Introduction to Bayesian Inference & MCMC in RevBayes

@trayc7

phylo@iastate.edu

phyloworks.org

TRACY HEATH

Ecology, Evolution, & Organismal Biology
Iowa State University

Video Lectures

Paul Lewis's Primer on Phylogenetics

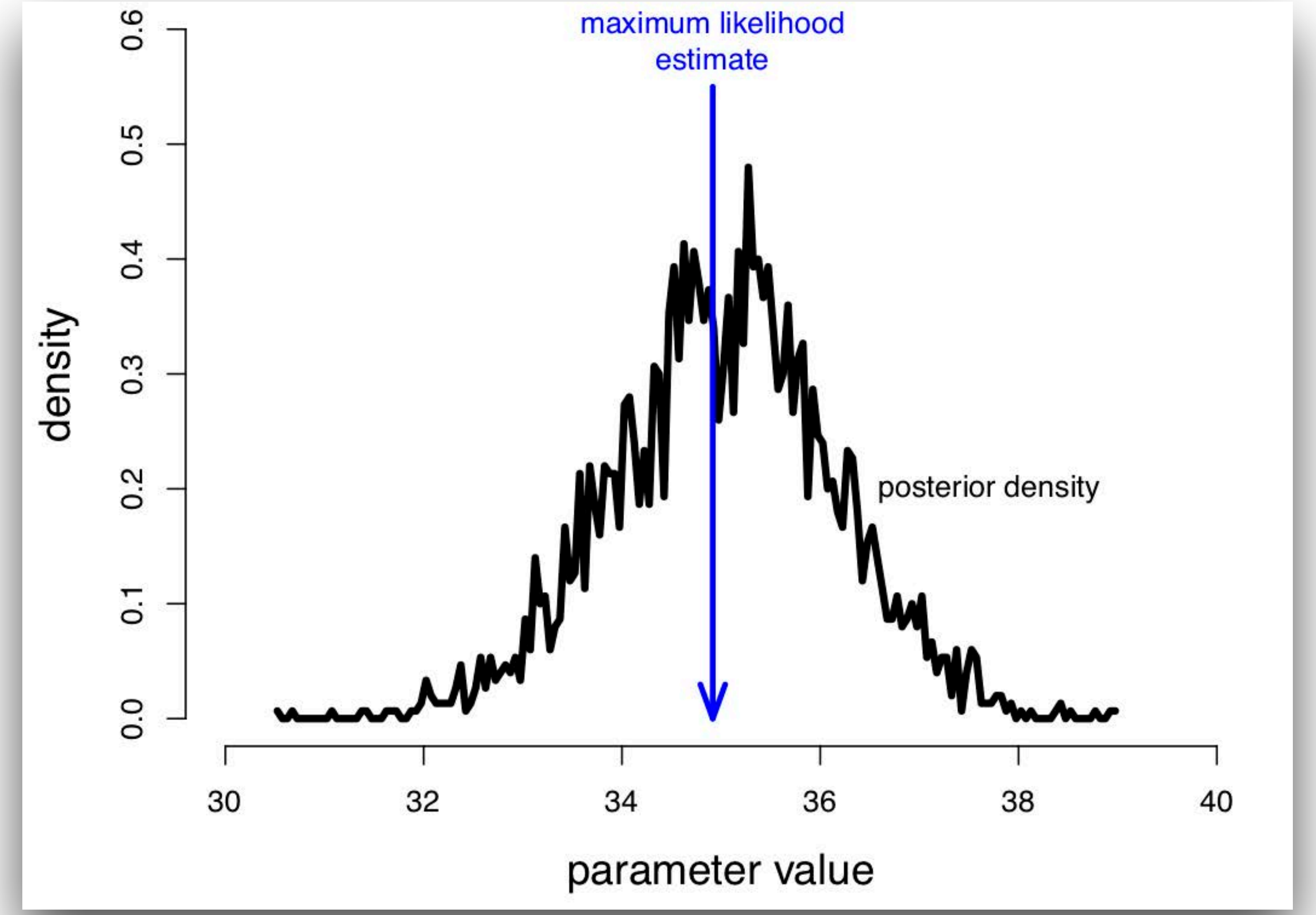
- Trees & Likelihood
- Substitution Models
- Bayesian Statistics & MCMC
- Bayesian Phylogenetics

phyloseminar.org
a free online seminar about phylogenetics

Bayesian or Maximum Likelihood

- estimates $\Pr(\theta \mid \mathbf{X})$
- estimates a **distribution**
- parameters are random variables
- average over nuisance parameters

Bayesian



- estimates $\Pr(\mathbf{X} \mid \theta)$
- **point** estimate
- parameters are fixed/ unknown
- optimize nuisance parameters

Maximum Likelihood

Bayes Rule

$$\underbrace{\Pr(\theta \mid D)}_{\text{POSTERIOR PROBABILITY}} = \frac{\overbrace{\Pr(D \mid \theta)}^{\text{LIKELIHOOD}}\ \overbrace{\Pr(\theta)}^{\text{PRIOR PROBABILITY}}}{\underbrace{\sum_{\theta} \Pr(D \mid \theta) \Pr(\theta)}_{\text{MARGINAL PROBABILITY OF THE DATA}}}$$

Bayesian Inference

Estimate the probability of a hypothesis (model) conditional on observed data

The probability represents a **researcher's degree of belief**

Bayes Rule (also called Bayes Theorem) specifies the conditional probability of the hypothesis given the data

Bayes Rule

the posterior probability of a discrete parameter δ conditional on the data D is

$$\Pr(\delta \mid D) = \frac{\Pr(D \mid \delta)\,\Pr(\delta)}{\sum_{\delta} \Pr(D \mid \delta)\,\Pr(\delta)}$$

the likelihood marginalized over all
possible values of δ

Bayes Rule

the posterior probability of a discrete parameter $\boldsymbol{\theta}$ conditional on the data D is

$$f(\theta \mid D) = \frac{f(D \mid \theta) f(\theta)}{\boxed{\int_\theta f(D \mid \theta) f(\theta)}}$$

the likelihood marginalized over all possible values of $\boldsymbol{\theta}$

Priors

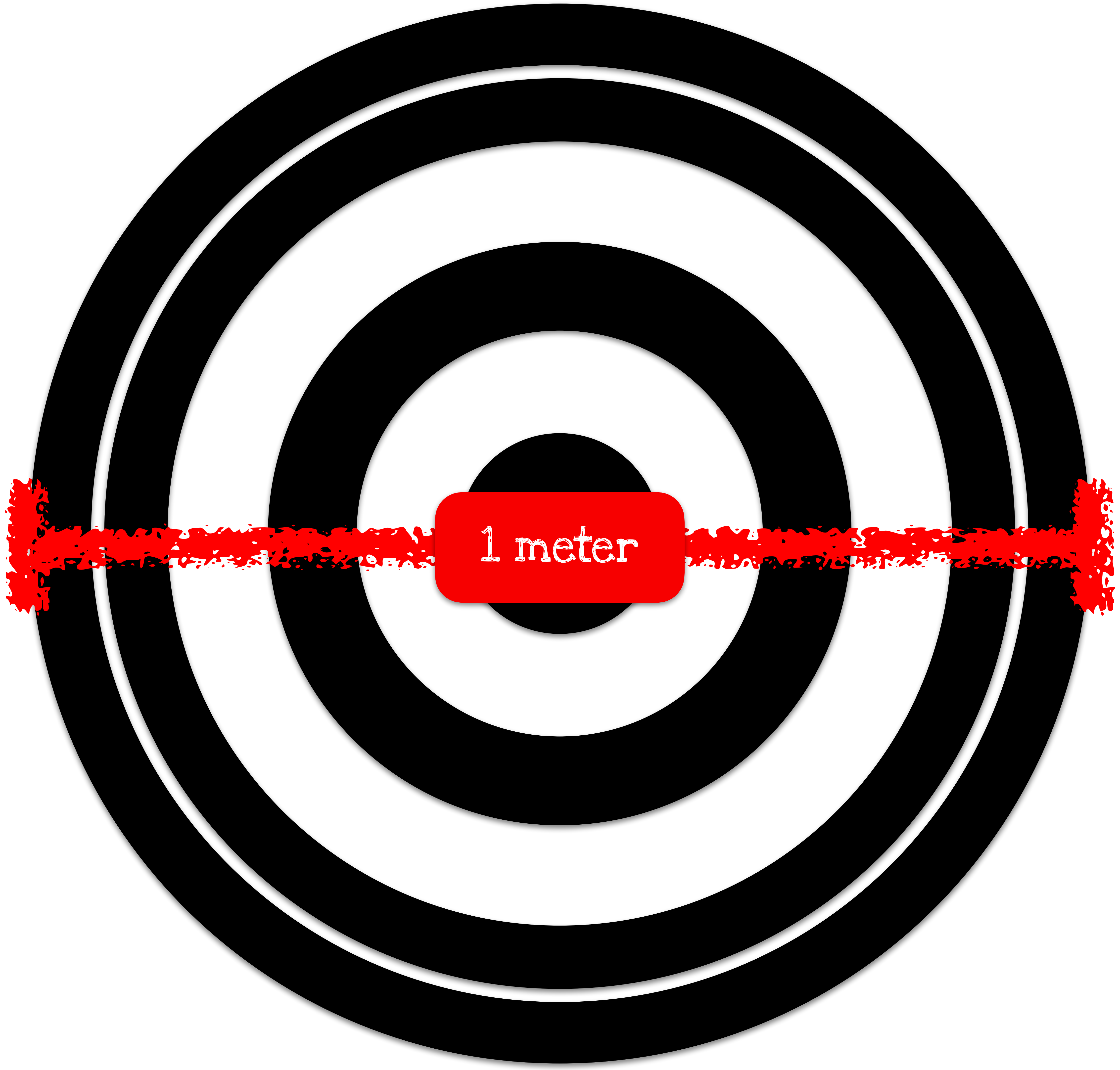
Prior distributions are an important part of Bayesian statistics

The distribution of $\boldsymbol{\theta}$ before any data are collected is the prior

$$f(\theta)$$

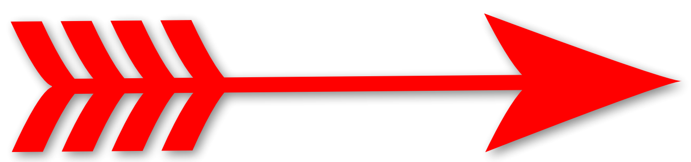
The prior describes your uncertainty in the parameters of your model

Priors: Archery



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery



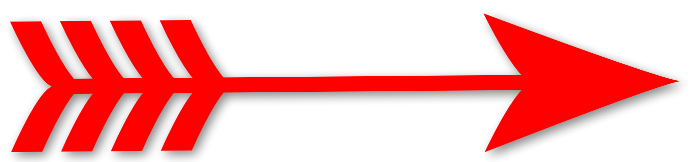
In this example we want to assess an archer's accuracy at hitting the bullseye

To quantify this, we will measure the distance *d* from the center of the target (in centimeters)

d is an absolute value

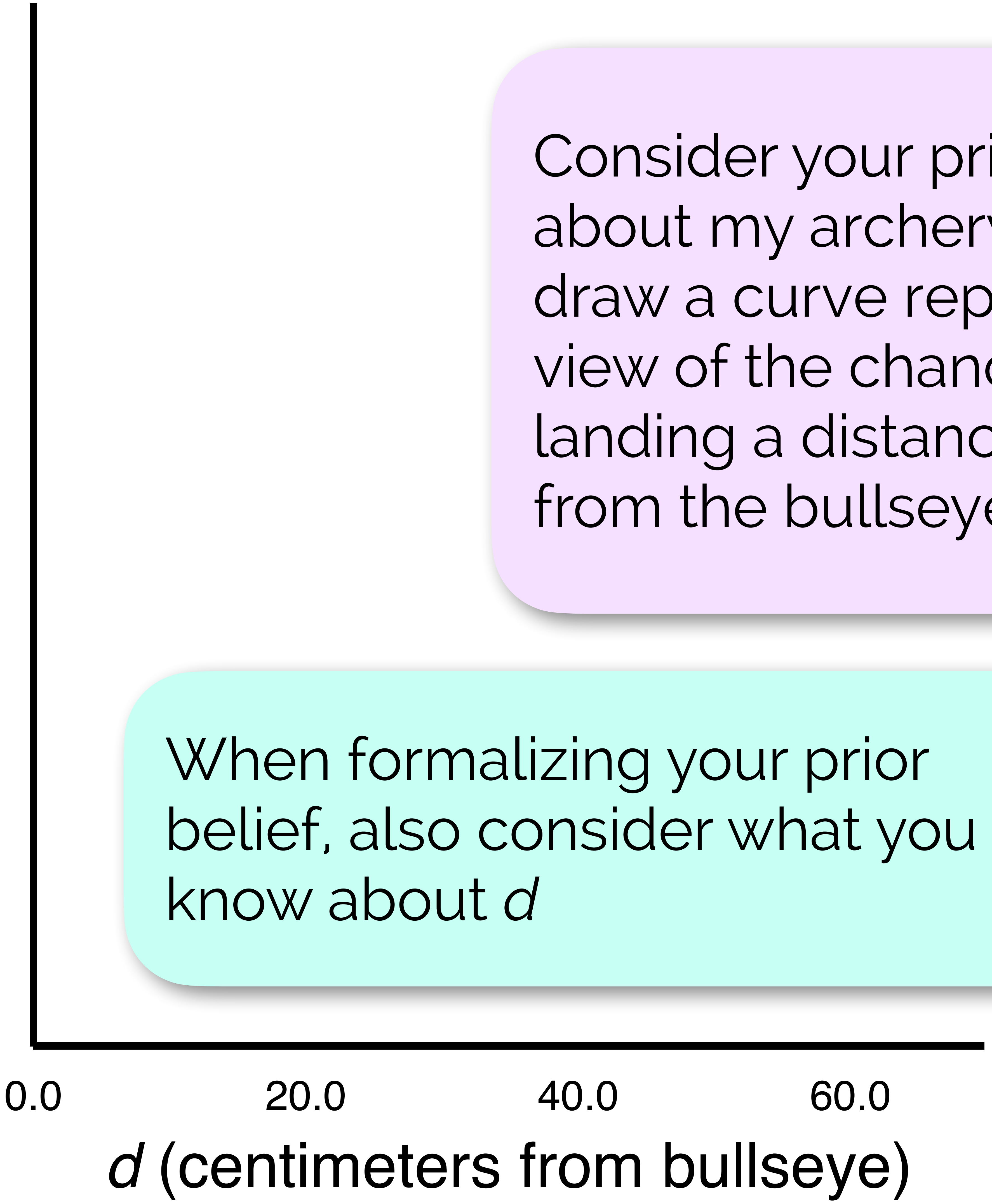
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery



Consider your prior knowledge about my archery abilities and draw a curve representing your view of the chances of my arrow landing a distance *d* centimeters from the bullseye

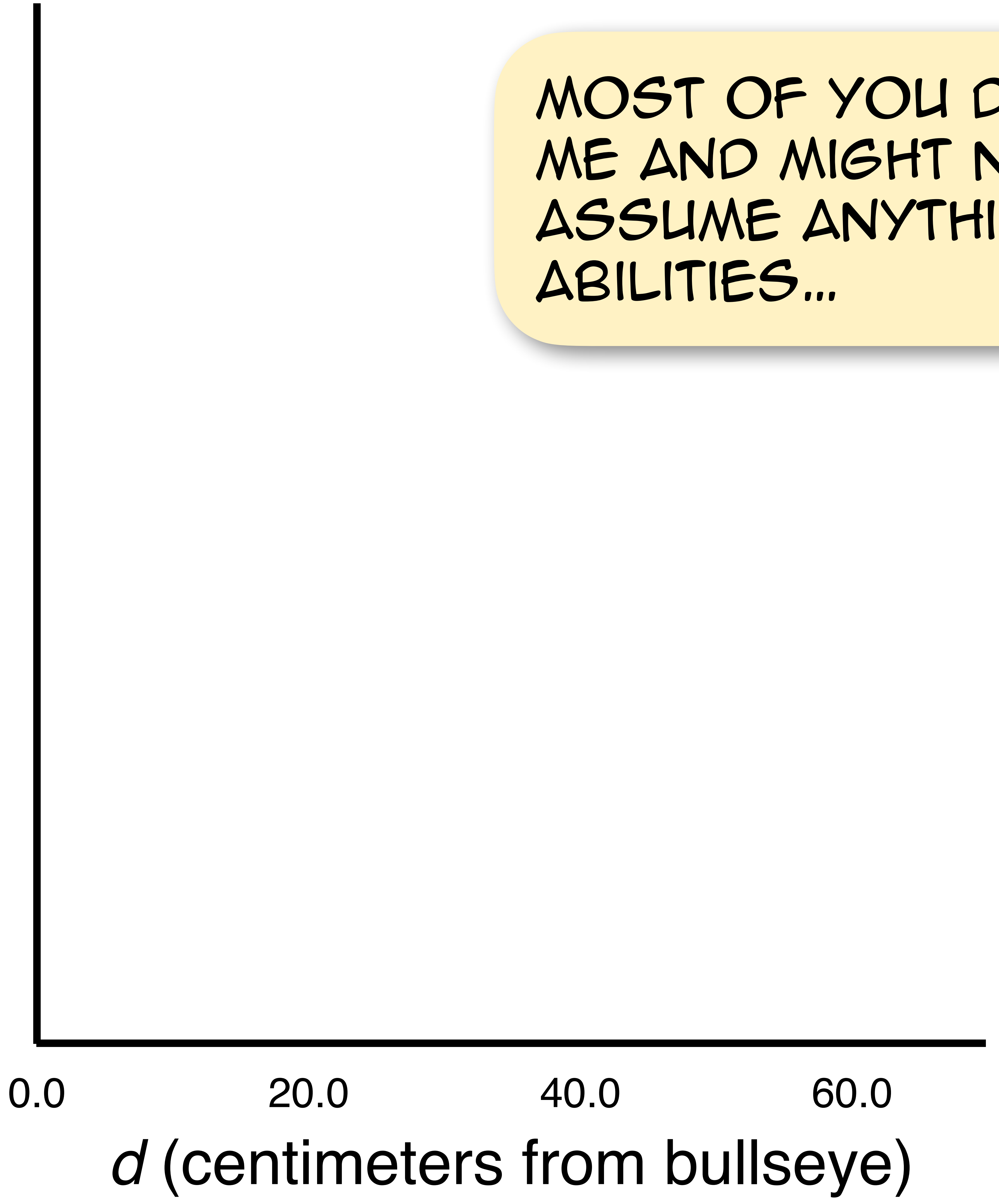
When formalizing your prior belief, also consider what you know about *d*



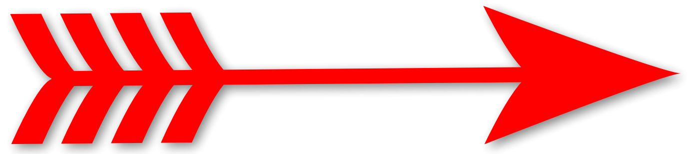
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery

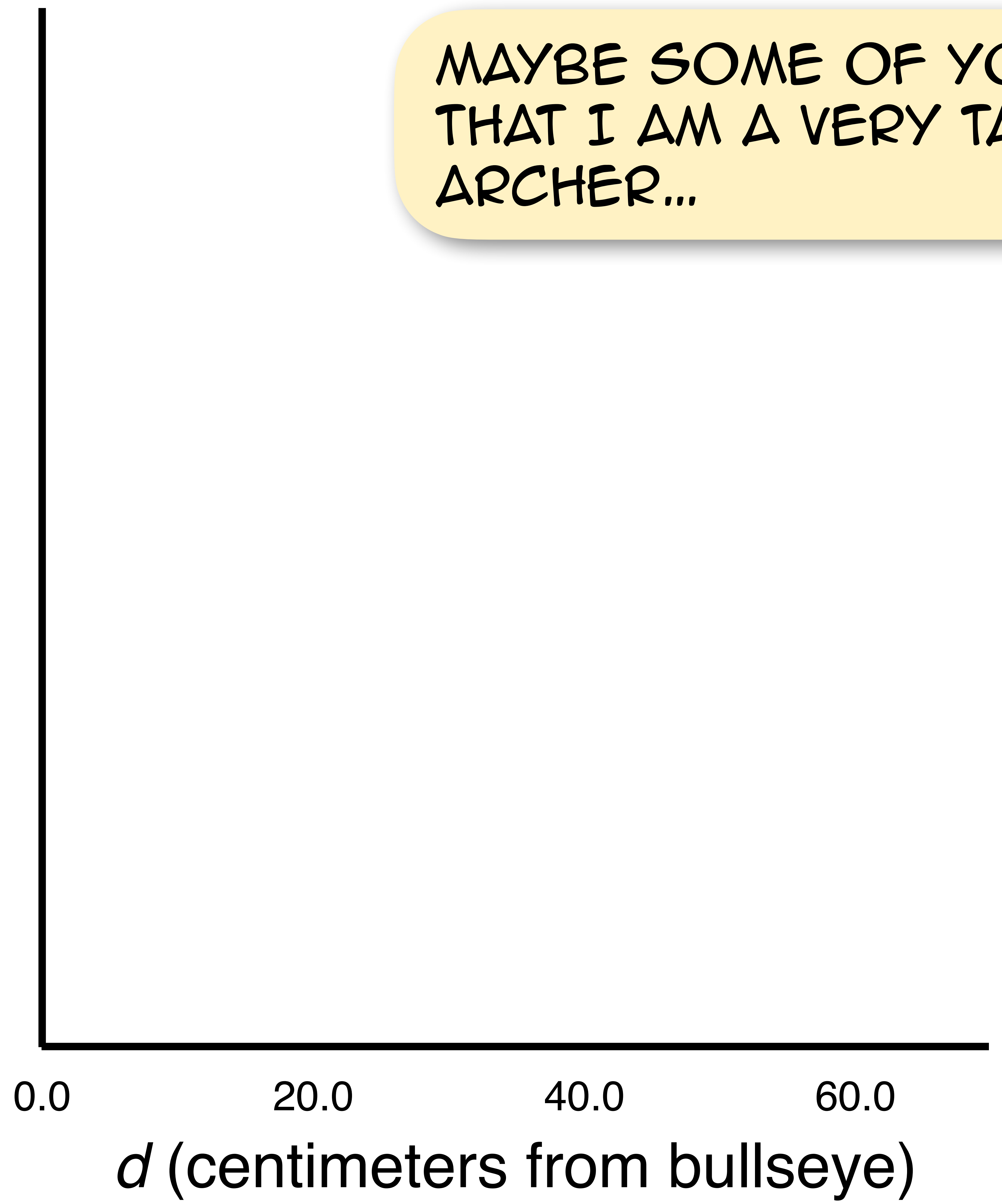
MOST OF YOU DON'T KNOW ME AND MIGHT NOT WANT TO ASSUME ANYTHING ABOUT MY ABILITIES...



Priors: Archery

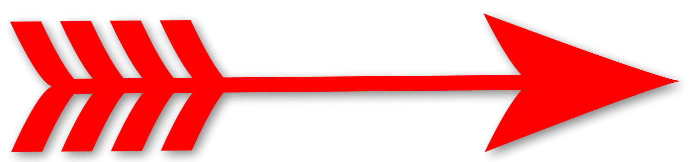


MAYBE SOME OF YOU ASSUME THAT I AM A VERY TALENTED ARCHER...

