Learning

- Unsupervised Learning
  - Hebbian Learning Rule
  - Pattern associator
  - Self-organized maps
  - Topographic structure
  - Pattern detectors
- Supervised Learning
  - Scalar (Reinforcement) Learning
    - ☞ Classical and Instrumental Conditioning
    - Sequential learning and Prediction
  - Vector-Based Learning
    - Generalized Delta Rule
    - Backpropagation
    - Deep Learning

Reinforcement Learning

- Conditioning
  - Simple Prediction
    Rescorla-Wagner Rule
  - Stimulus-Action Associations
    Actor-critic model, Q Learning
- Sequence Prediction
  - Method of Temporal Differences (TD)
  - Model-Free vs. Model-Based RL
- Challenges
  - Curse of dimensionality
    - *Hierarchical RL: policies and options*
    - *State space abstraction*
  - Explore-exploit
    - ***Meta-control***

Conditioning

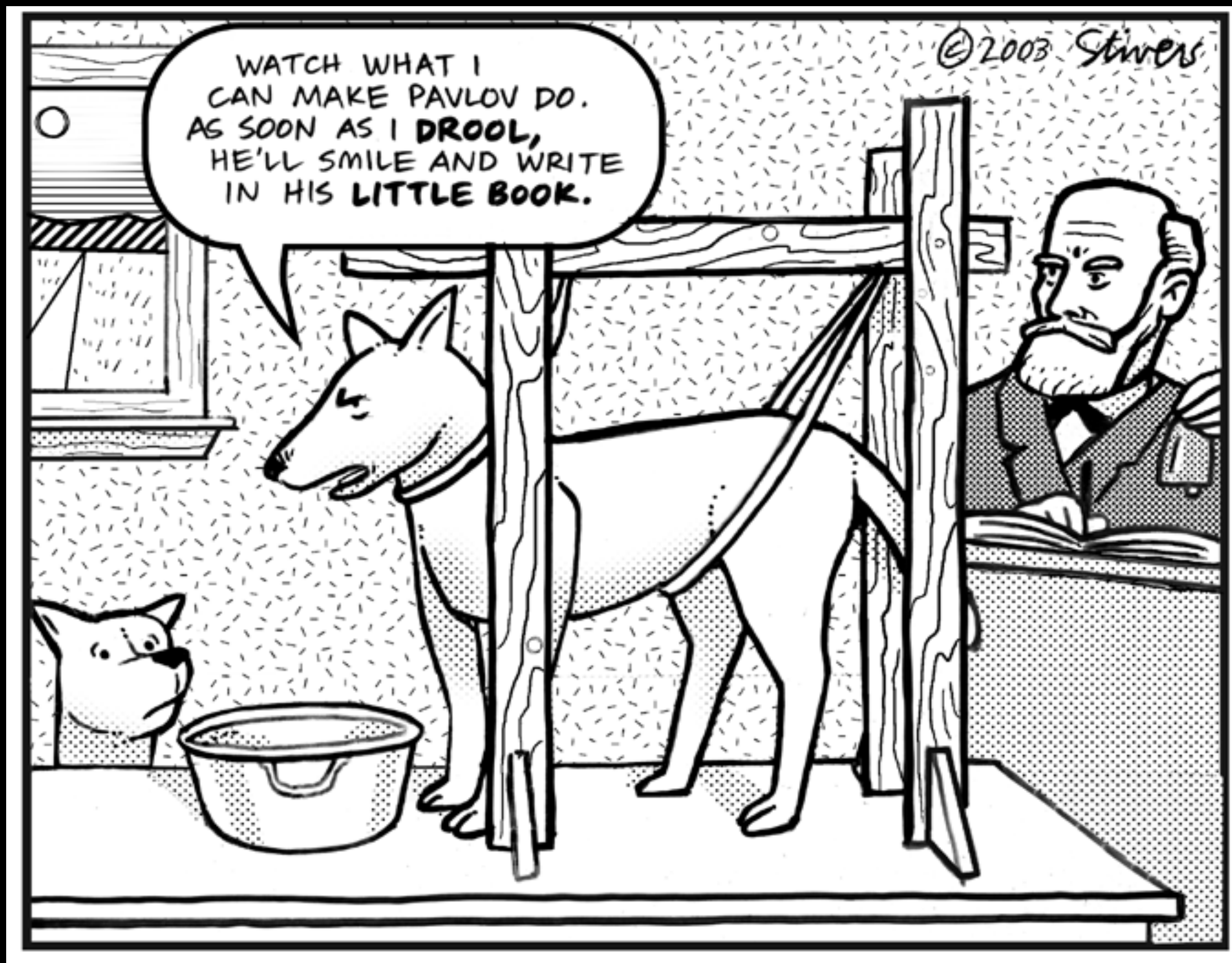
Conditioning

- Conditioning is “supervised” associational learning
  - Can be thought of as the experimenter controlling the environment
  - But the learning is still *associational*

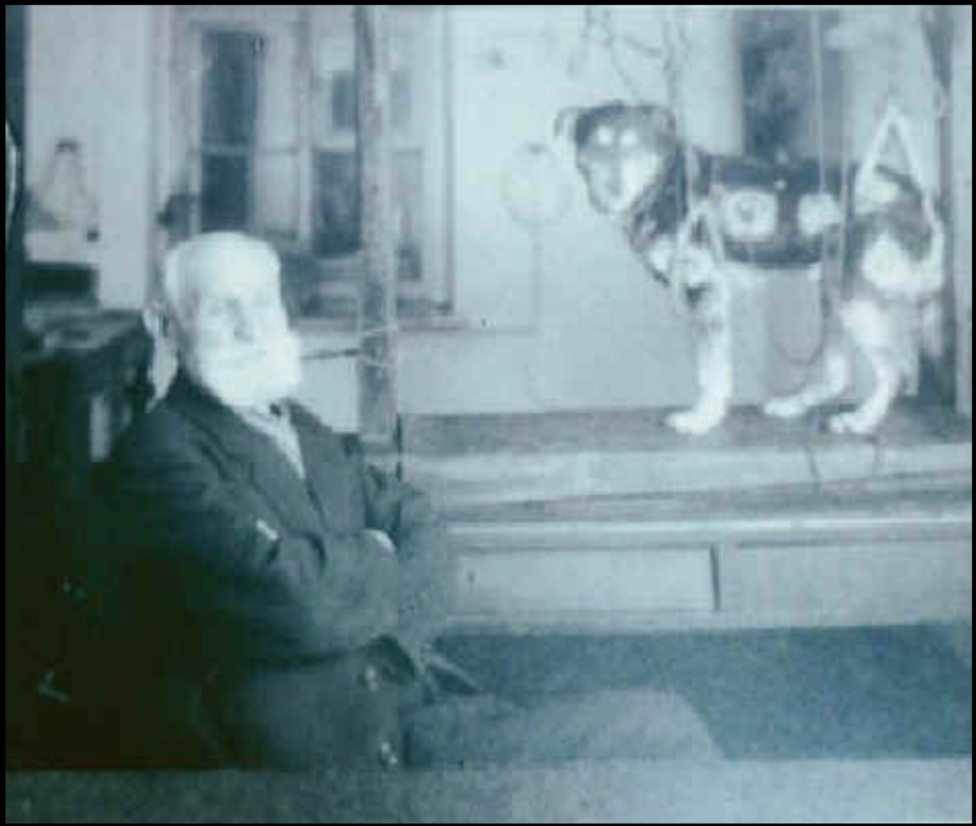
Conditioning

- Conditioning is “supervised” associational learning
  - Can be thought of as the experimenter controlling the environment
  - But the learning is still *associational*
- Based on similar principles of associative learning, but with a twist…

Conditioning

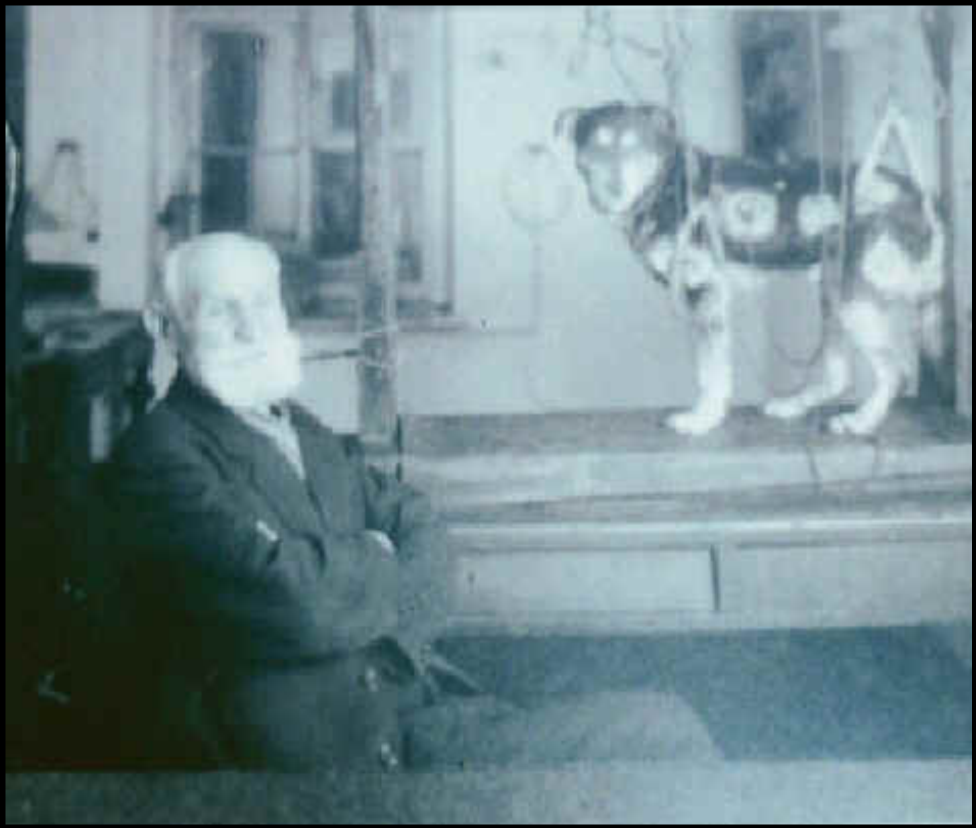


Classical (Pavlovian) Conditioning



pair stimulus (CS)

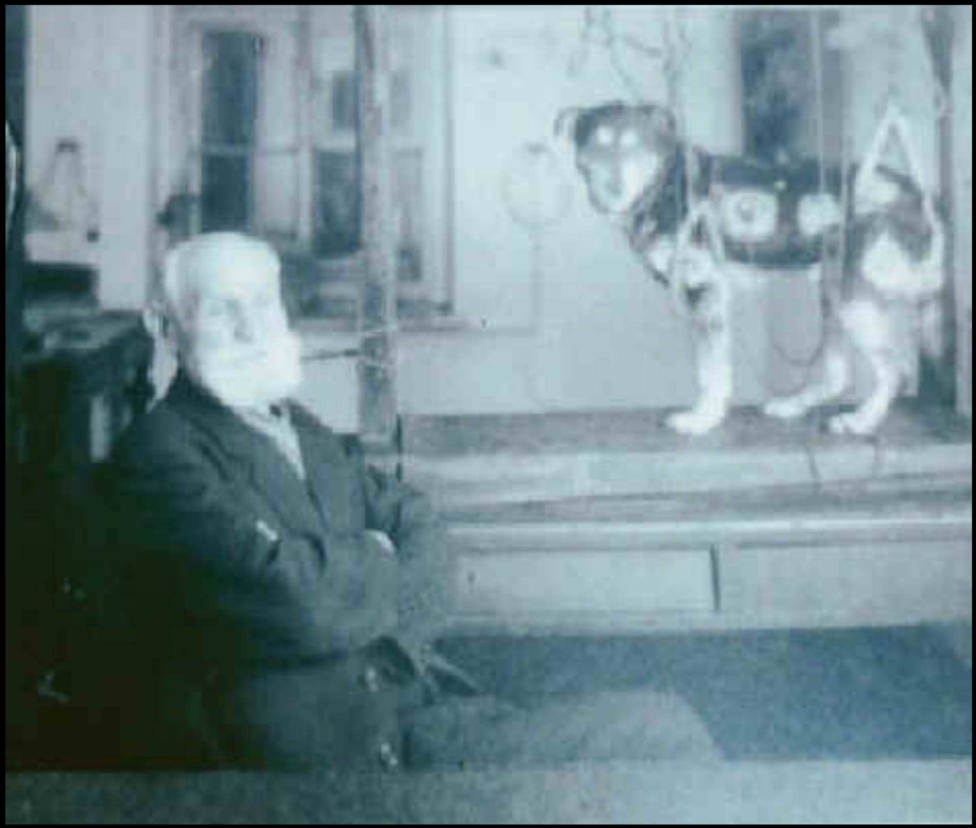
Classical (Pavlovian) Conditioning



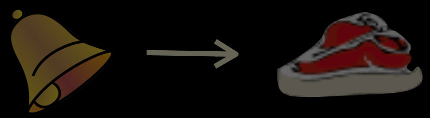
pair stimulus (CS)

...with significant event (US)

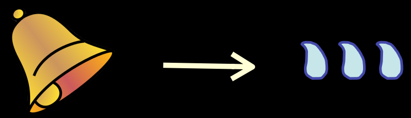
Classical (Pavlovian) Conditioning



pair stimulus (CS)
...with significant event (US)



measure anticipatory behavior (CR)



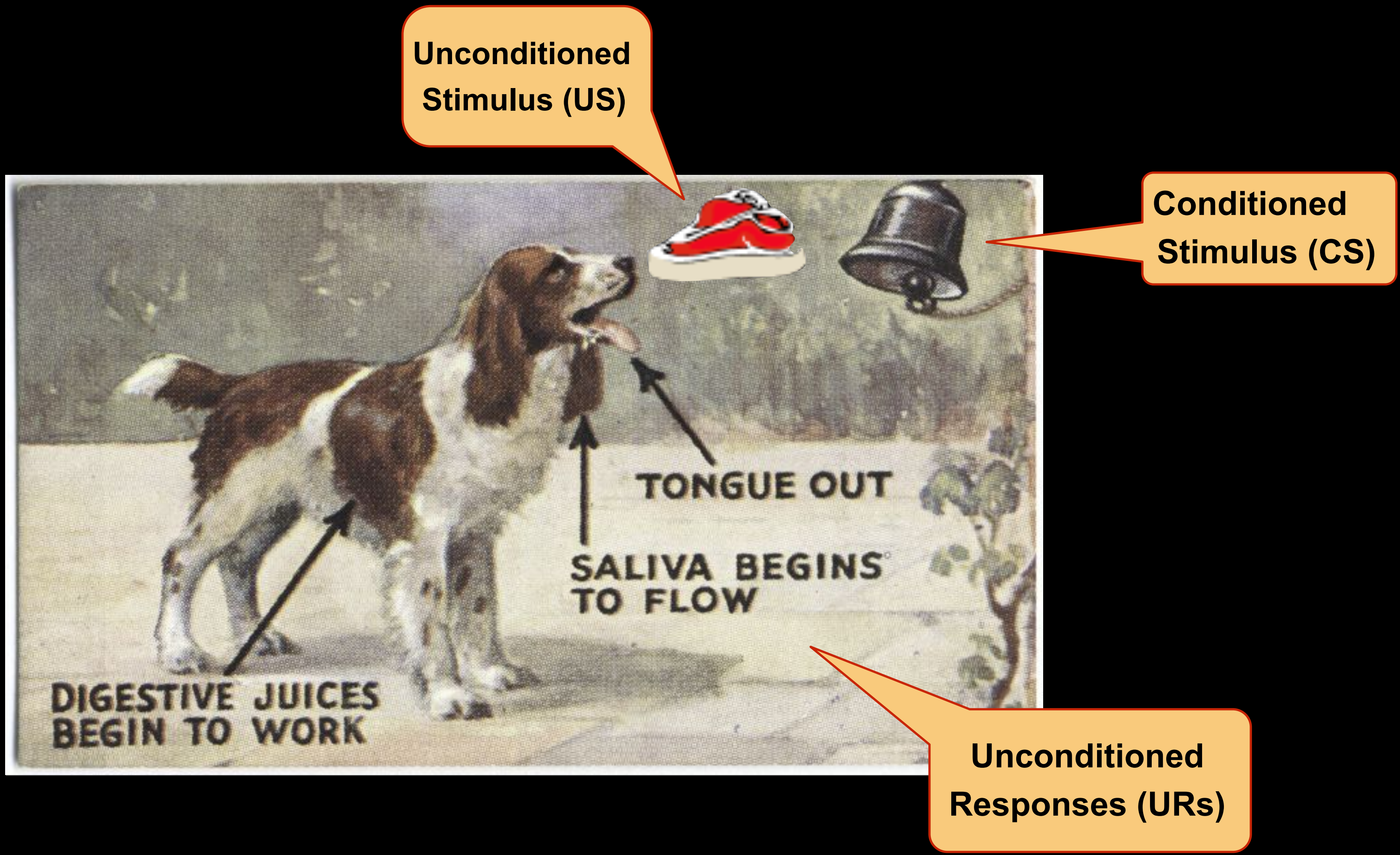
Classical (Pavlovian) Conditioning

Classical (Pavlovian) Conditioning

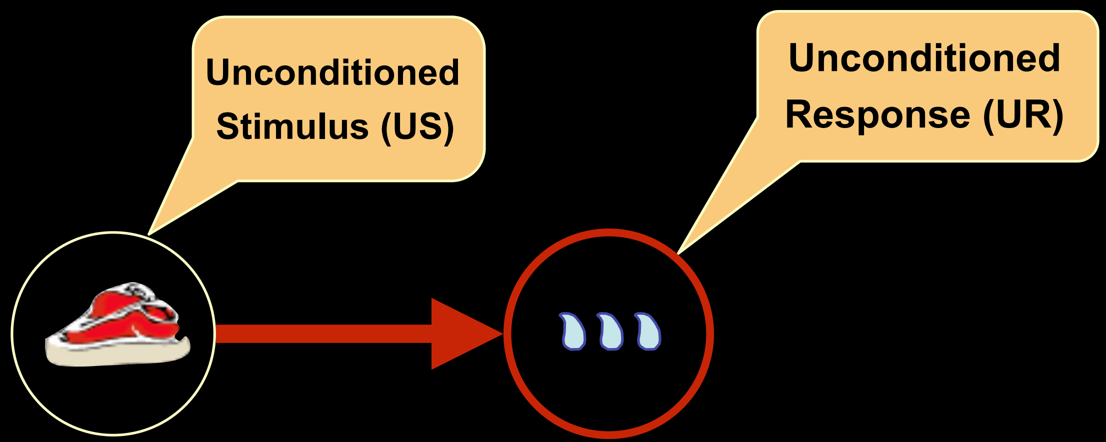
Conditioned Stimulus (CS)

