Review: probabilistic inference

Input

Bayesian network: $\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n)$

Evidence: $E = e$ where $E \subseteq X$ is subset of variables

Query: $Q \subseteq X$ is subset of variables

Output

$\mathbb{P}(Q = q \mid E = e)$ for all values q

X_1	X_2	
0	0	25
0	1	25
1	0	30
1	1	20

X_1	X_2	
0	0	0.25
0	1	0.25
1	0	0.3
1	1	0.2

Review: probabilistic inference

Input

Bayesian network: $\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n)$

Evidence: $E = e$ where $E \subseteq X$ is subset of variables

Query: $Q \subseteq X$ is subset of variables

Output

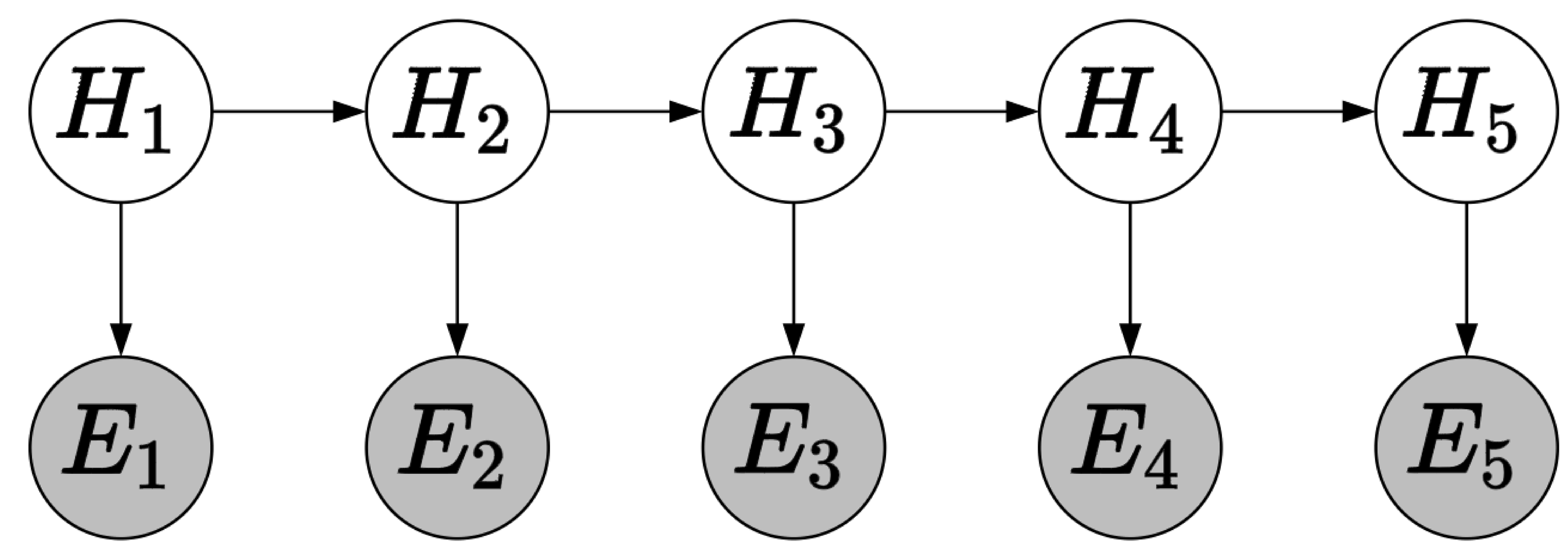
$\mathbb{P}(Q = q \mid E = e)$ for all values q

Example: if coughing but no itchy eyes, have a cold?

$$\mathbb{P}(C \mid H = 1, I = 0)$$

1. Remove everything irrelevant (not ancestors of Q / E)
2. Convert to factor graph
3. Condition on E
4. Variable elim on everything not in Q or E

Hidden Markov model



Problem: object tracking

$H_i \in \{1, \ldots, K\}$: location of object at time step i

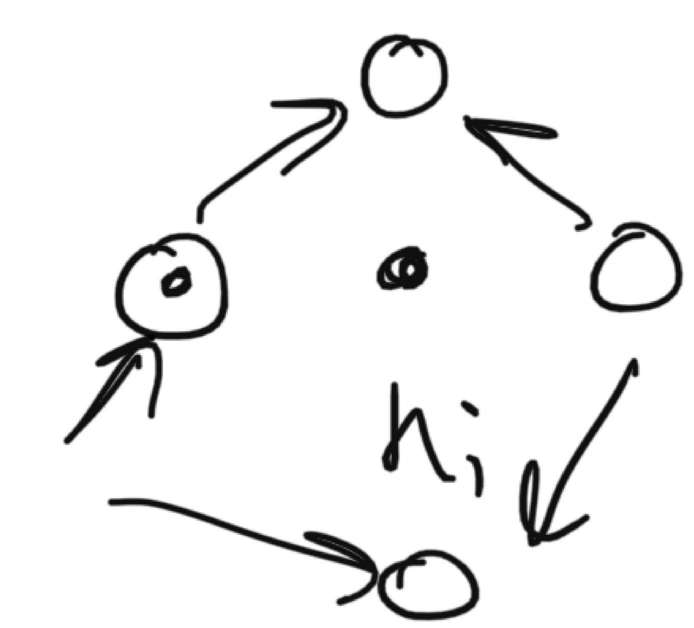
$E_i \in \{1, \ldots, K\}$: sensor reading at time step i

Start $p(h_1)$: e.g., uniform over all locations

Transition $p(h_i \mid h_{i-1})$: e.g., uniform over adjacent loc.

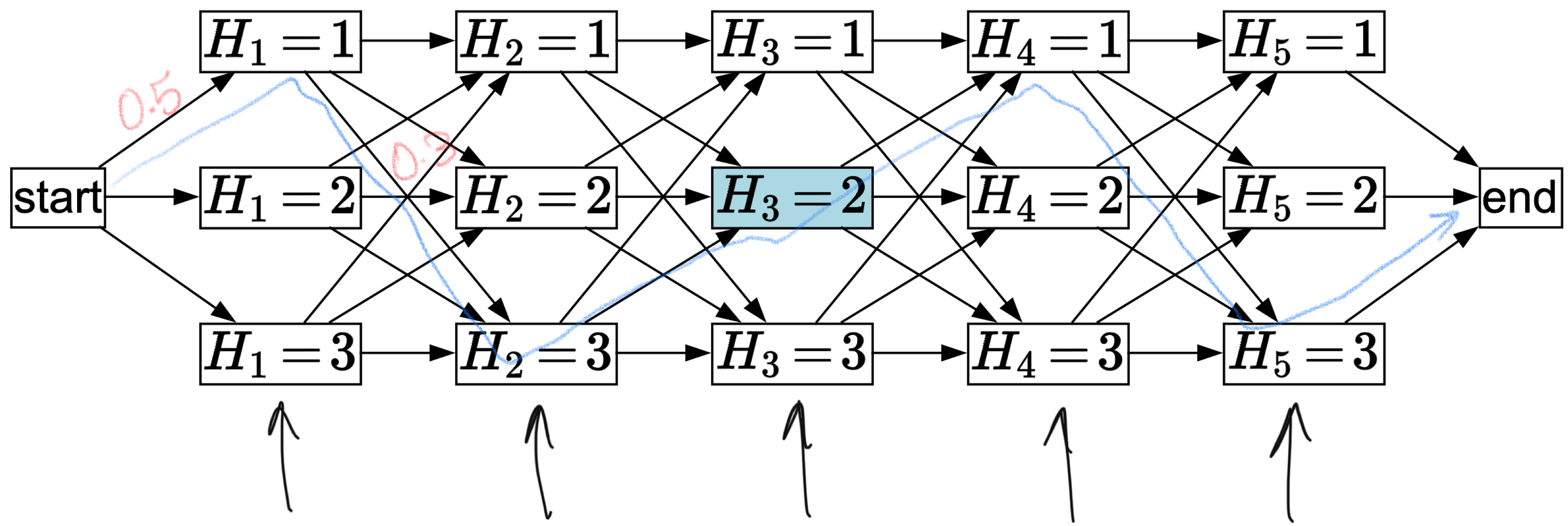
Emission $p(e_i \mid h_i)$: e.g., uniform over adjacent loc.

$$\mathbb{P}(H = h, E = e) = \underbrace{p(h_1)}_{\text{start}} \prod_{i=2}^{n} \underbrace{p(h_i \mid h_{i-1})}_{\text{transition}} \prod_{i=1}^{n} \underbrace{p(e_i \mid h_i)}_{\text{emission}}$$

