# Lecture Recording

- **Note: These lectures will be recorded and posted onto the IMPRS website**

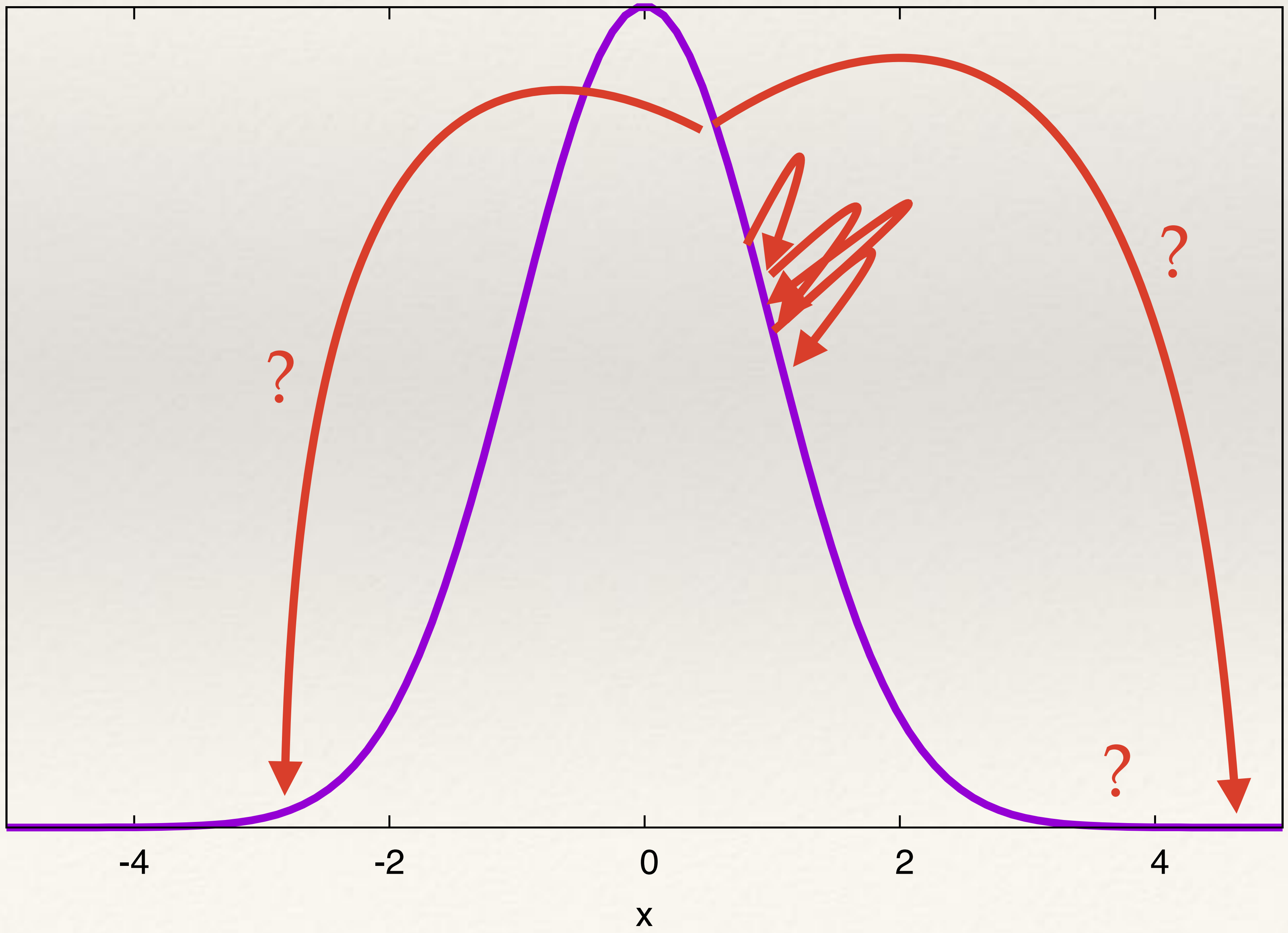
- Dear participants,
- We will record all lectures on "*Making sense of data: introduction to statistics for gravitational wave astronomy*", including possible Q&A after the presentation, and we will make the recordings publicly available on the IMPRS lecture website at:
  - https://imprs-gw-lectures.aei.mpg.de/2021-making-sense-of-data/
- By participating in this Zoom meeting, you are giving your explicit consent to the recording of the lecture and the publication of the recording on the course website.

# Making sense of data: introduction to statistics for gravitational wave astronomy

# Lecture 7: Sampling methods

*AEI IMPRS Lecture Course*

*Jonathan Gair* jgair@aei.mpg.de



# Working with Bayesian Posteriors

- The posterior distribution encodes all information about the parameters of interest after data has been observed. Sometimes these are analytic, but usually not.

- When they are not analytic, they can be approximated by the **Bayesian Central Limit Theorem.** We suppose that $X_1, \ldots, X_n \overset{\text{iid}}{\sim} p(x \mid \boldsymbol{\theta})$ and the prior $p(\boldsymbol{\theta})$ and likelihood $p(x \mid \boldsymbol{\theta})$ are twice differentiable near $\widehat{\boldsymbol{\theta}}_{\text{post}}$, the mode of the posterior distribution. Then, for large *n*,

$$p(\boldsymbol{\theta} \mid \mathbf{x}) \sim \mathrm{N}\left(\widehat{\boldsymbol{\theta}}_{\text{post}}, [I^{\text{post}}(\boldsymbol{\theta}, \mathbf{x})]^{-1}\right)$$

- where

$$I^{\text{post}}(\boldsymbol{\theta}, \mathbf{x}) = -\left[\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} \log p(\boldsymbol{\theta} \mid \mathbf{x})\right]_{\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_{\text{post}}}$$

# Working with Bayesian Posteriors

- As discussed in Lecture 5, the primary application of probability distributions is to compute expectation values of quantities of interest via integration.

- In low numbers of dimensions, such integrals can be computed by **direct evaluation** (numerical integration) on a grid of points.

- In larger numbers of dimensions it is better to use **stochastic (Monte Carlo) sampling**. We draw a set of samples $\{\vec{\theta}_1, \ldots, \vec{\theta}_M\}$ and then approximate

$$\int f(\vec{\theta})p(\vec{\theta}|\mathbf{x})\mathrm{d}\vec{\theta} \approx \frac{1}{M}\sum_{i=1}^{M} f(\vec{\theta}_i)$$

- Monte Carlo integration converges to the true integral asymptotically as *the number of samples M* tends to infinity, which can always be achieved with sufficient computational power, whereas the Central Limit Theorem relies on the number of *observations* to tend to infinity, which is much harder to ensure in practice.

- Samples can be obtained through **direct sampling** or **Markov Chain Monte Carlo**.

# Direct sampling: Method of Inversion

- If the posterior distribution has a cumulative density function (CDF) with a known inverse, samples can be generated by drawing samples from *U[0,1]*.

