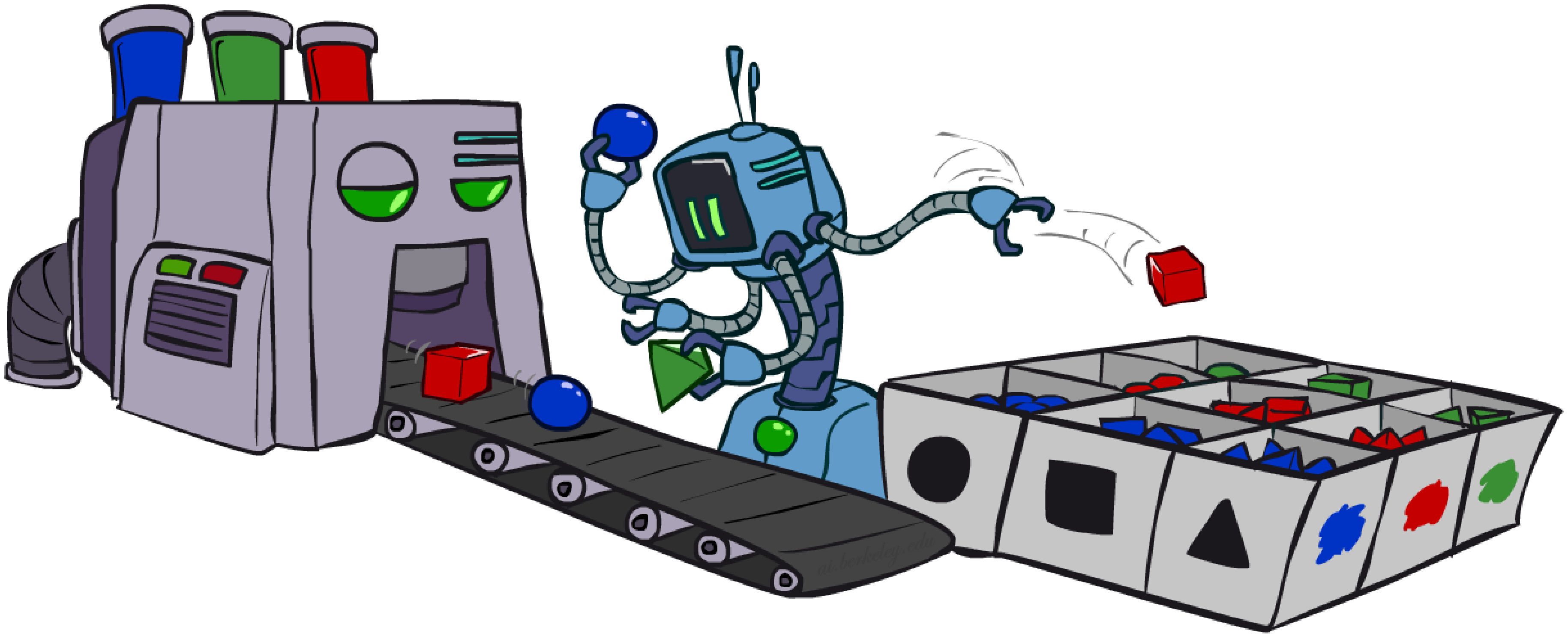
Announcement

- Homework 3
  - Due: Nov. 5, 11:59pm
- Programming Assignment 3
  - Due: Nov. 17, 11:59pm (2wk+5d!)

Bayes Nets: Approximate Inference



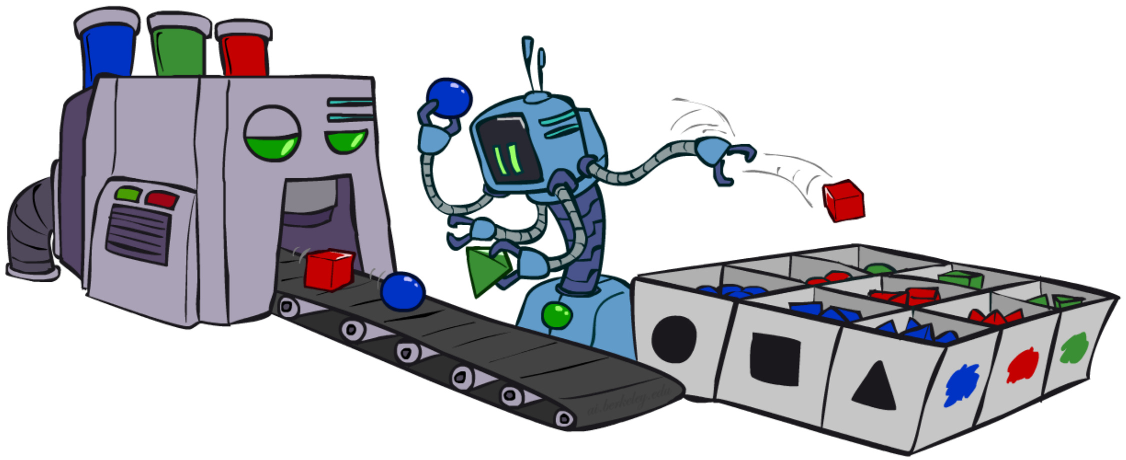
AIMA Chapter 14.5, PRML Chapter 11

[Adapted from slides by Dan Klein and Pieter Abbeel at UC Berkeley]

Approximate Inference by Sampling

- Goal: probability *P*
- Basic idea
  - Draw *N* samples from a sampling distribution *S*
  - Compute some quantity from the samples
  - Show this converges to the true probability *P*

- Why sample?
  - Often very fast to get a decent approximate answer
  - The algorithms are very simple and general (easy to apply to fancy models)
  - They require very little memory (O(n))



Sampling from a discrete distribution

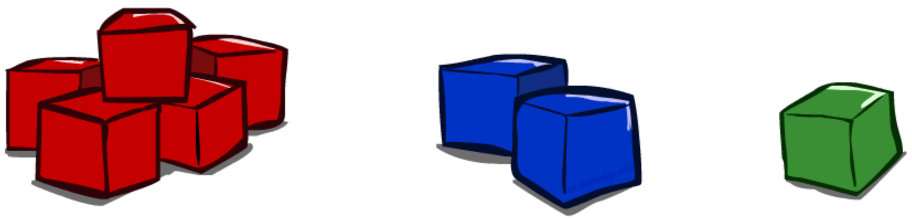
- Sampling from given distribution
  - Step 1: Get sample u from uniform distribution over [0, 1)
    - Random() in many programing languages
  - Step 2: Convert this sample u into an outcome for the given distribution by associating each outcome x with a $P(x)$-sized sub-interval of [0,1)

