

GPU-native nested sampling in BlackJAX

For simulation-based inference at scale

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

29th May 2025



UNIVERSITY OF
CAMBRIDGE

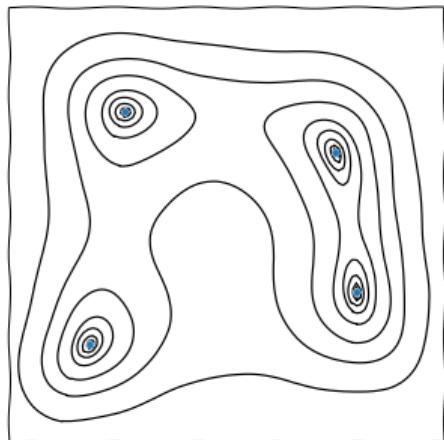


What is Nested Sampling?

- ▶ 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:

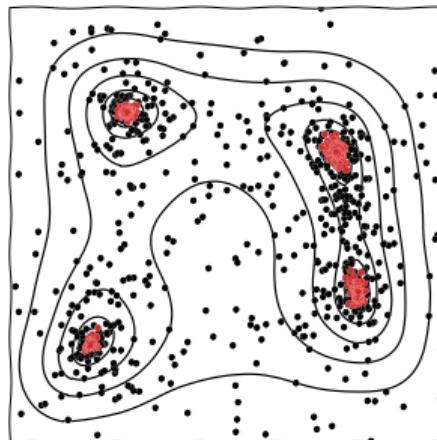
Optimisation

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



Exploration

draw/sample $\theta \sim f$



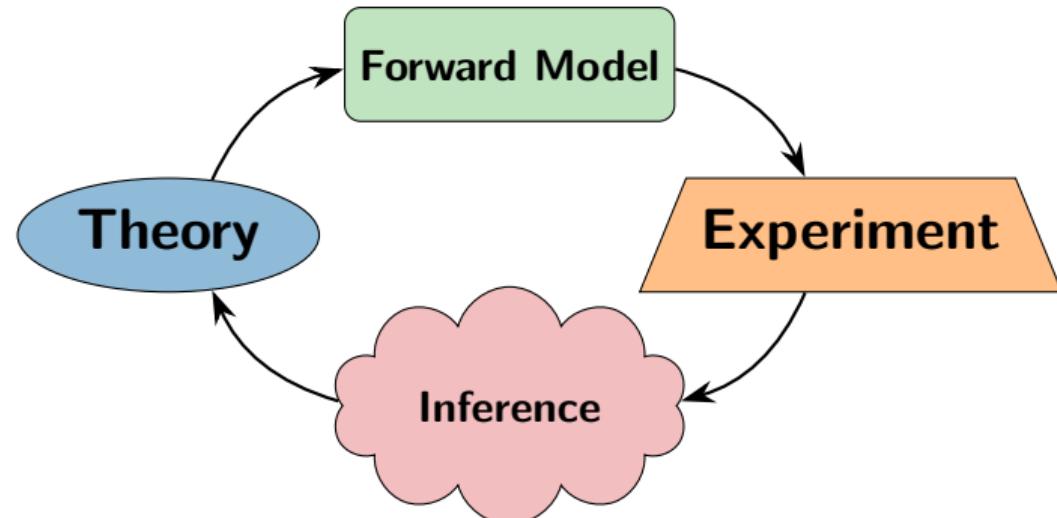
Integration

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



Why nested sampling for SBI?

- ▶ Other than NPE, all SBI methods (NLE, NRE, NJE etc) need a sampler
- ▶ Most existing samplers are limited:
 - ▶ **Legacy implementations:** MultiNest, PolyChord (Fortran)
 - ▶ **Slow Python codes:** dynesty, ultranest, nautilus
- ▶ BlackJAX solution:
 - ▶ GPU-native implementation
 - ▶ Open source & community-driven
 - ▶ Dissociated from original authors



The GPU imperative

- ▶ The future is GPU, whether we like it or not:
 - ▶ All future HPC heavily weighted toward GPUs
 - ▶ Driven by machine learning adoption
- ▶ JAX does two *separate* things:
 1. **Automatic differentiation**
 2. **Just-in-time compilation** for GPUs
- ▶ People often conflate these - they are separate and glorious!
- ▶ Alternative tools (harmonic, floz) get strength from GPU, not sampling method

GPU ecosystem growth

CMB cosmopower, candl

SNe BayesSN

GW ripple, jim

EP ExoJAX

[[github:JAXtronomy](https://github.com/JAXtronomy)]

BlackJAX nested sampling

David Yallup

PDRA



- ▶ Very recent development (December 2024!)
- ▶ GPU-native nested slice sampler
- ▶ Modern alternative to MultiNest/PolyChord
- ▶ Slice sampling with adaptive live point allocation
- ▶ Compatible with existing BlackJAX ecosystem

Learn more

- ▶ Theory: handley-lab.co.uk/nested-sampling-book
- ▶ Implementation: David Yallup's PDRA work
- ▶ Workshop: Hands-on tutorial today

Installation

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



AI-enabled research workflows

- ▶ Modern scientific computing benefits from AI integration:
 - ▶ GPU-accelerated inference
 - ▶ Neural networks in SBI methods
 - ▶ Automatic differentiation for sampling
- ▶ But the real coming disruption is from using AI to do day-to-day scientific tasks.
 - ▶ I wrote this talk with AI assistance [claude code].
This morning.
 - ▶ I have written and been awarded several grants that were majority written by AI.
 - ▶ My most recent papers are drafted with AI.

```
galaxy< 3:zsh 4:zsh 5:zsh 6:[tmux] 7:vim* 8:zsh- 9:zsh 10:perl will@maxwell 29 May 10:34
1 ❶Project Goal:** Develop a 1-hour short talk (approx. 10-15 minutes, -5 slides) and an accompanying hands-on workshop (approx. 45-50 minutes) on my new nested sampling algorithm in Bl
ackJAX.
1
2 Your Role: You are an AI assistant tasked with helping me conceptualize, outline, and dr
aft content for this talk and workshop. You should synthesize information from the provided
materials, propose structures, and help generate content. We will build a Git repository for
the workshop materials together.
3
4 Event Context: I
5 * Workshop I'm Attending: SBI Galev 2025 (Details: https://sbi-galev.github.io/2025/)
6 * My Session Slot: 1 hour total.
7
8 Key Input Materials & How to Use Them:
9
10 You will need to synthesize information from the following prompt-materials/ and external
links. Assume you can access and process the content within these resources based on my desc
riptions.
11
12 1. prompt-materials/intro_jax_sciml:
13   * Content: Workshop on JAX for scientific machine learning (by Viraj Pandya, Wedne
sday).
14   * Your Task: Refer to this directly for all relevant content on JAX and scientific
machine learning. Use it to understand what foundational JAX knowledge the audience might h
ave gained.
15
16 2. prompt-materials/ltu-il1:
17   * Content: Workshop on implicit likelihood inference in the "learning the universe
" framework (by Matt Ho, Tuesday).
18   * Your Task: Use this to understand the broader conference content and identify po
tential example problem domains that resonate with the audience.
19
20 3. prompt-materials/sbi-talk:
21   * Content: Introduction to simulation-based inference (talk by Maximilian von Wiet
ersheim-Kramsta, Tuesday), including LaTeX and figures.
22   * Your Task: Refer to this for material already covered on SBI, ensuring my talk b
uilds upon it without excessive repetition.
23
24 4. prompt-materials/talks/ (Git Repository - My Previous Talks):
25   * Content: A Git repository of my past talks. The README.md lists available talk
branches.
26   * Your Task: I need you to conceptually "explore" these branches (as if you were c
hecking them out and reading will_handley.tex for each) to identify relevant content for t
he new talk. The goal is to synthesize and adapt existing material onto a new branch bristo
1_2025.
27   * cosmoverse_2025: GPU machinery & LLMs.
28   * vietnam_2025: Most up-to-date views on SBI machinery.
29   * imperial_2024: Linear simulation-based inference.
```

1,1

Top

AI-enabled research workflows

- ▶ Modern scientific computing benefits from AI integration:
 - ▶ GPU-accelerated inference
 - ▶ Neural networks in SBI methods
 - ▶ Automatic differentiation for sampling
- ▶ But the real coming disruption is from using AI to do day-to-day scientific tasks.
 - ▶ I wrote this talk with AI assistance [claude code]. This morning.
 - ▶ I have written and been awarded several grants that were majority written by AI.
 - ▶ My most recent papers are drafted with AI.

```
galaxy:vim 3:zsh 4:zsh 5:zsh 6:[tmux]* 7:vim- 8:zsh 9:zsh 10:v>will@maxwell 29 May 10:37  
10:25:31 [2257/2300]  
> OK. Let's start with the talk. You should follow the instructions in the prompt.md and pre-prompt.md for going to the git repository in prompt-materials/talks, checking out branches and compiling into a new beamer talk of the same style on the bristol_2025 branch (which you should branch off from master for)  
  
I  
  
• Update Todos

- └  Navigate to talks repository and explore available branches
- └  Check out relevant branches to assess content for Bristol 2025 talk
- └  Create bristol_2025 branch from master
- └  Compile new beamer talk adapting relevant content

  
• Update Todos

- └  Navigate to talks repository and explore available branches
- └  Check out relevant branches to assess content for Bristol 2025 talk
- └  Create bristol_2025 branch from master
- └  Compile new beamer talk adapting relevant content

  
• Bash(cd /home/will/documents/workshop-blackjax-nested-sampling/prompt-materials/talks)...

- └ (No content)

  
• Bash(cd /home/will/documents/workshop-blackjax-nested-sampling/prompt-materials/talks && git branch -a)

- └ * master  
    remotes/origin/21cm_2020_workshop_cambridge  
    ... +66 lines (ctrl+r to expand)  
    remotes/origin/vietnam_2025  
    remotes/origin/zurich_2023

  
• Read(README.md)...

- └ Read 334 lines (ctrl+r to expand)

  
• Update Todos

- └  Navigate to talks repository and explore available branches
- └  Check out relevant branches to assess content for Bristol 2025 talk

```

Workshop goals

- ▶ Today we'll explore:
 1. Running nested sampling with BlackJAX
 2. Visualization with Anesthetic
 3. Performance comparison: nested sampling vs AIES
 4. Integration with your JAX workflows
- ▶ Hands-on notebook environment
- ▶ Google Colab compatible
- ▶ Build on Viraj's JAX/SciML workshop

Links

- ▶ BlackJAX: [[github:handley-lab/blackjax](https://github.com/handley-lab/blackjax)]
- ▶ Anesthetic: anesthetic.readthedocs.io
- ▶ Workshop materials: [[github:handley-lab](https://github.com/handley-lab)]

Performance promise

Compare BlackJAX nested sampling performance with traditional tools and see the GPU advantage firsthand!

