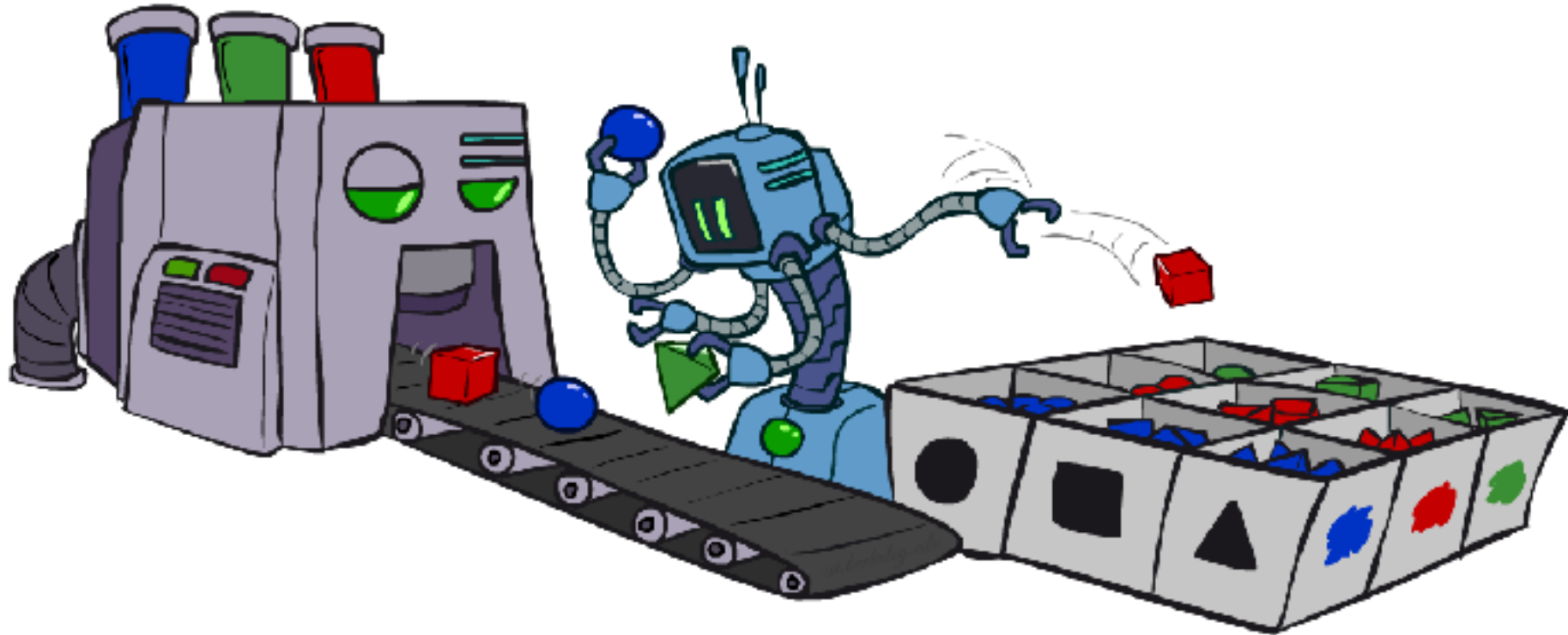


# CS 383: Artificial Intelligence

## Bayes Nets: Sampling



Prof. Scott Niekum — UMass Amherst

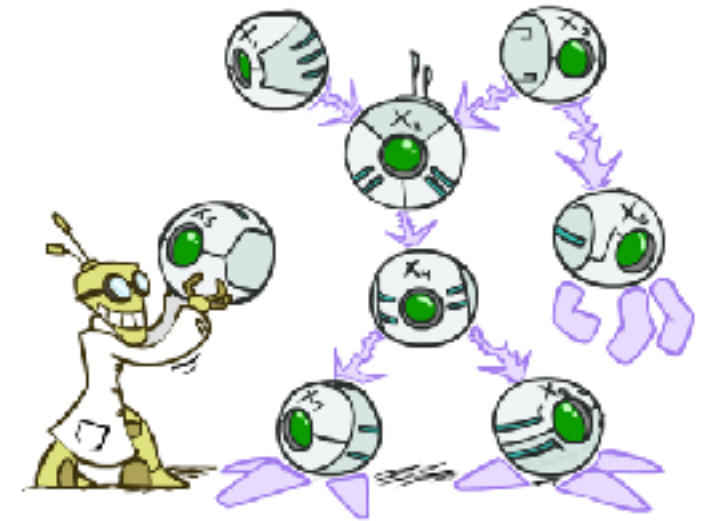
# Bayes Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over  $X$ , one for each combination of parents' values

$$P(X|a_1 \dots a_n)$$

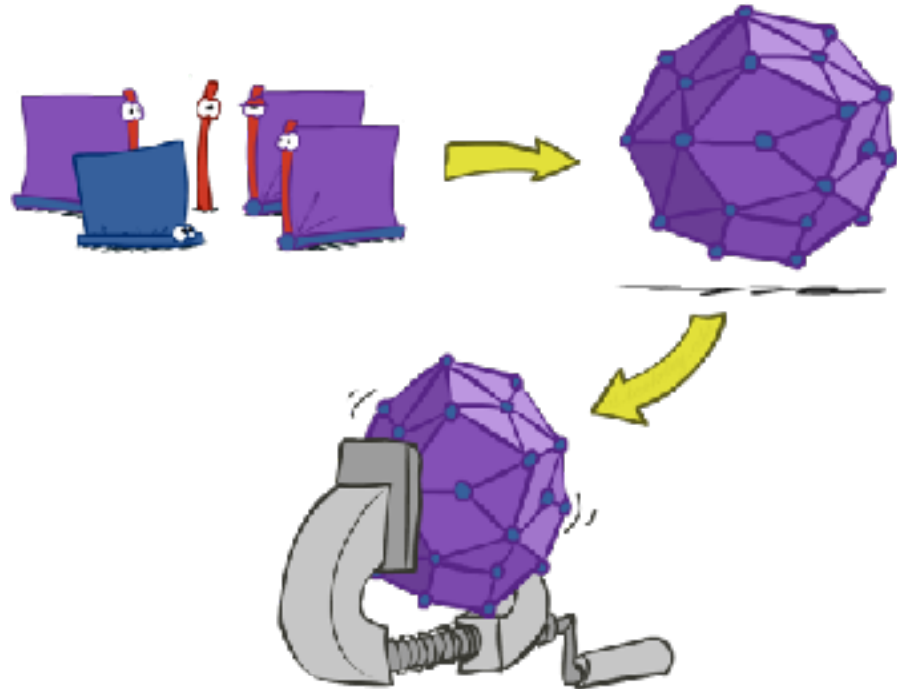
- Bayes nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

