



北京航空航天大学
COLLEGE OF SOFTWARE 软件学院
BEIHANG UNIVERSITY

人工智能

第13讲：机器学习-强化学习

基于价值的强化学习、Actor-Critic方法

张晶

2025年春季

- 参考资料：吴飞，《人工智能导论：模型与算法》，高等教育出版社
- 在线课程：<https://www.icourse163.org/course/ZJU-1003377027?from=searchPage>
- 本部分参考：李宏毅，《机器学习》课程，台湾大学



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、强化学习问题定义

二、基于策略的强化学习

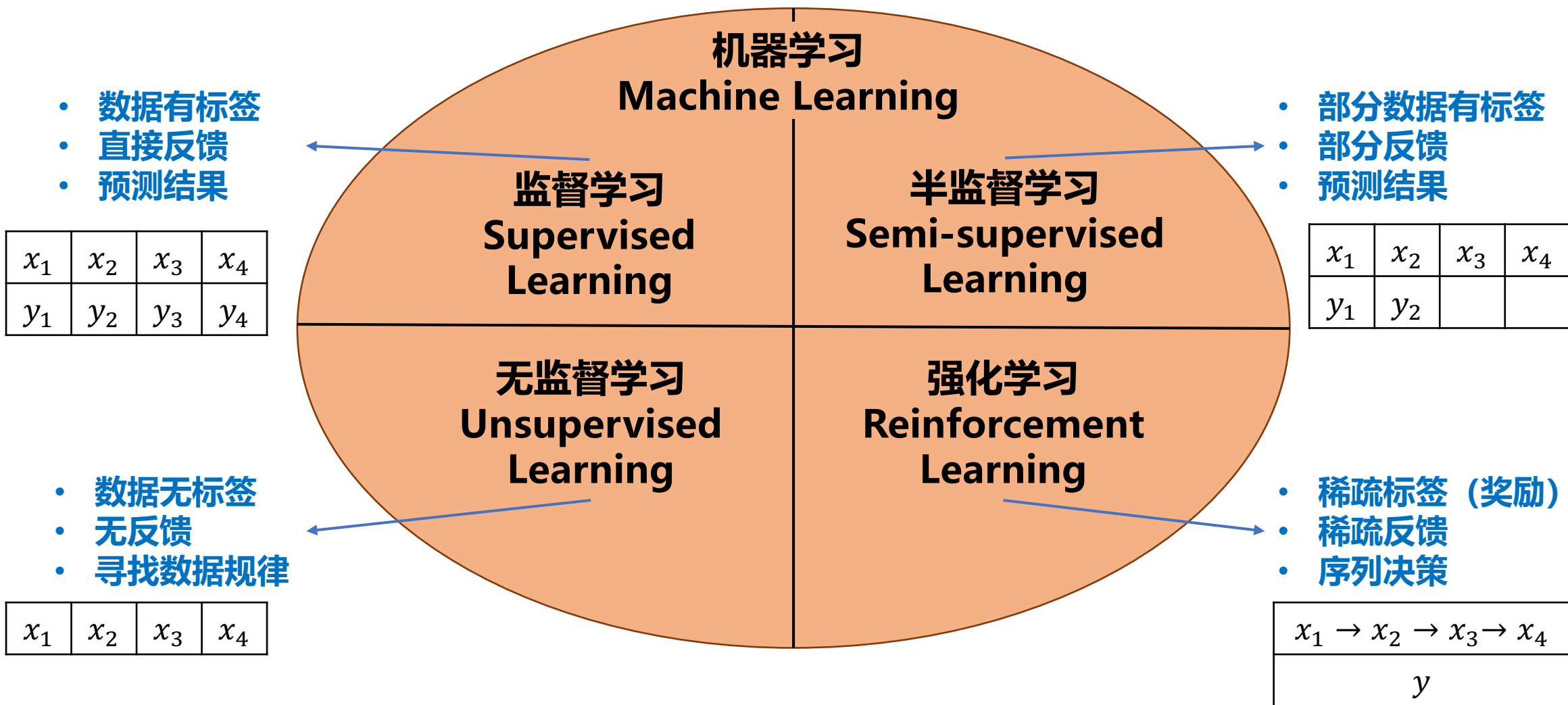
三、基于价值的强化学习

四、 Actor-Critic方法

五、其他强化学习方法



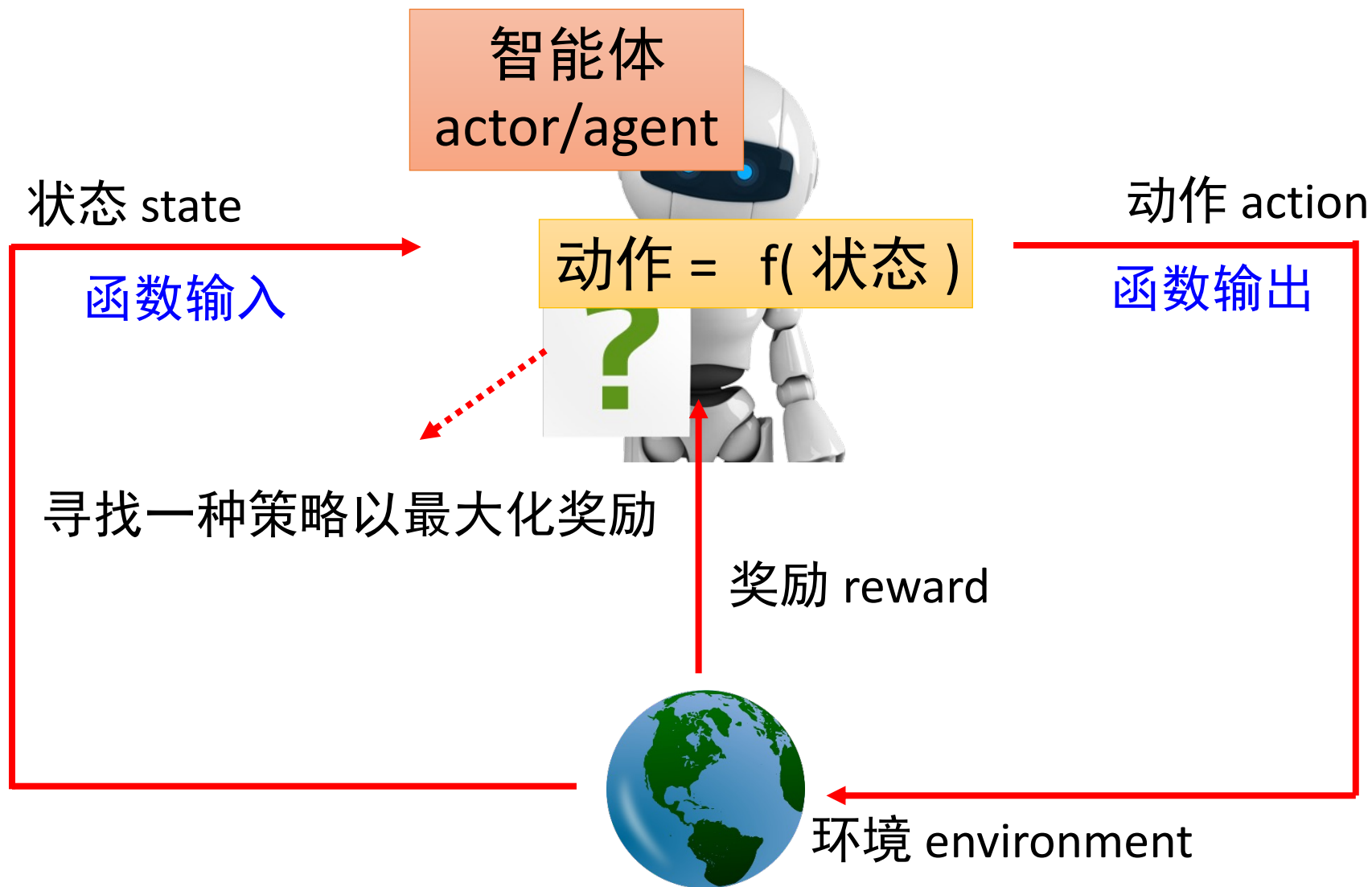
机器学习的分类（按数据标注情况分类）





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

回顾：机器学习≈函数拟合





回顾：强化学习系统要素

- 状态 (State) → “你现在在哪？”
 - 环境的完整内部描述，包含所有决定未来动态的信息： S_t
- 策略 (Policy) → “你应该做什么？”
 - 是学习智能体在特定时间的行为方式
 - 是从状态到动作的映射
 - 确定性策略 (Deterministic Policy) :
$$a = \pi(s)$$
 - 随机策略 (Stochastic Policy) :
$$\pi(a|s) = P(A_t = a|S_t = s)$$



回顾：强化学习系统要素

- **奖励 (Reward)** → “你做这个动作，能得多少分？”
 - 一个定义强化学习目标的标量： $r(s, a)$
 - 能立即感知到什么是“好”的
- **回报 (Return)** → “某个时刻之后总共能得多少分？”
 - 奖励的总和，描述了一种长期的收益
 - 无折扣回报： $G_t := r_{t+1} + r_{t+2} + r_{t+3} + \dots$
 - 折后回报： $G_t := r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots$



回顾：强化学习系统要素

• 价值函数（Value Function）

- 状态价值是一个标量，用于定义对于长期来说什么是“好”的
- 价值函数是对于未来累积奖励的预测，用于评估在给定策略 π 时，某个状态 s 的好坏

$$\begin{aligned} V_{\pi}(s) &= \mathbb{E}_{\pi}[G_t | S_t = s] = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | S_t = s] \\ &= \mathbb{E}_{\pi}[r_{t+1} + \gamma V_{\pi}(s') | S_t = s] \end{aligned}$$



$V_{\pi}(s_1)$



$V_{\pi}(s_2)$



回顾：强化学习系统要素

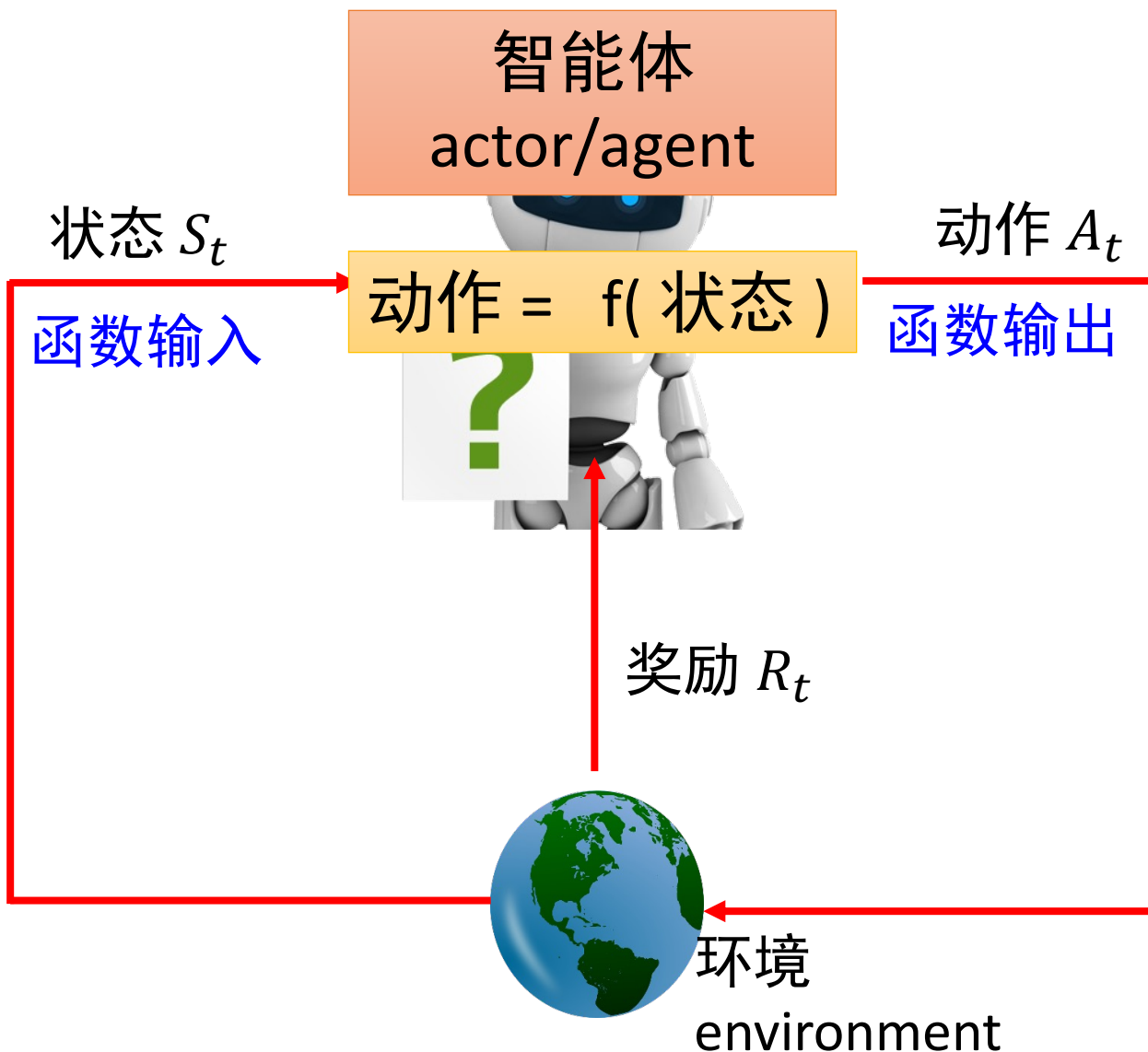
- 环境的模型（Model）用于模拟环境的行为

- 预测下一个状态

$$\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

- 预测下一个（立即）奖励

$$\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$





回顾：MDP五元组(S, A, P_{sa}, γ, R)

- S 是状态的集合 \rightarrow “你现在在哪？”
 - 比如，视频游戏中的当前屏幕显示；CartPole游戏中的 (x, v, θ, ω)
- A 是动作的集合 \rightarrow “你能做什么？”
 - 比如，视频游戏中手柄操纵杆方向和按钮；CartPole游戏中 $A = \{\text{左}, \text{右}\}$
- P_{sa} 是状态转移概率 \rightarrow “你做了某个动作后，会发生什么？”
 - 对每个状态和动作， $P_{sa} = Pr(S_{t+1}|S_t, a_t)$ 是下一个状态在 S 中的概率分布
- $\gamma \in [0, 1]$ 是对未来奖励的折扣因子
- R 是奖励函数 \rightarrow “你做这个动作，能得多少分？”
 - $R(S_t, a_t, S_{t+1}): S \times A \mapsto \mathbb{R}$



