Artificial Neural Networks and RL

Wulfram Gerstner

EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

Objectives for today:

- theory of exploration bonus
- understand surprise
- understand difference of novelty and surprise
- use of surprise to modulate learning rate
- use of novelty to guide exploration

Previous slide.

Background reading:

An analysis of model-based Interval Estimation for Markov Decision Processes
Strehl and Littman, 2008
https://www.sciencedirect.com/science/article/pii/S0022000008000767

Novelty is not Surprise: Human exploratory and adaptive behavior in sequential decision-making
HA Xu*, A Modirshanechi*, MP Lehmann, W Gerstner, MH Herzog, PLOS Comput. Biol. E1009070, (2021)

Learning in Volatile Environments with the Bayes Factor Surprise
V Liakoni*, A Modirshanechi*, W Gerstner, J Brea
Neural Computation 33 (2), 269-340 (2021)

A taxonomy of surprise definitions
A Modirshanechi, J Brea, W Gerstner
Journal of Mathematical Psychology 110, 102712 (2022)

Novelty and Surprise

Q1: What is an exploration bonus?

Q2: What is novelty?

Q3: What is surprise?

Q4: What is the difference between the two?

Q5: Why are they useful?

Q6: Why should we talk about novelty in an RL class?

Previous slide.

Today we will ask 6 questions:

What is an exploration bonus,
What is novelty, What is surprise, What is the difference, Why are they useful.

And why should we talk about it in a class on RL?

Artificial Neural Networks and RL

Wulfram Gerstner
EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Formal Exploration Bonus

(Thanks to Dr. Alireza Modirshanechi)

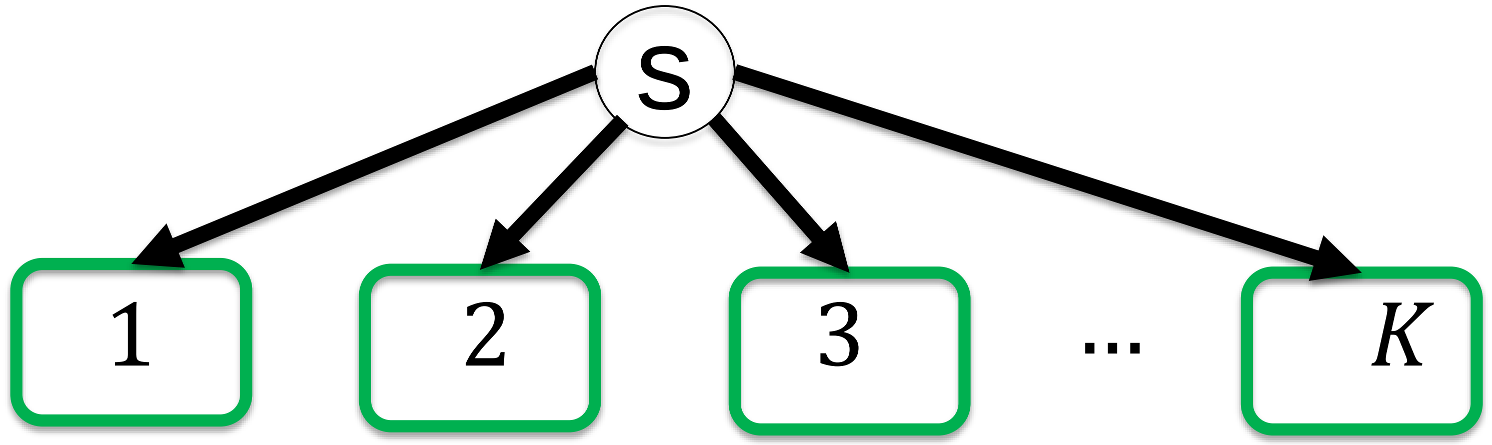
Previous slide.

We start with some results from the formal theory of exploration.

a) For multi-armed bandits (1-step horizon)
b) For full Markov Decision Problem (multi-step horizon)

Review: Multi-armed Bandits: MAB (1-step horizon)

- Single state. We have K possible actions:



Which action to choose at time t?

- With true average reward:

$\mu_i = E[r|a = i]$ $\quad \mu_1 \quad \mu_2 \quad \mu_3 \quad \ldots \quad \mu_K$

Optimal policy: $\boldsymbol{a_t = arg\,\max_i \mu_i}$

- Naïve estimates of averages:

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

$\hat{\mu}_1^{(t)} \quad \hat{\mu}_2^{(t)} \quad \hat{\mu}_3^{(t)} \quad \ldots \quad \hat{\mu}_K^{(t)}$

$T_i^{(t)} = \{\tau \leq t : a_\tau = i\}$

Not optimal: $\boldsymbol{a_t = arg\,\max_i \hat{\mu}_i^{(t)}}$

Solutions based on random exploration:
- Epsilon-greedy
- Softmax

- Comments for the previous slide:

- If we knew the exact average reward $\mu_i = E[r|a = i]$ of each arm, then the optimal solution would trivially be to choose the arm with highest average reward: $a_t = \arg\max_i \mu_i$

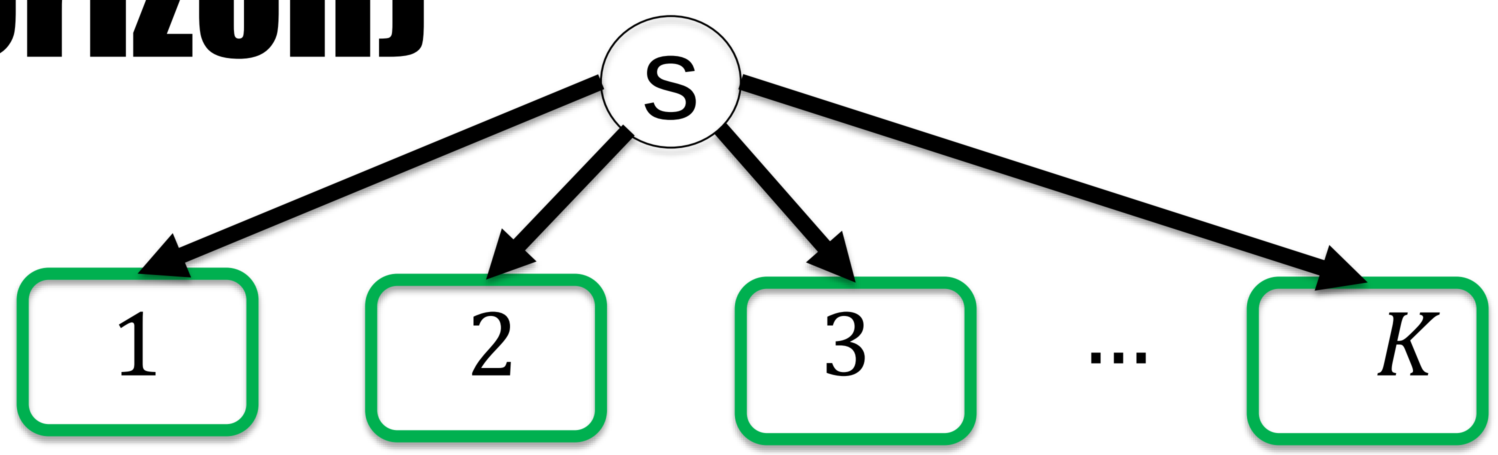
- A naïve approach is to estimate the average reward by the empirical averages and greedily choose the action with maximum estimated average reward: $a_t = \arg\max_i \hat{\mu}_i^{(t)}$

- The naïve greedy policy is prone to fail in finding the best action.

- You have seen epsilon-greedy and the softmax policy as two approaches for dealing with this problem by adding randomness to the action-selection.

- Our focus will be on “directed exploration” by using exploration bonuses.

Regret in Multi-armed Bandits (1-step horizon)



- MAB with K possible actions:

$$\mu_i = E[r|a=i]$$

Highest reward rate: $\mu^* = \max_i \mu_i$

- "Regret" of algorithm $\boldsymbol{A}$ (e.g., ϵ-greedy):

$$R_A(T) = E_A\left[\sum_{t=1}^{T} \mu^* - \mu_{a_t}\right]$$

with best action you can choose

with your actual choices

- Consistent algorithms:

$$\lim_{T\to\infty} \frac{R_A(T)}{T} = 0 \quad \Longrightarrow \quad \lim_{T\to\infty} \frac{E_A[\sum_{t=1}^{T} \mu_{a_t}]}{T} = \mu^*$$

- Theorem 1 of Lai and Robbins 1985:

  Under specific conditions, if algorithm A is consistent, then, loosely speaking, $R_A(T)$ is at least proportional to $\log T$.

a loose notion of optimality

Idea: you need to play other actions, even if that means that $R_A(T)$ increases

- Comments for the previous slide:

- Before discussing how differently one can deal with exploration-exploitation dilemma, we discuss a common method for evaluating different algorithms in multi-armed bandits.

- A key notion to evaluate an algorithm A is regret $R_A(T)$ measuring the expected difference between the choices of the algorithm and the best possible actions, summed over the first T steps.

- An algorithm is called consistent, if its average regret $\frac{R_A(T)}{T}$ vanishes over time.

- It is proven (under certain conditions; see Lai and Robbins 1985 in Advances in Applied Mathematics) that the regret of a consistent algorithm scales at least logarithmically with time T.

- This introduces a loose notion of optimality: An optimal algorithm is a consistent algorithm whose regret scales logarithmically with time T.

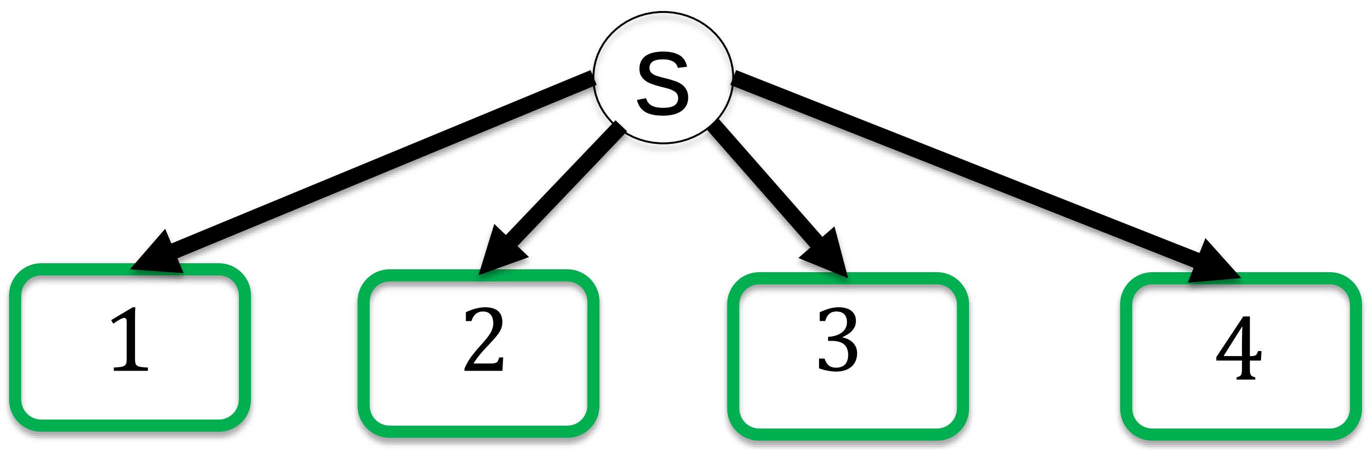
Example: average rewards in MAB (1-step horizon)

- MAB with **4** possible actions (Example):

$\mu_i = E[r|a=i]$

rewards are stochastic (binomial)

$P(r_t = 2\mu_i \,|a=i) \;\; = 0.5 \; = P(r_t = 0 \,|a=i)$



$\mu_1 = 1$

$\mu_2 = 0.9$

$\mu_3 = 9.9$

$\mu_4 = 10.0$

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

What is the probability that $\hat{\mu}_4^{(160)}$=2?

[ ] $\binom{N}{k} 2^{(-40)}$

[ ] between $10^{(-7)}$ and $10^{(-8)}$

1 $\hat{\mu}_i^{(t)}$ after 40 trials 10

for each action

- Comments for the previous slide:

- How likely is it in the above example that the ‘best’ action with mean reward 10 would have after 40 trials a value of 2?

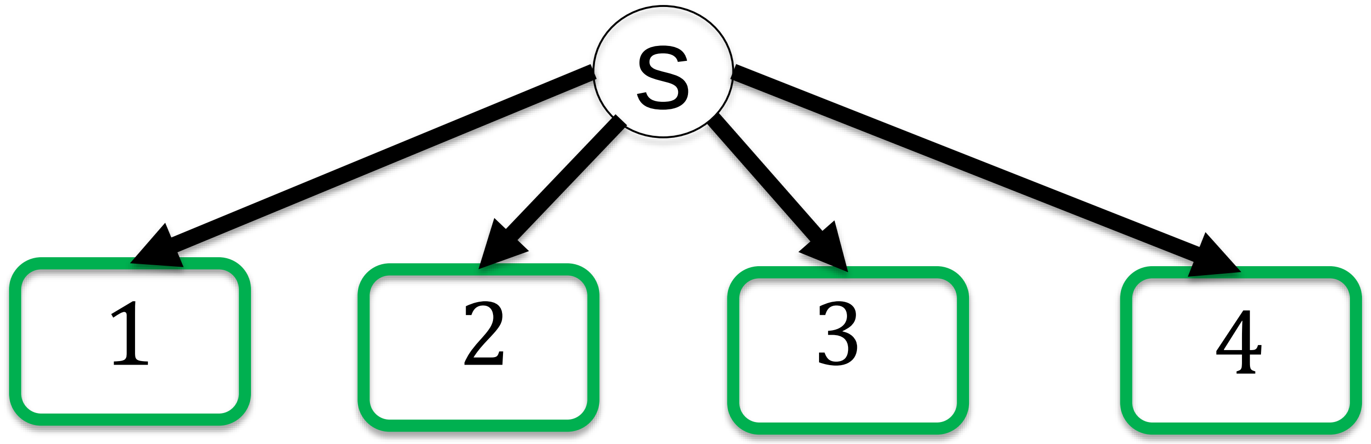
Example: average rewards in MAB (1-step horizon)

- MAB with **4** possible actions (Example):

$\mu_i = E[r|a = i]$

rewards are stochastic (binomial)

$P(r_t = 2\mu_i \mid a = i) = 0.5 = P(r_t = 0 \mid a = i)$



$$R_A(T) = E_A\left[\sum_{t=1}^{T} \mu^* - \mu_{a_t}\right]$$

$\mu_1 = 1$

$\mu_2 = 0.9$

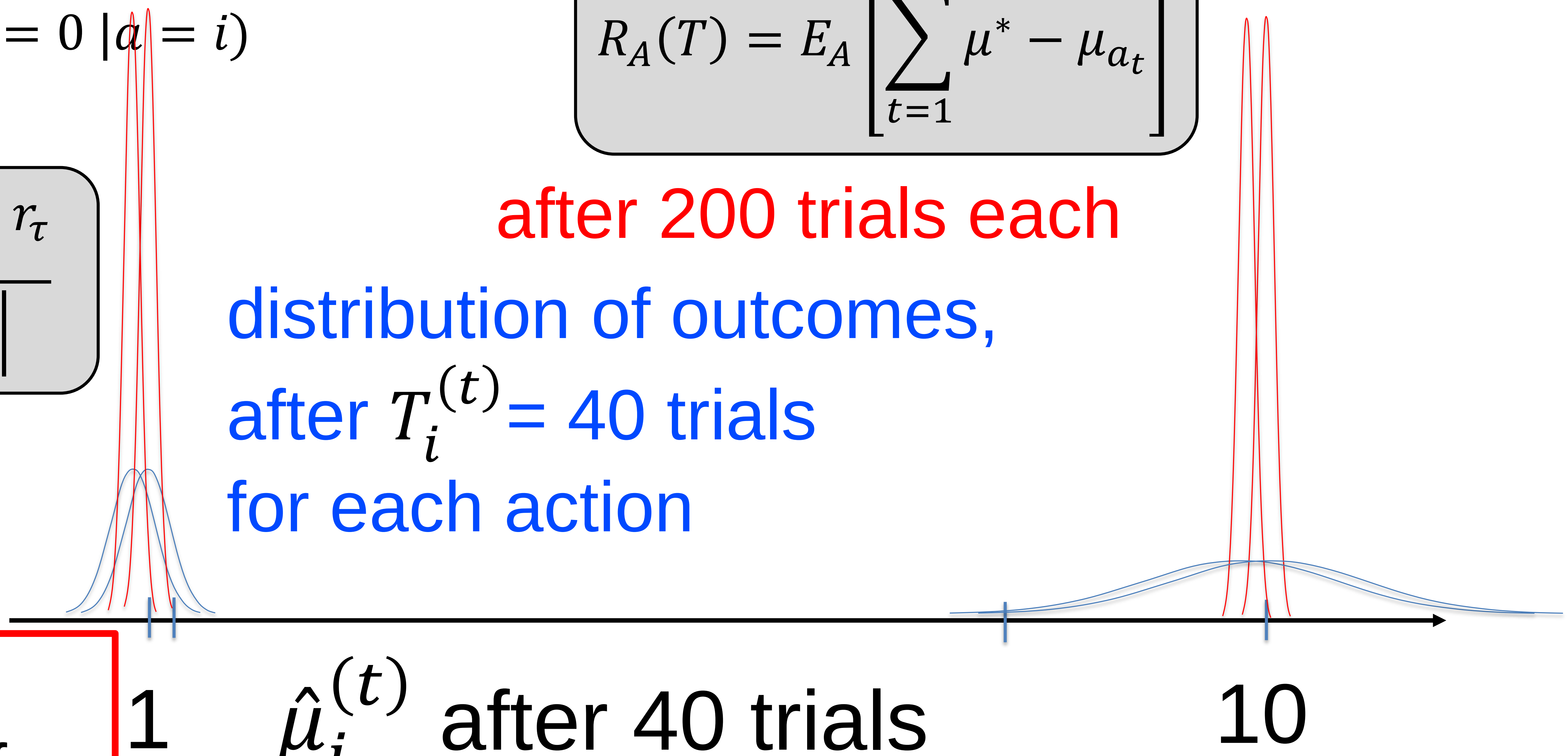
$\mu_3 = 9.9$

$\mu_4 = 10.0$

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

after 200 trials each

distribution of outcomes, after $T_i^{(t)}$ = 40 trials for each action



$\hat{\mu}_i^{(t)}$ after 40 trials for each action

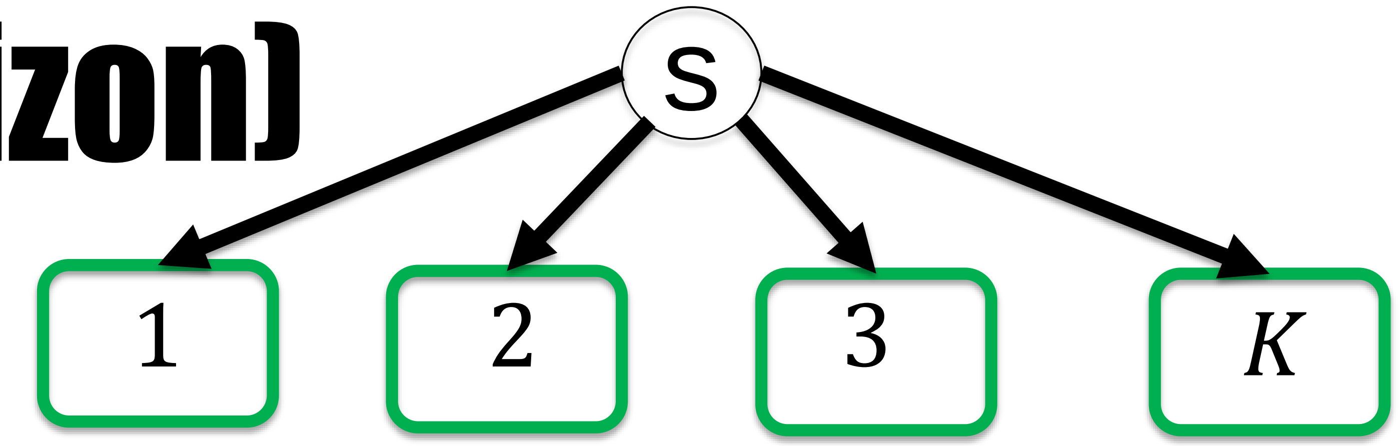
Idea: play other actions if tails of distribution overlap

- Comments for the previous slide:

- Example of MAB with 4 actions. Each action yields a reward with 50 percent probability.
- Two actions have low rewards (about 1); the two other have high rewards about 20.

