Reinforcement Learning: From neural processes modelling to Robotics applications

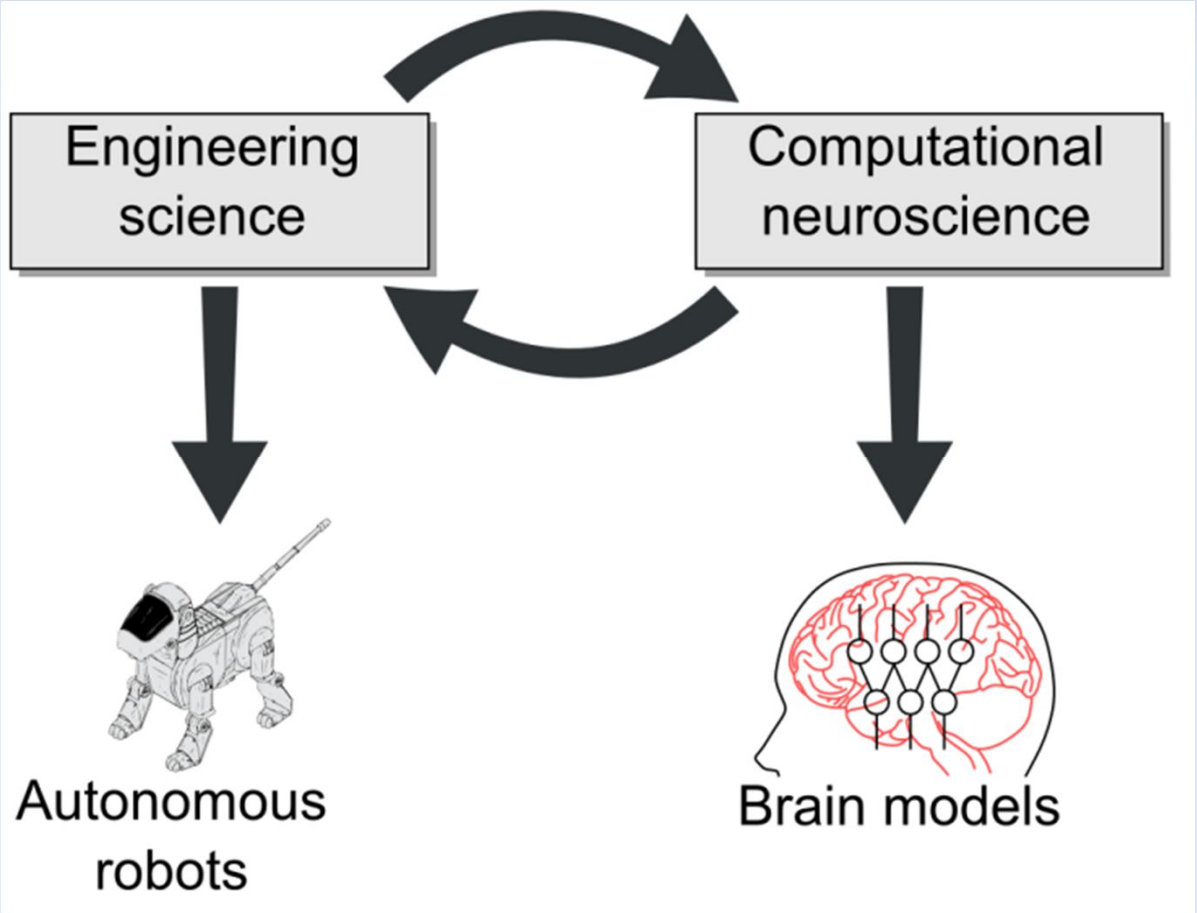
Mehdi Khamassi (CNRS, ISIR-UPMC, Paris)

30 January 2015

Michèle Sebag's course @ Univ. Orsay

Team: Architectures and Models of Adaptation and Cognition (AMAC)

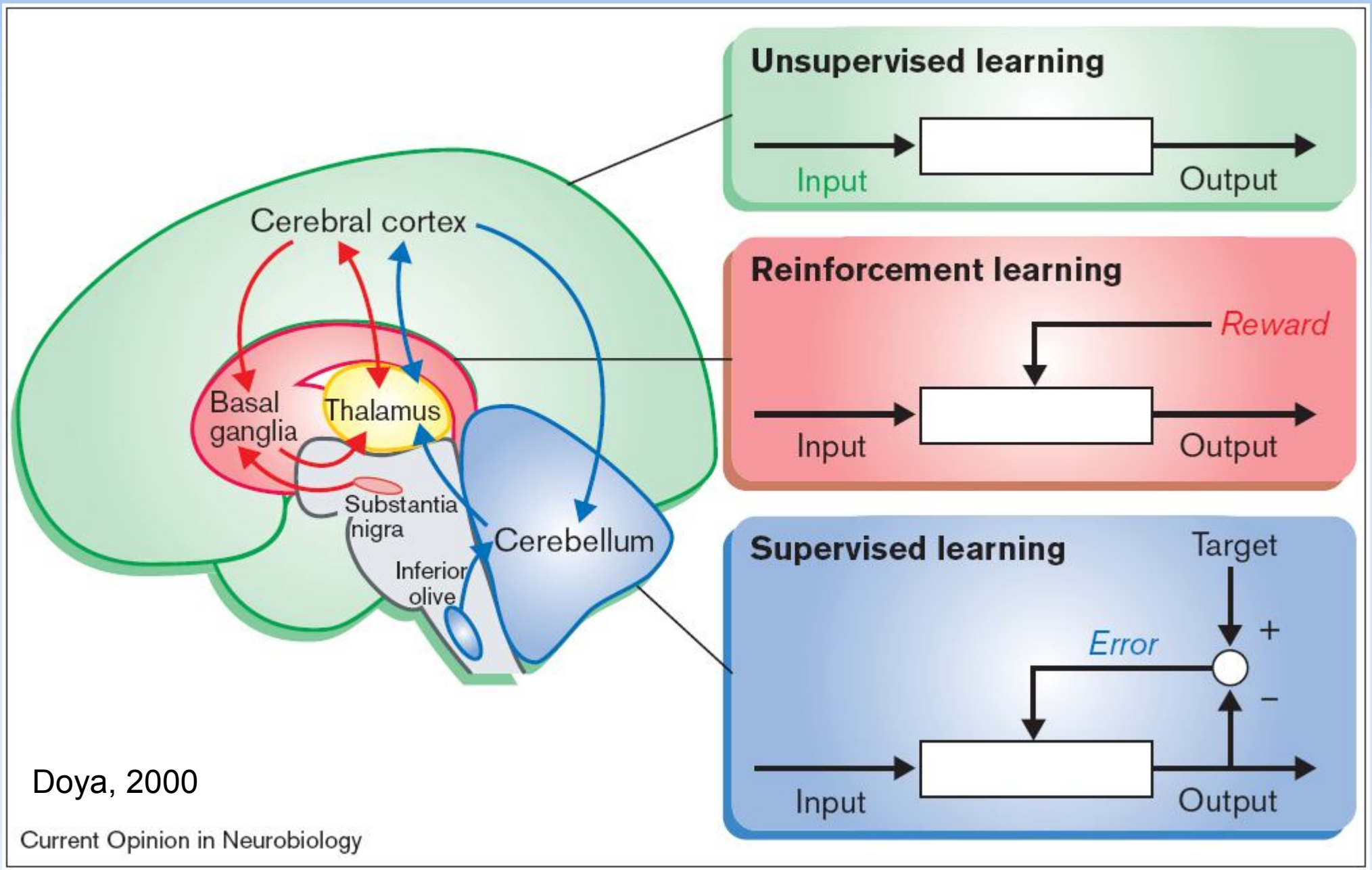
@ The Institute of Intelligent Systems and Robotics (ISIR), UPMC-CNRS



RESEARCH INTERESTS

- **Decision-making**: choice at each moment of the most appropriate behavior for an agent's survival, to solve a task.
- **Reinforcement Learning** (by trial/error): adaptation of this choice to maximize a particular reward function.
- **Complex problems**: noise, partial representation of states, non stationarity of the environment.
- **Modular/hierarchical structure** of different learning levels, enabling a better flexibility and autonomy of decision in animals and robots.

Global organization of learning in the brain (according to Doya 2000)



Doya, 2000

Current Opinion in Neurobiology

Outline

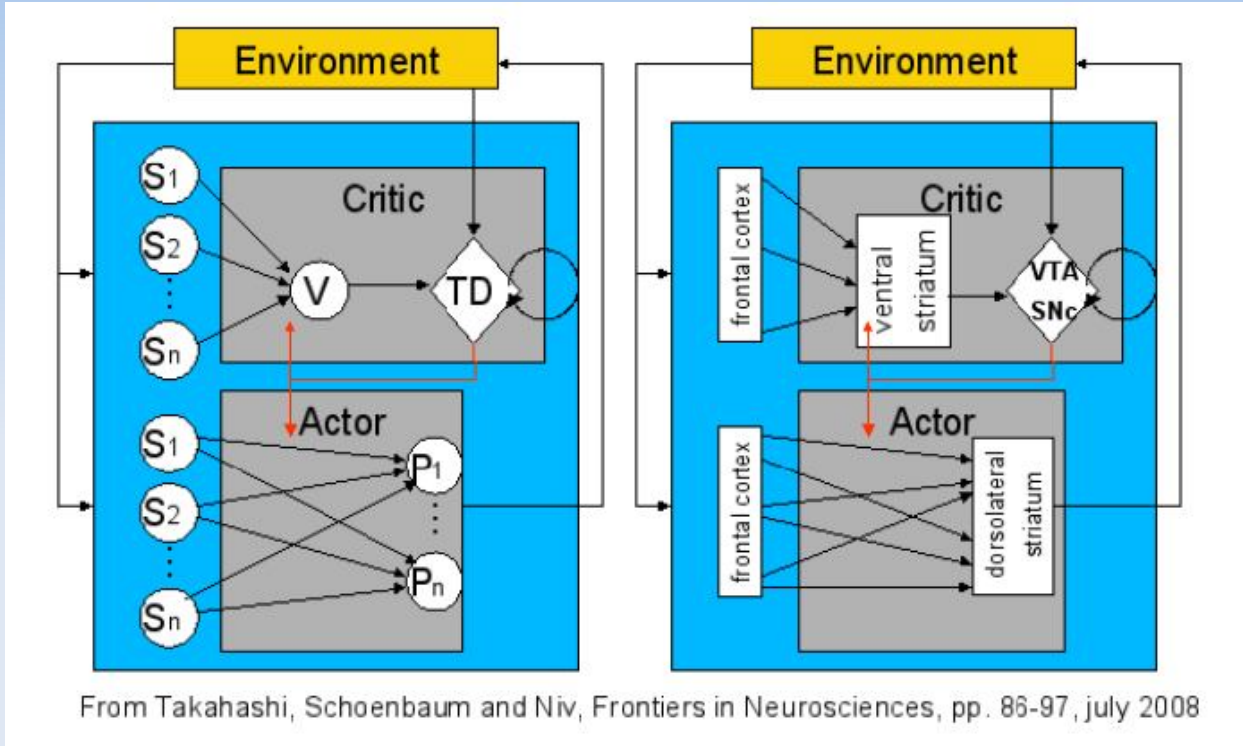
1. Model-free Reinforcement Learning
   - Temporal-Difference RL Algorithm
   - Dopamine activity
   - Wide application to Neuroscience of decision-making

2. Model-based Reinforcement Learning
   - Off-line learning / Replay during sleep
   - Dual-system RL
   - Online parameters tuning (meta-learning)
   - Link with Neurobehavioral data
   - Applications to Robotics

REINFORCEMENT LEARNING & DOPAMINE ACTIVITY

THE ACTOR-CRITIC MODEL

Sutton & Barto (1998) Reinforcement Learning: An Introduction



From Takahashi, Schoenbaum and Niv, Frontiers in Neurosciences, pp. 86-97, july 2008

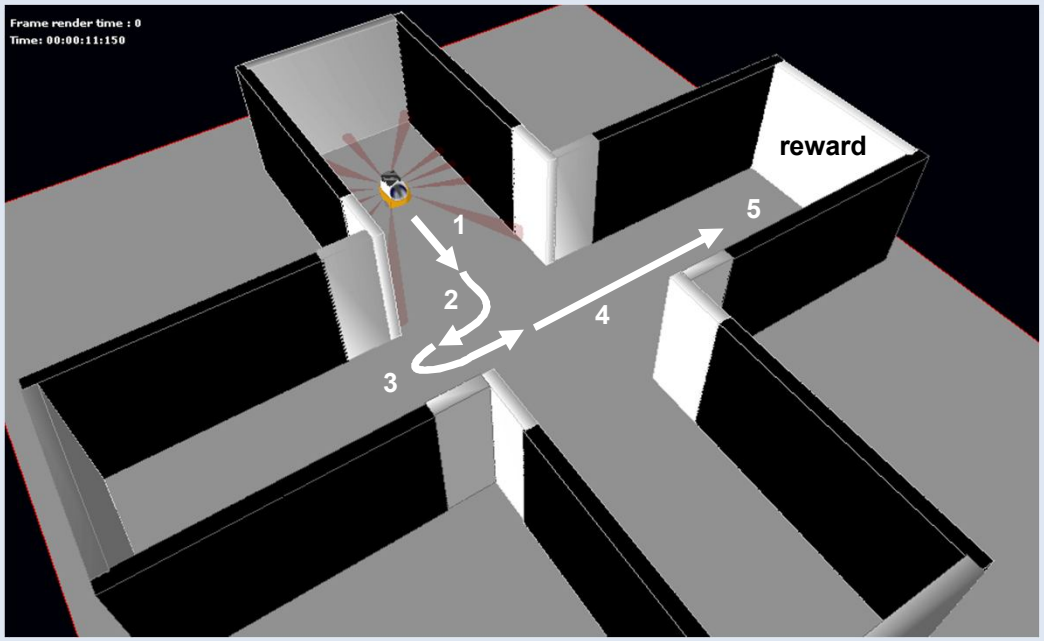
The Actor learns to select actions that maximize reward.

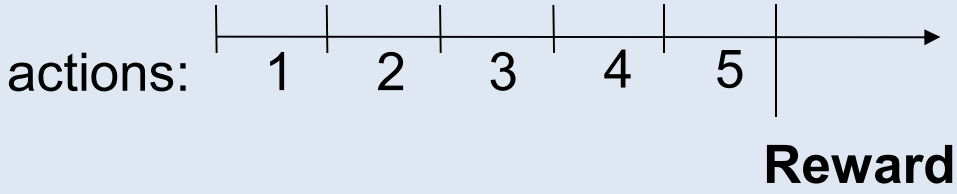
The Critic learns to predict reward (its value V).

A reward prediction error constitutes the reinforcement signal.

REINFORCEMENT LEARNING

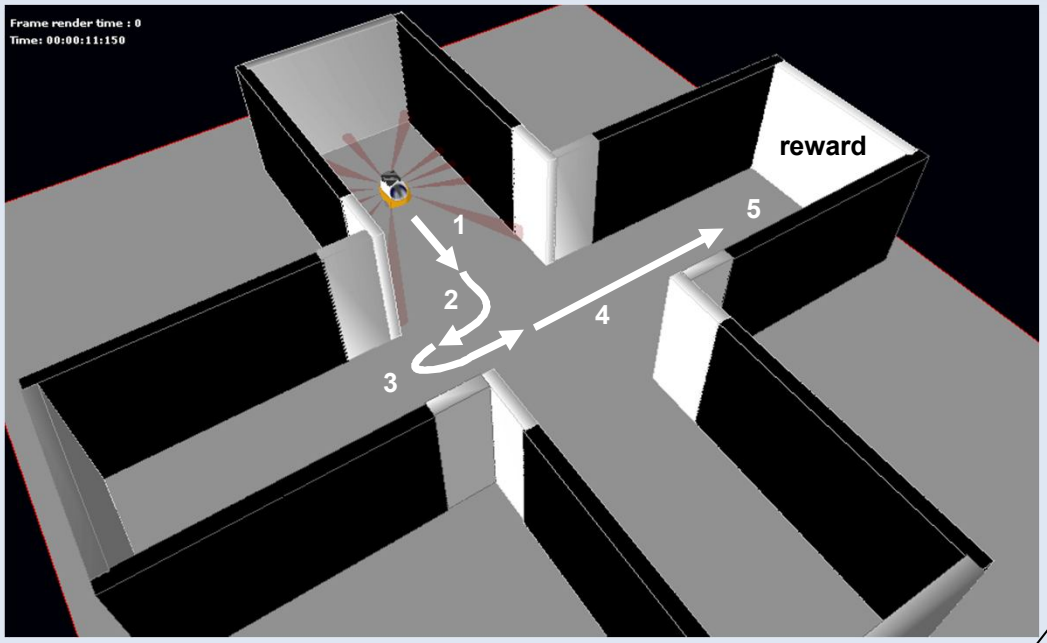
- Learning from delayed reward

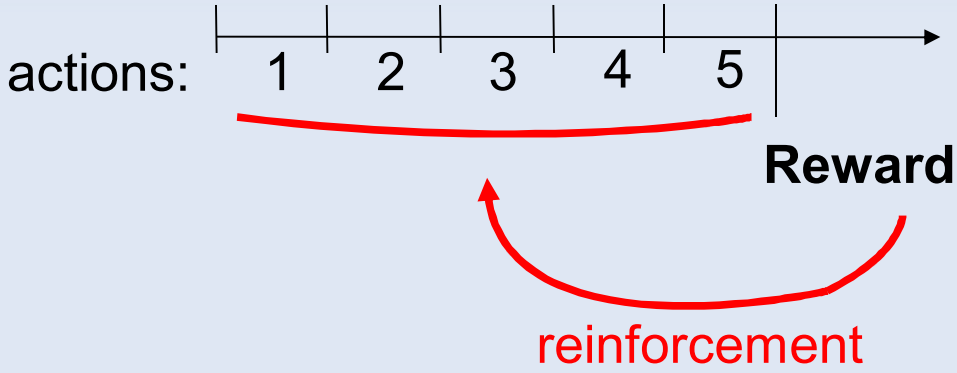




REINFORCEMENT LEARNING

- Learning from delayed reward





$$\delta_t = r_t$$

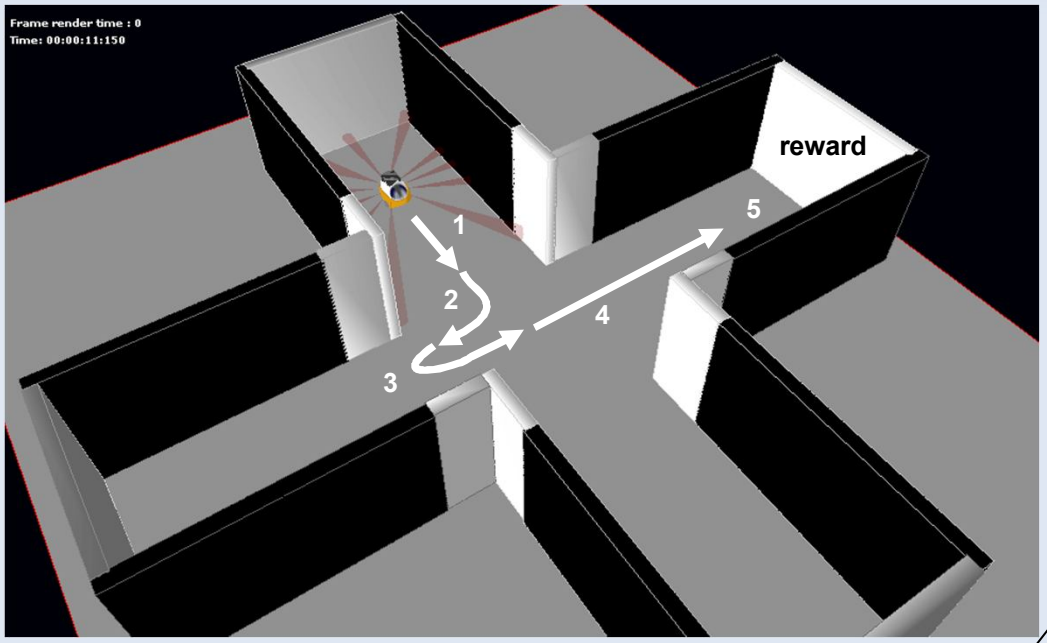
reinforcement

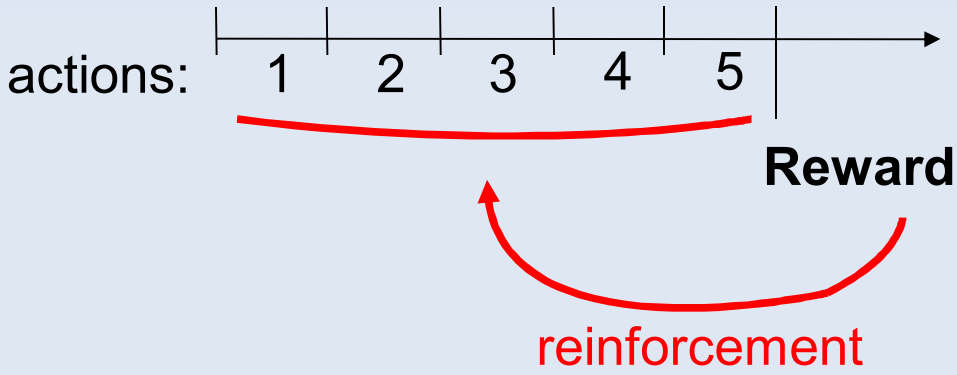
reward

REINFORCEMENT LEARNING

- Learning from delayed reward

Value estimation ("reward prediction"): $V(s_t)$





$$\delta_{t+n} = r_{t+n} - V(s_t)$$

reinforcement

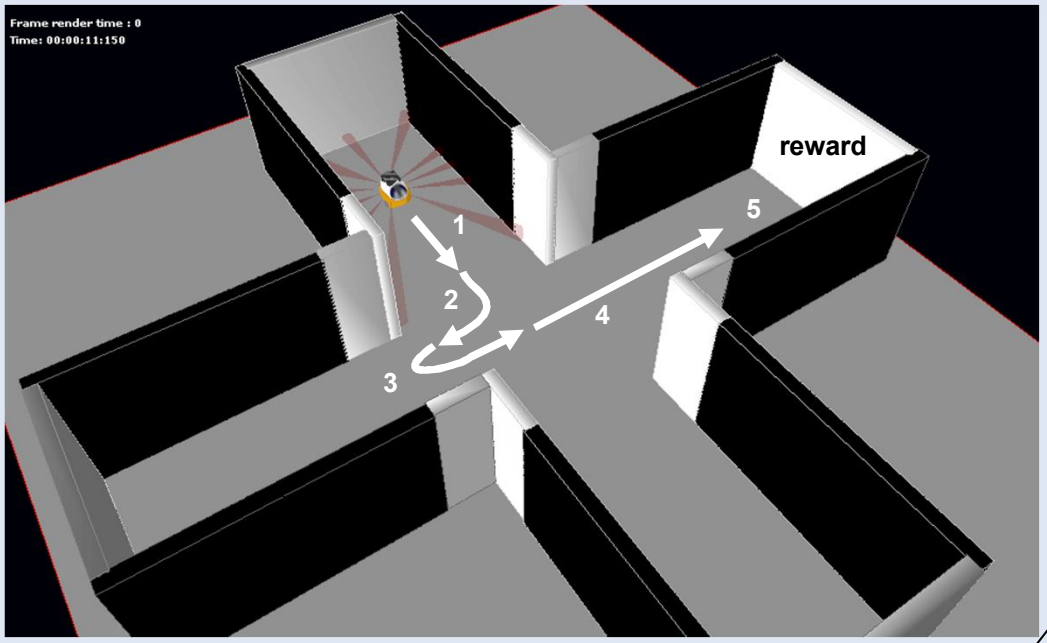
reward

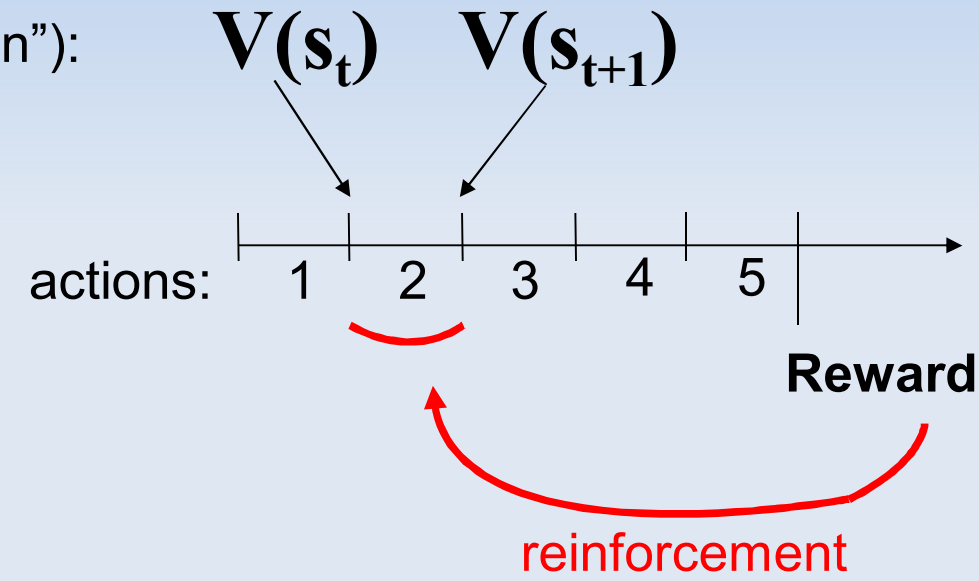
Rescorla and Wagner (1972).

REINFORCEMENT LEARNING

- Temporal-Difference (TD) learning

Value estimation ("reward prediction"):





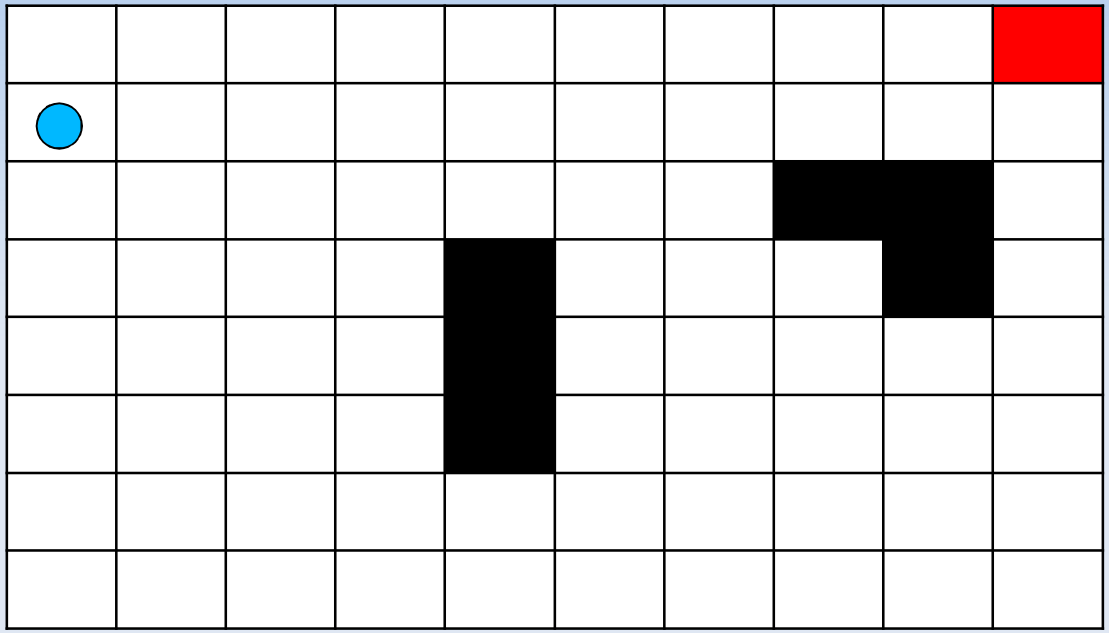
$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t) \qquad (\gamma < 1)$$

reinforcement

reward

Sutton and Barto (1998).

REINFORCEMENT LEARNING in a Markov Decision Process



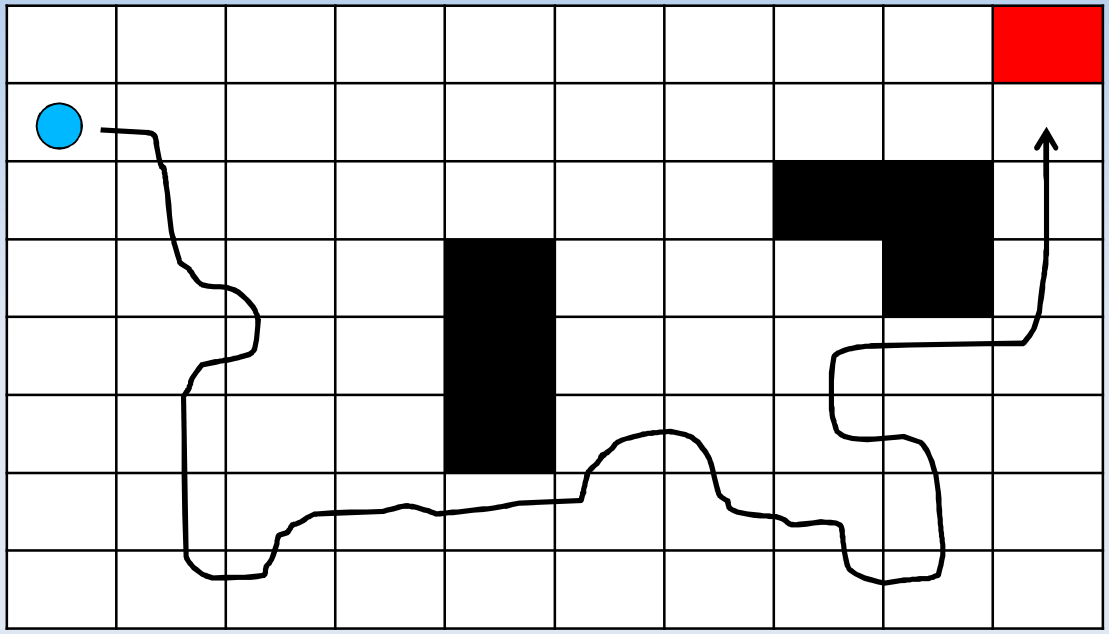
$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



0 = 0 + 0 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

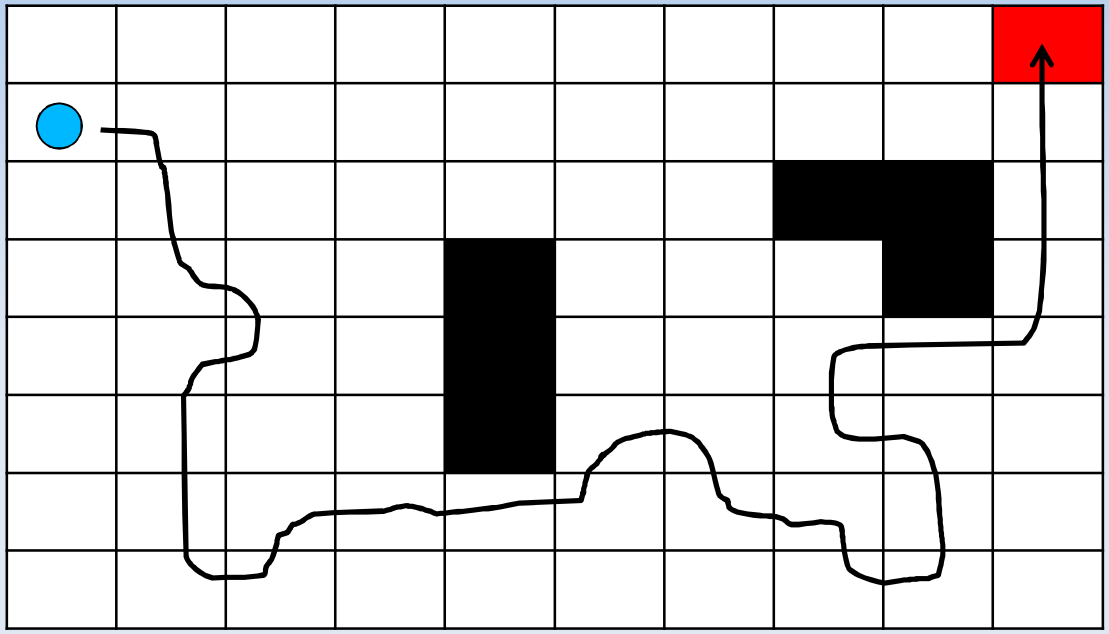
discount factor (=0.9)

0 = 0 + 0.9 * 0

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



1 = 1 + 0 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

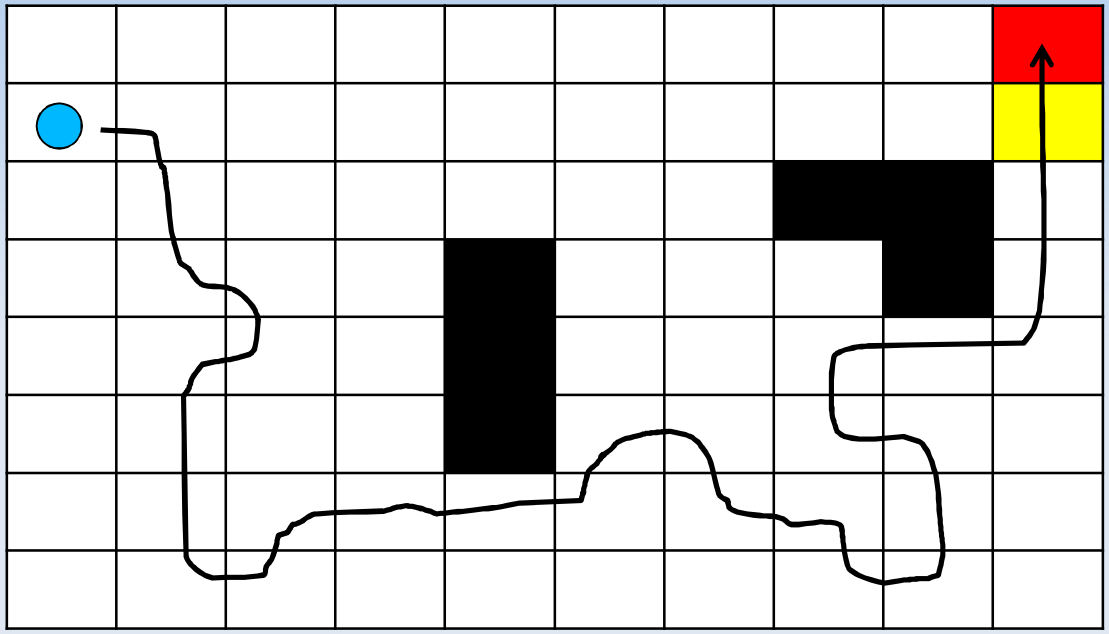
discount factor (=0.9)

0.9 = 0 + 0.9 * 1

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



Color indicates value

1 = 1 + 0 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

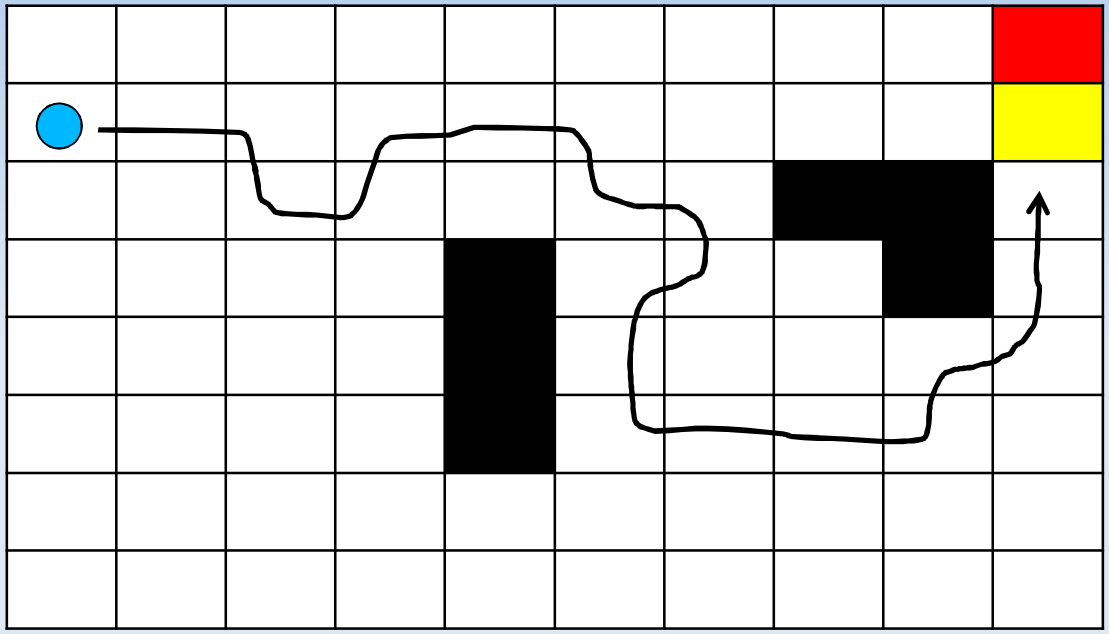
discount factor (=0.9)

0.9 = 0 + 0.9 * 1

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



0 = 0 + 0 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

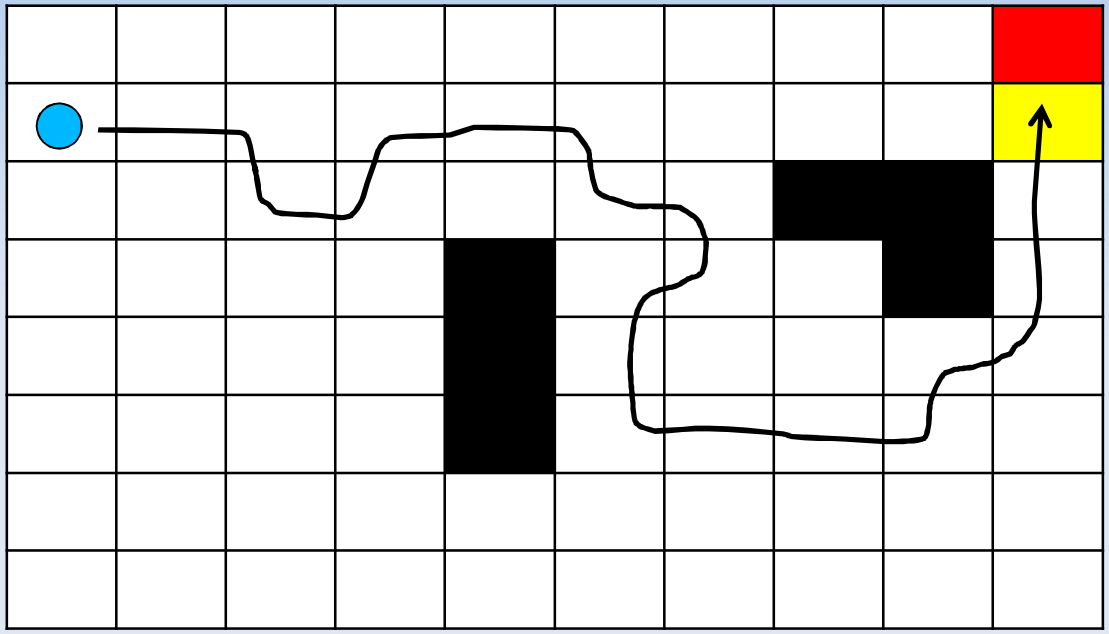
discount factor (=0.9)

0 = 0 + 0.9 * 0

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



0.81 = 0 + 0.9 * 0.9 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

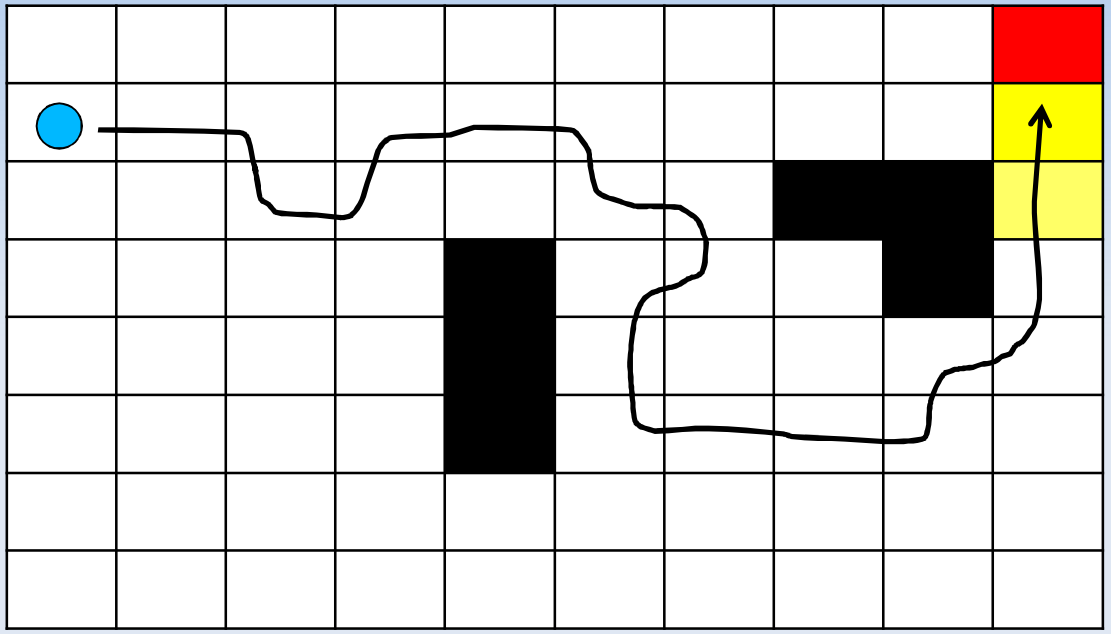
discount factor (=0.9)

