Markov chain Monte Carlo

Roadmap:

— Monte Carlo basics

— What is MCMC?

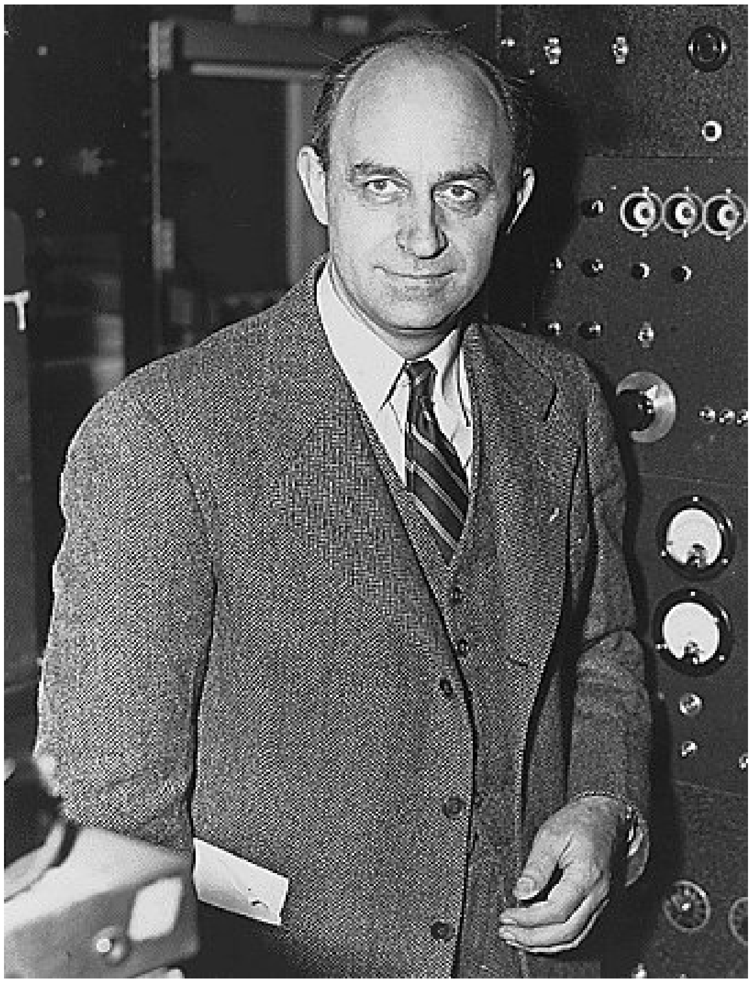
— Gibbs and Metropolis–Hastings

— Practical details

Iain Murray

http://iainmurray.net/

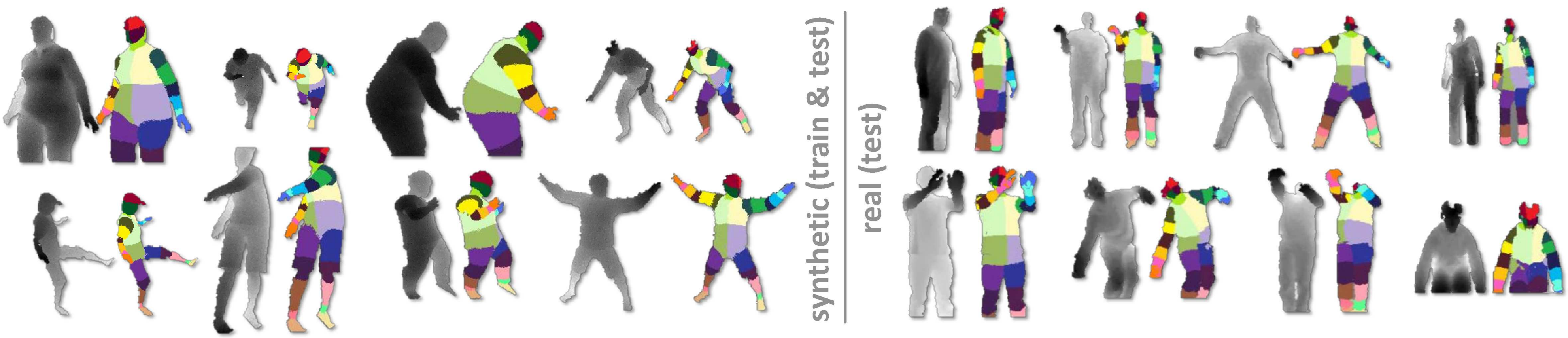
Monte Carlo and Insomnia



Enrico Fermi (1901–1954) took great delight in astonishing his colleagues with his remakably accurate predictions of experimental results. . . he revealed that his "guesses" were really derived from the statistical sampling techniques that he used to calculate with whenever insomnia struck in the wee morning hours!

—*The beginning of the Monte Carlo method*, N. Metropolis

Microsoft Kinect (Shotton et al., 2011)



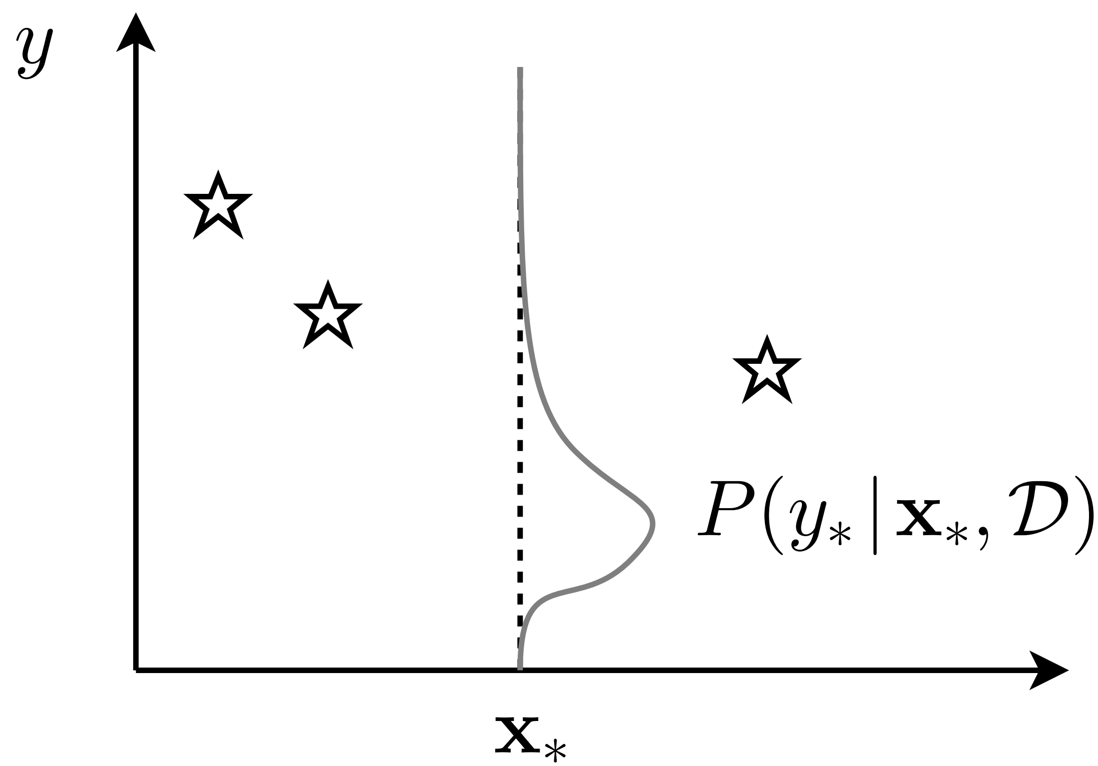
Eyeball modelling assumptions

Generate training data

Random forest applied to fantasies

The need for integrals

$$P(y_*\,|\,\mathbf{x}_*, \mathcal{D}) = \int \mathrm{d}\theta\; P(y_*, \theta\,|\,\mathbf{x}_*, \mathcal{D})$$
$$= \int \mathrm{d}\theta\; P(y_*\,|\,\theta, \cancel{\mathcal{D}})\; P(\theta\,|\,\cancel{\mathbf{x}_*}, \mathcal{D})$$



A statistical problem

What is the average height of the people in this room?

Method: measure our heights, add them up and divide by N.

What is the average height f of people p in London $\mathcal{L}$?

$$E_{p\in\mathcal{L}}[f(p)] \equiv \frac{1}{|\mathcal{L}|}\sum_{p\in\mathcal{L}} f(p), \quad \text{“intractable”?}$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} f\big(p^{(s)}\big), \quad \text{for random survey of } S \text{ people } \{p^{(s)}\} \in \mathcal{L}$$

Surveying works for large and notionally infinite populations.

Simple Monte Carlo

Statistical sampling can be applied to any expectation:

In general:

$$\int f(x) P(x)\, \mathrm{d}x \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$$

Example: making predictions

$$\begin{aligned} p(x|\mathcal{D}) &= \int P(x|\theta, \mathcal{D})\, P(\theta|\mathcal{D})\, \mathrm{d}\theta \\ &\approx \frac{1}{S}\sum_{s=1}^{S} P(x|\theta^{(s)}, \mathcal{D}), \quad \theta^{(s)} \sim P(\theta|\mathcal{D}) \end{aligned}$$

More examples: E-step statistics in EM, Boltzmann machine learning

Properties of Monte Carlo

Estimator: $\displaystyle \int f(x)\, P(x)\, \mathrm{d}x \;\approx\; \hat{f} \;\equiv\; \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$

Estimator is unbiased:

$$\mathbb{E}_{P(\{x^{(s)}\})}\left[\hat{f}\right] \;=\; \frac{1}{S}\sum_{s=1}^{S} \mathbb{E}_{P(x)}[f(x)] \;=\; \mathbb{E}_{P(x)}[f(x)]$$

Variance shrinks $\propto 1/S$:

$$\mathrm{var}_{P(\{x^{(s)}\})}\left[\hat{f}\right] \;=\; \frac{1}{S^2}\sum_{s=1}^{S} \mathrm{var}_{P(x)}[f(x)] \;=\; \mathrm{var}_{P(x)}[f(x)]\,/S$$

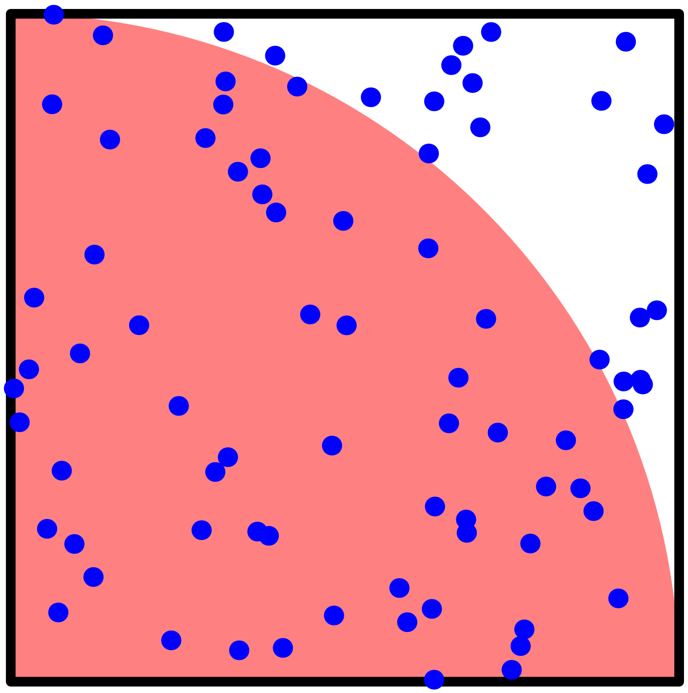
"Error bars" shrink like $\sqrt{S}$

Aside: don't always sample!

"Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse."

— Alan Sokal, 1996

A dumb approximation of π



$$P(x,y) = \begin{cases} 1 & 0<x<1 \text{ and } 0<y<1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}\left((x^2+y^2)<1\right) P(x,y)\, \mathrm{d}x\, \mathrm{d}y$$

```
octave:1> S=12; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.3333
octave:2> S=1e7; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.1418
```

Alternatives to Monte Carlo

There are other methods of numerical integration!

Example: (nice) 1D integrals are easy:

```
octave:1> 4 * quadl(@(x) sqrt(1-x.^2), 0, 1, tolerance)
```

Gives π to 6 dp's in 108 evaluations, machine precision in 2598.

(NB Matlab's `quadl` fails at `tolerance=0`, but Octave works.)

In higher dimensions sometimes determinstic approximations work:
Variational Bayes, EP, INLA, . . .

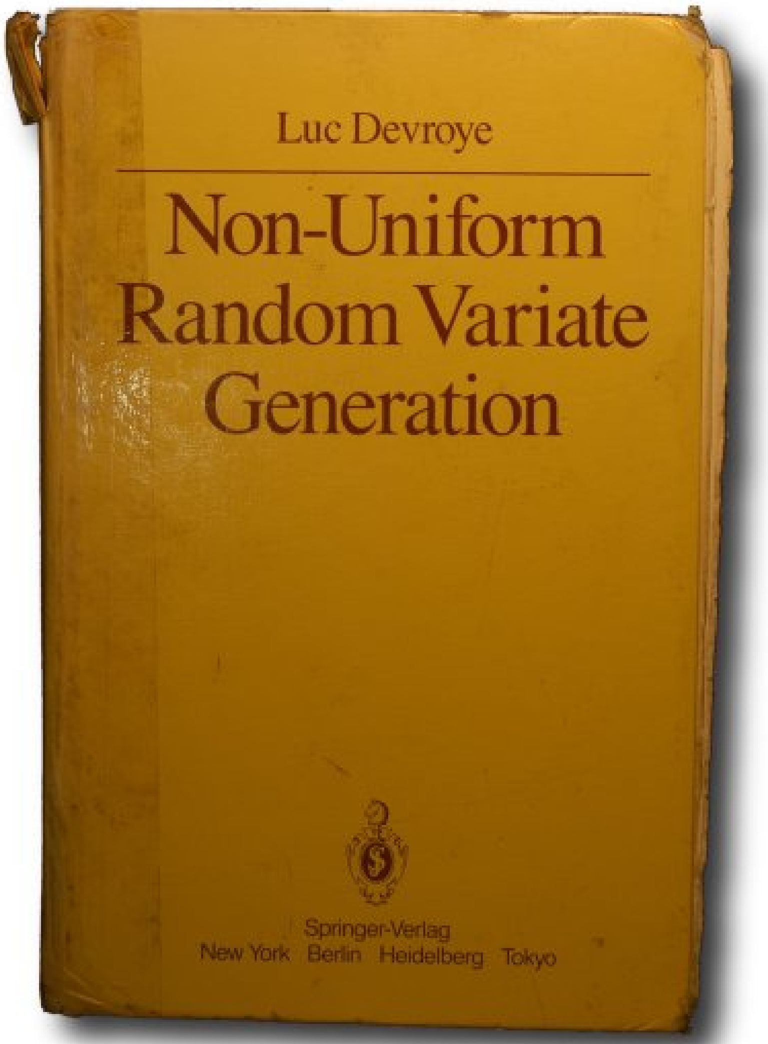
Reminder

Want to sample to approximate expectations:

$$\int f(x)P(x)\,\mathrm{d}x \;\approx\; \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$$

How do we get the samples?

Sampling simple distributions



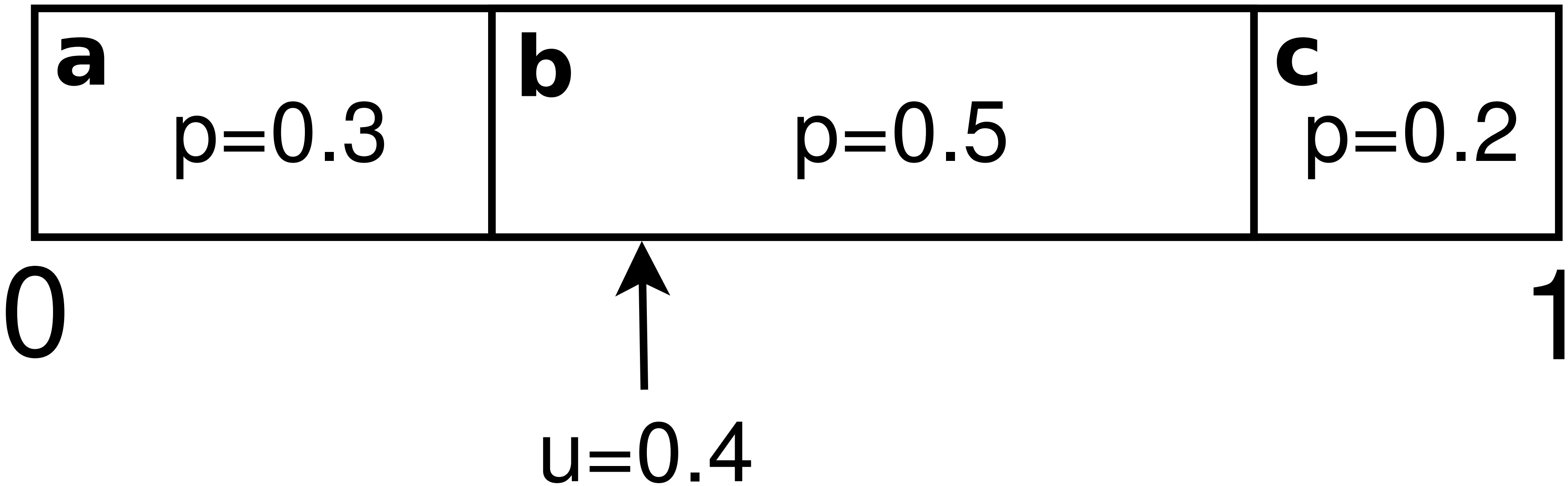
Use library routines for univariate distributions

(and some other special cases)

This book (free online) explains how some of them work

`http://cg.scs.carleton.ca/~luc/rnbookindex.html`

Sampling discrete values



$$u \sim \text{Uniform}[0,1]$$

$$u = 0.4 \quad \Rightarrow \quad x = \mathsf{b}$$

There are more efficient ways for large numbers of values and samples. See Devroye book.

Sampling from densities

How to convert samples from a Uniform[0,1] generator:

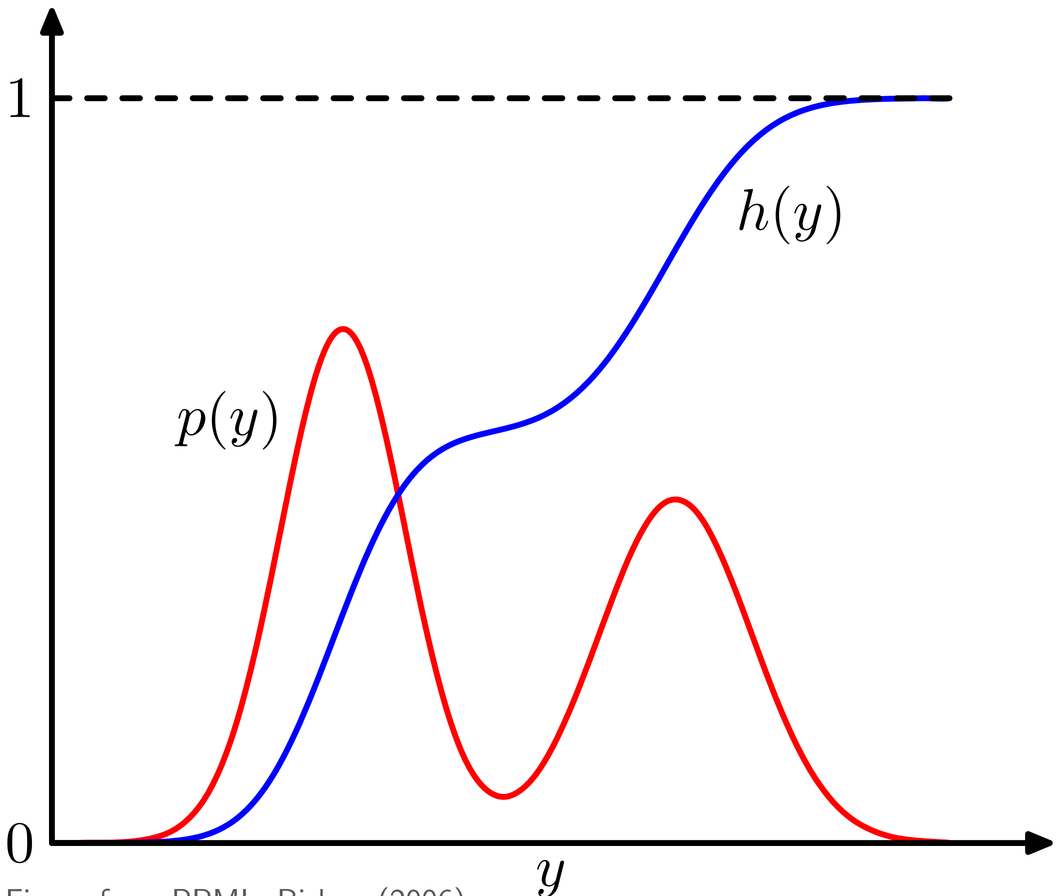


Figure from PRML, Bishop (2006)

$$h(y) = \int_{-\infty}^{y} p(y')\,\mathrm{d}y'$$

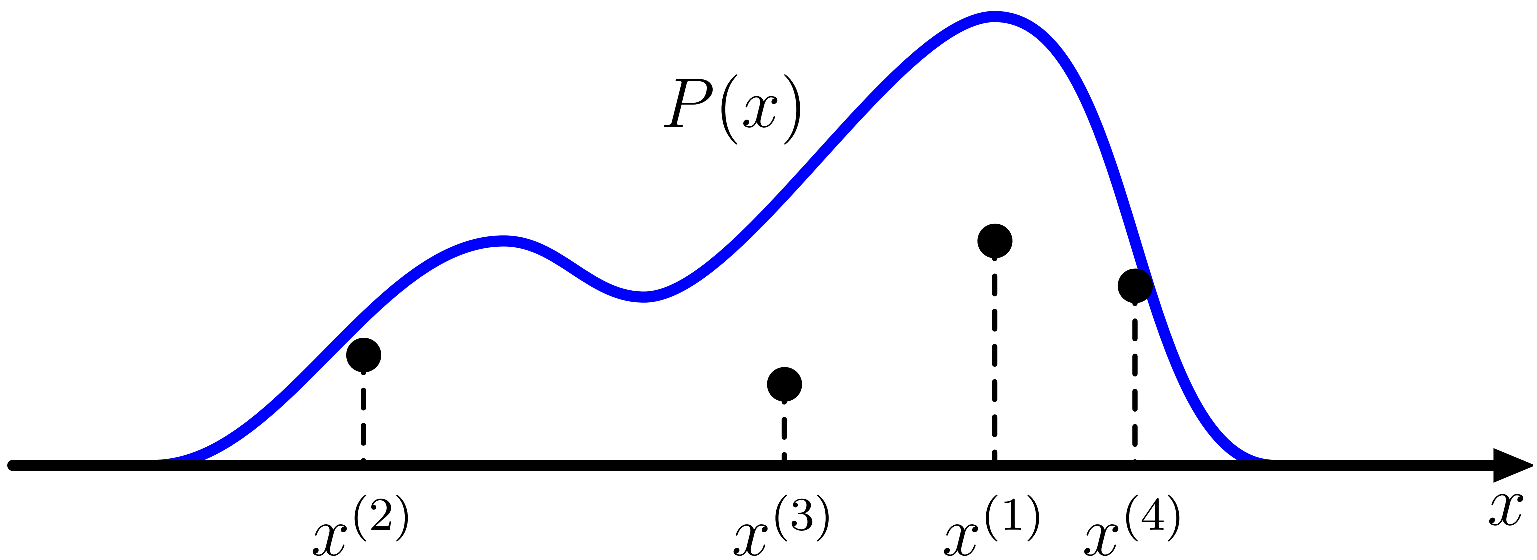
$u \sim$ Uniform[0,1]

Sample, $y(u) = h^{-1}(u)$

Although we can't always compute and invert $h(y)$

Sampling from densities

Draw points uniformly under the curve:



Probability mass to left of point $\sim$ Uniform[0,1]

Rejection sampling

Sampling from $\pi(x)$ using tractable $q(x)$:

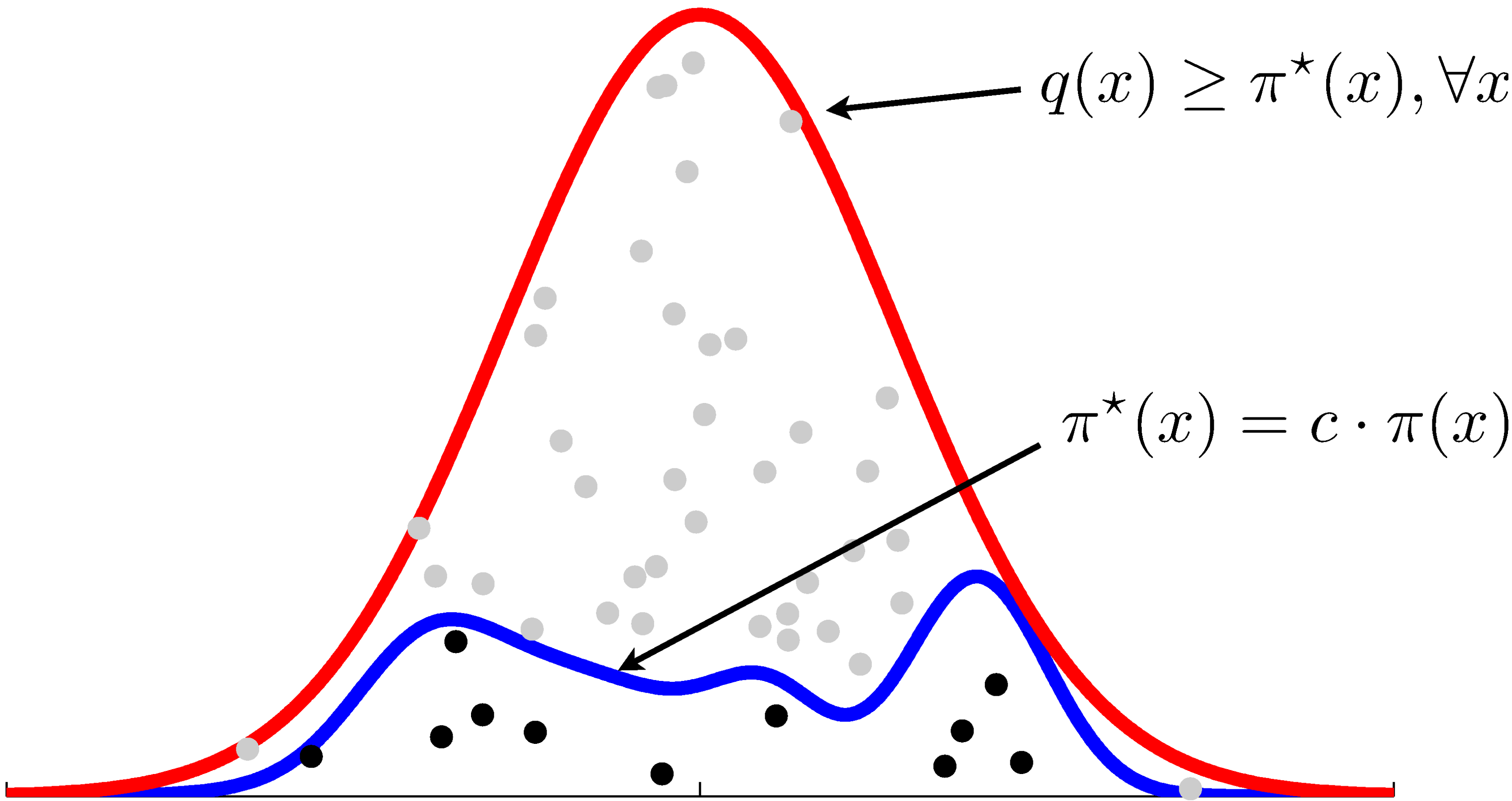


Figure credit: Ryan P. Adams

Importance sampling

Rewrite integral: expectation under simple distribution Q:

$$\int f(x)\,P(x)\,\mathrm{d}x \;=\; \int f(x)\,\frac{P(x)}{Q(x)}\,Q(x)\,\mathrm{d}x,$$

$$\approx\; \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\,\frac{P(x^{(s)})}{Q(x^{(s)})}, \qquad x^{(s)} \sim Q(x)$$

Simple Monte Carlo applied to any integral.

Unbiased and independent of dimension?

