

A COMPARISON OF EXPECTED SARSA AND CBDQ

| Feature | Expected SARSA | Cognitive Belief-Driven Q-learning |
|--------------------------|---|--|
| Policy Type | On-policy | Off-policy |
| Action Selection | Single policy $\pi(a s)$ for both experience generation and updates | Exploration policy for experience, $b_t(a s_{t+1})$ distribution for updates |
| Convergence Target | True action-value function of the current policy | Optimal Q-value function (under specific conditions) |
| Exploration-Exploitation | Controlled by single policy π | Exploration policy and b_t distribution can be adjusted independently |
| Sample Utilization | Only uses samples from current policy | Can utilize samples from any policy |
| Main Advantage | Directly evaluates current policy, potentially faster convergence | More flexible, potentially more stable, can find optimal policy |
| Suitable Scenarios | Online learning, need for quick policy evaluation | Offline learning, need to find optimal policy |

Table 1: Comparison between Expected SARSA and Smoothed Q-learning

B MBP EXPERIMENT

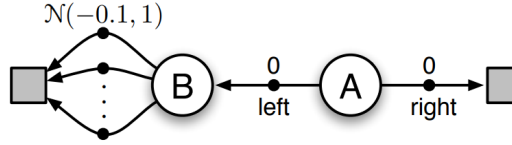


Figure 1: **Experimental Setup of the Maximization Bias Problem (MBP)**: The experiment starts in state A. The agent has two possible actions: *Right*, leading to terminal state C with zero reward, and *Left*, which leads to state B, also with zero reward. In state B, the agent has 8 actions, each leading to terminal state D with a reward sampled from a Gaussian distribution with a mean of -0.1 and a variance of 1. This setup illustrates maximization bias in traditional Q-learning algorithms, where overestimation can occur due to variance in the rewards (Sutton & Barto, 2018).

Purpose of the Experiment This setup underscores the issue of **maximization bias** in traditional Q-learning, where the algorithm selects actions based on the highest Q-value. In state B, the variability in rewards amplifies this bias, as Q-learning tends to overestimate the expected reward by favoring actions with initially higher but unreliable Q-values. Over time, this can lead the agent to consistently choose suboptimal actions, even when more stable options offer better long-term results.

C SMOOTHING STRATEGY

| Strategy | Formula |
|--------------------|---|
| Softmax | $b_t = \frac{e^{Q(s,a)}}{\sum_b e^{Q(s,b)}}$ |
| Clipped Max | $b_t = \begin{cases} 1 - \tau, & \text{if } a = a^* \\ \frac{\tau}{A-1}, & \text{if } a \neq a^* \end{cases}$ |
| Clipped Softmax | $b_t = \begin{cases} \frac{e^{\beta Q(s,a)}}{\sum_{b \in I} e^{\beta Q(s,b)}}, & \text{if } a \in I \\ 0, & \text{if } a \notin I \end{cases}$ |
| Bayesian Inference | $Q_{\text{adjusted}}(s, a) = Q(s, a) + \mu_{\text{prior}}$ $b_t = \frac{e^{Q_{\text{adjusted}}(s', a)}}{\sum_b e^{Q_{\text{adjusted}}(s', b)}}$ $\sigma_{\text{posterior}}^2 = \left(\frac{1}{\sigma_{\text{prior}}^2} + \frac{n}{\sigma_{\text{observation}}^2} \right)^{-1}$ $\mu_{\text{posterior}} = \sigma_{\text{posterior}}^2 \left(\frac{\mu_{\text{prior}}}{\sigma_{\text{prior}}^2} + \sum_{i=1}^n \frac{r_i}{\sigma_{\text{observation}}^2} \right)$ |

Table 2: Smoothing strategies with respective formulas

D CONVERGENCE PROOF

We outline a proof that builds upon the following result (Singh et al., 2000; Barber, 2023) for a formal statement) and follows the framework provided in (Melo, 2001):

Theorem 1 The random process $\{\Delta_t\}$ taking value in \mathbb{R} and defined as

$$\Delta_{t+1}(x) = (1 - \alpha_t(x))\Delta_t(x) + \alpha_t(x)F_t(x) \quad (1)$$

converges to 0 with probability 1 under the following assumptions:

- $0 \leq \alpha_t \leq 1$, $\sum_t \alpha_t(x) = \infty$, $\sum_t \alpha_t^2(x) < \infty$;
- $\mathbb{E}[\|F_t(x)\|_W] \leq \kappa \|\Delta_t\|_W + c_t$, $\kappa \in [0, 1)$ and $c_t \rightarrow 0$ with probability 1;
- $\text{var}(F_t(x)) \leq C(1 + \|\Delta_t\|_W)^2$, $C > 0$

where $\|\Delta_t\|_W$ denotes a weighted max norm.

We are interested in the convergence of Q_t towards the optimal value Q_* and therefore define

$$\Delta_t = Q_t(s_t, a_t) - Q_*(s_t, a_t) \quad (2)$$

It is convenient to write the smoothed update as

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t) (r_t + \gamma \langle Q(s_{t+1}, a) \rangle_a - Q_t(s_t, a_t)) \quad (3)$$

where $\langle f(x) \rangle_x$ means the expectation of the function $f(x)$ with respect to the distribution of x . Using the smoothed update, we can write

$$\Delta_{t+1}(s_t, a_t) = Q_{t+1}(s_t, a_t) - Q_*(s_t, a_t) \quad (4)$$

$$= (1 - \alpha_t)\Delta_t + \alpha_t (r_t + \gamma \langle Q(s_{t+1}, a) \rangle_a - Q_*(s_t, a_t)) \quad (5)$$

In terms of Theorem 1, we therefore define

$$F_t = r_t + \gamma \sum_a b_t(a|s_{t+1}) Q_t(s_{t+1}, a) - Q_*(s_t, a_t) \quad (6)$$

Proof D.1 For convergence, we need to bound the norm of the expected value of F_t . We can write

$$\frac{1}{\gamma} \mathbb{E}[F_t] = \mathbb{E}_{p_T}[G_t] \quad (7)$$

where

$$G_t = \sum_a b_t(a|s_{t+1})Q_t(s_{t+1}, a) - \max_a Q_*(s_{t+1}, a) \quad (8)$$

we can form the bound

$$\frac{1}{\gamma} \mathbb{E}[F_t]_\infty = \mathbb{E}[\|G_t\|_\infty] \leq \|G_t\|_\infty \quad (9)$$

which means that if we can bound $\|G_t\|_\infty$ appropriately, the mean criterion will be satisfied.

Assuming that b_t places $(1 - \delta_t)$ mass in the maximal state of Q , we can write

$$\|G_t\|_\infty \leq \left\| \max_a Q_t(s_{t+1}, a) - \max_a Q_*(s_{t+1}, a) \right\|_\infty + \delta_t \left\| \max_a Q_t(s_{t+1}, a) - \sum_{c \neq a} b_t(c|s_{t+1})Q_t(s_{t+1}, c) \right\|_\infty \quad (10)$$

$$\leq \|\Delta_t\|_\infty + \delta_t \left\| \max_a Q_t(s_{t+1}, a) - \sum_{c \neq a} b_t(c|s_{t+1})Q_t(s_{t+1}, c) \right\|_\infty \quad (11)$$

$$\leq \|\Delta_t\|_\infty + \delta_t \left(\left\| \max_a Q_t(s_{t+1}, a) \right\|_\infty + \|Q_t(s_{t+1}, c_-)\|_\infty \right) \quad (12)$$

where $c_- = \arg \min_{c \neq a} Q_t(s_{t+1}, c)$ and the penultimate line follows from the fact that only a maximum of δ_t mass can be placed in the minimal state c_- (since $(1 - \delta_t)$ mass is placed in state a_*). Putting this together we have

$$\mathbb{E}[F_t]_\infty \leq \gamma \|\Delta_t\|_\infty + \gamma \delta_t \left(\left\| \max_a Q_t(s_{t+1}, a) \right\|_\infty + \|Q_t(s_{t+1}, c_-)\|_\infty \right) \quad (13)$$

Since the Q_t are bounded and $\mathbb{E}[F_t]$ converges to zero with probability 1, provided δ_t converges to 0 with probability 1. The mean criterion is therefore satisfied.

