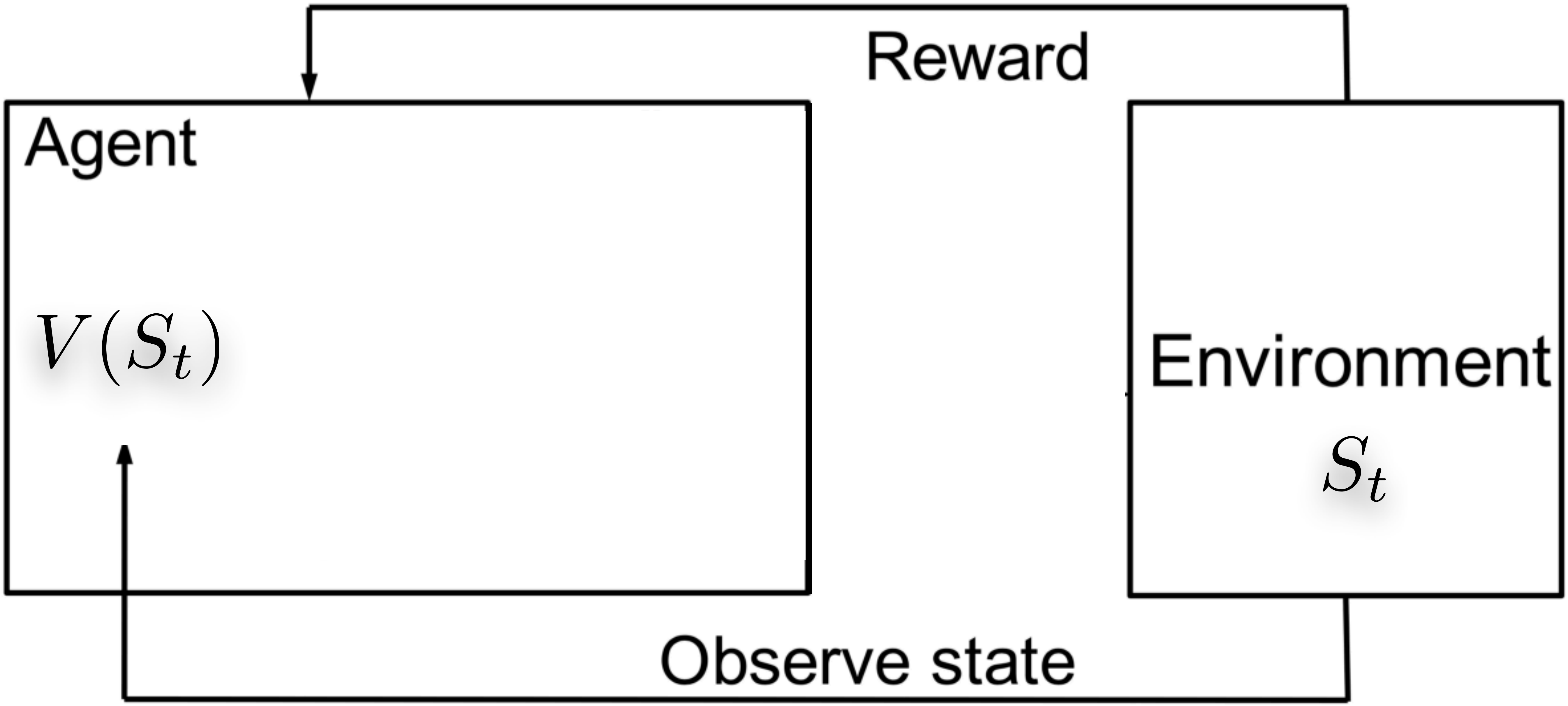
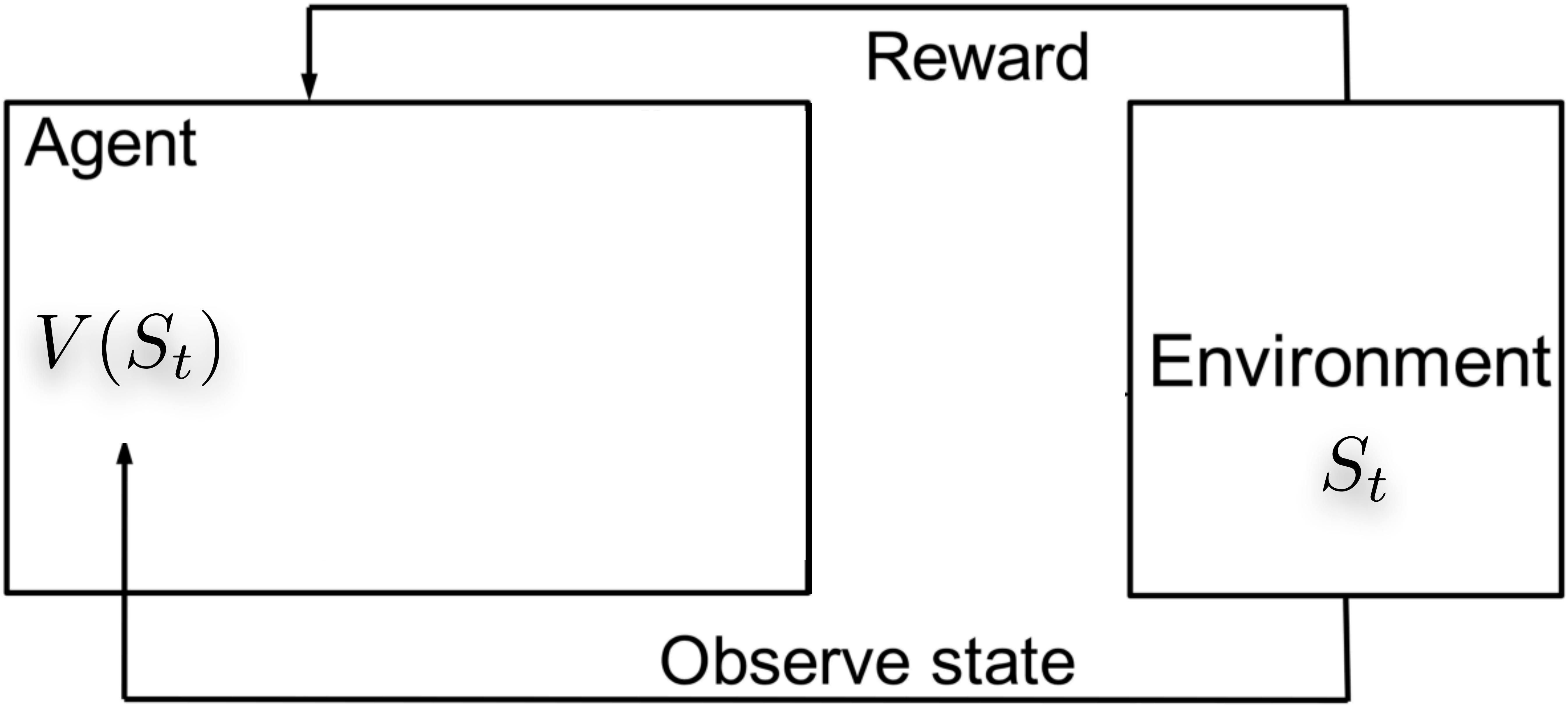
Reinforcement learning: Q Learning*

