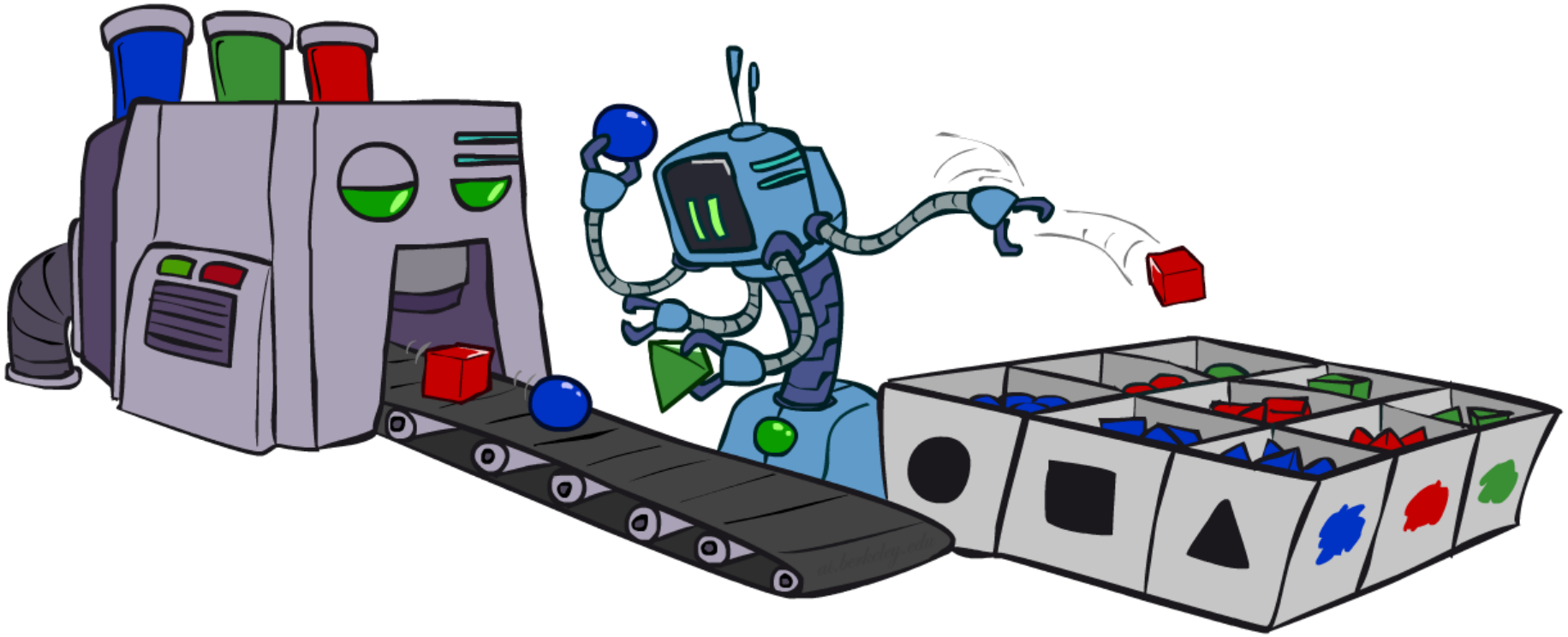
# CS 188: Artificial Intelligence

## Bayes' Nets: Sampling



Instructor: Oliver Grillmeyer --- University of California, Berkeley

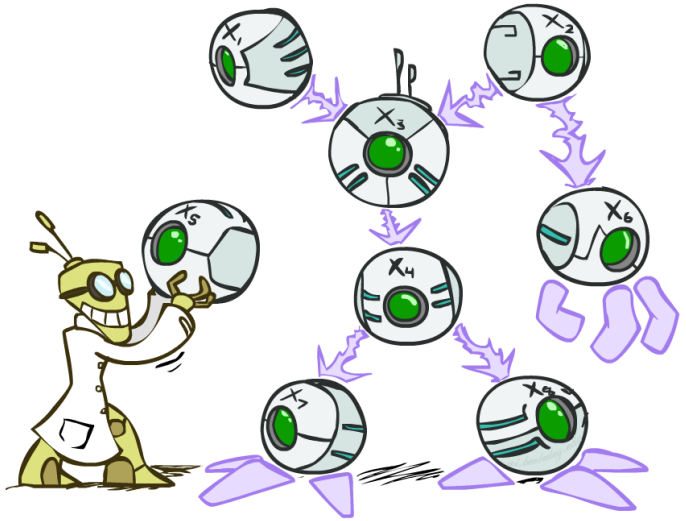
# Announcements

- HW6 is due **Thursday, July 17**, 11:59 PM PT
- Project 3 is due **Friday, July 18**, 11:59 PM PT
- Midterm is **Wednesday, July 23**, 7-9 PM PT in 155 Dwinelle
- Ignore assessment on HWs part B, but please show your work
- Email me topramen@berkeley.edu if you would attend MW 7-8 sections that focused on projects and homework

# Bayes' Net Representation

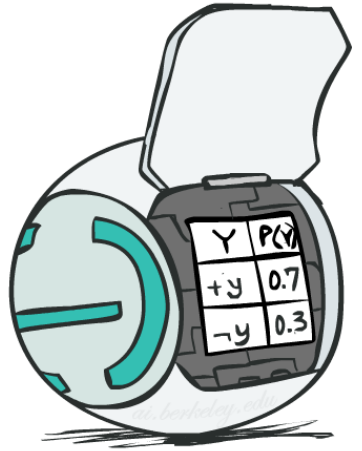
- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over X, one for each combination of parents' values

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



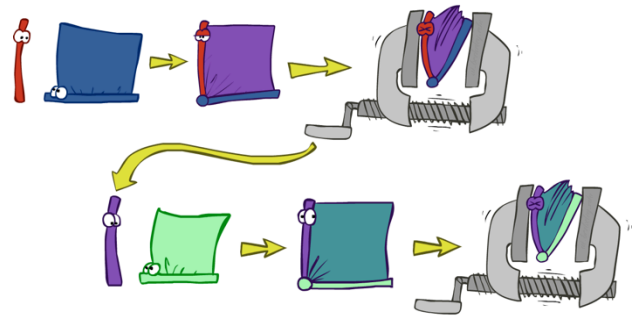

- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

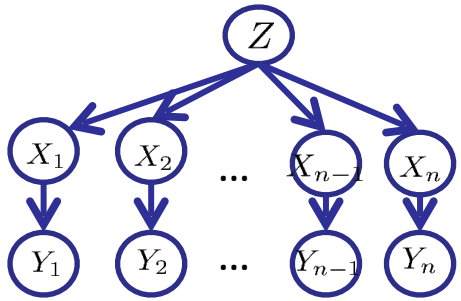
$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|parents(X_i))$$