- Example

C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = red$$
$$0.6 \leq u < 0.7, \rightarrow C = green$$
$$0.7 \leq u < 1, \rightarrow C = blue$$

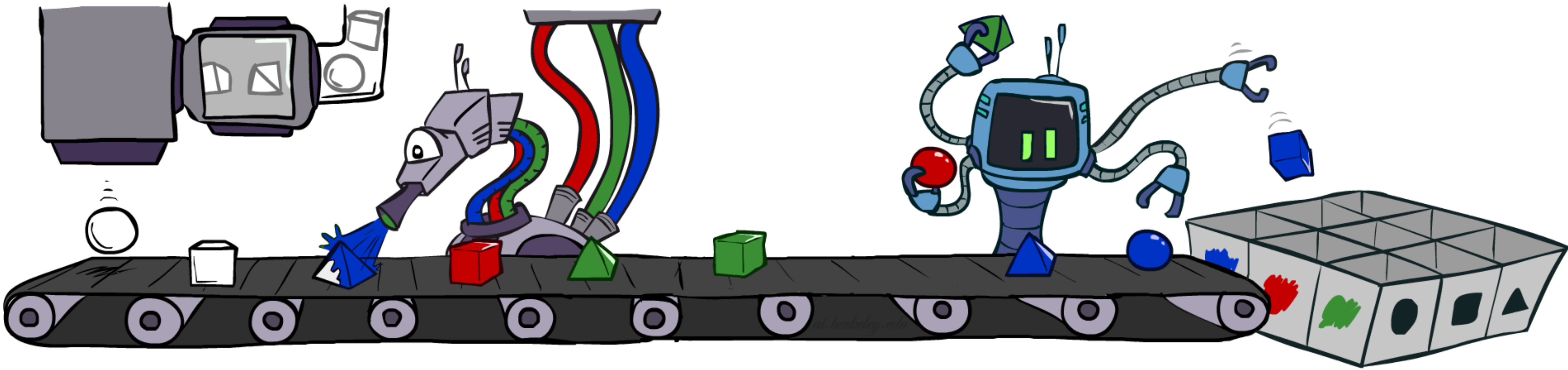
  - If random() returns u = 0.83, then our sample is C = blue
  - E.g, after sampling 8 times:



Sampling in Bayes Nets

- Prior Sampling
- Rejection Sampling
- Likelihood Weighting
- Gibbs Sampling

Prior Sampling



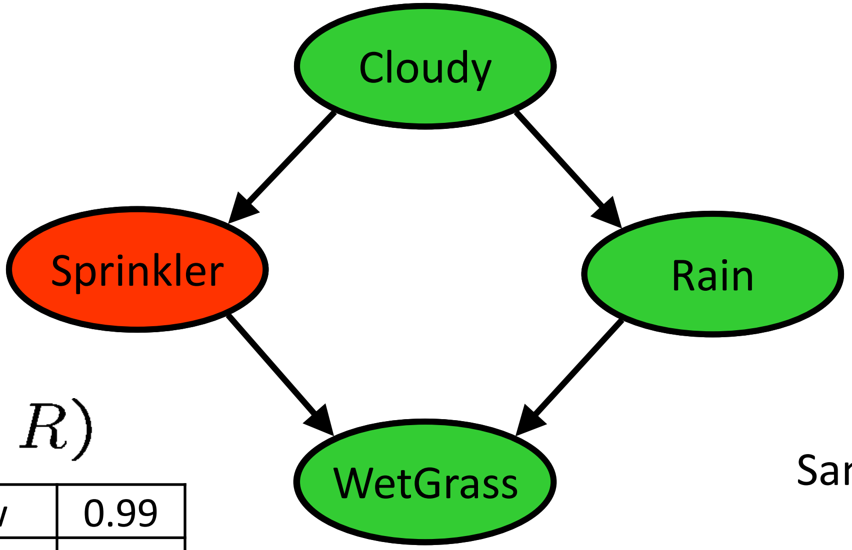
Prior Sampling

$P(C)$

c	0.5
¬c	0.5

$P(S|C)$

c	s	0.1
	¬s	0.9
¬c	s	0.5
	¬s	0.5



$P(R|C)$

c	r	0.8
	¬r	0.2
¬c	r	0.2
	¬r	0.8

$P(W|S,R)$

s	r	w	0.99
		¬w	0.01
	¬r	w	0.90
		¬w	0.10
¬s	r	w	0.90
		¬w	0.10
	¬r	w	0.01
		¬w	0.99

Samples:

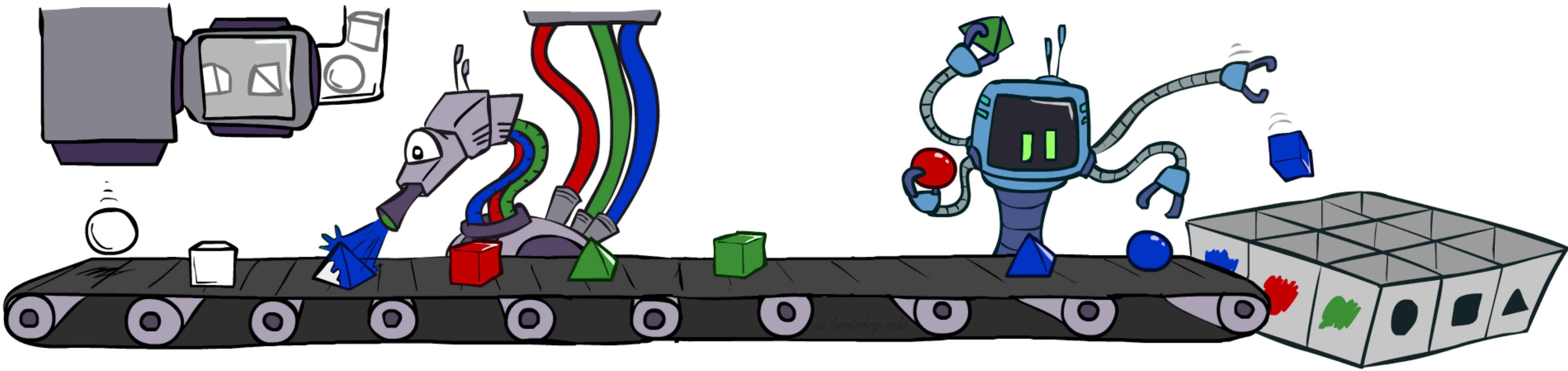
c, ¬s, r, w

¬c, s, ¬r, w

…

Prior Sampling

- For i=1, 2, …, n (in topological order)
  - Sample X_i from $P(X_i \mid parents(X_i))$
- Return $(x_1, x_2, \ldots, x_n)$



