基于策略蒸馏主仆框架的优势加权双行动者-评论家算法

杨皓麟, 刘全

引用本文

杨皓麟, 刘全. 基于策略蒸馏主仆框架的优势加权双行动者-评论家算法[J]. 计算机科学, 2024, 51(11): 81-94.

YANG Haolin, LIU Quan. Advantage Weighted Double Actors-Critics Algorithm Based on Key-Minor Architecture for Policy Distillation [J]. Computer Science, 2024, 51(11): 81-94.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000170

基于策略蒸馏主仆框架的优势加权双行动者-评论家算法

杨皓麟[1]　刘　全[1,2]
1 苏州大学计算机科学与技术学院　江苏 苏州 215006
2 苏州大学江苏省计算机信息处理技术重点实验室　江苏 苏州 215006
(20215227121@stu.suda.edu.cn)

摘　要　离线强化学习(Offline RL)定义了从固定批次的数据集中学习的任务,能够规避与环境交互的风险,提高学习的效率与稳定性。其中优势加权行动者-评论家算法提出了一种将样本高效动态规划与最大似然策略更新相结合的方法,在利用大量离线数据的同时,快速执行在线精细化策略的调整。但是该算法使用随机经验回放机制,同时行动者-评论家模型只采用一套行动者,数据采样与回放不平衡。针对以上问题,提出一种基于策略蒸馏并进行数据经验优选回放的优势加权双行动者-评论家算法(Advantage Weighted Double Actors-Critics Based on Policy Distillation with Data Experience Optimization and Replay,DOR-PDAWAC),该算法采用偏好新经验并重复回放新旧经验的机制,利用双行动者增加探索,并运用基于策略蒸馏的主从框架,将行动者分为主行为者和从行为者,提升协作效率。将所提算法应用到通用 D4RL 数据集中的 MuJoCo 任务上进行消融实验与对比实验,结果表明,其学习效率等均获得了更优的表现。

关键词:离线强化学习;深度强化学习;策略蒸馏;双行动者-评论家框架;经验回放机制
中图分类号　TP181

Advantage Weighted Double Actors-Critics Algorithm Based on Key-Minor Architecture for Policy Distillation

YANG Haolin[1] and LIU Quan[1,2]
1 School of Computer and Technology,Soochow University,Suzhou,Jiangsu 215006,China
2 Provincial Key Laboratory for Computer Information Processing Technology,Soochow University,Suzhou,Jiangsu 215006,China

Abstract　Offline reinforcement learning(Offline RL) defines the task of learning from a fixed batch of dataset,which can avoid the risk of interacting with environment and improve the efficiency and stability of learning. Advantage weighted actor-critic algorithm,which combines sample efficient dynamic programming with maximum likelihood strategy updating,makes use of a large number of offline data and quickly performs online fine-grained strategy adjustment. However,the algorithm uses a random experience replay mechanism,while the actor-critic model only uses one set of actors,and data sampling and playback are unbalanced. In view of the above problems,an advantage weighted double actors-critics algorithm based on policy distillation with data experience optimization and replay is proposed(DOR-PDAWAC),which adopts the mechanism of preferring new data and replaying old and new data repeatedly,uses double actors to increase exploration,and uses key-minor architecture for policy distillation to divide actors into key actor and minor actor to improve performance and efficiency. Applying algorithm to the MuJoCo task in the general D4RL dataset,and experimental results show that the proposed algorithm achieves better performance in terms of learning efficiency and other aspect.

Keywords　Offline reinforcement learning,Deep reinforcement learning,Policy distillation,Double actors-critics framework,Experience replay mechanism

到稿日期:2023-10-24　返修日期:2024-03-07
基金项目:国家自然科学基金(62376179,61772355,61702055,61876217,62176175);新疆维吾尔自治区自然科学基金(2022D01A238);江苏高校优势学科建设工程资助项目
This work was supported by the National Natural Science Foundation of China(62376179,61772355,61702055,61876217,62176175),Natural Science Foundation of Xinjiang Uygur Autonomous Region, China(2022D01A238) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions(PAPD).
通信作者:刘全(quanliu@suda.edu.cn)

1 引言

深度强化学习(Deep Reinforcement Learning,DRL)采用马尔可夫决策过程(Markov Decision Processes,MDPs),应用于序贯决策任务,能够在依赖非常少的先验知识的情况下,学习得到复杂的非线性策略[1]。强化学习具有强大的决策能力,而深度学习具备强大的感知能力,深度强化学习将强化学习(Reinforcement Learning,RL)与深度学习(Deep Learning,DL)相结合,在视频游戏[2]、机器人控制[3]、自动驾驶[4]、文本分类[5]、分子设计[6]等多个领域中取得了显著成果。

若根据不同的经验获取与学习的过程,强化学习算法可以分为两大类:在线强化学习(Online reinforcement learning,Online RL)与离线强化学习(Offline Reinforcement Learning,Offline RL)[7]。在线强化学习指在学习的过程中,智能体需要和环境进行交互获得其反馈的奖励等调整策略。离线强化学习指在学习过程中,智能体不需要与环境交互,直接从数据集中采集样本经验进行学习[8]。现实生活中,很多设想的在线交互是不切实际的或是成本昂贵的[7]。对于深度强化学习,若能够利用先前收集的离线数据驱动强化学习,不需要额外的学习与环境交互[9]就可以提高算法的可实用性和效率。

在离线强化学习中常常会利用批处理强化学习(Batch reinforcement learning)算法[10]。在该算法中,如果数据集与当前策略下的真实分布不相关,那么这些算法可能在批处理设置中失败。引发这种现象的问题被定义为外推误差(Extrapolation error)[10]。为了解决该问题,Fujimoto 等提出了批约束的深度 Q 学习(Batch-Constrained Deep Q-learning,BCQ)算法[10],最小化策略的状态-动作访问和批中包含的状态-动作对之间的不匹配。

随着离线强化学习的发展,更多具有代表性的离线强化学习算法被提出,例如引导误差累积减少原理(Bootstrapping Error Accumulation Reduction,BEAR)算法[11]、保守式 Q 学习(Conservative Q-Learning,CQL)算法[12]、结合行为克隆的双延迟深度确定性策略梯度(Twin Delayed Deep Deterministic Policy Gradient Algorithm Combined with Behavior Cloning,TD3+BC)算法[13]、优势加权行动者评论家(Advantage Weighted Actor-Critic,AWAC)算法[14]等。其中 AWAC 算法提出了一种新的深度强化学习使用模式,系统地研究了一种离线预训练和在线微调相结合的新模式,使得深度强化学习模型能够像 BERT 算法一样在预训练后进行微调[14],取得了较好的效果。

AWAC 算法采用经验回放机制。该机制中,最传统的是随机回放经验(Random Experience Replay)方式[15],但此种方式在不同样本的重要程度方面考虑欠缺。为了解决此问题,最常用的方法是基于 TD 误差(Time Difference Error,TD-error)的优先经验回放。该方法需要对每个经验的优先级进行计算后借助求和树结构进行存储,这导致其在经验回放缓冲池大小为 N 时,需要额外的时间复杂度 $O(\log N)$ 去计算优先级。并且经过该机制,经验采样会偏好于特定状态,这样会使智能体具有倾向性,算法探索性较差,其计算得出的结果也会由于自举产生较大偏差[16]。在离线强化学习中使用 TD 误差的优先经验回放机制,智能体同样会具有倾向性,算法探索性较差[17]。

文献[18]中根据经验生成的时间将其划分为新旧两种,其指出,在传统随机经验回放中,旧经验数据得到采样的次数比新经验更多。在离线学习中,上一次采样的时间距离当前时间较远的经验对当前策略贡献更大,偏差更小。将上一次采样时间距离当下较远的经验称为“新经验”,并对这些新经验进行偏好优选,可以使得值评估更加准确。文献[19]提出了分类经验回放方法,并分别基于时间误差以及立即奖赏对样本分类。在离线学习下,这种方法会忽视部分最高立即奖赏相对较低的状态,使得算法泛化程度受限。针对以上问题,本文提出一种新的分类经验回放策略,对新经验进行优先选择。通过分类回放而非计算优先级的方式,有效降低经验计算复杂度。通过对新经验优选回放,使得智能体尽可能全面覆盖学习各种状态,提升探索程度。将其应用于 AWAC 算法,提出数据采样优选的 AWAC(Advantage Weighted Double Actors-Critics Based on Data Experience Optimization and Replay,DOR-AWAC)算法。

此外,AWAC 算法采用行动者-评论家(Actor-Critic)框架[20],但对于两套评论家(Critic),只用一套行动者(Actor),该行动者仅针对第一套评论家进行改进,评论家采取单步自举方法评估行动者的性能,未考虑情节回报等信息,策略改进方向存在偏差。针对此问题,本文提出双行动者-评论家模型,使得两套行动者与评论家各自映射,评论家训练过程中分别用各套行动者自举,行动者改进过程中利用各套评论家,由此提升评论家的利用率。

在额外引入一套行动者进行学习的同时,本文采取策略蒸馏(Policy Distillation,PD)[21]方式使两套行动者相互协作。将其运用到主仆框架(Key-Minor Architecture),在训练的不同阶段,基于情节回报变化区分主行动者以及从行动者,进一步提升策略协作的效率。本文将运用主仆框架的策略协作(Key-Minor Architecture for Policy Association,KMAA)机制应用于 AWAC 算法,提出基于策略蒸馏的主仆框架的优势加权双行动者-评论家(Advantage Weighted Double Actors-Critics Based on Key-Minor Architecture for Policy Association with Policy Distillation,PDAWAC)算法。

本文的主要贡献可以总结为以下 4 点:

1)提出了一种对数据集中经验优选回放策略,应用于 AWAC 算法,并提出 DOR-AWAC。

2)提出双行动者-评论家模型,两个行动者与评论家一一映射,提升算法探索能力。

3)在利用上述模型的同时,提出一种基于策略蒸馏的主仆框架,并将行动者分为主次两种。将该框架应用于 AWAC 算法,提出 PDAWAC。

4)将主仆框架的双行动者-评论家模型与数据集优选回放策略相结合,提出 DOR-PDAWAC 算法。将其应用于通用的 D4RL 数据集中,结果表明该方法能够提升算法效率。

2 相关工作

2.1 强化学习

强化学习的目标是通过智能体与环境的交互，训练它学习到能够获得累计最多奖赏的策略[1]。其中交互指智能体在未知环境中行走的过程，在每一时间步 t，环境为智能体提供一个状态 S_{t+1}，在此基础上智能体根据策略选择一个动作 a_t 执行，在下一时间步 $t+1$ 获得来自环境反馈的奖赏 r_t 和下一时刻状态 S_{t+1}。此交互模型采用马尔可夫决策过程[1]建模，包含五元组$(\mathcal{S},\mathcal{A},\mathcal{R},P,\gamma)$，其中 $\mathcal{S}$ 表示状态集合，$\mathcal{A}$ 表示动作集合，$\mathcal{R}$ 表示奖赏函数，P 表示状态转移函数，γ 表示折扣因子。智能体从初始状态 S_0 开始，不断采用策略 $\pi:\mathcal{S}\rightarrow\mathcal{A}$ 与环境交互。假设 t 时刻智能体位于状态 S_t，运用以上策略采取动作 A_t，获取得到下一时刻的奖赏 R_{t+1}，并转移到下一状态 S_{t+1}，如此反复，直至在 T 时刻终止，则状态 S_t 下智能体获得的累积折扣奖赏可以定义为：

$$r_tG_t=\sum_{t=t'}^{T}\gamma^{t'-t}r_{t'} \tag{1}$$

其中，智能体的目标是找到一个最优策略 π，在该策略指导下可以获得最大的累计折扣奖赏。对于每个状态动作的得分，可以用状态-动作值函数 $q_\pi(s,a)$来衡量，其定义为：

$$q_\pi(s,a)=\mathbb{E}_\pi[G_t\mid S_t=s,A_t=a] \tag{2}$$

式(2)表示智能体在策略 π 的指导下，在状态 S_t 执行动作 A_t 所能获得的累计期望奖赏。在强化学习中，通常用状态-动作值函数 $q_\pi(s,a)$的大小来评估当前动作的好坏[1]，以更好地指导智能体完成指定任务。

在强化学习问题中，对于任意状态-动作对(s,a)，动作值函数遵循贝尔曼方程[1]：

$$q_\pi(s,a)=\mathbb{E}_{r\sim P}[r+\gamma E_{s'\sim\rho'\sim\pi}[q_\pi(s',a')]] \tag{3}$$

