Reward Learning

NEU/MOL 502A: **From Molecules to Systems to Behavior**

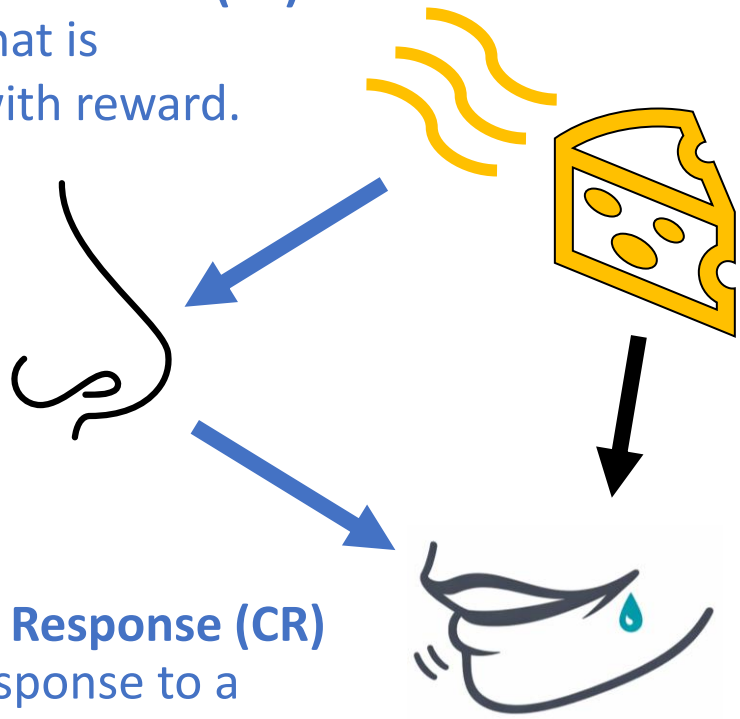
What are rewards and why do we care about them?

- Reward is something that leads to a positive outcome
  - →elicits characteristic approach responses & appetitive facial expressions
  - →has hedonic effects
  - →serves as a reinforcer – it increases the frequency of the actions that produce it
  - →serves as an incentive or goal for actions
- can be primary (e.g., food, water, sex) or secondary (e.g., money)

Predicting rewards from stimuli

One can learn to predict a rewarding event based on associated stimuli.

Conditioned Stimulus (CS)
A stimulus that is associated with reward.

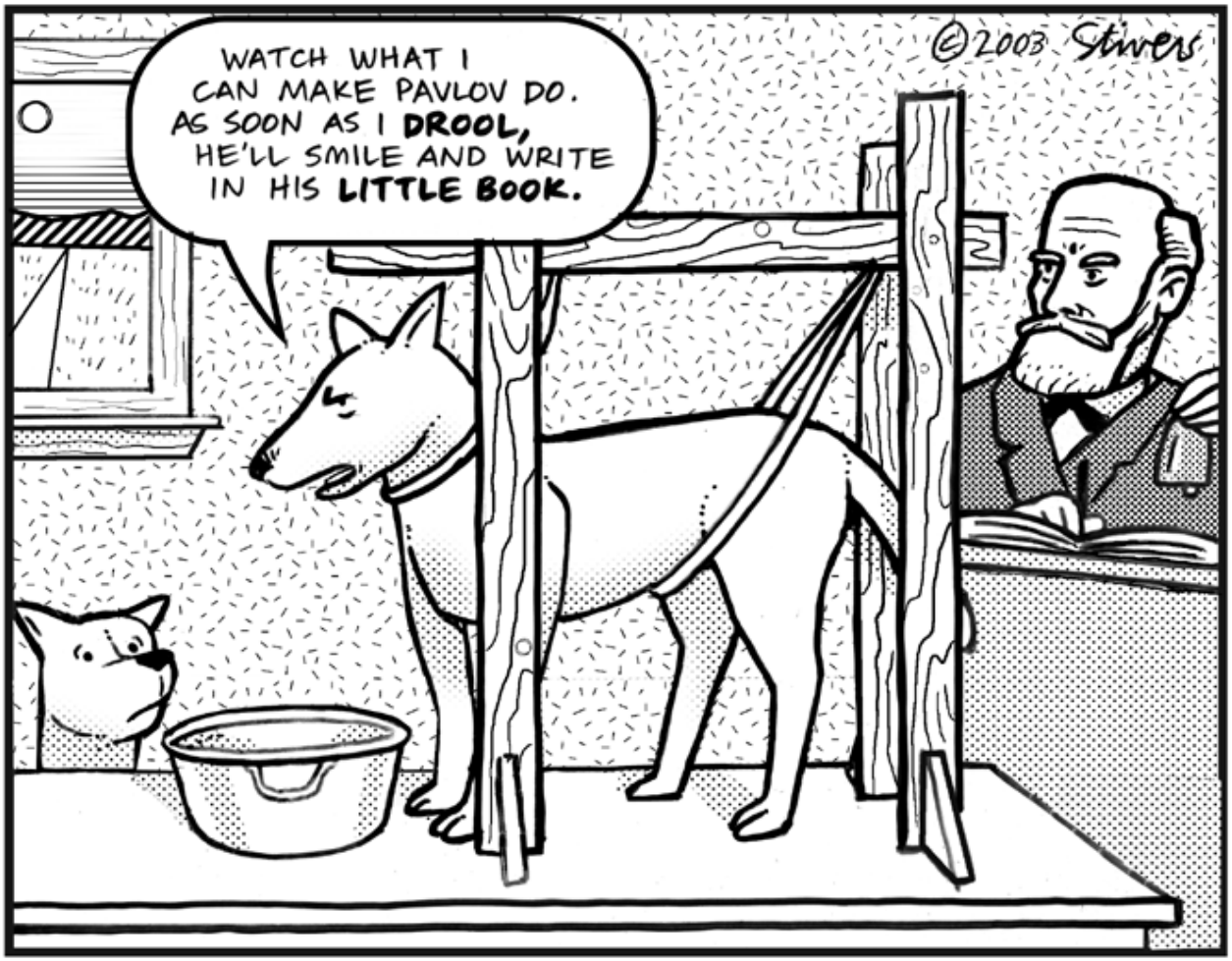


Unconditioned Stimulus (US)
An *inherently* rewarding stimulus.

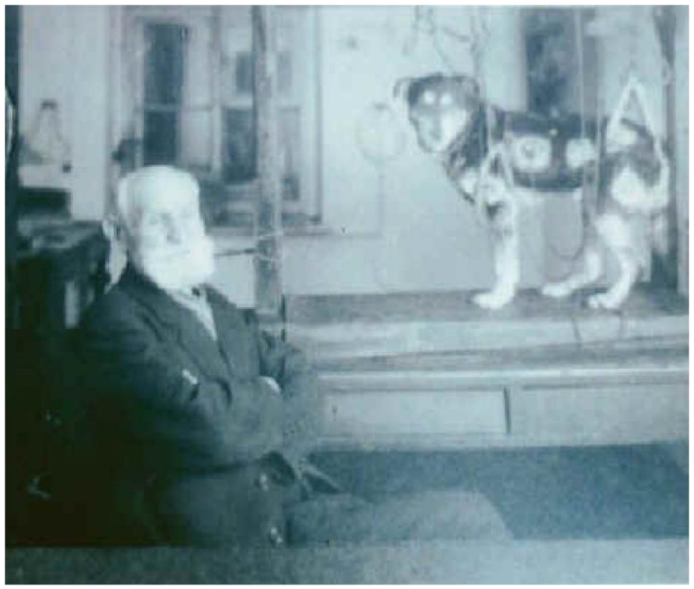
Unconditioned Response (UR)
An innate response to the stimulus.

Conditioned Response (CR)
A learned response to a stimulus based on its association with reward.

Classical (Pavlovian) conditioning: predicting future rewards



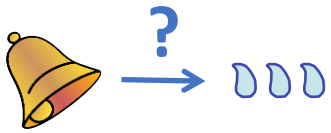
Classical (Pavlovian) conditioning: predicting future rewards

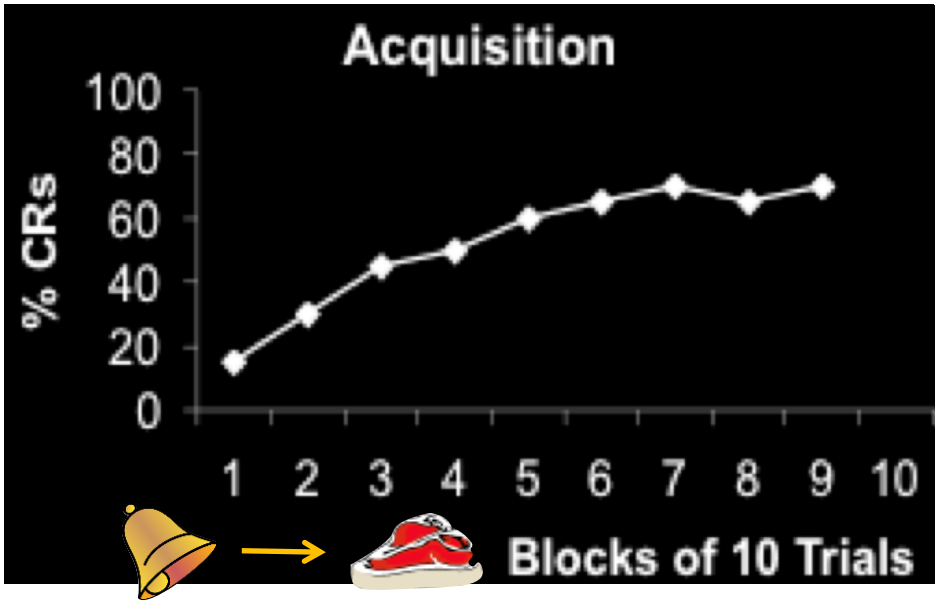


pair stimulus (CS)

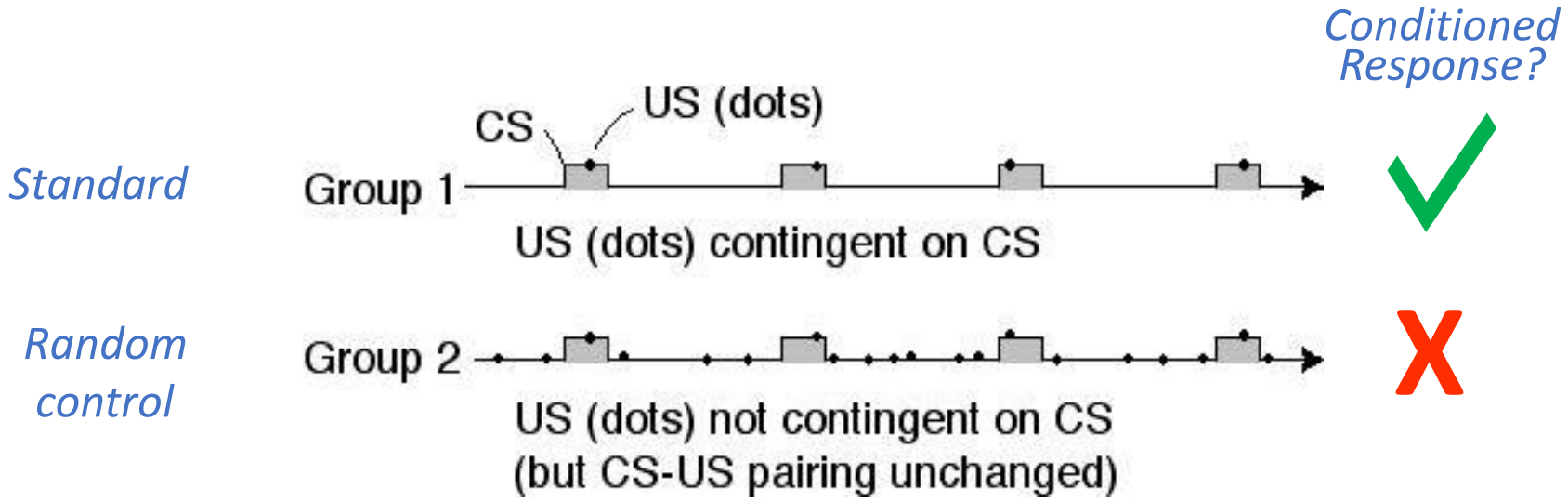
...with significant event (US)

measure anticipatory behavior (CR)





Conditioning requires a contingency between the stimulus and reward



Temporal alignment alone is <u>not sufficient</u>.

Predictions require there to be *information.* The CS predicts the US only if there is information about when the US occurs in the CS.

Learning to predict rewards

We want to predict the reward based on the current state (stimulus).

S ⟶ Stimulus R ⟶ Reward


We can do that by minimizing the error between the value of the stimulus and the observed reward:

$$\boldsymbol{E = \frac{1}{2}(V(S) - R)^2}$$

We want to update the value of the stimulus to minimize this error:

$$\boldsymbol{\partial E / \partial V(S) = V(S) - R}$$

$$\boldsymbol{\Delta V = -\partial E / \partial V(S)}$$

"Prediction Error"

Associations are strengthened to minimize prediction error:

$$\mathbf{w_{t+1} = w_t + \alpha \Delta V}$$

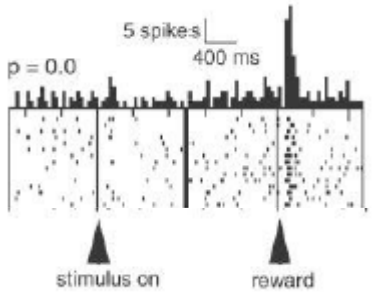
Association Weight **Learning Rate**

Example of reward learning

The ***Rescorla-Wagner*** update rule iteratively learns the stimuli that predict rewards.

		Prediction error			*Weight adjustment*		
		$\Delta v = r_t - v_t$			$w_{t+1} = w_t + \alpha \Delta v$		
Trial 1		+	+	0	+	0	
Trial 2		0	+	+	+		0

Dopamine neurons respond to prediction errors



reward following
0% predictive cue

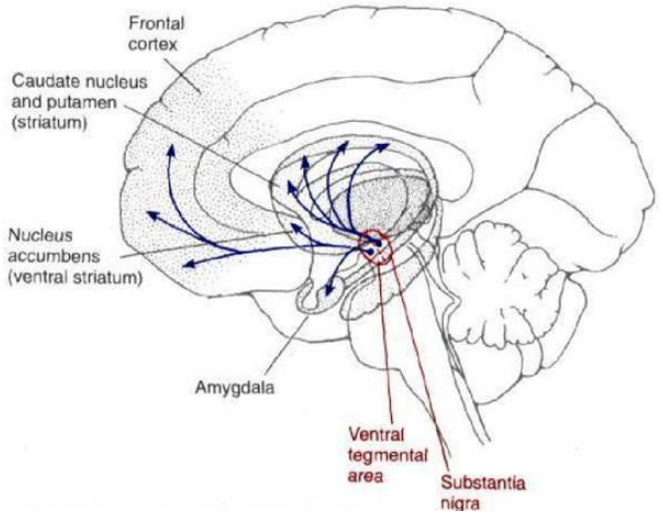
Prediction error

$\Delta v = r_t - v_t$

Trial 1 + + 0

Dopamine (DA) is released from two midbrain regions:

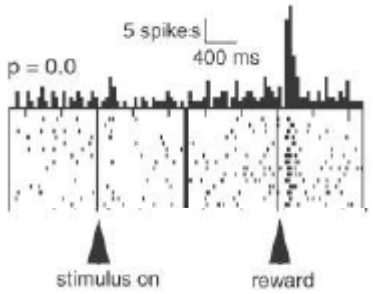
- Ventral tegmental area (VTA)
- Substrantia Nigra, pars compacta



Burst to
unexpected reward

Schultz et al. 1997; Fiorillo et al, 2003

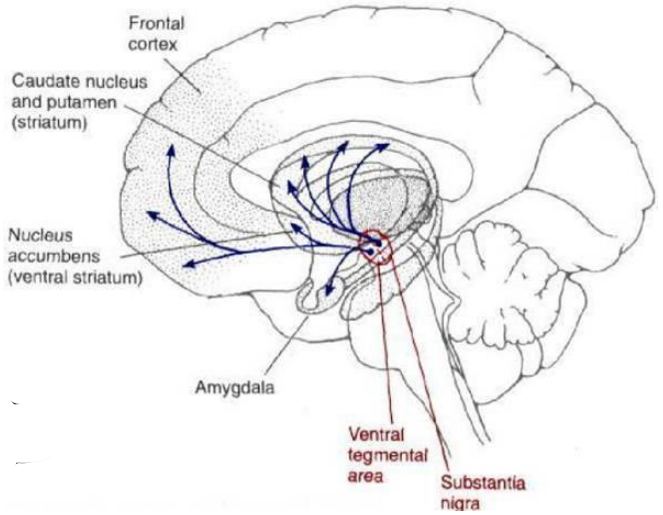
Dopamine neurons respond to prediction errors



reward following
0% predictive cue

Dopamine (DA) is released from two midbrain regions:

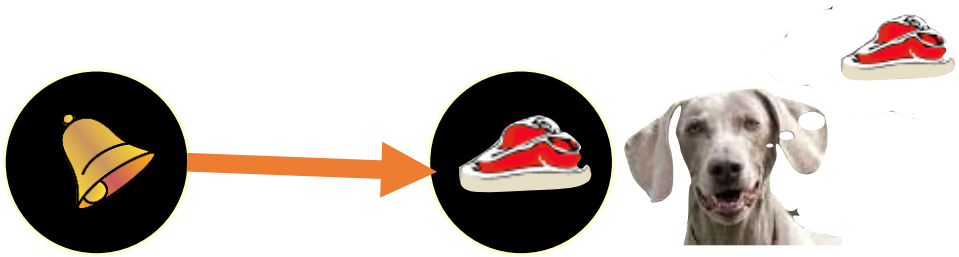
- Ventral tegmental area (VTA)
- Substrantia Nigra, pars compacta



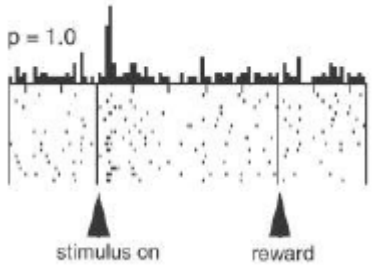
Prediction error

$\Delta v = r_t - v_t$

Trial 1	+	+	0
⋮			
Trial N	0	+	+



Burst to
unexpected reward

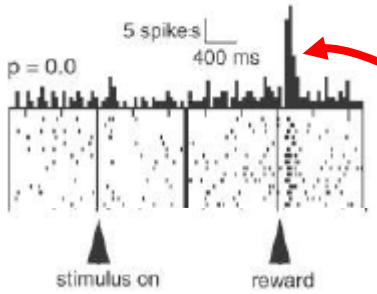


reward following
100% predictive cue

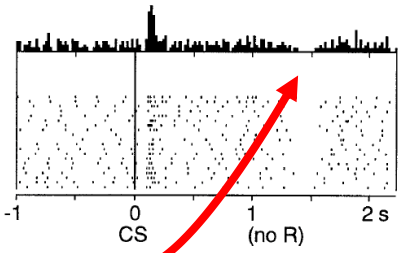
Response transfers
to reward predictors

Schultz et al. 1997; Fiorillo et al, 2003

Dopamine neurons respond to prediction errors



reward following
0% predictive cue



no reward following
100% predictive cue

Prediction error

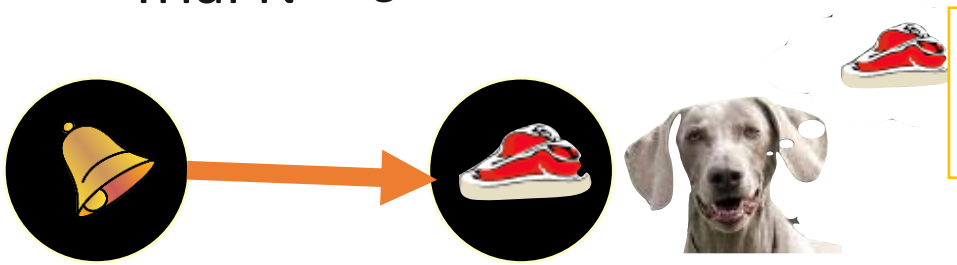
$\Delta v = r_t - v_t$

Trial 1	+	+	0
⋮			
Trial N	0	+	+

"Prediction Error"

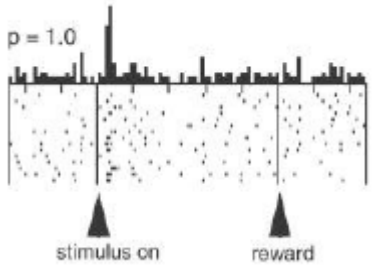
$\Delta V = R - V(S)$

Can be both positive and negative.



Suppressed when expected reward is omitted

Burst to unexpected reward

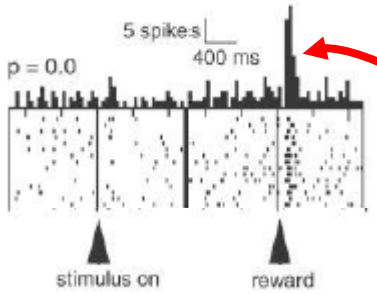


reward following
100% predictive cue

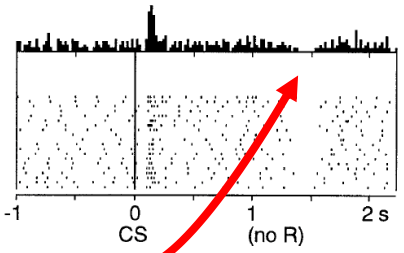
Response transfers to reward predictors

Schultz et al. 1997; Fiorillo et al, 2003

Dopamine neurons respond to prediction errors



reward following
0% predictive cue



no reward following
100% predictive cue

Prediction error

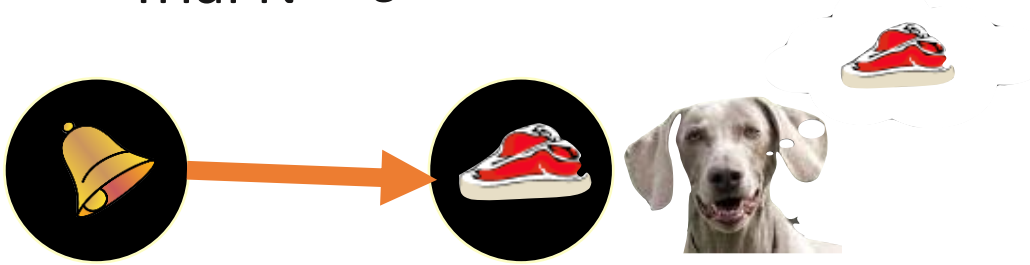
$\Delta v = r_t - v_t$

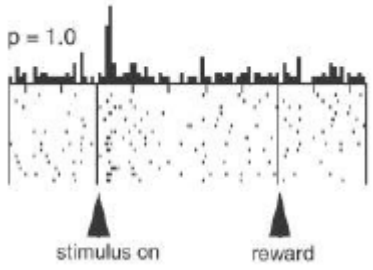
Trial 1	+	+	0
⋮			
Trial N	0	+	+

"Prediction Error"

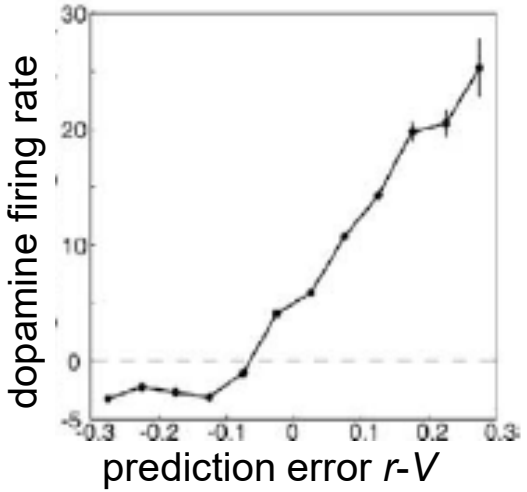
$\Delta V = R - V(S)$

Can be both positive and negative.





reward following
100% predictive cue



DA response to reward as function of prediction error ($r - V$) is quite linear. Negative error cut off due to low baseline response.

Schultz et al. 1997; Fiorillo et al, 2003

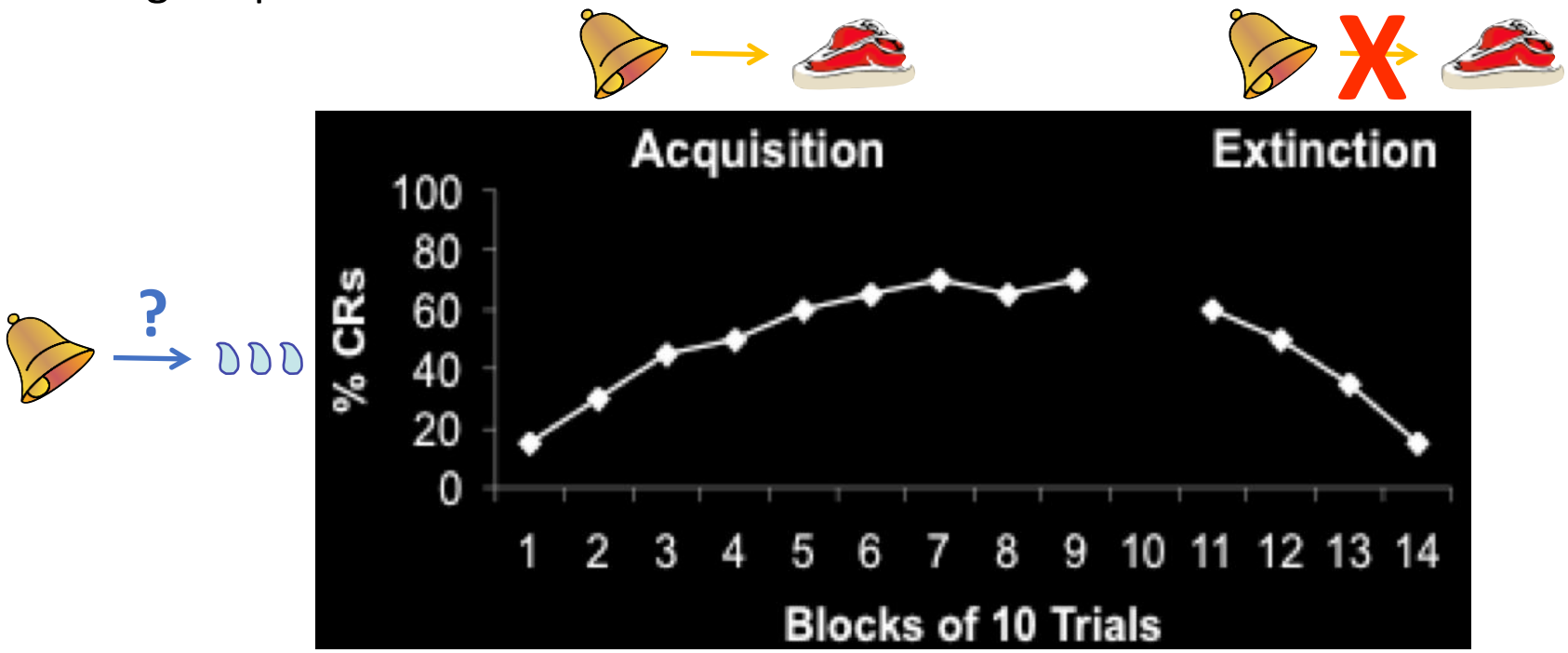
Example of reward learning

The ***Rescorla-Wagner*** update rule iteratively learns the stimuli that predict rewards.

		Prediction error			*Weight adjustment*		
		$\Delta v = r_t - v_t$			$w_{t+1} = w_t + \alpha \Delta v$		
Trial 1		+	+	0	+	0	
Trial 2		0	+	+	+		0
Trial 3		-	0	+	0	+	-
Trial 4							

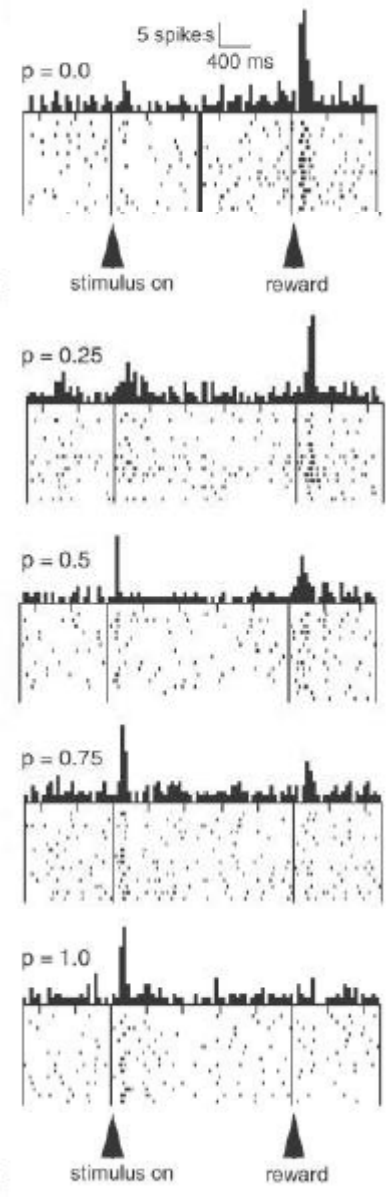
Extinction returns predictions to baseline

The same update rule will lead to extinction of associations when there is a change in predictions.



This reflects learning – allowing the animal to adapt to a changing environment.

Dopamine neurons respond to prediction errors



reward following
0% predictive cue

reward following
50% predictive cue

reward following
100% predictive cue

DA responses converge to the expected value of the reward.

$$\Delta V = (R - V(S))$$

$$w_{t+1} = w_t + \alpha \Delta V$$

$$\Delta V = 0 \rightarrow V(S) = \mathbb{E}[r]$$

$$w = \mathbb{E}[r|cue]$$

(note: α must be reasonably small)

Example for $\mathbb{E}[r] = 0.5$:

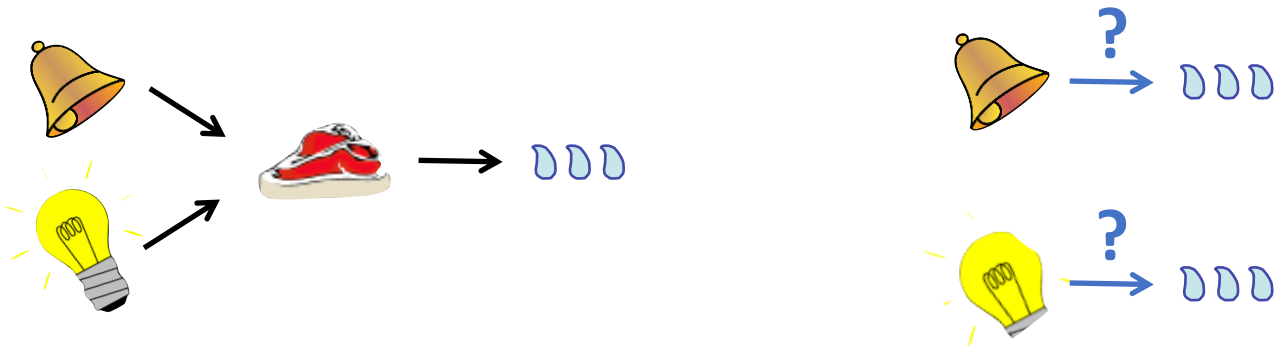
If reward: $R = 1 \rightarrow \Delta V = 0.5$
If no reward: $R = 0 \rightarrow \Delta V = -0.5$

Schultz et al. 1997; Fiorillo et al, 2003

Blocking of associations

Stage 1: Build association between Stimulus A (bell) and reward:

Stage 2: Add a second Stimulus B (light) that is also associated with reward:

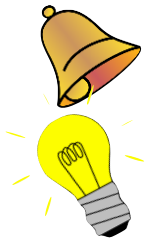


What does RW say about blocking?

Prediction error $\Delta v = r_t - v_t$

Weight adjustment $w_{t+1} = w_t + \alpha \Delta v$

Trial N+1



$$0 \quad + \quad \begin{bmatrix} + & + & + \\ 0 & 0 & 0 \end{bmatrix}$$

Because the reward is already fully predicted, there is no update to the weights. This emphasizes that it isn't association alone but the **prediction error**.

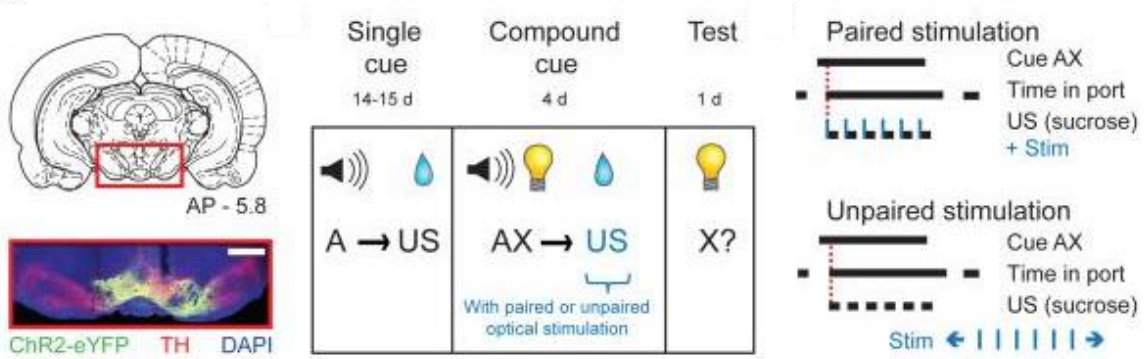
Blocking of associations

Stage 1: Build association between Stimulus A (bell) and reward:

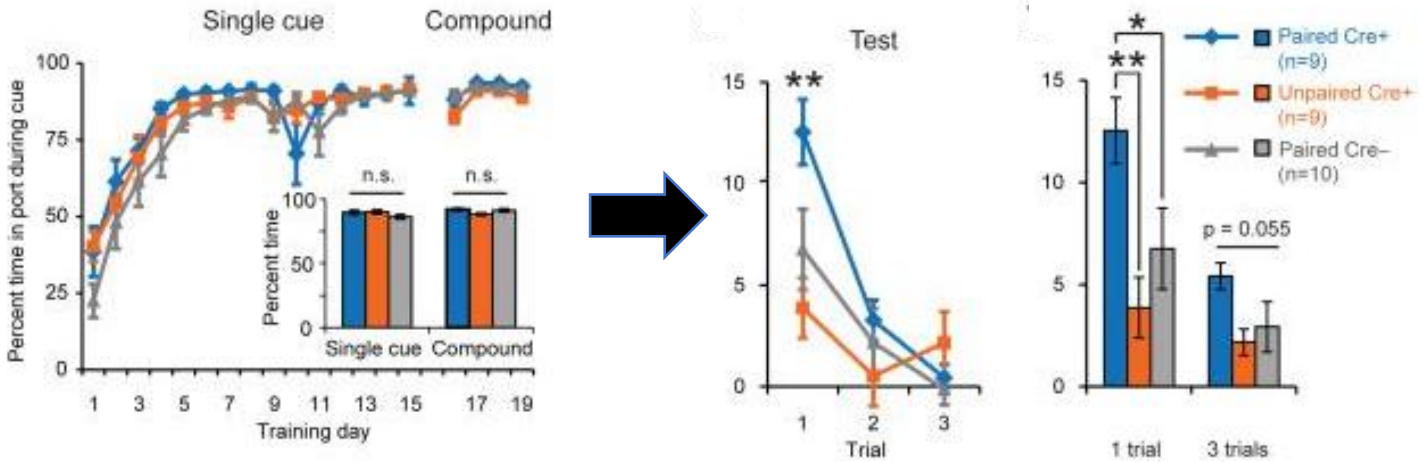
Stage 2: Add a second Stimulus B (light) that is also associated with reward:

Stimulating DA attenuates blocking

Classical conditioning paradigm with optical stimulation of DA+ neurons



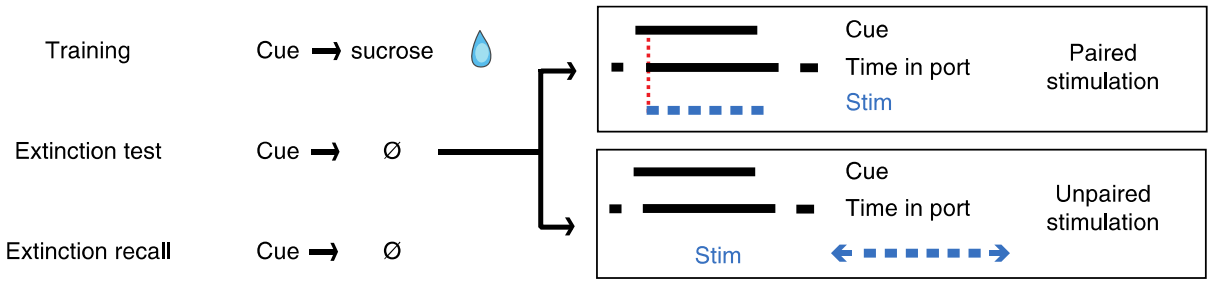
Mice learn associations, but blocking is decreased when paired with DA+:



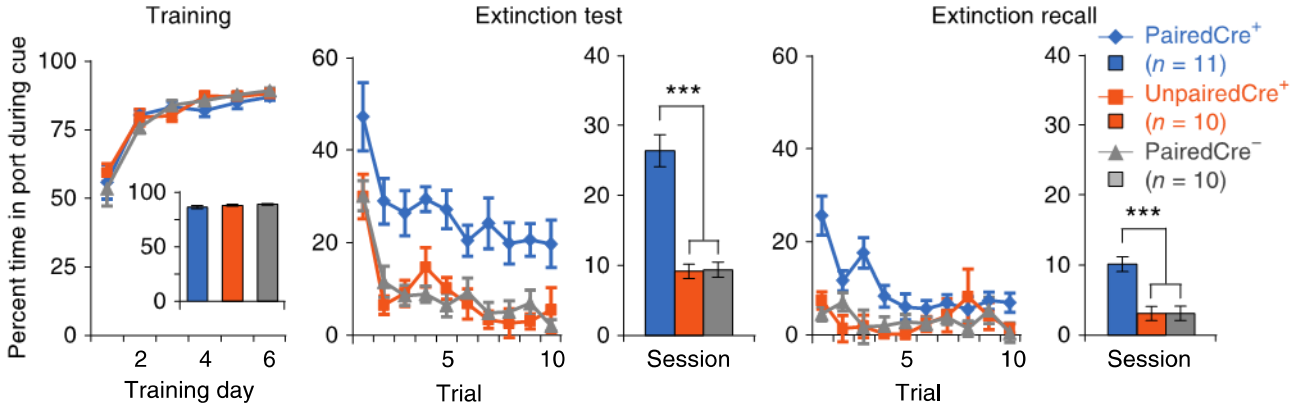
(steinberg et al 2013)

Stimulating DA attenuates blocking and extinction

Classic extinction test, but now either paired with optogenetic stimulation of DA+ neurons or not



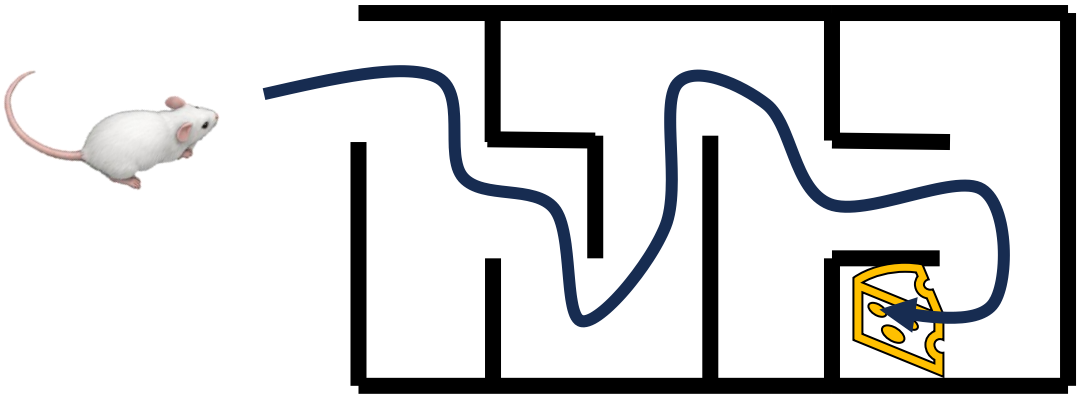
Association is extinguished, but slower when paired with DA+ stimulation:



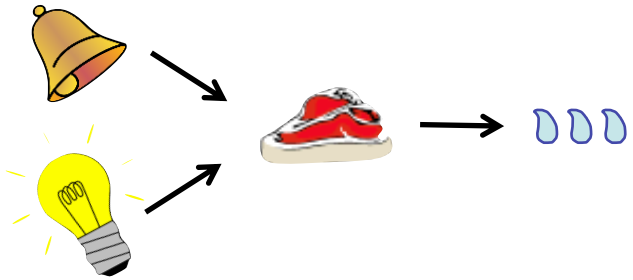
(steinberg et al 2013)

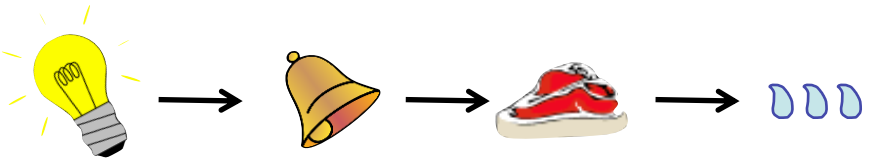
Two main limitations of RW updates

1) Everything so far has been about stimulus-reward associations. There is no agency here!



2) We want to make predictions into the future, but the update rule doesn't chain because there is no reward.



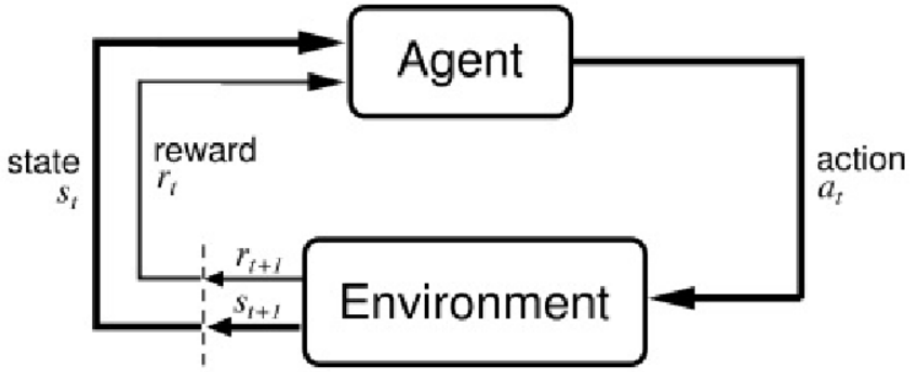


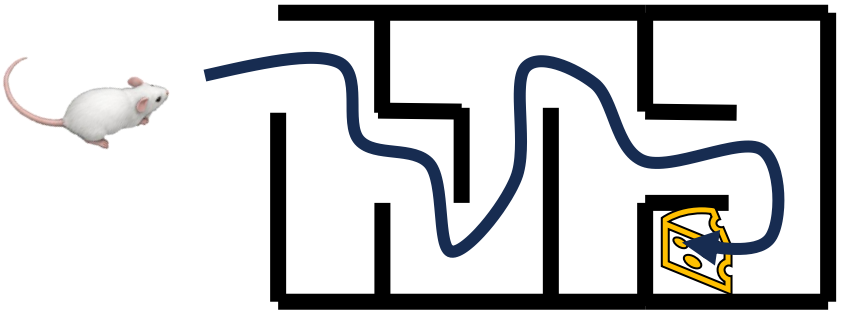
$\Delta v = 0 \quad r_t = 0$

$w_t = 0$

Behavioral policies define how one should act

Agents act on the world to achieve a desired outcome.





Current State

Action

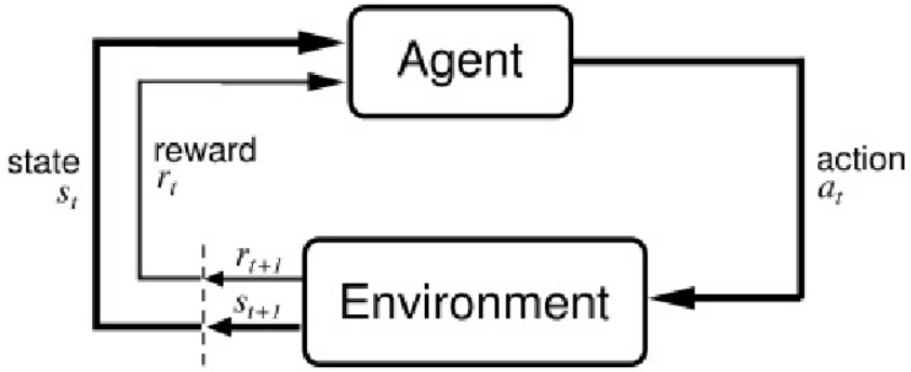
Value $\rightarrow Q(s_t, a_t)$

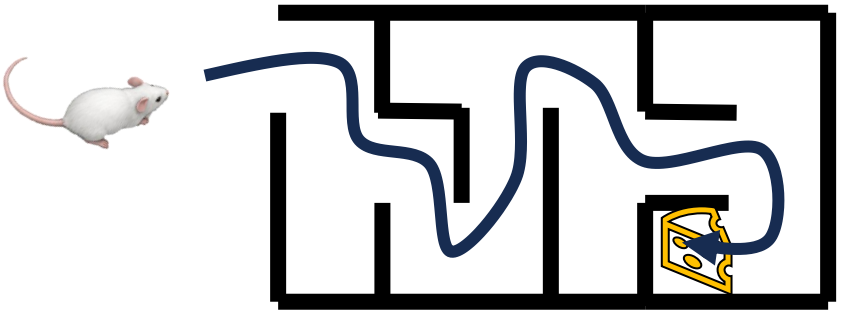
Given a set of expectations about upcoming reward, you can choose the behavior that maximizes your expected return:

$$a_t^* = argmax_a\big(Q(s_t, a_t)\big)$$

Behavioral policies define how one should act

Agents act on the world to achieve a desired outcome.





Current State

Action

Future Reward

Value

$Q(s_t, a_t)$

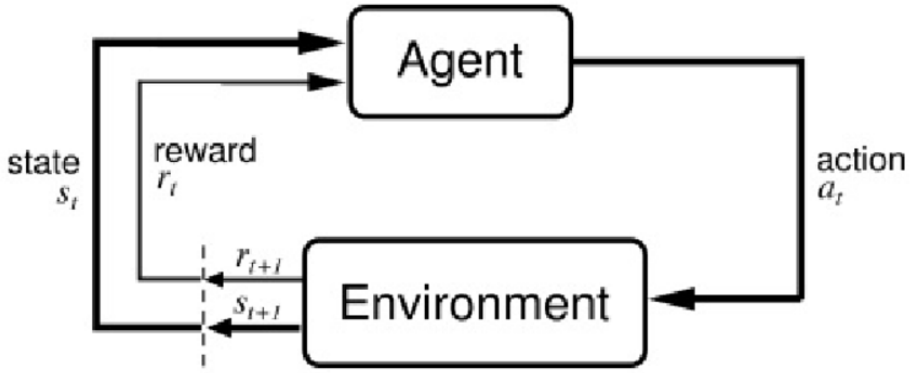
Current Reward

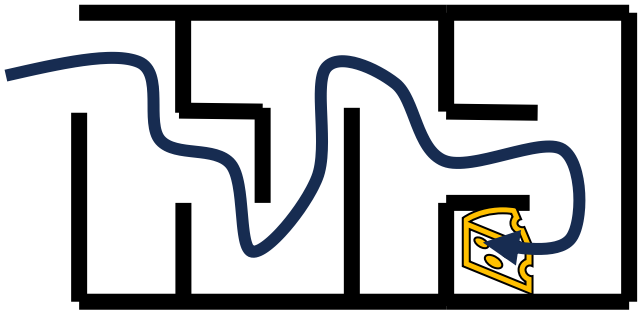
State Transitions

But, this is recursive! Your choices lead to a change in the state of the world, which leads to more choices to be made, etc.

So, you want to **choose a behavioral policy that maximizes total reward**.

Behavioral policies define how one should act





$$Q(s_t, a_t) = r(s_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) r(s_{t+1})$$

$$Q(s_t, a_t) = r(s_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \left[r(s_{t+1}) + \sum_{s_{t+2}} P(s_{t+2}|s_{t+1}, a_{t+1}) [r(s_{t+2}) + \ \ldots] \right]$$

$$Q(s_t, a_t) = r(s_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) Q(s_{t+1}, a_{t+1})$$

This is the **Bellman Equation** and can theoretically be used to estimate ideal behavior (a policy π) that maximizes reward. But this is hard for many reasons…

Temporal difference learning: Stochastic approximation of the Bellman Equation

$$Q(s_t, a_t) = r(s_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) Q(s_{t+1}, a_{t+1})$$

In reality, you *sample* from this distribution by taking a specific action and observing the transition.

So, if you are in state s_t and you perform action a_t and observe reward r_t and transition to state s_{t+1}, then your *observed total reward* is:

$$r(s_t) + Q(s_{t+1}, a_{t+1})$$

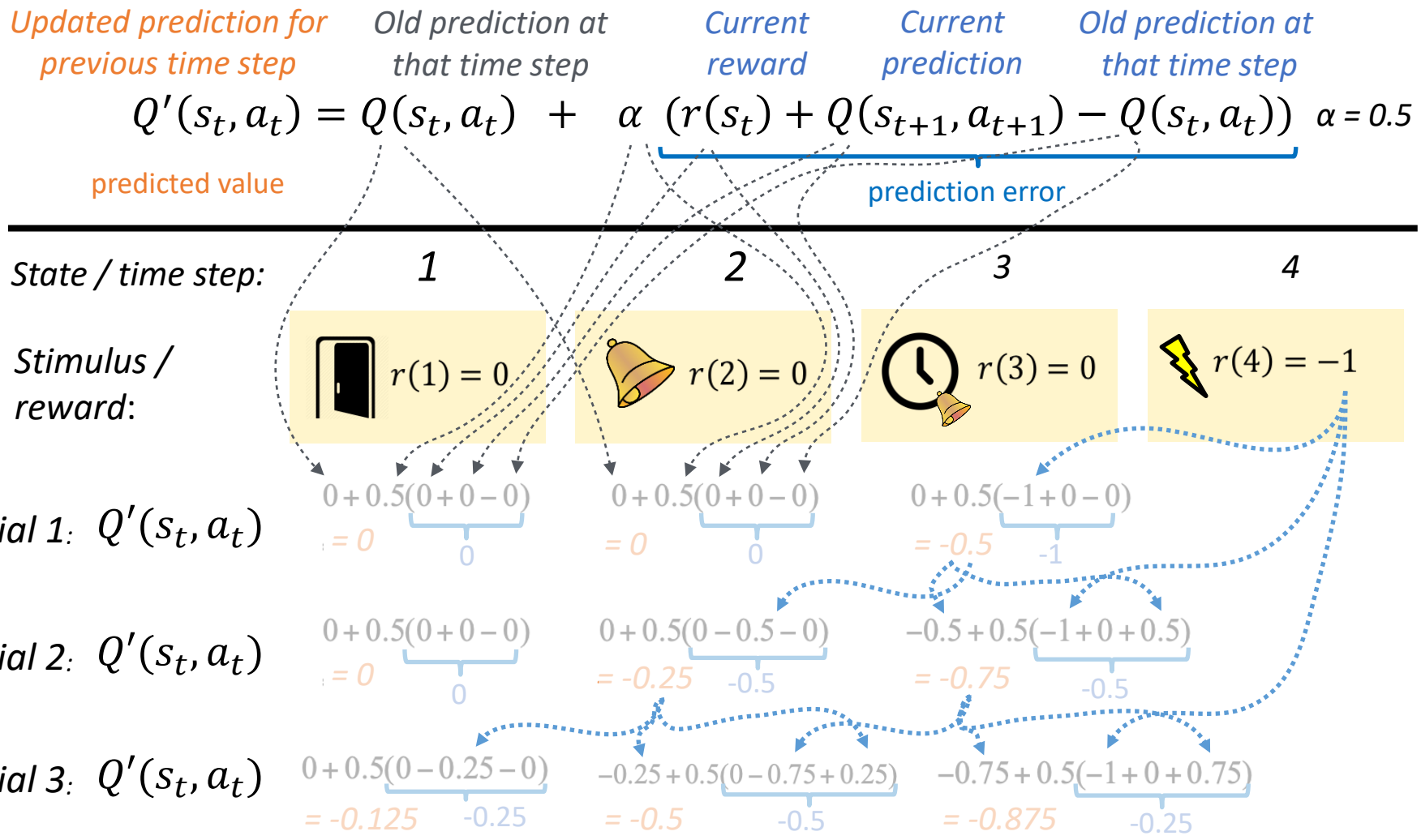
This cleverly incorporates all of that difficult effort to predict the future.

Just as before, we can now take the difference between this observed reward and the expected reward to update our expectations:

$$\delta_t = r(s_t) + Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$$

$$Q'(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \delta_t$$

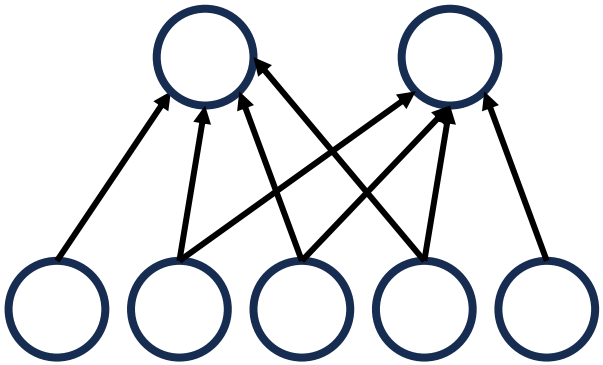
Example of temporal difference learning



Q-value updates are like weighted Hebbian updates

We can estimate the Q value from a neural network:

$$Q_{s,a} \sim w_{s,a} x_s$$



$Q_{s,a}$

$w_{s,a}$

x_s

How do we learn this network? We want to update our weights in a way that minimizes our prediction error:

$$L = \frac{1}{2}\delta^2 \qquad \Delta w_{s_i,a_j} = -\frac{\partial L}{\partial w_{s_i,a_j}}$$

$$\frac{\partial L}{\partial w_{s_i,a_j}} = \delta \frac{\partial (r(s_t) + Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}{\partial w_{s_i,a_j}}$$

$$-\frac{\partial L}{\partial w_{s_i,a_j}} = \delta \frac{\partial Q(s_t, a_t)}{\partial w_{s_i,a_j}}$$

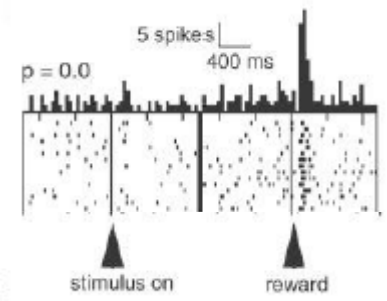
$$\Delta w_{s_i,a_j} = \delta * x_{s_i} * x_{a_j}$$

Hebbian weight update

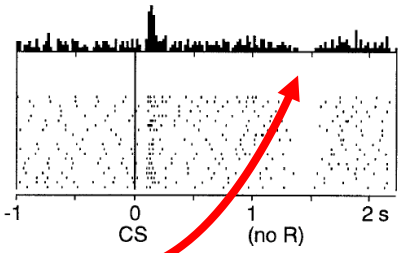
$$\Delta w_{i,j} = x_{pre} * x_{post}$$

The x_{a_j} comes from the fact that you're updating after an action choice and so that is already active.

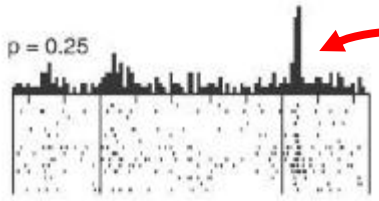
Dopamine neurons respond to prediction errors



reward following 0% predictive cue



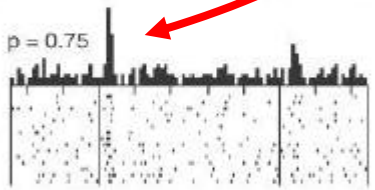
no reward following 100% predictive cue

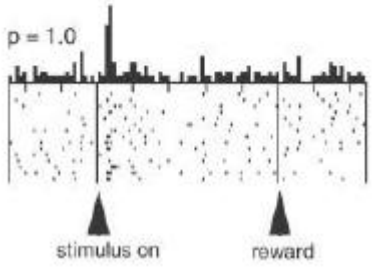


"Prediction Error"

$$\Delta V = R - V(S)$$

reward following 50% predictive cue



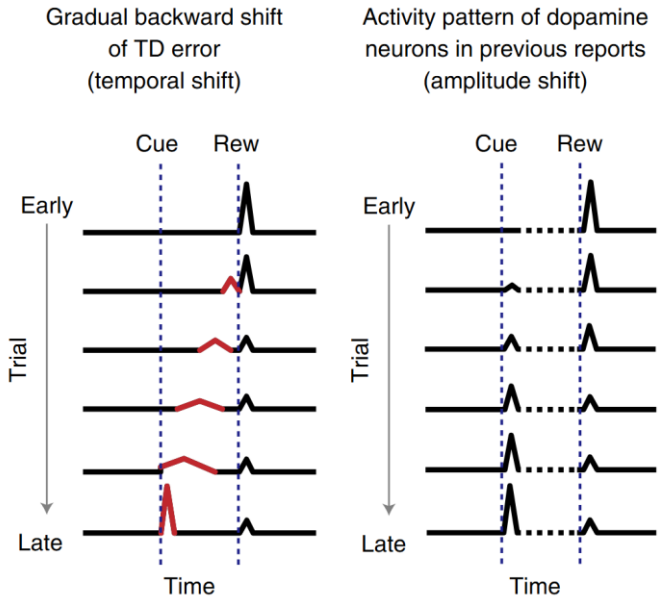


reward following 100% predictive cue

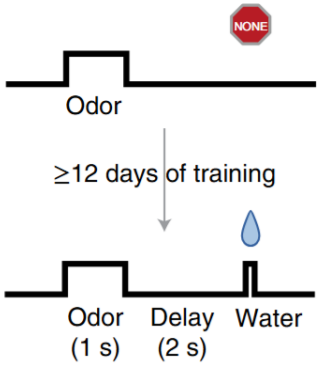
Schultz et al. 1997; Fiorillo et al, 2003

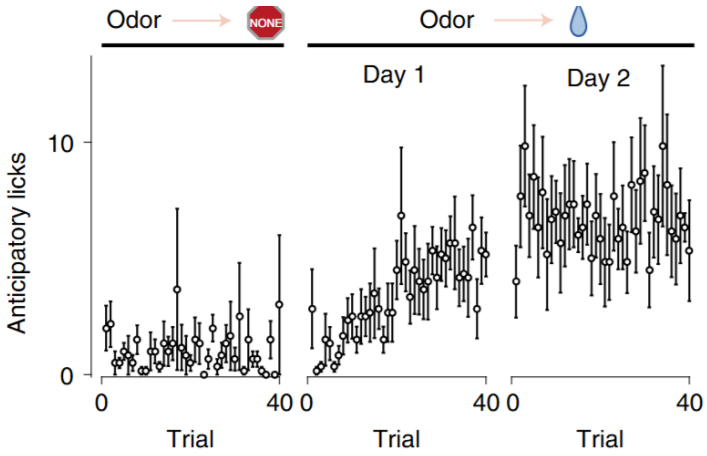
Evidence for temporal difference learning

TD predicts gradual flow backwards, but DA neurons often show a jump.



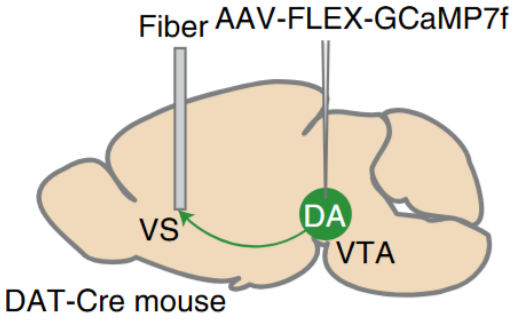
Mice were trained to perform classical conditioning task.

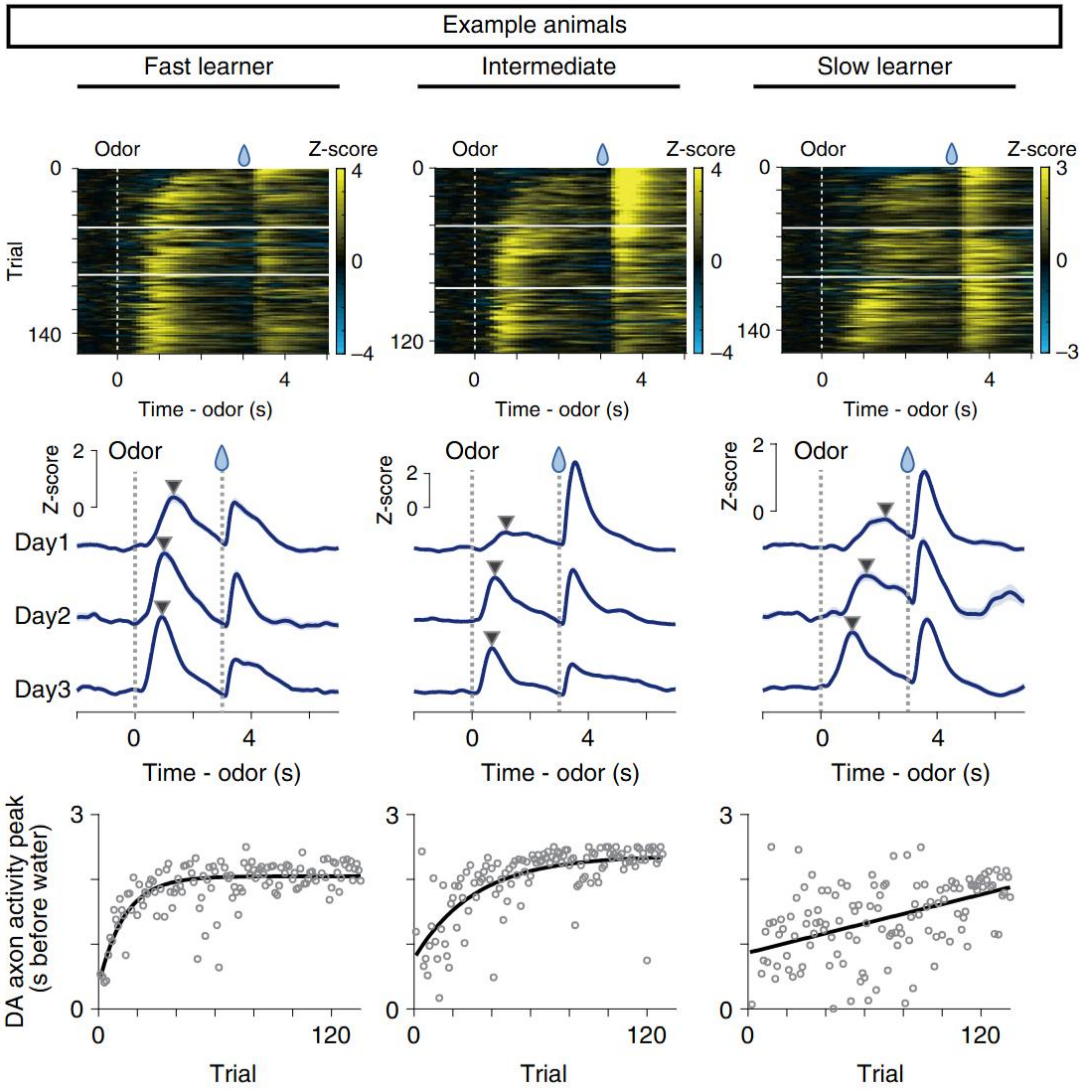




Evidence for temporal difference learning

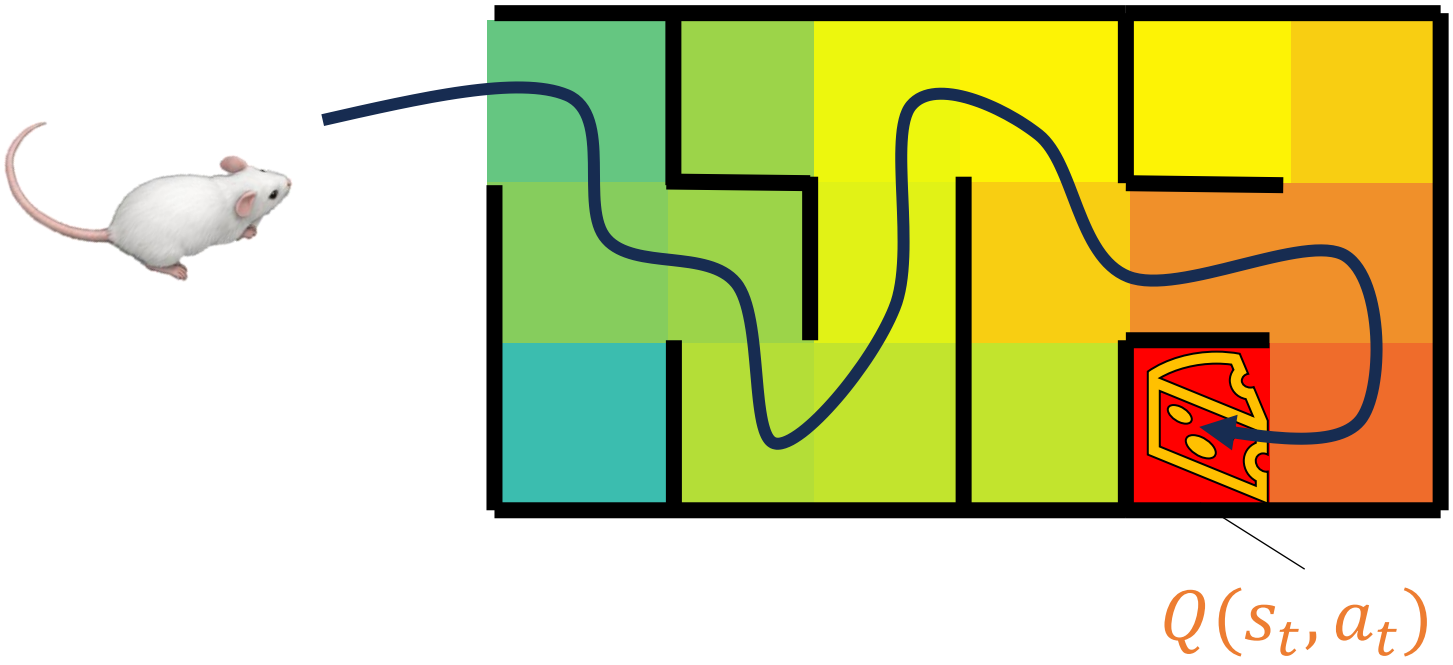
Imaging of DA terminals from VTA into the striatum.





Temporal difference learning for action planning…

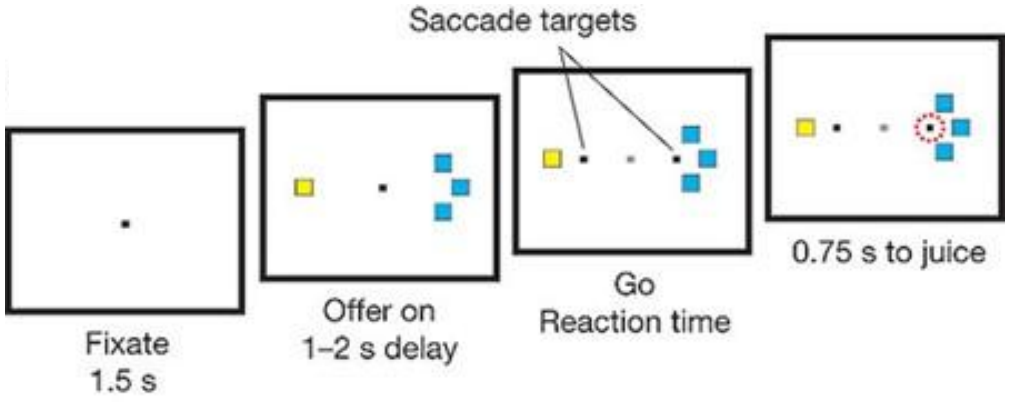
By iteratively passing rewards backwards in time through states and actions, one can eventually build a map of values that can be used to guide behavior.



How do you make a decision between different types of choices?

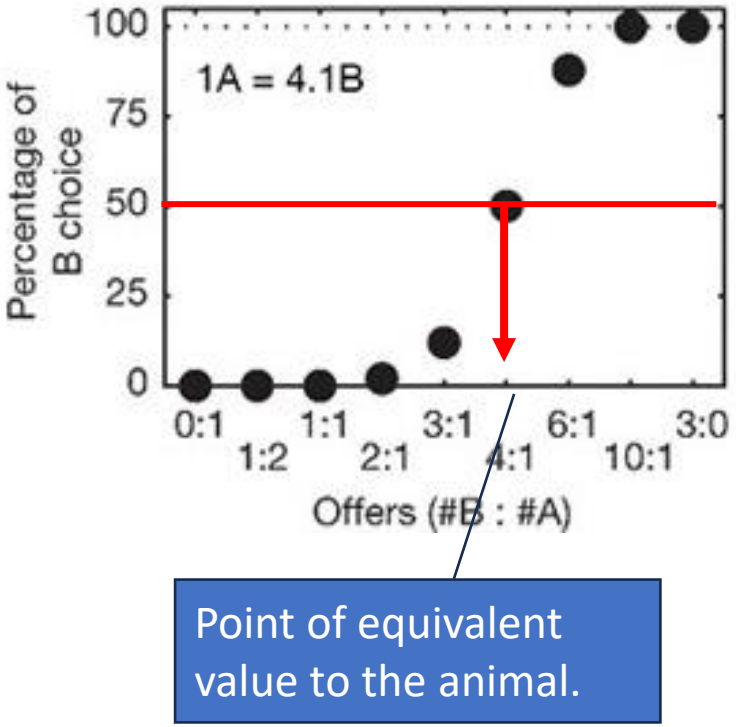
Neurons encode the value of a stimulus, regardless of the media of the reward.

Monkeys were allowed to choose between two options for rewards.



(favorite)

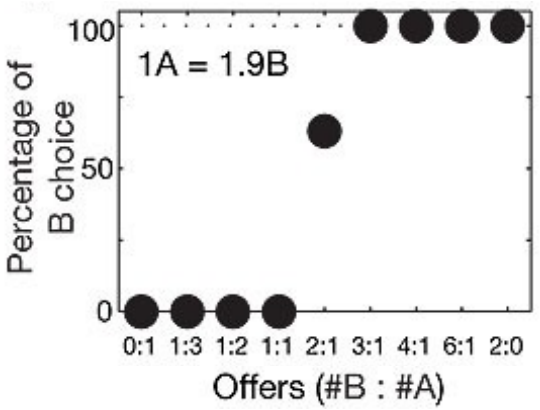
(least liked)

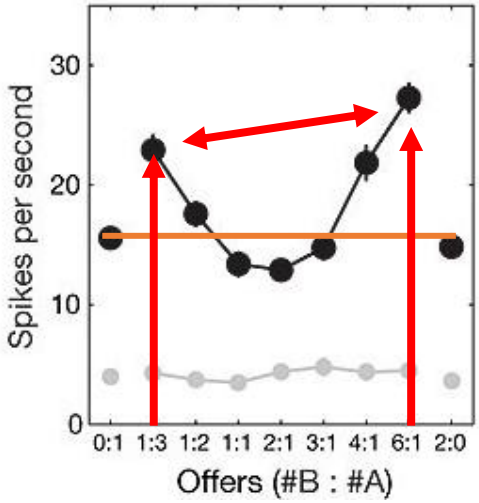


Padoa-Schioppa and Assad, 2006

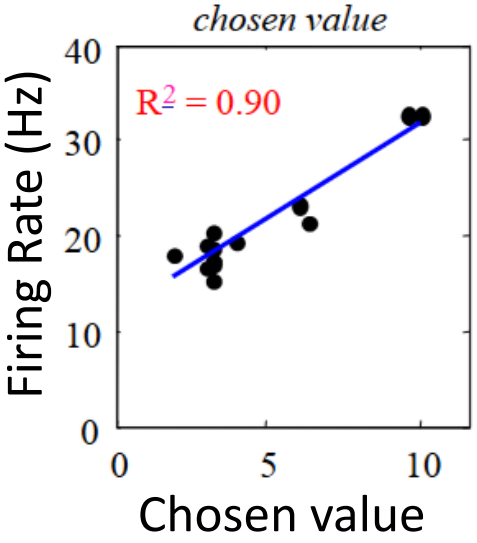
Neurons encode the value of a stimulus, regardless of the media of the reward.

Neurons in the Orbitofrontal Cortex (OFC) responded to the chosen value, regardless of the exact reward given.



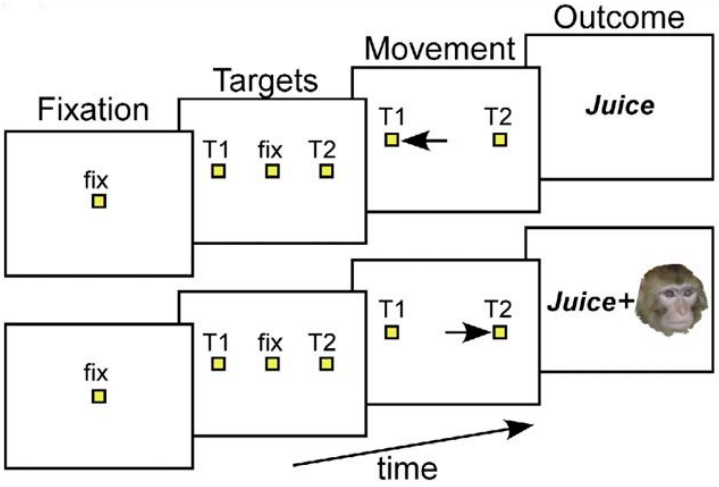


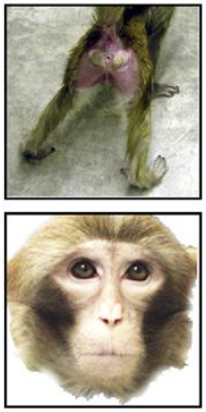
1 A ≈ 2B

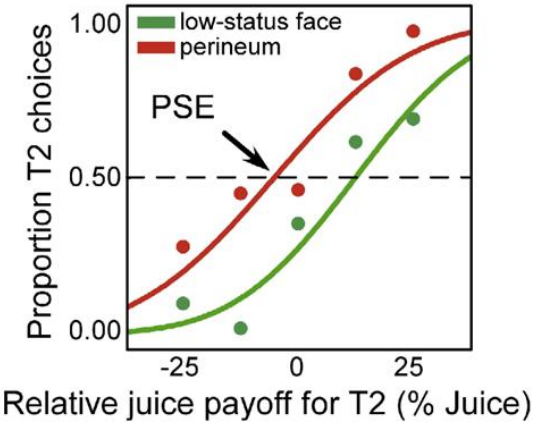


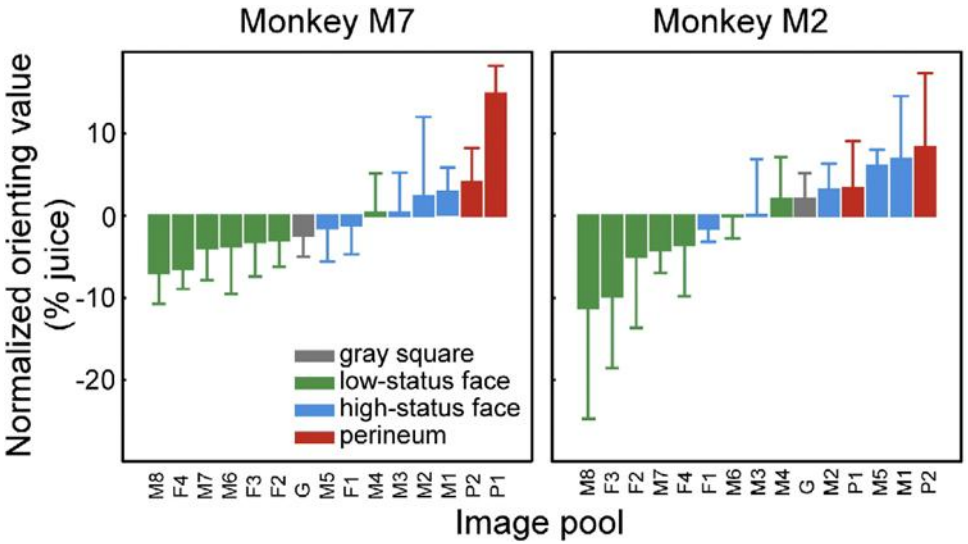
Padoa-Schioppa and Assad, 2006

Juice equivalency for monkey photos







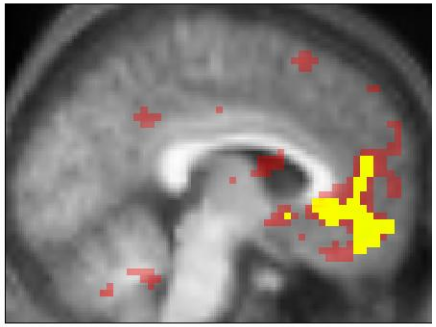


Monkeys will sacrifice juice to look at socially rewarding images and need to be paid juice to look at uninteresting photos.

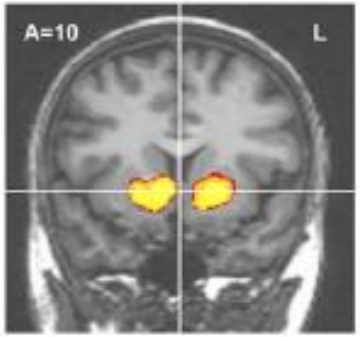
Deaner et al, 2005

Broad findings

Reward or reward anticipation activates ventromedial prefrontal cortex & orbitofrontal cortex, striatum (sometimes midbrain)

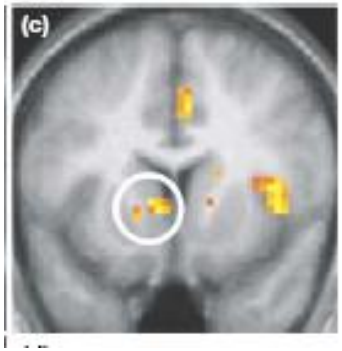


money
value predicted
(Daw et al 2006)



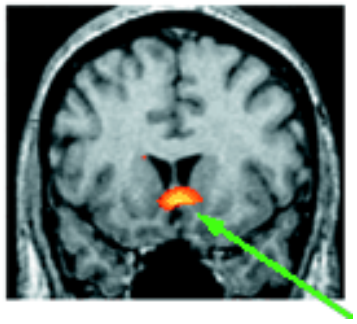
money
gain vs loss
(Kuhnen & Knutson 2005)

faces
attractiveness
(O'Doherty et al 2003)



food odors
valued vs devalued
(Gottfreid et al 2003)

Coke or Pepsi
degree favored
(McClure et al. 2004)

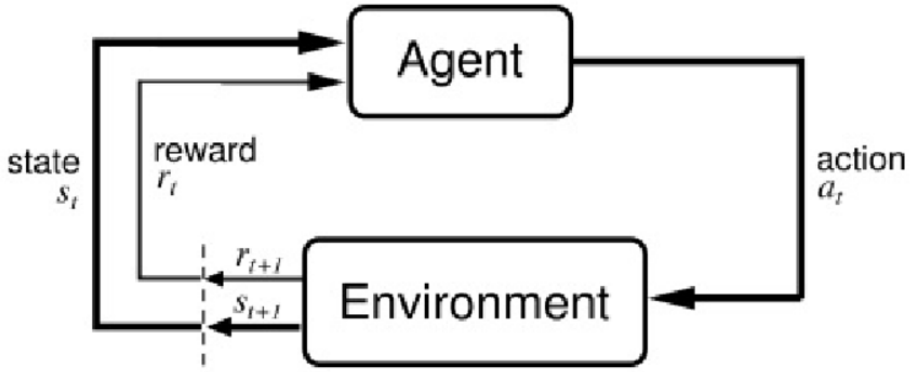


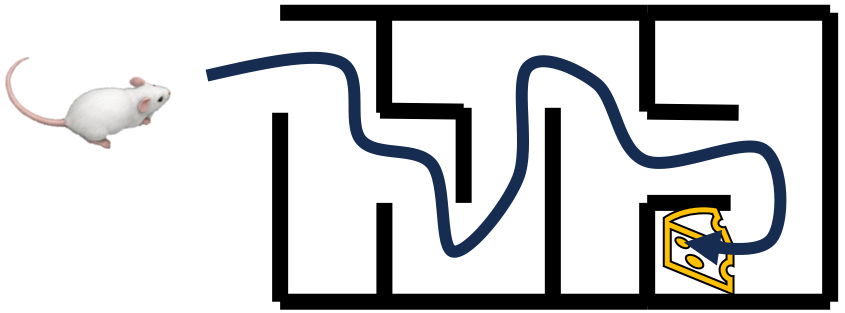
juice
unpredictable vs predictable
(Berns et al 2001)

→ commonality of responding across reinforcers suggests generalized appetitive function

Behavioral policies define how one should act in a given state

Agents act on the world to achieve a desired outcome.



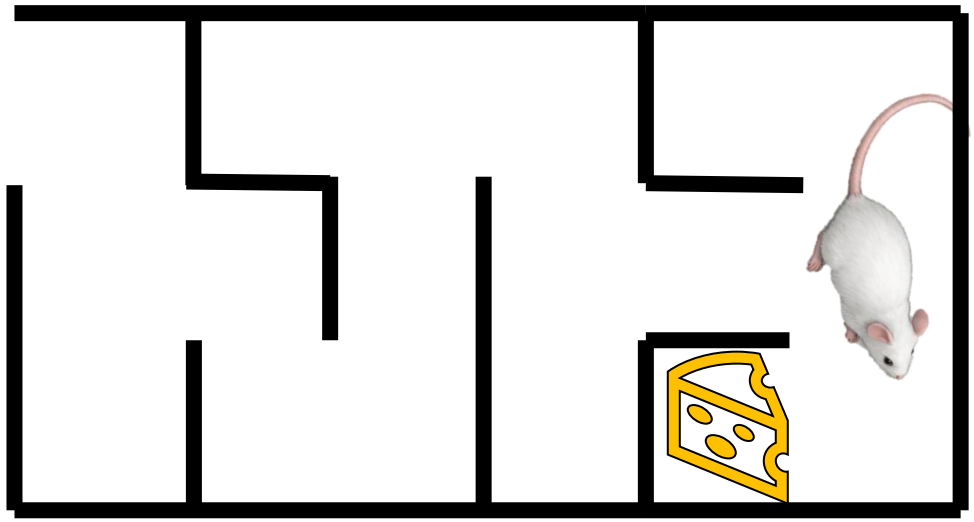


$$Q(s_t, a_t) = r(s_t) + \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) r(s_{t+1})$$

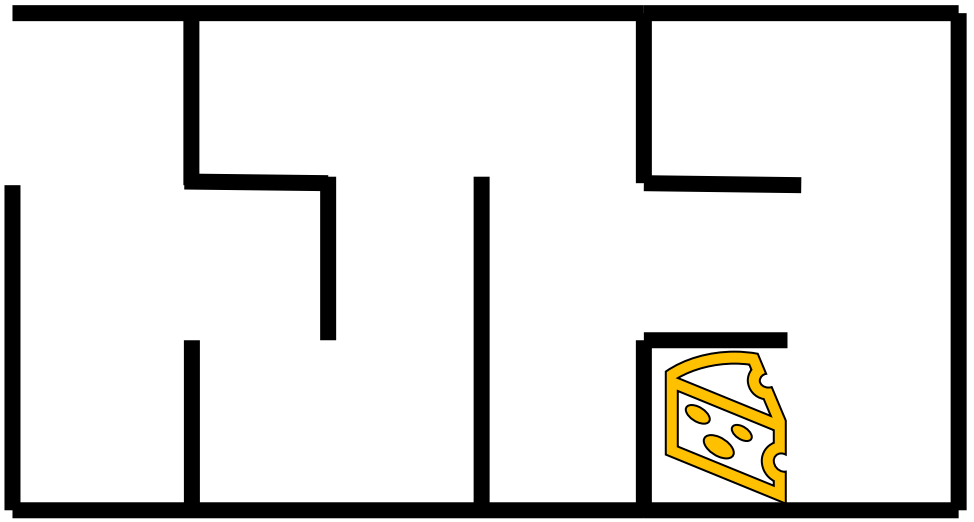
But, this is recursive! Your choices lead to a change in the state of the world, which leads to more choices to be made, etc.

So, you want to **choose a behavioral policy that maximizes total reward**.

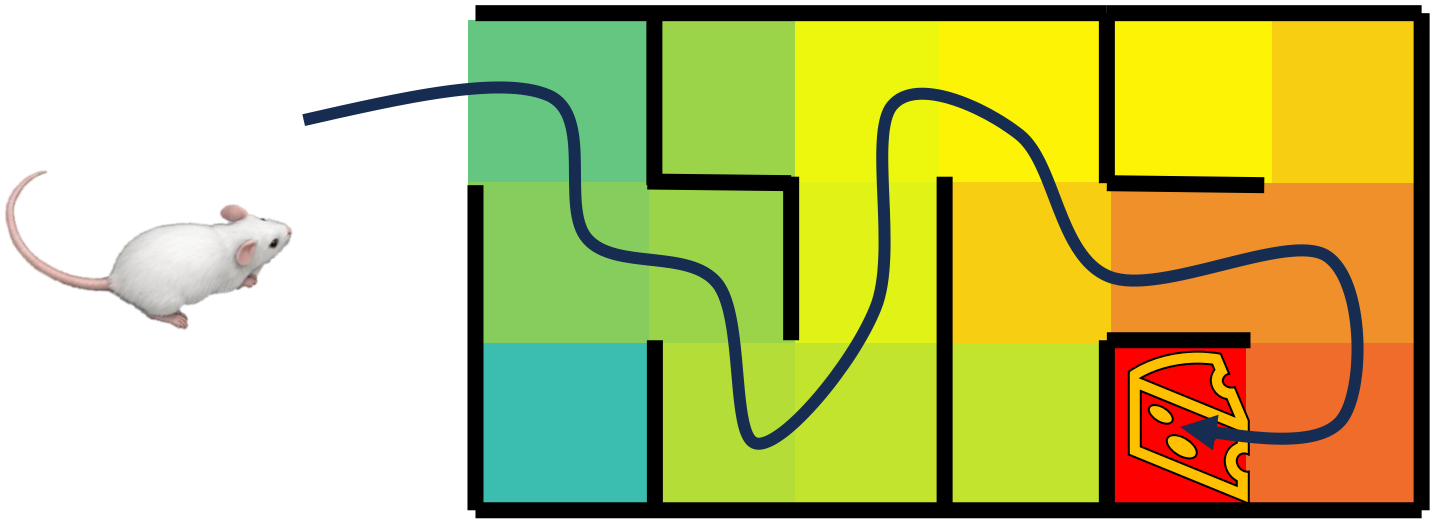
Predict rewards to facilitate behavior



Predict rewards to facilitate behavior

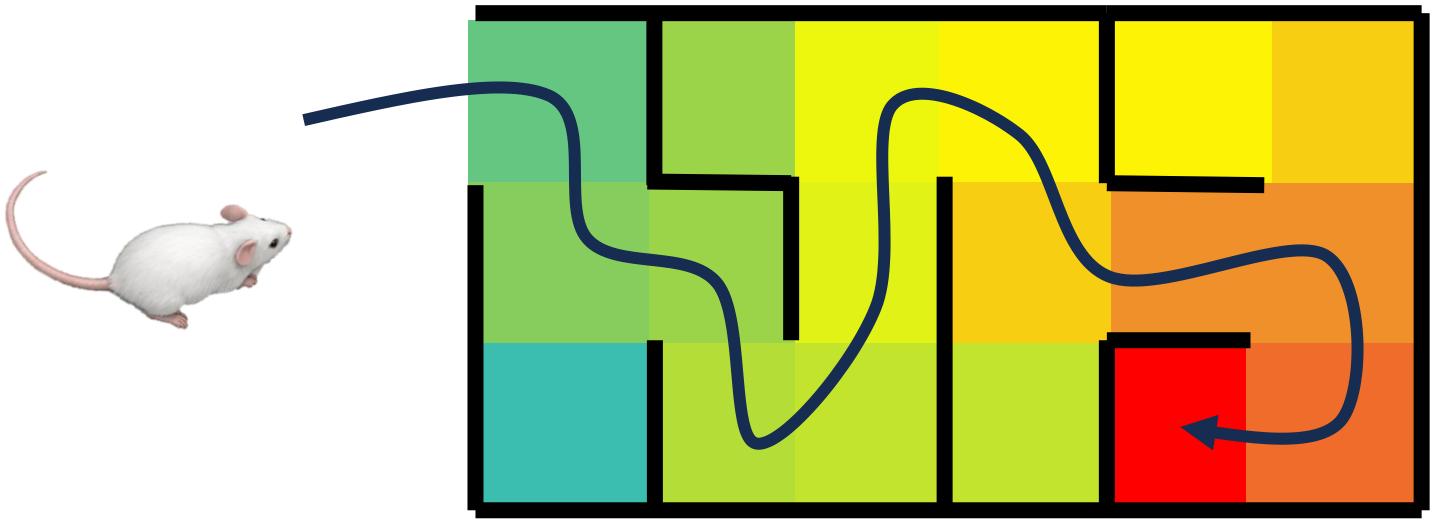


Predict rewards to facilitate behavior



Predict rewards to facilitate behavior

Predict rewards to facilitate behavior



Summary

- Predictions are central to behavior
  - Predictions allow one to out-compete other organisms for resources and escape from predation
- Predictions rely on “predictive” learning – you get feedback from the world simply by waiting in time.
  - This simple mechanism seems to be sufficient for learning complex behaviors, such as language
  - Models that only use semi-supervised learning to learn language have strong homology with neural responses in the brain.
  - Depth of language model may map onto cortical hierarchy.
- Visual predictions compensate for sparse information.
  - Interactions between brain regions allow for completion of ambiguous information.
  - Illusory contours and filling-in is observed in intermediate visual areas (V2 through V4).
  - This is thought to be due to a predictive coding framework where top-down contextual information interacts with bottom-up sensory drive.
- Rotation of neural representations may help avoid interference between sensory and memory representations.