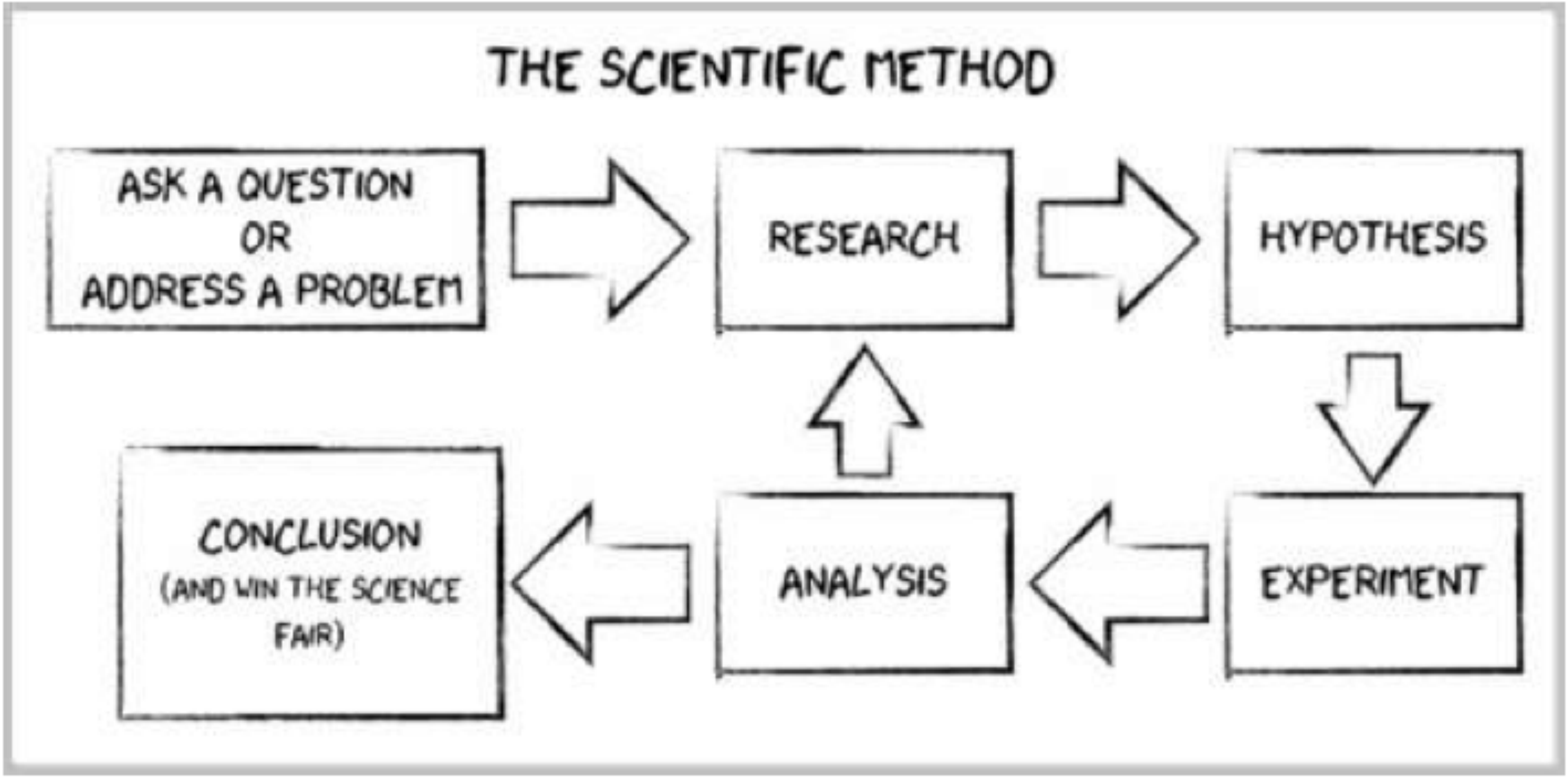
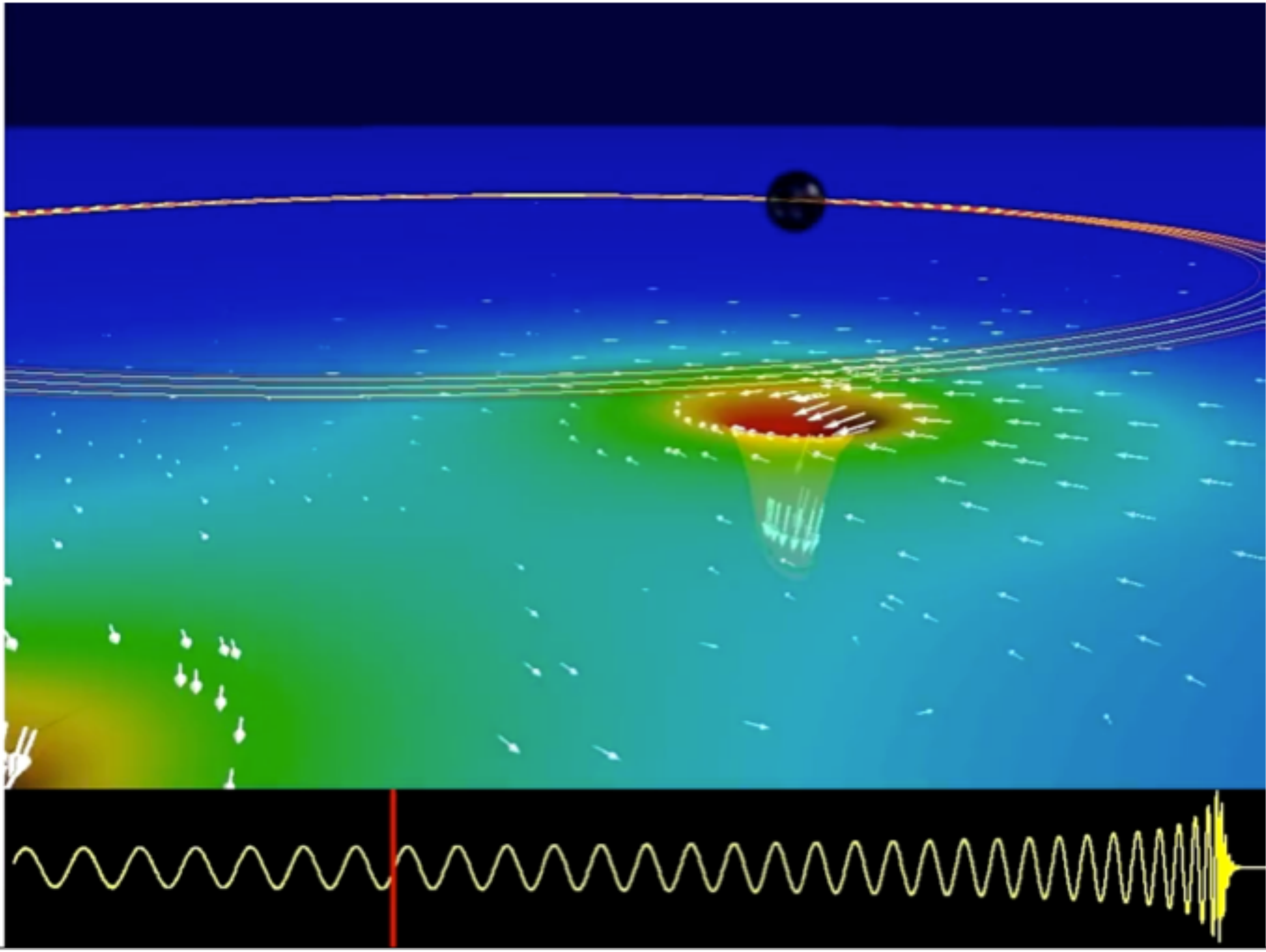
Using Probabilistic Programming for Bayesian Inference

