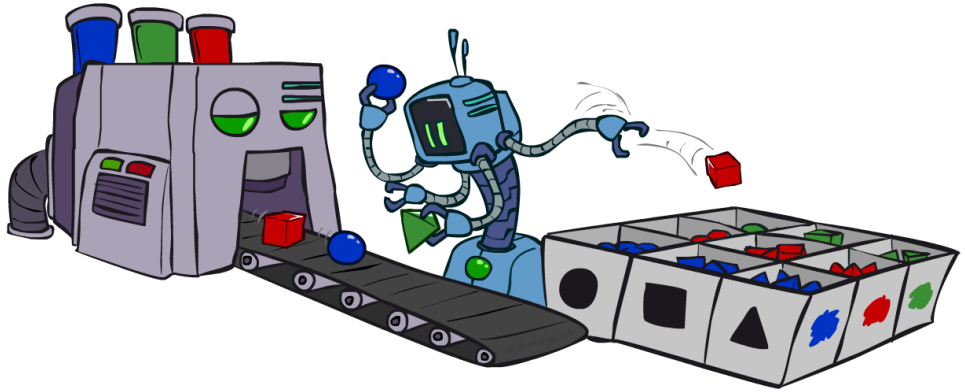
# Bayes Nets III: Sampling



CSE 473: Introduction to Artificial Intelligence

**James Weichert** (jpw@cs)
**cs.uw.edu/473**
acknowledgements



**Welcome!**
*What should the CSE 473 logo/mascot be?*
sli.do **#cse473**

# Administrivia

**ANNOUNCEMENTS**

- **Test 2** next Friday

**ASSIGNMENTS**

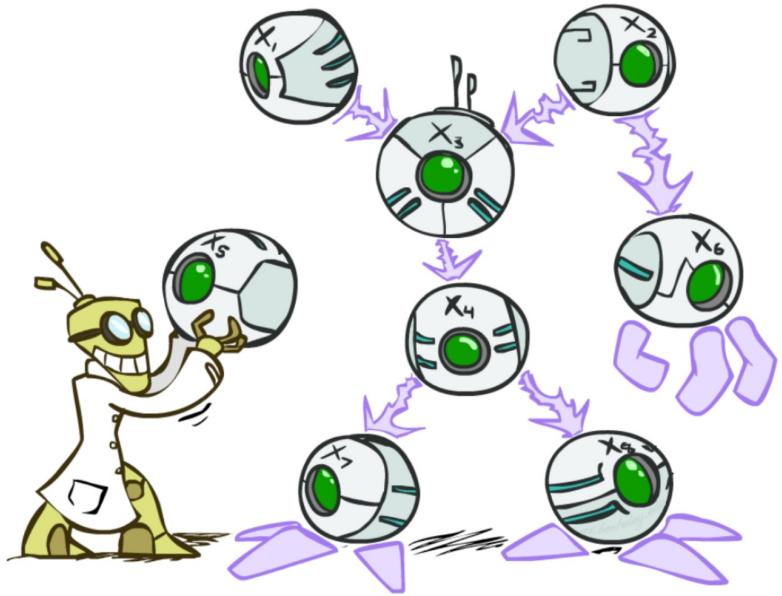
- **Project 3 due** this Thursday (7.30)
- **Homework 3 due** next Thursday (8.6)

# Bayes Nets

FOUNDATIONS

**Bayes Nets** provide a technique describing **complex joint distributions** using **simple conditional distributions**

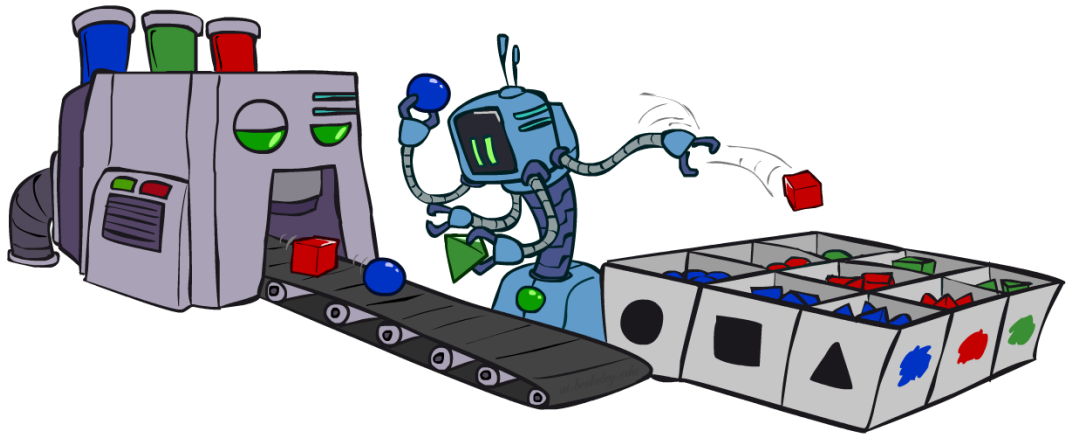
- Use local causality / conditional independence
- Trading expressivity of joint distributions for efficiency of conditional distributions



| **Representation** | **Inference** | **Sampling** |
|---|---|---|
| *Wednesday* | *Friday* | *Today* |

# Sampling

FOUNDATIONS



- Core Idea
  - Draw $N$ samples from sampling distribution $S$
  - Compute *approximate* posterior probability
  - Show this converges to true probability $P$
- Why Sample?
  - Often very fast to get a decent approximate answer
    - For **consistent** sampling methods, sample converges to true distribution
  - Algorithms are simple and general
  - Require little memory ($O(n)$)
  - Applicable to large models

# Sampling Scenario

EXAMPLE

Given two agent programs $\boldsymbol{A}$ and $\boldsymbol{B}$ that play the game Monopoly

- What is the probability that $\boldsymbol{A}$ wins?
  - Method 1:
    - Let $s$ be a sequence of dice rolls and card draws
    - Given $s$, the outcome $U(s)$ is computable (1 for a win, 0 for a loss)
    - $P(A \text{ wins}) = \sum_s P(s) \cdot U(s)$
    - **Problem**: infinitely many sequences $s$!
  - Method 2:
    - Sample $N$ sequences from $P(s)$ (play $N$ games)
    - Frequentist probability of A winning: $P(A \text{ wins}) \approx \frac{1}{N} \sum_i U(s_i)$

# Sampling Implementation

**ALGORITHM**

```
function RANDOM-SAMPLE(outcomes, probabilities) returns outcome
  r ← uniformly random number on [0, 1)
  cumulative ← 0
  for outcome, probability in outcomes, probabilities do
        cumulative ← cumulative + probability
        if r < cumulative then
            return outcome
```

QUESTIONS

# Questions?

live and on sli.do #cse473

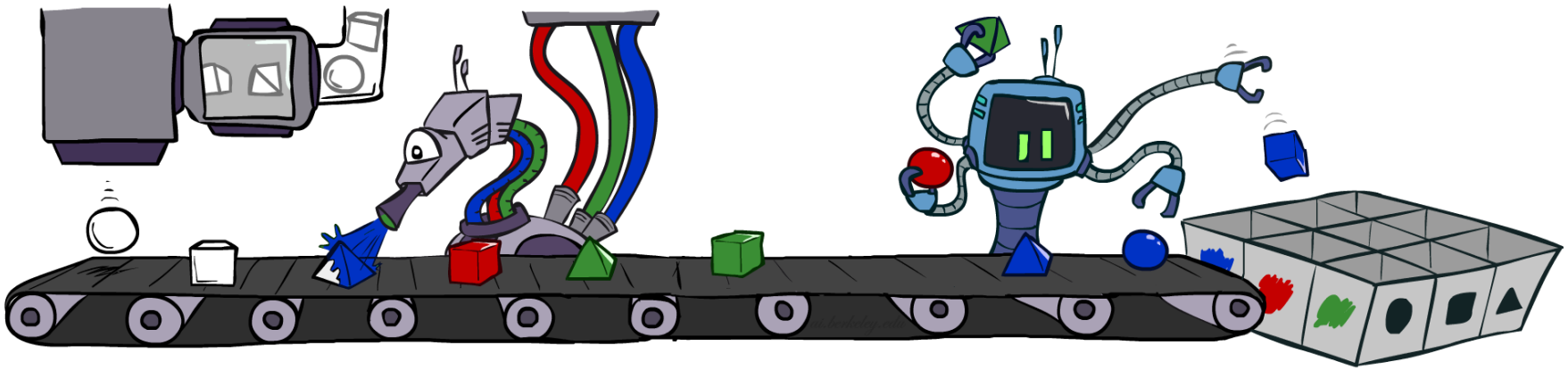
# Prior Sampling

FOUNDATIONS

- Work through network in topological order
- Sample $x_i$ from distribution $P\left(X_i \middle| x_{p_1}, \ldots, x_{p_k}\right)$
  - $x_{p_1}, \ldots, x_{p_k}$ are the sampled values for $\text{parents}(X_i)$