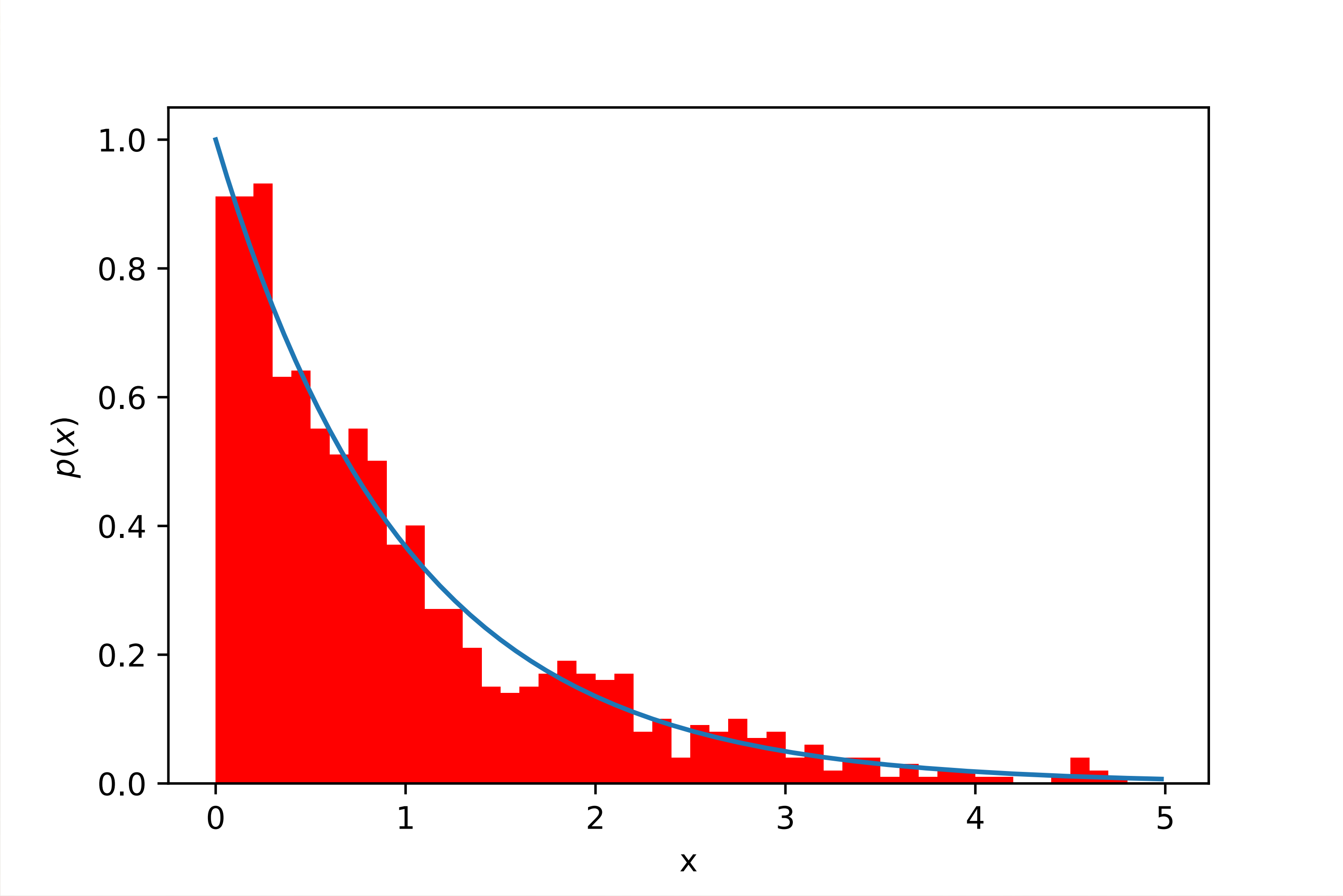
- If the CDF is denoted by *F*

$$F(\Theta|\mathbf{x}) = \mathbb{P}(\theta \leq \Theta|\mathbf{x})$$

- We simulate

$$u_i \sim U[0,1]$$
$$\theta_i = F^{-1}(u_i|\mathbf{x})$$

- The $x_i$ are samples from *f*.



**Example**: exponential with parameter *r*.
*p(t | r) = r* exp(-*rt*), *F(T)* = *1*-exp(-*rT*),
*F*$^{-1}$*(u) = ln(1/(1-u))/r.*

# Direct sampling: Rejection Sampling

- **Rejection sampling** uses samples drawn from another distribution that "contains" the distribution of interest. The algorithm is

$$\begin{aligned} \theta_i &\sim g(\theta) \\ y_i &\sim U[0, Mg(\theta)] \end{aligned}$$

$$\text{If } y_i \leq p(\theta_i|\mathbf{x}), \text{ accept } \theta_i$$

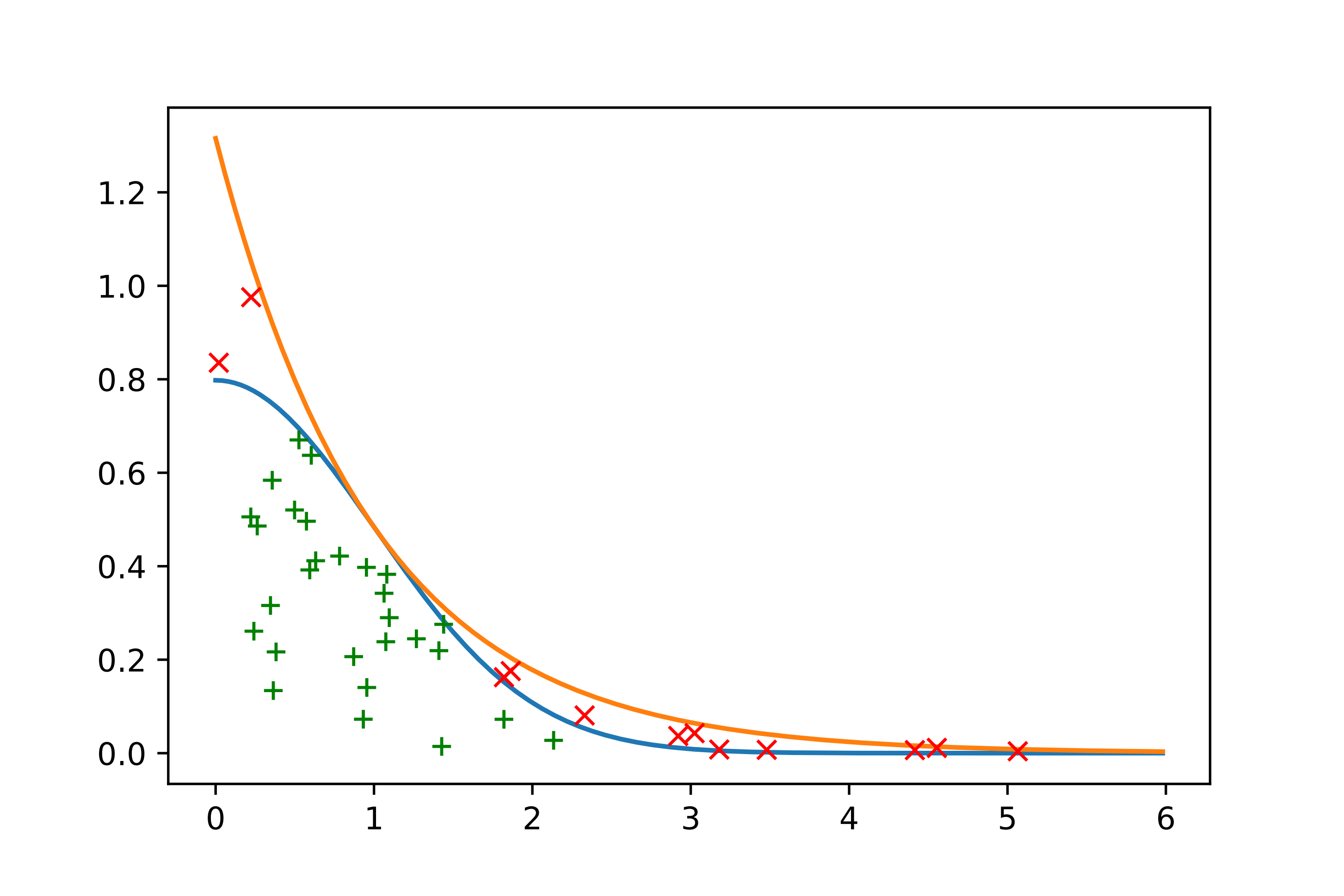
$$\text{as a sample from } p(\theta|\mathbf{x})$$

- We require

$$Mg(\theta) \geq p(\theta|\mathbf{x}) \ \ \forall\, \theta$$

- The "best" rejection method uses

$$M = \sup_{\theta} \left( \frac{p(\theta|\mathbf{x})}{g(\theta)} \right)$$



**Example: half-Normal distribution**. We want to sample from *N(0,1) I(x > 0)*. We draw samples from *Exp(1)*, for which we need *M = 1.3155*.

# Direct sampling: Importance Sampling

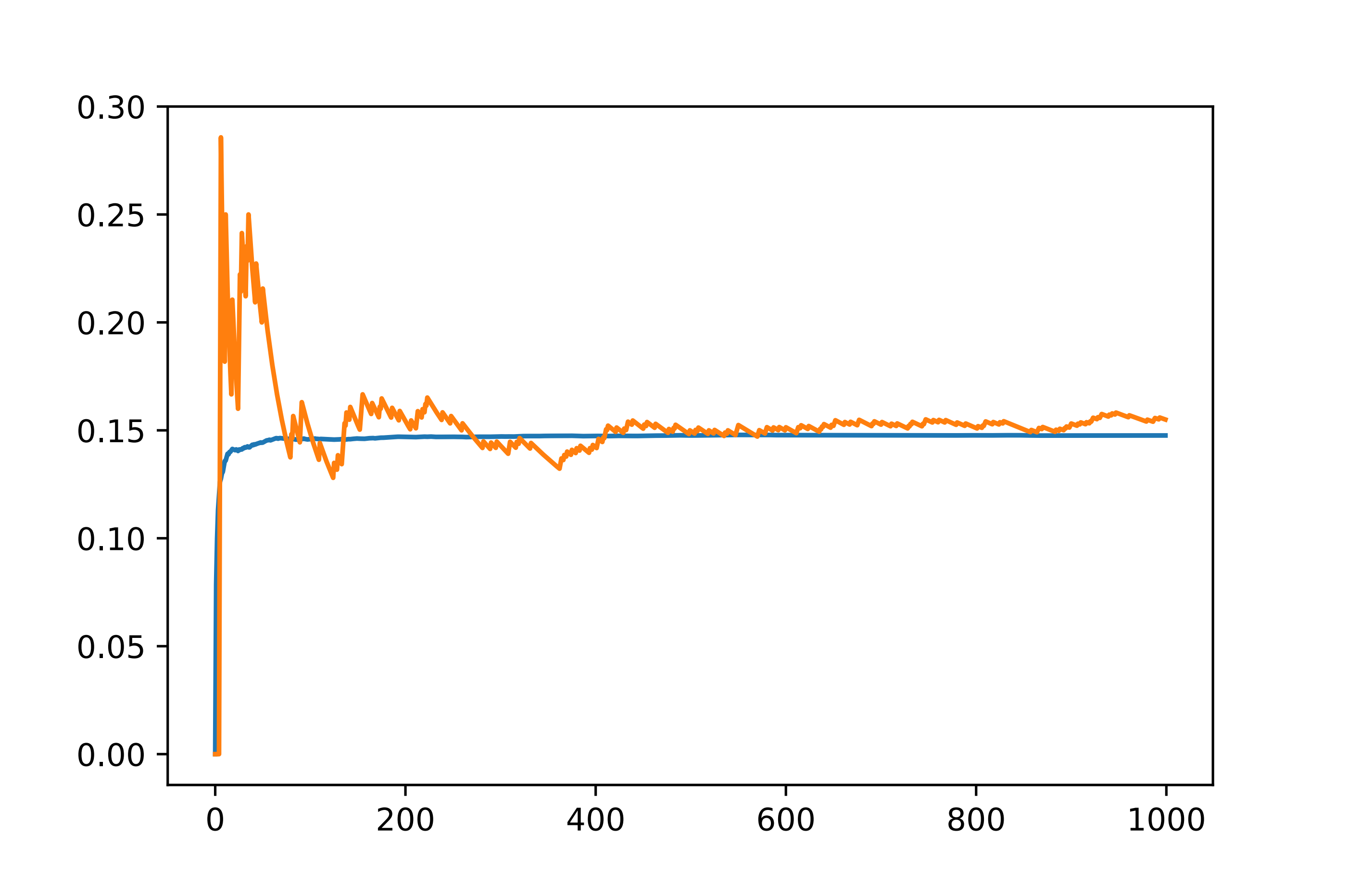
- **Importance sampling** also draws samples from another, easy-to-sample distribution, but now samples are not rejected but given weights