Rhys Green
Gravity Exploration Institute, Cardiff University

The Scientific Process



Motivating example



Bayesian Inference/ Parameter estimation

- We need to extract/infer the most information possible from the data

- e.g Learn source parameters for gravitational waves

- Also want to make predictions about the future

- Need to do this in a statistically sound way with a measure of certainty

Bayes Theorem

This is the prior: i.e. what you believed before
you saw the evidence.

This is the likelihood of seeing that
evidence if your hypothesis is correct.

This is the posterior

$$p(H \mid D) = \frac{p(H)\,p(D \mid H)}{p(D)}$$

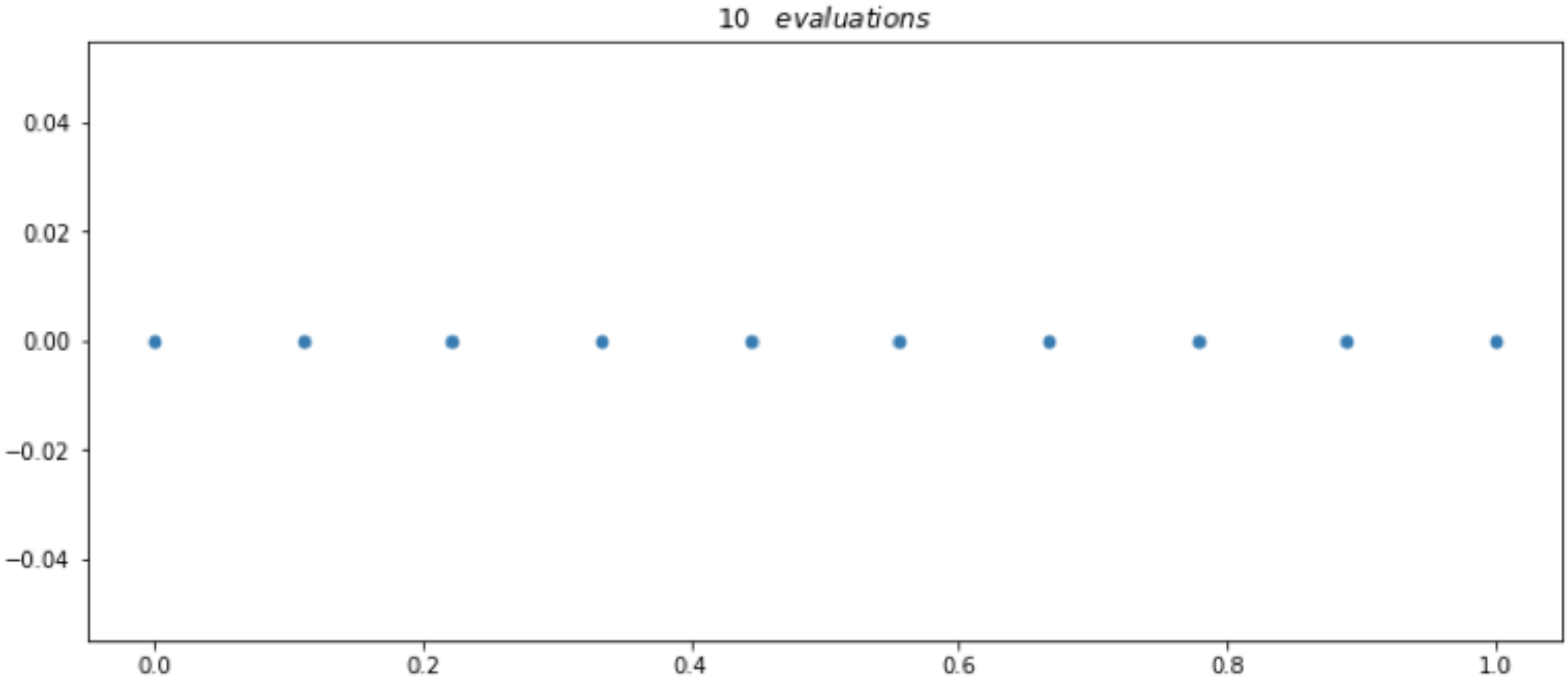
This is the normalizing constant:
i.e. The likelihood of that evidence under
any circumstances.