"Q" refers to the function that the algorithm computes
– the expected rewards for an action taken in a given state



Agent passively views the environment.



Agent passively views the environment.

Goal of learning: Minimize the error between predicted and actual reward.

Rescorla-Wagner Model: to learn reward associated with static state

$$V = w \cdot s$$

Temporal Difference model: to learn all future reward in a dynamic/uncertain environment

$$V(t) = \sum_{\tau} w(\tau) \cdot s(t - \tau)$$

Reward Prediction Error and Dopamine neuron signal

$$V = w \cdot s$$

$$\delta = r - V$$

$$w \leftarrow w + \epsilon \cdot \delta \cdot s$$

δ: prediction error

ϵ: learning rate

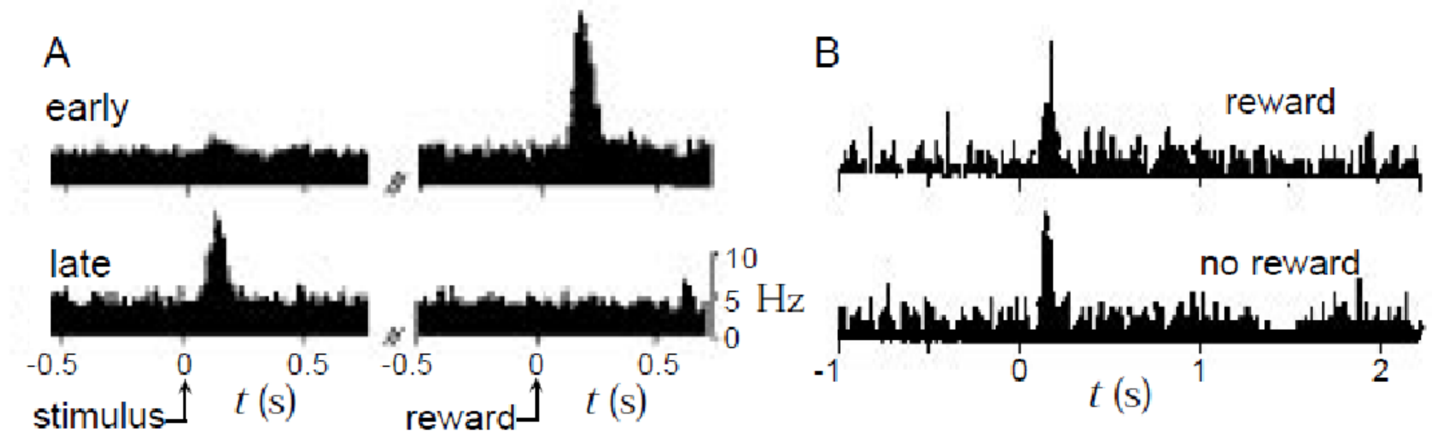
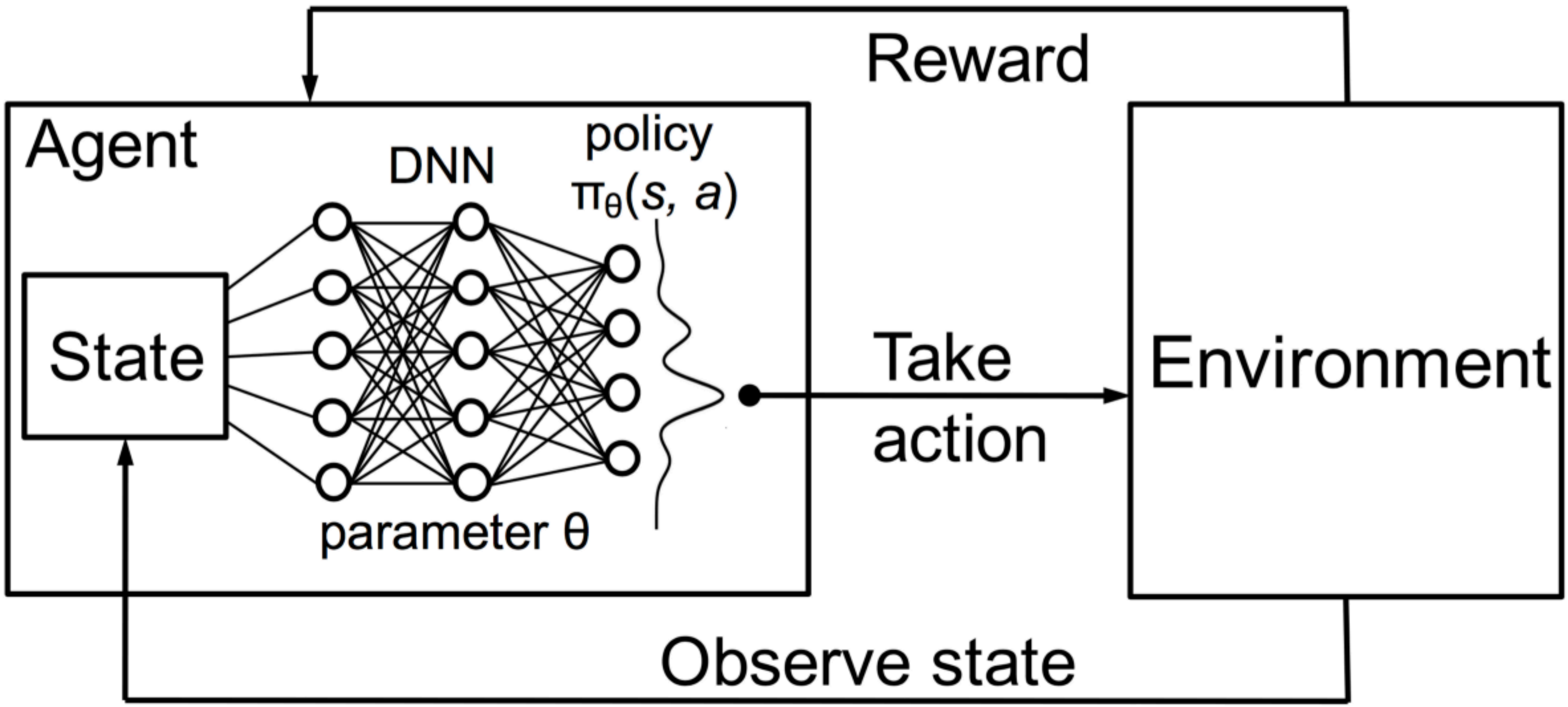