- Imagine that at the beginning you played each action 40 times and evaluate the mean return.
- If you repeated the game many times, each time starting with playing each action 40 times, you would get a distribution (hand-drawn here).
- As long as the distributions overlap,

Exploration Bonus for MAB (1-step horizon)



- MAB with K possible actions:

- Reminder: greedy algorithm

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

$\hat{\mu}_1^{(t)}$ $\hat{\mu}_2^{(t)}$ $\hat{\mu}_3^{(t)}$ … $\hat{\mu}_K^{(t)}$

$$a_t = \arg\max_i \hat{\mu}_i^{(t)}$$

- Upper Confidence Bound (UCB1 in Auer et al. 2002):

$$U_i^{(t)} = \hat{\mu}_i^{(t)} + \sqrt{\frac{2 \log t}{\left|T_i^{(t)}\right|}}$$

The naïve estimate of average reward

Bonus for exploration (compare: Monte Carlo Tree Search)

$U_1^{(t)}$ $U_2^{(t)}$ $U_3^{(t)}$ … $U_K^{(t)}$

$$a_t = \arg\max_i U_i^{(t)}$$

Theorem 1 of Auer et al. 2002:
$R_{\text{UCB1}}(T) \propto \log T + \text{const.}$

Play greedy, but with a modified 'value' U_k

→ Add exploration bonus to empirical average of reward

- Comments for the previous slide:

- A smart optimal algorithm is Upper Confidence Bound (UCB; proposed by Auer et al. 2002 in Machine Learning) that computes a confidence bound index $U_i^{(t)}$ for each action and chooses the one with highest index.

- The index is equal to the naïve estimate average reward $\hat{\mu}_i^{(t)}$ plus an exploration bonus that is (i) a decreasing function of how many times an arm has been chosen $\left|T_i^{(t)}\right|$ but (ii) an increasing function of how many actions have been taken in total (i.e. t).

- The regret for the UCB algorithm scales logarithmically with T, hence it is an “optimal” algorithm. The constants of the regret can be fine-tuned by some variations of the algorithm (see Auer et al. 2002).

Quiz: exploration Bonus (1-step horizon)

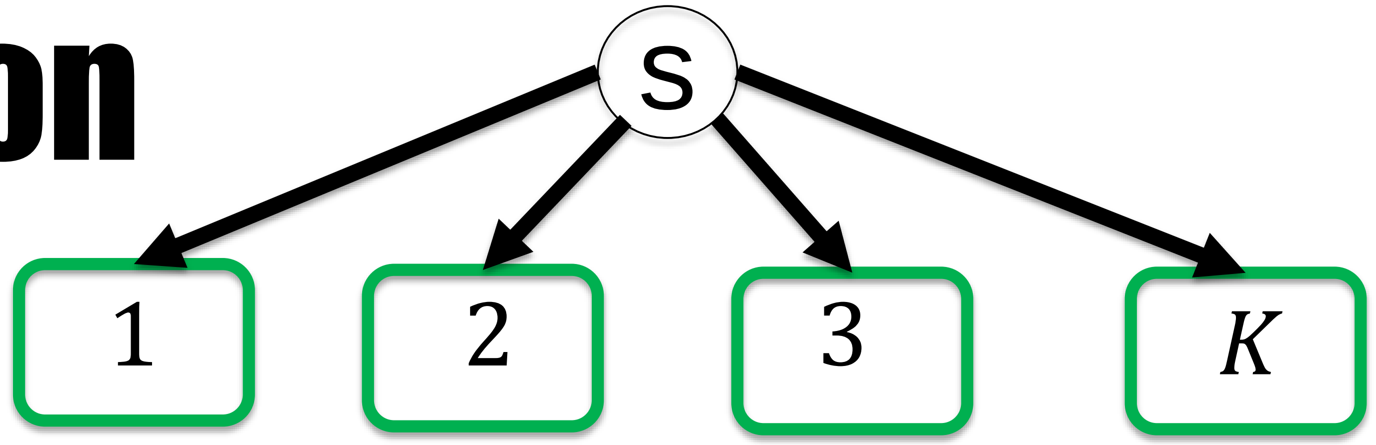
- ? • A consistent learning algorithm eventually achieves a **zero *average* regret** in Multi-Armed Bandits (MAB).
- ? • An optimal algorithm in MABs achieves a **constant *total* regret**.
- ? • A good exploration bonus is $\frac{\beta}{T_i^{(t)}}$.
- ? • A good exploration bonus is $\frac{\log(t)}{\sqrt{T_i^{(t)}}}$.

Teaching monitoring – monitoring of understanding

[ ] up to here, at least 60% of material was new to me.

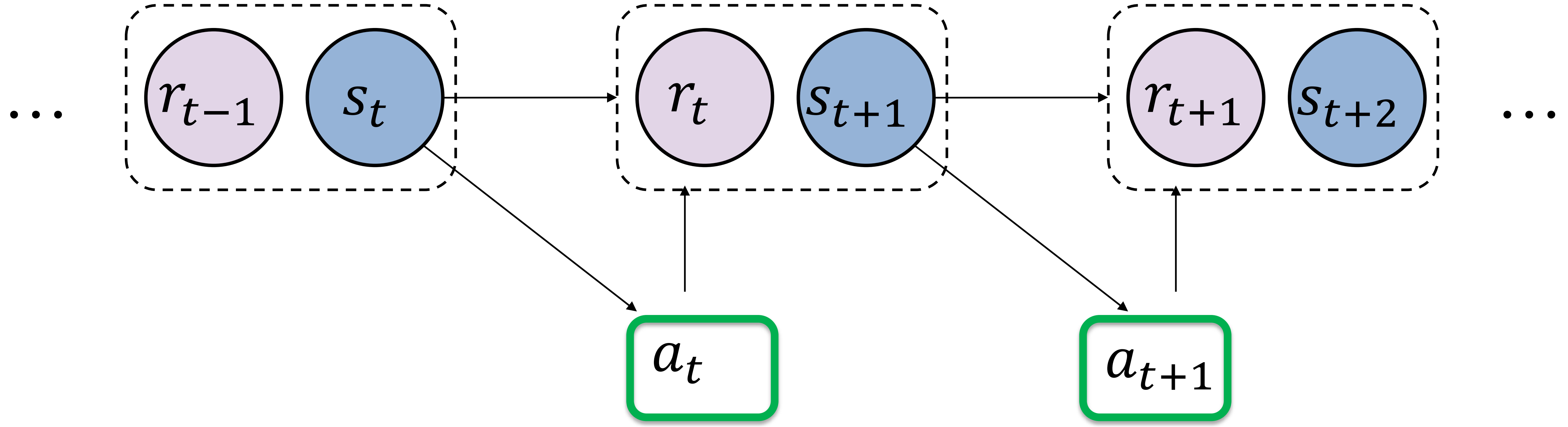
[ ] I have the feeling that I have been able to follow (at least) 80% of the lecture up to here.

Exploration Bonus for multi-step horizon



- MAB with K possible actions:

- Markov Decision Processes (MDP):
  - P: transition probabilities, e.g. $P(s'|s, a)$
  - R: expected reward, e.g. $R(s, a)$



- Comments for the previous slide:

- We now want to extend from 1-step horizon (MAB) to multi-step horizon. The Multistep horizon leads to the Markov Decision Problem (MDP).

Exploration Bonus for multi-step horizon

Bellman equation (optimal action choice)

- Dynamic programming with true $P(s'|s,a)$ and $R(s,a)$:

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

$$a_t = \arg\max_a Q^*(s_t,a)$$

- Naïve model-based (MB) RL:

$$\hat{Q}^{(t)}_{\text{MB}}(s,a) = \hat{R}^{(t)}(s,a) + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}^{(t)}_{\text{MB}}(s',a')$$

$$\hat{R}^{(t)}(s,a) = \frac{\sum_{\tau \in T^{(t)}_{s,a}} r_\tau}{\left|T^{(t)}_{s,a}\right|}$$

$$\hat{P}^{(t)}(s'|s,a) = \frac{\left|T^{(t)}_{s,a,s'}\right|}{\left|T^{(t)}_{s,a}\right|}$$

$$T^{(t)}_{s,a} = \{\tau \le t : a_\tau = a, s_\tau = s\}$$

$$T^{(t)}_{s,a,s'} = \{\tau \le t : a_\tau = a, s_\tau = s, s_{\tau+1} = s'\}$$

$$a_t = \arg\max_a \hat{Q}^{(t)}_{\text{MB}}(s_t,a)$$

The exploration-exploitation trade-off is even more serious in MDPs than MABs.

Any trick similar to UCB?

- Comments for the previous slide:

- Similar to the bandit setting, if we have access to the true transition probabilities and reward functions, then the optimal policy would be to use Dynamic Programming, solve the optimal Bellman equations, and use a greedy policy on the resulting Q-values: $a_t = \arg\max_a Q^*(s_t, a)$

- In the absence of the complete knowledge of the environment, a naïve model-based approach is to approximate the transition probabilities and the reward values, solve the optimal Bellman equations by using these estimates, and use a greedy policy on the resulting Q-values: $a_t = \arg\max_a \hat{Q}_{\mathrm{MB}}^{(t)}(s_t, a)$

- The naïve model-based approach is prone to be stuck in some parts of the environment and never find the optimal policy. You have seen epsilon-greedy and the softmax policy as to approaches to deal with this issue by adding randomness to the action-selection. Here, we ask whether we can find a directed exploration approach like UCB for MDPs. What is a good exploration bonus?

Exploration Bonus for multi-step horizon

- Dynamic programming with true $P(s'|s,a)$ and $R(s,a)$:

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

- Naïve model-based (MB) RL:

$$\hat{Q}_{\mathrm{MB}}^{(t)}(s,a) = \hat{R}^{(t)}(s,a) + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}_{\mathrm{MB}}^{(t)}(s',a')$$

- Model-based interval estimation with exploration bonus (MBIE+EB in Strehl and Littman 2008):

$$\hat{Q}_{\mathrm{MB}}^{(t)}(s,a) = \hat{R}^{(t)}(s,a) + \frac{\beta}{\sqrt{T_{s,a}^{(t)}}} + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}_{\mathrm{MB}}^{(t)}(s',a')$$

The naïve estimate of average reward

Bonus for exploration (different from UCB regarding $\log t$)

- Comments for the previous slide:

- Model-based interval estimation with exploration bonus (MBIE+EB; proposed by Strehl and Littman 2008 in the Journal of Computer and System Sciences) uses the exact same procedure as the naïve model-based approach except that it adds an exploration bonus to the reward function.

- The exploration bonus is a decreasing function of how many times a specific action is taken in a specific state, so it motivates taken actions that have been taken less often.

Exploration Bonus for multi-step horizon

- Model-based interval estimation with exploration bonus (MBIE+EB in Strehl and Littman 2008):

$$\hat{Q}^{(t)}_{\mathrm{MB}}(s,a) = \hat{R}^{(t)}(s,a) + \frac{\beta}{\sqrt{T^{(t)}_{s,a}}} + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}^{(t)}_{\mathrm{MB}}(s',a')$$

- Theorem 2 in Strehl and Littman 2008:

  MBIE+EB is Probably Approximately Correct for MDPs (= it is **PAC-MDP**).

  = loosely speaking, its choices are **good enough** with **high probability**.

- Alternative: Bayesian Exploration Bonus (BEB) by Kolter and Ng 2009

Bonus $= \dfrac{\beta}{1+T^{(t)}_{s,a}}$

It is not PAC-MDP but is near-Bayesian.

Theorem 2. Exploration based on a bonus proportional to $\left(T^{(t)}_{s,a}\right)^{-p}$ is not PAC-MDP if $p > 0.5$.

- Comments for the previous slide:

- MBIE+EB is proven to be PAC-MDP (see Strehl and Littman 2008): In short and loosely speaking, this means that, with high probability, most of the actions take by MBIE+EB are close to the actions that would have been taken by the optimal policy.

- Alternative exploration bonuses are possible, but they have different properties. For example, an exploration bonus proportional to one over $T_{s,a}^{(t)}$ is not PAC-MDP but is “near Bayesian” (i.e., another notion of optimality; see Kolter and Ng in ICML 2009).

Quiz: exploration Bonus (multi-step horizon)

?

- An exploration bonus $\frac{\beta}{\sqrt{T_{s,a}^{(t)}}}$ is always worse for MDPs than $\frac{\beta}{T_{s,a}^{(t)}}$.

?

- An exploration bonus $\frac{\beta}{\sqrt{T_{s,a}^{(t)}}}$ decreases more slowly than $\frac{\beta}{T_{s,a}^{(t)}}$.

Summary: Exploration Bonus for multi-step horizon

- Adding exploration bonus provably improves the performance of RL algorithms.

- Hence, to optimally seek a reward, best seek a ‘modified reward’ .

- There is, however, not a single (unique) approach to
    - define an exploration bonus
    - evaluate its performance.
- For MDP a possible exploration bonus:

$$\text{Bonus} = \frac{\beta}{1+T_{s,a}^{(t)}}$$

Teaching monitoring – monitoring of understanding

[ ] up to here, at least 60% of material was new to me.

[ ] I have the feeling that I have been able to follow (at least) 80% of the lecture up to here.

- Comments for the previous slide:

Artificial Neural Networks and RL

Wulfram Gerstner
EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Exploration Bonus
2. Definitions of Novelty and Surprise (tabular environment)

Previous slide.

Searching for something ‘novel’ could be a good heuristic exploration bonus.

We now turn to our intuitions of novelty and surprise.

Enjoy the images!

Novelty is not Surprise

Surprise is against models (beliefs)

Previous slide.

The video contains a sequence of about 15 flashed images.

Which ones are ‘novel’?

Which ones are ‘surprising’?

Novelty and Surprise

Q3: *What is the difference between the two?*

First answer – **novelty and surprise are not the same.**

Second answer (more precise):

Surprise is ‘against beliefs’ or ‘against expectations’ whereas novelty is not.

Previous slide.

Novelty and Surprise

Surprise is ‘against expectations’: an example

…

Previous slide.

Novelty in a tabular environment: discrete states

events = states s (e.g., one image). Total number is |s|

Novelty n:

1) count events of type s up to time t: $C^t(s)$

2) a higher count gives lower novelty.

3) the agent has spent a time t in the environment

4) the empirical observation frequency is $p_N(s) = \frac{C^t(s) + 1}{t + |s|}$

Definition: The **'Novelty' of a state s** at time t is

$$n_t(s) = -\log p_N(s)$$

Previous slide.

Novelty can be defined empirically as the negative logarithm of the empirical frequency.

This definition gives

- At the beginning (t=0), all states have the same high novelty (related to the total number of known states.
- The novelty of state s goes down if it has been observed several times, since its count increases.
- If a state has not been observed for a long time, it will slowly become novel again as time increases – and during that time other states have been observed.

Surprise in a tabular environment: discrete states and actions

events = transitions (s,a→s') given action a in state s.

Surprise $\boldsymbol{S}$**:**

1) count events of type (s,a→s') up to time t: $C^t(s, a \to s')$
2) a higher count gives lower surprise.
3) the agent has spent a time t in the environment
4) the empirical observation frequency is

$$p^t\ (s_{t+1} = s'|s_t, a_t) = \frac{C^t(s, a \to s') + 1}{\widetilde{C}^t(s, a) + |s|}$$

Definition: The **'Surprise' of a transition is**

$$S_{BF}^{t+1}(s') = \frac{prior}{p_s^t(s_{t+1} = s'|s_t, a_t)}$$

Bayes Factor Surprise

Previous slide.

Surprise is related to expectation – if you do not expect something, then you cannot be surprise. Hence surprise needs contexts and experience that enable an agent to build a belief. Expectations arise from the belief.

While novelty is derived from observation counts of states, surprise is derived from observation counts of transitions.

There are several definitions of surprise.
The specific surprise considered her is the Bayes Factor Surprise.

Definitions of Novelty and Surprise

Q1: What is novelty?

Definition: The **'Novelty' of a state s** is

$$n^t(s) = -\log p_N(s)$$

Q2: What is surprise?

Definition: The **'Surprise' of a transition is**

$$S_{BF}^{t+1}(s') = \frac{prior}{p_S^t(s_{t+1} = s' \mid s_t, a_t)}$$

There are 17 different definitions of surprise.
This here is the Bayes-Factor surprise.

Modirshanechi et al. (2022)

Previous slide. Summary.

Note that there are also other definitions of surprise.

Artificial Neural Networks and RL

Wulfram Gerstner

EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Formal Exploration Bonus
2. Definitions of Novelty and Surprise (tabular environment)
3. Why is Surprise useful?

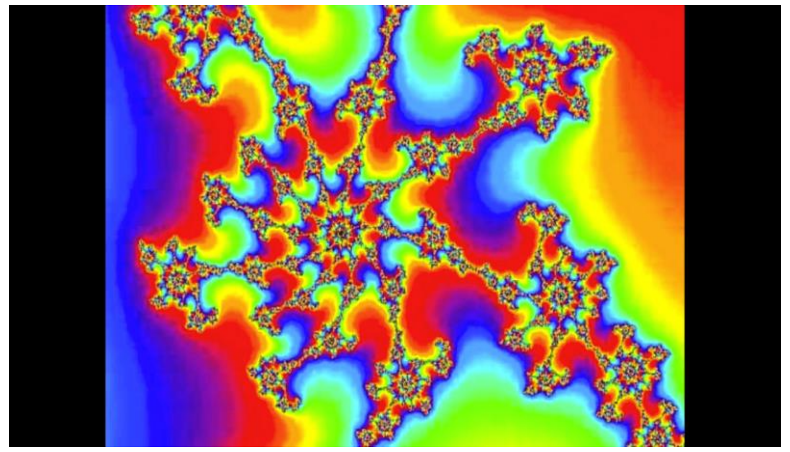
Previous slide. Summary.

We now turn to Question 4. Why is surprise (or novelty) useful?

We start with surprise.

When are we surprised?

3 9 7 3 9 7 3 9 7 3 9 7 3 9 4 3 9 7



Surprise against expectations from your current belief

- Expectations arise from models of the world
- We always make models
- We know that the models are not perfect
- **Surprise enables us to adapt the models**

→ Hypothesis:

Surprise boosts plasticity (3rd factor)/ increases the learning rate

Note: no reward!!!!