基于值函数的强化学习方法以式(3)为基础，而策略梯度方法则给出了另一种目标函数来解决强化学习问题[1]。该目标函数定义为关于策略的期望累积奖赏，用于对策略性能进行衡量，满足：

$$J(\theta)=\mathbb{E}_\pi[\sum_{t=0}^{\infty}\gamma^tR_{t+1}] \tag{4}$$

其中，θ 表示策略参数。

策略梯度算法通过梯度上升的方式对策略进行更新，使得策略性能逐步提升。其更新式如式(5)所示：

$$\theta_{t+1}=\theta_t+\alpha\nabla_\theta J \tag{5}$$

2.2 离线强化学习

离线强化学习问题可以定义为静态数据集驱动形式的强化学习问题，在不与环境交互的情况下，让目标最大化[7]。其最终的目标仍然是优化式(4)中的目标。但是，智能体不再使用行为策略与环境交互并收集额外的转换，而是提供一个静态转换数据集 $\mathcal{D}=\{(s_t^i,a_t^i,s_{t+1}^i,r_t^i)\}$，并且必须学习使用这个数据集的最佳策略[10]。在离线强化学习算法中将使用 π_β 来表示数据集中状态和动作的分布，状态动作元组$(s,a)\in\mathcal{D}$ 根据 $s\sim d^{\pi_\beta}(s)$采样，同时根据行为策略对动作进行采样，例如 $a\sim\pi_\beta(a\mid s)$[10]。

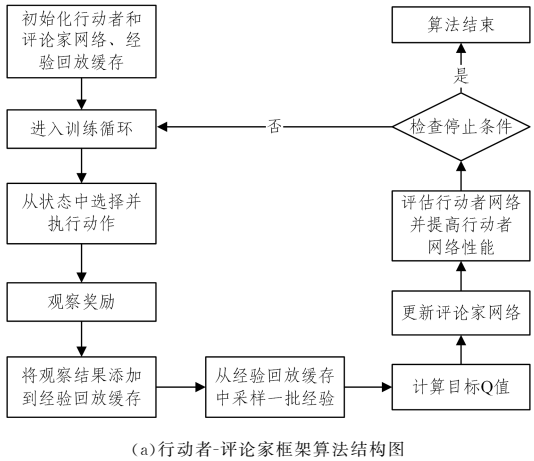
理论上，任何异策略的强化学习算法都可以作为离线强化学习算法使用[22]。大多数异策略深度强化学习算法都属于增长型批量学习(Growing Batch Learning)，其中数据被收集并存储到经验回放数据集(Experience Replay Datasets)中，用于在进一步收集数据之前训练智能体。然而，如果数据集与当前策略下的真实分布不相关，实验结果会很差。这种现象是由外推误差导致的，可归因于策略中的数据分布与批处理中的不匹配(Out of Distribution，OOD)[10]。未包含的动作的策略可能无法学习价值函数，因此不能简单地将在线强化学习算法直接应用于离线强化学习中。

2.3 行动者-评论家框架

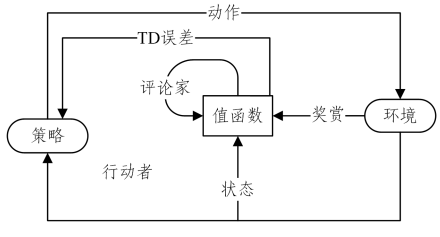
行动者-评论家(Actor-Critic)是强化学习中广泛使用的一种算法框架，结合了值函数(Critic)和策略函数(Actor)的学习[20]。其解决了基于值函数方法难以处理高维动作空间的问题，并且缓解了基于策略梯度方法训练方差较大导致算法不稳定的问题。

该算法框架包含两个部分：行动者以及评论家。在算法中，行动者是负责决策行动的策略，智能体根据行动者提供的信息与环境进行交互。评论家使用值函数方法评估策略并反馈评估结果，估计行动者采取该策略的价值。行动者根据结果采用策略梯度方式更新[23]。

行动者-评论家框架如图 1 所示，二者相互协作，实现特定任务的最优策略的学习。其可以解决连续动作空间中的控制问题，并且能够学习高维状态空间中的最佳策略。



(a)行动者-评论家框架算法结构图



(b)行动者-评论家框架算法示意图

图 1 行动者-评论家框架

Fig. 1 Schematic diagram of actor-critic framework algorithm

2.4 策略蒸馏

策略蒸馏由 Colmenarejo 等[21]提出，最初被应用于多任务强化学习中。策略蒸馏指每个任务探索完毕后，将训练效果

较佳的映射的策略的知识蒸馏提炼到广泛的策略模型中，在推广至其他任务时也会有较高的奖励。自策略蒸馏被提出后，出现很多以它为原型的研究，其在多智能体强化学习算法[24]、在线强化学习算法以及其他领域均成果显著。

策略蒸馏方法包含教师模型与学生模型，其中教师模型尤为重要[21]。如果教师模型无法依赖，则学生模型无法取得较好的学习效果。针对以上问题，Lai 等[25]提出了解决方案，该方案仅利用双线学生模型，被称为双策略蒸馏。随后将该方法应用于深度确定性策略梯度(DDPG)算法中，获得了较好的结果。但是在双线学生模型中，不同的训练时刻双模型间可能存在性能区别，导致效果不佳。针对此问题，Nagarajan 等[26]将知识蒸馏运用其中，提出了周期性知识蒸馏。在该方法中，同时进行多个策略的运算，在每个回合中随机采样一套策略使智能体进行具体交互，同时进行知识蒸馏，提取较佳的评论家知识。

本文将策略蒸馏模块与动态选主模块相结合，应用于主从行动者选择模块，实现对主行动者与从行动者的选择，从而实现更好的调度以及更高的算法效率。

2.5 经验回放机制

经验回放机制被应用于许多异策略在线强化学习算法中。传统的随机经验回放机制未考虑不同样本的重要性，效率较低。在 DQN 算法[22]中，根据 TD-error 对不同样本计算不同的优先级权重，对权重较高的经验进行优先回放。文献[18]从时间层面考虑，鼓励回放最近经验。文献[19]中分别对 TD-error 以及立即奖赏采用分类经验回放。文献[27]对传统 Q 学习进行研究，发现提升回放容量以及降低先前经验设定可以提升算法性能，并指出使用未修正 n-步回报算法表现更佳。文献[28]将经验回放应用于终身学习，提出了几种经验选择策略，并表明适当的经验选择方法可以避免灾难性遗忘。上述方法预定义了经验回放的规则，有一定的应用局限性。为了缓解上述问题，文献[29]采用元学习的思想，学习神经网络来选择经验进行回放。

类比在线强化学习算法，本文提出了一种数据回放策略，根据经验回放池中经验数据采用次数，分类进行优选。与基于优先级的经验回放方法相比，本文基于最大置信度上界，回放策略时间复杂度更低，算法相对更稳定。本文方法可以在鼓励智能体探索的同时实现探索与探索之间的权衡和利用。

2.6 优势加权行动者-评论家算法

2.6.1 AWAC 算法思想

离线数据集的先验数据可以有多种来源，由此 Nair 等提出了一种不以任何特权方式利用不同类型数据的算法——优势加权行动者-评论家(Advantage Weighted Actor Critic，AWAC)算法[14]。虽然完全离线的强化学习方法提供了一种利用离线数据的机制[7,10]，但此类方法对于在线数据的微调通常无效，因为它们通常过于保守[30]。AWAC 算法的主要思想是通过对先验数据集 D 进行离线学习来学习策略，然后通过在线交互进行微调[14]。首先，离线强化学习方法在训练开始时向算法提供静态转换数据集 $D=\{(s,a,s',r)_j\}$。利用 D 进行预训练，并使用少量在线交互来学习最优策略 $\pi^*(a|s)$，如图 2 所示。此设置代表了许多现实世界的强化学习设置，其中可以使用先前的数据，目的是有效地学习新技能，以便在收集有限数量的交互数据后达到专家级的表现[14]。

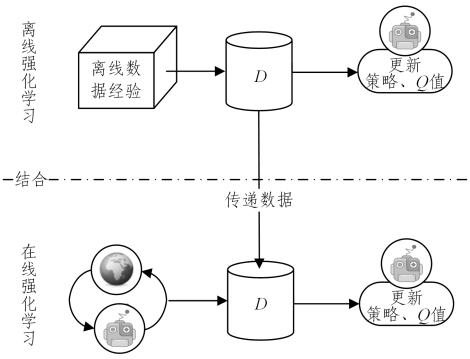


图 2 AWAC 算法思想示意图

Fig. 2 Schematic diagram of idea of AWAC algorithm

2.6.2 AWAC 算法介绍

AWAC 算法的策略改进是通过学习策略来实现的，该策略可以通过 TD 引导过程来最大化在策略评估步骤中学到的批评者的价值。通过限制策略分布来接近行动者更新期间的数据，同时最大化评论家的价值，避免外推误差。迭代至 k 步时，AWAC 算法优化策略最大化每个状态下的估计 Q 函数 $Q^{\pi_k}(s,a)$，同时限制其观察到的动作。优化 $Q^{\pi_k}(s,a)$ 相当于优化 $A^{\pi_k}(s,a)$。因此，可以将此优化策略写为：

$$\pi_{k+1}=\arg\max_{\pi\in\Pi}\mathbb{E}_{a\sim\pi(\cdot|s)}[A^{\pi_k}(s,a)] \tag{6}$$

$$\text{s.t. } D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi_\beta(\cdot|s))\leqslant\epsilon \tag{7}$$

通过合并显式学习行为模型[7,10,31]来强制执行约束会导致微调性能不佳，因此 AWAC 算法进行隐式策略约束。首先导出式(6)中约束优化的解，以获得行动者的非参数封闭形式。然后将该解决方案映射到参数策略类上，而无需任何显式行为模型。式(6)的解析解可以通过强制执行 KKT 条件来获得[31]，其中拉格朗日量为：

$$\mathscr{L}(\pi,\lambda)=\mathbb{E}_{a\sim\pi(\cdot|s)}[A^{\pi_k}(s,a)]+\lambda(\epsilon-D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi_\beta(\cdot|s))) \tag{8}$$

这个问题的封闭式解为：

$$\pi^*(a|s)\propto\pi_\beta(a|s)\exp\left(\frac{1}{\lambda}A^{\pi_k}(s,a)\right) \tag{9}$$

当使用函数逼近器时，需要将非参数解投影到策略空间。对于即时策略 π_θ，通过在数据分布 $\rho_{\pi_\beta}(s)$ 下最小化 π_θ 与最优策略 π^* 的 KL 散度来完成。

$$\arg\min_\theta\mathbb{E}_{\rho_{\pi_\beta}(s)}[D_{\mathrm{KL}}(\pi^*(\cdot|s)\|\pi_\theta(\cdot|s))]=\arg\min_\theta\mathbb{E}_{\rho_{\pi_\beta}(s)}[\mathbb{E}_{\pi^*(\cdot|s)}[-\log\pi_\theta(\cdot|s)]] \tag{10}$$

参数策略可以用 KL 散度的任一方向进行预测。选择反向 KL 会产生显式惩罚方法，该方法依赖于评估学习行为模型的密度。相反，使用前向 KL，可以通过直接从数据集 D 采样来计算策略更新。

$$\theta_{k+1}=\arg\max_\theta\mathbb{E}_{s,a\sim\beta}\left[\log\pi_\theta(a|s)\exp\left(\frac{1}{\lambda}A^{\pi_k}(s,a)\right)\right] \tag{11}$$

在算法的实际运行中，可以通过神经网络参数化行动者和评论家，并根据式(11)和式(12)执行更新：

$$\phi_k=\arg\min_{\phi}\mathbb{E}_{ij}[(Q_\phi(s,a)-y)^2] \tag{12}$$

3 DOR-PDAWAC 算法

本章采用 DOR-PDAWAC 算法对 AWAC 算法进行改进。DOR-PDAWAC 算法包括 3 个部分：1)对固定数据集中的经验数据进行优选的策略，并偏好新经验；2)双行动者-评论家模型；3)通过基于策略蒸馏的主从框架选取主行动者以及从行动者。

