MULTI-AGENT LEARNING

From theory to practice

Ann Nowé
AI Lab
ai.vub.ac.be

VRIJE UNIVERSITEIT BRUSSEL

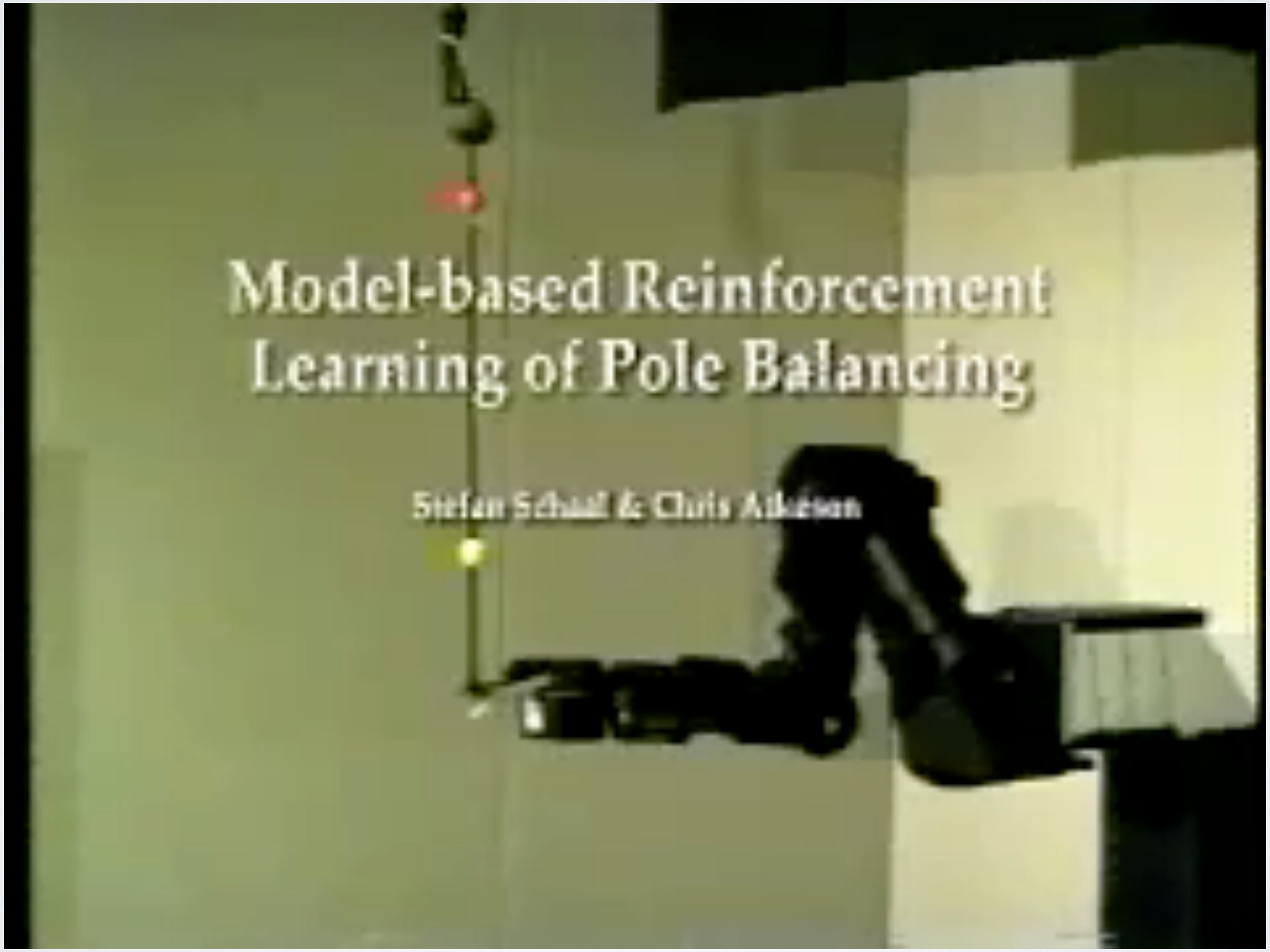
REINFORCEMENT LEARNING

- Origin in psychology
- Learning from interaction
- Learning **about**, **from**, and **while** interacting with
- Learning **what** to do - how to map situations to actions - so as to **maximise** a numerical

KEY FEATURES OF RL

- Learner is **not** told which action to take
- Trial-and-error approach
- Possibility of **delayed reward**
  - Sacrifice short term gains for greater long-term gains
- Need to balance **exploration** and **exploitation**
- Considers the whole problem of a goal-oriented agent interacting with an uncertain environment

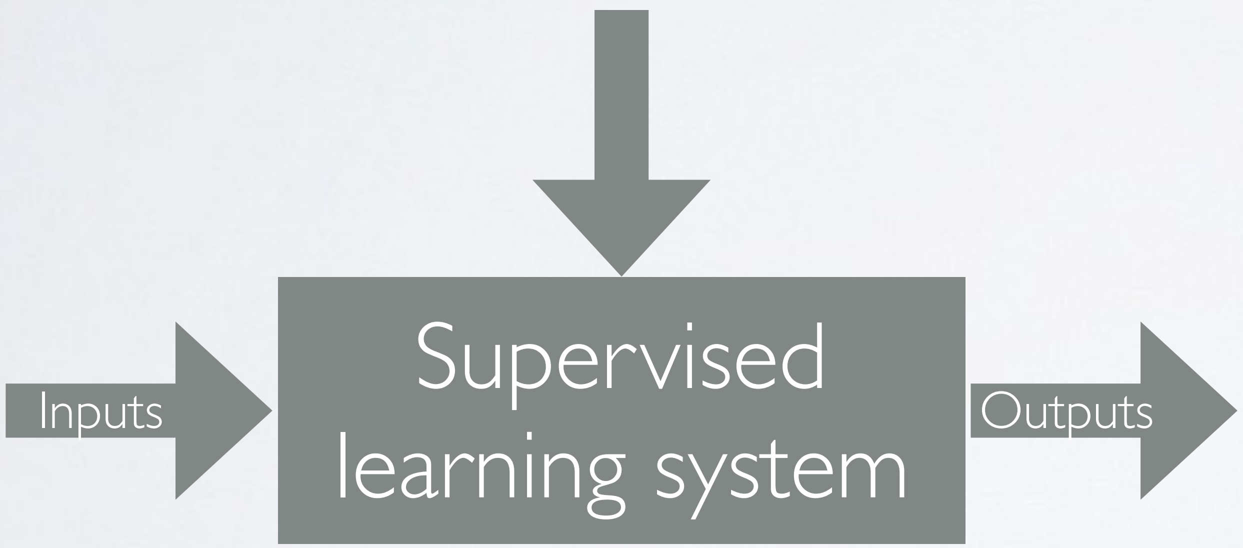
POLE BALANCING DEMO



SUPERVISED VS UNSUPERVISED

Supervised learning

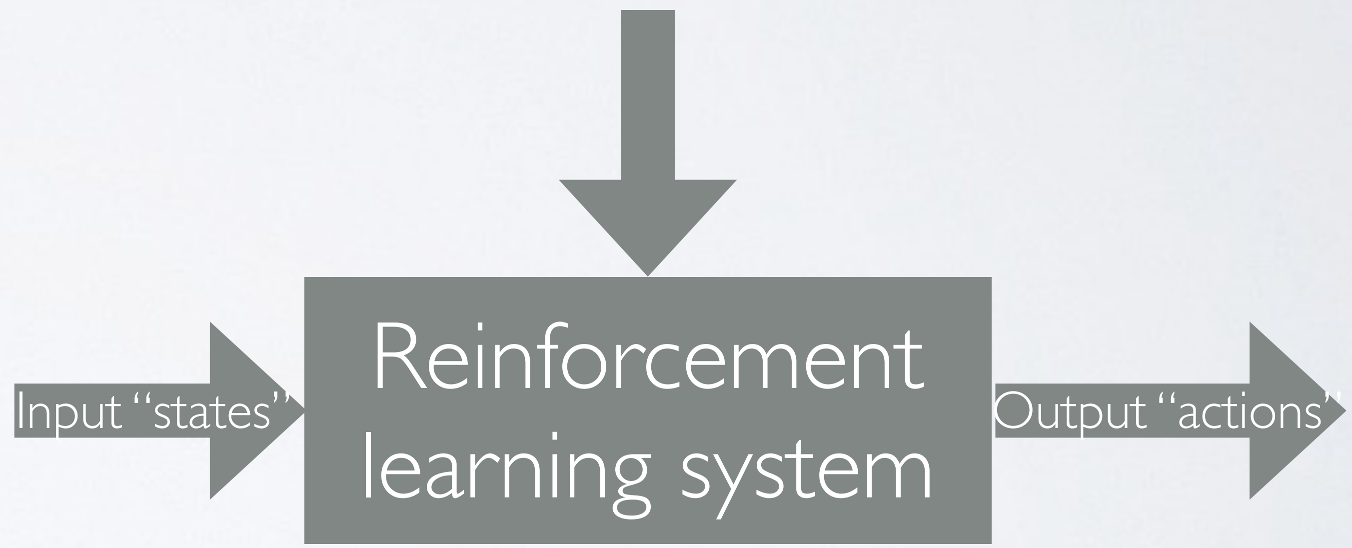
Training info = desired (target) outputs



Error = (target output - actual output)