Previous slide. Review

Similar to the video with the fractals, the series of numbers has a surprising element.

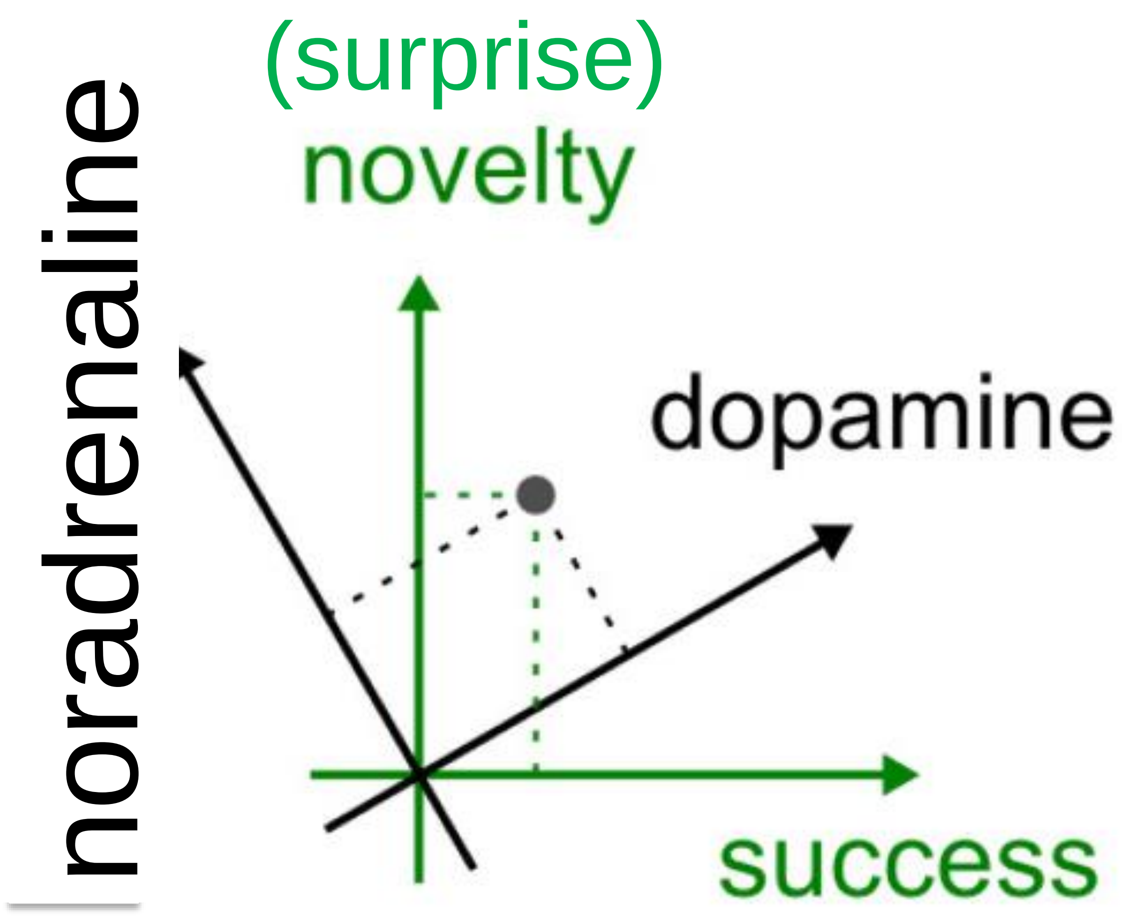
The world around us is incredibly complex. We try to understand it by making models. However, our brain is prewired (inference prior set by evolution) so that we know that our models are simplified and wrong.

At the moment when our expectations arising from our world model is wrong we get a surprise signal. The use of the surprise signal is to increase the learning rate so that we can rapidly re-adapt our model.

Review: Neuromodulators

- 4 or 5 neuromodulators
- near-global action
- internally created signals

Dopamine/reward/TD:
Schultz et al., 1997,
Schultz, 2002

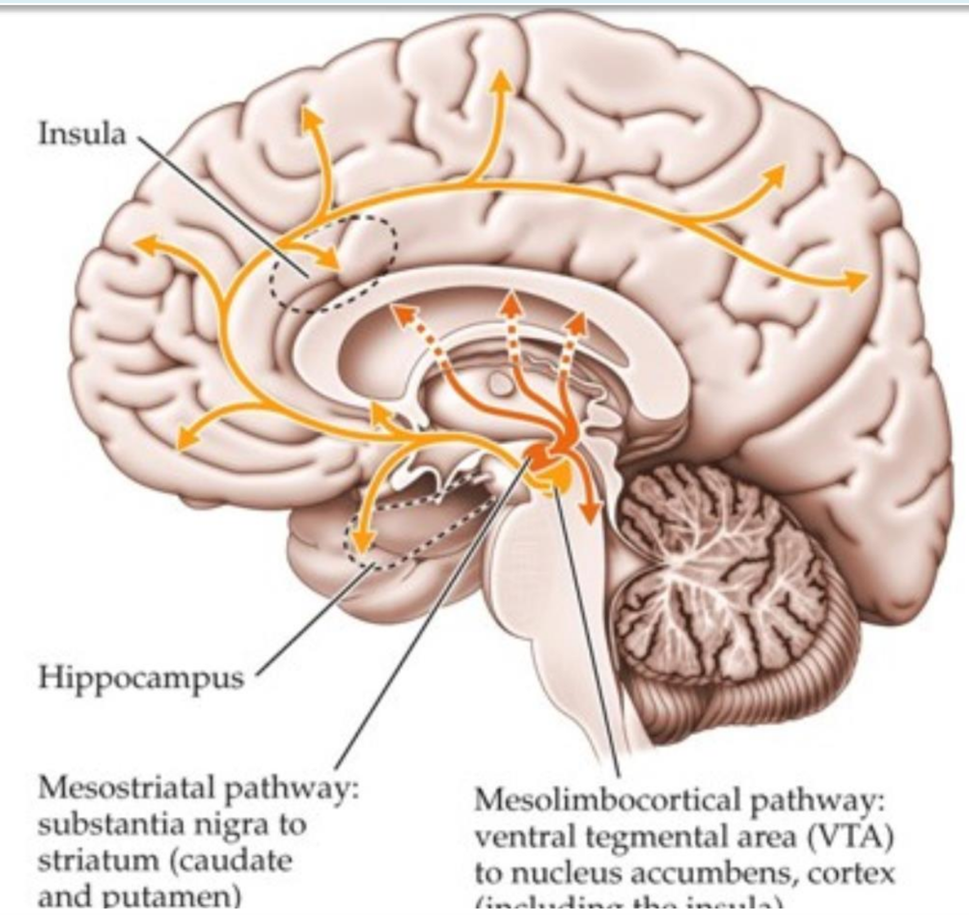


$$[r+\gamma V(s')-V(s)]$$

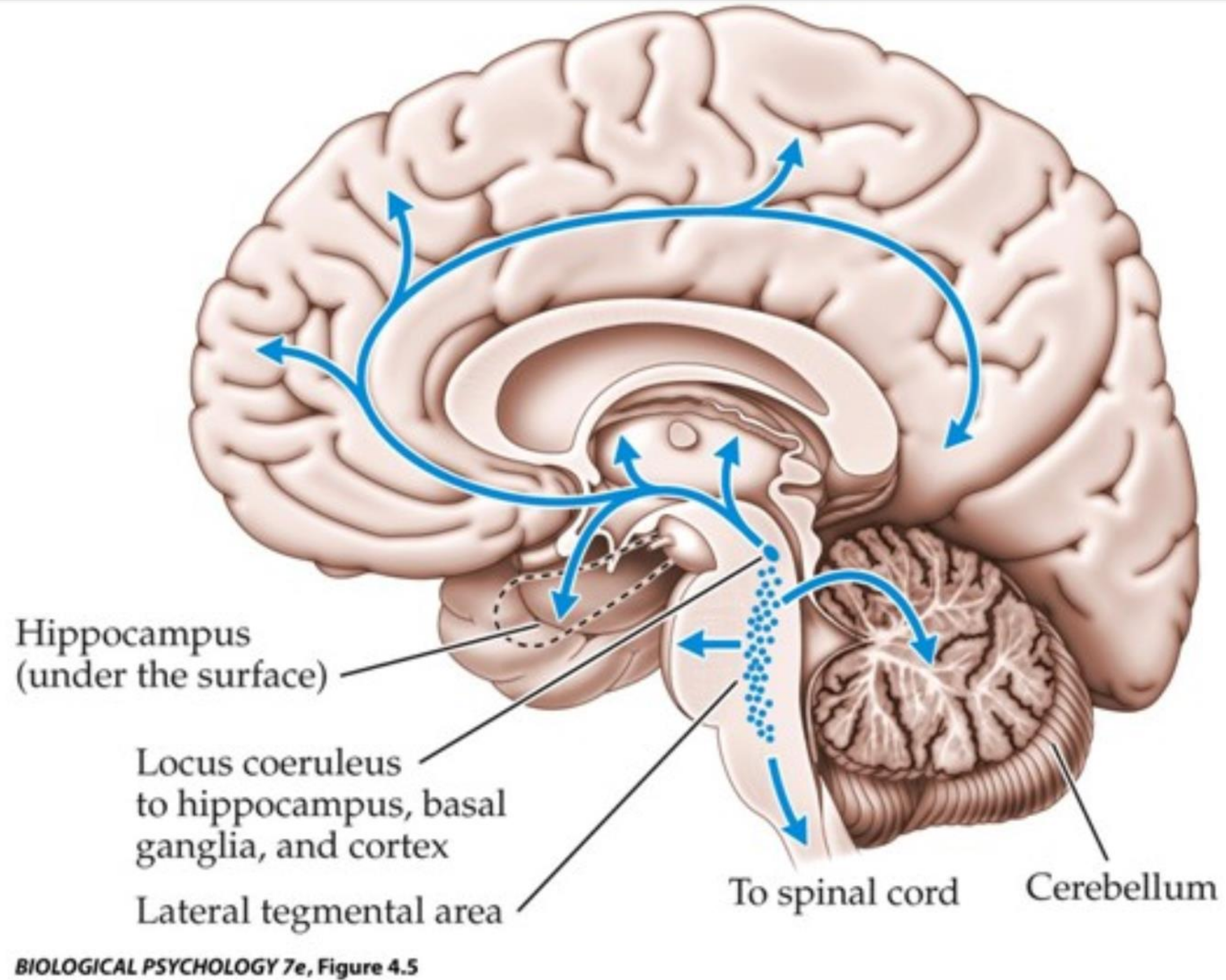
Image:
Fremaux and Gerstner, Frontiers (2016)

Image: Biological Psychology, Sinauer

Dopamine (DA)



Noradrenaline (NE)



Previous slide. Review
The most famous neuromodulator is dopamine (DA) which is related to reward, as we will see.

But there are other neuromodulators such as noradrenaline (also called norepinephrine, NE) which is related to surprise.

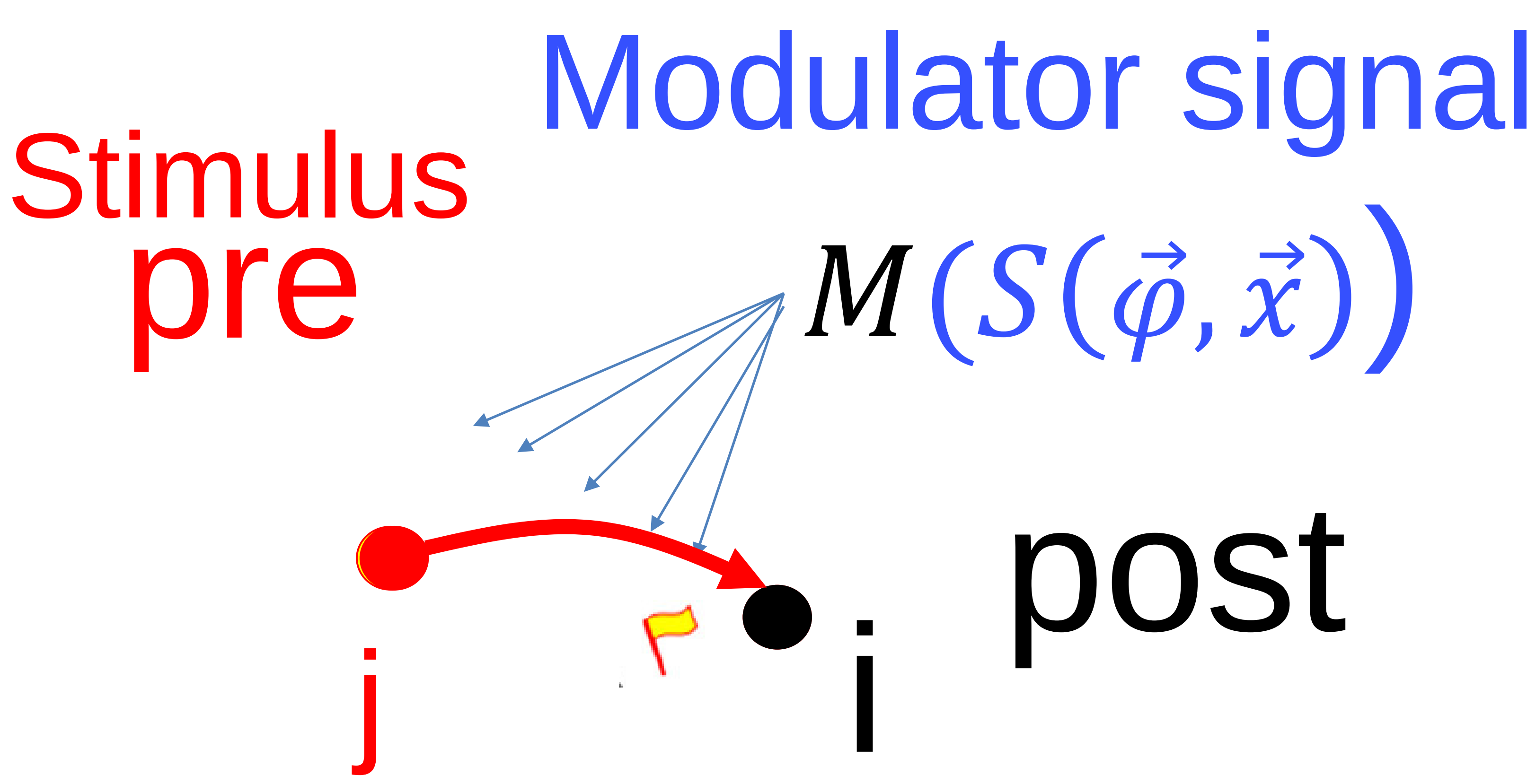
Left: the mapping between neuromodulators and functions is not one-to-one.
Indeed, dopamine also has a ‘surprise’ component.
Inversely, noradrenaline also has a reward component.
Right: most neuromodulators send axons to large areas of the brain, in particular to several cortical areas. The axons branch out in thousands of branches.
Thus the information transmitted by a neuromodulator arrives nearly everywhere.
In this sense, it is a ‘global’ signal, available in nearly all brain areas.

Note that the TD error is an internally created signal. The TD can be positive at time t even if no explicit reward is given at time t.
Similarly, surprise is an internally generated signal indicating model mismatch.

Review: Formalism of Three-factor rules with eligibility trace

x_j = activity of presynaptic neuron

φ_i = activity of postsynaptic neuron



Step 1: co-activation sets eligibility trace

$$\Delta z_{ij} = \eta \; f(\varphi_i)\, g(x_j)$$

Step 2: eligibility trace decays over time

$$z_{ij} \leftarrow \lambda \, z_{ij}$$

Step 3: eligibility trace translated into weight change

$$\Delta w_{ij} = \eta \, M\big(S(\vec{\varphi},\vec{x})\big) z_{ij}$$

$M(S)$:
- TD-error
- surprise

Previous slide.
Three-factor rules are implementable with eligibility traces.

1. The joint activation of pre- and postsynaptic neuron sets a ‘flag’. This step is similar to the Hebb-rule, but the change of the synapse is not yet implemented.

2. The eligibility trace decays over time

3. However, if a neuromodulatory signal M arrives before the eligibility trace has decayed to zero, an actual change of the weight is implemented.
The change is proportional to

- the momentary value of the eligibility trace
- the value of the neuromodulator signal

The neuromodulator could signal the

- TD-error
- or Surprise

Usefulness of Surprise? It modulates(similar to the TD error) the learning rate of RL! Surprising events increase the learning rate.

Artificial Neural Networks and RL

Wulfram Gerstner
EPFL, Lausanne, Switzerland

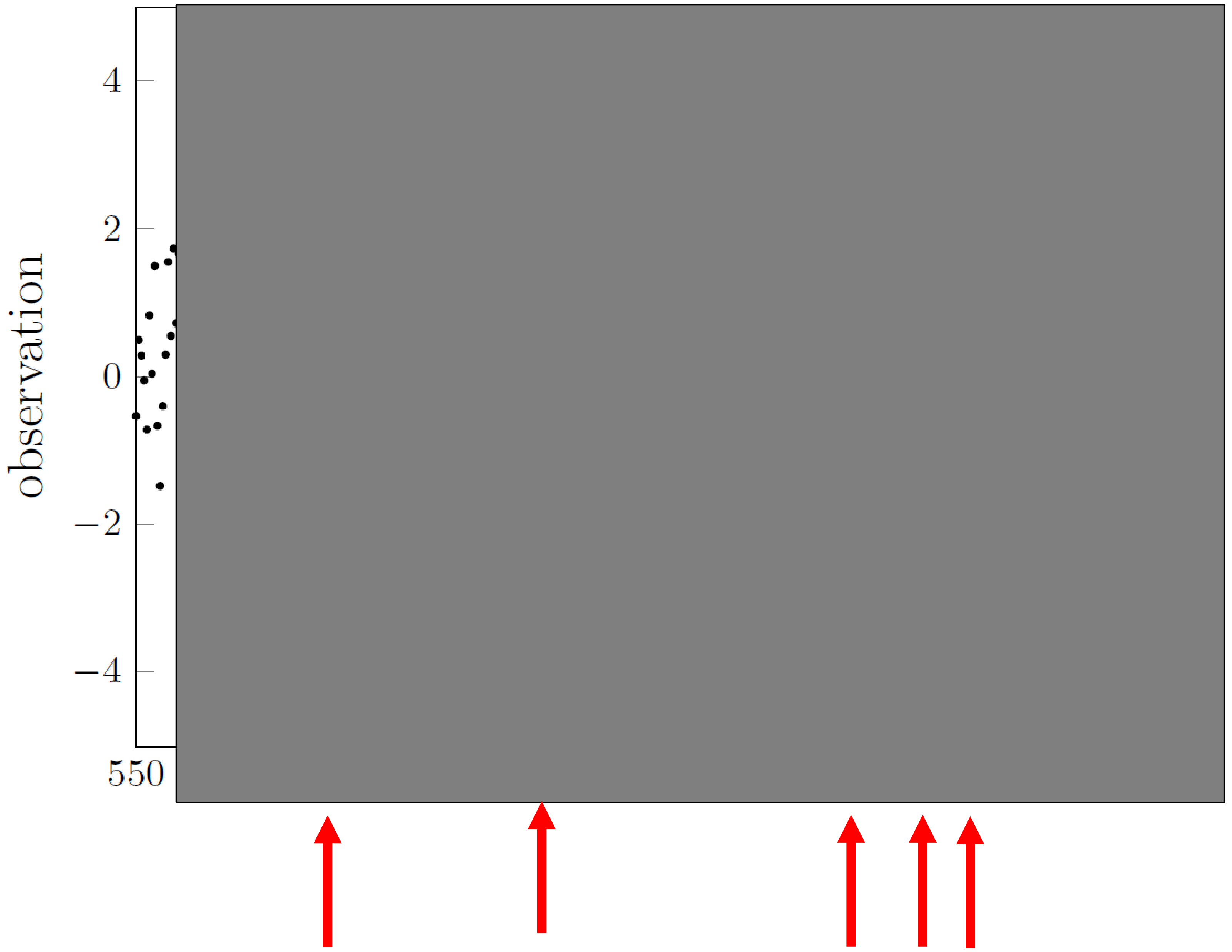
The role of exploration, novelty, and surprise in RL

1. Definitions of Novelty and Surprise (tabular environment)
2. Why is Surprise useful?
3. **Change-point detection by Bayes-Factor Surprise**

Previous slide.