Using samples

- We'll get a bunch of samples from the BN:

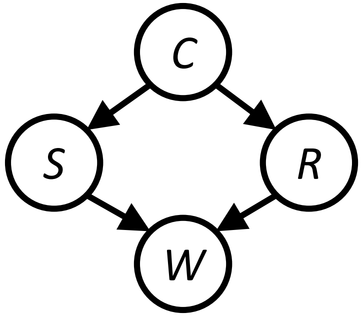
  c, ¬s, r, w

  ¬c, s, r, w

  ¬c, s, r, ¬w

  c, ¬s, r, w

  ¬c, ¬s, ¬r, w



- If we want to know P(*W*)
  - We have counts <w:4, ¬w:1>
  - Normalize to get P(*W*) = <w:0.8, ¬w:0.2>
  - This will get closer to the true distribution with more samples
- If we want to know P(*C*| *r, w*)
  - Count (*c, r, w*) and (¬*c, r, w*)
  - Normalize to get P(*C*| *r, w*) = <c:0.67, ¬c:0.33>

Prior Sampling

- This process generates samples with probability:

$$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{Parents}(X_i)) = P(x_1 \ldots x_n)$$

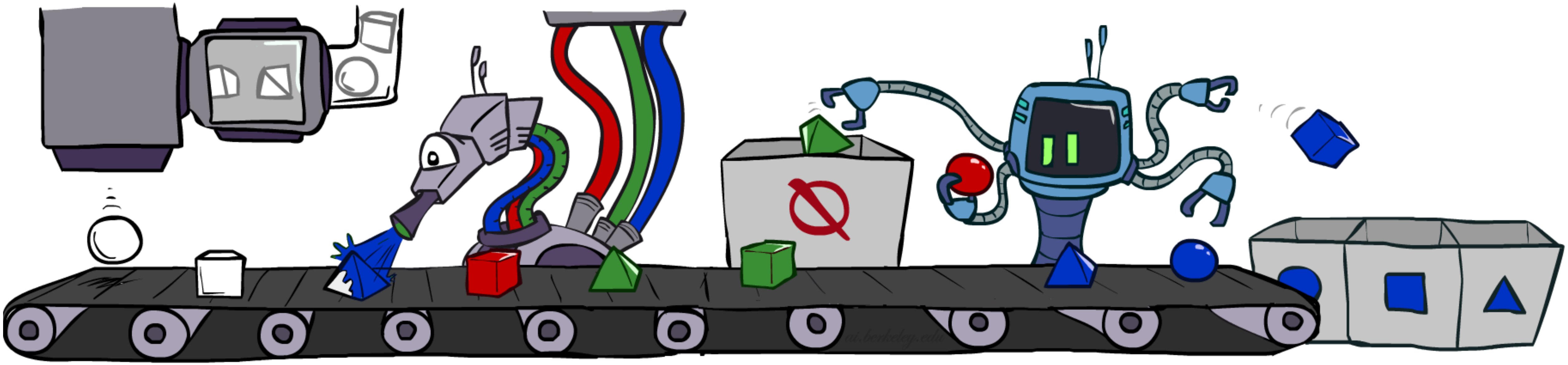
…i.e. the BN's joint probability

- Let the number of samples of an assignment be $N_{PS}(x_1 \ldots x_n)$
- So $\widehat{P}(x_1, \ldots, x_n) = N_{PS}(x_1, \ldots, x_n)/N$
- Then

$$\begin{aligned} \lim_{N\to\infty} \widehat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

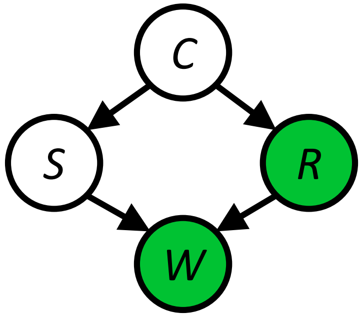
- I.e., the sampling procedure is consistent

Rejection Sampling



Rejection Sampling

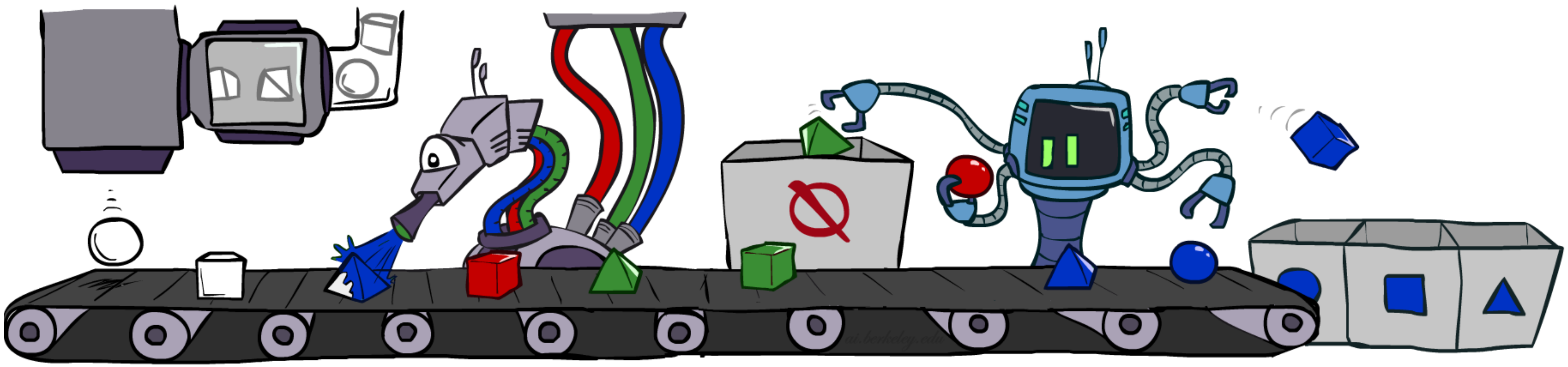
- A simple modification of prior sampling for conditional probabilities
- Let's say we want P(*C*| *r*, *w*)
- When generating a sample, reject it immediately if not *R*=true, *W*=true
- It is consistent for conditional probabilities (i.e., correct in the limit)



