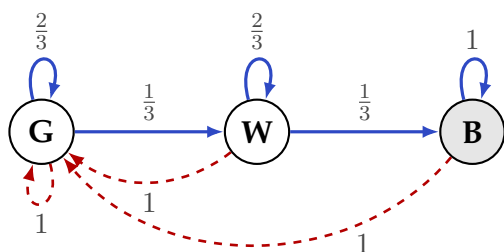


Lecture 11: Off-Policy Control: Q-Learning

How to use these notes. Board content in board order: figures, equations, tables, one-line statements. Today's task is *control from a stream whose actions are not the learner's own*: a log kept by another manager, or the learner's own exploring coin; the table must still end at Q^* . The note stands on its own: every symbol is defined where it is first used. Reading: [Sutton and Barto \(2018, Sections 6.5 and 6.7, Example 6.6\)](#); optional, [Szepesvári \(2010, Section 4.3.1\)](#). Throughout the course, the notes draw on the public materials of Stanford CS 234 ([Brunskill, 2026](#)), UIUC CS 443 ([Jiang, 2026](#)), and Cornell CS 4789 ([Sun, 2025](#)); some examples are adapted from them.

The Task: Control From Any Stream

A workshop runs one machine. Each week it is good (G), worn (W), or broken (B), and it is run or repaired. Wanted: the best rule, with P and r hidden. New today: the actions in the data were chosen by someone else, or by a coin. The machine's numbers below are the truth, for checking; the learner never sees them.



The machine, both actions. Run (solid): a good machine earns 8 and stays good w.p. $\frac{2}{3}$, else wears; a worn one earns 4 and stays worn w.p. $\frac{2}{3}$, else breaks; a broken one earns 0 and stays broken. Repair (dashed): costs 10, leaves it good. Gray: hidden from the learner.

The truth, for checking	G	W	B
$Q^*(s, \text{run})$	44.30	32.41	26.89
$Q^*(s, \text{repair})$	29.87	29.87	29.87
$\pi^*(s)$: the larger entry	run	run	repair
$V^*(s) = \max_a Q^*(s, a)$	44.30	32.41	29.87

"Repair when broken" is optimal; a wrong rule at B costs $29.87 - 26.89$ or worse. Known to the learner: $\mathcal{S} = \{G, W, B\}$, $\mathcal{A} = \{\text{run}, \text{repair}\}$, $\gamma = 0.9$, and the stream with its actions, whoever chose them. Hidden: P (18 numbers) and r (6).

Control: what is wanted (stated) (Sutton and Barto, 2018, Sections 3.5–3.6). The machine is a Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$: action a in state s pays a reward with mean $r(s, a)$ (on the machine, exactly $r(s, a)$) and moves to s' with probability $P(s' | s, a)$; $\gamma \in [0, 1)$ discounts; a policy $\pi(a | s)$ draws the action, written $\pi(s)$ when deterministic.

$$Q^\pi(s, a) := \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \dots | S_t = s, A_t = a] \quad \text{under } \pi: \text{do } a \text{ now, follow } \pi \text{ after}$$

$$= r(s, a) + \gamma \sum_{s'} P(s' | s, a) \sum_{a'} \pi(a' | s') Q^\pi(s', a') =: (\mathcal{T}^\pi Q^\pi)(s, a)$$

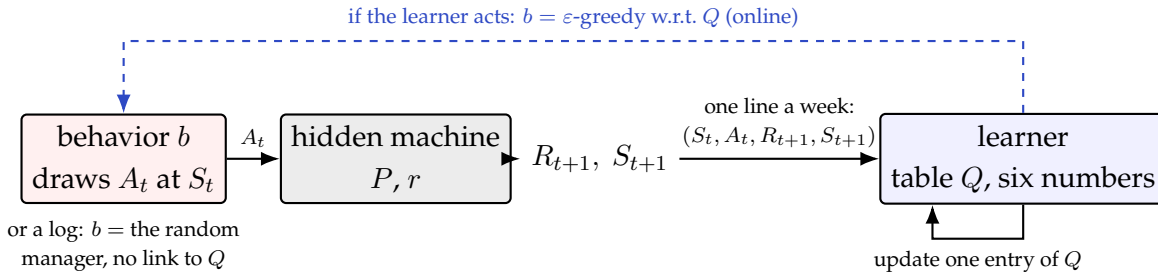
Bellman's equation on pairs; $\mathcal{T}^\pi Q$ is the same backup of any table Q

$$Q^*(s, a) := \max_{\pi} Q^\pi(s, a) = r(s, a) + \gamma \sum_{s'} P(s' | s, a) \max_{a'} Q^*(s', a') =: (\mathcal{T} Q^*)(s, a)$$

the optimality equation; the max over rules is attained

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a); \quad \|Q\|_\infty := \max_{s,a} |Q(s, a)| \quad \text{the best rule is a lookup; the max norm}$$

Both backups contract in the max norm, $\|\mathcal{T}^\pi Q - \mathcal{T}^\pi Q'\|_\infty \leq \gamma \|Q - Q'\|_\infty$ and the same for $\mathcal{T}$, so each equation has exactly one solution (Banach, 1922): **a table that solves $Q = \mathcal{T}Q$ is Q^* , however it was found.** Rewards lie in $[-10, 8]$, so $|Q| \leq 10/(1 - \gamma) = 100$. Today P and r are hidden, and the actions are someone else's.



