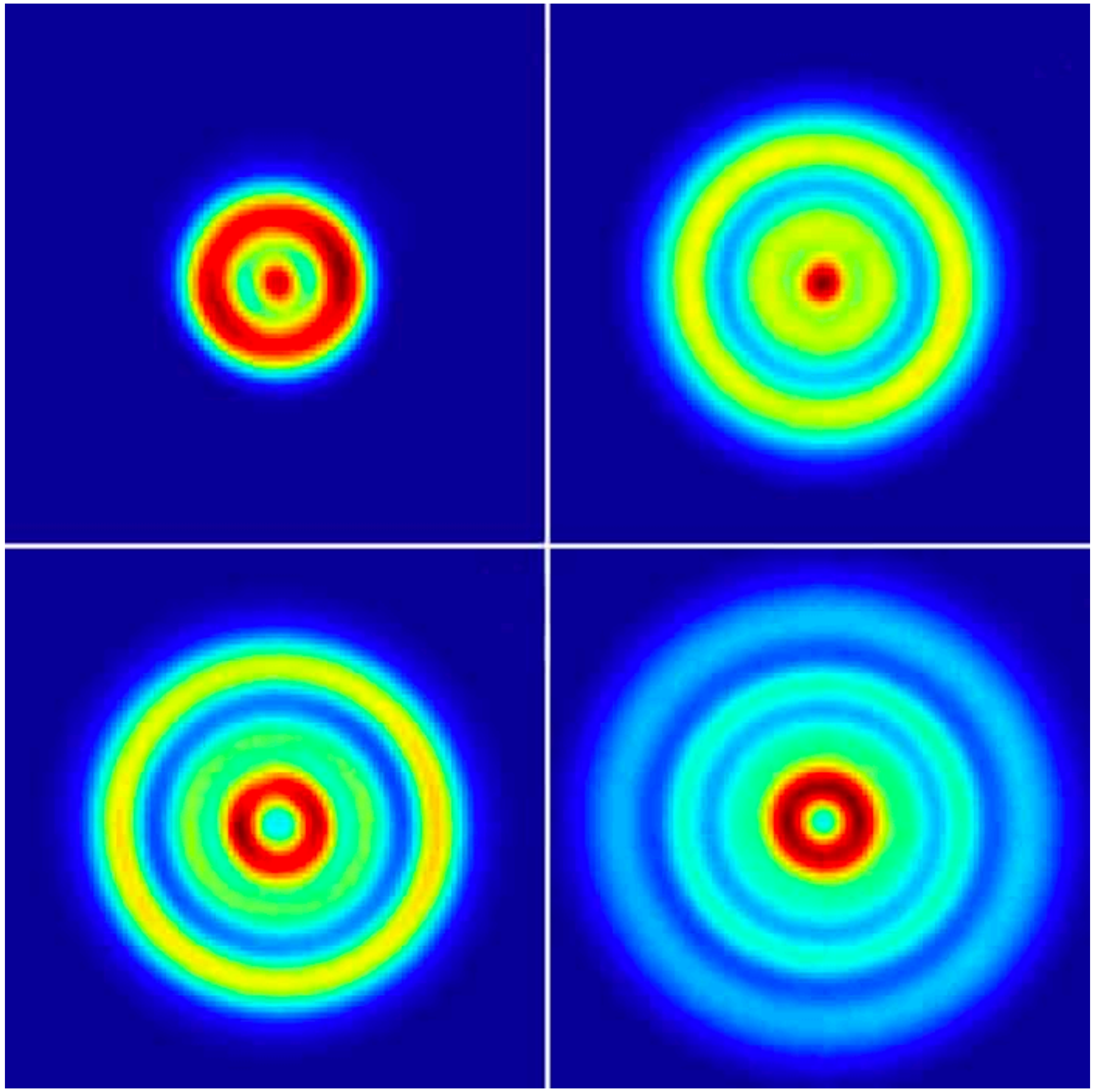
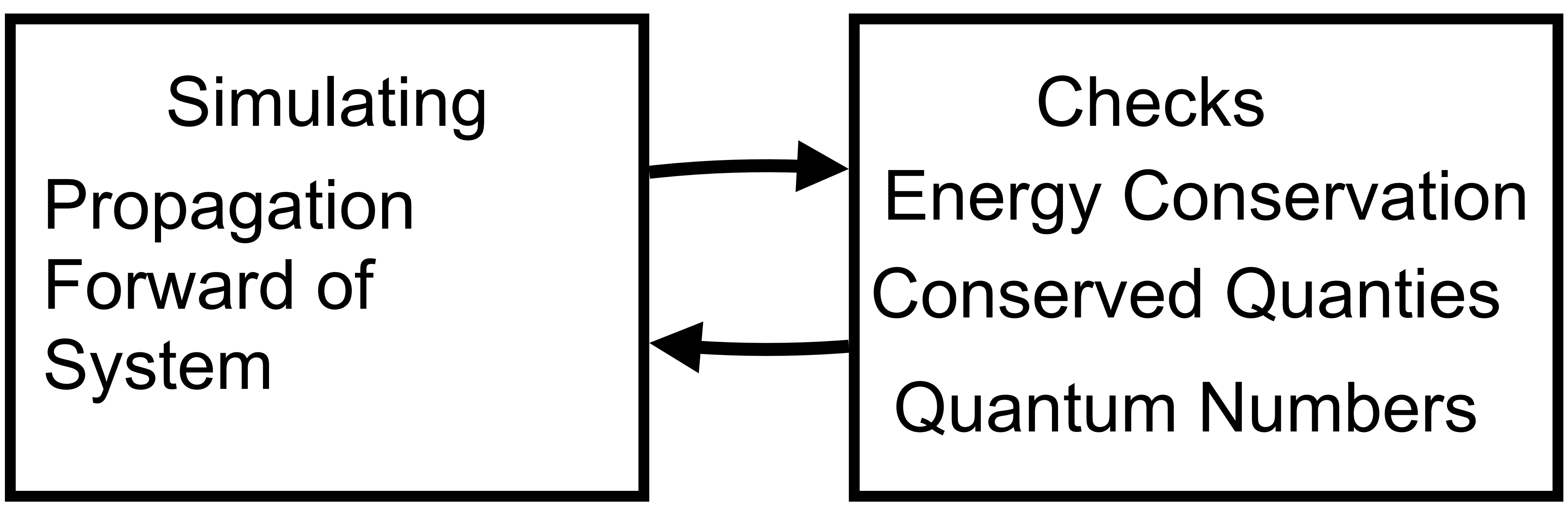


Lecture 22: Markov Chain Monte Carlo + More

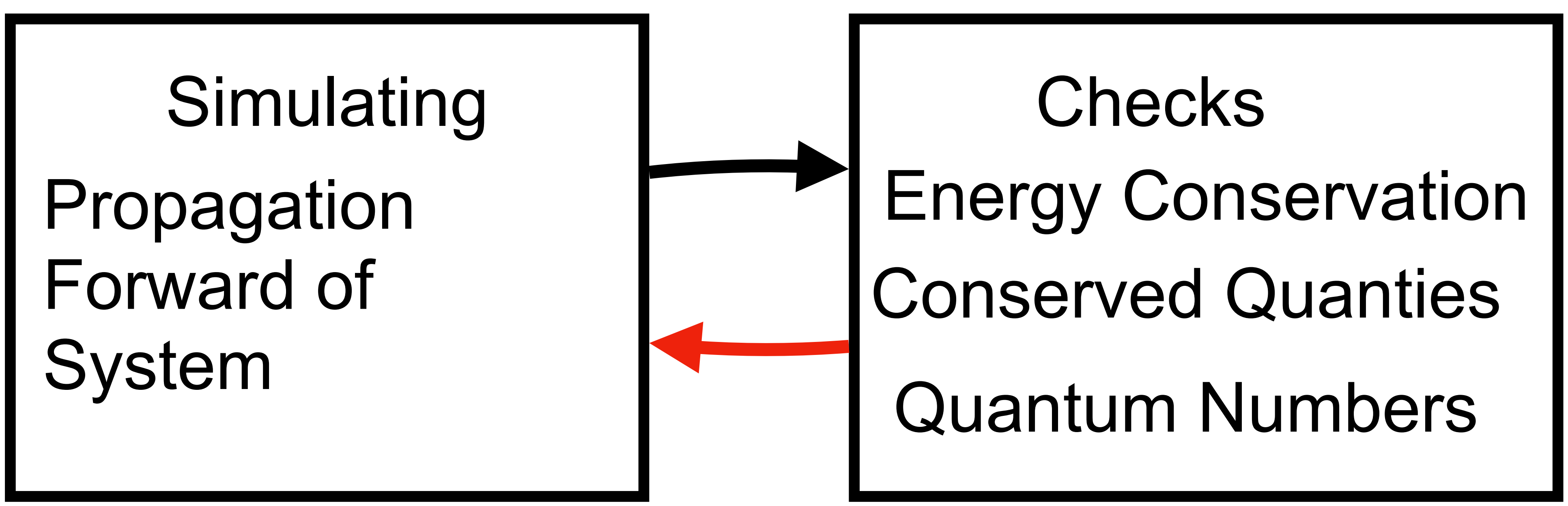
Making MC Better

- In this simulation part of this class
  - We have learned that simulations are not accurate
  - There are a few things we can do to make it better
    - Key is to have a notion of when we are going wrong



