北京郵電大學
Beijing University of Posts and Telecommunications

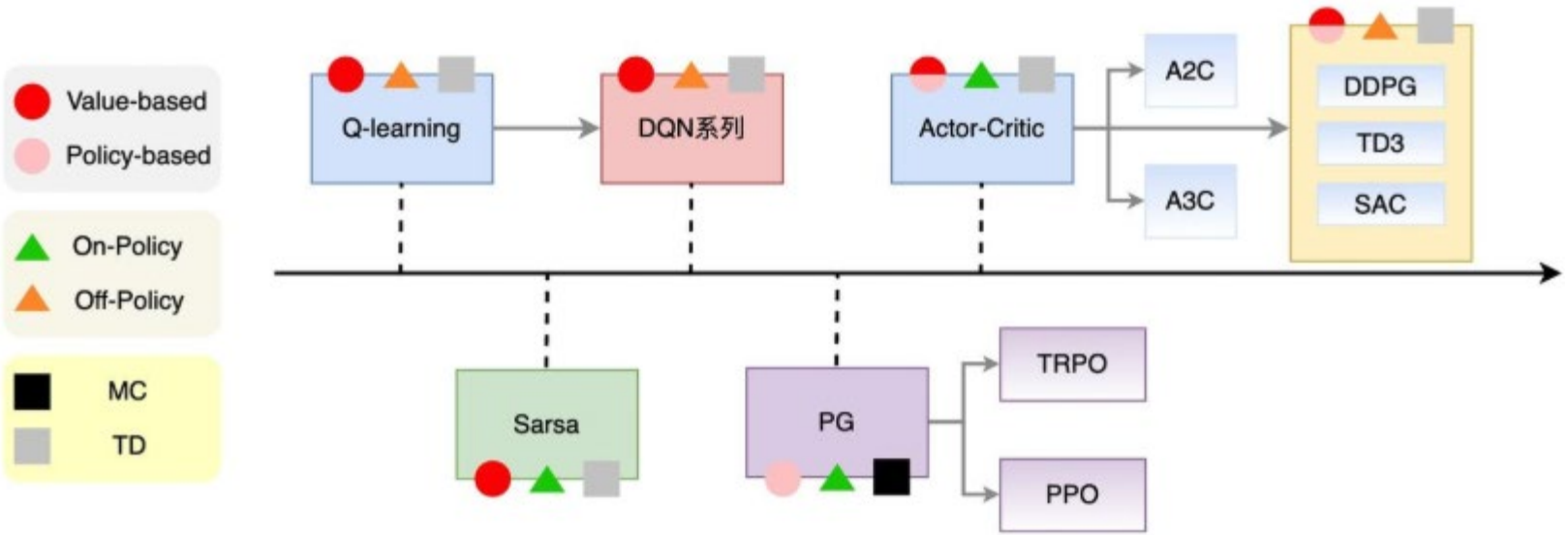
强化学习

张一凡

2025.6.2

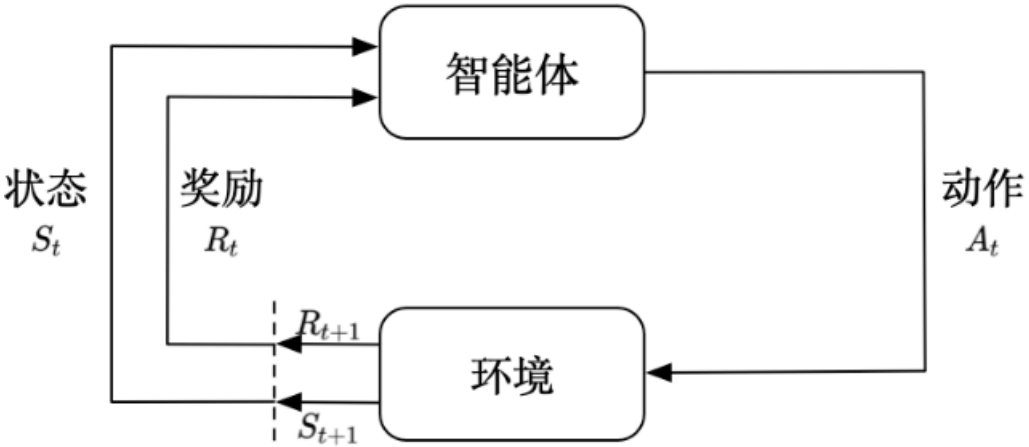
目录

1. 强化学习基础
2. 常用算法框架
3. 多智能体强化学习
4. RLHF



强化学习基础知识

- 强化学习讨论的问题是在复杂、不确定的环境中找到能使智能体获得的奖励最大的决策序列。
  - 时间相关的序列学习
  - 无标注数据，只有延时奖励
- 监督学习
  - 有明确标注数据
  - 样本往往假设是独立同分布的



