·人工智能与模式识别·　　文章编号：1000-3428(2023)07-0118-07　　文献标志码：A　　中图分类号：TP181

远离旧区域和避免回路的强化探索方法

蔡丽娇[1]，秦　进[1]，陈　双[2]

（1.贵州大学 计算机科学与技术学院 公共大数据国家重点实验室，贵阳 550025；
2.贵州道坦坦科技股份有限公司，贵阳 550025）

摘　要：以内在动机为导向的探索类强化学习中，通常根据智能体对状态的熟悉程度产生内在奖励，难以获得较合适的近似度量方法，且这种长期累计度量的方式没有重视状态在其所处episode中的作用。Anchor方法使用锚代替分层强化学习中的子目标，鼓励智能体以远离锚的方式进行探索。受Anchor方法的启发，根据转移状态与同一个episode中历史状态之间的距离设计内在奖励函数，进而提出远离旧区域和避免回路的强化探索方法。将当前episode中部分历史状态组成的集合作为区域，周期性更新区域为最近访问的状态集合，根据转移状态与区域的最小距离给予智能体内在奖励，使智能体远离当前最近访问过的旧区域。将转移状态的连续前驱状态作为窗口并规定窗口大小，根据窗口范围内以转移状态为终点的最短回路长度给予内在奖励，防止智能体走回路。在经典的奖励稀疏环境MiniGrid中的实验结果表明，该方法避免了对状态熟悉程度的度量，同时以一个episode为周期对环境进行探索，有效提升了智能体的探索能力。

关键词：深度强化学习；奖励稀疏任务；内在奖励；旧区域；回路

开放科学（资源服务）标志码（OSID）：

源代码链接：https://github.com/cailijiao/AAAL-codes

中文引用格式：蔡丽娇，秦进，陈双.远离旧区域和避免回路的强化探索方法[J].计算机工程，2023，49(7)：118-124，134.

英文引用格式：CAI L J，QIN J，CHEN S.Reinforcement exploration method to keep away from old areas and avoid loops[J].Computer Engineering，2023，49(7)：118-124，134.

Reinforcement Exploration Method to Keep Away from Old Areas and Avoid Loops

CAI Lijiao[1]，QIN Jin[1]，CHEN Shuang[2]

（1.State Key Laboratory of Public Big Data，College of Computer Science and Technology，Guizhou University，Guiyang 550025，China；2.Guizhou Door To Time Science and Technology Co.，Ltd.，Guiyang 550025，China）

【Abstract】In intrinsic motivation-oriented exploratory reinforcement learning，intrinsic rewards are typically generated based on an agent's familiarity with the states. An appropriate approximate measure is difficult to obtain，and this long-term cumulative measure does not consider the role of the state in an episode.The Anchor method replaces subgoals in hierarchical reinforcement learning with anchors，thus encouraging the agent to explore in areas distant from the anchors. Inspired by this，an intrinsic reward function is designed based on the distance between the next state and the historical states in the same episode，and a reinforcement exploration method to keep Away from old Areas and Avoid Loops（AAAL）is proposed. Considering the set of partial historical states in this episode as a area and periodically treating the most recently visited state set as a new area，an intrinsic reward is allocated to the agent based on the shortest distance between the next state and area such that the agent is distant from the currently visited old area.Treating the successive precursor states of the next state as a window and specifying the window size，an intrinsic reward is allocated based on the shortest loop length of the window，with the next state regarded as the end point such that the agent can avoid walking the circuit.The experimental results in the classic reward sparse MiniGrid environment show that the AAAL method no longer requires measurements of familiarity with the states；in fact，it can explore the environment with an episode as a cycle and effectively improve the exploration ability of the agent.

【Key words】deep reinforcement learning；reward sparse task；intrinsic reward；old area；loop

