Reinforcement learning

Gergő Orbán

Recap: Learning frameworks

Recap: Learning frameworks

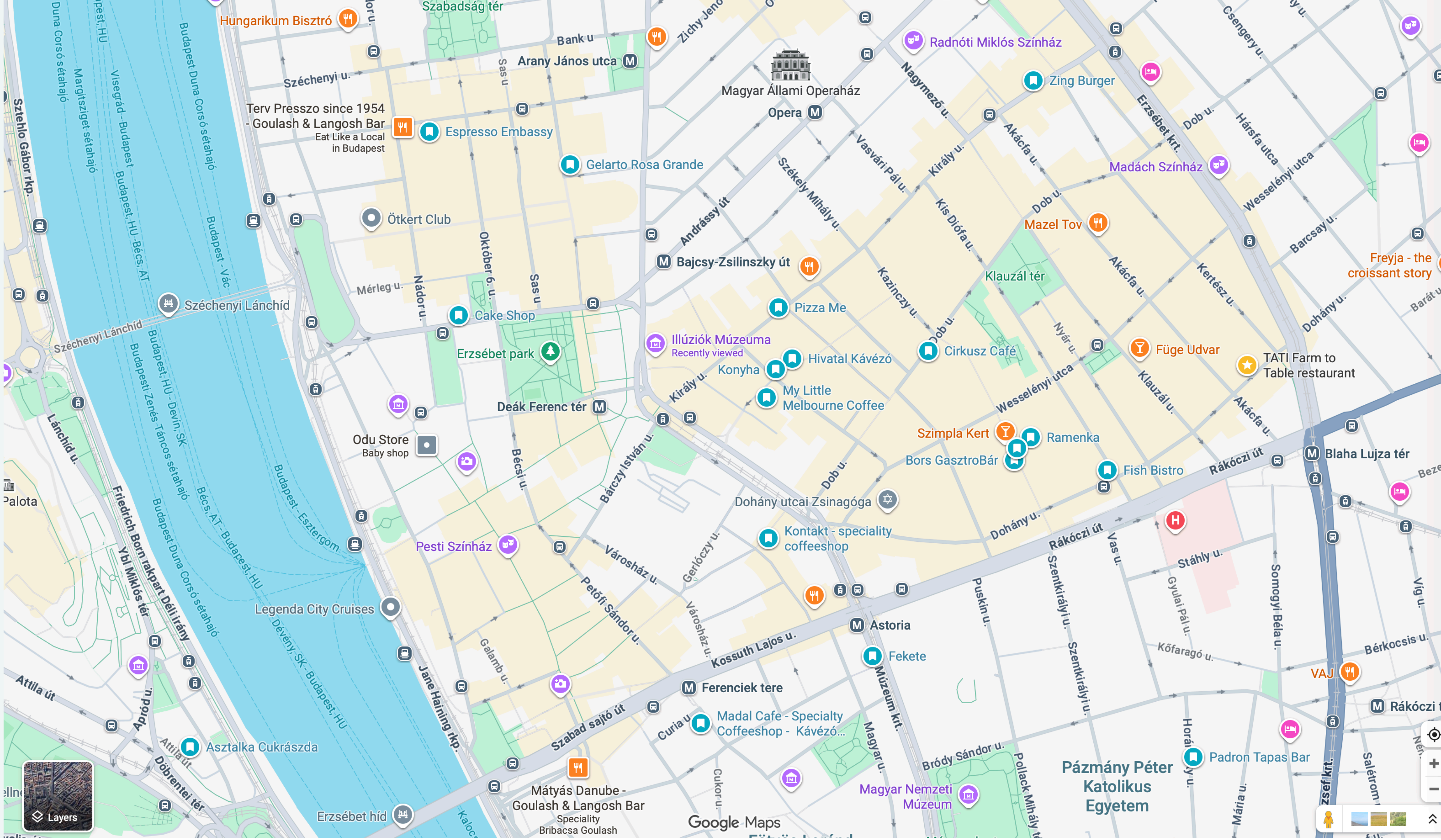
- Unsupervised

Recap: Learning frameworks

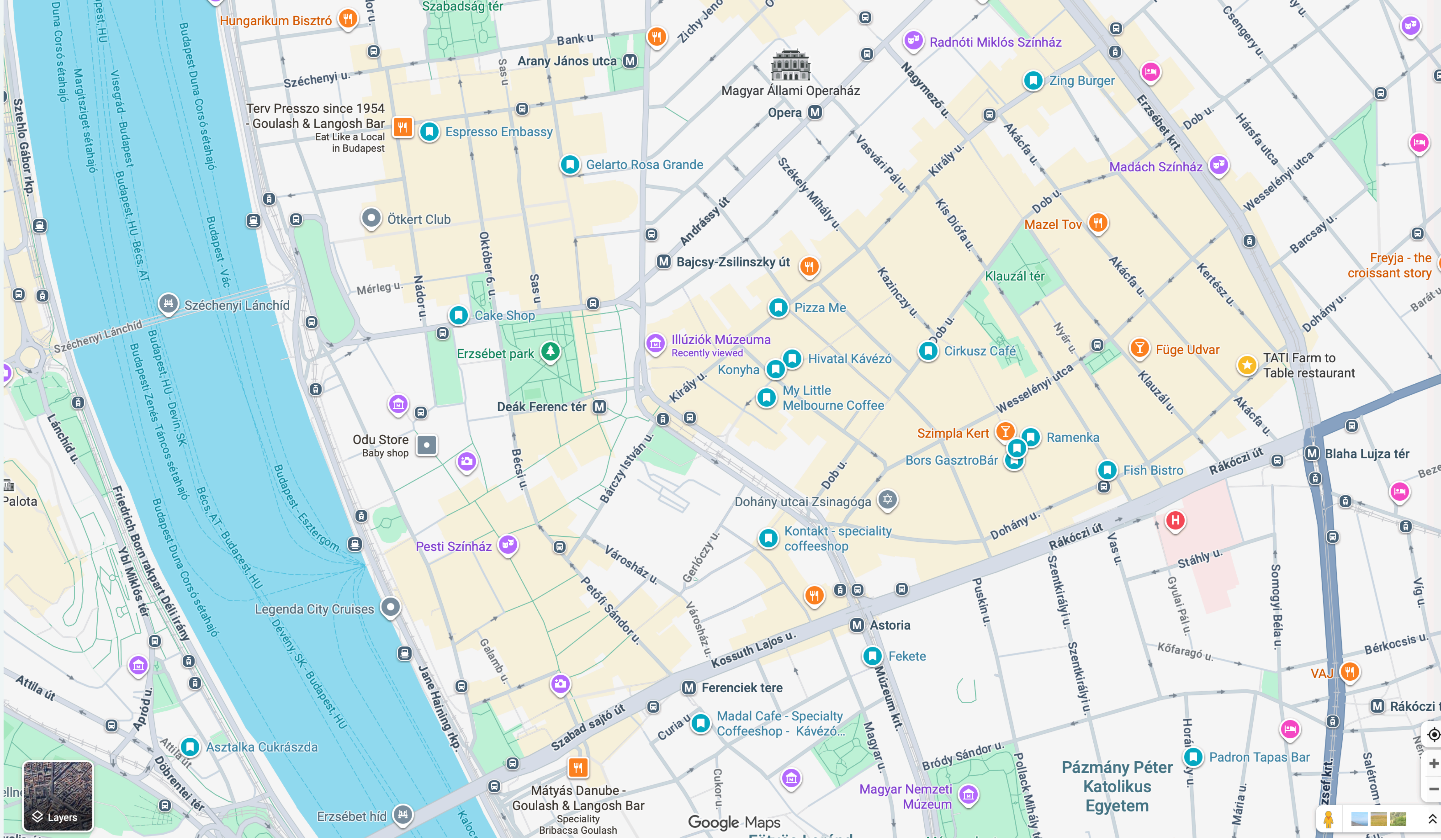
- Unsupervised
- Supervised learning: y=f(x) — essentially a mapping from input to output
  -> task specific
  -> requires labelled data points
  -> essentially optimization
  -> back-propagation could be easily obtained

Recap: Learning frameworks

- Unsupervised
- Supervised learning: y=f(x) — essentially a mapping from input to output
  -> task specific
  -> requires labelled data points
  -> essentially optimization
  -> back-propagation could be easily obtained
- Reinforcement learning
  -> phrasing learning as collection of reward
  -> sparser learning signal
  -> max E [Reward(input, action)]



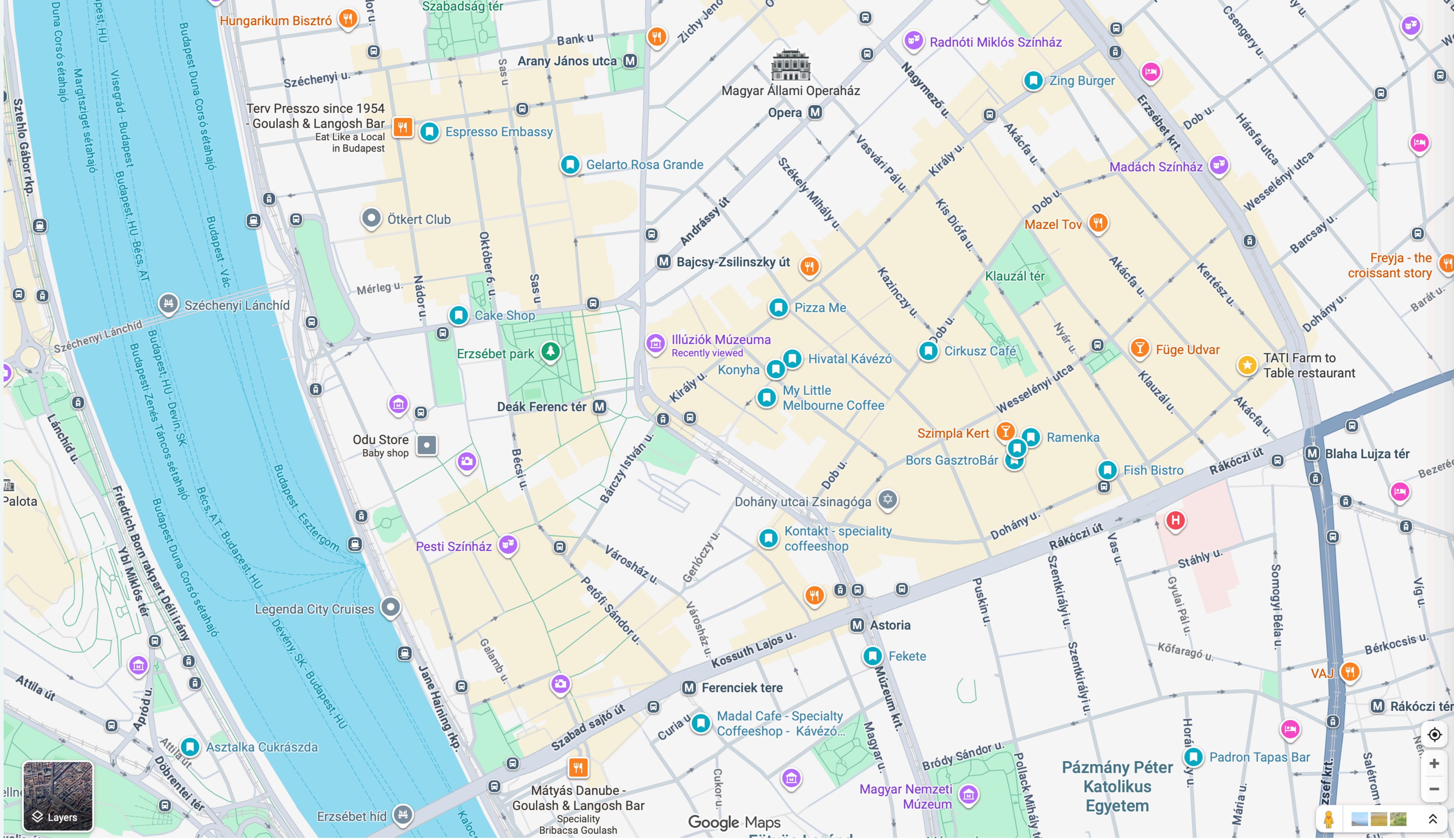
Hungarikum Bisztró
Szabadság tér
Bank u.
Arany János utca
Széchenyi u.
Magyar Állami Operaház
Opera
Radnóti Miklós Színház
Zing Burger
Terv Presszo since 1954 - Goulash & Langosh Bar
Eat Like a Local in Budapest
Espresso Embassy
Gelarto Rosa Grande
Madách Színház
Mazel Tov
Ötkert Club
Bajcsy-Zsilinszky út
Freyja - the croissant story
Klauzál tér
Mérleg u.
Széchenyi Lánchíd
Cake Shop
Pizza Me
Illúziók Múzeuma
Recently viewed
Hivatal Kávézó
Cirkusz Café
Füge Udvar
TATI Farm to Table restaurant
Erzsébet park
Konyha
My Little Melbourne Coffee
Deák Ferenc tér
Odu Store
Baby shop
Szimpla Kert
Ramenka
Bors GasztroBár
Fish Bistro
Blaha Lujza tér
Dohány utcai Zsinagóga
Kontakt - speciality coffeeshop
Pesti Színház
Legenda City Cruises
Astoria
Fekete
Ferenciek tere
Madal Cafe - Specialty Coffeeshop - Kávézó...
Asztalka Cukrászda
Mátyás Danube - Goulash & Langosh Bar
Speciality Bribacsa Goulash
Erzsébet híd
Magyar Nemzeti Múzeum
Pázmány Péter Katolikus Egyetem
Padron Tapas Bar
VAJ
Rákóczi tér
Rákóczi út
Kossuth Lajos u.
Szabad sajtó út
Wesselényi utca
Király u.
Dohány u.
Andrássy út
Layers
Google Maps



Dual challenge

reward is not immediate:

most of the actions are not rewarding by themselves, as rewards are distal



Dual challenge

reward is not immediate:

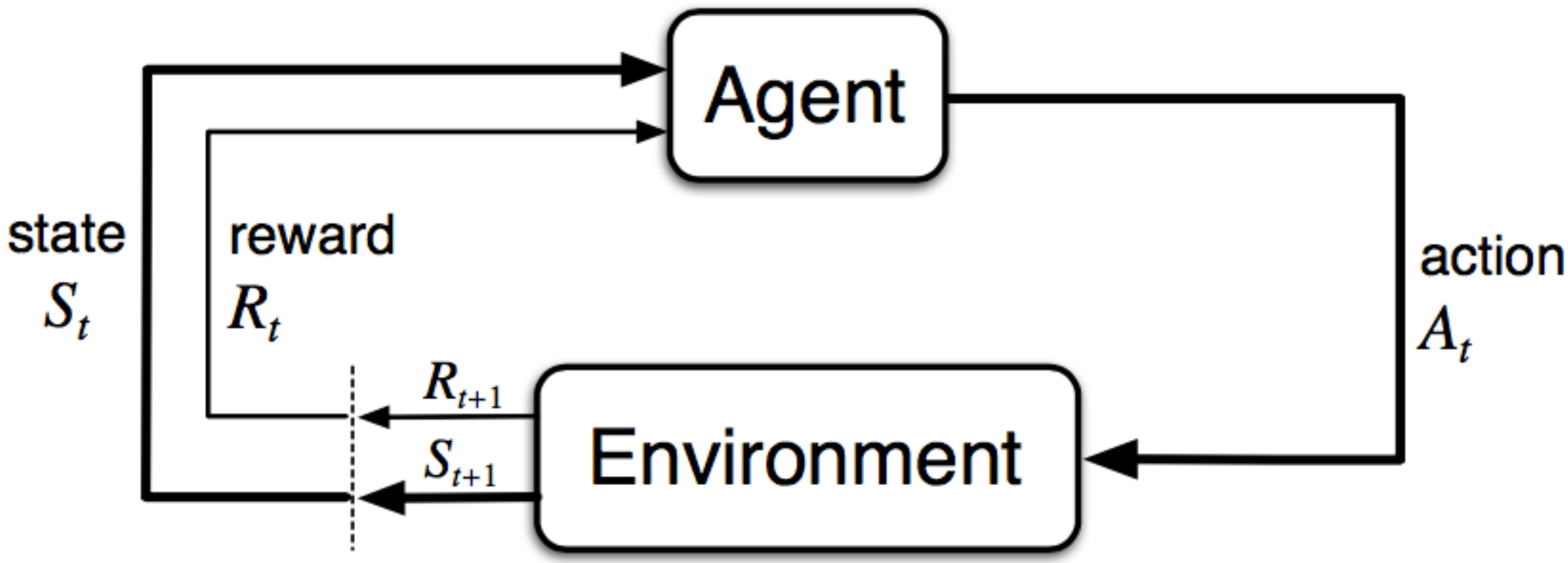
most of the actions are not rewarding by themselves, as rewards are distal

options are not readily available:

rewards are not known *and* one experience does not tell exactly how rewarding a state is