Figure 9.3: Activity of dopaminergic neurons in the VTA for a monkey performing a reaction time task. A) Histograms show the number of spikes per second for various time bins accumulated across trials and either time-locked to the light stimulus (left panels) or the reward (right panels) at the time marked zero. The top row is for early trials before the behavior is established. The bottom row is for late trials, when the monkey expects the reward on the basis of the light. B) Activity of dopamine neurons with and without reward delivery. The top row shows the normal behavior of the cells when reward is delivered. The bottom row shows the result of not delivering an expected reward. The basal firing rate of dopamine cells is rather low, but the inhibition at the time the reward would have been given is evident. (Adapted from Schultz, 1998.)

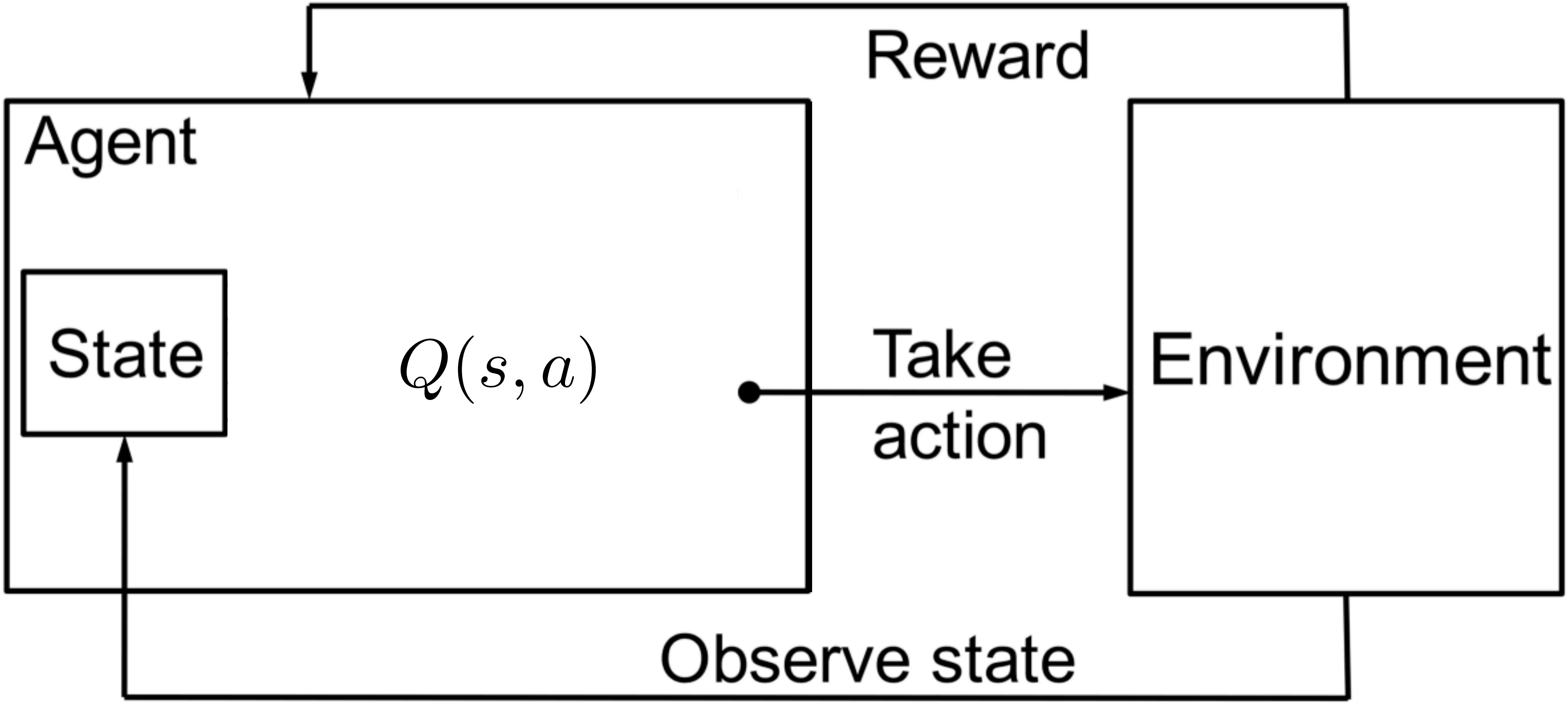


Agent allows interacting with the environment.

