Unit 2: Bayesian Learning

4. Inference by sampling

10/15/2020

Inference by sampling

1. Sampling algorithms like Markov chain Monte Carlo (MCMC) can be used to approximate Bayesian inference

2. Markov chain Monte Carlo can be used to uncover people’s mental representations

3. Sampling may be how the mind works at Marr’s algorithmic level

Marr's levels of analysis

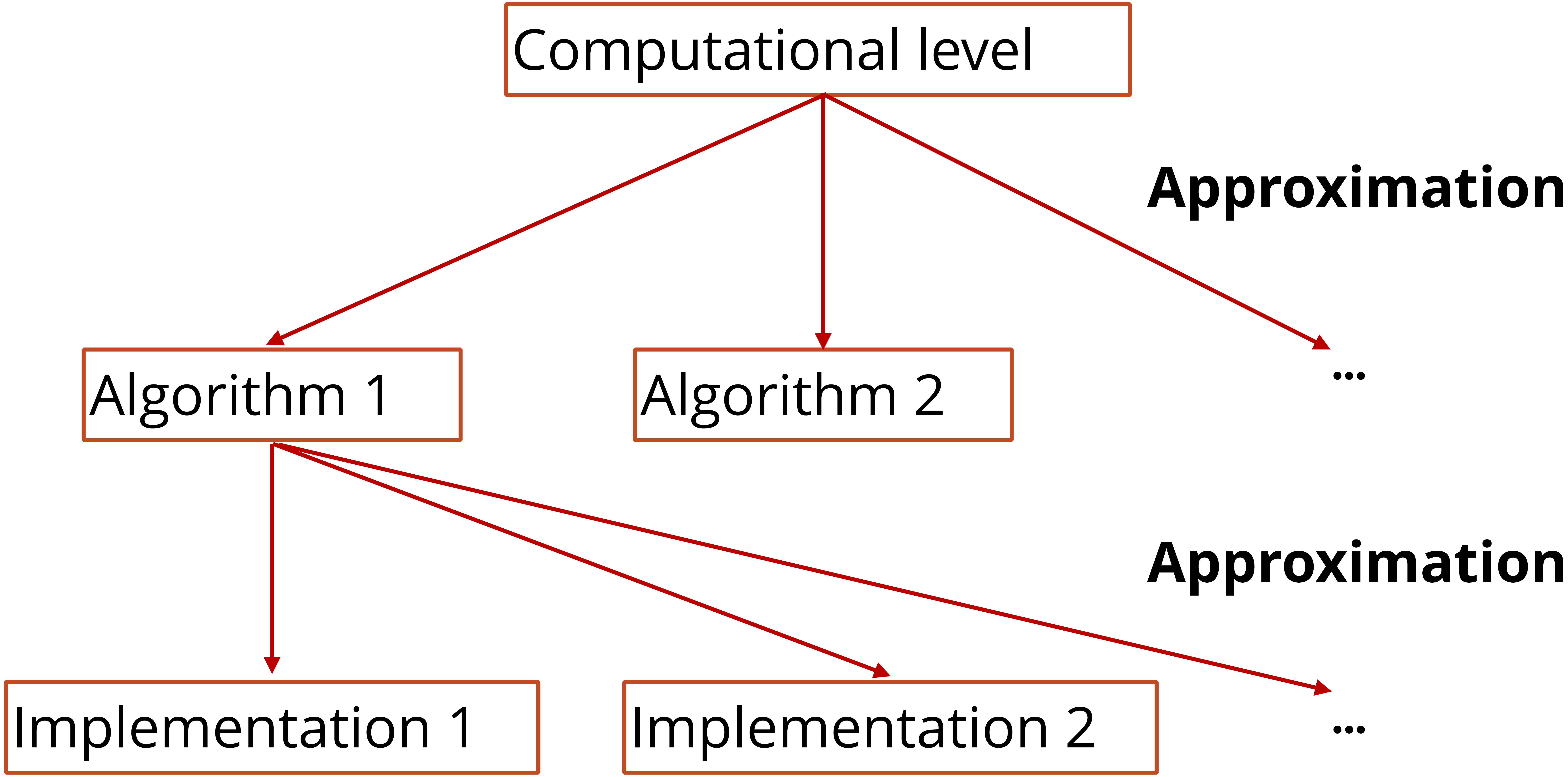
Computational Theory
What is the goal of the computation? What is the logic of the strategy by which it can be carried out?

Representation and algorithm
What is the representation for the input and output, and what is the algorithm for the transformation?

Hardware implementation
How can the representation and algorithm be realized physically?

Marr (1982)

Each level approximates the level above it



Exact inference works for small problems

60 80 10 30

$$P\left(h|X\right) = \frac{P\left(X|h\right)P\left(h\right)}{\sum_{h' \in H} P\left(X|h'\right)P\left(h'\right)}$$

Finite set of hypotheses,
Not too much data

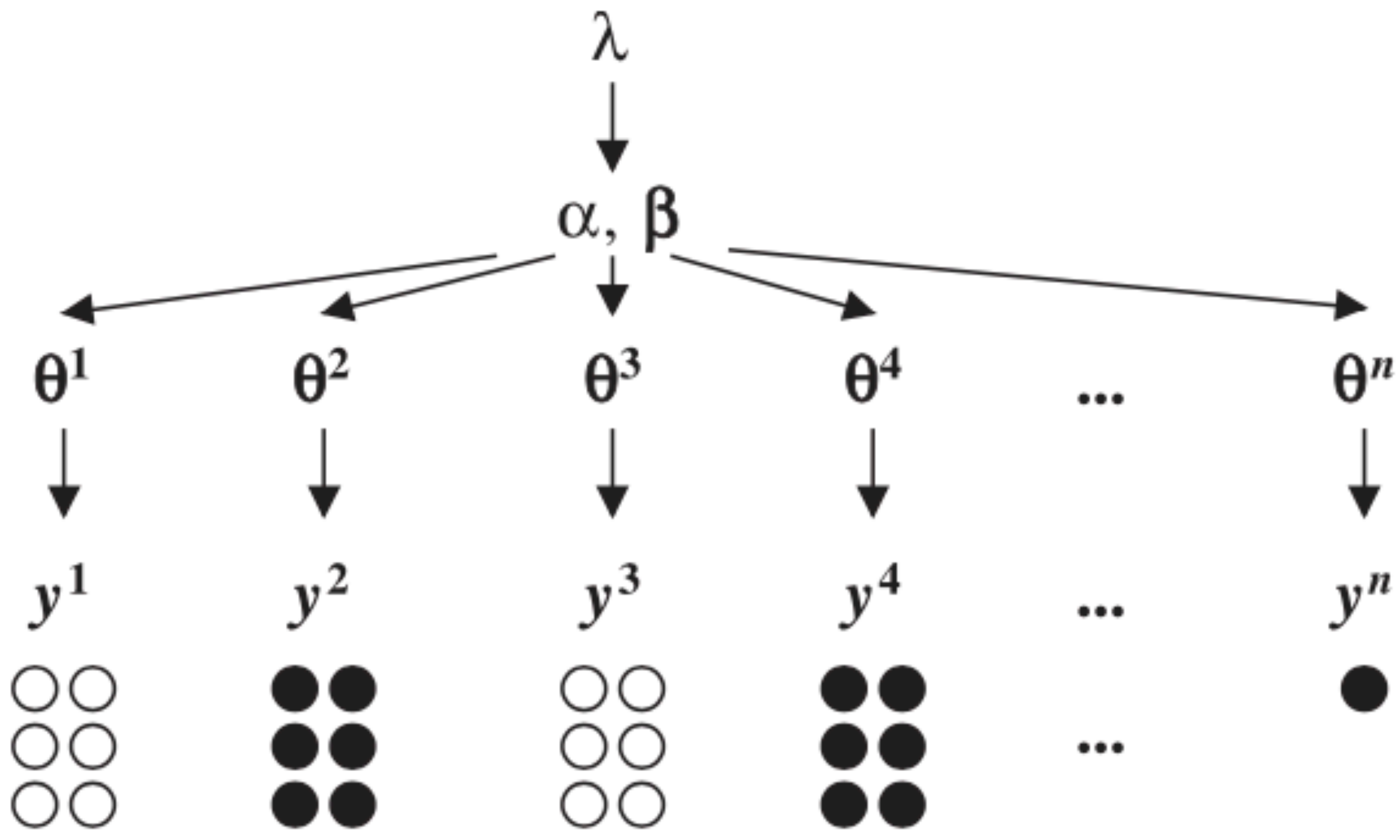
Exact inference does not scale up

$$\alpha \sim \text{Exponential}\,(\lambda)$$

$$\boldsymbol{\beta} \sim \text{Dirichlet}\,(\mathbf{1})$$

$$\boldsymbol{\theta}^{\boldsymbol{i}} \sim \text{Dirichlet}\,(\alpha\boldsymbol{\beta})$$

$$y^i \,|\, n^i \sim \text{Multinomial}\,(\boldsymbol{\theta}^{\boldsymbol{i}})$$



Kemp, Performs, & Tenenbaum (2007)

Overhypothesis about how uniform bags are: $p\,(\alpha\,|\,y)$

This can't be computed by exact inference

How to compute posteriors

1. **Exact inference**: Compute the analytic closed formula.
   Works for discrete hypotheses and simple,
   non-hierarchical models

2. **Grid search**: Just try every possible value of the parameters (or at least try lots of them).
   Works for low dimensional problems with
   diffuse posterior distributions

3. **Markov chain monte carlo**: Set up a sampling process that asymptotically approximates the posterior.
   Works (approximately) for any model given sufficient time

The parable of King Markov

King Markov

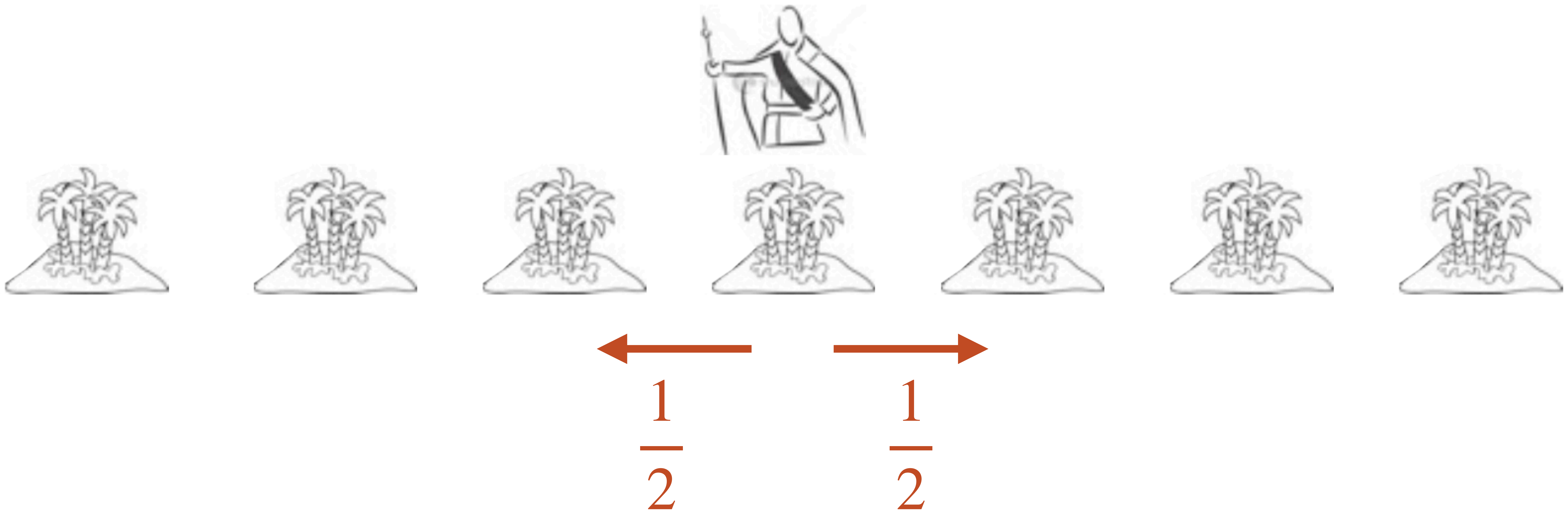
From Richard McElreath

The contract

Contract: King Markov must visit each island in proportion to its population size