For the variance criterion, since the rewards are bounded, Q_t and Δ_t are also bounded. This means that the variance is bounded. We can write:

$$\Delta F_t = \Delta r + \gamma \langle (Q_t(s_{t+1}, a) - \langle Q_t(s_{t+1}, a) \rangle) \rangle_{s_{t+1}, a} \quad (14)$$

$$= \Delta r + \gamma \langle (Q_t(s_{t+1}, a))_a - (Q_*(s_{t+1}, a))_a + (Q_*(s_{t+1}, a))_a - \langle Q_t(s_{t+1}, a) \rangle_a \rangle_{s_{t+1}} \quad (15)$$

$$= \Delta r + \gamma \left\langle Q_t(s_{t+1}, a) - Q_*(s_{t+1}, a) - \gamma \langle Q_t(s_{t+1}, a) \rangle_{s_{t+1}} \right\rangle_a \quad (16)$$

We can bound the variance using

$$\text{var}(F_t) = \|\langle \Delta F_t \rangle\|_\infty^2 \leq \|\Delta F_t\|_\infty^2 \quad (17)$$

and use the triangle inequality,

$$\|\Delta F_t\|_\infty \leq \|\Delta r\|_\infty + \gamma \langle \|Q_t(s_{t+1}, a) - Q_*(s_{t+1}, a)\|_a \rangle \quad (18)$$

and using $\|x\|_\infty \leq \|x\|_\infty$

$$\|\Delta F_t\|_\infty \leq \|\Delta r\|_\infty + \gamma \|\Delta_t\|_\infty + \gamma \langle \|Q_t(s_{t+1}, a) - Q_*(s_{t+1}, a)\| \rangle_\infty \quad (19)$$

We now write

$$\langle Q_t(s_{t+1}, a) - Q_*(s_{t+1}, a) \rangle_\infty \quad (20)$$

$$= \langle (Q_t(s_{t+1}, a))_a - (Q_*(s_{t+1}, a))_a + (Q_*(s_{t+1}, a))_a - \langle Q_t(s_{t+1}, a) \rangle_a \rangle_\infty \quad (21)$$

$$\leq \|\Delta_t\|_\infty + \langle \| (Q_*(s_{t+1}, a))_a - Q_*(s_{t+1}, a) \| \rangle_\infty \quad (22)$$

$$\leq \|\Delta_t\|_\infty + B \quad (23)$$

for some constant B_1 since the optimal Q_* is bounded (for $\gamma < 1$ and bounded rewards). Hence, since the rewards are bounded, there exists B such that

$$\|\Delta F_t\|_\infty \leq 2\gamma B + 2\gamma \|\Delta_t\|_\infty = 2\gamma B(1 + \|\Delta_t\|_W) \quad (24)$$

This shows that the variance condition is satisfied.

E EXPERIMENT SETTING

E.1 CLASSIC CONTROL AND BOX 2D ENVIRONMENT

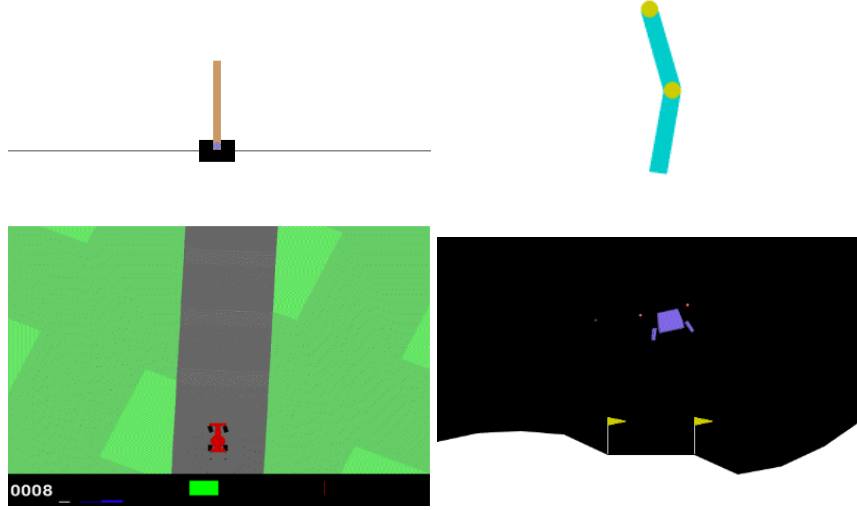


Figure 2: Cartpole(Top Left), Acrobot(Top Right), Car Racing(Bottom Left), and Lunar Lander(Bottom Right).