DOI：10. 19678/j. issn. 1000-3428. 0065296

基金项目：贵州省科技计划项目（黔科合基础[2020]1Y275，黔科合支撑[2020]3Y004）。

作者简介：蔡丽娇（1995—），女，硕士研究生，主研方向为强化学习；秦　进（通信作者），副教授、博士；陈　双，高级工程师、硕士。

收稿日期：2022-07-20　　**修回日期：**2022-09-16　　**E-mail：**2391725628@qq.com

0　概述

强化学习是解决序列决策问题的试错学习方法。高效解决奖励稀疏任务是强化学习的重要方向，当前处理这类任务的主要方法有探索类强化学习和分层强化学习。

在探索类强化学习[1]中，除了epsilon-greedy、Boltzman、UCB等经典方法外，近些年兴起的以内在动机为导向的利用预测误差[2]、好奇心[3-4]、信息增益[5]的探索方法发展迅速。通常这类方法以内在奖励的形式附加给智能体与任务目标无直接联系的内在动机以实现对环境的探索，从而间接促进奖励稀疏任务的完成，因此内在奖励的设计通常是这类方法的核心。BURDA等[6]提出的RND方法分别从训练网络和固定网络中提取状态特征，然后将这两个特征的误差作为内在奖励。BELLEMARE等[7]提出的Count方法的内在奖励与下一个状态的访问计数成反比。这两种内在奖励的本质都是鼓励智能体访问不熟悉的环境区域。PARISI等[8]认为Count方法是以智能体为中心的，又在此基础上增加了以环境为中心的部分，即对一个转移引起的环境变化进行计数，鼓励智能体探索环境中一些有趣的目标，据此提出C-BET方法。虽然这些方法设计的内在奖励在奖励稀疏任务上取得了一定的效果，但由于这些方法需要度量智能体对状态的熟悉程度，因此设计合理的度量方法是一个难点，而且通常这种度量是贯穿整个训练周期的，不论是同一个episode或者不同episode中的相同状态出现的频率都会被累加，这就可能会使智能体错过那些在训练前期访问量过高但对完成任务有潜在帮助的状态，从而不选择那些可能具有连接作用的状态。

分层强化学习被认为在解决奖励稀疏任务以及复杂任务上具有巨大潜力。1992年，DAYAN等[9]将多层级划分任务的思想应用在强化学习中，如今分层强化学习受到了广泛的关注。以基于子目标的分层强化学习方法为例，自基于状态的值函数被扩展为同时基于状态和目标的值函数以来[10]，基于子目标的分层强化学习框架多数聚焦于子目标的确定问题且发展较快。从需要人工选择子目标的元控制器和控制器算法框架[11]到端到端的无需人工干预的管理者和员工算法框架[12]，基于子目标的分层强化学习方法越来越具推广意义。ANDRYCHOWICZ等[13]提出的HER以接近随机的方式重置经验元组中的目标作为已经到达过的目标，然后将更改后的经验元组用于训练离轨策略。PATERIA等[14]提出将子目标离散化以减少上层策略的搜索空间，以局部领域为界确定子目标，找出瓶颈状态。由此可见，自动生成合理的子目标是基于子目标的分层强化学习发展的趋势，但其仍面临巨大挑战。LI等[15]提出Anchor，也就是锚的概念，利用锚代替子目标，鼓励智能体以远离锚的方式进行探索。相比于其他方法，Anchor以更加自然、简单的方式自动生成子目标。

针对上述探索类强化学习面临的问题，本文受Anchor方法的启发，设计一个用于强化探索的内在奖励函数。该内在奖励函数无须对不同episode中的相同状态频率进行累计度量，而是仅利用当前episode的历史信息使智能体远离当前episode已访问过的旧区域，并且避免智能体走回路，以此提高智能体探索环境的能力。

1　相关工作

1.1　深度强化学习

强化学习[16]利用智能体与环境交互所得的经验进行策略更新，目标是最大化累计外在奖励，每个episode在t时刻的累计外在奖励如式(1)所示：

$$R_t=\sum_{k=0}^{\infty}\gamma^k r_{t+k} \tag{1}$$

其中：$\gamma\in(0,1]$为折扣因子；r_{t+k}为第$t+k$时刻的即时外在奖励。

深度强化学习[17]是强化学习与深度学习[18]的结合。深度强化学习利用深度网络强大的表征能力，将表格型强化学习拓展为近似型强化学习，使得强化学习能更好地被应用在具有连续状态或连续动作空间的任务上，这将强化学习的应用范围扩展至机器人控制、自动驾驶、视频游戏等领域[19-21]。随着任务越来越复杂，任务规模越来越大，强化学习面临训练效率低下的问题，而分布式强化学习的发展有效解决了该问题[22-24]。分布式强化学习一般会增加更多的环境，使更多的智能体同时工作，并且使多个episode同时执行。例如，ESPEHOLT等[25]提出的一个经过修正的分布式深度强化学习框架IMPALA，该框架使用多个Actor在多个环境中并行交互、收集经验，然后将经验传递给Learner用于策略更新。因此，该框架在大规模任务中具有较高的效率和较好的性能。

1.2　Anchor方法

现有基于子目标的分层强化学习存在子目标难以确定的问题，Anchor方法有效解决了该问题。Anchor方法以一个episode为周期计算内在奖励，每次从当前episode的历史状态中选择一个状态作为锚，并周期性地更新锚，用锚代替子目标。利用当前状态和对应锚的欧氏距离计算内在奖励，两者距离越大，内在奖励越大，从而鼓励智能体以远离锚的方式对环境进行探索。这与传统的基于子目标的分层强化学习不同，传统方法是使智能体接近子目标。相较而言，Anchor方法生成子目标的方式更加简单自然，并且在奖励稀疏任务上取得了一定的效果。

2 强化探索方法

为了鼓励智能体探索新的状态，当前大部分利用内在动机进行探索的方法都需要度量智能体对状态的熟悉程度，据此给予新状态更多的内在奖励。一方面设计合理的度量方法是一个难点；另一方面通常对状态熟悉程度的度量是贯穿整个训练周期的，有学者指出这可能会导致智能体不选择那些可能具有连接作用的动作，因此以一个episode为周期对状态的熟悉程度进行度量[26]。

