软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2023,34(2):733−760 [doi: 10.13328/j.cnki.jos.006706]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

深度分层强化学习研究与发展*

黄志刚[1], 刘 全[1,2,3,4], 张立华[1], 曹家庆[1], 朱 斐[1,2,3,4]

[1](苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

[2](江苏省计算机信息处理技术重点实验室(苏州大学), 江苏 苏州 215006)

[3](符号计算与知识工程教育部重点实验室(吉林大学), 吉林 长春 130012)

[4](软件新技术与产业化协同创新中心(南京), 江苏 南京 210093)

通信作者: 刘全, E-mail: quanliu@suda.edu.cn

摘 要: 深度分层强化学习是深度强化学习领域的一个重要研究方向, 它重点关注经典深度强化学习难以解决的稀疏奖励、顺序决策和弱迁移能力等问题. 其核心思想在于: 根据分层思想构建具有多层结构的强化学习策略, 运用时序抽象表达方法组合时间细粒度的下层动作, 学习时间粗粒度的、有语义的上层动作, 将复杂问题分解为数个简单问题进行求解. 近年来, 随着研究的深入, 深度分层强化学习方法已经取得了实质性的突破, 且被应用于视觉导航、自然语言处理、推荐系统和视频描述生成等生活领域. 首先介绍了分层强化学习的理论基础; 然后描述了深度分层强化学习的核心技术, 包括分层抽象技术和常用实验环境; 详细分析了基于技能的深度分层强化学习框架和基于子目标的深度分层强化学习框架, 对比了各类算法的研究现状和发展趋势; 接下来介绍了深度分层强化学习在多个现实生活领域中的应用; 最后, 对深度分层强化学习进行了展望和总结.

关键词: 人工智能; 强化学习; 深度强化学习; 半马尔可夫决策过程; 深度分层强化学习

中图法分类号: TP18

中文引用格式: 黄志刚, 刘全, 张立华, 曹家庆, 朱斐. 深度分层强化学习研究与发展. 软件学报, 2023, 34(2): 733–760. http://www.jos.org.cn/1000-9825/6706.htm

英文引用格式: Huang ZG, Liu Q, Zhang LH, Cao JQ, Zhu F. Research and Development on Deep Hierarchical Reinforcement Learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(2): 733–760 (in Chinese). http://www.jos.org.cn/1000-9825/6706.htm

Research and Development on Deep Hierarchical Reinforcement Learning

HUANG Zhi-Gang[1], LIU Quan[1,2,3,4], ZHANG Li-Hua[1], CAO Jia-Qing[1], ZHU Fei[1,2,3,4]

[1](School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

[2](Jiangsu Key Laboratory for Computer Information Processing Technology (Soochow University), Suzhou 215006, China)

[3](Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University), Changchun 130012, China)

[4](Collaborative Innovation Center of Novel Software Technology and Industrialization (Nanjing), Nanjing 210093, China)

Abstract: Deep hierarchical reinforcement learning (DHRL) is an important research field in deep reinforcement learning (DRL). It focuses on sparse reward, sequential decision, and weak transfer ability problems, which are difficult to be solved by classic DRL. DHRL decomposes complex problems and constructs a multi-layered structure for DRL strategies based on hierarchical thinking. By using temporal abstraction, DHRL combines lower-level actions to learn semantic higher-level actions. In recent years, with the development of research, DHRL has been able to make breakthroughs in many domains and shows a strong performance. It has been applied to visual

* 基金项目: 国家自然科学基金(61772355, 61702055, 61876217, 62176175); 江苏省高等学校自然科学研究重大项目(18KJA520011, 17KJA520004); 吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04, 93K172017K18, 93K172021K08); 苏州市应用基础研究计划工业部分(SYG201422); 江苏高校优势学科建设工程资助项目

收稿时间: 2021-08-02; 修改时间: 2021-11-28, 2022-03-30; 采用时间: 2022-05-17; jos 在线出版时间: 2022-07-22

navigation, natural language processing, recommendation system and video description generation fields in real world. In this study, the theoretical basis of hierarchical reinforcement learning (HRL) is firstly introduced. Secondly, the key technologies of DHRL are described, including hierarchical abstraction techniques and common experimental environments. Thirdly, taking the option-based deep hierarchical reinforcement learning framework (O-DHRL) and the subgoal-based deep hierarchical reinforcement learning framework (G-DHRL) as the main research objects, those research status and development trend of various algorithms are analyzed and compared in detail. In addition, a number of DHRL applications in real world are discussed. Finally, DHRL is prospected and summarized.

Key words: artificial intelligence; reinforcement learning; deep reinforcement learning; semi-Markov decision process; deep hierarchical reinforcement learning

强化学习(reinforcement learning, RL)是机器学习领域的一个重要分支, 它以马尔可夫决策过程(Markov decision process, MDP)为理论基础, 是一种交互式学习方法[1]. 深度强化学习(deep reinforcement learning, DRL)作为深度学习(deep learning, DL)[2]和 RL 的结合算法, 同时具备了 DL 的感知能力和 RL 的决策能力, 初步形成从输入原始数据到输出动作控制的完整智能系统. 近些年, 刘全等人[3]对 DRL 进行了全面的分析和解读, 总结了深度 Q 网络(deep q-learning network, DQN)[4]、深度确定性策略梯度(deep deterministic policy gradient, DDPG)[5]和异步行动者-评论家(asynchronous advantage actor-critic, A3C)[6]等经典算法, 并介绍了多种前沿研究方向.

分层强化学习(hierarchical reinforcement learning, HRL)[7]作为RL的重要分支, 与经典RL方法的最大区别在于: 它以半马尔可夫决策过程(semi-Markov decision process, SMDP)[8]为理论基础, 基于分层抽象技术, 从结构上对 RL 进行改进, 重点关注 RL 难以解决的稀疏奖励、顺序决策和弱迁移能力等问题, 实现了更强的探索能力和迁移能力. 但是 HRL 仍然存在计算能力不足、无法对状态特征进行高效表达的问题, 通常只能处理离散状态-动作空间任务. 在 DRL 的成功应用后, 深度分层强化学习(deep hierarchical reinforcement learning, DHRL)[9]同样将 DL 方法引入 HRL 框架, 不仅从理论层面对 HRL 进行了拓展, 还利用深度网络实现了更强的特征提取能力和策略学习能力, 构建了更有效、更灵活的分层结构, 可以有效解决更复杂的任务[10]. 随着 DHRL 理论的发展和完善, 逐步形成了以下层策略学习基础任务实现能力、上层策略学习下游任务解决方案的问题求解路线. 目前, DHRL 已被广泛应用于视觉导航[11]、自然语言处理[12]、推荐系统[13]和视频描述生成[14]等真实世界应用领域.

为了对 DHRL 进行系统的分析和总结, 我们首先在中国计算机学会推荐的国际学术会议和期刊以及 CNKI 的论文数据库中, 以“hierarchicalre inforcement learning”“optionre inforcement learning”和“subgoalre inforcement learning”等关键词进行检索, 并在谷歌学术中, 将被引次数超过 500 的核心论文[1,15−17]作为基准, 检索引用了这些论文的 HRL 和 DHRL 论文; 然后, 通过人工审查方式对已检索的论文进行筛选, 排除与研究问题无关和已被收录的网络论文. 我们用图 1 和图 2 对所筛选论文进行展示. 图 1 反映了从 1998 年(HRL 理论基础被提出的年份[8])至 2021 年(截止到 2021 年 6 月), 在各类会议、期刊和网络上较有影响力的 HRL 和 DHRL 相关论文的数量及刊载情况, 它们中的绝大多数都被收录于 CCFA 类会议(112 篇)、CCFB 类会议(25 篇)、SCI 一区期刊(7 篇)和 SCI 二区期刊(22 篇). 图 2 反映了从 1998–2021 年(截止到 2021 年 6 月), HRL 和 DHRL 相关论文的被引次数. 从图 1 和图 2 可以看出: 一方面, HRL 与 DHRL 的研究热度逐年增加, 尤其是在 2016 之后, 随着 DL 的发展和 DRL 的出现, 国内外学者对 DHRL 的关注程度与日俱增; 另一方面, 相关论文的被引次数在 1999 年和 2017 年出现高峰, 这与 RL 奠基工作的开展和 DRL 的飞跃式发展有着密切关系.

本文以 HRL 基础理论为研究脉络, 重点关注 DHRL 的研究现状和发展趋势.

第 1 节对 HRL 的基础理论进行介绍. 第 2 节描述 DHRL 的核心技术, 包括可以解决的问题、常用实验环境和 DHRL 主流框架的划分依据. 第 3 节、第 4 节分析两种 DHRL 框架下的核心算法, 详细说明各类算法的发展历程、研究重心和优缺点. 第 5 节对 DHRL 在现实生活领域中的应用进行介绍. 第 6 节、第 7 节对 DHRL 进行展望和总结.

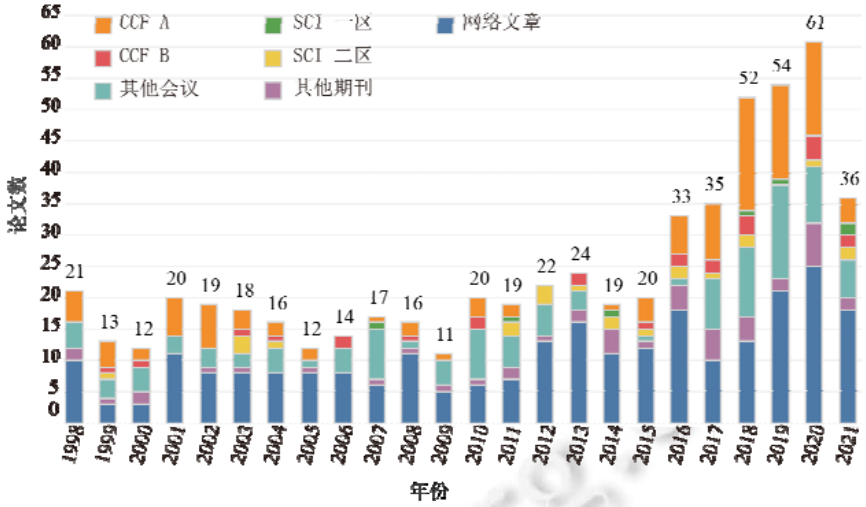

图 1 HRL 和 DHRL 论文的发表数量及刊载情况

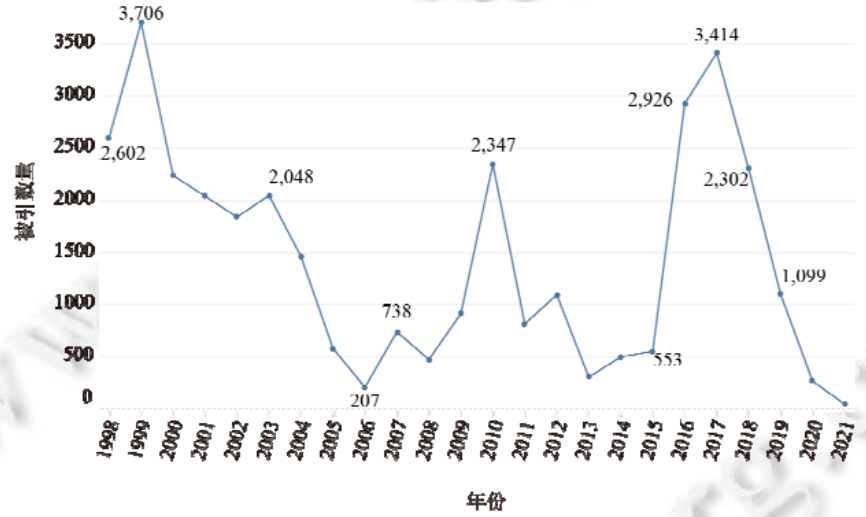

图 2 HRL 和 DHRL 论文的被引次数

1 预备知识

1.1 马尔可夫决策过程

MDP 作为 RL 的理论基础, 以四元组$(\mathcal{S},\mathcal{A},P,\mathcal{R})$来表示状态空间、动作空间、状态转移概率函数和奖励函数, 对状态 s、动作 a、状态转移概率 p 和奖励 r 进行抽象表达, 学习最优策略π^*, 获取最大累计奖励.

策略π表示状态到动作的映射函数$\pi:\mathcal{S}\to\mathcal{A}$, 引导智能体进行动作选择. 根据映射的概率分布形式, 策略可以分为确定性策略 $a=\pi(s)$和随机性策略 $a\sim\pi(a|s)$. 当给定一个策略π时, 智能体会依据该策略与环境进行交互, 得到一个状态-动作序列$(s_0,a_0,s_1,a_1,\ldots,s_T)$. MDP 将回报值 $G_t=r_{t+1}+r_{t+2}+\ldots+r_T$ 定义为智能体从 t 时刻状态 s_t 开始, 到终止状态 s_T 所得到的累计奖励.

对于确定性环境, 可以直接使用累计奖励 G_t 作为策略优劣的评判指标. 但在实际情况中, 由于状态转移概率p的存在, 环境往往具有随机性, 智能体从当前状态到达终止状态可能存在多条不同的路径, 亦即存在多个不同的回报 G_t, 在这种情况下, 使用以回报的期望值作为策略优劣的评判指标成为了 MDP 的核心观点. 定

义智能体在状态 s 处, 遵循策略π所得到的期望回报为状态值函数(statevalue function): $V_\pi(s)=\mathbb{E}_\pi(G_t|s_t=s)$, 定义智能体在状态 s 处执行动作 a, 而后遵循策略π所得到的期望回报为状态-动作对值函数(state-actionpairvalue function): $Q_\pi(s,a)=\mathbb{E}_\pi(G_t|s_t=s,a_t=a)$, 这两种值函数满足如下所示的贝尔曼方程(Bellmane quation):

$$V_\pi(s)=\sum_a \pi(a|s)\left[r+\gamma\sum_{s'}p(s'|s,a)V_\pi(s')\right] \tag{1}$$

$$Q_\pi(s,a)=r+\gamma\sum_{s'}p(s'|s,a)V_\pi(s') \tag{2}$$

其中, s'表示下一时刻的状态, a'表示下一时刻的动作, γ表示单位时间步的折扣系数.

在有限 MDP 中, 由于状态空间和动作空间是有限的, 所以策略也是有限的. 此时, 利用值函数可以精确地得到至少一个最优策略π^*, 优于或等价于其他所有策略. 在优化过程中, 对于一个更优策略π', 智能体执行该策略时, 所有状态的期望回报都应该大于或等于执行其他策略π的期望回报. 也就是说, 对于所有 $s\in\mathcal{S}$, 若π'优于π, 则存在 $v_{\pi'}(s)\geqslant v_\pi(s)$.

1.2 半马尔可夫决策过程

在考虑 MDP 问题时, 智能体的动作是在单时间步内完成的. 为了使智能体可以采取序列动作、构建分层框架, Sutton 等人[8]提出了跨时序的 SMDP 理论. 在 SMDP 下, 智能体可以通过单次执行多步动作来到达更远的状态, 它与基于 MDP 的状态轨迹的对比情况如图 3 所示.

图 3 MDP 与 SMDP 状态轨迹示意图

根据动作序列的执行时间不同, SMDP 可以分为离散时间 SMDP[18]和连续时间 SMDP[19]. 尽管连续时间 SMDP 的灵活度更高, 但离散时间 SMDP 已足够处理大多数的现实问题, 且可以拓展至连续时间情况, 所以接下来只讨论离散时间 SMDP.

