GPU Accelerated Bayesian Inference for Astronomy

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

22nd October 2025


UNIVERSITY OF CAMBRIDGE


Inference Across Science: Cosmology

Parameter estimation and model comparison for the universe

Parameter Estimation

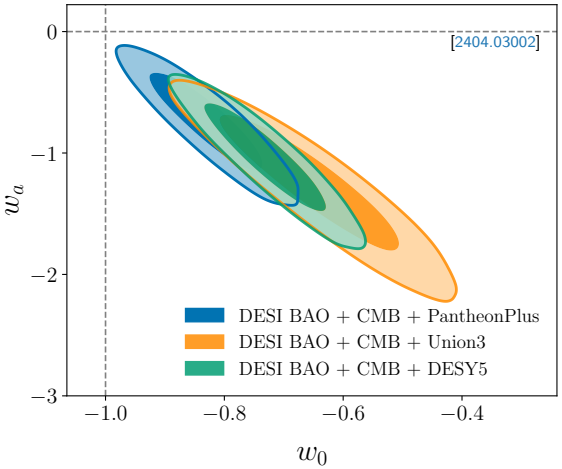
Goal: Measure cosmological parameters

- Hubble constant H_0, matter density Ω_m
- Dark energy equation of state w

Model Comparison

Goal: Determine best cosmological model

- ΛCDM vs wCDM vs $w_0 w_a$CDM
- Is dark energy evolving?



Inference Across Science: Gravitational Waves

Parameter estimation and model comparison for transient events

Parameter Estimation

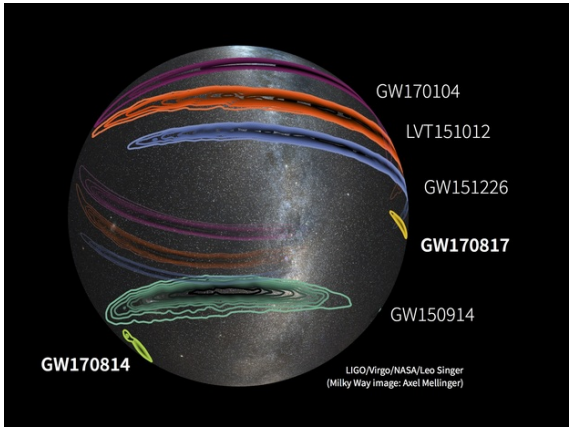
Goal: Localize the source on the sky

- Masses, spins, distance, sky position
- Critical for electromagnetic follow-up

Model Comparison

Goal: Determine the event type

- NS-NS vs NS-BH vs BH-BH
- Different physical implications



Inference Across Science: Exoplanets

Parameter estimation and model comparison for planetary systems

Parameter Estimation

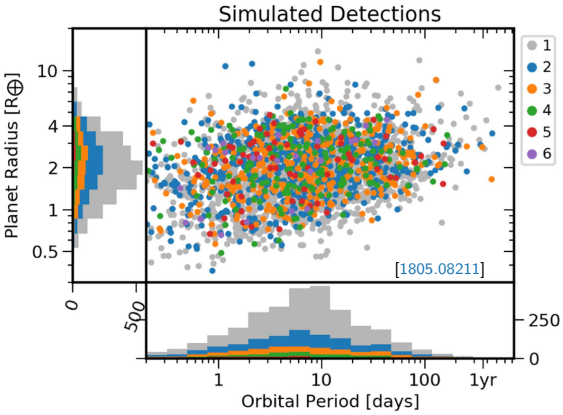
Goal: Characterize planet properties

- Radius, mass, period, orbital parameters
- Atmospheric composition

Model Comparison

Goal: Determine number of planets

- 1 planet vs 2 planets vs 3 planets...
- Avoid overfitting noise as signals



The Common Challenge: Complex Parameter Spaces

Why these problems are statistically hard

High-Dimensional

Many parameters to infer

- Cosmo: 6+ base parameters, nuisance/extensions
- GW: 15+ parameters (masses, spins, distance, sky location)
- EP: ∼5-10/planet

Challenge: Volume of parameter space grows exponentially

Degenerate

Parameters strongly correlated

- Cosmo: H_0-Ω_m banana
- GW: Mass-distance degeneracy
- EP: Eccentricity-period correlations

Challenge: Different parameter combinations produce similar observations

Multimodal

Multiple distinct solutions

- Cosmo: Data tensions create separated modes
- GW: Bimodal inclination
- EP: Orbital period aliases

Challenge: Easy to miss physically plausible explanations

The Language of Inference

How we quantify what we learn from data

Posterior $\mathcal{P}(\theta|D)$

What we know about the parameters *after* seeing the data. It's our updated state of knowledge.

Likelihood $\mathcal{L}(D|\theta)$

The probability of observing the data given the parameters. It connects our theoretical model to observations.

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

Prior $\pi(\theta)$

What we believe about the parameters *before* we see the data. Our physical assumptions.

Evidence $\mathcal{Z}(D)$

The total probability of the data given the model. It quantifies model plausibility and enables comparisons.

The Simplest Approach: Optimization (e.g., χ^2 Minimization)

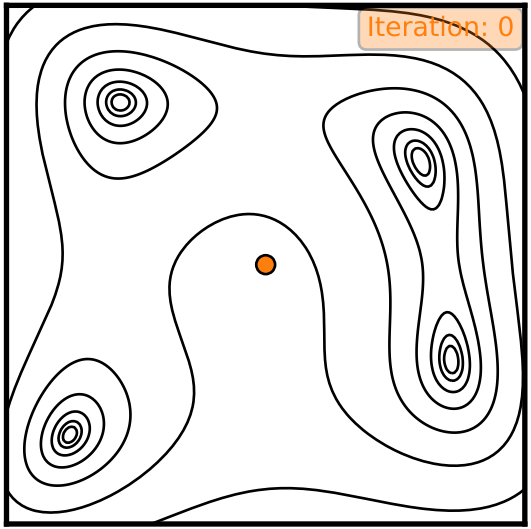
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

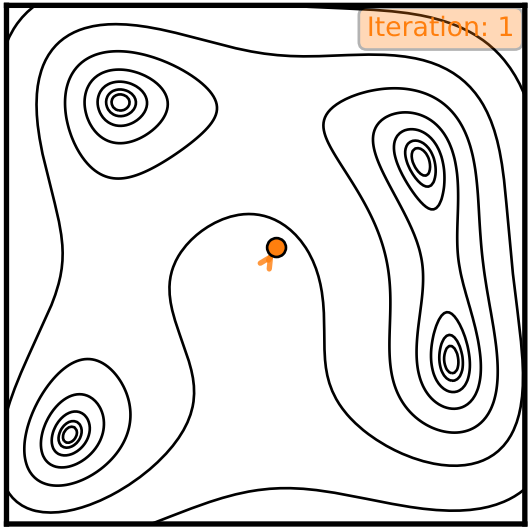
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

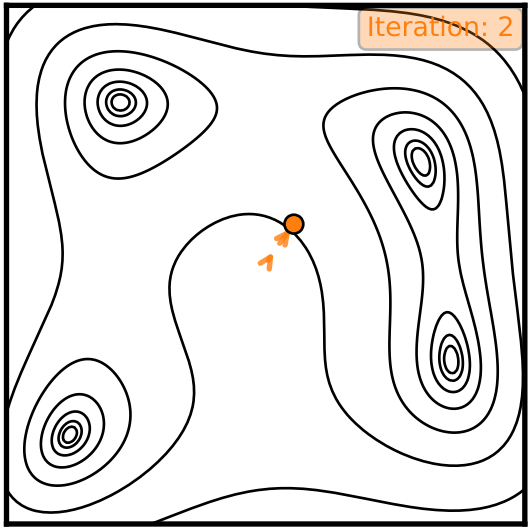
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

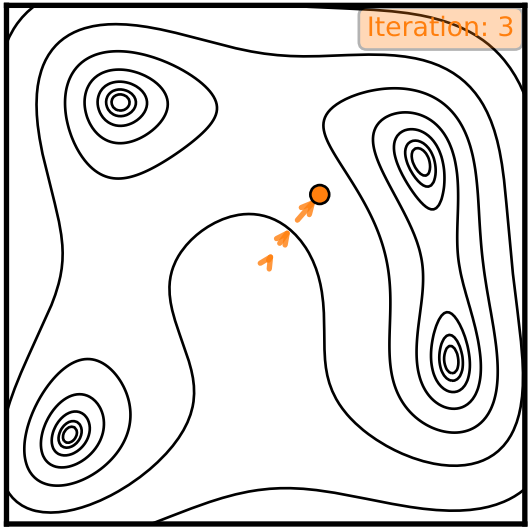
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

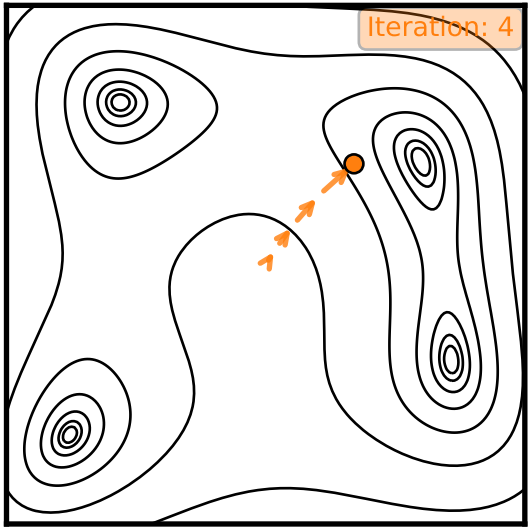
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