本文提出的远离旧区域和避免回路（keep Away from old Areas and Avoid Loops，AAAL）的强化探索方法无须度量智能体对状态的熟悉程度，且对状态的探索是以一个episode为周期的，不会错过可能具有连接作用的动作。AAAL方法设计了一个由两部分构成的内在奖励函数：第一部分用来鼓励智能体远离当前最近访问过的一片旧区域而非一个锚，以此增大探索力度；第二部分用来避免智能体走回路，从而防止重复探索相同的区域。

2.1 远离旧区域的内在奖励

第一部分鼓励智能体远离当前最近访问过的一片区域。区域是由状态组成的集合，周期性更换区域为最近访问过的状态集合。从区域中找出离转移状态最近的状态，再根据转移状态和最近状态的距离计算内在奖励，从而使智能体远离该最近状态，最大程度地远离最近访问过的区域。

假设智能体在第i个episode访问如下状态：

$$S_{i,0},S_{i,1},\cdots,S_{i,j},S_{i,j+1},\cdots,S_{i,T_i}$$

其中：$S_{i,j}$表示第i个episode的第j个时刻的状态；最后一个状态的下标T_i表示第i个episode的总步数。

第i个episode的第j个时刻的内在奖励$r_{i,j}^{(1)}$的定义如下：

$$r_{i,j}^{(1)}=-\frac{1}{\min\|S_{i,j+1}-S_{i,k}\|+\eta}$$

$$\text{if } j<N,0\leqslant k\leqslant j$$

$$\text{if } j\geqslant N,\left(\left\lfloor\frac{j}{N}\right\rfloor-1\right)N\leqslant k\leqslant\left\lfloor\frac{j}{N}\right\rfloor N-1 \tag{2}$$

其中：$\eta>0$，是一个用作正则化的超参数；N是一个远小于T_i的正整数超参数；$\lfloor x\rfloor$表示不超过x的最大整数。由此得到：$-\frac{1}{\eta}\leqslant r_{i,j}^{(1)}<0$。

2.2 完整的内在奖励

第二部分的内在奖励的作用是避免智能体走回路。在此以一种滑动窗口的形式来避免回路。窗口长度是n，n是一个正整数，则窗口范围包含转移状态$S_{i,j+1}$的n个连续前驱状态$S_{i,j-n+1},S_{i,j-n+2},\cdots,S_{i,j}$。若前驱状态数量不足$n$，则将所有前驱状态包含在窗口内。

在每个时刻，从滑动窗口中寻找以转移状态为终点的最短回路并记录该回路长度为$n_{\min}$，据此计算该部分的内在奖励。特别地，当在窗口范围内不存在回路时，该部分内在奖励就等于0。内在奖励$r^{(2)}$定义如下：

$$r^{(2)}=\begin{cases}0, n_{\min}=0\\ \beta\alpha^{n_{\min}}, n_{\min}>0\end{cases} \tag{3}$$

其中：$\beta<0$，用来控制该部分内在奖励的范围；$0<\alpha\leqslant 1$，类似于折扣因子，用来调节回路长度$n_{\min}$对该部分内在奖励的影响，其值越小，内在奖励受回路长度的影响越大。由此得到：$\beta\alpha\leqslant r^{(2)}\leqslant 0$。

因此，AAAL方法完整的内在奖励$r_{i,j}^{\text{in}}$定义如下：

$$r_{i,j}^{\text{in}}=r_{i,j}^{(1)}+r^{(2)} \tag{4}$$

由此得到的总即时奖励$r_{i,j}^{s}$定义如下：

$$r_{i,j}^{s}=r_{i,j}+\theta^{\text{in}}r_{i,j}^{\text{in}} \tag{5}$$

其中：$r_{i,j}$为即时外在奖励；θ^{in}为权重系数，用来平衡外在奖励和内在奖励对策略更新的影响。

3 实验与结果分析

3.1 实验环境设置

为了验证AAAL方法的有效性，在迷宫类环境MiniGrid中进行测试。MiniGrid是一个经典的奖励稀疏强化学习环境，通常作为稀疏强化学习算法的基准测试环境。从该测试环境中选取Unlock、DoorKey-8x8、KeyCorridorS3R3、UnlockPickup、ObstructedMaze1Dlh、BlockedUnlockPickup、MultiRoom-N6、Obstructed Maze2Dlhb等8个难度不同的任务作为实验对象，其中后3个的难度较大。在所有任务环境中，每一个episode都会随机产生迷宫，房间布局、智能体和物体（钥匙、门、箱子、球）的位置以及颜色都会发生变化，智能体需要在迷宫中导航，与不同的物体交互，完成任务。图1给出了在8个任务环境中的实验截图。