1.2.1 option 策略

为了将 MDP 的理论基础拓展至 SMDP, Sutton 等人[8]基于时序抽象法定义了智能体的基础动作(primitive action) $a\in\mathcal{A}$与技能(option) $o\in\mathcal{O}$(技能在有些文章中也被称为宏动作(macro-action)[20]), 用 option o 表达一组动作序列, 用中断函数(terminal function) β_o 确定 option o 的执行步长, 用转移函数 $p(s'|s,o)$表达智能体在状态 s 执行 option o 转移到状态 s'的概率. 假定智能体执行 option o 的执行步长为 N, 其所能获得的奖励为 $r(s,o)$:

$$r(s,o)=\mathbb{E}[r_{t+1}+\gamma r_{t+2}+\ldots+\gamma^{N-1}r_{t+N}|s_t=s,o_t=o] \tag{3}$$

SMDP 存在 option 策略 $\pi_\mathcal{O}(o|s)$ 和 option 内部策略$\pi_o(a|s)$两种策略. option 策略也称为 option 选择策略, 智能体每次根据 option 策略 $\pi_\mathcal{O}$ 选择一个 option o, 然后开始执行 o 对应的 option 内部策略π_o. option 内部策略负责在接下来的 N 步选择基础动作 a, 直到该 option 中断.

根据以上定义, 可以得到智能体在状态 s 下, 遵循 option 策略 $\pi_\mathcal{O}$ 的期望回报 $V_\mathcal{O}(s)$, 即 SMDP 状态值函数(以下公式均简写 $\pi_\mathcal{O}(o|s)$ 和$\pi_o(a|s)$的下标)为

$$V_\mathcal{O}(s)=\sum_{o\in\mathcal{O}_s}\pi(o|s)\left[r(s,o)+\gamma^N\sum_{s'}p(s'|s,o)V_\mathcal{O}(s')\right] \tag{4}$$

其中, $\mathcal{O}_s$表示智能体在状态 s 处的 option 集合, γ^N表示累计奖励在 N 步执行时长上的总衰退程度. 可以看出: SMDP 贝尔曼方程在形式上与 MDP 贝尔曼方程高度统一, 是 MDP 贝尔曼方程在时序抽象上的拓展.

相应地, 定义智能体在状态 s 下执行 option o, 而后遵循 option 策略 $\pi_\mathcal{O}$ 的期望回报为 $Q_\mathcal{O}(s,o)$, 即 SMDP

状态-option 对值函数为

$$Q_{\mathcal{O}}(s,o)=r(s,o)+\gamma^{N}\sum_{s'}p(s'\mid s,o)V_{\mathcal{O}}(s') \tag{5}$$

将 Q 学习(Q-learning)思想应用于 SMDP 可得 SMDP Q-learning 算法, $Q_{\mathcal{O}}(s,o)$ 的更新方程为

$$Q_{\mathcal{O}}(s,o)\leftarrow Q_{\mathcal{O}}(s,o)+\alpha\left[r(s,o)+\gamma^{N}\max_{o'\in\mathcal{O}_{s'}}Q_{\mathcal{O}}(s',o')-Q_{\mathcal{O}}(s,o)\right] \tag{6}$$

其中, α表示学习率, o'表示下一时序间隔的 option.

1.2.2 option 内部策略

在执行 option 内部策略时, SMDP 把状态-option 空间($\mathcal{S},\mathcal{O}$)看成状态空间的增广形式, 定义智能体在增广状态(s,o)下, 遵循 option 内部策略$\pi(a|s)$的期望回报为 option 值函数, 其形式与 MDP 状态值函数定义类似:

$$Q_{\mathcal{O}}(s,o)=\sum_{a}\pi(a\mid s)Q_{U}(s,o,a) \tag{7}$$

其中, $Q_U:\mathcal{S}\times\mathcal{O}\times\mathcal{A}\to\mathbb{R}$表示智能体在增广状态($s,o$)下执行基础动作 a, 而后遵循 option 内部策略$\pi(a|s)$的期望回报, 即增广状态-动作对值函数, 其形式与 MDP 状态-动作对值函数定义类似:

$$Q_{U}(s,o,a)=r(s,a)+\gamma\sum_{s'}p(s'\mid s,a)U(s',o) \tag{8}$$

与 MDP 状态-动作对值函数的区别在于: SMDP 增广状态-动作对值函数包含中断函数β_o, 智能体在到达下一状态后获得的期望回报, 需要考虑当前 option o 的中断情况.公式(8) $U:\mathcal{S}\times\mathcal{O}\to\mathbb{R}$反映的正是智能体执行 option o 后到达状态 s'的价值函数, 即抵达价值函数(upon arrival in the state):

$$U(s',o)=(1-\beta_o(s'))Q_{\mathcal{O}}(s',o)+\beta_o(s')V_{\mathcal{O}}(s') \tag{9}$$

上式表示:

- 若 option o 没有中断: $\beta_o(s')=0$, 则下一状态的价值由 option 值函数描述;
- 若 option o 中断: $\beta_o(s')=1$, 则需要考虑新的 option, 下一状态的价值由状态值函数 $V_{\mathcal{O}}(s')$ 来描述.

如果基于公式(6)来更新价值函数, 智能体将无法在执行 option 内部策略时更新 option 策略, 使得样本利用率变低. 为解决该问题, Sutton 等人[8]根据公式(7)–公式(9)定义了 option 内部的 SMDPQ-learning 计算公式:

$$Q_{\mathcal{O}}(s,o)\leftarrow Q_{\mathcal{O}}(s,o)+\alpha[r+\gamma U(s',o)-Q_{\mathcal{O}}(s,o)] \tag{10}$$

$$U(s',o)=(1-\beta_o(s'))Q_{\mathcal{O}}(s',o)+\beta_o(s')\max_{o'\in\mathcal{O}}Q_{\mathcal{O}}(s',o') \tag{11}$$

2 深度分层强化学习技术

2.1 概 述

SMDP 和时序抽象法作为 DHRL 方法的核心技术, 是构造分层结构的基础, 我们将这两种核心技术统称为分层抽象技术. 当一个序列动作包含多个序列动作或多个基础动作时, 可以认为前者是比后者层次更高、语义更强的动作[8]. 将这些动作以一定规则进行组合, 便形成了 DHRL 的多层结构. 与经典 DRL 相比, DHRL 算法有更强的问题解决能力. 具体来说, DHRL 常用来解决以下 3 种问题.

(1) 稀疏奖励

DRL 的本质是利用奖励函数强化行为的过程, 好的奖励函数可以反映任务的特性, 引导状态和动作的价值被正确估计, 进一步优化策略. 但经典 DRL 把状态空间看成一个巨大的、平坦的搜索空间[21], 这意味着智能体从初始状态到终止状态的路径非常长, 过长的路径会产生奖励信号变弱、延迟增高等问题. 一旦环境只能提供稀疏奖励信号, 问题会变得更为棘手. 此外, ε-贪婪策略和动作噪音作为 DRL 常用的探索方案[22,23], 只能辅助智能体探索临近的、有限的状态空间, 尤其在稀疏奖励环境下, 无法为智能体提供探索更广阔状态空间的动力. 反过来, 探索能力又会影响算法在稀疏奖励环境中的性能. 而 DHRL 利用分层抽象技术, 可以组合多个时序扩展动作, 帮助智能体实现更大范围的状态空间快速覆盖, 强化探索能力; 同时, 也可以快速捕获外部奖励或收集内部奖励, 以此克服稀疏奖励问题.

(2) 顺序决策

许多任务的实现需要遵循一定的顺序决策过程, 例如在蒙特祖玛的复仇中, 需要先拿到钥匙才可以打开门. 该问题有时也被看成部分可观测马尔可夫决策过程(partially observable MDP, POMDP)[24,25]. 因为从本质上来说, 如果不给予先验知识, 钥匙的获取对智能体来说是不可观测的. 经典 DRL 往往无法记录中间过程, 或找不到决策规律. DHRL 的多层结构可以关注不同水平的知识结构[26], 智能体在得到一些关键信息后切换上层策略, 以实现对顺序决策信息的隐性表达.

(3) 弱迁移能力

经典 DRL 通常存在策略可迁移能力不足的问题, 即每一个任务都需要学习专属的网络[4], 且一种算法往往只可以在单一或少数几个任务上取得较优结果. DHRL 能够学到具有高迁移能力的 option, 在面对相似任务时, 智能体可以快速获得学习能力[27]. 同时, DHRL 充分利用状态抽象法, 将不同状态转化为相似的抽象特征, 建立有效的状态特征表达机制[28], 辅助 option 在相似状态区域上的重用.

DHRL 具有较强的学习能力, 可以说, DHRL 对复杂问题的求解能力正是源于分层抽象技术的应用. 但分层抽象技术同样也会引入一些额外问题, 包括分层结构参数过多、训练时间过长、option 学习过程与组合过程的矛盾、异策略分层同步训练不稳定以及子目标太远难以到达等问题. 对于一些更具体的情况, 我们将在后续章节于每一个核心算法的论述中进行说明, 并介绍更为优秀的算法如何在前文基础上进行改进, 以解决这些额外问题.

2.2 常用实验环境

不同于经典 DRL 常用的实验环境, DHRL 实验环境通常强调奖励的稀疏性和顺序决策要求, 大多数经典 DRL 算法都难以在这样的环境中取得效果. 本节对 DHRL 常用的实验环境进行分类介绍.

(1) 空白房间: 这是一种无障碍、只有四壁的环境[29], 常用于训练多样性的 option, 评估智能体对简单环境状态空间的可覆盖能力, 或用于验证智能体抵达某一目标的能力;

(2) 多房间格子世界: 这是一种存在多种形式的离散状态-动作空间环境. Sutton 等人[8]设计了 4 房间任务环境, 每个房间大小相同, 分别由 4 个通道相互连接, 智能体随机开始于某一位置, 到达另一房间的目标位置. Fox 等人[30]修改了房间的连接通道, 使走廊更长, 可用于验证智能体能否学到多样的 option. Rafati 等人[24]同时考虑了稀疏奖励和顺序决策问题, 在 4 房间的基础上增加了钥匙和小车, 智能体在拿到钥匙时只能得到较小奖励, 拿到钥匙后再到达小车位置才算成功;

(3) 迷宫世界: 这是一种存在多种形式的连续状态-动作空间环境. Mankowitz 等人[31]设计了一个 S 形跑道环境, 强调环境的稀疏奖励特性. Osa 等人[32]将两块区域用并排的多个障碍隔开, 验证智能体能否越过不同障碍到达不同终点. Campos 等人[33]用多面墙壁不规则地切割空间, 制造多个瓶颈状态区域, 增加智能体的探索难度;

(4) 电玩游戏(arcade learning environment, ALE)[34]: 作为通用的像素类游戏实验平台, 提供了数百个 Atari 2600 游戏环境接口, 为 DRL 的无模型学习[35]、基于模型规划[36]和模仿学习[37]等研究方向提供了验证环境. 大部分 Atari 游戏属于密集奖励环境, 如潜艇和打方块, 这些环境更加关注智能体对当前状态的快速反应[38]. 但蒙特祖玛的复仇和 pitfall[39]包含了明显的稀疏奖励和顺序决策特点, 与经典 DRL 相比, DHRL 往往可以取得突破性成绩;

(5) gym[40]: 作为 OpenAI 针对连续状态-动作空间任务设计的实验平台, 提供了 31 个动力学控制任务, 包括经典控制(如平衡杆)、Mujoco(如蚂蚁行走)和机械臂(如抓取)等, 且兼容了 ALE 中的 Atari 游戏, 为 DHRL 贡献了更加丰富多样且简单有效的实验环境;

(6) 蚂蚁迷宫: 在 Mujoco 的基础上, 为蚂蚁行走设计的 3D 导航任务(包括蚂蚁寻物[41]和蚂蚁推箱[42]), 具有更苛刻的实验条件, 智能体不仅需要学会多种运动模式, 还需要在复杂迷宫中到达目标点, 或推动箱子完成指定任务;

(7) 猎豹越障: 在 Mujoco 的基础上, 使用带坡度或阶梯的跑道来验证猎豹越过障碍的能力[43], 通常作

为迁移学习的目标任务, 对智能体所学策略有极强的可迁移能力要求.

图 4 展示了部分上述实验环境: 图 4(a)、图 4(b)为多房间格子世界, 图 4(c)–图 4(e)为迷宫世界, 图 4(f)为 Atari 蒙特祖玛的复仇, 图 4(g)为机械臂, 图 4(h)、图 4(i)为蚂蚁迷宫, 图 4(j)为猎豹越障. 表 1 综合描述了不同实验环境的特点, 其中, 符号“●”表示环境通常具备的特性. 需要注意的是: 环境奖励和顺序决策要求是可以灵活调整的, 它们往往会根据算法侧重点进行相应的设定. 例如在多房间格子世界中, 独立训练多样性的 option 时使用无奖励设定[44], 强调 option 探索能力时使用稀疏奖励设定[45], 验证算法性能时使用密集奖励设定[46]. 当需要强调算法对顺序决策问题的求解能力时, 可以在房间中放置通关所必需的钥匙[47]. 而对于出租车和蒙特祖玛的复仇这类环境来说, 其本身就具备了顺序决策要求. 总之, 无奖励、稀疏奖励和密集奖励以及有无顺序决策要求都可以作为 DHRL 算法的验证条件.

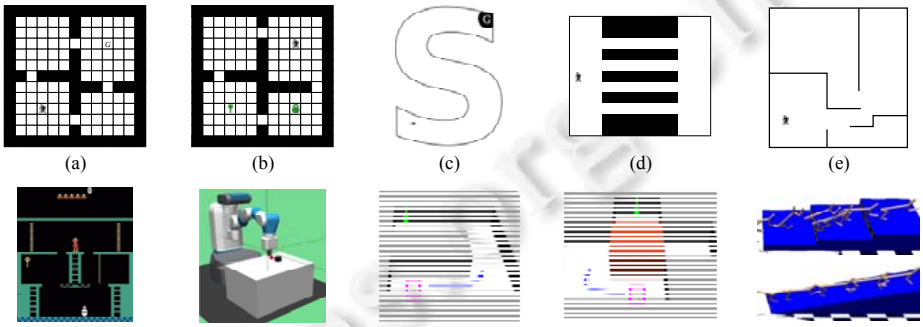

图 4 DHRL 常用实验环境示意图

表 1 DHRL 常用实验环境特点

实验环境	环境特性				
	离散空间	连续空间	2D 导航	3D 导航	动力学控制
空白房间[29,48]	●	●	●	–	–
多房间格子世界[8,24,30]	●	–	●	–	–
迷宫世界[31–33]	–	●	●	–	–
出租车任务[49,50]	●	–	●	–	–
ALE[34]	●	–	●	–	–
Doom[51]	●	–	–	●	–
Minecraft[52,53]	●	–	–	●	–
Deepmind lab[54]	–	●	–	●	–
Labyrinth[55,56]	●	●	●	●	–
gym[40]	●	●	●	●	●
蚂蚁迷宫[42,57]	–	●	–	●	●
猎豹越障[58,59]	–	●	–	●	●

2.3 核心框架

基于分层抽象技术, 学者们提出了丰富多样的 DHRL 方法, 根据求解思路的差异, 我们将它们分为:

(1) 基于技能的深度分层强化学习框架(option-based DHRL, O-DHRL) (option 在 O-DHRL 中常被称为技能(skill), 为保证符号的统一, 下文依然用符号 o 来表示). 下层网络学习一组技能, 然后由上层网络调用这些技能, 使用不同的组合技能来解决下游任务;

(2) 基于子目标的深度分层强化学习框架(subgoal-based DHRL, G-DHRL). 利用神经网络提取状态特征, 然后, 将状态特征作为子目标空间. 上层网络学习产生子目标, 下层网络根据内部驱动来实现子目标;

(3) 除此之外, 早些年, 有学者提出了基于子任务的分层强化学习框架(subtask-based HRL, S-HRL)[60]. 但该框架引入了严重的先验知识问题, 需要人工经验进行任务分解, 如果原问题复杂难分, 则难以使用该方法. 因此, 在追求端到端解决问题的 DRL 领域中, 极少有论文[61–63]以 S-HRL 为基础进行

拓展, 不足以构成完整的 DHRL 研究方向, 故本文不讨论该支线.