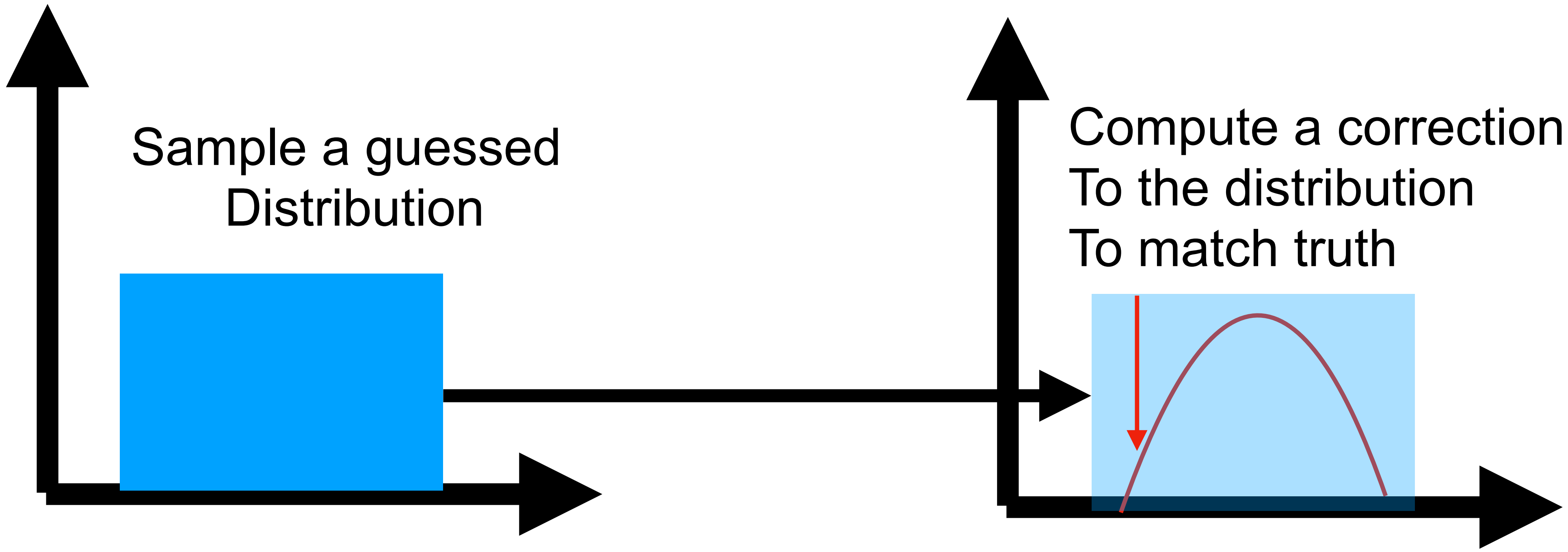
Correcting

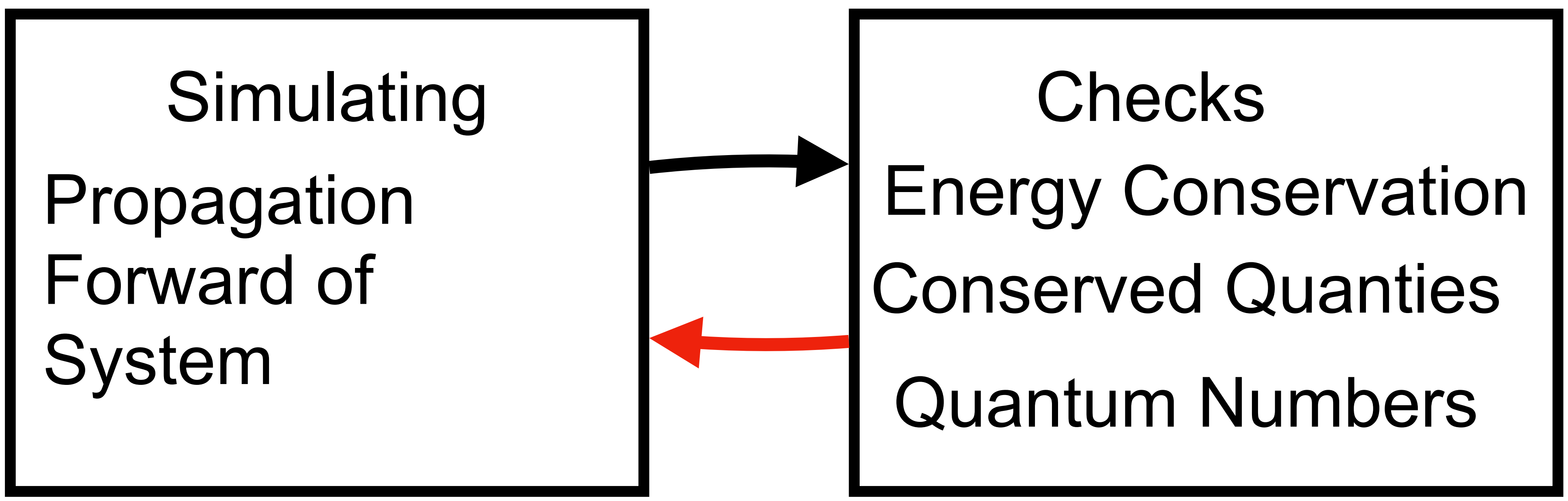
- In this simulation part of this class
    - We have learned that simulations are not accurate
    - There are a few things we can do to make it better
        - Key is to have a notion of when we are going wrong



Correct Our Simulation Through a Probabilistic Rescaling

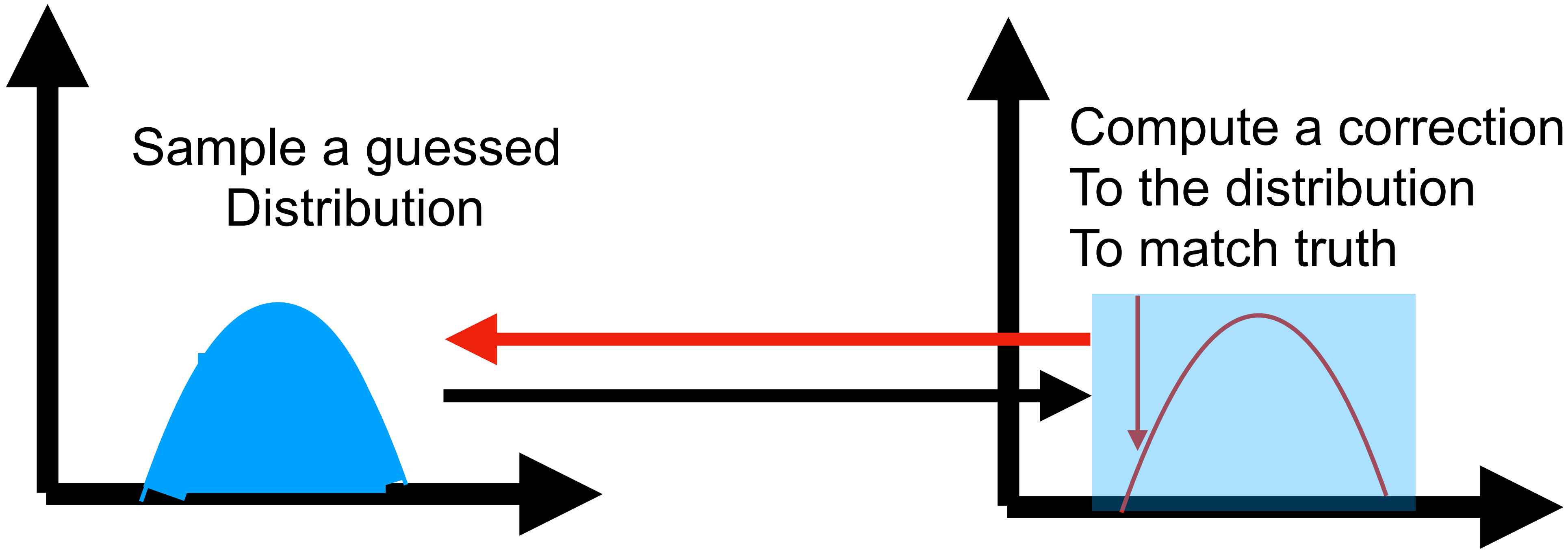
Markov Chain MC

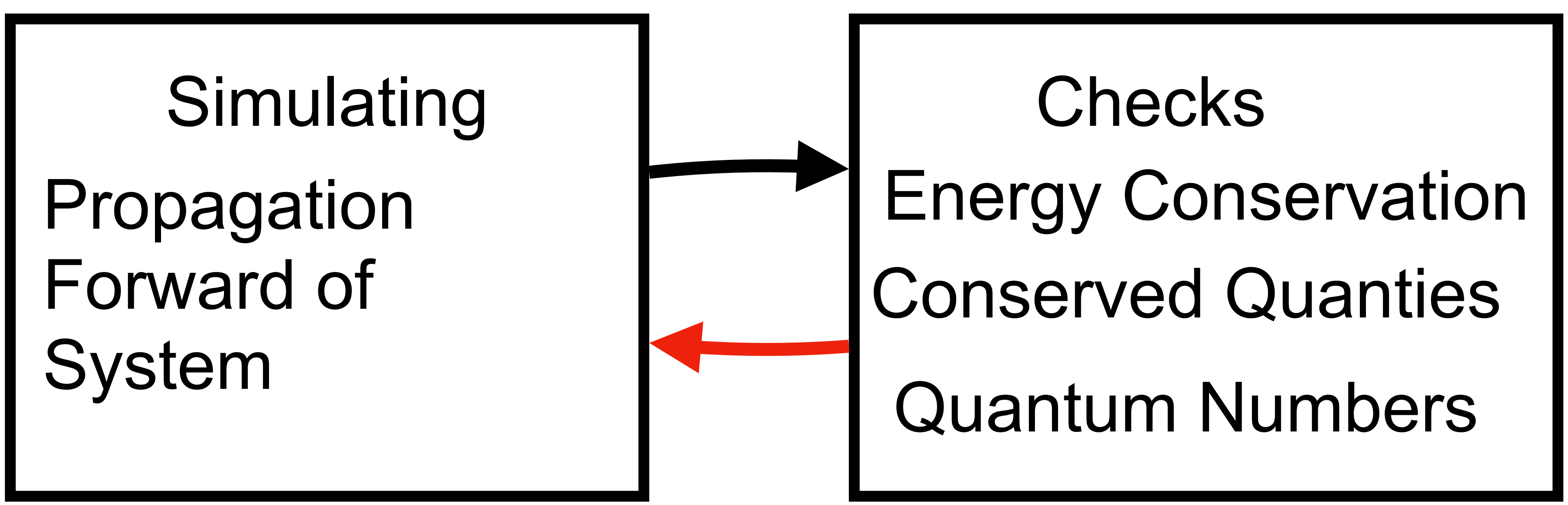




Correct Our Simulation Through a Probabilistic Rescaling

Markov Chain MC





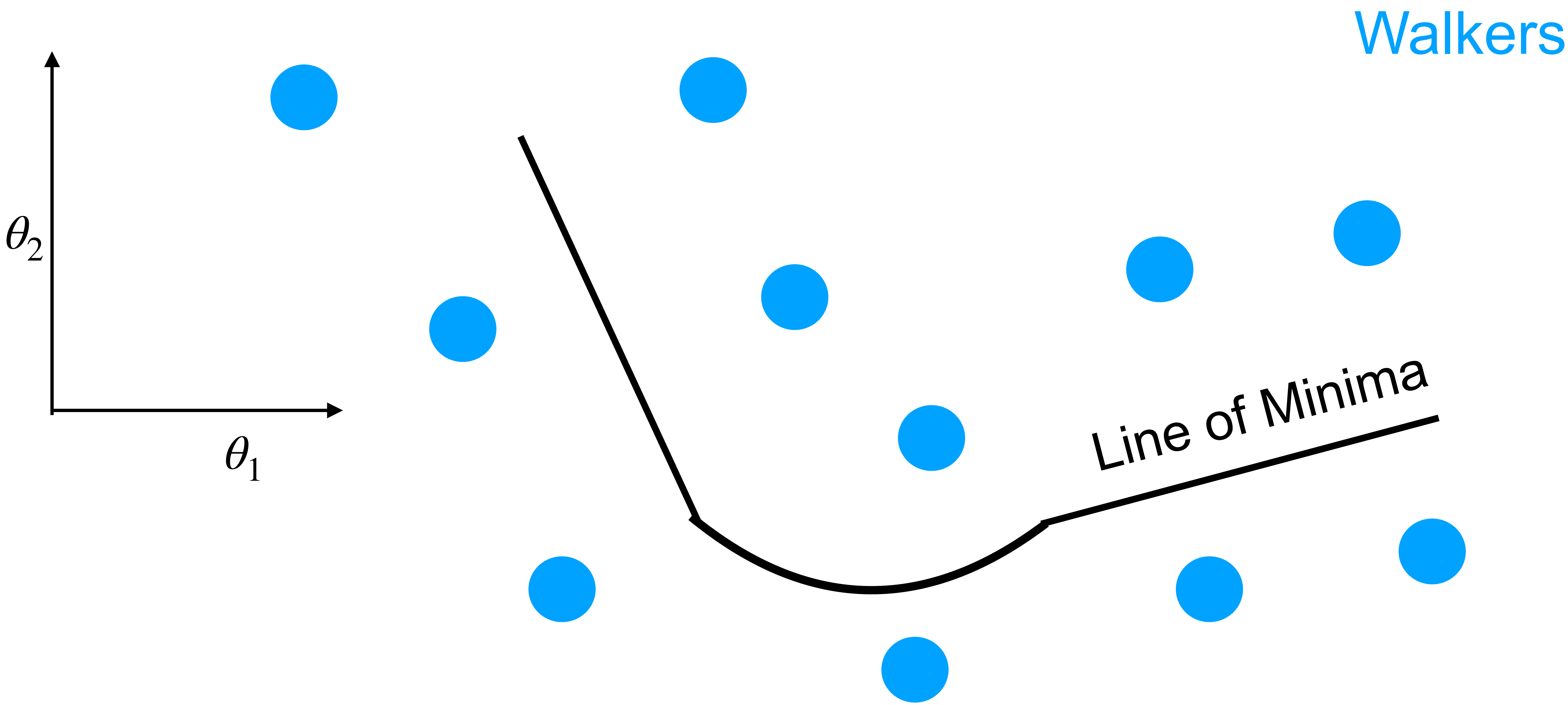
Correct Our Simulation Through a Probabilistic Rescaling