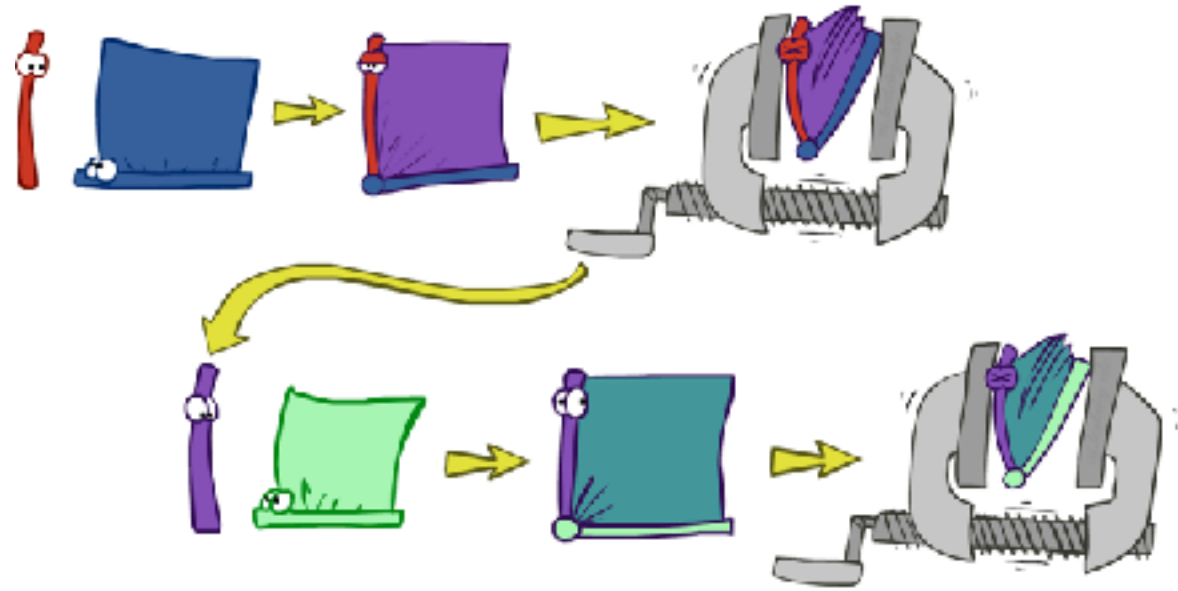


# Inference by Enumeration vs. Variable Elimination

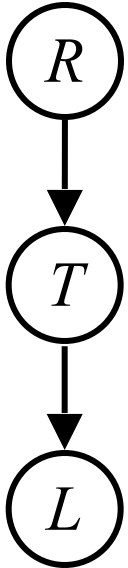
- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables



- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration



# Traffic Domain



$$P(L) = ?$$

## ■ Inference by Enumeration

$$= \sum_t \sum_r \underbrace{P(L|t)P(r)P(t|r)}_{\text{Join on } r}$$
$$\underbrace{\hspace{10em}}_{\text{Join on } t}$$
$$\underbrace{\hspace{10em}}_{\text{Eliminate } r}$$
$$\underbrace{\hspace{10em}}_{\text{Eliminate } t}$$

## ■ Variable Elimination

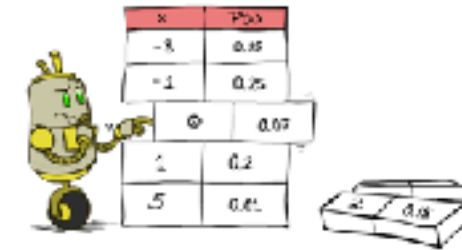
$$= \sum_t P(L|t) \underbrace{\sum_r P(r)P(t|r)}_{\text{Join on } r}$$
$$\underbrace{\hspace{10em}}_{\text{Eliminate } r}$$
$$\underbrace{\hspace{10em}}_{\text{Join on } t}$$
$$\underbrace{\hspace{10em}}_{\text{Eliminate } t}$$

# General Variable Elimination

■ Query:  $P(Q|E_1 = e_1, \dots, E_k = e_k)$

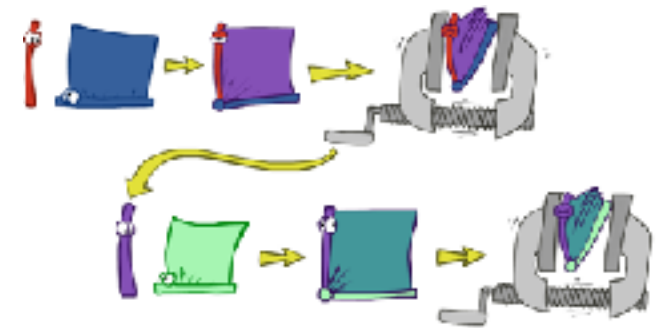
■ Start with initial factors:

- Local CPTs (but instantiated by evidence)



■ While there are still hidden variables (not Q or evidence):

- Pick a hidden variable H
- Join all factors mentioning H
- Eliminate (sum out) H

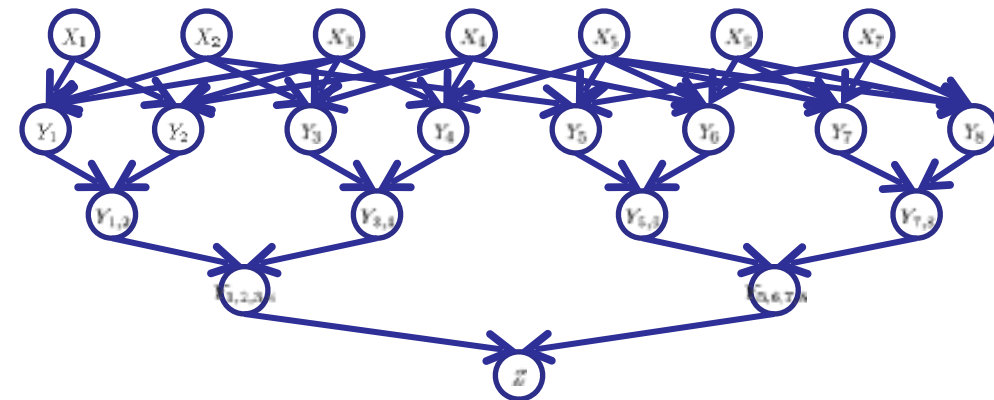
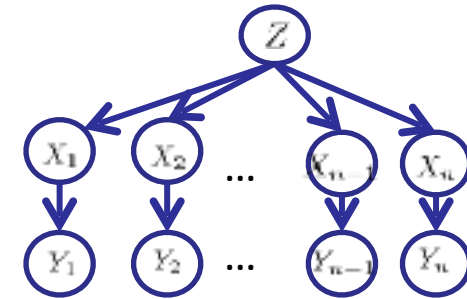
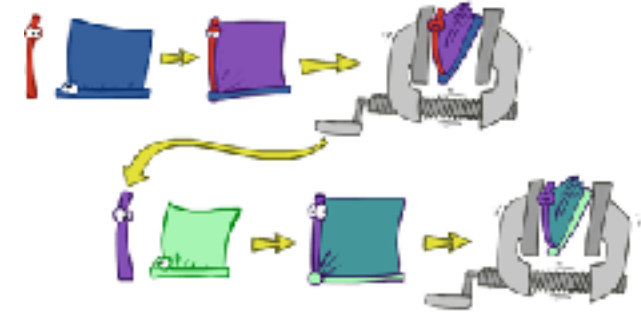


■ Join all remaining factors and normalize



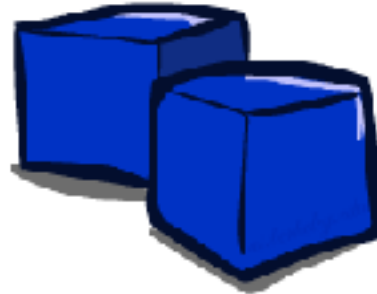
# Variable Elimination Efficiency

- Interleave joining and marginalizing, instead of fully joining all at once (i.e. enumeration)
- $d^k$  entries computed for a factor over  $k$  variables with domain sizes  $d$
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes net (NP-hard)