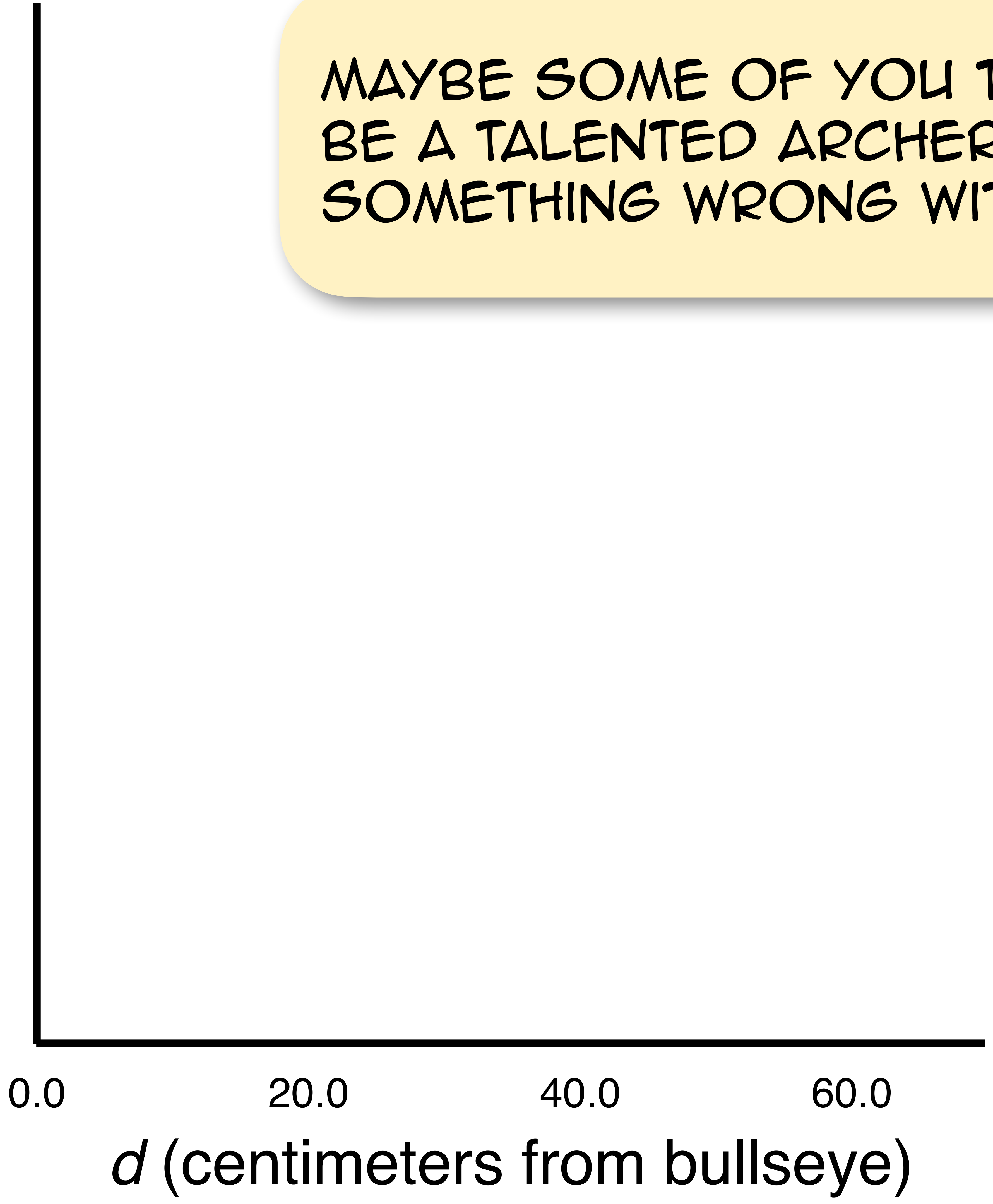


(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery

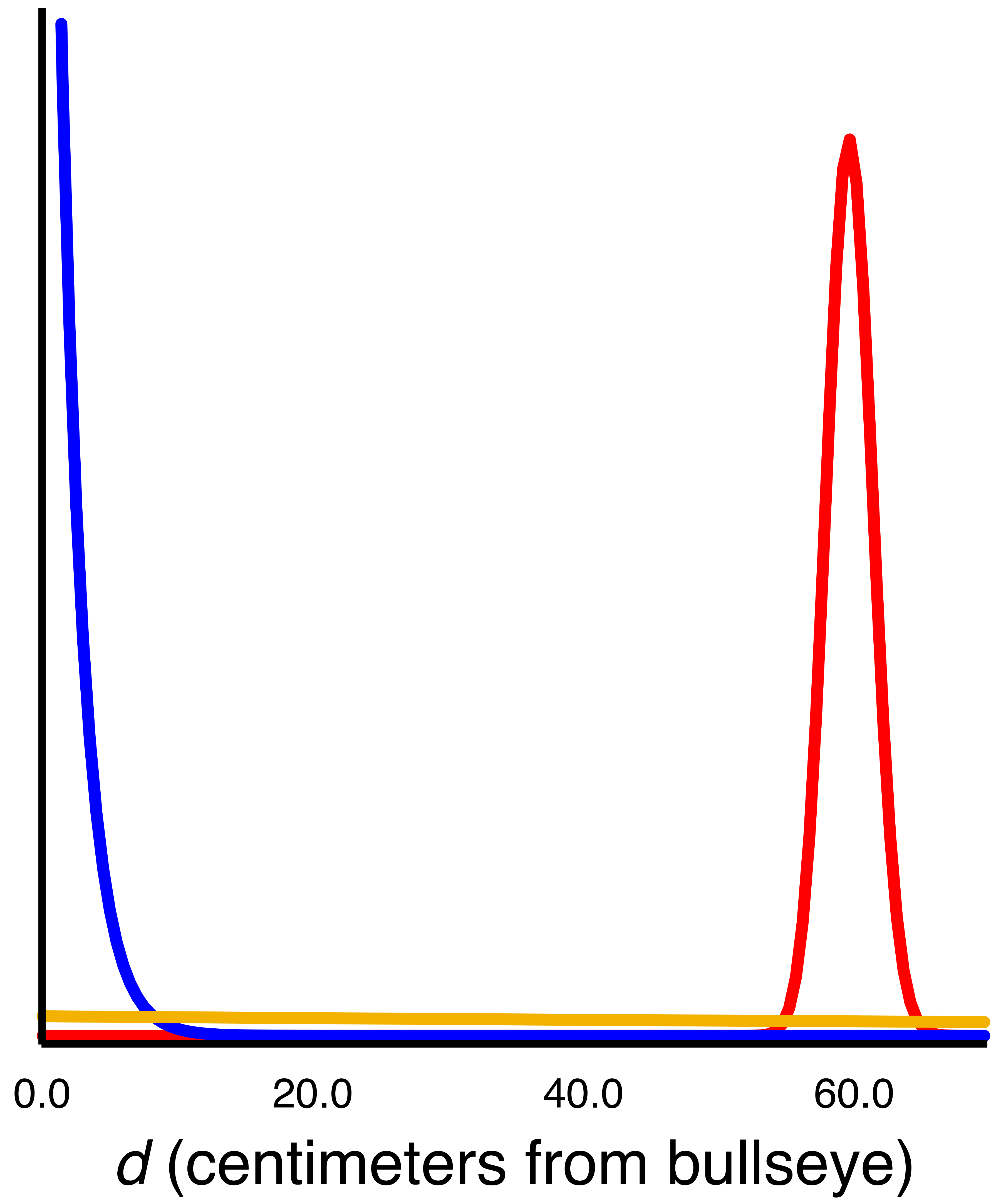


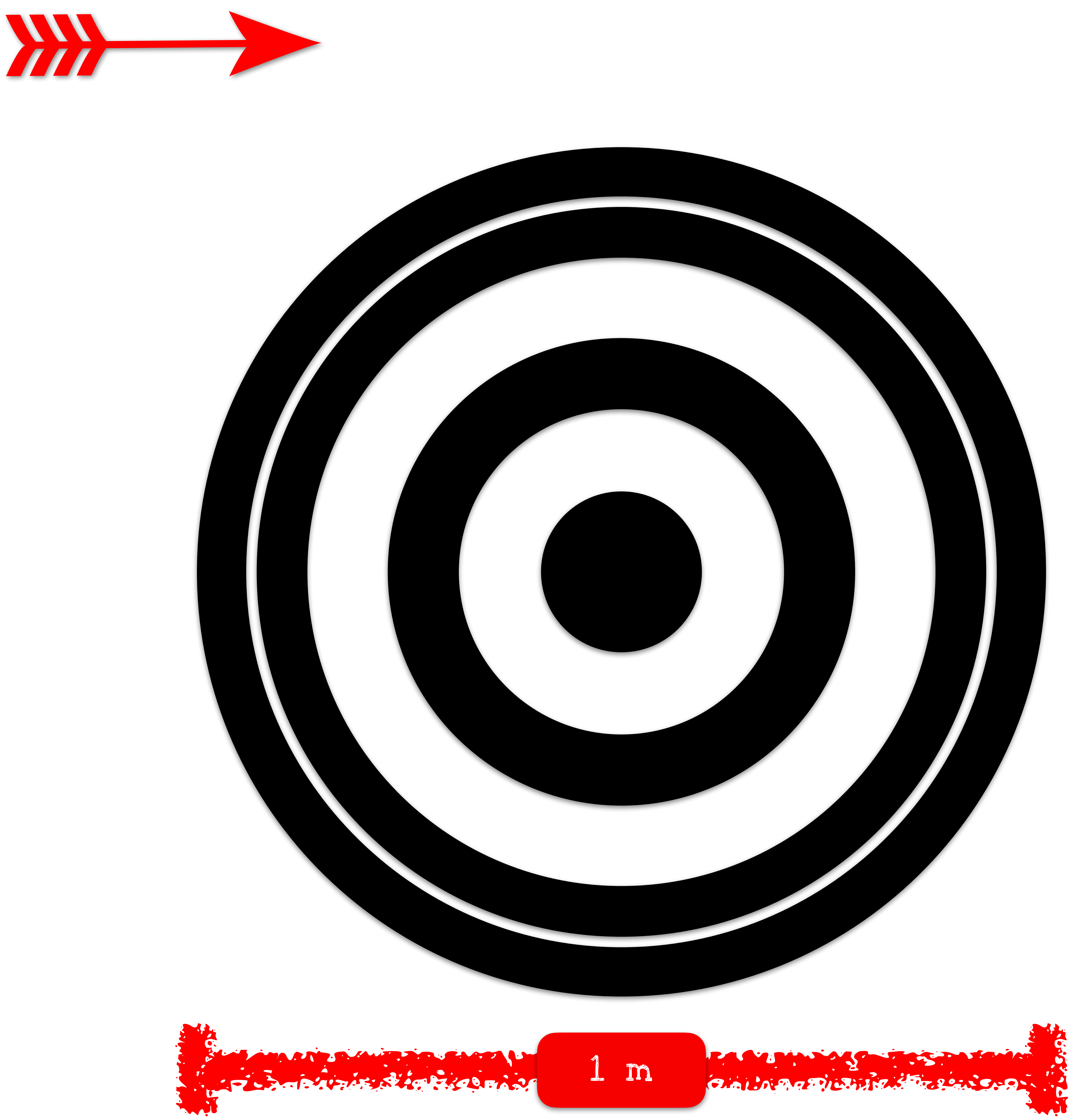
MAYBE SOME OF YOU THINK I MIGHT BE A TALENTED ARCHER AND THERE IS SOMETHING WRONG WITH MY BOW...



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

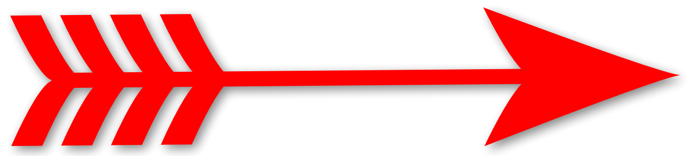
Priors: Archery

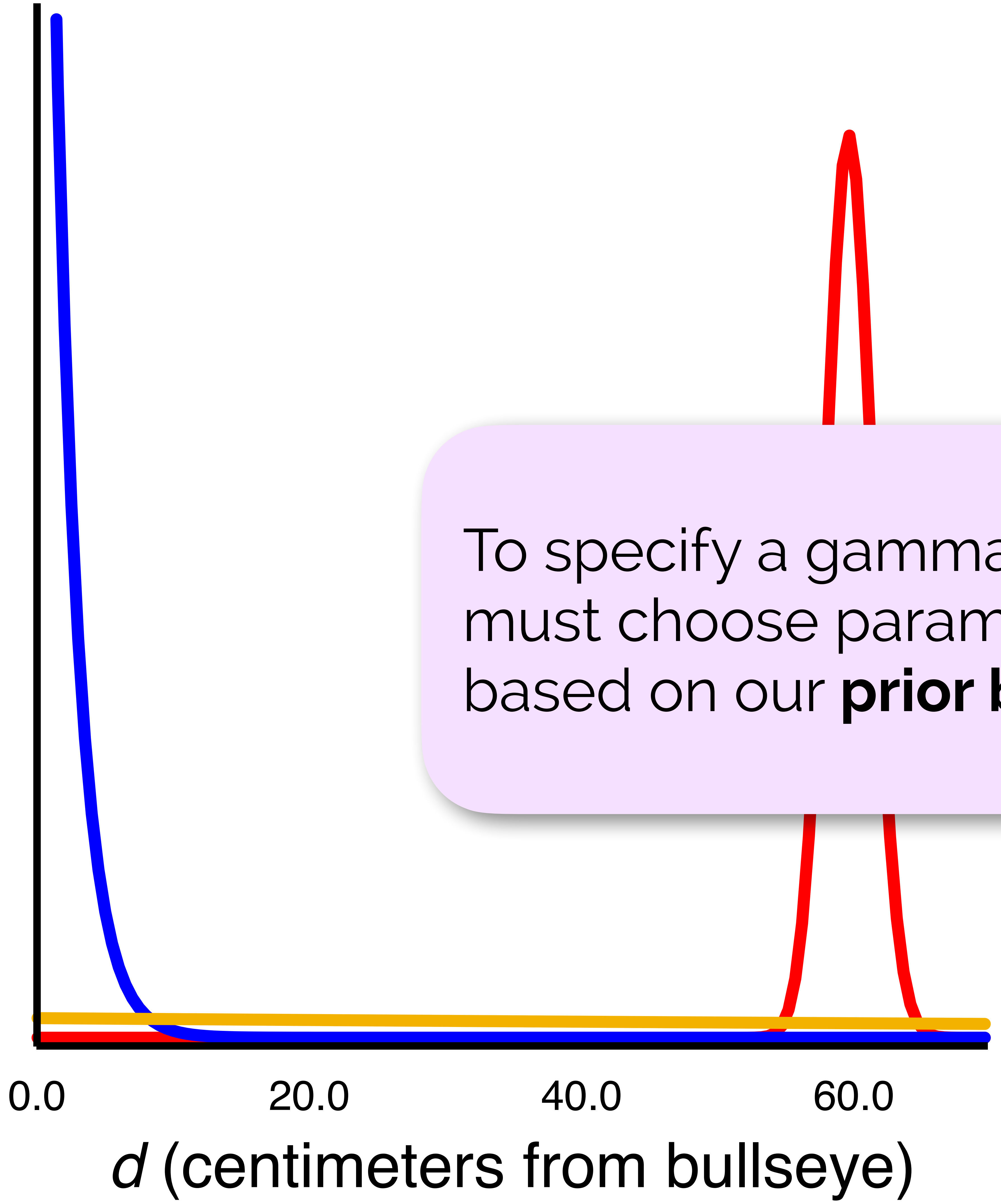




(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery





To specify a gamma prior, we must choose parameter values based on our **prior belief**

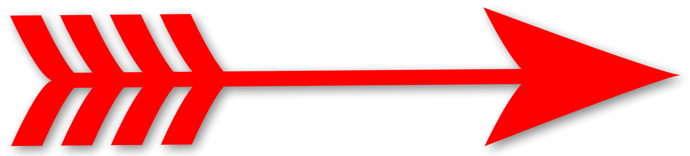
Each of these prior densities can be defined using a gamma distribution.

$$d \sim \text{Gamma}(\alpha, \beta)$$

$$f(d \mid \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} d^{\alpha-1} e^{-\frac{d}{\beta}}$$

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

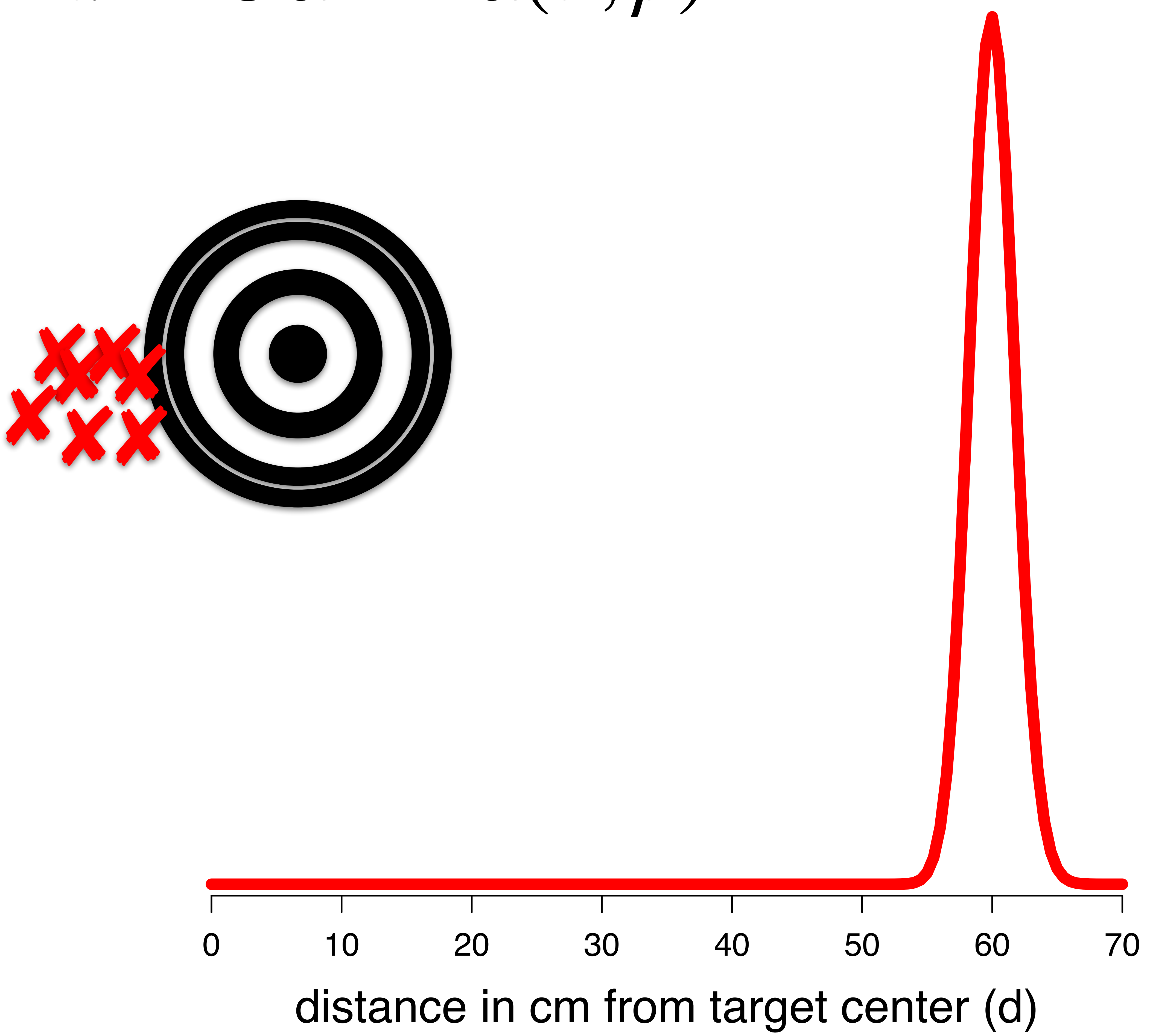
Priors: Archery



Let's assume that I will consistently miss the target

$$d \sim \text{Gamma}(\alpha, \beta)$$

This is a gamma distribution with a mean (*m*) of 60 and a variance (*v*) of 3

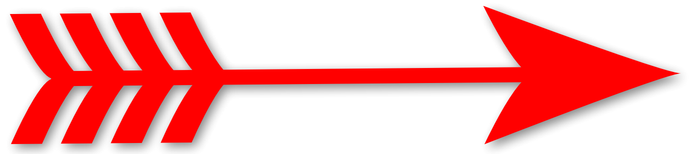


mean = accuracy

variance = precision

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery

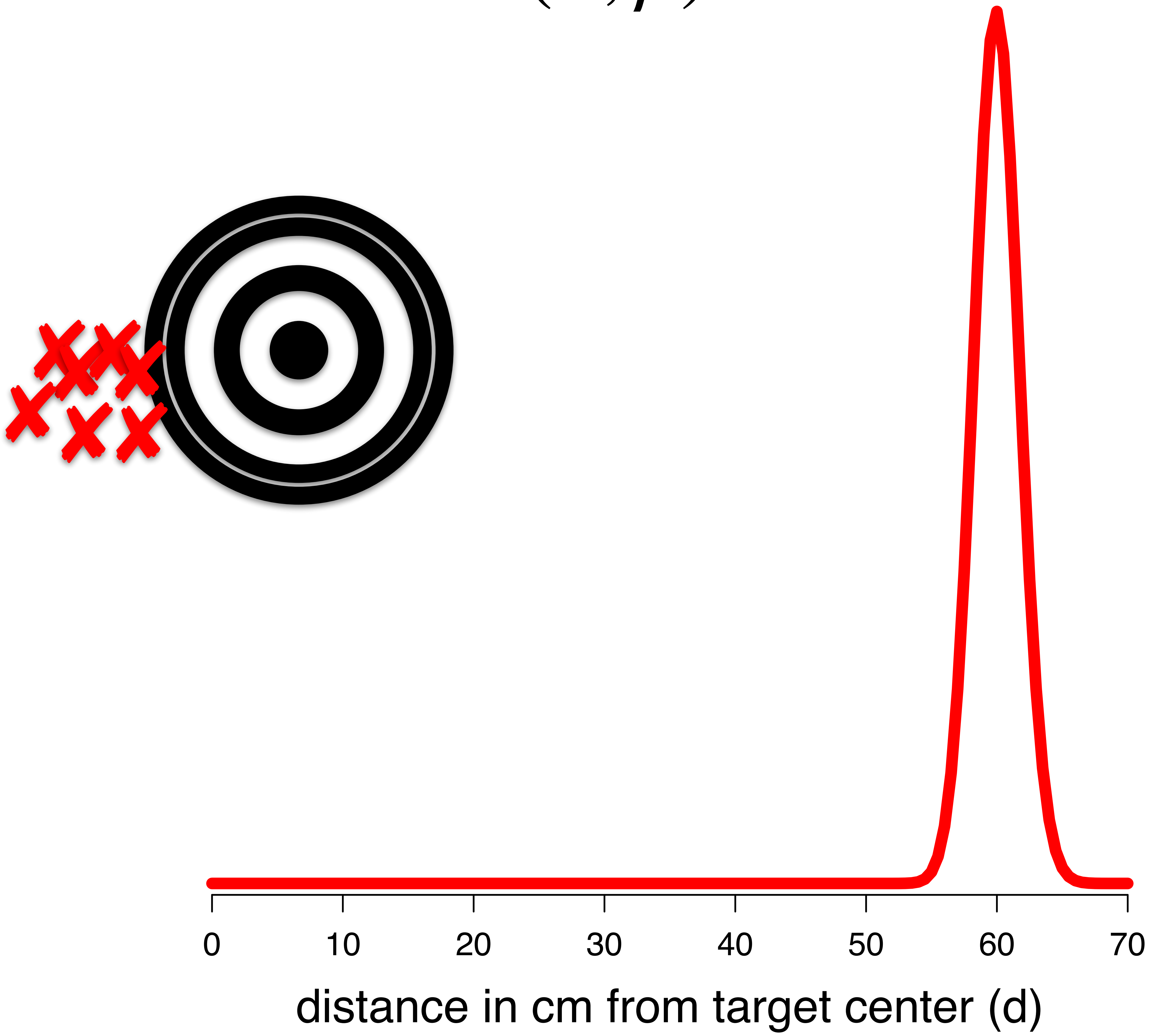


If we have prior knowledge of the mean and variance of the gamma distribution, we can compute the shape and rate parameters

$$m = \frac{\alpha}{\beta}, \; \alpha = \frac{m^2}{v}$$

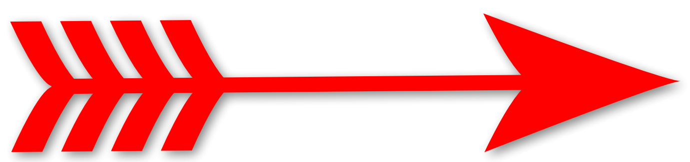
$$v = \frac{\alpha}{\beta^2}, \; \beta = \frac{m}{v}$$

$d \sim \text{Gamma}(\alpha, \beta)$



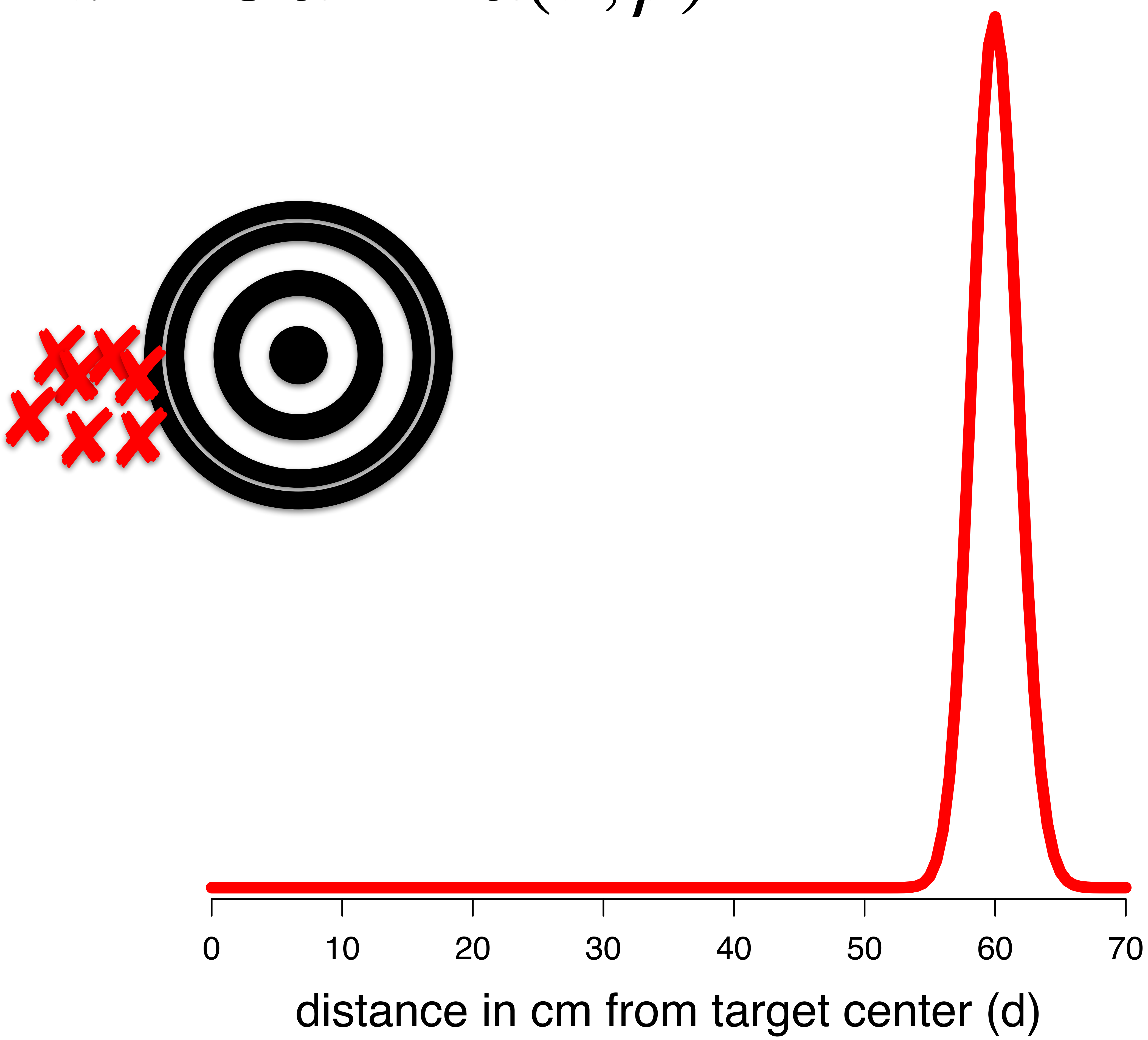
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Priors: Archery