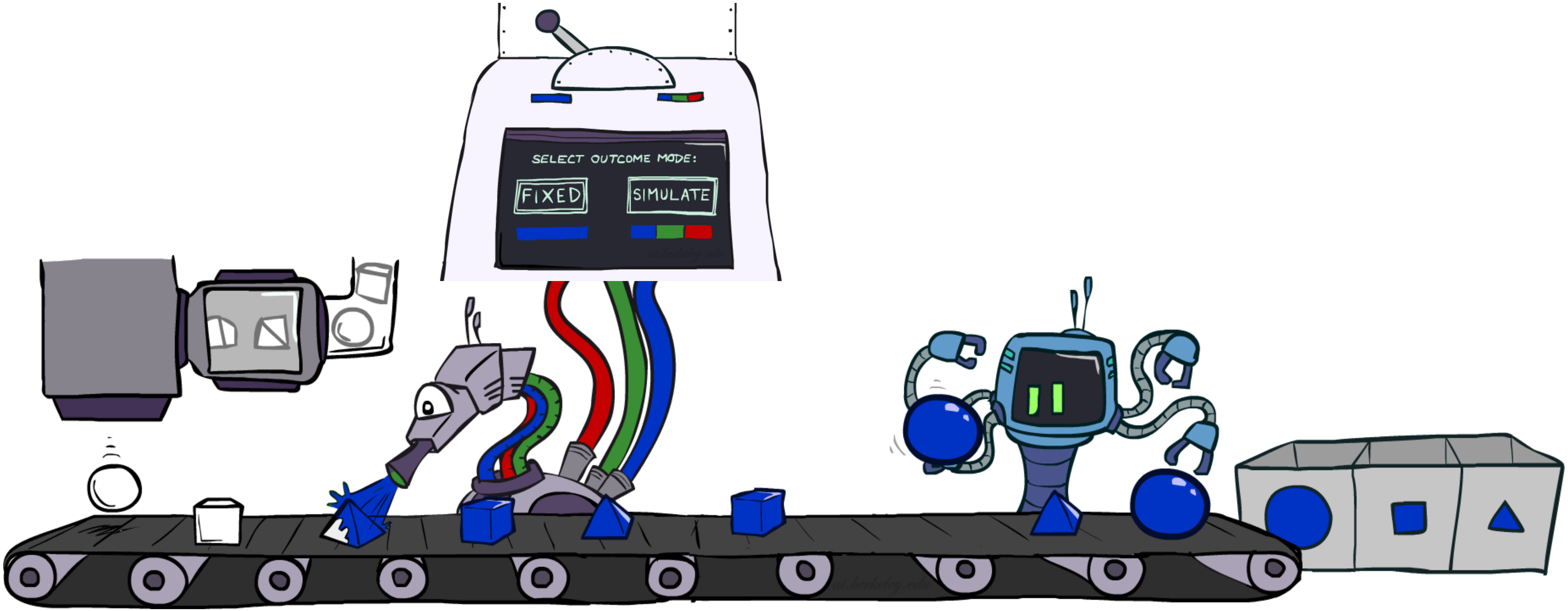
c, ¬s, r, w
~~c, s, ¬r~~
~~¬c, s, r, ¬w~~
~~c, ¬s, ¬r~~
¬c, ¬s, r, w

Rejection Sampling

- Input: evidence $e_1,..,e_k$
- For i=1, 2, …, n
  - Sample X_i from $P(X_i \mid parents(X_i))$
  - If x_i not consistent with evidence
    - Reject: Return, and no sample is generated in this cycle
- Return $(x_1, x_2, \ldots, x_n)$

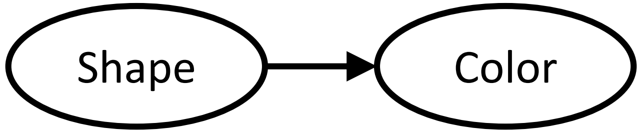


Likelihood Weighting

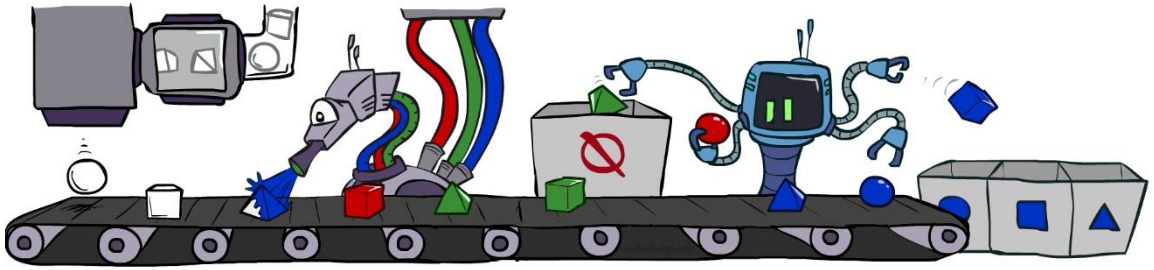


Likelihood Weighting

- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(*Shape*|*Color=blue*)

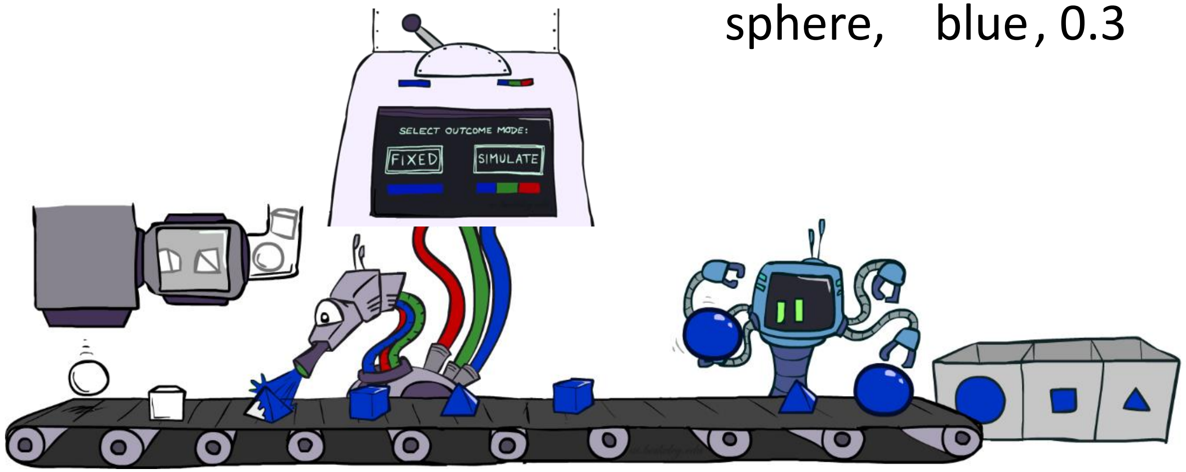


~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



- Idea: fix evidence variables, sample the rest
  - Problem: sample distribution not consistent!
  - Solution: ***weight*** each sample by probability of evidence variables given parents

pyramid, blue , 0.4
pyramid, blue , 0.4
sphere, blue , 0.3
cube, blue , 0.8
sphere, blue , 0.3