# Approximate Inference: Sampling

---



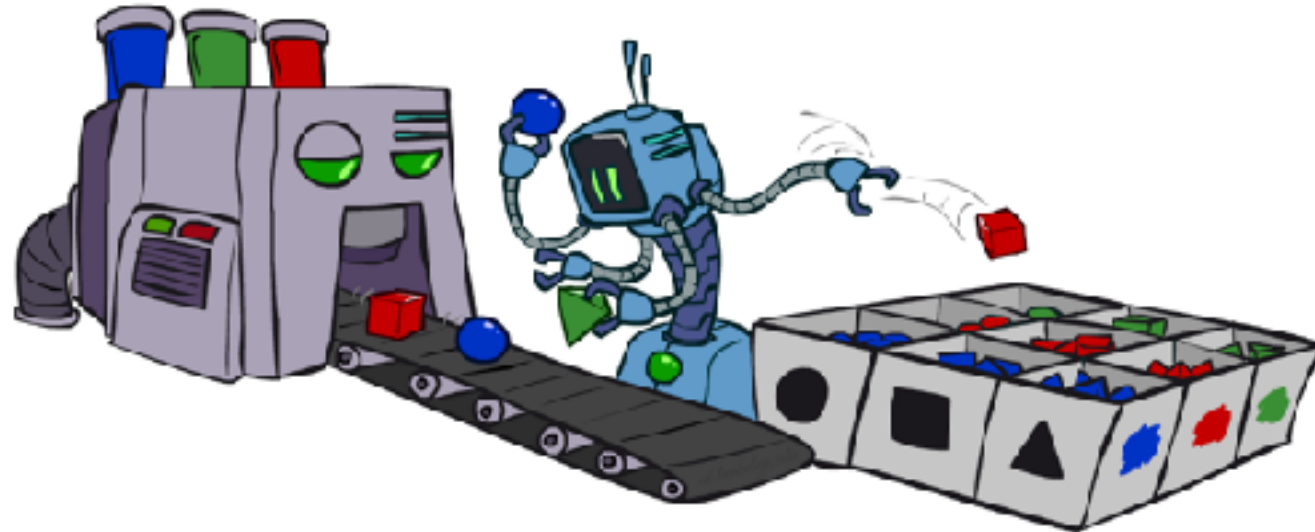
# Sampling

- Basic idea

- Draw  $N$  samples from a sampling distribution  $S$
- Compute an approximate posterior probability
- Show this converges to the true probability  $P$

- Why sample?

- Learning: get samples from a distribution you don't know
- Inference: getting samples can be faster than computing the right answer (e.g. with variable elimination)





# Sampling

- Sampling from given distribution

- Step 1: Get sample  $u$  from uniform distribution over  $[0, 1)$ 
  - E.g. `random()` in python
- Step 2: Convert this sample  $u$  into an outcome for the given distribution by having each outcome associated with a sub-interval of  $[0,1)$  with sub-interval size equal to probability of the outcome