0.72 = 0 + 0.9 * 0.81

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



Color indicates value

0.81 = 0 + 0.9 * 0.9 - 0

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

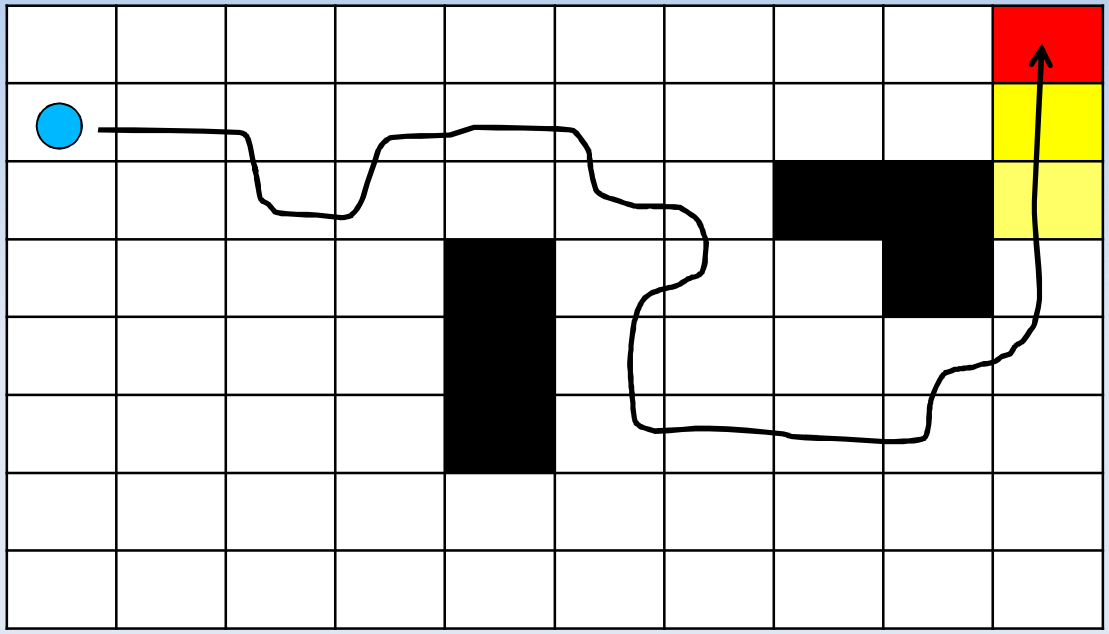
discount factor (=0.9)

0.72 = 0 + 0.9 * 0.81

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



0.1 = 1 + 0 - 0.9

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

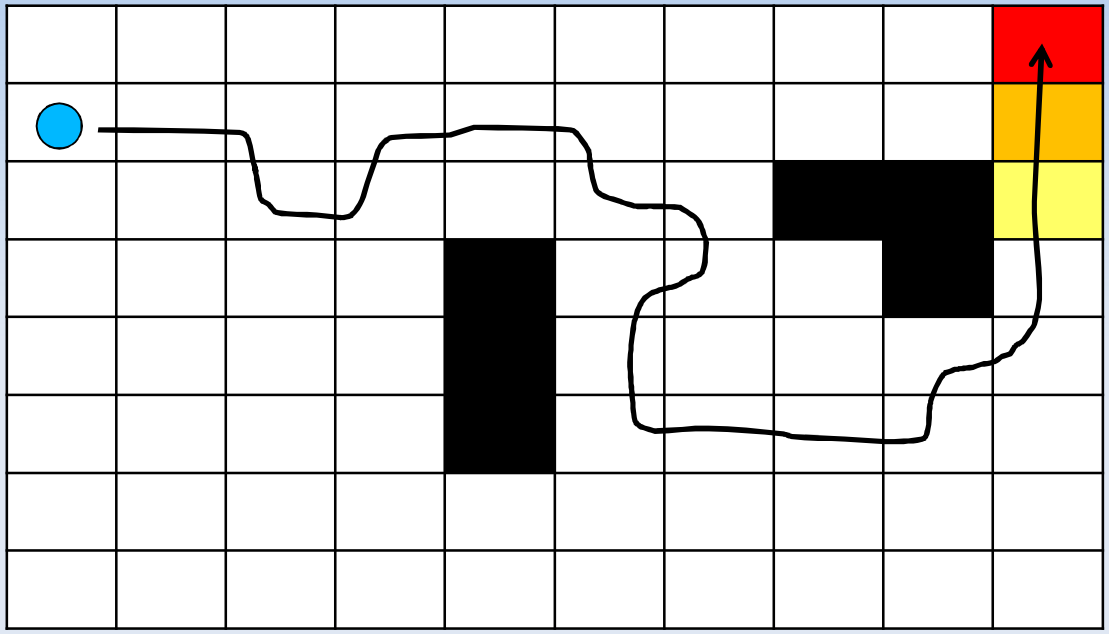
discount factor (=0.9)

0.99 = 0.9 + 0.9 * 0.1

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process



Color indicates value

0.1 = 1 + 0 - 0.9

$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

0.99 = 0.9 + 0.9 * 0.1

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

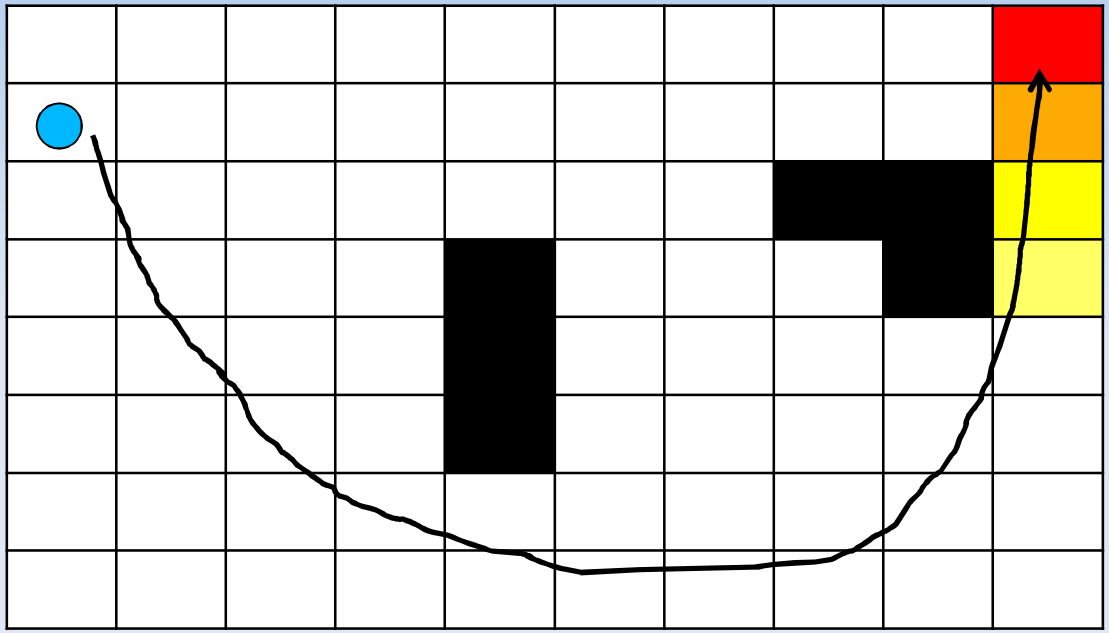
learning rate (=0.9)

REINFORCEMENT LEARNING in a Markov Decision Process

After

N simulations

Very long!



$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.1)

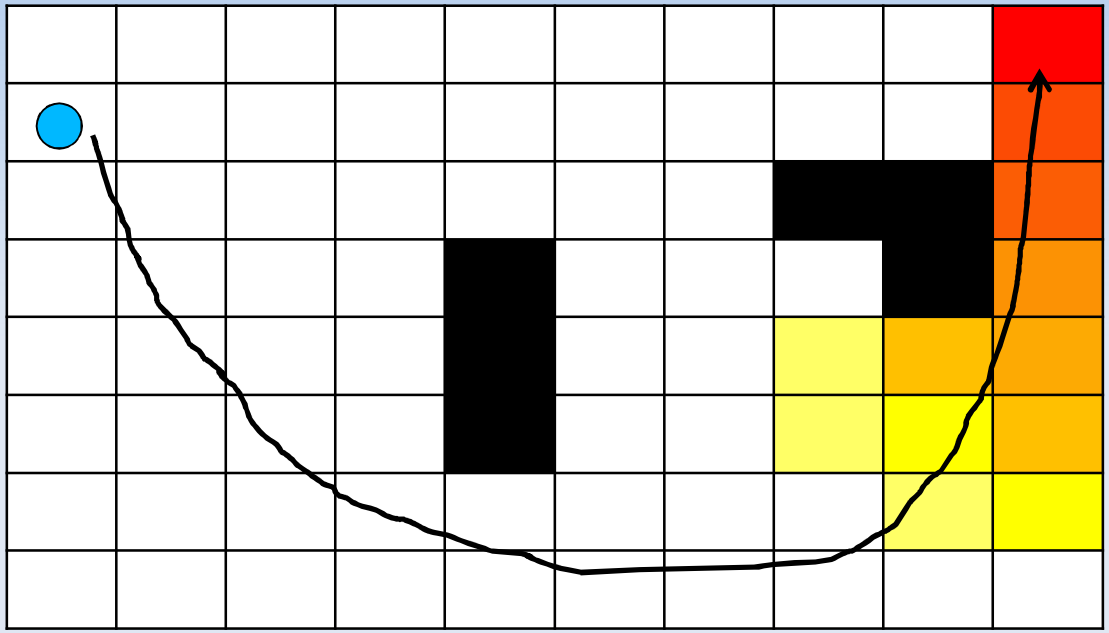
usually small for stability

REINFORCEMENT LEARNING in a Markov Decision Process

After

N simulations

Very long!



$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

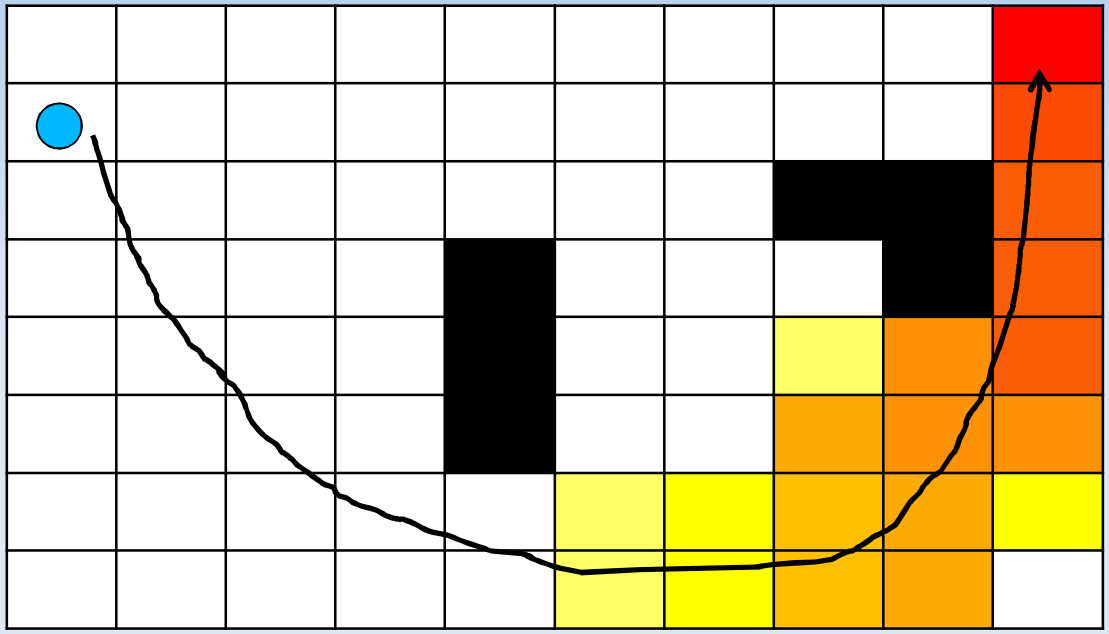
learning rate (=0.1)

REINFORCEMENT LEARNING in a Markov Decision Process

After

N simulations

Very long!



$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

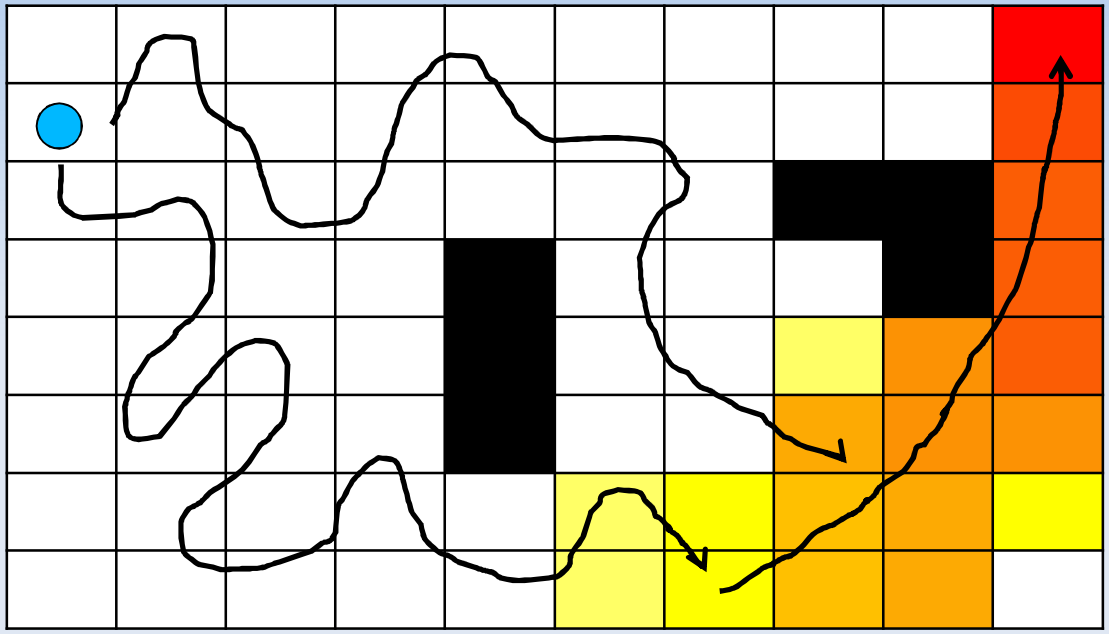
discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.1)

REINFORCEMENT LEARNING in a Markov Decision Process

May converge to a sub-optimal solution!



$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \cdot \delta_{t+1}$$

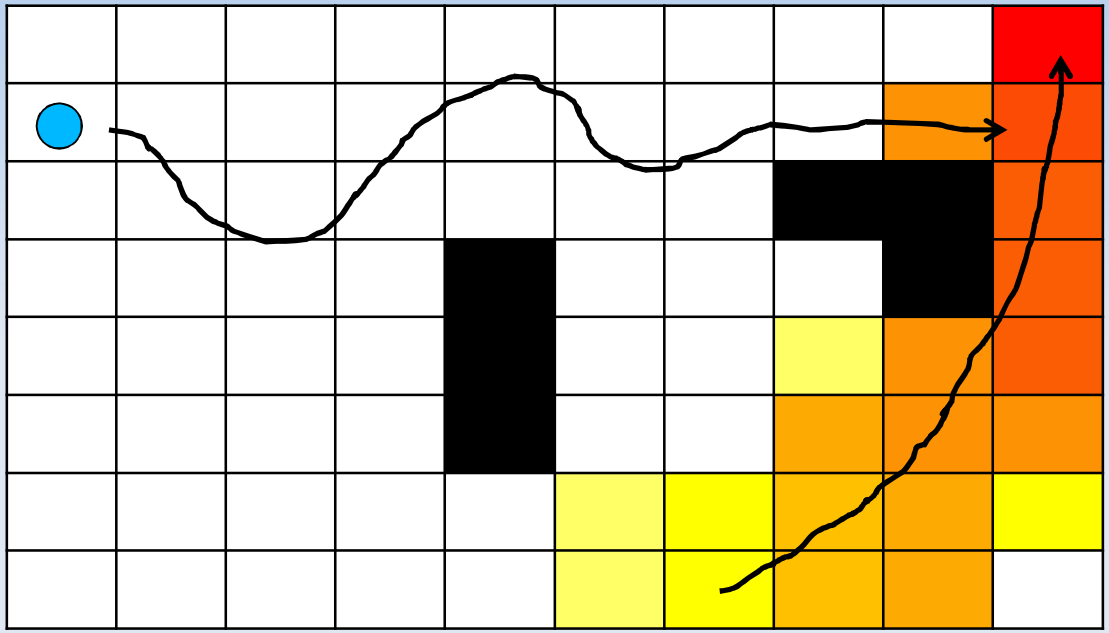
learning rate (=0.1)

REINFORCEMENT LEARNING in a Markov Decision Process

Exploration-

Exploitation

trade-off



$$\delta_{t+1} = r_{t+1} + \gamma \,.\, V(s_{t+1}) - V(s_t)$$

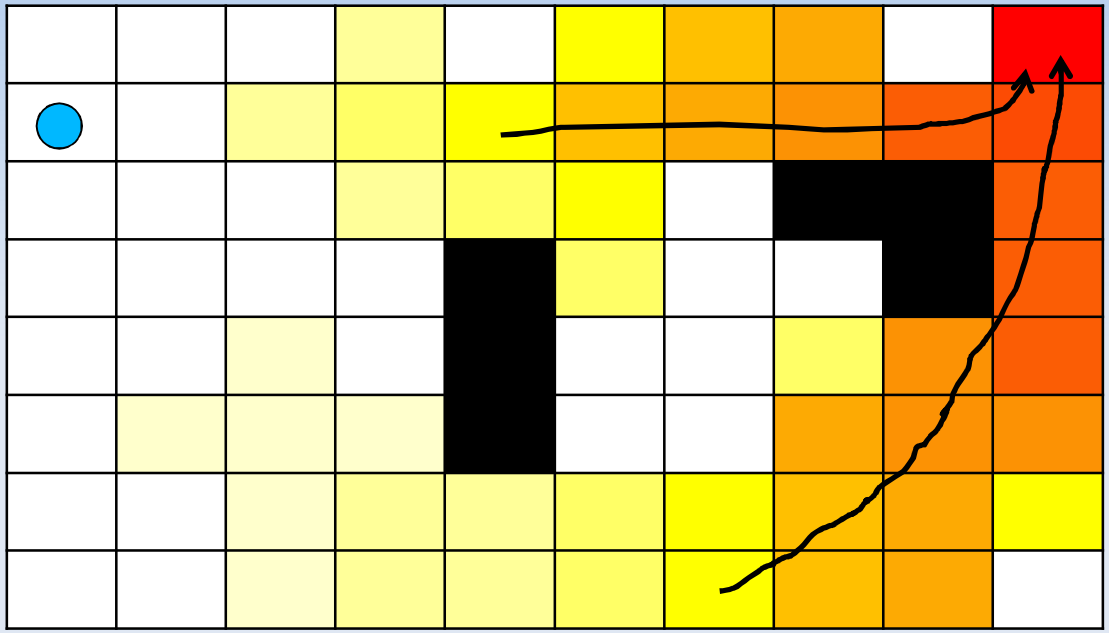
discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \,.\, \delta_{t+1}$$

learning rate (=0.1)

REINFORCEMENT LEARNING in a Markov Decision Process

Finds best solution after infinite time!



$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

discount factor (=0.9)

$$V(s_t) = V(s_t) + \alpha \cdot \delta_{t+1}$$

learning rate (=0.1)

How can the agent learn a policy?

How to learn to perform the right actions

How can the agent learn a policy?

How to learn to perform the right actions

$\boldsymbol{S}$: state space

$\boldsymbol{A}$: action space

Policy function $\boldsymbol{\pi} : S \longrightarrow A$

What we have learned so far:

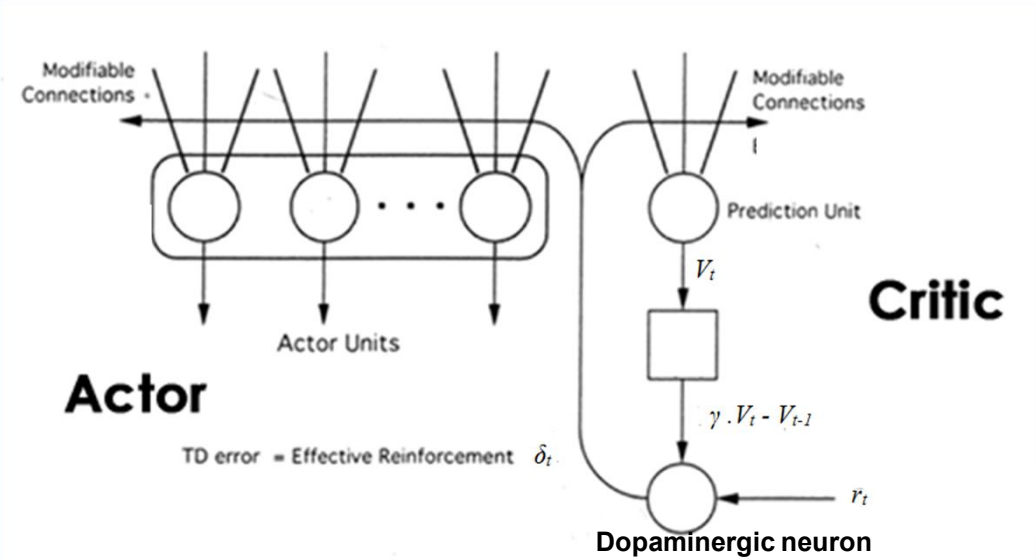
Value function $\boldsymbol{V} : S \longrightarrow \mathbb{R}$

The Actor-Critic model

How can the agent learn a policy?

How to learn to perform the right actions

a solution: parallely update a policy and a value function



$$P^{\pi}(a_t|s_t) = P^{\pi}(a_t|s_t) + \alpha \, . \, \delta_{t+1}$$

$$V(s_t) = V(s_t) + \alpha \, . \, \delta_{t+1}$$

The Q-learning model

How can the agent learn a policy?

How to learn to perform the right actions

other solution: learning Q-values (qualities)

$Q : (S,A) \longrightarrow \mathbb{R}$

Q-table:

state / action	a1 : North	a2 : South	a3 : East	a4 : West
s1	0.92	0.10	0.35	0.05
s2	0.25	0.52	0.43	0.37
s3	0.78	0.9	1.0	0.81
s4	0.0	1.0	0.9	0.9
...	...	...	...	...

The Q-learning model

How can the agent learn a policy?

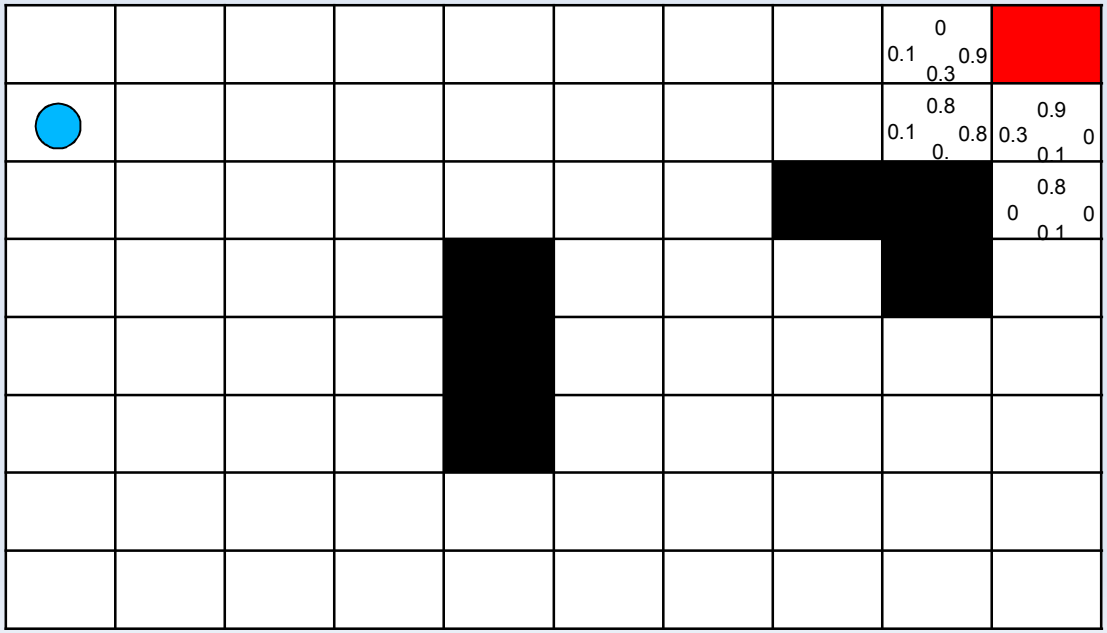
How to learn to perform the right actions

other solution: learning Q-values (qualities)

$Q : (S,A) \longrightarrow \mathbb{R}$

Q-table:

state / action	a1 : North	a2 : South	a3 : East	a4 : West
s1	0.92	0.10	0.35	0.05
s2	0.25	0.52	0.43	0.37
s3	0.78	0.9	1.0	0.81
s4	0.0	1.0	0.9	0.9
...	...	...	...	...



The Q-learning model

How can the agent learn a policy?

How to learn to perform the right actions

other solution: learning Q-values (qualities)

$Q : (S,A) \longrightarrow \mathbb{R}$

Q-table:

state / action	a1 : North	a2 : South	a3 : East	a4 : West
s1	0.92	0.10	0.35	0.05
s2	0.25	0.52	0.43	0.37
s3	0.78	0.9	1.0	0.81
s4	0.0	1.0	0.9	0.9
...	...	...	...	...

$$P(a) = \frac{\exp(\boldsymbol{\beta} \cdot Q(s,a))}{\Sigma_b \exp(\boldsymbol{\beta} \cdot Q(s,b))}$$

The $\boldsymbol{\beta}$ parameter regulates the exploration – exploitation trade-off.

Different Temporal-Difference (TD) methods

- ACTOR-CRITIC

$$V(s_t) \leftarrow V(s_t) + \alpha[\underbrace{r_{t+1} + \gamma V(s_{t+1}) - V(s_t)}]$$

State-dependent Reward Prediction Error

(independent from the action)

Different Temporal-Difference (TD) methods

- ACTOR-CRITIC

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

State-dependent Reward Prediction Error

(independent from the action)

Also used to update the ACTOR

$$P(a_t | s_t) \leftarrow P(a_t | s_t) + \alpha\, \delta_{t+1}$$

Different Temporal-Difference (TD) methods

- ACTOR-CRITIC

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

- SARSA

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

Reward Prediction Error dependent on the action chosen to be performed next

Different Temporal-Difference (TD) methods

- ACTOR-CRITIC

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

- SARSA

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

- Q-LEARNING

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[\underbrace{r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)}]$$

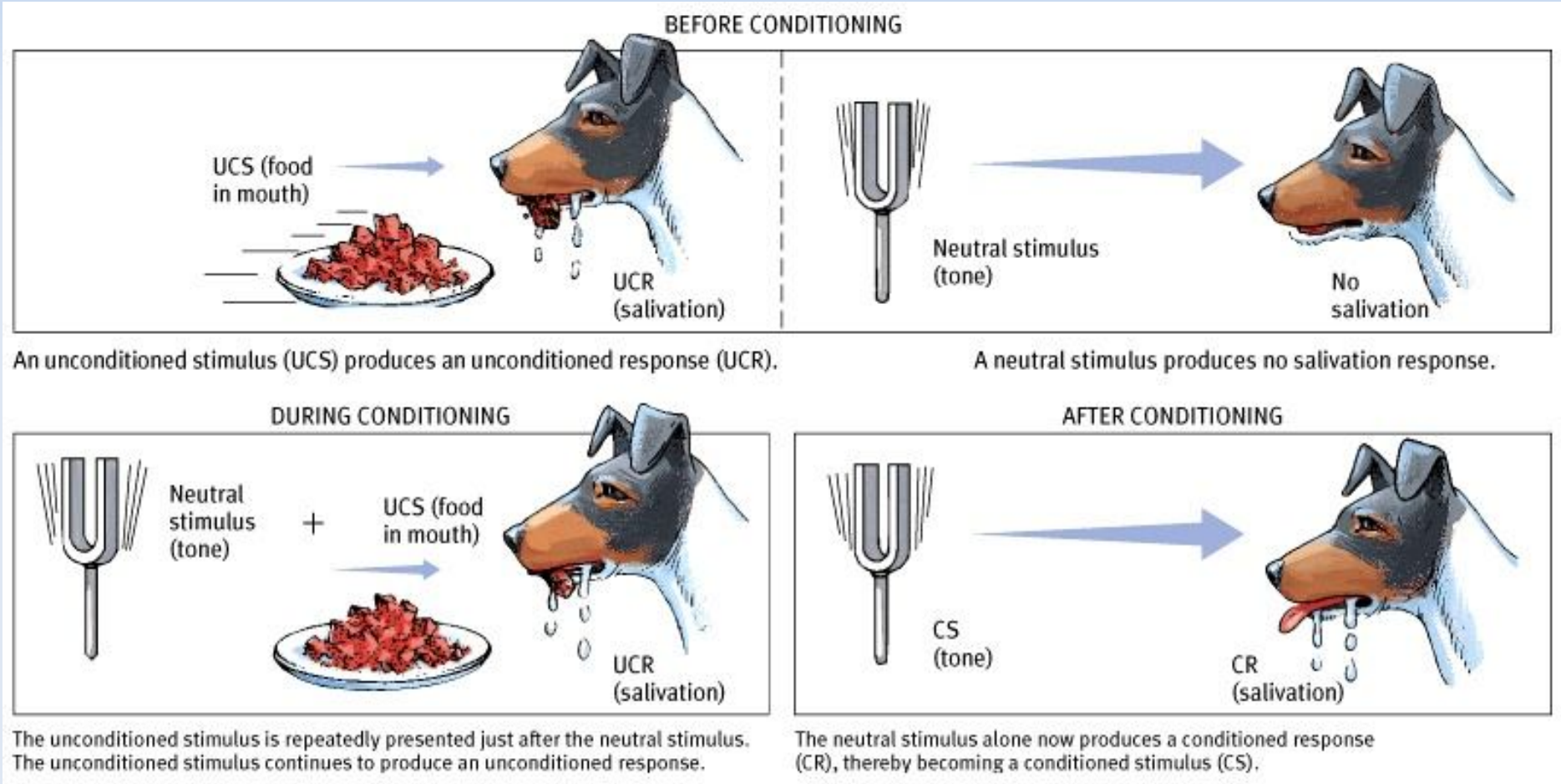
Reward Prediction Error dependent on the best action

Links with biology

Activity of dopaminergic neurons

CLASSICAL CONDITIONING

TD-learning explains classical conditioning (predictive learning)



Taken from Bernard Balleine's lecture at Okinawa Computational Neuroscience Course (2005).

REINFORCEMENT LEARNING

- Analogy with dopaminergic neurons' activity

Schultz et al. (1993);
Houk et al. (1995); Schultz et al. (1997).

$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

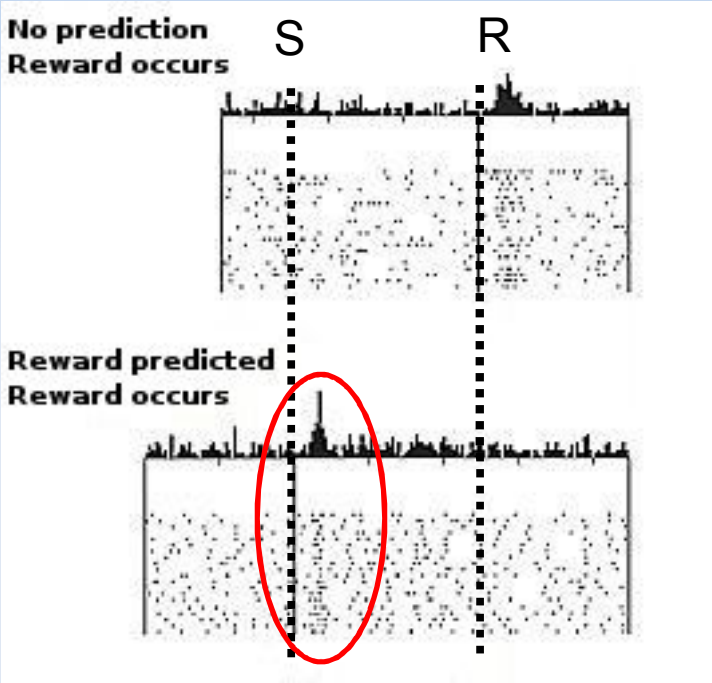
+1

reinforcement

reward

REINFORCEMENT LEARNING

- Analogy with dopaminergic neurons' activity



Schultz et al. (1993);
Houk et al. (1995); Schultz et al. (1997).

+1

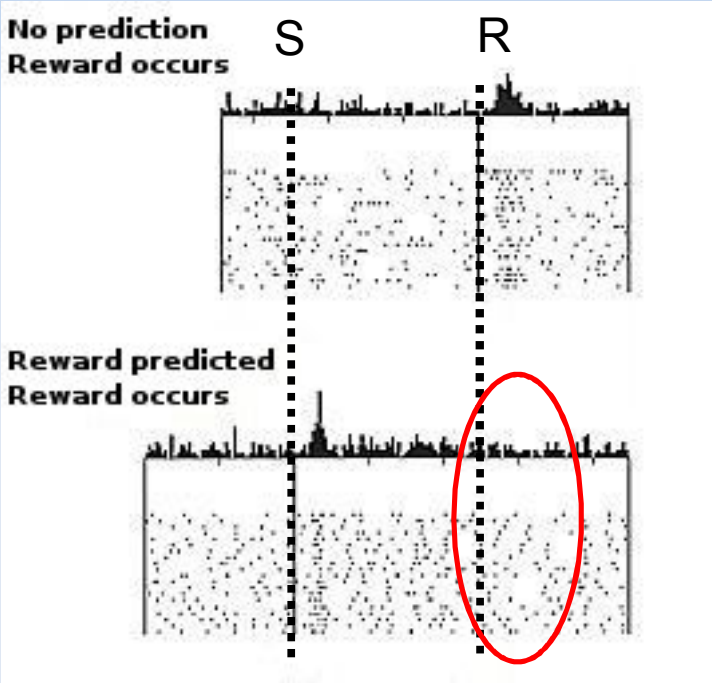
$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

reinforcement

reward

REINFORCEMENT LEARNING

- Analogy with dopaminergic neurons' activity



Schultz et al. (1993);
Houk et al. (1995); Schultz et al. (1997).

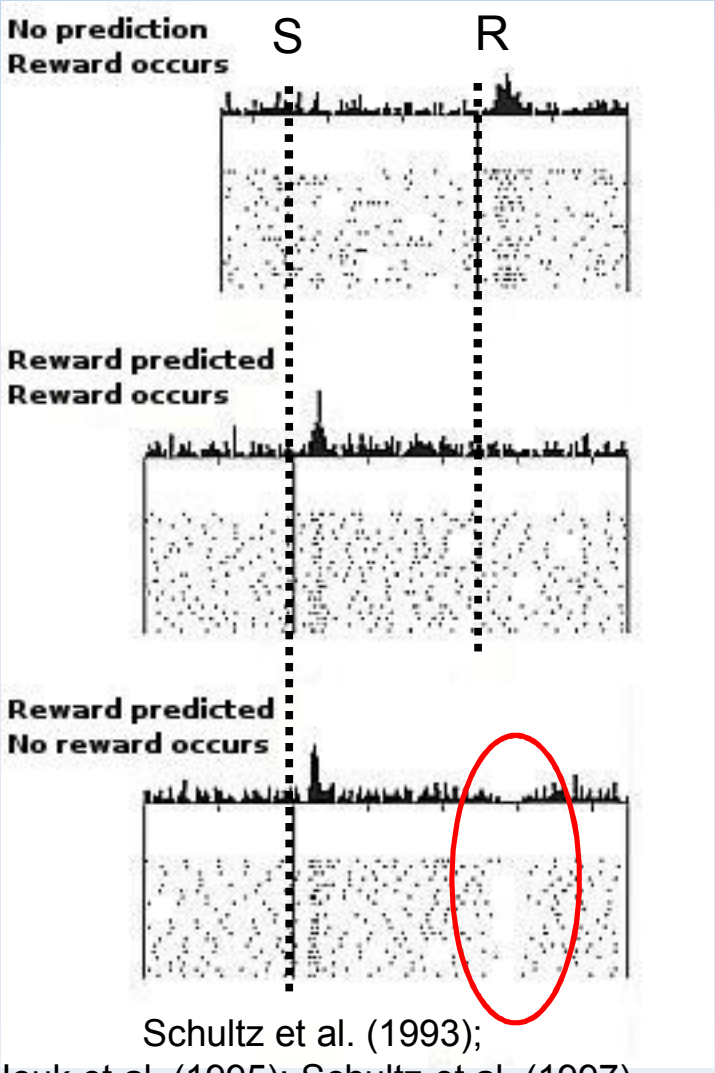
$$\delta_{t+1} = r_{t+1} + \gamma \cdot \overbrace{V(s_{t+1})}^{0} - V(s_t)$$

reinforcement

reward

REINFORCEMENT LEARNING

- Analogy with dopaminergic neurons' activity



Schultz et al. (1993);
Houk et al. (1995); Schultz et al. (1997).

$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

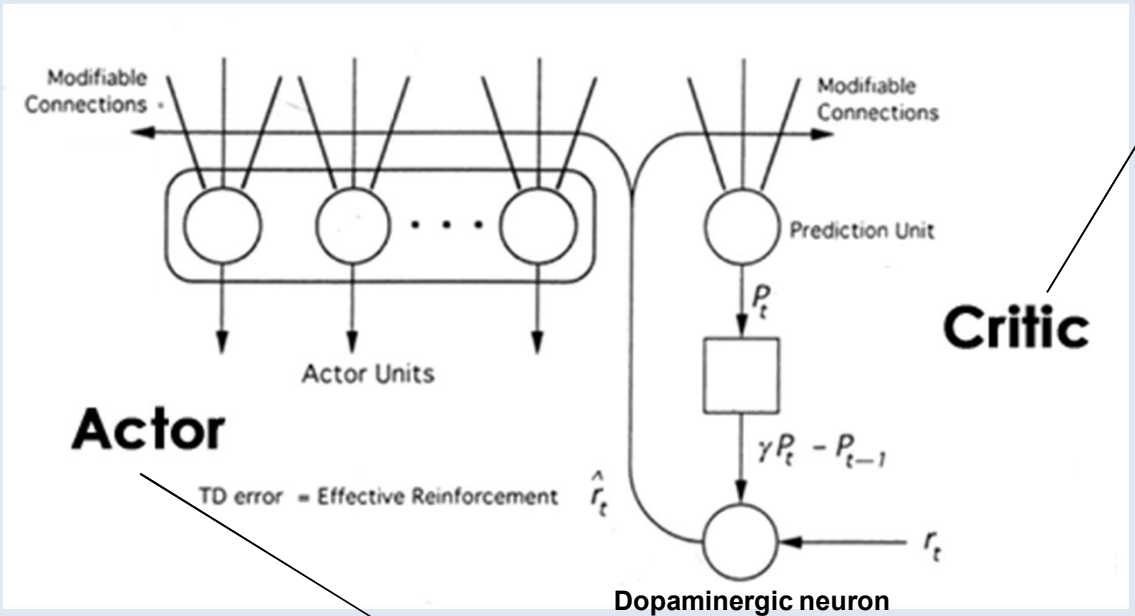
-1

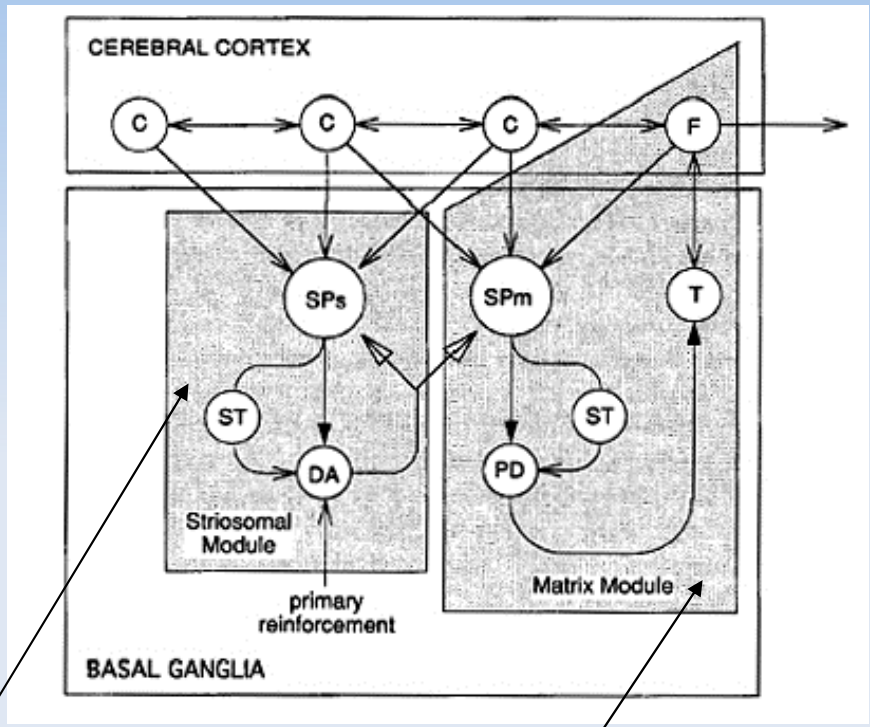
reinforcement

reward

The Actor-Critic model

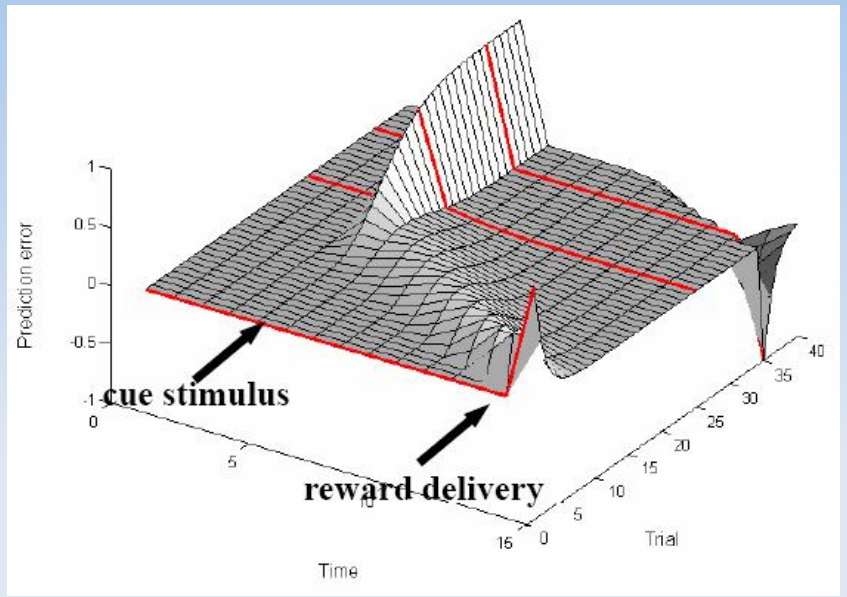
Barto (1995); Montague et al. (1996); Schultz et al. (1997); Berns and Sejnowski (1996); Suri and Schultz (1999); Doya (2000); Suri et al. (2001); Baldassarre (2002).
see Joel et al. (2002) for a review.