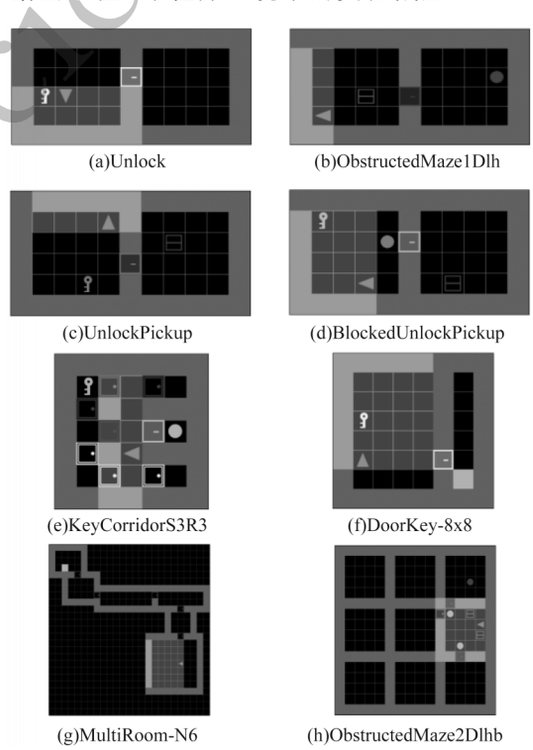

图1 MiniGrid环境

Fig.1 MiniGrid environments

所有任务的最大限定episode长度均遵循默认设置，如表1所示。

表1　不同任务的最大episode长度

Table 1　Maximum episode length for different tasks　　单位：步

任务	最大episode长度
Unlock	288
DoorKey-8x8	640
KeyCorridorS3R3	270
UnlockPickup	288
ObstructedMaze1Dlh	288
BlockedUnlockPickup	576
MultiRoom-N6	120
ObstructedMaze2Dlhb	576

将RND、Count、C-BET方法和不假定期望目标的Anchor版本的Anchor-2方法作为baseline。这4个baseline是最近几年较新的以内在动机为导向的探索类强化学习方法，与AAAL属于同一类方法，且性能较好。为了提高实验效率，所有方法都将IMPALA作为基本方法，在此将IMPALA作为一个baseline，记为ExtrinsicOnly，缩写为E-Only，该方法没有内在奖励，只使用外在奖励进行策略更新。

由于AAAL方法的内在奖励函数由两部分组成，第一部分用来远离最近访问过的区域，第二部分用来避免回路，因此会增加一部分消融实验。首先单独使用第一部分作为内在奖励说明其探索作用，称为AAAL_cut方法，然后在此基础上加上第二部分说明避免回路的正向作用，称为完整的AAAL方法。

在所有方法中，源于IMPALA的参数设置均相同。由于参数数量较多，不一一叙述，仅对一些特有、与内在奖励相关的参数进行说明。参照各方法在该实验环境中已有的参数设置，将RND中内在奖励的权重系数θ^{in}设置为0.1，其他方法设置为0.005。Anchor-2方法中锚的更新周期与AAAL和AAAL_cut方法中区域的更新周期N的作用类似，因此在各任务中两者设置相同，在KeyCorridorS3R3中设置为20，在MultiRoom-N6中设置为30，在其他任务中设置为10，η设置为1。AAAL方法中特有的参数β在Unlock、DoorKey-8x8、KeyCorridorS3R3和MultiRoom-N6中均设置为−0.4，在其他任务中均设置为−0.1；α在所有任务中均设置为0.8；滑动窗口的长度n在所有任务中均设置为10。

3.2　结果分析

平均累计外在奖励和收敛速度是决定深度强化学习性能的重要指标。平均累计外在奖励越大，收敛速度越快，算法性能越好。为了使对比更加清晰，将所有方法在所有任务上的实验结果分别进行绘制，图2~图9展示了7种方法（包括消融实验AAAL_cut和AAAL方法）在8个任务中所获得的平均累计外在奖励的动态变化情况。