Goal of learning: Maximizing the all future reward by taking optimal action(s)

Q-learning (model-free and model-based)

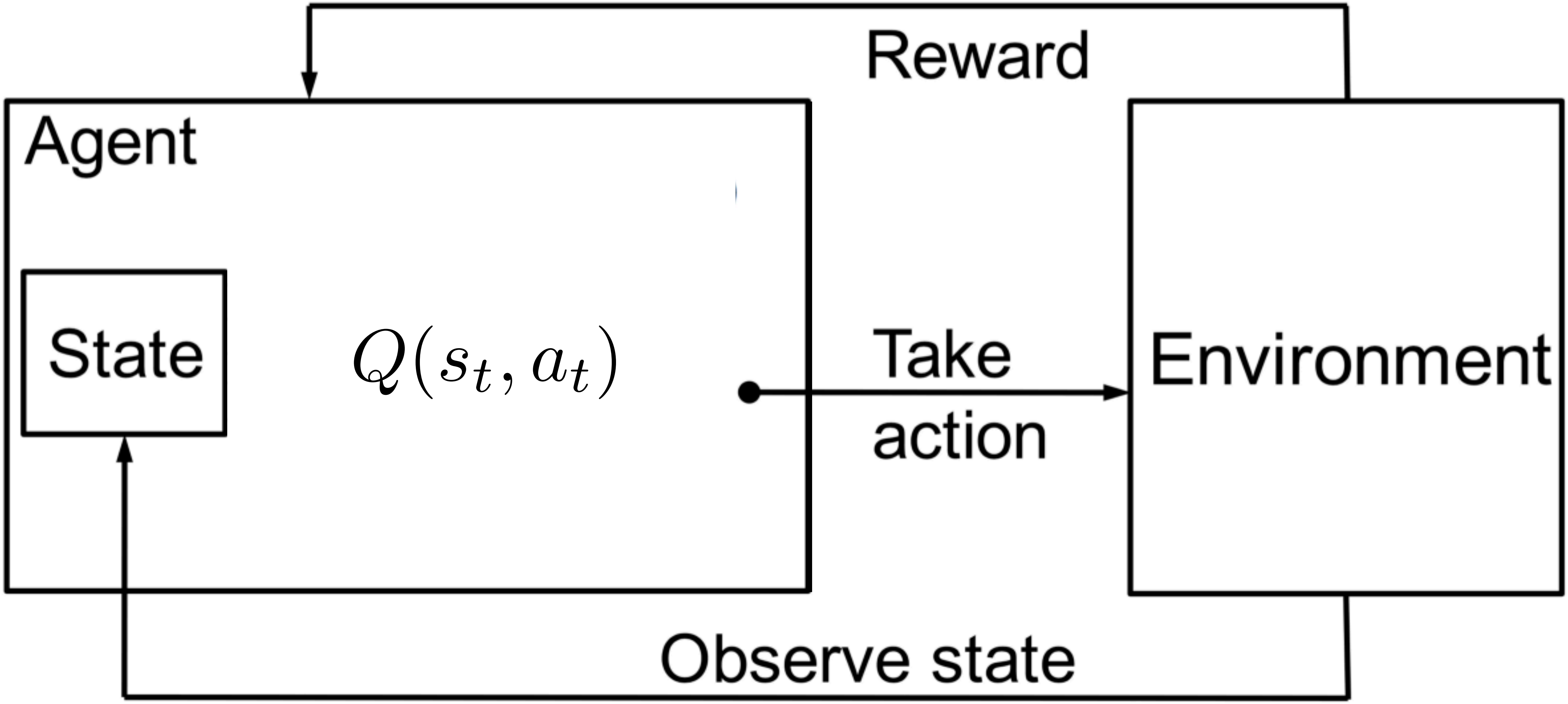
Algorithm: Q, SARSA, Actor-Critic, probabilistic (Kalman filter)



interacting with the static environment.

Goal of learning: Maximizing the all future reward by taking optimal action(s)

Action policy π $\quad$ $a_t^\star = \max_a Q(s_t, a)$ $\quad$ $s_t \xrightarrow{a^\star} s_{t+1}$