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、强化学习问题定义

二、基于策略的强化学习

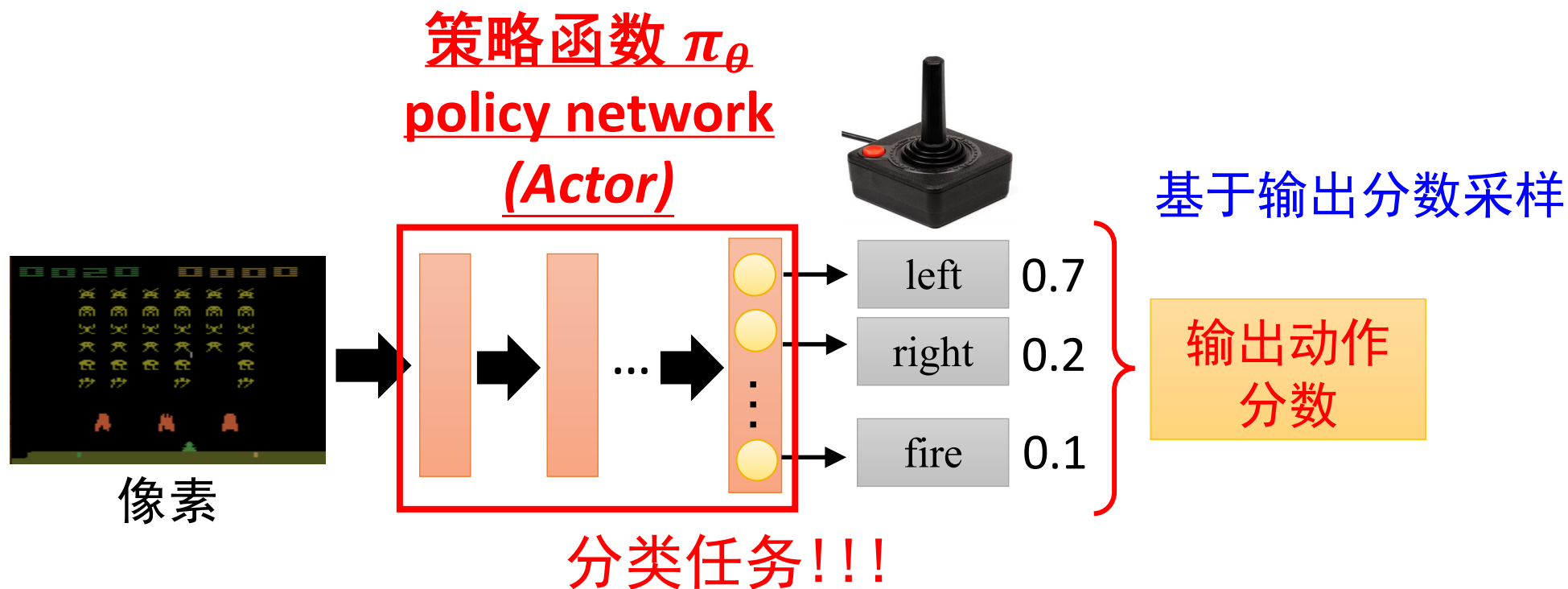
三、基于价值的强化学习

四、 Actor-Critic方法

五、其他强化学习方法



步骤 1: 定义带有未知参数的函数



- 神经网络的输入：向量或矩阵形式的状态数据
- 神经网络的输出：每个神经元对应的动作（基于输出分数采样）



步骤 2: 定义损失函数

初始状态 s_1



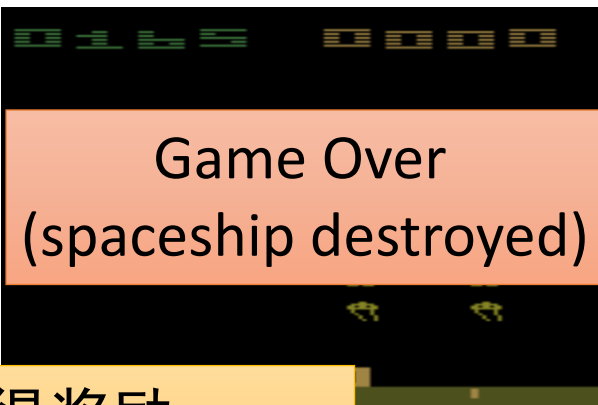
状态 s_2



状态 s_3



After many turns



获得奖励 r_T

动作 a_T

This is an episode (片段).

累积奖励 (回报):

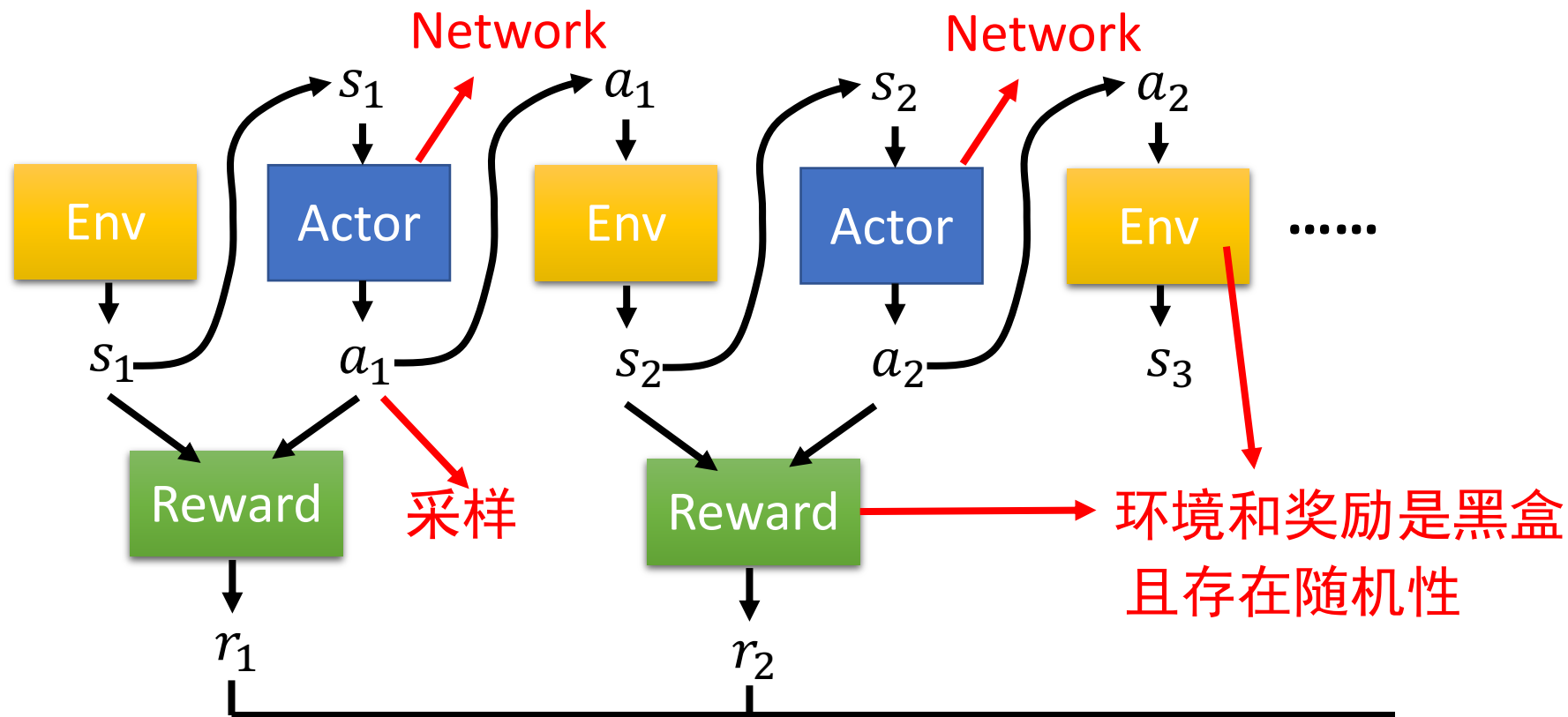
$$G = \sum_{t=1}^T r_t$$

最大化回报




步骤 3: 优化

Trajectory (轨迹) $\tau = \{s_1, a_1, s_2, a_2, \dots\}$

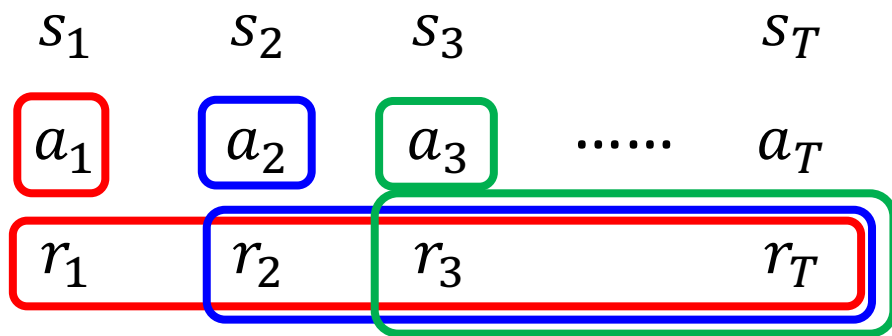


如何进行模型的优化是RL的主要挑战

c.f. GAN

$$G(\tau) = \sum_{t=1}^T r_t$$


版本 3



奖励的好坏是“相对的”

如果所有的 $r_n \geq 10$, 则 $r_n = 10$ 是负奖励...

减去一个基线值 b

???

使得 G'_t 具有正负奖励值

优势函数
advantage function

Training Data

$$\{s_1, a_1\} \quad A_1 = G'_1 - b$$

$$\{s_2, a_2\} \quad A_2 = G'_2 - b$$

$$\{s_3, a_3\} \quad A_3 = G'_3 - b$$

\vdots

\vdots

$$\{s_T, a_T\} \quad A_T = G'_T - b$$

$$G'_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$$



策略梯度 policy gradient

- 初始化策略网络参数 π^0
- 迭代训练 $i = 1$ to N
 - 用策略网络 π^{i-1} 进行交互
 - 获得训练数据 $\{s_1, a_1\}, \{s_2, a_2\}, \dots, \{s_T, a_T\}$
 - 计算 A_1, A_2, \dots, A_T
 - 计算损失 L
 - $\pi^i \leftarrow \pi^{i-1} - \eta \nabla L$

数据采集是在训练迭代过程的“for 循环”内部完成.



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、强化学习问题定义

二、基于策略的强化学习

三、基于价值的强化学习

四、 Actor-Critic方法

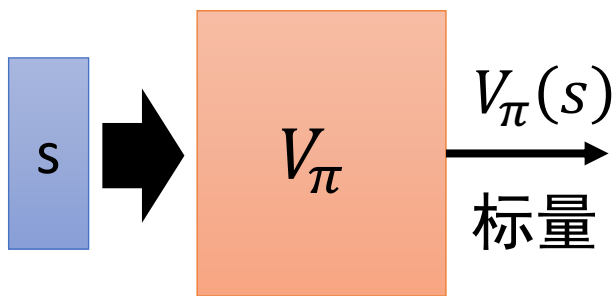
五、其他强化学习方法



Critic (评判器: 价值函数)

$$G'_1 = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots$$

- Critic: 对于策略 π , 评判状态 s 的好坏
- 价值函数 $V_\pi(s)$:
 - 使用 π 时, 观察到状态 s 后未来获得的折扣累积奖励(discounted *cumulated* reward)的期望。



$V_\pi(s)$ 大



$V_\pi(s)$ 小

Critic的输出值与当前策略 π 有关



Critic（评判器：价值函数）

- 价值函数（Value Function）： $V: S \mapsto \mathbb{R}$ ，即在第 t 步状态为 s 时，按照策略 π 行动后在未来所获得回报值的期望

$$\begin{aligned} V_{\pi}(s) &= \mathbb{E}_{\pi}[G_t | S_t = s] \\ &= \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s] \end{aligned}$$

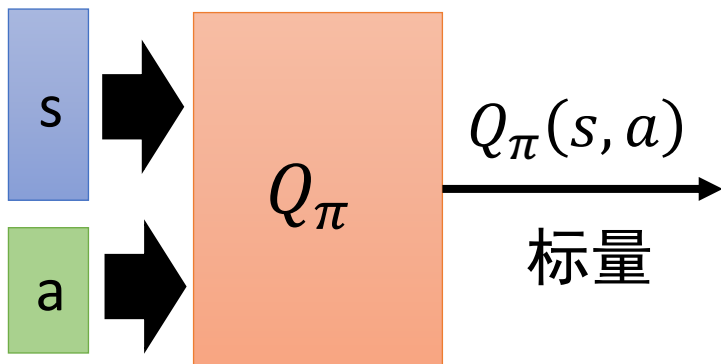


另一种Critic（评判器：动作-价值函数）

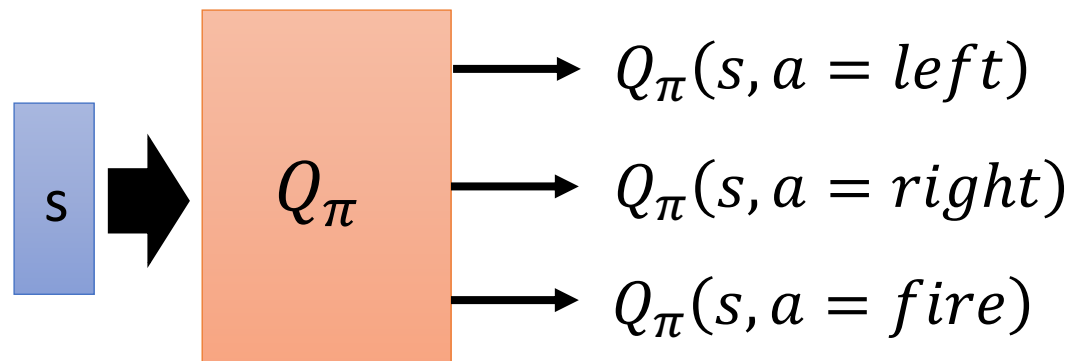
$$G'_1 = r_1 + \gamma r_2 + \gamma^2 r_3 + \dots$$

- Critic: 对于策略 π , 状态 s (且执行动作 a)的好坏
- 动作-价值函数 $Q_{\pi}(s, a)$:
 - 对于策略 π , 在状态 s 下执行动作 a 后未来获得的折扣累积奖励(discounted cumulated reward)的期望。