如何在环境中探索？
如何利用经验（数据）？

强化学习组成

- 策略(policy)：智能体会用策略来选取下一步的动作。
  - 随机性策略 $\pi(a|s) = p(a = a_t | s = s_t)$ 1
  - 确定性策略 $a^* = \arg\max \ \pi(a|s)$ 2
- 价值函数(value function)：对累计获得奖励的描述。
  - 状态价值函数(V)

$$V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s] \quad 4$$

  - 动作价值函数(Q)

$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a] \quad 5$$

- 模型(model)：表示智能体对环境的状态进行理解，它决定了环境中世界的运行方式。
  - 马尔可夫决策过程
  - 部分可观测马尔可夫决策过程

$$(S, A, T, R, \Omega, O, \gamma) \quad 6$$

$$T := T(s' | s, a)$$

$$\Omega := (o | s, a)$$

$$R: = \mathbb{E}[r_{t+1} | s_t, a_t]$$

马尔可夫过程

- 马尔可夫奖励过程：计算马尔可夫过程中的奖励累计/状态价值函数。
  - 蒙特卡洛(Monte Carlo,MC)法
  - 贝尔曼方程

$$V^t(s)=\mathbb{E}[G_t \mid s_t=s] \quad (1)$$

$$=\mathbb{E}[r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\ldots+\gamma^{T-t-1} r_T \mid s_t=s]$$

$$V(s)=R(s)+\gamma \mathbb{E}[V(s_{t+1}) \mid s_t=s)]=R(s)+\gamma \sum p(s' \mid s)V(s') \quad (2)$$

- 马尔科夫决策过程：未来的状态不仅依赖于当前的状态，也依赖于在当前状态智能体采取的动作。

$$P_\pi(s' \mid s)=\sum_{a\in A}\pi(a \mid s)p(s' \mid s,a) \quad (3)$$

$$r_\pi(s)=\sum_{a\in A}\pi(a \mid s)R(s,a) \quad (4)$$

$$Q(s,a)=R(s,a)+\gamma \sum_{s'\in S} p(s' \mid s,a)V(s') \quad (5)$$

$$V_\pi(s)=\sum_{a\in A}\pi(a \mid s)Q_\pi(s,a) \quad (6)$$

$$Q_\pi(s,a)=R(s,a) \quad (7)$$

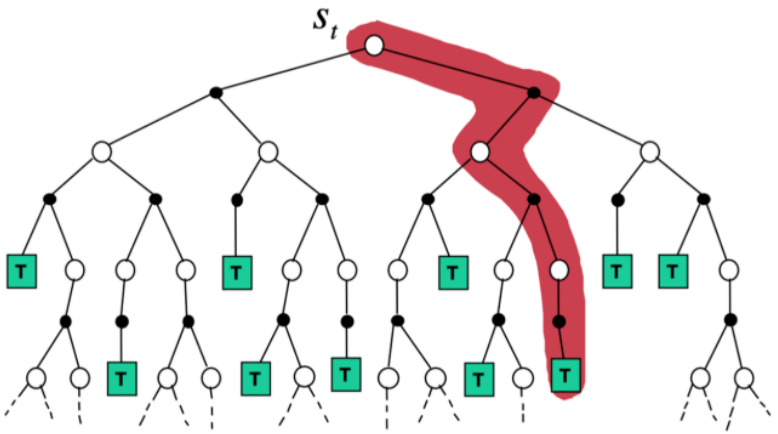
$$+\gamma \sum_{s'\in S} p(s' \mid s,a)\sum_{a'\in A}\pi(a' \mid s')Q_\pi(s',a')$$

