计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
2025,34(8):228−236 [doi: 10.15888/j.cnki.csa.009900] [CSTR: 32024.14.csa.009900]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于样本独特性的强化学习经验回放机制①

周梓芸, 孔　燕

(南京信息工程大学 软件学院, 南京 210044)
通信作者: 孔　燕, E-mail: kongyan4282@163.com

摘　要: 在深度强化学习领域, 特别是在高维连续的任务中, 如何高效利用有限的训练数据, 避免过拟合, 同时提高模型的泛化能力, 是一个重要的研究课题. 传统的强化学习算法通常采用单一经验池机制, 这种方法在处理高维连续状态和动作空间时, 往往面临探索效率低下和样本利用率不足的问题. 一种基于样本独特性的强化学习经验回放机制 DER (distinctive experience replay) 被提出, 该机制通过选择具有显著独特性的样本进行经验回放, DER 的核心思想是在训练过程中识别并选择具有显著独特性的样本, 将其存储在专门的独特性样本经验池中. 该机制不仅能够有效利用多样化的样本, 避免神经网路过拟合, 还能提高智能体在复杂环境中的学习效率和决策质量. 实验结果表明, DER 在经典强化学习环境中显著提高了智能体的学习效率和最终性能.

关键词: 深度强化学习; 经验回放; 样本效率; 双经验池机制

引用格式: 周梓芸,孔燕.基于样本独特性的强化学习经验回放机制.计算机系统应用,2025,34(8):228–236. http://www.c-s-a.org.cn/1003-3254/9900.html

Reinforcement Learning Experience Replay Mechanism Based on Sample Distinctiveness

ZHOU Zi-Yun, KONG Yan

(School of Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: In the field of deep reinforcement learning, particularly for high-dimensional continuous tasks, efficiently utilizing limited training data, preventing overfitting, and enhancing the model's generalization ability are crucial research challenges. Traditional reinforcement learning algorithms typically rely on a single experience replay buffer, which often faces low exploration efficiency and insufficient sample utilization, when applied to high-dimensional continuous state and action spaces. A reinforcement learning experience replay mechanism based on sample distinctiveness called distinctive experience replay (DER) is proposed. This mechanism selects samples with notable distinctiveness for experience replay. The core concept of DER is to identify and select significantly distinctive samples during training and store them in a dedicated experience pool. This mechanism not only effectively utilizes diverse samples to prevent neural network overfitting but also enhances the agent's learning efficiency and decision-making quality in complex environments. Experimental results show that DER significantly improves the agent's learning efficiency and final performance in classic reinforcement learning environments.

Key words: deep reinforcement learning; experience replay; sample efficiency; dual experience replay mechanism

在强化学习 (reinforcement learning, RL) 的领域中, 智能体必须通过与环境的交互来学习最优的决策策略. 这一过程通常涉及从经验中学习, 其中经验池 (experience replay buffer) 作为一种关键技术, 允许智能体存储并重用过去的交互数据, 以此来提高样本效率和打破样本间的相关性[1]. 然而, 在高维连续的任务中, 传统的单一

① 收稿时间: 2024-12-09; 修改时间: 2025-01-02; 采用时间: 2025-02-11; csa 在线出版时间: 2025-04-25
CNKI 网络首发时间: 2025-04-27

经验池机制往往难以应对高维连续状态[2]和动作空间带来的挑战[3], 导致探索效率低下和样本利用率不足[4].

为了解决这些问题, 研究者们一直在探索如何更高效地利用有限的训练数据, 避免过拟合, 同时提高模型的泛化能力[5]. 一种常见的方法是引入多种经验池策略, 通过将多个经验池结合在一起, 来更好地覆盖不同的状态空间和动作空间, 从而增强样本的多样性和充分性[6]. 近年来, 一些基于模型的强化学习方法也在此方面取得了显著进展. 例如, Shi 等人[7]提出了一种基于模型的算法, 结合分布鲁棒价值迭代和面对不确定性时的悲观原则设计的数据驱动惩罚项来惩罚鲁棒价值估计; Sujit 等人[8]提出一种基于从样本中学到多少来优先排序样本的方法, 优先考虑具有高可学习性的样本, 同时对那些难以学习的样本赋予较低的优先级, 这些样本通常由噪声或随机性引起; Yin 等人[9]提出了一种不确定性优先局部规划 (UFLP) 的算法框架, 利用了模拟器通常很容易将环境重置为之前观察到的状态这一特性. 此外, 策略梯度方法在大规模强化学习中是最有效的方法之一, Zhang 等人[10]考虑了经典的策略梯度方法, 这些方法在软最大化参数化和对数障碍正则化下, 通过单个轨迹或固定大小的轨迹小型批量计算近似梯度, 并结合了 REINFORCE 梯度估计过程. 在当前的研究与实践背景之下, 通过深入探索与研究, 创新性地提出了一种异策略强化学习经验回放机制, 即基于样本独特性所构建的 DER (distinctive experience replay). 该机制的核心优势在于其能够识别并优先选择那些具有显著独特性的样本, 这些样本在训练过程中被认为对智能体策略的改进尤为关键.