- Example

C	P(C)
red	0.6
green	0.1
blue	0.3

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

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

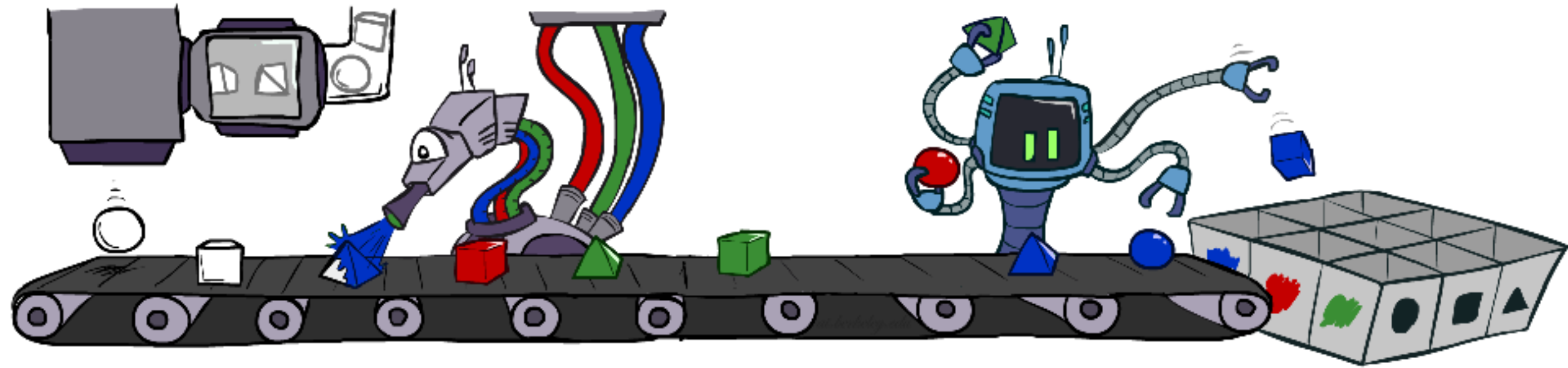


# Sampling in Bayes Nets

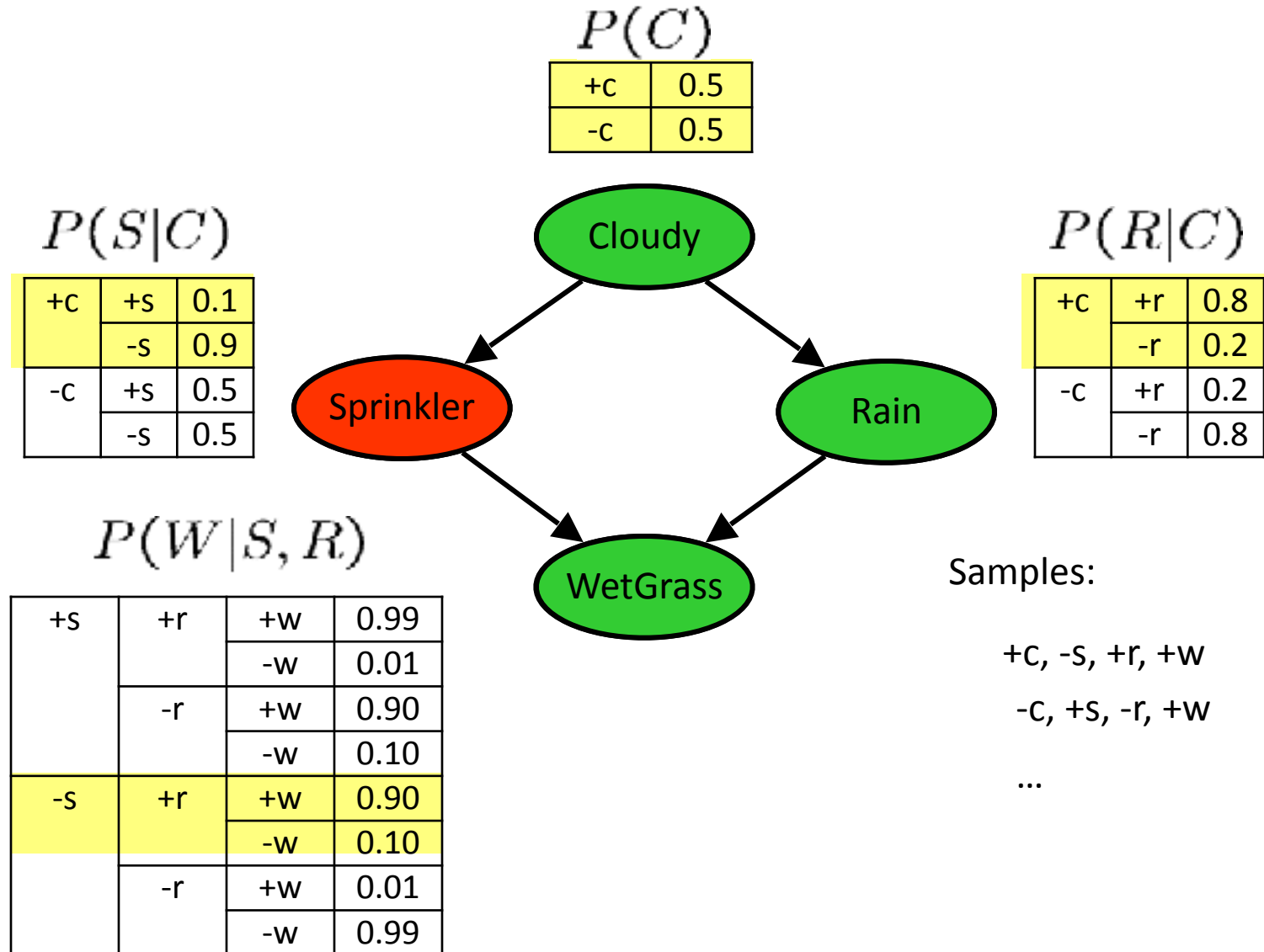
---

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

# Prior Sampling

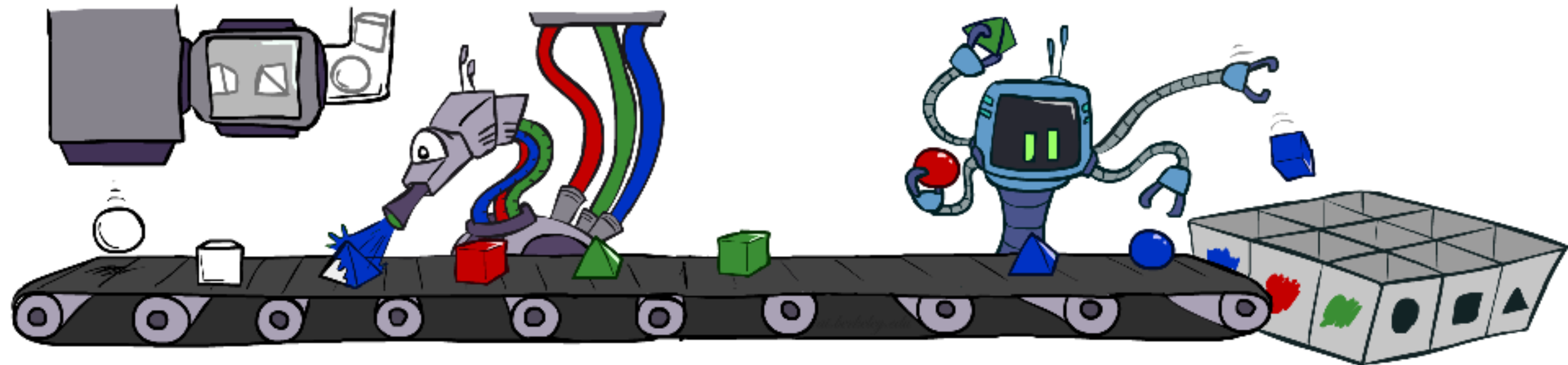


# Prior Sampling



# Prior Sampling

- For  $i=1, 2, \dots, n$ 
  - Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$
- Return  $(x_1, x_2, \dots, x_n)$



# Prior Sampling

- This process generates samples with probability:

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

...i.e. the BN's joint probability

- Let the number of samples of a particular event be  $N_{PS}(x_1 \dots x_n)$  and the total number of samples of all events be  $N$ .

- Then 
$$\begin{aligned} \lim_{N \rightarrow \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \rightarrow \infty} N_{PS}(x_1, \dots, x_n) / N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$

- I.e., the sampling procedure is **consistent**

# Example

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

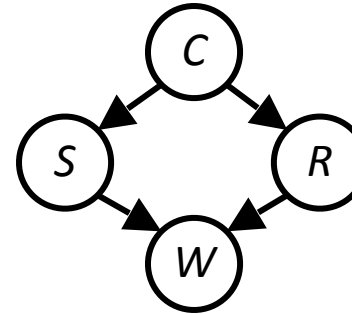
+c, -s, +r, +w

+c, +s, +r, +w

-c, +s, +r, -w

+c, -s, +r, +w

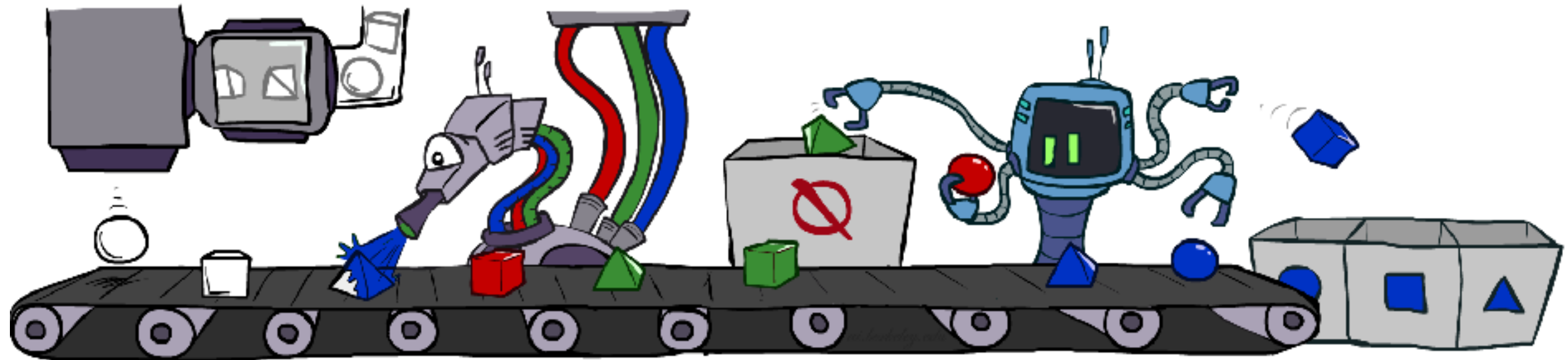
-c, -s, -r, +w



- If we want to know  $P(W)$

- We have counts  $\langle +w:4, -w:1 \rangle$
- Normalize to get  $P(W) = \langle +w:0.8, -w:0.2 \rangle$
- This will get closer to the true distribution with more samples
- Can estimate anything else, too
- What about  $P(C \mid +w)$ ?  $P(C \mid +r, +w)$ ?  $P(C \mid -r, -w)$ ?
- Fast: can use fewer samples if less time (what's the drawback?)

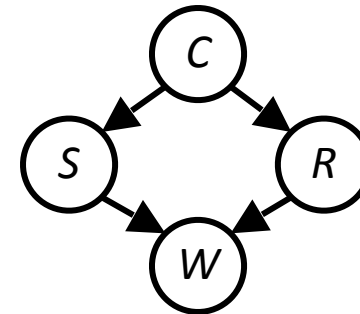
# Rejection Sampling





# Rejection Sampling

- Let's say we want  $P(C)$ 
  - No point keeping all samples around
  - Just tally counts of  $C$  as we go
- Let's say we want  $P(C \mid +s)$ 
  - Same thing: tally  $C$  outcomes, but ignore (reject) samples which don't have  $S=+s$
  - This is called rejection sampling
  - It is also consistent for conditional probabilities (i.e., correct in the limit)