interacting with the dynamic environment

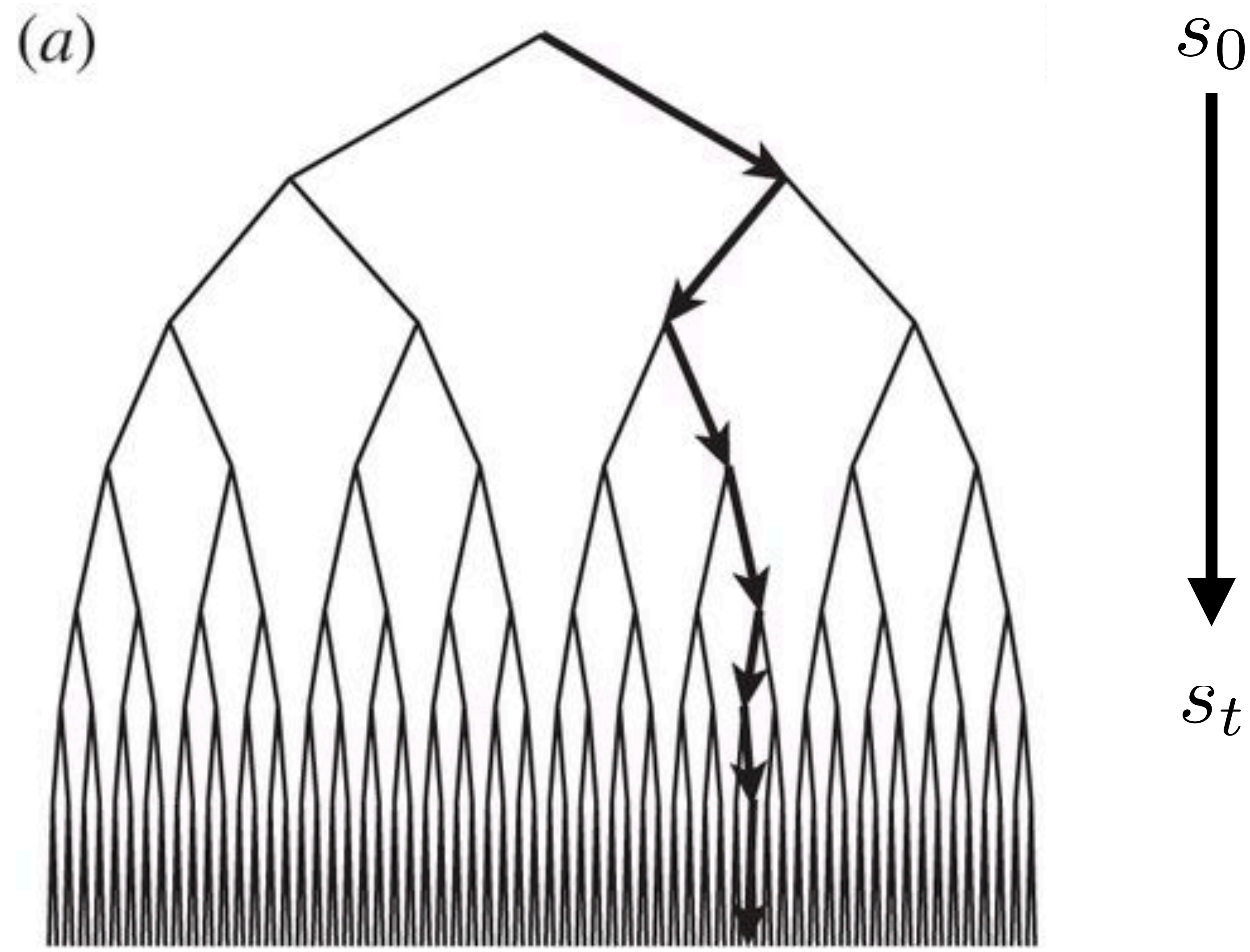
Q-learning rules

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

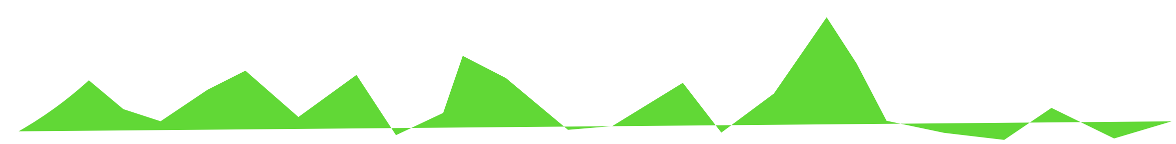
Action policy

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

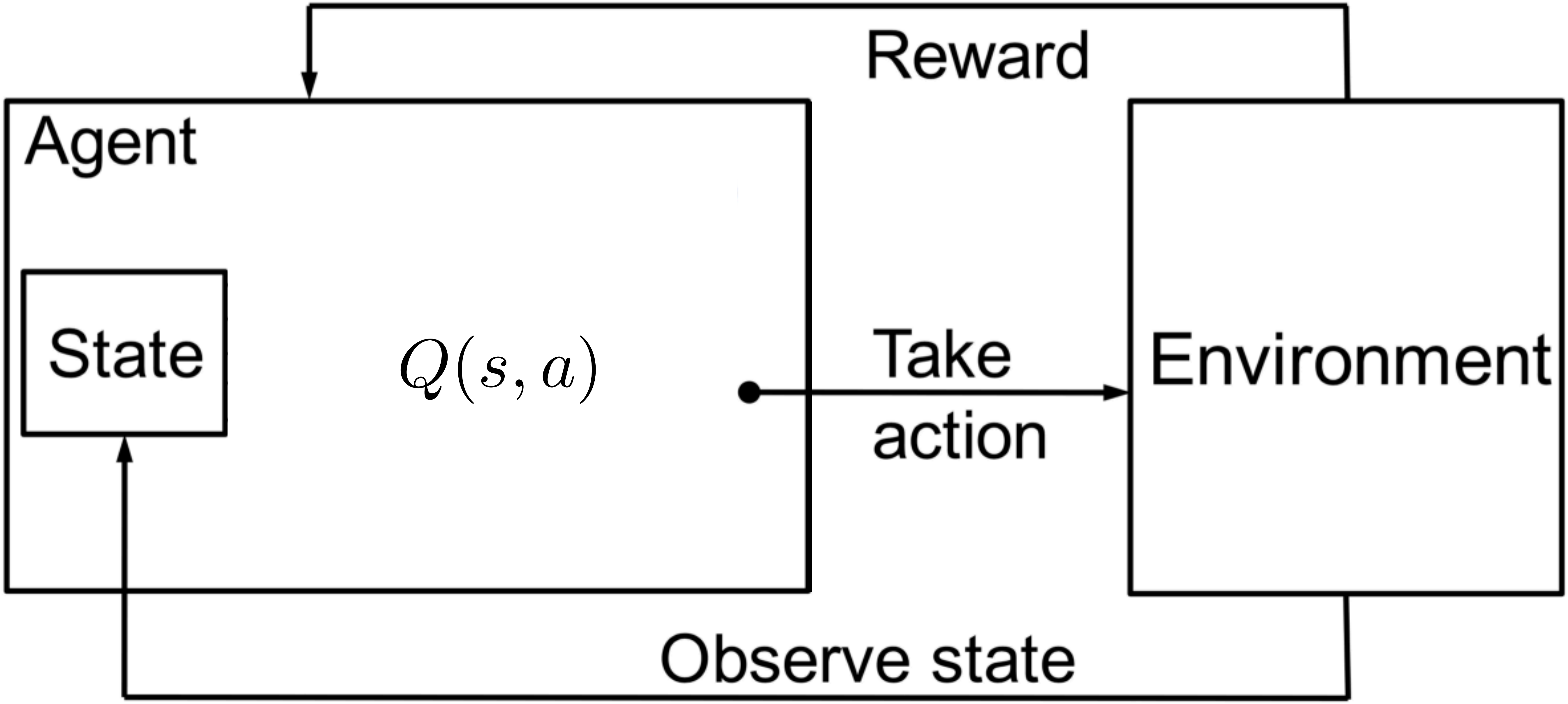
$$s_t \xrightarrow{a^\star} s_{t+1}$$



Cumulated Q value



$$\underbrace{Q^\pi(s_t, a_t)}_{\text{Q value for that state given that action}} = \underline{E}[\underbrace{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots}_{\text{Expected discounted cumulative reward ...}} | \underbrace{s_t, a_t}_{\text{given that state and that action}}]$$



interacting with the static environment.