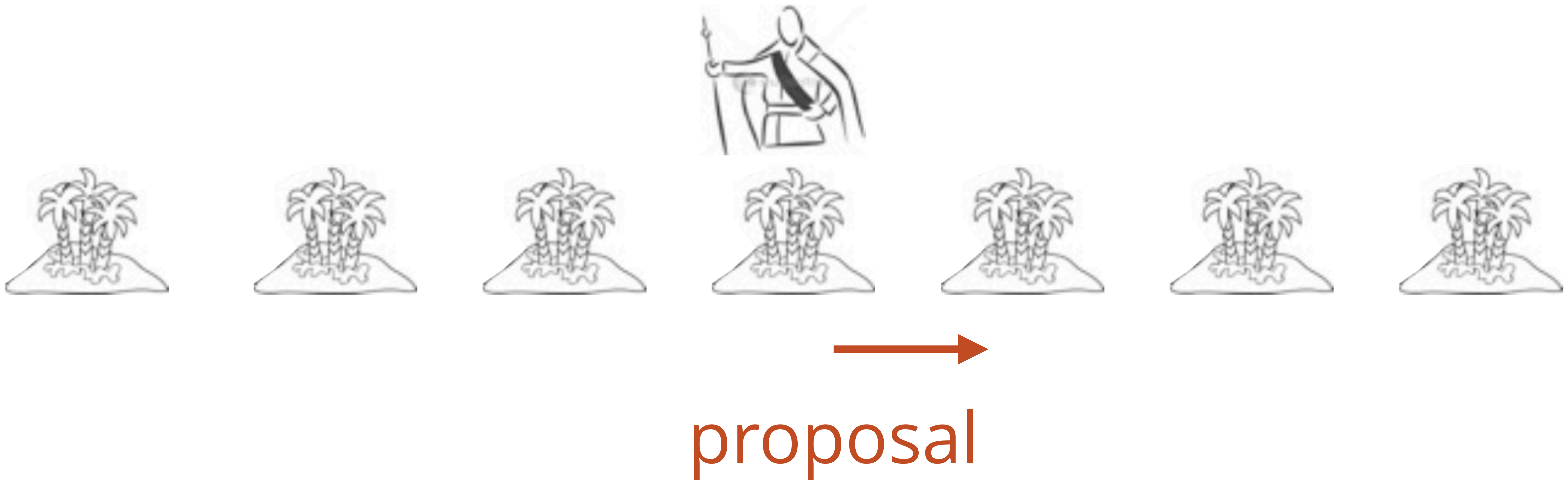
The Metropolis Archipelago

An algorithm to fulfill the contract



1. Flip a coin to choose the island on the left or right.
   This is the “proposal” island

An algorithm to fulfill the contract



1. Flip a coin to choose the island on the left or right. This is the “proposal” island

An algorithm to fulfill the contract

2. Find the population of the proposal island

An algorithm to fulfill the contract

p_4 p_5

3. Find the population of the current island

An algorithm to fulfill the contract

p_4 p_5

4. Move to the proposed island with probability $\dfrac{p_5}{p_4}$

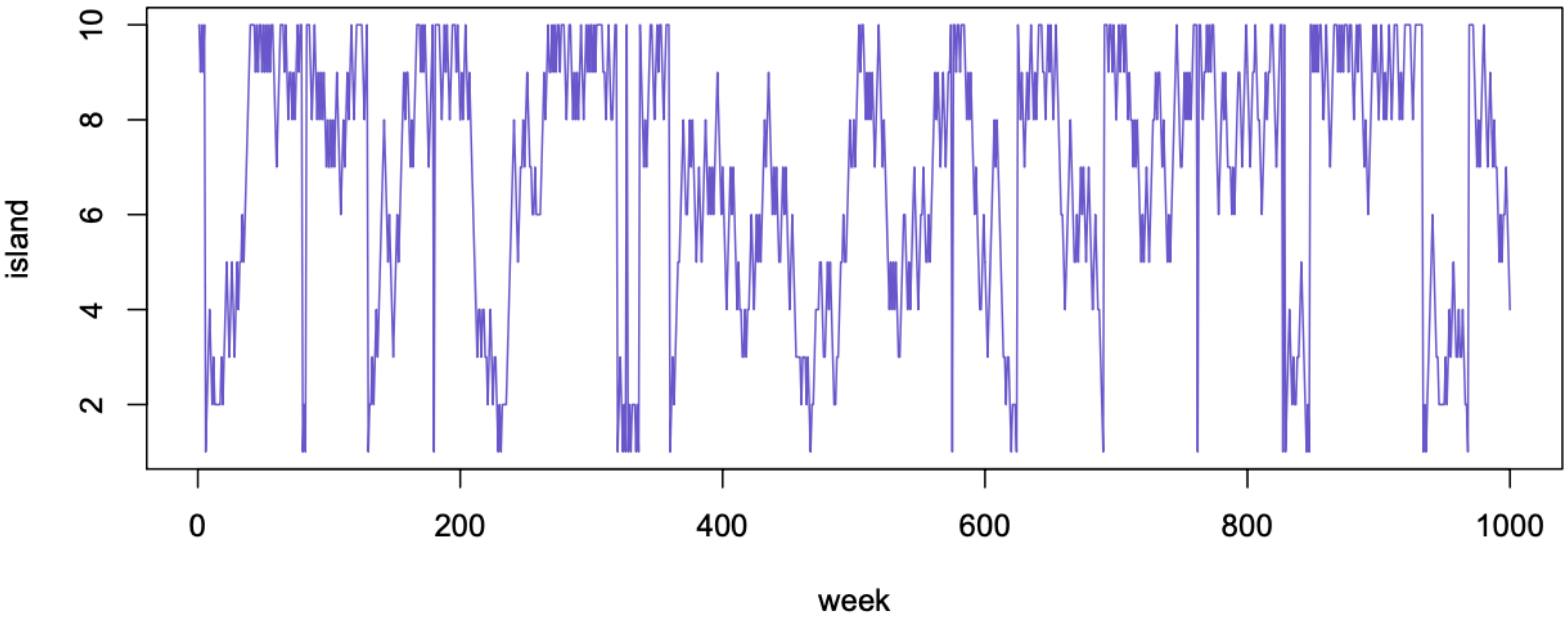
An algorithm to fulfill the contract

5. Repeat forever

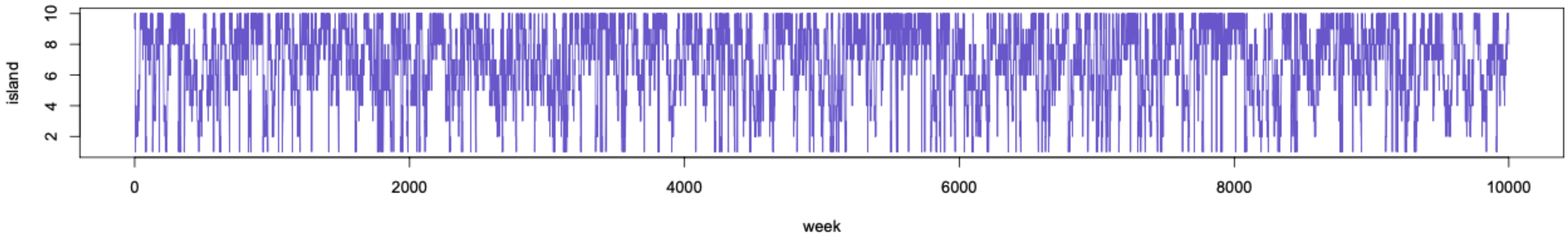
This algorithm is guaranteed to fulfill the contract given infinite time

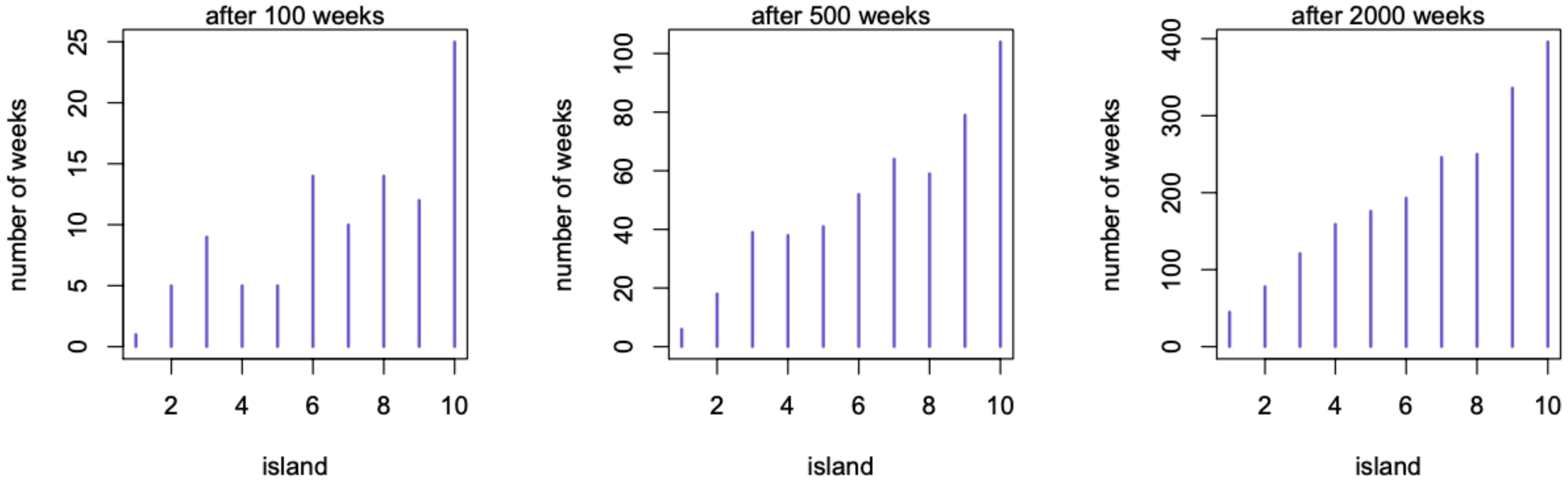
1. Flip a coin to choose the island on the left or right.
   This is the “proposal” island
2. Find the population of the proposal island p_p
3. Find the population of the current island p_c
4. Move to the proposed island with probability $\dfrac{p_p}{p_c}$
5. Repeat forever