Metropolis-Hastings

- Step 0: Randomly sample a parameter $\boldsymbol{x_1}$
- Step 1: Sample a new parameter $\mathbf{x_2}$
  - Use a chosen "Proposal Function" for sampling
  - Compute the probability of stepping $\mathbf{x_2}$ to stepping $\mathbf{x_1}$
- Step 2: Sample a flat distribution from 0 to 1 ($\mathbf{s_2}$)
  - Accept $\mathbf{x_2}$ if $s_2 < \dfrac{p(x_2)}{p(x_1)}$
- Step 3 : Go back to step 1

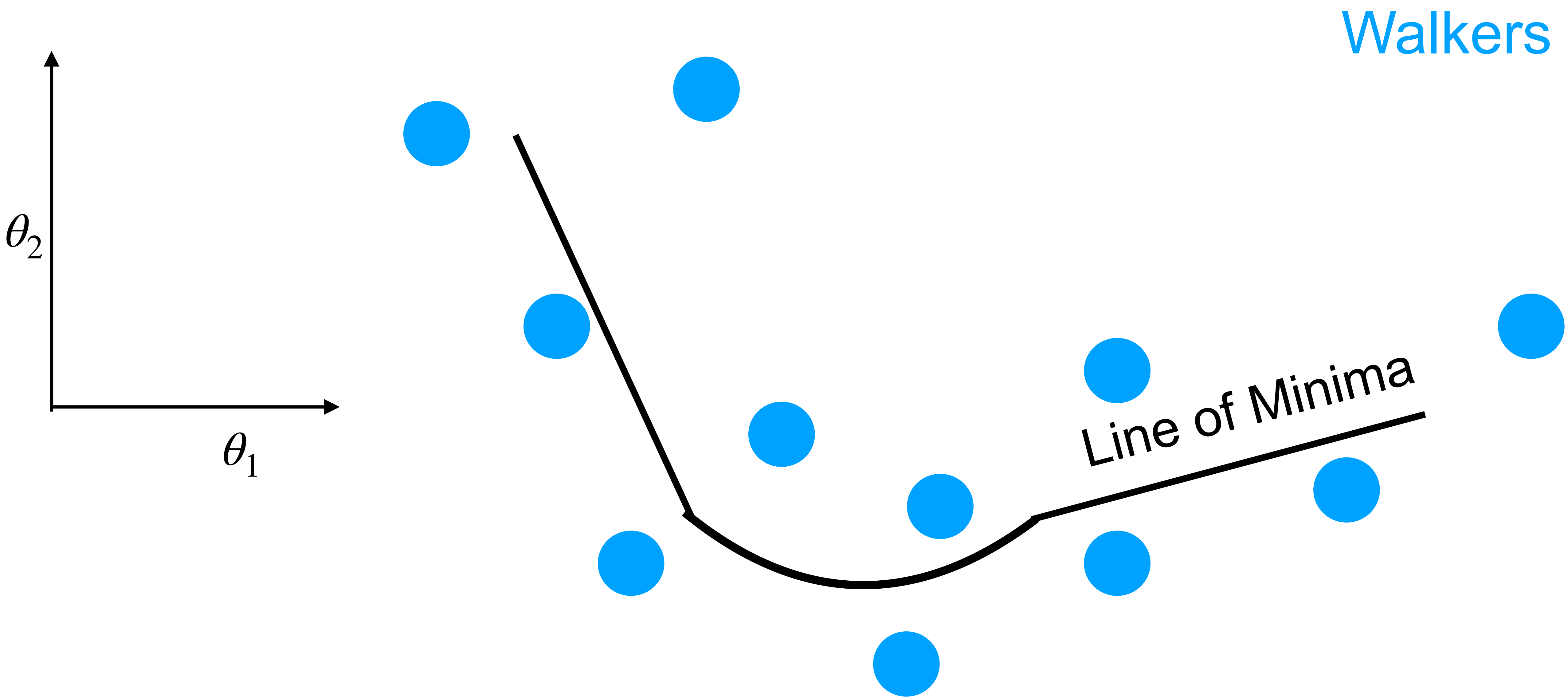
Speeding up MCMC

- We can consider having many walkers probe our space
  - Many walkers at the same time speed up convergence



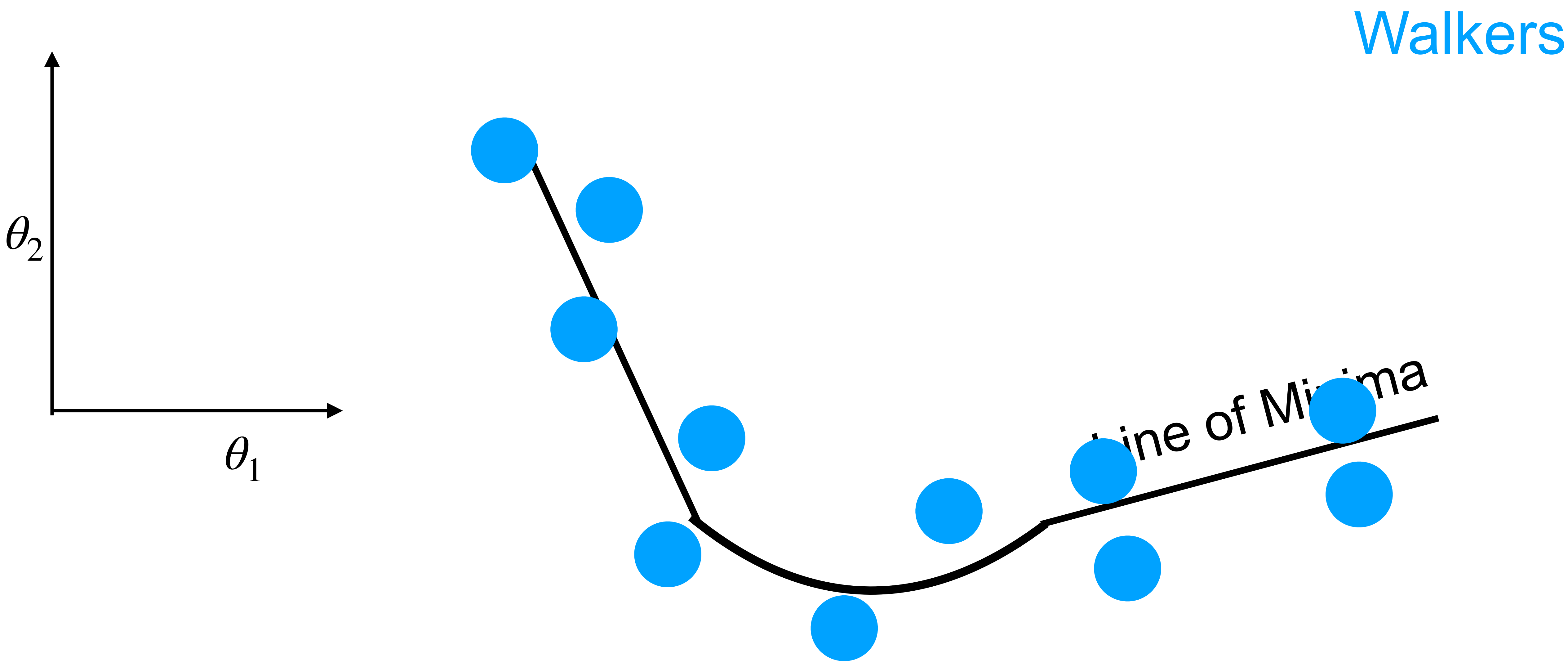
Speeding up MCMC

- We can consider having many walkers probe our space
  - Many walkers at the same time speed up convergence