具体来说, 之所以将 option 和子目标作为 DHRL 框架的划分依据, 是因为它们有着完全不同的关注点和待解决的问题. O-DHRL 框架通常面对以下两个开放子问题.

(1) 如何发现技能: 技能的多样性和可区分度往往决定了 O-DHRL 的学习能力, 一个好的技能发现方法可以从理论上拓展 DHRL 的研究方向. 既可以用表达学习法来隐式地定义技能[64,65], 也可以用元学习法学得具备强迁移能力的技能[66], 或是从信息论的角度定义与状态分布有关的技能[67]. 可以说, O-DHRL 的核心观点正是如何发现技能. 此外, 技能过短会退化为基础动作, 过长会导致探索与利用过程失衡, 甚至只用单一技能来解决任务, 即产生分层退化问题[68], 失去分层结构优势;

(2) 如何组合技能: 技能的组合方案同样会影响 O-DHRL 在目标任务中的性能. 既可以只关注分层结构对复杂任务的解决能力, 直接在目标任务中交替执行技能学习和技能组合过程[69], 也可以关注技能的可迁移能力, 在源任务中预先学得技能集合, 然后与上层策略进行同步调整[70], 以适应目标任务.

G-DHRL 框架通常面对以下两个开放子问题.

(1) 如何发现子目标: 子目标的定义方法和表达方式往往决定了 G-DHRL 的学习性能. 既可以选取频繁出现在轨迹中的瓶颈状态作为子目标[71], 也可以将聚类后的状态簇心作为子目标[72], 或是将已到达的中间状态直接替换为子目标[73]. 实际上, 任意状态 $s\in\mathcal{S}$都可以找到对应的子目标 $g\in\mathcal{G}$, 也就是说, 存在从状态空间到子目标空间的映射关系 $m:\mathcal{S}\to\mathcal{G}$, 服从$\forall s\in\mathcal{S}$, $f_{m(s)}(s)=1$. 此外, 从表达方式角度出发, 子目标既可以用像素信息表达[74], 也可以用特征编码表达[75];

(2) 如何定义内部驱动: 内部驱动是指由内部奖励驱动的学习行为[76], 用以辅助智能体探索状态空间. 内部奖励既可以采用二元判定函数, 包括 $\{0,1\}$型[77]和$-[|s-g|>\varepsilon]$型[78], 也可以采用距离函数 $D(s,g)$[79], 而采用何种形式往往与子目标的定义方法直接相关.

下面将用两节内容, 详细分析 O-DHRL 和 G-DHRL 框架的算法特点和发展脉络.

3 基于技能的深度分层强化学习

O-DHRL 与 SMDP 密不可分, 而求解 SMDP 问题的关键在于如何定义和寻找 option. 从内容上看, option 既可以由先验知识定义, 也可以由算法学习产生. 从形式上看, option 既可以是单步的基础动作, 也可以是一组动作序列, 或是另一组 option.

O-DHRL 的每个 option 可以由一个三元组$\langle I,\pi_o,\beta_o\rangle$来表示[80], 该三元组的含义分别是:

(1) I 表示 option 初始状态集, 当且仅当状态 $s\in I$ 时, option 才会被执行. 初始条件 I 也可以被看成 option 策略 $\pi_{\mathcal{O}}$, 智能体通过 option 策略 $\pi_{\mathcal{O}}$ 选择当前的 option;

(2) π_o 表示 option o 的内部策略, 用于产生序列动作或序列 option;

(3) β_o 表示 option o 的中断函数, 当某一状态满足β_o条件时, 该 option 结束.

通常, 智能体在某一初始状态选择某一 option 后, 执行该 option 内部策略; 在到达某一状态或满足中断函数时, 停止该 option, 并以此刻状态为初始状态, 继续执行下一 option. 尽管 O-DHRL 增加了 MDP 的复杂性, 但它具有易实现和分层易拓展的优点.

根据近几年 O-DHRL 的技术发展路线, 以上下层策略是否同步训练, 将 O-DHRL 框架分为同步式技能(synchronous option, SO)和异步式技能(asynchronous option, AO).

(1) 在 SO-DHRL 中, 技能和上层策略的训练过程是同步的, 根据对任务处理能力和技能迁移能力的侧重差异, SO-DHRL 又分为独立型技能和共享型技能. SO-DHRL 可以针对特定任务, 直接得到与任务高度相关的技能组合, 具有明显的性能优势, 但单次训练的成本较高;

(2) 在 AO-DHRL 中, 技能和上层策略的训练过程是分离的, 根据求解步骤, AO-DHRL 又分为技能学习和技能组合. 下层网络(技能网络)在训练好数个技能后, 由上层策略在下游任务中调用这些技能. 它通常要求技能在任务无关的环境下进行训练, 使学到的技能具有较好的状态覆盖能力和可迁移

能力. 但技能的多样性难以被量化, 组合技能也不一定总是优于非分层算法.

3.1 同步式技能的深度分层强化学习

3.1.1 独立型技能

独立型 SO-DHRL 强调分层抽象技术在稀疏奖励环境问题中的性能优势, 使用一组上下层策略解决一个任务. 技能-评论家框架(option-critic, OC)[16,81]作为独立型 SO-DHRL 经典算法, 采用和行动者-评论家框架(actor-critic, AC)[82,83]相同的结构, 基于策略梯度定理, 通过优化 option 内部策略和中断函数来最大化期望回报. OC 框架的最大贡献在于: 将 option 概念构建于 AC 框架之上, 采用神经网络对 option 内部策略$\pi_{o,\boldsymbol{\theta}}$和中断函数$\beta_{o,\nu}$进行表达, 并在 SMDP 理论基础上推导出 option 内部策略梯度定理(intra-option policy gradient theorem)和中断函数梯度定理(termination gradient theorem), 保证了 OC 目标函数服从期望折扣回报的形式:

$$\frac{\partial Q_{\mathcal{O}}(s_0,o_0)}{\partial\theta}=\sum_{s,o}\mu_{\mathcal{O}}(s,o\mid s_0,o_0)\sum_{a}\frac{\partial\pi_{o,\theta}(a\mid s,o)}{\partial\theta}Q_U(s,o,a) \tag{12}$$

$$\frac{\partial U(o_0,s_1)}{\partial\nu}=-\sum_{s',o}\mu_{\mathcal{O}}(s',o\mid s_1,o_0)\frac{\partial\beta_{o,\nu}(s',o)}{\partial\nu}A_{\mathcal{O}}(s',o) \tag{13}$$

公式(12)、公式(13)中的各符号均遵循 SMDPoption 内部策略公式(10)、公式(11), 其中,

- $\mu_{\mathcal{O}}(s,o\mid s_0,o_0)$ 表示从(s_0,o_0)到(s,o)的轨迹的折扣权重: $\mu_{\mathcal{O}}(s,o\mid s_0,o_0)=\sum_{t=0}^{\infty}\gamma^t P(s_t=s,o_t=o\mid s_0,o_0)$;
- $A_{\mathcal{O}}$ 表示优势函数: $A_{\mathcal{O}}(s',o)=Q_{\mathcal{O}}(s',o)-V_{\mathcal{O}}(s')$.

与 AC 框架相比, OC 框架在策略选择过程中增加了关于 option 的切换和执行, 将 option 内部策略π_o和中断函数β_o看成行动者的一部分, 并在评论家网络中估计增广状态的值函数 $Q_{\mathcal{O}}(s,o)$ 和优势函数 $A_{\mathcal{O}}(s',o)$. AC 框架与 OC 框架示意图如图 5 所示, 绝大多数的 SO-DHRL 算法均采用了该框架.

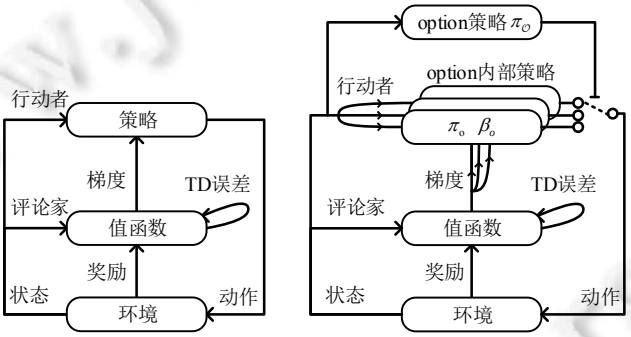

图 5 AC 框架与 OC 框架[16]

在 OC 的基础上, Riemer 等人[84]提出了分层技能-评论家算法(hierarchical option-critic, HOC), 拓展双层 OC 框架至多层 OC 框架. HOC 将 option 内部策略梯度定理和中断函数梯度定理应用于所有层级, 自上而下选择 option, 自下而上判定中断函数, 并逐层调用或中断 option. HOC 在 4 房间任务和 Atari 游戏中的性能远超双层 OC 框架, 但多层结构也使得 HOC 的训练时长大幅增加. 考虑到 OC 和 HOC 框架的每个成分都被分配了相互独立的参数, 即 $\pi_{\mathcal{O}}\cap\pi_o\cap\beta_o=\varnothing$, 这既阻隔了各成分的知识共享, 又增加了算法的训练时长. 为解决参数独立性问题, Riemer 等人[85]继续提出了技能-评论家策略梯度算法(option-critic policy gradient, OCPG). OCPG 不再针对每个成分单独求导, 而是使用期望回报对一组全局共享参数求导, 并推导出针对 OC 和 HOC 框架的共享参数策略梯度定理, 以单一规则优化整个系统. 此外, OCPG 将策略更新的重点放在那些有可能中断 option 的状态上, 根据 option 中断可能性和每个 option 被执行的可能性来分配当前状态下 option 策略的重要性指标, 使梯度更新更有针对性.

由于上下层策略需要同步训练, OC 框架始终面临 option 区分度低和 option 语义不明的问题. 为解决该问题, Osa 等人[86]提出了优势加权信息最大化算法(advantage-weighted information maximization, AdInfo). AdInfo

假设在理想状态下, 每个 option 都有各自对应的模态, 通过引入优势加权重要性权重方法, 可以基于互信息最大化来获取状态-动作对空间($\mathcal{S},\mathcal{A}$)的模态信息, 以学得不同空间区域对应的 option, 保证了 option 的可区分度, 提供了理解 option 语义的途径. 尽管 AdInfo 的思想十分新颖, 但实验效果并不理想. Hou 等人[87]将 AdInfo 思想构建于 SAC 框架上, 提出了基于优势加权混合策略算法(advantage weighted mixture policy, AWMP), 使算法性能大幅提升, 也反过来验证了 AdInfo 思想在复杂环境中的有效性. 基于中断多样性的技能-评论家算法(termination diversity-enriched option-critic, TDEOC)[88]同样引入了信息论方法, 在环境奖励中扩充了 option 内部策略熵, 以增强 option 的多样性; 同时, 使用该奖励的标准型替换 OC 中断函数的优势函数, 以此调整中断概率, 增加不同 option 的利用率, 避免了分层退化问题. 需要注意的是: 虽然 TDEOC 进行了迁移实验说明, 但过高的迁移环境相似度以及性能差距不明显等问题, 均反映其 option 不具备显著的迁移能力. 这一问题同样出现在其他独立型技能算法[89,90]中.

3.1.2 共享型技能

共享型 SO-DHRL 是一种强调迁移能力的算法, 通常分为两个阶段: 在联合训练阶段, 它注重上层策略和技能的同步学习, 在解决源任务的同时, 学得一批具有一定迁移能力的技能; 在迁移学习阶段, 它注重上层策略的学习, 只对技能进行微调.

元学习(meta learning)[91]强调的不再是模型解决某项具体任务的能力, 而是解决一系列任务的能力, 学到一个快速适应任务分布的元策略(meta policy)[92]. 基于元学习的思想, Frans 等人[66]提出了元学习共享分层算法(meta learning shared hierarchies, MLSH), 从任务集中学习可共享的技能. MLSH 在联合训练阶段交替执行以下两个过程: (1) 固定技能网络参数, 更新上层网络; (2) 将技能与当前状态组成扩展状态, 同时对上层网络和被激活的技能网络进行更新. 在迁移学习阶段, 由上层策略调用已学技能, 进行同步学习, 使模型可以快速适应新任务. 实验结果表明: MLSH 可以在蚂蚁寻物游戏中取得较好效果, 且相比于独立型技能算法, 具有极强的迁移能力. 此外, Frans 等人也对 MLSH 的技能多样性作了解释性说明, 执行不同技能时, 智能体的行动方式是完全不同的.

更进一步, 多样性驱动的可扩展分层强化学习算法(diversity-driven extensible HRL, DEHRL)[93]在保证技能可迁移性的同时强调技能的多样性, 用当前技能 o_t 和其他技能 $\neg o_t$ 到达状态的距离之和 $\sum_{s\in\{s_{t+T}|\neg o_t\}} D(s_{t+T}, s)$ 来定义内部奖励, 使明显区别其他技能的当前技能得到更大的内部奖励. 同时, 将双层结构拓展至 3 层, 通过层层传递技能, 强化层级关系. 在胡闹厨房(overcooked)实验中, DEHRL 的性能明显优于 MLSH 和其他 DRL 算法, 且表现出更强的快速适应能力. 但胡闹厨房本身就是 3 层结构任务, DEHRL 的 3 层结构更多地是在特殊任务上进行的特殊处理, 没有发挥出多层结构的一般性优势. 此外, 元学习类 SO-DHRL 算法为了保证技能足够鲁棒, 需要从大量的任务中进行数据采样, 存在数据利用率低、可控性差和稳定性低的问题.

不同于元学习类 SO-DHRL, 环境感知分层强化学习算法(environment-aware HRL, EAHRL)[94]采用了一种更直接的方法, 在第一人称射击(first-person-shooter, FPS)游戏中, 利用先验知识学习具有正交性的技能, 使用杀死敌人或收集资源等行为作为不同技能的内部奖励, 使智能体可以同步执行技能而不相互影响. 在迁移学习阶段, 在保持技能相对稳定的情况下学习上层策略, 快速适应新任务. 但 EAHRL 严格采用了 FPS 的正交序列动作, 难以应用于更复杂的控制任务.

3.2 异步式技能的深度分层强化学习

3.2.1 技能学习

基于赋能(empowerment)[95,96]的技能学习法是 AO-DHRL 的主要技术手段, 通过赋能学得的技能也被称为赋能型技能. 与好奇心机制[97,98]从智能体对环境的理解和预测来探索环境相反, 赋能从信息论的角度衡量智能体对环境的最大可控能力, 即考虑智能体在某状态下采取某动作(或动作序列)能多大程度地影响环境. 赋能型技能使用互信息目标学习法[99], 让算法可以在无奖励环境下发现技能与状态的分布关系, 学习智能体对环境的控制能力[100].

变分内部控制算法(variational intrinsic control, VIC)[44]引入了赋能思想, 采用一组随机潜在变量 $o\sim p(o)$来表达一组技能, 接收环境的持续反馈, 强调以当前状态和技能为条件的动作策略$\pi(a|s,o)$, 通过最大化状态 s 和潜在变量 o 之间的互信息来发现技能:

$$I(S;O)=H(O)-H(O|S)=H(S)-H(S|O) \tag{14}$$

公式(14)的左半部分为逆向通道表达(reverse expression), 利用 KL 散度(kullback-leibler divergence)的非负性, 使用ϕ参数化的变分分布 $q_\phi(o|s)$来近似 $I(S;O)$:

$$I(S;O)=\mathbb{E}_{s,o\sim p(s,o)}[\log p(o|s)]-\mathbb{E}_{o\sim p(o)}[\log p(o)]\geqslant \mathbb{E}_{s,o\sim p(s,o)}[\log q_\phi(o|s)]-\mathbb{E}_{o\sim p(o)}[\log p(o)] \tag{15}$$

其中, 采样分布$p(s,o)=p(o)p(s|o)$, $p(s|o)$是由策略$\pi(a|s,o)$引导的状态分布. 最大化$I(S;O)$意味着最大化$H(O)$, 保证技能的多样性; 同时, 最小化 $H(O|S)$保证可以通过状态来推断所执行的技能. $\log q_\phi(o|s)-\log p(o)$作为内部奖励来优化技能.