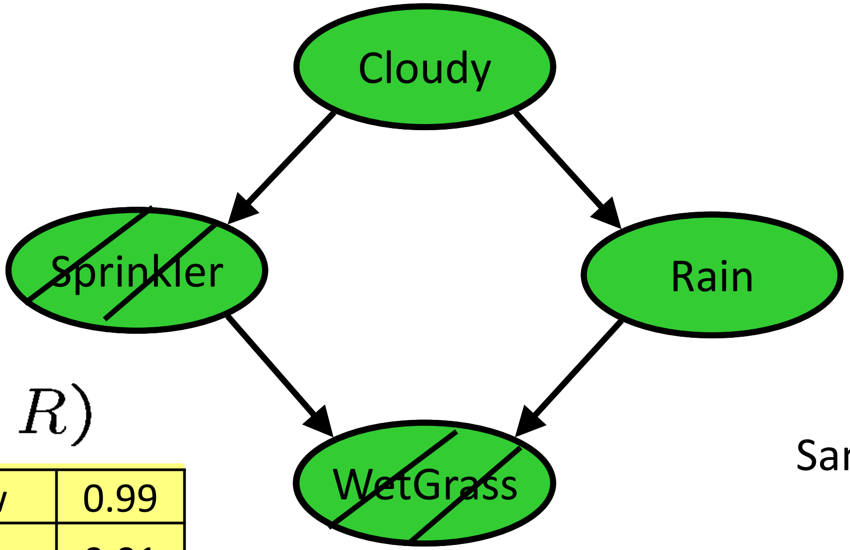
Likelihood Weighting

$P(C)$

c	0.5
¬c	0.5

$P(S|C)$

c	s	0.1
	¬s	0.9
¬c	s	0.5
	¬s	0.5



$P(R|C)$

c	r	0.8
	¬r	0.2
¬c	r	0.2
	¬r	0.8

$P(W|S,R)$

s	r	w	0.99
		¬w	0.01
	¬r	w	0.90
		¬w	0.10
¬s	r	w	0.90
		¬w	0.10
	¬r	w	0.01
		¬w	0.99

Samples:

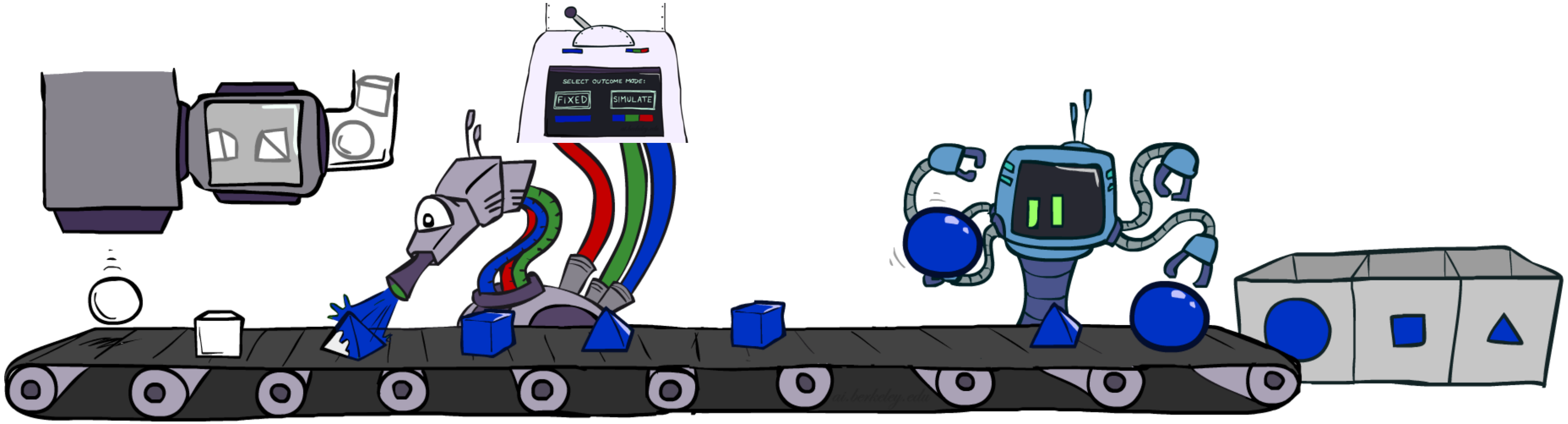
c, s, r, w

...

$$w = 1.0 \times 0.1 \times 0.99$$

Likelihood Weighting

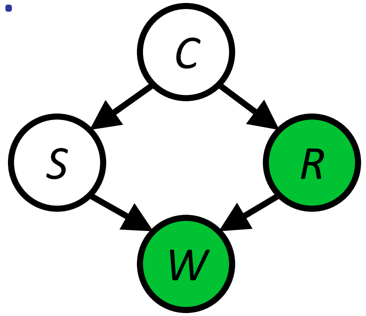
- Input: evidence $e_1,..,e_k$
- w = 1.0
- for i=1, 2, …, n
  - if X_i is an evidence variable
    - x_i = observed value$_i$ for X_i
    - Set $w = w * P(x_i \mid Parents(X_i))$
  - else
    - Sample x_i from $P(X_i \mid Parents(X_i))$
- return $(x_1, x_2, \ldots, x_n)$, w