$$w_i = \frac{p(\theta_i|\mathbf{x})}{g(\theta_i)}$$

- Integrals over the posterior are approximated by weighted sums

$$\int f(\theta)p(\theta|\mathbf{x})\ \mathrm{d}\theta \approx \frac{1}{N}\sum_{i=1}^{N} w_i f(\theta_i)$$

- One advantage is that the normalisation of the posterior does not need to be known. But, the algorithm suffers from high sampling variance.



**Example: Cauchy distribution**. We want samples from $p(\theta) = 1/(\pi(1+\theta^2))$. We draw samples from $g(\theta) = 2/\theta^2$ and use importance sampling to estimate $\mathbb{P}(\theta > 2)$.

# Direct sampling: Sampling importance Resampling

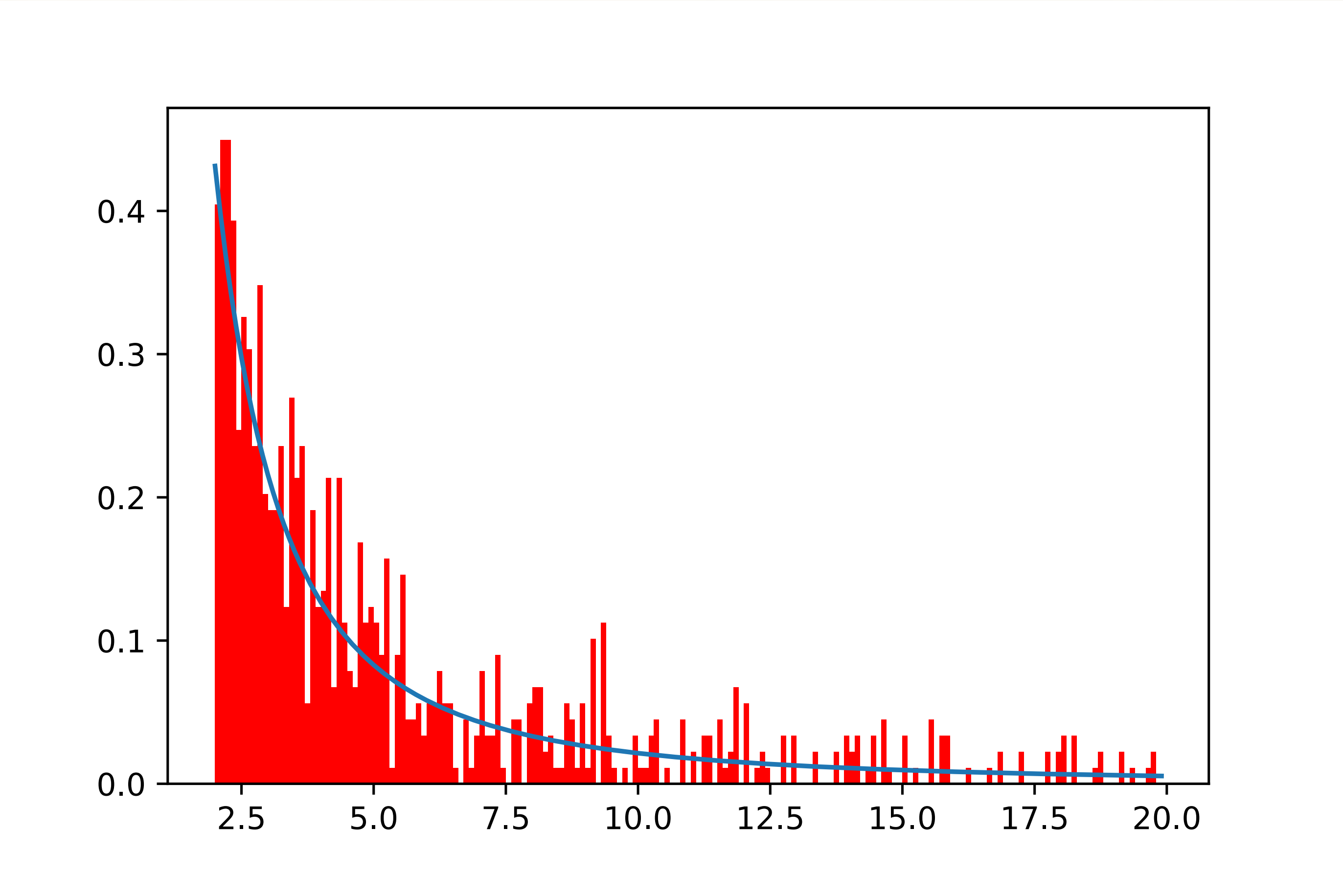
- **Sampling importance resampling** is a variant of importance sampling.

- Importance samples are first drawn using the algorithm on the previous slide and the weights renormalised

$$w_i = \frac{w_i}{\sum_{j=1}^{n} w_j}$$

- New samples are then drawn at random, with replacement, from the first set, with relative probabilities given by $w_{i.}$

- This is a form of **particle filtering**. It can suffer from **particle depletion**, when a small number of samples carry the majority of the weight.



**Example**: histogram of resampled points from first 1000 importance samples from previous slide.

# Markov Chain Monte Carlo

- Often direct sampling methods cannot be devised, because the target distribution is too complicated. In those cases, stochastic methods can be used based on Markov Chain Monte Carlo methods.

- The idea is to generate a reversible Markov chain (i.e., a sequence such that each element depends only on the previous one and not longer past history), with a stationary distribution that equals the target distribution.

- Such a Markov chain must satisfy *detailed balance*

$$p(\vec{\theta})\, p(\vec{\theta}, \vec{\theta}') = p(\vec{\theta}')\, p(\vec{\theta}', \vec{\theta})$$

- In which

$$p(\vec{\theta}, \vec{\theta}') = p(\vec{\theta}_i = \vec{\theta}' | \vec{\theta}_{i-1} = \vec{\theta})$$

- and $p(\vec{\theta})$ denotes the target distribution, in our case $p(\vec{\theta}|d, M)$ .

# Gibbs Sampling

- **Gibbs sampling** draws consecutive samples from the full conditional distributions. It relies on the conditionals taking known forms. The algorithm is as follows
  - Initialise the parameters at some starting values

$$\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$$

  - For *s = 1, …, S:*
    - Draw $\theta_1^{(s)} \sim p(\theta_1 \mid \theta_2^{(s-1)}, \theta_3^{(s-1)}, \ldots, \theta_p^{(s-1)}, \mathbf{x})$
    - Draw $\theta_2^{(s)} \sim p(\theta_2 \mid \theta_1^{(s)}, \theta_3^{(s-1)}, \ldots, \theta_p^{(s-1)}, \mathbf{x})$
    - ….
    - Draw $\theta_p^{(s)} \sim p(\theta_p \mid \theta_1^{(s)}, \theta_2^{(s)}, \ldots, \theta_{p-1}^{(s)}, \mathbf{x})$
- For sufficiently large *s* $(\theta_1^{(s)}, \ldots, \theta_p^{(s)}) \overset{\text{approx.}}{\sim} p(\theta_1, \ldots, \theta_p \mid \mathbf{x})$

# Metropolis Hastings Algorithm

- Gibbs sampling relies on being able to define the full conditional distributions. When this is not possible, the Metropolis-Hastings algorithm provides another way to compute a suitable Markov chain.

- We initialise by choosing a (random) starting point. Then, at step i:

  - propose a new point, $\vec{\theta}'$, by drawing from a *proposal distribution*, $q(\vec{\theta}', \vec{\theta}_i)$ .