Markov's chain of visits



Markov's chain of visits



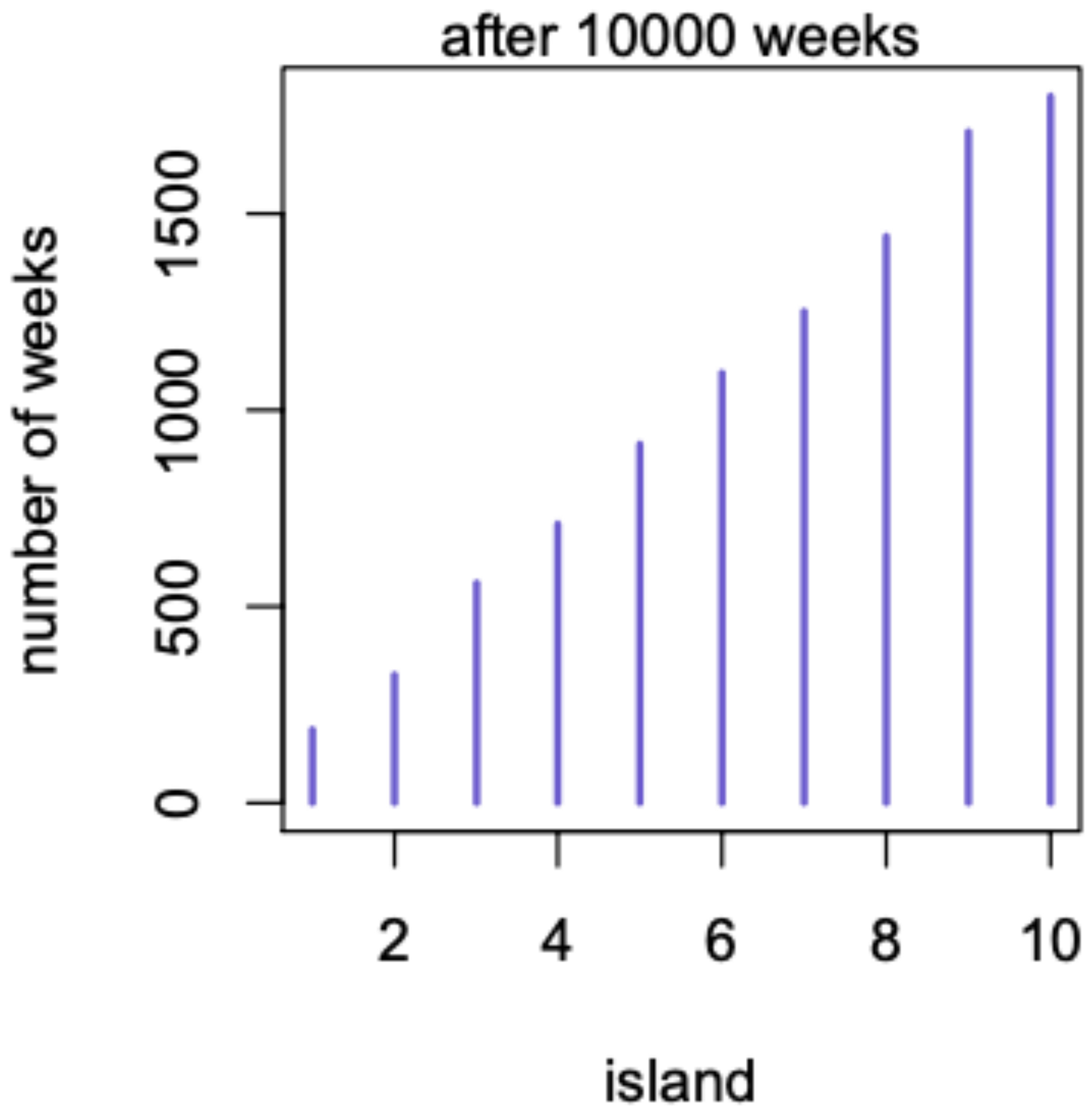


The Metropolis algorithm for estimating distributions

The **Metropolis** algorithm converges to correct proportions in the long run

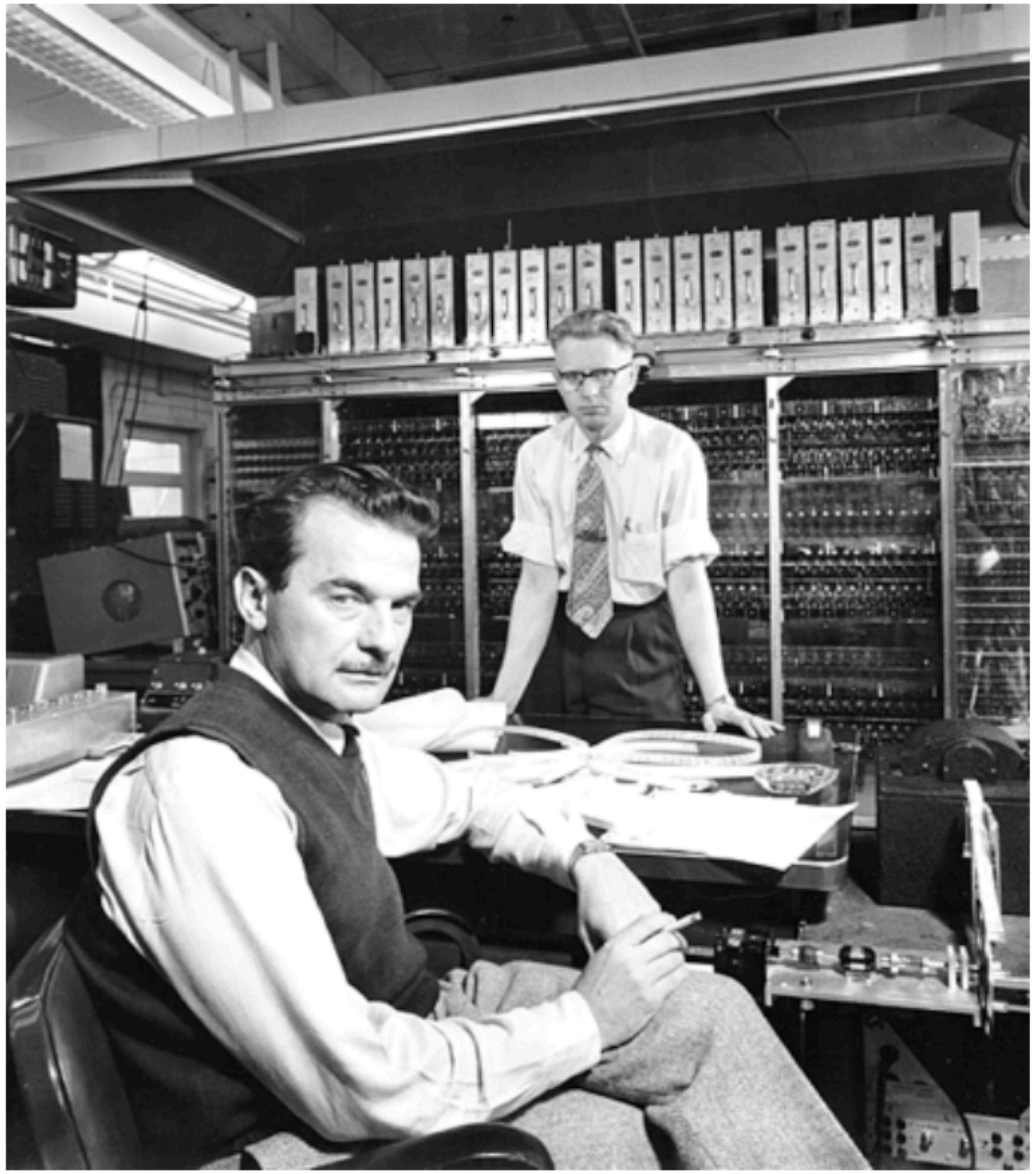
We can use this same algorithm to draw samples from distributions we don't know a closed form for

- Islands - parameter values
- Population size - posterior probability



Works for any number of dimensions.
Works for both discrete and continuous parameters

The origins of Markov Chain Monte Carlo



Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a process for setting up a chain whose long run (ergodic) distribution is the probability distribution of interest

Markov: A process where only the last state matters

Monte Carlo: Random. Refers to the Monte Carlo Casino in Monaco. Used as a code word between von Neumann and Ulam who were working on the Manhattan project.

The Metropolis algorithm in Math

To get samples from a function f for which you can compute a density but not a probability:

The Metropolis algorithm in Math

To get samples from a function f

Pick a random starting point x_0

For each time step t,

1. Proposed a point p_t according to a distribution $g\left(p_t | x_{t-1}\right)$

   People often use $g\left(p_t | x_{t-1}\right) \sim$ Normal $\left(x_{t-1}, \sigma\right)$

   But the only constraint is that it has to be symmetric

$$g\left(p_t | x_{t-1}\right) = g\left(x_{t-1} | p\right)$$

The Metropolis algorithm in Math

To get samples from a function f

Pick a random starting point x_0

For each time step t,

1. Proposed a point p_t according to a distribution $g\left(p_t | x_{t-1}\right)$

2. Calculate an acceptance ratio $\alpha = \dfrac{f\left(p_t\right)}{f\left(x_{t-1}\right)}$

The Metropolis algorithm in Math

To get samples from a function f

Pick a random starting point x_0

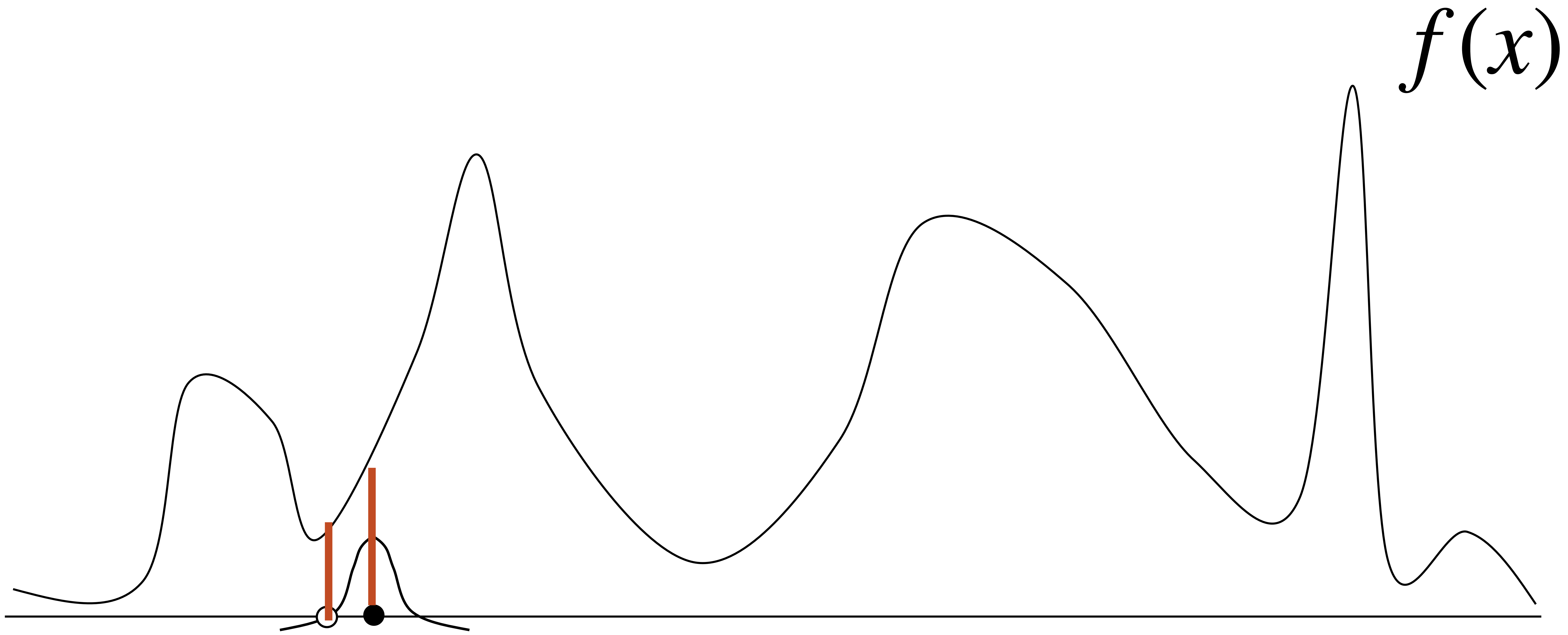
For each time step t,

1. Proposed a point p_t according to a distribution $g\left(p_t \mid x_{t-1}\right)$

2. Calculate an acceptance ratio $\alpha = \dfrac{f\left(p_t\right)}{f\left(x_{t-1}\right)}$

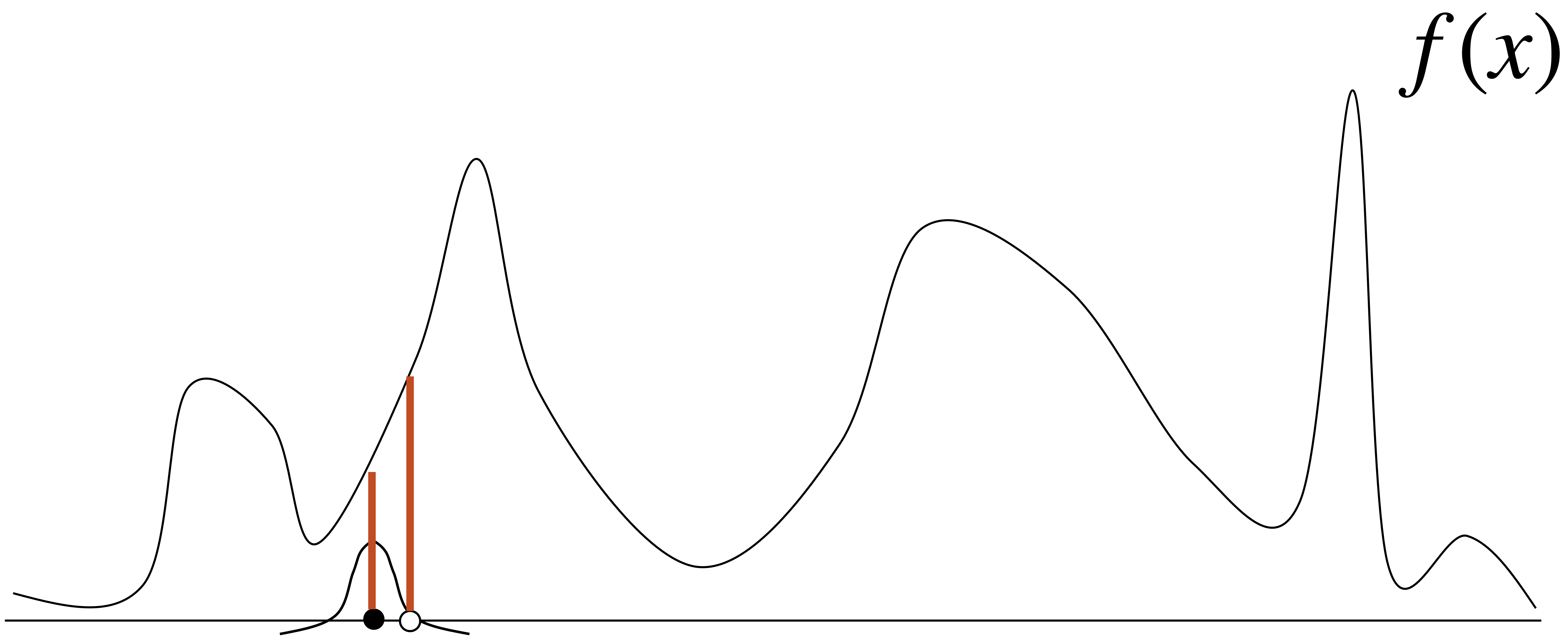
3. With probability $\min(\alpha, 1)$, add p_t to the chain
   Otherwise add x_{t-1} again