在 DER 里, 运用了一种创新的样本选取机制. 这一机制以样本彼此之间的独特属性为依据, 能够动态地对采样策略加以调整. 这种机制使得智能体能够持续从探索中获得最大化的学习收益, 从而在每一次迭代中都能有效地改进其行为策略. 此外, 该机制在设计上考虑了与异策略学习环境的兼容性, 使其能够在多种不同的强化学习设置中发挥作用.

为了验证 DER 的有效性, 在经典强化学习环境 MuJoCo (multi-joint dynamics with contact)[11]中进行了实验. 实验结果表明, DER 能够显著提高智能体的学习效率和最终性能, 特别是在那些复杂和动态的环境中, 所训练出的智能体展现出了更强的适应性和鲁棒性. 这些发现表明, 该机制为解决强化学习中的样本效率、过拟合和泛化能力问题提供了一种有效的解决方案. 本文的主要贡献如下.

(1) 提出 DER 的框架模型.

(2) 对 DER 机制中关键参数提出多种模式.

(3) 进行对比实验证明 DER 的优势.

1 相关工作

在强化学习领域, 所谓的“智能体”是指一种能够进行决策的机器学习实体. 与监督学习中用于进行预测的“模型”不同, 强化学习中的“智能体”不仅能够感知环境信息, 还能通过自己的决策来影响环境. 强化学习的核心目标是学习如何将环境状态与智能体的行为联系起来, 以实现最大化累积回报[12].

在强化学习的每一轮迭代中, 智能体首先观察环境的状态, 然后根据这些观察结果, 依据自身的策略选择并执行一个动作[13]. 这个动作会导致环境状态的变化, 同时环境会根据动作的效果给予智能体相应的奖励或惩罚. 智能体通过不断重复这一过程, 直到达到预设的终止条件, 完成当前回合的学习[14].

1.1 马尔可夫决策过程

几乎所有的强化学习问题都可以使用马尔可夫决策过程 (Markov decision process, MDP)[15]框架进行建模, MDP 由五元组 $\langle S,A,P,r,\gamma\rangle$构成, 其中$S$ 表示当前环境中有限状态的集合; A表示智能体在该环境中可以选择的动作集合; $P(s'|s,a)$为状态转移函数, 表示在状态 s下选择动作a后到达下一状态s' 的概率; $r(s,a)$是奖励函数, 表示在状态s 下执行动作a获得的奖励; $\gamma\in(0,1)$表示折扣因子[12]. 在 MDP 中, 在任意时间步 t 通常使用折扣回报计算累计奖励R_t, 定义如下:

$$R_t=\sum_{t=1}^{T}\gamma^{t-1}r(s_t,a_t) \tag{1}$$

1.2 经验回放机制

在众多强化学习算法中, 例如 DQN[16]、SAC 等, 经验回放是其重要的组成部分之一, 也是强化学习中的一种关键策略, 尤其在深度强化学习中扮演着重要角色. 该机制通过维护一个经验池, 存储智能体与环境交互过程中产生的状态、动作、奖励和新状态等数据, 然后从中随机抽取样本进行训练, 以打破数据间的顺序依赖, 提高学习的稳定性和效率. 经验回放机制的优势在于以下几点: 1) 提升决策能力: 有助于智能体从历史经验中学习, 避免了因连续获取的数据过于相似而

导致的学习偏差. 2) 提高数据利用率: 通过重复使用存储的经验, 可以更充分地利用有限的交互数据, 从而提高学习效率. 3) 提高泛化能力: 经验回放允许智能体从不同时间点和不同状态下学习, 有助于智能体学习到更广泛的策略, 提高其泛化能力[17]. 下面介绍两种较为经典的算法.

● 优先经验回放 (prioritized experience replay, PER): 该方法通过为存储在回放缓冲区中的经验分配优先级来提高强化学习算法的效率和性能[18]. 优先级是基于所存储的经验的时间差分误差 (TD-error) 来计算的, 即预测值和实际值之间的独特具体计算参考式 (2). TD-error 越大通常就会被赋予更高的优先级, 采样概率更大.

$$\delta_i = r + \gamma V(s') - V(s) \tag{2}$$

● 注意力经验回放 (attentive experience replay, AER)[19]: AER 根据转换中状态与智能体当前状态之间的相似性来采样转换, 使得智能体能够更加关注当前策略下重要的经验, 实现了比 PER 更高的经验利用率和算法收敛速度. 针对图片形式状态的环境和向量形式状态的环境, AER 分别设计了不同的相似性函数来计算状态之间的相似性[20].