Importance sampling (2)

If only know $P(x) = P^*(x)/\mathcal{Z}_P$ up to constant:

$$\int f(x)\, P(x)\, \mathrm{d}x \;\approx\; \frac{\mathcal{Z}_Q}{\mathcal{Z}_P}\, \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})\, \underbrace{\frac{P^*(x^{(s)})}{Q^*(x^{(s)})}}_{w^{*(s)}}, \quad x^{(s)} \sim Q(x)$$

$$\approx\; \cancel{\frac{1}{S}} \sum_{s=1}^{S} f(x^{(s)})\, \frac{w^{*(s)}}{\cancel{\frac{1}{S}} \sum_{s'} w^{*(s')}}$$

This estimator is **consistent** but **biased**

Exercise: Prove that $\mathcal{Z}_P/\mathcal{Z}_Q \approx \frac{1}{S} \sum_s w^{*(s)}$

Summary so far

- **Monte Carlo**
  approximate expectations with a sample average
- **Rejection sampling**
  draw samples from complex distributions
- **Importance sampling**
  apply Monte Carlo to 'any' sum/integral

Next: High dimensional problems: MCMC

Application to large problems

Approximations scale badly with dimensionality

Example: $P(x) = \mathcal{N}(0, \mathbb{I}), \quad Q(x) = \mathcal{N}(0, \sigma^2\mathbb{I})$

Rejection sampling:

Requires $\sigma \geq 1$. Fraction of proposals accepted $= \sigma^{-D}$

Importance sampling:

$\mathrm{Var}[P(x)/Q(x)] \;=\; \left(\frac{\sigma^2}{2-1/\sigma^2}\right)^{D/2} - 1$

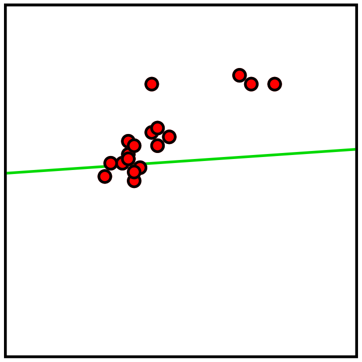
Infinite / undefined variance if $\sigma \leq 1/\sqrt{2}$

Reminder

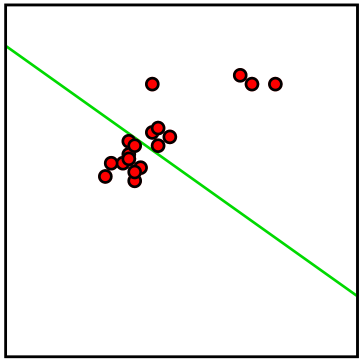
Need to sample large, non-standard distributions:

$$P(x\,|\,\mathcal{D}) \;\approx\; \frac{1}{S}\sum_{s=1}^{S} P(x\,|\,\theta), \quad \theta \sim P(\theta\,|\,\mathcal{D}) = \frac{P(\mathcal{D}\,|\,\theta)\, P(\theta)}{P(\mathcal{D})}$$

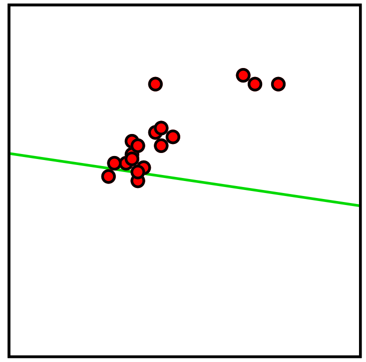
Importance sampling weights



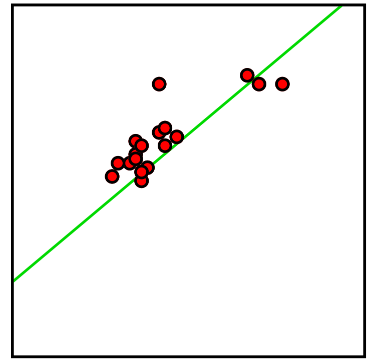
$w = 0.00548$



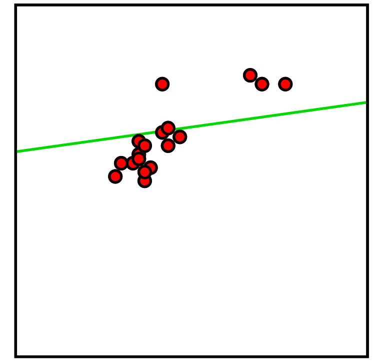
$w = 1.59\text{e-}08$



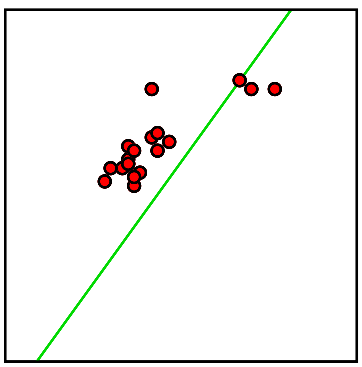
$w = 9.65\text{e-}06$



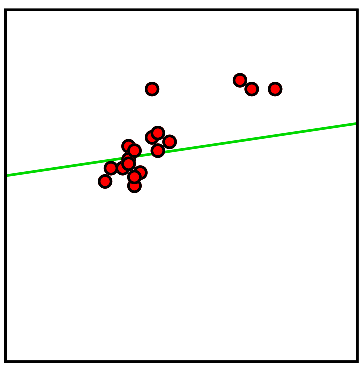
$w = 0.371$



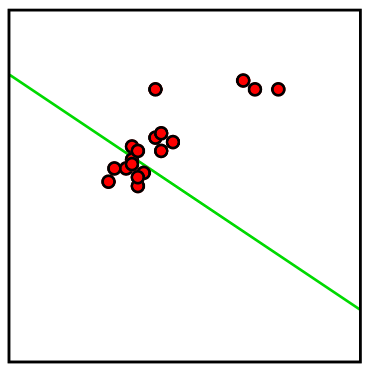
$w = 0.103$



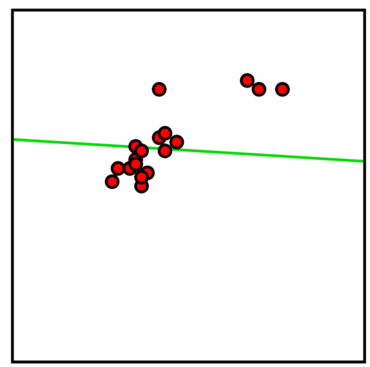
$w = 1.01\text{e-}08$



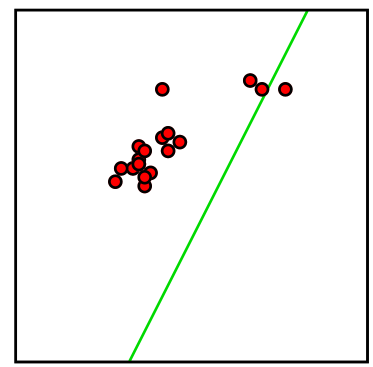
$w = 0.111$



$w = 1.92\text{e-}09$

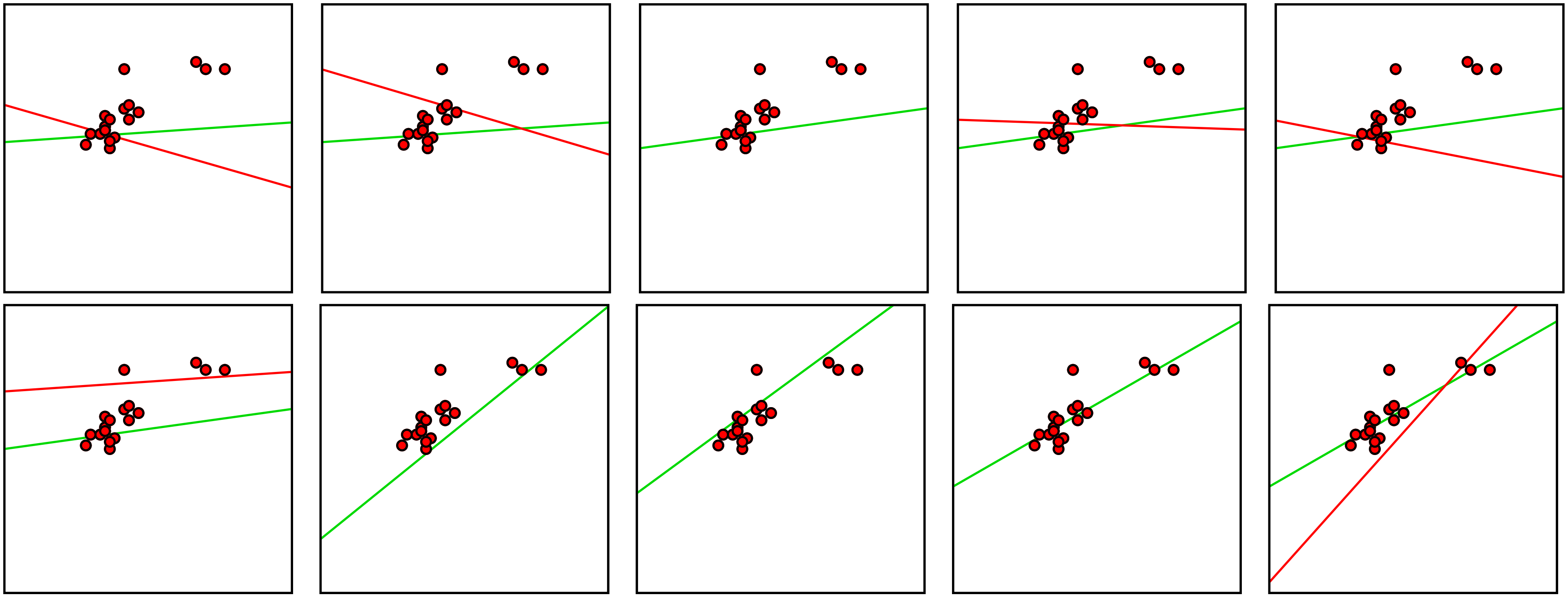


$w = 0.0126$

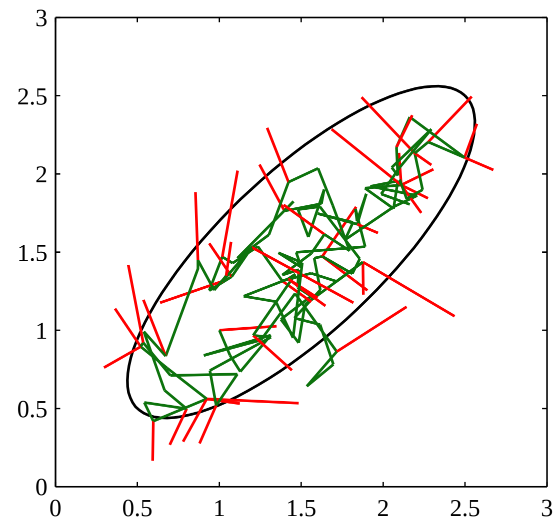


$w = 1.1\text{e-}51$

Metropolis algorithm



- Perturb parameters: $Q(\theta';\theta)$, e.g. $\mathcal{N}(\theta,\sigma^2)$
- Accept with probability $\min\left(1, \dfrac{\tilde{P}(\theta'|\mathcal{D})}{\tilde{P}(\theta|\mathcal{D})}\right)$
- Otherwise **keep old parameters**



This subfigure from PRML, Bishop (2006)

Detail: Metropolis, as stated, requires $Q(\theta';\theta) = Q(\theta;\theta')$

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

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)

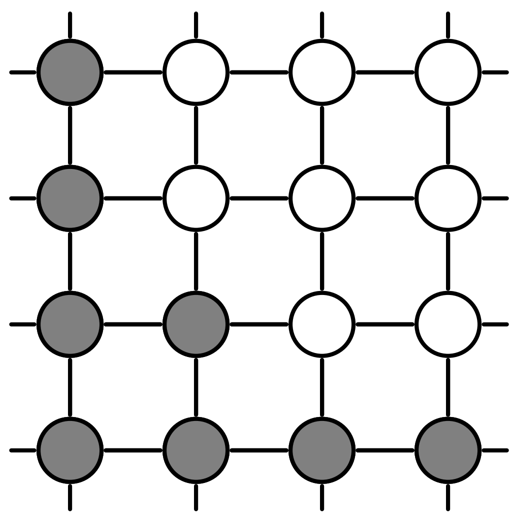
THE purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed,

$>$20,000 citations

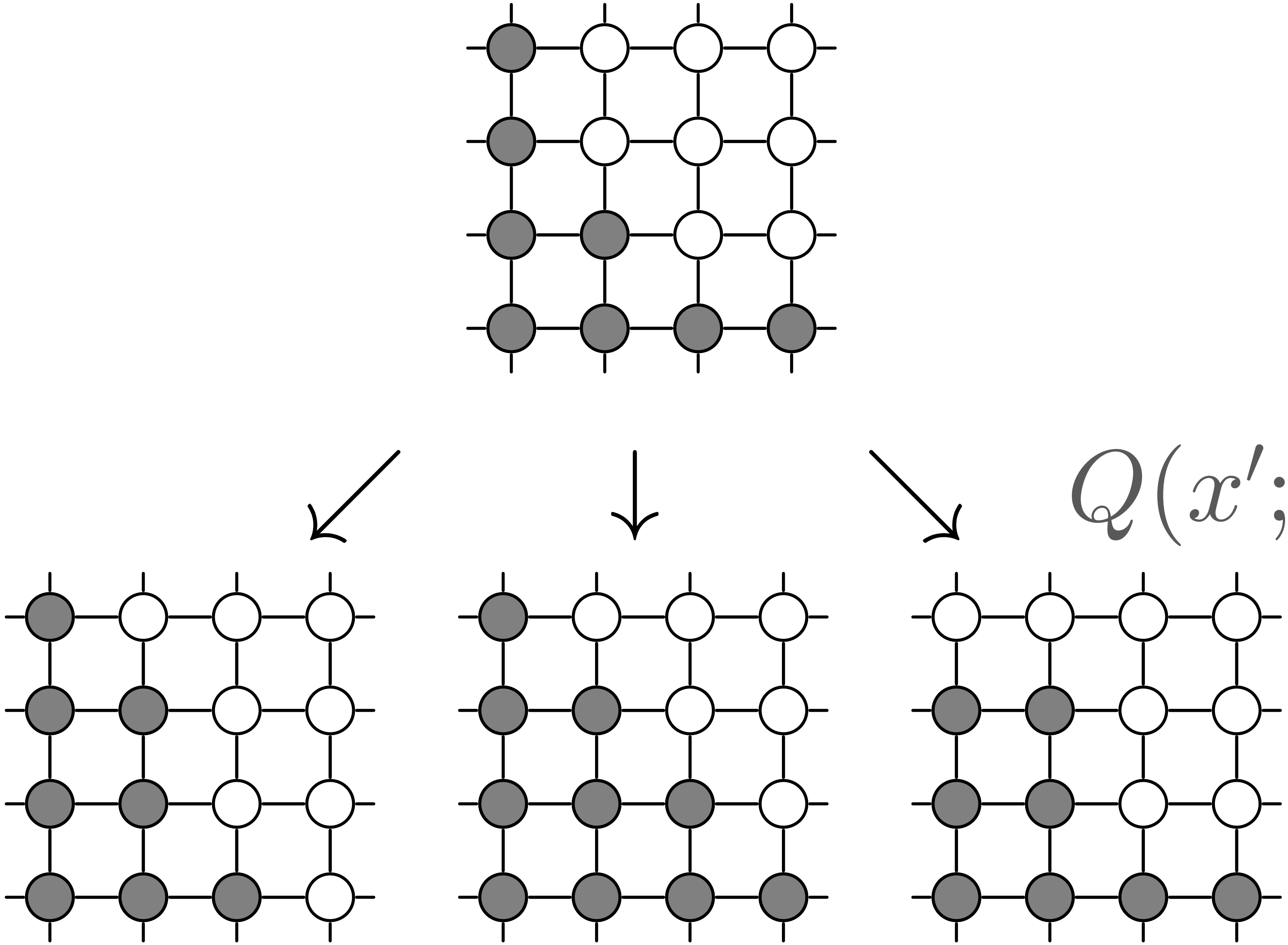
Target distribution

$$P(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}$$

e.g. $\mathbf{x} =$

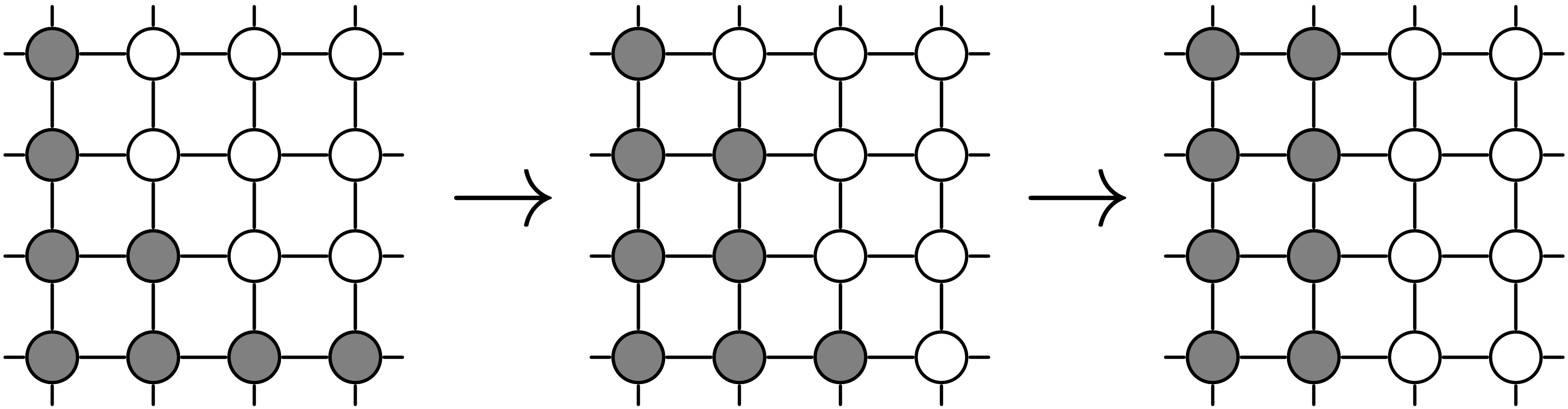


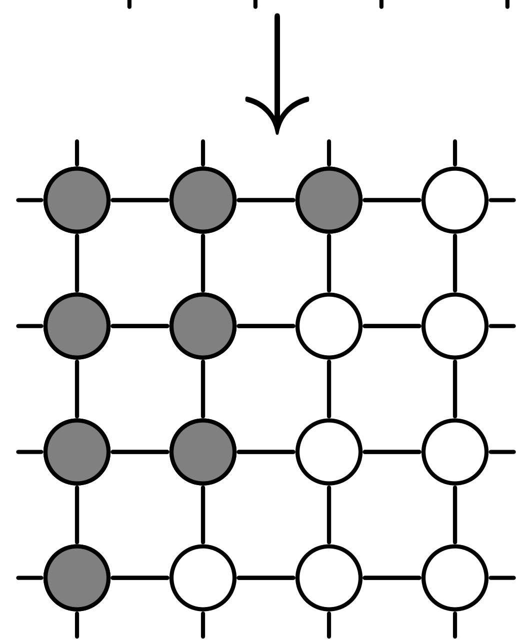
Local moves



$$Q(x'; x)$$

Markov chain exploration





Goal: a Markov chain,

$x_t \sim T(x_t \leftarrow x_{t-1})$, such that:

$P(x^{(t)}) = e^{-E(x^{(t)})}/Z$ for large t.

Invariant/stationary condition

If $x^{(t-1)}$ is a sample from P,

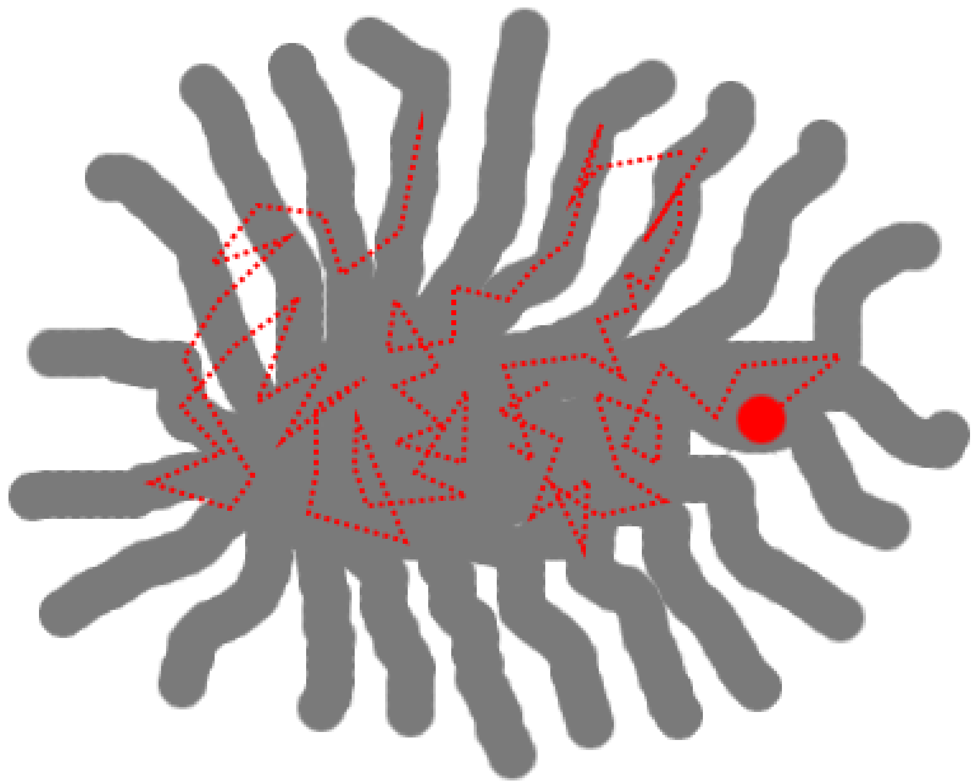
$x^{(t)}$ is also a sample from P.

$$\sum_{x} T(x' \leftarrow x) P(x) = P(x')$$

Ergodicity

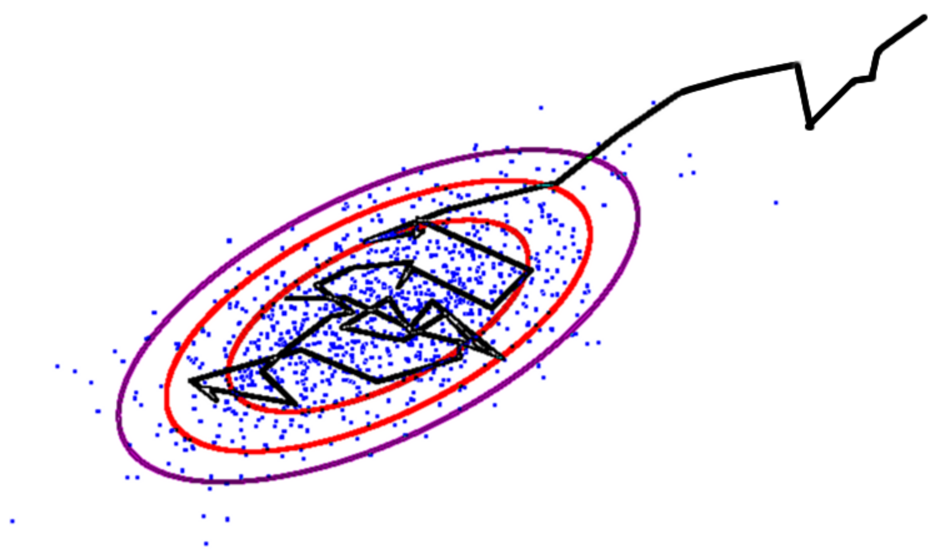
Unique invariant distribution

if 'forget' starting point, $x^{(0)}$



Quick review

MCMC: biased random walk exploring a target dist.



Markov steps,

$$x^{(s)} \sim T\left(x^{(s)} \leftarrow x^{(s-1)}\right)$$

MCMC gives approximate, correlated samples

$$\mathbb{E}_P[f] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})$$

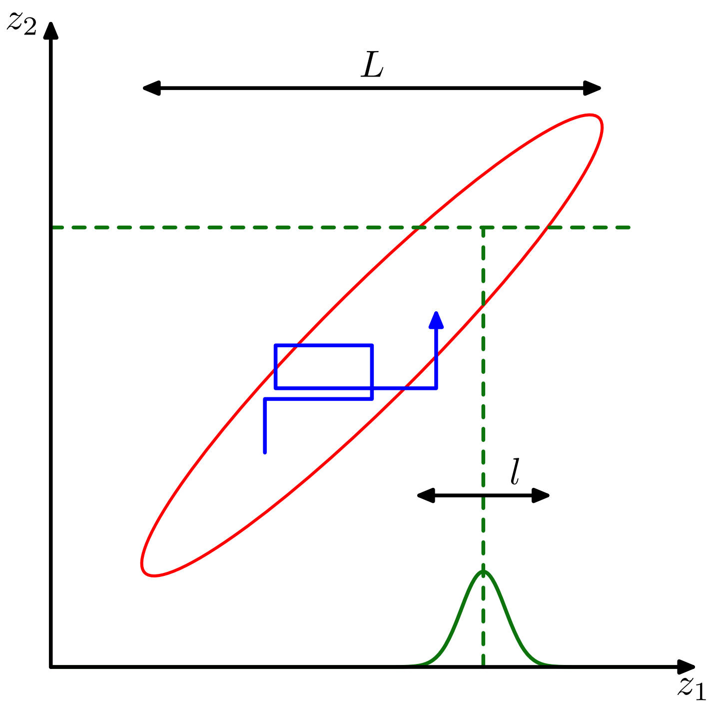
T must leave target invariant

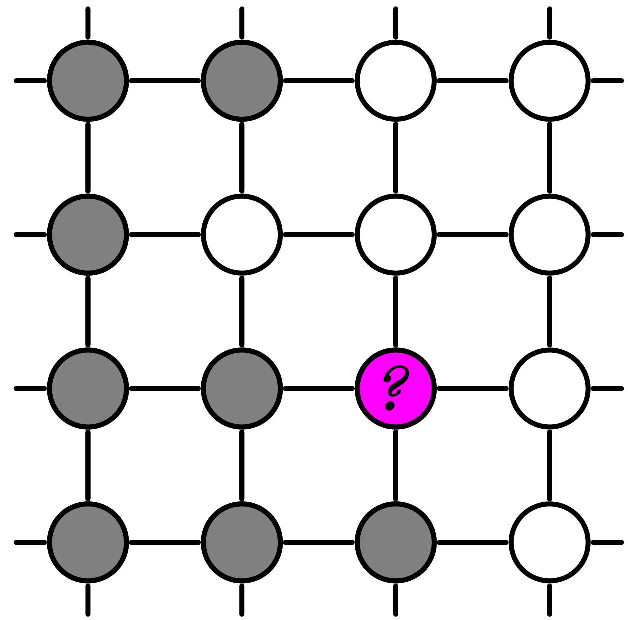
T must be able to get everywhere in K steps

Gibbs sampling

Pick variables in turn or randomly,

and resample $P(x_i|\mathbf{x}_{j\neq i})$





$$T_i(\mathbf{x}' \leftarrow \mathbf{x}) = P(x_i' \,|\, \mathbf{x}_{j\neq i})\, \delta(\mathbf{x}'_{j\neq i} - \mathbf{x}_{j\neq i})$$

Gibbs sampling correctness

$$P(\mathbf{x}) = P(x_i \,|\, \mathbf{x}_{\setminus i})\, P(\mathbf{x}_{\setminus i})$$

Simulate by drawing $\mathbf{x}_{\setminus i}$, then $x_i \,|\, \mathbf{x}_{\setminus i}$

Draw $\mathbf{x}_{\setminus i}$: sample $\mathbf{x}$, throw initial x_i away

Reverse operators

If T leaves $P(x)$ stationary, define a *reverse operator*

$$R(x \leftarrow x') = \frac{T(x' \leftarrow x)\, P(x)}{\sum_x T(x' \leftarrow x)\, P(x)} = \frac{T(x' \leftarrow x)\, P(x)}{P(x')}.$$

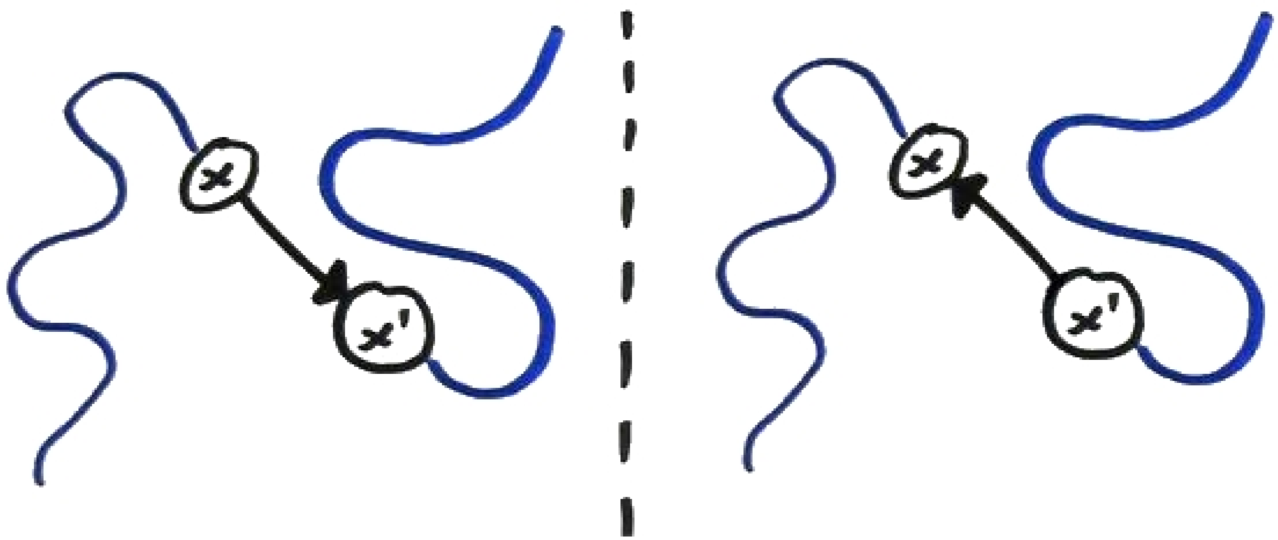
A necessary condition: there exists R such that:

$$T(x' \leftarrow x)\, P(x) \;=\; R(x \leftarrow x')\, P(x'), \qquad \forall x, x'.$$

If $R = T$, known as **detailed balance** (not necessary)

Balance condition

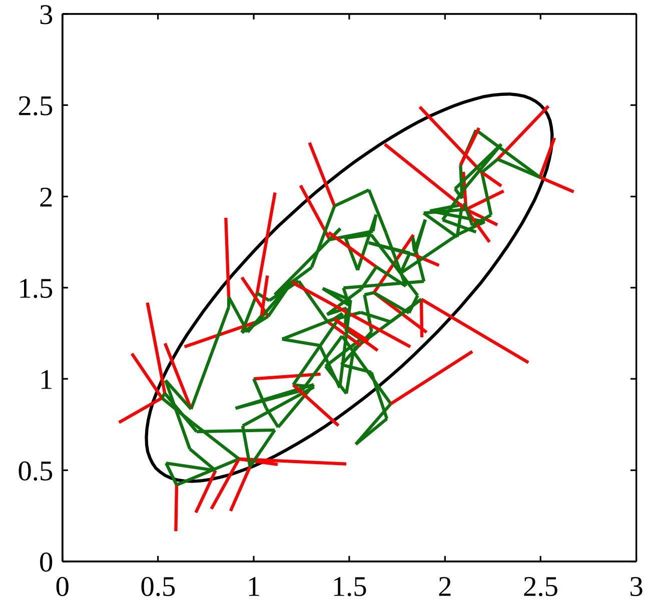
$$T(x' \leftarrow x)\, P(x) \;=\; R(x \leftarrow x')\, P(x')$$



Implies that $P(x)$ is left invariant:

$$\sum_x T(x' \leftarrow x)\, P(x) \;=\; P(x') \cancelto{1}{\sum_x R(x \leftarrow x')}$$

Metropolis–Hastings



PRML, Bishop (2006)

Arbitrary proposals $\sim Q$:

$$Q(x'; x)\, P(x) \;\neq\; Q(x; x')\, P(x')$$

Satisfies detailed balance by rejecting moves:

$$T(x' \leftarrow x) = \begin{cases} Q(x'; x) \min\!\left(1,\; \frac{P(x')\, Q(x; x')}{P(x)\, Q(x'; x)}\right) & x' \neq x \\ \ldots & x' = x \end{cases}$$

Metropolis–Hastings

Transition operator

- Propose a move from the current state $Q(x'; x)$, e.g. $\mathcal{N}(x, \sigma^2)$
- Accept with probability $\min\left(1, \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right)$
- Otherwise next state in chain is a copy of current state

Notes

- Can use $P^* \propto P(x)$; normalizer cancels in acceptance ratio
- Satisfies detailed balance (shown below)
- Q must be chosen so chain is ergodic

$$P(x) \cdot T(x' \leftarrow x) = P(x) \cdot Q(x'; x) \min\left(1, \frac{P(x')Q(x; x')}{P(x)Q(x'; x)}\right) = \min\Big(P(x)Q(x'; x),\ P(x')Q(x; x')\Big)$$
$$= P(x') \cdot Q(x; x') \min\left(1, \frac{P(x)Q(x'; x)}{P(x')Q(x; x')}\right) = P(x') \cdot T(x \leftarrow x')$$

Matlab/Octave code for demo

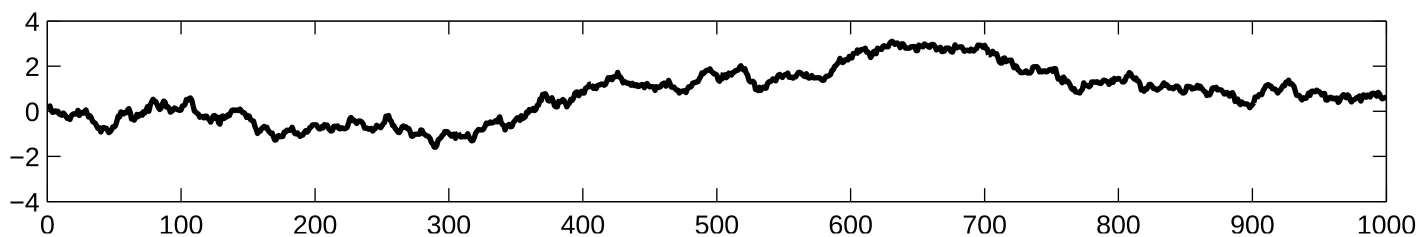
```
function samples = dumb_metropolis(init, log_ptilde, iters, sigma)

D = numel(init);
samples = zeros(D, iters);

state = init;
Lp_state = log_ptilde(state);
for ss = 1:iters
    % Propose
    prop = state + sigma*randn(size(state));
    Lp_prop = log_ptilde(prop);
    if log(rand) < (Lp_prop - Lp_state)
        % Accept
        state = prop;
        Lp_state = Lp_prop;
    end
    samples(:, ss) = state(:);
end
```

Step-size demo

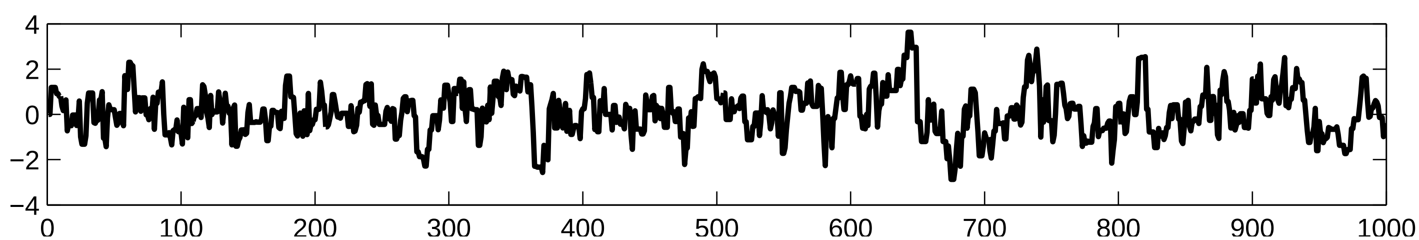
Explore $\mathcal{N}(0,1)$ with different step sizes σ

```
sigma = @(s) plot(dumb_metropolis(0, @(x)-0.5*x*x, 1e3, s));
```