Unsupervised learning

Training info = evaluations

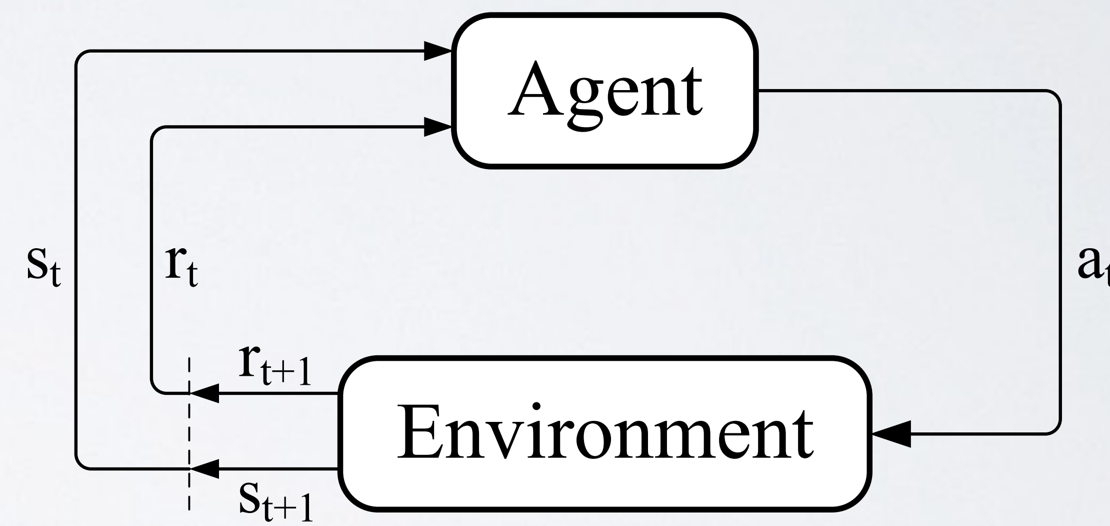


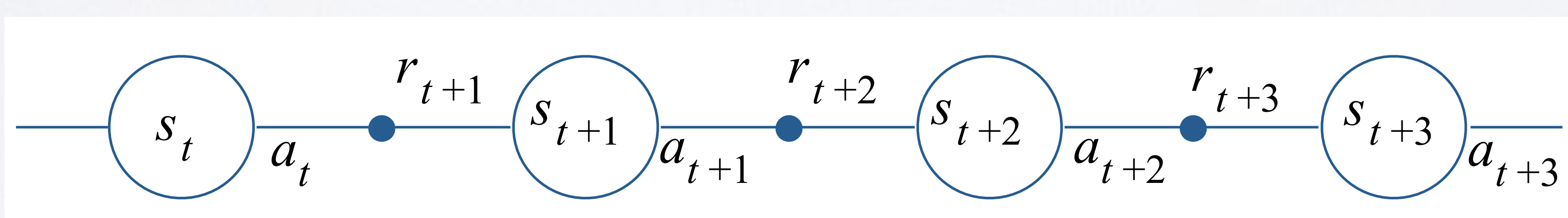
Objective: get as much reward as possible

THE AGENT-ENVIRONMENT INTERFACE

Agent interacts at discrete time steps $t = 0, 1, 2, \ldots$

- Observes state $s_t \in S$
- Selects action $a_t \in A(s_t)$
- Obtains immediate reward $r_{t+1} \in \mathbb{R}$
- Observes resulting state s_{t+1}





LEARNING HOW TO BEHAVE

- The agent's **policy** π at time t is
  - a mapping from states to action probabilities
  - $\pi_t(s, a) = P(a_t = a | s_t = s)$

- Reinforcement learning methods specify **how** the agent changes its policy as a result of experience
- Roughly, the agent's goal is to **get as much reward** as it can over the long run

THE OBJECTIVE

- Episodic tasks: interaction breaks naturally into episodes, e.g., plays of a game

$$R_t = r_{t+1} + \underbrace{r_{t+2} + \ldots + r_T}_{\text{Long term reward}}$$

Immediate reward

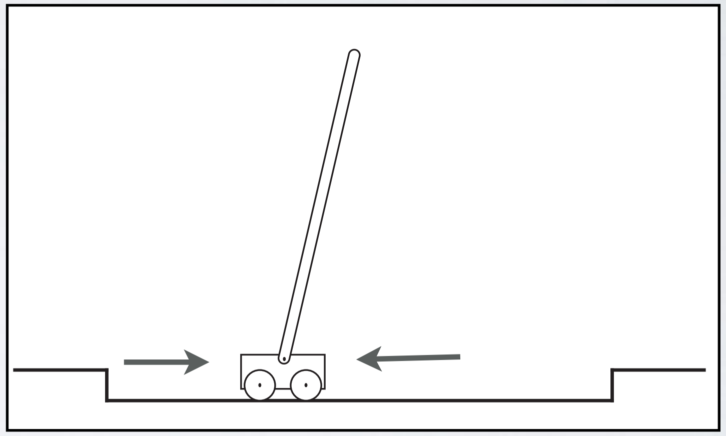
- Continuing tasks: interaction does to have natural episodes

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

- where $\gamma \in [0, 1]$ is the **discount factor**

EXAMPLE: POLE BALANCING

- An **episodic** task where episode ends upon failure:
  - reward = +1 for each step
  - return = # steps before failure



- A **continuing** task with discounted return:
  - reward = -1 upon failure
  - return $= -\gamma^k$, for *k* step before failure
- Return is maximized by avoiding failure as long as possible

$$\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

THE OBJECTIVE

- Goal: learn $\pi : S \rightarrow A$ (could be stochastic)
- That maximises:

$$
\begin{aligned}
V^{\pi}(s) &= E\{r_{r+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \ldots | s_t = s, \pi\} \\
&= E\{r_{t+1}\} + \gamma \cdot E\{r_{t+2} + \gamma \cdot r_{t+3} + \ldots | s_{t+1} = s, \pi\}
\end{aligned}
$$

rewards

$$V^*(s) = \max_{\pi} V^{\pi}(s) \forall s$$

INTUITIVELY

$V^{\pi}(s)$ values express how good a state is given a policy π

$Q^{\pi}(s, a)$ express how good it is to apply action a in state s, and from the next state on apply π

INTUITIVELY

$V^{\pi^*}(s)$ values express how good a state is given the optimal policy π^*

$Q^{\pi^*}(s,a)$ express how good it is to apply action a in state s, and from the next state on apply π^*

$$V^*(s) = \max_a Q^*(s,a)$$

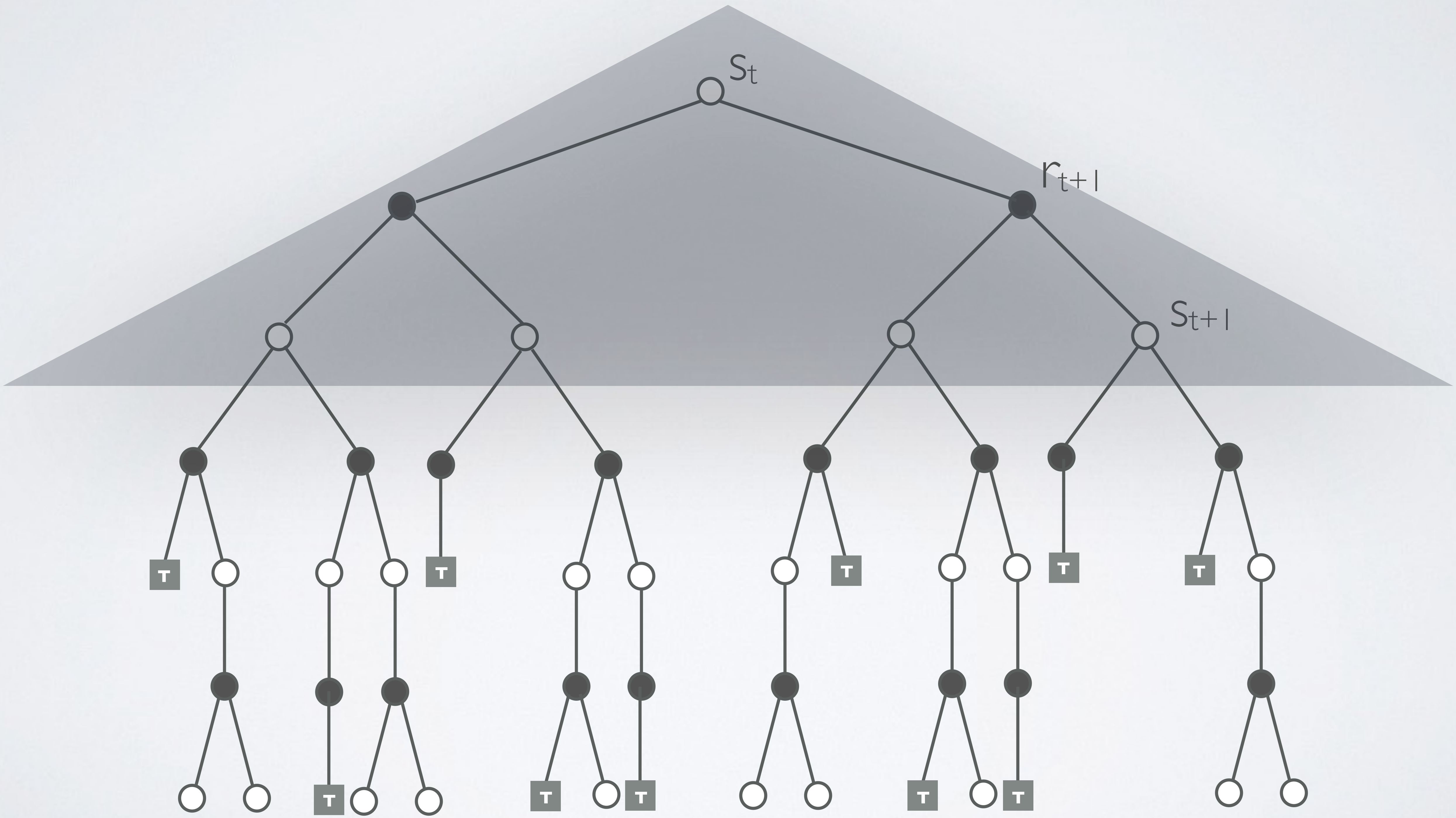
Q-LEARNING

One-step Q-learning:

