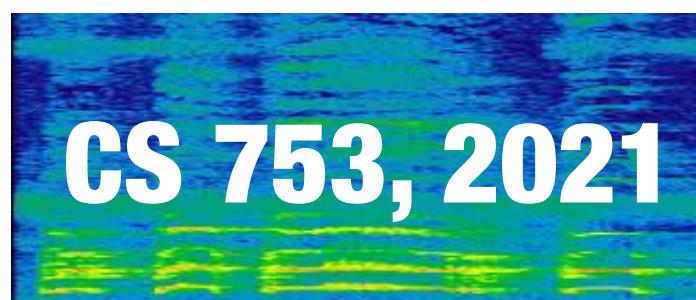


# **HMMs for Acoustic Modeling**

Lecture 1a



Instructor: Preethi Jyothi, IITB

## Recall: Statistical ASR

Let  $\mathbf{O}$  be a sequence of acoustic features corresponding to a speech signal. That is,  $\mathbf{O} = \{O_1, \dots, O_T\}$ , where  $O_i \in \mathbb{R}^d$  refers to a  $d$ -dimensional acoustic feature vector and  $T$  is the length of the sequence.

Let  $\mathbf{W}$  denote a word sequence. An ASR decoder solves the following problem:

$$\begin{aligned}\mathbf{W}^* &= \arg \max_{\mathbf{W}} \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_{\mathbf{W}} \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W}) \\ &\approx \arg \max_{\mathbf{W}} \sum_{\mathbf{Q}} \Pr(\mathbf{O} | \mathbf{Q}) \Pr(\mathbf{Q} | \mathbf{W}) \Pr(\mathbf{W})\end{aligned}$$

# Recall: Statistical ASR

Let  $\mathbf{O}$  be a sequence of acoustic features corresponding to a speech signal. That is,  $\mathbf{O} = \{O_1, \dots, O_T\}$ , where  $O_i \in \mathbb{R}^d$  refers to a  $d$ -dimensional acoustic feature vector and  $T$  is the length of the sequence.

Let  $\mathbf{W}$  denote a word sequence. An ASR decoder solves the following problem:

$$\begin{aligned}\mathbf{W}^* &= \arg \max_W \Pr(\mathbf{W} | \mathbf{O}) \\ &= \arg \max_W \Pr(\mathbf{O} | \mathbf{W}) \Pr(\mathbf{W}) \\ &\approx \arg \max_W \sum_Q \Pr(\mathbf{O} | Q) \Pr(Q | \mathbf{W}) \Pr(W)\end{aligned}$$

Acoustic  
Model

# Generative Model of Speech

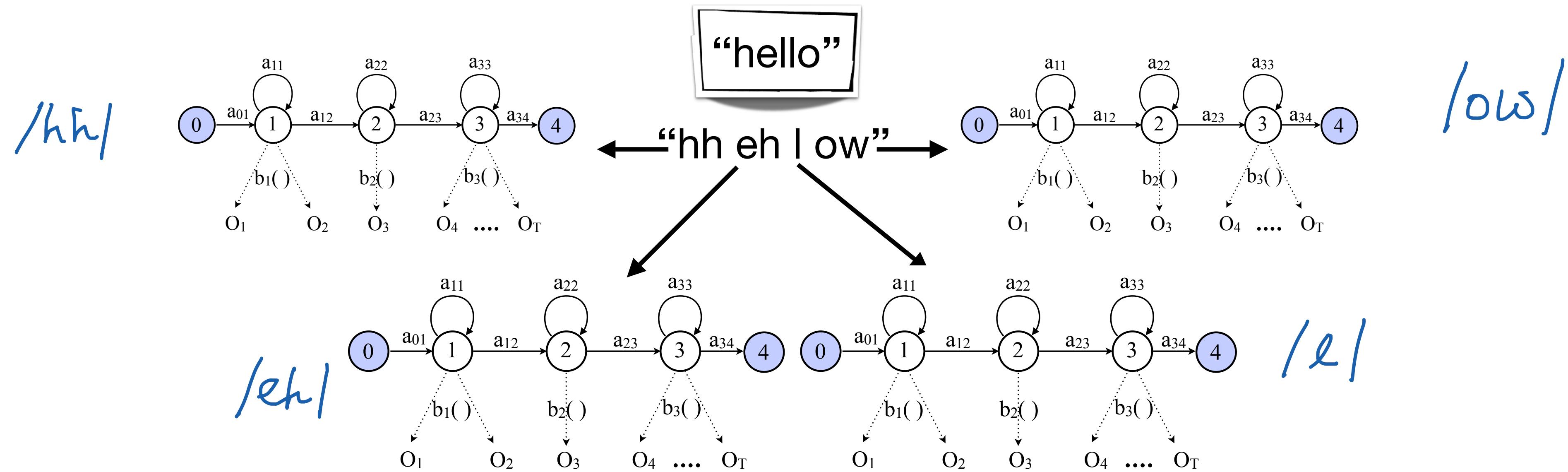
“hello”

# Generative Model of Speech

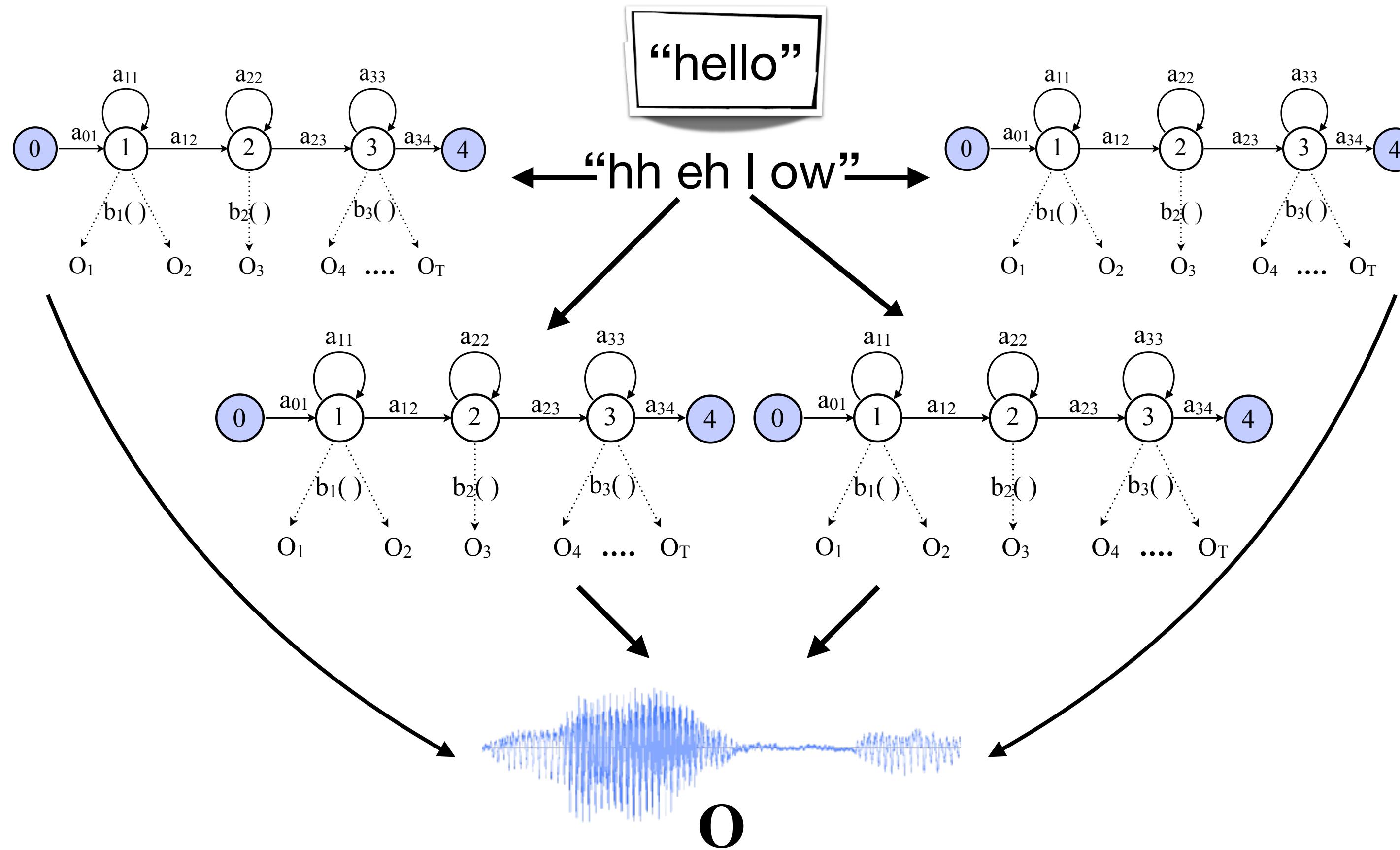
“hello”