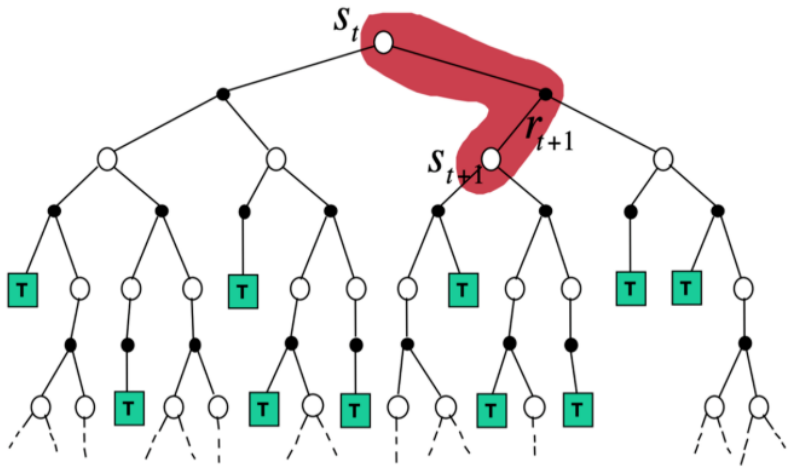
状态价值函数计算方法/采样方法

- 蒙特卡洛法是指我们可以采样大量的轨迹，计算所有轨迹的真实回报，然后计算平均值。
- 时序差分方法（Temporal Difference,TD)的目的是对于某个给定的策略，在线算出它的价值函数。

$$
\begin{aligned}
n=1 \ (\mathrm{TD}) \quad & G_t^{(1)} = r_{t+1} + \gamma V(s_{t+1}) \\
n=2 \quad & G_t^{(2)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 V(s_{t+2}) \\
& \vdots \\
n=\infty \ (\mathrm{MC}) \quad & G_t^{\infty} = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{T-t-1} r_T
\end{aligned}
$$

$$
\begin{aligned}
V(s_t) &\leftarrow V(s_t) + \alpha(r_{t+1} + \gamma V(s_{t+1}) - V(t)) \\
&\leftarrow V(s_t) + \alpha(G_{i,t} - V(s_t))
\end{aligned}
$$





On policy与Off policy

- On policy同策略方法，使用同一个策略进行探索和利用。
- Off policy异策略方法分离了目标策略与行为策略。

对于每一个回合进行循环。
初始化 S。
使用从Q中衍生出的策略（例如ε-贪心策略）从S中选择A。
对一个回合中的每一步进行循环。
执行动作A,观测R, S'。
使用从Q中衍生出的策略（例如ε-贪心策略）从S'中选择A'。
$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma Q(S',A') - Q(S,A)]$。
$S \leftarrow S'; A \leftarrow A'$。
直到S到达终点。

对每一个回合进行循环。
初始化 S。
对一个回合中的每一步进行循环。
使用从Q中衍生出的策略（例如ε-贪心策略）从S中选择A。
执行动作A,观测R, S'。
$$Q(S,A) \leftarrow Q(S,A) + \alpha\left[R + \gamma \max_a Q(S',a) - Q(S,A)\right]$$
$S \leftarrow S'$。
直到S到达终点。

策略梯度(Policy Gradient)

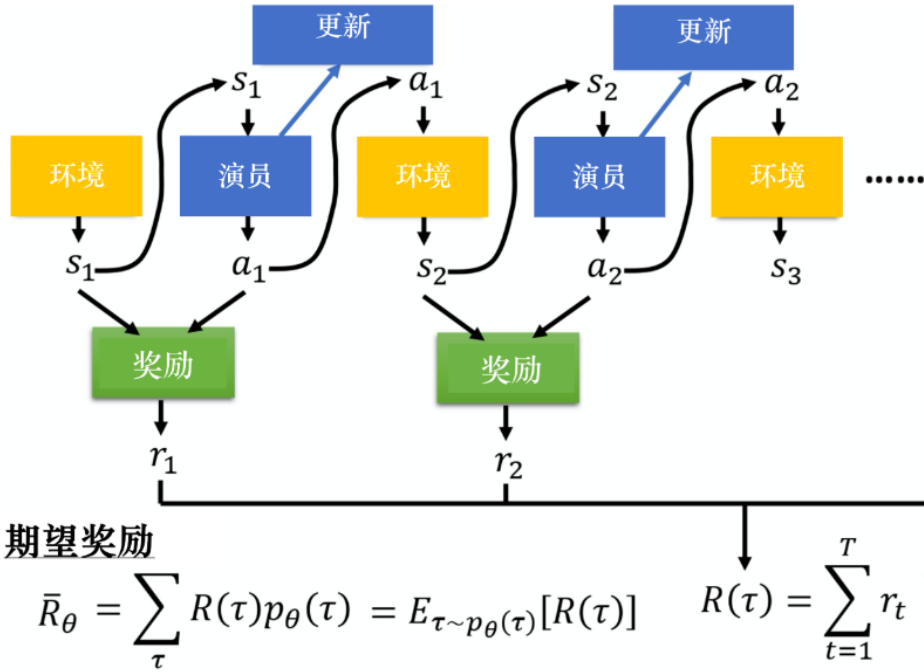
- 基于值函数的方法有一些缺陷，最关键的问题在于要求动作空间是离散的，更直接的方法是直接关注策略本身的策略梯度法。

