Lecture 12
Actor-critic models

Jochen Braun

Otto-von-Guericke-Universität Magdeburg,
Cognitive Biology Group

Theoretical Neuroscience II, SS 2020

Credits:

Dayan & Abbott (2001) "Theoretical Neuroscience", Chapter 9

Foster, Morris & Dayan (2000) Hippocampus 10: 1–16.

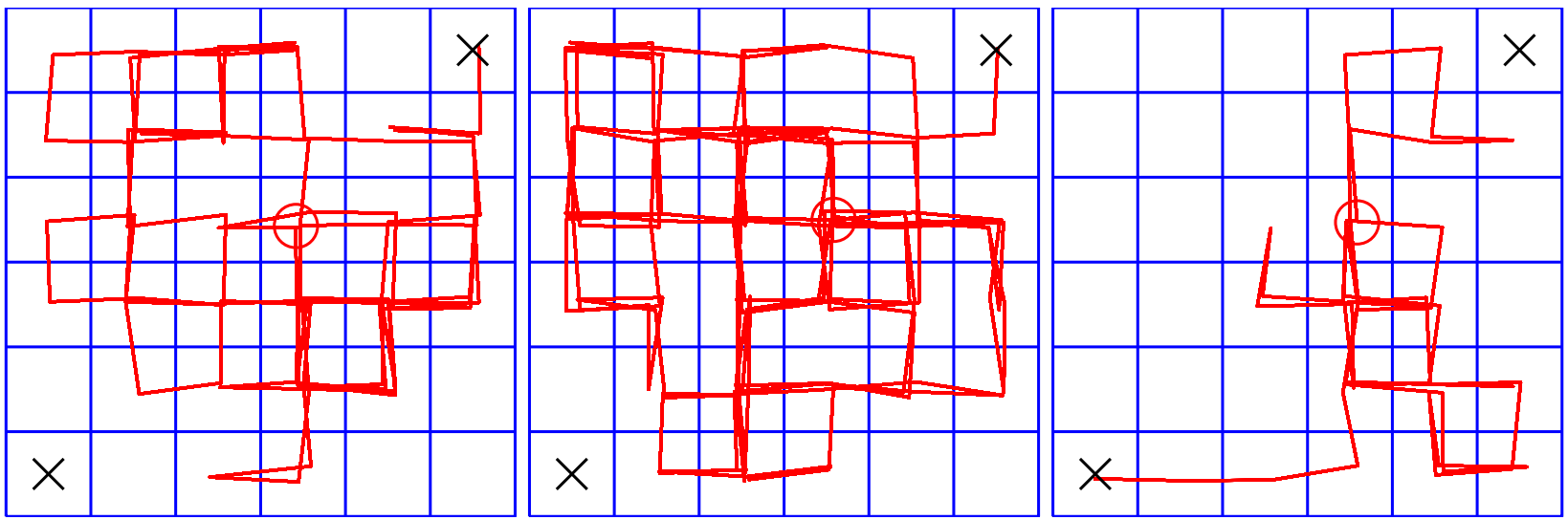
12. Actor critic models

Maze task *is classic paradigm where sequential action is required to obtain delayed reward. We use maze task to review concepts (policy improvement, evaluation & iteration) and to gain intuitive insight into the operation of actor-critic models.* **Water maze** *is common paradigm for hippocampus-dependent learning by rats. We discuss classic publication Foster et al. (2000). In first model version, training of 'critic' (value map) and 'actor' (local direction preferences) proves slow and inflexible. In a new environment, the model must unlearn old environment before learning new one. In a second model version, 'actor' is complemented with a global direction preference. This lets model transfer learning between environments and improves correspondence to observation. Conclude with* **summary of reinforcement**.

Outline

0 Recap

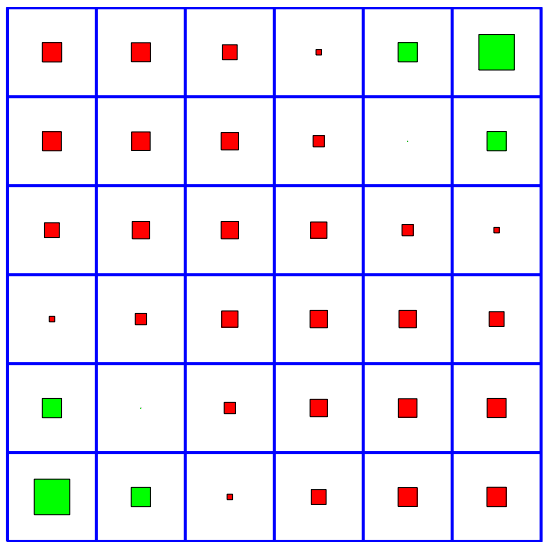
In the last lectures, we have considered a "gridworld" of $N \times N$ states, in which an agent moves randomly until it accidentally finds one of two exits:



Policy evaluation

As our target of learning, we introduced the *value function*, namely, the total reward expected after a state s, discounted over all future steps:

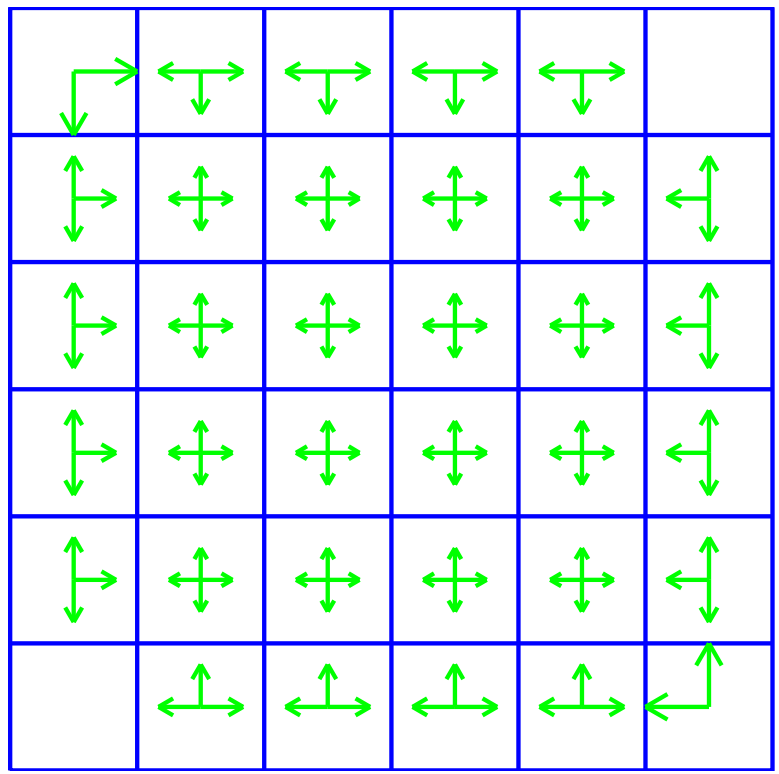
$$V_\pi(s) = \sum_a \pi(a,s) \left[r(a,s) + \gamma V_\pi(s') \right], \qquad s \xrightarrow{a} s'$$



Policy improvment

To translate learned information into improved behavior, we further introduced a *policy* π, namely, a probabilistic mapping from each state s to all available actions a:

$$\pi(a, s)$$



TD learning

We concurrently improved both value function and policy, as follows:

Action and state transition:

$$a: \qquad s \rightarrow s'$$

TD prediction error:

$$\delta_a = r_{s'} + \gamma V_{s'} - V_s$$

Policy evaluation:

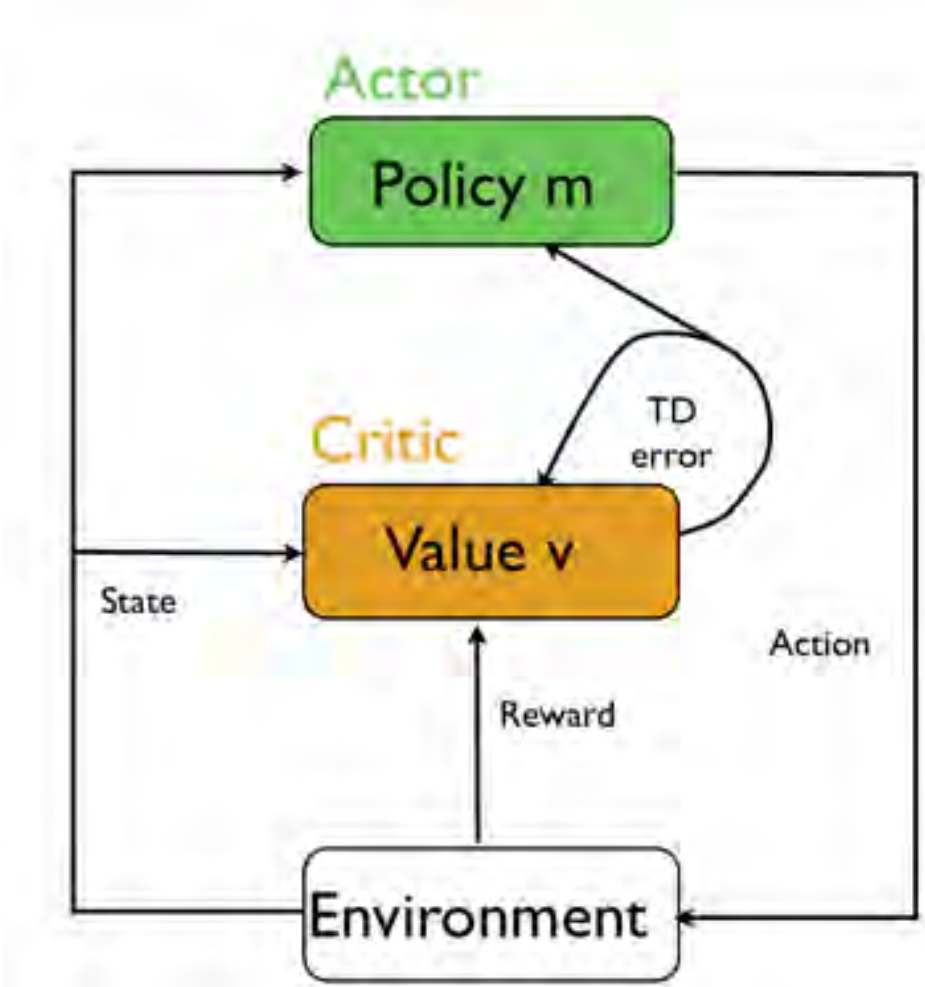
$$V_s \rightarrow V_s + \alpha \, \delta_a$$

Policy improvement:

$$m_a \rightarrow m_a + \epsilon \, \delta_a$$

Actor-critic model

- Policy $\pi(a, s)$ is the *actor*, as it sets action preferences.
- Value function $V_\pi(s)$ is the *critic*, as it evaluates total reward expected from policy.



1 Sequential action choice

Seeking to generalize this approach, we consider a classic situation involving sequential actions and delayed rewards, a *maze task*:

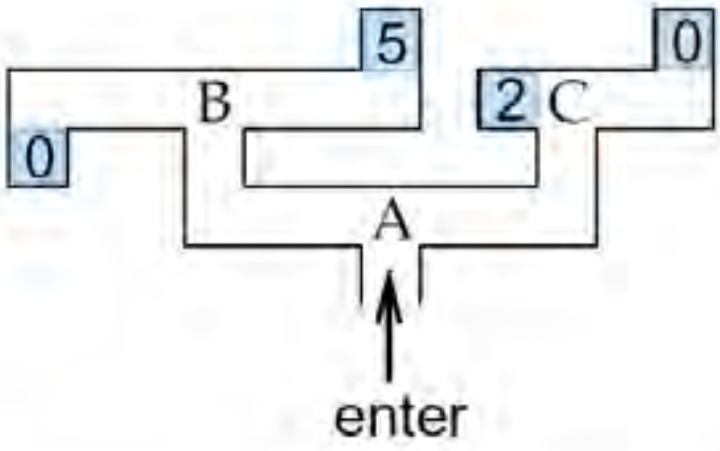
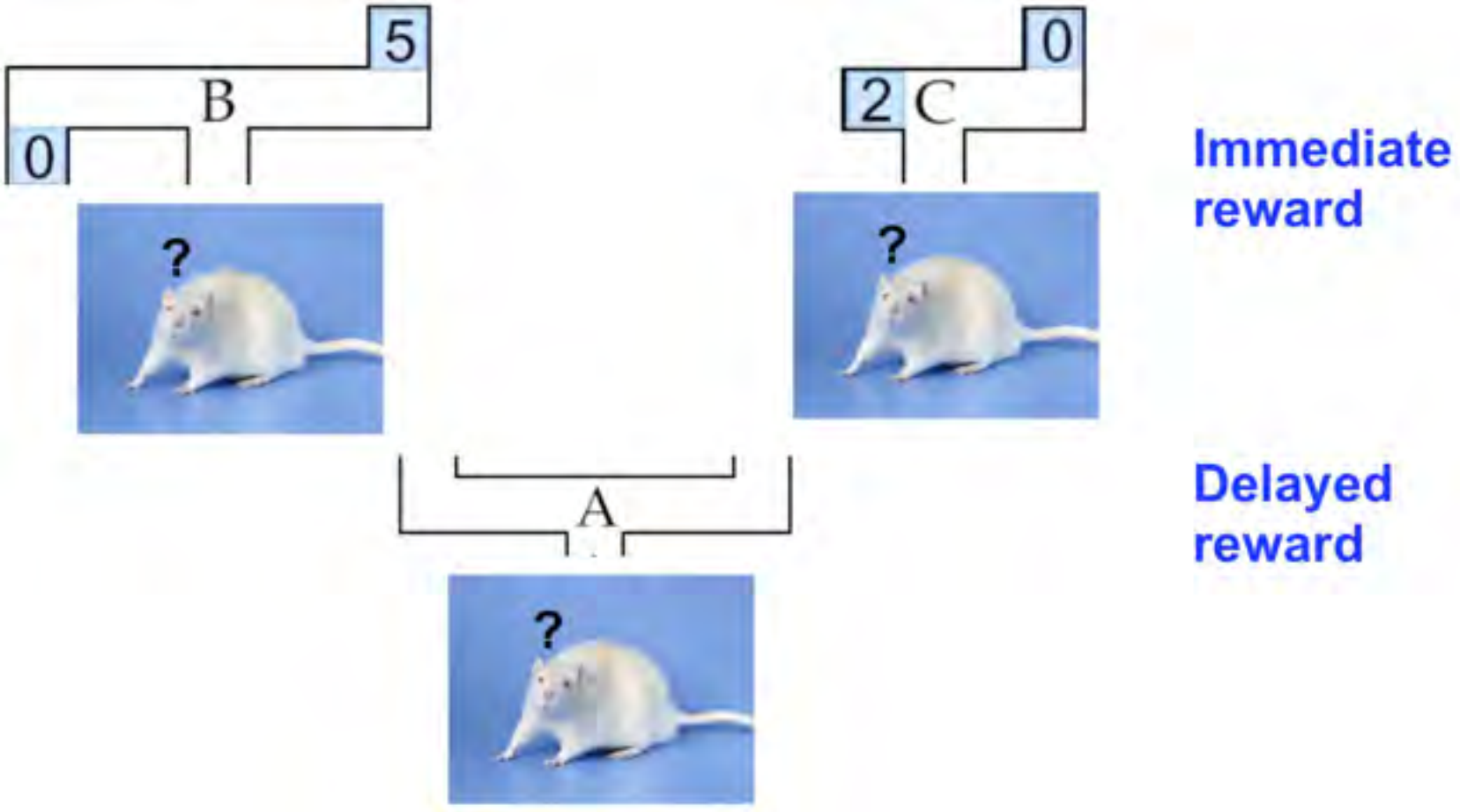


Figure 9.7: The maze task. The rat enters the maze from the bottom and has to move forward. Upon reaching one of the end points (the shaded boxes), it receives the number of food pellets indicated and the trial ends. Decision points are A, B, and C.