Our claim is that the Bayes-Factor surprise is ideal for detecting change points.

Surprise boosts plasticity in volatile environments



Volatile environment:
abrupt changes with small probability
→ ‘change points’

→ you have to reset model after a change point

generative model = nonstationary stochastic process

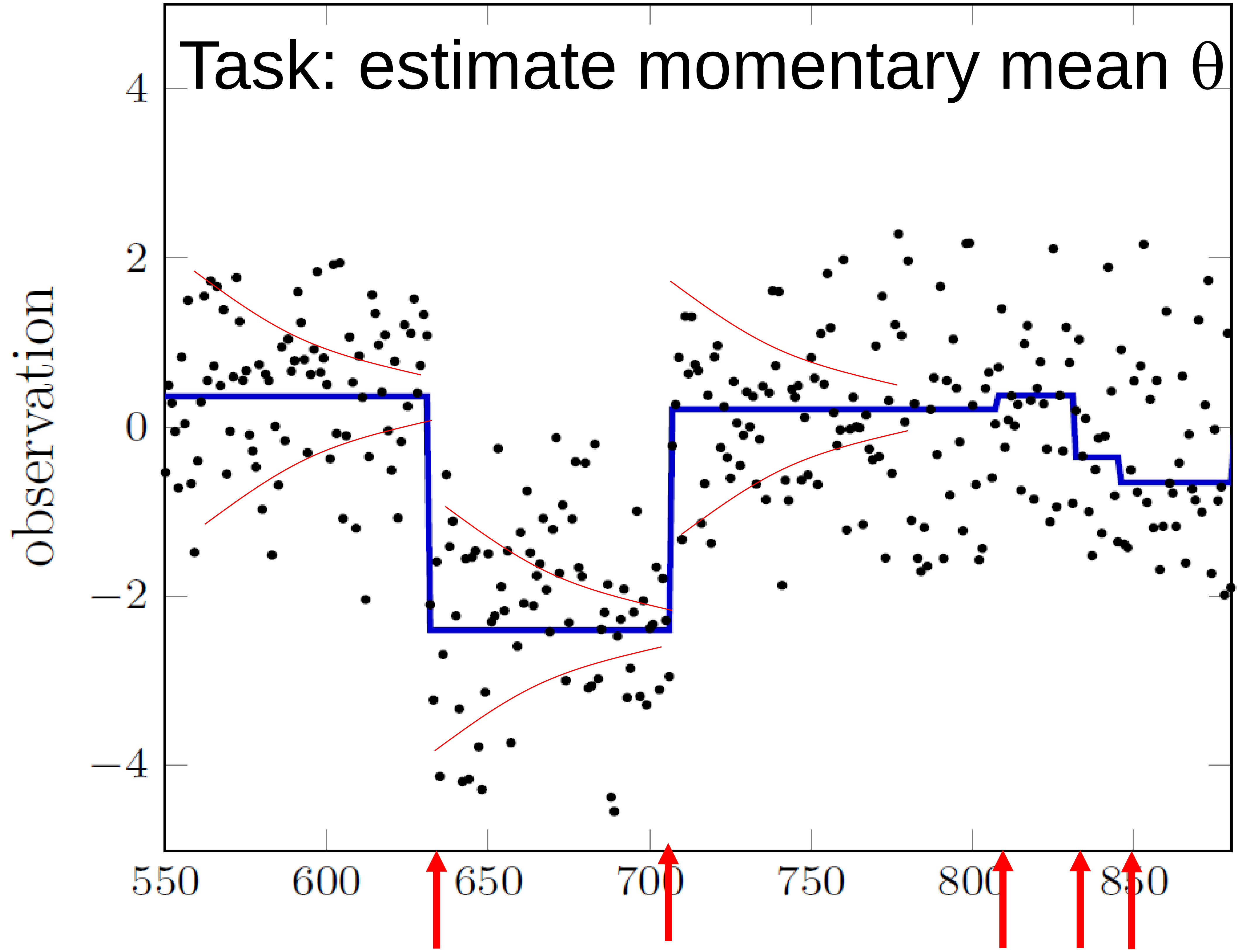
here: - mean of Gaussian is fixed for many steps
- mean jumps at ‘change points’ : probability << 1
- variance is fixed
- task is to estimate **momentary mean** of Gaussian

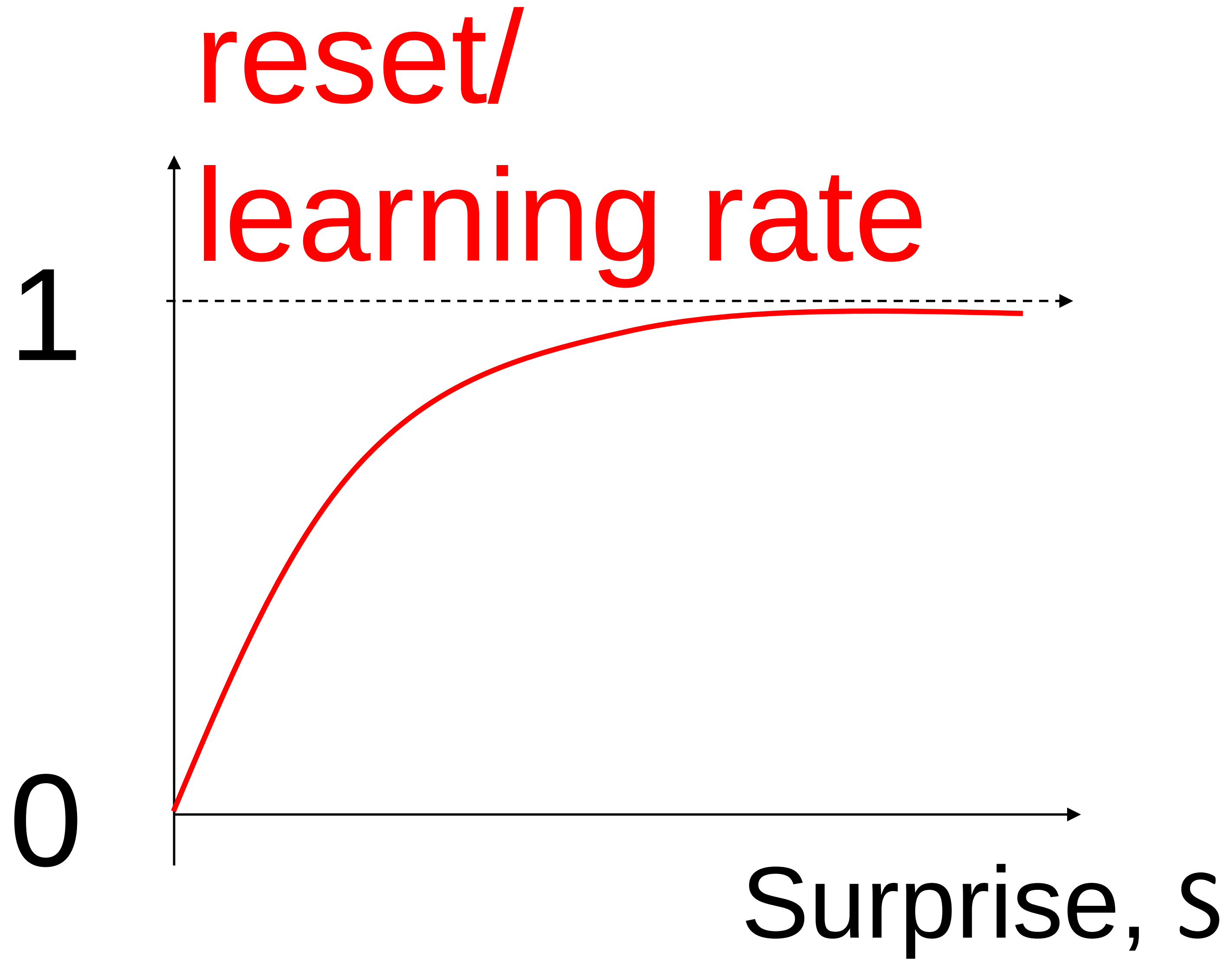
Previous slide.

The volatile environment has stationary segments, interrupted by unpredictable ‘change points’ that occur at low probability.

If you want to make predictions about the next stimulus (or here: its mean), then the best strategy is to reset your model completely if you have detected a change point.

Surprise boosts plasticity in volatile environments





in volatile environment, best approach (Bayesian):

- reset your belief to prior, if observation does not make sense
- plasticity of system must increase if ‘surprising observation’

Previous slide.

The volatile environment has stationary segments, interrupted by unpredictable change points that occur at low probability.
During the stationary segment your belief gets more precise, and your predictions (regarding the mean of the distribution) get therefore better.

But the best strategy is to reset your model completely if you have detected a change point. So the challenge is to detect the change points.
The optimal way of doing this is the Bayes-Factor surprise.
Plasticity of the model must then increase when you detect a change point, so that you reset to the prior and integrate new data points starting from the prior.

Plasticity (learning rate) of the model must then increase when you detect a change point, so that you reset to the prior and integrate new data points starting from the prior.

Surprise boosts plasticity in volatile environments

$$S_{\mathrm{BF}}(y_{t+1};\pi^{(t)}) = \frac{P(y_{t+1};\pi^{(0)})}{P(y_{t+1};\pi^{(t)})}\,.$$

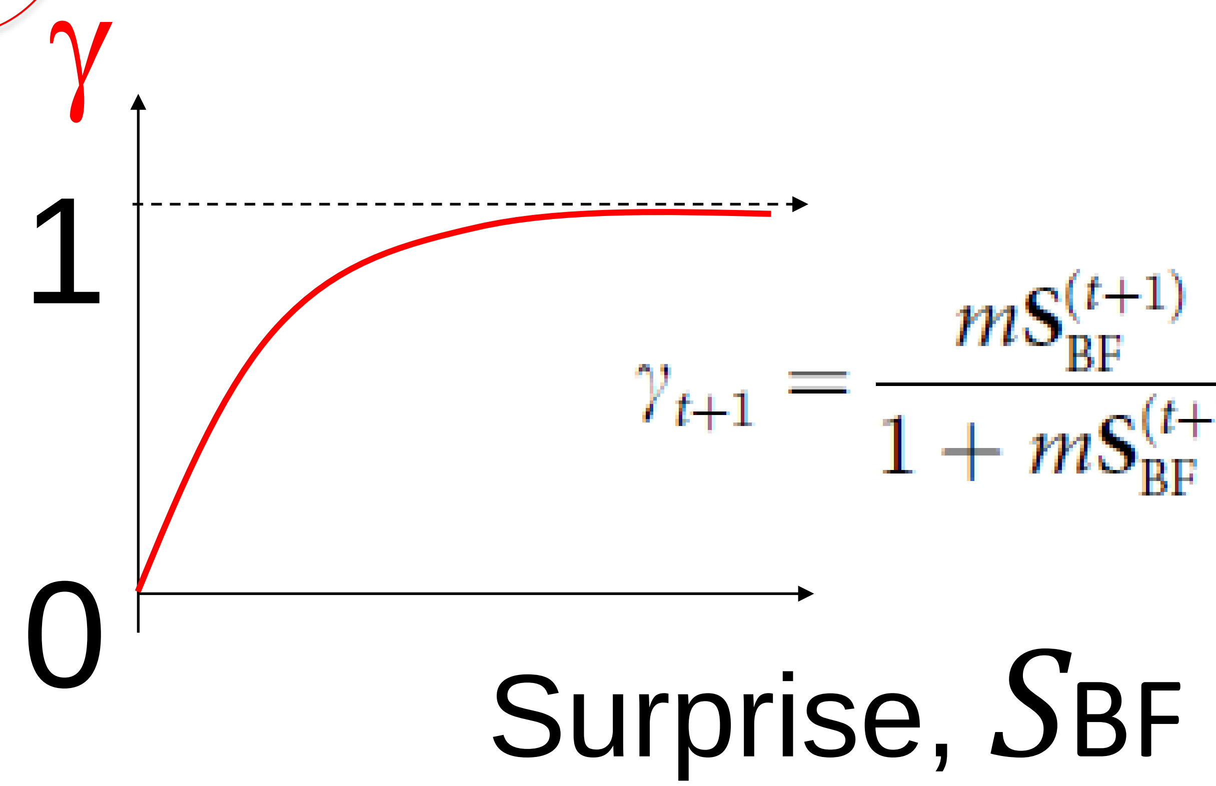
Probability of observation y under prior belief $\pi^{(0)}$

Probability of observation y under current belief $\pi^{(t)}$

→reset your belief to prior, if observation y does not make sense

$$\pi^{\mathrm{new}}(\theta) = (1-\gamma)\,\pi^{\mathrm{integration}}(\theta|y^{\mathrm{new}},\pi^{\mathrm{old}}) + \gamma\,\pi^{\mathrm{reset}}(\theta|y^{\mathrm{new}},\pi^{(0)}).$$

→'exact Bayesian inference' in volatile environment modulates update with factor γ



$$\gamma_{t+1} = \frac{m S_{\mathrm{BF}}^{(t+1)}}{1 + m S_{\mathrm{BF}}^{(t+1)}} \in [0,1],$$

Previous slide.

We claimed that plasticity (learning rate) of the model must increase when you detect a change point, so that you reset to the prior and integrate new data points starting from the prior.

This is formalized in the long equation in the middle.

Using a careful analysis of the statistical estimation in the presence of change points you find that:

If it unlikely (small γ) that there was a change point between the previous data and the current data point (observation y^new), then you should use standard statistical updates of your estimates to INTEGRATE the new data into your current belief.
If it is likely (γ close to 1) that there was a change point, then you should reset to your prior and integrate the new data point using statistical updates starting with the prior as your current belief.

Moreover, this factor γ depends monotonically on the Bayes-Factor Surprise S_{BF}

Surprise boosts plasticity in volatile environments

$$S_{\mathrm{BF}}(y_{t+1};\pi^{(t)}) = \frac{P(y_{t+1};\pi^{(0)})}{P(y_{t+1};\pi^{(t)})} .$$

Probability of observation y under prior belief $\pi^{(0)}$

Probability of observation y under current belief $\pi^{(t)}$

→reset your belief to prior, if observation y does not make sense

Exact update rule not implementable, but

Bayes-Factor Surprise plays crucial role in approximate methods:

- Particle Filter with N particles,
- Message-Passing with N messages,
- Published approximations

V. Liakoni et al., ***Neural Computation 2021***

Previous slide.

The general theoretical framework cannot be integrated out over several time steps. Therefore approximations are necessary.

However, what is important is the gist of the argument:
A high surprise indicates that the learning rate should be increased.

Summary: Definitions of Novelty and Surprise

What is novelty?

$$p_N(s) = \frac{C^t(s) + 1}{t + |s|}$$

Definition: The **'Novelty' of a state s** is

$$n^t(s) = -\log p_N(s)$$

What is surprise?

$$p^t\,(s_{t+1} = s'|s_t, a_t) = \frac{C^t(s, a \to s') + 1}{\widetilde{C}^t(s,a) + |s|}$$

Definition: The **'Surprise' of a transition is**

$$S_{BF}^{t+1}(s') = \frac{prior}{p_s^t(s_{t+1} = s' |\, s_t, a_t)}$$

There are 17 different definitions of surprise.
This here is the Bayes-Factor surprise.

Modirshanechi et al. (2022)

Summary: Why is surprise useful?

- **Detect change points** in environment statistics
- **Adapt learning rate** after change point**.**
- **Bayes-Factor Surprise** is a good surprise measure for this

Artificial Neural Networks and RL

Wulfram Gerstner
EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Definitions of Novelty and Surprise (tabular environment)
2. Why is Surprise useful?
3. Change-point detection by Bayes-Factor Surprise
4. **Why is Novelty useful?**

Previous slide.

We are done with surprise and turn now to the second part of Question 4.
Why is novelty useful?

We start with a detour in order to review well-known results from RL, in particular TD learning and eligibility traces.

Why is Novelty useful?
→ helps to explore

Next lecture at 14h15

Exercise 1. How fast can we find the goal state with a stationary policy?

Consider an environment with the state space $\mathcal{S}$, a goal (terminal) state $G \in \mathcal{S}$, and an action space $\mathcal{A}$ in non-gaol states (i.e., $\mathcal{S} - \{G\}$). After taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, the agent moves to state $s' \in \mathcal{S}$ with the transition probability $p(s'|s,a)$. These transition probabilities are unknown to the agent. We use T to denote the first time an agent find the goal state G, i.e., $s_T = G$. If we assume that the agent uses a stationary policy π, then we can define the average of T given each initial state $s \in \mathcal{S}$ as

$$\mu_\pi(s) := \mathbb{E}_\pi[T|s_0 = s],$$

where s_0 is the state at time $t = 0$. In this exercise, we study $\mu_\pi(s)$ in its most general case.

a. What is the value of $\mu_\pi(G)$?

   Hint: Note that T is equal to the smallest $t \geq 0$ when we have $s_t = G$.

b. What is the realtionship between $\mathbb{E}_\pi[T|s_1 = s]$ and $\mu_\pi(s)$?

   Hint: Note that $\mu_\pi(s)$ is the average of T if the agent starts in state s at time $t = 0$, whereas $\mathbb{E}_\pi[T|s_1 = s]$ is the average of T if the agent starts in state s at time $t = 1$.

c. Find a system of linear equations for finding $\mu_\pi(s)$ for $s \in \mathcal{S} - \{G\}$.

   Hint: Use the fact that $p_\pi(s'|s) = \sum_{a\in\mathcal{A}} \pi(a|s)p(s'|s,a)$.

Exercise 2. The magic of seeking novelty.

Previous slide.

We are done with surprise and turn now to the second part of Question 4.
Why is novelty useful?

We start with a detour in order to review well-known results from RL, in particular TD learning and eligibility traces.