1. Cartpole: A pole is attached by an unactuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.
2. Acrobot: A two-link pendulum system with only the second joint actuated. The task is to swing the lower link to a sufficient height in order to raise the tip of the pendulum above a target height. The environment challenges the agent’s ability to apply precise control for coordinating multiple linked joints.
3. Car Racing: The easiest control task to learn from pixels - a top-down racing environment. The generated track is random in every episode.
4. Lunar Lander: It is a classic rocket trajectory optimization problem. According to Pontryagin’s maximum principle, it is optimal to fire the engine at full throttle or turn off. This is why this environment has discrete actions: engine on or off.

E.2 METADRIIVE BLOCK TYPE DESCRIPTION

Table 3: Block Types Used in Experiments

| ID | Block Type |
|----|----------------|
| S | Straight |
| C | Circular |
| r | InRamp |
| R | OutRamp |
| O | Roundabout |
| X | Intersection |
| y | Merge |
| Y | Split |
| T | T-Intersection |

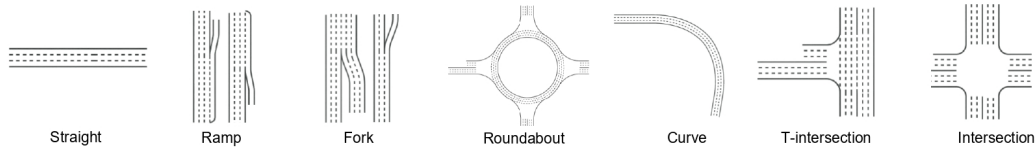


Figure 3: Various block types used in the MetaDrive environment. These blocks represent common road structures such as straight roads, ramps, forks, roundabouts, curves, T-intersections, and intersections, used for evaluating the vehicle’s path planning and decision-making capabilities.

E.3 MAP DESIGN AND TESTING OBJECTIVES

E.3.1 MAP 1: SROYCTRY

This map consists of straight roads, roundabouts, intersections, T-intersections, splits, and ramps. The environment presents a highly complex combination of multiple intersections, dynamic traffic flow, and varying road structures.

Testing Objective: The focus of this environment is to evaluate the algorithm’s smooth decision-making and multi-intersection handling, mimicking human driving behavior. The challenges include adjusting vehicle paths in real-time and ensuring smooth lane transitions in the presence of complex road structures such as roundabouts and ramps.

E.3.2 MAP 2: CORXSRT

This map combines circular roads, roundabouts, straight roads, intersections, ramps, and T-intersections. The environment is designed to assess the vehicle’s decision-making capabilities when dealing with continuous changes in road grades and multiple intersection types.

Testing Objective: This environment tests the algorithm’s ability to dynamically adjust to **grade changes** and **multi-intersection interactions**, replicating human-like behavior. The goal is to observe how well the algorithm adjusts vehicle speed and direction, ensuring stability in scenarios involving ramps and complex road networks.

E.3.3 MAP 3: RXTSC

This map consists of ramps, intersections, T-intersections, straight roads, and circular roads. The environment simulates multiple road interactions, testing the vehicle’s path selection and stability, particularly at intersections and ramps.

Testing Objective: This environment evaluates the algorithm’s performance in handling intersections and T-junctions with real-time path selection. The challenge is to ensure human-like adaptability when encountering multiple directional options, maintaining decision stability in dynamic traffic situations.

E.3.4 MAP 4: YORSX

This map includes splits, roundabouts, straight roads, circular roads, and intersections. The environment is tailored to test the vehicle’s ability to make path decisions in high-speed settings, particularly when merging traffic and navigating through complex junctions.