$$m = 60, \quad v = 3$$

$$d \sim \text{Gamma}(\alpha, \beta)$$

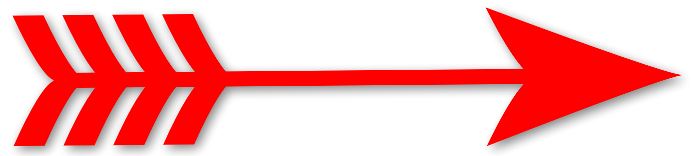


$$\alpha = \frac{60^2}{3} = 1200$$

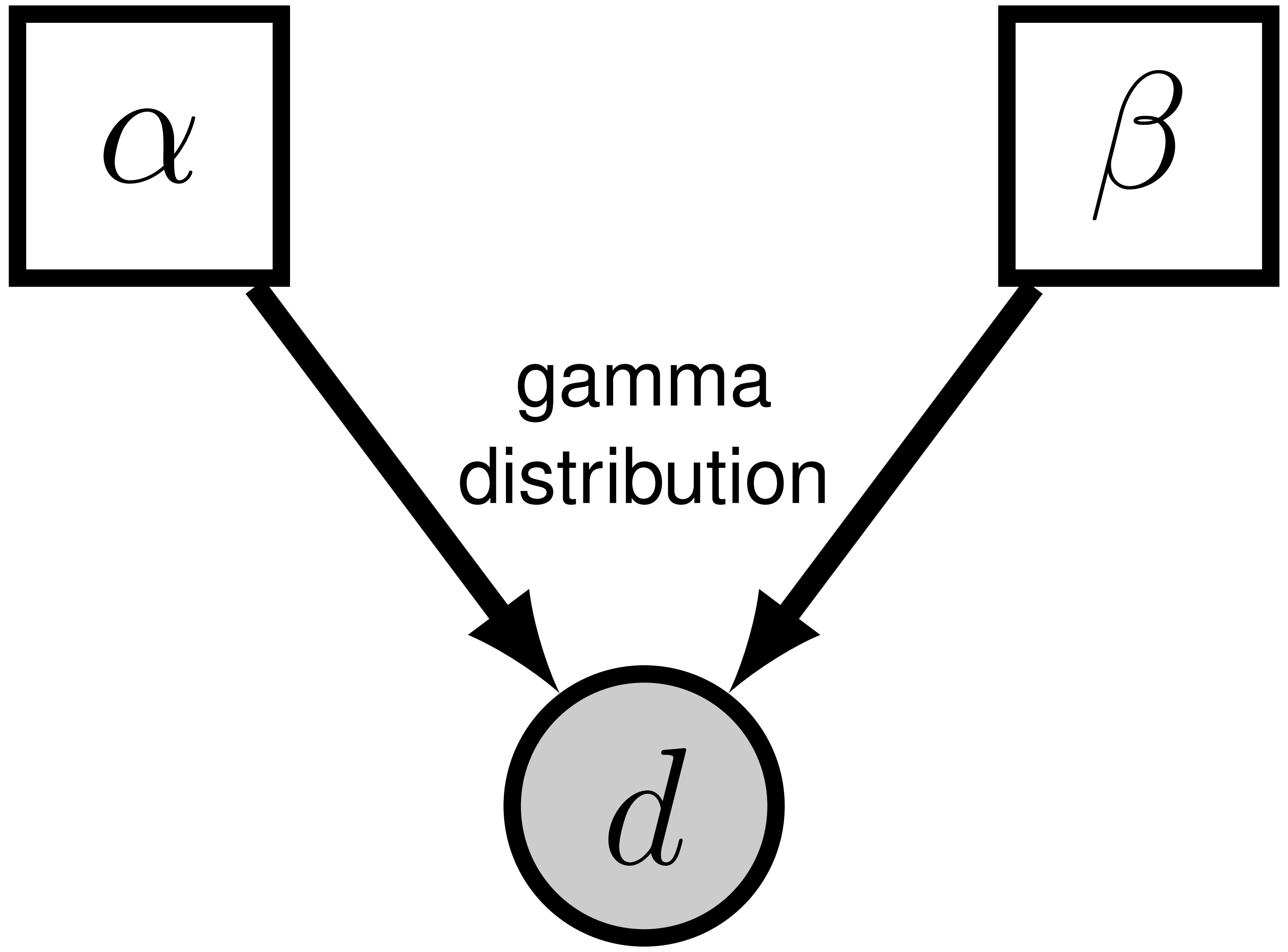
$$\beta = \frac{60}{3} = 20$$

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

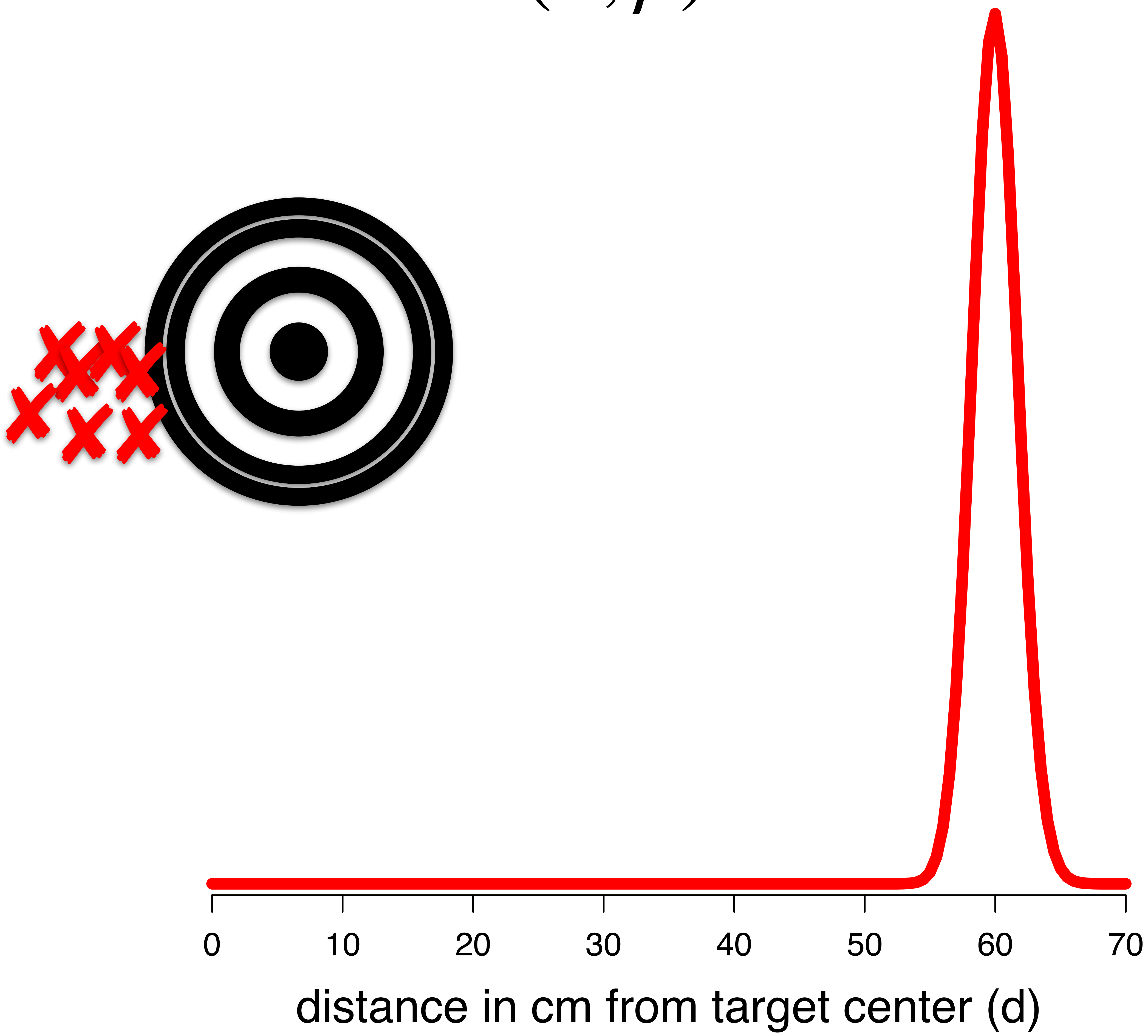
Priors: Archery



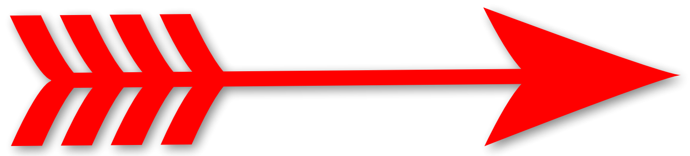
Another way of expressing this distribution is with a probabilistic graphical model



$d \sim \text{Gamma}(\alpha, \beta)$

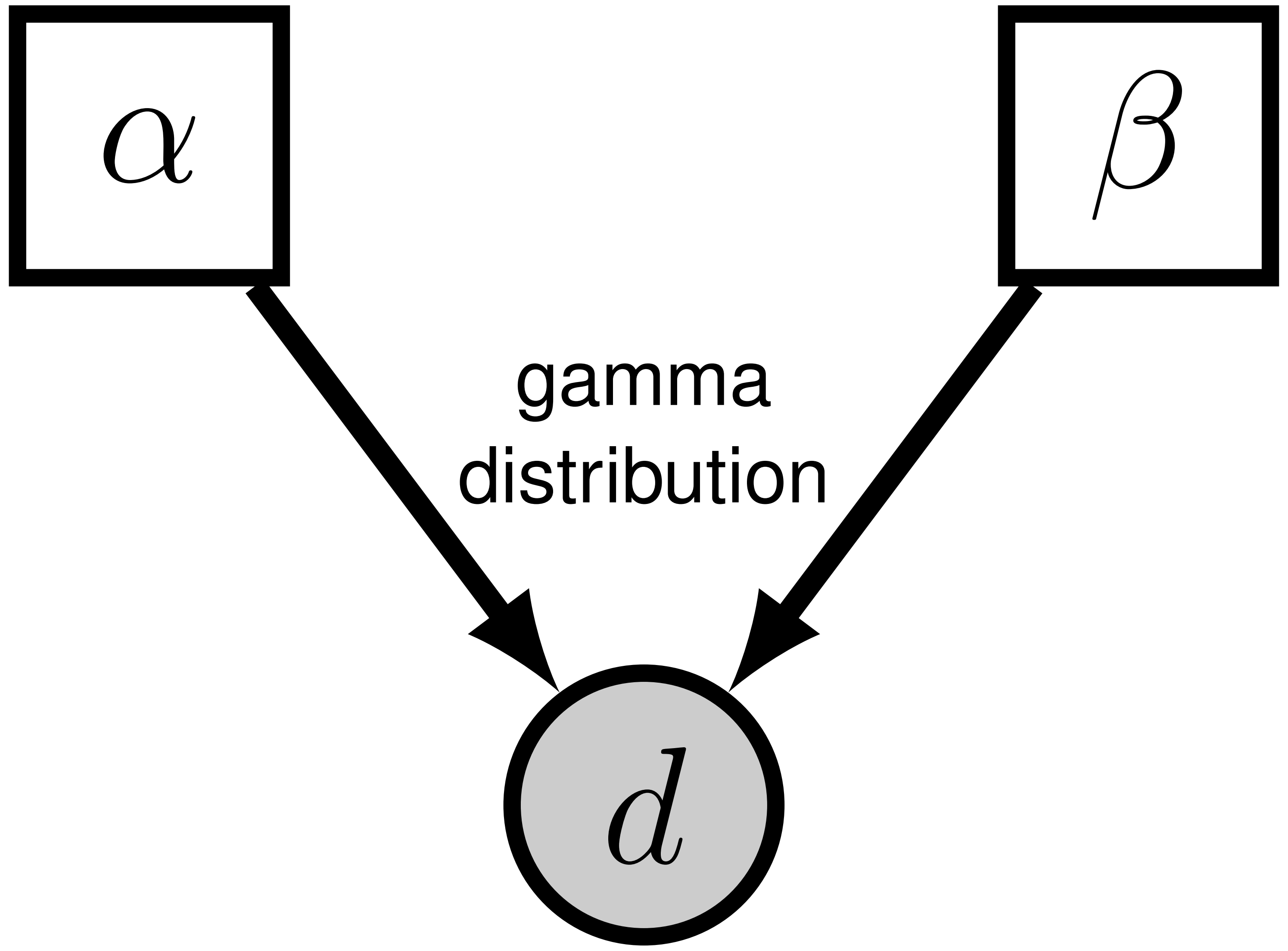


Priors: Archery



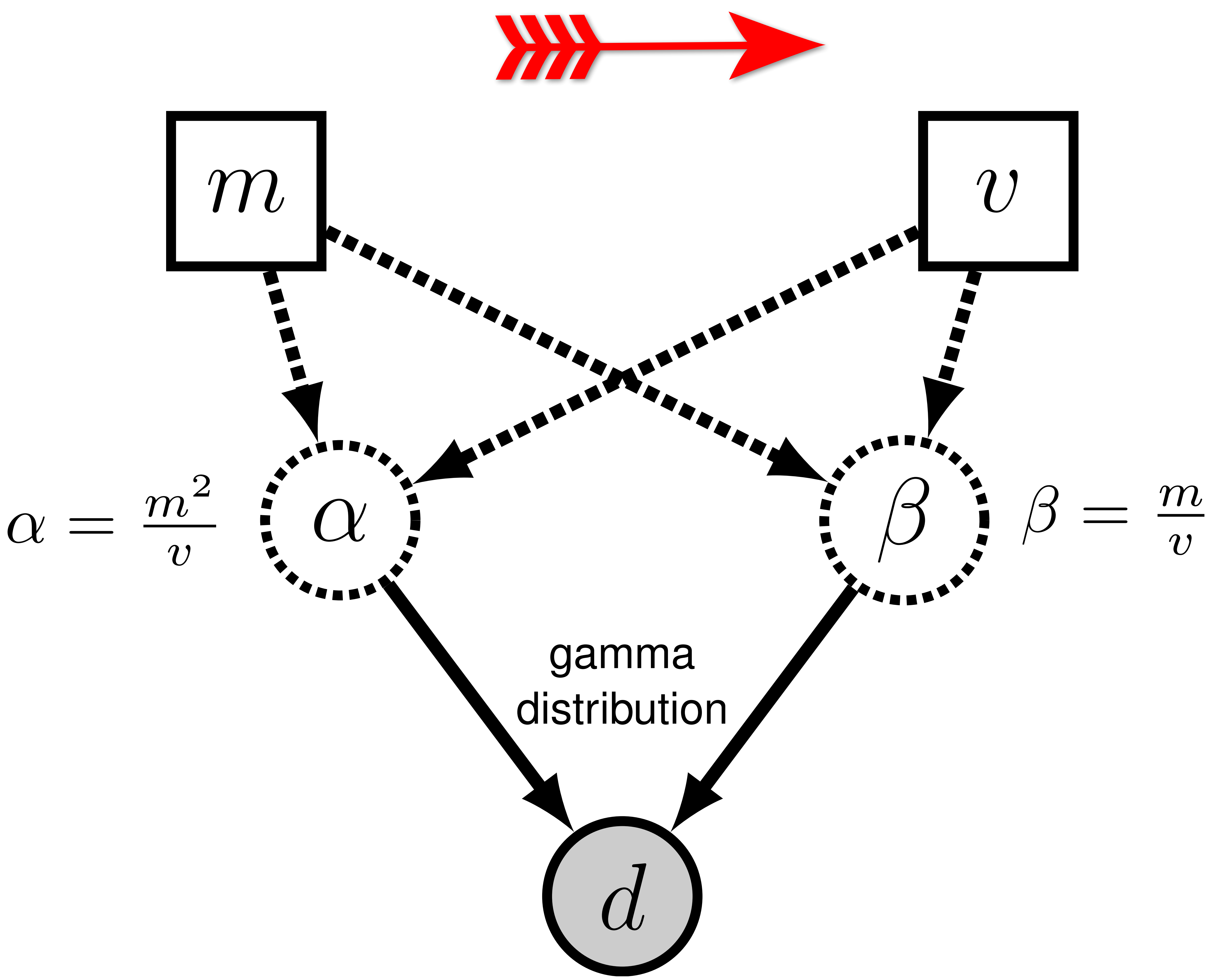
$$d \sim \text{Gamma}(\alpha, \beta)$$

This shows that our observed datum (d = a single observed shot) is conditionally dependent on the shape (**α**) and rate (β) of the gamma distribution



Priors: Archery

We can parameterize the model using the mean (*m*) and variance (*v*), where **α** and *β* are computed using *m* and *v*



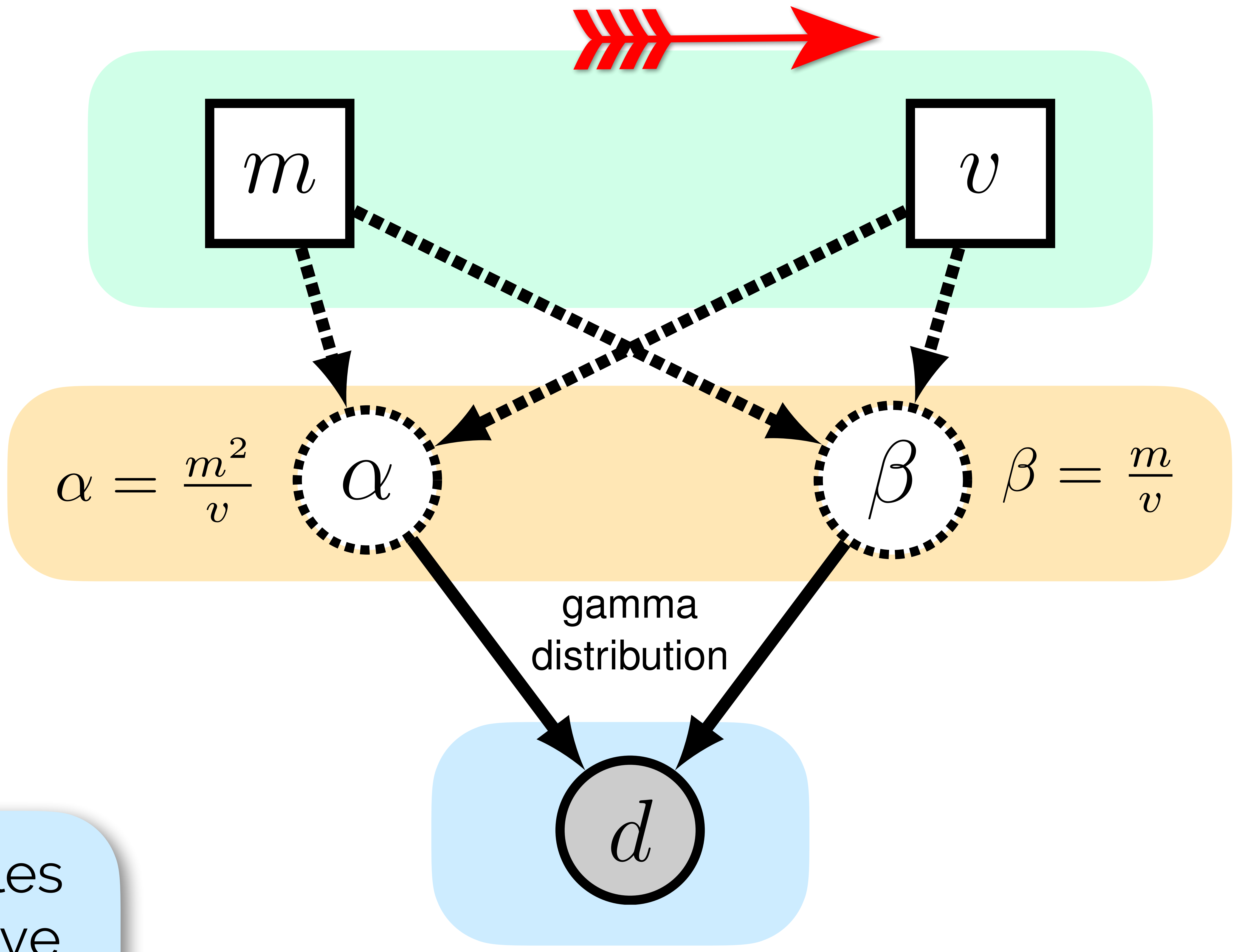
We may have more intuition about the mean and variance than we do about the shape and rate.

Priors: Archery

Constant nodes represent a fixed value that is asserted or known

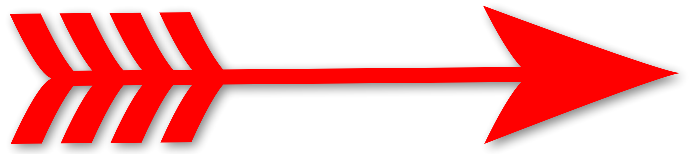
Deterministic nodes represent unknown random variable whose values are determined by other nodes

Stochastic nodes are random variables generated by the model. If we observe the value of a stochastic node, we fix it to that value

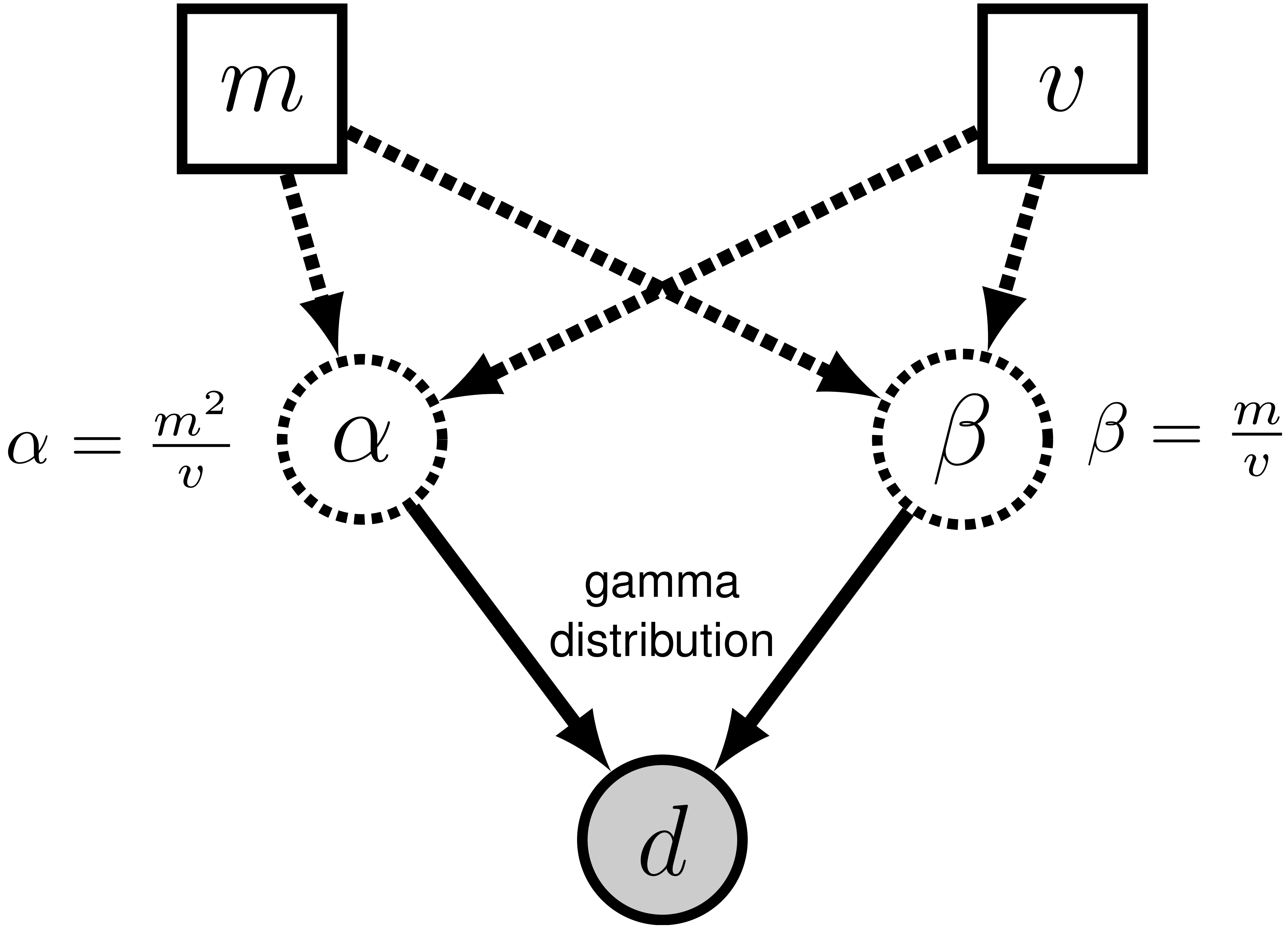


This graphical model has 3 types of nodes

Priors: Archery



If we set *m* and *v* to values corresponding to our assumed model, then we can calculate the likelihood of any observed shot



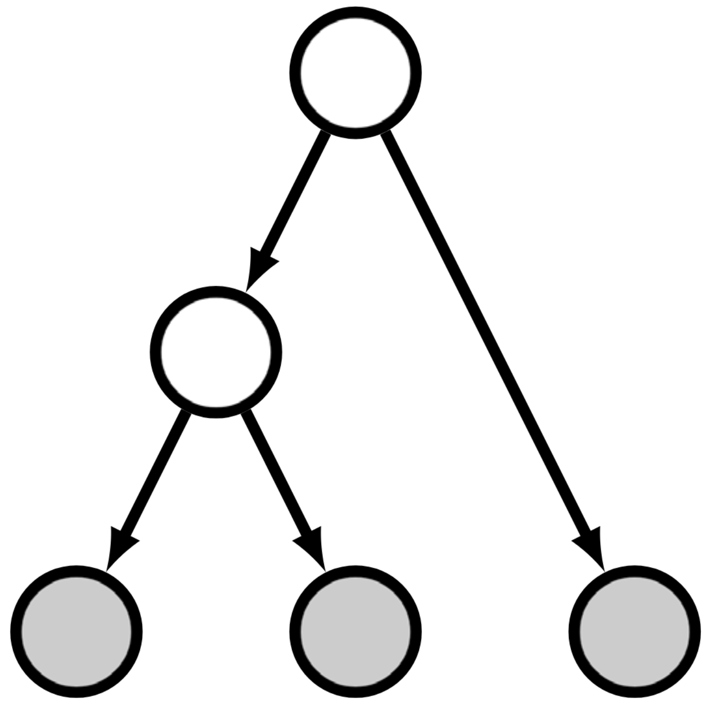
$$f(d \mid \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} d^{\alpha-1} e^{-\frac{d}{\beta}}$$

$$f(d = 39.76 \mid \alpha = 1200, \beta = 20) = 7.89916e - 40$$

RevBayes Demo: Archery

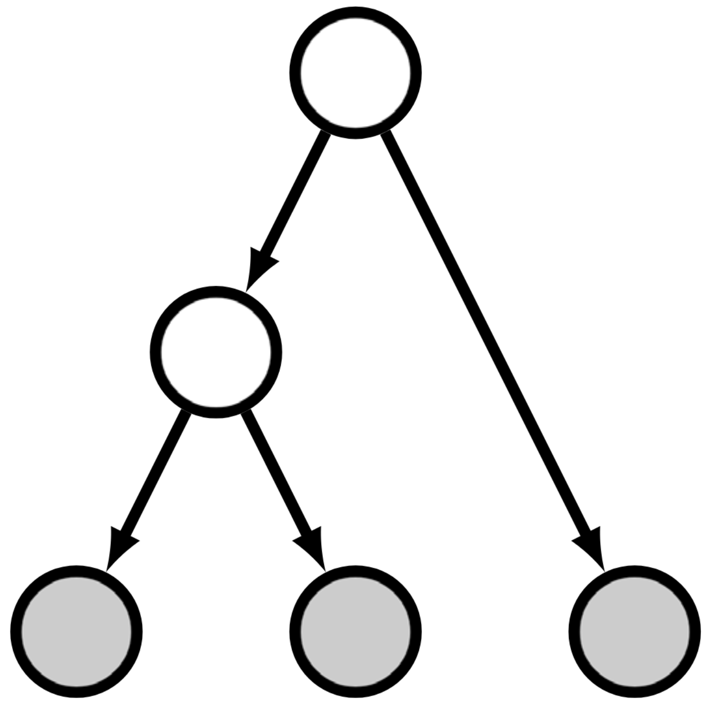
Bayesian Inference of Evolutionary Parameters

Fully integrative Bayesian inference of phylogenetic parameters using probabilistic graphical models and an interpreted language



http://revbayes.com

https://github.com/revbayes

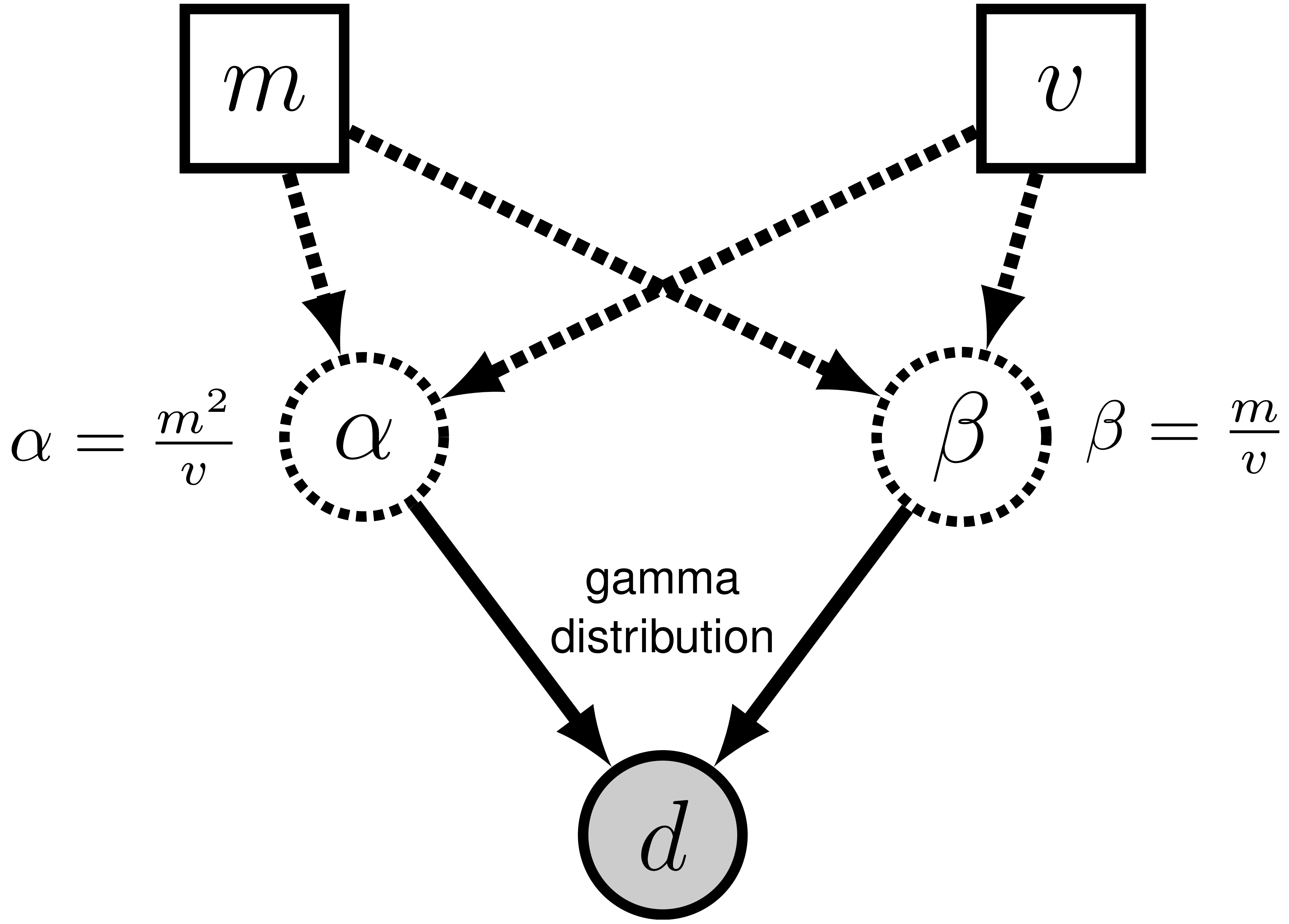


Höhna et al. 2016. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology*. (doi: 10.1093/sysbio/syw021)

Graphical Models in RevBayes

Graphical models provide a way to visually and computationally represent complex, parameter-rich probabilistic models

We can depict the conditional dependence structure of various parameters and other variables



Höhna et al. 2014. Probabilistic graphical model representation in Phylogenetics. *Systematic Biology*. (doi: 10.1093/sysbio/syu039)

RevBayes Demo: Archery

The Rev language for calculating the probability of 1 data observation (observed_shot) given a mean and variance

```
mean <- 60
var <- 3

alpha := (mean * mean) / var
beta := mean / var

observed_shot = 39.76

d ~ dnGamma(alpha, beta)
d.clamp(observed_shot)

d.lnProbability()
   -90.03665
```

Priors: Archery

What if we do not know *m* and *v*?