Using samples

- We'll get a bunch of weighted samples from the BN:

c, ¬s, r, w	0.1
c, s, r, w	0.2
¬c, s, r, w	0.3
c, ¬s, r, w	0.1
¬c, ¬s, r, w	0.5

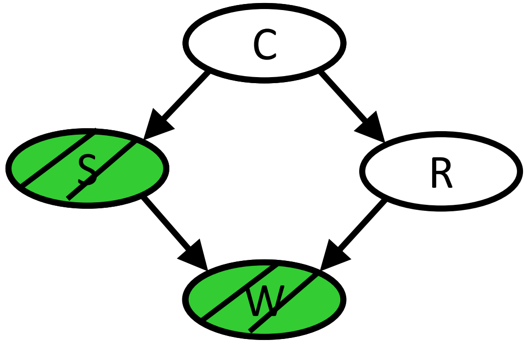


- If we want to know $P(C \mid r, w)$
  - We have weight sums <(c, r, w): 0.4, $(\neg c, r, w)$: 0.8>
  - Normalize to get $P(C \mid r, w)$ = <c: 0.33, $\neg c$: 0.67>
  - This will get closer to the true distribution with more samples

Likelihood Weighting

- Sampling distribution (z is sampled and e is fixed evidence)

$$S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|\text{Parents}(Z_i))$$



- Now, samples have weights

$$w(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{m} P(e_i|\text{Parents}(E_i))$$

- Together, weighted sampling distribution is consistent

$$S_{\text{WS}}(z,e) \cdot w(z,e) = \prod_{i=1}^{l} P(z_i|\text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i|\text{Parents}(e_i))$$

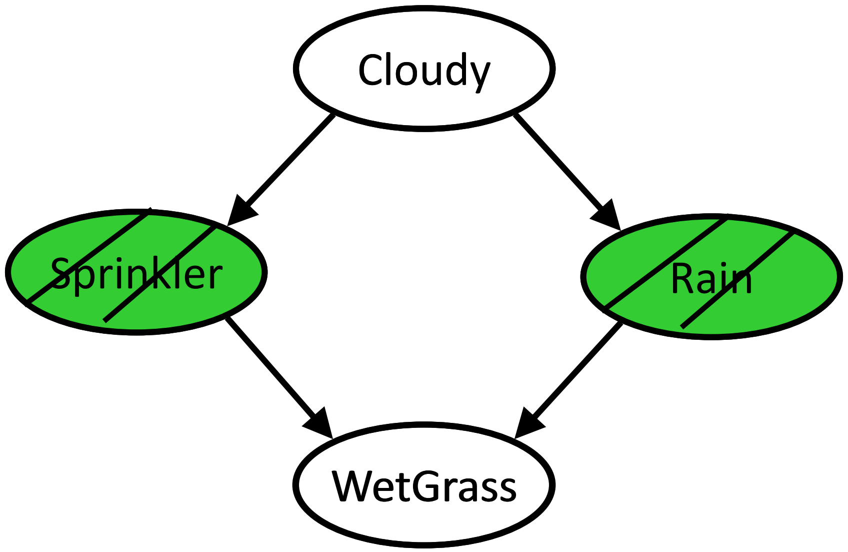
$$= P(\mathbf{z},\mathbf{e})$$

Importance Sampling

- Likelihood weighting is an instance of importance sampling
  - Suppose it is difficult to sample from p(x)
  - Generate samples from a proposal distribution q(x)
    - q(x) is easy to draw samples from
  - Weight each sample by p(x)/q(x)
- The choice of q(x) would greatly influence the speed of convergence
  - If you want to estimate the expectation of f(x)
  - Then q(x) should be close to being proportional to |f(x)|p(x)

Likelihood Weighting

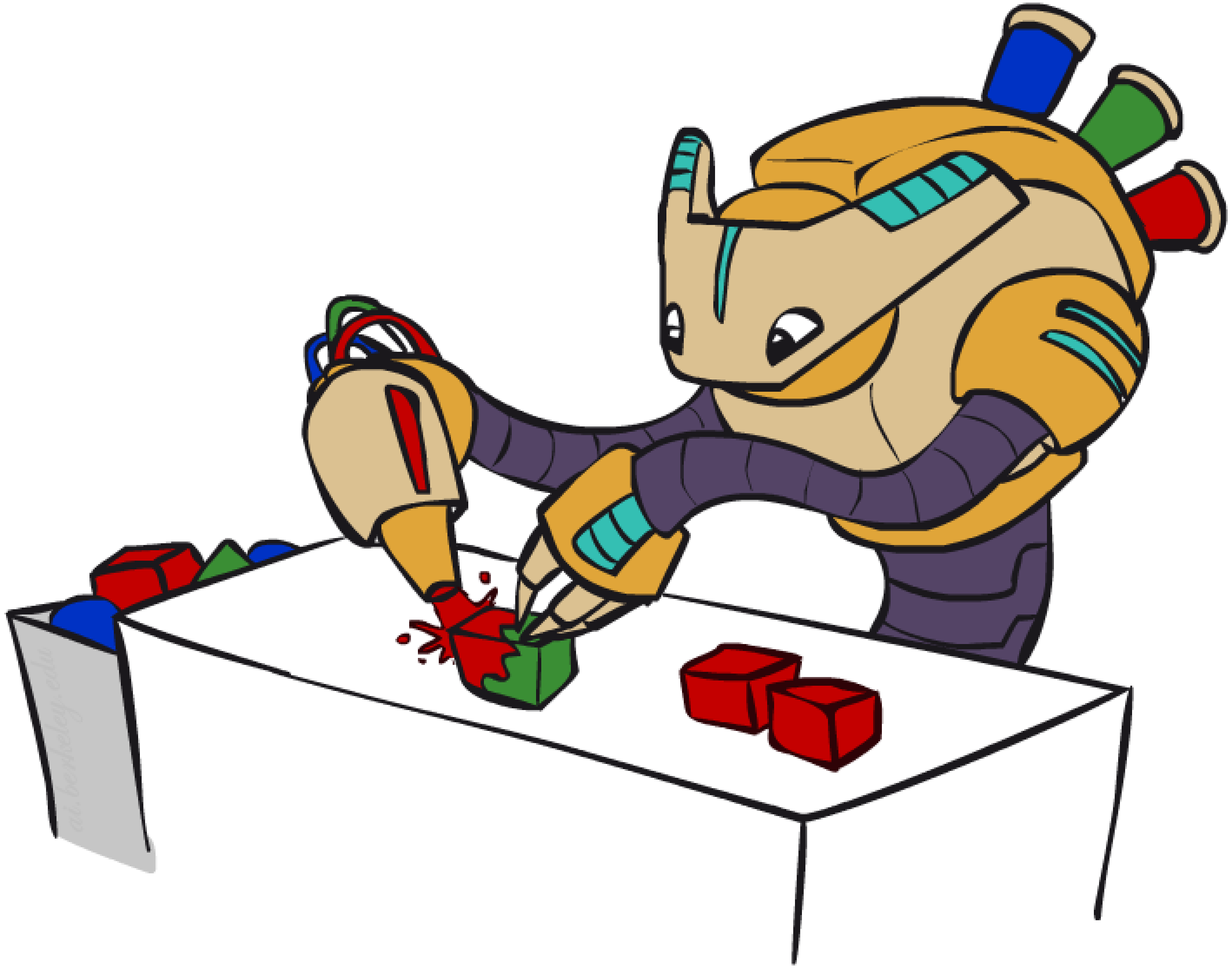
- Likelihood weighting is good
  - All samples are used
  - The values of ***downstream*** variables are influenced by ***upstream*** evidence



