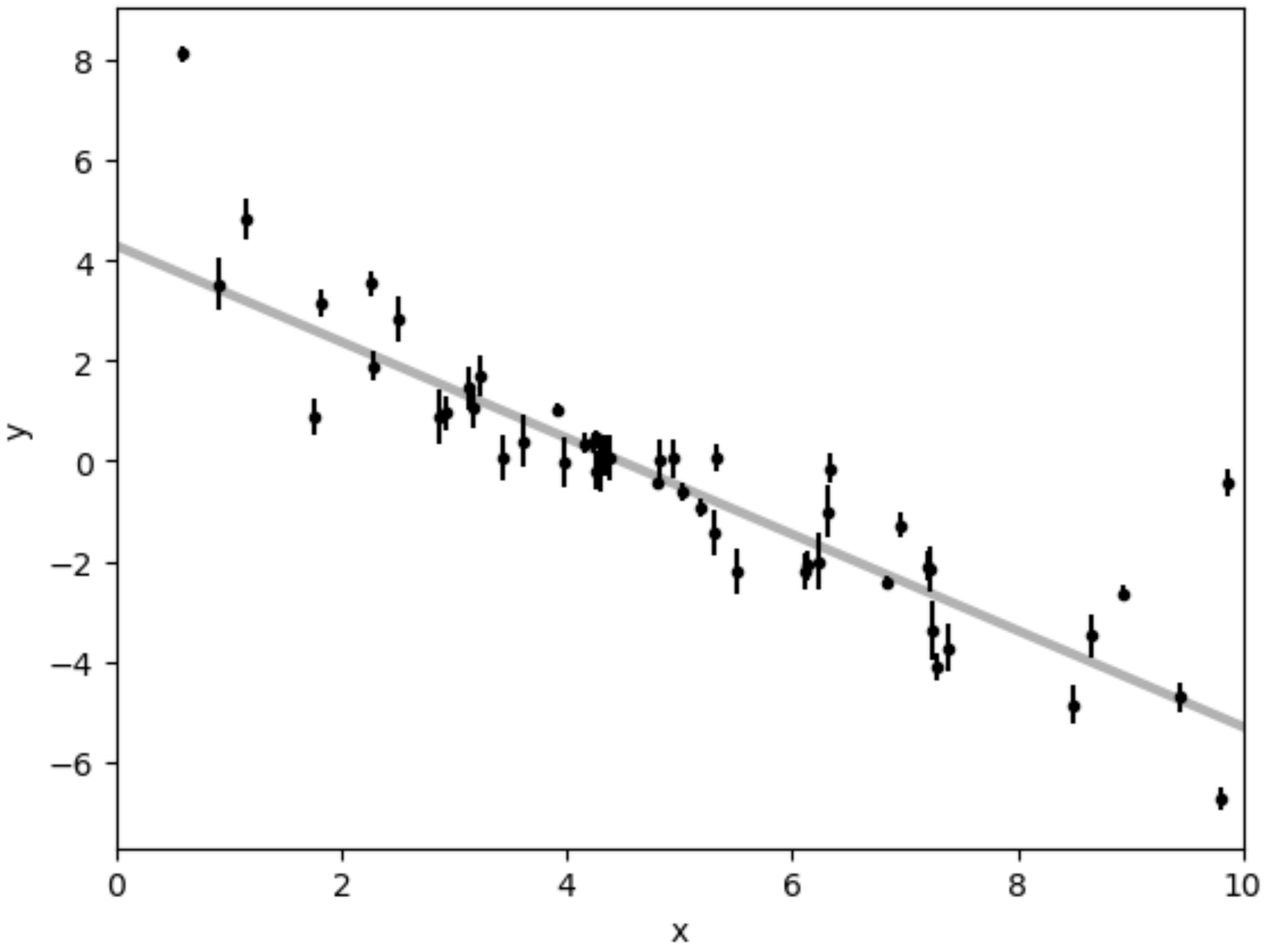
Advanced Sampling Techniques

LSST Discovery Fellowship Program Day 3

Greg Gilbert | LSST Discovery Workshop | 22 May 2025

Fitting a line to data



Modeling choices

Physical
What processes do you include?
What approximations do you make?

Statistical
Are data i.i.d.?
Is there correlated noise?
Do you account for data collection?

Model specification
Parameterization
Priors
Convergence criteria

Sampler
Grid search
Maximum likelihood
Markov Chain Monte Carlo
Nested Sampling

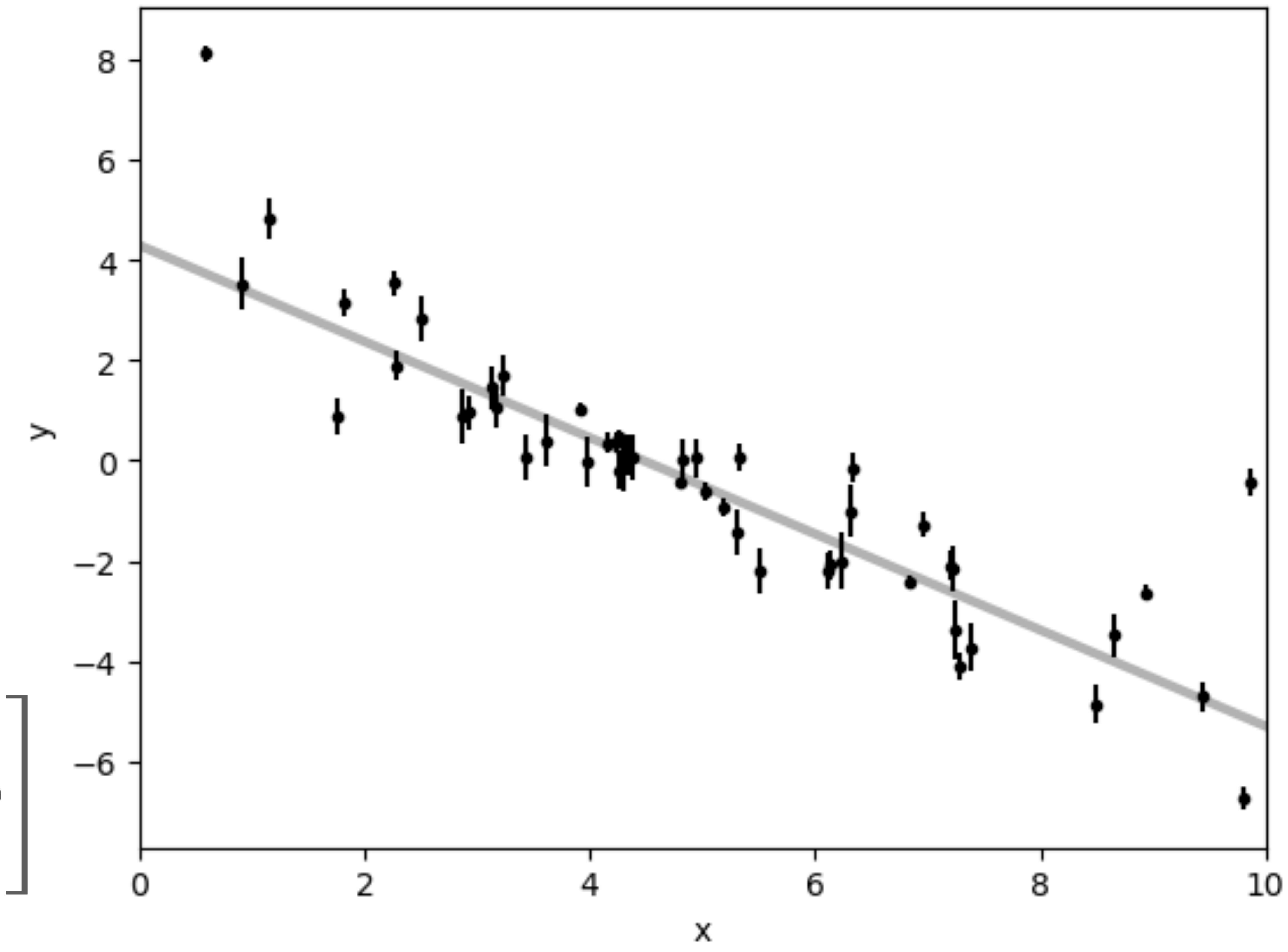
Fitting a line to data

We will build a generative model

$$y_{\rm mod} = mx + b$$

$$\sigma^2_{\rm tot} = \sigma^2_{\rm obs} + s^2$$

$$\ln \mathscr{L}(\theta) = -\frac{1}{2}\sum_i \left[\frac{(y_{{\rm obs},i} - y_{{\rm mod},i})^2}{\sigma^2_{{\rm tot},i}} + \ln(2\pi\sigma^2_{{\rm tot},i})\right]$$