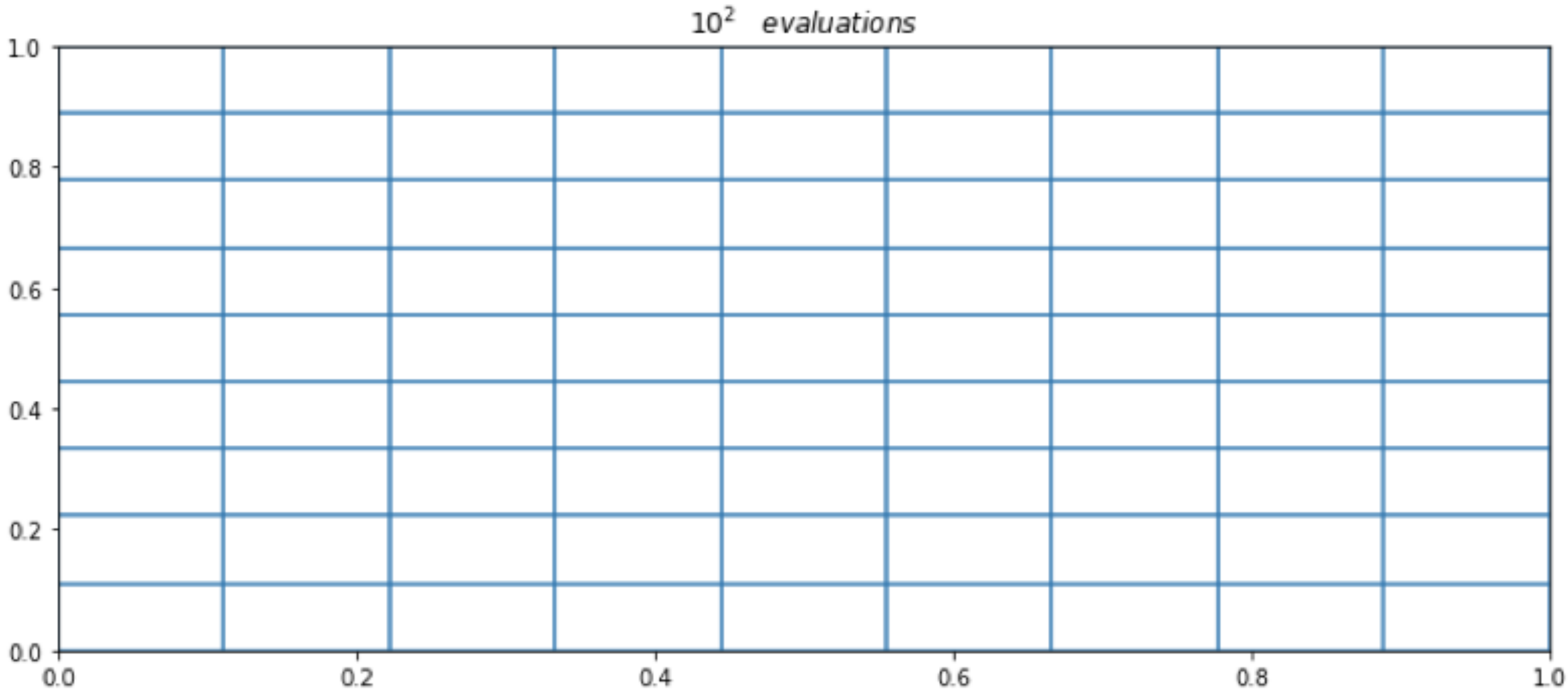
The Problem

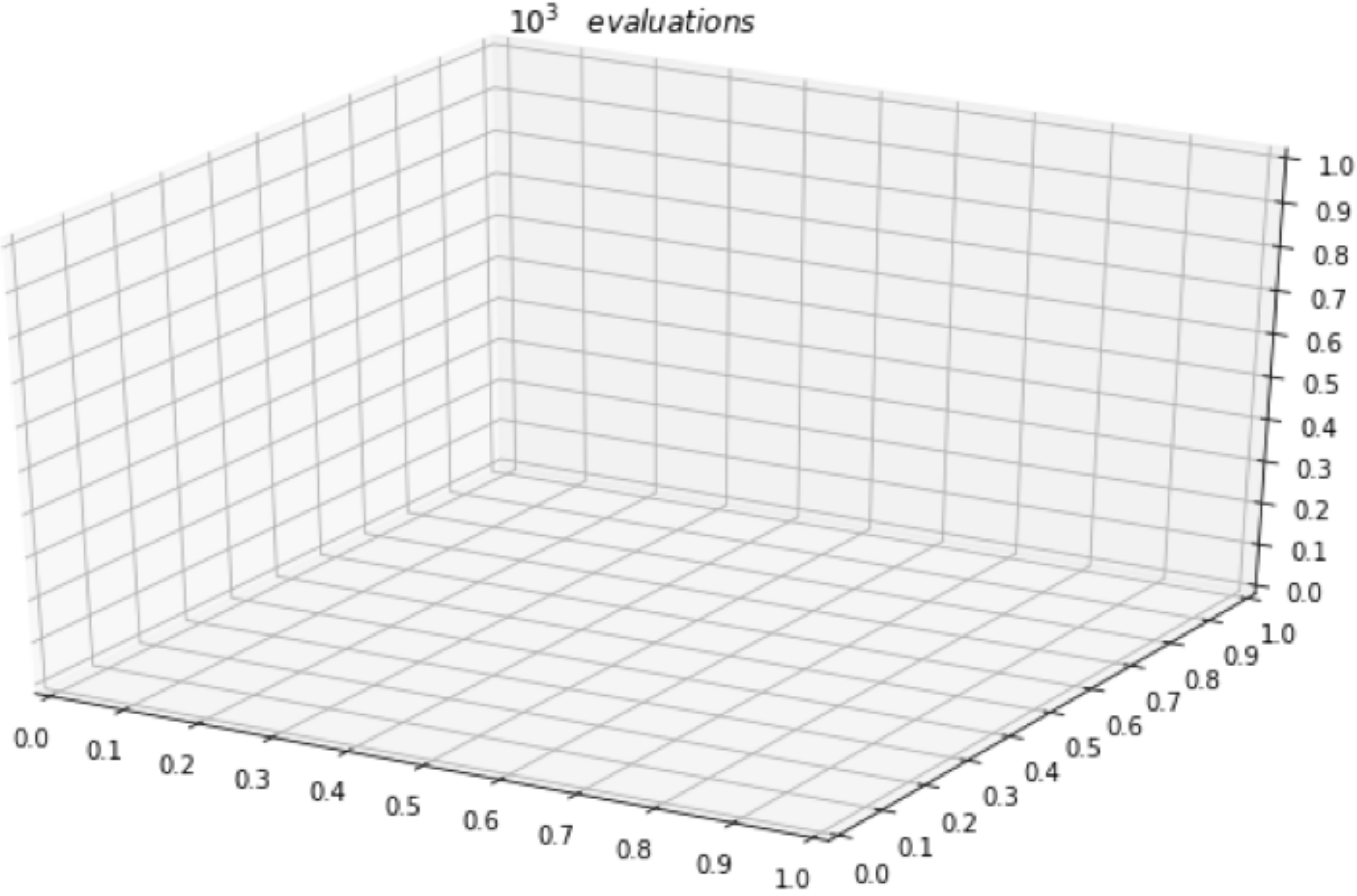
- In General not possible to calculate an analytic solution
- The evidence term is usually intractable
- $p(D) = \int p(D \mid H) p(H) dH \approx \sum_{i}^{n} p(D \mid H_i) p(H_i)$
- Need to resort to approximations