Control from any stream. The actions come from a behavior policy b , the rule that draws A_t : the learner's own exploring rule (dashed link), or another manager's log (no link). Either way the learner reads one transition a week and updates one of its six numbers.

The data: one stream, the actions drawn by b .

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots, \quad A_t \sim b(\cdot | S_t)$$

b : whatever rule drew the actions; it may change from week to week

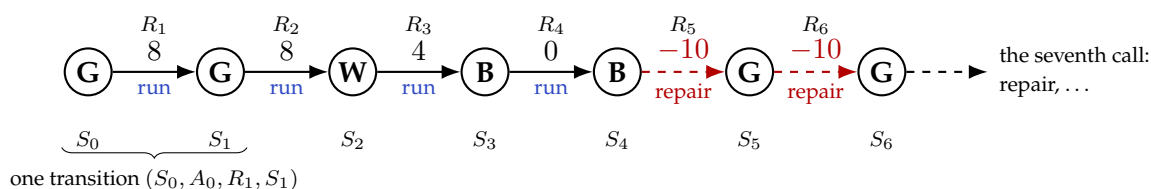
$$S_{t+1} \sim P(\cdot | S_t, A_t), \quad R_{t+1} = r(S_t, A_t) \text{ on the machine}$$

one step of the hidden machine; in general R_{t+1} has mean $r(S_t, A_t)$

- Two behaviors today; a third, once, shows what fails. *The random manager: $b(\text{run} | s) = b(\text{repair} | s) = \frac{1}{2}$, a coin flip every week, written down as a log. The learner's own*

coin: b is ε -greedy w.r.t. its current table.

- ε -greedy w.r.t. Q : the greedy action $\arg\max_a Q(s, a)$ w.p. $1 - \varepsilon + \varepsilon/|\mathcal{A}|$, each other action w.p. $\varepsilon/|\mathcal{A}|$; on the machine with $\varepsilon = 0.1, 0.95$ and 0.05 ; at a tie the greedy mass is split.
- One *transition*: $(S_t, A_t, R_{t+1}, S_{t+1})$. Today's update reads nothing further, not even the next action.
- The random manager is worth Q^b : run $(-6.72, -13.66, -19.38)$, repair -23.68 , a loss of 1.62 a week (hidden too). Wanted: a table whose greedy policy is π^* , from his log.



Six weeks of the random manager's log, used by hand below. Every action is his coin: four runs, then three repairs, two of them on a good machine. The learner reads the log; it chose nothing.

Control from a stream whose actions follow someone else's rule: which target heads to Q^* whatever that rule was, what must the rule still do, what does acting on Q^* while exploring cost, and when does the max lie?

Today's targets.

1. Q-learning: the target $R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$ has mean $(\mathcal{T}Q)(s, a)$ for every behavior; settled with every pair visited forever, the table is Q^* . Theorem stated. *Off-policy*: b decides which pair is updated, never what it moves toward.
2. By hand, six weeks of the log: targets 8, 8, 4, 0, -4.6 , -4.6 ; SARSA's read -10 twice, the manager's waste billed to the pair before it. By computer, one log, two limits: 2.3 from Q^* , or 3.7 from Q^b .
3. Cliff Walking: Q-learning's table says the edge (-13) , SARSA's the top row (-17) ; walking with the coin, Q-learning's walker earns -51 an episode, SARSA's -27 : the better table, the worse walker.
4. Maximization bias: $\mathbb{E}[\mathcal{T}Q_t] \geq \mathcal{T}(\mathbb{E}[Q_t])$ across runs. Double Q separates selection and evaluation. Replay of a fixed log lands on the certainty-equivalence table (stated).

Q-Learning: the Optimality Backup, Sampled

Two targets at one transition (Rummery and Niranjan, 1994; Watkins, 1989). For any table Q and a transition $(S_t, A_t, R_{t+1}, S_{t+1})$ with $A_t \sim b$, call the quantity an entry moves toward its *target*. Two choices:

$$\text{SARSA: } R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}), \quad A_{t+1} \sim b(\cdot \mid S_{t+1}); \quad \mathbb{E}[\cdot \mid s, a] = (\mathcal{T}^b Q)(s, a)$$

R has mean $r(s, a)$, $S' \sim P(\cdot \mid s, a)$, $A' \sim b$: the backup of b , so the table heads to Q^b : on-policy

