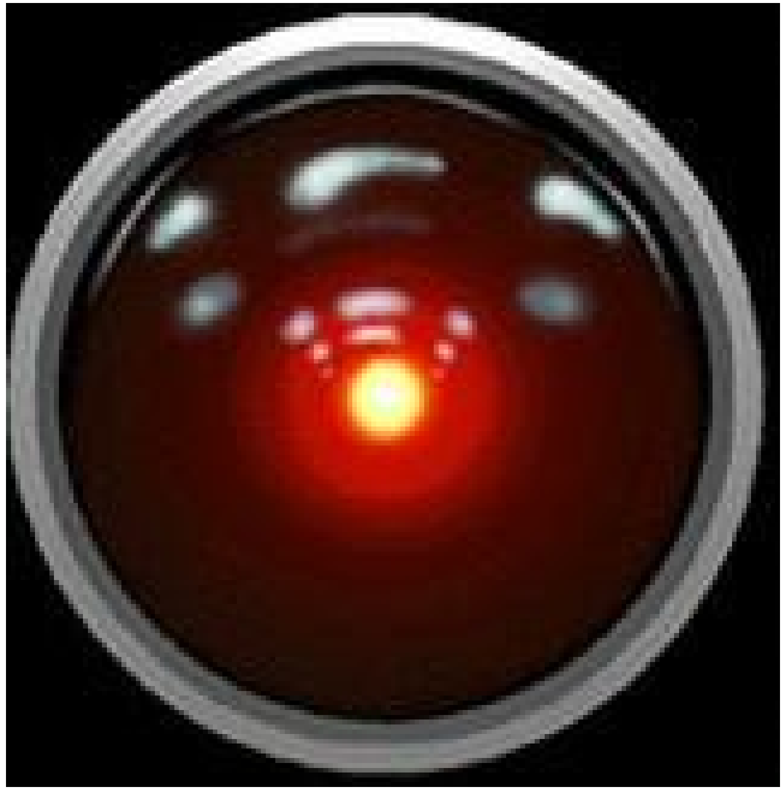
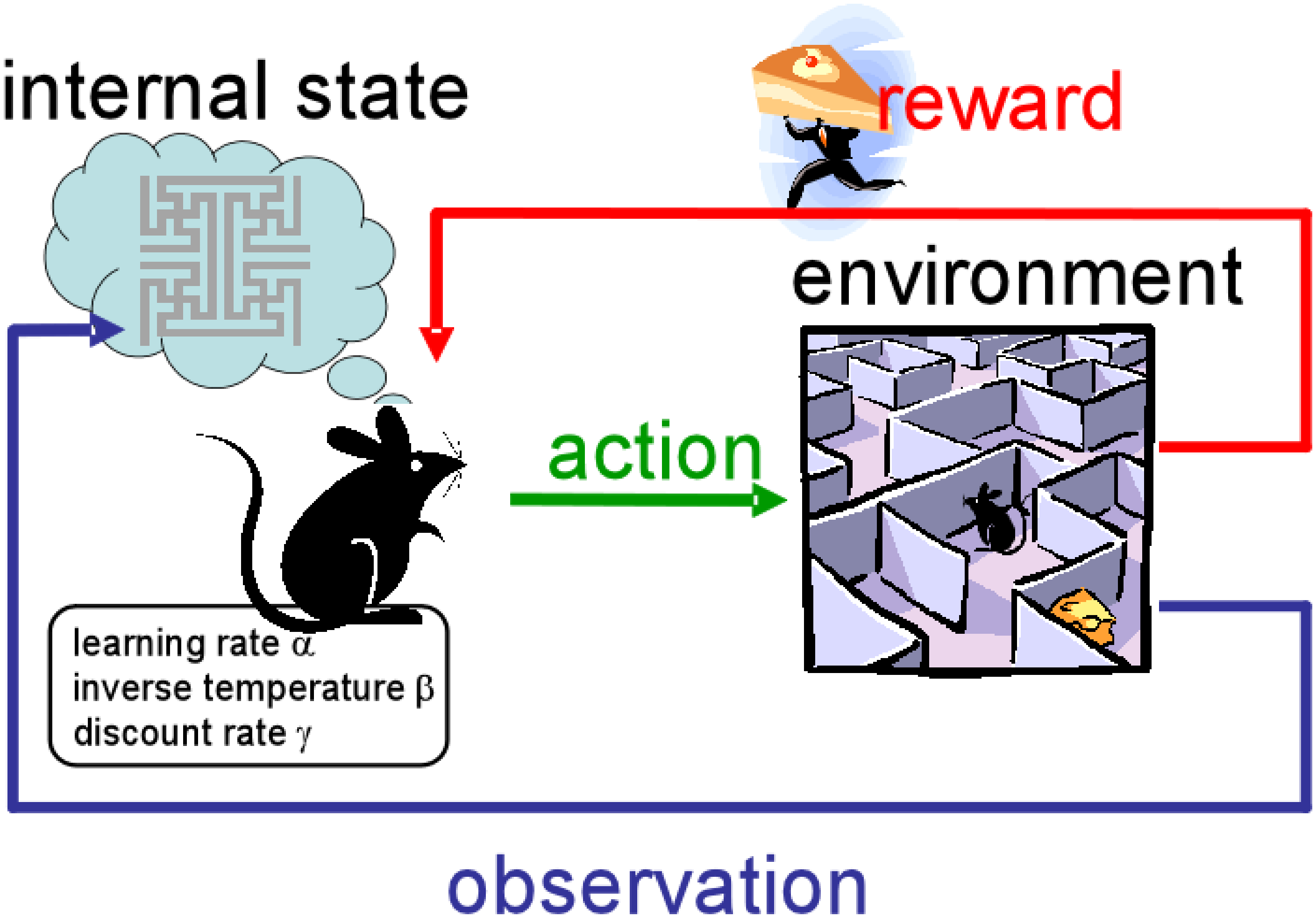


CS 309: Autonomous Intelligent Robotics

Instructor: Jivko Sinapov

http://www.cs.utexas.edu/~jsinapov/teaching/cs309_spring2017/

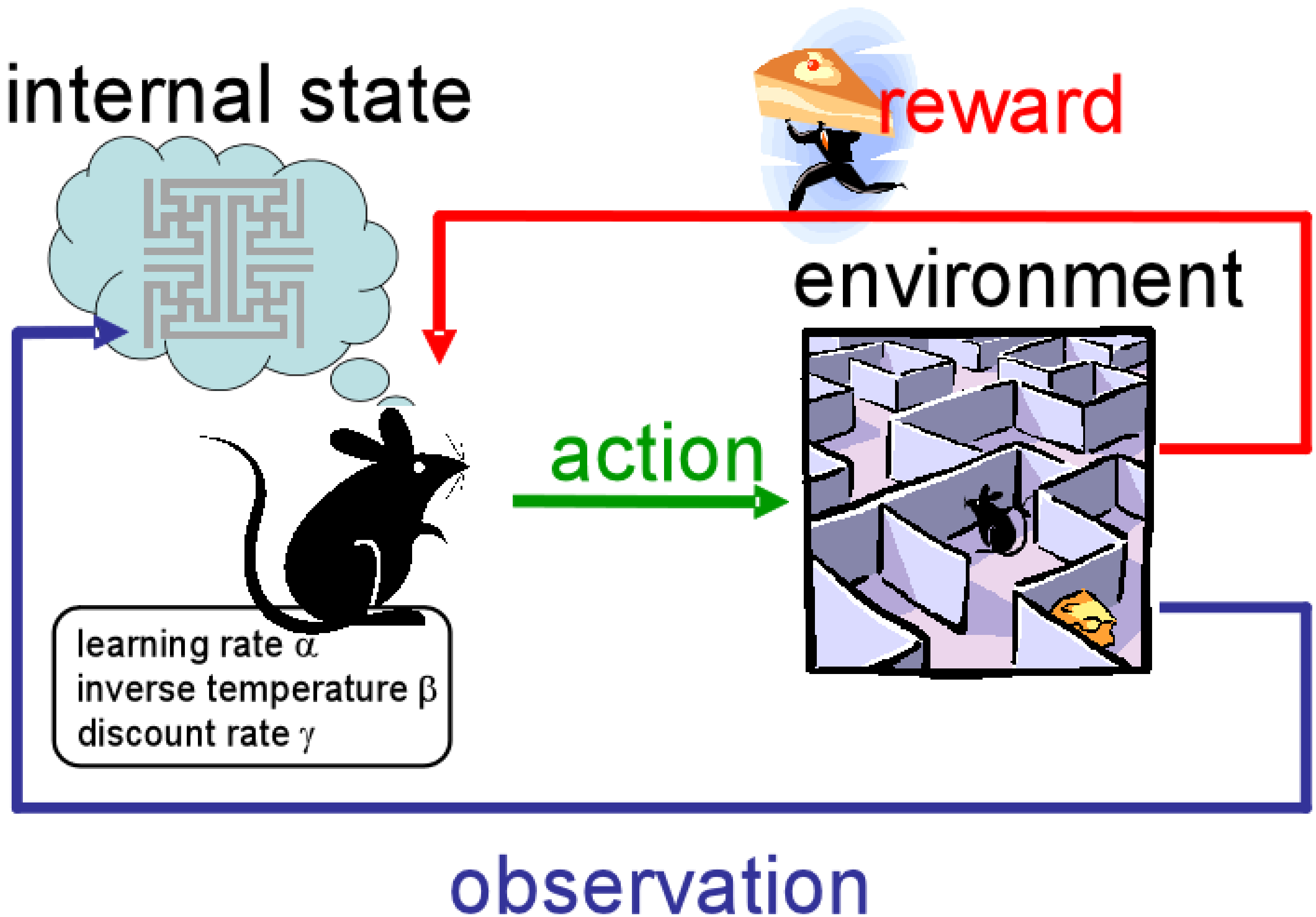
Reinforcement Learning



A little bit about next semester...

- New robots: robot arm, HSR-1 robot
- Virtually all of the grade will be based on a project
- There will still be some lectures and tutorials but much of the class time will be used to give updates on your projects and for discussions

Reinforcment Learning



Activity: You are the Learner

At each time step, you receive an observation (a color)

You have three actions: “clap”, “wave”, and “stand”

After performing an action, you may receive a reward

Next time...

How can we formalize the strategy for solving this RL problem into an algorithm?