Action policy π $\qquad a_t^\star = \max_a Q(s_t, a)$

Q-learning rules

$$Q(s_t,a_t) \leftarrow \underbrace{Q(s_t,a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_{t+1}}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1},a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t,a_t)}_{\text{old value}} \right)$$

Q(s, a) table for GO-NOGO task

action / stimulus	Go	NoGo
odor 1	R+	R-
odor 2	R-	R-
odor 3	R+	R-
odor 4	R-	R-

Q(s, a) table for GO-NOGO task

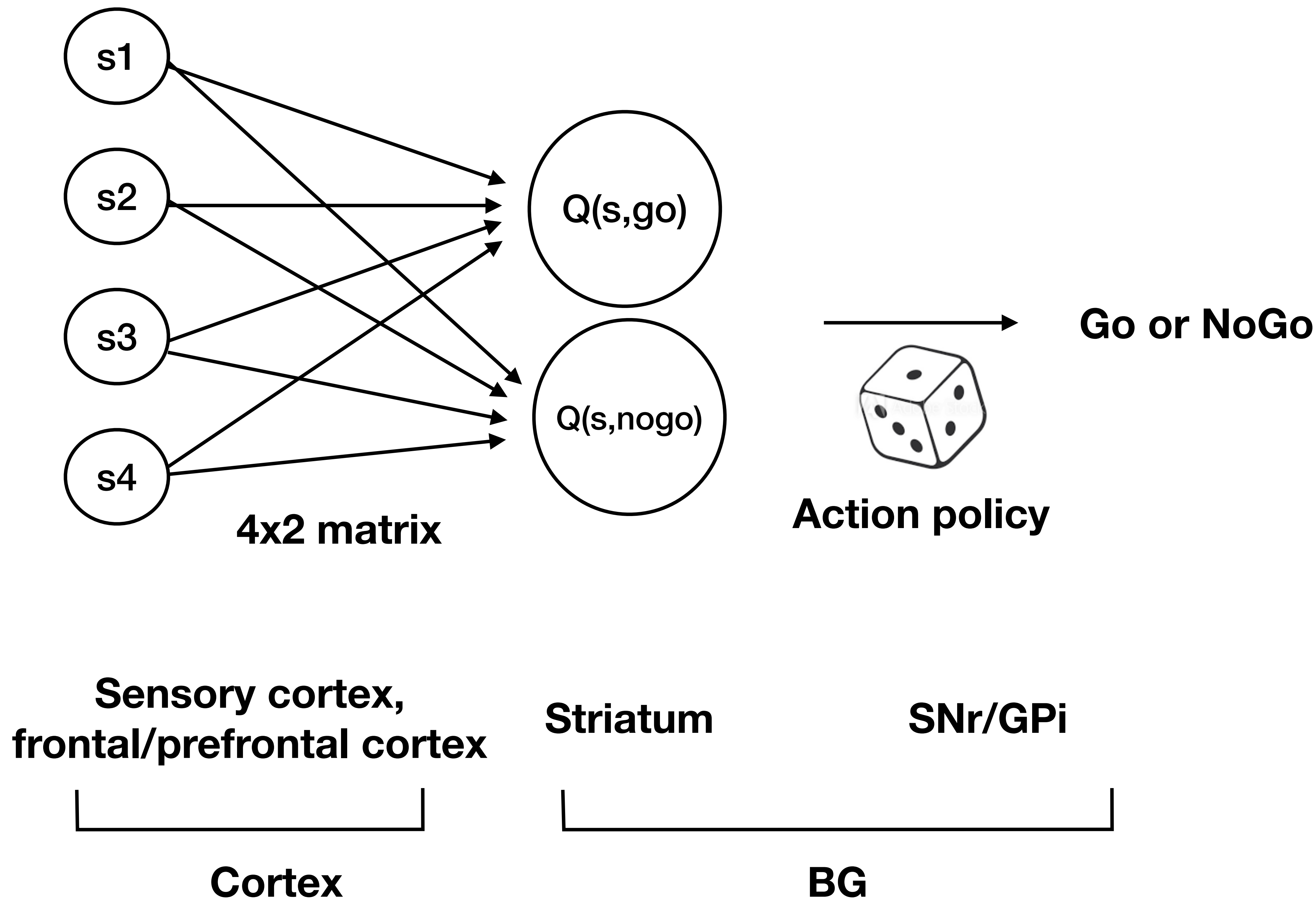
Q(s, a)	**Go**	**NoGo**
odor 1	R+	R-
odor 2	R- go cost	R-
odor 3	R+, Stim	R-
odor 4	R- go cost	R-, Stim