$$\nabla \bar{R}_\theta = \sum_\tau R(\tau)\nabla p_\theta(\tau) \quad (1)$$

$$= \sum_\tau R(\tau) p_\theta(\tau) \nabla \log p_\theta(\tau)$$

$$= \mathbb{E}_{\tau \sim p_\theta(\tau)}[R(\tau)\nabla \log p_\theta(\tau)]$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T} R(\tau^n)\nabla \log p_\theta(a_t^n \mid s_t^n)$$



期望奖励

$$\bar{R}_\theta = \sum_\tau R(\tau)p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)] \qquad R(\tau) = \sum_{t=1}^{T} r_t$$

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\left(\sum_{t'=t}^{T}\gamma^{t'-t} r_{t'}^{\,n} - b\right)\nabla \log p_\theta(a_t^n \mid s_t^n) \quad (2)$$

Q-learning

- Q-Table法
- 深度Q网络（deep Q-network，DQN）是基于深度学习的Q学习算法，结合了价值函数近似与神经网络技术。

$$Q \approx Q_\theta$$

States

Q(st, at)	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10
a1	22.0	19.8	17.8	16.0	14.4	34.4	22.0	19.8	11.7	13.0
a2	21.0	19.2	7.8	13.0	15.4	33.4	20.0	19.9	21.7	18.0
a3	21.6	21.2	6.8	13.5	15.1	33.9	20.7	12.9	21.3	18.8
a4	11.6	21.1	6.2	23.5	35.1	23.9	10.7	22.9	24.3	28.8

Actions

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

近端策略优化Proximal Policy Optimization

- 重要性采样(Importance Sampling)：如何用一个简单的分布去估计原始分布的期望。
- 近端策略优化(PPO):KL散度版和clip版。
  - 限制策略更新幅度
  - 使用广义优势估计

1 $$\nabla \bar{R}_\theta = \mathbb{E}_{\tau \sim p_\theta(\tau)}[R(\tau)\nabla \log p_\theta(\tau)]$$

$$E = \mathbb{E}_{x \sim p(x)} f(x)$$

2 $$= \int p(x) f(x) dx$$

$$= \int q(x) \frac{p(x)}{q(x)} f(x) dx$$

$$= \mathbb{E}_{x \sim q(x)}[\frac{p(x)}{q(x)} f(x)]$$

3 $$\nabla \bar{R}_\theta = \mathbb{E}_{\tau \sim p_{\theta'}(\tau)}[\frac{p_\theta(\tau)}{p_{\theta'}(\tau)} R(\tau)\nabla \log p_\theta(\tau)]$$

4 $$\nabla \bar{R}_\theta = \mathbb{E}_{\tau \sim p_{\theta'}(\tau)}[\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)} A^{\theta'} \nabla \log p_\theta(a_t \mid s_t)]$$

5 $$\nabla \bar{R}_\theta = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} (\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}^{\,n} - b) \nabla \log p_\theta(a_t^n \mid s_t^n)$$

$$L^{KL}(\theta) = \mathbb{E}_{\tau \sim p_{\theta'}(\tau)}[\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)} A^{\theta'}(s_t, a_t) - \beta KL(p_{\theta'}, p_\theta)]$$ 6

$$L^{CLIP}(\theta) = \mathbb{E}_{\tau \sim p_{\theta'}(\tau)}[\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)} A^{\theta'}(s_t, a_t), clip(\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)}, 1-\epsilon, 1+\epsilon) A^{\theta'}(s_t, a_t))$$ 7