- Likelihood weighting still has weaknesses
  - The values of ***upstream*** variables are unaffected by ***downstream*** evidence
  - With many downstream evidence, we may
    - mostly get samples that are inconsistent with the evidence and thus have very small weights
    - get a few lucky samples with very large weights, which dominate the result

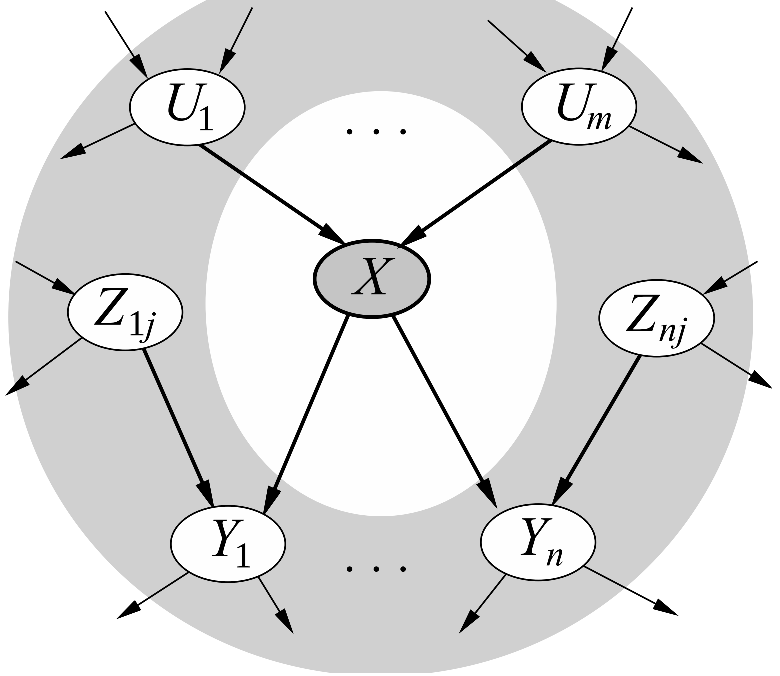
- We would like each variable to "see" ***all*** the evidence!

Gibbs Sampling



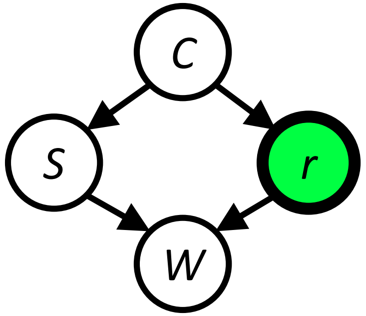
Gibbs Sampling

- Generate each sample by making a random change to the preceding sample
  - Evidence variables remain fixed. For each of the non-evidence variable, sample its value conditioned on all the other variables
  - $X_i' \sim P(X_i \mid x_1,..,x_{i-1},x_{i+1},..,x_n)$
  - In a Bayes net
    $P(X_i \mid x_1,..,x_{i-1},x_{i+1},..,x_n)$
    $= P(X_i \mid markov\_blanket(X_i))$
    $= \alpha\ P(X_i \mid u_1,..,u_m) \prod_j P(y_j \mid parents(Y_j))$

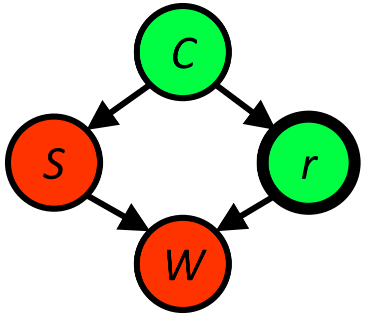


Gibbs Sampling Example: *P(S | r)*

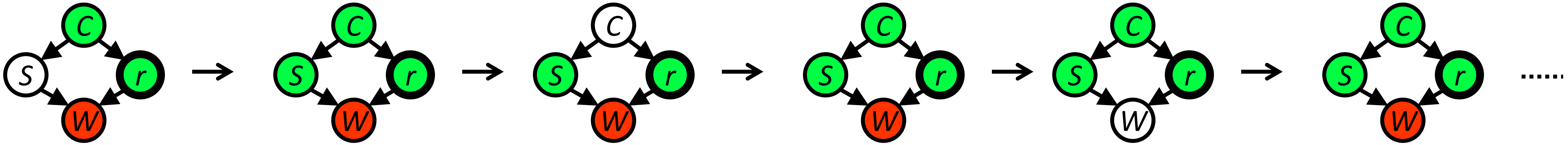
- Step 1: Fix evidence
  - *R* = true



- Step 2: Initialize other variables
  - Randomly



- Step 3: Repeat
  - Choose an arbitrary non-evidence variable *X*
  - Resample *X* from *P*(*X* | *markov_blanket*(*X*))