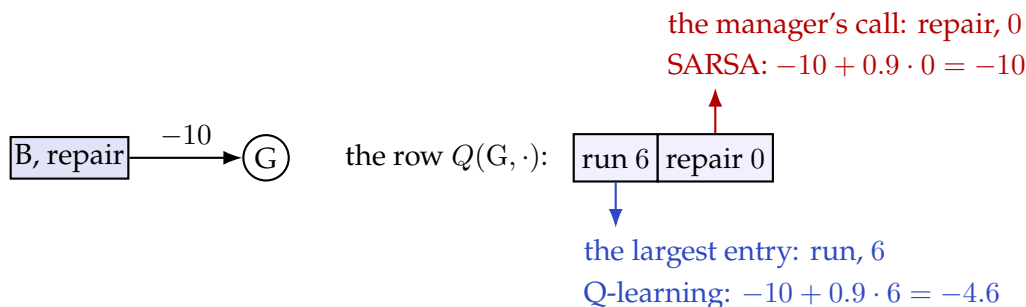
$$\text{Q-learning: } R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$$

the max is a function of S' alone; no next action is drawn: b appears nowhere

$$\mathbb{E}[\cdot \mid s, a] = r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) \max_a Q(s', a) = (\mathcal{T}Q)(s, a)$$

$S' \sim P(\cdot \mid s, a)$: the optimality backup, sampled

b decides *which* pair is updated, never *what* the entry moves toward. Off-policy: the target evaluates a policy other than b , here the greedy one.



The same transition, two targets. A repair at B lands on G , where a table says $(6, 0)$. SARSA reads the entry of the manager's next call, here a repair of a good machine; Q-learning reads the largest entry. The manager's waste is billed to the pair before it, or not.

Q-learning (Watkins, 1989), (Sutton and Barto, 2018, Section 6.5).

- Input: a stream whose actions come from any b that keeps trying every pair; a step size $\alpha \in (0, 1]$. Output: a table Q , read as the policy $\arg\max_a Q(s, a)$.
- Start: $Q_0(s, a) = 0$ for every pair (if episodes end, $Q(\text{terminal}, \cdot) = 0$ always); observe S_0 .

- Each week t : A_t from b (ε -greedy w.r.t. Q_t , or the log's line); observe R_{t+1}, S_{t+1} ; then

$$\delta_t = R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)$$

the TD error, the greedy continuation inside

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha \delta_t \quad \text{the template at one entry; the other five unchanged}$$

- *Off-policy*: the action inside the target is the greedy one; the action taken is b 's. The next action is not needed: the quadruple S, A, R, S' is enough.
- Nothing stored but Q : memory $|S||\mathcal{A}|$; $|\mathcal{A}|$ lookups a week for the max.

The expected update, and where Q-learning can stop (Sutton and Barto, 2018, Section 6.5).

$$\mathbb{E}[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) \mid Q_t, S_t = s, A_t = a] = (\mathcal{T}Q_t)(s, a)$$

the line above; the behavior does not enter

$$\mathbb{E}[Q_{t+1}(s, a) \mid Q_t, S_t = s, A_t = a] = (1 - \alpha) Q_t(s, a) + \alpha (\mathcal{T}Q_t)(s, a)$$

damped value iteration at one entry, from a sample; contracts toward Q^* , modulus $1 - \alpha(1 - \gamma)$ (stated)

$$\text{settled at } \hat{Q}, \text{ every pair visited forever: } \mathbb{E}[\delta_t \mid s, a] = (\mathcal{T}\hat{Q})(s, a) - \hat{Q}(s, a) = 0$$

a nonzero mean increment repeated forever would move the entry

$$\hat{Q} = \mathcal{T}\hat{Q} \Rightarrow \hat{Q} = Q^* \quad \text{the optimality equation has one solution; the same for every } b \quad \square$$

SARSA's settled table solves $\hat{Q} = \mathcal{T}^{\hat{b}}\hat{Q}$ instead, $\hat{b}$ being the rule that draws the actions once the table has settled, so it is $Q^{\hat{b}}$: on the random manager's log, $\hat{b} = b$ and the table is Q^b ; with the learner's coin, $\hat{b}$ is ε -greedy w.r.t. $\hat{Q}$ itself, written π_ε , and the table is Q^{π_ε} , 4.04 from Q^* on the machine (stated).

Theorem (stated): Q-learning converges (Watkins and Dayan, 1992; Tsitsiklis, 1994; Jaakkola et al., 1994). Rewards bounded; every pair (s, a) visited infinitely often; at every pair the step sizes satisfy $\sum \alpha = \infty, \sum \alpha^2 < \infty$ (Robbins and Monro, 1951). Then $Q_t \rightarrow Q^*$ w.p. 1, whatever policy drew the actions. Proof by stochastic approximation; not given.

- *Coverage* is all that is asked of b : every pair keeps appearing. A fixed coin, a log, or the learner's ε -greedy rule with a fixed ε all serve; b never has to become greedy.
- On the machine every ε -soft b ($b(a \mid s) \geq \varepsilon/|\mathcal{A}|$) visits every pair forever: repair leads to G, and run from G reaches W and B.
- Not asked: that b be good. The random manager loses 1.62 a week; his log still yields Q^* .

Example: what must the behavior still do? The log of a manager who repairs whenever the machine is worn or broken. Which entries ever move, and what does the table say at W?