Sampling based solutions

- We can sample the parameter space and find the set of parameters that fits the data best
- If we take infinite samples across the parameter space we are guaranteed to find the true solution
- Taking infinite samples is clearly not possible
- Curse of dimensionality is a huge problem here

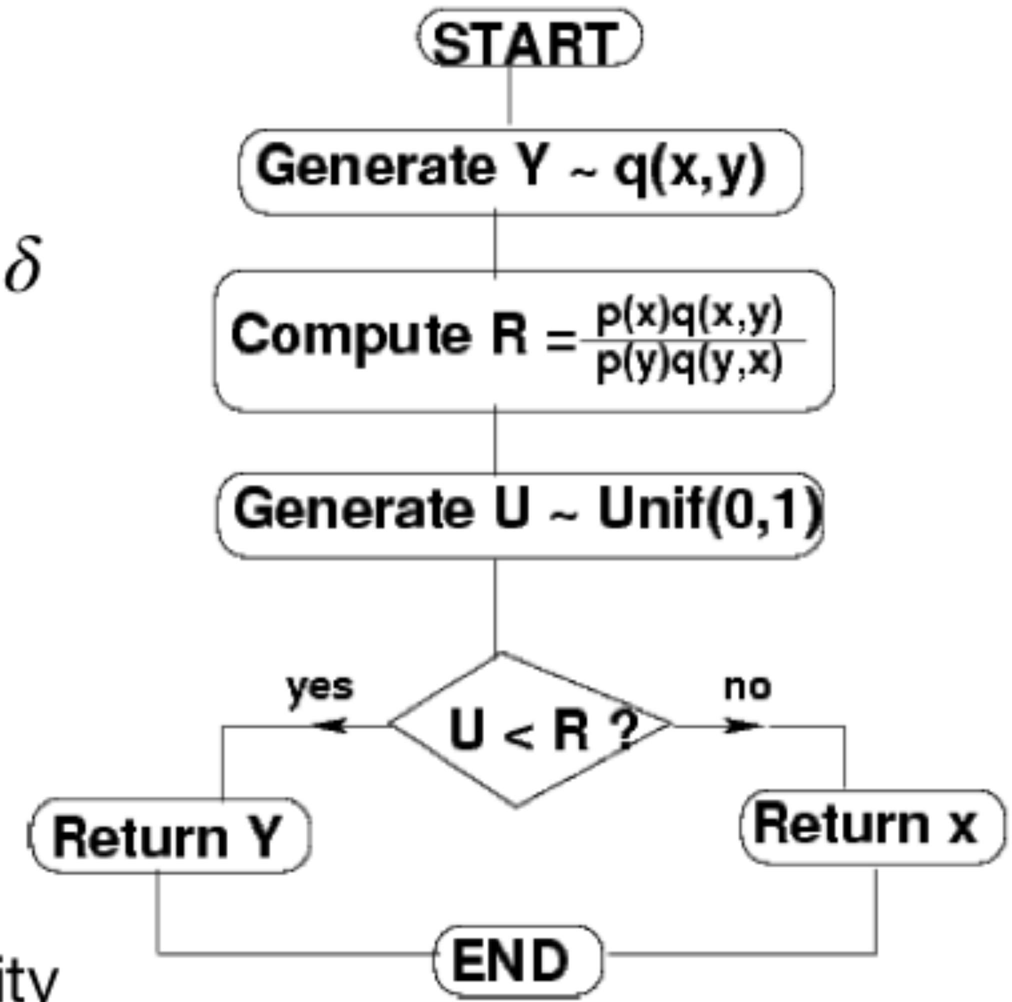