对于高维离散状态空间如 Atari 2600 游戏, 直接计算状态之间的相似性是不可行的. 为解决这个问题, 可以使用深度卷积神经网络 (CNN) 来将原始状态嵌入到一个低维的连续特征空间中. 具体步骤如下: 使用一个预先训练好的 CNN 将原始状态 s 转换为特征向量 $x = \phi(s)$, 在特征空间中, 可以使用欧几里得距离或余弦相似性来度量两个状态的相似性. AER 中使用的是负的欧几里得距离, 计算公式如式 (3). 其中 $\phi(s)$ 是状态 s 经过 CNN 嵌入后的 512 维特征向量, $\|\phi(s_1) - \phi(s_2)\|_2$ 是两个特征向量差的欧几里得距离. 使用负的欧几里得距离作为相似性度量, 值越小表示两个状态越相似.

$$\mathcal{F}(s_1, s_2) = -\|\phi(s_1) - \phi(s_2)\|_2 \tag{3}$$

对于连续状态空间如 MuJoCo, 状态向量通常具有较低的维度, 并且可以直接进行相似性计算. 一种常用的相似性度量是余弦相似性, 计算公式如式 (4). 其中 s_1 和 s_2 是两个状态向量, $\|s\|$ 表示状态向量的欧几里得范数即向量的长度, $s_1 \cdot s_2$ 是两个向量的点积. 余弦相似性的值范围在[−1, 1]之间, 值越大表示两个状态越相似.

$$\mathcal{F}(s_1, s_2) = \frac{s_1 \cdot s_2}{\|s_1\|\|s_2\|} \tag{4}$$

1.3 SAC (soft actor-critic)

在该机制中选择了 SAC (soft actor-critic)[21]经典算法来进行前期经验池的采样以及智能体的训练. SAC 算法是一种基于最大熵的无模型深度强化学习算法, 特别适合于连续动作空间的任务. 它结合了演员-评论家 (actor-critic) 方法的优点以及最大熵强化学习框架的特点. SAC 的核心思想是最大化累积奖励的同时, 也最大化策略的熵. 通过引入熵正则项, 该算法鼓励智能体采取更加随机化的策略, 从而增强探索能力. SAC 的目标函数可以表示为式 (5), 其中 $r(s_t, a_t)$ 表示时间步 t 的即时奖励; α 是一个正则化的系数, 用来控制熵的重要程度; $\mathcal{H}(\pi(\cdot|s_t))$ 是策略 π 在状态 s_t 下的熵.

$$J(\pi) = \mathcal{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t(r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t)))\right] \tag{5}$$

SAC 算法主要由下面几部分组成.

(1) 策略网络 (policy network) 通常是一个神经网络, 用于输出给定状态下动作的概率分布. 对于连续动作空间, 通常使用高斯分布来建模动作概率, 这里使用最大化预期的 Q 值减去熵正则项来更新策略网络如式 (6).

$$L(\pi) = \mathbb{E}_{s\sim D}[\mathbb{E}_{a\sim\pi}[\alpha\log\pi(a|s) - Q(s, a)]] \tag{6}$$

(2) 价值网络 (value network) 用于估计状态的价值函数 $V(s)$, SAC 中使用了一个软价值函数 $V(s)$, 考虑了策略的熵, 使用式 (7) 的损失函数更新价值网络.

$$L(V) = \mathbb{E}_{s\sim D}\left[\frac{1}{2}(V(s) - \mathbb{E}_{a\sim\pi}[Q(s, a) - \alpha\log\pi(a|s)])^2\right] \tag{7}$$

(3) Q 值网络 (Q-network) 用于估计状态-动作对 Q 值 $Q(s, a)$, SAC 使用了两个 Q 值网络来稳定训练过程, 这里双重 Q 值网络参照了 TD3 中双重 Q 学习机制[21]. Q 值网络更新使用 Bellman 误差最小化的方式更新 Q 值网络具体损失函数如式 (8).

$$L(Q) = \mathbb{E}_{(s,a,r,s')\sim D}\left[\frac{1}{2}(Q(s, a) - (r + \gamma(V(s') - \alpha\log\pi(a'|s'))))^2\right] \tag{8}$$

(4) 目标网络 (target network) 为了稳定训练, SAC 使用了目标 Q 值网络和目标价值网络, 这些网络的参数通过指数移动平均 (EMA) 方式缓慢更新.

2 方法

本研究构建了一种强化学习经验回放机制——DER. 该机制创新之处在于其对样本差异性的识别与筛选, 它将那些具有显著特征独特的样本选入一个特别设立的缓冲区, 以实现对这些独特经验的有效利用. 这种方法不仅丰富了学习过程中的样本多样性, 还有助于防止智能体在训练中出现过度拟合, 从而增强了模型的泛化能力和适应性. 通过专注于这些独特样本的收集和再利用, 该机制为智能体在复杂和动态变化的环境中提供了更精确的学习指导, 优化了决策路径, 提高了整体性能.

2.1 DER 结构框架

DER 的主体架构展示于图 1. 此架构运用了双经验池机制, 在智能体训练起始阶段, 其运作模式与传统的单经验池算法存在一定相似性, 具体表现为采用 SAC 算法来执行前期经验池的采样操作, 同时依托该算法开展智能体的前期训练流程, 以此为后续更复杂的训练环节奠定基础. 在双经验池中, 全部经验池 *AD* 负责存储智能体训练过程中产生的所有样本经验, 而独特经验池 *SD* 则专门用于存放独特性样本经验.