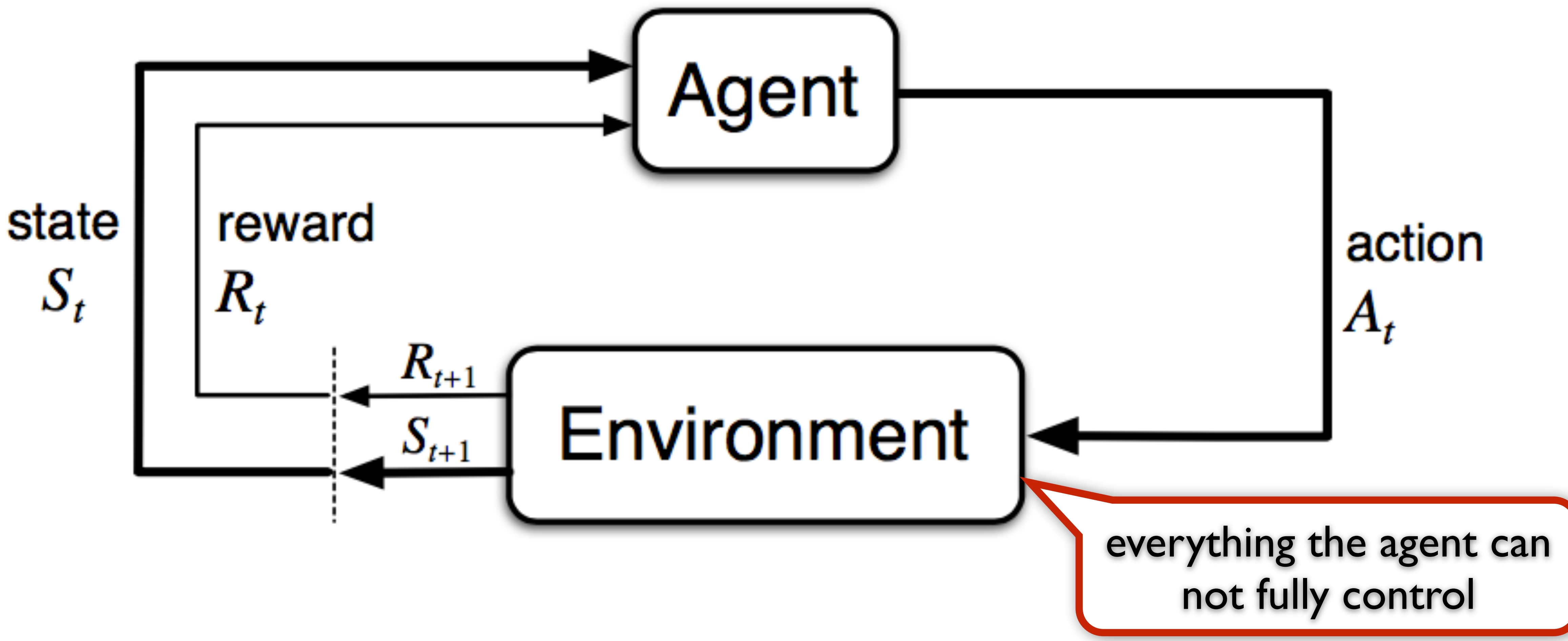
Reinforcement learning

Reinforcement learning



Reinforcement learning

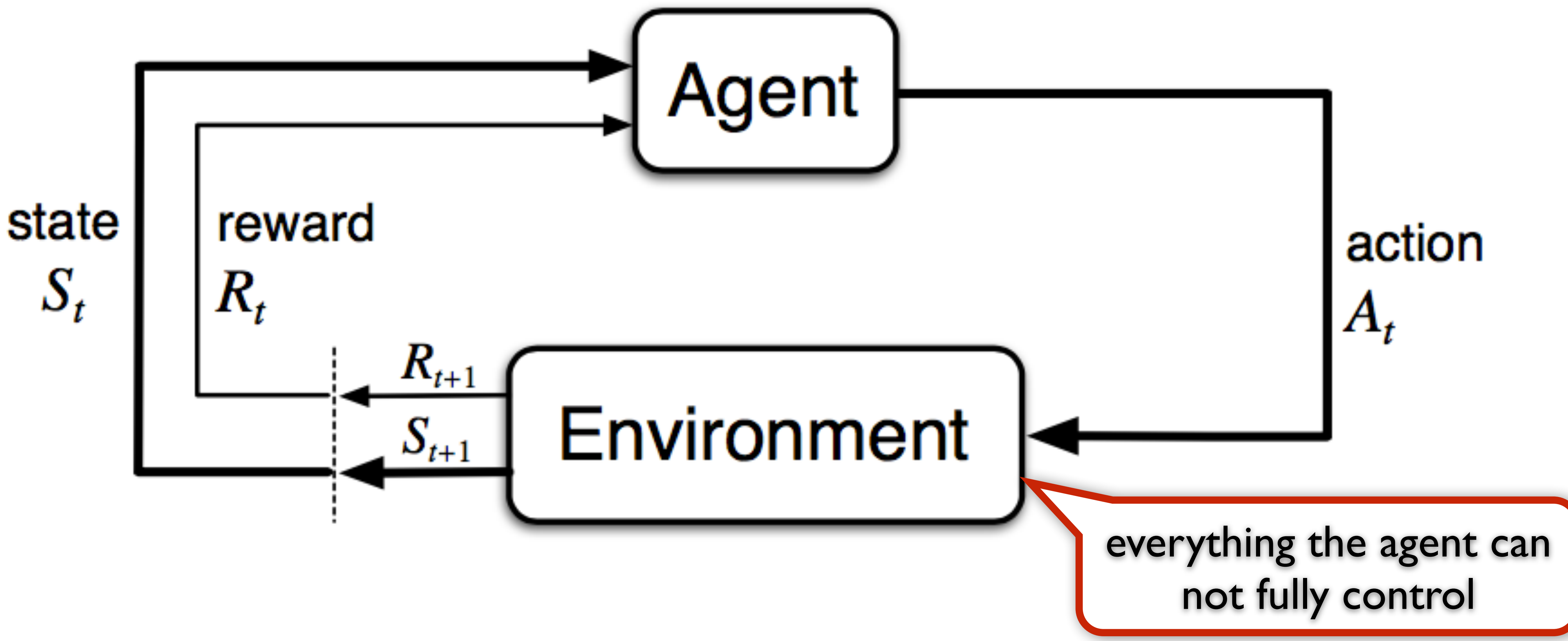
State



Reinforcement learning

State

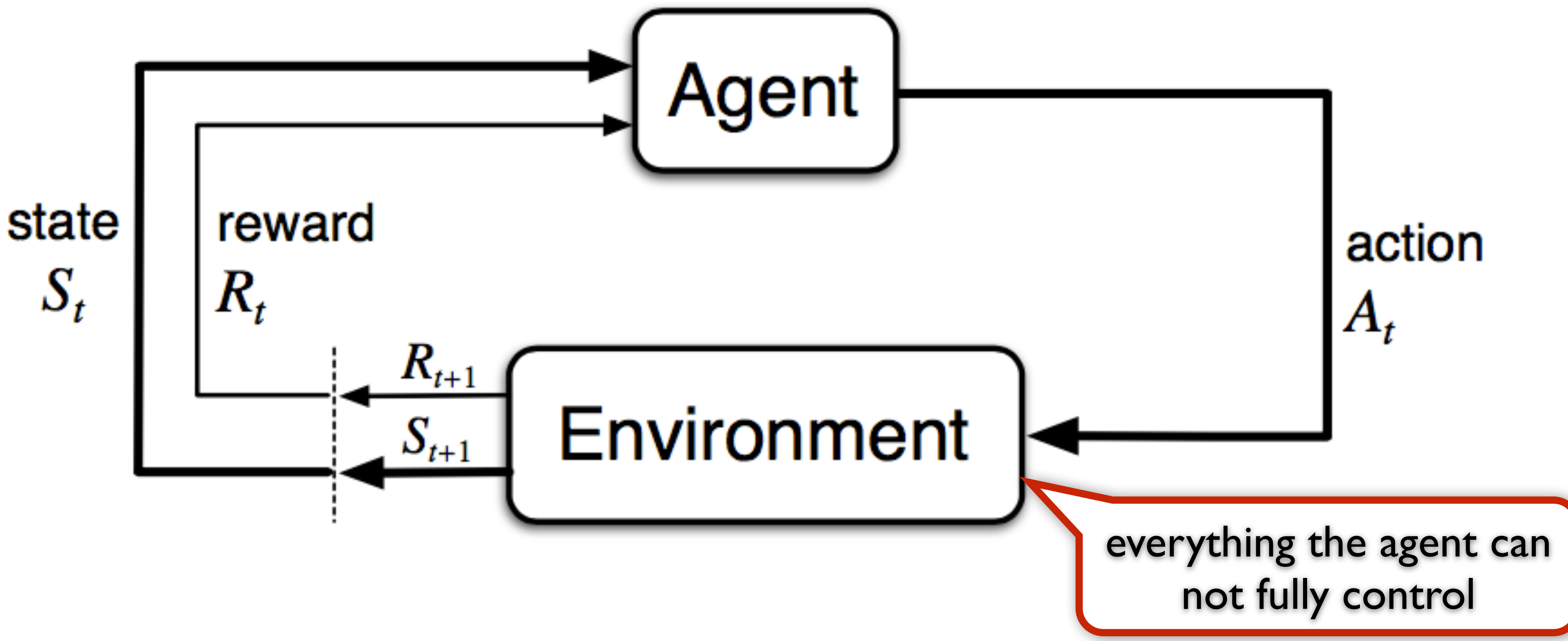
- state of the environment



Reinforcement learning

State

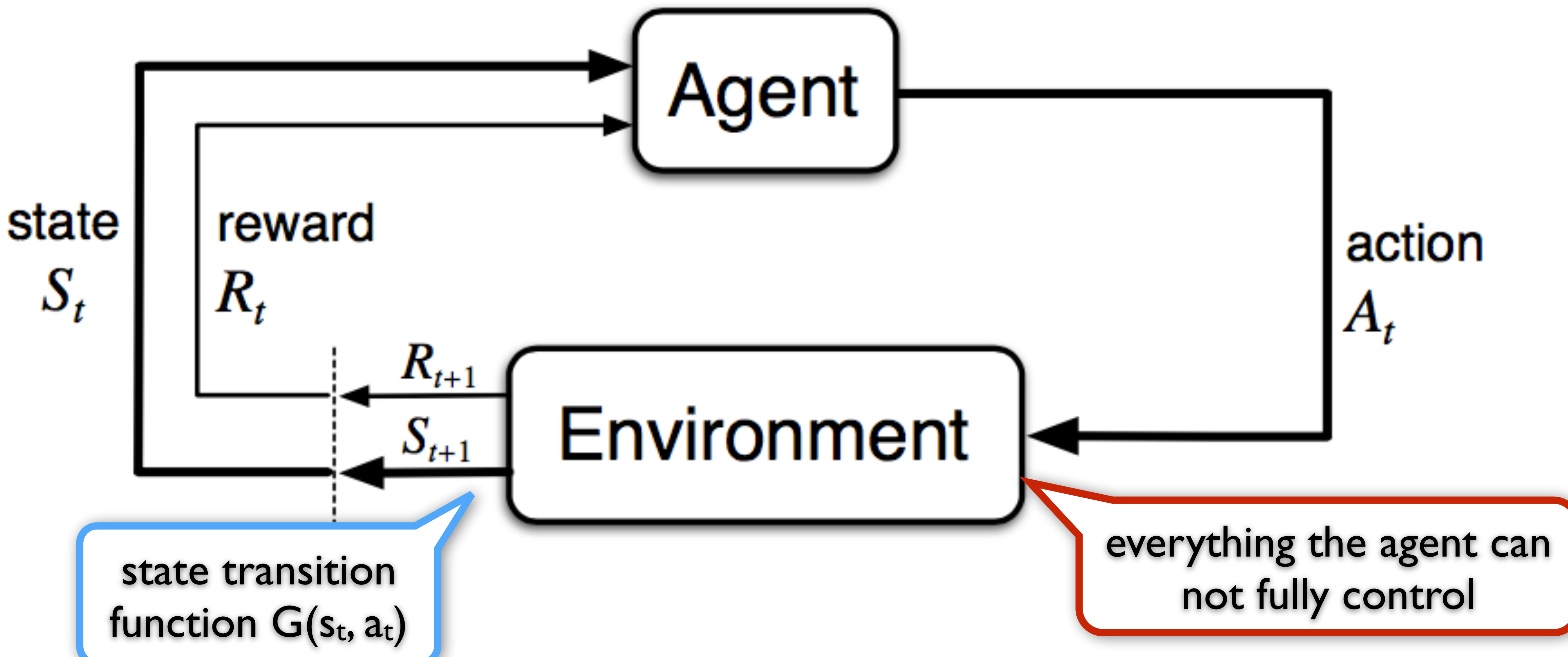
- state of the environment
- partially observed



Reinforcement learning

State

- state of the environment
- partially observed
- Markov property

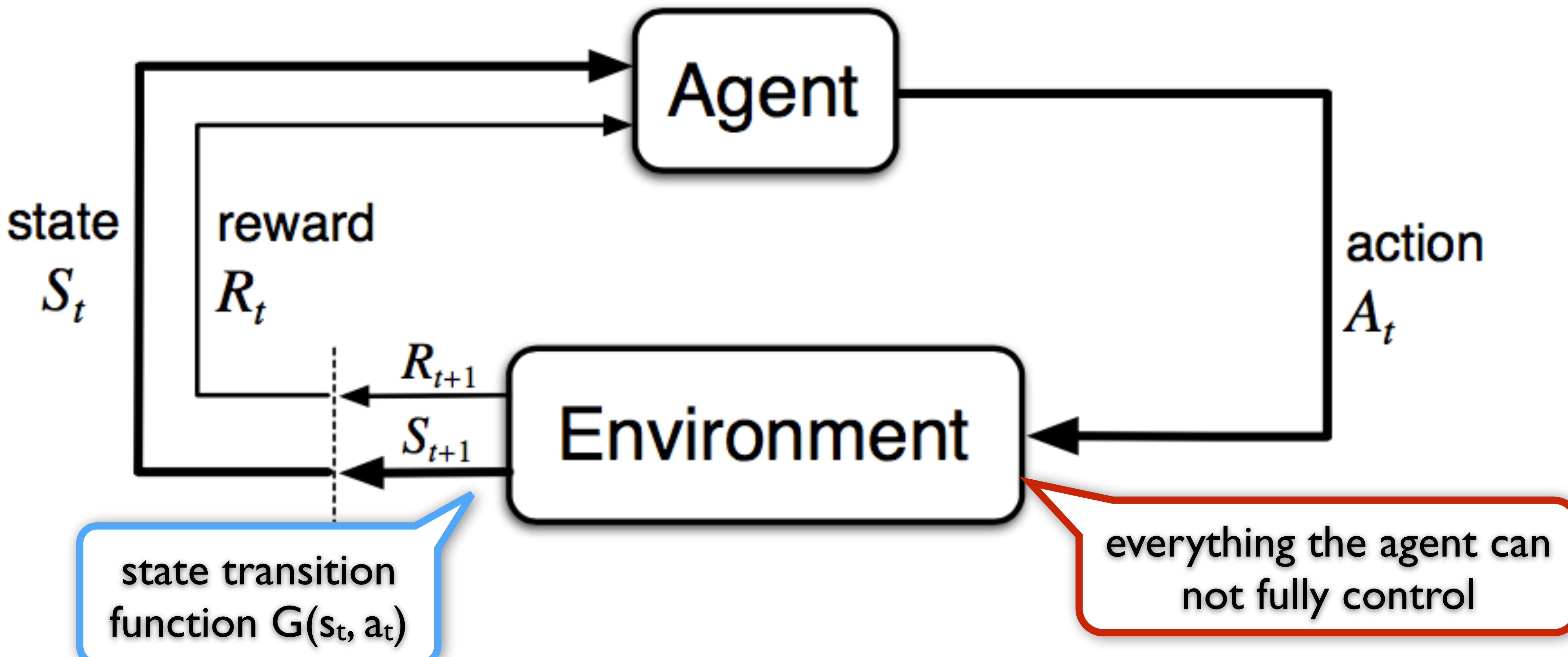


Reinforcement learning

State

- state of the environment
- partially observed
- Markov property

Action



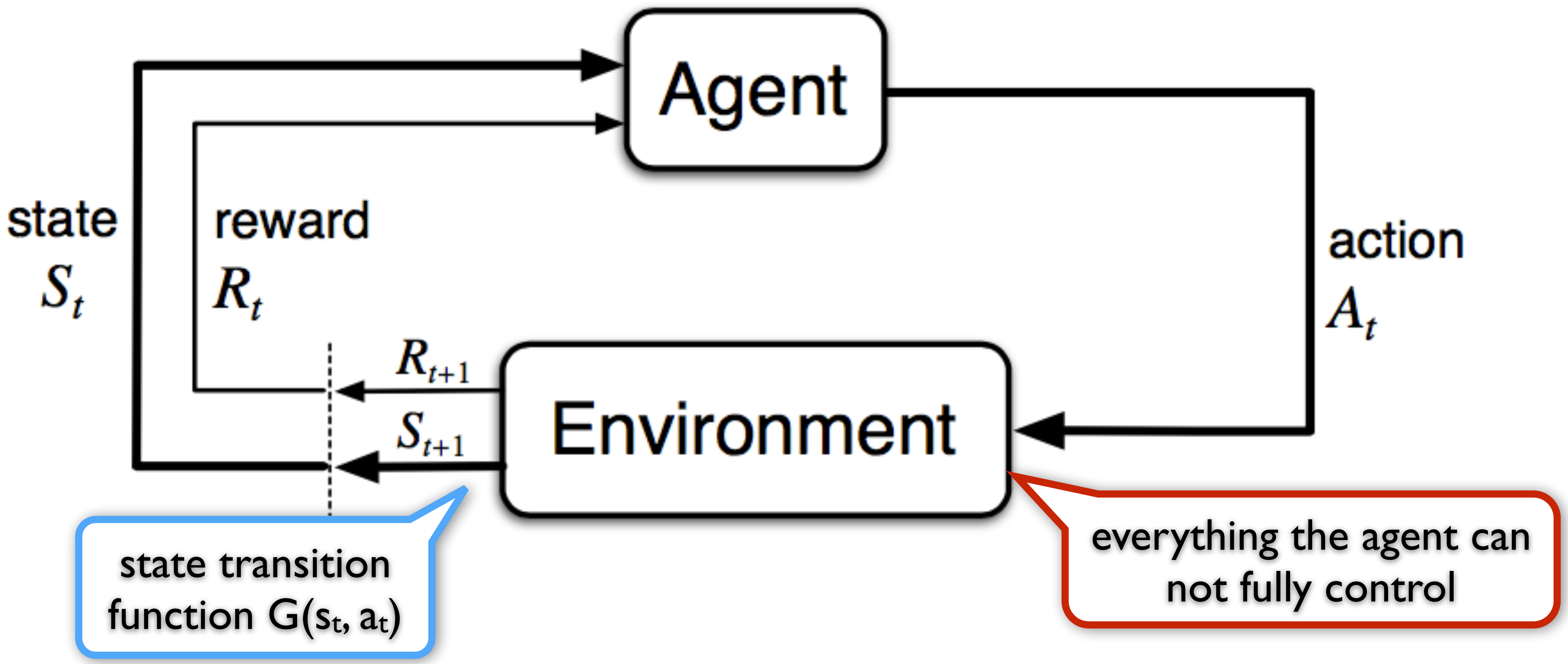
Reinforcement learning

State

- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions



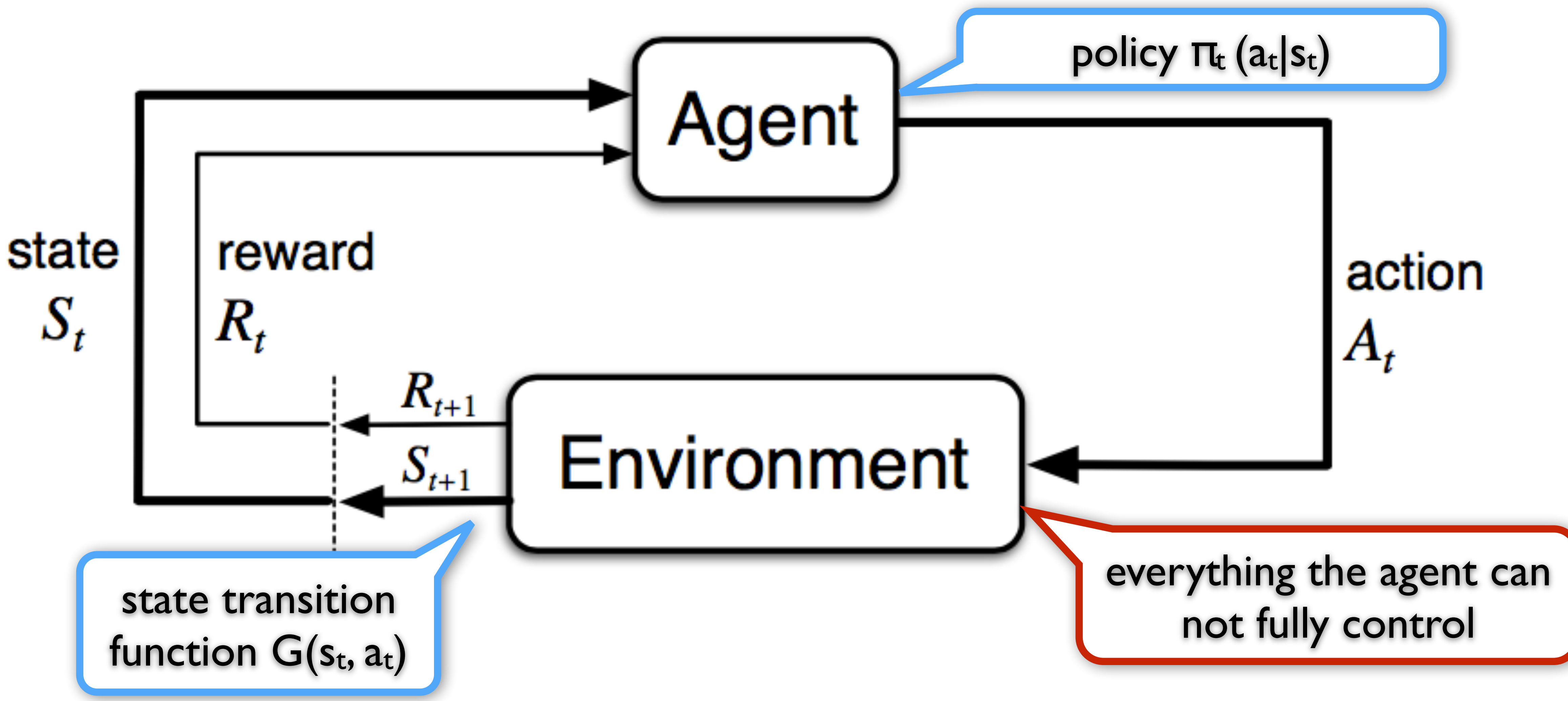
Reinforcement learning

State

- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions



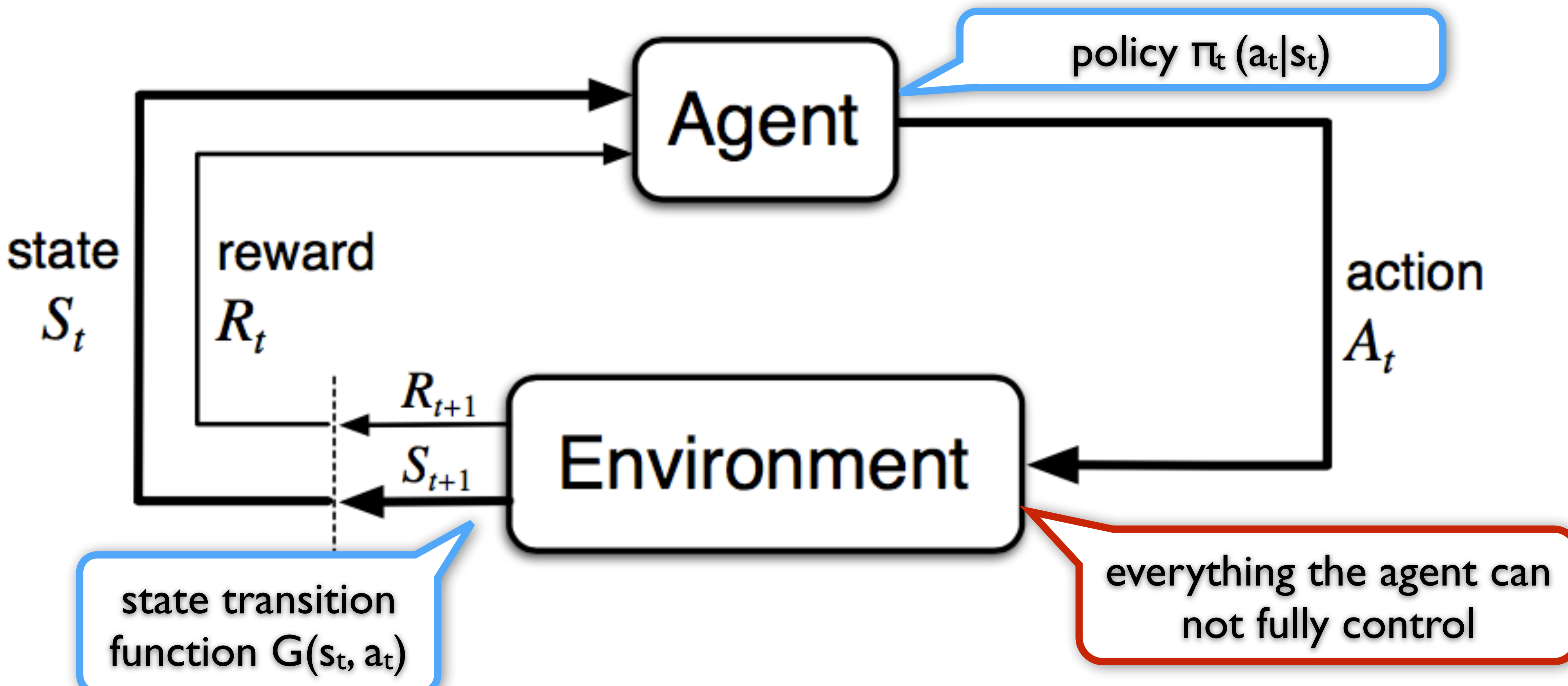
Reinforcement learning

State

- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions
- actions may change the state and/or lead to reward



Reinforcement learning

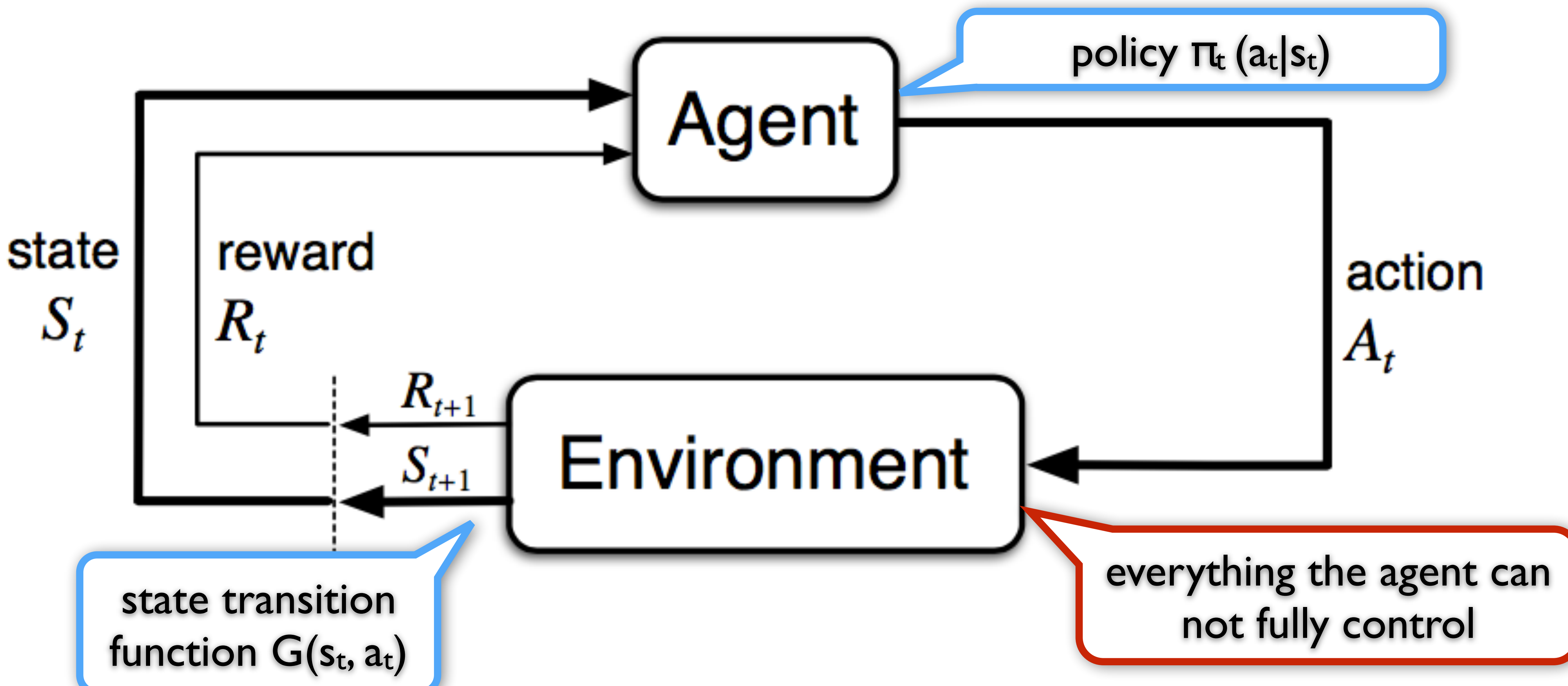
State

- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions
- actions may change the state and/or lead to reward

Reward



Reinforcement learning

State

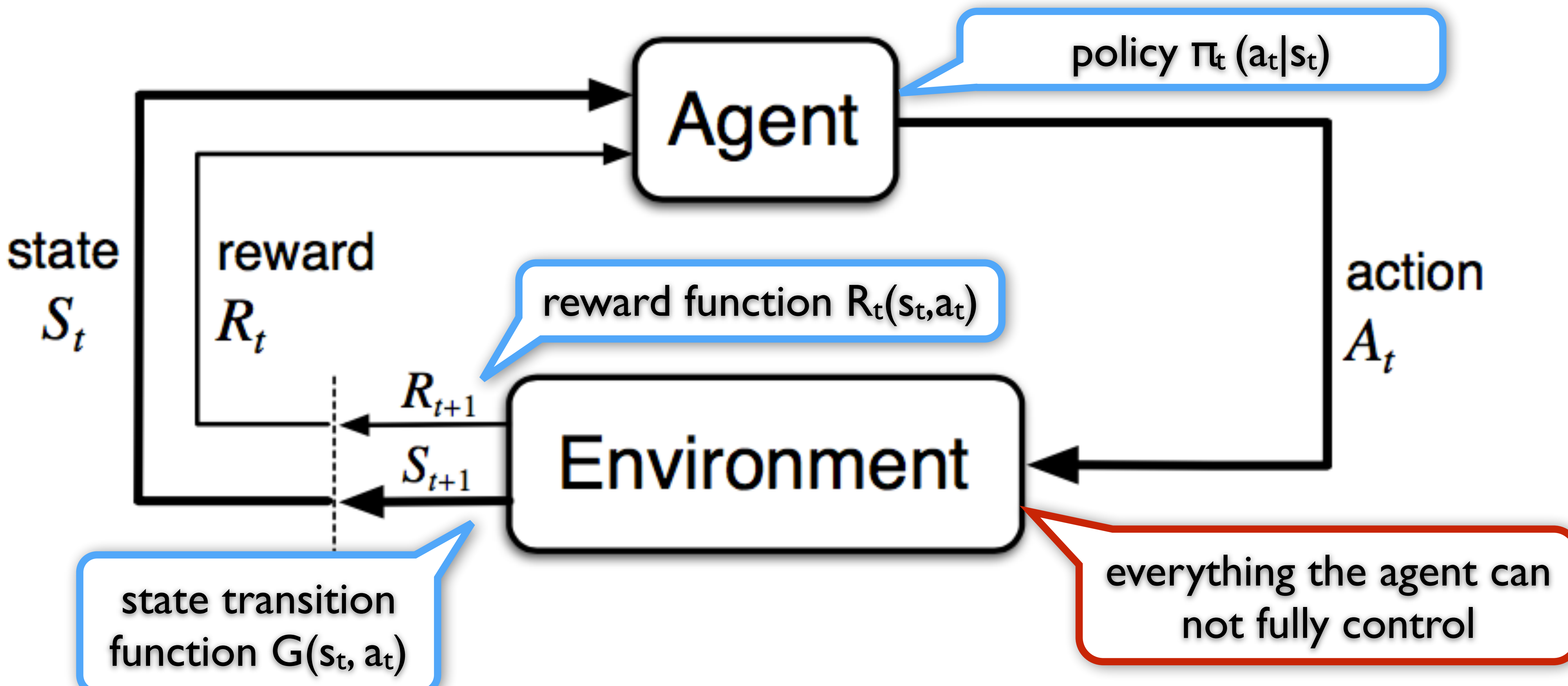
- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions
- actions may change the state and/or lead to reward