3.1 偏好新经验数据的数据优选方法

传统的随机数据回放方式中，假设算法从 0 时刻开始训练，直到时间步 N 结束。缓冲池的大小为 N，在智能体从固定的数据集 D 学习的每一个时刻，都会抽取新的经验并添加到缓冲池中，同时对算法进行训练。从数据集中随机抽取固定数量的样本。假设每次的回放次数为 M，则在整个训练过程中任意时刻产生的经验，其期望回放次数满足如式(13)所示：

$$N_t=\frac{M}{t}+\frac{M}{t+1}+\cdots+\frac{M}{N} \tag{13}$$

其期望被回溯的总的权重和如式(14)所示：

$$N_{w_t}=\frac{1}{M}\left(\frac{M}{t}+\frac{M}{t+1}+\cdots+\frac{M}{N}\right) \tag{14}$$

此时，如果 $t_1<t_2$，那么二者对应的经验期望的被回放次数之差为如式(15)所示：

$$N_{t_1}-N_{t_2}=\frac{M}{t_1}+\cdots+\frac{M}{t_2-1}>0 \tag{15}$$

同时，对应经验期望被回放的总权重和为：

$$N_{w_{t_1}}-N_{w_{t_2}}=\frac{1}{M}(N_{t_1}-N_{t_2})>0 \tag{16}$$

较旧的经验数据相比较新的经验数据重放次数可能会更多，旧经验重放的总权重更高，经验训练不平衡。数据集中的数据分布并不对应于当前的学习策略，同样可能会造成价值函数的估值不准确。但在实践中，新的数据经验应该比老的数据经验具有更高的权重，其更值得被采用和学习。

本小节提出 DOR 机制，通过另一种分类机制，在强调新经验的同时，缓解了上述问题。该方法在传统随机回放策略的基础上，按照经验的被回放次数划分权重。设定一个阈值 m，将回放次数 n 较少，即 $n\leqslant$m 的经验视为“新经验”，将回放次数较多的经验视为“旧经验”。回放时，从原数据集已经抽取过的经验中随机抽取 N_1 个样本作为一部分；同时，对算法偏好新经验样本进行优选，从所有新经验中额外抽取 N_2 个样本，作为另一部分。根据经验的采用次数确定优先级。为了避免局部最优和过拟合的情况发生，引入一个动态的优先级调整策略。计算评论家损失函数时，两者借助参数 α 加权合并。由此，所有数据经验均有一定概率被回放，但随着时间的推移权重逐渐减小。借助加权的方式，回放次数较少的新数据得到偏好，同时平衡新旧经验的影响，减轻过拟合的风险。该方法的评论家损失函数如式(17)所示：

$$L(q^{Q_k})=\frac{\alpha}{N_1}\sum_i(y_i-Q_k(S_i,A_i|\boldsymbol{\theta}^{Q_k}))^2+\frac{1-\alpha}{N_2}\sum_j(y_j-Q_k(S_j^{\text{new}},A_j|\boldsymbol{\theta}^{Q_k}))^2 \tag{17}$$

其中，S_i 表示旧经验的状态，S_j^{new} 表示新经验，用 i 与 j 标注以便区分。

行动者参数更新如式(18)所示：

$$\nabla_{\boldsymbol{\theta}^\mu}J(\boldsymbol{\theta}^\mu)\approx\frac{\alpha}{N_1}\sum_i\lambda+\frac{1-\alpha}{N_2}\sum_j\eta \tag{18}$$

其中，λ 与 η 参数指代如下：

$$\nabla_aQ_1(s,a|\boldsymbol{\theta}^{Q_1})|_{s=S_i,a=\mu(S_i|\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}^\mu}\mu(s|\boldsymbol{\theta}^\mu)|_{s=S_i}$$

$$\nabla_aQ_1(s,a|\boldsymbol{\theta}^{Q_1})|_{s=S_j^{\text{new}},a=\mu(S_j^{\text{new}}|\boldsymbol{\theta}^\mu)}\nabla_{\boldsymbol{\theta}^\mu\mu(s|\boldsymbol{\theta}^\mu)}|_{s=S_j^{\text{new}}}$$

3.2 双行动者-评论家模型

AWAC 算法在算法过程中使用一套行动者网络，该网络根据第一个评论家网络 $Q_1(s,a|\boldsymbol{\theta}^{Q_1})$ 的输出值进行优化。另一个评论家网络 $Q_2(s,a|\boldsymbol{\theta}^{Q_2})$ 仅在评论家的评价中起到缓解高估问题的作用，但部分行动者没有得到利用，导致网络利用率较低。另外，在训练过程中，两个评论家网络的拟合目标是一致的，相关性偏高。

AWAC 算法存在的问题如下：

1)仅存在一套行动者-评论家，算法易提前收敛。

2)只使用单一评论家，其他评论家没有得到利用。

3)评论家单步自举，以此来评价行动者，但是没有采纳偏差小的更多步数以及更多奖励信息，导致策略更新偏差。

本节介绍一种双行动者-评论家模型，该模型包含两组行动者网络，以达到行动者网络与评论家网络一一映射的效果。行动者网络 $\mu_1(s|\boldsymbol{\theta}^{\mu_1})$ 根据 $Q_1(s,a|\boldsymbol{\theta}^{Q_1})$ 进行改进，$\mu_2(s|\boldsymbol{\theta}^{\mu_2})$ 则根据 $Q_2(s,a|\boldsymbol{\theta}^{Q_2})$ 进行改进，行动者损失函数如式(19)所示：

$$\nabla_{\boldsymbol{\theta}^{\mu_i}}J(\boldsymbol{\theta}^{\mu_i})\approx\frac{1}{N}\sum_i\tau \tag{19}$$

其中，i 的取值为 1 和 2。τ 参数指代如下：

$$\tau=\nabla_aQ_i(s,a|\boldsymbol{\theta}^{Q_i})|_{s=S_i,a=\mu_i(S_i|\boldsymbol{\theta}^{\mu_i})}\nabla_{\theta^{\mu_i}}\mu_i(s|\boldsymbol{\theta}^{\mu_i})i$$

在与环境交互时，将两个评论家网络输出之和作为评价指标，从 $\mu_1(s|\boldsymbol{\theta}^{\mu_1})$ 和 $\mu_2(s|\boldsymbol{\theta}^{\mu_2})$ 中选取值较高的，加入高斯噪声来选择动作，行为策略 π_β 满足：

$$\pi_\beta(s)=\arg\max(Q_1(s,a|\boldsymbol{\theta}^{Q_1})+Q_2(s,a|\boldsymbol{\theta}^{Q_2}))+\boldsymbol{N} \tag{20}$$

其中，$a\in\{\mu_i(s|\boldsymbol{\theta}^{\mu_i})\}_{i=1,2}$。

与只使用一套行动者网络相比，本节提出的双行动者-评论家模型可以更好地实现对状态动作空间的探索，算法效率更高。

对于评论家网络，评论家 $Q_1(s,a|\boldsymbol{\theta}^{Q_1})$ 训练时在行动者 $\mu_1'(s|\boldsymbol{\theta}^{\mu_1'})$ 的基础上添加噪声平滑自举，评论家 $Q_2(s,a|\boldsymbol{\theta}^{Q_2})$ 在行动者 $\mu_2'(s|\boldsymbol{\theta}^{\mu_2'})$ 的基础上添加噪声平滑自举。二者自举所采用的目标动作 $\tilde{a}_i$ 如式(21)所示：

$$\tilde{a}_i=\mu_i'(s|\boldsymbol{\theta}^{\mu_i'})+\varepsilon,i=1,2 \tag{21}$$

评论家评估目标 y_i 满足：

$$y_i=r'+\gamma\min_{k=1,2}[\lambda Q_k'(s',\tilde{a}_i|\boldsymbol{\theta}^{Q_k'})-(\pi(s')-\tilde{a}_i)^2] \tag{22}$$

其中，i 的取值为 1 和 2。

每个时间步，分别选取动作 $a^*=\arg\max_a Q^A(s,a)$，$b^*=\arg\max_b Q^B(s,a)$。

对于两个表格式值函数估计 Q^A 以及 Q^B，算法更新时，目标值为 y_A，y_B，分别满足式(23)和式(24)：

$$y_A=r'+\gamma\min[[\lambda Q^A(s',a^*)-(\pi(s')-a^*)^2],[\lambda Q^B(s',a^*)(\pi(s')-a^*)^2]] \tag{23}$$

$$y_B=r'+\gamma\min[[\lambda Q^A(s',b^*)-(\pi(s')-b^*)^2],[\lambda Q^B(s',b^*)(\pi(s')-b^*)^2]] \tag{24}$$

分别对两个动作值函数进行更新，更新式如式(25)和式(26)：

$$Q^A(s,a)=Q^A(s,a)+\alpha_t(s,a)(y_A-Q^A(s,a)) \tag{25}$$

$$Q^B(s,a)=Q^B(s,a)+\alpha_t(s,a)(y_B-Q^B(s,a)) \tag{26}$$

接下来证明函数 Q^A 和 Q^B 的期望收敛性。

引理 1 考虑随机过程(ζ_t,Δ_t,F_t)，$t\geqslant 0$，其中 $\zeta_t,\Delta_t,F_t:X\to\mathbb{R}$ 满足式(27)：

$$Q^B(s,a)=Q^B(s,a)+\alpha_t(s,a)(y_B-Q^B(s,a)) \tag{27}$$

其中，$x_t\in X$ 且 $t=0,1,2,\cdots$。令 P_t 为一递增 σ 域，满足 ζ_0 以及 Δ_0 为 P_0 可估计且 ζ_t,Δ_t 以及 F_{t-1} 为 P_t 可计算，$t=1,2,\cdots$。假定以下条件满足：

1)集合 X 为有限集；

2)$\zeta_t(x_t)\in[0,1]$，$\sum_t\zeta_t(x_t)=\infty$，$\sum_t(\zeta_t(x_t))^2<\infty$ 且 $\forall x\neq x_t:\zeta_t(x)=0$；

3)$\|\mathbb{E}\{F_t|P_t\}\|\leqslant\kappa\|\Delta_t\|+c_t$，其中 $\kappa\in[0,1)$ 且 c_t 以 1 的概率收敛至 0；

4)$Var\{F_t(x_t)|P_t\}\leqslant K(1+\kappa\|\Delta_t\|)^2$，其中 K 为常数；

则 Δ_t 以 1 的概率收敛至 0。

引理 1 的证明参考文献[23]。

定理 1 假定以下条件满足，在已经给出各个状态的马尔可夫决策过程中，若学习的策略是合适的，则该策略可以给出很大数目的经验，包含状态-动作对以及对应奖赏，使用本节算法更新的 Q^A 以及 Q^B 将以 1 的概率收敛至最优值函数 Q^*。条件如下：

1)马尔可夫决策过程有限，即$|S\times A|<\infty$；

2)$\gamma\in[0,1)$；

3)动作值存储在查询表中；

4)Q^A 以及 Q^B 均更新无数次；

5)$\alpha_t(s,a)\in[0,1]$，$\sum_t\alpha_t(s,a)=\infty$，$\sum_t(\alpha_t(s,a))^2<\infty$ 且 $\forall(s,a)\neq(s_t,a_t):\alpha_t(s,a)=0$；

6)$\forall s,a,s':Var\{R^{s'}_{sa}\}<\infty$。

证明：函数 Q^A 与 Q^B 对称，证明任意一个即可。

以下证明函数 Q^A 的收敛性。

应用引理至 P_t，P_t 表示如下：

$$P_t=\{Q^A_0,Q^B_0,S_0,A_0,\alpha_0,R_1,S_1,\cdots,S_t,A_t\} \tag{28}$$

$X=S\times A$ 且 $\Delta_t=Q^A_t-Q^*$，$\zeta=\alpha$。

同令 $F_t(S_t,A_t)$ 如下：

$$F_t(S_t,A_t)=R_{t+1}+\gamma\min(Q^A_t(S_{t+1},a^*),Q^B_t(S_{t+1},a^*))-Q^*_t(S_t,A_t)$$

$$F_t(S_t,A_t)=R_{t+1}+\gamma\min(Q^A_t(S_{t+1},a^*),Q^B_t(S_{t+1},a^*))-Q^*_t(S_t,A_t) \tag{29}$$