  - evaluate the target distribution at the new point. Compute the *Metropolis-Hastings ratio*

$$\mathcal{H} = \frac{p(\vec{\theta}')q(\vec{\theta}_i, \vec{\theta}')}{p(\vec{\theta}_i)q(\vec{\theta}', \vec{\theta}_i)}$$

  - and draw a random sample, $\alpha$, from a U[0,1] distribution. If $\alpha < \mathcal{H}$ then set $\vec{\theta}_{i+1} = \vec{\theta}'$, otherwise set $\vec{\theta}_{i+1} = \vec{\theta}_i$ . NB if $\mathcal{H} > 1$the proposed move is definitely accepted.

# Proposal Distributions

- Sampling efficiency is strongly influenced by the choice of proposal distribution.

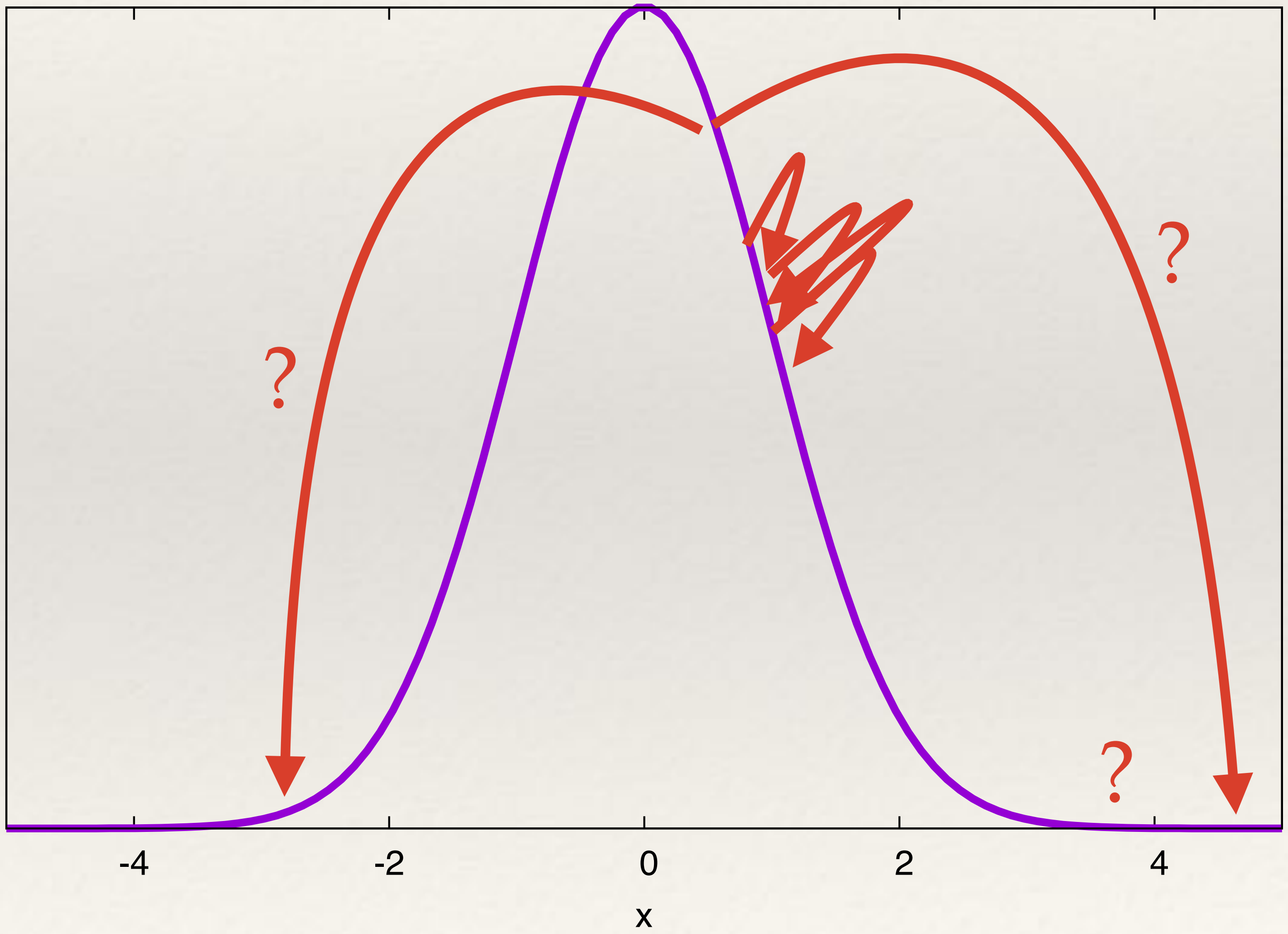
- Uniform proposal (random sampling) very inefficient - better to use a grid.

- Ideally want a proposal tuned to the distribution you are sampling.

- A Gaussian is often a good choice, but need to tune width.

    - **too wide:** low acceptance rate;

    - **too narrow:** high acceptance rate; low effective samples.



# Annealing

- One way to accelerate convergence is to use *simulated annealing*.

- "Heat up" posterior by making the replacement

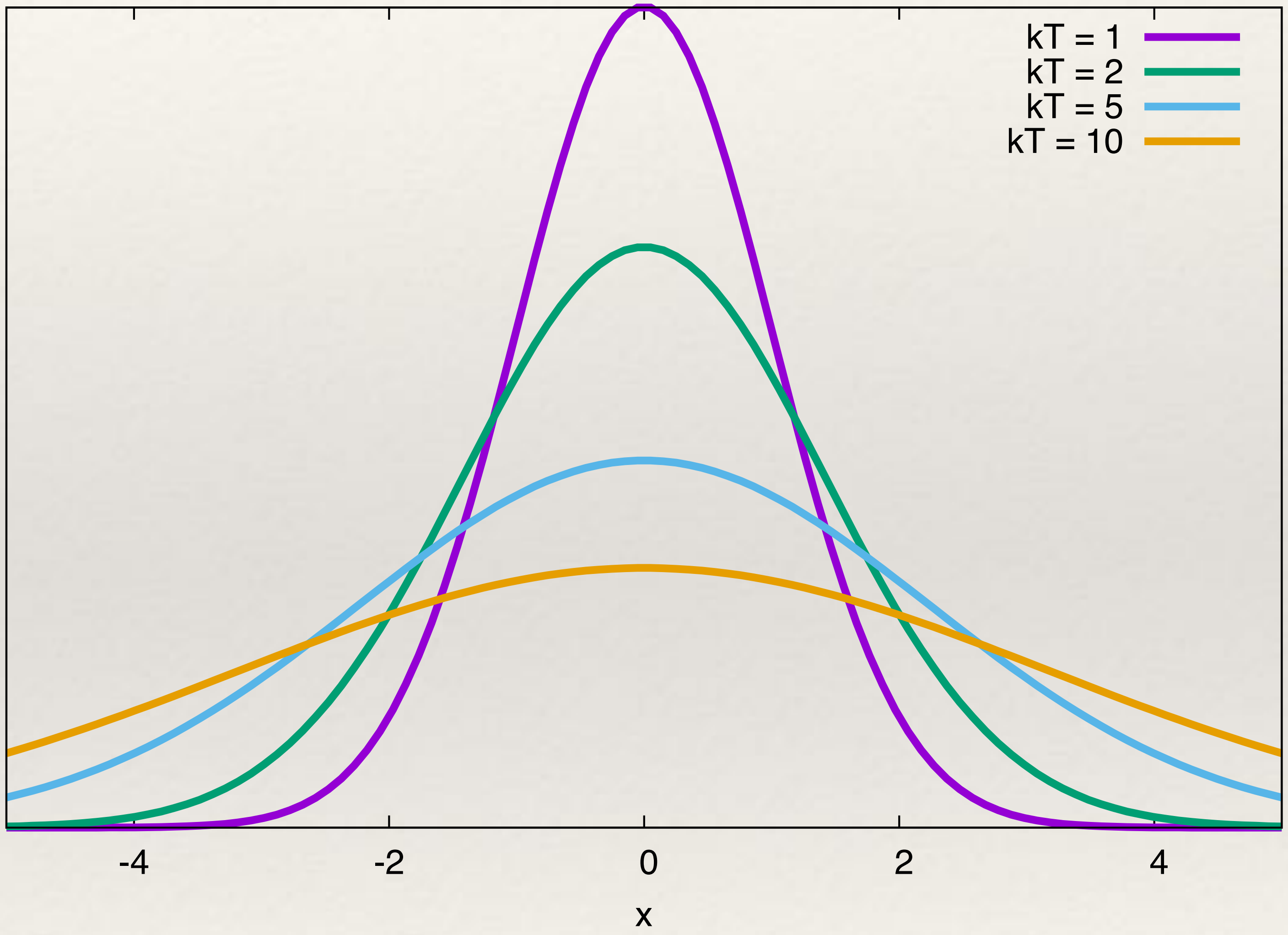
$$p(\vec{\theta}|d, M) \rightarrow \left[p(\vec{\theta}|d, M)\right]^{\beta}$$

- where

$$\beta = \frac{1}{kT}$$

- Choosing a high temperature smoothes out the posterior which can then be more easily sampled.