公式(14)的右半部分为正向通道表达(forward expression), 同样使用变分分布 $q_\phi(s|o)$来近似:

$$I(S;O)=\mathbb{E}_{s,o\sim p(s,o)}[\log p(s|o)]-\mathbb{E}_{s\sim p(s)}[\log p(s)]\geqslant \mathbb{E}_{s,o\sim p(s,o)}[\log q_\phi(s|o)]-\mathbb{E}_{s\sim p(s)}[\log p(s)] \tag{16}$$

最大化 $I(S;O)$意味着最大化 $H(S)$, 增加状态的多样性; 同时, 最小化 $H(S|O)$保证在给定技能的条件下, 尽可能准确地预测到达状态. $\log q_\phi(s|o)-\log p(s)$作为内部奖励来优化技能.

VIC 确立了在 AO-DHRL 中使用信息论的基本观点, 但未能充分考虑智能体与环境的交互信息. 在 VIC 的基础上, Eysenbach 等人[29]增加了状态、动作和技能的关联性, 提出了多样性分层强化学习算法(diversity is all you need, DIAYN). DIAYN 有 3 个核心观点: (1) 状态的差异可以促使技能产生可区分度, 因此需要增强状态和技能的相关性, 通过访问状态推断出所执行的技能; (2) 不同动作可能引导智能体到达相同的状态, 因此需要消除动作对技能可区分度的影响; (3) 智能体要尽可能随机地行动, 以保证技能的多样性. 根据这 3 个观点, 通过最大化互信息来学习状态、动作和技能的推断关系, 构造如下所示目标函数:

$$F=I(S;O)-I(A;O|S)+H(A|S) \tag{17}$$

其中, 最大化 $I(S;O)$、最小化 $I(A;O|S)$和最大化 $H(A|S)$分别对应上述 3 个观点. DIAYN 在迷宫世界和 Mujoco 环境中对所学技能进行了多样性测试, 在蚂蚁迷宫和猎豹越障环境中对技能组合进行了适应性验证, 均取得了较好的效果.

如果直接使用 DIAYN 来训练大量的技能, 不仅无法保证技能间的可区分度, 还会导致技能难以收敛. 为了获得足够数量且具有差异的技能, 技能变分自编码学习算法(variational autoencoding learning of options by reinforcement, VALOR)[101]在 DIAYN 的基础上学习从轨迹到技能的编码器 $P_D(o|s)$, 然后, 基于课程学习思想, 当编码器足够稳定时, 逐渐增加技能数量. 另一方面, DIAYN 虽然可以学到和状态高关联性的技能, 但其轨迹所覆盖的状态空间往往很小. 为强化智能体在互信息引导下的探索能力, 无监督状态覆盖算法(explore, discover and learn, EDL)[33]将整个学习过程划分为相互独立的探索、技能发现和技能学习这 3 个步骤: (1) 使用状态边缘匹配算法(state marginal matching, SMM)[102]学习状态分布 $p(s)$, 并收集经验数据; (2) 固定状态分布 $p(s)$, 使用变分自动编码器(variational autoencoder, VAE)[103], 将技能发现过程转变为学习对 $p(o|s)$和 $p(s|o)$的建模过程; (3) 采用简化后的正向通道内部奖励 $\log q_\phi(s|o)$来学习策略, 使用 Sibling Rivalry 方法[104]帮助智能体脱离局部最佳状态. 由于 $p(s)$是固定的, 所以技能发现过程不受状态覆盖的影响, 可以引导智能体探索更广阔的区域.

3.2.2 技能组合

在训练好技能后, AO-DHRL 需要考虑如何组合这些技能. 需要注意的是: 虽然共享型 SO-DHRL 同样是将技能进行组合来解决下游问题, 但 AO-DHRL 的技能是独立学习得到的, 往往与环境奖励无关, 上层策略在使用或训练这些技能之前, 都无法验证它们对具体任务的求解能力. 所以从本质上来说, 共享型 SO-DHRL 与 AO-DHRL 的技能组合过程是完全不同的.

随机神经网络分层强化学习算法(stochastic neural networks for HRL, SNN4HRL)[48]在学得 K 个技能后, 固定技能网络, 然后为下游任务构建一个独立的上层网络, 根据当前状态学习待执行的技能 o, 组成扩展状态

(s,o), 输入技能网络, 得到高斯分布$(\mu(s,o),\Sigma(s,o))$引导的动作策略. 实验结果表明: SNN4HRL 既可以学到具有一定区分度的技能, 也可以在稀疏奖励蚂蚁迷宫任务中得到较好效果. 但如果仅使用固定技能来配合上层网络进行动作策略的学习, 往往只能得到次优解. 为解决技能无法在下游任务中自适应的问题, 目前有两种主流改进方法.

(1) 动态调整法

该方法考虑在学得多样性的技能后, 如何与上层策略同步学习来适应下游任务, 典型算法有分层近端策略优化算法(hierarchical proximal policy optimization, HiPPO)[59]. HiPPO 对策略梯度进行修正, 提出了近似分层策略梯度, 使用外部奖励同时训练上层策略和技能, 避免对内部奖励的过度依赖. 此外, HiPPO 还使用了随机长度的技能, 使技能更灵活, 也更稳定. 除了直接对策略梯度进行修正, 基于优势函数与辅助奖励的分层强化学习算法(HRL with advantage-based auxiliary rewards, HAAR), 使用基于上层优势函数 $A_h(s,o)$的内部奖励来优化被调用的技能, 将奖励优势平均分配给技能所引导的所有基础动作 $r^{in}=A_h(s_t,o_t)/k$, 实现内部奖励的信用分配, 体现技能相对其他技能的整体优势, 得到更有针对性的技能调整方案. HiPPO 和 HAAR 算法都对固定技能和动态调整技能进行了对比实验, 结果表明: 动态调整技能可以得到更好的性能, 且具有更好的快速适应能力.

(2) 技能提纯法

动态调整法的技能学习和技能组合过程是完全独立的, 这种独立性可能会影响算法解决下游任务的能力. 技能提纯法考虑如何平衡技能学习过程和技能组合过程, 预先消除技能间的冲突动作后, 再应用于下游任务, 典型算法有独立技能迁移算法(independent skill transfer, IST)[43]. IST 使用主成分分析法(principal components analysis, PCA)对原始技能(primitive skill)进行分解, 消除产生相似动作的不同原始技能的相关性, 以获得低维度、高可组合性的独立技能(independent skill); 然后, 将独立技能迁移到下游任务中. 该方法可以有效地降低技能维度, 减少技能数量, 提高技能的可组合能力.

3.3 分析对比

综上所述, SO-DHRL 与 AO-DHRL 从不同的角度发展了 O-DHRL 框架. 比起上层策略, 如何学习技能几乎是所有方法的核心观点, 也是这些方法的最大差异所在. 表 2 对本节提及算法的主要创新点和缺陷以表格的形式进行描述, 表 3 对本节提及算法的主要技术细节和实验环境以表格的形式进行补充说明. 其中, 编号 1–编号 3 分别表示算法所能解决的稀疏奖励、顺序决策和弱迁移能力这 3 个问题, 符号“●”表示算法使用了某一环境. 表中所涉及的前文未说明的底层算法, 分别有优势行动者-评论家(advantage actor critic, A2C)[105]、孪生延迟深度确定性策略梯度(twin delayed deep deterministic policy gradient, TD3)[106]、最大熵行动者-评论家(soft actor-critic, SAC)[107]、置信域策略优化(trust region policy optimization, TRPO)[108]和近端策略优化算法(proximal policy optimization, PPO)[109]. 从表 2 和表 3 可以明显看出: 随着 O-DHRL 算法的发展, 它们逐渐向高性能、少先验经验和高可迁移能力的方向发展, 并试图解决决策更为复杂的问题.

表 2 O-DHRL 算法分析与对比

算法类型	经典算法	创新点	缺陷
SO-DHRL	OC[16]	首次构建了基于 DRL 算法的 O-DHRL 框架, 使用策略梯度优化函数, 能够较好地解决稀疏奖励问题, 可拓展性高	严重的分层退化问题, option 可区分度不高, 无明显 option 语义信息
	HOC[84]	将双层 OC 框架拓展至多层, 实现了具备一般性的多层架构, 相比于 OC 框架, 性能显著提升	计算消耗大, 训练时间长, option 可区分度不高, 无明显 option 语义信息
	OCPG[85]	用一组全局共享参数取代 OC 和 HOC 框架中的独立参数, 保证性能的同时, 大幅缩短训练时长	option 可区分度不高, 无明显 option 语义信息, 缺乏与最新算法的横向对比
	AdInfo[86]	以发现 option 对应模态为解决方案, 保证了 option 的可区分度, 提供了理解 option 语义的途径	算法性能差, 甚至无法胜过非分层的底层算法, 且对比算法性能可信度低
	AWMP[87]	大幅提升 AdInfo 算法性能, 验证了以 option 对应模态为解决方案的有效性	与非分层的底层算法性能大致相同, 没有体现分层抽象技术带来的收益

表 2 O-DHRL 算法分析与对比(续)

算法类型	经典算法	创新点	缺陷
SO-DHRL	TDEOC[88]	在环境奖励中加入 option 内部策略熵, 提高了 option 的多样性, 增加了 option 的利用率, 避免了分层退化问题	计算消耗大, 训练时间长, 性能对 option 内部策略熵的系数敏感, 稳定性不足
	MLSH[66]	首次在 DHRL 中引入元学习思想, 可以学到强迁移能力的技能, 且技能具有一定的可解释性	采样要求高, 数据利用率低
	DEHRL[93]	采用 3 层结构的 O-DHRL, 验证了多层语义表达的可能性, 在保证技能可迁移性的同时, 强化了技能的多样性	采样要求高, 数据利用率低, 多层结构不具有一般性
	EAHRL[94]	技能语义强, 可同步执行而不互相影响, 数据利用率高	针对特定环境, 技能不具有一般意义
AO-DHRL	VIC[44]	首次将最大赋能型技能应用于 DHRL, 能够习得任务无关的技能	只有得到奖励的技能才会被提升, 导致被调用的技能数目较少, 存在分层退化问题
	SNN4HRL[48]	学习多模态分布下的策略, 添加了基于互信息的正则项, 保证了一定程度的技能多样性	互信息正则项并不总有效, 技能多样性差, 探索能力不足, 容易陷入局部最优解
	DIAYN[29]	充分利用信息论的观点来发现状态、动作和技能的分布关系, 能够得到与状态高相关性的多样性技能	技能探索能力不足, 容易陷入局部最优解, 能够学得有效多样性技能的数量有限
	VALOR[101]	通过课程学习可以学得大量且足够多样性的技能	技能多样性的程度对课程学习参数敏感
	EDL[33]	首次结合最大赋能型技能的前后通道表达, 极大地增强了技能探索能力	仅在格子世界环境中进行了验证, 缺乏对高维空间或动力学任务的处理能力
	HiPPO[59]	提出近似分层梯度方法, 采用动态调整法增强技能对新任务的适应能力. 可以动态调整技能长度, 使技能更灵活	技能学习和技能组合过程是完全独立的, 算法对下游任务的处理能力不足
	HAAR[57]	采用上层优势函数定义内部奖励, 将优势值分配给技能引导的所有动作, 增强技能对新任务的适应能力	技能学习和技能迁移过程是完全独立的, 算法对下游任务的处理能力不足
	IST[43]	将强相关性的原始技能分解为独立技能, 降低技能维度, 减少技能数量, 提高技能的可组合能力	目前最优算法, 无其他明显缺点

表 3 O-DHRL 算法底层算法和实验环境

经典算法	年份	底层算法	解决问题	空白房间	多房间	迷宫世界	ALE	gym	蚂蚁迷宫	猎豹障碍	其他
OC[16]	2017	DQN	1	●	●	–	●	–	–	–	●
HOC[84]	2018	A3C	1	–	●	–	●	–	–	–	–
OCPG[85]	2019	A3C	1	–	●	–	●	–	–	–	–
AdInfo[86]	2019	TD3	1	–	–	–	–	●	–	–	–
AWMP[87]	2020	SAC	1	–	–	–	–	●	–	–	–
TDEOC[88]	2020	PPO	1	–	●	–	–	●	–	●	●
MLSH[66]	2018	DQN, A3C TRPO, PPO	1, 2, 3	–	●	–	–	–	●	–	●
DEHRL[93]	2019	PPO	1, 2, 3	–	–	–	–	–	–	–	●
EAHRL[94]	2019	A2C	1, 2, 3	–	–	–	–	–	–	–	●
VIC[44]	2016	*Q*-learning	1, 3	●	●	●	–	–	–	–	●
SNN4HRL[48]	2017	TRPO	1, 3	●	–	–	–	●	●	–	–
DIAYN[29]	2019	SAC	1, 2, 3	●	–	●	–	●	●	●	–
VALOR[101]	2018	A2C	1, 2, 3	●	–	●	–	●	–	–	●
EDL[33]	2020	SAC	1, 3	●	–	●	–	–	–	–	–
HiPPO[59]	2019	PPO	1, 2, 3	–	–	–	–	–	●	●	–
HAAR[57]	2019	TRPO	1, 2, 3	–	–	–	–	–	●	–	–
IST[43]	2020	SAC	1, 2, 3	–	–	–	–	–	–	●	●

4 基于子目标的深度分层强化学习

G-DHRL 不直接定义 option, 而是设计了特定状态作为子目标集$\mathcal{G}$, 把智能体从当前状态 s 到达子目标$g\in\mathcal{G}$的过程看成一个 option, 然后根据外部奖励(环境奖励)r^{out} 学习一个上层策略π_h:$\mathcal{S}\to\mathcal{G}$, 逐步引导智能体完成任务[110]. 子目标的设计过程等价于构建了多个 SMDP, 将原问题划分为数个小任务, 缩小了状态空间; 子

目标的切换过程等价于实现了某一子任务, 是解决顺序决策过程的重要手段; 子目标的中断函数常简化为固定步长.

通用价值函数逼近算法(universal value function approximators, UVFA)[15]为 G-DHRL 框架奠定了理论基础, 使得大部分经典 DRL 算法可以直接作为 G-DHRL 的底层算法. G-DHRL 的通用框架如图 6 所示, 它以状态 s 为上层网络(上层控制器)输入, 输出子目标 g; 以状态-子目标对(s,g)作为增广状态, 输入下层网络(下层控制器), 学习增广状态下的下层策略$\pi_l:\mathcal{S}\times\mathcal{G}\to\mathcal{A}$. 上层策略根据外部奖励进行训练, 下层策略根据内部驱动进行训练. 对任意增广状态(s,g)和内部奖励 r^{in}, 定义增广状态值函数 $V_\pi(s,g)$的计算公式为

$$V_\pi(s,g)=\mathbb{E}\left[\sum_{t=0}^{\infty}r^{in}(s_t,a_t)\prod_{k=0}^{t}\gamma_g(s_k)\,|\,s_0=s\right] \tag{18}$$

其中, γ_g 表示子目标限制下的折扣系数, $\gamma_g(s)=0$ 当且仅当状态 s 与子目标 g 重合.

相应地, 定义增广状态-动作值函数 $Q_\pi(s,a,g)$的计算公式为

$$Q_\pi(s,a,g)=\mathbb{E}_{s'}[r^{in}(s,a)+\gamma_g(s')V_\pi(s',g)] \tag{19}$$

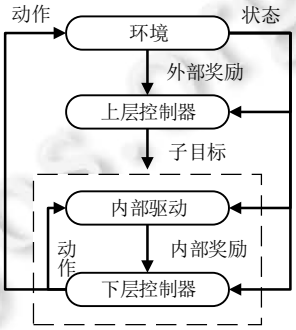

图 6 G-DHRL 通用框架[9]

根据近几年 G-DHRL 的技术发展路线, 以子目标定义方式, 将 G-DHRL 框架分为先见子目标(foresight subgoal, FG)和后见子目标(hindsight subgoal, HG).