方法1:



方法2:



仅适用于离散动作



Critic（评判器：动作-价值函数）

- 动作-价值函数（Action-Value Function）： $Q: S \times A \mapsto \mathbb{R}$ ，表示在第 t 步状态为 s 时，按照策略 π 采取动作 a 后，在未来所获得回报值的期望

$$\begin{aligned} Q_{\pi}(s, a) &= \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a] \\ &= \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a] \end{aligned}$$



贝尔曼方程 (Bellman Equation): 递推关系

- 贝尔曼方程 (Bellman Equation) 也被称作动态规划方程 (Dynamic Programming Equation)
- 由理查德·贝尔曼 (Richard Bellman) 提出。
- 是强化学习的重要手段
- 核心思想：递推关系。比如

(暂且忽略期望)

$$V_{\pi}(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \dots$$

$$V_{\pi}(s_{t+1}) = r_{t+1} + \gamma r_{t+2} + \dots$$

$$V_{\pi}(s_t) = \gamma V_{\pi}(s_{t+1}) + r_t$$



贝尔曼方程 (Bellman Equation): 递推关系

- 价值函数 (Value Function)

$$V_{\pi}(s) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s]$$

- 动作-价值函数 (Action-Value Function)

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a]$$

$$\begin{aligned} V_{\pi}(s) &= \mathbb{E}_{\pi}[r_{t+1} + \gamma V_{\pi}(s') | S_t = s] \\ &= \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P(s' | s, a) [r(s, a, s') + \gamma V_{\pi}(s')] \\ &= \mathbb{E}_{a \sim \pi(s, \cdot)} \mathbb{E}_{s' \sim P(\cdot | s, a)} [r(s, a, s') + \gamma V_{\pi}(s')] \end{aligned}$$

- 价值函数取值与时间没有关系，只与策略 π 、在策略 π 下的瞬时奖励以及未来状态价值有关。



贝尔曼方程 (Bellman Equation): 递推关系

- 价值函数 (Value Function)

$$V_{\pi}(s) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s]$$

- 动作-价值函数 (Action-Value Function)

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a]$$

$$\begin{aligned} Q_{\pi}(s, a) &= \mathbb{E}_{\pi}[r_{t+1} + \gamma Q_{\pi}(s', a') | S_t = s, A_t = a] \\ &= \sum_{s' \in \mathcal{S}} P(s' | s, a) \left[r(s, a, s') + \gamma \sum_{a' \in \mathcal{A}} \pi(s', a') Q_{\pi}(s', a') \right] \\ &= \mathbb{E}_{s' \sim P(\cdot | s, a)} \left[r(s, a, s') + \gamma \mathbb{E}_{a' \sim \pi(s', \cdot)} [Q_{\pi}(s', a')] \right] \end{aligned}$$

- 动作-价值函数取值与时间没有关系，只与策略 π 、在策略 π 下的与瞬时奖励和未来动作价值有关。



价值函数与动作-价值函数的关系

- 价值函数 (Value Function)

$$V_{\pi}(s) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s]$$

- 动作-价值函数(Action-Value Function)

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a]$$

$$\begin{aligned} V_{\pi}(s) &= \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s] \\ &= \mathbb{E}_{a \sim \pi(s, \cdot)} [\mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a]] \\ &= \sum_{a \in A} \underbrace{\pi(s, a)}_{\text{采取动作 } a \text{ 的概率}} \times \underbrace{Q_{\pi}(s, a)}_{\text{采取动作 } a \text{ 后带来的回报期望}} \\ &= \sum_{a \in A} \pi(s, a) Q_{\pi}(s, a) \end{aligned}$$



价值函数与动作-价值函数的关系

- 价值函数 (Value Function)

$$V_{\pi}(s) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s]$$

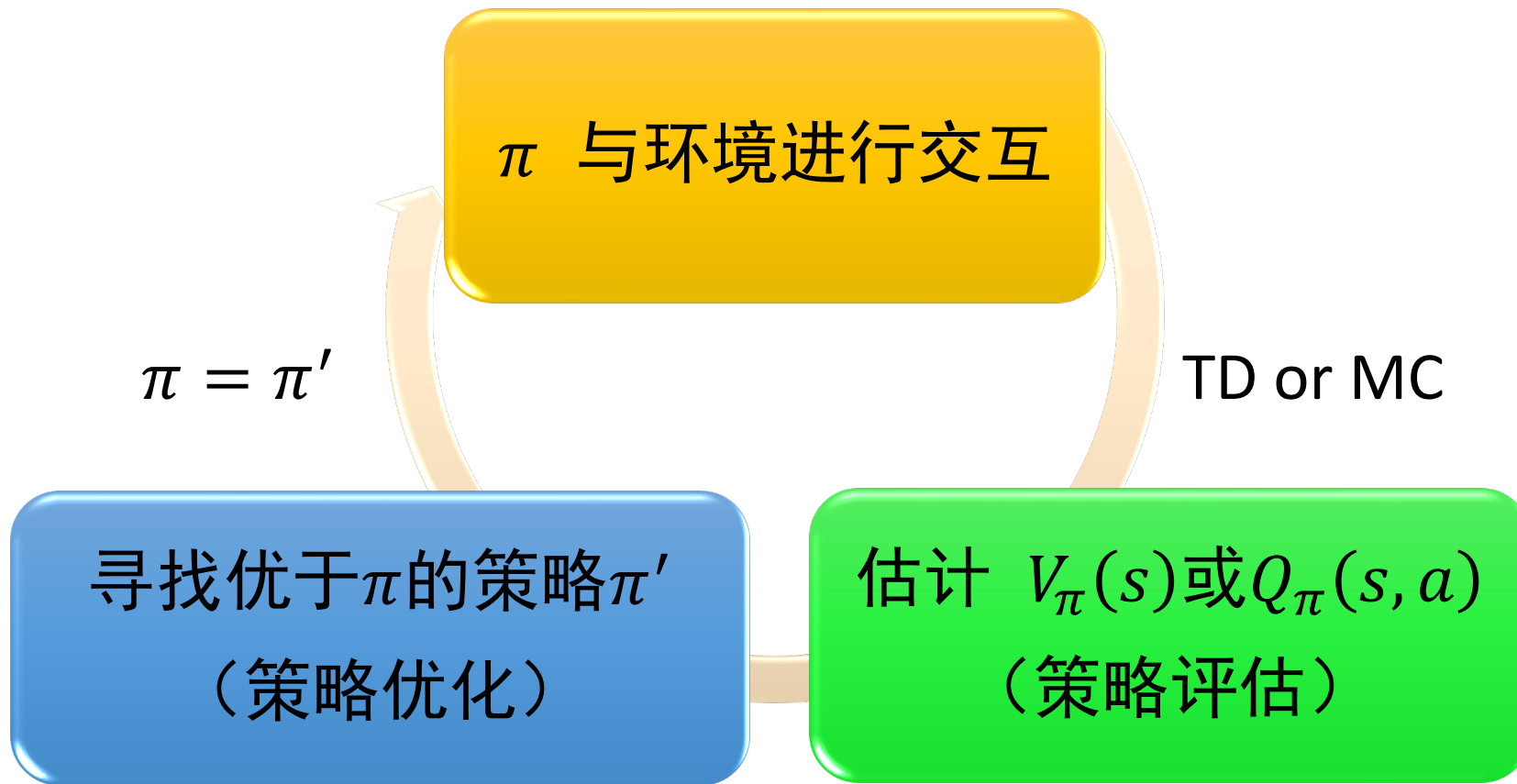
- 动作-价值函数(Action-Value Function)

$$Q_{\pi}(s, a) = \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a]$$

$$\begin{aligned} Q_{\pi}(s, a) &= \mathbb{E}_{\pi}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | S_t = s, A_t = a] \\ &= \mathbb{E}_{s' \sim P(\cdot | s, a)}[r(s, a, s') + \gamma \mathbb{E}_{\pi}[r_{t+2} + \gamma r_{t+3} + \cdots | S_{t+1} = s']] \\ &= \sum_{s' \in \mathcal{S}} \underbrace{P(s' | s, a)}_{\text{在状态 } s \text{ 采取行动 } a \text{ 进入状态 } s' \text{ 的概率}} \times \left[\underbrace{r(s, a, s')}_{\text{在 } s \text{ 采取 } a \text{ 进入 } s' \text{ 得到的回报}} + \gamma \times \underbrace{V_{\pi}(s')}_{\text{在 } s' \text{ 获得的回报期望}} \right] \\ &= \sum_{s' \in \mathcal{S}} P(s' | s, a) [r(s, a, s') + \gamma V_{\pi}(s')] \end{aligned}$$



基于价值的强化学习：策略评估与策略优化

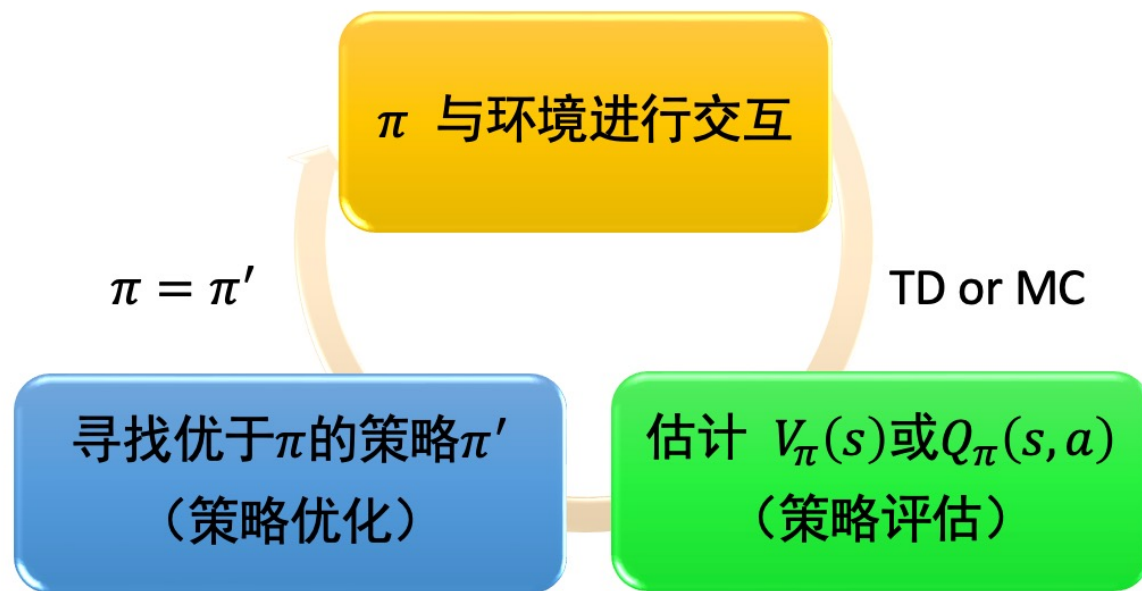




基于价值的强化学习：策略评估与策略优化

为了求解最优策略 π^* ：

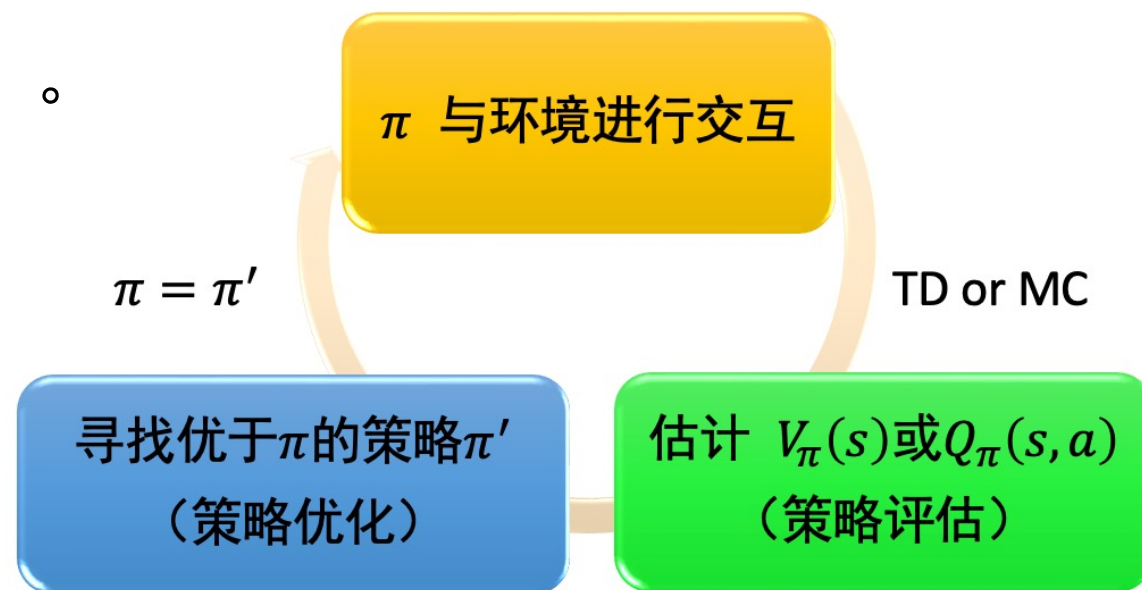
- 从任意策略 π 开始，先计算该策略下价值函数（或动作-价值函数）
- 然后根据价值函数调整改进策略使其更优
- 不断迭代这个过程直到策略收敛