We can use maximum likelihood or Bayesian methods to estimate their values

Maximum likelihood methods require us to find the values of *m* and *v* that maximize

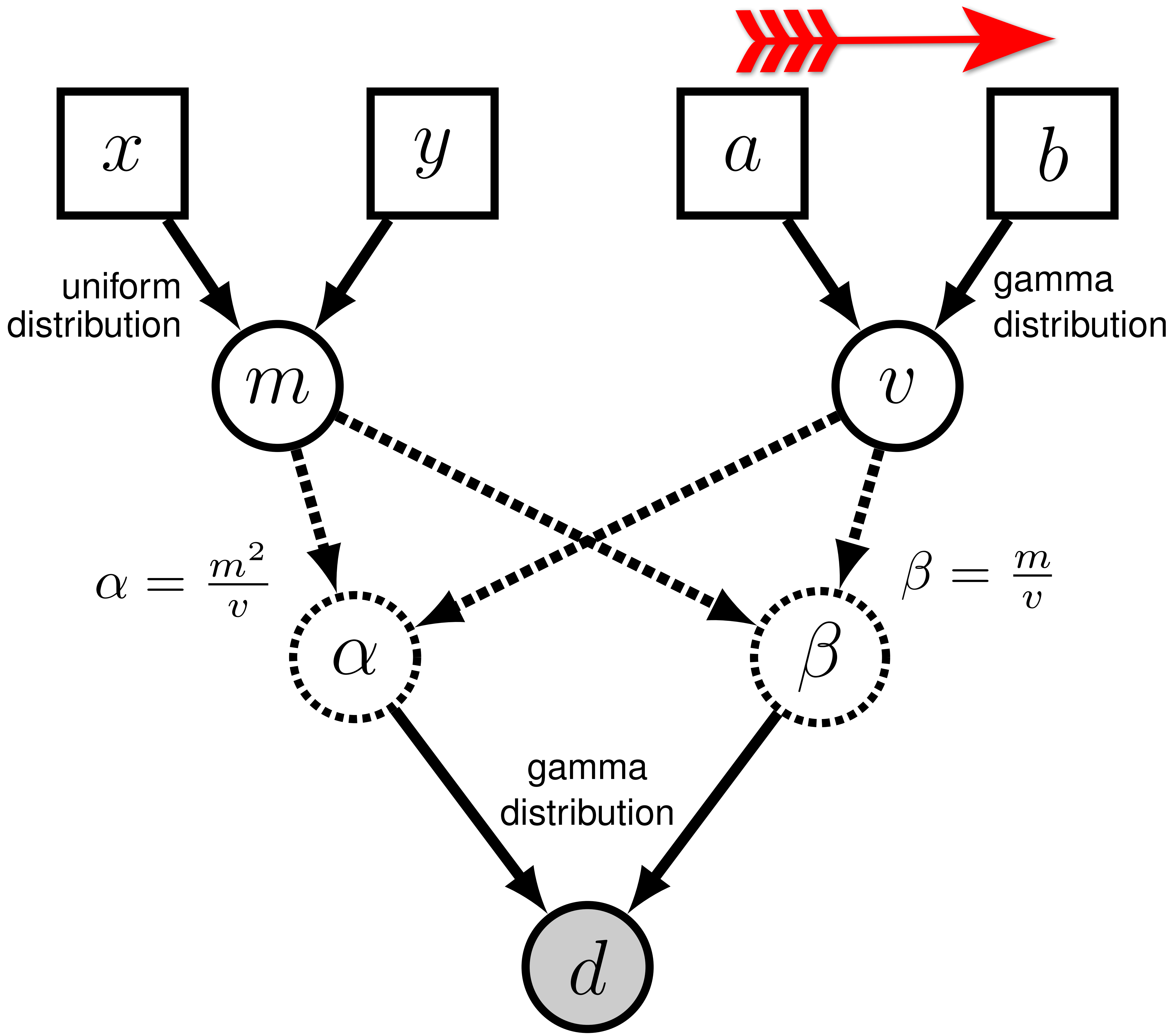
$$f(d \mid m, v)$$

Bayesian methods use prior distributions to describe our uncertainty in *m* and *v* and estimate

$$f(m, v \mid d)$$

Priors: Archery

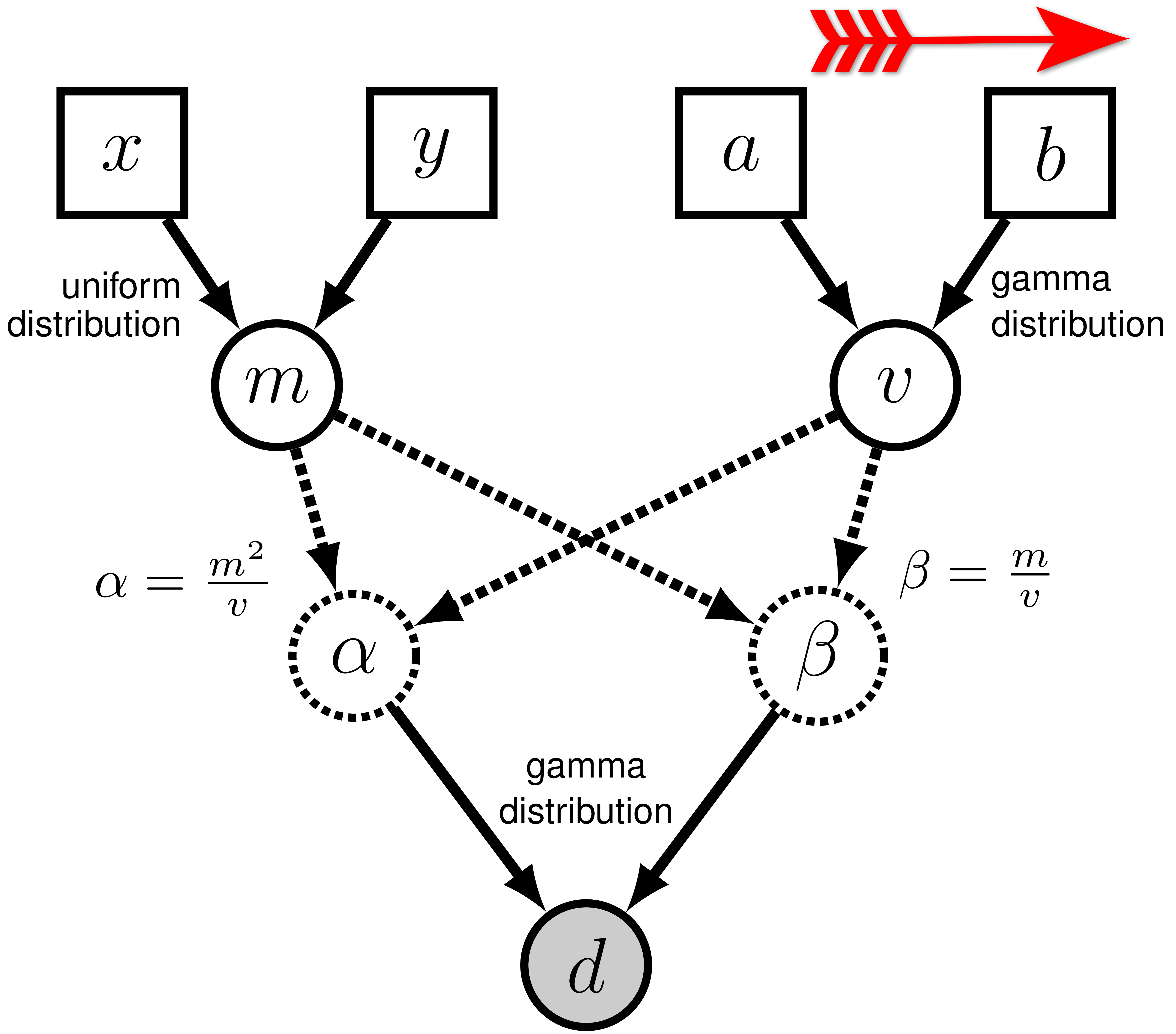
We must define prior distributions for *m* and *v* to account for uncertainty and estimate the posterior densities of those parameters



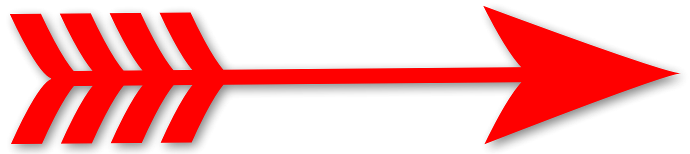
Priors: Archery

Now *x* and *y* are the parameters of the uniform prior on *m*

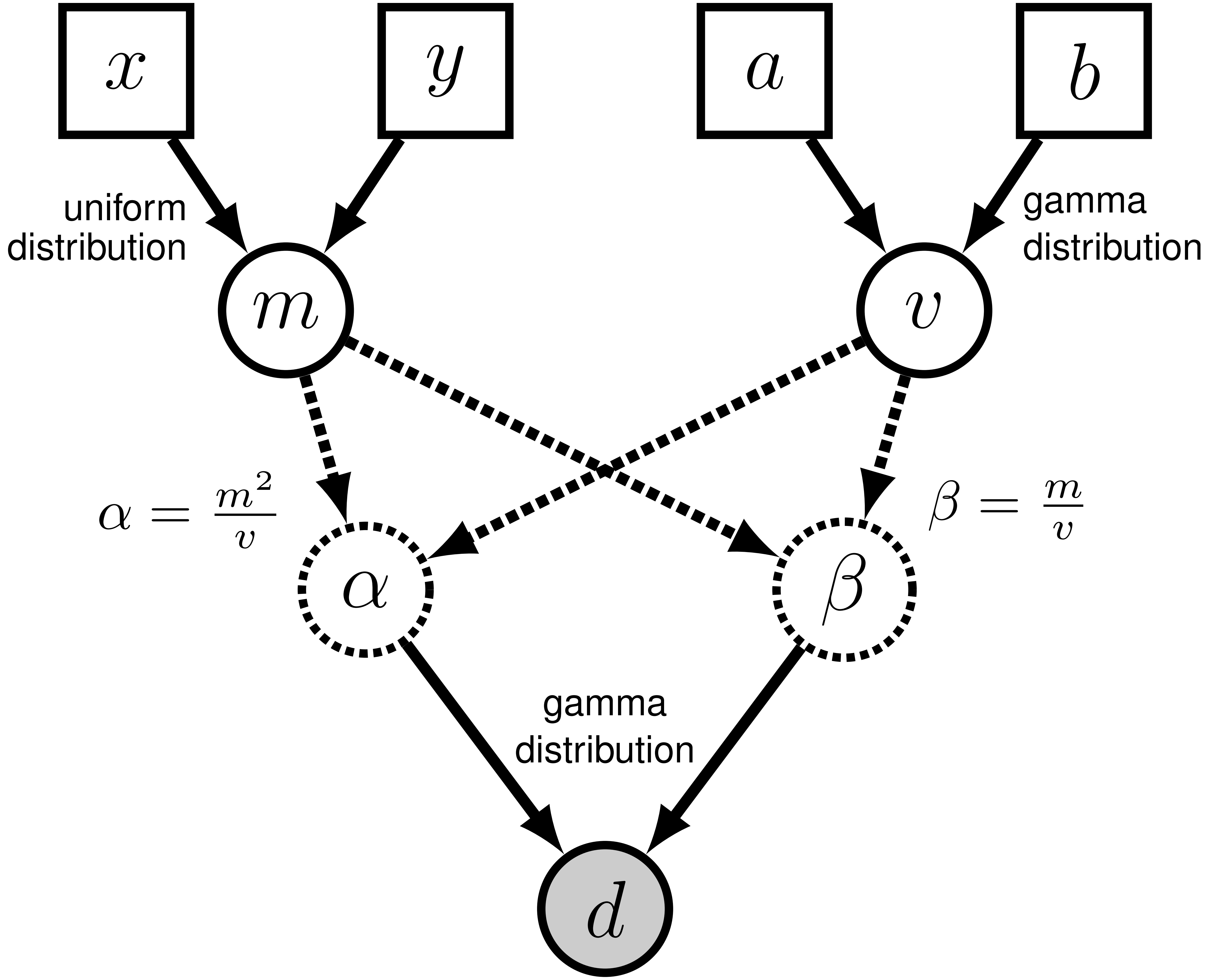
And *a* and *b* are the shoe and rate parameters of the gamma prior on *v*



Priors: Archery



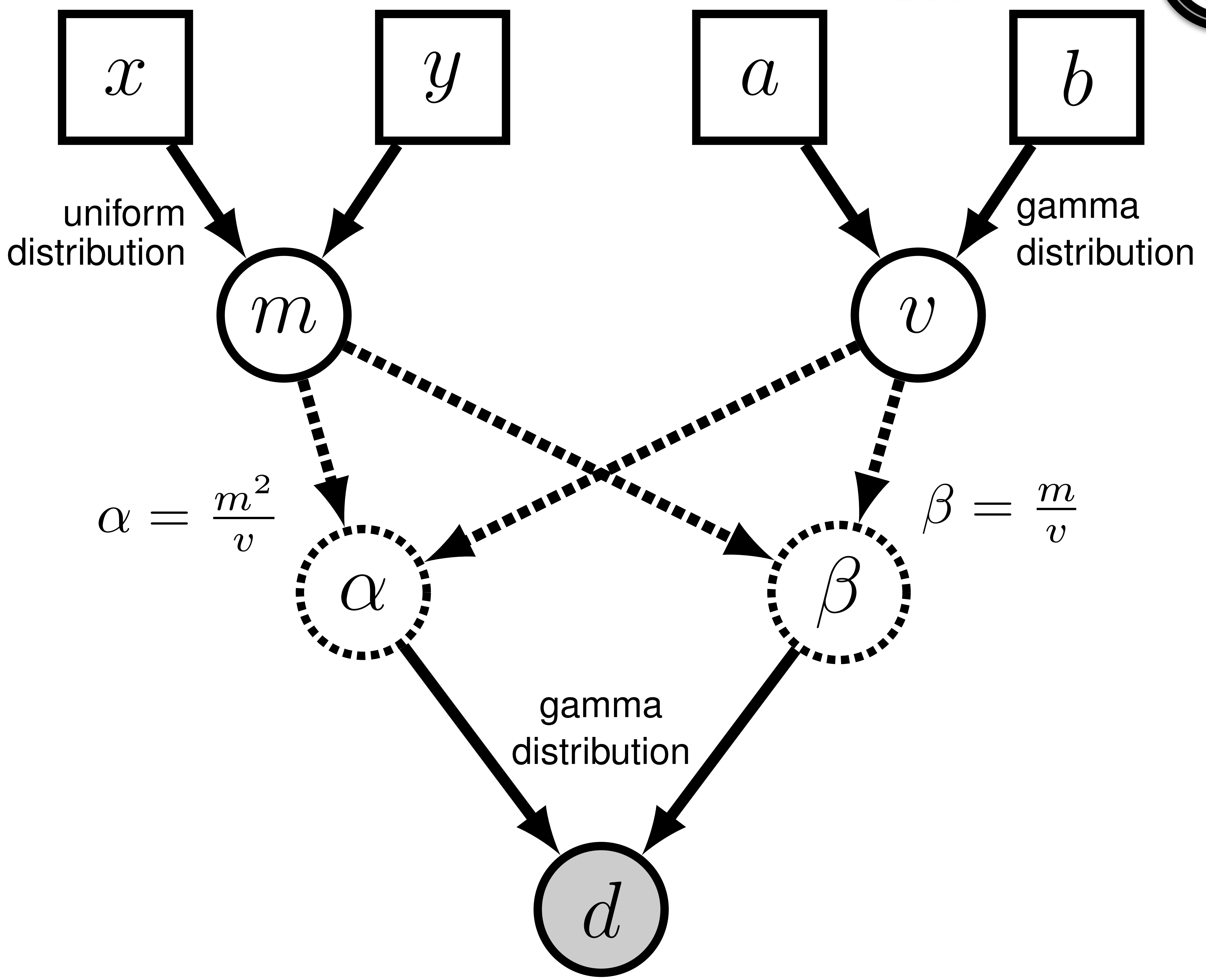
Stochastic nodes that are not observed are random variables that are unknown and estimated



Priors: Archery

The values we choose for the parameters of these prior distributions should reflect our prior knowledge

If we observed a previous shot at 39.76 cm, the we can use this to parameterize our priors for analysis of future observations



Priors: Archery

$m \sim \text{Uniform}(x, y)$

$x = 10$

$y = 50$

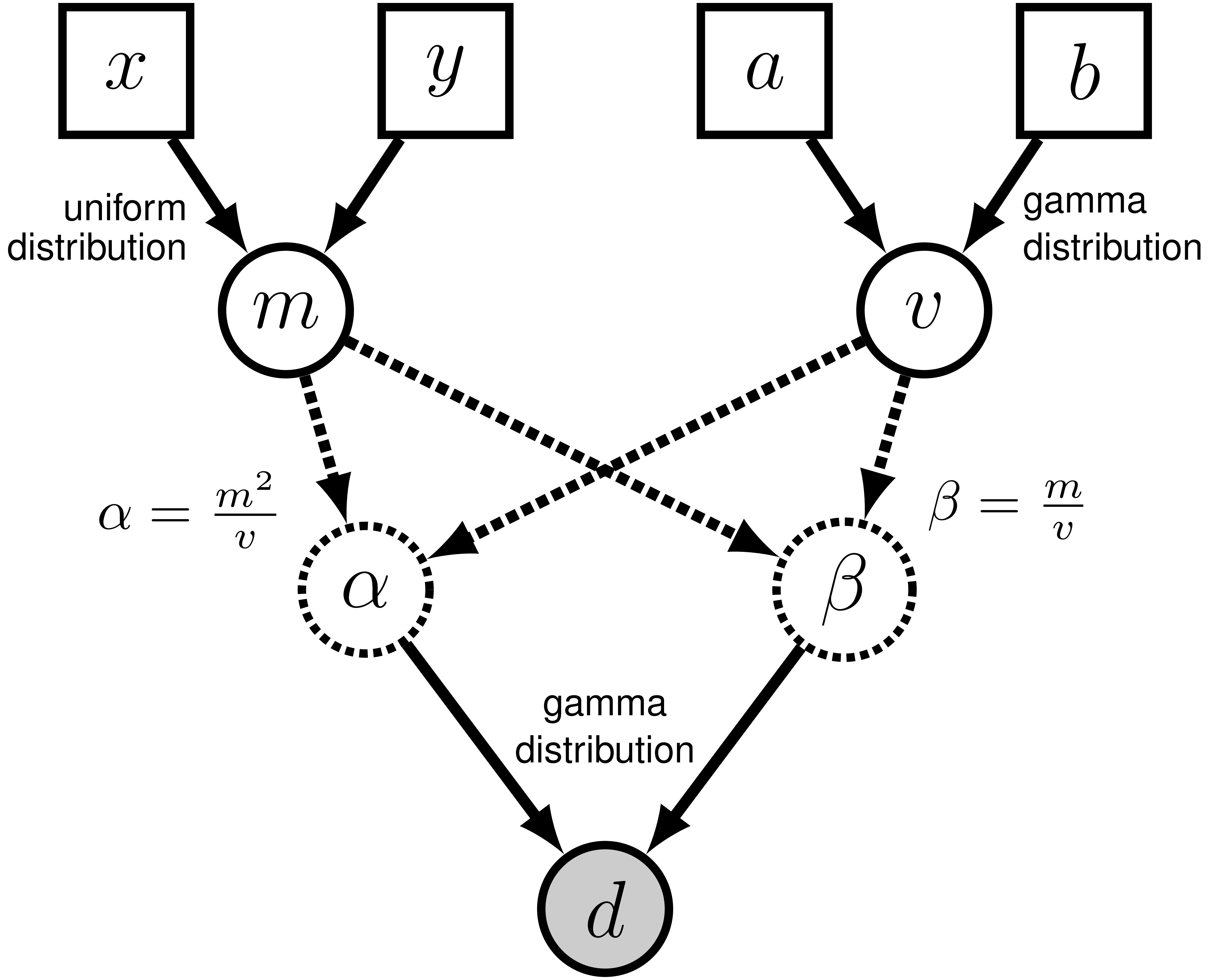
$\mathbb{E}(m) = 30$

$v \sim \text{Gamma}(a, b)$

$a = 20$

$b = 2$

$\mathbb{E}(v) = 10$



RevBayes Demo: Archery

The Rev language specifying a hierarchical model on shot accuracy based on 1 new observation

```
mean ~ dnUnif(10, 50)
var ~ dnGamma(20, 2)

alpha := (mean * mean) / var
beta := mean / var

observed_shot = 35.21

d ~ dnGamma(alpha, beta)
d.clamp(observed_shot)

d.lnProbability()
```

depends on initial value of mean & var

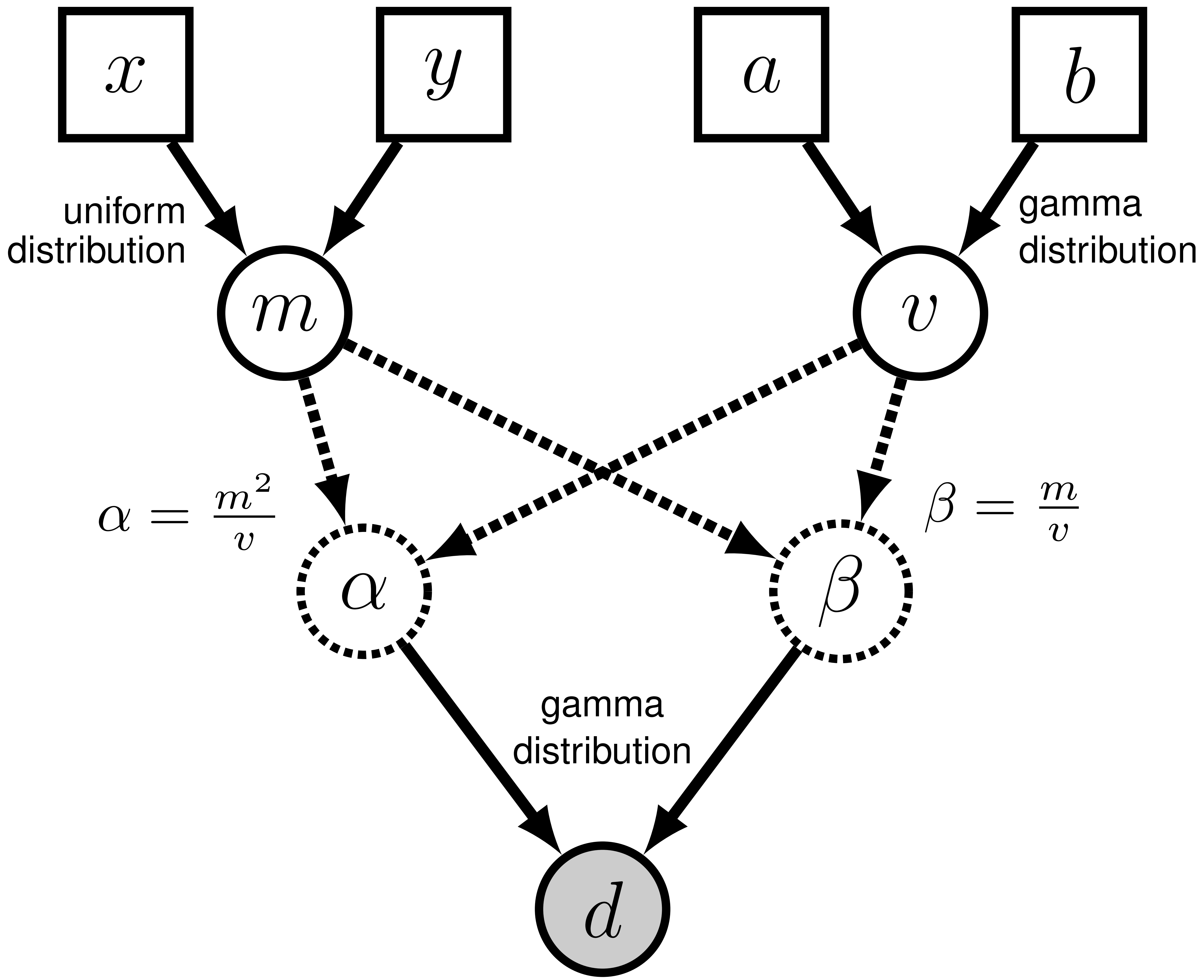
Priors: Archery

Now that we have a defined model, how do we estimate the posterior probability density?