Conclusions

[github:handley-lab]



- ▶ **Nested sampling is essential** for most SBI methods (except NPE)
- ▶ **BlackJAX** provides GPU-native, community-driven implementation
- ▶ **JAX's dual power:** autodiff + JIT compilation for unprecedented performance
- ▶ **The future is GPU-accelerated** scientific computing with AI integration

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

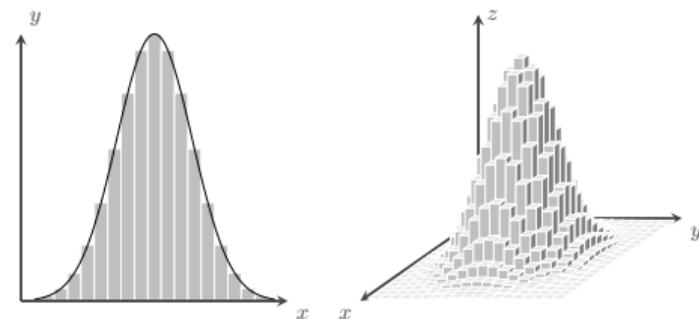
$$\Psi = \int e^{iS} \mathcal{D}x$$

Bayesian marginals

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

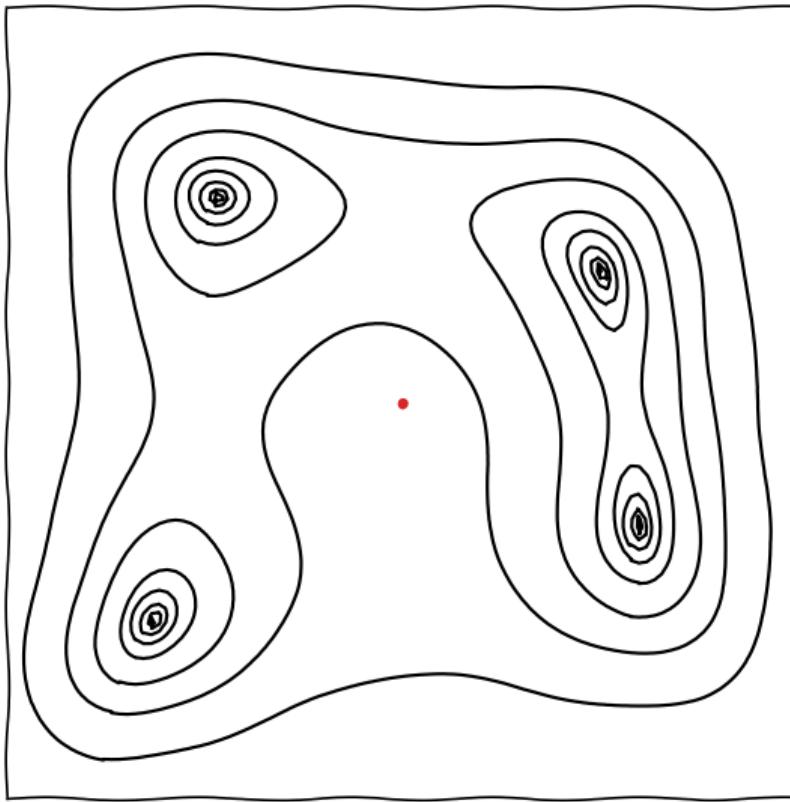
- ▶ Need numerical tools if analytic solution unavailable.
- ▶ High-dimensional numerical integration is hard.
- ▶ Riemannian strategy estimates volumes geometrically:

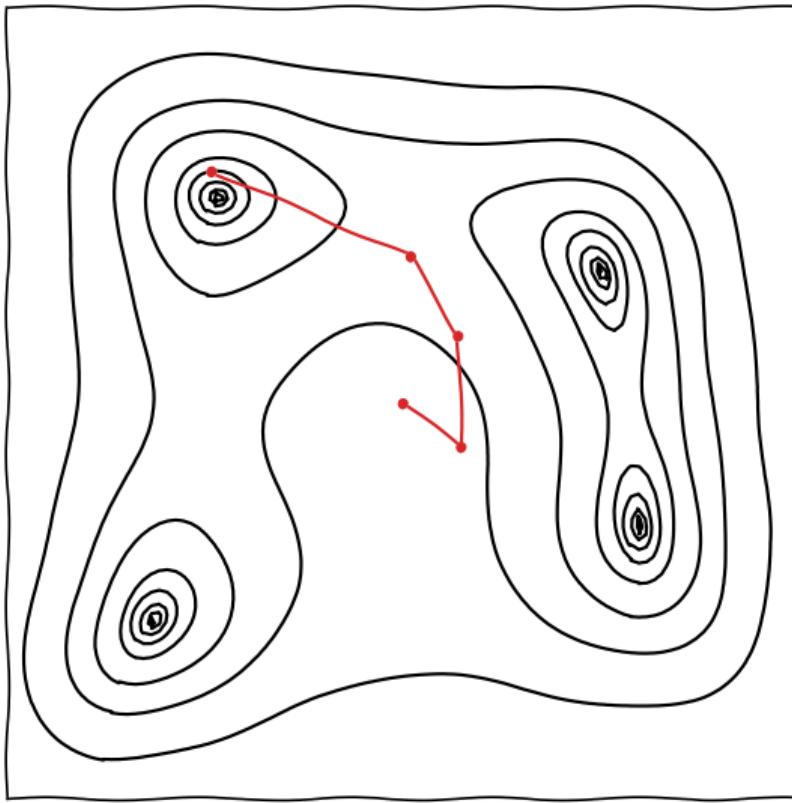
$$\int f(x) d^n x \approx \sum_i f(x_i) \Delta V_i \sim \mathcal{O}(e^n)$$



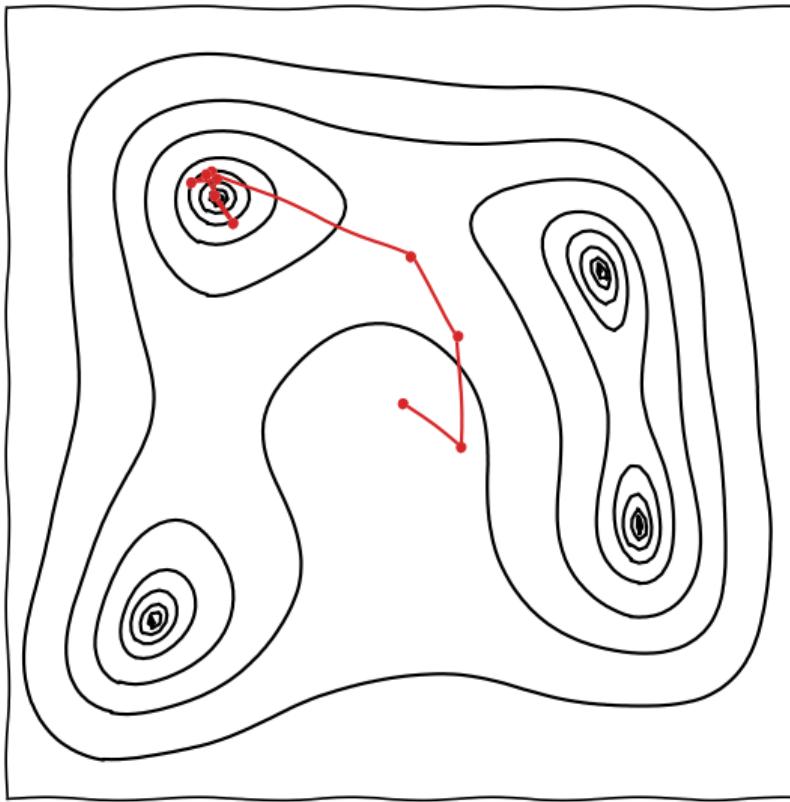
- ▶ Curse of dimensionality \Rightarrow exponential scaling.