Reward

- $R_t(s,a)$ is the reward given at time t in state s and action a



Reinforcement learning

State

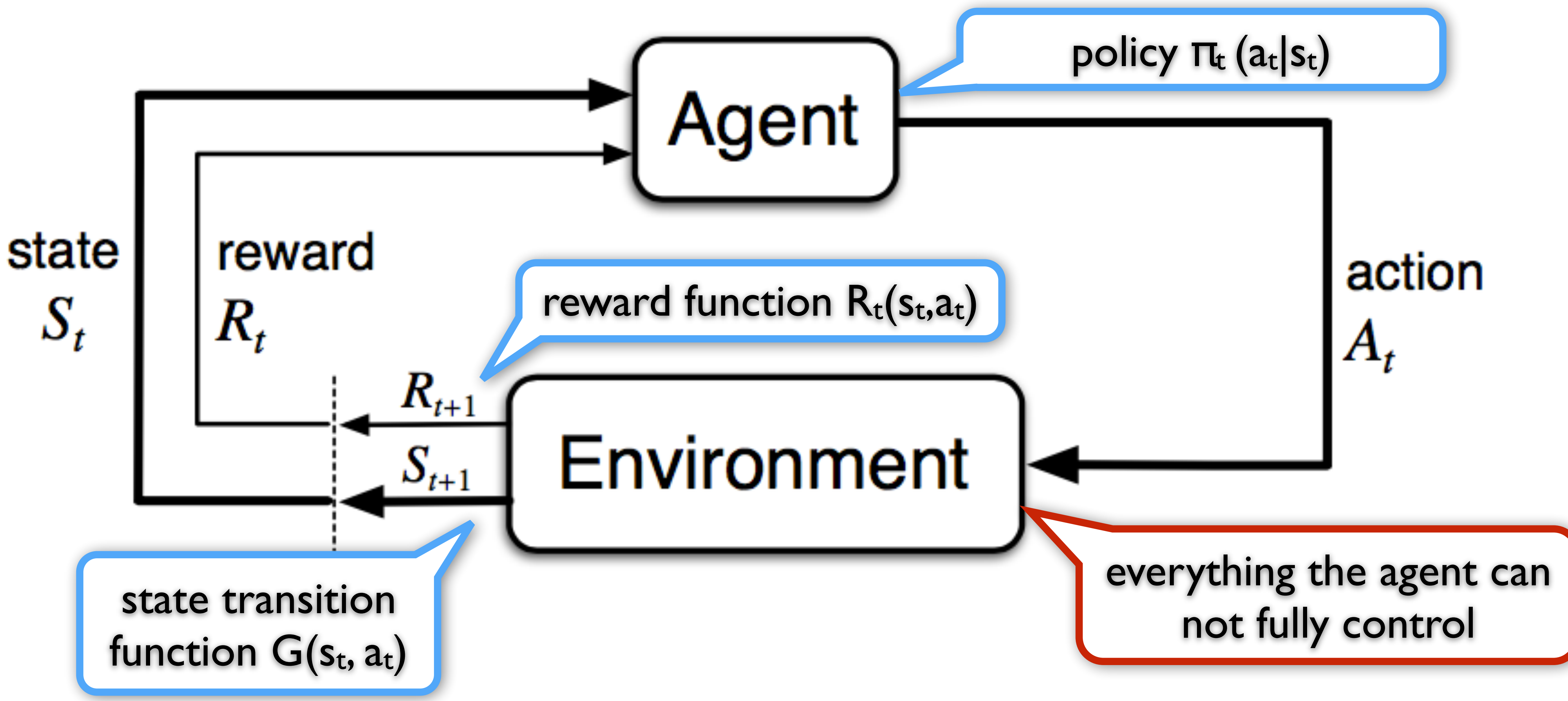
- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions
- actions may change the state and/or lead to reward

Reward

- $R_t(s,a)$ is the reward given at time t in state s and action a
- negative or positive



Reinforcement learning

State

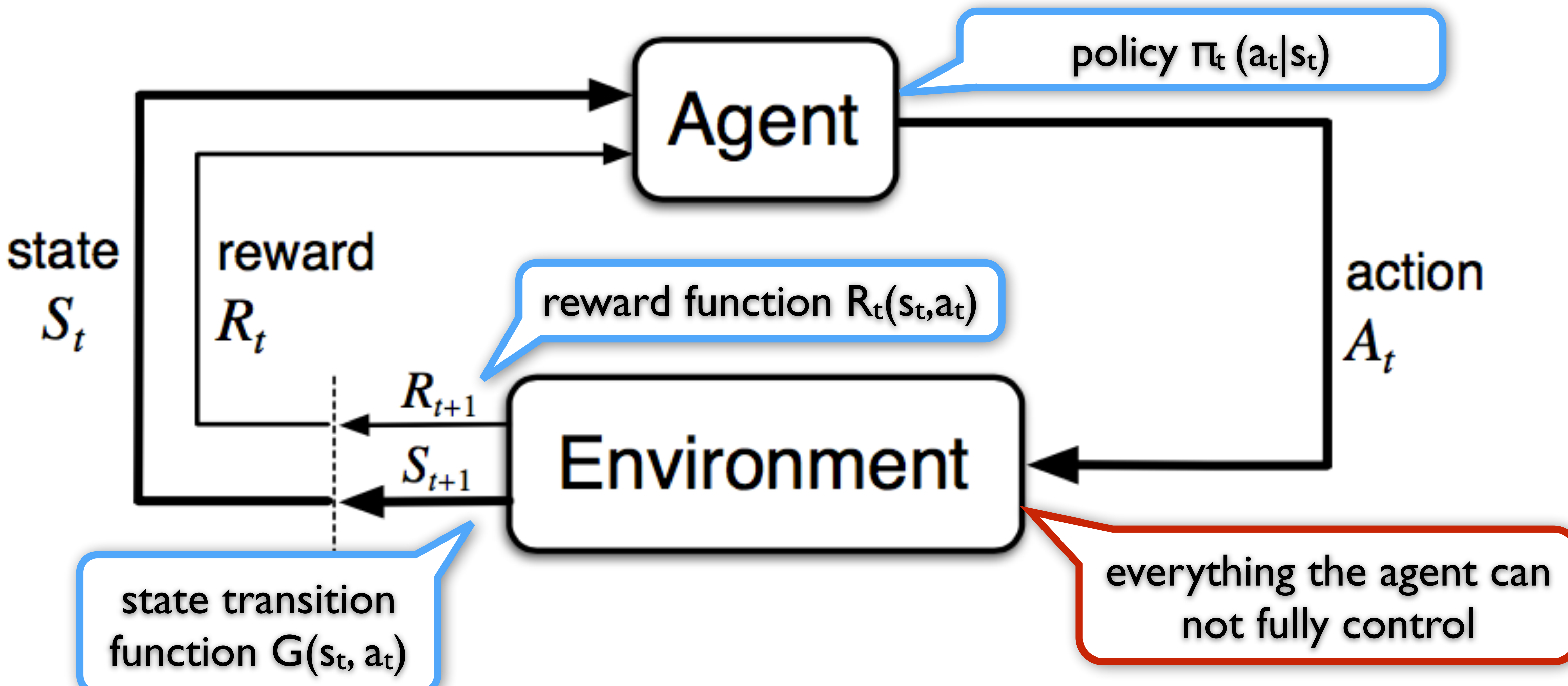
- state of the environment
- partially observed
- Markov property

Action

- agents interact with the environment with actions
- policy maps states to actions
- actions may change the state and/or lead to reward

Reward

- $R_t(s,a)$ is the reward given at time t in state s and action a
- negative or positive
- goal: maximise total reward



Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

more precisely: $V(s) = \mathbb{E}[\mathscr{R}_t \,|\, s]$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

more precisely: $V(s) = \mathbb{E}[\mathscr{R}_t \,|\, s]$

more more precisely: $V_\pi(s) = \mathbb{E}_\pi[\mathscr{R}_t \,|\, s]$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

more precisely: $V(s) = \mathbb{E}[\mathscr{R}_t \,|\, s]$

more more precisely: $V_\pi(s) = \mathbb{E}_\pi[\mathscr{R}_t \,|\, s]$

policy:
$\pi(a_t \,|\, s_t)$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

more precisely: $V(s) = \mathbb{E}[\mathscr{R}_t \,|\, s]$

more more precisely: $V_\pi(s) = \mathbb{E}_\pi[\mathscr{R}_t \,|\, s]$

policy:
$\pi(a_t \,|\, s_t)$

thus, make explicit that not all states are equivalent: $r_{t+1} = r(s_t, a_t, s_{t+1})$

Reinforcement learning

cumulative reward: $\mathscr{R} = \sum_k r_k$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k$

formal goal: $V = \mathbb{E}[\mathscr{R}]$

more precisely: $V(s) = \mathbb{E}[\mathscr{R}_t \,|\, s]$

more more precisely: $V_\pi(s) = \mathbb{E}_\pi[\mathscr{R}_t \,|\, s]$

policy:
$\pi(a_t \,|\, s_t)$

thus, make explicit that not all states are equivalent: $r_{t+1} = r(s_t, a_t, s_{t+1})$

along with how the world works — $p(s_{t+1} \,|\, s_t, a_t)$ — we have all the ingredients

Simple model environment: gridworld

Simple model environment: gridworld

what is the value associated with a given state under a policy?

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \,|\, S_t = s]$$

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \,|\, S_t = s]$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right]$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right]
\end{aligned}
$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \middle| S_t = s'\right]\right]
\end{aligned}$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \,|\, S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s) \left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s) \left[r(s,a,s') + \gamma V_\pi(s')\right]
\end{aligned}
$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

what is the value associated with a given state under a policy?
Bellmann equation:

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \,|\, S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s) \left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s) \left[r(s,a,s') + \gamma V_\pi(s')\right]
\end{aligned}
$$

Slight generalization: Bellmann equation for Q:

$$
Q_\pi(s,a) =
$$
$$
= \sum_{s'} P(s'|a,s) \left[r(s,a,s') + \sum_a \pi(a|s')\, \gamma\, Q_\pi(s',a)\right]
$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Challenges in RL

Challenges in RL

Why is it hard?

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!
- state dynamics is unknown

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!
- state dynamics is unknown
- state dynamics can be stochastic - noisy environment or noisy action

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!
- state dynamics is unknown
- state dynamics can be stochastic - noisy environment or noisy action
- rewards can be stochastic

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!
- state dynamics is unknown
- state dynamics can be stochastic - noisy environment or noisy action
- rewards can be stochastic
- states are only partially observed

Challenges in RL

Why is it hard?

- rewards are distal - credit assignment problem
- exploration - exploitation dilemma
- state-space is huge - searching takes a long time!
- state dynamics is unknown
- state dynamics can be stochastic - noisy environment or noisy action
- rewards can be stochastic
- states are only partially observed
- rules change with time

Challenge 1: uncertain outcomes

We consider the simple setting, where every action might be rewarding, only the expected reward for the state is changing

Challenge 1: uncertain outcomes

We consider the simple setting, where every action might be rewarding, only the expected reward for the state is changing

Challenge 1: uncertain outcomes

We consider the simple setting, where every action might be rewarding, only the expected reward for the state is changing

Every state has a Q value

Keep up the current state as long as the value of it exceeds the value of others.
OR
Keep the current state as long as the value of it exceeds the average value of states

Challenge 1: uncertain outcomes

We consider the simple setting, where every action might be rewarding, only the expected reward for the state is changing

Every state has a Q value

Keep up the current state as long as the value of it exceeds the value of others.
OR
Keep the current state as long as the value of it exceeds the average value of states

Challenge 1: uncertain outcomes

We consider the simple setting, where every action might be rewarding, only the expected reward for the state is changing

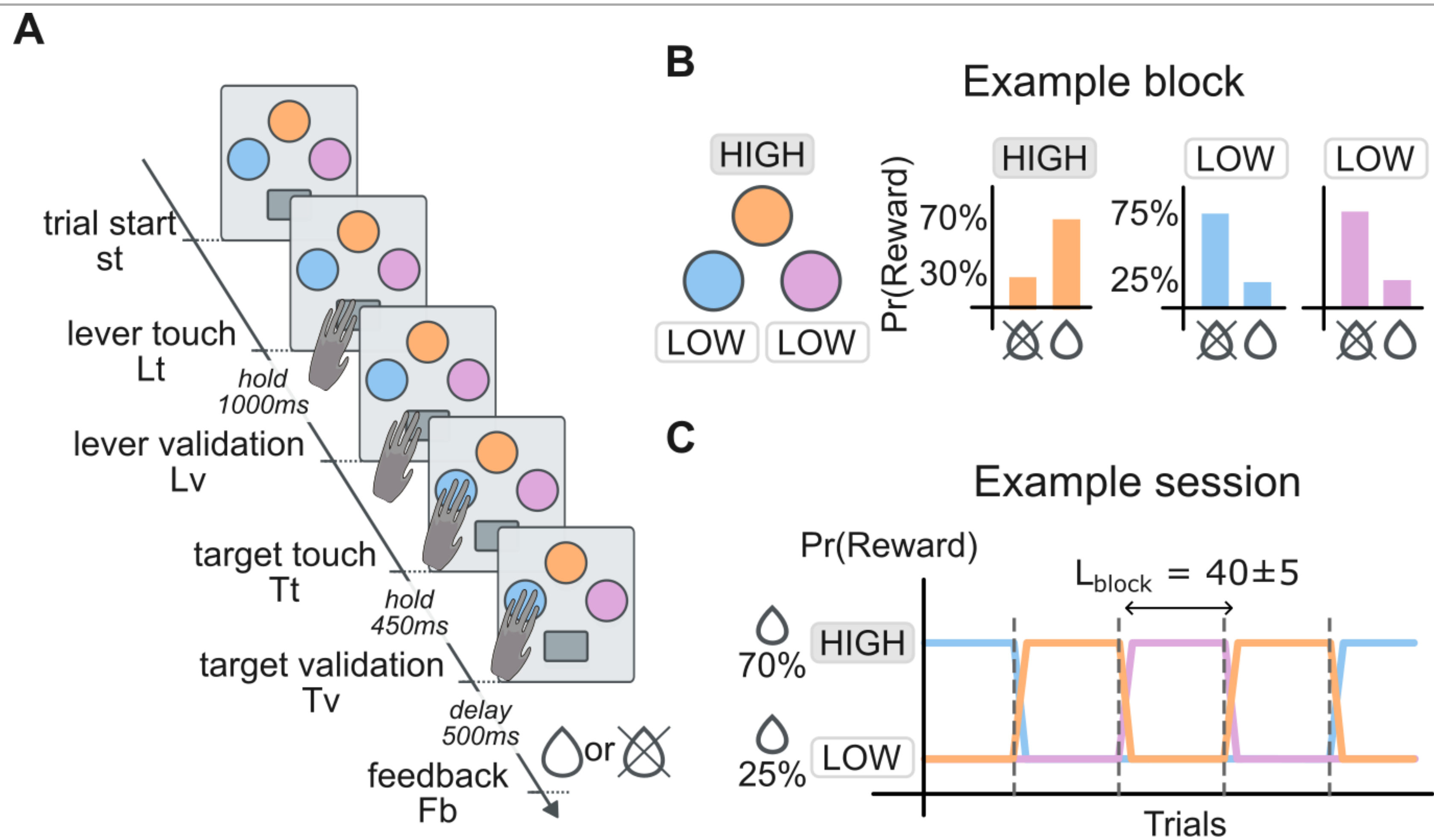
formal goal: $Q(s) = \mathbb{E}[\mathscr{R}]$

discounted reward: $\mathscr{R} = \sum_k \alpha^k r_k(s)$

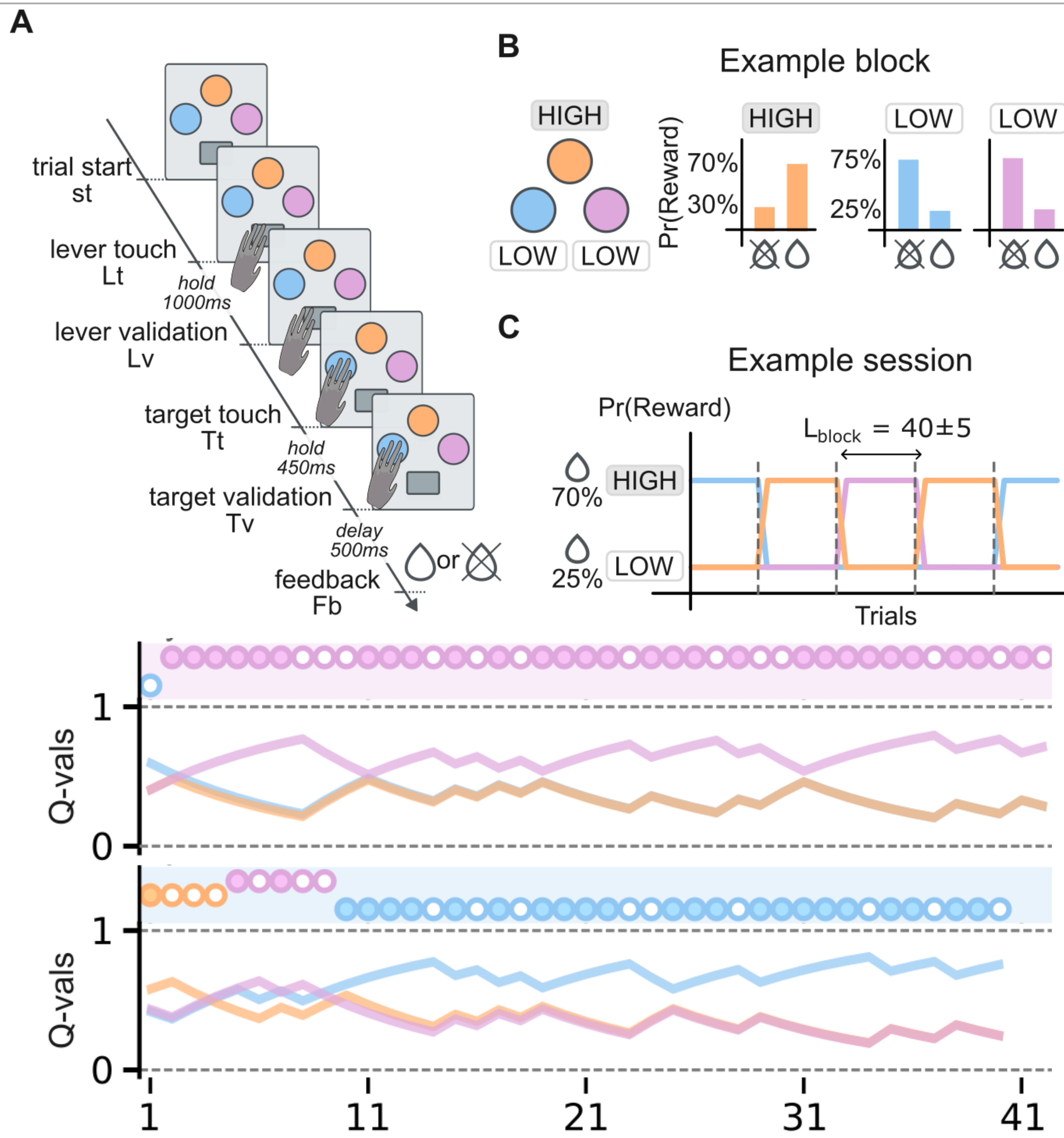
Every state has a Q value

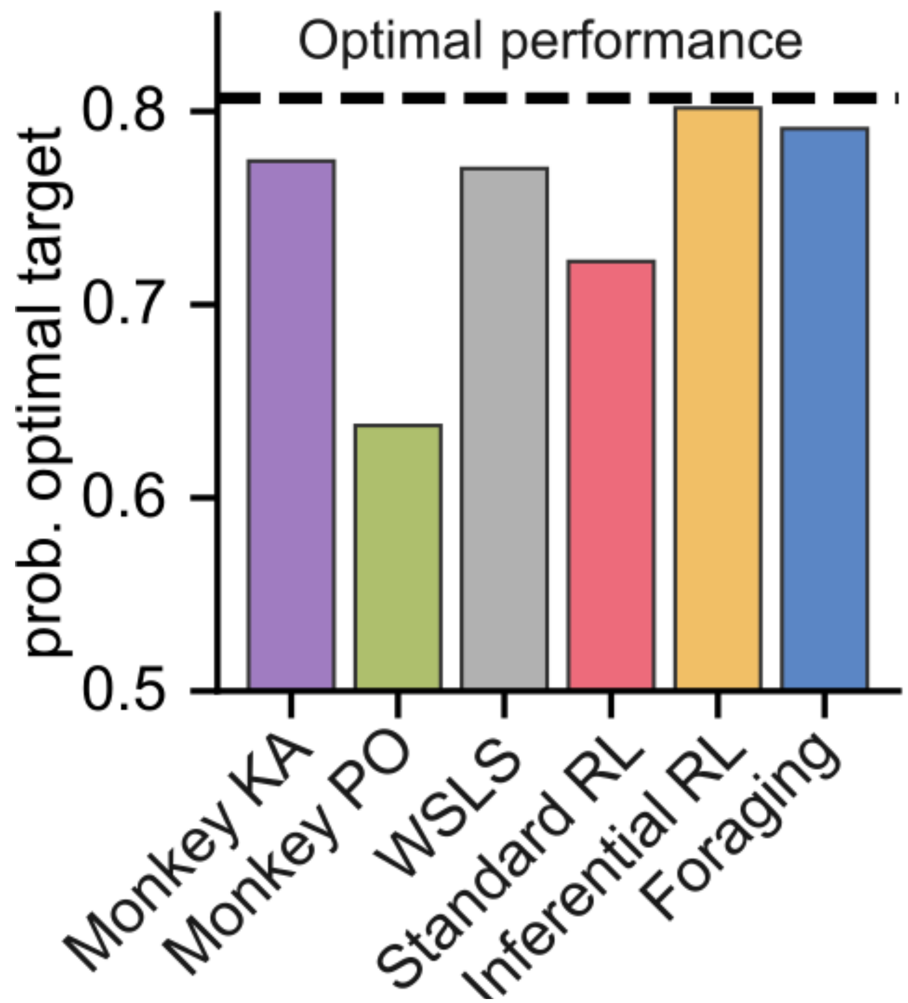
Keep up the current state as long as the value of it exceeds the value of others.
OR
Keep the current state as long as the value of it exceeds the average value of states

