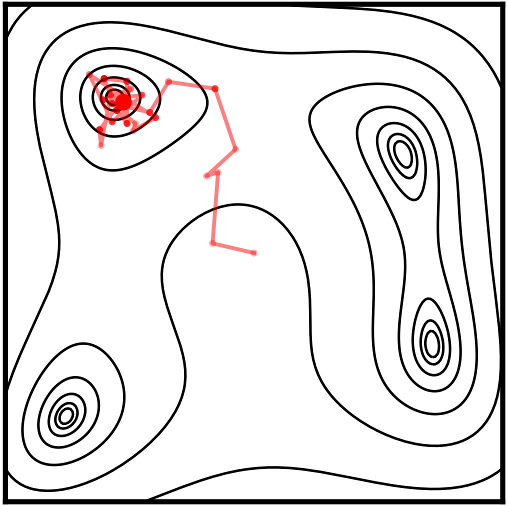
MCMC



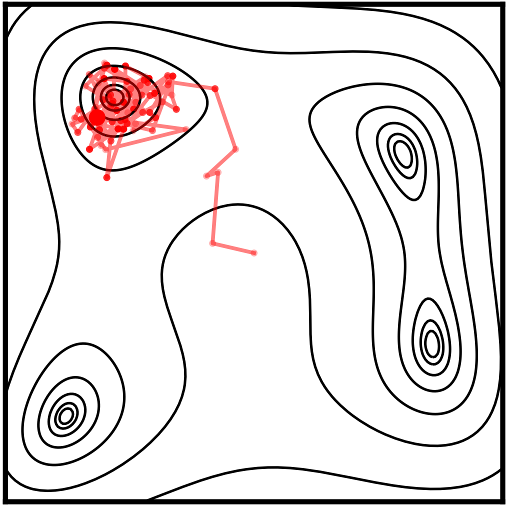
MCMC



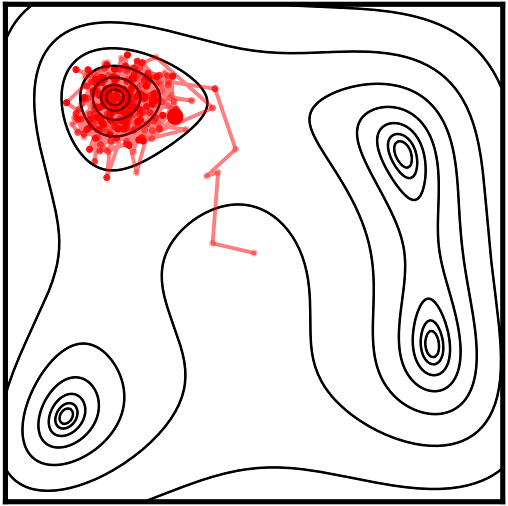
MCMC



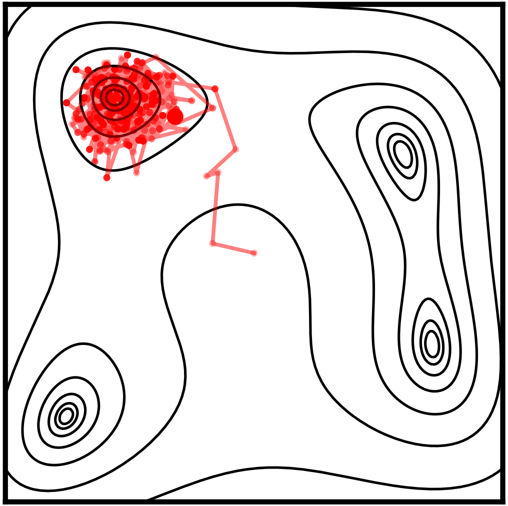
MCMC



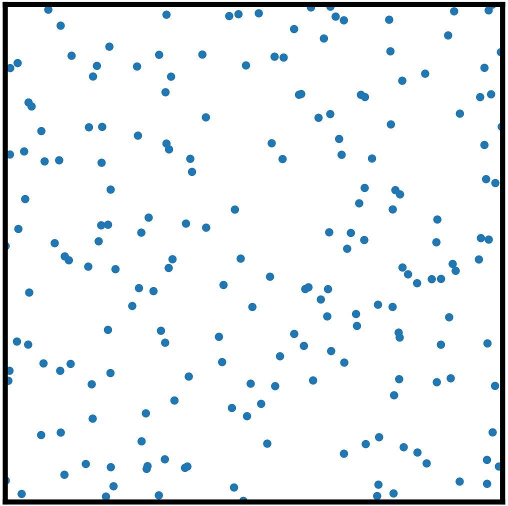
MCMC



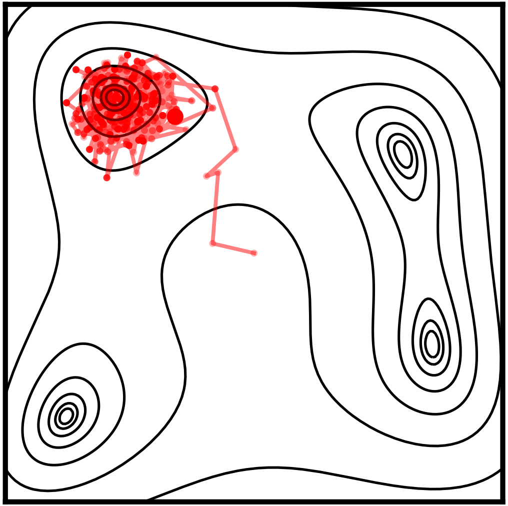
MCMC



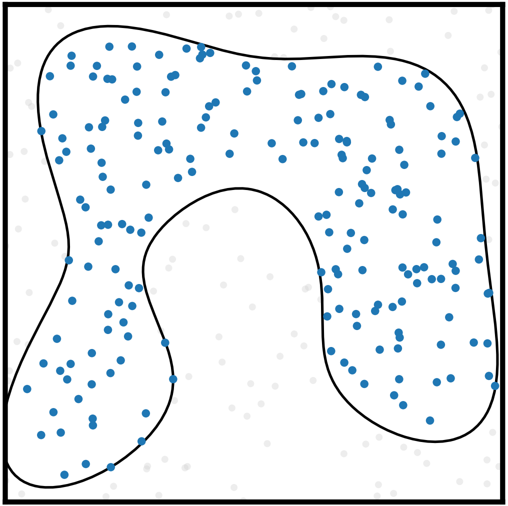
Nested sampling



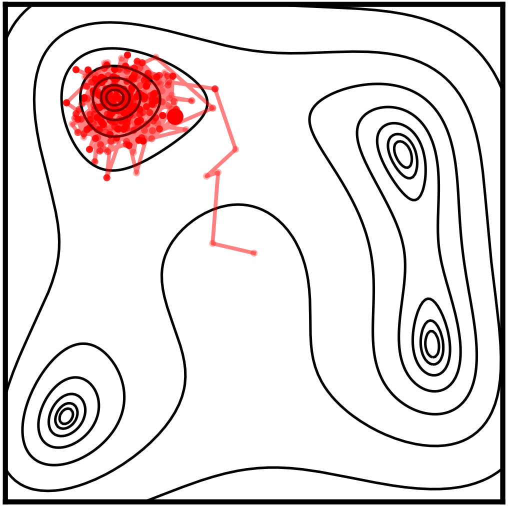