The machine never breaks: every worn week is a repair. The log holds (G, run) and (W, repair) only: $Q(\text{G}, \text{run}) \rightarrow 38.46$, $Q(\text{W}, \text{repair}) \rightarrow 24.62$, four entries stay 0. At W the table compares 24.62 with an untouched 0: repair, the log's own rule; the truth says run, 32.41 against 29.87. Q-learning cannot disagree with a log where the other action was never tried.

By Hand: the Random Manager's Six Weeks

Example: Q-learning by hand. $\alpha = \frac{1}{2}$, $Q = 0$, the log of the figure. Fill the table one week at a time: the target and the error first, then the one entry that moves. Beside each week, SARSA's target with the manager's next call: where do the two differ?

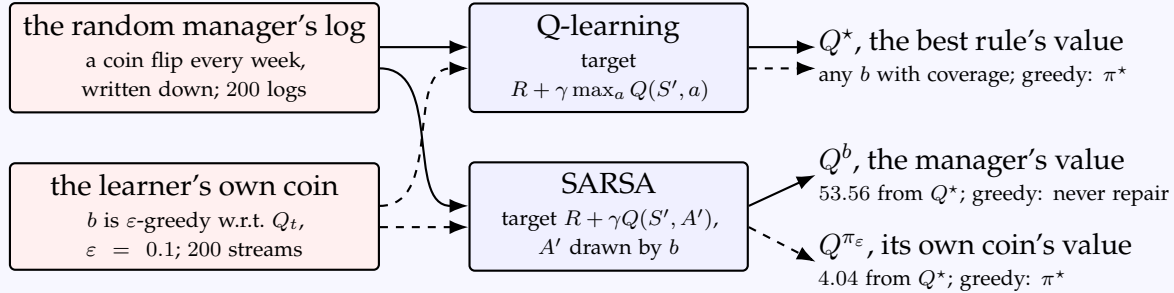
week t	$(S_t, A_t, R_{t+1}, S_{t+1})$	Q-learning target	δ_t	$Q(\cdot, \text{run})$	$Q(\cdot, \text{repair})$	A_{t+1}	SARSA's target
0	(G, run, 8, G)	$8 + 0.9 \cdot 0 = 8$	8	(4, 0, 0)	(0, 0, 0)	run	$8 + 0.9 \cdot 0 = 8$
1	(G, run, 8, W)	$8 + 0.9 \cdot 0 = 8$	4	(6 , 0, 0)	(0, 0, 0)	run	$8 + 0.9 \cdot 0 = 8$
2	(W, run, 4, B)	$4 + 0.9 \cdot 0 = 4$	4	(6, 2 , 0)	(0, 0, 0)	run	$4 + 0.9 \cdot 0 = 4$
3	(B, run, 0, B)	$0 + 0.9 \cdot 0 = 0$	0	(6, 2, 0)	(0, 0, 0)	repair	$0 + 0.9 \cdot 0 = 0$
4	(B, repair, -10, G)	$-10 + 0.9 \cdot 6 = -4.6$	-4.6	(6, 2, 0)	(0, 0, -2.3)	repair	$-10 + 0.9 \cdot 0 = -10$
5	(G, repair, -10, G)	$-10 + 0.9 \cdot 6 = -4.6$	-4.6	(6, 2, 0)	(-2.3 , 0, -2.3)	repair	$-10 + 0.9 \cdot 0 = -10$
truth Q^* , for comparison				(44.30, 32.41, 26.89)	(29.87, 29.87, 29.87)		

Six weeks, one update each; the entry that moves is bold, and week 3 moves nothing (a zero priced against zeros); the table columns show the entries after the update. Q-learning's target $R_{t+1} + 0.9 \max_a Q(S_{t+1}, a)$ reads the largest entry of the row at S_{t+1} : 0 at an untouched row, 6 at G from week 4 on. SARSA's target $R_{t+1} + 0.9 Q(S_{t+1}, A_{t+1})$ agrees whenever the manager's next call is that entry or the row is all zeros, and differs (red) at weeks 4 and 5: the next call was a repair of a good machine.

- The verdict: run (6, 2, 0), repair (-2.3, 0, -2.3); greedy says never repair. A repair was priced as a 10 bill plus a good machine the table valued at 6: too early, not the manager's fault.
- Weeks 4 and 5: SARSA bills the manager's next waste, $-10 + 0.9 Q(\text{G}, \text{repair}) = -10$, to the pair before it; Q-learning bills the best continuation the table knows, $-10 + 0.9 Q(\text{G}, \text{run}) = -4.6$.
- Run on its own table, SARSA ends with repair entries (-5, 0, -5): the price of a repair under the random manager. By computer the two tables part for good: repair toward Q^b 's -23.68, or toward Q^* 's 29.87.