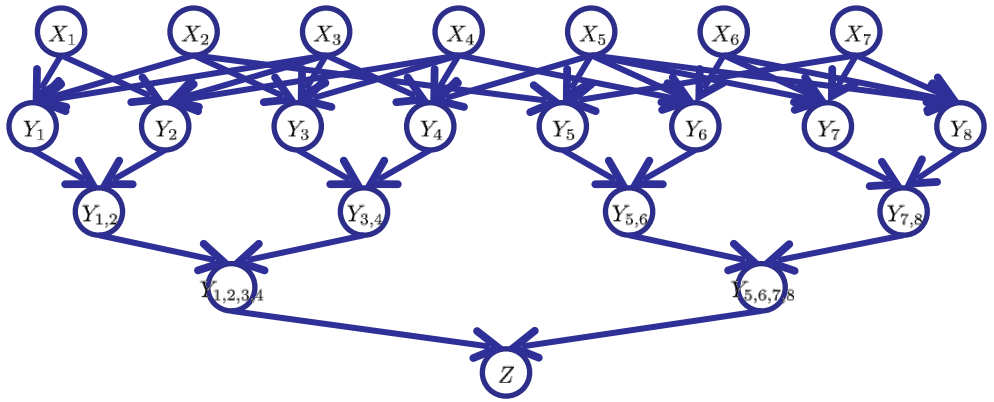


# Variable Elimination

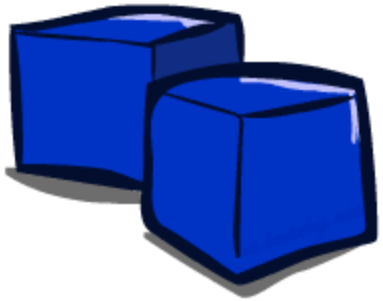
- Interleave joining and marginalizing
- $d^k$ entries computed for a factor over k variables with domain sizes d
- Ordering of elimination of hidden variables can affect size of factors generated
- Worst case: running time exponential in the size of the Bayes' net





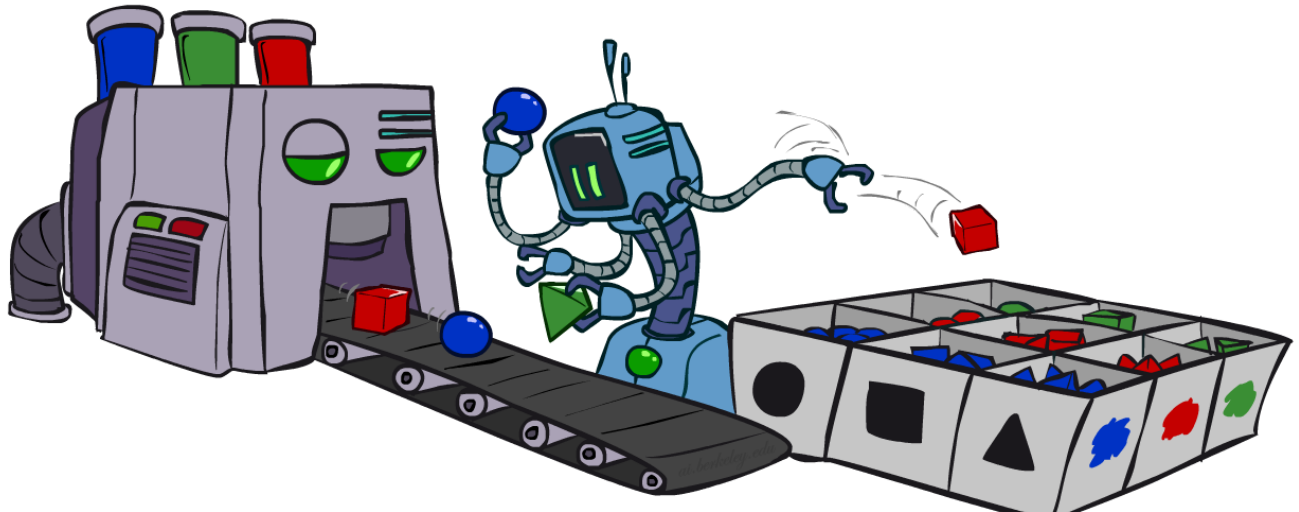



# Approximate Inference: Sampling



# Sampling

- Sampling is a lot like repeated simulation
  - Predicting the weather, basketball games, …
- 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 a sample is faster than computing the right answer (e.g. with variable elimination)



# Sampling

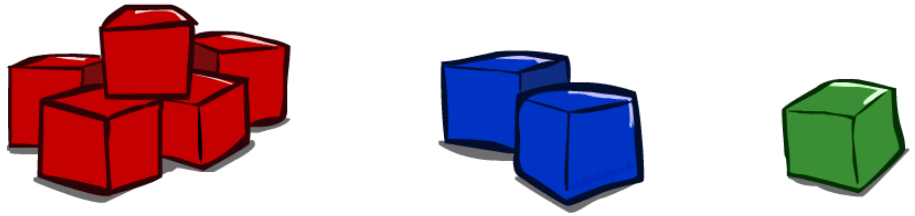
- Sampling from given distribution
  - Step 1: Get sample *u* from uniform distribution over [0, 1)
    - E.g. random() in python
  - Step 2: Convert this sample *u* into an outcome for the given distribution by having each target outcome associated with a sub-interval of [0,1) with sub-interval size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

- Example

| C | P(C) |
|---|---|
| red | 0.6 |
| green | 0.1 |
| blue | 0.3 |

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

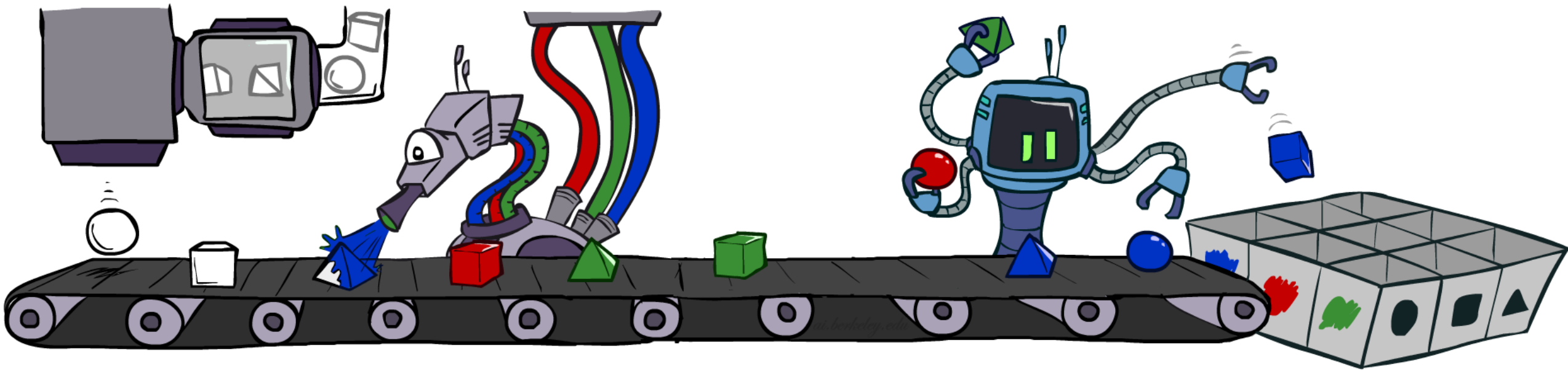
  - If random() returns $u = 0.83$, then our sample is $C$ = blue
  - E.g, after sampling 8 times:



# Sampling in Bayes' Nets

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

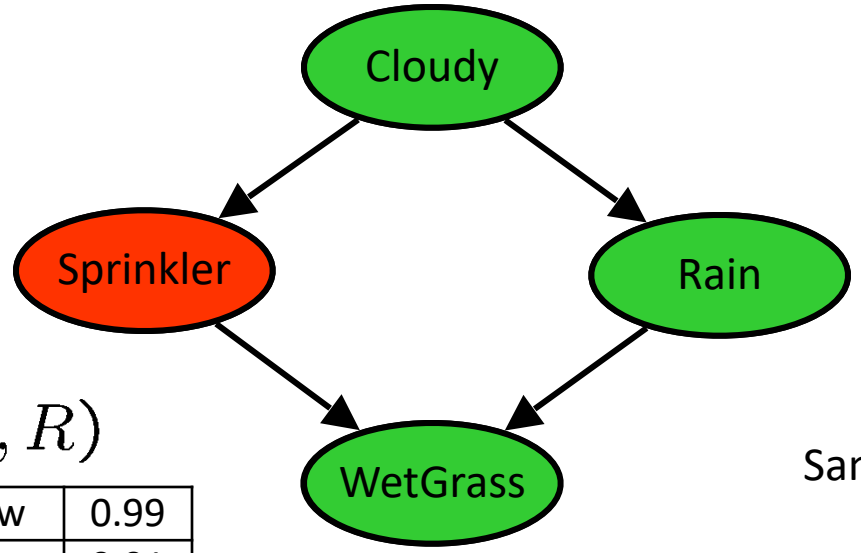
# Prior Sampling (no evidence)



# 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:

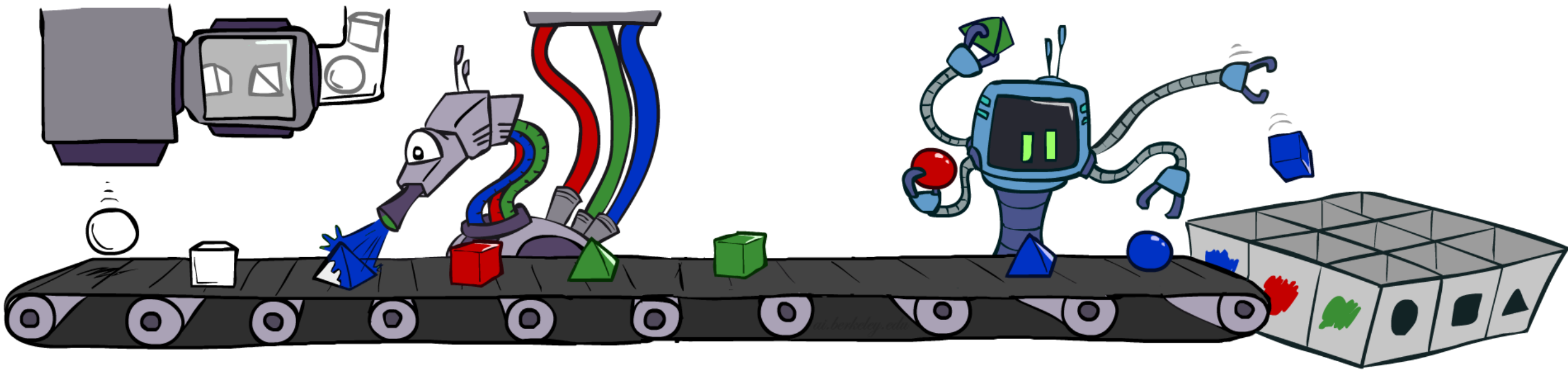
+c, -s, +r, +w

-c, +s, -r, +w

…

# Prior Sampling

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



# 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 event be $N_{PS}(x_1 \ldots x_n)$

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

- I.e., the sampling procedure is consistent

# Example

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

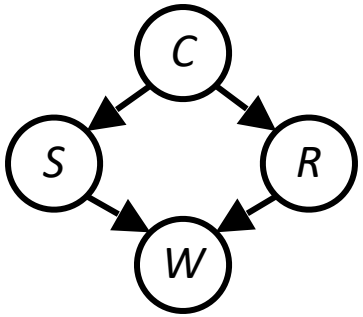
  +c, -s, +r, +w

  +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 <+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
  - Can estimate anything else, too
  - What about P(C | +w)? P(C | +r, +w)? P(C | -r, -w)?
  - Fast: can use fewer samples if less time (what's the drawback?)

# Reflections on Prior Sampling

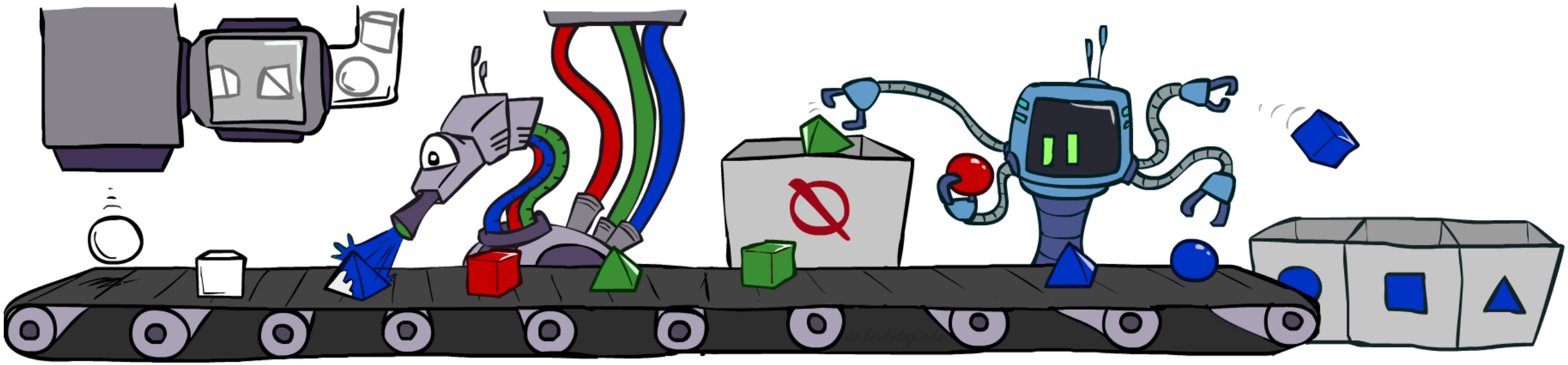
Pros:

- Much simpler than enumeration or variable elimination: We only ever need samples (scalar values) for each variable, not probabilities.
- Therefore we only ever need k rows from one CPT table to sample a variable $X$ that has k possible values. No potentially exponential increase with number of variables.
- Therefore it doesn't matter as much how sparse the graph is: we still only need k rows from each CPT table, regardless of how many rows (how many parents) it has.

Cons:

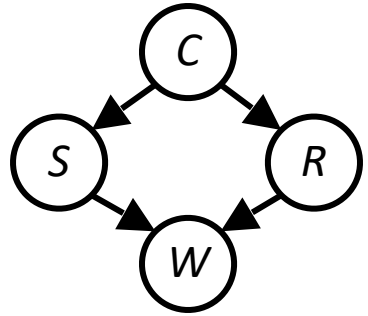
- So far we can't deal with evidence.
- We don't get exact values, and its expensive to get accurate estimates of small probabilities. E.g. estimating a 0.001 probability with 1% relative error requires around $10^7$ samples.