Actor-Critic结构

- 提出动机：提高策略梯度方法的轨迹利用效率，并且降低方差。

$$Q^{\pi_\theta}(s_t^n, a_t^n) - V^{\pi_\theta}(s_t^n)$$

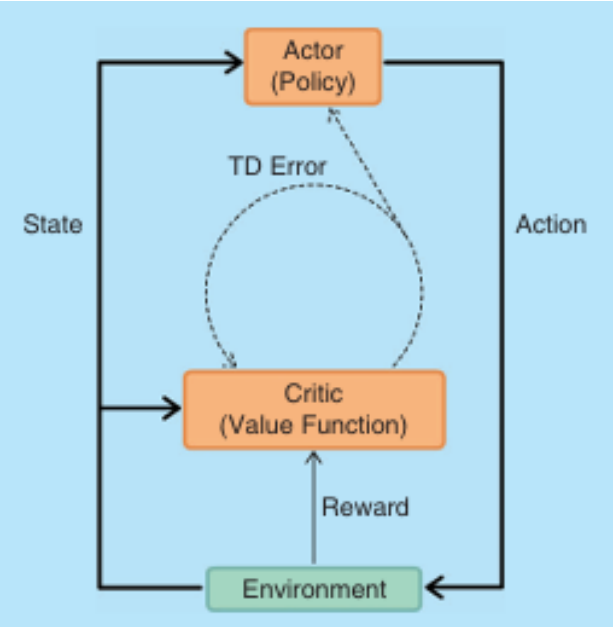
$$V^{\pi_\theta}(s_t^n)$$

基线

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\left(\sum_{t'=t}^{T_n}\gamma^{t'-t}r_{t'}^n - b\right)\nabla log p_\theta(a_t^n|s_t^n)$$

