Announcements

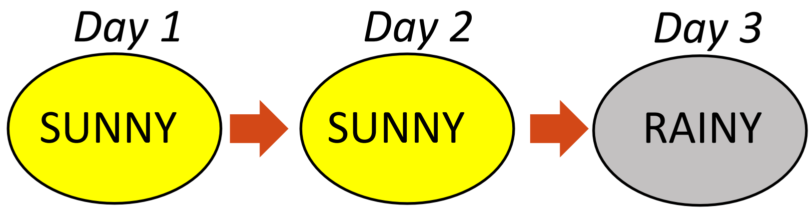
- pset2 due today.
- pset 3 to release today, due in 3 weeks (3/27)
- No classes next week (Spring break)

Last time

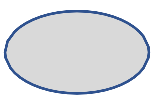
- Markov Chain
- Hidden Markov Model
- Decoding HMMs

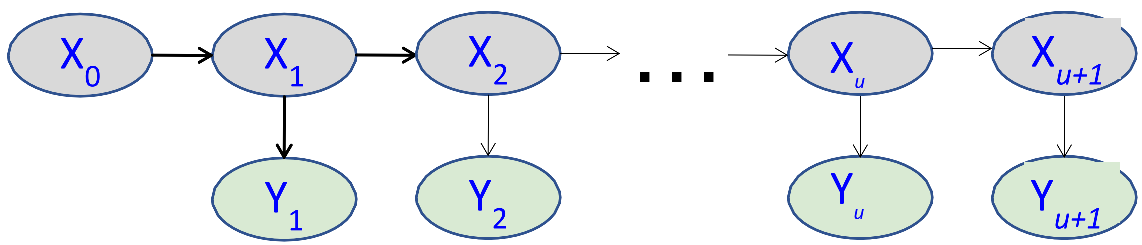
Recall: Markov vs Hidden

Markov



Hidden

 *unknown*



Which of the following statements is true about speech recognition? Select all that apply.

Which of the following statements is true about speech recognition? Select all that apply.

Acoustic signals are observed variables and phonemes are the unobserved variables

90%

Phonemes are observed variables and acoustic signals are unobserved variables.

15%

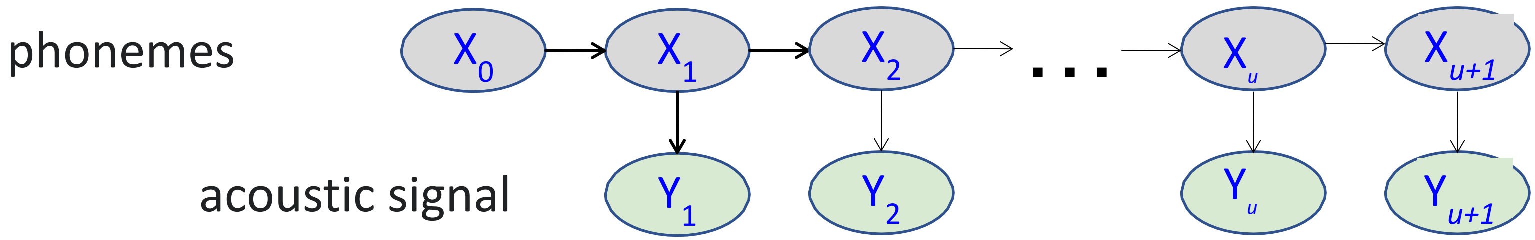
Transition probability corresponds to probability of going from one observed state to another

62%

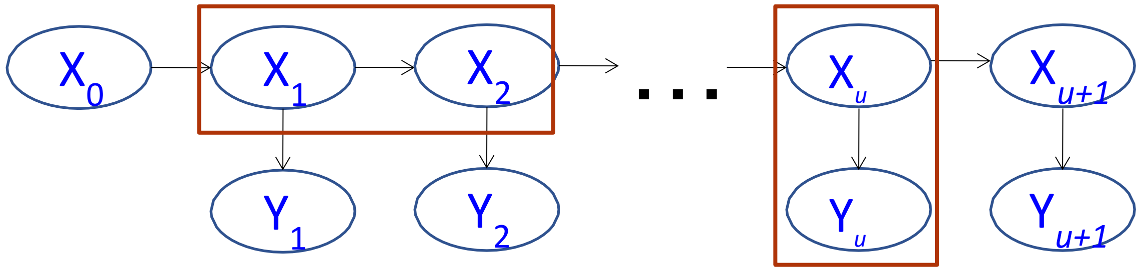
Transition probability corresponds to probability of going from one unobserved state to another

46%

Popular use case: Speech to text



The Joint Distribution



- Transition model: $P(X_{u+1} = j | X_u = i)$
- Observation model: $P(Y_u | X_u = i)$
- How do we compute the full joint probability table $P(X_{0:u+1} | Y_{0:u+1})$?

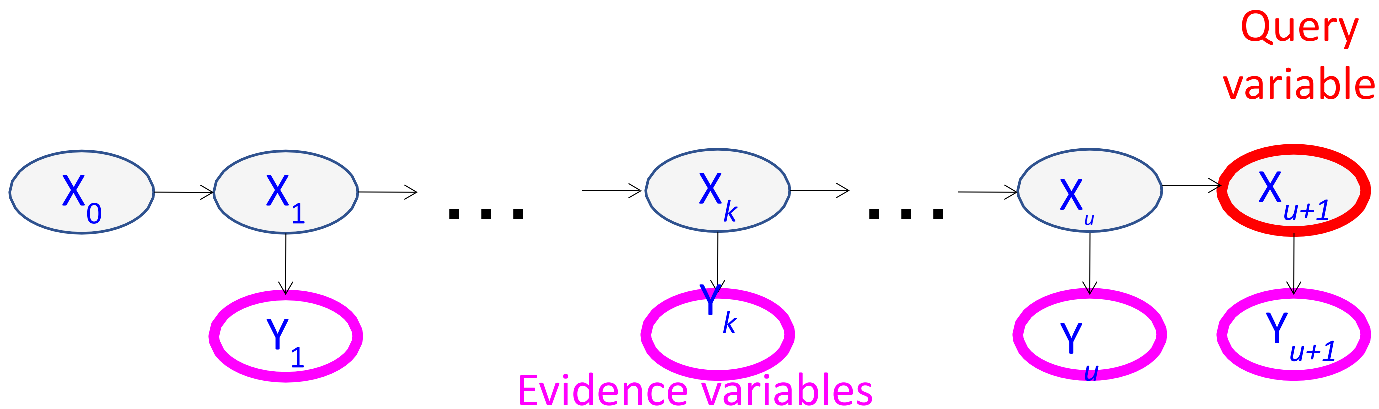
Bayes' Theorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$$P(X_{0:u+1} | Y_{0:u+1}) = P(X_0) \prod_{i=1}^{u+1} P(X_i | X_{i-1}) P(Y_i | X_i)$$

HMM inference tasks

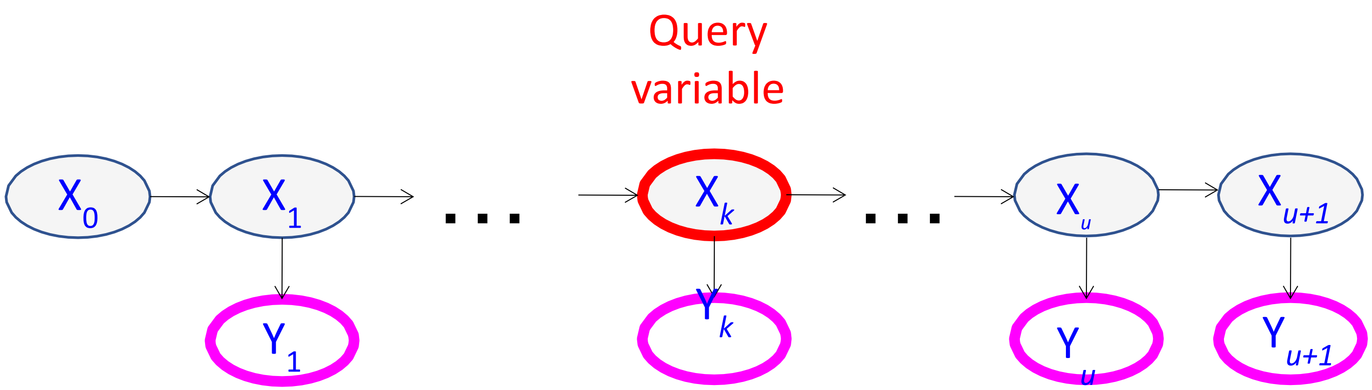
- **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{Y}_{1:t}$?   (example: is it currently raining?)



We use forward algorithm

HMM inference tasks

- **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{Y}_{1:t}$?
- **Smoothing:** what is the distribution of some state X_k (k<t) given the entire observation sequence $\mathbf{Y}_{1:t}$? (example: did it rain on Sunday?)



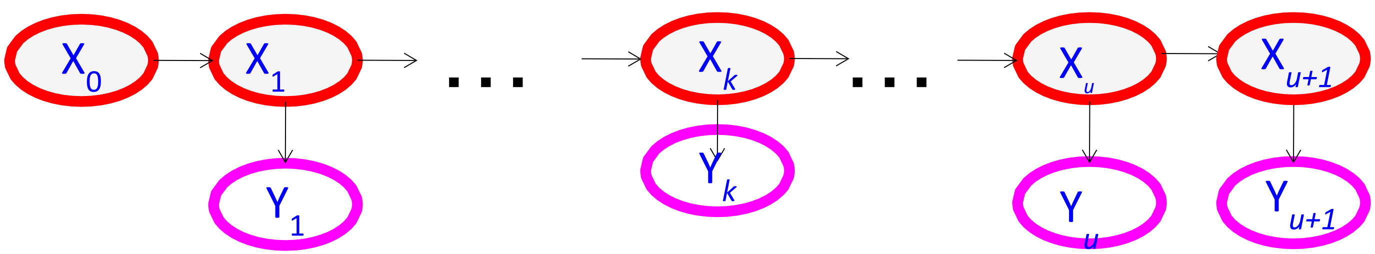
- We use backward algorithm

HMM inference tasks

- **Inference tasks**
  - **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{Y}_{1:t}$
  - **Smoothing:** what is the distribution of some state X_k (k<t) given the entire observation sequence $\mathbf{Y}_{1:t}$?
  - **Evaluation:** compute the probability of a given observation sequence $\mathbf{Y}_{1:t}$
  - **Decoding:** what is the most likely state sequence $\mathbf{X}_{0:t}$ given the observation sequence $\mathbf{Y}_{1:t}$?

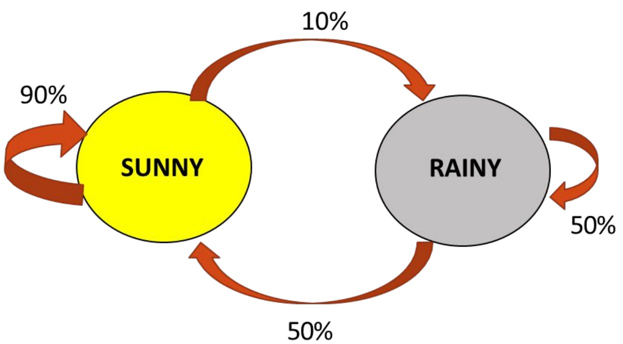
HMM inference tasks

- **Filtering:** what is the distribution over the current state X_t given all the evidence so far, $\mathbf{Y}_{1:t}$
- **Smoothing:** what is the distribution of some state X_k (k<t) given the entire observation sequence $\mathbf{Y}_{1:t}$?
- **Evaluation:** compute the probability of a given observation sequence $\mathbf{Y}_{1:t}$
- **Decoding:** what is the most likely state sequence $\mathbf{X}_{0:t}$ given the observation sequence $\mathbf{Y}_{1:t}$? (example: what's the weather every day?)



Recall: Transitions

Transition Matrix:

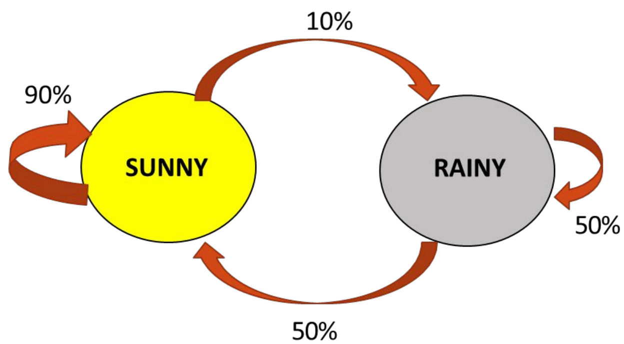


	Sunny	Rainy
Sunny	0.9	0.5
Rainy	0.1	0.5

- What is the most likely weather in 3 days?

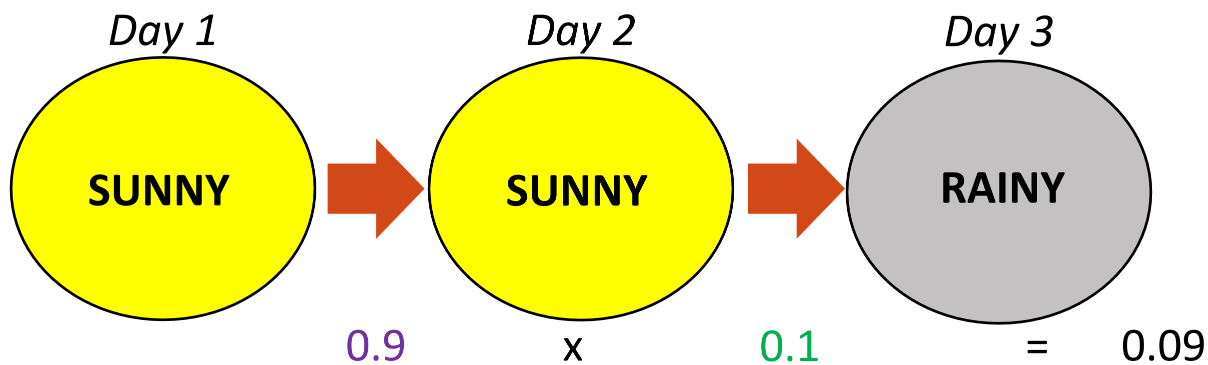
Option#1: Brute force

Transition Matrix:

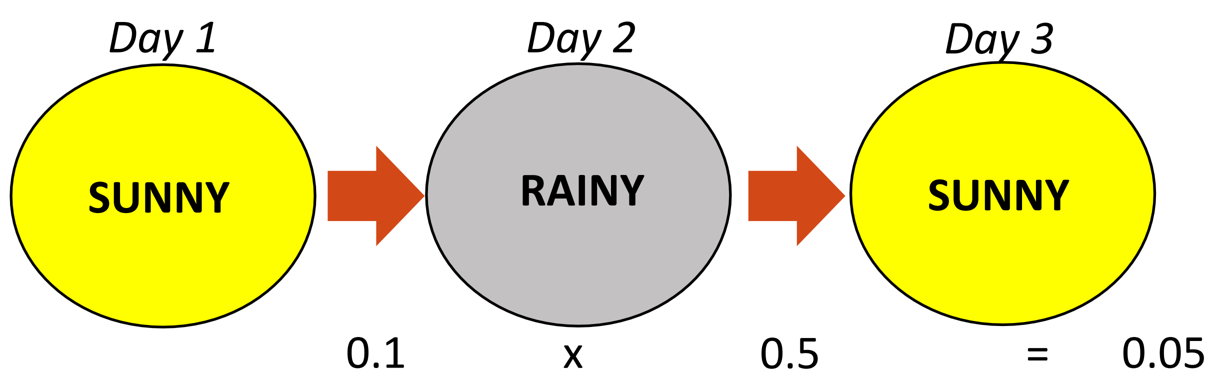


	Sunny	Rainy
Sunny	0.9	0.5
Rainy	0.1	0.5

Sequence A:

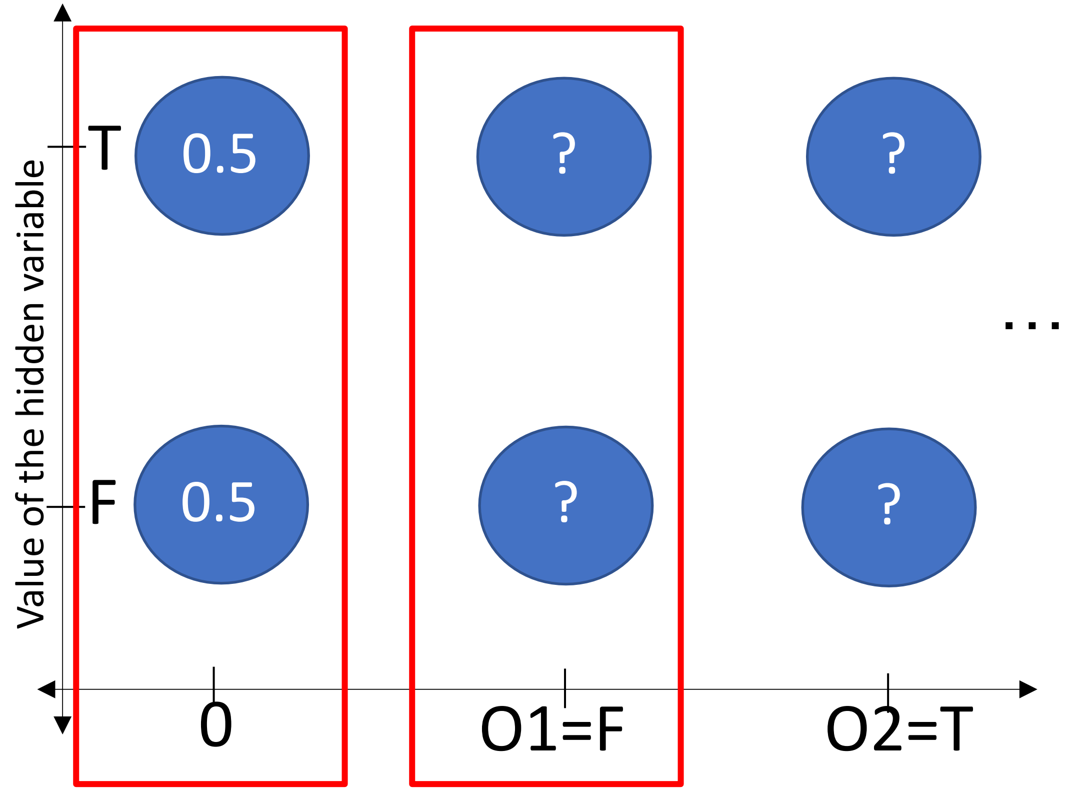


Sequence B:



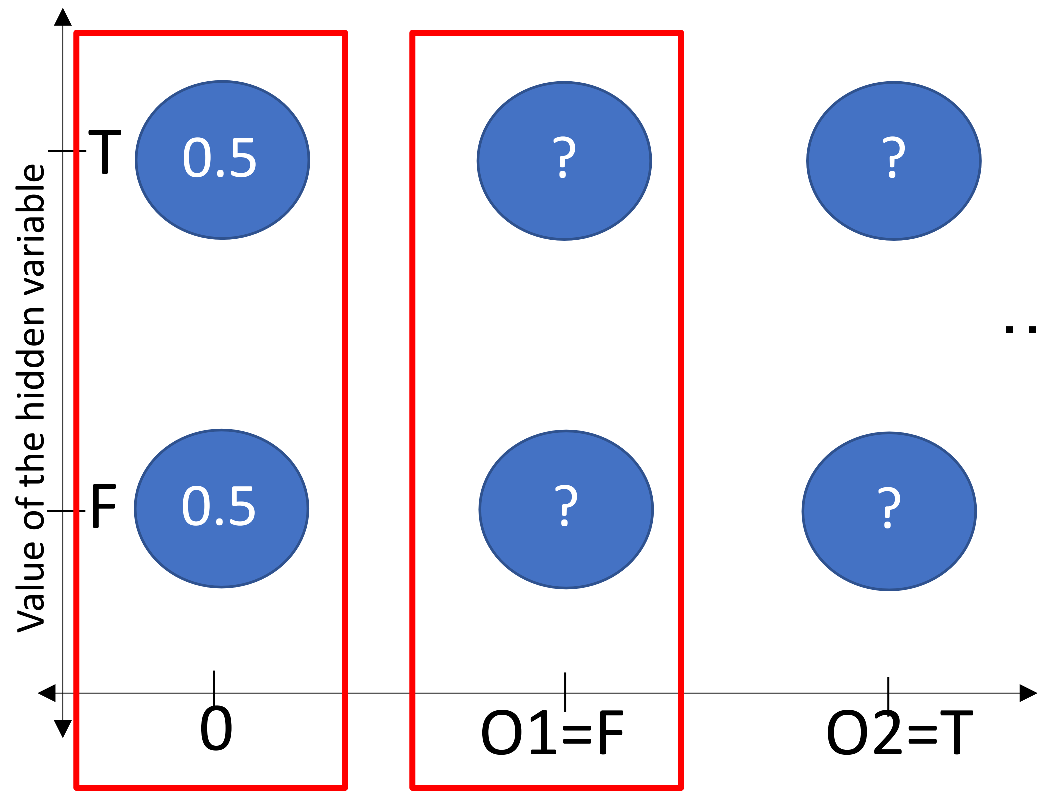
Option#2: Viterbi algorithm

- **Trellis graph:** Unpack the markov chain.
  - Nodes are ordered into vertical slices.
  - A node from time t-1 connected to node from t.



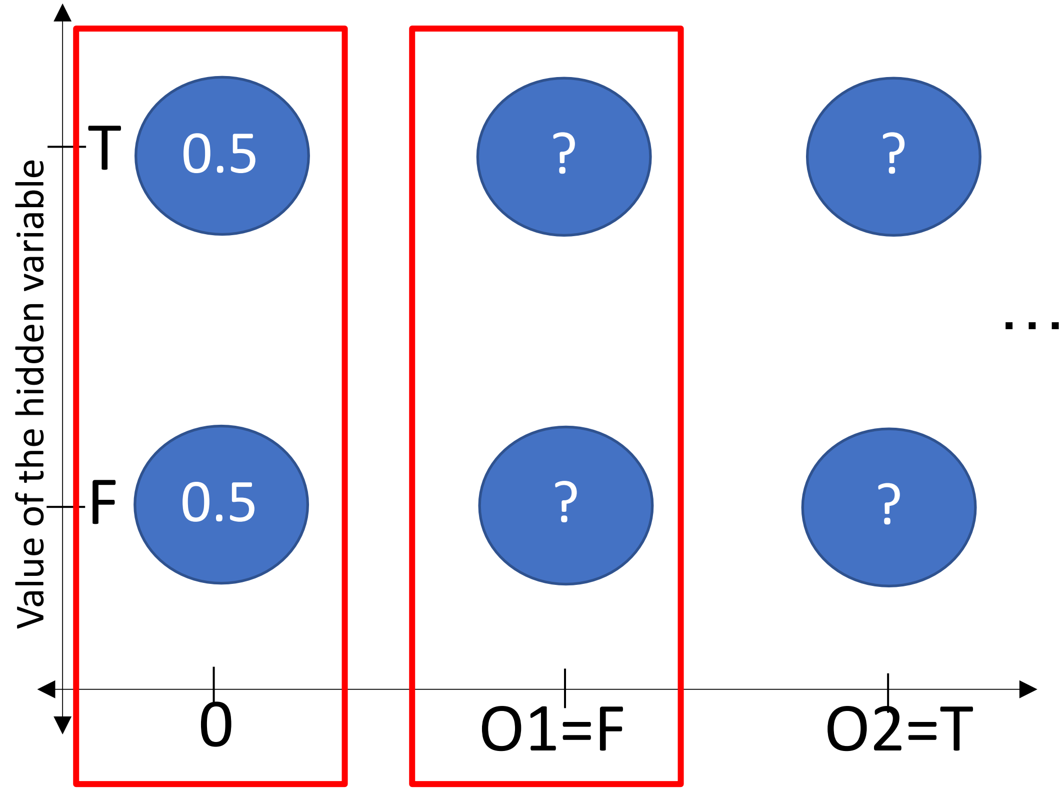
Option#2: Viterbi algorithm

- **Trellis graph:** Unpack the markov chain.
  - **Node** = a value of the hidden variable at a given time
  - **Numerical value of the node** = probability that the hidden variable takes that value



Option#2: Viterbi algorithm

- **Trellis graph:** Unpack the markov chain.
  - **Edge** = a possible transition
  - **Numerical value of the edge** = transition probability.



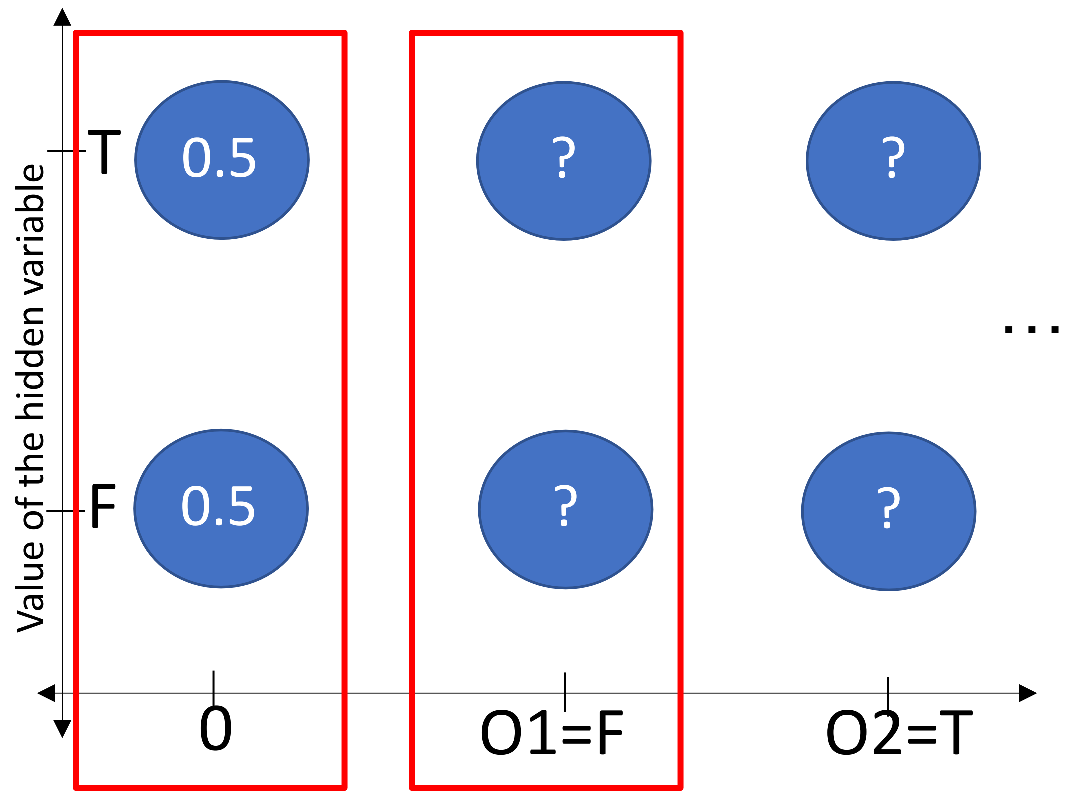
Trellis Graph

- Node = a value of the hidden variable at a given time
- Numerical value of the node = probability that the hidden variable takes that value

- Edge = a possible transition
- Numerical value of the edge = transition probability.

Option#2: Viterbi algorithm

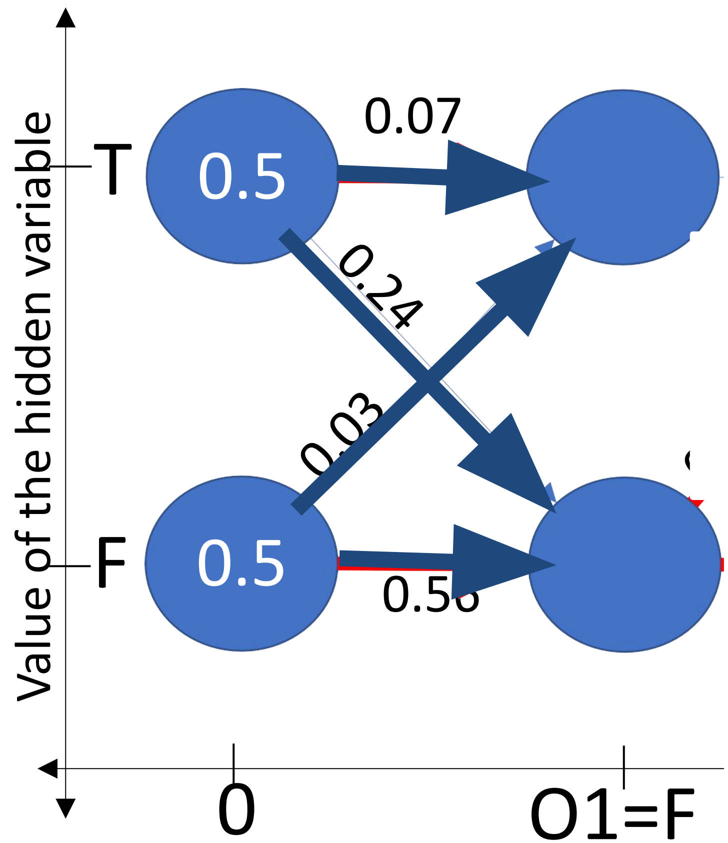
- **Trellis graph:** Unpack the markov chain.
  - Node = a value of the hidden variable at a given time
  - Numerical value of the node = probability that the hidden



- **Viterbi path:** Path in the trellis graph which has the most maximum likelihood.