“hh eh l ow”

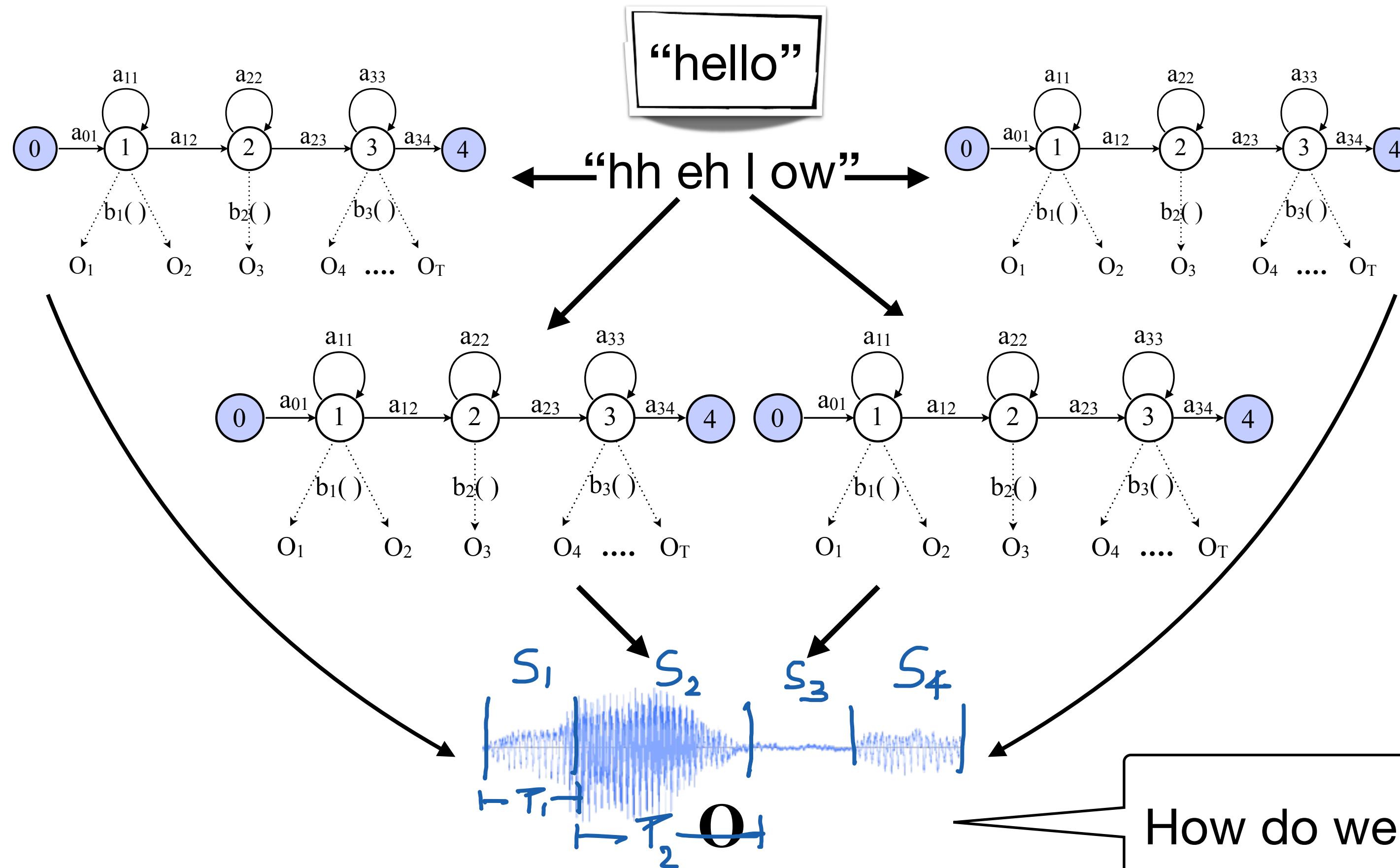
# Generative Model of Speech



# Generative Model of Speech



# Generative Model of Speech



$$P(O | \text{hello}) = P(S_1 | hh) P(S_2 | eh) P(S_3 | l) P(S_4 | ow)$$

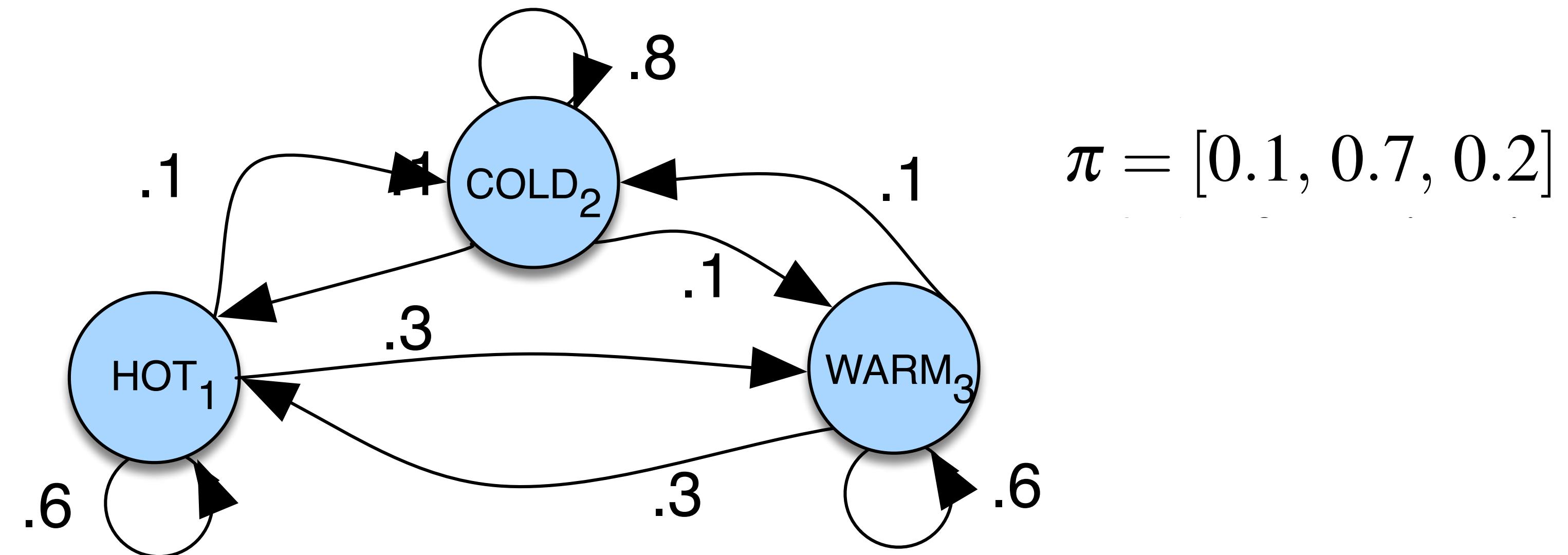
$P(O_1, \dots, O_{T_1} | hh)$

$P(O_{T_1+1}, \dots, O_{T_1+T_2} | eh, \dots)$

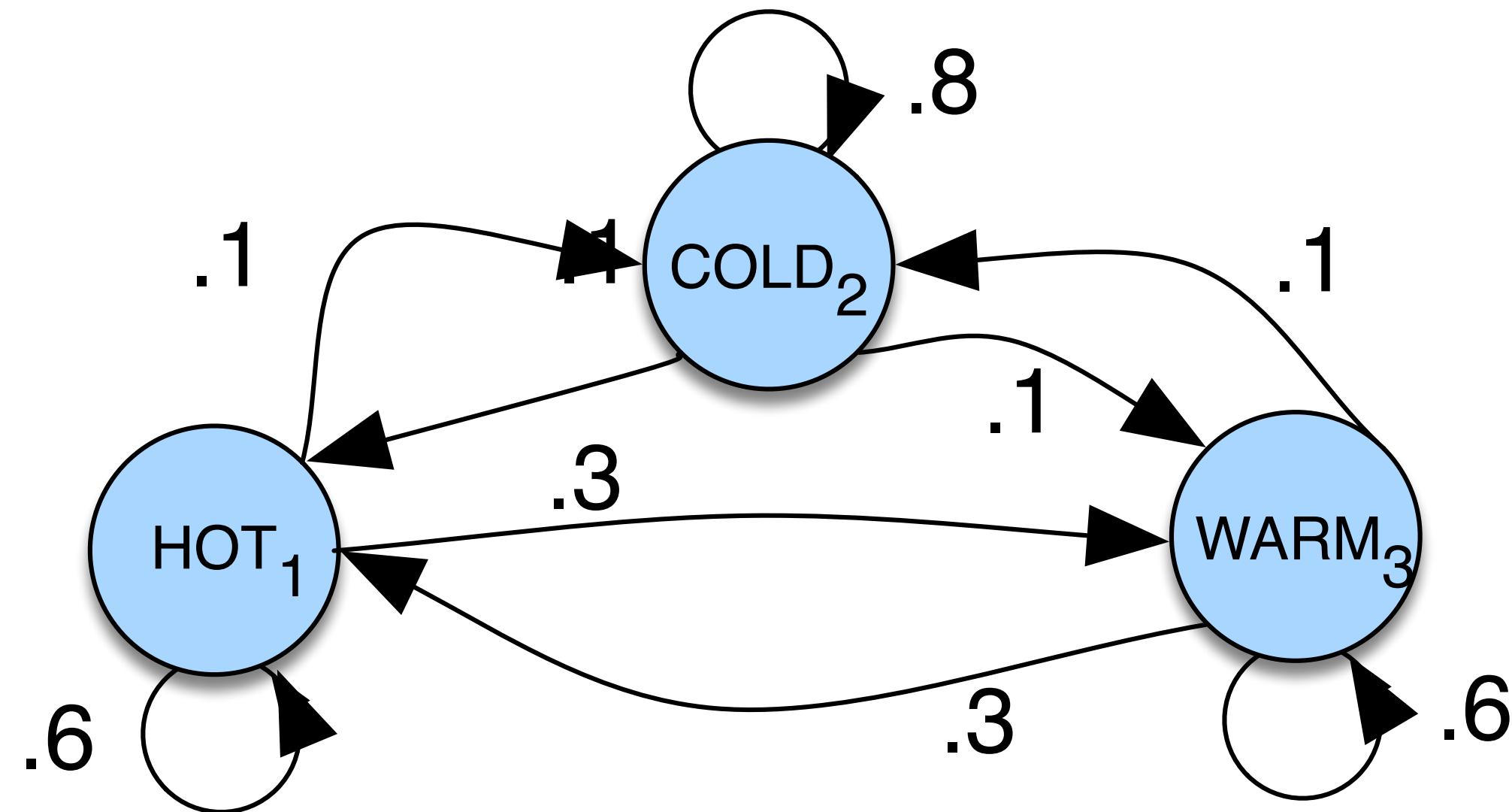
# **What are Hidden Markov Models (HMMs)?**

Following slides contain figures/material from “Hidden Markov Models”,  
“Speech and Language Processing”, D. Jurafsky and J. H. Martin, 2019.  
(<https://web.stanford.edu/~jurafsky/slp3/A.pdf>)

# Markov Chains



# Markov Chains



$$\pi = [0.1, 0.7, 0.2]$$

$$P(HHH)$$

$$P(WHW)$$

$$P(CHW)$$

$$Q = q_1 q_2 \dots q_N$$

a set of  $N$  states

$$A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}$$