Review: TD-learning in the general sense

$$Q(s,a) = \sum_{s'} P^{a}_{s \to s'} \left[R^{a}_{s \to s'} + \gamma \sum_{a'} \pi(s',a') Q(s',a') \right]$$

SARSA

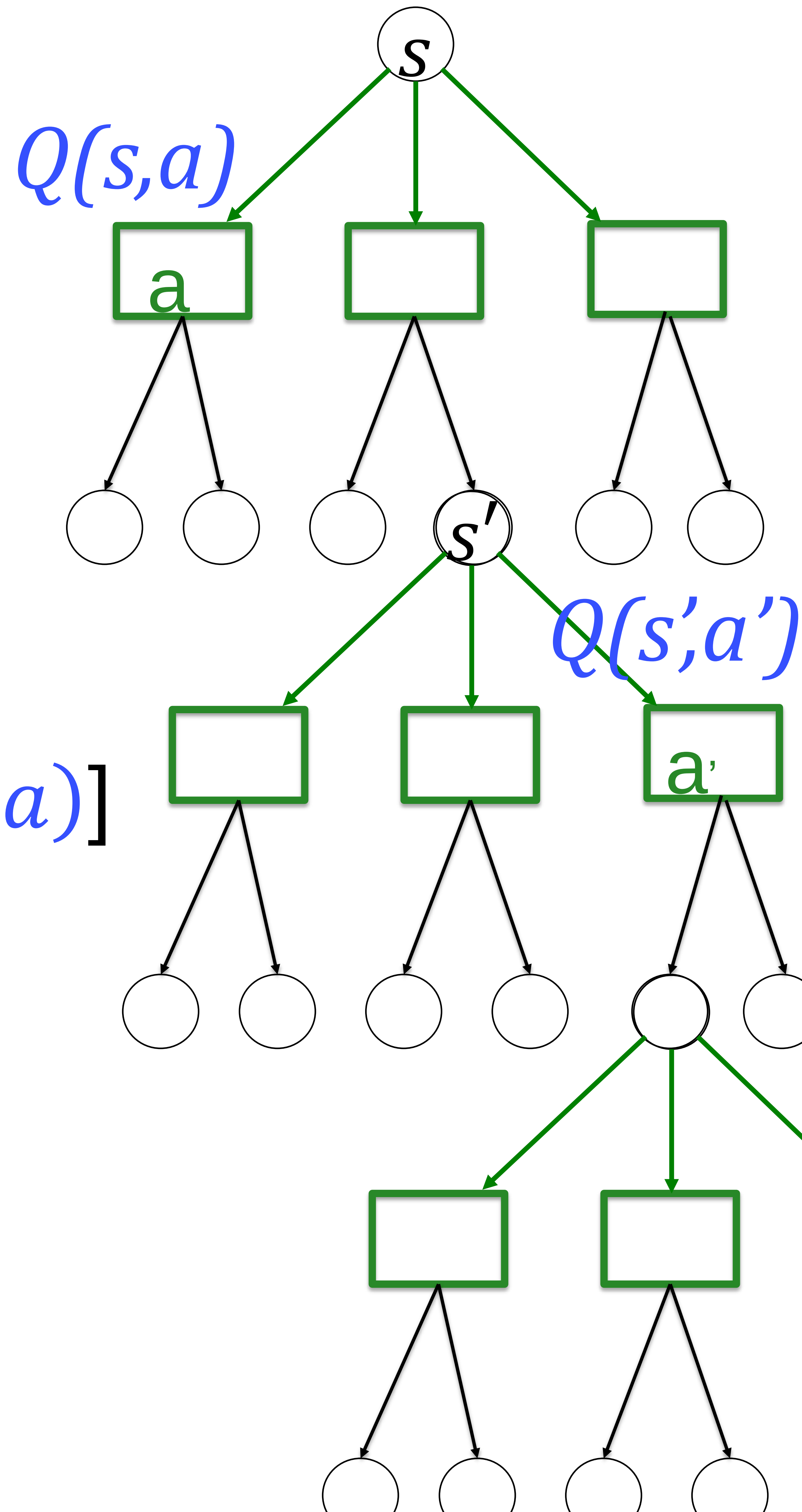
$$\Delta Q(s,a) = \eta [r_t + \gamma Q(s',a') - Q(s,a)]$$

Expected SARSA

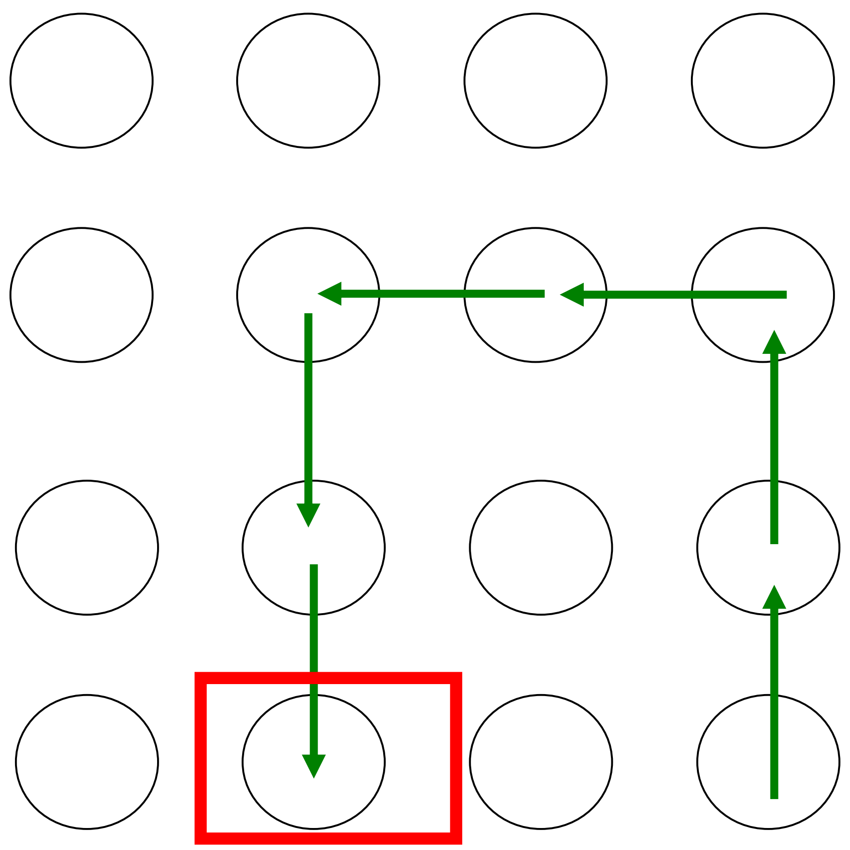
$$\Delta Q(s,a) = \eta [r_t + \gamma \{\textstyle\sum_{a'} \pi(s',a') Q(s',a')\} - Q(s,a)]$$

Q-learning

$$\Delta Q(s,a) = \eta [r_t + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$



Review: Eligibility Traces, SARSA(λ)

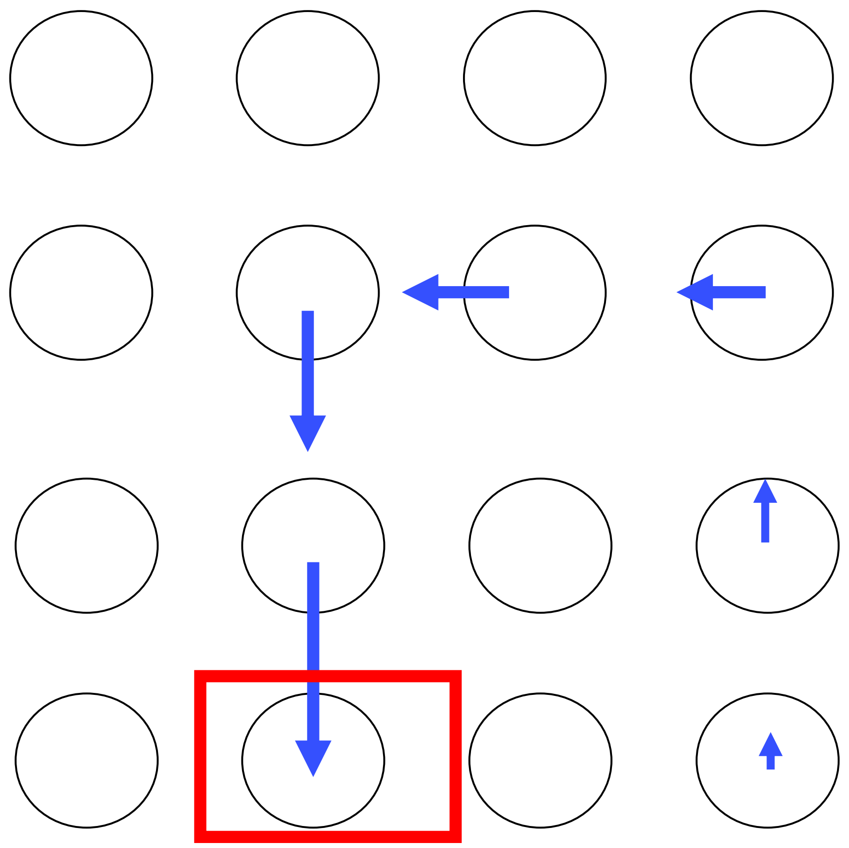


Idea:

- keep memory of previous state-action pairs
- memory decays over time
- update eligibility trace for **all** state-action pairs

$$e(s,a) \leftarrow \lambda\, e(s,a)$$ decay of **all** traces

$$e(s,a) \leftarrow e(s,a) + 1$$ if action a chosen in state s



- update **all** Q-values at **all time steps** t:

$$\Delta Q(s,a) = \eta\, [r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)]\, e(s,a)$$

RPE = TD error δ_t

Note: λ=0 gives standard SARSA

Review: Model-based versus Model-free

Model-based

- learns model of environment 'transition matrix'
- knows 'rules' of game
- planning ahead is possible
- can update Bellman equation in 'background' without action
- can simulate action sequences (without taking actions)
- is not

Model-free

- does not
- does not
- cannot plan ahead
- cannot
- cannot
- Eligibility traces and V-values keep memory of past
- completely online, causal, forward in time.

Reward-based learning versus Novelty-based learning

Reward-based learning

rewards r_t

Q-values $Q_R^{(t)}(s,a)$

Bellman eq. estimation/update

- Model-based → prioritized sweeping → $Q_{MB,R}^{(t)}(s,a)$
- Model-free → eligibility traces → $Q_{MF,R}^{(t)}(s,a)$

Novelty-based learning

novelty n_t

Q-values $Q_N^{(t)}(s,a)$

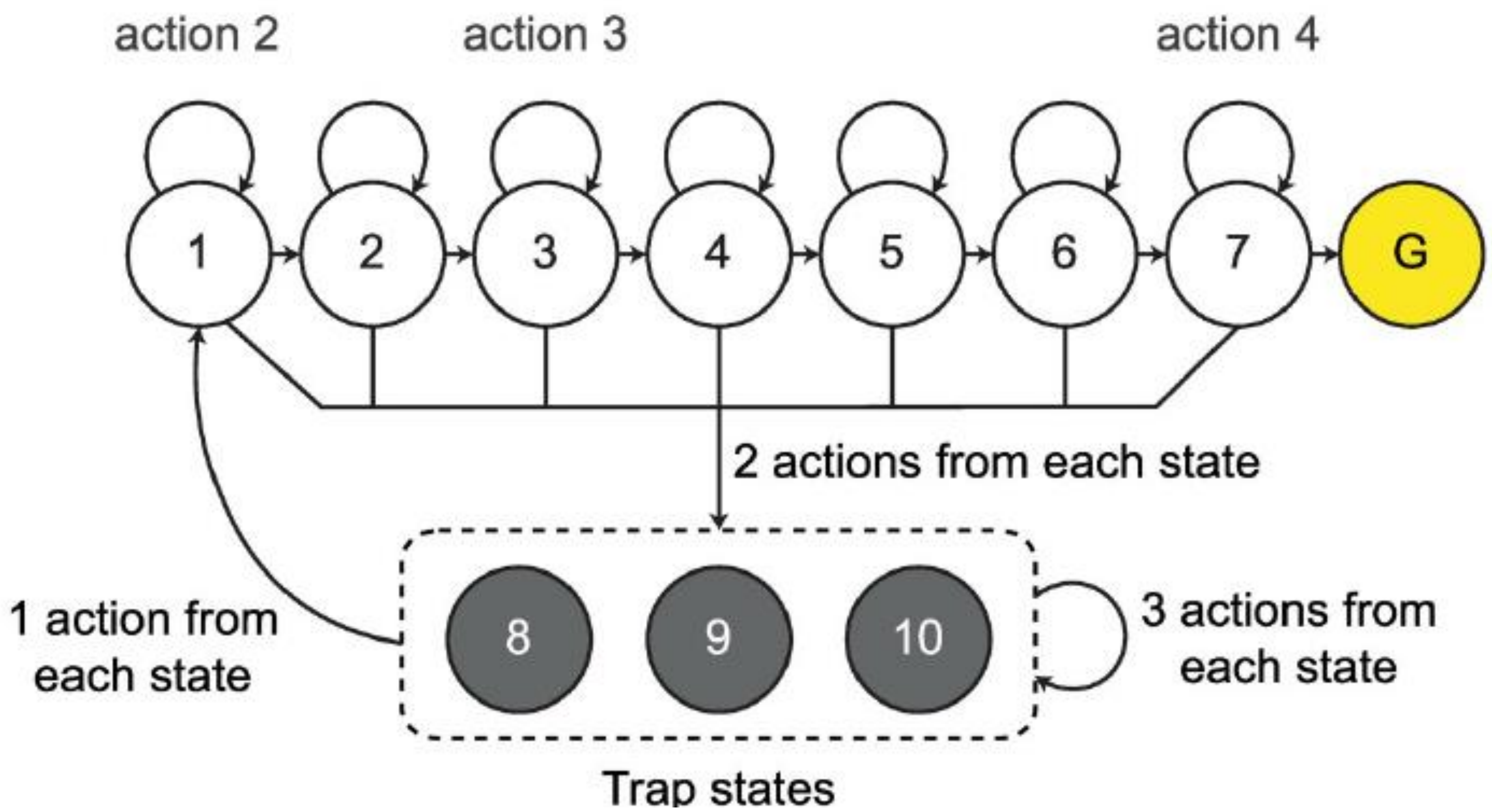
Bellman eq. estimation/update

- Model-based → prioritized sweeping → $Q_{MB,N}^{(t)}(s,a)$
- Model-free → eligibility traces → $Q_{MF,N}^{(t)}(s,a)$

Initial exploration of an environment

Environment with 10 states (+ goal)

4 actions per state



Actions are deterministic.

Fixed random assignment.

Start in state 1:
With random policy, how many actions on average before finding goal?

[ ] 100-500

[ ] 1000 – 5000

[ ] more than 10000

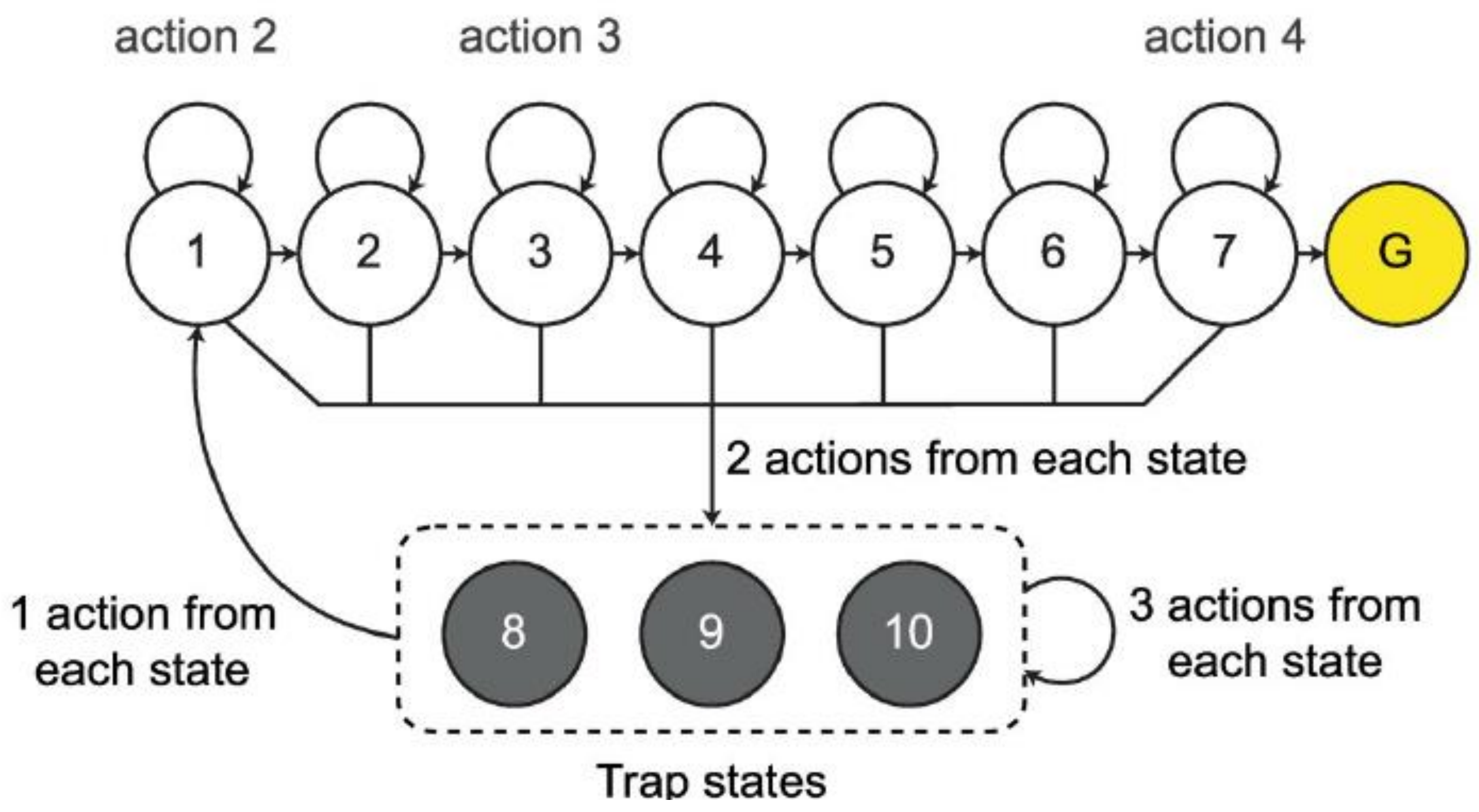
Previous slide.

With random exploration, how long would it take on average to find the goal?
There are only 10 states with four actions each, plus the goal.

Improve exploration of an environment

Focus on 1st episode, before any reward.

Improve exploration! Solutions?