The Metropolis algorithm in sketches



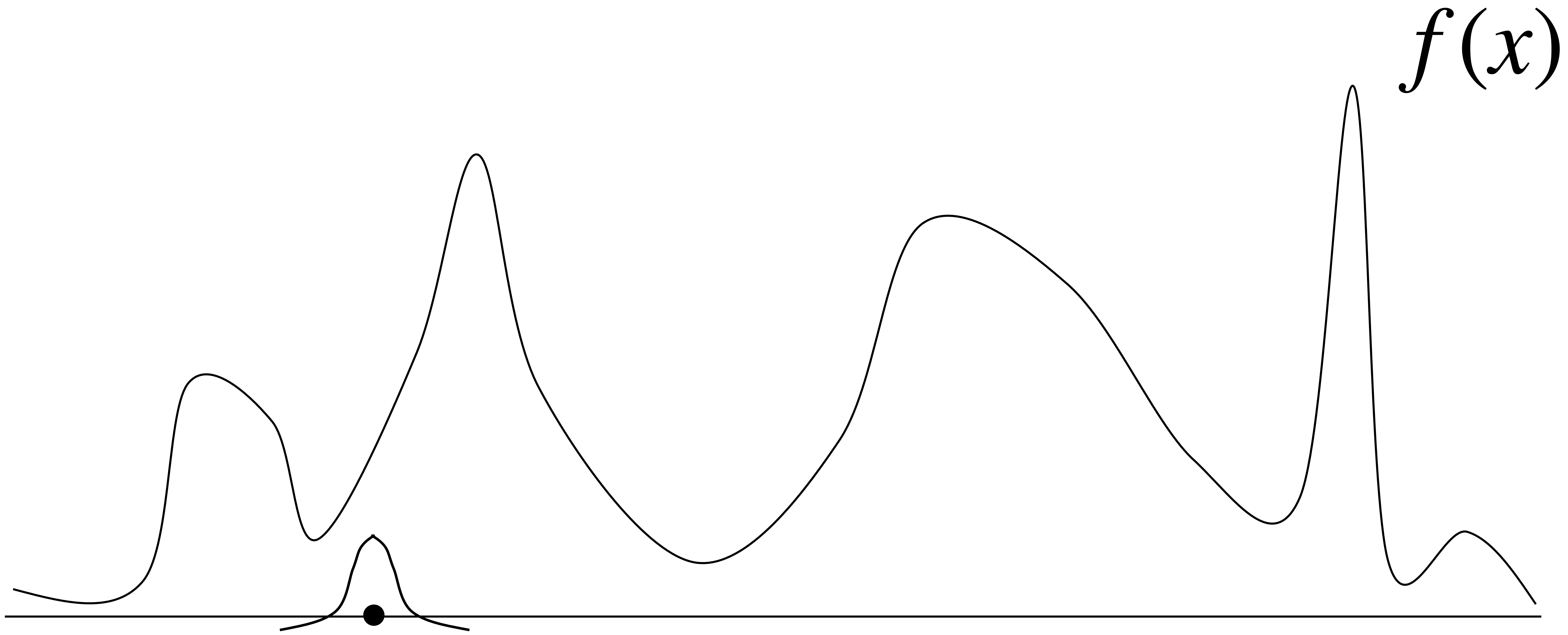
$$\alpha = \frac{f(p_t)}{f(x_{t-1})}$$

The Metropolis algorithm in sketches

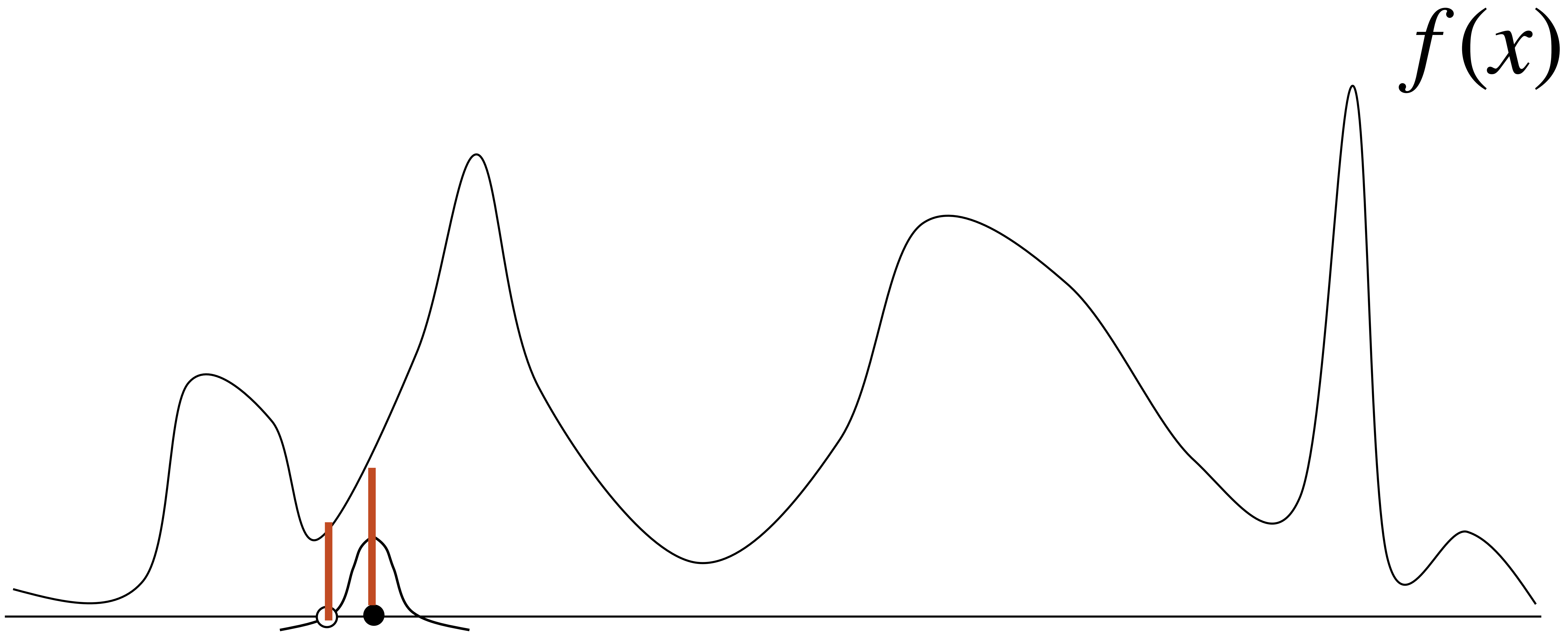


$$\alpha = \frac{f(p_t)}{f(x_{t-1})} > 1$$

The Metropolis algorithm in sketches

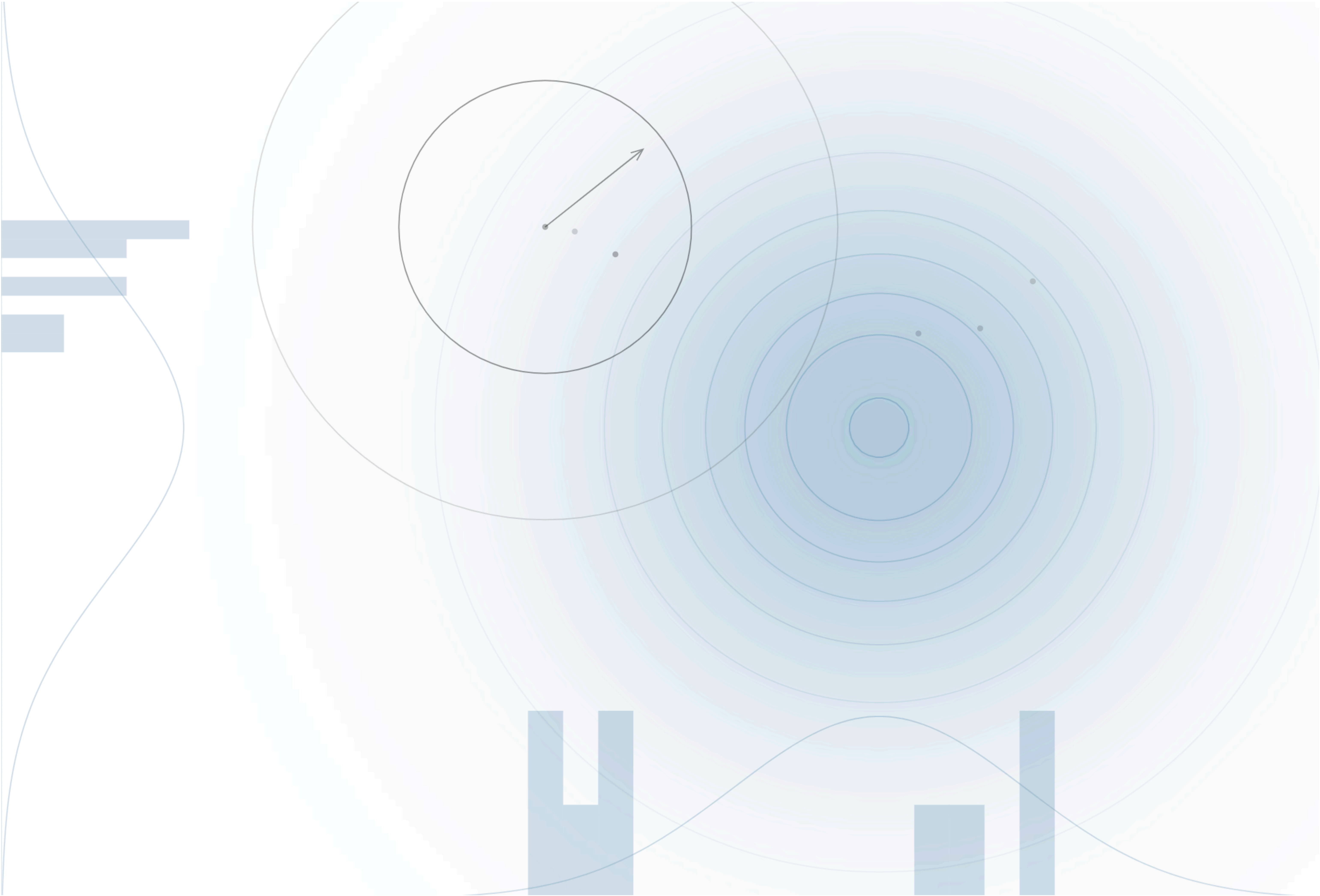


The Metropolis algorithm in sketches



$$\alpha = \frac{f\left(p_t\right)}{f\left(x_{t-1}\right)}$$

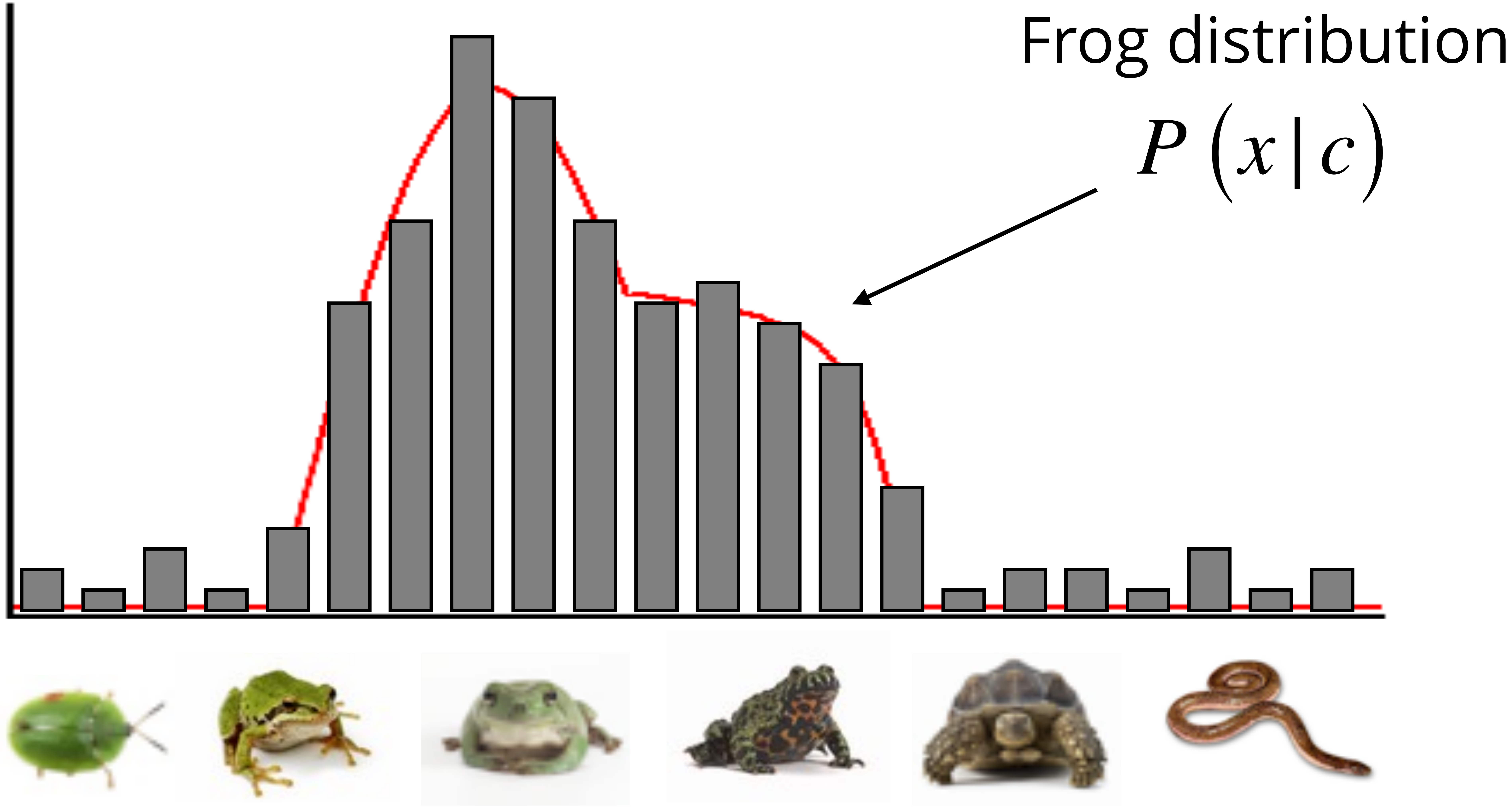
Metropolis in action



http://chi-feng.github.io/mcmc-demo/

Categories are central to cognition

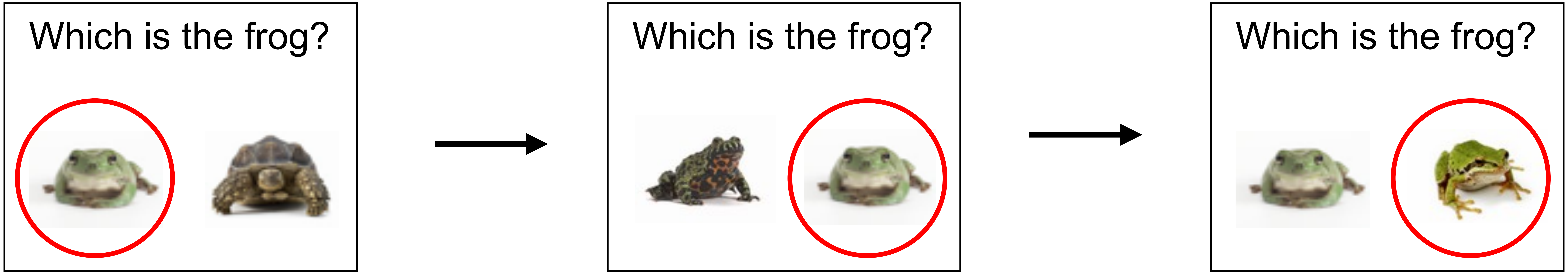
Using MCMC to understand people (Sanborn, Griffiths, & Shiffrin, 2010)