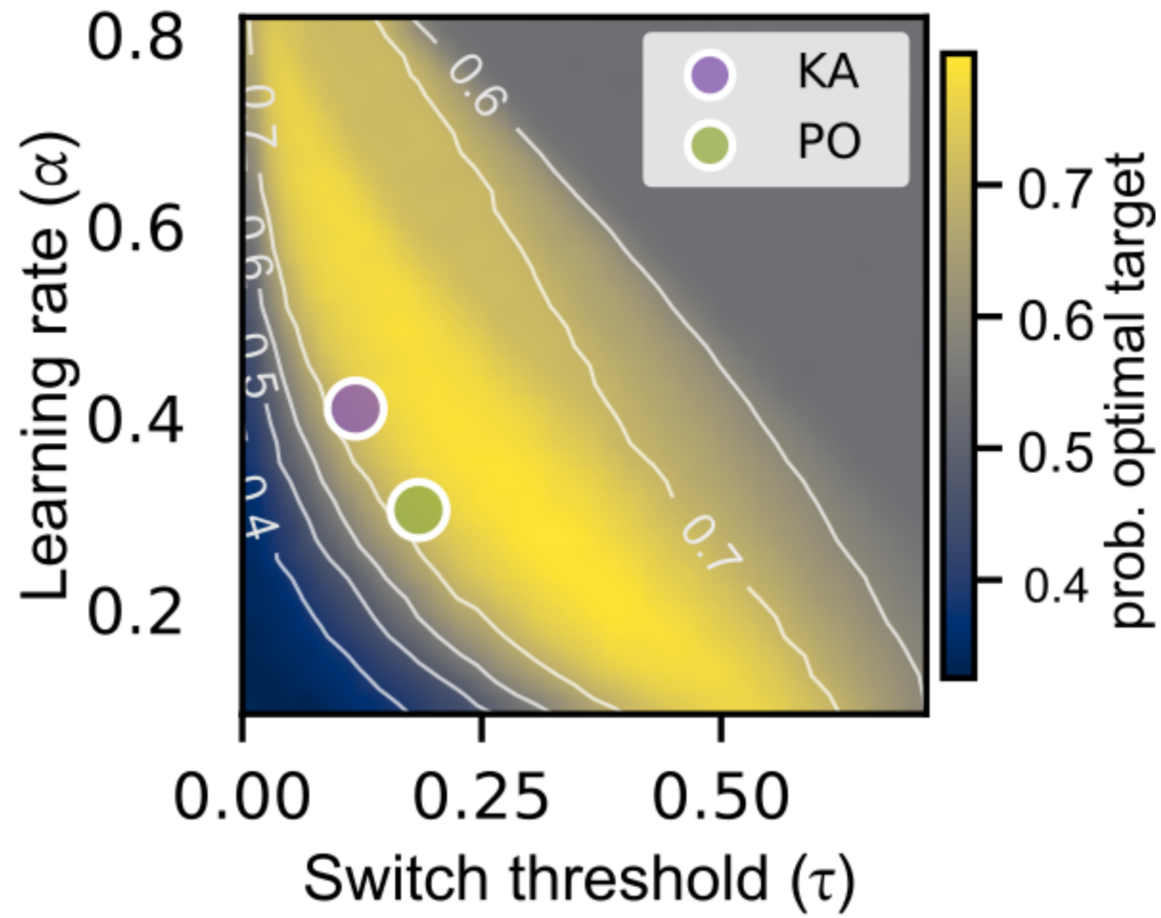
Neural correlates of RL

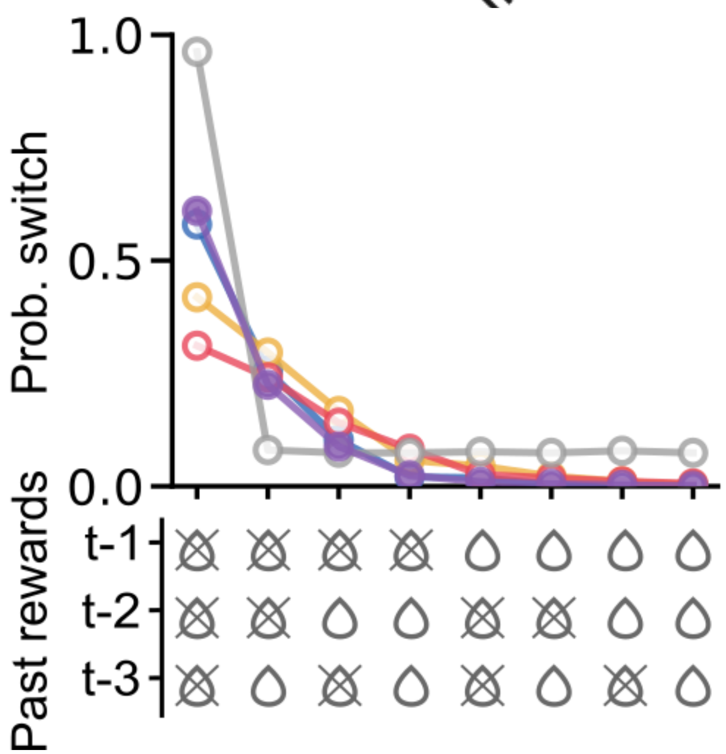


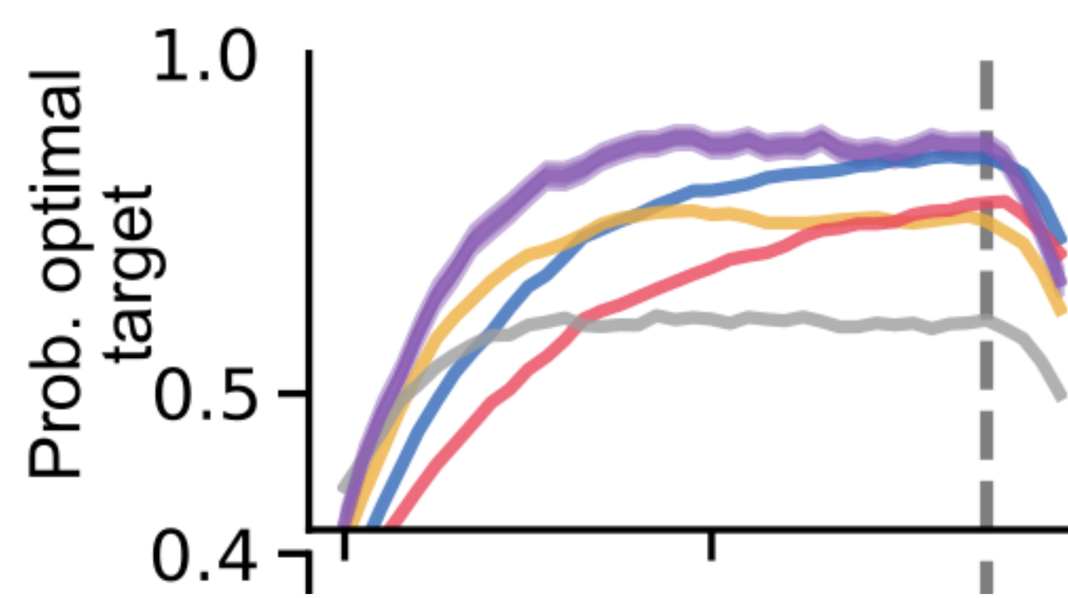
Neural correlates of RL



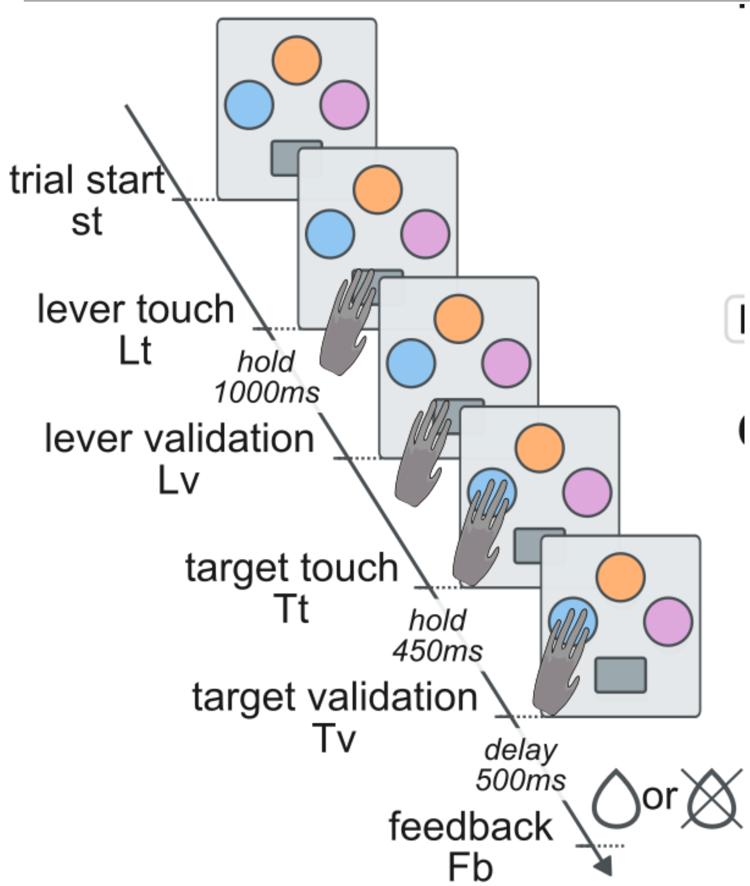

Optimal performance
prob. optimal target
0.8
0.7
0.6
0.5
Monkey KA
Monkey PO
WSLS
Standard RL
Inferential RL
Foraging

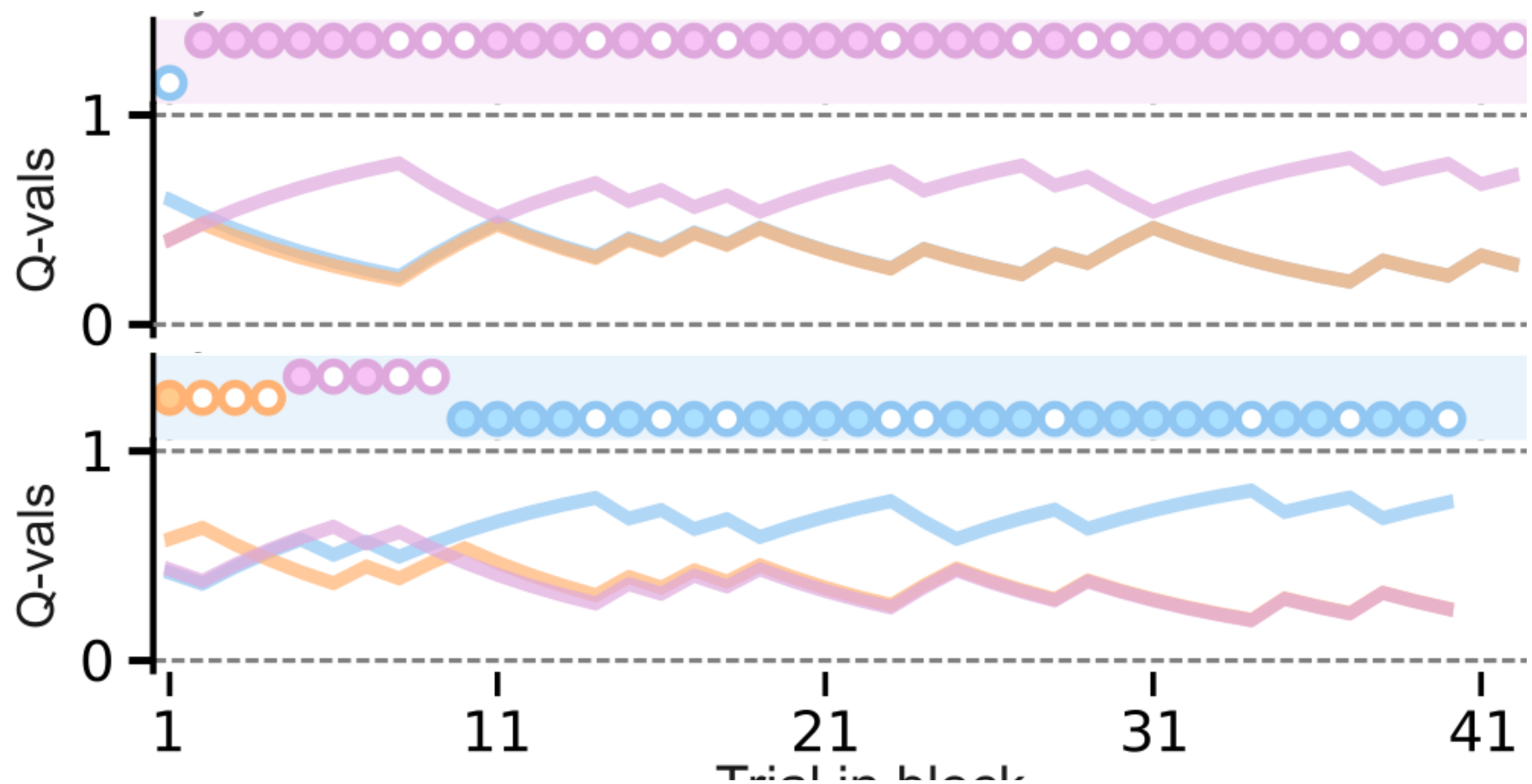

KA
PO
Learning rate (α)
0.8
0.6
0.4
0.2
0.00
0.25
0.50
Switch threshold (τ)
prob. optimal target
0.7
0.6
0.5
0.4