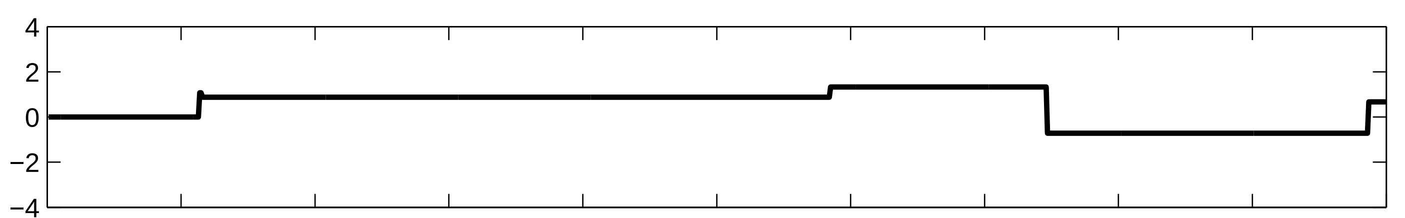
`sigma(0.1)`
99.8% accepts



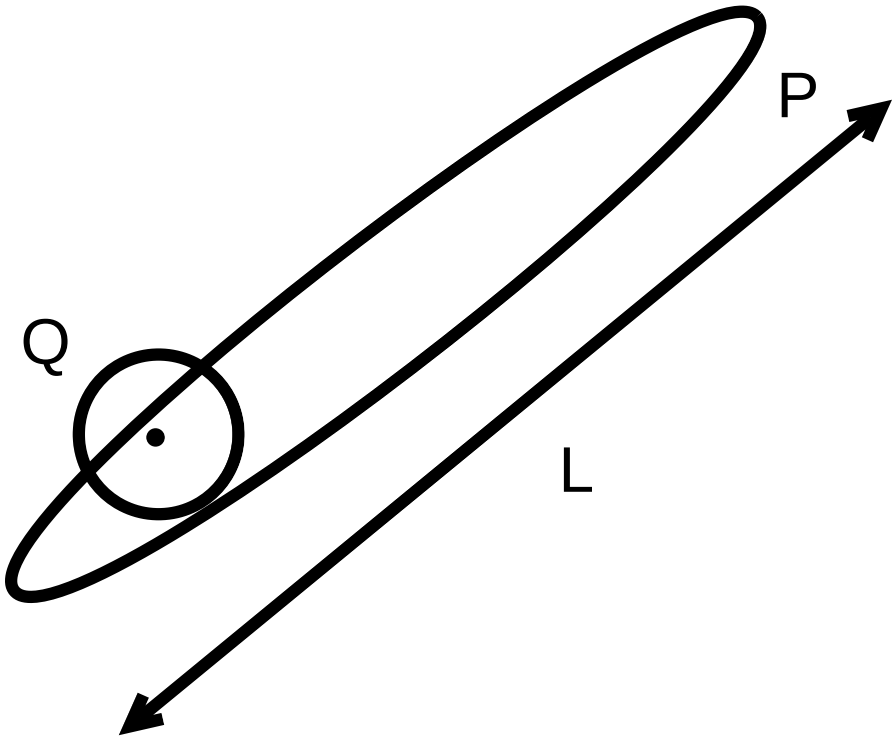
`sigma(1)`
68.4% accepts



`sigma(100)`
0.5% accepts



Diffusion time



Generic proposals use
$Q(x';x) = \mathcal{N}(x, \sigma^2)$

σ **large** $\rightarrow$ **many rejections**

σ **small** $\rightarrow$ **slow diffusion:**
$\sim(L/\sigma)^2$ iterations required

Adapted from MacKay (2003)

An MCMC strategy

Come up with good proposals $Q(x'; x)$

Combine transition operators:

$$
\begin{aligned}
x_1 &\sim T_A(\cdot \leftarrow x_0) \\
x_2 &\sim T_B(\cdot \leftarrow x_1) \\
x_3 &\sim T_C(\cdot \leftarrow x_2) \\
x_4 &\sim T_A(\cdot \leftarrow x_3) \\
x_5 &\sim T_B(\cdot \leftarrow x_4) \\
&\;\;\ldots
\end{aligned}
$$

Summary so far

- We need approximate methods to solve sums/integrals
- Monte Carlo does not explicitly depend on dimension, although simple methods work only in low dimensions
- Markov chain Monte Carlo (MCMC) can make local moves. By assuming less, it's more applicable to higher dimensions
- simple computations $\Rightarrow$ "easy" to implement (harder to diagnose).

http://www.kaggle.com/c/DarkWorlds

WINTON

Observing Dark Worlds

Finished

Friday, October 12, 2012 | $20,000 • 353 teams | Sunday, December 16, 2012

Dashboard

Home
- Data

Information
- Description
- Evaluation
- Rules
- Prizes
- About the Sponsor
- An Introduction to E…
- Getting Started (wit…
- Submission Instructi…
- Winners

Forum

Leaderboard

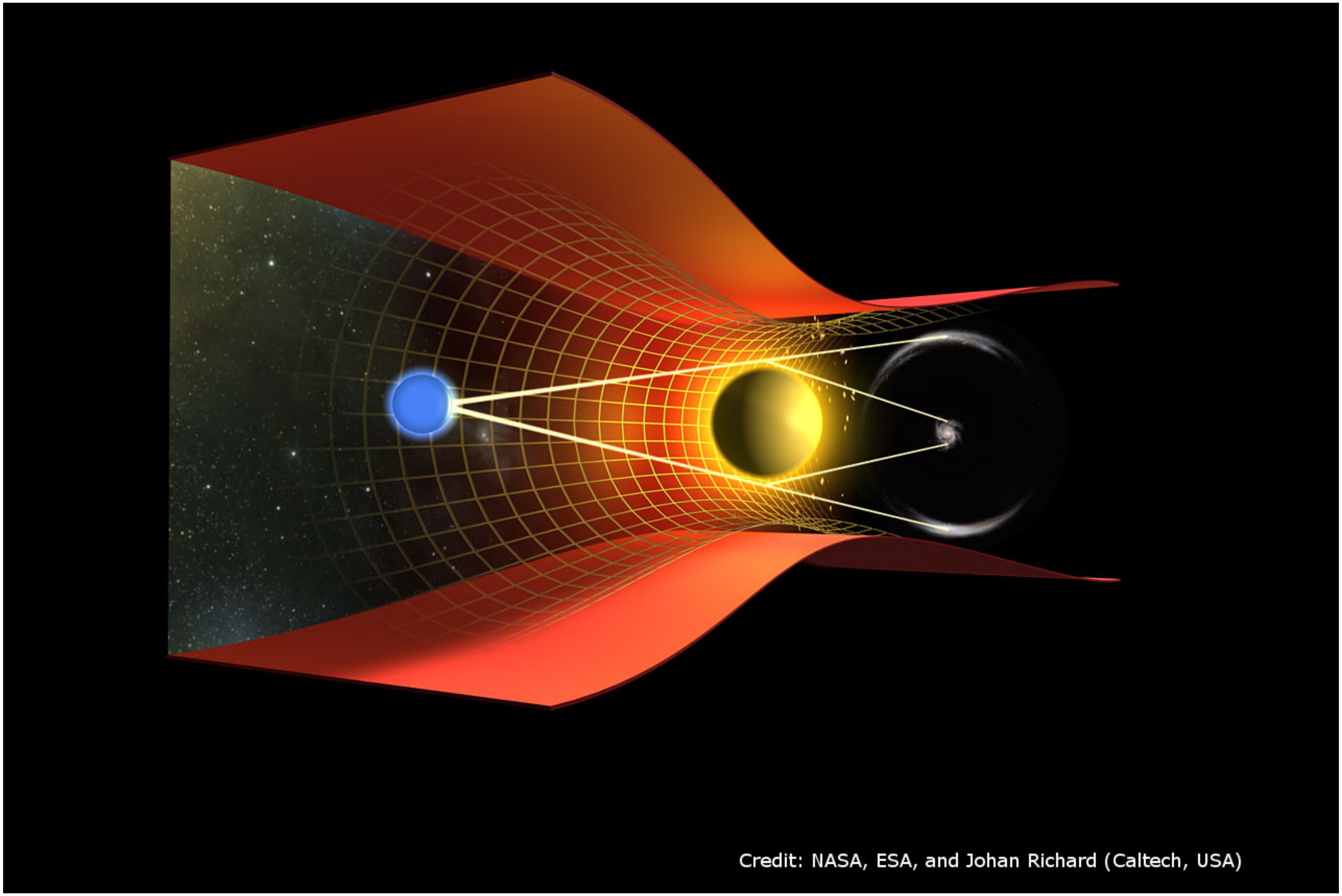
Can you find the Dark Matter that dominates our Universe? Winton Capital offers you the chance to unlock the secrets of dark worlds.

There is more to the Universe than meets the eye. Out in the cosmos exists a form of matter that outnumbers the stuff we can see by almost 7 to 1, and we don't know what it is. What we do know is that it does not emit or absorb light, so we call it ***Dark Matter***.

Such a vast amount of aggregated matter does not go unnoticed. In fact we observe that this stuff aggregates and forms massive structures called ***Dark Matter Halos***.

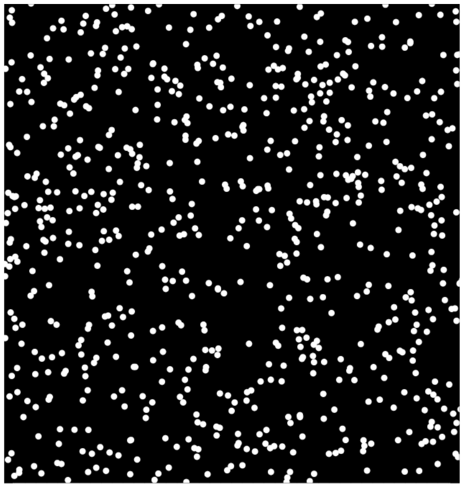
Although dark, it warps and bends spacetime such that any light from a background galaxy which passes close to the *Dark Matter* will have its path altered and changed. This bending causes the galaxy to appear as an ellipse in the sky.

Dark Matter

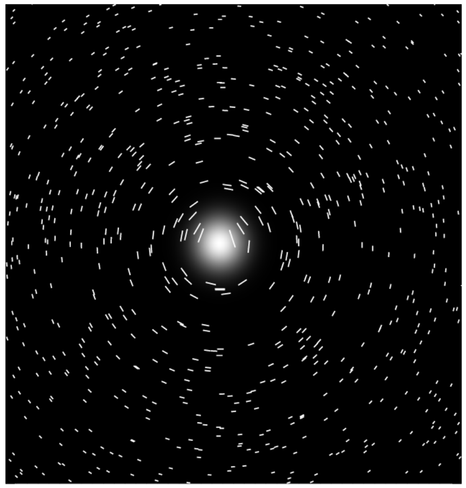


http://www.kaggle.com/c/DarkWorlds

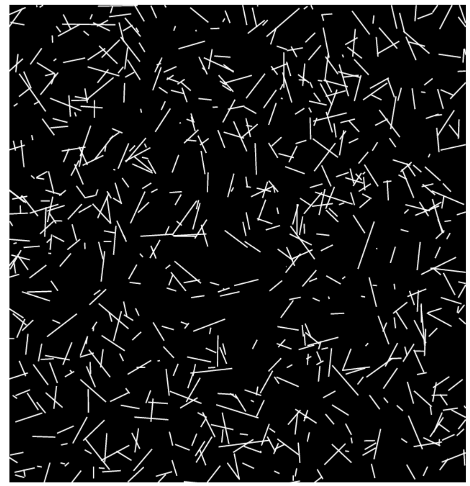
Dark Matter



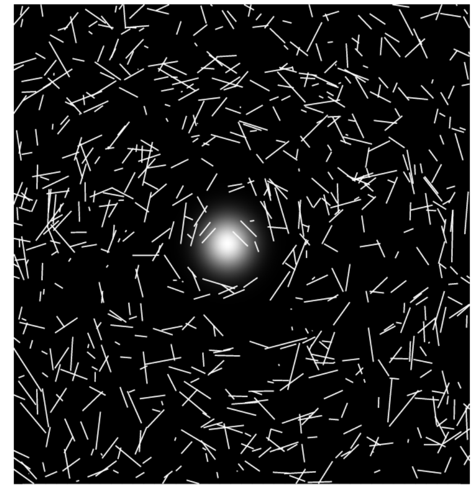
A. Distant circular galaxies (or dots in this case) are randomly distributed in the sky. Each galaxy has an (x,y) coordinate corresponding to the position in the sky from 0:4200



B. By placing a Dark Matter halo in the middle of the sky between us and the background galaxies, they are altered such that they become elliptical. The lines show the orientation and size of the major axis of the galaxy.

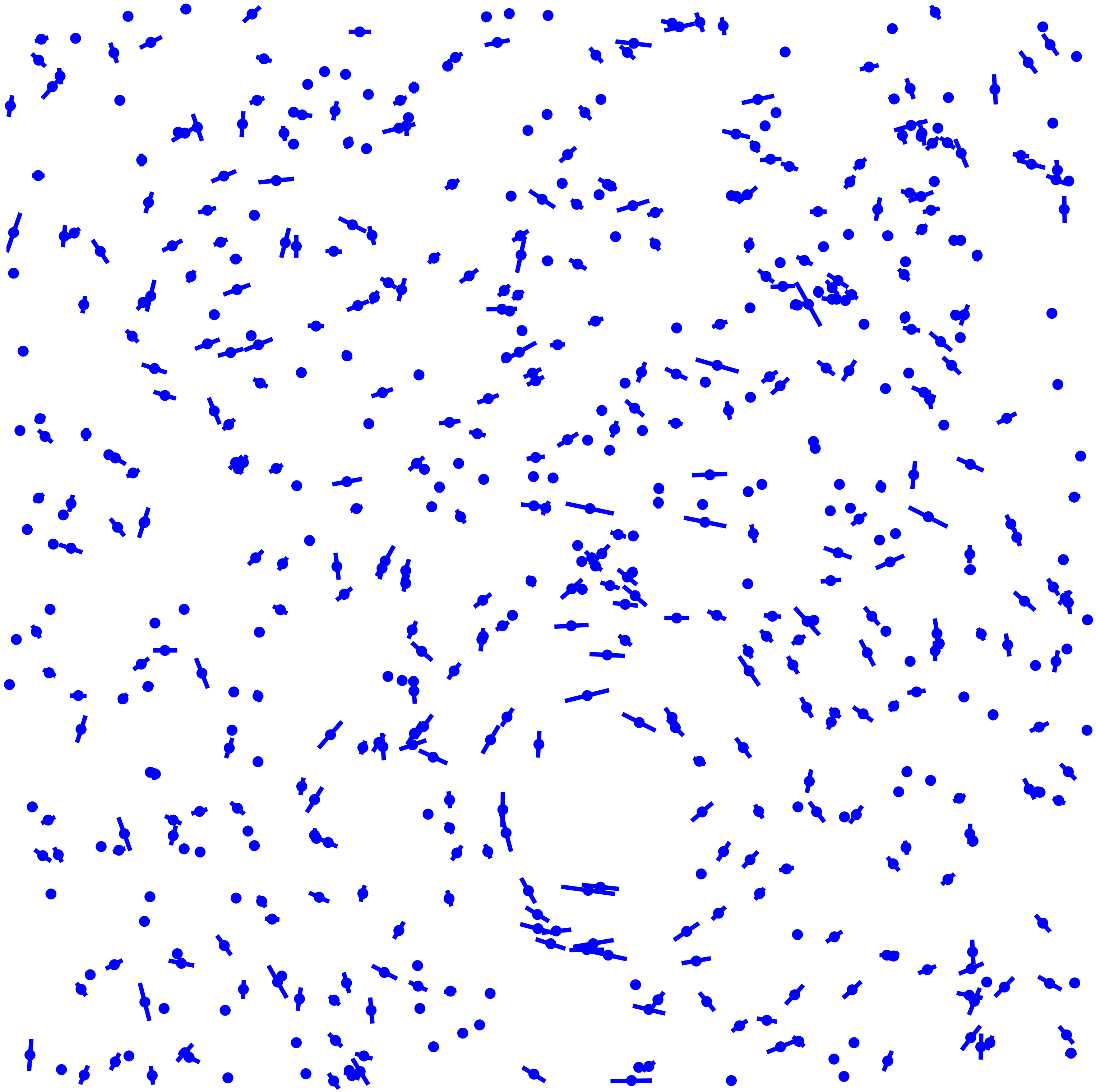


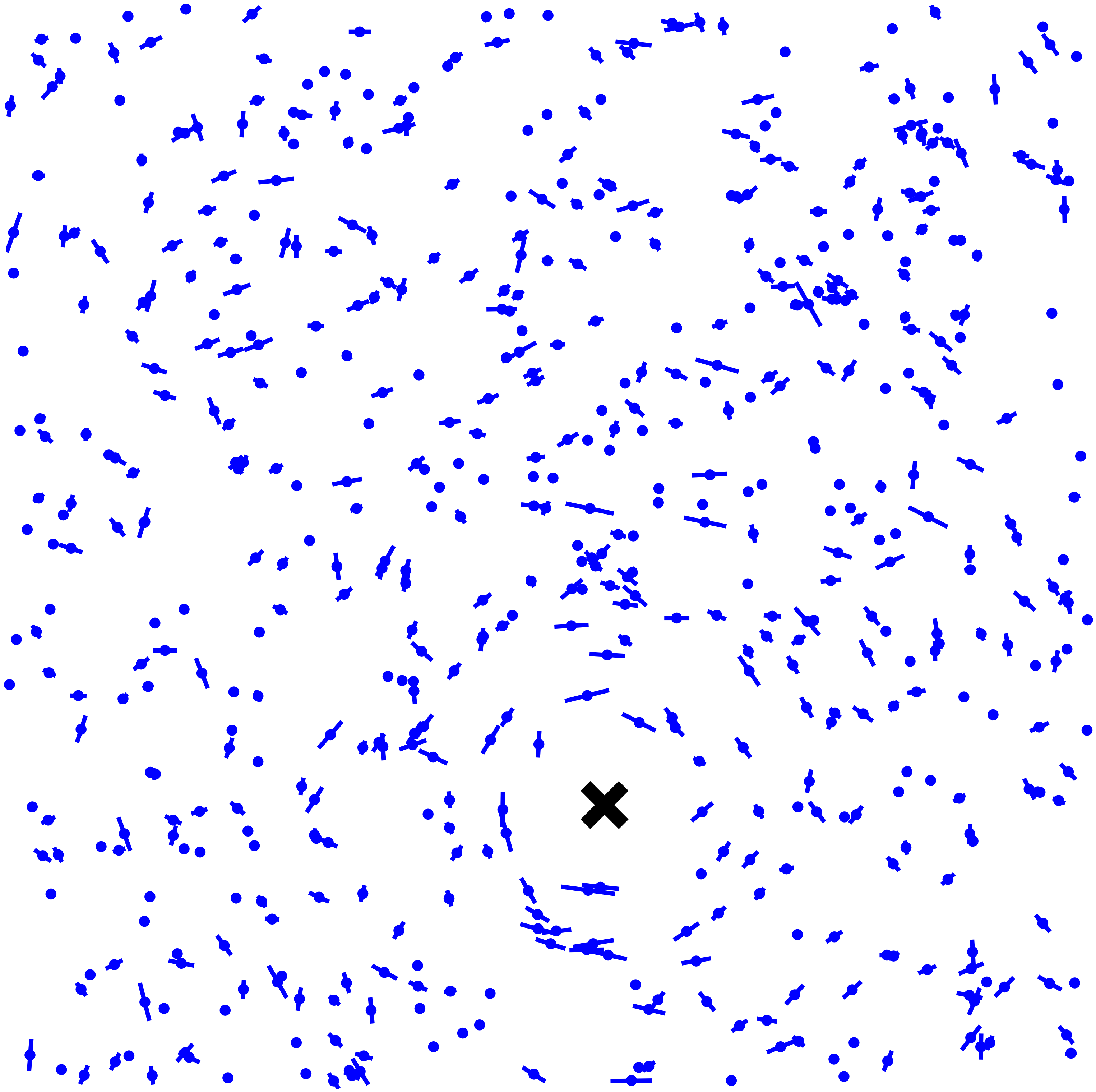
C. However unfortunately galaxies are NOT circular and infact they are inherently elliptical. This property is random, however since the Universe has no preferred ellipticity this averages out to zero in the case of no other influence.

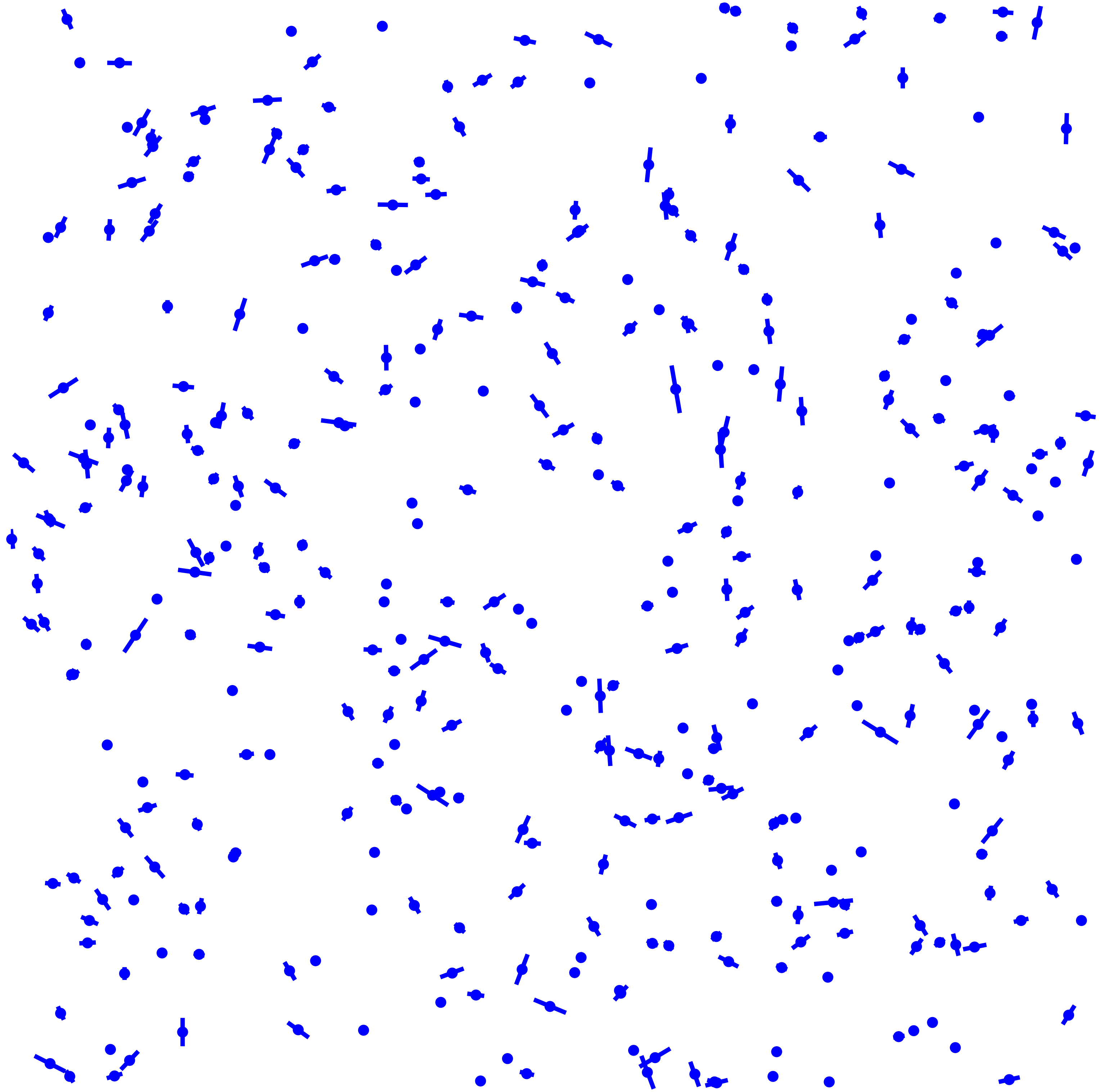


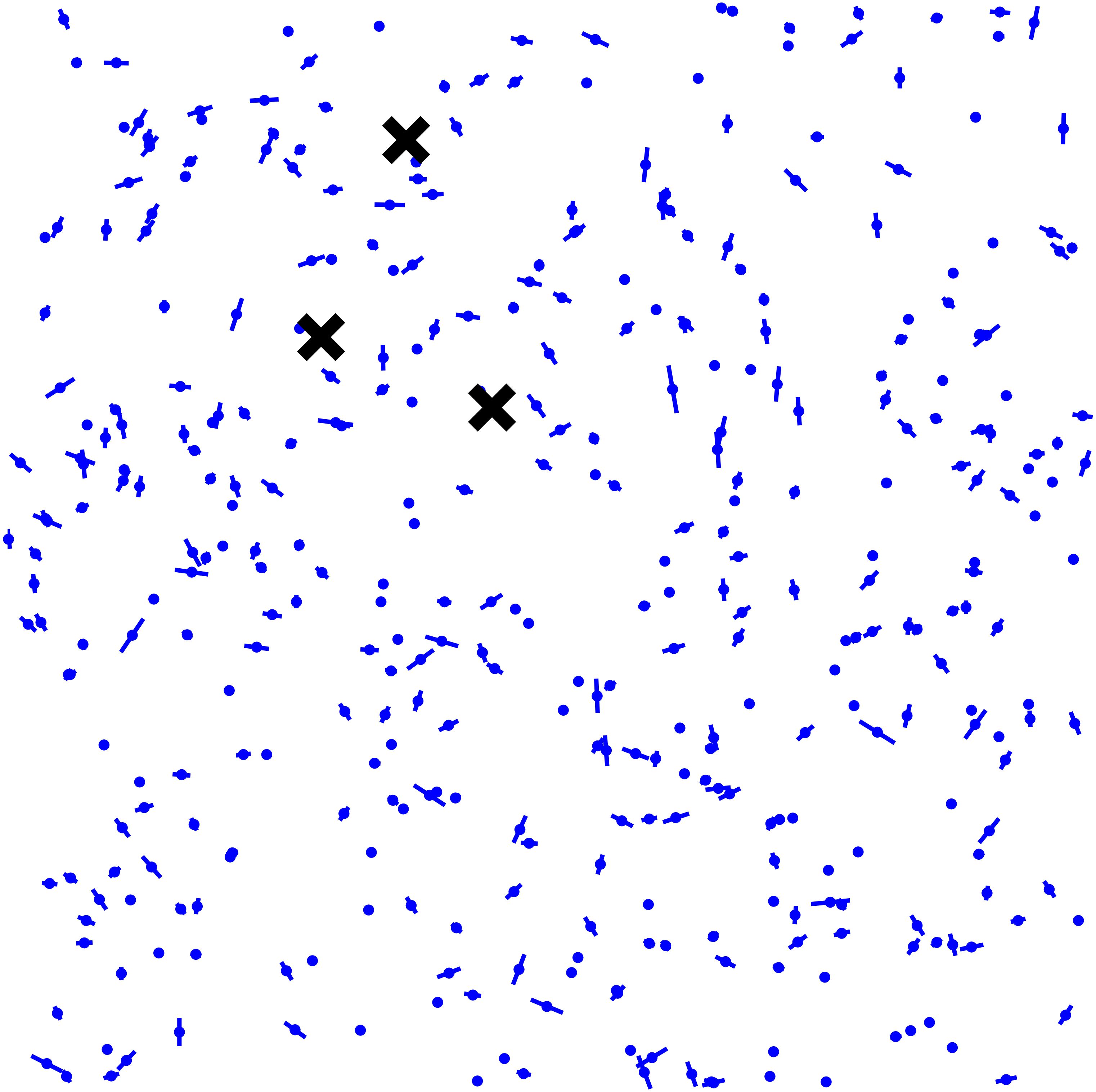
D. Therefore if we placed a Dark Matter halo into a field of randomly elliptical galaxies we would get a field that does not average out to zero. If we can use the fact that Dark Matter makes the pattern seen in B, we should be able to detect the position of the central halo.