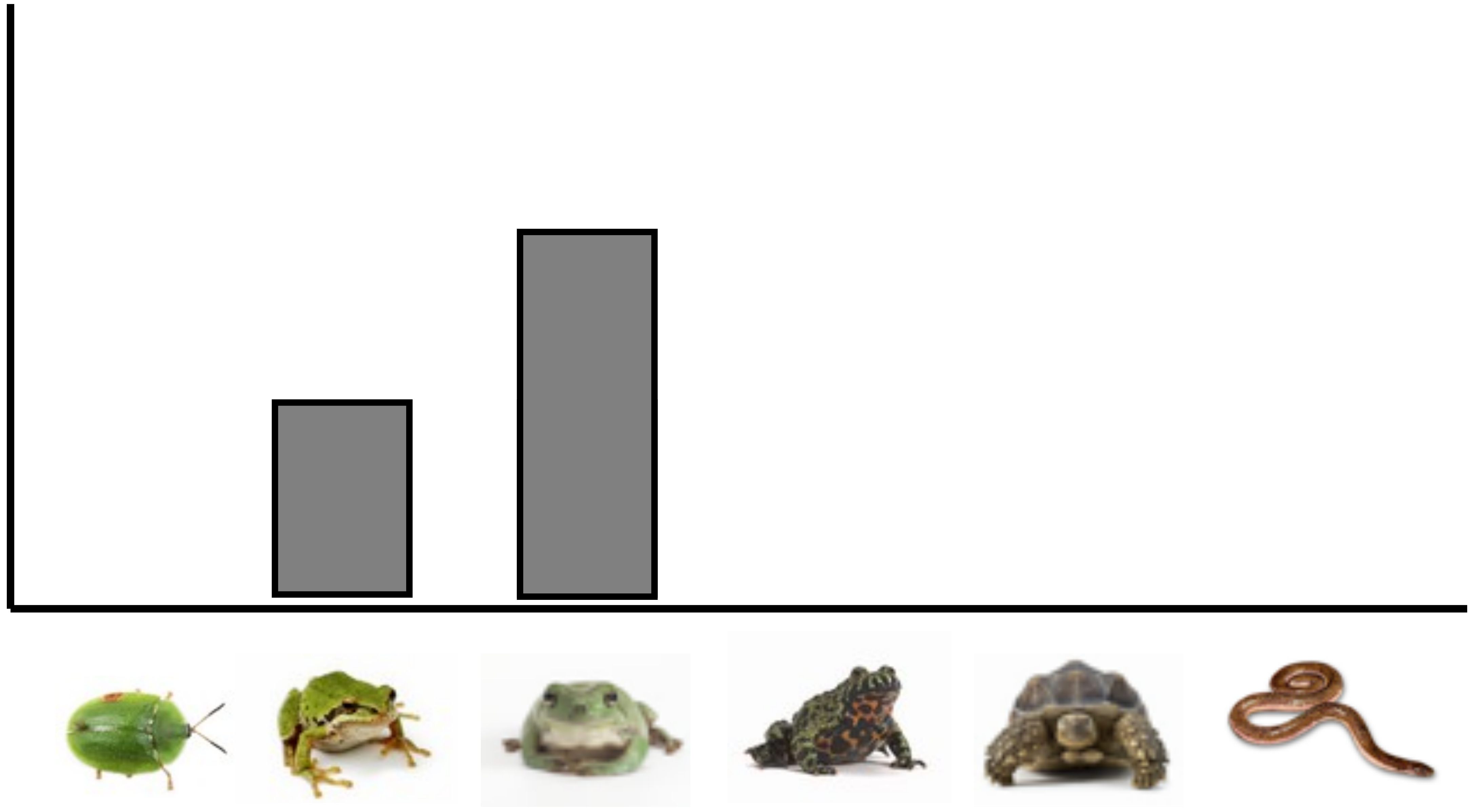


The experimental task idea

Which animal is a frog?

The experimental task idea

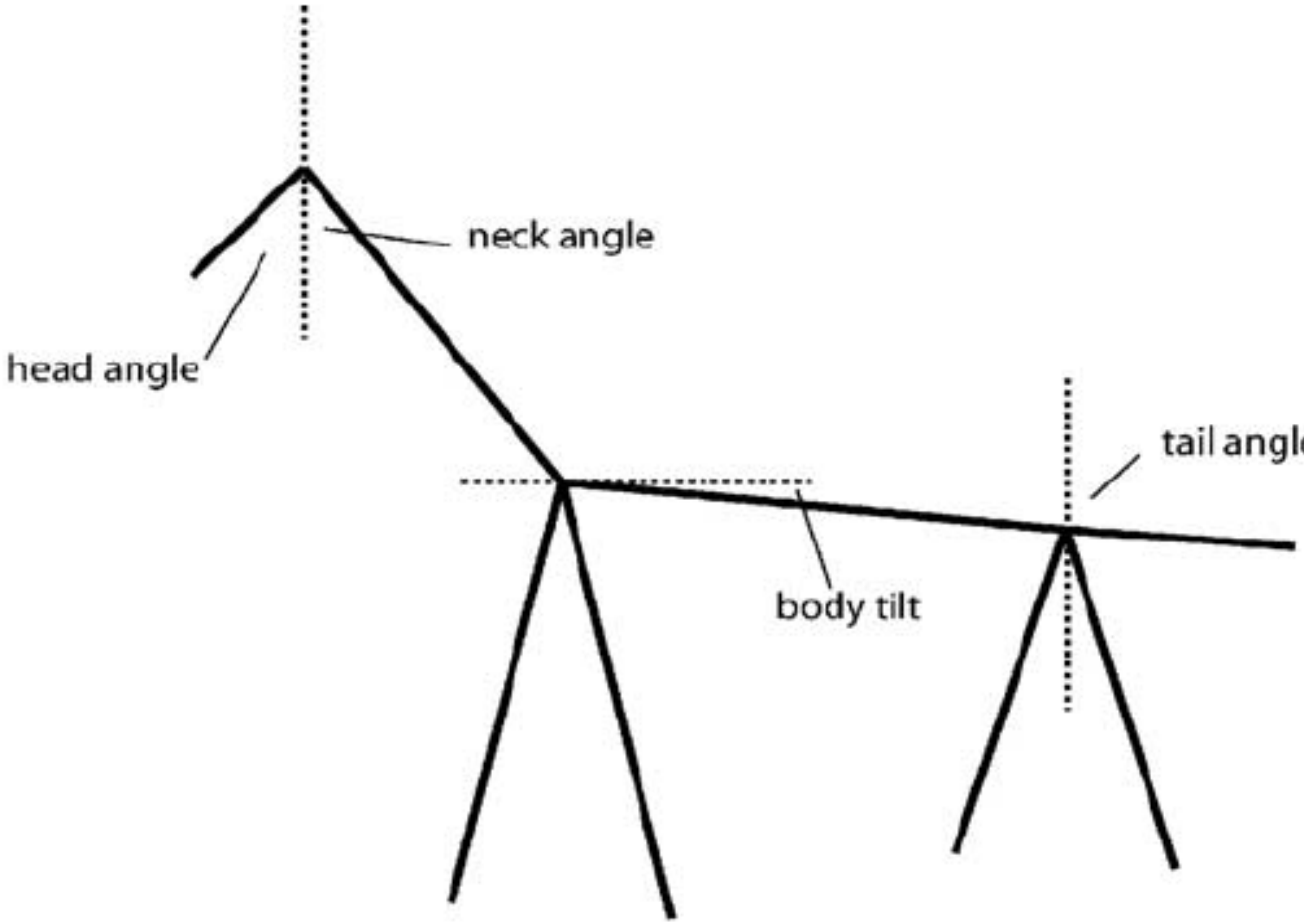




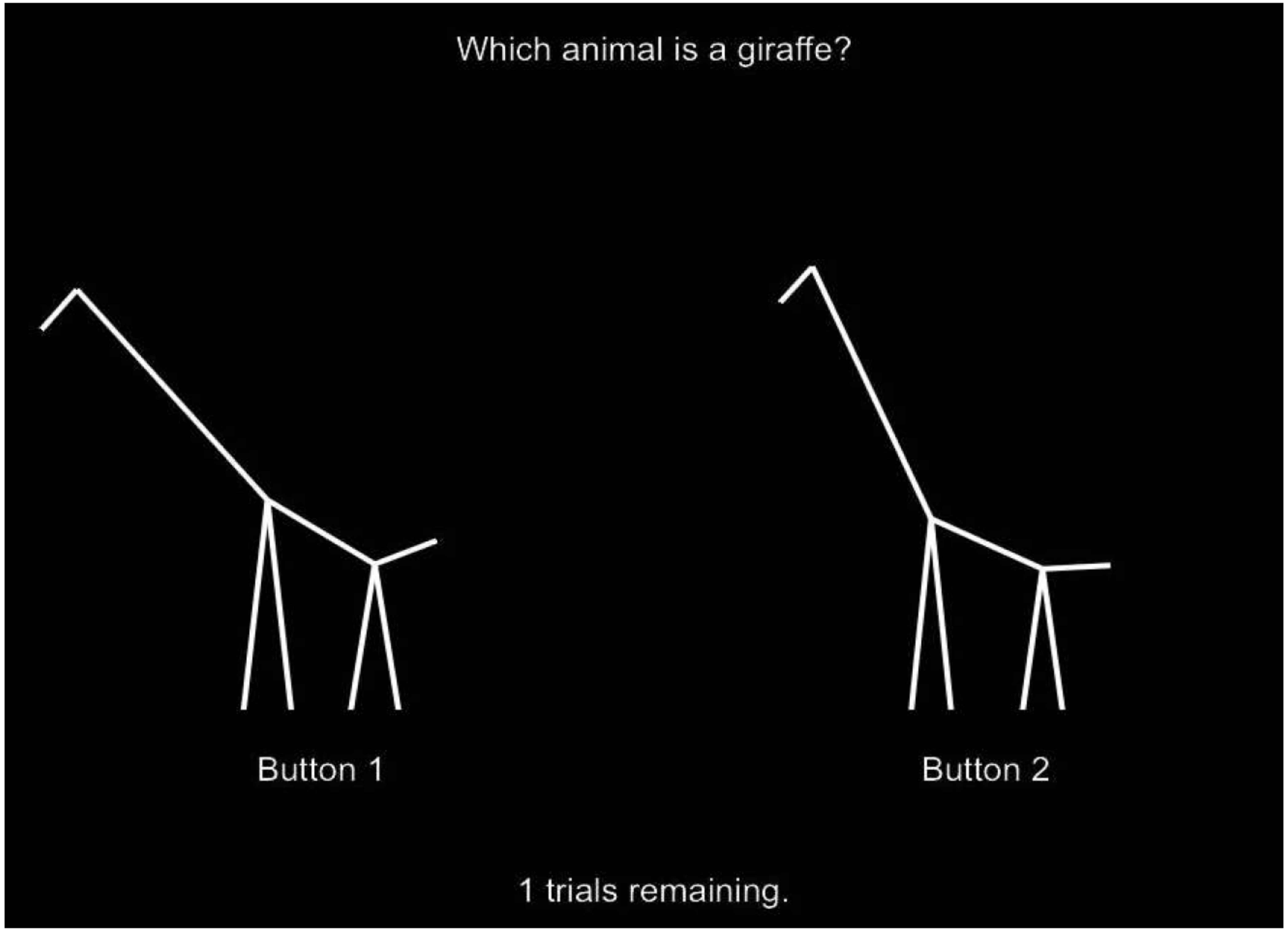
The experimental task in practice

Examined distributions for four categories:

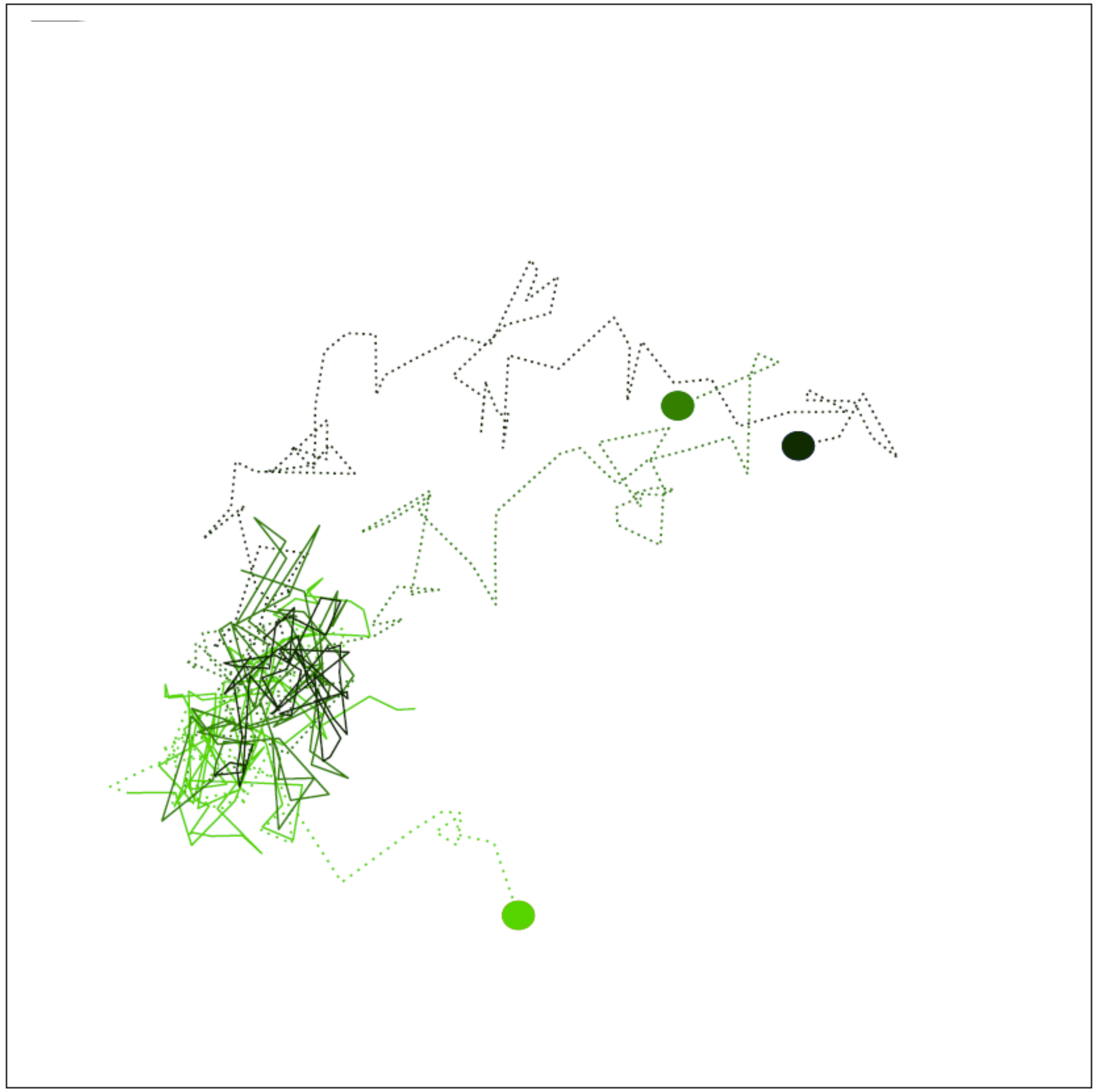
1. giraffes
2. horses
3. cats
4. dogs

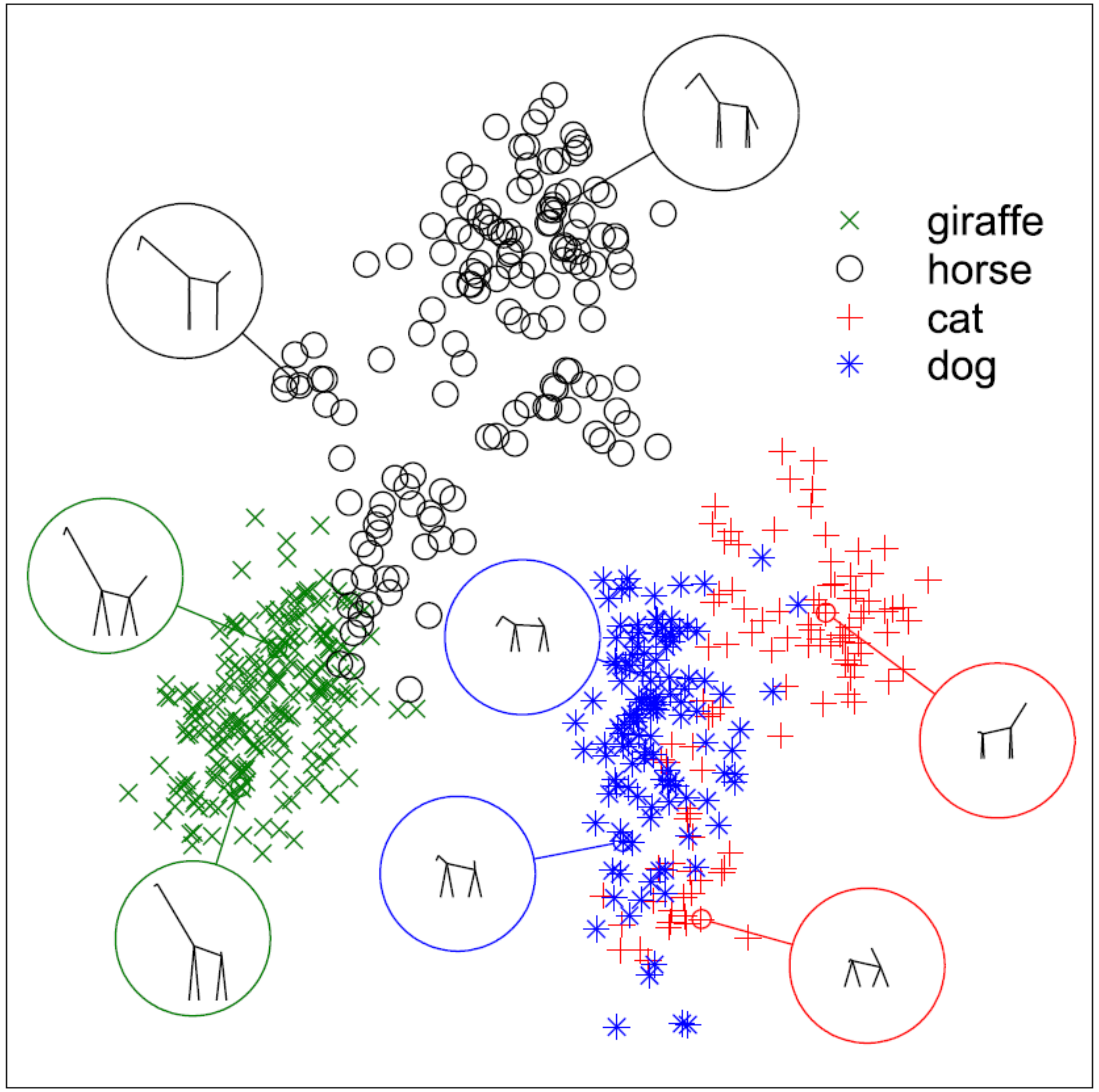


The experimental task in practice

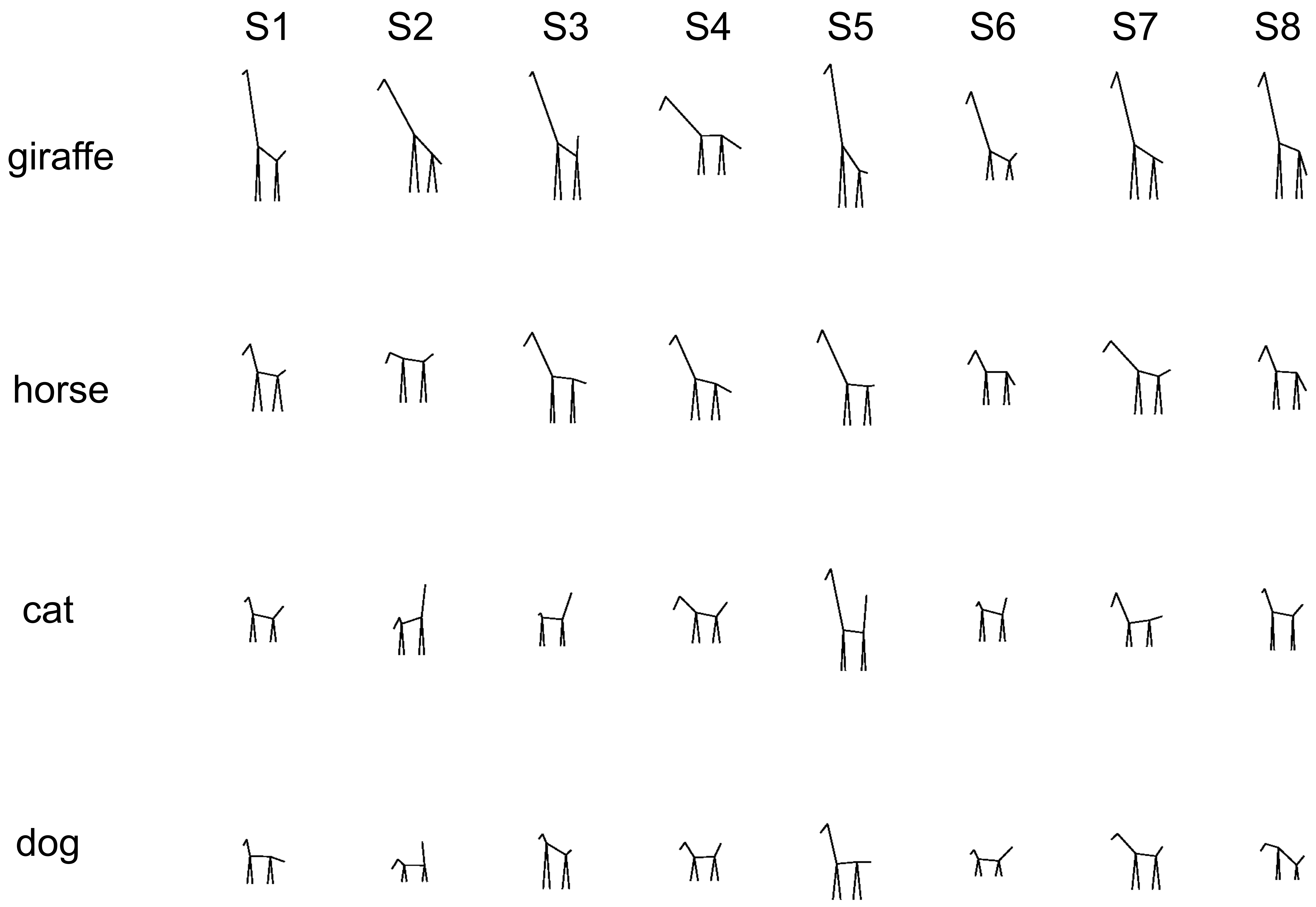


Samples for one experimental participant

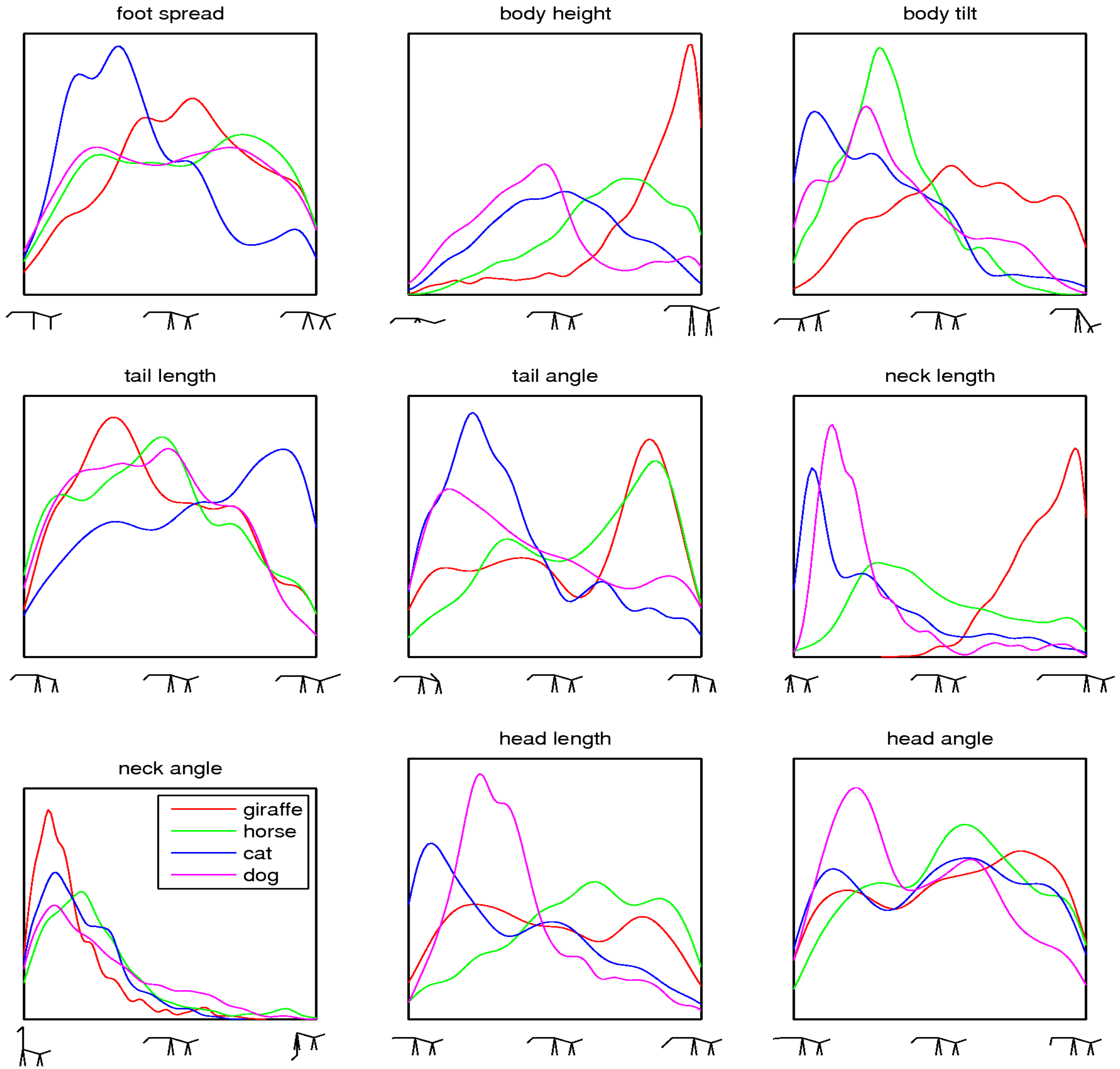




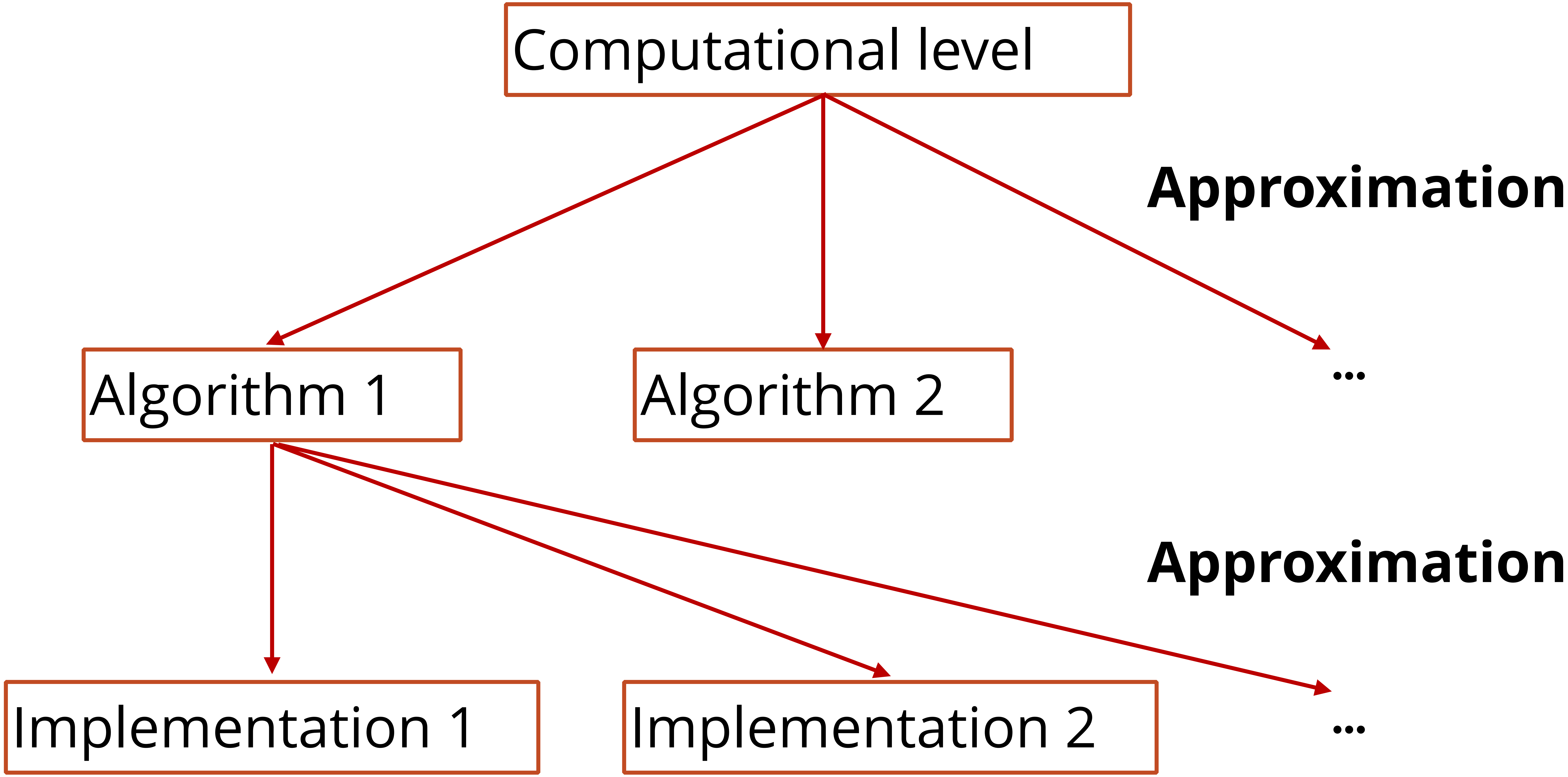
Mean animals for each of the 8 participants



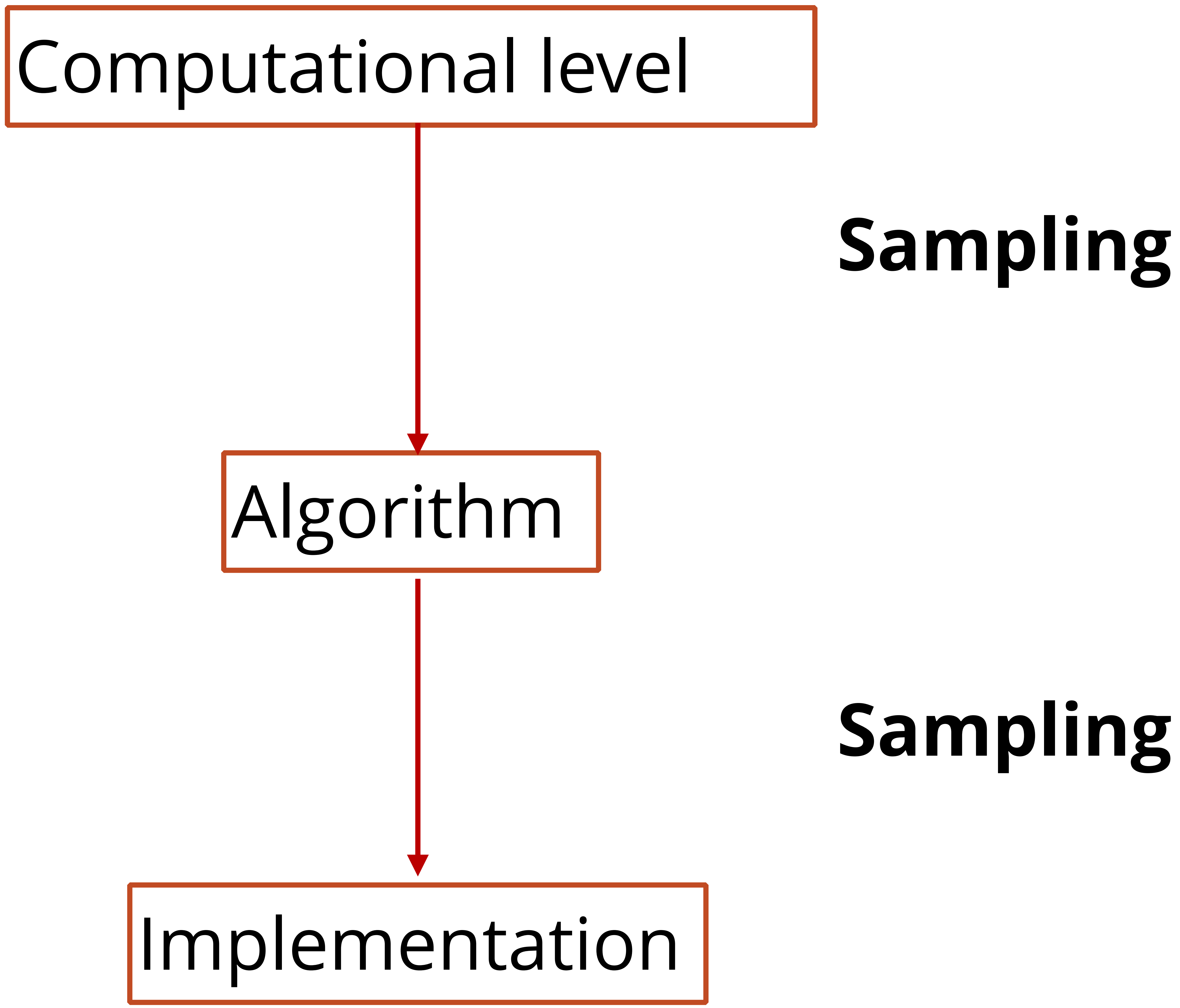