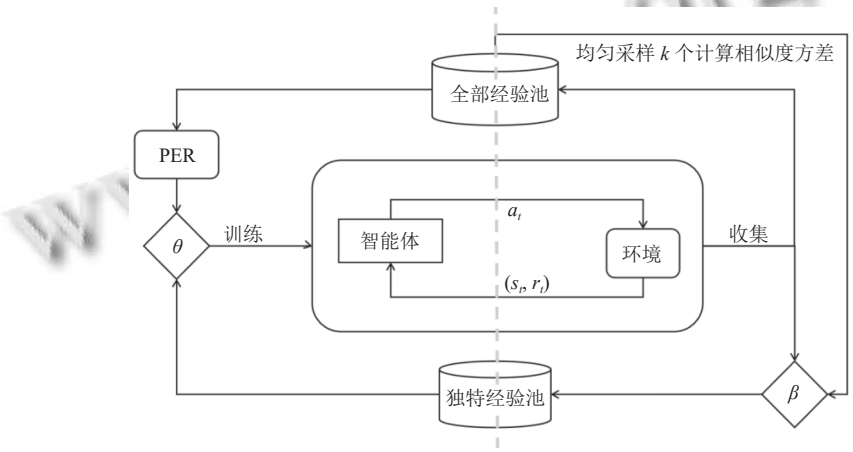

图 1 DER 框架图

在每一轮训练的迭代中, 从 *AD* 经验池中用均匀采样的方法挑选出 k 个样本. 随后, 对这些样本与当前训练阶段所获得的样本进行相似度方差的计算. 这一过程的目的是识别那些与现有策略显著不同的经验. 当某个样本的相似度方差超出了预先设定的阈值, 这表明该样本具有显著的独特性, 它随即被标记为独特性样本. 这些被认定的独特性样本因其潜在的高价值而被转移到 *SD* 经验池中, 以供后续训练过程中的进一步利用和分析. 随着训练过程的不断深入, 特别是在后期阶段, 采取了一种更加精细化的采样策略. 这时, 引入了重要性采样的方法, 它根据样本的重要性, 以一种加权的方式从 *AD* 和 *SD* 两个经验池中抽取经验数据. 这种策略确保了在训练的后期, 智能体能够更加关注那些对策略改进至关重要的经验.

这种机制有效地确保了智能体既能从丰富的历史经验中学习, 又能充分利用差异性较大的样本, 避免过拟合, 并促进学习过程的稳定性与效率.

2.2 DER 流程

综合上述内容, DER 具体步骤流程如算法 1.

算法 1. 基于样本独特性的强化学习经验回放机制 DER

输入: 小批次 k, 学习率 η, 回放周期 K, 经验池 N, 转变步 CS, 系数 λ, 衰减系数 δ, 权重调节系数 w, 总步数 TS, 相似度阈值 β.

1) 初始化: 全部经验回放池 *AD*, 独特经验回放池 *SD*, 相似度缓冲区 L
2) 观测环境初始状态 s_0, 选择动作 $a_0 \sim \pi(a|s_0;\varphi)$
3) For t=1 to CS do
4) 观测下一状态 s_t 和当前奖励 r_{t-1}
5) 将 $(s_{t-1},a_{t-1},r_{t-1},s_t)$ 存储到 *AD*
6) If $t \equiv 0 \bmod k$ then
7) For j=1 to $\lambda \cdot k$ then
8) 从 *AD* 均匀采样 (s_j,a_j,r_j,s'_j)
9) 计算相似度 $l_j=sim(s_j,s_t)$ 存放至 L
10) End for
11) 计算相似度方差 v
12) If $v>\beta$ then
13) 将 $(s_{t-1},a_{t-1},r_{t-1},s_t)$ 存储到 *SD*
14) End if
15) End if
16) 从 *AD* 中选择 k 个样本训练

17) End for
18) For t=CS to FS do
19) 重复步骤 4)–14)
20) 从 AD 和 SD 中分别选择$(k-\theta k)$和θk个样本训练
21) 选择动作$a_t \sim \pi(a|s_t;\varphi)$
22) End for

2.3 具体细节

在算法流程中, 通过相似度计算和比较实现独特样本的筛选, 参考算法 1 的第 6)–15) 行. 具体来说, 首先将当前阶段训练得到的样本与从经验池 AD 中均匀采样的 k 个样本分别计算相似度值, 然后对这k个相似度值计算方差. 接着, 将方差与设定的相似度方差阈值进行比较. 如果该方差大于阈值, 则认为当前阶段训练得到的样本具有较高的独特性, 将其存储到差异性经验池 SD 中.

对于样本相似度计算有以下几种方式, 实验中根据实验环境选择适合的计算公式.

(1) 对于高维离散状态空间 (如 Atari 2600 游戏), 可以采用式 (3) 的方法进行计算; 对于连续状态空间 (如 MuJoCo), 则可以参考式 (4). 具体的计算方式详见第 1.2 节.

(2) 在强化学习中, 时间步可能会影响样本的相关性, 可以结合时间权重来衡量相似度参考式 (9). 其中δ表示衰减系数, t表示时间步.

$$sim_{\text{time}}((s,a,t),(s',a',t')) = \mathrm{e}^{-\delta|t-t'|}\cdot\left\|s-s'\right\|_2 \tag{9}$$

(3) 奖励是强化学习中的核心信息, 样本相似度可以融合奖励来评估策略之间的效果. 基于奖励加权的相似度计算参考式 (10). 其中w是权重调节系数, r是样本的即时奖励.

$$sim_{\text{reward}}((s,a,r),(s',a',r')) = \frac{\|s-s'\|_2}{1+w|r-r'|} \tag{10}$$

(4) 在本研究中实验环境选择 MuJoCo 仿真平台, 故采取相似度计算公式是基于式 (4) 结合引入时间步相关性和融合奖励信息所产生的, 如式 (11).

$$sim_{t\&r}((s,a,r,t),(s',a',r',t')) = \mathrm{e}^{-\delta|t-t'|}\cdot\frac{\|s-s'\|_2}{1+w|r-r'|} \tag{11}$$

此外, DER 中部分参数的设置细节如下.