MCMC



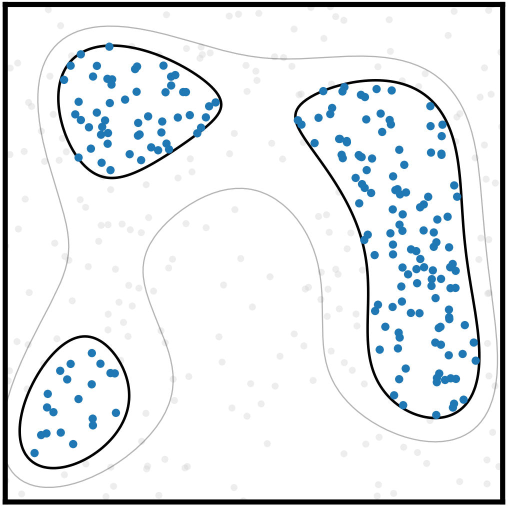
Nested sampling



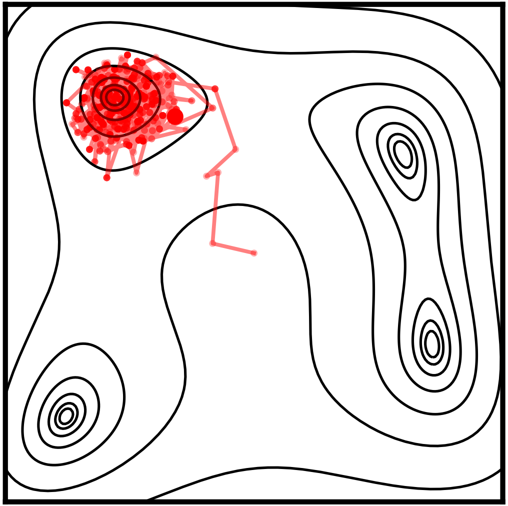
MCMC



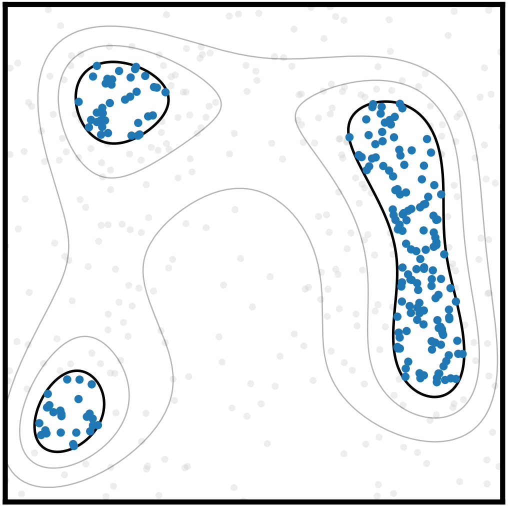
Nested sampling



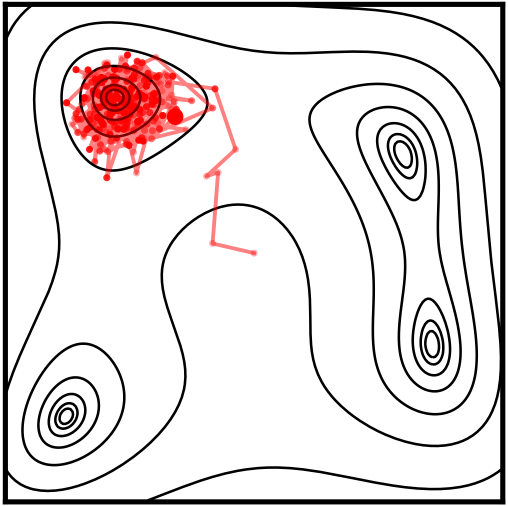
MCMC



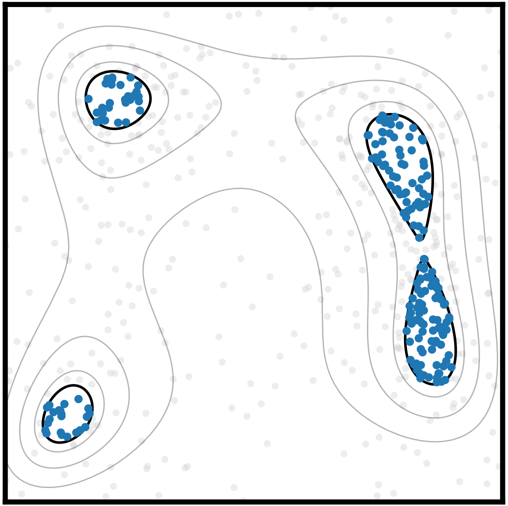
Nested sampling



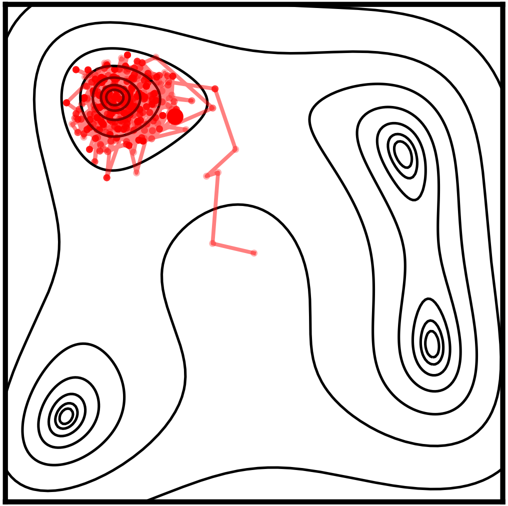
MCMC



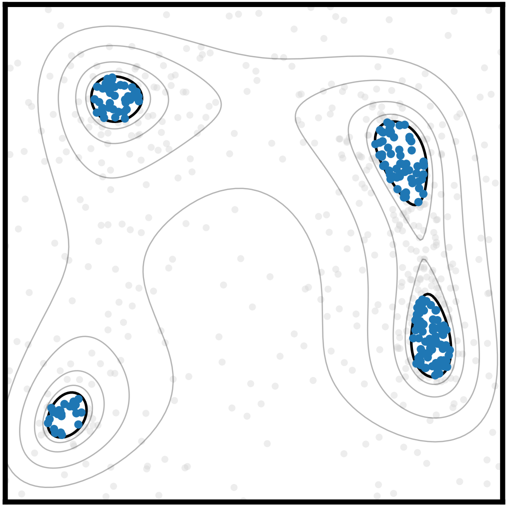