$$m \sim \text{Uniform}(x, y)$$

$$v \sim \text{Gamma}(a, b)$$

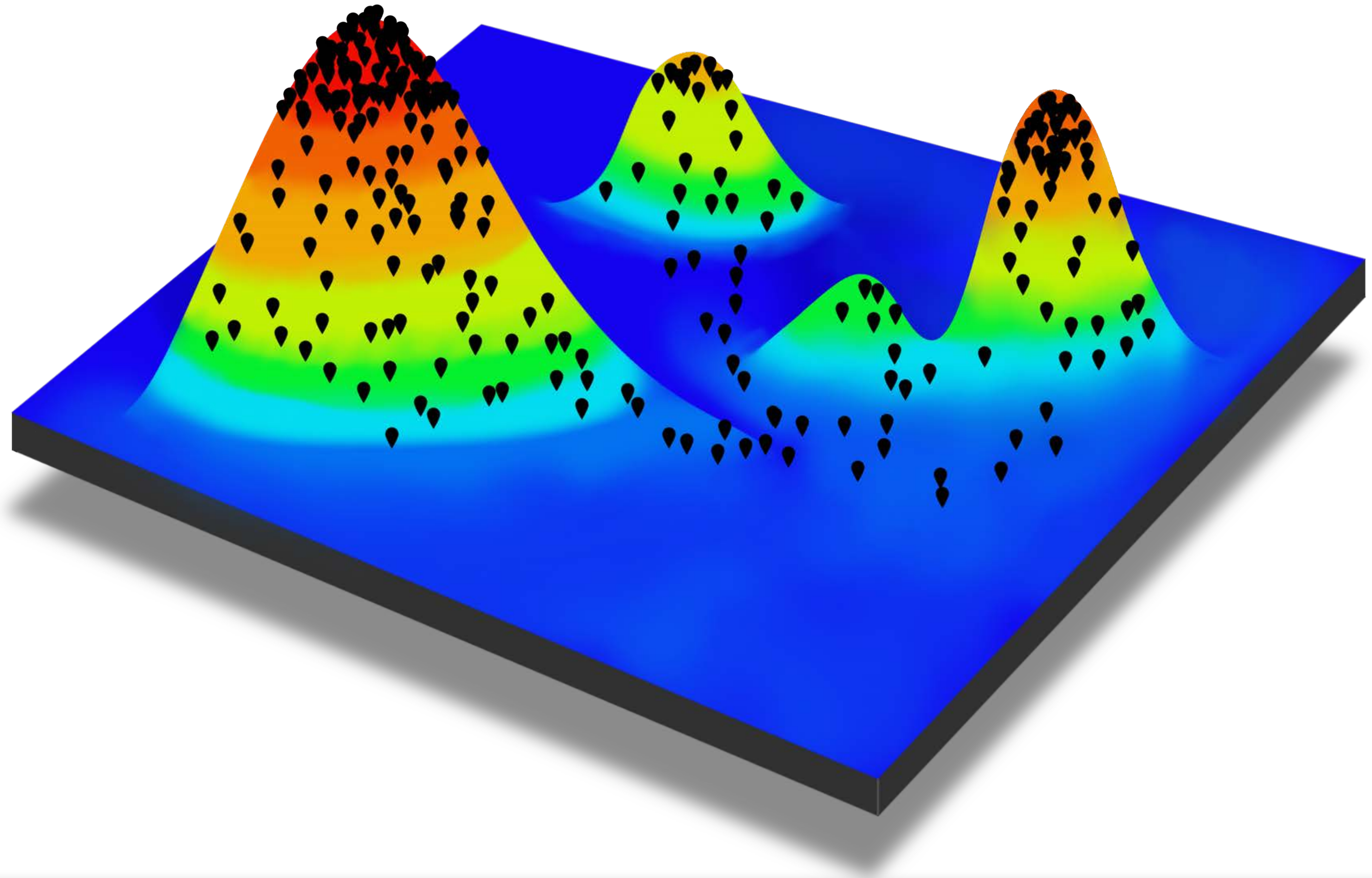
$$d \sim \text{Gamma}(\alpha, \beta)$$



$$f(m, v \mid d, a, b, x, y) \propto f(d \mid , \alpha = \frac{m^2}{v}, \beta = \frac{m}{v}) f(m \mid x, y) f(v \mid a, b)$$

Markov Chain Monte Carlo

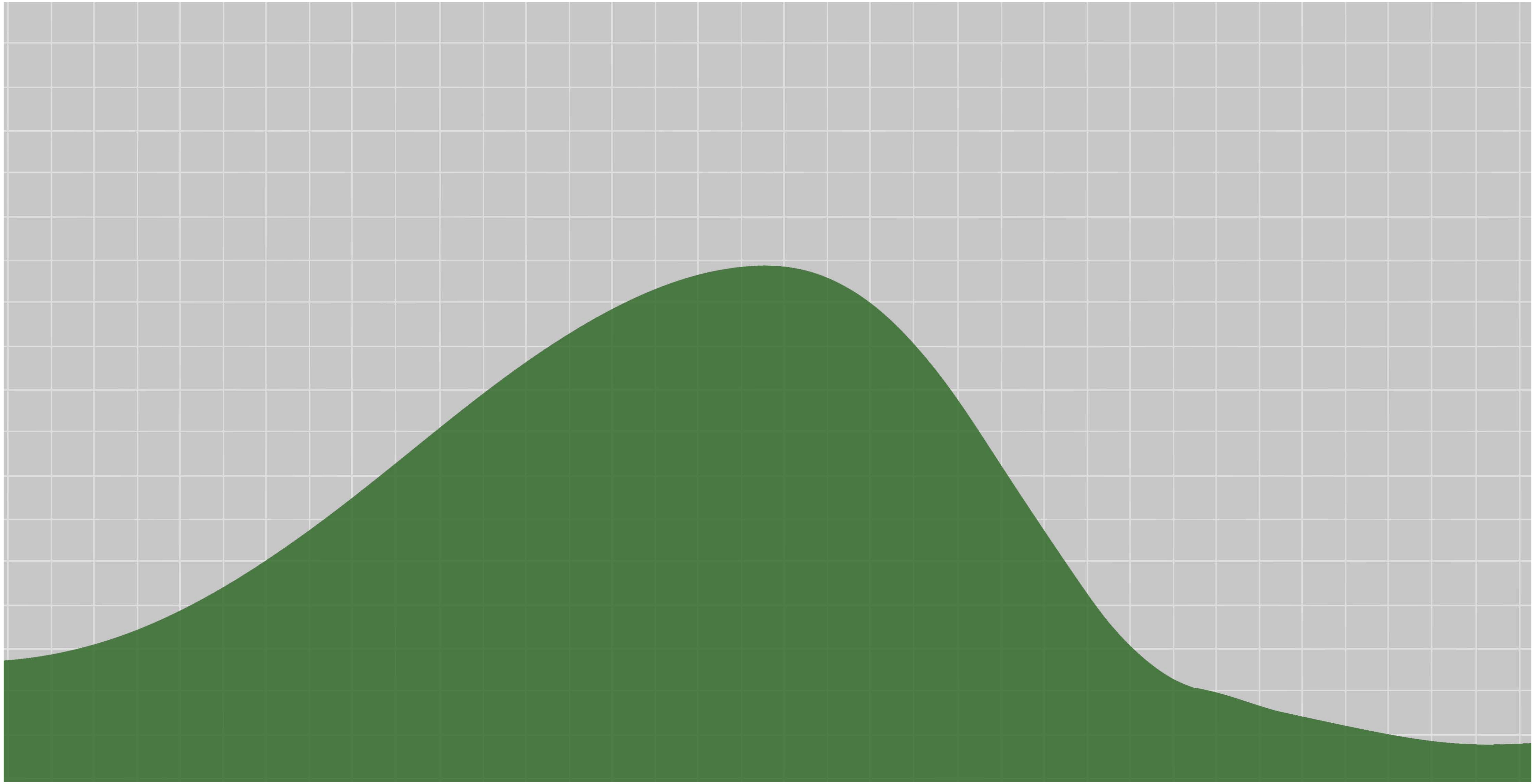
An algorithm for approximating the posterior distribution



Metropolis, et al. 1953. Equations of state calculations by fast computing machines. J. Chem. Phys.

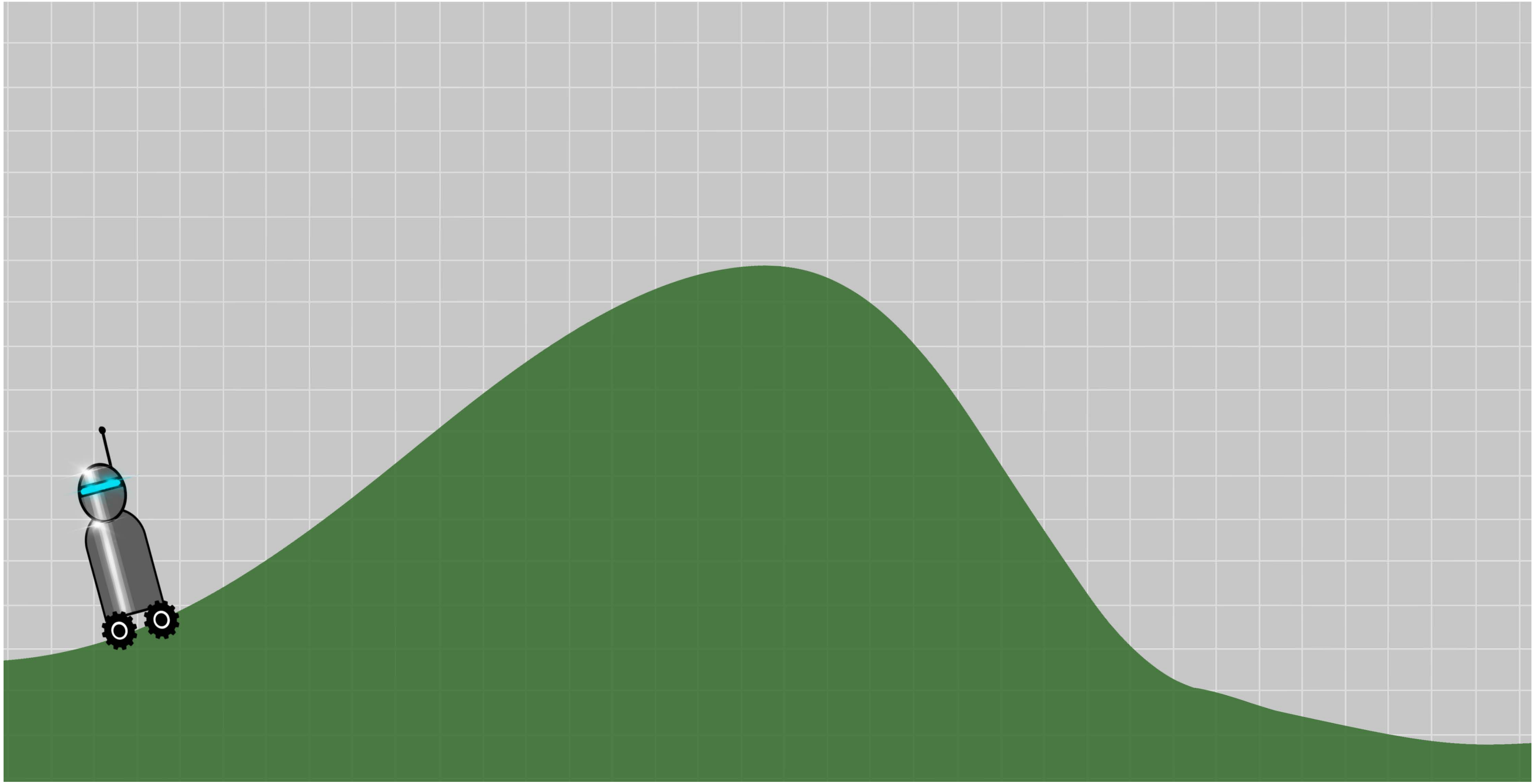
Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika.

MCMC Robot Rules



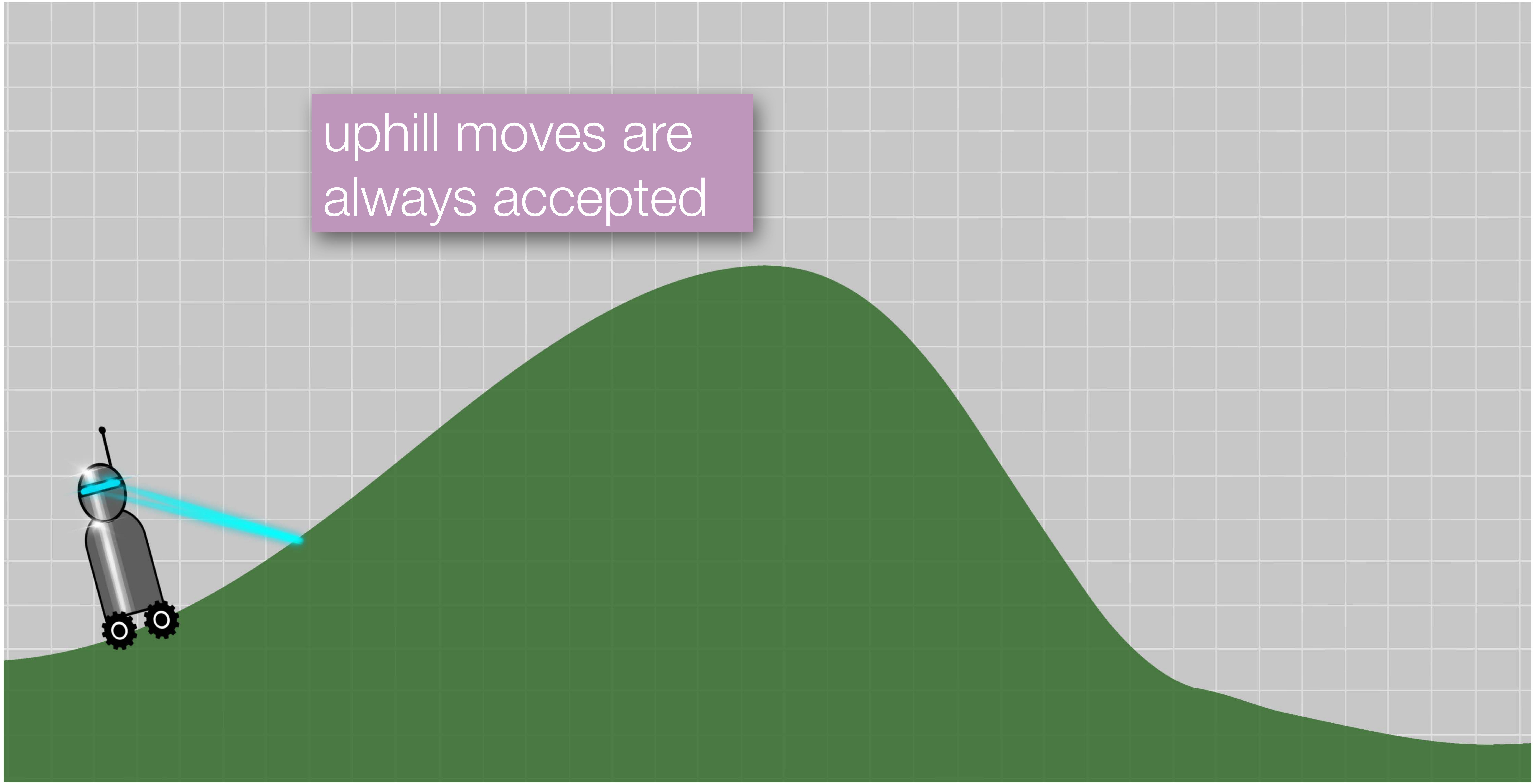
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

MCMC Robot Rules



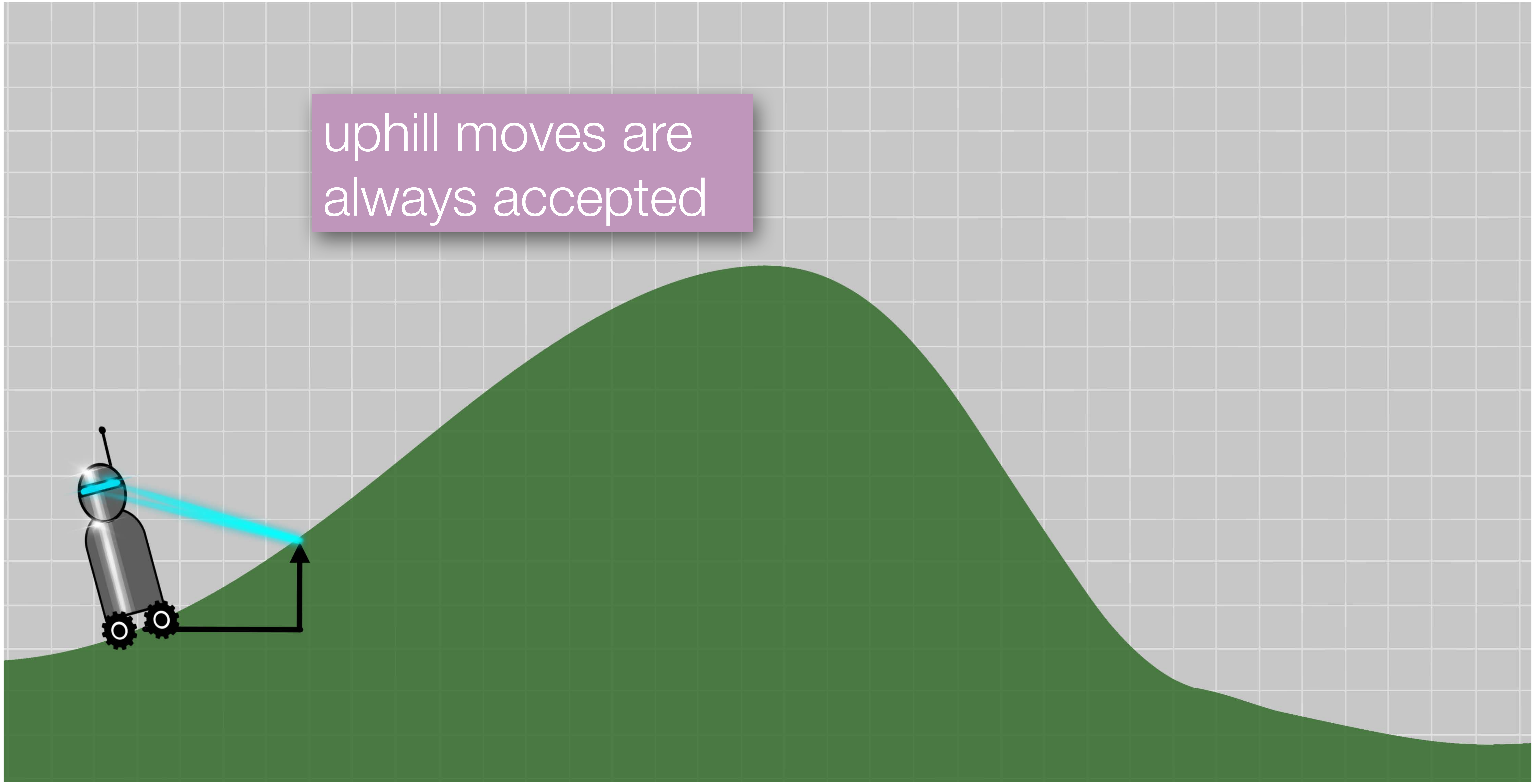
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

MCMC Robot Rules

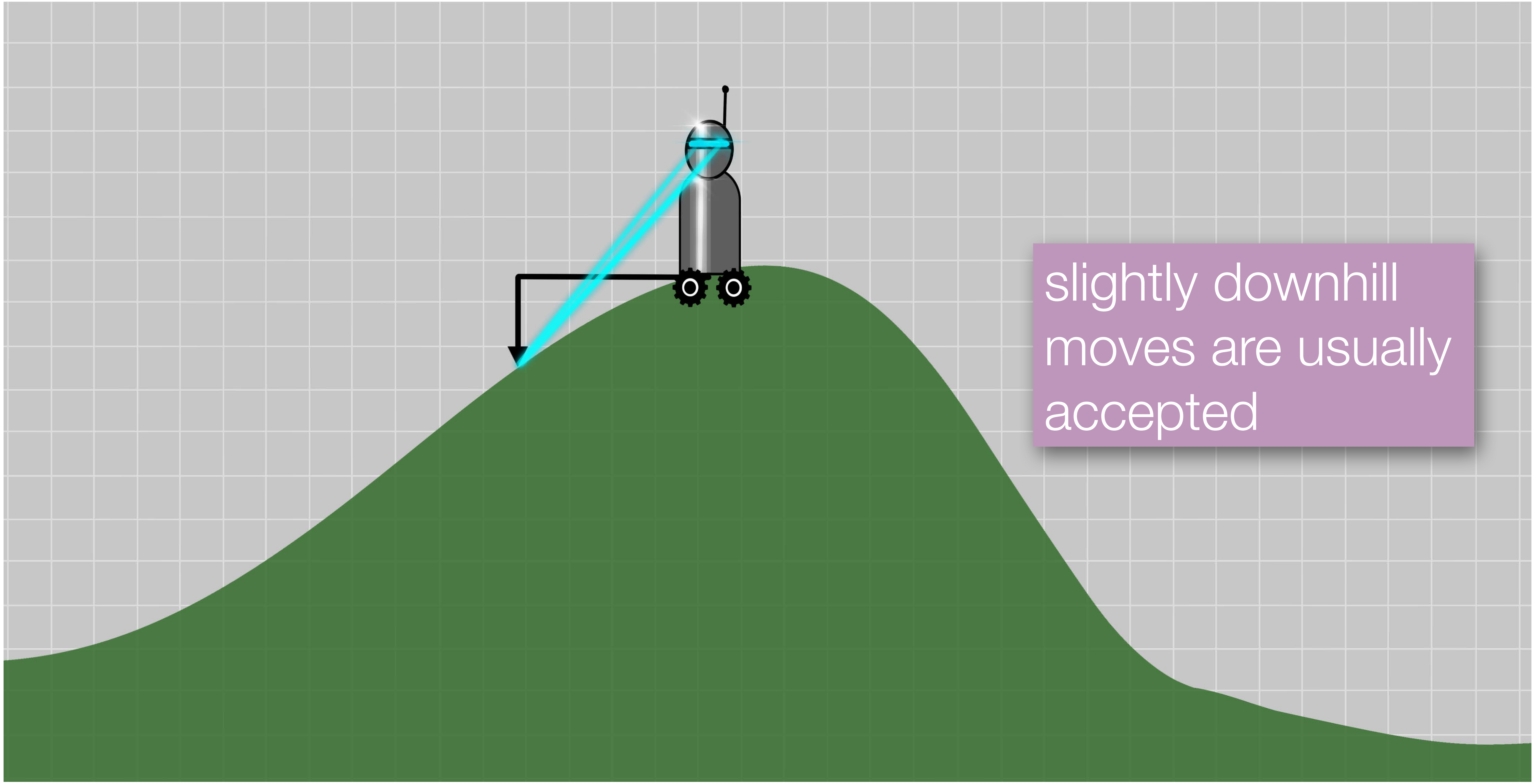


(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

MCMC Robot Rules

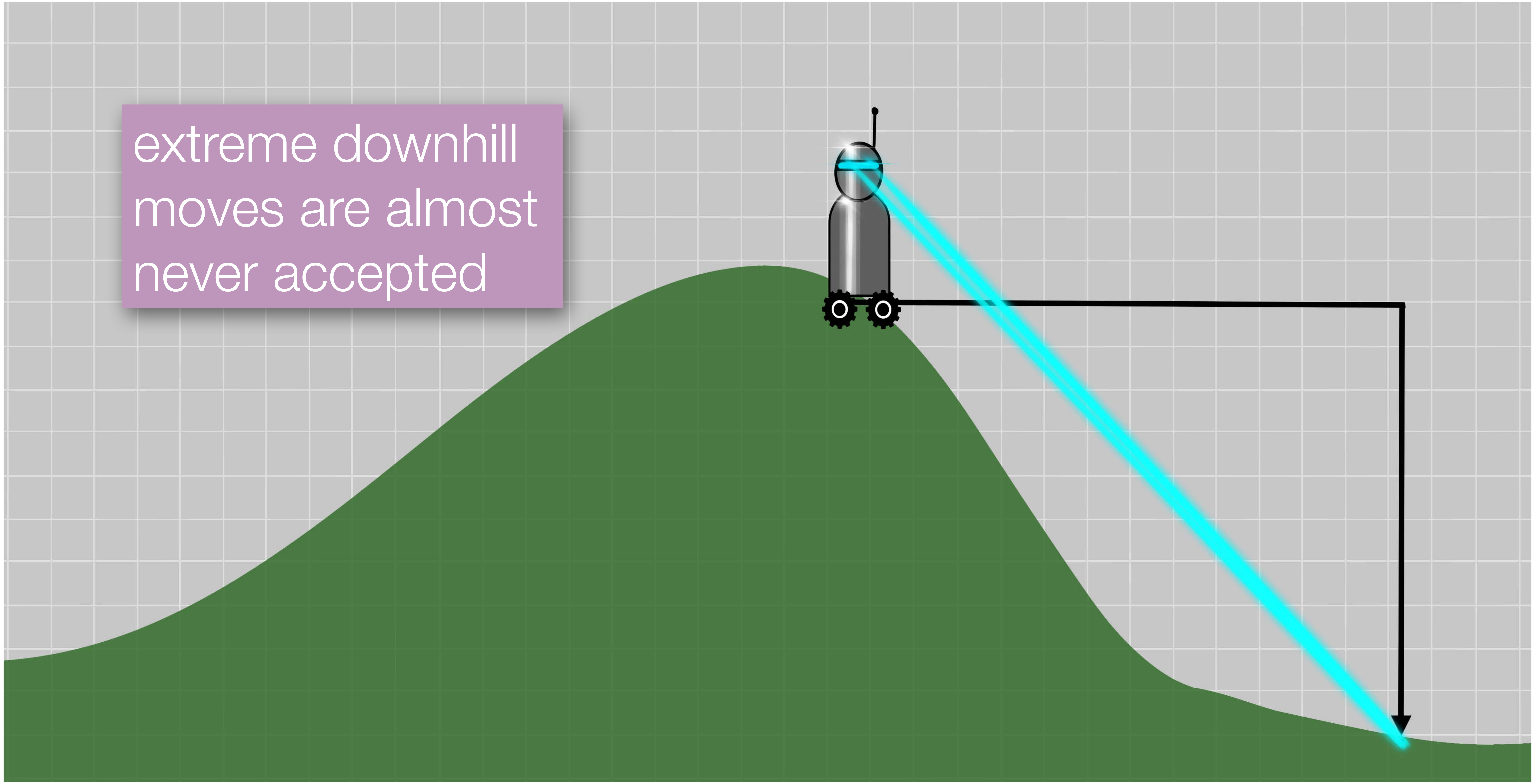


MCMC Robot Rules



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

MCMC Robot Rules



Actual Rules (Metropolis Algorithm)

Metropolis et al. 1953. Equation of state calculations by fast computing machines. *J. Chem. Physics.*

when the robot makes a move it evaluates the new value compared to its current state