http://www.kaggle.com/c/DarkWorlds





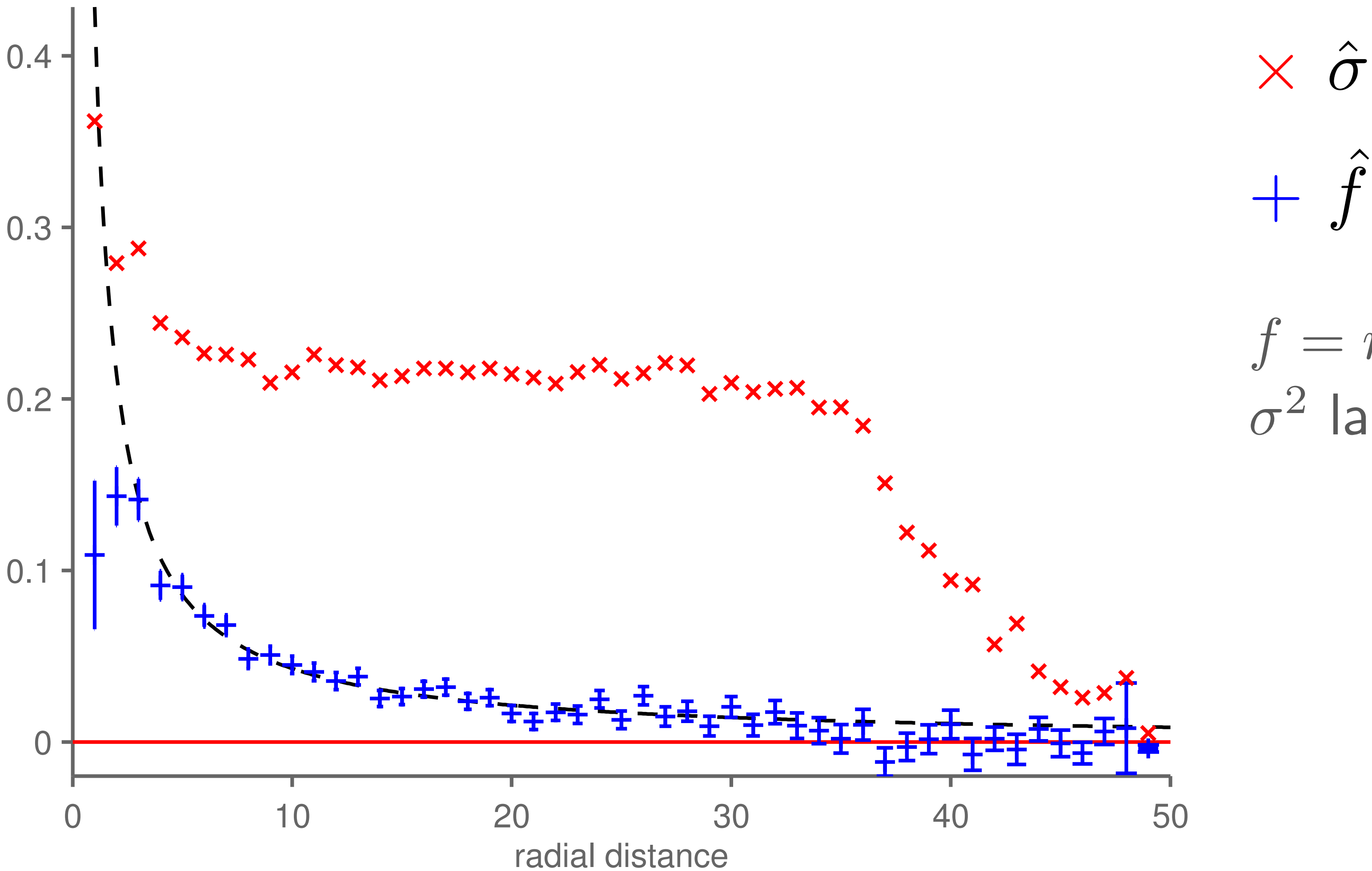




Probabilistic model

$$e_1^{(n)} \sim \mathcal{N}\big(f^{(n)} \cos 2\theta^{(n)},\ \sigma^2\big) \qquad f^{(n)} = m/r^{(n)}$$

$$e_2^{(n)} \sim \mathcal{N}\big(f^{(n)} \sin 2\theta^{(n)},\ \sigma^2\big)$$

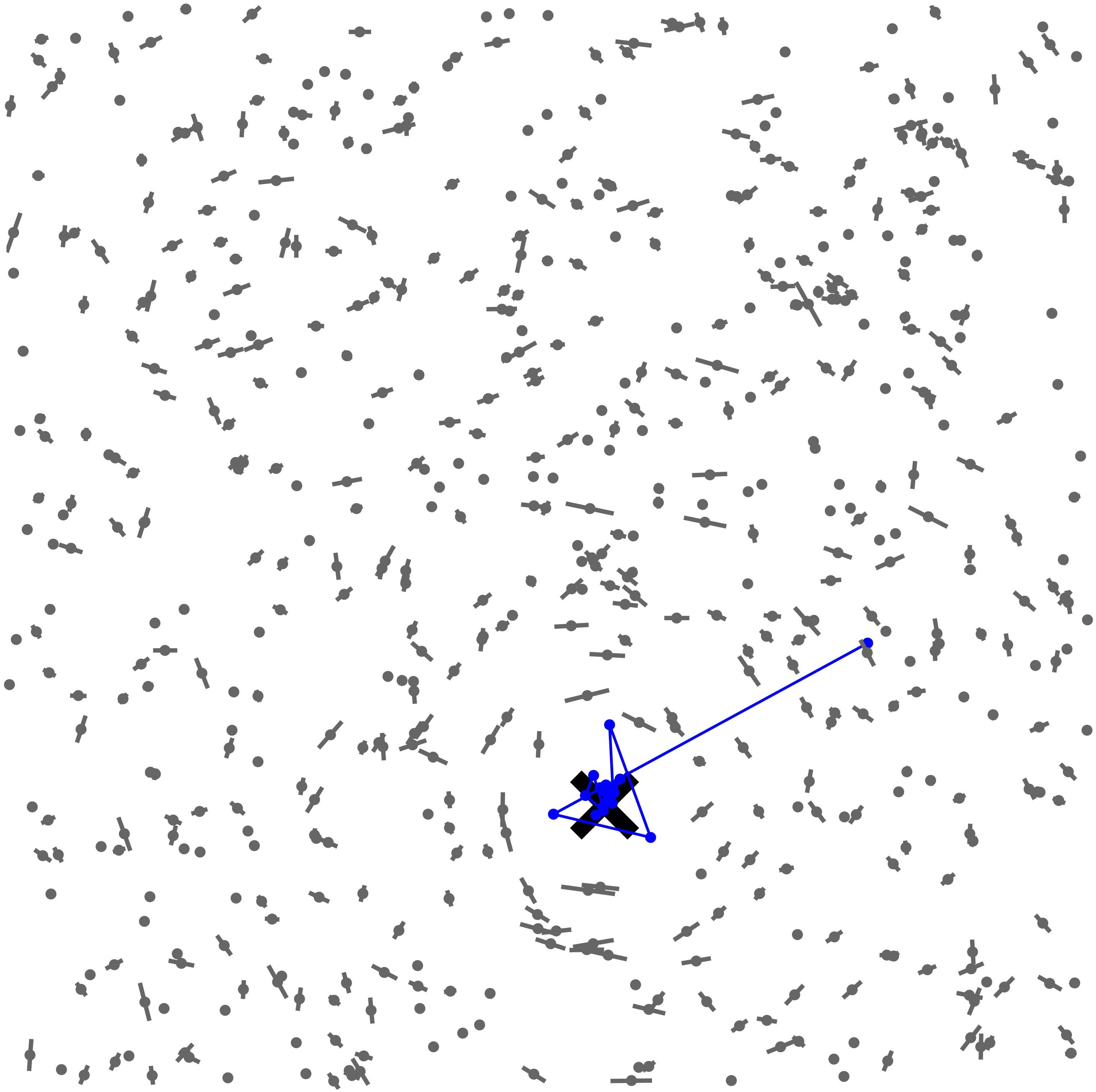


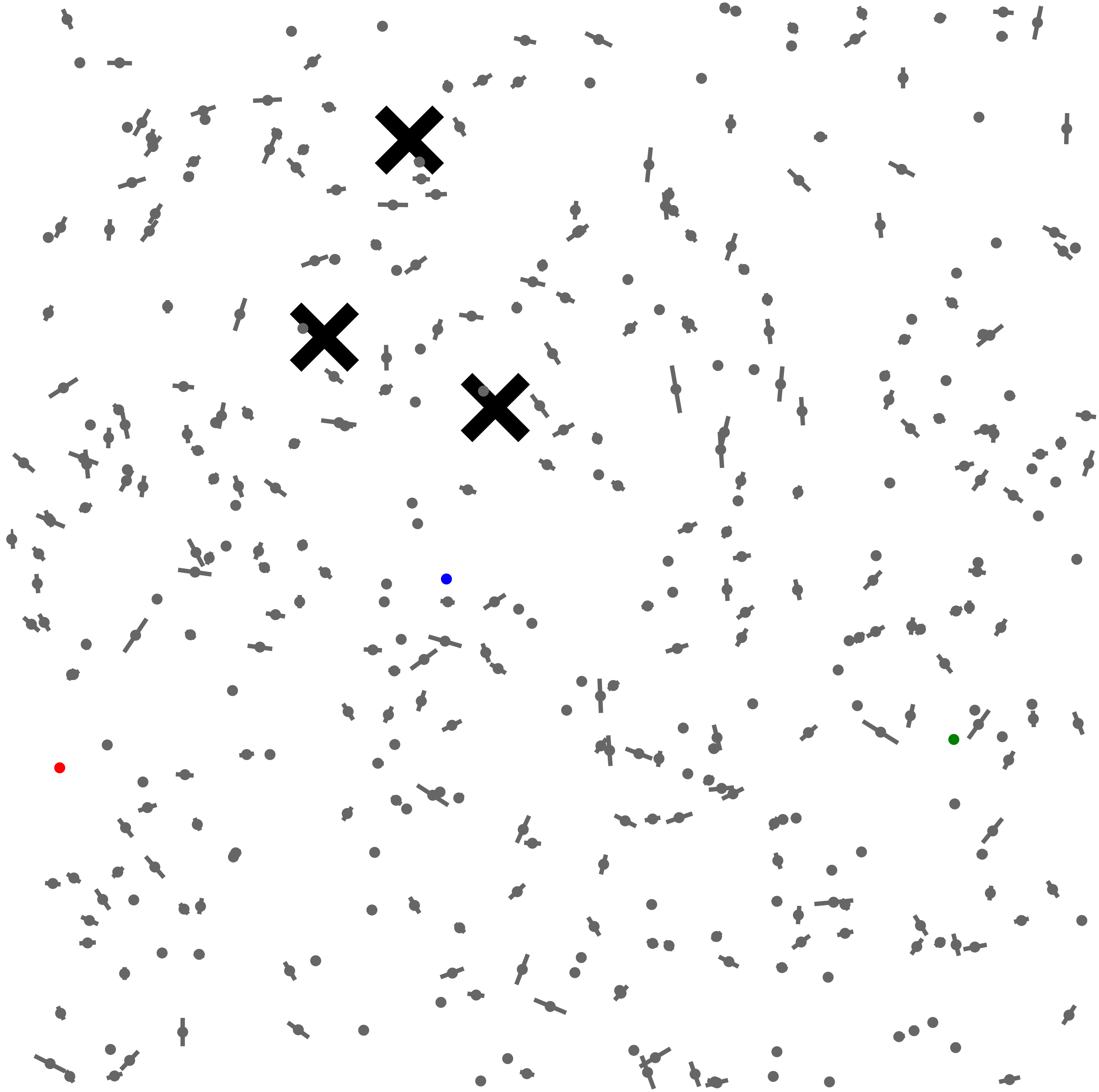
$f = m/\min(r_0, r)$

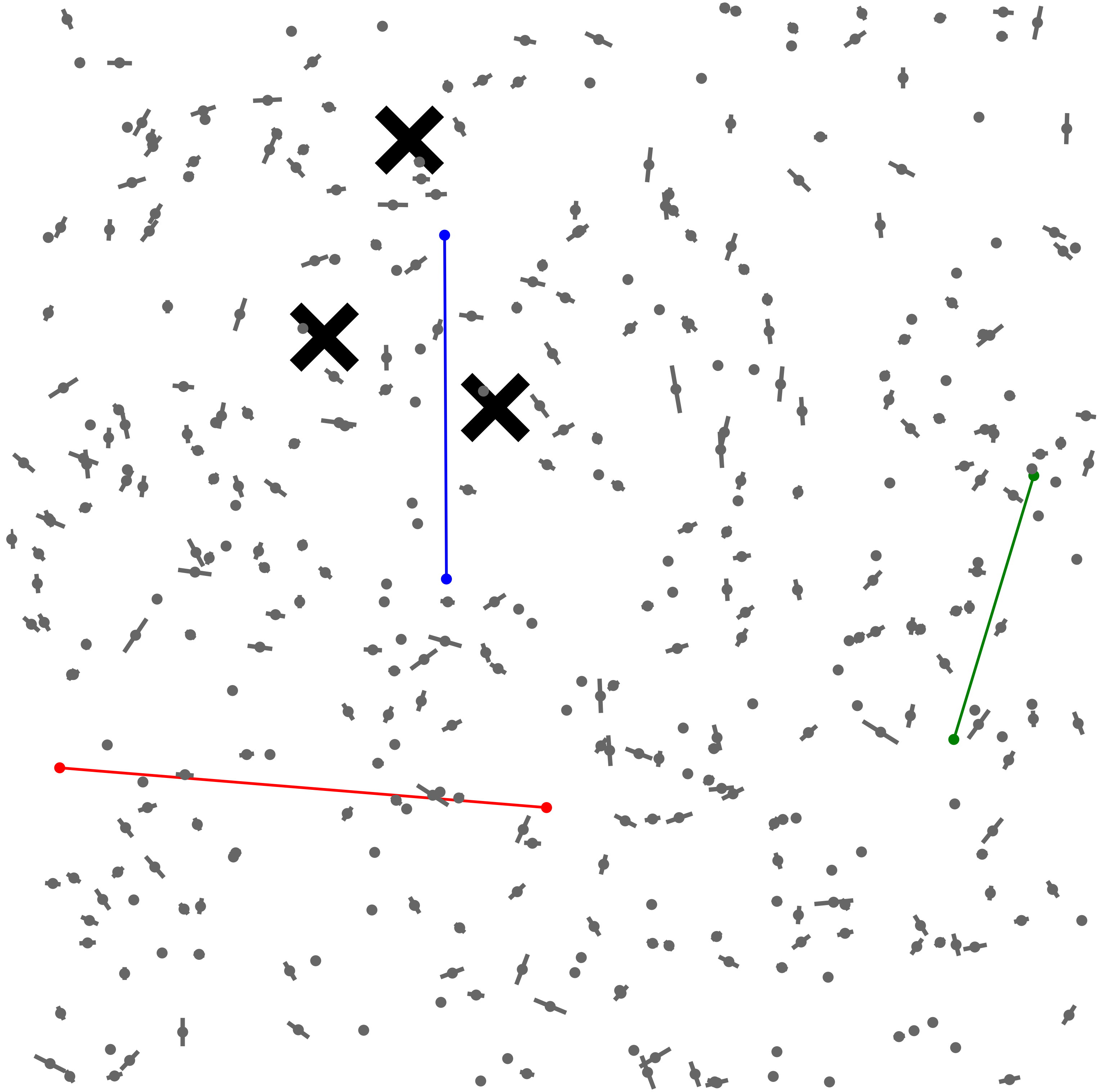
σ^2 larger for small r

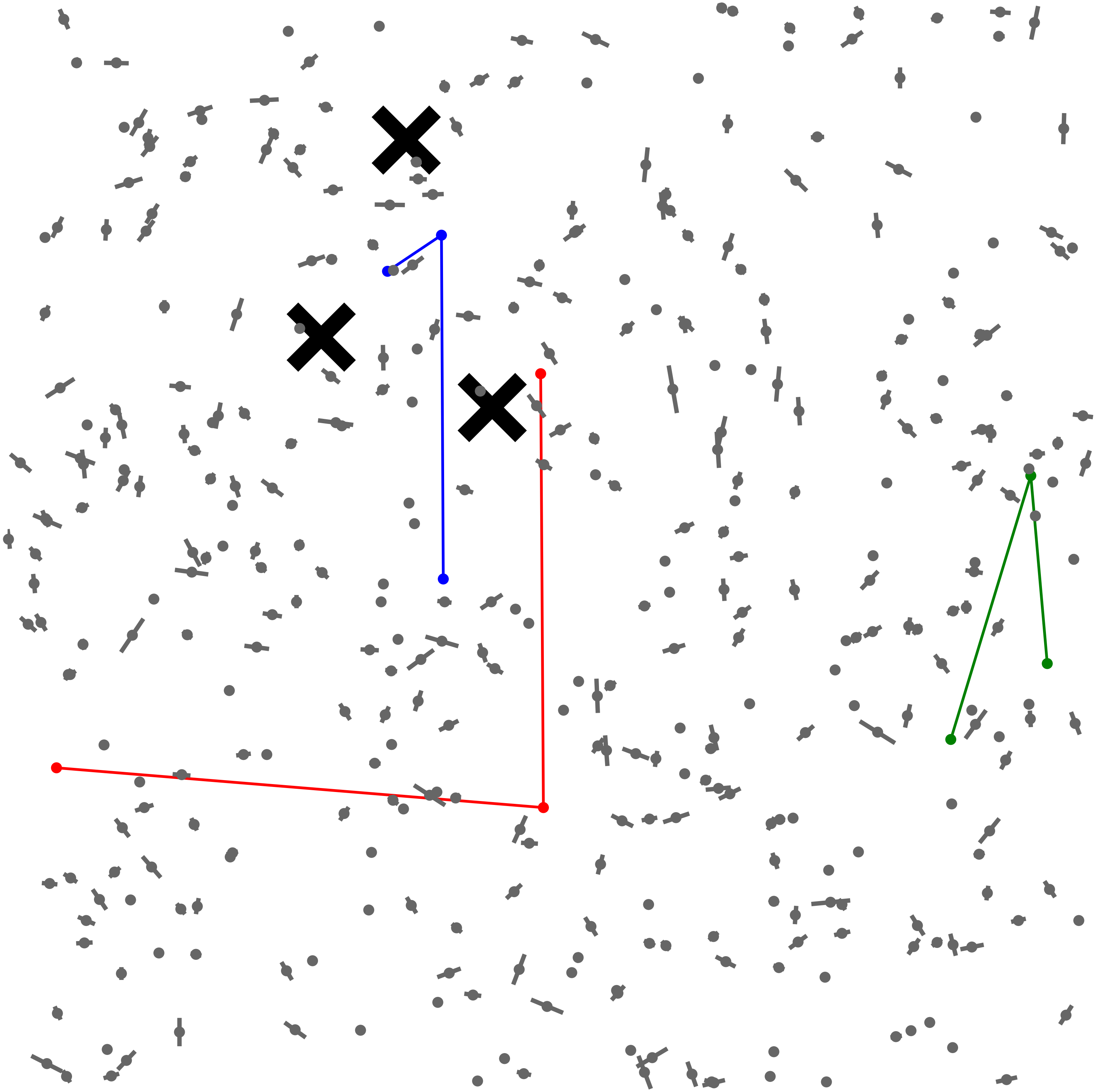
Inference

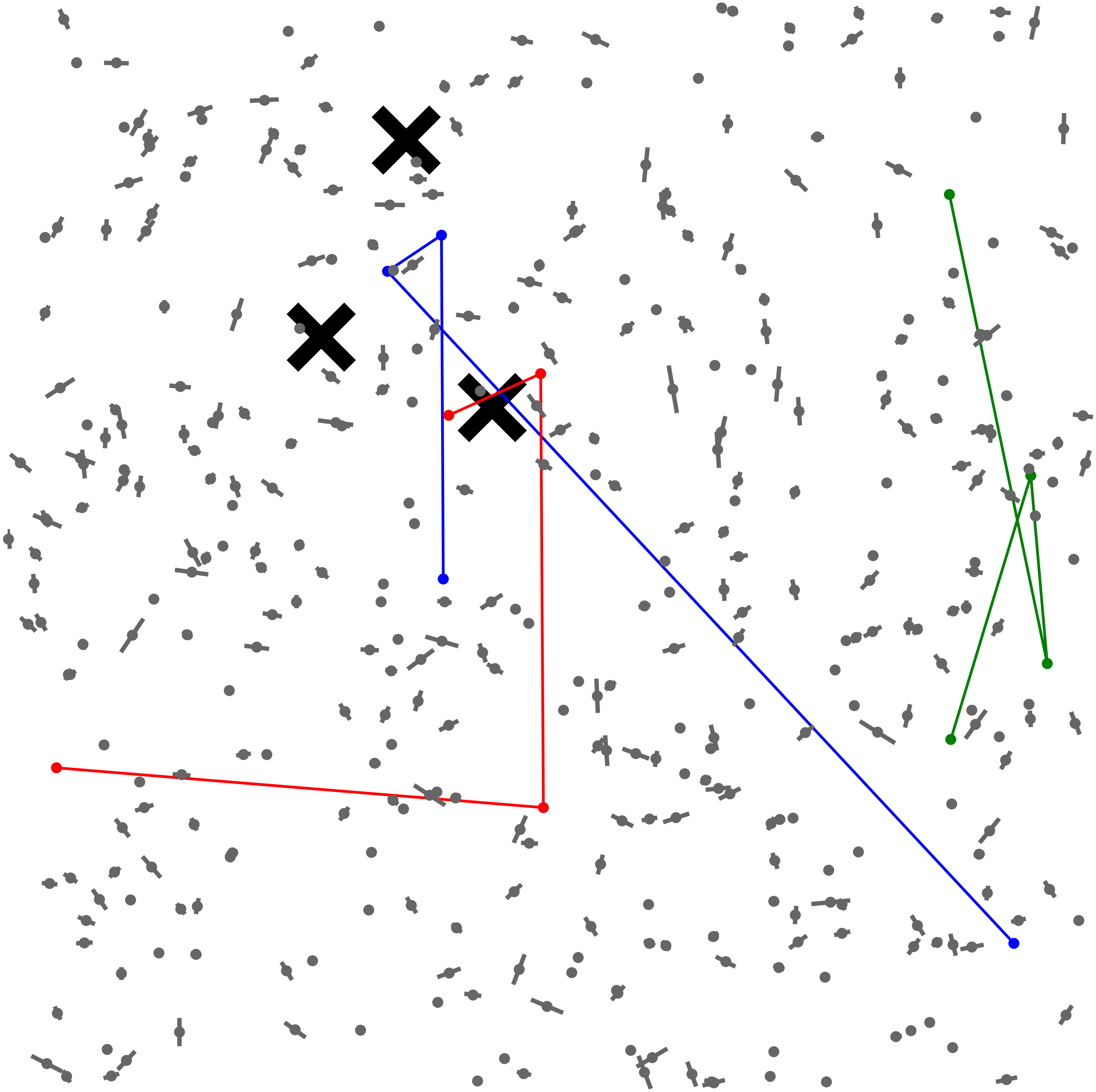
Markov chain Monte Carlo (MCMC)