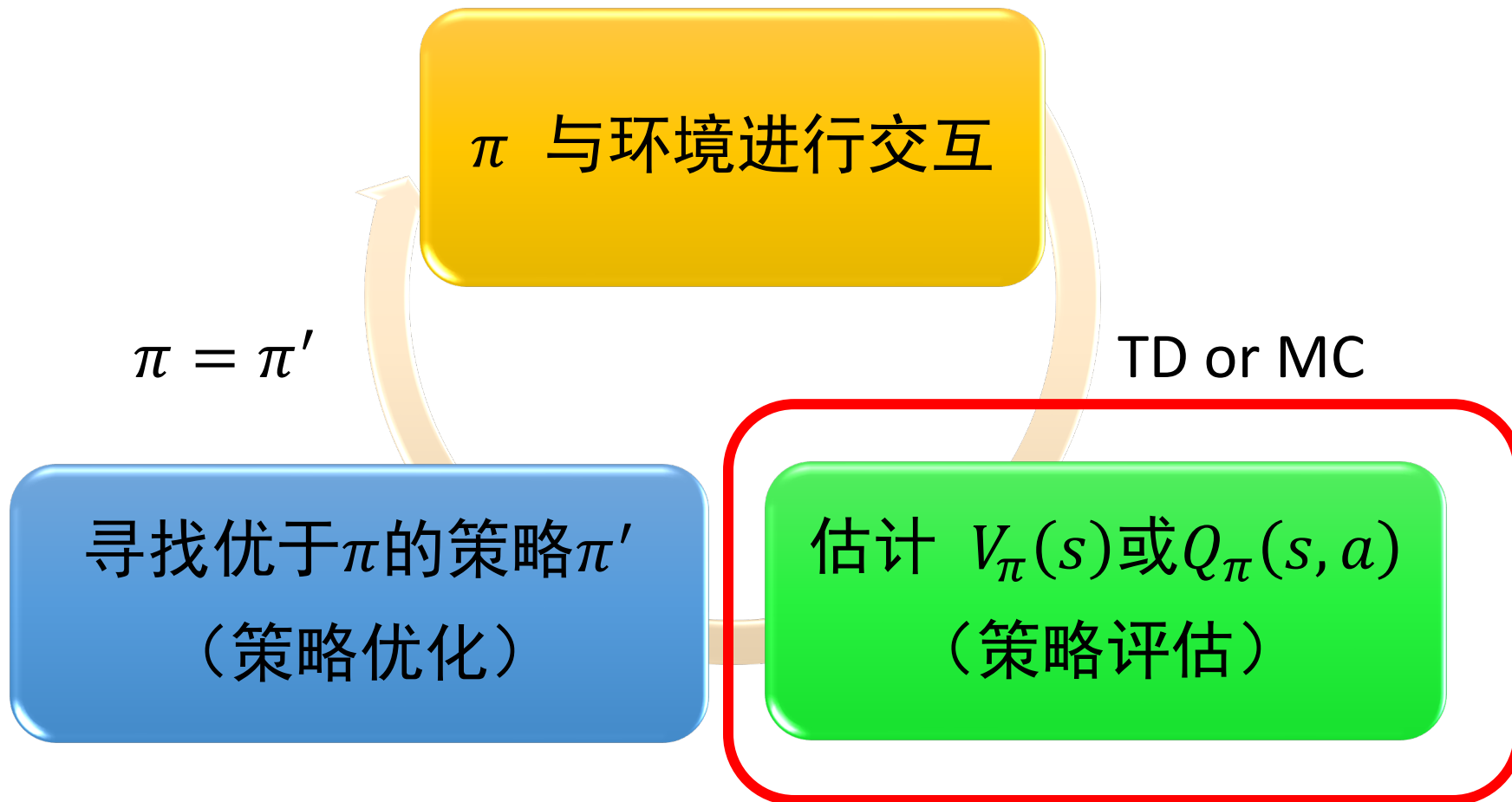
基于价值的强化学习：策略评估与策略优化

- 通过策略计算价值函数的过程叫做**策略评估**（policy evaluation）
- 通过价值函数优化策略的过程叫做**策略优化**（policy improvement）
- 策略评估和策略优化交替进行的强化学习求解方法叫做**通用策略迭代**（Generalized Policy Iteration, GPI）。





基于价值的强化学习：策略评估与策略优化





强化学习中的策略评估方法

- 假定当前策略为 π ，策略评估指的是根据策略 π 来计算相应的价值函数 V_π 或动作-价值函数 Q_π 。三种常见的策略评估方法包括
 - 动态规划（**Model-based** Reinforcement Learning，状态转移概率已知）
 - 蒙特卡洛采样（**Model-free** Reinforcement Learning）
 - 时序差分（Temporal Difference）（**Model-free** Reinforcement Learning）



策略评估：动态规划 - 估计 $V^\pi(s)$

- 动态规划 Dynamic Programming (**DP**)

初始化 V_π 函数： $V_\pi(s) = 0$ （或随机值）

循环

枚举 $s \in S$

$$Q_\pi(s, a) = \sum_{s' \in S} P(s'|s, a) [r(s, a, s') + \gamma V_\pi(s')]$$

$$V_\pi(s) \leftarrow \sum_{a \in A} \pi(s, a) Q_\pi(s, a)$$

直到 V_π 收敛

- 动态规划法的缺点：

- 智能体需要事先知道状态转移概率和奖励函数；无法处理状态集合大小无限的情况



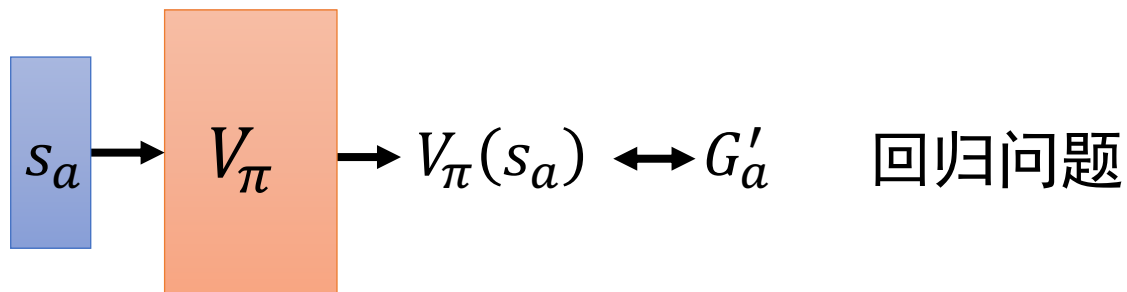
策略评估：蒙特卡洛采样 - 估计 $V^\pi(s)$

- 蒙特卡洛法 Monte-Carlo (MC)

- Critic 观看 π 与环境进行交互。

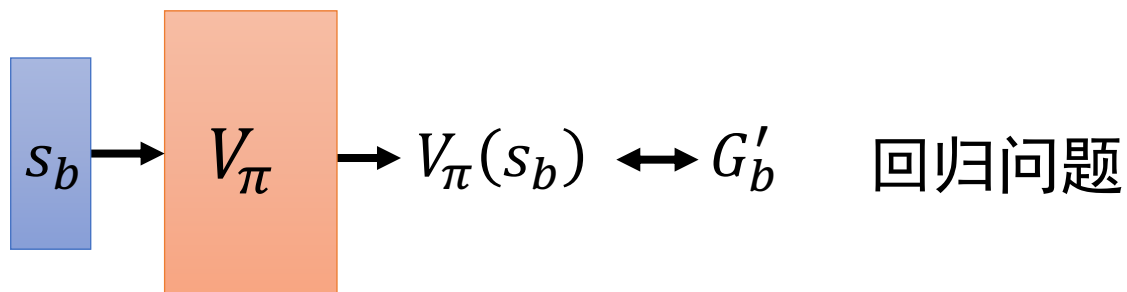
观察到 s_a 后,

在一个片段(episode)结束后,
累积奖励为 G'_a



观察到 s_b 后,

在一个片段(episode)结束后,
累积奖励为 G'_b





策略评估：蒙特卡洛采样 - 估计 $V^\pi(s)$

- 蒙特卡洛法 Monte-Carlo (MC)

- Critic 观看 π 与环境进行交互。

选择不同的起始状态，按照当前策略 π 采样若干轨迹，记它们的集合为 D
枚举 $s \in S$

计算 D 中 s 每次出现时对应的回报 G_1, G_2, \dots, G_k

更新 $V_\pi(s) \leftarrow \frac{1}{k} \sum_{i=1}^k G_i$



策略评估：时序差分 - 估计 $V^\pi(s)$

- 动机：贝尔曼方程
- 核心思想：递推关系。比如

(暂且忽略期望)

$$V_\pi(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \dots$$

$$V_\pi(s_{t+1}) = r_{t+1} + \gamma r_{t+2} + \dots$$

$$V_\pi(s_t) = \gamma V_\pi(s_{t+1}) + r_t$$



策略评估：时序差分 - 估计 $V^\pi(s)$

- 时序差分法 Temporal-difference (TD)

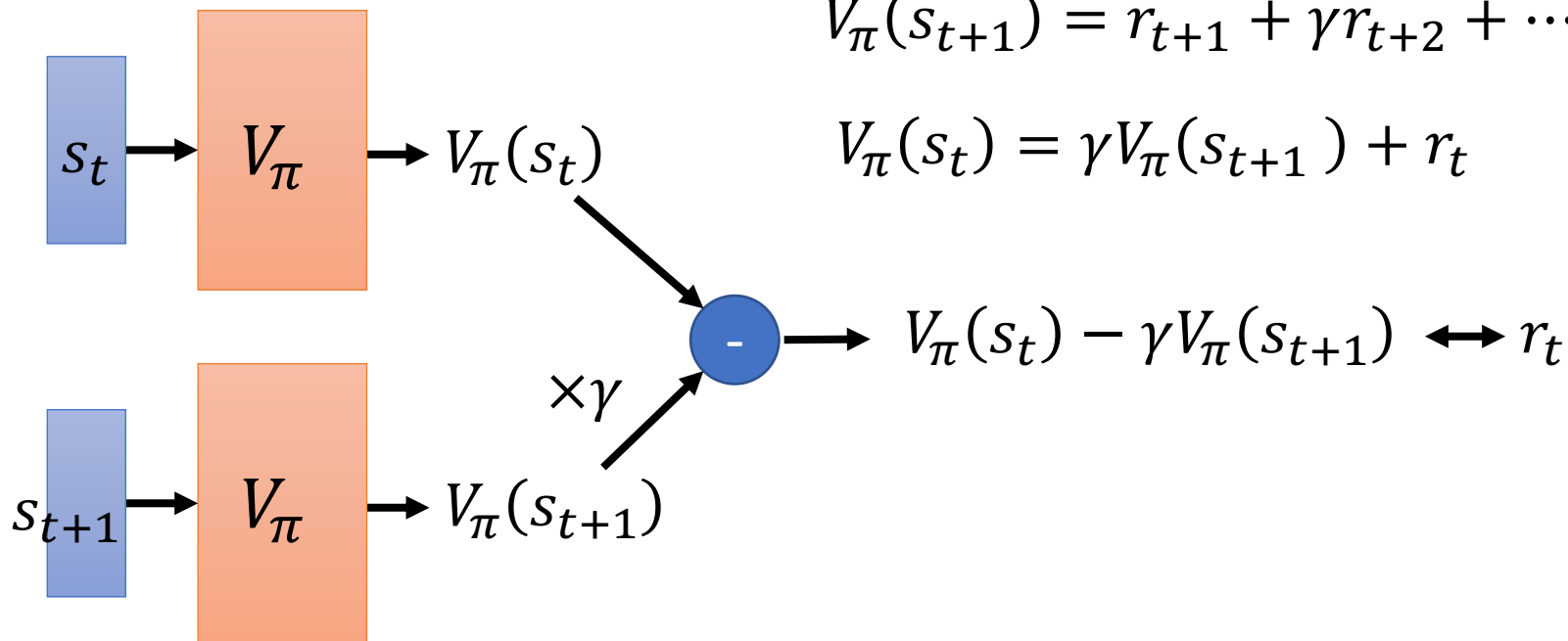
$\cdots S_t, a_t, r_t, S_{t+1} \cdots$

(暂且忽略期望)

$$V_\pi(s_t) = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots$$

$$V_\pi(s_{t+1}) = r_{t+1} + \gamma r_{t+2} + \cdots$$

$$V_\pi(s_t) = \gamma V_\pi(s_{t+1}) + r_t$$





策略评估：时序差分 - 估计 $V^\pi(s)$

- 时序差分法 Temporal-difference (TD)

初始化 V_π 函数

循环

 初始化 s 为初始状态

 循环

$a \sim \pi(s, \cdot)$

 执行动作 a ，观察奖励 r 和下一个状态 s'

 更新 $V_\pi(s) \leftarrow V_\pi(s) + \alpha[r(s, a, s') + \gamma V_\pi(s') - V_\pi(s)]$

$s \leftarrow s'$

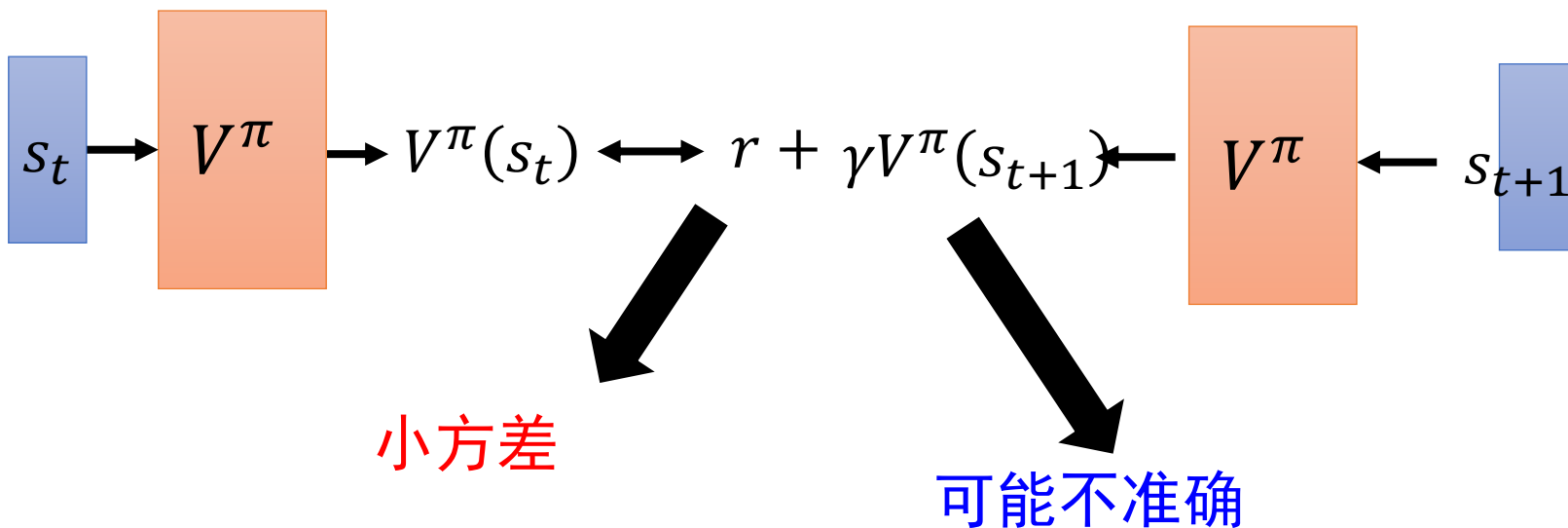
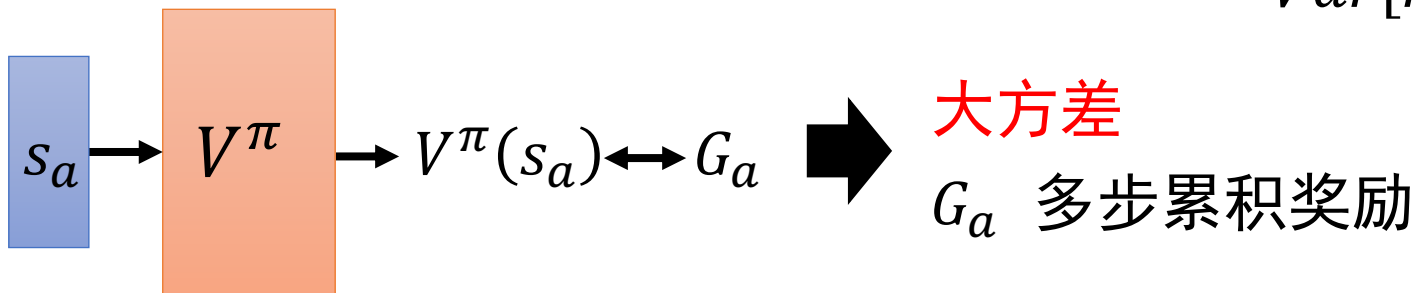
 直到 s 是终止状态

直到 V_π 收敛



蒙特卡洛采样 (MC) v.s. 时序差分 (TD)

$$\text{Var}[kX] = k^2 \text{Var}[X]$$



- 假设critic观察到了以下8个片段(episodes) [Sutton, v2, Example 6.4]

- $s_a, r = 0, s_b, r = 0, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 1, \text{END}$
- $s_b, r = 0, \text{END}$

$$V^\pi(s_b) = 3/4$$

$$V^\pi(s_a) = ?$$

蒙特卡洛法: $V^\pi(s_a) =$ [填空1]

时序差分法: $V^\pi(s_a) =$ [填空2]

(假设 $\gamma = 1$, 并忽略动作.)