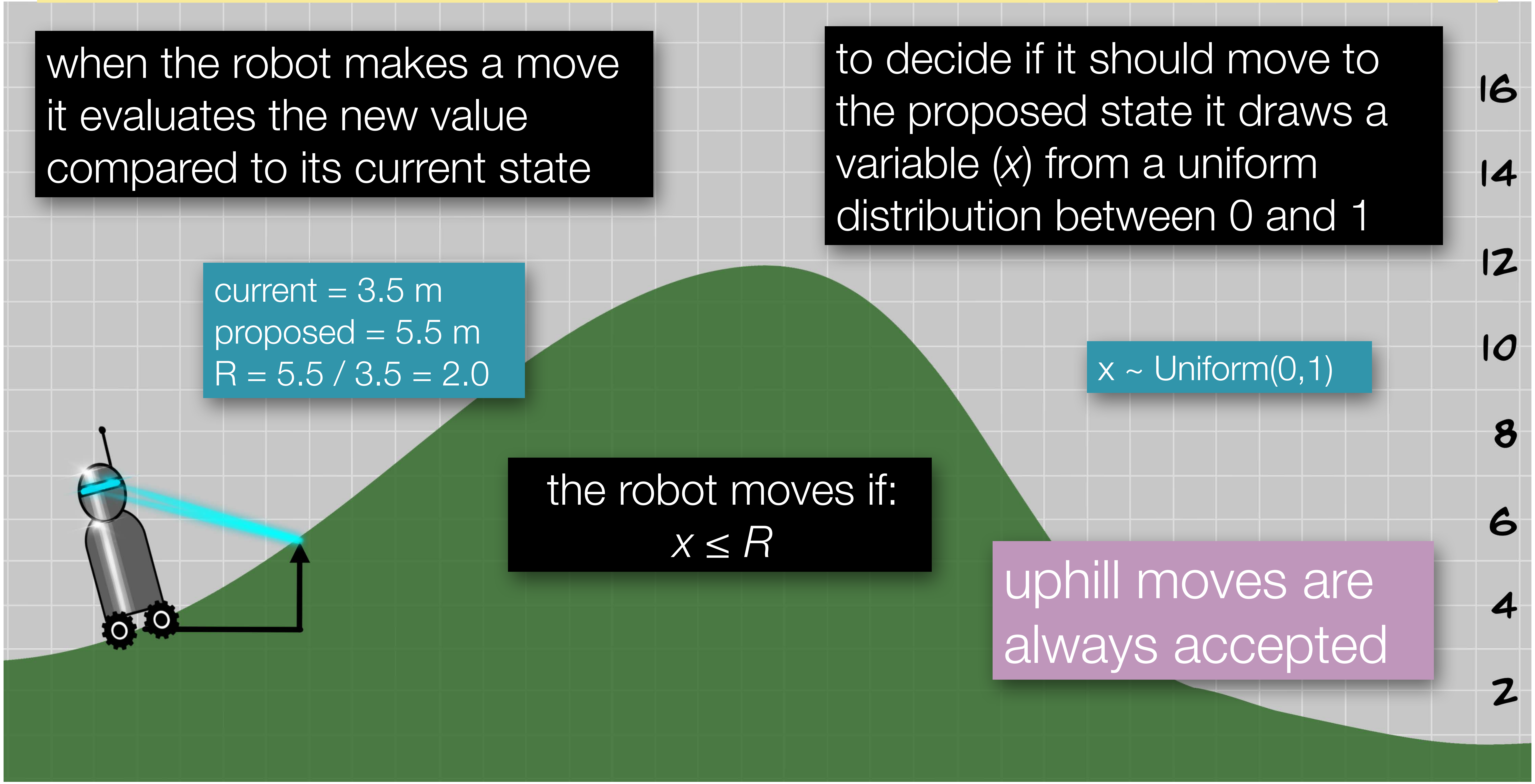
to decide if it should move to the proposed state it draws a variable (x) from a uniform distribution between 0 and 1

current = 3.5 m
proposed = 5.5 m
R = 5.5 / 3.5 = 2.0

x ~ Uniform(0,1)

the robot moves if:
$x \leq R$

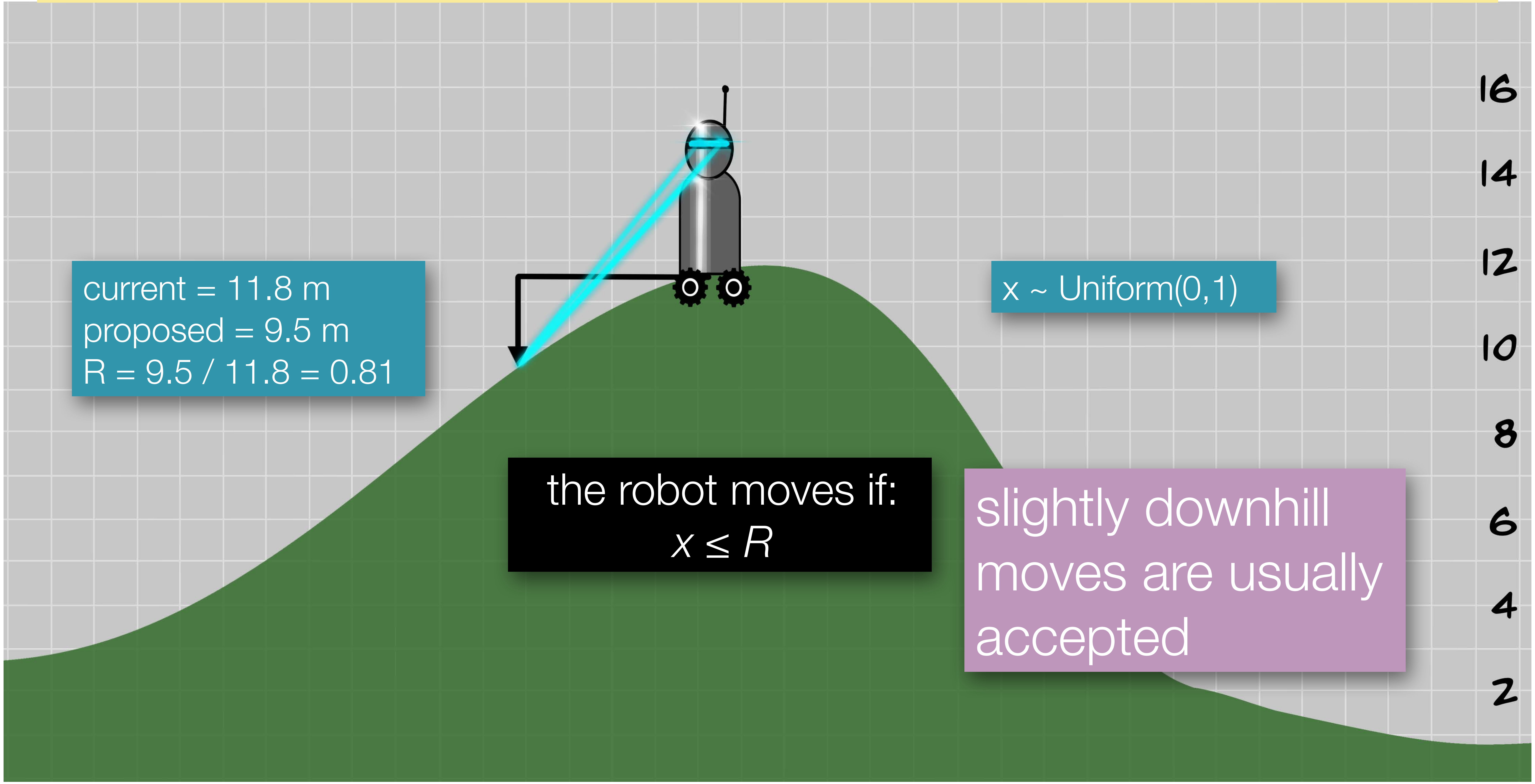
uphill moves are always accepted



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Actual Rules (Metropolis Algorithm)

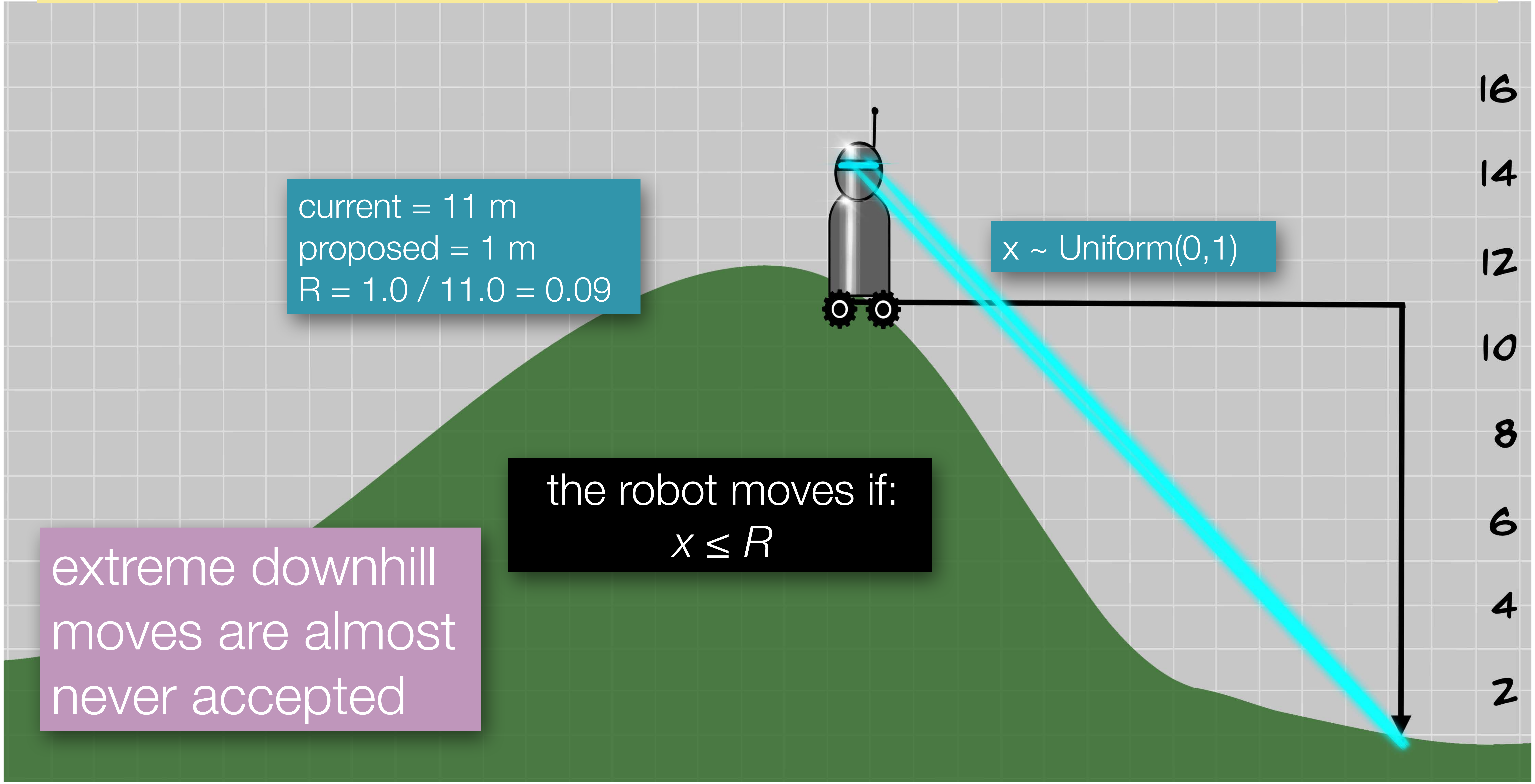
Metropolis et al. 1953. Equation of state calculations by fast computing machines. *J. Chem. Physics.*



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Actual Rules (Metropolis Algorithm)

Metropolis et al. 1953. Equation of state calculations by fast computing machines. *J. Chem. Physics.*



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Bayes Rule

POSTERIOR PROBABILITY

LIKELIHOOD

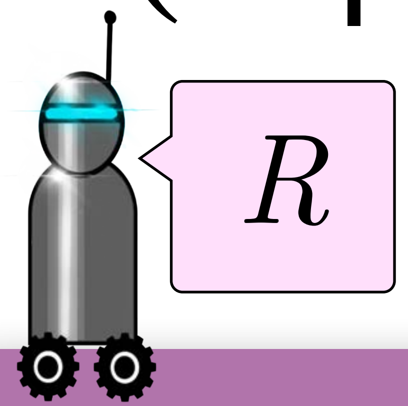
PRIOR PROBABILITY

$$\underbrace{\Pr(\theta \mid D)}_{\text{posterior probability}} = \frac{\overbrace{\Pr(D \mid \theta)}^{\text{likelihood}}\ \overbrace{\Pr(\theta)}^{\text{prior probability}}}{\underbrace{\sum_{\theta} \Pr(D \mid \theta)\Pr(\theta)}_{\text{marginal probability of the data}}}$$

MARGINAL PROBABILITY
OF THE DATA

Canceling Out the Marginal Likelihood

$$\frac{\Pr(\theta^* \mid D)}{\Pr(\theta \mid D)} = \frac{\dfrac{\Pr(D \mid \theta^*)\,\Pr(\theta^*)}{\cancel{\Pr(D)}}}{\dfrac{\Pr(D \mid \theta)\,\Pr(\theta)}{\cancel{\Pr(D)}}} = \frac{\Pr(D \mid \theta^*)\,\Pr(\theta^*)}{\Pr(D \mid \theta)\,\Pr(\theta)}$$



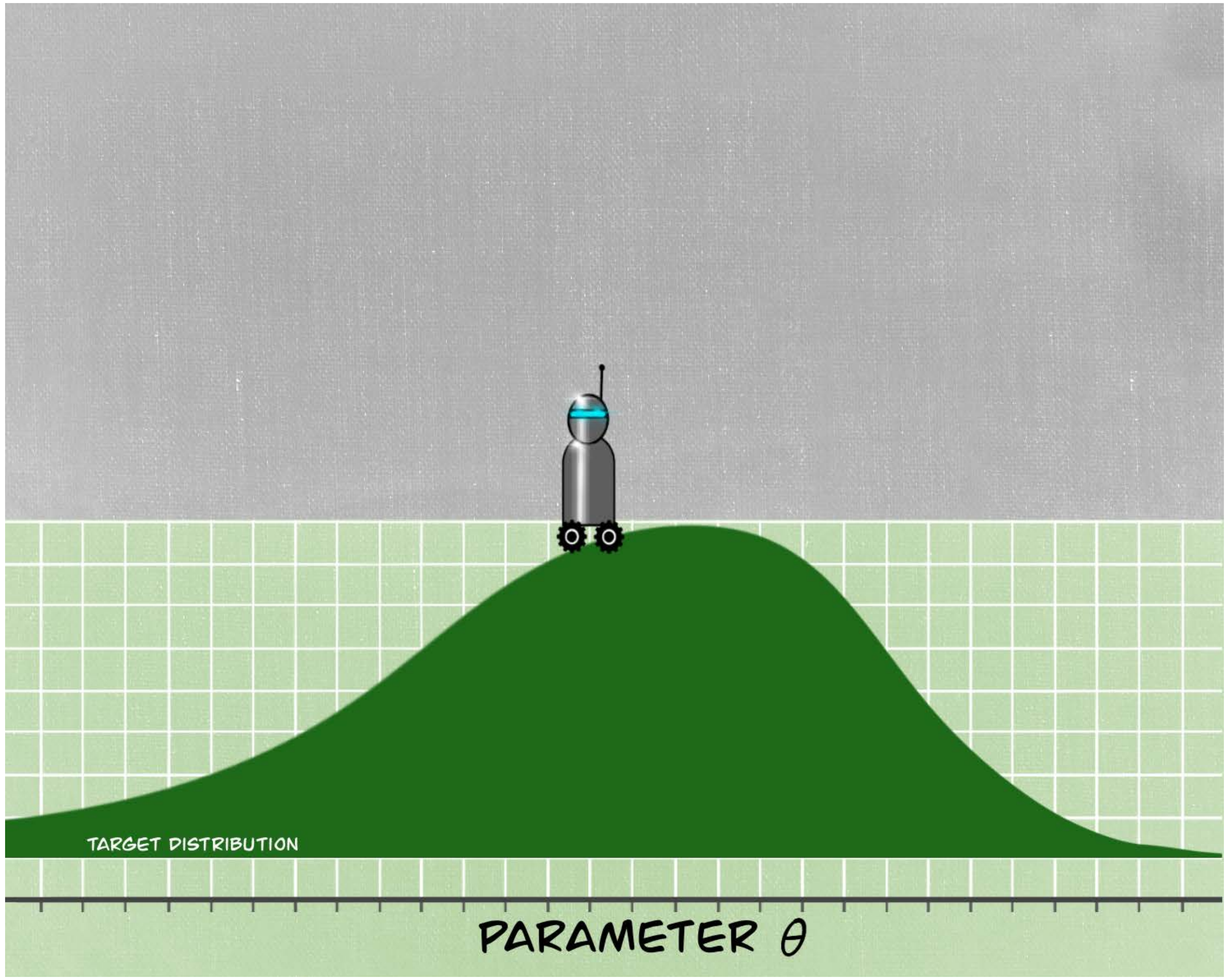
Posterior odds

Bayes Rule!

Likelihood ratio

Prior odds

Target and Proposal Distributions

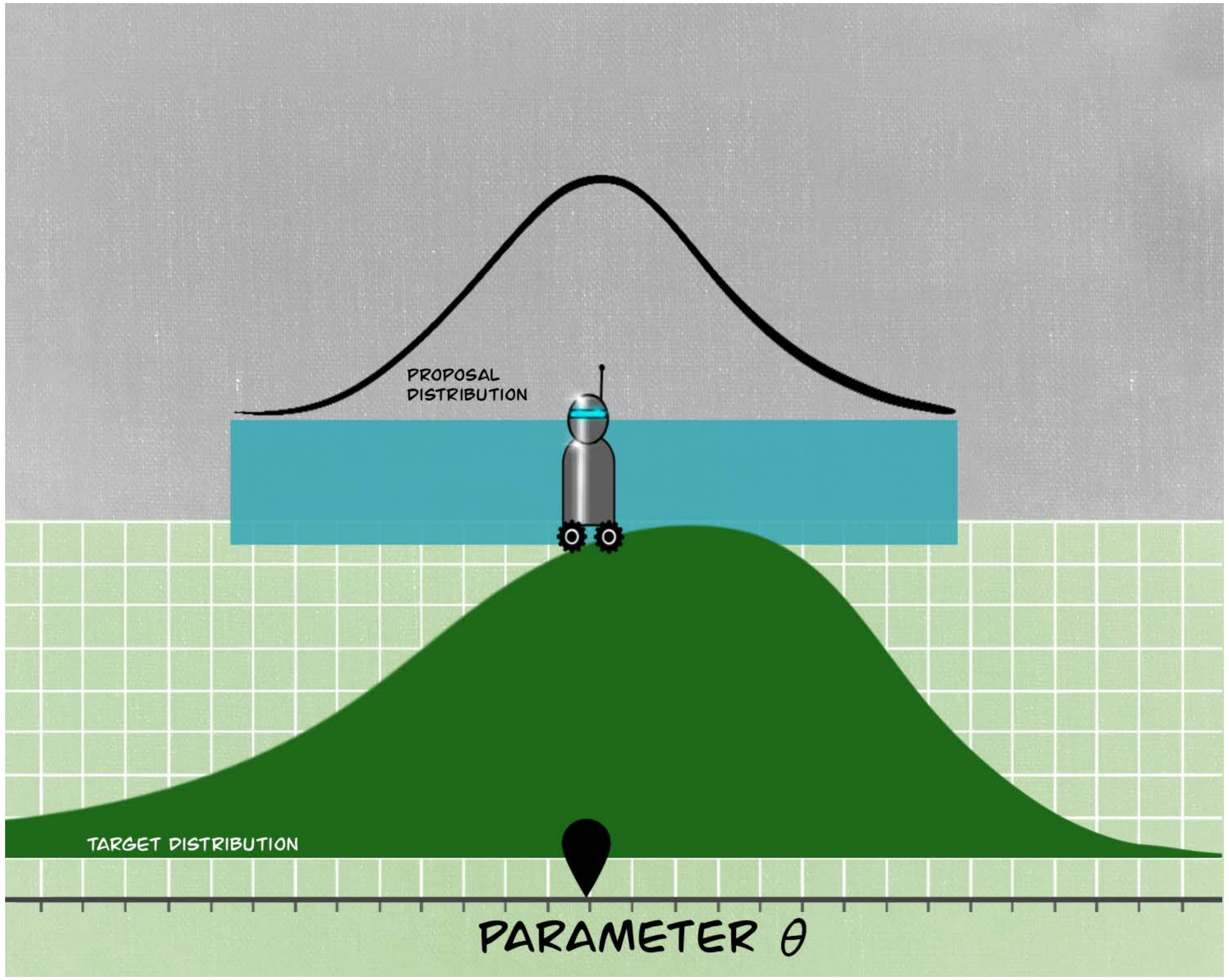


the target distribution is the landscape mapped by the robot

typically, this is the posterior distribution

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Target and Proposal Distributions

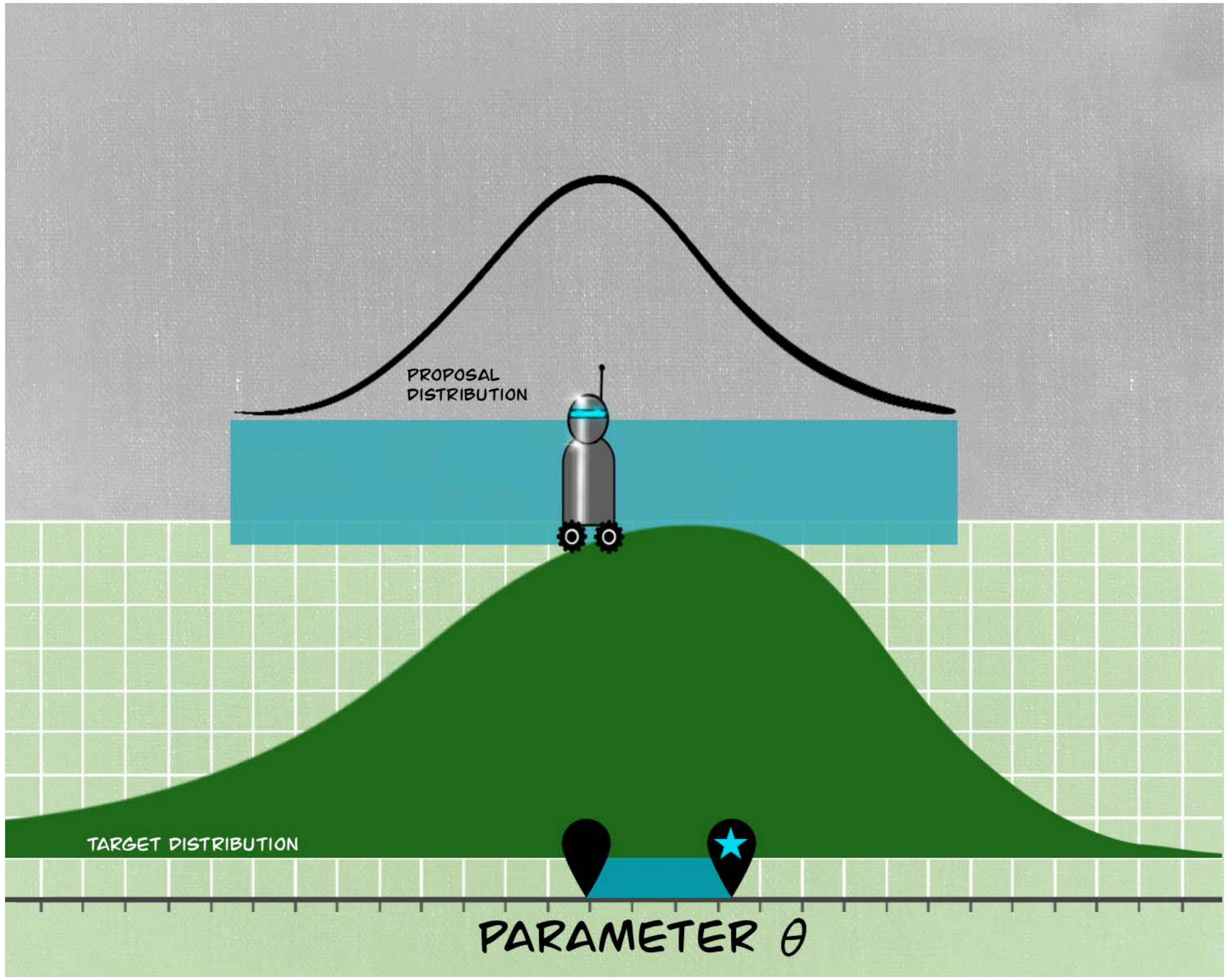


the proposal distribution is separate from the target distribution

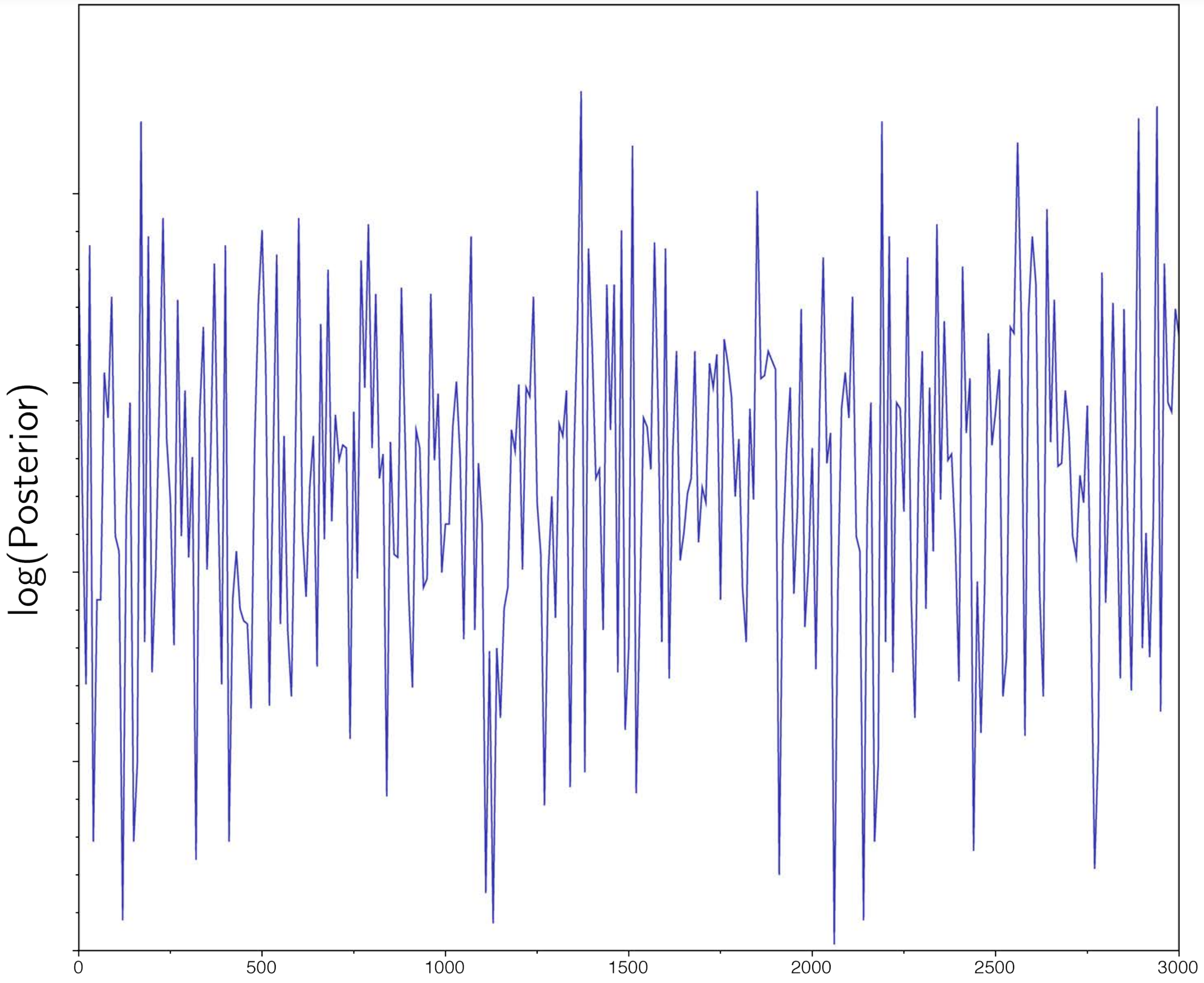
the robot uses the proposal distribution to choose the next spot to move

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Target and Proposal Distributions



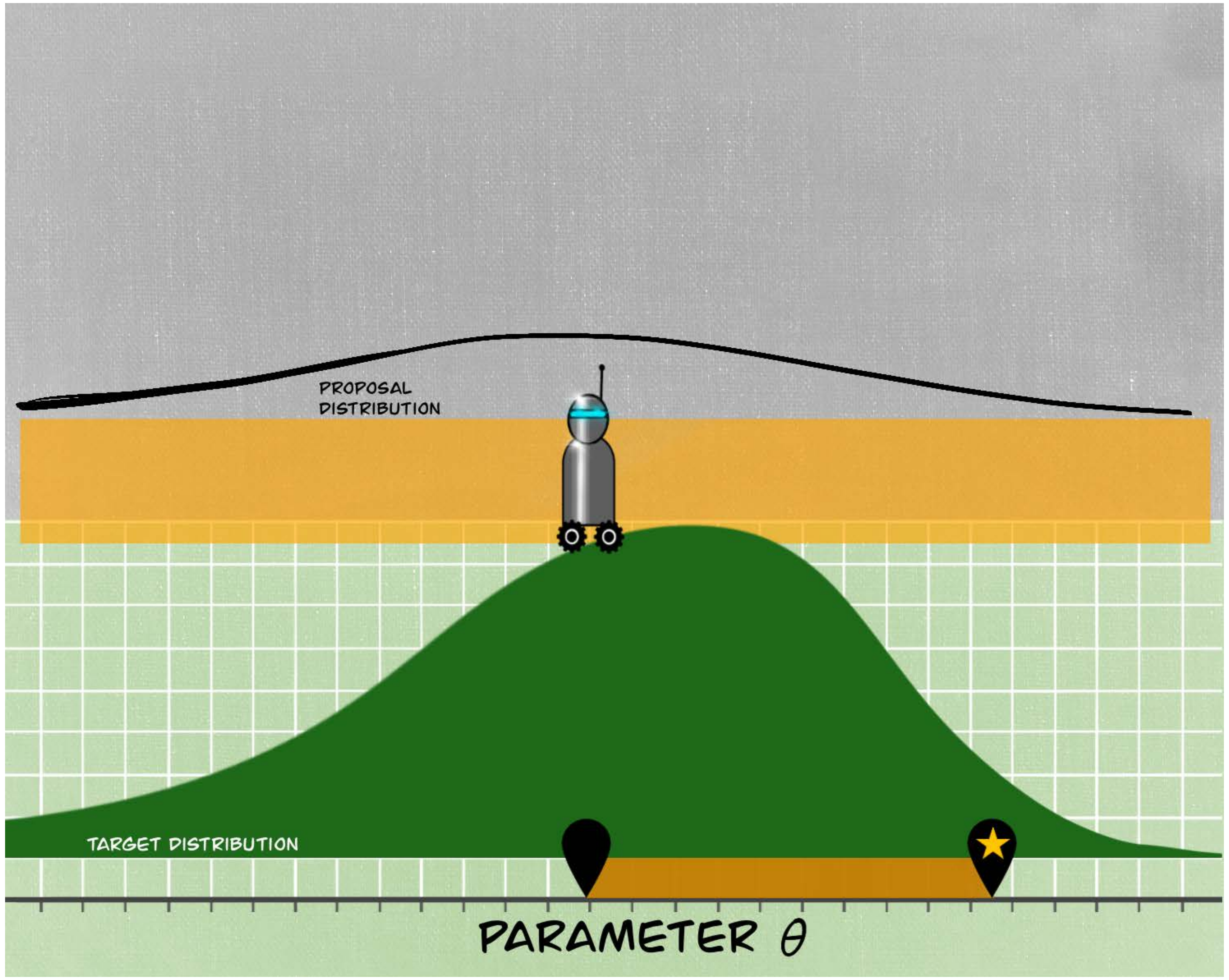
a good proposal distribution samples the target distribution effectively (i.e., "good mixing")



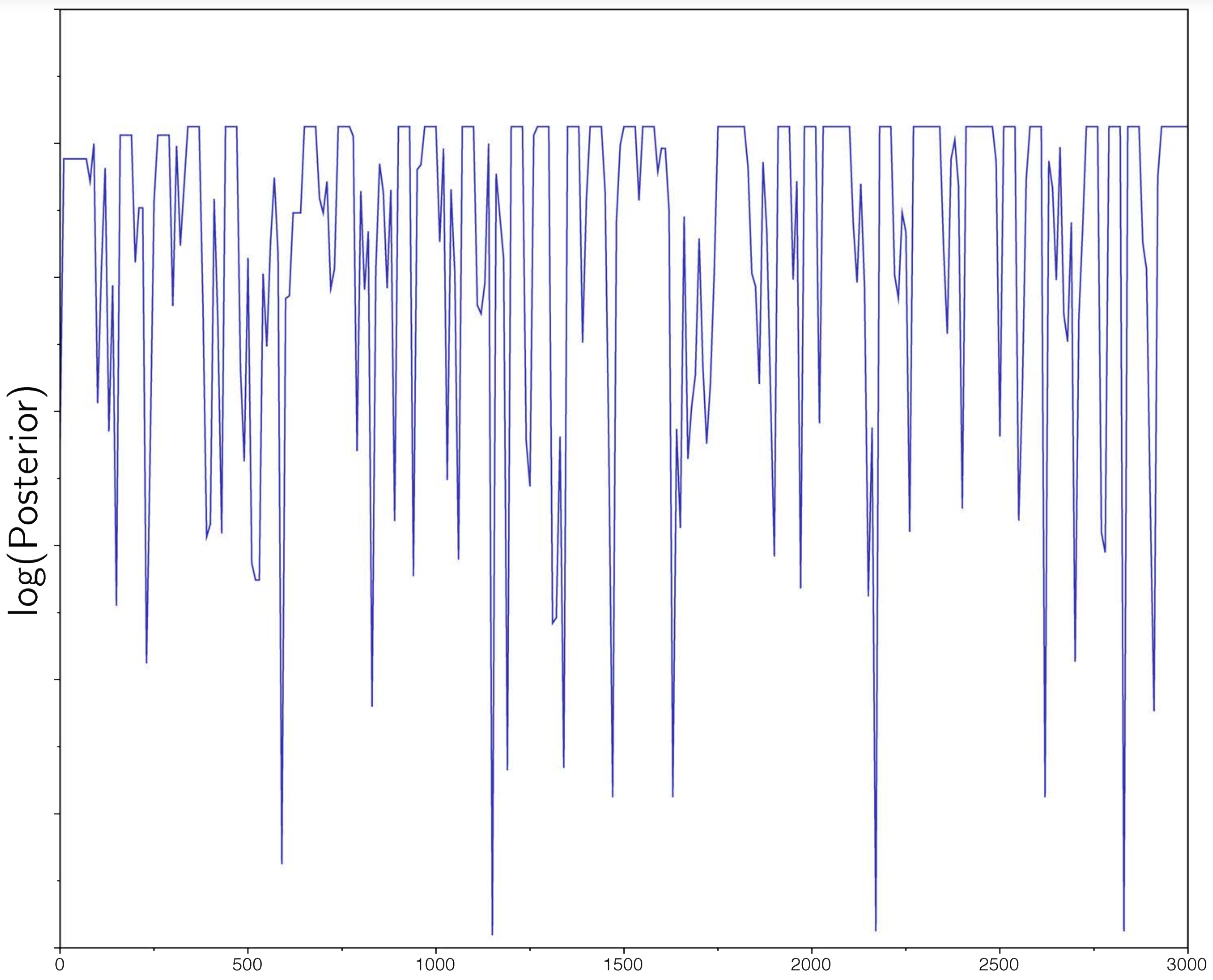
a trace plot of the sampled parameter values looks like white noise

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Target and Proposal Distributions



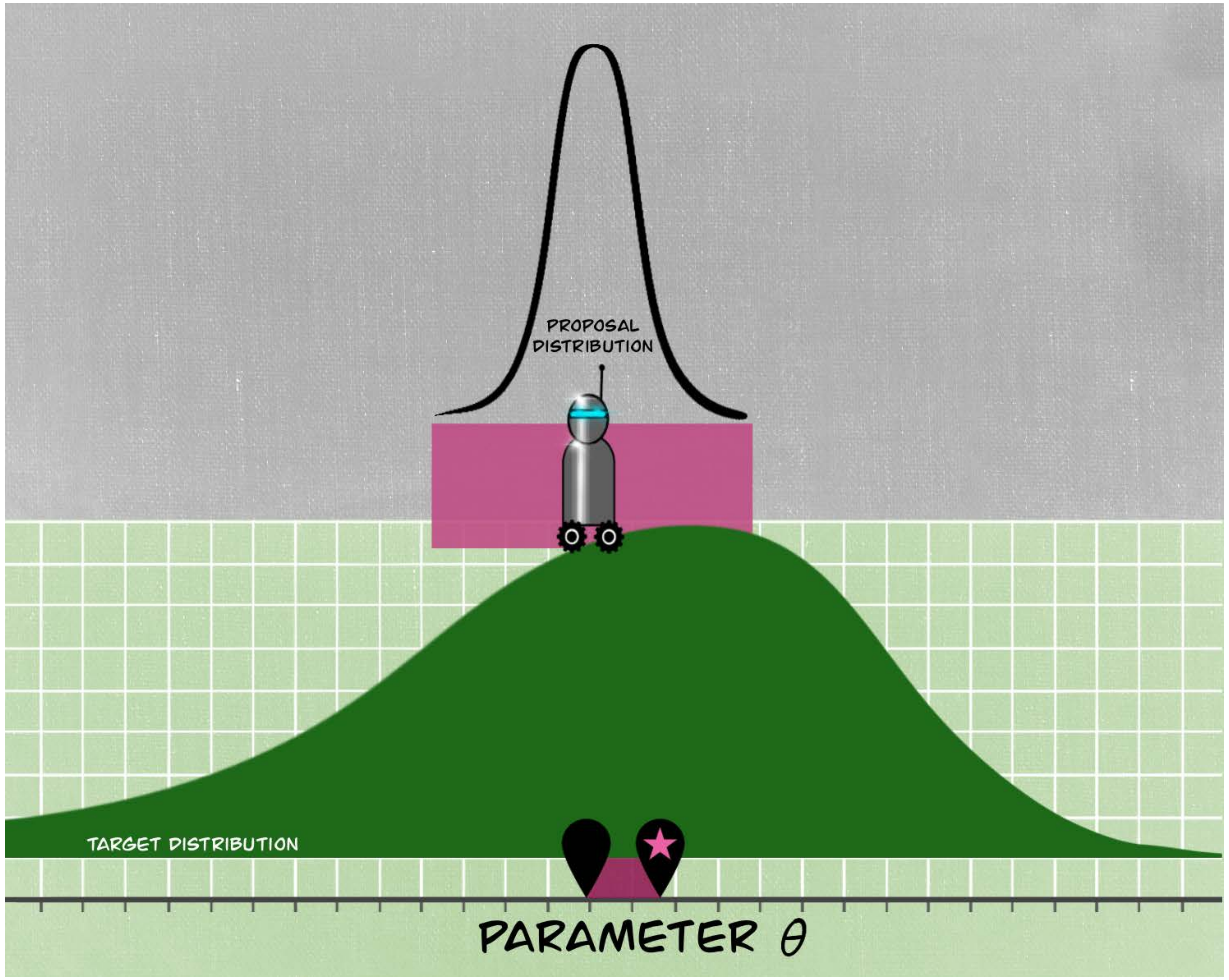
an overly bold proposal results in many rejected moves



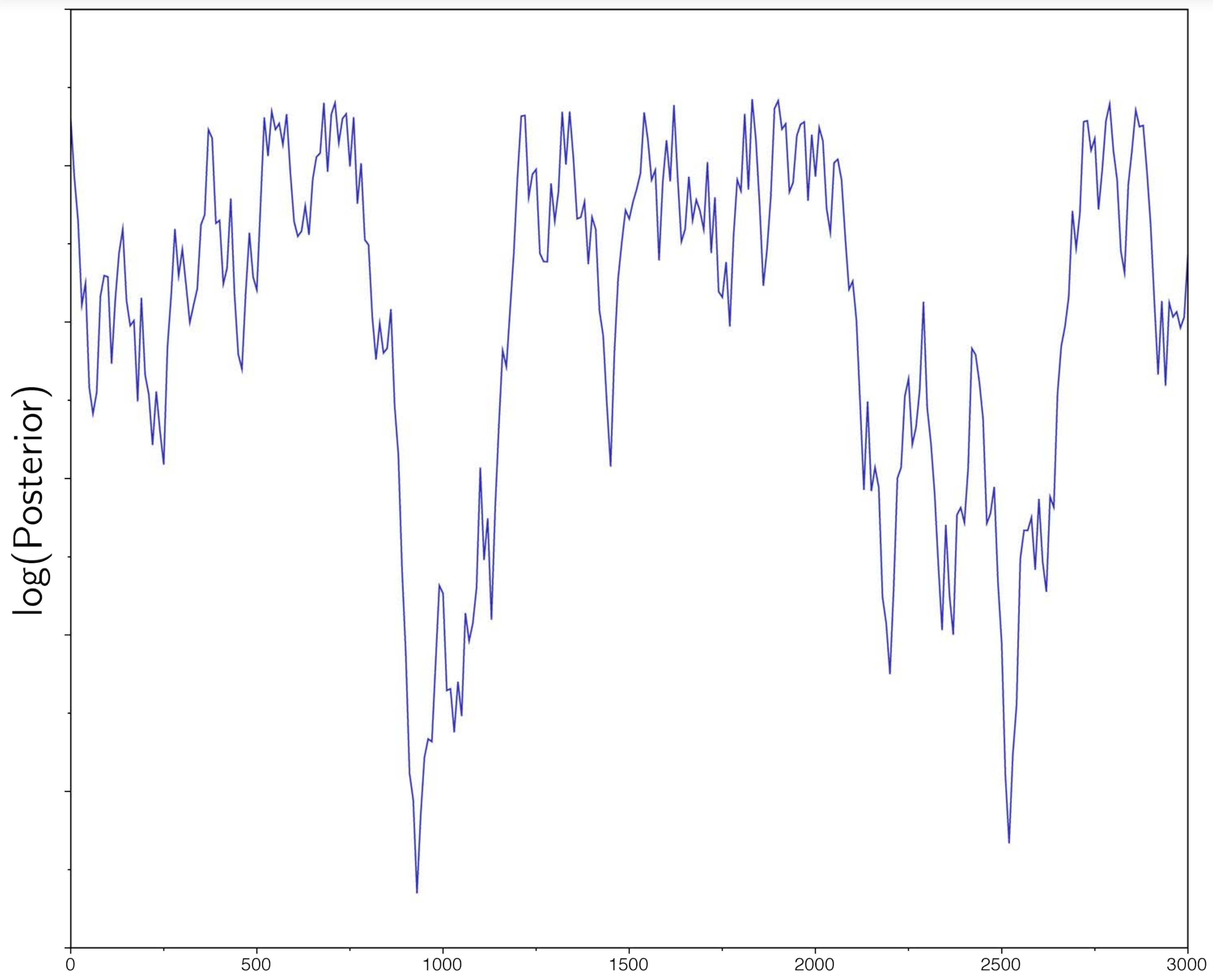
this causes the robot to get stuck, seen as plateaus in the trace plot

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Target and Proposal Distributions



a proposal distribution that only allows for baby steps results in lots of accepted moves



this causes big waves in the trace plot as the robot takes small incremental samples

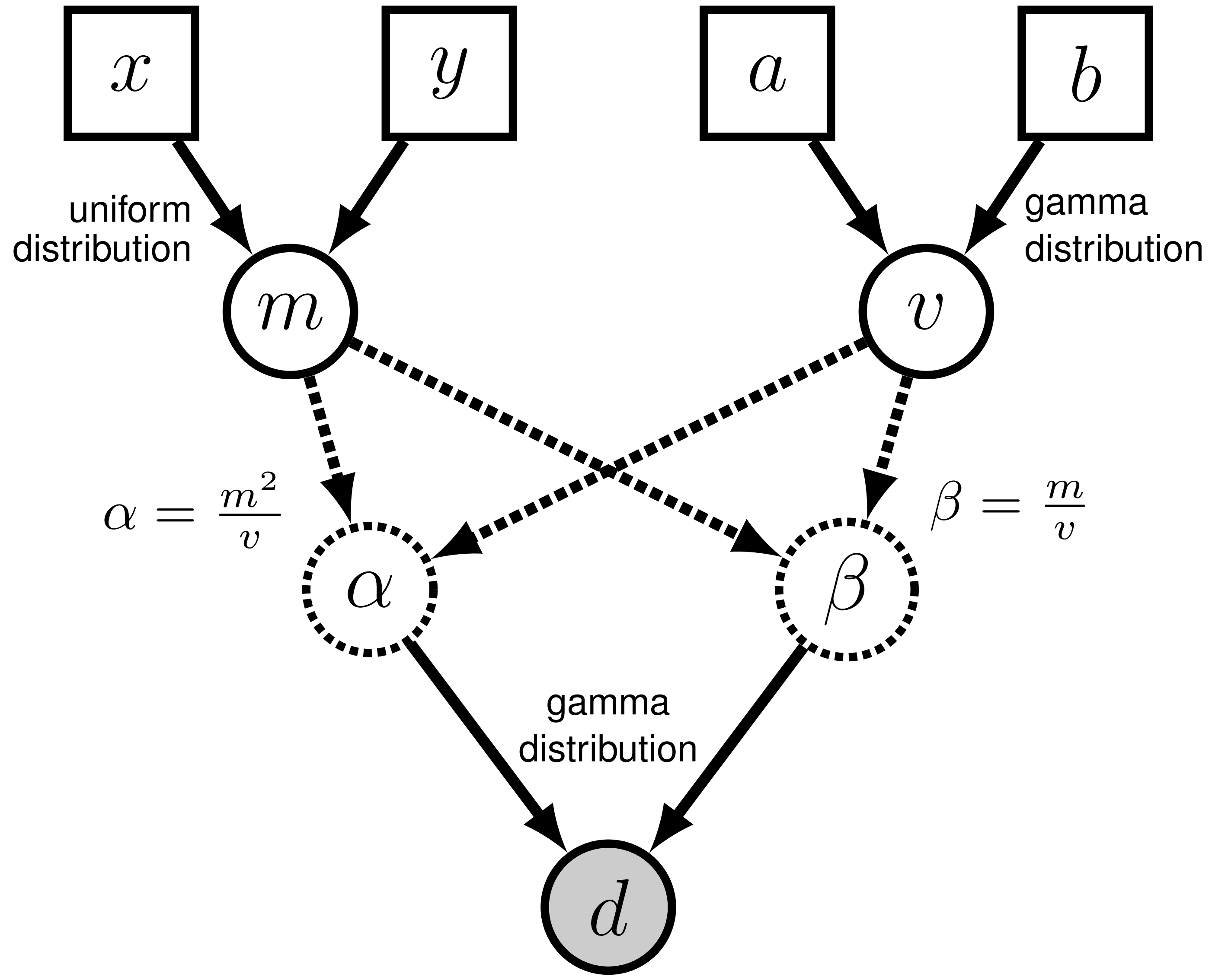
(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

RevBayes Demo: Archery

We can use MCMC to estimate m and v

First, let's generate our observed data using simulation

```
true_accuracy = 35.0
true_variance = 4.0
true_alpha = (true_accuracy^2) / true_variance
true_beta = true_accuracy / true_variance

num_shots = 6
observed_shots = rgamma(num_shots, true_alpha, true_beta)
```



The values in `observed_shots` are data generated from the underlying distribution

RevBayes Demo: Archery

Now we can specify the model for our new observations

```
mean ~ dnUnif(10,40)
var ~ dnGamma(20,2)

alpha := (mean * mean) / var
beta := mean / var

for(i in 1:num_shots){
    d[i] ~ dnGamma(alpha,beta)
    d[i].clamp(observed_shots[i])
}
```

RevBayes Demo: Archery

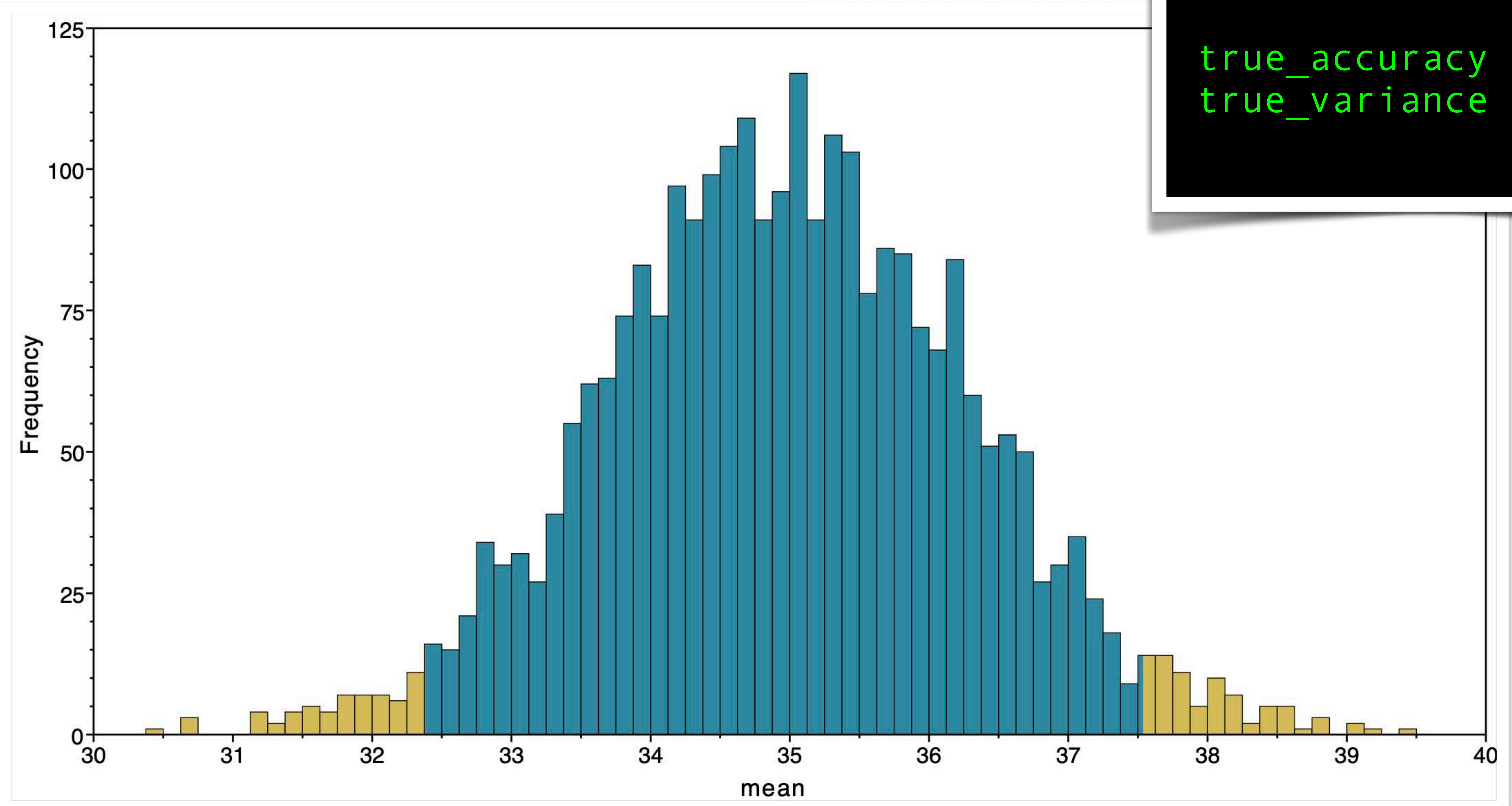
The Rev language specifying the MCMC sampler for the hierarchical model on archery accuracy

```
mymodel = model(beta)

moves[1] = mvSlide(mean, delta=1.0, weight=3.0)
moves[2] = mvScale(var, lambda=1.0, weight=3.0)

monitors[1] = mnModel(filename="archery_mcmc_1.log", printgen=10)
monitors[2] = mnScreen(printgen=1000, mean, var)

mymcmc = mcmc(mymodel, monitors, moves)

mymcmc.burnin(generations=10000,tuningInterval=1000)

mymcmc.run(generations=30000)
```

MCMC screen output

RevBayes Demo: Archery

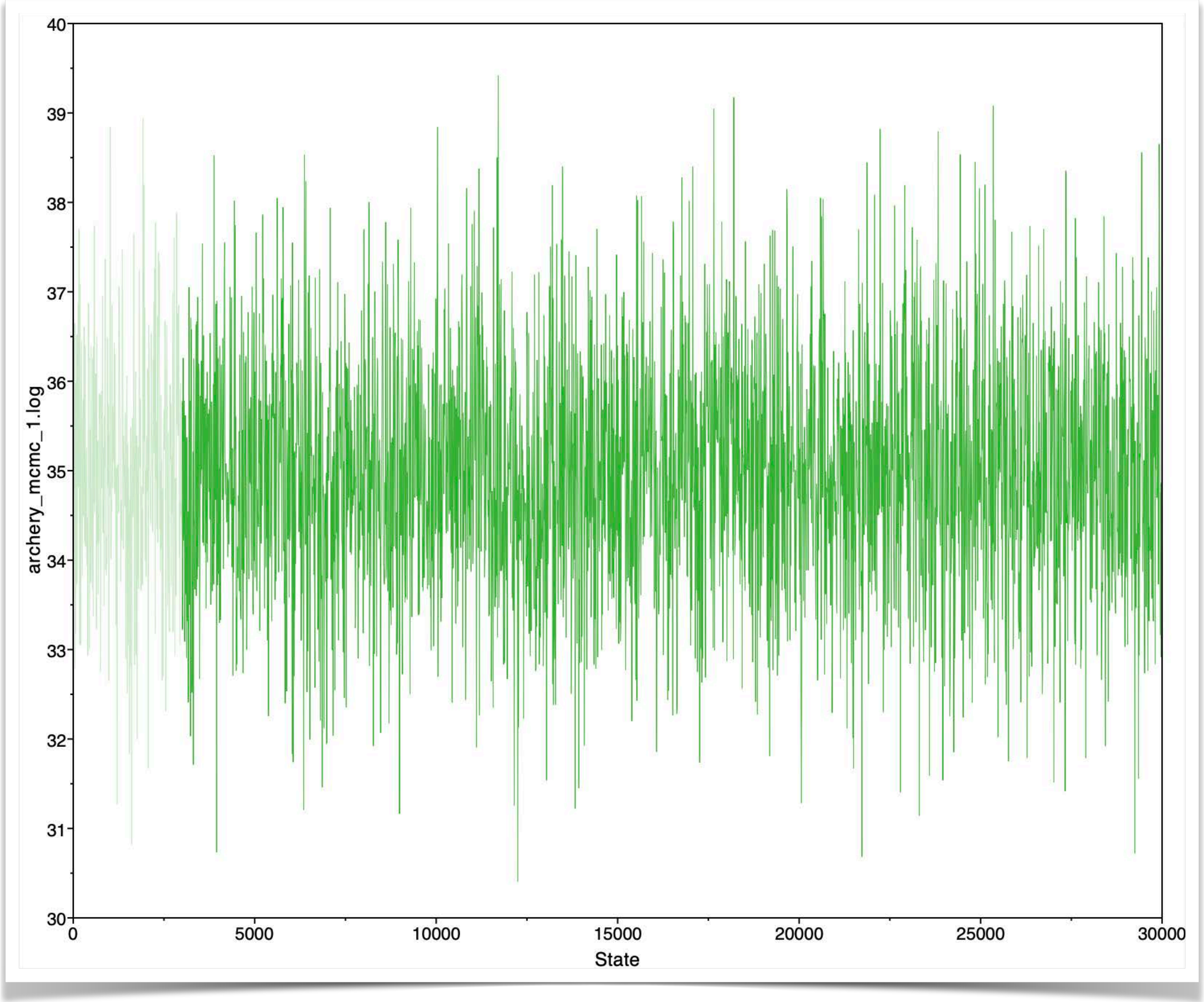
Summary of the MCMC sample for the mean distance from the target center




```
true_accuracy = 35.0
true_variance = 4.0
```

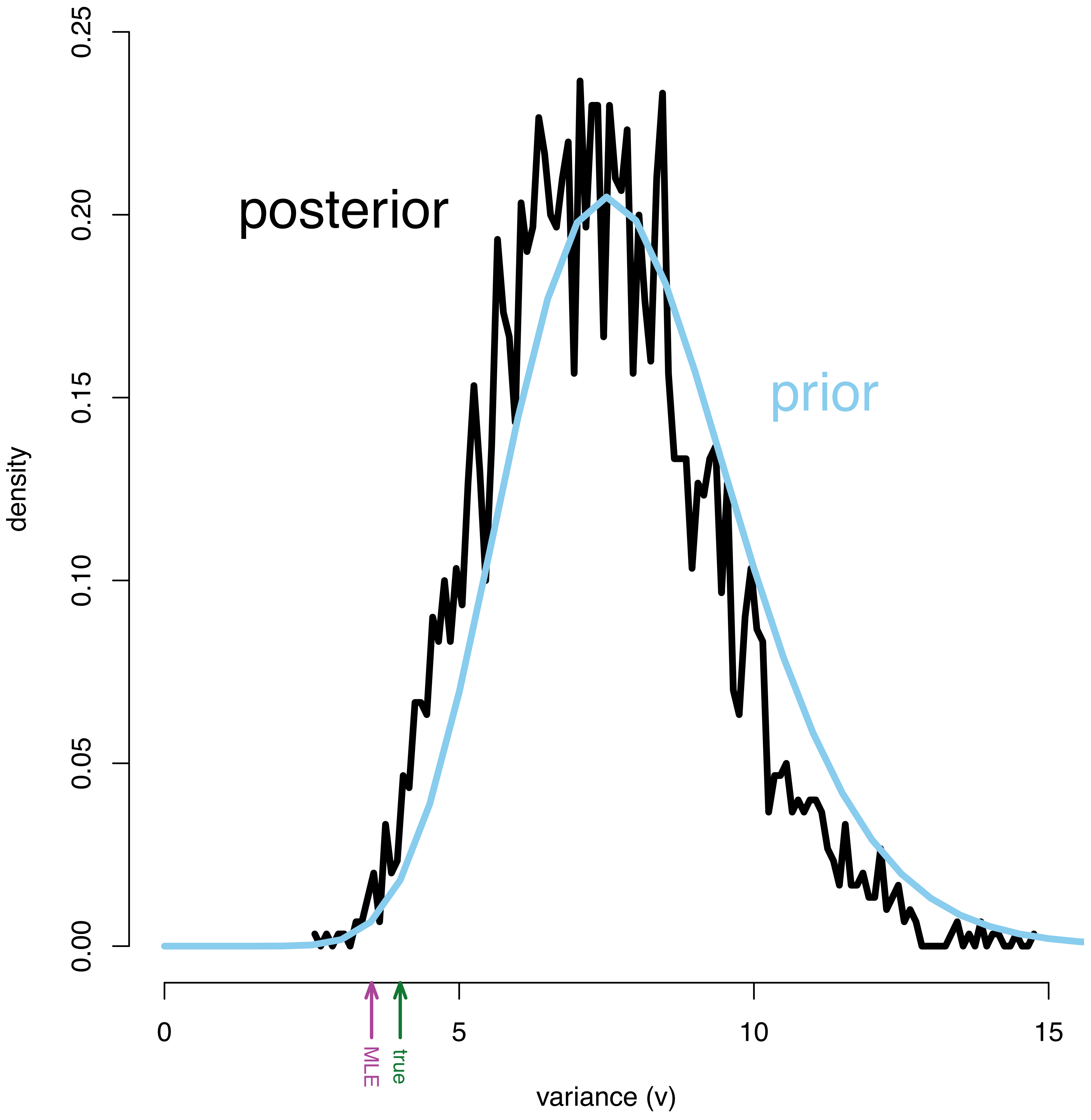
RevBayes Demo: Archery

The trace plot of the MCMC samples for the mean distance from the target center



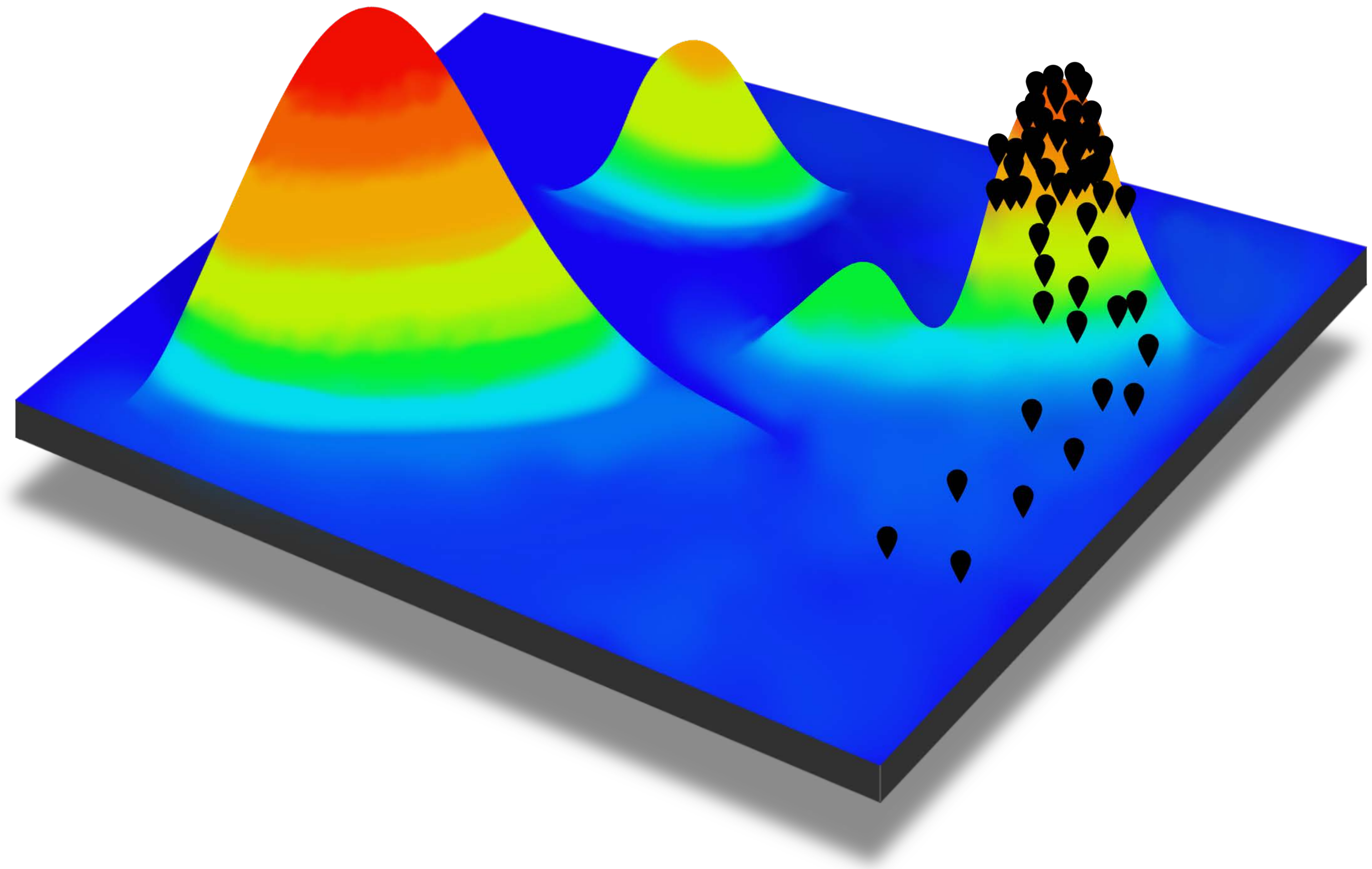
RevBayes Demo: Archery

The posterior estimate fo the variance (v) is quite different from the true value (4.0) and from the highest likelihood value found by our MCMC (MLE = 3.51374)



This indicates that the prior is having a strong influence on the posterior. Why do you think that is?

Metropolis Coupled MCMC

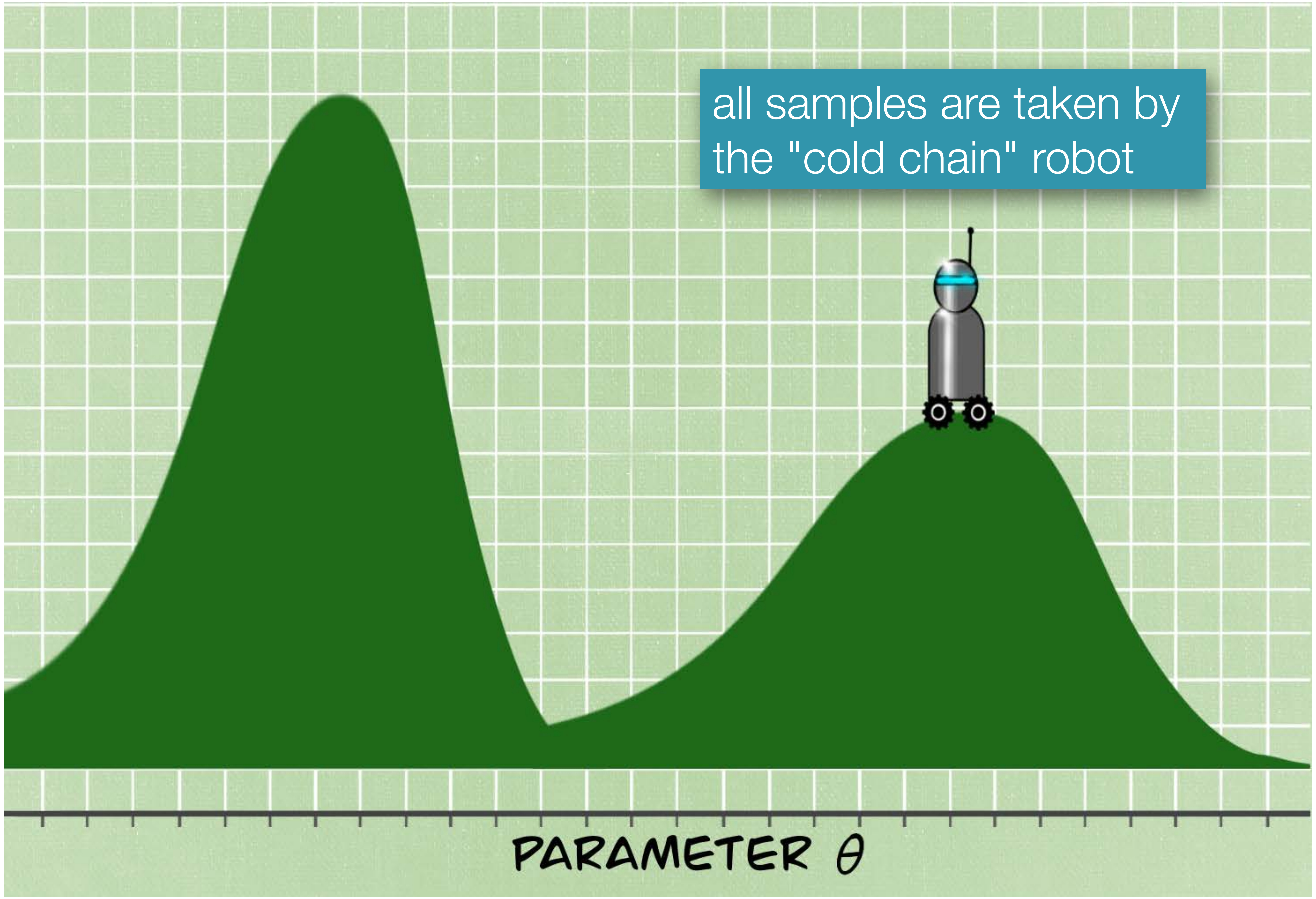


sometimes even good robots need help

MCMCMC introduces helper robots that act as scouts to explore more parameter space

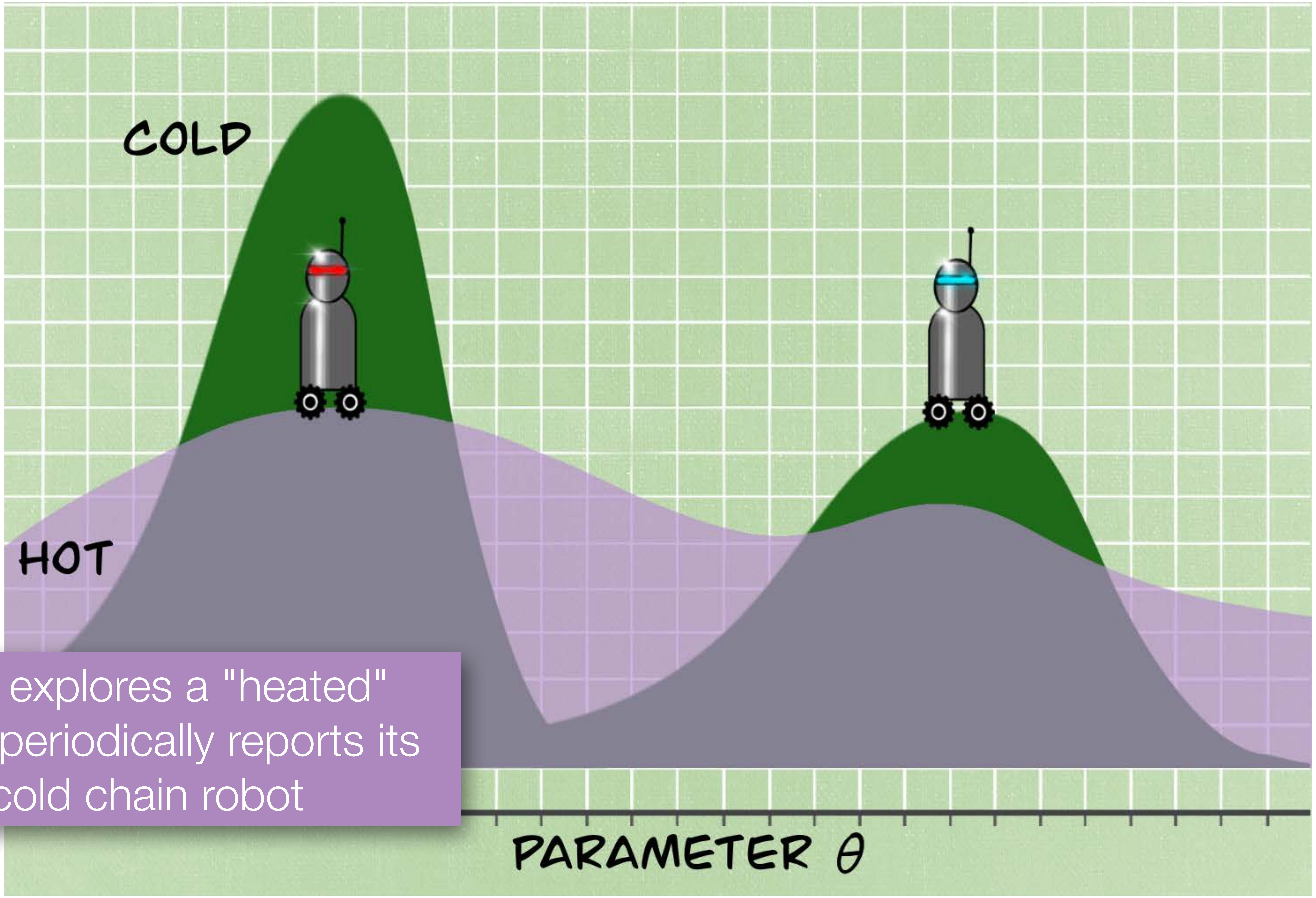
Geyer, C. J. 1991. Markov chain Monte Carlo maximum likelihood for dependent data. Pages 156-163 in Computing Science and Statistics (E. Keramidas, ed.).

Metropolis Coupled MCMC



(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

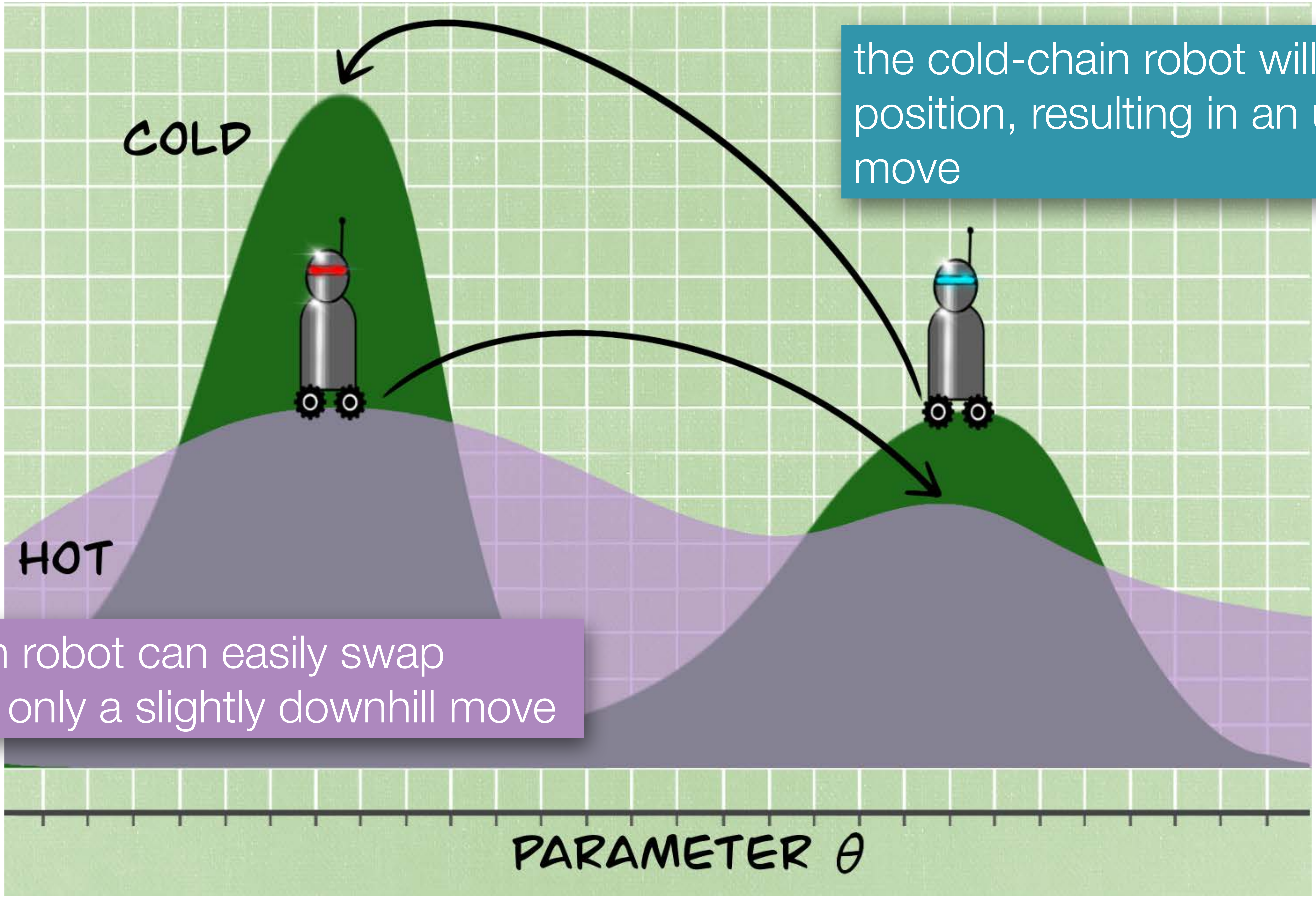
Metropolis Coupled MCMC



the scout robot explores a "heated" landscape and periodically reports its position to the cold chain robot

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

Metropolis Coupled MCMC



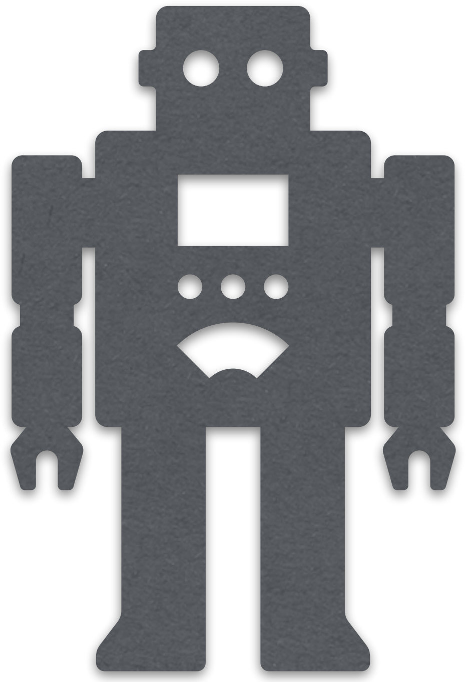
the cold-chain robot will swap position, resulting in an uphill move

the hot-chain robot can easily swap because it is only a slightly downhill move

(based on slides by Paul Lewis https://molevolworkshop.github.io/faculty-lewis)

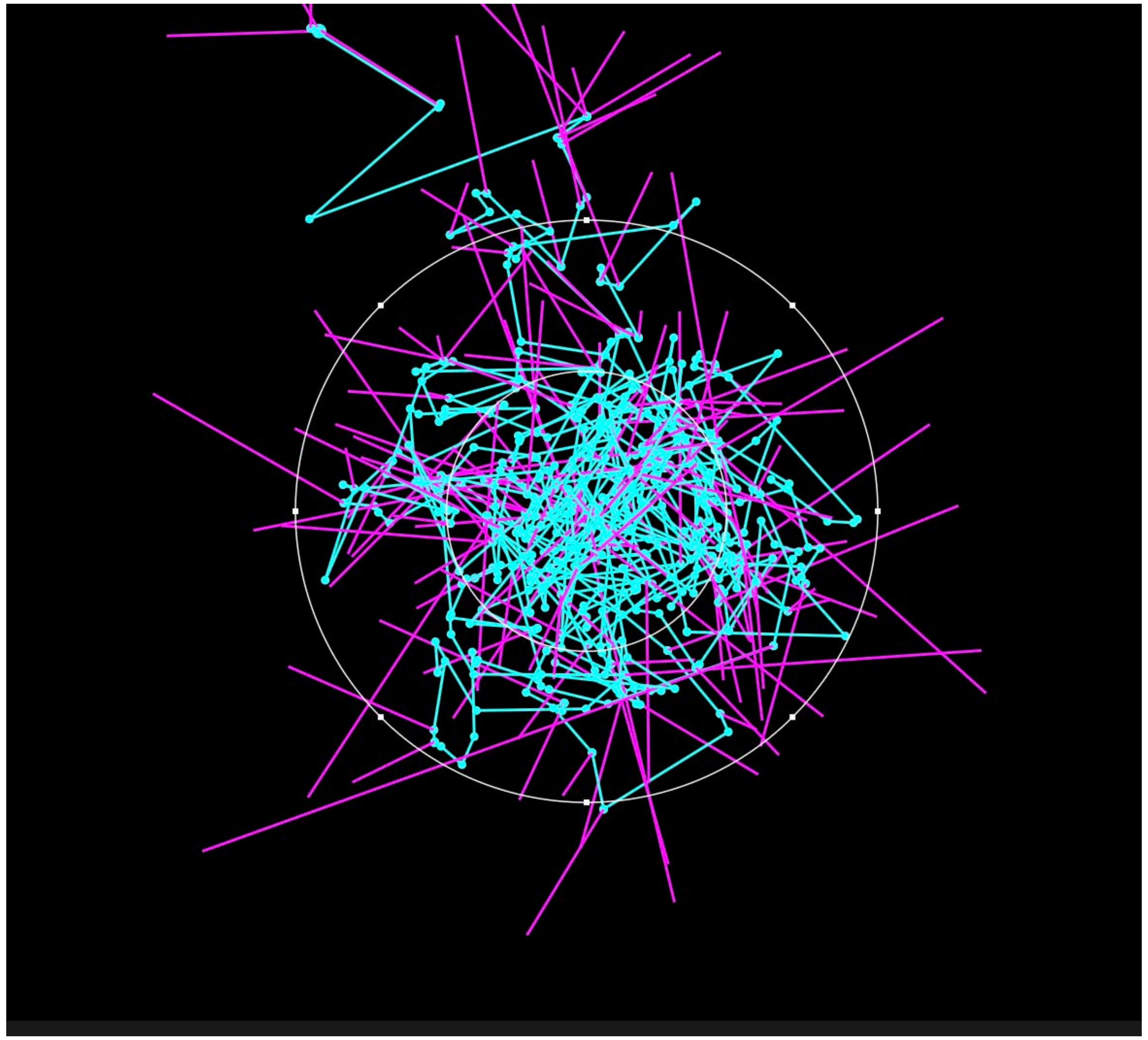
Markov Chain Monte Carlo

Learn more about MCMC!



https://phylogeny.uconn.edu/mcmc-robot/

MCMCRobot, a helpful tool for learning MCMC by Paul Lewis



Markov Chain Monte Carlo

Learn more about MCMC!

REVIEW ARTICLE
DOI: 10.1038/s41559-017-0280-x

nature ecology & evolution

A biologist's guide to Bayesian phylogenetic analysis

Fabrícia F. Nascimento[1,4*], Mario dos Reis[2] and Ziheng Yang[3*]

https://thednainus.wordpress.com/2017/03/03/tutorial-bayesian-mcmc-phylogenetics-using-r/