a **transition probability matrix**  $A$ , each  $a_{ij}$  representing the probability of moving from state  $i$  to state  $j$ , s.t.  
 $\sum_{j=1}^n a_{ij} = 1 \quad \forall i$

$$\pi = \pi_1, \pi_2, \dots, \pi_N$$

an **initial probability distribution** over states.  $\pi_i$  is the probability that the Markov chain will start in state  $i$ . Some states  $j$  may have  $\pi_j = 0$ , meaning that they cannot be initial states. Also,  $\sum_{i=1}^n \pi_i = 1$

# Hidden Markov Model

$$Q = q_1 q_2 \dots q_N$$

$$A = a_{11} \dots a_{ij} \dots a_{NN}$$

$$O = \underline{o_1 o_2 \dots o_T}$$

$$B = \underbrace{b_i(o_t)}_{P(s_t | q_t = i)}$$

$$\pi = \pi_1, \pi_2, \dots, \pi_N$$

a set of  $N$  **states**

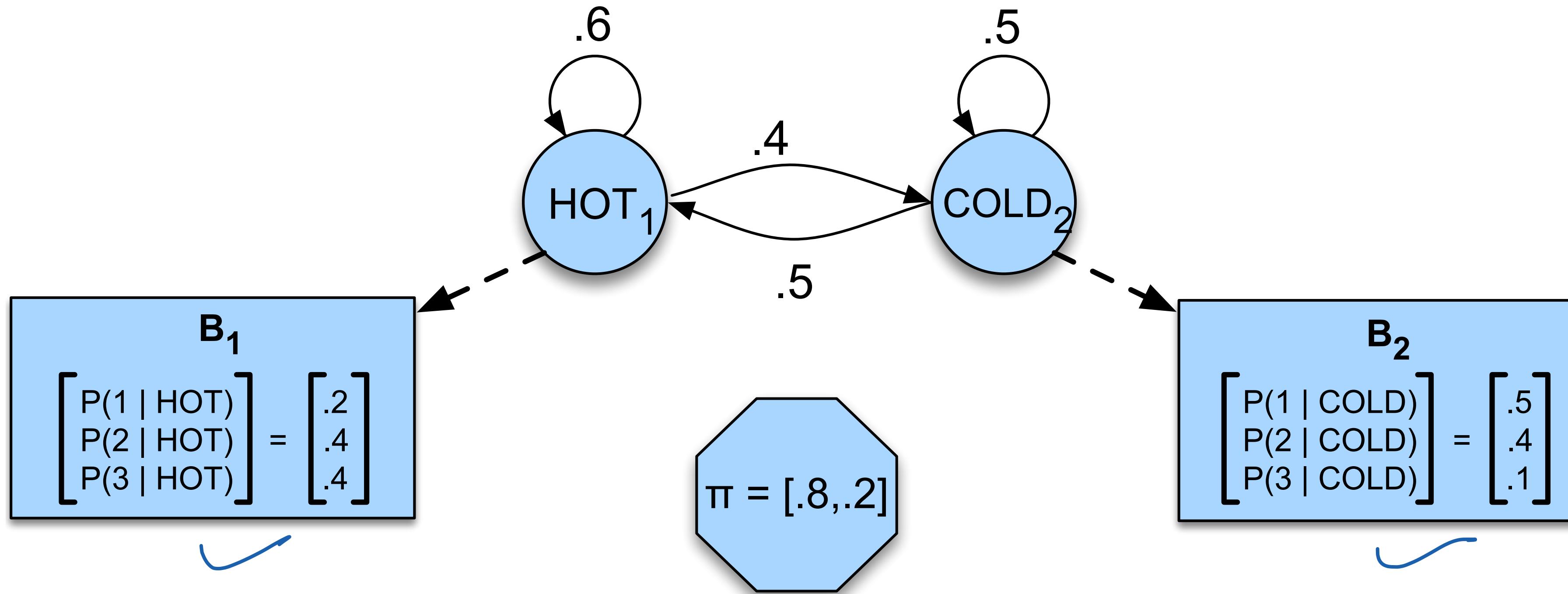
a **transition probability matrix**  $A$ , each  $a_{ij}$  representing the probability of moving from state  $i$  to state  $j$ , s.t.  $\sum_{j=1}^N a_{ij} = 1 \quad \forall i$