其中，F^Q_t 可表达为式(30)：

$$F^Q_t=R_{t+1}+\gamma Q^A_t(S_{t+1},a^*)-Q^*_t(S_t,A_t) \tag{30}$$

已知$\mathbb{E}\{F^Q_t|P_t\}\leqslant\gamma\|\Delta_t\|$，为了应用引理 1，令：

$$c_t=\gamma(\min(Q^A_t(S_{t+1},a^*),Q^B_t(S_{t+1},a^*))-Q^A_t(S_{t+1},a^*)) \tag{31}$$

$$\Delta^{BA}_t=Q^B_t(S_t,A_t)-Q^A_t(S_t,A_t) \tag{32}$$

此时若 Δ^{BA}_t 收敛至 0，则 c_t 收敛至 0。可推得式(33)：

$$\Delta^{BA}_{t+1}=\Delta^{BA}_t+\alpha_t(S_t,A_t)(F^B_t(S_t,A_t)-F^A_t(S_t,A_t)) \tag{33}$$

其中，$F^A_t(S_t,A_t)$ 与 $F^B_t(S_t,A_t)$ 可分别表达为式(34)和式(35)：

$$F^A_t(S_t,A_t)=R_{t+1}+\gamma\min(Q^A_t(S_{t+1},a^*),Q^B_t(S_{t+1},a^*))-Q^A_t(S_t,A_t) \tag{34}$$

$$F^B_t(S_t,A_t)=R_{t+1}+\gamma\min(Q^A_t(S_{t+1},b^*),Q^B_t(S_{t+1},b^*))-Q^B_t(S_t,A_t) \tag{35}$$

则可以得到：

$$\begin{aligned}\Delta^{BA}_{t+1}&=\Delta^{BA}_t+\alpha_t(S_t,A_t)(F^B_t(S_t,A_t)-F^A_t(S_t,A_t))\\&=(1-\alpha_t(S_t,A_t))\Delta^{BA}_t+\alpha_t(S_t,A_t)F^{BA}_t(S_t,A_t)\end{aligned} \tag{36}$$

其中 F^{BA}_t 可以表达为式(37)：

$$\begin{aligned}F^{BA}_t&=\gamma\min(Q^A_t(S_{t+1},b^*),Q^B_t(S_{t+1},b^*))-\gamma\min(Q^A_t(S_{t+1},a^*),Q^B_t(S_{t+1},a^*))\\&=\gamma Q^A_t(S_{t+1},b^*)-\gamma Q^B_t(S_{t+1},a^*)\end{aligned} \tag{37}$$

假定$\mathbb{E}\{Q^A_t(S_{t+1},b^*)|P_t\}\geqslant\mathbb{E}\{Q^B_t(S_{t+1},a^*)|P_t\}$，由定义可知：

$$Q^A_t(S_{t+1},a^*)=\max_a Q^A_t(S_{t+1},a^*)\geqslant Q^A_t(S_{t+1},b^*) \tag{38}$$

因此可以得到式(39)：

$$\begin{aligned}|\mathbb{E}\{F^{BA}_t(S_t,A_t)|P_t\}|&=\gamma\,\mathbb{E}\{Q^A_t(S_{t+1},b^*)-Q^B_t(S_{t+1},a^*)|P_t\}\\&\leqslant\gamma\,\mathbb{E}\{Q^A_t(S_{t+1},a^*)-Q^B_t(S_{t+1},a^*)|P_t\}\\&\leqslant\gamma\|\Delta^{BA}_t\|\end{aligned} \tag{39}$$

反之，如果假定$\mathbb{E}\{Q^B_t(S_{t+1},a^*)|P_t\}\gg\mathbb{E}\{Q^A_t(S_{t+1},b^*)|P_t\}$，由定义可知：

$$Q^B_t(S_{t+1},b^*)=\max_a Q^B_t(S_{t+1},b^*)\geqslant Q^B_t(S_{t+1},a^*) \tag{40}$$

则可以得到式(41)：

$$\begin{aligned}|\mathbb{E}\{F^{BA}_t(S_t,A_t)|P_t\}|&=\gamma\,\mathbb{E}\{Q^B_t(S_{t+1},a^*)-Q^A_t(S_{t+1},b^*)|P_t\}\\&\leqslant\gamma\,\mathbb{E}\{Q^B_t(S_{t+1},b^*)-Q^A_t(S_{t+1},b^*)|P_t\}\\&\leqslant\gamma\|\Delta^{BA}_t\|\end{aligned} \tag{41}$$

由于两种情形都可以实现，因此可以得到式(42)：

$$|\mathbb{E}\{F^{BA}_t|P_t\}|\leqslant\gamma\|\Delta^{BA}_t\| \tag{42}$$

综上所述，Δ^{BA}_t 期望值收敛至 0，c_t 期望值收敛至 0，因此 Δ_t 期望值收敛至 0，Q^A 期望值收敛。函数 Q^A 与 Q^B 更新对称，同理，可证得 Q^B 期望值收敛。

由此，该问题得证。

3.3 策略蒸馏的主仆框架

本节对上节中 AWAC 算法的双行动者-评论家模型额外加入一套行动者，同时使用策略蒸馏与主仆框架(KMAA)来

增强两套行动者之间的合作。在训练中根据不同奖励情况区分主行动者与从行动者,进一步加强协作的精度与效率。

基于策略蒸馏的主仆框架主要包括以下两部分:

1)动态选主(Dynamic Master Selection,DMS)部分:在训练时针对不同的情节,将行动者区分为主要行动者和仆从行动者,行动者中对训练效果提升作用大的一方作为主要行动者引导训练过程,仆从行动者作为作用小的一方,从主要行动者的知识模型中进行学习。

2)策略蒸馏(Policy Distillation,PD)部分:主要负责仆从行动者对主行动者知识模型的提炼,参考不同训练阶段中主行动者和仆从行动者对训练提升帮助的差异大小,区分二者的接近程度,从而提升整体算法效率。

3.3.1 动态选主部分(DMS)

将主要行动者表示为 π_{ϕ_m},主要评论家表示为 Q_{θ_m},仆从行动者表示为 π_{ϕ_s},仆从评论家表示为 Q_{θ_s}。在回合开始之前,使用动态主选择模块(DMS)进行采样,采样的行动者将成为主行动者并主导训练,而另一个行动者则成为仆从行动者并在当前回合中跟随主行动者。在这一回合中,主行动者 π_{ϕ_m} 执行训练程序。π_{ϕ_m} 根据关联评论家采取确定性策略梯度进行更新,而 π_{ϕ_s} 通过策略蒸馏提取 π_{ϕ_m} 的知识。在回合结束时,DMS 根据获得的回合返回进行更新,整个过程重复 T 个时间步长。

DMS 的作用是自适应地选择表现相对更好并且对当前训练过程更有用的行动者作为主行动者。为了满足这一要求,使用指数权重框架来派生自适应算法。在回合 m 之前考虑双重行动者作为候选人。决策 $i_m\in\{1,2\}$ 从分布 p_m 中采样,形式为 $p_m(i)\propto e^{s_m(i)}$,其中 s_m 表示分数并更新如下:

$$s_{m+1}(i)=\begin{cases}s_m(i)+\eta\dfrac{G_m-G_{m-1}}{p_m(i)}, & i_m=i\\ s_m(i), & i_m\neq i\end{cases}\tag{43}$$

其中,G_m 表示在第 m 情节获得的累计回报。通过这种方式,如果当前回合的回报比之前更高,则倾向于继续使用当前的主行动者。如果情况相反,则表明当前回合使用的主行动者的学习效率下降,切换主行动者的概率增加。因此,可以假设主行动者在当前阶段表现较好,而仆从行动者表现相对较差。

主行动者 π_{ϕ_m} 用于与环境交互,行为策略 μ_b 在此基础上添加高斯噪声,σ_b 为方差。表达式如下:

$$\mu_b(s)=\pi_{\phi_m}(s)+\mathcal{N}(0,\sigma_b)\tag{44}$$

利用上述策略,生成的经验质量相对较高,从而进一步提高训练效率。

3.3.2 策略蒸馏部分(PD)

首先从理论上证明仆从行动者向主行动者靠近时会影响算法性能并使之提升。将主要行动者表示为 μ,仆从行动者表示为 $\tilde{\mu}$,仆从行动者的改进前后状态分别表示为 $\tilde{\mu}_{\text{old}}$ 和 $\tilde{\mu}_{\text{new}}$。将改进前的仆从行动者的动作值表示为 $q^{\tilde{\mu}_{\text{old}}}(s,a)$,改进后为动作值表示为 $q^{\tilde{\mu}_{\text{new}}}(s,a)$。

也就是说,针对每一个状态 s,如果 $q^{\tilde{\mu}_{\text{old}}}(s,\tilde{\mu}_{\text{new}}(s))\geqslant q^{\tilde{\mu}_{\text{old}}}(s,\tilde{\mu}_{\text{old}}(s))$ 成立,则 $q^{\tilde{\mu}_{\text{new}}}(s,a)\geqslant q^{\mu_{\text{old}}}(s,a)$ 也成立。

证明如下:

$$\begin{aligned}
q^{\mu_{\text{old}}}(s,a)&=\mathbb{E}[r_{t+1}+\gamma q^{\mu_{\text{old}}}(s_{t+1},\tilde{\mu}_{\text{old}}(s_{t+1}))\mid s_t=s,a_t=a]\\
&\leqslant\mathbb{E}[r_{t+1}+\gamma q^{\mu_{\text{old}}}(s_{t+1},\tilde{\mu}_{\text{new}}(s_{t+1}))\mid s_t=s,a_t=a]\\
&=\mathbb{E}[r_{t+1}+\gamma E_{\tilde{\mu}_{\text{new}}}[r_{t+2}+\gamma q^{\tilde{\mu}_{\text{old}}}(s_{t+2},\tilde{\mu}_{\text{old}}(s_{t+2}))\mid s_{t+1}]\mid s_t=s,a_t=a]\\
&\leqslant\mathbb{E}[r_{t+1}+\gamma E_{\tilde{\mu}_{\text{new}}}[r_{t+2}+\gamma q^{\tilde{\mu}_{\text{old}}}(s_{t+2},\tilde{\mu}_{\text{new}}(s_{t+2}))\mid s_{t+1}]\mid s_t=s,a_t=a]\\
&=\mathbb{E}_{\tilde{\mu}_{\text{new}}}[r_{t+1}+\gamma r_{t+2}+\gamma^2 q^{\tilde{\mu}_{\text{old}}}(s_{t+2},\tilde{\mu}_{\text{new}}(s_{t+2}))\mid s_t=s,a_t=a]\\
&\cdots\\
&\leqslant\mathbb{E}_{\tilde{\mu}_{\text{new}}}[r_{t+1}+\gamma r_{t+2}+\gamma^2 r_{t+3}+\cdots\mid s_t=s,a_t=a]\\
&=q^{\tilde{\mu}_{\text{new}}}(s,a)
\end{aligned}$$

从以上证明可知,仆从行动者向主行动者靠近时算法性能会得到提升,而策略蒸馏部分的主要作用是找到使主从行动者互相靠近的方法。本节同样将主要行动者表示为 π_{ϕ_m},主要评论家表示为 Q_{θ_m},仆从行动者表示为 π_{ϕ_s},仆从评论家表示为 Q_{θ_s}。在训练过程中,π_{ϕ_m} 根据确定性策略梯度更新:

$$\nabla_{\phi_m}J(\phi_m)=\mathbb{E}_{s\sim D}[\nabla_a Q_{\theta_m}(s,a)|_{a=\pi_{\phi_m}(s)}\nabla_{\phi_m}\pi_{\phi_m}(s)]\tag{45}$$

算法进行 PD 过程时,使用从经验池 D 中抽取的样本进行训练。由于两组行动者动态变化,区别经验池变得困难,即很难确定当前状态对应于哪个行动者。使用策略蒸馏使从行动者向主行动者逼近的同时,设两组行动者具有几乎同样的状态矩阵,通过该方法,算法可以使用同一个经验池,样本利用率也会获得提升。π_{ϕ_s} 靠近 π_{ϕ_m},最小化目标如下:

$$J^w(\phi_m,\phi_s)=\mathbb{E}_{s\sim D}[\|\pi_{\phi_m}(s)-\pi_{\phi_s}(s)\|_2^2 e^{\alpha\eta(s)}]\tag{46}$$

其中,$\eta(s)$ 表示从重播缓冲区 D 采样的每个状态的优势。由于要训练不同阶段评论家估计的偏差,该算法并不基于 $\eta(s)$ 的值来评判仆行动者在接近主行动者时是否主导积极和消极决策,而是使用加权行为克隆方式进行策略蒸馏,其公式如下:

$$\eta(s)=Q_{\theta_m}(s,\pi_{\phi_m}(s))-Q_{\theta_m}(s,\pi_{\phi_s}(s))\tag{47}$$