G_t^n：通过交互获取

$$E[G_t^n] = Q^{\pi_\theta}(s_t^n, a_t^n)$$

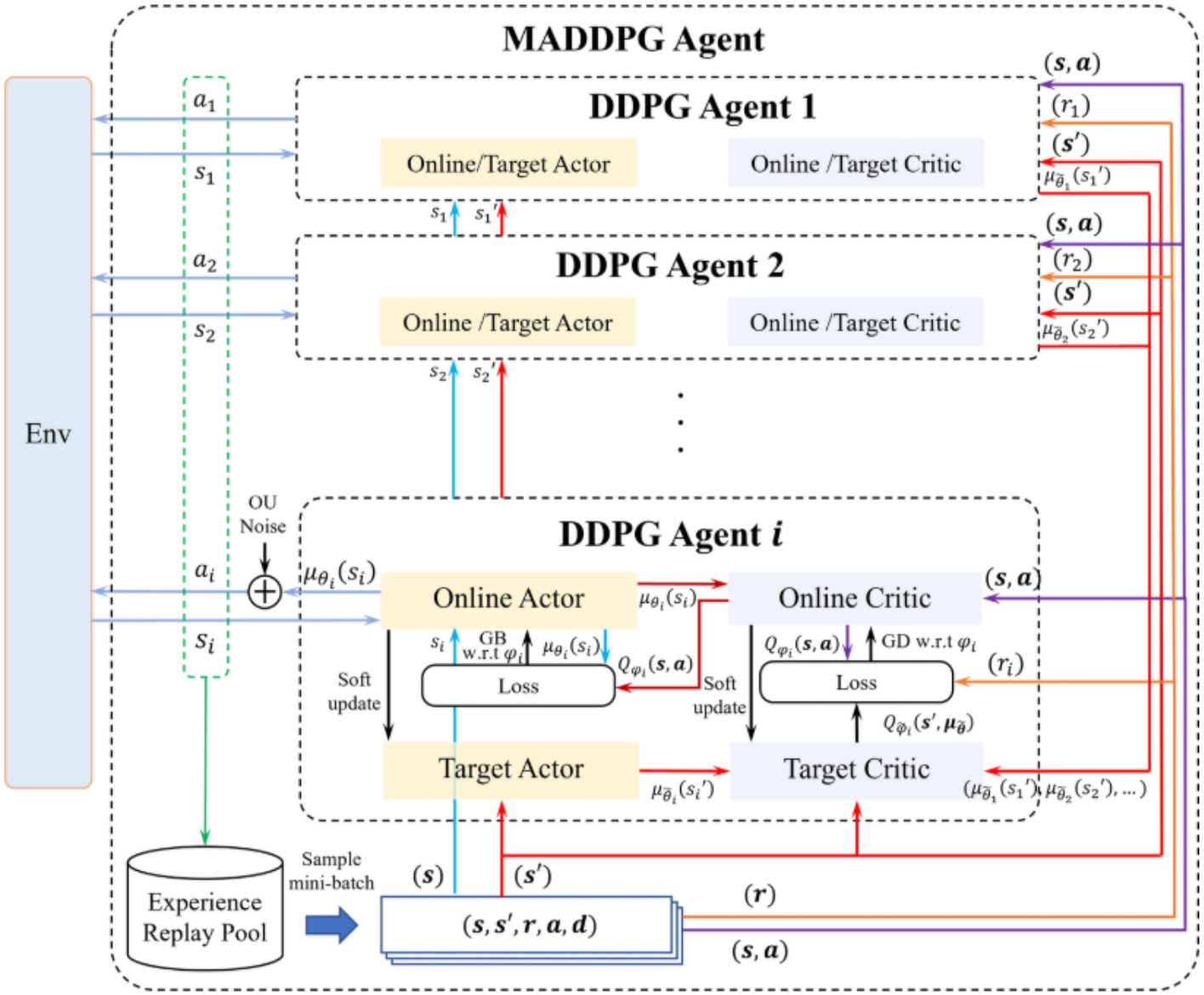


Algorithm 1 Q Actor Critic

Initialize parameters s, θ, w and learning rates α_θ, α_w; sample $a \sim \pi_\theta(a|s)$.
for $t = 1 \ldots T$: **do**
  Sample reward $r_t \sim R(s, a)$ and next state $s' \sim P(s'|s, a)$
  Then sample the next action $a' \sim \pi_\theta(a'|s')$
  Update the policy parameters: $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a)\nabla_\theta \log \pi_\theta(a|s)$; Compute the correction (TD error) for action-value at time t:
    $\delta_t = r_t + \gamma Q_w(s', a') - Q_w(s, a)$
  and use it to update the parameters of Q function:
    $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s, a)$
  Move to a $\leftarrow a'$ and s $\leftarrow s'$
end for

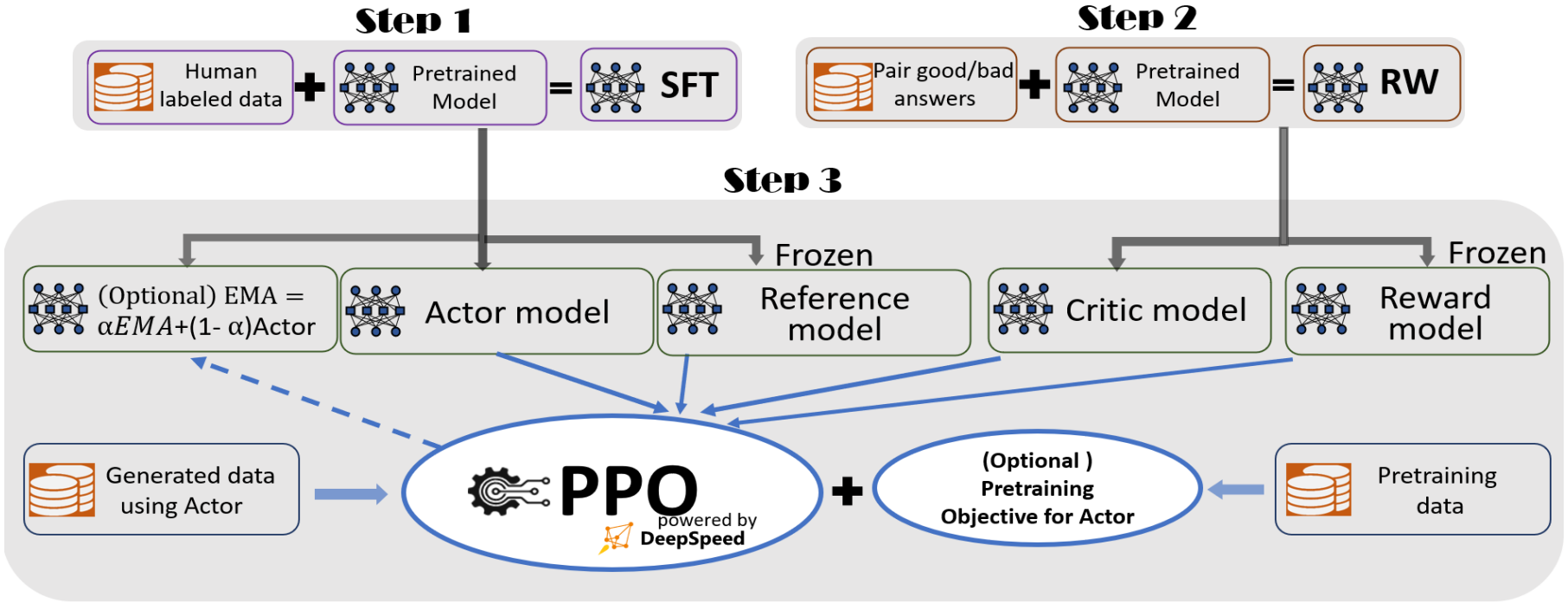
多智能体强化学习

- 对于单个Agent，模型为部分可观测马尔可夫过程
- 对于整个强化学习过程采用集中式训练，分布式执行的方式.