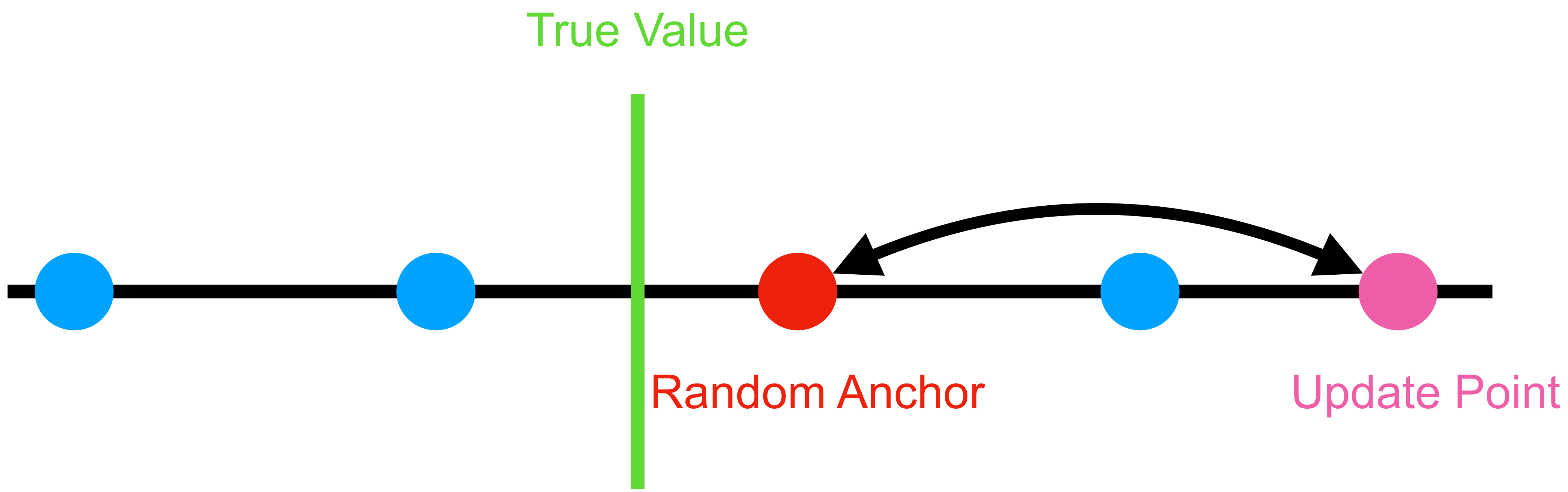


Speeding up MCMC

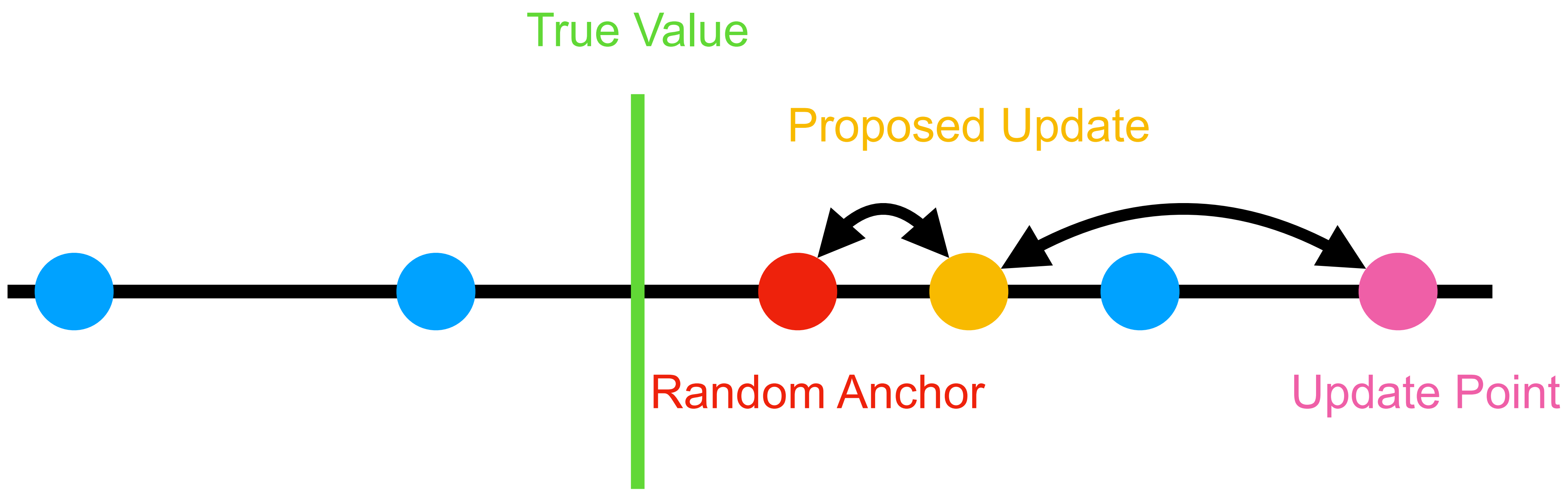
- We can consider having many walkers probe our space
  - Many walkers at the same time speed up convergence