- Allows identification of interesting parts of parameter space.



# Annealing

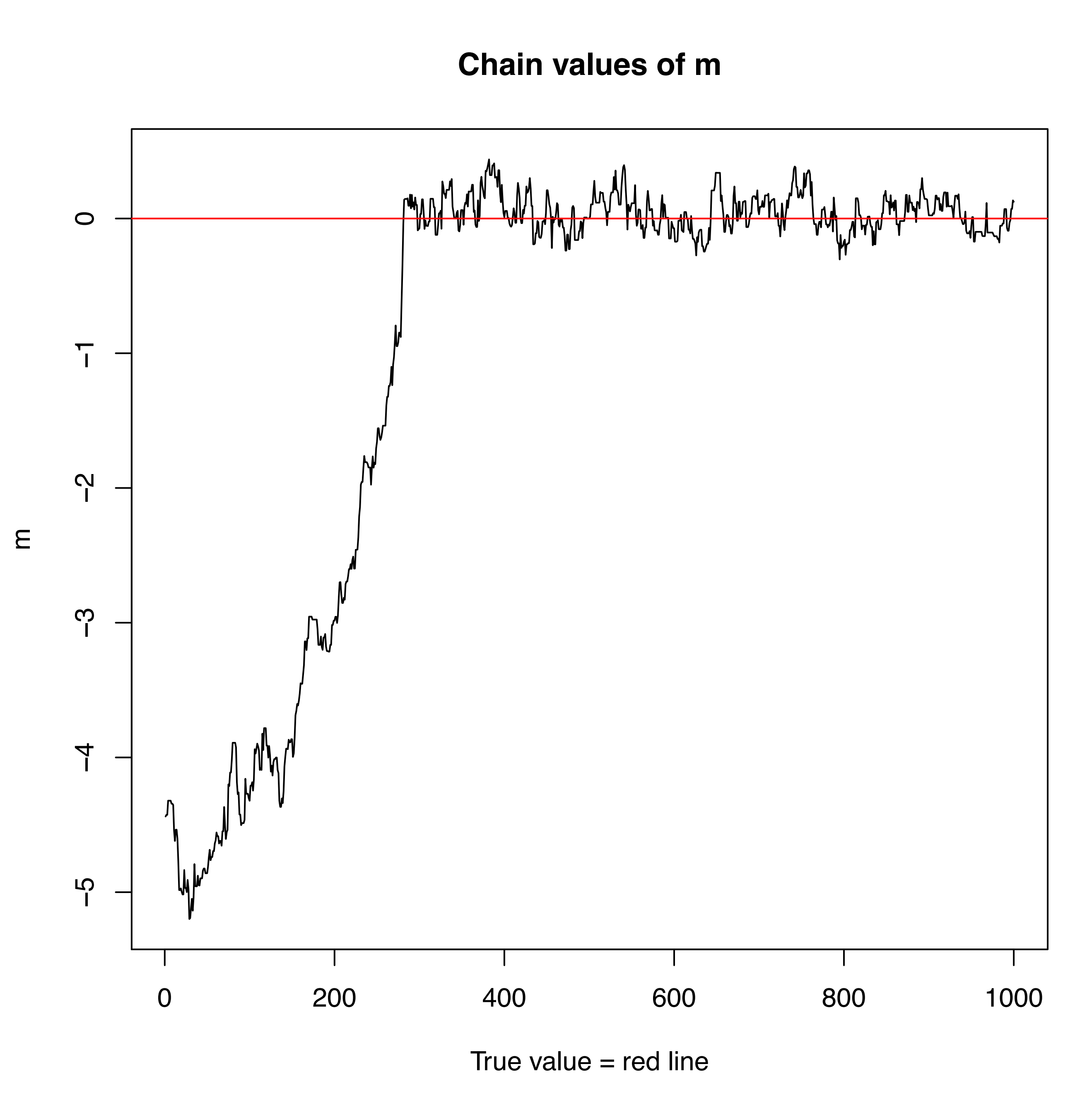
- It is common to use *parallel tempering*. A sequence of M MCMC chains are run simultaneously at different temperatures, $\{T_1, \ldots, T_M\}$.
- The chains can exchange information, which is achieved by proposing a swap of the states of two chains with different temperatures. The swap is accepted with probability

$$\min\left(1, \frac{p_i(\vec{\theta}_j)\, p_j(\vec{\theta}_i)}{p_i(\vec{\theta}_i)\, p_j(\vec{\theta}_j)}\right)$$

- where i, j label the two temperature chains, $\vec{\theta}_k$ denotes the current state of the k'th chain and $p_k(\vec{\theta})$ denotes the target (annealed) distribution for the k'th chain.

# Burn-in

- The MCMC chain does not sample from the target distribution immediately.
- There is a residual "memory" of the initial state. Need to discard the first few samples.
- This is called the **burn-in**.
- Can identify number of samples to discard by looking at *trace plots*.
- Usually a few hundred to a thousand samples is sufficient for burn-in.



# Autocorrelation and Effective sample size

- Consecutive samples in the MCMC chain are not independent samples from the target distribution.

- Can use all samples for posterior inference *but* do need to know how many *independent* samples the chain contains in order to assess the precision of inferences.

- Compute the (lag-k) autocorrelation

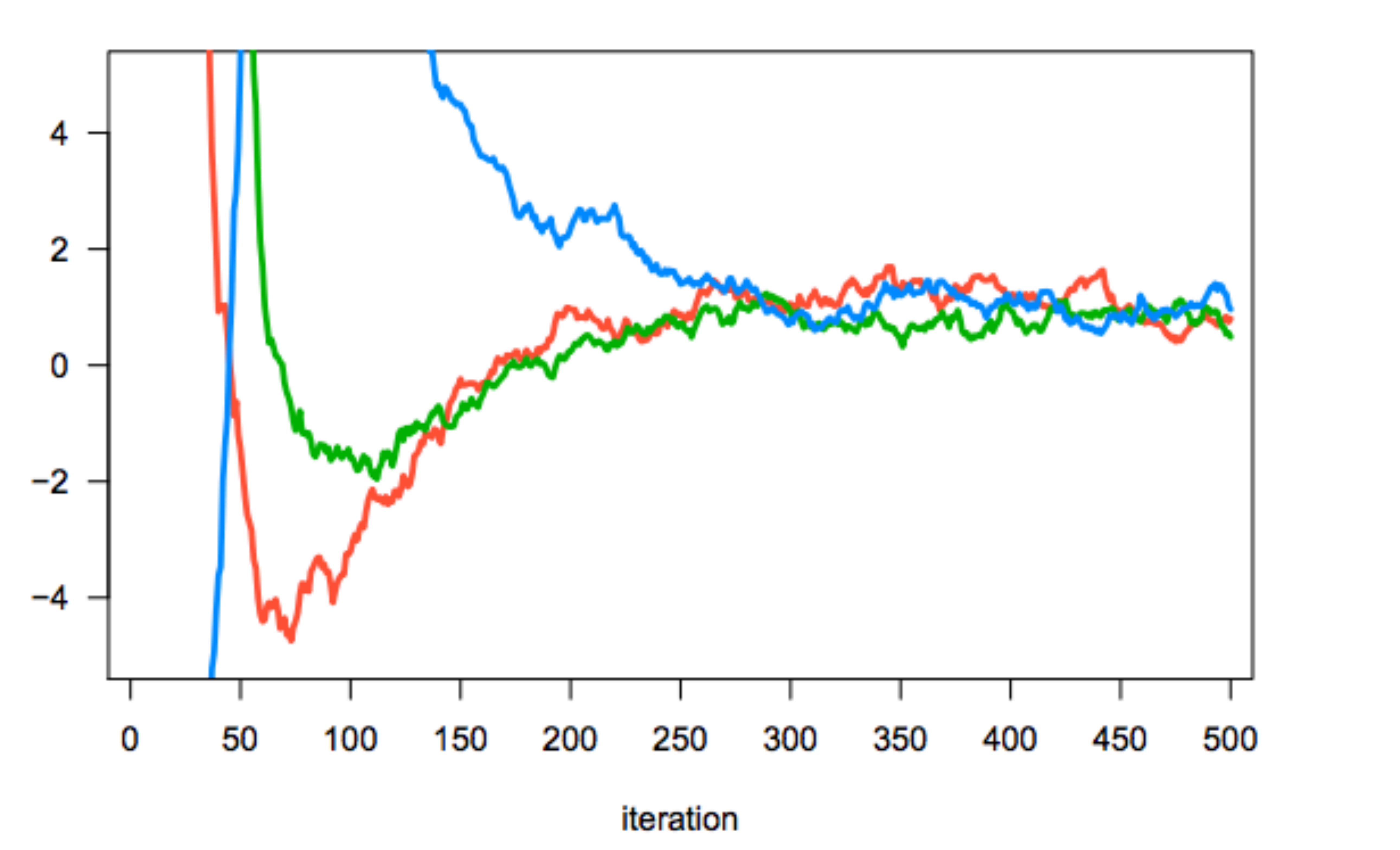
$$\rho_k = \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

- where x now denotes one of the components of $\vec{\theta}$. Choose *k*=*K* large enough that the autocorrelation $\rho_k << 1$. The **effective sample size** is $\sim N/K$ and formally defined

$$\mathrm{ESS} = \frac{N}{1 + 2\sum_{i=1}^{\infty}\rho_k}$$

- Can "thin" chain by keeping only every *K*'th sample without affecting accuracy of posterior inference.