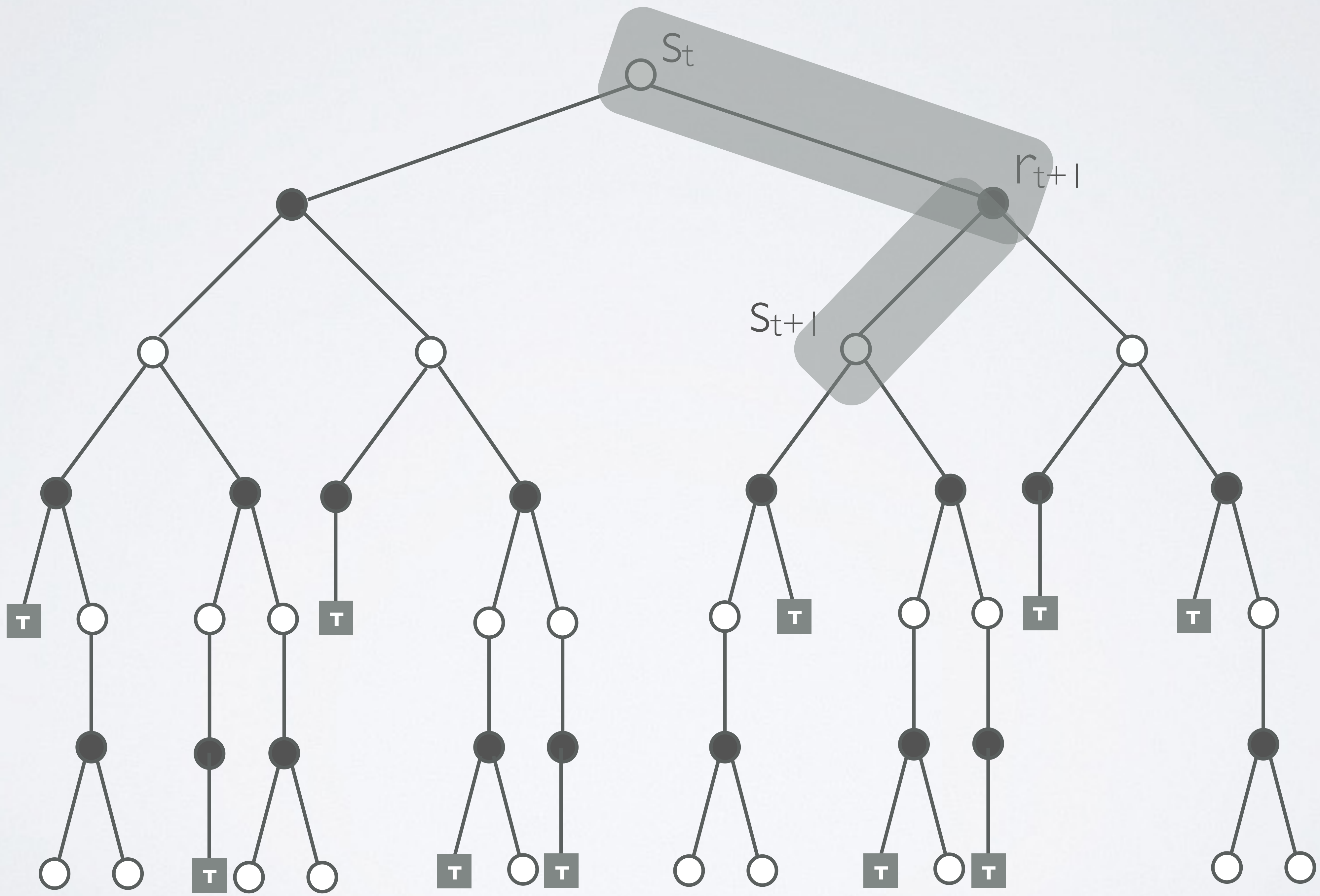
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s,a)]$$

```
Initialize Q(s,a) arbitrarily
Repeat (for each episode):
    Initialize s
    Repeat (for each step of episode):
        Choose a from s using policy derived from Q (e.g., ε-greedy)
        Take action a, observe r, s'
        Q(s,a) ← Q(s,a) + α[r + γ max_a' Q(s',a') − Q(s,a)]
        s ← s';
    until s is terminal
```

DYNAMIC PROGRAMMING : MODEL BASED



REINFORCEMENT LEARNING : MODEL FREE



EXPLORATION - EXPLOITATION

- Random action selection
- Greedy action selection $a_t = a_t^* = \operatorname*{argmax}_a Q_t(a)$
- ϵ-Greedy action selection

$$a_t = \begin{cases} a_t^* & \text{with probability 1 - } \epsilon \\ \text{random action} & \text{with probability } \epsilon \end{cases}$$

- Softmax action selection $\dfrac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}}$
- Exploration bomus (curiosity driven)
- Regret minimisation

EXTENSIONS FOR PRACTICAL APPLICATIONS

Continuous states and actions

- Deep NN, Kernels, Tile coding, fuzzy, etc.

Take advantage of **asynchronous** updates

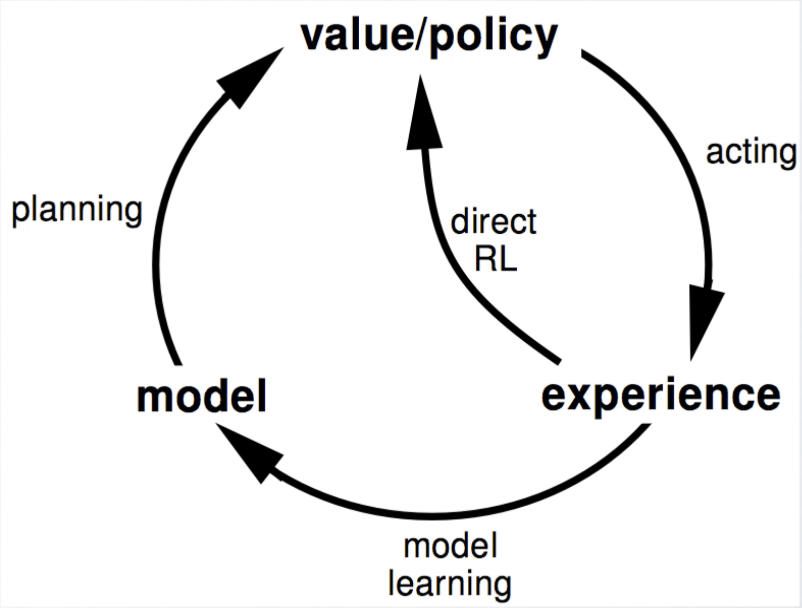
- Propagate interesting information more quickly
- Prioritized sweeping, eligibility traces

Incorporate **domain knowledge**

- Initialise policy
- Steer exploration
- Combine with model information (planning)

Continuous **time** extensions

Multi-criteria



CONVERGENCE OF Q-LEARNING

Q-learning is guaranteed to converge in an MDP setting

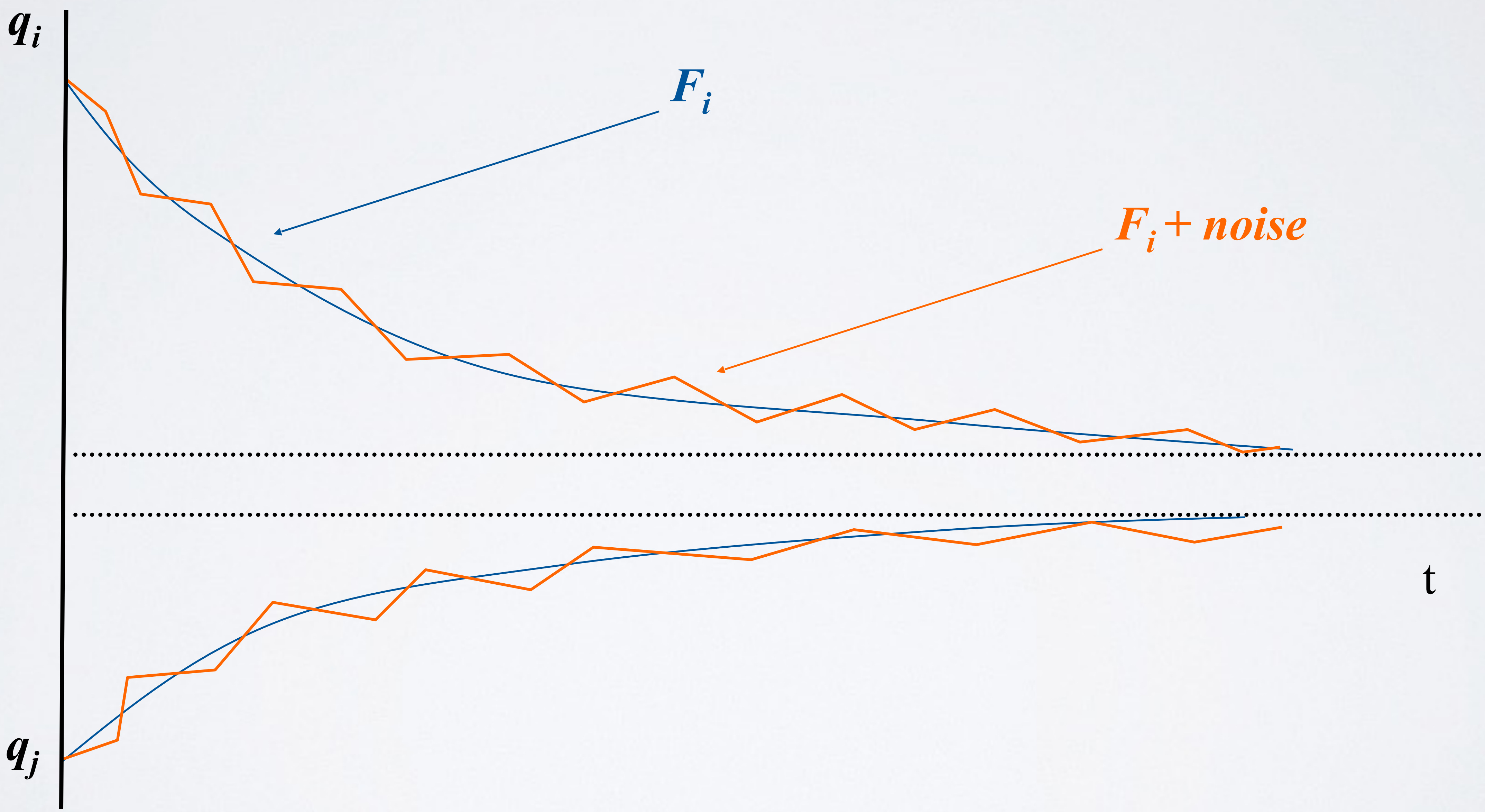
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(t) \Big(\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \Big] - Q(s_t, a_t) \Big)$$

$$\begin{aligned} Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(t) \Big(& E\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \Big] - Q(s_t, a_t)] \\ & + \Big((r_{t+1} + \gamma \max_a Q(s_{t+1}, a)) - E\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) \Big] \Big) \Big) \end{aligned}$$

Tsitsiklis, J.N. *Asynchronous Stochastic Approximation and Q-learning. Machine Learning,* Vol 16:pp185-202, 1994.