a sequence of  $T$  **observations**, each one drawn from a vocabulary  $V = v_1, v_2, \dots, v_V$

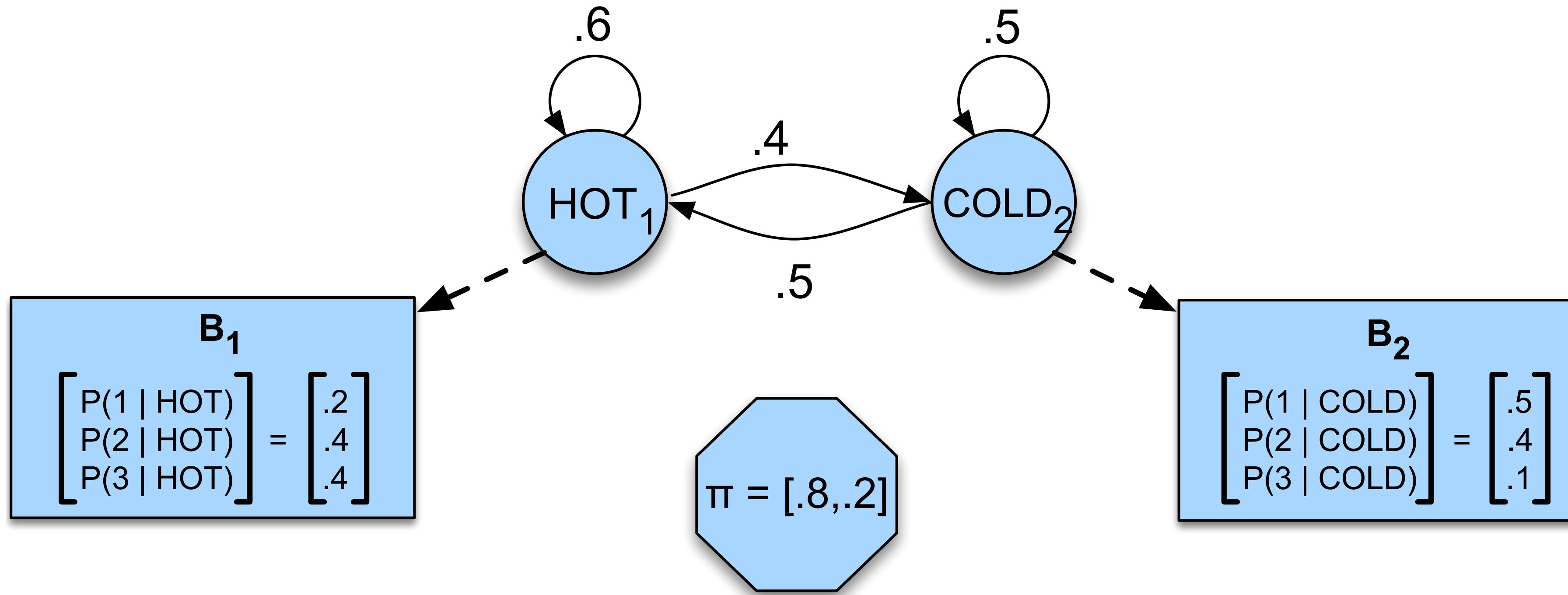
a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation  $o_t$  being generated from a state  $i$

an **initial probability distribution** over states.  $\pi_i$  is the probability that the Markov chain will start in state  $i$ . Some states  $j$  may have  $\pi_j = 0$ , meaning that they cannot be initial states. Also,  $\sum_{i=1}^n \pi_i = 1$

# HMM Assumptions

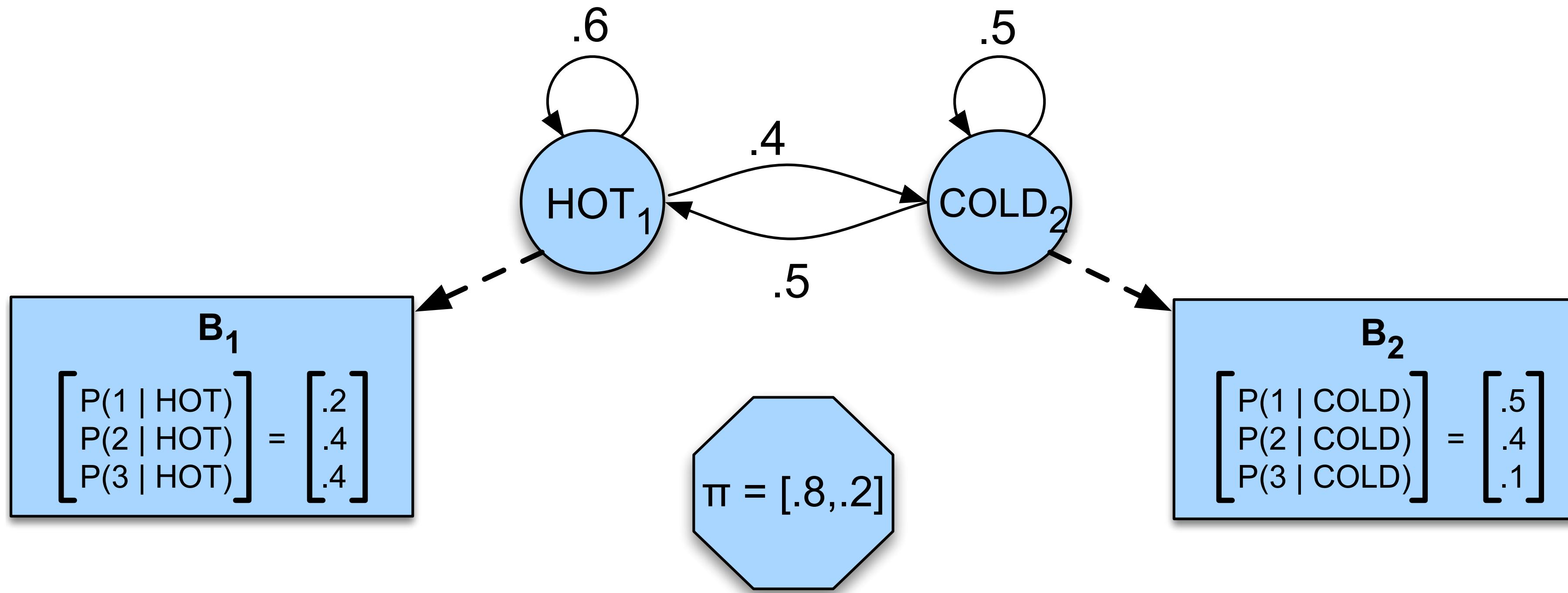


# HMM Assumptions



**Markov Assumption:**  $P(\underline{q_i} | q_1 \dots \underline{q_{i-1}}) = P(q_i | q_{i-1})$

# HMM Assumptions



**Markov Assumption:**  $P(q_i | q_1 \dots q_{i-1}) = P(q_i | q_{i-1})$

**Output Independence:**  $P(\underline{o_i} | q_1 \dots \underline{q_i}, \dots, q_T, o_1, \dots, \underline{o_{i-1}}, \dots, o_T) = P(\underline{o_i} | \underline{q_i})$