# Diagnostics



# Gelman-Rubin convergence diagnostic

- Run *m* (at least 2) chains and discard first half of samples from each.
- Calculate the within-chain variance

$$W = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{N-1}\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)^2$$

- Calculate the between-chain variance

$$B = \frac{N}{m-1}\sum_{j=1}^{m}(\bar{x}_j-\bar{\bar{x}})^2, \qquad \bar{\bar{x}} = \frac{1}{m}\sum_{j=1}^{m}\bar{x}_j$$

- Calculate the estimated variance of a given parameter

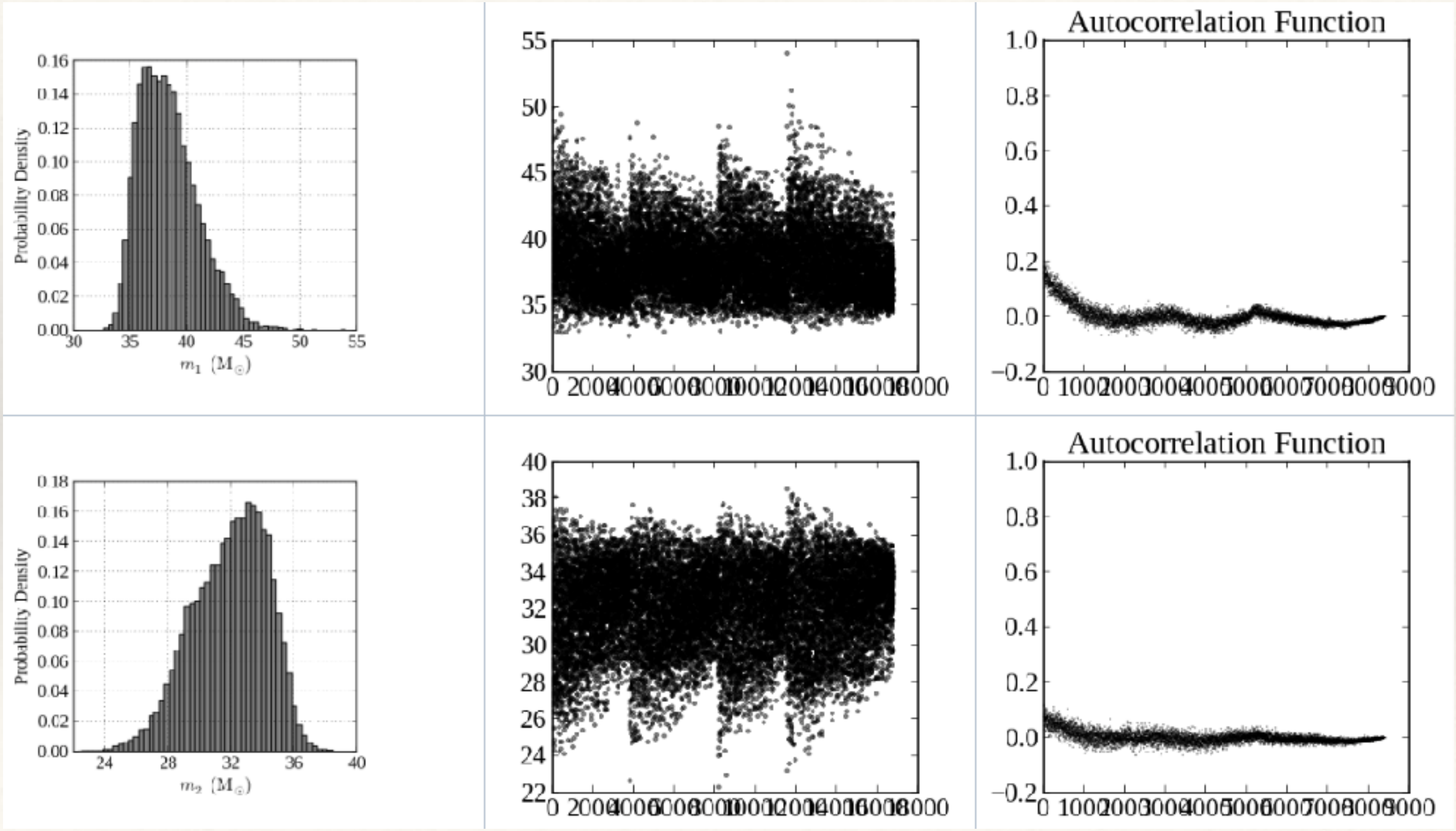
$$\mathrm{Var}(x) = \left(1-\frac{1}{N}\right)W + \frac{1}{N}B$$

- Calculate the potential scale-reduction factor

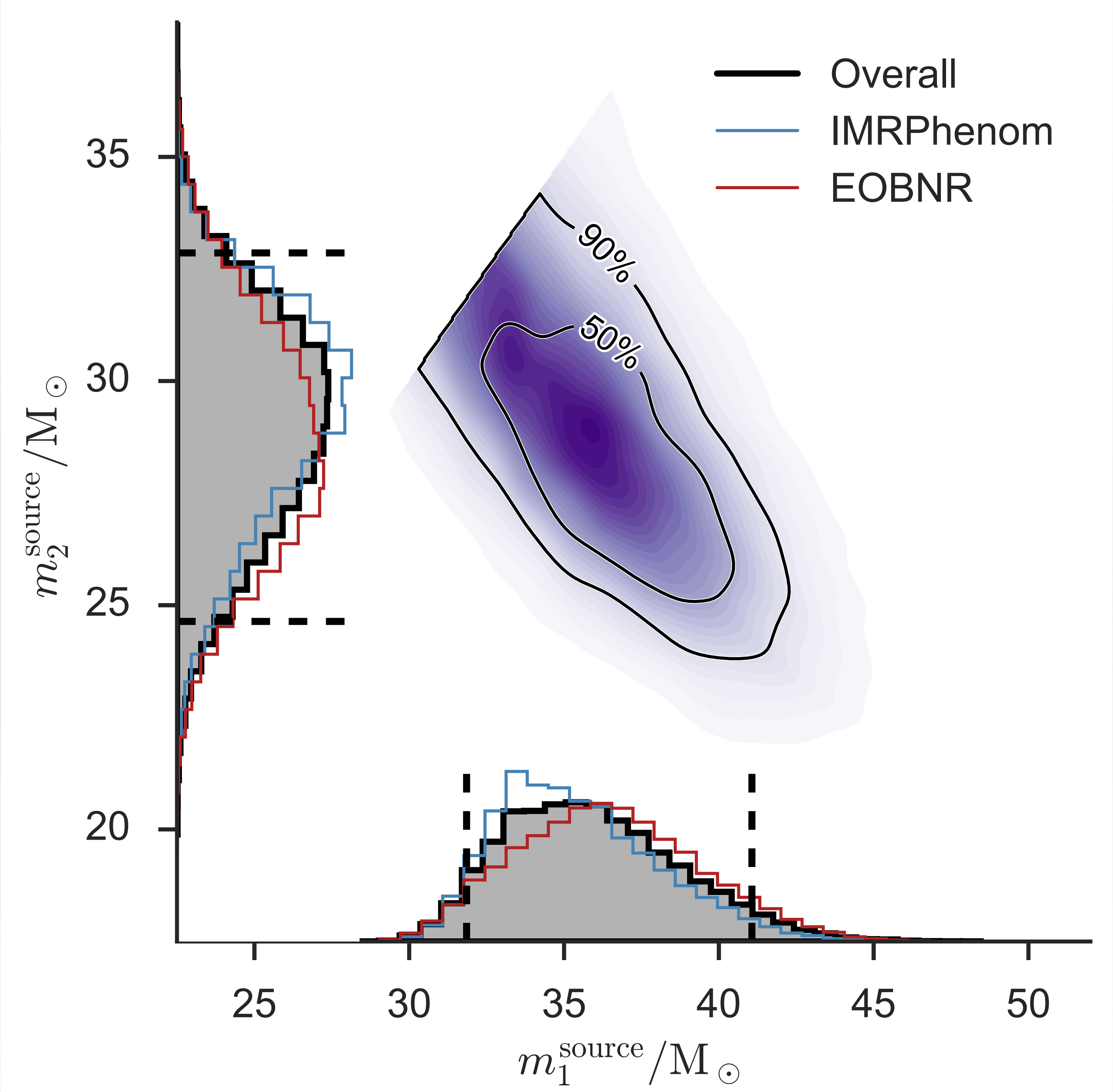
$$\hat{R} = \sqrt{\frac{\hat{\mathrm{Var}}(x)}{W}}$$

- If R is greater than ~1.1 or 1.2, need to run chains for longer.