Houk et al. (1995)

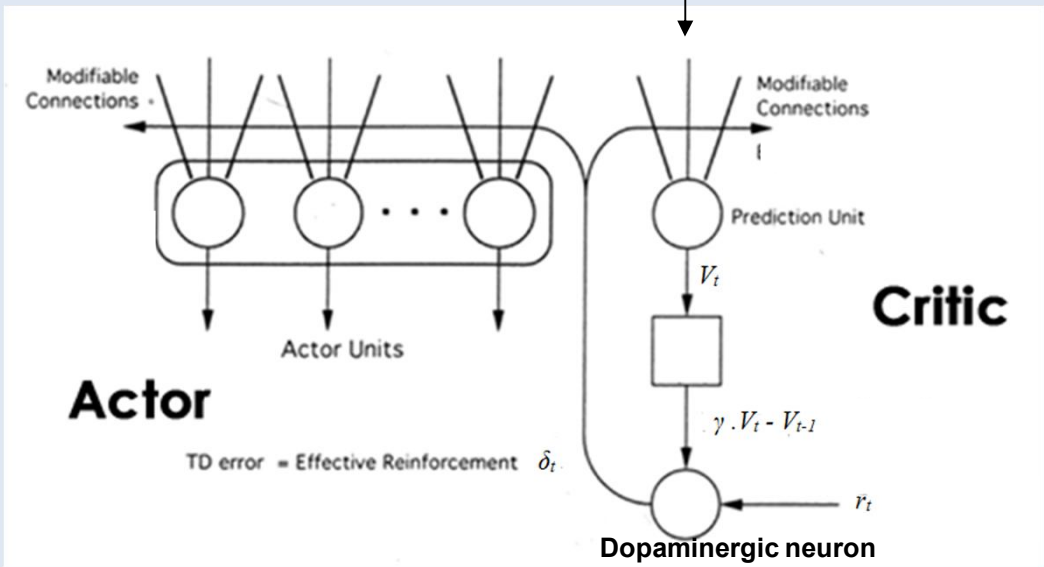
The Actor-Critic model



Which state space as an input?

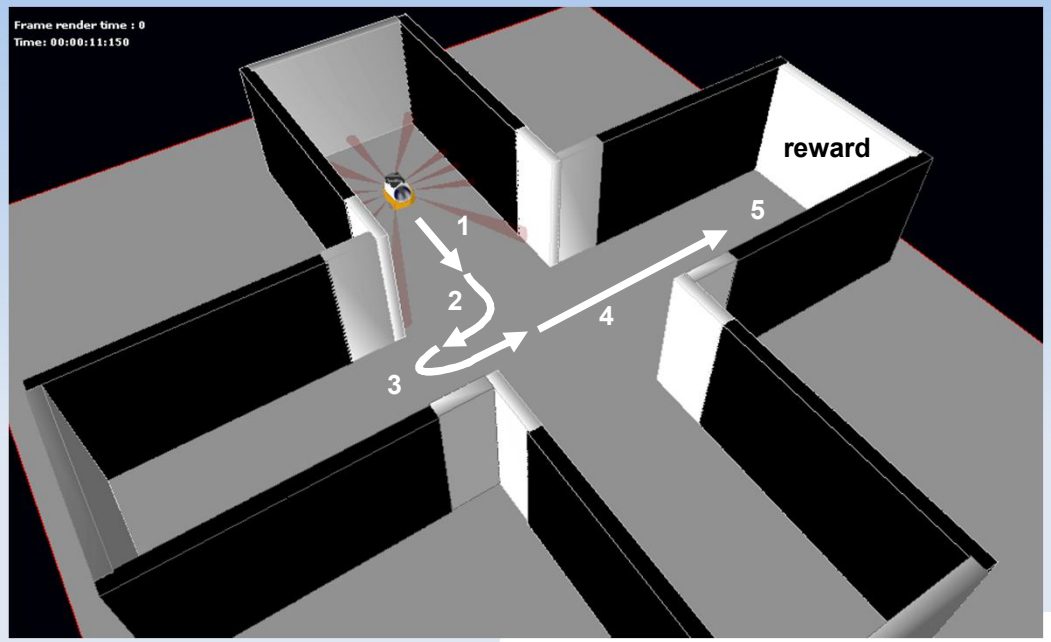
Temporal-order input
[0 0 1 0 0 0 0]

also called:
Tapped-delay line



Montague et al. (1996); Suri & Schultz (2001)
Daw (2003); Bertin et al. (2007).

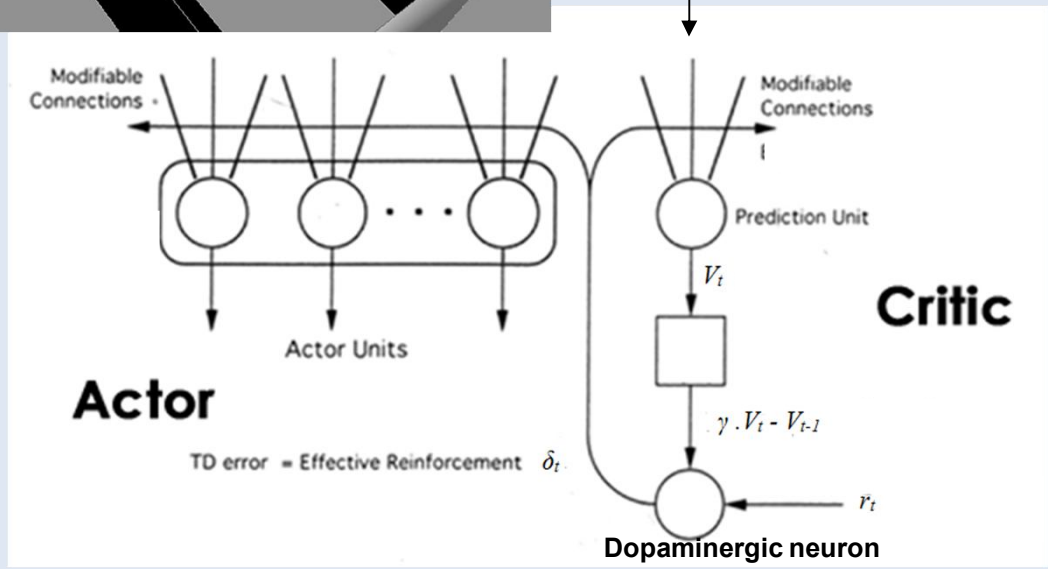
The Actor-Critic model



Which state space as an input?

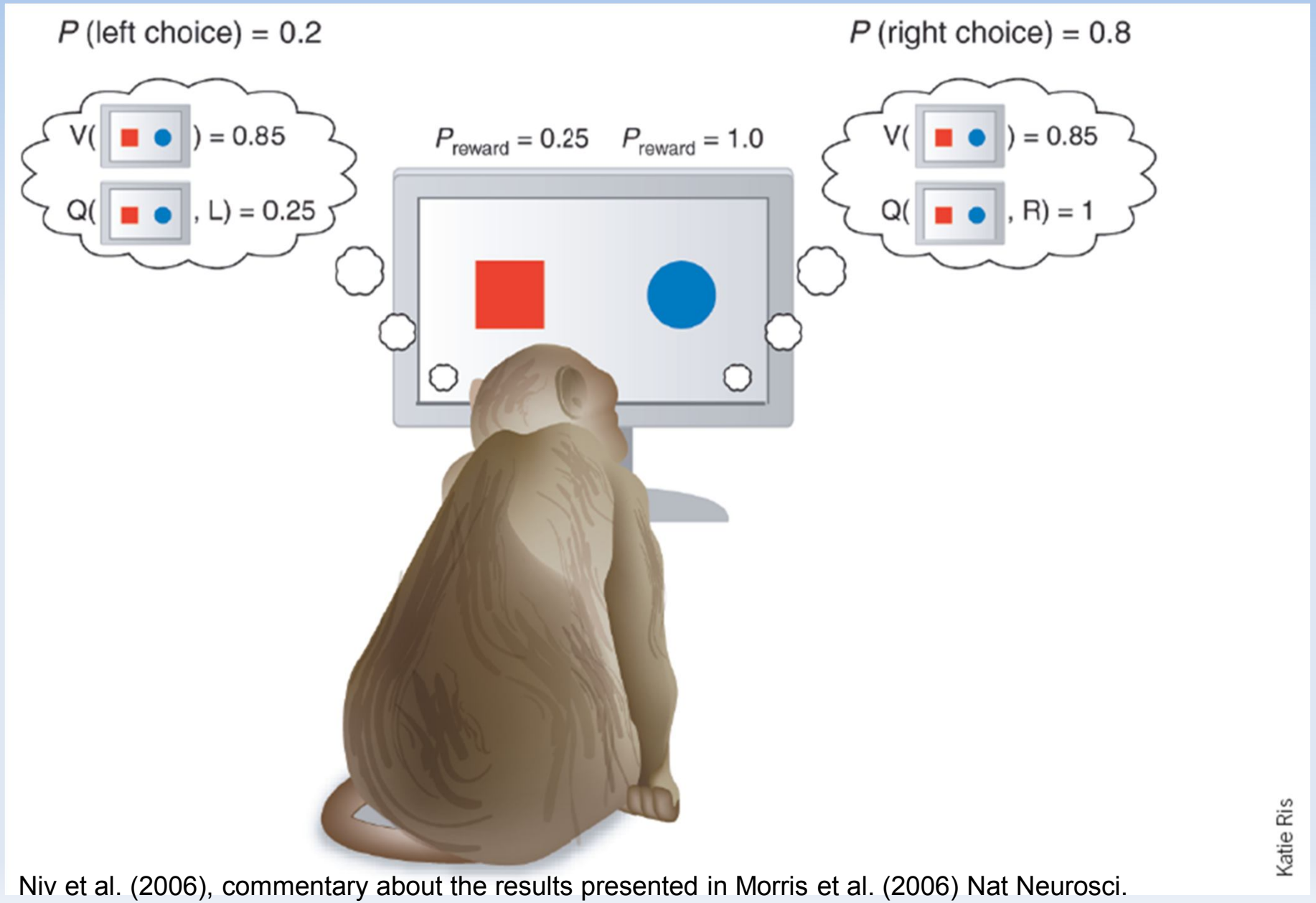
Temporal-order input
[0 1 0 0 0 0]

or spatial or visual information



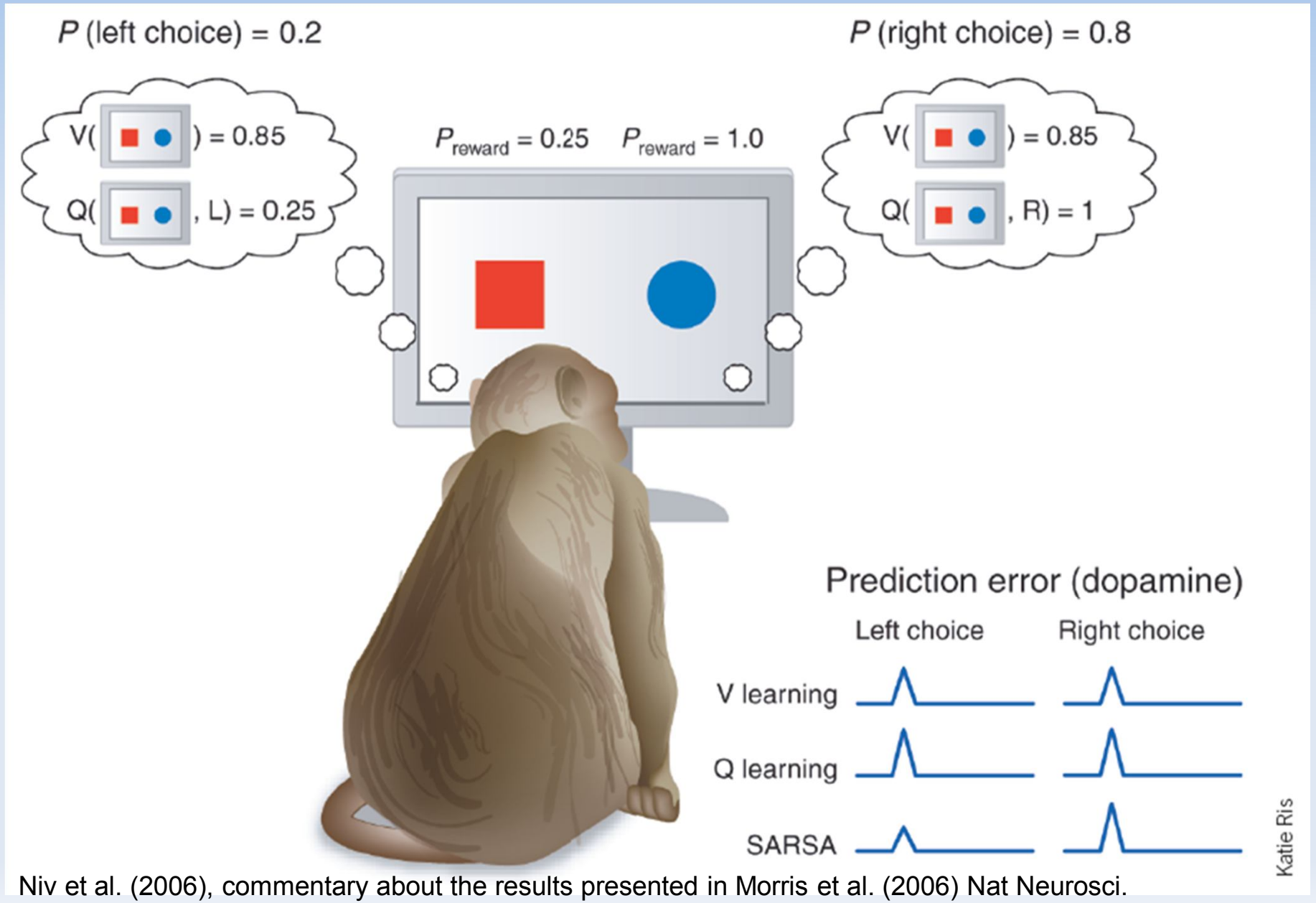
Wide application of RL models to model-based analyses of behavioral and physiological data during decision-making tasks

Typical probabilistic decision-making task



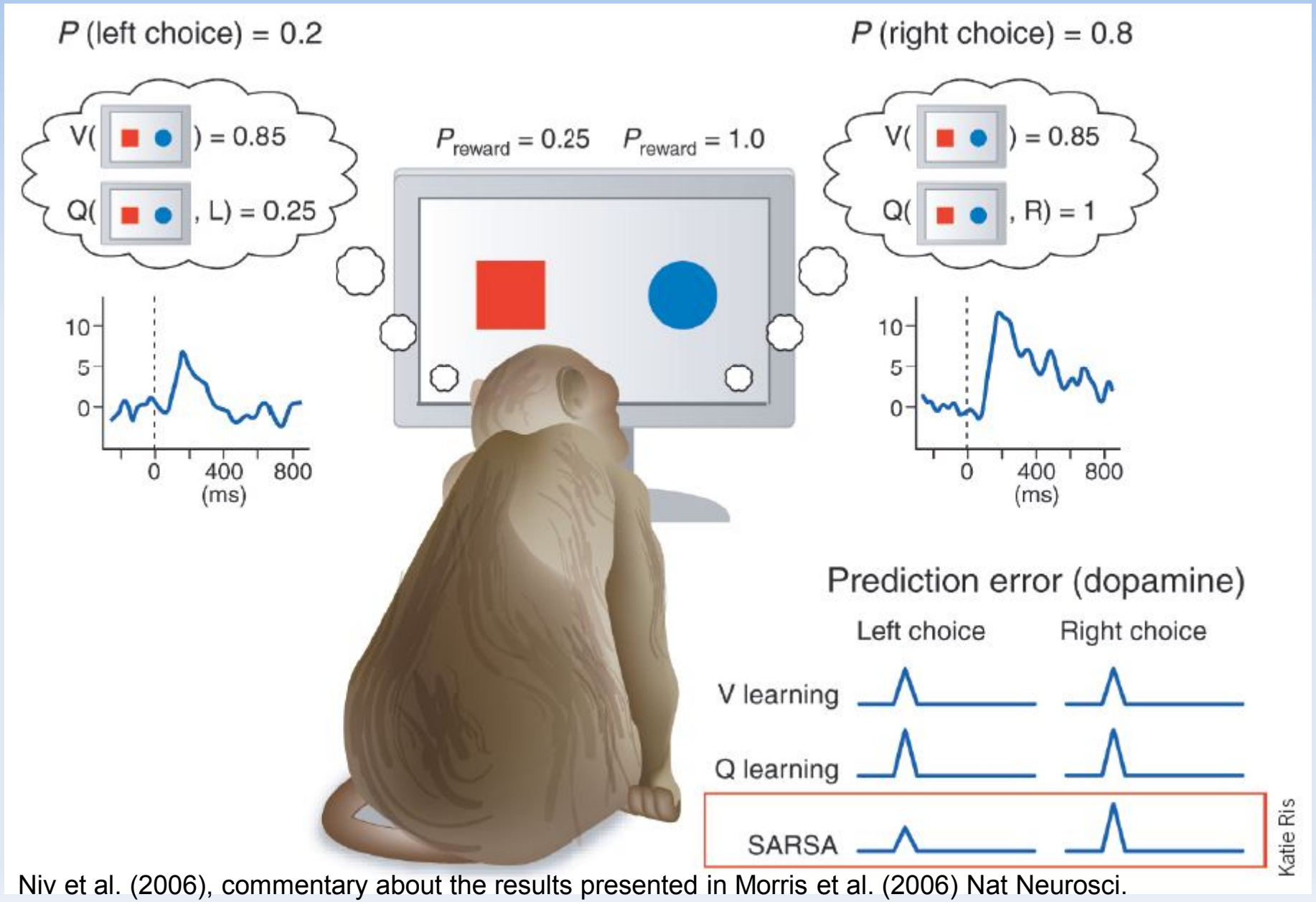
Niv et al. (2006), commentary about the results presented in Morris et al. (2006) Nat Neurosci.

Typical probabilistic decision-making task



Niv et al. (2006), commentary about the results presented in Morris et al. (2006) Nat Neurosci.

Typical probabilistic decision-making task



Niv et al. (2006), commentary about the results presented in Morris et al. (2006) Nat Neurosci.

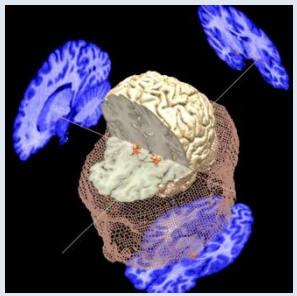
Model-based analysis of brain data

Sequence of observed trials : Left (Reward); Left (Nothing); Right (Nothing); Left (Reward); …

fMRI scanner

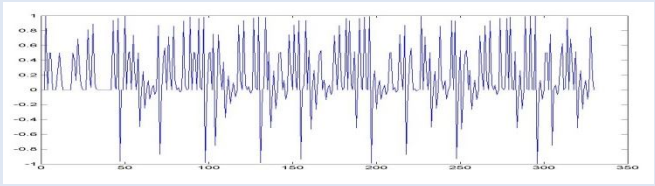
RL model

Brain responses



?

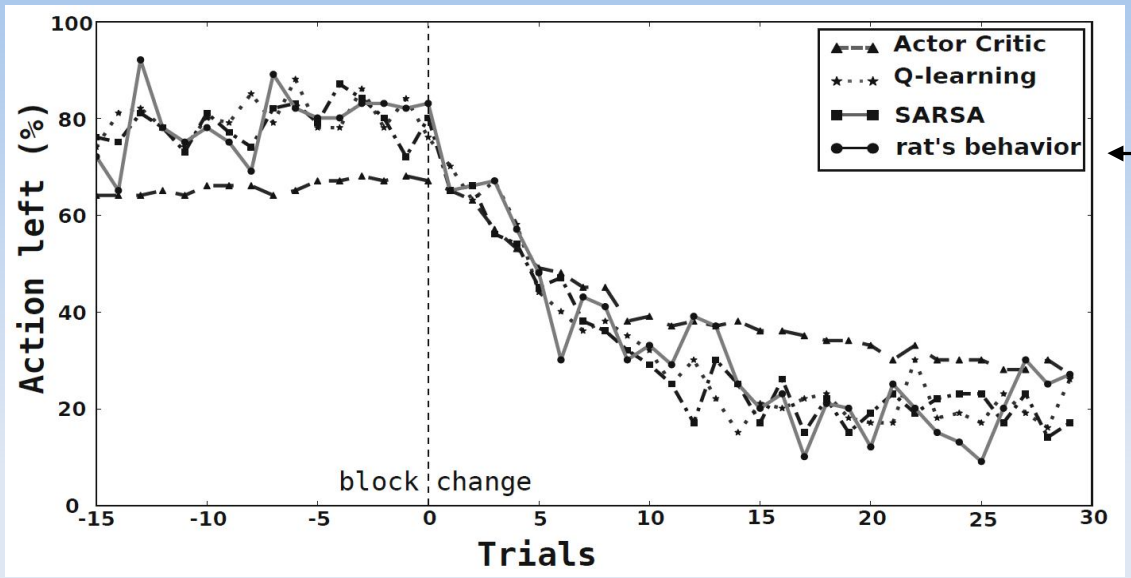
Prediction error



cf. travail de Mathias Pessiglione (ICM)

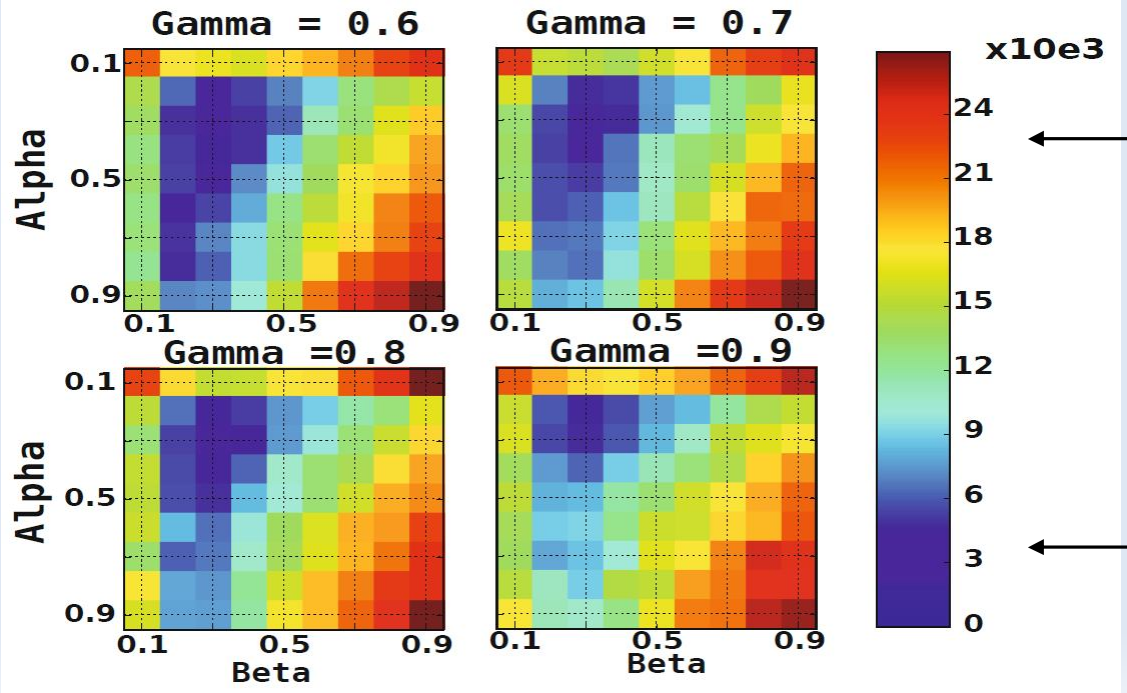
ou Giorgio Coricelli (ENS)

Model-based analysis
Work by Jean Bellot (PhD student)



TD-learning models

Behavior of the animal

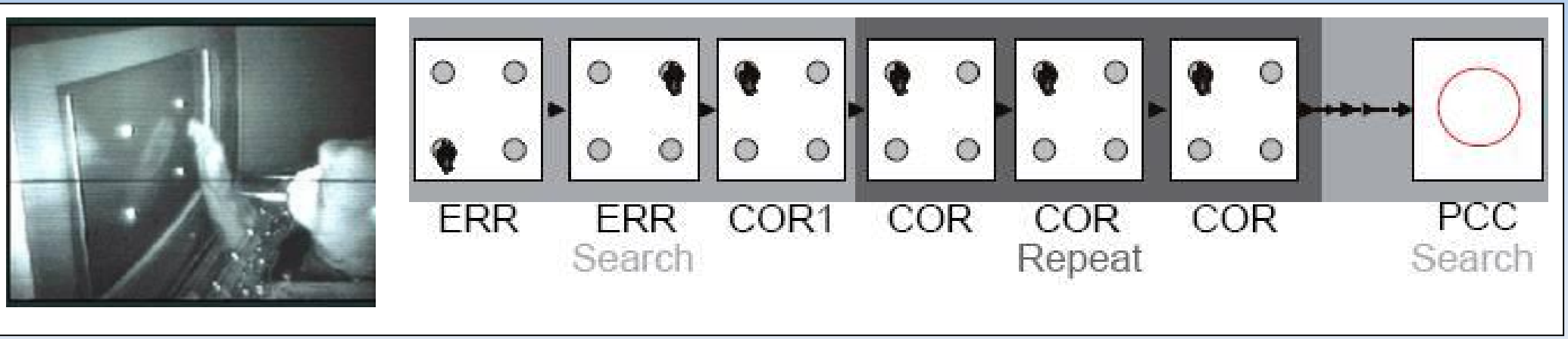


High fitting error

Low fitting error

Bellot, Sigaud, Khamassi (2012) SAB conference.

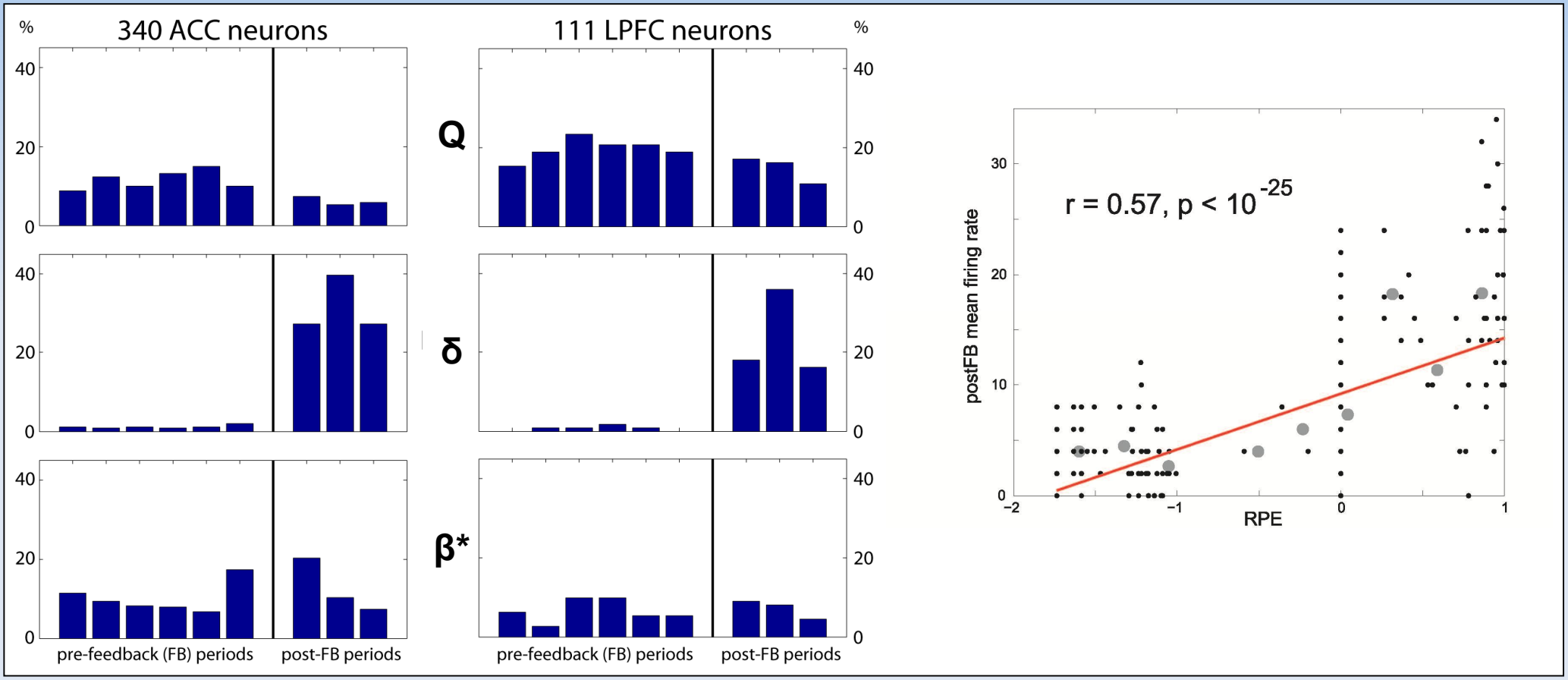
Model-based analysis
My post-doc work



- Analysis of single neurons recorded in the monkey dorsolateral prefrontal cortex and anterior cingulate cortex
- Correlates of prediction errors? Action values? Level of control/exploration?

Khamassi et al. (2013) Prog Brain Res; Khamassi et al. (in revision)

Model-based analysis
My post-doc work



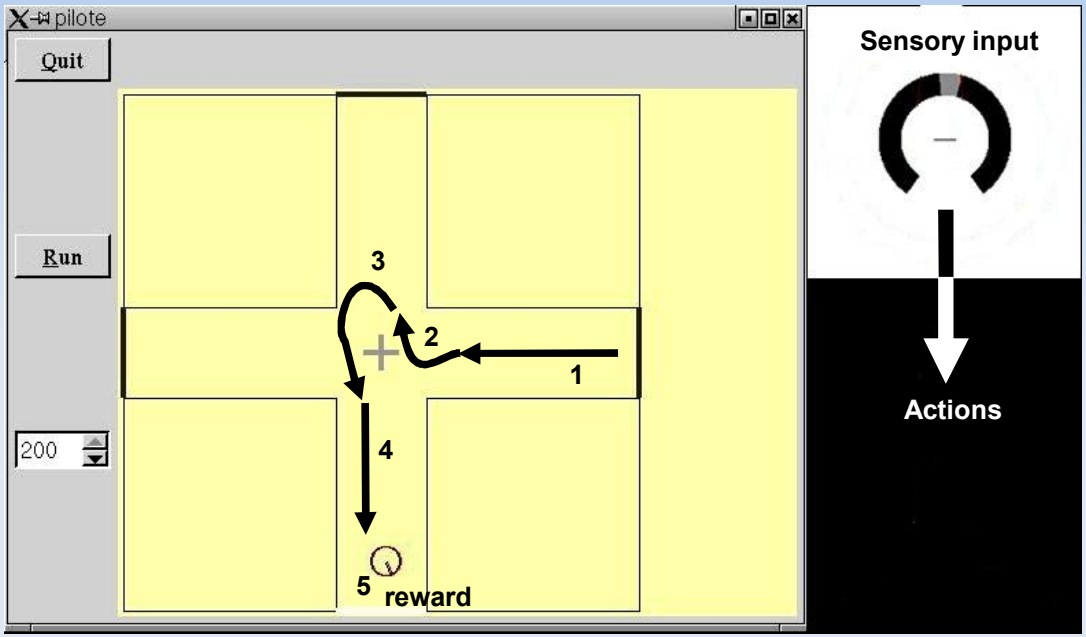
Multiple regression analysis with bootstrap

Khamassi et al. (2013) Prog Brain Res; Khamassi et al. (in revision)

This works well, but…

- *Most experiments are single-step*
- *All these cases are discrete*
- *Very small number of states, actions*
- *We supposed a perfect state identification*

Continuous reinforcement learning

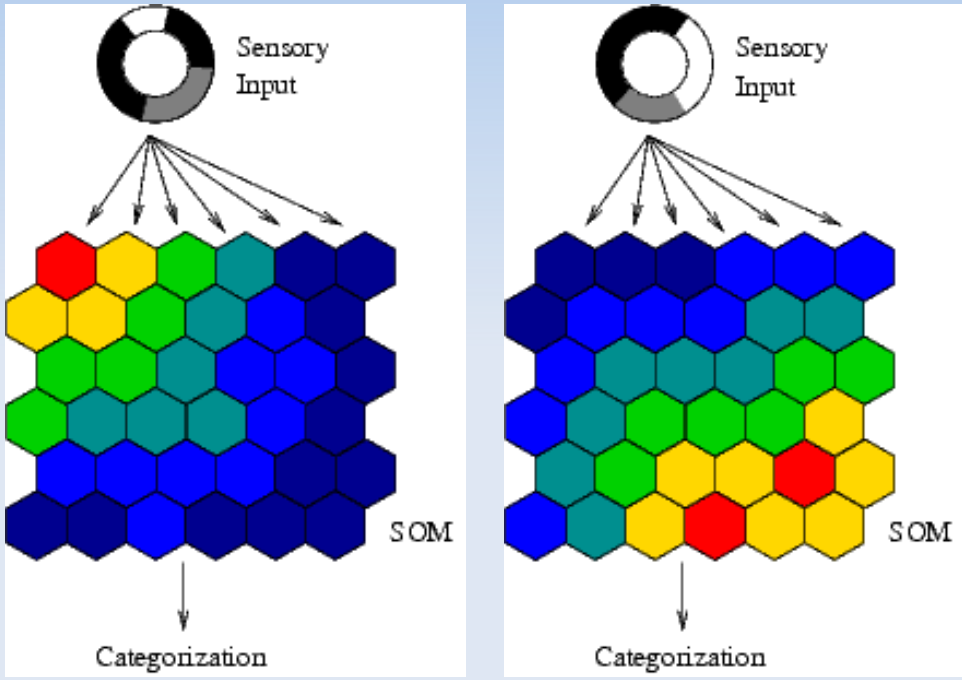


TD-Learning model applied to spatial navigation behavior learning in a robot performing the bio-inspired plus-maze task

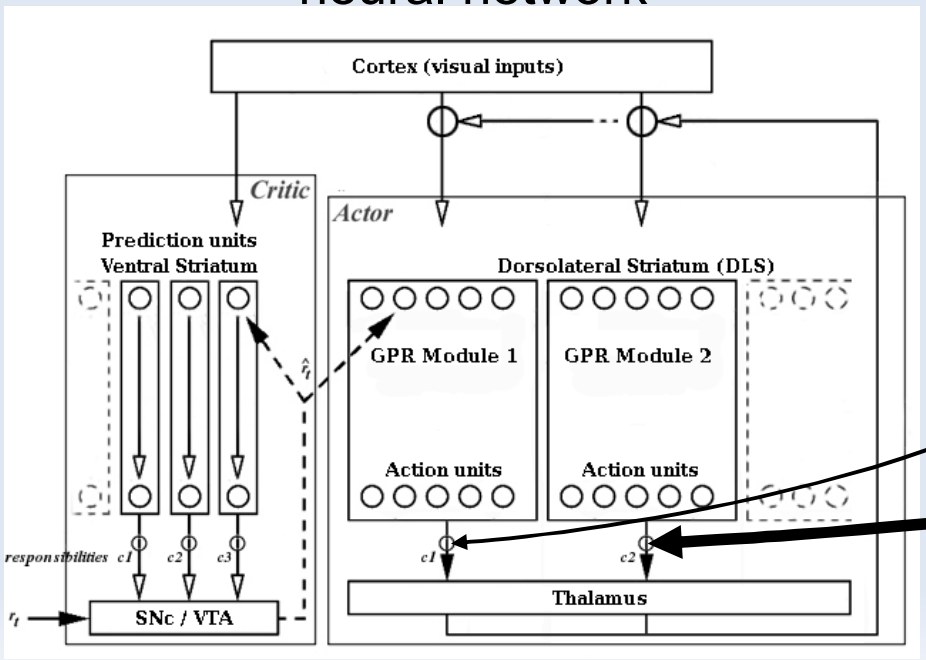
Khamassi et al. (2005). *Adaptive Behavior.*
Khamassi et al. (2006). *Lecture Notes in Computer Science*

Continuous reinforcement learning

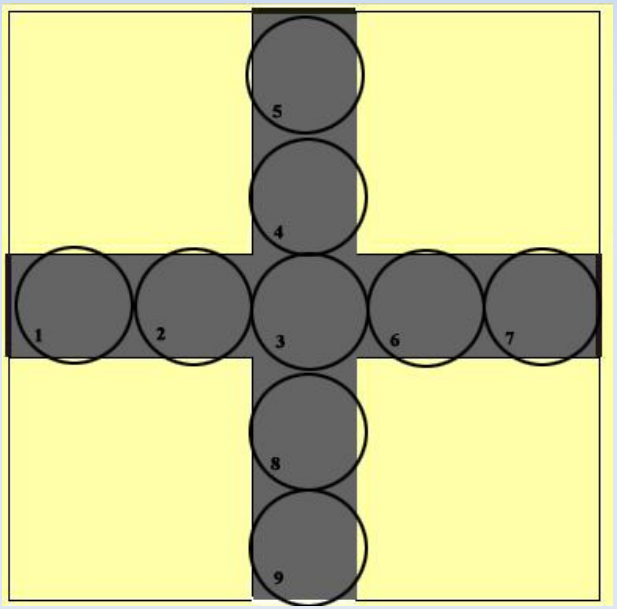
Coordination by a self-organizing map



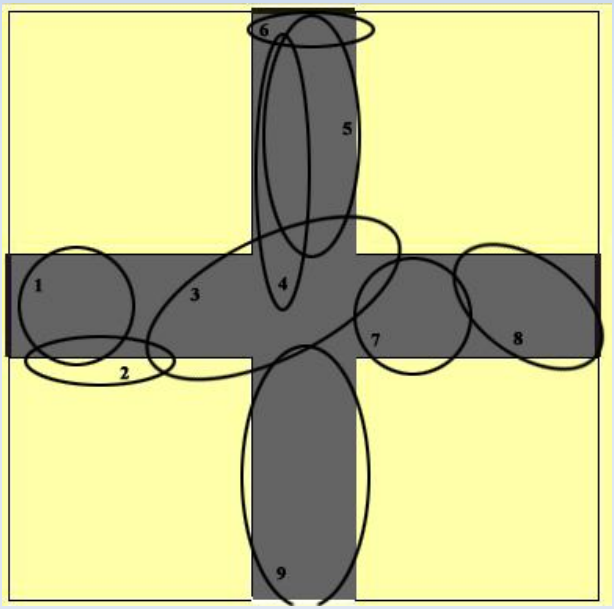
Actor-Critic multi-modules neural network



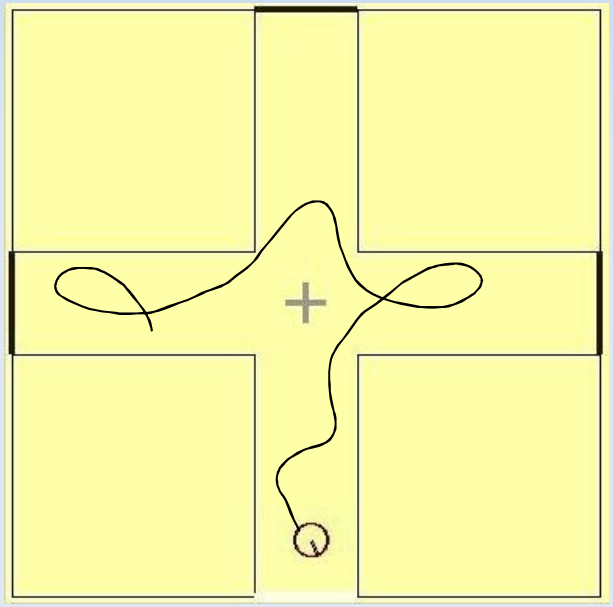
Continuous reinforcement learning



Hand-tuned

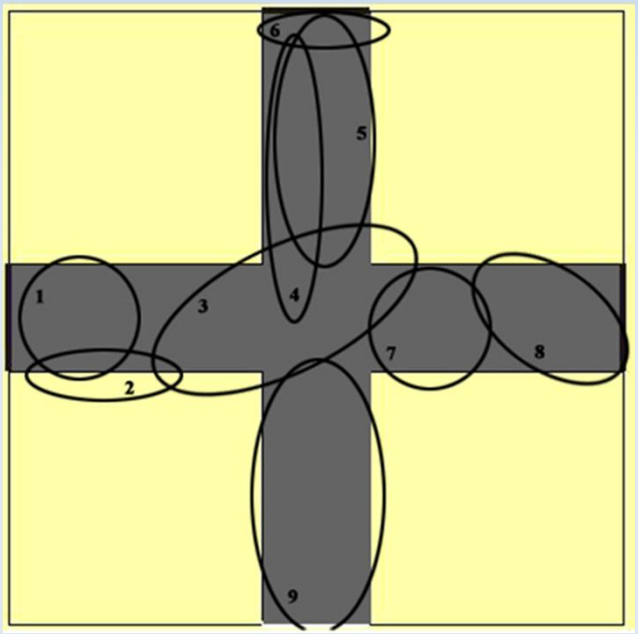


Autonomous



Random

Continuous reinforcement learning



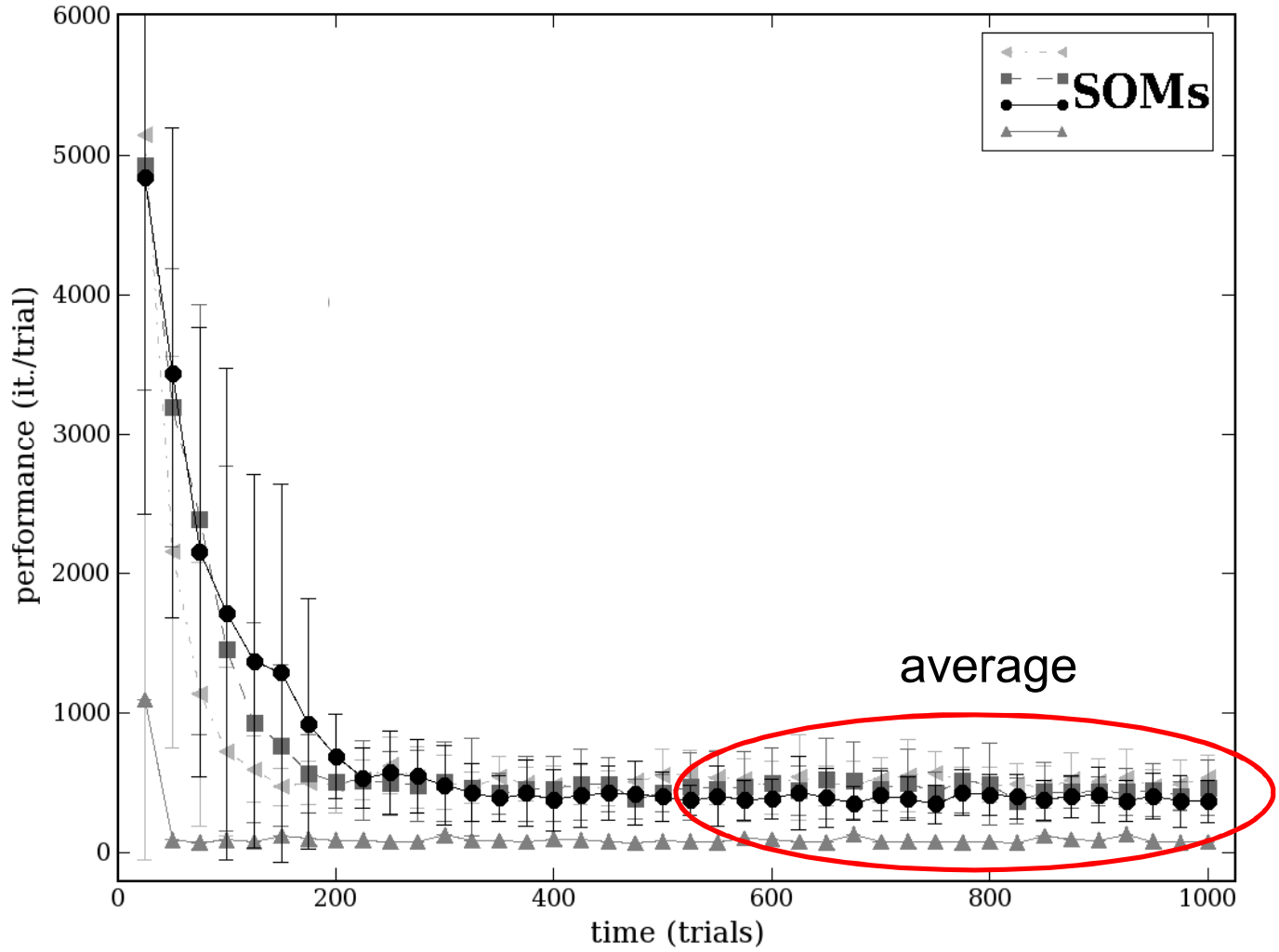
Autonomous

Two methods :

1. Self-Organizing Maps (SOMs)

2. specialization based on performance (tests modules' capacity for state prediction) Baldassarre (2002); Doya et al. (2002). Within a particular subpart of the maze, only the module with the most accurate reward prediction is trained. Each module thus becomes an expert responsible for learning in a given task subset.