α 控制置信水平,负责控制评论家的估计误差对 PD 过程的影响。故应根据价值函数估计的准确程度来选择该参数。其中 π_{ϕ_m} 越有利,对蒸馏就越有帮助,否则 π_{ϕ_s} 应当维持当前的策略。采用行为克隆方式与使用参数进行调整的方式,仆从行动者的学习过程更有效。

3.4 数据经验优选回放的主仆框架双行动者 AWAC 算法

本节介绍运用主仆架构双行动者-评论家模型,同时进行数据经验优选回放的 DOR-PDAWAC 算法。在本算法中,按情节划分阶段,单个情节中的流程如下:

1)在训练开始前,运用动态主选择部分选取主要行动者进行训练主过程。

2)训练中,根据离线数据采样,主行动者生成经验到经验池,根据一一映射的主评论家,主行动者根据其估计的值进行改进。仆从行动者应用策略蒸馏部分评估接近主行动者的程度。

3)训练结束后,根据即时的奖励与之前相比的差值来对动态主选择部分进行更新。

DOR-PDAWAC 算法简化结构如图 3 所示,其展示了整个算法的大致运行流程,其中简化了目标网络。具体如算法 1 所示。

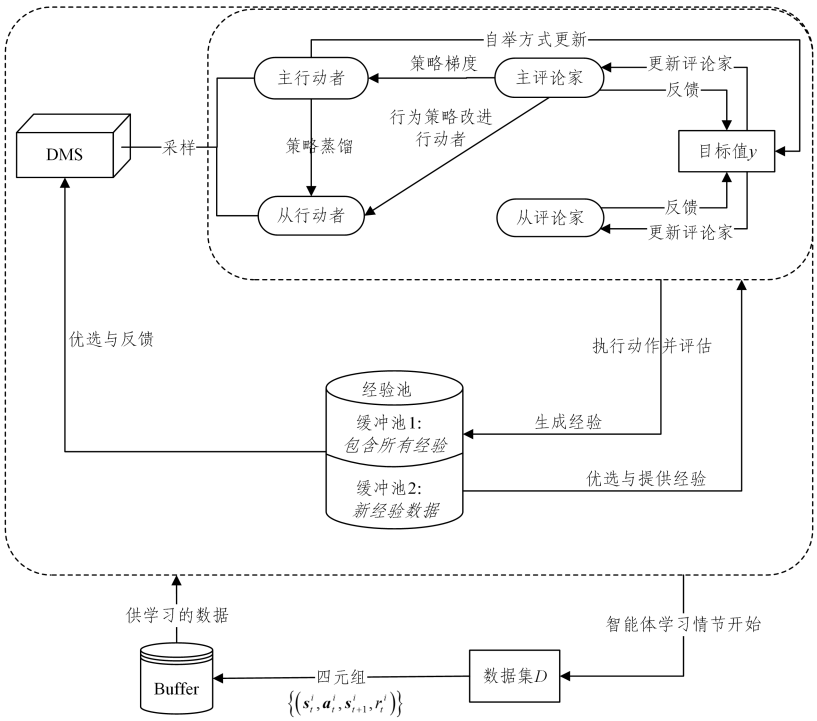


图 3 DOR-PDAWAC 算法结构示意图

Fig. 3 Schematic diagram of DOR-PDAWAC algorithm structure

算法 1 DOR-PDAWAC 算法

输入：超参数 λ，数据集 D，训练缓冲池 E，最大时间步 T，延迟更新频率 d，扰动截断值 c，软更新系数 τ，策略蒸馏参数 α

输出：评论家网络参数 $\boldsymbol{\theta}^{Q_1}$ 及 $\boldsymbol{\theta}^{Q_2}$，行动者网络参数 $\boldsymbol{\theta}^{\mu_1}$ 及 $\boldsymbol{\theta}^{\mu_2}$

1. 初始化：评论家网络参数 $\boldsymbol{\theta}^{Q_1}$ 及 $\boldsymbol{\theta}^{Q_2}$，行动者网络参数 $\boldsymbol{\theta}^{\mu_1}$ 及 $\boldsymbol{\theta}^{\mu_2}$，目标评论家网络参数 $\boldsymbol{\theta}^{Q_1'}$ 及 $\boldsymbol{\theta}^{Q_2'}$，目标行动者网络参数 $\boldsymbol{\theta}^{\mu_1'}$ 及 $\boldsymbol{\theta}^{\mu_2'}$

2. for episode=1 to M do

3. 初始化抽取样本数据 $\{(s_t^i, a_t^i, s_{t+1}^i, r_t^i)\}$

4. 以 0.5 的概率选取一套行动者网络为主行动者

5. for t=0 to T do

6. 从数据集 D 中随机批量抽取 N_1 个旧经验的四元组 $(S_i, A_i, S_{i+1}, R_{i+1})$

7. 缓冲区 β=D

8. 样本批次数据 $(s, a, s', r) \sim \beta$

9. 根据行为策略采样选取动作 $a_t \sim \mu(s_t)$，行为策略 $\mu(s_t)=\pi_{\phi_m}(s_t)+\mathcal{N}$

10. 将以上所有经验添加至训练缓冲池 E 中

11. 从数据集 D 中优选出回放次数小于阈值 m 的经验并随机批量抽取 N_2 个新经验 $(S_j^{new}, A_j^{new}, R_{j+1}^{new}, S_{j+1}^{new})$

12. 更新以上所有经验对应训练次数 n←n+1

13. 添加高斯噪声 $\varepsilon \sim \mathcal{N}(0, \sigma)$ 得到动作 $A_t=\bar{A}_t+\varepsilon$，根据目标行动者网络以及噪声获取动作

14. 第 1 套行动者的目标动作 $\widetilde{A}_{i_2}=\mu_1'(S_{i+1}|\boldsymbol{\theta}^{\mu_1'})+\varepsilon$，$\widetilde{A}_{j_1}=\mu_1'(S_{j+1}^{new}|\boldsymbol{\theta}^{\mu_1'})+\varepsilon'$

第 2 套行动者的目标动作 $\widetilde{A}_{i_2}=\mu_2'(S_{i+1}|\boldsymbol{\theta}^{\mu_2'})+\varepsilon$，$\widetilde{A}_{j_2}=\mu_2'(S_{j+1}^{new}|\boldsymbol{\theta}^{\mu_2'})+\varepsilon'$

15. 噪声 $\varepsilon \sim \mathrm{clip}(\mathbf{N}(0, \tilde{\sigma}), -c, c)$，$\varepsilon' \sim \mathrm{clip}(\mathbf{N}(0, \tilde{\sigma})', -c, c)$

16. 分别限制两个策略之间的 KL 散度，计算优势函数的最大期望值

$$y_{i_0}=r(S_i, A_i)+\gamma\, \mathbb{E}_{S', A'}[Q_{\varphi_{k-1}}(S_{i+1}, A_{i+1}')]$$

$$y_{j_0}=r(S_j, A_j)+\gamma\, \mathbb{E}_{S', A'}[Q_{\varphi_{k-1}}(S_{j+1}, A_{j+1}')]$$

17. 通过拉格朗日乘数法求多元函数的最值

$$\mathcal{L}(\pi, \lambda)=\mathbb{E}_{\mathbf{a}\sim\pi(\cdot|s)}[A_{i,j}(\mathbf{s}, \mathbf{a})]+\lambda(\epsilon-D_{KL}(\pi(\cdot|\mathbf{s}) \| \pi_\beta(\cdot|\mathbf{s})))$$

18. 最小化当前策略和最佳策略的 KL 散度

$$\arg\min_\theta \mathbb{E}_{\rho_{\pi_\beta}(s)}[D_{KL}(\pi^*(\cdot|s) \| \pi_\theta(\cdot|s))]=\arg\min_\theta \mathbb{E}_{\rho_{\pi_\beta}(s)}[\lim_{\pi^*(\cdot|s)}[-\log \pi_\theta(\cdot|s)]]$$

$$\phi_i \leftarrow \arg\min_\phi \mathbb{E}_{\mathcal{D}}[(Q_{\phi_i}(S, A_i)-y_{i_0})^2]$$

$$\phi_j \leftarrow \arg\min_\phi \mathbb{E}_{\mathcal{D}}[(Q_{\phi_j}(S, A_j)-y_{j_0})^2]$$

19. 与第 1 套评论家网络关联的目标值

$$y_{i_1}=R_{i+1}+\gamma \min_{k=1,2} Q_k'(S_{i+1}, \widetilde{A}_{i_2}|\boldsymbol{\theta}^{Q_k'})$$

$$y_{j_1}=R_{j+1}^{new}+\gamma \min_{k=1,2} Q_k'(S_{j+1}^{new}, \widetilde{A}_{j_2}|\boldsymbol{\theta}^{Q_k'})$$

20. 与第 2 套评论家网络关联的目标值

$$y_{i_2}=R_{i+1}+\gamma \min_{k=1,2} Q_k'(S_{i+1}, \widetilde{A}_{i_2}|\boldsymbol{\theta}^{Q_k'})$$

$$y_{j_2}=R_{j+1}^{new}+\gamma \min_{i=1,2} Q_k'(s_{j+1}^{new}, \widetilde{A}_{j_2}|\boldsymbol{\theta}^{Q_k'})$$

21. 最小化损失函数

$$L(\boldsymbol{\theta}^{Q_1})=\frac{\alpha}{N_1}\sum_i(y_{i_1}-Q_1(S_i, A_i|\boldsymbol{\theta}^{Q_1}))^2+\frac{1-\alpha}{N_2}\sum_j(y_{j_1}-Q_1(S_j^{new}, A_j^{new}|\boldsymbol{\theta}^{Q_1}))^2$$

$$L(\boldsymbol{\theta}^{Q_2})=\frac{1-\alpha}{N_2}\sum_j(y_{j_2}-Q_2(S_j^{new}, A_j^{new}|\boldsymbol{\theta}^{Q_2}))^2$$

以更新评论家网络参数

22. 进行策略更新

$$\boldsymbol{\theta}_{t+1}=\arg\max_\theta \mathbb{E}_{\mathbf{s}, \mathbf{a}\sim\beta}\left[\log \pi_{\theta_t}(a|s)\exp\left(\frac{1}{\lambda}A(s, a)\right)\right]$$

23. if t mod d=0 then：

24. 计算梯度用于更新主行动者网络

$$\frac{1}{N}\sum_i \nabla_a Q_{\theta_m}(s,a)|_{s=s_i,a=\pi_{\phi_m}(s_i)} \nabla_{\phi_m}\pi_{\phi_m}(s)|_{s=s_i}$$

25. 最小化式(46)以更新仆从行动者网络

26. 借助训练缓冲池 E，以相反顺序回放经验，求得动作 $\widetilde{A}_{i_2}=\mu_1'(S'_{i+1}|\theta^{\mu_2'})+\varepsilon''$

27. 噪声 $\varepsilon''\sim \mathrm{clip}(\mathbf{N}(0,\widetilde{\sigma}),-c,c)$

28. 计算目标值 $y_{j_2}=R_{j+1}^{new}+\gamma \min_{i=1,2} Q_k'(s_{j+1}^{new},\widetilde{A}_{j_2}|\theta^{Q_k'})$

29. 最小化损失函数

$$L(\theta^{Q_2})=\frac{\alpha}{N_1}\sum_i (y_{i_2}-Q_2(S_i',A_i'|\theta^{Q_2}))^2$$

以更新评论家网络参数

30. 清空训练缓冲池 E

31. 计算策略梯度并更新行动者网络参数

$$\nabla_{\theta^{\mu_1}} J(\theta^{\mu_1})\approx\frac{\alpha}{N_1}\sum_i \nabla_a Q_1(s,a|\theta^{Q_1})|_{s=S_i,a=\mu_1(S_i;\theta^{\mu_1})}\nabla_{\theta^{\mu_1}}\mu_1(s|\theta^{\mu_1})|_{s=S_i}+\frac{1-\alpha}{N_2}\sum_j \nabla_a Q_1(s,a|\theta^{Q_1})|_{s=S_j^{new},a=\mu_1(S_j^{new};\theta^{\mu_1})}\nabla_{\theta^{\mu}}\mu_1(s|\theta^{\mu_1})|_{s=S_j^{new}}$$

$$\nabla_{\theta^{\mu_2}} J(\theta^{\mu_2})\approx\frac{\alpha}{N_1}\sum_i \nabla_a Q_2(s,a|\theta^{Q_2})|_{s=S_i',a=\mu_2(S_i'|\theta^{\mu_2})}\nabla_{\theta^{\mu_1}}\mu_1(s|\theta^{\mu_2})|_{s=S_i'}+\frac{1-\alpha}{N_2}\sum_j \nabla_a Q_2(s,a|\theta^{Q_2})|_{s=S_j^{new},a=\mu_2(S_j^{new}|\theta^{\mu_2})}\nabla_{\theta^{\mu}}\mu_2(s|\theta^{\mu_2})|_{s=S_j^{new}}$$