10 evaluations
0.04
0.02
0.00
−0.02
−0.04
0.0
0.2
0.4
0.6
0.8
1.0

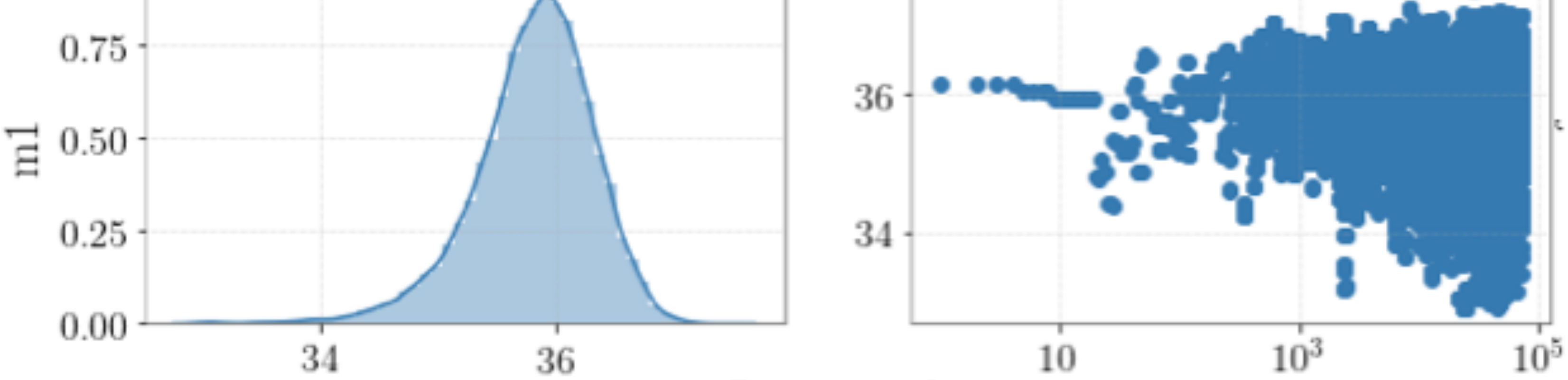

10^2 evaluations
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


10^3 evaluations
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

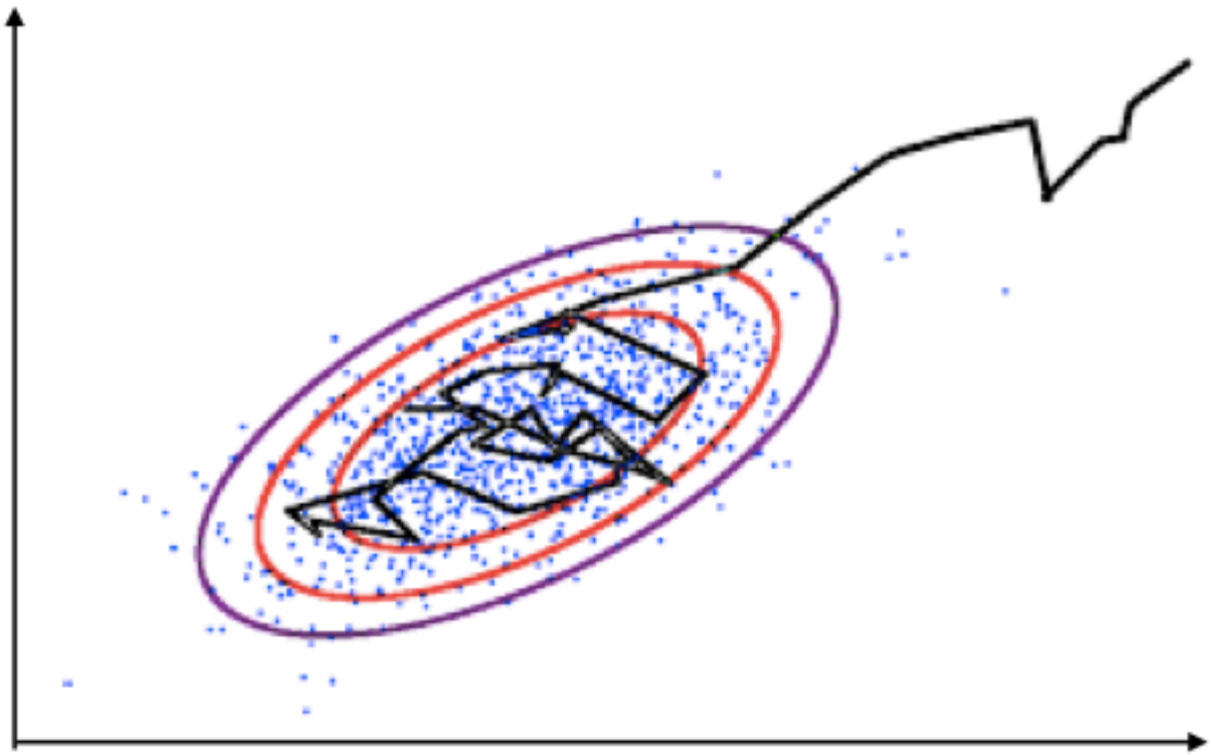
Monte Carlo Markow Chain (MCMC)