# Rejection Sampling



# Rejection Sampling

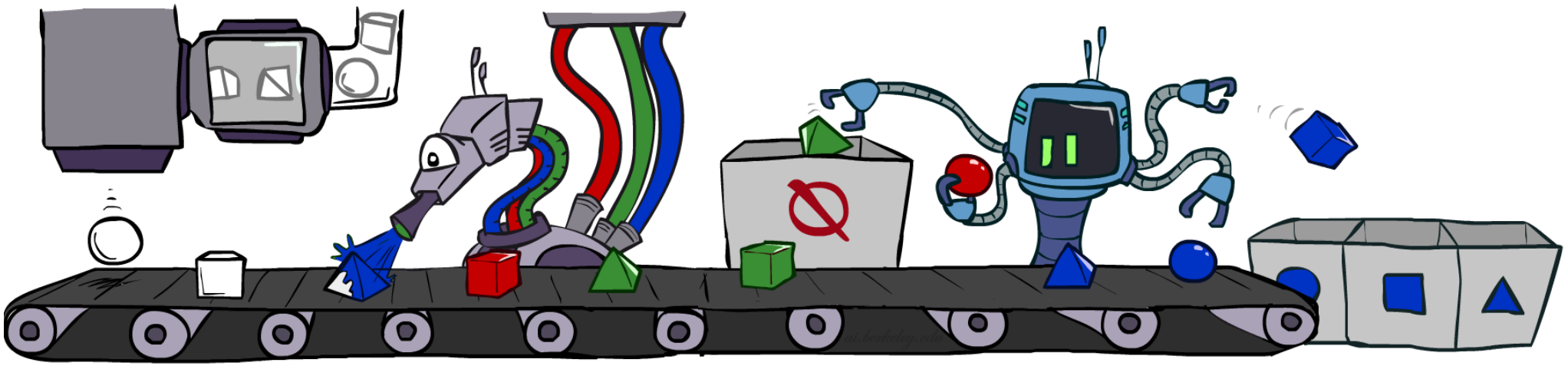
- Let's say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go
- Let's say we want P(C | +s)
  - Same thing: tally C outcomes, but ignore (reject) samples which don't have S=+s
  - This is called rejection sampling
  - It is also consistent for conditional probabilities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

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



# Reflections on Rejection Sampling

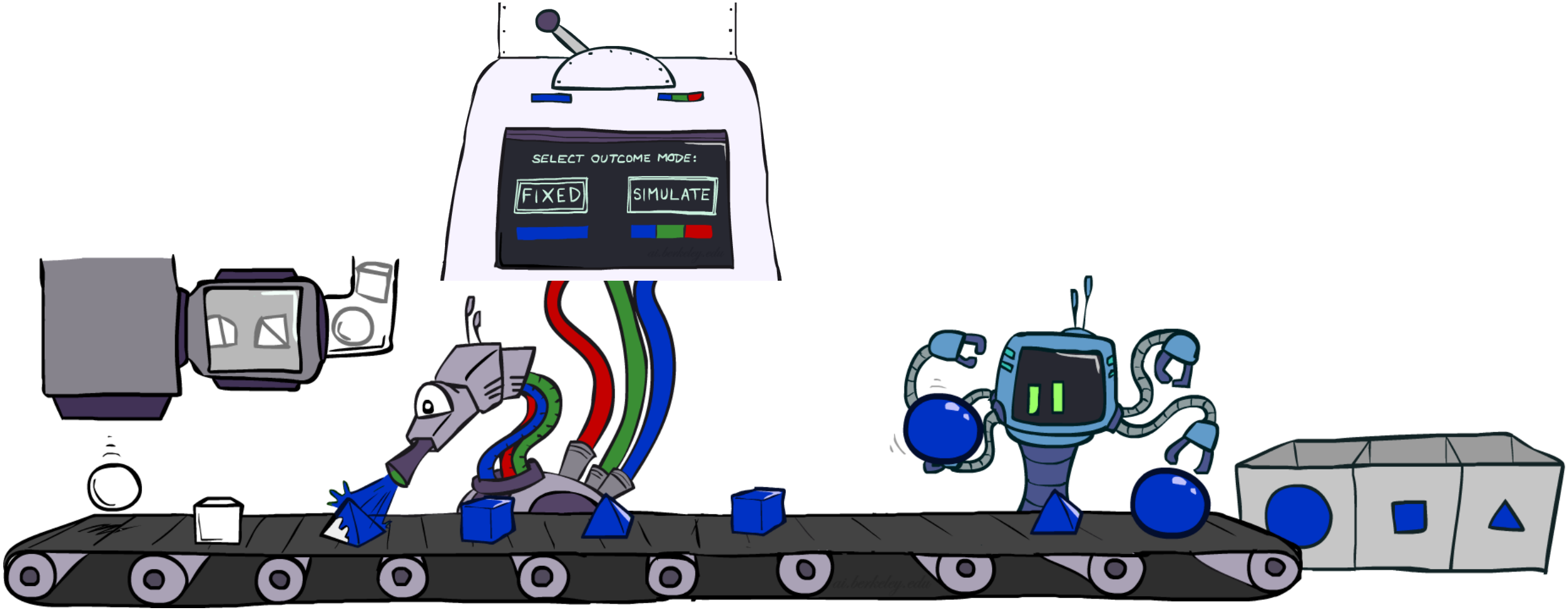
Pros:

- Inherits all the pros of prior sampling.
- Now we can deal with evidence.

Cons:

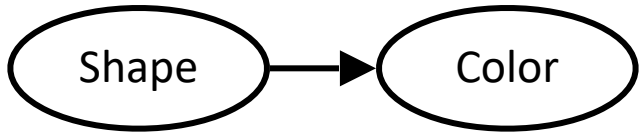
- Dealing with evidence can be very costly. Our rejection rate is 1- the marginal probability of the evidence, and getting N good samples requires taking N/p samples overall, where p is the marginal probability of the evidence.
- We still don't get exact values, and its still expensive to get accurate estimates of small probabilities. E.g. estimating a 0.001 probability with 1% relative error requires around $10^7$ samples, multiplied by 1/marginal probability of the evidence.

# Likelihood Weighting

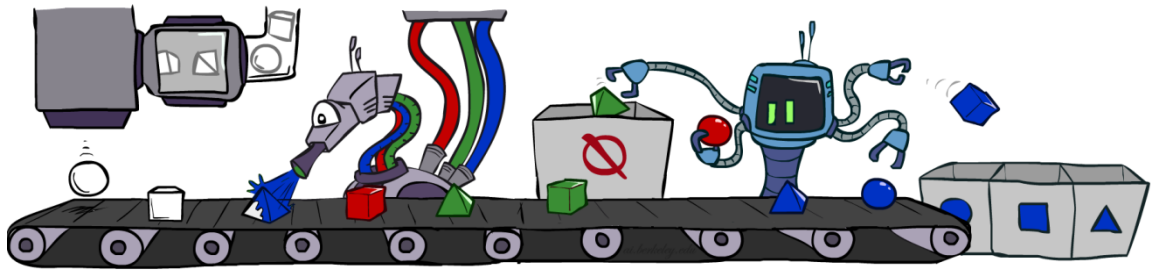


# Likelihood Weighting

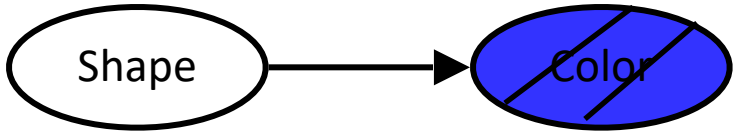
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P( Shape | blue )



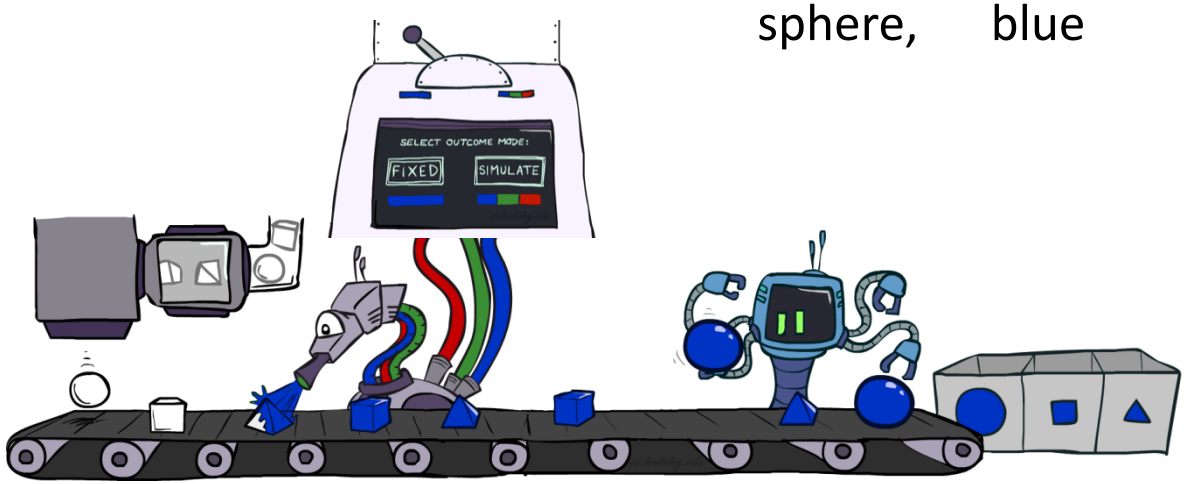

~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~



- 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



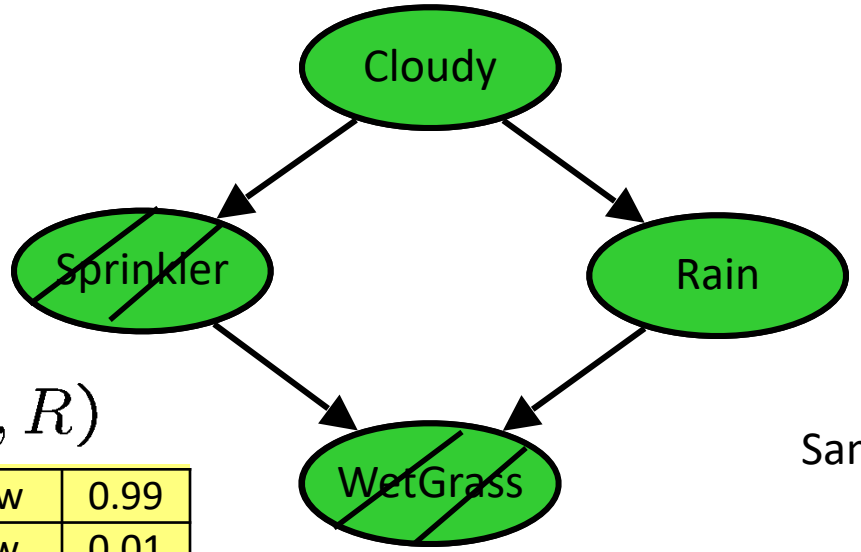
# Likelihood Weighting

$P(C)$

| | |
|---|---|
| +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:

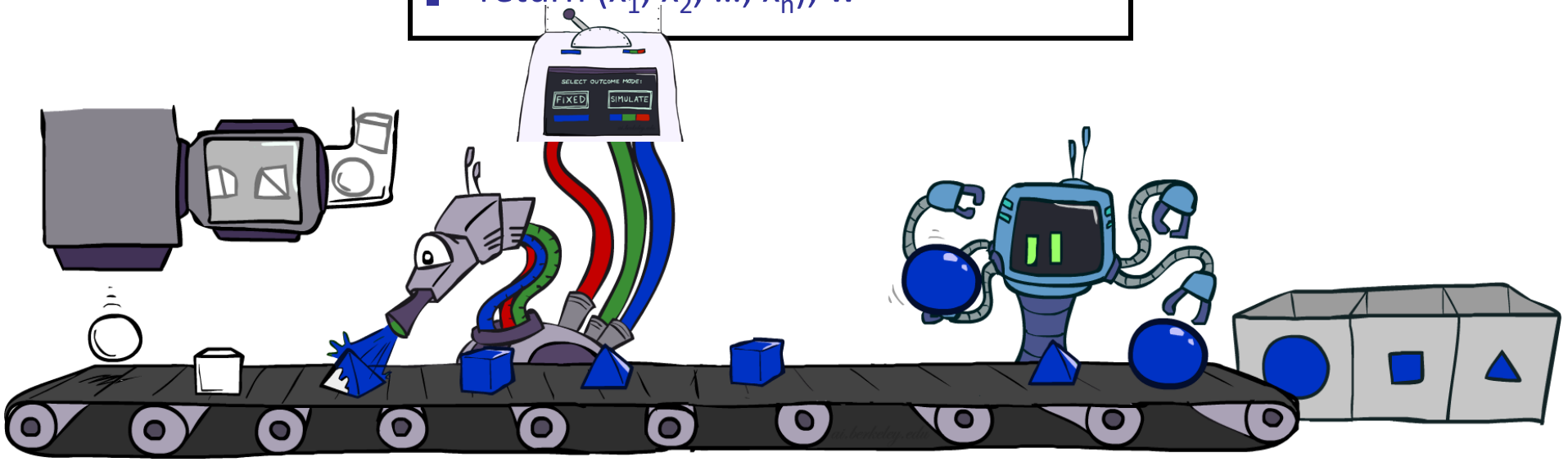
+c, +s, +r, +w

…

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

# Likelihood Weighting

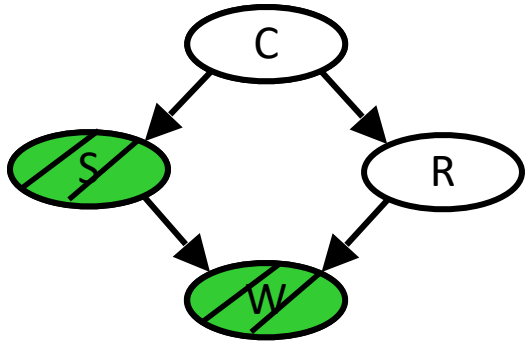
- Input: evidence instantiation
- w = 1.0
- for i = 1, 2, …, n
  - if $X_i$ is an evidence variable
    - $X_i$ = observation $x_i$ for $X_i$
    - Set w = w * $P(x_i \mid Parents(X_i))$
  - else
    - Sample $x_i$ from $P(X_i \mid Parents(X_i))$
- return $(x_1, x_2, \ldots, x_n)$, w



# Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

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




- Now, samples have weights

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

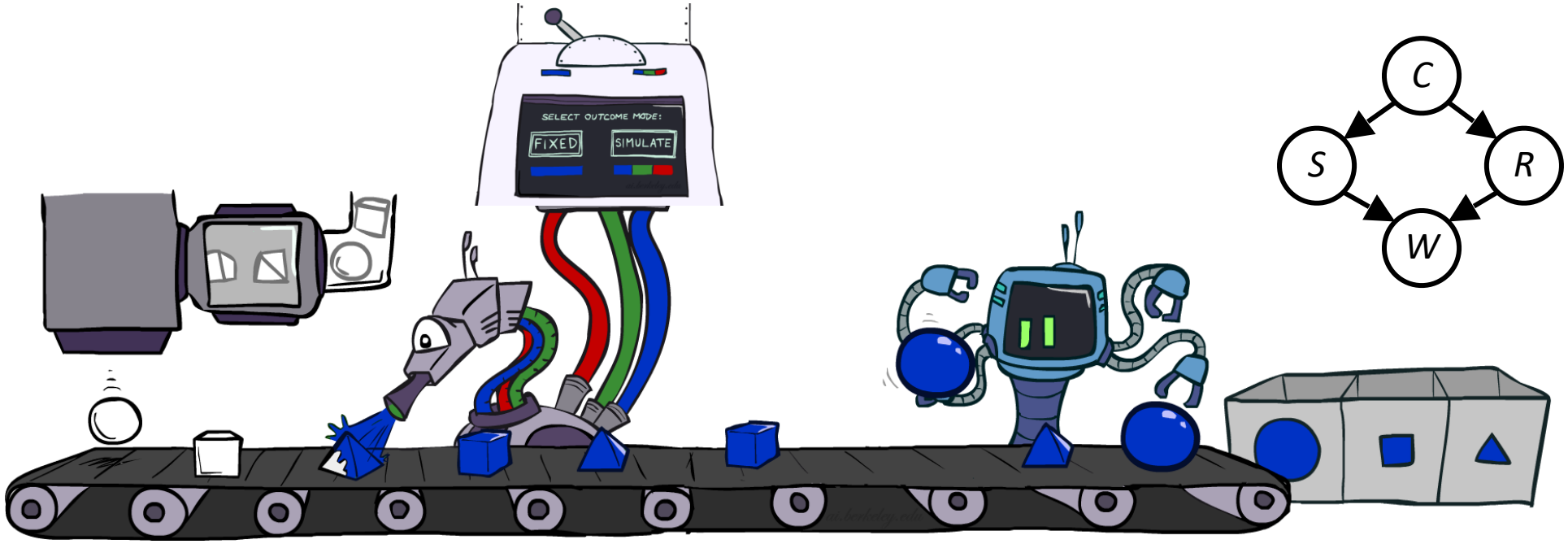
- Together, weighted sampling distribution is consistent

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

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

# Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence
- Likelihood weighting doesn't solve all our problems
  - Evidence influences the choice of downstream variables, but not upstream ones (C isn't more likely to get a value matching the evidence)
- We would like to consider evidence when we sample every variable (leads to Gibbs sampling)



# Reflections on Likelihood Weighting

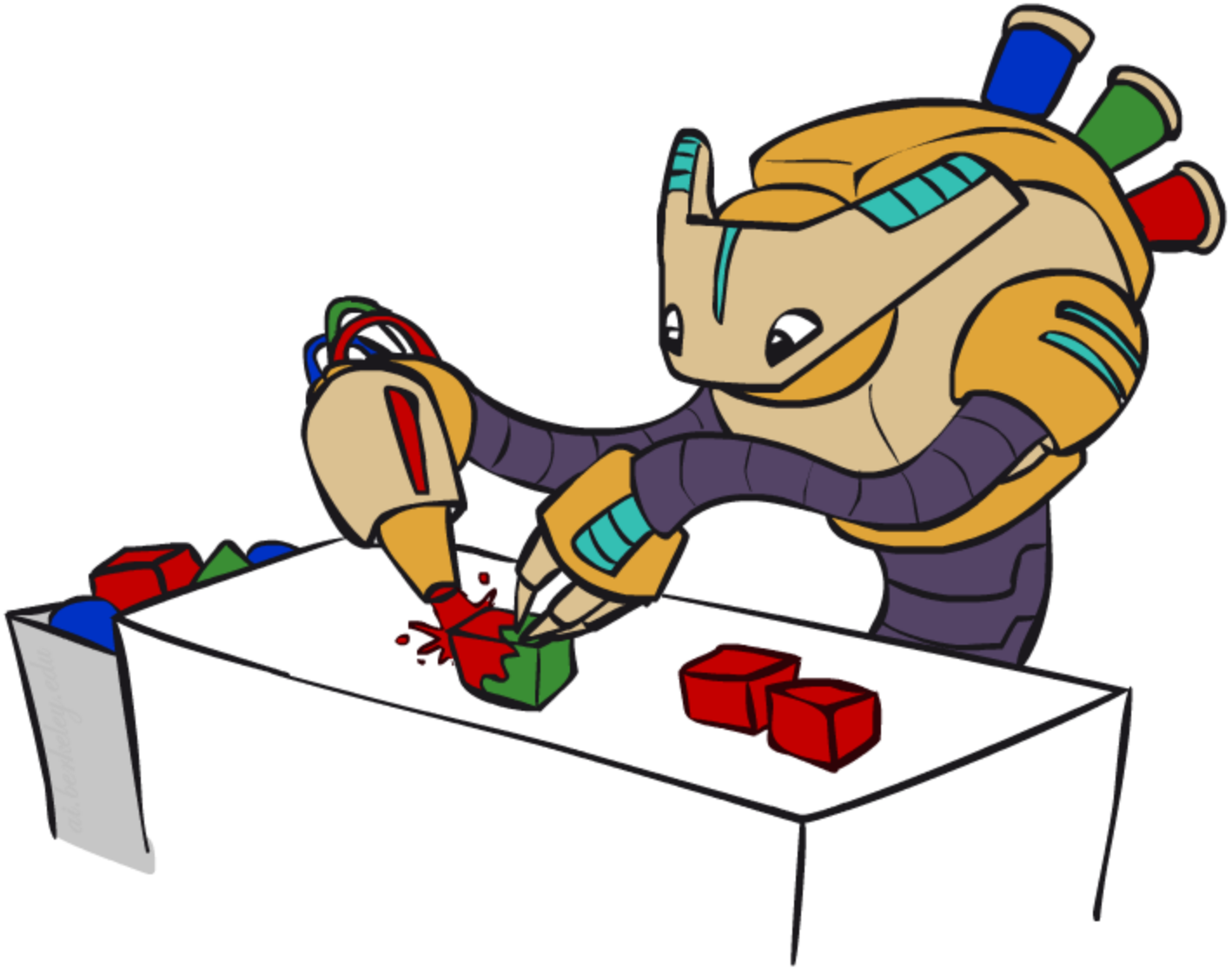
Pros:

- Inherits all the pros of Rejection Sampling, including dealing with evidence.

Cons:

- Dealing with evidence is more efficient, but can still be costly.
- We still don't get exact values, and it may still be expensive to get accurate estimates of small probabilities (although typically better than rejection sampling). We don't reject samples anymore, but if some samples have much higher weight than others, the variance of estimates we make from those samples will be dominated by the high-weight samples. In bad cases, this can be as inefficient as rejection sampling. Consider the Meningitis example…

# Gibbs Sampling

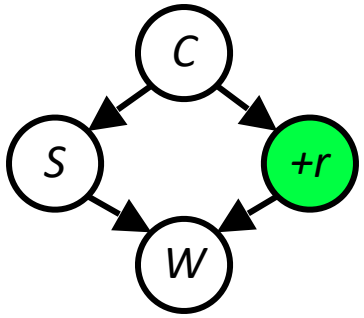


# Gibbs Sampling

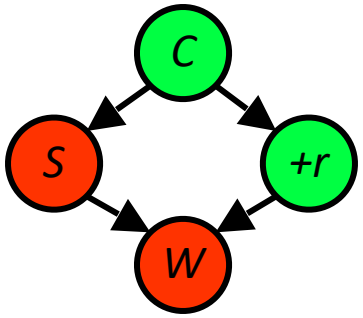
- *Procedure:* keep track of a full instantiation $x_1, x_2, \ldots, x_n$. Start with an arbitrary instantiation consistent with the evidence. Sample one variable at a time, conditioned on all the rest, but keep evidence fixed. Keep repeating this for a long time.
- *Property:* in the limit of repeating this infinitely many times the resulting samples come from the correct distribution (i.e. conditioned on evidence).
- *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 we want high weight.

# Gibbs Sampling Example: P( S | +r)

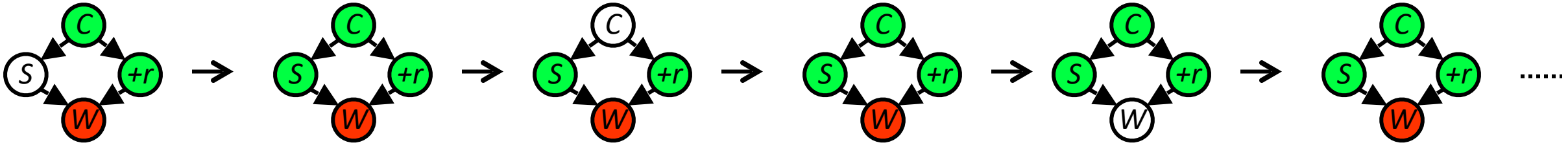
- Step 1: Fix evidence
  - R = +r




- Step 2: Initialize other variables
  - Randomly



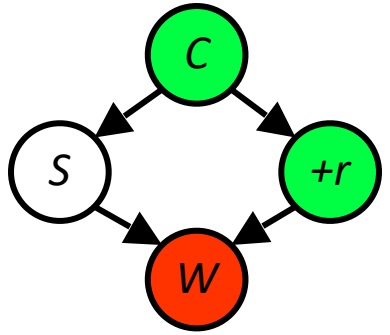

- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P( X | all other variables)




Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

# Efficient Resampling of One Variable

- Sample from P(S | +c, +r, -w)

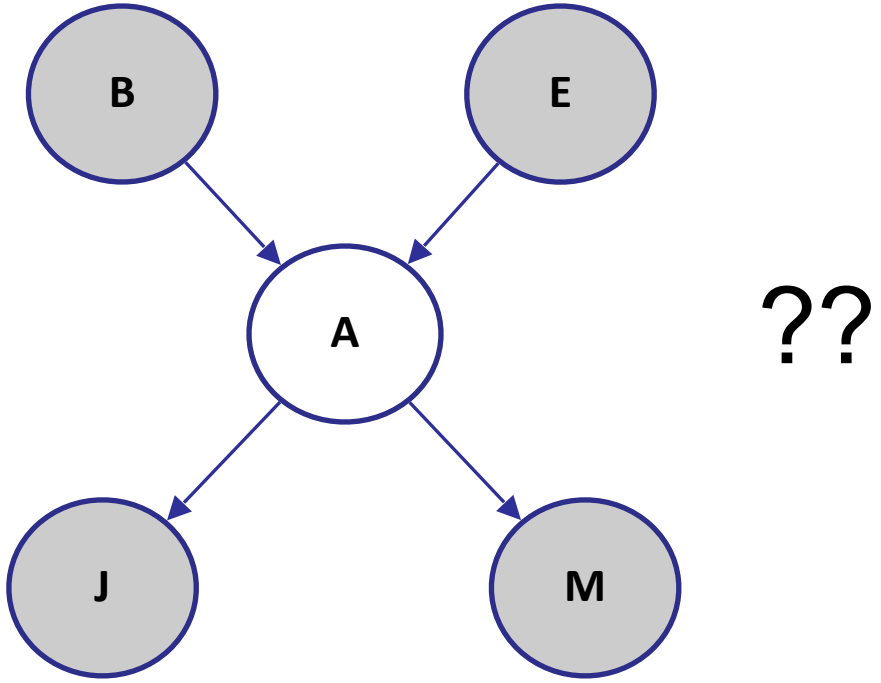



$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s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

- 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: Conditioning Variables

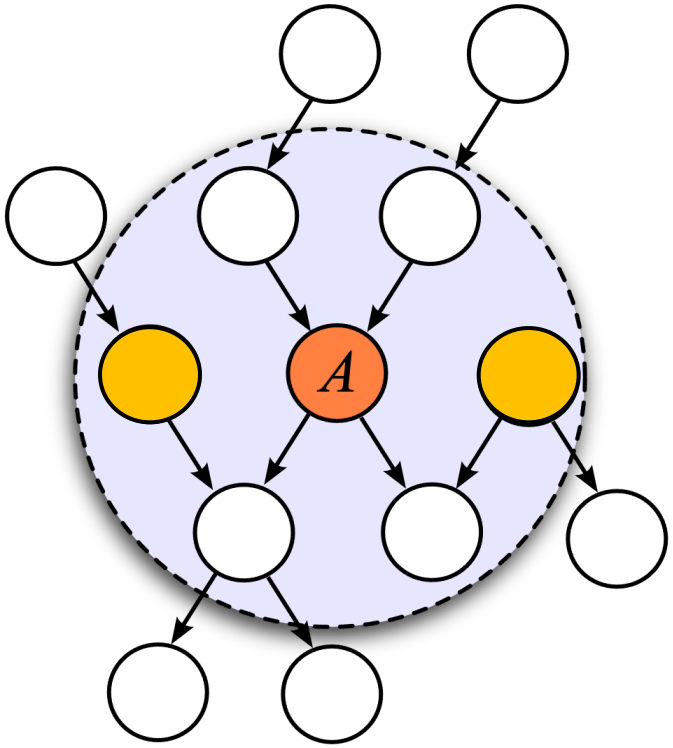
- Each node's probability is conditioned only on a subset of nearby nodes. We could start with the parents and children of the node to be sampled. Is that enough?




Note: a gray node here doesn't mean evidence, it just means that the node has been sampled and has a fixed value.

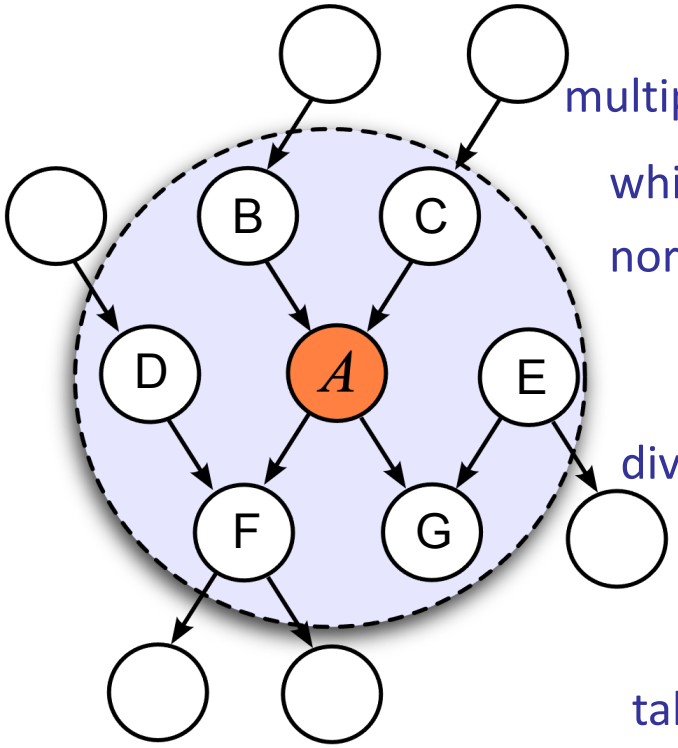
# Gibbs Sampling: Conditioning Variables

- No! Remember that the node equations for children depend on their other parents. The complete set is called the Markov Blanket of the node, and includes the other parents of the sampled node's children (orange):



# Gibbs Sampling: Node probability

- To sample, we first compute a factor that includes all node CPTs that depend on A.
- Then we normalize it to get a conditional probability for A.
- Finally we sample to get a new value for A.




multiply CPTs with A: $p(A \mid b, c)\, p(f \mid A, d)\, p(g \mid A, e)$

which is the factor: $p(A, f, g \mid b, c, d, e)$

normalize (sum over A and divide):

$$Z = \sum_a P(a, f, g \mid b, c, d, e)$$

divide:

$$p(A \mid b, c, d, e, f, g) = p(A, f, g \mid b, c, d, e)/Z$$

take a sample of A from this distribution.

# Reflections on Gibbs Sampling

Pros:

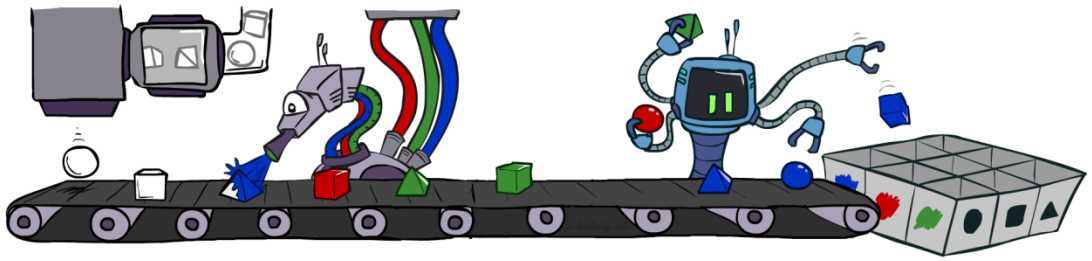
- Similar to other sampling methods: Only needs k rows from each CPT table for a variable X with k values.
- It also doesn't matter how sparse the graph is, in fact Gibbs sampling typically converges faster on denser graphs, because its *mixes* faster.
- Samples are unweighted, and come from the exact posterior probability conditioned on the evidence (eventually). So estimates can be fairly fast (modulo mixing).

Cons:

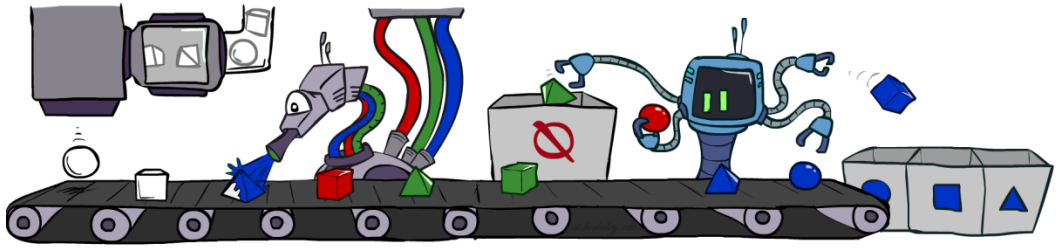
- There is a "warm-up" period for the sampler to reach the final distribution.
- Because samples are correlated, need more of them to get estimates at a given accuracy compared to other sampling methods.
- Both of the above depend on "mixing time," for which smaller is better.
- There is much theory and many techniques to improve Gibbs sampling.

# Bayes' Net Sampling Summary

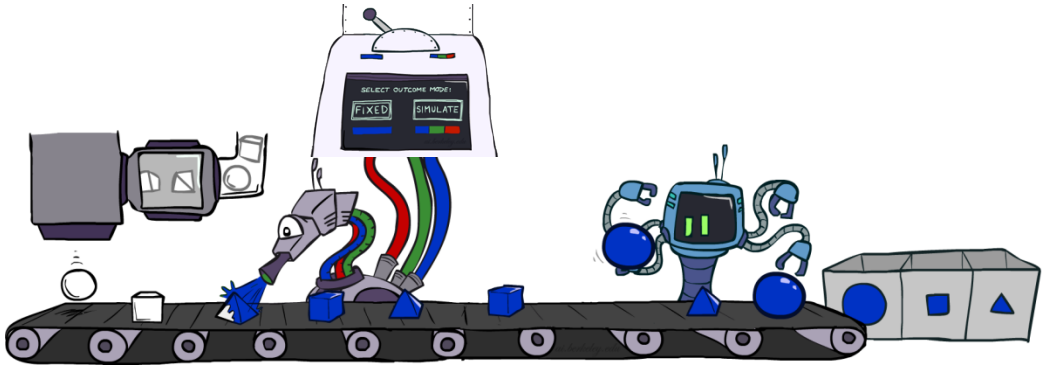
- Prior Sampling P( Q )



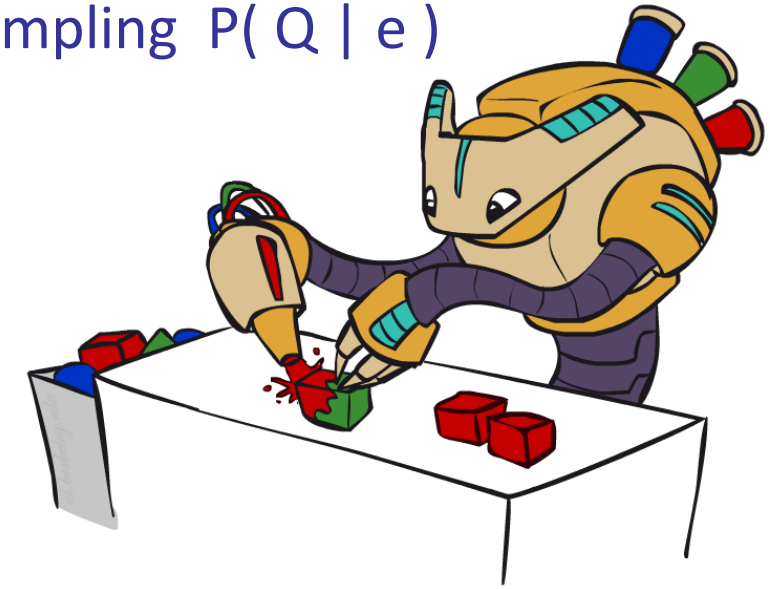
- Rejection Sampling P( Q | e )



- Likelihood Weighting P( Q | e)




- Gibbs Sampling P( Q | e )



# 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