Prob. switch
1.0
0.5
0.0
Past rewards
t-1
t-2
t-3

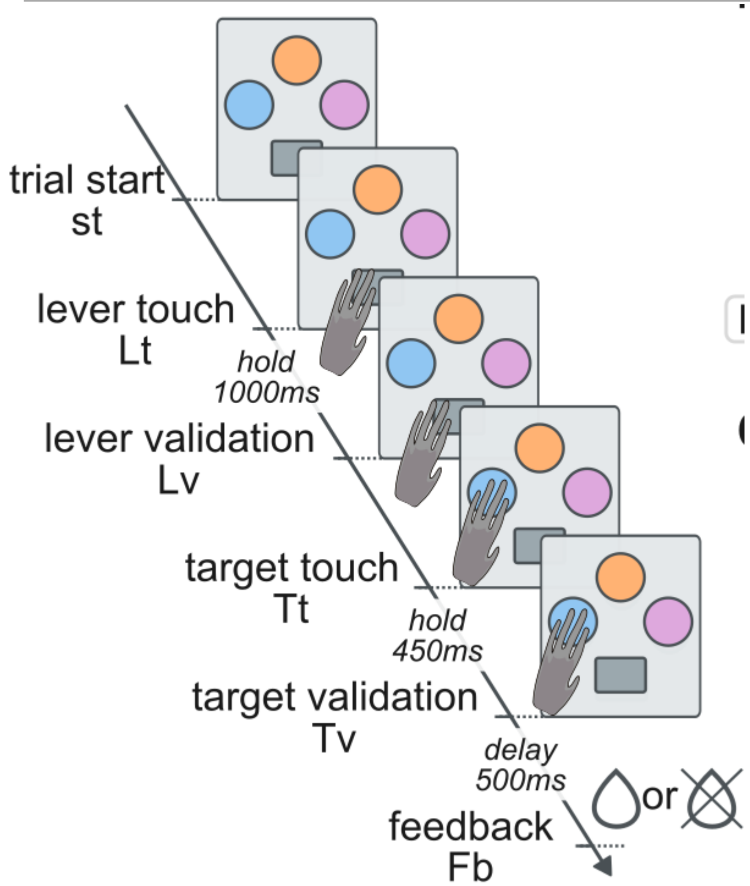

Prob. optimal target
1.0
0.5
0.4

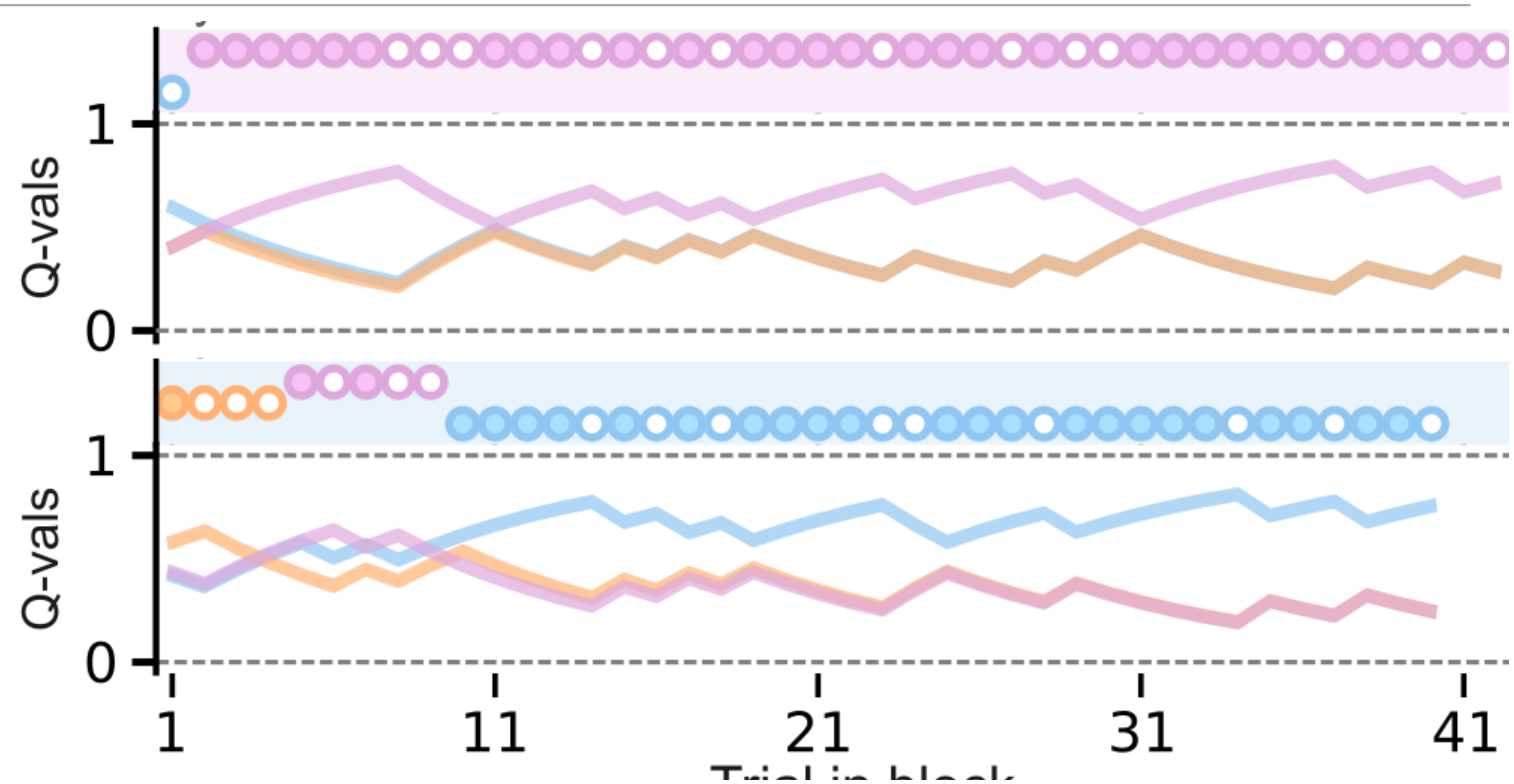
Neural correlates of RL

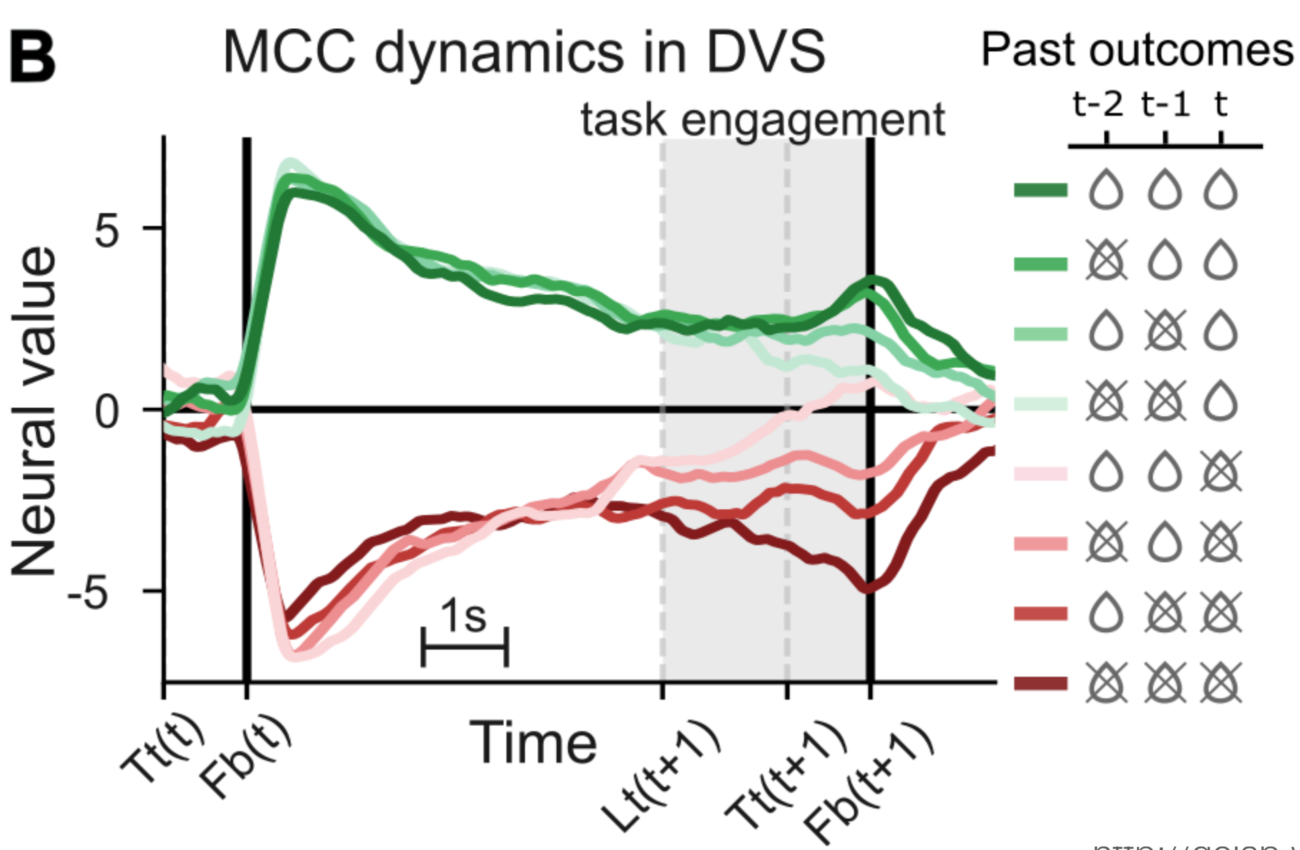




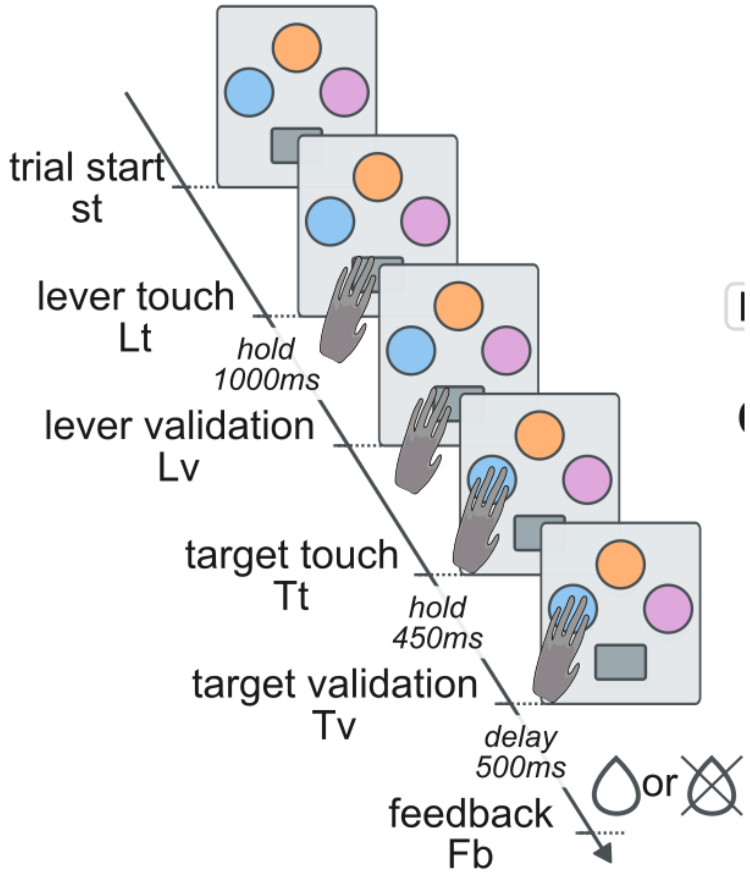
Neural correlates of RL

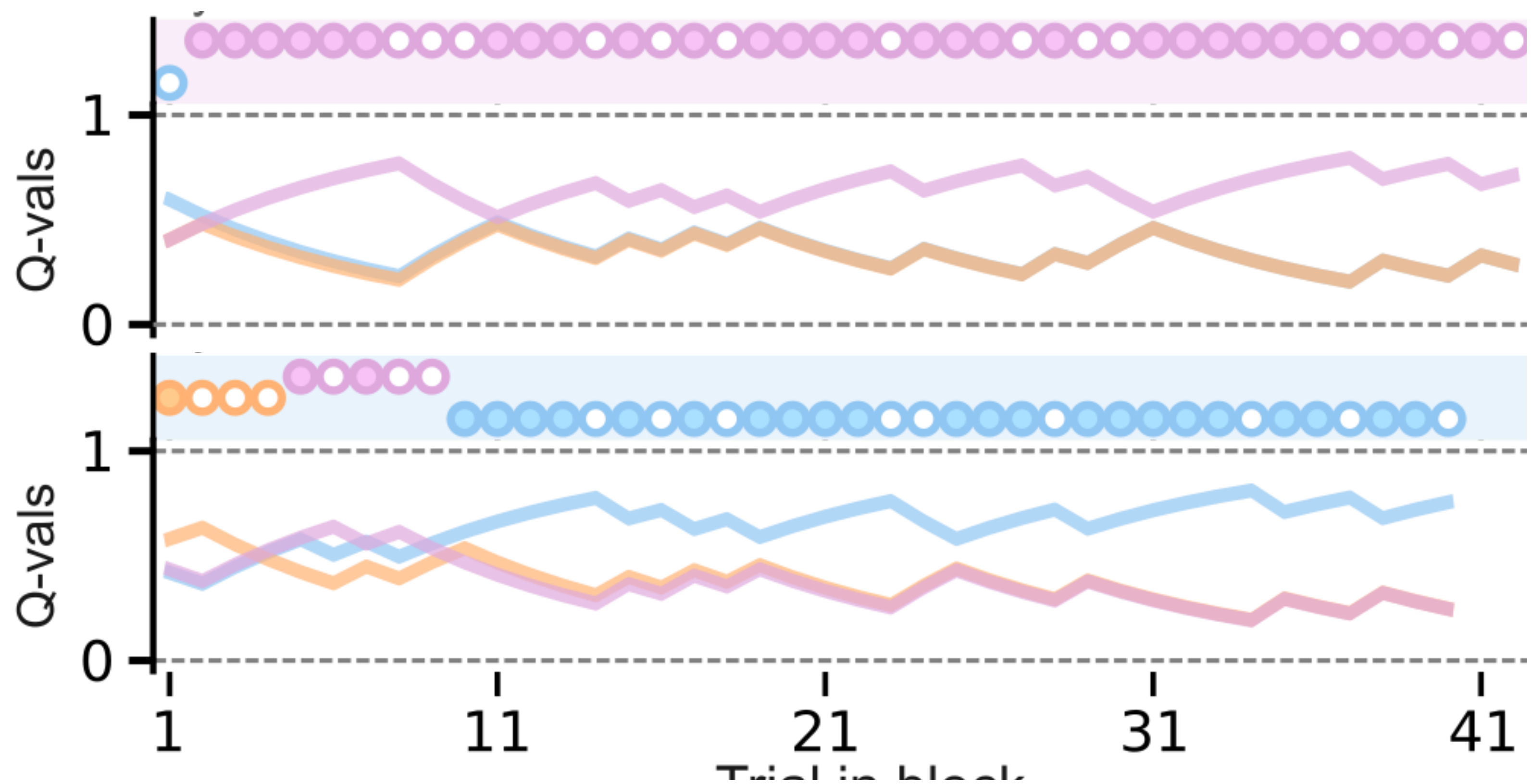


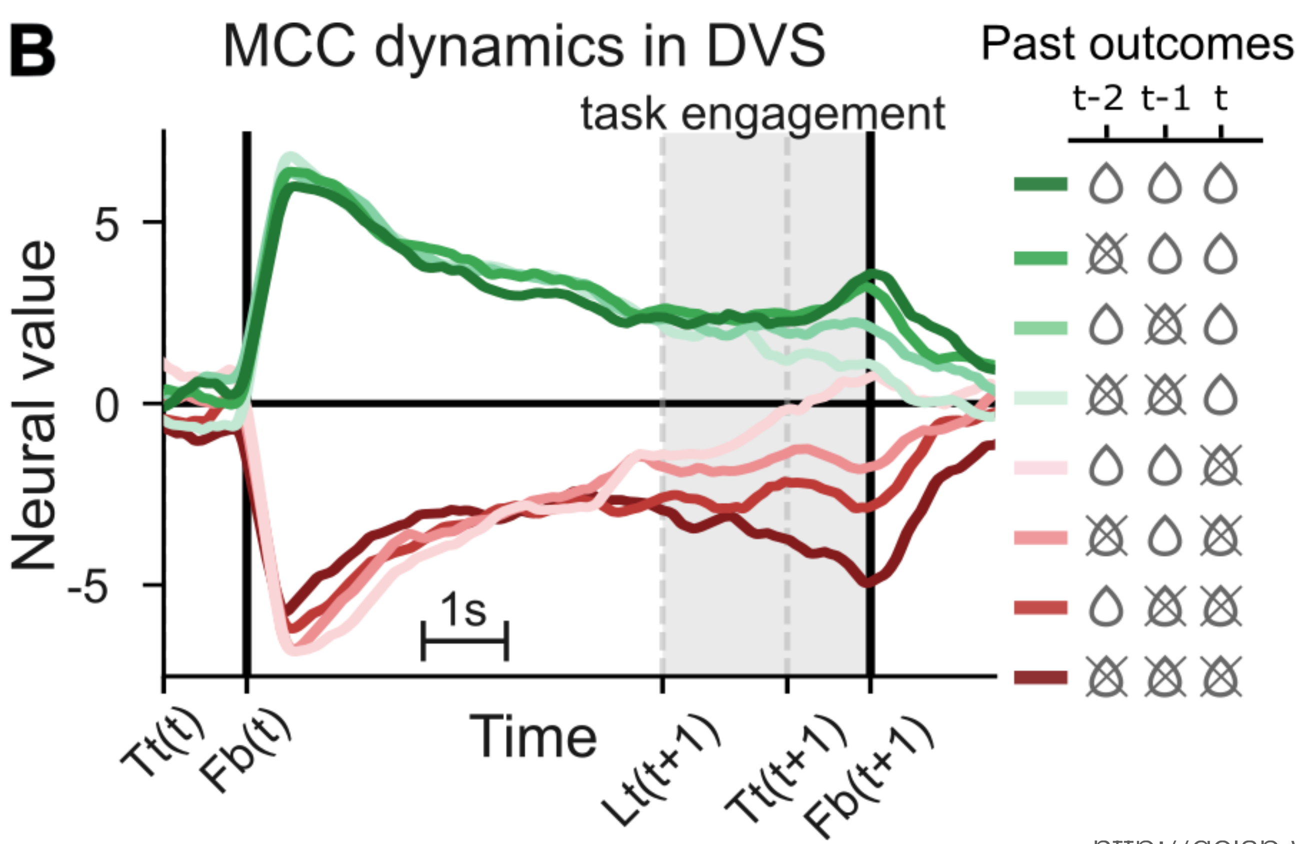


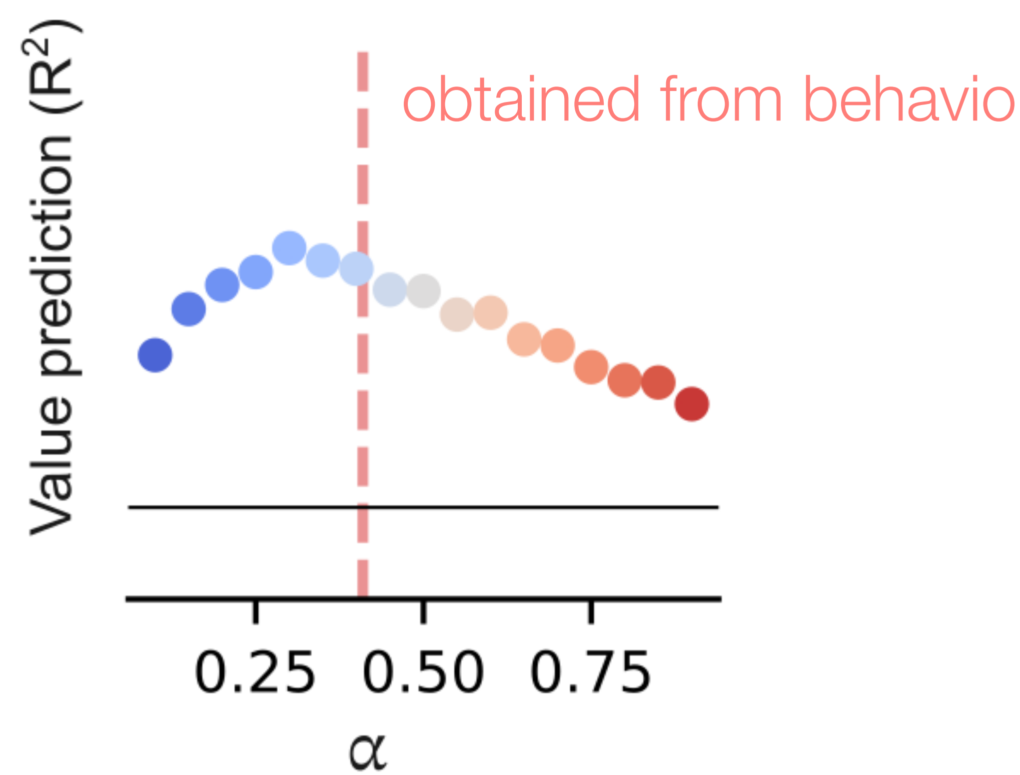


Neural correlates of RL



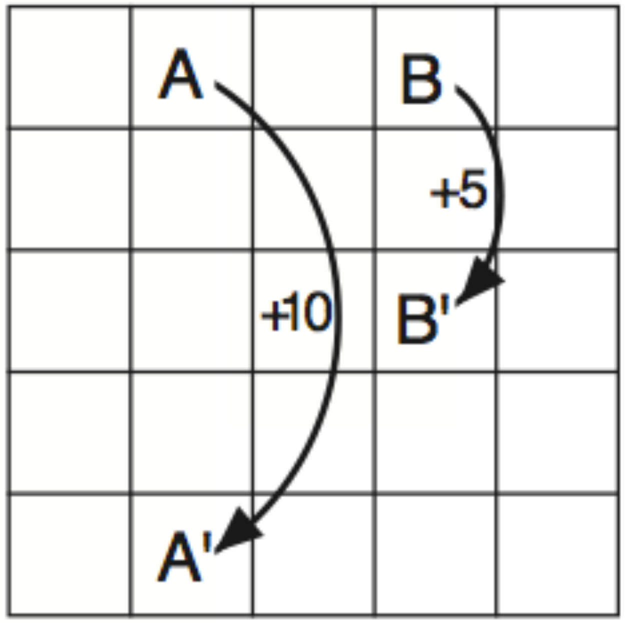




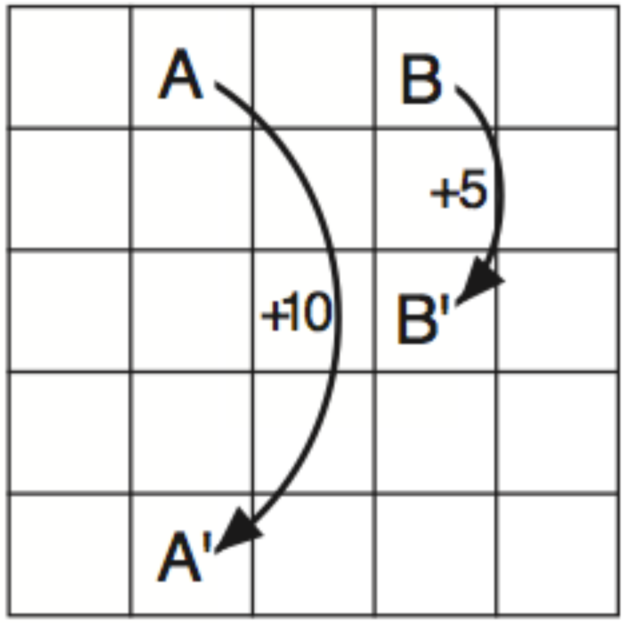


Simple model environment: gridworld

Simple model environment: gridworld

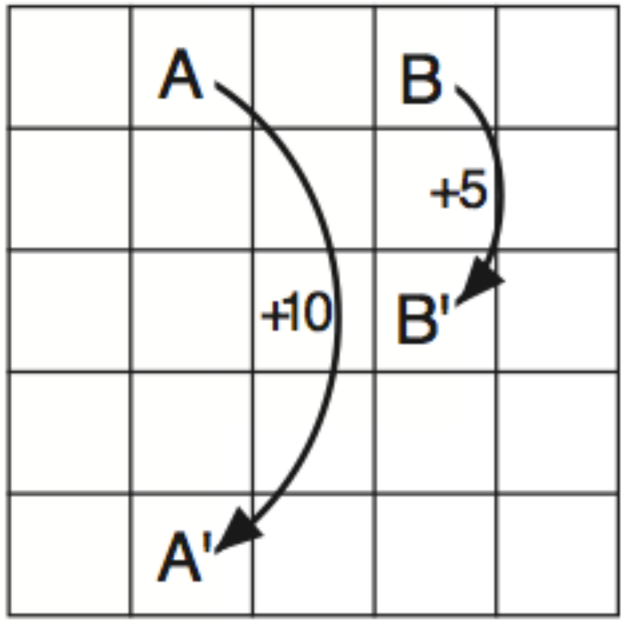


Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

Simple model environment: gridworld



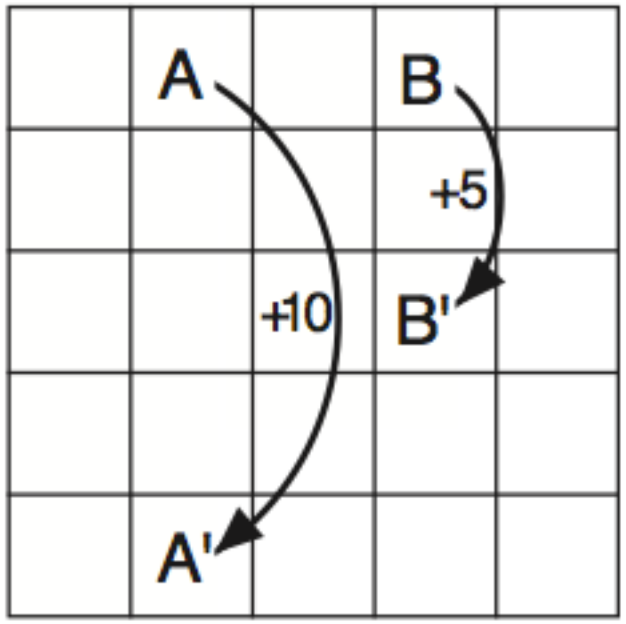
state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

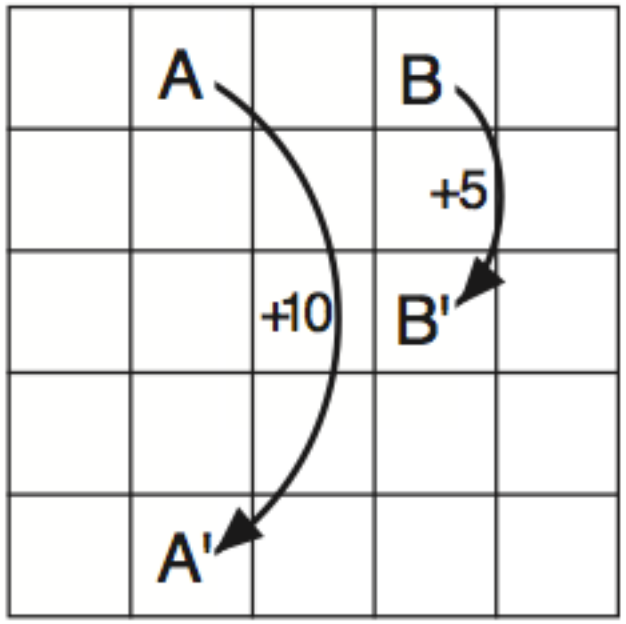
rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

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

policy: $\pi(a_t|s_t)$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

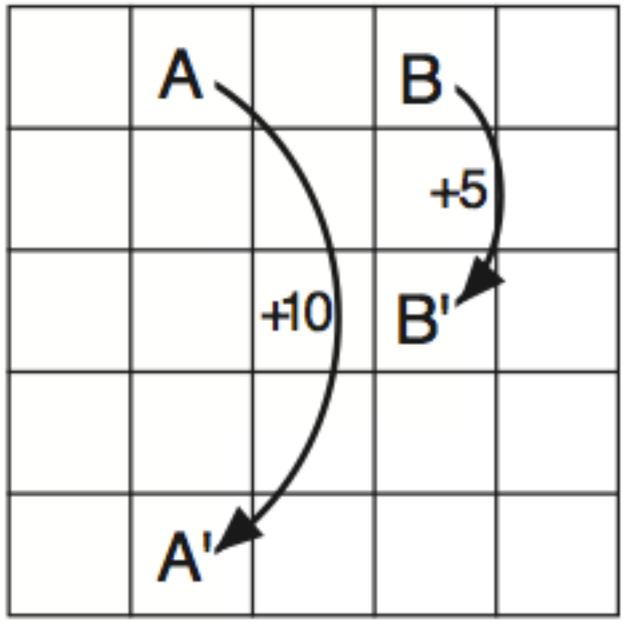
discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

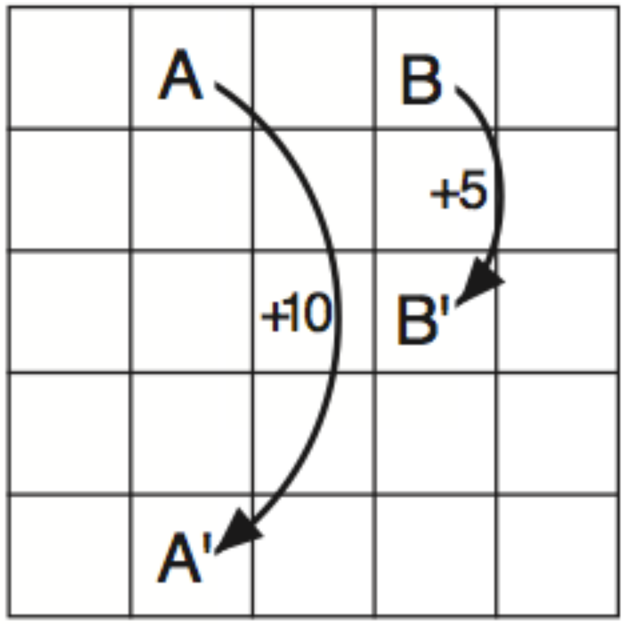
discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

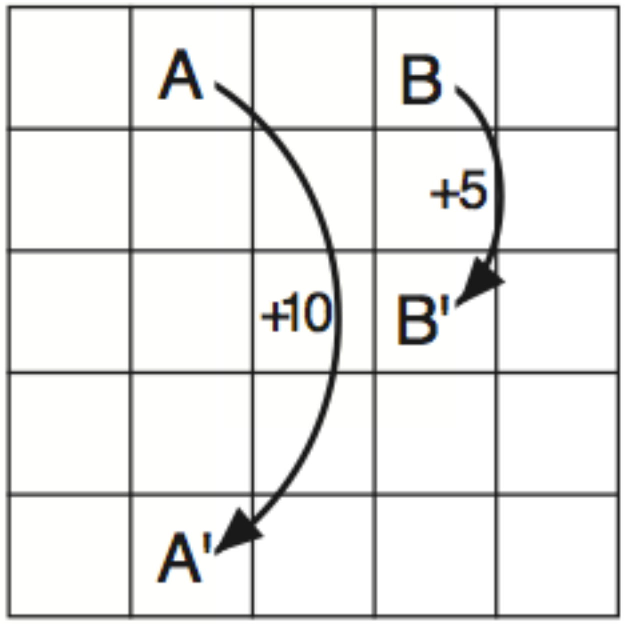
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

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

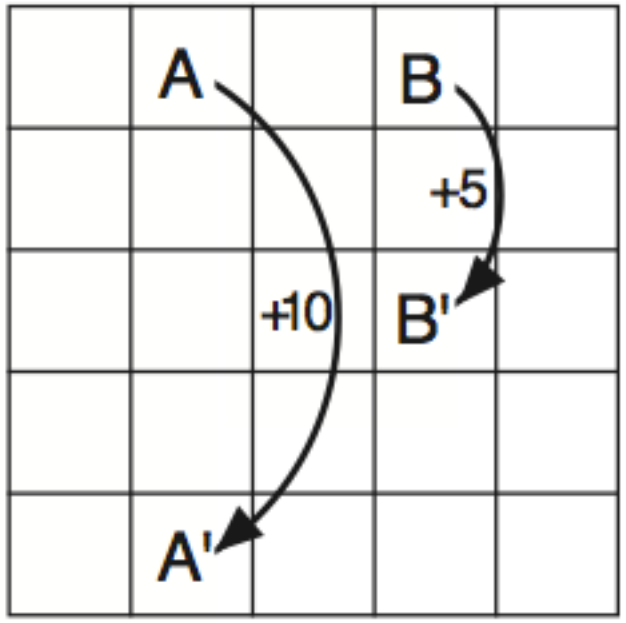
policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right]$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

policy: $\pi(a_t|s_t)$

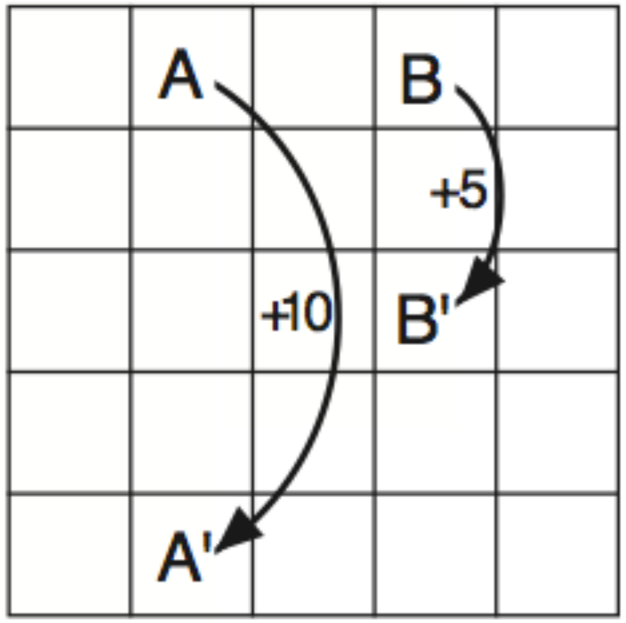
what is the value associated with a given state under a policy?
Bellmann equation:

$$\color{red}{V_\pi(s)} = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$

$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right]$$

$$= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right]$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

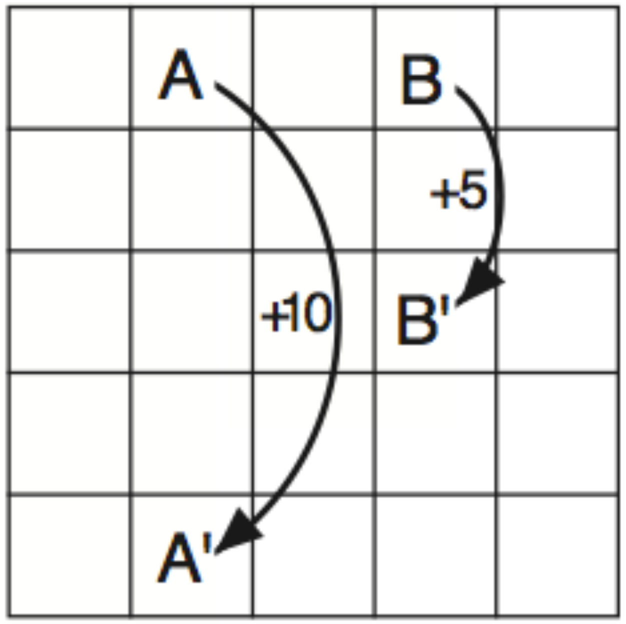
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
\color{red}{V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right]
\end{aligned}$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