Uncovering animal representations



Each level approximates the level above it



Maybe each level approximates the level above it by sampling!



Sampling in human cognition (Vul & Pashler, 2008)

The wisdom of crowds:
you can get better answers to numerical questions
(“What percent of the worlds airports are in the United States?)
by averaging over multiple people

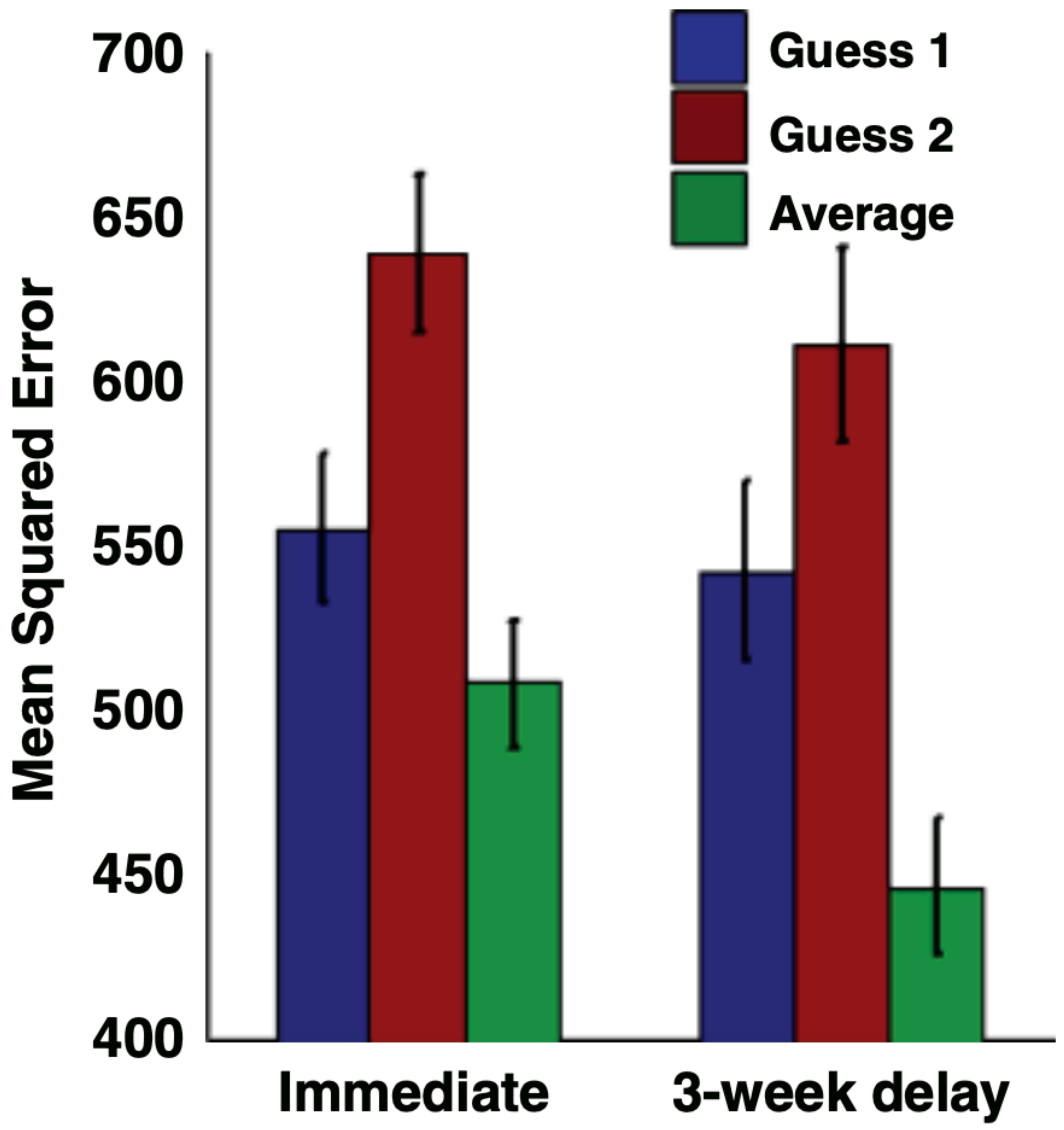
Why?