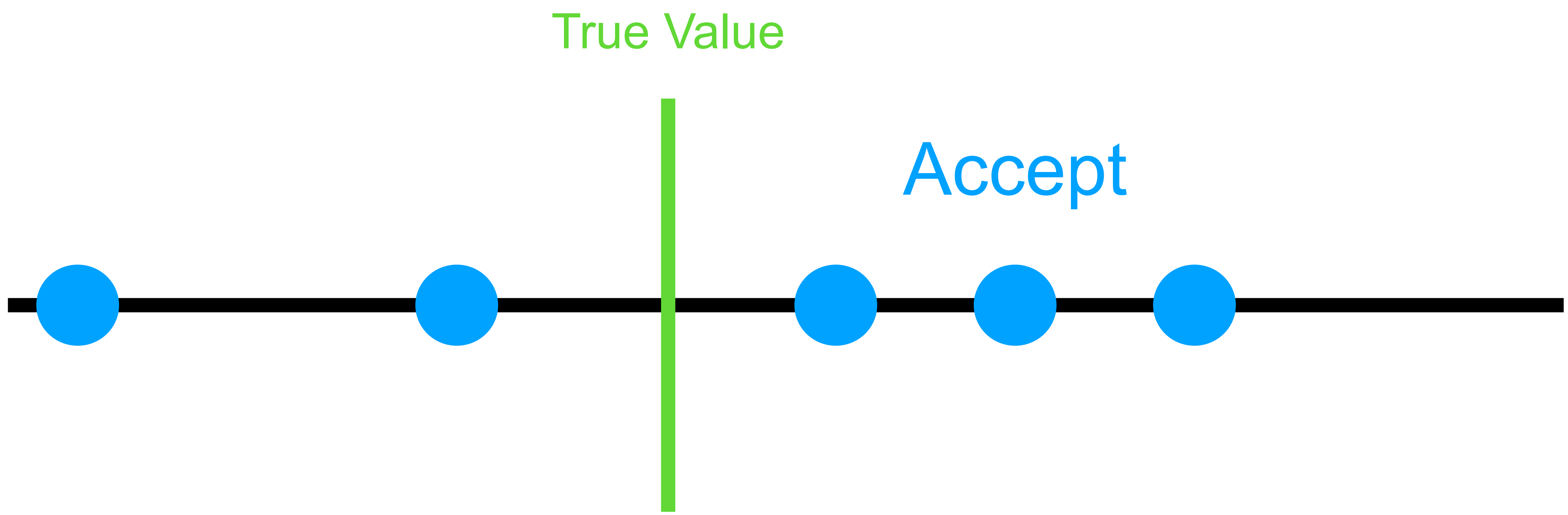
Example Fit



Example Fit

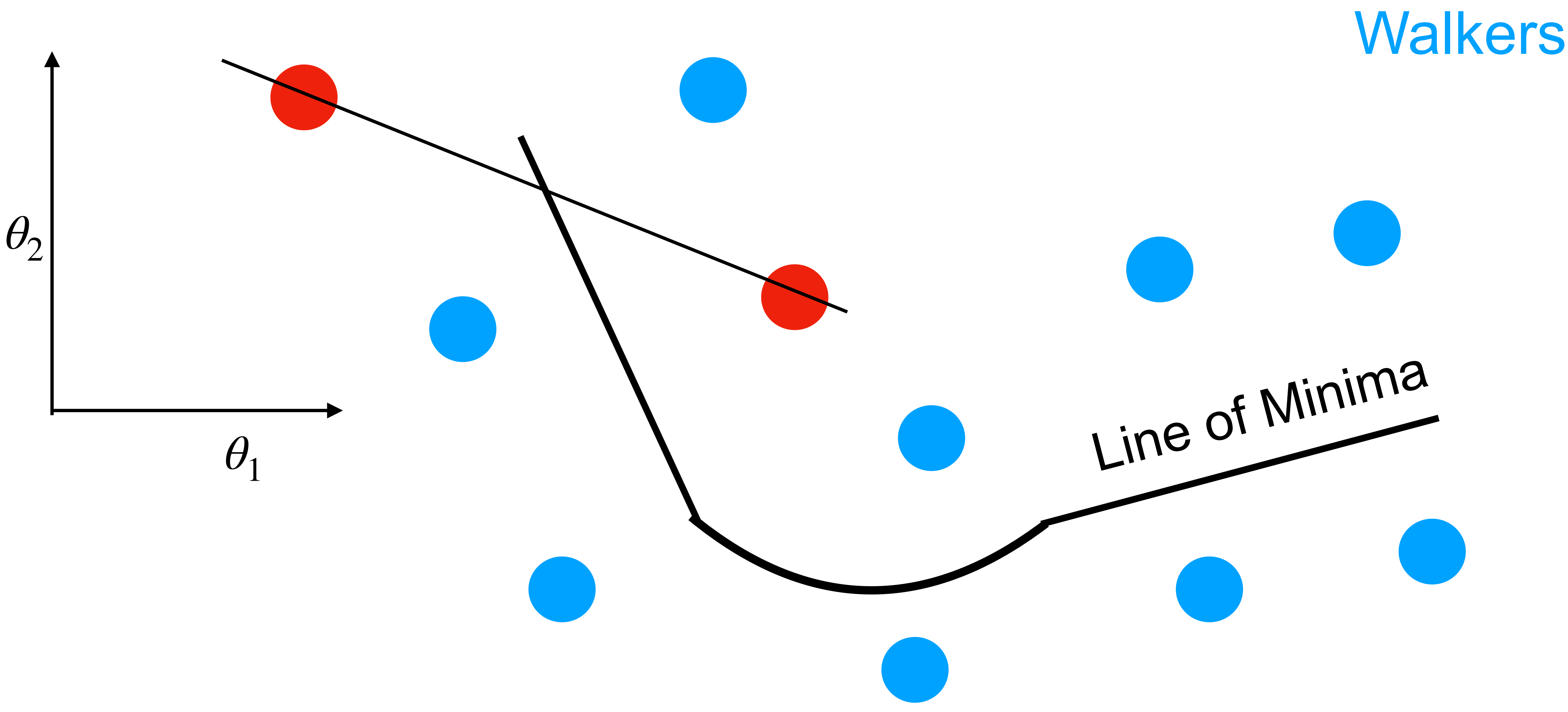


Example Fit



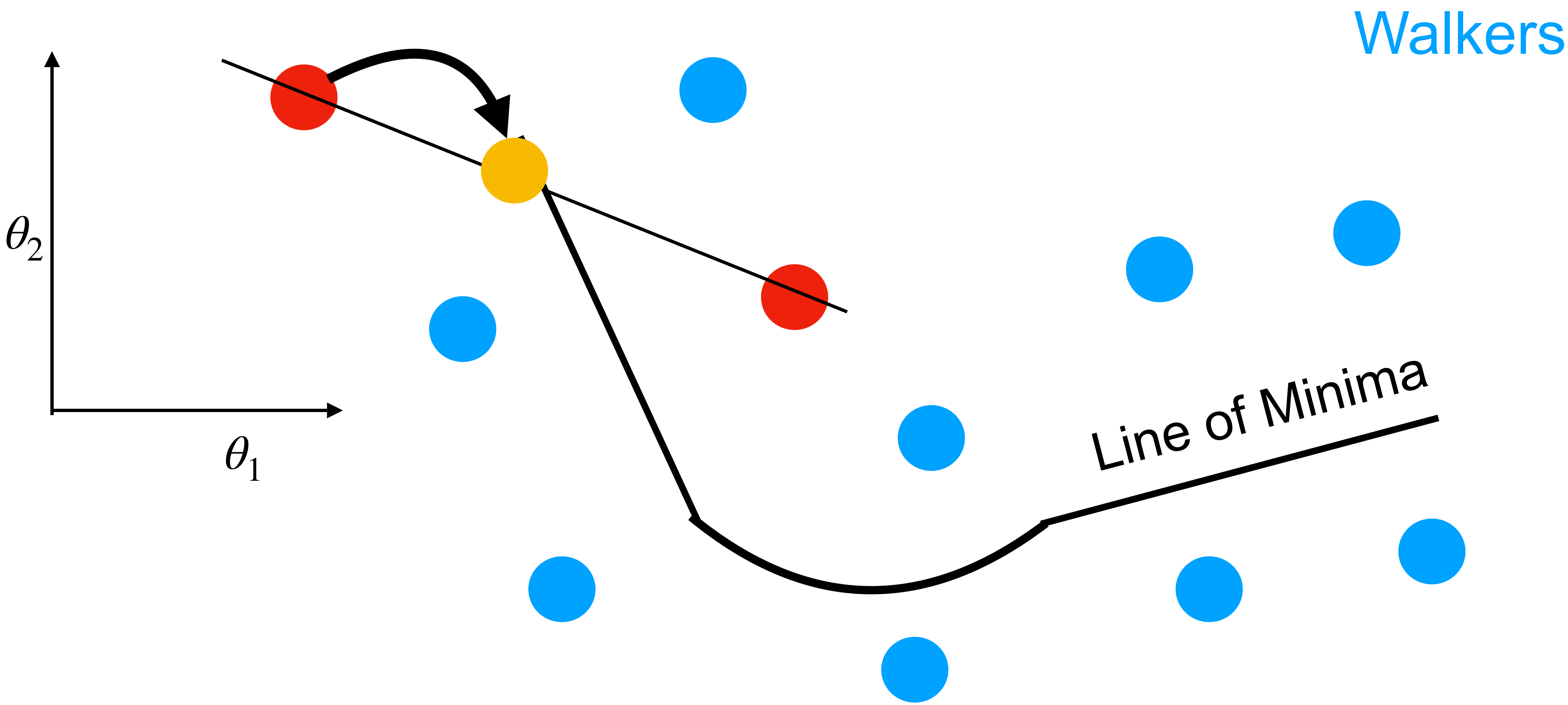
Updating w/Random Points

- We randomly choose a pair of points
  - Move one of the points along the line between them



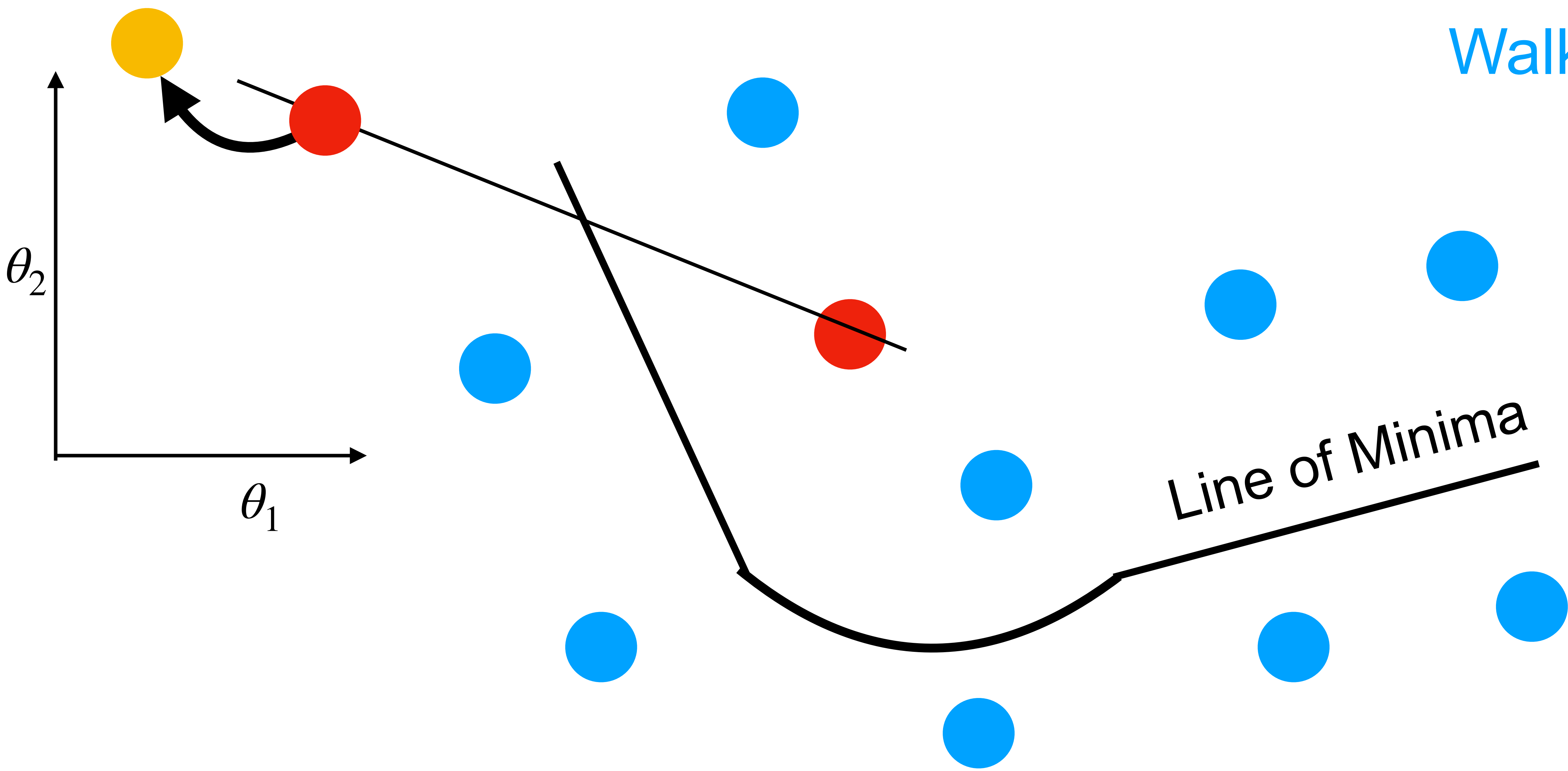
Updating w/Random Points

- We randomly choose a pair of points
  - Move one of the points along the line between them



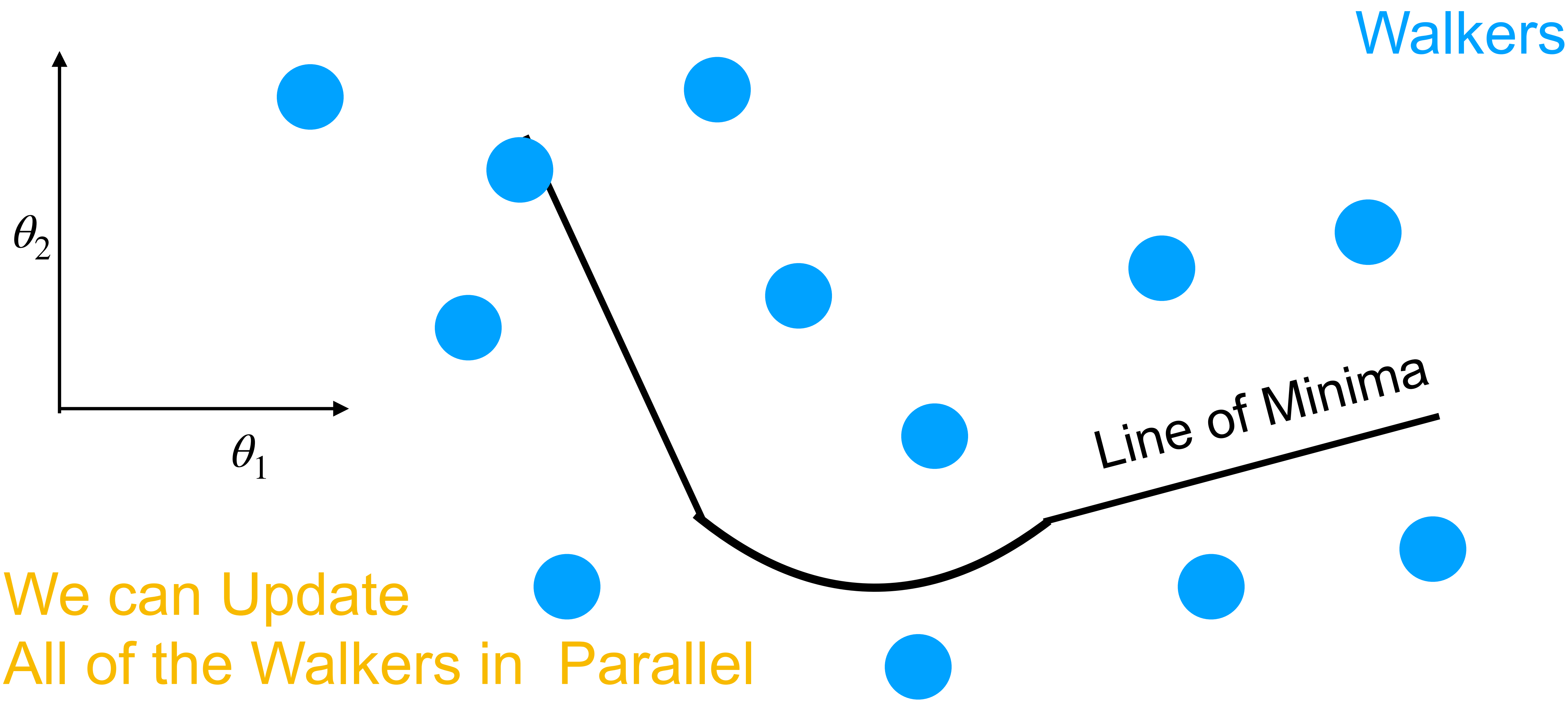
Updating w/Random Points

- We randomly choose a pair of points
  - Move one of the points along the line between them