32. 软更新目标网络参数：

$$\theta^{Q_1'}=\tau\theta^{Q_1}+(1-\tau)\theta^{Q_1'},\theta^{Q_2'}=\tau\theta^{Q_2}+(1-\tau)\theta^{Q_2'}$$

$$\theta^{\mu_1'}=\tau\theta^{\mu_1'}+(1-\tau)\theta^{\mu_1},\theta^{\mu_2'}=\tau\theta^{\mu_2'}+(1-\tau)\theta^{\mu_2}$$

33. end if

34. if 情节结束 then：

35. 根据式(43)更新 DMS 模块

36. 根据更新后的 DMS 模块采样计算选取主行动者

37. 初始化状态作为下一情节的起始点

38. end if

39. end for

40. end for

其中，第 5－12 行是智能体从经验池中抽取的过程，使用数据优选机制进行筛选，选出适合当前回合学习的元组；第 11 行是挑选新经验的过程，将回放次数不超过某阈值的经验视作新经验，再从其中额外抽取一定数目的经验用于回放；第 13 行使用高斯噪声获取动作；第 16－18 行与第 22 行是 AWAC 算法特有的在线微调并离线学习的过程；第 25 行为仆从行动者的训练，其采取提出的策略蒸馏方式靠近主行动者；第 34－38 行为 DMS 模块的计算和运用的过程，其中第 35 行为计算情节回报提升量并更新 DMS 模块的过程，第 36 行为借助更新后的 DMS 模块采样主行动者用于下一情节处理流程的过程；第 11－38 行是智能体每一步训练，更新行动者以及评论家网络，实现其学习的过程。

4 实验与结果

4.1 实验环境介绍

在离线环境应用的初期，没有固定的评估标准，随着其不断发展，Kumar 等在文献[32]中提出了离线强化学习评估的基准数据集 D4RL，并且给出了衡量标准，以方便对比各算法的性能。D4RL 数据集中包含了种类丰富的模拟环境及任务，包括 Maze2D，AntMaze，Gym-MuJoCo 等多项任务环境。同时，D4RL 提供了非常简单的 API 接口，便于学习者直接获取数据集以完成智能体的训练。另外，D4RL 还提供了丰富的离线强化学习经典的基准算法，包括 BCQ，BEAR，BRAC 等多个经典的离线强化学习算法可供对比。现在，D4RL 数据集已经成为最常用的离线强化学习评估标准数据集。

由于数据的不稳定性(非平稳性)可能影响模型的收敛性，故本文利用官方 D4RL 数据集中提供的 Gym-MuJoCo[34] 任务环境，其中每个任务包括 5 种不同等级的数据集以供实验对比。1)随机数据(Random Data)：数据由一个随机初始化策略的智能体收集得到。2)中级数据(Medium Data)：数据由使用 SAC 方法训练的智能体收集得到。3)中级回放数据(Medium-replay Data)：数据包括从训练开始直到满足中级等级表现的智能体训练过程中收集的全部数据。4)中级专家数据(Medium-expert Data)：数据由训练到专家的智能体和经过一定训练的智能体数据混合得到。5)专家数据(Expert Data)：数据由专家样本组成。

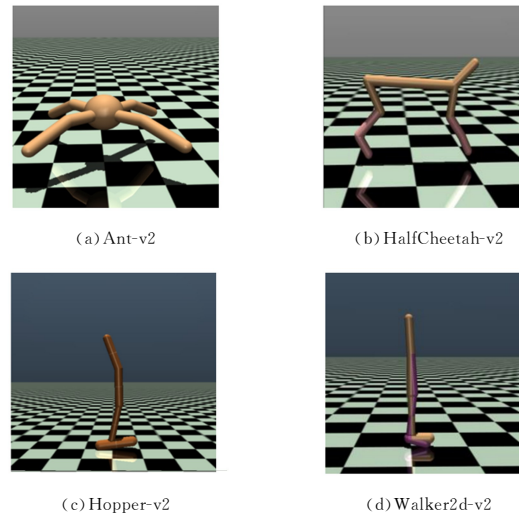
如图 4 所示，实验在 Gym-MuJoCo 任务环境的 4 个任务上进行。

1)Ant-v2 任务：让四脚的平趴体态机器人在三维空间中能够正常且快速地前进。

2)HalfCheetah-v2 任务：让两脚的马型体态机器人在二维空间中能够正常且快速地前进。

3)Hopper-v2 任务：让单腿的杆型体态机器人在二维空间中能够正常且快速地跳着前进。

4)Walker2d-v2 任务：让双腿的杆型体态机器人在二维空间中能够正常且快速地前进。



(a)Ant-v2 (b)HalfCheetah-v2

(c)Hopper-v2 (d)Walker2d-v2

图 4 4 种 Mujoco 环境

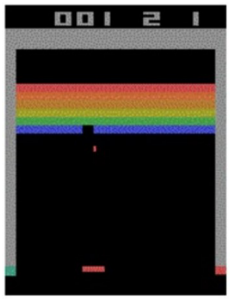
Fig. 4 Four Mujoco environments

本节首先分别说明数据优选与经验偏好机制、策略蒸馏的主仆架构与双行动者-评论家模型的效果；然后对融合以上改进的 DOR-PDAWAC 及单纯的 PDAWAC 等算法进行消融实验对比，证实所提出方法的有效性；最后与离线强化学习

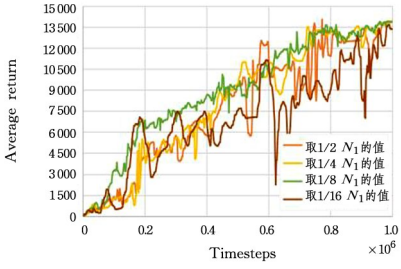
经典算法进行对比实验，进一步证明算法的有效性。为防止计算资源不足引起的无法进行足够次数的迭代来达到收敛的问题，本文实验使用较先进的 Intel Core™ CPU i7-10500U 处理器，使用 GeForce RTX 3070ti GPU 进行辅助加速计算。

4.2 实验参数设置

在实验中，DOR-AWAC 算法采用一个当前行动者网络和一个目标行动者网络，以及两个当前评论家网络及目标评论家网络，而 DOR-PDAWAC 与 PDAWAC 则额外增加一个当前行动者网络以及目标行动者网络。行动者网络以及评论家网络采用 3 层全连接网络模型。不同于传统优化问题，离线强化学习涉及非凸优化，为避免不收敛的情况发生，本文采用 Adam 优化器进行优化。在以策略梯度算法为基础的所有算法中，学习率通常设置为 3×10^{-4}，缓冲池大小均设置为 10000。批量选取样本时，为保持探索性的同时防止过拟合，旧经验 $N_1=256$。为确定 N_2 的值如何选取，在图 5(a)所示的 OpenAI Gym-Atari 环境的 BreakOut 游戏中测试如何选取新旧经验的比例使算法效率最优。如图 5(b)所示，实验显示新旧经验抽取比例为 1∶8 时算法表现较佳，算法较稳定，波动较小，算法效率较高，故选取 $N_2=32$。阈值 m 稍小于每次抽取的新经验数效果较好，设定 $m=30$。



(a)BreakOut 游戏环境截图



(b)测试选取新旧经验的比例实验结果

图 5 按几种比例选取新经验的实验及结果

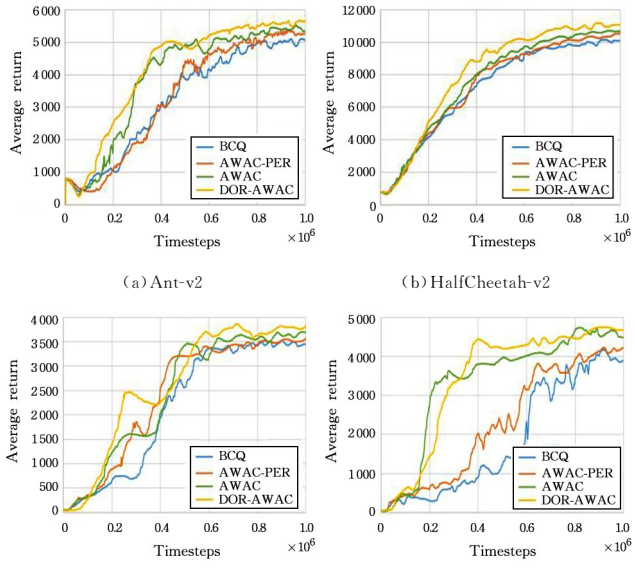
Fig. 5 Experiments and results with new experience selected according to several proportions

新经验回放部分与全部数据回放部分加权时，设定权重 α 为 0.8，以在鼓励探索的同时降低局部最优的可能性。在离线强化学习算法中，每个情节最大时间步一般设置为 $T_{max}=1000$，超过该步数情节重新开始。与 AWAC 算法相同，折扣因子一般设置为 $\gamma=0.99$，目标网络更新时软更新参数设置为 $\tau=0.005$。智能体抽取数据集时，高斯噪声部分一般设定为 $\sigma=0.2$，目标噪声平滑时方差一般为 $\bar{\sigma}=0.1$，截断参数一般为 $c=0.5$。在实验中，一般每隔 5000 步，算法在测试环境中评估 10 个情节，计算平均回报。

4.3 实验与结果

4.3.1 数据经验优选与偏好新经验机制的实验和结果

本小节在 4 个连续控制任务上对数据集优选的 DOR-AWAC 算法以及原始的 AWAC 算法进行对比。同时为了进一步说明数据优选与偏好新经验回放机制的优势，添加了融合传统的优先经验回放机制的 AWAC-PER 算法作为对比。另外，实验中还加入了经典的 BCQ 算法作为评估基准，实验结果如图 6 所示。可以看出，在 4 个环境下，AWAC-PER 算法由于不适于传统的优先经验回放机制，表现不佳；AWAC 算法训练总体性能相比 BCQ 算法表现较佳，且优于 AWAC-PER 算法。AWAC 算法在训练早期，智能体在环境中探索，表现一般，而 DOR-AWAC 由于优选经验，会进一步放大这些较差探索经验的影响。而总体上 DOR-AWAC 算法的性能优于 AWAC 算法。



(a)Ant-v2 (b)HalfCheetah-v2

(c)Hopper-v2 (d)Walker2d-v2

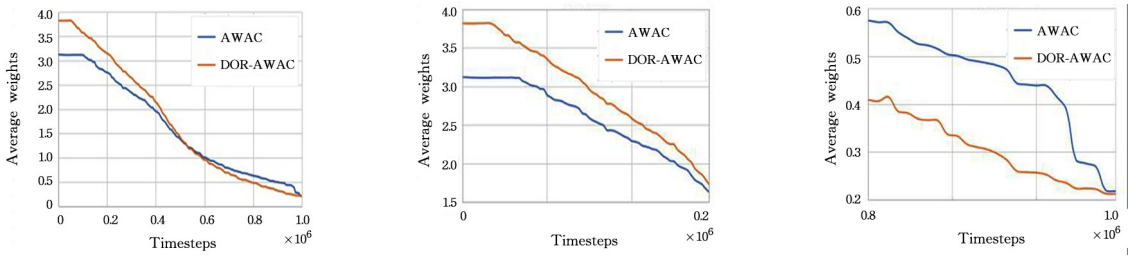
图 6 4 种方法在 4 个 D4RL Gym-MuJoCo 任务上的实验情况

Fig. 6 Experimental results of four approaches on four D4RL Gym-MuJoCo tasks

随着训练的进行，AWAC 算法的性能逐步提升，此时智能体与环境交互获得新的数据经验，对应的累积奖赏相对更大，相对价值更高，因而 DOR-AWAC 算法进一步加快了智能体学习改进策略的速度，使得算法在训练效果上超过 AWAC 算法。其中在 Hopper-v2 环境中，由于单腿机器人的特殊性，DOR-AWAC 算法收敛不及 AWAC 算法，BCQ 算法表现出不稳定、波动大的现象。在 Walker2d-v2 环境中，DOR-AWAC 算法表现最佳。DOR-AWAC 呈现出收敛速度快、收敛效果好的特点，而 AWAC-PER 算法表现出收敛过慢、波动极大的现象。