Project Breakout Session

Meet with your group

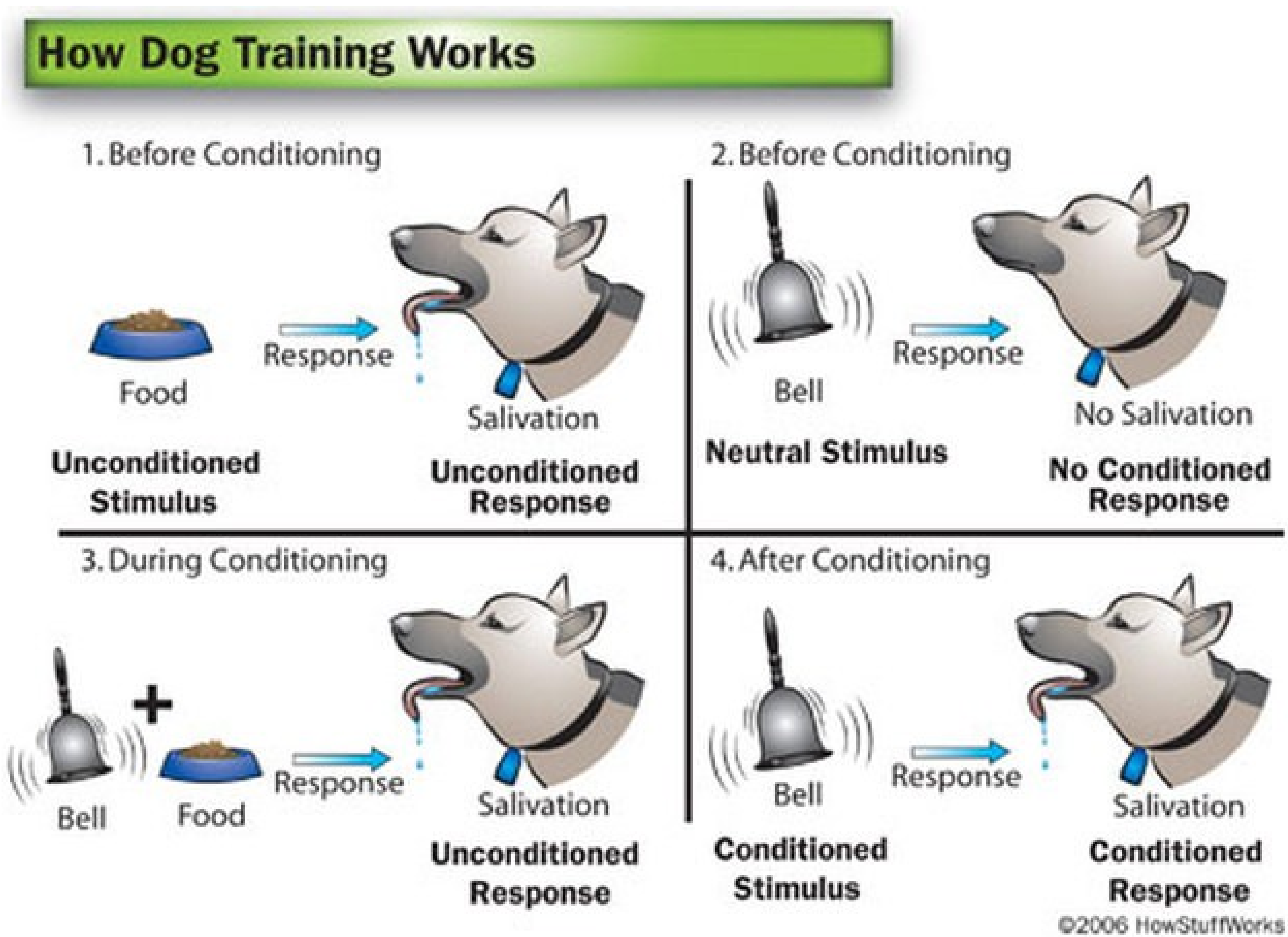
Summarize what you've done so far, identify next steps

Come up with questions for me, the TAs, and the metors

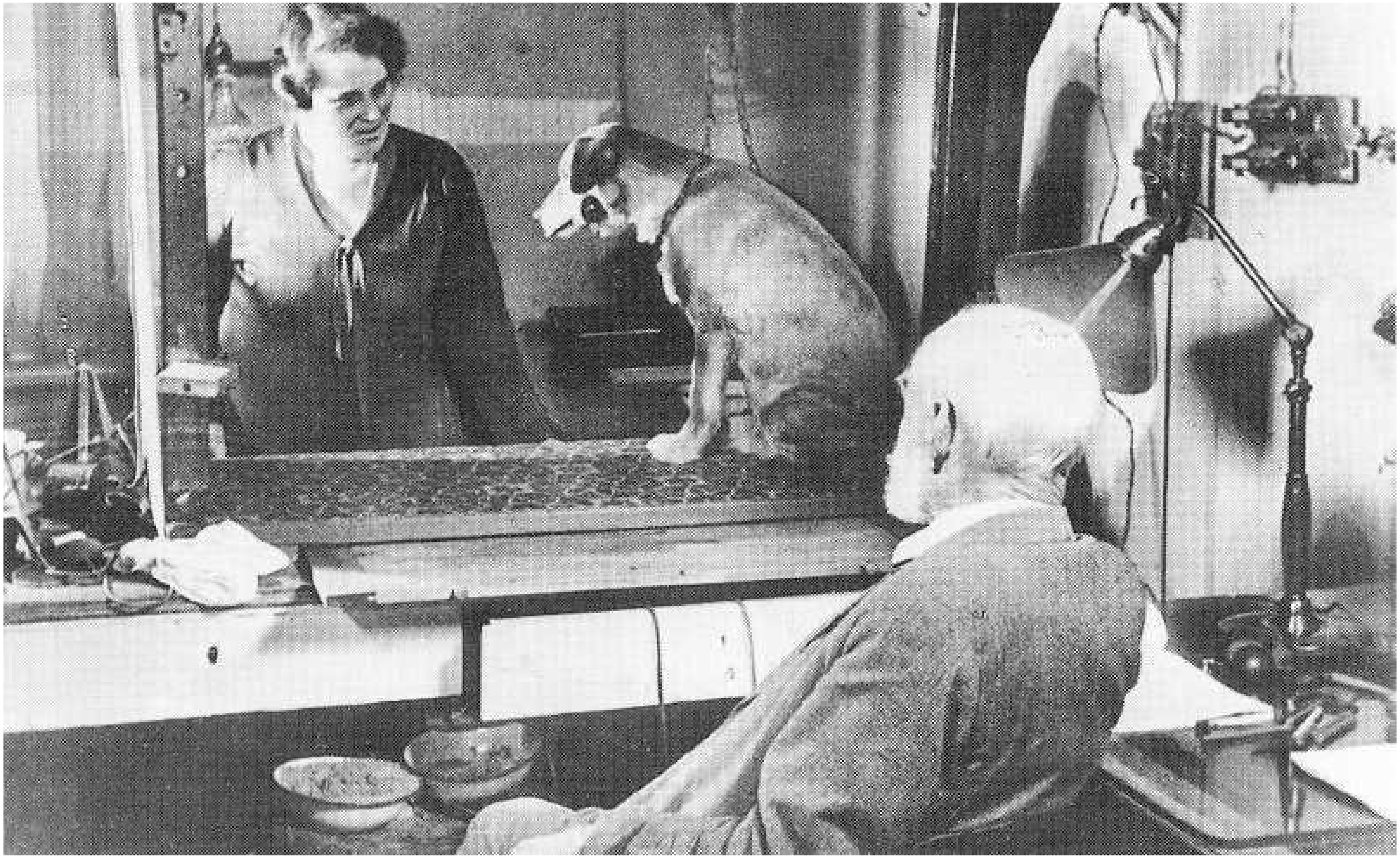
Main Reference

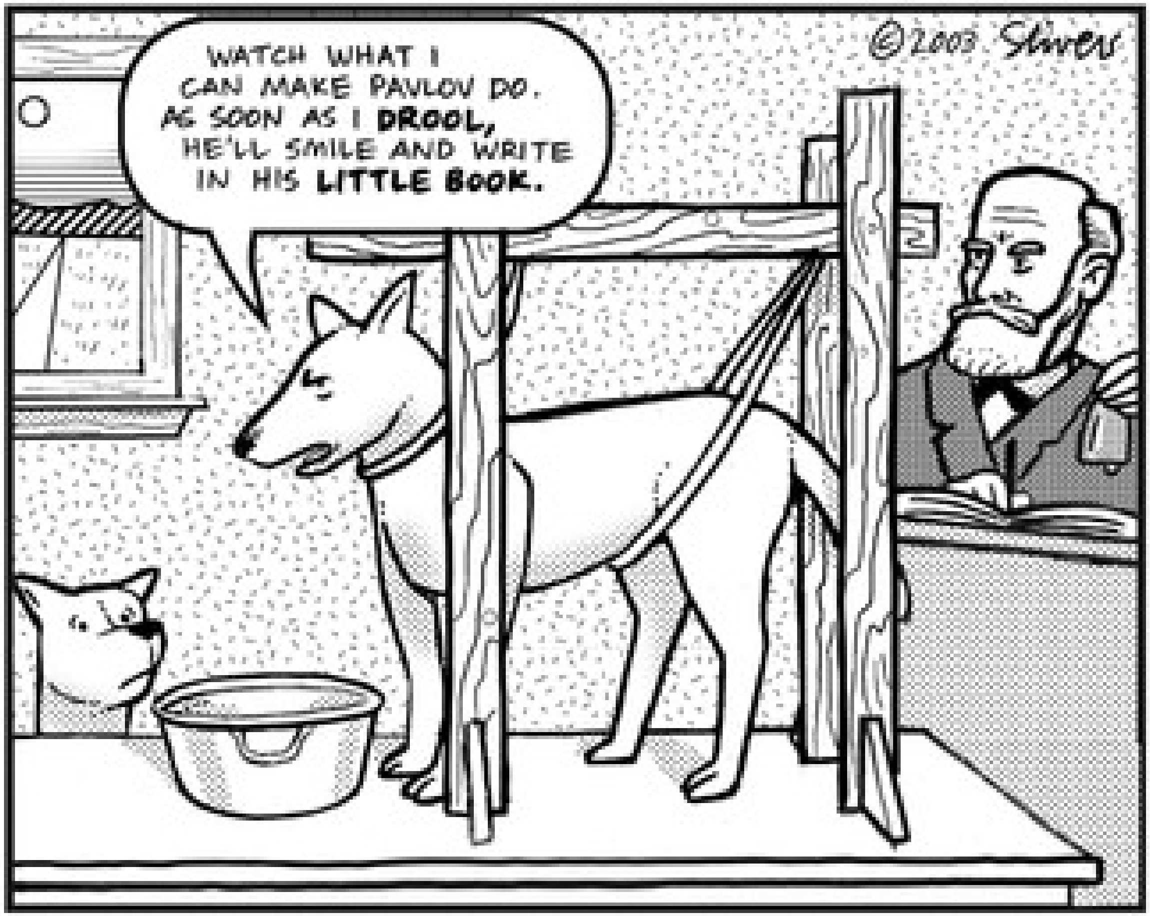
Sutton and Barto, (2012). Reinforcement Learning: An Introduction, Chapter 1-3

What is Reinforcement Learning (RL)?