# Three problems for HMMs

**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

# Three problems for HMMs

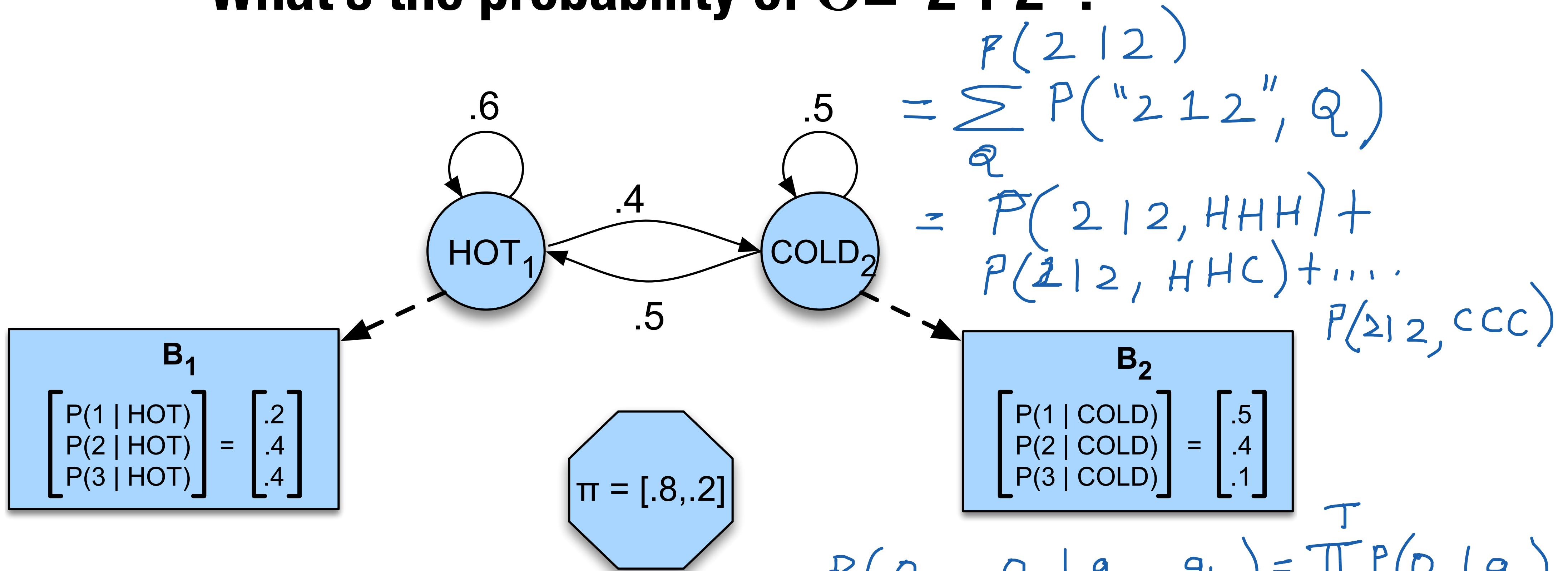
**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

**Computing Likelihood:** Given an HMM  $\lambda = \underline{(A, B)}$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

# What's the probability of $O = "2 1 2"$ ?



$$\begin{aligned}
 P(2|2) &= \sum_Q P("2 1 2", Q) \\
 &= P(2|2, \text{HHH}) + \\
 &\quad P(2|2, \text{HHC}) + \dots \\
 &\quad P(2|2, \text{CCC})
 \end{aligned}$$

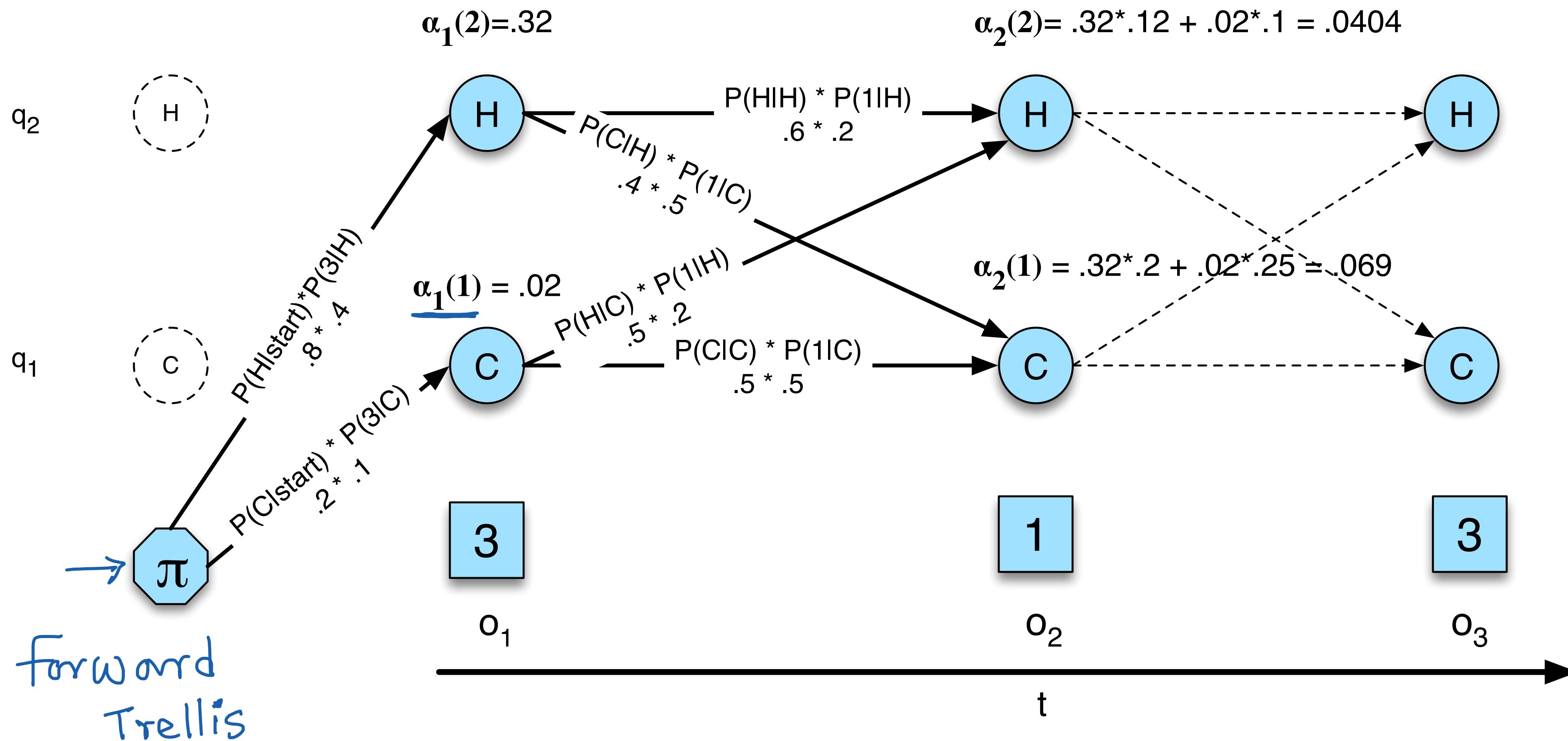
$$P(O_1 \dots O_T | q_1 \dots q_T) = \prod_{t=1}^T P(O_t | q_t)$$