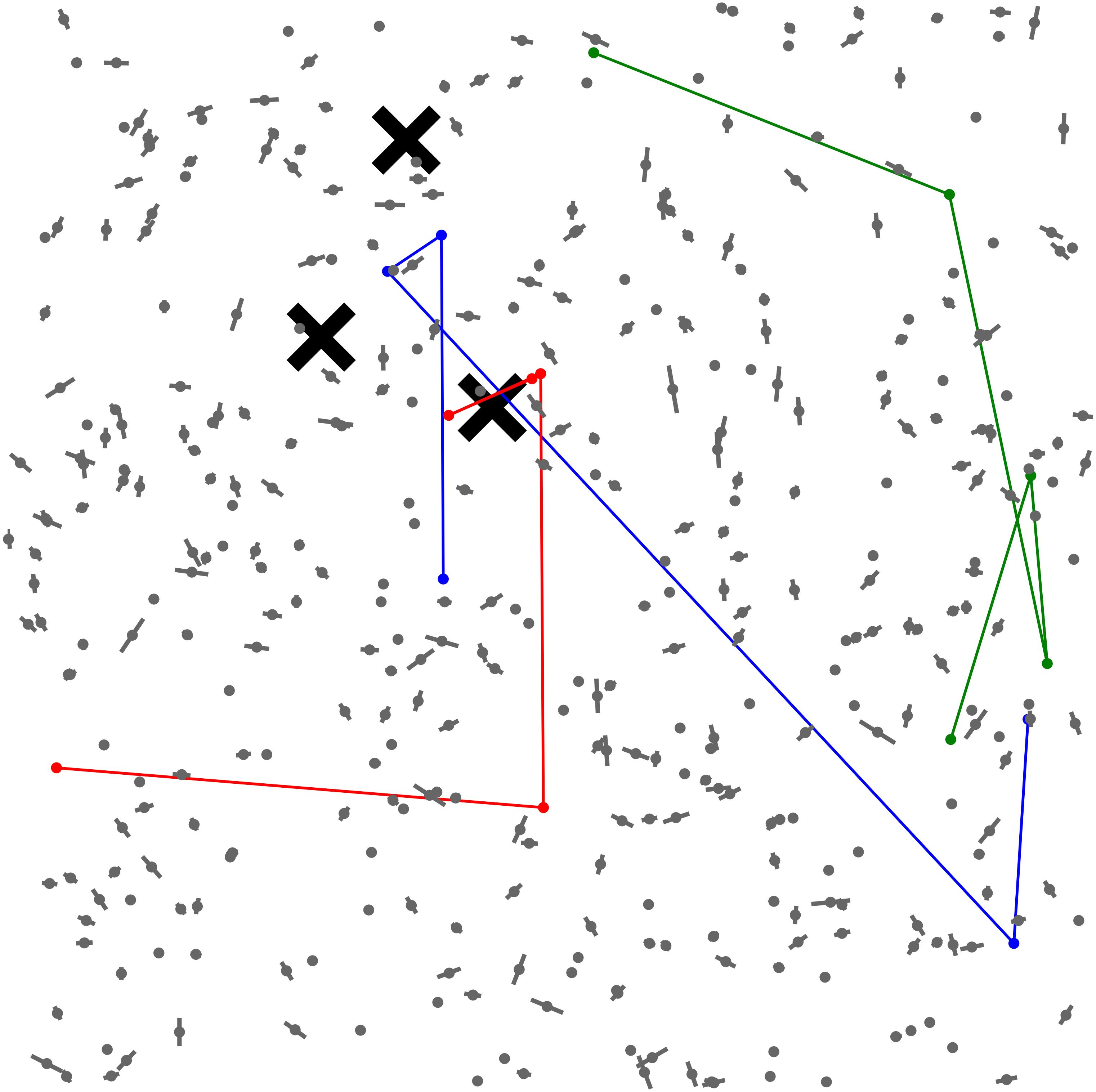


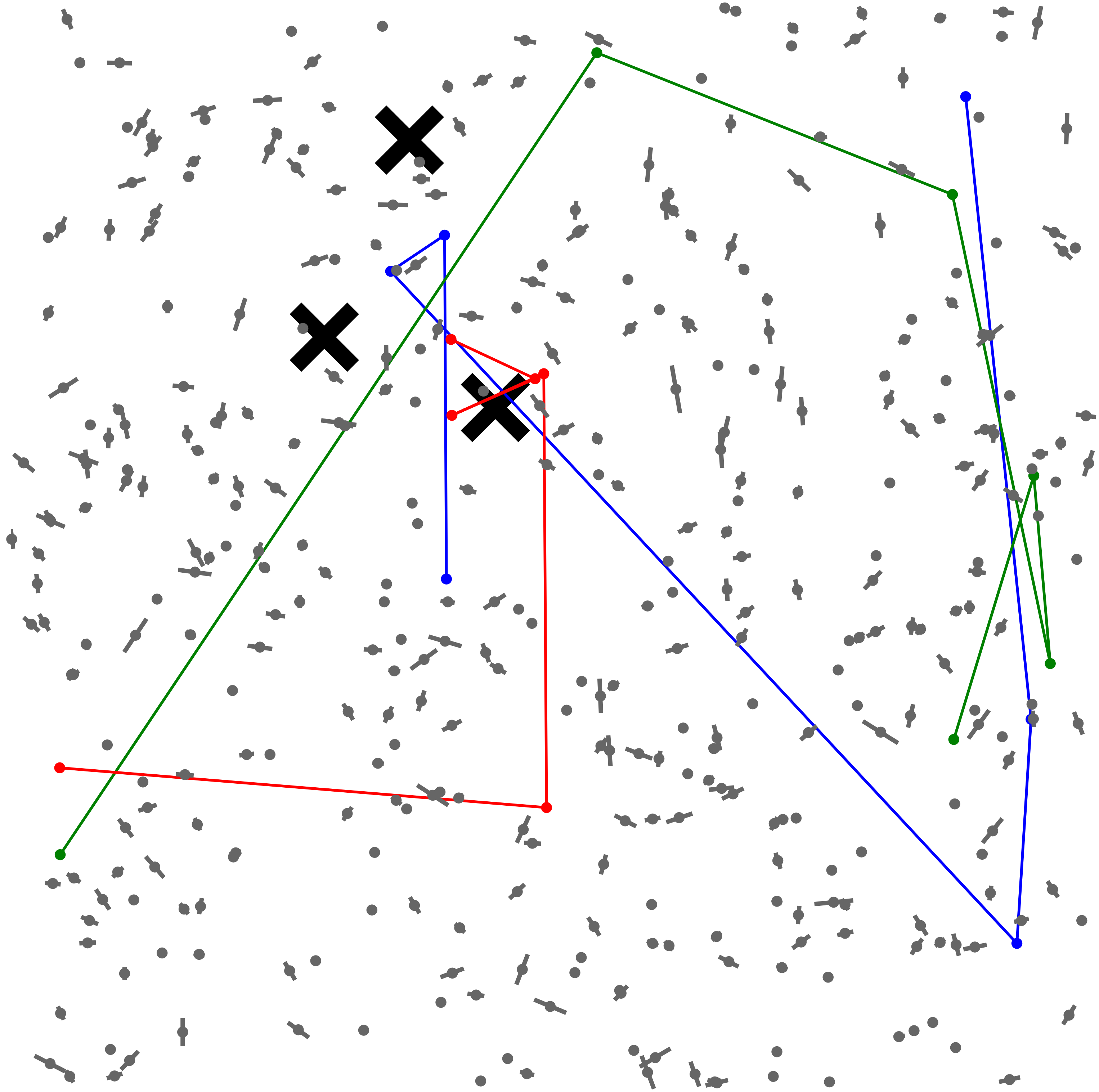


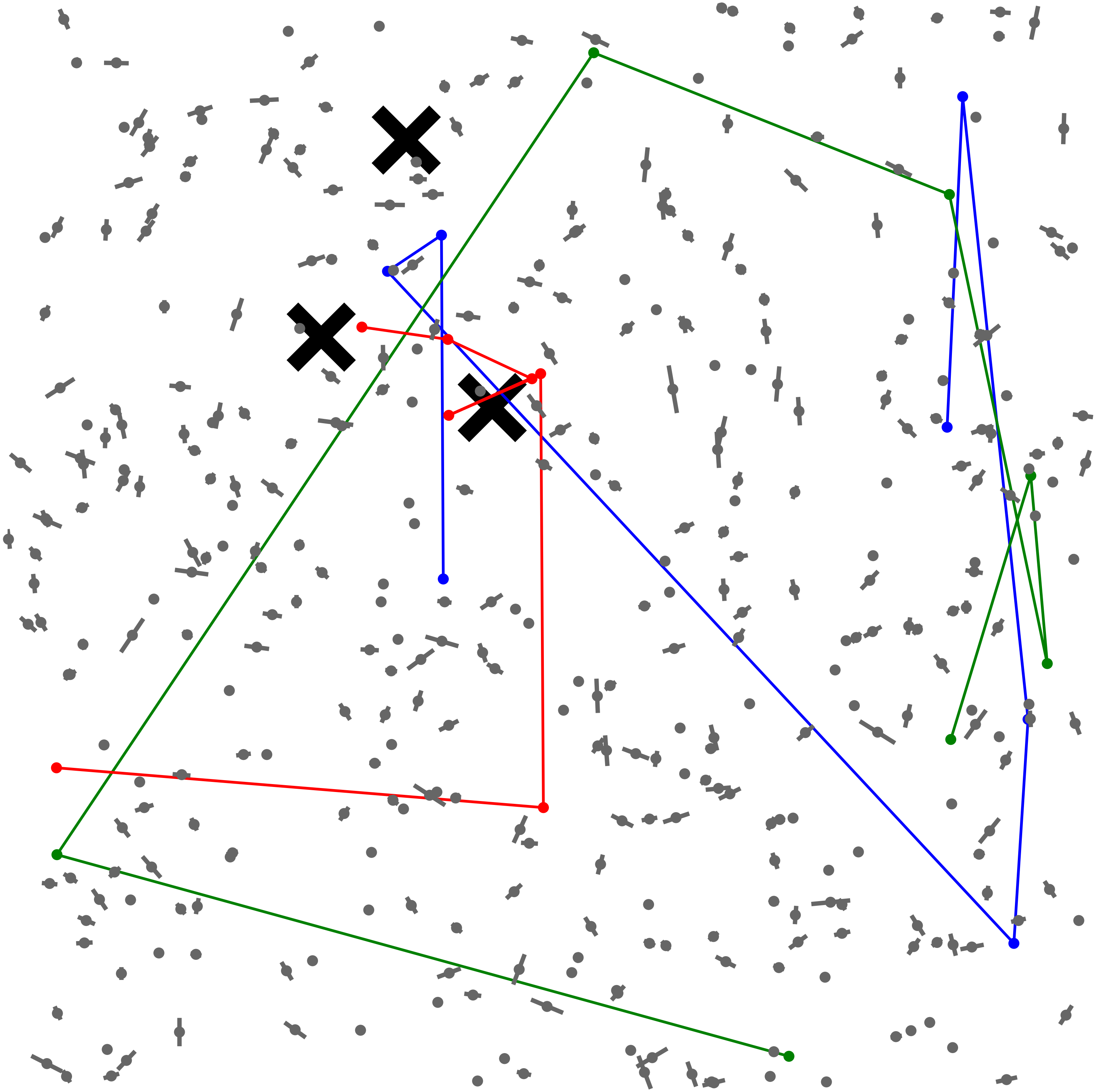


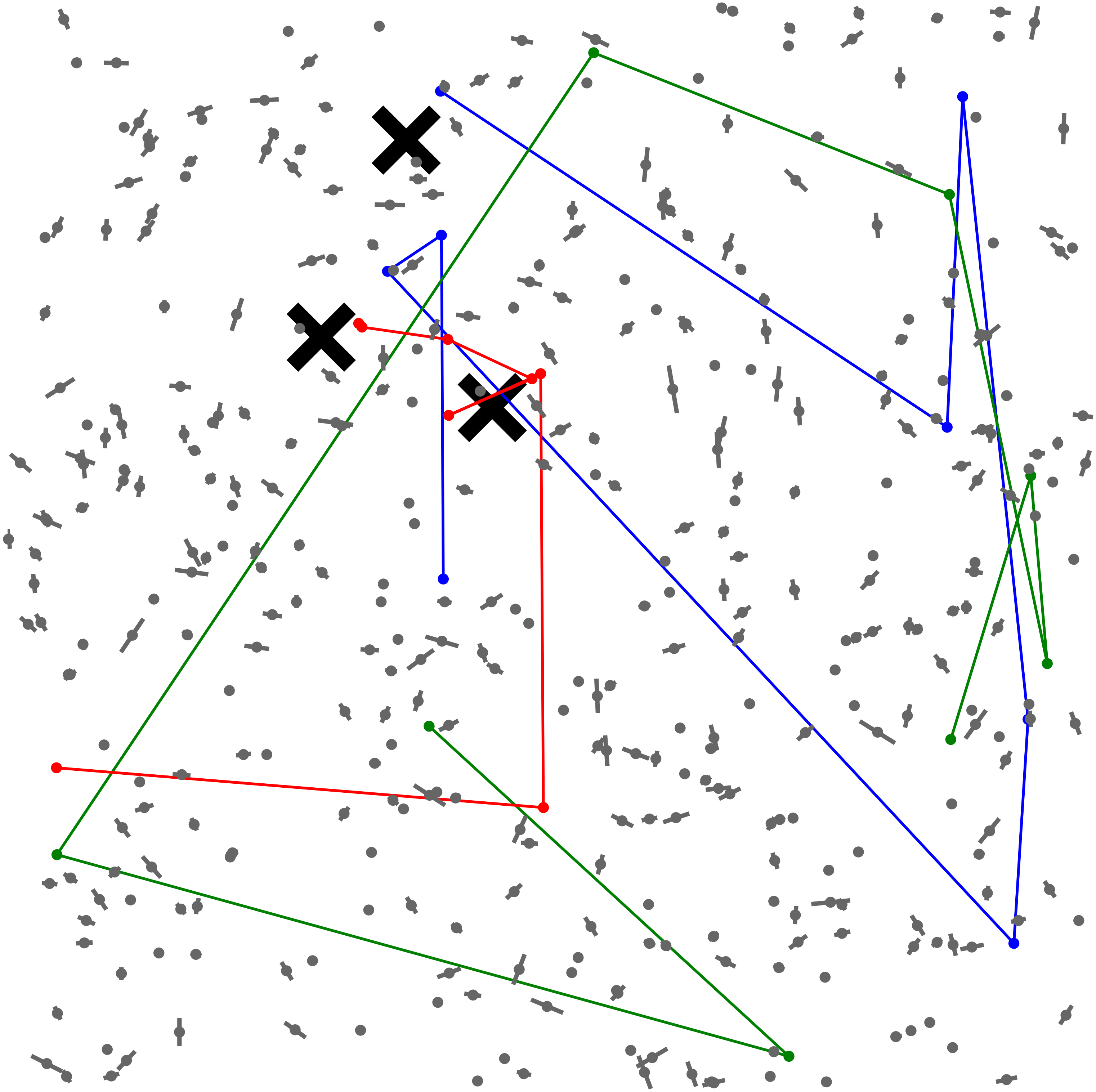


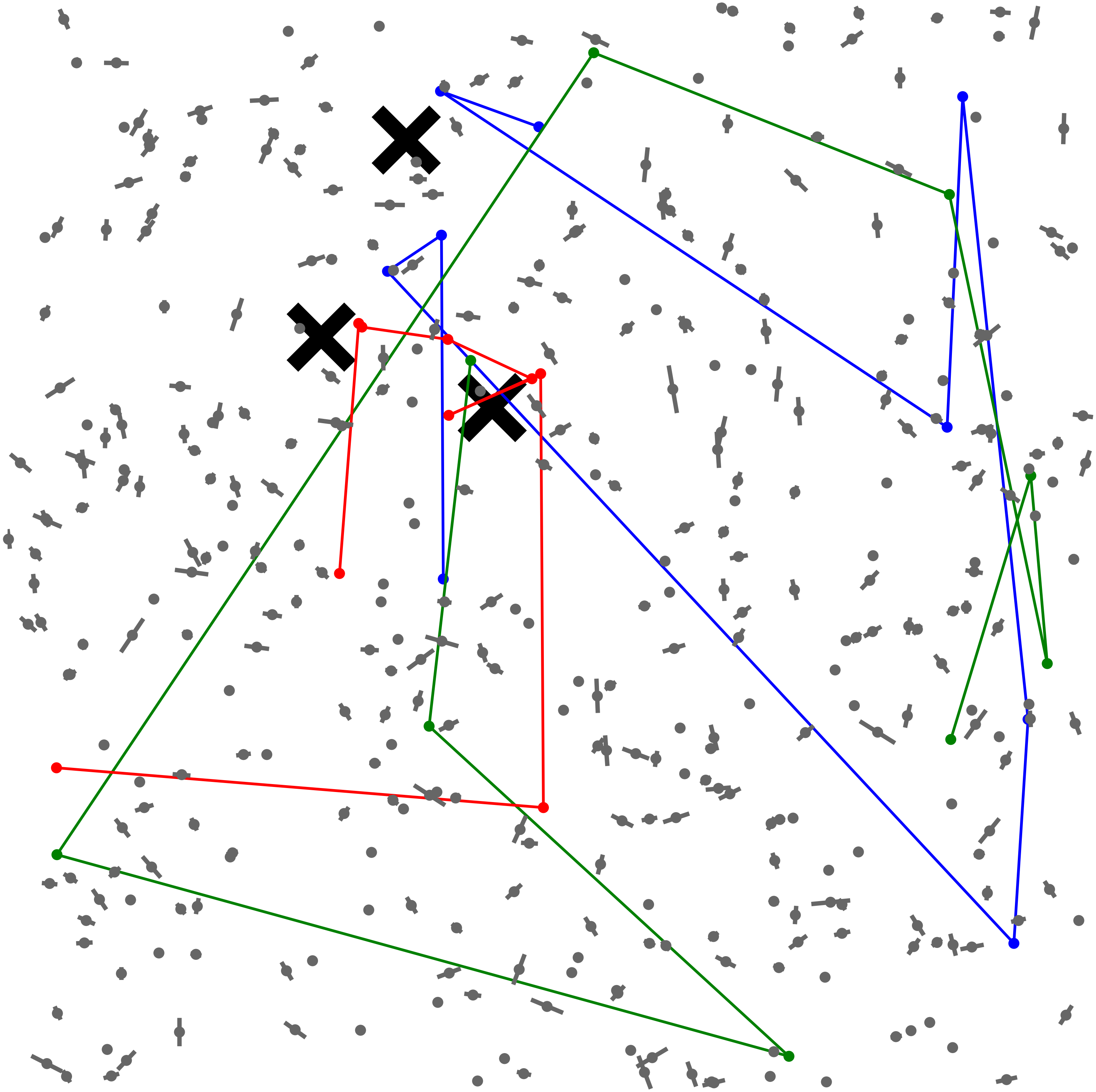


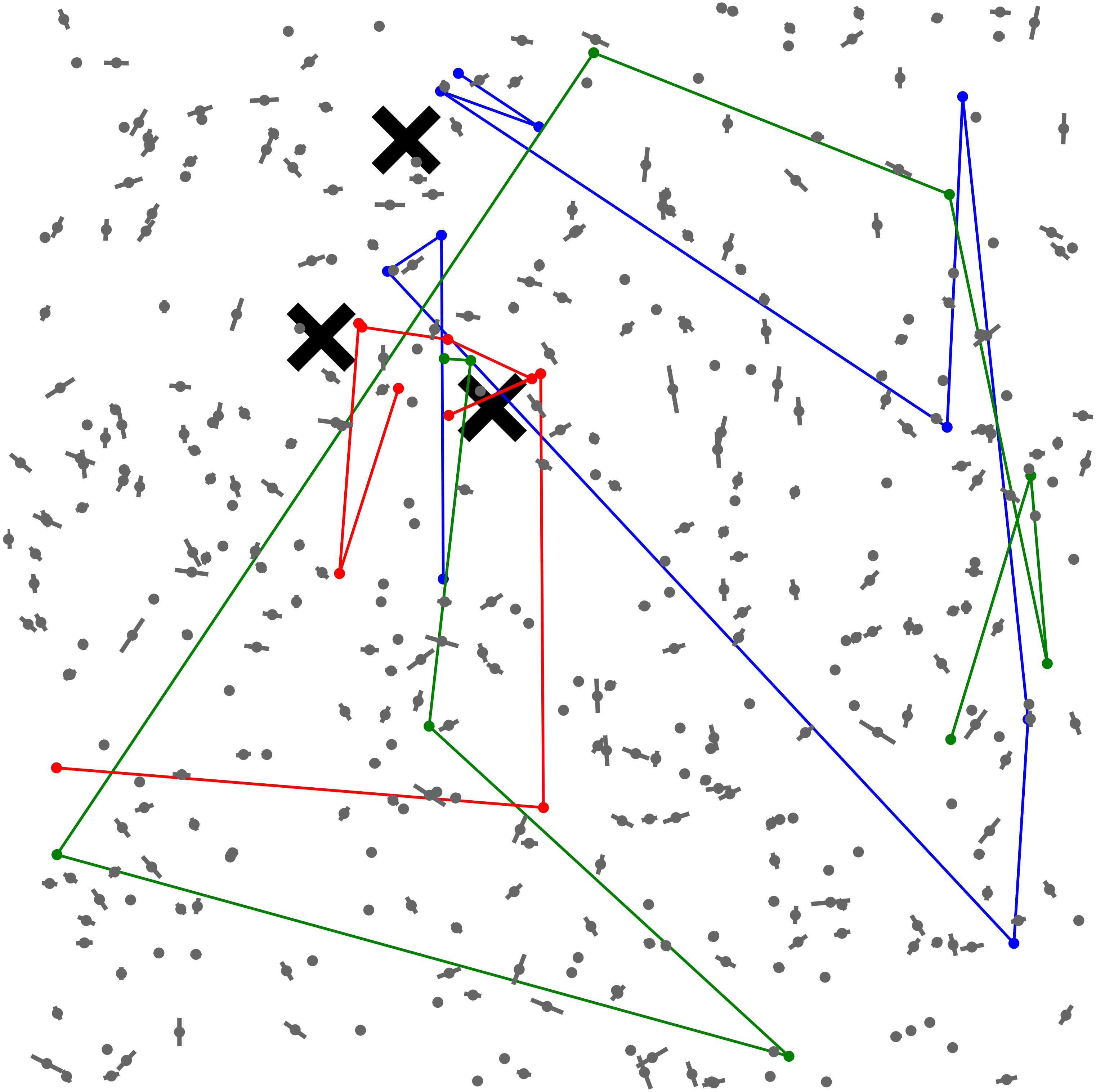


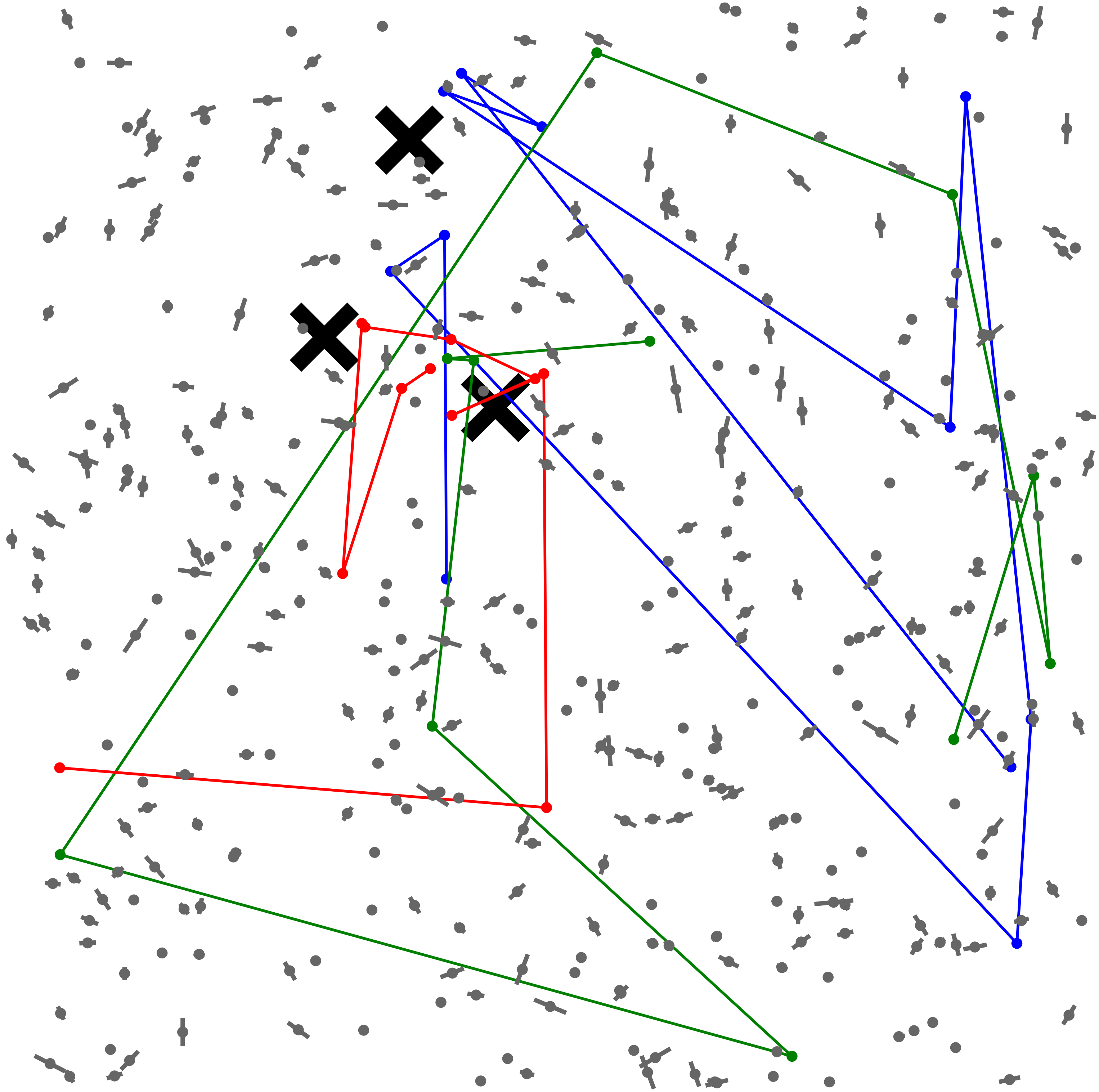


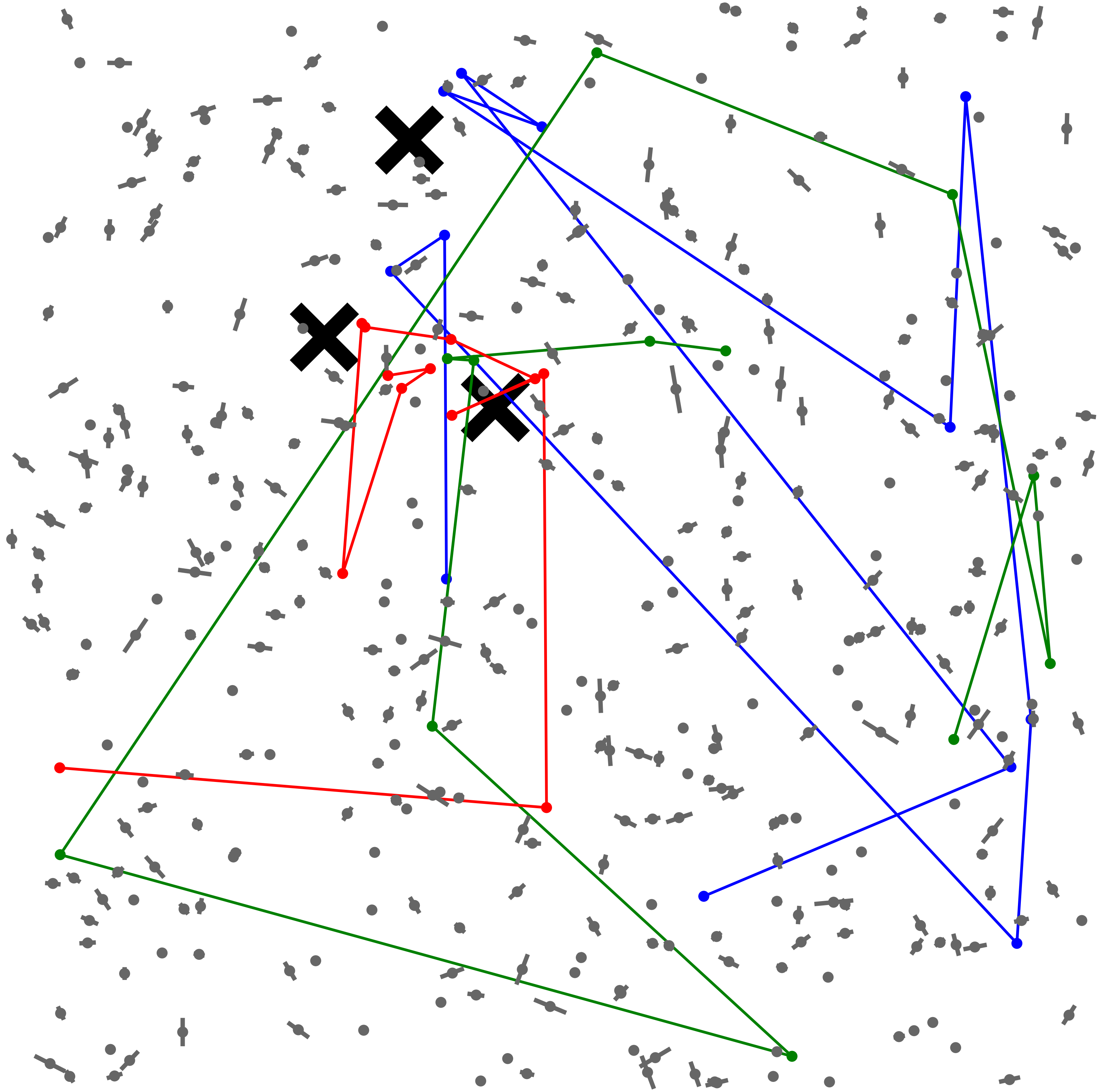


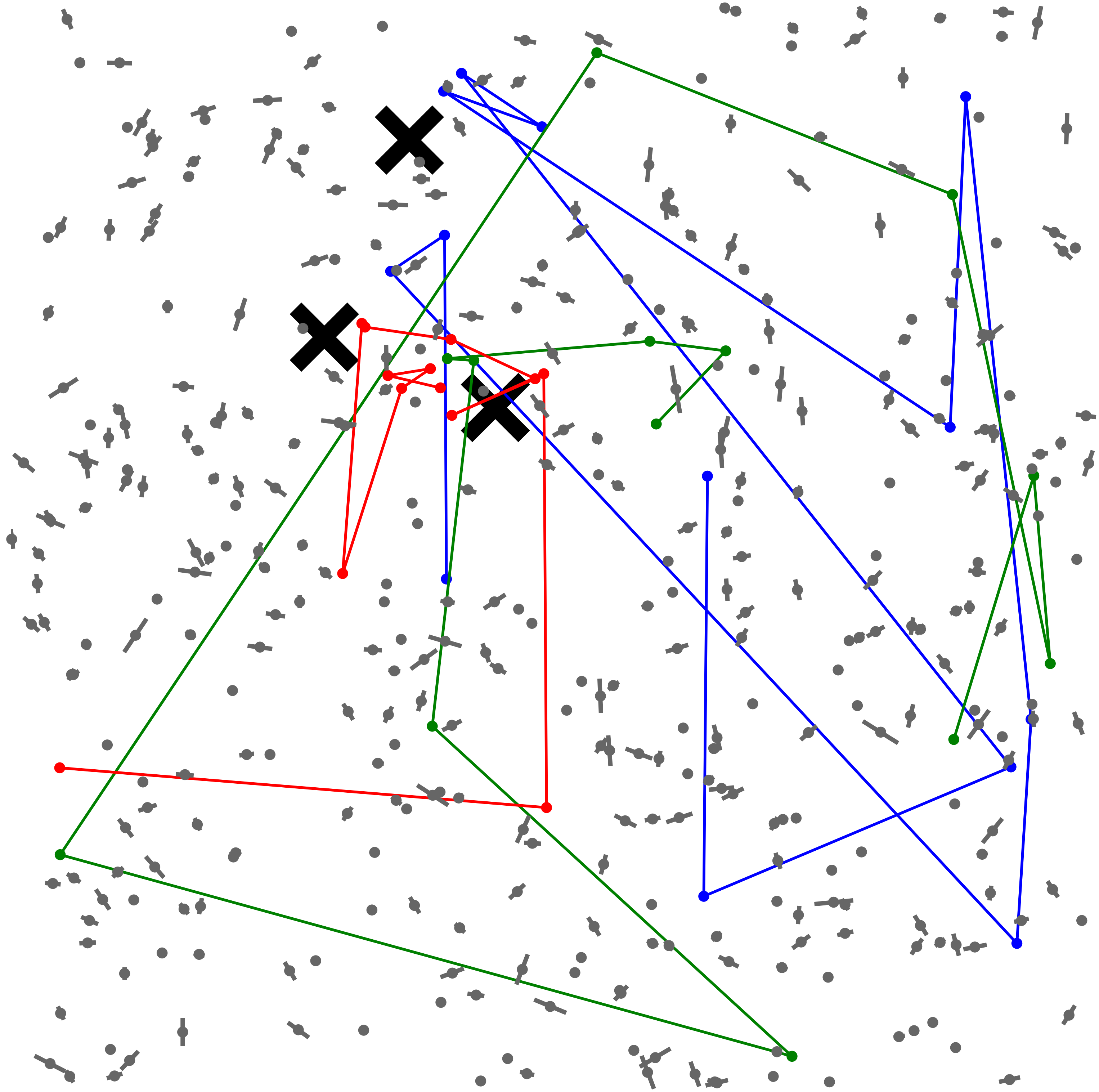


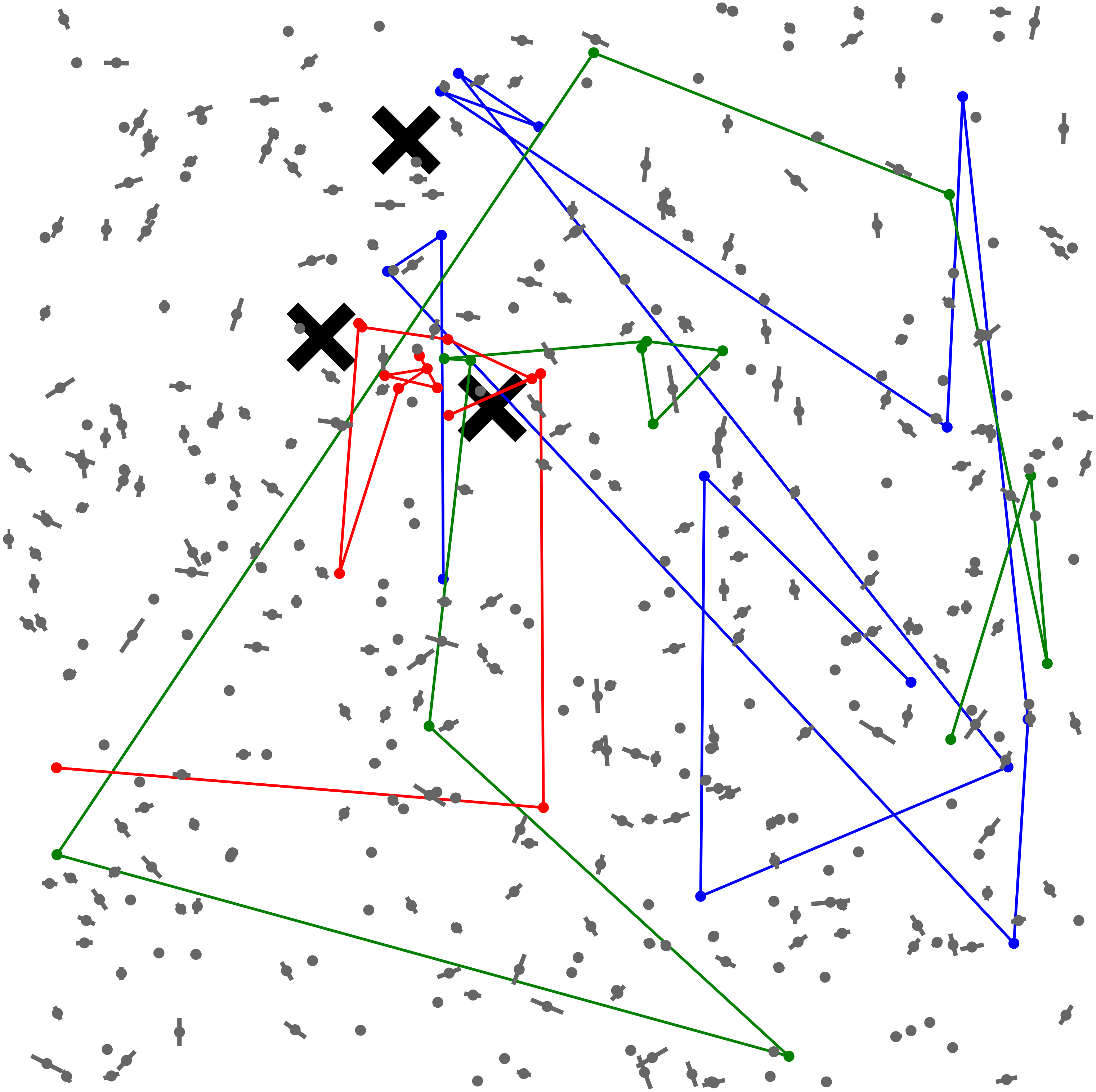


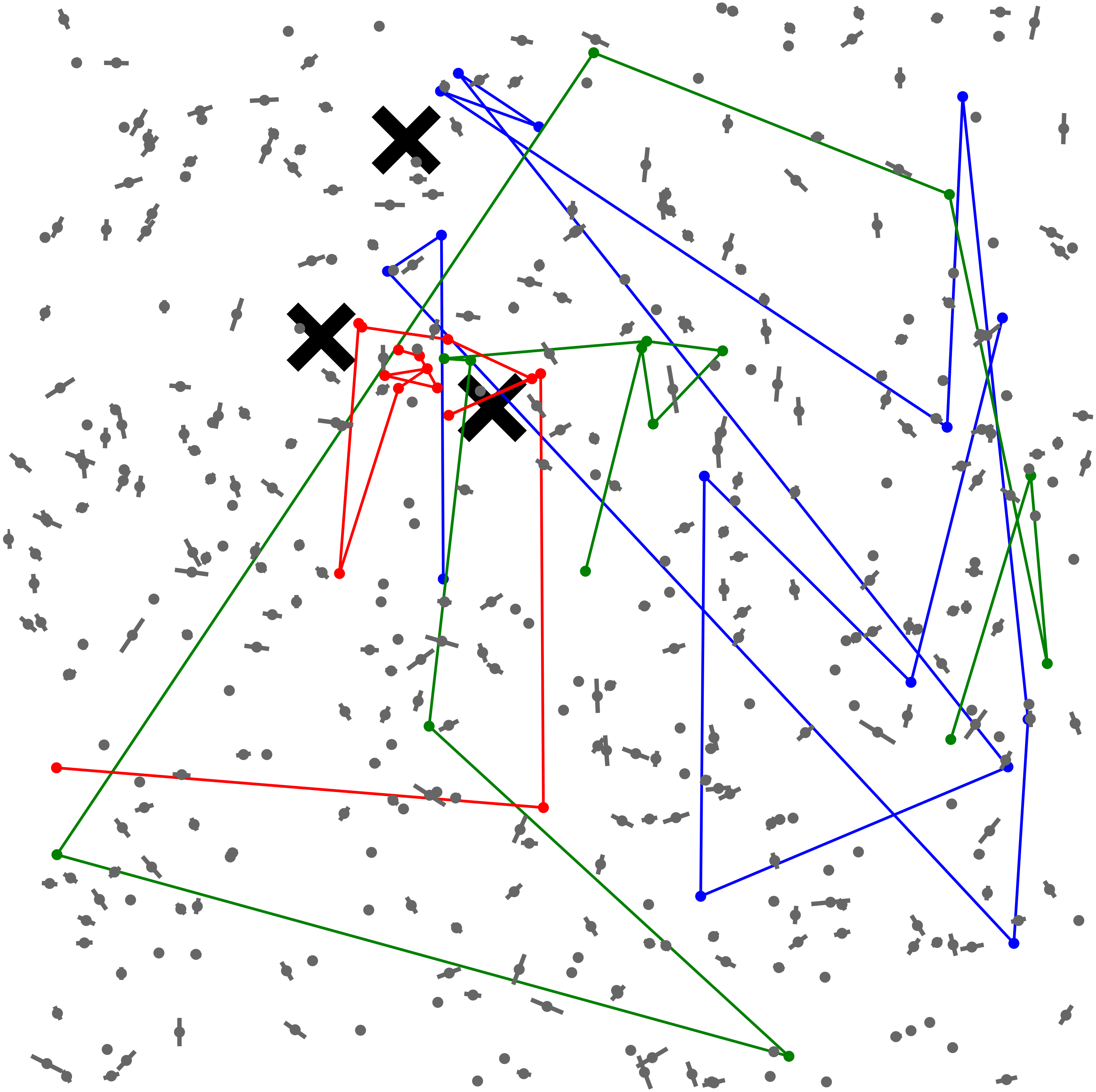


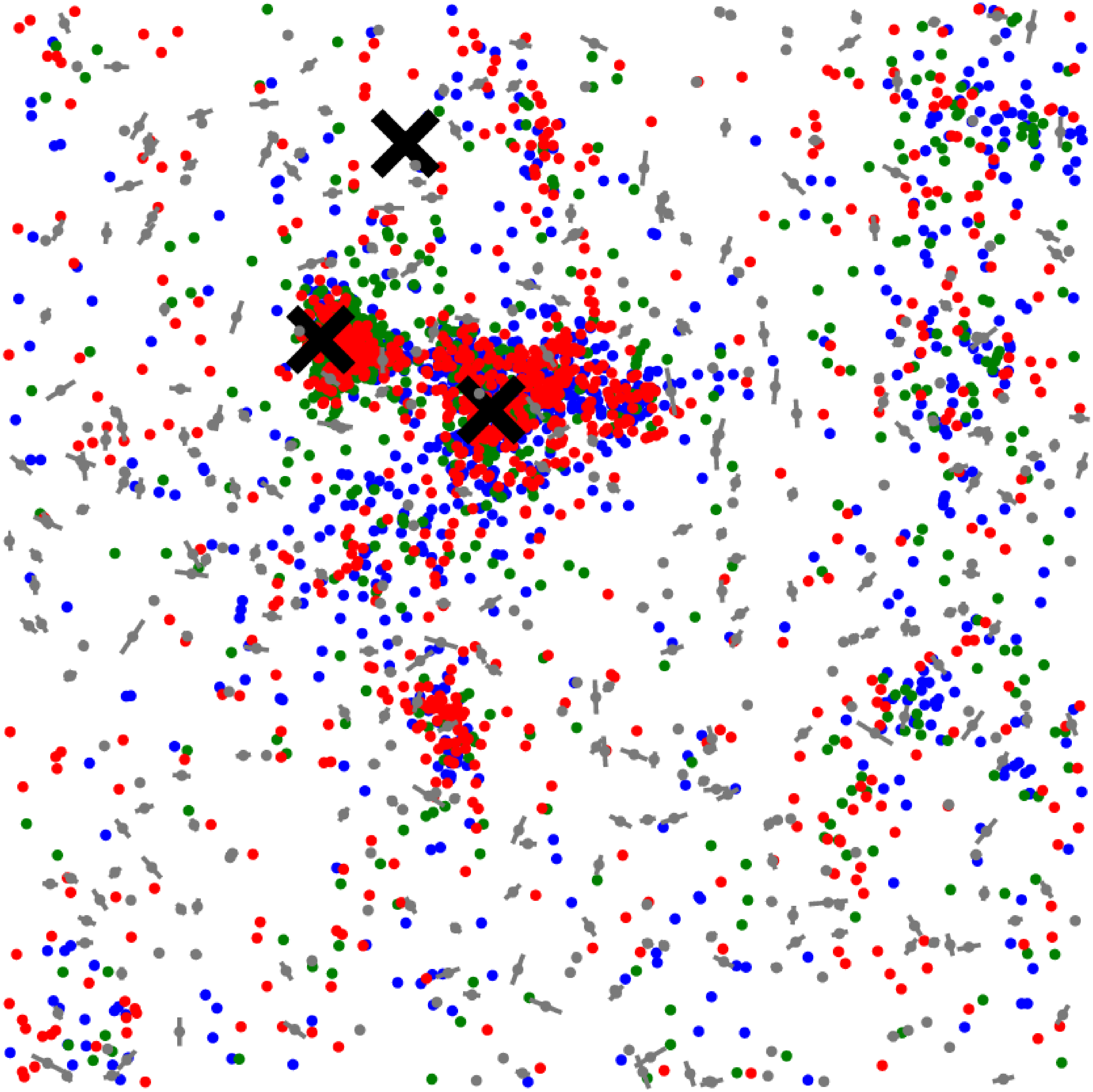


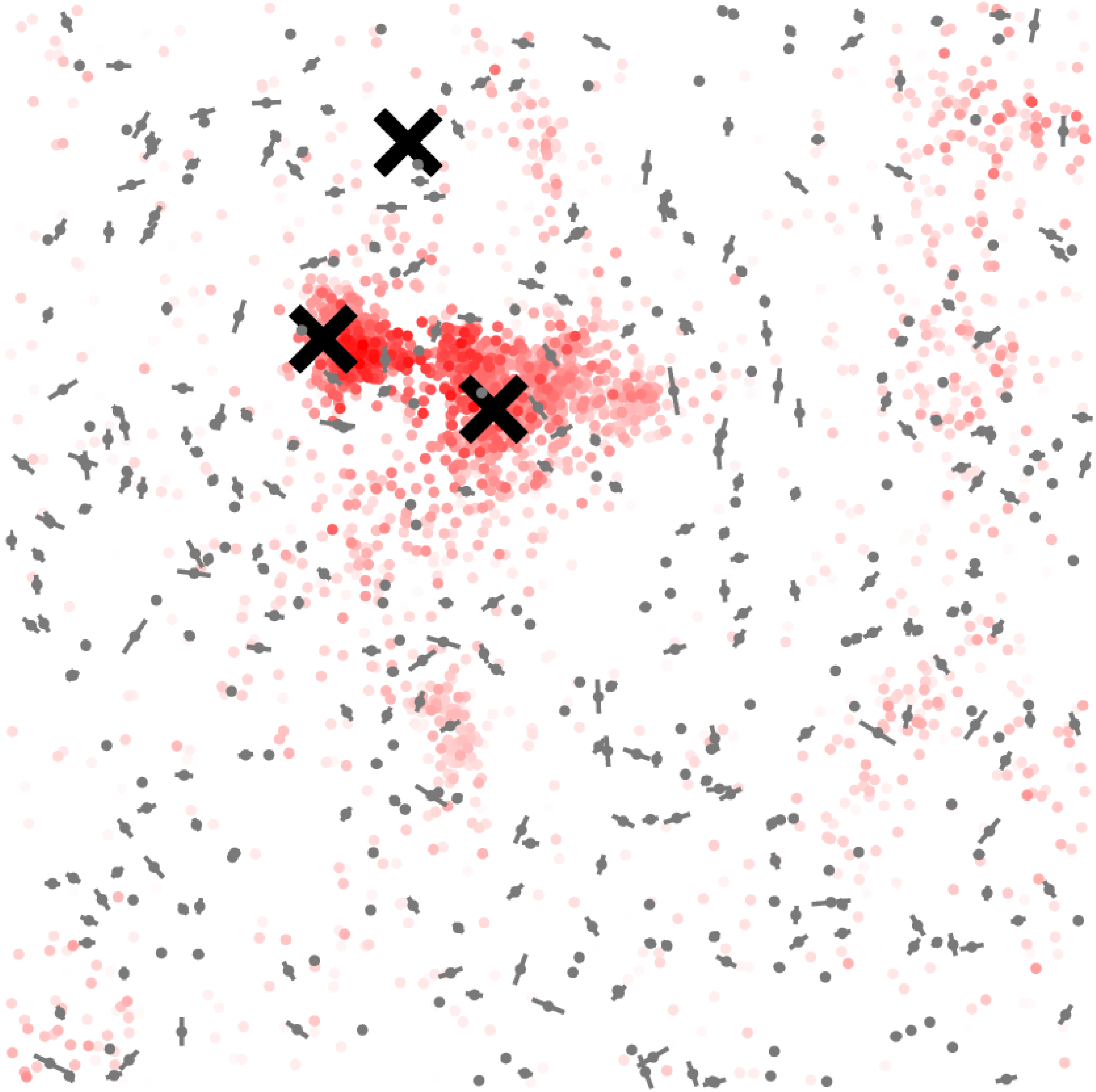












Reporting results?