Nested sampling



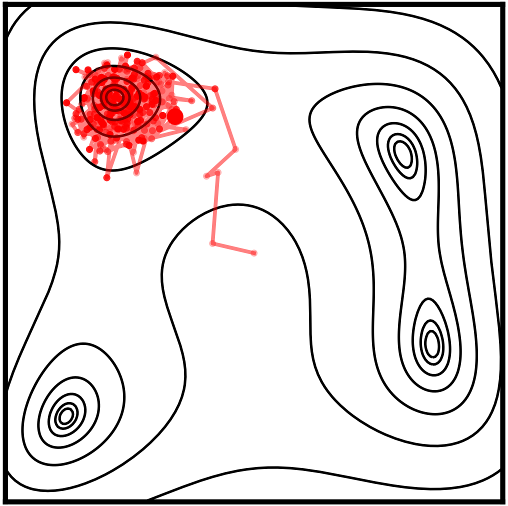
MCMC



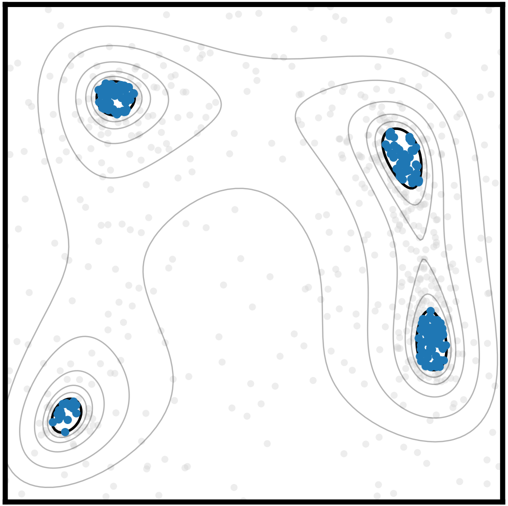
Nested sampling



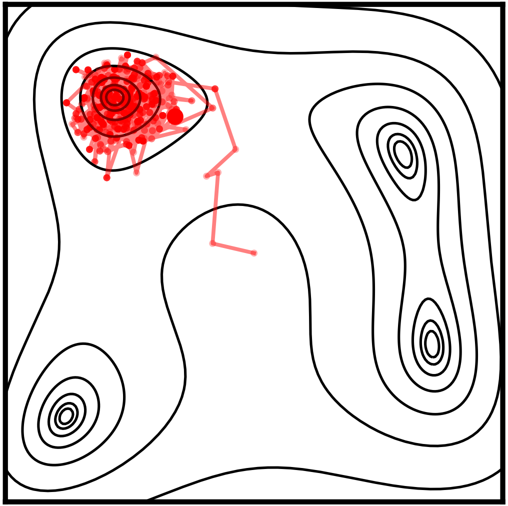
MCMC



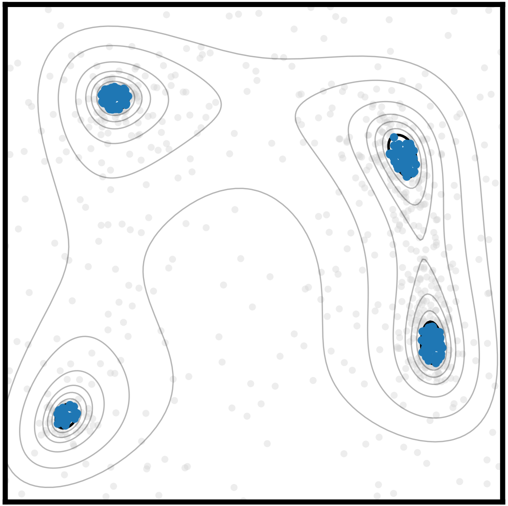
Nested sampling



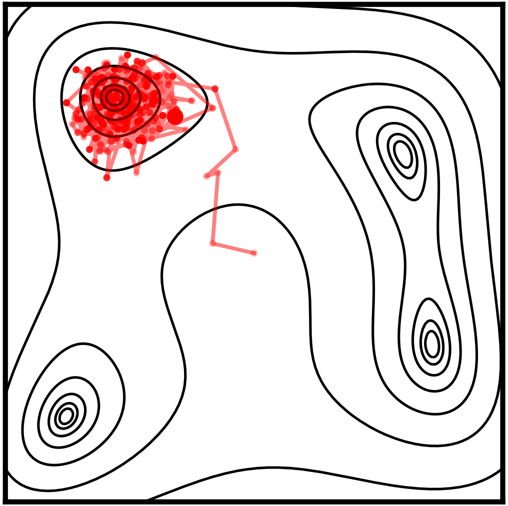
MCMC



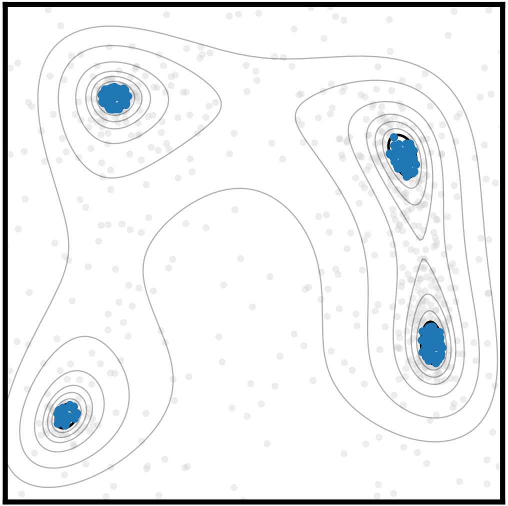
Nested sampling



MCMC

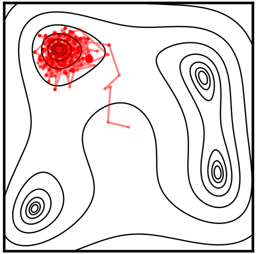