作答



蒙特卡洛采样 (MC) v.s. 时序差分 (TD)

[Sutton, v2, Example 6.4]

- 假设critic观察到了以下8个片段(episodes)

- $s_a, r = 0, s_b, r = 0, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 1, \text{END}$

- $s_b, r = 0, \text{END}$

$$V^\pi(s_b) = 3/4$$

$$V^\pi(s_a) = ? \quad 0? \quad 3/4?$$

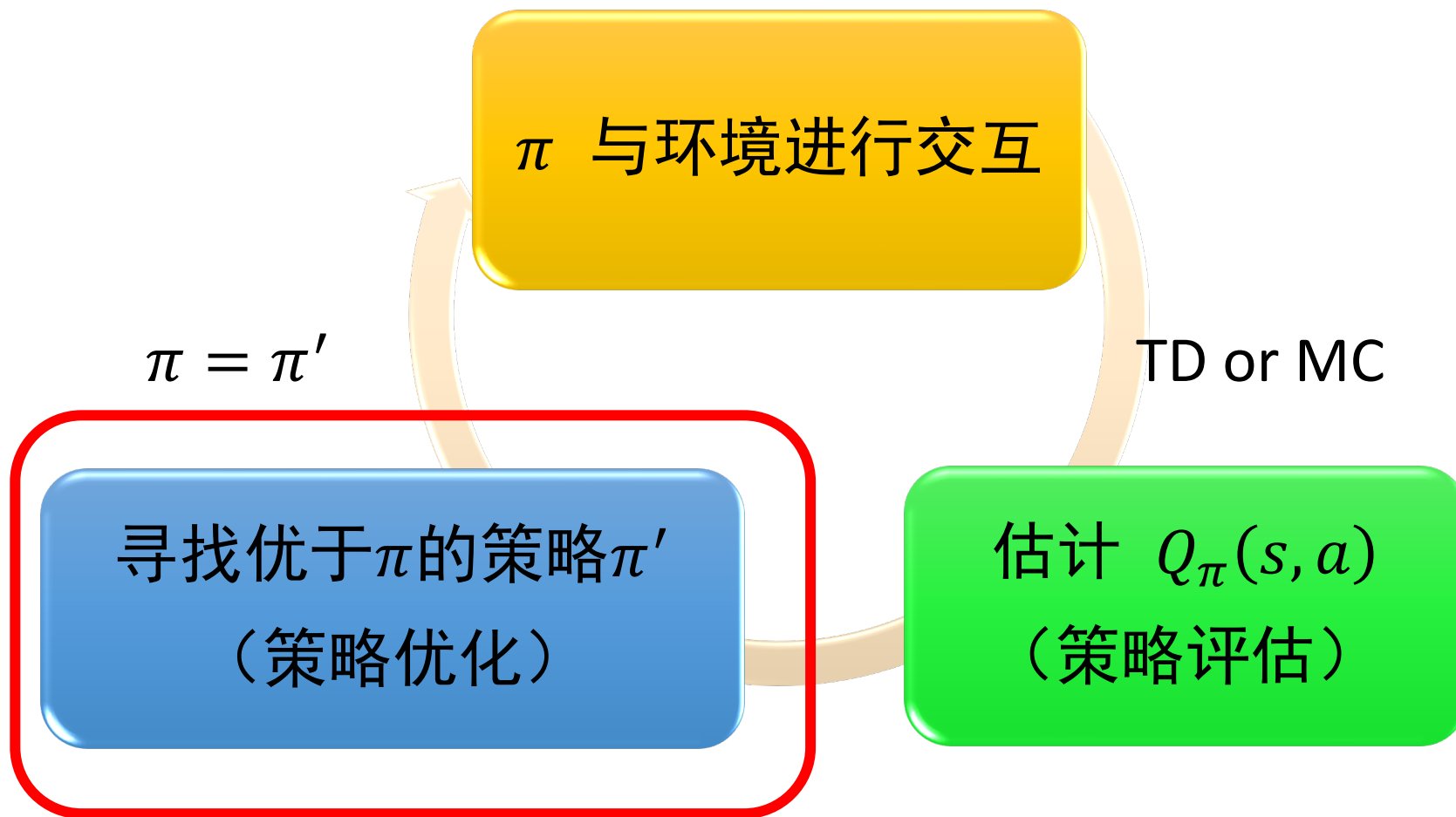
蒙特卡洛法: $V^\pi(s_a) = 0$

时序差分法: $V^\pi(s_a) = V^\pi(s_b) + r$
 $3/4 \quad 3/4 \quad 0$

(假设 $\gamma = 1$, 并忽略动作.)



基于价值的强化学习：策略评估与策略优化





强化学习中的策略优化

策略优化定理：

- 先看某一步特定动作带来的影响
- 对于任意的两个确定的策略 π 和 π' ，如果对于任意状态 $s \in S$

$$Q_{\pi}(s, \pi'(s)) \geq Q_{\pi}(s, \pi(s))$$

注意，不等式左侧的含义是只在当前这一步将动作修改为 $\pi'(s)$ ，未来的动作仍然按照 π 的指导进行

- 那么对于任意状态 $s \in S$ ，有

$$V_{\pi'}(s) \geq V_{\pi}(s)$$

即策略 π' 不比 π 差



强化学习中的策略优化

策略优化定理：

- 延伸到所有可能的动作
- 给定当前策略 π 、价值函数 V_π 和动作-价值函数 Q_π 时，可如下构造新的策略 π' ，只要 π' 满足如下条件：

$$\pi'(s) = \operatorname{argmax}_a Q_\pi(s, a) \quad (\text{对于任意 } s \in S)$$

- π' 便是对 π 的一个改进。于是对于任意 $s \in S$ ，有

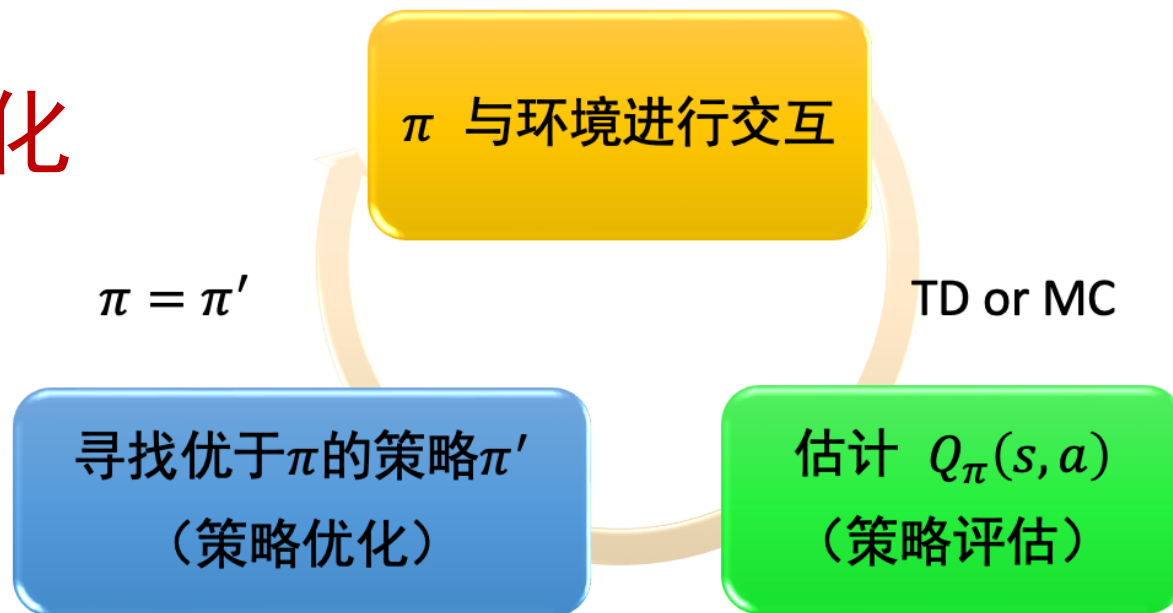
$$Q_\pi(s, \pi'(s)) = Q_\pi(s, \operatorname{argmax}_a Q_\pi(s, a))$$

$$= \max_a Q_\pi(s, a)$$

$$\geq Q_\pi(s, \pi(s)) = V_\pi(s)$$



Q-Learning算法的策略优化



- 给定 $Q_\pi(s, a)$, 找出一个比 π 更优的 π'
 - “更优”: $V_{\pi'}(s) \geq V_\pi(s)$, for all state s

$$\pi'(s) = \arg \max_a Q_\pi(s, a)$$

➤ π' 无需额外可学习的参数, 仅依赖于 Q



基本的Q-Learning 算法

初始化 Q_π 函数

循环片段

初始化 s 为初始状态

循环

$$a = \arg\max_{a'} Q_\pi(s, a')$$

执行动作 a ，观察奖励 r 和下一个状态 s'

$$\text{更新 } Q_\pi(s, a) \leftarrow Q_\pi(s, a) + \alpha \left[r + \gamma \max_{a'} Q_\pi(s', a') - Q_\pi(s, a) \right]$$