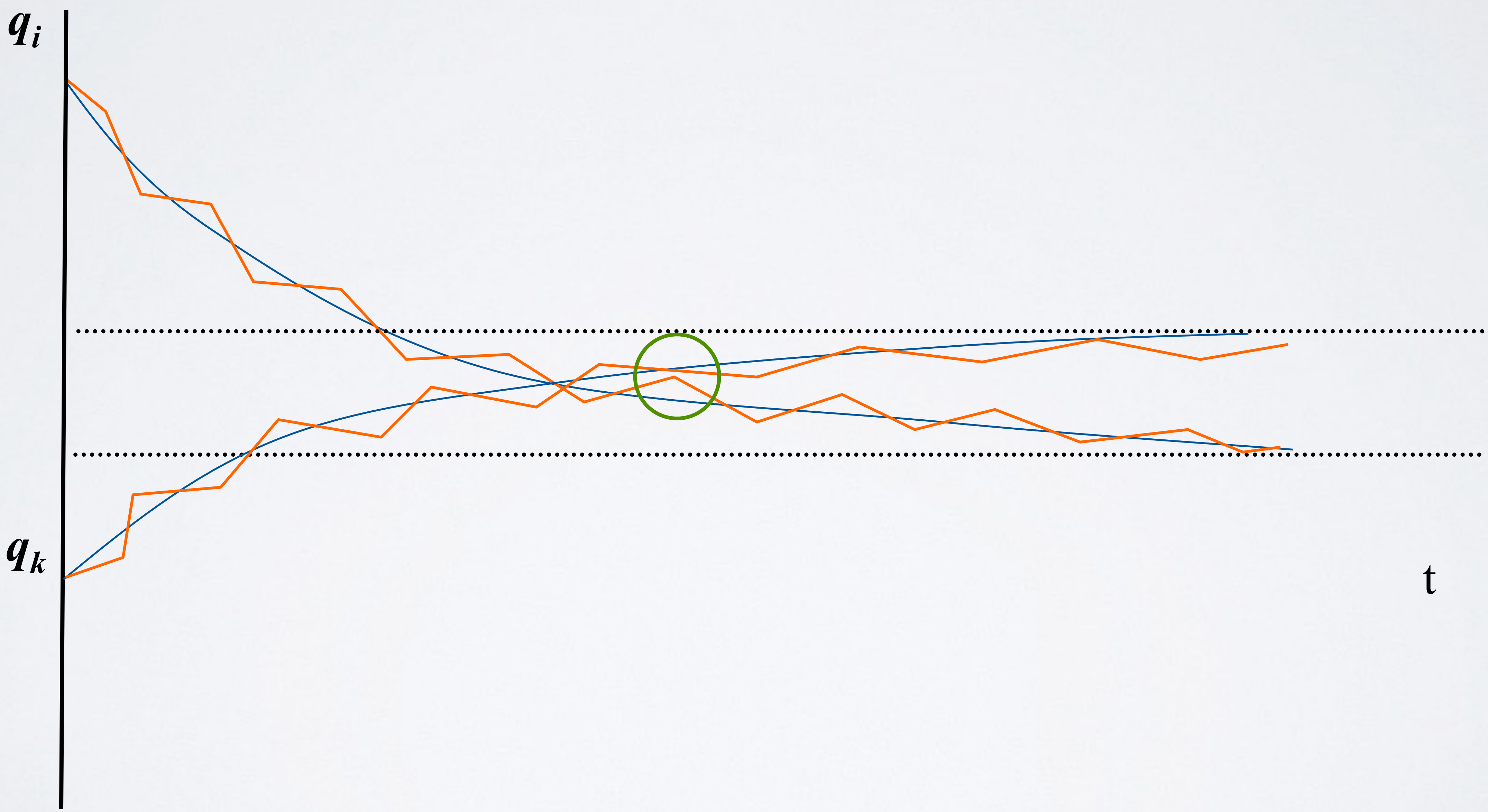
PROOF BY TSITSIKLIS, CONT.

Stochastic approximation

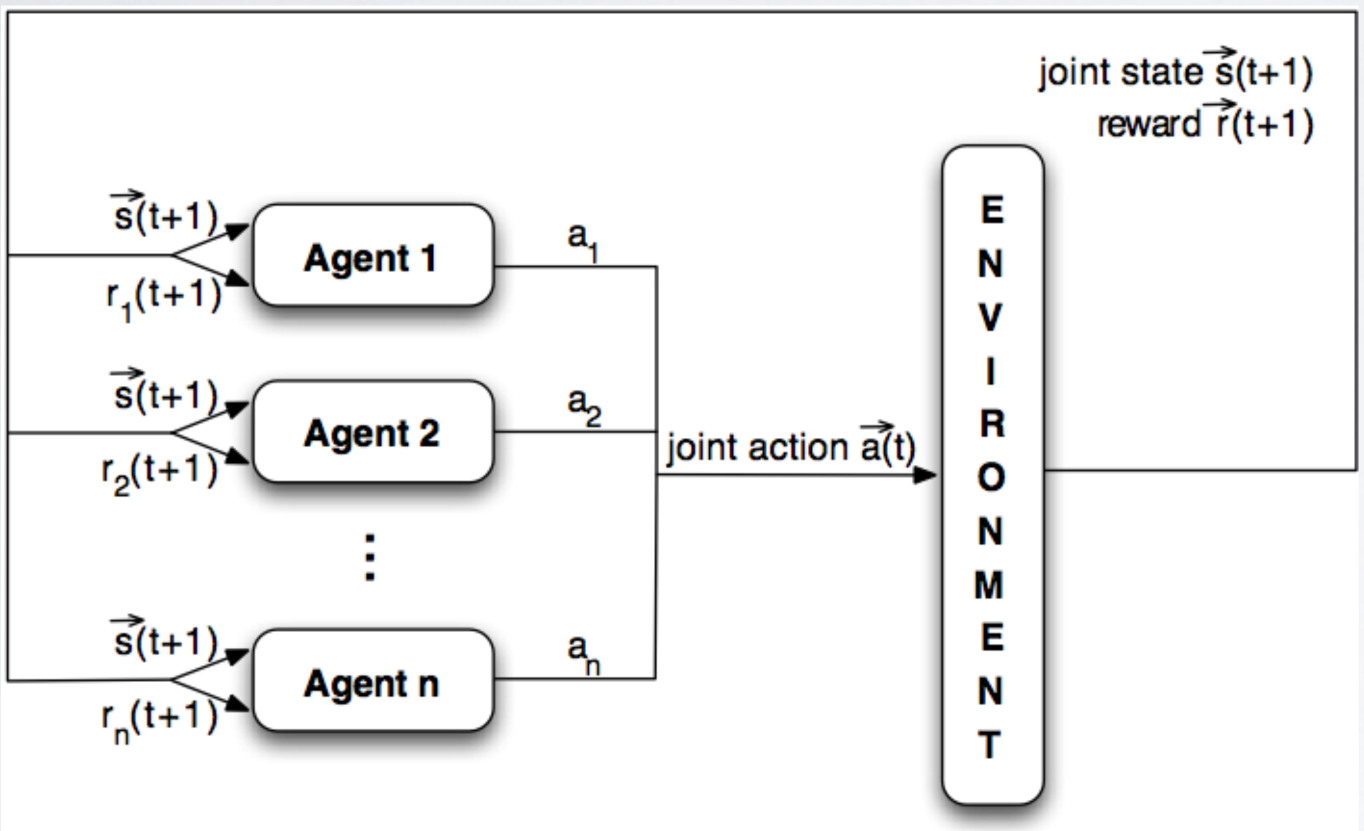


PROOF BY TSITSIKLIS, CONT.

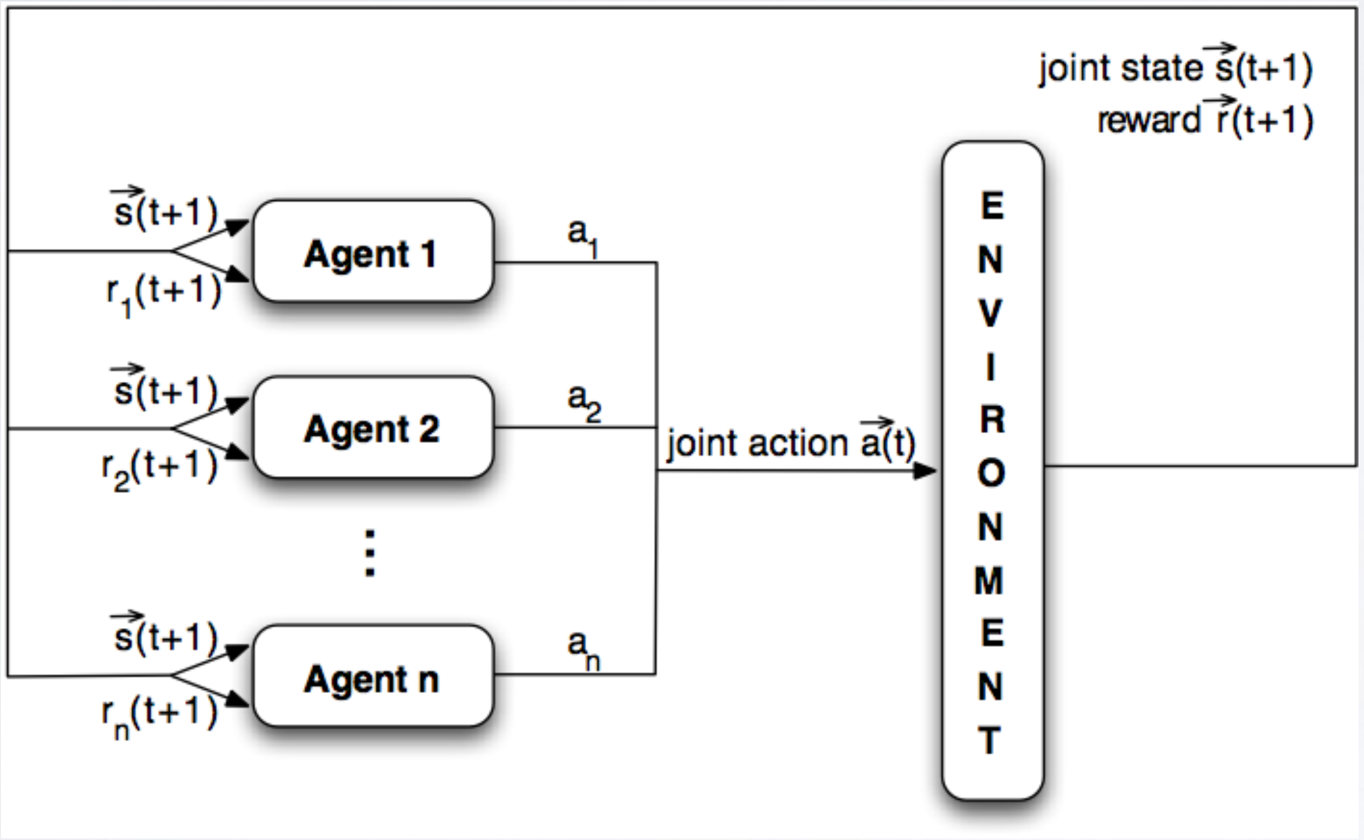
Stochastic approximation

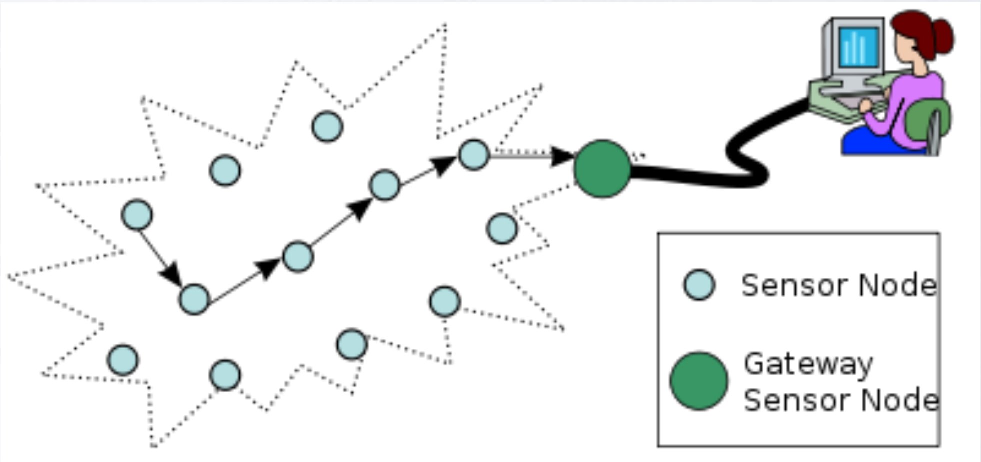


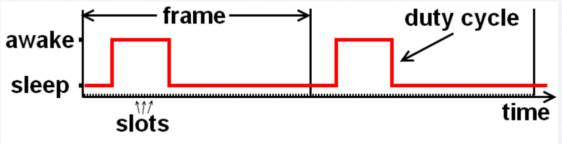
MULTI-AGENT RL



MULTI-AGENT RL







LEARNING AUTOMATA

Players in an n-person non-zero sum game who use independently a reward-inaction update scheme with an arbitrarily small step size will always converge to a pure equilibrium point. (Narendra and Wheeler, 1989)

If the game has a pure NE, the equilibrium point will be one of the pure NE.