Classical (Pavlovian) Conditioning

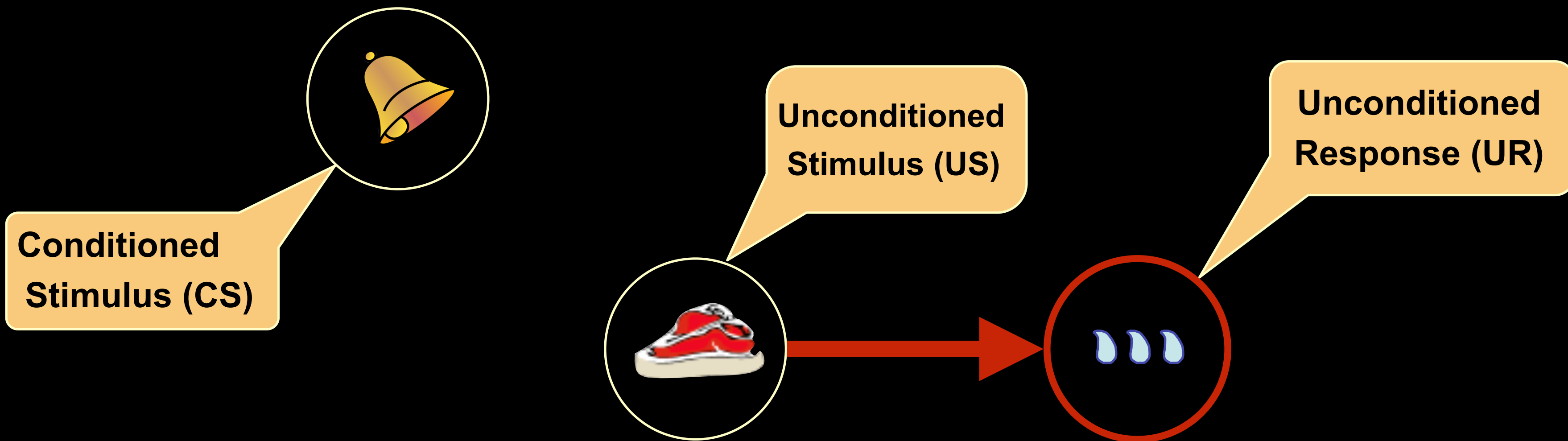
Classical (Pavlovian) Conditioning



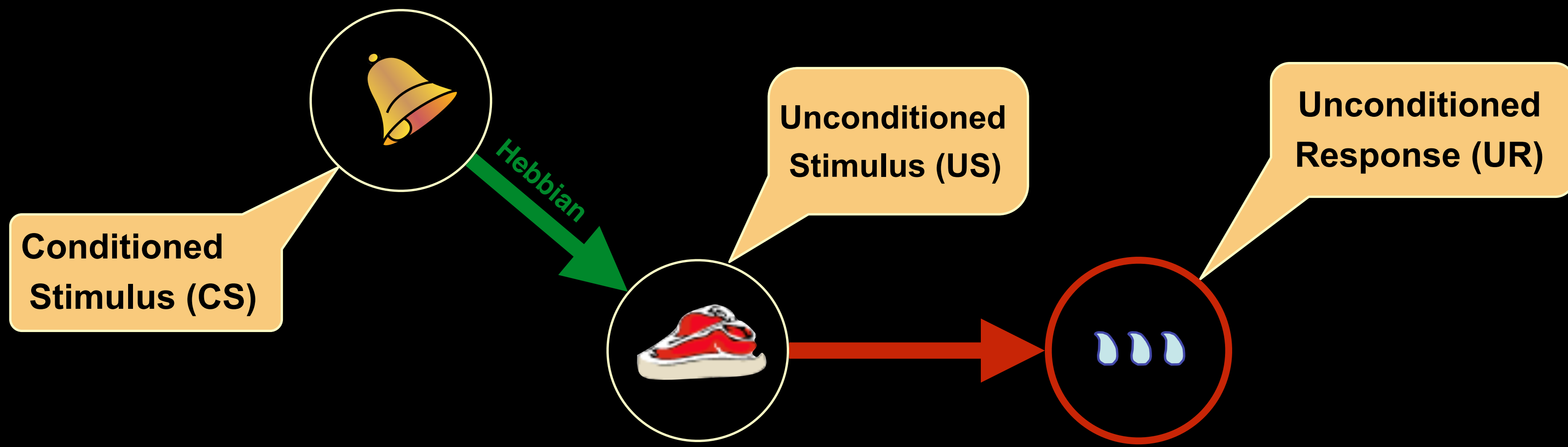
Conditioning



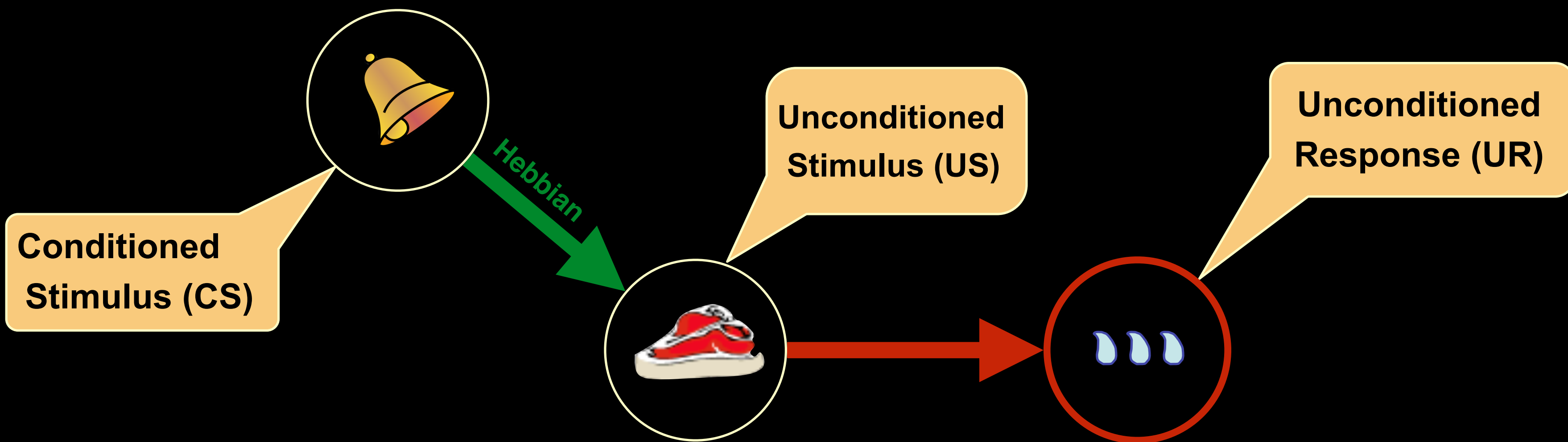
Conditioning



Conditioning

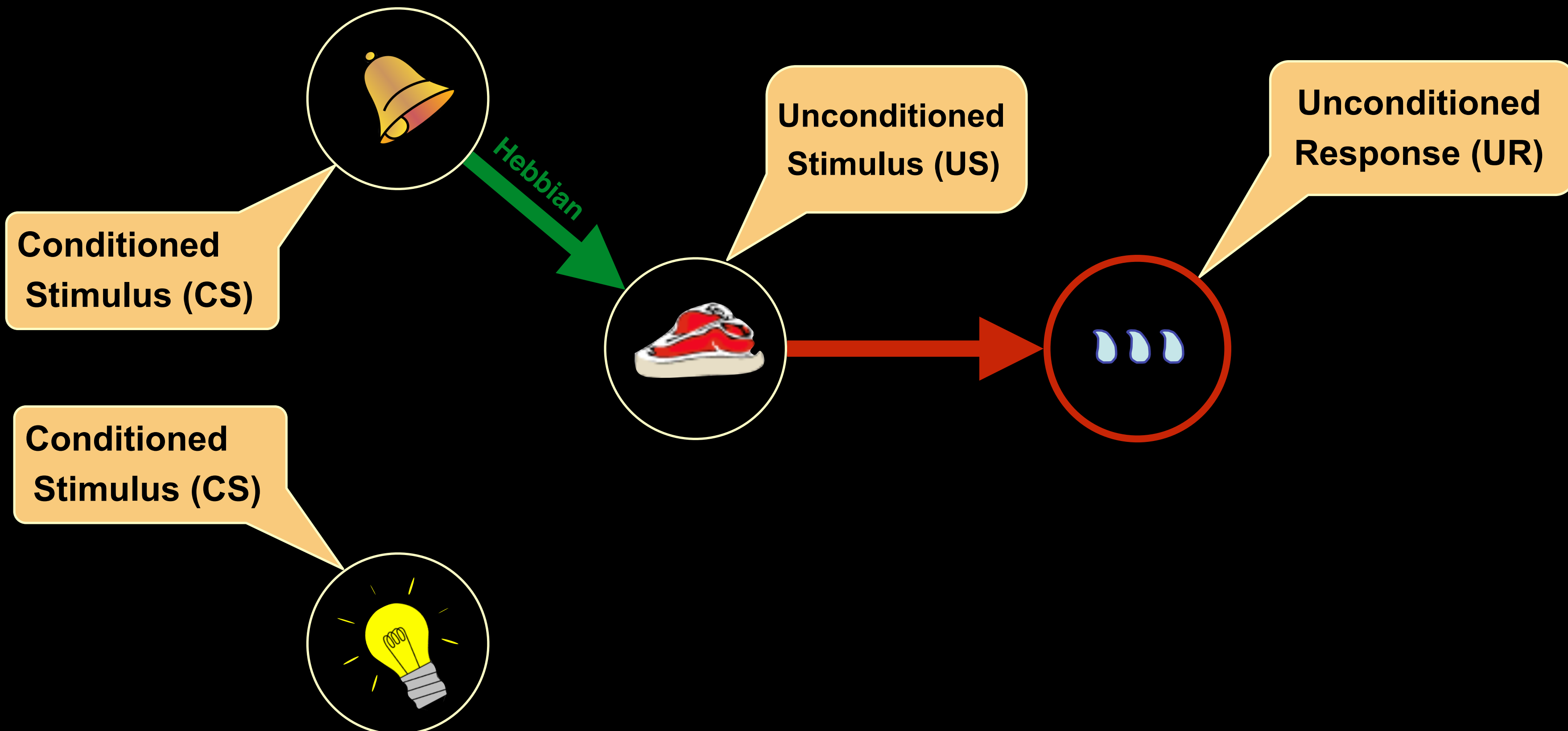


Conditioning



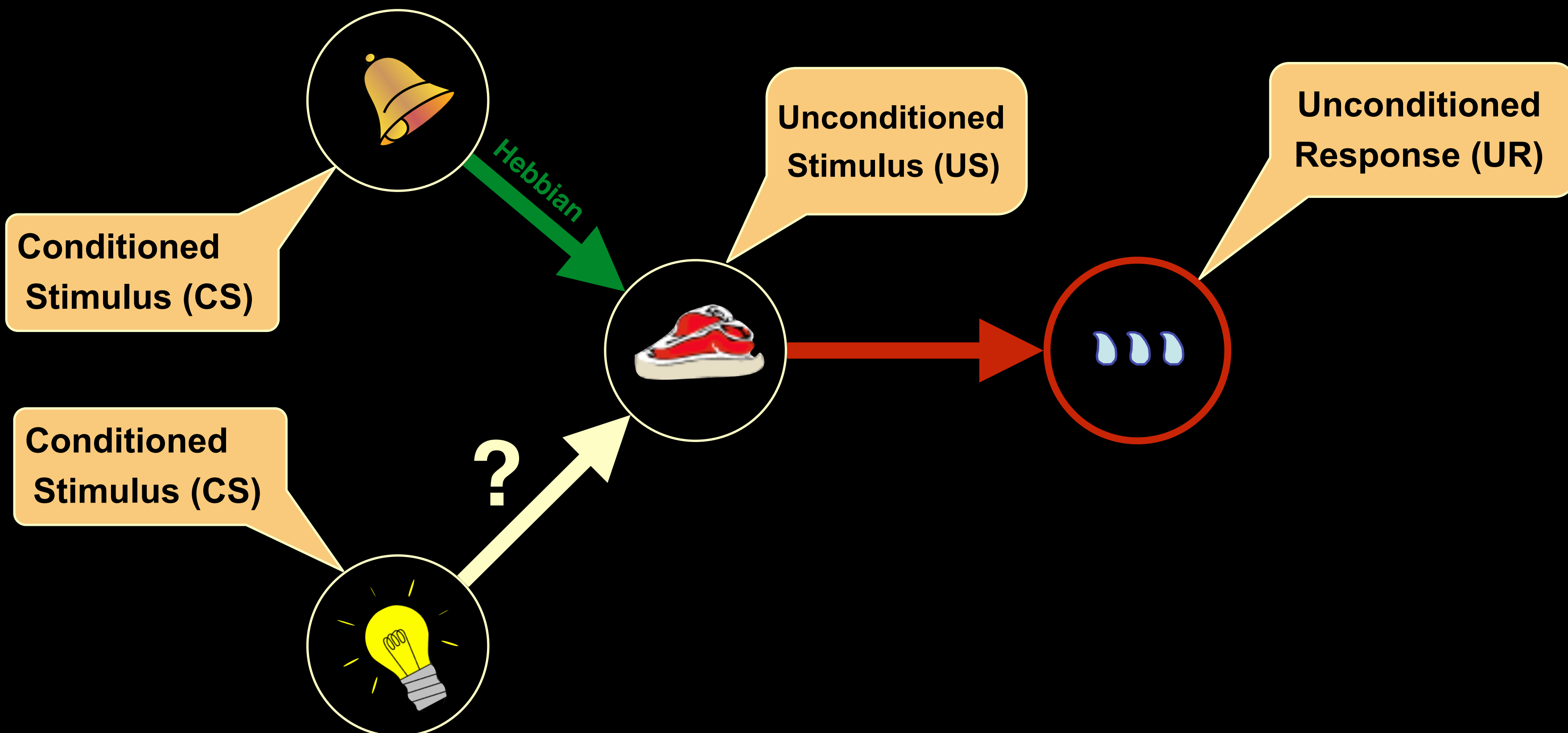
Pure Hebbian learning is purely about *contiguity*

Conditioning



Pure Hebbian learning is purely about *contiguity*

Conditioning

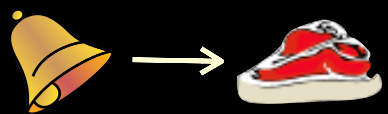


Pure Hebbian learning is purely about *contiguity*

Blocking

Kamin (1968)

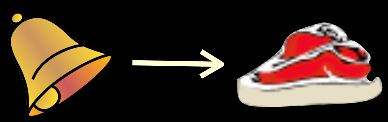
Phase I



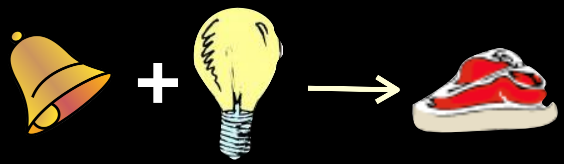
Blocking

Kamin (1968)

Phase I



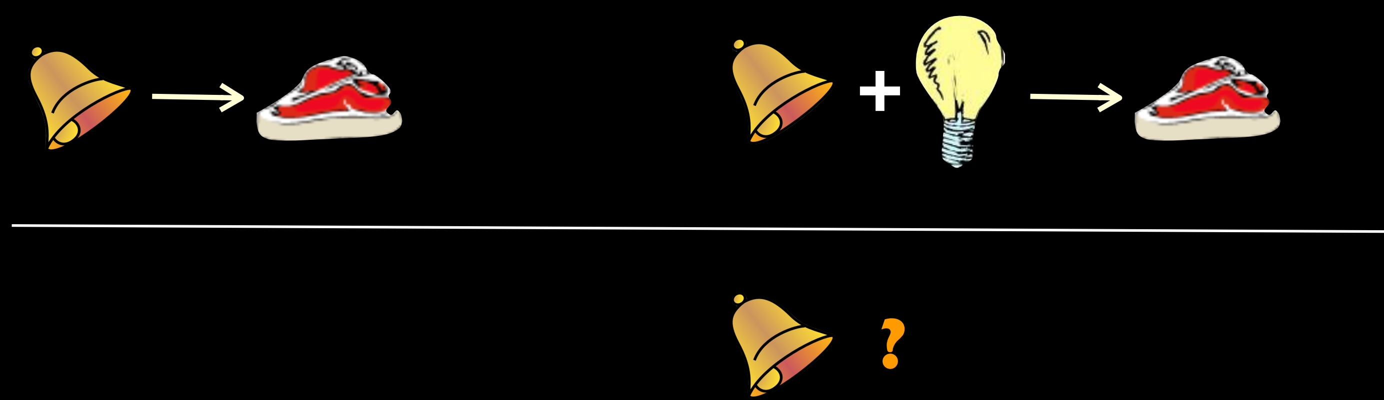
Phase II



Blocking

Kamin (1968)

Phase I **Phase II**

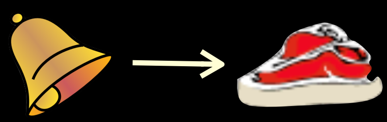


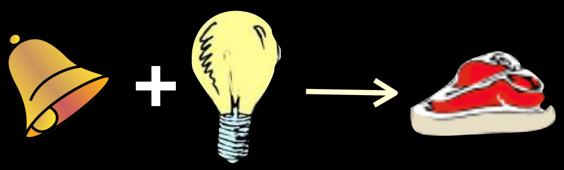
Blocking

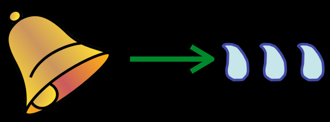
Kamin (1968)

Phase I

Phase II



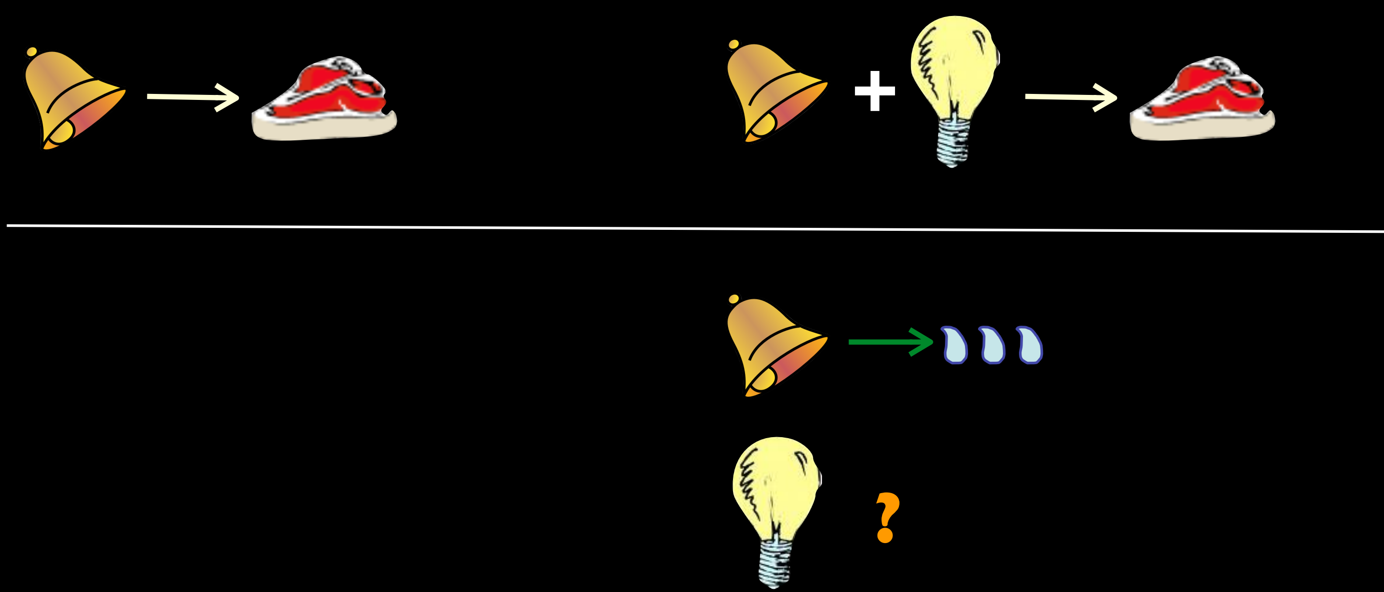




Blocking

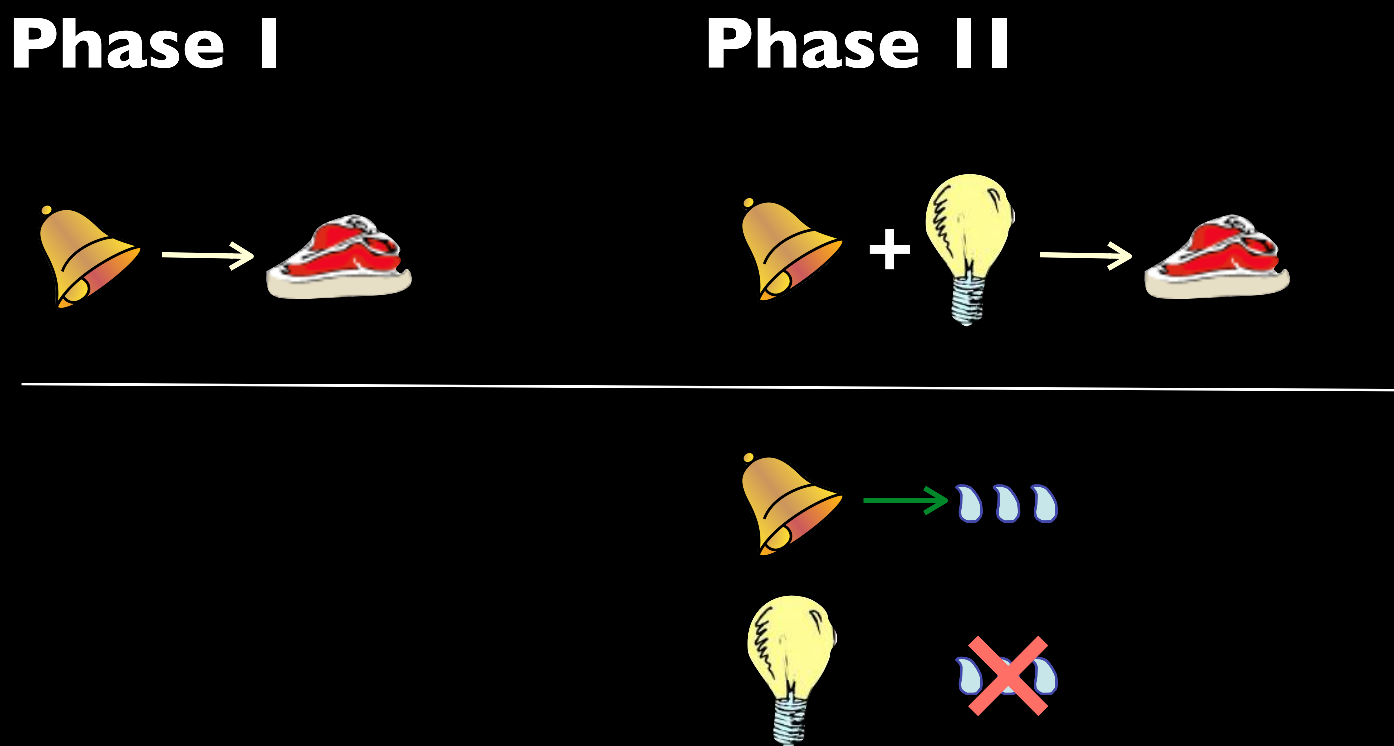
Kamin (1968)

Phase I **Phase II**

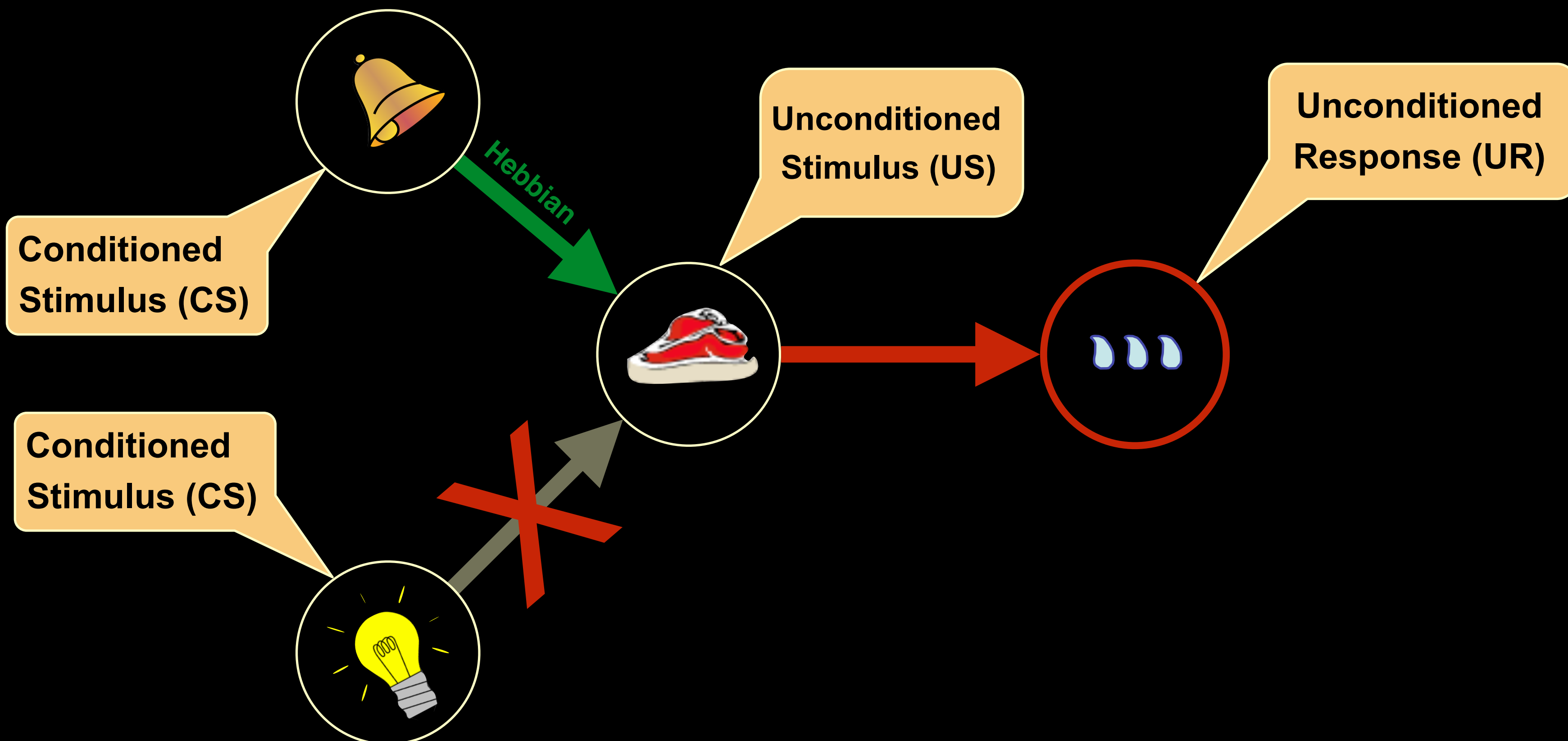


Blocking

Kamin (1968)