By Computer: Two Limits From One Log

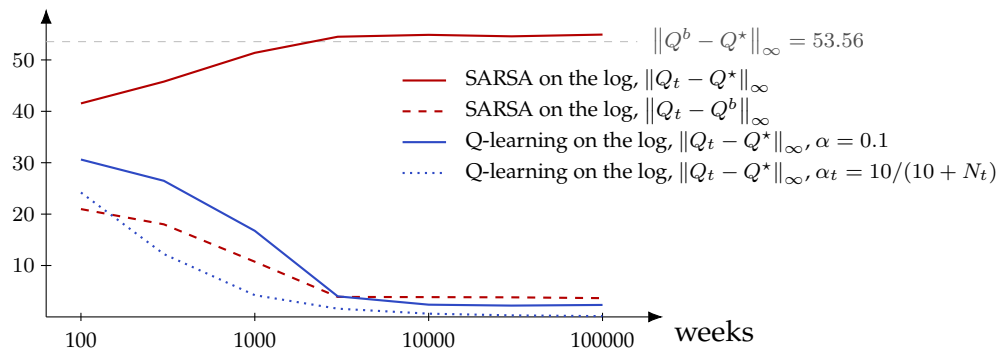
The experiment: two behaviors, two learners, and where each table is headed.



- Both learners on both behaviors: $Q_0 = 0, \alpha = 0.1, 100,000$ weeks from G, one pass; ties in the coin broken at random. Solid arrows: the log. Dashed: the coin.
- After t weeks read the greedy policy off Q_t and the distances $\|Q_t - Q^*\|_\infty, \|Q_t - Q^b\|_\infty$; report the share of logs or streams with π^* , the mean distances, and the behavior's earnings per week.

learner, data; week $t =$	greedy policy of Q_t is π^* : share			mean $\ Q_t - Q^*\ _\infty$			$\ Q_t - Q^b\ _\infty$ at 100,000	b earns per week
	1,000	10,000	100,000	1,000	10,000	100,000		
Q-learning, the random manager's log	0.570	1.000	1.000	16.75	2.38	2.33	52.87	-1.62
SARSA, the random manager's log	0.000	0.025	0.005	51.38	54.88	54.93	3.65	-1.62
Q-learning, the log, $\alpha_t = 10/(10 + N_t(s, a))$	1.000	1.000	1.000	4.23	0.63	0.15	53.51	-1.62
Q-learning, its own coin, $\varepsilon = 0.1$	1.000	0.995	1.000	20.25	2.55	2.65	—	3.35
SARSA, its own coin, $\varepsilon = 0.1$	1.000	0.990	1.000	22.62	6.85	6.85	—	3.34
Q-learning, the repair-when-worn log	0.000	0.000	0.000	32.41	32.41	32.41	—	3.50

Two learners, three behaviors, 200 logs or streams each. On the random manager's log the two tables go to different places: Q-learning's to Q^* , SARSA's to Q^b , 53.56 from Q^* . With the learner's own coin both read π^* off the table: all 200 streams at weeks 1,000 and 100,000, 199 and 198 at 10,000. The repair-when-worn log moves two entries and leaves four at 0; $N_t(s, a)$ counts the visits to the pair.

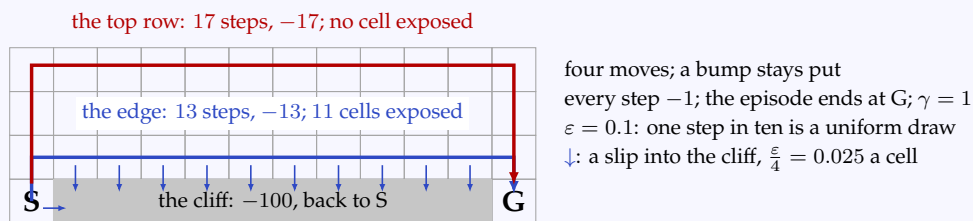


The random manager's log, the week axis logarithmic. Q-learning's distance to Q^* falls to the constant step's floor, lower still with a shrinking step; SARSA's distance to Q^b falls instead.

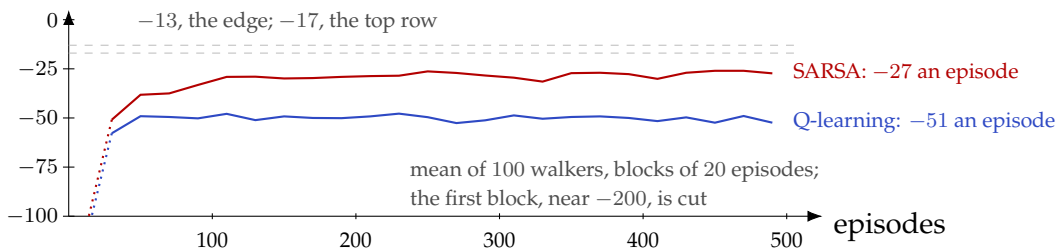
- Q-learning on the log: π^* in all 200 logs at weeks 10,000 and 100,000; 2.33 from Q^* , the constant step's floor; 0.15 and falling with the shrinking step. A log that loses 1.62 a week teaches the rule earning 3.71.
- SARSA on the same log: 3.65 from Q^b , 54.9 from Q^* ; its greedy rule is never repair, as Q^b 's is. One log, two questions: how good is the best rule, how good is this manager.
- Its own coin: Q-learning 2.65 from Q^* , SARSA 6.85 (4.04: the fixed point Q^{π_ϵ}); π^* read off both tables in all 200 streams at 100,000; earnings 3.35, 3.34 a week. The coin sets the earnings, the target sets the table.
- The repair-when-worn log: two entries move, four never; the rule at W is the log's; 32.41 from Q^* forever. Coverage failed, not the algorithm.