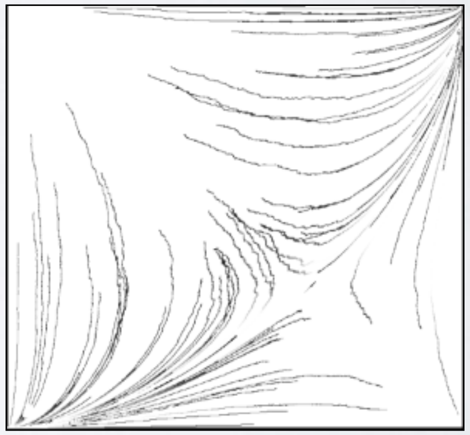
Interesting building block to design MARL algorithms.

Dynamics can be studied through evolutionary game theory

Daan Bloembergen, Karl Tuyls, Daniel Hennes, Michael Kaisers:
Evolutionary Dynamics of Multi-Agent Learning: A Survey. J. Artif. Intell. Res. 53: 659-697 (2015)

Has also been used to understand convergence of ACO

Verbeeck K., Nowé A. *Colonies of Learning Automata*, IEEE Transactions on Systems, Man and Cybernetics 2002.



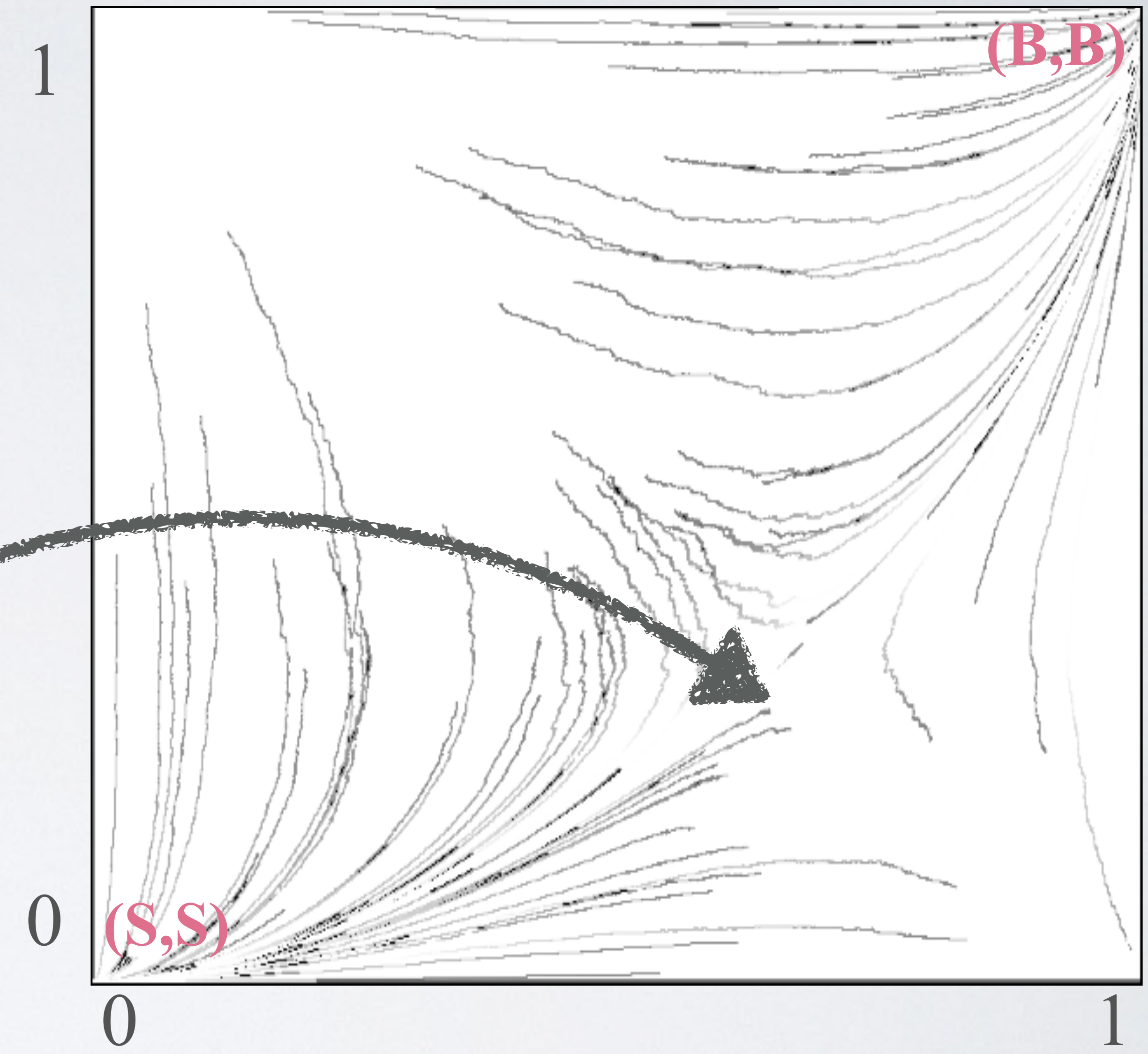
LA IN STRATEGIC GAMES

Battle of the sexes

	Bach	Strav.
Bach	2,1	0
Strav.	0	1,2

2 pure Nash equilibria

1 mixed Nash equilibrium
((2/3 B, 1/3 S) , (1/3 B, 2/3 S))



Paths induced by a linear reward-inaction LA.

Starting points are chosen randomly

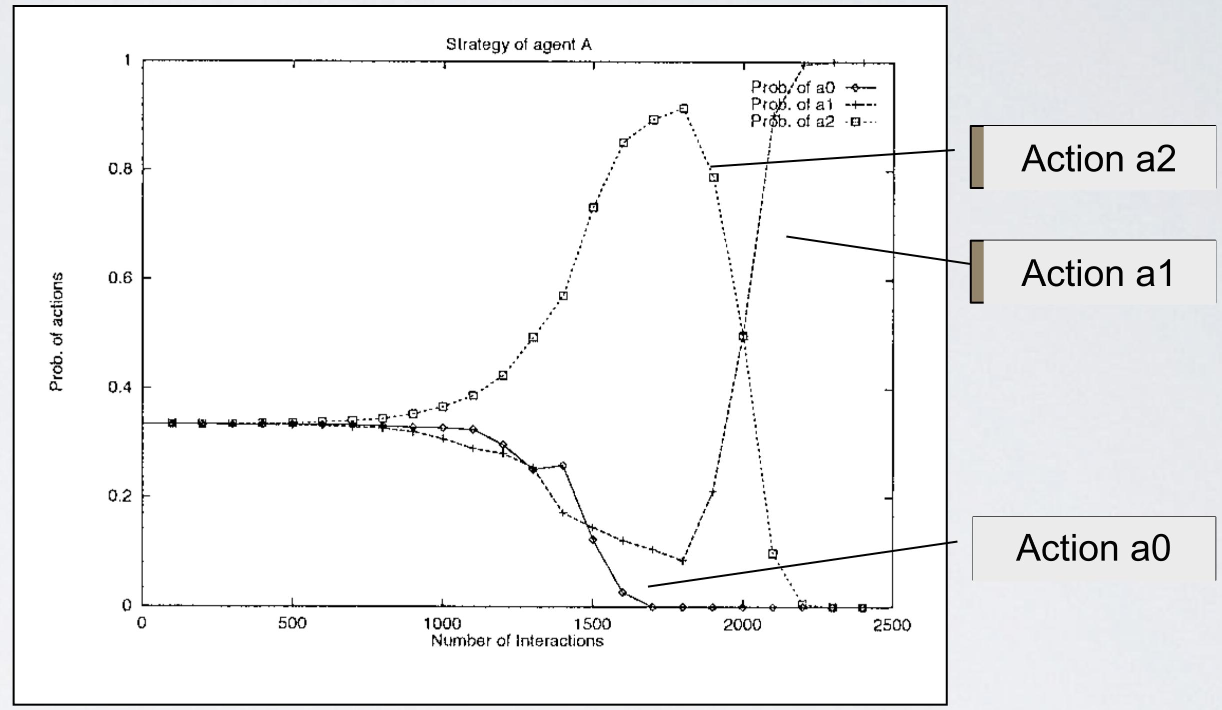
x-axis = prob. of the first player to play Bach