$$s \leftarrow s'$$

直到 s 是终止状态

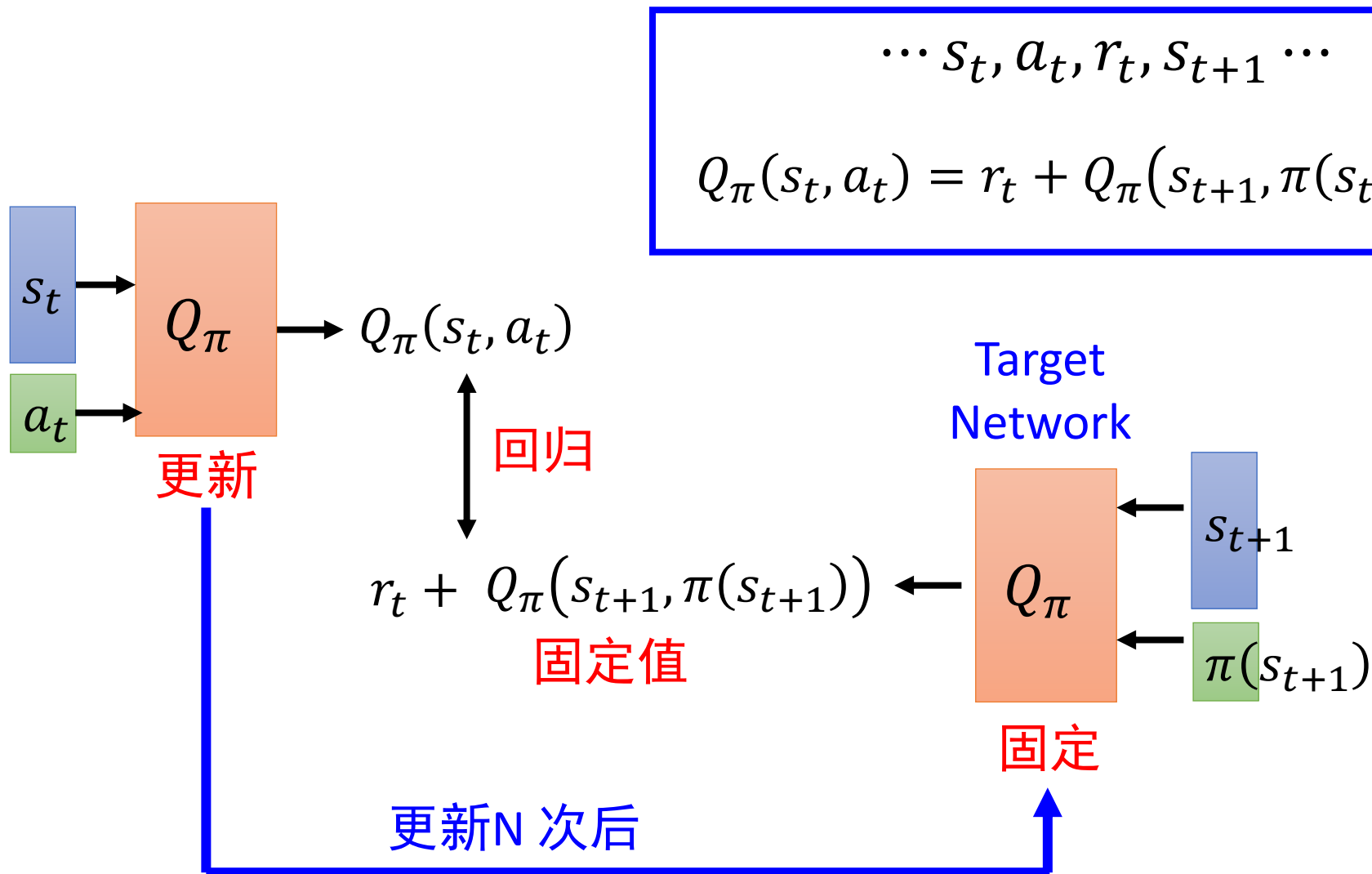
直到 Q_π 收敛

策略优化: $\pi'(s) = \arg\max_a Q_\pi(s, a)$

策略评估: 时序差分

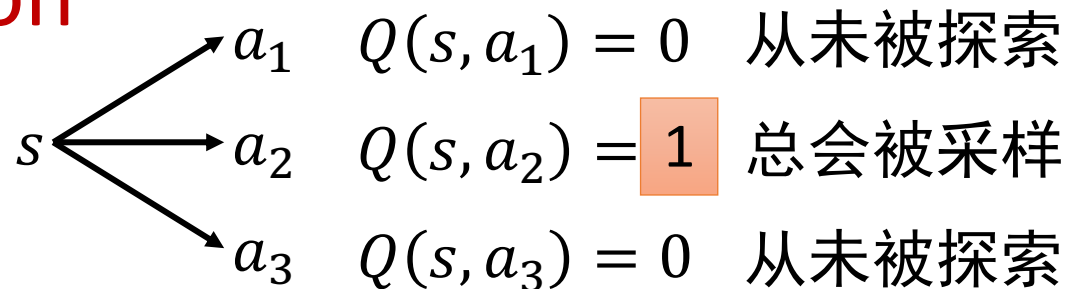


Q-Learning技巧: Target Network





Q-Learning技巧: Exploration



- 根据Q-function可以得到策略

$$a = \arg \max_a Q(s, a)$$

欠优的数据采集方式

Epsilon Greedy

ε 可随训练衰退

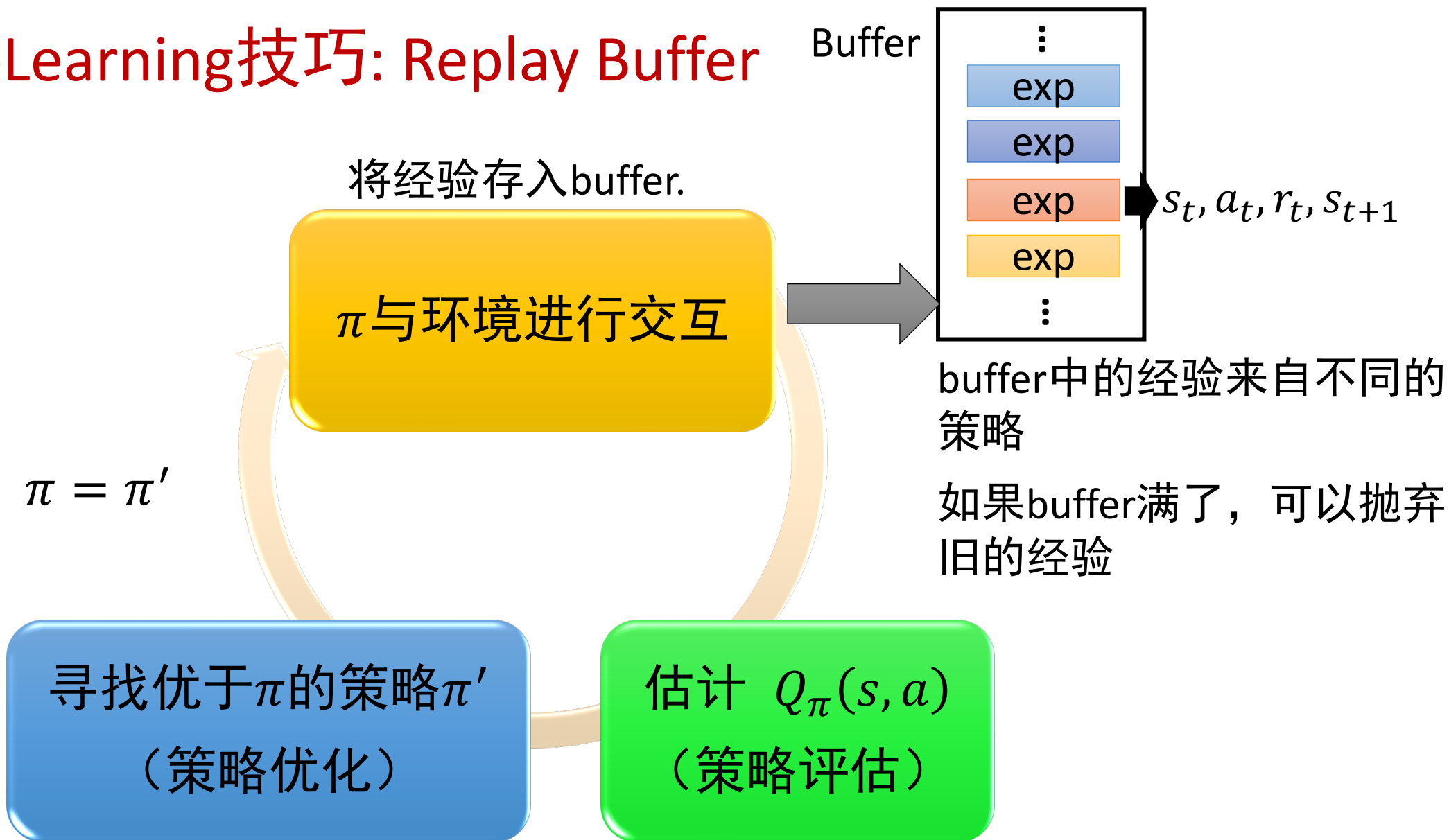
$$a = \begin{cases} \arg \max_a Q(s, a), & \text{with probability } 1 - \varepsilon \\ \text{random}, & \text{otherwise} \end{cases}$$

Boltzmann Exploration

$$P(a|s) = \frac{\exp(Q(s, a))}{\sum_a \exp(Q(s, a))}$$

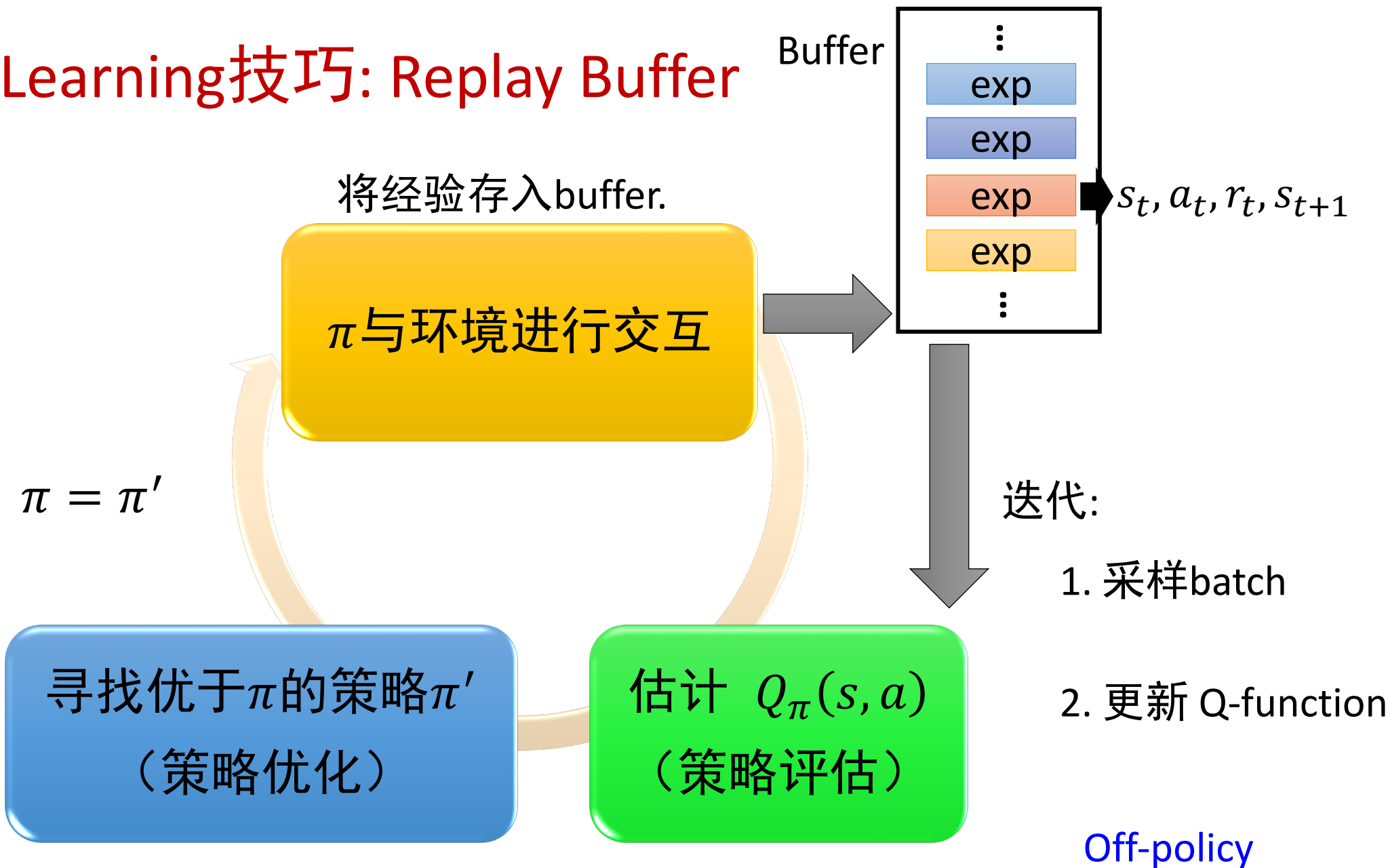


Q-Learning技巧: Replay Buffer





Q-Learning技巧: Replay Buffer





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、强化学习问题定义

二、基于策略的强化学习

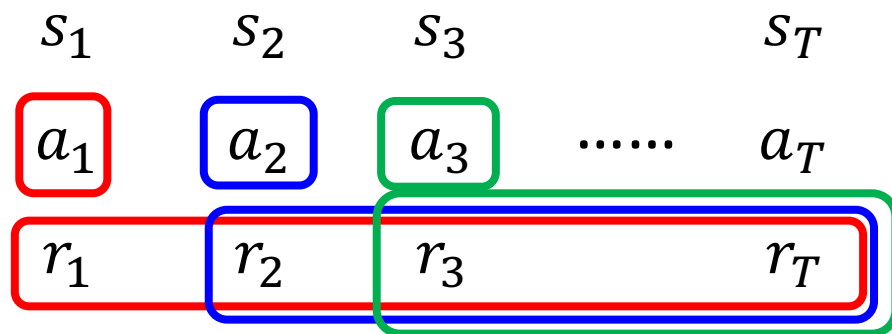
三、基于价值的强化学习

四、Actor-Critic方法

五、其他强化学习方法



回顾：基于策略的强化学习（版本 3）



奖励的好坏是“相对的”

如果所有的 $r_n \geq 10$, 则 $r_n = 10$ 是负奖励...

减去一个基线值 b

???