Exact Inference Analogy:

$$\boldsymbol{P(Q)}$$



# Prior Sampling Pseudocode

**ALGORITHM**

```
function PRIOR-SAMPLE(nodes) returns sample
  n ← NUM(nodes)
  sample ← ()
  for i = 1 to n in topological order do
        sample[i] ← sample from P(nodes[i] | sample[PARENTS(nodes[i])])
  return sample
```

# Prior Sampling Example (1)

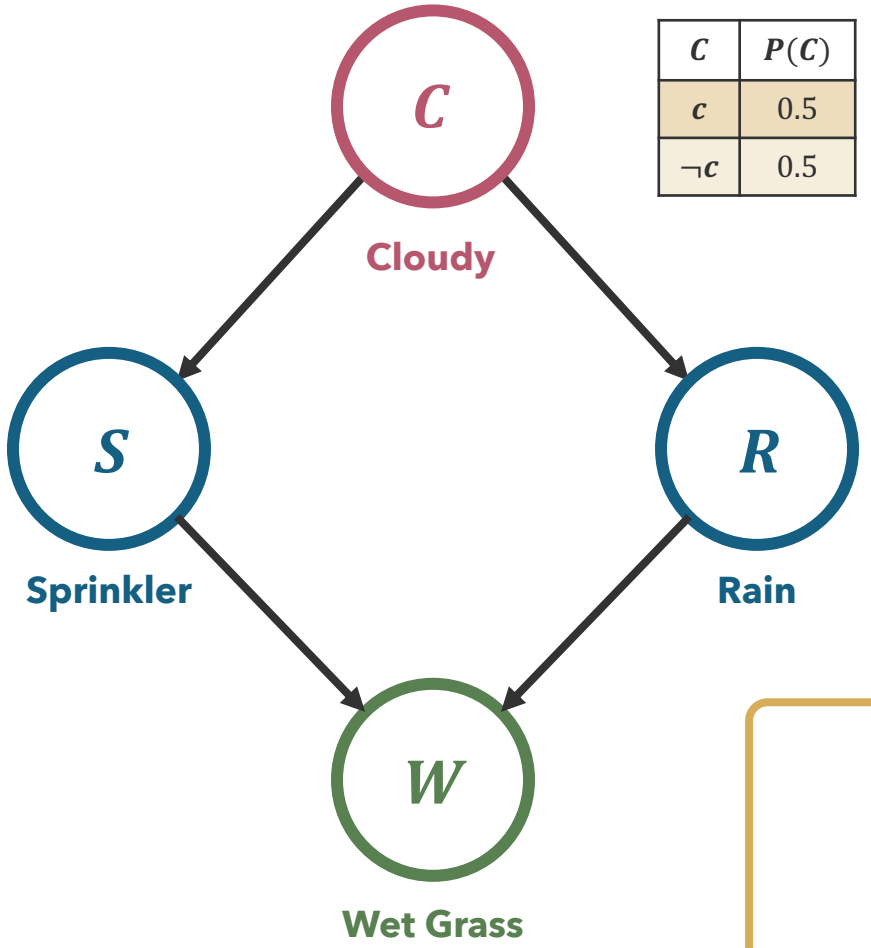
EXAMPLE

$P(S|C)$

| $C$ | $S$ | $P(S|C)$ |
|---|---|---|
| $c$ | $s$ | 0.1 |
| | $\neg s$ | 0.9 |
| $\neg c$ | $s$ | 0.5 |
| | $\neg s$ | 0.5 |

$P(W|S,R)$

| $S$ | $R$ | $W$ | $P(W|S,R)$ |
|---|---|---|---|
| $s$ | $r$ | $w$ | 0.99 |
| | | $\neg w$ | 0.01 |
| | $\neg r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| $\neg s$ | $r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| | $\neg r$ | $w$ | 0.01 |
| | | $\neg w$ | 0.99 |




$P(C)$

| $C$ | $P(C)$ |
|---|---|
| $c$ | 0.5 |
| $\neg c$ | 0.5 |

$P(R|C)$

| $C$ | $R$ | $P(R|C)$ |
|---|---|---|
| $c$ | $r$ | 0.8 |
| | $\neg r$ | 0.2 |
| $\neg c$ | $r$ | 0.2 |
| | $\neg r$ | 0.8 |

**Samples:**

(1) $c,\ \neg s,\ r,\ w$

# Prior Sampling Example (2)

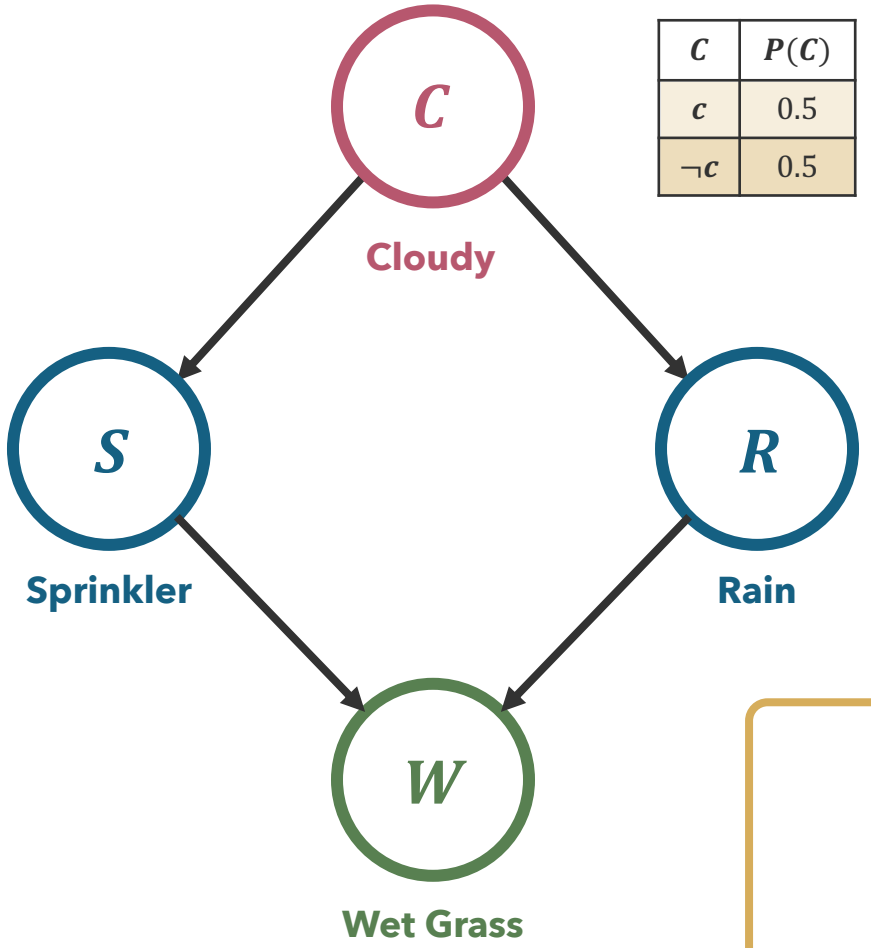
EXAMPLE

$P(S|C)$

| $C$ | $S$ | $P(S|C)$ |
|---|---|---|
| $c$ | $s$ | 0.1 |
| | $\neg s$ | 0.9 |
| $\neg c$ | $s$ | 0.5 |
| | $\neg s$ | 0.5 |