(1) 相似度方差阈值β: 该参数是 DER 重要的参数之一, 它的设置会影响独特性样本的选择. β可以随着时间产生变化, 因此在每一轮中β可能都是不一样的值, 对于β的设置, 本文考虑以下两种模式.

- 固定值: 在算法开始选择一个合适的值赋给β, 其中$\beta\in\mathbb{R}$是一个常数.
- 动态平均值: β更新根据式 (12):

$$\beta_j = \begin{cases} \dfrac{1}{w}\displaystyle\sum_{i=1}^{w} v_i, & j > w \\ \beta_0, & j < w \end{cases} \tag{12}$$

其中, $\beta_0\in\mathbb{R}$为β的初始值, 在算法开始时设定. $w\in\mathbb{Z}$表示每次连续选择相似度v_i的数量, 该值根据环境和需要进行设定. v_i为每轮采样计算得到的相似度, 则β取窗口内相似度的平均值.

(2) 独特性样本训练比例θ: 该参数决定后期训练在两个经验池中分别选择样本比例, 也可以随着时间推移而改变. 对于θ的设置, 本文考虑以下两种模式.

- 固定值: 在算法开始选择一个合适的值赋给θ, 其中$\theta\in\mathbb{R}$是一个常数.
- 优先值: 根据式 (13) 更新θ:

$$\theta = \frac{H_{SD}^{\alpha_p}}{H_{AD}^{\alpha_p}+H_{SD}^{\alpha_p}} \tag{13}$$

其中, H_{SD}和H_{AD}表示分别在两个经验池 SD 和 AD 中每一轮训练批次的 TD-error, $\alpha_p\in[0,1]$是一个用于调节优先级影响程度的超参数; 当α接近 0 时, 采样概率趋向于均匀分布; 当α增大时, 高优先级的样本被采样的概率会更高.

3 实验

为了评估 DER 在不同优化策略下的表现及其相对于其他强化学习算法的鲁棒性, 该研究还在 MuJoCo 这一多关节动力学仿真平台 (专门用于模拟接触动力学) 上对 DER 和 SAC 进行了对比实验. 实验结果表明, DER 在达到收敛的速度、训练过程的稳定性以及智能体在测试阶段的表现等方面, 均展现出显著的优越性.

3.1 实验环境介绍

在该研究中选择 MuJoCo 仿真平台作为实验环境如图 2 所示, 以验证 DER 的性能. MuJoCo 是一个高级物理引擎, 专为需要快速且准确模拟的领域设计, 如机器人学、生物力学、图形和动画等. 以下是实验中使用的部分环境具体介绍.

HalfCheetah-v2: 这是一个 2D 机器人环境, 模拟了

一个半猎豹机器人, 由 9 个链接和 8 个关节组成 (包括两只脚). 目标是通过施加关节扭矩, 使机器人尽可能快地向前 (向右) 移动.

Swimmer-v2: 此任务涉及一个三连杆游泳机器人, 在粘性流体中通过控制两个关节, 使其尽可能快地向前游泳.

Ant-v2: 此环境模拟了一个 3D 机器人, 由 1 个躯干和 4 条腿组成, 每条腿有两个链接. 目标是通过控制连接躯干和腿部的 8 个铰链的扭矩, 协调 4 条腿向前.

Walker2d-v2: 这是一个二维双足机器人, 目标是使机器人通过控制 6 个铰链的扭矩, 尽可能快地向前 (右) 方向移动.

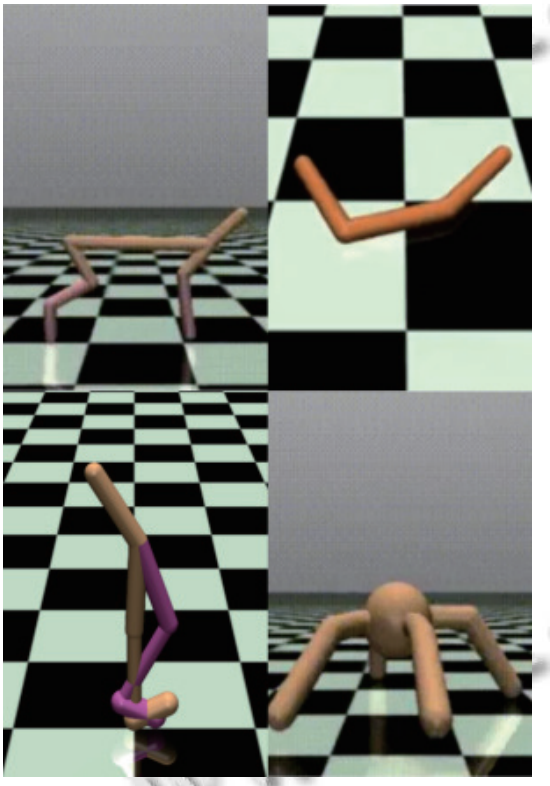

图 2 MuJoCo 环境示意

这些环境均具有连续的动作空间和复杂的动态特性, 是测试强化学习算法性能的理想选择. 通过在这些环境中进行训练和测试, 这样能够全面评估 DER 算法在不同复杂度的任务中的性能表现. 实验结果将展示 DER 算法在收敛速度、训练稳定性以及智能体在测试期间的表现上的优越性.