使得 G'_t 具有正负奖励值

Training Data

$$\{s_1, a_1\} \quad A_1 = G'_1 - b$$

$$\{s_2, a_2\} \quad A_2 = G'_2 - b$$

$$\{s_3, a_3\} \quad A_3 = G'_3 - b$$

\vdots

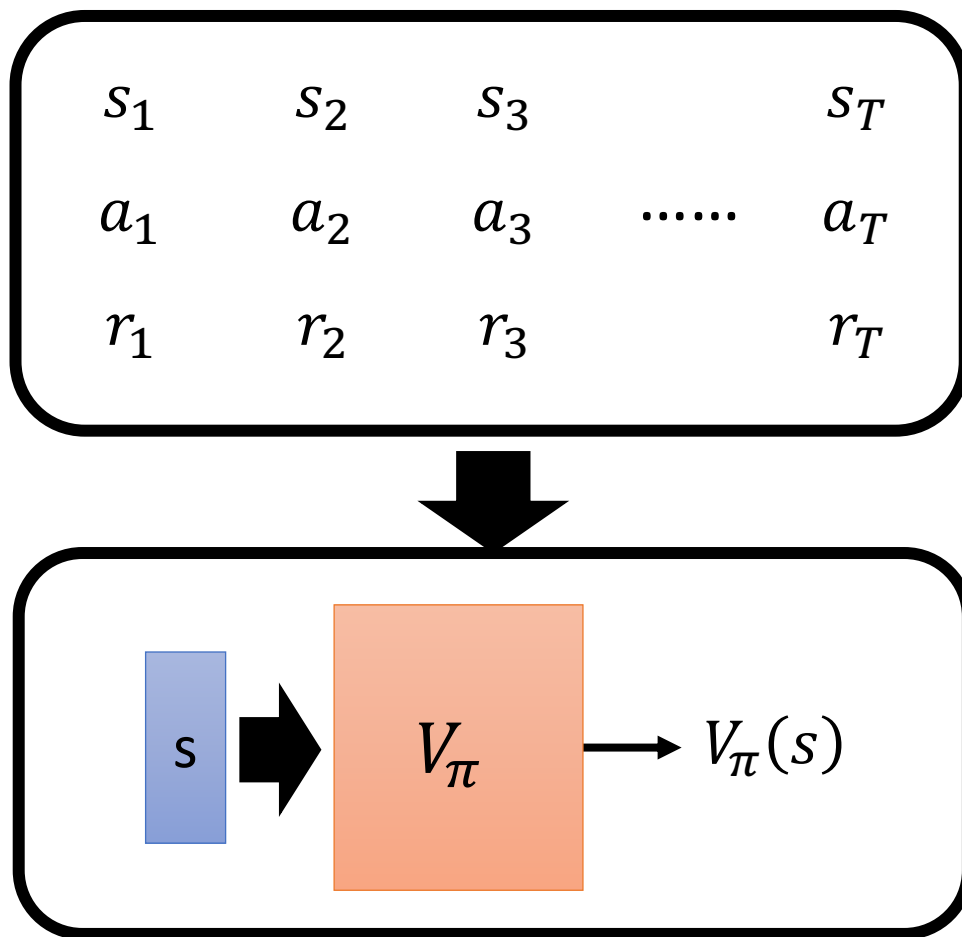
\vdots

$$\{s_T, a_T\} \quad A_T = G'_T - b$$

$$G'_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'}$$



Actor-Critic方法（版本 3.5）

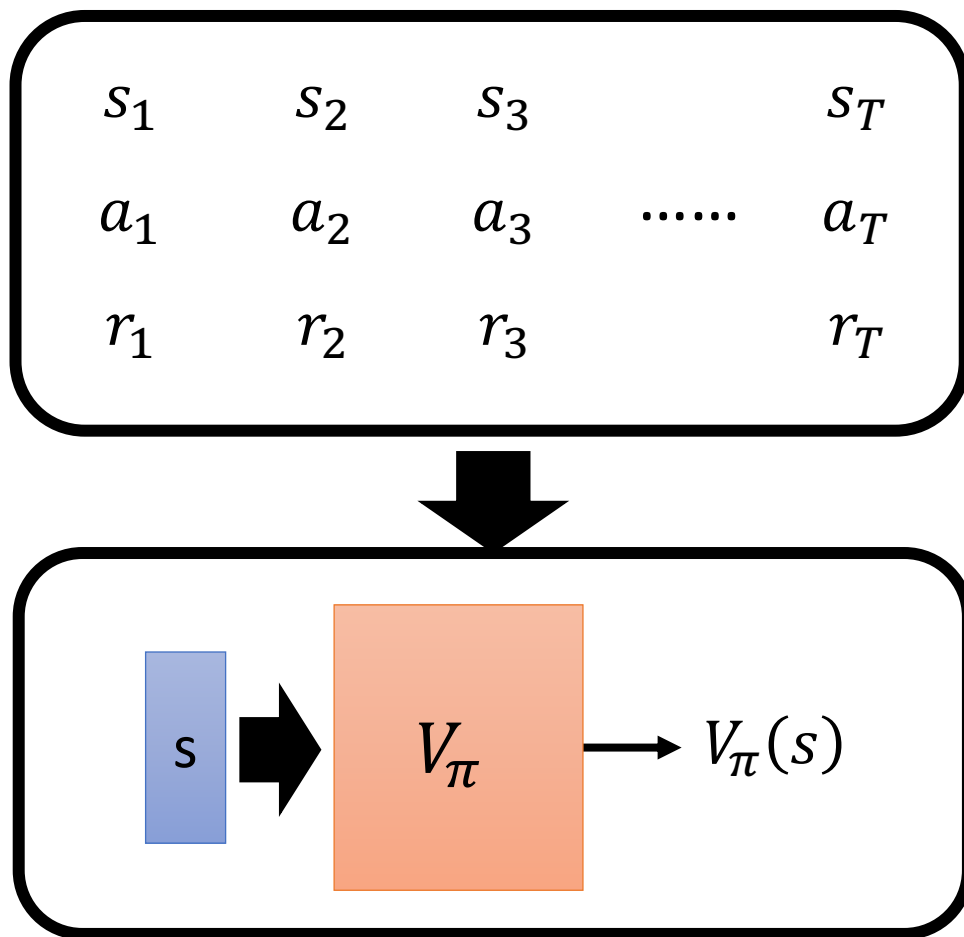


Training Data

$$\begin{array}{ll} \{s_1, a_1\} & A_1 = G'_1 - b \\ \{s_2, a_2\} & A_2 = G'_2 - b \\ \{s_3, a_3\} & A_3 = G'_3 - b \\ \vdots & \vdots \\ \{s_T, a_T\} & A_T = G'_T - b \end{array}$$



Actor-Critic方法（版本 3.5）

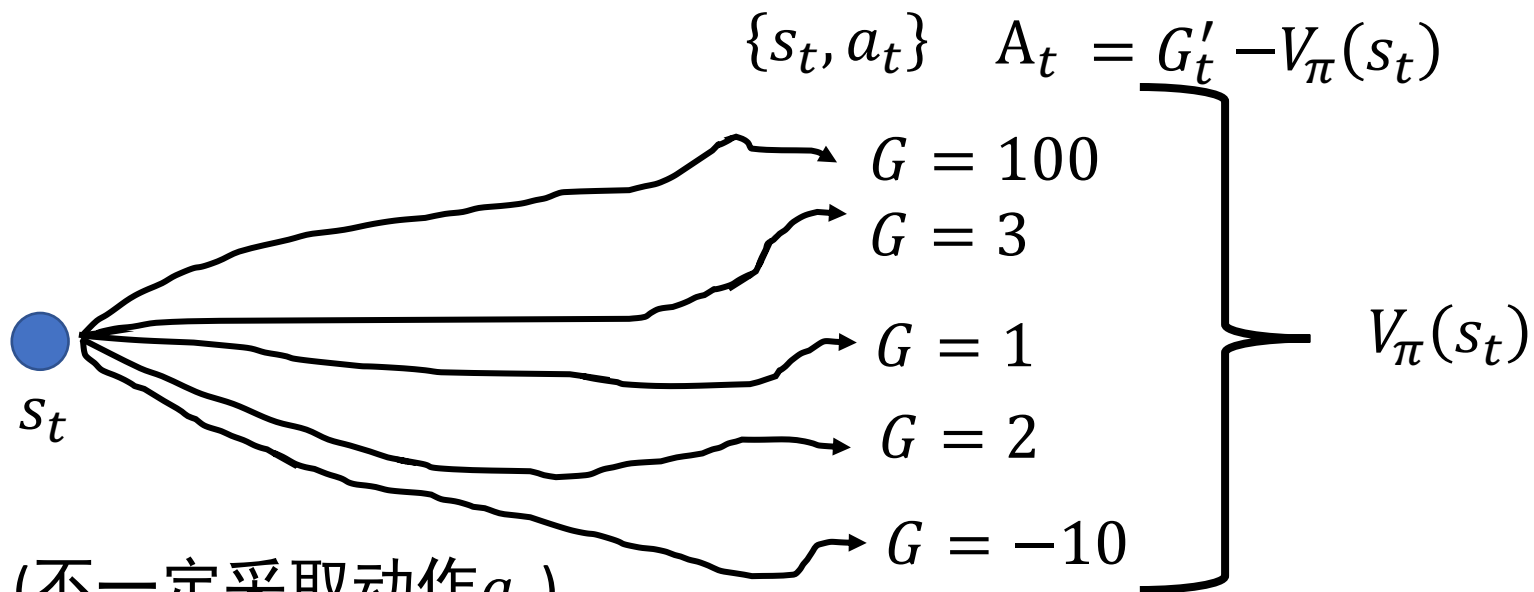


Training Data

$$\begin{aligned} \{s_1, a_1\} \quad A_1 &= G'_1 - V_\pi(s_1) \\ \{s_2, a_2\} \quad A_2 &= G'_2 - V_\pi(s_2) \\ \{s_3, a_3\} \quad A_3 &= G'_3 - V_\pi(s_3) \\ &\vdots \\ \{s_T, a_T\} \quad A_T &= G'_T - V_\pi(s_T) \end{aligned}$$

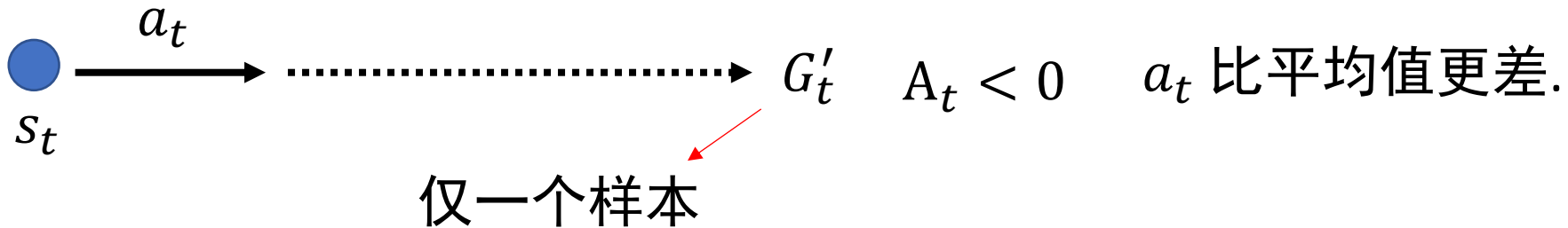


Actor-Critic方法（版本 3.5）



(不一定采取动作 a_t)
(根据分布进行动作采样)

$A_t > 0$ a_t 比平均值更好.



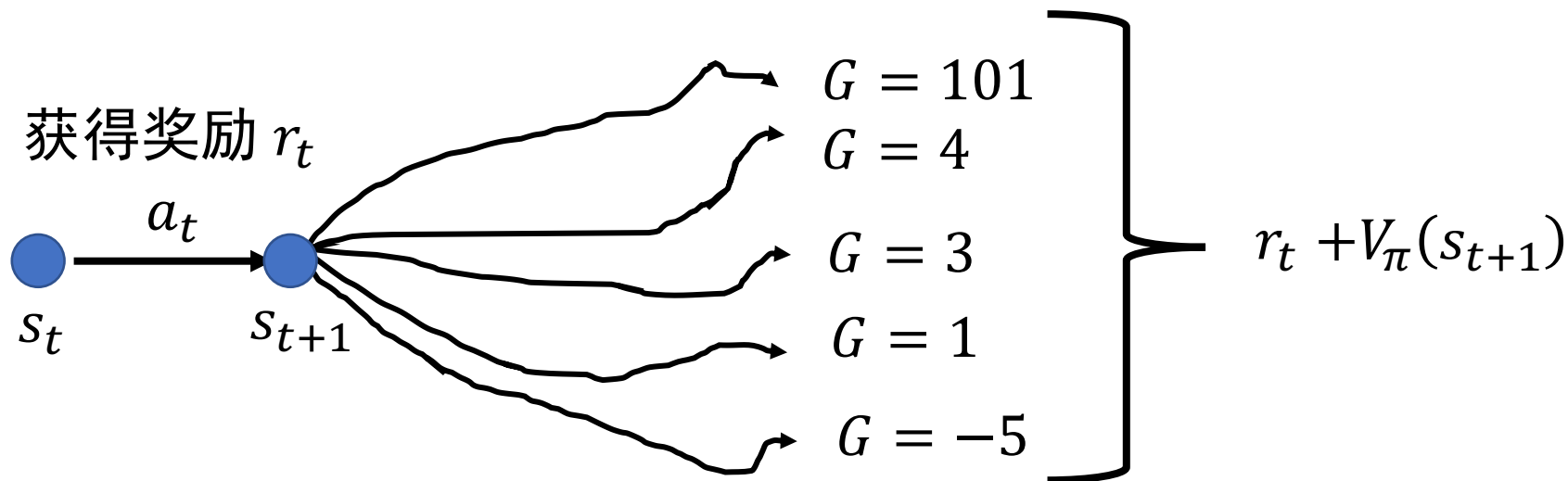
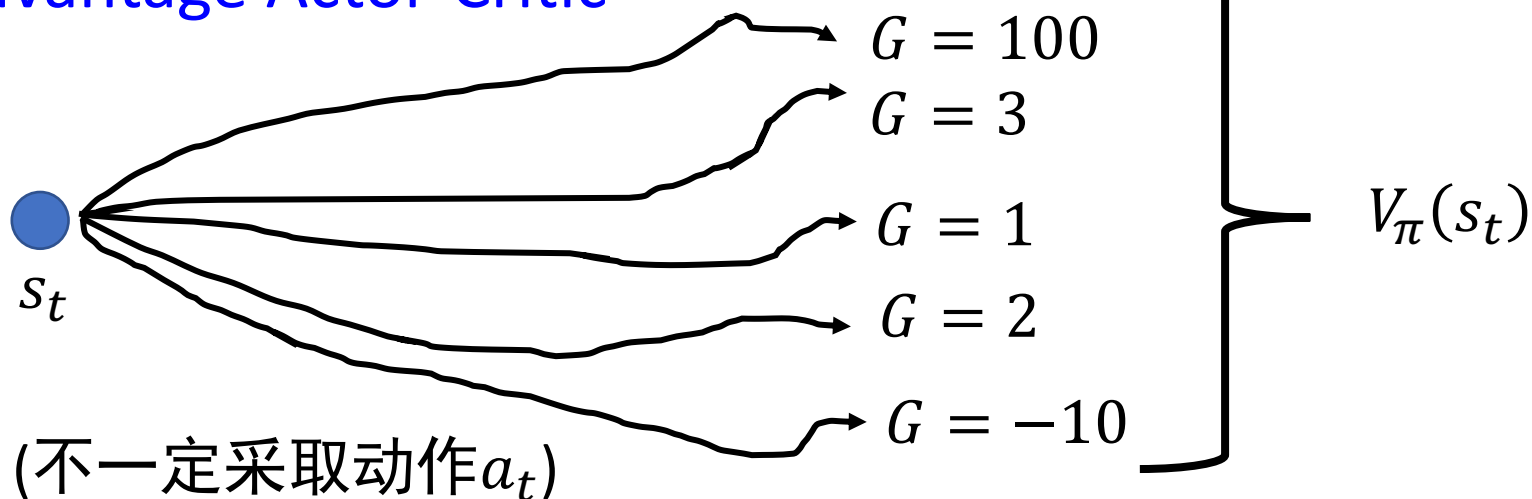


Actor-Critic方法（版本 4）

$$r_t + V_\pi(s_{t+1}) - V_\pi(s_t)$$

Advantage Actor-Critic

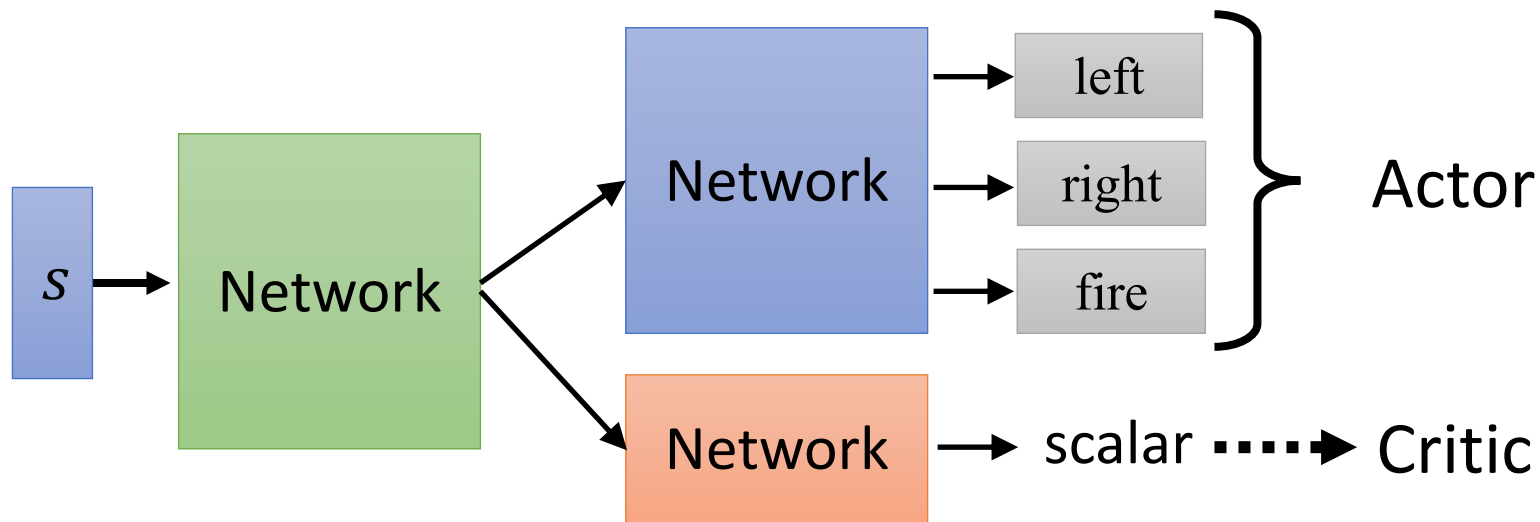
$$\{s_t, a_t\} \quad A_t = \cancel{G'_t - V_\pi(s_t)}$$