1. Optimistic initialization?

Initialize $Q_R(s,a) = 10$ for all s,a

$$\Delta Q_R(s,a) = \eta[r_t + \gamma \max_{a'} Q_R(s',a') - Q_R(s,a)]$$

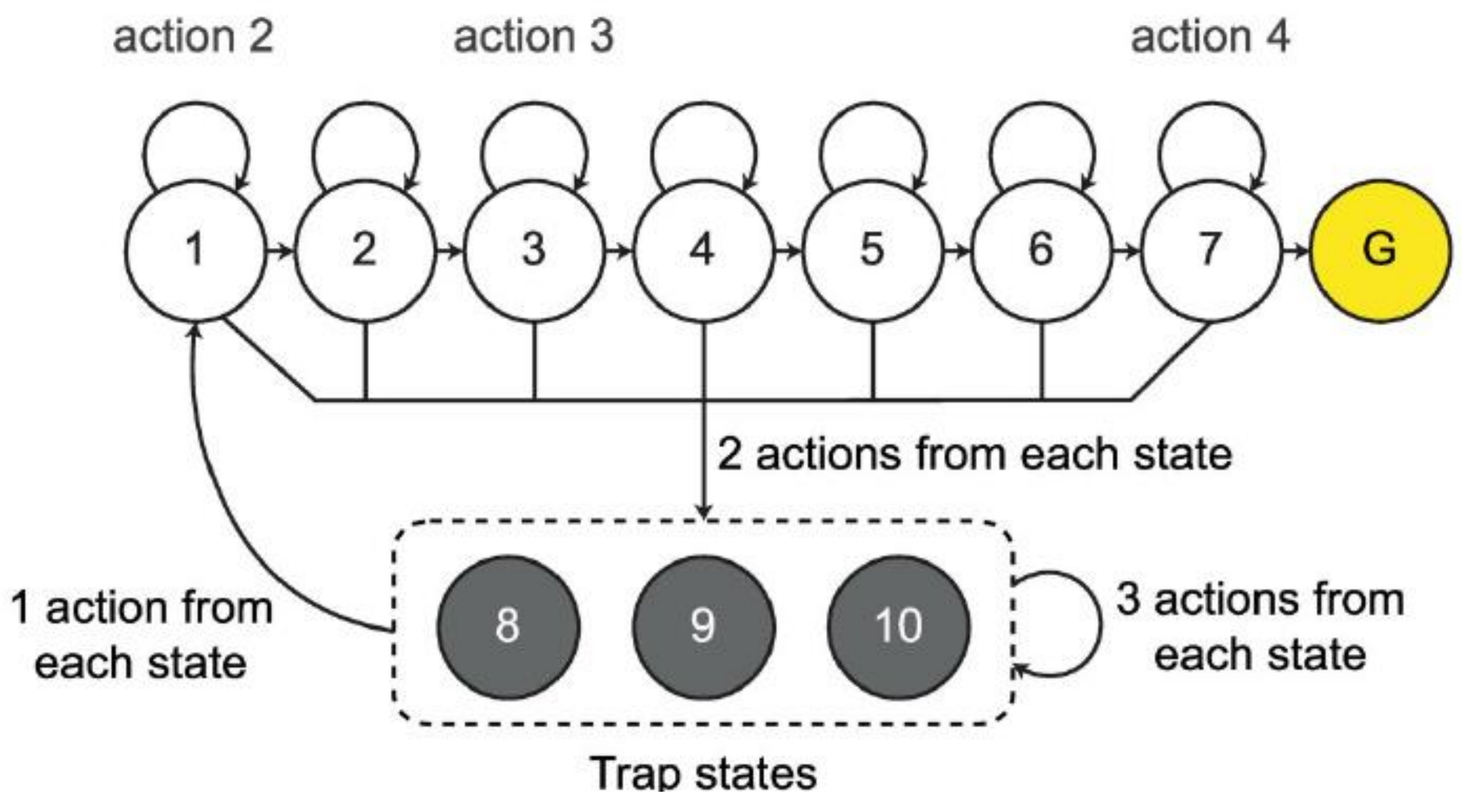
→ Possible but comparatively slow.
→ Does not generalize well for episode 2.

Previous slide.
Optimistic initialization is not sufficient to drive exploration.

Novelty encourages exploration of an environment

Focus on 1st episode, before any reward.

Improve exploration! Solutions?



2. Novelty at time t is n_t

Novelty Prediction Error (NPE)

$$\Delta Q_N(s,a) = \eta[n_t + \gamma \max_{a'} Q_N(s',a') - Q_N(s,a)]$$

→ Separate Q-value for novelty!

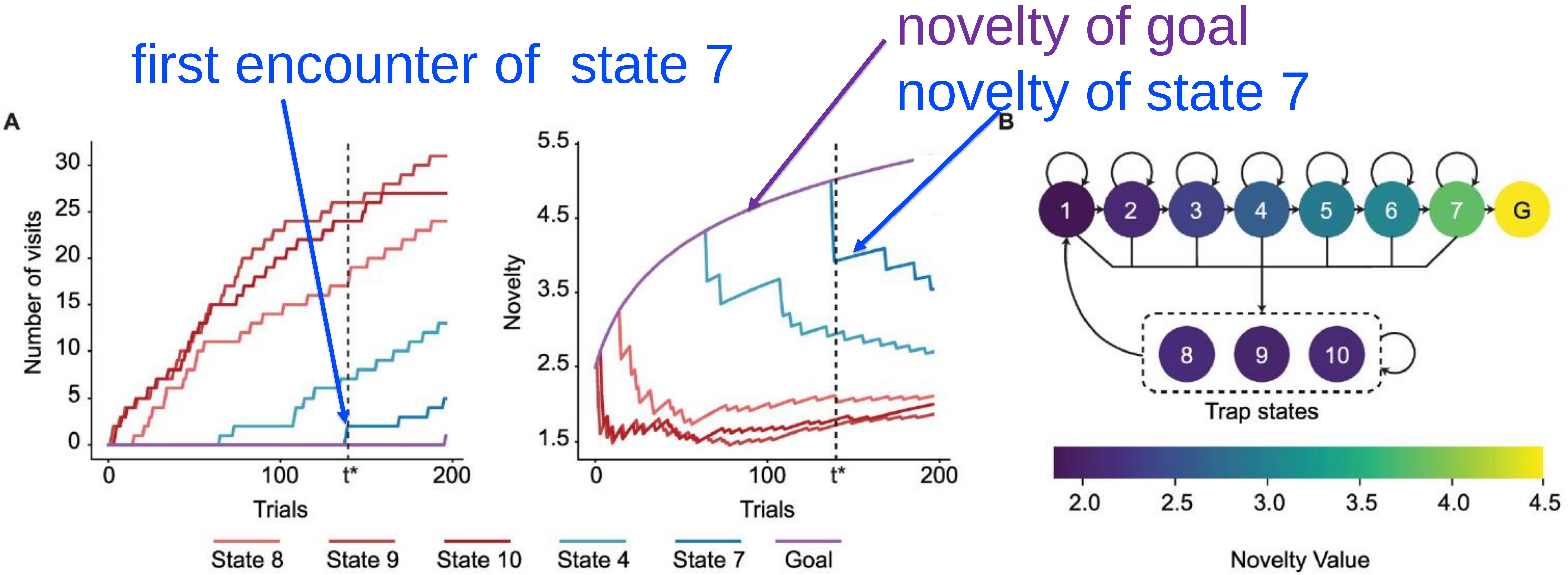
Previous slide.
We now use the novelty-Q-values.
Note that every state has some level of novelty. So the novelty prediction error NPE gives non-zero values for most transitions.

Does this lead to good novelty values? To answer this let us look at the next slide.

RPE: Reward Prediction Error
NPE: Novelty Prediction Error

Novelty encourages exploration of an environment

Focus on 1st episode, before any reward; with some policy



→ use novelty values $Q_N^{(t)}(s,a)$ for action policy!

Previous slide.

The novelty of state 7 or of the goal state increases over time during episode 1.

The plot on the right shows novelty Q-values at the moment when state 7 was found for the first time. There is a nice gradient of increasing novelty towards the goal.
This suggests that novelty Q-values are useful to guide exploration

Fig 3. Novelty in episode 1 of block 1. A. The number of state visits (left panel) and novelty (right panel) as a function of time for one representative participant: The number of visits increases rapidly for the trap states and remains 0 for a long time for the states closer to the goal. Novelty of each state is defined as the negative log-probability of observing that state (see Eqs 1 and 2) and, hence, increases for states which are not observed as time passes. The first time participants encounter state 7 (the state before the goal state) is denoted by t^*. **B**. Average (over participants) novelty (color coded) at t^*: Novelty of each state is a decreasing function of its distance from the goal state.

Artificial Neural Networks and RL

Wulfram Gerstner

EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Definitions of Novelty and Surprise (tabular environment)
2. Why is Surprise useful?
3. Change-point detection by Bayes-Factor Surprise
4. Why is Novelty useful?
5. **Hybrid Model with Novelty, Surprise, and Reward**

Previous slide.
Now we study a specific model that combines many aspects.

Reminder:

RPE: Reward Prediction Error = TD error of reward-consistency
NPE: Novelty Prediction Error = TD error of novelty consistency

Combine Novelty and Reward: ideas

→ use separate novelty values $Q_N^{(t)}(s,a)$ for action policy!
→ **exploration**

→ use separate reward values $Q_N^{(t)}(s,a)$ for action policy!
→ **exploitation**

→ Combine the two and switch relative importance
→ Switch from **exploration to exploitation (and back)**

Note: do not simply add exploration bonus!

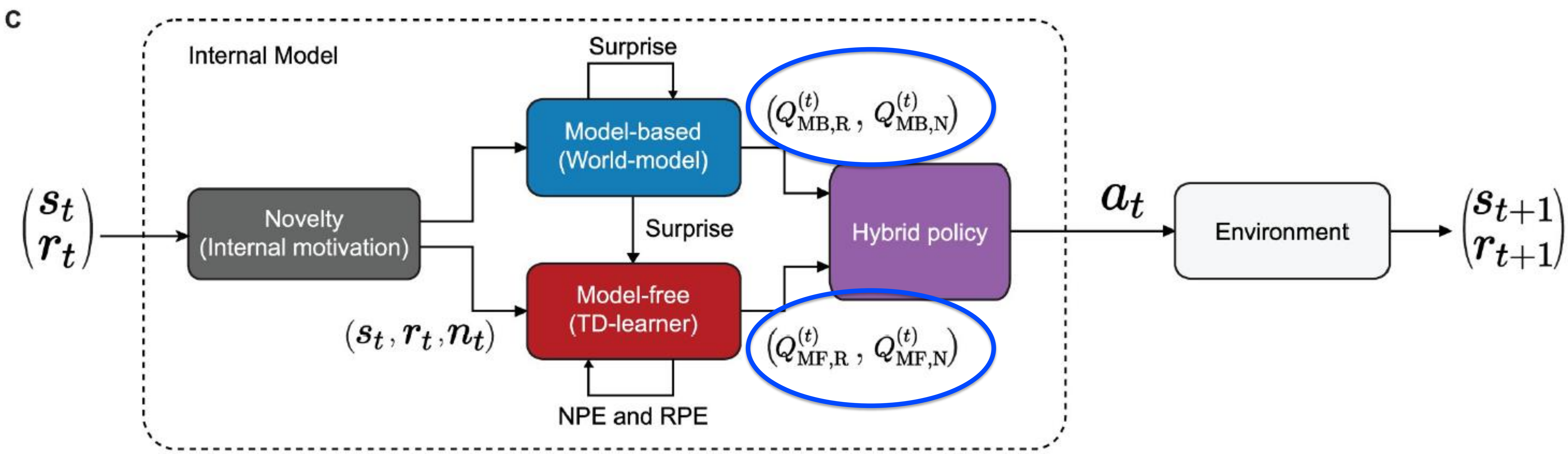
$$\hat{Q}^{(t)}_{\mathrm{MB}}(s,a) = \hat{R}^{(t)}(s,a) + \frac{\beta}{\sqrt{T^{(t)}_{s,a}}} + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}^{(t)}_{\mathrm{MB}}(s',a')$$

Previous slide.
Now we study a specific model that combines many aspects.

Reminder:

RPE: Reward Prediction Error = TD error of reward-consistency
NPE: Novelty Prediction Error = TD error of novelty consistency

Hybrid model with separate paths for Novelty and Reward (learning rate controlled by Surprise)



RPE = $[r_t + \gamma \max_{a'} Q_R(s', a') - Q_R(s, a)]$

NPE = $[n_t + \gamma \max_{a'} Q_N(s', a') - Q_N(s, a)]$

4 separate sets of Q-values!

Previous slide.

In total we have in this Hybrid model 4 sets of Q-values:

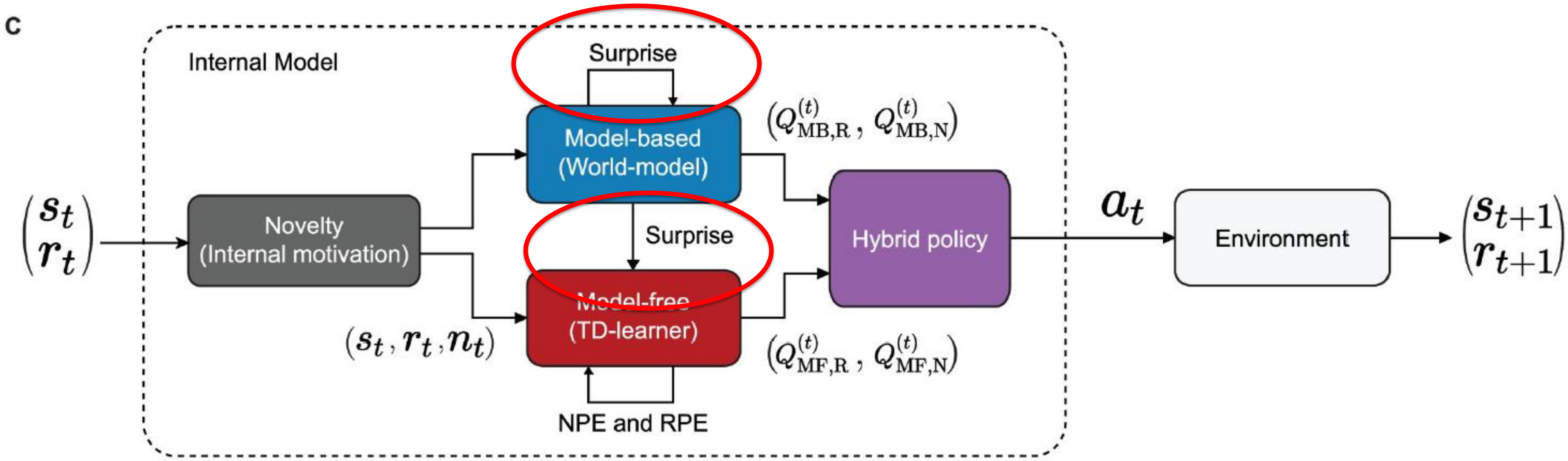
Reward-driven Q-values, in the versions model-free and model based.
Novelty-driven Q-values, in the versions model-free and model based.

All 4 Q-values are then combined in a softmax fashion to choose the best action.

The relative weighting factors can be changed.
Before the first episode, it might be good to give more importance to novelty,
and after the first episode more importance to rewards.

algorithm: Information of state s_t and reward r_t at time t is combined with novelty n_t (grey block) and passed on to the world-model (blue block, implementing the model-based branch of SurNoR) and TD learner (red block, implementing the model-free branch). The surprise value computed by the world-model modulates the learning rate of both the TD-learner and the world-model. The output of each block is a pair of Q-values, i.e, Q-values for estimated reward $Q_{MF,R}$ and $Q_{MB,R}$ as well as for estimated novelty $Q_{MF,N}$ and $Q_{MB,N}$. The hybrid policy (in purple) combines these values.

Hybrid model with separate paths for Novelty and Reward (learning rate controlled by Surprise)



World model: estimated transition matrix
→ used in model-based Q-learning for background updates
→ used to evaluate surprise (Bayes Factor Surprise)
→ surprise influences learning rate

Previous slide.

In total we have in this Hybrid model 4 sets of Q-values:

Note that the model-based version need a ‘world model’.

- The world model can be used for background updates.
- The world model contains estimated transition probabilities
- The world model can then also used to evaluate ‘surprise’
- We use the Bayes Factor surprise
- Surprise will influence the learning rates of ALL four RL algorithms

Artificial Neural Networks and RL

Wulfram Gerstner
EPFL, Lausanne, Switzerland

The role of exploration, novelty, and surprise in RL

1. Definitions of Novelty and Surprise (tabular environment)
2. Why is Surprise useful?
3. Change-point detection by Bayes-Factor Surprise
4. Why is Novelty useful?
5. Hybrid Model with Novelty, Surprise, and Reward
6. An Experiment (Markov Decision Problem for humans!)

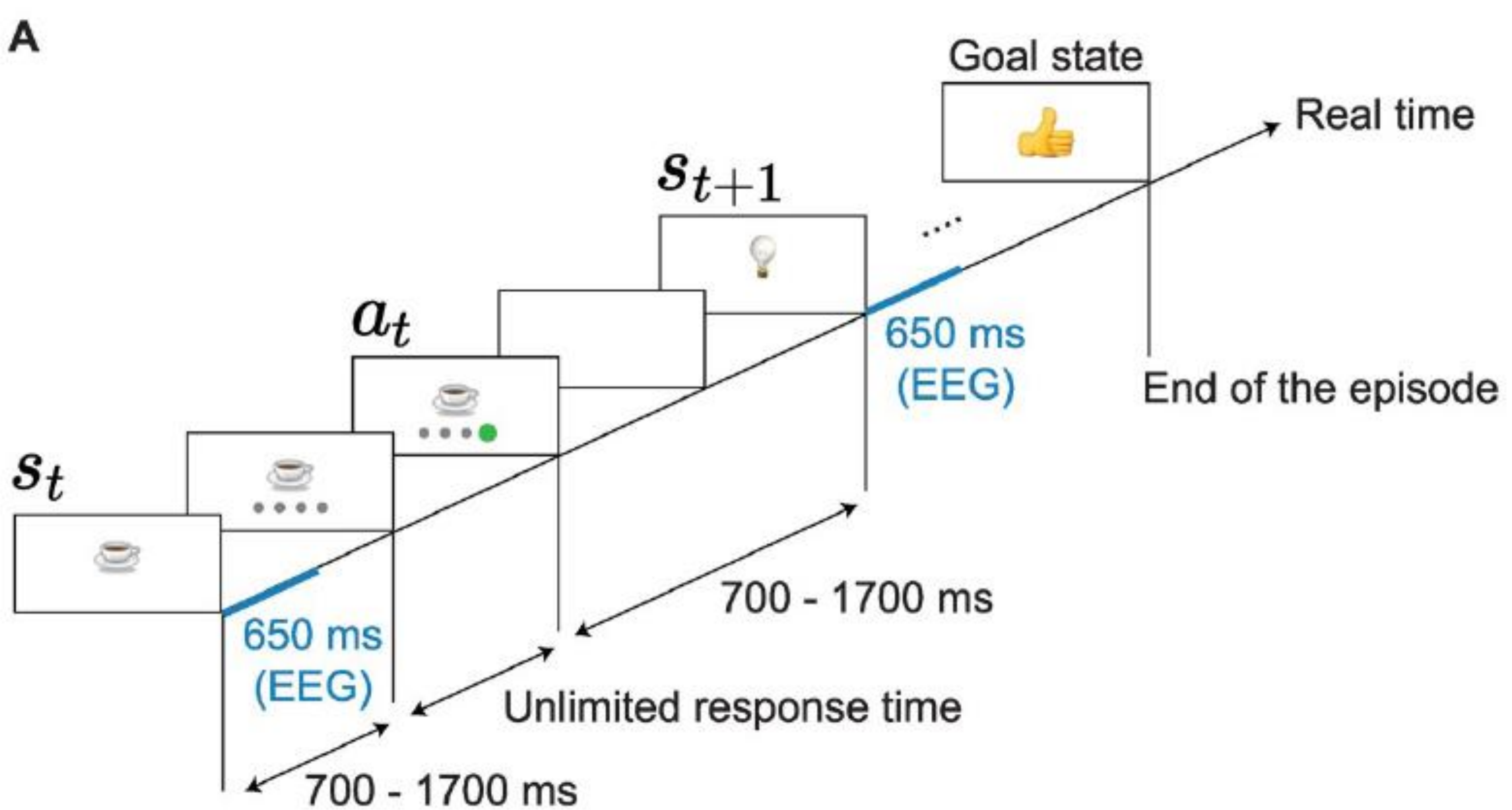
Previous slide.