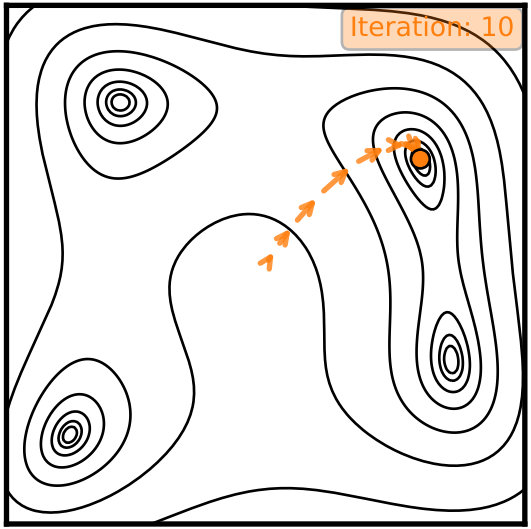
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



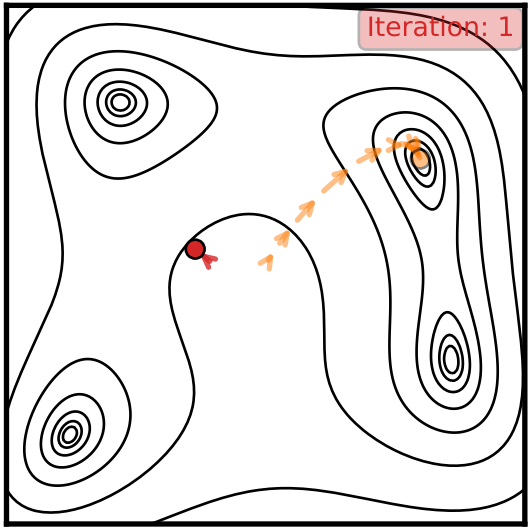
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



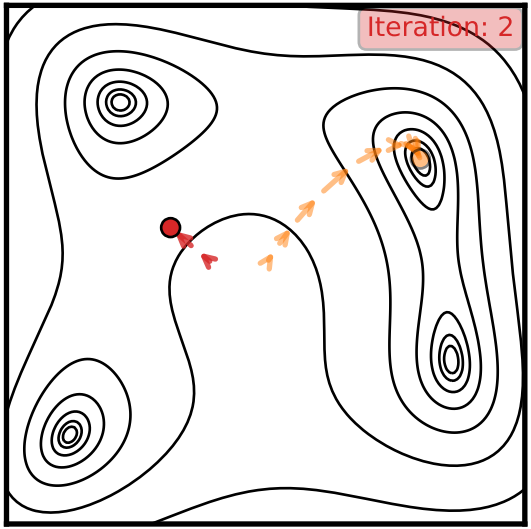
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



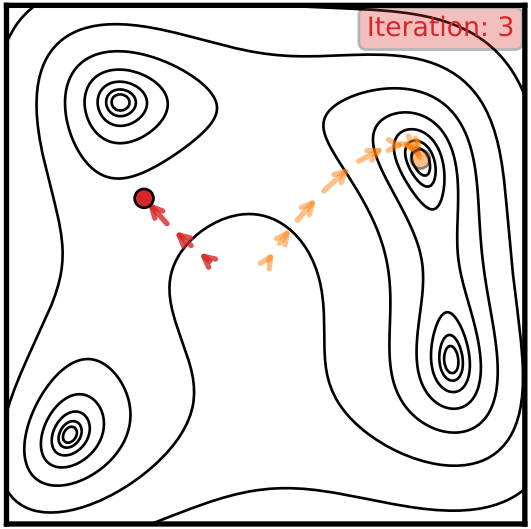
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



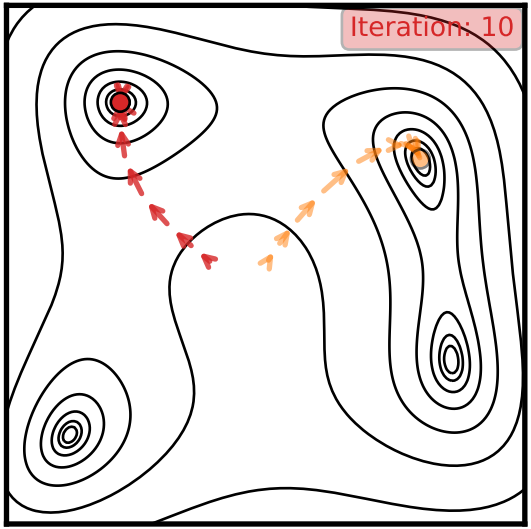
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.

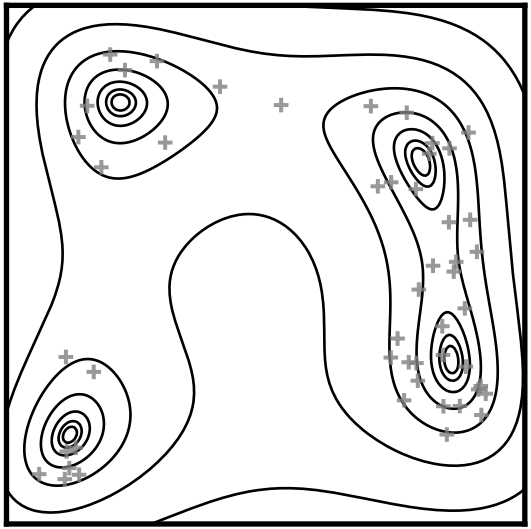


Why do sampling?

- The cornerstone of numerical Bayesian inference is working with **samples**.
- Generate a set of representative parameters drawn in proportion to the posterior $\theta \sim \mathcal{P}$.
- The magic of marginalisation $\Rightarrow$ perform usual analysis on each sample in turn.
- The golden rule is **stay in samples** until the last moment before computing summary statistics/triangle plots because

$$\boxed{f(\langle X \rangle) \neq \langle f(X) \rangle}$$

- Generally need $\sim \mathcal{O}(12)$ independent samples to compute a value and error bar.

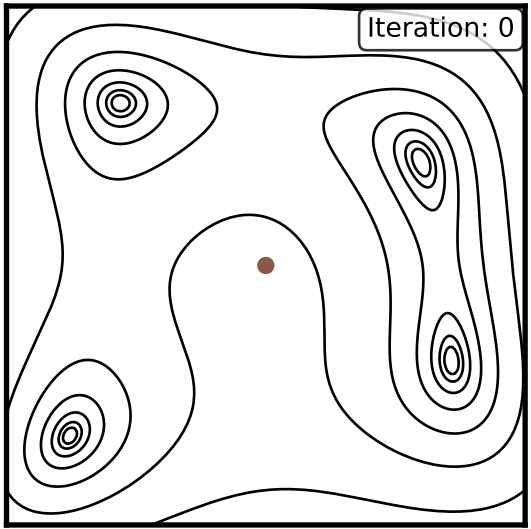


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

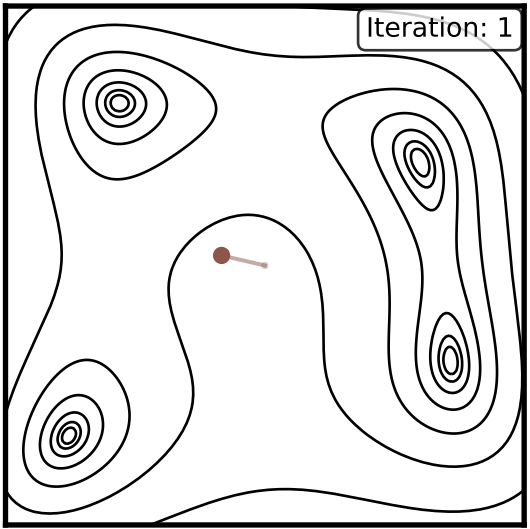


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

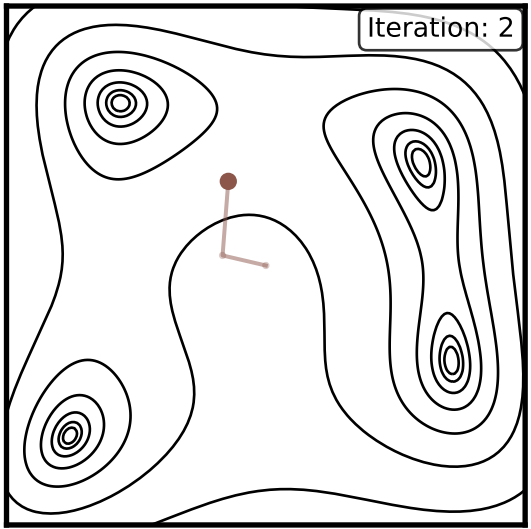


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

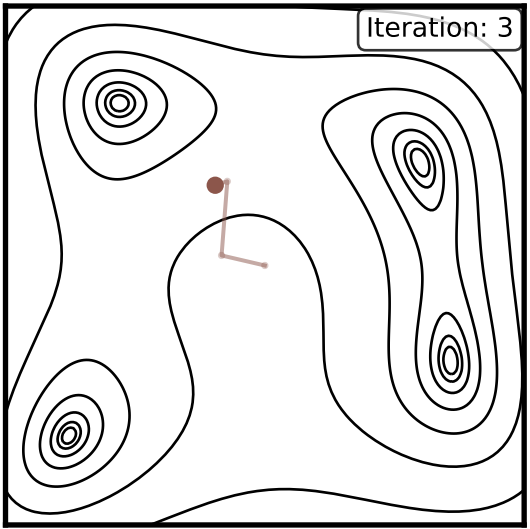


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

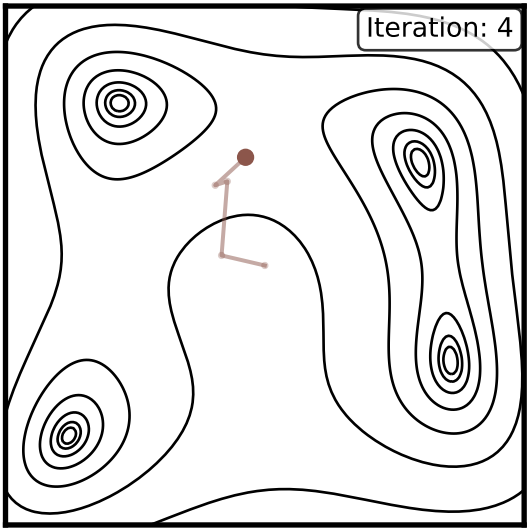


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.