Testing Objective: The map focuses on testing the vehicle’s ability to handle **high-speed lane merging** and **dynamic path planning**. The algorithm must mimic human drivers by making real-time adjustments in a high-speed environment, choosing optimal paths while maintaining speed control and safety through complex intersections and roundabouts.

E.3.5 MAP 5: XTOC

This map features circular roads, T-intersections, and straight roads, creating a unique combination of continuous curves and abrupt directional changes. The environment presents the challenge of maintaining speed while negotiating tight turns and quick transitions at T-intersections.

270 **Testing Objective:** The focus is on testing the vehicle’s ability to handle **sharp directional changes**
271 and maintain control during high-speed maneuvers. The algorithm needs to balance speed with
272 precision, ensuring safe navigation through tight turns and abrupt intersections.
273

274 E.3.6 MAP 6: SSSC

275
276 This map consists of three consecutive straight roads followed by a circular roundabout. It is de-
277 signed to test the basic driving capabilities of the vehicle, such as lane keeping, speed control, and
278 smooth roundabout navigation.

279 **Testing Objective:** The main challenge is to evaluate the vehicle’s ability to maintain **lane stability**
280 and make appropriate **speed adjustments** while navigating long straight roads and transitioning into
281 a circular roundabout. The algorithm must ensure smooth control and decision-making, simulating
282 human-like behavior in handling both high-speed straight roads and slower, more controlled turns
283 in the roundabout.
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

E.4 ENVIRONMENT PARAMETER & AGENT PARAMETER

Table 4: Q-family vs PPO Algorithm and Environment Parameters

| Parameter | Q-Family | PPO |
|---------------------------------|-----------------|-------------------------------------|
| Discrete Action Space | | True |
| Policy | Basic_Q_network | Categorical_AC |
| Representation | | Basic_MLP |
| Runner | | DRL |
| Representation Hidden Size | [256, 256] | [512,] |
| Q/Actor Hidden Size | [256, 256] | [256, 256] |
| Critic Hidden Size | N/A | [256, 256] |
| Activation Function | relu | leaky_relu |
| Activation for Actions | N/A | tanh |
| Seed | | 123 / 321 / 666 |
| Number of Parallels | | 10 |
| Buffer Size | 500,000 | Horizon_Size * Parallels (128 * 10) |
| Batch Size | 64 | N/A |
| Horizon Size | N/A | 128 |
| Number of Epochs | N/A | 4 |
| Number of Minibatches | N/A | 4 |
| Learning Rate | | 0.00025 |
| Start Greedy | 1.0 | N/A |
| End Greedy | 0.01 | N/A |
| Decay Step for Greedy | 50,000 | N/A |
| Sync Frequency | 50 | N/A |
| Training Frequency | 1 | N/A |
| Start Training Step | 1,000 | N/A |
| Running Steps | | 2,000,000 |
| Use Gradient Clipping | N/A | True |
| Value Function Coefficient | N/A | 0.25 |
| Entropy Coefficient | N/A | 0.0 |
| Target KL Divergence | N/A | 0.001 |
| Clip Range | N/A | 0.2 |
| Clip Gradient Norm | N/A | 0.5 |
| Gamma | | 0.99 |
| Use GAE | N/A | True |
| GAE Lambda | N/A | 0.95 |
| Use Advantage Normalization | N/A | True |
| Use Observation Normalization | False | True |
| Use Reward Normalization | False | True |
| Observation Normalization Range | | 5 |
| Reward Normalization Range | | 5 |
| Test Steps | | 10,000 |
| Evaluation Interval | 50,000 | 5,000 |
| Test Episodes | | 5 |

F EXPERIMENTAL SUPPLEMENTAL RESULTS

Table 5: We showcase the rewards of mean \pm std for each algorithm in Box2D Environments

| Environment/Method | CBDDQN | PPO | Duel DQN | DDQN | DQN |
|--------------------|--------------------------------------|--------------------|---------------------|---------------------|--------------------|
| Cartpole | 469.98 \pm 20.26 | 427.29 \pm 16.62 | 92.24 \pm 10.56 | 222.14 \pm 19.71 | 294.79 \pm 16.41 |
| Acrobot | -80.57 \pm 12.63 | -500.00 \pm 0 | -104.54 \pm 40.55 | -100.78 \pm 21.07 | -87.20 \pm 14.07 |
| CarRacing | 819.08 \pm 28.72 | 272.08 \pm 27.02 | -27.29 \pm 6.78 | 788.13 \pm 37.61 | 724.76 \pm 37.17 |
| LunarLander | 158.07 \pm 46.14 | 89.34 \pm 70.44 | -76.54 \pm 84.85 | 73.04 \pm 56.16 | 91.86 \pm 70.44 |