Nested sampling



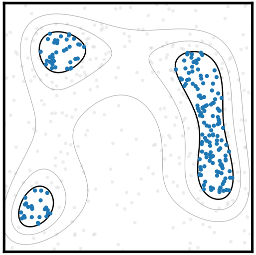
MCMC

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



Nested sampling

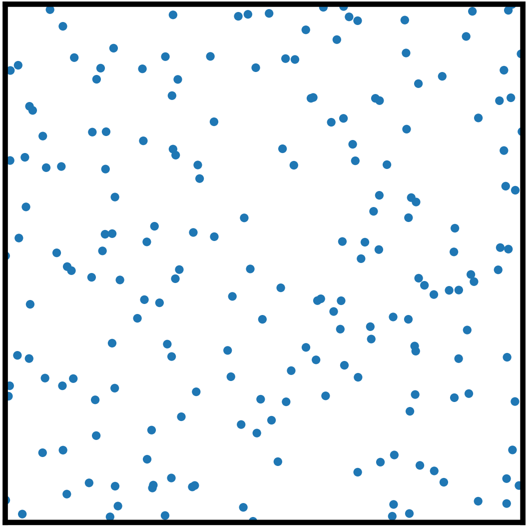
- Ensemble of "live points"
- Scans from prior to peak of likelihood
- Slower, no tuning required
- Parameter estimation, model comparison
- Channel capacity optimised for computing partition function