为了进一步说明算法对新经验的偏好性，本小节在 Walker2D-v2 任务上进行消融实验，比较 DOR-AWAC 算法与 AWAC 算法不同时间步经验在整个训练过程中实际被评论家回放的总权重。评估过程中，每 1000 步作为一段，对其中所有经验被回，放的权重求取平均值。为了更清楚地说明不同

算法对于旧经验及新经验回放权重的差异，再分别选取前 20 万步以及后 20 万步，同时每 200 步作为一段求取平均值。实验结果如图 7 所示。从图中可看出，随着经验生成时刻的增加，经验累积被回放的权重呈现衰减趋势，但 DOR-AWAC 算法训练相比采用 AWAC 算法训练，对生成时刻较早的老经验回放权重较少，且对生成时刻较晚的新经验回放权重更高，说明 DOR-AWAC 算法更加偏好新经验，各经验被回放权重和差异更小，对于不同时刻经验的训练更加平衡。



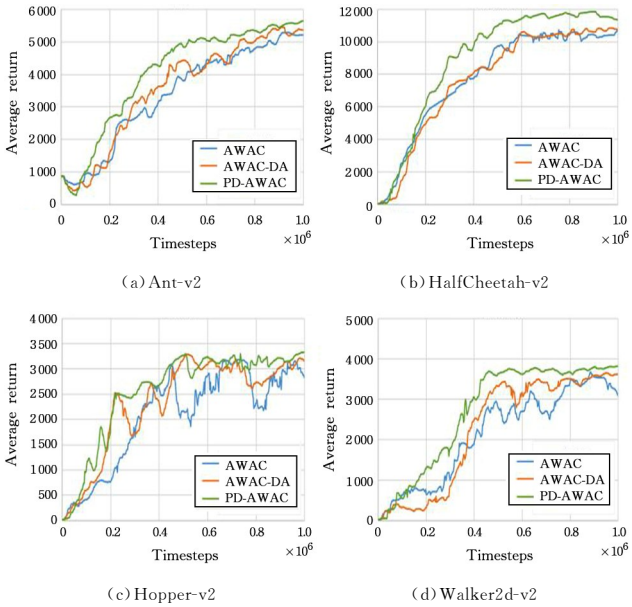
(a)整体经验累积被回放的权重示意图 (b)前 20 万步经验累积被回放的权重示意图 (c)后 20 万步经验累积被回放的权重示意图

图 7 两种方法在 Walker2D 任务上各经验的回放权重

Fig. 7 Replay weights of all experiences of two approaches on Walker2D

4.3.2 主从框架的双行动者-评论家实验和结果

本小节对采用主仆架构以及双行动者评论家模型的 PD-AWAC 算法，仅运用双行动者评论家模型的 AWAC-DA 以及 AWAC 算法进行实验对比，实验结果如图 8 所示。



(a) Ant-v2 (b) HalfCheetah-v2

(c) Hopper-v2 (d) Walker2d-v2

图 8 3 种方法在 4 个 D4RL Gym-MuJoCo 任务上的实验情况

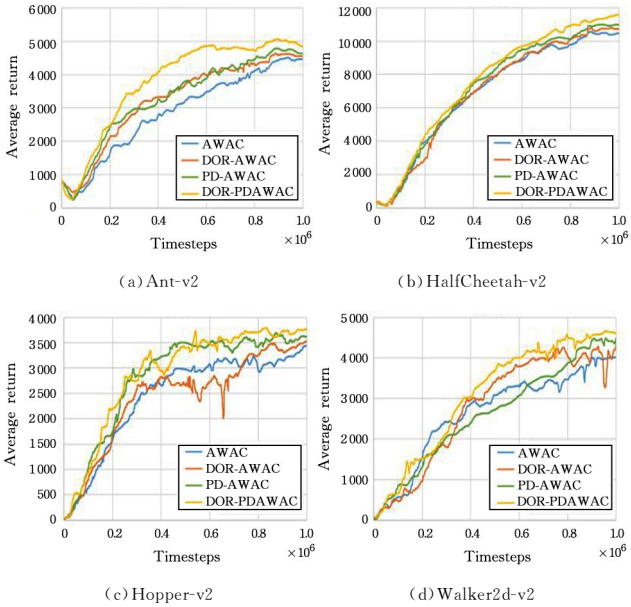
Fig. 8 Experimental results of four approaches on three D4RL Gym-MuJoCo tasks

从图中可以看出，在 Ant-v2 与 HalfCheetah-v2 环境中，PD-AWAC 算法的学习速度、收敛效果均更佳。与其他两种算法相比，AWAC-DA 算法由于没有采用策略蒸馏的 KMAA 主从框架，在这两个环境中的表现甚至不如 AWAC 算法，其行动者无法达到更高效率的协同。针对以上训练结果进行分析，在该环境中，运用双行动者-评论家模型的算法，能够借助双行动者实现比单个行动者更多的探索，同时借助策略蒸馏的主从框架，可以进一步加强两个行动者之间的协同合作，对高低估进行更好的权衡，实现更为有效的探索以及利用。

在 Hopper-v2 与 Walker2d-v2 环境中，PD-AWAC 算法与 AWAC-DA 的收敛效果最好，收敛速度也较佳。在 Hopper-v2 环境中的最终收敛方面，PD-AWAC 算法与 AWAC-DA 类似。在 Walker2d-v2 环境中，PD-AWAC 算法的收敛速度较其他算法最快。在训练所有阶段，PD-AWAC 算法的表现都比 AWAC 算法更优，进一步说明主从框架的双行动者-评论家模型的重要性。

4.3.3 数据集优选回放的双行动者-评论家模型实验和结果

上述部分分别对主从框架的双行动者-评论家模型以及数据优选回放机制进行了研究。为了进一步提升算法水平，本小节对将主从框架的双行动者-评论家模型与数据优选回放机制相结合的算法进行消融实验。其中，将只采用数据优选回放机制的 DOR-AWAC 算法与只采用了策略蒸馏的主从框架的双行动者-评论家的 PD-AWAC 算法、AWAC 算法和 DOR-PDAWAC 算法进行对比。可以看出，DOR-PDAWAC 算法性能表现更佳。



(a) Ant-v2 (b) HalfCheetah-v2

(c) Hopper-v2 (d) Walker2d-v2

图 9 4 种方法在 4 个 D4RL Gym-MuJoCo 任务上的实验情况

Fig. 9 Experimental results of four approaches on four D4RL Gym-MuJoCo tasks

由图 9 可以看出，在算法收敛时其在 4 个 D4RL Gym-Mu-

JoCo 任务中都取得了最佳收敛效果。进一步，根据训练中期算法的表现结果可知，DOR-PDAWAC 在 Ant-v2 与 Hopper-v2 环境中收敛速度表现较佳；而 DOR-AWAC 在 Walker2d-v2 与 HalfCheetah-v2 环境中，由于复杂度稍低，收敛速度表现较佳。整体而言，DOR-PDAWAC 算法相对最为优秀。

表 1 列出了 AWAC，DOR-AWAC，PDAWAC，DOR-PDAWAC 这 4 种算法在 3 个 D4RL Gym-MuJoCo 任务上的实验结果。从总计回报一行可以看出，与其他 3 种算法相比，DOR-PDAWAC 算法在指导智能体解决 3 种任务时表现更佳，且在其相对表现一般的任务上，大部分算法取得的平均回报也属于中上水平，说明了数据优选、双行动者-评论家模型以及策略蒸馏的主从框架几种改进相结合的优势。最终，DOR-PDAWAC 算法相较于 AWAC 算法效果提升约 18.9%。

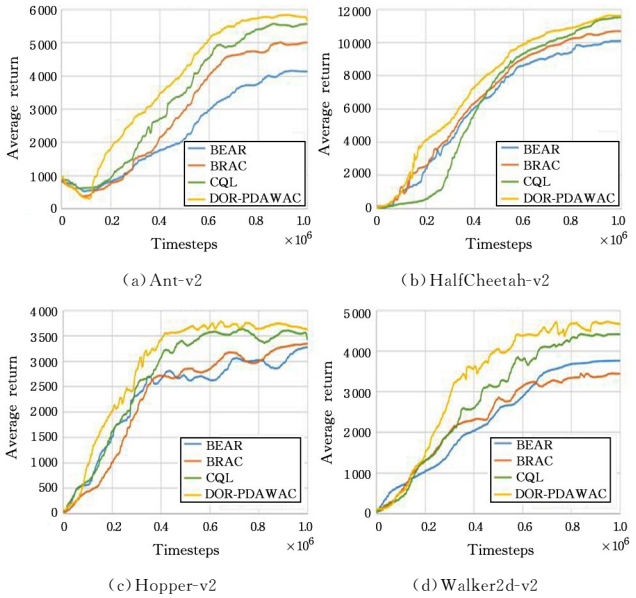
为了进一步证明 DOR-PDAWAC 算法的有效性，同时更清楚地展现其优势，在 4 个 D4RL Gym-MuJoCo 任务中增加了与经典离线强化学习 baseline 算法 BEAR，BRAC，CQL 的对比实验。

从图 10 可以看出，DOR-PDAWAC 算法相比其他经典离线强化学习算法有明显的性能优势。

表 1 各算法在 D4RL Gym-MuJoCo 任务上的实验结果

Table 1 Experimental results of each algorithm on D4RL Gym-MuJoCo

不同等级的数据集	任务名称	AWAC	DOR-AWAC	PDAWAC	DOR-PDAWAC
随机数据	HalfCheetah	32.3±0.2	33.6±1.2	**42.5±3.6**	38.4±2.1
	Hopper	11.5±0.1	**16.7±0.8**	15.6±0.3	14.8±0.2
	Walker2d	9.6±0.4	11.5±1.3	**14.2±0.4**	13.9±1.6
中级数据	HalfCheetah	45.8±0.7	46.7±1.9	53.2±1.2	**59.6±0.8**
	Hopper	59.2±0.7	62.3±0.3	69.7±0.6	**72.2±0.7**
	Walker2d	64.1±6.8	68.2±18.5	78.4±11.1	**86.7±10.2**
中级回放数据	HalfCheetah	52.7±2.3	51.8±11.2	**64.2±5.6**	63.5±3.8
	Hopper	35.8±4.5	39.7±1.3	44.6±1.3	**49.4±1.7**
	Walker2d	38.7±2.6	44.2±9.6	55.7±2.6	**59.9±2.9**
中级专家数据	HalfCheetah	48.1±5.6	52.5±4.3	62.7±8.0	**65.4±8.6**
	Hopper	102.5±13.4	108.2±11.5	112.6±12.4	**118.7±11.4**
	Walker2d	56.7±8.7	54.2±12.9	**60.2±4.7**	58.6±8.9
专家数据	HalfCheetah	85.2±0.7	84.7±4.2	92.3±9.5	**101.8±5.4**
	Hopper	117.5±2.1	121.7±0.2	126.9±0.4	**131.5±0.5**
	Walker2d	89.2±12.4	92.7±17.6	96.2±16.8	**108.7±12.2**
总计回报		877.2±61.4	888.7±96.8	989±78.5	**1043.1±71.0**



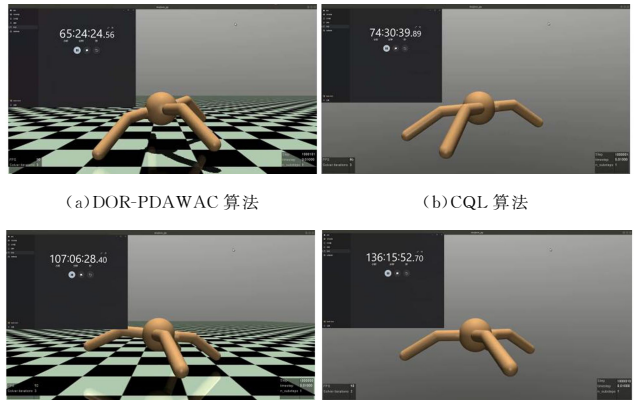
(a) Ant-v2 (b) HalfCheetah-v2

(c) Hopper-v2 (d) Walker2d-v2

图 10 4 种方法在 4 个 D4RL Gym-MuJoCo 任务上的实验情况

Fig. 10 Experimental results of four approaches on four D4RL Gym-MuJoCo tasks

为了进一步直观展现 DOR-PDAWAC 算法的性能优势，使用 Windows 10 系统自带的“时钟”软件计时，计算 4 种算法在 Ant-v2 环境下从开始运行至 100 万个情节所用的时长，结果如图 11 所示。从图中可以看出，DOR-PDAWAC 算法的效率相较而言更高。