3.2 实验结果分析

实验研究的方法是用 SAC 验证的, 标准超参数在中被设置为默认值, 除了折扣因子$\gamma = 0.99$和 SAC 熵最大化项$\alpha = 0.1$, 经验池大小设置为 250 000, 交换步设置为 500 000.

如表 1 展示了在多个经典的 MuJoCo 环境中, SAC 算法和不同条件下 DER 的评估结果. 评估标准为在 100 万步内获得的最大奖励, 最佳结果以粗体显示.

表 1 实验结果比较

方法	Ant-v2	Walker2d-v2	Swimmer-v2	HalfCheetah-v2
SAC	3 020.8±634.1	3 609.3±519.2	48.1±2.4	7 145.2±402.1
DER	3 998.7±805.4	3 902.5±764.1	**85.6±27.1**	**8 012.8±51.4**
DER-A	3 198.6±1 856.4	**4 325.9±593.2**	65.4±3.1	7 206.0±395.4
DER-B	**4 106.8±798.0**	4 196.5±206.8	59.1±20.4	7 185.4±305.7

DER 方法为相似度方差阈值β取固定值 5 和独特性样本训练比例θ取固定值 0.5 的情况, DER-A 方法为相似度方差阈值β使用动态平均值更新和独特性样本训练比例θ取固定值 0.5 的情况, DER-B 方法为相似度方差阈值β取固定值 5 和独特性样本训练比例θ使用优先值方法更新的情况.

如图 3 分别展示了在 4 个不同的 MuJoCo 环境中, 几种强化学习算法的性能比较. 每个子图的横轴代表总时间步数 (以百万为单位), 纵轴代表奖励值.

图 3 中比较了 SAC、DER 以及两种变体 DER-A 和 DER-B 的性能. 在 Ant 环境中, 所有算法的奖励值随着时间步数的增加而提高, 其中 DER-B 在后期表现最佳, 奖励值接近 5 000. SAC 和 DER-A 的表现相近, 而 DER 在中期表现突出, 但最终略低于 DER-B. 在 Walker2d 环境中, DER-A 和 DER-B 在奖励值上领先, 且两者表现非常接近, 最终奖励值超过 4 000. SAC 和 DER 的表现相似, 但表现略低于 DER-A 和 DER-B. Swimmer 环境中, DER-B 同样表现最佳, 奖励值在后期稳定在 100 左右. SAC 和 DER 的表现相近, 但 DER 在中期有一段明显的提升. DER-A 的表现最不稳定, 奖励值波动较大. 在 HalfCheetah 环境中, DER-B 再次展现出最佳性能, 奖励值最终超过 8 000. SAC 和 DER 的表现相似, 但 DER 在中期有一段显著的提升, 最终略低于 DER-B. DER-A 的表现与 SAC 和 DER 相近, 但略低.

总体来看, DER-B 算法在所有环境中都展现出了优异的性能, 尤其是在 Ant 和 HalfCheetah 环境中. 这表明 DER-B 算法在处理不同类型强化学习任务时具有较好的适应性和稳定性. 图 3 中的阴影区域表示奖励值的方差, DER-B 的方差在大多数情况下较小, 进一步证明了其稳定性.