Viterbi Algorithm: The Trellis

- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

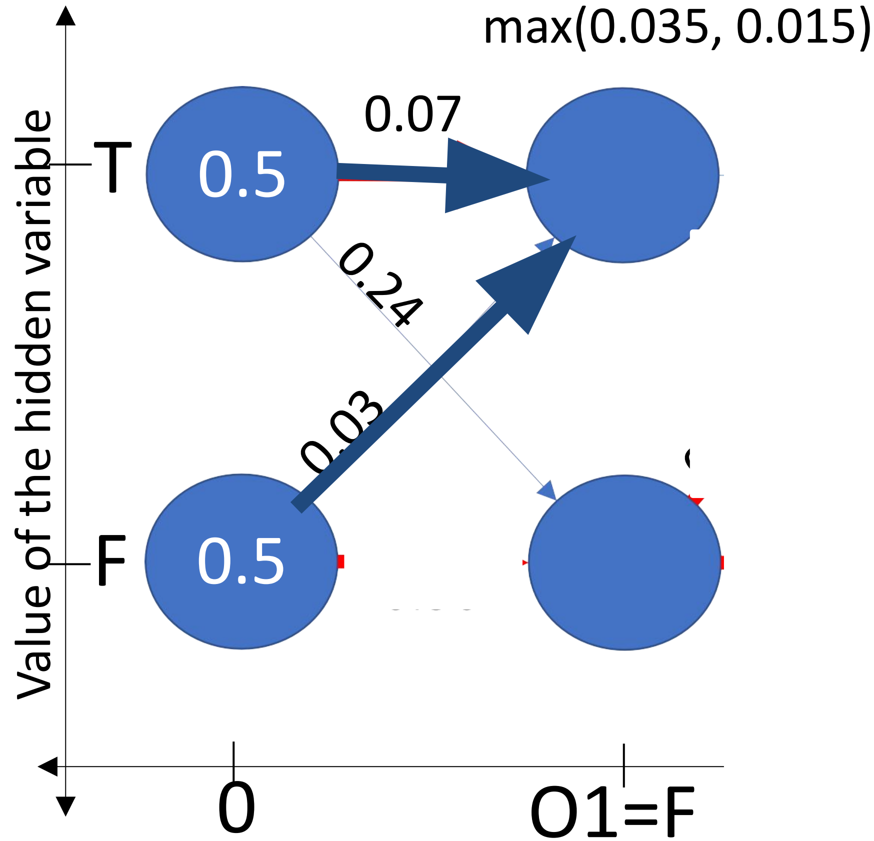


Viterbi Algorithm: The Trellis

- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Viterbi algorithm is:

$$v_{jt} = \max_i v_{i,t-1} e_{ijt}$$

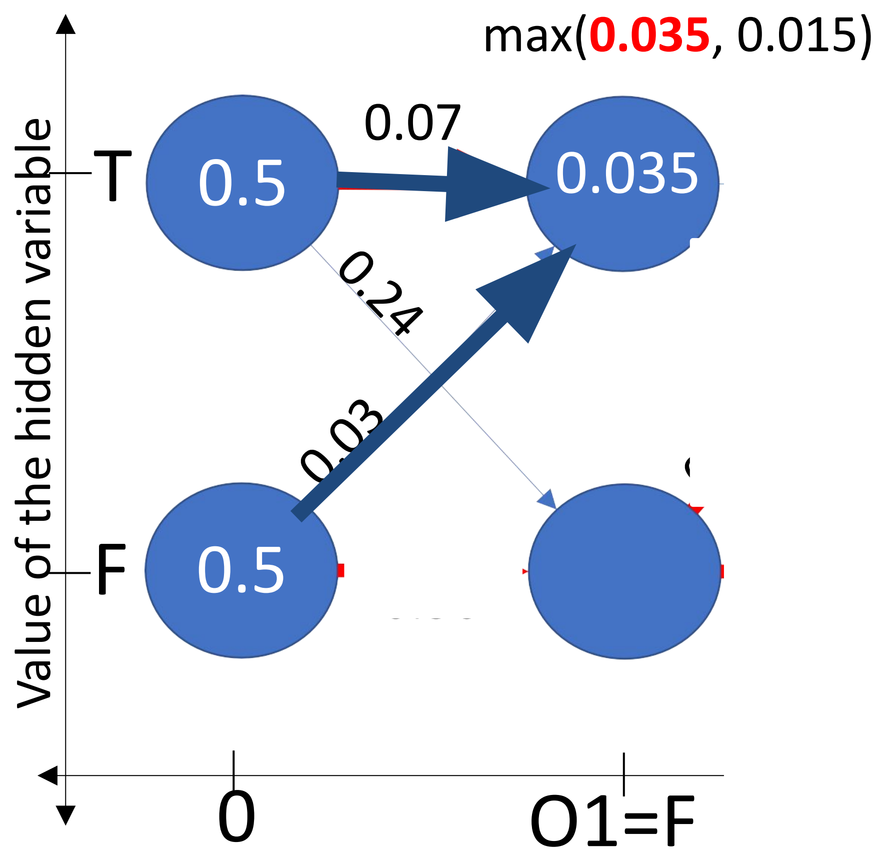


Viterbi Algorithm: The Trellis

- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Viterbi algorithm is:

$$v_{jt} = \max_i v_{i,t-1} e_{ijt}$$



Viterbi Algorithm: The Trellis

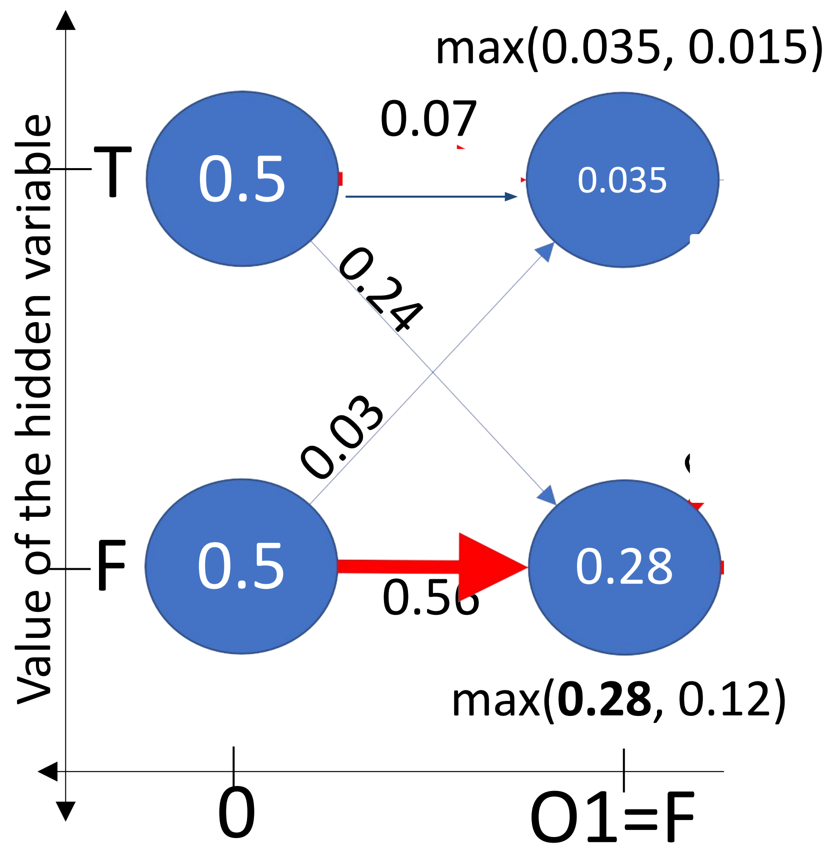
- v_{it} = value of i[th] node at time t
- e_{ijt} = edge connecting node $v_{i,t-1}$ to v_{jt}

Viterbi algorithm is:

$$v_{jt} = \max_i v_{i,t-1} e_{ijt}$$

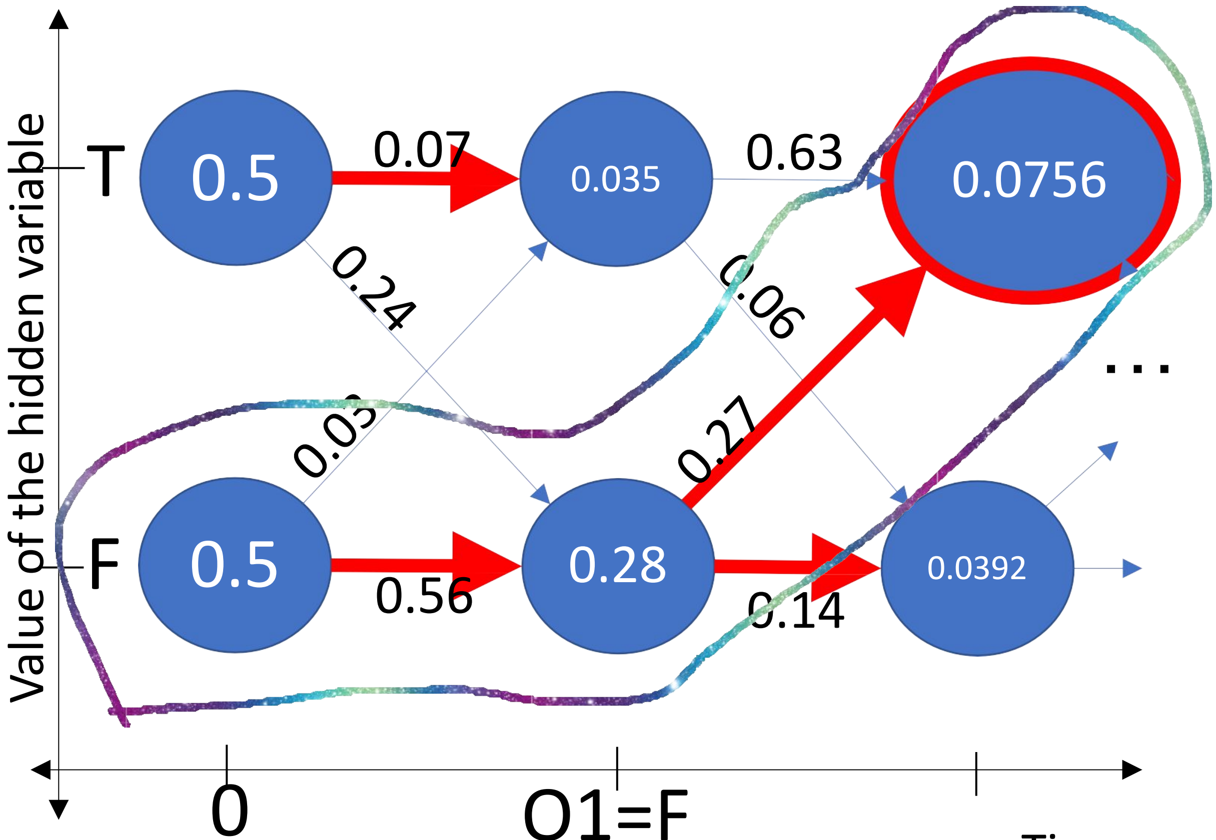
Backpointer is:

$$i^*(j,t) = \operatorname{argmax}_i v_{i,t-1} e_{ijt}$$



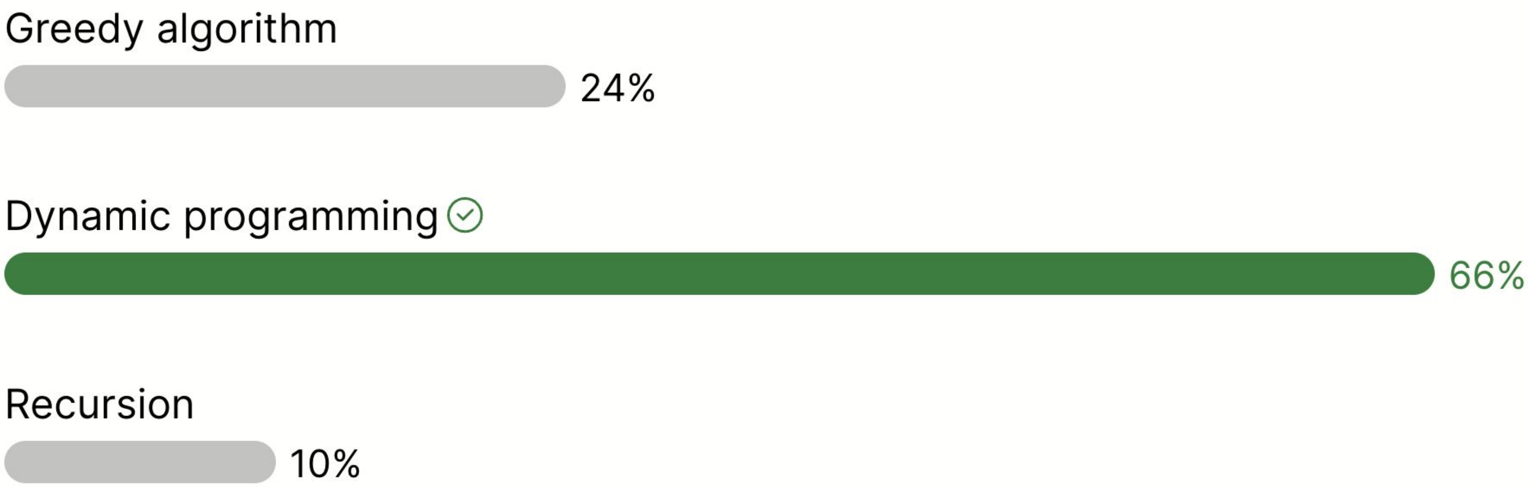
Viterbi Algorithm: Termination

Best path probability = 0.0756



What is the optimal optimization algorithm to use to predict Viterbi path?

What is the optimal optimization algorithm to use to predict Viterbi path?