$$P("2 1 2" | \lambda)$$

$$\begin{aligned}
 &P("2 1 2" | "H C H") \\
 &= P(2|H) P(1|C) P(2|H)
 \end{aligned}$$

emission  
probs

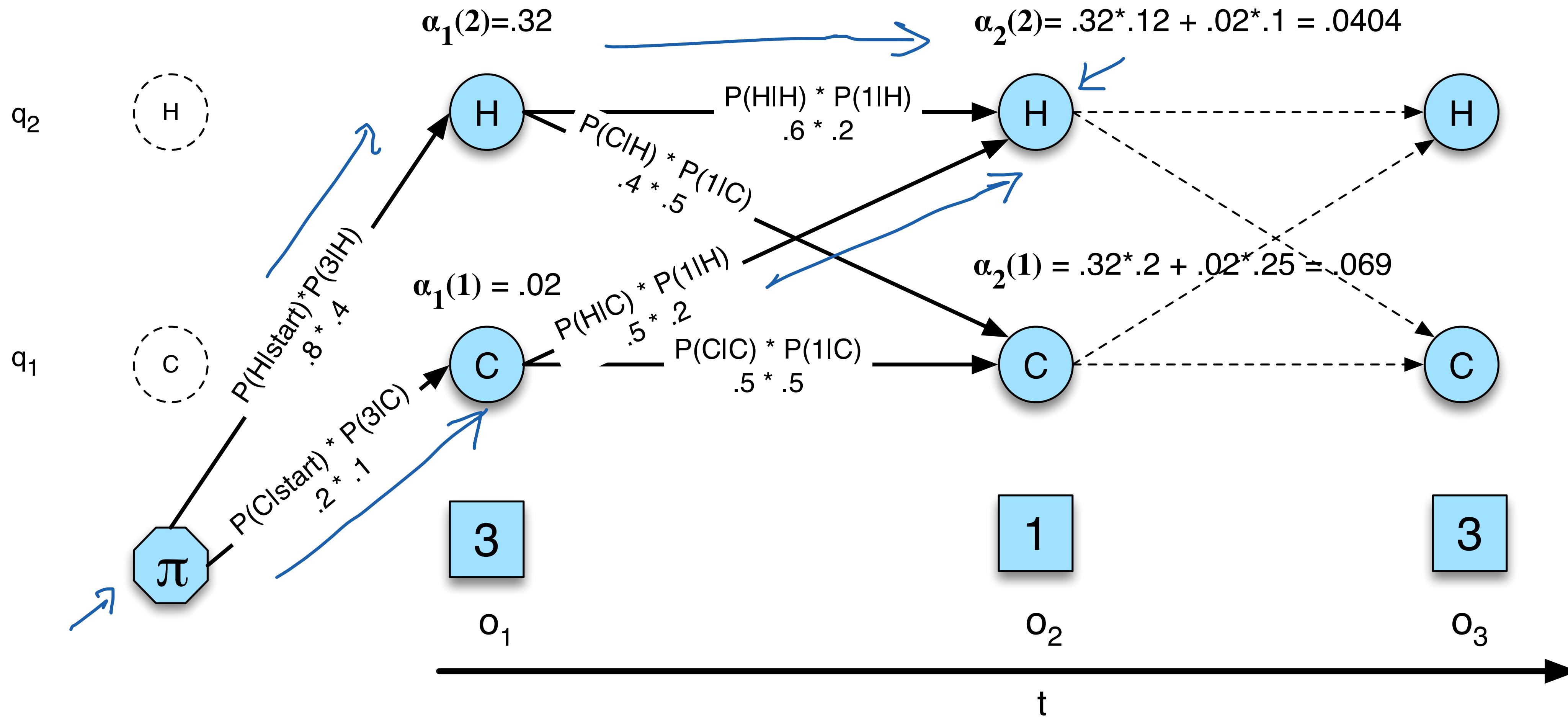
# Forward Algorithm



# Forward Algorithm

Forward Probability

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = j | \lambda)$$

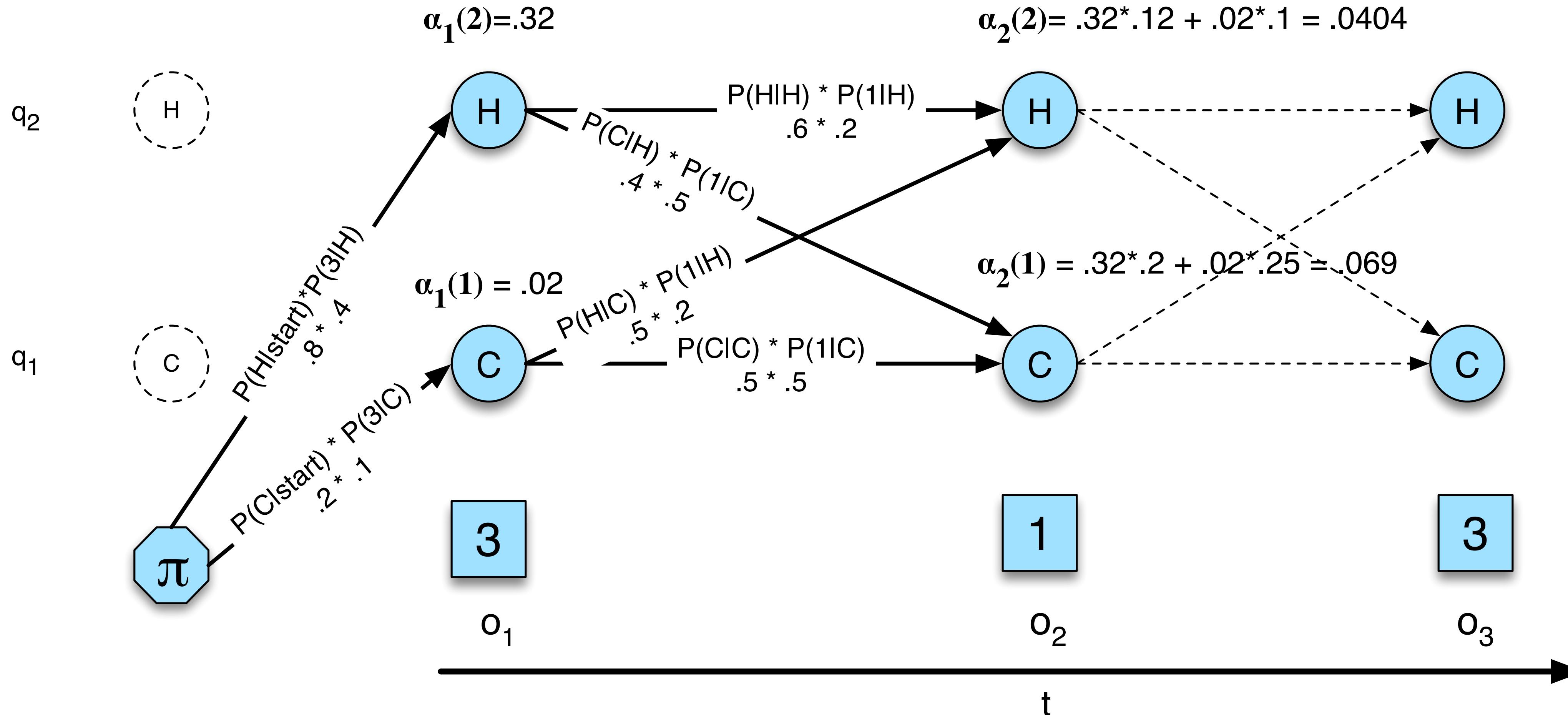


# Forward Algorithm

**Forward Probability**

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = j | \lambda)$$

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t)$$



# Forward Algorithm

Forward Probability

$$\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j | \lambda)$$

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

$$\alpha_t(j) = P(o_1, o_2, \dots, o_t, q_t = j | \lambda)$$

$$= \sum_i P(o_1, o_2, \dots, o_t, q_t = j, q_{t-1} = i | \lambda)$$

$$= \sum_i P(o_t | o_{1:t-1}, q_t = j, q_{t-1} = i).$$

$$P(o_{1:t-1}, q_t = j, q_{t-1} = i)$$

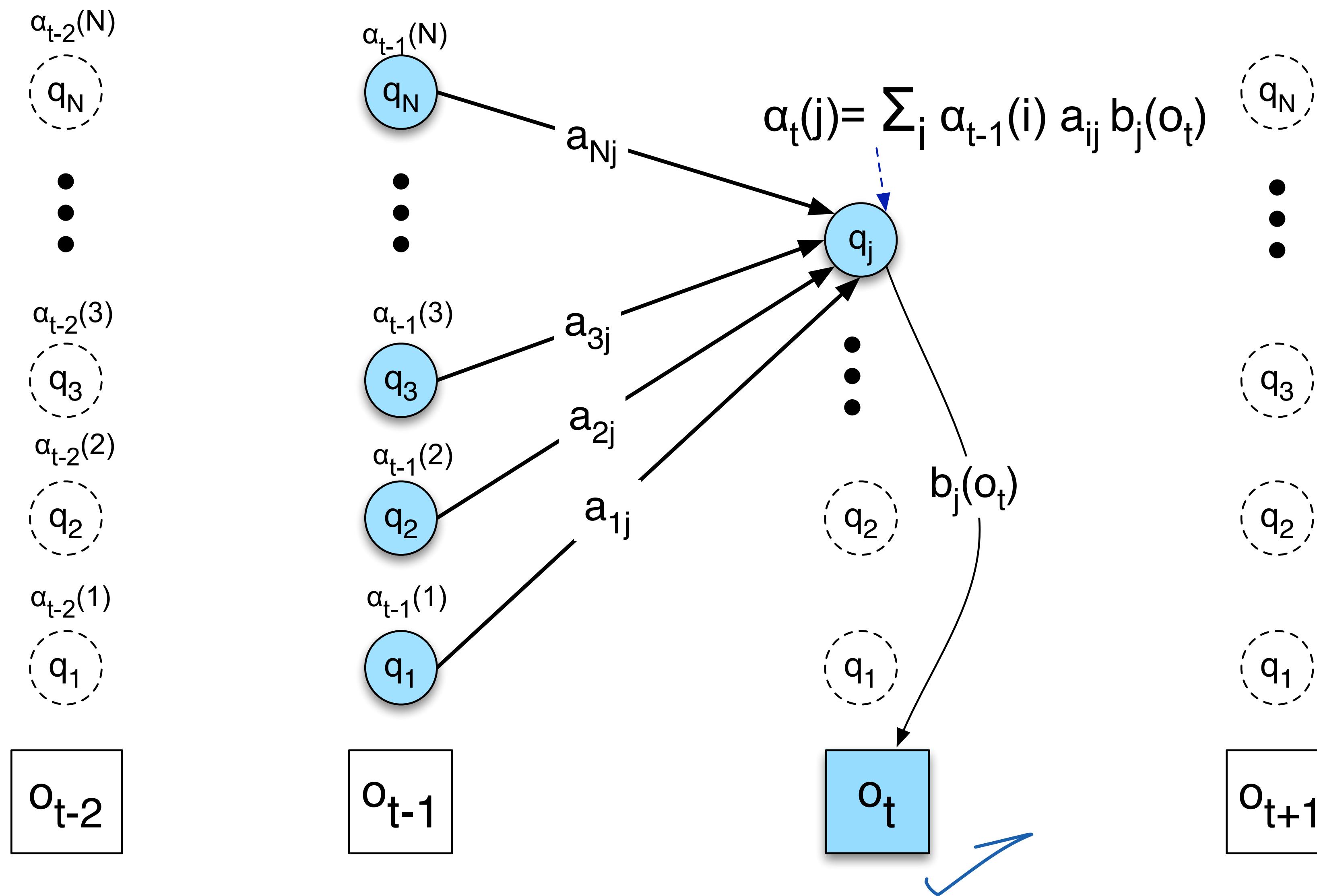
$$= \sum_i P(o_t | q_t = j) P(q_t = j | q_{t-1} = i, o_{1:t-1}) P(q_{t-1} = i, o_{1:t-1})$$

$$= \sum_i P(o_t | q_t = j) P(q_t = j | q_{t-1} = i) \alpha_{t-1}(i)$$

$$\downarrow b_j(o_t)$$

$$\downarrow a_{ij}$$

# Visualizing the forward recursion



# Forward Algorithm

1. Initialization:

$$\alpha_1(j) = \pi_j b_j(o_1) \quad 1 \leq j \leq N$$

2. Recursion:

$$\alpha_t(j) = \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

3. Termination:

$$\underline{P(O|\lambda)} = \sum_{i=1}^N \alpha_{\underline{T}}(i)$$

# Three problems for HMMs

**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

# Three problems for HMMs

**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

**Decoding:** Given as input an HMM  $\lambda = (A, B)$  and a sequence of observations  $O = o_1, o_2, \dots, o_T$ , find the most probable sequence of states  $Q = q_1 q_2 q_3 \dots q_T$ .

# Viterbi Trellis

"Viterbi Algorithm"

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(\underbrace{q_1 \dots q_{t-1}}_{\text{Viterbi Path}}, \underbrace{o_1, o_2 \dots o_t}_{\text{Observations}}, q_t = j | \lambda)$$