Cliff Walking: the Better Table, the Worse Walker

Cliff Walking (Sutton and Barto, 2018, Example 6.6). Two paths from S to G, the edge in 13 steps and the top row in 17; which is better depends on who walks it.



By computer: 100 walkers, $\epsilon = 0.1$, $\alpha = 0.5$, $Q_0 = 0$, 500 episodes, both learners; each episode's return, its cliff falls, and the greedy path read off the final table.



Return per episode while learning, the exploring walker's earnings; dashed, the two greedy paths.

episodes 401-500	return per episode	cliff falls per episode	greedy path read off the final table, of 100 walkers
SARSA, $\epsilon = 0.1$	-27.3	0.050	the top row, -17: 75; a detour, -19: 5; a rule that loops: 20
Q-learning, $\epsilon = 0.1$	-51.0	0.343	the edge, -13: 100
SARSA, $\epsilon = 0.01$	-16.7	0.006	the middle row, -15: 59; the top row, -17: 28; loops: 12
Q-learning, $\epsilon = 0.01$	-16.0	0.026	the edge, -13: 100

The last 100 episodes. Q-learning's walker falls one episode in three; SARSA's one in twenty. A table whose greedy arrows loop between two cells never arrives: its values away from the goal are rough.

	a walker who never slips Q-learning's question	the walker with the coin SARSA's question, $\varepsilon = 0.1$
the edge 13 steps, 11 cells exposed	−13 the shortest path	−51 one fall in three
the top row 17 steps, none exposed	−17 four steps longer	−27 one fall in twenty
	Q-learning's table picks this column's best: the edge	SARSA's table picks this column's best: the top row

The better table, the worse walker. Each target prices one column, Q-learning's the path walked without a slip, SARSA's the exploring walker, and each table picks the best cell of its column. The right column is what each learner's own walker earned in episodes 401–500.

- Q-learning's table says the edge, 100 of 100: the best path for a walker who never slips. Its walker slips: one fall in three, −51.
- SARSA's table says the top row, 75 of 100: priced for a walker who slips one step in ten, at 100 a slip. One fall in twenty, −27.
- Both tables answer their own question: Q^* prices the greedy walker; only the on-policy target prices the exploring one.
- $\varepsilon = 0.01$: the two questions nearly agree, −16.0 and −16.7 an episode. With $\varepsilon \rightarrow 0$ slowly, both tables go to Q^* (stated).

Maximization Bias and Double Q-Learning

Maximization bias in the Bellman backup (Sutton and Barto, 2018, Section 6.7).

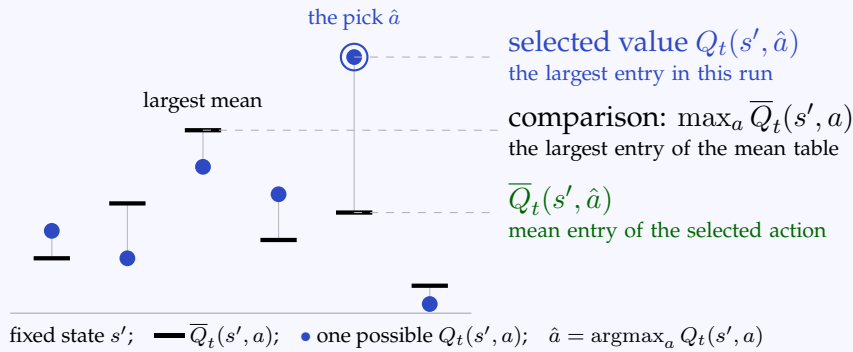
$$\mathbb{E}[R_{t+1} + \gamma \max_a Q_t(S_{t+1}, a) \mid Q_t, S_t = s, A_t = a] = (\mathcal{T}Q_t)(s, a)$$

one update: fix the table, sample the transition; the target is conditionally unbiased

Across training runs at the same time t , the table itself is random. Write $\bar{Q}_t = \mathbb{E}[Q_t]$ for its mean. Then, entry by entry,

$$\mathbb{E}[\mathcal{T}Q_t] \geq \mathcal{T}(\mathbb{E}[Q_t]) = \mathcal{T}\bar{Q}_t \quad \text{left: average the actual backups; right: back up the average table}$$

$\mathcal{T}\bar{Q}_t$ is a comparison, not the update's target: the gap isolates the effect of taking a max over a random table.