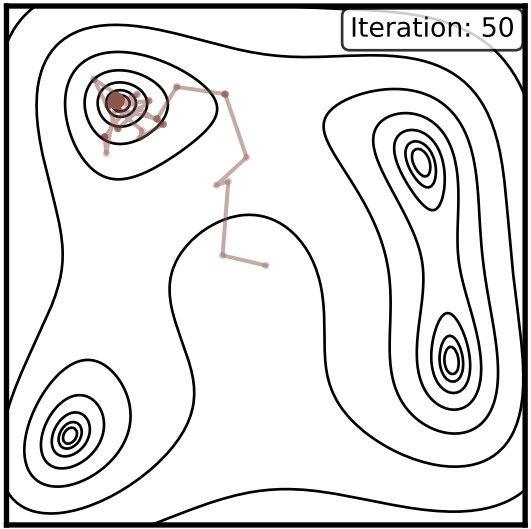
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of astronomical inference.



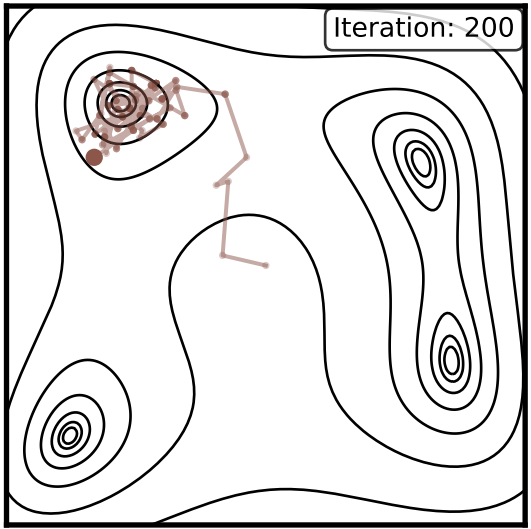
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of astronomical inference.



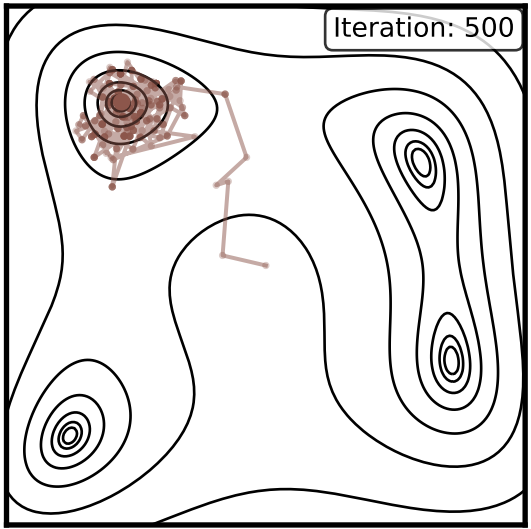
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of astronomical inference.



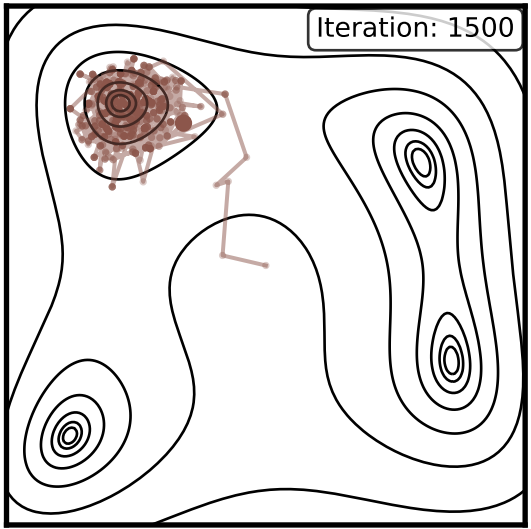
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of astronomical inference.

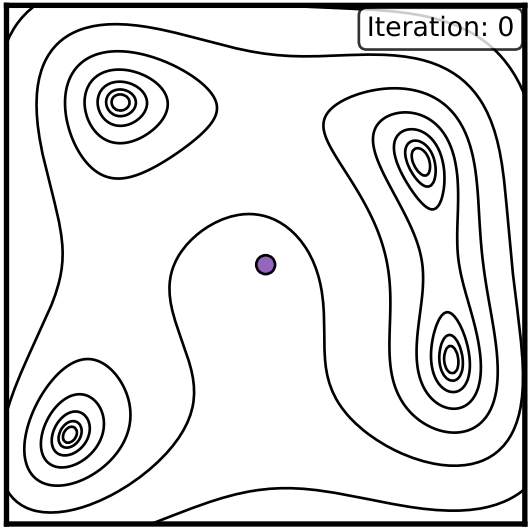


Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

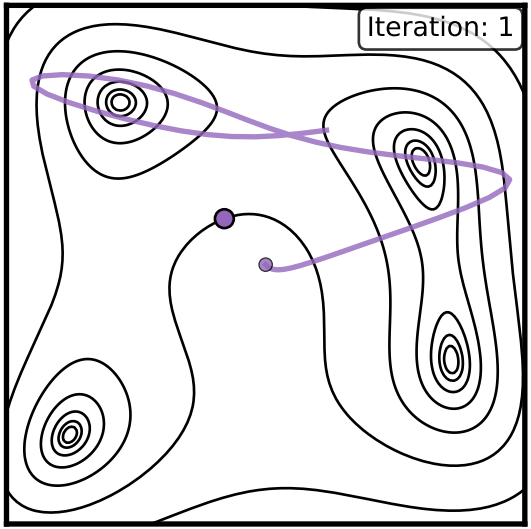


Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

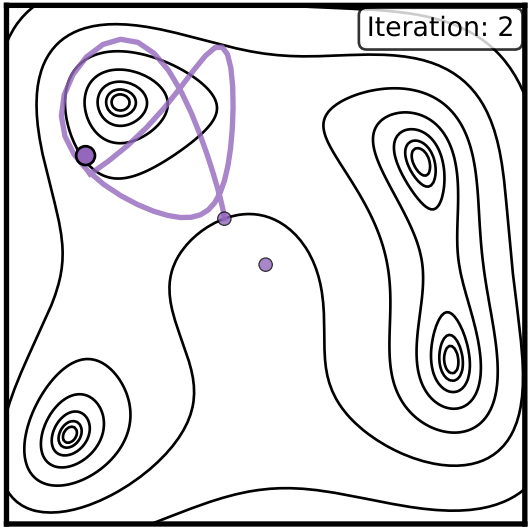


Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

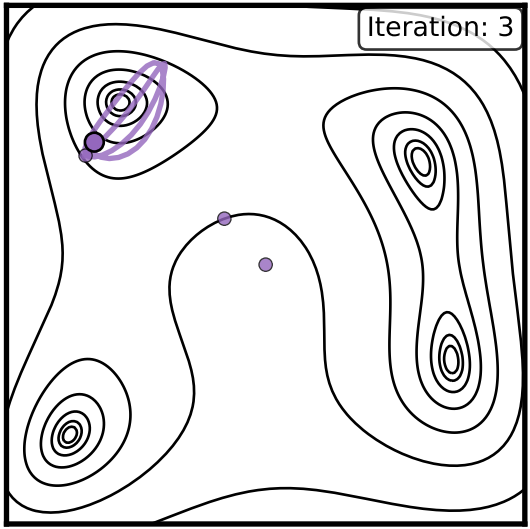


Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

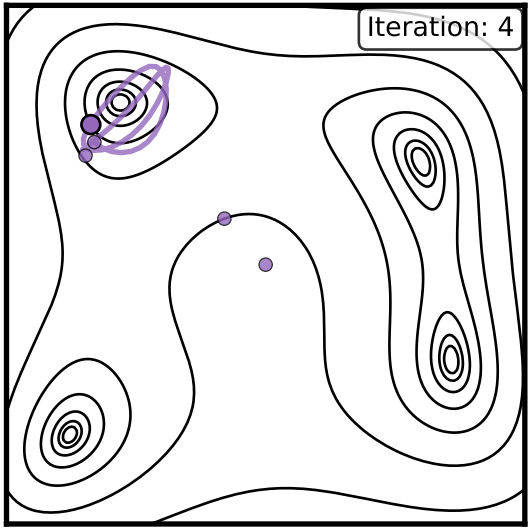


Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.