$P(C)$

| $C$ | $P(C)$ |
|---|---|
| $c$ | 0.5 |
| $\neg c$ | 0.5 |




$P(R|C)$

| $C$ | $R$ | $P(R|C)$ |
|---|---|---|
| $c$ | $r$ | 0.8 |
| | $\neg r$ | 0.2 |
| $\neg c$ | $r$ | 0.2 |
| | $\neg r$ | 0.8 |

$P(W|S,R)$

| $S$ | $R$ | $W$ | $P(W|S,R)$ |
|---|---|---|---|
| $s$ | $r$ | $w$ | 0.99 |
| | | $\neg w$ | 0.01 |
| | $\neg r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| $\neg s$ | $r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| | $\neg r$ | $w$ | 0.01 |
| | | $\neg w$ | 0.99 |

**Samples:**

(1) $c,\ \neg s,\ r,\ w$

(2) $\neg c,\ s,\ \neg r,\ w$

# Prior Sampling Properties

DEFINITION

Generates samples randomly from the joint distribution

$$S_{\text{prior}}(x_1, \ldots, x_n) = \prod_i P(X_i | \text{parents}(X_i)) = P(x_1, \ldots, x_n)$$

- Let the number of samples of an event $(x_1, \ldots, x_n)$ be $N_{\text{prior}}(x_1, \ldots, x_n)$
- Estimate from $N$ samples is $Q_N(x_1, \ldots, x_n) = \frac{1}{N} \cdot N_{\text{prior}}(x_1, \ldots, x_n)$
- Convergence: $\lim_{N \to \infty} Q_N(x_1, \ldots, x_n) = S_{\text{prior}}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$
  - Therefore, prior sampling is **consistent**

QUESTIONS

# Questions?

live and on sli.do #cse473

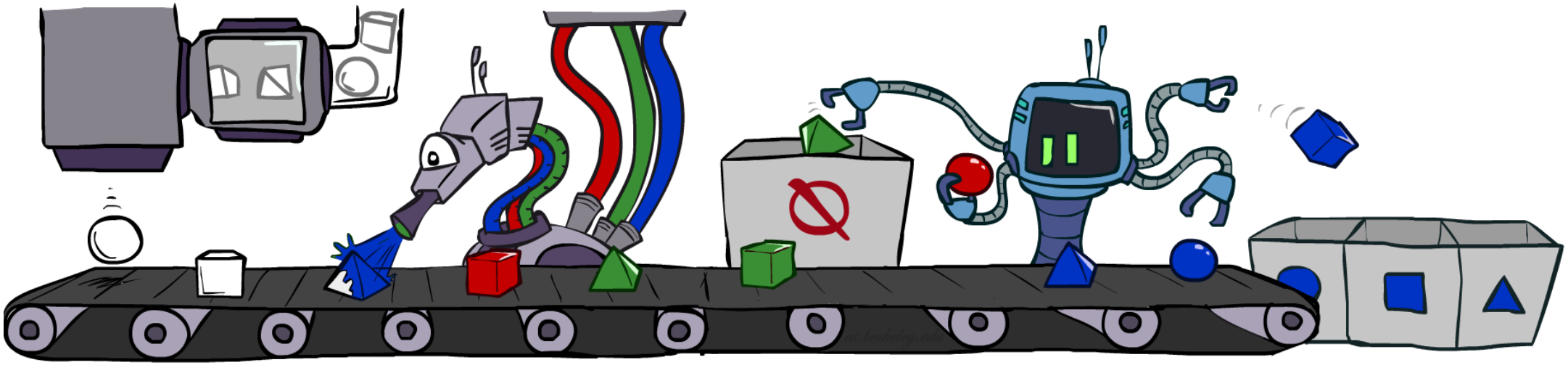
# Rejection Sampling

FOUNDATIONS

- Given evidence $e_1, \ldots, e_k$
  - Only count samples consistent with the evidence
  - But if evidence is unlikely, reject most samples…

Exact Inference Analogy:

$$P(Q|e_1, \ldots, e_k)$$



# Rejection Sampling Pseudocode

ALGORITHM

```
function REJECTION-SAMPLE(nodes, evidence) returns sample or failure
  n ← NUM(nodes)
  sample ← ()
  for i = 1 to n in topological order do
        x_i ← sample from P(nodes[i] | sample[PARENTS(nodes[i])])
        if x_i not consistent with evidence then
            return without sample
        else sample[i] ← x_i
  return sample
```

# Rejection Sampling Example

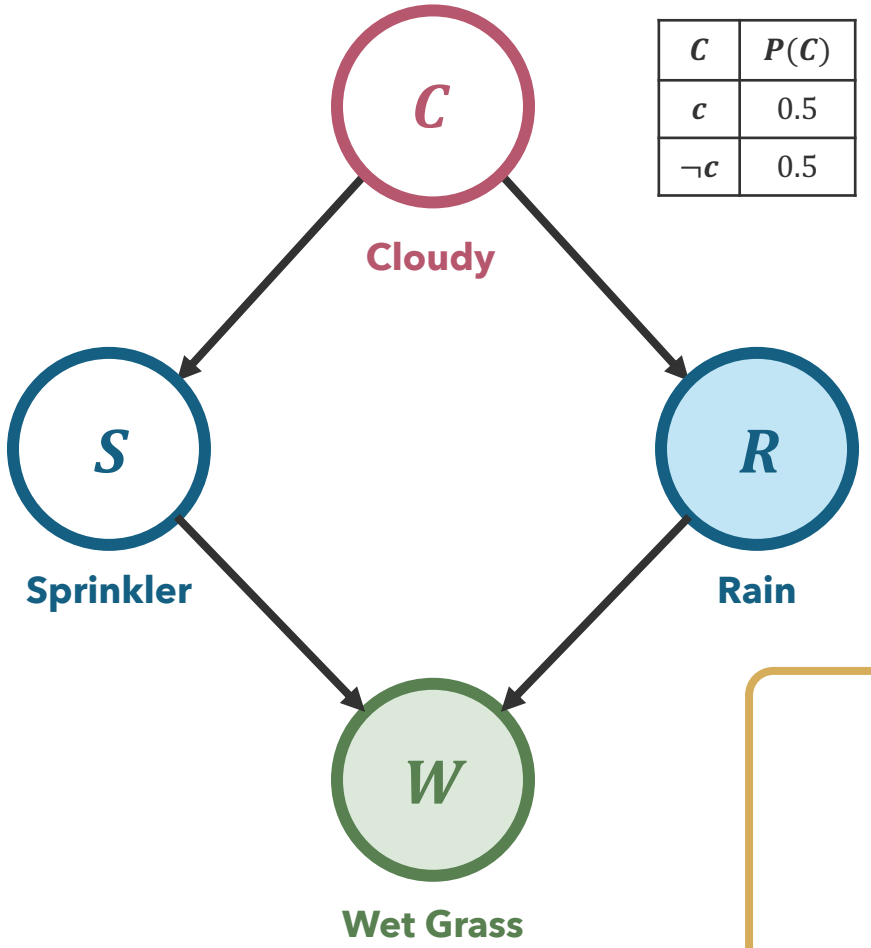
EXAMPLE

$P(S|C)$

| $C$ | $S$ | $P(S|C)$ |
|---|---|---|
| $c$ | $s$ | 0.1 |
| | $\neg s$ | 0.9 |
| $\neg c$ | $s$ | 0.5 |
| | $\neg s$ | 0.5 |

$P(C)$

| $C$ | $P(C)$ |
|---|---|
| $c$ | 0.5 |
| $\neg c$ | 0.5 |




$P(R|C)$

| $C$ | $R$ | $P(R|C)$ |
|---|---|---|
| $c$ | $r$ | 0.8 |
| | $\neg r$ | 0.2 |
| $\neg c$ | $r$ | 0.2 |
| | $\neg r$ | 0.8 |

$P(W|S,R)$

| $S$ | $R$ | $W$ | $P(W|S,R)$ |
|---|---|---|---|
| $s$ | $r$ | $w$ | 0.99 |
| | | $\neg w$ | 0.01 |
| | $\neg r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| $\neg s$ | $r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| | $\neg r$ | $w$ | 0.01 |
| | | $\neg w$ | 0.99 |