Continuous reinforcement learning

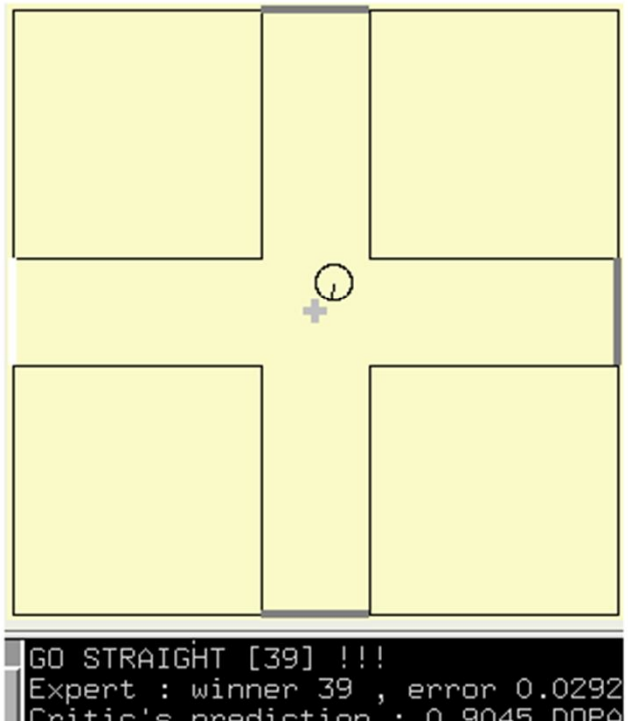


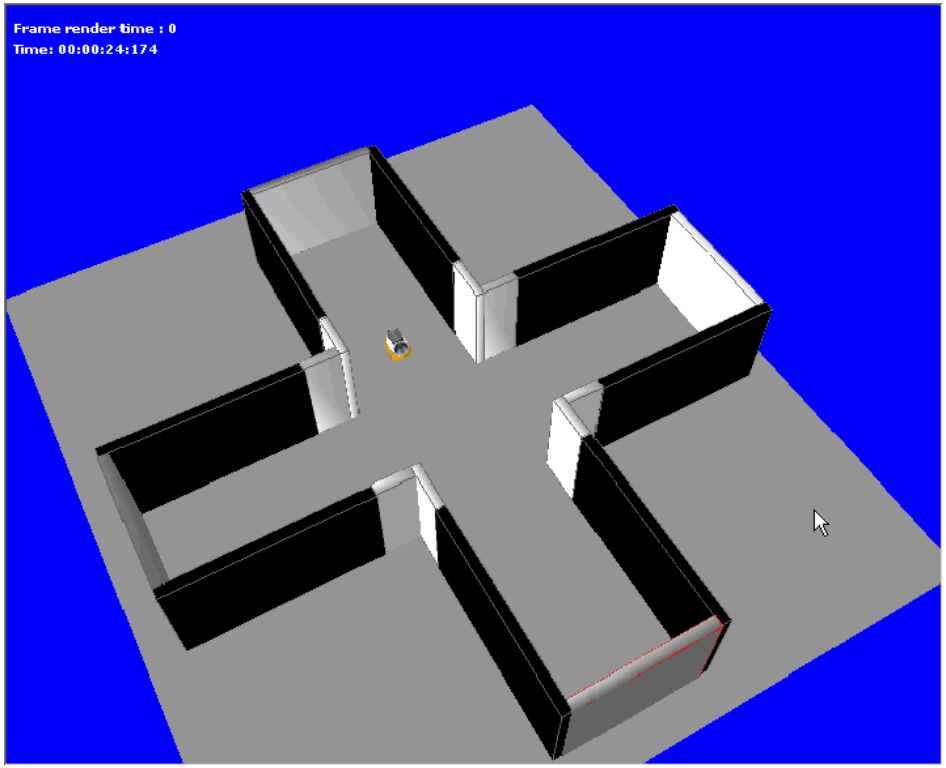
Continuous reinforcement learning

Nb of iterations required

(Average performance during the second half of the experiment)

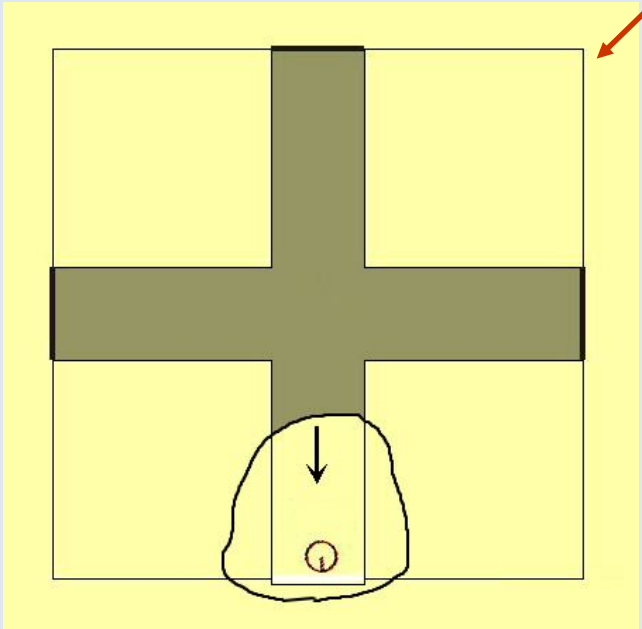
1. hand-tuned	94
2. specialization based on performance	3,500
3. autonomous categorization (SOM)	404
4. random robot	30,000





Continuous reinforcement learning

	Nb of iterations required (Average performance during the second half of the experiment)
1. hand-tuned	94
2. specialization based on performance	3,500
3. autonomous categorization (SOM)	404
4. random robot	30,000



Outline

1. Model-free Reinforcement Learning
   - Temporal-Difference RL Algorithm
   - Dopamine activity
   - Wide application to Neuroscience of decision-making

2. Model-based Reinforcement Learning
   - Off-line learning / Replay during sleep
   - Dual-system RL
   - Online parameters tuning (meta-learning)
   - Link with Neurobehavioral data
   - Applications to Robotics

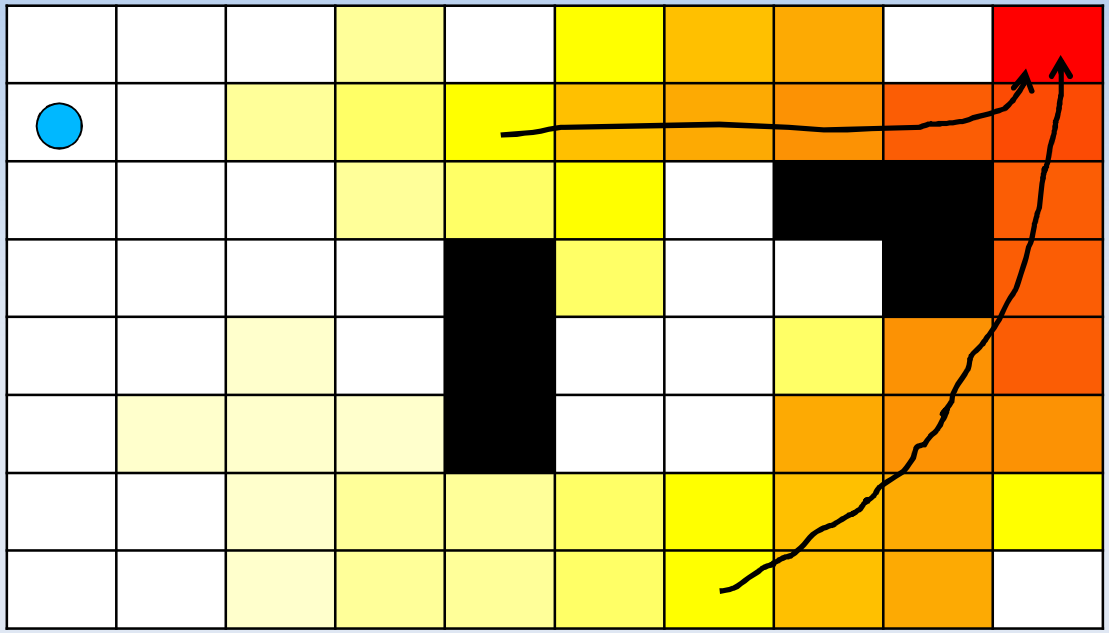
Off-learning (Model-based RL) and hippocampal & prefrontal cortex activity replay during sleep

REINFORCEMENT LEARNING

After

N simulations

Very long!



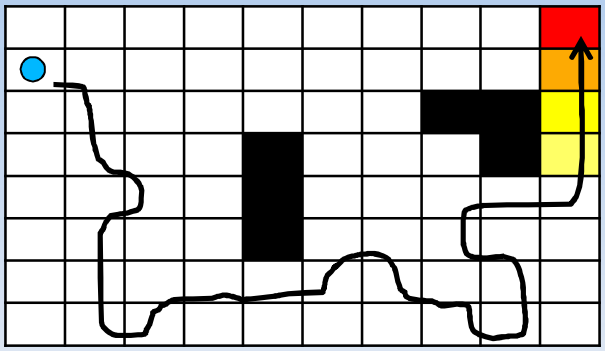
$$\delta_{t+1} = r_{t+1} + \gamma \cdot V(s_{t+1}) - V(s_t)$$

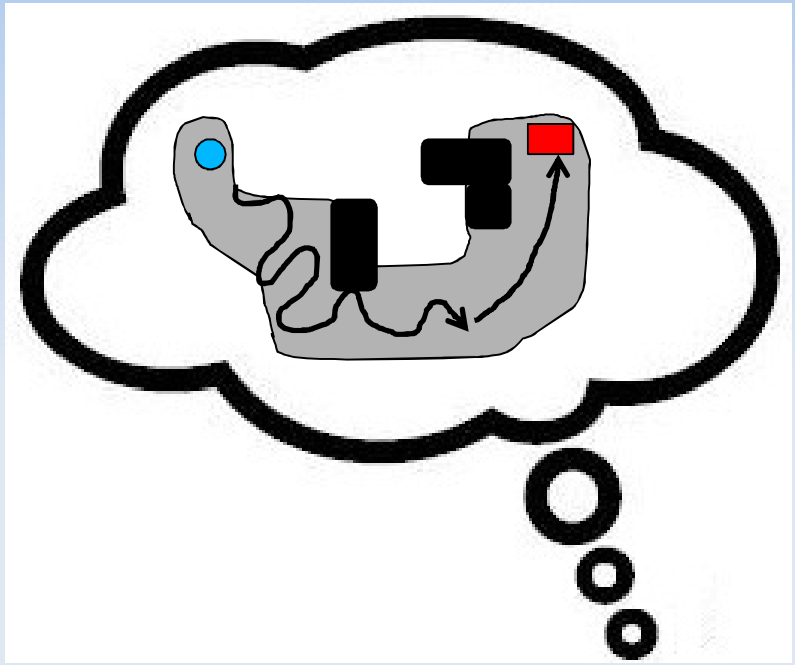
discount factor (=0.9)

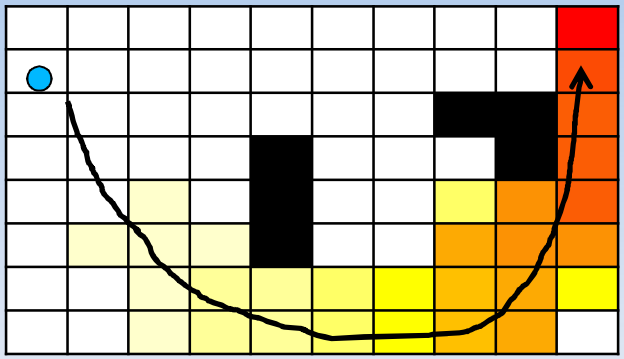
$$V(s_t) = V(s_t) + \alpha \cdot \delta_{t+1}$$

learning rate (=0.1)

TRAINING DURING SLEEP







Method in Artificial Intelligence:
Off-line Dyna-Q-learning
(Sutton & Barto, 1998)

Model-based Reinforcement Learning

To incrementally learn a model of transition and reward functions, then plan within this model by updates "in the head of the agent" (Sutton, 1990).

$\boldsymbol{S}$: state space

$\boldsymbol{A}$: action space

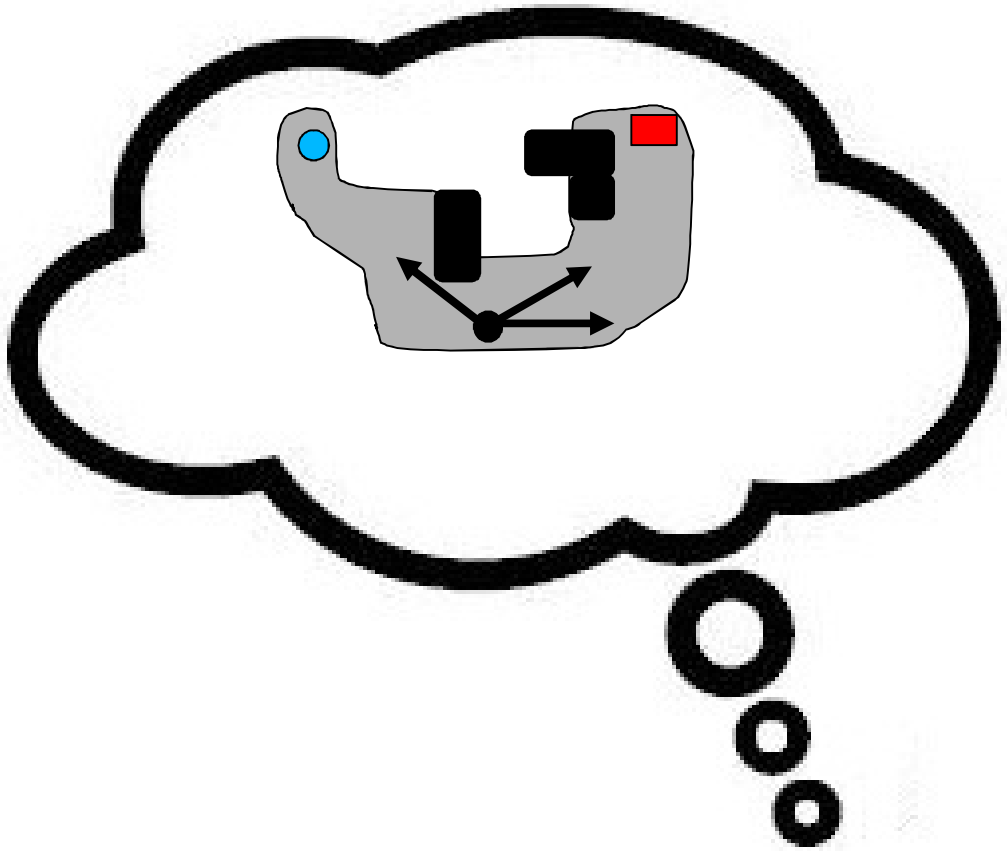
Transition function $\boldsymbol{T} : S \times A \longrightarrow S$

Reward function $\boldsymbol{R} : S \times A \longrightarrow \mathbb{R}$

} Internal model

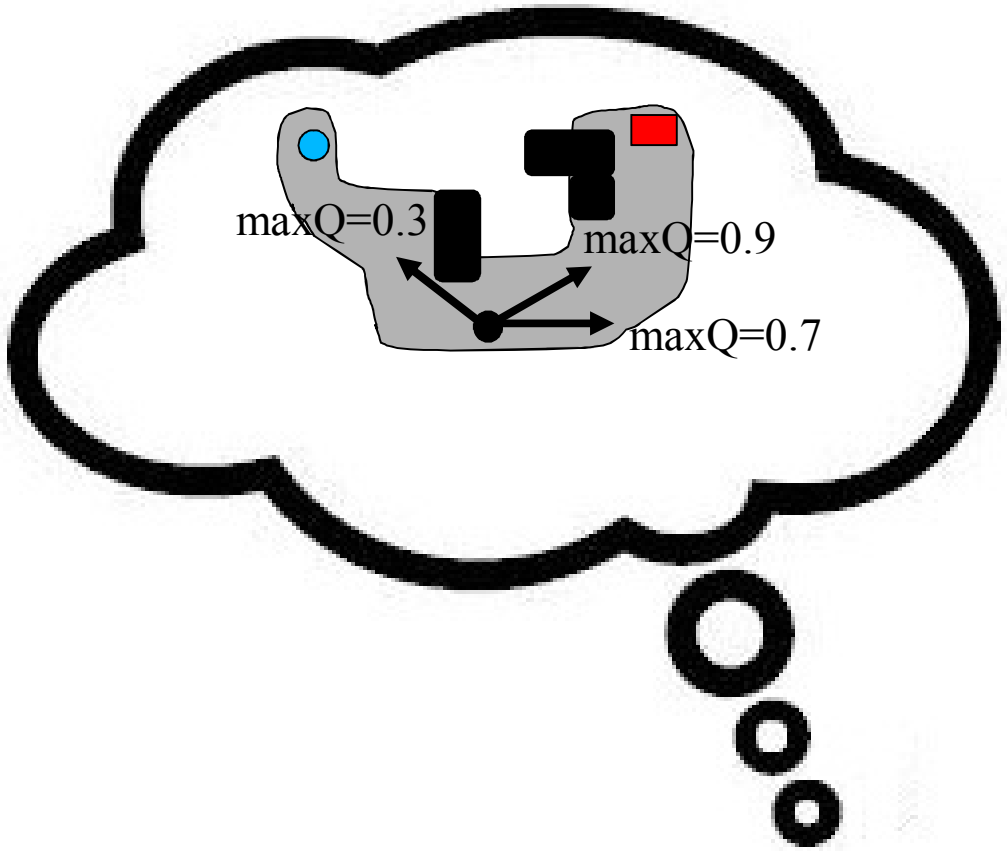
Model-based Reinforcement Learning

s : state of the agent (●)



Model-based Reinforcement Learning

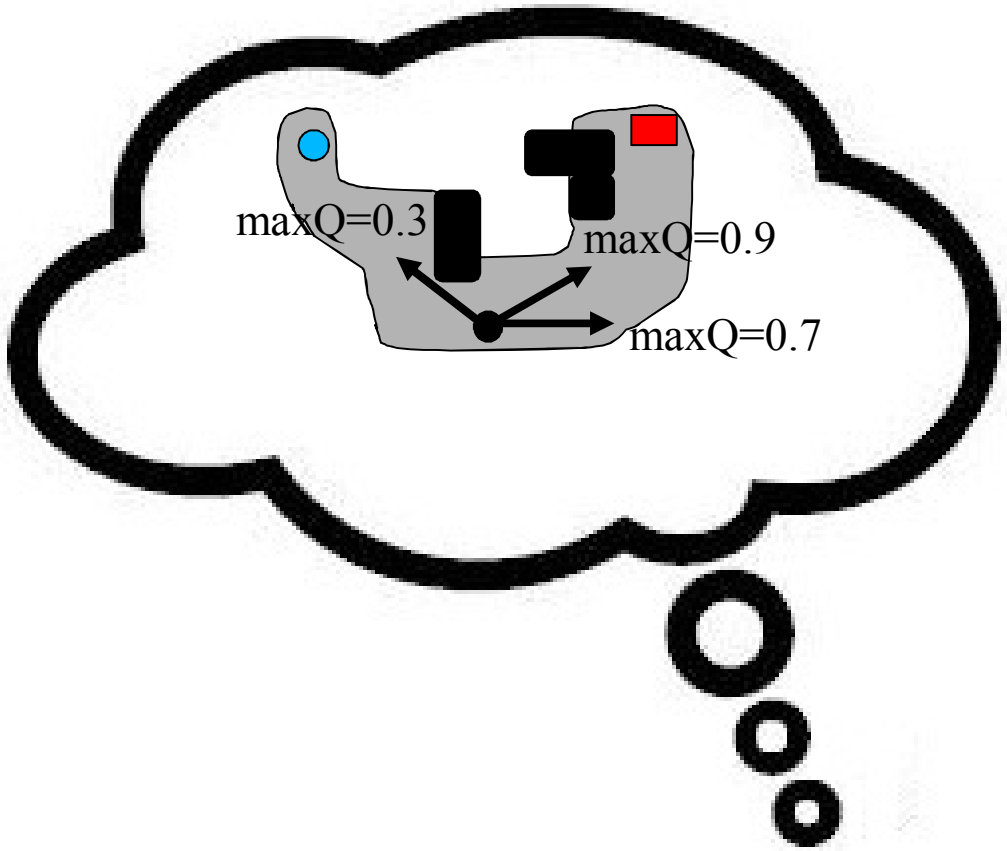
s : state of the agent (●)



Model-based Reinforcement Learning

s : state of the agent (●)

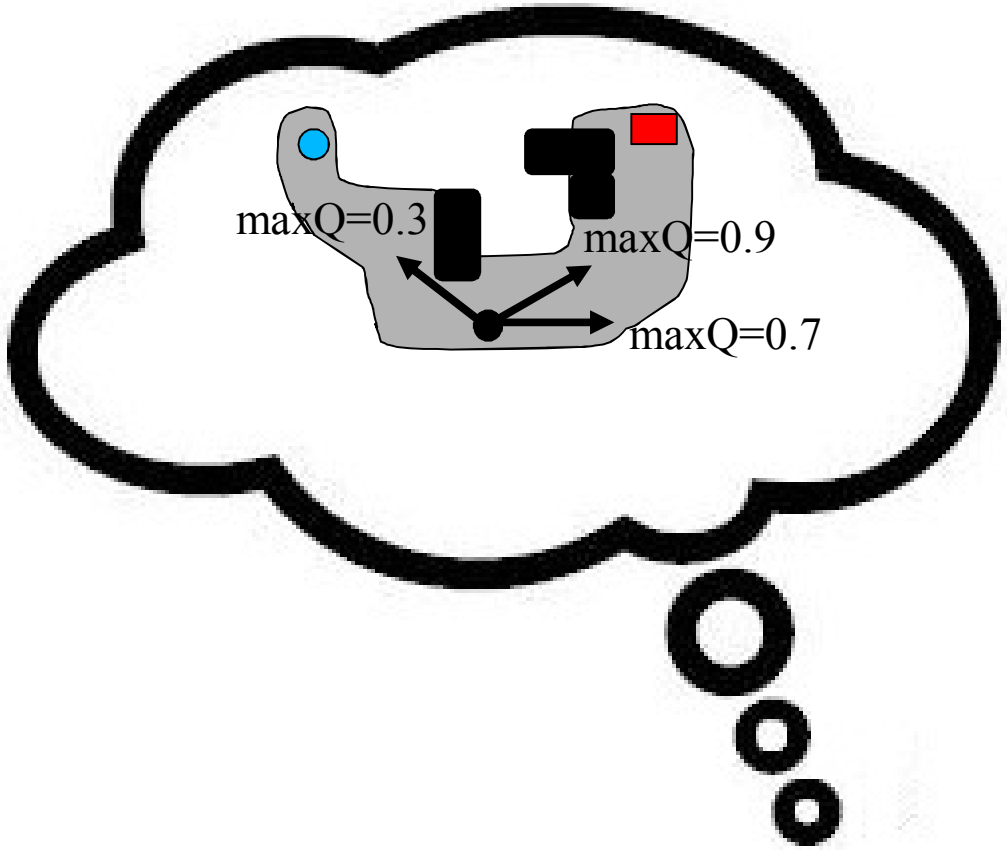
a : action of the agent (**go east**)



Model-based Reinforcement Learning

s : state of the agent (●)

a : action of the agent (**go east**)



stored transition function T:

proba(→) = 0.9

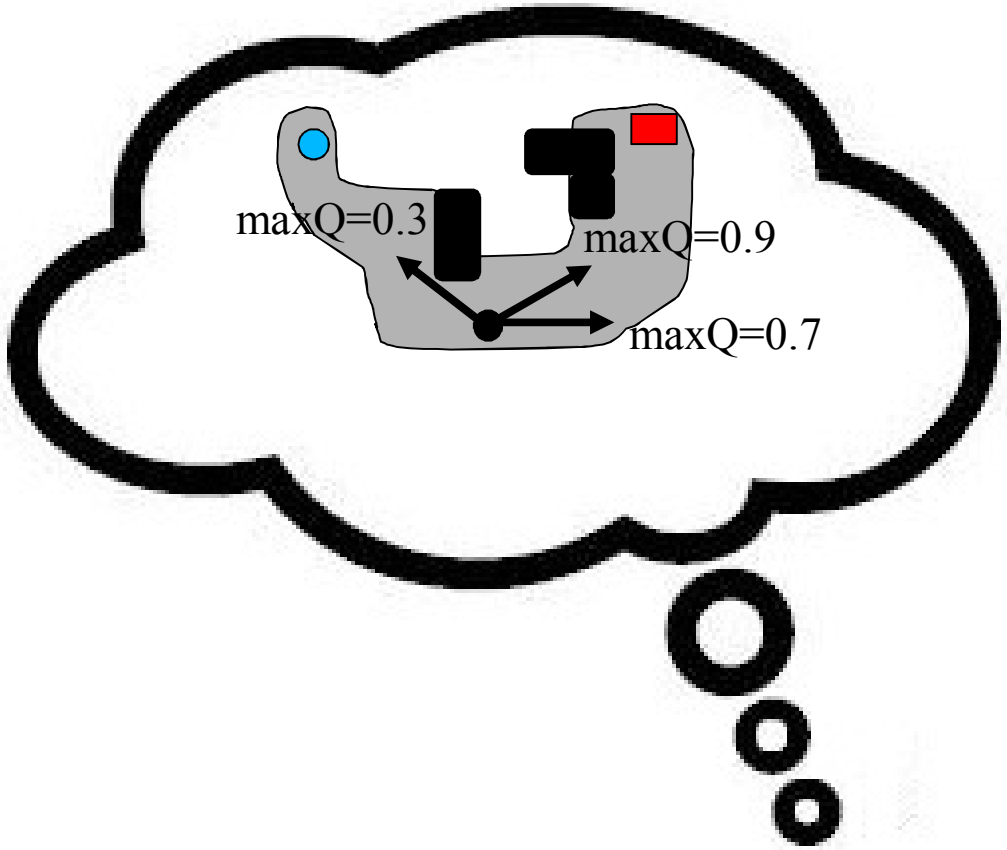
proba(↗) = 0.1

proba(↖) = 0

Model-based Reinforcement Learning

s : state of the agent (●)

a : action of the agent (**go east**)



stored transition function T:

proba(→ *) = 0.9*

proba(↗ *) = 0.1*

proba(↖ *) = 0*

$$\mathcal{Q}(s,a) \leftarrow \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s'|s,a) \max_{a'} \mathcal{Q}(s',a')$$

0.6 0 0.9*0.7 + 0.1*0.9 + 0*0.3 + …

Model-based Reinforcement Learning

No reward prediction error!

Only:

Estimated Q-values

Transition function

Reward function

This process is called **Value Iteration or Dynamic prog.**

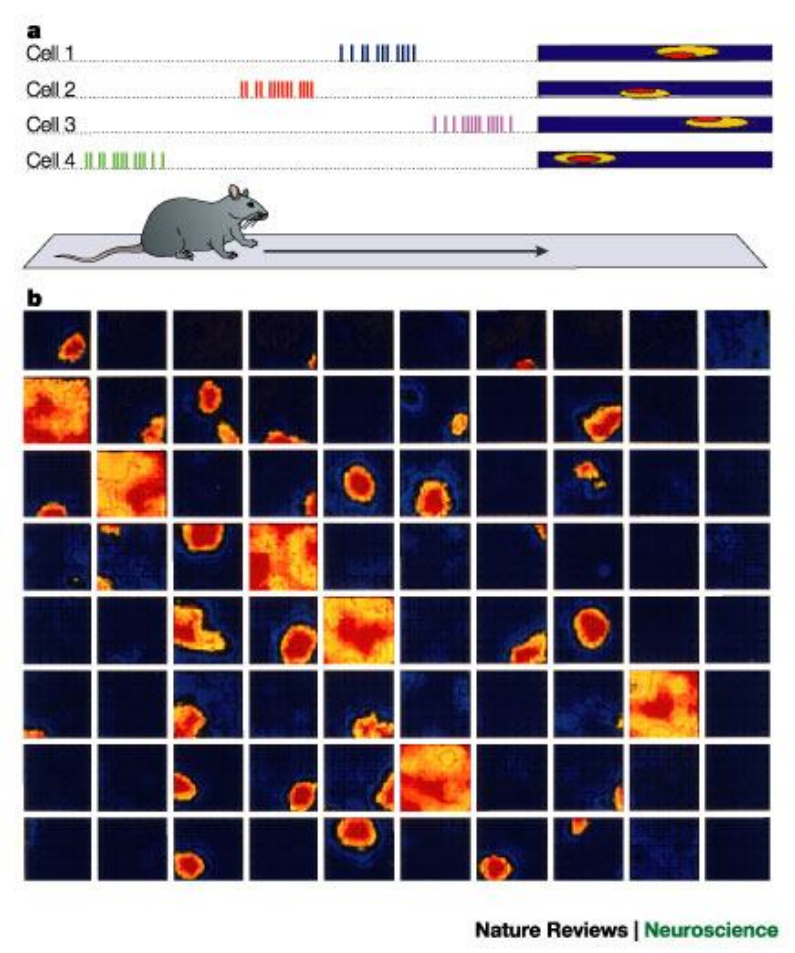
$$\mathcal{Q}(s,a) \leftarrow \mathcal{R}(s,a) + \gamma \sum_{s'} \mathcal{T}(s'|s,a) \max_{a'} \mathcal{Q}(s',a')$$

Model-based Reinforcement Learning

Links with neurobiological data

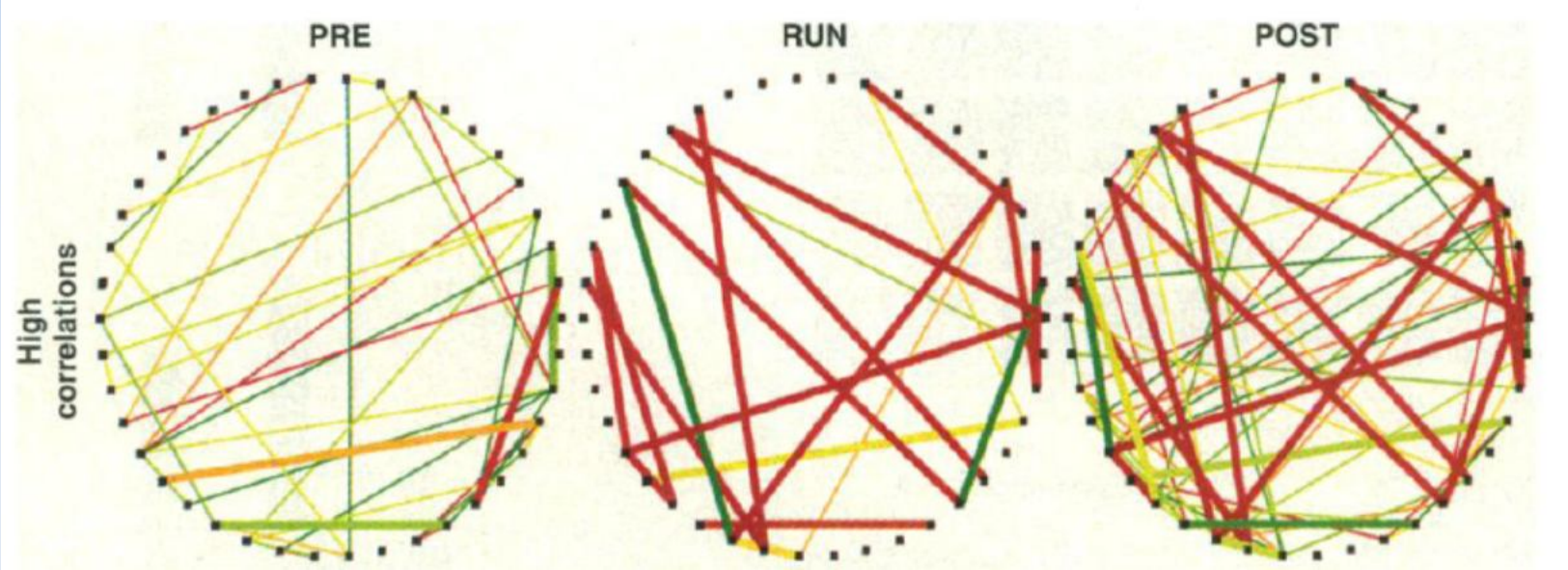
Activity of hippocampal place neurons

Hippocampal place cells



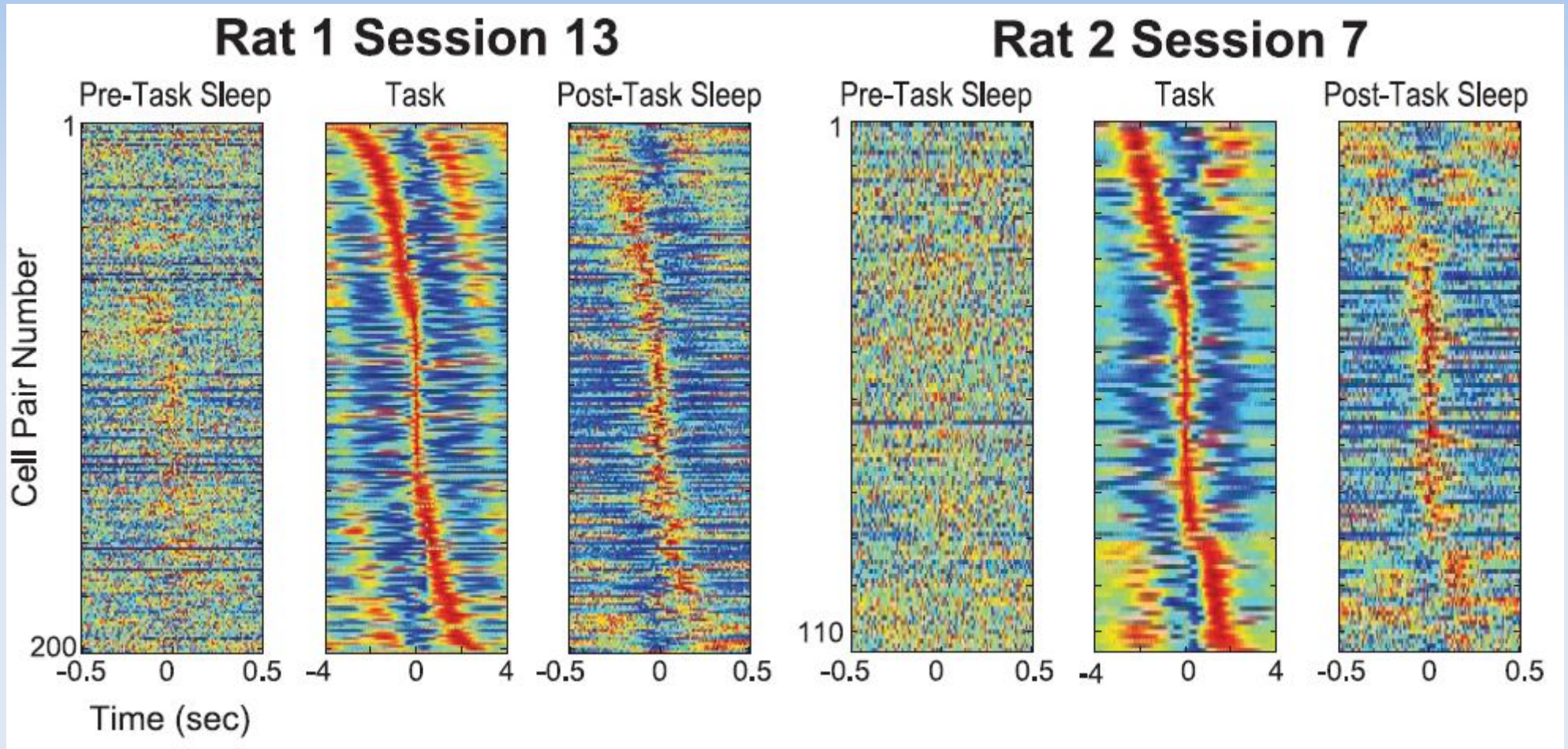
Nakazawa, McHugh, Wilson, Tonegawa (2004) Nature Reviews Neuroscience

Hippocampal place cells



- Reactivation of hippocampal place cells during sleep (Wilson & McNaughton, 1994, Science)

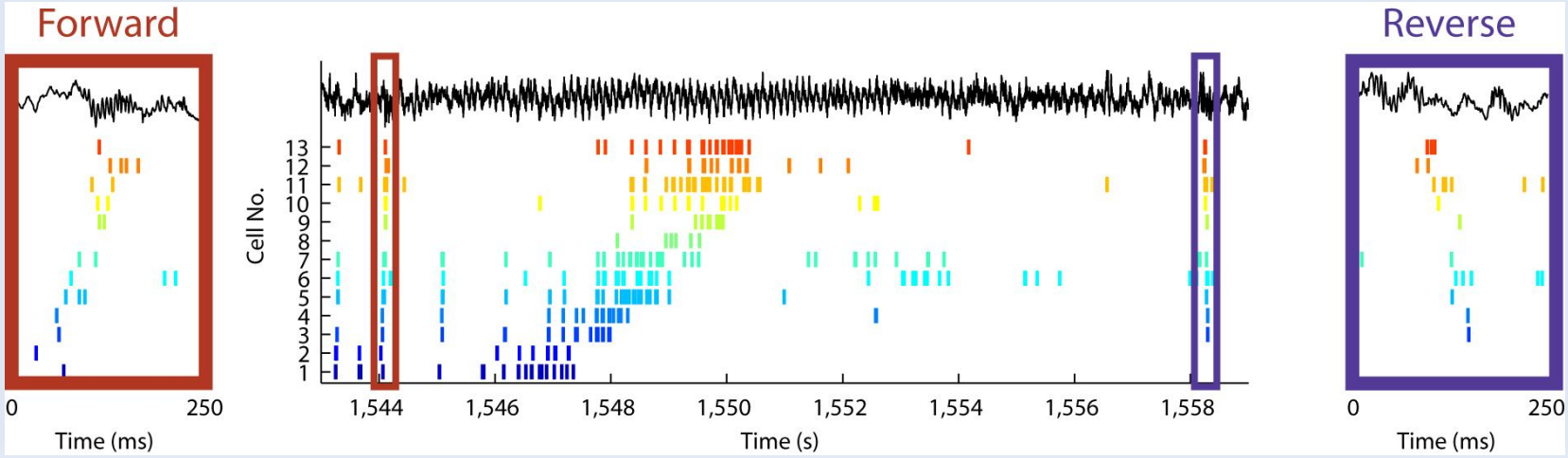
Hippocampal place cells



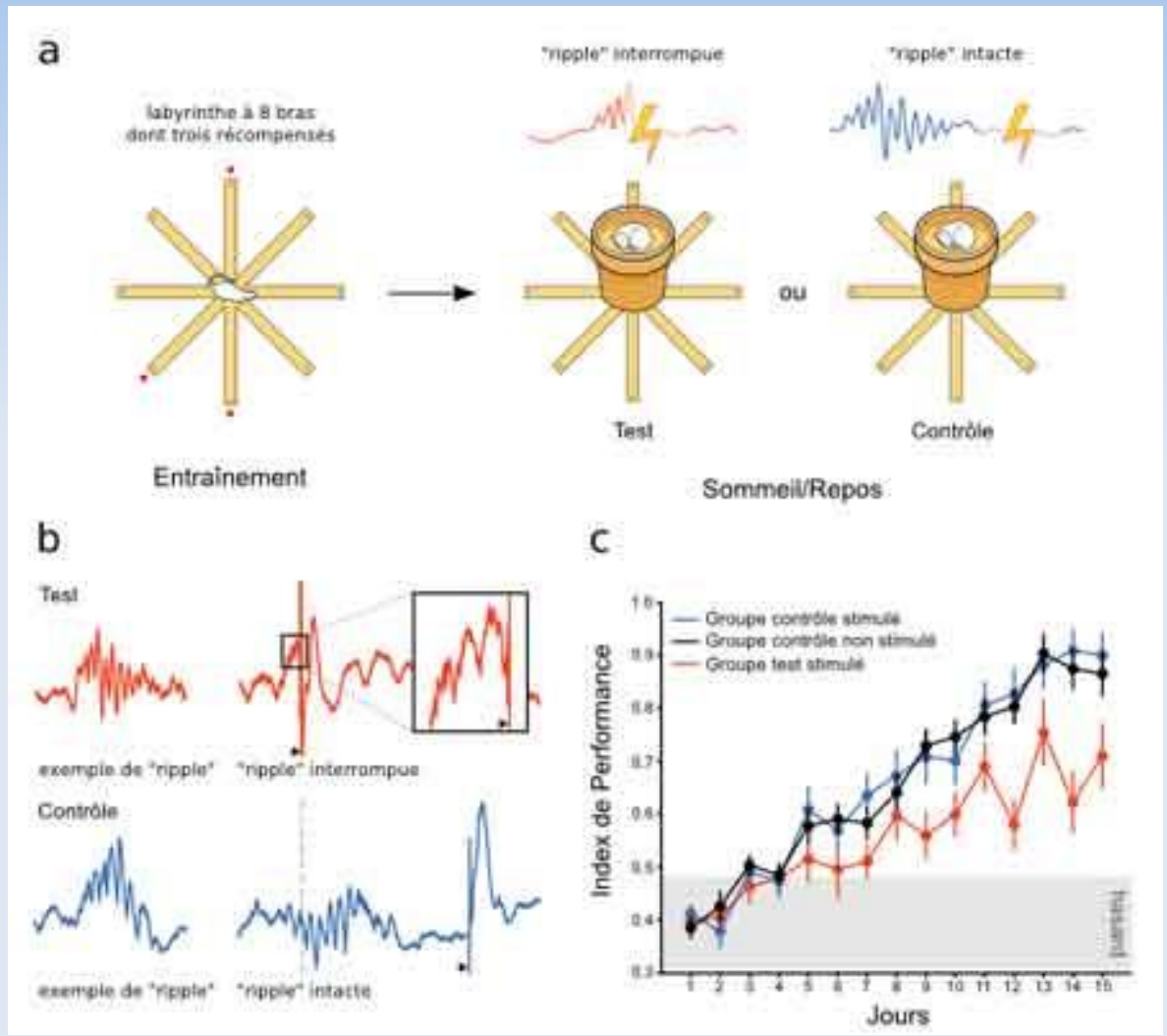
- Forward replay of hippocampal place cells during sleep (sequence is compressed 7 times) (Euston et al., 2007, Science)

Sharp-Wave Ripple (SWR) events

- "Ripple" events = irregular bursts of population activity that give rise to brief but intense high-frequency (100-250 Hz) oscillations in the CA1 pyramidal cell layer.



Selective suppression of SWRs impairs spatial memory



- Girardeau G, Benchenane K, Wiener SI, Buzsáki G, Zugaro MB (2009) Nat Neurosci.

Contribution to decision making (forward planning) and evaluation of transitions

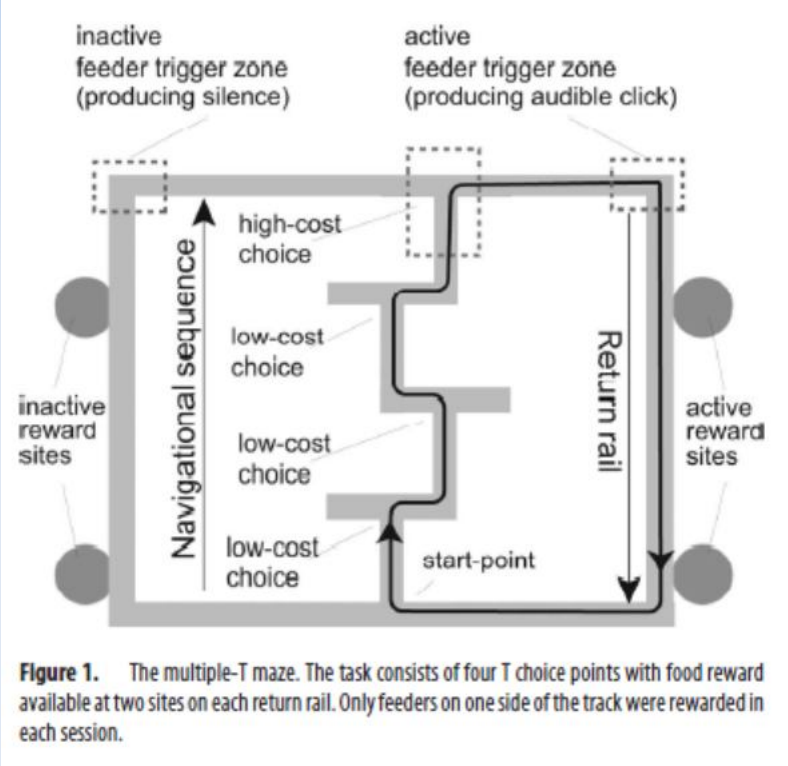
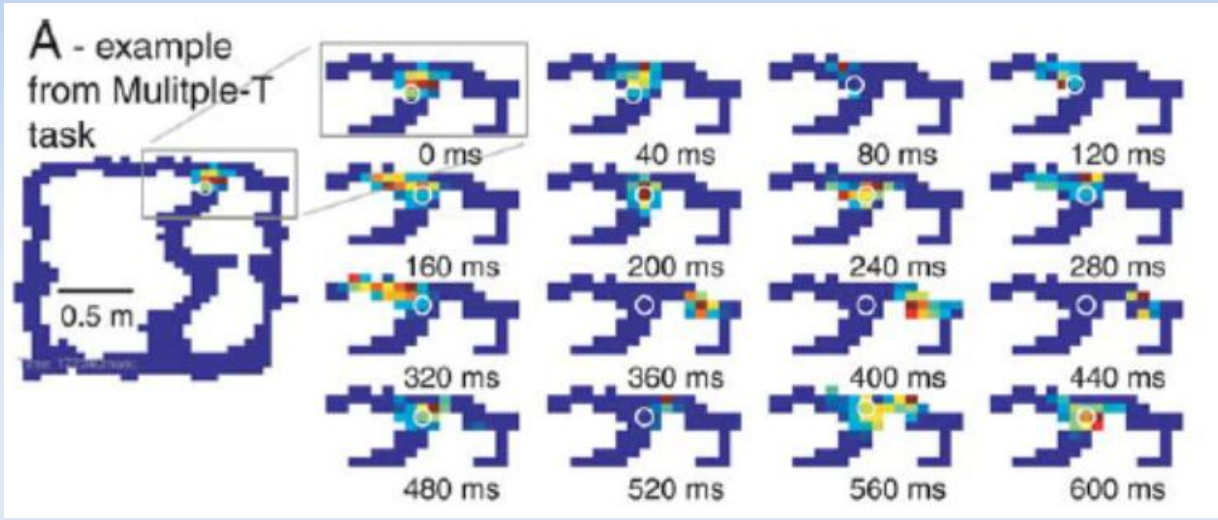


Figure 1. The multiple-T maze. The task consists of four T choice points with food reward available at two sites on each return rail. Only feeders on one side of the track were rewarded in each session.



Johnson & Redish (2007) J Neurosci

Hippocampal place cells

SUMMARY OF NEUROSCIENCE DATA

Replay their sequential activity during sleep (Foster & Wilson, 2006; Euston et al., 2007; Gupta et al., 2010)

Performance is impaired if this replay is disrupted (Girardeau, Benchenane et al. 2012; Jadhav et al. 2012)

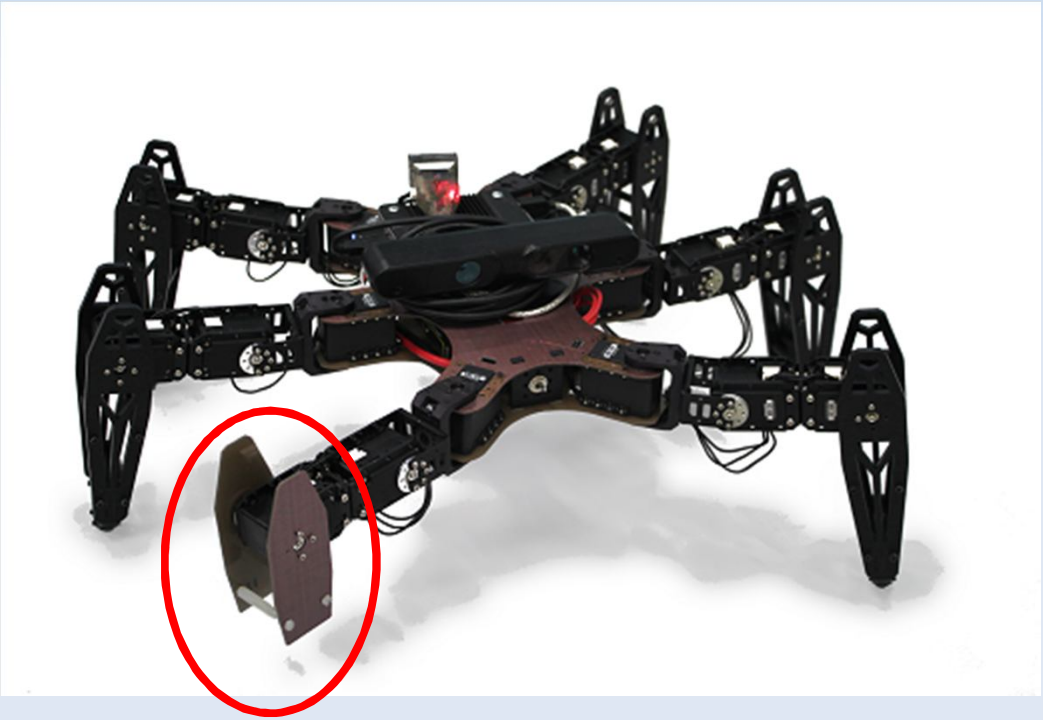
Only task-related replay in PFC (Peyrache et al., 2009)

Hippocampus may contribute to model-based navigation strategies, striatum to model-free navigation strategies (Khamassi & Humphries, 2012)

Applications to robot off-line learning

Work of Jean-Baptiste Mouret et al. @ ISIR

How to recover from damage without needing to identify the damage?

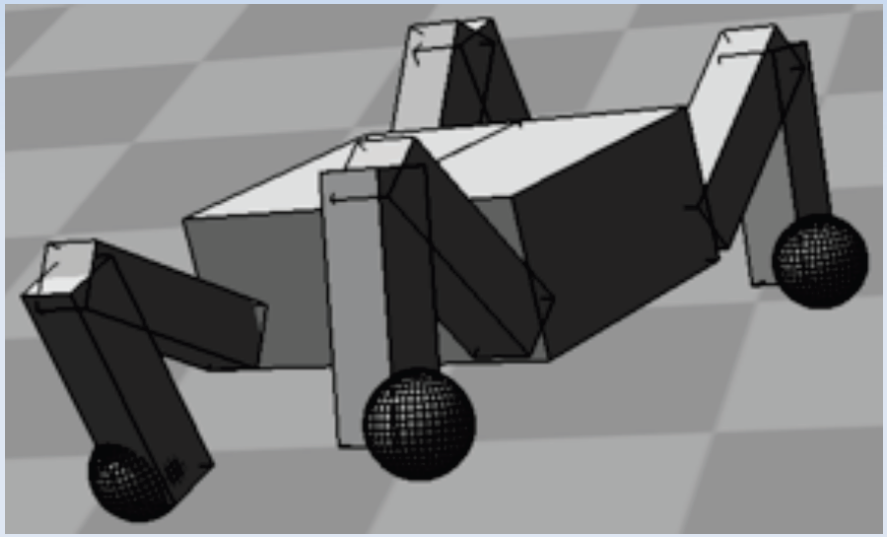


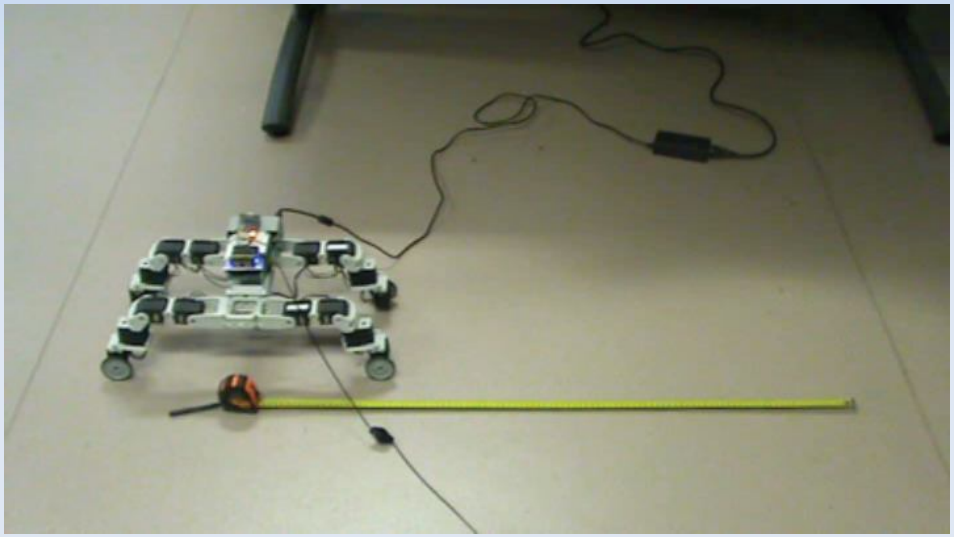
Applications to robot off-line learning

Work of Jean-Baptiste Mouret et al. @ ISIR

The reality gap

Self-model vs reality: how to use a simulator?





Solution: Learn a transferability function (how well does the simulation match reality?) with SVM or neural networks.

Idea: the damage is a large reality gap.

Koos, Mouret & Doncieux. IEEE Trans Evolutionary Comput 2012

Applications to robot off-line learning

Work of Jean-Baptiste Mouret et al. @ ISIR

Experiments

Koos, Cully & Mouret. Int J Robot Res 2013

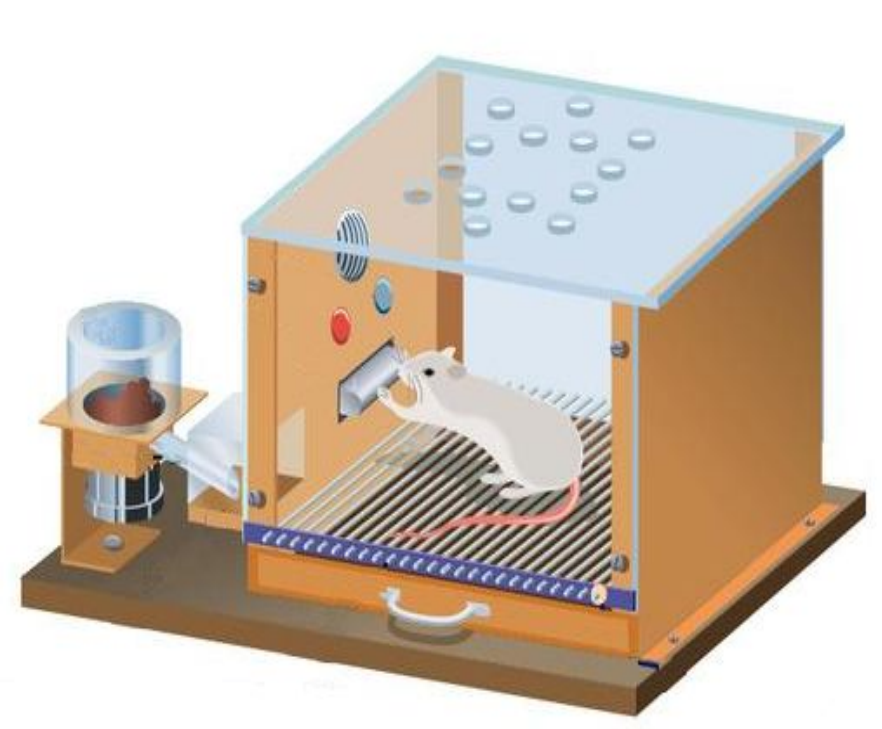
META-LEARNING

(regulation of decision-making)

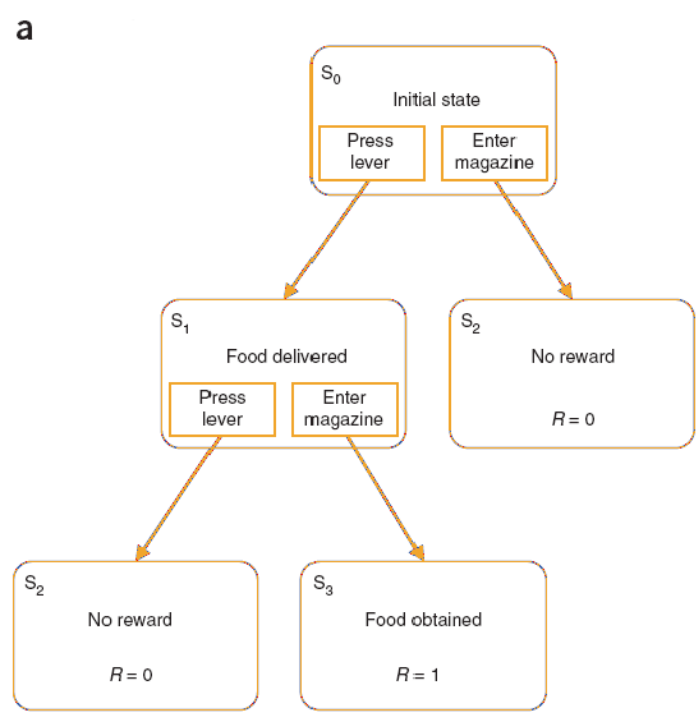
1. Dual-system RL coordination
2. Online parameters tuning

Multiple decision systems

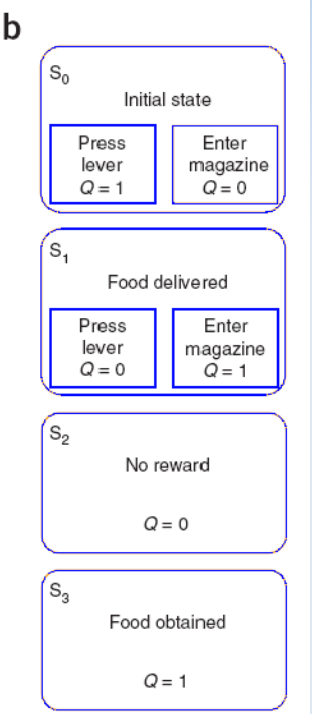
Skinner box (instrumental conditioning)



Model-based system



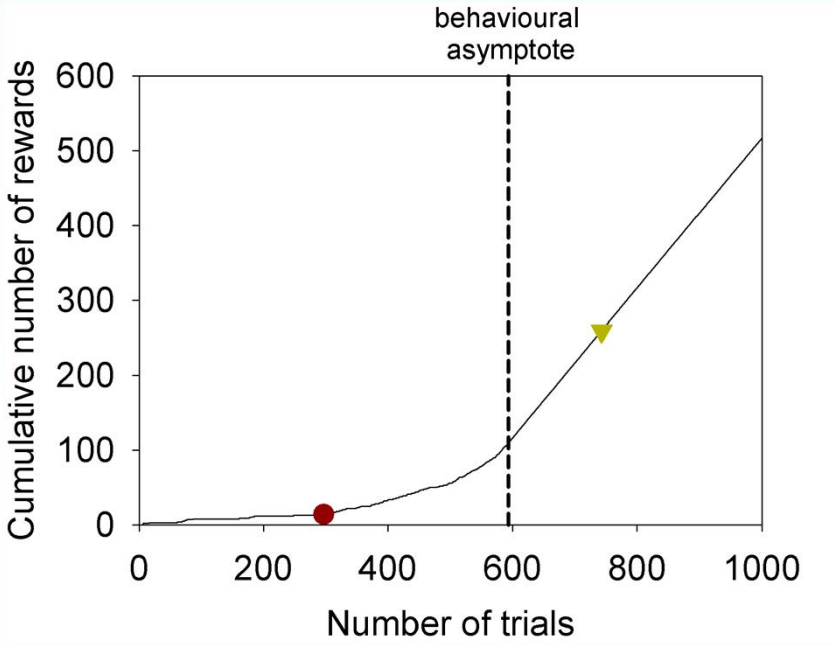
Model-free sys.



(Daw Niv Dayan 2005, Nat Neurosci)

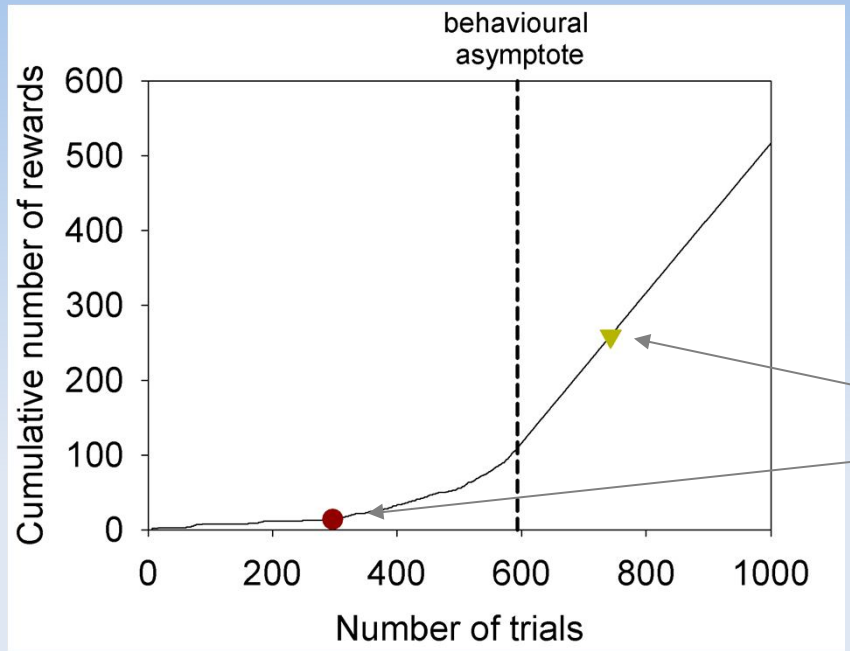
Behavior is initially model-based and becomes model-free (habitual) with overtraining.

Habitual vs goal-directed: sensitive to changes in outcome



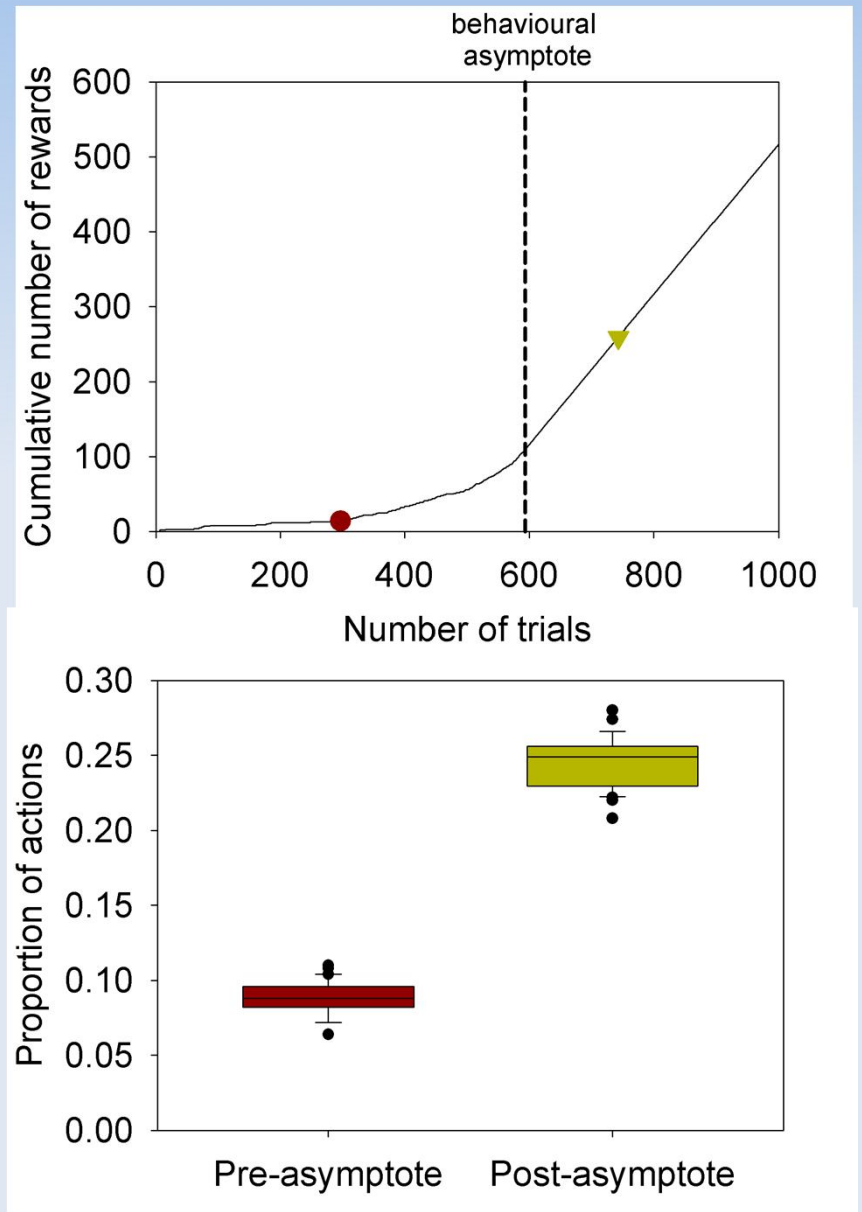
Yin et al. 2004; Balleine 2005; Yin & Knowlton 2006

Habitual vs goal-directed: sensitive to changes in outcome



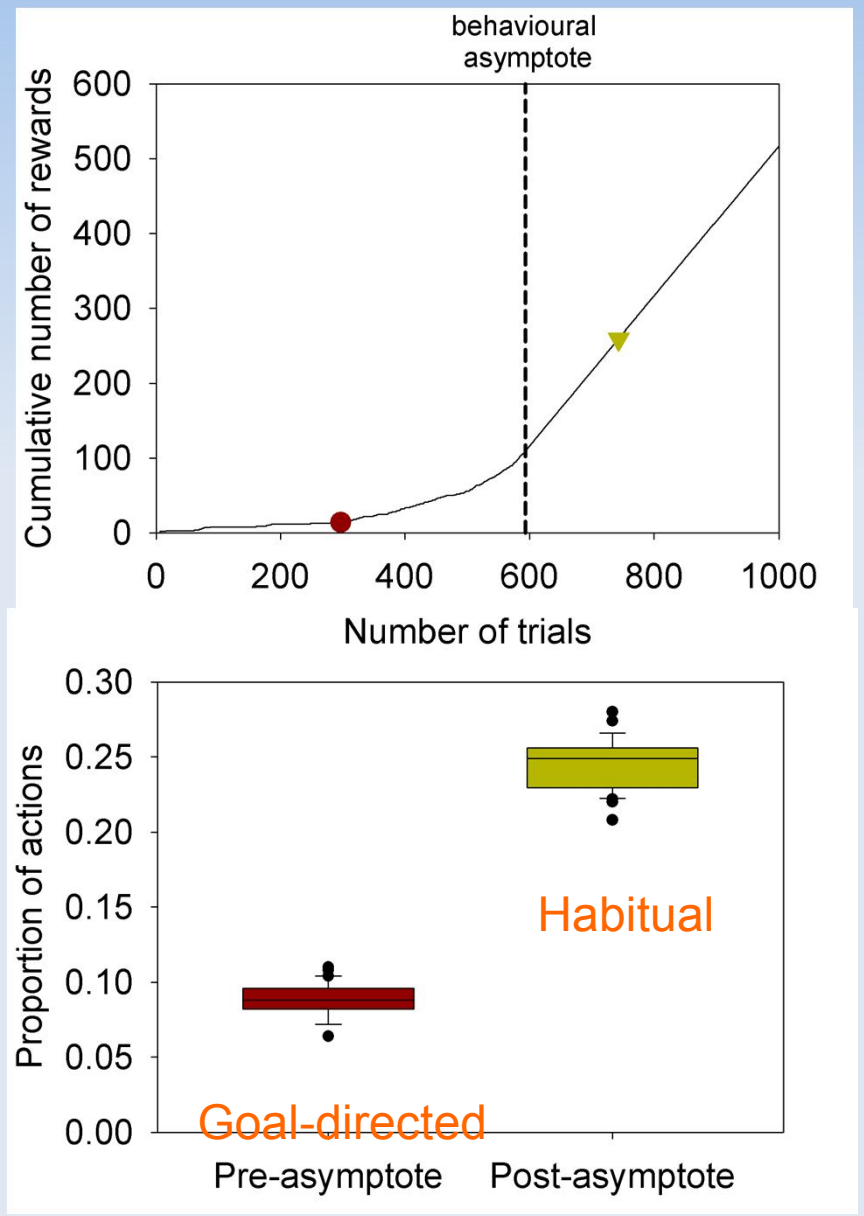
Yin et al. 2004; Balleine 2005; Yin & Knowlton 2006

Habitual vs goal-directed: sensitive to changes in outcome



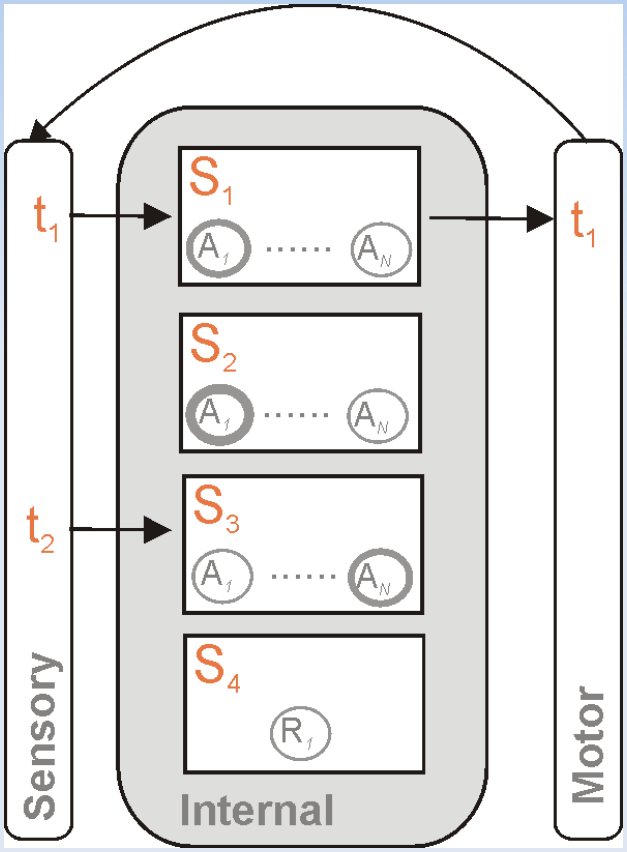
Yin et al. 2004; Balleine 2005; Yin & Knowlton 2006

Habitual vs goal-directed: sensitive to changes in outcome



Yin et al. 2004; Balleine 2005; Yin & Knowlton 2006

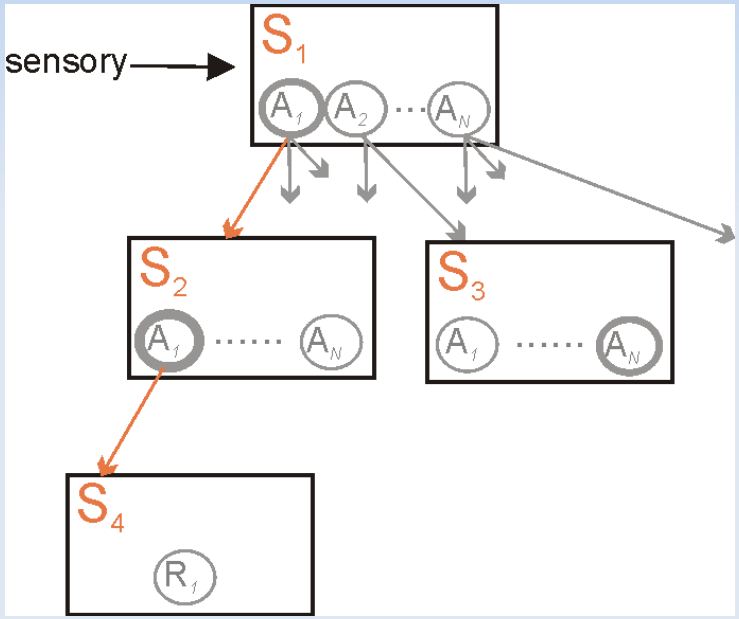
Model-free vs model-based: outcome sensitivity



Change R: slow to update

Habitual

Switch with experience [reduce computational load]



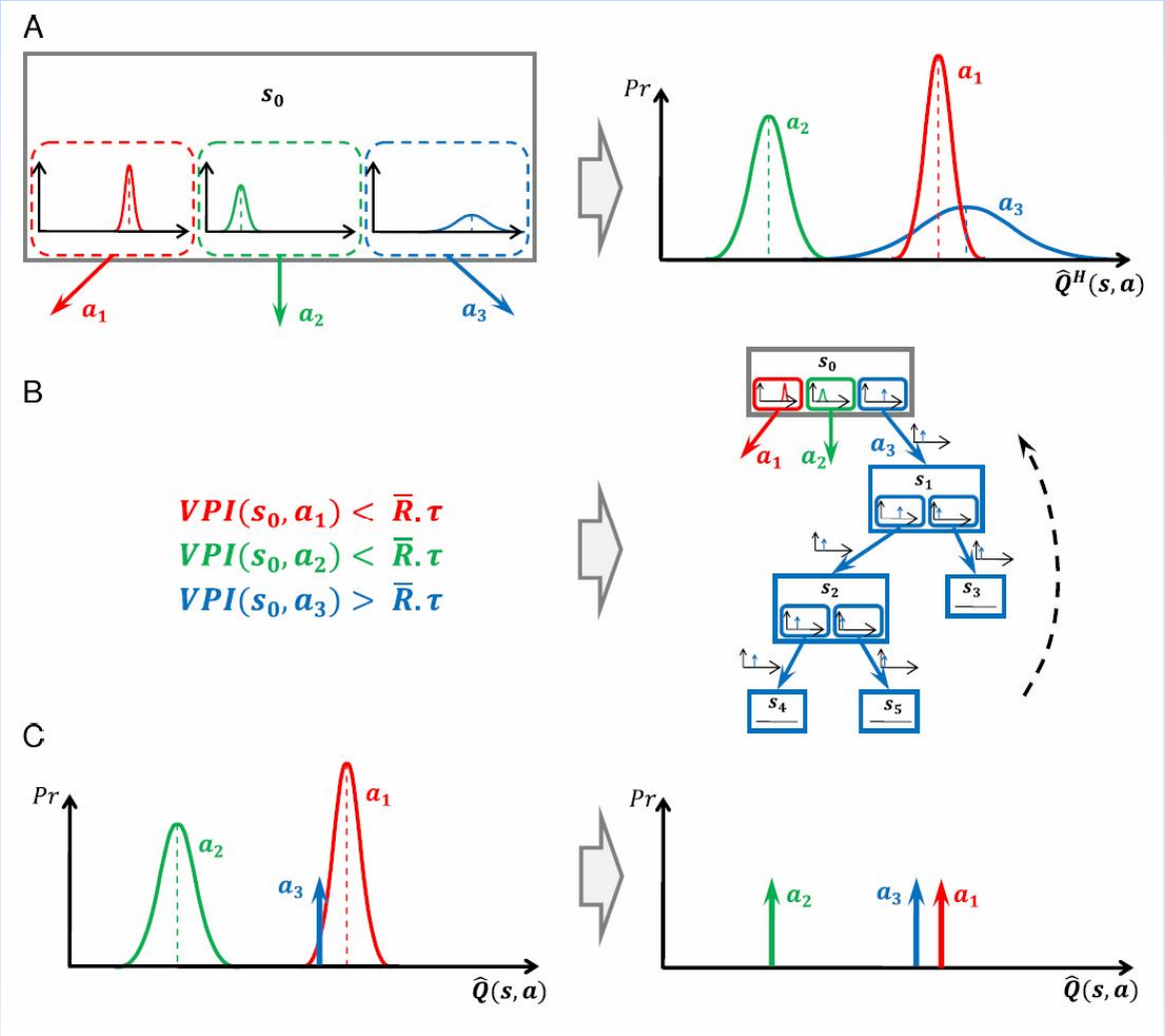
Change R: fast to update

Goal-directed

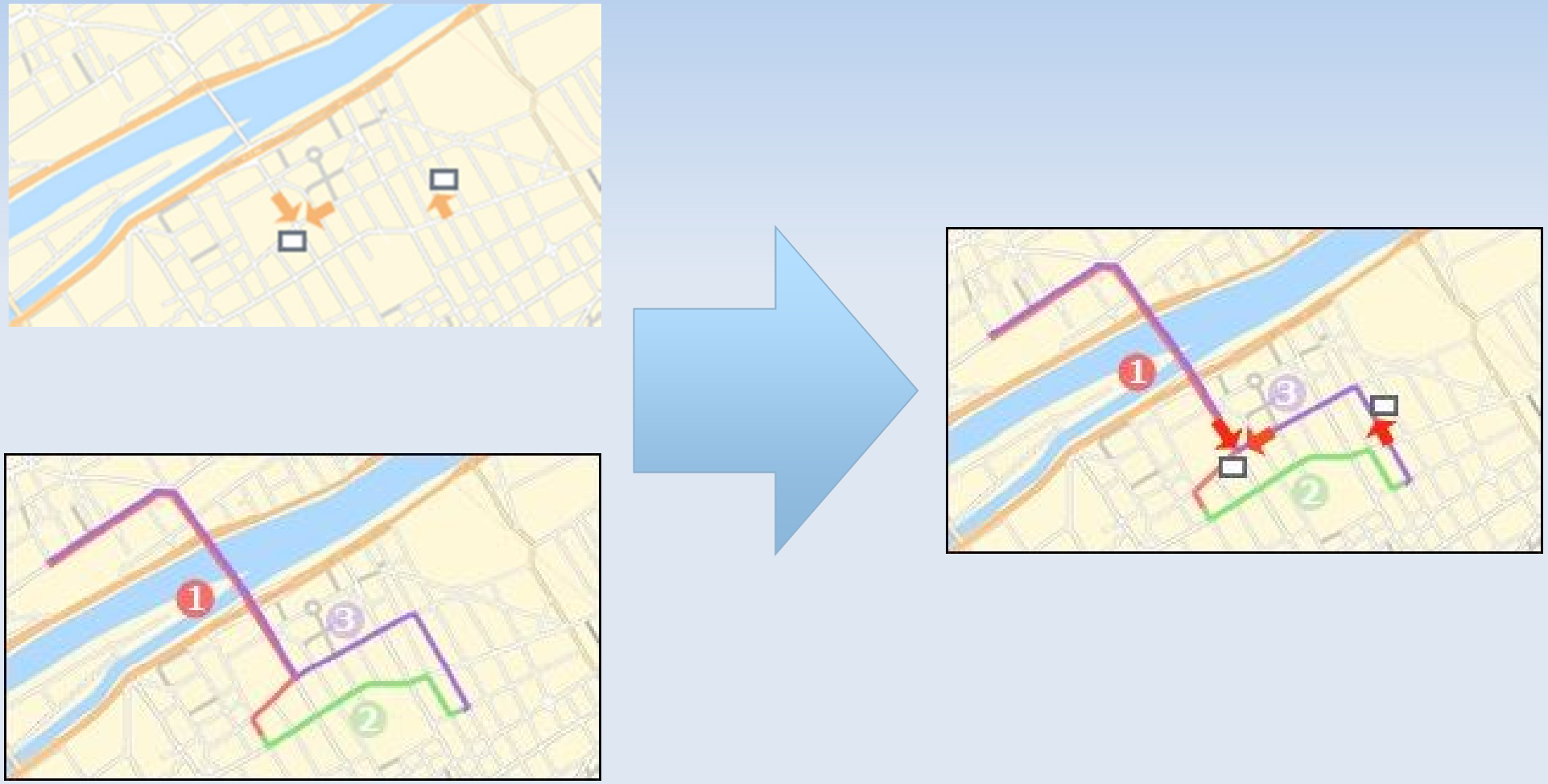
Daw et al 2005 Nat Neurosci

Multiple decision systems

Keramati et al. (2011): extension of the Daw 2005 model with a speed-accuracy trade-off arbitration criterion.



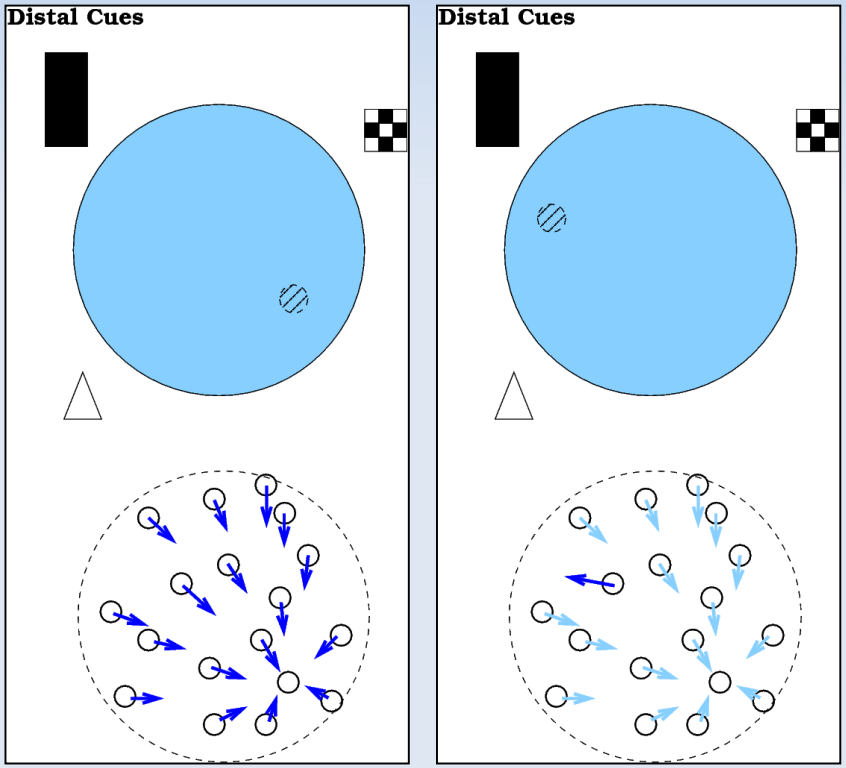
Progressive shift from model-based navigation to model-free navigation



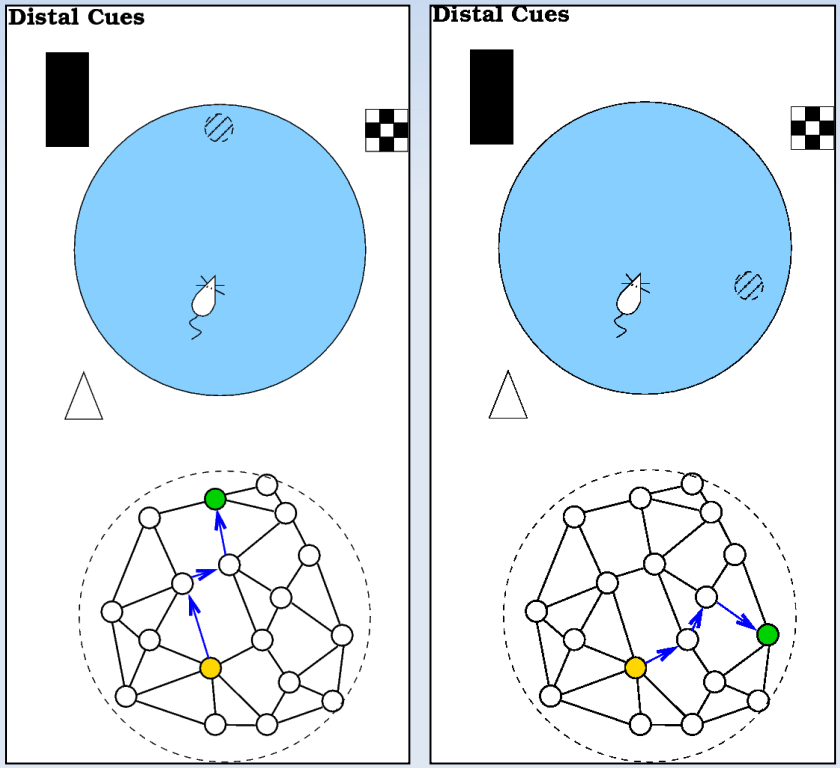
Khamassi & Humphries (2012) Frontiers in Behavioral Neuroscience

Model-based and model-free navigation strategies

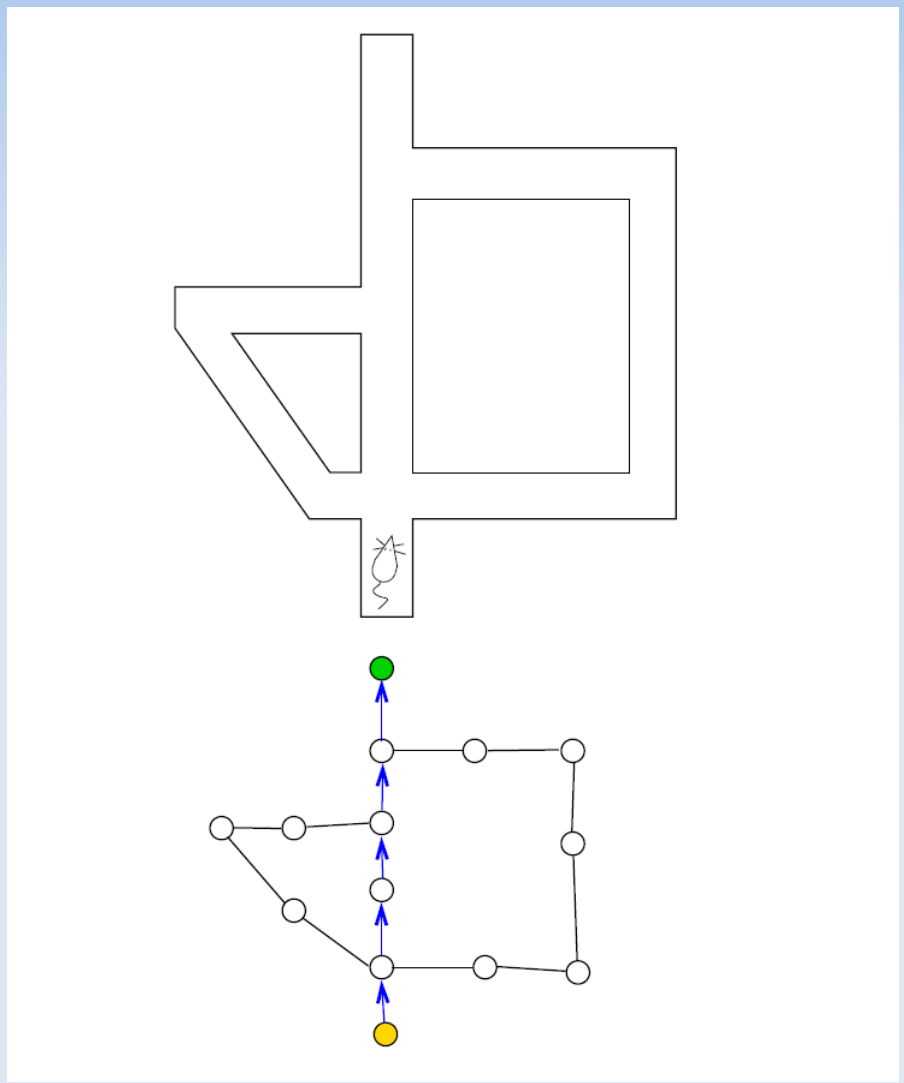
Model-based navigation



Model-free navigation

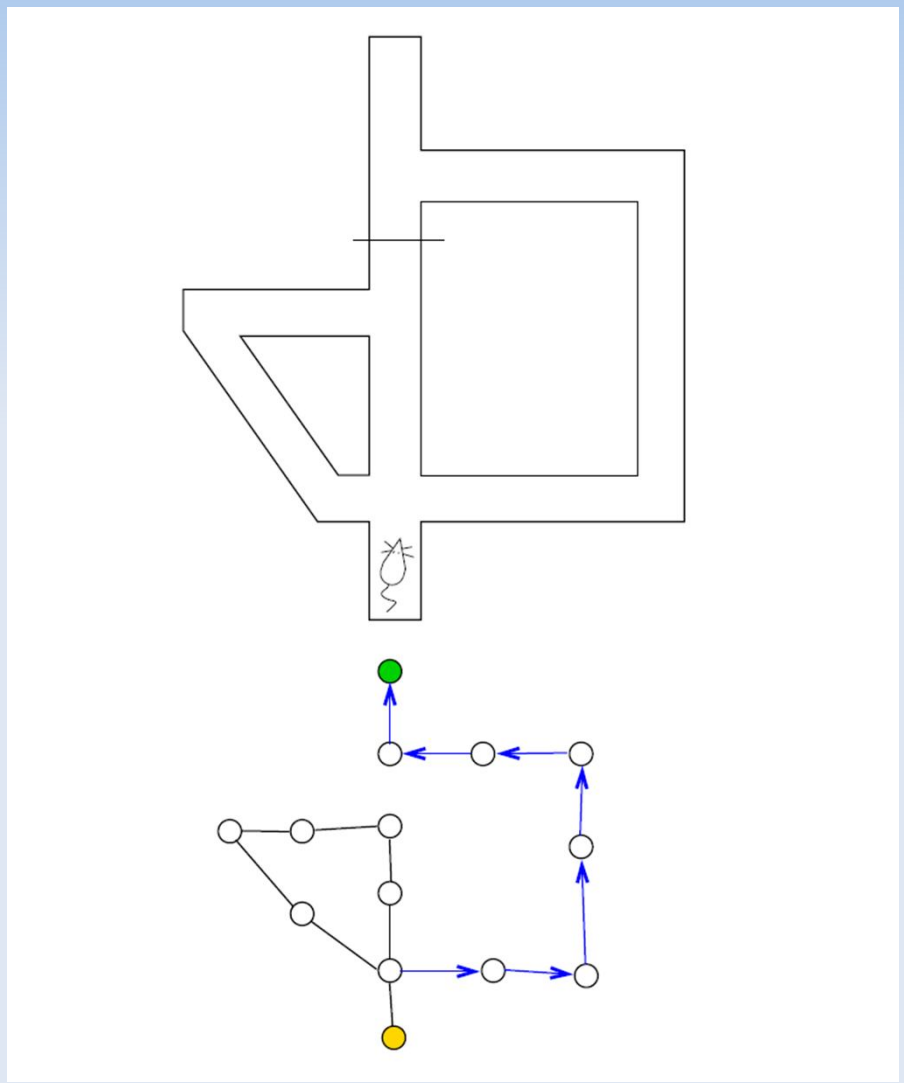


Old behavioral evidence for Place-based model-based RL



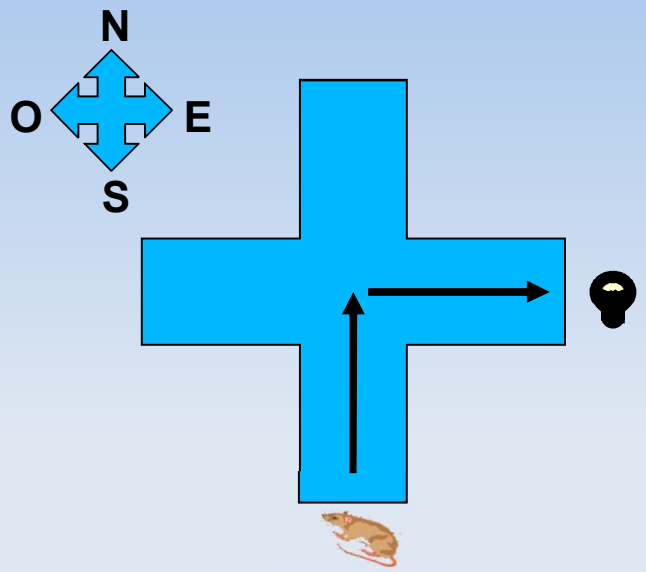
Martinet et al. (2011) model applied to the Tolman maze

Old behavioral evidence for Place-based model-based RL

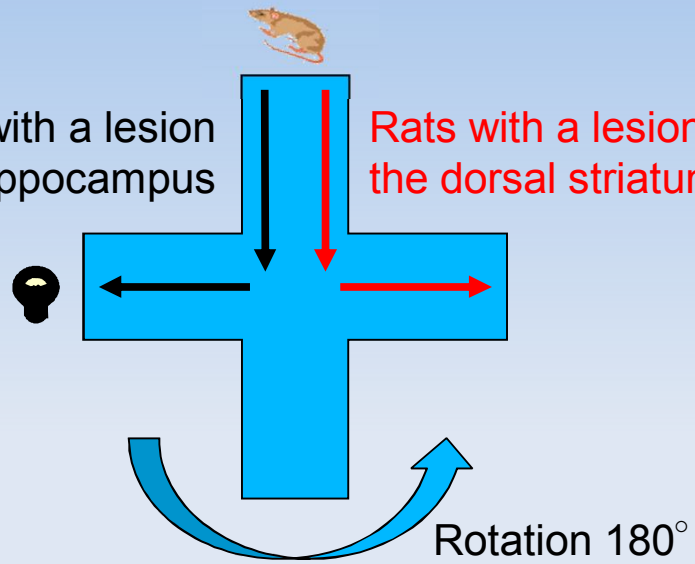


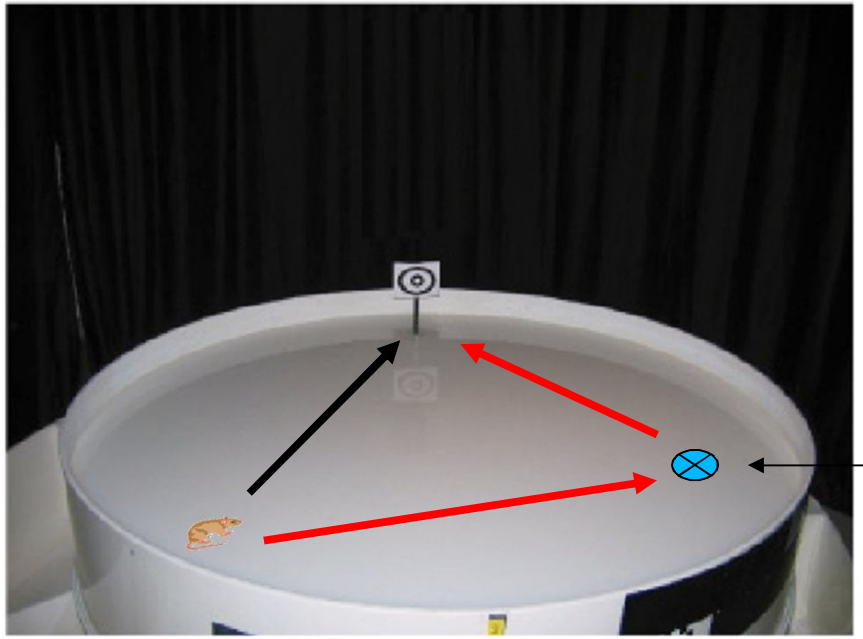
Martinet et al. (2011) model applied to the Tolman maze

MULTIPLE NAVIGATION STRATEGIES IN THE RAT



Packard and Knowlton, 2002



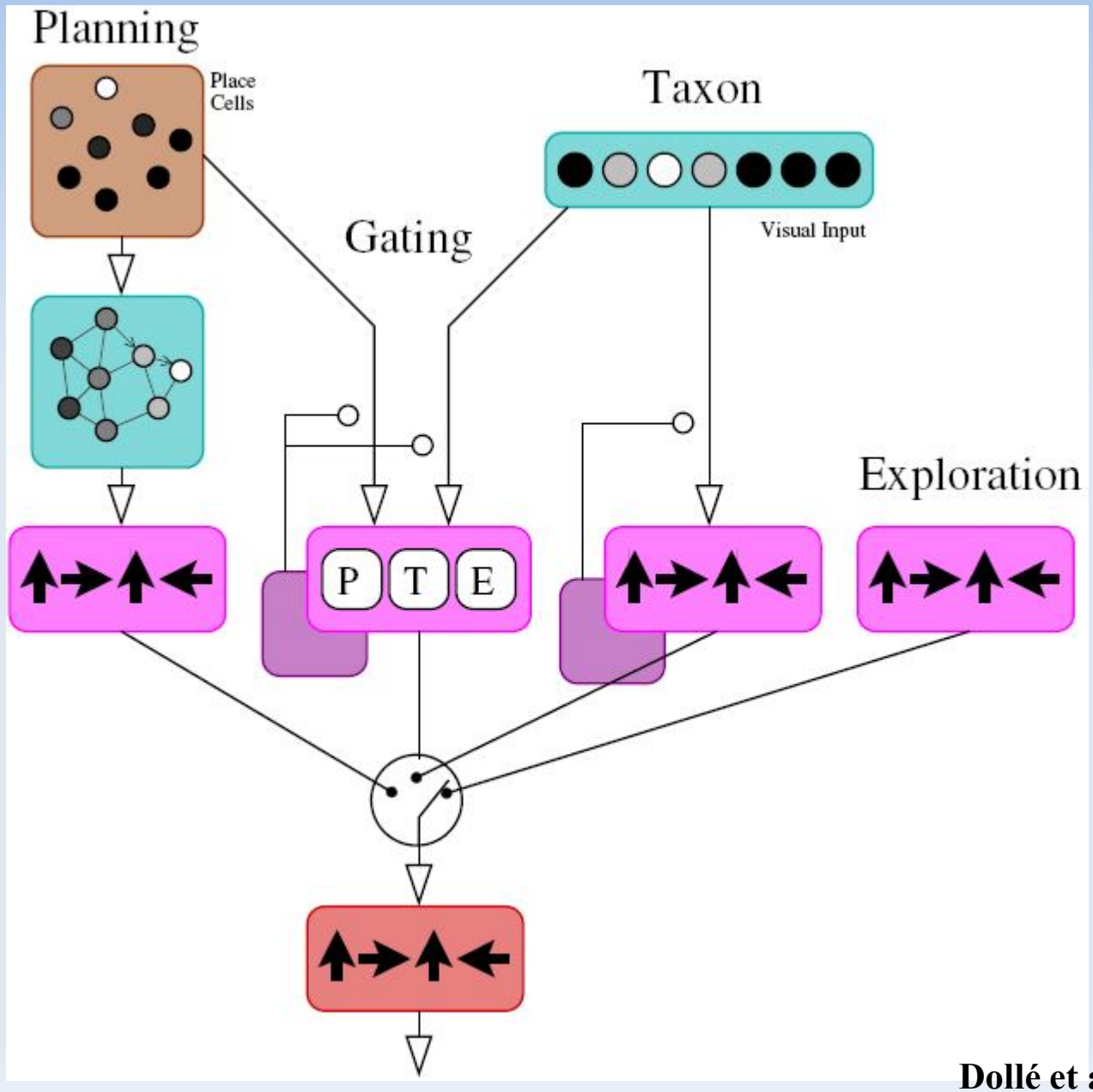


Devan and White, 1999

MULTIPLE DECISION SYSTEMS IN A NAVIGATION MODEL

Model-based system

(hippocampal place cells)



Model-free system

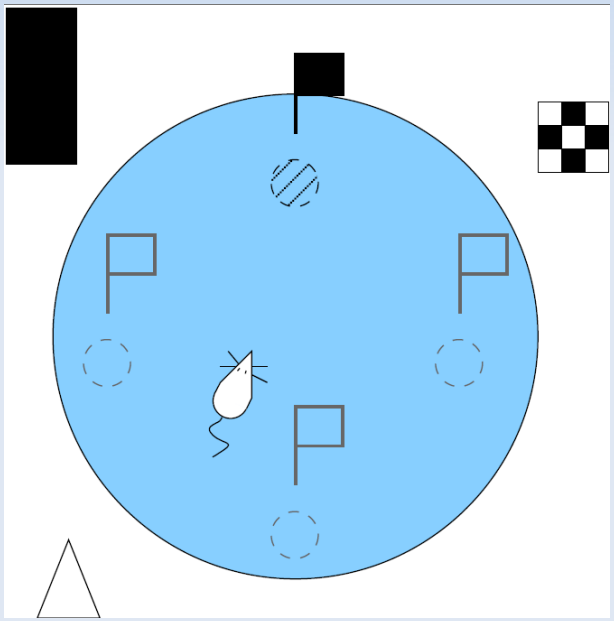
(basal ganglia)

Work by Laurent Dollé:

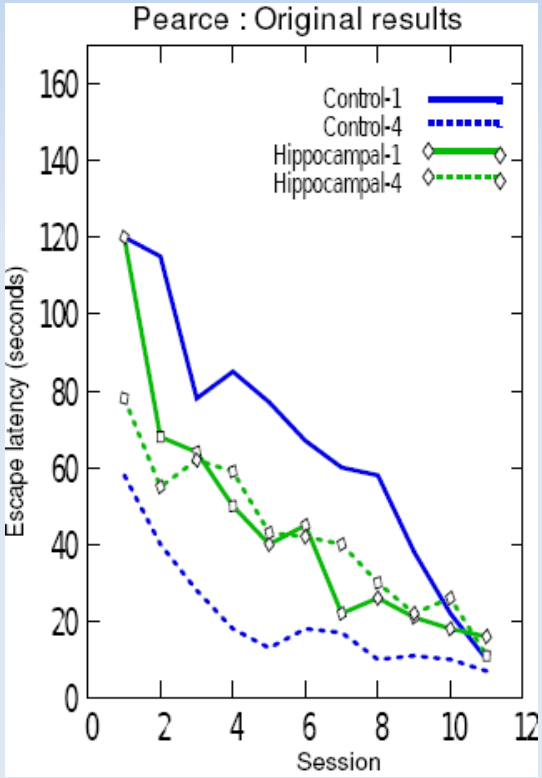
Dollé et al., 2008, 2010, submitted

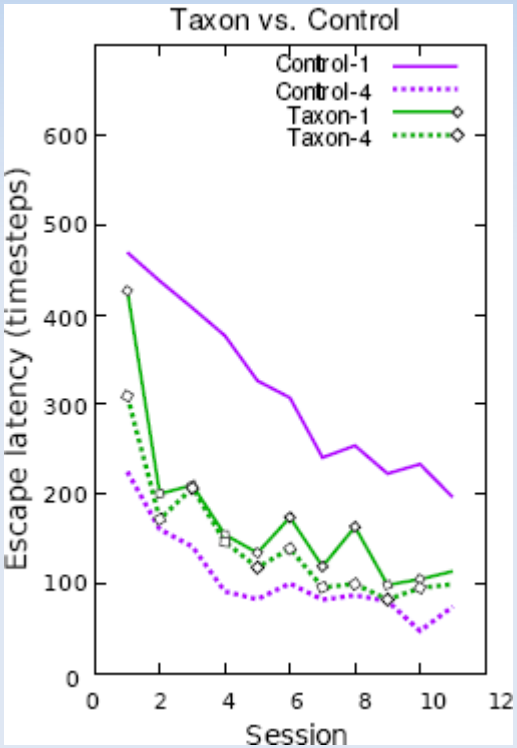
MULTIPLE NAVIGATION STRATEGIES IN A TD-LEARNING MODEL

Task with a cued platform (visible flag) changing location every 4 trials



Task of Pearce et al., 1998

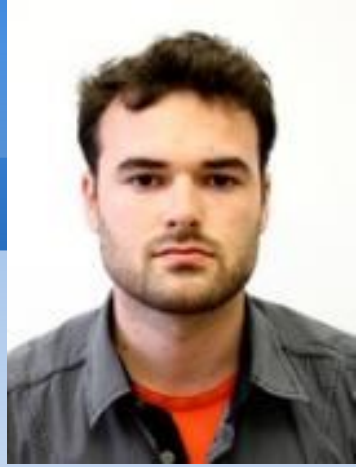


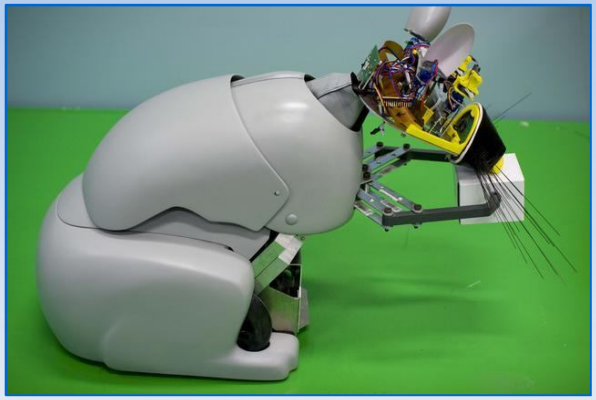


Model:

Dollé et al., 2010

PSIKHARPAX ROBOT





Work by:

Ken Caluwaerts (2010)

Steve N'Guyen (2010)

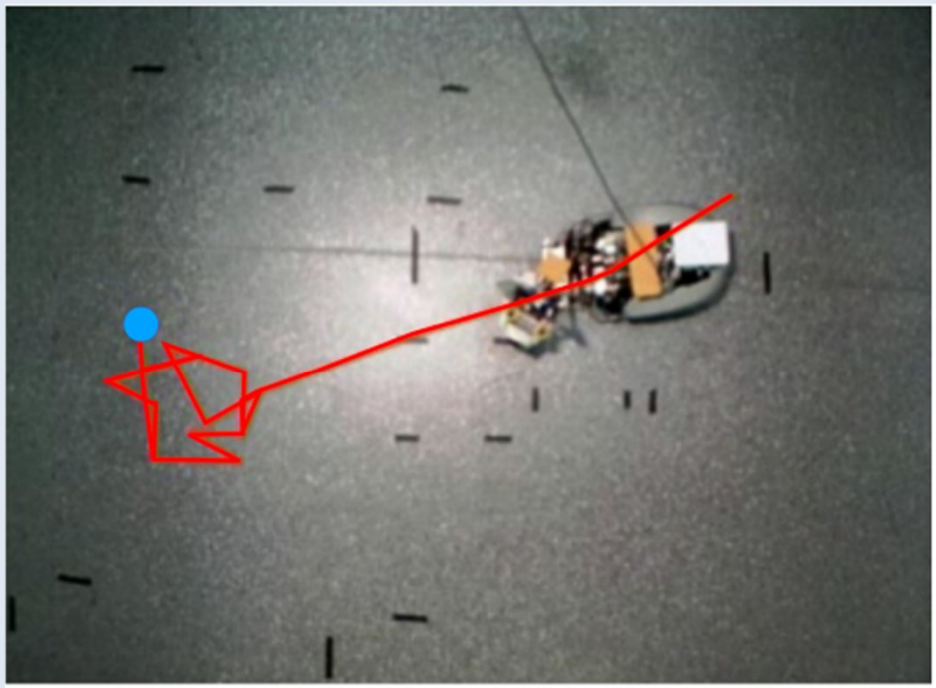
Mariacarla Staffa (2011)

Antoine Favre-Félix (2011)

Caluwaerts et al. (2012) Biomimetics & Bioinspiration

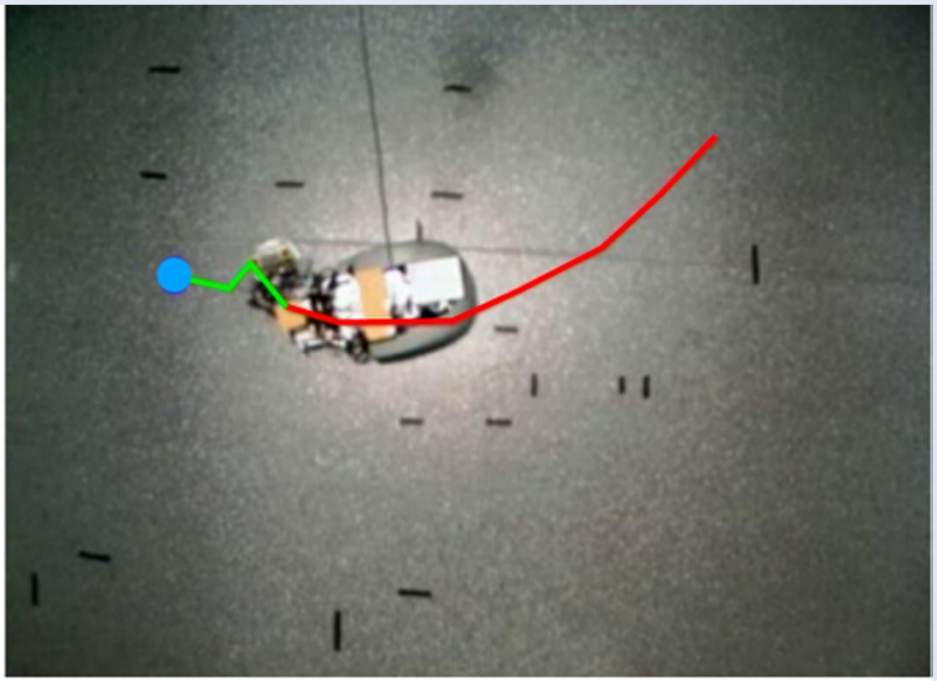
PSIKHARPAX ROBOT

Planning strategy only



(*a*)

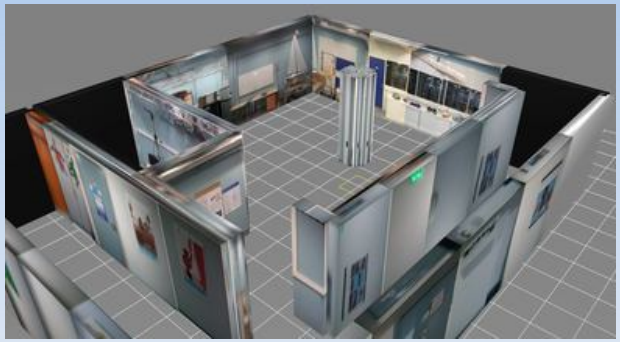
Planning strategy + Taxon strategy



(*b*)

Caluwaerts et al. (2012) Biomimetics & Bioinspiration

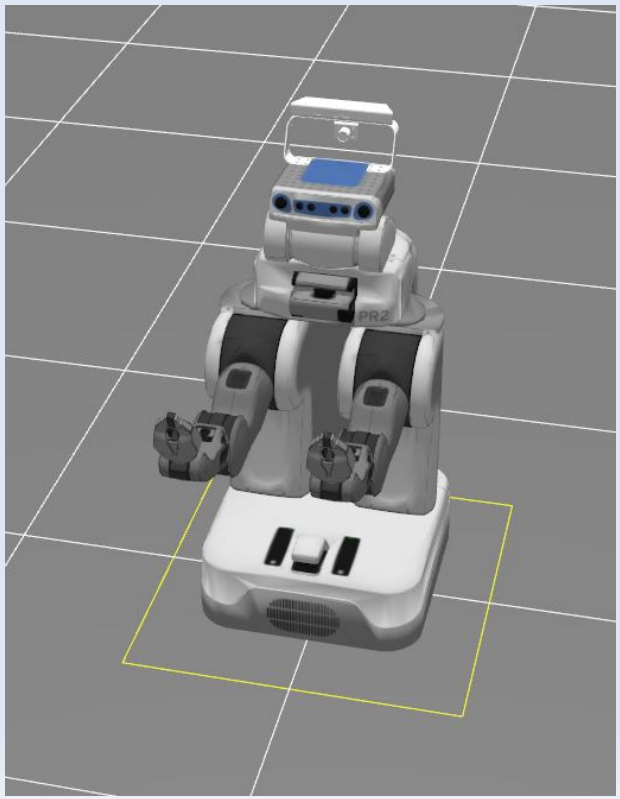
CURRENT APPLICATIONS TO THE PR2 ROBOT

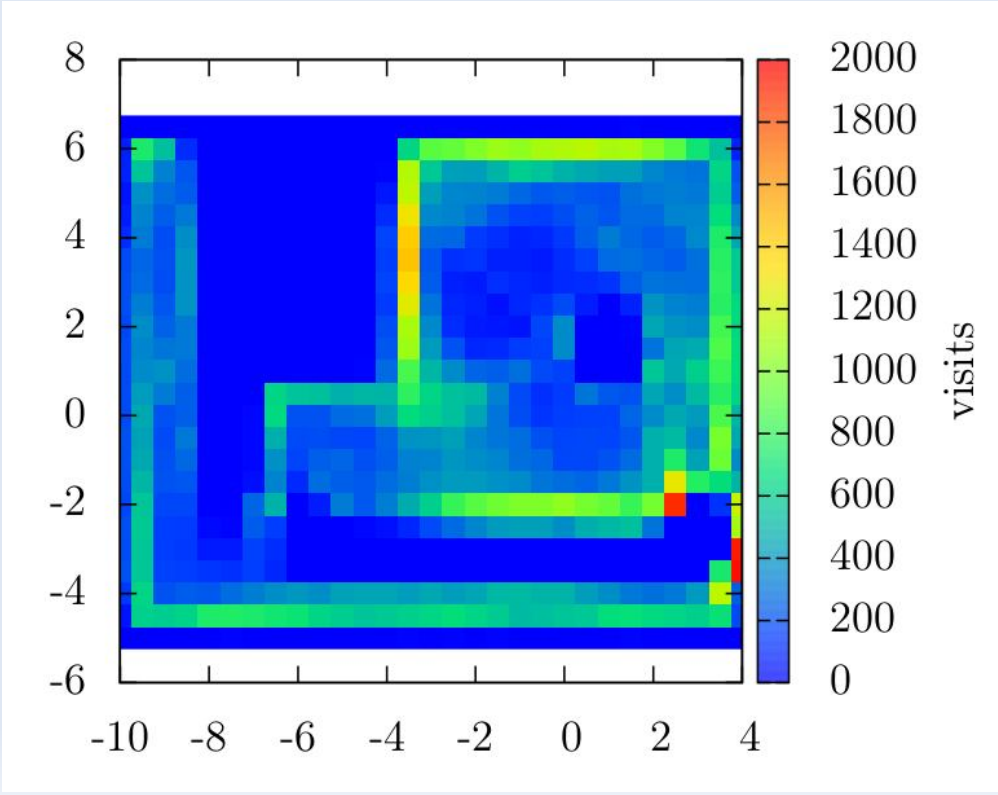


Travaux de :

Erwan Renaudo

Omar Islas Ramirez





CURRENT APPLICATIONS TO HUMAN-ROBOT INTERACTION

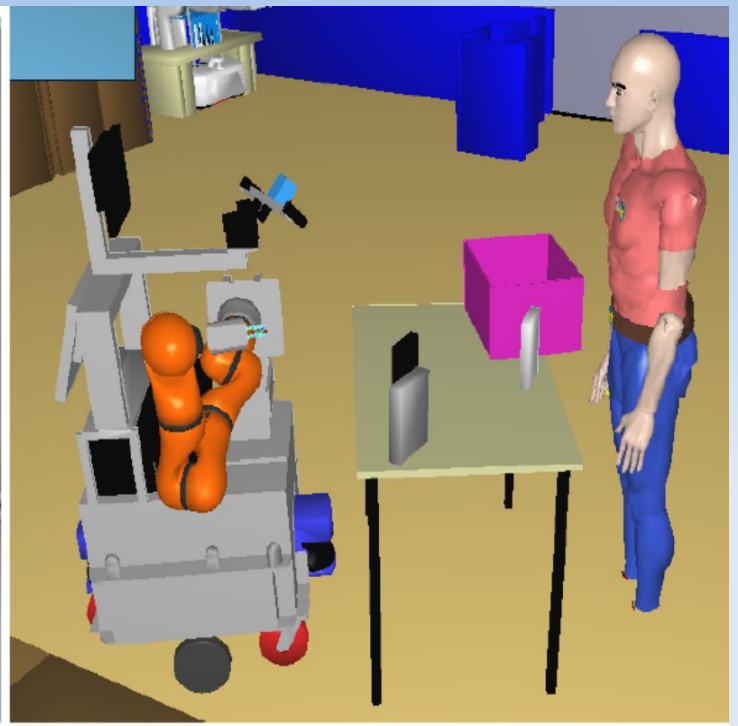
Travaux de :

Erwan Renaudo

Collaboration :

Alami et al (LAAS)

(a) Initial state

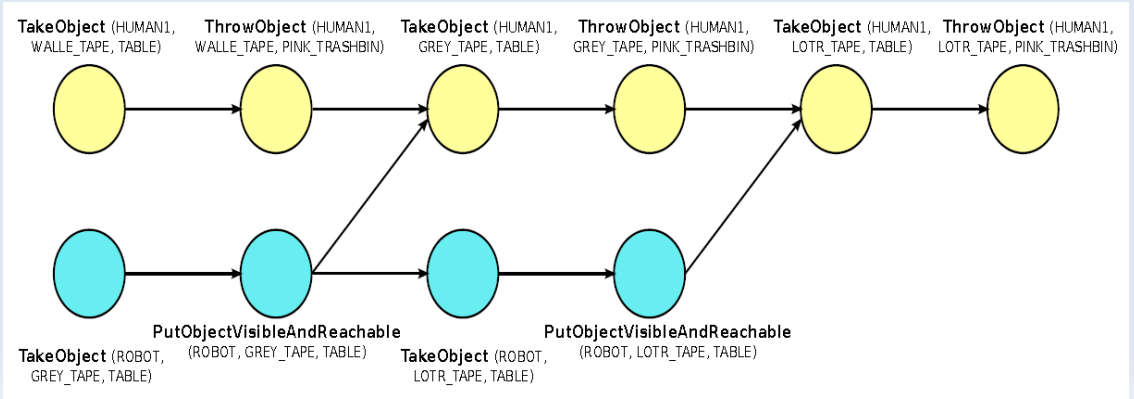


(b) 3d model view of initial state

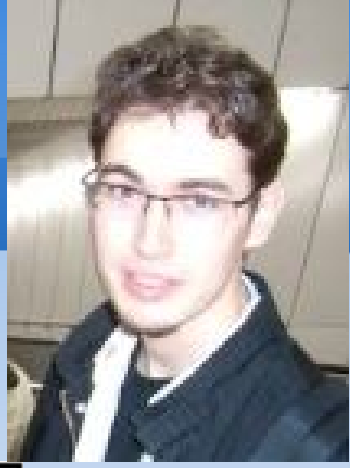
Task: Clean the table

Current state: A priori given action plan (right image)

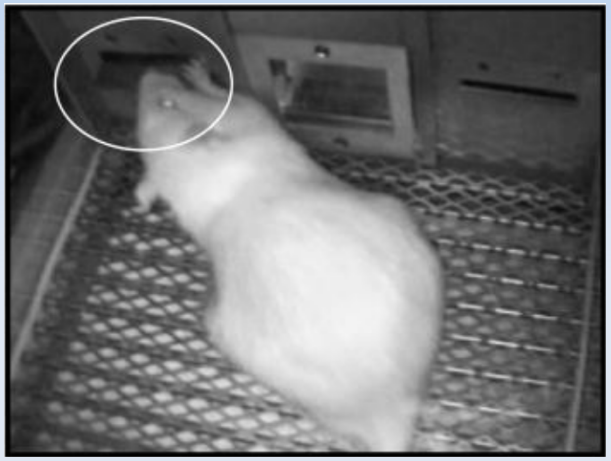
Goal: Autonomous learning by the robot



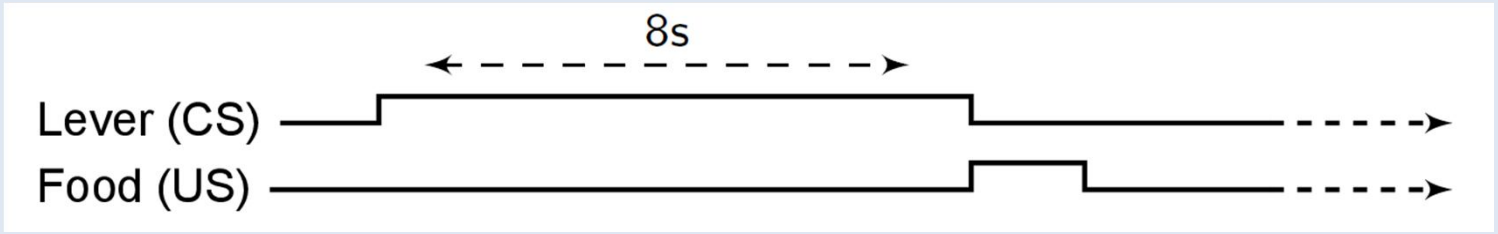
Pavlovian autoshaping



Sign-trackers



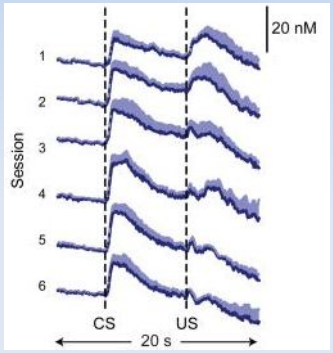
Goal-trackers



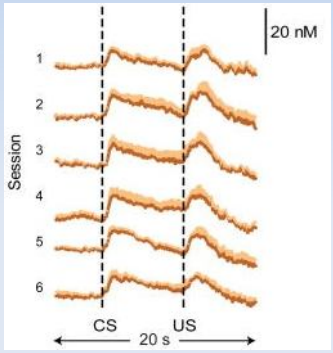
Flagel et al. (2011). “A selective role for dopamine in stimulus-reward learning”. Nature, 469:53:7.

Pavlovian autoshaping

Sign-trackers



Goal-trackers

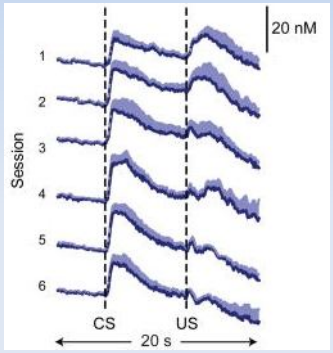


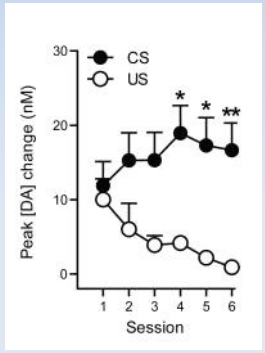
Fast Scan Cyclic Voltammetry (FSCV) in the ventral striatum.

Flagel et al. (2011). “A selective role for dopamine in stimulus-reward learning”. Nature, 469:53:7.

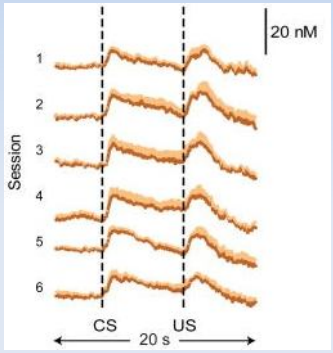
Pavlovian autoshaping

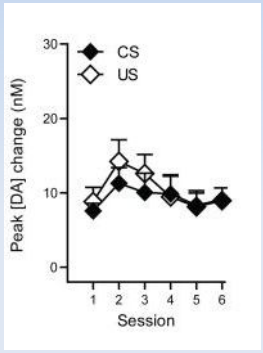
Sign-trackers





Goal-trackers



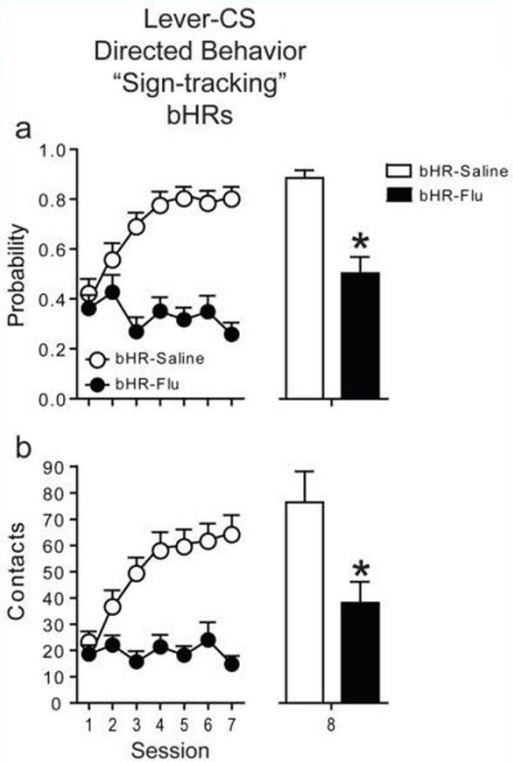


Fast Scan Cyclic Voltammetry (FSCV) in the ventral striatum.

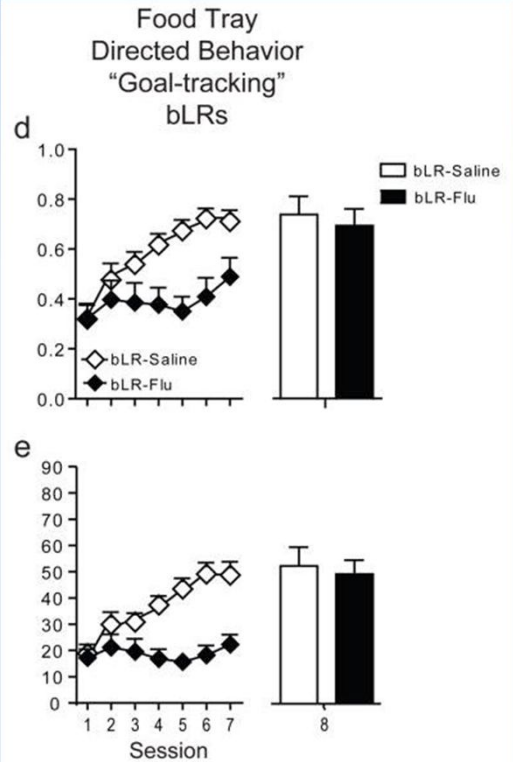
Flagel et al. (2011). “A selective role for dopamine in stimulus-reward learning”. Nature, 469:53:7.

Pavlovian autoshaping

Sign-trackers



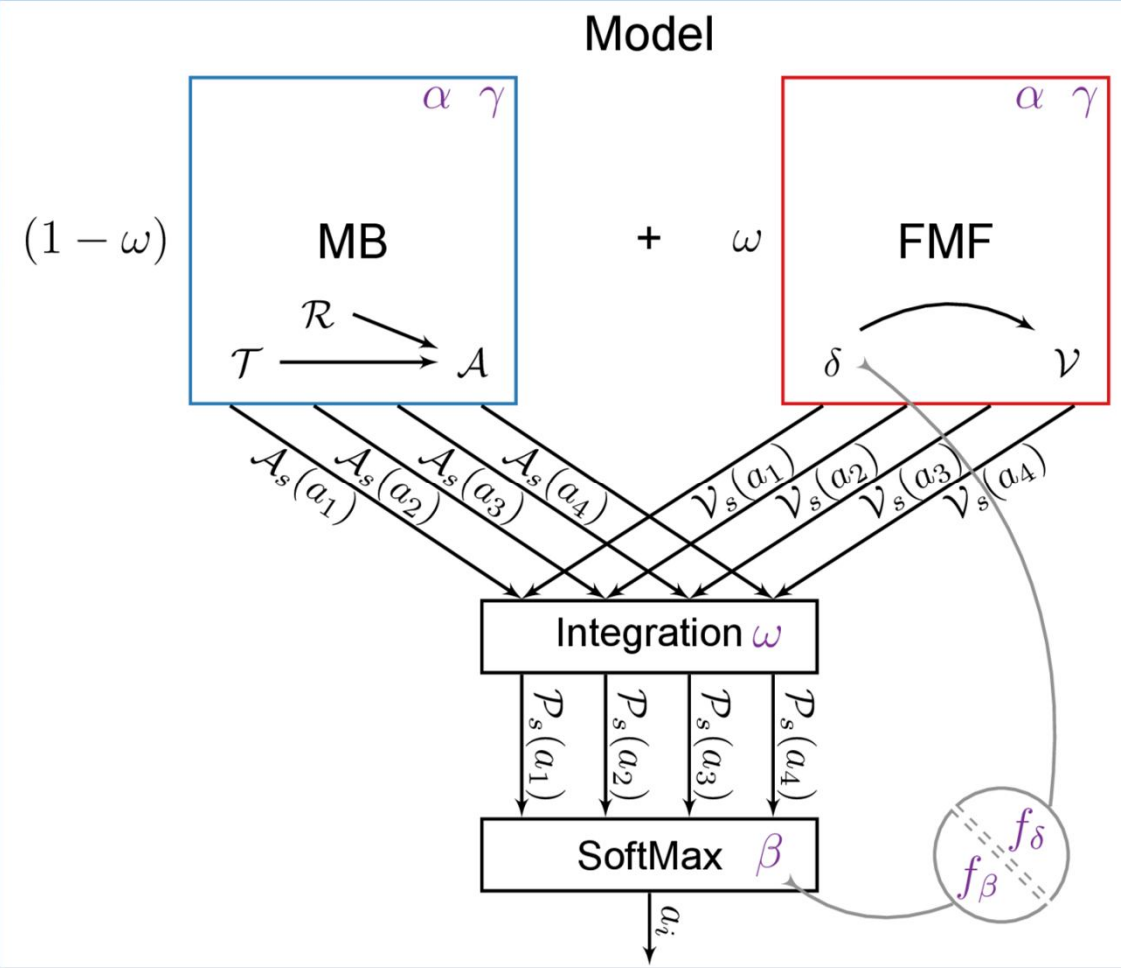
Goal-trackers



Systemic injection of flupentixol prior to each session.

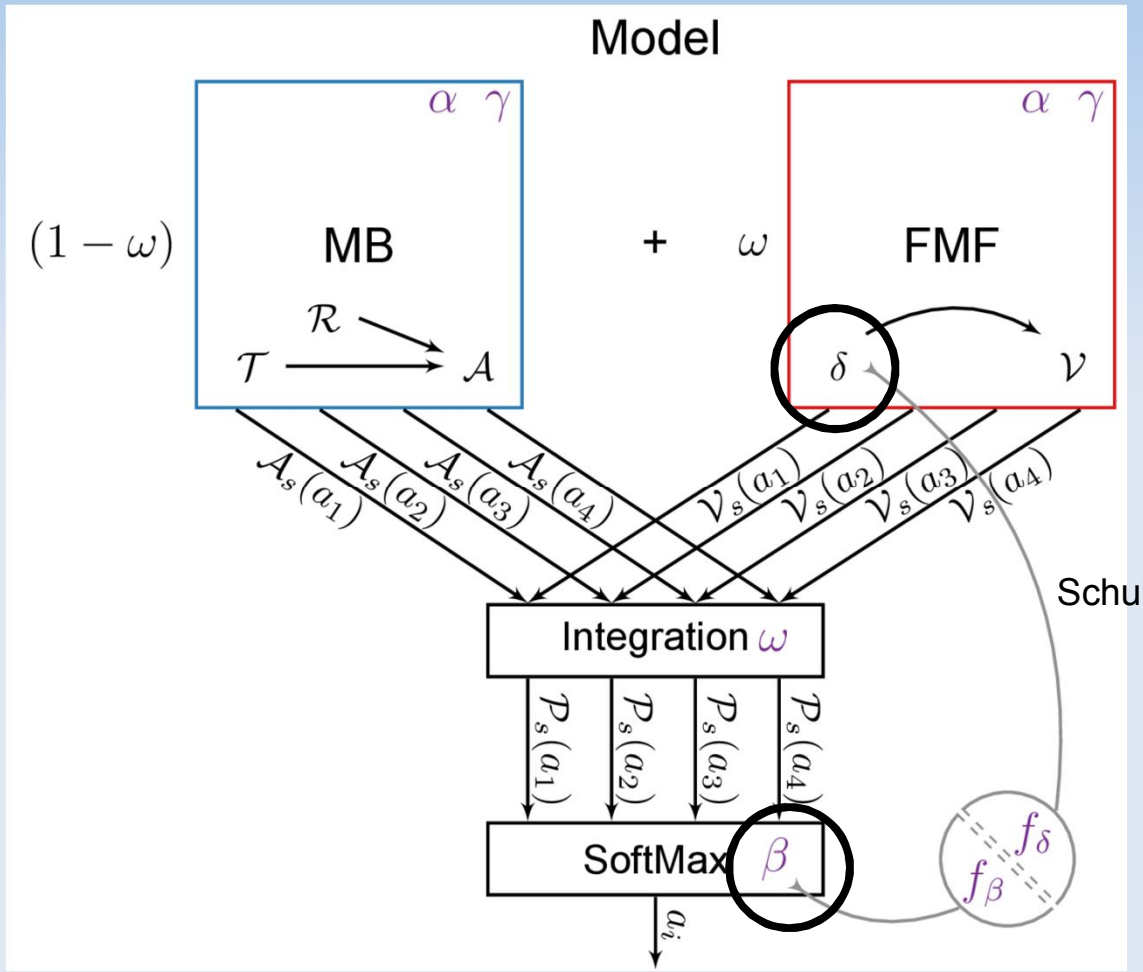
Flagel et al. (2011). "A selective role for dopamine in stimulus-reward learning". Nature, 469:53:7.

Computational model



Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model



Schultz et al. (1997)

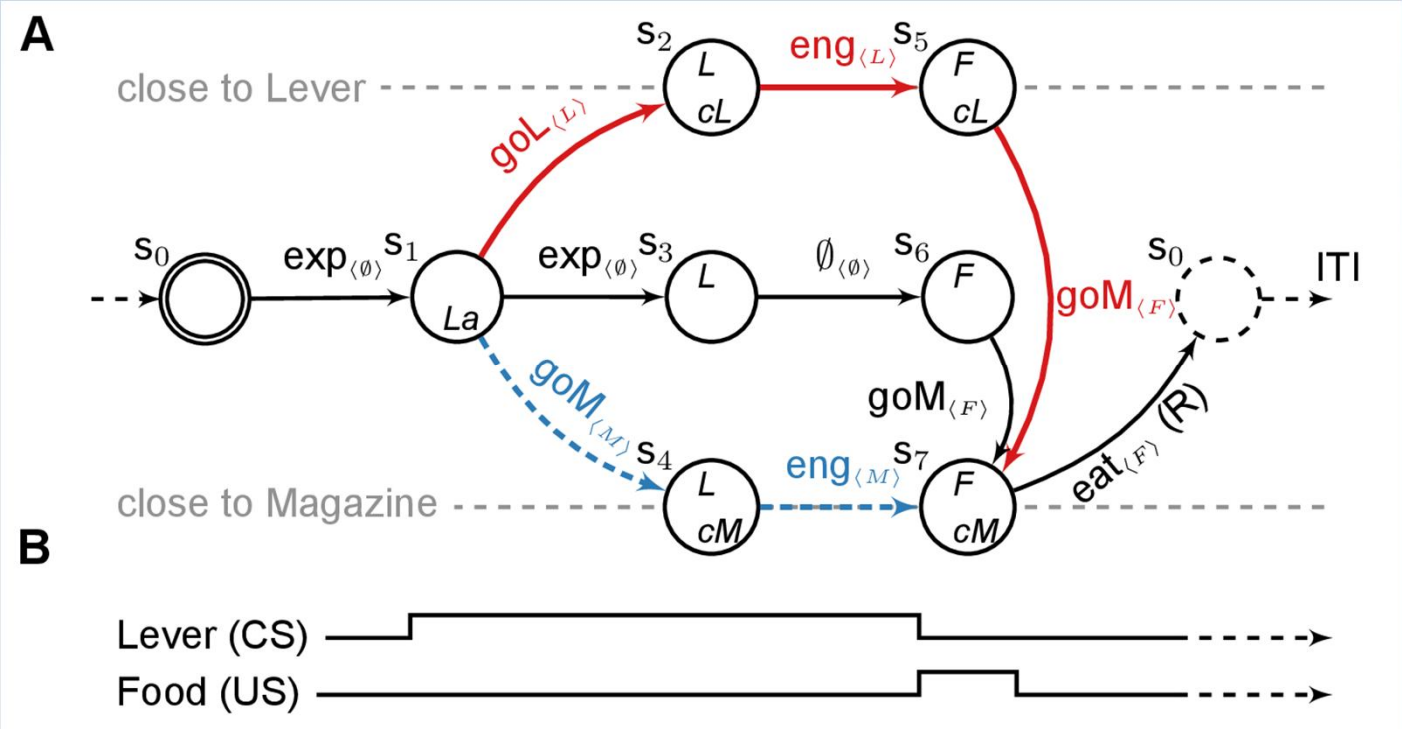
Dopamine

McClure et al. (2003);
Humphries et al. (2012)

Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

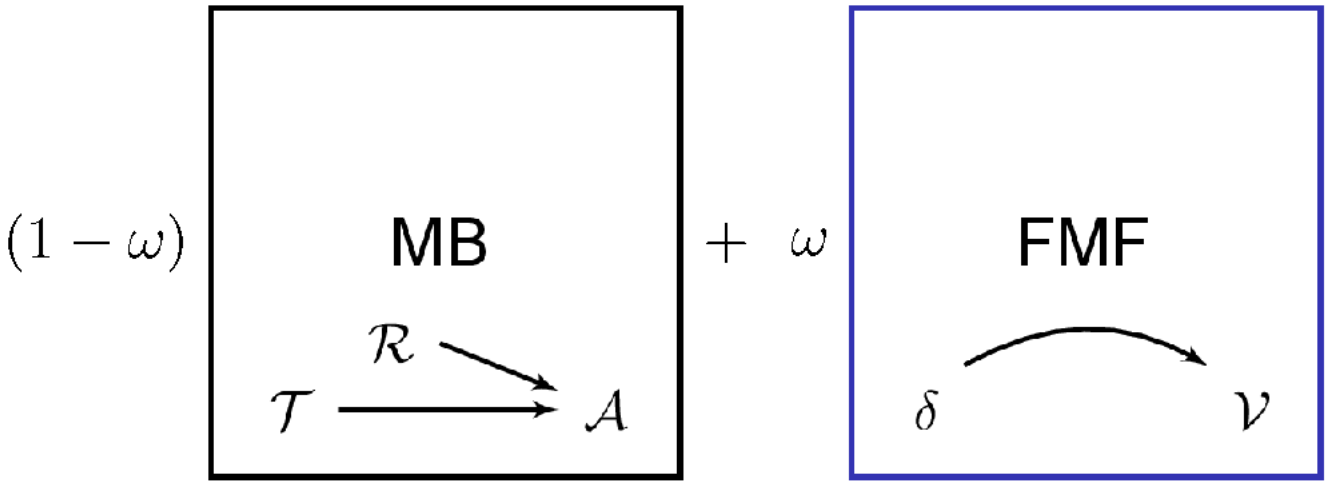
Computational model

Modelling the task as a Markov Decision Process



Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

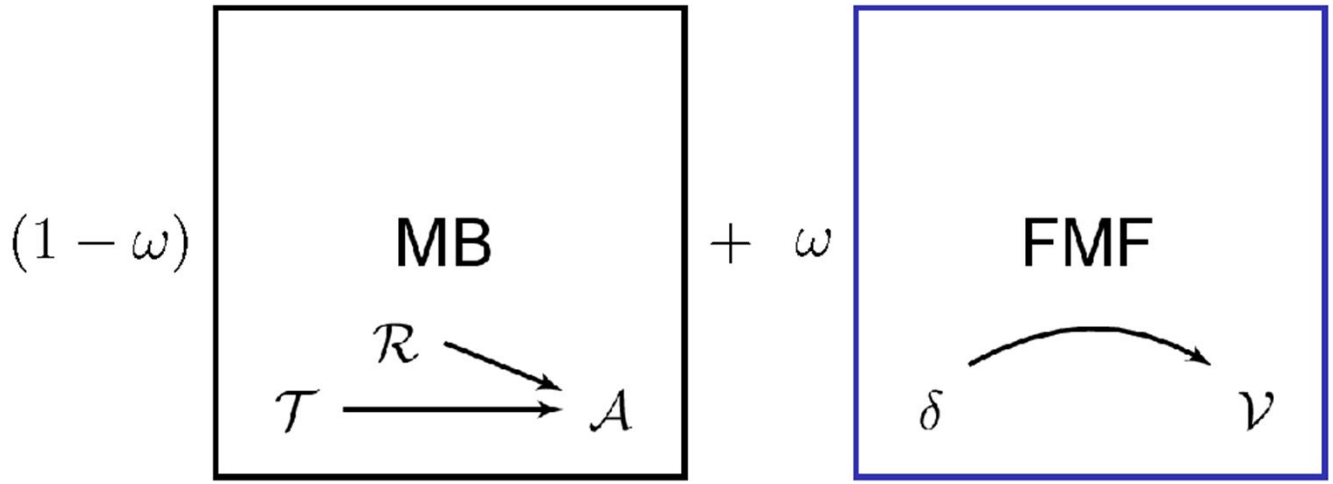


$$\begin{cases} \mathcal{P}_{goL}(s_1) = (1-\omega)\ (\mathcal{Q}_{goL}(s_1) - \max_{a'} \mathcal{Q}_{a'}(s_1)) + \omega\ \mathcal{V}(L) \\ \mathcal{P}_{goM}(s_1) = (1-\omega)\ (\mathcal{Q}_{goM}(s_1) - \max_{a'} \mathcal{Q}_{a'}(s_1)) + \omega\ \mathcal{V}(M) \end{cases}$$

with ω = 0.499 (STs), ω = 0.048 (GTs), ω = 0.276 (IGs)

Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model



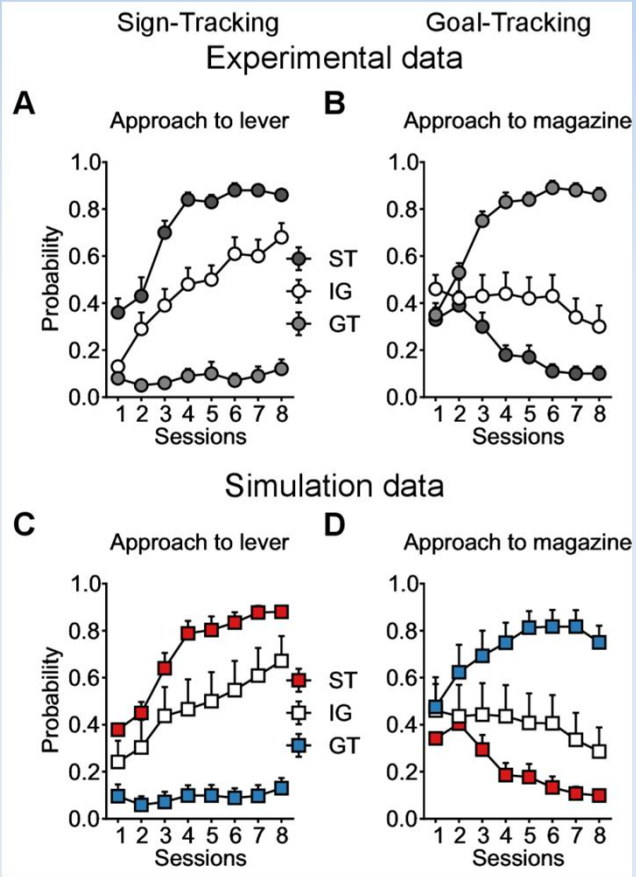
$$\begin{cases} \mathcal{P}_{goL}(s_1) = (1-\omega)\ (\mathcal{Q}_{goL}(s_1) - \max_{a'} \mathcal{Q}_{a'}(s_1)) + \omega\ \mathcal{V}(L) \\ \mathcal{P}_{goM}(s_1) = (1-\omega)\ (\mathcal{Q}_{goM}(s_1) - \max_{a'} \mathcal{Q}_{a'}(s_1)) + \omega\ \mathcal{V}(M) \end{cases}$$

with ω = 0.499 (STs), ω = 0.048 (GTs), ω = 0.276 (IGs)

Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

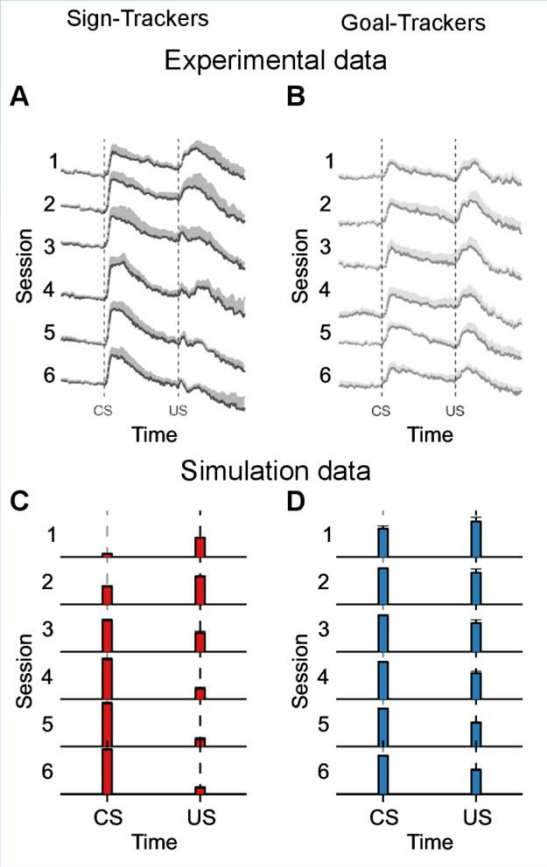
Behavioral results



Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

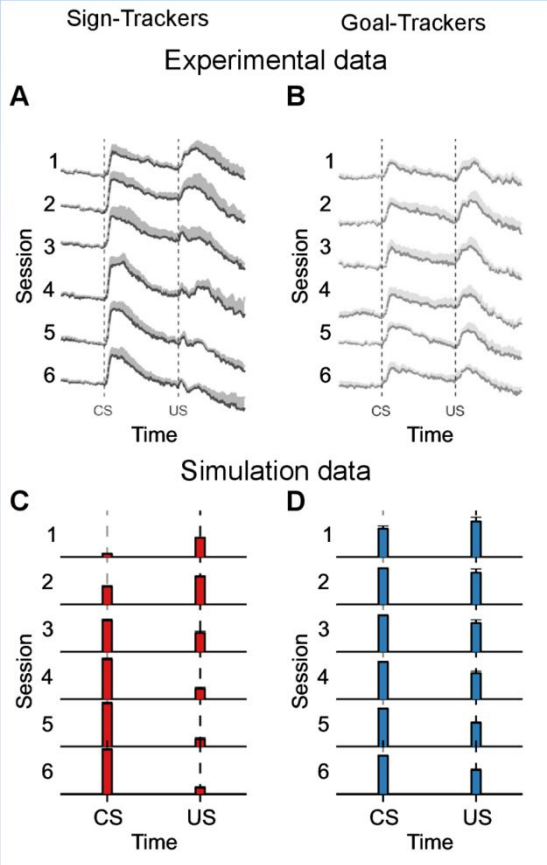
Physiological results

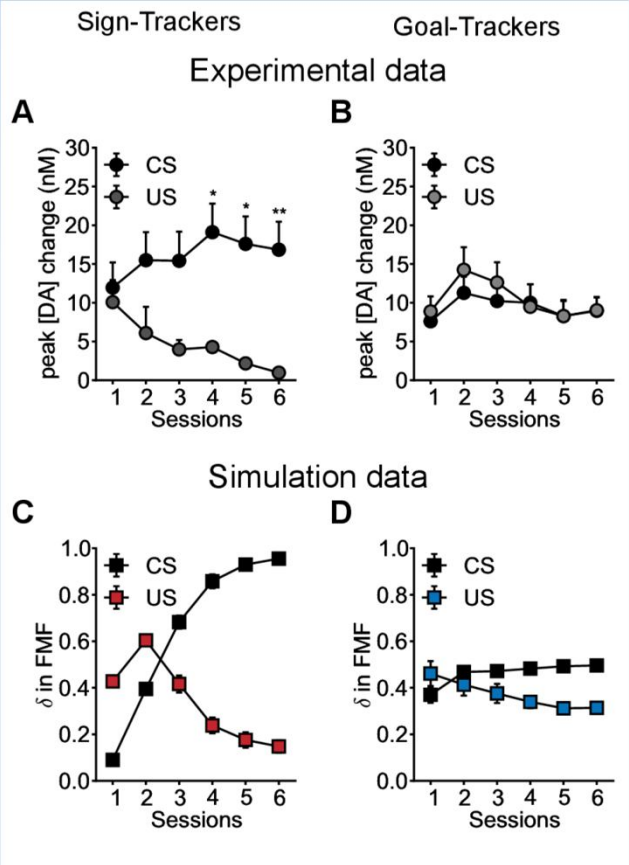


Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

Physiological results

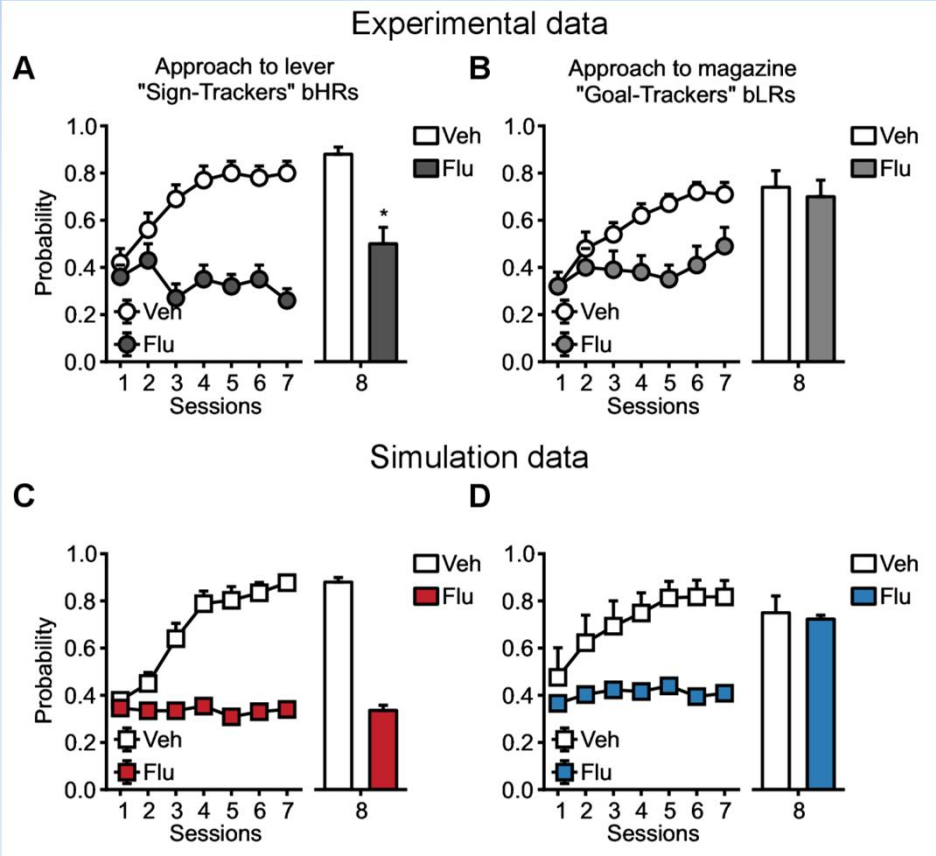




Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

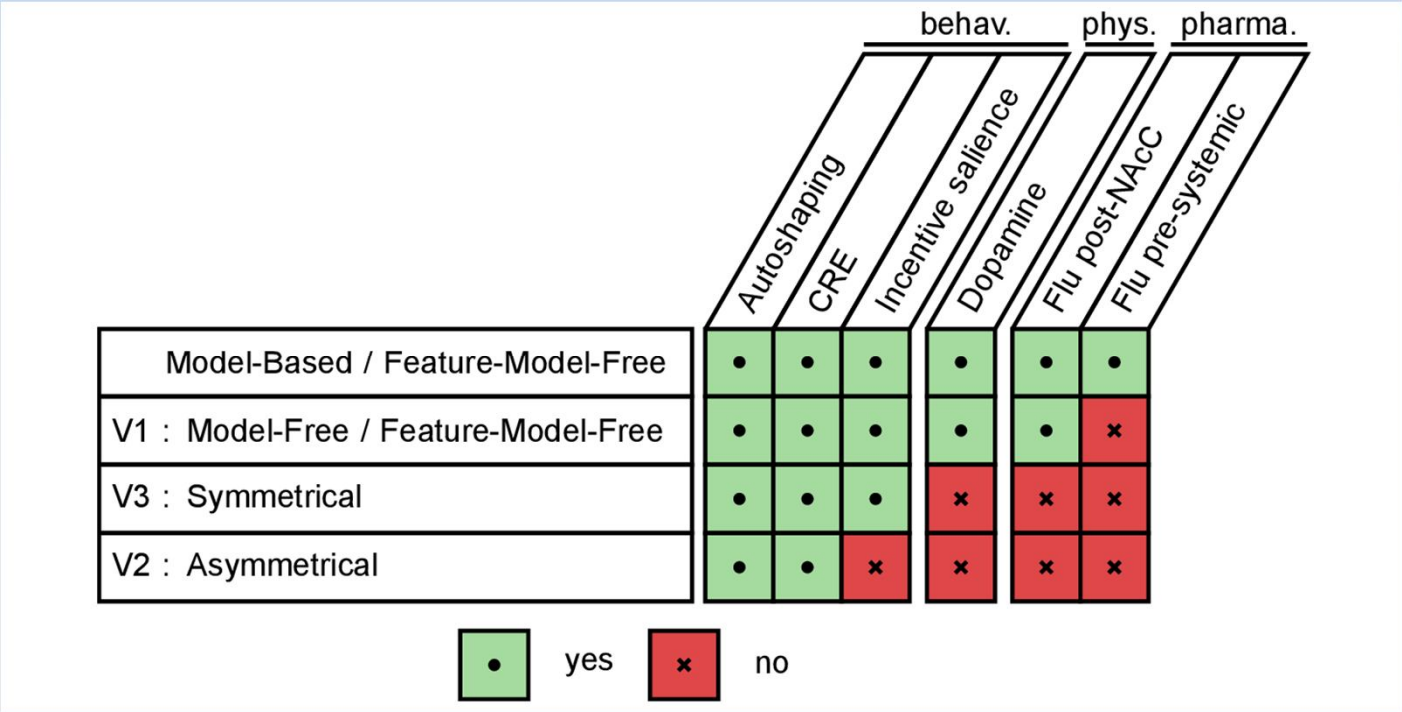
Pharmacological results



Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

Summary of the simulation results



Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

Computational model

Experimental predictions

- DA dip at each magazine visit during ITI.
- DA patterns in the intermediate group.
- Shortening the ITI should change DA pattern in GTs.
- Removing the magazine during ITI should abolish the difference in DA patterns between STs and GTs.
- Reducing the ITI duration should increase the tendency to goal-track in the overall population.

Lesaint, Sigaud, Flagel, Robinson, Khamassi (2014) PLOS Computational Biology.

META-LEARNING

(regulation of decision-making)

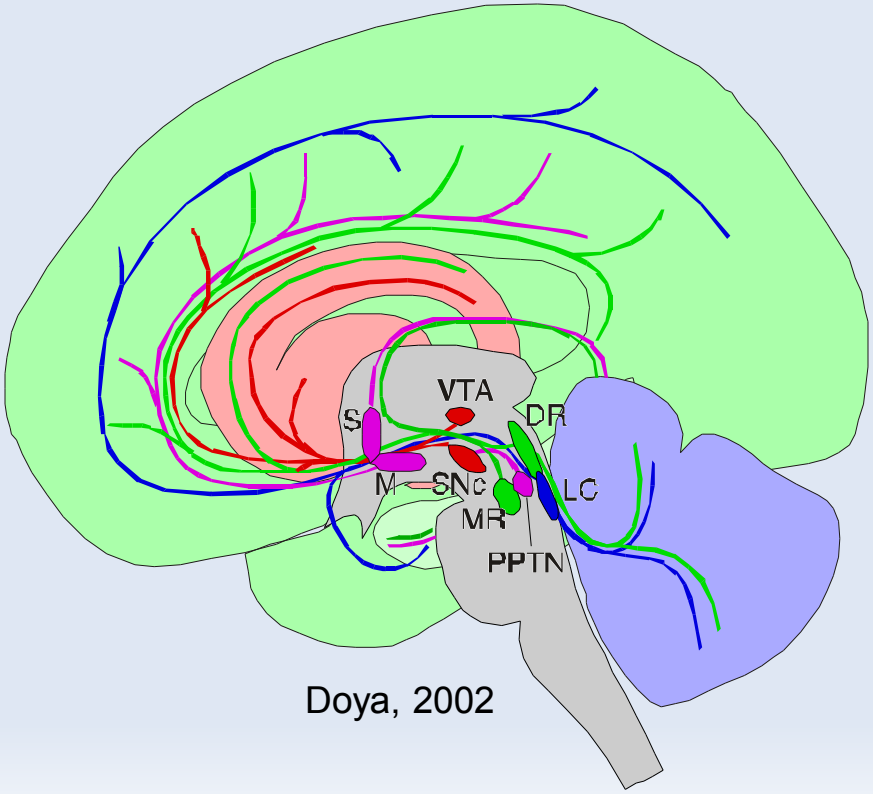
1. Dual-system RL coordination
2. Online parameters tuning

META-LEARNING

$$Q(s,a) \leftarrow Q(s,a) + \alpha \,.\, \delta$$ **Action values update**

$$\delta = r + \gamma \,.\, \max[Q(s',a')] - Q(s,a)$$ **Reinforcement signal**

$$P(a) = \frac{\exp(\beta \,.\, Q(s,a))}{\sum_b \exp(\beta \,.\, Q(s,b))}$$ **Action selection**



Doya, 2002

Dopamine: TD error δ

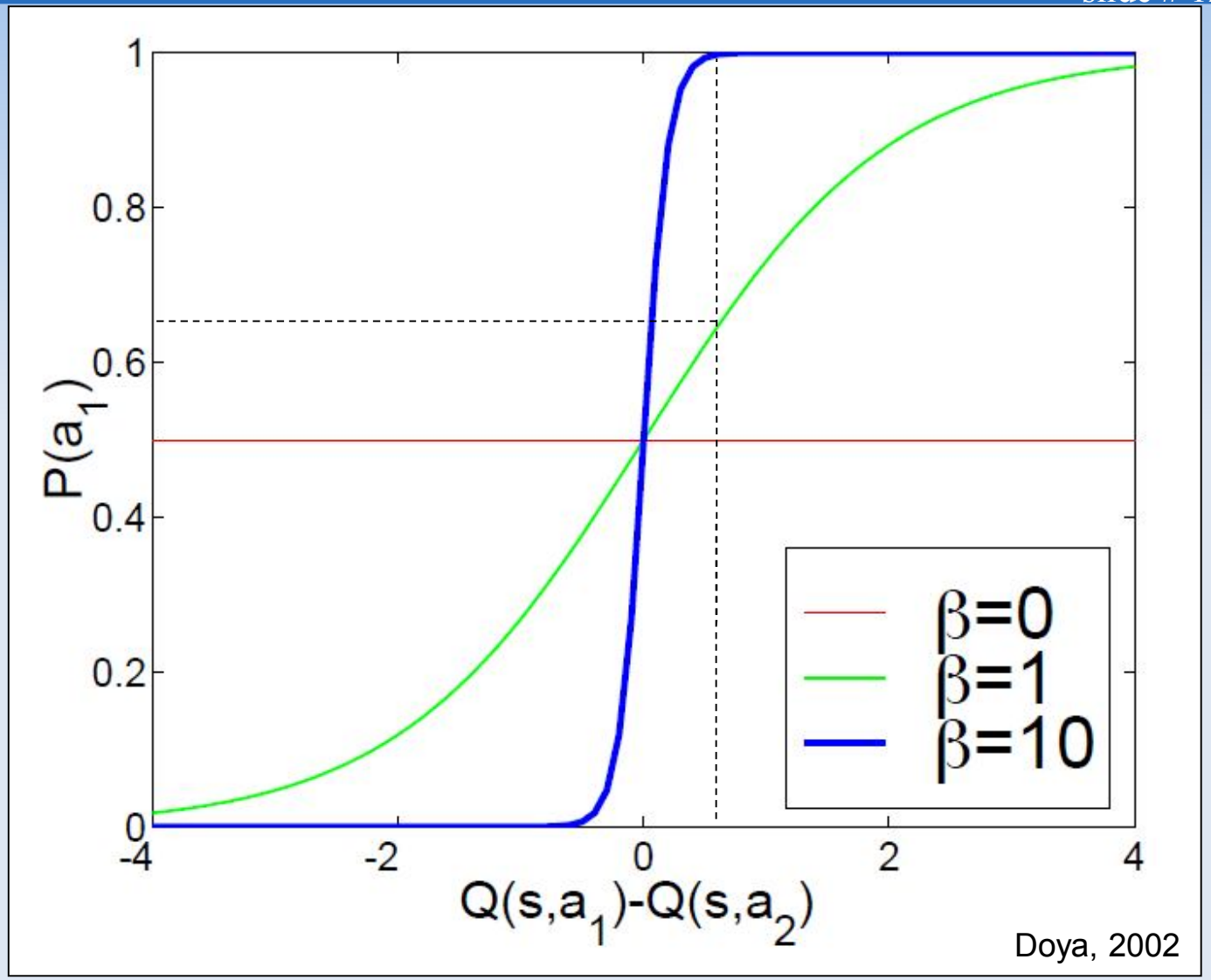
Acetylcholine: learning rate α

Noradrenaline: exploration β

Serotonin: temporal discount γ

META-LEARNING

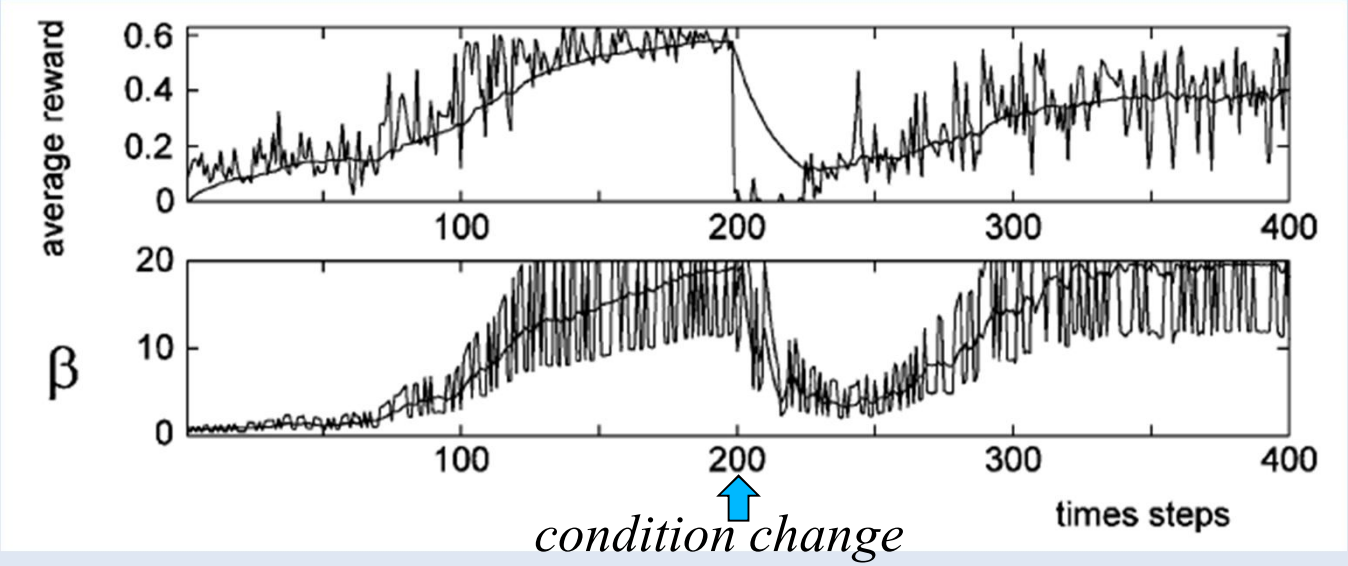
Effect of β on exploration



Boltzmann softmax equation: $P(a) = \frac{\exp(\boldsymbol{\beta} \cdot Q(s,a))}{\sum_b \exp(\boldsymbol{\beta} \cdot Q(s,b))}$

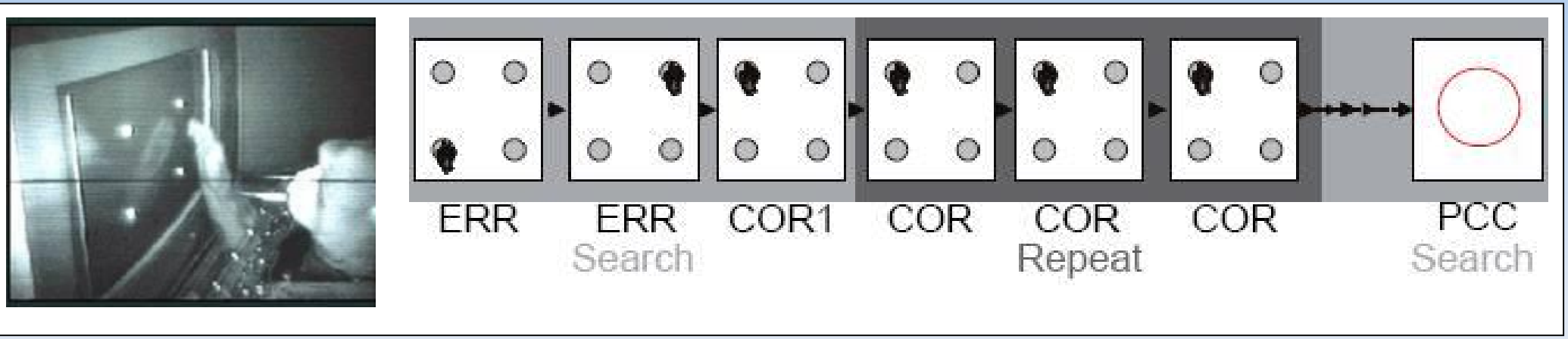
META-LEARNING

- Meta-learning methods propose to tune RL parameters as a function of average reward and uncertainty (Schweighofer & Doya, 2003).