(1) FG-DHRL 从先验经验或采样数据中选取子目标, 根据内部驱动是建立在二元判定机制上还是相对距离机制上, FG-DHRL 又分为判定型子目标和引导型子目标. FG-DHRL 具有极高的可拓展性, 但在具体实现时可能面临先验设定问题, 如初始子目标集的选定和子目标数量的定义等;

(2) HG-DHRL 将已到达的中间状态直接替换为子目标, 并更改相应的内部奖励, 模仿人类从错误中学习知识的过程, 给予智能体从错误经验中学习的能力. HG-DHRL 算法在应对稀疏奖励环境时十分有效, 避免了多种先验设定问题, 但作为一种启发式学习方法, 收敛性至今仍无法得到有效证明.

4.1 先见子目标的深度分层强化学习

4.1.1 判定型子目标

判定型 FG-DHRL 根据智能体是否到达子目标来提供二元内部奖励, 首次被应用于 ALE 平台的 FG-DHRL 算法是 Kulkarni 等人[9]提出的分层 Q 网络(hierarchical deep reinforcement learning, HDQN). HDQN 基于 DQN[4]构建上下层网络, 可以在不同时间尺度上优化上层策略(子目标策略)和下层策略(动作策略), 使智能体具有长时间链的决策能力. HDQN 具体算法流程如下.

(1) 上层网络接收当前状态 s, 学习上层策略π_h, 输出子目标 g.

此时, 关于增广状态(s,g)的最优价值函数 $Q_h^*(s,g)$ 的计算公式为

$$Q_h^*(s,g)=\max_{\pi_h}\mathbb{E}[R_t^{out}+\gamma\max_{g'}Q_h^*(s_{t+N},g')\,|\,s_t=s,g_t=g] \tag{20}$$

其中, R_t^{out} 表示从 t 时刻开始的 N 步时序间隔的累计外部奖励: $R_t^{out}=\sum_{t'=t}^{t+N}r_{t'}^{out}$, g'表示下一时序间隔的子目

标. 上层网络的损失函数如下所示:

$$L(\boldsymbol{\theta}_h)=\mathbb{E}_{(s,g,R^{out},s')\sim D_h}\left[(R^{out}+\gamma\max_{g'}Q_h(s',g';\boldsymbol{\theta}_h)-Q(s,g;\boldsymbol{\theta}_h))^2\right] \tag{21}$$

其中, $\boldsymbol{\theta}_h$表示上层网络参数, $(s,g,R^{out},s')\sim D_h$表示从上层经验池中采样. 当智能体到达子目标或超出规定时间步时, 上层网络会重新选择子目标.

(2) 下层网络接收增广状态(s,g), 学习下层策略π_l, 输出基础动作 a.

此时, 最优增广状态-动作对值函数 $Q_l^*(s,a,g)$ 的计算公式为

$$Q_l^*(s,a,g)=\max_{\pi_l}\mathbb{E}\left[r_t^{in}+\gamma\max_{a'}Q_l^*(s_{t+1},a',g)\mid s_t=s,a_t=a,g_t=g\right] \tag{22}$$

其中, r_t^{in} 表示 t 时刻的内部奖励. 每一次选择动作, 内部奖励函数都会根据智能体是否实现子目标来生成内部奖励 r^{in}. 下层网络的损失函数如下所示:

$$L(\boldsymbol{\theta}_l)=\mathbb{E}_{(s,a,g,r^{in},s')\sim D_l}\left[(r^{in}+\gamma\max_{a'}Q_l(s',a',g;\boldsymbol{\theta}_l)-Q_l(s,a,g;\boldsymbol{\theta}_l))^2\right] \tag{23}$$

其中, $\boldsymbol{\theta}_l$表示下层网络参数, $(s,a,g,r^{in},s')\sim D_l$表示从下层经验池中采样.

实验结果表明, HDQN 可以在 DQN 无法取得任何效果的蒙特祖玛的复仇中取得较好成绩, 验证了 HDQN 解决稀疏奖励、长时段延迟奖励和顺序决策等问题的能力. 但其缺点也十分明显, HDQN 的子目标空间是人为设定的, 上层网络只是学习这些子目标的排列顺序, 难以对算法进行一般性推广.

为了减少先验知识的引入, 让算法可以自动找到子目标, Rafati 等人[24]提出了通用分层强化学习算法(unified HRL, UHRL). UHRL 使用无监督法和异常检测法, 从经验池中选择簇心状态和附带奖励的状态作为子目标集, 在收集到更多经验时, 持续更新子目标集, 保证子目标集可以覆盖更大的状态空间. 实验结果表明, UHRL 在 4 房间任务和蒙特祖玛的复仇中均取得了较好的成绩. 但 UHRL 经验池的拓展严重依赖于随机探索过程, 如果探索不够充分, 子目标集的更新会受到影响. 此外, HDQN 和 UHRL 仅在蒙特祖玛的复仇中进行了测试, 而蒙特祖玛的复仇与 4 房间一样具有天然的顺序决策特性, 算法巧妙地利用了这一特性, 却没有在其他 Atari 游戏中进行实验, 失去了一般性.

4.1.2 引导型子目标

引导型 FG-DHRL 使用状态和子目标的距离或余弦相似度作为内部奖励. 与判定型子目标相比, 使用引导型子目标可以在一定程度上保持子目标的稳定性, 持续引导智能体向子目标移动; 同时, 在上层策略接收到有意义的监督信号之前就可以获得内部奖励, 实现快速学习; 此外, 还有利于上层控制器调用相似的下层策略, 算法可迁移能力强[111].

封建领主网络(feudal networks, FuN)[75]模型是典型的引导型 FG-DHRL 算法, 以封建领主强化学习(feudal reinforcement learning, FRL)[112]模型为理论基础, 构建了一个模块化、上下层梯度分离且端到端可微的双层网络. FuN 在不同层级中重复使用感知网络和长短期记忆单元(long short-term memory, LSTM)[113,114], 获得强可观测性的隐层状态, 然后将该隐层状态作为子目标, 赋予其在低维状态空间上的方向语义. 从实验来看, FuN 可以处理的 Atari 游戏包括但不限于蒙特祖玛的复仇, 既提升了实验效果, 又保证了对不同问题的求解能力, 整体性能远超 HDQN. 但 FuN 模型过于复杂, 存在着调参难度大的问题, 且上下层都建立在同策略(on-policy)算法的基础上, 虽提高了分层结构的稳定性, 但也限制了样本利用率.

为了克服 FuN 存在的问题, Nachum 等人[42]提出了异策略修正分层强化学习算法(HRL with off-policy correction, HIRO), 构建了一个建立异策略(off-policy)算法上的分层结构. 该算法用期望到达状态与当前状态的方向残差作为子目标: $g_{t+1}=s_t+g_t-s_{t+1}$. 然后, 以当前状态与子目标位置的残差作为内部奖励. 虽然异策略提高了 HIRO 的样本利用率, 但也引入了分层同步训练的非稳定性问题, 即相对于下层策略的改进, 上层策略的更新存在滞后性, 相同的子目标不再能引导智能体得到与历史样本相同的状态转移和奖励. 针对这一问题, HIRO 提出了上层经验修正方法, 在采样后固定奖励, 找到一个修正子目标 $\tilde{g}_t$ 来替换原样本中的 g_t, 使下式概率最大:

$$\tilde{g}_t = \arg\max_{\tilde{g}} \pi_l(a_{t:t+N-1} \mid s_{t:t+N-1}, \tilde{g}_{t:t+N-1}) \tag{24}$$

修正的意义在于: 更换一组间隔目标, 使当前下层策略产生与历史下层策略相同的状态-动作序列. 在蚂蚁迷宫的消融实验中, 验证了 HIRO 修正方法能够明显消除分层非稳定性问题的影响.

在 HIRO 的基础上, Wang 等人[115]构建了基于交互影响的分层强化学习框架(interactive influence-based HRL, I^{2}HRL). 该框架包含 3 种分层非稳定性问题的消除方案: (1) 建立了双向通信机制, 上层控制器根据当前状态和上一步的下层策略编码 $m=f(\pi_l)$输出子目标; (2) 通过子目标与下层策略编码之间的互信息 $I(G;M|S)$来减小下层策略对算法稳定性的影响; (3) 将上层策略编码器和固定分布的 KL 散度 $D_{KL}(p_{enc}(z|s,m)\|q(z))$扩充进环境奖励, 作为上层策略学习的正则项, 引导智能体访问更多的不稳定状态, 频繁更新不稳定状态的价值函数, 提高采样效率. 从消融实验的结果来看, 这 3 种方案对非稳定性问题的消除均有积极作用, 但考虑到对比实验使用了弱效果的 HIRO 代码(非原文版本), 所以实际提升的性能较为有限.

从另一个角度看, 引导型子目标既可以作为智能体期望到达的状态, 也可以作为智能体意图远离的状态. 锚点分层强化学习算法(anchor HRL, AHRL)[116]将每 c 步到达的状态定义为锚点(anchor), 以状态 s_{t+1} 和 t 时刻锚点 $g_{[t/c]c}$([·]表示向下取整函数)之间的距离定义内部奖励 r_t^{in}, 鼓励智能体离开锚点. 同时, AHRL 收集智能体在远离锚点 $g_{[t/c]c}$过程中的外部奖励 R_t, 对内部奖励进行加权: $\tilde{r}_t = f(R_t)r_t^{in}$, 以鼓励智能体朝正确的方向移动.

此外, 对子目标空间进行限制也是引导型子目标的常用技巧. 例如: 基于邻接约束子目标的分层强化学习算法(HRL with adjacency constraint, HRAC)[117]使用邻接约束条件, 将子目标空间从整个状态空间限制到当前状态的 k 步相邻区域, 既缓解了上层策略的探索压力和价值函数逼近负担, 又为下层策略提供了更有效的学习信号. 慢性动态子目标表达(learning subgoal representations with slow dynamics, LESSON)[118]算法利用分层抽象技术特性, 最大化上层相邻时间步之间的特征变化 $\max_{\mathcal{F}}[\|f(s_t)-f(s_{t+N})\|_2]$, 最小化下层相邻时间步之间的特征变化 $\min_{\mathcal{F}}[\|f(s_t)-f(s_{t+1})\|_2]$, 采用三元组损失方法整合上下层特征变化, 学习子目标的慢性动态表达, 实现子目标空间的缩减. LESSON 学得的慢性动态表达不仅可以为分层策略提供可解释性, 还可以随同下层策略在多个任务间进行迁移.

从性能角度看, HIRO 和 LESSON 等算法在蚂蚁迷宫环境中的效果明显优于 FuN 和 SNN4HRL 等其他 DHRL 算法. 但这些蚂蚁迷宫被设定为密集奖励环境, 与强调解决稀疏奖励任务的 FuN 和 SNN4HRL 进行对比, 不具有严格的定性条件. 此外, 也正是密集奖励环境导致了累计奖励随策略更新而频繁变化的现象, 使得分层同步训练的非稳定性问题更加严重. 正如 HRAC 使用的是密集奖励蚂蚁迷宫环境, 侧重点便不再聚焦于消除分层非稳定性问题的影响, 这也是稀疏奖励环境极少讨论该问题的原因.

4.2 后见子目标的深度分层强化学习

与 DRL 算法相比, 人类在无法得到明确反馈的情况下, 依然可以积累一定的经验. 通常情况下, 智能体在稀疏奖励环境中产生的轨迹$(s_0,\ldots,s_N)$必然难以到达预设子目标. 但换一个角度, 尽管这些失败的轨迹无法引导智能体学习如何到达预设子目标, 但它却告诉智能体如何到达该轨迹的终止状态 s_N. 基于这一思想, 后见经验回放(hindsight experience replay, HER)[17]机制直接将终止状态 s_N 当作用于回放的额外子目标 g', 针对所有经验转移样本重新定义奖励函数 $r'_{t+1}=r_{g'}(s_t,a_t)$, 以获得新的经验转移样本 $(s_t,a_t,r'_{t+1},s_{t+1},g')$. 通过这种设定, 增加了包含正奖励的情节, 采用渐进式的学习方法, 逐步降低问题的难度, 使奖励变得稠密.

Levy 等人[73]将 HER 的思想应用于分层结构, 提出了分层行动者-评论家(hierarchical actor-critic, HAC)算法, 用于解决 HIRO 中描述的分层非稳定性问题, 同时实现多层策略的并行学习.

HAC 设计了 3 种转移方法.

(1) 后见动作转移(hindsight action transitions): 该转移是针对上层策略设定的. 对于非最底层策略来说, 选择动作就是选择子目标, 若下层策略无法引导智能体到达上层动作(子目标), 就用实际到达的状态替换该上层动作. 以此假定下层策略已是最优策略, 训练上层策略而暂时忽略下层策略的变化;

(2) 后见子目标转移(hindsight goal transitions): 该转移是针对下层策略设计的. 对子目标进行后见经验回放, 用实际到达的状态替换下层接收到的子目标, 保证下层状态转移可以得到奖励;

(3) 子目标测试转移(subgoal testing transitions): 以一定概率验证下层策略能否实现回放后的子目标, 给予无法实现子目标的上层转移样本惩罚.

相比于单层结构的 HER, HAC 利用分层抽象技术强化了智能体的探索能力, 而叠加这 3 种转移方法, 通过层层传递子目标, 可以将双层 HAC 拓展至 3 层, 使探索能力进一步增强; 又因为转移方法可以完全消除下层策略变化对上层状态转移的影响, 解决异策略带来的非稳定性问题, 所以 HAC 始终保持稳定. 实验结果表明: HAC 在 4 房间、倒立杆和蚂蚁迷宫任务中均有极好的表现, 且 3 层结构的 HAC 明显强于双层结构的 HAC.

尽管后见子目标作为一种启发式学习方法, 收敛性至今仍无法得到有效证明, 但其简单有效的求解问题思路, 使算法具有优秀的可拓展性[119]. 自动课程生成分层强化学习(automatic curriculum generation by HRL, ACGHRL)[120]算法在 HAC 基础上引入课程学习法, 通过输出插值系数α, 在已到达子目标 g_a 和终止目标 g 之间进行插值, 得到当前子目标: $g_i=g_a+(g-\alpha g_a)$, 以此引导智能体突破已到达子目标的边界, 实现向终止目标稳步递进的过程. 理想情况下, 训练收敛后, 无需再次设置子目标, 就可以令智能体完成远距离的目标任务. 好奇心分层行动者-评论家(curious hierarchical actor-critic, CHAC)[121]算法在 HAC 基础上引入好奇心机制, 采用基于预测误差的好奇心模型[122], 为基础动作 $a^{i=0}\in\mathcal{A}$和子目标 $a^i\in\mathcal{A}_i$, $i\neq 0$ 构建前向模型 $f^i_{fw}(s_t,a^i_t)\Rightarrow\hat{s}^i_{t+1}$, 然后定义基于预测误差的好奇心内部奖励 $r^i_t=(s^i_{t+1}-\hat{s}^i_{t+1})^2/2$, 训练包括前向模型在内的所有模块参数.

4.3 分析对比

综上所述, G-DHRL 的核心观点在于如何定义子目标, 这也是这些方法的最大差异所在. 表 4 对本节提及算法的主要创新点和缺陷以表格的形式加以描述, 表 5 对本节提及算法的主要技术细节和实验环境以表格的形式进行补充说明. 其中, 编号 1–编号 3 分别表示算法所能解决的稀疏奖励、顺序决策和弱迁移能力这 3 个问题, 符号“●”表示算法使用了某一环境.