~~+c, -s, +r, +w~~

+c, +s, +r, +w

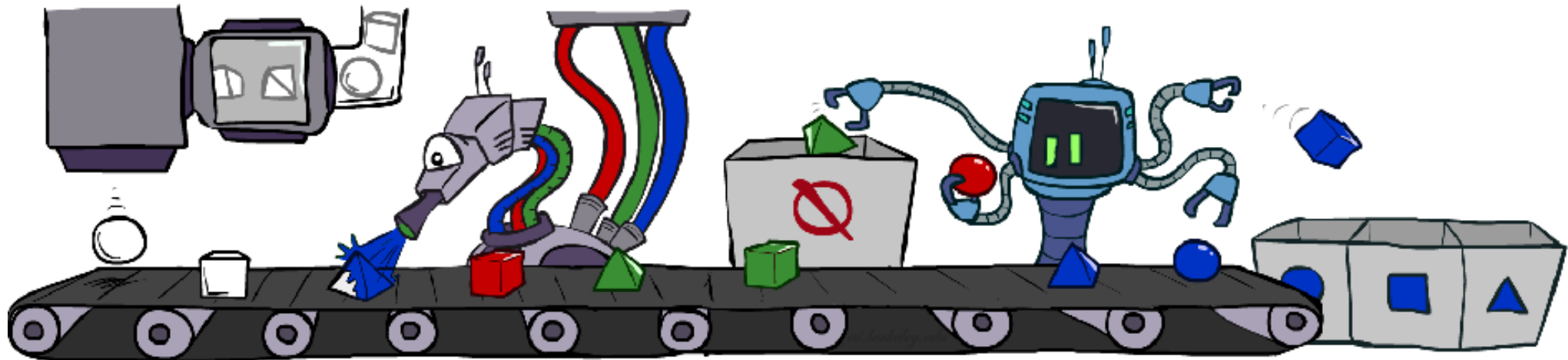
-c, +s, +r, -w

~~+c, -s, +r, +w~~

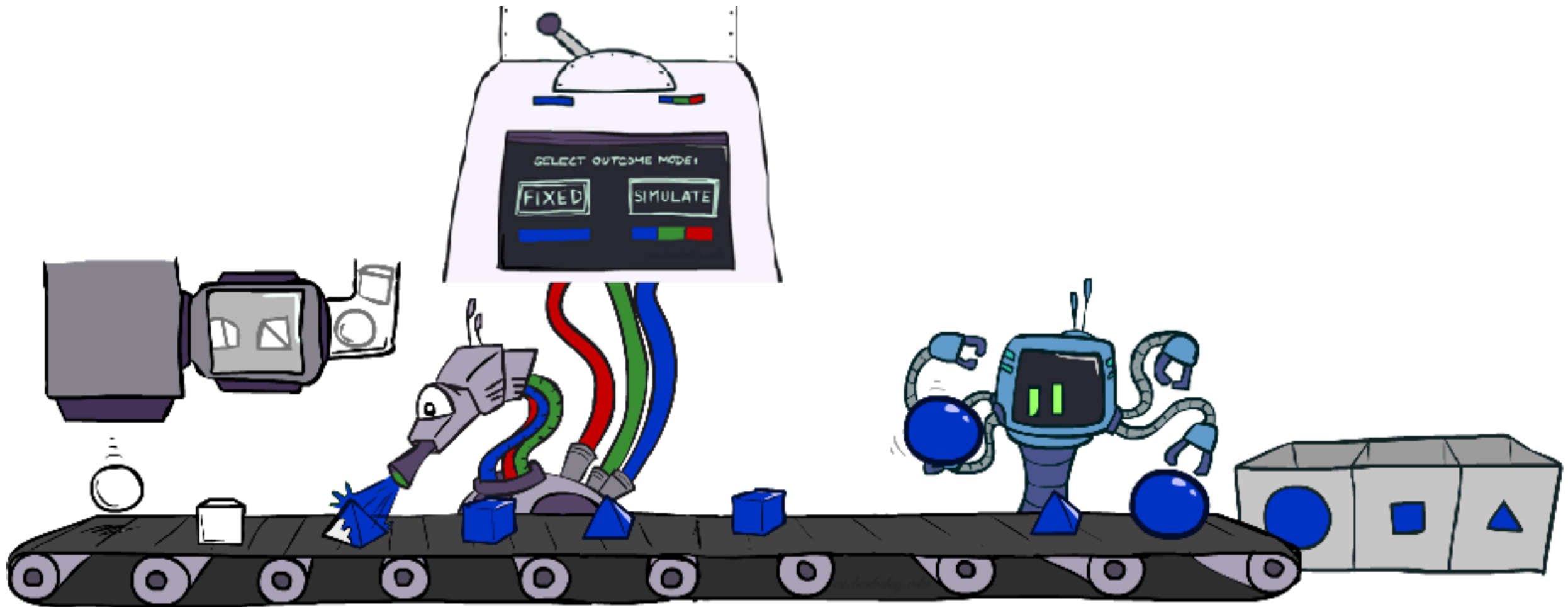
~~-c, -s, -r, +w~~

# Rejection Sampling

- IN: evidence instantiation
- For  $i=1, 2, \dots, n$ 
  - Sample  $x_i$  from  $P(X_i \mid \text{Parents}(X_i))$
  - If  $x_i$  not consistent with evidence
    - Reject: Return, and no sample is generated in this cycle
- Return  $(x_1, x_2, \dots, x_n)$

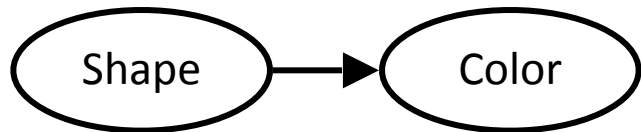


# Likelihood Weighting

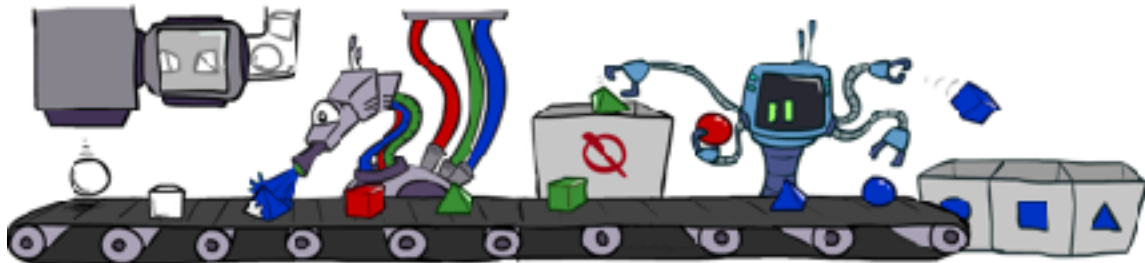


# Likelihood Weighting

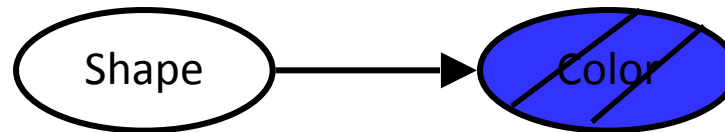
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider  $P(\text{Shape} \mid \text{blue})$



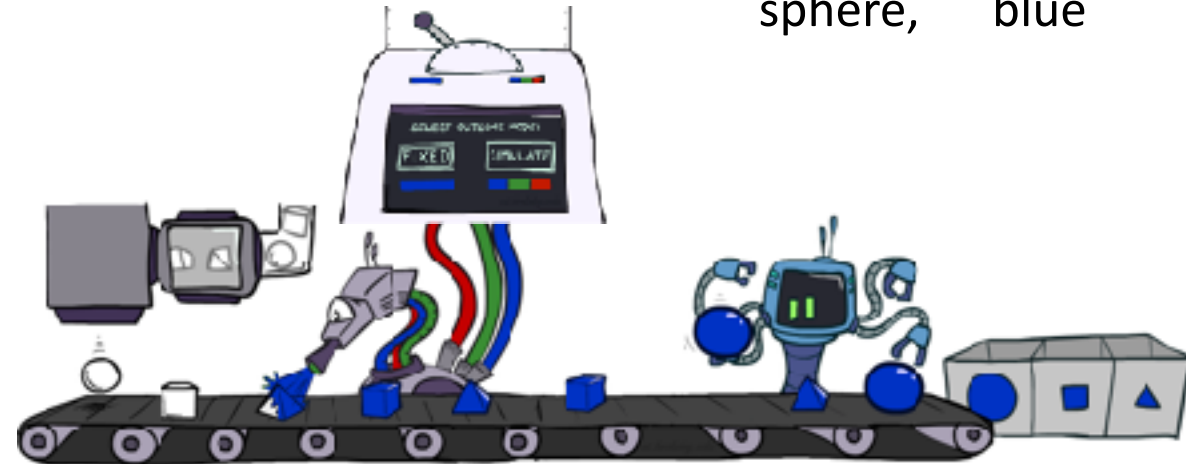
~~pyramid, green~~  
~~pyramid, red~~  
sphere, blue  
~~cube, red~~  
~~sphere, green~~