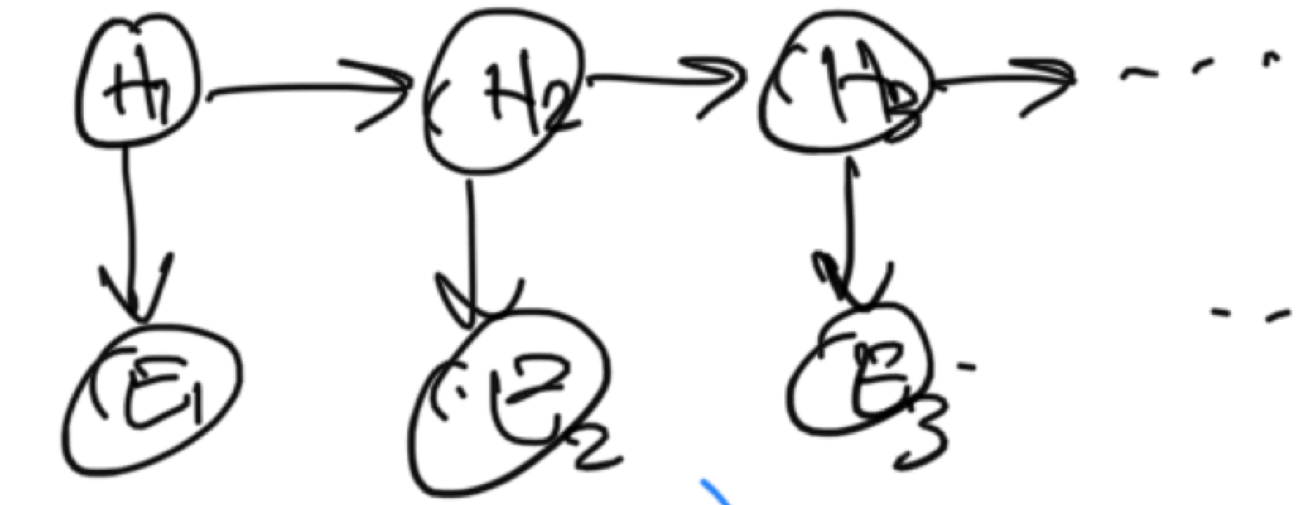


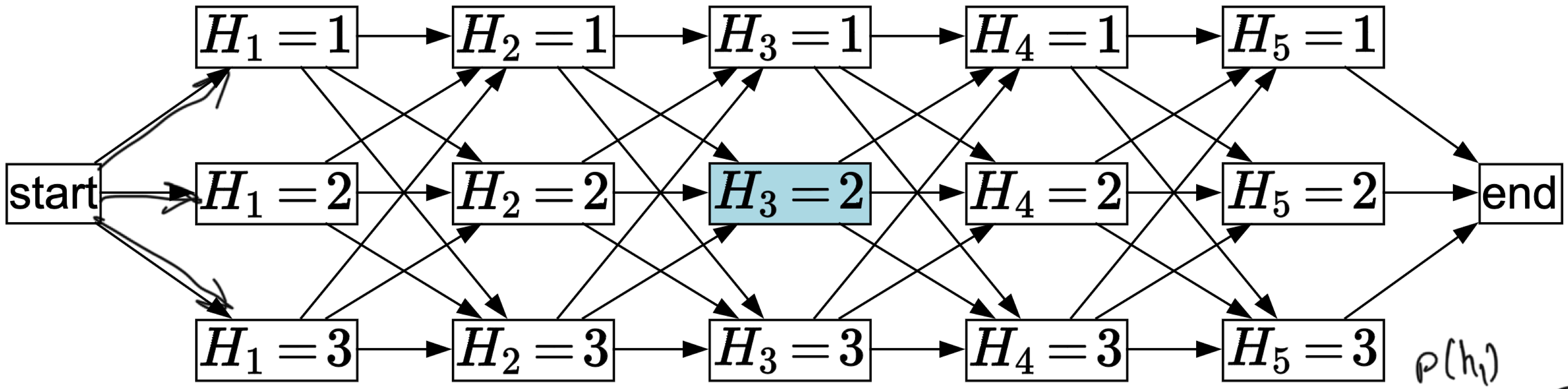
$$p(h_i \mid h_{i-1}) = \begin{cases} \frac{1}{4} & \text{if } h_i \sim \\ 0 & \text{o.w.} \end{cases}$$