表 4 G-DHRL 算法分析与对比

算法类型	经典算法	创新点	缺陷
判定型 FG-DHRL	UVFA[15]	为 FG-DHRL 的发展奠定基础, 设定了以增广状态为下层策略输入、以子目标实现判定为内部驱动的理论基础	没有提出有效的子目标发现方法, 算法性能一般
	HDQN[9]	首次在 ALE 平台中实现的 FG-DHRL 算法, 可以处理当时其他 DRL 算法完全无法解决的稀疏奖励任务	子目标集需要人工设定, 需追加额外的智能体位置判定机制, 计算成本高
	UHRL[24]	在 HDQN 的基础上增加了基于无监督和异常检测的子目标自动发现方法, 缓解了子目标需要人为设定的要求	子目标集的初始设定受限, 需追加额外的智能体位置判定机制, 计算成本高
引导型 FG-DHRL	FuN[112]	构建了端到端、模块化、可微的双层网络, 首次采用引导型子目标, 使算法更为灵活, 且可以处理连续空间任务	网络模块复杂, 性能过于依赖超参, 调参难度大, 细微改变足以使模型崩塌
	HIRO[42]	首次对异策略 FG-DHRL 的分层非稳定性问题进行了探讨, 并通过子目标修正法缓解了该问题	方差大, 算法难以平稳收敛, 与同策略分层训练算法的性能仍存在较大差距
	I^2HRL[115]	构建了基于影响力的框架, 能够缓解分层同步训练的非稳定性问题的影响	使用了弱效果的 HIRO 代码, 实际提升的性能有限
	AHRL[116]	一种新颖的子目标设定方式, 以离开当前锚点定义内部奖励, 并使用外部奖励进行方向修正, 子目标可解释性强	在主要实验环境中, 缺乏与最新算法的横向对比
	HRAC[117]	将子目标空间进行邻域限制, 为下层策略提供了更有效的子目标, 缓解了上层策略的探索难题和值函数逼近负担	需要构建邻域矩阵, 当状态空间过大时, 将严重影响学习效率
	LESSON[118]	可以学到子目标的慢性动态表达, 为分层策略提供了可解释性, 强化了算法的迁移能力	三元组损失系数与环境大小高度绑定, 需要严格匹配环境大小来进行设置

表 4 G-DHRL 算法分析与对比(续)

算法类型	经典算法	创新点	缺陷
HG-DHRL	HAC[73]	将 HER 拓展至多层结构, 实现了 HG-DHRL 算法在稀疏奖励环境下的应用, 消除了非稳定性问题, 可拓展极高	收敛性不可证, 目前最优算法, 无明显单独的缺点
	ACGHRL[120]	引入课程学习法, 通过在已实现子目标和终止目标间进行插值, 以获得高可达性子目标	收敛性不可证, 对 HAC 的性能提升有限
	CHAC[121]	首次将好奇心引入 HG-DHRL, 增强了智能体对状态空间的预测和覆盖能力	收敛性不可证, 需要训练的网络模块较多, 训练速度慢

表 5 G-DHRL 算法的底层算法和实验环境

经典算法	年份	底层算法	解决问题	空白房间	多房间	迷宫世界	ALE	gym	蚂蚁迷宫	其他
UVFA[15]	2015	Q-learning	1	–	●	–	–	–	–	●
HDQN[9]	2016	DQN	1, 2	–	–	–	●	–	–	●
UHRL[24]	2019	DQN	1, 2	●	●	–	●	–	–	–
FuN[112]	2017	A3C	1, 2, 3	●	–	–	●	–	–	–
HIRO[42]	2018	TD3	2	–	–	–	–	–	●	–
I^{2}HRL[115]	2020	TD3	2	–	–	–	–	–	●	–
AHRL[116]	2021	TD3	2	–	–	●	–	●	●	–
HRAC[117]	2020	上层 TD3, 下层 A3C	1, 2	●	●	–	–	–	●	–
LESSON[118]	2020	SAC	1, 2, 3	–	–	–	–	–	●	●
HAC[73]	2019	DDPG	1, 2	–	●	–	–	●	●	–
ACGHRL[120]	2020	PPO	1	–	●	–	–	–	–	●
CHAC[121]	2020	DDPG	1, 2	●	–	–	–	●	●	●

从表 4 和表 5 可以明显看出: 随着 G-DHRL 算法的发展, 它们逐渐向子目标自动设定、子目标高可达性和高数据利用率的方向发展, 并试图解决奖励更稀疏和决策更复杂的问题.

5 深度分层强化学习应用

目前, DHRL 方法已被广泛用于视觉导航、自然语言处理、推荐系统和视频描述生成等真实世界应用领域, 以解决现实生活中的稀疏奖励和顺序决策等问题, 并展现出巨大的商业价值. 图 7 描述了从 2016–2021 年(截至 2021 年 6 月), DHRL 在不同真实世界应用领域的论文数量占比情况(共 78 篇).

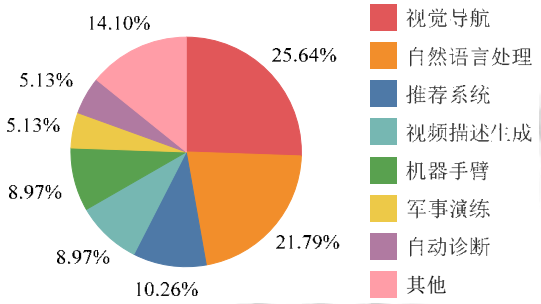

图 7 DHRL 在真实世界应用领域的论文数量

5.1 视觉导航领域

视觉导航领域包括自动驾驶模拟和目标导向机器人两种任务, 它们不仅要求智能体具备对图像数据的表达能力, 还要求控制器能够以不同频率更新路况信息和动作策略. 考虑到这些特性, 将分层抽象技术应用在视觉导航领域, 可以发挥重要作用.

大量研究表明: 许多动物在自我定位和路径规划方面所形成的空间表达的能力, 都依赖于大脑对原始感知信号的特征编码. 在自动驾驶模拟任务中, 慢性特征分析(slow feature analysis, SFA)[123]算法从视觉图像中学得拓扑地图, 利用 DHRL 从拓扑地图中学得丰富的环境层级表达, 为车辆在不同空间尺度上实现自我定位

和方向检测. 在红绿灯通行问题中, Chen 等人[11]提出了一种分层策略梯度方法, 学习数个简单且有差异的技能, 然后组合技能来获得对复杂问题的求解能力, 使车辆在交通灯变黄时做出正确选择. 这些算法的实验结果均表明: 相比于经典 DRL, DHRL 有更好的学习能力, 可以帮助车辆实现模拟驾驶, 包括并道和等待红绿灯等操作(如图 8 所示).

在目标导向机器人应用中, 出于稳定和安全的考虑, 位置估计器需要以较低频率更新, 而动作控制器必须在几毫秒内计算出电机指令. Jain 等人[124]针对 4 足机器人路径跟踪任务, 充分利用 DHRL 的分层结构特性和时序解耦方案, 为上下层控制器使用不同的状态表达, 强调位置估计和动作控制的不同关注点, 确保下层策略的可重用能力; 并在可变的时间尺度上更新上下层策略, 减轻硬件对上层状态信息的处理需求.

Li 等人[125]在 18 自由度机器人的多目标导向任务中, 对技能进行预训练, 得到可以实现简单目标的技能(如转弯和直线行走), 然后对技能进行规划学习. 这种分层学习方式不仅可以利用预训练技能提高对多目标任务的求解能力, 还可以减少构建上层模型所需的硬件数据.

图 8 自动驾驶模拟和目标导向机器人任务

5.2 自然语言处理领域

DHRL 在自然语言处理领域常用于任务导向型对话生成(task-oriented)和开放域对话生成(open-domain)方向, 与经典环境 Atari 相比, 这些任务的动作维度要高出多个数量级.

在任务导向型对话生成任务中, Budzianowski 等人[126]利用 DHRL 的强迁移能力来学习跨领域对话系统. 考虑到不同领域中存在着相似的子域, 如订购房间和购买书本主域都有付款子域, 该算法在不同主域的相似子域中学习可共享的信息, 以训练通用的下层策略. Saha 等人[127,128]利用 DHRL 框架来学习多意图对话策略. 考虑到大多数对话系统只使用了用户语义而忽略了用户行为和情感在对话中的作用, 该算法将基于情感的即时奖励引入到对话系统基础奖励中, 使问答机器人具有自适应能力, 意图获得最大用户满意度. 实验结果表明, 用户情感和行为等信息在创造复合性的问答机器人和最大化用户满意度方面均发挥了重要作用.

在开放域对话生成领域, 经典 DRL 方法[129,130]往往只能在单词层面上构建奖励模型, 这种低水平的控制将不利于信用分配, 导致奖励模型难以跟踪长期对话目标. 为克服这一挑战, Saleh 等人[131]提出了变分对话模型分层强化学习(variational sequence model HRL, VHRL)算法. 该算法不再单纯考虑单词级别的信息, 而是在话语层次上建立奖励模型, 提高模型的全局视野和灵活性, 以学习长期的对话回报. VHRL 避免了在电影这类长对话数据中可能产生的不适当、有偏见或攻击性的文本, 在人类评估和自动指标性能方面均超过了最先进的对话模型[132].

5.3 推荐系统领域

推荐系统具有巨大的商业价值, 序列推荐(sequential recommendations)作为推荐系统中与 DRL 技术紧密相关的研究方向[133,134], 意图通过交互获得的项目序列(item sequence)来刻画用户偏好.

对于同质项目(homogeneous items)(如不同类型的文章), 注意力机制方法[132]已经可以区分不同历史项目对推荐目标项目的贡献程度. 但当用户记录存在过多噪音时, 注意力机制的效果会变差. 为了消除用户记录的噪音, Zhang 等人[135]将推荐问题形式化为顺序决策过程, 在由数据集和基础推荐模型构成的环境反馈下, 上层控制器判断用户记录是否需要修改, 下层控制器对需要修改的项目进行判定和删除. 该算法在慕课(open online courses, MOOCs)数据集中进行了验证, 结果显示, 可以有效消除用户噪音的影响. 此外, 为了克服项目数据过大和用户记录稀疏的问题, Wang 等人[136]提出了基于聚类的分层强化学习(clustering-based

reinforcement learning, CHRL)算法. 该算法首先对基础推荐系统进行预训练, 然后设计分层结构来过滤可能误导推荐系统的交互, 同时加入聚类策略, 以减少项目数据的稀疏问题.

相比于同质项目推荐系统, 综合推荐系统[137]需要在一个页面中同时推荐异质项目(heterogeneous item)(如文章和视频). Xie 等人[138]提出了综合推荐分层强化学习框架(HRL framework for integrated recommendation, HRL-Rec), 在该框架中: 上层控制器作为频道选择器, 负责在列表推荐器中生成频道序列; 下层控制器作为项目推荐器, 负责在频道列表中选择项目, 以此捕获用户不同粒度的偏好. 目前, 该方案已应用于微信“看一看”线上系统, 实现了 DHRL 在推荐系统领域的商业价值.

5.4 视频描述生成领域

视频描述(video captioning, VC)作为集视觉和文本两个维度的多模态任务, 具有更高的复杂度. 当前, 基于 DL 的视频描述方法通常利用自动编码器(auto-encoder), 来学习从视频序列到文本序列的转移过程[139], 但是这些方法往往只能提取到粗粒度的视频特征, 无法在噪音背景下捕获明确的对象, 损失了对重要内容的理解能力.

为了消除视频噪音, 提取细粒度的视频描述特征, Wang 等人[14]在 VC 领域中引入 DHRL, 将文本和视频语境视为强化学习环境, 定义任务为一个顺序决策过程. 在该算法中, 上层控制器为新文本片段产生子目标, 下层控制器按序列产生的单词来生成文本片段, 采用二元判定机制评估当前子目标是否被实现. 为了克服更具挑战的多语句生成问题, Huang 等人[140]提出了一种 DHRL 框架, 上层控制器为每个图像序列生成语义连贯的主题, 下层控制器根据主题, 使用语义合成网络生成句子描述, 将句子生成建立在主题的基础上. 该算法在视觉故事(visual storytelling, VIST)数据集上的评测结果表明, 其性能明显优于其他 DL 模型[141]. 此外, Chen 等人[142]首次将 DHRL 应用于视频摘要生成领域, 将整个任务分解成若干子任务, 通过定义子目标和内部奖励来解决稀疏奖励问题. 该算法在视频摘要数据集上的表现不仅超越了最先进的无监督方法[143], 甚至超越了它的有监督扩展方法[144].

6 深度分层强化学习展望

综上所述, 各类 DHRL 算法的成功正是源于分层抽象技术, 但现阶段的分层技术还远不如人类般有效. 从 DHRL 的不足之处出发, 为进一步实现更加智能、灵活和稳定的算法, 我们认为, 未来仍需朝以下几个方向发展.

(1) 在 DHRL 框架中实现参数共享

由于分层结构的存在, DHRL 的训练成本总是大于它的底层算法. 这一缺陷十分明显, 却未能得到学者们的重视, 目前也仅有极少论文[85]尝试使用全局共享参数来优化计算成本. 实际上, 在多任务 DL 领域中, 共享机制[145]已经得到了广泛应用, 它包括硬共享[146]、软共享[147]和分层共享[148]这 3 种形式, 不仅可以缩短训练时长, 还可以增强网络泛化性, 使同一个网络能够快速适应新任务. 因此, 未来的 DHRL 有必要引入多种类型的共享机制, 在缩短训练时长的同时, 充分共享下层知识, 提高信息的利用率.

(2) 在 DHRL 框架中引入课程式学习法

在面对奖励过于稀疏或顺序决策过于复杂的问题时, DHRL 算法也可能无法取得令人满意的效果, 或需要消耗大量的计算资源. 鉴于人类在面对复杂问题时采用由易到难的求解思路, DHRL 可以引入课程学习法, 辅助 O-DHRL 自动学习分层结构, 包括技能长度和技能数量; 辅助 G-DHRL 自动学习可实现的子目标, 突破当前状态覆盖边界, 将子目标拓展到未知领域. 在近一年的论文中, 引入课程式学习法已经初见成效[101,120], 未来需要实现更有效的融合机制, 在不同框架下, 减少课程学习对原算法的不稳定影响, 实现即插即用的嵌入方式.

(3) 发展信息论在 O-DHRL 框架中的应用

信息论是一门理论完备、论证严格和应用广泛的学科, 从 2017 年开始, O-DHRL 框架便引入信息论, 作为其约束条件或目标函数. 但目前的信息论多停留于发现状态、动作和技能的统计学特征上, 无法挖掘到更

深层的语义信息, 如发现 gym 中的人型模拟器运动规律. 因此, 推动信息论与 O-DHRL 的深入结合、加强先验知识的有效介入或采用更具语义信息的数据处理技术(如引入自监督[149]、对比学习[150]或注意力机制[151,152])找出信息的内在联系, 将有利于 DHRL 在理论层面的进一步提升, 增强技能的可解释性, 使 O-DHRL 模型有更好的可迁移能力和通用性.

(4) 发展异策略下的 G-DHRL 框架

在处理密集奖励任务时, 由于异策略 G-DHRL 分层同步训练的非稳定性问题, 绝大部分的 G-DHRL 都必须使用基于同策略的上层策略. 但如果能够消除或减少这种非稳定性问题, 异策略 G-DHRL 将拥有更高的采样效率和数据利用率. 尽管目前已有一些算法进行了尝试[42,73]且取得了不错效果, 但这些文献也指出, 异策略 DHRL 算法的性能与同策略标准下的性能仍有差距. 因此, 未来仍需探索如何设计更有效的子目标修正法或后见子目标法, 以解决上下层策略同步训练的非稳定性问题.

(5) 设计更符合分层策略的实验环境

目前常用的 DHRL 实验环境主要分为通用的实验平台(如 ALE 和 gym)以及专为 DHRL 设计的单一环境(如蚂蚁迷宫和猎豹越障). 前者多数环境不具备稀疏奖励或顺序决策条件, 一些优秀的 DHRL 算法在这些环境下无法展现出其应有的能力, 甚至难以和经典 DRL 算法匹敌. 后者通常是为算法自身特性所定制的, 无法为算法提供大量的可迁移能力验证途径, 将算法换一个环境或是将其他算法应用于该环境, 性能均难以令人满意. 此外, AO-DHRL 的技能多样性量化检验方法仍然缺失, 目前只能依靠人类经验进行判定. 所以, 为了提高 DHRL 方法在不同场景下的应用潜力, 保证不同 DHRL 算法性能对比的相对公平, 必须整合或提出更具一般性, 兼备稀疏奖励、顺序决策和可迁移能力验证条件的实验平台.

7 结束语

DHRL 作为目前 DRL 最热门的研究方向之一, 已经得到了越来越多的关注. 本文从 O-DHRL 和 G-DHRL 两个框架出发, 详细描述了 DHRL 的研究现状, 分析并对比了 O-DHRL 同步式技能训练方法(包括独立型技能和共享型技能)和异步式技能训练方法(包括技能学习和技能组合)的特点以及 G-DHRL 先见子目标训练方法(包括判定型子目标和引导型子目标)和后见子目标训练方法的特点. 尽管两个框架有着完全不同的核心思想和求解路线, 但它们的共性是十分明显的: 一方面, 这些方法几乎都采用了 SMDP 理论和时序抽象法, 所面对的问题也是经典 DRL 方法难以解决的稀疏奖励、顺序决策和弱迁移能力等问题; 另一方面, 这些方法是经典 DRL 方法的拓展, 仍需面对探索与利用、奖励函数设计和样本利用率等问题. 据我们所知: 在 DHRL 领域, 目前还没有采用这一划分方式的文献, 甚至没有梳理不同算法差异的论述, 对近年来较多重要 DHRL 研究工作的介绍和讨论也存在缺失. 而本文首次提出了一系列明确的划分依据, 总结不同类型算法的创新点与迭代思想, 使 DHRL 的研究脉络更为分明, 研究工作更加完整, 为 DHRL 的发展描绘了更加清晰的边界和路线.

综上所述, 有效的理论基础和多样的发展路线成就了 DHRL 的通用性、高性能和灵活性, 具体表现在: (1) 通过分层抽象技术, 将复杂困难的任务分解为若干个简单的、规模较小的任务, 可以在经典 DRL 算法无法解决的问题中取得优异表现; (2) 通过分层, 使不同层次的策略有不同水平的关注点; (3) 通过引入信息论, 扩展了 O-DHRL 的理论结构, 使智能体可以学到更具通用性的技能; (4) 通过子目标设定, 实现了具有强语义的上层策略, 隐式划分了任务的状态空间. 可以预见的是: 随着对 DHRL 研究的进一步深入, 融入更多的知识体系, 这一研究方向一定可以推动 DRL 方法以及人工智能在更广阔的领域发挥不可替代的作用.

致谢 本文工作受软件新技术与产业化协同创新中心部分资助.

References:

[1] Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge, 2018.

[2] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep Learning. Cambridge, 2016.

[3] Liu Q, Zhai JW, Zhang ZZ, Zhong S, Zhou Q, Zhang P, Xü J. A survey on deep reinforcement learning. Chinese Journal of Computers, 2018, 41(1): 1−27 (in Chinese with English abstract).

[4] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529−533.

[5] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations. 2016.

[6] Babaeizadeh M, Frosio I, Tyree S, Clemons J, Kautz J. Reinforcement learning through asynchronous advantage actor-critic on a GPU. In: Proc. of the Int'l Conf. on Learning Representations. 2017.

[7] Lai J, Wei JY, Chen XL. Overview of hierarchical reinforcement learning. Computer Engineering and Applications, 2021, 57(3): 72−79 (in Chinese with English abstract).

[8] Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence, 1999, 112(1−2): 181−211. [doi: 10.1016/s0004-3702(99)00052-1]

[9] Kulkarni TD, Narasimhan K, Saeedi A, Tenenbaum J. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In: Advances in Neural Information Processing Systems. MIT, 2016. 3675−3683.

[10] Tang H, Hao J, Lv T, Chen Y, Zhang Z, Jia H, Ren C, Zheng Y, Meng Z, Fan C. Hierarchical deep multiagent reinforcement learning with temporal abstraction. arXiv:1809.09332, 2018.

[11] Chen J, Wang Z, Tomizuka M. Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. In: Proc. of the IEEE Intelligent Vehicles Symp. IEEE, 2018. 1239−1244. [doi: 10.1109/ivs.2018.8500368]

[12] Liu J, Pan F, Luo L. Gochat: Goal-oriented chatbots with hierarchical reinforcement learning. In: Proc. of the Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 1793−1796. [doi: 10.1145/3397271.3401250]

[13] Zhao D, Zhang L, Zhang B, Zheng L, Bao Y, Yan W. MAHRL: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations. In: Proc. of the Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 871−880. [doi: 10.1145/3397271.3401170]

[14] Wang X, Chen W, Wu J, Wang YF, Wang WY. Video captioning via hierarchical reinforcement learning. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2018. 4213−4222. [doi: 10.1109/cvpr.2018.00443]

[15] Schaul T, Horgan D, Gregor K, Silver D. Universal value function approximators. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2015. 1312−1320.

[16] Bacon PL, Harb J, Precup D. The option-critic architecture. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2017. 1726−1734.

[17] Andrychowicz M, Wolski F, Ray A, Schneider J, Fong R, Welinder P, Mcgrew B, Tobin J, Pieter Abbeel O, Zaremba W. Hindsight experience replay. In: Advances in Neural Information Processing Systems. MIT, 2017. 5048−5058.

[18] Thrun S, Schwartz A. Finding structure in reinforcement learning. In: Advances in Neural Information Processing Systems. MIT, 1995. 385−392.

[19] Mahadevan S, Marchalleck N, Das TK, Gosavi A. Self-improving factory simulation using continuous-time average-reward reinforcement learning. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 1997. 202−210.

[20] Hauskrecht M, Meuleau N, Kaelbling LP, Dean T, Boutilier C. Hierarchical solution of markov decision processes using macro-actions. In: Proc. of the Conf. on Uncertainty in Artificial Intelligence. AUAI, 1998. 220−229.

[21] Yang WY, Bai CJ, Cai C, Zhao YN, Liu P. Survey on sparse reward in deep reinforcement learning. Computer Science, 2020, 47(3): 182−191 (in Chinese with English abstract).

[22] Watkins CJ, Dayan P. *Q*-learning. Machine Learning, 1992, 8(3−4): 279−292.

[23] Hasselt H. Double *q*-learning. In: Advances in Neural Information Processing Systems. MIT, 2010. 2613−2621.

[24] Rafati J, Noelle DC. Learning representations in model-free hierarchical reinforcement learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 10009−10010. [doi: 10.1609/aaai.v33i01.330110009]

[25] Murphy KP. A survey of POMDP solution techniques. Environment, 2000, 2: 3.

[26] Ye X, Yang Y. Hierarchical and partially observable goal-driven policy learning with goals relational graph. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. IEEE, 2021. 14101−14110. [doi: 10.1109/cvpr46437.2021.01388]

[27] Lee Y, Sun SH, Somasundaram S, Hu ES, Lim JJ. Composing complex skills by learning transition policies. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[28] Mousavi SS, Schukat M, Howley E. Deep reinforcement learning: An overview. In: Proc. of the SAI Intelligent Systems Conf. Springer, 2016. 426–440.

[29] Eysenbach B, Gupta A, Ibarz J, Levine S. Diversity is all you need: Learning skills without a reward function. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[30] Fox R, Krishnan S, Stoica I, Goldberg K. Multi-level discovery of deep options. arXiv:1703.08294, 2017.

[31] Mankowitz DJ, Mann TA, Mannor S. Iterative hierarchical optimization for misspecified problems (IHOMP). arXiv:1602.03348, 2016.

[32] Osa T, Sugiyama M. Hierarchical policy search via return-weighted density estimation. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2017. 3860–3867.

[33] Campos Camúñez V, Trott A, Xiong C, Socher R, Giró Nieto X, Torres Viñals J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2020. 1–17.

[34] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. OpenAI gym. arXiv:1606.01540, 2016.

[35] Hasselt HV, Guez A, Silver D. Deep reinforcement learning with double *q*-learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2016. 2094–2100.

[36] Kaiser L, Babaeizadeh M, Milos P, Osinski B, Campbell RH, Czechowski K, Erhan D, Finn C, Kozakowski P, Levine S. Model-based reinforcement learning for Atari. In: Proc. of the Int'l Conf. on Learning Representations. 2020.

[37] Reddy S, Dragan AD, Levine S. SQIL: Imitation learning via reinforcement learning with sparse rewards. In: Proc. of the Int'l Conf. on Learning Representations. 2019.

[38] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In: Proc. of the Int'l Conf. on Learning Representations. 2016.

[39] Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar M, Silver D. Rainbow: Combining improvements in deep reinforcement learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2018. 3215–3222.

[40] Duan Y, Chen X, Houthooft R, Schulman J, Abbeel P. Benchmarking deep reinforcement learning for continuous control. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2016. 1329–1338.

[41] Kreidieh AR, Berseth G, Trabucco B, Parajuli S, Levine S, Bayen AM. Inter-level cooperation in hierarchical reinforcement learning. arXiv:1912.02368, 2019.

[42] Nachum O, Gu SS, Lee H, Levine S. Data-efficient hierarchical reinforcement learning. In: Advances in Neural Information Processing Systems. MIT, 2018. 3303–3313.

[43] Tian Q, Wang G, Liu J, Wang D, Kang Y. Independent skill transfer for deep reinforcement learning. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence, 2019. 2901–2907. [doi: 10.24963/ijcai.2020/401]

[44] Gregor K, Rezende DJ, Wierstra D. Variational intrinsic control. arXiv:1611.07507, 2016.

[45] Manoharan A, Ramesh R, Ravindran B. Option encoder: A framework for discovering a policy basis in reinforcement learning. In: Proc. of the Machine Learning and Knowledge Discovery in Databases. Springer, 2020. 509–524. [doi: 10.1007/978-3-030-67661-2_30]

[46] Zahavy T, Hasidim A, Kaplan H, Mansour Y. Planning in hierarchical reinforcement learning: Guarantees for using local policies. In: Proc. of the Algorithmic Learning Theory. Springer, 2020. 906–934.

[47] Li C, Xia F, Martin-Martin R, Savarese S. HRL4in: Hierarchical reinforcement learning for interactive navigation with mobile manipulators. In: Proc. of the Conf. on Robot Learning. PMLR, 2020. 603–616.

[48] Florensa C, Duan Y, Abbeel P. Stochastic neural networks for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations. 2017. [doi: 10.1002/rcm.765]

[49] Konidaris G. Constructing abstraction hierarchies using a skill-symbol loop. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2016. 1648.

[50] Lyu D, Yang F, Liu B, Gustafson S. SDRL: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 2970–2977. [doi: 10.1609/aaai.v33i01.33012970]

[51] Kempka M, Wydmuch M, Runc G, Toczek J, Jaśkowski W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In: Proc. of the IEEE Conf. on Computational Intelligence and Games. IEEE, 2016. 1−8. [doi: 10.1109/cig.2016.7860433]

[52] Brittain M, Wei P. Hierarchical reinforcement learning with deep nested agents. arXiv:1805.07008, 2018.

[53] Khetarpal K, Klissarov M, Chevalier-Boisvert M, Bacon PL, Precup D. Options of interest: Temporal abstraction with interest functions. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2020. 4444−4451. [doi: 10.1609/aaai.v34i04.5871]

[54] Beattie C, Leibo JZ, Teplyashin D, Ward T, Wainwright M, Küttler H, Lefrancq A, Green S, Valdés V, Sadik A. Deepmind lab. arXiv:1612.03801, 2016.

[55] Jaderberg M, Mnih V, Czarnecki WM, Schaul T, Leibo JZ, Silver D, Kavukcuoglu K. Reinforcement learning with unsupervised auxiliary tasks. In: Proc. of the Int'l Conf. on Learning Representations. 2017.

[56] Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2016. 1928−1937.

[57] Li S, Wang R, Tang M, Zhang C. Hierarchical reinforcement learning with advantage-based auxiliary rewards. In: Advances in Neural Information Processing Systems. MIT, 2019. 1409−1419.

[58] Nachum O, Gu S, Lee H, Levine S. Near-optimal representation learning for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[59] Li AC, Florensa C, Clavera I, Abbeel P. Sub-policy adaptation for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations. 2019.

[60] Dietterich TG. The MAXQ method for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 1998. 118−126.

[61] Sohn S, Oh J, Lee H. Hierarchical reinforcement learning for zero-shot generalization with subtask dependencies. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. Springer, 2018. 7156−7166.

[62] Esteban D, Rozo L, Caldwell DG. Hierarchical reinforcement learning for concurrent discovery of compound and composable policies. In: Proc. of the IEEE Int'l Conf. on Intelligent Robots and Systems. IEEE, 2019. 1818−1825. [doi: 10.1109/iros40897.2019.8968149]

[63] Kokel H, Manoharan A, Natarajan S, Ravindran B, Tadepalli P. Reprel: Integrating relational planning and reinforcement learning for effective abstraction. In: Proc. of the Int'l Conf. on Automated Planning and Scheduling. AAAI, 2021. 533−541.

[64] Machado MC, Bellemare MG, Bowling M. A Laplacian framework for option discovery in reinforcement learning. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2017. 2295−2304.

[65] Co-Reyes JD, Liu Y, Gupta A, Eysenbach B, Abbeel P, Levine S. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. arXiv:1806.02813, 2018.

[66] Frans K, Ho J, Chen X, Abbeel P, Schulman J. Meta learning shared hierarchies. In: Proc. of the Int'l Conf. on Learning Representations. 2017.

[67] Baumli K, Warde-Farley D, Hansen S, Mnih V. Relative variational intrinsic control. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2021. 6732−6740.

[68] Dukkipati A, Banerjee R, Ayyagari RS, Udaybhai DP. Stay alive with many options: A reinforcement learning approach for autonomous navigation. arXiv:2102.00168, 2021.

[69] Jain A, Khetarpal K, Precup D. Safe option-critic: Learning safety in the option-critic architecture. The Knowledge Engineering Review, 2021, 36. [doi: 10.1017/s0269888921000035]

[70] Zhang J, Yu H, Xu W. Hierarchical reinforcement learning by discovering intrinsic options. In: Proc. of the Int'l Conf. on Learning Representations. 2021.

[71] Ghazanfari B, Mozayani N. Extracting bottlenecks for reinforcement learning agent by holonic concept clustering and attentional functions. Expert Systems with Applications, 2016, 54: 61−77. [doi: 10.1016/j.eswa.2016.01.030]

[72] Guo X, Zhai Y. *K*-means clustering based reinforcement learning algorithm for automatic control in robots. Int'l Journal of Simulation: Systems, 2016, 17: 24. [doi: 10.5013/ijssst.a.17.24.06]

[73] Levy A, Konidaris G, Platt R, Saenko K. Learning multi-level hierarchies with hindsight. In: Proc. of the Int'l Conf. on Learning Representations. 2017.

[74] Dilokthanakul N, Kaplanis C, Pawlowski N, Shanahan M. Feature control as intrinsic motivation for hierarchical reinforcement learning. IEEE Trans. on Neural Networks and Learning Systems, 2019, 30(11): 3409−3418. [doi: 10.1109/tnnls.2019.2891792]

[75] Vezhnevets AS, Osindero S, Schaul T, Heess N, Jaderberg M, Silver D, Kavukcuoglu K. Feudal networks for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2017. 3540−3549.

[76] Szepesvari C, Sutton RS, Modayil J, Bhatnagar S. Universal option models. In: Advances in Neural Information Processing Systems. MIT, 2014. 990−998.

[77] Jothimurugan K, Bastani O, Alur R. Abstract value iteration for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Artificial Intelligence and Statistics. PMLR, 2021. 1162−1170.