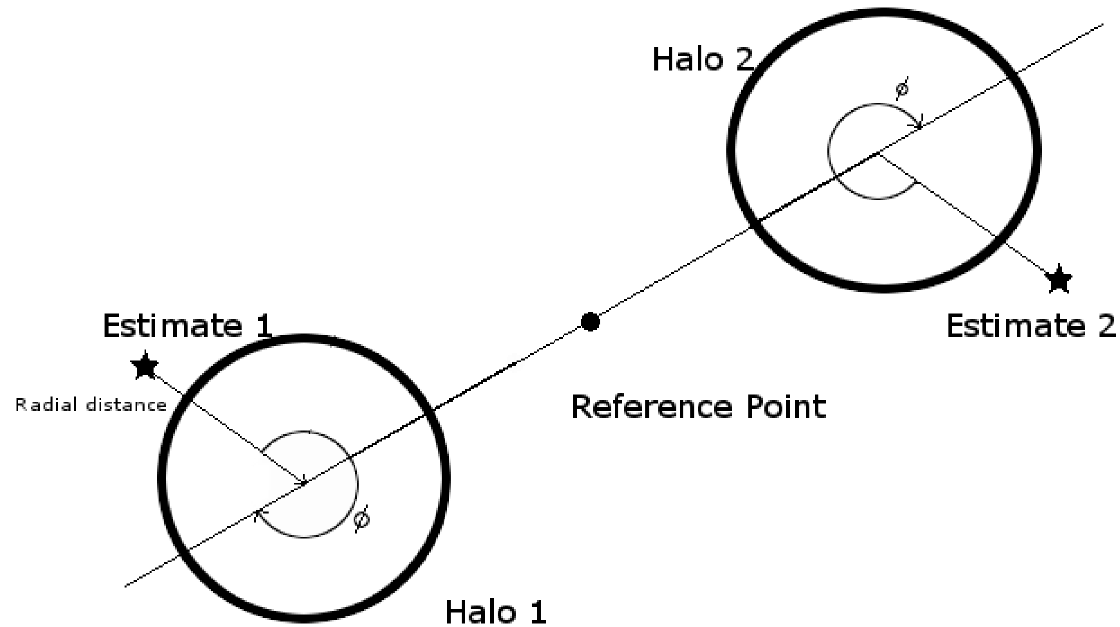
— Average/mean sample?

— Most probable sample?

— Cluster?

Evaluation

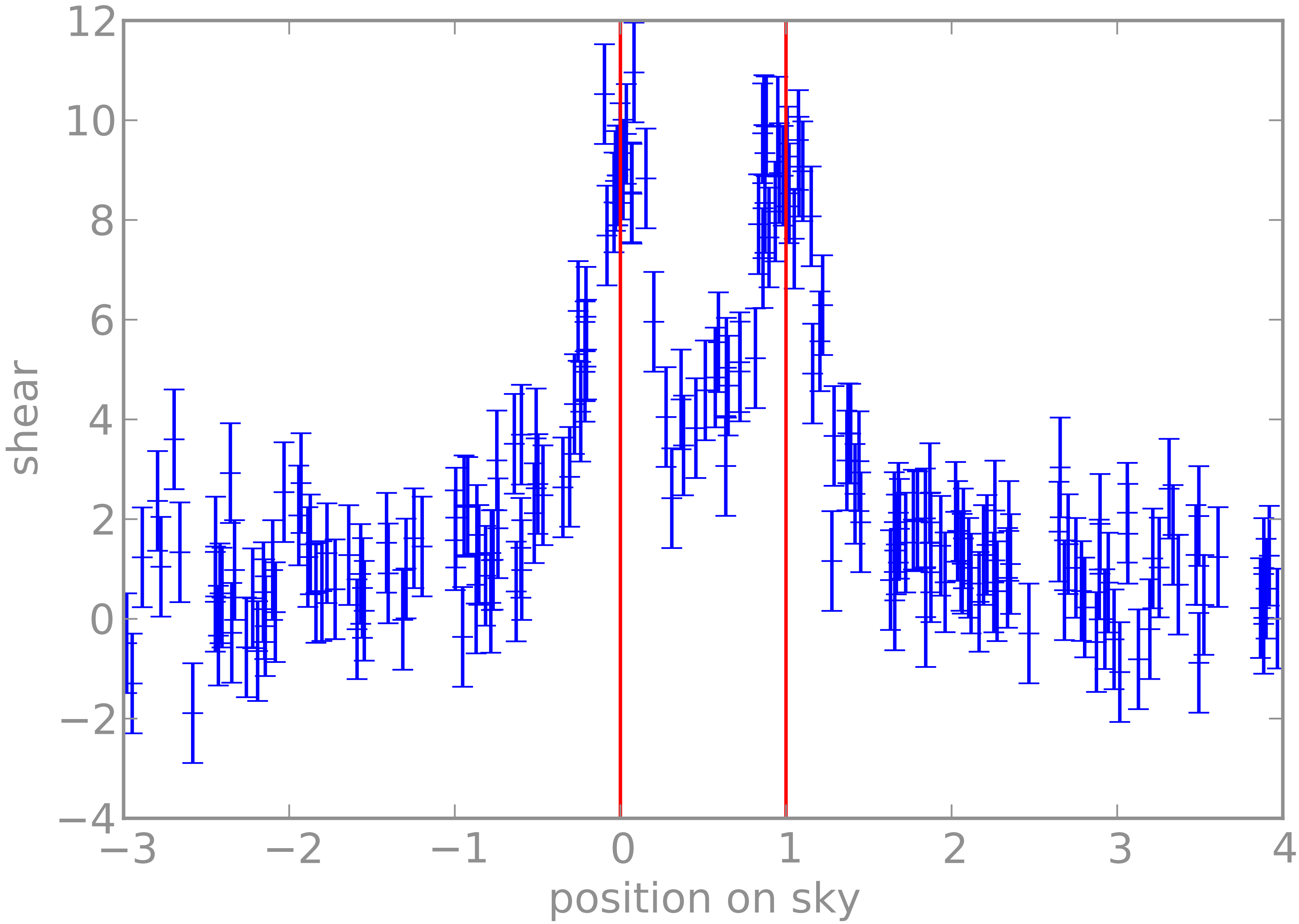
Cost: $\mathrm{RMSE}/1000 + G$



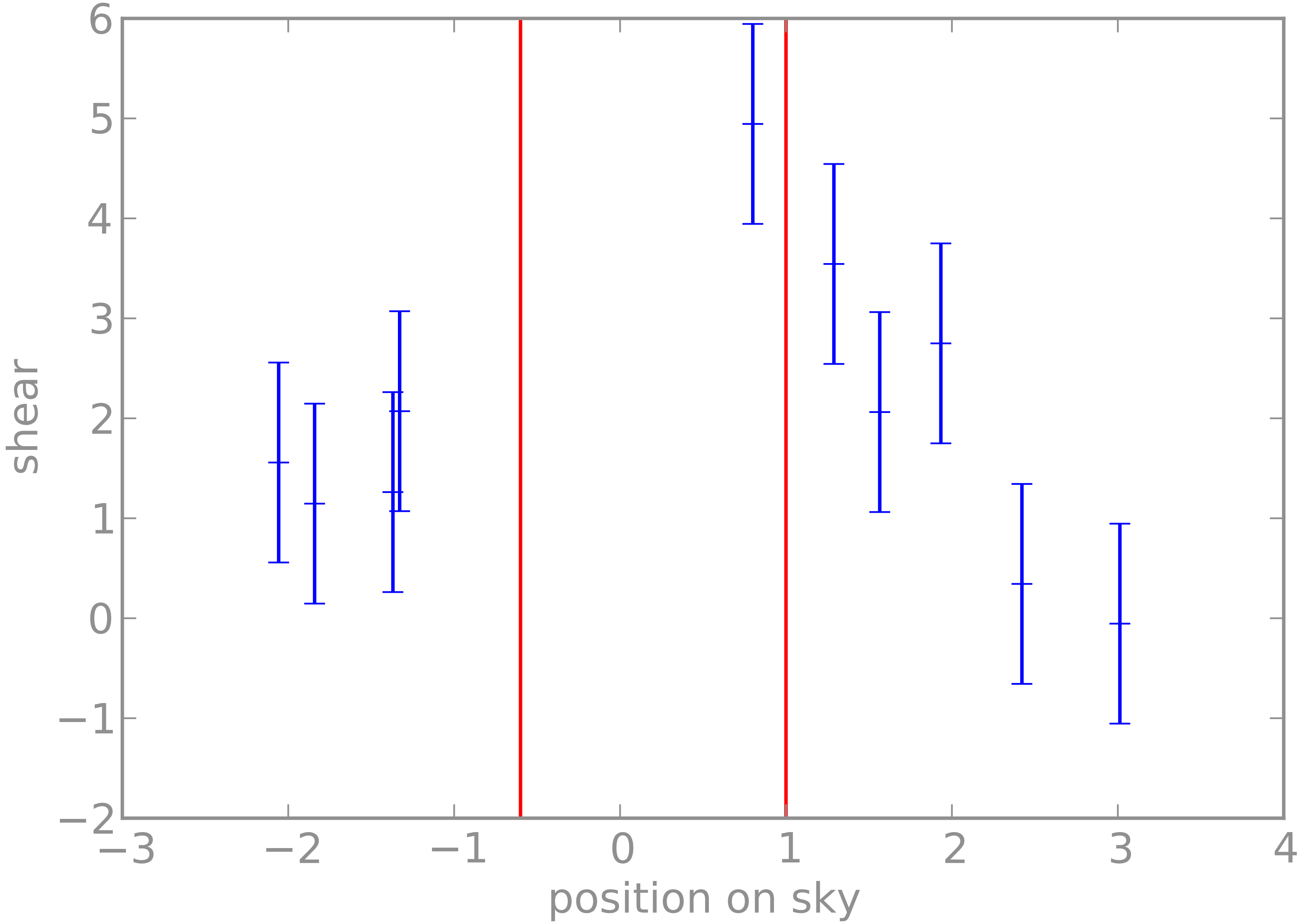
$$G = \sqrt{\left(\frac{1}{N}\sum_{n=1}^{N} \cos\phi_n\right)^2 + \left(\frac{1}{N}\sum_{n=1}^{N} \sin\phi_n\right)^2}$$

http://www.kaggle.com/c/DarkWorlds/details/evaluation

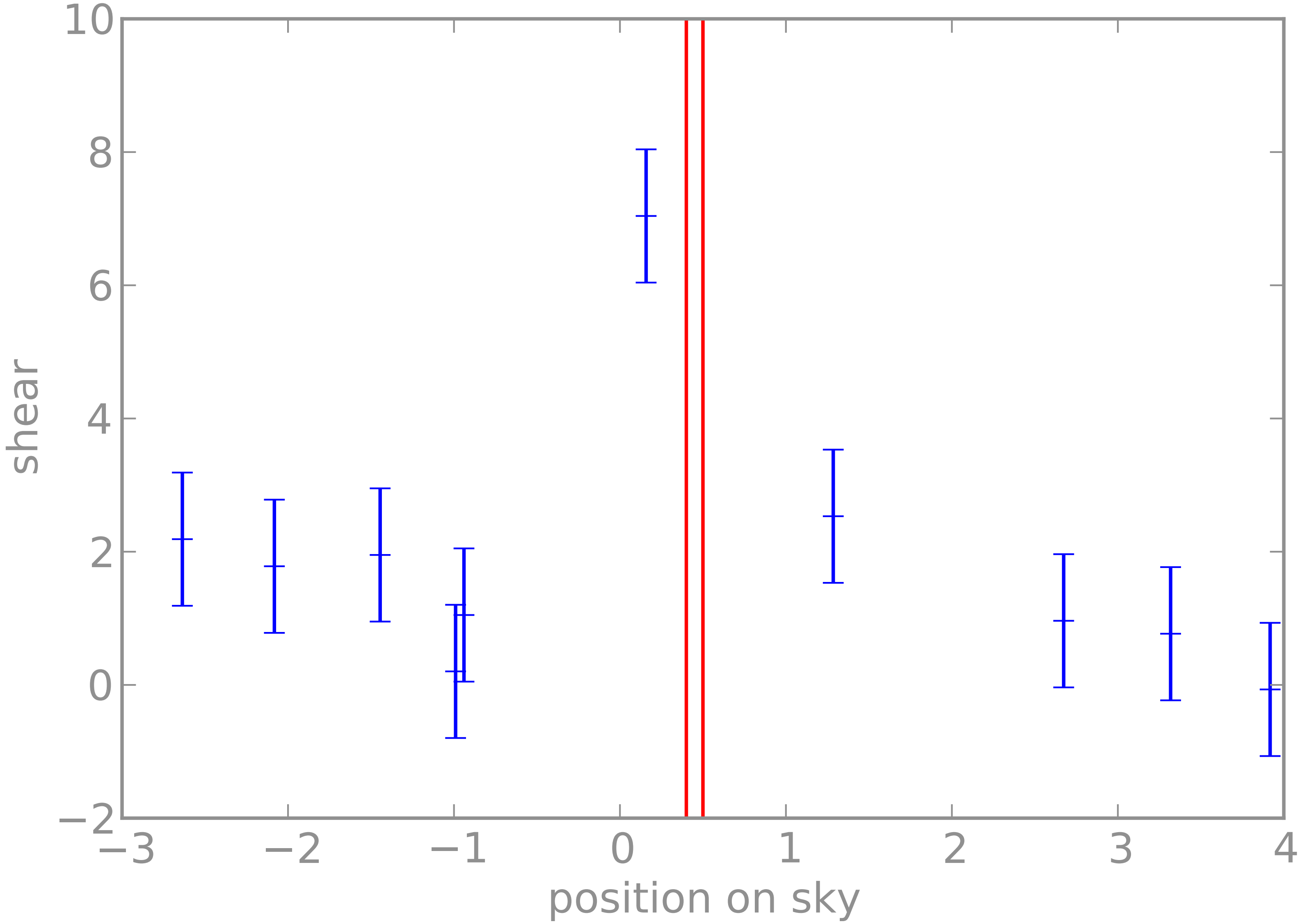
Toy demo



Toy demo

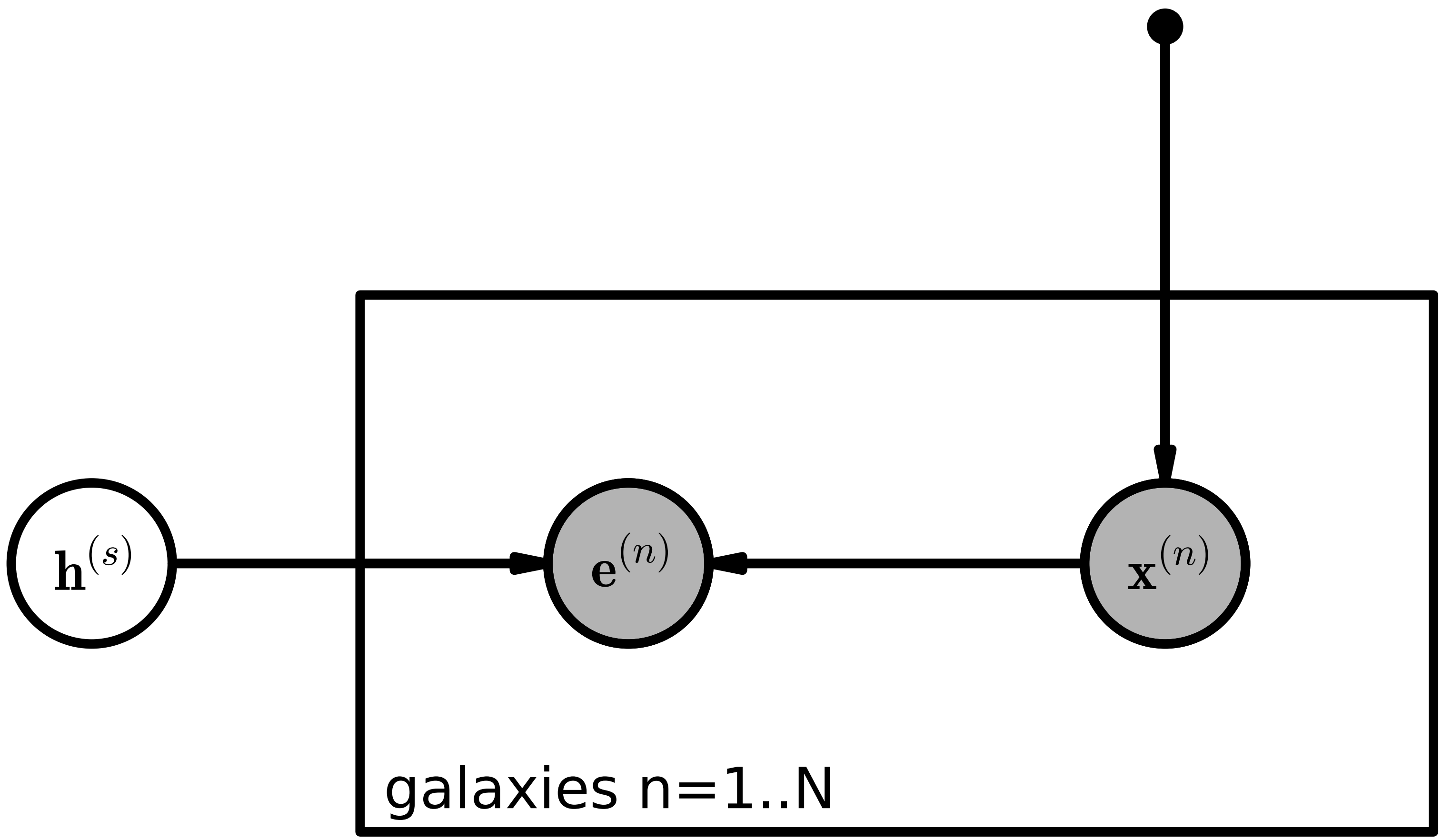


Toy demo



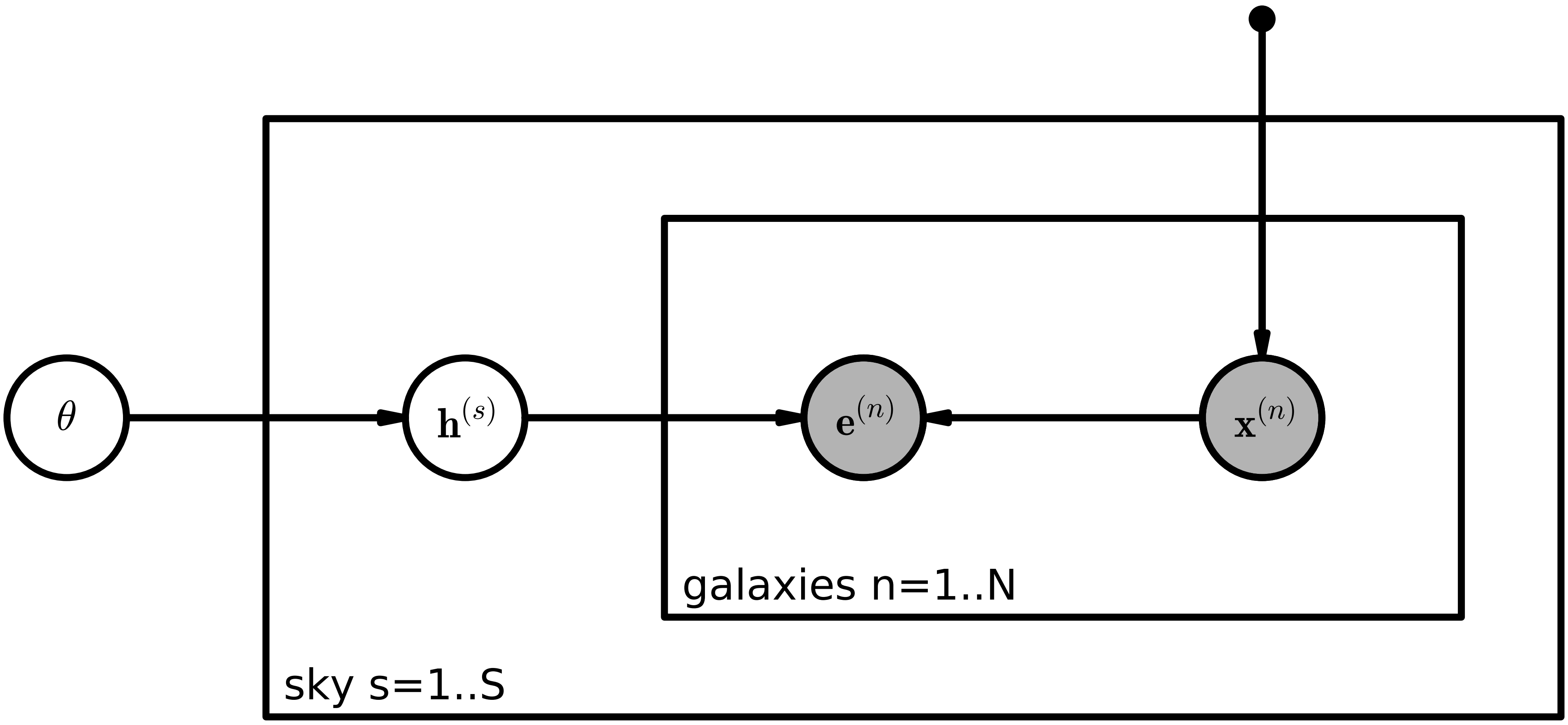
Ave. max. likelihood separation = 0.96, 4% too close

Graphical model



Made with `http://daft-pgm.org/`

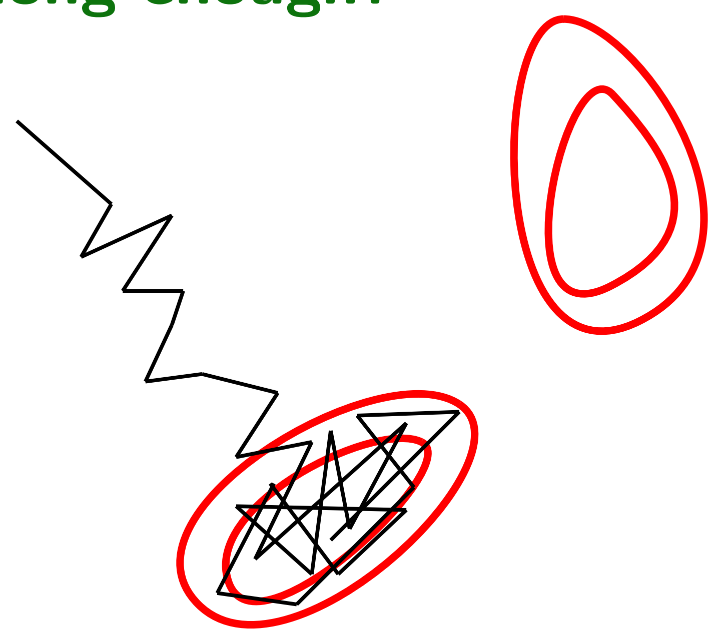
Graphical model



Made with `http://daft-pgm.org/`

How should we run MCMC?

- The samples aren't independent. Should we **thin**, only keep every Kth sample?
- Arbitrary initialization means starting iterations are bad. Should we discard a **"burn-in" period**?
- Maybe we should perform **multiple runs?**
- How do we know if we have run for **long enough?**



Forming estimates

Approximately independent samples can be obtained by *thinning*.
However, **all the samples can be used.**

Use the simple Monte Carlo estimator on MCMC samples. It is:

— consistent

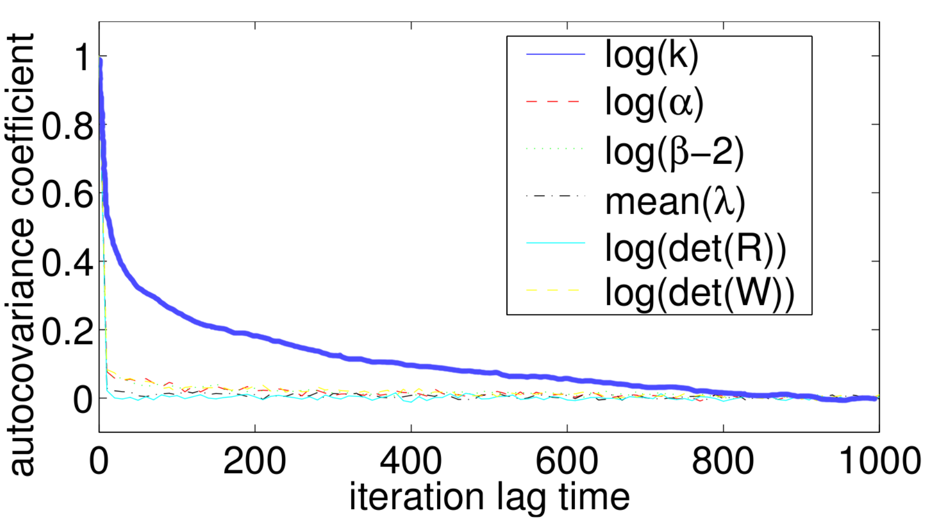
— unbiased if the chain has "burned in"

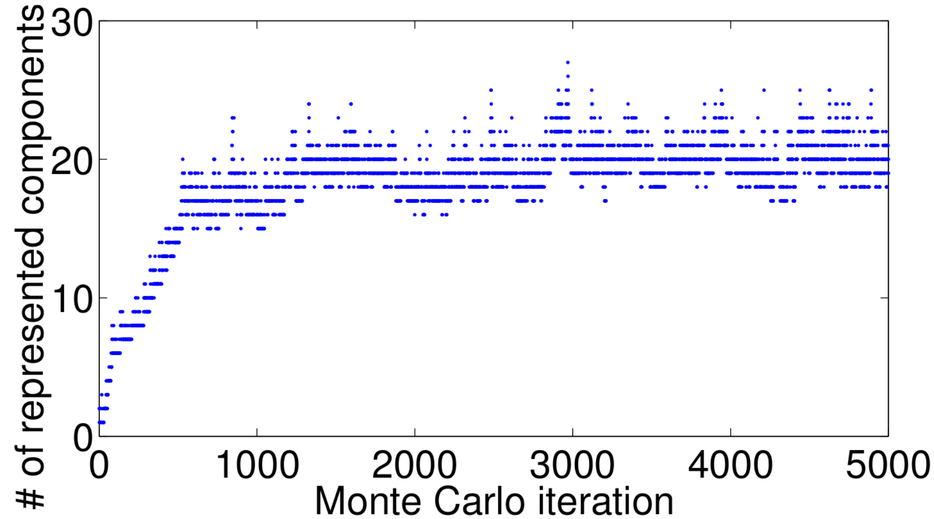
The correct motivation to thin: if computing $f(\mathbf{x}^{(s)})$ is expensive

In some special circumstances strategic thinning can help.

Steven N. MacEachern and Mario Peruggia, Statistics & Probability Letters, 47(1):91–98, 2000.
`http://dx.doi.org/10.1016/S0167-7152(99)00142-X` — Thanks to Simon Lacoste-Julien for the reference.

Empirical diagnostics





Rasmussen (2000)

Recommendations

For diagnostics:

Standard software packages like `R-CODA`

For opinion on thinning, multiple runs, burn in, etc.

Practical Markov chain Monte Carlo
Charles J. Geyer, *Statistical Science*. 7(4):473–483, 1992.
`http://www.jstor.org/stable/2246094`

Consistency checks

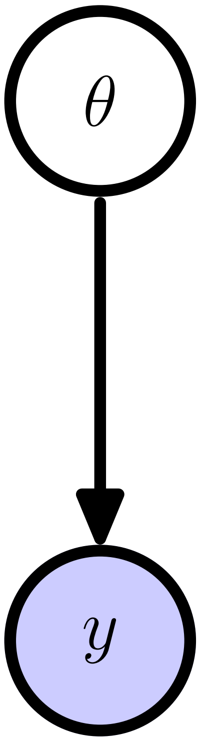
Do I get the right answer on tiny versions of my problem?