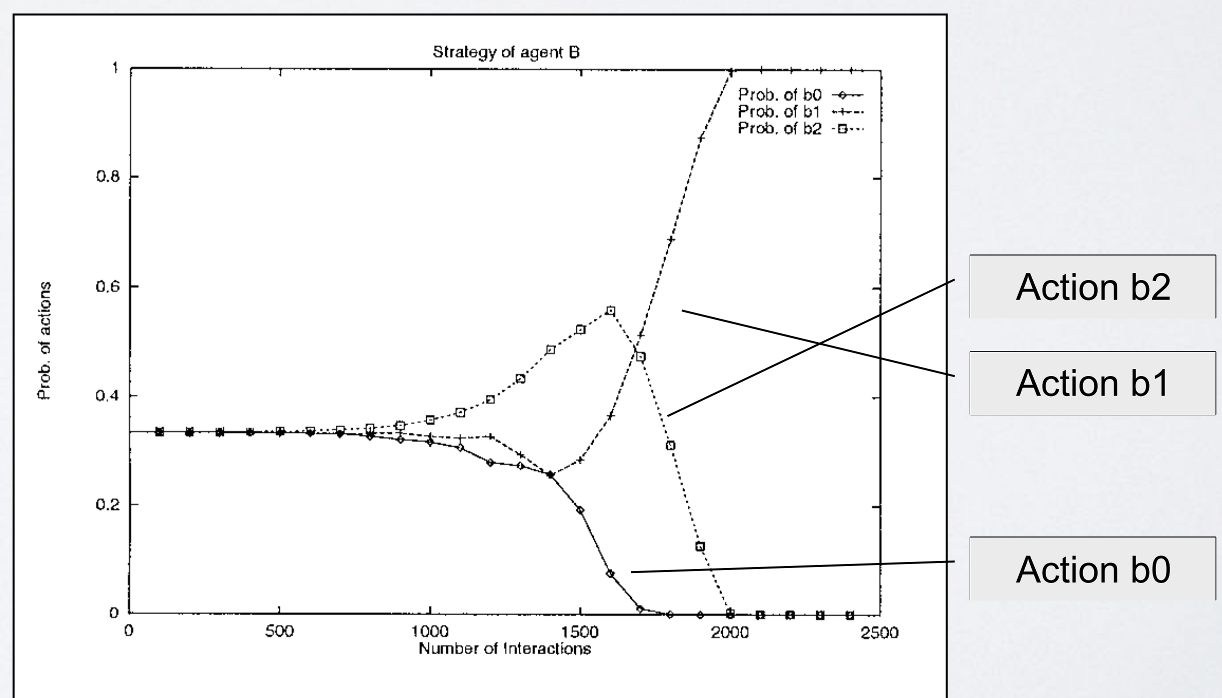
y-axis = prob. of the second player to play Bach