Note that neither choice at A is rewarded immediately. If the rat goes *left* and *right*, both first and second choice are correct. However, reinforcement of the first choice is delayed until the second choice is also correct.

General approach

As before, our model for reinforcement learning in this task will include the following components:

- **Actor:** Maintains policy and selects actions (given current evaluation).
- **Policy improvement:** Some mechanism for gradual improvement of policy.
- **Critic:** Maintains estimates of expected total reward (given current policy).
- **Policy evaluation:** Some mechanism for gradual improvement of reward estimates.

For didactic reasons, we consider and compare two approaches:

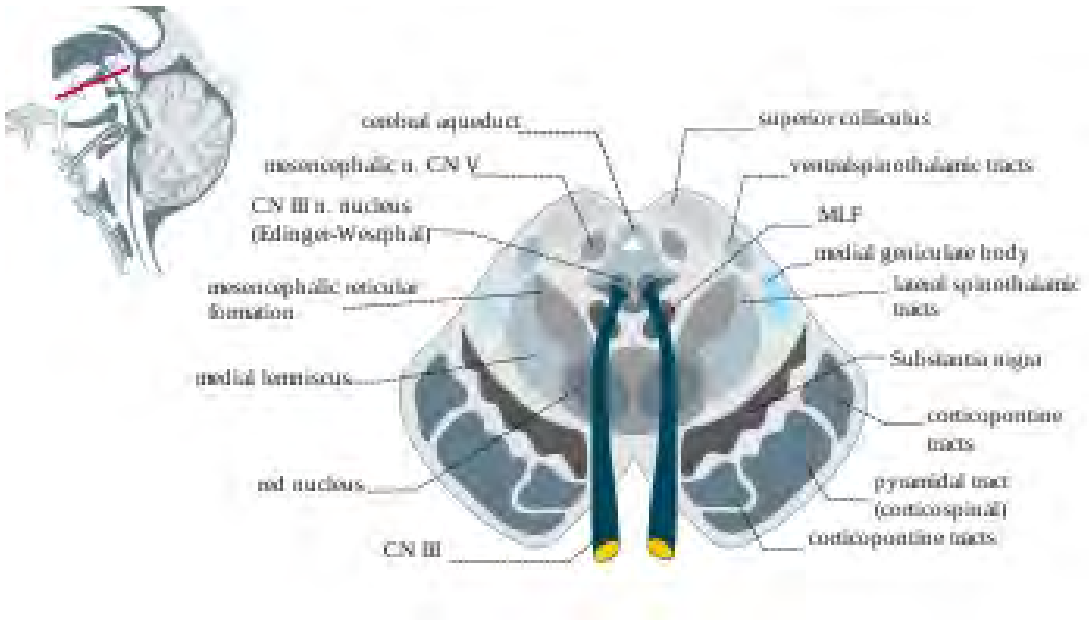
- **Slow policy iteration:** Alternating slowly between policy improvement and reward evaluation (session by session).

- **Fast policy iteration:** Alternating quickly between policy improvement and reward evaluation (trial by trial).

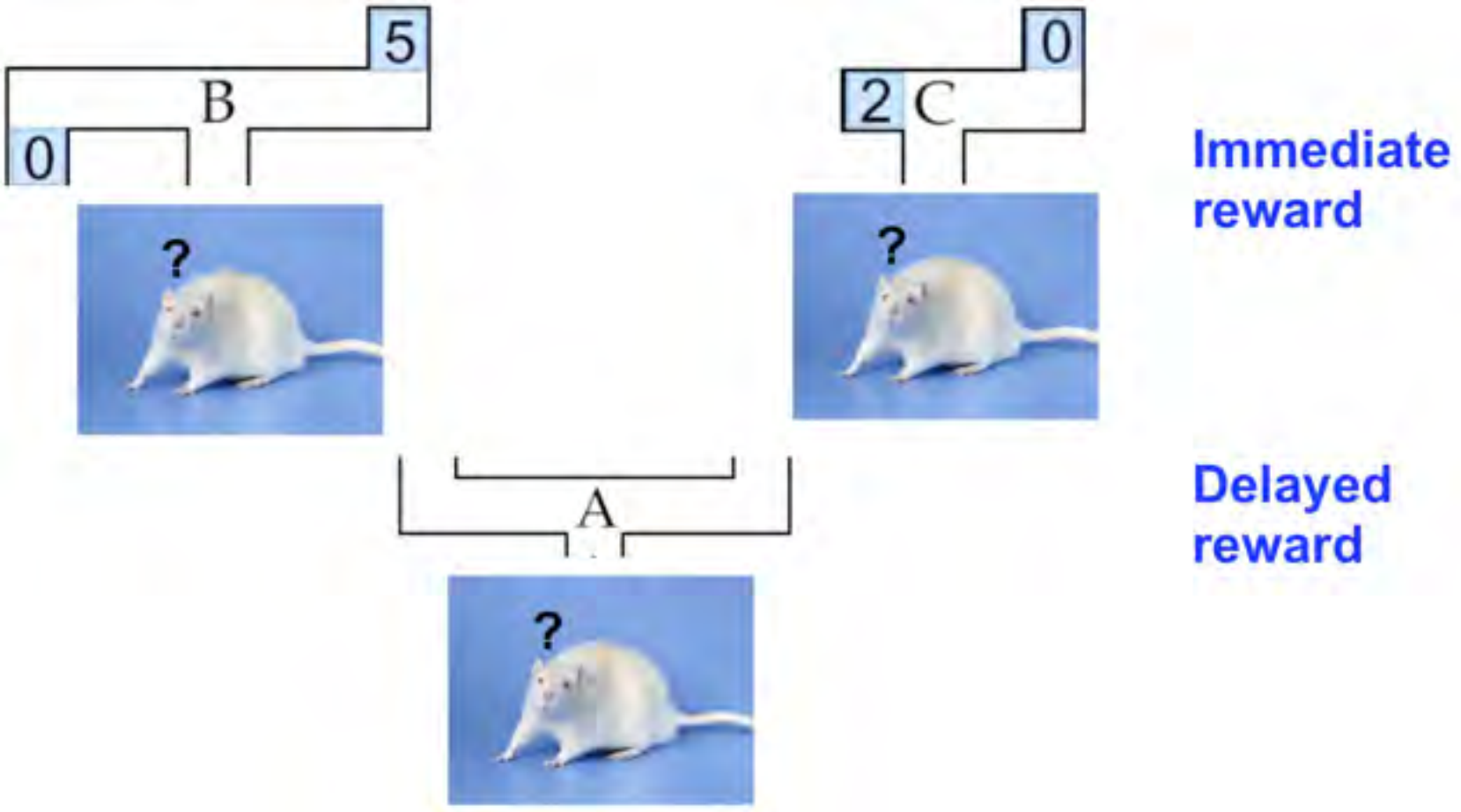
TD prediction error

Reinforcement of both policy and reward are driven by the TD prediction error δ.

The dorsal striatum is involved in the selection and sequencing of actions. Terminals of axons projecting from the substatial nigra pars compacta (SNPC) release dopamine onto synapses within the striatum, consistent with such a role. Place learning strategies are impaired in SNPC lesioned rats (Da Cunha, Silva, Wietikoski, et al., 2006).



2 AC for small maze



Actor: stochastic policy

Initially, we set action values $m_a(s)$ to zero, so that choice probabilities become 50% (random policy):

Action values:

$$m_{L,R}(A) = \{0, 0\}, \qquad m_{L,R}(B) = \{0, 0\}, \qquad m_{L,R}(C) = \{0, 0\}$$

Choice probabilities:

$$P_L(u) = \frac{1}{1 + e^{\beta\,[m_R(u) - m_L(u)]}} = 0.5$$

$$P_R(u) = \frac{1}{1 + e^{\beta\,[m_L(u) - m_R(u)]}} = 0.5$$

Critic: policy evaluation

Given a policy, the total future reward at each position A, B, or C can be learnt with the temporal difference rule. If the rat chooses to move from x to y, we modify the expected rewards $v(x)$ as follows:

$$v(x) \rightarrow v(x) + \epsilon\,\delta \qquad\qquad \delta(x) = r(x) + v(y) - v(x)$$

where $\delta(x)$ is the *TD prediction error*, ϵ is a learning rate, and $r(x)$ is the immediate reward, if any. In addition, the prediction error includes the *change of 'hopes'*, i.e. of total expected rewards, between $v(x)$ and $v(y)$.

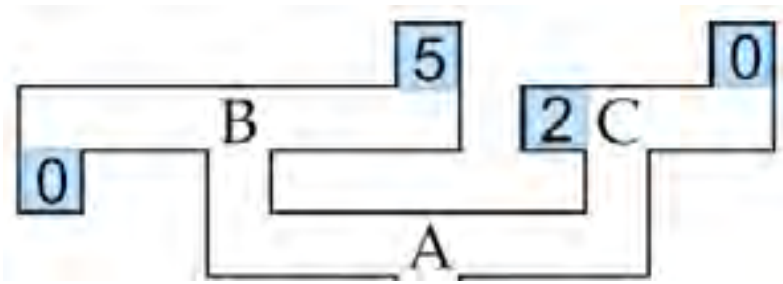
In this way, any rewards expected at y, but not yet at x, are gradually shifted back to x, until they are also expected at x.

Given a *random policy*, inspection of the maze shows the *average* expected future reward to be

$$v(B) = \frac{r_L(B) + r_R(B)}{2} = \frac{0+5}{2} = 2.5$$

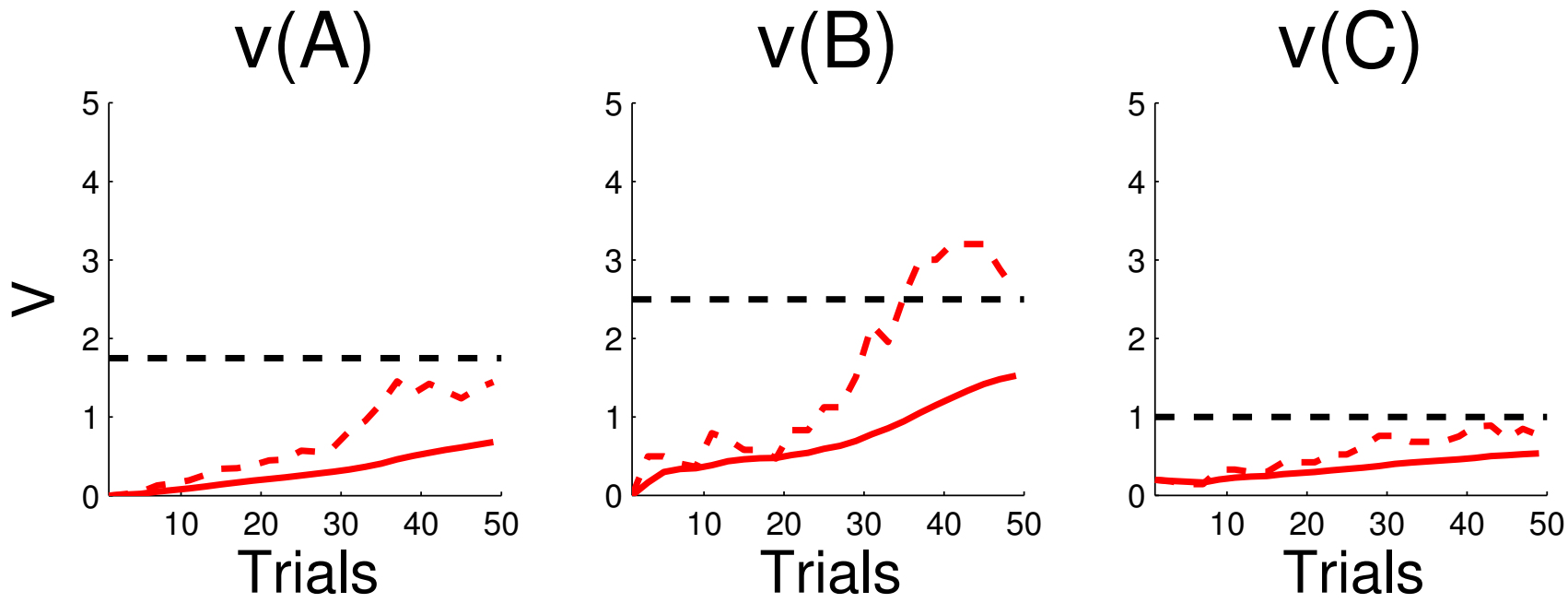
$$v(C) = \frac{r_L(C) + r_R(C)}{2} = \frac{0+2}{2} = 1$$

$$v(A) = \frac{v(B) + v(C)}{2} = 1.75$$



Policy evaluation only

Given random policy, the learned values approach theoretical predictions.



Dashed: trial average. Solid: cumulative average. Dashed black: theoretical prediction for random policy.

Direct actor: policy improvement

Given a TD prediction error (i.e., expected total reward), the policy (action values) at each position A, B, or C are improved *directly*. After turning from x either left or right to y, we adjust the action values $m_{L,R}(x)$ as follows:

Left turn:

$$\begin{aligned} m_L(x) &\rightarrow m_L(x) + \epsilon\,[1 - P_L(x)]\,\delta \\ m_R(x) &\rightarrow m_R(x) - \epsilon\,P_R(x)\,\delta \end{aligned}$$

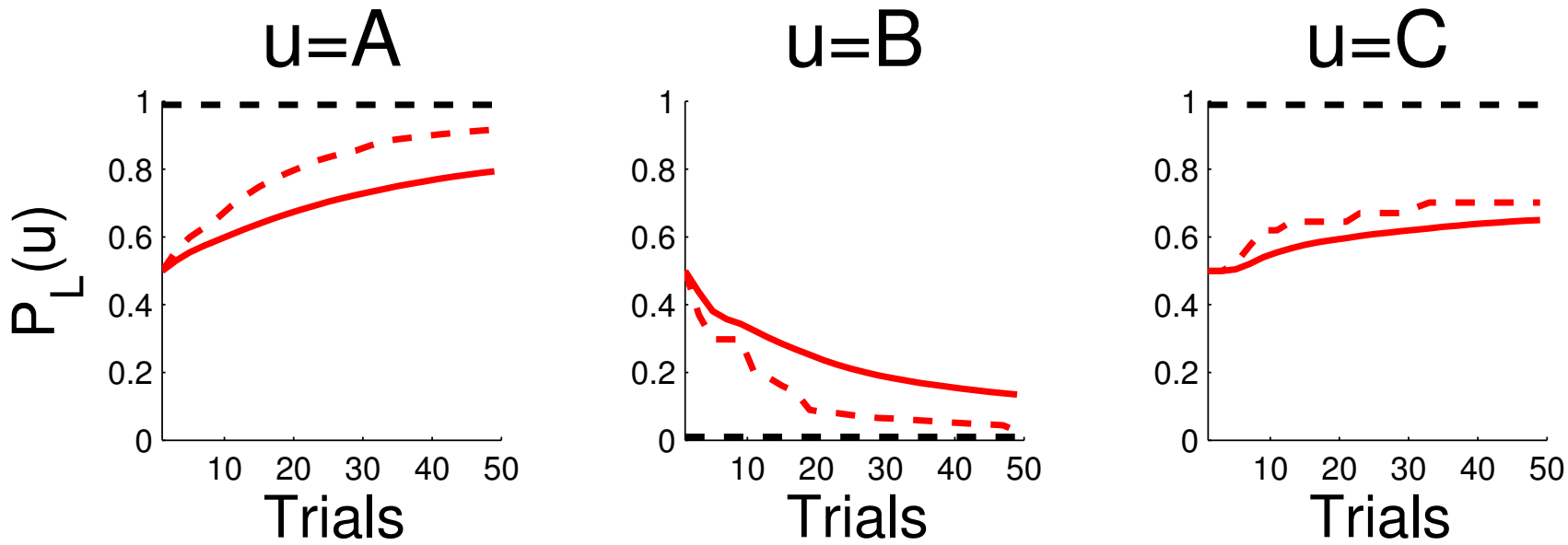
Right turn:

$$\begin{aligned} m_L(x) &\rightarrow m_L(x) - \epsilon\,P_L(x)\,\delta \\ m_R(x) &\rightarrow m_R(x) + \epsilon\,[1 - P_R(x)]\,\delta \end{aligned}$$

$$\delta = r(y) + v(y) - v(x)$$

Policy improvement only (first slow iteration)

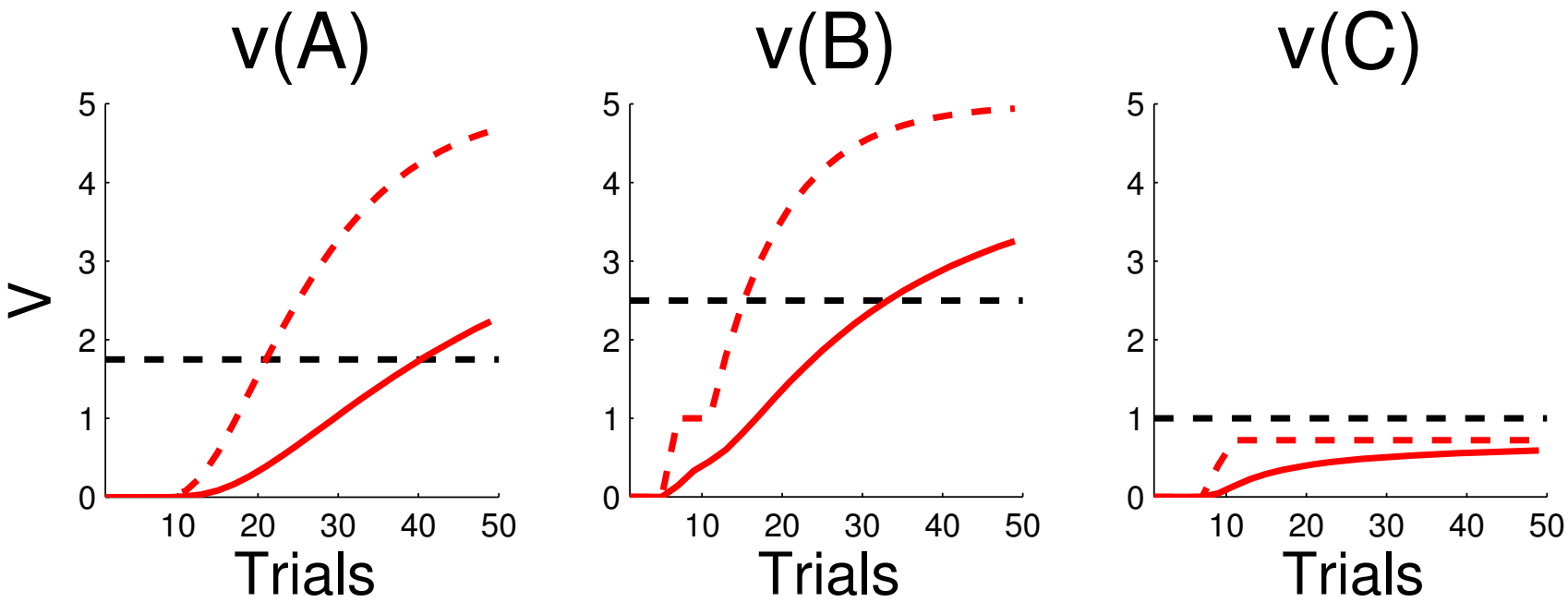
Based on *values v* learnt in the earlier evaluation phase, we can now improve policy (action preferences). Model begins to prefer the right choices (turns):



Dashed: trial average. Solid: cumulative average. Dashed black: ideal choice probability.

Fast policy iteration (trial-by-trial)

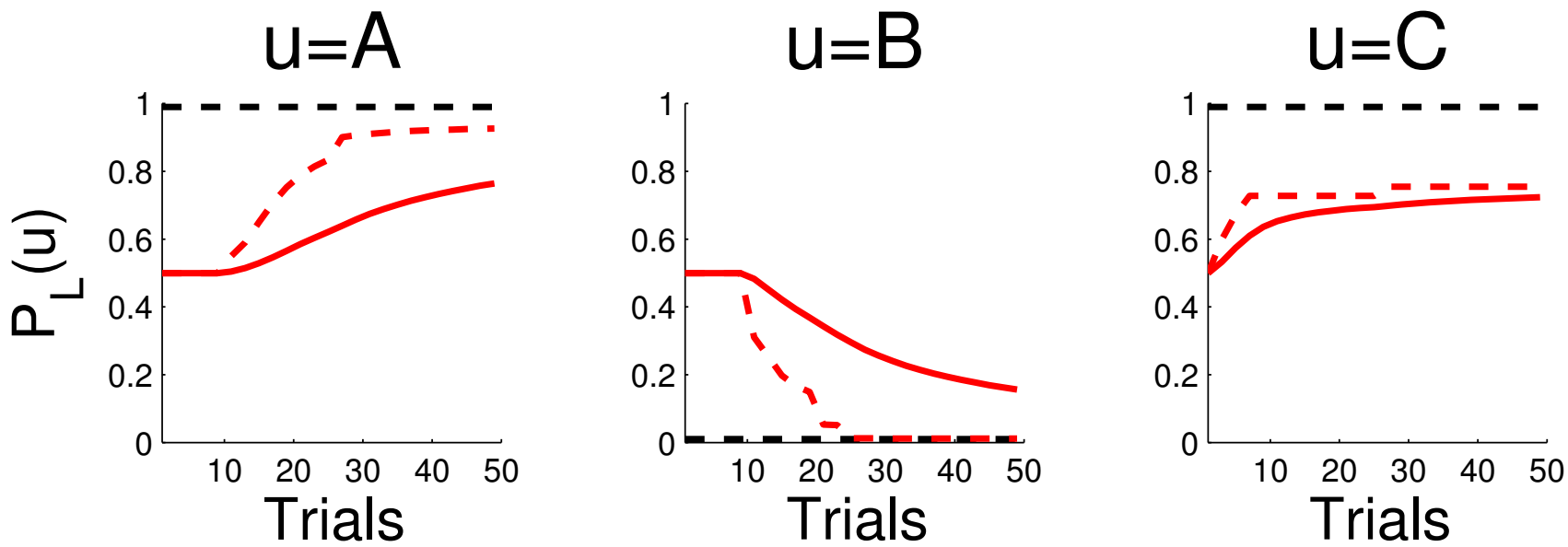
Now that policy has improved, the values exceed theoretical predictions for *random policy*. Given the right policy, total expected value at A and B becomes maximal.



Dashed: trial average. Solid: cumulative average. Dashed black: theoretical values for *random policy*.

ctd.

Quasi-deterministic choice at A retards further learning at C.



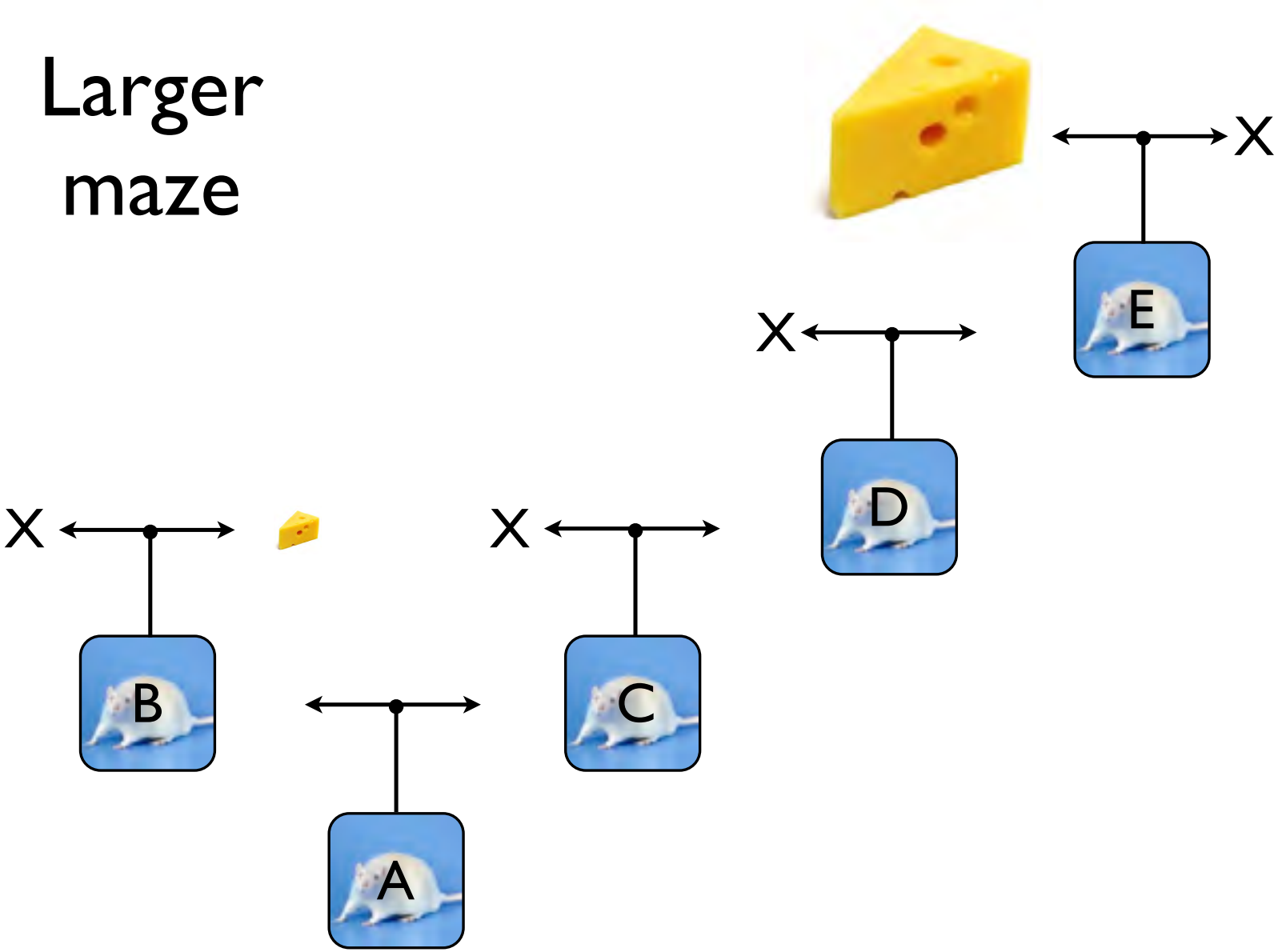
Dashed: trial average. Solid: cumulative average. Dashed black: ideal choice probabilities.

Points to note

- Using the TD prediction error, we have implemented a *direct actor* with slow and fast *policy improvement*.
- In the absence of rewards, we adopt a random policy.
- When rewards are reached and slowly become expected, this expectation gradually becomes associated with earlier and earlier steps.
- Eventually, rewards are anticipated already at the first step.
- In general, as policy improves, values functions increase, and *vice versa*.

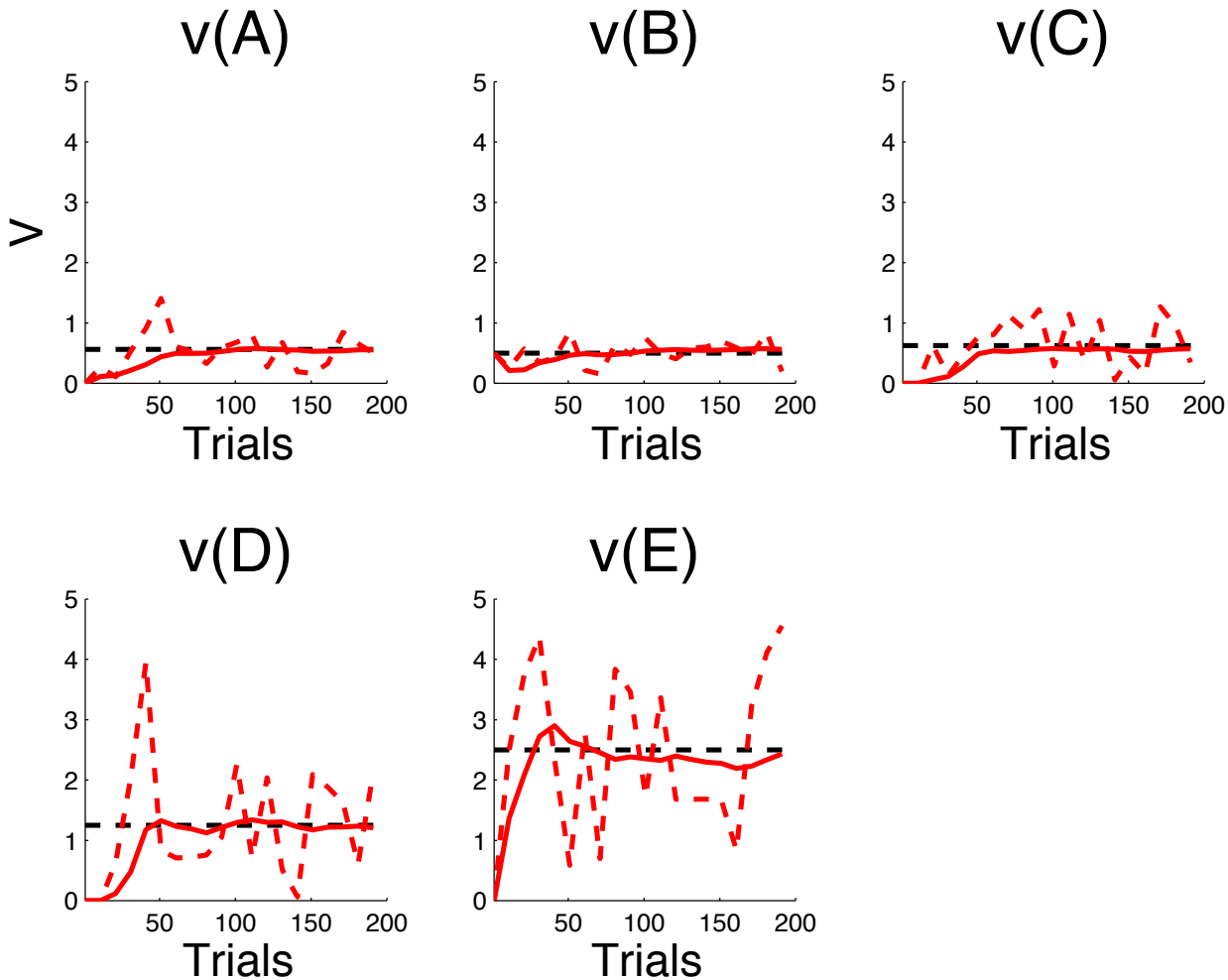
3 AC for larger maze

Larger maze



Policy evaluation only, first round

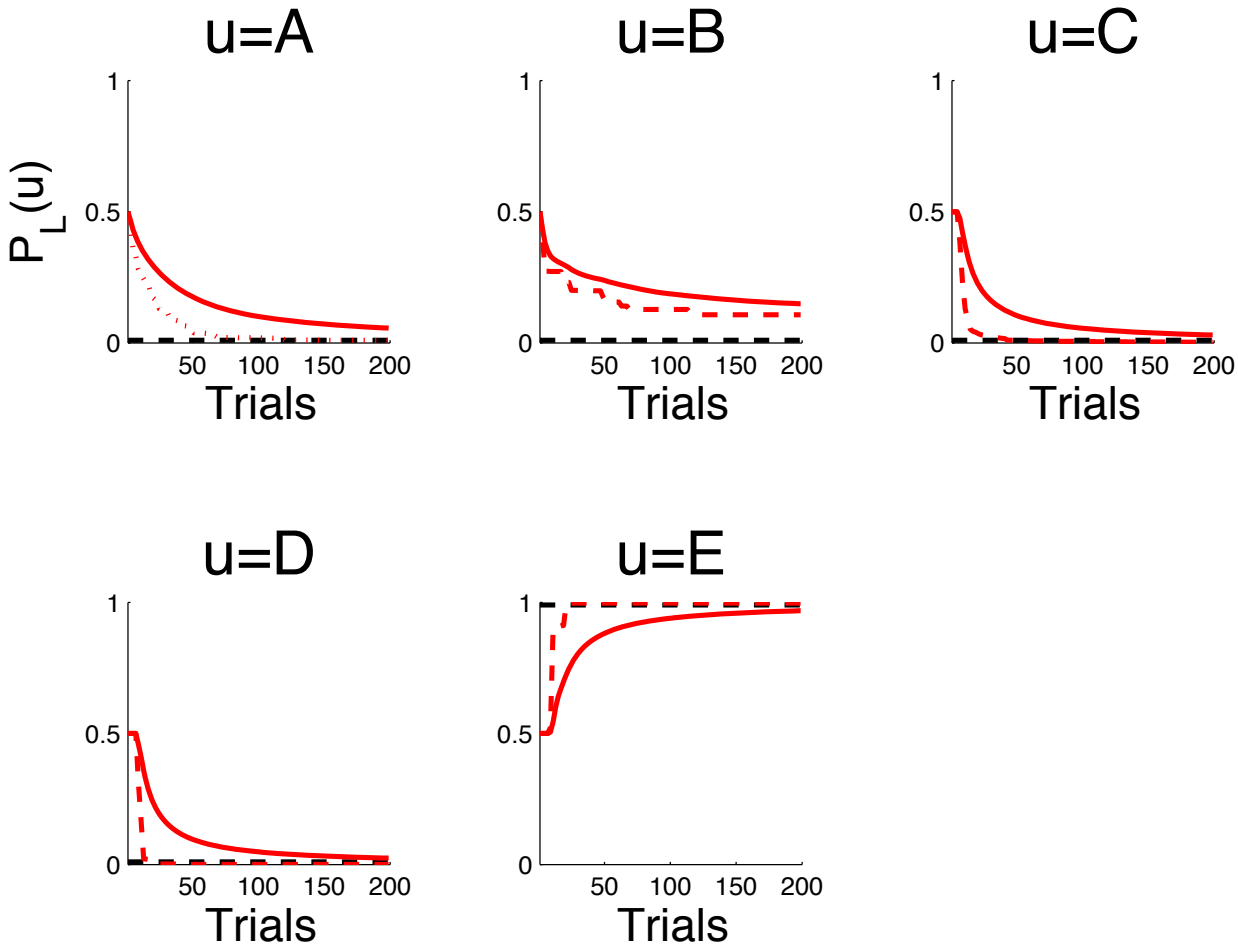
Values based on *random policy*.



Dashed: trial average. Solid: cumulative average. Dashed black: theoretical prediction for *random policy*.

Policy improvement only, first round

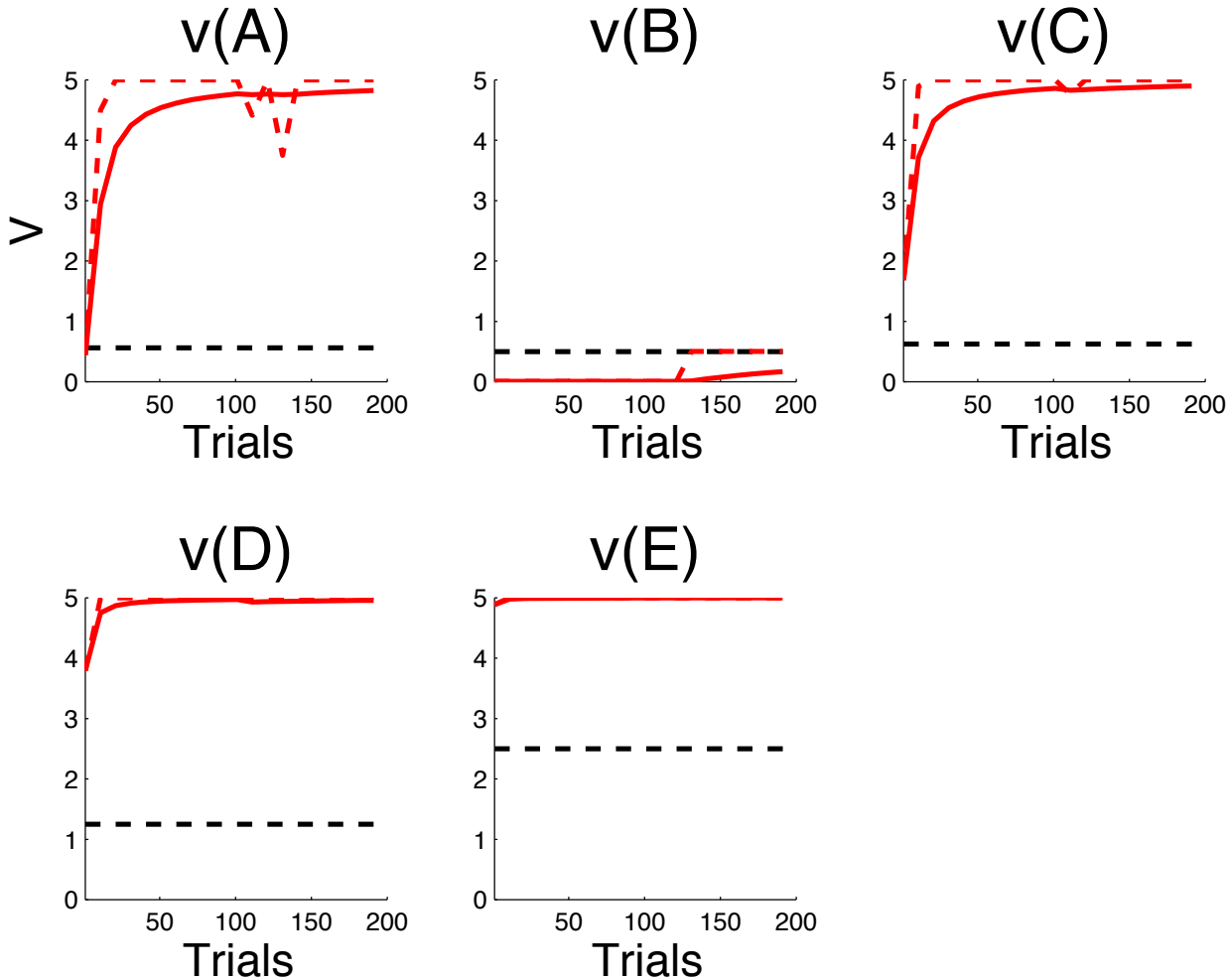
Based on *values* v from earlier evaluation phase.



Dashed: trial average. Solid: cumulative average. Dashed black: ideal choice probability.

Policy evaluation only, second round

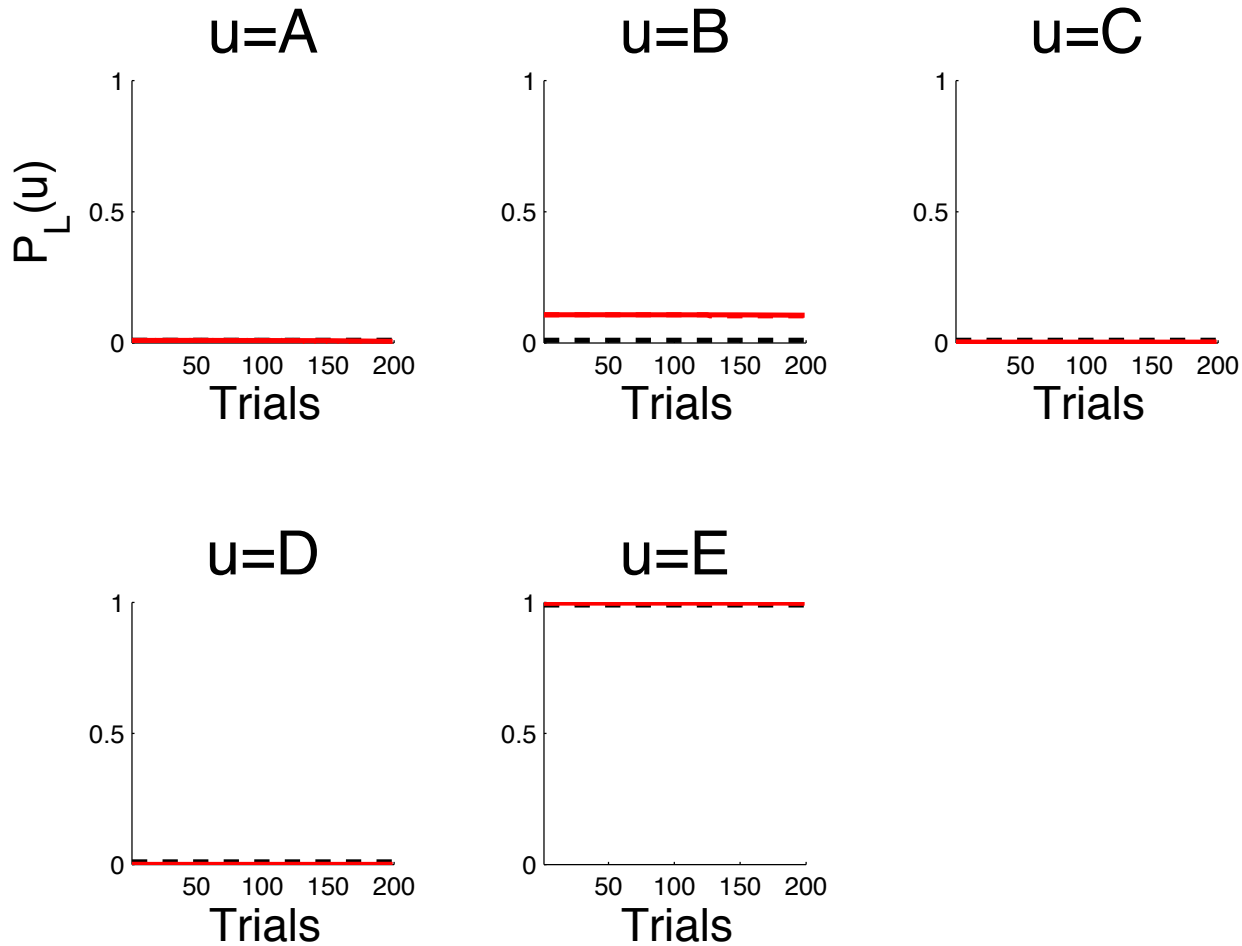
With improving policy, values exceed the theoretical predictions for *random policy*.



Dashed: trial average. Solid: cumulative average. Dashed black: theoretical predictions for *random policy*.

Policy improvement only, second round

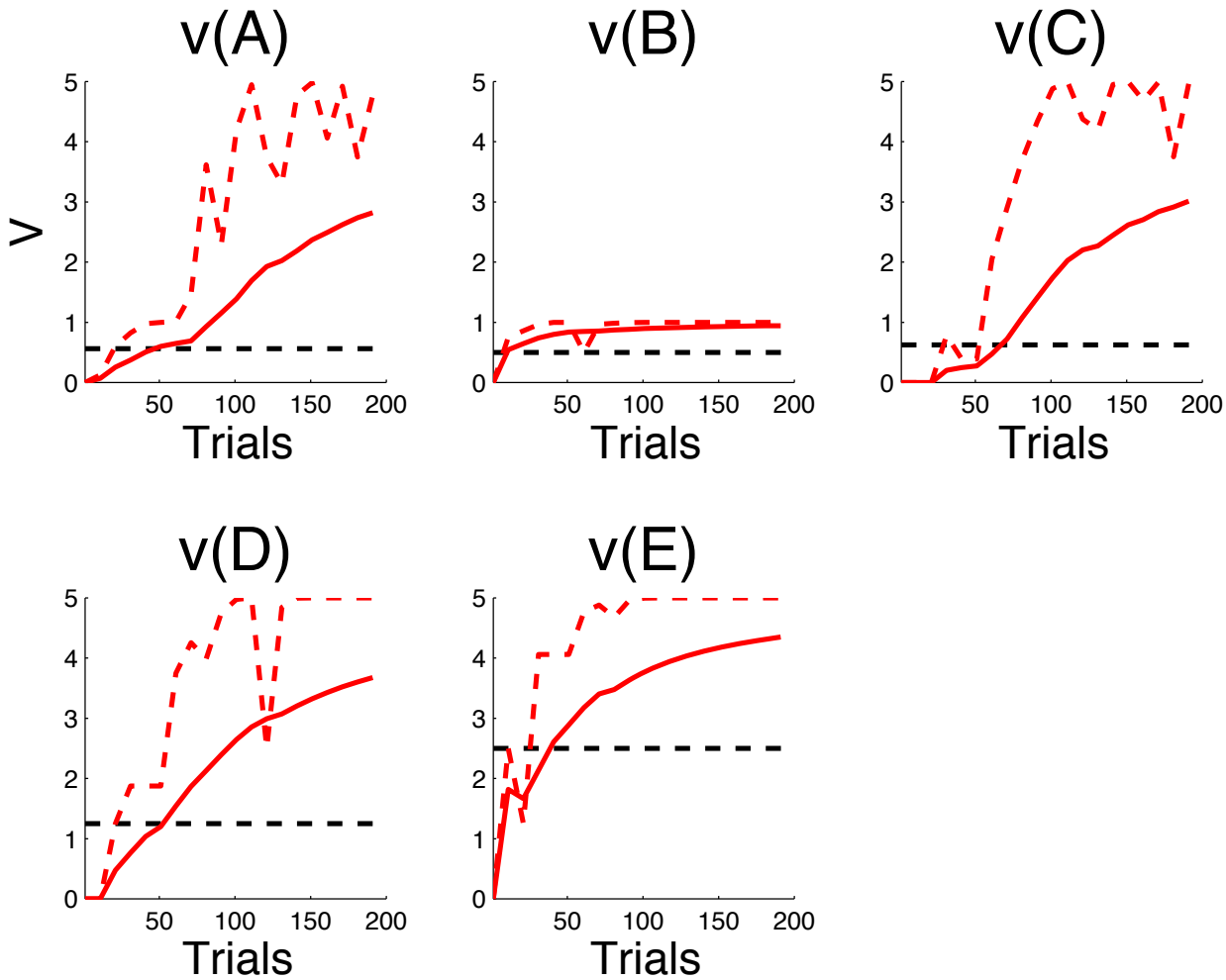
Retaining values from preceding evaluation phase.



Dashed: trial average. Solid: cumulative average. Dashed black: ideal choice probability.

Fast policy iteration (trial-by-trial)

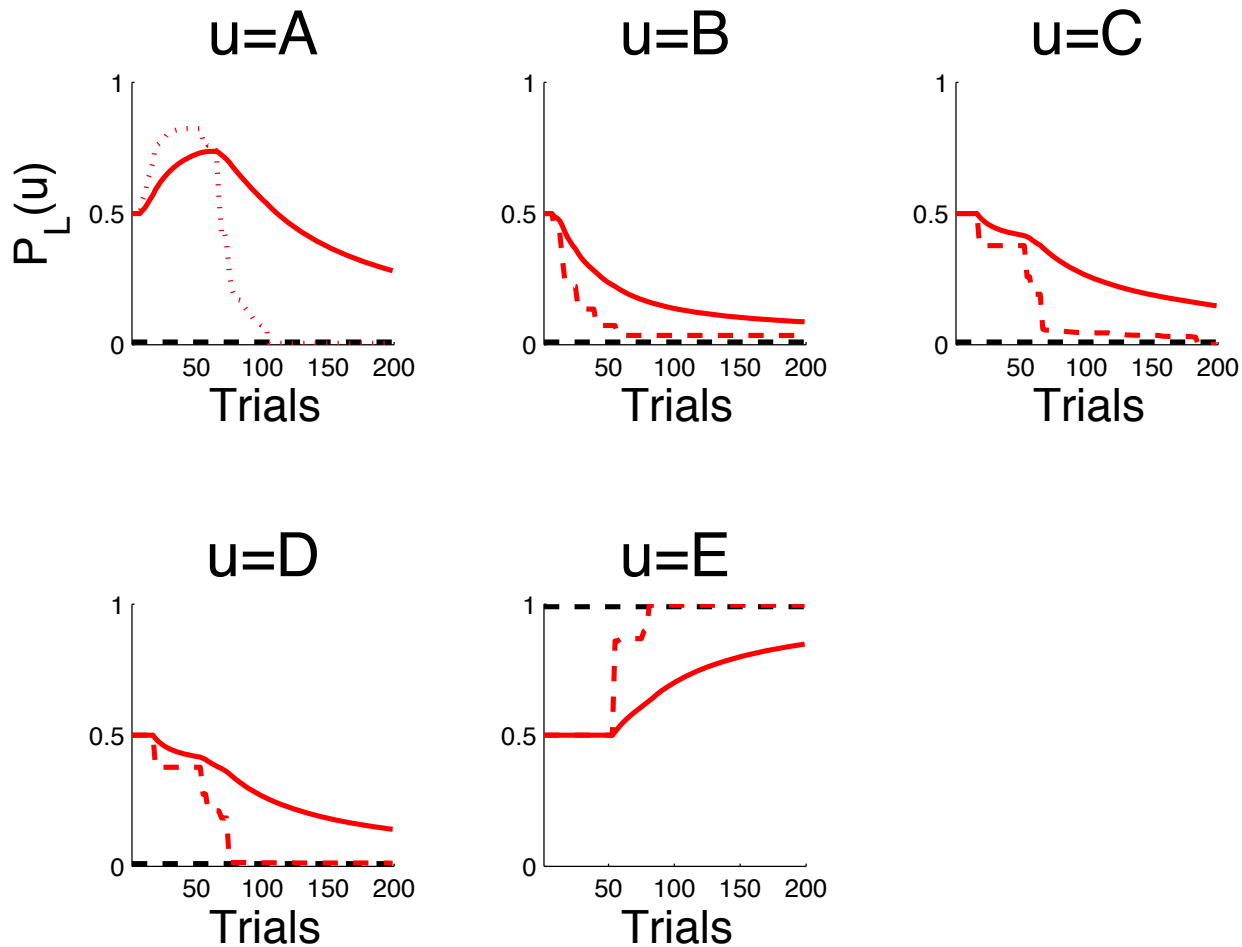
Compare learning rates for values at A, C, D, E!



Dashed: trial average. Solid: cumulative average. Dashed black: theoretical predictions for *random policy*.

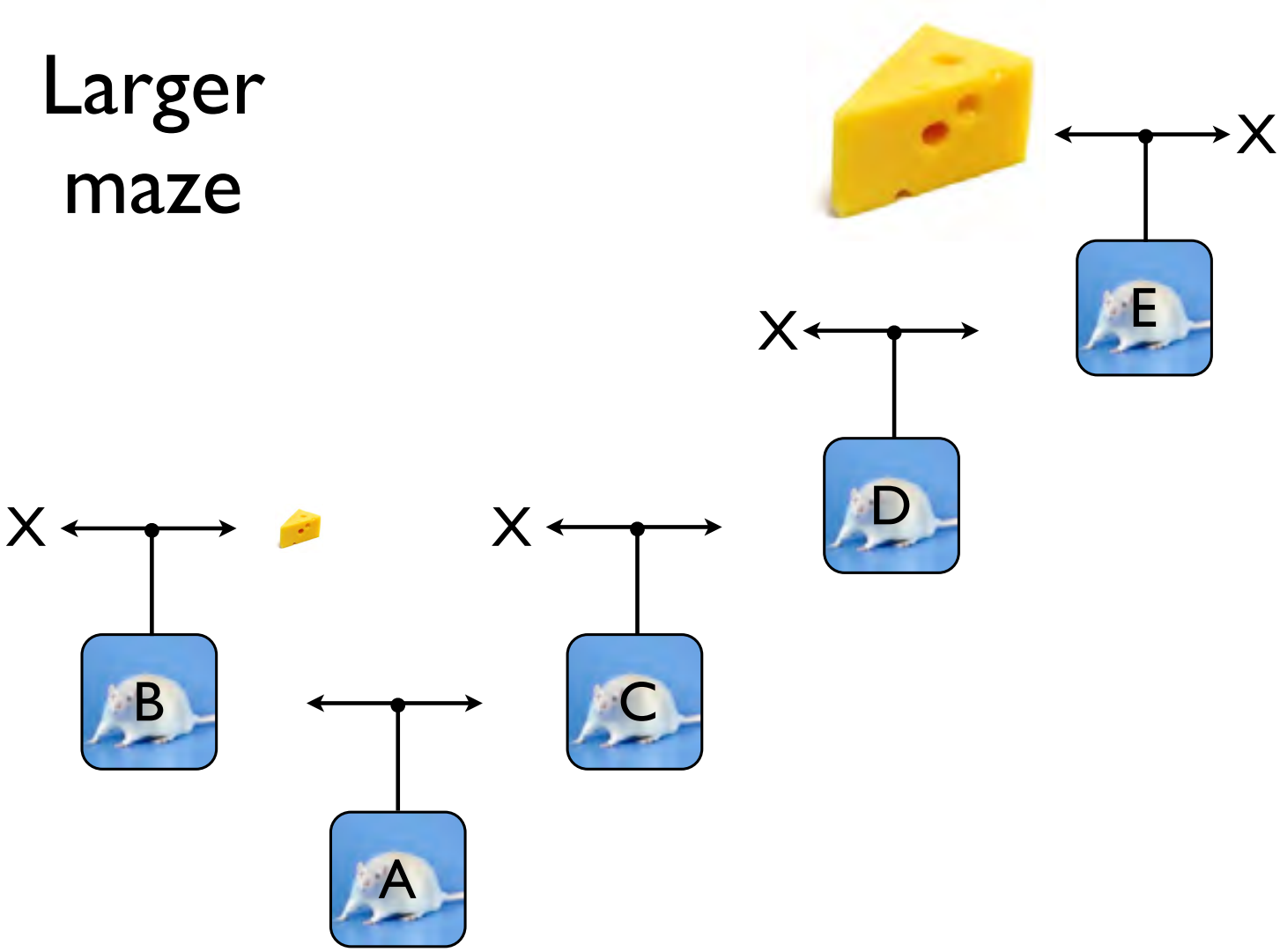
Fast policy iteration (trial-by-trial)

Consider evolution of choice A over time! Why does preferred choice switch from left to right? Because total expected reward after right turn comes to include large reward at E!



Dashed: trial average. Solid: cumulative average. Dashed black: ideal choice probabilities.

Larger maze



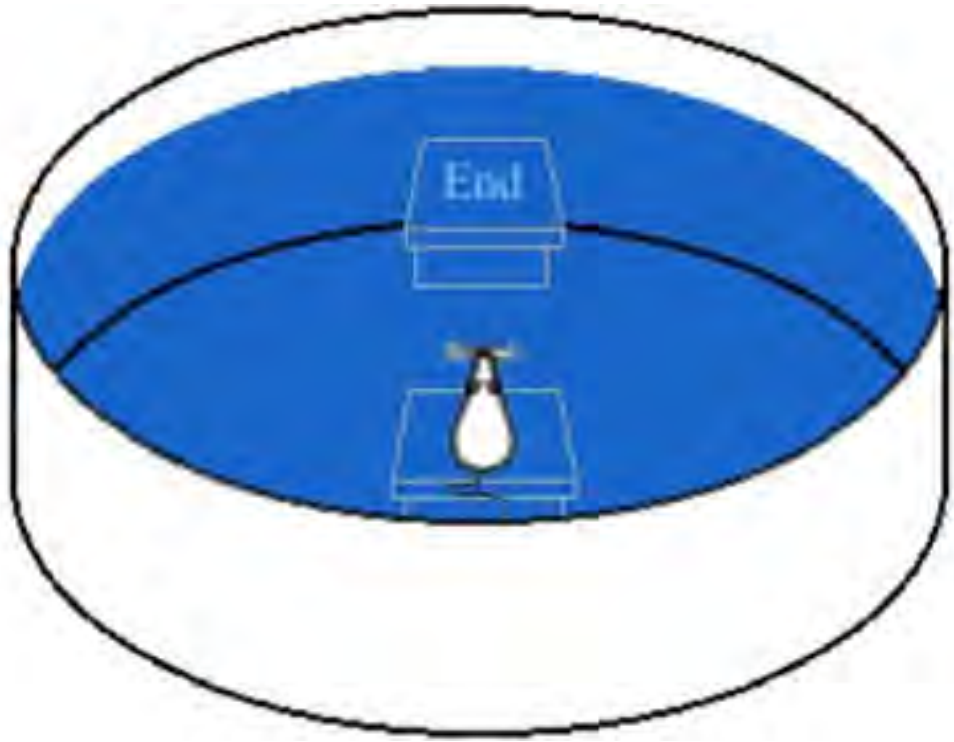
Points to note

- Using the TD prediction error, we have implemented a *direct actor* with slow and fast *policy improvement*.
- Reward expectations are gradually passed form later states back to earlier states.
- As more information about delayed rewards becomes available, earlier choices may change.
- The optimal policy/value combination may be reached, provided that enough trials are available.

4 Water maze

Rats are placed repeatedly in milky water and swim to find a small submerged platform. Eventually, the animals learn to swim directly to the platform.

Foster, Morris, Dayan (2000) Hippocampus: "Models of hippocampally dependent navigation, using the temporal difference rule"



Experimental results

Place: 20 trials with platform at same location, further 4 trials at different location.

Random: Platform position varied from trial to trial.

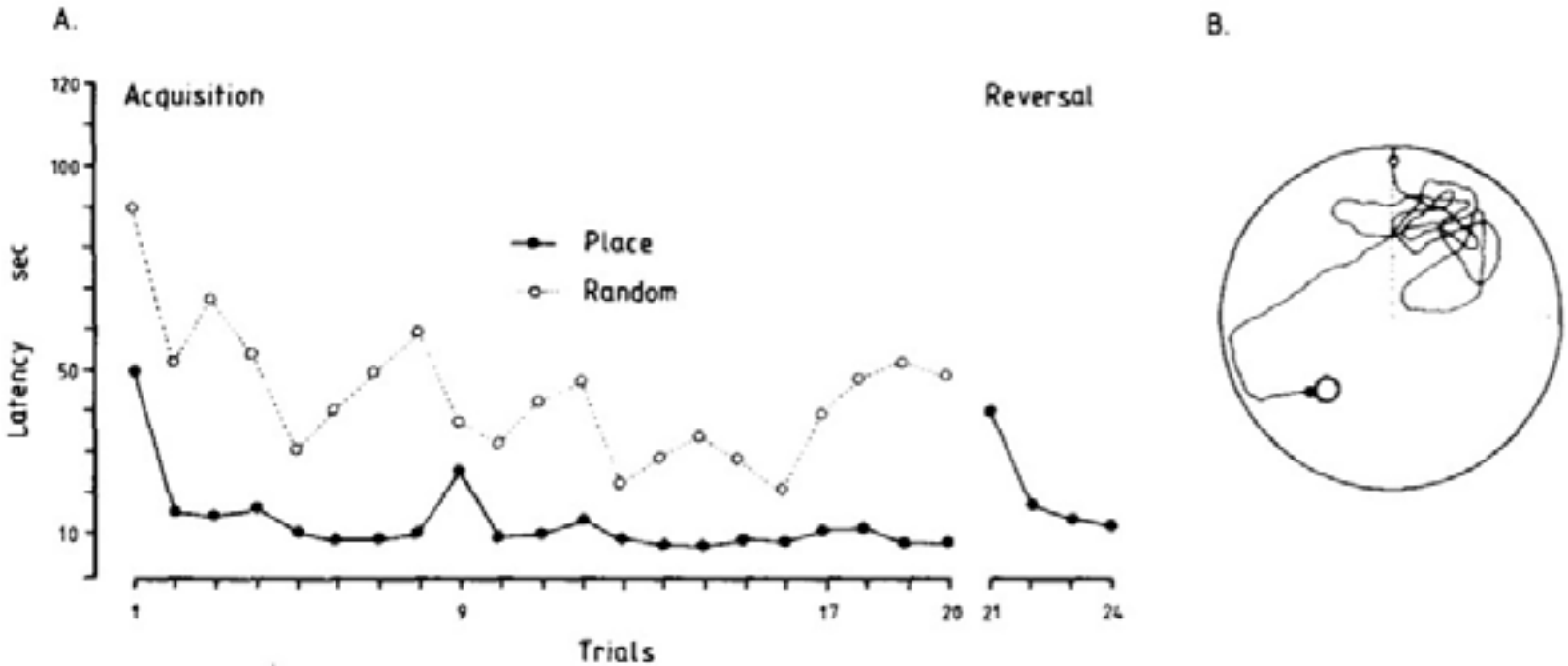


Fig. 4. A: latency to escape from the water over 20 trials of acquisition by Groups Place and Random. The rapid escape by rats of Group Place cannot be due to local cues from the platform, reflected waves etc. Group Place were then given 4 trials of 'reversal'. Note increased latency on trial 21. B: path taken on trial 21 by one typical rat. Note initial searching in former platform location in NE. prior to finding platform now in SW.

Critic

Total expected reward is represented by the firing of a single cell C, which reflects the weighted sum of place cell inputs $f_i(p)$:

$$C(p) = \sum_i w_i \, f_i(p)$$

The weights w_i of active place cells are adjusted with

$$w_i \rightarrow w_i + \epsilon \, \delta_t \, f_i(p)$$

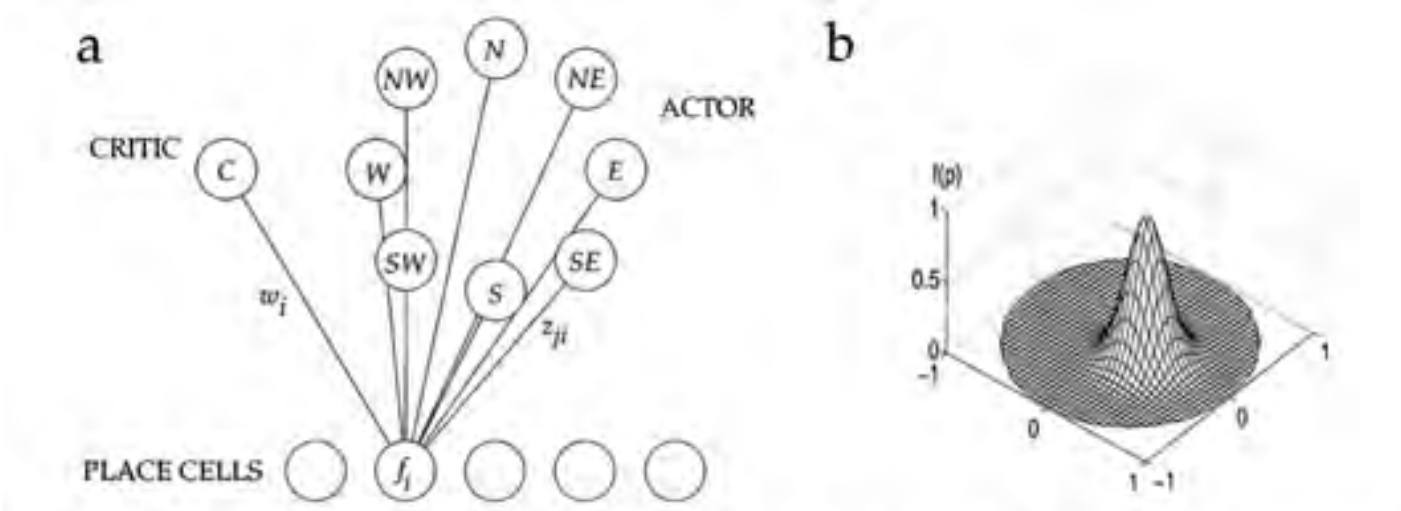


FIGURE 2. The actor-critic system. a: An input layer of place cells projects to the critic cell, C, whose output is used to evaluate behavior. Place cells also project to eight action cells, which the actor uses to select between eight possible directions of movement from any given location. b: An example of a Gaussian place field (x and y axes represent location, z axis represents firing rate).

Actor

At each location p, eight 'action cells' $a_j(p)$ represent different swim directions. Their activity is a weighted sum of place cell activity:

$$a_j(p) = \sum_i z_{ji}\, f_i(p), \qquad P_j = \frac{\exp(2\, a_j)}{\sum_k \exp(2\, a_k)}$$

Together, 'action cell' activities determines the probability of swimming in a particular direction:

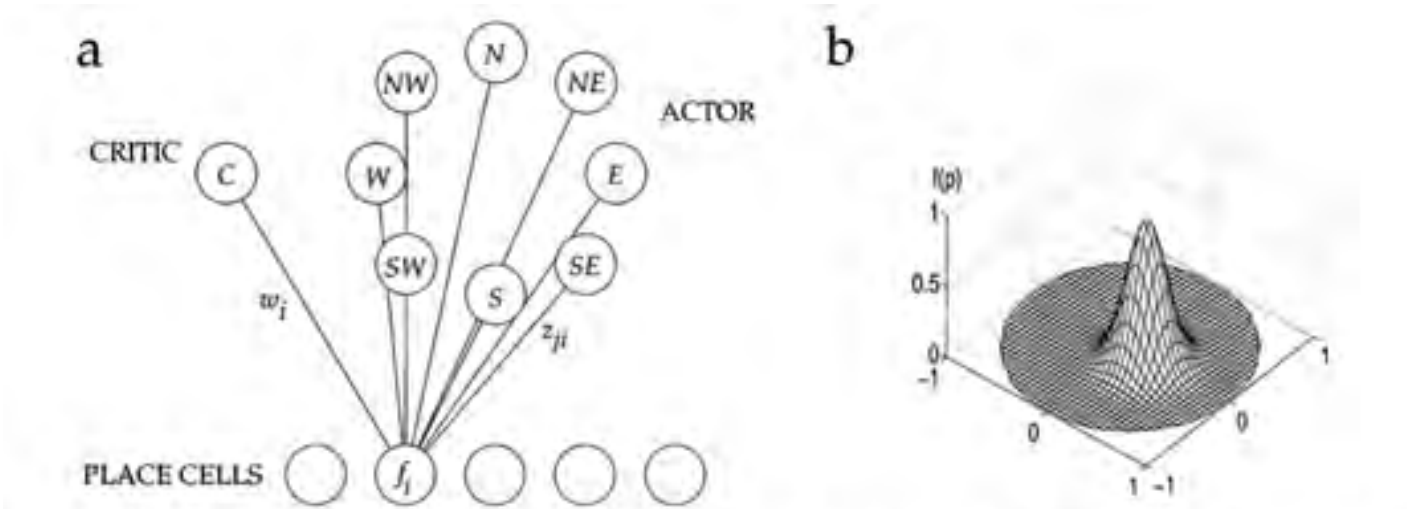


FIGURE 2. The actor-critic system. a: An input layer of place cells projects to the critic cell, C, whose output is used to evaluate behavior. Place cells also project to eight action cells, which the actor uses to select between eight possible directions of movement from any given location. b: An example of a Gaussian place field (x and y axes represent location, z axis represents firing rate).

Place values (top) and action values (bottom)

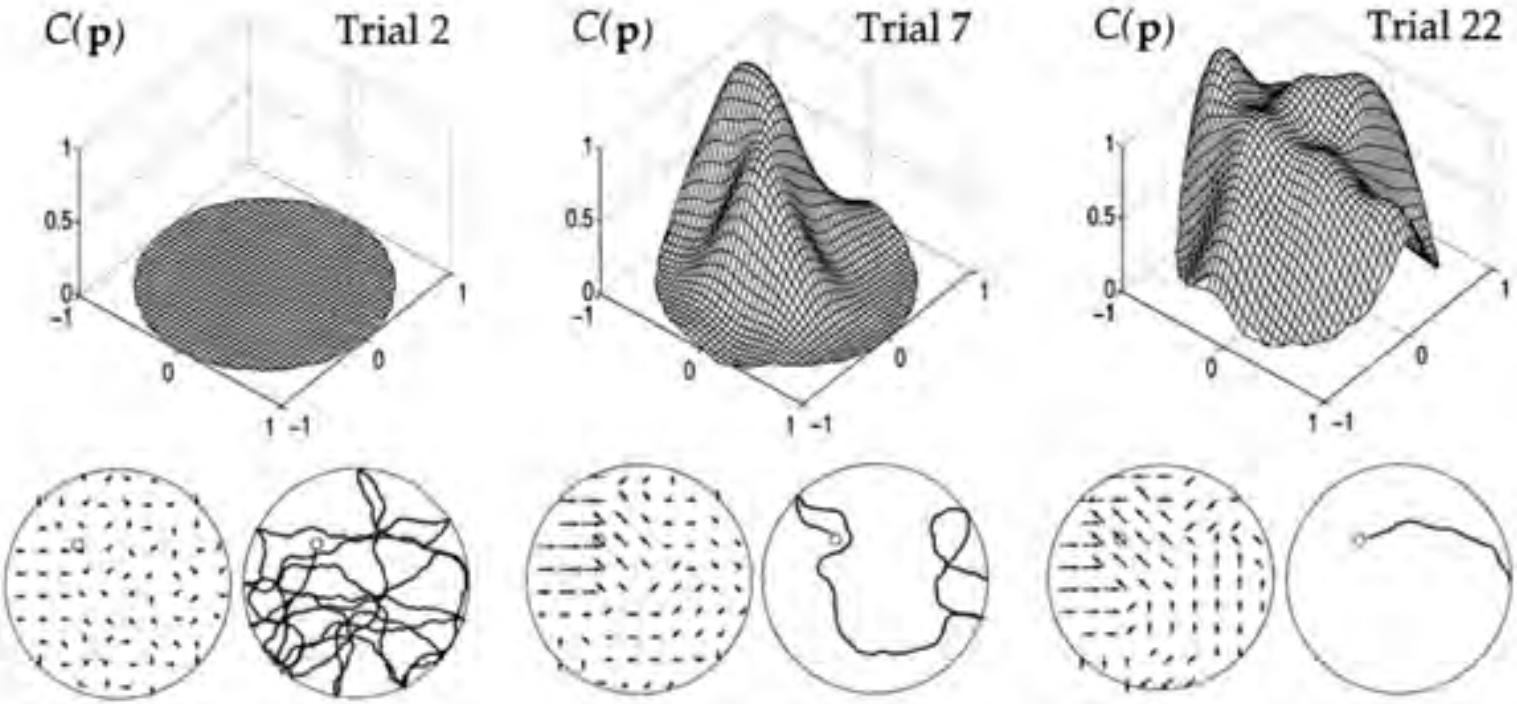


FIGURE 3. Learning in the actor-critic system in RMW. For each trial, the critic's value function $C(p)$ is shown in the upper, three-dimensional plot; at lower left, the preferred actions at various locations are shown (the length of each arrow is related to the probability that the particular action shown is taken by a logarithmic scale); at lower right is a sample path. Trial 2: After a timed-out first trial, the critic's value function remains zero everywhere, the actions point randomly in different directions, and a long and tortuous path is taken to the platform. Trial 7: The critic's value function having peaked in the northeast quadrant of the pool, the preferred actions are correct for locations close to the platform, but not for locations further away. Trial 22: The critic's value function has spread across the whole pool and the preferred actions are close to correct in most locations, and so the actor takes a direct route to the platform.

Animal performance

Same position (1-7 and 8-9)

New position each day

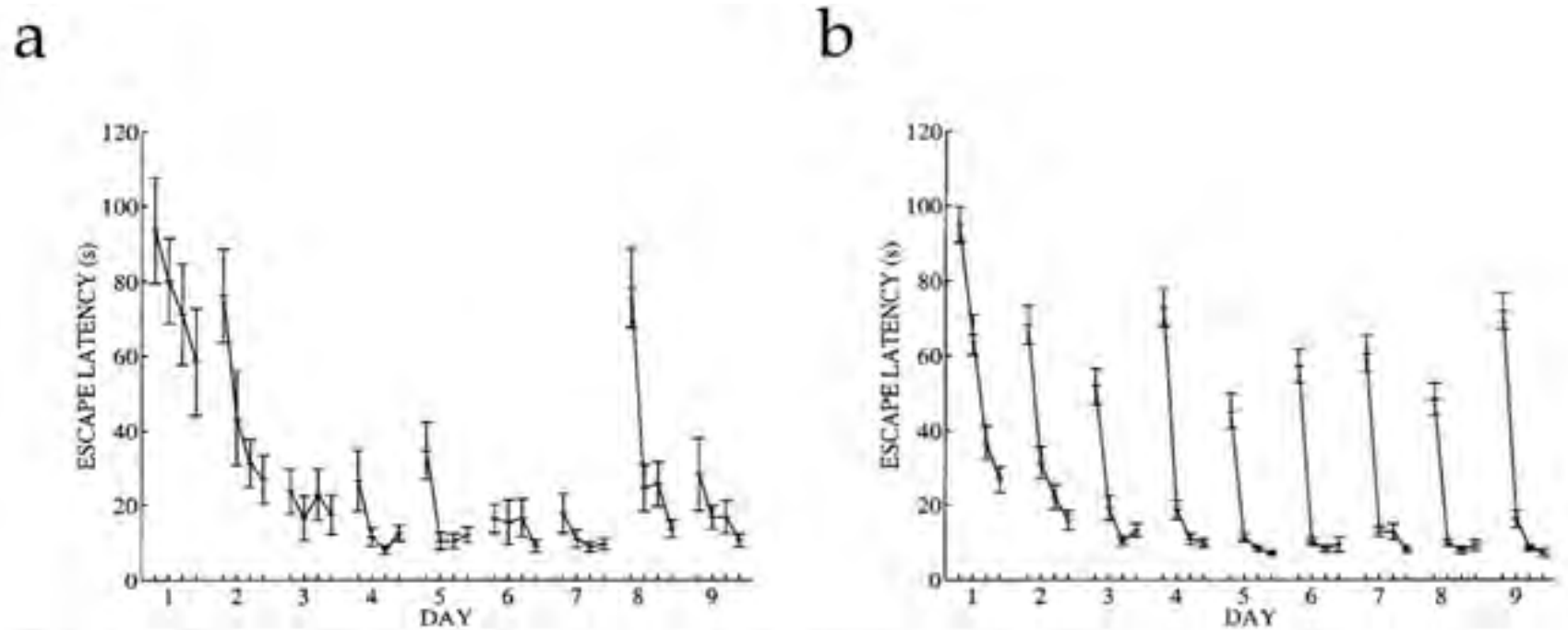


FIGURE 1. Performance of rats on (a) reference memory (RMW), N = 12, and (b) delayed matching-to-place (DMP), N = 62. For both, escape latency (time taken to reach platform) is plotted across days (RMW task: 4 trials/day, fixed platform location, days 1–7; reversal to new platform location, days 8–9; DMP task: 4 trials/day, new platform location each day). Note 1) asymptotic performance in RMW task, 2) one-trial learning in DMP task, and 3) difference in escape latency on second trial of day 8, between the two tasks. Trial 1 performance differs from day to day, due to platform position. It was observed that platforms nearer the center of the pool, or near to a starting position, were easier to find under random search than others. (b) is from Steele and Morris (1999); data for (a) were obtained in the same apparatus and using the same methods as those described for the DMP task by Steele and Morris (1999), with permission, except that: 1) the platform remained in the same location across days, until moved to the opposite quadrant on day 8; and 2) the intertrial interval was always 15 s.

Model performance

Same position (1-7 and 8-9)

New position each day

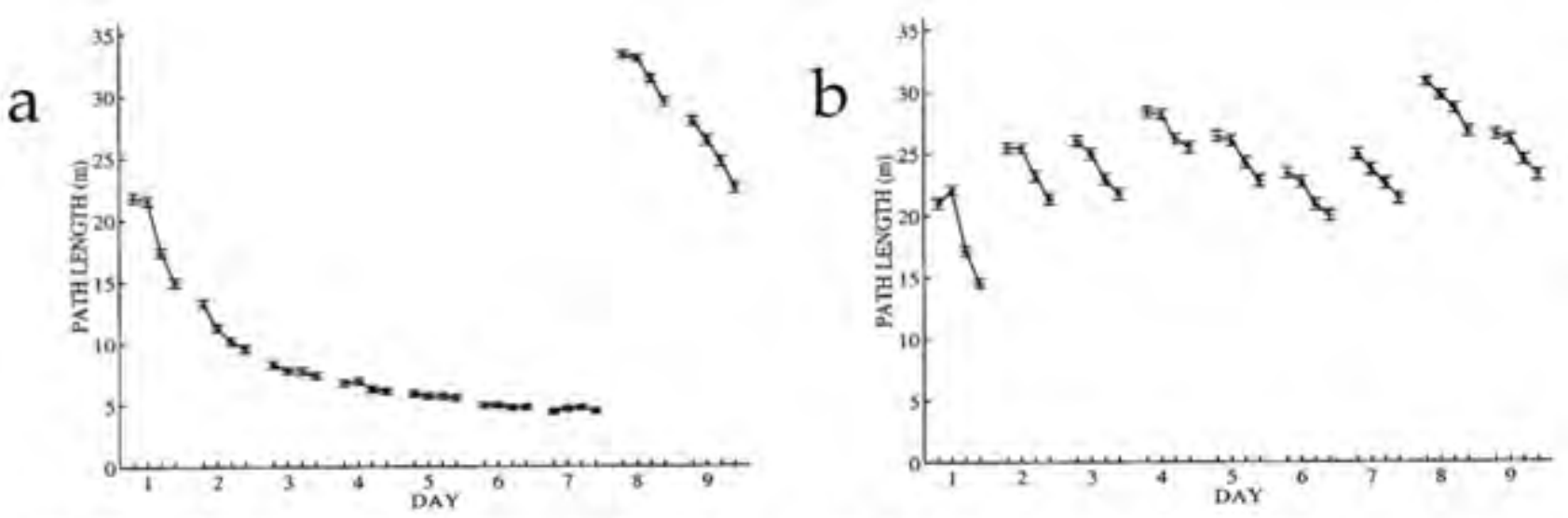


FIGURE 4. Performance of the actor-critic model. For each data point, the mean and standard error in the mean are obtained from 1,000 simulation runs. (a) RMW task, in which the platform occupies the same location. The actor-critic captures acquisition, producing direct paths after around 10 trials. For the last eight trials, however (days 8 and 9), the platform is moved to a different position (reversal), and the model fails to adapt rapidly enough. These simulation results can be compared to Figure 1a. (b) DMP task, in which the platform remains in the same position within a day, but occupies a novel position on each new day. The actor-critic model captures acquisition for the four trials of day 1, for which the task is indistinguishable from RMW. However, as a soon as the platform is moved, the actor-critic not only fails to generalize to the new goal location, but suffers from interference from the previous days' goal locations. Rats suffer neither of these limitations (Fig. 1b).

Model must unlearn old position before learning new position!

Points to note

- Our MWM model includes *place cells* i, an *expected value cell* (critic), and several *action cells* j.
- Learning increments feedforward weights w_i (place $\rightarrow$ value) and z_{ij} (place $\rightarrow$ action).
- Initially, the value function peaks at the platform location. Subsequently, it spreads over the entire pool.
- Action values gradually favor the swim direction pointing towards the platform.
- Value function and action values are incremented with the TD prediction error.
- We observe incremental learning of expected value.

Comparing animal and model performance

- Model does not need to be refreshed, unlike animal.
- Model fails to generalize, even partially, to new platform positions, unlike animal.
- Model learning starts from neutral or unfavourable initial condition, unlike animal.

5 Extended actor-critic model

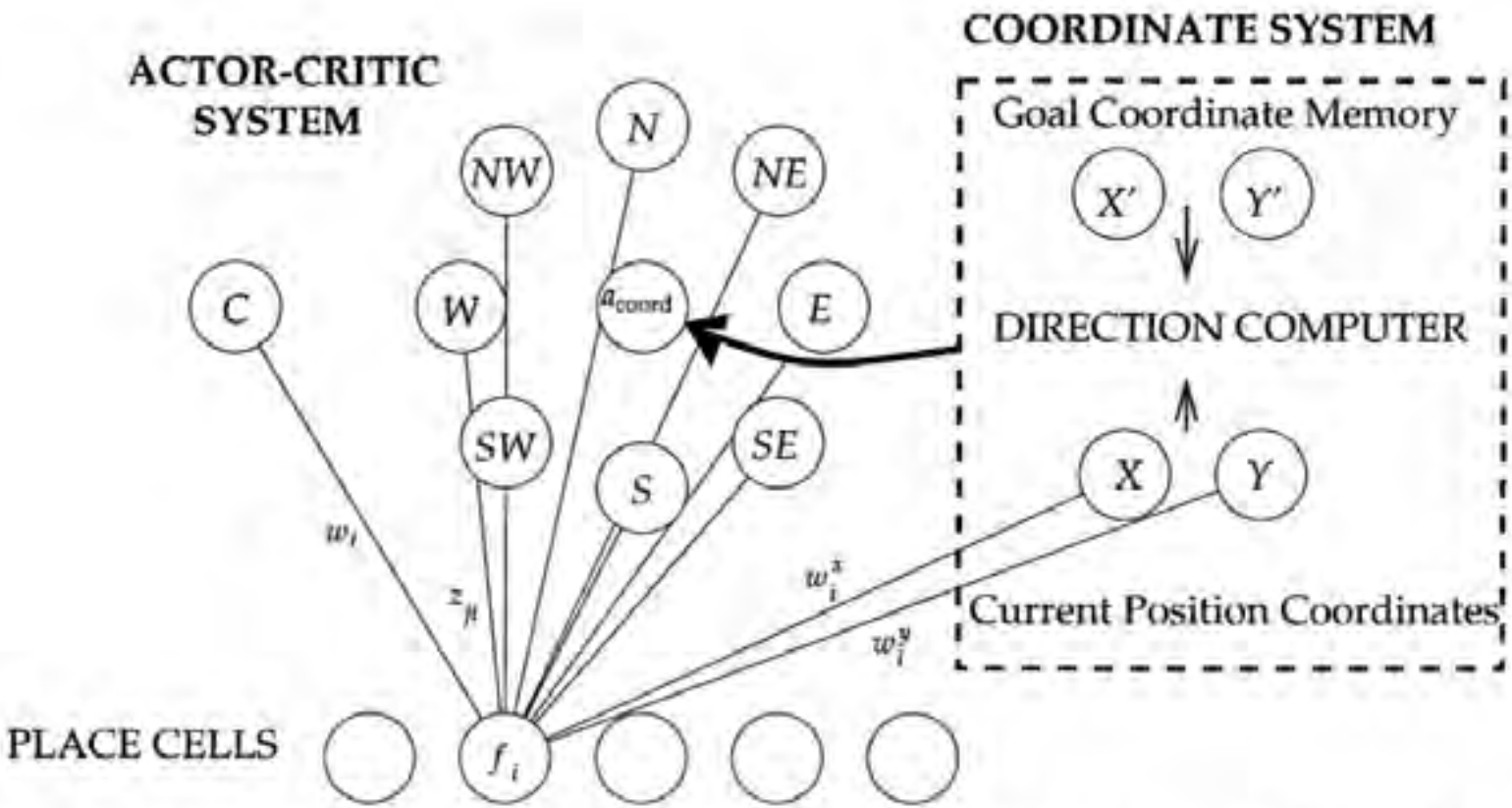


FIGURE 5. The combined coordinate and actor-critic model incorporates both the actor-critic system and a coordinate system. The coordinate system consists of three components: 1) a coordinate representation of current position made up of two cells X and Y, the firing of which is a function of place cell input; 2) a goal coordinate memory consisting of two cells, X' and Y', whose firing reflects the coordinate location of the last place at which the platform was found; and 3) a mechanism which computes the direction in which to swim to get from the current position to the goal. The output direction from the coordinate system is integrated with that from the actor-critic through the "abstract action," marked a_{coord} which receives reinforcement depending on its performance.