**Samples:**

✅ $c,\ \neg s,\ r,\ w$

❌ $c,\ s,\ \neg r, \ldots$

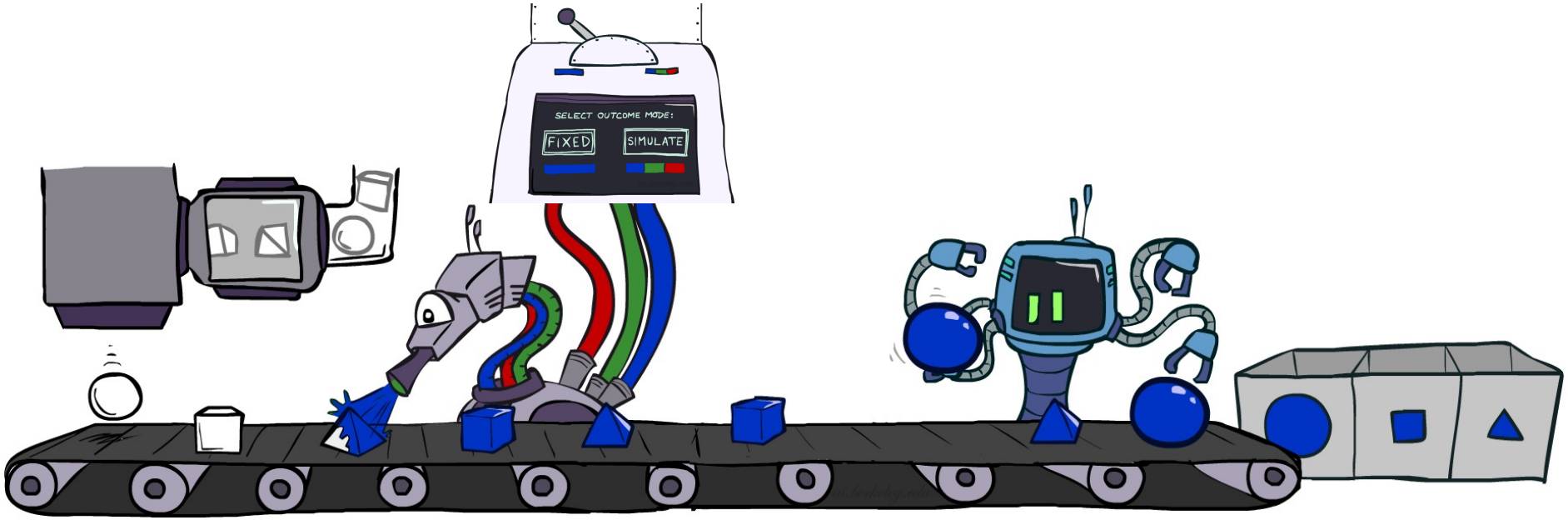
❌ $\neg c,\ \neg s,\ r, \neg w$

✅ $\neg c,\ s,\ r,\ w$

# Likelihood Weighting

FOUNDATIONS

- Improve on rejection sampling by fixing evidence
  - **Problem**: now the sampling distribution is inconsistent (does not converge to true distribution)
  - **Solution**: weight each sample by the probability of evidence given parents



# Likelihood Weighting Pseudocode

**ALGORITHM**

```
function LIKELIHOOD-WEIGHTING(nodes, evidence) returns sample, weight
  n ← NUM(nodes)
  w ← 1.0
  sample ← ()
  for i = 1 to n in topological order do
        if nodes[i] in evidence then
            sample[i] ← evidence[i]
            w ← w * P(sample[i] | sample[PARENTS(nodes[i])])
        else
            sample[i] ← sample from P(nodes[i] | sample[PARENTS(nodes[i])])
return sample, w
```

# Likelihood Weighting: $P(C, R|s, w)$

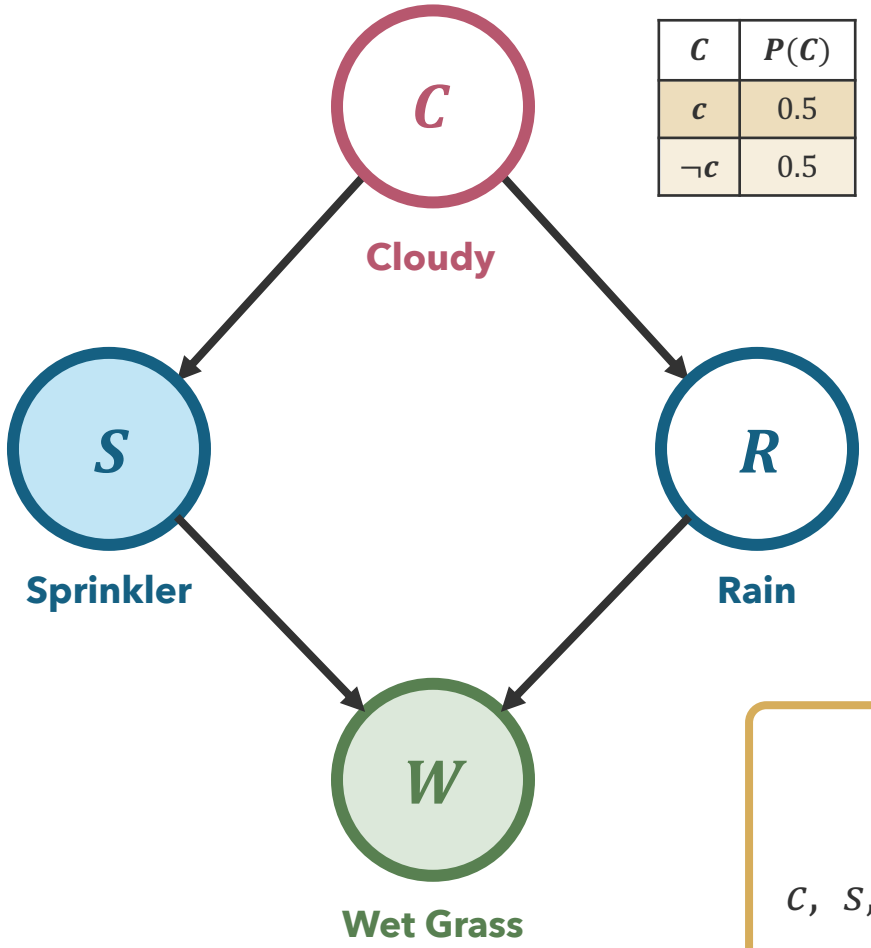
EXAMPLE

$P(S|C)$

| $C$ | $S$ | $P(S|C)$ |
|---|---|---|
| $c$ | $s$ | 0.1 |
| | $\neg s$ | 0.9 |
| $\neg c$ | $s$ | 0.5 |
| | $\neg s$ | 0.5 |

$P(W|S, R)$

| $S$ | $R$ | $W$ | $P(W|S, R)$ |
|---|---|---|---|
| $s$ | $r$ | $w$ | 0.99 |
| | | $\neg w$ | 0.01 |
| | $\neg r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| $\neg s$ | $r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| | $\neg r$ | $w$ | 0.01 |
| | | $\neg w$ | 0.99 |




$P(C)$

| $C$ | $P(C)$ |
|---|---|
| $c$ | 0.5 |
| $\neg c$ | 0.5 |

$P(R|C)$

| $C$ | $R$ | $P(R|C)$ |
|---|---|---|
| $c$ | $r$ | 0.8 |
| | $\neg r$ | 0.2 |
| $\neg c$ | $r$ | 0.2 |
| | $\neg r$ | 0.8 |