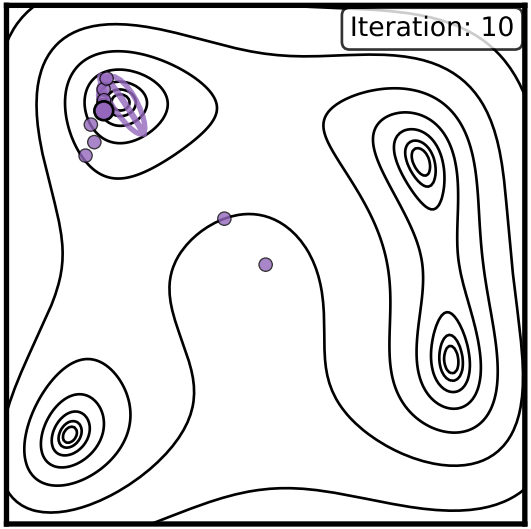
Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



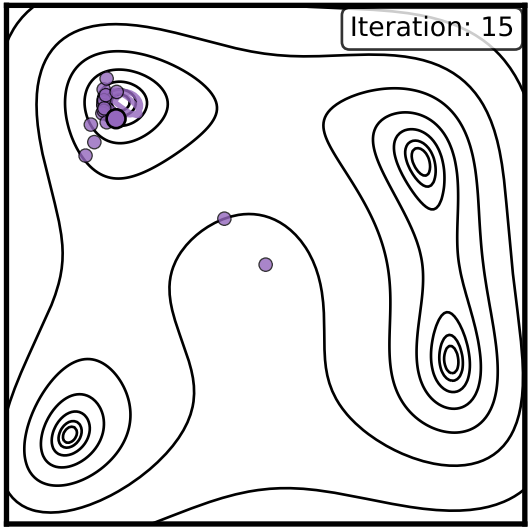
Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



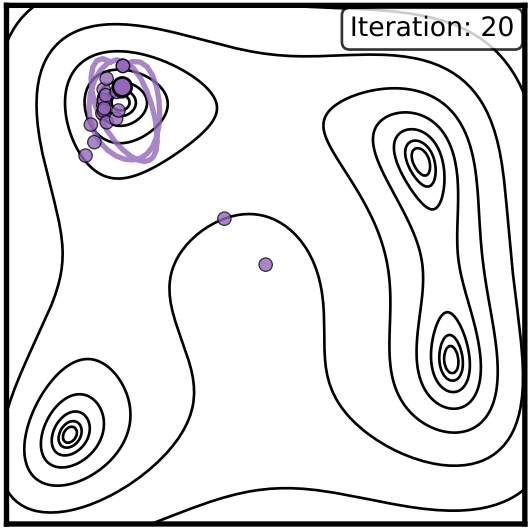
Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



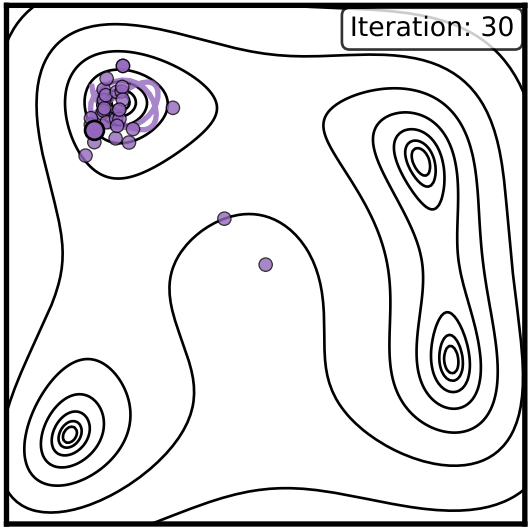
Gradient-Guided Sampling: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.

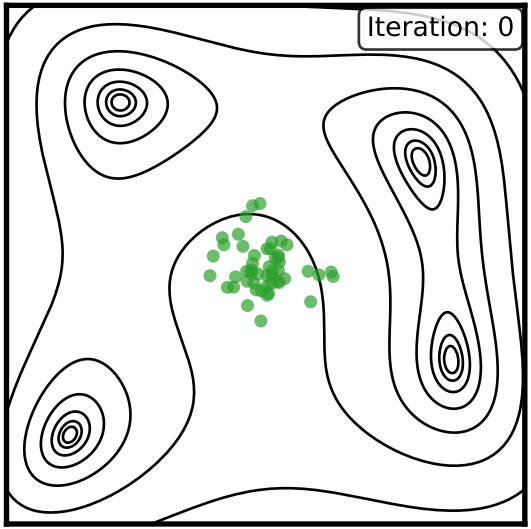


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

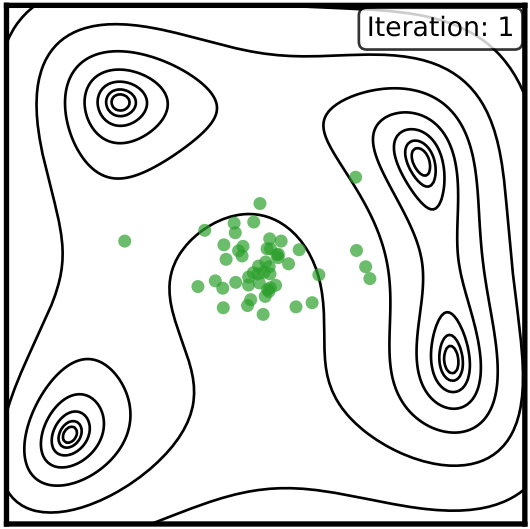


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

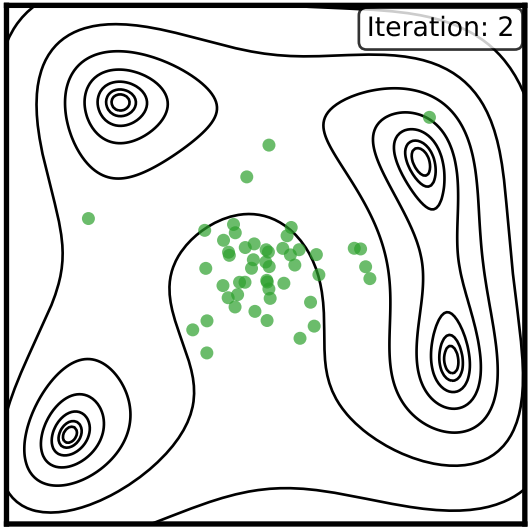


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

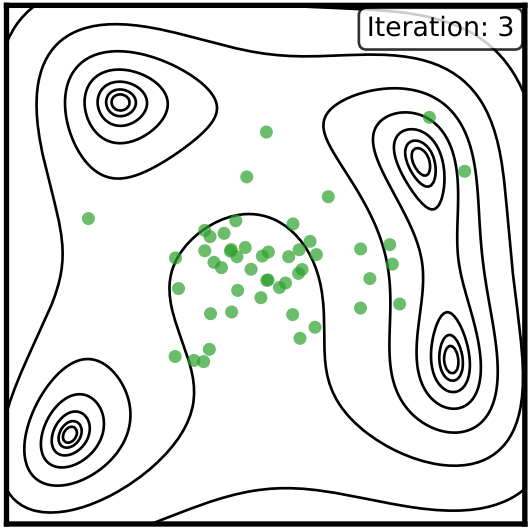


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

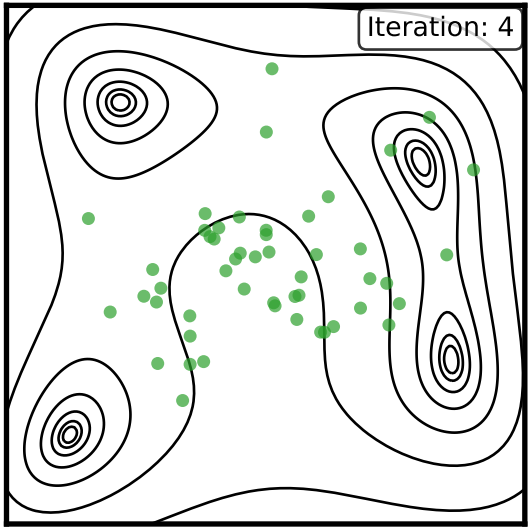


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.



A Better Way: Ensemble Sampling (e.g., `emcee`)

Advantages

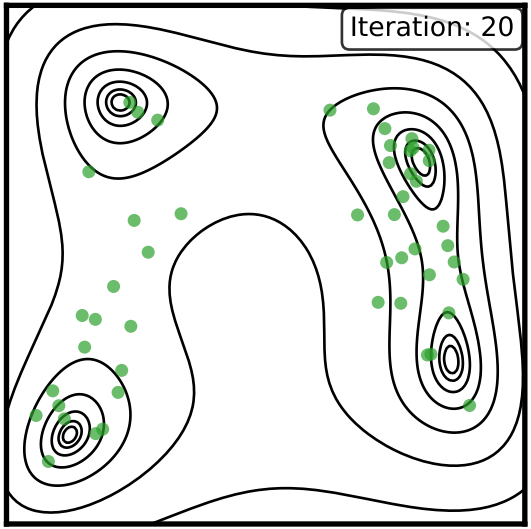
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like `emcee` are a major improvement for many problems, especially those with parameter degeneracies.