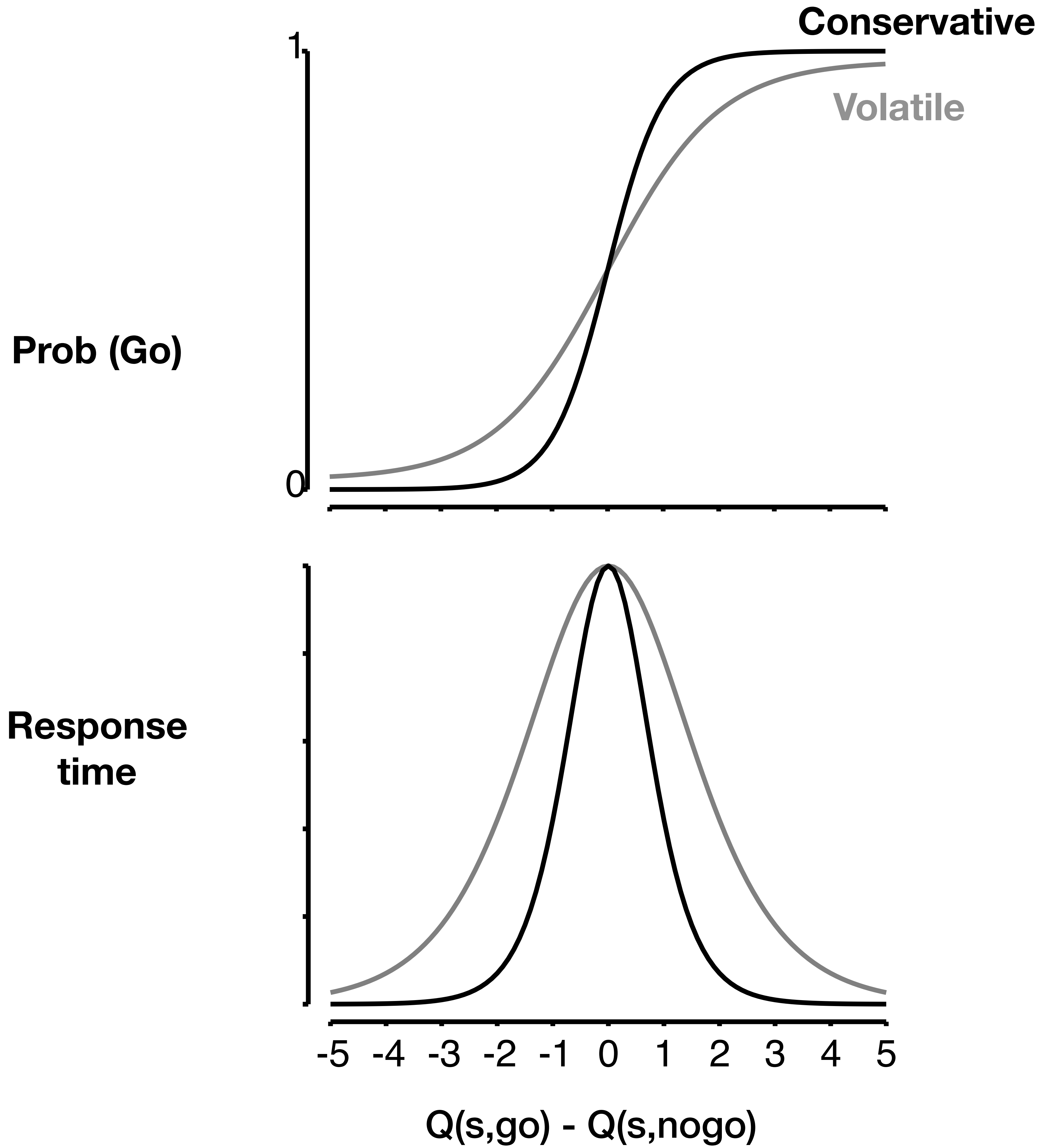
Q(s, a) table for GO-NOGO task

Assign values:
Reward: 1
Cost of Go: 0.1
Stim: 0.2

Q(s, a)	Go	NoGo
odor 1	1	0
odor 2	0 - 0.1	0
odor 3	1 + 0.2	0
odor 4	0 - 0.1	0 + 0.2

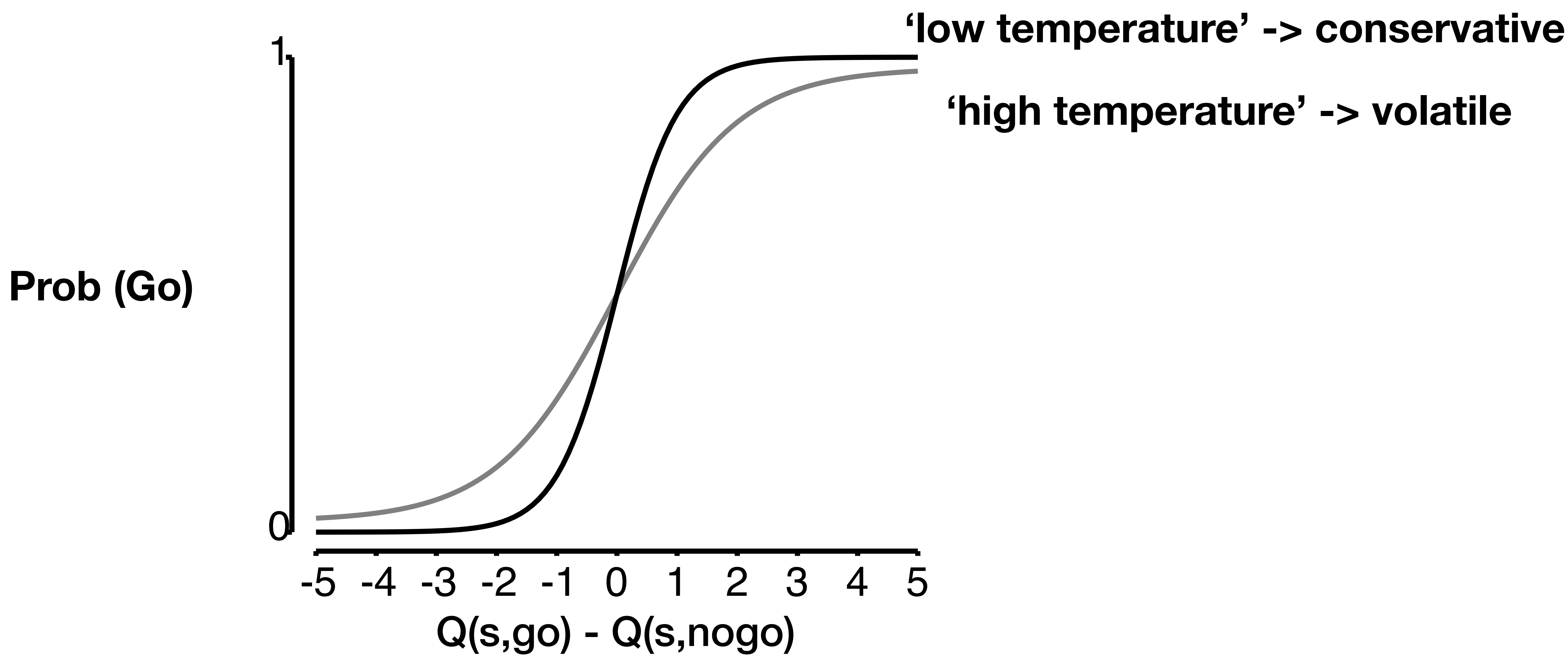
Q value and action selection





Conservative
Volatile
1
0
Prob (Go)
Response time
-5 -4 -3 -2 -1 0 1 2 3 4 5
Q(s,go) - Q(s,nogo)