**Sample:**

$c,\ s,\ r,\ w$ weight $= 0.1 \cdot 0.99$

# Likelihood Weighting: $P(C, R|s, \neg w)$

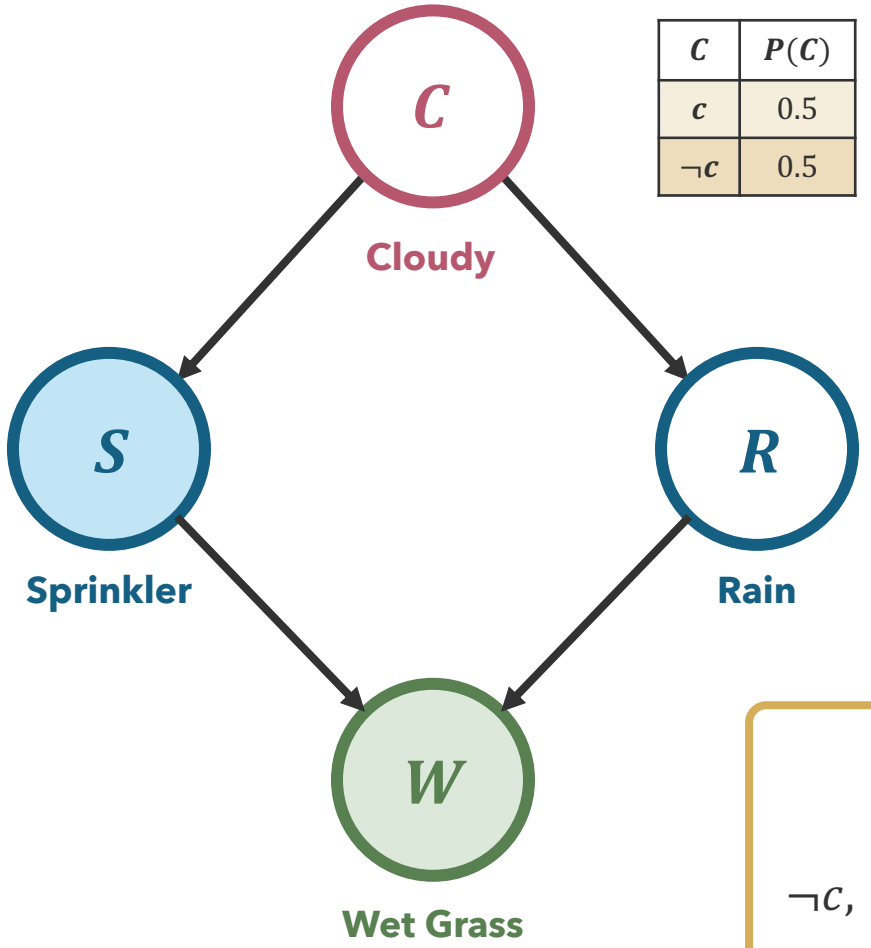
EXAMPLE

$P(S|C)$

| $C$ | $S$ | $P(S|C)$ |
|---|---|---|
| $c$ | $s$ | 0.1 |
| | $\neg s$ | 0.9 |
| $\neg c$ | $s$ | 0.5 |
| | $\neg s$ | 0.5 |

$P(W|S, R)$

| $S$ | $R$ | $W$ | $P(W|S, R)$ |
|---|---|---|---|
| $s$ | $r$ | $w$ | 0.99 |
| | | $\neg w$ | 0.01 |
| | $\neg r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| $\neg s$ | $r$ | $w$ | 0.90 |
| | | $\neg w$ | 0.10 |
| | $\neg r$ | $w$ | 0.01 |
| | | $\neg w$ | 0.99 |




$P(C)$

| $C$ | $P(C)$ |
|---|---|
| $c$ | 0.5 |
| $\neg c$ | 0.5 |

$P(R|C)$

| $C$ | $R$ | $P(R|C)$ |
|---|---|---|
| $c$ | $r$ | 0.8 |
| | $\neg r$ | 0.2 |
| $\neg c$ | $r$ | 0.2 |
| | $\neg r$ | 0.8 |

**Sample:**

$\neg c,\ s,\ r, \neg w$ weight $= 0.5 \cdot 0.01$

# Likelihood Weighting Properties

**DEFINITION**

Generates samples randomly from the distribution of $Z$ with fixed evidence

$$S_{\text{weighted}}(\boldsymbol{z}, \boldsymbol{e}) = \prod_j P\left(z_j | \text{parents}\left(Z_j\right)\right)$$

- with sample weights $w(\boldsymbol{z}, \boldsymbol{e}) = \prod_k P(e_k | \text{parents}(E_k))$

- Combined:

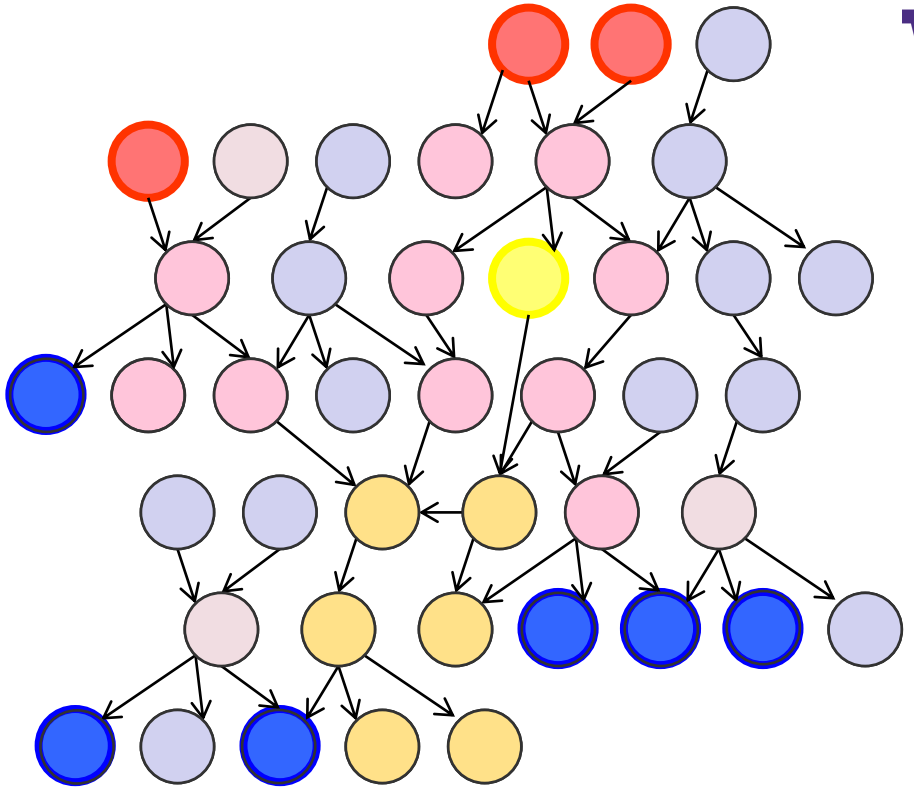
all other variables evidence variables

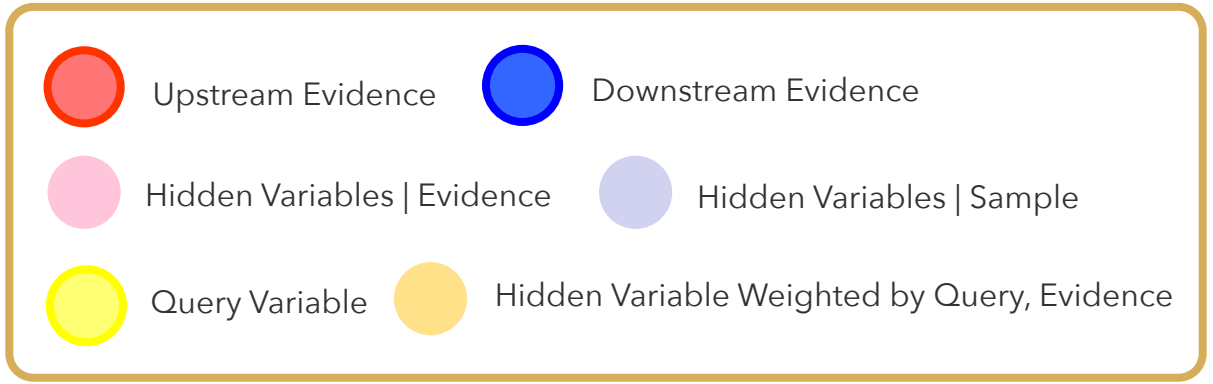
$$S_{\text{weighted}}(\boldsymbol{z}, \boldsymbol{e}) \cdot w(\boldsymbol{z}, \boldsymbol{e}) = \prod_j P\left(z_j | \text{parents}\left(Z_j\right)\right) \cdot \prod_k P(e_k | \text{parents}(E_k)) = P(\boldsymbol{z}, \boldsymbol{e})$$

# Impact of Evidence

DEFINITION

- With Likelihood Weighting:
  - All samples are used
  - ***Downstream* variables** are influenced by ***upstream* evidence**
- But…
  - ***Upstream* variables** are not influenced by ***downstream* evidence**
  - With evidence in $k$ leaf nodes, weights will be $O(d^{-k})$
  - With high probability, one lucky sample will have much higher weight, dominating result
- **Goal**: Each variable "sees" all evidence!