Sampling in human cognition (Vul & Pashler, 2008)

The crowd within:
do you get better answers to numerical questions
by asking the same person multiple times?

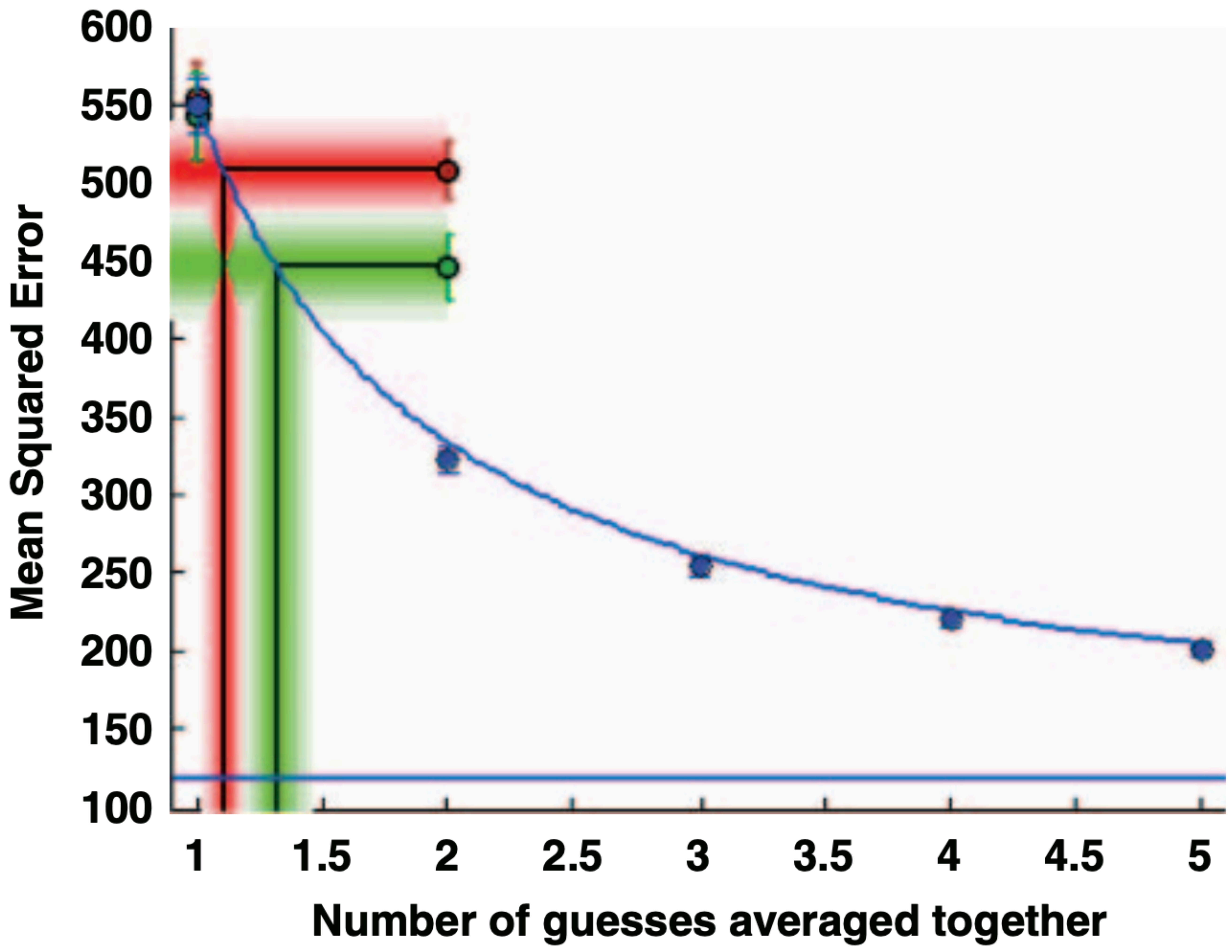
Why?

Averaging two guess from the same person is better than their best guess

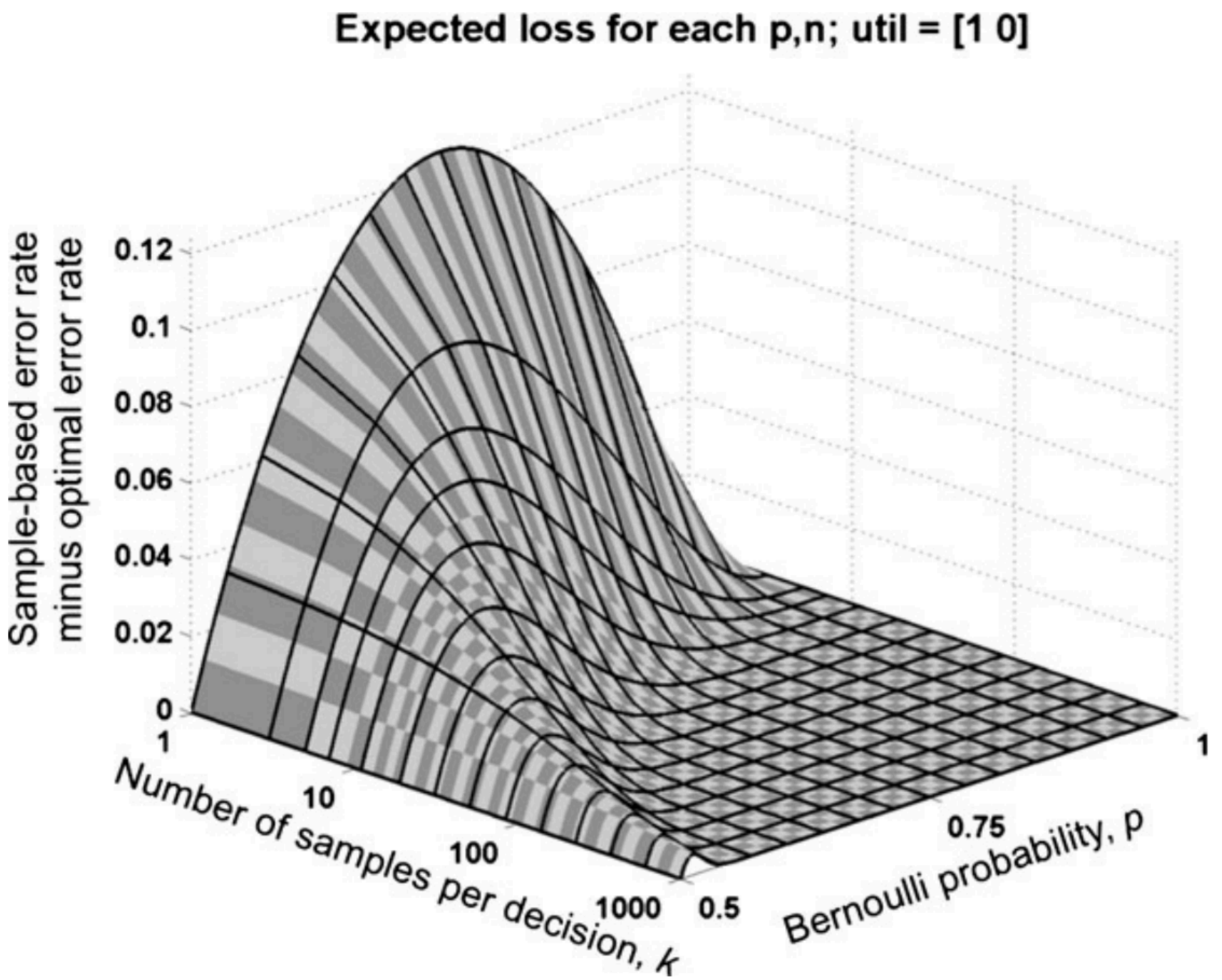


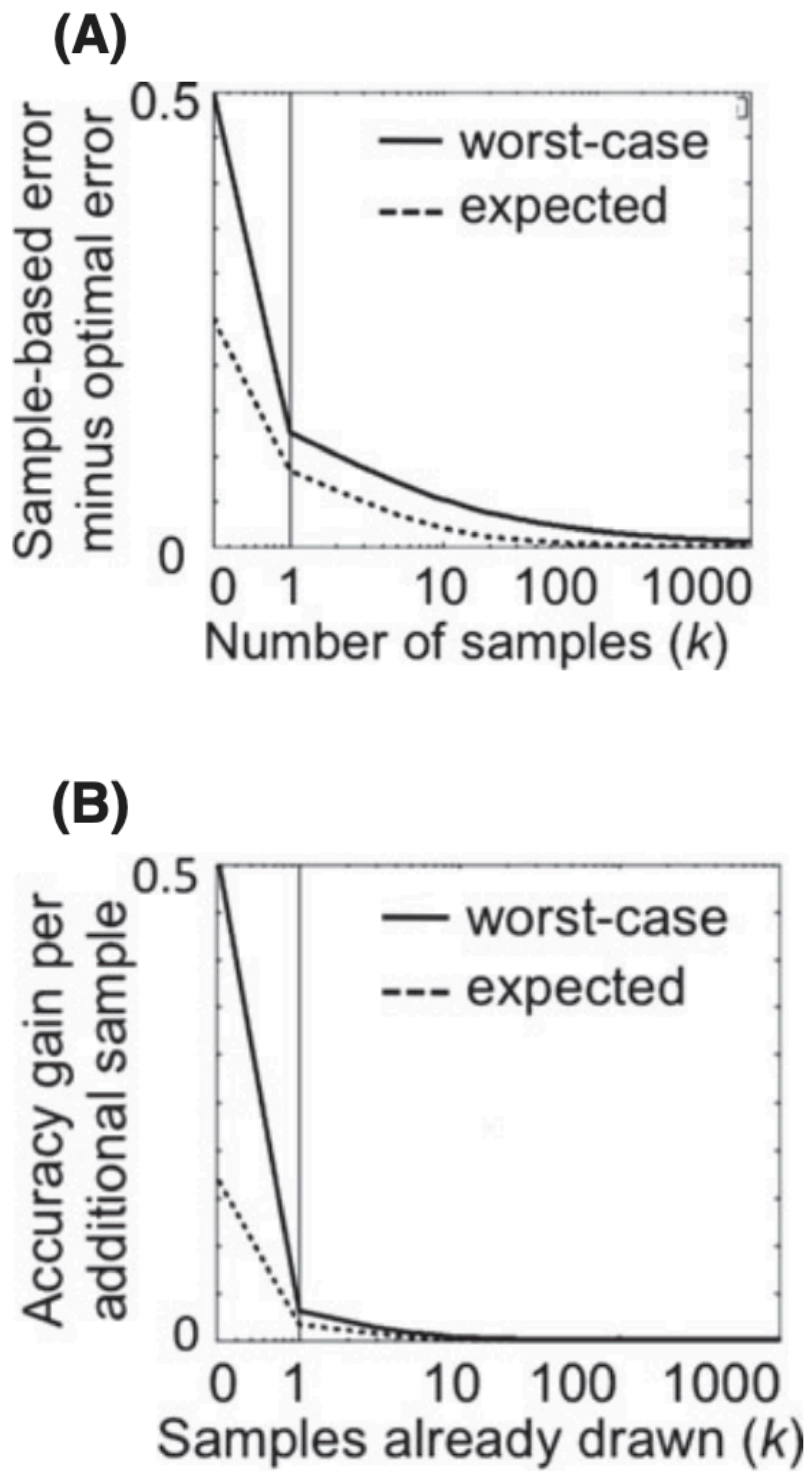
Why is asking after a longer delay better?

Comparing the crowd to the crowd within



Sampling as a rational approximation (Vul, Goodman, Griffiths, & Tenenbaum, 2014)





Inference by sampling

1. Sampling algorithms like Markov chain Monte Carlo (MCMC) can be used to approximate Bayesian inference

2. Markov chain Monte Carlo can be used to uncover people's mental representations

3. Sampling may be how the mind works at Marr's algorithmic level