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

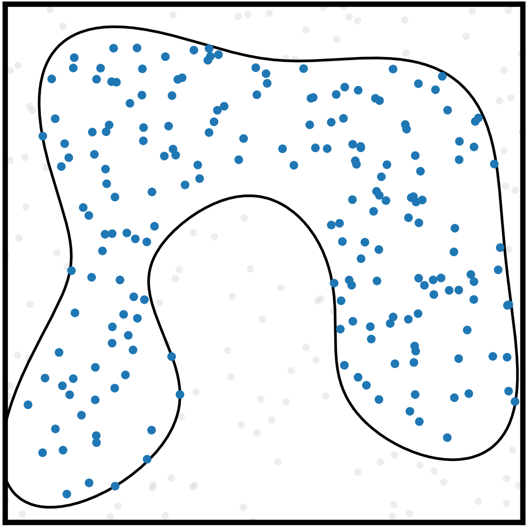
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \qquad V_i = V_0 e^{-i/n}$$



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

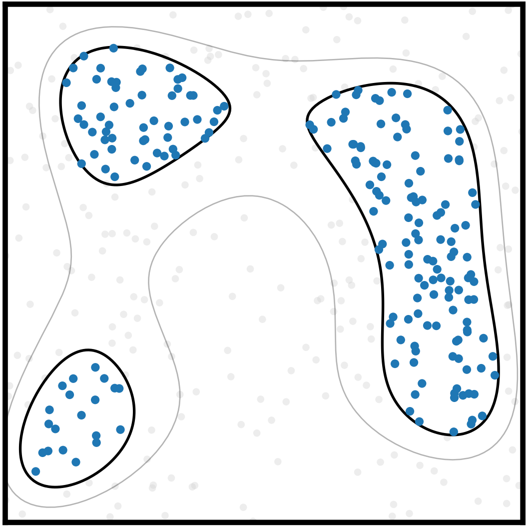
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \qquad V_i = V_0 e^{-i/n}$$