Can I make good inferences about synthetic data drawn from my model?

Getting it right: joint distribution tests of posterior simulators, John Geweke, *JASA*, 99(467):799–804, 2004.

Posterior Model checking: Gelman et al. Bayesian Data Analysis textbook and papers.

Getting it right



We write MCMC code to update $\theta \,|\, y$

Idea: also write code to sample $y \,|\, \theta$

Both codes leave $P(\theta, y)$ invariant

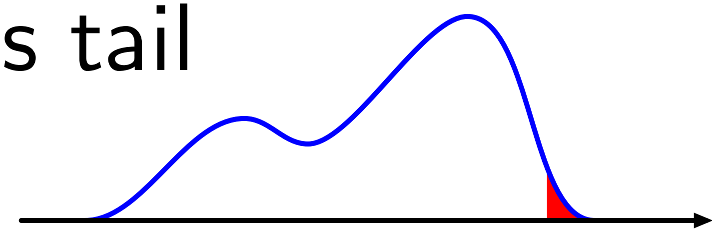
Run codes alternately. Check θ's match prior

Doing some analytic math

Collapsed sampler: marginalize some variables

Is the standard estimator too noisy?

e.g. need many samples from a distribution to estimate its tail



Maybe we can use samples better

Finding $P(x_i = 1)$

Method 1: fraction of time $x_i = 1$

$$P(x_i = 1) = \sum_{x_i} \mathbb{I}(x_i = 1) P(x_i) \approx \frac{1}{S} \sum_{s=1}^{S} \mathbb{I}(x_i^{(s)}), \quad x_i^{(s)} \sim P(x_i)$$

Method 2: average of $P(x_i = 1 | \mathbf{x}_{\setminus i})$

$$\begin{aligned} P(x_i = 1) &= \sum_{\mathbf{x}_{\setminus i}} P(x_i = 1 | \mathbf{x}_{\setminus i}) P(\mathbf{x}_{\setminus i}) \\ &\approx \frac{1}{S} \sum_{s=1}^{S} P(x_i = 1 | \mathbf{x}_{\setminus i}^{(s)}), \quad \mathbf{x}_{\setminus i}^{(s)} \sim P(\mathbf{x}_{\setminus i}) \end{aligned}$$

Example of "Rao-Blackwellization". See also "waste recycling".

Processing samples

This is easy

$$I = \sum_{\mathbf{x}} f(x_i) P(\mathbf{x}) \approx \frac{1}{S} \sum_{s=1}^{S} f(x_i^{(s)}), \quad \mathbf{x}^{(s)} \sim P(\mathbf{x})$$

But this might be better

$$I = \sum_{\mathbf{x}} f(x_i) P(x_i | \mathbf{x}_{\setminus i}) P(\mathbf{x}_{\setminus i}) = \sum_{\mathbf{x}_{\setminus i}} \left(\sum_{x_i} f(x_i) P(x_i | \mathbf{x}_{\setminus i}) \right) P(\mathbf{x}_{\setminus i})$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \left(\sum_{x_i} f(x_i) P(x_i | \mathbf{x}_{\setminus i}^{(s)}) \right), \quad \mathbf{x}_{\setminus i}^{(s)} \sim P(\mathbf{x}_{\setminus i})$$

A more general form of “Rao-Blackwellization”.

Summary so far

- MCMC is general and often easy to implement
- Running it *is* a bit messy. . .
  . . . but there are some established procedures.
- There can be a choice of estimators

Can we prove anything?

It's usually hard to have many guarantees.

Sometimes convergence theory can be practical:

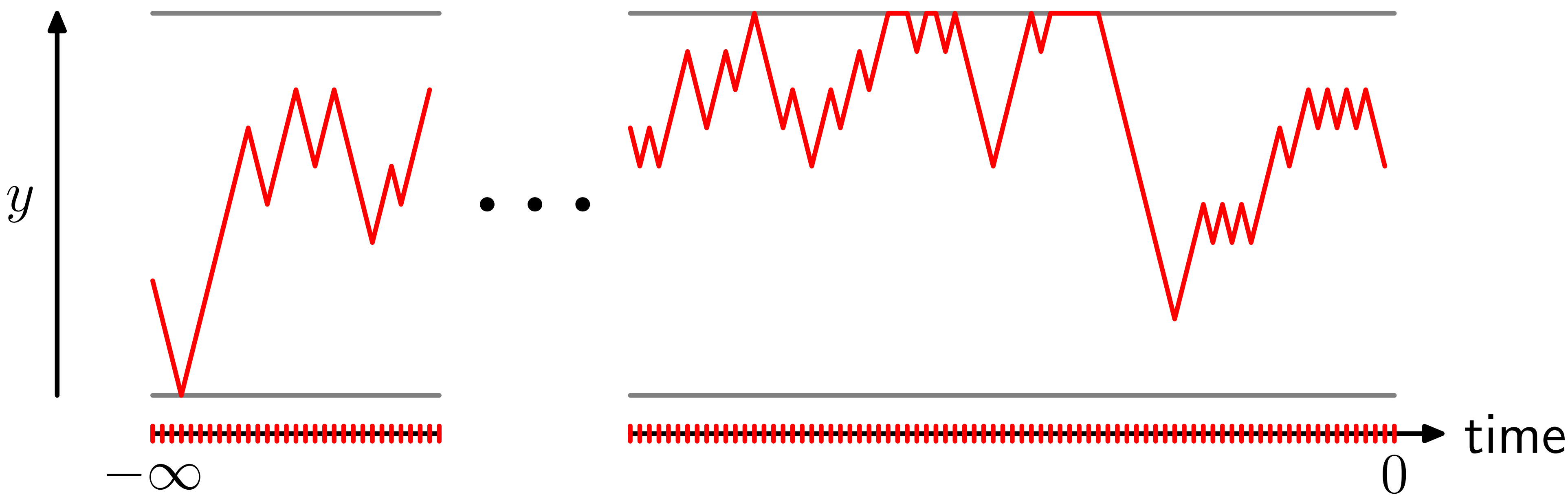
Markov chain Monte Carlo algorithms: theory and practice
Jeffrey S. Rosenthal
http://probability.ca/jeff/ftpdir/mcqmcproc.pdf

Text with more math than I give:

Monte Carlo Statistical Methods
Christian P. Robert, George Casella

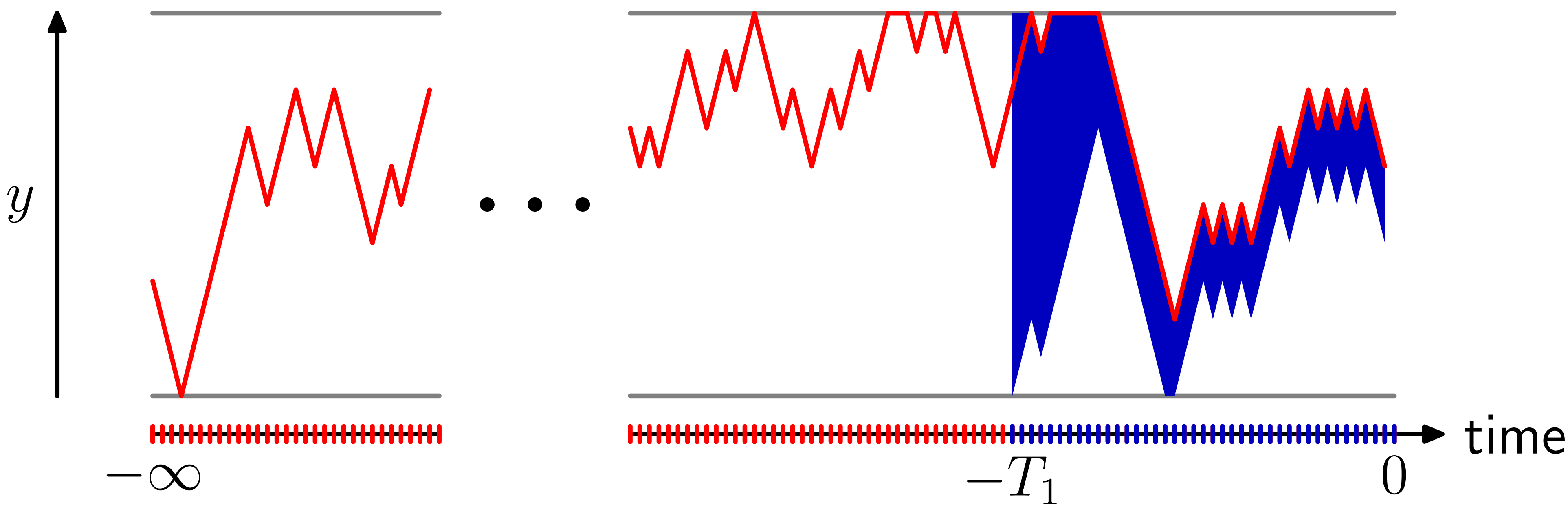
Exact sampling — *amazing* when it works

Exact sampling with MCMC



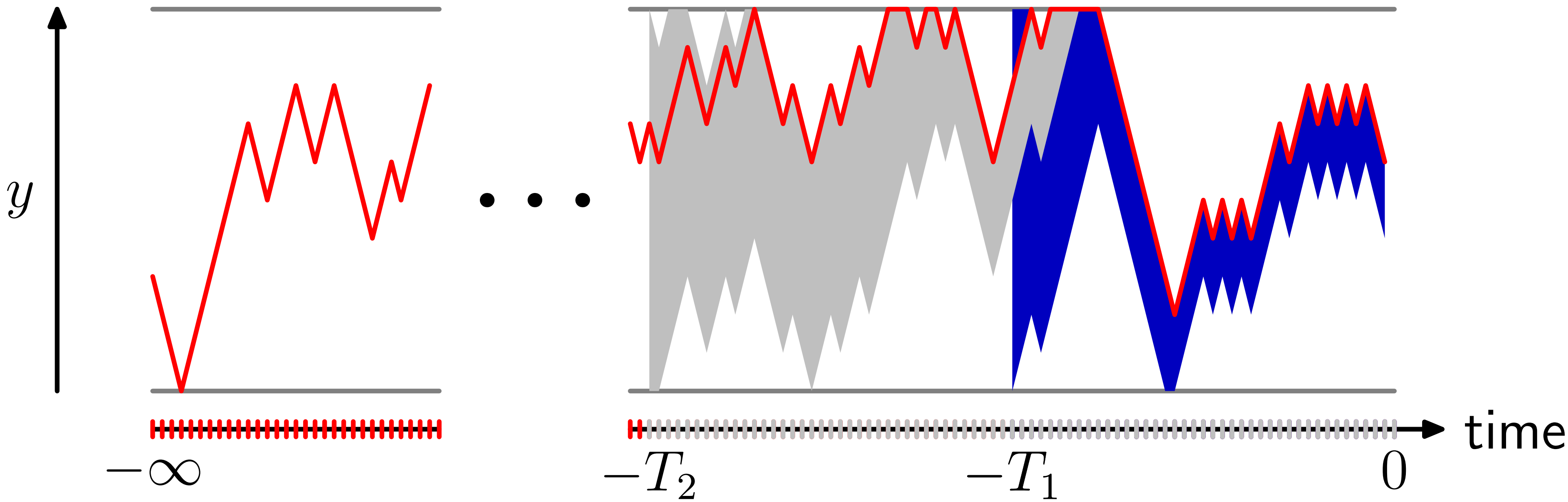
A chain that has run for ever

Exact sampling with MCMC



Try to find final state with finite number of random numbers

Exact sampling with MCMC



Takes a random amount of time.

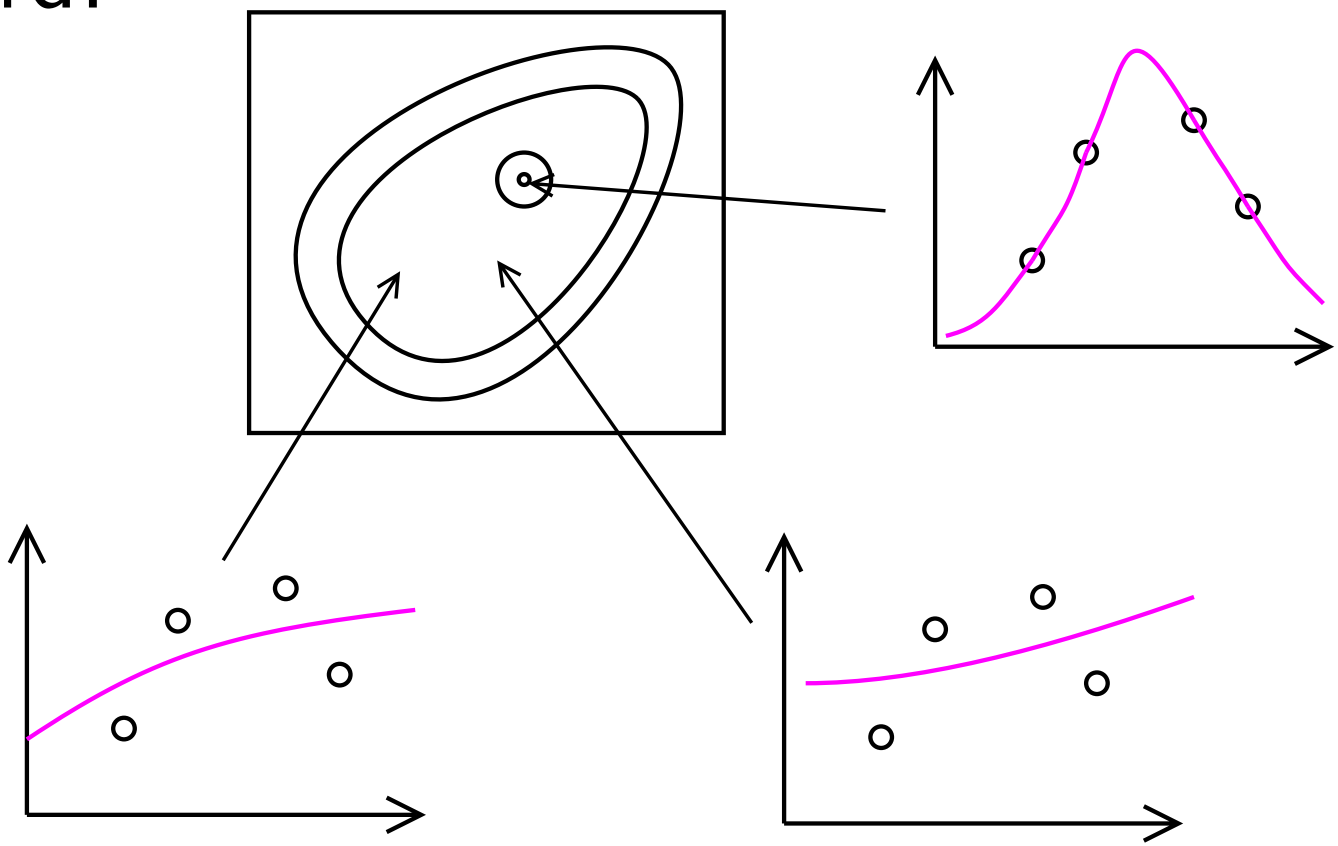
See `http://dbwilson.com/exact/`

(Google: "exact sampling" or "perfect sampling")

Building better chains

Come up with better proposals, Q?

Can be hard!



Many MCMC methods take a surprising approach. . .

Auxiliary variables

The point of MCMC is to marginalize out variables, but one can introduce more variables:

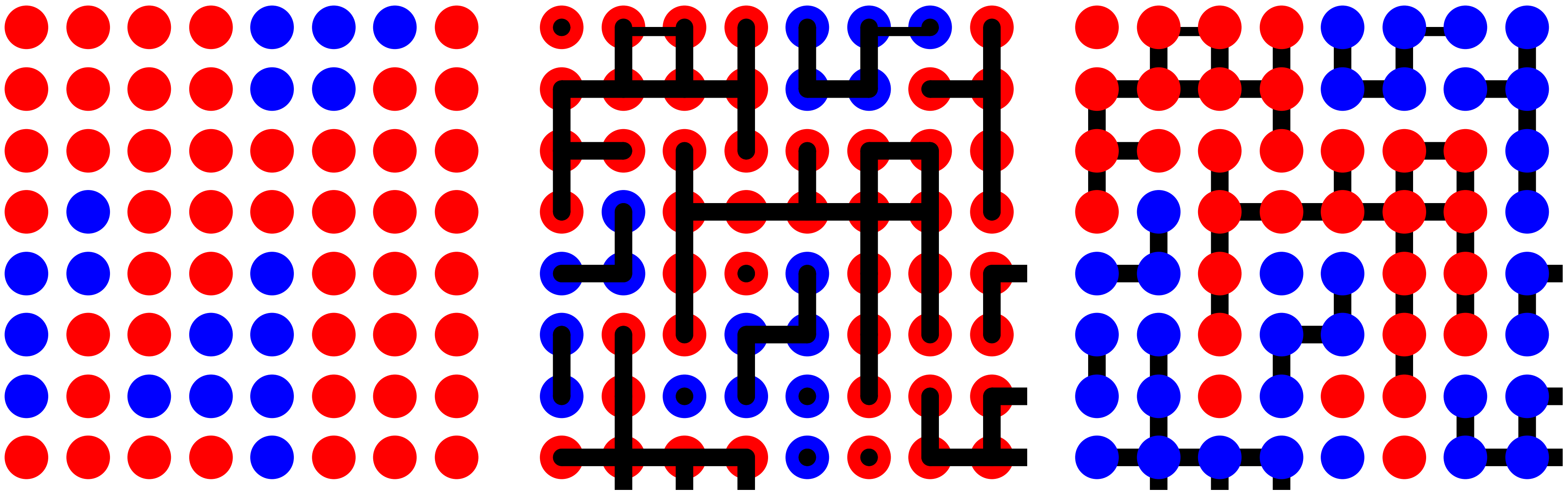
$$\int f(x)P(x)\,\mathrm{d}x = \int f(x)P(x,v)\,\mathrm{d}x\,\mathrm{d}v$$
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x,v \sim P(x,v)$$

We might want to introduce v if:

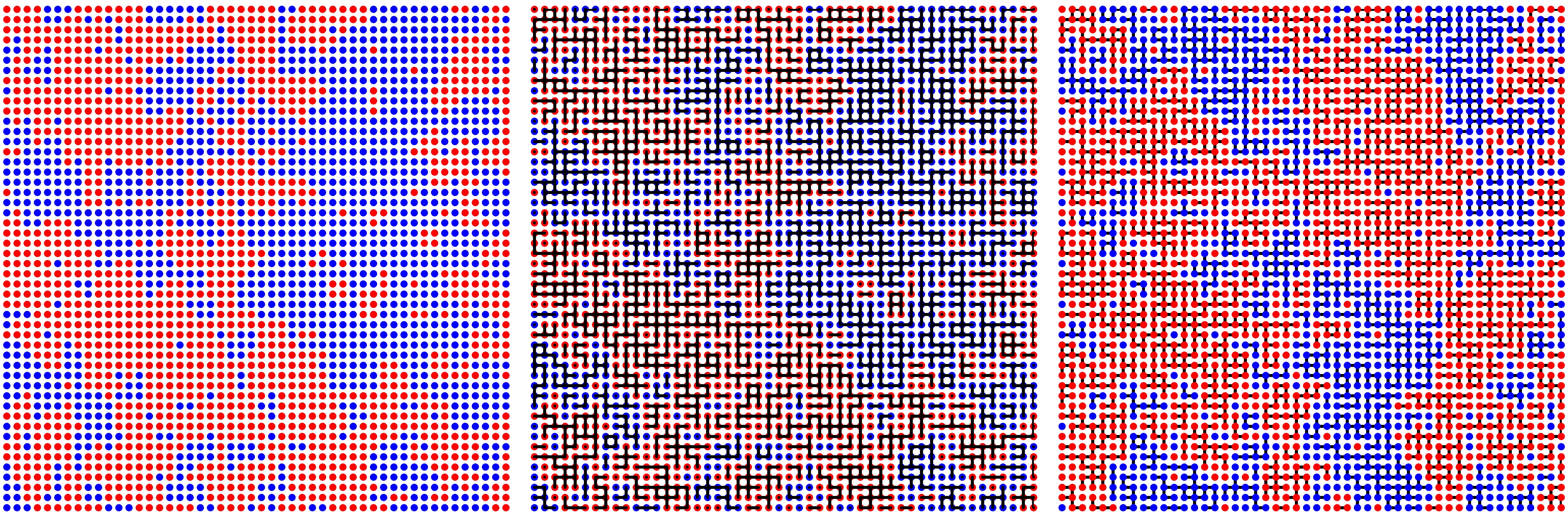
- $P(x|v)$ and $P(v|x)$ are simple
- $P(x,v)$ is otherwise easier to navigate

Swendsen–Wang (1987)

Seminal algorithm using auxiliary variables



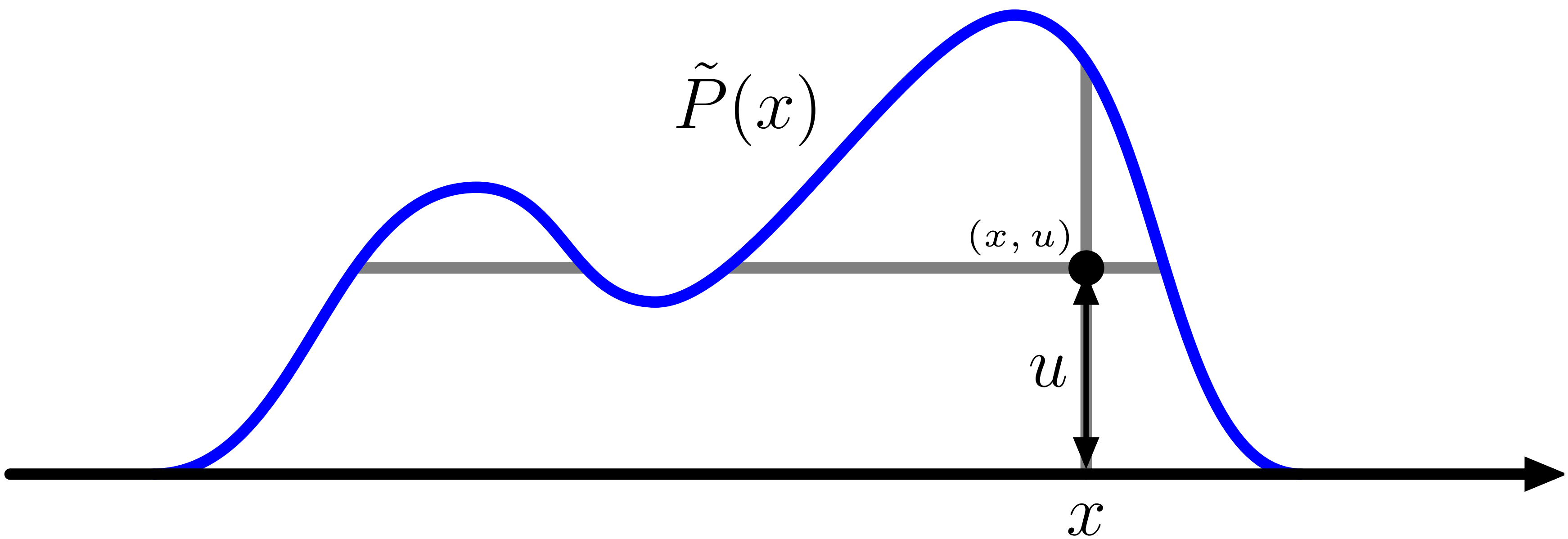
Swendsen–Wang (1987)



Edwards and Sokal (1988) identified and generalized the "Fortuin-Kasteleyn-Swendsen-Wang" auxiliary variable joint distribution that underlies the algorithm.

Slice sampling idea

Sample point uniformly under curve $\tilde{P}(x) \propto P(x)$

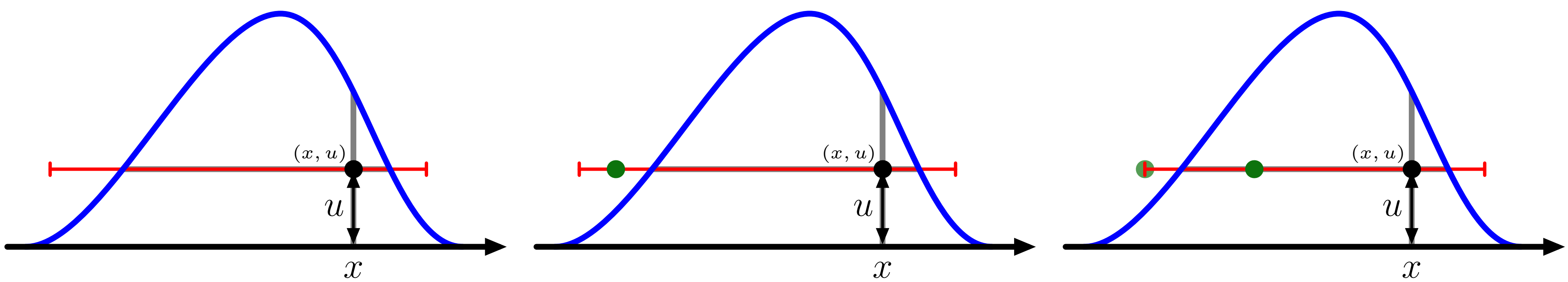


$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

$$p(x|u) \propto \begin{cases} 1 & \tilde{P}(x) \geq u \\ 0 & \text{otherwise} \end{cases} = \text{“Uniform on the slice”}$$

Slice sampling

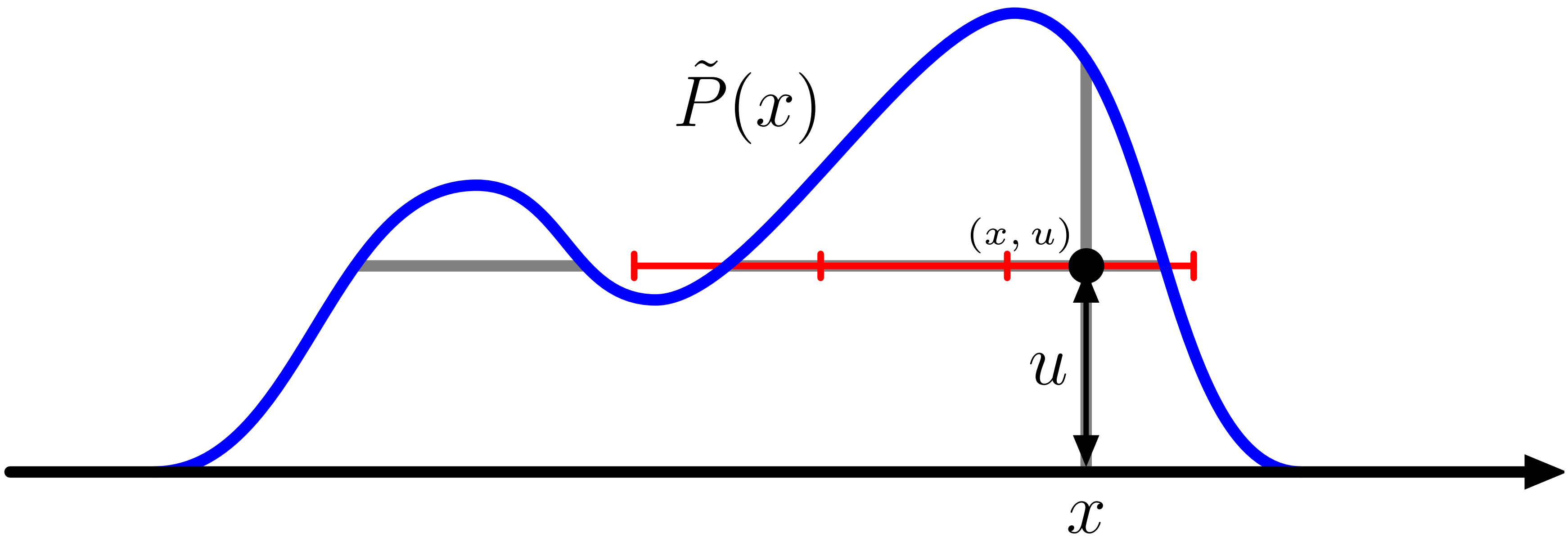
Unimodal conditionals



- bracket slice
- sample uniformly within bracket
- shrink bracket if $\tilde{P}(x) < u$ (off slice)
- accept first point on the slice

Slice sampling

Multimodal conditionals



- place bracket randomly around point
- linearly step out until bracket ends are off slice
- sample on bracket, shrinking as before

Satisfies detailed balance, leaves $p(x|u)$ invariant

Slice sampling

Advantages of slice-sampling:

- Easy — only require $\tilde{P}(x) \propto P(x)$ pointwise
- No rejections
- Tweak params less important than Metropolis

More advanced versions of slice sampling have been developed.
Neal (2003) contains *many* ideas.