A Better Way: Ensemble Sampling (e.g., `emcee`)

Advantages

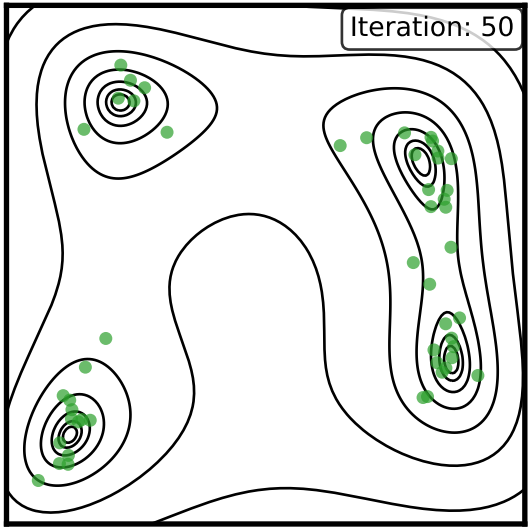
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like `emcee` are a major improvement for many problems, especially those with parameter degeneracies.



A Better Way: Ensemble Sampling (e.g., `emcee`)

Advantages

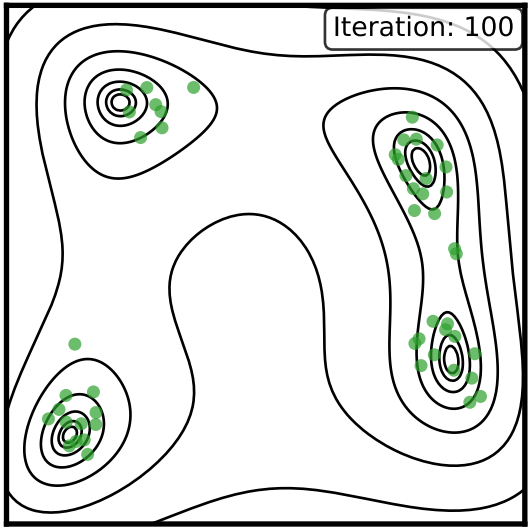
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like `emcee` are a major improvement for many problems, especially those with parameter degeneracies.



A Better Way: Ensemble Sampling (e.g., `emcee`)

Advantages

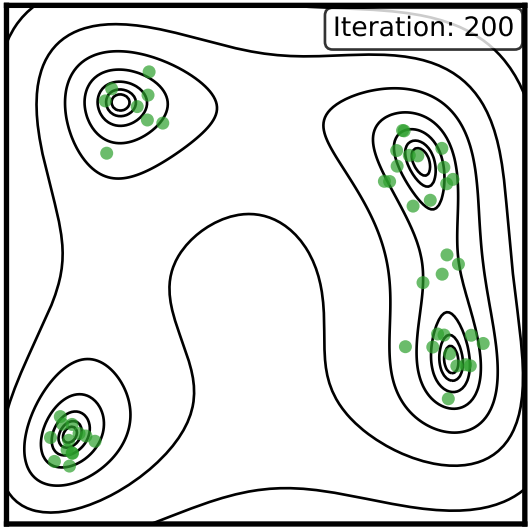
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like `emcee` are a major improvement for many problems, especially those with parameter degeneracies.



The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For astronomy: Which physical model best explains the observations?

- **Occam's Razor**: Automatic complexity penalty

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\mathsf{KL}}(\mathcal{P}||\pi)$$

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

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

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For astronomy: Which physical model best explains the observations?

- **Occam's Razor**: Automatic complexity penalty

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

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

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

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

- The difficulty is **not** that most of parameter space has $\mathcal{L} \approx 0$...

The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For astronomy: Which physical model best explains the observations?

- **Occam's Razor**: Automatic complexity penalty

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

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

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

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

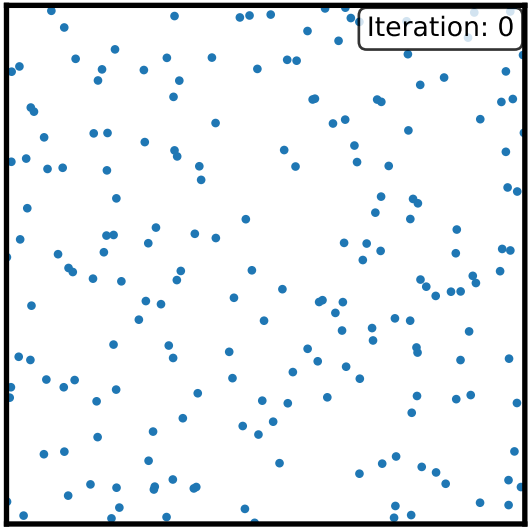
- The difficulty is **not** that most of parameter space has $\mathcal{L} \approx 0$...
- The difficulty is that we can't estimate **volume** $d\theta$ in high dimensions!

The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

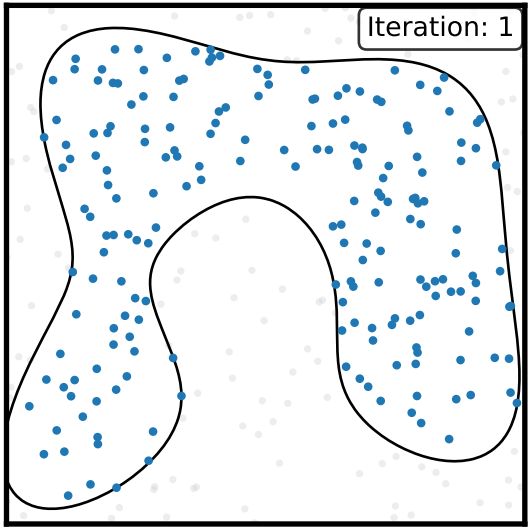


The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

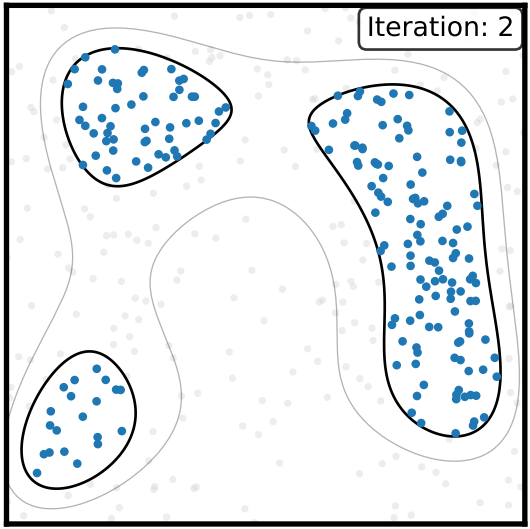


The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

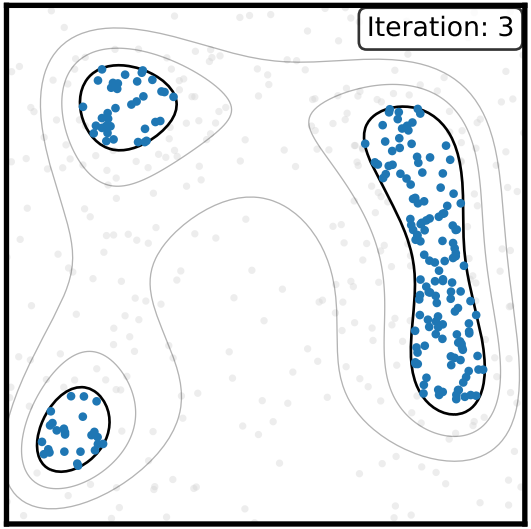


The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.



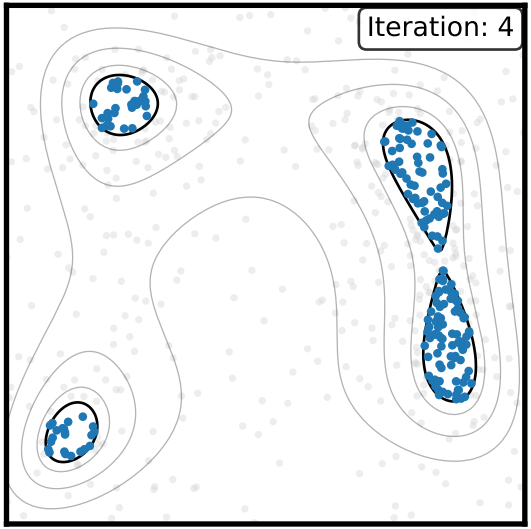
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



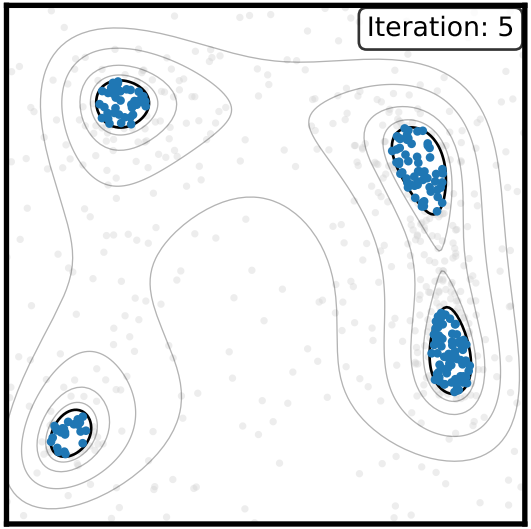
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