[78] Chane-Sane E, Schmid C, Laptev I. Goal-conditioned reinforcement learning with imagined subgoals. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2021. 1430−1440.

[79] Nachum O, Tang H, Lu X, Gu S, Lee H, Levine S. Why does hierarchy (sometimes) work so well in reinforcement learning? arXiv: 1909.10618, 2019.

[80] Sutton RS, Precup D, Singh SP. Intra-option learning about temporally abstract actions. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 1998. 556−564.

[81] Harb J, Bacon PL, Klissarov M, Precup D. When waiting is not an option: Learning options with a deliberation cost. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2018. 3165−3172.

[82] Konda VR, Tsitsiklis JN. Actor-critic algorithms. In: Advances in Neural Information Processing Systems. MIT, 2000. 1008−1014.

[83] Zhu F, Zhu HJ, Liu Q, Chen DH, Fu YC. True online natural actor-critic algorithm for the continuous space problem. Ruan Jian Xue Bao/Journal of Software, 2018, 29(2): 267−282 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5251.htm [doi: 10.13328/j.cnki.jos.005251]

[84] Riemer M, Liu M, Tesauro G. Learning abstract options. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. Springer, 2018. 10445−10455.

[85] Riemer M, Cases I, Rosenbaum C, Liu M, Tesauro G. On the role of weight sharing during deep option learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2020. 5519−5526. [doi: 10.1609/aaai.v34i04.6003]

[86] Osa T, Tangkaratt V, Sugiyama M. Hierarchical reinforcement learning via advantage-weighted information maximization. In: Proc. of the Int'l Conf. on Learning Representations. 2019.

[87] Hou Z, Zhang K, Wan Y, Li D, Fu C, Yu H. Off-policy maximum entropy reinforcement learning: Soft actor-critic with advantage weighted mixture policy (SAC-awmp). arXiv:2002.02829, 2020.

[88] Kamat A, Precup D. Diversity-enriched option-critic. arXiv:2011.02565, 2020.

[89] Li C, Ma X, Zhang C, Yang J, Xia L, Zhao Q. Soac: The soft option actor-critic architecture. arXiv:2006.14363, 2020.

[90] Klissarov M, Precup D. Flexible option learning. In: Advances in Neural Information Processing Systems. MIT, 2021.

[91] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2017. 1126−1135.

[92] Zhao KL, Zhan XL, Wang YZ. Survey on few-shot learning. Ruan Jian Xue Bao/Journal of Software, 2021, 32(2): 349−369 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6138.htm [doi: 10.13328/j.cnki.jos.006138]

[93] Song Y, Wang J, Lukasiewicz T, Xu Z, Xu M. Diversity-driven extensible hierarchical reinforcement learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 4992−4999. [doi: 10.1609/aaai.v33i01.33014992]

[94] Song S, Weng J, Su H, Yan D, Zou H, Zhu J. Playing FPS games with environment-aware hierarchical reinforcement learning. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2019. 3475−3482.

[95] Klyubin AS, Polani D, Nehaniv CL. Empowerment: A universal agent-centric measure of control. In: Proc. of the IEEE Congress on Evolutionary Computation. IEEE, 2005. 128−135. [doi: 10.1109/cec.2005.1554676]

[96] Salge C, Glackin C, Polani D. Empowerment—An Introduction. Epping, 2014. https://arxiv.org/pdf/1310.1863.pdf

[97] Savinov N, Raichuk A, Marinier R, Vincent D, Pollefeys M, Lillicrap T, Gelly S. Episodic curiosity through reachability. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[98] Burda Y, Edwards H, Pathak D, Storkey A, Darrell T, Efros AA. Large-scale study of curiosity-driven learning. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[99] Kumar NM. Empowerment-driven exploration using mutual information estimation. arXiv preprint arXiv: 1810.05533, 2018.

[100] Dai S, Xu W, Hofmann A, Williams B. An empowerment-based solution to robotic manipulation tasks with sparse rewards. In: Proc. of the Robotics: Science and Systems. 2021. [doi: 10.15607/rss.2021.xvii.001]

[101] Achiam J, Edwards H, Amodei D, Abbeel P. Variational option discovery algorithms. arXiv:1807.10299, 2018.

[102] Lee L, Eysenbach B, Parisotto E, Xing E, Levine S, Salakhutdinov R. Efficient exploration via state marginal matching. arXiv: 1906.05274, 2019.

[103] Van Den Oord A, Vinyals O, Kavukcuoglu K. Neural discrete representation learning. In: Advances in Neural Information Processing Systems. MIT, 2017. 6309−6318.

[104] Trott A, Zheng S, Xiong C, Socher R. Keeping your distance: Solving sparse reward tasks using self-balancing shaped rewards. In: Advances in Neural Information Processing Systems. MIT, 2019. 10376−10386.

[105] Dhariwal P, Hesse C, Klimov O, Nichol A, PlapPert M. Openai baselines. 2017. https://github.com/openai/baselines

[106] Fujimoto S, Van Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2018. 1582−1591.

[107] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2018. 1861−1870.

[108] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: Proc. of the Int'l Conf. on Machine Learning. ACM, 2015. 1889−1897.

[109] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

[110] Furelos-Blanco D, Law M, Russo A, Broda K, Jonsson A. Induction of subgoal automata for reinforcement learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2020. 3890−3897. [doi: 10.1609/aaai.v34i04.5802]

[111] Chen D, Yan Q, Guo S, Yang Z, Su X, Chen F. Learning effective subgoals with multi-task hierarchical reinforcement learning. In: Proc. of the Scaling-up Reinforcement Learning Workshop. 2019.

[112] Dayan P, Hinton GE. Feudal reinforcement learning. In: Advances in Neural Information Processing Systems. MIT, 1993. 271−278.

[113] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735−1780.

[114] Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. In: Proc. of the Int'l Conf. on Learning Representations. 2015.

[115] Wang R, Yu R, An B, Rabinovich Z. I^2HRL: Interactive influence-based hierarchical reinforcement learning. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence. 2020. 3131−3138.

[116] Li R, Cai Z, Huang T, Zhu W. Anchor: The achieved goal to replace the subgoal for hierarchical reinforcement learning. Knowledge-based Systems, 2021, 225: 107128. [doi: 10.1016/j.knosys.2021.107128]

[117] Zhang T, Guo S, Tan T, Hu X, Chen F. Generating adjacency-constrained subgoals in hierarchical reinforcement learning. In: dvances in Neural Information Processing Systems. MIT, 2020. 85−114.

[118] Fu H, Tang H, Hao J, Liu W, Chen C. MGHRL: Meta goal-generation for hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Distributed Artificial Intelligence. Springer, 2020. 29−39. [doi: 10.1007/978-3-030-64096-5_3]

[119] Li S, Zheng L, Wang J, Zhang C. Learning subgoal representations with slow dynamics. In: Proc. of the Int'l Conf. on Learning Representations. 2020.

[120] He Z, Gu C, Xu R, Wu K. Automatic curriculum generation by hierarchical reinforcement learning. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. Springer, 2020. 202−213. [doi: 10.1007/978-3-030-63833-7_17]

[121] Röder F, Eppe M, Nguyen PD, Wermter S. Curious hierarchical actor-critic reinforcement learning. In: Proc. of the Int'l Conf. on Artificial Neural Networks. Springer, 2020. 408−419. [doi: 10.1007/978-3-030-61616-8_33]

[122] Friston K, Mattout J, Kilner J. Action understanding and active inference. Biological Cybernetics, 2011, 104(1): 137−160. [doi: 10.1007/s00422-011-0424-z]

[123] Zhou X, Bai T, Gao Y, Han Y. Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning. Sensors, 2019, 19(7): 1576. [doi: 10.3390/s19071576]

[124] Jain D, Iscen A, Caluwaerts K. Hierarchical reinforcement learning for quadruped locomotion. In: Proc. of the IEEE Int'l Conf. on Intelligent Robots and Systems. IEEE, 2019. 7551−7557. [doi: 10.1109/iros40897.2019.8967913]

[125] Li T, Lambert N, Calandra R, Meier F, Rai A. Learning generalizable locomotion skills with hierarchical reinforcement learning. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation. IEEE, 2020. 413−419. [doi: 10.1109/icra40945.2020.9196642]

[126] Budzianowski P, Ultes S, Su PH, Mrkšić N, Wen TH, Casanueva I, Rojas-Barahona L, Gašić M. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In: Proc. of the Annual SIGdial Meeting on Discourse and Dialogue. ACL, 2017. 86−92. [doi: 10.18653/v1/w17-5512]

[127] Saha T, Gupta D, Saha S, Bhattacharyya P. Towards integrated dialogue policy learning for multiple domains and intents using hierarchical deep reinforcement learning. Expert Systems with Applications, 2020, 162: 113650. [doi: 10.1016/j.eswa.2020.113650]

[128] Saha T, Saha S, Bhattacharyya P. Towards sentiment aided dialogue policy learning for multi-intent conversations using hierarchical reinforcement learning. PloS ONE, 2020, 15(7): 1−28. [doi: 10.1371/journal.pone.0235367]

[129] Yu L, Zhang W, Wang J, Yu Y. SeqGan: Sequence generative adversarial nets with policy gradient. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2017. 2852−2858.

[130] Ghandeharioun A, Shen JH, Jaques N, Ferguson C, Jones N, Lapedriza A, Picard R. Approximating interactive human evaluation with self-play for open-domain dialog systems. In: Proc. of the Annual Conf. on Neural Information Processing Systems. MIT, 2019. 13658−13669.

[131] Saleh A, Jaques N, Ghandeharioun A, Shen J, Picard R. Hierarchical reinforcement learning for open-domain dialog. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2020. 8741−8748. [doi: 10.1609/aaai.v34i05.6400]

[132] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the Int'l Conf. on Neural Information Processing Systems. Springer, 2017. 6000−6010.

[133] Tang X, Chen Y, Li X, Liu J, Ying Z. A reinforcement learning approach to personalized learning recommendation systems. British Journal of Mathematical and Statistical Psychology, 2019, 72(1): 108−135. [doi: 10.1111/bmsp.12144]

[134] Wang P, Fan Y, Xia L, Zhao WX, Niu S, Huang J. Kerl: A knowledge-guided reinforcement learning model for sequential recommendation. In: Proc. of the Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 209−218. [doi: 10.1145/3397271.3401134]

[135] Zhang J, Hao B, Chen B, Li C, Chen H, Sun J. Hierarchical reinforcement learning for course recommendation in Moocs. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 435−442. [doi: 10.1609/aaai.v33i01.3301435]

[136] Wang X, Wang Y, Guo L, Xu L, Gao B, Liu F, Li W. Exploring clustering-based reinforcement learning for personalized book recommendation in digital library. Information, 2021, 12(5): 198. [doi: 10.3390/info12050198]

[137] Zhang Y, Ai Q, Chen X, Croft WB. Joint representation learning for top-*n* recommendation with heterogeneous information sources. In: Proc. of the ACM on Conf. on Information and Knowledge Management. ACM, 2017. 1449−1458. [doi: 10.1145/3132847.3132892]

[138] Xie R, Zhang S, Wang R, Xia F, Lin L. Hierarchical reinforcement learning for integrated recommendation. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2021. 4521−4528.

[139] Shetty R, Laaksonen J. Frame-and segment-level features and candidate pool evaluation for video caption generation. In: Proc. of the ACM Int'l Conf. on Multimedia. ACM, 2016. 1073−1076. [doi: 10.1145/2964284.2984062]

[140] Huang Q, Gan Z, Celikyilmaz A, Wu D, Wang J, He X. Hierarchically structured reinforcement learning for topically coherent visual story generation. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 8465−8472. [doi: 10.1609/aaai.v33i01.33018465]

[141] Huang TH, Ferraro F, Mostafazadeh N, Misra I, Agrawal A, Devlin J, Girshick R, He X, Kohli P, Batra D. Visual storytelling. In: Proc. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. ACL, 2016. 1233−1239.

[142] Chen Y, Tao L, Wang X, Yamasaki T. Weakly supervised video summarization by hierarchical reinforcement learning. In: Proc. of the ACM Multimedia Asia. ACM, 2019. 1−6. [doi: 10.1145/3338533.3366583]

[143] Zhang K, Chao WL, Sha F, Grauman K. Video summarization with long short-term memory. In: Proc. of the European Conf. on Computer Vision. Springer, 2016. 766−782. [doi: 10.1007/978-3-319-46478-7_47]

[144] Zhou K, Qiao Y, Xiang T. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2018. 7582−7589.

[145] Sun T, Shao Y, Li X, Liu P, Yan H, Qiu X, Huang X. Learning sparse sharing architectures for multiple tasks. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2020. 8936−8943. [doi: 10.1609/aaai.v34i05.6424]

[146] Subramanian S, Trischler A, Bengio Y, Pal CJ. Learning general purpose distributed sentence representations via large scale multi-task learning. In: Proc. of the Int'l Conf. on Learning Representations. 2018.

[147] Ruder S, Bingel J, Augenstein I, Søgaard A. Latent multi-task architecture learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. AAAI, 2019. 4822−4829. [doi: 10.1609/aaai.v33i01.33014822]

[148] Søgaard A, Goldberg Y. Deep multi-task learning with low level tasks supervised at lower layers. In: Proc. of the Annual Meeting of the Association for Computational Linguistics. ACL, 2016. 231−235. [doi: 10.18653/v1/p16-2038]

[149] Oord AVD, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv:1807.03748, 2018.

[150] Laskin M, Lee K, Stooke A, Pinto L, Abbeel P, Srinivas A. Reinforcement learning with augmented data. In: Advances in Neural Information Processing Systems. MIT, 2020.

[151] Qureshi AH, Johnson JJ, Qin Y, Henderson T, Boots B, Yip MC. Composing task-agnostic policies with deep reinforcement learning. In: Proc. of the Int'l Conf. on Learning Representations. 2020.

[152] Liang XX, Feng YH, Huang JC, Wang Q, Ma Y, Liu Z. Novel deep reinforcement learning algorithm based on attention-based value function and autoregressive environment model. Ruan Jian Xue Bao/Journal of Software, 2020, 31(4): 948−966 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5930.htm [doi: 10.13328/j.cnki.jos.005930]

附中文参考文献:

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

[7] 赖俊, 魏竞毅, 陈希亮. 分层强化学习综述. 计算机工程与应用, 2021, 57(3): 72−79.

[21] 杨惟轶, 白辰甲, 蔡超, 等. 深度强化学习中稀疏奖励问题研究综述. 计算机科学, 2020, 47(3): 182−191.

[83] 朱斐, 朱海军, 刘全, 陈冬火, 伏玉琛. 一种解决连续空间问题的真实在线自然梯度 AC 算法. 软件学报, 2018, 29(2): 267−282. http://www.jos.org.cn/1000-9825/5251.htm [doi: 10.13328/j.cnki.jos.005251]

[92] 赵凯琳, 靳小龙, 王元卓. 小样本学习研究综述. 软件学报, 2021, 32(2): 349−369. http://www.jos.org.cn/1000-9825/6138.htm [doi: 10.13328/j.cnki.jos.006138]

[152] 梁星星, 冯旸赫, 黄金才, 王琦, 马扬, 刘忠. 基于自回归预测模型的深度注意力强化学习方法. 软件学报, 2020, 31(4): 948−966. http://www.jos.org.cn/1000-9825/5930.htm [doi: 10.13328/j.cnki.jos.005930]

黄志刚(1993－), 男, 博士生, 主要研究领域为分层强化学习, 深度强化学习.

曹家庆(1994－), 男, 博士生, CCF 学生会员, 主要研究领域为强化学习.

刘全(1969－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为强化学习, 深度强化学习, 自动推理.

朱斐(1978－), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为强化学习, 深度强化学习, 文本挖掘.

张立华(1992－), 男, 博士生, CCF 学生会员, 主要研究领域为逆向强化学习, 深度强化学习.