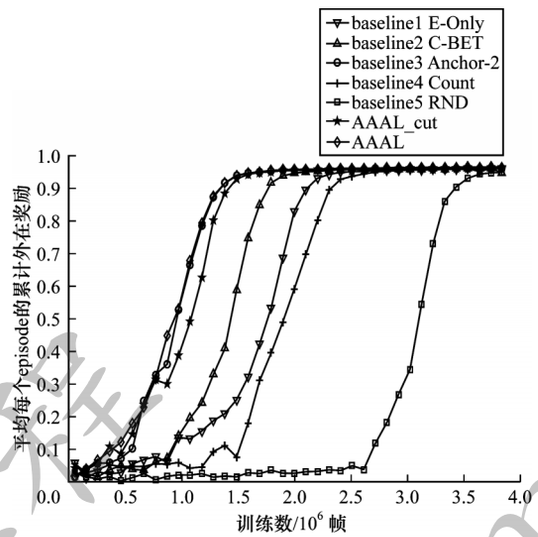

图2　7种方法在Unlock上的平均奖励

Fig.2　Average rewards of seven methods on Unlock

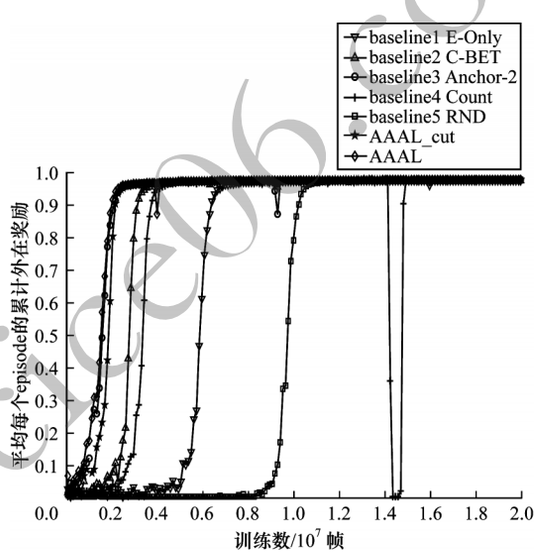

图3　7种方法在DoorKey-8x8上的平均奖励

Fig.3　Average rewards of seven methods on DoorKey-8x8

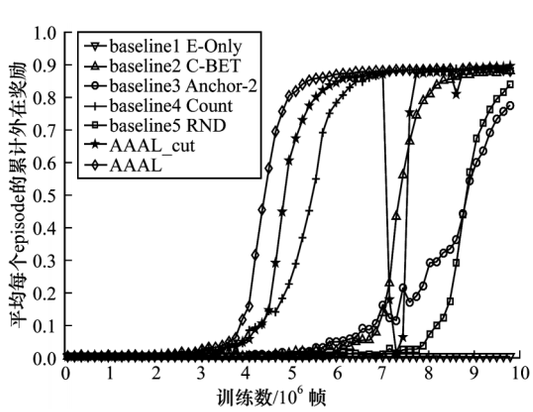

图4　7种方法在KeyCorridorS3R3上的平均奖励

Fig.4　Average rewards of seven methods on KeyCorridorS3R3

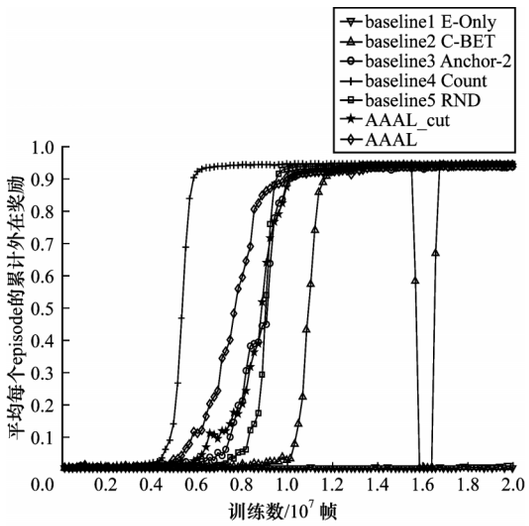

图5 7种方法在UnlockPickup上的平均奖励

Fig.5 Average rewards of seven methods on UnlockPickup

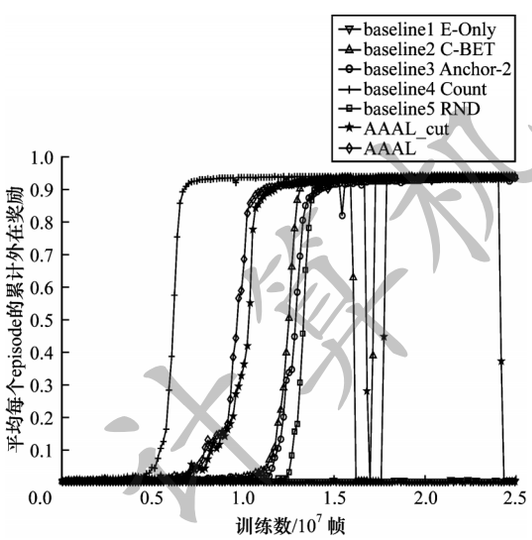

图6 7种方法在ObstructedMaze1Dlh上的平均奖励

Fig.6 Average rewards of seven methods on ObstructedMaze1Dlh

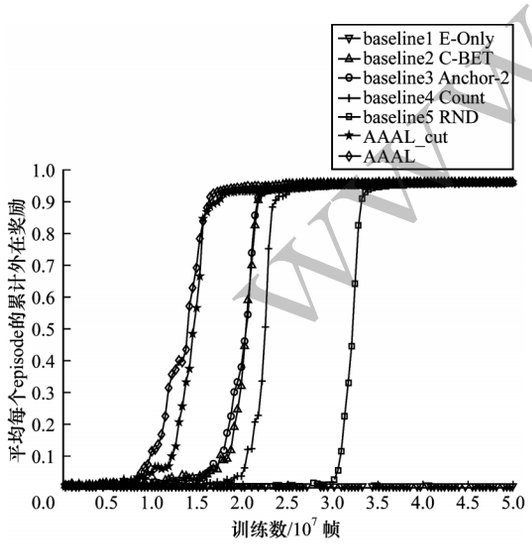

图7 7种方法在BlockedUnlockPickup上的平均奖励

Fig.7 Average rewards of seven methods on BlockedUnlockPickup

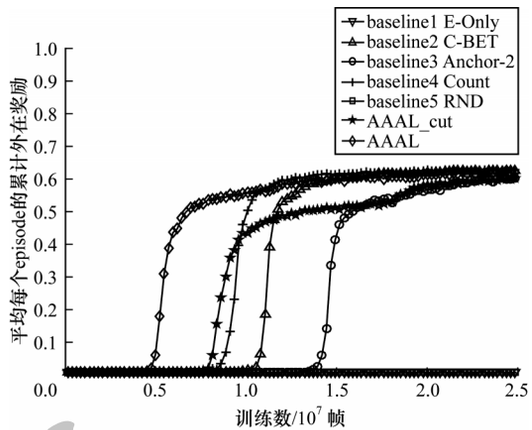

图8 7种方法在MultiRoom-N6上的平均奖励

Fig.8 Average rewards of seven methods on MultiRoom-N6