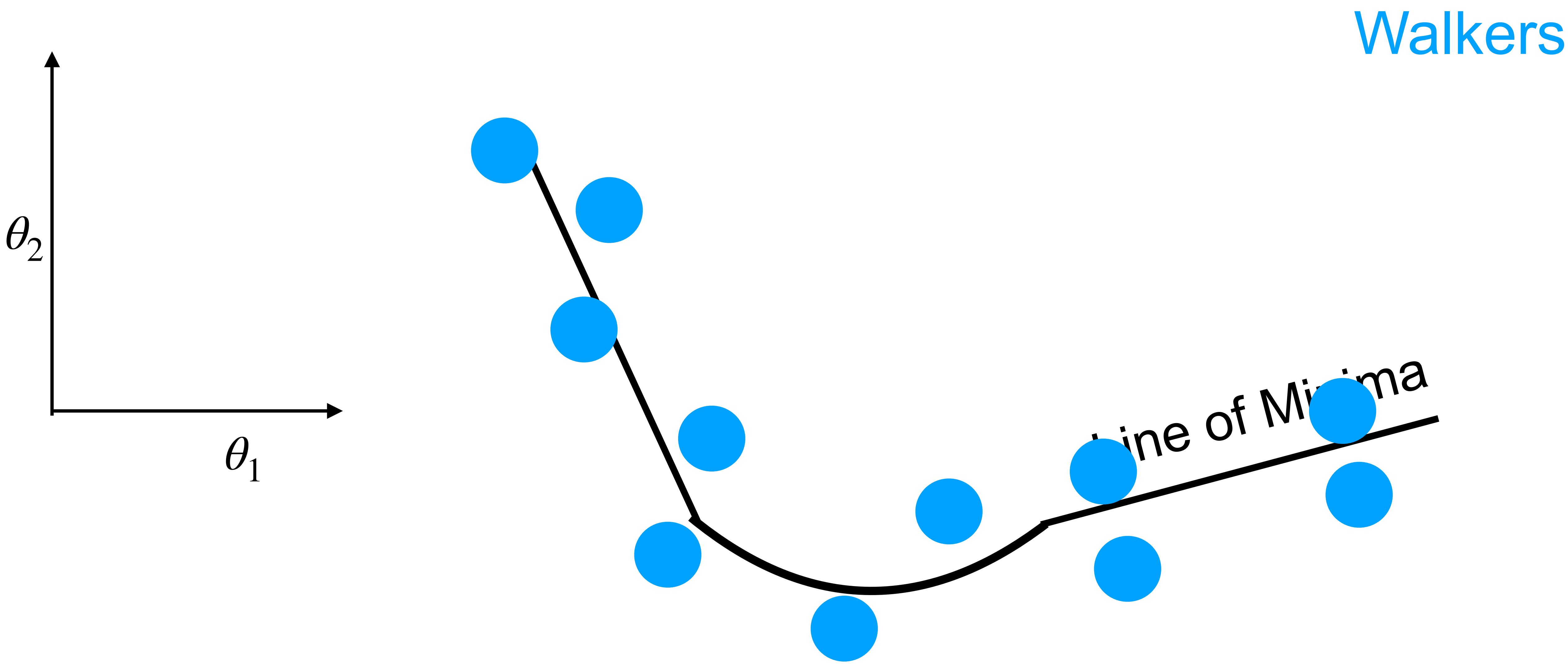
Updating w/Random Points

- We randomly choose a pair of points
  - Move one of the points along the line between them



Speeding up MCMC

- We randomly choose a pair of points
  - Move one of the points along the line between them



Visualizing in Bayes

Posterior probability of the model in light of the data

Likelihood function of the data given the model

Prior probability of the model

$$\boxed{P(\text{model}|\text{data})} = \frac{\boxed{P(\text{data}|\text{model})}\,\boxed{P(\text{model})}}{P(\text{data})}$$

Equilibrated Best Fit

Sampled Distribution
Aka Proposal

Probability Driving
Decision

Evidence [can typically be absorbed into the normalization of the posterior]

The Likelihood reweights the Prior to the Posterior

Quantum Monte Carlo

- Can use the same MCMC to populate a wave function
    - We can then scan paramaters to solve Shroedinger's Eq

$$\psi(\vec{r}\,|\,\vec{\theta}) = Ae^{-r/\theta_0}$$

Guess a Form for the wavefunction

$$p(\vec{r}\,|\,\vec{\theta}) = \frac{\psi^*(\vec{r}\,|\,\vec{\theta})\psi(\vec{r}\,|\,\vec{\theta})}{\langle\psi\,|\,\psi\rangle}$$

We can define probability from wavefunction

$$w_{i+1} = \frac{p(\overrightarrow{r_{i+1}}\,|\,\vec{\theta})}{p(\overrightarrow{r_i}\,|\,\vec{\theta})} = \frac{\psi^*(\vec{r}_{i+1})\psi(\vec{r}_{i+1})}{\psi^*(\vec{r}_i)\psi(\vec{r}_i)}$$

Our proposal Doesn't need integral Aka $\langle\psi\,|\,\psi\rangle$

Multiple Walkers Populate

- The key is to MCMC evolve the wave function many times
  - We can use the aggregate Particles solve QM stuff

$$\sum_j \psi_j(\vec{r}\,|\,\vec{\theta}) = Ae^{-r/\theta_0}$$

Guess a Form for the wavefunction

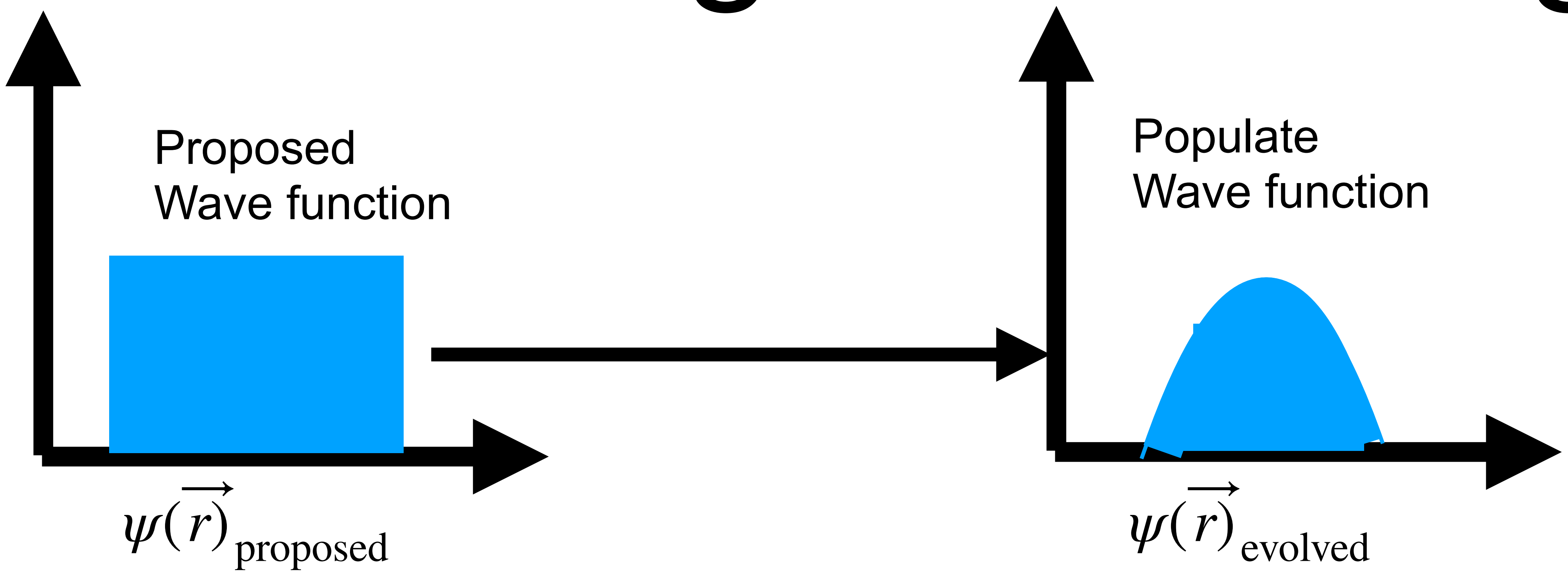
$$\sum_j p_j(\vec{r}\,|\,\vec{\theta}) = \frac{\psi_j^*(\vec{r}\,|\,\vec{\theta})\psi_j(\vec{r}\,|\,\vec{\theta})}{\langle\psi\,|\,\psi\rangle}$$

We can define probability from wavefunction

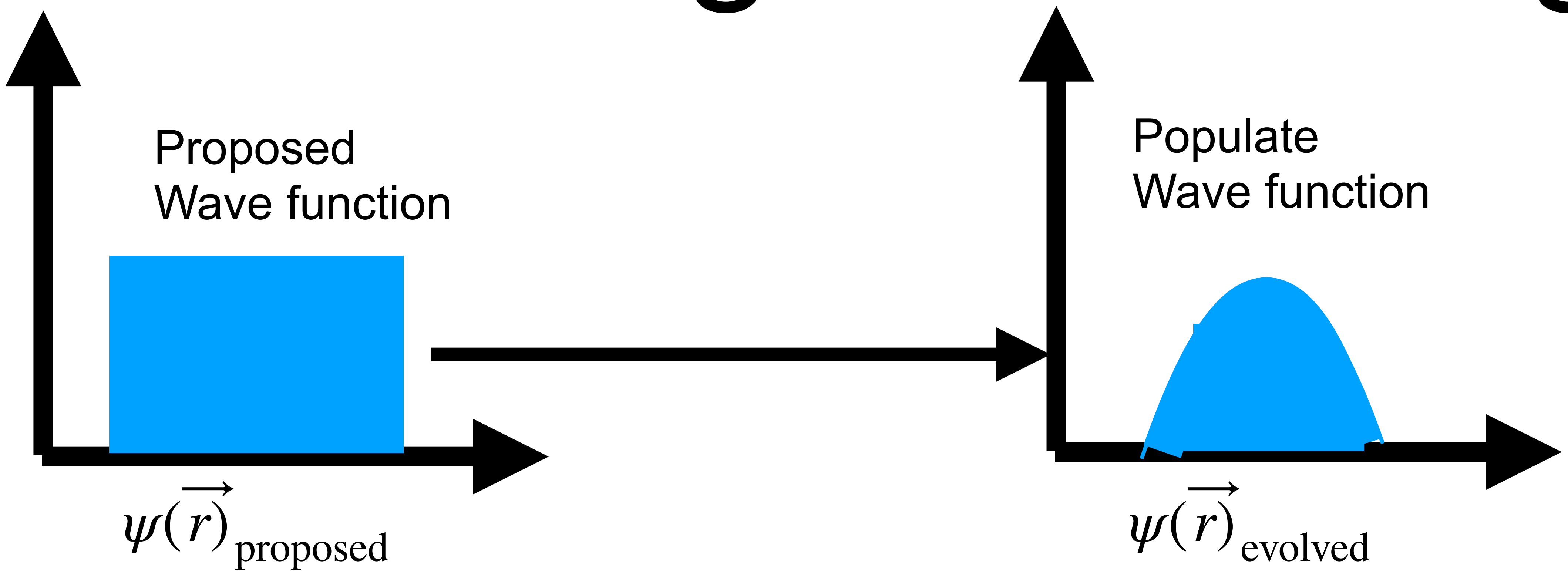
$$\sum_j w_{i+1}^j = \frac{p_j(\overrightarrow{r_{i+1}}\,|\,\vec{\theta})}{p_j(\overrightarrow{r_i}\,|\,\vec{\theta})}$$