- Current position is x
- Propose a new position y = x + δ
- Move if Metropolis Hastings condition satisfied
- "Walks" around the parameter space to approximate posterior
- Moves are stochastic but walk tends to region of high probability





1D example



2D example

More advanced sampling

More advanced sampling approaches

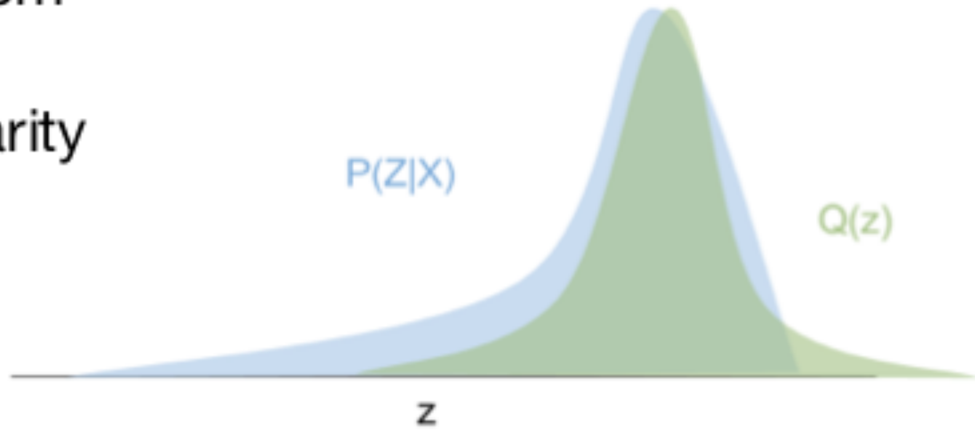
- Gibbs Sampling
- Nested sampling
- Hamiltonian Monte Carlo sampling
- NUTS Sampler
- Higher order (Riemannian Geometry) Based Sampling

Pros/cons with Sampling

- Guarantee asymptotically correct solutions
- Can be applied in principle to any model
- Can be very slow to explore large high dimensional data sets
- Can struggle with highly correlated or multi-modal parameter spaces
- Can be difficult to code up more advanced algorithms

Variational Inference

- Propose some simple q(z) ~ p(z|x) then minimise KL-divergence between them



- KL Divergence is a measure of similarity between distributions

- Not a distance metric

- KL divergence tells you how much information you lose by approximating one distribution with another

- $KL(q(z)||p(z|x)) = \int q(z)log(\frac{q(z)}{p(z|x)})dx$

- Can then use standard optimisation (similar to neural networks) techniques to minimise this

Pros\cons with VI

- Much Faster than sampling
- Scale to high dimensional datasets
- No guarantee of convergence to global optimum
- Also in general difficult to code up

Probabilistic Programming

- STAN, Pyro, Tensorfow Probability, PyMC, Bilby …
- Attempts to make the computer science/statistics of doing science easy
- User friendly API which allows scientists to perform bayesian inference without coding difficult algorithms
- Similar to Tensorflow, Pytorch for neural networks

Thanks for listening
Any questions ?