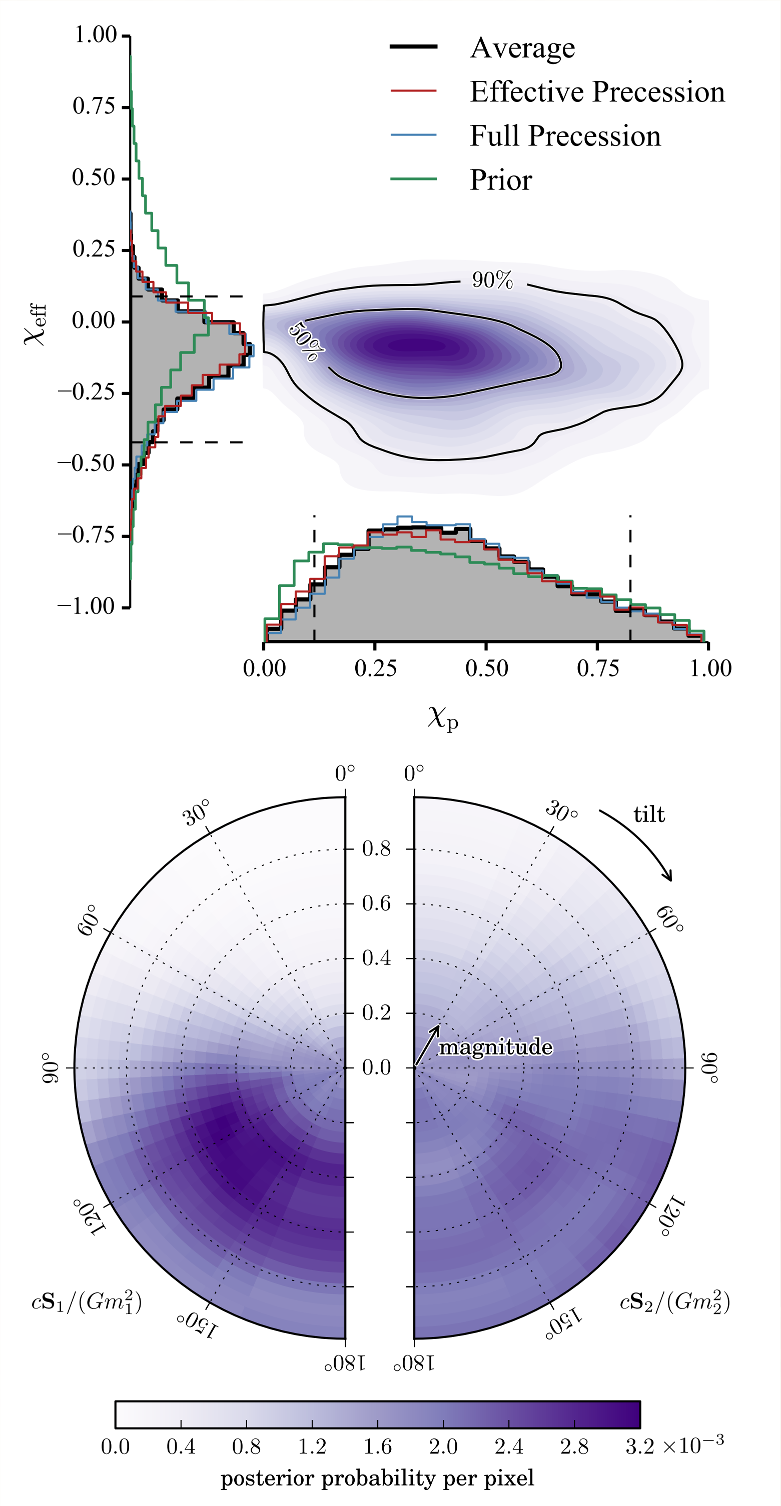
# Convergence diagnostics: GW150914



# Examples of Parameter Posteriors



LVC, *Phys. Rev. Lett.* **116**, 061102 (2016)



# Reversible Jump MCMC

- Often the number of sources in the data set is also unknown.

- **Reversible Jump Markov Chain Monte Carlo** is a technique applied in such situations, by periodically proposing jumps **between models**. In GW applications these normally correspond to different numbers of events.

- Represent a proposed move by tuples *(x, u)* and *(x′, u′)*. Here **x** and **x′** denote the parameters of the current and proposed state (which may have different numbers of dimensions) and **u**, **u′** are sets of random numbers that lead to a proposed move from **x** to **x′** and back.

- Generalisation of acceptance ratio is

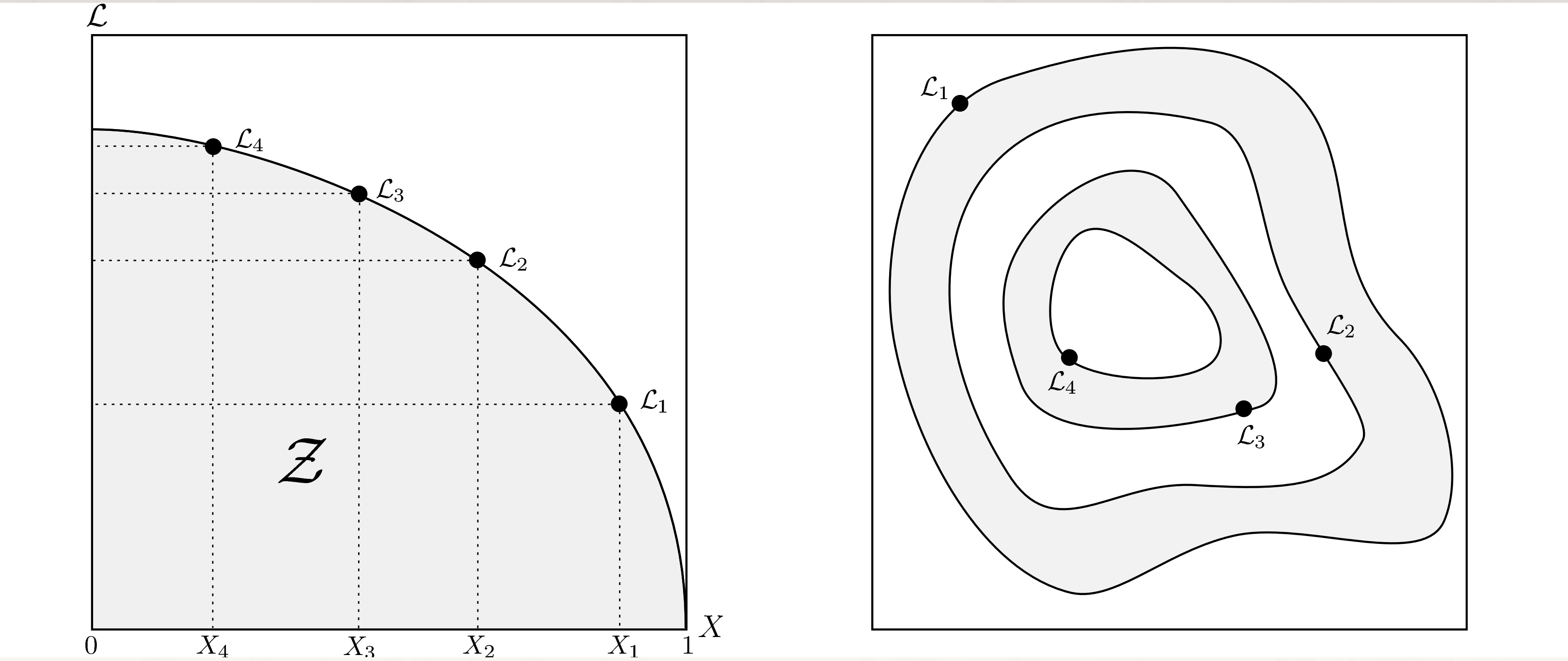
$$\alpha = \min\left(1, \frac{p(\mathbf{x}')q(\mathbf{u}')}{p(\mathbf{x})q(\mathbf{u})}\left|\frac{\partial(\mathbf{x}',\mathbf{u}')}{\partial(\mathbf{x},\mathbf{u})}\right|\right)$$

# Nested Sampling

- Nested Sampling (Skilling 04) provides an efficient way to compute evidences, using a 1D integral over the prior