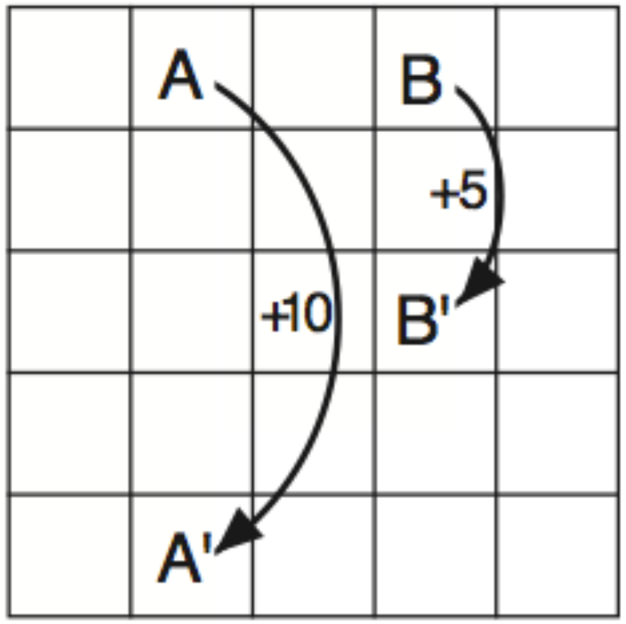
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma {\color{red}V_\pi(s')}\right]
\end{aligned}$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

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

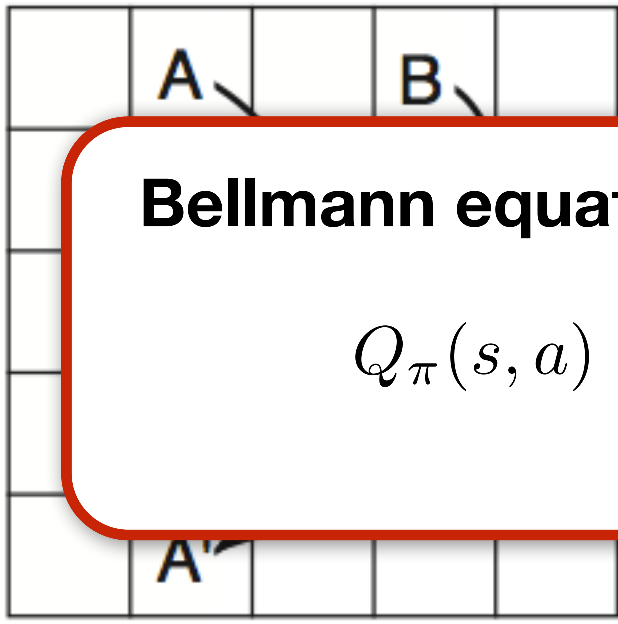
policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma {\color{red}V_\pi(s')}\right]
\end{aligned}$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

Bellmann equation for Q:

$$Q_\pi(s,a) = \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \sum_a \pi(a|s')\,\gamma\, Q_\pi(s',a)\right]$$

policy: $\pi(a_t|s_t)$

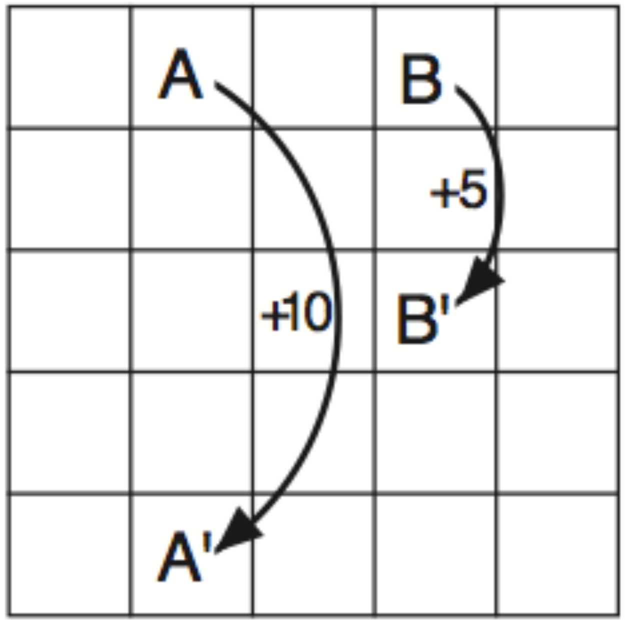
what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma V_\pi(s')\right]
\end{aligned}$$

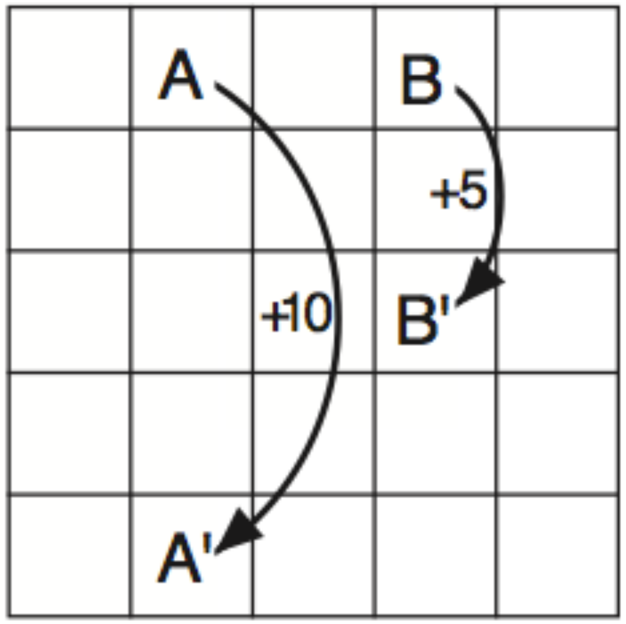
- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld

Simple model environment: gridworld

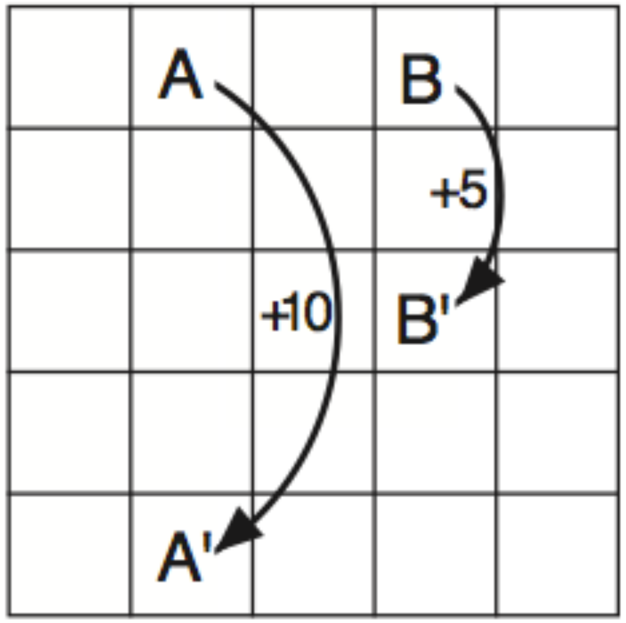


Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

Simple model environment: gridworld

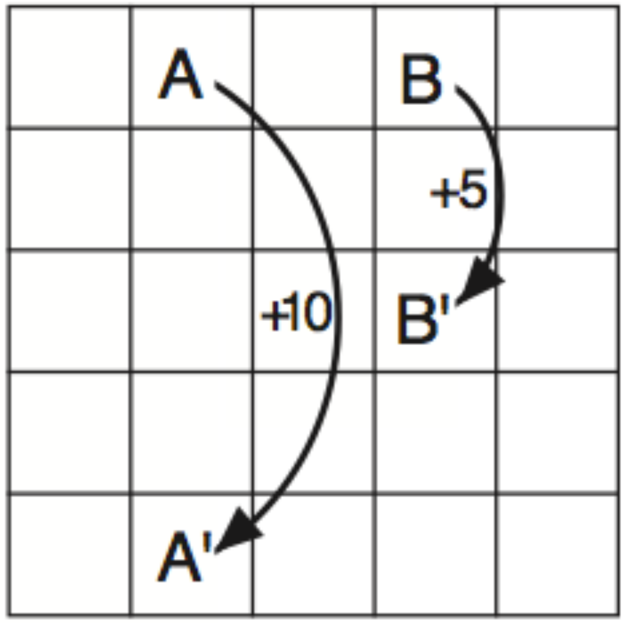


state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $$\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$$
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

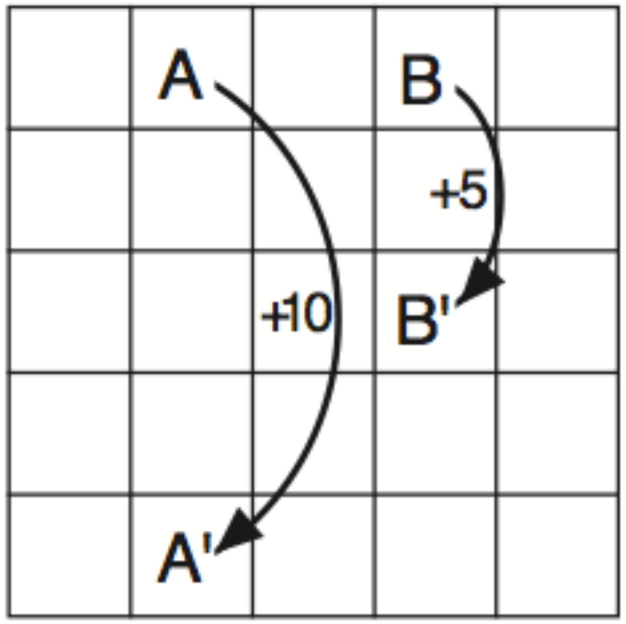
rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

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

policy: $\pi(a_t|s_t)$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

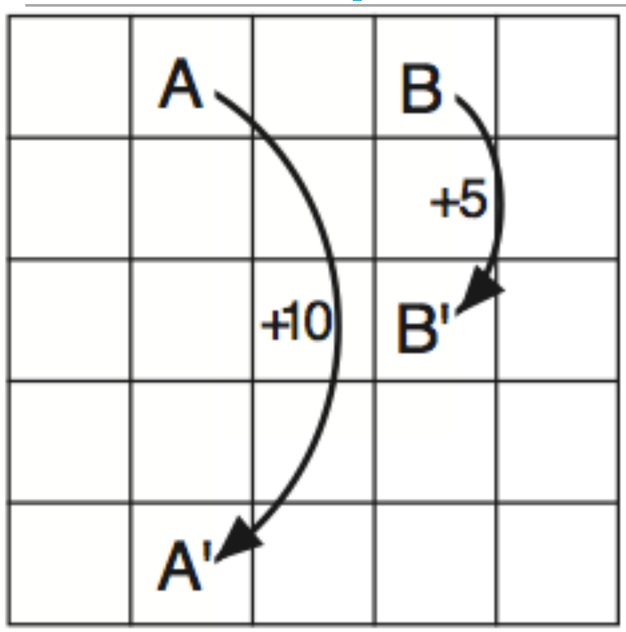
discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

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

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

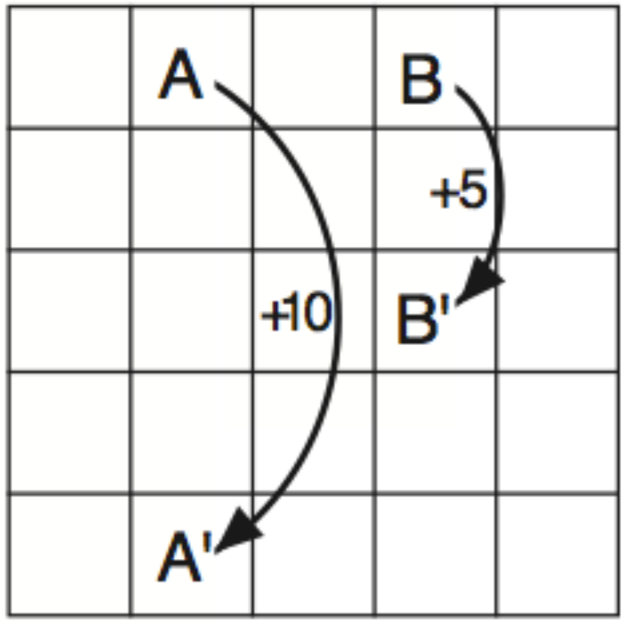
discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

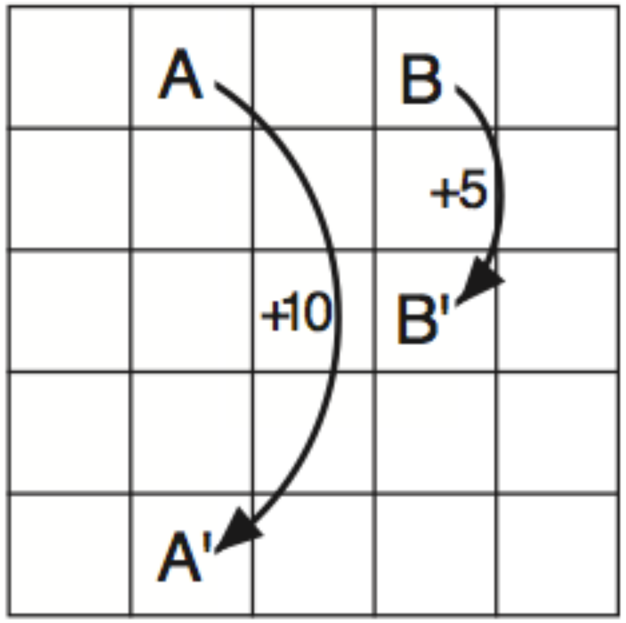
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \dots$

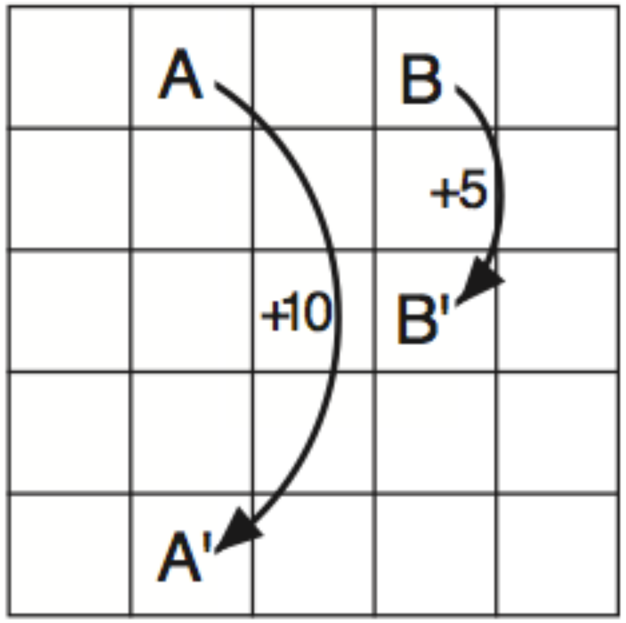
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$V_\pi(s) = \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s]$$
$$= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right]$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

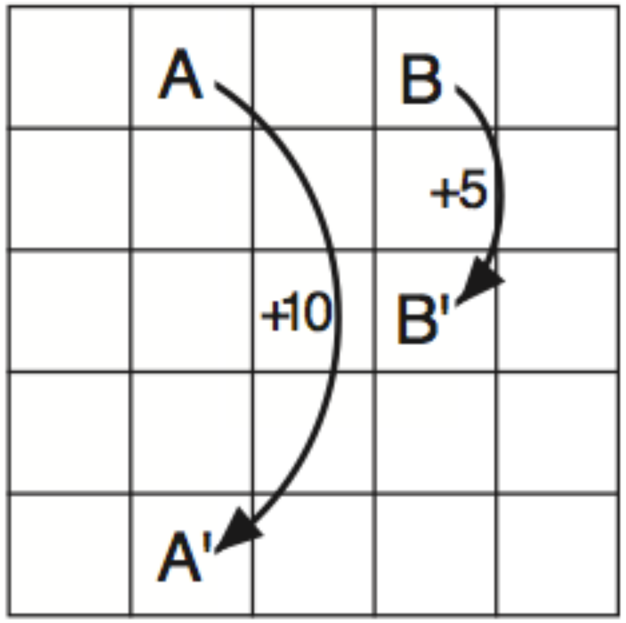
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$
\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \,|\, S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right]
\end{aligned}
$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

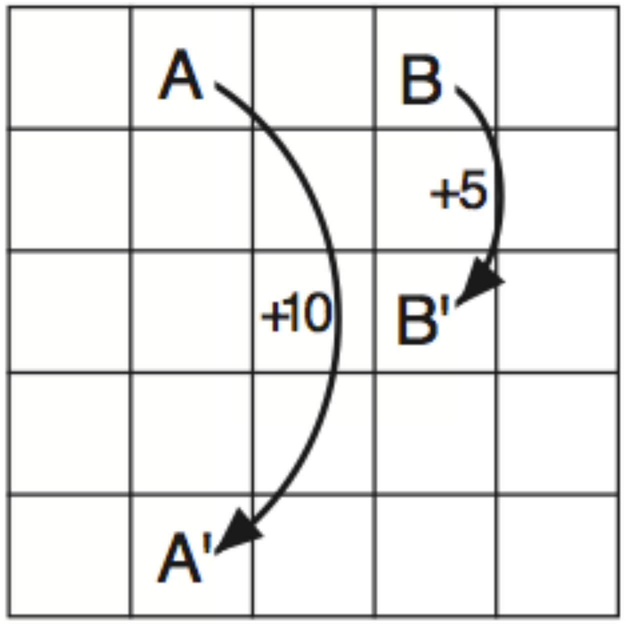
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\ &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\ &= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\ &= \sum_a \pi(a|s) \sum_{s'} P(s'|a, s)\left[r(s, a, s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \end{aligned}$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

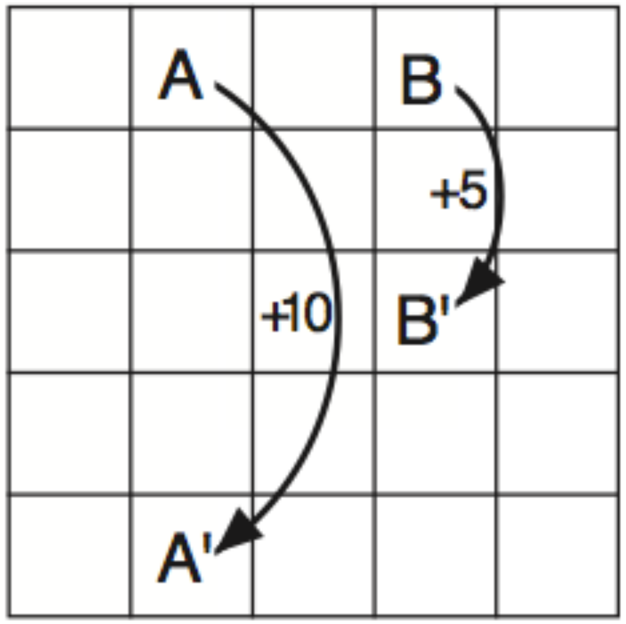
$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma {\color{red}V_\pi(s')}\right]
\end{aligned}$$

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

discounting: $\mathcal{R}_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+2} + \ldots$

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

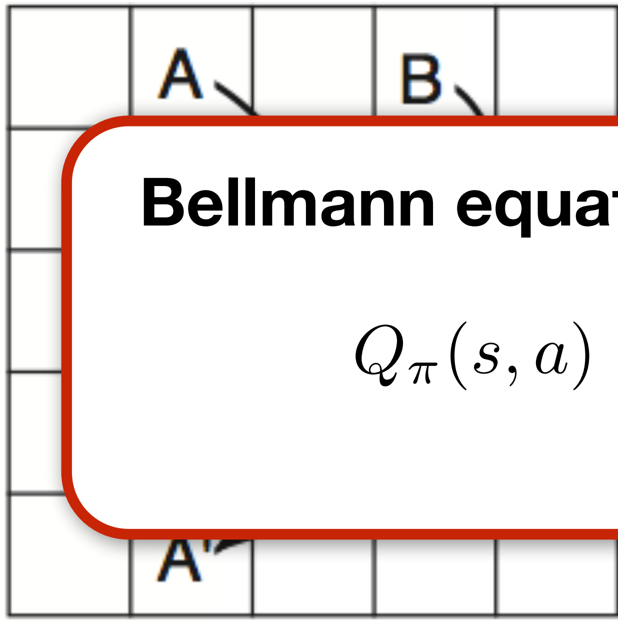
policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
{\color{red}V_\pi(s)} &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma {\color{red}V_\pi(s')}\right]
\end{aligned}$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

Simple model environment: gridworld



state transitions $P(s_{t+1}|s_t, a_t)$

rewards: $r(s_t, a_t, s_{t+1})$

Bellmann equation for Q:

$$Q_\pi(s,a) = \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \sum_{a} \pi(a|s')\,\gamma\, Q_\pi(s',a)\right]$$

policy: $\pi(a_t|s_t)$

what is the value associated with a given state under a policy?
Bellmann equation:

$$\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi[\mathcal{R}_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, S_t = s\right] \\
&= \mathbb{E}_\pi\left[r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, S_t = s'\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s'} P(s'|a,s)\left[r(s,a,s') + \gamma V_\pi(s')\right]
\end{aligned}$$

- consistency relationship between states
- depends on policy
- optimal policy: highest value
- learning: find the optimal policy

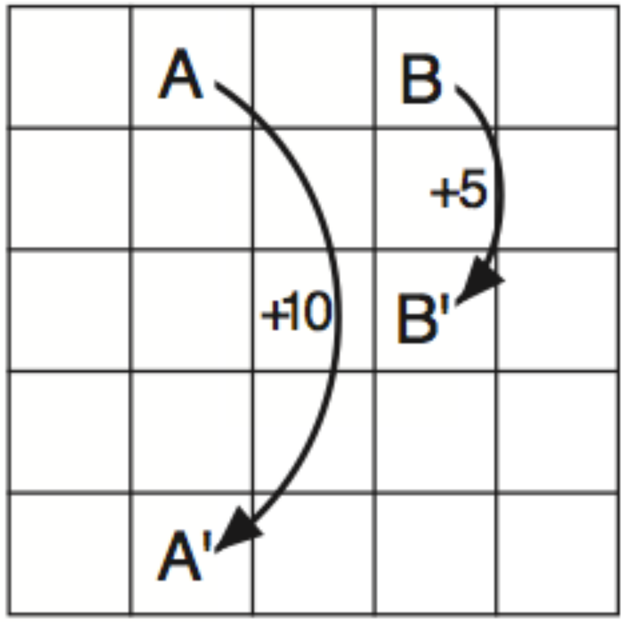
Simple model environment: gridworld

Computing the value function, V(s)

Simple model environment: gridworld

Computing the value function, V(s)

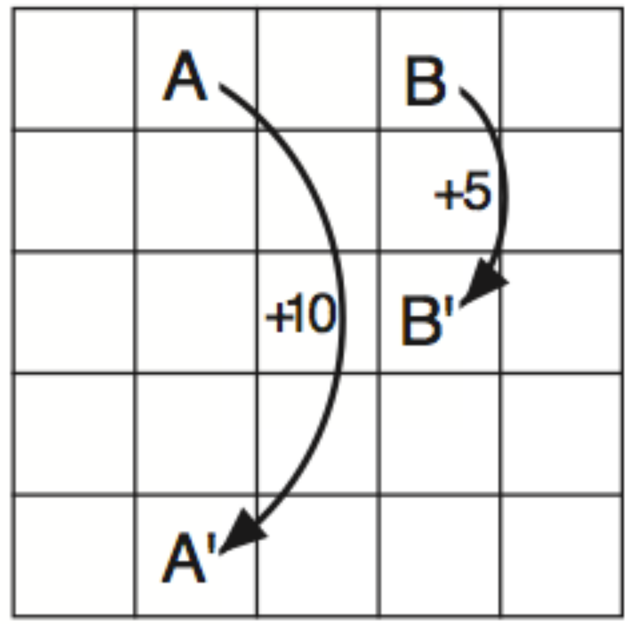
random policy



Simple model environment: gridworld

Computing the value function, V(s)

random policy

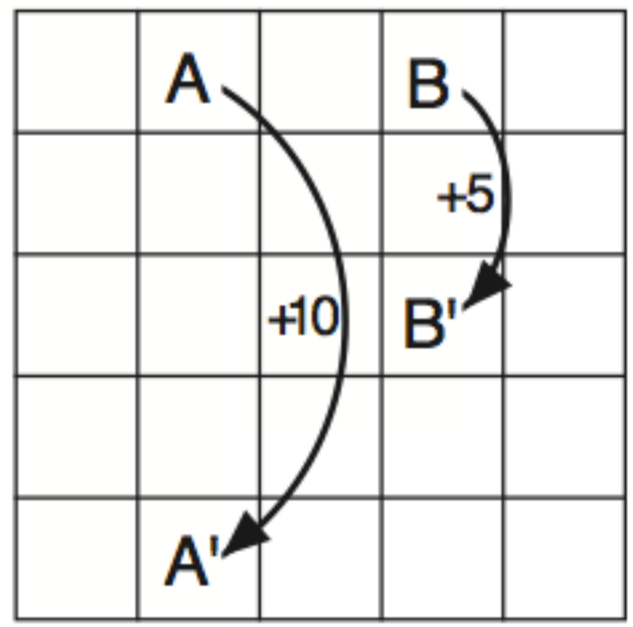


3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Simple model environment: gridworld