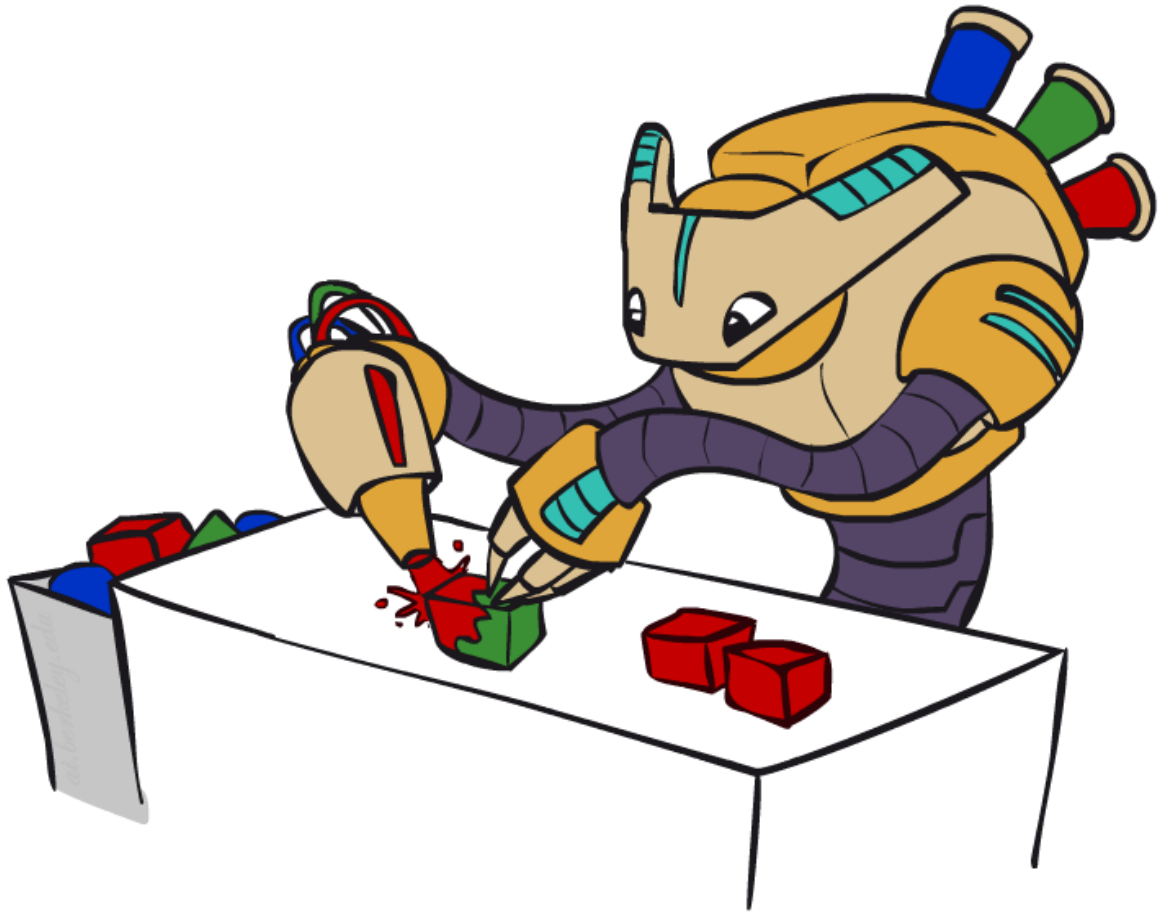
QUESTIONS

# Questions?

live and on sli.do #cse473

# A Different Approach: Gibbs Sampling

CONTEXT



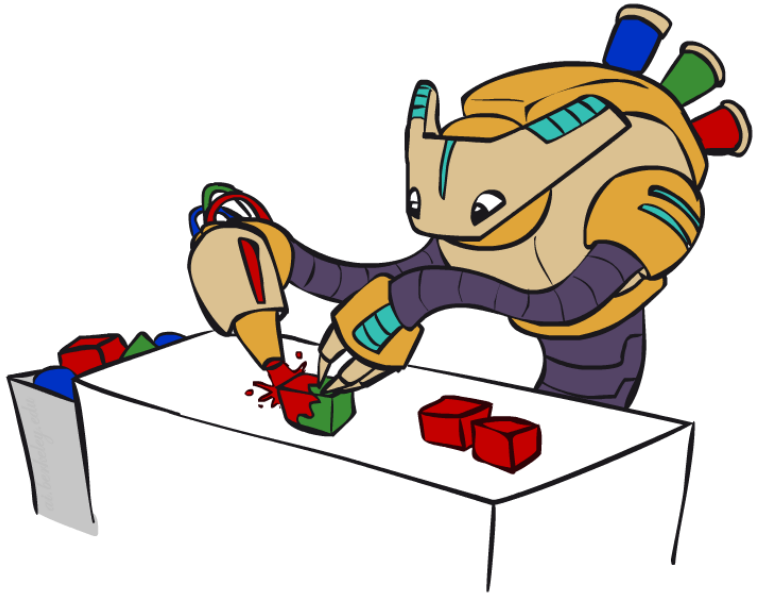
# Markov Chain Monte Carlo

CONTEXT

- Monte Carlo algorithms provide sampling methods with some probability of producing an incorrect answer
- **Markov Chain Monte Carlo** is family of randomized algorithms for approximating some quantity over very large state spaces
  - A Markov chain involves a sequence of randomly chosen states ("random walk"), where each state is chosen conditioned (only) on the previous state
  - In short: MCMC involves wandering around for a bit, then taking the average of observations

# Gibbs Sampling

FOUNDATIONS

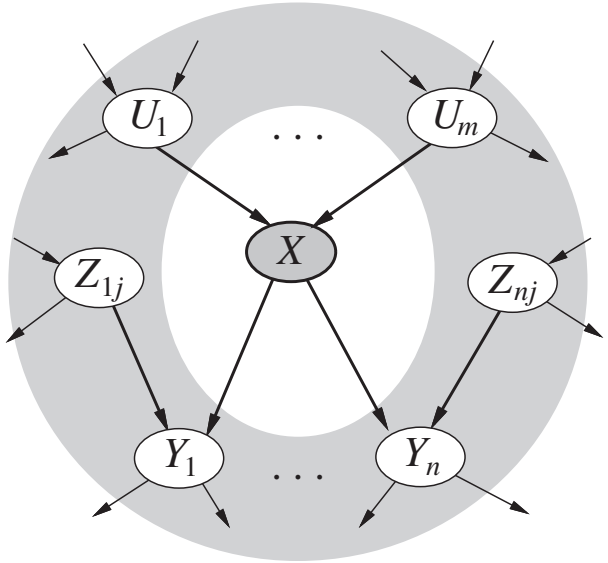


- **Gibbs sampling** is a specific kind of MCMC
  - States are complete assignments of all variables
  - Evidence variables are fixed, other variables change
- To generate the next state:
  - Pick a variable $X_i'$ and sample according to

$$X_i' \sim P(X_i | x_1, \ldots x_{i-1}, x_i, \ldots, x_n)$$

- Results:
  - Tends towards states with higher probability
  - In a Bayes net, take advantage of **Markov blanket**

$$P(X_i | x_1, \ldots, x_{i-1}, x_i, \ldots, x_n) = P(X_i | \text{markov\_blanket}(X_i))$$