Tip of Actor-Critic

- 策略模型(Actor)和评判模型(Critic)参数可以共享





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、强化学习问题定义

二、基于策略的强化学习

三、基于价值的强化学习

四、 Actor-Critic方法

五、其他强化学习方法



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

Reward Shaping



稀疏奖励 Sparse Reward

$$A_t = r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$$

Training Data

s_1	s_2	s_3		s_T
a_1	a_2	a_3	a_T
r_1	r_2	r_3		r_T

$\{s_1, a_1\}$	A_1
$\{s_2, a_2\}$	A_2
$\{s_3, a_3\}$	A_3
\vdots	\vdots
$\{s_T, a_T\}$	A_T

如何多数情况 $r_t = 0$ \longrightarrow 难以判断某个动作的好坏

e.g., 机械手臂栓螺丝

定义额外奖励来指导智能体的学习 \longrightarrow *reward shaping*



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

Reward Shaping

VizDoom

<https://openreview.net/forum?id=Hk3mPK5gg¬eId=Hk3mPK5gg>



Visual Doom AI Competition @ CIG 2016

<https://www.youtube.com/watch?v=94EPSjQH38Y>

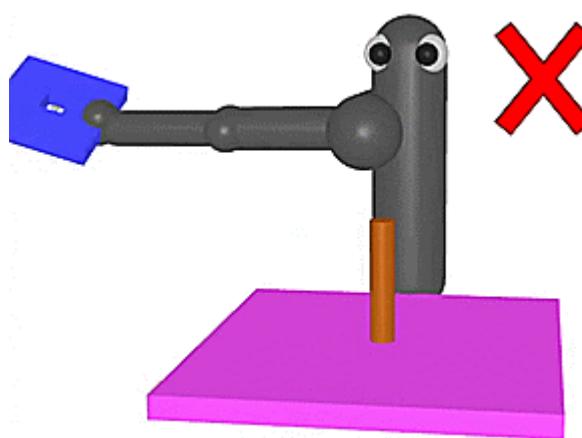
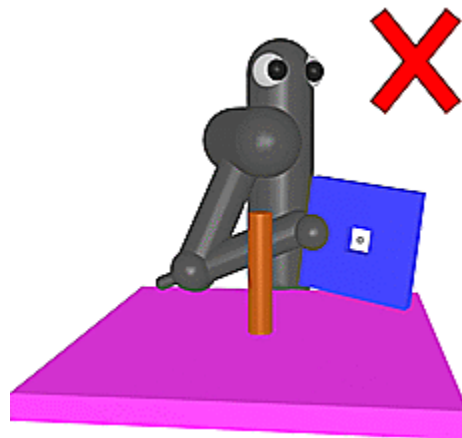
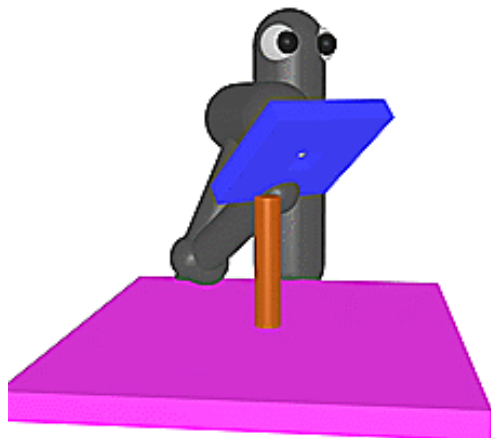


Reward Shaping

VizDoom

<https://openreview.net/forum?id=Hk3mPK5gg¬eId=Hk3mPK5gg>

Parameters	Description	FlatMap	CIGTrack1
living	Penalize agent who just lives	-0.008 / action	
health_loss	Penalize health decrement	-0.05 / unit	
ammo_loss	Penalize ammunition decrement	-0.04 / unit	
health_pickup	Reward for medkit pickup	0.04 / unit	
ammo_pickup	Reward for ammunition pickup	0.15 / unit	
dist_penalty	Penalize the agent when it stays	-0.03 / action	
dist_reward	Reward the agent when it moves	9e-5 / unit distance	





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

No Reward: Learning from Demonstration



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

动机

- 有些任务难以定义奖励。
- 手工设计奖励可能会导致不可控行为



Three Laws of Robotics:

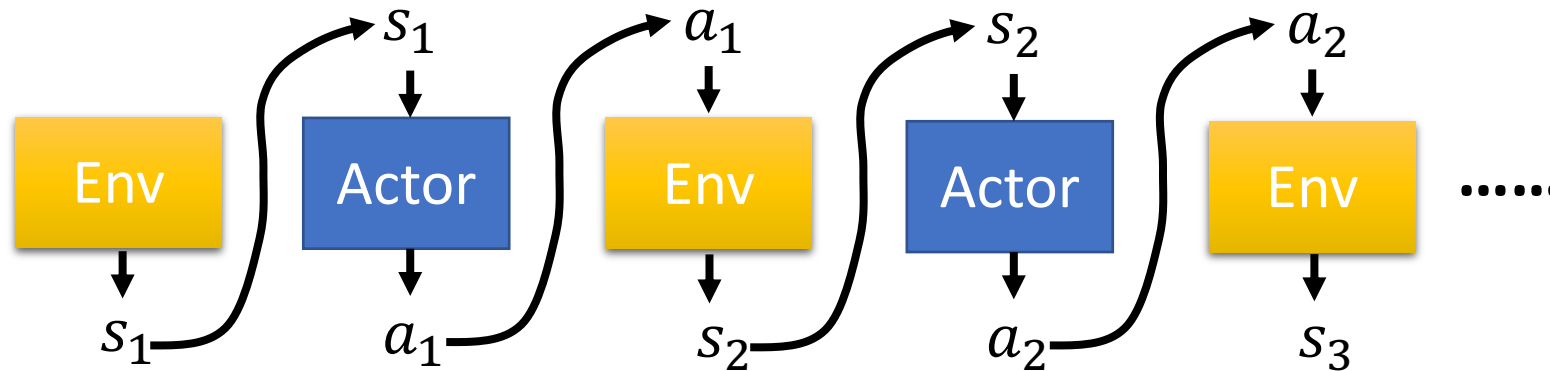
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.



restraining individual human behavior and sacrificing
some humans will ensure humanity's survival

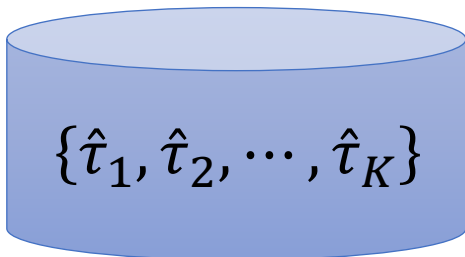


模仿学习 Imitation Learning



智能体可以与环境交互，但是奖励函数未知

专家示范



\hat{t} 为专家轨迹

自动驾驶：记录司机驾驶行为

机械手臂：握住机械手臂抓取



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

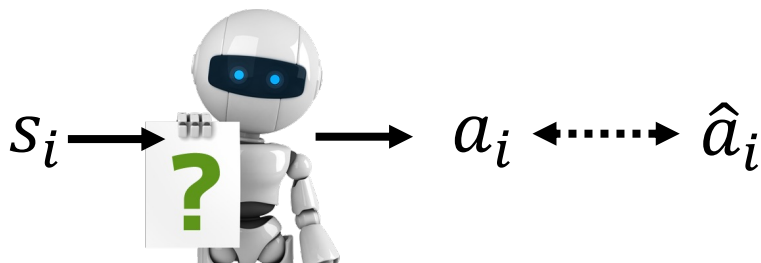
监督学习?

- 以自动驾驶为例

$$\hat{\tau} = \{s_1, \hat{a}_1, s_2, \hat{a}_2, \dots\}$$

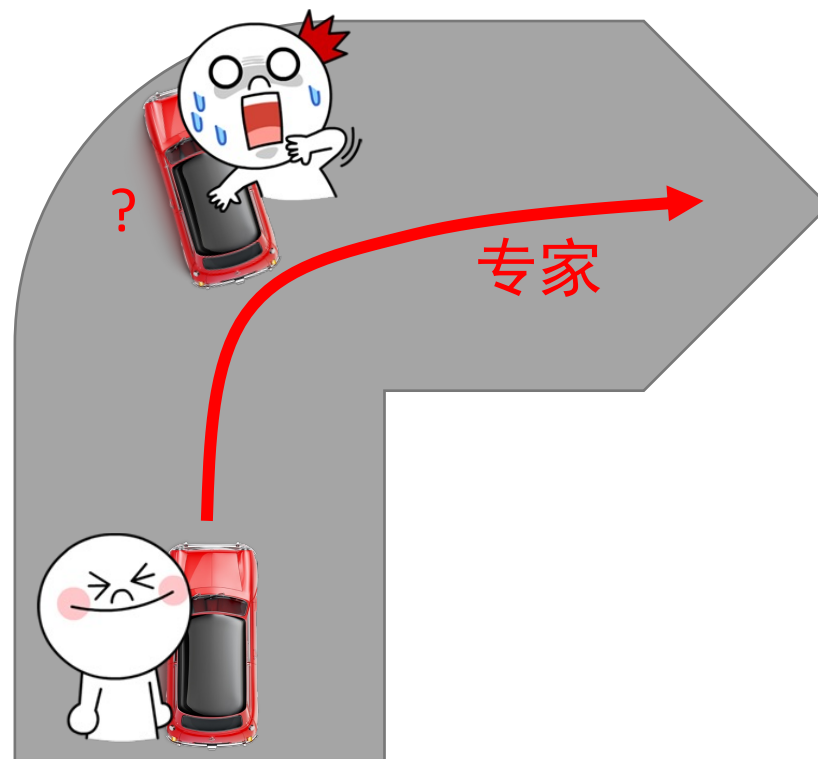


forward



问题：只能采样有限数据

Yes, also known as
Behavior Cloning





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

More problem

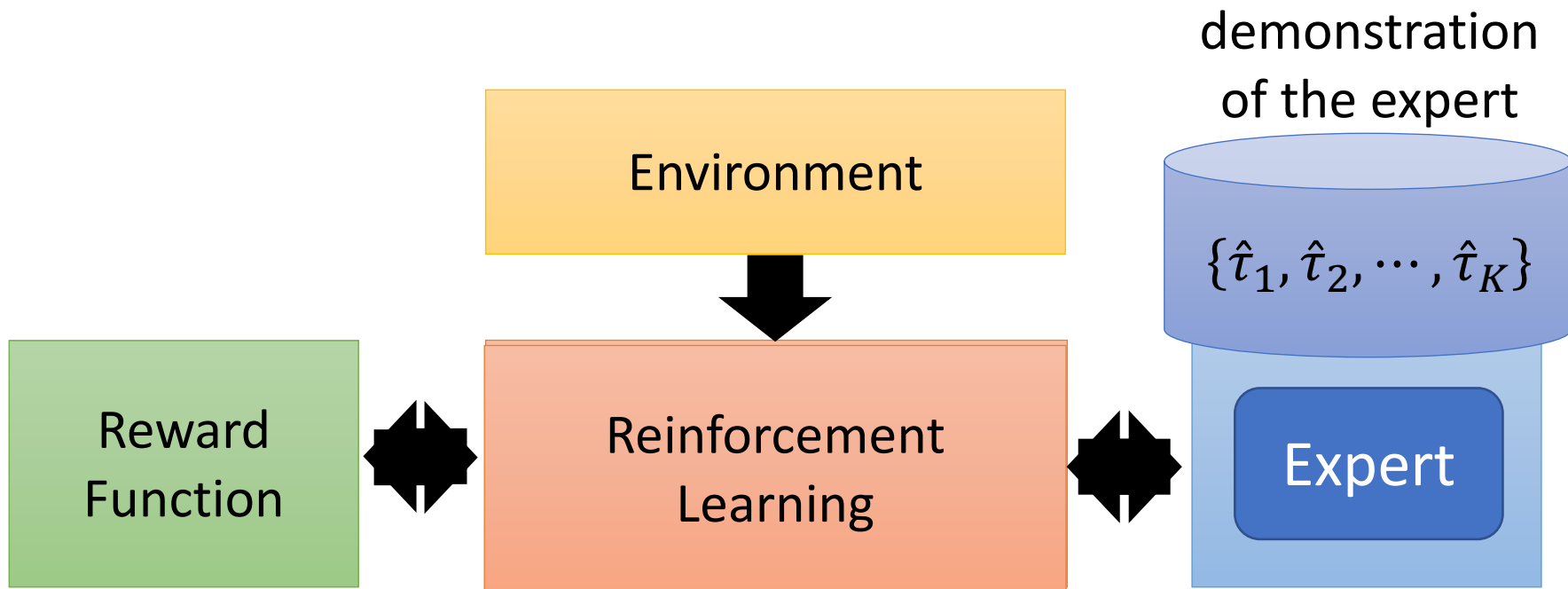
The agent will copy every behavior, even irrelevant actions.



<https://www.youtube.com/watch?v=j2FSB3bseek>



逆强化学习 Inverse Reinforcement Learning



Using the reward function to find the *optimal actor*.



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

更多阅读

- 书籍: Reinforcement Learning: An Introduction
 - <http://incompleteideas.net/sutton/book/the-book.html>