How Dog Training Works
1. Before Conditioning
Food
Response
Salivation
Unconditioned Stimulus
Unconditioned Response
2. Before Conditioning
Bell
Response
No Salivation
Neutral Stimulus
No Conditioned Response
3. During Conditioning
Bell
Food
Response
Salivation
Unconditioned Response
4. After Conditioning
Bell
Response
Salivation
Conditioned Stimulus
Conditioned Response
©2006 HowStuffWorks

Ivan Pavlov (1849-1936)



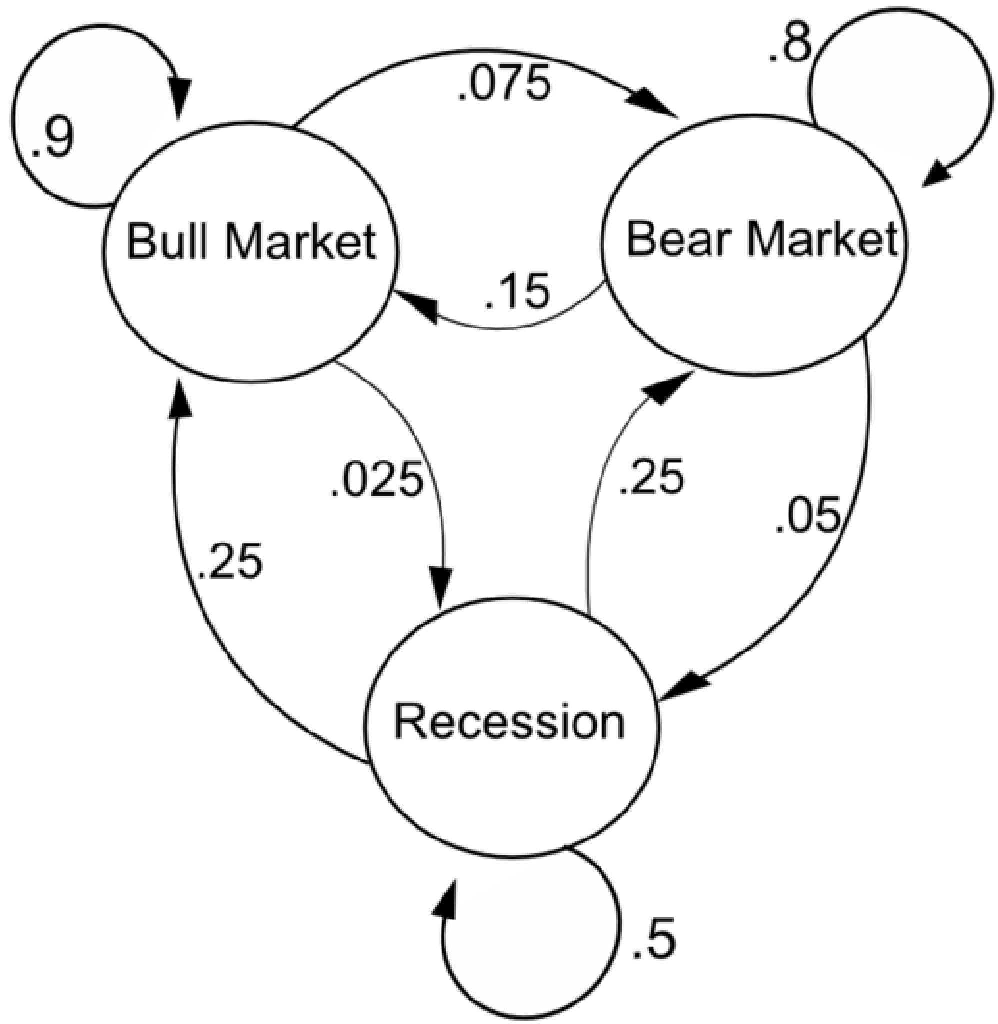

WATCH WHAT I CAN MAKE PAVLOV DO. AS SOON AS I DROOL, HE'LL SMILE AND WRITE IN HIS LITTLE BOOK.
©2003 Shiers

From Pavlov to Markov

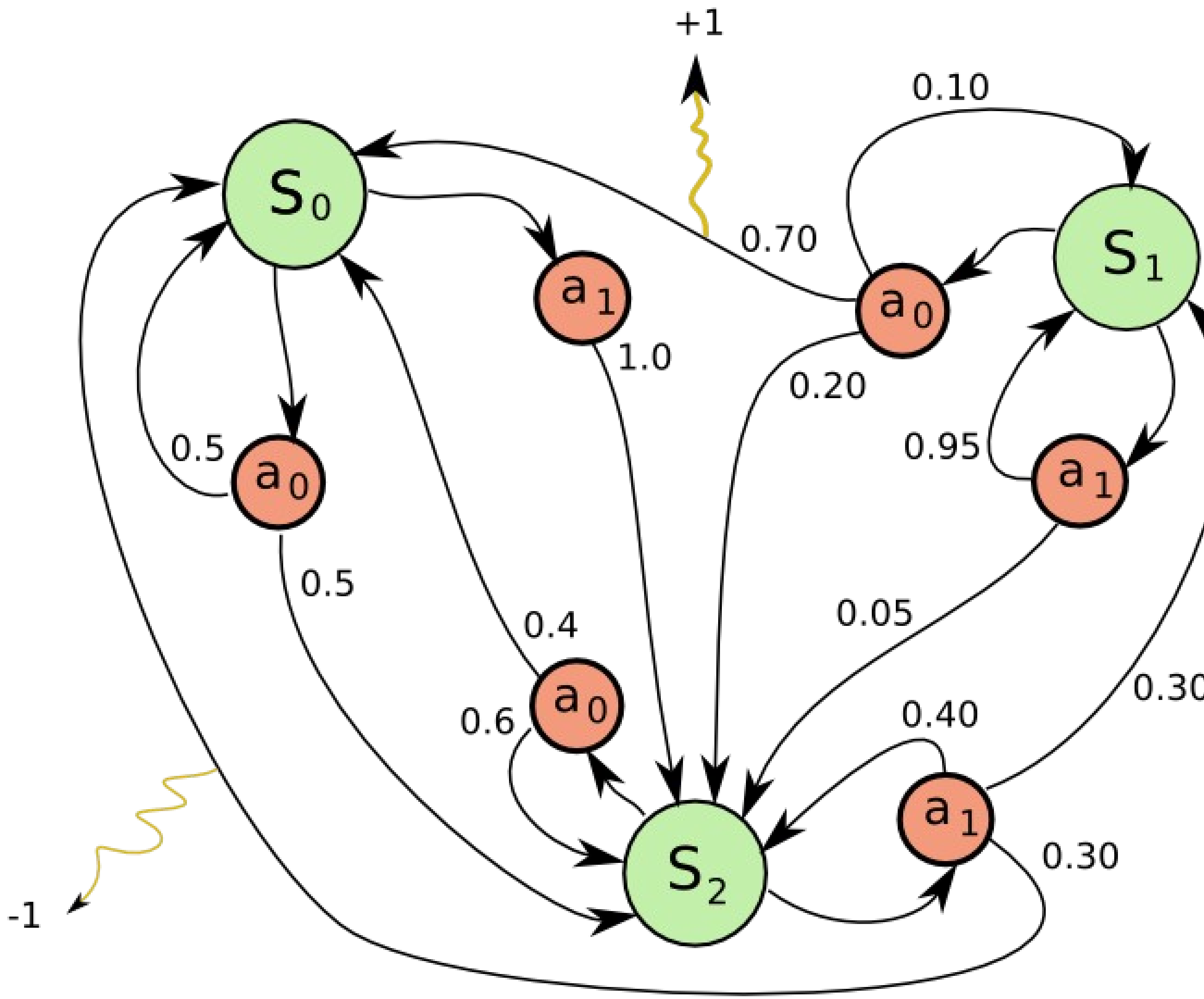
Andrey Andreyevich Markov (1856 – 1922)

[http://en.wikipedia.org/wiki/Andrey_Markov]

Markov Chain



Markov Decision Process



The Multi-Armed Bandit Problem

a.k.a. how to pick between Slot Machines (one-armed bandits) so that you walk out with the most $$$ from the Casino

Arm 1

Arm 2

. . . .

Arm k

How should we decide which slot machine to pull next?

How should we decide which slot machine to pull next?

0 1 1 0 1

0 0 0 50 0

How should we decide which slot machine to pull next?

1 with prob = 0.6 and 0 otherwise

50 with prob = 0.01 and 0 otherwise

Value Function

A value function encodes the “value” of performing a particular action (i.e., bandit)

Rewards observed when performing action *a*

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}.$$

Value function Q

# of times the agent has picked action *a*

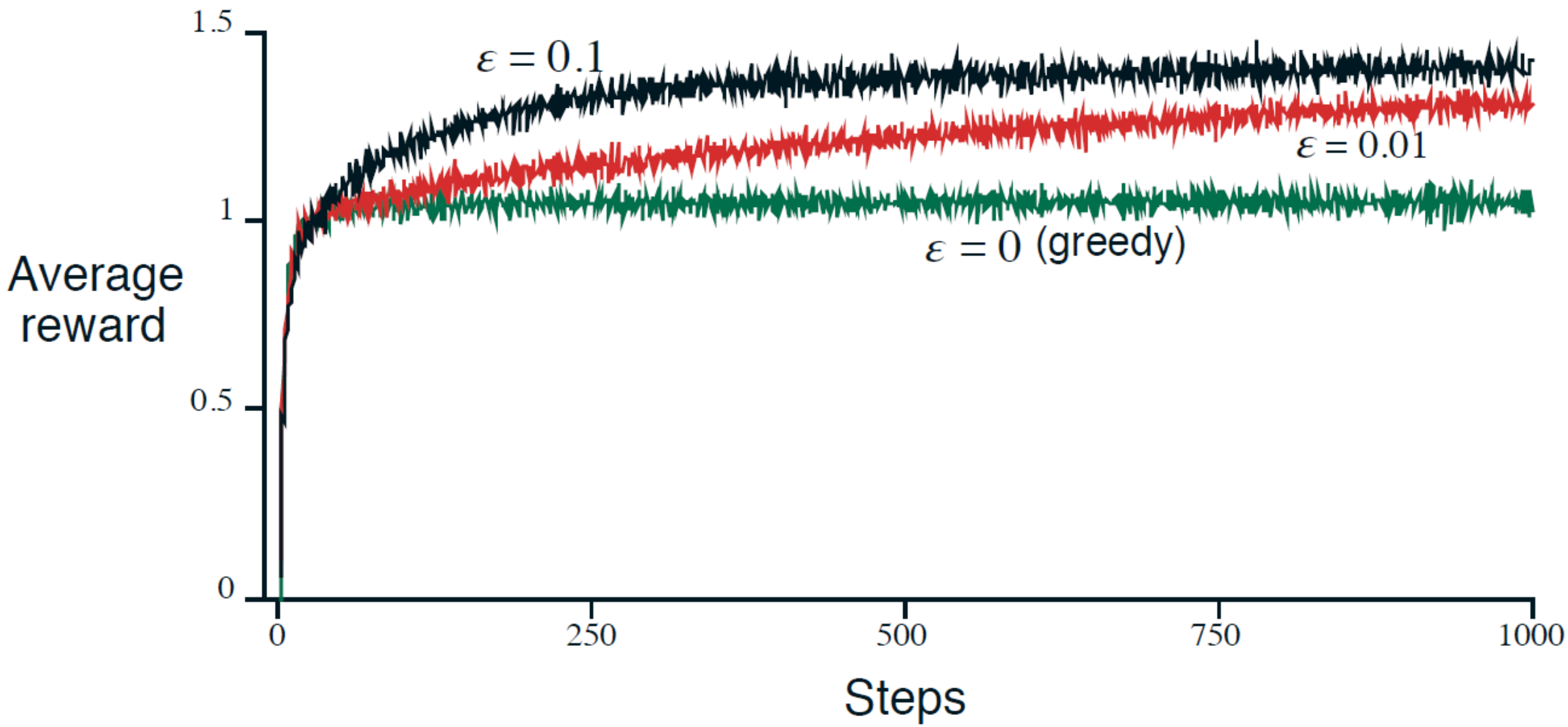
How do we choose next action?