MCMC

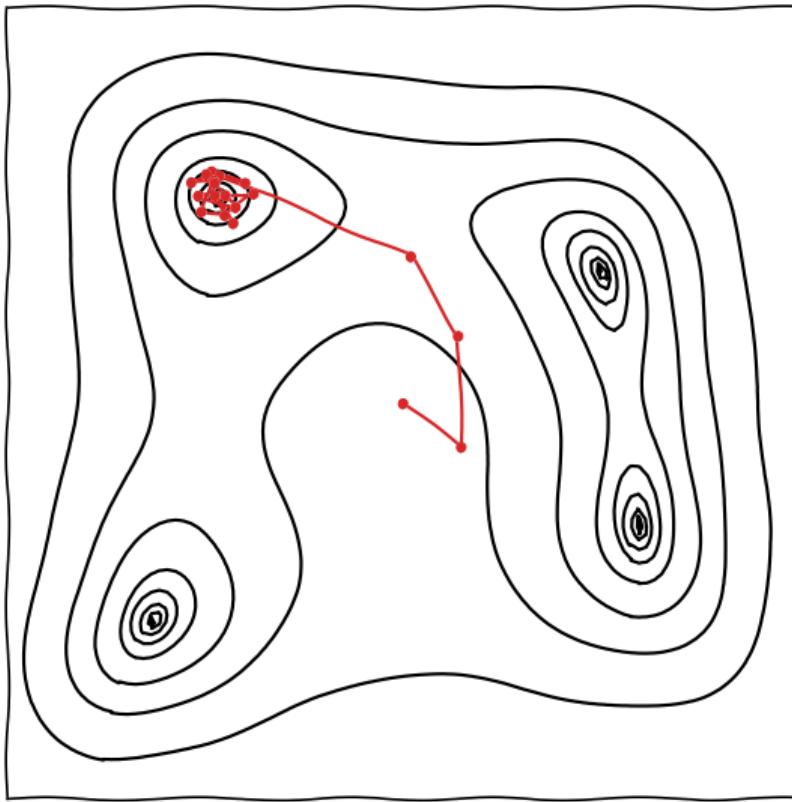




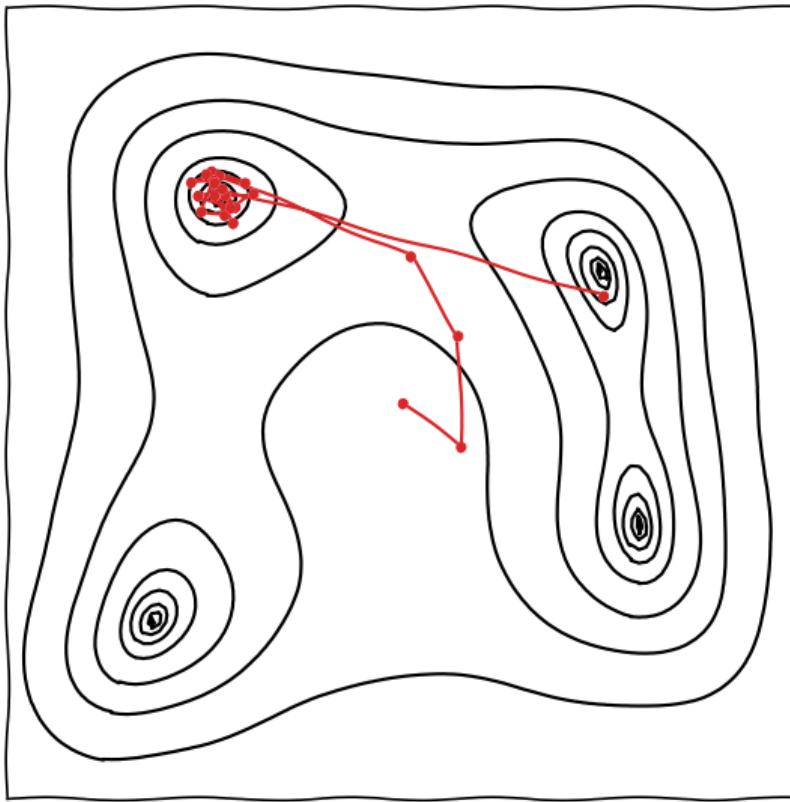
MCMC



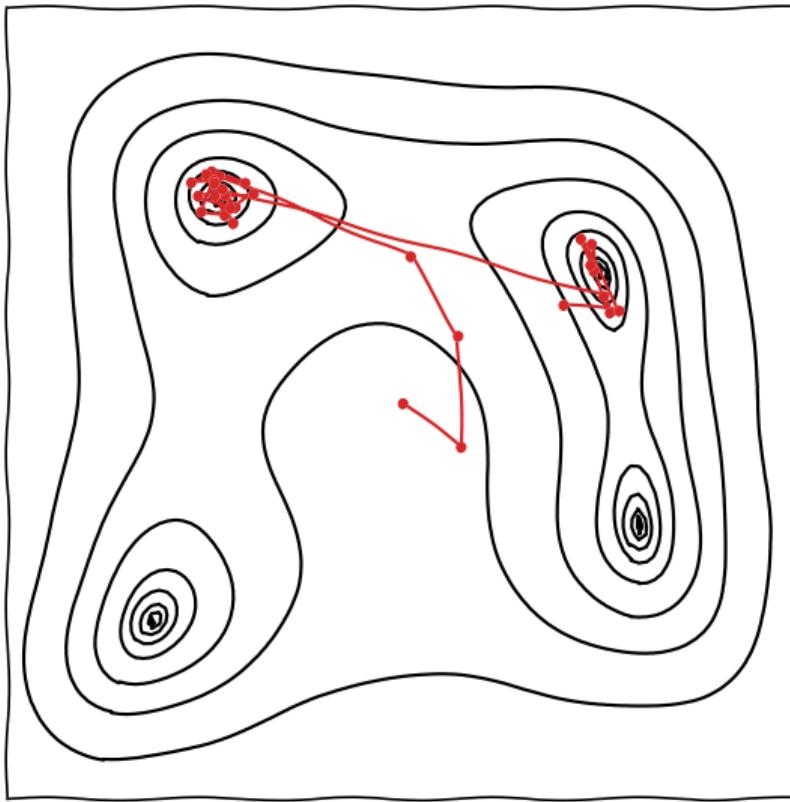
MCMC



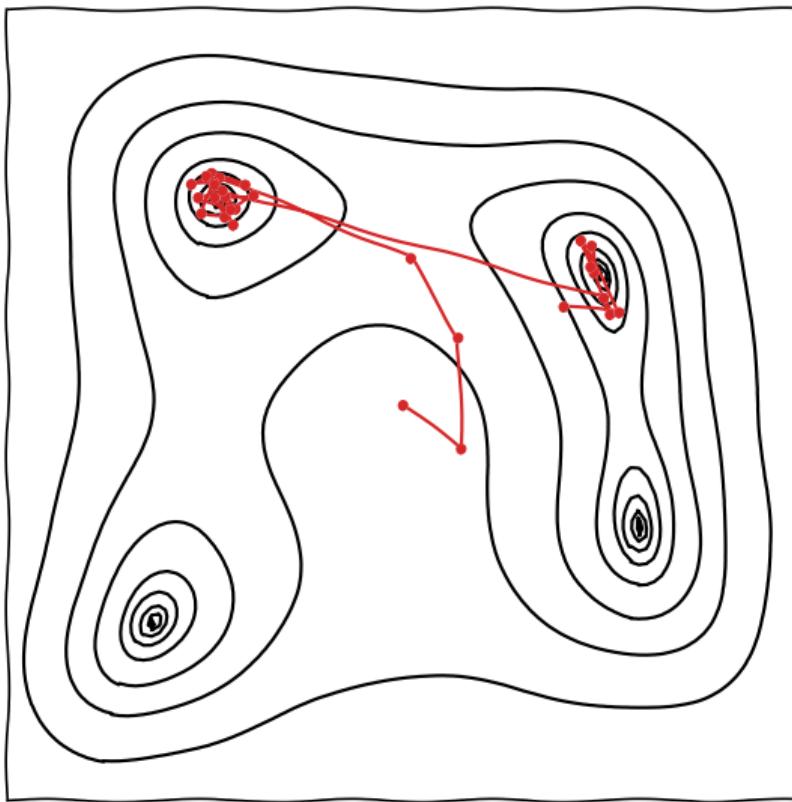
MCMC



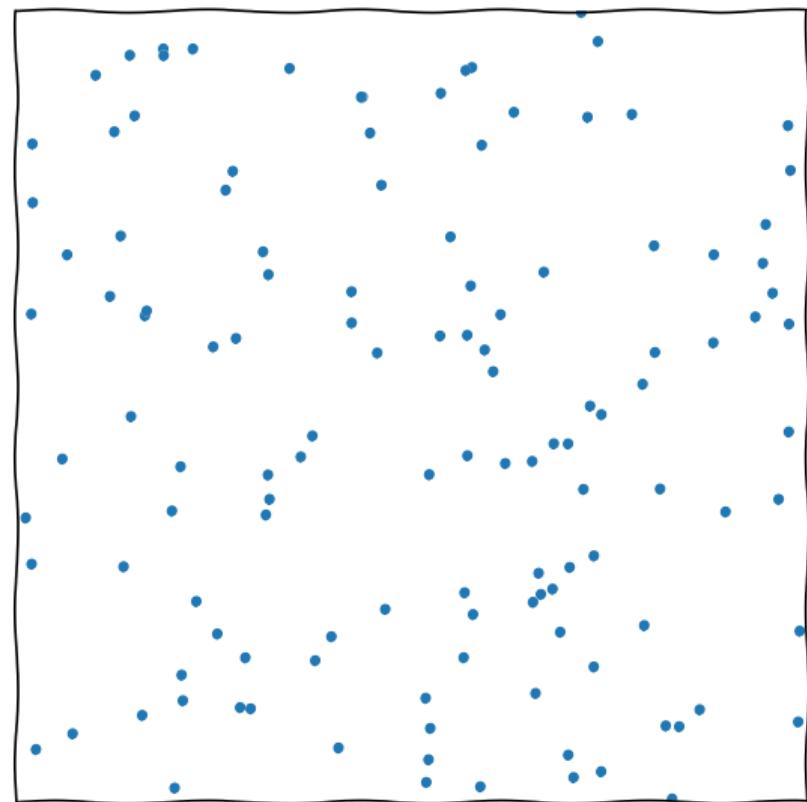
MCMC



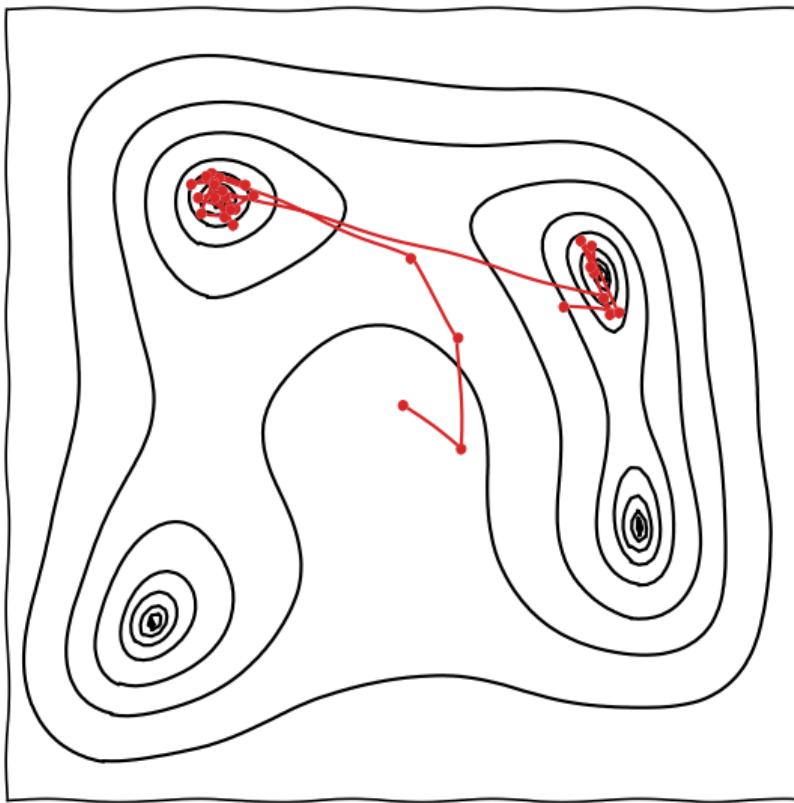
MCMC



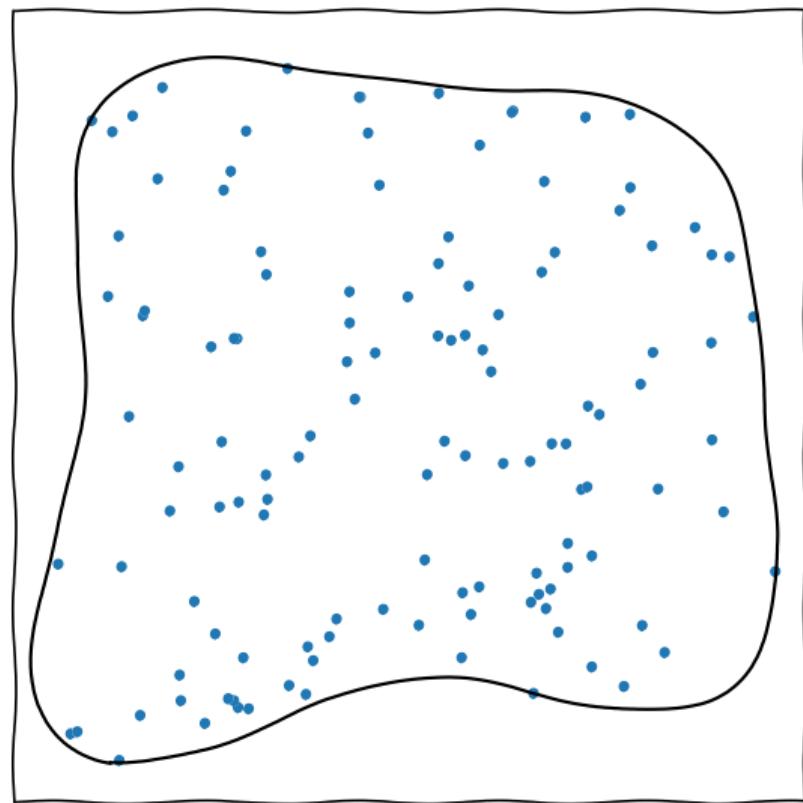
Nested sampling



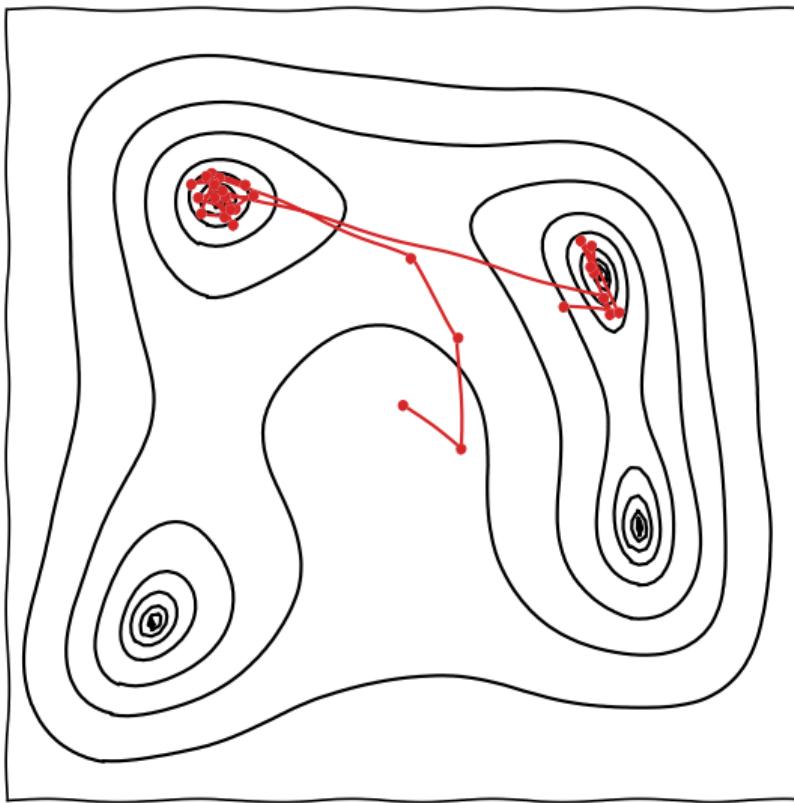
MCMC



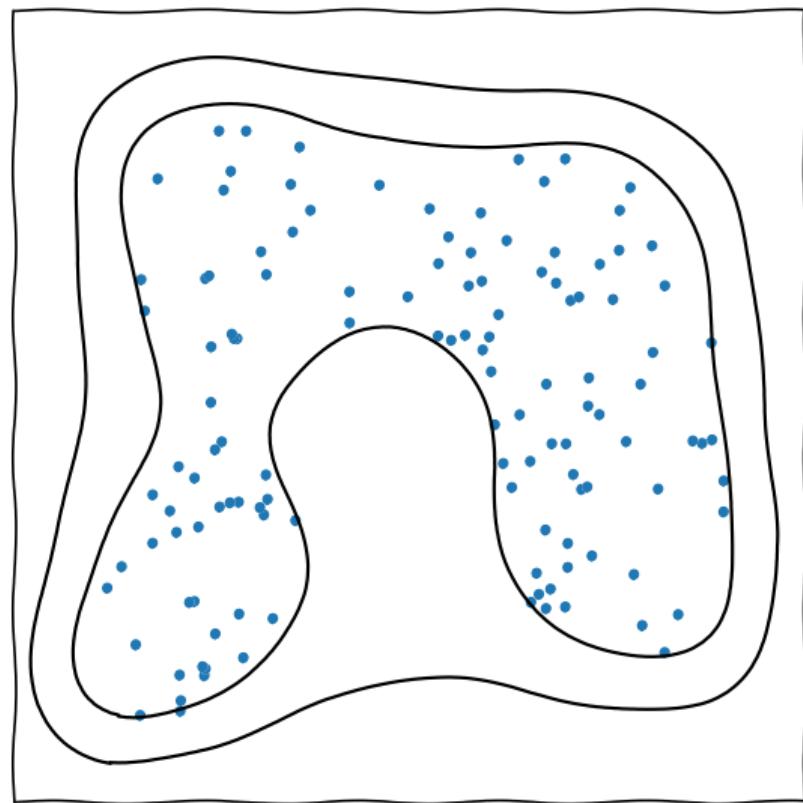
Nested sampling



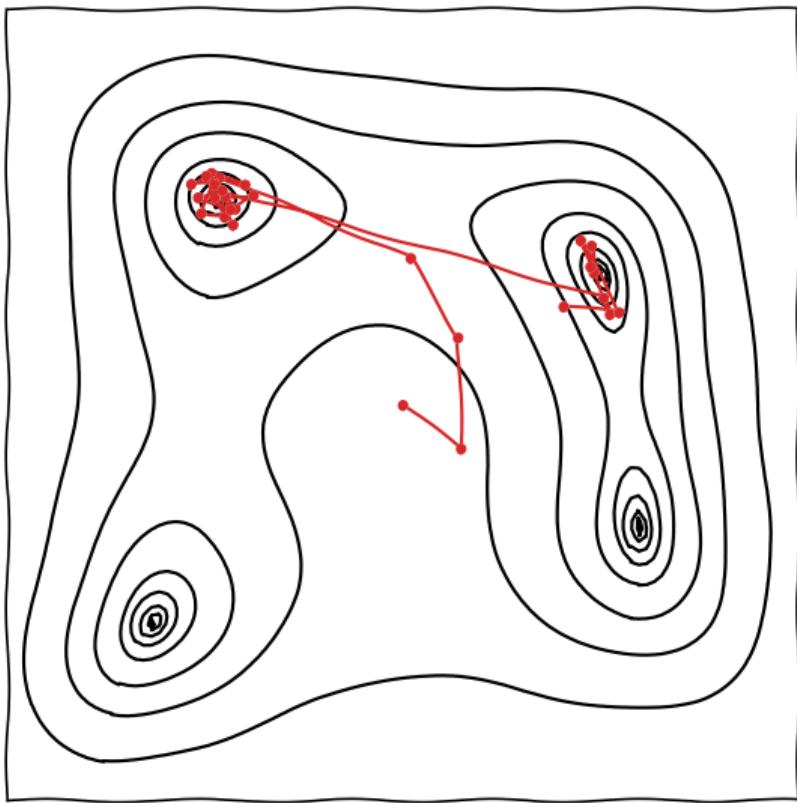
MCMC



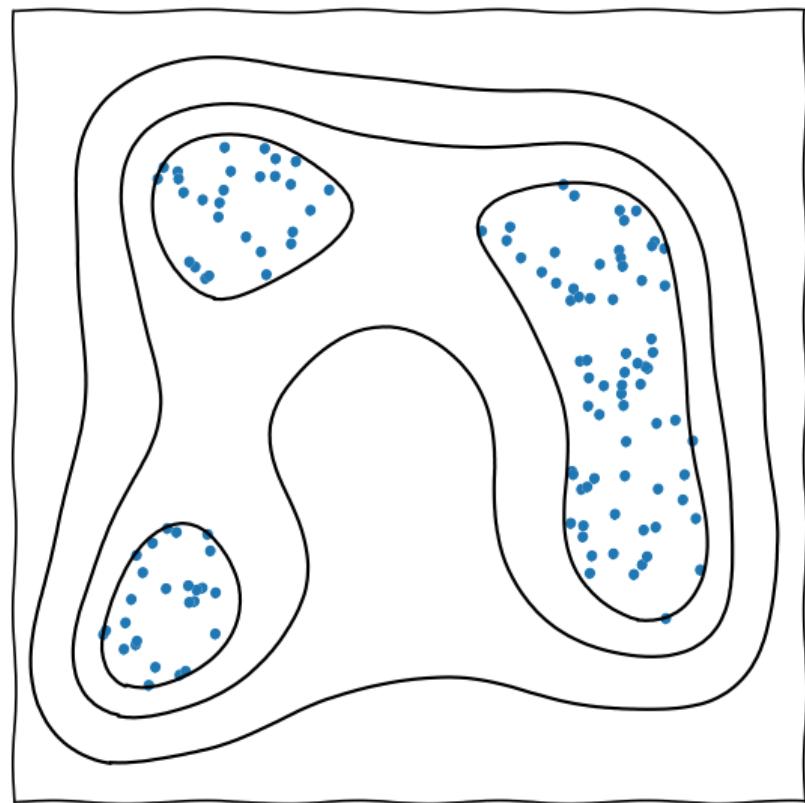
Nested sampling



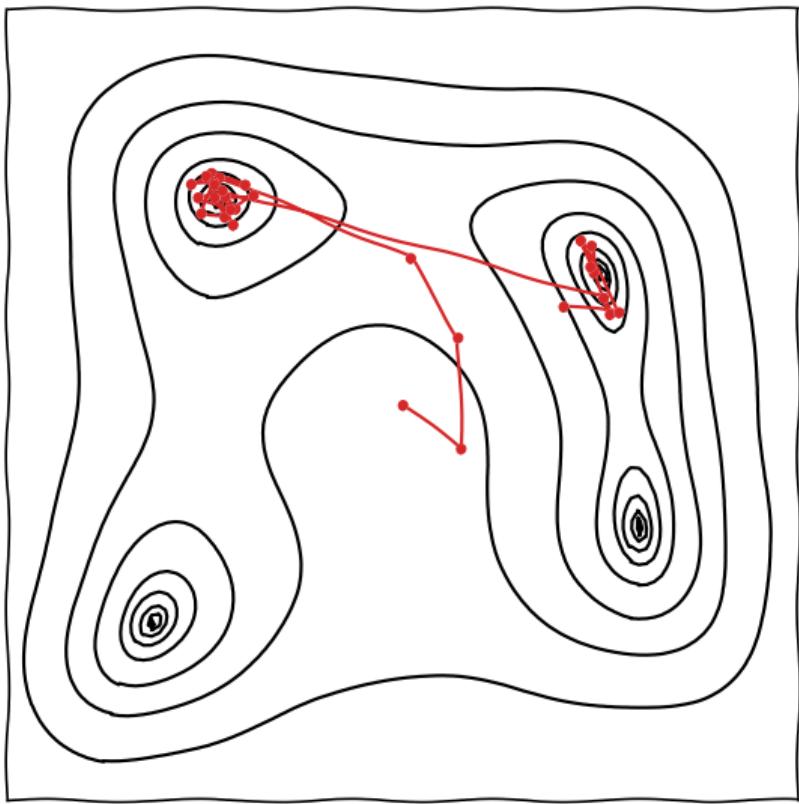
MCMC



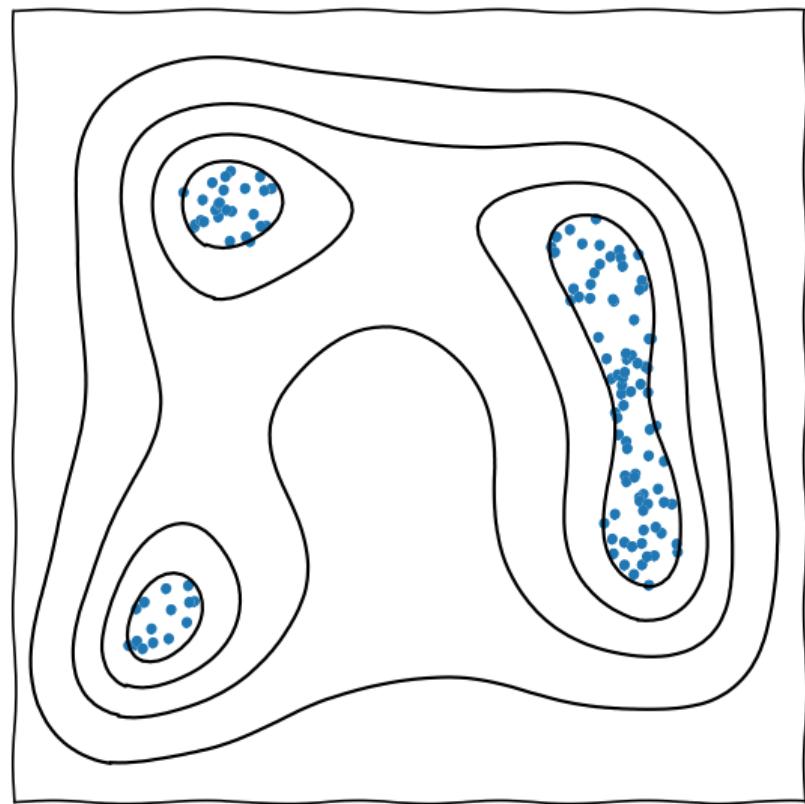
Nested sampling



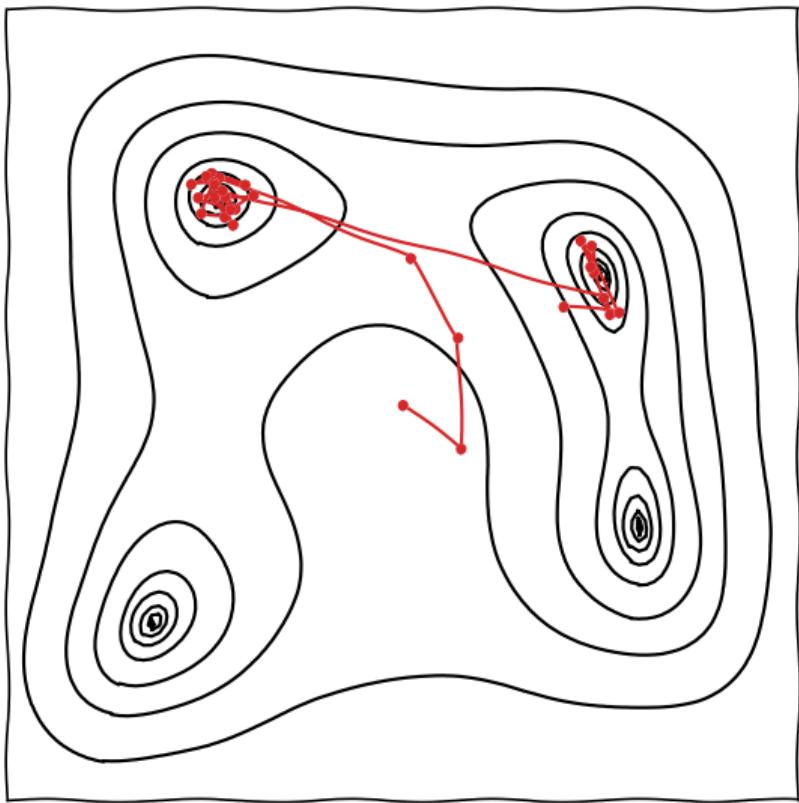
MCMC



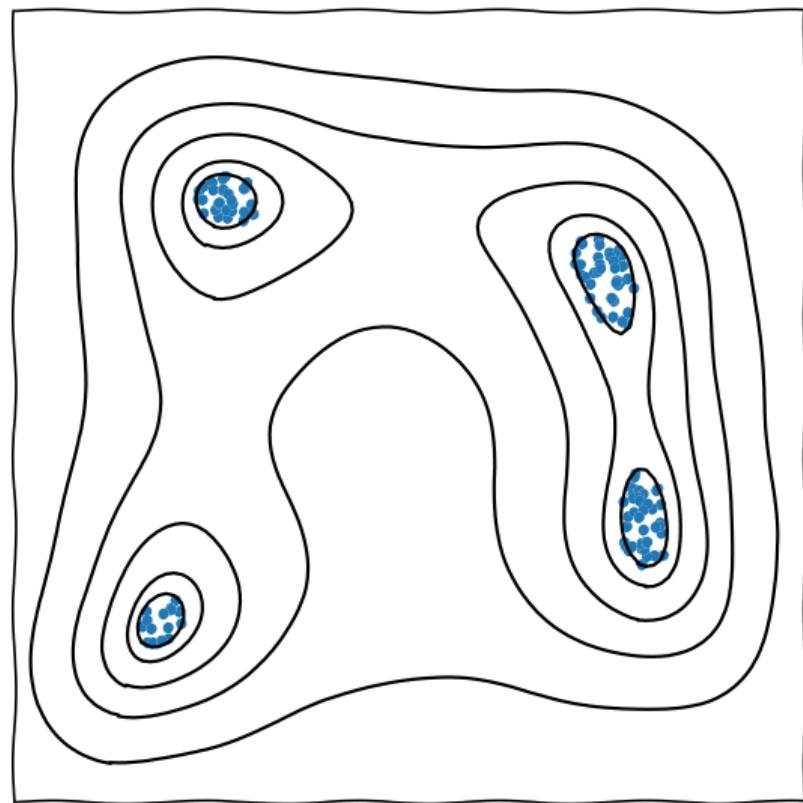
Nested sampling



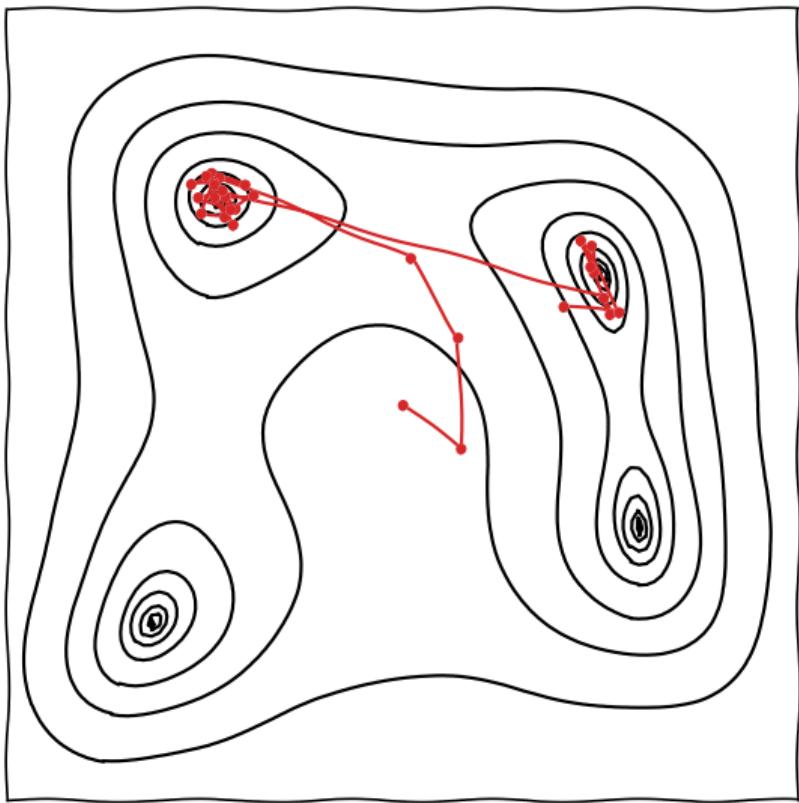
MCMC



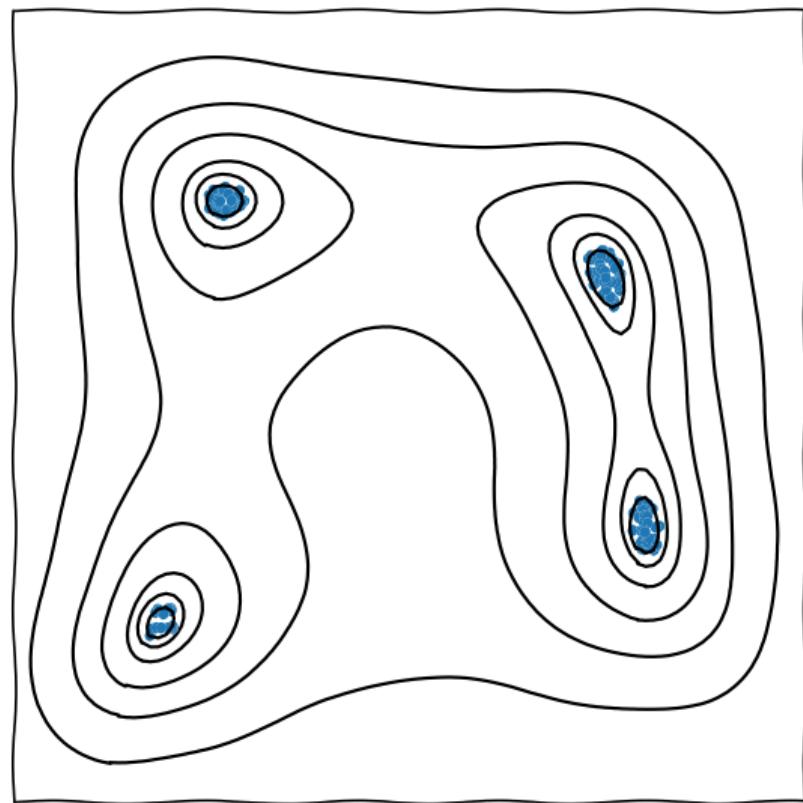
Nested sampling



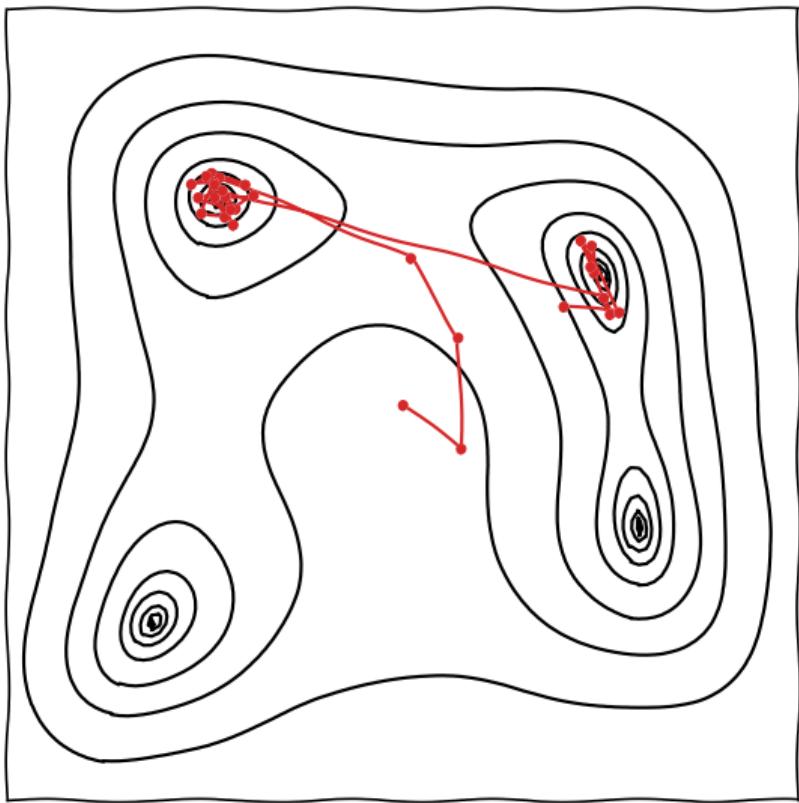
MCMC



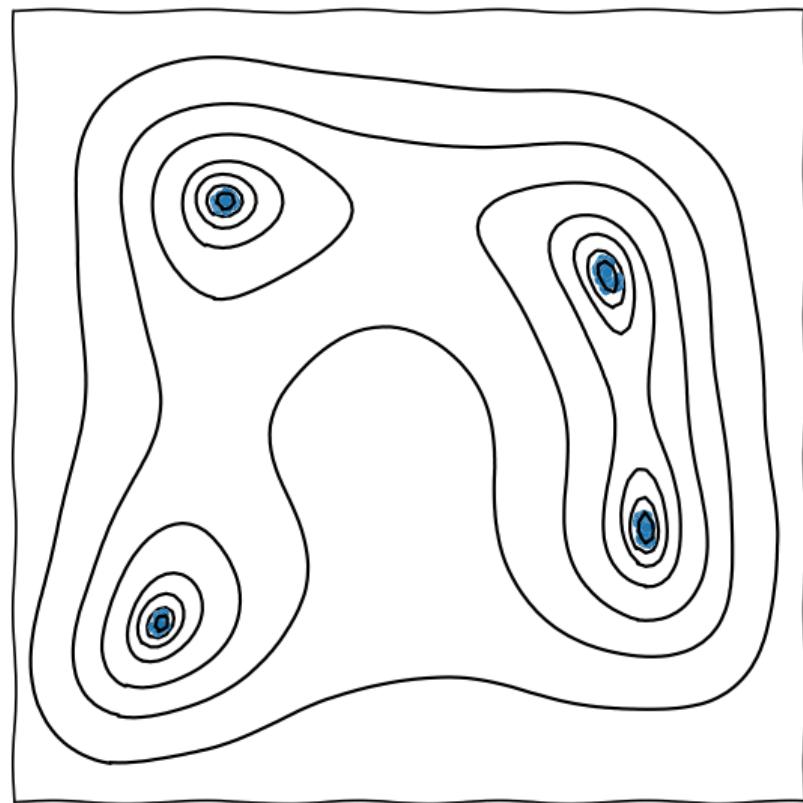
Nested sampling



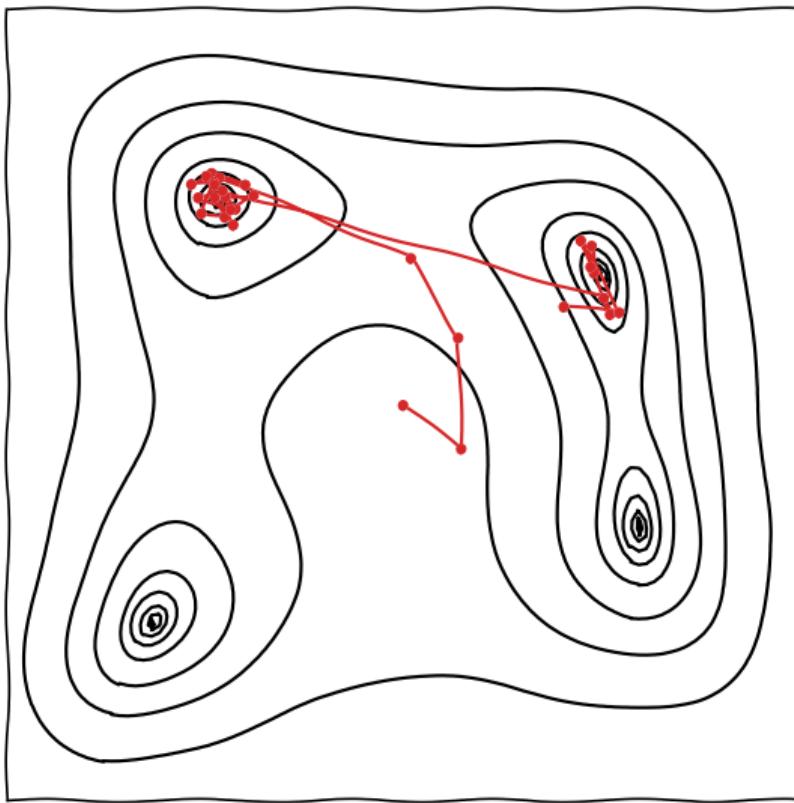
MCMC



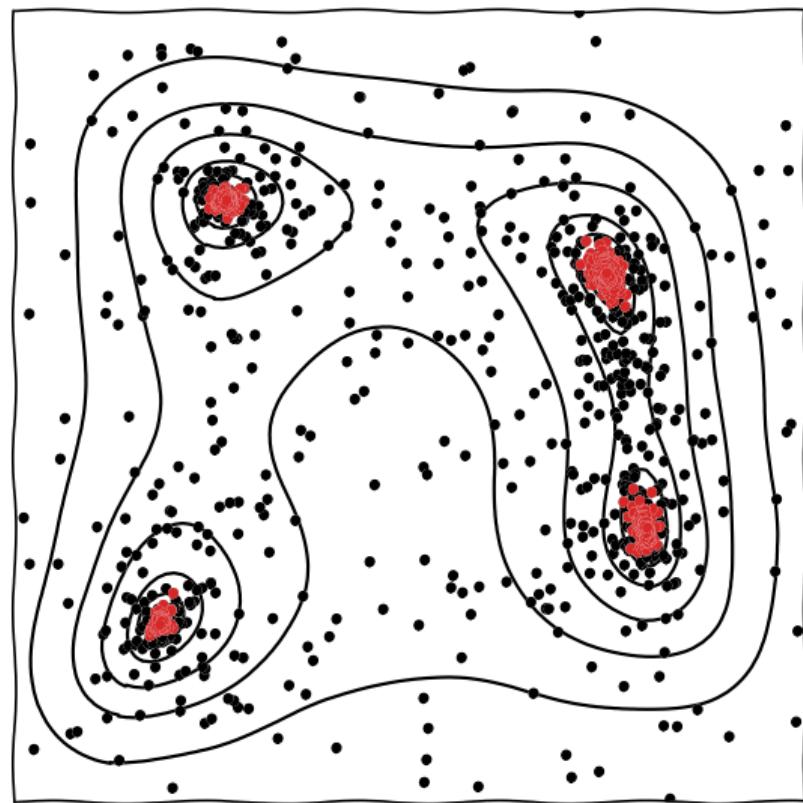
Nested sampling



MCMC

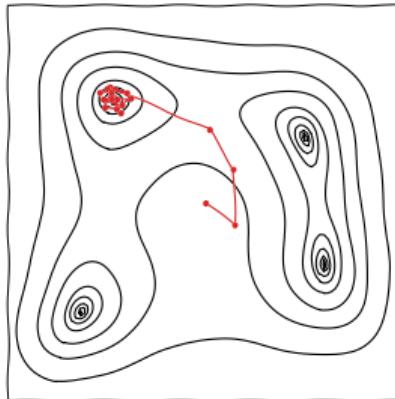


Nested sampling



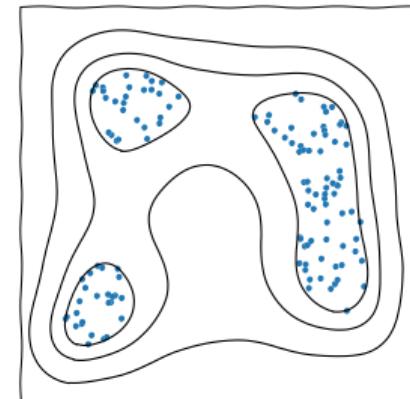
MCMC

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



Nested sampling

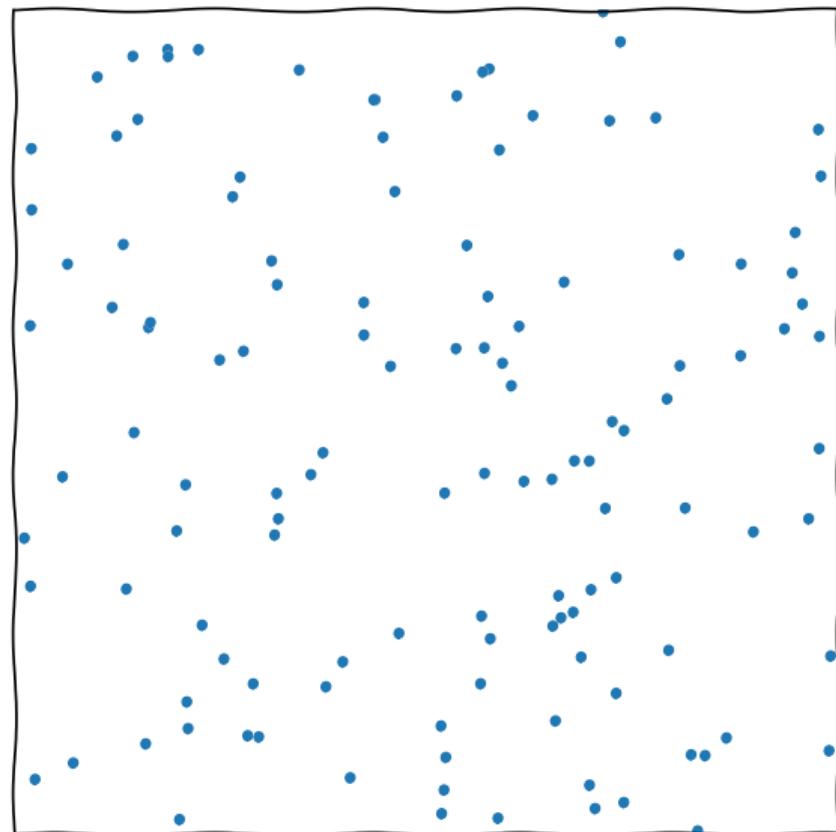
- ▶ Ensemble of “live points”
- ▶ Scans from prior to peak of likelihood
- ▶ Slower, no tuning required
- ▶ Parameter estimation, model comparison, tension quantification
- ▶ Channel capacity optimised for computing partition function



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

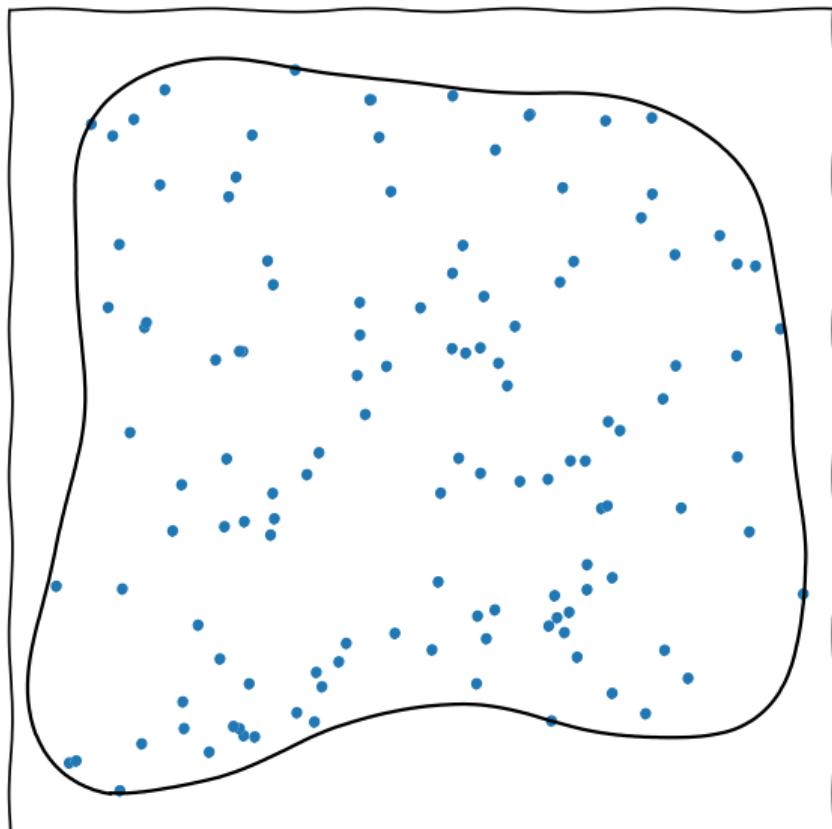
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

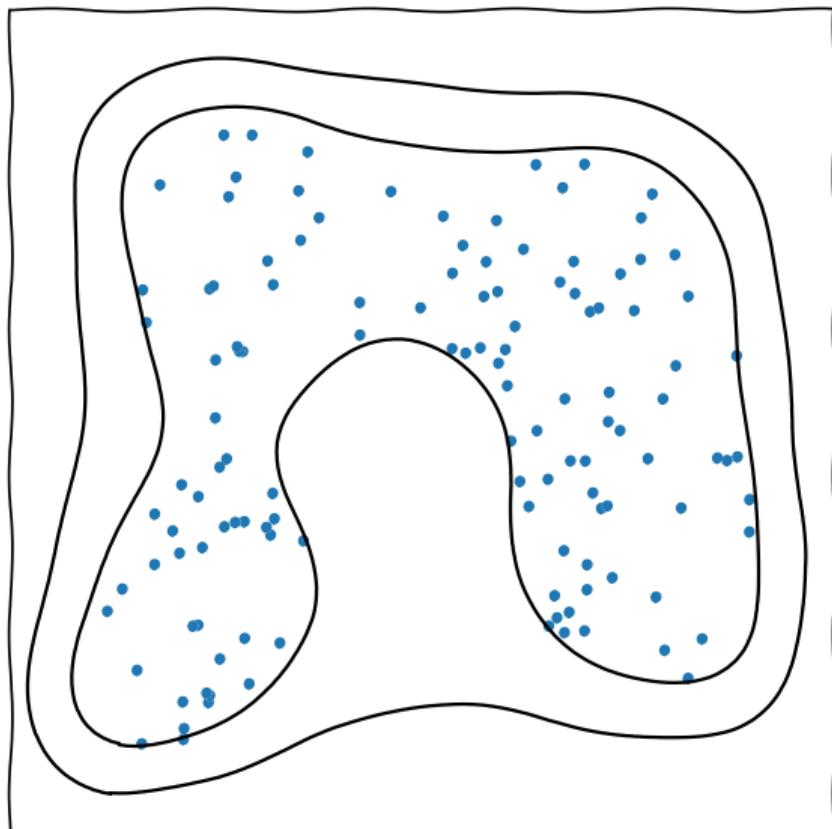
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

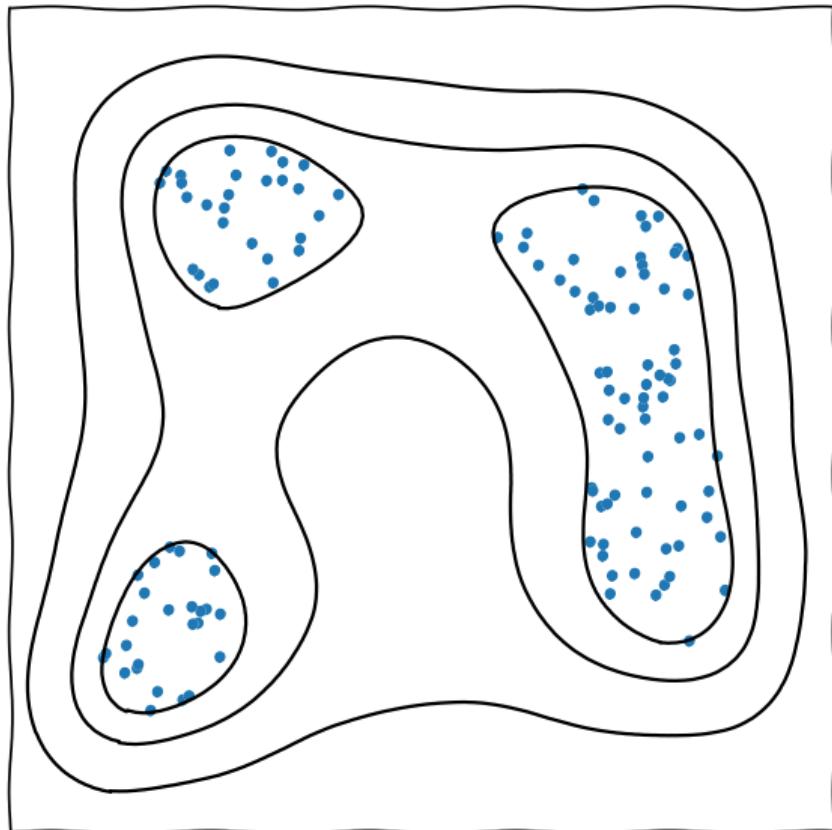
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

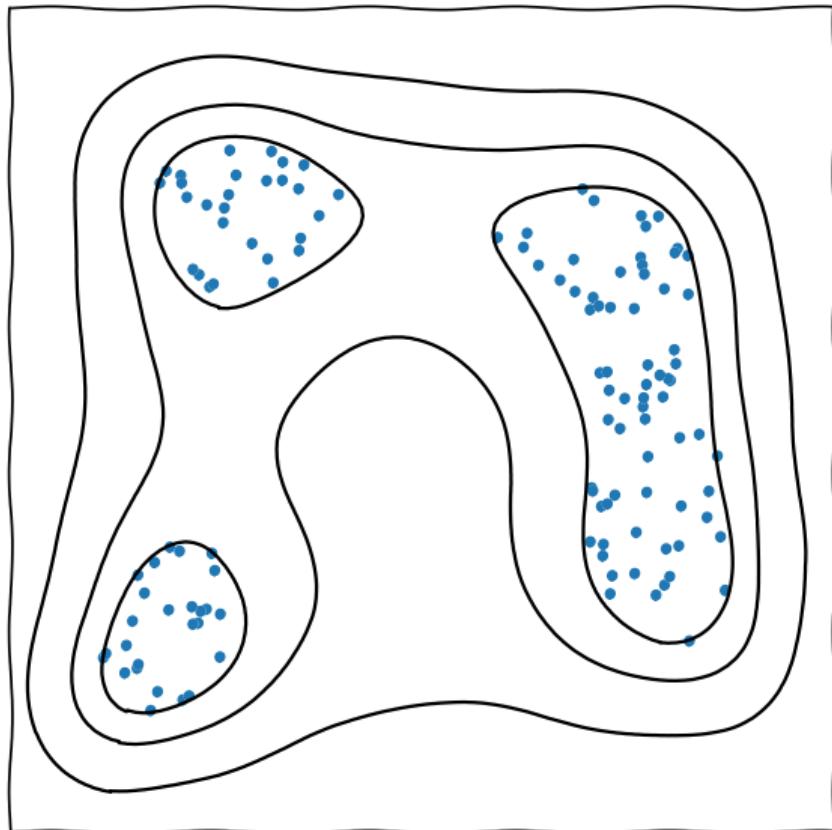
$$\int f(x)dV \approx \sum_i f(x_i)\Delta V_i, \quad V_i = V_0 e^{-i/n}$$



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

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



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 $d > d_0 \sim 10$.

- ▶ 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, 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} .

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.