CLIMBING GAME

	a_0	a_1	a_2
b_0	11	-30	0
b_1	-30	7	6
b_2	0	0	5

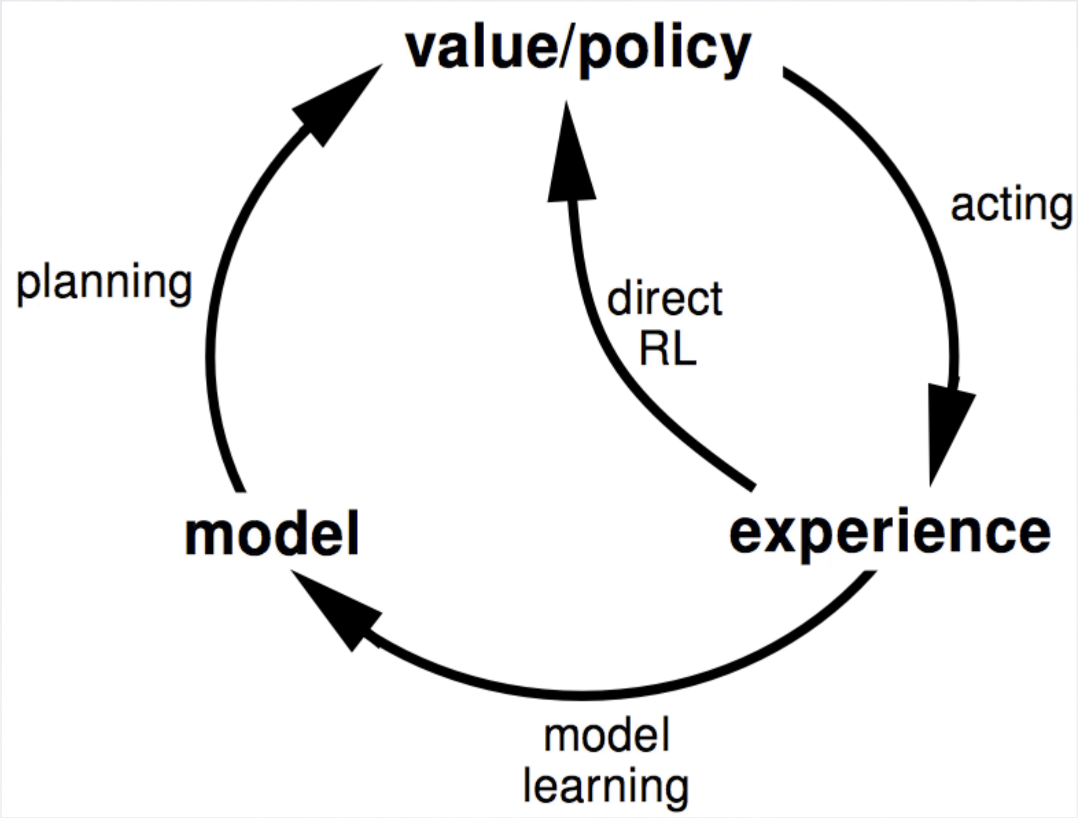
2 Nash Equilibria , 1 optimal



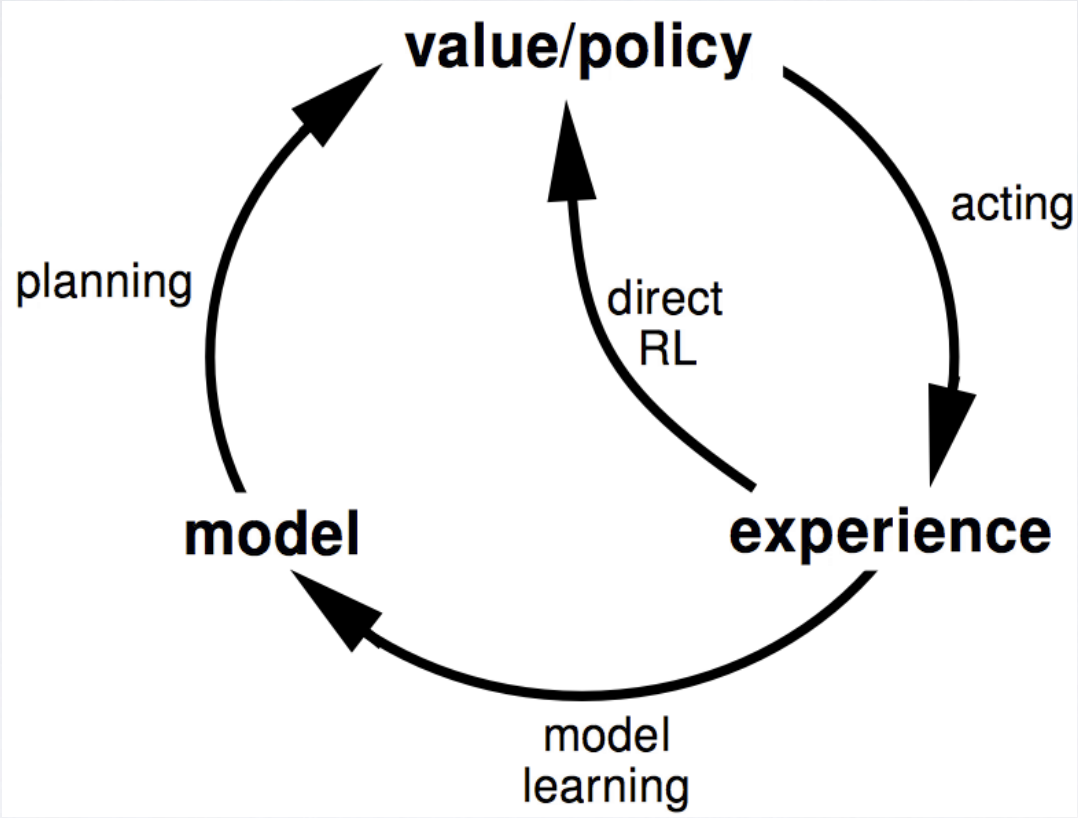


initial temperature 10000 is decayed at rate 0.995

WHERE (MULTI-AGENT) REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET

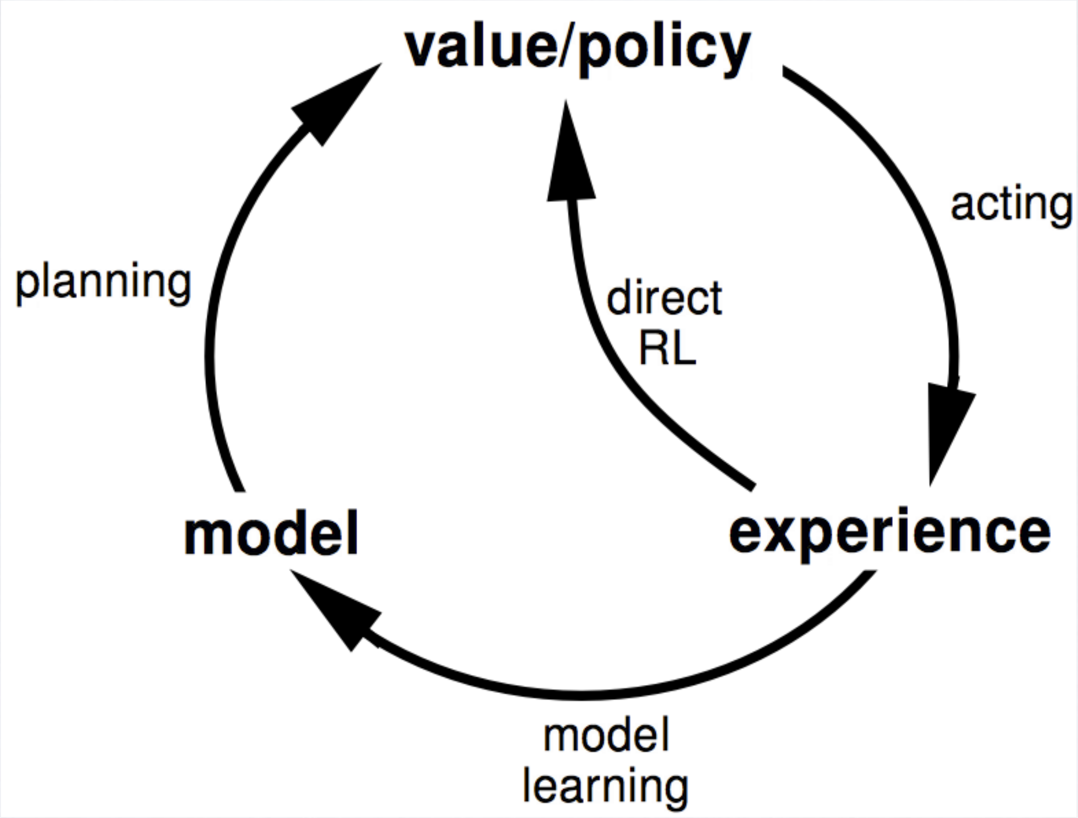


WHERE (MULTI-AGENT) REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET



Mainly for sample efficiency

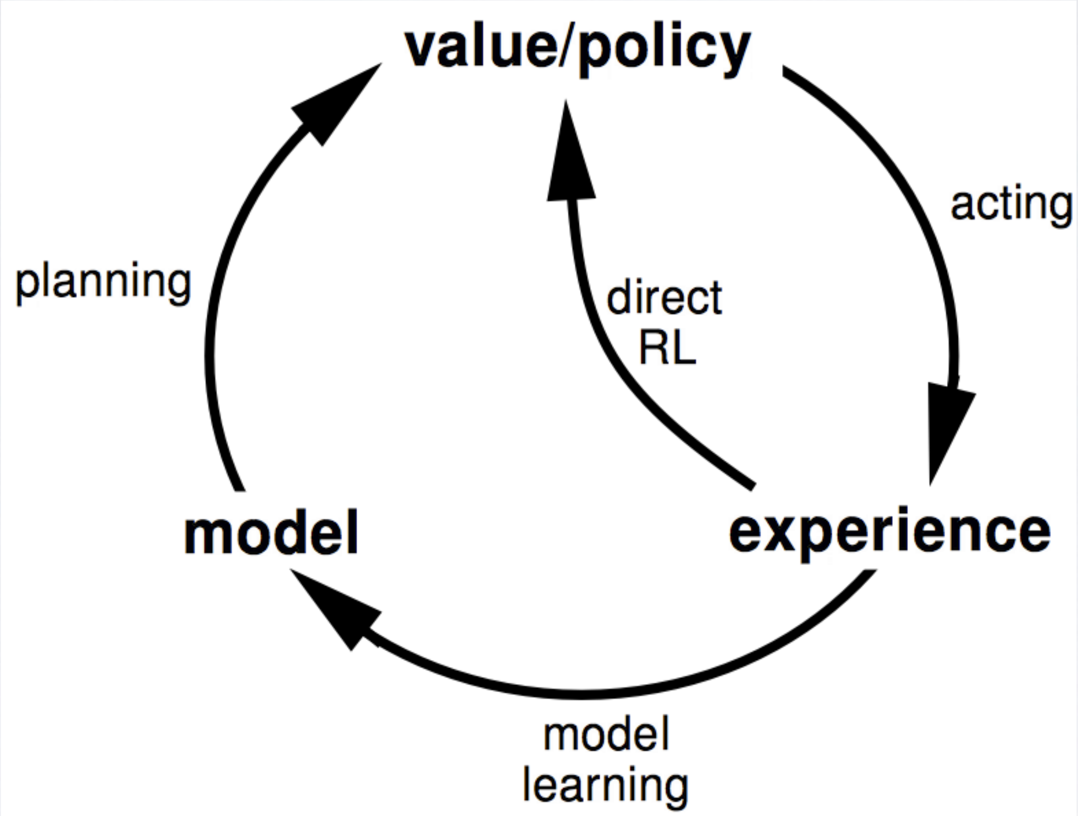
WHERE (MULTI-AGENT) REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET



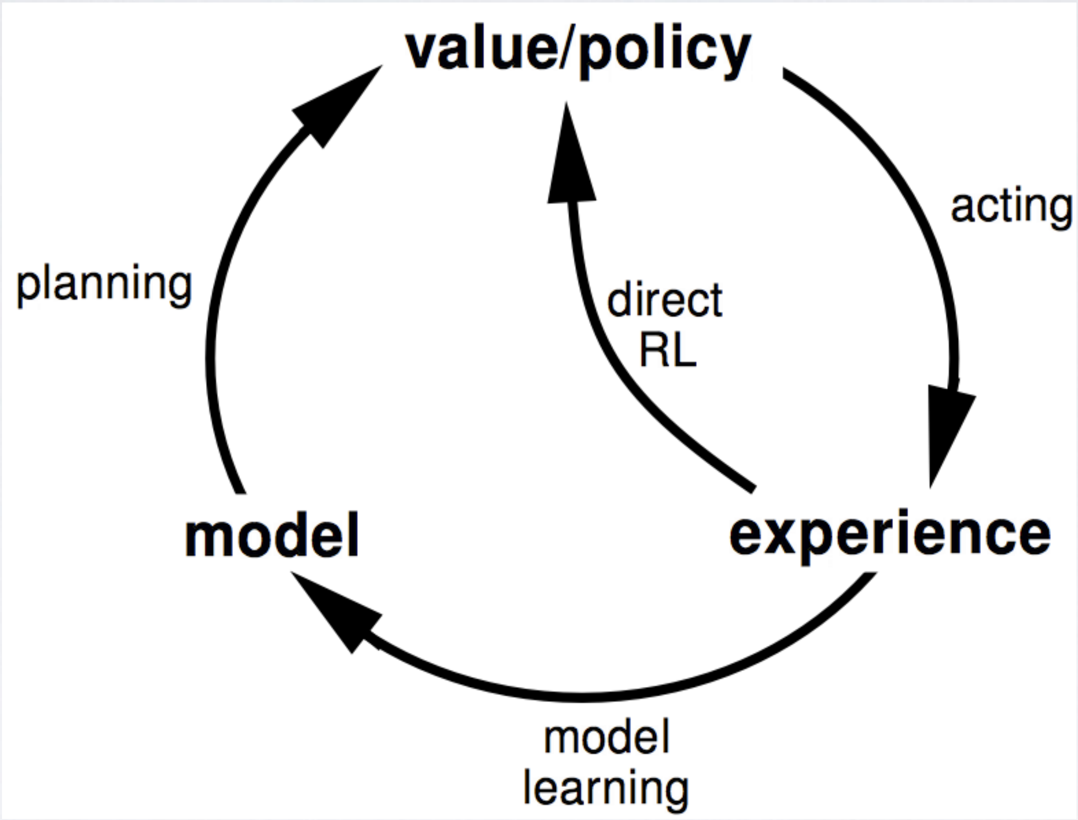
Mainly for sample efficiency

Model can be anything

WHERE MULTI-AGENT REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET



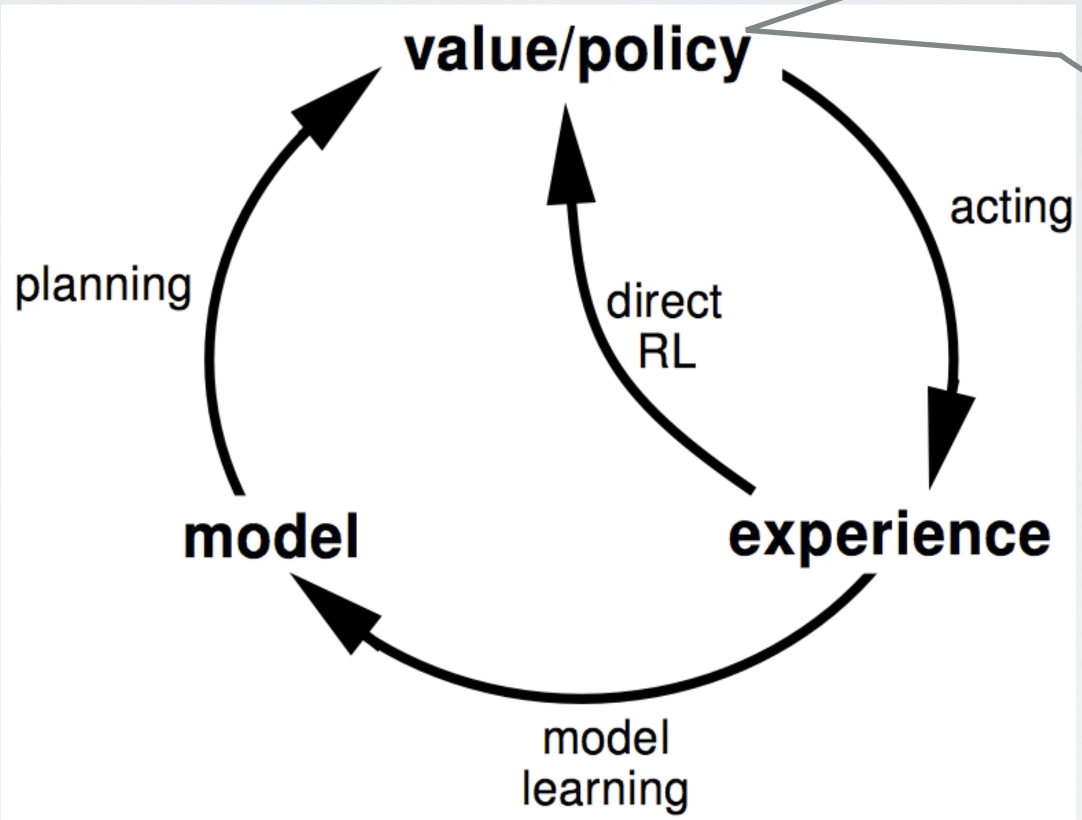
WHERE MULTI-AGENT REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET



Is assumed to be given

WHERE MULTI-AGENT REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET

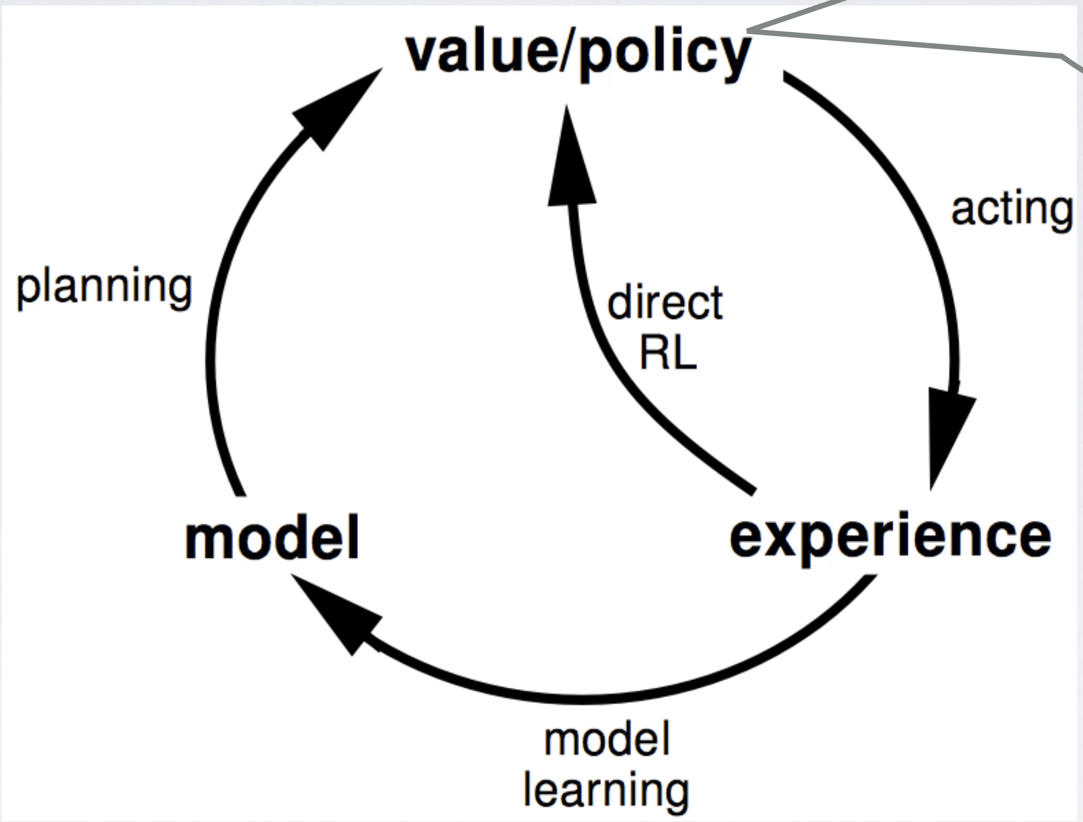
Is specified in a more generic way
And checked