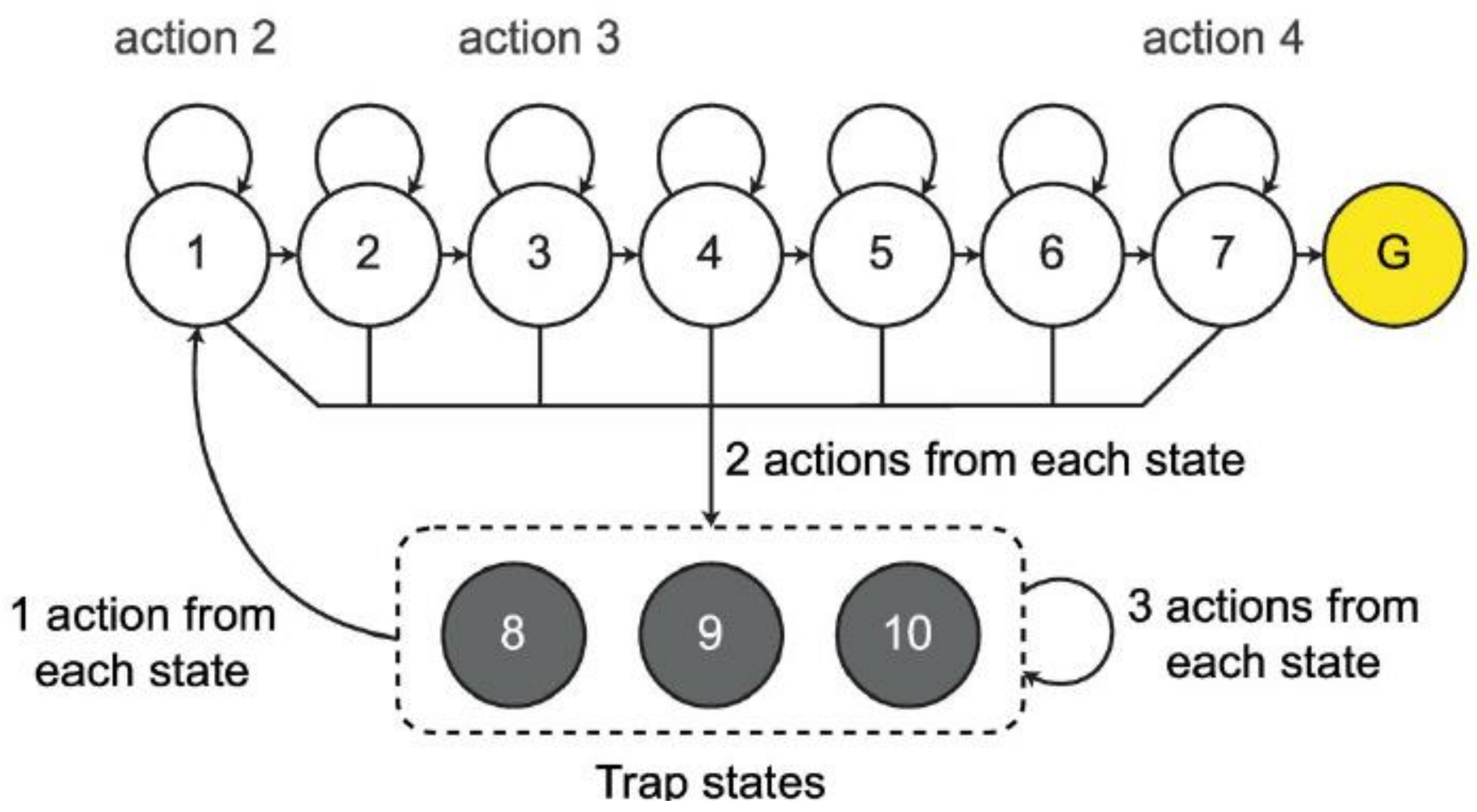
RL algorithms are inspired by human and animal behavior.

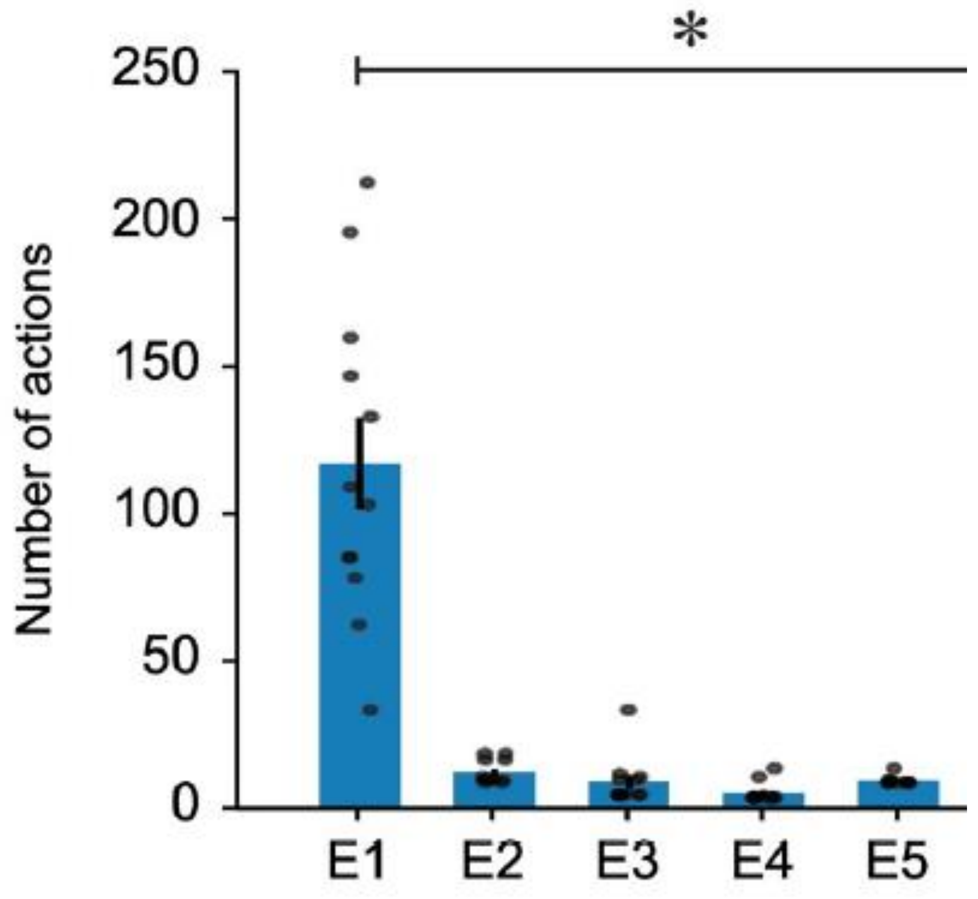
Thus, sometimes it is a good idea, how humans would perform in a given environment.

Markov Decision Processes are ideal testbeds for tabular RL algorithms.
So, let us test humans in such an environment!

Environment: Markov Decision Process







Finding 1)
Participants need about 150 actions in episode 1

Finding 2)
In episode 2, participants go straight to goal

Previous slide.

Human participants are put into a Markov Decision Process.
They have four action buttons to navigate from one image to the next.
The have been told before the experiment that there are 10 states and one goal states, each identified by an image. The 11 images (including goal) have been shown once.

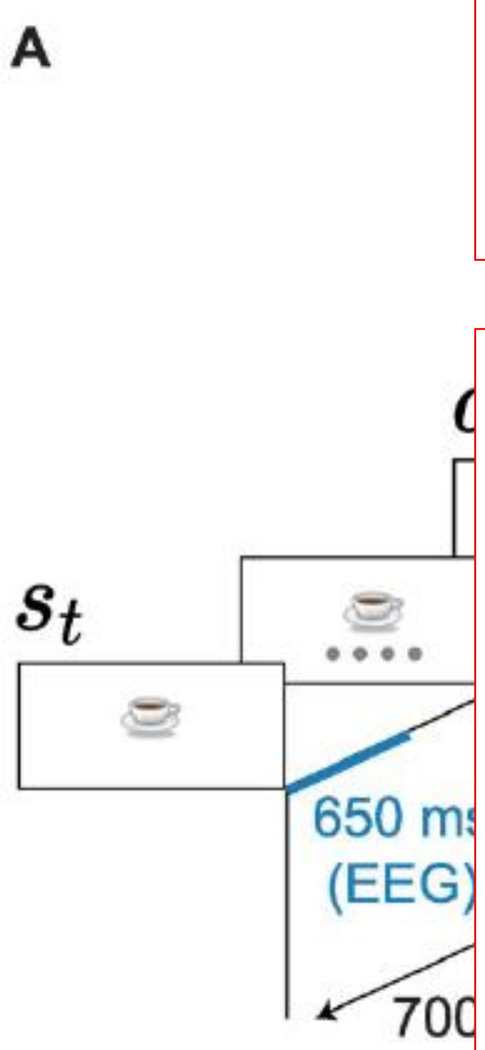
Until image onset, participants have to wait for a time of about 1s until four grey disks were present – these are the action buttons.
The goal image in this example is the thumb-up image.

Right: Structure of the environment for the first 5 episodes (block 1).

Finding1) humans are MUCH faster than the random exploration strategy to find the goal for the first time.
Finding 2) humans are extremely good in episodes 2-5 to return to the goal. The starting condition is not always state 1, but can also be a different state (varies across episodes, but the same starting state for all participants).

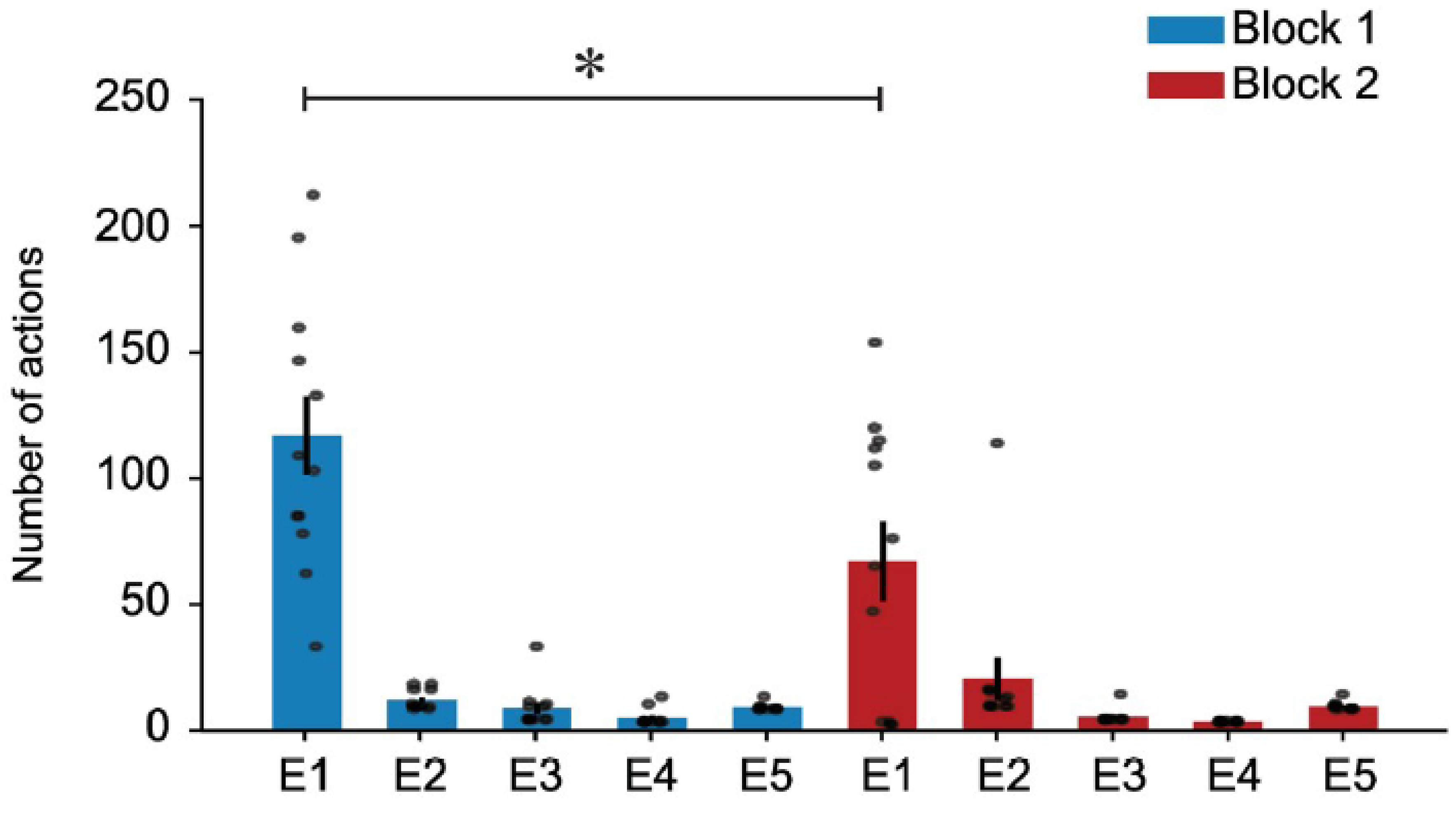
Volatile Environment: Switch after episode 5

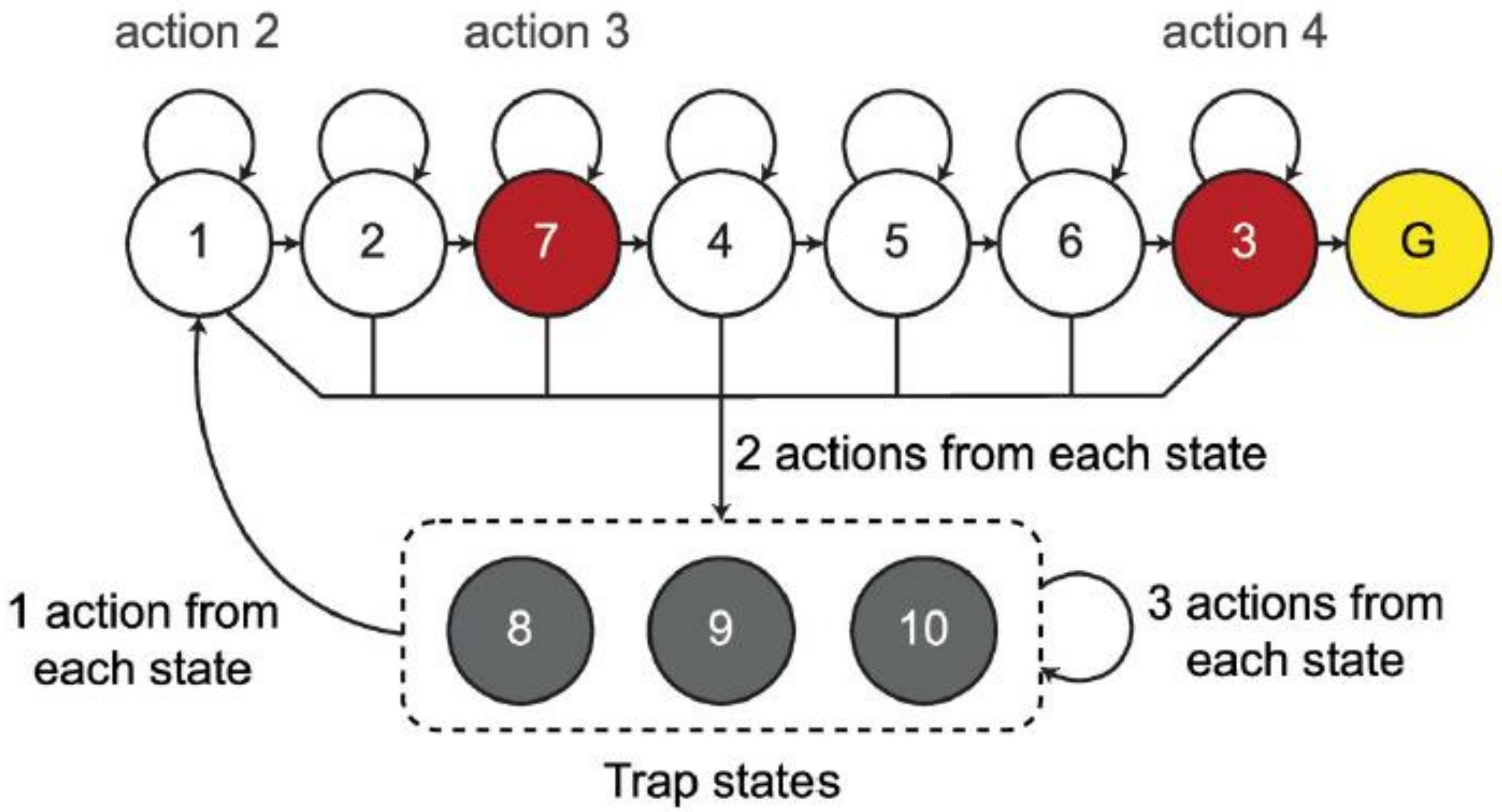
Finding 3)
In episodes 5 and 6, participants rapidly relearn!



Questions:

- Is Surprise necessary to explain relearning?
- Are humans model-based or model-free?
- Is novelty a good explanation of results?



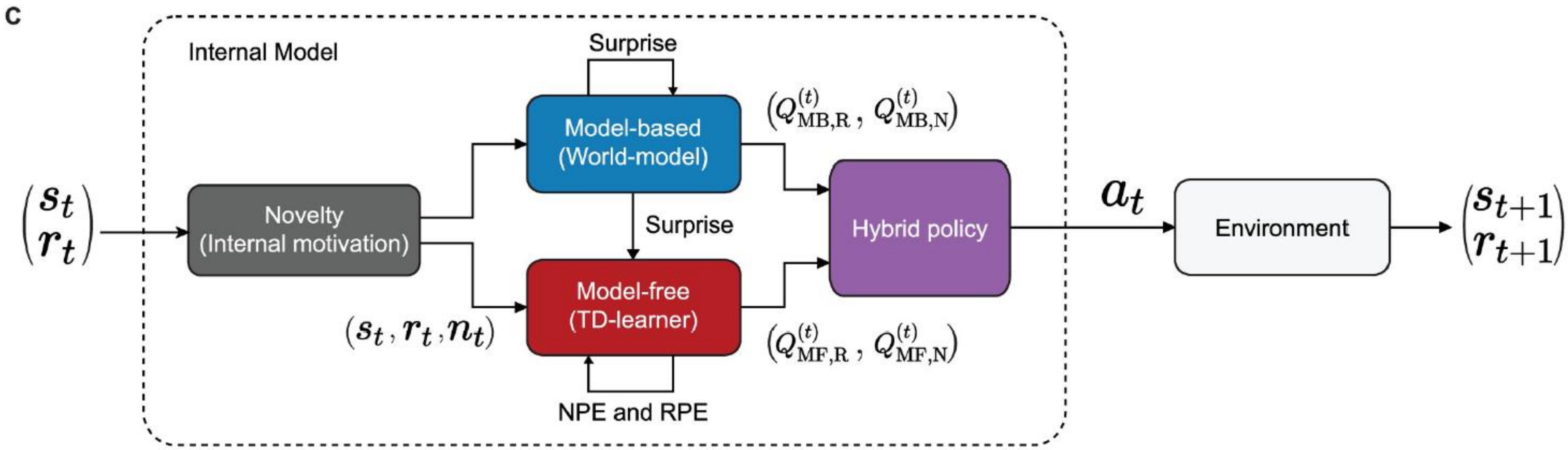


Previous slide.