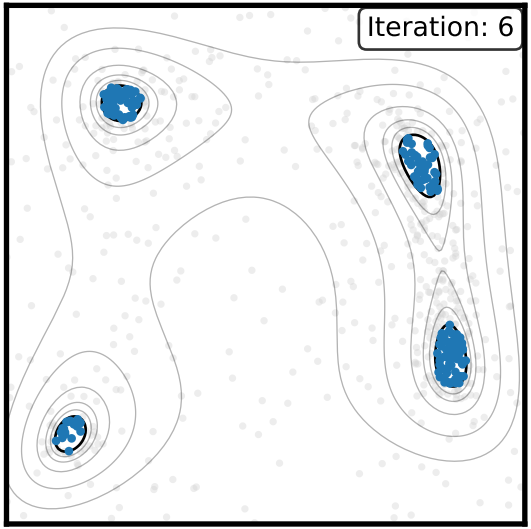
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



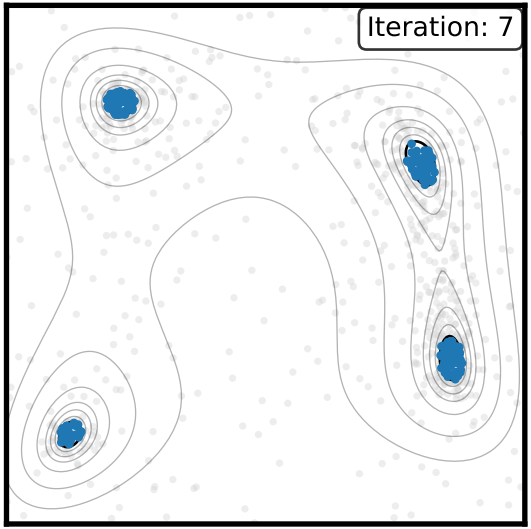
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

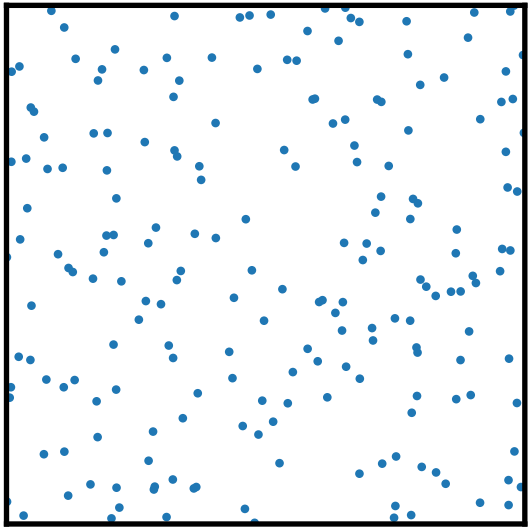
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

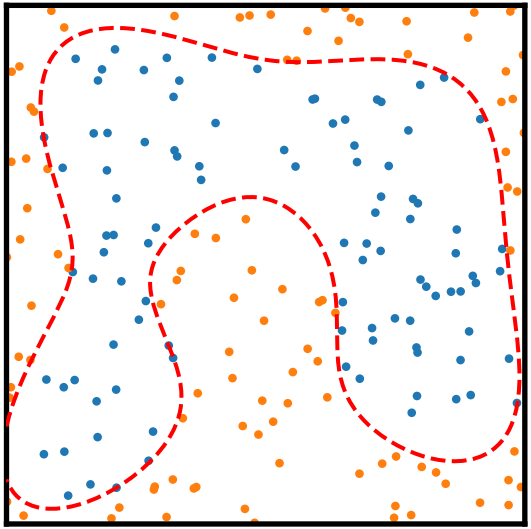
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

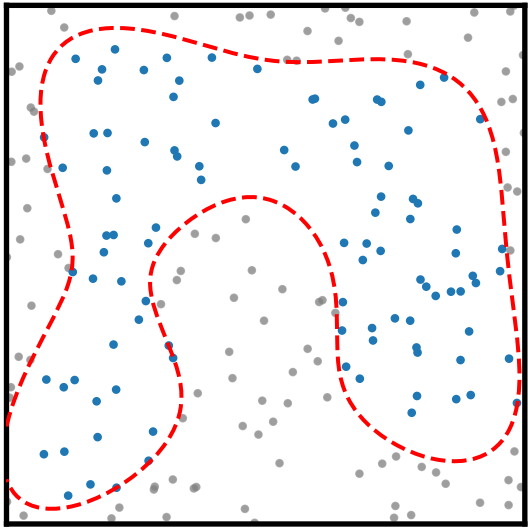
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

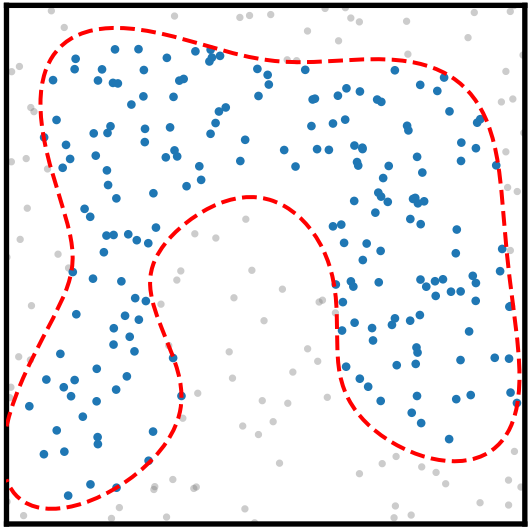
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

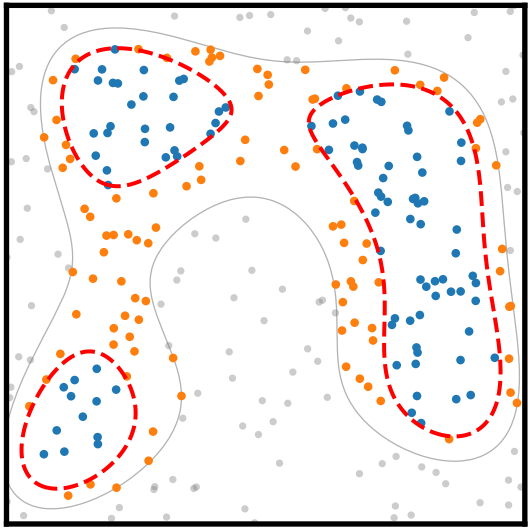
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

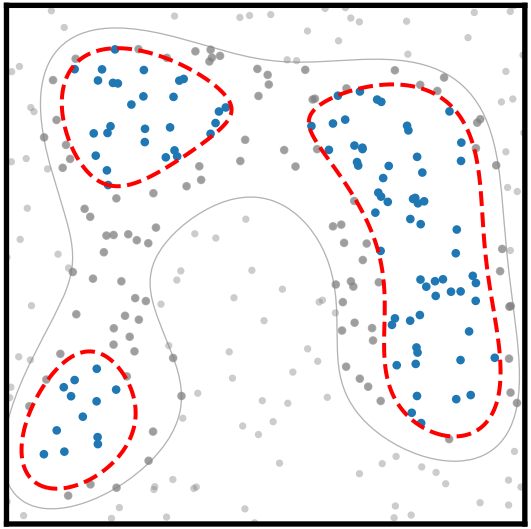
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

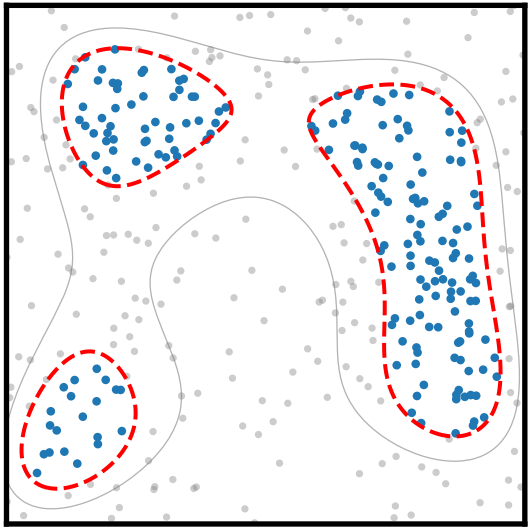
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

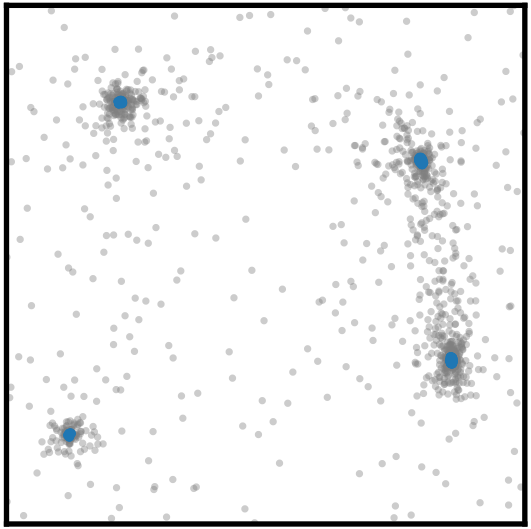
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