- Greedy: pick the action that maximizes the value function, i.e.,

$$Q_t(A_t^*) = \max_a Q_t(a)$$

- ε-Greedy: with probability ε pick a random action, otherwise, be greedy

10-armed Bandit Example



Soft-Max Action Selection

Exponent of natural logarithm (~ 2.718)

$$\frac{e^{Q_t(a)/\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/\tau}}$$

"temperature"

As temperature goes up, all actions become nearly equally likely to be selected; as it goes down, those with higher value function outputs become more likely

What happens after choosing an action?

Batch: $$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

Incremental:
$$\begin{aligned}
Q_{k+1} &= \frac{1}{k}\sum_{i=1}^{k} R_i \\
&= \frac{1}{k}\left(R_k + \sum_{i=1}^{k-1} R_i\right) \\
&= \frac{1}{k}\Big(R_k + (k-1)Q_k + Q_k - Q_k\Big) \\
&= \frac{1}{k}\Big(R_k + kQ_k - Q_k\Big) \\
&= Q_k + \frac{1}{k}\Big[R_k - Q_k\Big],
\end{aligned}$$

Updating the Value Function

$$NewEstimate \leftarrow OldEstimate + StepSize\left[Target - OldEstimate\right]$$

What happens when the payout of a bandit is changing over time?

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

What happens when the payout of a bandit is changing over time?

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

Earlier rewards may not be indicative of how the bandit performs now

What happens when the payout of a bandit is changing over time?

$$Q_{k+1} = Q_k + \alpha \Big[R_k - Q_k \Big]$$

instead of

$$Q_k + \frac{1}{k} \Big[R_k - Q_k \Big]$$

How do we construct a value function at the start (before any actions have been taken)

How do we construct a value function at the start (before any actions have been taken)

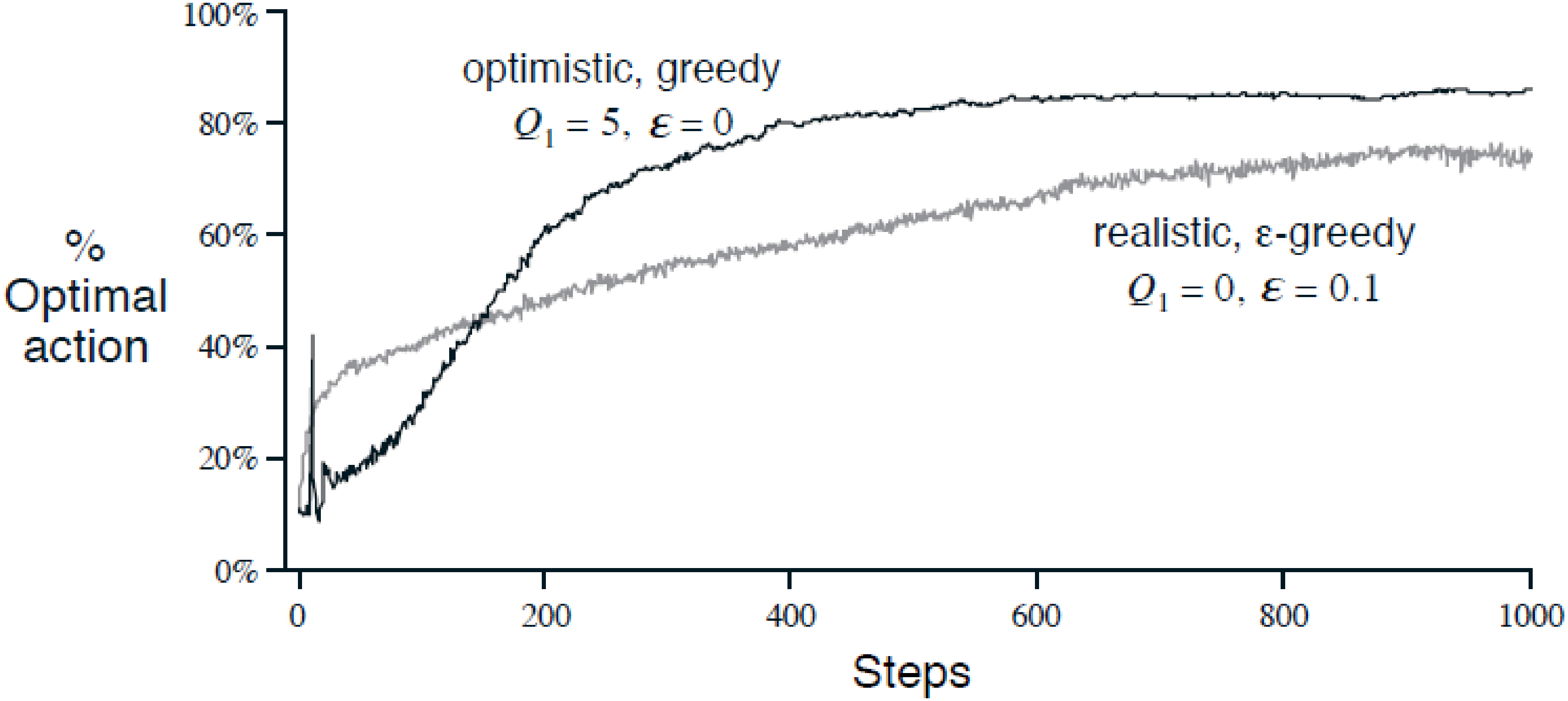
Zeros:	0	0		0
Random:	-0.23	0.76		-0.9
Optimistic:	+5	+5		+5

Arm 1

Arm 2

. . . .

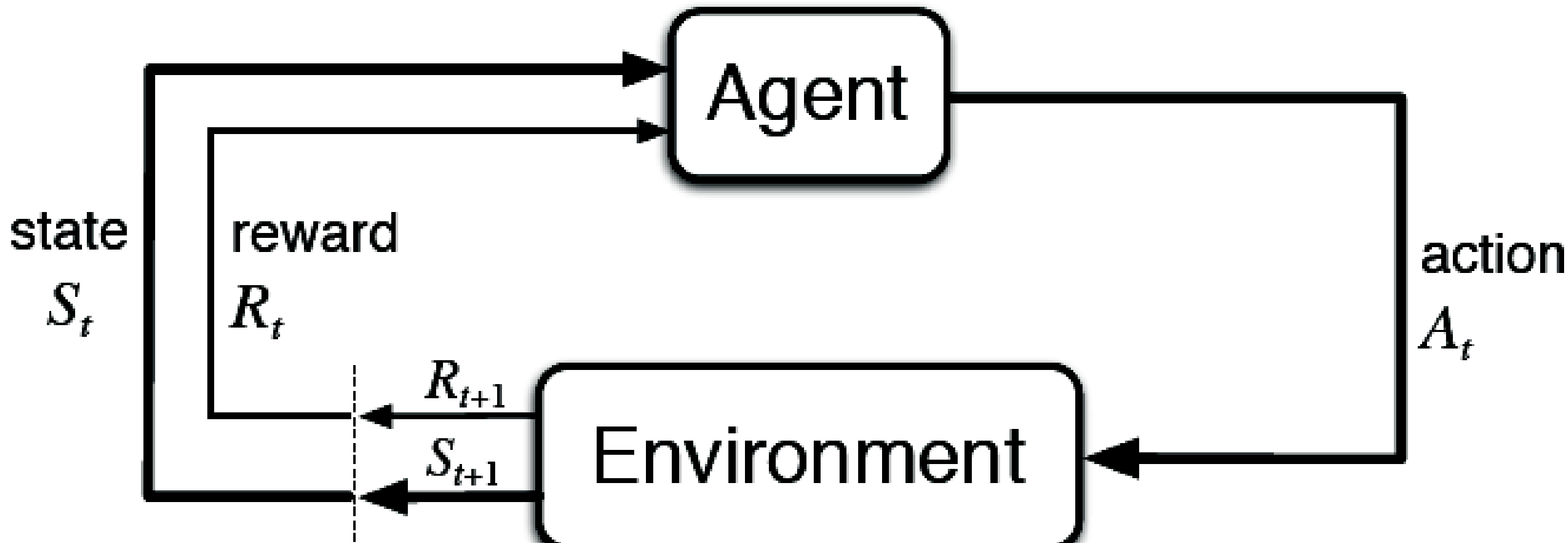
Arm k


100%
80%
60%
40%
20%
0%
% Optimal action
optimistic, greedy
$Q_1 = 5,\ \varepsilon = 0$
realistic, ε-greedy
$Q_1 = 0,\ \varepsilon = 0.1$
0
200
400
600
800
1000
Steps

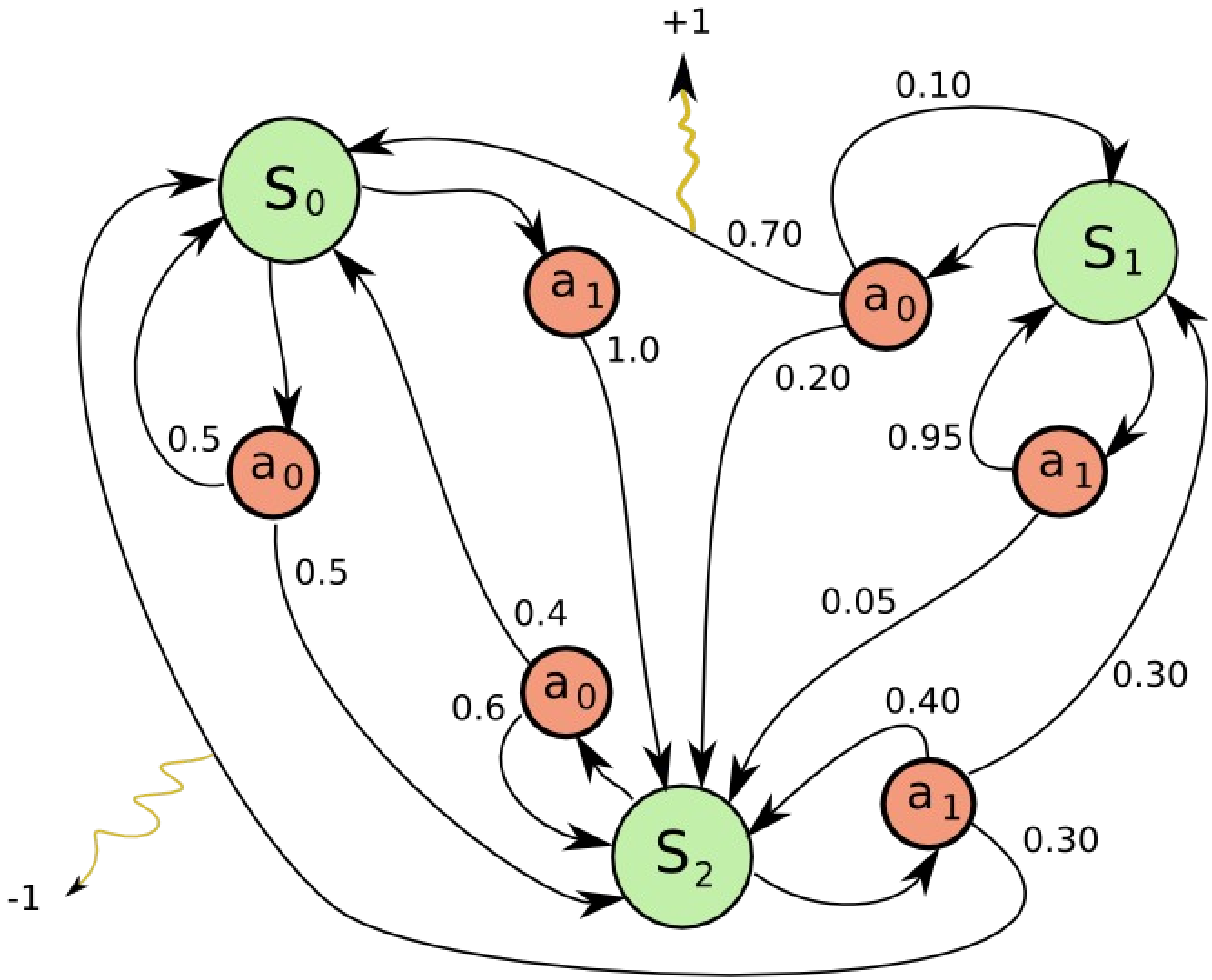
The Multi-Armed Bandit Problems

The casino always wins – so why is this problem important?