Computing the value function, V(s)

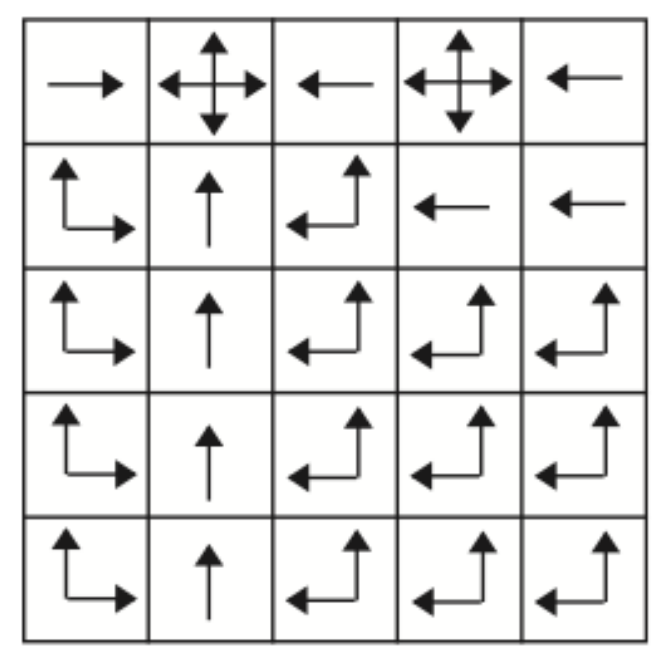
random policy



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

optimal policy

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7



Alternative solutions to Bellmann equation

Alternative solutions to Bellmann equation

dynamic programming

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_{\pi}(s)$ | $\boldsymbol{\pi}(a|s)$

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s) \mid \boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s) \mid \mathbf{V}_\pi(s)$

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s) \mid \boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s) \mid \mathbf{V}_\pi(s)$

Monte Carlo techniques

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_{\pi}(s)$ | $\boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s)$ | $\mathbf{V}_{\pi}(s)$

Monte Carlo techniques

- wait until the reward arrives

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s) \mid \boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s) \mid \mathbf{V}_\pi(s)$

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\Big[\mathcal{R}_t - Q_\pi(s_t, r_t)\Big]$$

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s)$ | $\boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s)$ | $\mathbf{V}_\pi(s)$

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\Big[\mathcal{R}_t - Q_\pi(s_t, r_t)\Big]$$

temporal difference (TD-) learning

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi$(s) | $\boldsymbol{\pi}$(a|s)
- policy improvement: $\boldsymbol{\pi}$(a|s) | $\mathbf{V}_\pi$(s)

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\mathcal{R}_t - Q_\pi(s_t, r_t) \Big]$$

temporal difference (TD-) learning

- don't wait with the updates until rewards!

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s) \mid \boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s) \mid \mathbf{V}_\pi(s)$

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\mathcal{R}_t - Q_\pi(s_t, r_t) \Big]$$

temporal difference (TD-) learning

- don't wait with the updates until rewards!
- use intermediate value estimates to update the action-values!

Alternative solutions to Bellmann equation

dynamic programming

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_{\pi}$(s) | $\boldsymbol{\pi}$(a|s)
- policy improvement: $\boldsymbol{\pi}$(a|s) | $\mathbf{V}_{\pi}$(s)

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_{\pi}(s_t, a_t) \leftarrow Q_{\pi}(s_t, a_t) + \alpha \Big[\mathcal{R}_t - Q_{\pi}(s_t, r_t) \Big]$$

temporal difference (TD-) learning

- don't wait with the updates until rewards!
- use intermediate value estimates to update the action-values!

$$Q_{\pi}(s_t, a_t) \leftarrow Q_{\pi}(s_t, a_t) + \alpha \Big[\underbrace{r_{t+1} + \gamma\, Q_{\pi}(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_{\pi}(s_t, a_t) \Big]$$

Alternative solutions to Bellmann equation

dynamic programming

model-based

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi(s) \mid \boldsymbol{\pi}(a|s)$
- policy improvement: $\boldsymbol{\pi}(a|s) \mid \mathbf{V}_\pi(s)$

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\Big[\mathcal{R}_t - Q_\pi(s_t, r_t)\Big]$$

temporal difference (TD-) learning

- don't wait with the updates until rewards!
- use intermediate value estimates to update the action-values!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\Big[\underbrace{r_{t+1} + \gamma\, Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t)\Big]$$

Alternative solutions to Bellmann equation

dynamic programming

model-based

- use simulations to
- solve the Bellmann equations iteratively
- need an accurate model of the environment
- policy evaluation: $\mathbf{V}_\pi$(s) | $\boldsymbol{\pi}$(a|s)
- policy improvement: $\boldsymbol{\pi}$(a|s) | $\mathbf{V}_\pi$(s)

Monte Carlo techniques

- wait until the reward arrives
- update value functions based on average returns

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\mathcal{R}_t - Q_\pi(s_t, r_t) \Big]$$

temporal difference (TD-) learning

model-free

- don't wait with the updates until rewards!
- use intermediate value estimates to update the action-values!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\underbrace{r_{t+1} + \gamma\, Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t) \Big]$$

Intuition for Temporal Difference Learning

Intuition for Temporal Difference Learning

temporal difference learning

Intuition for Temporal Difference Learning

temporal difference learning

- don't wait with the updates until the reward!

Intuition for Temporal Difference Learning

temporal difference learning

- don't wait with the updates until the reward!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\underbrace{r_{t+1} + \gamma\, Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t) \Big]$$

Intuition for Temporal Difference Learning

temporal difference learning

- don't wait with the updates until the reward!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \left[\underbrace{r_{t+1} + \gamma\, Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t) \right]$$

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Intuition for Temporal Difference Learning

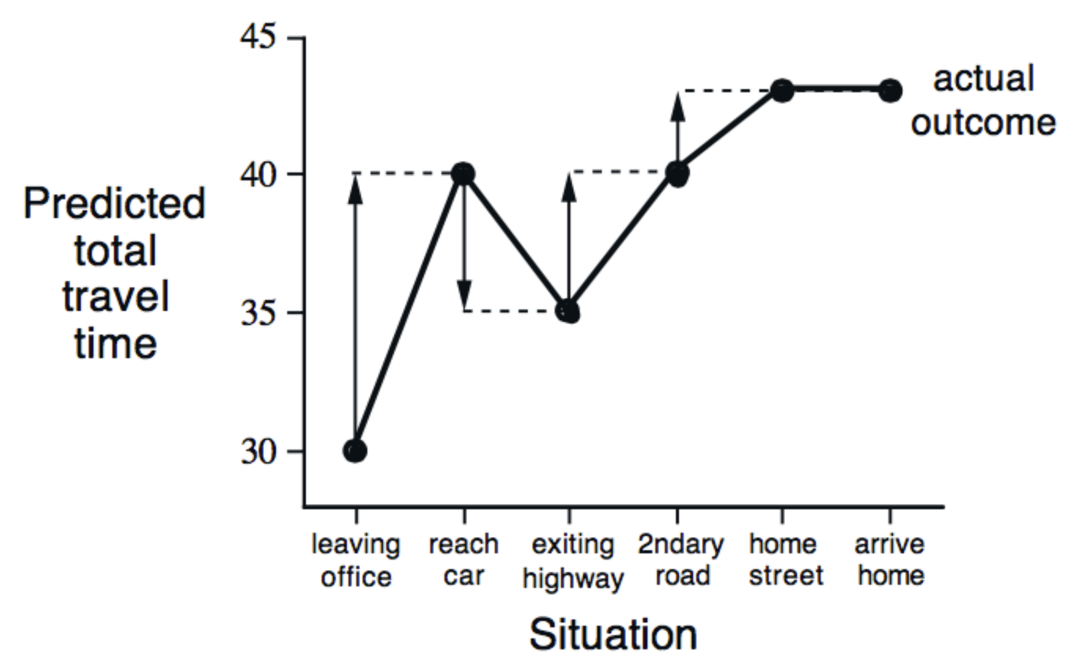
temporal difference learning

- don't wait with the updates until the reward!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha\left[\underbrace{r_{t+1} + \gamma\, Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t)\right]$$

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

temporal difference



Intuition for Temporal Difference Learning

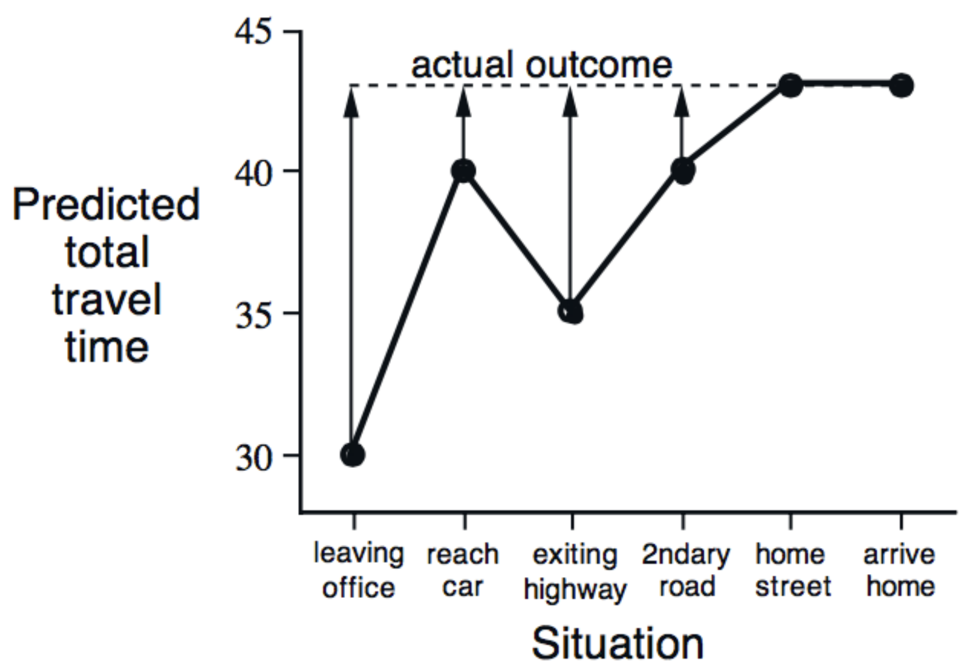
temporal difference learning

- don't wait with the updates until the reward!

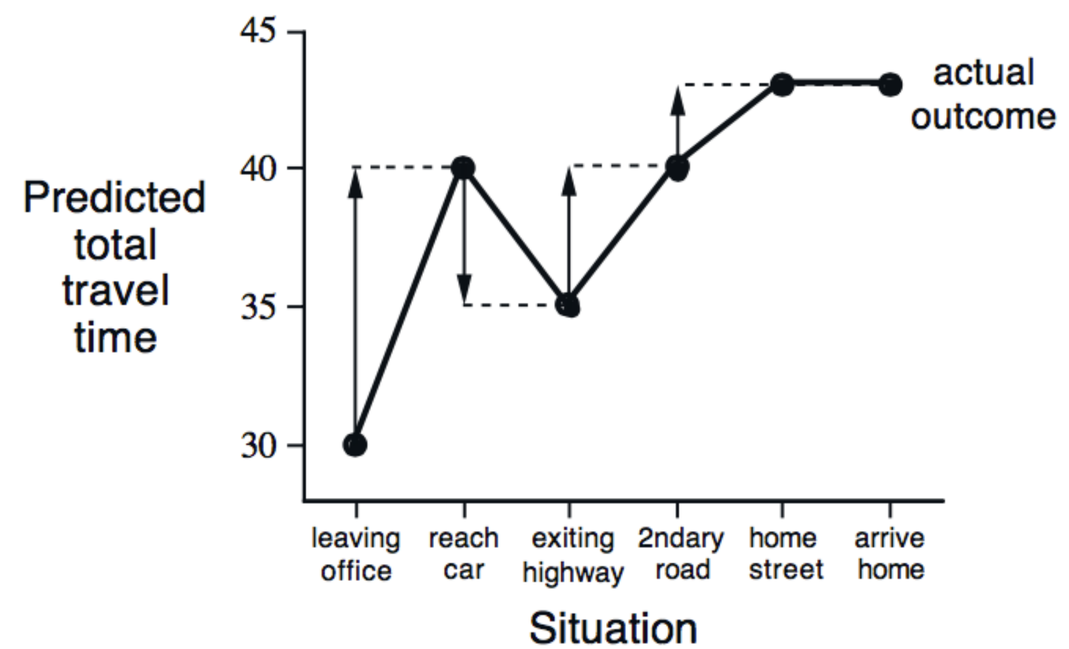
$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \left[\underbrace{r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q_\pi(s_t, a_t) \right]$$

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43

Monte Carlo



temporal difference



RL in practice

temporal difference learning

- don’t wait with the updates until the end of the trial!

$$Q_\pi(s_t, a_t) \leftarrow Q_\pi(s_t, a_t) + \alpha \Big[\underbrace{r_{t+1} + \gamma\, Q(s_{t+1}, a_{t+1})}_{\text{estimate}} - Q(s_t, a_t) \Big]$$

- Q-learning:
  a powerful algorithm that has been applied to many different real-word problems

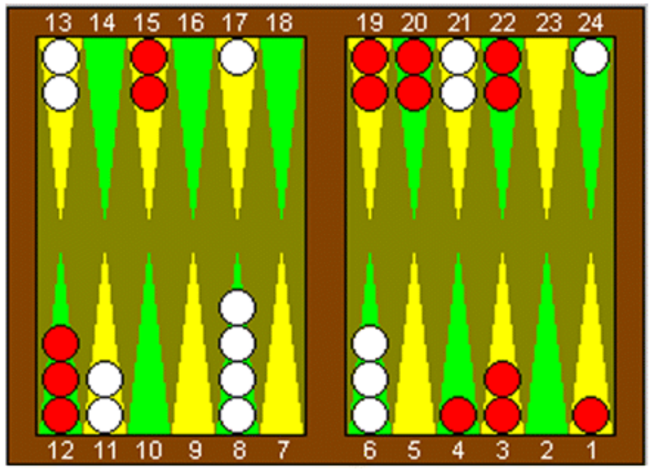
$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \underbrace{\Big[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \Big]}_{\text{prediction error}}$$

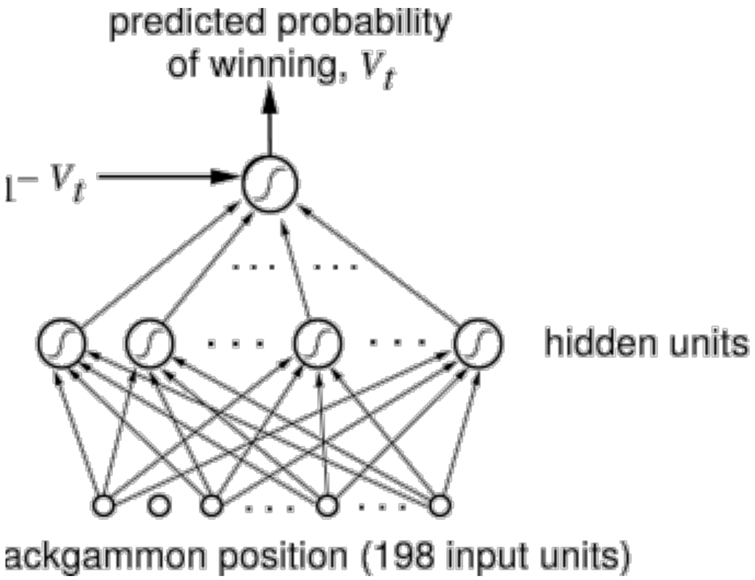
neuronal implementation:

- learn the state space - representational learning
- tabular vs. function approximation
- learning is based on prediction error
- is reward prediction error calculated by the brain?

Decision making with a neural network and TD learning

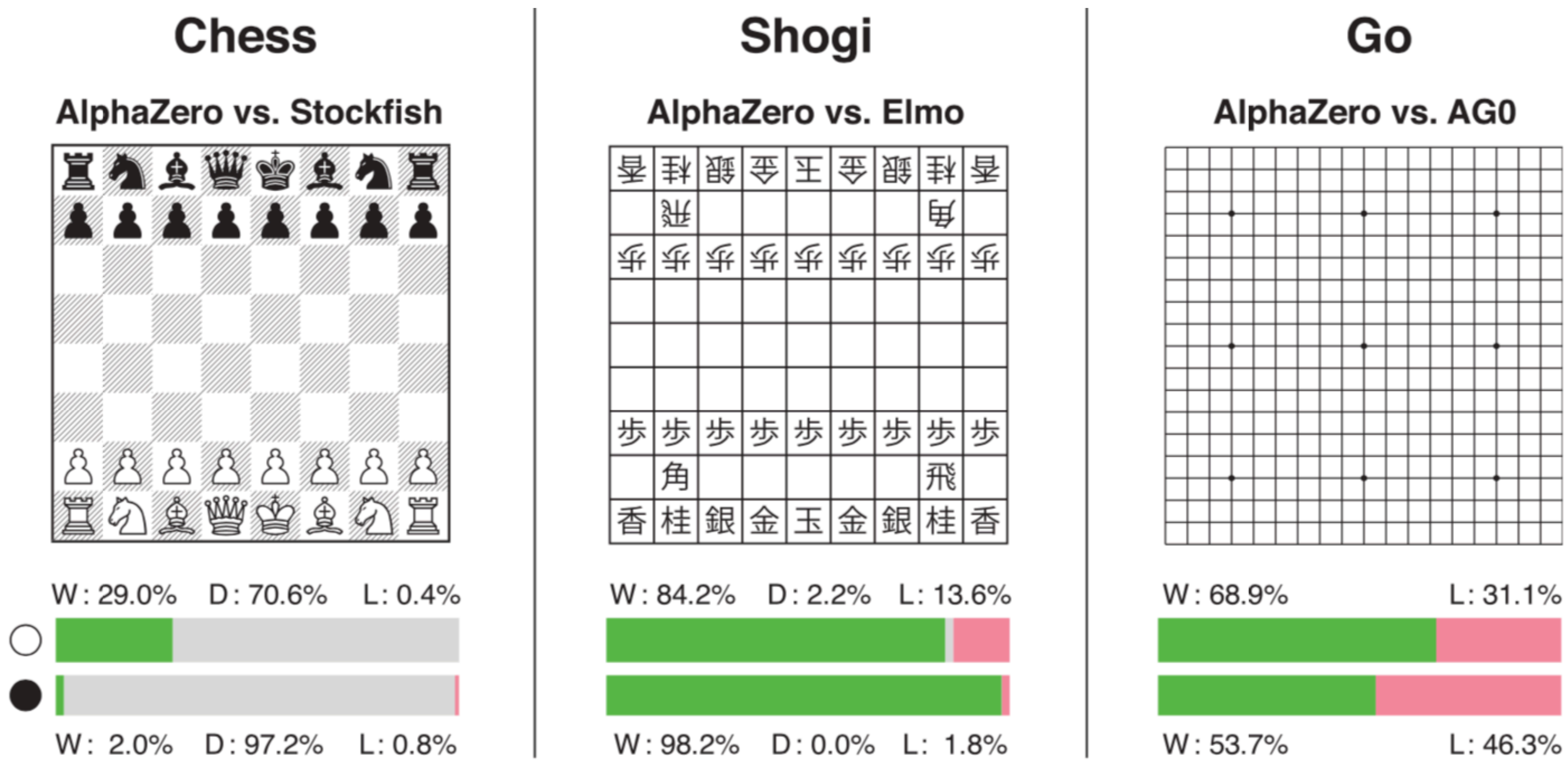
- Gerard Tesauro TD-backgammon
  - Multi-layer neural network
  - Input: possible states achieved by potential moves
  - Output: the probability of winning from an actual state
- Based on these, a policy can be established
- Result: performance is compatible with the best human players
- Training the algorithm takes about 5s





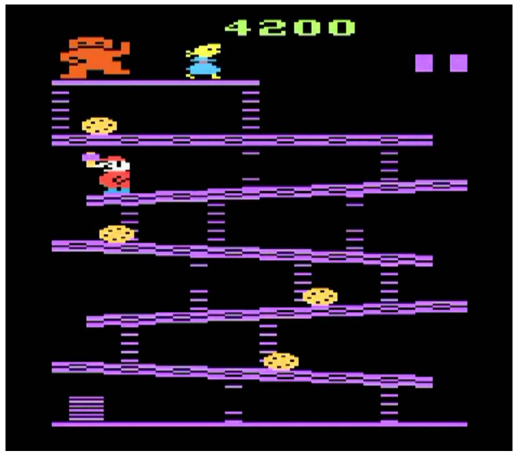
Decision making with a neural network and TD learning Deep Q learning

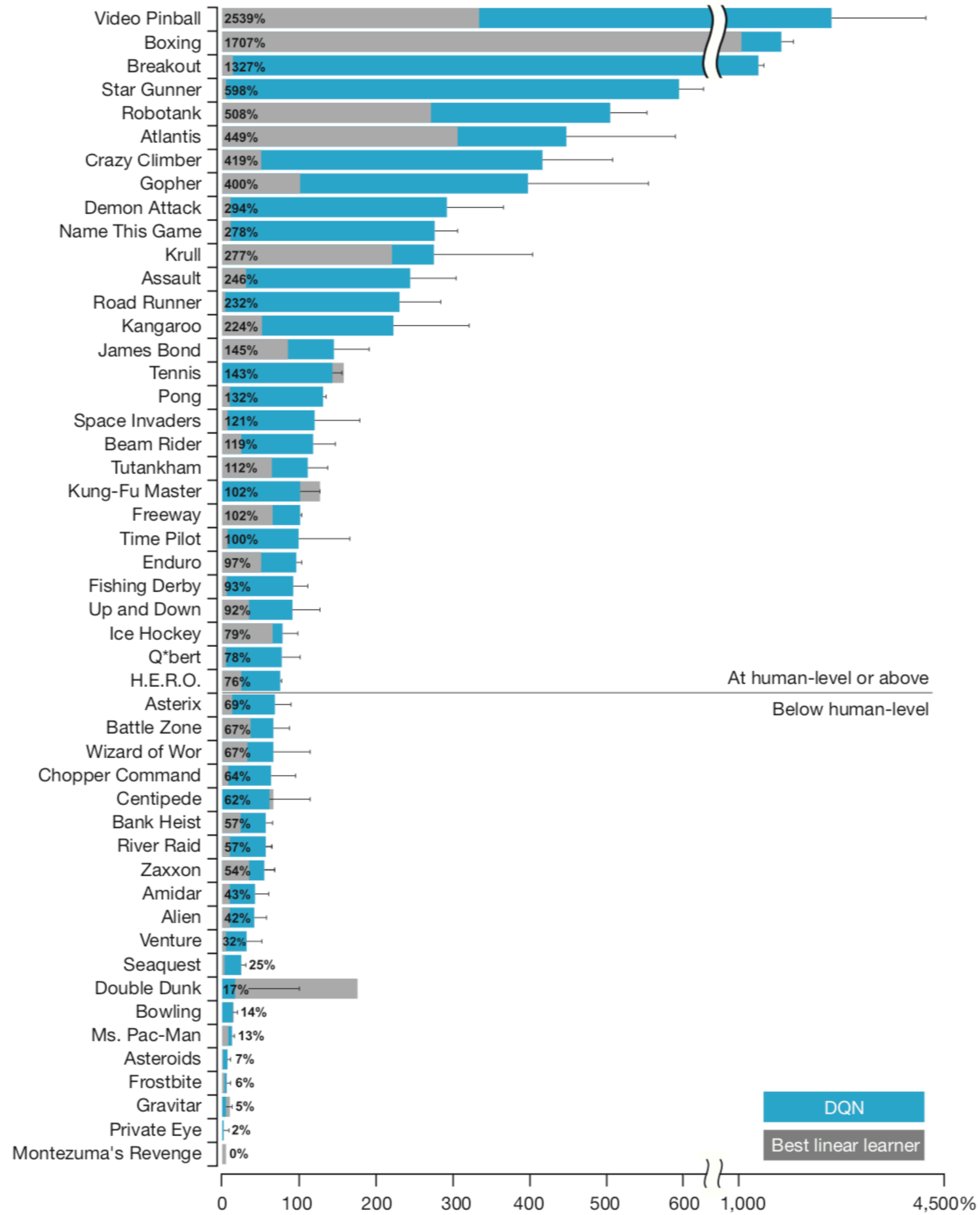
AlphaZero (Silver et al., 2018)