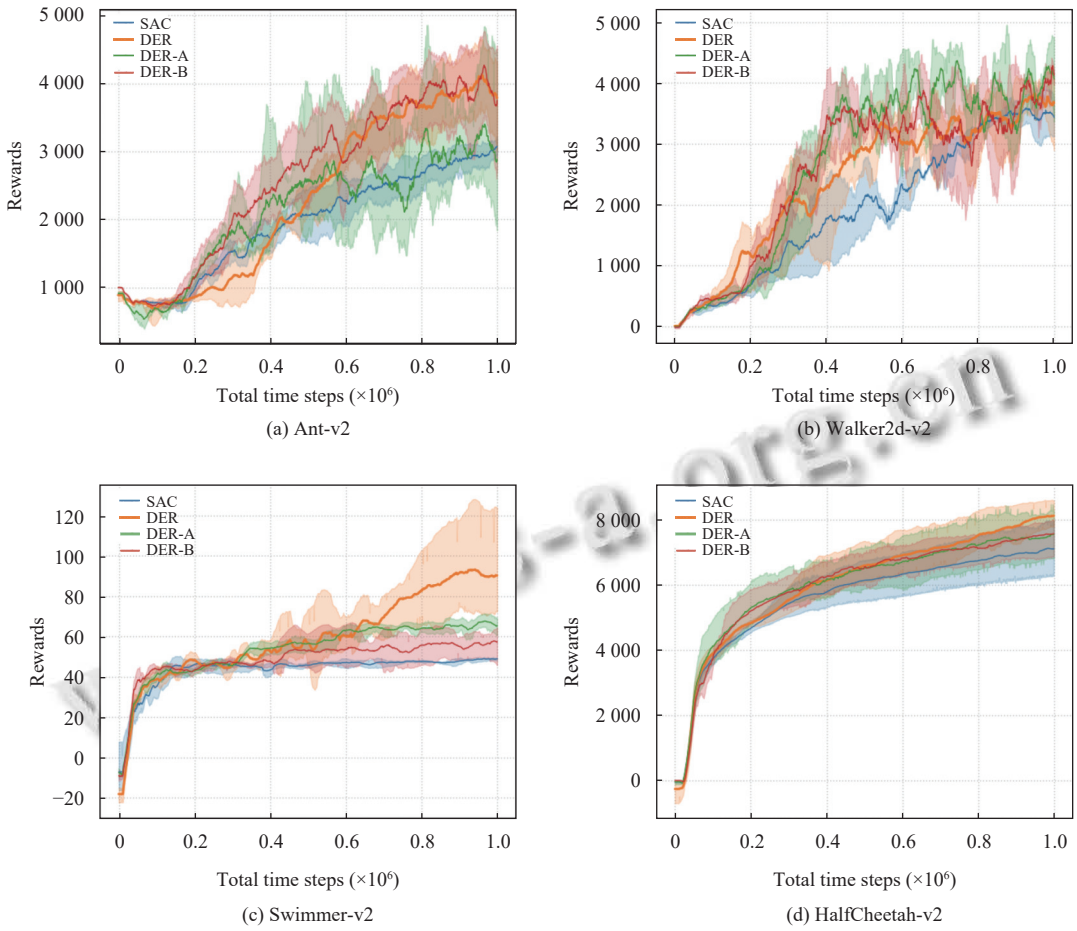

图 3 SAC、DER 以及两种变体 DER-A 和 DER-B 的实验结果对比图

3.3 消融实验

图 4 全面地展示了在 4 个多样化的 MuJoCo 仿真环境中进行的消融实验结果, 旨在深入比较 SAC 算法与集成了 DER 模块的 SAC 算法的性能独特. 消融实验作为一种精细的科研方法, 通过有意识地移除或调整算法的关键组件, 能够准确评估这些组件对算法整体性能的具体影响.

这些实验将传统的 SAC 算法与融入 DER 模块的 SAC 算法进行了直接对比. 通过这种对比, 不仅能够量化 DER 模块对性能提升的贡献, 而且能够进一步验证 DER 算法在强化学习任务中的实际效用和优越性. 实验结果在 4 个 MuJoCo 环境中均显示, 融入 DER 模块的 SAC 算法在学习和适应效率上均超越了标准 SAC 算法. 在训练的早期阶段, DER 算法便展现出更快的奖励增长速度, 并在训练的中后期持续保持领先, 这表明 DER 算法在处理复杂动力学和精细控制任务时具有显著优势.

4 结论与展望

本文提出一种强化学习经验回放机制——DER. 其目的在于精准识别并优先运用独特性显著的样本, 以此提升学习效率, 优化智能体性能. DER 机制运行的关键之处, 在于可从经验池中动态拣选对当前策略改进作用最大的样本, 这些样本凭借在状态空间里的代表性与差异性获选. 而且, DER 持续引入全新独特样本, 促使智能体拓展探索策略空间, 进而增强模型泛化能力.

展望后续工作, 该机制投入实际应用时, 尚需进一步实验验证与精细调优. 比如, 针对相似度方差阈值, 可借助神经网络训练, 以获取更适配的值; 至于其他如何精准识别、科学评估样本独特性, 以及怎样平衡样本多样性与学习效率二者关系, 皆是 DER 机制在实际运用中大概率要应对的难题. 不仅如此, DER 的具体实现细节、性能呈现, 同样需要在各类强化学习任务与环境下深入测试、细致分析.