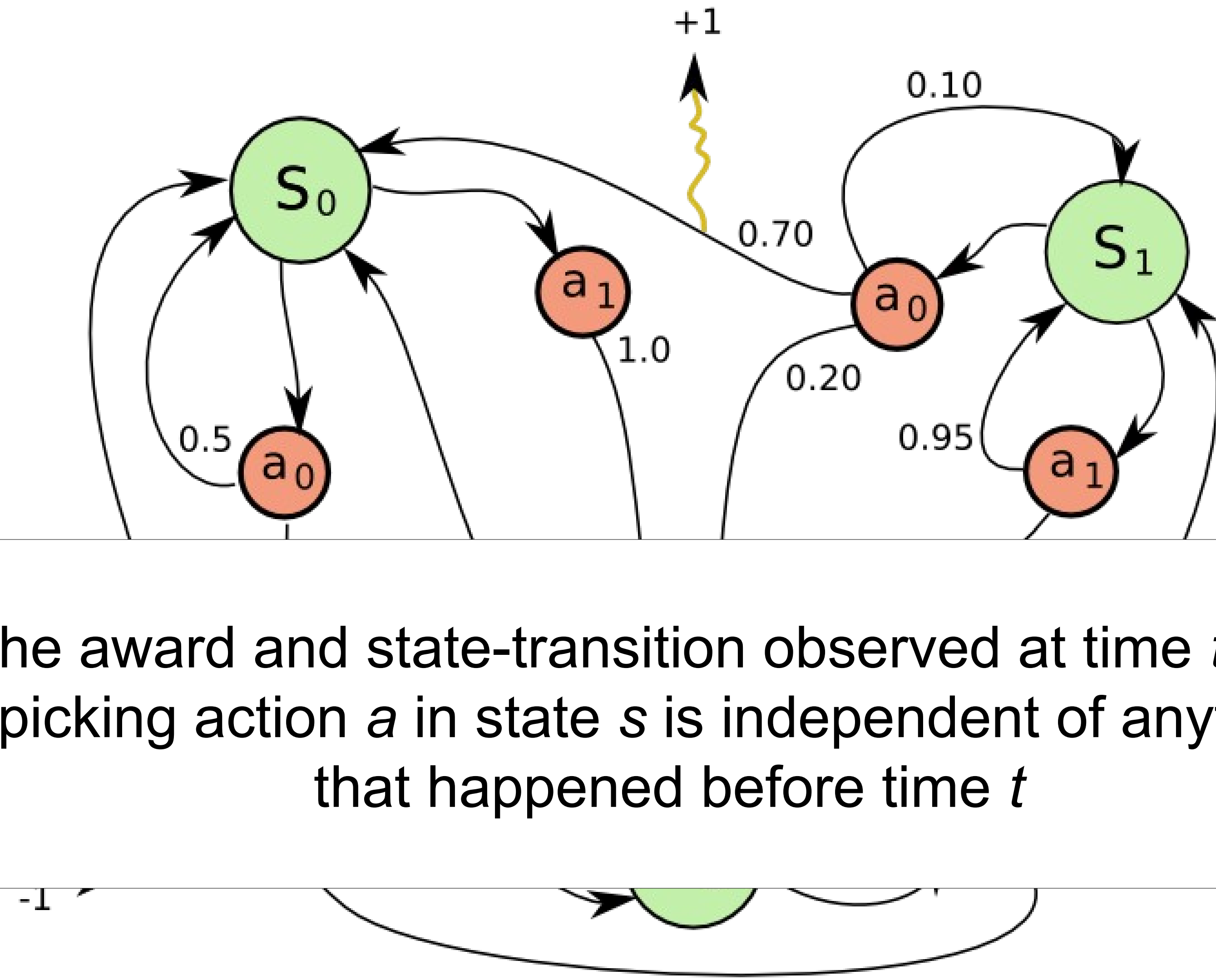
The Reinforcement Learning Problem



RL in the context of MDPs

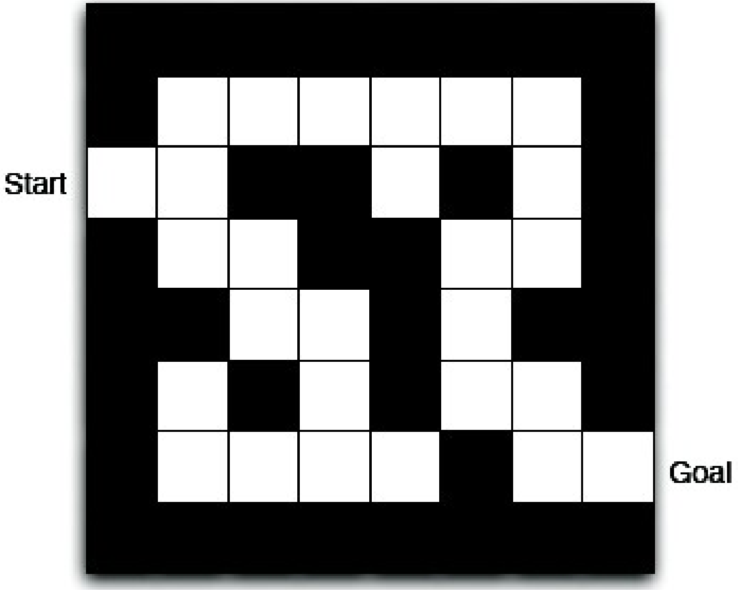


The Markov Assumption



The award and state-transition observed at time *t* after picking action *a* in state *s* is independent of anything that happened before time *t*

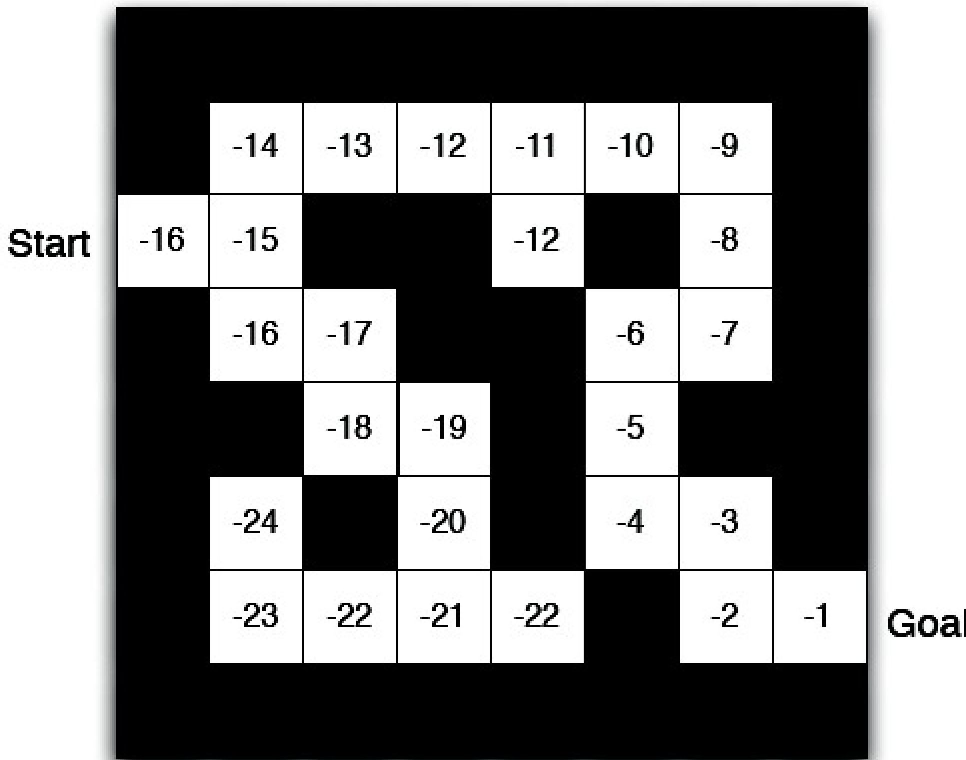
Maze Example



- Rewards: -1 per time-step
- Actions: N, E, S, W
- States: Agent's location

[slide credit: David Silver]

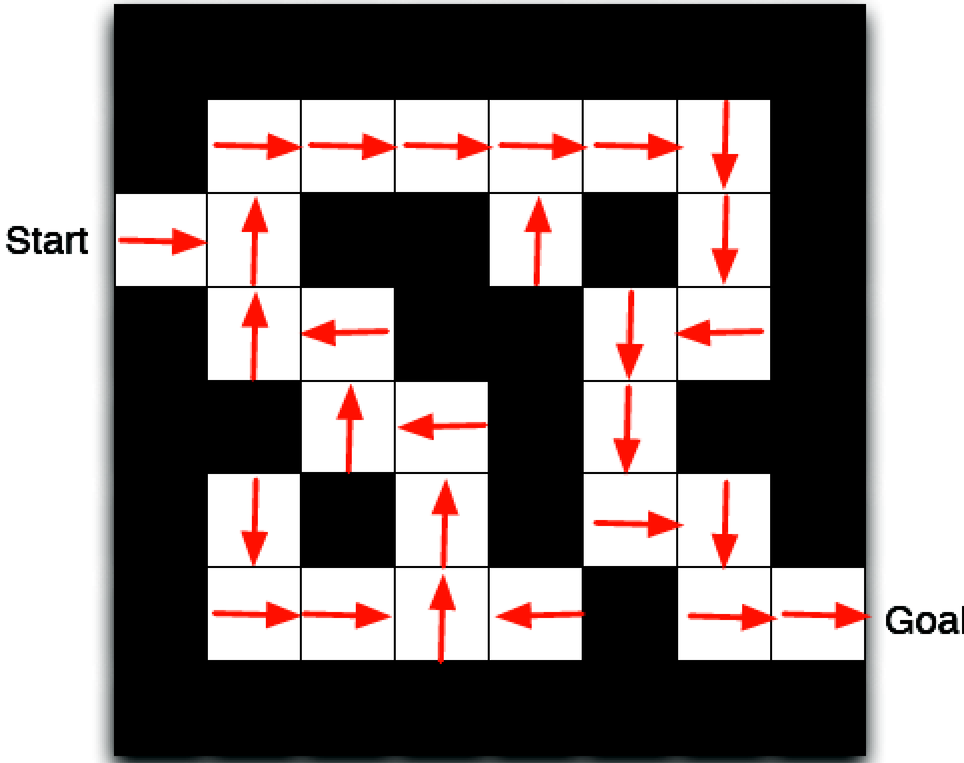
Maze Example: Value Function



- Numbers represent value $v_\pi(s)$ of each state s

[slide credit: David Silver]