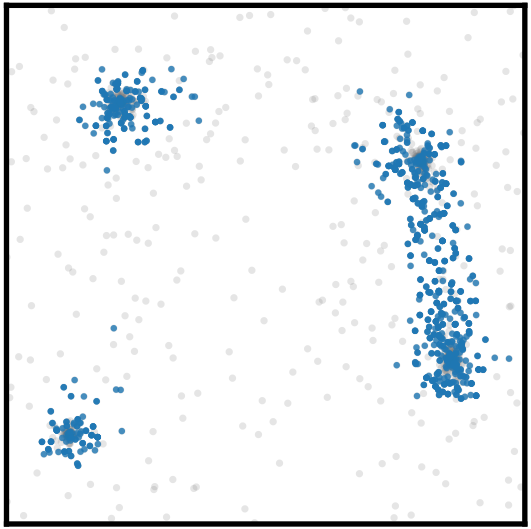
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



Practical Guidance: How to Use Nested Sampling

github.com/williamjameshandley/talks/tree/sydney_2024

Rejection Samplers

- e.g. `MultiNest`, `UltraNest`, `nessai`
- Construct bounding regions, reject invalid points
- Efficient in low dimensions ($d \lesssim 10$)
- Exponentially inefficient in high dimensions

Chain-based Samplers

- e.g. `PolyChord`, `dynesty`, `blackjax`
- Run Markov chains from live points
- Linear $\sim \mathcal{O}(d)$ scaling penalty
- Better for high-dimensional problems

Key Parameters

- **Resolution parameter** n_{live}: Improves results as $\sim \mathcal{O}(n_{\text{live}}^{-1/2})$
- **Reliability parameters**: Don't improve results if set arbitrarily high, but introduce systematic errors if set too low
  - `MultiNest` efficiency `eff`, `PolyChord` chain length `n_repeats`, `dynesty` `slices`

Choosing Your Tool: A Summary

No single best method, only the right tool for the job

Method	Speed	Uncertainties?	Multimodal?	Evidence?
Optimization (χ^2)	Very Fast	No	No	No
MCMC (`pymc` etc)	Medium	Yes	Poorly	No
HMC (`stan`, `numpyro`)	Medium	Yes	Okay	No*
Ensemble (`emcee`)	Medium	Yes	Okay	No
Nested (`dynesty`)	Slower	Yes	Excellently	Yes!

Practical Guidance

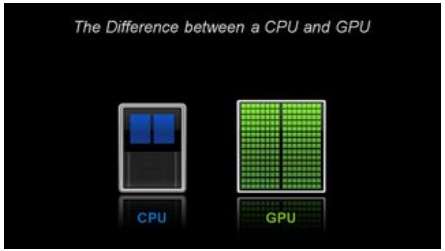
- **Quick exploration / Sanity check?** → Use Optimization.
- **Simple, well-behaved posterior?** → `emcee` is a great choice.
- **Smooth, differentiable posterior?** → HMC can be very efficient.
- **Complex, possibly multimodal posterior?** → Use `dynesty`.
- **Need to compare different physical models?** → You *must* use Nested Sampling.

The Solution: GPU Acceleration

Why the future of astronomical inference is parallel computing

The Problem

- Nested sampling is powerful but **slow**
- Complex parameter spaces require many likelihood evaluations
- Traditional CPU approaches struggle to scale



CPU vs GPU: Different Approaches

- **CPU**: Few powerful cores ($\sim$10s), sequential tasks
- **GPU**: Many simple cores ($\sim$1000s), parallel tasks

Perfect Match for Inference

- Nested sampling = many independent live points
- MCMC = parallel chains
- Likelihood evaluations naturally parallel

The Future is GPU

All new HPC facilities are GPU-based. Whether we like it or not, we must adapt our codes.

Modern Languages: Two Independent Capabilities

Differentiable programming languages: JAX, PyTorch, TensorFlow, Julia, Stan, ...

Capability 1: Free Gradients

- **Automatic differentiation**: $\nabla_\theta \log \mathcal{L}(\theta)$.
- Enables gradient-based MCMC (HMC, NUTS).
- Essential for modern optimization.

Traditional Physics Benefits

- **Nested sampling**: Massive parallelization.
- **21cm signals**: Vectorized across frequency/time/angle.
- **N-body sims**: GPU acceleration.

Capability 2: GPU Parallelization

- **Vectorization across ensembles**.
- Run 1000s of parallel chains/particles.
- Evaluate likelihoods simultaneously.

Key Insight: Often Confused

These are completely independent.
People mistake one for the other.
You can use gradients on CPU.
You can GPU parallelize without gradients.
They serve different purposes.

BlackJAX: GPU Native Sampling

A unified framework for GPU-accelerated inference

David Yallup

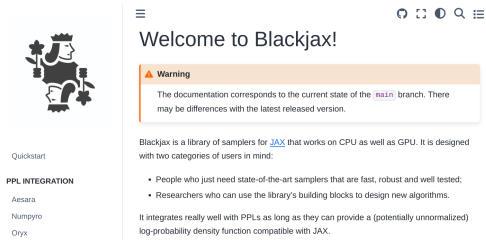
Postdoc

The Challenge

- Sampling traditionally CPU-bound
- Different algorithms, same GPU challenge
- Need unified GPU-native framework

BlackJAX Solution

- Full JAX ecosystem integration
- All core algorithms GPU-accelerated:
  - Optimization (gradient descent)
  - MCMC (Metropolis-Hastings)
  - HMC/NUTS (gradient-guided)
  - SMC (particle methods)
  - Ensemble (emcee) [PR #797]
  - Nested sampling [PR #755]



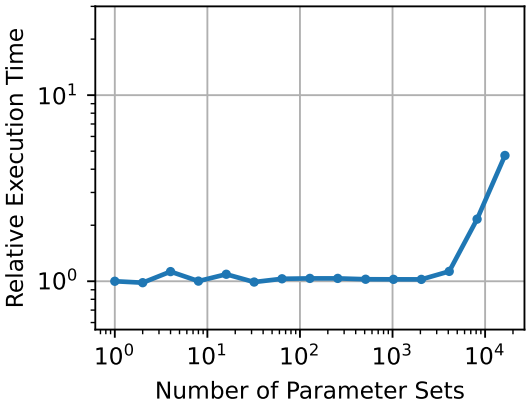
Framework Design

- Like `numpy/scipy`
- Not like `cobaya/cosmosis`
- Composable building blocks
- Maximum flexibility

Recent GPU-Accelerated Applications

Case study 1/4: CMB and Cosmic Shear [2509.13307]

- **CMB (6 params)**: 300× speedup vs CPU PolyChord
- **Cosmic Shear (37 params)**: Days vs months (>1000× vs CPU; 10× vs GPU NUTS)
- **Method**: JAX neural emulators + GPU NS
- **Evidence**: Direct calculation with error bars
- **Models**: ΛCDM vs $w_0 w_a$ comparison
- **Impact**: NS competitive with MCMC+evidence methods

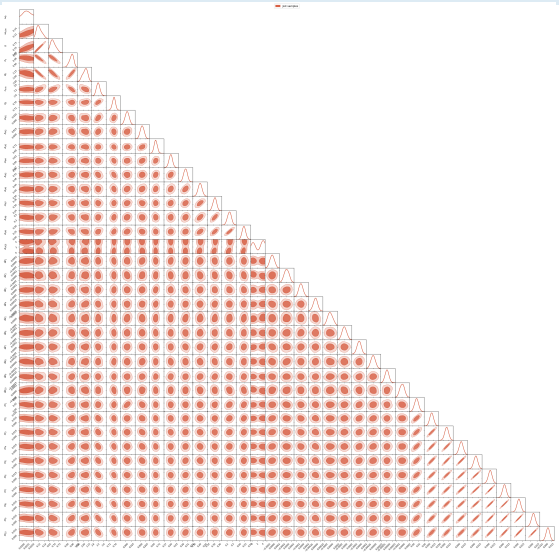


Recent GPU-Accelerated Applications

Case study 1/4: CMB and Cosmic Shear [2509.13307]

Toby Lovick
PhD

- **CMB (6 params)**: 300× speedup vs CPU PolyChord
- **Cosmic Shear (37 params)**: Days vs months (>1000× vs CPU; 10× vs GPU NUTS)
- **Method**: JAX neural emulators + GPU NS
- **Evidence**: Direct calculation with error bars
- **Models**: ΛCDM vs $w_0 w_a$ comparison
- **Impact**: NS competitive with MCMC+evidence methods

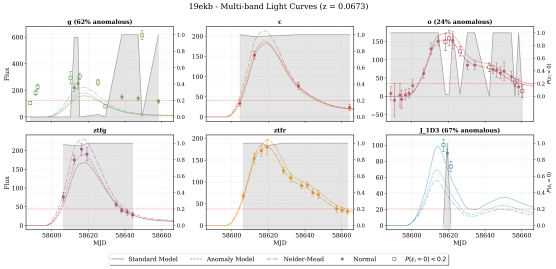


Recent GPU-Accelerated Applications

Case study 2/4: Bayesian Anomaly Detection for Type Ia Supernovae [2509.13394]

Sam Leeney
PhD

- **Problem**: Manual photometric rejection not scalable for LSST
- **Solution**: Bayesian anomaly detection integrated into SALT3 fitting
- **Method**: Model contamination probability per measurement
- **Result**: Automatic outlier/corrupted band rejection
- **Finding**: Contaminants systematically brighter/bluer
- **Impact**: Essential for unbiased cosmology at scale