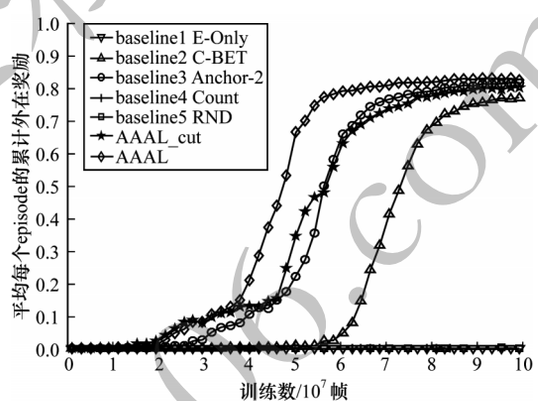

图9 7种方法在ObstructedMaze2Dlhb上的平均奖励

Fig.9 Average rewards of seven methods on ObstructedMaze2Dlhb

观察图2~图9的实验结果可知，Unlock和DoorKey-8x8是两个非常简单的任务，即使是仅使用外在奖励进行策略更新的方法E-Only也能很快探索到奖励，并且快速收敛。在这两个任务中，与5种baseline相比，AAAL与Anchor-2方法并列最优，看不出明显的差异，表现相当。这两种方法在Unlock和DoorKey-8x8中分别不到1.5×10^6帧、2.5×10^6帧就收敛，相比其他任务，使用的训练帧数最少，这会导致AAAL方法不能及时学习到一个完整的探索策略，在一定程度上限制了AAAL方法的性能表现。在KeyCorridorS3R3、BlockedUnlockPickup、ObstructedMaze2Dlhb等3个任务中，AAAL方法明显优于其他方法，取得了较好的结果。在MultiRoom-N6任务中，AAAL方法在前期的表现最好，较早就探索到了奖励，远优于其他方法，但在后期的收敛速度较慢，与表现最好的Count方法相比略差。在UnlockPickup任务中，Count方法表现最好，AAAL在前期优于RND方法，后期较之略差。在ObstructedMaze1Dlh中，Count方法也是表现最好的方法，AAAL仅次于Count方法。整体而言，AAAL

方法在8个任务中的表现较好，这说明了AAAL方法在一些简单或困难的奖励稀疏任务中均具有较好的探索性能，从而高效地完成任务。

AAAL方法在AAAL_cut方法的基础上加入了避免回路的内在奖励部分。与AAAL_cut方法相比，AAAL方法在所有任务中都有不同程度的性能提升，其中在KeyCorridorS3R3、UnlockPickup、MultiRoom-N6、ObstructedMaze2Dlhb等4个任务中的提升较明显，这说明了避免回路有利于提升探索的有效性，进而加快探索速度。