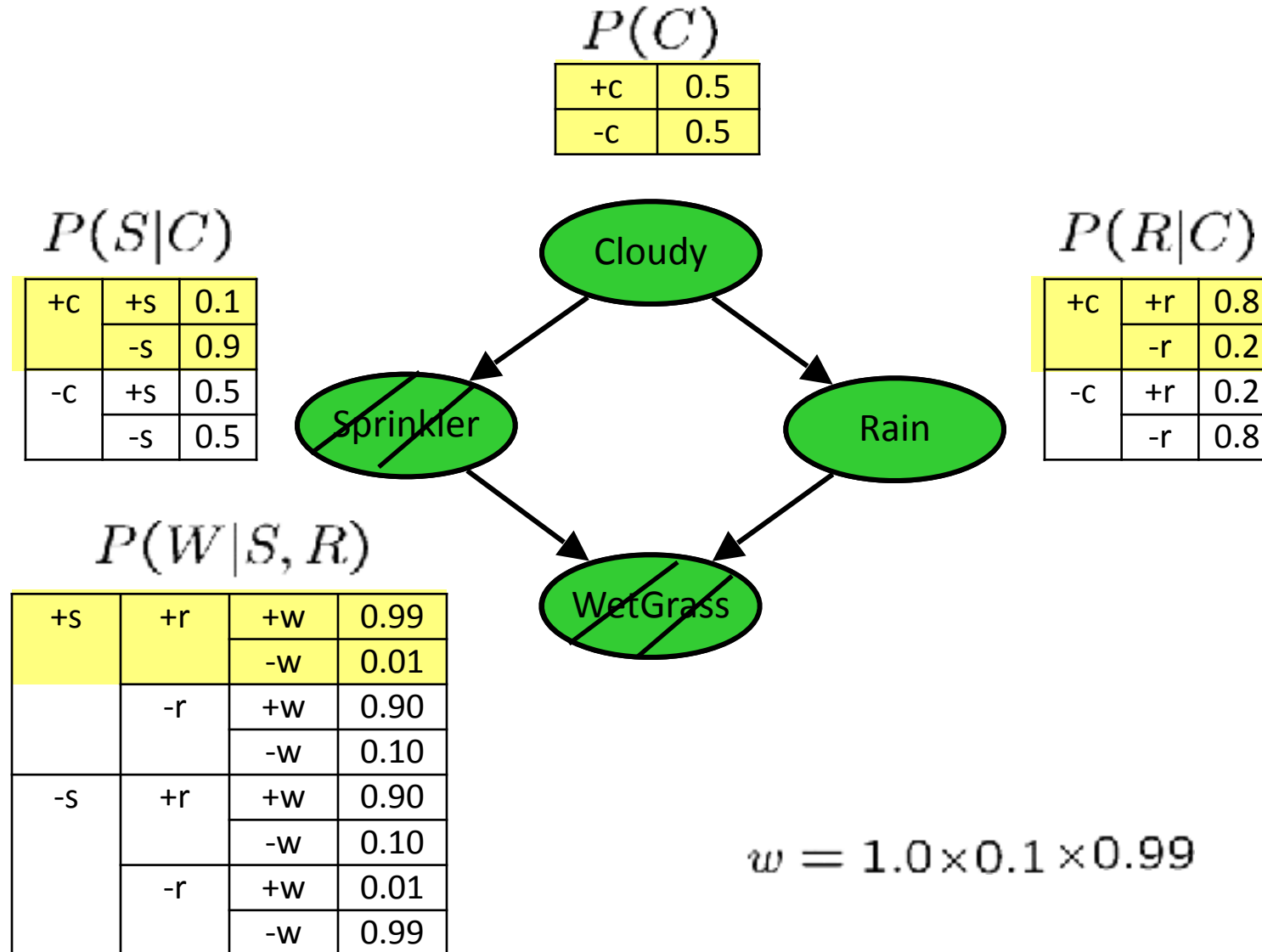
- Idea: fix evidence variables and sample the rest
  - Problem: sample distribution not consistent!
  - Solution: weight by probability of evidence given parents



pyramid, blue  
pyramid, blue  
sphere, blue  
cube, blue  
sphere, blue



# Likelihood Weighting



Weight = probability of evidence given parents

Evidence:

+s, +w

Sample:

+c, +s, +r, +w

Weight = ?

iClicker:

A: 1.0

B: 0.99

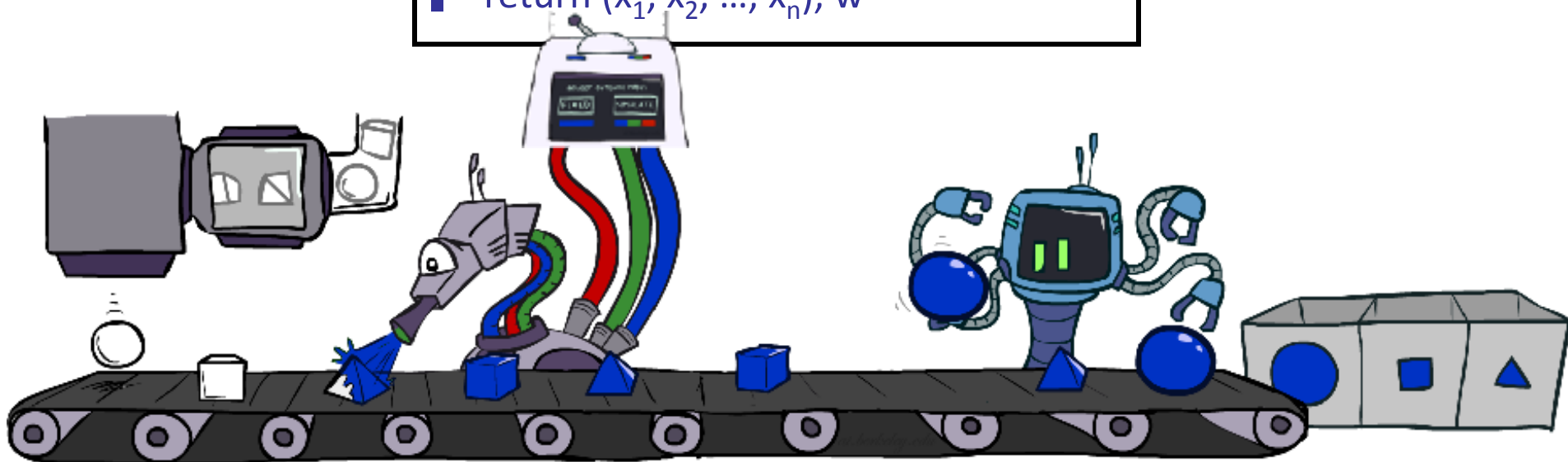
C: 0.1

D: 0.099

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

# Likelihood Weighting

- IN: evidence instantiation
- $w = 1.0$
- for  $i=1, 2, \dots, n$ 
  - if  $X_i$  is an evidence variable
    - $X_i = \text{observation } x_i \text{ for } X_i$
    - Set  $w = w * P(x_i | \text{Parents}(X_i))$
  - else
    - Sample  $x_i$  from  $P(X_i | \text{Parents}(X_i))$
- return  $(x_1, x_2, \dots, x_n), w$