$$\mathbb{E}[\max_a Q_t(s', a)] \geq \max_a \mathbb{E}[Q_t(s', a)] = \max_a \bar{Q}_t(s', a)$$

at each s' ; the Bellman backup adds these inequalities with nonnegative weights

By hand. Two entries with means 1 and 0, each independently off by ± 1 : $\{0, 2\}$ and $\{-1, 1\}$. The largest mean is 1. Compare using the same table to select and evaluate with using an independent copy to evaluate the selected action:

same table: $\frac{0+1+2+2}{4} = \frac{5}{4}$ *four equally likely cases; the mean maximum exceeds 1 by $\frac{1}{4}$*

independent evaluation: $\frac{3}{4} \cdot 1 + \frac{1}{4} \cdot 0 = \frac{3}{4}$

the lower-mean action is selected only at $(0, 1)$, w.p. $\frac{1}{4}$: a downward bias

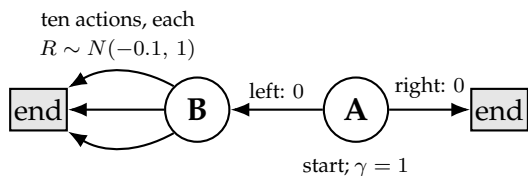
- The max favors upward errors. Independent evaluation removes that selection advantage, but the chosen action can have a lower mean.
- The gap is relative to $\mathcal{T}\bar{Q}_t$; $\bar{Q}_t$ need not equal Q^* . A positive gap alone does not rank the algorithms.

Double Q-learning (van Hasselt, 2010), (Sutton and Barto, 2018, Section 6.7). Two tables Q_1, Q_2 ; act ε -greedy w.r.t. $Q_1 + Q_2$; each week flip a fair coin:

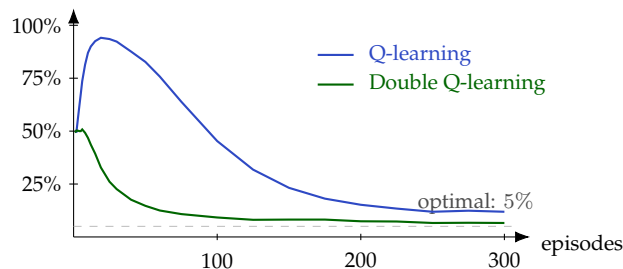
heads: $Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha(R_{t+1} + \gamma Q_2(S_{t+1}, a^*) - Q_1(S_t, A_t))$
 $a^* = \operatorname{argmax}_a Q_1(S_{t+1}, a)$: Q_1 picks the action, Q_2 prices it

tails: the same with 1 and 2 exchanged *each table is updated on half the weeks*

- One table selects, the other evaluates: splitting the roles can reduce maximization bias. Memory doubles; each transition updates one table.
- Bootstrapping couples the tables; independence is not guaranteed, and finite-time estimates may still be high or low.
- Both tables converge to Q^* with coverage and suitable step sizes for each pair in each table (stated) (van Hasselt, 2010).



Sutton and Barto (2018, Example 6.7). Truth: left is worth -0.1 , right 0; an ϵ -greedy learner that knows this goes left 5% of the time.



Share of left at A, 10,000 runs, $\epsilon = 0.1$, $\alpha = 0.1$,
 $Q_0 = 0$, ties at random.

Example: how often does Q-learning go left? Left's true value is -0.1 , right's 0. Over the first 100 episodes: about 5%, half, or most of the time?

Most: 94% at the peak (episode 19), 45% at 100, 12% at 300. The backup to A maximizes ten noisy entries at B; positive selected estimates can make left look better than right.

Double Q-learning: 51% at its peak, 9% at 100, 7% at 300; the floor is 5%.

Replay Is Certainty Equivalence

Replay of a fixed record (Lin, 1992), (Sutton and Barto, 2018, Section 6.3). A record D of n transitions (s_i, a_i, r_i, s'_i) , from any behavior. *Replay*: draw i uniformly from D , apply the Q-learning update at (s_i, a_i) , repeat.

$$\mathbb{E}[r_i + \gamma \max_a Q(s'_i, a) \mid (s_i, a_i) = (s, a)] = \hat{r}(s, a) + \gamma \sum_{s'} \hat{P}(s' \mid s, a) \max_a Q(s', a)$$

$\hat{P} = \text{counts}/n(s, a)$, $\hat{r} = \text{the average reward at } (s, a)$: the record's empirical model, never written down

$$=: (\hat{T}Q)(s, a)$$

the optimality backup of the empirical model

settled: $\hat{Q} = \hat{T}\hat{Q} \Rightarrow \hat{Q} = \hat{Q}^*$, the optimal table of $(\hat{P}, \hat{r})$ (stated)

certainty equivalence: build $(\hat{P}, \hat{r})$, then solve; replay reaches its answer without building

A pair the record never holds is never updated: it keeps its start value, and $\hat{Q}^*$ is computed with that entry frozen there.

Example: the six weeks replayed. The log's counts: (G, run) twice, to G and to W; (G, repair), (W, run), (B, run), (B, repair) once each; (W, repair) never. Replay the six lines at random until the table settles: which table?