Hamiltonian dynamics

Construct a landscape

Gravitational potential energy, E(x):

$$P(x) \propto e^{-E(x)}, \qquad E(x) = -\log P^*(x)$$

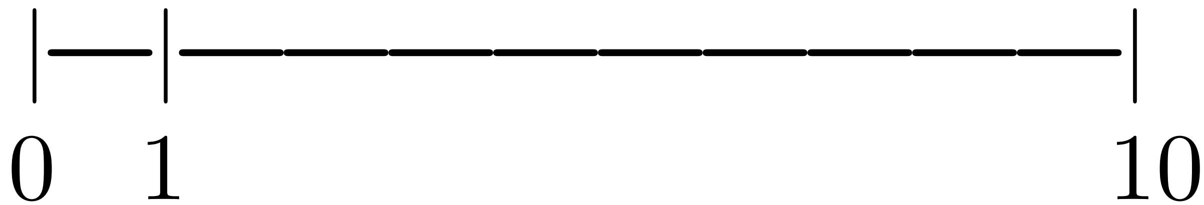
Roll a ball with velocity v

$$P(x, v) = e^{-E(x) - v^\top v/2}$$

Recommended reading:

MCMC using Hamiltonian dynamics
Radford M. Neal, 2011, in Handbook of Markov Chain Monte Carlo
`http://www.cs.toronto.edu/~radford/ftp/ham-mcmc.pdf`

Example / warning



Proposal: $$\begin{cases} x_{t+1} = 9x_t + 1, & 0 < x_t < 1 \\ x_{t+1} = (x_t - 1)/9, & 1 < x_t < 10 \end{cases}$$

Accept move with probability:

$$\min\left(1,\ \frac{P(x')\,Q(x;x')}{P(x)\,Q(x';x)}\right) = \min\left(1,\ \frac{P(x')}{P(x)}\right) \quad (\text{Wrong!})$$

Summary of auxiliary variables

— Swendsen–Wang
— Slice sampling
— Hamiltonian (Hybrid) Monte Carlo

Some of my auxiliary representation work:

Doubly-intractable distributions

Population methods for better mixing (on parallel hardware)

Being robust to bad random number generators

Recent slice-sampling work

Parting thoughts

Please be careful running MCMC

Try Gibbs or slice sampling, then:

— Try to find a better representation

— Try to find a better Q, e.g., data-driven MCMC

— Consider fancier methods

Remember operators can be concatenated

(Mix in simple updates with fancy ones)

Finding normalizers is hard

Standard Monte Carlo problem?

$$P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|\theta,\mathcal{M})\, P(\theta|\mathcal{M})\, \mathrm{d}\theta$$
$$= \frac{1}{S}\sum_{s=1}^{S} P(\mathcal{D}|\theta^{(s)},\mathcal{M}), \quad \theta^{(s)} \sim P(\theta|\mathcal{M})$$

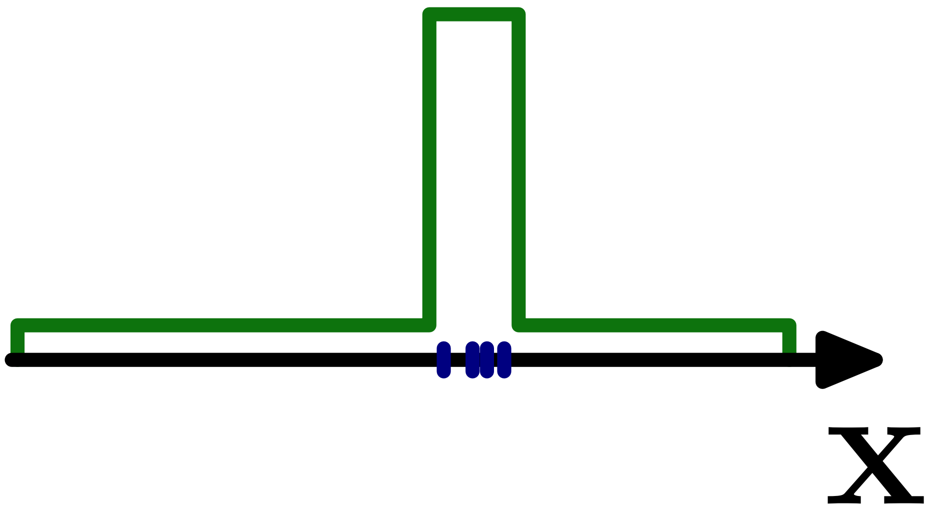
. . . usually has huge variance

Similarly for undirected graphs:

$$P(\mathbf{x}) = \frac{P^*(\mathbf{x})}{\mathcal{Z}}, \qquad \mathcal{Z} = \sum_{\mathbf{x}} P^*(\mathbf{x})$$

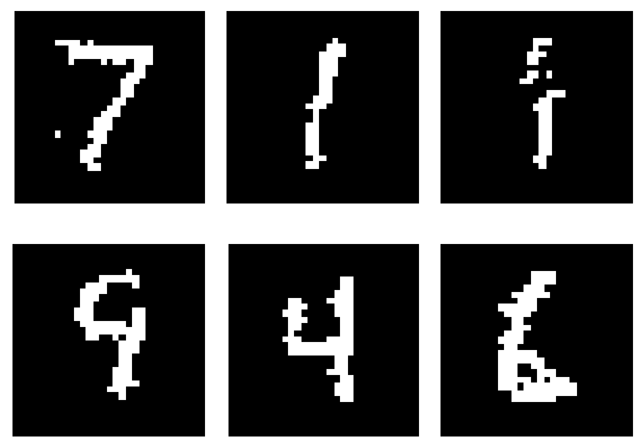
I will use this as an easy-to-illustrate case-study

$\mathcal{Z}$ a normalizer



$\mathbf{x} \sim$ Uniform

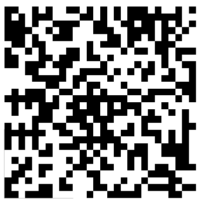
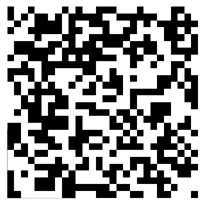
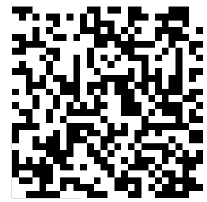
$\mathbf{x} \sim$ Model

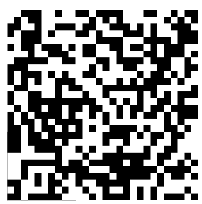


$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{\mathcal{Z}}$$

Simple Importance Sampling

$$\mathcal{Z} \;=\; \sum_{\mathbf{x}} \frac{P^*(\mathbf{x})}{Q(\mathbf{x})} Q(\mathbf{x}) \;\approx\; \frac{1}{S} \sum_{s=1}^{S} \frac{P^*(\mathbf{x}^{(s)})}{Q(\mathbf{x})}, \quad \mathbf{x}^{(s)} \sim Q(\mathbf{x})$$

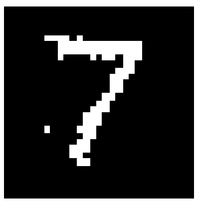
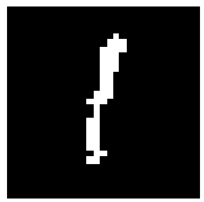
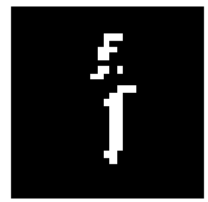
$\mathbf{x}^{(1)} =$ , $\mathbf{x}^{(2)} =$ , $\mathbf{x}^{(3)} =$ ,

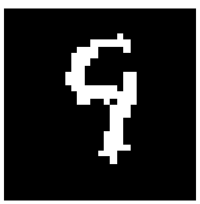
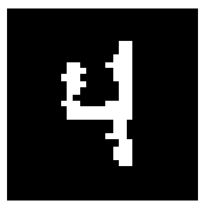
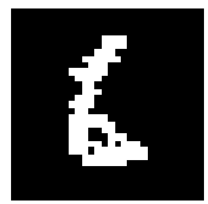
$\mathbf{x}^{(4)} =$ , $\mathbf{x}^{(5)} =$ , $\mathbf{x}^{(6)} =$ , $\ldots$

$$\mathcal{Z} \;=\; 2^D \sum_{\mathbf{x}} \frac{1}{2^D} P^*(\mathbf{x}) \;\approx\; \frac{2^D}{S} \sum_{s=1}^{S} P^*(\mathbf{x}^{(s)}), \quad \mathbf{x}^{(s)} \sim \text{Uniform}$$

"Posterior" Sampling

Sample from $P(\mathbf{x}) = \dfrac{P^*(\mathbf{x})}{\mathcal{Z}}$, $\quad \left[\text{or } P(\theta|\mathcal{D}) = \dfrac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}\right]$

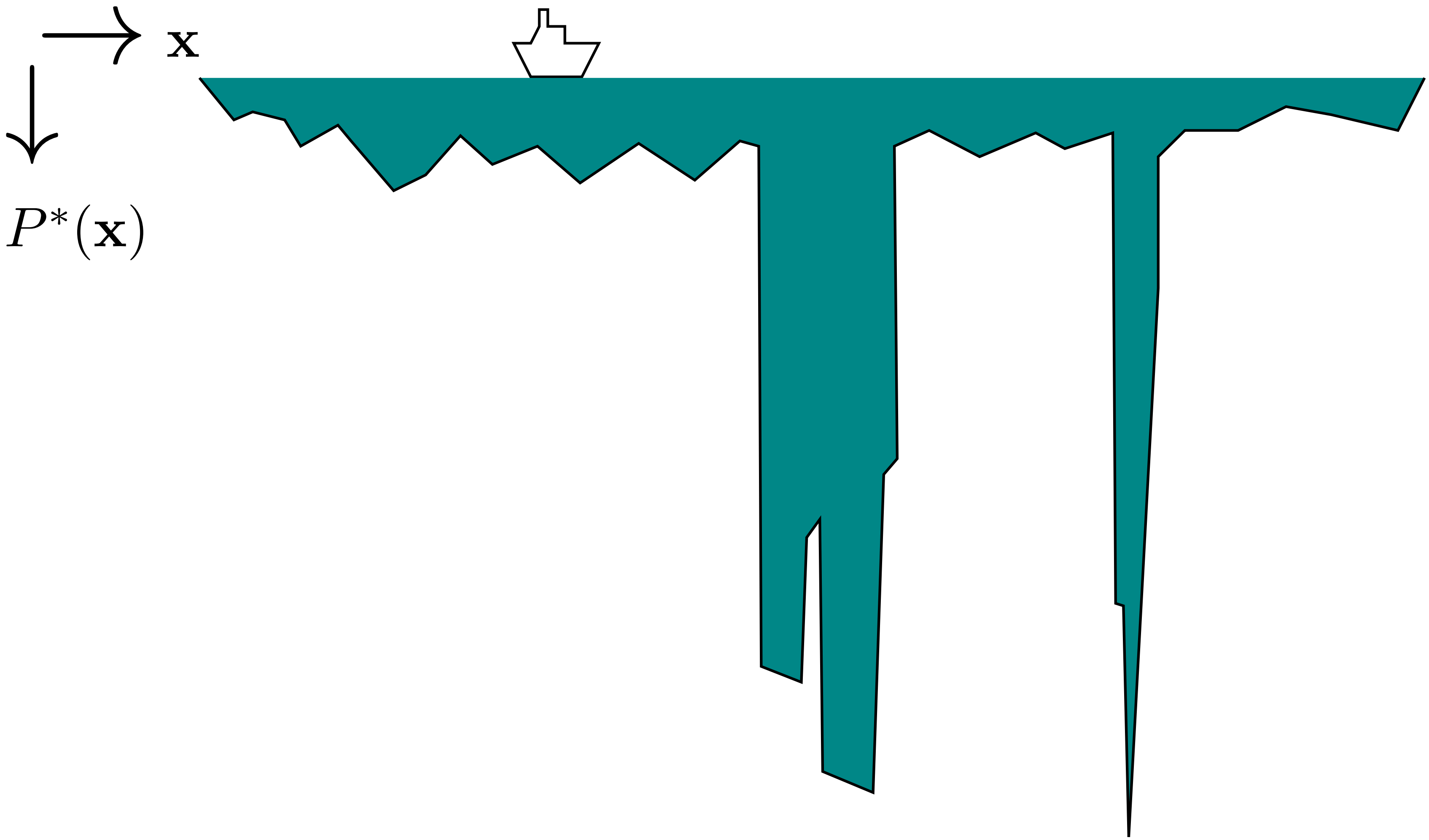
$\mathbf{x}^{(1)} =$ , $\quad \mathbf{x}^{(2)} =$ , $\quad \mathbf{x}^{(3)} =$ ,

$\mathbf{x}^{(4)} =$ , $\quad \mathbf{x}^{(5)} =$ , $\quad \mathbf{x}^{(6)} =$ , $\ldots$

$$\mathcal{Z} = \sum_{\mathbf{x}} P^*(\mathbf{x})$$

$$\mathcal{Z} \text{ ``}\approx\text{''} \; \frac{1}{S} \sum_{s=1}^{S} \frac{P^*(\mathbf{x})}{P(\mathbf{x})} = \mathcal{Z}$$

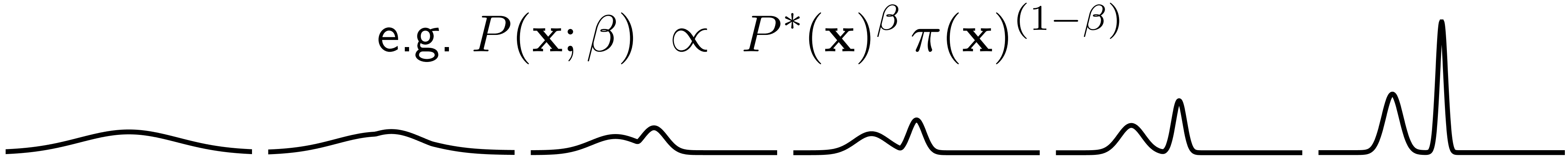
Finding a Volume



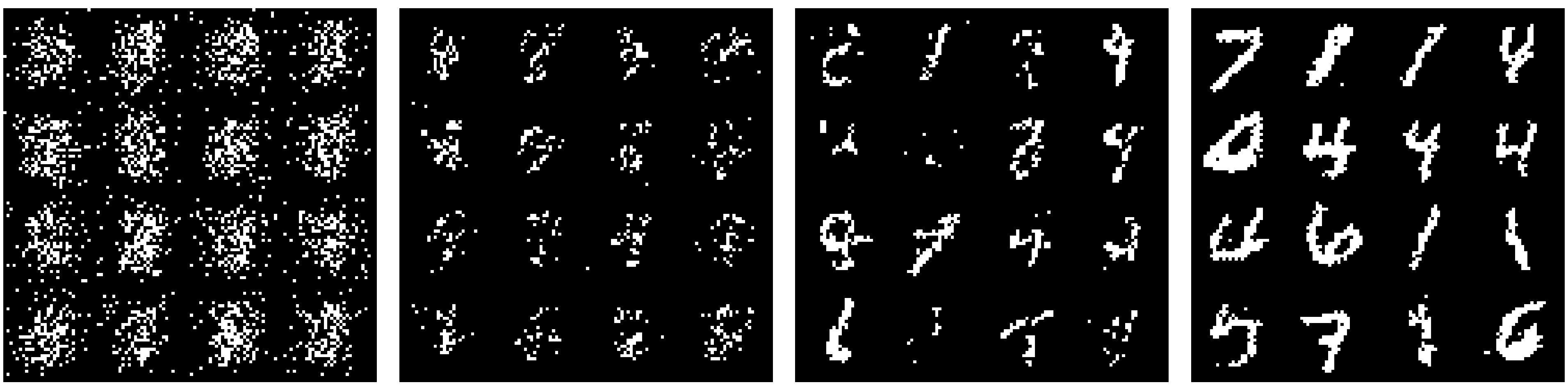
Lake analogy and figure from MacKay textbook (2003)

Annealing / Tempering

e.g. $P(\mathbf{x};\beta) \;\propto\; P^*(\mathbf{x})^\beta\,\pi(\mathbf{x})^{(1-\beta)}$



$\beta = 0$ $\beta = 0.01$ $\beta = 0.1$ $\beta = 0.25$ $\beta = 0.5$ $\beta = 1$

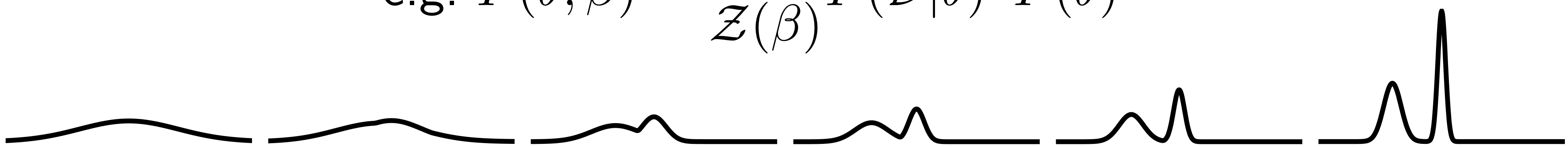


$1/\beta \;=\;$ “temperature”

Using other distributions

$Chain$ between posterior and prior:

$$\text{e.g. } P(\theta;\beta) = \frac{1}{\mathcal{Z}(\beta)} P(\mathcal{D}|\theta)^{\beta} P(\theta)$$



$\beta = 0$ $\quad$ $\beta = 0.01$ $\quad$ $\beta = 0.1$ $\quad$ $\beta = 0.25$ $\quad$ $\beta = 0.5$ $\quad$ $\beta = 1$

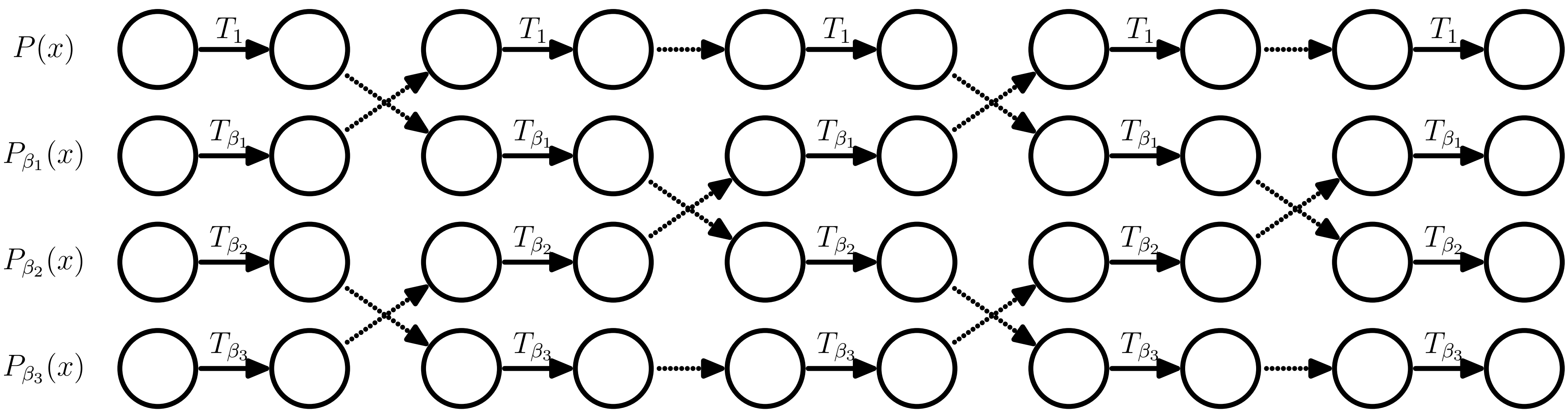
Advantages:

- mixing easier at low β, good initialization for higher β?
- $\dfrac{\mathcal{Z}(1)}{\mathcal{Z}(0)} = \dfrac{\mathcal{Z}(\beta_1)}{\mathcal{Z}(0)} \cdot \dfrac{\mathcal{Z}(\beta_2)}{\mathcal{Z}(\beta_1)} \cdot \dfrac{\mathcal{Z}(\beta_3)}{\mathcal{Z}(\beta_2)} \cdot \dfrac{\mathcal{Z}(\beta_4)}{\mathcal{Z}(\beta_3)} \cdot \dfrac{\mathcal{Z}(1)}{\mathcal{Z}(\beta_4)}$

Related to *annealing* or *tempering*, $1/\beta =$ "temperature"

Parallel tempering

Normal MCMC transitions + swap proposals on $P(X) = \prod_{\beta} P(X; \beta)$

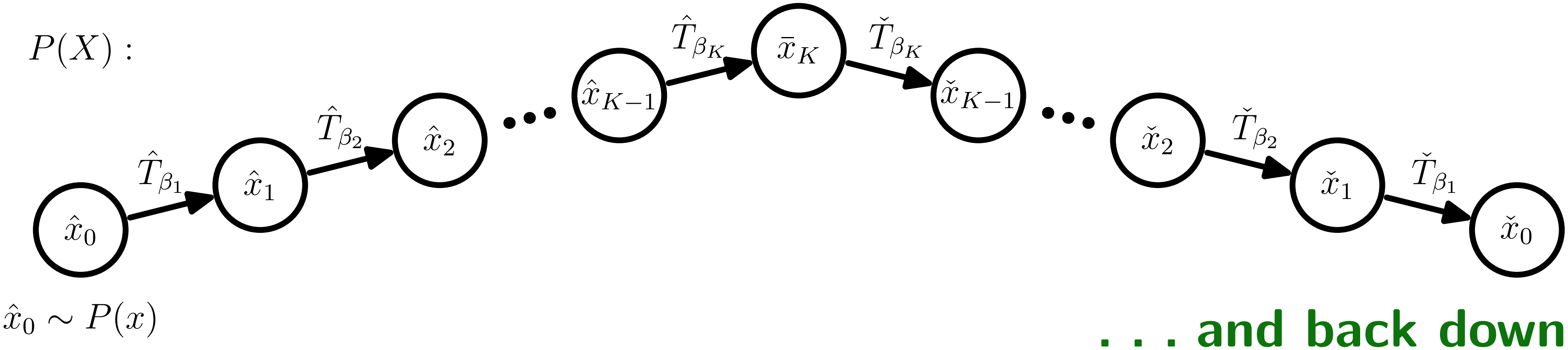


Problems / trade-offs:

- obvious space cost
- need to equilibriate larger system
- information from low β diffuses up by slow random walk

Tempered transitions

Drive temperature up. . .



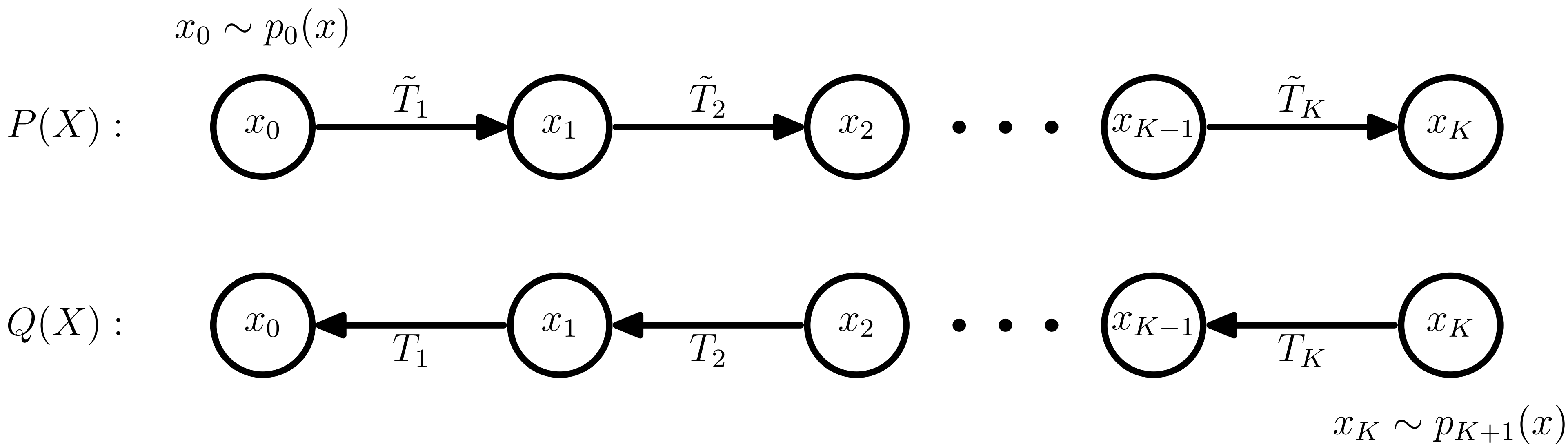
. . . and back down

Proposal: swap order of points so final point $\check{x}_0$ putatively $\sim P(x)$

Acceptance probability:

$$\min\left[1,\ \frac{P_{\beta_1}(\hat{x}_0)}{P(\hat{x}_0)} \cdots \frac{P_{\beta_K}(\hat{x}_{K-1})}{P_{\beta_{K-1}}(\hat{x}_0)} \frac{P_{\beta_{K-1}}(\check{x}_{K-1})}{P_{\beta_K}(\check{x}_{K-1})} \cdots \frac{P(\check{x}_0)}{P_{\beta_1}(\check{x}_0)}\right]$$

Annealed Importance Sampling

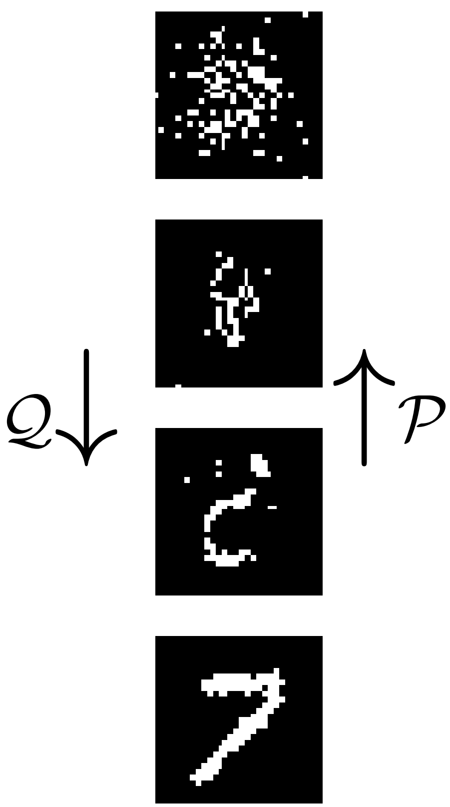


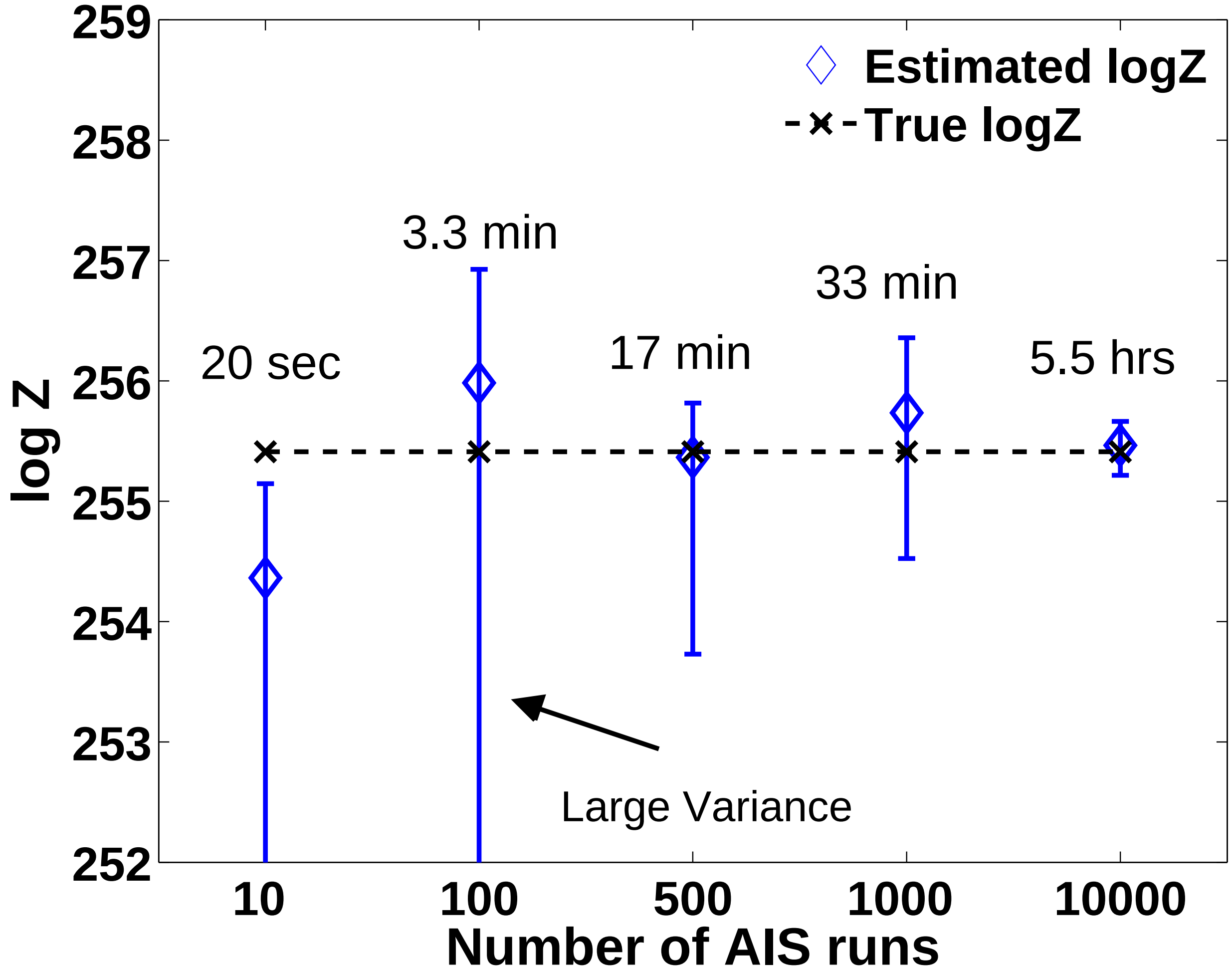
$$\mathcal{P}(X) = \frac{P^*(\mathbf{x}_K)}{\mathcal{Z}} \prod_{k=1}^{K} \widetilde{T}_k(\mathbf{x}_{k-1}; \mathbf{x}_k), \qquad \mathcal{Q}(X) = \pi(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k; \mathbf{x}_{k-1})$$

Then standard importance sampling of $\mathcal{P}(X) = \frac{\mathcal{P}^*(X)}{\mathcal{Z}}$ with $\mathcal{Q}(X)$

Annealed Importance Sampling

$$\mathcal{Z} \approx \frac{1}{S}\sum_{s=1}^{S}\frac{\mathcal{P}^*(X)}{\mathcal{Q}(X)}$$





Summary on $\mathcal{Z}$

Whirlwind tour of some estimators of $\mathcal{Z}$

Methods must be *good* at exploring the distribution

So watch these approaches for general use on the hardest problems.

See the references for more.

References

Further reading (1/2)

General references:

Probabilistic inference using Markov chain Monte Carlo methods, Radford M. Neal, Technical report: CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993. http://www.cs.toronto.edu/~radford/review.abstract.html

Various figures and more came from (see also references therein):

Advances in Markov chain Monte Carlo methods. Iain Murray. 2007. http://www.cs.toronto.edu/~murray/pub/07thesis/

Information theory, inference, and learning algorithms. David MacKay, 2003. http://www.inference.phy.cam.ac.uk/mackay/itila/

Pattern recognition and machine learning. Christopher M. Bishop. 2006. http://research.microsoft.com/~cmbishop/PRML/

Specific points:

If you do Gibbs sampling with continuous distributions this method, which I omitted for material-overload reasons, may help:

Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation, Radford M. Neal, *Learning in graphical models*, M. I. Jordan (editor), 205–228, Kluwer Academic Publishers, 1998. http://www.cs.toronto.edu/~radford/overk.abstract.html

An example of picking estimators carefully:

Speed-up of Monte Carlo simulations by sampling of rejected states, Frenkel, D, *Proceedings of the National Academy of Sciences*, 101(51):17571–17575, The National Academy of Sciences, 2004. http://www.pnas.org/cgi/content/abstract/101/51/17571

A key reference for auxiliary variable methods is:

Generalizations of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm, Robert G. Edwards and A. D. Sokal, *Physical Review*, 38:2009–2012, 1988.

Slice sampling, Radford M. Neal, *Annals of Statistics*, 31(3):705–767, 2003. http://www.cs.toronto.edu/~radford/slice-aos.abstract.html

Bayesian training of backpropagation networks by the hybrid Monte Carlo method, Radford M. Neal, Technical report: CRG-TR-92-1, Connectionist Research Group, University of Toronto, 1992. http://www.cs.toronto.edu/~radford/bbp.abstract.html

An early reference for parallel tempering:

Markov chain Monte Carlo maximum likelihood, Geyer, C. J, *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, 156–163, 1991.

Sampling from multimodal distributions using tempered transitions, Radford M. Neal, *Statistics and Computing*, 6(4):353–366, 1996.

Further reading (2/2)

Software:

Gibbs sampling for graphical models: `http://mathstat.helsinki.fi/openbugs/` `http://www-ice.iarc.fr/~martyn/software/jags/`
Neural networks and other flexible models: `http://www.cs.utoronto.ca/~radford/fbm.software.html`

CODA: http://www-fis.iarc.fr/coda/

Other Monte Carlo methods:

Nested sampling is a new Monte Carlo method with some interesting properties:
Nested sampling for general Bayesian computation, John Skilling, *Bayesian Analysis*, 2006.
(to appear, posted online June 5). `http://ba.stat.cmu.edu/journal/forthcoming/skilling.pdf`

Approaches based on the "multi-canonicle ensemble" also solve some of the problems with traditional tempterature-based methods:
Multicanonical ensemble: a new approach to simulate first-order phase transitions, Bernd A. Berg and Thomas Neuhaus, *Phys. Rev. Lett*, 68(1):9–12, 1992. `http://prola.aps.org/abstract/PRL/v68/i1/p9_1`

A good review paper:
Extended Ensemble Monte Carlo. Y Iba. Int J Mod Phys C [Computational Physics and Physical Computation] 12(5):623-656. 2001.

Particle filters / Sequential Monte Carlo are famously successful in time series modeling, but are more generally applicable.
This may be a good place to start: `http://www.cs.ubc.ca/~arnaud/journals.html`

Exact or perfect sampling uses Markov chain simulation but suffers no initialization bias. An amazing feat when it can be performed:
Annotated bibliography of perfectly random sampling with Markov chains, David B. Wilson
`http://dbwilson.com/exact/`

MCMC does not apply to *doubly-intractable* distributions. For what that even means and possible solutions see:
An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants, J. Møller, A. N. Pettitt, R. Reeves and K. K. Berthelsen, *Biometrika*, 93(2):451–458, 2006.
MCMC for doubly-intractable distributions, Iain Murray, Zoubin Ghahramani and David J. C. MacKay, *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Rina Dechter and Thomas S. Richardson (editors), 359–366, AUAI Press, 2006.
`http://www.gatsby.ucl.ac.uk/~iam23/pub/06doubly_intractable/doubly_intractable.pdf`