Viterbi Path  
Probability

# Viterbi Trellis

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j | \lambda)$$

Viterbi Path  
Probability

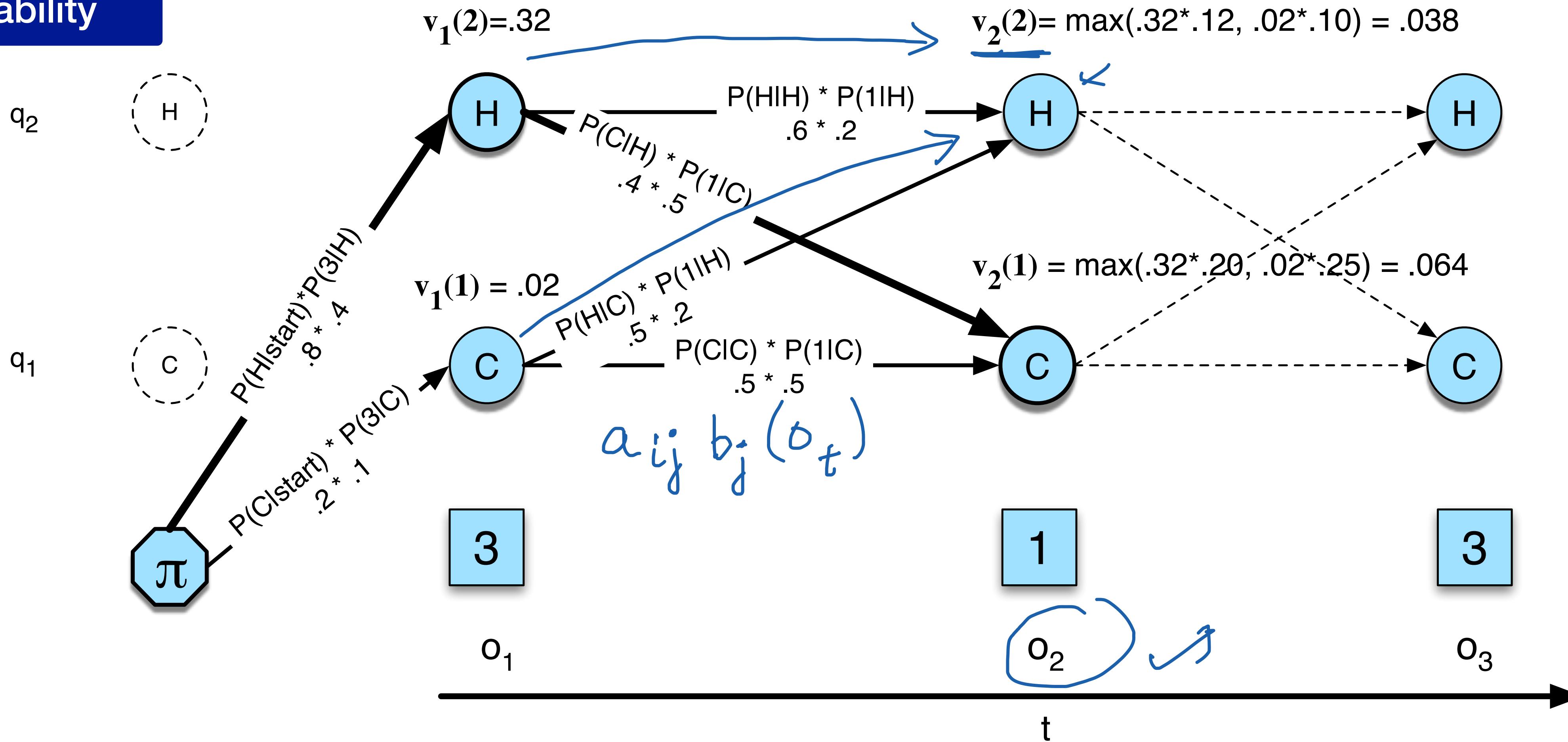
$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t)$$

# Viterbi Trellis

$$v_t(j) = \max_{q_1, \dots, q_{t-1}} P(q_1 \dots q_{t-1}, o_1, o_2 \dots o_t, q_t = j | \lambda)$$

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) \underbrace{a_{ij} b_j(o_t)}$$

Viterbi Path Probability



# Viterbi recursion

## 1. Initialization:

$$\begin{aligned} v_1(j) &= \pi_j b_j(o_1) & 1 \leq j \leq N \\ \underline{bt_1(j)} &= 0 & 1 \leq j \leq N \end{aligned}$$

## 2. Recursion

$$v_t(j) = \max_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

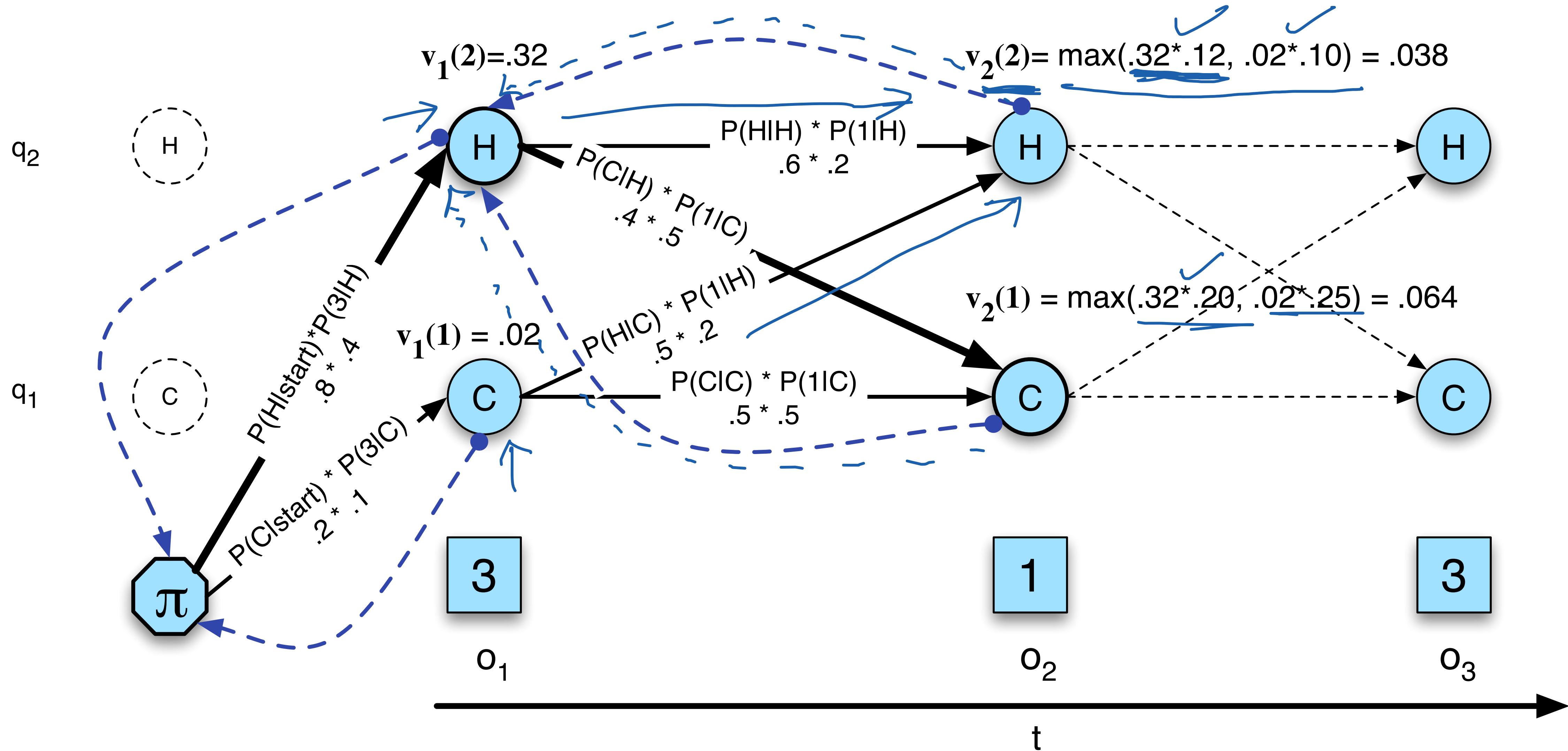
$$bt_t(j) = \operatorname{argmax}_{i=1}^N v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

## 3. Termination:

The best score:  $P* = \max_{i=1}^N v_T(i)$

The start of backtrace:  $q_T* = \operatorname{argmax}_{i=1}^N v_T(i)$

# Viterbi backtrace



# Next Module: Learning in HMMs

**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

**Learning:** Given an observation sequence  $O$  and the set of possible states in the HMM, learn the HMM parameters  $A$  and  $B$ .

# Next Module: Learning in HMMs

**Problem 1 (Likelihood):** Given an HMM  $\lambda = (A, B)$  and an observation sequence  $O$ , determine the likelihood  $P(O|\lambda)$ .

**Problem 2 (Decoding):** Given an observation sequence  $O$  and an HMM  $\lambda = (A, B)$ , discover the best hidden state sequence  $Q$ .

**Problem 3 (Learning):** Given an observation sequence  $O$  and the set of states in the HMM, learn the HMM parameters  $A$  and  $B$ .

**Learning:** Given an observation sequence  $O$  and the set of possible states in the HMM, learn the HMM parameters  $A$  and  $B$ .

HMM training: **Forward-backward or Baum-Welch algorithm**