The empirical model's optimum. With $x = \hat{Q}(\text{G, run})$ and $y = \hat{Q}(\text{W, run})$: $x = 8 + 0.9(\frac{1}{2}x + \frac{1}{2}y)$, $y = 4 + 0.9(-10 + 0.9x)$, so $x = 31.00$, $y = 20.11$; repair entries $-10 + 0.9x = 17.90$ at G and B, $\hat{Q}(\text{B, run}) = 0.9 \cdot 17.90 = 16.11$. Greedy: repair when broken, π^* , from the six weeks one pass read as never repair. The model wears the machine too fast ($\hat{P}(\text{W} \mid \text{G, run}) = \frac{1}{2}$, $\hat{P}(\text{B} \mid \text{W, run}) = 1$), so its numbers are low: the right rule, the wrong numbers.

On-Policy and Off-Policy, Side by Side

Learner	Target	Its mean at (s, a)	Settles at	What b must do
SARSA	$R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}),$ $A_{t+1} \sim b$	$(\mathcal{T}^b Q)(s, a)$	Q^b ; with the ε -greedy coin Q^{π_ε}	be the policy evaluated; to reach Q^* the coin must die out: on-policy
Expected SARSA	$R_{t+1} + \gamma \sum_a \pi(a \mid S_{t+1}) Q(S_{t+1}, a)$	$(\mathcal{T}^\pi Q)(s, a)$	Q^π	cover every pair; π may differ from b : either
Q-learning	$R_{t+1} + \gamma \max_a Q(S_{t+1}, a)$	$(\mathcal{T} Q)(s, a)$	Q^*	cover every pair: off-policy
Double Q-learning	$R_{t+1} + \gamma Q_2(S_{t+1}, a^*),$ $a^* = \operatorname{argmax}_a Q_1(S_{t+1}, a)$	$(\mathcal{T}^{\pi_1} Q_2)(s, a), \pi_1$ greedy w.r.t. Q_1 : the pick priced by the other table	Q^* (stated)	cover every pair: off-policy

Four targets, one template $Q \leftarrow Q + \alpha(\text{target} - Q)$ (*van Seijen et al., 2009; Singh et al., 2000*). The behavior decides which pairs are visited and what is earned meanwhile; the target decides what the table converges to.

Summary

Control from any stream, on one board

- Task: a table whose greedy policy is π^* from a stream whose actions a behavior b drew: the random manager's log (worth Q^b , a loss of 1.62 a week) or the learner's own coin. Truth: Q^* run (44.30, 32.41, 26.89), repair 29.87.
- Q-learning: $\delta_t = R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)$, $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \delta_t$. Mean target $(\mathcal{T}Q)(s, a)$ for every b ; settled with every pair visited forever means Q^* . Theorem stated; coverage is all b owes.
- By hand: targets 8, 8, 4, 0, -4.6, -4.6; SARSA's -10 at weeks 4 and 5. By computer, one log: Q-learning 2.33 from Q^* (0.15 with a shrinking step), SARSA 3.65 from Q^b ; a log without coverage leaves four entries at 0.
- Cliff Walking: the edge (-13) against the top row (-17); walking with the coin, -51 against -27 an episode. Q^* prices greedy acting; the exploring walker's earnings are the exploring rule's value.
- Maximization bias: $\mathbb{E}[\mathcal{T}Q_t] \geq \mathcal{T}(\mathbb{E}[Q_t])$ across runs; each sampled target remains conditionally unbiased for $\mathcal{T}Q_t$. Double Q separates selection and evaluation; left 94% at the peak against 51%.
- Replay of a fixed record: the expected update is the empirical model's optimality backup, so it settles at $\hat{Q}^*$: certainty equivalence without the model. Six weeks replayed: π^* , with low numbers.

References

- Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.
- Emma Brunskill. CS 234: Reinforcement learning. Stanford University course materials, 2026. URL <https://web.stanford.edu/class/cs234/>. Winter 2026 offering.
- Tommi Jaakkola, Michael I. Jordan, and Satinder P. Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, 6(6):1185–1201, 1994.
- Nan Jiang. CS 443: Reinforcement learning. University of Illinois Urbana-Champaign course materials, 2026. URL <https://nanjiang.cs.illinois.edu/cs443/>. Spring 2026 offering.
- Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

- Gavin A. Rummery and Mahesan Niranjana. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, 1994.
- Satinder Singh, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, 38(3):287–308, 2000.
- Wen Sun. CS 4/5789: Introduction to reinforcement learning. Cornell University course materials, 2025. URL https://wensun.github.io/CS4789_spring_2025.html. Spring 2025 offering.
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018. URL <http://incompleteideas.net/book/the-book-2nd.html>.
- Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, 2010. URL <https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf>.
- John N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16(3):185–202, 1994.
- Hado van Hasselt. Double Q-learning. In *Advances in Neural Information Processing Systems* 23, pages 2613–2621, 2010.
- Harm van Seijen, Hado van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of Expected Sarsa. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 177–184, 2009.
- Christopher J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King’s College, University of Cambridge, 1989.
- Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3–4):279–292, 1992.