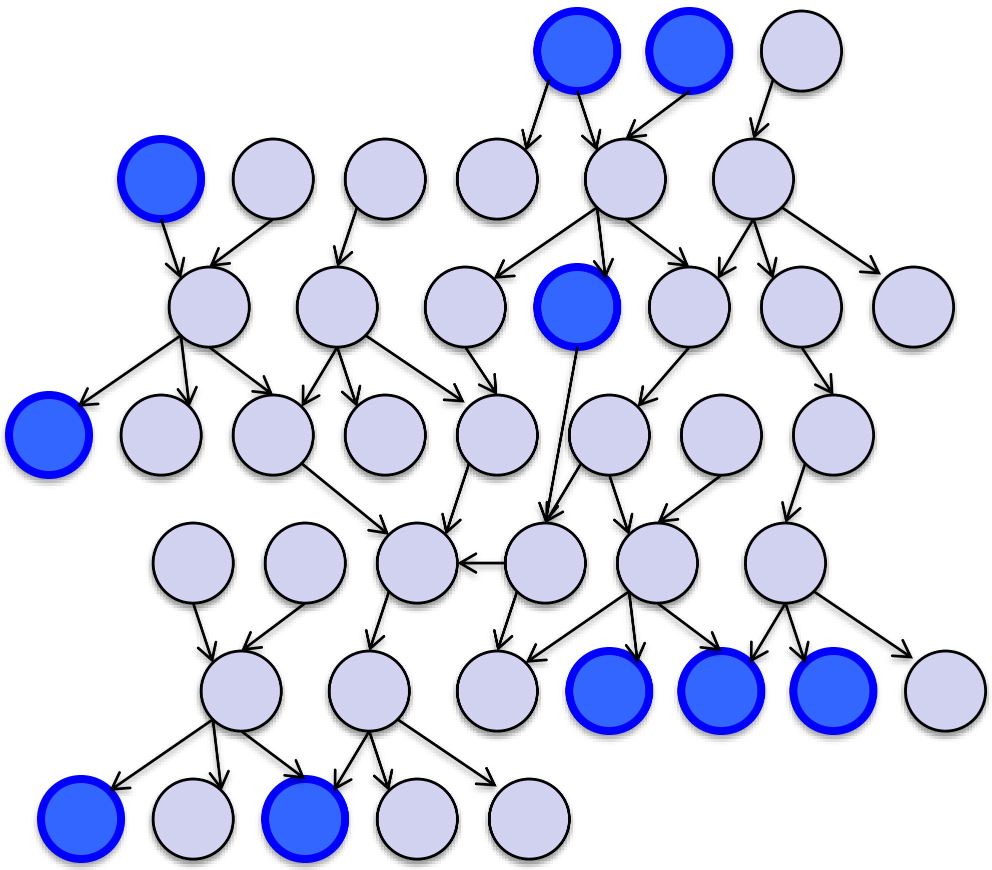


Sample *S* ~ *P*(*S* | *c*, *r*, ¬*w*)

Sample *C* ~ *P*(*C* | *s*, *r*)

Sample *W* ~ *P*(*W* | *s*, *r*)

Why doing this?



- Samples soon begin to reflect all the evidence in the network
- Eventually they are being drawn from the true posterior!
- Theorem: Gibbs sampling is consistent

- MCM … ing
  som …
  - M … walk”),
    w …
  - M … ino

Markov Chain Monte Carlo (MCMC)

- MCMC is a family of randomized algorithms for approximating some quantity of interest over a very large state space
  - Markov chain = a sequence of randomly chosen states (“random walk”), where each state is chosen conditioned on the previous state
  - ~~Monte Carlo = a very expensive city in Monaco with a famous casino~~
  - Monte Carlo = an algorithm (usually based on sampling) that is likely to find a correct answer
- MCMC = sampling by constructing a Markov chain
- Gibbs, Metropolis-Hastings, Hamiltonian, Slice, etc.

Metropolis-Hastings

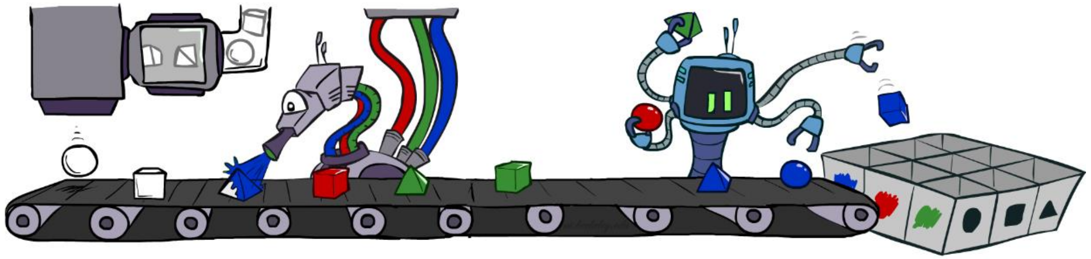
- Repeat
  1. Draw a sample from a proposal distribution $g(x'|x)$
     - $g(x'|x)$ is typically easy to sample from
  2. Accept this sample with probability

$$\min\left(1, \frac{P(x')g(x|x')}{P(x)g(x'|x)}\right)$$

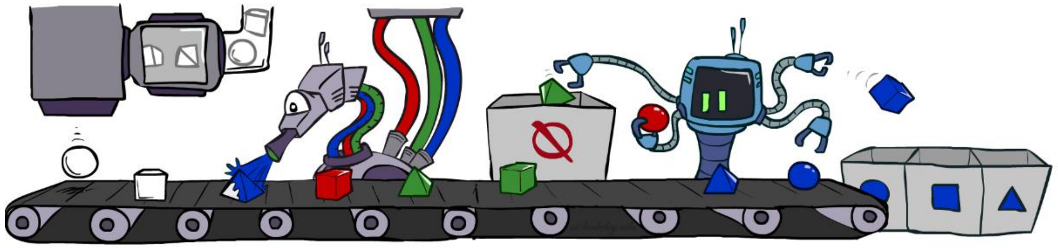
- Gibbs is a special case of Metropolis-Hastings in which the acceptance rate is always 1

Summary

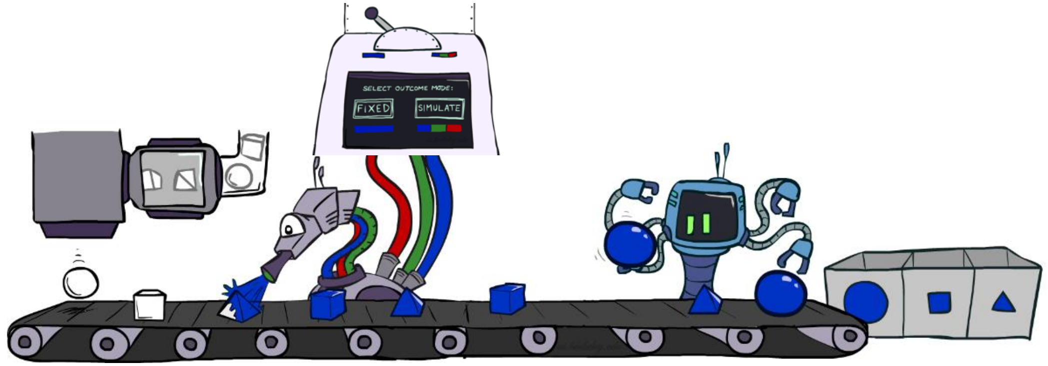
- Prior Sampling P



- Rejection Sampling P(Q | e)



- Likelihood Weighting P(Q | e)



- Gibbs Sampling P(Q | e)