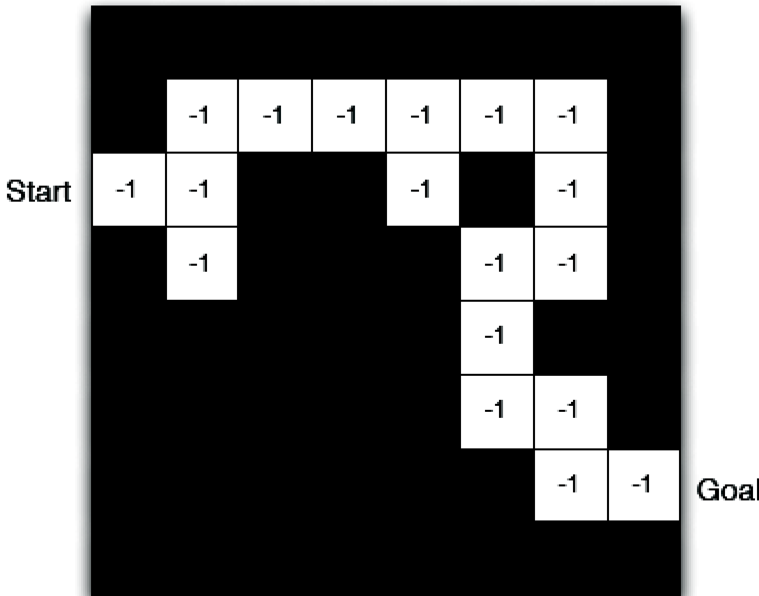
Maze Example: Policy



- Arrows represent policy $\pi(s)$ for each state s

[slide credit: David Silver]

Maze Example: Model



- Agent may have an internal model of the environment
- Dynamics: how actions change the state
- Rewards: how much reward from each state
- The model may be imperfect

- Grid layout represents transition model $\mathcal{P}^a_{ss'}$
- Numbers represent immediate reward $\mathcal{R}^a_s$ from each state s (same for all a)

[slide credit: David Silver]

Notation

Set of States: $\mathcal{S}$

Set of Actions: $\mathcal{A}$

Transition Function:

$$\mathcal{P} : \mathcal{S} \times \mathcal{A} \mapsto \Pi(\mathcal{S})$$

Reward Function:

$$\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$$

Action-Value Function

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$

Action-Value Function

Discount factor (between 0 and 1)

Probability of going to state *s'* from *s* after *a*

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$

The value of taking action *a* in state *s*

The reward received after taking action *a* in state *s*

a' is the action with the highest action-value in state s'

Action-Value Function

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$

Common algorithms to learn the action-value function include Q-Learning and SARSA

The policy consists of always taking the action that maximize the action-value function

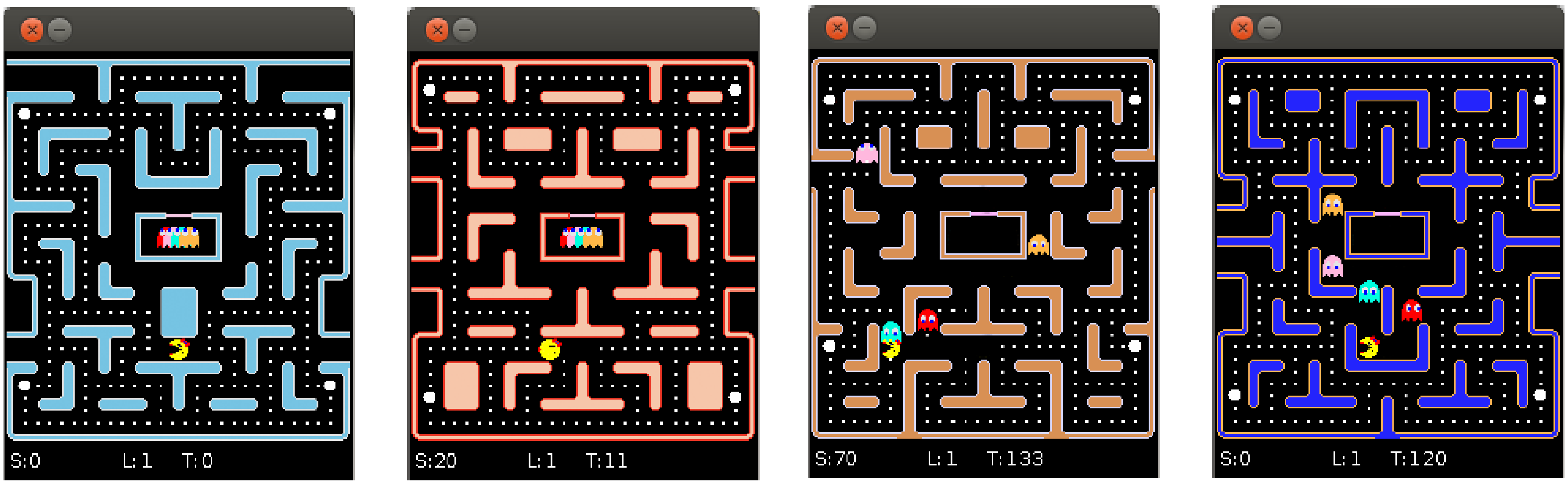
Q-Learning Example

- Example Slides

Q-Learning Algorithm

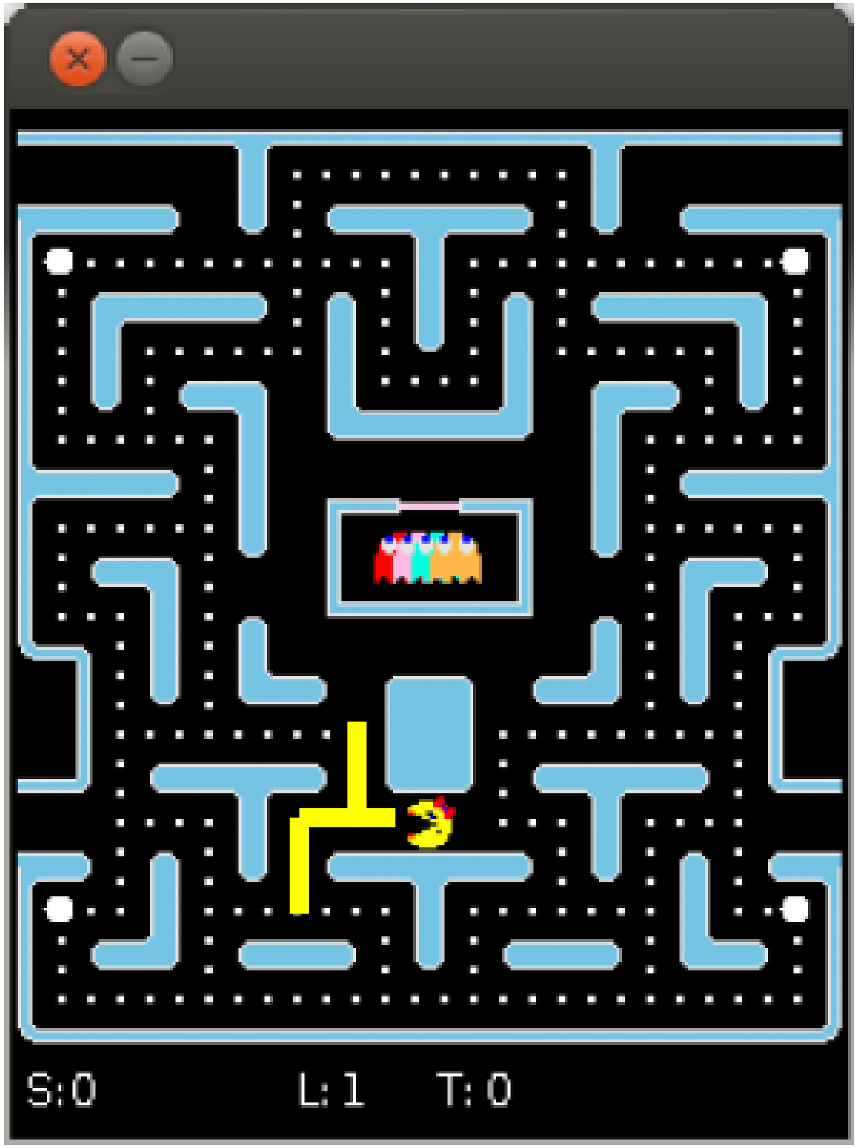
Initialize $Q(s, a)$ and $Model(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Do forever:
 (a) $s \leftarrow$ current (nonterminal) state
 (b) $a \leftarrow \varepsilon$-greedy(s, Q)
 (c) Execute action a; observe resultant state, s', and reward, r
 (d) $Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$
 (e) $Model(s, a) \leftarrow s', r$ (assuming deterministic environment)
 (f) Repeat N times:
  $s \leftarrow$ random previously observed state
  $a \leftarrow$ random action previously taken in s
  $s', r \leftarrow Model(s, a)$
  $Q(s, a) \leftarrow Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q(s', a') - Q(s, a)\right]$