Viterbi algorithm

1. **Initialization:**

$$\begin{aligned} v_1(j) &= a_{0j} b_j(o_1) \;\; 1 \le j \le N \\ bt_1(j) &= 0 \end{aligned}$$

2. **Recursion** (recall that states 0 and q_F are non-emitting):

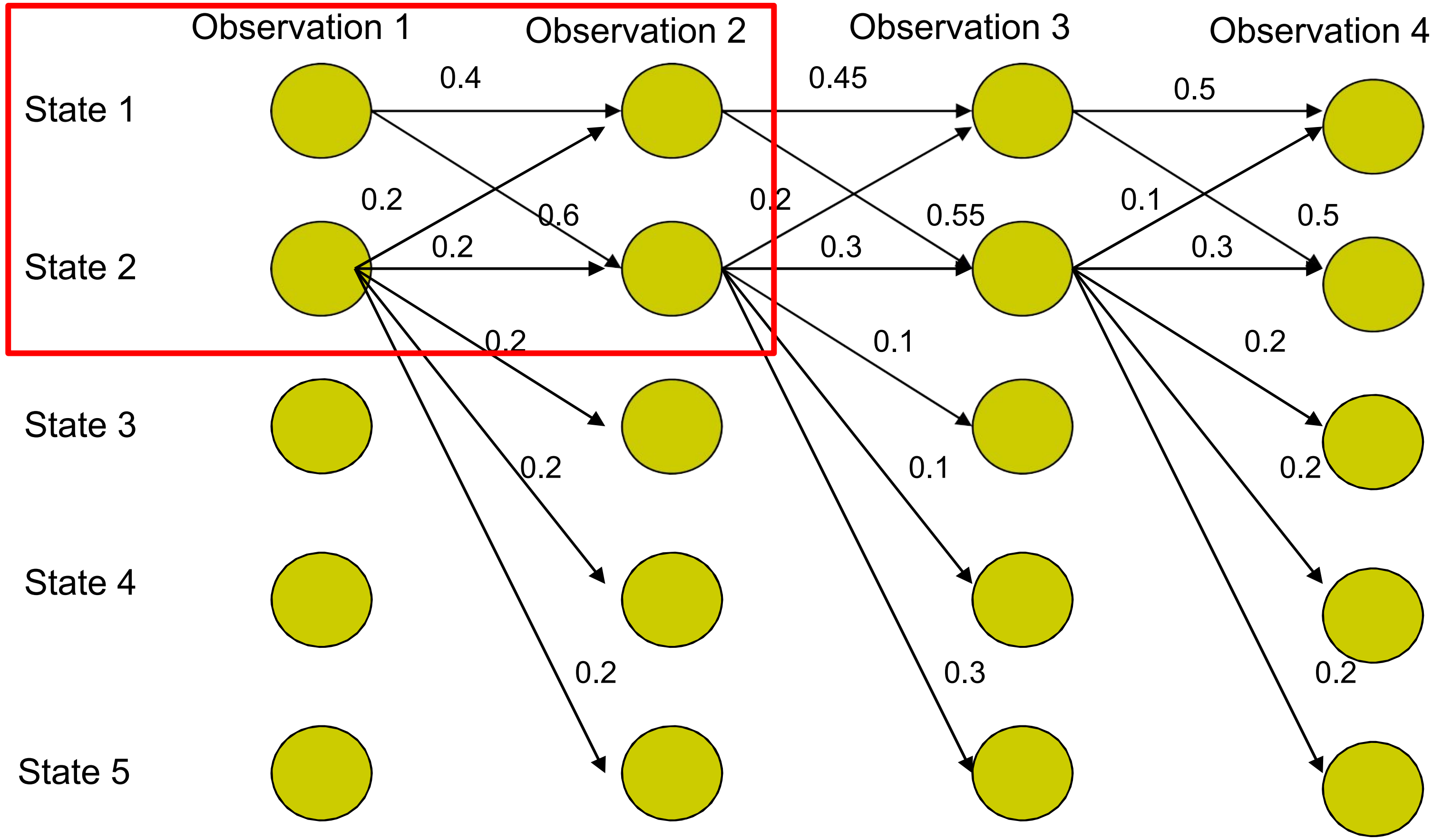
$$\begin{aligned} v_t(j) &= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \\ bt_t(j) &= \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \end{aligned}$$

3. **Termination:**

$$\text{The best score:} \quad P* = v_T(q_F) \;=\; \max_{i=1}^{N} v_T(i) * a_{iF}$$

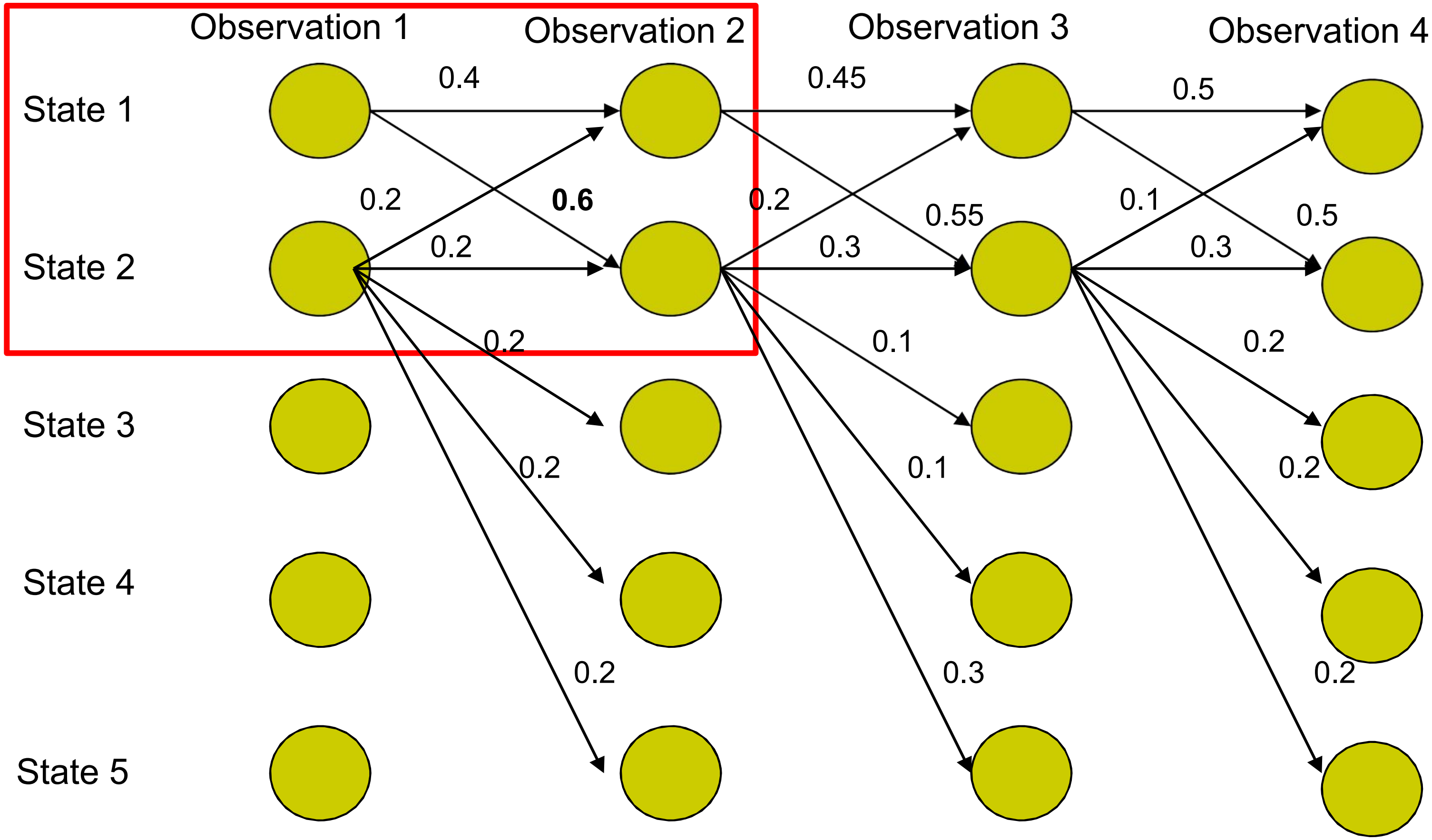
$$\text{The start of backtrace:} \quad q_T* = bt_T(q_F) \;=\; \operatorname*{argmax}_{i=1}^{N} v_T(i) * a_{iF}$$

Local transition probabilities



-

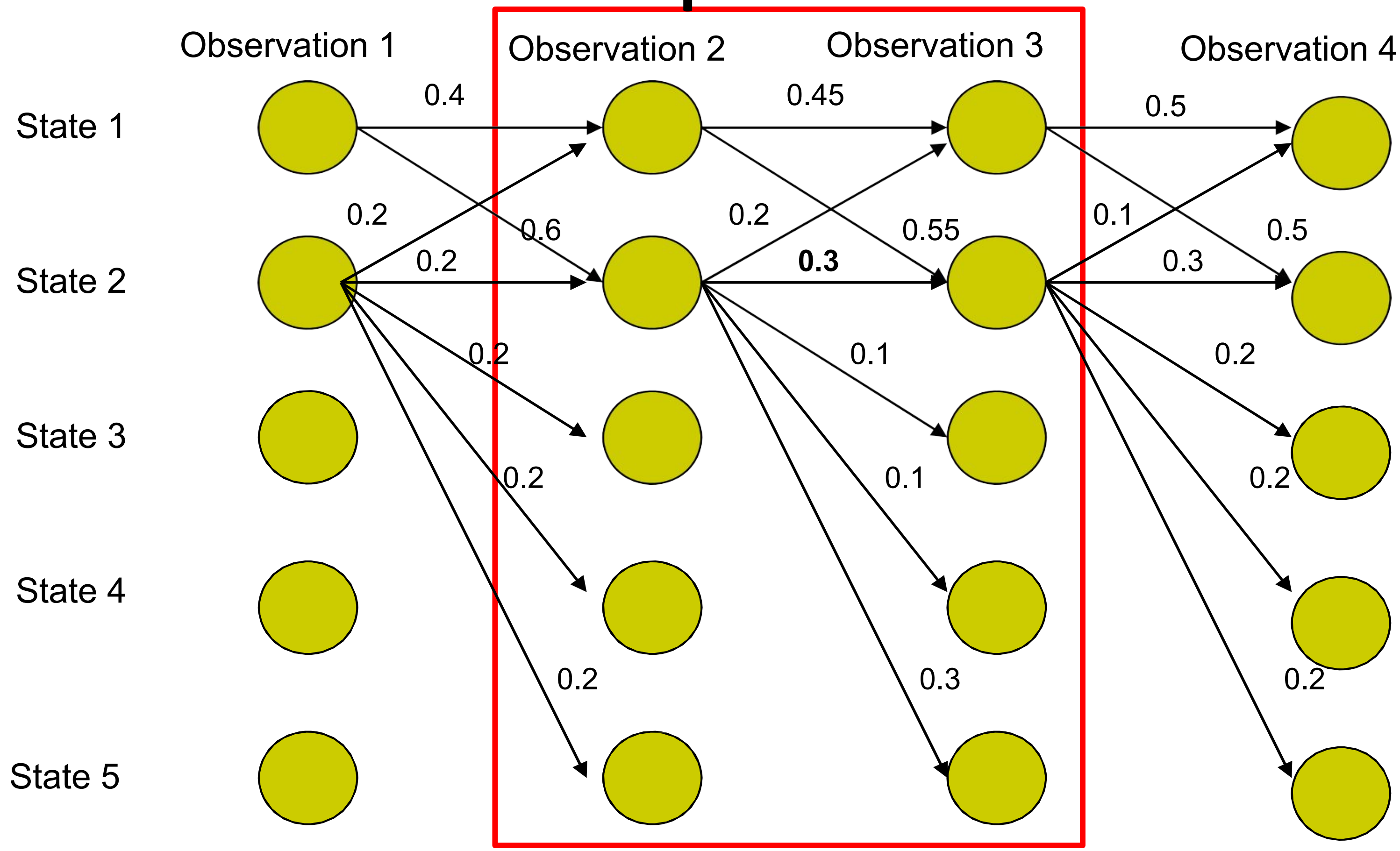
Local transition probabilities



What the local transition probabilities say:

- State 1 almost always prefers to go to state 2

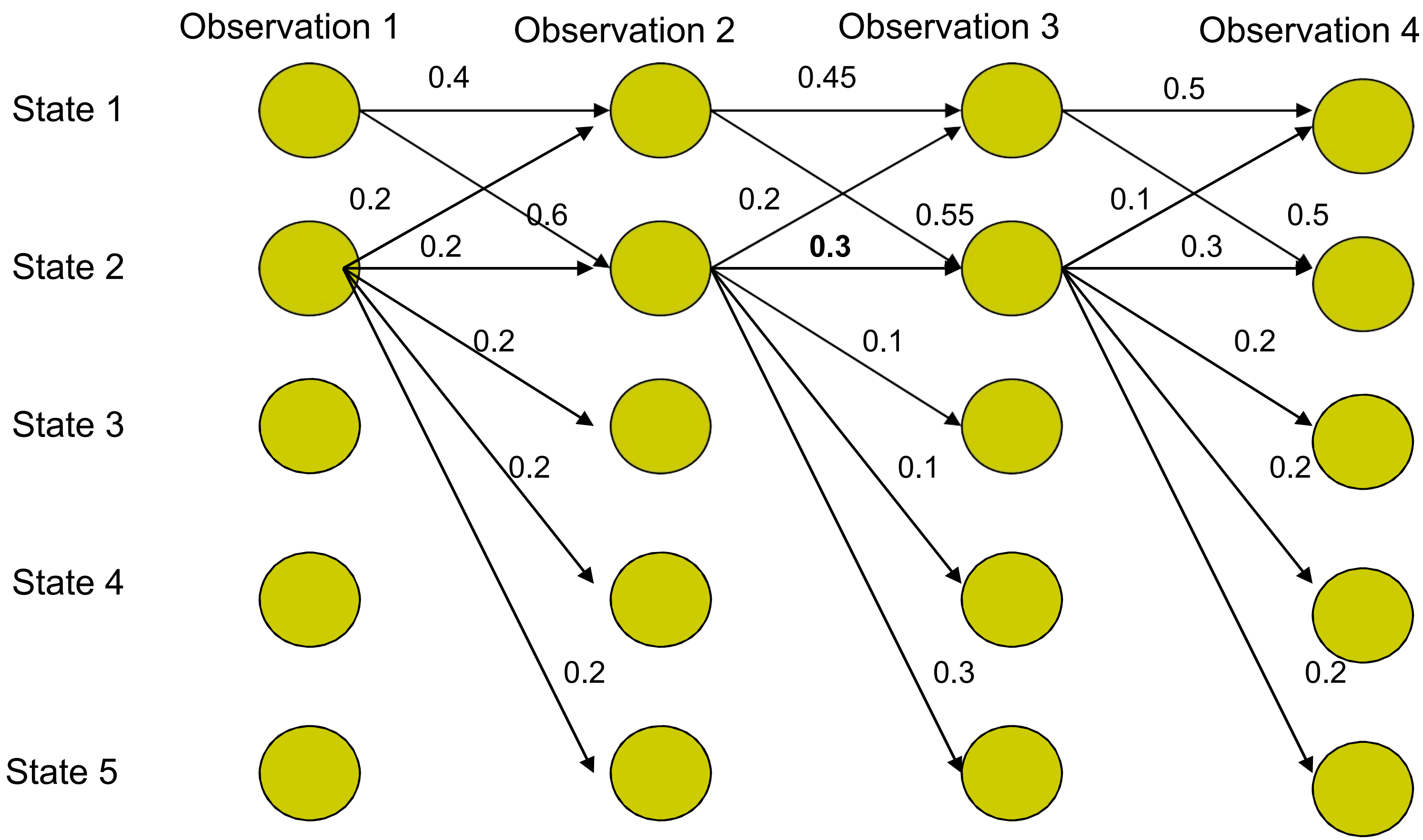
Local transition probabilities



What the local transition probabilities say:

- State 2 almost always prefers to stay in state 2

Local transition probabilities



Calculate probabilities for the following paths:

- Probability of path 1-> 1-> 1-> 1:
- Probability of path 2->2->2->2 :
- Probability of path 1->2->1->2:

⏰ (2 mins to compute)

Do not edit
How to change the design

Select all that apply

Presenting with animations, GIFs or speaker notes? Enable our Chrome extension

slido

Select all that apply

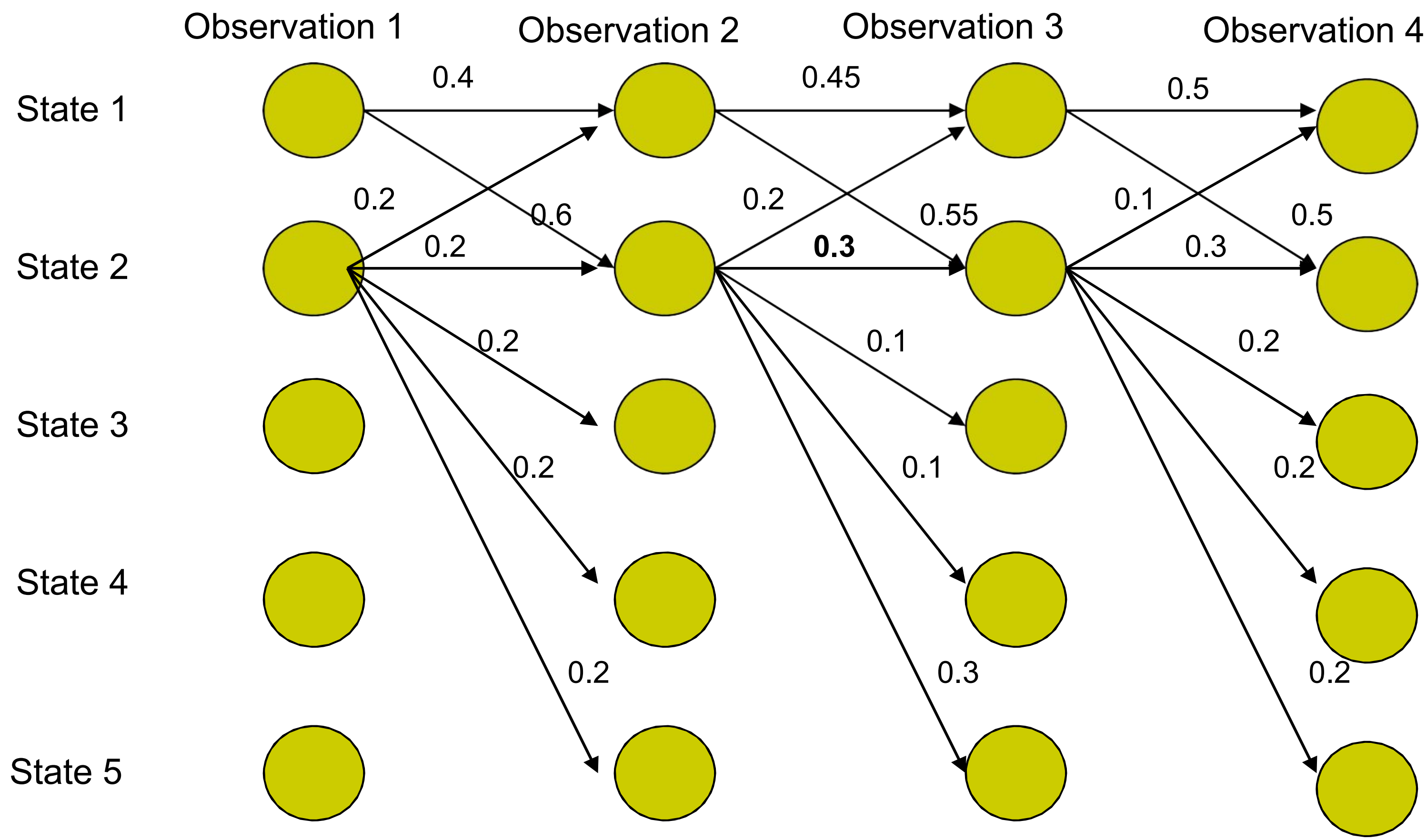
path: 1 → 1→ 1 → 1 : 0.09 and path 1 → 2→ 1 → 2: 0.06 ✓
51%

path: 1 → 1→ 1 → 1 : 0.09 and path 2 → 2→ 2 → 2: 0.06
29%

path: 1 → 1→ 1 → 1 : 0.09 // path 2 → 2→ 2 → 2: 0.018 ✓
67%

path: 1 → 1→ 1 → 1 : 0.018 // path 2 → 2→ 2 → 2: 0.06
4%

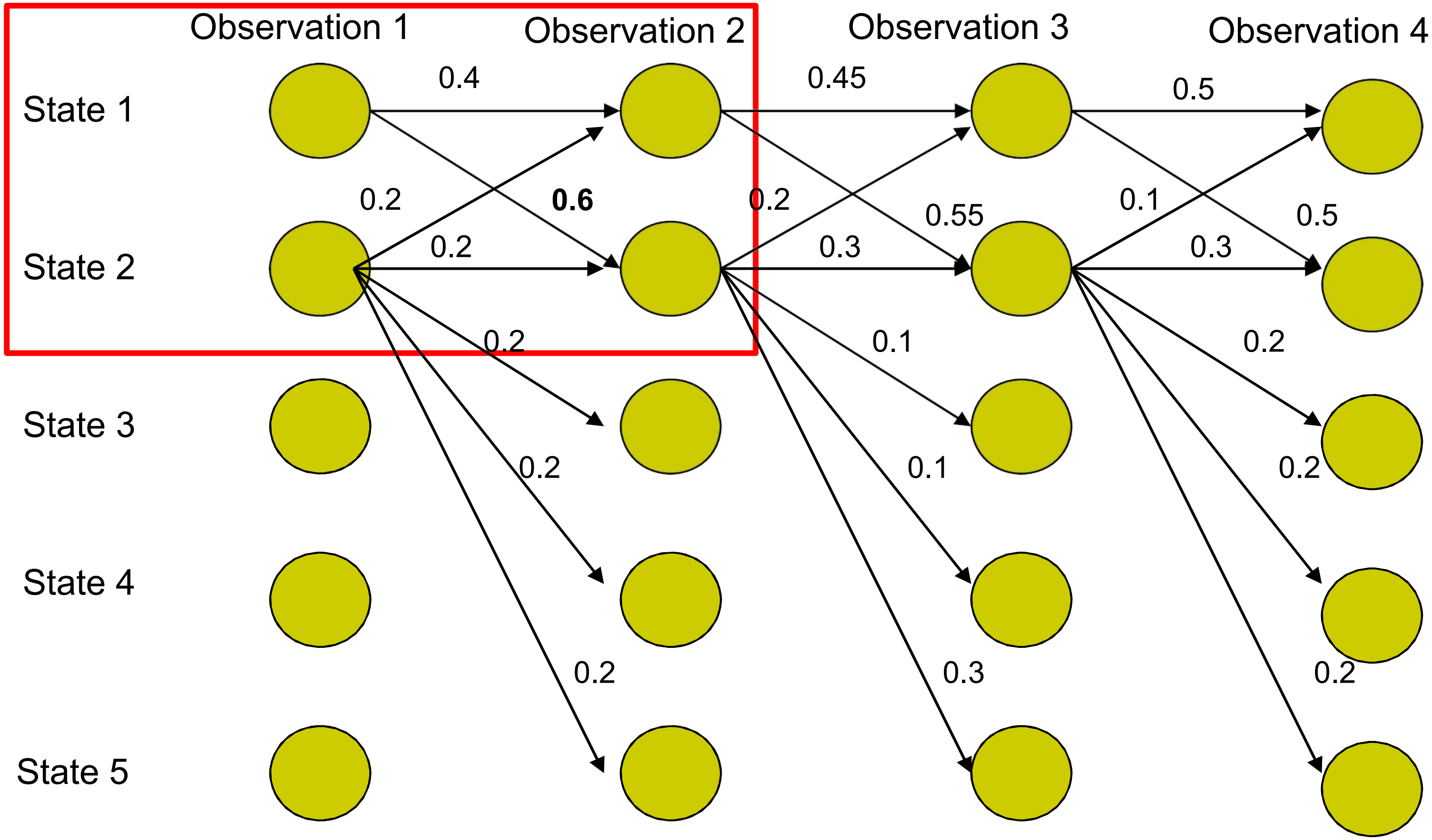
Transition probabilities



Calculate probabilities for the following paths:

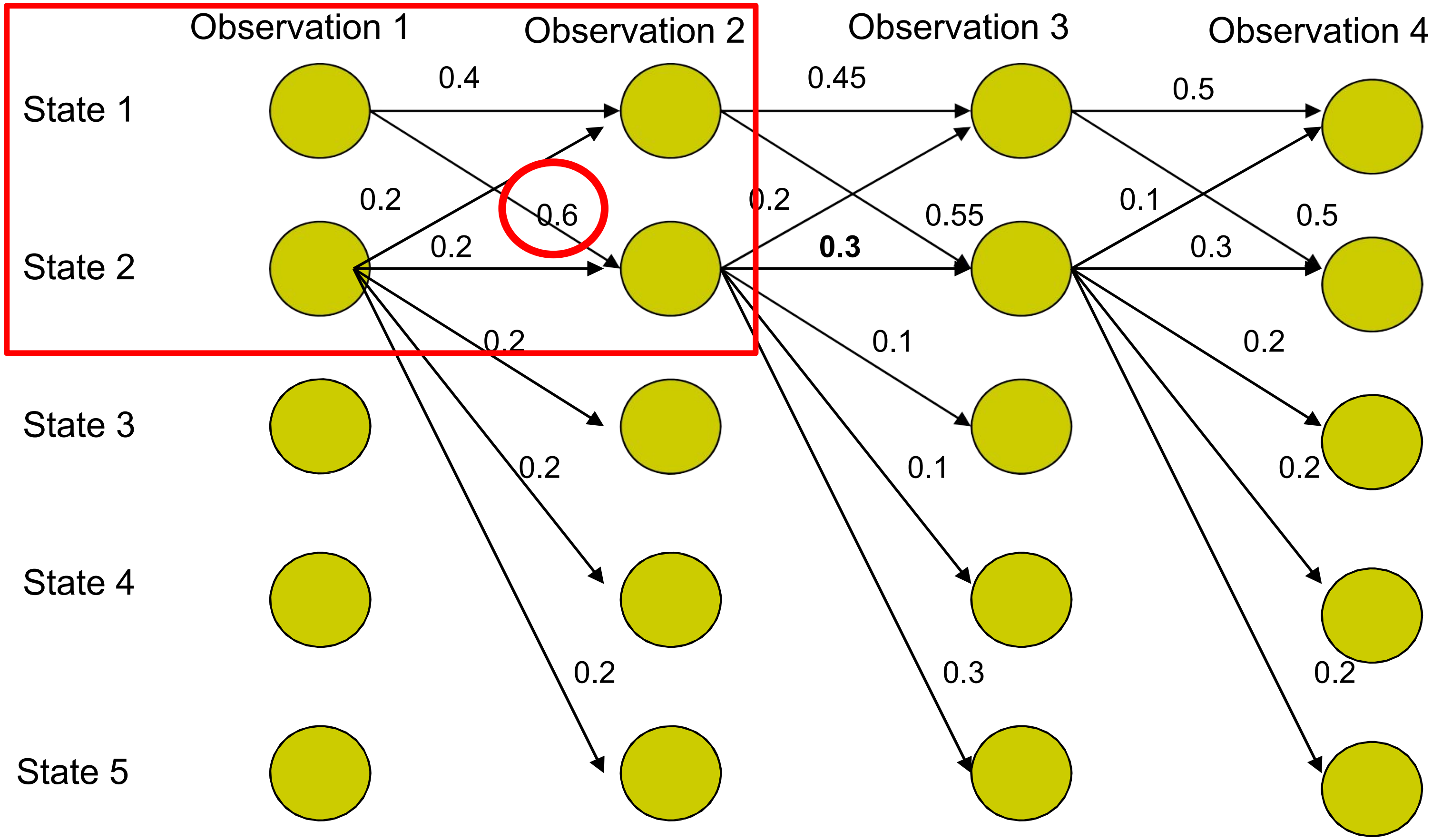
- Probability of path 1-> 1-> 1-> 1: 0.09
- Probability of path 2->2->2->2 : 0.018
- Probability of path 1->2->1->2: 0.06

Recall: Local bias



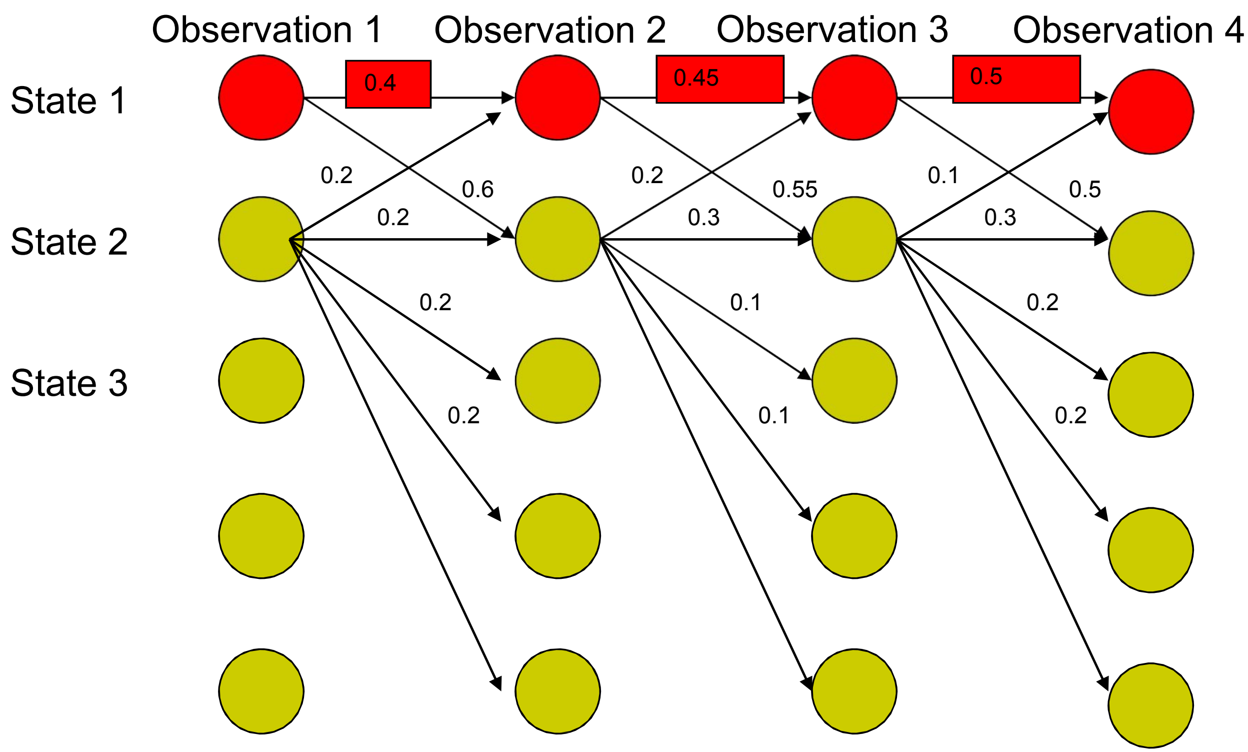
What the local transition probabilities say:

- State 1 almost always prefers to go to state 2

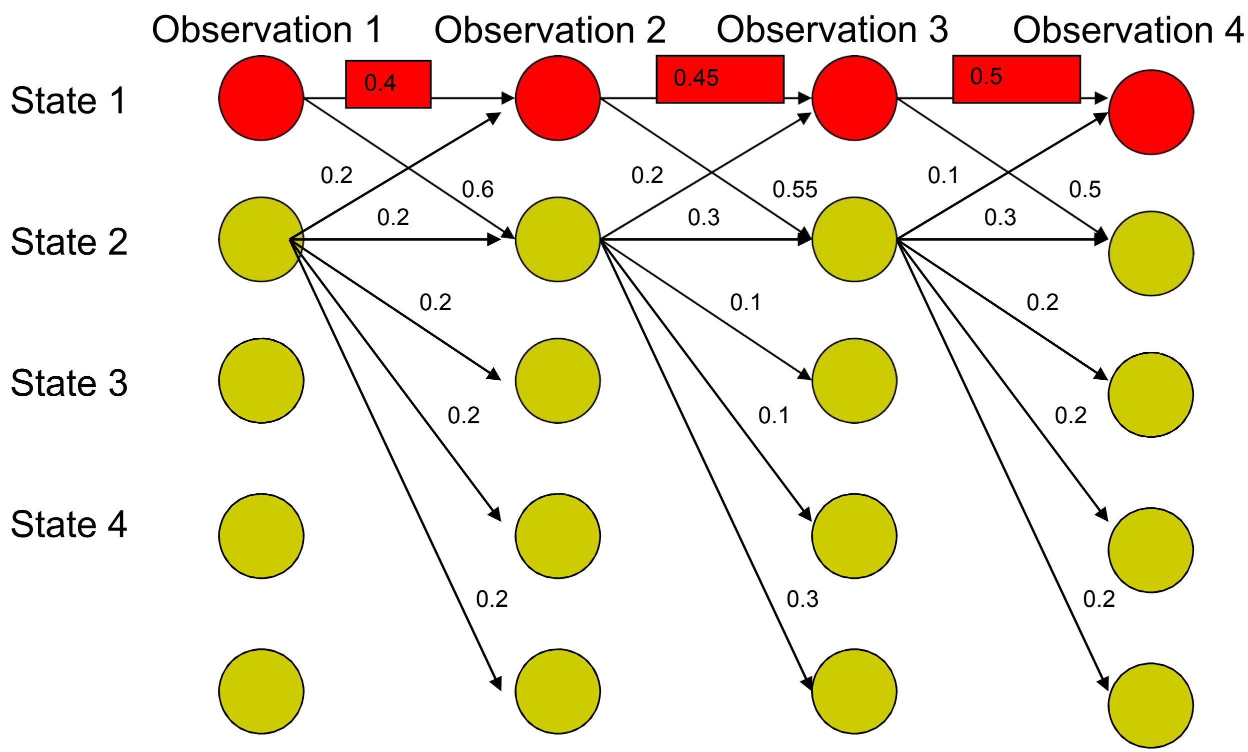


Calculate probabilities for the following paths:

- **Probability of path 1-> 1-> 1-> 1: 0.09**
- Probability of path 2->2->2->2 : 0.018
- Probability of path 1->2->1->2: 0.06

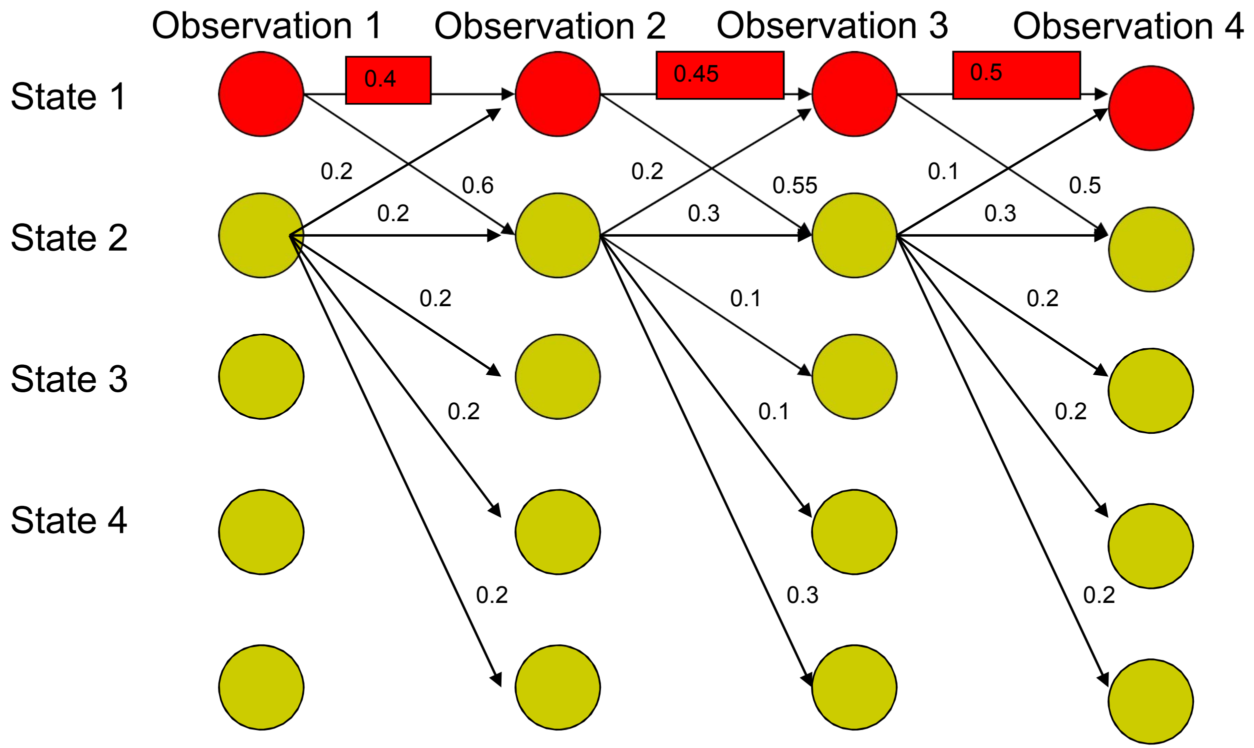


- **Probability of path 1-> 1-> 1-> 1: 0.09**
- Probability of path 2->2->2->2 : 0.018
- Probability of path 1->2->1->2: 0.06



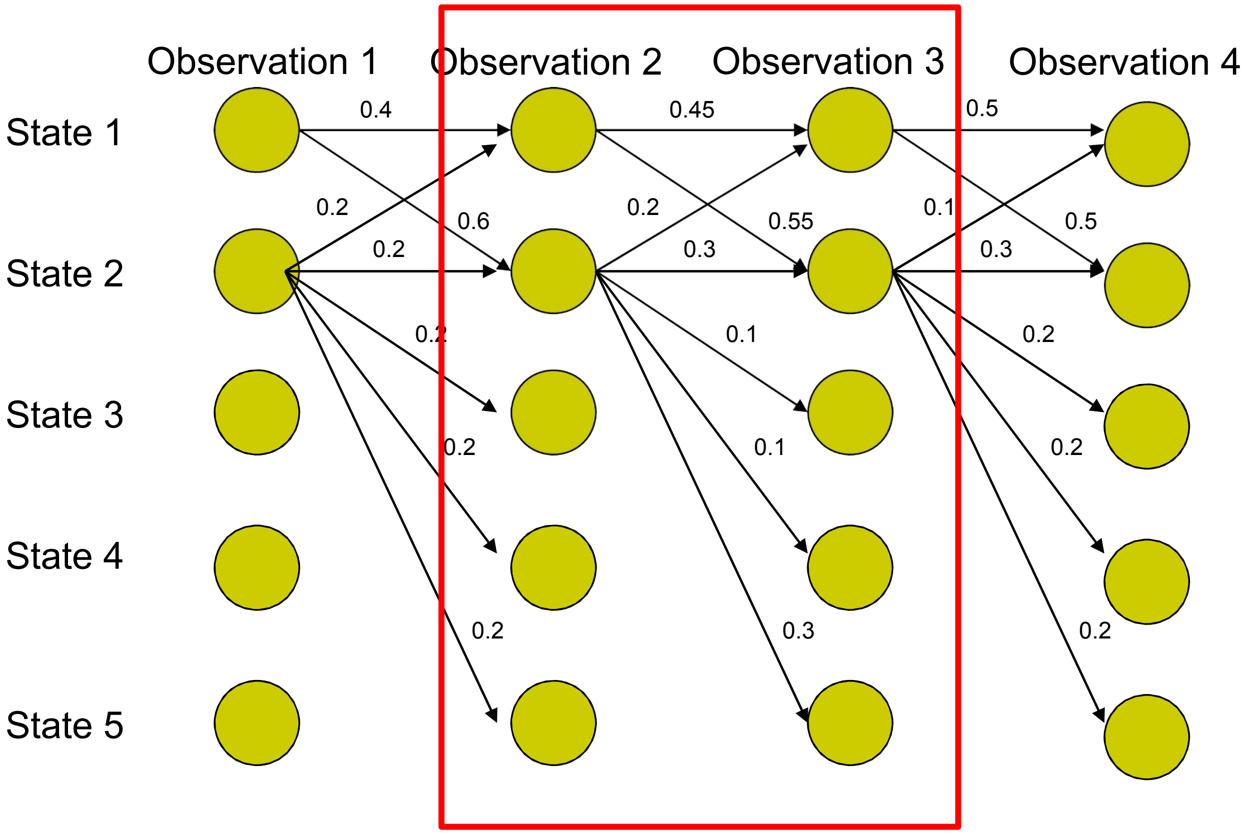
- **Why? State 1 has only two transitions but state 2 has 5:**
  - **Average transition probability from state 2 is lower**

The Label bias problem

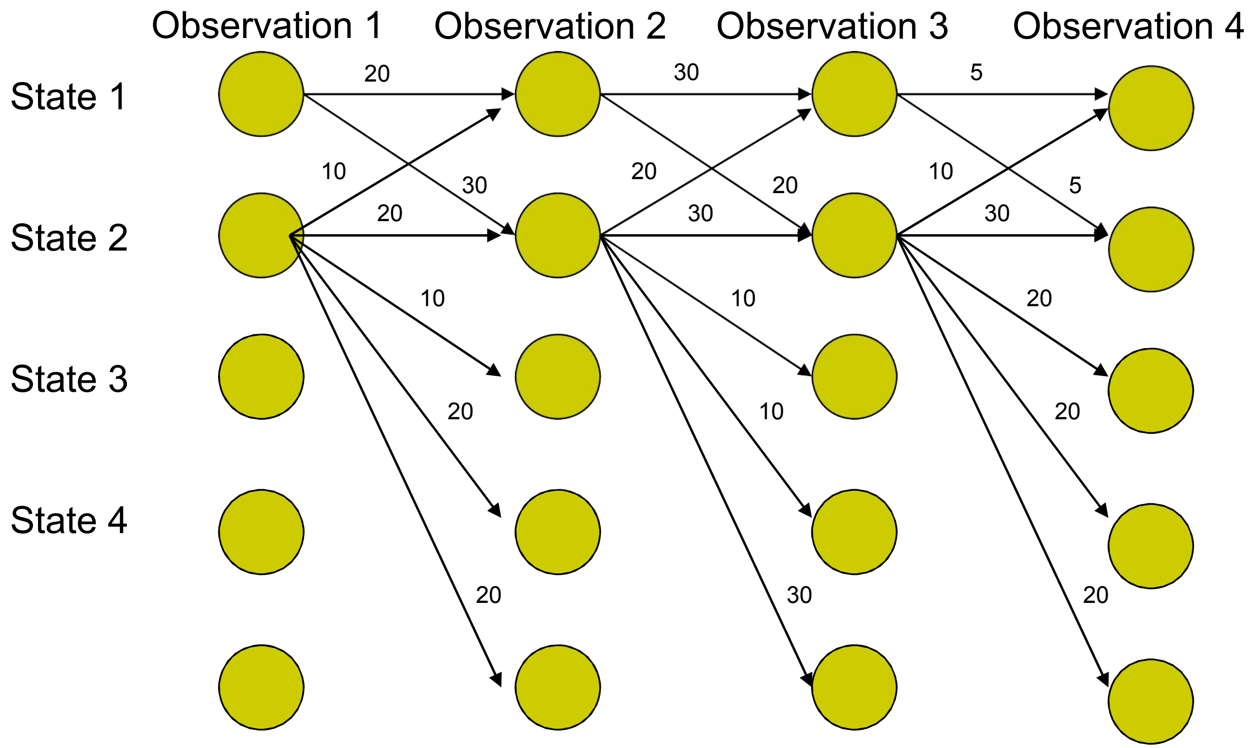


Preference of states with lower number of transitions over others - why?

Idea: Do not normalize probabilities locally



Option-1: Leave them as is



- From local probabilities to local potentials

Option-2: Use conditional random fields

- Global optimization

$$\underset{X_1,\dots,X_m}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{i=N} V^*(X_i)}_{\text{unary potentials}} + \underbrace{\sum_{i=1}^{i=N-1} E^*(X_i, X_{i+1})}_{\text{pairwise potentials}}$$

- Optimization packages have solvers, e.g., sequential QP

Generative models + Hidden Markov Models

Generative media scenarios

Input

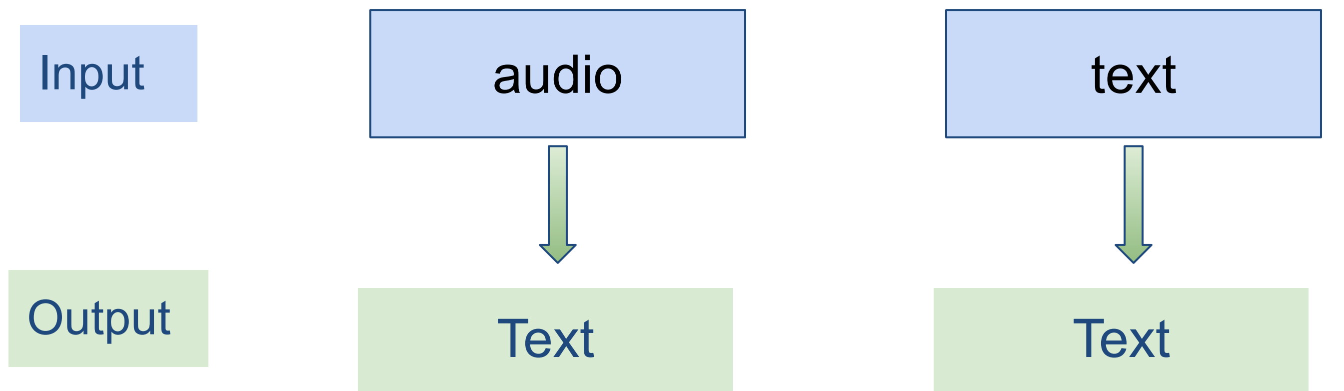
Text
Image
Video
Audio

Many to many

Output

Text
Image
Video
Audio

Generative models + Hidden Markov Models



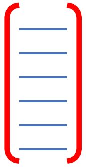
Both (audio and text) are sequences

Recall: Speech Recognition

- **Representing observations:** FFT of of the speech signal.