Is assumed to be given

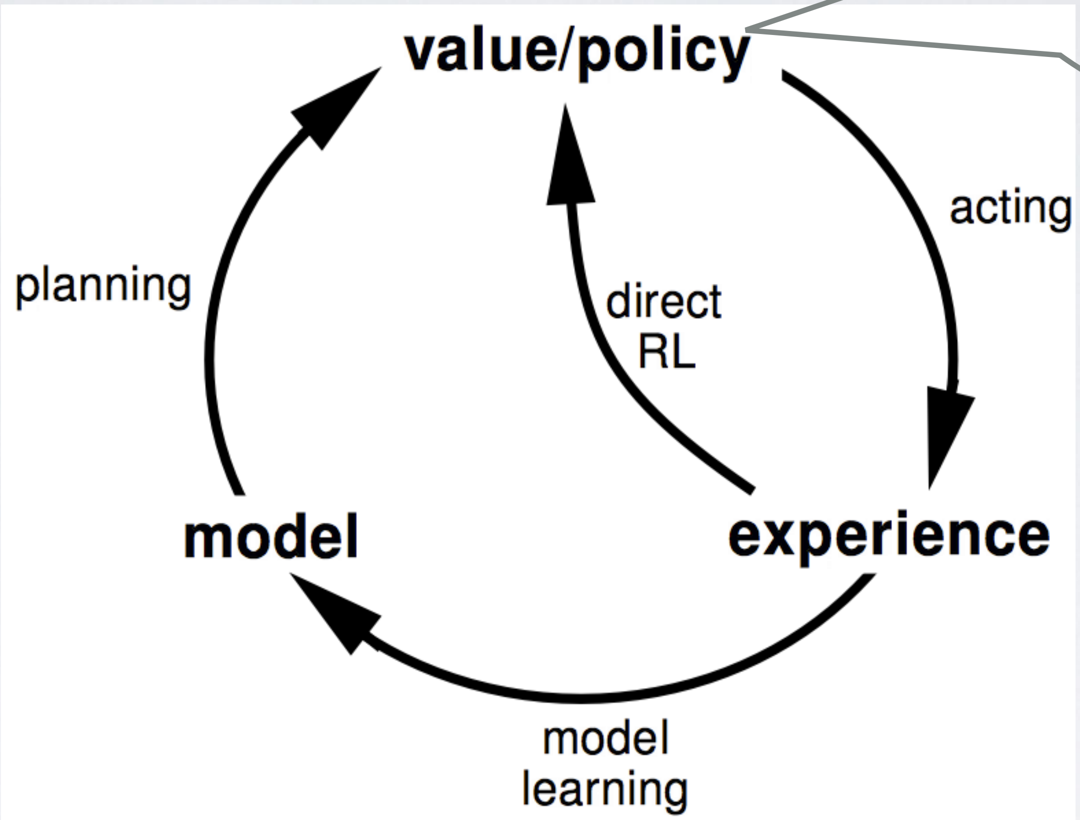
WHERE MULTI-AGENT REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET

Is specified in a more generic way
And checked



Is assumed to be given

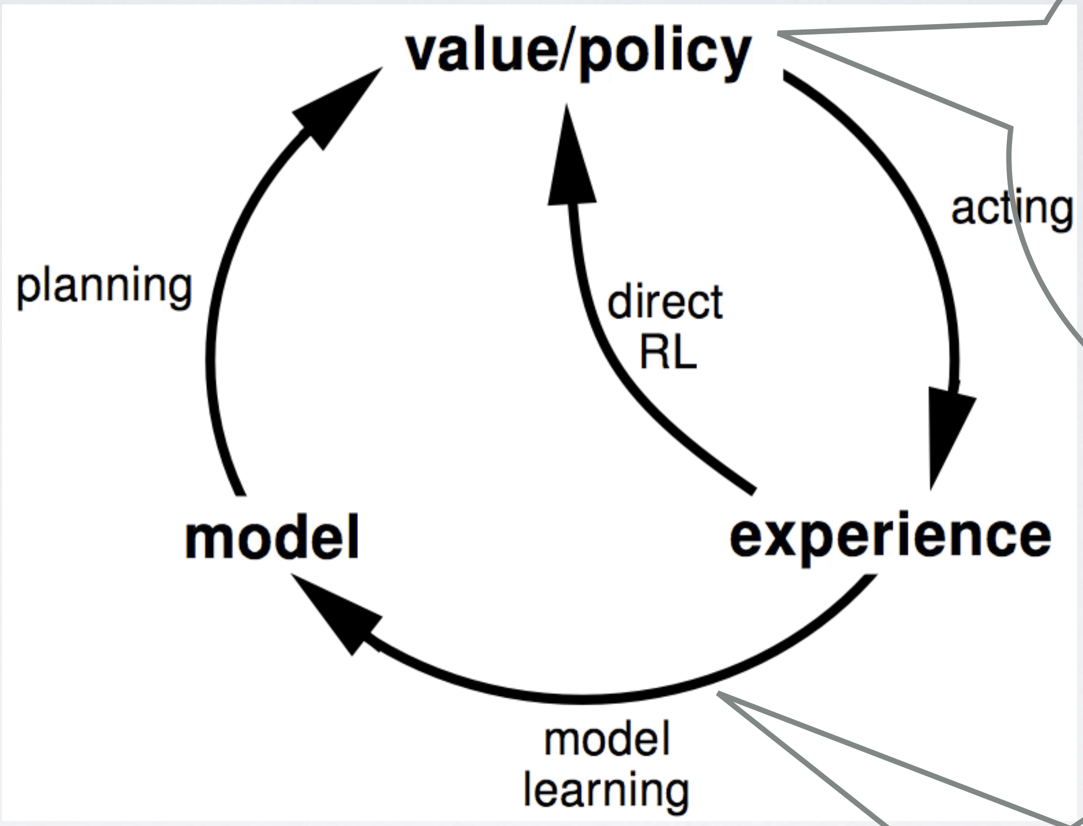
WHERE MULTI-AGENT REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET



Is specified in a more generic way
And checked

Is assumed to be given

WHERE (MULTI-AGENT) REINFORCEMENT LEARNING AND VERIFICATION **MIGHT** MEET

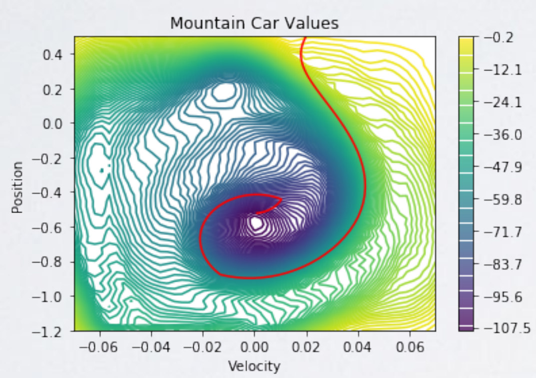


Can be synthesised in to a more abstract or symbolic representation that allows generalisation

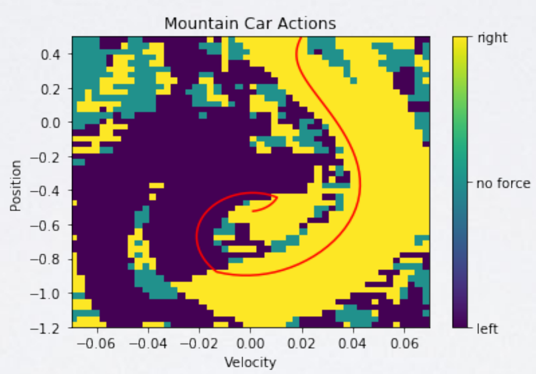
Model with right level of abstraction to allow verification of the Model wrt the real world and the synthesised/ generalised plan

POLICY ABSTRACTION

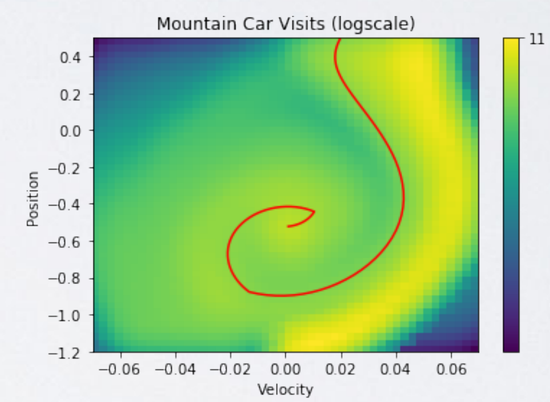
A flavour of how to use meta-information



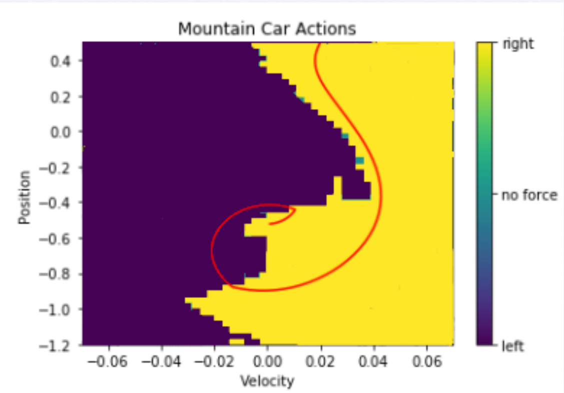
(a) V-function



(b) Greedy Policy



(c) # of state visits



(b) Greedy Policy