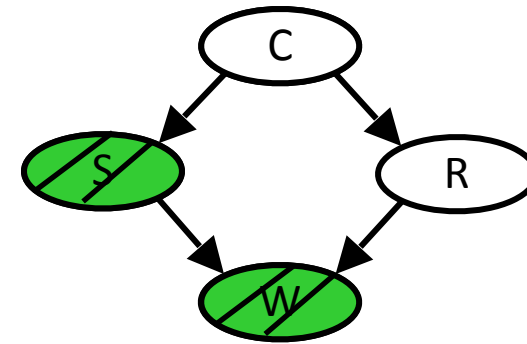
# Likelihood Weighting

- Sampling distribution if  $z$  sampled and  $e$  fixed evidence

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

- Now, samples have weights

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

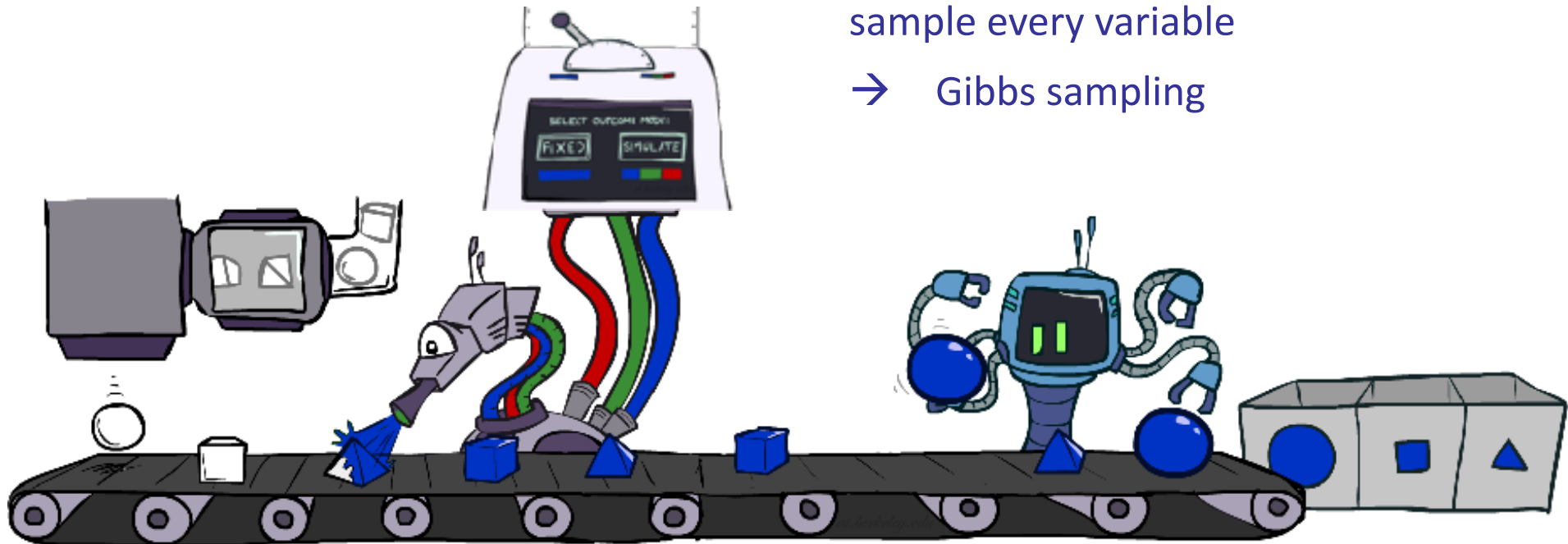


- Together, weighted sampling distribution is consistent

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

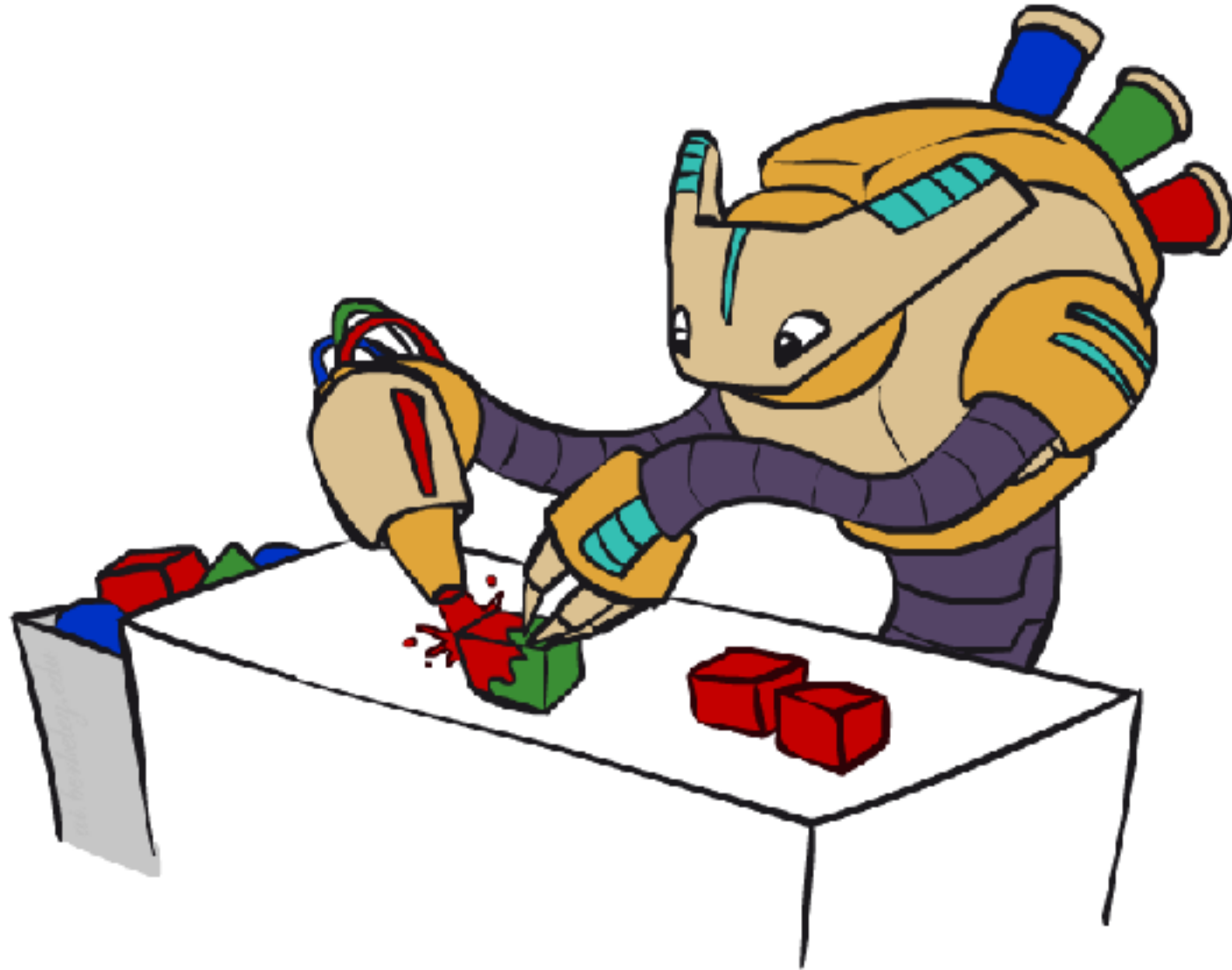
# Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - Our samples will reflect the state of the world suggested by the evidence
  - No need for rejection!
- Likelihood weighting doesn't solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones (not more likely to get a value matching the evidence)
  - Can cause many very small weights → inefficient!
- We would like to consider evidence when we sample every variable
  - Gibbs sampling





# Gibbs Sampling



# Gibbs Sampling

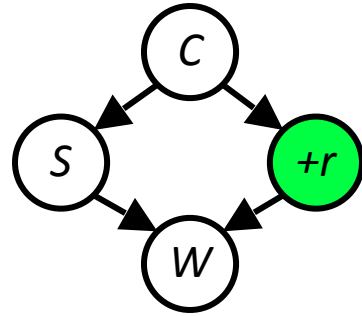
---

- *Procedure:* keep track of a full instantiation  $x_1, x_2, \dots, x_n$ . Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property:* in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution
- *Rationale:* both upstream and downstream variables condition on evidence.
- In contrast: likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small. Sum of weights over all samples is indicative of how many “effective” samples were obtained, so want high weight.