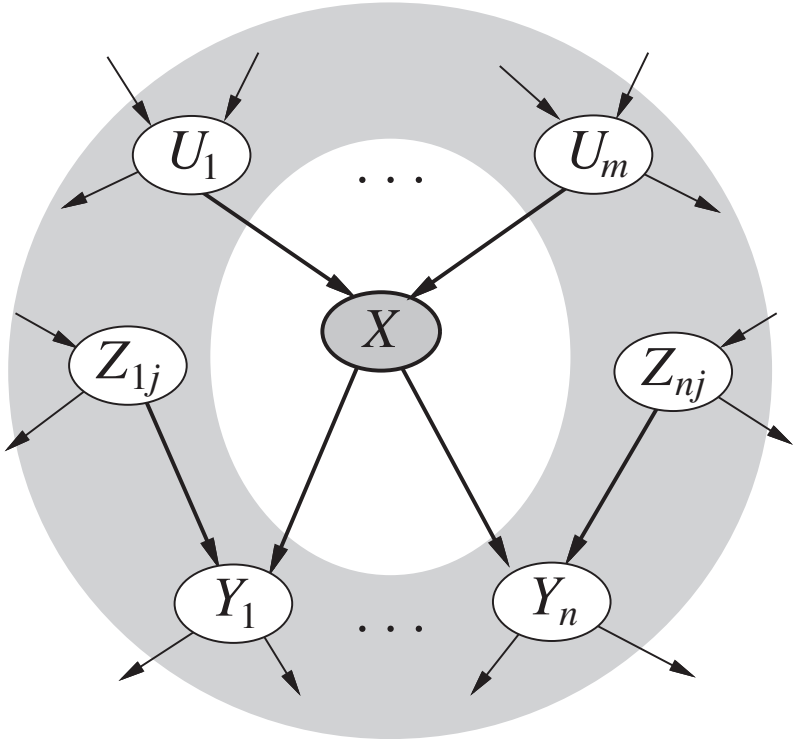


# Implementing Gibbs Sampling

DEFINITION

- Keep track of full instantiation $x_1, \ldots, x_n$
- Repeat many times:
  - Sample non-evidence variable $X_i$ from

$$P(X_i|x_1, \ldots, x_{i-1}, x_i, \ldots, x_n) = P(X_i|\text{markov\_blanket}(X_i))$$
$$= \alpha \cdot P(X_i|\text{parents}(X_i)) \prod_j P(y_j|\text{parents}(Y_j))$$




## Theorem: Gibbs sampling is consistent *

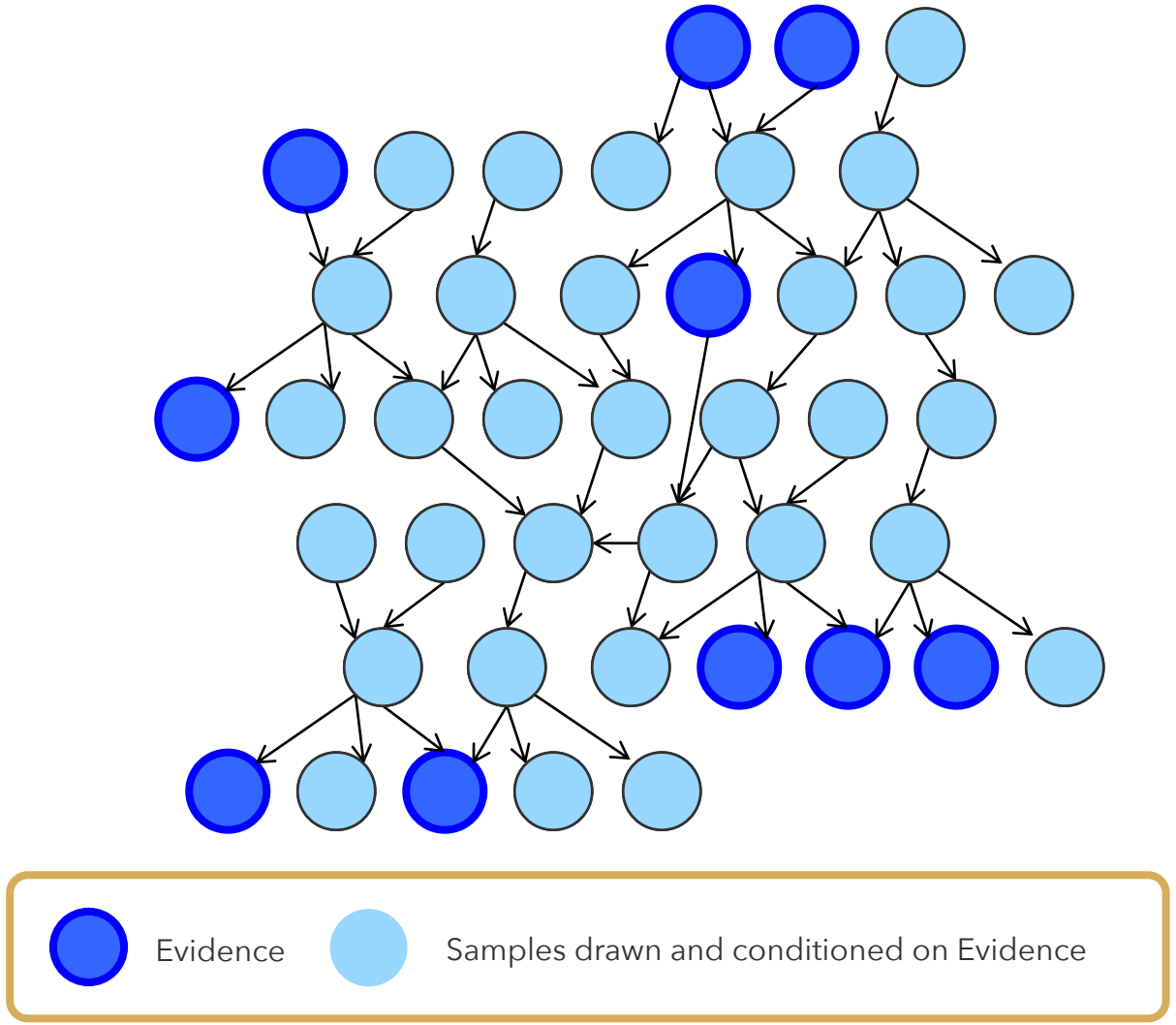
- Rationale: Both upstream and downstream variables condition on evidence

* Provided Gibbs distributions are bounded away from 0 and 1 and variable selection is fair

# Influence of Evidence

DEFINITION

- Samples soon begin to reflect all evidence in the network
- Eventually, samples are being drawn from the true posterior $P(X|\boldsymbol{e})$!



QUESTIONS

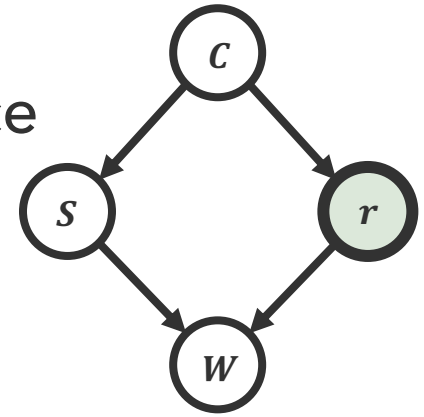
# Questions?

live and on sli.do #cse473

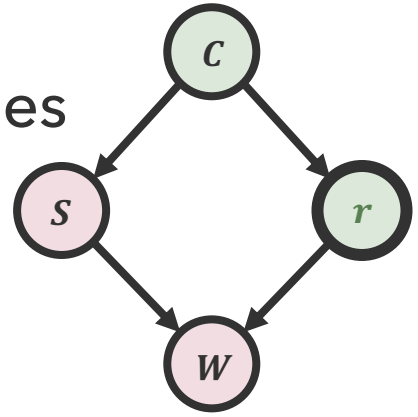
# Gibbs Sampling: $P(S|r)$

EXAMPLE

- Step 1: Fix evidence

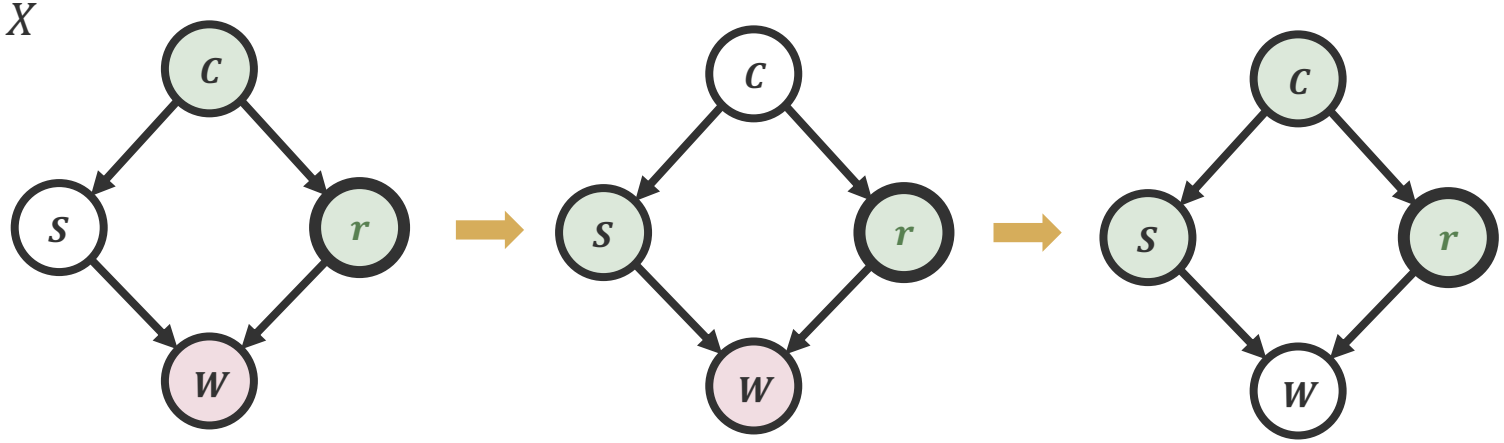



- Step 2: Initialize other variables




- Step 3: Repeat:
  - Choose non-evidence variable $X$
  - Resample $x$ from

    $P(X|\text{markov\_blanket}(X))$




sample $S \sim P(S|c, r, \neg w)$ sample $C \sim P(C|s, r)$ sample $W \sim P(W|s, r)$

# A Timely Connection…

CONTEXT

*How large must $N$ be for MCMC to converge to the true distribution?*

- CSE Prof. Oveis Gharan's research on this problem recently contributed to his winning the 2026 IMU Abacus Medal:

> "Oveis Gharan…answered this question, thereby resolving the Mihail-Vazirani conjecture, which had been open for three decades. Oveis Gharan et al established a bound on how long the algorithm needs to run to sample sufficiently. That they found exactly the right bound is extraordinary"
>
> - A. Jackson, IMU Write-Up

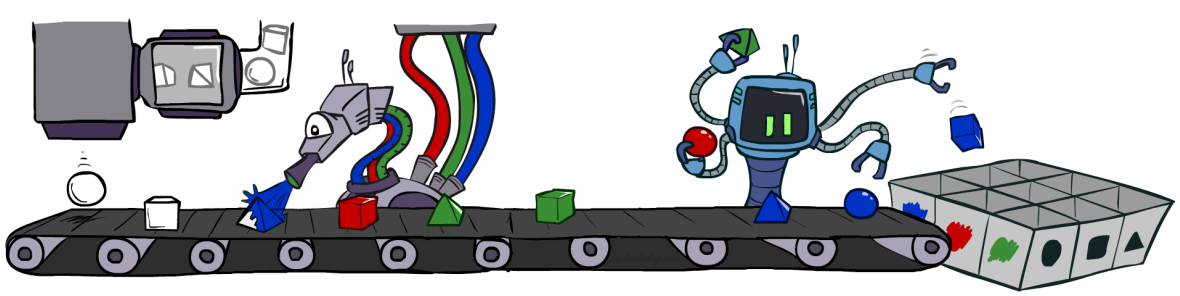
**Shayan Oveis Gharan**

CSE Professor

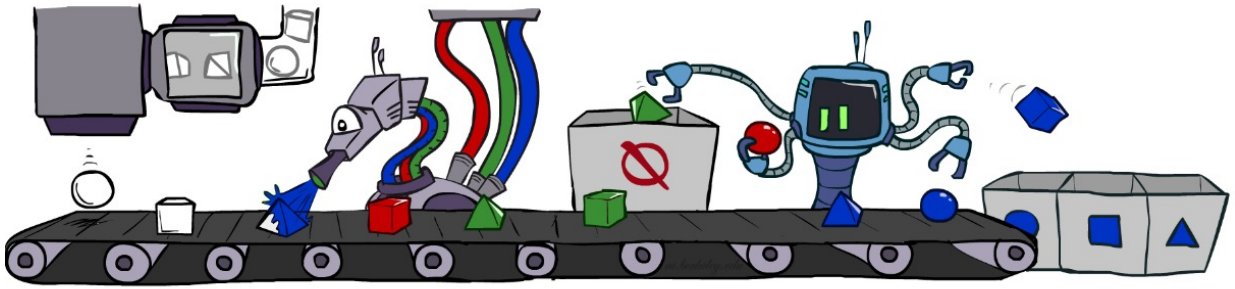
# Sampling Summary

CONTEXT

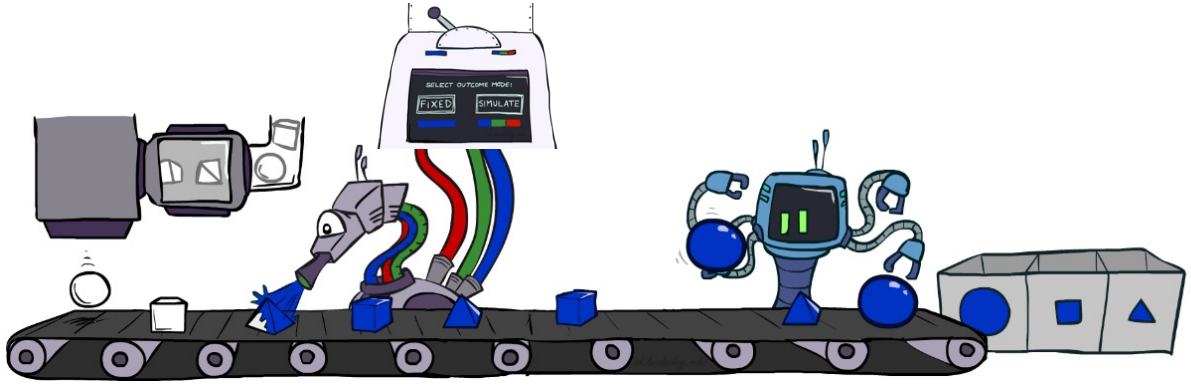
**Prior Sampling** $P(Q)$



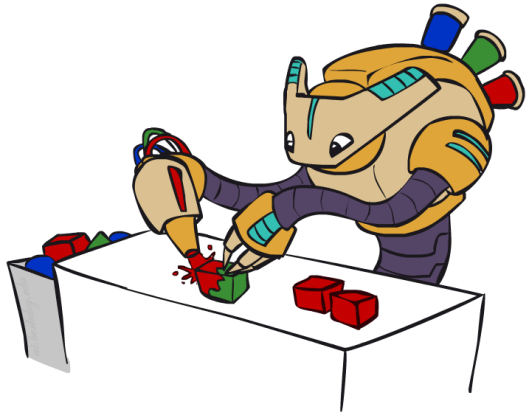
**Rejection Sampling** $P(Q|e)$



**Likelihood Weighting** $P(Q|e)$




**Gibbs Sampling** $P(Q|e)$



QUESTIONS

# Questions?

live and on sli.do #cse473

# that's it for today!

**SUMMARY**

- Achieve approximate inference results by sampling from local distributions
- With consistent sampling method, sample converges to true query distribution

**UPCOMING**

- Markov Models
- Particle Filters

**REMINDERS**

- Complete **Practice Problem 13**
- Do **Project 3** (due 7.30)
- Do **Homework 3** (due 8.6)