Blocking



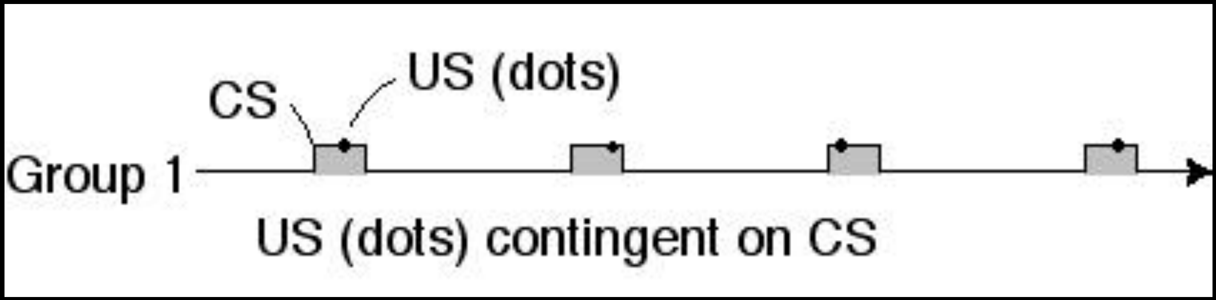
Conditioning must be about more than *just contiguity*…

Contingency

Contingency

Rescorla's experiment:

Standard

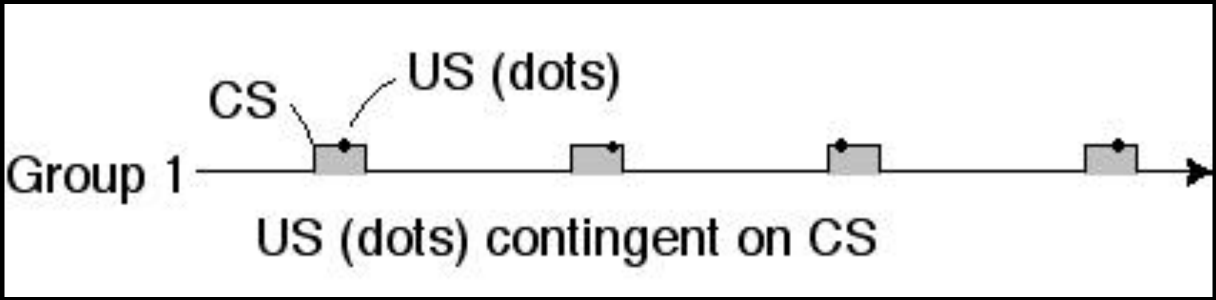


Contingency

Rescorla's experiment:

Conditioning:

Standard



√

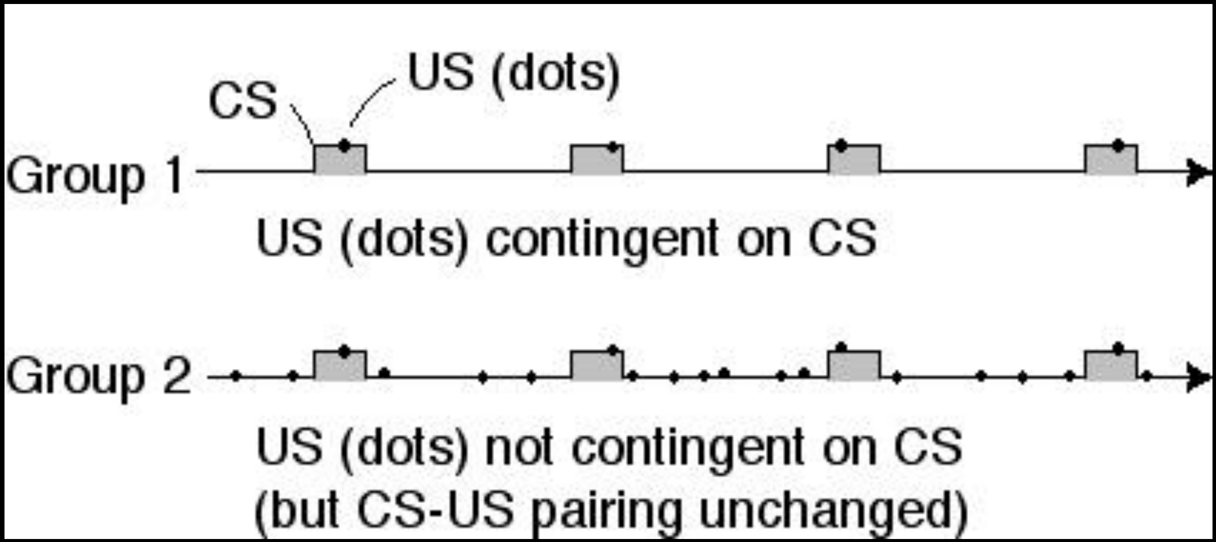
Contingency

Rescorla's experiment:

Conditioning:

Standard

Random control



√

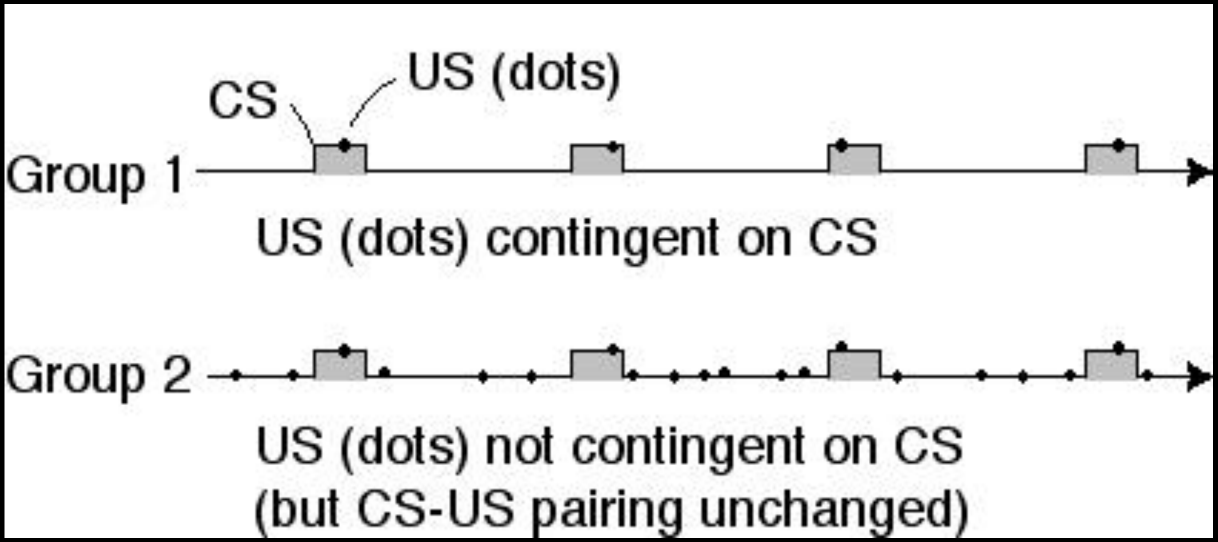
Contingency

Rescorla's experiment:

Conditioning:

Standard

Random control



√

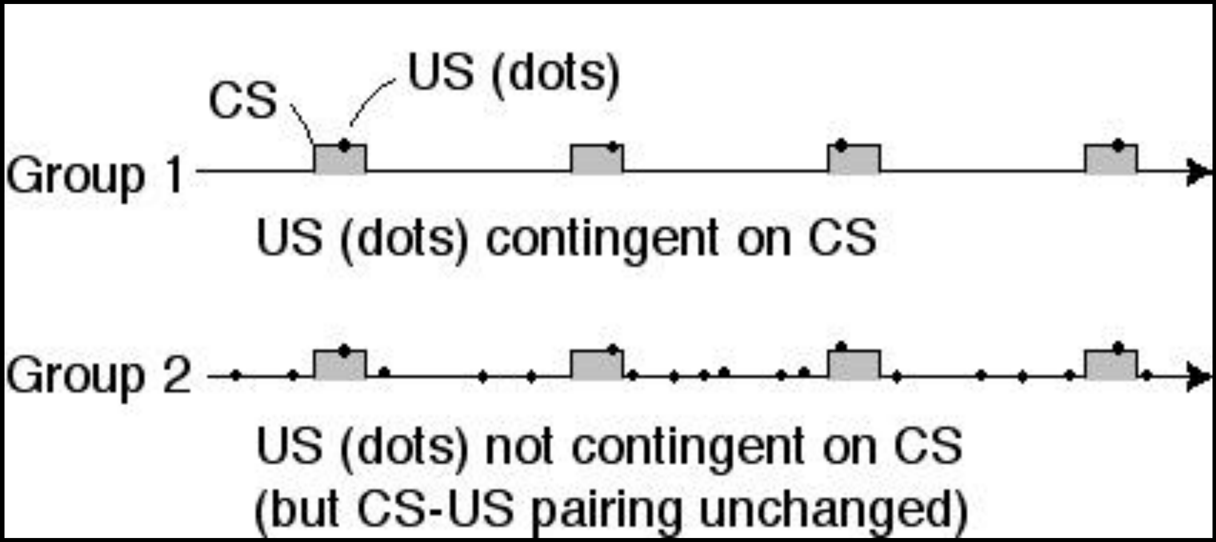
X

Contingency

Conditioning:

Standard

Random control



√

X

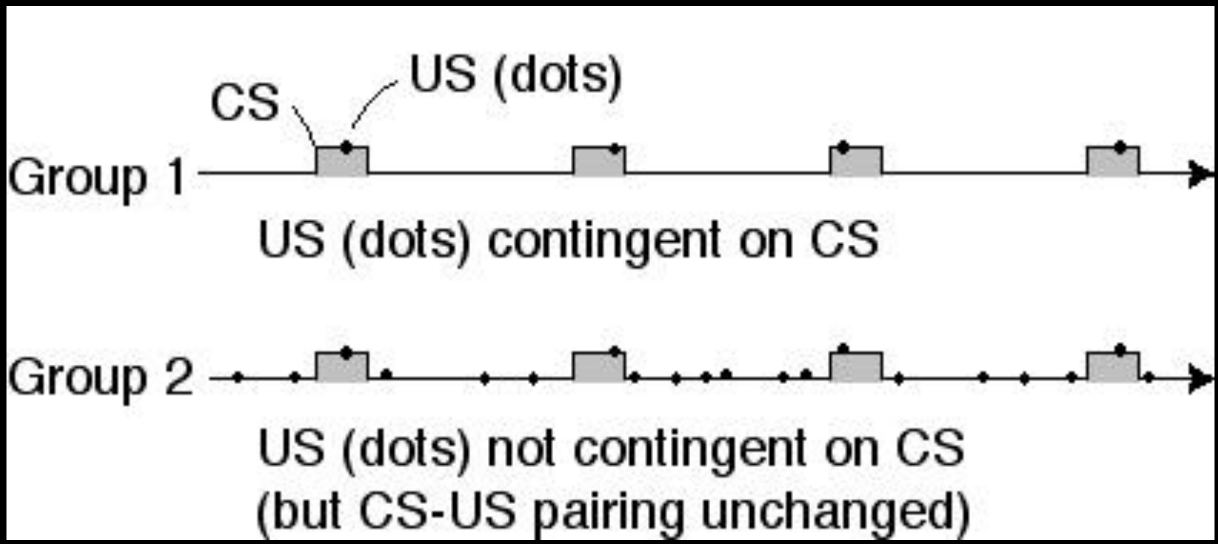
Temporal contiguity is not enough, need *contingency*

Contingency

Conditioning:

Standard

Random control



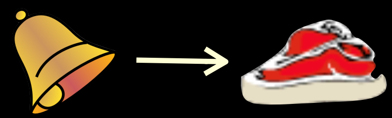
√

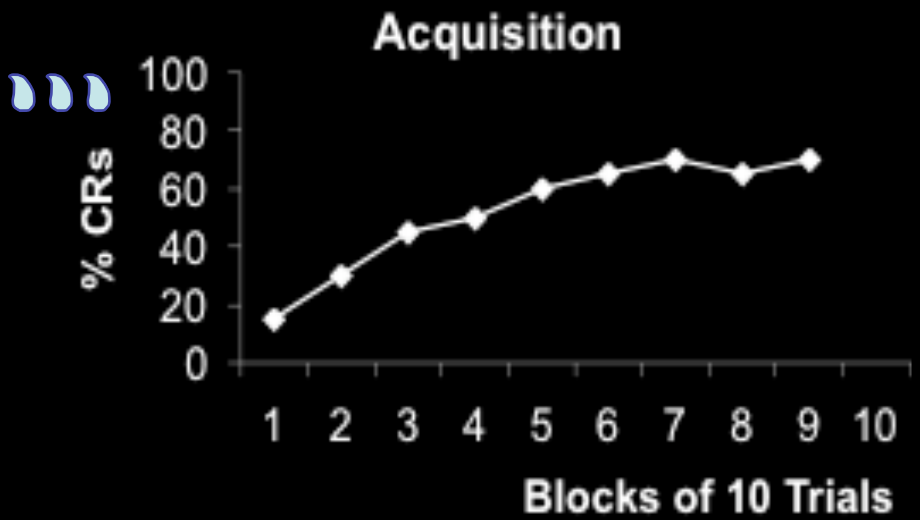
X

Contingency ⇒ *Prediction*

Contingency

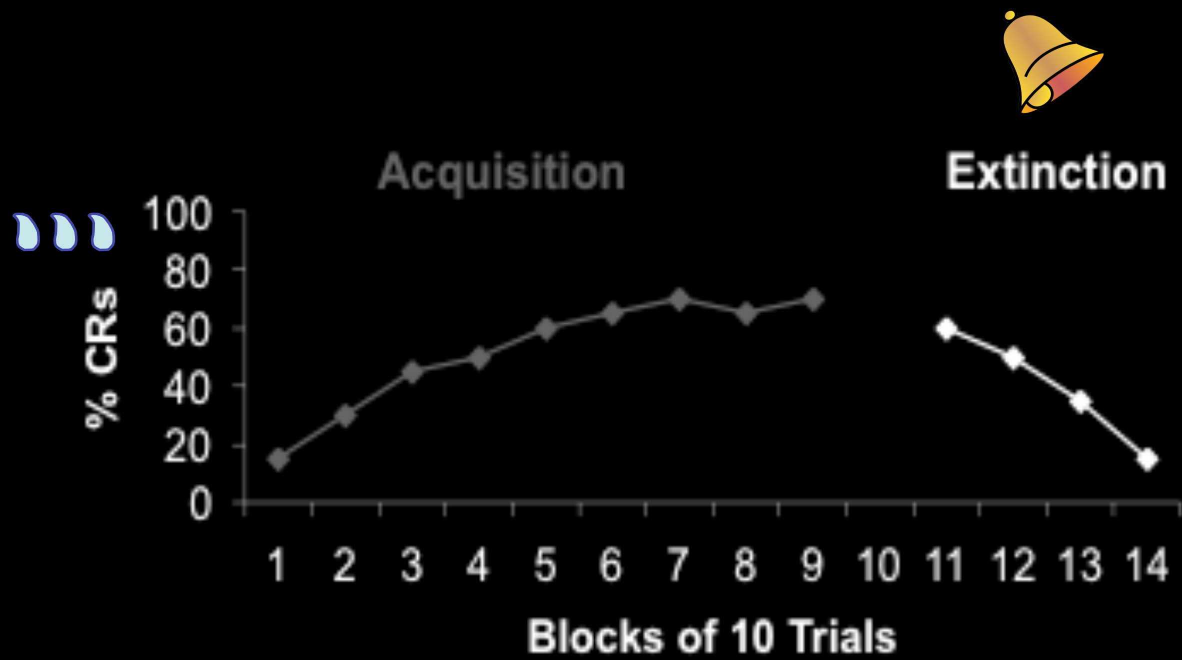
Prediction





Contingency

Failure of *Prediction*



Prediction Learning

Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as “delta rule,” or “reward prediction learning”)

Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

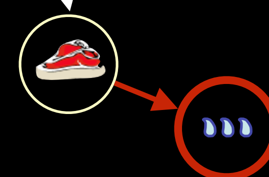
Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

Prediction error

$$\triangle v = r_t - v_t$$



where:

r_t *is the actual value of the CS*

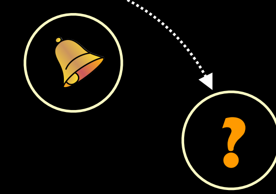
Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

Prediction error

$$\Delta v = r_t - v_t$$



where:

r_t is the actual value of the CS
v_t is the predicted value of the CS

Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

Prediction error

$$\Delta v = r_t - v_t$$

?

where:

r_t is the actual value of the CS
v_t is the predicted value of the CS
Δv is the reward prediction error

Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

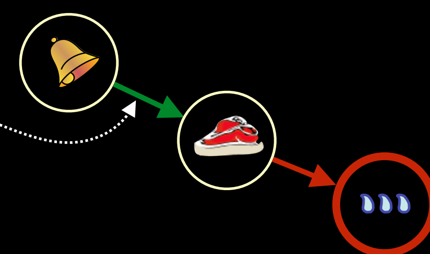
    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

Prediction error

$$\Delta v = r_t - v_t$$

Weight adjustment

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



where:

r_t is the actual value of the CS
v_t is the predicted value of the CS
Δv is the reward prediction error
w_t is the strength of the association between the CS and US

Prediction Learning

- **Rescorla-Wagner Learning Rule** *(1972)*
  (also known as "delta rule," or "reward prediction learning")
  - **the idea:** ***error-driven learning***

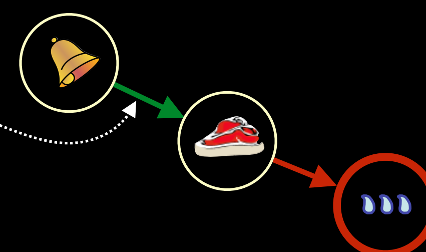
    associations are learned (strengthened) as a function of the difference between the *actual* and *predicted* outcome:

Prediction error

$$\Delta v = r_t - v_t$$

Weight adjustment

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



where:

r_t is the actual value of the CS
v_t is the predicted value of the CS
Δv is the reward prediction error
w_t is the strength of the association between the CS and US
α is the learning rate

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Trial 1

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Trial 1

Reward Prediction

Prediction error

$$\Delta v = r_t \;-\; v_t$$

Trial 1

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

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Trial 1

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

Reward Prediction

Prediction error

$$\Delta v = r_t \; - \; v_t$$

Trial 1

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

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Trial 1

$$+ \left[\; 0 \; \right]$$

Reward Prediction

Prediction error

$$\Delta v = r_t - v_t$$

Trial 1

+ +

$$\left[\ 0\ \right]$$

Reward Prediction

Prediction error | *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_t + \alpha\, \Delta v$$

Trial 1

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

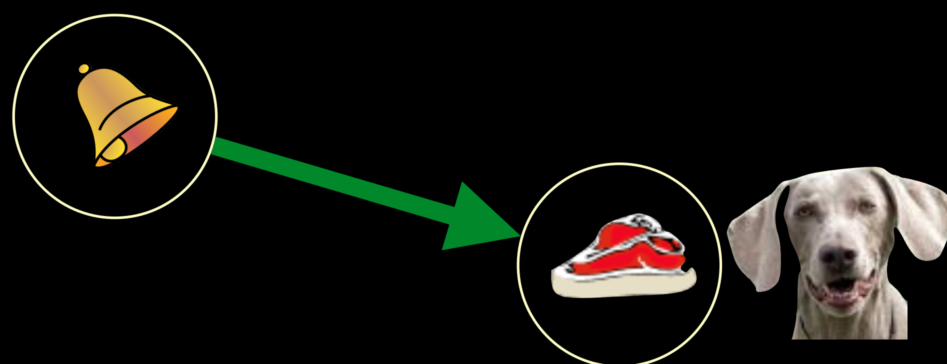
Reward Prediction

Prediction error *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+1} = w_t + \alpha\, \Delta v$$