为了进一步定量比较这些方法，表2对每种方法在每个任务的训练过程中获得的平均累计外在奖励以及算法收敛时最终获得的外在奖励进行统计，其中最优指标值用加粗字体标示。

表2　平均奖励和最终奖励对比

Table 2　Comparison of average rewards and final rewards

任务	奖励	E-Only	C-BET	Anchor-2	Count	RND	AAAL_cut	AAAL
Unlock	Mean	0.561	0.628	0.738	0.511	0.216	0.721	**0.745**
	Final	0.960	0.960	0.961	0.960	0.950	0.961	**0.964**
DoorKey-8x8	Mean	0.700	0.845	0.901	0.793	0.505	0.891	**0.906**
	Final	0.974	0.974	**0.976**	**0.976**	**0.976**	0.975	**0.976**
KeyCorridorS3R3	Mean	0.000	0.227	0.129	0.416	0.098	0.417	**0.491**
	Final	0.000	0.882	0.776	0.884	0.844	**0.897**	0.892
UnlockPickup	Mean	0.002	0.395	0.533	**0.699**	0.526	0.540	0.587
	Final	0.014	0.944	0.938	**0.946**	0.945	0.942	0.940
ObstructedMaze1Dlh	Mean	0.001	0.458	0.478	**0.723**	0.461	0.508	0.595
	Final	0.000	**0.943**	0.937	**0.943**	0.940	0.937	0.939
BlockedUnlockPickup	Mean	0.000	0.581	0.580	0.535	0.346	0.685	**0.707**
	Final	0.000	0.961	0.960	**0.963**	0.959	0.961	0.961
MultiRoom-N6	Mean	0.000	0.345	0.247	0.390	0.000	0.355	**0.462**
	Final	0.000	**0.629**	0.605	**0.629**	0.000	0.603	0.621
ObstructedMaze2Dlhb	Mean	0.000	0.228	0.379	0.000	0.001	0.391	**0.464**
	Final	0.000	0.776	0.819	0.000	0.000	**0.823**	0.812

由表2所示，与5种baseline相比，AAAL方法在6个任务中都获得了最高的平均累计外在奖励。在Unlock和DoorKey-8x8这2个任务中，AAAL方法获得的平均累计外在奖励比Anchor-2方法略高，分别提高了0.007和0.005，这与图2和图3描述的情况相符，说明即使在简单的任务中，AAAL方法也有较好的表现。在KeyCorridorS3R3、BlockedUnlockPickup、MultiRoom-N6、ObstructedMaze2Dlhb等4个任务中，AAAL方法获得的平均累计外在奖励明显高于其他方法。虽然图8显示AAAL方法在MultiRoom-N6后期的表现略逊于Count方法，但由于其在训练前期约5.0×10^6帧时就探索到了奖励，而Count方法在训练约1.0×10^7帧时才探索到奖励，因此AAAL方法的平均累计外在奖励明显比Count方法高。在UnlockPickup和ObstructedMaze1Dlh这2个任务中，Count方法获得了最高的平均累计外在奖励，AAAL方法仅次于该方法。与AAAL_cut方法相比，AAAL方法在所有8个任务中都获得了较高的平均累计外在奖励，依次比AAAL_cut方法增加了0.024、0.015、0.074、0.047、0.087、0.022、0.107和0.073，这也说明了避免回路有利于提升探索能力，进而提高策略性能。总体而言，AAAL方法在8个任务中都具有较好的探索性能，获得了非常高的平均累计外在奖励。另外，经观察可发现，在所有方法中，在各任务上有效的方法在收敛时获得的最终奖励都处于相同水平，主要差别在于收敛速度。

当episode的长度小于任务的最大限定步数时，每个episode的长度就是智能体本次完成任务需要使用的步数，也就是智能体找到的最优路径的长度，因此episode的长度在一定程度上也能体现方法性能的优劣。表3对各方法在每个任务训练过程中的平均episode长度进行统计，其中最优指标值用加粗字体标示。由表3可以看出，与5种baseline相比，AAAL方法在6个任务中的平均episode长度都是最短的。AAAL方法在Unlock、DoorKey-8x8、KeyCorridorS3R3、BlockedUnlockPickup、MultiRoom-N6、ObstructedMaze2Dlhb这6个任务中的平均episode长度分别比在这些任务中表现最好的baseline少了2、3、22、73、7和50步，其中，AAAL方法在KeyCorridorS3R3、Blocked UnlockPickup、ObstructedMaze2Dlhb这3个任务中的平均episode长度明显最短。在UnlockPickup和ObstructedMaze1Dlh

这2个任务中，Count方法在训练过程中的平均episode长度最短，明显优于其他方法，AAAL方法仅次于Count方法。与AAAL_cut方法相比，AAAL方法在8个任务中的平均episode长度更短，依次比AAAL_cut方法少了7、9、22、13、25、13、12和43步，这说明避免回路对智能体的探索能力的提升具有正向作用，进一步提高了策略性能。总体而言，使用了AAAL方法的智能体在训练过程中能找到更短的路径，从而更快地完成任务。

表3 平均episode长度对比

Table 3 Comparison of average episode length

单位：步

任务	E-Only	C-BET	Anchor-2	Count	RND	AAAL_cut	AAAL
Unlock	129	110	78	144	228	83	**76**
DoorKey-8x8	198	103	66	135	319	72	**63**
KeyCorridorS3R3	270	211	237	161	245	161	**139**
UnlockPickup	287	176	136	**88**	138	134	121
ObstructedMaze1Dlh	288	158	152	**82**	157	144	119
BlockedUnlockPickup	576	245	245	270	379	185	**172**
MultiRoom-N6	120	82	93	76	120	81	**69**
ObstructedMaze2Dlhb	576	450	364	576	575	357	**314**

4 结束语

本文设计一个内在奖励函数，进而提出基于该内在奖励函数的AAAL强化探索方法，在提升智能体探索力度的同时避免智能体走回路。实验结果表明，相比于同类方法，该方法在奖励稀疏环境MiniGrid中的大部分任务中性能更好。下一步将利用当前的累计外在奖励、内在奖励、最短路径长度等信息动态调整区域更换周期和滑动窗口长度，使AAAL方法内在奖励函数的区域更换周期和滑动窗口长度能够以一种自适应的方式用于策略更新，进一步提高智能体的探索能力。

参考文献

[1] HAO J Y, YANG T P, TANG H Y, et al. Exploration in deep reinforcement learning: from single-agent to multiagent domain[EB/OL]. [2022-06-04]. https://arxiv.org/abs/2109.06668.