Action policy



Action policy: Softmax

$$P(\mathrm{go}) = \frac{\exp(Q(s,\mathrm{go})/kT)}{\exp(Q(s,\mathrm{go})/kT) + \exp(Q(s,\mathrm{nogo})/kT)}$$

Updating the Q(s,a) with trial-by-trial sampling

Initial value

Q(s, a)	Go	NoGo
odor 1	0.2	0
odor 2	0.2	0
odor 3	0.2	0
odor 4	0.2	0

after learning

Q(s, a)	Go	NoGo
odor 1	1	0
odor 2	0 - 0.1	0
odor 3	1 + 0.2	0
odor 4	0 - 0.1	0 + 0.2

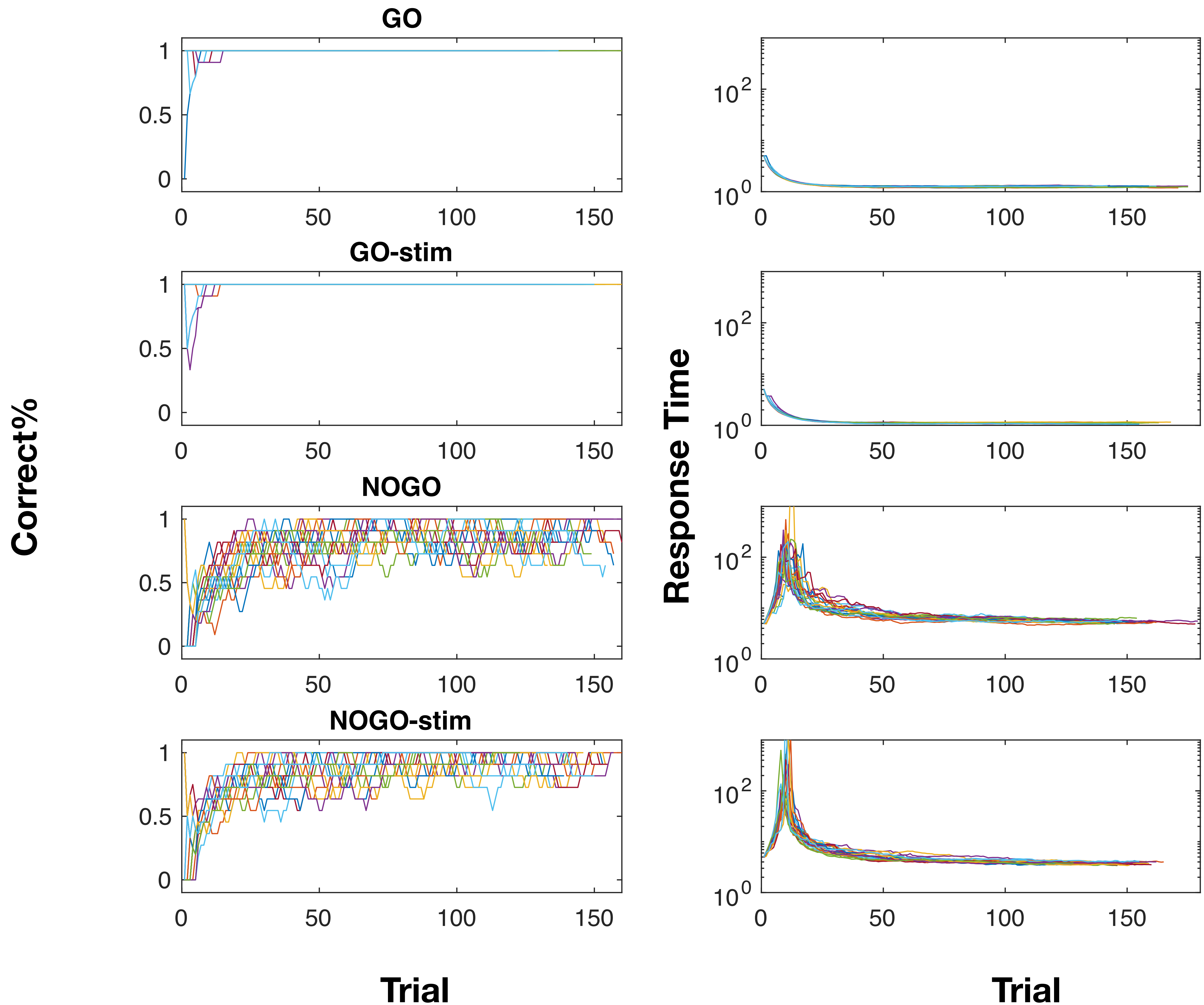
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) - Q(s,a)]$$

$$P(\mathrm{go}) = \frac{\exp(Q(s,\mathrm{go})/kT)}{\exp(Q(s,\mathrm{go})/kT) + \exp(Q(s,\mathrm{nogo})/kT)}$$

Parameters

α **: learning rate**

$\beta = 1/kT$ **: volatility**



GO
GO-stim
NOGO
NOGO-stim
Correct%
Response Time
Trial
Trial

Under the framework of Q-learning, What does the behavioral tell us?

- Animals' initial Q value is higher for Go than NoGo
- The learning curve was determined by the Q(s, a) at animal's best estimate.
- The relation between reaction times and Q(s,a).

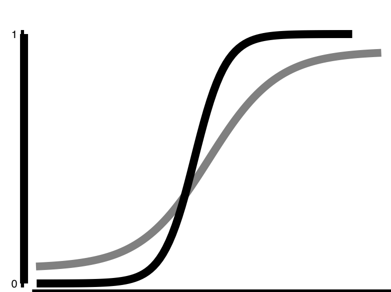
what to do with the model?

Infer **value** and **parameters** in the task:

- D1/D2 stimulation (reward, prediction error, or something else)
- Value of water, Cost of Go, cost of waiting in the Go
- Individual difference: learning speed (learning rate), stability(‘temperature’)

Design experiment to test or manipulate the elements:

- ‘Temperature’

  Learning rate