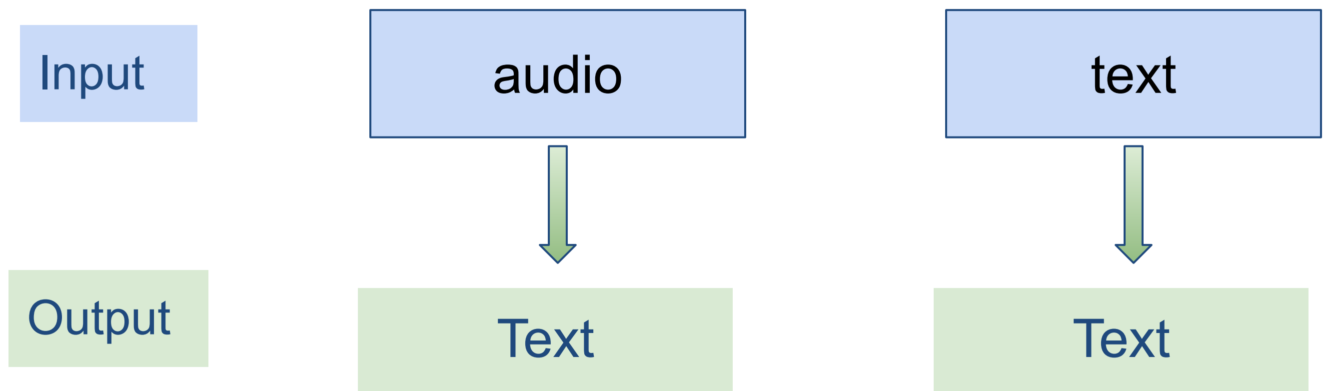
Acoustic wave form
Sampled at 16KHz, quantized to 8-12 bits

Fast Fourier Transform (FFT) **of one frame (10ms)** is the HMM observation, once per 10ms



Observation = compressed version of the log magnitude FFT, from one 10ms frame

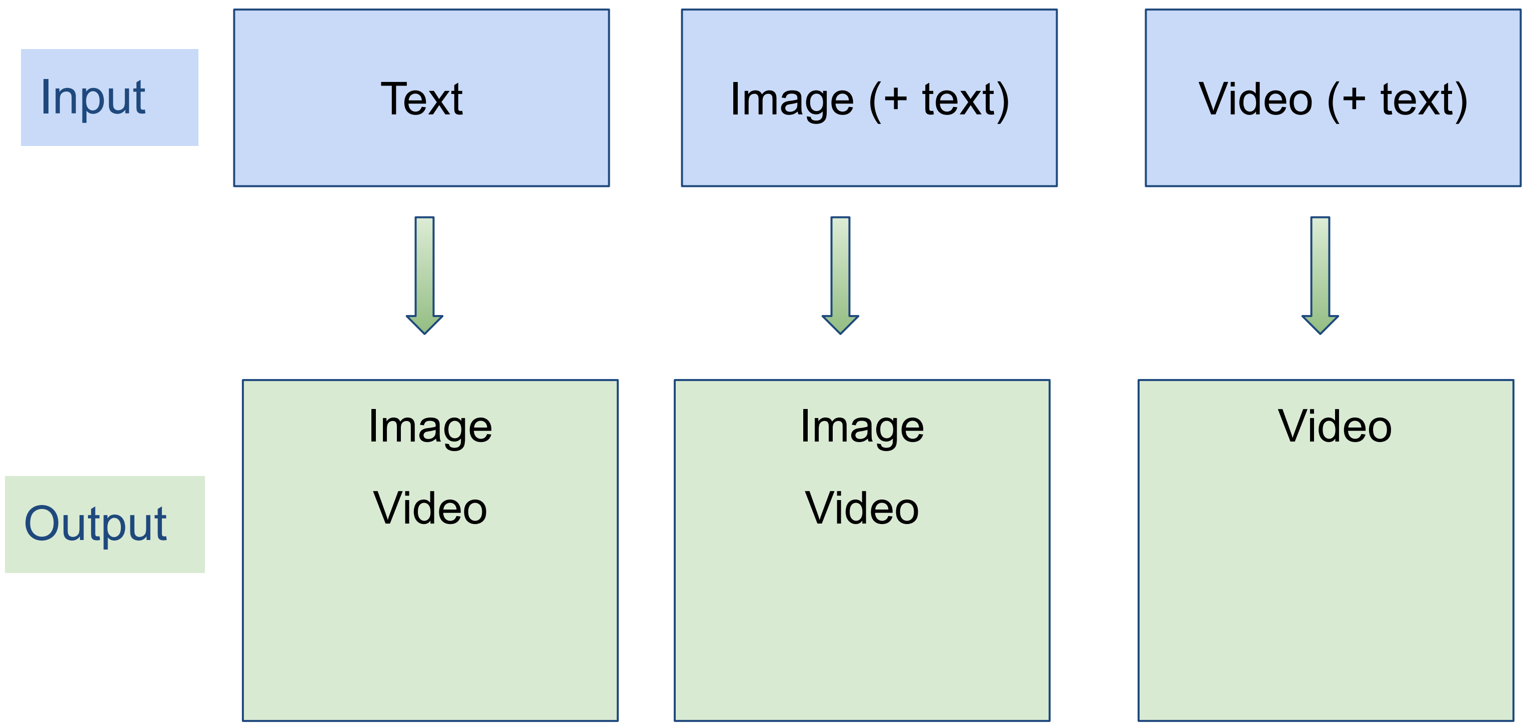
Generative models + Hidden Markov Models



Both (audio and text) are sequences

- Tokenize (audio: discrete FFT, text: bag of words)
- Model using markov chain

Generative multimodal media scenarios



Generative media scenarios

Input: Text

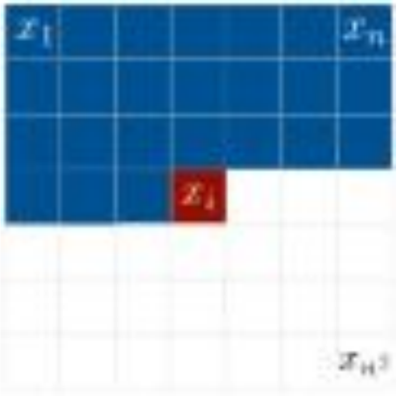
Output: **Image**, Video, Audio

An oil painting of two rabbits in the style of American Gothic, wearing the same clothes as in the original

Autoregressive modeling

- Predicting the next pixel based on the previous pixels.
- State 0: a blank canvas
- State 1: first pixel (0,0) painted.
- …

- State MN: last pixel (M-1, N-1) painted.



What are some of the downsides of representing image states as the state of a painted canvas?

What are some of the downsides of representing image states as the state of a painted canvas?

The underlying model would be generalizable to different image resolutions

32%

The underlying model would not be generalizable to different image resolutions ✓

59%

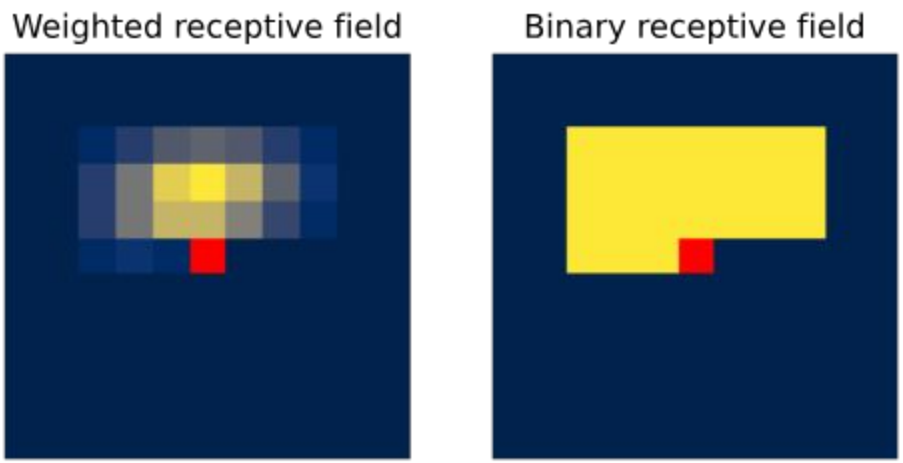
The underlying model would be take into account all the previous states, making the image generation slower ✓

81%

The underlying model would be take into account all the previous states. This is crucial for a smooth image generation

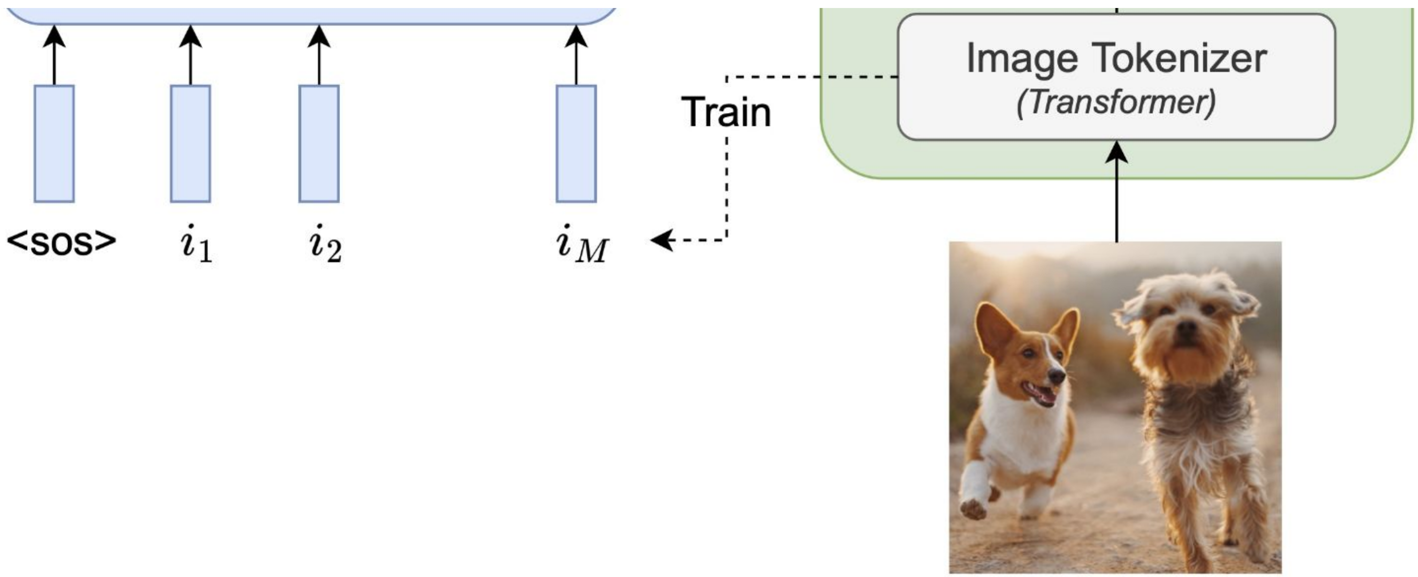
12%

Mitigation-1: receptive fields



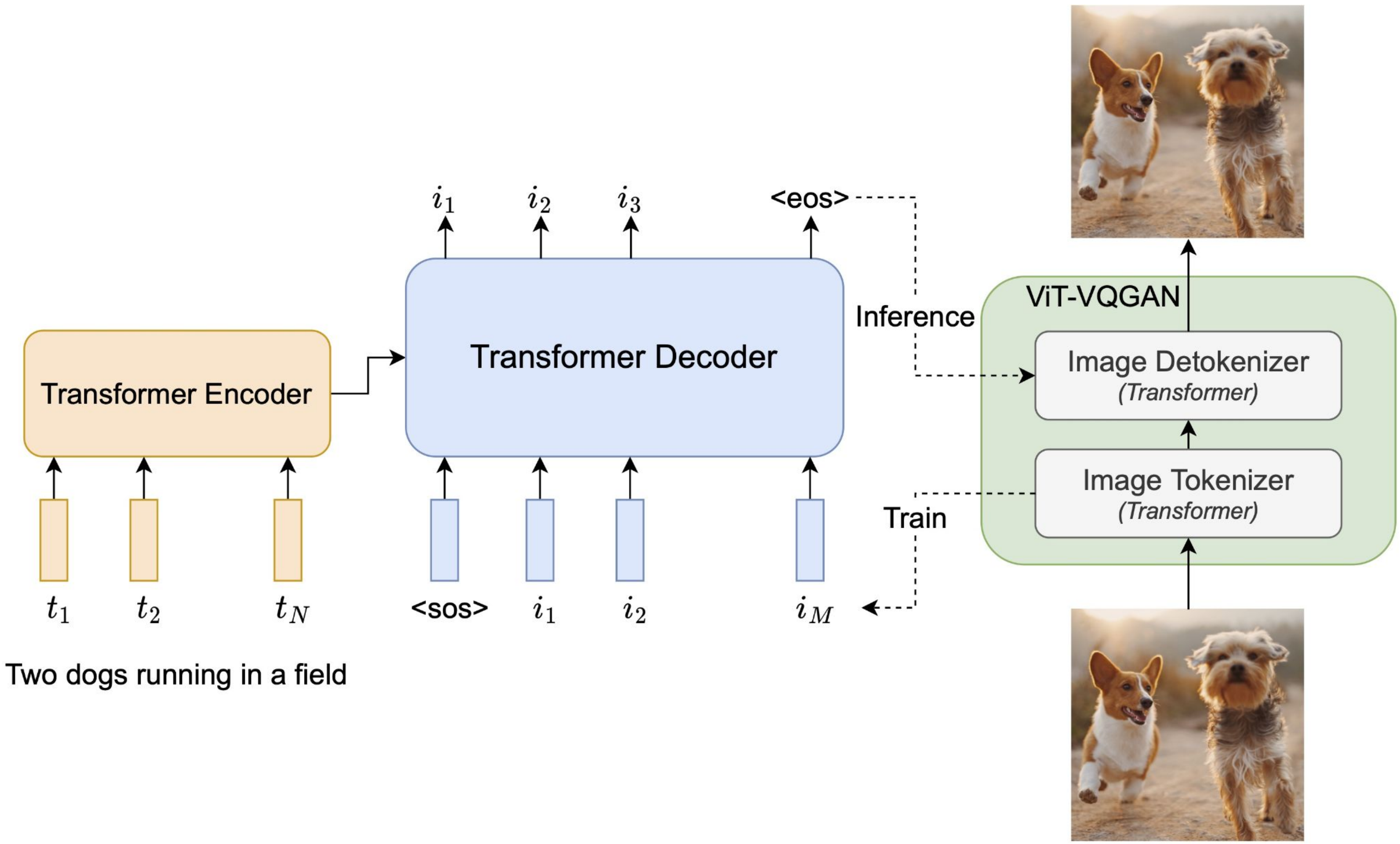
Every pixel does not (and need not) contribute to every state.