After episode 5, states 3 and 7 have been swapped. Thus the environment is not stationary (volatile environment).

Humans rapidly readapt.
Would algorithms also re-adapt?

Review: Hybrid model with separate paths Surprise, Novelty, Reward (SurNoR)



RPE = $[r_t + \gamma \max_{a'} Q_R(s', a') - Q_R(s, a)]$

NPE = $[n_t + \gamma \max_{a'} Q_N(s', a') - Q_N(s, a)]$

4 separate sets of Q-values!

Previous slide.

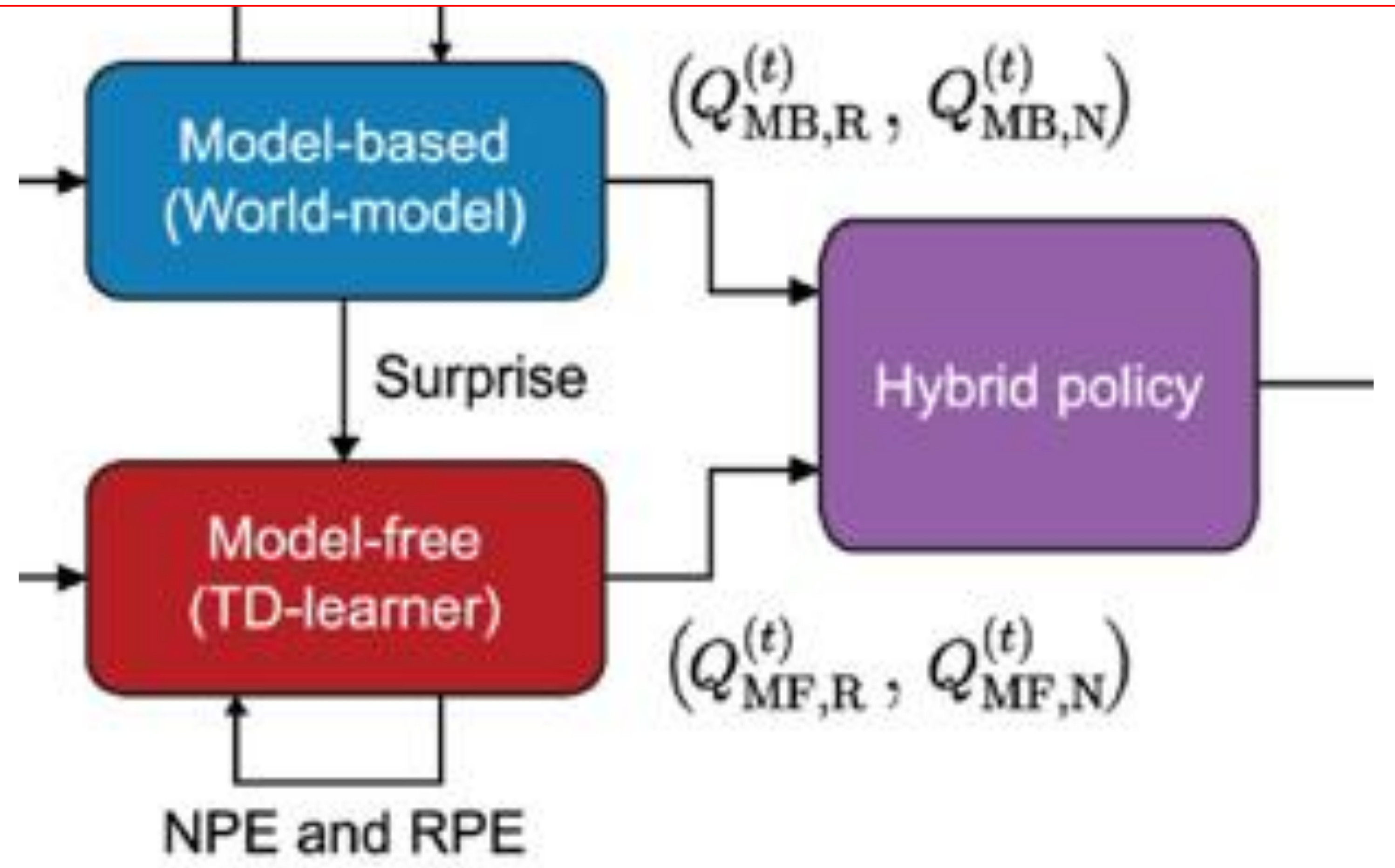
Note that in the formal theory of exploration bonus, we simply added the bonus in the Bellman equation.

However, here we claim that it is useful to develop two separate Bellman equations, one for novelty and one for reward. Each one has separate Q-values.

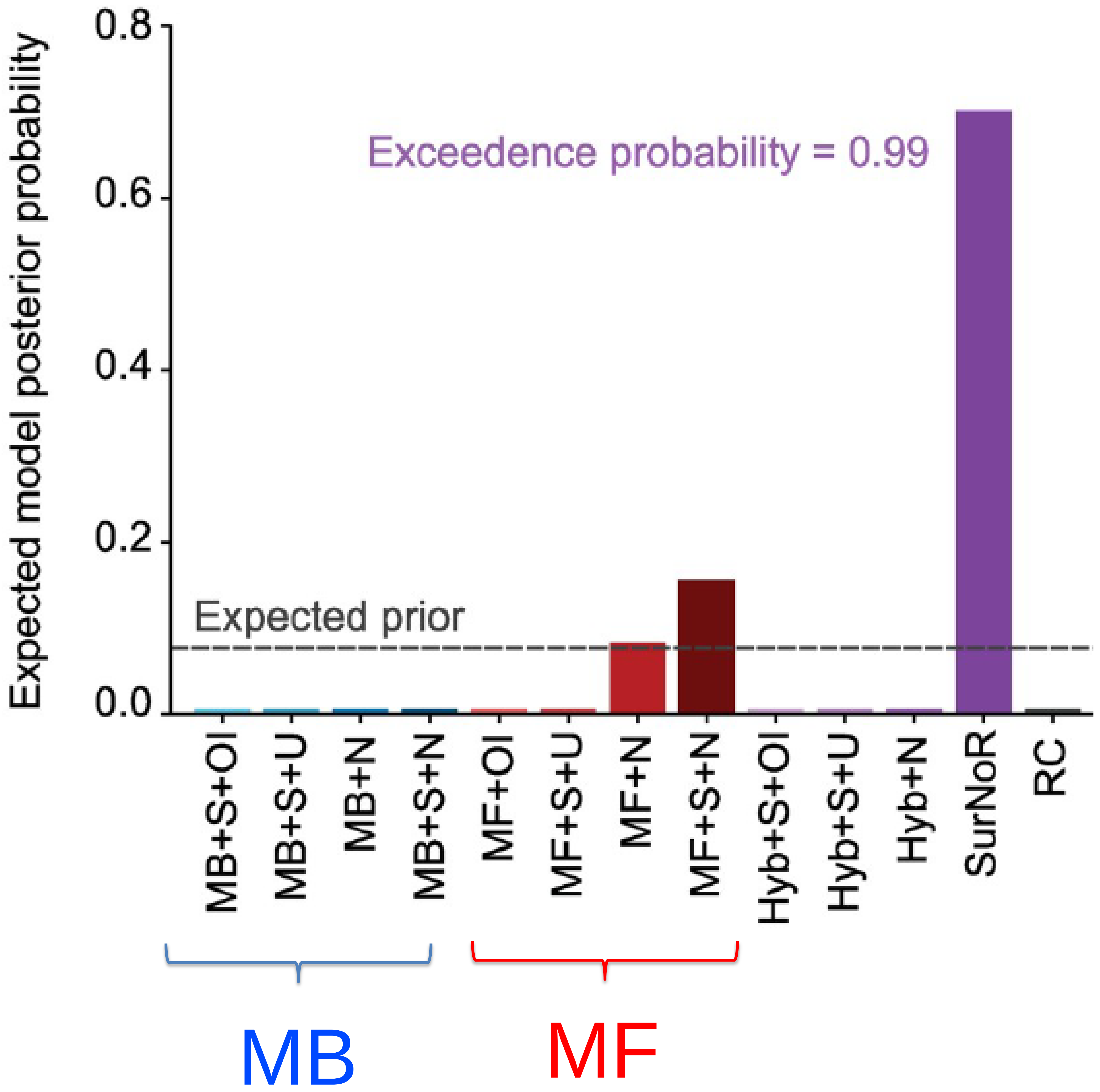
Each one of these, can be implemented as model-free or model-based.

Comparison of Models: Surprise, Novelty, Reward

Finding 4)
Rapid relearning needs surprise



- Turn off novelty
- Turn off surprise
- Turn off model-based →MF
- Turn off model-free →MB
- OI = Optimistic Initialization



Previous slide.

The best model is the combination of Surprise, Novelty and Reward (SuRNoR).

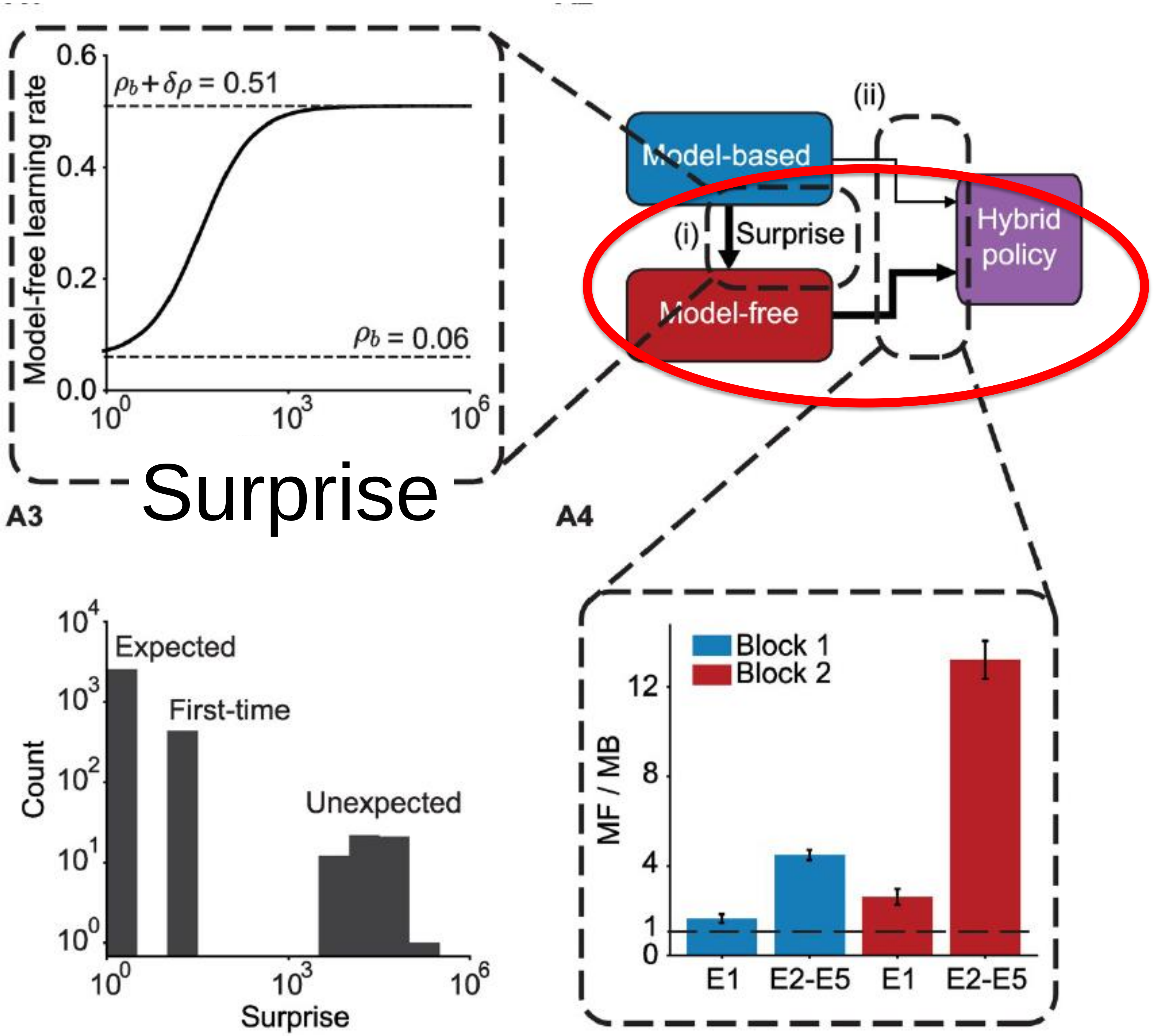
The second best model is model-free (MF) with surprise, novelty, and reward.
Turning of surprise lowers the performance (Hybrid model and surprise).

Model-based compares less well with human data than model-free.

Relative importance of model-based versus model-free

Finding 5)
Model-free dominates
Human behavior!

surprise-modulated learning rate



Previous slide.

One can separately analyze the relative importance of the model-free and the model-based pathway to the hybrid policy in the SuRNoR model.
One finds that model-based never dominates, so that we conclude that human participants are best described by model-free algirthms with surprise.

Surprise is used modulate learning in RL

Finding 6)
Surprise is against expectations.
Hence surprise needs a **world model.**

However, world model is
- Not used to do planning!
- Only used to extract surprise!

World-model not used for planning!

Previous slide.

Surprise needs a world model, but we said that the model-free algorithm better explains the behavior.

The interpretation is that human participants develop a model of the world, but they only use it to detect surprise (change points) which allows them to re-adapt the model.

But they do not use it to plan ahead or do updates of the Bellman equation in the background.

Surprise, World models, and Planning

Finding 6)
World model is available to humans
- But not used to do planning!
- Only used to extract surprise!

For humans:
- Planning is hard (not intuitive/natural)
- Exception: Planning in 2-dim or 3-dim environments
- Planning needs ‘paper and pencil’: “let’s work this out”

Humans are not ‘optimal’. Humans use heuristics.
Heuristics is mostly good for natural tasks.
Markov Decision Problems are ‘not natural’

Previous slide.

Planning is simple for humans in 2-dim or 3-dim environments.

But not for Markov Decision Problems.

Abstract problems require (for most humans) a slow process of math-like solution process: whenever you feel, it would be easier to work something out with paper or pencil, you try to use a ‘world model’ that is non-intuitive for humans.

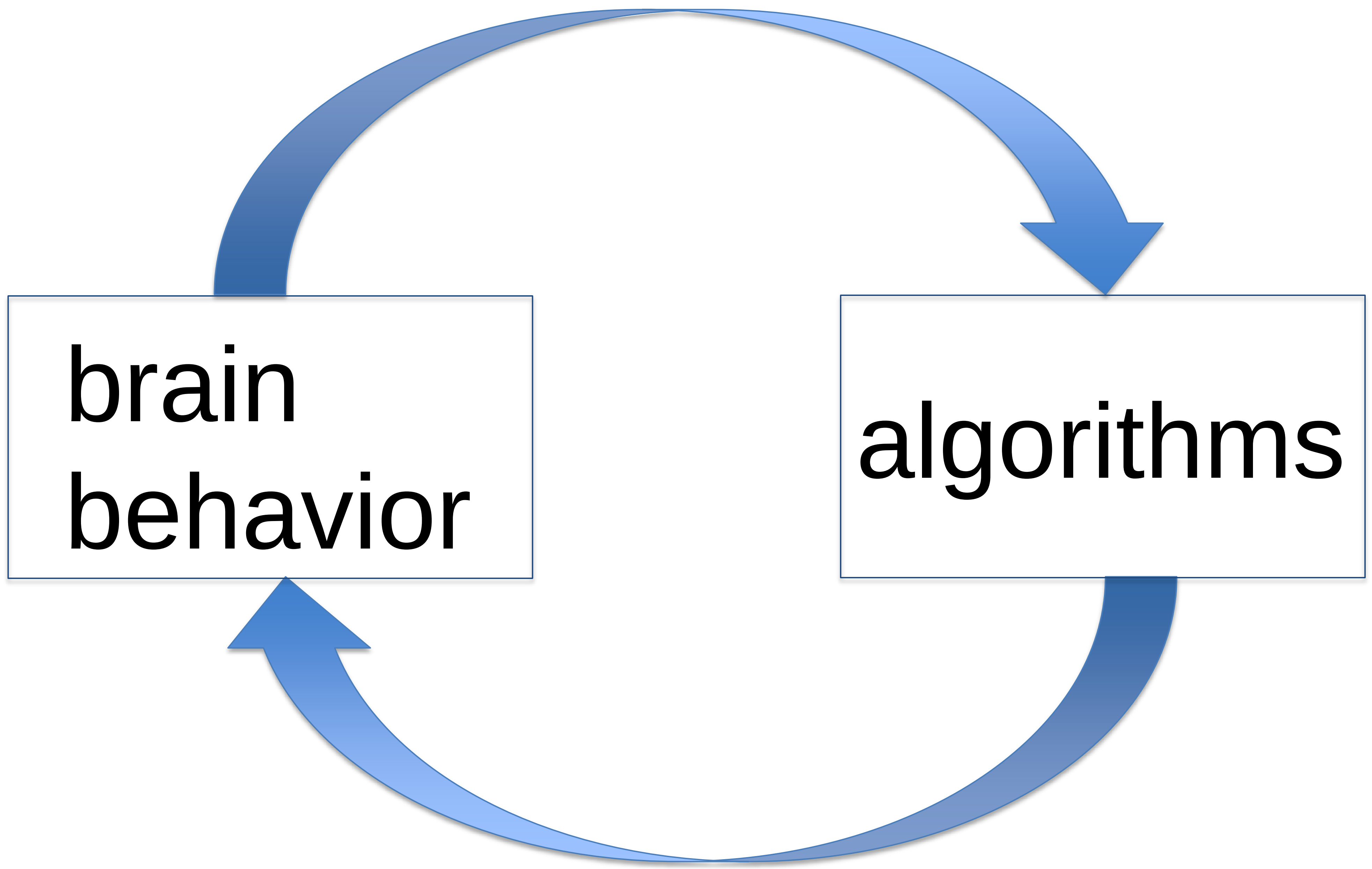
Reward-based learning versus Surprise-based learning

Reward-based learning		Surprise-based learning
Reward-Prediction Error	→	Surprise
defined as TD error	→	defined as Bayes Factor Surprise
stimulated by chocolate, money, praise, …	→	stimulated by observations not consistent with momentary model of environment
modulates learning rate	→	modulates learning rate

Previous slide.

Summary: Comparison of Reward Prediction Error and Surprise.

Current Research in Reinforcement Learning:



- Exploration → not exploration bonus, but separate modules
- Novelty → Novelty supports exploration
- Surprise → Surprise detects changes/adapts learning

Previous slide. Review from previous lectures.

RL has two roots: optimization for Markov Decision Problems and Brain sciences/psychology

The interaction has not stopped. Modern RL still takes up influences from Brain Sciences. Examples are the role of novelty, surprise, and their roles for exploration and in volatile environments.

In-depth course evaluation!

Please take time NOW!!!

Everybody!!!! - EPFL needs a high return rate.

- Agepoly wants a high return rate

*No need to write **comments**. However, if you do:*

→ *mention whether you mainly attend in-class/follow videos*

The END

... of today.

We talk about exam procedures next week.

You can prepare questions as well!

Previous slide: