软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007184] [CSTR: 32375.14.jos.007184]

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

基于分组对比学习的序贯感知技能发现*

杨尚东[1,2,3], 余淼盈[1], 陈兴国[1], 陈 蕾[1]

[1](南京邮电大学 计算机学院、软件学院、网络空间安全学院, 江苏 南京 210023)

[2](计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)

[3](广西多源信息挖掘与安全重点实验室 (广西师范大学), 广西 桂林 541004)

通信作者: 陈蕾, E-mail: chenlei@njupt.edu.cn

摘 要: 强化学习在智能对话系统等决策任务中取得了令人瞩目的结果, 但其在复杂的、奖励稀疏的任务中学习效率较低. 研究人员在强化学习中引入技能发现框架, 以最大化不同技能间的差异为目标构建技能策略, 提升了智能体在上述任务中的学习效率. 然而, 受到采样轨迹数据多样性的限制, 现有的技能发现方法局限于在一个强化学习回合中学习一种技能, 导致其在一回合中具有序贯技能组合的复杂任务中表现欠佳. 针对该问题, 提出一种基于分组对比学习的序贯感知技能发现方法 (group-wise contrastive learning based sequence-aware skill discovery, GCSSD), 该方法将对比学习融合到技能发现框架中. 首先, 为了提升轨迹数据的多样性, 将与环境交互的完整轨迹分段并进行分组, 利用分组轨迹构建对比损失学习技能嵌入表征; 其次, 结合技能嵌入表征与强化学习进行技能策略训练; 最后, 为了提升在具有不同序贯技能组合任务上的性能, 对采样轨迹进行分段技能表征并将其嵌入策略网络, 实现对已学技能策略的序贯组合. 实验结果表明, GCSSD 方法在具有序贯技能组合的稀疏奖励任务中具有较好的训练效果, 并且在具有与训练任务不同的序贯技能组合任务中, 能够利用已学技能对该任务进行快速适应.

关键词: 强化学习; 轨迹分组; 对比学习; 序贯感知; 技能发现

中图法分类号: TP18

中文引用格式: 杨尚东, 余淼盈, 陈兴国, 陈蕾. 基于分组对比学习的序贯感知技能发现. 软件学报. http://www.jos.org.cn/1000-9825/7184.htm

英文引用格式: Yang SD, Yu MY, Chen XG, Chen L. Group-wise Contrastive Learning Based Sequence-aware Skill Discovery. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7184.htm

Group-wise Contrastive Learning Based Sequence-aware Skill Discovery

YANG Shang-Dong[1,2,3], YU Miao-Ying[1], CHEN Xing-Guo[1], CHEN Lei[1]

[1](School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

[2](State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

[3](Guangxi Key Lab of Multi-source Information Mining & Security (Guangxi Normal University), Guilin 541004, China)

Abstract: Reinforcement learning has achieved remarkable results in decision-making tasks like intelligent dialogue systems, yet its efficiency diminishes notably in scenarios with intricate structures and scarce rewards. Researchers have integrated the skill discovery framework into reinforcement learning, aiming to maximize skill disparities to establish policies and boost agent performance in such tasks. However, the constraint posed by the limited diversity of sampled trajectory data confines existing skill discovery methods to learning a single skill per reinforcement learning episode. Consequently, this limitation results in subpar performance in complex tasks requiring sequential skill combinations within a single episode. To address this challenge, a group-wise contrastive learning based sequence-aware skill discovery method (GCSSD) is proposed, which integrates contrastive learning into the skill discovery framework. Initially, to augment

* 基金项目: 国家自然科学基金 (62206133, 62276142); 江苏省重点研发计划 (BE2021093); 南京大学计算机软件新技术国家重点实验室资助项目 (KFKT2022B12); 广西多源信息挖掘与安全重点实验室开放基金 (MIMS22-01); 江苏省双创博士项目 (JSSCBS20210539)
收稿时间: 2023-09-20; 修改时间: 2023-12-25; 采用时间: 2024-03-27; jos 在线出版时间: 2024-11-20

trajectory data diversity, the complete trajectories interacting with the environment are segmented and grouped, employing contrastive loss to learn skill embedding representations from grouped trajectories. Subsequently, skill policy training is conducted by combining the skill embedding representation with reinforcement learning. Lastly, to enhance performance in tasks featuring diverse sequential skill combinations, the sampled trajectories are segmented into skill representations and embedded into the learned policy network, facilitating the sequential combination of learned skill policies. Experimental results demonstrate the efficacy of the GCSSD method in tasks characterized by sparse rewards and sequential skill combinations, showcasing its capability to swiftly adapt to tasks with varying sequential skill combinations using learned skills.

Key words: reinforcement learning; trajectory grouping; contrastive learning; sequence-aware; skill discovery

在强化学习 (reinforcement learning, RL) 中, 智能体 (agent) 与未知环境交互获得奖励信号, 目的是学习到一个最大化累计奖励的策略. 近年来, 伴随神经网络技术的发展, 强化学习已广泛应用于智能对话系统[1−3]、机器人控制[4,5]等任务中. 然而, 与可以通过较低代价获得交互样本的场景不同, 在诸如自动驾驶[6]、无人机对抗[7]等应用中, 获取大量的、奖励稠密的样本代价较高, 直接采用经典的强化学习算法并不能获得让人满意的效果.

研究者们对上述问题从不同的视角开展了较为广泛的研究. 如从挖掘任务间关联的视角, 开展了多任务强化学习 (multi-task RL) 研究[8,9]; 从元知识的获取与泛化视角, 开展了元强化学习 (meta RL) 研究[10]; 从利用无需与环境在线交互的视角, 开展了离线强化学习 (offline RL) 研究[11]; 从策略演进与提升的视角, 开展了迁移强化学习 (transfer RL)、终身强化学习 (lifelong RL) 研究[12,13]. 当前, 最新的研究成果往往从上述多个视角出发解决上述问题, 其中, 结合了元强化学习与迁移学习等思想的技能发现 (skill discovery) 方法是当前的一个研究热点.

强化学习中的技能发现方法, 受到了人类有能力通过结合过去在相似任务中学习到的多种技能来完成新任务这一特点的启发. 与目前大多数直接针对训练任务进行端到端策略优化的强化学习方法[14−16]不同, 基于技能发现的强化学习方法通过学习与使用由原子动作组成的策略抉择 (option)[17], 本文称之为技能, 来解决复杂的长期任务. 若任务间具有类似的技能, 则智能体面临类似的新任务时, 他们可以高效地结合这些技能完成这项任务. 技能发现可以通过抉择框架形式化[17], 该框架用抉择的概念定义了原子动作组成的动作序列. 为了便于学习, 抉择或技能通常通过在普通策略中引入技能潜在参数 z 来制定, 从而形成一种形式为 $\pi(a \mid s,z)$ 的技能策略, 在强化学习一个回合内的多个时间步或整个回合内保持相同的 z [18−20]. 然而, 现有的技能发现方法有以下方面不足. 首先, 这些方法中的技能需要额外的人工定义或数据进行学习[18,21,22]. 例如, 预先手动设计技能和正确的关于任务的技能组合, 以便更高级别的策略学习, 或者通过大量演示数据学习技能. 其次, 这方法中的问题设定较为简单. 他们在固定的环境中学习技能, 每个回合目标相对固定[19,20,23]、一个回合内具有一种技能时, 学习效果较好. 但对于一个回合内存在多个目标, 且不同回合间的目标参数不同, 导致一个回合内存在序贯技能组合的任务, 这类方法存在明显不足. 为此, 本文关注稀疏奖励下的多目标强化学习问题, 并设计面向该问题的技能发现方法. 该方法可在上述复杂的训练任务中学习到不同技能, 并且在测试任务中可利用已学技能进行序贯组合.

具体而言, 我们提出了一种基于分组对比学习的序贯感知技能发现方法 (GCSSD). GCSSD 方法包含 3 个部分, 即基于分组对比学习的技能嵌入表征、基于技能表征的强化学习训练和测试场景中的技能序贯组合. 第 1 部分, 我们认为, 智能体在采样策略下的轨迹包含了序贯技能信息, 则对轨迹进行分段并将分段后的轨迹进行分组, 分组后的轨迹可能包含技能信息. 因此, 使用对比学习对分组轨迹进行编码, 并与其他分组的轨迹进行比较, 以获得技能的嵌入表征. 第 2 部分, 将技能嵌入表征按照技能和任务的对应信息进行组合, 作为智能体策略的联合输入, 指导其执行与任务相关的技能. 采用联合训练嵌入网络和策略网络, 以便在很少人工设计的情况下学习技能和任务相关技能序贯组合. 第 3 部分, 当与训练任务不同的序贯技能组合的测试任务时, 利用所学策略对该任务采样轨迹与分组编码, 自适应获得新任务的技能嵌入表征作为已学习策略的联合输入. 我们在两个实验场景中与目前的基准强化学习算法和基准技能发现算法进行了比较, 训练和测试实验的结果表明了本文所提方法的有效性.

本文的主要贡献主要包括以下 3 点.

(1) 面向实际应用中存在的奖励稀疏、技能序贯组合挑战, 提出了稀疏奖励下多目标导向强化学习问题, 针对稀疏奖励, 通过分组对比学习的方式学习技能嵌入表征.

(2) 此外, 针对多目标导向强化学习问题, 分别在训练、测试阶段对分段轨迹进行序贯技能嵌入表征, 并结合

策略网络实现序贯技能策略的训练和高效利用.

(3) 实验结果表明, 在离散和连续控制任务中, 所提算法可以有效表征强化学习一个回合内的多种技能, 并且能够在测试任务实现对技能序贯组合.

本文第 1 节介绍强化学习中技能发现的相关方法和研究现状. 第 2 节介绍本文所要解决的问题定义, 主要为稀疏奖励下的多目标导向强化学习. 第 3 节介绍本文提出的基于分组对比学习的序贯感知技能发现方法 GCSSD. 第 4 节通过在典型场景中的实验验证了所提方法的有效性. 最后总结全文.

1 相关工作

在本节中, 我们主要介绍和本文研究内容相关的研究工作.

● 基于抉择框架的分层强化学习. 为了促进长期行为的学习, 在基于抉择的框架[17,18]中, 智能体可以在一个回合中在不同抉择之间切换, 其中抉择通过带有终止条件的策略转换为动作序列. 基于抉择的分层强化学习建模 MDP 中的层次结构[24−27]. 对于典型的两级层次结构, 较高级别的策略产生抉择, 较低级别的策略输出原子动作. 然而, 这方法面对稀疏奖励任务适用性较弱, 且问题设定较为简单, 他们在固定的环境中学习技能, 每个回合目标相对固定. 当面临多任务中不断变化的环境动态时, 他们必须手动设计辅助子任务学习抉择.

● 基于技能发现的强化学习. 针对稀疏奖励的强化学习任务, 技能发现方法从智能体轨迹信息中发现技能. 其中, DIAYN[19]和 VIC[28]通过最大化轨迹与其相应技能之间的互信息来学习技能. VALOR[22]通过最大化抉择的概率来学习抉择, 并给出其结果观察轨迹. DADS[20]通过最大化智能体执行动作后的状态和技能表征之间的互信息学习技能. 这些方法在一个回合内具有一种技能时, 学习效果较好, 但对于一个回合内存在多个目标, 导致一个回合内存在序贯技能组合的任务, 这类方法存在不足.

● 对比学习. 对比学习是一种自监督的学习方法, 旨在从无标注的数据中学习特征表示, 并用于下游类似任务中. 近年来, CPC[29]、MoCo[30]、SimCLR[31]、SimSiam[32]和 BYOL[33]等对比学习方法在计算机视觉领域取得了许多成功. 本文参考 SimCLR 中的损失函数, 因其是一种简单有效的框架. 尽管有许多新的工作来改进 SimCLR, 但我们仍然选择了 SimCLR 作为本文对比学习的框架, 因为我们的目的不在于改进对比学习本身, 我们参考了 SimCLR 的核心的对比损失构建部分.

● 目标导向强化学习. 本文研究目标导向强化学习[34], 在该问题中, 智能体很少获得奖励. 目标导向的强化学习中, 智能体的目标是学习一个在给定状态和目标的联合输入下的最优策略, 并在诸多工作了开展了相关研究[35−38]. 尽管提出了一些重新标记[39]等技术来解决学习目标导向强化学习中的稀疏奖励问题, 但在一些大规模决策任务中, 学习上述目标导向的强化学习策略仍然存在挑战[40]. 本文参考 UVFA[35], 将目标与状态联合作为强化学习策略输入.

● 稀疏奖励强化学习. 稀疏奖励强化学习一直是一个热点的研究领域, 研究人员通过分层建模[24−27]或鼓励探索[41,42]来解决顺序决策中的稀疏奖励问题. 在本文中, 我们所提的序贯感知技能发现方法也可以被视为解决具有挑战性的稀疏奖励问题的一种潜在有效解决方案.

综上所述, 本文面向实际应用中存在的奖励稀疏、技能序贯组合挑战, 提出稀疏奖励下的多目标导向强化学习问题, 该问题的示意图如图 1 所示. 多目标导致的不同序贯技能组合构成不同的任务类型, 一个任务从一种任务类型中采样产生, 一个任务回合内包括多个目标, 同一任务类型下的任务中目标参数可能不同, 完成一个或者多个目标构成了一种技能, 对于智能体而言, 训练和测试时的任务类型可能不同.

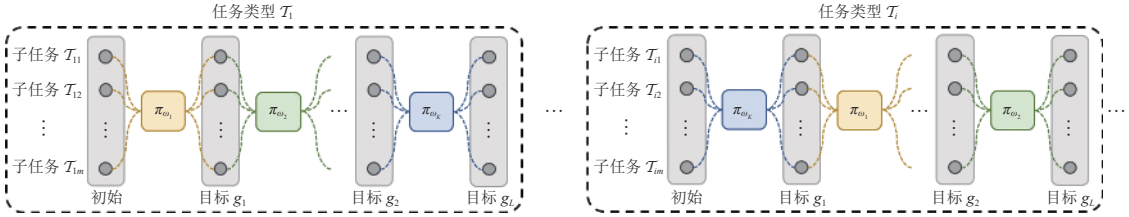

图 1 稀疏奖励下的多目标导向强化学习示意图

2 问题定义

在本节中, 主要介绍本文研究的稀疏奖励下的多目标导向强化学习问题定义. 在介绍该定义之前, 首先介绍强化学习的定义, 及强化学习中技能定义与技能发现的求解目标.

2.1 强化学习

在强化学习中, 智能体的目标是在训练阶段通过与未知环境的交互, 学习到一个可以获得最高累计奖励的策略, 这个交互过程可形式化地定义为马尔可夫决策过程 (Markov decision process, MDP). 与之前基于技能的强化学习研究[18]类似, 本文研究无限视界带折扣的 MDP (infinite discounted MDP), 我们先给出该 MDP 的基本定义和相关的符号解释. 之后, 再给出技能定义和本文要解决的具体任务.

定义 1. 马尔可夫决策过程 (MDP). MDP 可以用五元组 $\langle \mathcal{S},\mathcal{A},R,T,\gamma\rangle$ 描述, 其中, $\mathcal{S}$ 是包含 $|\mathcal{S}|$ 个状态的状态空间, $\mathcal{A}$ 是包含 $|\mathcal{A}|$ 个可选动作的动作空间, $R:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}$ 是产生即时奖励的奖励函数, $T:\mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ 是状态转移函数, $\gamma\in[0,1)$ 是折扣因子.

对于一个具体的强化学习任务而言, 在一个回合 (episode) 开始前, 智能体位于初始状态 s_0, 在第 t 个时间步, 智能体根据策略 $\pi_t:\mathcal{S}\times\mathcal{A}\to[0,1]$ 和当前状态 s_t 选择并执行动作 a_t, 环境接收到动作后以 $T(s_t,a_t,s_{t+1})=P(s_{t+1}\mid s_t,a_t)$ 的概率转移到下一个状态 s_{t+1}, 智能体获得即时奖励 $r_t=R(s_t,a_t,s_{t+1})$. 智能体的目标是找到一个最大化累计奖励的最优策略:

$$\pi^*=\arg\max_{\pi}\mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty}\gamma^t\cdot r_t\mid\pi\Big].$$

为了评估和学习策略, 通常定义状态值函数和状态动作值函数, 其中策略 π 的状态值函数定义为:

$$V^{\pi}(s)=\mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty}\gamma^t\cdot r_t\mid\pi,s_0=s\Big].$$

状态动作值函数定义为:

$$Q^{\pi}(s,a)=\mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty}\gamma^t\cdot r_t\mid\pi,s_0=s,a_0=a\Big].$$

为方便表述, 本文分别用 $V(s)$ 和 $Q(s,a)$ 代替 $V^{\pi}(s)$ 和 $Q^{\pi}(s,a)$. 最优状态值函数 $V^*(s)=\max_{\pi}V^{\pi}(s)$, 最优状态动作值函数 $Q^*(s,a)=\max_{\pi}Q^{\pi}(s,a)$, 最优策略可以通过学习上述两个最优函数得到.

2.2 强化学习中技能与技能发现

强化学习中的技能发现问题可以形式化成基于抉择 (option) 框架[17]的学习问题, 该框架将上述 MDP 定义中的原子动作 a 扩展到由原子动作序列构成的抉择 o. 在传统的抉择学习框架中, 研究者大多关注奖励较为稠密的场景, 在稀疏奖励的任务中, 该框架训练层次化的抉择策略较为困难. 为了利于在稀疏奖励的环境中学习抉择, 即技能, 我们引入潜在参数 (latent parameter) $z\in\mathcal{Z}$ 来表示强化学习智能体在环境中的技能. 直觉上, 同一技能产生的轨迹较为类似, 不同技能产生的轨迹差异较大, 故我们希望在智能体的策略学习中, 引入技能潜在参数 z 能够丰富策略轨迹的多样性. 与现有的技能发现研究类似[19−23], 我们将 z 嵌入到智能体的策略学习中, 得到技能策略 $\pi(a\mid s,z)$. z 同时也是一种强化学习策略轨迹的一种表征, 我们希望不同的表征可以表征不同的轨迹, 并且相似的轨迹能够拥有类似的表征. 基于此, 下面我们将定义在一个 MDP 中, 技能发现的学习目标.

定义 2. 技能发现的学习目标. 在一个马尔可夫决策过程 $\mathcal{M}$ 中, 给定一个策略 $\pi(a\mid s)$, 该策略下产生的一条轨迹定义为 $\tau=(s_0,a_0,r_1,s_1,\ldots,s_H)$, 该轨迹服从的分布为:

$$\tau\sim p(\tau)=P(s_0)\prod\nolimits_{t=0}^{H-1}\pi(a_t\mid s_t)P(s_{t+1}\mid s_t,a_t).$$

在技能发现框架下, 我们将技能发现问题定义为学习一个潜在的有条件的技能策略 $\pi(a\mid s,z)$, 其中, $z\in\mathcal{Z}$ 代表技能潜在参数, 是一个 d 维的向量, 每一维是一个连续值. 技能学习的目标能够学习一类任务下的轨迹表征分布 $q(z)$, 从而最大化在不同表征下产生的轨迹分布的信息熵, 形式化为:

$$q^*(z)=\arg\max\nolimits_{z\sim q(z)}h\left(p(\tau\mid\pi(a\mid s,z))\right),$$

其中, $h(\cdot)$ 表示对一个随机变量的随机程度的度量, 具体而言:

$$h(x)=\mathbb{E}_{x\sim p}[-\log(x)].$$

2.3 稀疏奖励下的多目标导向强化学习

第 1 节提到过, 传统的基于抉择框架的分层强化学习方法在处理奖励信号较为稀疏的任务中取得的效果较差. 当前基于技能发现的方法虽然可以在无奖励情况下学习策略, 当面对稀疏奖励任务、一个回合中存在序贯技能组合、不同回合之间技能组合存在差异时, 往往不能训练出有效的策略[43]. 本文研究一种稀疏奖励下的多目标导向强化学习 (multi goal-oriented RL with sparse rewards) 问题. 下面, 我们将形式化定义该问题, 并且约定训练任务和测试任务的具体差异.

定义 3. 稀疏奖励下的多目标导向强化学习. 基于定义 1, 该问题可以扩展成为一个七元组 $\langle\mathcal{S},\mathcal{A},R_g,T,\mathcal{G},\phi,\gamma\rangle$, 其中, $\mathcal{S}$、$\mathcal{A}$、T 和 γ 与定义 1 中的一致, $\mathcal{G}$ 是目标空间, ϕ 一个映射函数, 将状态 s 从状态空间 $\mathcal{S}$ 映射至目标空间 $\mathcal{G}$ 内一个向量的函数, R_g 是奖励函数, 与目标 $g\in\mathcal{G}$ 相关, 该奖励函数是稀疏的, 只有在智能体达到目标时, 环境才会给出奖励信号, 即:

$$R_g(s_t,a_t,s_{t+1})=\begin{cases} r, & f(s_{t+1},g)\in[0,\delta] \\ 0, & \text{otherwise} \end{cases},$$

其中, δ 是一个比较小的正实数.

在上述定义中, 若 g 是需接近类型目标, 则可令 $r=1$, 若 g 是需躲避类型目标, 则可令 $r=-1$, 二者 $f(s_{t+1},g)=\|\phi(s_{t+1})-g\|_2$; 智能体在其他状态采取动作, 不会获得奖励, 不失一般性, 这里用 0 表示, 并且值函数初始亦置为 0.

在稀疏奖励下的目标导向强化学习中, 由于目标的类型不同, 且可能存在多个目标, 则不同的目标类型会导致不同的任务类型. 我们定义 $\mathcal{T}_i$ 为某种任务类型, 任务类型 $\mathcal{T}_i$ 中不同目标类型构成了 $\{\mathcal{T}_{i1},\mathcal{T}_{i2},\ldots,\mathcal{T}_{im}\}$ 子任务集合, 所有的任务类型构成了 $\{\mathcal{T}_1,\mathcal{T}_2,\ldots,\mathcal{T}_I\}$ 任务类型集合. 在一个回合开始前, 智能体会被随机分配至某个初始状态 $s_0\in\mathcal{S}$, 环境根据从任务类型集合中采样的 $\mathcal{T}_i$ 产生 L 个目标 $\{g_1,g_2,\ldots,g_L\}\in\mathcal{G}$, 在 $\mathcal{T}_i$ 中不同的回合, L 个目标对应的子任务相同, 但每个目标的参数可能不同 (例如, 目标的坐标位置不同), 智能体需要依次完成这些目标才能完成整个任务. 智能体训练时的任务类型和测试时的任务类型可能不同.

由于每个回合产生的目标不同, 故基于技能框架的稀疏奖励下的多目标导向强化学习的目标是学习带有嵌入技能表征的策略 $\pi(a\mid s,\vec{g},\vec{z})$, 使以下目标函数最大化:

$$\mathbb{E}_{(s_0,\vec{g})\sim\nu_0}[V^{\pi}(s_0,\vec{g},\vec{z})],$$

其中, $\vec{g}=(g_1,g_2,\ldots,g_L)$, 是 L 个目标参数向量的拼接, $\vec{z}=(z_1,z_2,\ldots,z_N)$ 是 N 个技能潜在参数向量的拼接, 这里的 $N\leqslant L$, 因为一个技能可能包含多个目标, ν_0 是回合开始时环境产生初始状态 s_0 与目标 $\vec{g}$ 的联合分布.

3 基于分组对比学习的序贯感知技能发现方法 GCSSD

本节我们将详细介绍 GCSSD 中如何将分组对比学习融合至强化学习的策略网络中学习到具有不同技能的嵌入表征, 以及怎样利用学习到的技能嵌入表征在具有不同序贯技能组合的测试任务中进行快速适应. GCSSD 方法分为基于分组对比学习的技能嵌入表征、基于技能表征的强化学习训练和测试任务中的技能序贯组合 3 个部分. 下面, 将逐一介绍算法每个部分的流程.

3.1 基于分组对比学习的技能嵌入表征

在稀疏奖励下的目标导向强化学习中, 智能体需要在某任务类型 $\mathcal{T}_i$ 中依次执行相应的技能策略. 为了判断不同任务类型下应该执行哪些技能以及技能的执行顺序, 我们需要用采样策略获得轨迹来评估任务类型. 先对不同任务类型下的采样轨迹进行分段, 再根据技能分组方法对分段轨迹进行分组, 每一组相似的轨迹对应一种技能, 对不同技能的分组轨迹进行对比学习, 来获得技能的嵌入表征. 因此, 我们将学习技能嵌入表征分为采样和对比学习两个阶段. 下面将分别介绍这两个阶段的实现.

首先是采样阶段. 从任务类型集合中采样一个训练任务类型 $\mathcal{T}_i\in\{\mathcal{T}_1,\mathcal{T}_2,\ldots,\mathcal{T}_I\}$, 设定初始随机采样策略

$\pi_{\text{ran}}(a \mid s, \vec{g})$. 根据采样策略在环境中采样 H 步, 获取智能体的运动轨迹 τ_i 和奖励轨迹 ρ_i. 运动轨迹 τ_i 与定义 2 中的一致, 即:

$$\tau_i = (s_0, a_0, r_1, s_1, \ldots, s_H),$$

奖励轨迹 ρ_i 由 τ_i 中的即时奖励序列构成, 即:

$$\rho_i = (r_1, r_2, \ldots, r_H).$$

利用奖励轨迹可以得到关于任务类型 $\mathcal{T}_i$ 的信息. 例如, 奖励值低的地方可能存在需要躲避的目标, 奖励值高的地方可能存在需要接近的目标. 那么, 在躲避目标前, 智能体需完成躲避的技能, 相应地, 在接近目标前, 需完成靠近的技能. 由于每个任务类型可能由多个技能组成, 我们将运动轨迹 τ_i 和奖励轨迹 ρ_i 均分为 N 段分段轨迹, 即:

$$\left(\tau_i^1, \tau_i^2, \ldots, \tau_i^N\right) \text{与} \left(\rho_i^1, \rho_i^2, \ldots, \rho_i^N\right),$$

其中, 每段轨迹长度为 h, 即:

$$\tau_i^n = (s_{(n-1)\cdot h}, a_{(n-1)\cdot h+1}, r_{(n-1)\cdot h+1}, s_{(n-1)\cdot h+1}, \ldots, s_{n\cdot h}),$$

$$\rho_i^n = (r_{(n-1)\cdot h+1}, r_{(n-1)\cdot h+2}, \ldots, r_{n\cdot h}).$$

这里运动轨迹和奖励轨迹的最后一段可能不足 h 步, 以实际所剩步数为准. 若 ρ_i^n 中存在非零奖励, 则其中可能存在需要某种技能完成的目标. 我们令

$$R_i^n = \sum\nolimits_{d=(n-1)\cdot h+1}^{n\cdot h} r_d,\ r_d \in \rho_i^n,$$

其中, R_i^n 的大小反应了某些目标存在这段轨迹里的可能. 例如, 如果 $R_i^n < 0$, 则该段轨迹之前需执行躲避技能, 反之则需执行靠近的技能. 在 GCSSD 中, 我们对训练任务类型 $\mathcal{T}_i$ 进行多次轨迹采样, 获得 N 段分段轨迹奖励均值, 即:

$$\left(\bar{R}_i^1, \bar{R}_i^2, \ldots, \bar{R}_i^N\right),\ \bar{R}_i^n = \mathbb{E}_{\tau_i \sim \mathcal{T}_i} R_i^n.$$

同样地, 将该采样方法运用到其他任务类型, 得到其他任务类型的 N 段分段轨迹奖励均值, 多次采样计算不同段奖励均值的相似度, 即:

$$w_{i_1,i_2}^{n_1,n_2} = \frac{\bar{\rho}_{i_1}^{n_1\top} \bar{\rho}_{i_2}^{n_2}}{\left\|\bar{\rho}_{i_1}^{n_1}\right\| \left\|\bar{\rho}_{i_2}^{n_2}\right\|},$$

其中, $\bar{\rho}_i^n$ 表示轨迹 τ_i 第 n 段的多次采样后每个维度上奖励的均值, 将相似度接近的分段轨迹分为一组, 即可得到 K 组相似轨迹, 本文认为一组轨迹是通过一种技能策略产生.

下面, 介绍对比学习阶段. 在采样阶段得到多个训练任务类型产生的 K 组轨迹后, 智能体需学习各技能对应的技能表征. 按任务的技能执行顺序将 N 个技能表征拼接, 从而得到任务嵌入表征. 然后将该表征嵌入策略网络中, 指导智能体按序执行不同的技能并做出相应动作.

本文设定初始技能参数及对应策略网络为 $\pi_\theta(a \mid s, \vec{g}, \vec{z}_0)$, 其中 $\vec{z}_0$ 初始值均为 0, 维度为 $N \cdot \dim(z_i)$. 对训练任务类型集合, 经过采样阶段获得了 K 组轨迹, 每组轨迹对应一种技能. 将第 k 组轨迹存入相应的技能轨迹回放缓存 (replay buffer) $\mathcal{B}_k$ 中, 构成总体回放缓存 $\mathcal{B}$, 即:

$$\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_K\}.$$

为便于描述, 以 $N = 2$, 即将采样轨迹均分为 2 段为例. 如图 2 所示, $\mathcal{T}_1$ 的分段轨迹 (τ_1^1, τ_1^2) 分别对应技能 1 与技能 2, $\mathcal{T}_2$ 的分段轨迹 (τ_2^1, τ_2^2) 分别对应技能 2 与技能 1, 因此将 τ_1^1 和 τ_2^2 作为一组分组轨迹存入 $\mathcal{B}_1$, 将 τ_1^2 和 τ_2^1 作为另一组分组轨迹存入 $\mathcal{B}_2$, 便于后续利用这些分组轨迹进行对比学习.

用技能表征网络 $e_\eta(\cdot)$ 编码分组轨迹, 得到相应的技能表征, 分组轨迹对应的技能表征可以代表该段轨迹的信息, 并且这些技能表征具有一致性和区分性. 首先, 同一个技能轨迹回放缓存中的分组轨迹通过编码得到的技能表征应尽可能相似. 其次, 不同的技能轨迹回放缓存中的分组轨迹通过编码得到的技能表征应尽可能不同. 通过对分组轨迹编码并进行对比学习, 来得到这样的表征网络. 对第 k 种技能, 从 $\mathcal{B}_k$ 中取 M 条分组轨迹, 用技能表征网络 $e_\eta(\cdot)$ 对这 M 条分组轨迹进行编码得到 M 个技能表征向量 $(z_{k,1}, z_{k,2}, \ldots, z_{k,M})$, 求其每个维度平均值作为该技能表征

的中心 $\bar{z}_k$. 对所有技能重复上述过程, 则最终得到 K 个技能表征中心 $(\bar{z}_1,\bar{z}_2,\ldots,\bar{z}_K)$.

用采样阶段的 K 组相似轨迹对应的奖励轨迹均值 $\bar{\rho}_k$, 计算第 k 、 p 种技能表征的相似度 $w_{k,p}$, 来区分技能表征中心 $\bar{z}_k$ 的正负样本. 将与第 k 种技能表征相似度高的技能表征对应的分组轨迹视作正样本, 使它们与 $\bar{z}_k$ 相似; 相似度低的技能表征对应的分组轨迹视作负样本使它们与 $\bar{z}_k$ 不同. 另外, 在计算损失时以它们的相似度作为权重. 我们借鉴了文献 [31] 的损失函数, 并对其做出了改进, 使其能够满足上述要求. 损失函数 $\mathcal{L}_\eta$ 如下:

$$\begin{cases} \mathcal{L}_\eta = \dfrac{1}{K}\sum_{k=1}^{K} l_\eta(k) \\ l_\eta(k) = \sum_{p=1}^{K} l_\eta(k,p) \\ l_\eta(k,p) = -w_{k,p}\dfrac{1}{M}\sum_{m=1}^{M}\log\dfrac{\mathrm{e}^{c_{k,p}^m/\kappa}}{\sum_{u=1}^{K}\sum_{m=1}^{M}\mathrm{e}^{c_{k,p}^m/\kappa}} \end{cases} \tag{1}$$

其中, $c_{k,\mathrm{p}}^m = \dfrac{\bar{z}_k^{\top} z_{p,m}}{\|\bar{z}_k\|\|z_{p,m}\|}$, $\bar{z}_k = \sum_{m=1}^{M} z_{k,m}$, $z_{k,m} = e_\eta(\tau_{k,m}), w_{k,p} = \dfrac{\bar{\rho}_k^{\top}\bar{\rho}_p}{\|\bar{\rho}_k\|\|\bar{\rho}_p\|}$, $\bar{\rho}_k = \dfrac{1}{M}\sum_{m=1}^{M}\rho_k^m$. 其中, $\bar{z}_k$ 表示第 k 种技能表征的中心, $\tau_{k,m}$ 表示从第 k 种技能轨迹回放缓存 $\mathcal{B}_k$ 中取出的第 m 条分组轨迹, ρ_k^m 表示上述 m 条分组轨迹对应的轨迹奖励. 本文认为该优化目标 (1) 是定义 2 中技能发现目标的一种直接近似.

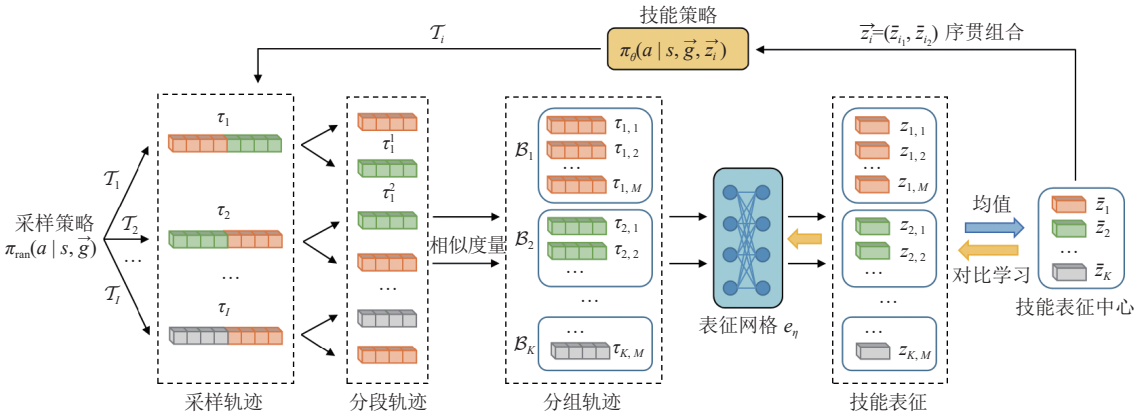

图 2 GCSSD 中技能嵌入表征与强化学习策略联合训练框架

3.2 基于技能表征的强化学习训练

强化学习的目标是学习策略来最大化期望回报, 最优策略的学习可以通过基于值函数的算法、基于策略梯度的算法和演员-评论家 (actor-critic, AC) 算法实现. 其中, 基于值函数的算法通过最小化时序差分 (temporal difference, TD) 误差 δ 更新策略网络; 基于策略梯度的强化学习方法将 $\pi_\theta(a \mid s)$ 视作一个关于 θ 的连续可微函数, 通过梯度上升的方法优化参数 θ 来最大化目标函数, 即:

$$\mathcal{L}_\theta = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H}\left(\sum_{t'=t}^{H}\gamma^{t'-t}r_{t'}\right)\log\pi_\theta(a_t \mid s_t)\right].$$

文本基于 AC 算法实现序贯感知技能发现, 在任务类型 $\mathcal{T}_i$ 中技能嵌入表征为 $\vec{z}_i = (z_{i_1}, z_{i_2}, \ldots, z_{i_N})$, 通过价值函数网络 $V_\varphi(s_t, \vec{g}, \vec{z}_i)$ 最小化 TD 误差, 根据策略梯度优化策略函数 $\pi_\theta(a_t \mid s_t, \vec{g}, \vec{z}_i)$. 在 GCSSD 中, 可以结合在策略 (on-policy) 和离策略 (off-policy) 的强化学习算法与嵌入表征进行联合学习. 为了方便描述, GCSSD 中结合嵌入表征 $\vec{z}_i$ 的在策略算法 PPO, 给出策略优化目标为:

$$\mathcal{L}_\theta = -\mathbb{E}_{\pi_\theta}\left[\min\left(l_t(\theta)A_t, \mathrm{clip}\left(l_t(\theta), 1-\varepsilon, 1+\varepsilon\right)A_t\right)\right] \tag{2}$$

其中, $l_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t, \vec{g}, \vec{z}_i)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t, \vec{g}, \vec{z}_i)}$, $A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \ldots + (\gamma\lambda)^{H-t+1}\delta_{H-1}$, $\delta_t = r_t + \gamma V_\varphi(s_{t+1}, \vec{g}, \vec{z}_i) - V_\varphi(s_t, \vec{g}, \vec{z}_i)$, $\lambda \in [0,1]$, 下

面我们将介绍如何对表征网络 e_η 和策略网络 π_θ 进行交替训练.

在采样阶段, 首先采样一个训练任务 $\mathcal{T}_i$, 根据初始随机策略 $\pi_{\text{ran}}(a \mid s, \vec{g})$ 在环境中采样, 获取智能体的运动轨迹 τ_i 和奖励轨迹 ρ_i, 通过计算得到任务技能及其执行顺序. 其中, 每次以采样得到的采样轨迹 τ_i 作为数据, 对策略网络 π_θ 进行优化.

在对比学习阶段, 我们将协同训练表征网络和策略网络. 每隔固定数量回合, 从分组轨迹缓冲区中取得最近的分组轨迹数据对表征网络 e_η 进行优化, 在此期间保持策略网络不变. 更新一次表征网络后, 用其对分组轨迹进行编码, 则得到新的技能表征. 将新的技能表征按任务的技能执行顺序拼接得到 $\vec{z}_i$, 如图 2 所示, 对于训练任务 $\mathcal{T}_1$, $\vec{z}_1 = (z_1, z_2)$; 对于训练任务 $\mathcal{T}_N$, $\vec{z}_N = (z_K, z_1)$. 将 $\vec{z}_i$ 嵌入策略网络 $\pi_\theta(a_t \mid s_t, \vec{g}, \vec{z}_i)$, 并且用此策略网络得到最新的轨迹. 每个回合, 以最新的轨迹作为数据, 再对策略网络 π_θ 进行优化, 并在此期间保持表征网络不变. 重复上述交替过程, 直至表征网络和策略网络收敛, 完成训练.

3.3 测试任务中的技能序贯组合

在这个阶段, 智能体遇到的任务类型可能和训练时的任务类型不同. 在 GCSSD 中, 在测试任务中, 不再对表征网络和策略网络进行额外的学习, 而是直接使用已经训练好的技能表征网络 e_η 对采样的分段轨迹依次编码, 得到相应的技能表征, 将其作为嵌入表征输入到训练好的策略网络 π_θ 中, 指导智能体针对新任务执行相应的技能. 虽然, 整体而言, 测试任务与训练任务不同, 但其各个子部分仍对应原有的技能, 因此仍可以利用训练后的表征网络对分段轨迹进行编码, 得到对应的技能表征. 虽然策略网络没有进行迁移训练, 但由于技能表征部分在之前训练中出现过, 因此在测试时仍具有一定的适应能力. 反之, 如果不进行轨迹分段, 而直接使用训练好的表征网络对采样轨迹进行编码, 则会得到一个全新的任务表征, 因为从整体而言, 数据发生了改变. 将该全新的任务表征嵌入之前训练好的策略网络则不再有适应的效果. 下面将介绍测试阶段的具体过程.

测试时采样任务 $\mathcal{T}_j \in \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_J\}$, 采样成功轨迹 τ_j 并将其均分为 N 段轨迹 $\tau_{j,1}, \tau_{j,2}, \ldots, \tau_{j,N}$, 存入测试轨迹回放缓存 $\mathcal{B}_{\text{test}} = \{\{\mathcal{B}_{1,1}, \ldots, \mathcal{B}_{1,N}\}, \ldots, \{\mathcal{B}_{J,1}, \ldots, \mathcal{B}_{J,N}\}\}$ 的 $\{\mathcal{B}_{j,1}, \ldots, \mathcal{B}_{j,N}\}$ 中. 如图 3 所示, 将轨迹 τ_j 分为 $\tau_{j,1}, \tau_{j,2}$ 并分别存入 $\{\mathcal{B}_{j,1}, \mathcal{B}_{j,2}\}$. 多次采样任务及其成功轨迹, 每次按任务 j 将划分的 N 段轨迹存入对应的测试轨迹回放缓存中, 直到各个缓存内的轨迹数量多于 M 条.

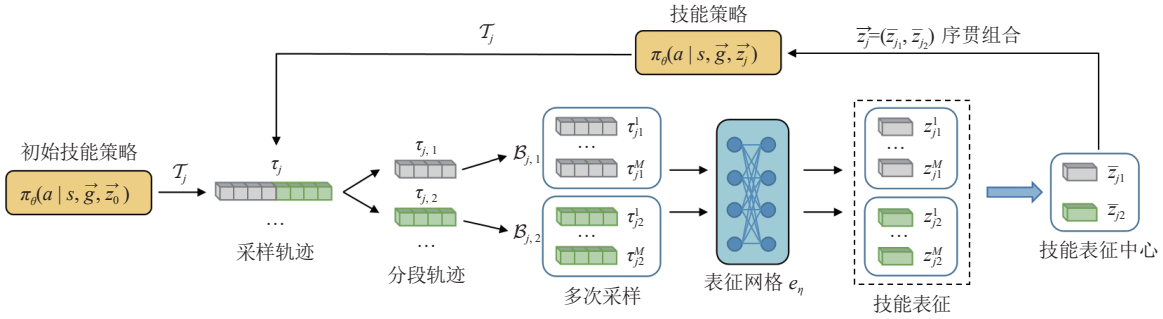

图 3 GCSSD 中预训练技能嵌入表征与强化学习策略测试框架

当再次面对测试任务 $\mathcal{T}_j$ 时, 从对应的分段轨迹测试轨迹回放缓存 $\{\mathcal{B}_{j,1}, \ldots, \mathcal{B}_{j,N}\}$ 中各取出 M 条分段轨迹 $\{(\tau_{j,1}^1, \ldots, \tau_{j,1}^M), \ldots, (\tau_{j,N}^1, \ldots, \tau_{j,N}^M)\}$, 用表征网络对这 M 条分段轨迹进行编码并取均值作为技能表征, 即:

$$\bar{z}_{j,1} = \frac{1}{M} \sum_{m=1}^{M} e_\eta\left(\tau_{j,1}^m\right) \tag{3}$$

最终得到任务 $\mathcal{T}_j$ 的 N 个技能表征 $(z_{j,1}, \ldots, z_{j,N})$, 将其拼接得到 $\vec{z}_j = (z_{j,1}, \ldots, z_{j,N})$, 作为技能表征嵌入策略网络中指导智能体执行相应的技能, 测试算法框架如图 3 所示.

在 GCSSD 中, 策略训练如算法 1 所示.

算法 1. GCSSD 算法 (训练).