Pac-Man RL Demo

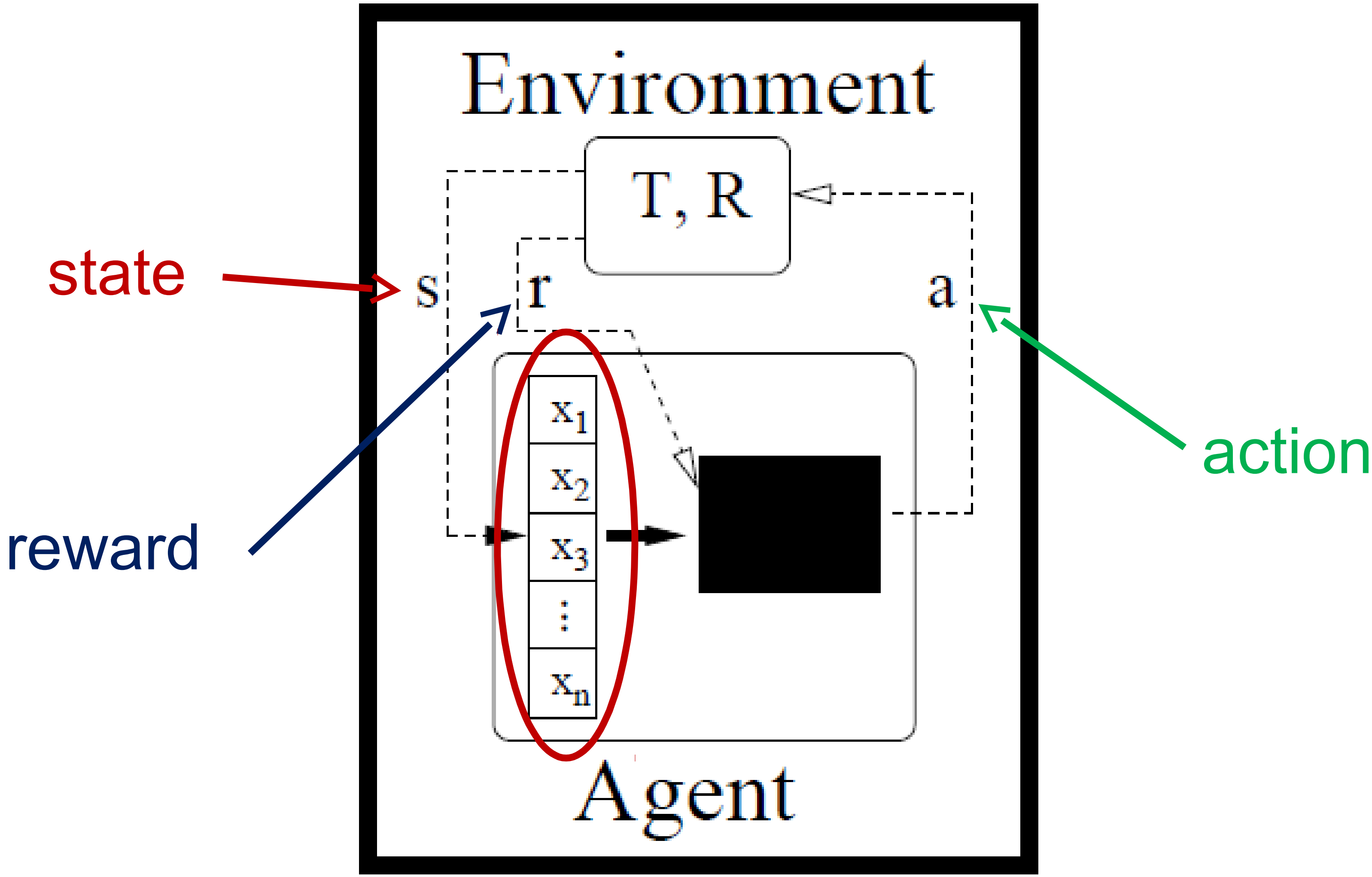


How does Pac-Man "see" the world?

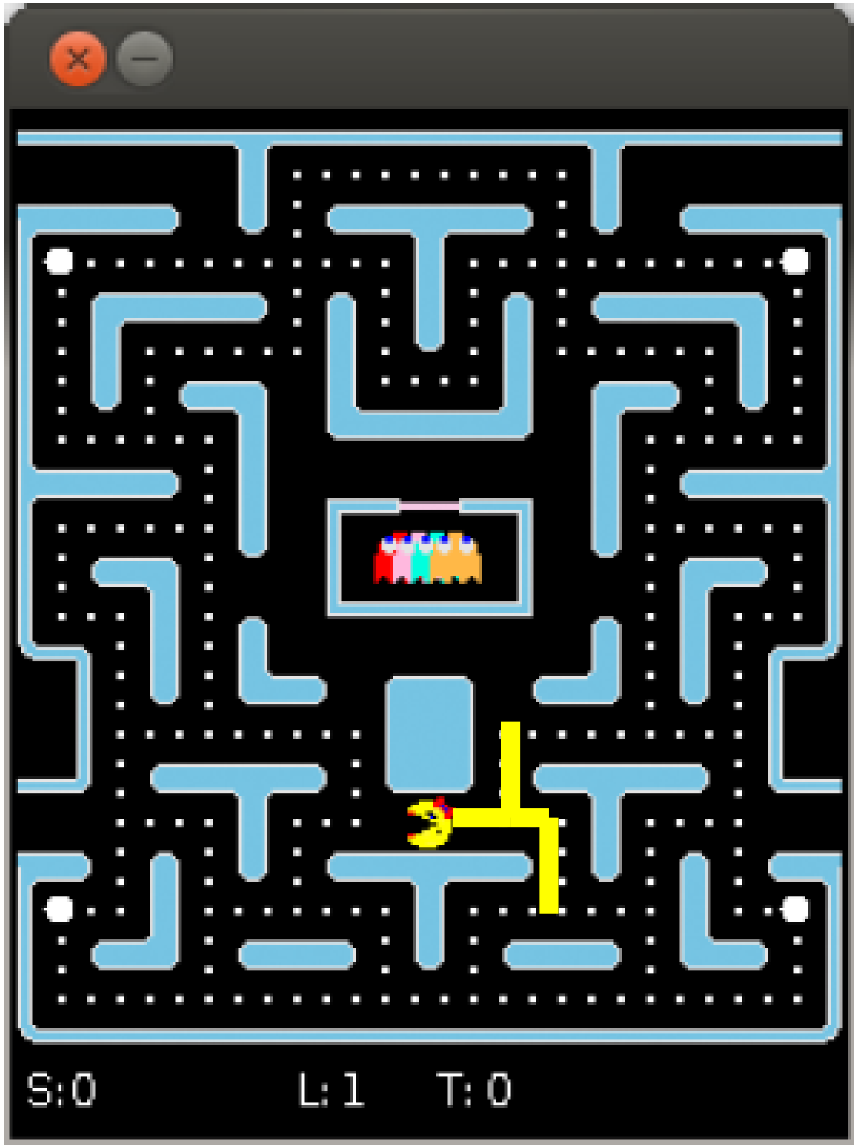
How does Pac-Man “see” the world?



The state-space may be continuous...



How does Pac-Man "see" the world?

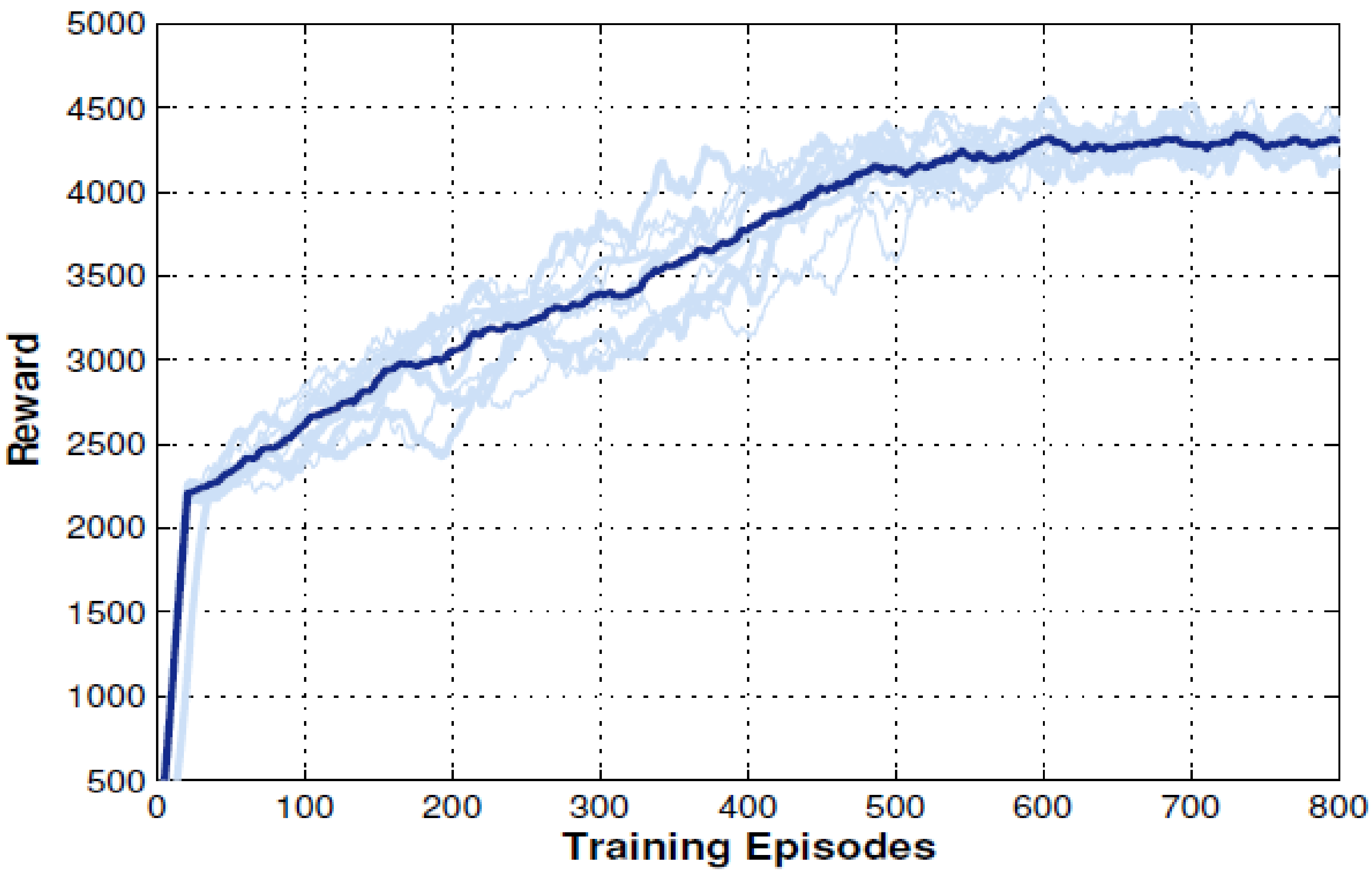


Q-Function Approximation

$$Q^*(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{P}(s'|s,a) \max_{a'} Q^*(s',a')$$

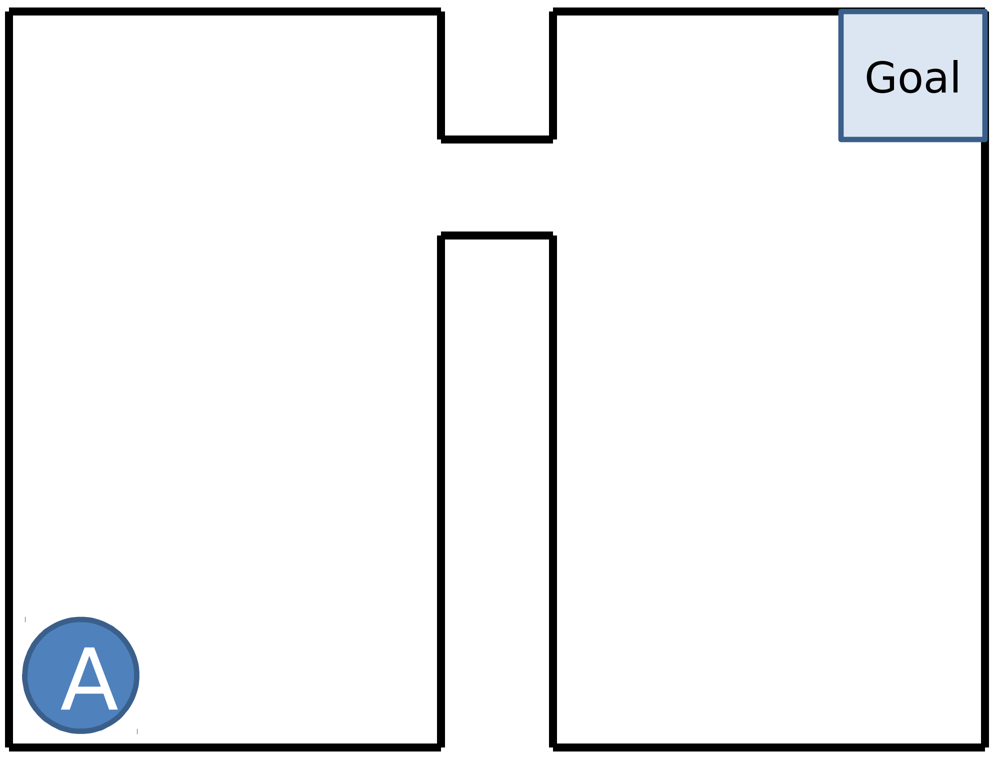
$a_1 * x_1 + a_2 * x_2 + \ldots + a_n * x_n$