# Gibbs Sampling Example: $P(S \mid +r)$

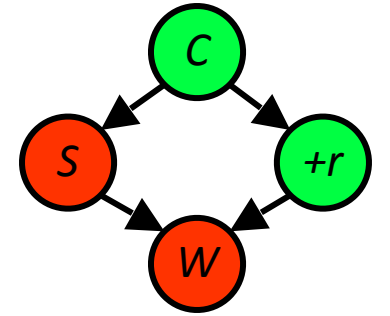
- Step 1: Fix evidence

- $R = +r$



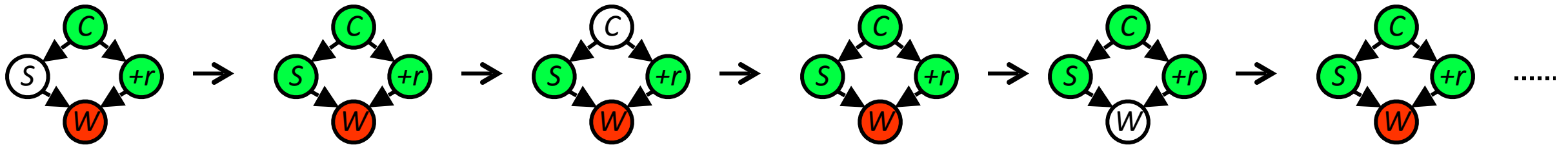
- Step 2: Initialize other variables

- Randomly



- Step 3: Repeat the following:

- Choose a non-evidence variable  $X$
- Resample  $X$  from  $P(X \mid \text{all other variables})$



Sample from  $P(S \mid +c, -w, +r)$

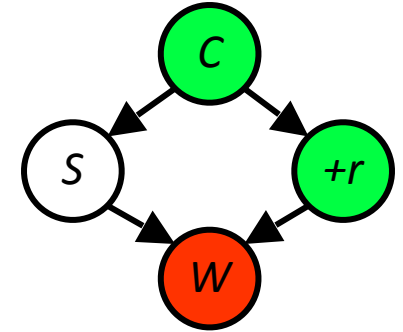
Sample from  $P(C \mid +s, -w, +r)$

Sample from  $P(W \mid +s, +c, +r)$

# Efficient Resampling of One Variable

- Sample from  $P(S \mid +c, +r, -w)$

$$\begin{aligned} P(S \mid +c, +r, -w) &= \frac{P(S, +c, +r, -w)}{P(+c, +r, -w)} \\ &= \frac{P(S, +c, +r, -w)}{\sum_s P(s, +c, +r, -w)} \\ &= \frac{P(+c)P(S \mid +c)P(+r \mid +c)P(-w \mid S, +r)}{\sum_s P(+c)P(s \mid +c)P(+r \mid +c)P(-w \mid s, +r)} \\ &= \frac{P(+c)P(S \mid +c)P(+r \mid +c)P(-w \mid S, +r)}{P(+c)P(+r \mid +c) \sum_s P(s \mid +c)P(-w \mid s, +r)} \\ &= \frac{P(S \mid +c)P(-w \mid S, +r)}{\sum_s P(s \mid +c)P(-w \mid s, +r)} \end{aligned}$$



- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

# Gibbs Sampling

---

- How is this better than sampling from the full joint?
  - In a Bayes Net, sampling a variable given all the other variables (e.g.  $P(R|S,C,W)$ ) is usually much easier than sampling from the full joint distribution
  - Only requires a join on the variable to be sampled (in this case, a join on R)

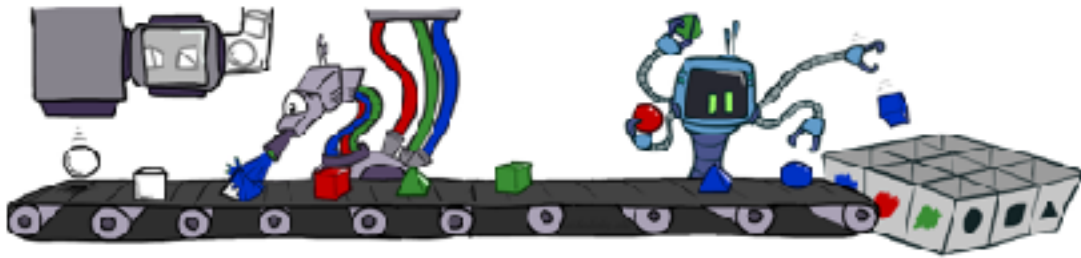
# Further Reading on Gibbs Sampling\*

---

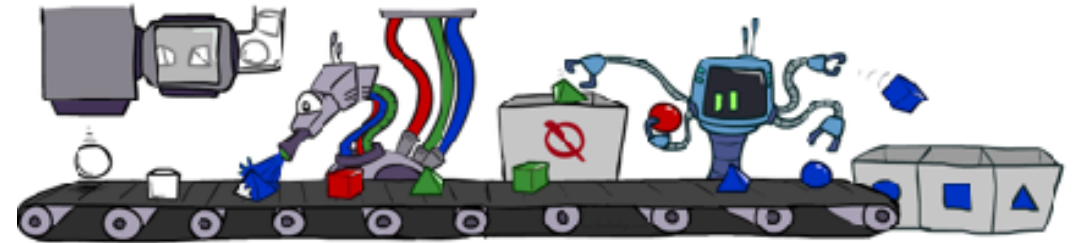
- Gibbs sampling produces sample from the query distribution  $P(Q | e)$  in limit of re-sampling infinitely often
- Gibbs sampling is a special case of more general methods called Markov chain Monte Carlo (MCMC) methods
  - Metropolis-Hastings is one of the more famous MCMC methods (in fact, Gibbs sampling is a special case of Metropolis-Hastings)
- You may read about Monte Carlo methods – they're just sampling

# Bayes Net Sampling Summary

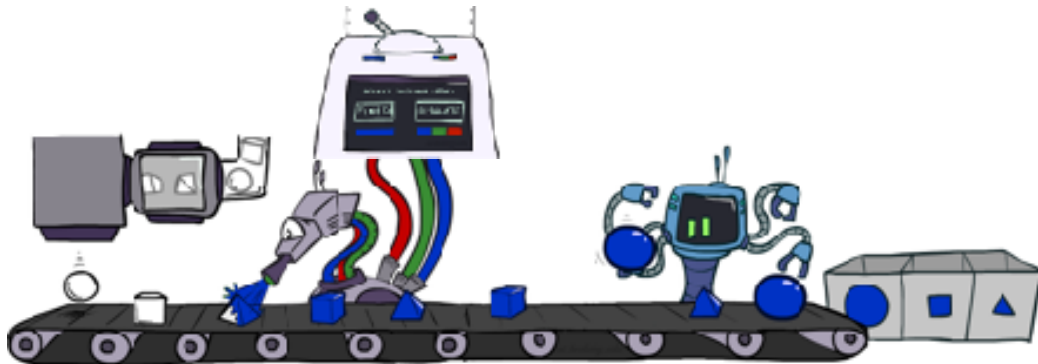
- Prior Sampling  $P$



- Rejection Sampling  $P(Q | e)$



- Likelihood Weighting  $P(Q | e)$



- Gibbs Sampling  $P(Q | e)$