→Can we use such meta-learning principles to better understand neural mechanisms in the prefrontal cortex ?

Back to my post-doc work

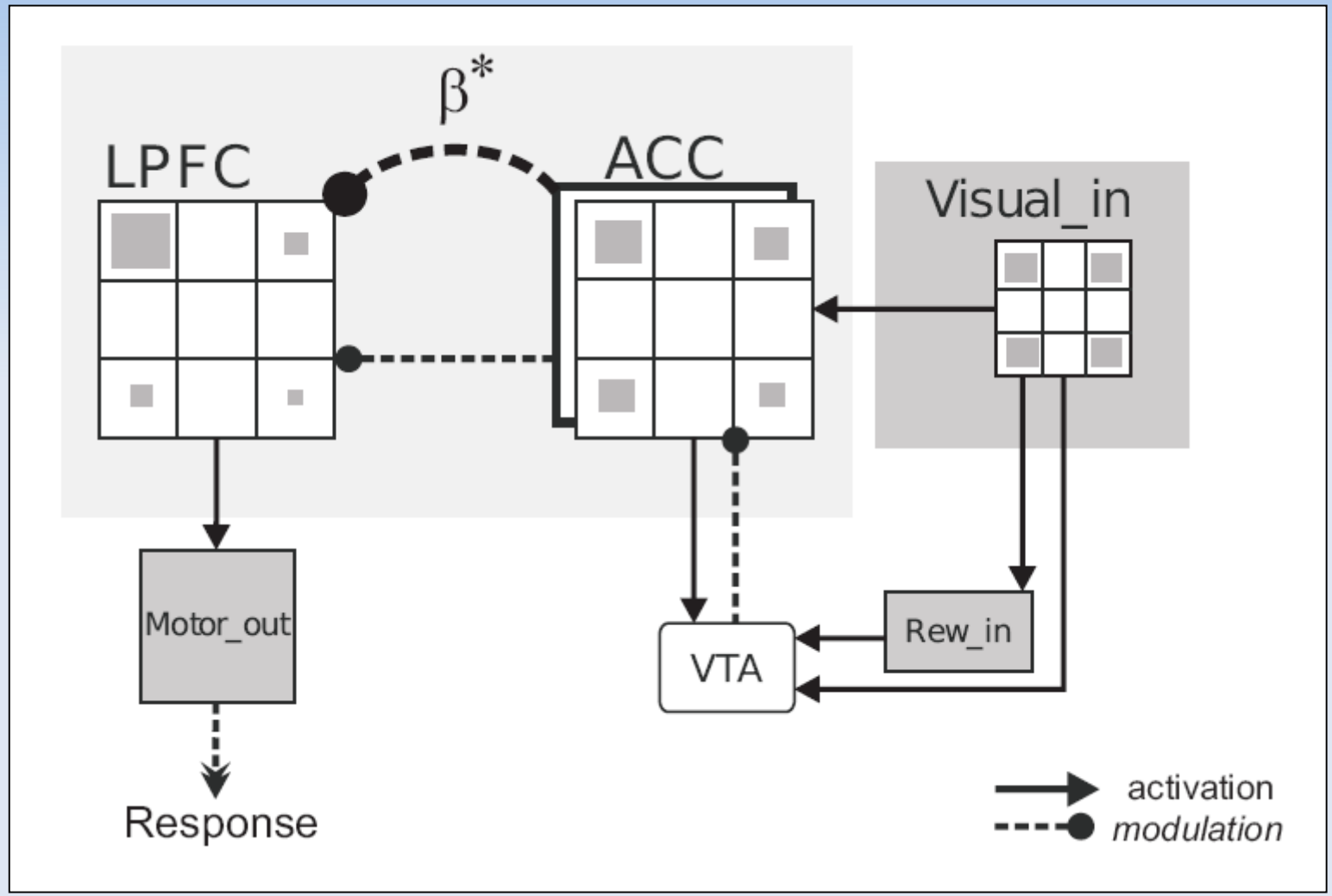


Question: How did the monkeys learn to re-explore after each presentation of the PCC signal?

Hypothesis: By trial-and-error during pretraining.

Khamassi et al. (2011) Front in Neurorobotics; Khamassi et al. (2013) Prog Brain Res

Computational model



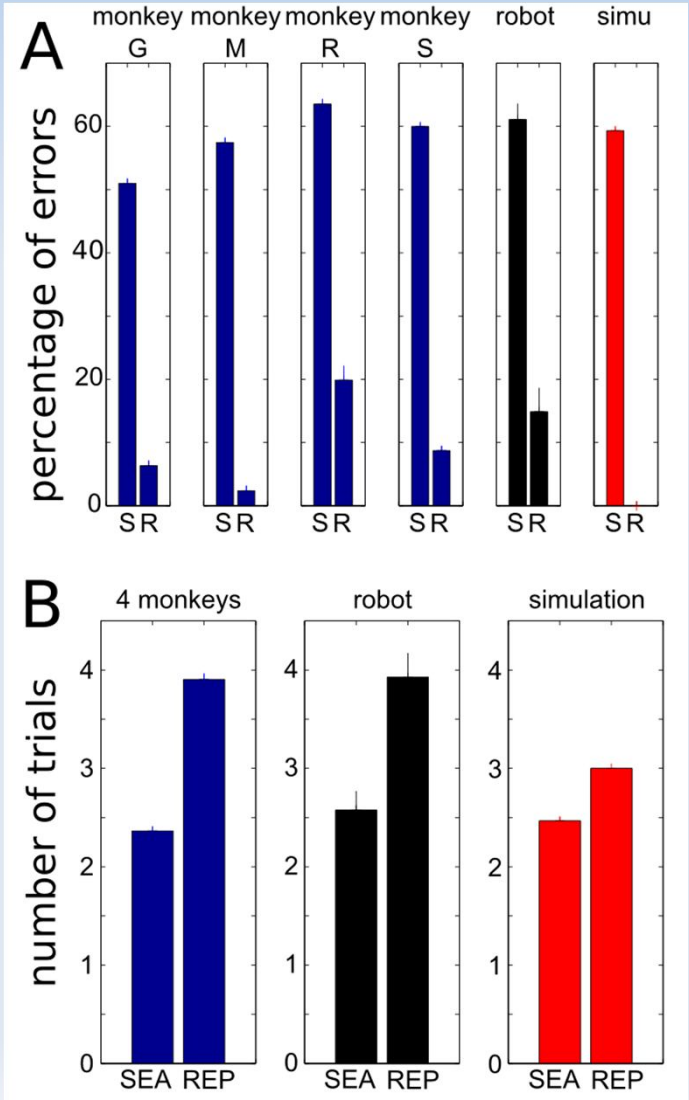
β^*: exploratory variable used to modulate β

Khamassi et al. (2011) Frontiers in Neurorobotics

Robotic model of monkey behavior in this task

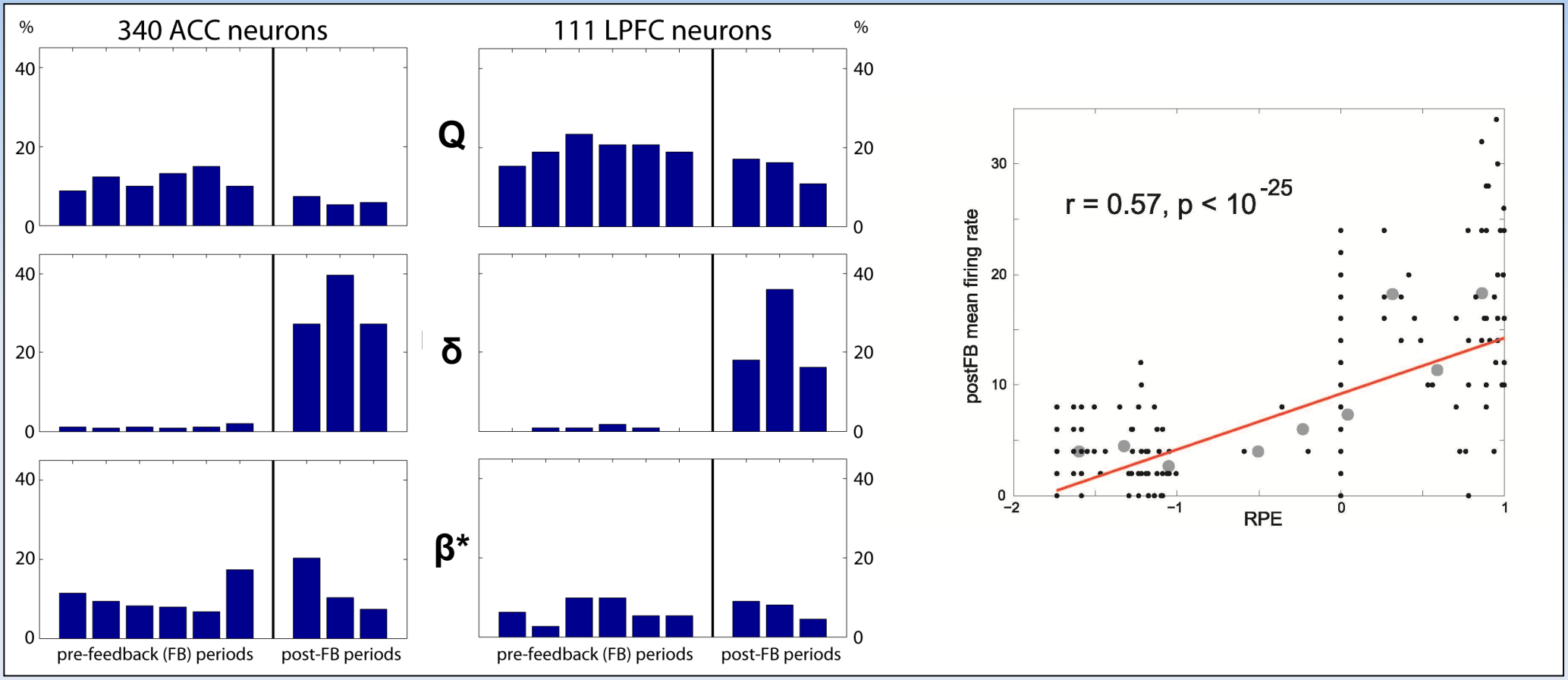
Computational model

- Reproduction of the global properties of monkey performance in the PS task.



Khamassi et al. (2011) Frontiers in Neurorobotics

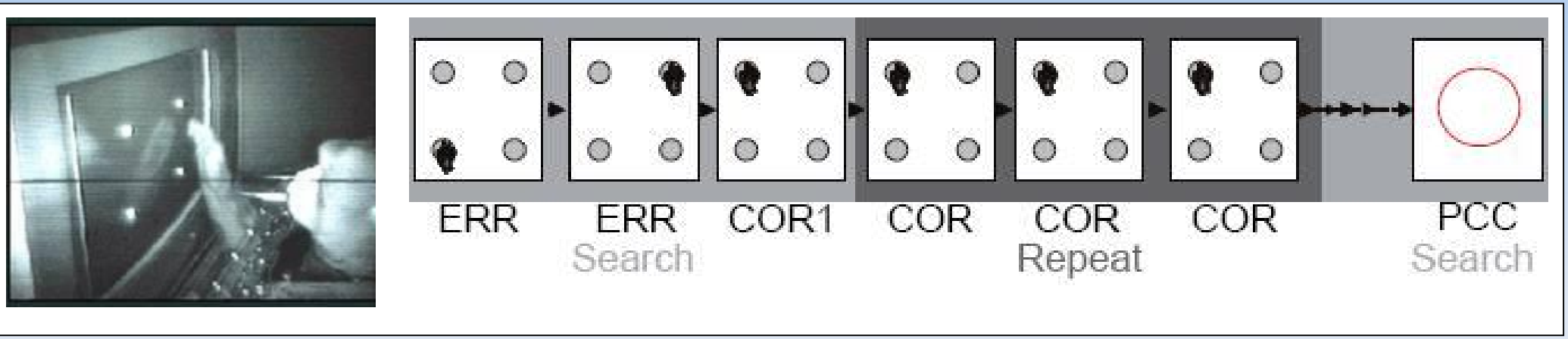
Model-based analysis
My post-doc work



Multiple regression analysis with bootstrap

Khamassi et al. (2013) Prog Brain Res; Khamassi et al. (2014) Cerebral Cortex

Meta-learning applied to Human-Robot Interaction

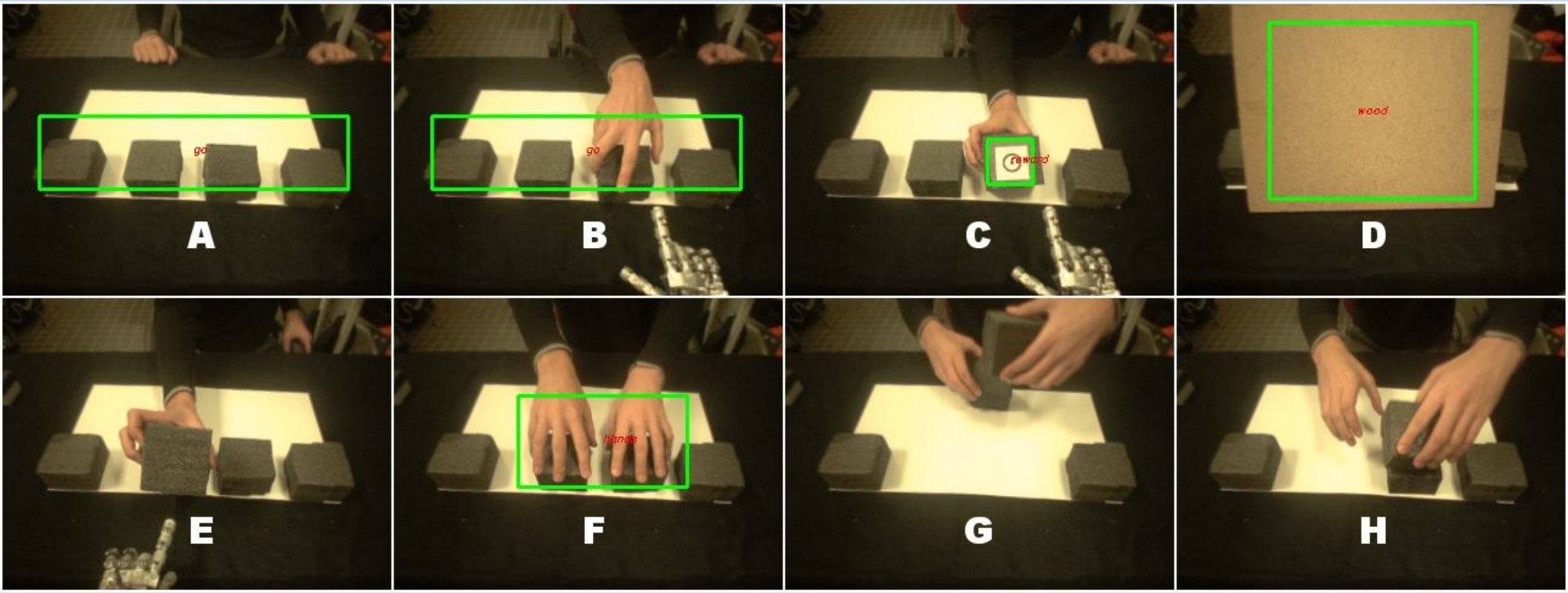


- In the previous task, monkeys and the model a priori 'know' that *PCC* means a reset of exploration rate and action values.
- Here, we want the iCub robot to learn it by itself.

Application to simple learning in humanoid robot

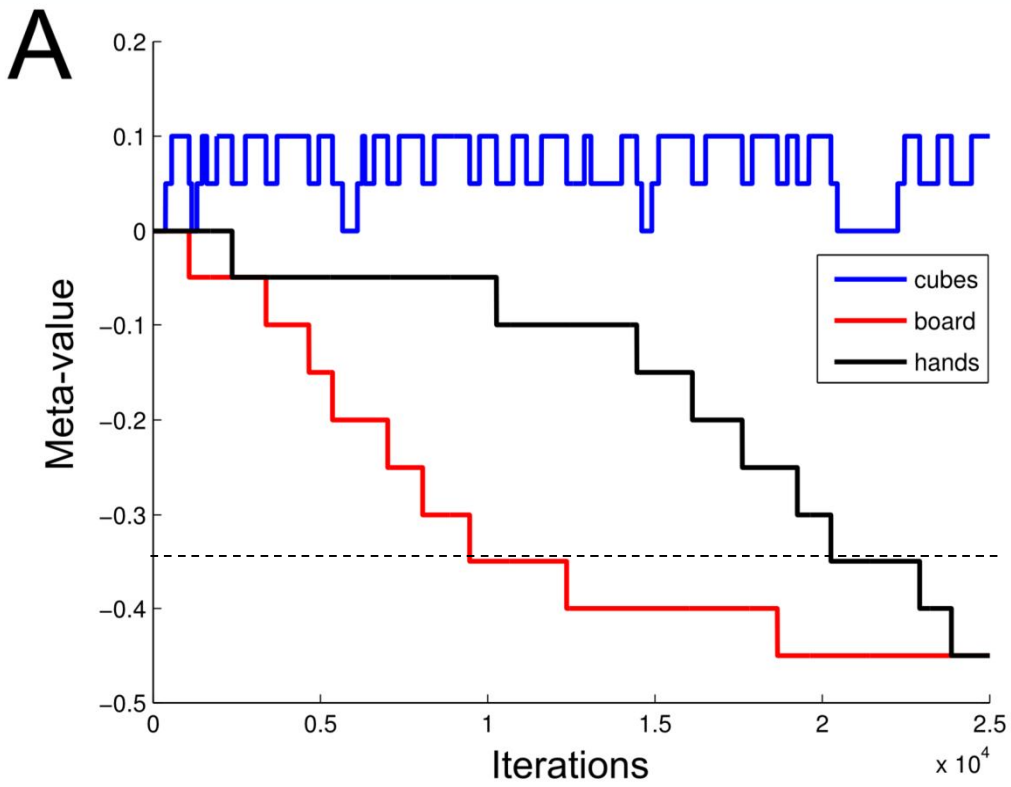
Khamassi et al. (2011) Frontiers in Neurorobotics

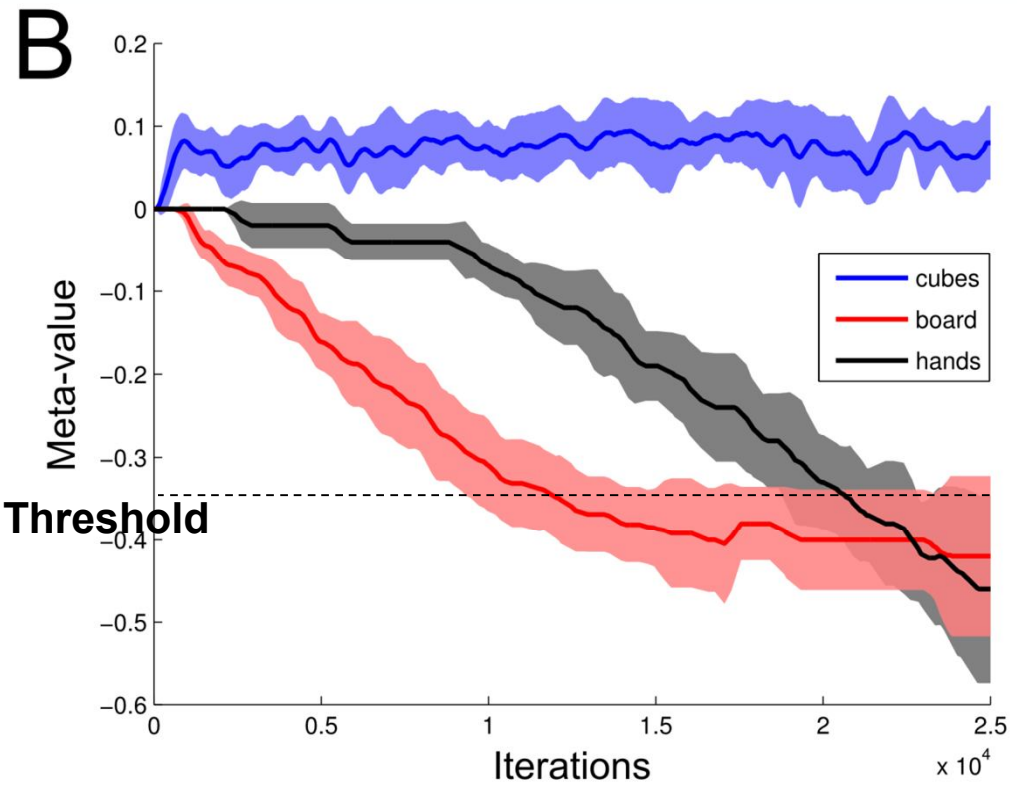
Meta-learning applied to Human-Robot Interaction



Meta-learning applied to Human-Robot Interaction

meta-value(i) ← meta-value(i) + α'. Δ[averageReward]





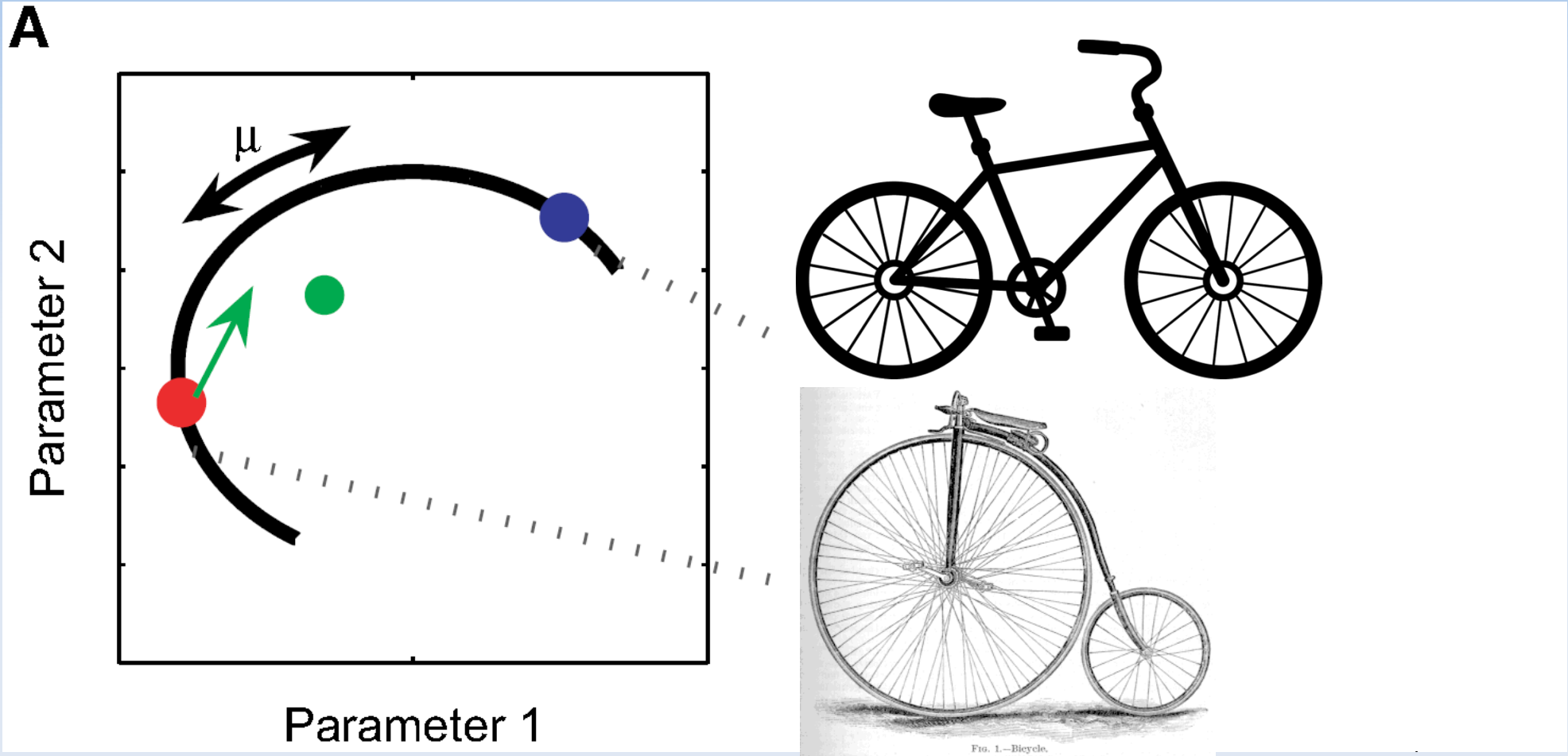
CONCLUSION OF THE ACC-LPFC META-LEARNING PART

- ACC is in an appropriate position to evaluate feedback history to modulate the exploration rate in LPFC.
- ACC-LPFC interactions could regulate exploration based on mechanisms capturable by the meta-learning framework.
- Such modulation could be subserved via noradrenaline innervation in LPFC.
- Such a pluridisciplinary approach can contribute both to a better understanding of the brain and to the design of algorithms for autonomous decision-making.

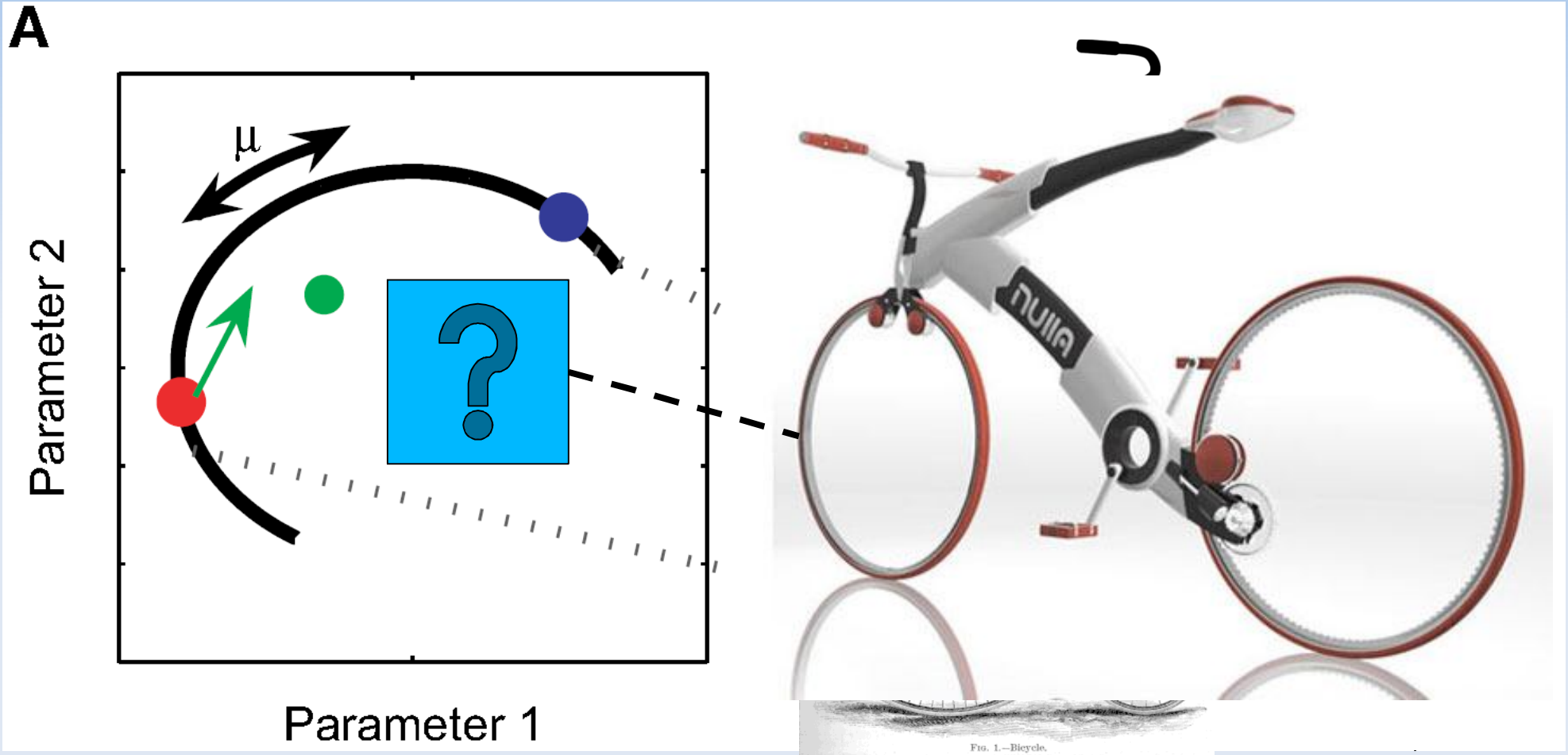
Meta-learning and motor learning

- Can meta-learning principles be useful for the integration of reinforcement learning and motor learning?

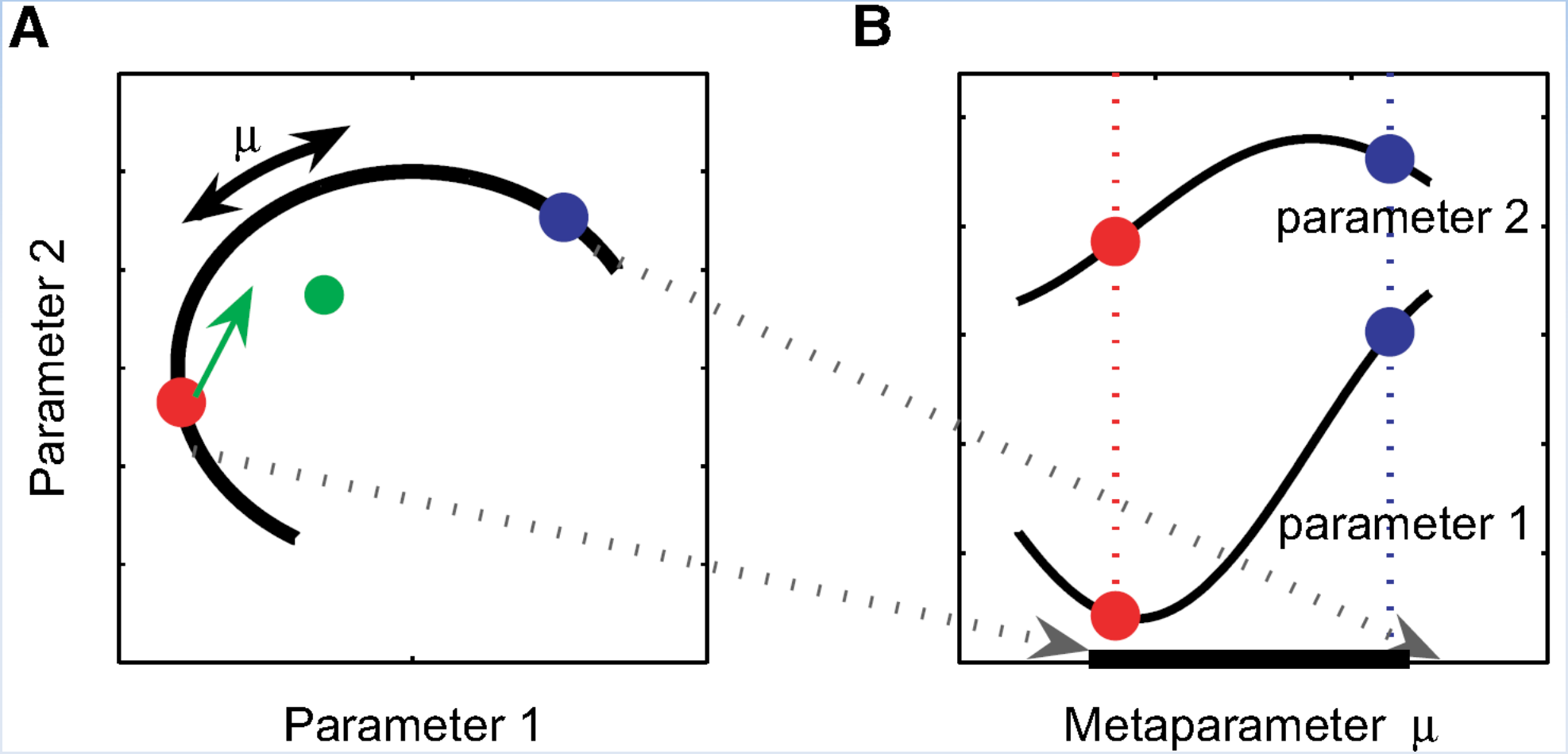
Structure learning
(Braun Aertsen Wolpert Mehring 2009)



Structure learning
(Braun Aertsen Wolpert Mehring 2009)

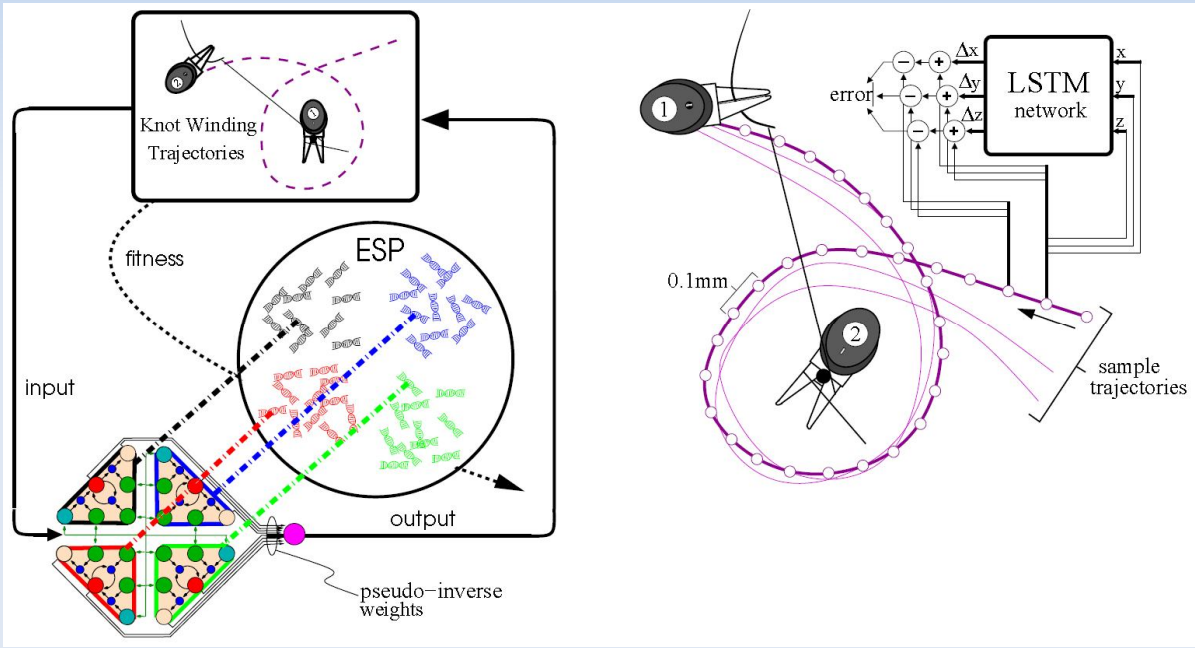


Structure learning
(Braun Aertsen Wolpert Mehring 2009)



Schmidhuber on meta-learning (1)

- Recurrent neural-networks applied to Robotics



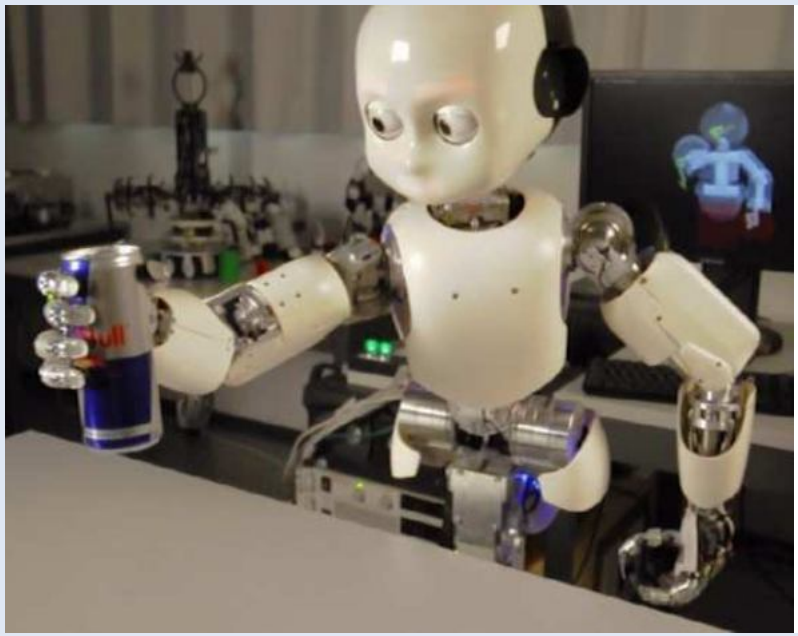
Mayer et al. (IROS 2006)

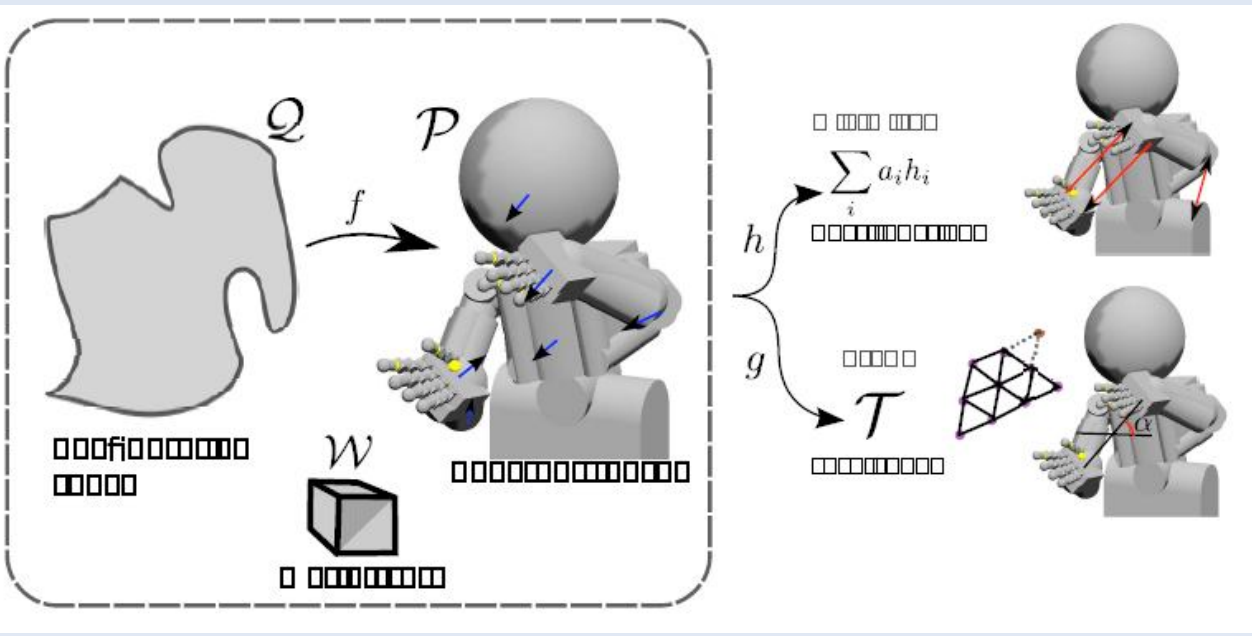
Schmidhuber on meta-learning (2)

- RL with self-modifying policies (actions that can edit the policy itself)
- Success-story criterion (time varying set V of past checkpoints that led to long-term reward accelerations)

Schmidhuber on motor learning

- Learning maps of task-relevant motor behaviors under specified constraints (e.g. maintain hands parallel ; do not touch box nor table ; …)
- How can these primitive constrained motor behaviors be used by decision system and high-level goal-directed learning?





Stollenga et al. (IROS 2013)

SUMMARY

- Dopamine neurons encode a reward prediction error.
- Model-based analysis in Neurosci of Decision-making
- Reinforcement Learning models need to be refined to explain behavior / neural activity:
  - multiple parallel decision systems.
  - off-line learning during sleep.
  - meta-learning (ACC-DLPFC interactions).
- These model improvements can produce testable experimental predictions (Pavlovian autoshaping ; Navigation ; L-DOPA in Parkinson disease ; …)

CONCLUSION

- The Reinforcement Learning framework provides algorithms for autonomous agents.
- It can also help explain neural activity in the brain.
- Such a pluridisciplinary approach can contribute both to a better understanding of the brain and to the design of algorithms for autonomous decision-making.

ACKNOWLEDGMENTS

ISIR (CNRS – UPMC)

Nassim Aklil

Jean Bellot

Ken Caluwaerts

Raja Chatila

Laurent Dollé

Benoît Girard

Florian Lesaint

Olivier Sigaud

Guillaume Viejo

LAAS (CNRS Toulouse)

Rachid Alami

Aurélie Clodic

Univ. Manchester

Mark D. Humphries

Financial support

FP6 IST 027189
European project

Learning under
Uncertainty Project ;
ROBOERGOSUM Project

HABOT Project
Emergence(s) Program