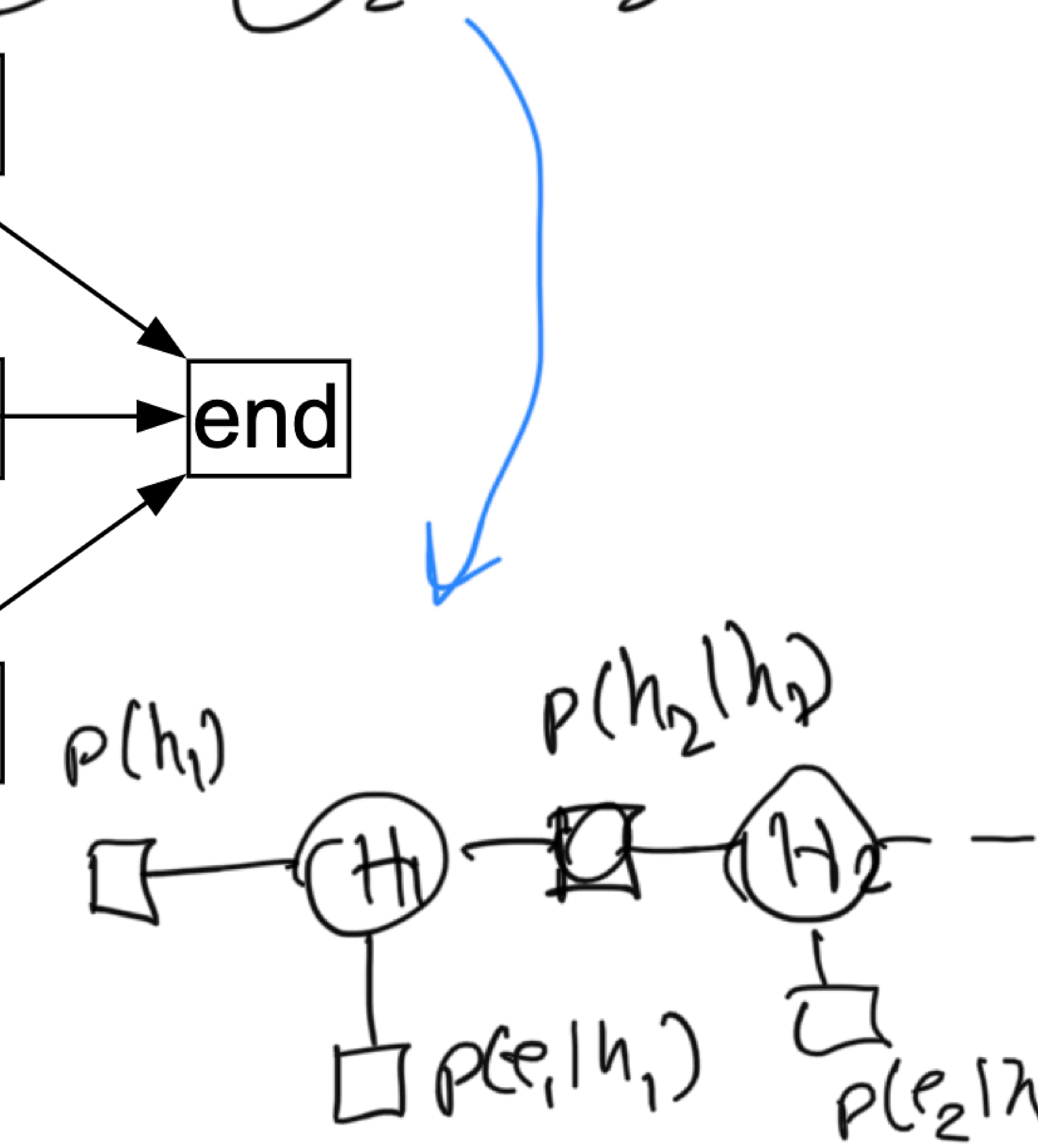
Lattice representation



Lattice representation

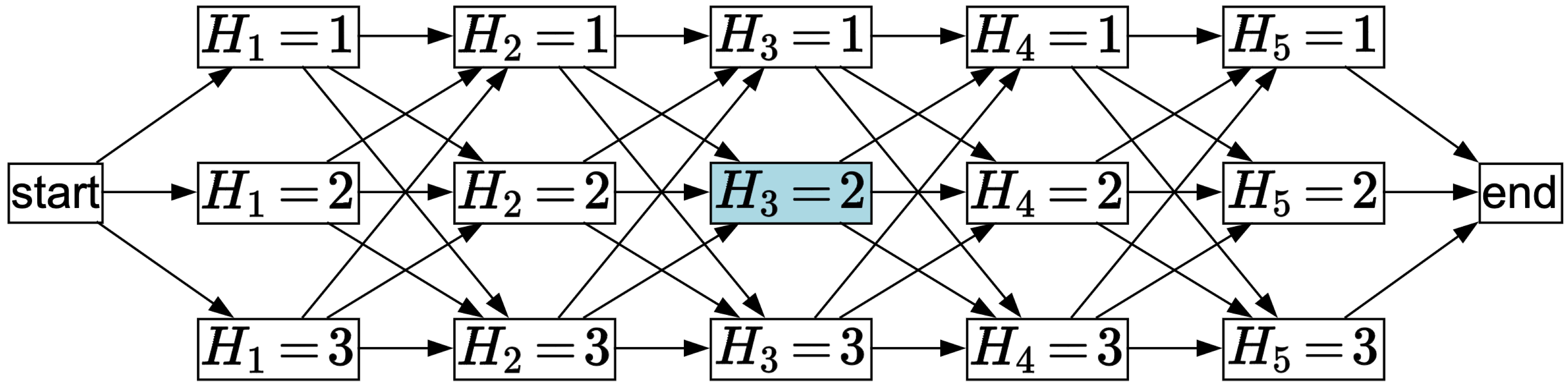






- Edge $\boxed{\text{start}} \Rightarrow \boxed{H_1 = h_1}$ has weight $p(h_1)p(e_1 \mid h_1)$
- Edge $\boxed{H_{i-1} = h_{i-1}} \Rightarrow \boxed{H_i = h_i}$ has weight $p(h_i \mid h_{i-1})p(e_i \mid h_i)$
- Each path from $\boxed{\text{start}}$ to $\boxed{\text{end}}$ is an assignment with weight equal to the product of node/edge weights

Lattice representation

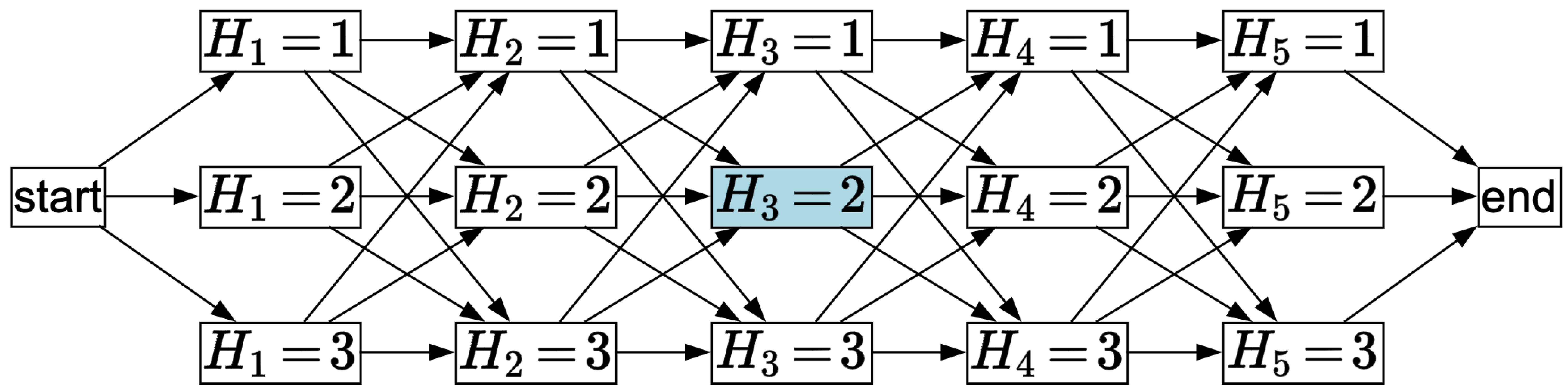


- Edge $\boxed{\text{start}} \Rightarrow \boxed{H_1 = h_1}$ has weight $p(h_1)p(e_1 \mid h_1)$
- Edge $\boxed{H_{i-1} = h_{i-1}} \Rightarrow \boxed{H_i = h_i}$ has weight $p(h_i \mid h_{i-1})p(e_i \mid h_i)$
- Each path from $\boxed{\text{start}}$ to $\boxed{\text{end}}$ is an assignment with weight equal to the product of node/edge weights

$\mathbb{P}(H_3 \mid E_1 = e_1, E_2 = e_2, \ldots)$

$\boxed{\mathbb{P}(H_3 = h_3 \wedge E_1 = e_1 \wedge E_2 = e_2 \wedge \ldots)}$

Lattice representation

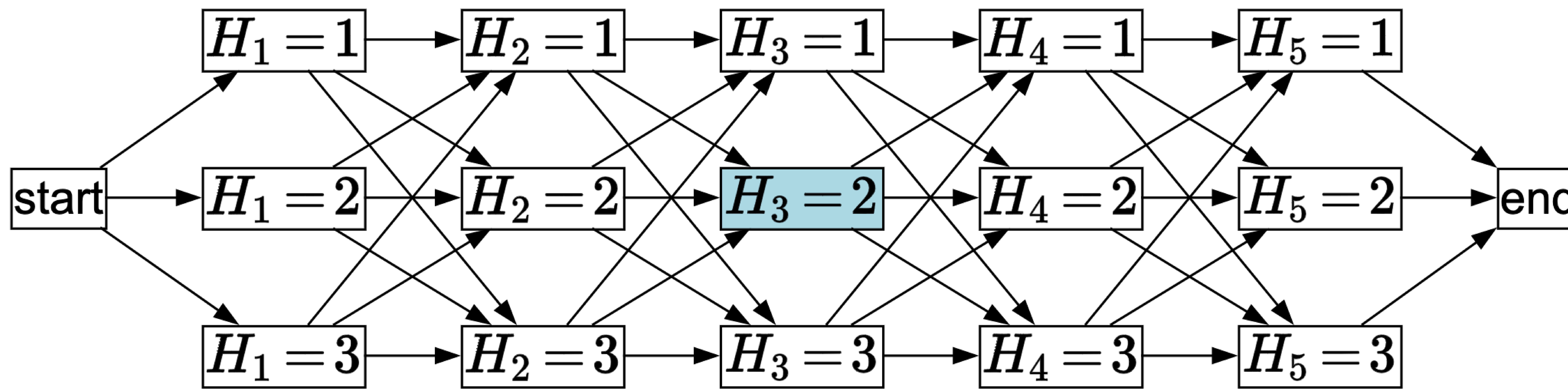


$$\sum_{\substack{\text{paths} \\ \text{going} \\ \text{through } h_3}} \text{weight(path)}$$

$$= \left(\sum_{\substack{\text{partial} \\ \text{path from} \\ \text{start to } h_3}} \text{weight(p.p.)} \right) \cdot \left(\sum_{\substack{\text{partial} \\ \text{path from} \\ h_3 \text{ to end}}} \text{weigh(p.p)} \right)$$

Lattice representation

Sum of all paths through $H_i = h_i$:



$$\underbrace{\mathbb{P}[H_i = h_i \wedge E_1 = e_1 \wedge E_2 = e_2 \wedge \cdots]}_{\text{for all } h_i}$$

$\Big\Downarrow$ normalize

$$\mathbb{P}[H_i = h_i \mid E_1 = e_1 \wedge E_2 = e_2 \wedge \cdots]$$

Forward: $F_i(h_i) = \sum_{h_{i-1}} F_{i-1}(h_{i-1}) w(h_{i-1}, h_i)$

sum of weights of paths from $\boxed{\text{start}}$ to $\boxed{H_i = h_i}$

Backward: $B_i(h_i) = \sum_{h_{i+1}} B_{i+1}(h_{i+1}) w(h_i, h_{i+1})$

sum of weights of paths from $\boxed{H_i = h_i}$ to $\boxed{\text{end}}$

Define $S_i(h_i) = F_i(h_i) B_i(h_i)$:

sum of weights of paths from $\boxed{\text{start}}$ to $\boxed{\text{end}}$ through $\boxed{H_i = h_i}$

Beam search

End result:

- Candidate list C is set of particles
- Use C to compute marginals

$\Pr[X_1 = 1]$

Step 1: propose

Old particles: $\approx \mathbb{P}(H_1, H_2 \mid E_1 = 0, E_2 = 1)$

[0, 1]
[1, 0]

$\mathbb{P}(H_1, \ldots, H_t \mid E_1 = e_1, \ldots, E_t = t)$

Step 2: weight

Old particles: $\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1)$

[0, 1, 1]
[1, 0, 0]

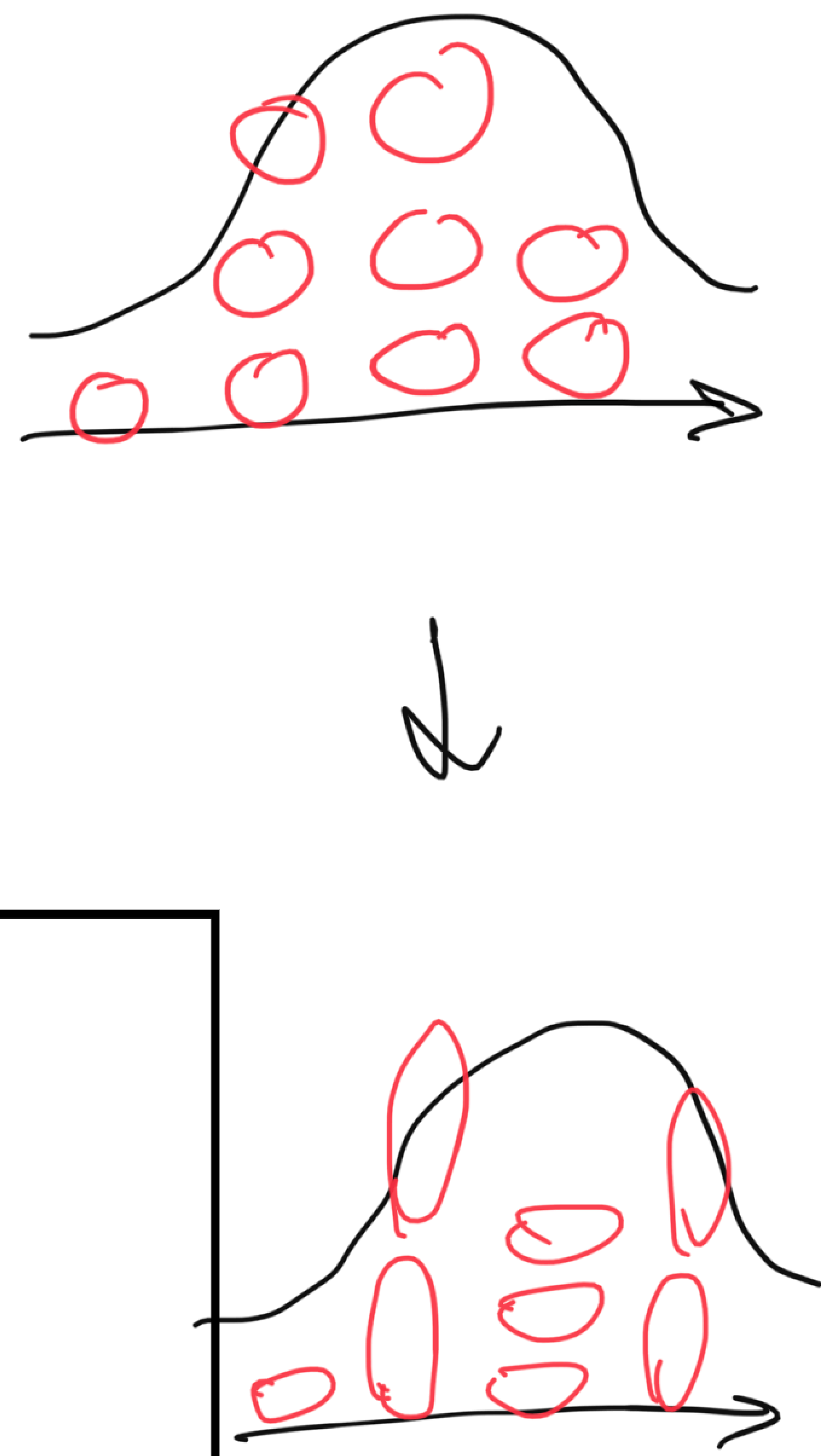
$\mathbb{P}(H_1, \ldots, H_{t+1} \mid E_1, \ldots, E_{t+1}) \propto$

$\mathbb{P}(H_1, \ldots, H_{t+1} \mid E_1, \ldots, E_t) \cdot p(E_{t+1} = e_{t+1} \mid H_{t+1})$

Step 2: weight

Old particles: $\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1)$

[0, 1, 1]
[1, 0, 0]



Key idea: weighting

For each old particle (h_1, h_2, h_3), weight it by $w(h_1, h_2, h_3) = p(e_3 \mid h_3)$.

New particles:

$$\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1, E_3 = 1)$$

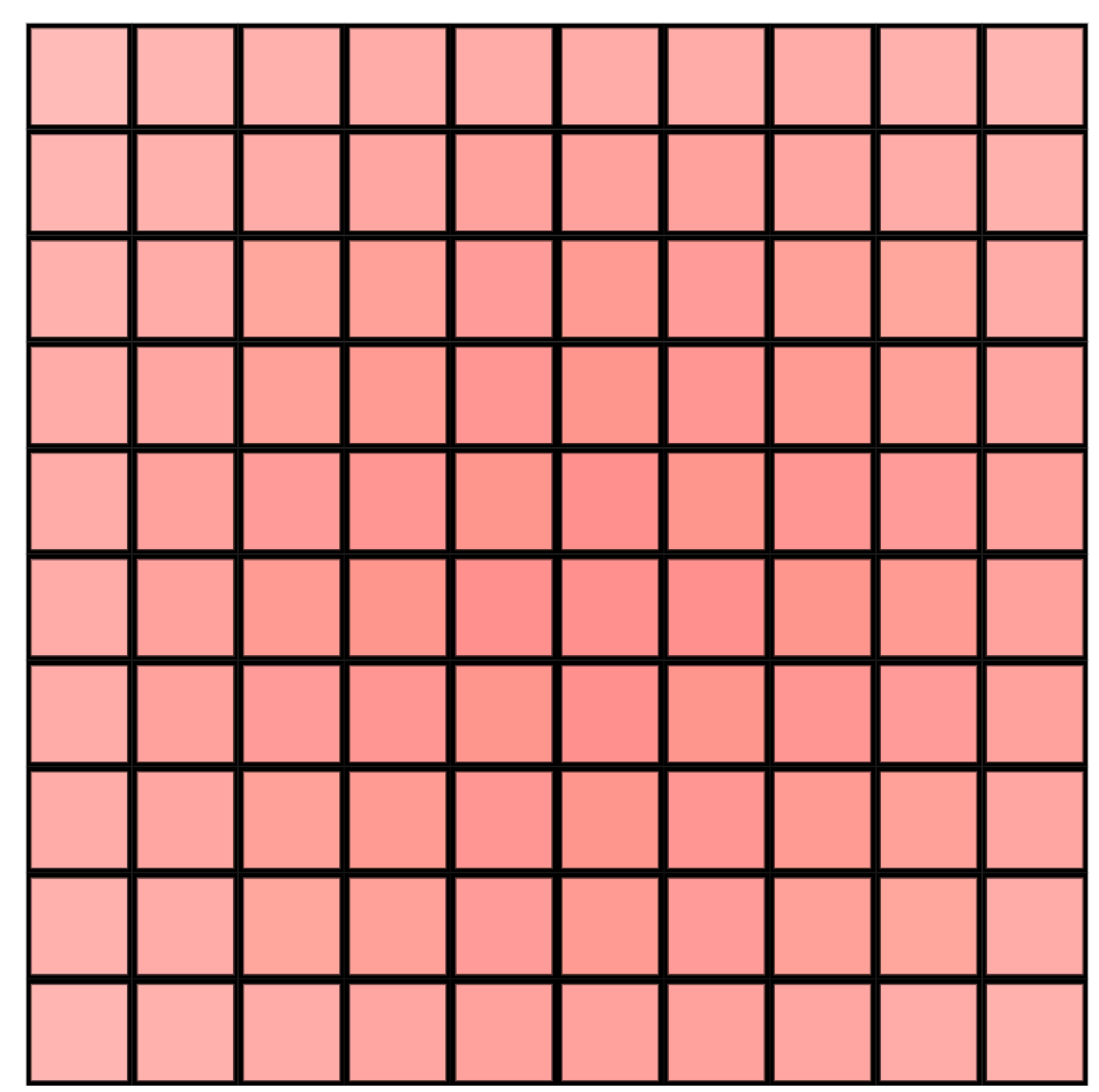
[0, 1, 1] (0.8)
[1, 0, 0] (0.4)

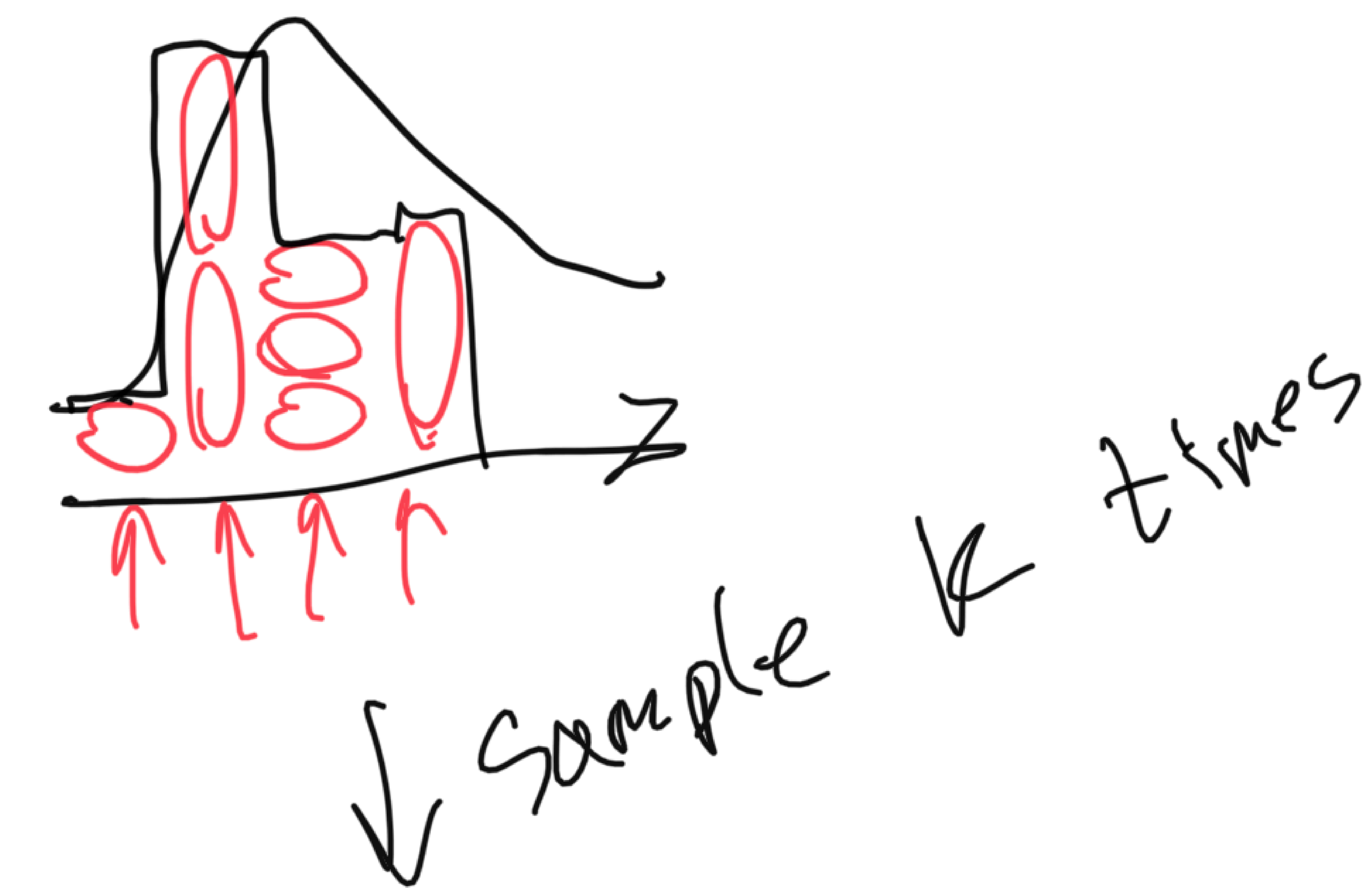
Step 3: resample

Question: given weighted particles, which to choose?

Tricky situation:

- Target distribution close to uniform
- Fewer particles than locations





Step 2: weight

Old particles: $\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1)$

$\mathbb{P}(H_1, H_2, H_3 \mid E_1, E_2, E_3) \propto \mathbb{P}(H_1, H_2, H_3 \mid E_1, E_2) \cdot \mathbb{P}(E_3 \mid H_3)$

[0, 1, 1]
[1, 0, 0]

Key idea: weighting

For each old particle (h_1, h_2, h_3), weight it by
$w(h_1, h_2, h_3) = p(e_3 \mid h_3)$.

New particles:

$$\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1, E_3 = 1)$$

[0, 1, 1] (0.8)
[1, 0, 0] (0.4)

Gibbs sampling

$X_1 \quad X_2 \;\text{---}\; X_n$

Setup:

$$\text{Weight}(x)$$

Πf
f depends on X_2

$\leftarrow X_1 \quad 0 \quad X_3 \;\text{--}\; X_n$

$\leftarrow X_1 \quad 1 \quad X_3 \;\text{---}\; X_n$

$\leftarrow X_1 \quad 2 \quad X_3 \;\text{---}\; X_n$

$\vdots$

$\leftarrow X_1 \quad \text{domain} - 1 \quad X_3 \;\text{---}\; X_n$

Algorithm: Gibbs sampling

Initialize x to a random complete assignment
Loop through $i = 1, \ldots, n$ until convergence:
  Compute weight of $x \cup \{X_i : v\}$ for each v
  Choose $x \cup \{X_i : v\}$ with probability prop. to weight

Examples: [vote] [csp] [pair] [chain] [track] [alarm] [med] [dep] [delay] [mln] [new]
[Background] [Documentation]

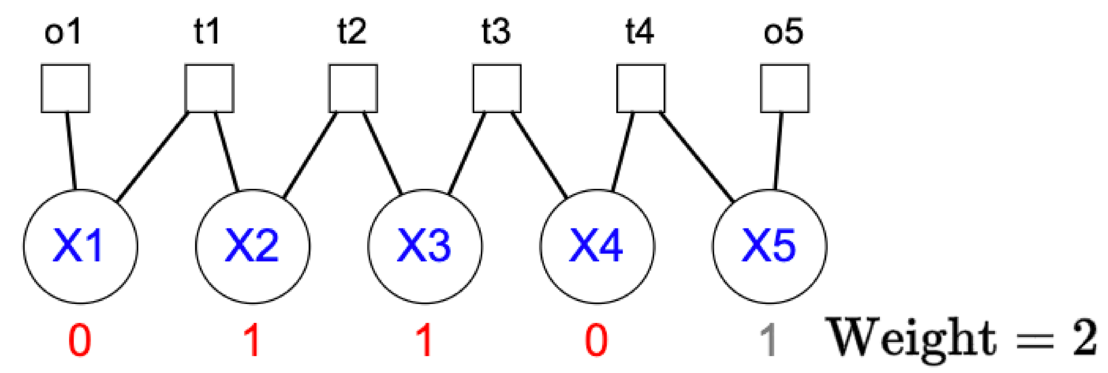
```
// Start at 0, end at 1, where in between?
variable('X1', [0, 1])
variable('X2', [0, 1])
variable('X3', [0, 1])
variable('X4', [0, 1])
variable('X5', [0, 1])

function nearby(a,b) { return a == b ? 2 : 1; }
factor('o1', 'X1', function(a) { return a == 0; })
factor('t1', 'X1 X2', nearby)
factor('t2', 'X2 X3', nearby)
factor('t3', 'X3 X4', nearby)
factor('t4', 'X4 X5', nearby)
factor('o5', 'X5', function(a) { return a == 1; })

query('X1 X2 X3'); gibbsSampling({steps:1})
```

Step (or press ctrl-enter in text box)

Query: $\mathbb{P}(\mathrm{X1}, \mathrm{X2}, \mathrm{X3})$
Algorithm: **Gibbs sampling**



Sampling variable X5 given everything else:

X5:?	t4	o5	Weight	$\mathbb{P}(\mathrm{X5} = ?)$
0	2	0	0	0
1	1	1	1	1

Choose X5:1

Estimate of query based on 25 samples:

X1	X2	X3	count	$\hat{\mathbb{P}}(\mathrm{X1,X2,X3})$
0	1	1	10	0.4
0	1	0	9	0.36
0	0	1	5	0.2
0	0	0	1	0.04

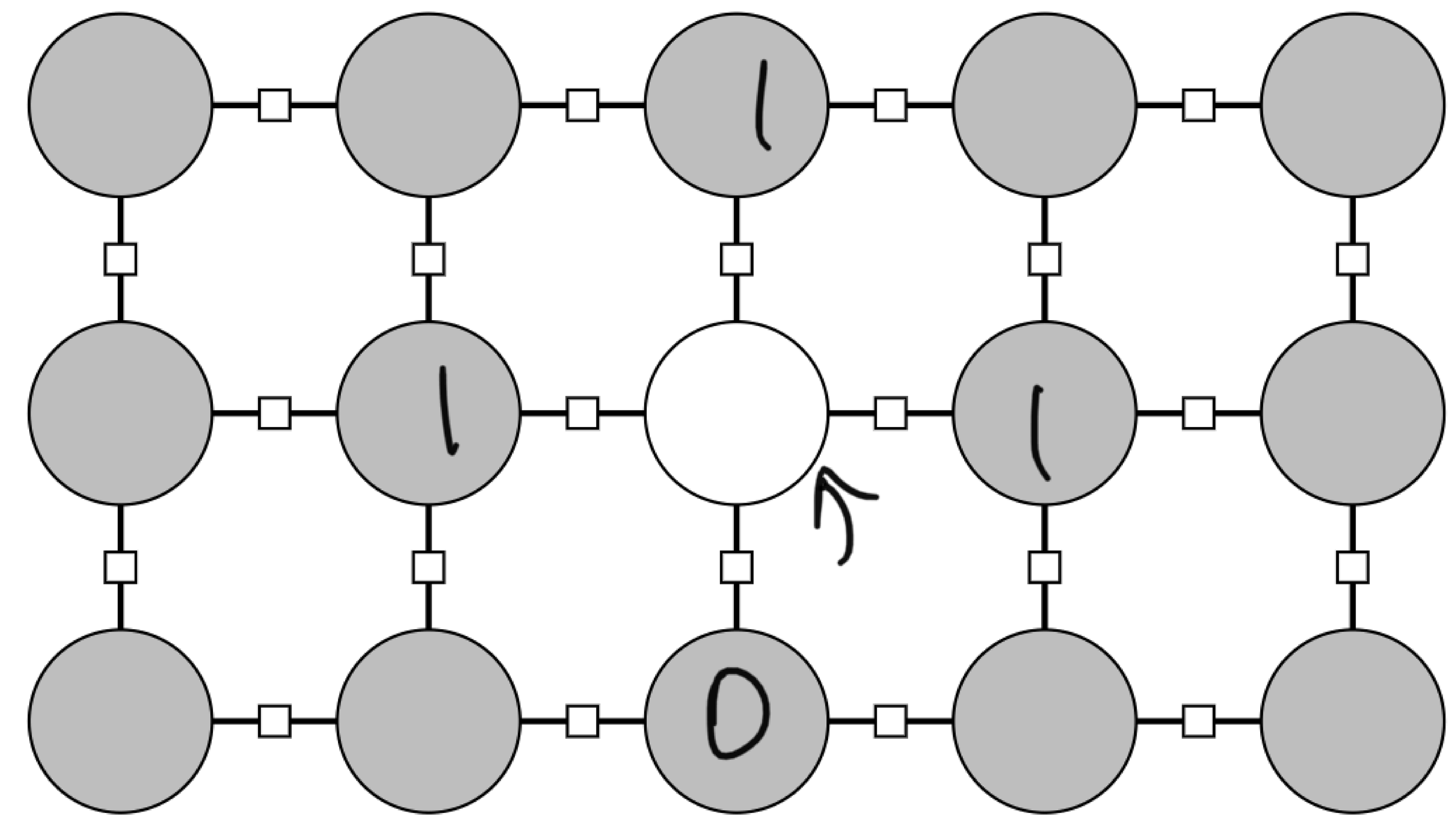
$(X_1^{(0)}, \ldots, X_n^{(0)})$

$(X_1^{(1)}, \ldots, X_n^{(1)})$

$\vdots$

$(X_1^{(t)}, \ldots, X_n^{(t)})$

Application: image denoising



$0 : 2 \times 1 \times 1 \times 1 \to 2$

$1 : 2 \times 2 \times 2 \times 1 \to 8$

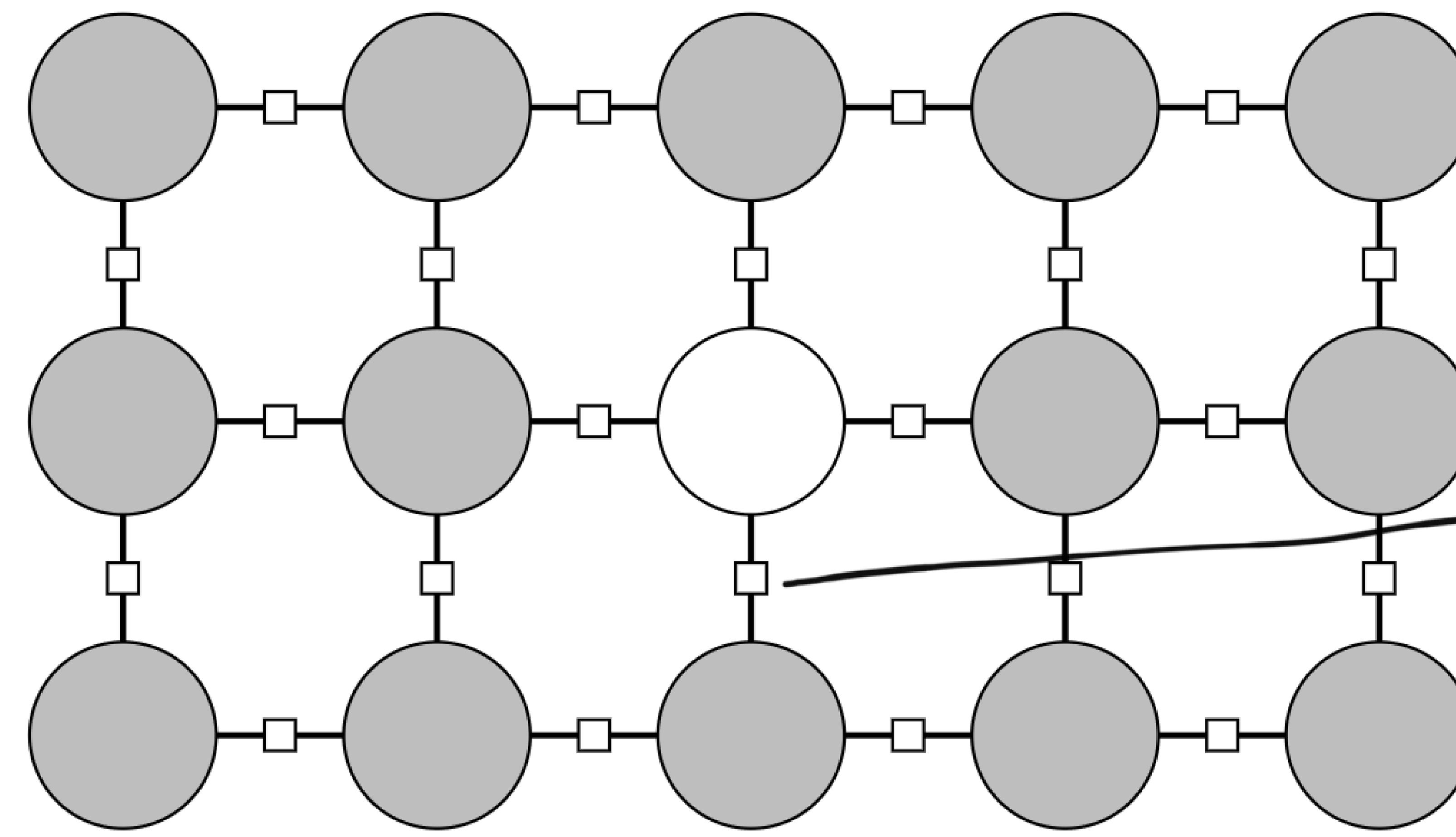
[whiteboard]

Example: image denoising

If neighbors are $1, 1, 1, 0$ and X_i not observed:

$$\mathbb{P}(X_i = 1 \mid X_{-i} = x_{-i}) = \frac{2 \cdot 2 \cdot 2 \cdot 1}{2 \cdot 2 \cdot 2 \cdot 1 + 1 \cdot 1 \cdot 1 \cdot 2} = 0.8$$

Application: image denoising



$$\begin{cases} 2 & \text{if agree} \\ 1 & \text{if disagree} \end{cases}$$

[whiteboard]

Example: image denoising

If neighbors are $1, 1, 1, 0$ and X_i not observed:

$$\mathbb{P}(X_i = 1 \mid X_{-i} = x_{-i}) = \frac{2 \cdot 2 \cdot 2 \cdot 1}{2 \cdot 2 \cdot 2 \cdot 1 + 1 \cdot 1 \cdot 1 \cdot 2} = 0.8$$

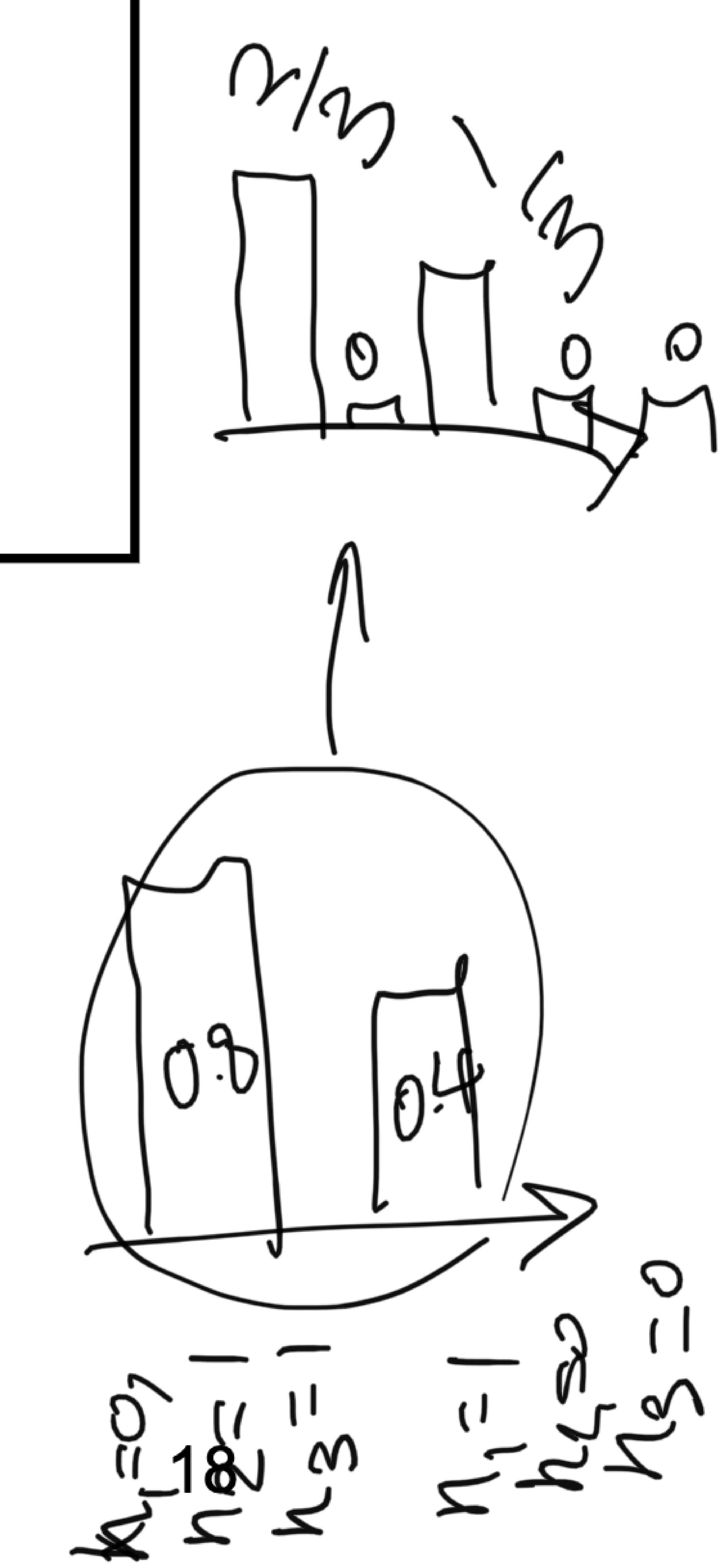
Step 2: weight

Old particles: $\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1)$

[0, 1, 1]
[1, 0, 0]

Key idea: weighting

For each old particle (h_1, h_2, h_3), weight it by
$w(h_1, h_2, h_3) = p(e_3 \mid h_3)$.



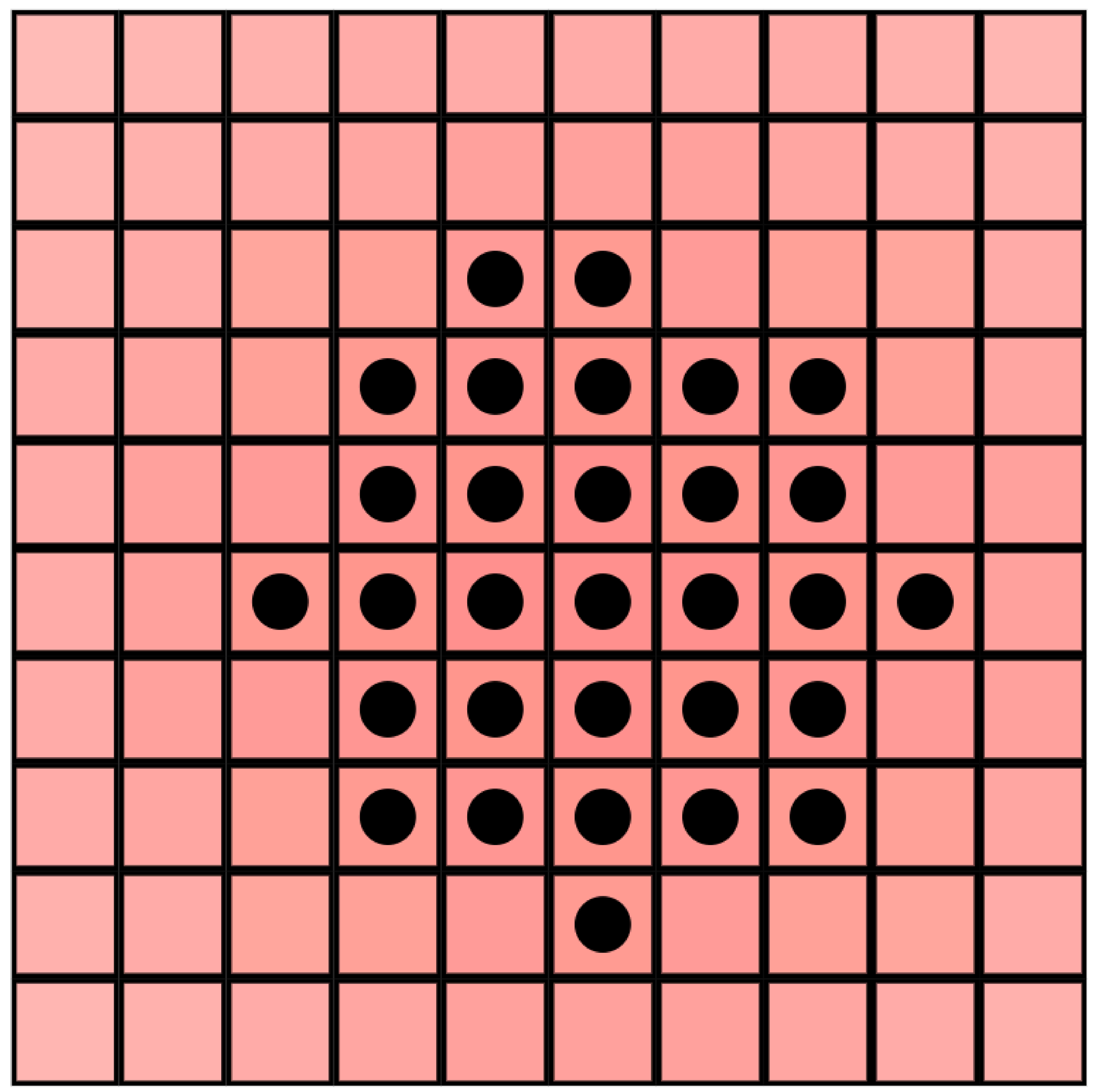
New particles:

$$\approx \mathbb{P}(H_1, H_2, H_3 \mid E_1 = 0, E_2 = 1, E_3 = 1)$$

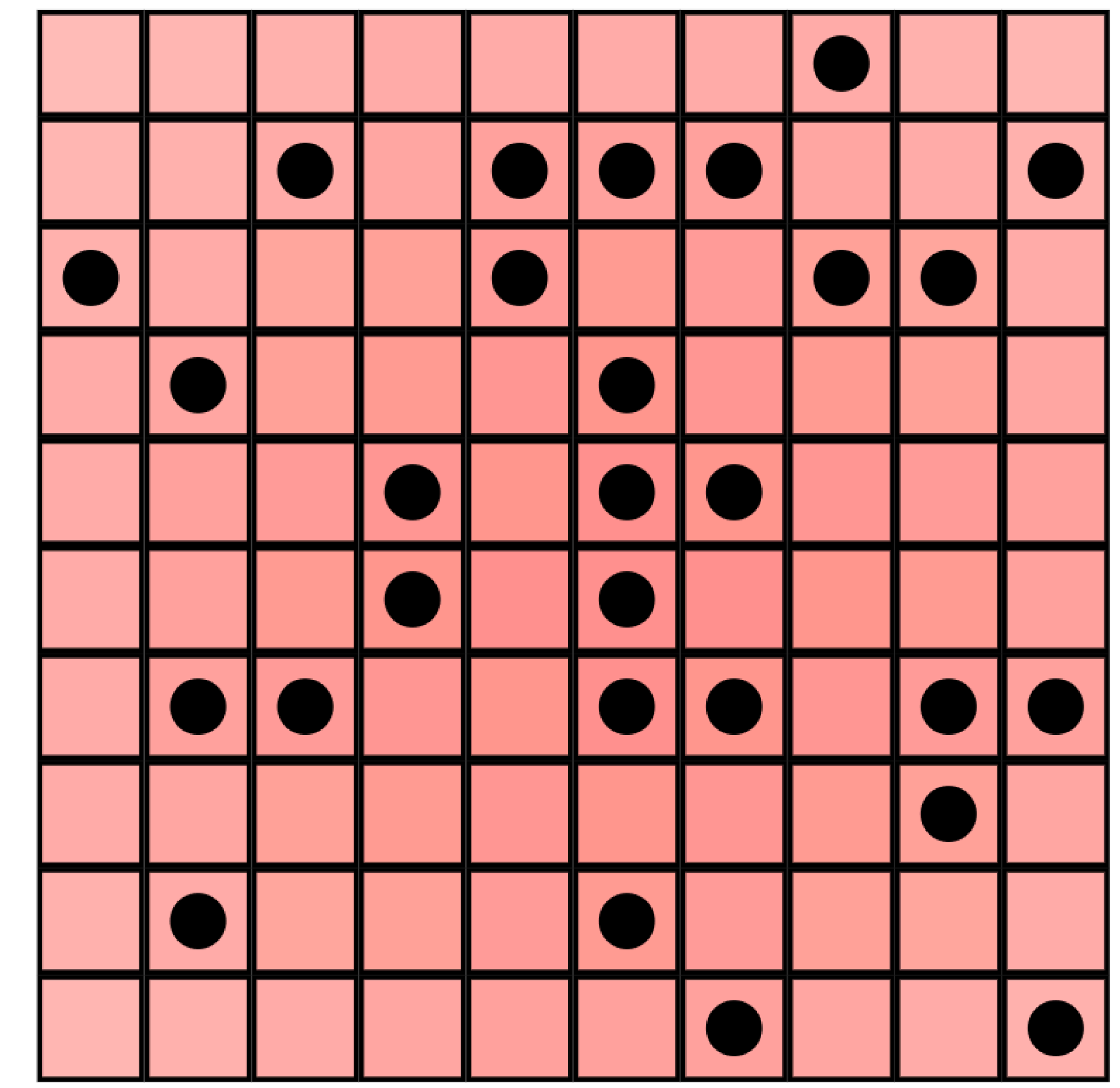
[0, 1, 1] (0.8)
[1, 0, 0] (0.4)

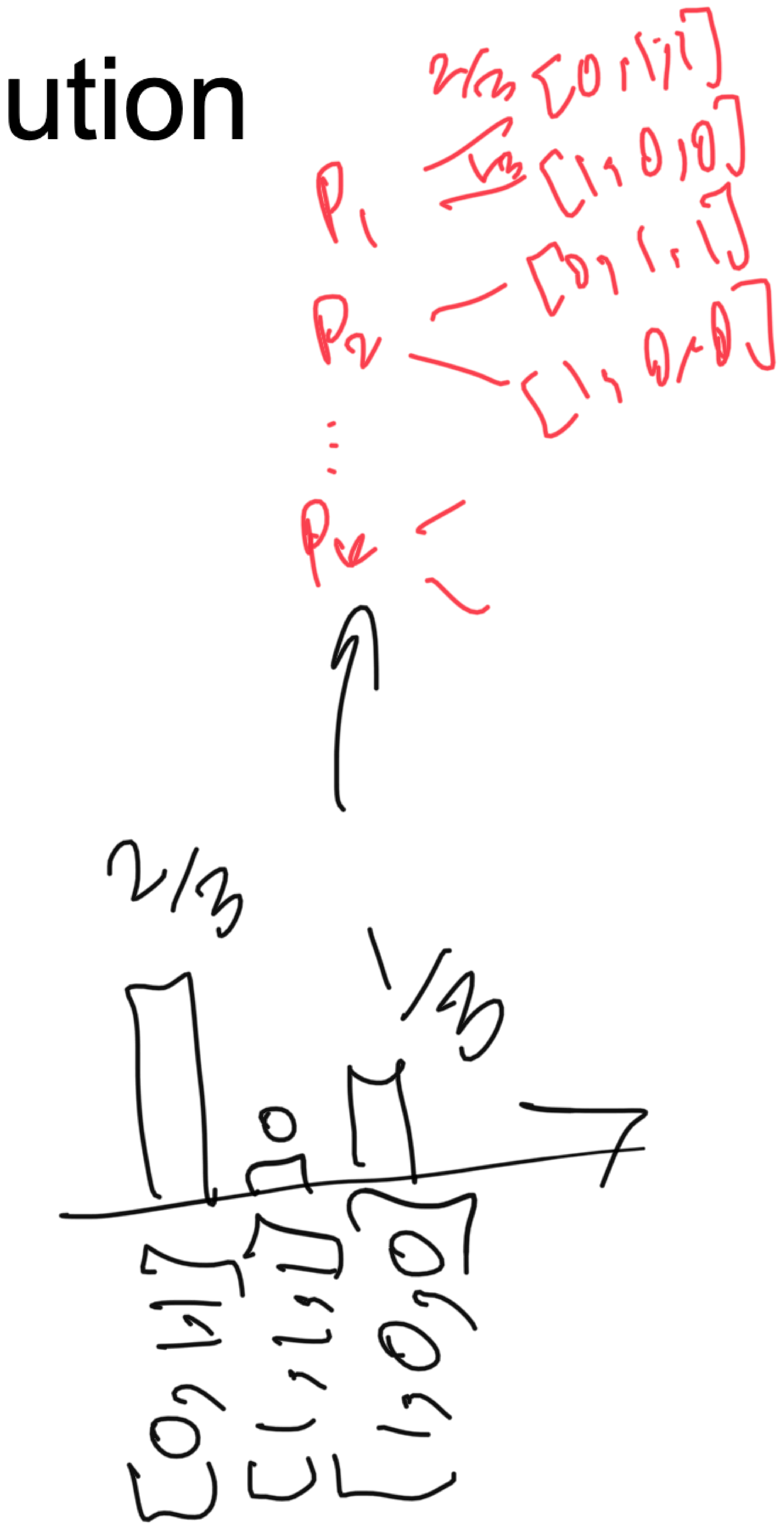
Step 3: resample

K with highest weight



K sampled from distribution





Intuition: top K assignments not representative.

Maybe random samples will be more representative...