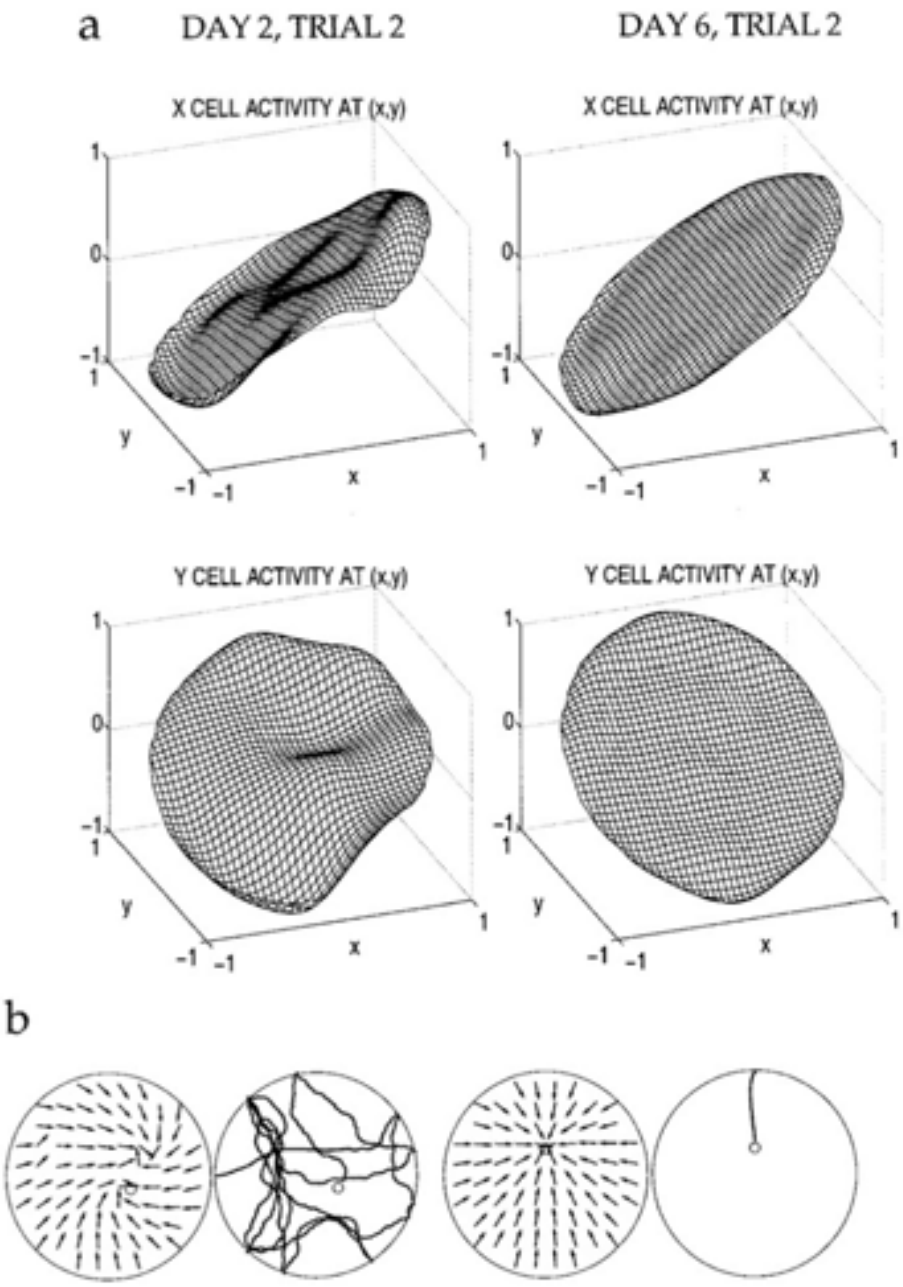
Extended model includes information that generalizes to new platform positions:

- X-Y cells learn to represent position within pool (from place cell input).
- X'-Y' cells represent current platform location ('wherever I am, this is the direction to take').
- 'Global' swim direction is computed from comparison.

Unchanged:

- 'Local' swim direction still learned from prediction error.
- Value function still learned from prediction error.
- Prediction error still computed from reward (reaching platform) and from change in value function.

X-Y coordinate cells



Gradient of coordinate functions

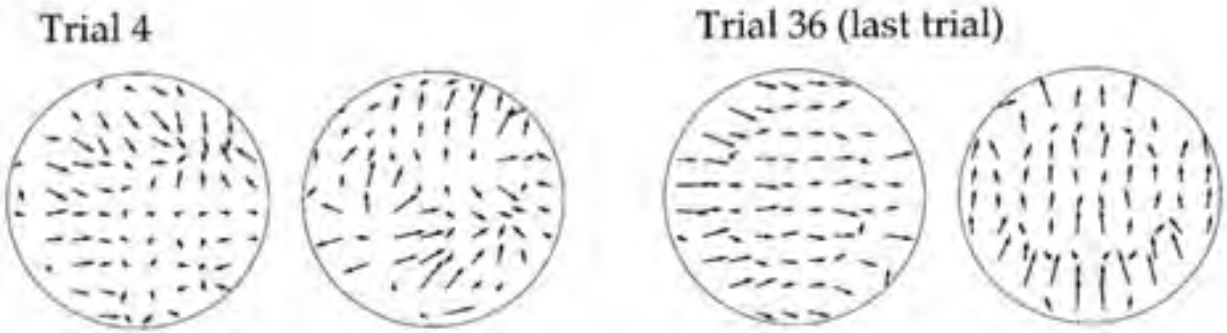


FIGURE 7. Gradient of the coordinate functions. The gradient is a very sensitive measure of smoothness. On trial 4, coordinates are still not at all smooth; navigation based on these functions alone would be prone to catastrophic loops, i.e., would never reach the platform. By comparison, the actor-critic scheme develops effective values and actions for control by trial 4 (Fig. 3), and it is this control that allows the rat to move through the environment, and so improve its coordinate functions. By trial 36, coordinates are smoother and the gradients reflect the *X* and *Y* directions.

Performance of combined model

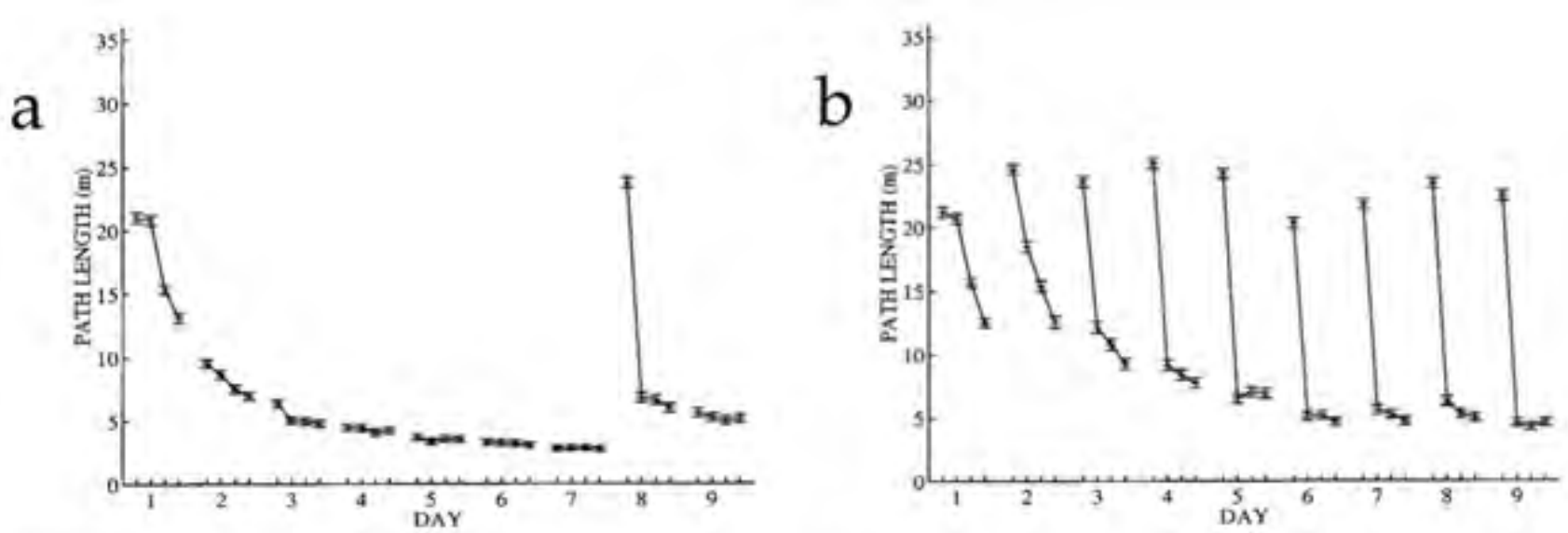


FIGURE 8. Performance of the combined coordinate and actor-critic model. For each data point, the mean and standard error in the mean are obtained from 1,000 simulation runs. (a) RMW task, in which the platform occupies the same location. The combined model captures both acquisition, producing direct paths after around 10 trials, and reversal, producing rapid adaptation to the change in platform position on day 8 (see Fig. 1a). (b) DMP task, in which the platform remains in the same position within a day, but occupies a novel position on each new day. The combined model captures the acquisition of one-trial learning: the improvement within each day is gradual early in training, but becomes a one-trial improvement by day 6. The model provides a good match to the data (Fig. 1b).

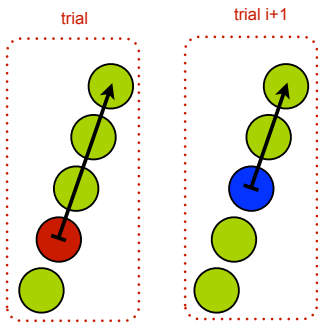
Comparing animal and model performance

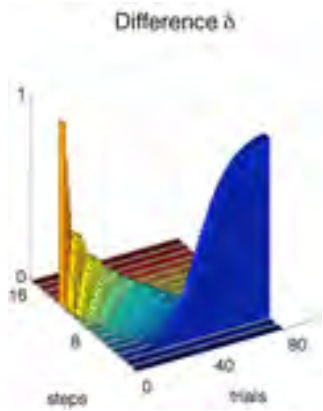
- Model does not need to be refreshed.
- Model now generalizes, at least partially, to new platform positions.
- Global swim direction (coordinate cells) does not need to be relearned.
- Learning of local swim direction starts over from zero.

Reinforcement summary I

- Reinforcement learning explains *classical* and *instrumental conditioning*.

- *Rescorla-Wagner* rule evaluates environment in terms of expected immediate reward: *Pavlovian – Extinction – Partial – Blocking – Inhibitory – Overshadowing*
- *Temporal-difference* rule in terms of expected total reward (immediate and delayed): *Trace – Secondary*



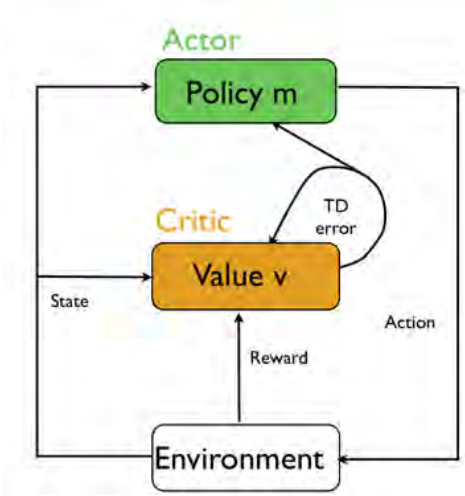


Reinforcement summary II

- *TD prediction error* shifts from reward to earlier, predictive events.
- *TD prediction error* combines actual reward with 'change in hopes' (total expected reward):

$$\delta(s,a) = r(s,a) - v(s) + v(s') \qquad\qquad s \xrightarrow{a} s'$$

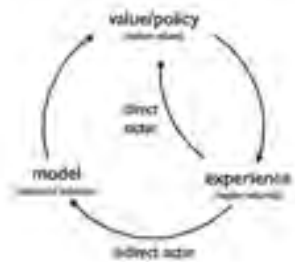
- *TD prediction error* 'bootstraps' newer estimates from older estimates (and occasional reward).
- *TD prediction error* can serve to increment both values and policies.



Reinforcement summary III

In static action choice, learning of *action policy* maximizes expected immediate reward.

- *Direct actor* maximizes total expected reward.
- *Indirect actor* learns expected rewards for individual actions.

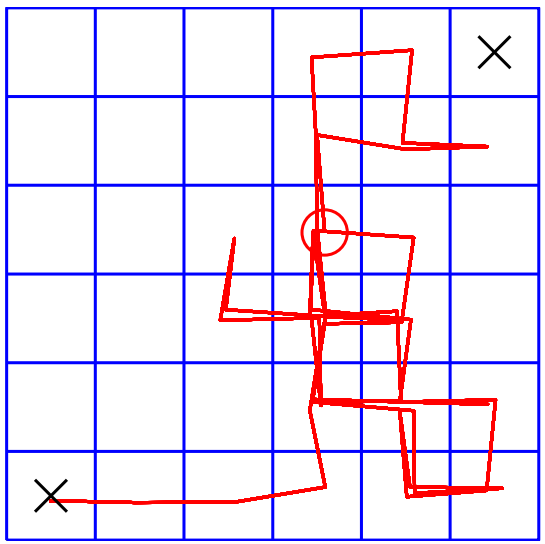


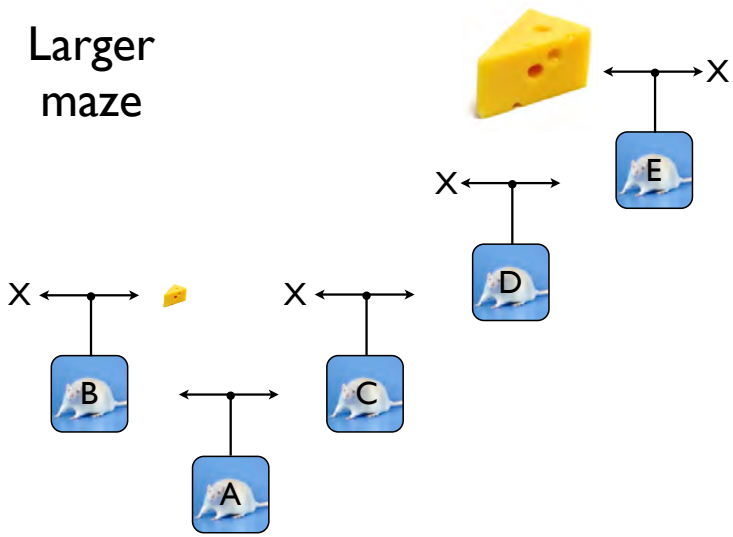
In sequential action choice, learning maximizes expected total reward (immediate and delayed).

- *Policy evaluation* assesses the value (total expected reward) of states, given a fixed policy.
- *Policy improvement* assesses the policy, given a fixed value.
- *Policy iteration* alternates between evaluation and improvement.

Reinforcement summary IV

- In *actor-critic models*, evaluation is performed by the *critic* (value function) and improvement by the *actor* (policy).
- Actor and critic may be incremented either in alternation or concurrently, using the TD prediction error.
- Actor-critic models develop foresight when sequential actions lead to delayed rewards.





Next:
Representational learning