Trial 1

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



Reward Prediction

Prediction error | *Weight adjustment*

$\Delta v = r_t - v_t$

$w_{t+1} + \alpha\, \Delta v$

 $\left[\;\; + \;\;\right]$

 $\left[\;\; 0 \;\;\right]$

Reward Prediction

Prediction error | *Weight adjustment*

$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha \, \Delta v$

Trial 2

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

Reward Prediction

Prediction error *Weight adjustment*

$\Delta v = r_t - v_t$ $w_{t+2} = w_{t+1} + \alpha \Delta v$

Trial 2

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

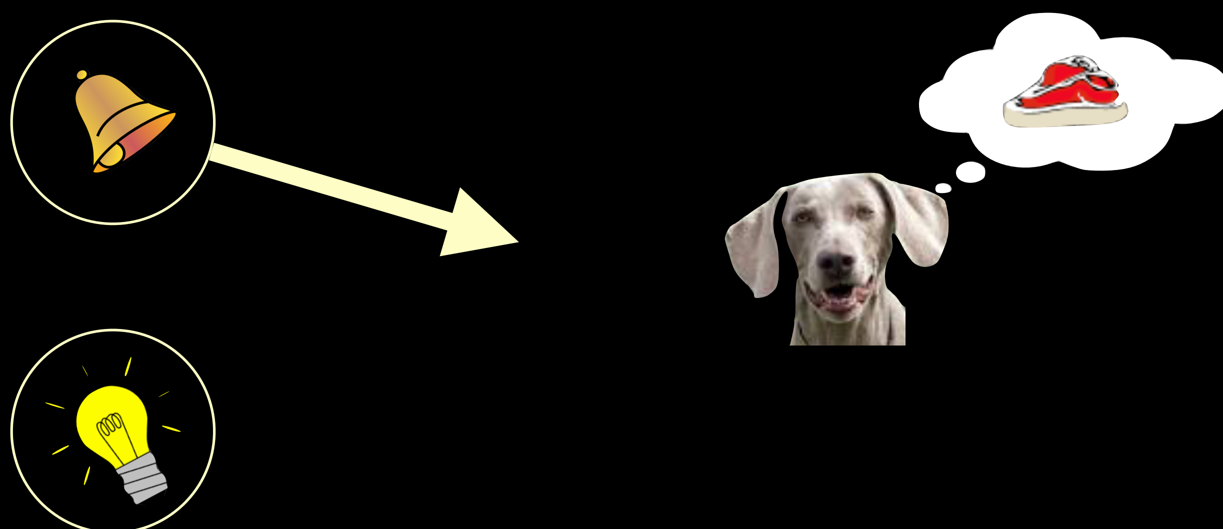
Reward Prediction

Prediction error | *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha \, \Delta v$$

Trial 2

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



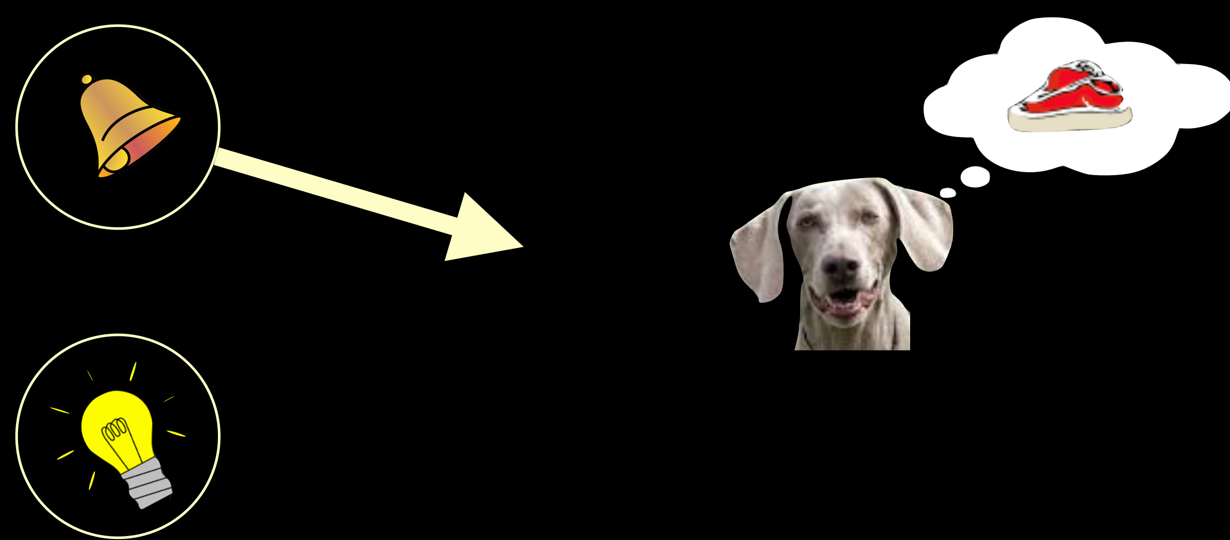
Reward Prediction

Prediction error   *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha\,\Delta v$$

Trial 2

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



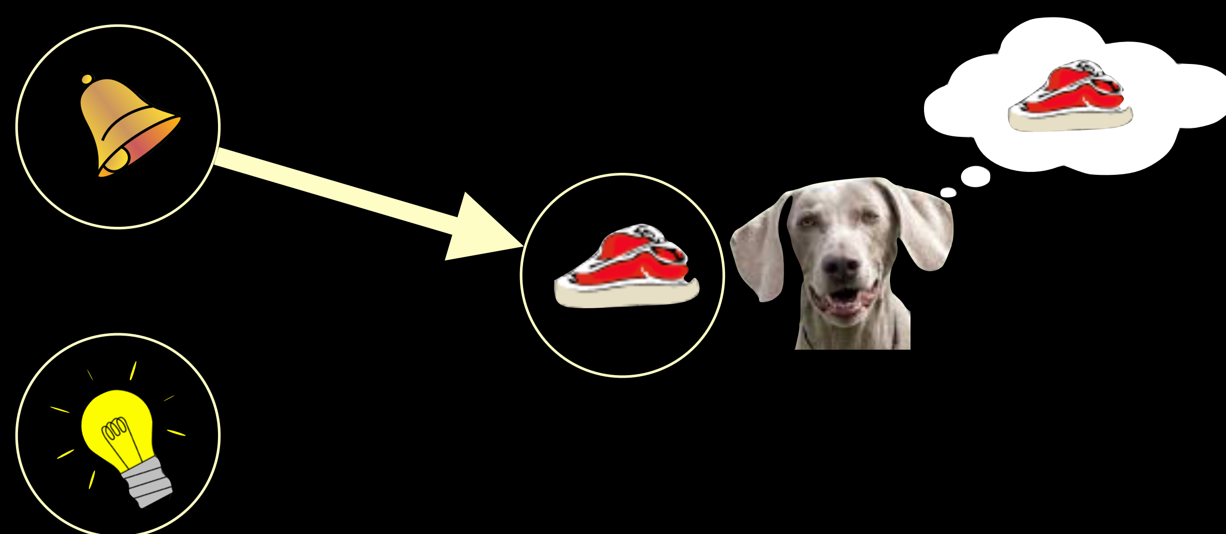
Reward Prediction

Prediction error | *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha \, \Delta v$$

Trial 2

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



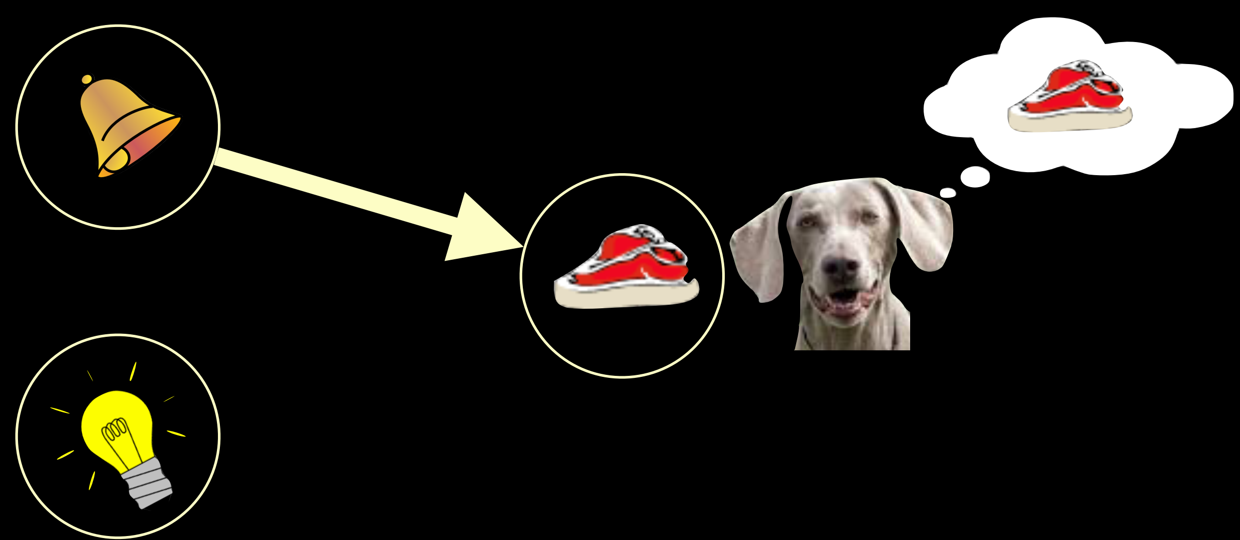
Reward Prediction

Prediction error *Weight adjustment*

$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha\, \Delta v$

Trial 2


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



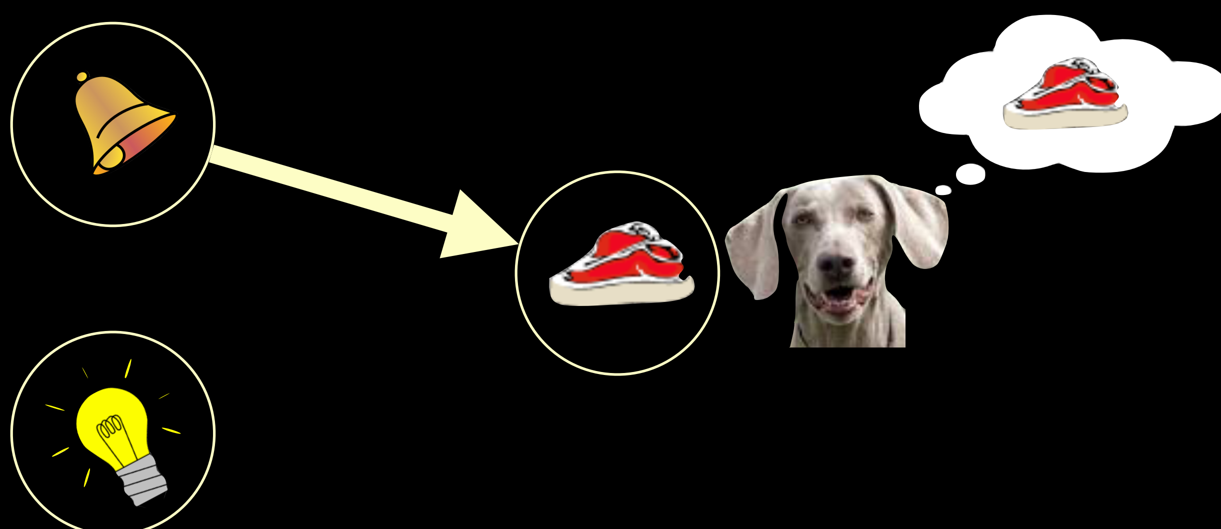
Reward Prediction

Prediction error *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha\,\Delta v$$

Trial 2

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



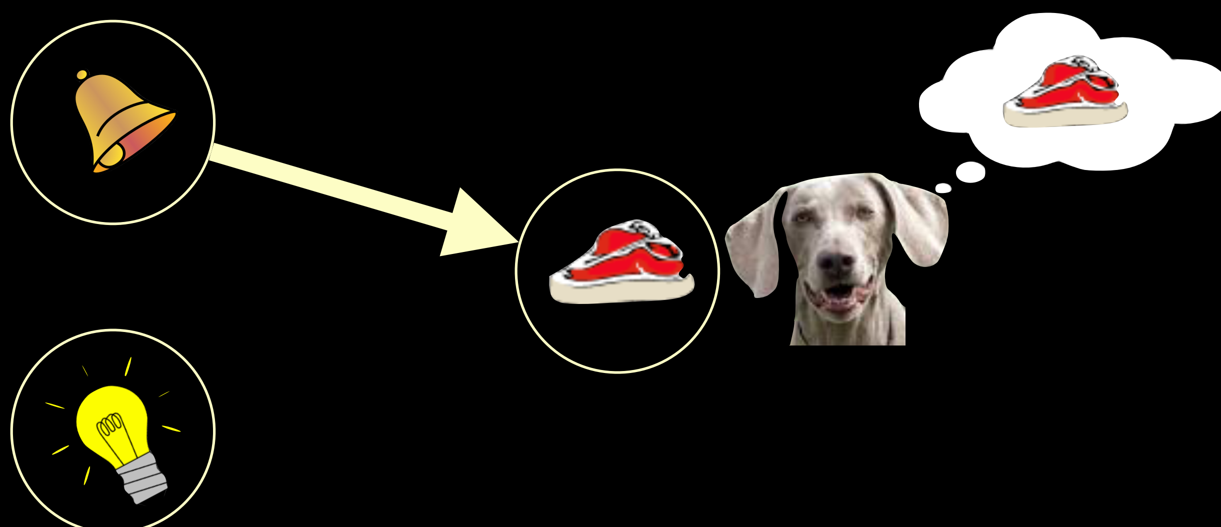
Reward Prediction

Prediction error | *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha \, \Delta v$$

Trial 2

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



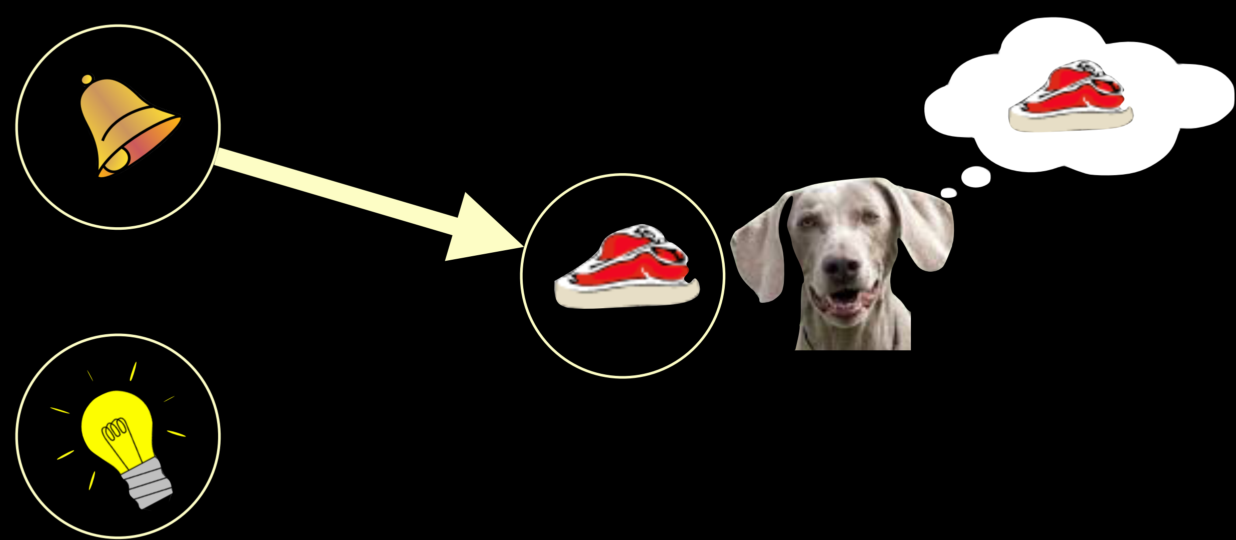
Reward Prediction

Prediction error | *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha\, \Delta v$$

Trial 2

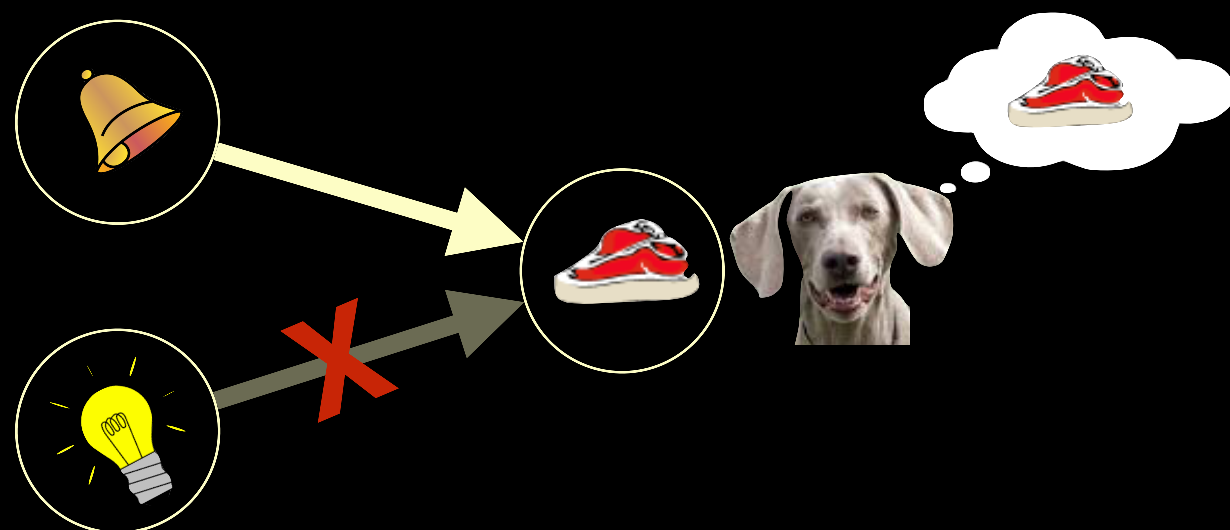
$$0 \quad + \quad \begin{bmatrix} + & + & + \\ 0 & 0 & 0 \end{bmatrix} \quad 0$$



Reward Prediction

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

Trial 2 $\quad 0 \; + \; \begin{bmatrix} + & + & + \\ 0 & 0 & 0 \end{bmatrix} \; 0$



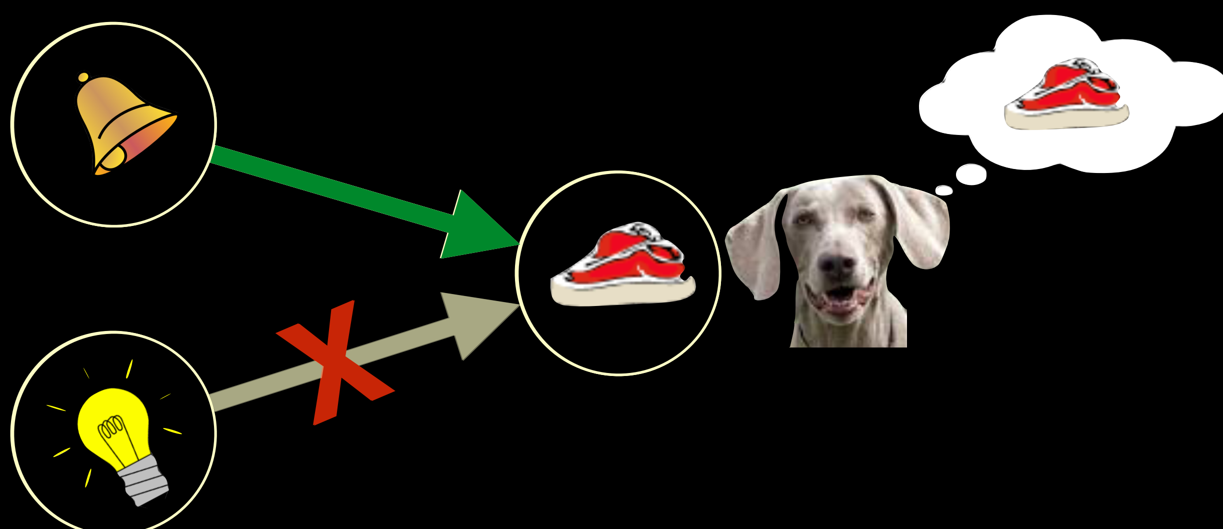
Reward Prediction

Prediction error *Weight adjustment*

$$\Delta v = r_t - v_t \qquad w_{t+2} = w_{t+1} + \alpha\, \Delta v$$

Trial 2

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



Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error

$$\Delta w_S = r_{S(t)} - w_{S(t)}$$

Weight adjustment

$$w_{S(t+1)} = w_{S(t)} + \alpha \, \Delta w_S$$

Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error ***Weight adjustment***

$$\alpha \quad r_{S(t)} - w_{S(t)}$$

Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error ***Weight adjustment***

$$w_{S(t+1)} = w_{S(t)} + \alpha r_{S(t)} - \alpha w_{S(t)})$$

Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error ***Weight adjustment***

$$w_{S(t+1)} = w_{S(t)} - \alpha w_{S(t)} \;\; + \alpha r_{S(t)})$$

Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error ***Weight adjustment***

$$w_{s(t+1)} = (1-\alpha)\, w_{s(t)} + \alpha\, r_{s(t)}\,)$$

Prediction Learning

- **Rescorla-Wagner Learning Rule**

Prediction error

Weight adjustment

$$w_{s(t+1)} = (1-\alpha)\, w_{s(t)} + \alpha\, r_{s(t)}\,)$$

Learn to predict rewards by averaging:

Prediction Learning

- Rescorla-Wagner Learning Rule

Prediction error

Weight adjustment

$$w_{S(t+1)} = (1-\alpha)\, w_{S(t)} + \alpha\, r_{s(t)})$$

Learn to predict rewards
by averaging: learned predictions

Prediction Learning

- Rescorla-Wagner Learning Rule

Prediction error

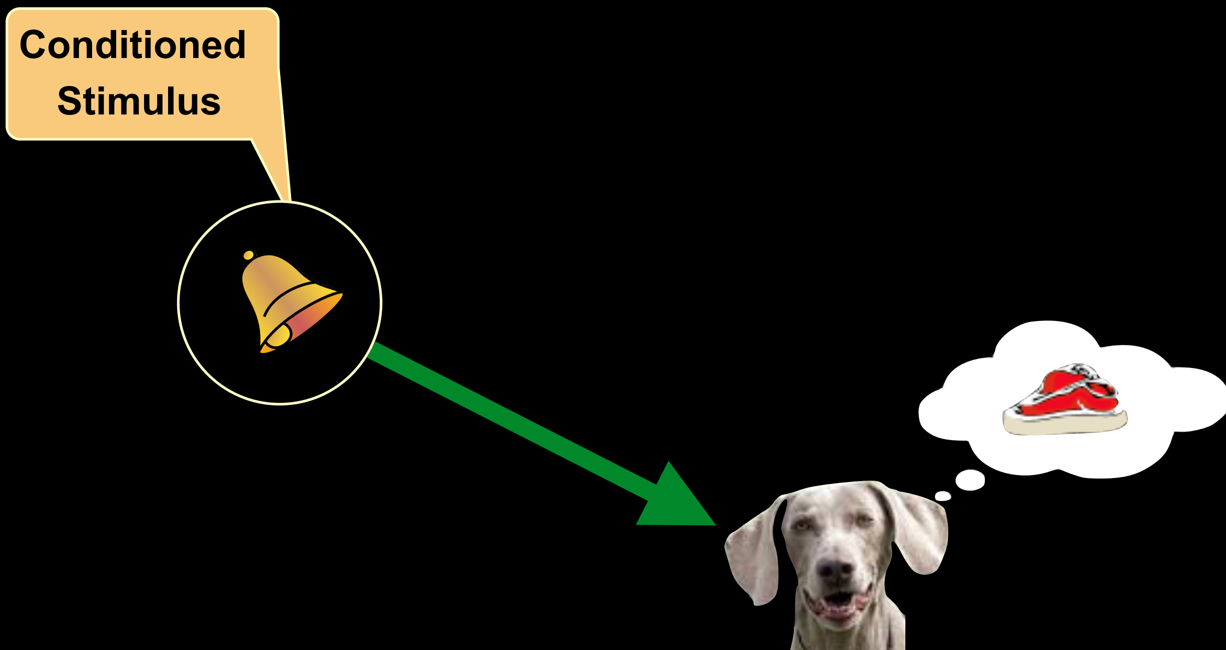
Weight adjustment

$$w_{S(t+1)} = (1-\alpha)\, w_{S(t)} + \alpha\, r_{S(t)}\,)$$

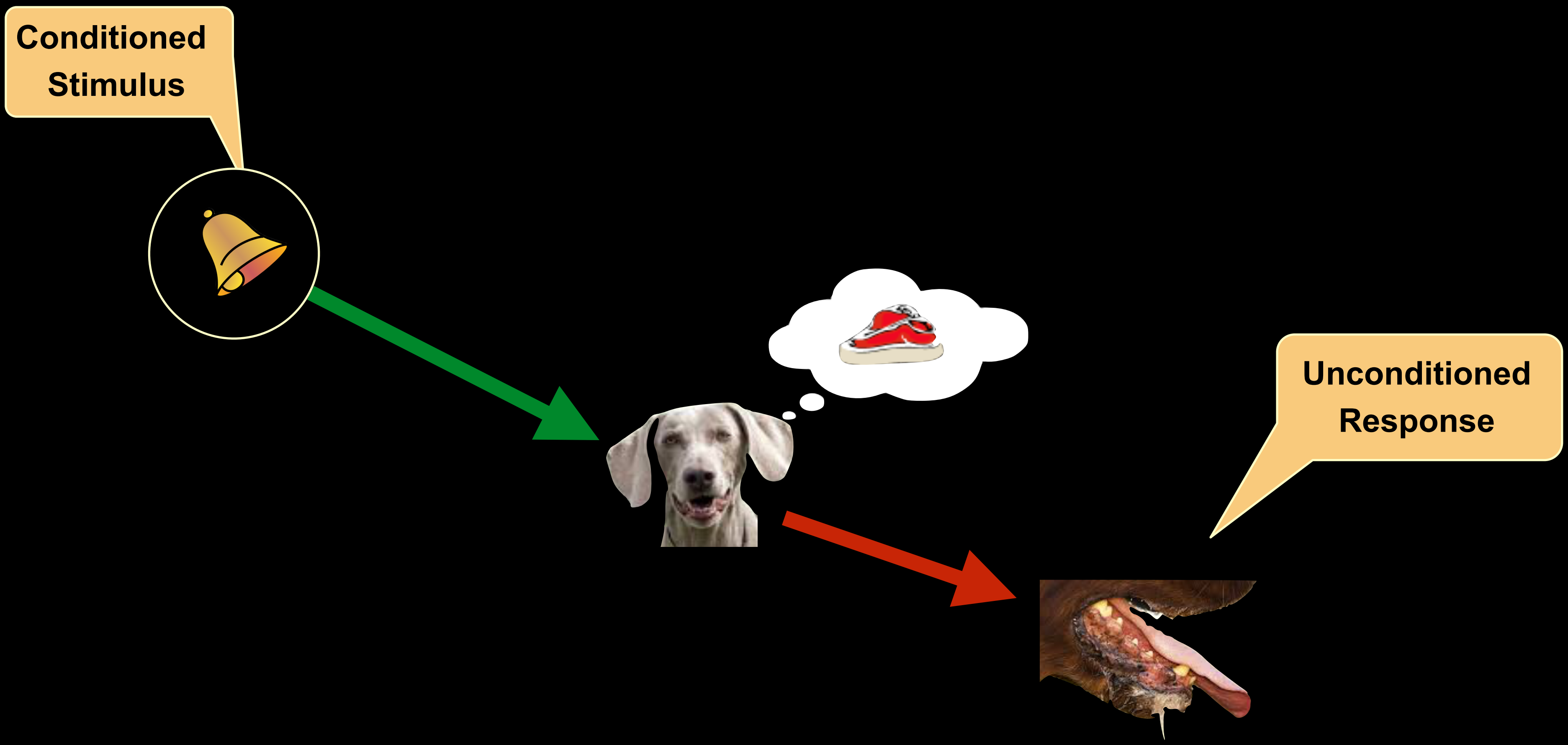
Learn to predict rewards
by averaging: learned predictions with present reward

Conditioning and Prediction

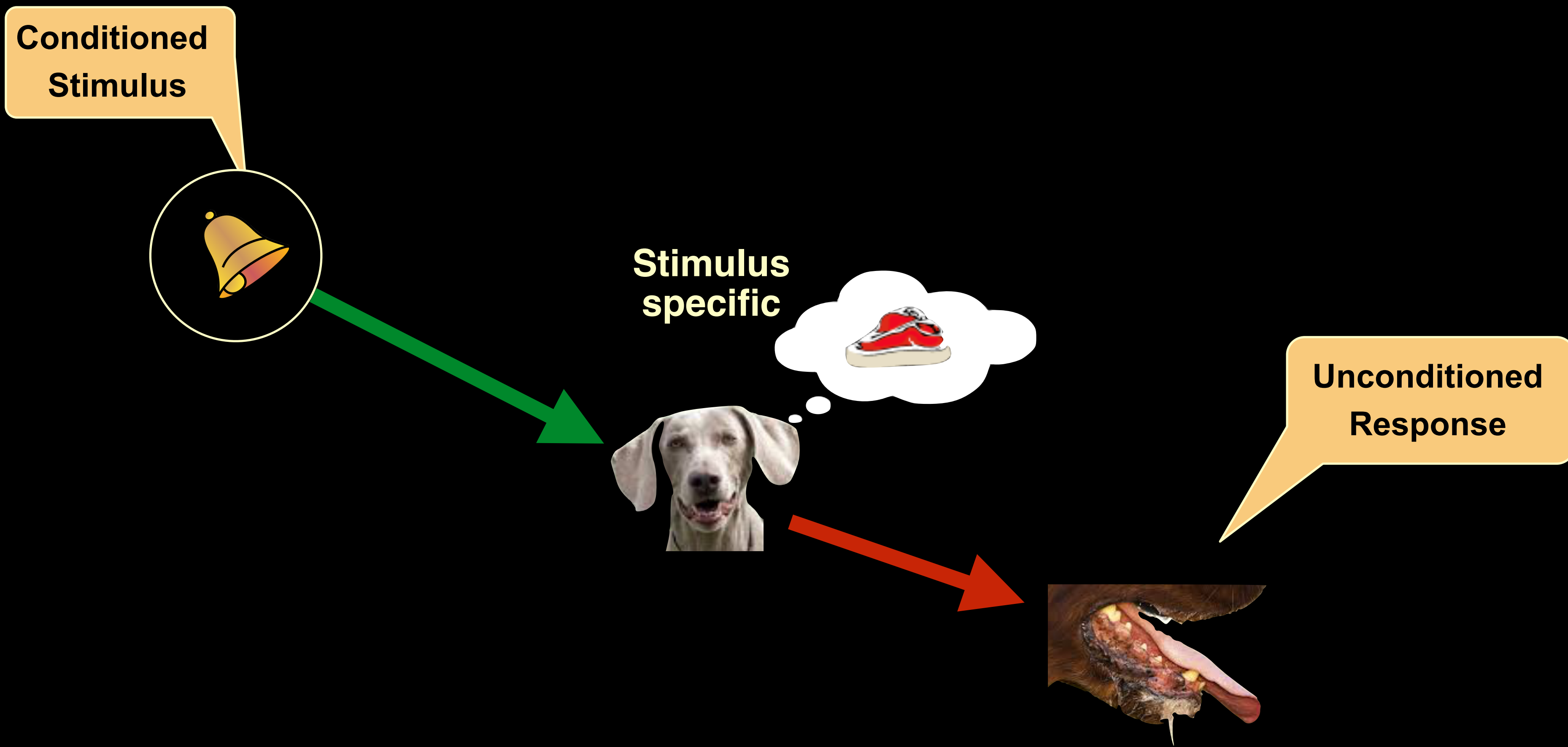
Conditioning and Prediction



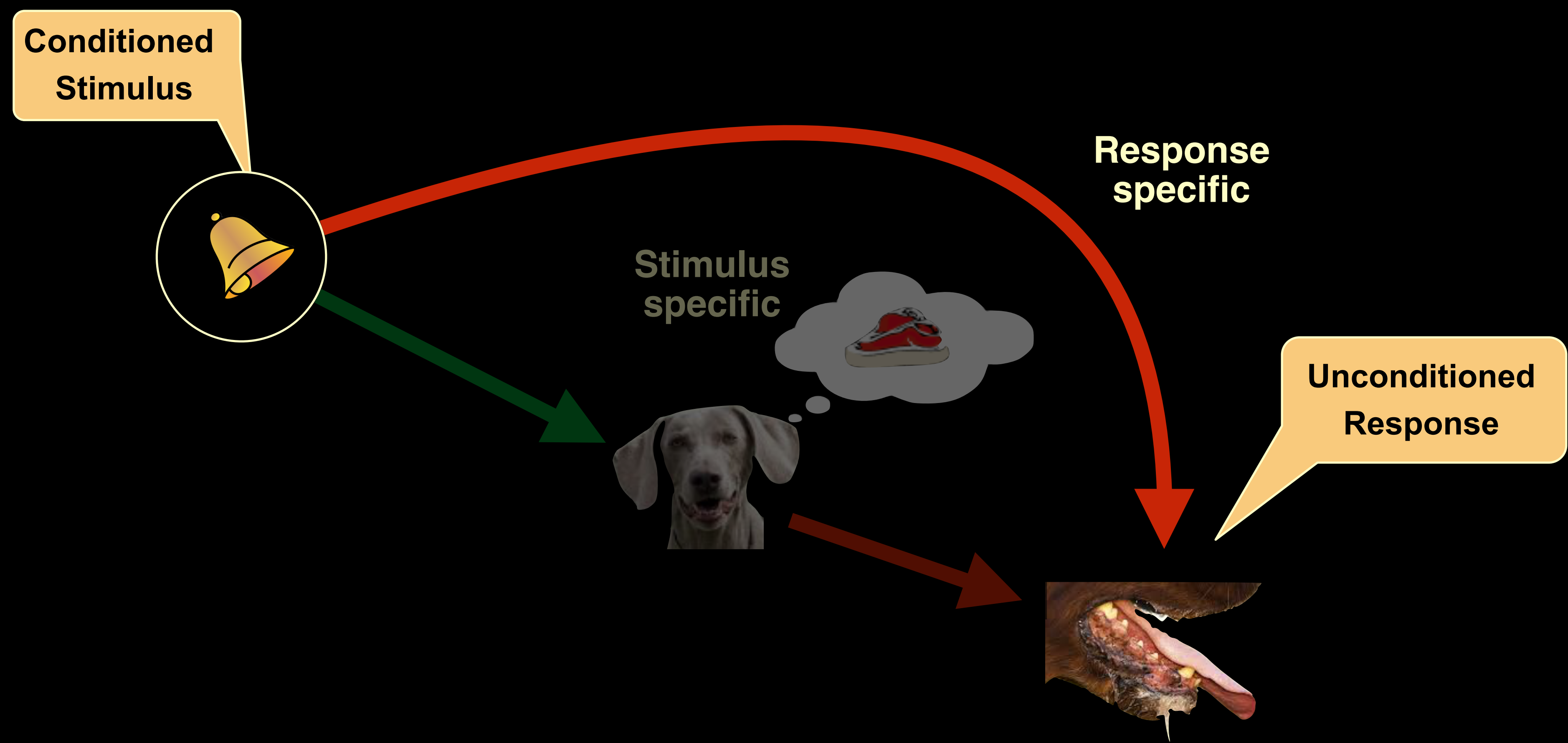
Conditioning and Prediction



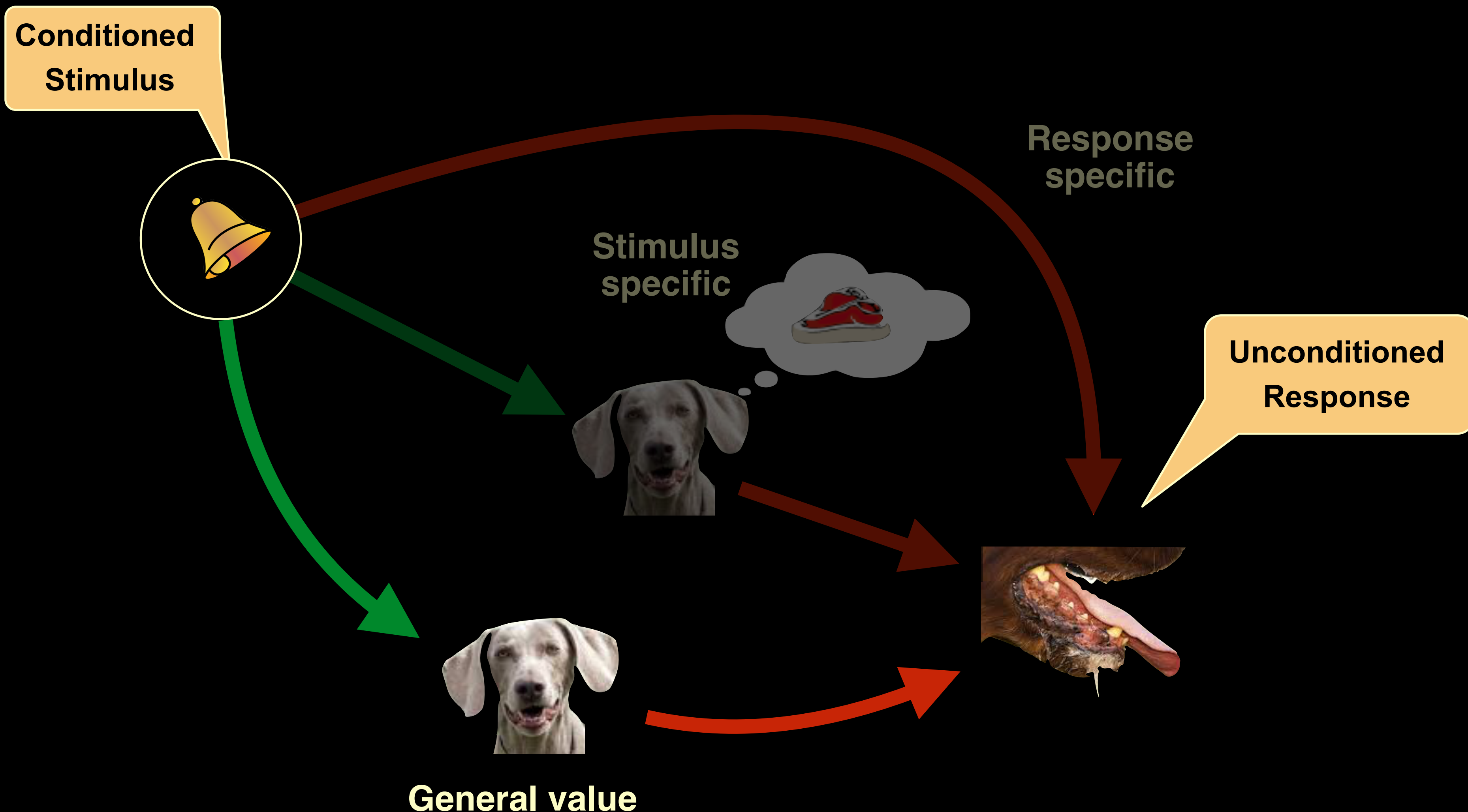
Conditioning and Prediction



Conditioning and Prediction



Conditioning and Prediction



2nd Order Conditioning

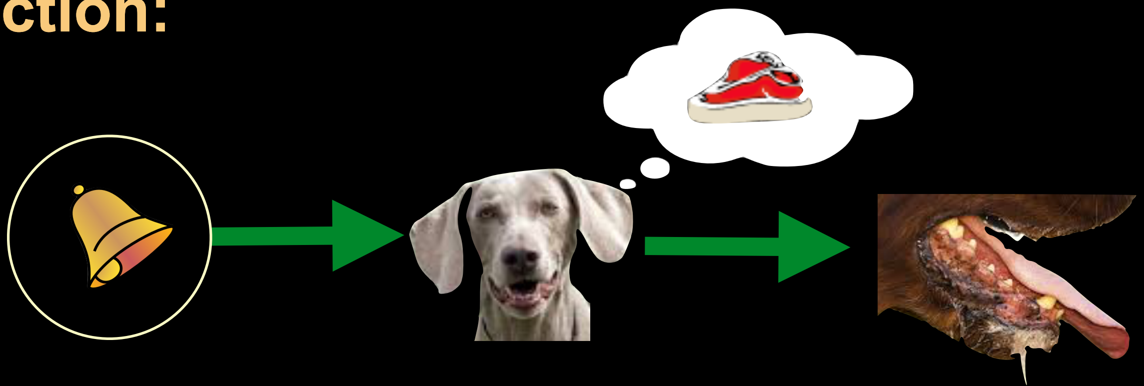
Sequential prediction:

2nd Order Conditioning

Sequential prediction:

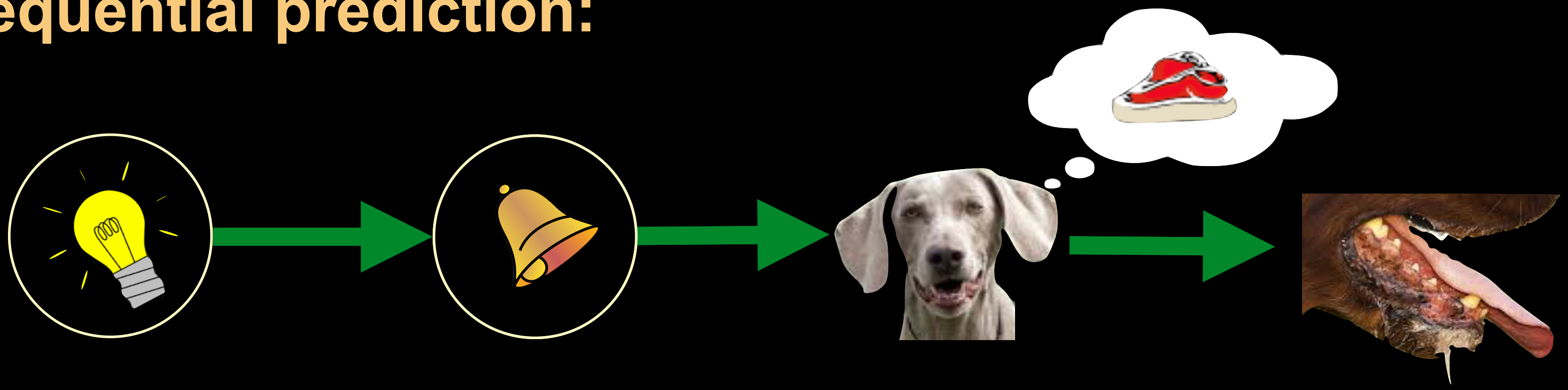
2nd Order Conditioning

Sequential prediction:



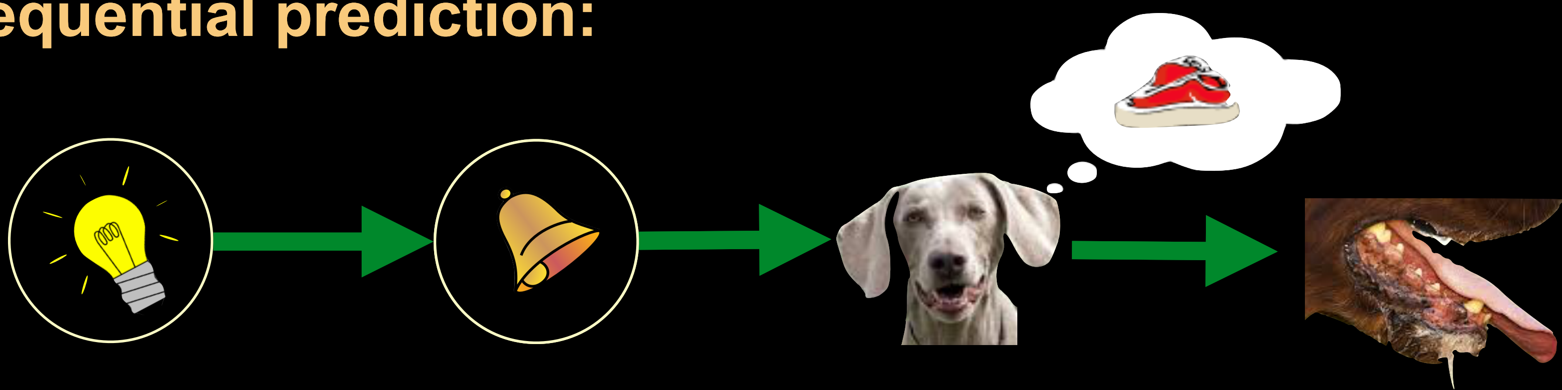
2nd Order Conditioning

Sequential prediction:



2nd Order Conditioning

Sequential prediction:

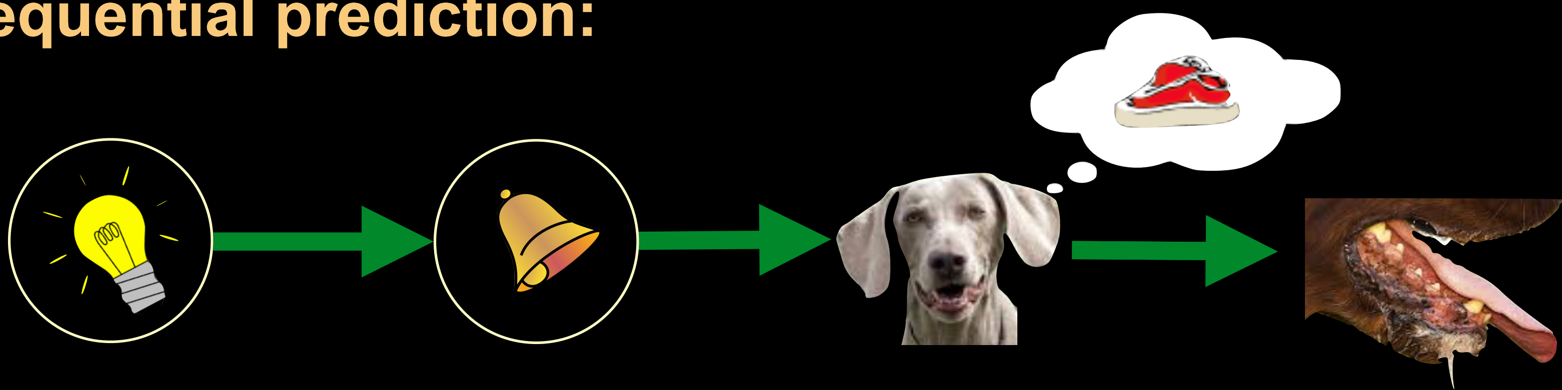


Rescorla-Wagner Rule can't handle this,

can only learn to predict the value of the current event
(cue does not generate an *actual* reward)

2nd Order Conditioning

Sequential prediction:

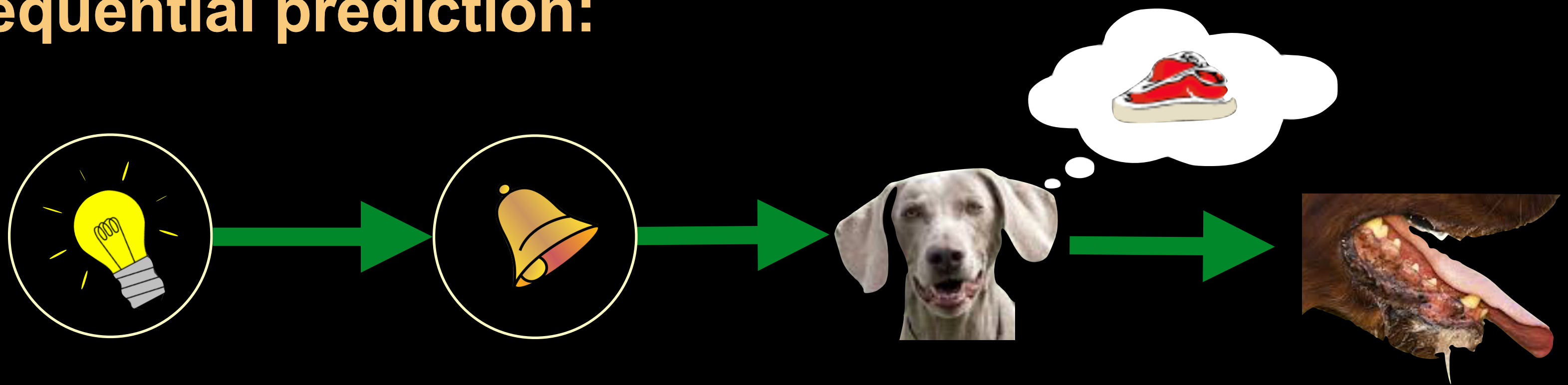


Rescorla-Wagner Rule can't handle this,

can only learn to predict the value of the current event
(cue does not generate an *actual* reward)

2nd Order Conditioning

Sequential prediction:



Rescorla-Wagner Rule can't handle this,

can only learn to predict the value of the current event
(cue does not generate an *actual* reward)

Supervised Learning: Scalar

- Conditioning
  - Simple Prediction
    Rescorla-Wagner Rule
  - Stimulus-Action Associations
    Actor-critic model, Q Learning
- 👉 Sequence Prediction
  - Method of Temporal Differences (TD)
  - Model-Free vs. Model-Based RL
- Challenges
  - Curse of dimensionality
    - *Hierarchical RL: policies and options*
    - *State space abstraction*
  - Explore-exploit
    - ***Meta-control***

Prediction Learning

Prediction error | *Weight adjustment*

Rescorla-Wagner: $\Delta w_S = r_{S(t)} - w_{S(t)}$ $\qquad w_{S(t+1)} = w_{S(t)} + \alpha \Delta w_S$

Predict current reward:
$w_{S(t)} = r_{S(t)}$

Prediction Learning

Prediction error | *Weight adjustment*

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\quad w_{s(t+1)} = w_{s(t)} + \alpha \, \Delta w_s$

Predict current reward:
$w_{s(t)} = r_{s(t)}$

Currently
predicted
reward

Prediction Learning

Prediction error | *Weight adjustment*

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\qquad$ $w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Prediction Learning

Rescorla-Wagner:

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Prediction error

$$\Delta w_s = r_{s(t)} - w_{s(t)}$$

Received reward

Currently predicted reward

Weight adjustment

$$w_{s(t+1)} = w_{s(t)} + \alpha \, \Delta w_s$$

Updated prediction

Old prediction of expected reward

Prediction Learning

Prediction error — **Weight adjustment**

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\quad w_{s(t+1)} = w_{s(t)} + \alpha \, \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Temporal Difference:
(Sutton & Barto, 1981)

*Predict **all future** rewards:*

$$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$$
$$= r_{s(t)} + w_{s(t+1)}$$

[Bellman equation]

by updating existing ("OLD") predictions

Prediction Learning

Prediction error | *Weight adjustment*

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Temporal Difference: $\Delta w_s = r_{s(t)} + w_{s(t)} - w_{s(t-1)}$

(Sutton & Barto, 1981)

*Predict **all future** rewards:*

$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$

$= r_{s(t)} + w_{s(t+1)}$

[Bellman equation]

by updating existing ("OLD") predictions

Prediction Learning

Prediction error | *Weight adjustment*

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\quad w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Temporal Difference: $\Delta w_s = r_{s(t)} + w_{s(t)} - w_{s(t-1)}$

(Sutton & Barto, 1981)

Previously expected reward

*Predict **all future** rewards:*

$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$

$= r_{s(t)} + w_{s(t+1)}$

[Bellman equation]

by updating existing ("OLD") predictions

Prediction Learning

Prediction error | **Weight adjustment**

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\quad w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Temporal Difference: $\Delta w_s = r_{s(t)} + w_{s(t)} - w_{s(t-1)}$

(Sutton & Barto, 1981)

Predict ***all future*** *rewards:*

$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$

$= r_{s(t)} + w_{s(t+1)}$

[Bellman equation]

by updating existing (“OLD”) predictions

Previously expected reward

Currently predicted reward

Prediction Learning

Prediction error — *Weight adjustment*

Rescorla-Wagner:

$$\Delta w_s = r_{s(t)} - w_{s(t)} \qquad w_{s(t+1)} = w_{s(t)} + \alpha \, \Delta w_s$$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Temporal Difference:
(Sutton & Barto, 1981)

$$\Delta w_s = r_{s(t)} + w_{s(t)} - w_{s(t-1)}$$

*Predict **all future** rewards:*

$$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$$
$$= r_{s(t)} + w_{s(t+1)}$$

[Bellman equation]

by updating existing ("OLD") predictions

Currently received reward

Currently predicted reward

Previously expected reward

Prediction Learning

Prediction error — *Weight adjustment*

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $\quad w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward — Currently predicted reward — Updated prediction — Old prediction of expected reward

Reward Prediction Error

Temporal Difference: $\Delta w_s = r_{s(t)} + w_{s(t)} - w_{s(t-1)}$

(Sutton & Barto, 1981)

Predict ***all future*** *rewards:*

$$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$$
$$= r_{s(t)} + w_{s(t+1)}$$

[Bellman equation]

by updating existing ("OLD") predictions

Currently received reward — Currently predicted reward — Previously expected reward

Prediction Learning

Prediction error | **Weight adjustment**

Rescorla-Wagner:

$$\Delta w_s = r_{s(t)} - w_{s(t)}$$

$$w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Reward Prediction Error

Temporal Difference:

(Sutton & Barto, 1981)

*Predict **all future** rewards:*

$$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \ldots$$

$$= r_{s(t)} + w_{s(t+1)}$$

[Bellman equation]

by updating existing ("OLD") predictions

$$w_{s(t-1)} = w_{s(t-1)} + \alpha \Delta w_s$$

Prediction Learning

Prediction error ***Weight adjustment***

Rescorla-Wagner: $\Delta w_s = r_{s(t)} - w_{s(t)}$ $w_{s(t+1)} = w_{s(t)} + \alpha \Delta w_s$

Predict current reward:

$w_{s(t)} = r_{s(t)}$

Received reward

Currently predicted reward

Updated prediction

Old prediction of expected reward

Reward Prediction Error

Temporal Difference:
(Sutton & Barto, 1981)

*Predict **all future** rewards:*

$$w_{s(t)} = r_{s(t)} + r_{s(t+1)} + r_{s(t+2)} + \dots$$
$$= r_{s(t)} + w_{s(t+1)}$$

[Bellman equation]

by updating existing ("OLD") predictions

$$w_{s(t-1)} = w_{s(t-1)} + \alpha \Delta w_s$$

Updated "Old" prediction

"Old" reward prediction

Prediction Learning

- **Overall logic:**

Prediction Learning

- **Overall logic:**
  - ***Want:*** $V_s(t) = r_s(t)$ *(predicted value in state s at time t)*

Prediction Learning

- **Overall logic:**
  - ***Error signal:*** $\boldsymbol{\delta(t) = r_s(t) - V_s(t)}$ *(observed minus predicted)*

Prediction Learning

- **Overall logic:**
  - ***Update:*** $V_s(t+1) \leftarrow V_s(t) + \varepsilon * \delta(t)$

Prediction Learning

- **Overall logic:**
  - ***Update:*** $V_s(t+1) \leftarrow V_s(t) + \varepsilon * \delta(t)$
- **Predictions:**
  - ***Predictions*** **are** ***weights*** **that designate** ***exepcted value*****:**

Prediction Learning

- Overall logic:
  - *Update:* $V_s(t+1) \leftarrow V_s(t) + \varepsilon * \delta(t)$
- Predictions:
  - *Implicit*, not "active"

Prediction Learning

- Overall logic:
  - *Update:* $V_s(t+1) \leftarrow V_s(t) + \varepsilon * \delta(t)$
- Predictions:
  - *Conditional* (i.e., from a given state) not general

Prediction Learning

- **Overall logic:**
  - ***Update:*** $V_s(t+1) \leftarrow V_s(t) + \varepsilon * \delta(t)$
- **Predictions:**

Normally think of prediction as something active,
but easier to think about them here as weights
so you can think about existing ones you would make
in a given state

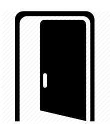
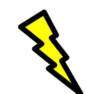
Temporal Difference Learning

Updated prediction for previous time step · *Old prediction at that time step* · *Current reward* · *Current prediction* · *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}} \qquad \alpha = 0.5$$

predicted value

prediction error

State / time step:	***1***	***2***	***3***	***4***
Stimulus / reward:	 $r(1) = 0$	 $r(2) = 0$	 $r(3) = 0$	 $r(4) = -1$

Temporal Difference Learning

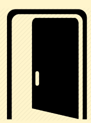
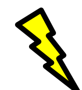
Updated prediction for previous time step | Old prediction at that time step | Current reward | Current prediction | Old prediction at that time step

$$W_S^{new}(t-1) = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

$\alpha = 0.5$

predicted value

prediction error

State / time step:	***1***	***2***	***3***	***4***
Stimulus / reward:	 $r(1) = 0$	 $r(2) = 0$	 $r(3) = 0$	 $r(4) = -1$

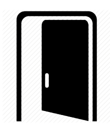
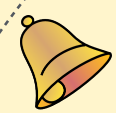
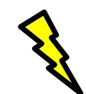
trial 1: W_S^{new}*(t-1)*

Temporal Difference Learning

Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

α = 0.5

State / time step:	***1***	***2***	***3***	***4***
Stimulus / reward:	 $r(1) = 0$	 $r(2) = 0$	 $r(3) = 0$	 $r(4) = -1$

trial 1*:* $W_S^{new}(t\text{-}1)$ $\quad 0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $\quad$ ***= 0***

Temporal Difference Learning

Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

α = 0.5

State / time step:	***1***	***2***	***3***	***4***
Stimulus / reward:	$r(1) = 0$	$r(2) = 0$	$r(3) = 0$	$r(4) = -1$
trial 1: W_S^{new}*(t-1)*	$0 + 0.5\underbrace{(0+0-0)}_{0}$ ***= 0***	$0 + 0.5\underbrace{(0+0-0)}_{0}$ ***= 0***		

Temporal Difference Learning

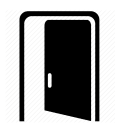
Updated prediction for previous time step | *Old prediction at that time step* | *Current reward* | *Current prediction* | *Old prediction at that time step*

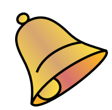
$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha\,\underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

$\alpha = 0.5$

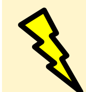
State / time step: **1** **2** **3** **4**

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

trial 1: $W_S^{new}(t\text{-}1)$

$0 + 0.5\underbrace{(0+0-0)}_{0}$ $= 0$

$0 + 0.5\underbrace{(0+0-0)}_{0}$ $= 0$

$0 + 0.5\underbrace{(-1+0-0)}_{-1}$ $= -0.5$

Temporal Difference Learning

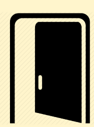
Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha\ \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

α = 0.5

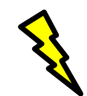
State / time step: ***1*** ***2*** ***3*** ***4***

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ *= 0*

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ *= 0*

$0 + 0.5\underbrace{(-1 + 0 - 0)}_{-1}$ *= -0.5*

trial 2: W_S^{new}*(t-1)*

Temporal Difference Learning

Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

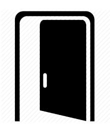
$\alpha = 0.5$

predicted value

prediction error

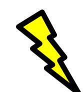
State / time step: **1** **2** **3** **4**

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

$0 + 0.5\underbrace{(0+0-0)}_{0}$ = 0

$0 + 0.5\underbrace{(0+0-0)}_{0}$ = 0

$0 + 0.5\underbrace{(-1+0-0)}_{-1}$ = -0.5

trial 2: $W_S^{new}(t\text{-}1)$ $0 + 0.5\underbrace{(0+0-0)}_{0}$ **= 0**

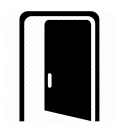
Temporal Difference Learning

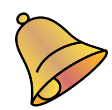
Updated prediction for previous time step · *Old prediction at that time step* · *Current reward* · *Current prediction* · *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}} \qquad \alpha = 0.5$$

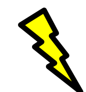
State / time step: **1** **2** **3** **4**

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

Time step 1: $0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

Time step 2: $0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

Time step 3: $0 + 0.5\underbrace{(-1 + 0 - 0)}_{-1}$ $= -0.5$

trial 2: $W_S^{new}(t\text{-}1)$

Time step 1: $0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

Time step 2: $0 + 0.5\underbrace{(0 - 0.5 - 0)}_{-0.5}$ $= -0.25$

Temporal Difference Learning

Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha\ \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

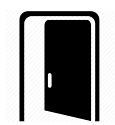
α = 0.5

predicted value

prediction error

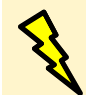
State / time step: ***1*** ***2*** ***3*** ***4***

Stimulus / reward:


$r(1) = 0$


$r(2) = 0$


$r(3) = 0$


$r(4) = -1$

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ *= 0*

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ *= 0*

$0 + 0.5\underbrace{(-1 + 0 - 0)}_{-1}$ *= -0.5*

***trial 2**:* W_S^{new}*(t-1)*

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ *= 0*

$0 + 0.5\underbrace{(0 - 0.5 - 0)}_{-0.5}$ *= -0.25*

$-0.5 + 0.5\underbrace{(-1 + 0 + 0.5)}_{-0.5}$ ***= -0.75***

Temporal Difference Learning

Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

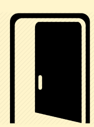
α = 0.5

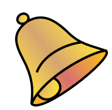
predicted value

prediction error

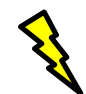
State / time step: **1** **2** **3** **4**

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

1	2	3
$0 + 0.5\underbrace{(0+0-0)}_{0}$ *= 0*	$0 + 0.5\underbrace{(0+0-0)}_{0}$ *= 0*	$0 + 0.5\underbrace{(-1+0-0)}_{-1}$ *= -0.5*
$0 + 0.5\underbrace{(0+0-0)}_{0}$ *= 0*	$0 + 0.5\underbrace{(0-0.5-0)}_{-0.5}$ *= -0.25*	$-0.5 + 0.5\underbrace{(-1+0+0.5)}_{-0.5}$ *= -0.75*

trial 3: W_S^{new}*(t-1)*

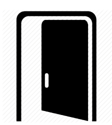
Temporal Difference Learning

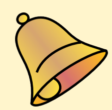
Updated prediction for previous time step — *Old prediction at that time step* — *Current reward* — *Current prediction* — *Old prediction at that time step*

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}} \qquad \alpha = 0.5$$

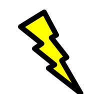
State / time step: **1** **2** **3** **4**

Stimulus / reward:

 $r(1) = 0$

 $r(2) = 0$

 $r(3) = 0$

 $r(4) = -1$

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

$0 + 0.5\underbrace{(-1 + 0 - 0)}_{-1}$ $= -0.5$

$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$

$0 + 0.5\underbrace{(0 - 0.5 - 0)}_{-0.5}$ $= -0.25$

$-0.5 + 0.5\underbrace{(-1 + 0 + 0.5)}_{-0.5}$ $= -0.75$

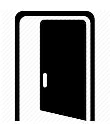
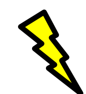
trial 3: $W_S^{new}(t\text{-}1)$ $0 + 0.5\underbrace{(0 - 0.25 - 0)}_{-0.25}$ $= -0.125$

Temporal Difference Learning

Updated prediction for previous time step | Old prediction at that time step | Current reward | Current prediction | Old prediction at that time step

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha\ \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}}$$

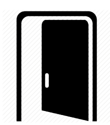
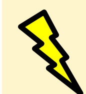
$\alpha = 0.5$

State / time step:	1	2	3	4
Stimulus / reward:	 $r(1) = 0$	 $r(2) = 0$	 $r(3) = 0$	 $r(4) = -1$
	$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(-1 + 0 - 0)}_{-1}$ $= -0.5$	
	$0 + 0.5\underbrace{(0 + 0 - 0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(0 - 0.5 - 0)}_{-0.5}$ $= -0.25$	$-0.5 + 0.5\underbrace{(-1 + 0 + 0.5)}_{-0.5}$ $= -0.75$	
trial 3: $W_S^{new}(t\text{-}1)$	$0 + 0.5\underbrace{(0 - 0.25 - 0)}_{-0.25}$ $= -0.125$	$-0.25 + 0.5\underbrace{(0 - 0.75 + 0.25)}_{-0.5}$ $= -0.5$		

Temporal Difference Learning

Updated prediction for previous time step	Old prediction at that time step	Current reward	Current prediction	Old prediction at that time step

$$\underbrace{W_S^{new}(t-1)}_{\text{predicted value}} = W_S^{old}(t-1) + \alpha \underbrace{\left(r_S(t) + W_S^{old}(t) - W_S^{old}(t-1)\right)}_{\text{prediction error}} \qquad \alpha = 0.5$$

State / time step:	*1*	*2*	*3*	*4*
Stimulus / reward:	 $r(1) = 0$	 $r(2) = 0$	 $r(3) = 0$	 $r(4) = -1$
	$0 + 0.5\underbrace{(0+0-0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(0+0-0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(-1+0-0)}_{-1}$ $= -0.5$	
	$0 + 0.5\underbrace{(0+0-0)}_{0}$ $= 0$	$0 + 0.5\underbrace{(0-0.5-0)}_{-0.5}$ $= -0.25$	$-0.5 + 0.5\underbrace{(-1+0+0.5)}_{-0.5}$ $= -0.75$	
trial 3: $W_S^{new}(t\text{-}1)$	$0 + 0.5\underbrace{(0-0.25-0)}_{-0.25}$ $= -0.125$	$-0.25 + 0.5\underbrace{(0-0.75+0.25)}_{-0.5}$ $= -0.5$	$-0.75 + 0.5\underbrace{(-1+0+0.75)}_{-0.25}$ **= -0.875**	

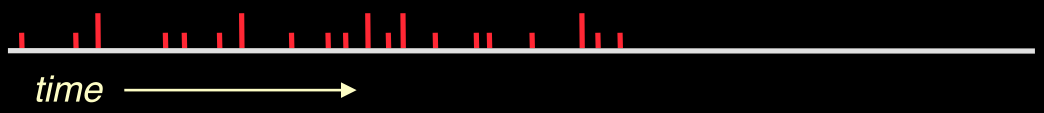
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996

Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996

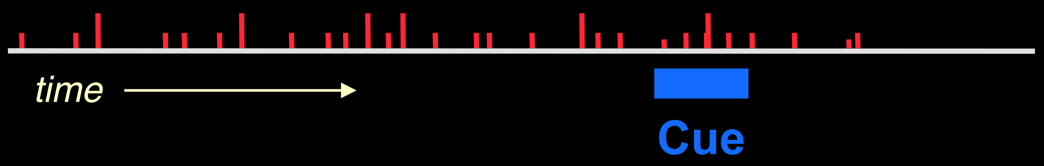
Before learning



Dopamine and TD Learning

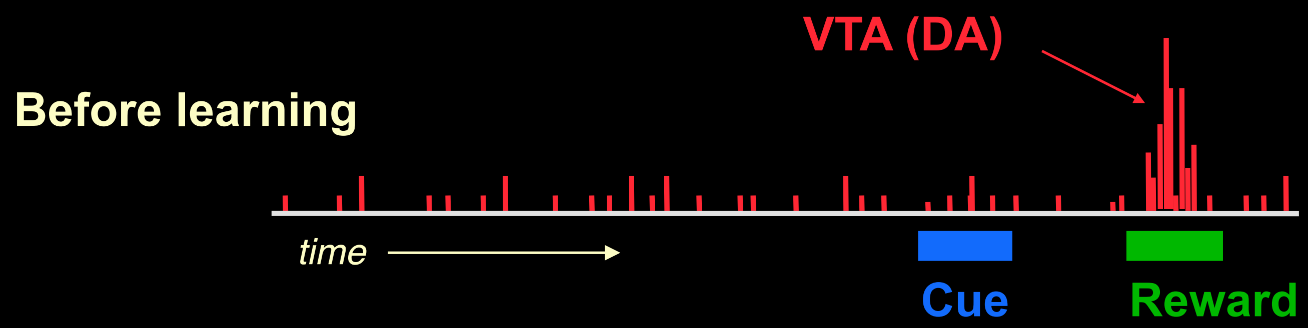
Schulz, 1993; Montague, Dayan & Sejnowski, 1996

Before learning



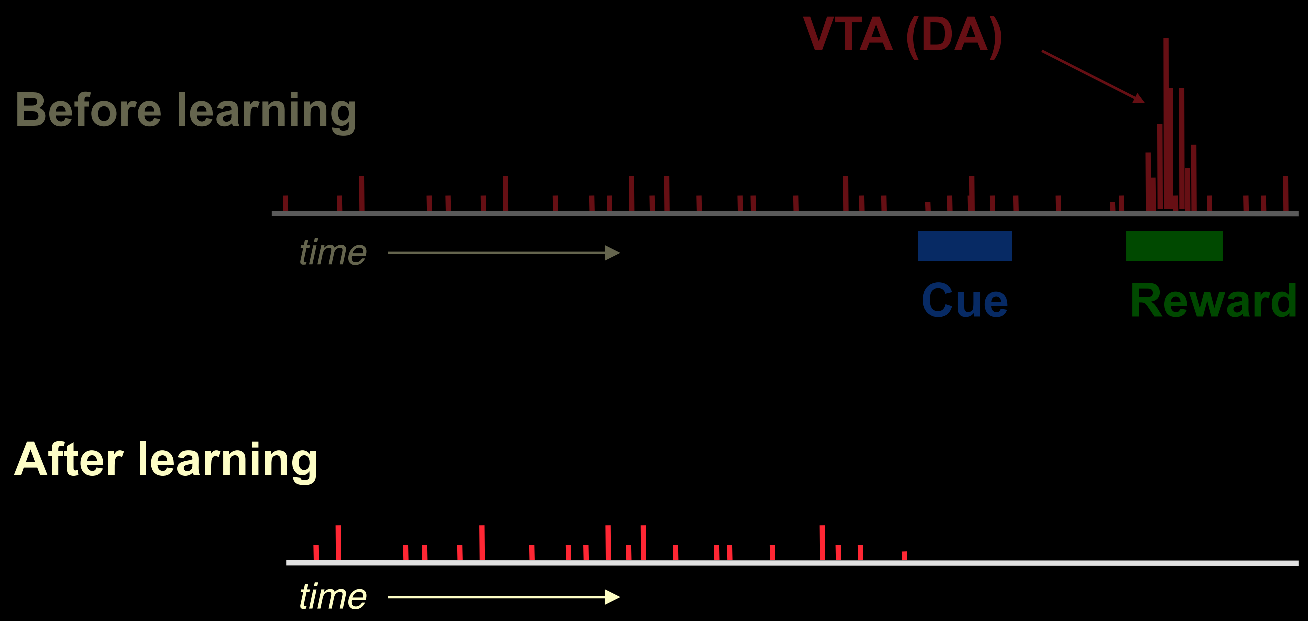
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



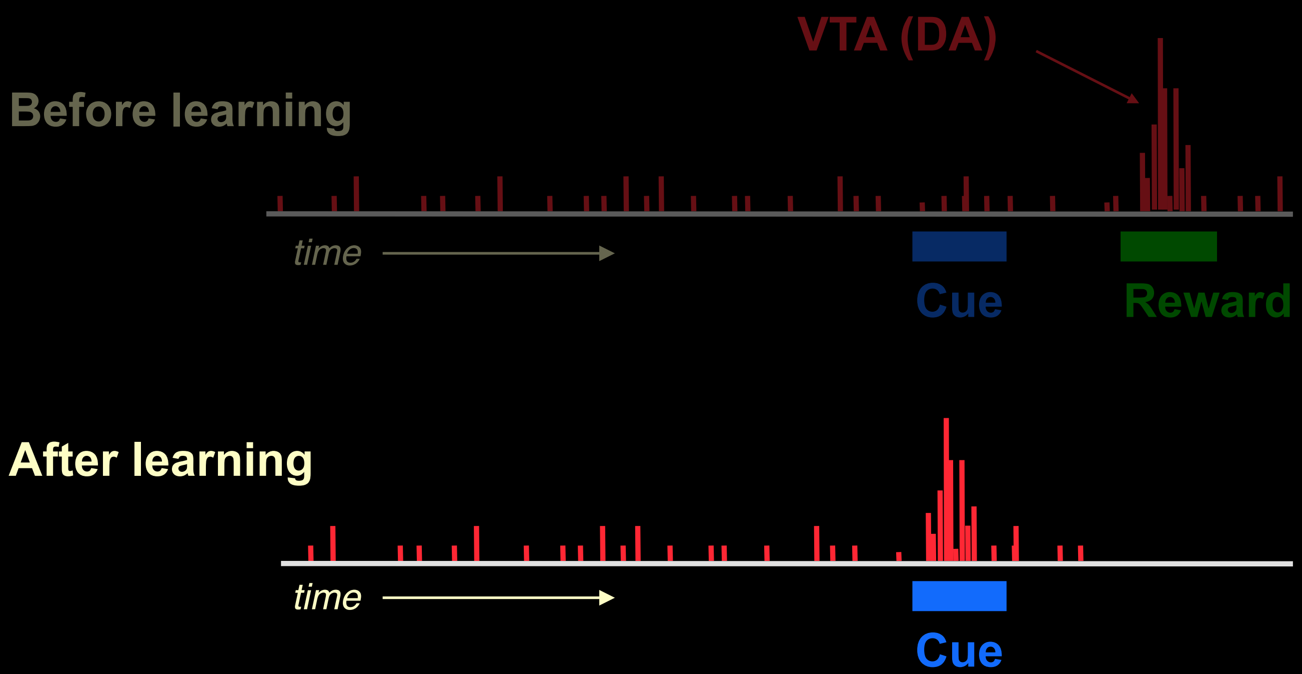
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



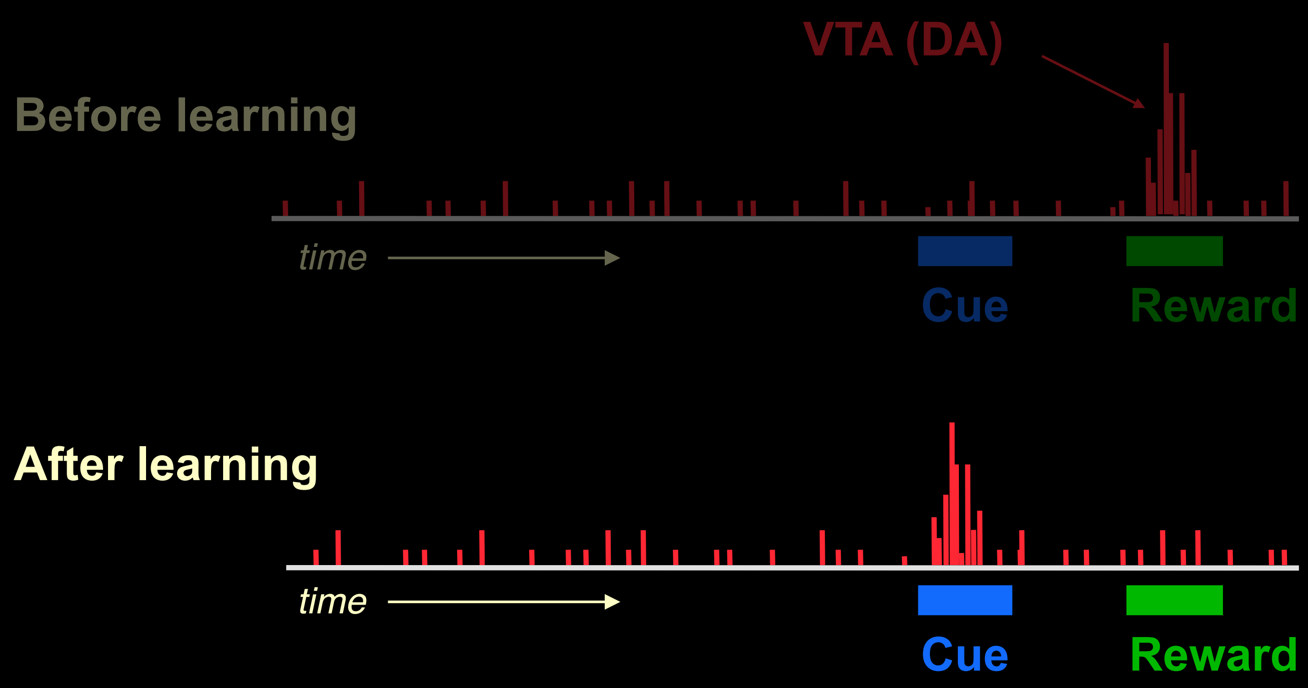
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



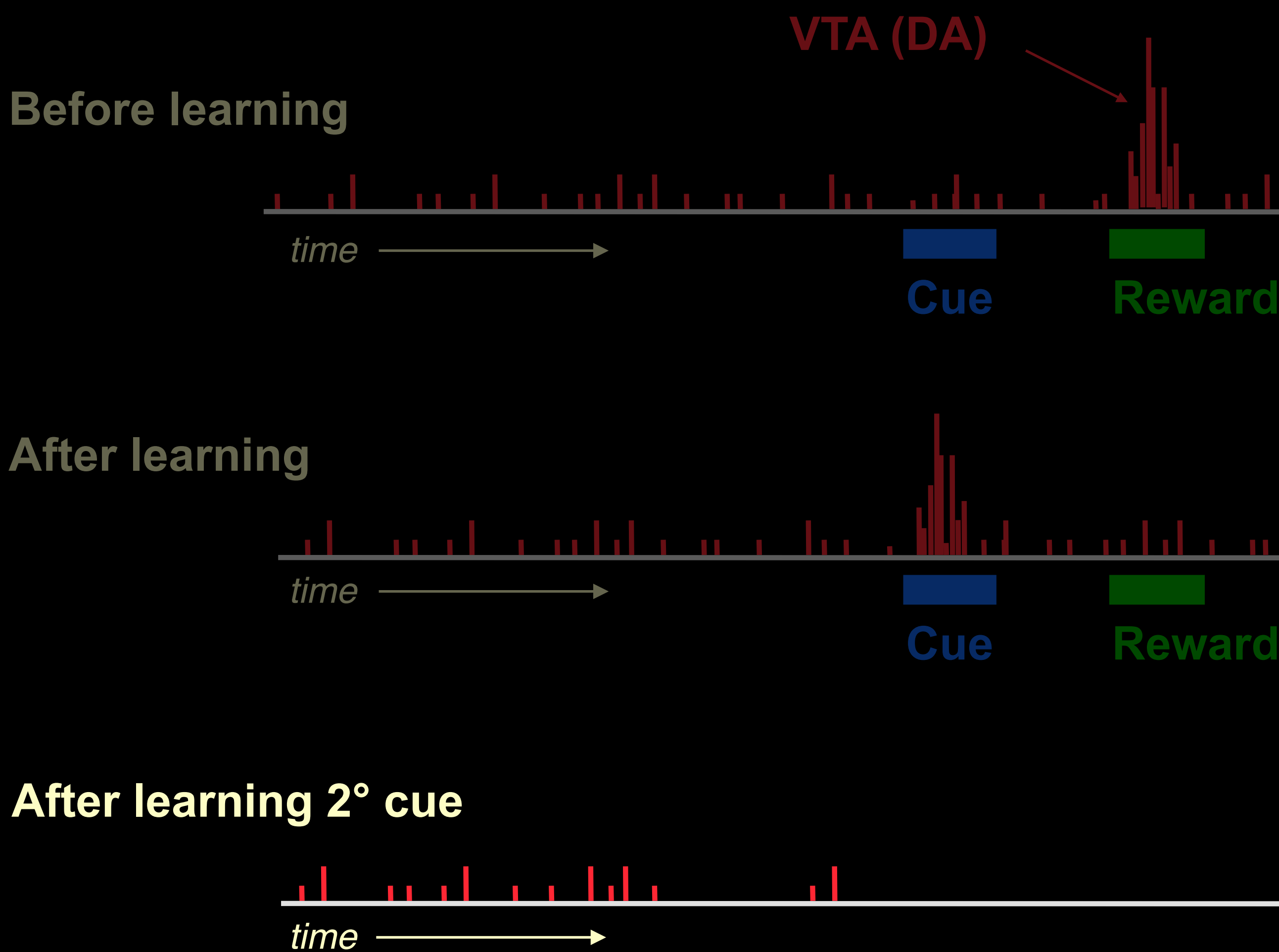
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



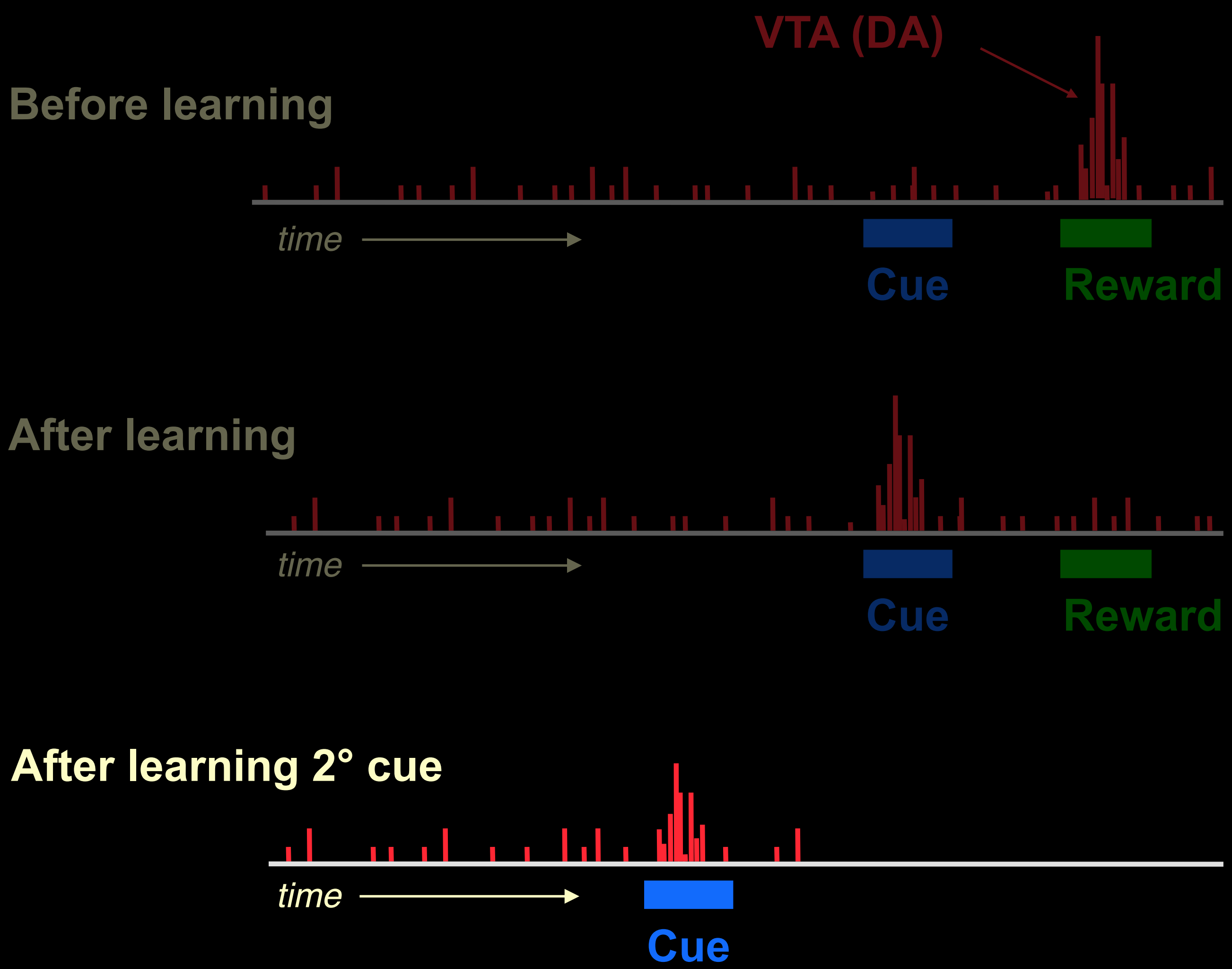
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



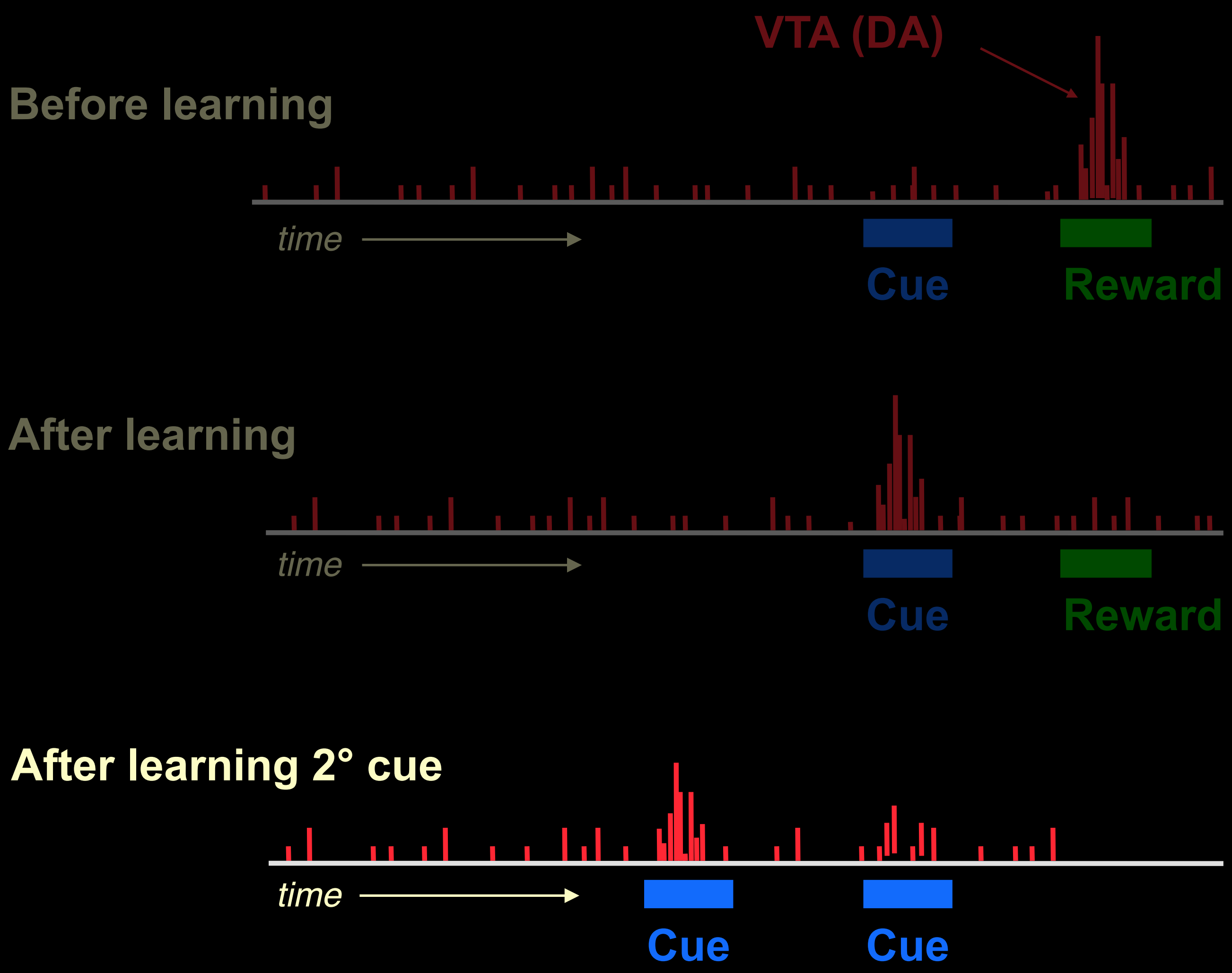
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