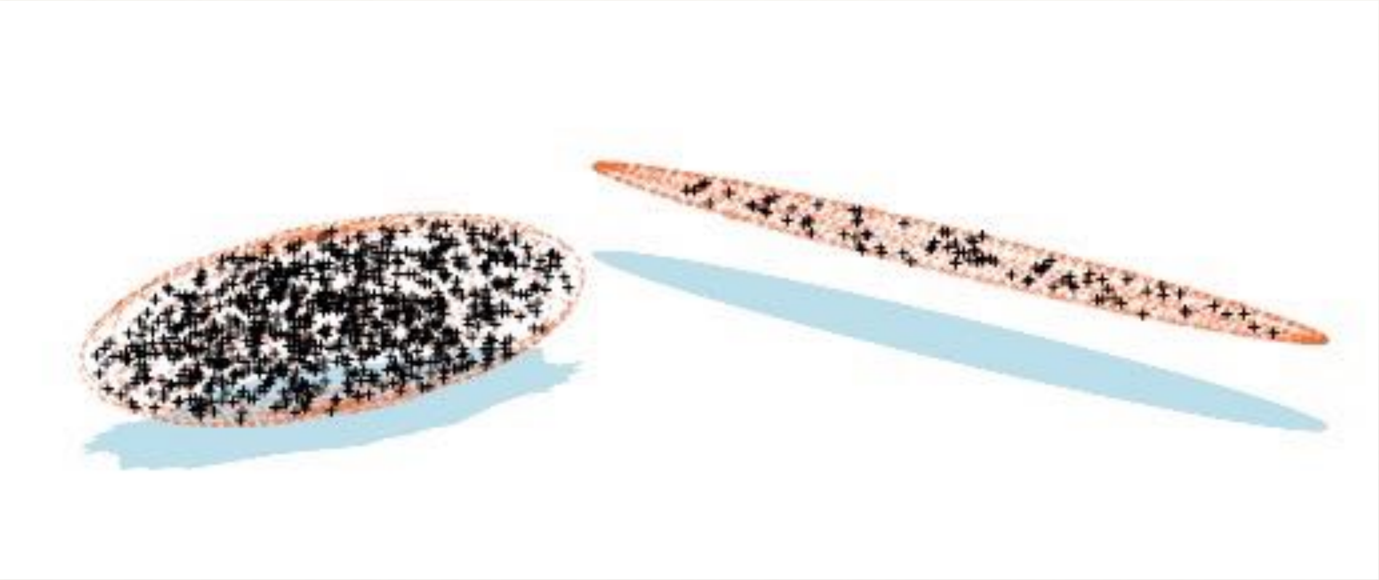
$$\mathcal{Z} = \int \mathcal{L}(\boldsymbol{\Theta})\pi(\boldsymbol{\Theta})\mathrm{d}^N\boldsymbol{\Theta} = \int_0^1 \mathcal{L}(X)\mathrm{d}X, \ \text{where } X(\lambda) = \int_{\mathcal{L}(\boldsymbol{\Theta})>\lambda} \pi(\boldsymbol{\Theta})\mathrm{d}^N\boldsymbol{\Theta}$$

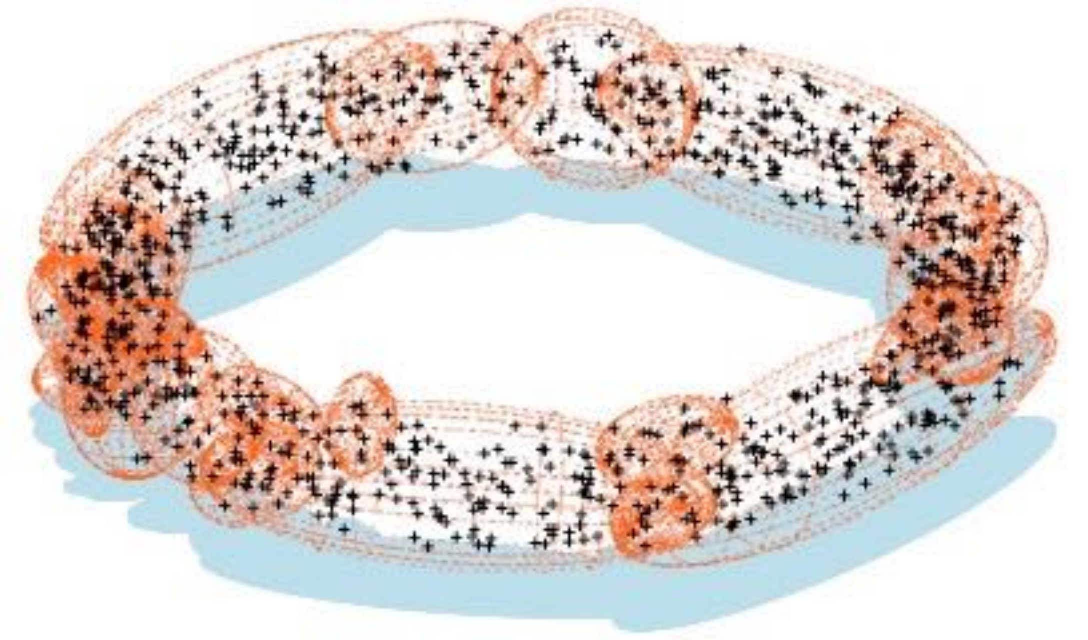
- Use N 'live points', initially chosen at random from the prior. At step i, the point of lowest likelihood, $\mathcal{L}_i$, is replaced by a new point with likelihood $\mathcal{L} > \mathcal{L}_i$ . The prior volume is reduced by a factor *t*, drawn from $p(t) = Nt^{N-1}$, at each step. We climb through nested contours of increasing likelihood as the algorithm proceeds.



# MultiNest

- The trick is to sample efficiently from the prior within the hard constraint that $\mathcal{L} > \mathcal{L}_i$ . MultiNest achieves this using an ellipsoidal rejection sampling scheme. The live point set is partitioned into a number of (possibly overlapping) ellipsoids.

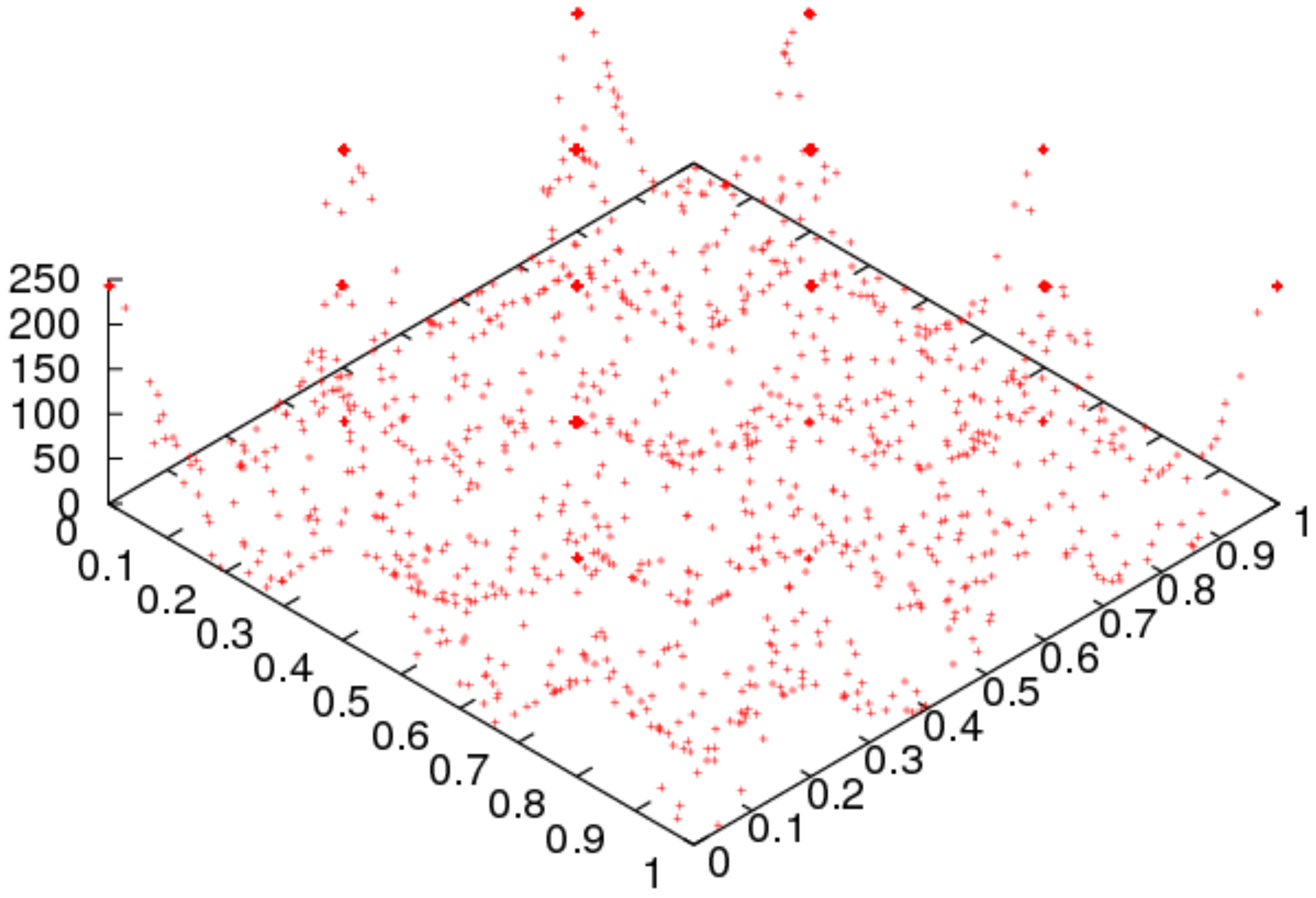




- The algorithm is well suited to exploring likelihoods with multiple modes. Other algorithms (e.g., *cpnest*) update live points using short MCMC explorations.

- Although designed to compute evidences, nested sampling algorithms also return the posterior probability distribution.

# MultiNest


250
200
150
100
50
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1