Our proposal Doesn't need $\langle\psi\,|\,\psi\rangle$

Solving Schroedinger



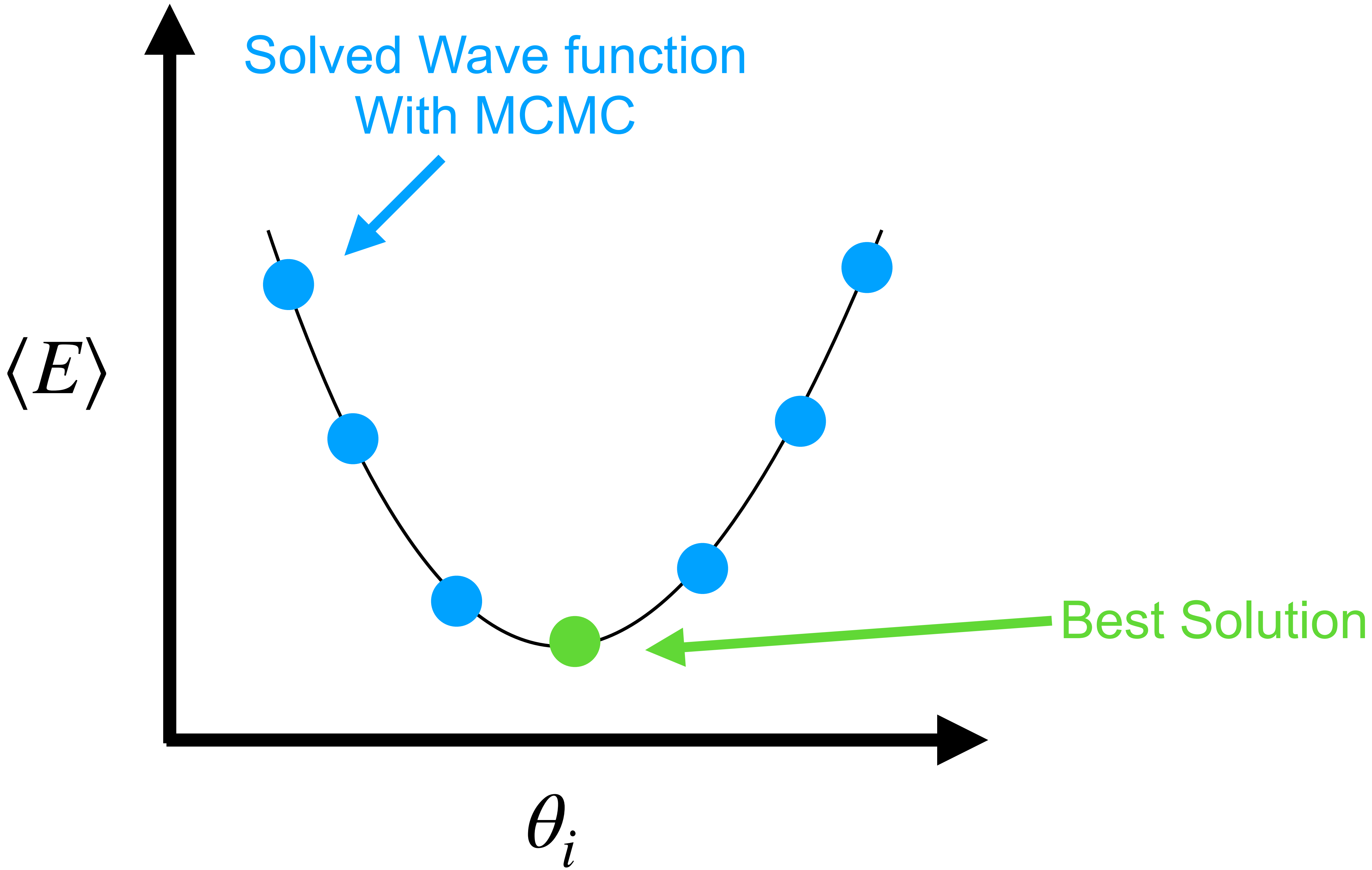
Solving Schroedinger



- Once we have the evolved wave funciton
  - We can compute expectations
  - No need to integrate (Really this is MC integration)

$$\boxed{\langle E\rangle = \sum_j p_j(\vec{r}\,|\,\vec{\theta})E_j(\vec{r}\,|\,\vec{\theta}) = \sum_j \psi_j^*(\vec{r}\,|\,\vec{\theta})\psi_j(\vec{r}\,|\,\vec{\theta})E_j(\vec{r}\,|\,\vec{\theta})}$$

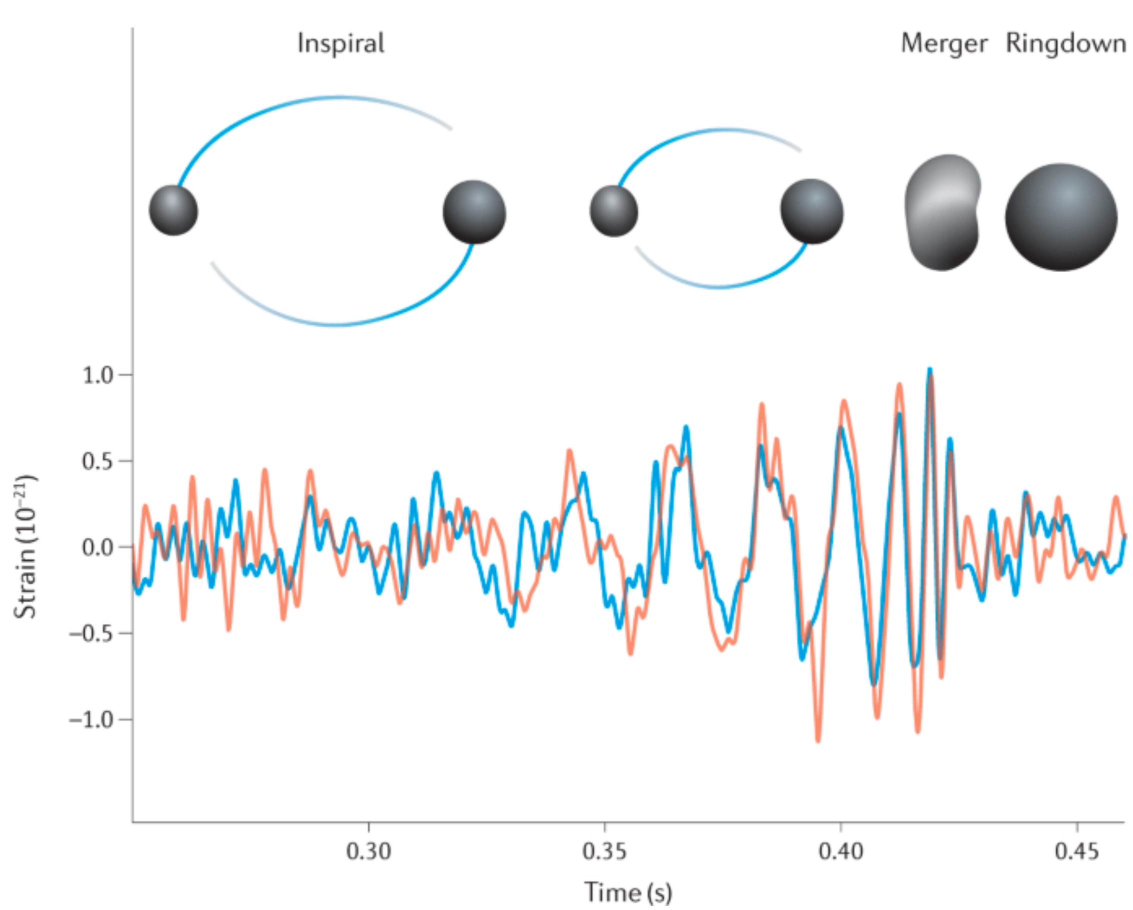
Solving Schroedinger

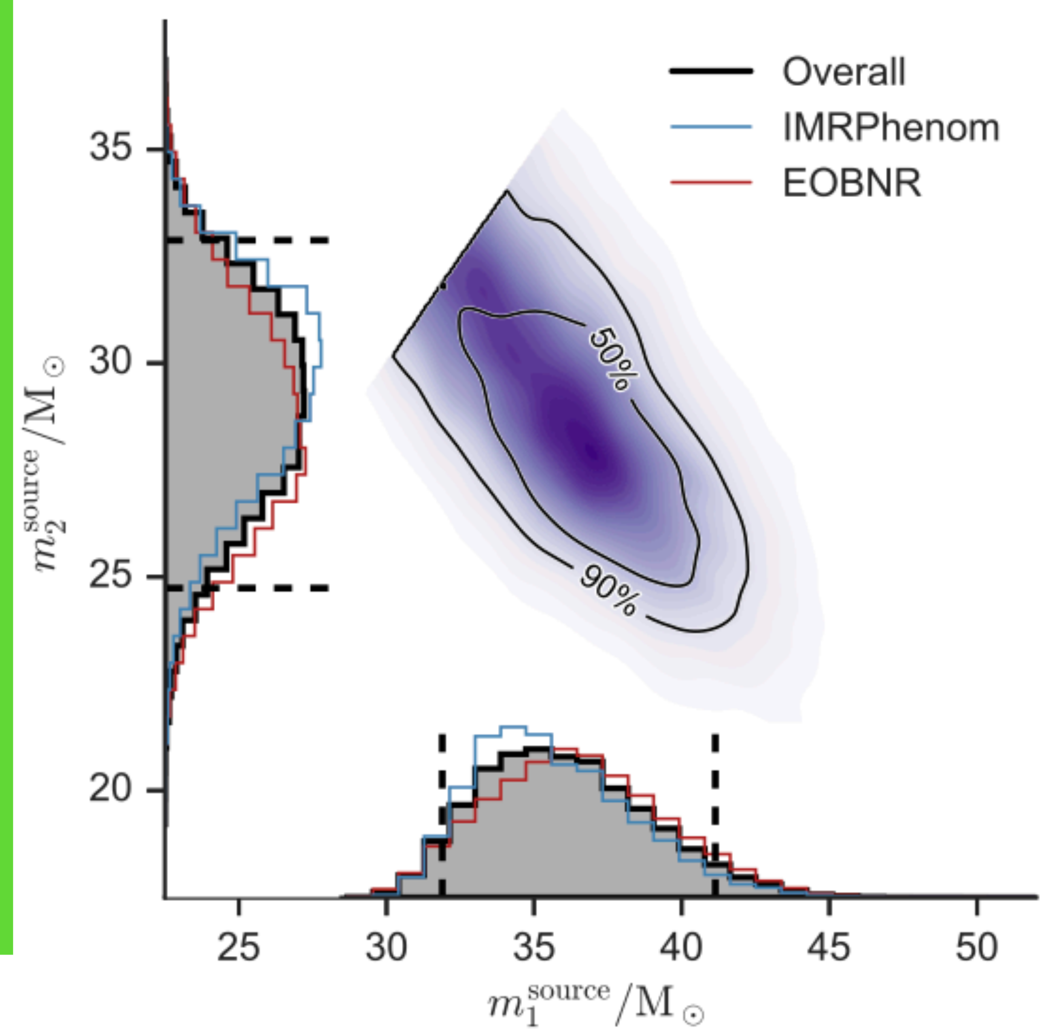


Our goal is to minimize the Energy given a wave functional form

Full Blow MCMC

- Ultimately the big gain from this are complex fits
  - Cases where normal gradient descent breaks down
  - What better case than to go back to LIGO





Full Waveform not differentiable (cannot fit it)

Observations

- There is some elegance to the MCMC approach
  - Builds directly to Bayesian fitting
  - MC allows us to explore parameters and correlations