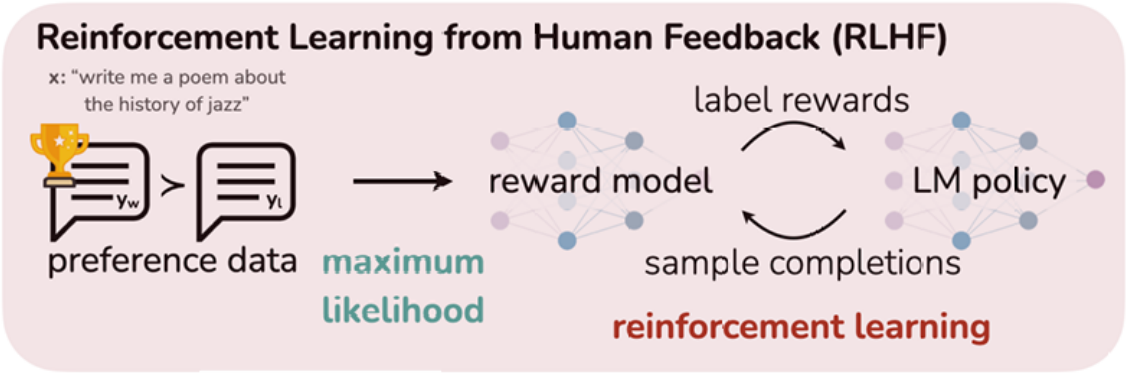
Reinforcement Learning from Human Feedback

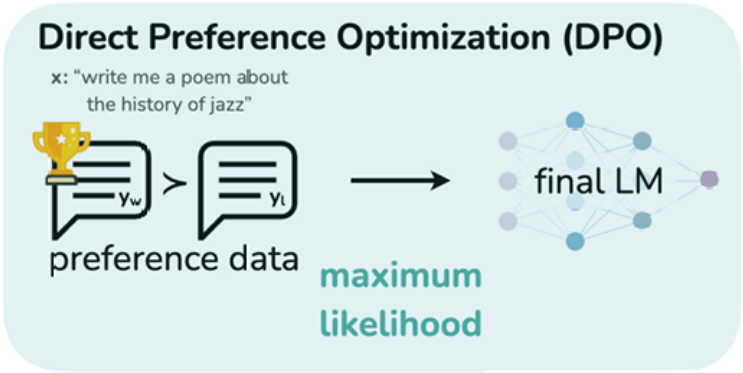
- 现有的模型通常以预测下一个单词的方式和基于简单的损失函数对模型进行建模，并不能很好地对齐人类偏好，没有显式地引入人类的偏好和主观意见，因此采用基于人类反馈的方法对大模型进行对齐。
- 对齐人类偏好：有用（遵循指令的能力）、诚实（不容易胡说八道）、安全（不容易生成不合法的，有害，有毒的信息）。