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

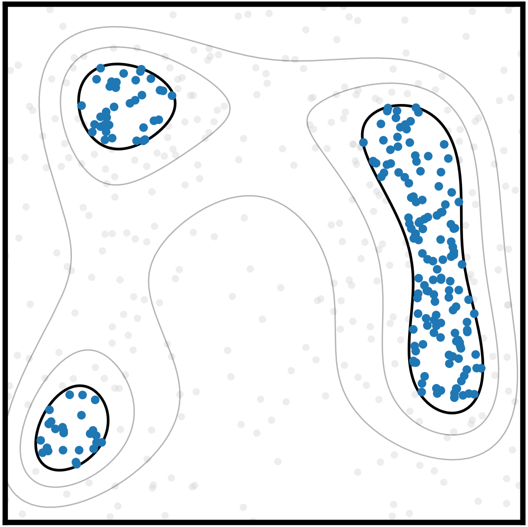
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \qquad V_i = V_0 e^{-i/n}$$



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

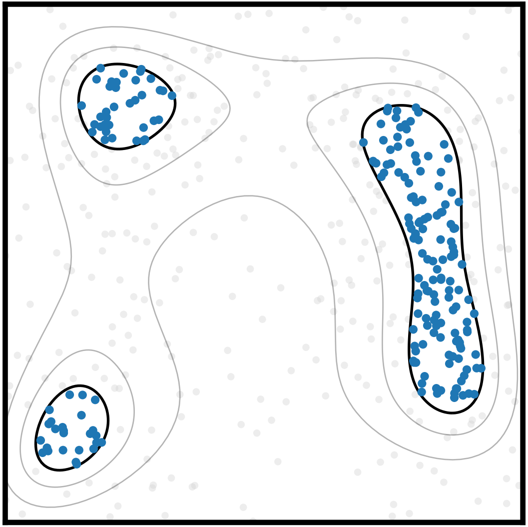
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \qquad V_i = V_0 e^{-i/n}$$



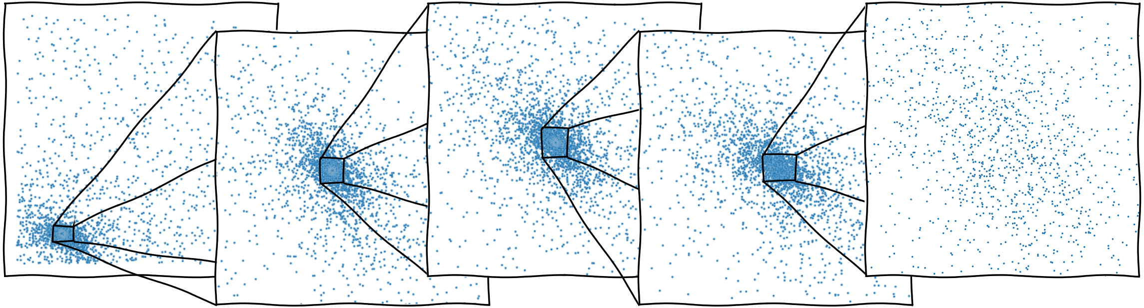
The nested sampling meta-algorithm: live points

- 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

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



The nested sampling meta-algorithm: dead points



- At the end, left with a set of discarded “dead points”.
- Dead points have a unique scale-invariant distribution $\propto \frac{dV}{V}$.
- Each dead point gets a **posterior weight**: $w_i = \mathcal{L}_i \Delta V_i$

Key Output

- **Posterior samples** θ_i
  weight $w_i = \mathcal{L}_i \Delta V_i$
- **Evidence** $\mathcal{Z} = \sum_i w_i$

Nested Sampling as Partition Function Calculator $\log \mathcal{Z}(\beta)$

The Key Insight

- Nested sampling directly estimates the **density of states**:

$$g(\mathcal{L}) = \int \delta(\mathcal{L}(\theta) - \mathcal{L})\pi(\theta)d\theta$$

- This is the **partition function** at inverse temperature β:

$$\mathcal{Z}(\beta) = \int g(\mathcal{L})\mathcal{L}^\beta d\mathcal{L}$$

- Evidence is special case: $\mathcal{Z} = \mathcal{Z}(\beta = 1)$
- **In practice**: $\mathcal{Z}(\beta) \approx \sum_i \mathcal{L}_i^\beta \Delta V_i$

Statistical Physics Connection

- **Canonical ensemble**: $p(\theta|\beta) \propto \mathcal{L}(\theta)^\beta \pi(\theta)$
- **Free energy**: $\beta F = -\log \mathcal{Z}$
- **Internal energy**: $U = -\frac{\partial \log \mathcal{Z}}{\partial \beta}$
- **Heat capacity**: $C = \frac{\partial U}{\partial \beta}$

Nested sampling provides the fundamental thermodynamic quantities
for any probabilistic model

Why GPUs? The Future of High-Performance Computing

GPU Advantages (Often Confused!)