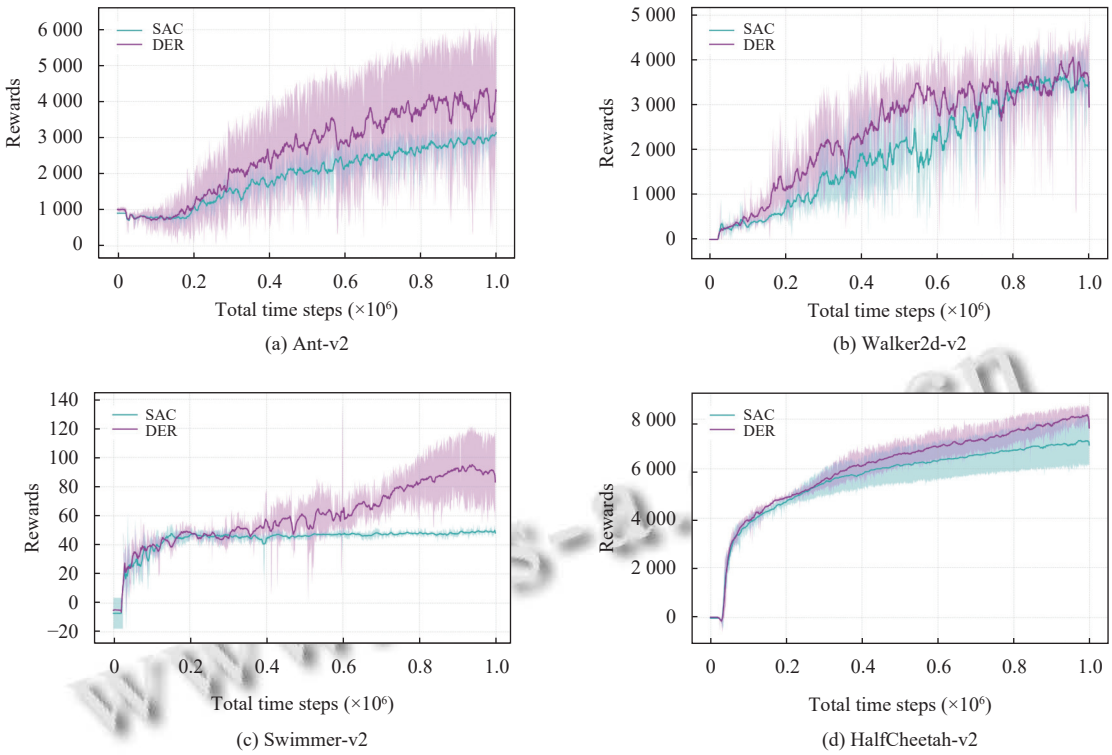

图 4 消融实验结果对比

参考文献

1 Fujimoto S, Meger D, Precup D. An equivalence between loss functions and non-uniform sampling in experience replay. Proceedings of the 34th Annual Conference on Neural Information Processing Systems. 2020. 14219–14230.

2 Wang ZH, Wang J, Zhou Q, *et al*. Sample-efficient reinforcement learning via conservative model-based actor-critic. Proceedings of the 36th AAAI Conference on Artificial Intelligence. AAAI, 2022. 8612–8620.

3 Mnih V, Kavukcuoglu K, Silver D, *et al*. Playing Atari with deep reinforcement learning. arXiv:1312.5602, 2013.

4 Levine S, Kumar A, Tucker G, *et al*. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv:2005.01643, 2020.

5 张峻伟, 吕帅, 张正昊, 等. 基于样本效率优化的深度强化学习方法综述. 软件学报, 2022, 33(11): 4217–4238. [doi: 10.13328/j.cnki.jos.006391]

6 Hao JY, Yang TP, Tang HY, *et al*. Exploration in deep reinforcement learning: From single-agent to multiagent domain. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(7): 8762–8782. [doi: 10.1109/TNNLS.2023.3236361]

7 Shi LX, Chi YJ. Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. Journal of Machine Learning Research, 2024, 25(200): 1–91.

8 Sujit S, Nath S, Braga PHM, *et al*. Prioritizing samples in reinforcement learning with reducible loss. Proceedings of the 37th International Conference on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2023. 1008.

9 Yin D, Thiagarajan S, Lazic N, *et al*. Sample efficient deep reinforcement learning via local planning. arXiv:2301.12579, 2023.

10 Zhang JZ, Kim J, O'Donoghue B, *et al*. Sample efficient reinforcement learning with REINFORCE. Proceedings of the 35th AAAI Conference on Artificial Intelligence. 2021. 10887–10895.

11 Todorov E, Erez T, Tassa Y. MuJoCo: A physics engine for model-based control. Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. Vilamoura-Algarve: IEEE, 2012. 5026–5033.

12 Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018.

13 Silver D, Huang A, Maddison CJ, *et al*. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587): 484–489. [doi: 10.1038/nature16961]

14 Arulkumaran K, Deisenroth MP, Brundage M, *et al*. Deep reinforcement learning: A brief survey. IEEE Signal

Processing Magazine, 2017, 34(6): 26–38. [doi: 10.1109/MSP.2017.2743240]

15 Bellman R, Kalaba R. Dynamic programming and statistical communication theory. Proceedings of the National Academy of Sciences of the United States of America, 1957, 43(8): 749–751.

16 Mnih V, Kavukcuoglu K, Silver D, *et al*. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533. [doi: 10.1038/nature14236]

17 Mnih V, Badia AP, Mirza M, *et al*. Asynchronous methods for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning. New York: JMLR.org, 2016. 1928–1937.

18 Schaul T, Quan J, Antonoglou I, *et al*. Prioritized experience replay. Proceedings of the 4th International Conference on Learning Representations. San Juan: OpenReview.net, 2016.

19 Sun PQ, Zhou WG, Li HQ. Attentive experience replay. Proceedings of the 34th AAAI Conference on Artificial Intelligence. New York: AAAI, 2020. 5900–5907.

20 Burda Y, Edwards H, Pathak D, *et al*. Large-scale study of curiosity-driven learning. Proceedings of the 7th International Conference on Learning Representations. New Orleans: OpenReview.net, 2019. 1–17.

21 Haarnoja T, Zhou A, Abbeel P, *et al*. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proceedings of the 35th International Conference on Machine Learning. Stockholm: PMLR, 2018. 1861–1870.

(校对责编: 张重毅)