Example Learning Curve

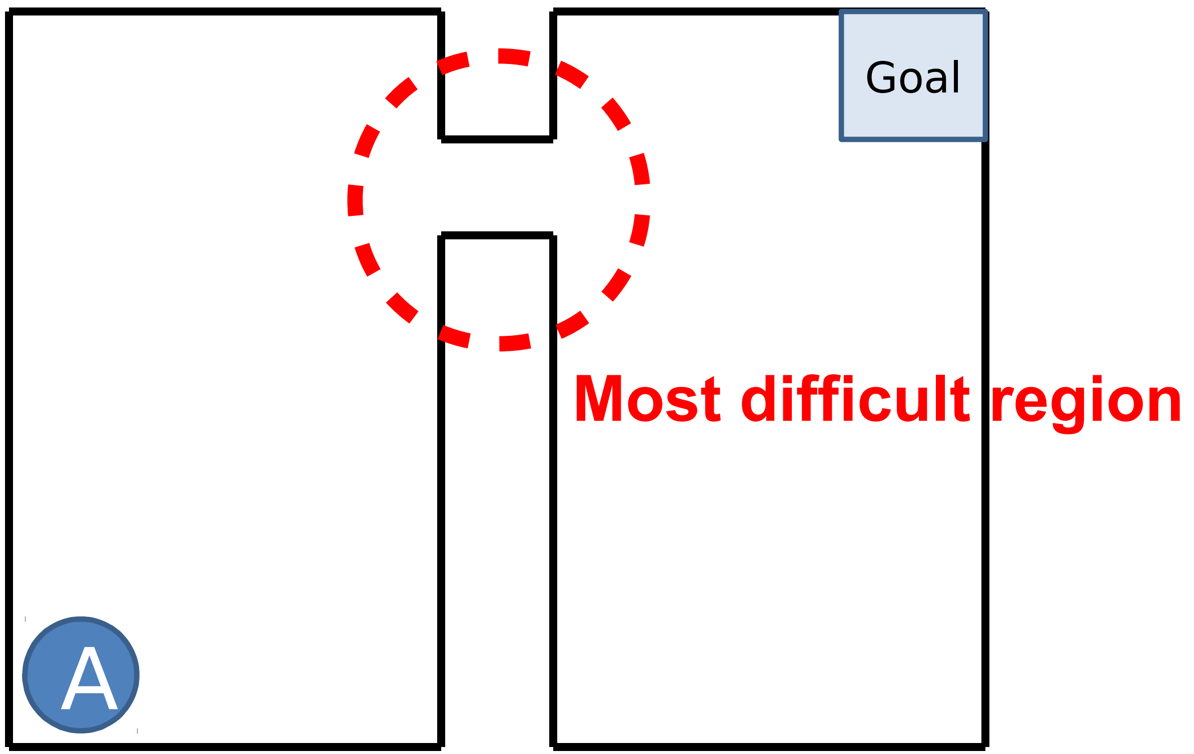


Sinapov *et al.* (2015). Learning Inter-Task Transferability in the Absence of Target Task Samples. In proceedings of the 2015 ACM Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Istanbul, Turkey, May 4-8, 2015.

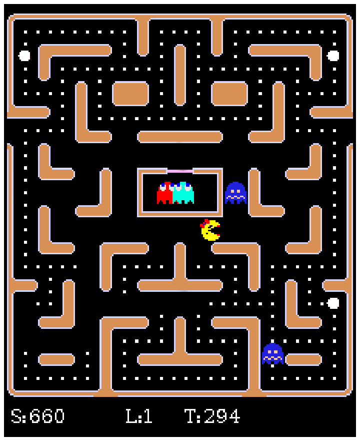
Curriculum Development for RL Agents



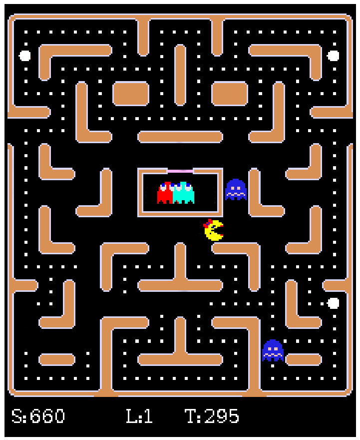
Curriculum Development for RL Agents



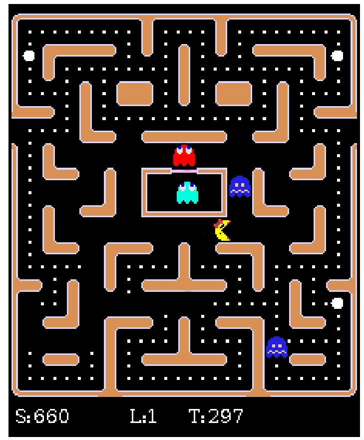
Main Approach



t-21

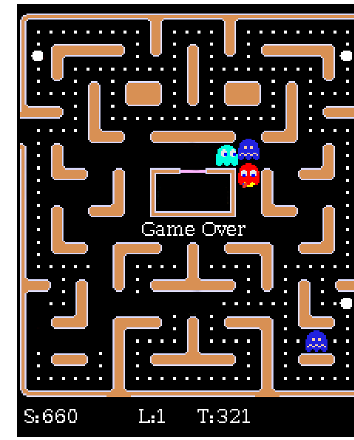


t-20



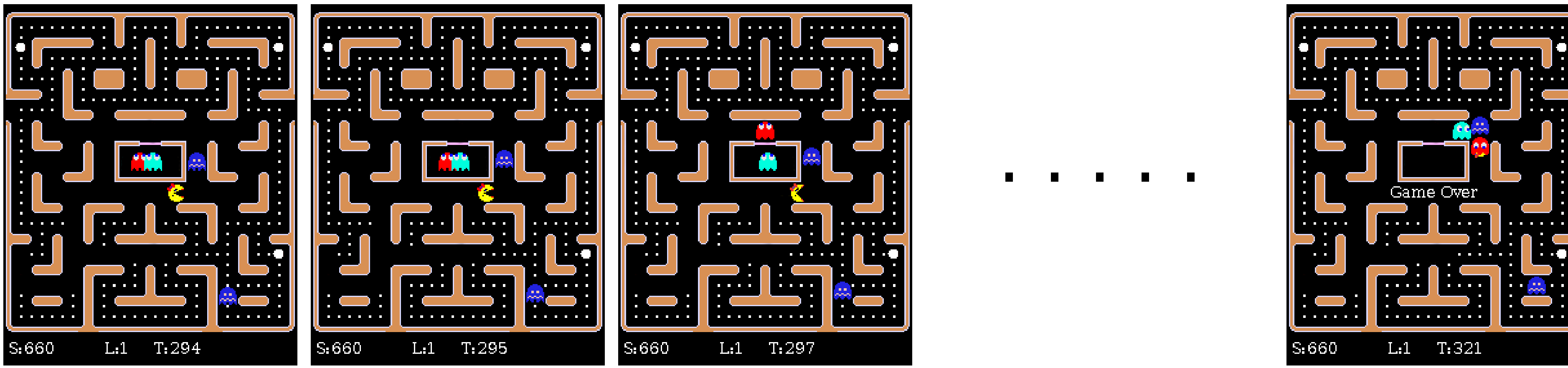
t-19

. . . . . .



t

Main Approach



t-21 t-20 t-19 t

Rewind back *k* game steps and branch out

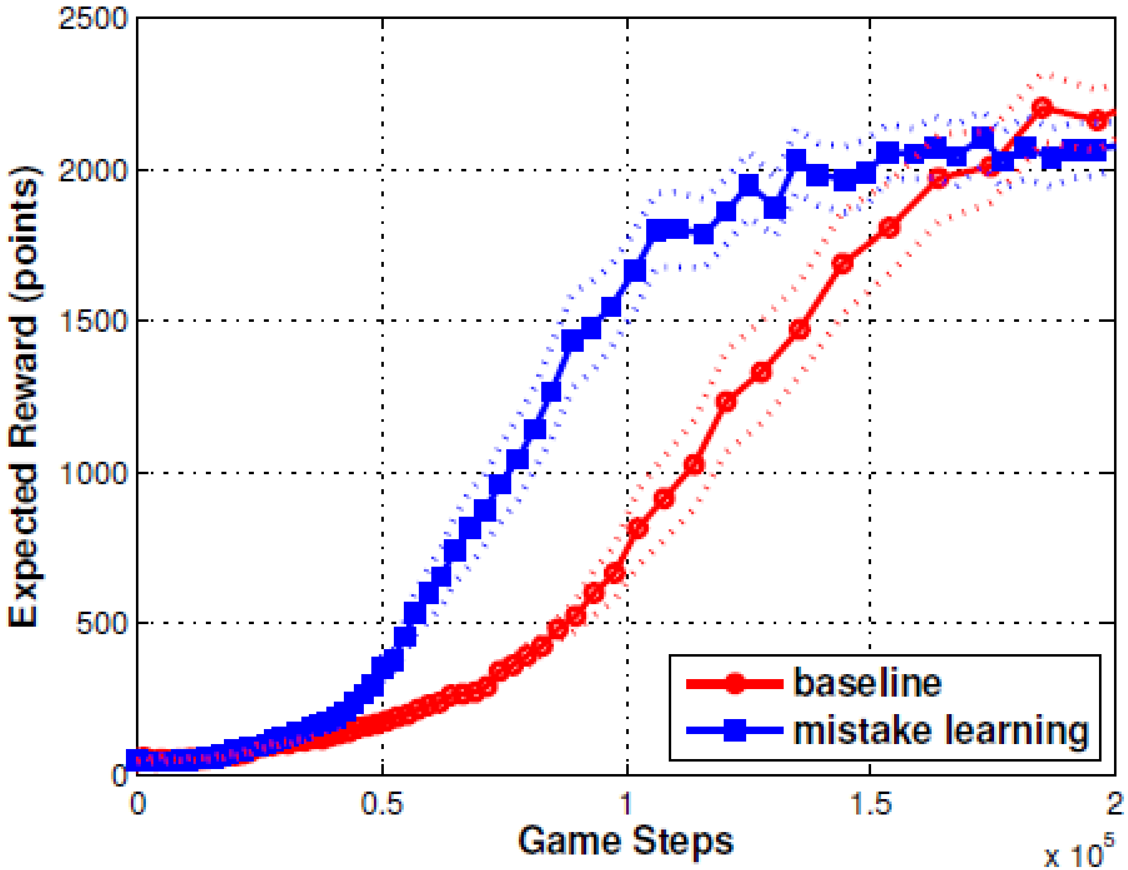


Figure 4: Results of MistakeLearning applied to the Ms. Pac-Man domain. See Section 5.1.2 for details. Dashed lines indicate standard error.

Narvekar, S., Sinapov, J., Leonetti, M. and Stone, P. (2016). Source Task Creation for Curriculum Learning. To appear in proceedings of the 2016 ACM Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)

Resources

- BURLAP: Java RL Library: http://burlap.cs.brown.edu/
- Reinforcement Learning: An Introduction

  http://people.inf.elte.hu/lorincz/Files/RL_2006/SuttonBook.pdf

THE END