(a) DOR-PDAWAC 算法 (b) CQL 算法

(c) BRAC 算法 (d) BEAR 算法

图 11 4 种方法在 Ant-v2 任务上进行 100 万步左右的耗时情况

Fig. 11 Time consumption of four methods running about 1 million steps on Ant-v2 tasks

结束语 本文在离线强化学习的 AWAC 算法上进行研究。针对该算法采用随机数据集采样机制、训练过程中新数据与旧数据期望训练次数不平衡、潜在具备更高价值的新数据被回放次数较少的问题，本文提出了数据优选回放的技术，并将其融入 AWAC 中，得到 DOR-AWAC 算法，并证实了其

相比 AWAC 训练速度更快，表现更佳。针对 AWAC 中评论家自举得到的目标值一致，且两者关联性依旧较强的问题，本文提出了双行动者-评论家模型，并将其融入 AWAC 算法中得到 DA-AWAC 算法。在额外引入一套行动者学习的同时，采取策略蒸馏方式使两套行动者相互协作，将其运用到主仆结构，加强协作，得到 PDAWAC 算法。为了进一步提升算法效果，本文将以上三点改进进行融合，得到 DOR-PDAWAC 算法。在 D4RL 数据集上 Gym-MuJoCo 环境中进行任务的实验表明，该算法相较于其他算法，总体表现更佳，算法学习速度更快，采样效率更高。

未来的研究工作包括选取更有效的模型以区分新旧经验，而非简单采用阈值进行划分。接下来将探究针对加入多套行动者-评论家后对算法效率的改进，评判策略的优劣，并且可以考虑选取更有效的评判标准。

参 考 文 献

[1] SUTTON R S, BARTO A G. Reinforcement learning: An introduction [M]. MIT Press, 2018.

[2] GOVINDARAJAN L N, LIU R G, LINSLEY D, et al. Diagnosing and exploiting the computational demands of videos games for deep reinforcement learning[J]. arXiv: 2309.13181, 2023.

[3] WU Q, SUN N, YANG T, et al. Deep Reinforcement Learning-Based Control for Asynchronous Motor-Actuated Triple Pendulum Crane Systems With Distributed Mass Payloads[J]. IEEE Transactions on Industrial Electronics, 2023, 71(2): 1853-1862.

[4] ZHOU X, WU L, ZHANG Y, et al. A robust deep reinforcement learning approach to driverless taxi dispatching under uncertain demand[J]. Information Sciences, 2023, 646: 119401.

[5] CHAI D, WU W, HAN Q, et al. Description Based Text Classification with Reinforcement Learning[C]// International Conference on Machine Learning. PMLR, 2020: 1371-1382.

[6] LI S, HU C, KE S, et al. LS-MolGen: Ligand-and-Structure Dual-Driven Deep Reinforcement Learning for Target-Specific Molecular Generation Improves Binding Affinity and Novelty [J]. Journal of Chemical Information and Modeling, 2023, 63(13): 4207-4215.

[7] LEVINE S, KUMAR A, TUCKER G, et al. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems[J]. arXiv: 2005.01643, 2020.

[8] LIU Q, ZHAI J W, ZHANG Z Z, et al. A survey on deep reinforcement learning [J]. Chinese Journal of Computers, 2018, 41(1): 1-27.

[9] SCHWEIGHOFER K, DINU M, RADLER A, et al. A Dataset Perspective on Offline Reinforcement Learning[C]// Conference on Lifelong Learning Agents. PMLR, 2022: 470-517.

[10] FUJIMOTO S, MEGER D, PRECUP D. Off-Policy Deep Reinforcement Learning without Exploration [C] // International Conference on Machine Learning. PMLR, 2019: 2052-2062.

[11] KUMAR A, FU J, TUCKER G, et al. Stabilizing off-policy Q-learning via bootstrapping error reduction[C]// Proceedings of the 33rd International Conference on Neural Information Processing Systems. 2019: 11784-11794.

[12] KUMAR A, ZHOU A, TUCKER G, et al. Conservative Q-Learning for Offline Reinforcement Learning[C]// Advances in Neural Information Processing Systems. 2020: 1179-1191.

[13] FUJIMOTO S, GU S. A Minimalist Approach to Offline Reinforcement Learning[C]// Advances in Neural Information Processing Systems. 2021: 20132-20145.

[14] NAIR A, GUPTA A, DALAL M, et al. Awac: Accelerating online reinforcement learning with offline datasets [J]. arXiv: 2006.09359, 2020.

[15] LUO Y, WANG Y, DONG K, et al. Relay hindsight experience replay: Continual reinforcement learning for robot manipulation tasks with sparse rewards[J]. arXiv: 2208.00843, 2022.

[16] LI J, YU T, ZHANG X, et al. Efficient experience replay based deep deterministic policy gradient for AGC dispatch in integrated energy system[J]. Applied Energy, 2021, 285: 116386.

[17] GAI S, WANG D, HE L. Offline Experience Replay for Continual Offline Reinforcement Learning [J]. arXiv: 2305.13804, 2023.

[18] WANG C, WU Y, VUONG Q, et al. Striving for simplicity and performance in off-policy DRL: Output normalization and non-uniform sampling [C] // International Conference on Machine Learning. PMLR, 2020: 10070-10080.

[19] SHI S M, LIU Q. Deep deterministic policy gradient with classified experience replay[J]. Automatica Sinica, 2022, 48(7): 1816-1823.

[20] BARTO A G, SUTTON R S, ANDERSON C W. Neuronlike adaptive elements that can solve difficult learning control problems[J]. IEEE Transactionson Systems, Man, And Cybernetics, 1983(5): 834-846.

[21] RUSU A A, COLMENAREJO S G, GULCEHRE C, et al. Policy distillation[J]. arXiv: 1511.06295, 2015.

[22] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[23] KONDA V R, TSITSIKLIS J N. Actor-citic agorithms[C]// Proceedings of the 12th International Conference on Neural Information Processing Systems. 1999: 1008-1014.

[24] CHEN D, ZHANG Q. Context-Aware Bayesian Network Actor-Critic Methods for Cooperative Multi-Agent Reinforcement Learning[J]. arXiv: 2306.01920, 2023.

[25] LAI K H, ZHA D, LI Y, et al. Dual policy distillation[J]. arXiv: 2006.04061, 2020.

[26] HONG Z W, NAGARAJAN P, MAEDA G. Periodic intra-ensemble knowledge distillation for reinforcement learning[C]// Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13－17, 2021, Proceedings, Part I 21. Springer International Publishing, 2021: 87-103.

[27] FEDUS W, RAMACHANDRAN P, AGARWAL R, et al. Revisiting fundamentals of experiencereplay [C] // International Conference on Machine Learning. PMLR, 2020: 3061-3071.

[28] ZHENG G, ZHOU S, BRAVERMAN V, et al. Selective experience replay compression using coresets for lifelong deep reinforcement learning in medical imaging[J]. arXiv: 2302.11510,

2023.
[29] PACKER C, ABBEEL P, GONZALEZ J E. Hindsight task relabelling: Experience replay for sparse reward meta-rl[J]. Advances in Neural Information Processing Systems, 2021, 34: 2466-2477.
[30] LI J, TANG C, TOMIZUKA M, et al. Hierarchical planning through goal-conditioned offline reinforcement learning[J]. IEEE Robotics and Automation Letters, 2022, 7(4): 10216-10223.
[31] CHEN X, GHADIRZADEH A, YU T, et al. Latent-variable advantage-weighted policy optimization for offline rl[J]. arXiv: 2203.08949, 2022.
[32] FU J, KUMAR A, NACHUM O, et al. D4rl: Datasets for deep data-driven reinforcement learning[J]. arXiv: 2004.07219, 2020.

YANG Haolin, born in 1999, postgraduate, is a member of CCF (No. J1794G). His main research interests include offline reinforcement learning and deep reinforcement learning.

LIU Quan, born in 1969, Ph.D, professor, Ph.D supervisor, is a member of CCF (No. 15231S). His main research interests include deep reinforcement learning and automated reasoning.

(责任编辑:杨雪敏)

王怀民院士:开源生态演变趋势及其影响

2024 年 9 月 25 日上午,作为 2024 世界计算大会论坛之一的"开源生态构建数字未来"主题研讨在长沙召开。CCF 会士、中国科学院院士、CCF 开源发展委员会主任教授王怀民教授发表了题为《开源生态演变趋势及其影响》的主题演讲。王怀民以"开源为什么能成功?开源创新为中国带来了什么?中国开源创新体系如何创新升级?"三个话题展开论述。

开源为什么能成功?

当前全球开源社区蓬勃发展,Github 注册开发者数量达到了惊人的 1.3 亿人,其中包括超过 900 万的中国开发者,是开源蓬勃发展重要标志。今天的商业公司也在积极拥抱开源,例如 GitHub 被微软以 75 亿美元收购,以及 Red Hat 被 IBM 以 340 亿美元并购等案例,都表明了开源商业模式的成功。随着互联网到来,开源作为一种新型的生产协作方式,通过一种新型的面向服务的产业生态,使得"Open Source"开源的理念获得成功。

开源创新为中国带来了什么?

王怀民强调,开源的商业逻辑是通过开源,以更低的成本吸引更多的"新潮"用户参与到新产品的成熟和传播之中,以寻求迅速从边缘低端产品变成主流高端产品。因而,开源是一种新型的生产关系。通过开放源代码,开发者可以自由地参与项目,从而形成自组织的群体协作模式,这有助于激发高水平的创新产品出现。当产业进入一个不确定性的阶段时,那些能够快速复制并广泛传播的技术或解决方案将占据优势。开源正是这样一种机制,它鼓励广泛的参与和贡献,使得创新的可能性得以最大化。我们现在把它总结为"群智范式",即通过集体智慧来推动技术创新和发展。这种方式超越了传统公司体制下组织软件开发的方式,展示了一种更灵活、更具包容性的合作模式。

王怀民表示,开源为中国提供了学习和借鉴的对象,让中国有机会参与到全球的开源生态中,并为未来在开源领域的领导地位提供了一个发展方向。从早期的学习 Unix 到后来的学习 Linux,中国逐渐融入了国际开源社区,并从中获得了宝贵的经验和技术。过去 20 年对国际开源生态发展贡献,无论在代码上还是在治理上中国的力量都是不可或缺的。

中国开源创新体系如何创新升级?

关于中国如何引领开源发展,王怀民通过以下几点详细阐述:1. 基础设施建设至关重要。中国正在推动开源基础设施的建设,如代码托管平台和群智范式,这些平台支持开发者并促进了根社区的形成,如 OpenEuler、OpenKylin 和 Open Harmony。这些根社区面向世界,为中国提供了自主发展的能力,并在云端操作系统和 AI OT 系统软件创新方面取得了进展。2. 政策支持与制度建设。中国政府首次在"十四五"规划中明确提到支持数字技术开源社区的发展,并完善相关知识产权和法律体系。同时,国家还在推进开源产业化、人才培养和供应链安全等方面的工作。3. 以开源平台抵御国际竞争。在当前复杂的国际竞争背景下,拥有自己的开源平台可以减少对外部平台的依赖,增强抵御外部制裁的能力。通过建立强大的开源项目和平台,中国可以在全球开源生态中占据更有利的位置。4. 社会力量的参与。社会组织在开源生态系统中扮演着重要角色,但在中国,这类组织的身份合法化问题仍然存在。王怀民呼吁进一步深化改革,使社会组织能够更有效地参与到科技创新中来,为创新活动提供社会服务。5. 综合推动机制。王怀民呼吁建立有为政府、有效市场和有机社会三者有机结合的机制,以促进开源生态的发展。这种机制有助于释放中国的计算资源和治理资源,构建一个更加开放、共享、分工合作的生态系统。6. 未来发展方向。展望未来,希望通过纵向算力基础设施和横向云际互联架构,构建一个开发运维一体的开源生态,提升中国在人工智能时代的竞争力。

王怀民在本次报告中强调了中国在开源领域的成就和发展方向,指出了政府、企业和社会各界共同努力的重要性,并提出了进一步推动开源创新的具体建议。开源是当今世界最具活力科技创新范式,王怀民呼吁所有热爱开源的人士积极参与进来,共同为中国乃至全世界的科技创新贡献力量。

据 CCF 微信公众号