Decision making with a neural network and TD learning Deep Q learning

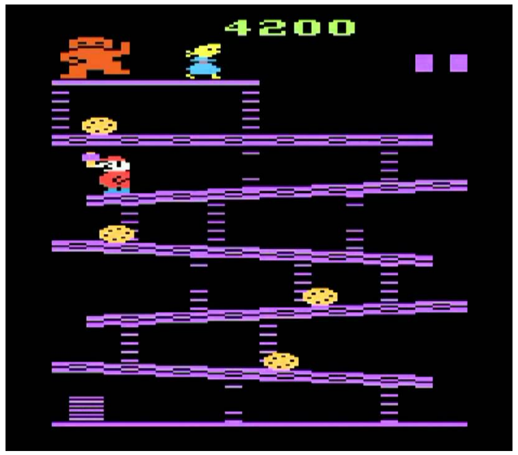
Atari games (Mnih et al., 2015)

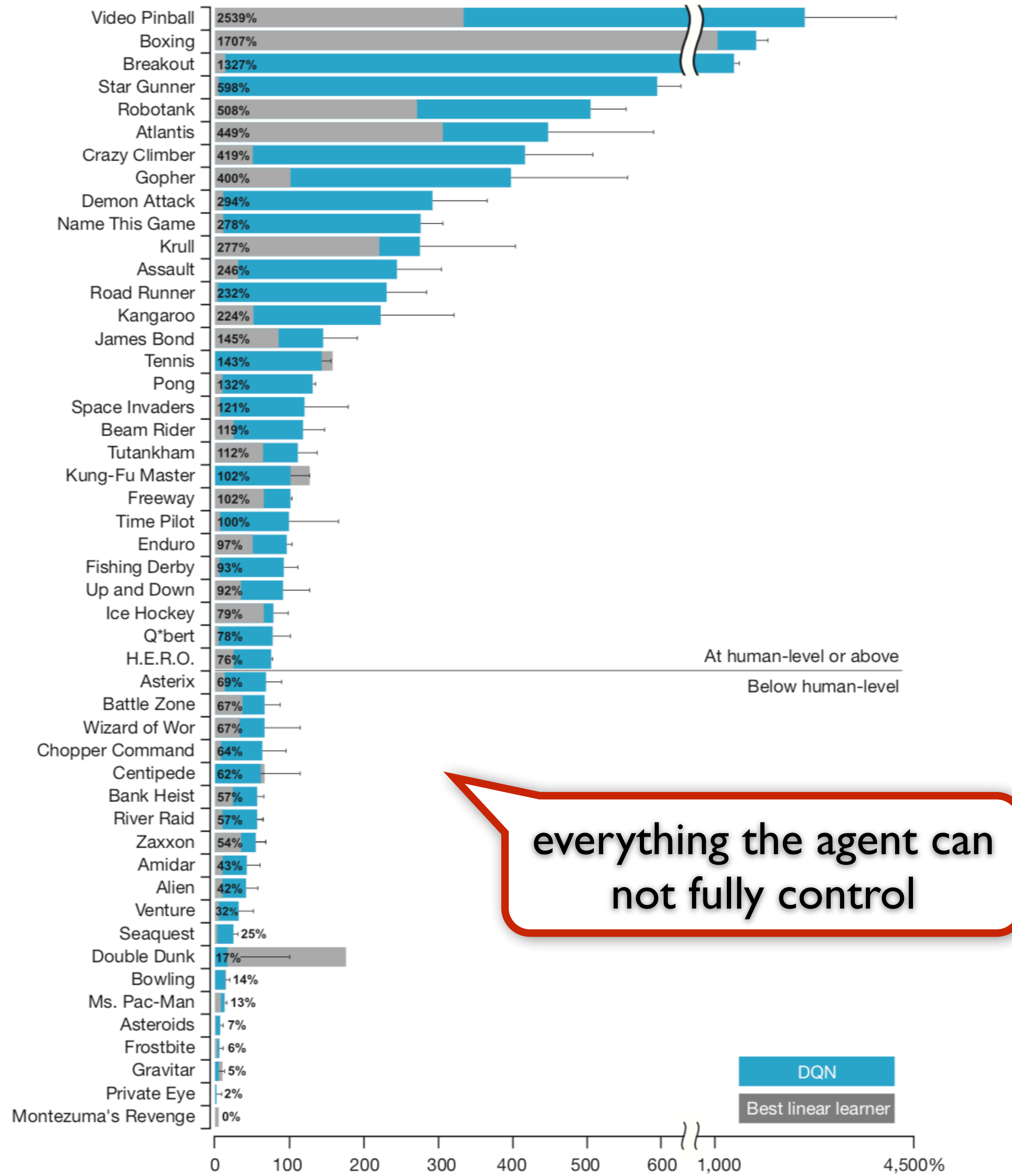




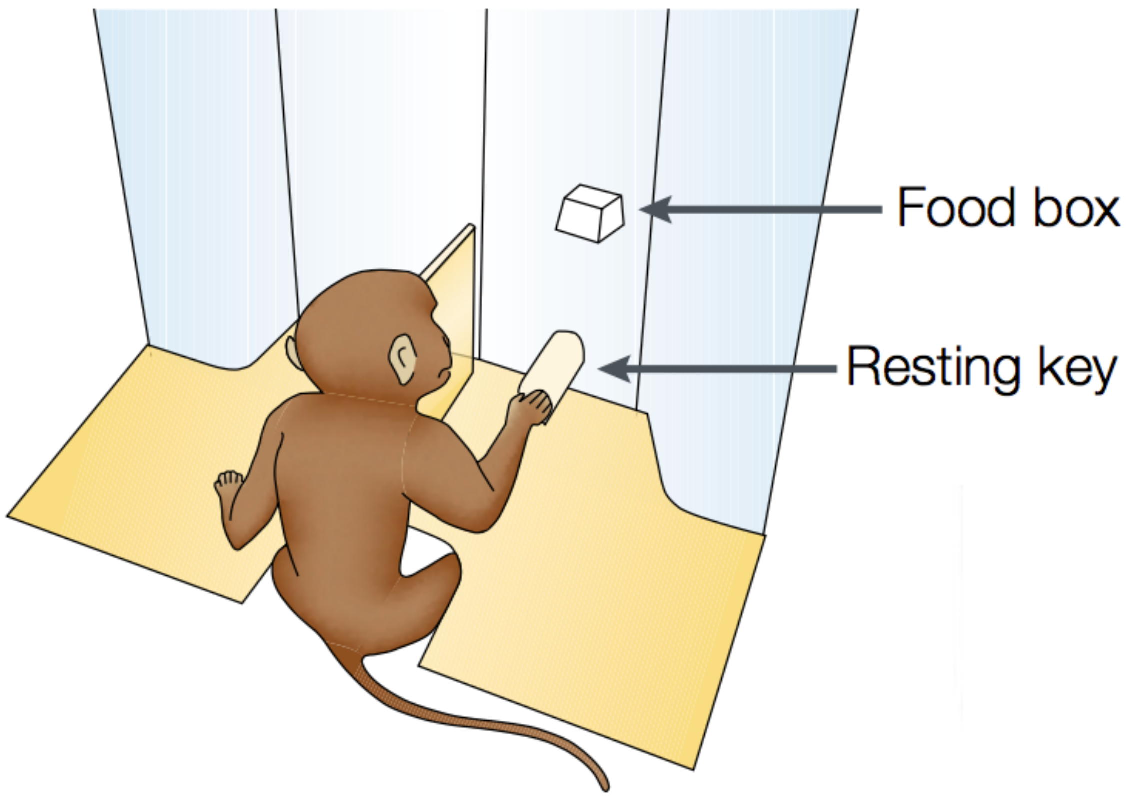
Decision making with a neural network and TD learning Deep Q learning

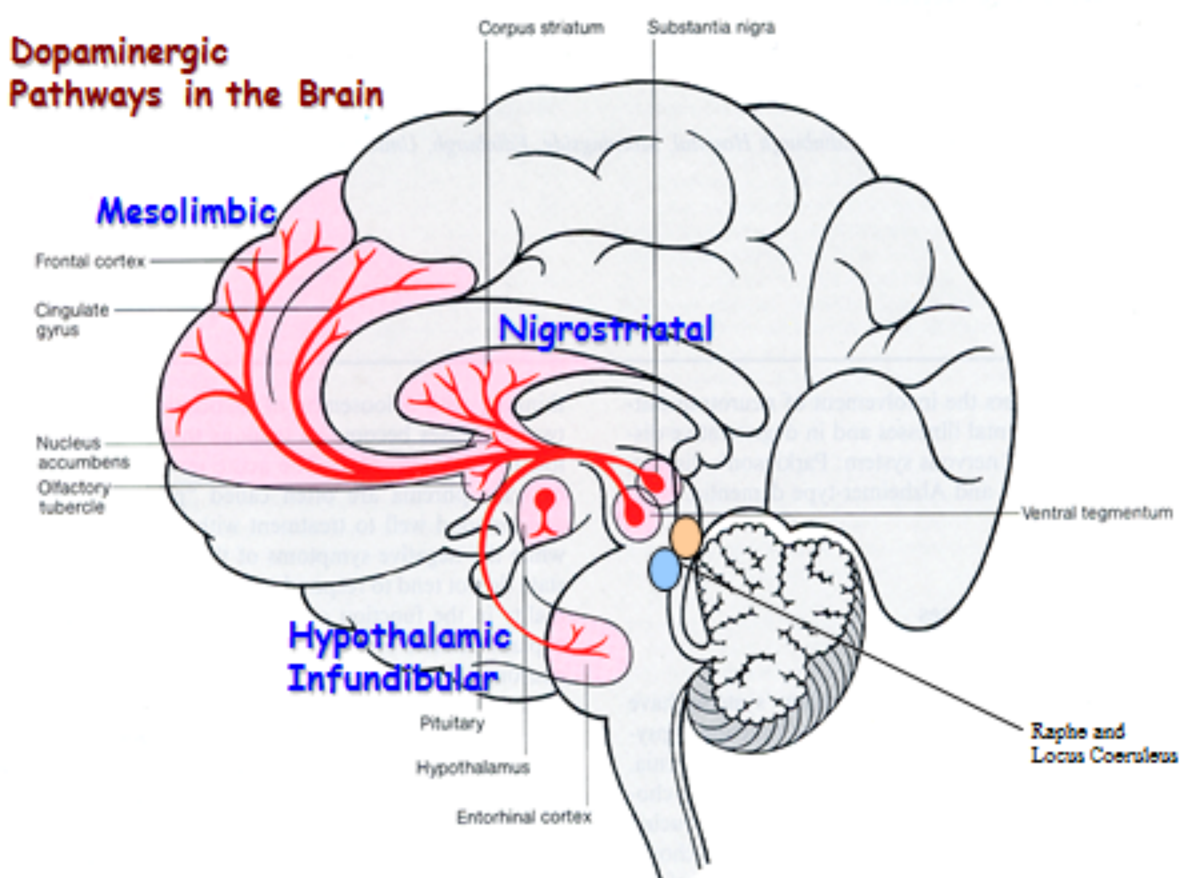
Atari games (Mnih et al., 2015)

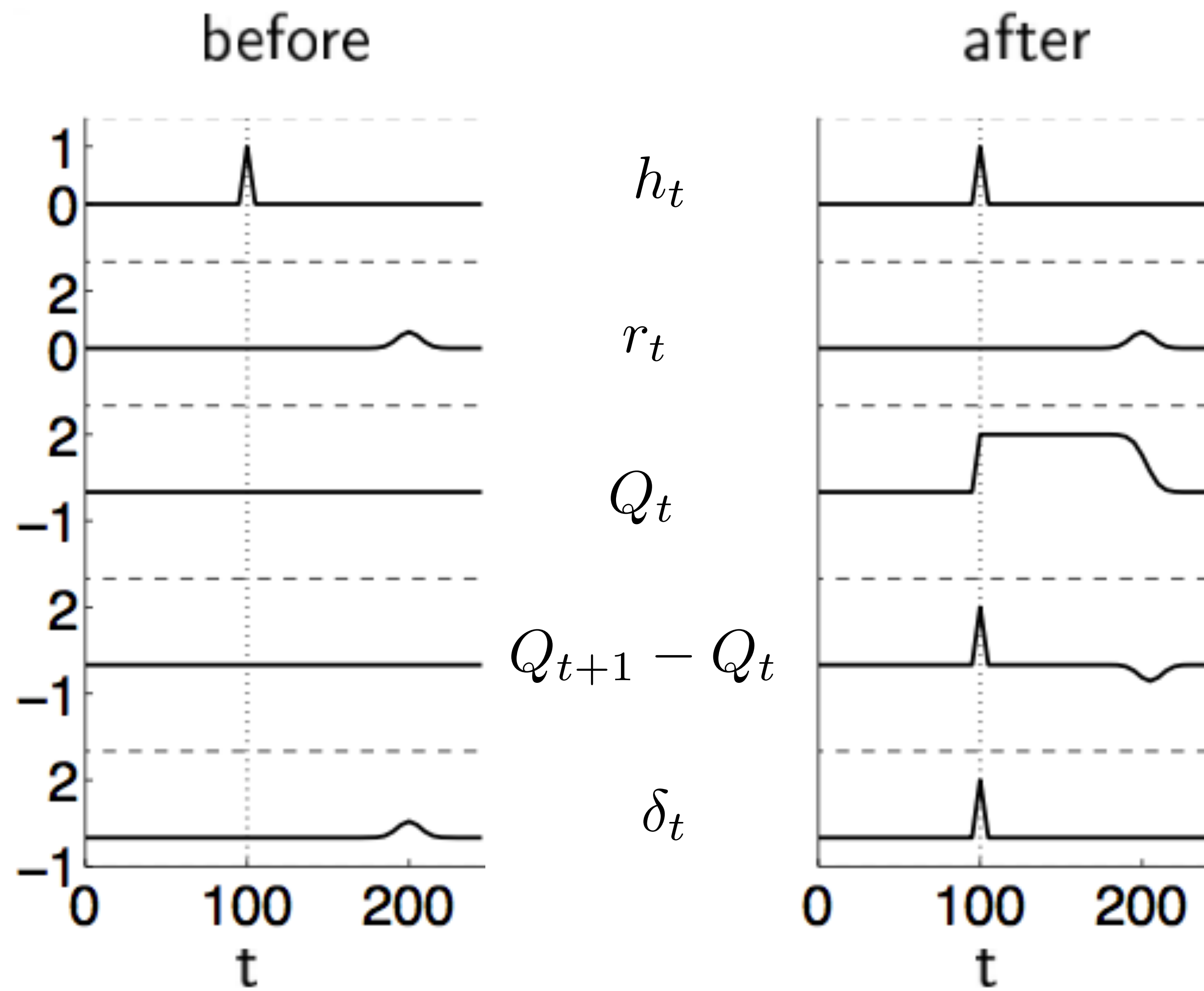




Neural representation: dopamine signal

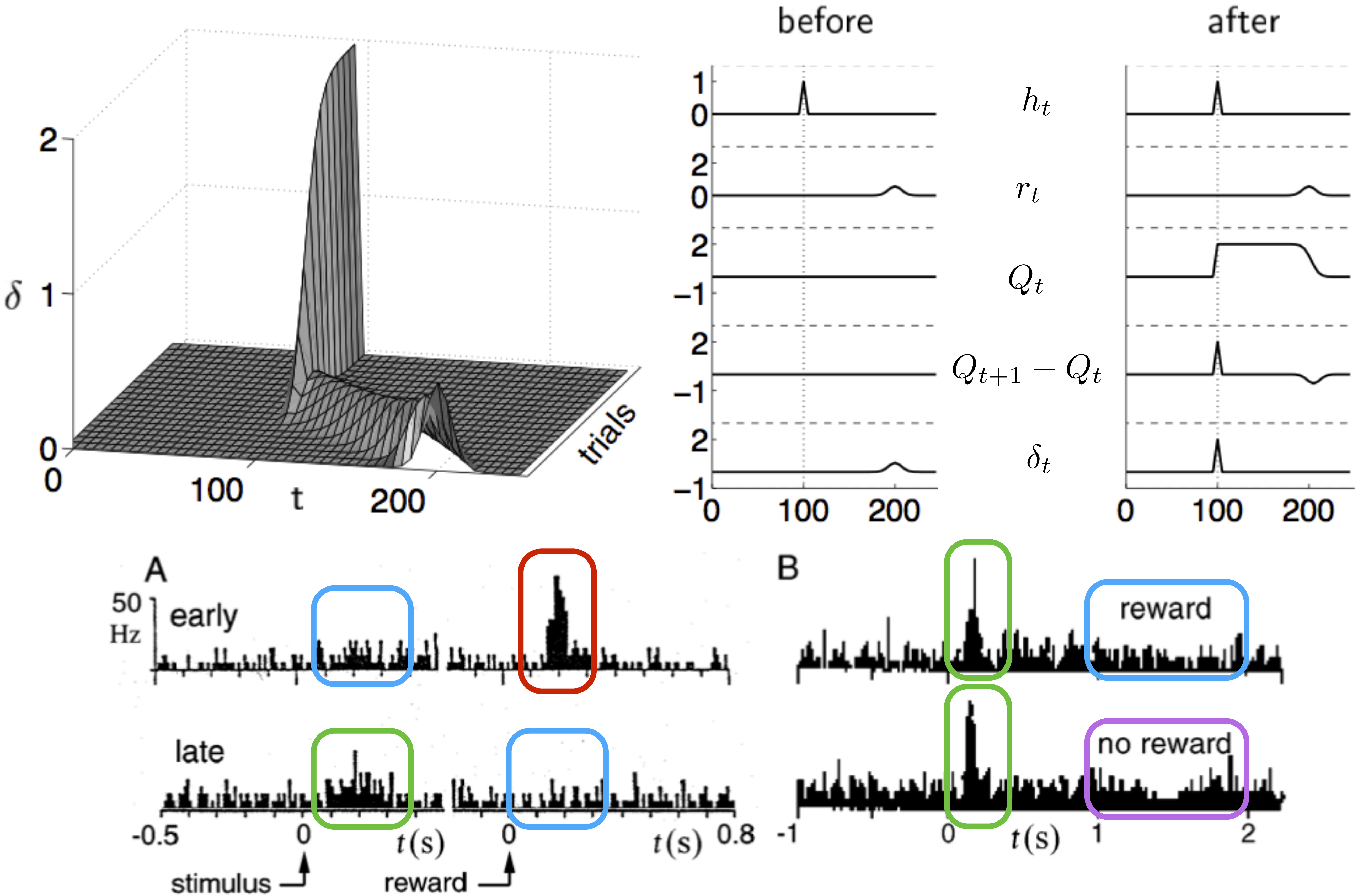






$$\alpha\Big[\underbrace{r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)}_{\delta_t}\Big]$$

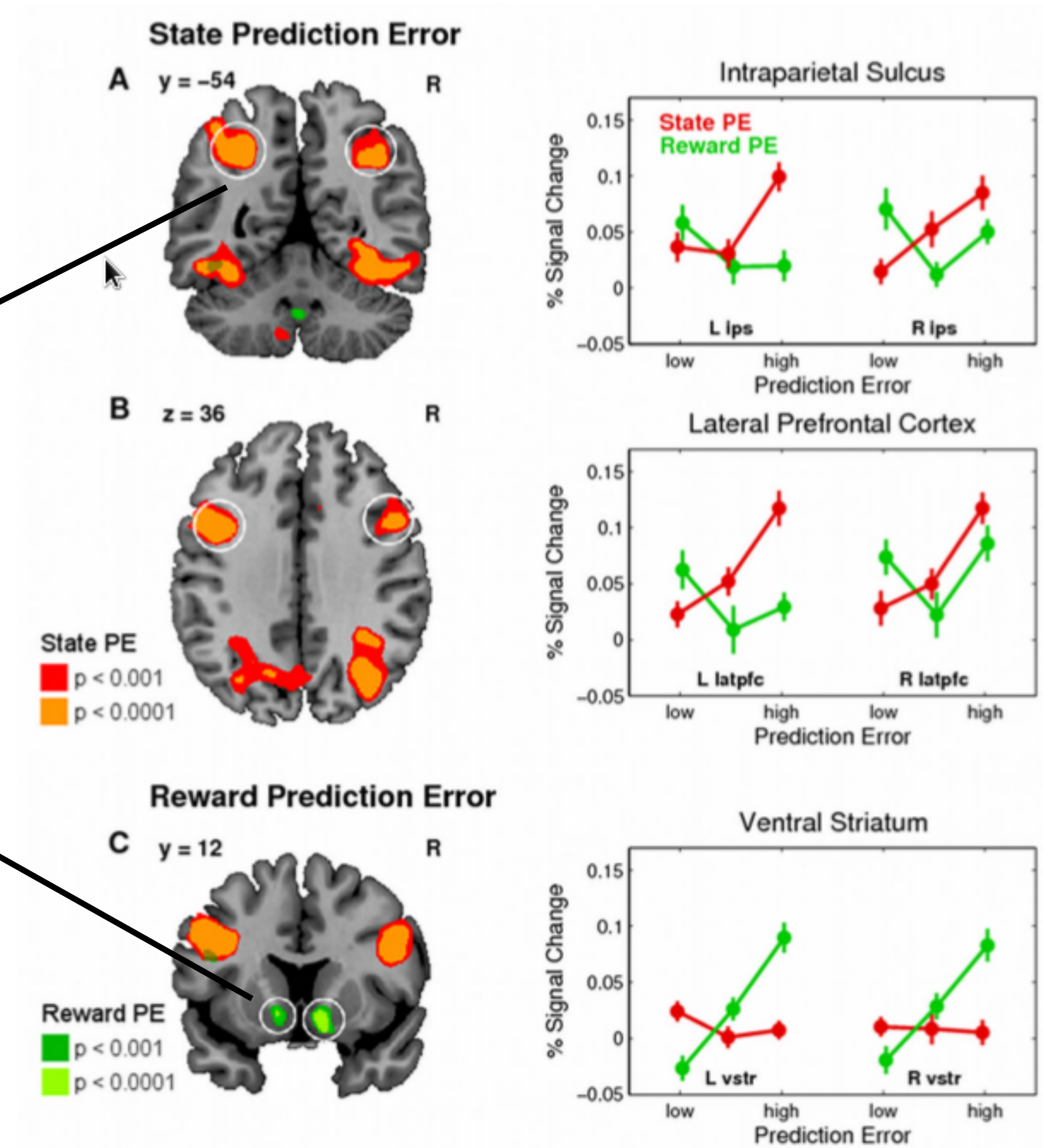
Neural representation: dopamine signal



Model-based RL in the brain

Deliberative, model-based RL
prefrontal and parietal cortices

Reactive, model-free RL
subcortical structures



Conclusions

Conclusions

- reinforcement learning:

Conclusions

- reinforcement learning:
  - learning from interaction with the environment

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:
    - midbrain dopamine signals prediction error (PE)

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:
    - midbrain dopamine signals prediction error (PE)
    - relevant for TD learning

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:
    - midbrain dopamine signals prediction error (PE)
    - relevant for TD learning
    - PE is communicated to many brain regions

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:
    - midbrain dopamine signals prediction error (PE)
    - relevant for TD learning
    - PE is communicated to many brain regions
    - error based learning is widespread in the brain

Conclusions

- reinforcement learning:
  - learning from interaction with the environment
- **computation**: learning by interaction
- **algorithm**:
  - TD-learning
  - model-based
- **implementation**:
  - TD-learning:
    - midbrain dopamine signals prediction error (PE)
    - relevant for TD learning
    - PE is communicated to many brain regions
    - error based learning is widespread in the brain