Direct Preference Optimization

- DPO

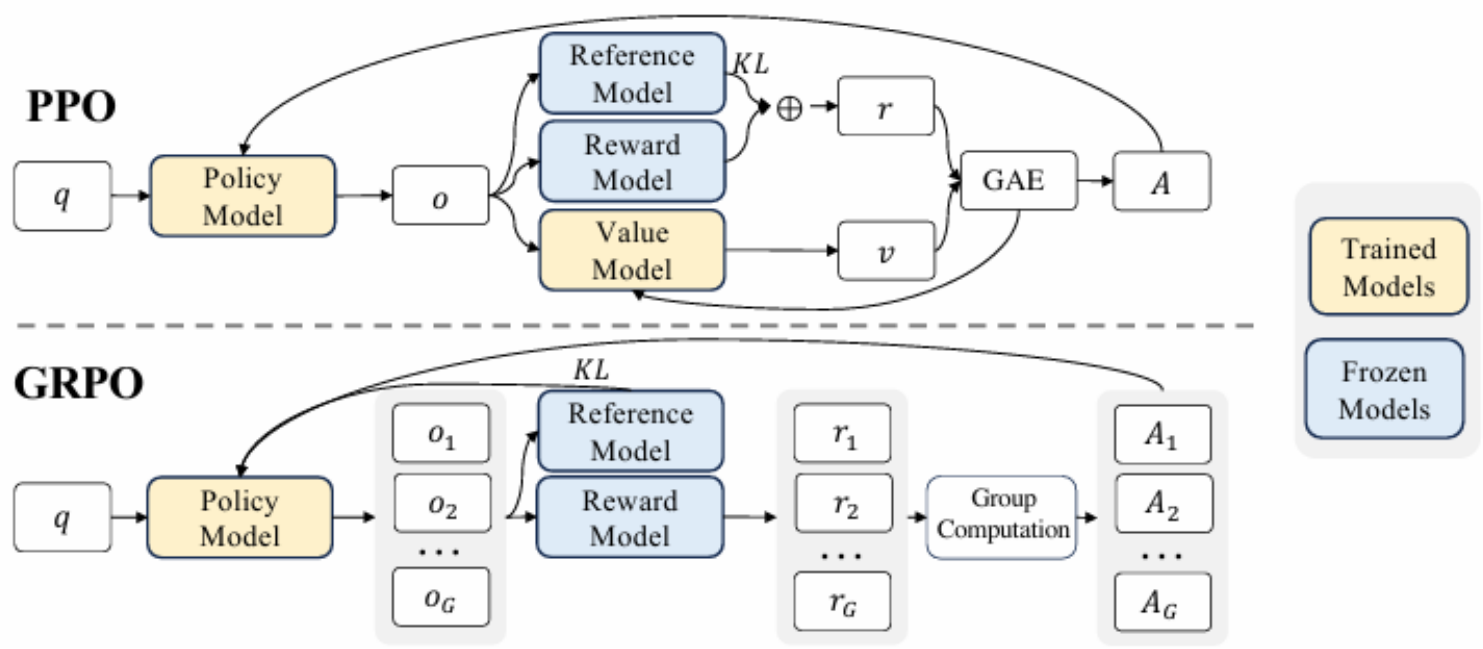




$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

Group Relative Policy Optimization

➢ GRPO



$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}$$

一些工具

- Gym/Gymnasium

  https://github.com/Farama-Foundation/Gymnasium/

- Stable-baselines3

  https://github.com/DLR-RM/stable-baselines3

- LLaMa-Factory

  https://github.com/hiyouga/LLaMA-Factory

$$
\begin{aligned}
\mathbf{E}[V(S_{t+1}) \mid S_t = s] &= \mathbf{E}\Big[\mathbf{E}[G_{t+1} \mid S_{t+1}] \mid S_t = s\Big] \\
&= \sum_{s' \in \mathcal{S}} P(S_{t+1} = s' \mid S_t = s) \cdot \mathbf{E}\Big[\mathbf{E}[G_{t+1} \mid S_{t+1}] \mid S_t = s, S_{t+1} = s'\Big] \\
&= \sum_{s' \in \mathcal{S}} P(S_{t+1} = s' \mid S_t = s) \cdot \mathbf{E}\Big[\mathbf{E}[G_{t+1} \mid S_{t+1}] \mid S_{t+1} = s'\Big] \\
&= \sum_{s' \in \mathcal{S}} P(S_{t+1} = s' \mid S_t = s) \cdot \mathbf{E}[G_{t+1} \mid S_{t+1} = s'] \\
&= \sum_{s' \in \mathcal{S}} P(S_{t+1} = s' \mid S_t = s) \cdot \mathbf{E}[G_{t+1} \mid S_{t+1} = s', S_t = s] \\
&= \mathbf{E}[G_{t+1} \mid S_t = s]
\end{aligned}
$$

$$
\begin{aligned}
\nabla \log p_\theta(\tau) &= \nabla \left(\log p(s_1) + \sum_{t=1}^{T} \log p_\theta(a_t|s_t) + \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) \right) \\
&= \nabla \log p(s_1) + \nabla \sum_{t=1}^{T} \log p_\theta(a_t|s_t) + \nabla \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) \\
&= \nabla \sum_{t=1}^{T} \log p_\theta(a_t|s_t) \\
&= \sum_{t=1}^{T} \nabla \log p_\theta(a_t|s_t)
\end{aligned}
$$