- **Massive Parallelization**:
  - 1000s of cores vs 10s on CPU
  - Perfect for ensemble algorithms
  - Vectorization across particles/chains
  - Independent likelihood evaluations
- **Automatic Differentiation**:
  - GPU-accelerated gradients "for free"
  - JAX/PyTorch ecosystem make this possible
  - Essential for modern optimization

The HPC Reality

- **Future HPC is GPU dominated**:
- **Legacy CPU codes becoming obsolete**

Apples-to-Apples comparison

- Quantifying GPU advantage
  - GPUs $40\times$ more expensive to rent
  - GPUs $100\times$ rarer in HPC allocations
- Sometimes you don't care about walltime.

Why BlackJAX? Unified GPU Framework for Bayesian Inference

The Fragmentation Problem

- **Scattered ecosystem**: `MultiNest`, `PolyChord`, `dynesty`, `UltraNest`, `nautilus`, `nessai`, ...

BlackJAX Solution

- **Community JAX codebase**
- **Fair benchmarking** with identical GPU infrastructure
- **Composable algorithms** with shared components
- **Modern ML ecosystem integration**

Algorithm-Hardware Matching

- **Ensemble methods $\leftrightarrow$ GPU parallelization**:
  - Nested sampling: 100-1000 live points
  - SMC: 1000s of particles
  - Embarrassingly parallel operations
- **Scientific problems are compute-bound**:
  - Unlike ultra-large DL models
  - GPU memory rarely limiting
  - Perfect match for vectorization

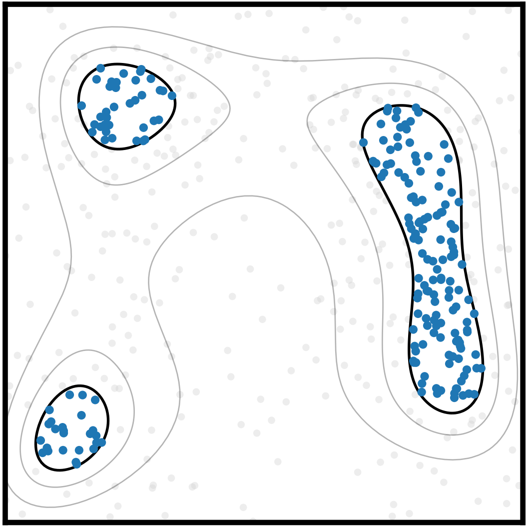
David Yallup
PDRA

Nested Sampling Meta-Algorithm

- **Framework is kernel-agnostic**:
  - Original: Metropolis Hastings (Skilling 2006)
  - MultiNest: rejection ellipsoids
  - PolyChord: slice sampling
  - nessai/nautilus: ML techniques

Our Choice: Slice Sampling

- **First scalable generic solution** in BlackJAX
- **No tuning required** (unlike MCMC proposal matrices)
- **Robust across dimensions** & problem types



GW150914 Binary Black Hole Merger

Metha Prathaban
PhD

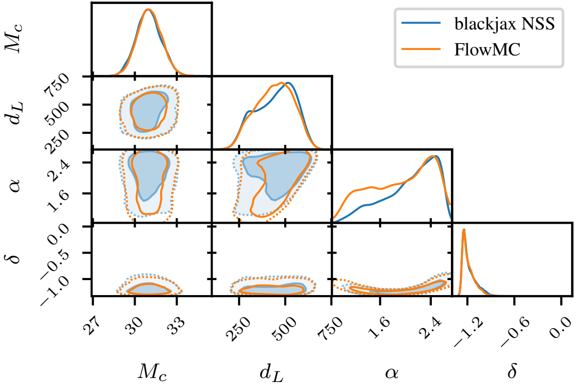
Performance on Real Data

- **BlackJAX GPU-NS**: 207 seconds (1 GPU)
- **FlowMC (GPU MCMC)**: 742 seconds (1 GPU)
- **Bilby/Dynesty**: 2 hours (400 CPUs)

Orders of magnitude speedup over CPU
Comparable to other GPU-native methods

Key Achievement

- Nested sampling now competitive on GPUs
- Direct evidence computation included



Good agreement between BlackJAX and FlowMC posteriors

CMB & Cosmic Shear Analysis

Toby Lovick

PhD

CMB Power Spectrum (6 params)