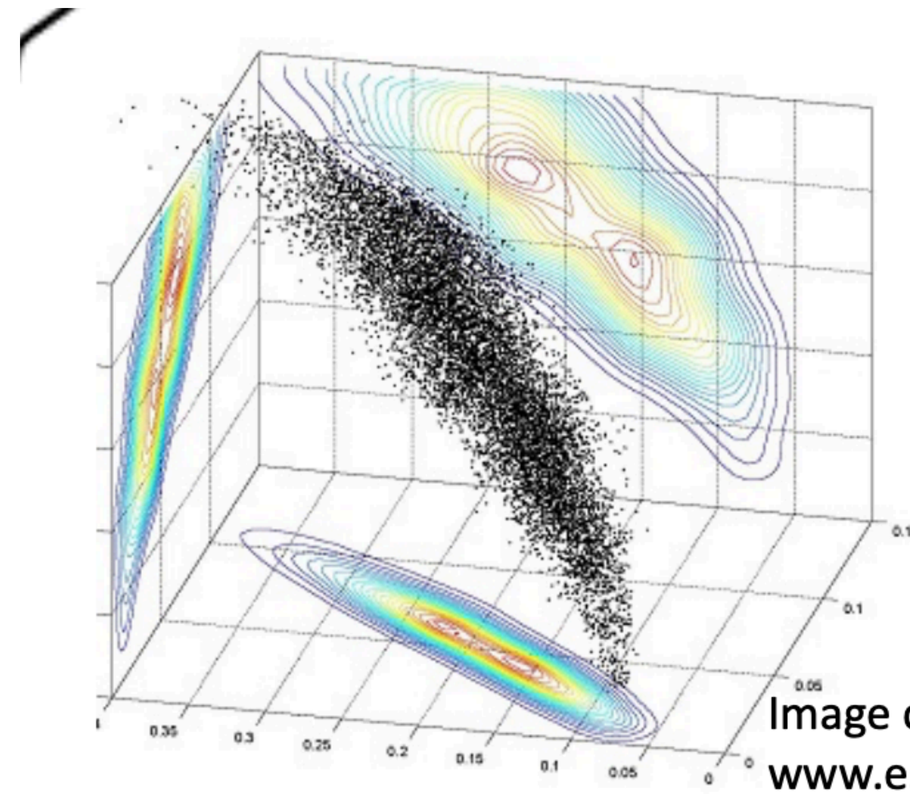


Image credit: www.essenceps.com

- However, it is really slow
  - Migration towards differentiable loss is becoming popular
  - Training an NN to replace part of sampling helps with this
  - Project 3 starts to illustrate modern approach to this all

Visualizing in Bayes

Posterior probability of the model in light of the data

Likelihood function of the data given the model

Prior probability of the model

$$\boxed{P(\text{model}|\text{data})} = \frac{\boxed{P(\text{data}|\text{model})}\,\boxed{P(\text{model})}}{P(\text{data})}$$

Equilibrated Best Fit

Sampled Distribution
Aka Proposal

Probability Driving
Decision

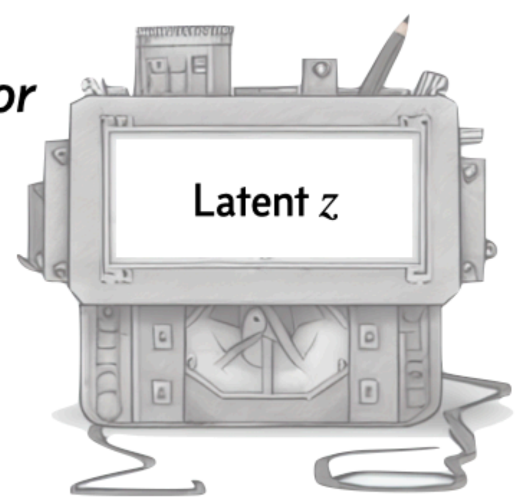
Evidence [can typically be absorbed into the normalization of the posterior]

The Likelihood reweights the Prior to the Posterior

Simulation Based Inference

Simulation-based inference (SBI)

Simulator

Parameters θ →  → Observations x

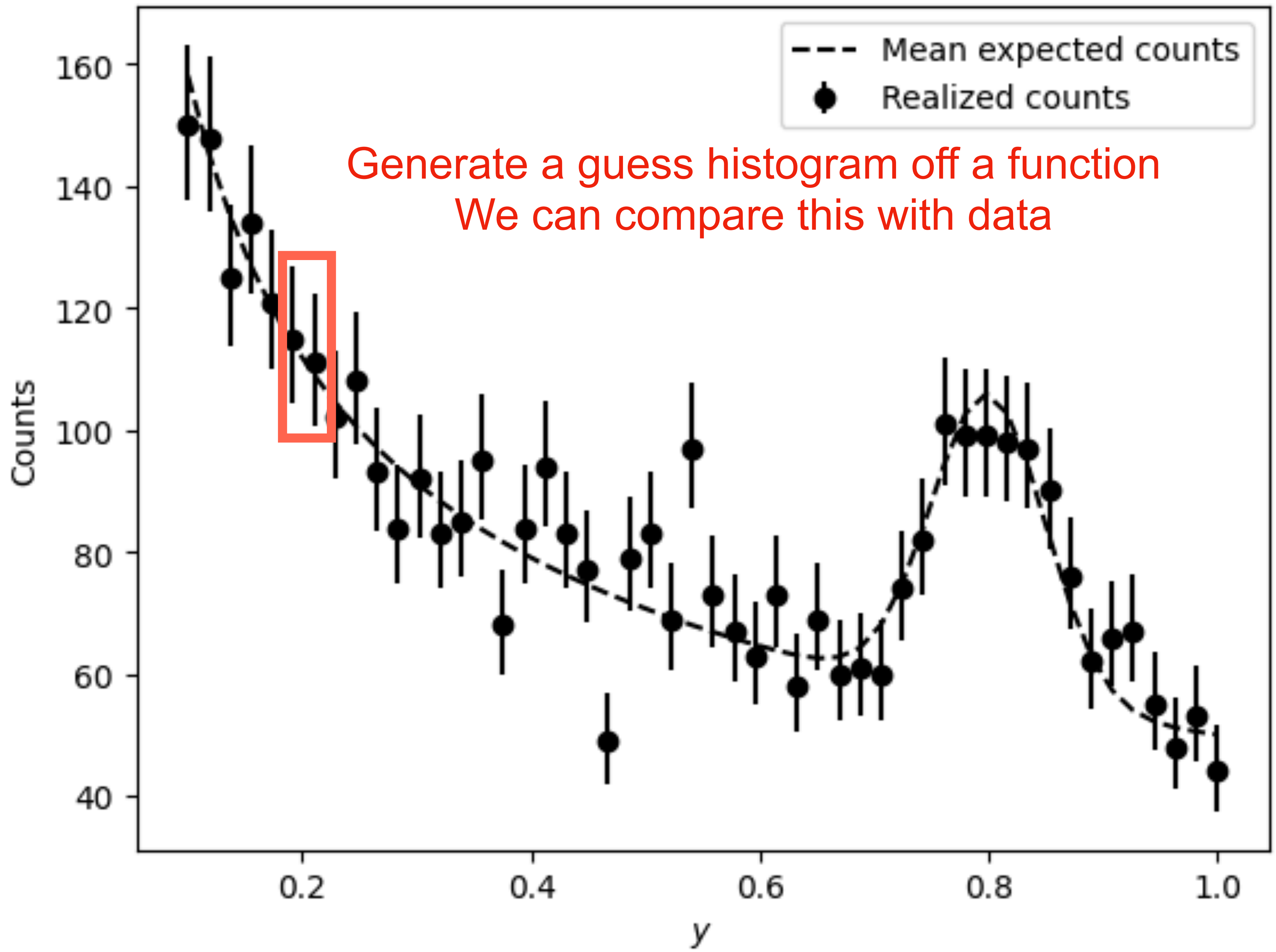
Prediction:

- Mechanistic forward model
- We can generate samples from a simulator $x \sim p(x \mid \theta)$

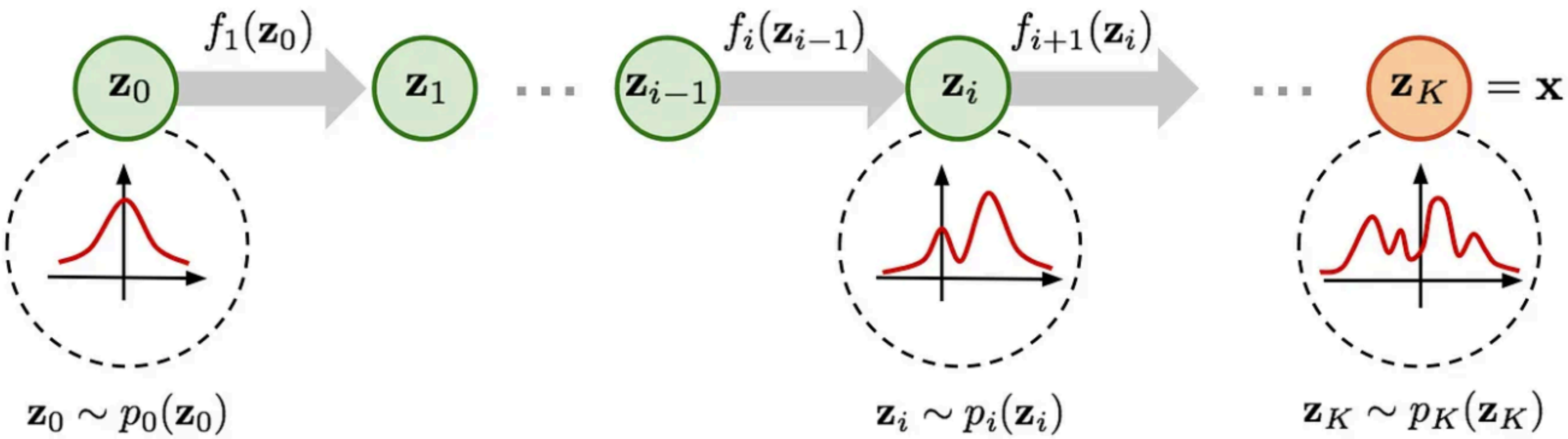
Inference:

- Likelihood $p(x \mid \theta) = \int \mathrm{d}z \, p(x, z \mid \theta)$ is intractable
- *Inference is challenging*

Poisson Fluctuation



Normalizing Flow



$$z \sim p_\theta(z) = N(z; 0, 1)$$

$$x = f_\theta(z) = f_K \circ \ldots f_2 \circ f_1(z)$$

each f_i is invertible

How Does it Work?

$$f : Z \to X,\ f\, is\, invertible$$

$$p_\theta(z)\, defined\, over\, z \in Z$$

$$p_\theta(x) \neq p_\theta(f_\theta^{-1}(x)))$$

Change of variable formula:

$$p_\theta(x) = p_\theta(f_\theta^{-1}(x))) \left| det(\frac{\partial f^{-1}(x))}{\partial x}) \right|$$

$$p_\theta(x) = p_\theta(z)) \left| det(\frac{\partial z}{\partial x}) \right|$$

$$\log\left(p_\theta(x)\right) = \log\left(p_\theta(z)\right) \ + \ \sum_{i=1}^{K} \log \left| det(\frac{\partial f_i^{-1}}{\partial z_i}) \right|$$

exact likelihood evalutation

Poisson flucutate a func

“Traditional” SBI: *Approximate Bayesian Computation*

[Rubin 1984]