[2] PATHAK D, AGRAWAL P, EFROS A A, et al. Curiosity-driven exploration by self-supervised prediction[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops. Washington D. C., USA: IEEE Press, 2017: 488-489.

[3] BADIA A P, SPRECHMANN P, VITVITSKYI A, et al. Never give up: learning directed exploration strategies[EB/OL]. [2022-06-04]. https://arxiv.org/abs/2002.06038.

[4] ZHANG T J, XU H Z, WANG X L, et al. NovelD: a simple yet effective exploration criterion[EB/OL]. [2022-06-04]. https://openreview.net/forum?id=CYUzpnOkFJp.

[5] HOUTHOOFT R, CHEN X, DUAN Y, et al. VIME: variational information maximizing exploration[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1605.09674.

[6] BURDA Y, EDWARDS H, STORKEY A, et al. Exploration by random network distillation[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1810.12894.

[7] BELLEMARE M G, SRINIVASAN S, OSTROVSKI G, et al. Unifying count-based exploration and intrinsic motivation[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2016: 1479-1487.

[8] PARISI S, DEAN V, PATHAK D, et al. Interesting object, curious agent: learning task-agnostic exploration[EB/OL]. [2022-06-04]. https://arxiv.org/abs/2111.13119.

[9] DAYAN P, HINTON G E. Feudal reinforcement learning[C]//Proceedings of NIPS'92. New York, USA: ACM Press, 1992: 271-278.

[10] SCHAUL T, HORGAN D, GREGOR K, et al. Universal value function approximators[C]//Proceedings of the 32nd International Conference on Machine Learning. New York, USA: ACM Press, 2015: 1312-1320.

[11] KULKARNI T D, NARASIMHAN K R, SAEEDI A, et al. Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1604.06057.

[12] VEZHNEVETS A S, OSINDERO S, SCHAUL T, et al. FeUdal networks for hierarchical reinforcement learning[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1703.01161.

[13] ANDRYCHOWICZ M, WOLSKI F, RAY A, et al. Hindsight experience replay[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1707.01495.

[14] PATERIA S, SUBAGDJA B, TAN A H, et al. End-to-end hierarchical reinforcement learning with integrated subgoal discovery[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33(12): 7778-7790.

[15] LI R J, CAI Z L, HUANG T Y, et al. Anchor: the achieved goal to replace the subgoal for hierarchical reinforcement learning[J]. Knowledge-Based Systems, 2021, 225: 107128.

[16] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[M]. London, UK: MIT Press, 2018.

[17] 刘全，翟建伟，章宗长，等. 深度强化学习综述[J]. 计算机学报，2018，41(1)：1-27.
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. (in Chinese)

[18] MINAR M R, NAHER J. Recent advances in deep learning: an overview[EB/OL]. [2022-06-04]. https://arxiv.org/abs/1807.08169.

（下转第134页）

（上接第124页）

[19] LEVINE S, FINN C, DARRELL T, et al. End-to-end training of deep visuomotor policies[J]. Journal of Machine Learning Research, 2016, 17: 39: 1-40.
[20] ARADI S. Survey of deep reinforcement learning for motion planning of autonomous vehicles [J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(2): 740-759.
[21] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.
[22] BABAEIZADEH M, FROSIO I, TYREE S, et al. Reinforcement learning through asynchronous advantage actor-critic on a GPU [EB/OL]. [2022-06-04]. https://arxiv. org/abs/1611. 06256.
[23] ESPEHOLT L, MARINIER R, STANCZYK P, et al. SEED RL: scalable and efficient deep-RL with accelerated central inference[EB/OL]. [2022-06-04]. https://arxiv. org/abs/1910. 06591.
[24] WIJMANS E, KADIAN A, MORCOS A, et al. DD-PPO: learning near-perfect PointGoal navigators from 2. 5 billion frames [EB/OL]. [2022-06-04]. https://arxiv. org/abs/1911. 00357.
[25] ESPEHOLT L, SOYER H, MUNOS R, et al. IMPALA: scalable distributed deep-RL with importance weighted actor-learner architectures[EB/OL]. [2022-06-04]. https://arxiv. org/abs/1802. 01561.
[26] STANTON C, CLUNE J. Deep curiosity search: intra-life exploration can improve performance on challenging deep reinforcement learning problems[EB/OL]. [2022-06-04]. https://arxiv. org/abs/1806. 00553.

编辑 陆燕菲