Mitigation-2: Tokenize

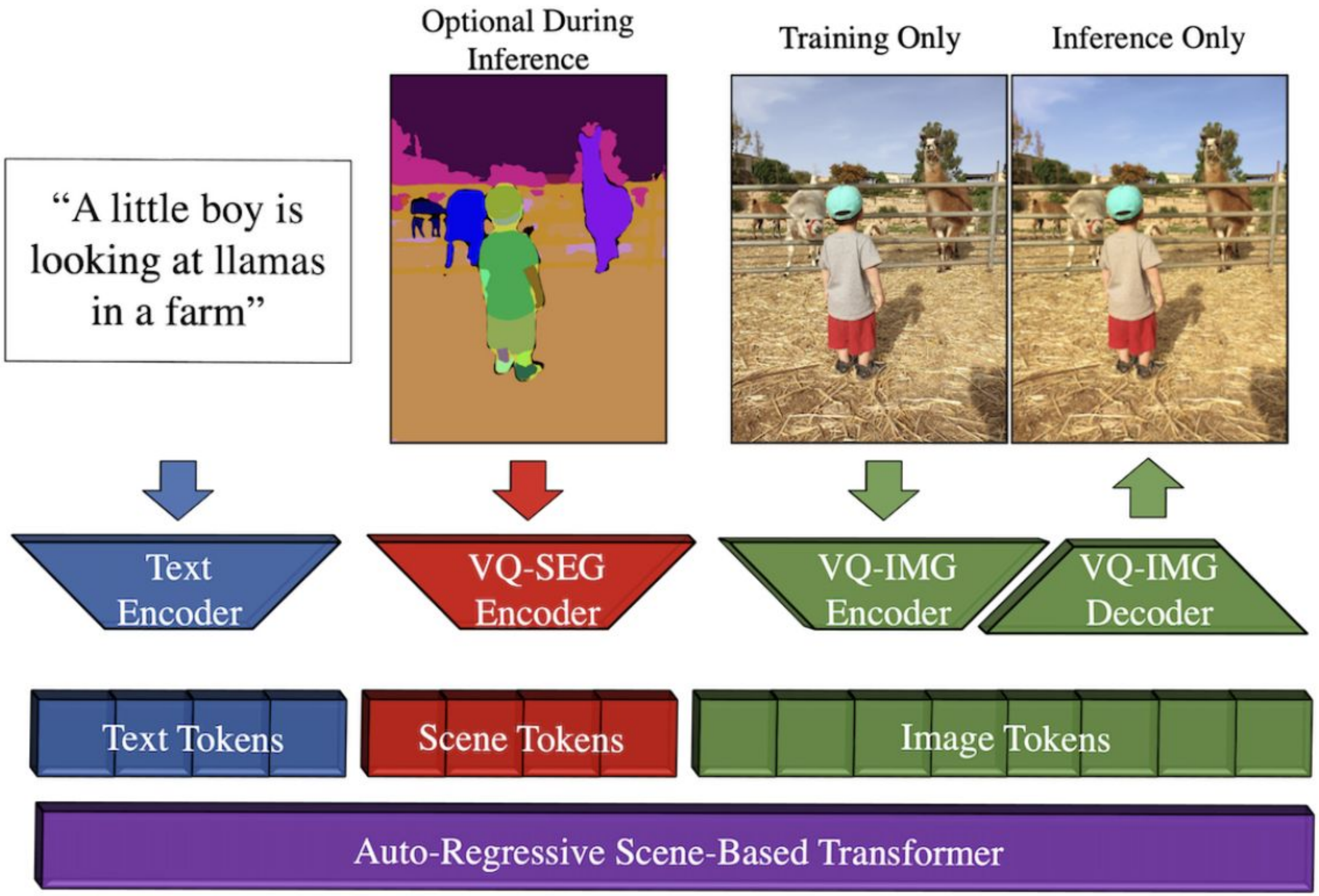


- Operate at a token level - instead of pixel-canvas level

- Tokenize
- Model using a markov chain (auto-regressive)

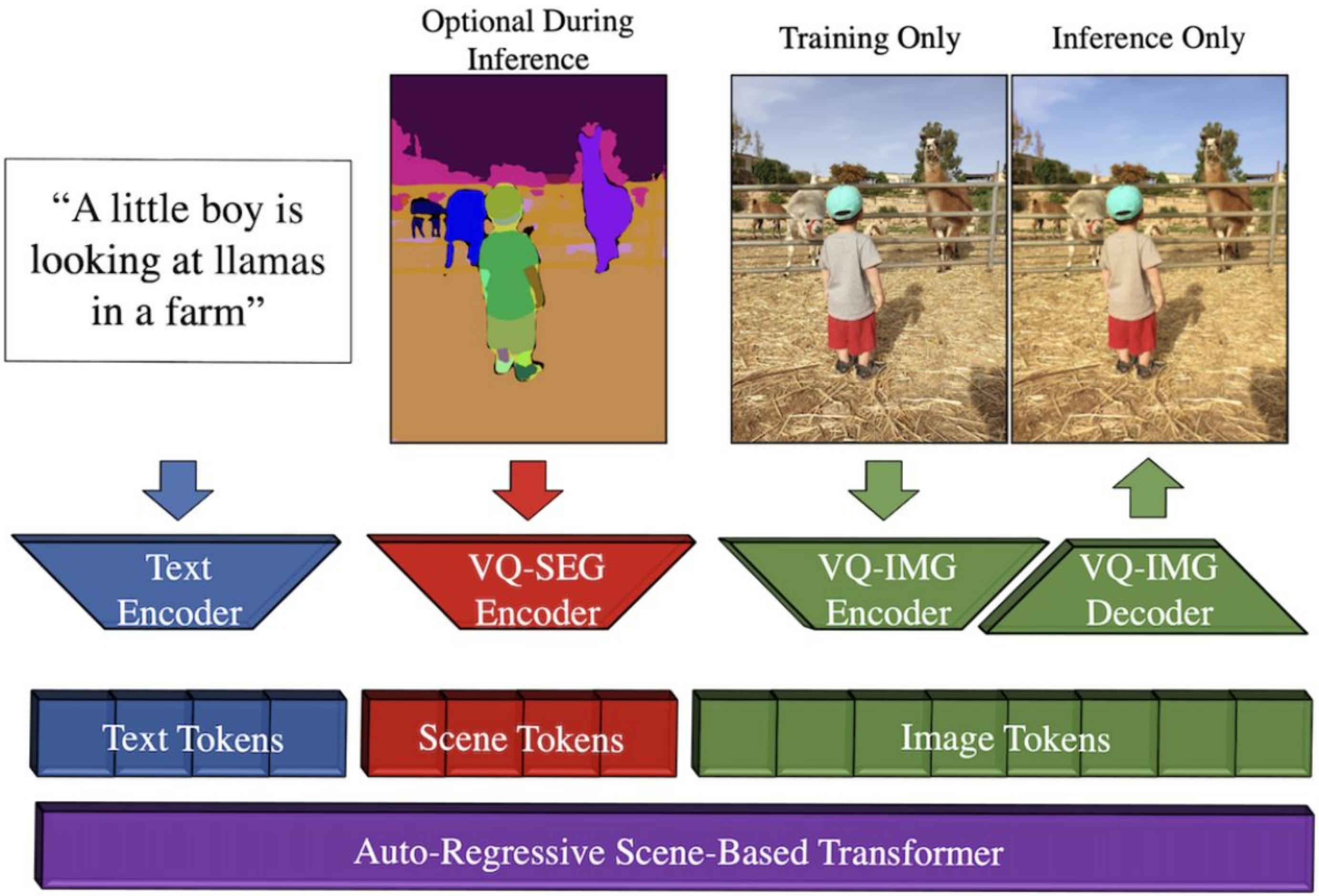


Make-a-scene! (Meta)



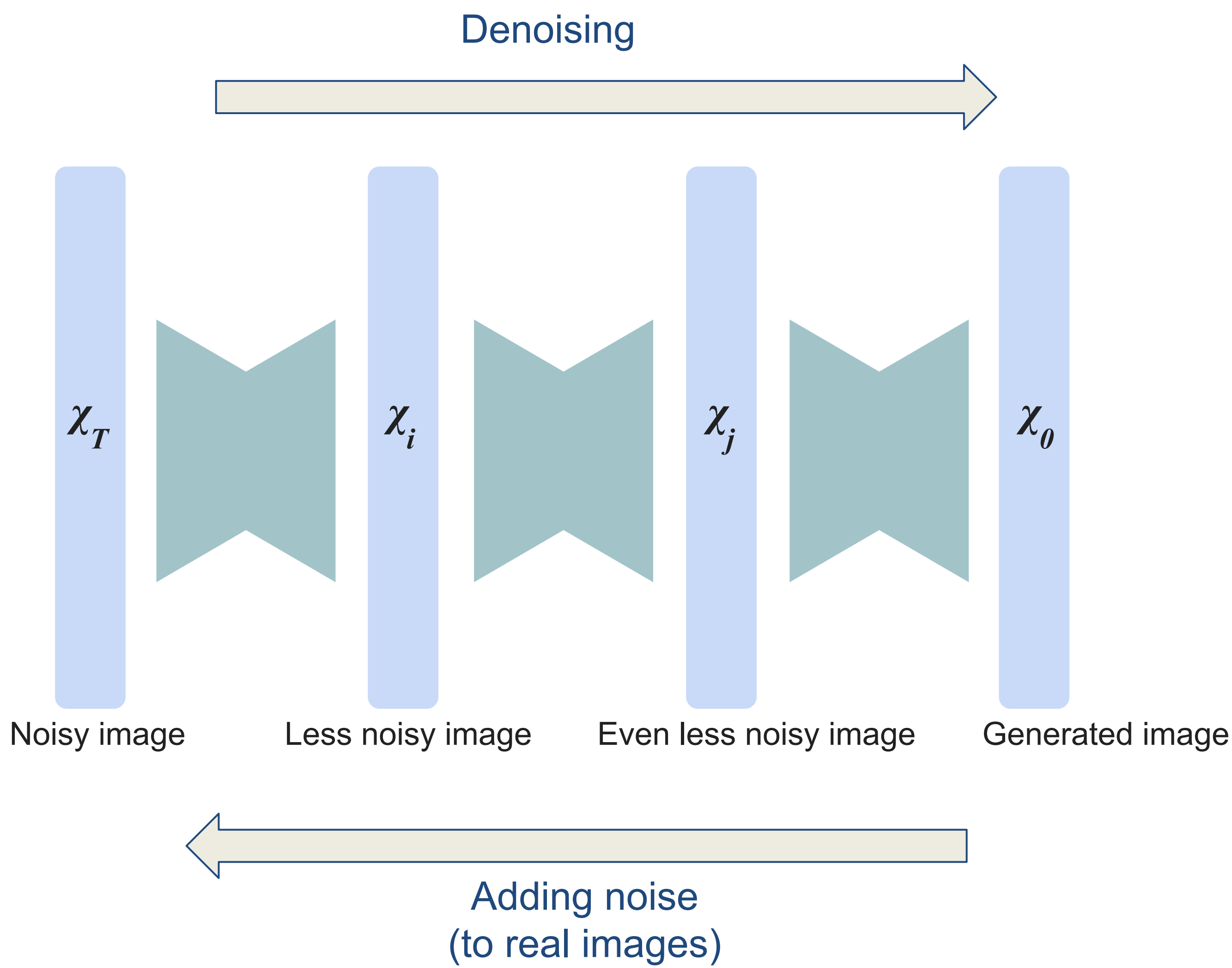
Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors, , ECCV'22

- Tokenize
- Model using a markov chain (auto-regressive)



Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors, , ECCV'22

Landscape of generative models: Diffusion models



Generative media scenarios

Input | Text

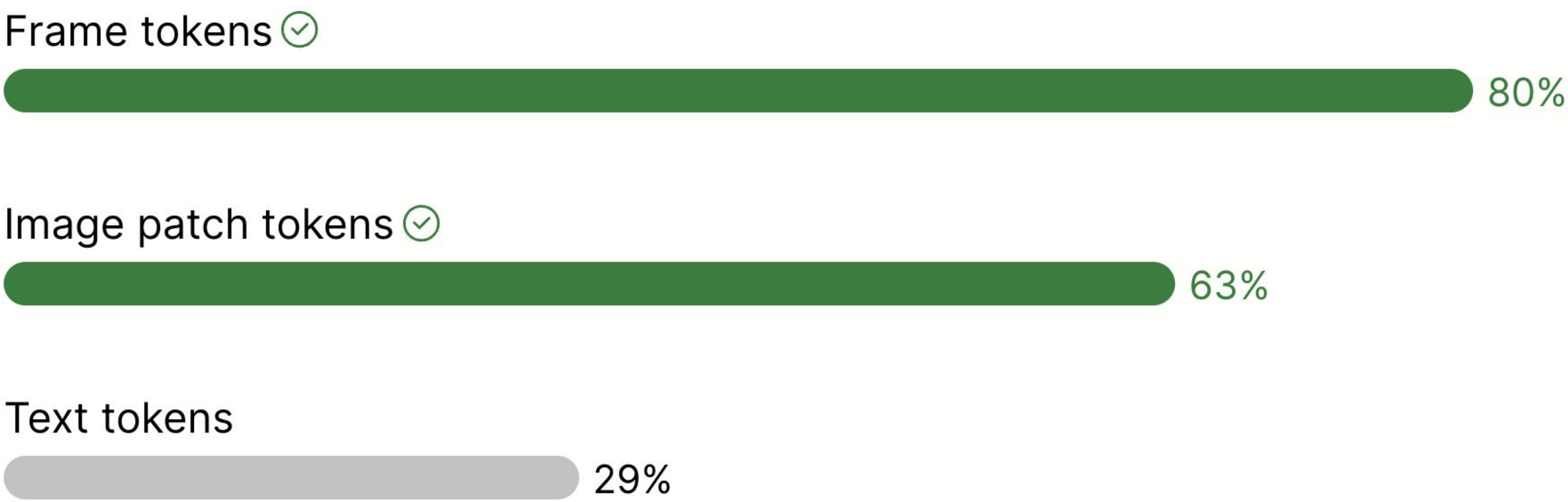
↓

Output | Image
Video
Audio

Close up shot of a living flame wisp darting through a bustling market in the night.

What is the temporal sequence composed of in a video autoregressive model?

What is the temporal sequence composed of in a video autoregressive model?



Next Class

Neural Networks I: Artificial neuron, MLP, activation functions, learning with gradient descent

Reading: Forsyth Ch 16.1, 16.3-16.4