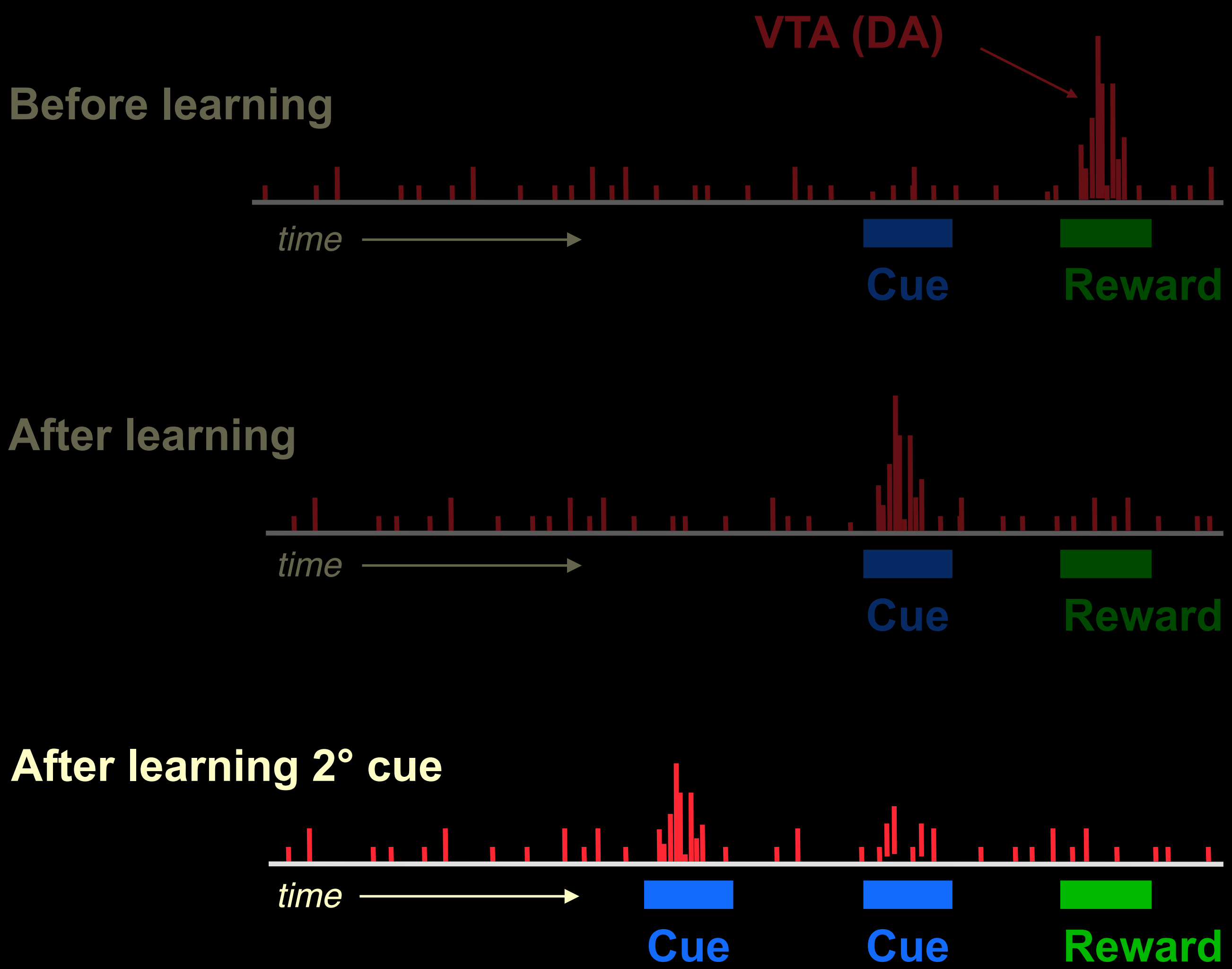
Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



Dopamine and TD Learning

Schulz, 1993; Montague, Dayan & Sejnowski, 1996



Temporal Difference Learning

Temporal Difference Learning

- Starting point for most RL applications
  - TD Gammon (1992)
  - Atari (2013)

Temporal Difference Learning

- Starting point for most RL applications
  - TD Gammon (1992)
  - Atari (2013)
  - Alpha Go (2015)
- Optimal learner, given:
  - a stationary (stable) environment
  - sufficient time to learn

Temporal Difference Learning

- Starting point for most RL applications
  - TD Gammon (1992)
  - Atari (2013)
  - Alpha Go (2015)

- Optimal learner, given:
  - a stationary (stable) environment
  - sufficient time to learn
  - all information needed for prediction is observable at the time it is relevant…

- Closely related to Markov Chains (MC):
  - MC: set of *observable states*
    and *transition probabilities* between them (e.g., a maze)

Temporal Difference Learning

- Starting point for most RL applications
  - TD Gammon (1992)
  - Atari (2013)
  - Alpha Go (2015)

- Optimal learner, given:
  - a stationary (stable) environment
  - sufficient time to learn
  - all information needed for prediction is observable at the time it is relevant…

- Closely related to Markov Chains (MC):
  - MC: set of *observable states* and *transition probabilities* between them (e.g., a maze)
  - RL learns the summed (discounted) value of future rewards for each state, based on the the transition probabilities emanating from each state

Temporal Difference Learning

- Starting point for most RL applications
  - TD Gammon (1992)
  - Atari (2013)
  - Alpha Go (2015)
- Optimal learner, given:
  - a stationary (stable) environment
  - sufficient time to learn
  - all information needed for prediction is observable at the time it is relevant…
- Closely related to Markov Chains (MC):
  - MC: set of *observable states* and *transition probabilities* between them (e.g., a maze)
  - RL learns the summed (discounted) value of future rewards for each state, based on the the transition probabilities emanating from each state
- **TD can be used to predict, but what about *actions?***

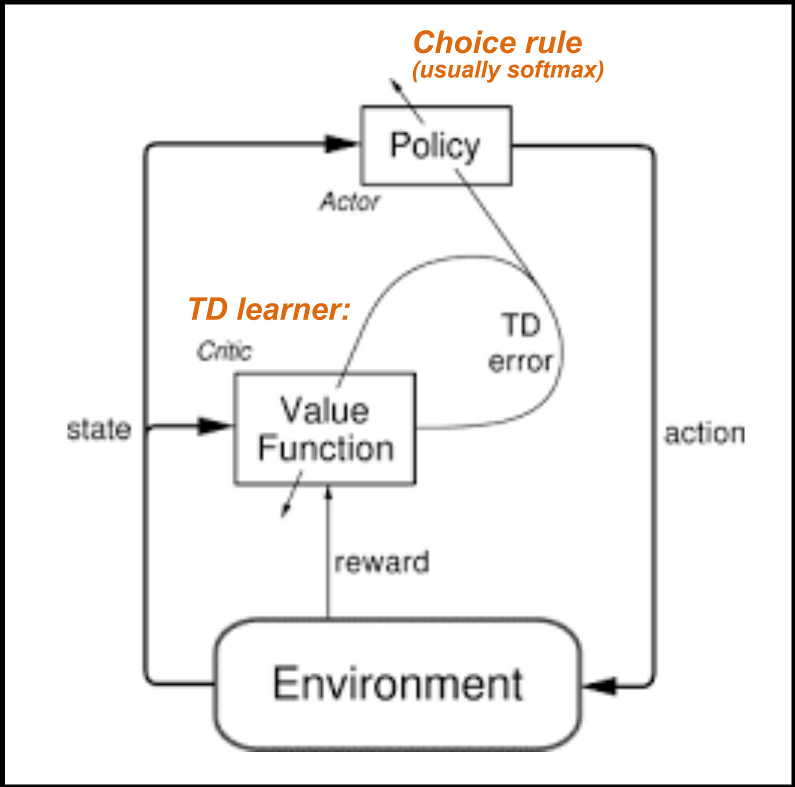
TD and Action Learning

TD and Action Learning

- TD error can be used to assign value not just to states, but also actions (based on the states to which they lead)

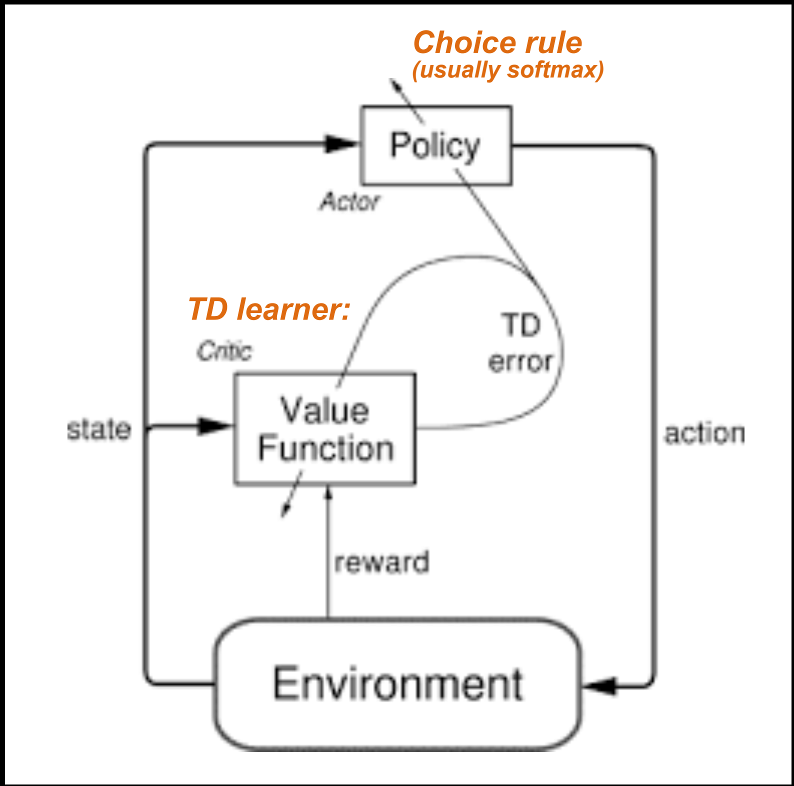
TD and Action Learning

- TD error can be used to assign value not just to states, but also actions (based on the states to which they lead)
  - Actor-critic model



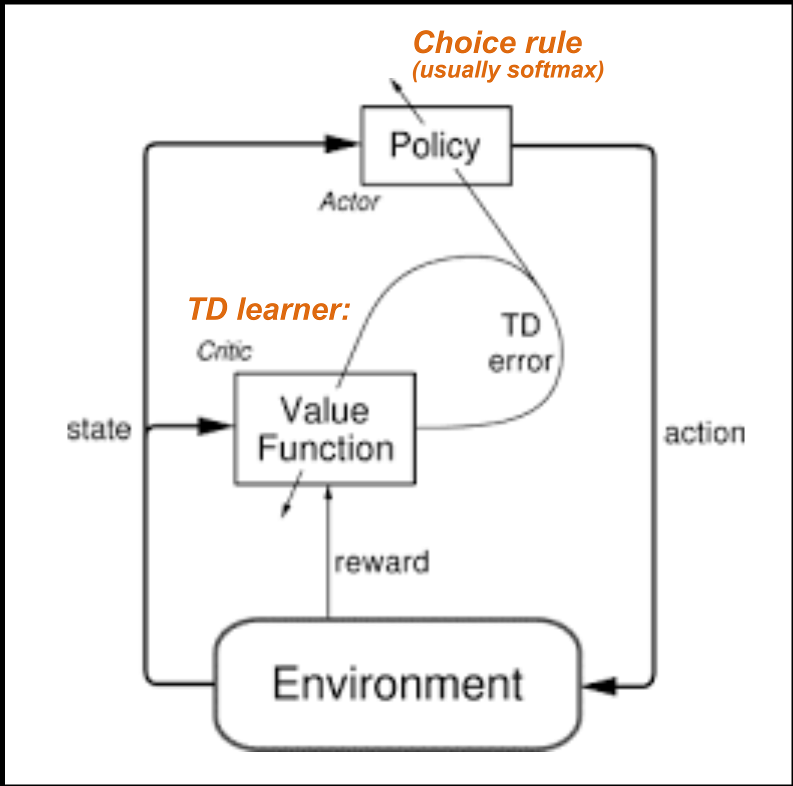
TD and Action Learning

- TD error can be used to assign value not just to states, but also actions (based on the states to which they lead)
  - Actor-critic model
  - Q-learning
    (value of an action in a given state values)



TD and Action Learning

- TD error can be used to assign value not just to states, but also actions (based on the states to which they lead)
  - Actor-critic model
  - Q-learning
    (value of an action in a given state values)
  - Gating and LSTMs…



Challenges

Challenges

- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***

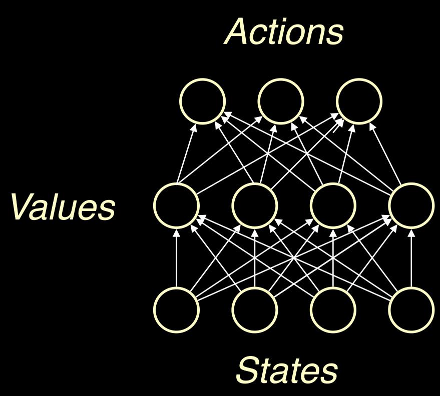
Challenges

- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***
  - **state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:**

Challenges

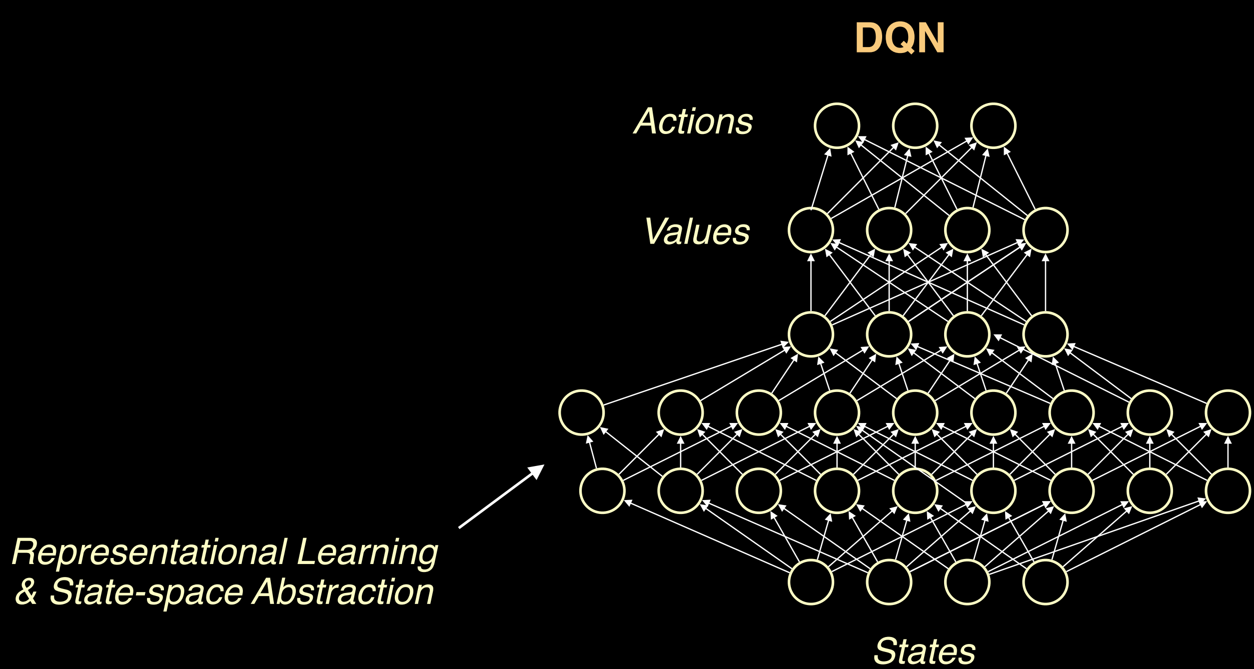
- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***
  - **state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:**
    - **state-space learning:**

Standard RL



Challenges

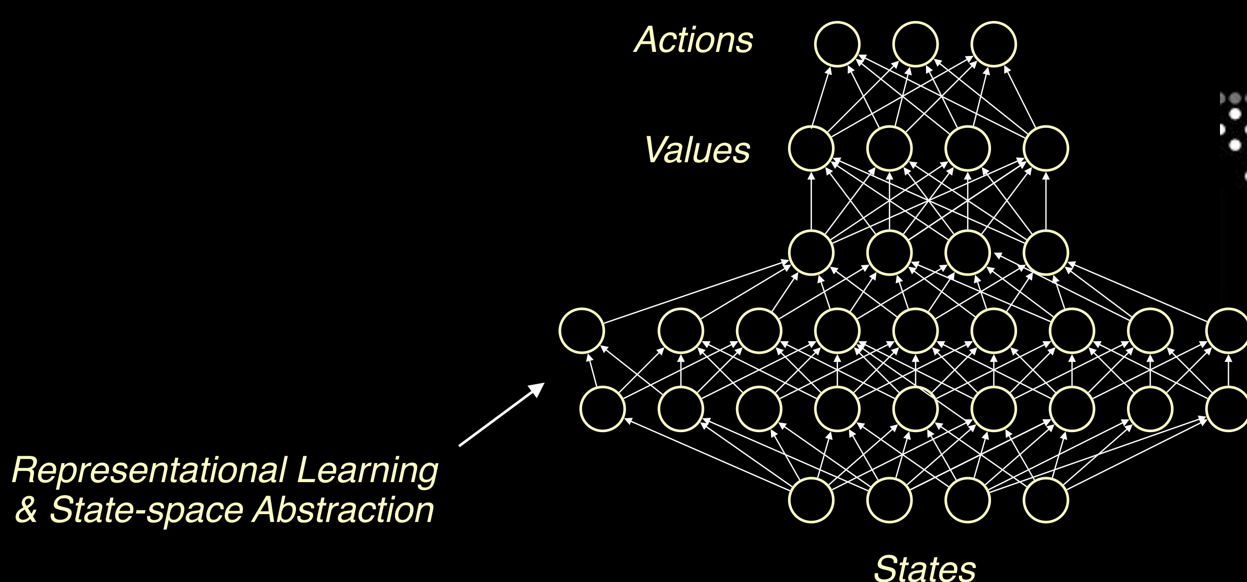
- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***
  - **state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:**
    - **state-space learning: Deep Q-Networks (DQN):**



Challenges

- **Curse of dimensionality**
  - as the number of states increases, the amount of experience required to learn about them can *grow combinatorially*
  - state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:
    - **state-space learning: Deep Q-Networks (DQN):**

DQN



Challenges

Challenges

- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***
  - **state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:**
    - **Hierarchical Reinforcement Learning (HRL): *policies vs. "options"***

Challenges

- Curse of dimensionality
  - as the number of states increases, the amount of experience required to learn about them can *grow combinatorially*
  - state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:
    - Hierarchical Reinforcement Learning (HRL): *policies vs. "options"*
- What if the world changes?

Challenges

- Curse of dimensionality
  - as the number of states increases, the amount of experience required to learn about them can *grow combinatorially*
  - state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:
    - Hierarchical Reinforcement Learning (HRL): *policies vs. "options"*
- What if the world changes?
  - Standard RL is stuck between between flexible and rigid
    - if the learning rate is high:
      - adjusts quickly, but may thrash

Challenges

- **Curse of dimensionality**
  - **as the number of states increases, the amount of experience required to learn about them can *grow combinatorially***
  - **state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:**
    - **Hierarchical Reinforcement Learning (HRL): *policies vs. "options"***
- **What if the world changes?**
  - **Standard RL is stuck between between flexible and rigid**
    - **if the learning rate is high:**
      - **adjusts quickly, but may thrash**
    - **if the learning rate is low:**
      - **integrates over more experience (wisdom), but is slow to adapt**

Challenges

- Curse of dimensionality
  - as the number of states increases, the amount of experience required to learn about them can *grow combinatorially*
  - state space abstraction: organize into meaningful *(generalizable)* "chunks:" states that share the same goals or subgoals:
    - Hierarchical Reinforcement Learning (HRL): *policies vs. "options"*
- What if the world changes?
  - Standard RL is stuck between between flexible and rigid
    - if the learning rate is high:
      - adjusts quickly, but may thrash
    - if the learning rate is low:
      - integrates over more experience (wisdom), but is slow to adapt
  - Model-free vs. model-based RL

Model-Free vs. Model-Based RL

- **Model-free** *("habit learning")***:**

Model-Free vs. Model-Based RL

- **Model-free** *("habit learning")*:
  - **Standard RL: learn "cached value" for each state**
    (i.e., summed, discounted future value of that state)

Model-Free vs. Model-Based RL

- **Model-free** *("habit learning")***:**
  - **Standard RL: learn "cached value" for each state**
    (i.e., summed, discounted future value of that state)
  - **Advantages:**
    - easy and quick decisions: pick the state with the greatest value
    - guaranteed to be optimal if the world is stable and observable

Model-Free vs. Model-Based RL

- **Model-free** *("habit learning")*:
  - **Standard RL: learn "cached value" for each state** *(i.e., summed, discounted future value of that state)*
  - **Advantages:**
    - easy and quick decisions: pick the state with the greatest value
    - guaranteed to be optimal if the world is stable and observable
  - **Problems:**
    - takes a long time to learn about all the states and dependencies
    - what if the world changes?

Model-Free vs. Model-Based RL

- Model-based (“deliberation”):

Model-Free vs. Model-Based RL

- **Model-based ("deliberation"):**
  - **Maintain a "mental model" of the value of each state on its own** *(i.e., not its future value, just the value of being in that state)*

Model-Free vs. Model-Based RL

- **Model-based (“deliberation”):**
  - **Maintain a “mental model” of the value of each state on its own** *(i.e., not its future value, just the value of being in that state)*
  - **When in a state, mentally simulate all paths forward, and pick the one that has the best cumulative (or ultimate) outcome**

Model-Free vs. Model-Based RL

- **Model-based ("deliberation"):**
  - **Maintain a "mental model" of the value of each state on its own** *(i.e., not its future value, just the value of being in that state)*
  - **When in a state, mentally simulate all paths forward, and pick the one that has the best cumulative (or ultimate) outcome**
  - **Advantage:**

Model-Free vs. Model-Based RL

- **Model-based ("deliberation"):**
  - **Maintain a "mental model" of the value of each state on its own** *(i.e., not its future value, just the value of being in that state)*
  - **When in a state, mentally simulate all paths forward, and pick the one that has the best cumulative (or ultimate) outcome**
  - **Advantage:**
    - flexible: easy to update the value of a state to its current value

Model-Free vs. Model-Based RL

- **Model-based ("deliberation"):**
  - **Maintain a "mental model" of the value of each state on its own** *(i.e., not its future value, just the value of being in that state)*
  - **When in a state, mentally simulate all paths forward, and pick the one that has the best cumulative (or ultimate) outcome**
  - **Advantage:**
    - flexible: easy to update the value of a state to its current value
  - **Problem:**
    - slow, effortful and possibly intractable to simulate all future paths

Supervised Learning: Scalar

- Conditioning
  - Simple Prediction
    - *Rescorla-Wagner Rule*
  - Stimulus-Action Associations
    - *Actor-critic model, Q Learning*
- Sequence Prediction
  - Method of Temporal Differences (TD)
  - Model-Free vs. Model-Based RL
- Challenges
  - Curse of dimensionality
    - *State space abstraction*
    - *Hierarchical RL: policies and options*
  - Explore-exploit
    - ***Meta-control***