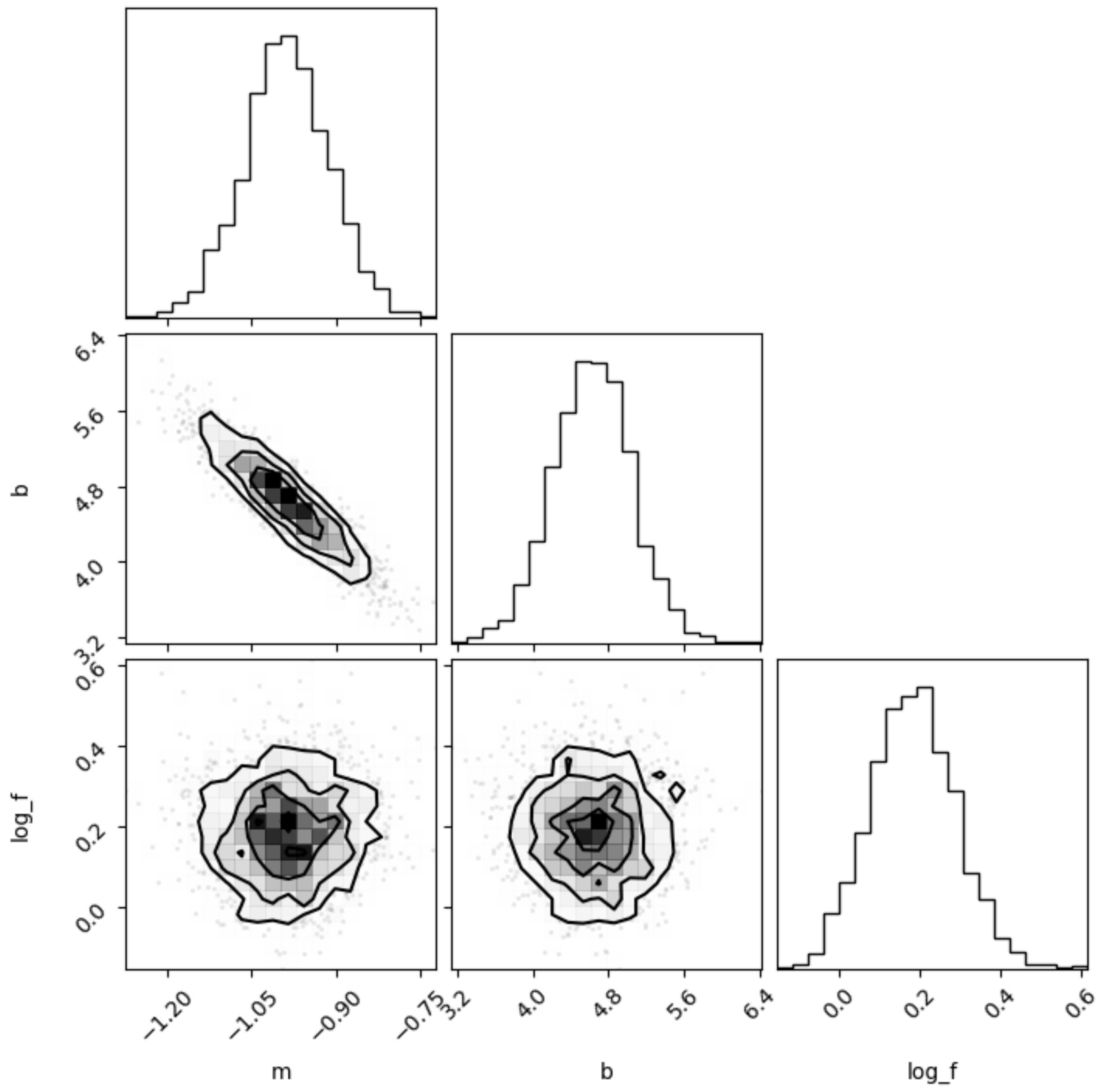
We have already made many implicit and explicit assumptions about the data generating process

Fitting a line to data

```python
import pymc3 as pm
import pymc3_ext as pmx

with pm.Model() as model:
    m = pm.Uniform("m", lower=-10, upper=10)
    b = pm.Uniform("b", lower=-10, upper=10)
    log_f = pm.Normal("log_f", mu=0, sd=10)

    y_mod = pm.Deterministic("y_mod", m*x + b)
    s_mod = pm.math.sqrt(pm.math.exp(log_f)**2 + y_err**2)


    lnlike = pm.Normal("lnlike", mu=y_mod, sd=s_mod, observed=y_obs)

with model:
    trace = pmx.sample(chains=2, tune=1000, draws=1000, target_accept=0.9, return_inferencedata=True)
```

```
Multiprocess sampling (2 chains in 4 jobs)
NUTS: [log_f, b, m]
```

```
100.00% [4000/4000 00:00<00:00 Sampling 2 chains, 0 divergences]
```



Look at those lovely Gaussian posteriors!

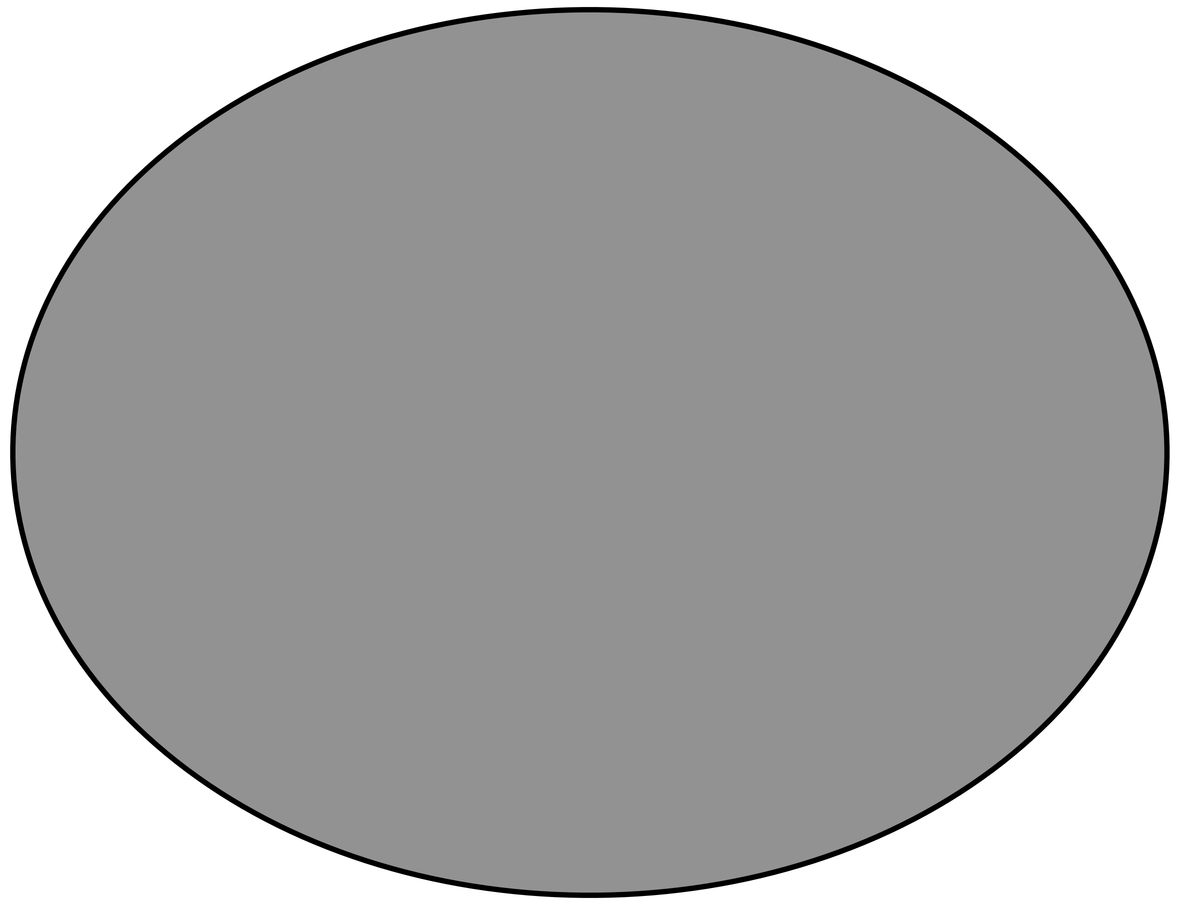
Easy mode: low-dimensional Gaussian

Small parameter covariances

Smooth, homogenous, isotropic posterior topology

Computationally cheap

So simple!



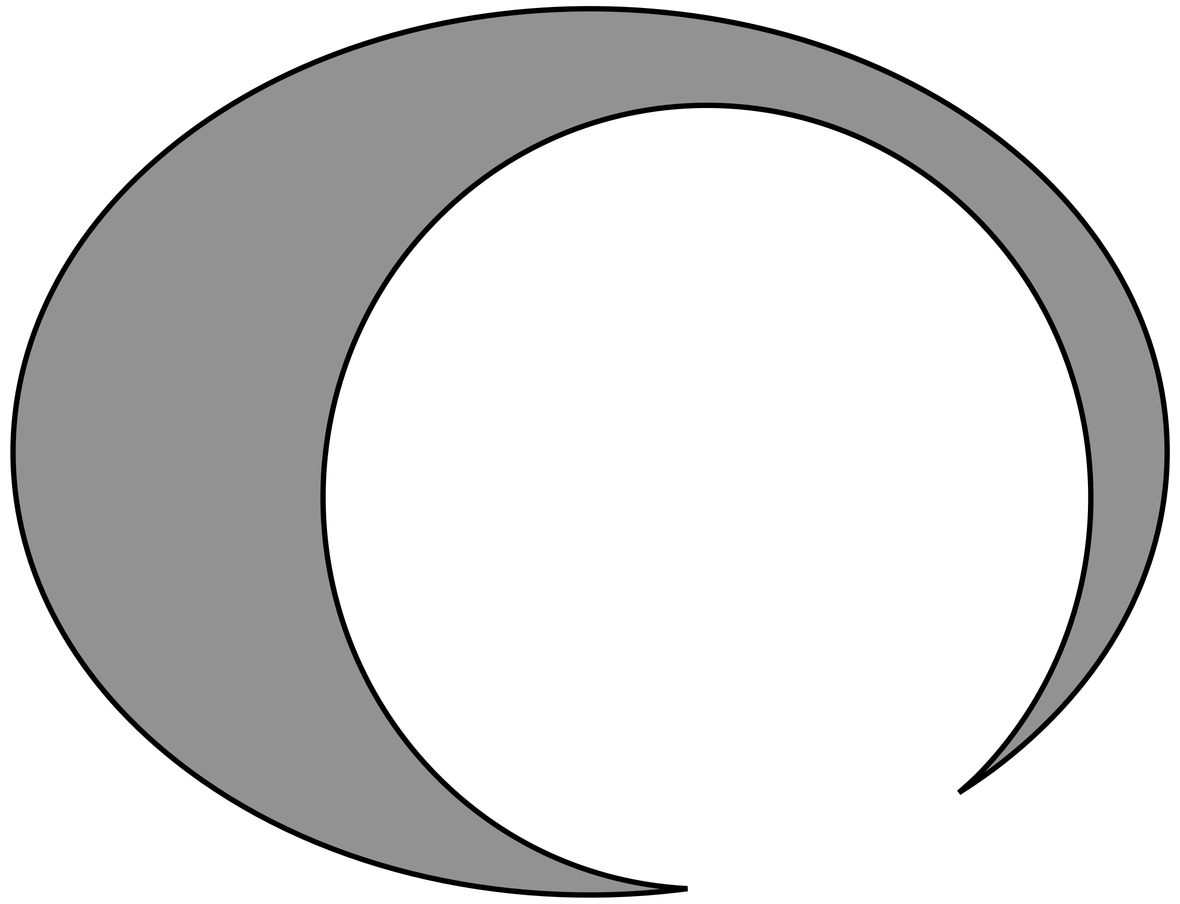
Hard mode: real data

Strong or unknown covariances

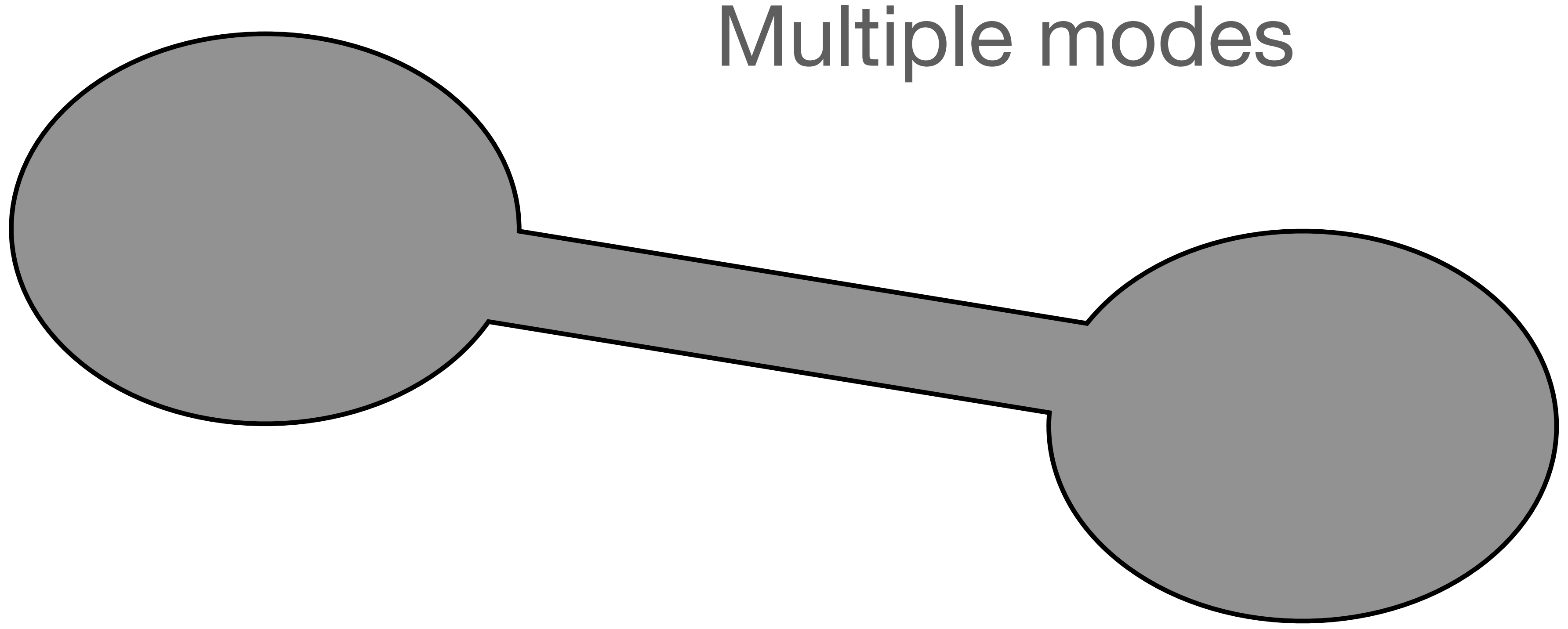
Inhomogeneous (and unknown) posterior topology

Computational cost rapidly scales with number of free parameters

Strong curvature

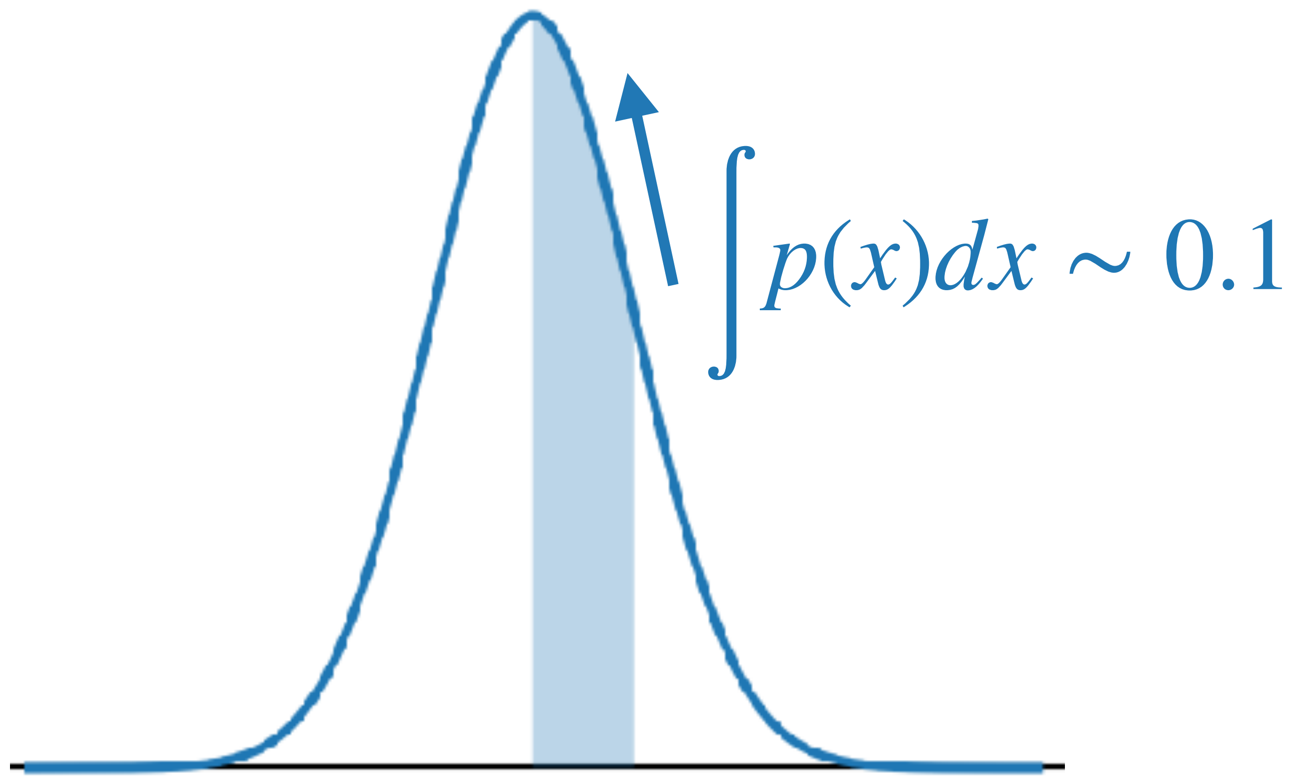


Multiple modes

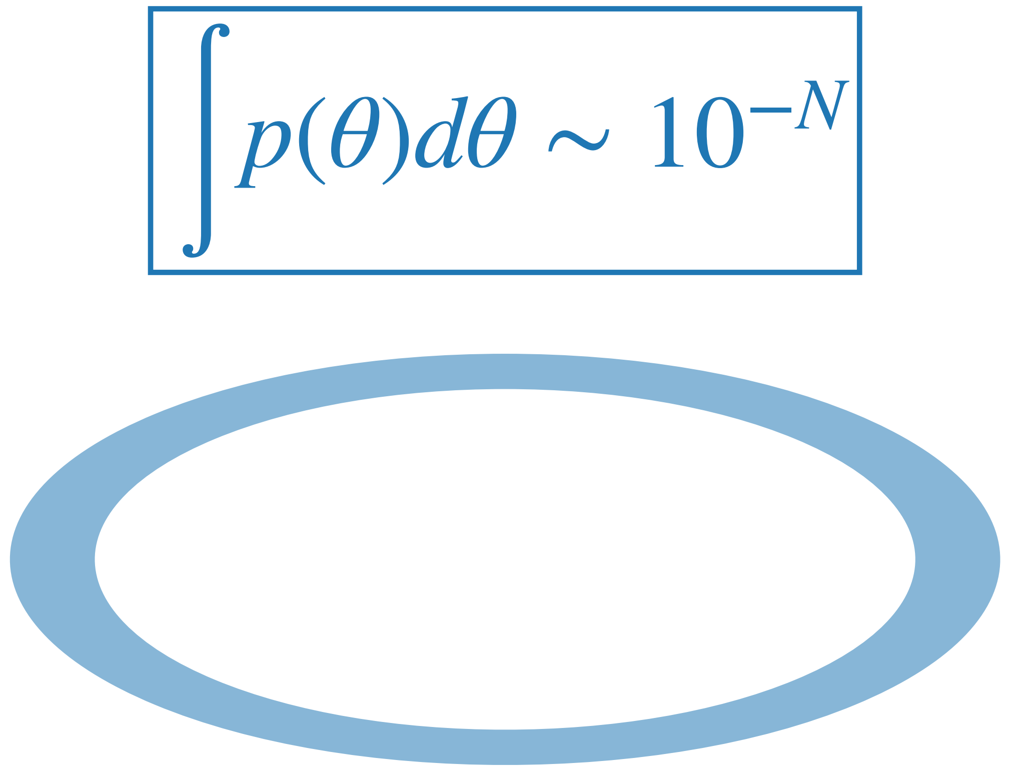


Sneaky mode: high dimensions

As the number of free parameters increases, the "typical set" is a thin shell, even for low-covariance topologies

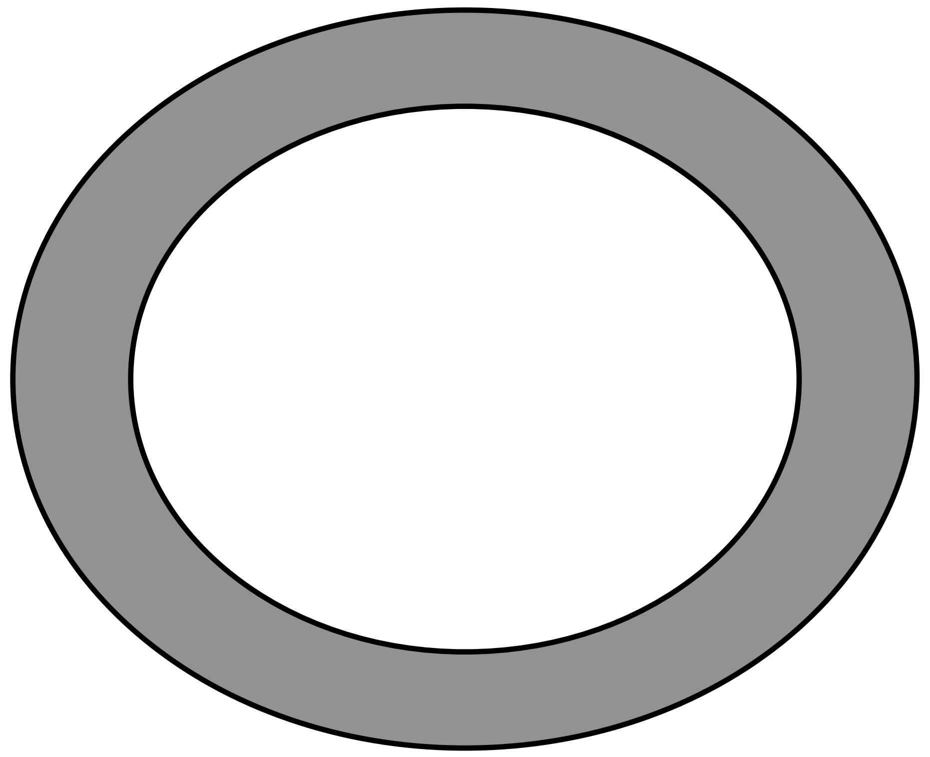


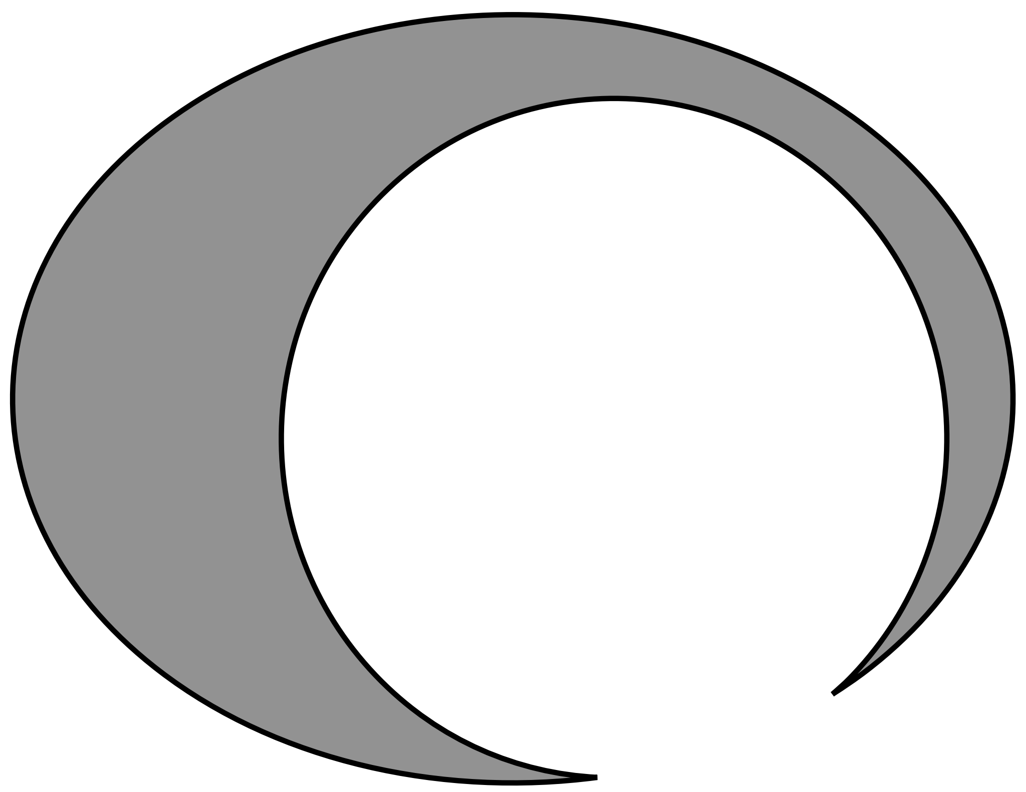
Near maximum likelihood, relatively little posterior volume interior to current position

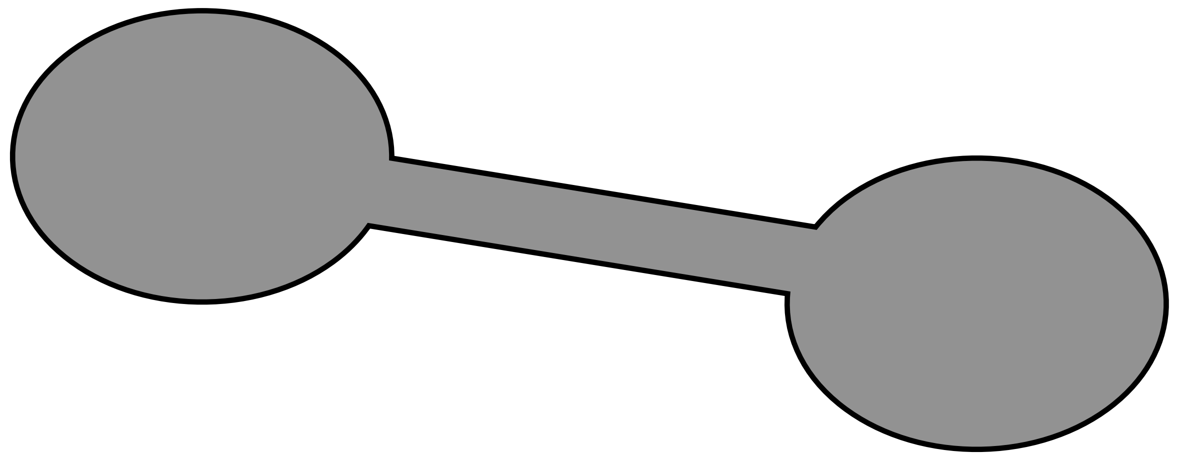


In high dimension, chance of all parameters simultaneously moving toward maximum likelihood becomes vanishingly small

Mission: generate samples from complicated posterior topologies







Option 1: Change the sampler

Ensemble Samplers
Hamiltonian Monte Carlo
Nested sampling

Option 2: Change the topology

Re-parameterize
Importance sampling
Umbrella sampling

Bayes Theorem

Likelihood

Prior

$$P(\theta \mid D) = \frac{P(D \mid \theta)\, P(\theta)}{P(D)}$$

Posterior

Evidence

MCMC generates **samples from the posterior**

Bayes Theorem

$$P(\theta \mid D, M) = \frac{P(D \mid \theta, M)\, P(\theta \mid M)}{P(D \mid M)}$$

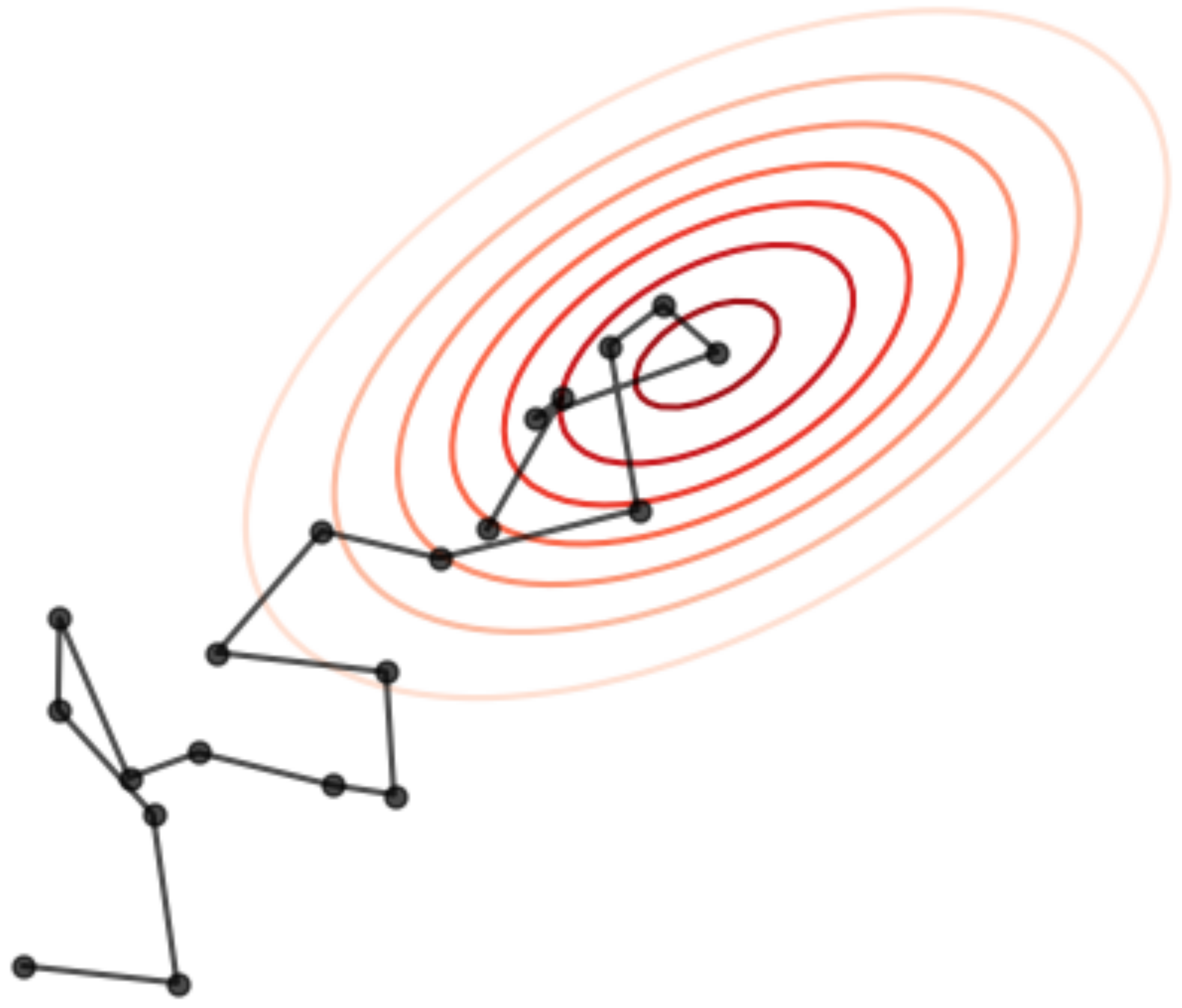
Implicit assumption of a particular **model**

Random Walk Monte Carlo

1. Choose an initial θ

2. Loop over N iterations
   a. Propose a new θ' from proposal $q(\theta)$
   b. Compute acceptance ratio $\alpha = p(\theta')/p(\theta)$
   c. Generate random number $u \sim U(0,1)$
      - If $u \leq \alpha \rightarrow$ accept, $\theta_{i+1} = \theta'$
      - If $u > \alpha \rightarrow$ reject, $\theta_{i+1} = \theta$

3. Stop when N_{eff} is above desired threshold



MCMC Variations

Method	**Description**
Metropolis-Hastings	Walkers move according to fixed proposal distribution
Gibbs Sampling	Walkers propose steps one variable at a time
Differential Evolution	Scales steps size according to ensemble of walkers
Affine Invariant	Adapts proposals to geometry of walker ensemble
Parallel Tempering	Runs ensemble of walkers at different “temperatures”
Hamiltonian	Simulates “momentum” for each walker to take long steps through parameter space

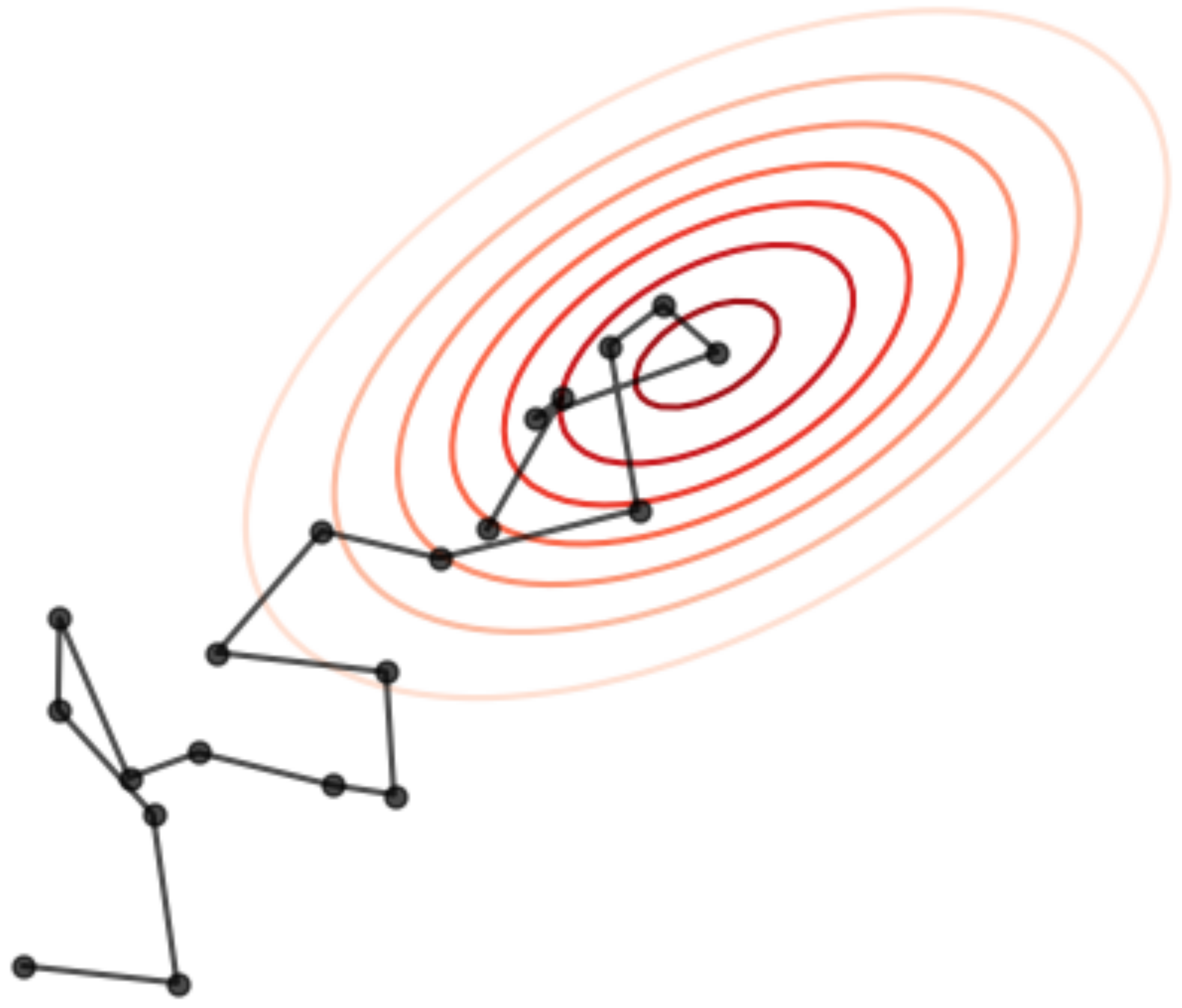
Metropolis-Hastings Monte Carlo

Almost identical to Random Walk Monte Carlo, but algorithm is modified to allow for asymmetric proposal steps

$$\text{Random Walk}: \alpha = \frac{p(\theta')}{p(\theta)}$$

$$\text{Metropolis-Hastings}: \alpha = \frac{p(\theta')}{p(\theta)} \frac{q(\theta \,|\, \theta')}{q(\theta' \,|\, \theta)}$$

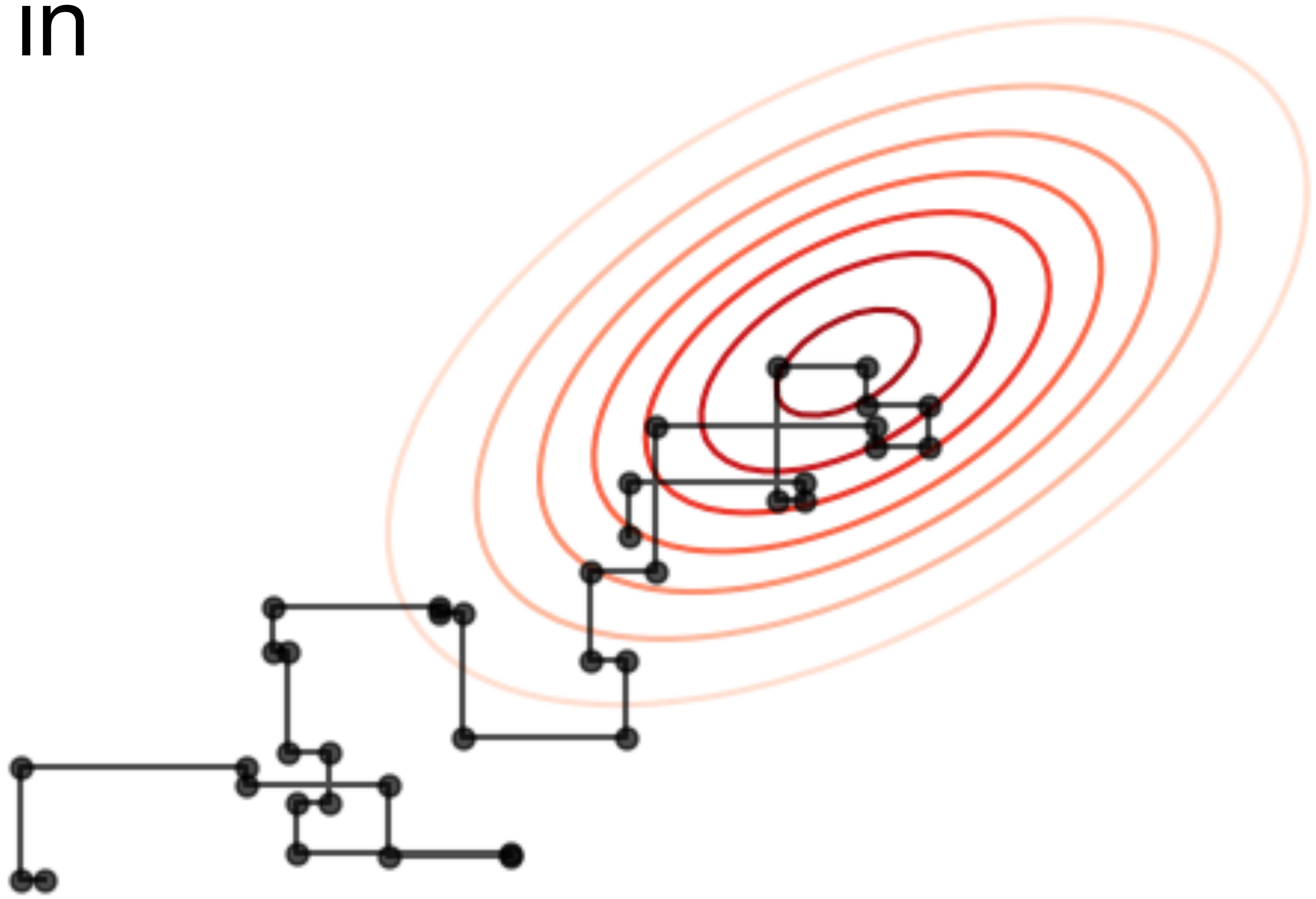
Proposal Distribution



Gibbs Sampling

Very similar to Metropolis-Hastings, but steps in one parameter at a time

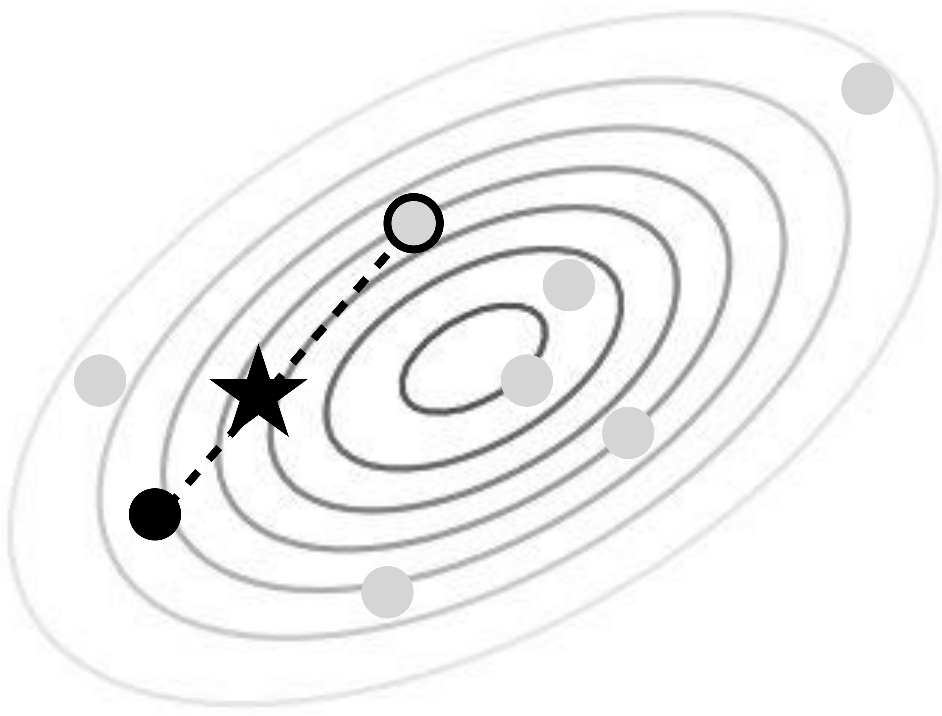
This can be beneficial for high-dimensional problems when acceptance fraction would be low by stepping in all parameters at once.



Ensemble Samplers

Differential Evolution

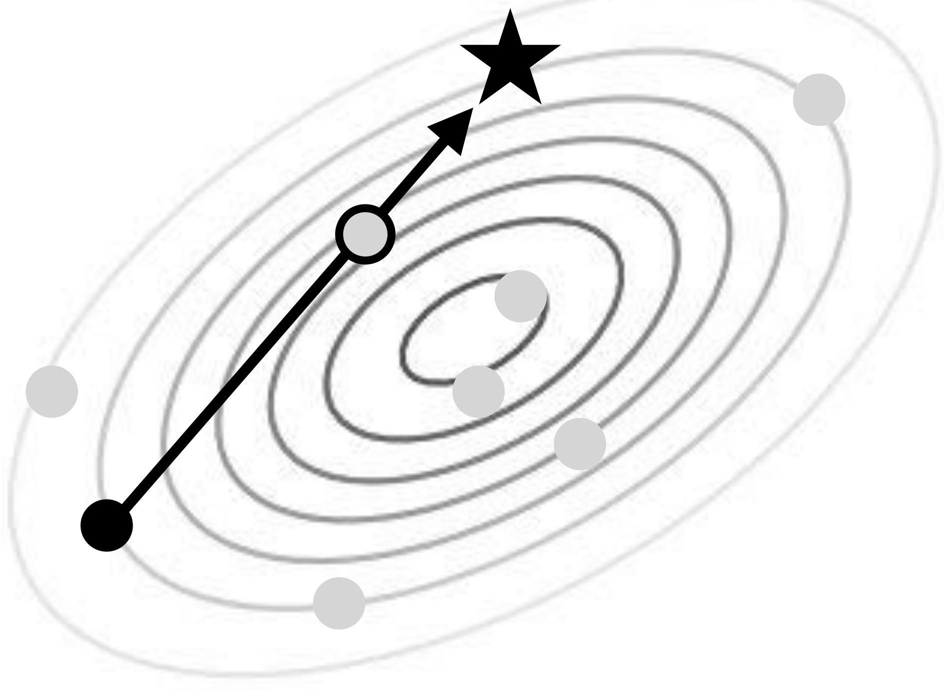
"Genetic mutation" proposal centered around current walker



Ter Braak (2004, 2006)

Affine Invariant

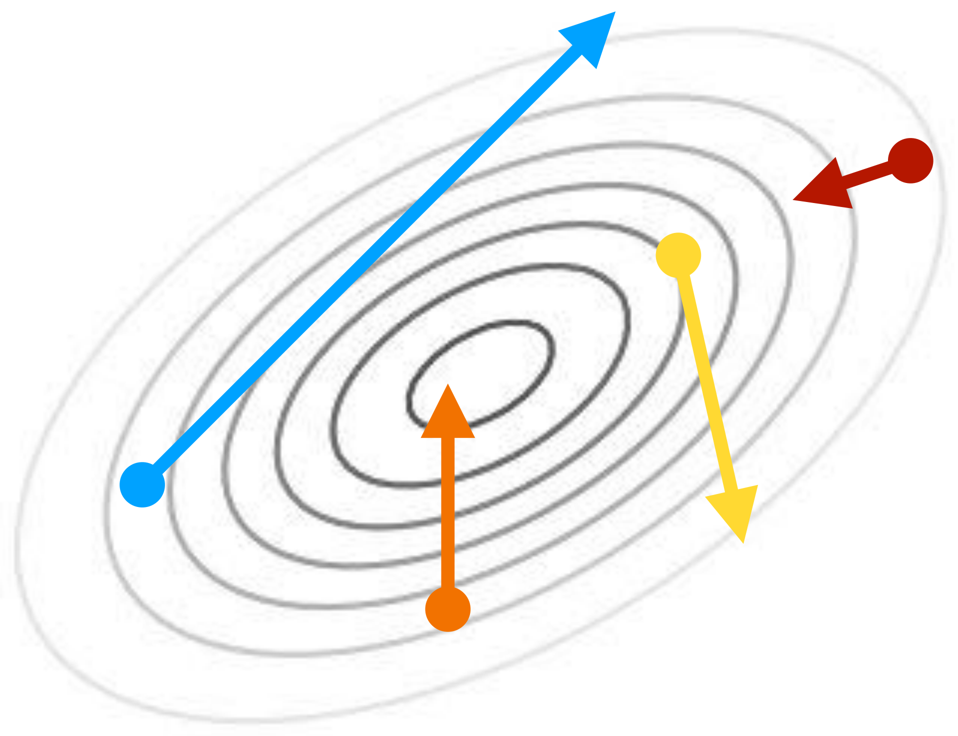
"Stretch move" proposal centered around reference walker



Goodman & Weare (2010)

Parallel Tempering

Walkers have different "temperatures" and occasionally swap positions



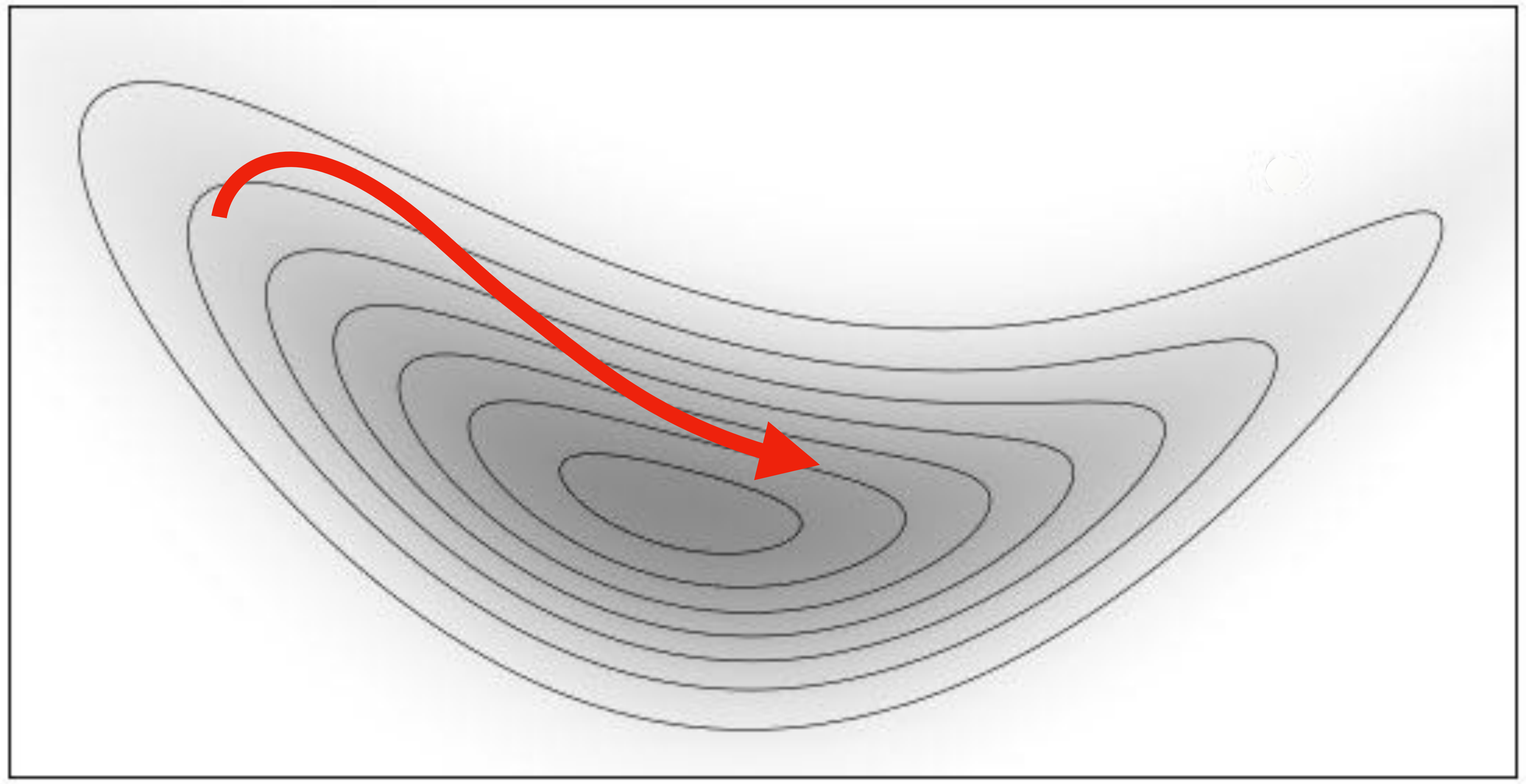
Earl & Deem (2005)

Hamiltonian Monte Carlo

Instead of “walkers” we have “particles” described by both position θ and momentum ν

Particle motion is analogous to rolling a ball around a basin

Higher computational cost per step
Higher acceptance fraction $\alpha \gtrsim 0.9$
Short autocorrelation length $\tau \approx 2$



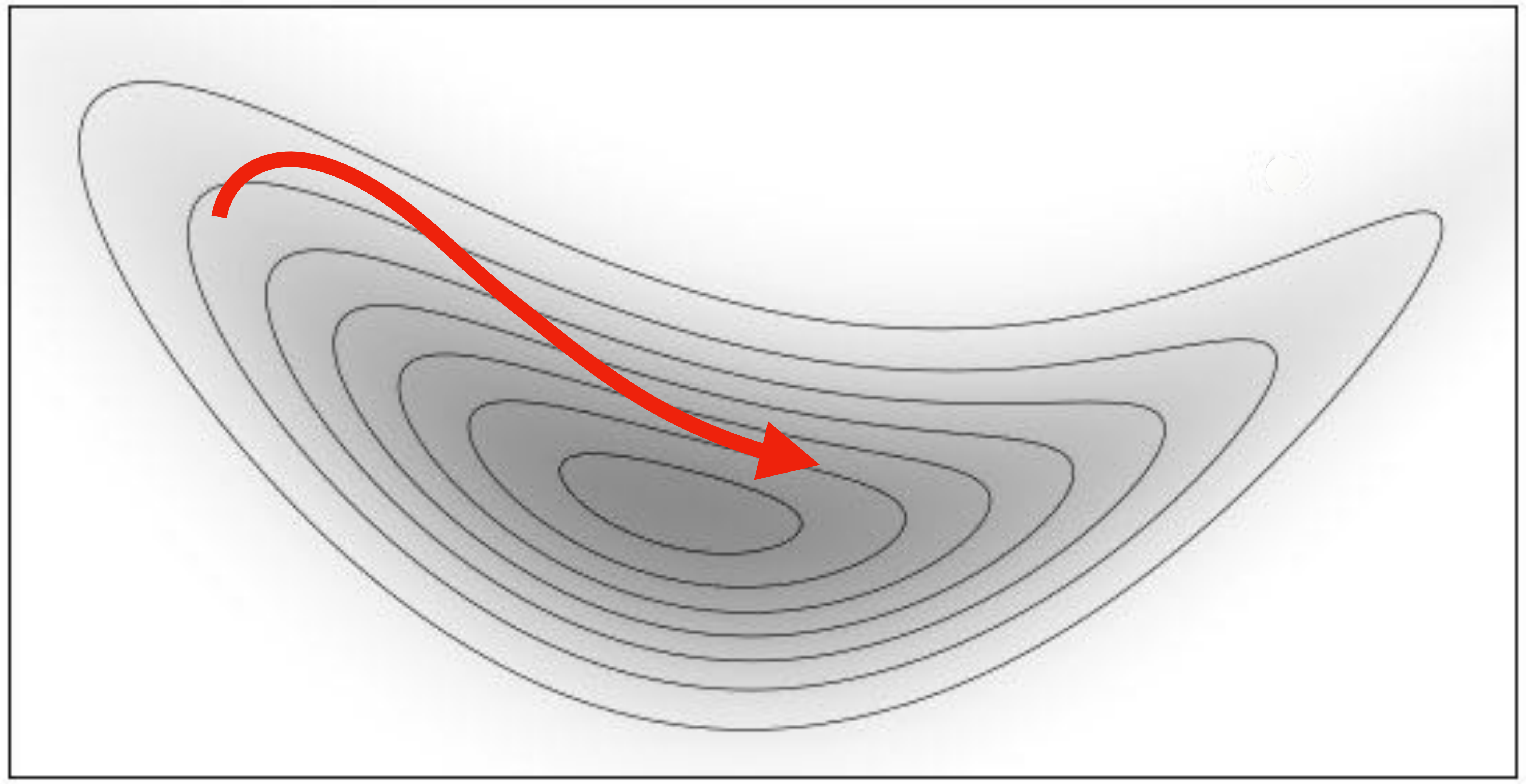
Step size can now be large and traces the curvature of the posterior topology

Hamiltonian Monte Carlo

1. Define a Hamiltonian system
   $H = U(\theta) + K(\nu)$

2. For each step…
   a. Give momentum a 'kick'
   b. Integrate particle trajectory
   c. Almost always accept proposal

3. Stop at predetermined N



$\boxed{U(\theta) \propto -\log[p(\theta \,|\, D)]}$

Bayes Theorem

Likelihood, L

Prior, Π

$$P(\theta \mid D) = \frac{P(D \mid \theta)\, P(\theta)}{P(D)}$$

Evidence, Z

Nested sampling generates **samples from the prior***

*subject to constraint $L > \lambda$, with the goal of calculating Z

MCMC vs Nested Sampling

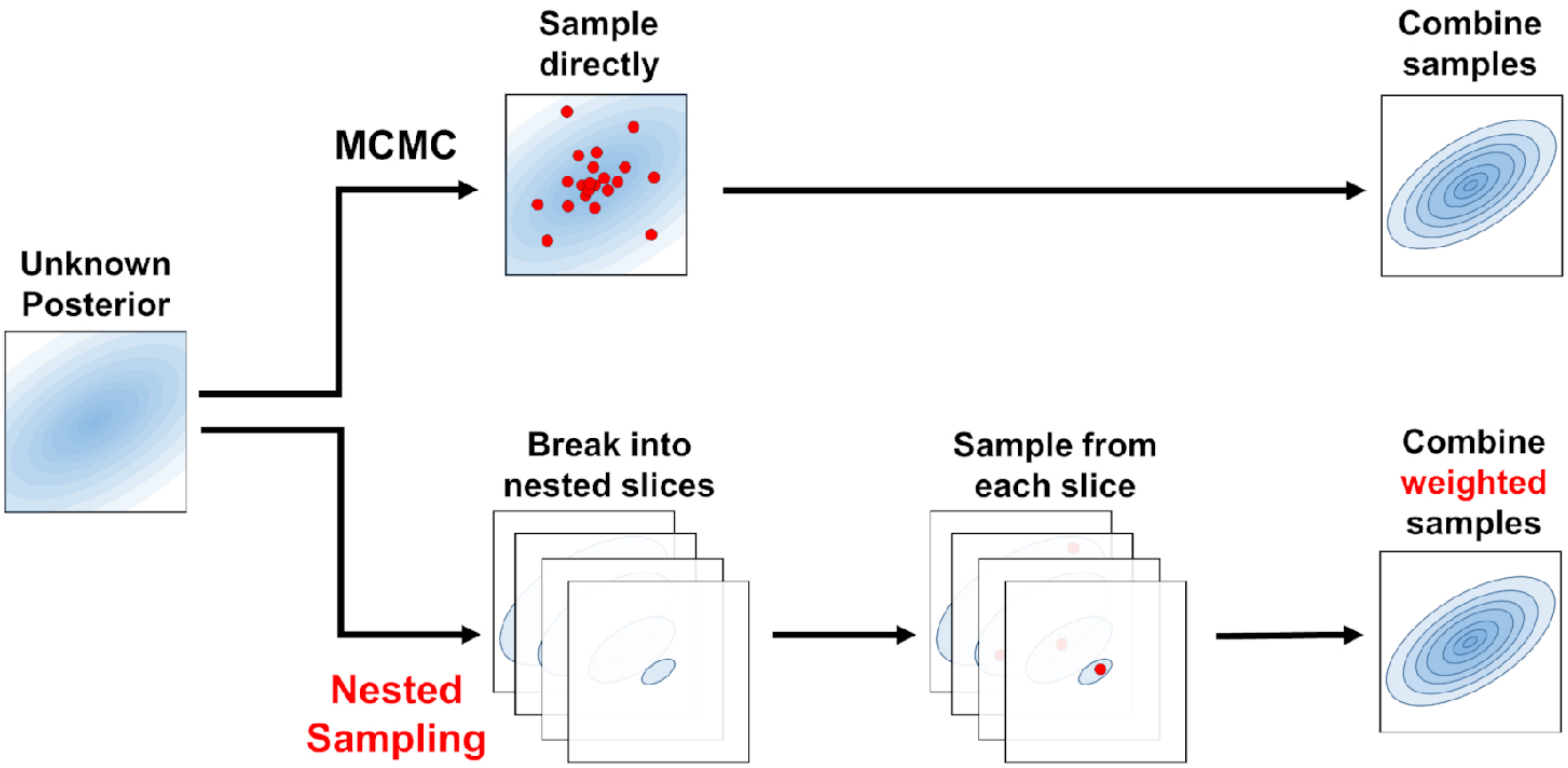


Image Credit: Speagle 2020

Nested Sampling

```
// Initialize live points.
```
Draw K “live” points$\{\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_K\}$ from the prior $\pi(\mathbf{\Theta})$.
```
// Main sampling loop.
```
while stopping criterion not met **do**

  Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.

  Add the kth live point $\mathbf{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of “dead” points.

  Sample a new point $\mathbf{\Theta}'$ from the prior subject to the constraint$\mathcal{L}(\mathbf{\Theta}') \geq \mathcal{L}^{\min}$.

  Replace $\mathbf{\Theta}_k$ with $\mathbf{\Theta}'$.
```
  // Check whether to stop.
```
  Evaluate stopping criterion.

end
```
// Add final live points.
```
while $K > 0$ **do**

  Compute the minimum likelihood $\mathcal{L}^{\min}$ among thecurrent set of live points.

  Add the kth live point $\mathbf{\Theta}_k$ associated with $\mathcal{L}^{\min}$to a list of “dead” points.

  Remove $\mathbf{\Theta}_k$ from the set of live points.

  Set $K = K - 1$.

end

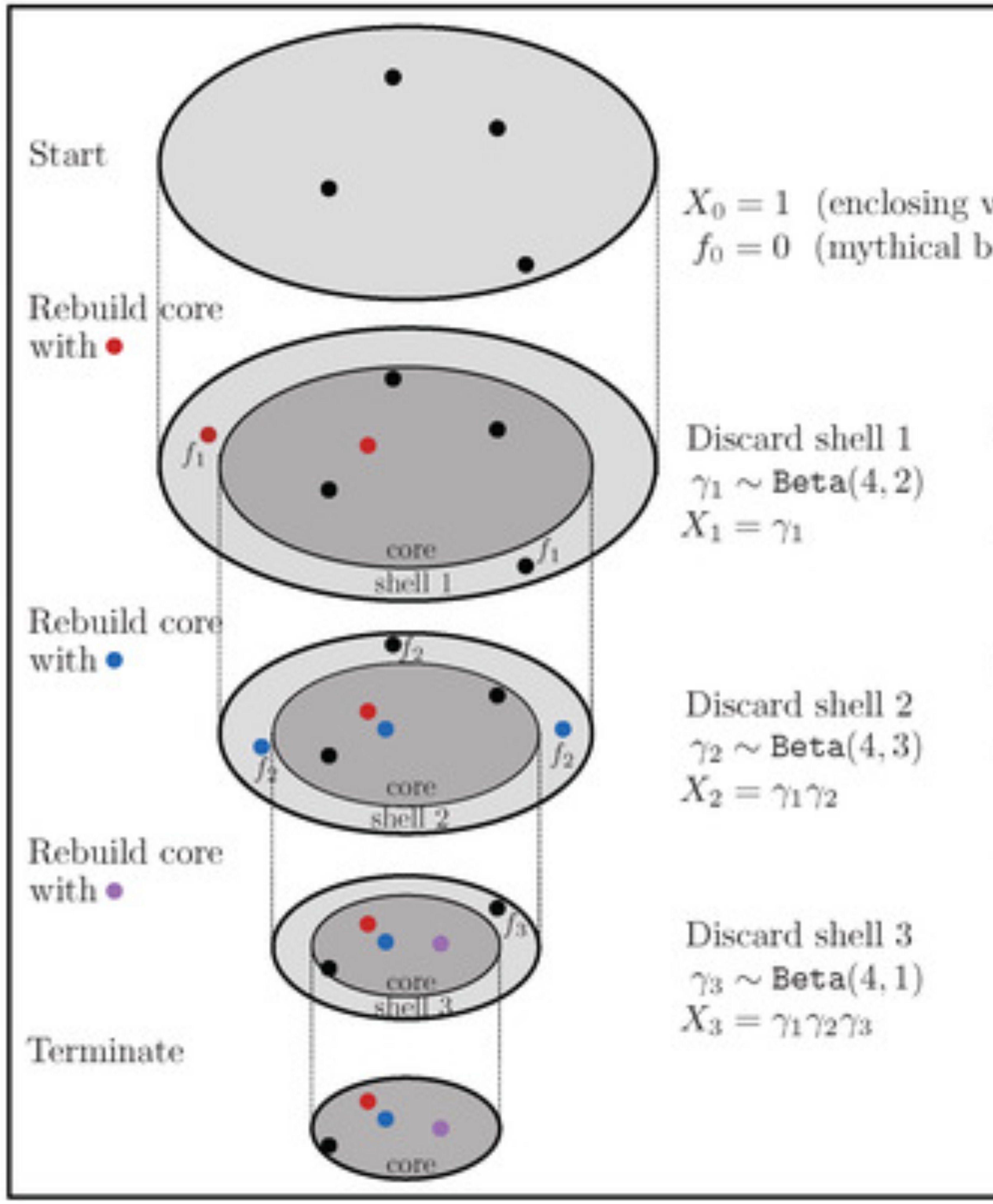


Image Credit: Skilling 2004

Nested Sampling

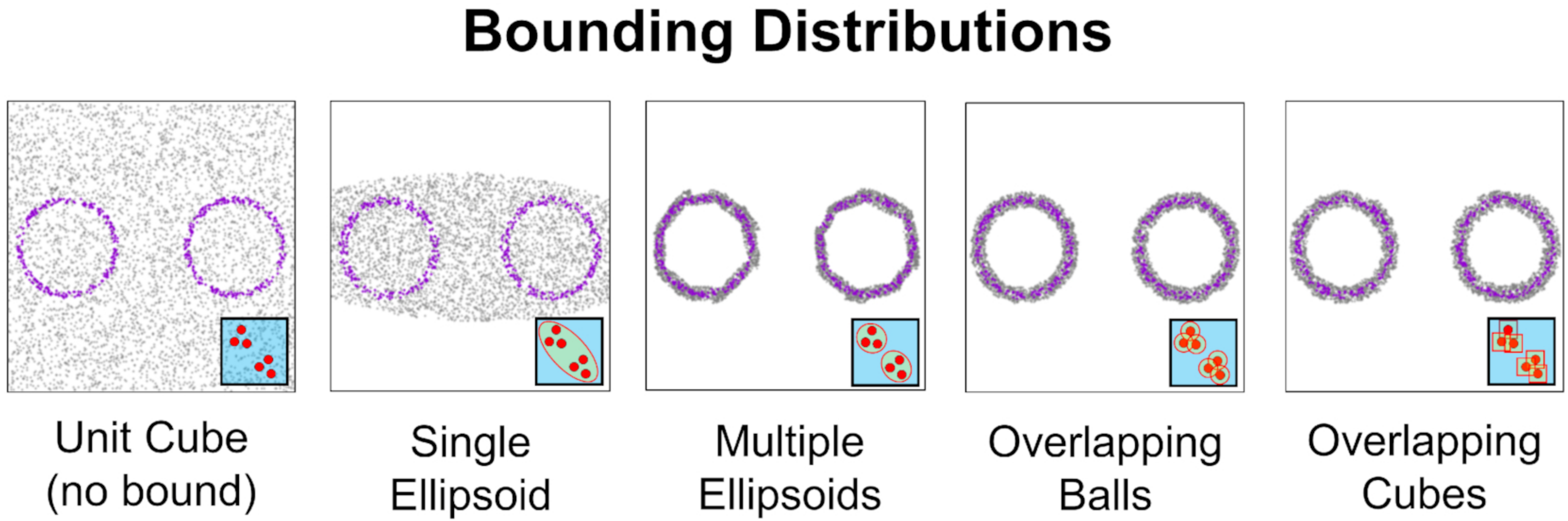


Image Credit: Speagle 2020

Nested Sampling

The evidence Z is the likelihood of the data D marginalized over all possible parameter values θ for a given model M

$$Z = \int P(D \mid \theta)\, P(\theta)\, d\theta$$

Marginal Likelihood

The evidence Z can be re-expressed as an integral over the prior volume X; this volume can be constrained by likelihood contours $\mathscr{L} > \lambda$

$$Z = \int_0^1 \mathscr{L}(X)\, dX$$

Integral over prior volume X

$$X(\lambda) \equiv \int_{\mathscr{L}(\theta) > \lambda} P(\theta) d\theta$$

The evidence Z can be calculated by summing over the likelihood $\mathscr{L}_i$ of each nested sample, weighted by from the contained prior volume

$$Z \approx \sum_{i=1}^{n} \mathscr{L}_i \cdot w_i$$

From weighted samples

$$w_i = X_{i-1} - X_i$$

$$X_i \approx e^{-i/N}$$

Nested Sampling

From the evidence, we can generate samples from the posterior as an automatic byproduct

$$Z \approx \sum_{i=1}^{n} \mathscr{L}_i \cdot w_i \qquad \Rightarrow \qquad P(\theta_i \mid D) \approx \frac{\mathscr{L}_i \cdot w_i}{Z}$$

Recall that weights w_i are proportional to the prior volume X_i contained within likelihood contour $\mathscr{L} > \lambda$

Popular Implementations

- MultiNest | https://github.com/JohannesBuchner/MultiNest
- UltraNest | https://johannesbuchner.github.io/UltraNest/index.html
- DyNesty | https://dynesty.readthedocs.io/en/v2.1.5/

Which sampler should I use?

Random Walk or Metropolis-Hastings
Good for learning the fundamental of MCMC, but you probably won't use these in practice.

Gibbs Sampling
Special cases dictated by the problem geometry.

Affine Invariant of Differential Evolution MCMC
Reasonable default choice for low-to-moderate number of dimensions, provided covariances are not pathological. The python package `emcee` is popular, making affine invariant sampling popular.

Parallel Tempering
Good for complicated or multimodal posteriors. Nicely balances exploration and sampling.

Hamiltonian Monte Carlo
Topology must be differentiable (i.e. you can compute a gradient). Particularly good for high dimensional problems, but struggles with multimodal distributions.

Nested Sampling
Excellent for complicated or multimodal distributions. Performance slows in higher dimensions.

Further reading

Bayesian Analysis

Bayesian Data Analysis, by A. Gelman et al. Third Edition, Boca Raton, FL: Chapman & Hall 2014

Model Fitting

Hogg, Bovy, & Lang 2010, “Data analysis recipes: fitting a model to data”, arXiv:1008.4686

Affine-Invariant MCMC

Foreman-Mackey et al., 2014, “emcee: The MCMC Hammer”, PASP, 125, 925

Hamiltonian Monte Carlo

Betancourt 2017, “A Conceptual Introduction to Hamiltonian Monte Carlo”, arXiv:1701.02434

Nested Sampling

Speagle 2020, “DYNESTY: a dynamic nested sampling package for estimating Bayesian posteriors and evidences” MNRAS, 493, 3