Table 6: We present the rewards of mean \pm std for each algorithm in Metadrive Environments

| Map/Method | CBDDQN | PPO | Duel DQN | DDQN | DQN |
|------------------|--------------------------------------|--------------------|-------------------|--------------------|--------------------|
| SrOYCTryS | 130.27 \pm 52.43 | 75.38 \pm 17.80 | 39.20 \pm 3.87 | 100.72 \pm 39.01 | 105.02 \pm 41.69 |
| COrXSrT | 117.90 \pm 22.62 | 89.27 \pm 19.99 | 53.02 \pm 1.95 | 29.15 \pm 7.03 | 117.18 \pm 15.34 |
| rXTSC | 189.22 \pm 59.94 | 156.74 \pm 47.77 | 39.62 \pm 3.00 | 185.55 \pm 56.03 | 82.05 \pm 30.27 |
| YOrSX | 232.55 \pm 83.76 | 165.46 \pm 52.43 | 77.65 \pm 14.21 | 81.03 \pm 24.40 | 221.44 \pm 40.26 |

Table 7: We present the rewards of mean \pm std for different traffic density in Metadrive XTOC map

| Traffic Density/Method | CBDDQN | PPO | Duel DQN | DDQN | DQN |
|------------------------|--------------------------------------|--------------------|--------------------|--------------------|--------------------|
| 0.1 | 443.14 \pm 59.63 | 73.90 \pm 2.00 | 65.85 \pm 8.43 | 151.42 \pm 47.66 | 272.57 \pm 91.25 |
| 0.3 | 303.15 \pm 38.20 | 293.72 \pm 56.28 | 67.58 \pm 7.49 | 156.52 \pm 39.27 | 170.73 \pm 42.62 |
| 0.5 | 303.07 \pm 40.61 | 256.18 \pm 26.69 | 139.46 \pm 39.78 | 164.34 \pm 58.03 | 176.83 \pm 56.12 |
| 0.8 | 161.91 \pm 34.52 | 67.91 \pm 3.42 | 60.71 \pm 10.58 | 150.06 \pm 36.45 | 147.92 \pm 35.21 |

Table 8: We present the rewards of mean \pm std for different accident probabilities in Metadrive SSSC map

| Traffic Density/Method | CBDDQN | PPO | Duel DQN | DDQN | DQN |
|------------------------|-------------------------------------|-------------------|------------------|-------------------|-------------------|
| 0.1 | 64.62 \pm 10.41 | -1.72 \pm 0.55 | 40.32 \pm 4.60 | 45.63 \pm 4.56 | 46.73 \pm 7.38 |
| 0.3 | 69.23 \pm 6.46 | 45.31 \pm 12.04 | 40.99 \pm 1.83 | 43.42 \pm 10.48 | 55.14 \pm 9.41 |
| 0.5 | 69.23 \pm 6.46 | 45.60 \pm 10.24 | 41.12 \pm 1.71 | 43.42 \pm 10.48 | 55.14 \pm 9.41 |
| 0.8 | 73.25 \pm 6.78 | -5.29 \pm 0.16 | 43.78 \pm 4.27 | 9.10 \pm 3.22 | 55.17 \pm 11.03 |

G RUNNING SETTING

For the Cartpole and Lunar Lander environments, the training process utilizes 1 RTX 3060 and typically runs less than 30 minutes. For the Carracing environment, we require 1 RTX 3060 and 2 hours of running. For the Metadrive environments, the training process utilizes 1 RTX 3060 and typically runs around 3-6 hours according to different complexity.

REFERENCES

- David Barber. Smoothed q-learning. *arXiv preprint arXiv:2303.08631*, 2023.
- Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.
- Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38:287–308, 2000.
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.