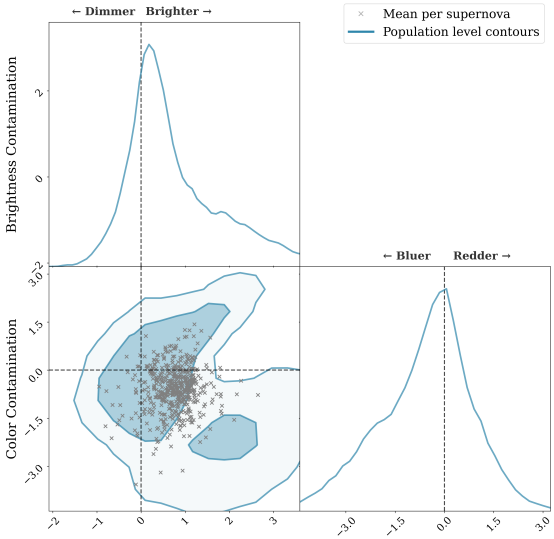


Recent GPU-Accelerated Applications

Case study 2/4: Bayesian Anomaly Detection for Type Ia Supernovae [2509.13394]

Sam Leeney
PhD

- **Problem**: Manual photometric rejection not scalable for LSST
- **Solution**: Bayesian anomaly detection integrated into SALT3 fitting
- **Method**: Model contamination probability per measurement
- **Result**: Automatic outlier/corrupted band rejection
- **Finding**: Contaminants systematically brighter/bluer
- **Impact**: Essential for unbiased cosmology at scale

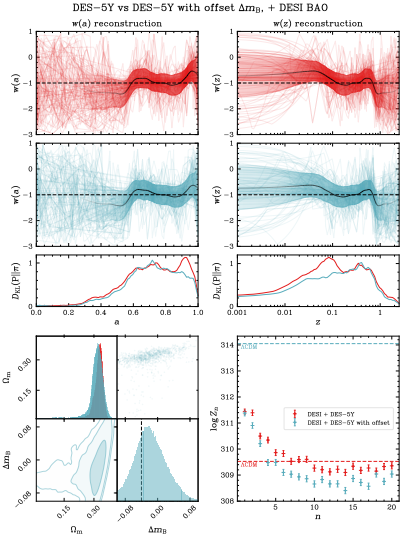


Recent GPU-Accelerated Applications

Case study 3/4: Dark Energy vs Supernova Systematics [2509.13220]

Adam Ormondroyd
PhD

- **Question**: DESI+DES $w_0 w_a$ preference - new physics or systematics?
- **Method**: Bayesian model comparison
- **Models**: Dynamic DE vs redshift-dependent SN bias
- **Result**: Systematics fit equally well with lower complexity
- **Evidence**: Favors systematic explanation
- **Lesson**: Test mundane before claiming exotic

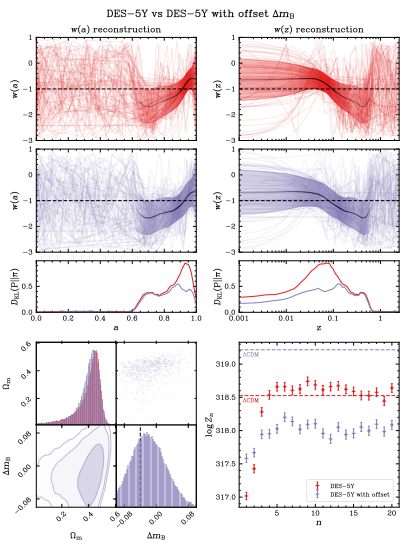


Recent GPU-Accelerated Applications

Case study 3/4: Dark Energy vs Supernova Systematics [2509.13220]

Adam Ormondroyd
PhD

- **Question**: DESI+DES $w_0 w_a$ preference - new physics or systematics?
- **Method**: Bayesian model comparison
- **Models**: Dynamic DE vs redshift-dependent SN bias
- **Result**: Systematics fit equally well with lower complexity
- **Evidence**: Favors systematic explanation
- **Lesson**: Test mundane before claiming exotic

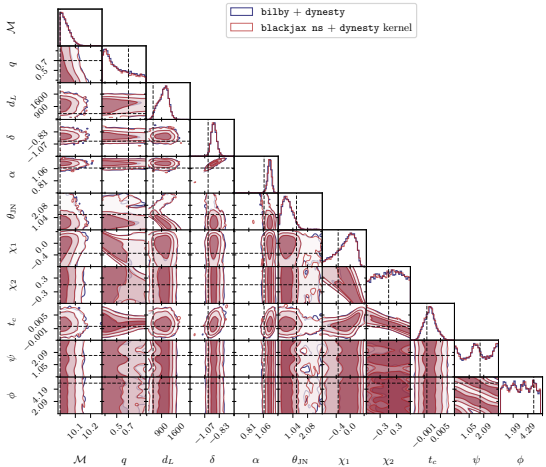


Recent GPU-Accelerated Applications

Case study 4/4: Gravitational Wave Inference [2509.04336]

Metha Prathaban
PhD

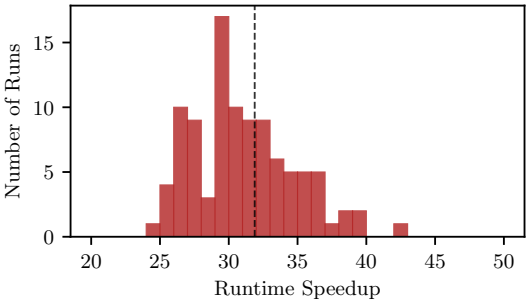
- **Goal**: GPU-accelerate bilby's acceptance-walk NS
- **Implementation**: Faithful port to blackjax-ns
- **Performance**: 20-40× speedup for BBH
- **Validation**: Identical posteriors/evidences
- **Hardware**: Single GPU vs CPU clusters
- **Impact**: Clean baseline for future methods



Recent GPU-Accelerated Applications

Case study 4/4: Gravitational Wave Inference [2509.04336]

- **Goal**: GPU-accelerate bilby's acceptance-walk NS
- **Implementation**: Faithful port to blackjax-ns
- **Performance**: 20-40× speedup for BBH
- **Validation**: Identical posteriors/evidences
- **Hardware**: Single GPU vs CPU clusters
- **Impact**: Clean baseline for future methods



Getting Started with BlackJAX

Practical steps to GPU-accelerated inference

Installation

```
pip install git+https://github.com/
    handley-lab/blackjax
```

Resources

- **Nested Sampling Book**: handley-lab.co.uk/nested-sampling-book
- **Workshop & Tutorials**: [github:handley-lab/workshop]
- **BlackJAX NS PR**: [github:blackjax-devs/blackjax] #755

Minimal Example

```
import jax
import blackjax

# Define model
logL = lambda x: -0.5 * jnp.sum(x**2)
prior = lambda key: jax.random.normal(key, (5,))

# Initialize nested sampling
ns = blackjax.ns(logL, prior, n_live=500)

# Run on GPU
key = jax.random.PRNGKey(0)
state = ns.init(key)
results = ns.run(state, max_samples=5000)
```

Key Points

- Use `jax.jit` to compile likelihoods
- `vmap` for batch operations
- Keep data on device (avoid CPU/GPU transfers)

The Future: AI in Scientific Code Development

The Real AI Revolution: LLMs

The biggest impact of AI will not be in analyzing data, but in helping us write the code to do it.

- **Automated code translation**: LLMs can help port legacy Fortran/C++ models to modern, GPU-friendly & differentiable frameworks like JAX or PyTorch.

The 80/20 Rule of Scientific Work

- **80% "boring" tasks**: Writing code, debugging, drafting & reviewing papers, munging data, organising meetings...
- **20% "hard thinking"**: The actual scientific insight.

AI's biggest immediate impact is automating and accelerating the 80%, freeing up human time for the 20%.

Key Message

AI is not just a tool for analysis; it's about to fundamentally change how we develop, optimize, and deploy our science

AI-Assisted Code Development Case Studies

Modernizing scientific software with LLM assistance

Rapid Development

- **Sam Leeney**: JAX-bandflux [2504.08081] – first draft vibe-coded with Roo in 3 hours
- **Toby Lovick**: 100× speedup using Claude Code to optimize existing JAX code [2509.13307]
- **Will Handley**: This talk + emcee [PR #797] written with LLM help. Transformational capability using Gemini and GPT-5 for interactive debugging, optimization, porting between samplers (cosmosis/cobaya/montepython), writing grants, reviewing papers, and transcribing meetings.

The Real Revolution: Interactive Development

Not full automation, but enabling rapid prototyping, testing, and development for Metha Prathaban, Sinah Legner, David Yallup, Wei-Ning Deng.

Conclusions

github.com/handley-lab/group

1. **Statistical Foundations Matter**
   - From optimization to sampling to model comparison
   - Nested sampling uniquely computes evidence
2. **GPU $\neq$ Machine Learning: Two Independent Capabilities**
   - GPUs accelerate any parallel algorithm.
   - Automatic differentiation + massive parallelization.
   - Often confused, serve different purposes.
3. **Classical Algorithms on GPU Competitive with ML State of the Art**
   - Traditional physics methods + GPU = superior performance.
   - 300$\times$ speedup for CMB, 20-40$\times$ for gravitational waves
4. **AI Accelerates Development as well as Computation**
   - LLMs solve the GPU porting challenge at scale.
   - 10$\times$ development speedup enables widespread adoption.

Get Started with GPU-Accelerated Sampling

handley-lab.co.uk/nested-sampling-book