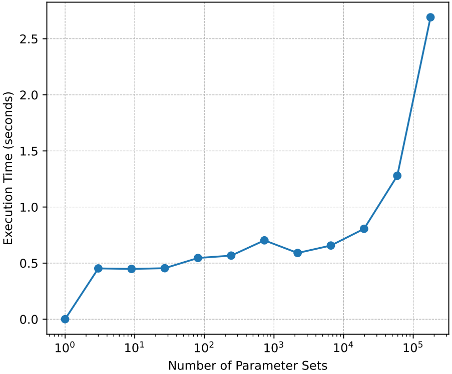
- **PolyChord (CPU)**: 1 hour
- **BlackJAX (GPU)**: 12 seconds

300× speedup

Cosmic Shear (37 params)

- **PolyChord (48 CPUs)**: 8 months
- **NUTS (12 A100 GPUs)**: 2 days
- **BlackJAX (1 A100 GPU)**: 4.5 hours

>1000× speedup vs CPU
10× speedup vs existing GPU approach[2405.12965]



CMB & Cosmic Shear Analysis

Toby Lovick
PhD

CMB Power Spectrum (6 params)

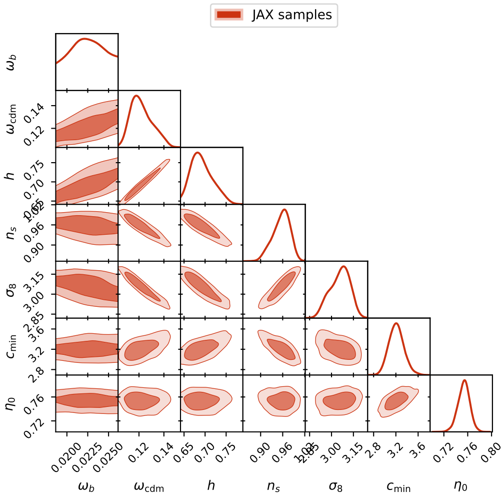
- **PolyChord (CPU)**: 1 hour
- **BlackJAX (GPU)**: 12 seconds

300× speedup

Cosmic Shear (37 params)

- **PolyChord (48 CPUs)**: 8 months
- **NUTS (12 A100 GPUs)**: 2 days
- **BlackJAX (1 A100 GPU)**: 4.5 hours

>1000× speedup vs CPU
10× speedup vs existing GPU approach[2405.12965]



The Real AI Revolution: LLMs as the Missing Piece

LLMs: The GPU Code Translator

- Automated translation: Fortran/C++ → JAX/PyTorch
- Bridges the gap between legacy science and modern hardware

The 80/20 Rule of Scientific Work

- **80% "boring" tasks**: forms, papers, grants, reviews, grading, code writing...
- **20% hard thinking**: Novel insights, experimental design, theory
- **AI's biggest impact**: Automating the 80%, not the 20%

Beyond Scientific Analysis

- **Common focus**: Using LLMs for analysis
- **Real transformation**: Automating workflow
- **Already happening**:
  - Grant writing assistance
  - Paper drafting and review
  - Code generation and debugging
  - Literature review automation

The Productivity Explosion

- **Quality control**: Becomes the limiting factor
- **Focus shift**: writing → critical thinking

Resources

- **Installation**: `pip install git+https://github.com/handley-lab/blackjax`
- **Documentation**: handley-lab.co.uk/nested-sampling-book

BlackJAX Implementation

- **BlackJAX**: [github:handley-lab/blackjax]
- **Nested sampling**: In PR to [github:blackjax-devs/blackjax] #755

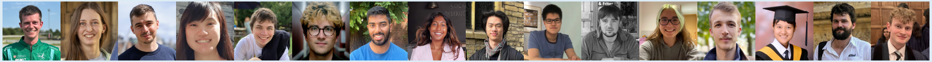
Theory & Background

- **Review papers**: [2205.15570], [2101.09675]
- **Original paper**: [Skilling (2006)]

Workshop & Learning

- **GPU Nested Sampling Workshop**: github.com/handley-lab/workshop
- **Interactive tutorials**: JAX, BlackJAX, GPU acceleration

Conclusions



github.com/handley-lab/group

- **Nested sampling is widely used** across physical sciences for parameter estimation and model comparison
- **BlackJAX provides GPU-native implementation** with 10×-100× speedups
- **JAX ecosystem integration** enables modern scientific workflows
- **Real applications** from gravitational waves to cosmology benefit immediately
- **The future is GPU-accelerated** scientific computing with AI integration