输入: 训练任务分布集合 $\mathcal{T}_{\text{train}}=\{\mathcal{T}_1,\ldots,\mathcal{T}_I\}$, 初始化技能表征网络参数 η、策略网络参数 θ, 训练轨迹回放缓存 $\mathcal{B}_{\text{train}}$, 分组技能轨迹回放缓存 $\mathcal{B}=\{\mathcal{B}_1,\mathcal{B}_2,\ldots,\mathcal{B}_K\}$;
输出: 技能表征网络参数 η、策略网络参数 θ.

1. 对各任务, 根据 $\pi_{\text{ran}}(a\mid s,\vec{g})$ 采样得到运动轨迹 τ_i 和奖励轨迹 ρ_i, 通过计算得到 K 种技能, 初始化 $z_{\text{train}}=\{z_1,\ldots,z_K\}$, 以及每个任务类型 $\mathcal{T}_i$ 对应的技能执行顺序.
2. For $e=0,\ldots,E$ do
3. 采样任务 $\mathcal{T}_i$, 根据 $\mathcal{T}_i$ 技能顺序 $i_1,\ldots,i_N$, 用 $\pi_\theta(a\mid s,\vec{g},\vec{z}_i)$ 采样轨迹 τ 存入 $\mathcal{B}_{\text{train}}$, 其中 $\vec{z}_i=(z_{i_1},z_{i_2},\ldots,z_{i_N})$, $i_1,\ldots,i_N\in\{1,\ldots,K\}$.
4. if τ_i 是成功轨迹, 则将 τ_i 分为 $(\tau_i^1,\tau_i^2,\ldots,\tau_i^N)$, 分别存入 $\mathcal{B}$ 中的 $\{\mathcal{B}_{i_1},\mathcal{B}_{i_2},\ldots,\mathcal{B}_{i_N}\}$, $i_1,\ldots,i_N\in\{1,\ldots K\}$.
5. 从 $\mathcal{B}_{\text{train}}$ 中取轨迹数据, 用公式 (2) 计算 $\mathcal{L}_\theta$ 并更新 π_θ.
6. if e % Freq = 0 and $\forall k\in 1,\ldots,K,|\mathcal{B}_k|\geqslant M$
7. 从 $\mathcal{B}$ 的 K 个缓存中各取 M 条分组轨迹数据, 用公式 (1) 计算 $\mathcal{L}_\eta$ 并更新 e_η.
8. 更新 $z_{\text{train}}=\{z_1,z_2,\ldots,z_K\}$, 其中 $z_k=\bar{z}_k$, $k=1,\ldots,K$.
9. End

在 GCSSD 中, 策略测试如的算法 2 所示.

算法 2. GCSSD 算法 (测试).

输入: 训练任务分布集合 $\mathcal{T}_{\text{test}}=\{\mathcal{T}_1,\ldots,\mathcal{T}_J\}$, 训练完成的技能表征网络参数 η、策略网络参数 θ, 测试回放缓存 $\mathcal{B}_{\text{test}}=\{\{\mathcal{B}_{1,1},\ldots,\mathcal{B}_{1,N}\},\ldots,\{\mathcal{B}_{J,1},\ldots,\mathcal{B}_{J,N}\}\}$;
输出: 测试结果.

1. For $e=0,\ldots,E$ do
2. 采样任务 $\mathcal{T}_j$, 用 $\pi_\theta(a\mid s,\vec{g},\vec{z}_j)$ 采样 τ_j.
3. if τ_j 是成功轨迹, 则将 τ_j 分为 $(\tau_j^1,\ldots,\tau_j^N)$, 分别存入 $\mathcal{B}_{\text{test}}$ 中的 $(\mathcal{B}_{j,1},\ldots,\mathcal{B}_{j,N})$.
4. if e % Freq = 0 and $\forall n\in 1,\ldots,N,|\mathcal{B}_{j,n}|\geqslant M$
5. 从 $\{\mathcal{B}_{j,1},\ldots,\mathcal{B}_{j,N}\}$ 中各取 M 条分段轨迹, 用公式 (3) 计算并更新技能表征 $\vec{z}_j$.
6. End

4 实验及结果分析

在本节中, 我们通过两个不同场景下的仿真实验来验证所提出的 GCSSD 方法的可行性, 并与其他相关的强化学习方法进行比较. 首先, 我们给出了实验的仿真场景, 并介绍其相关设置; 随后, 我们介绍了用于对比的基线方法, 以及 GCSSD 的参数设定; 最后, 结合实验结果和仿真特性分析 GCSSD 的优势.

4.1 实验平台

在本文中, 我们使用一个离散动作空间任务网格世界和一个连续动作空间任务质点控制测试所提出的 GCSSD 算法. 在离散任务中, 我们用在策略的强化学习算法作为基线进行比较, 在连续任务中, 我们采用离策略的强化学习算法作为基线进行比较. 下面将会对两个实验平台进行阐述.

4.1.1 网格世界

本文设计了一种面向稀疏奖励下的多目标导向强化学习的网格世界, 在这个任务中, 智能体在每个状态可选“上、下、左、右”这 4 个动作, 如图 4 所示.

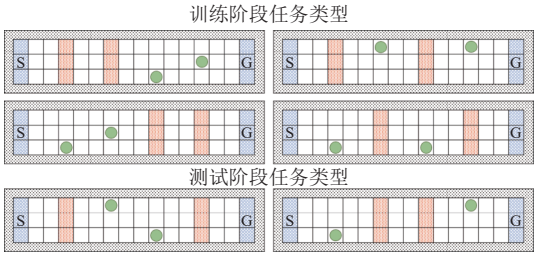

图 4 网格世界环境示意图

智能体需从左边的起点出发, 每个回合起始点从左侧浅蓝色 3 个状态中随机产生. 碰到 2 个从浅红色区域随机产生的目标, 并且躲开途中的 2 个位置固定的绿色目标, 最终到达回合开始前从最右边蓝色区域随机产生的目标点. 图中灰色网格为边界墙, 如果走到灰色网格将退回原来的网格. 训练环境具有 4 种任务类型, 每种任务类型有 3^4 种具体任务; 测试环境具有两种任务类型, 每种任务分布同样有 3^4 种具体任务. 智能体每走一步奖励值为−1, 碰到一个红色目标奖励值为+10, 碰到一个绿色目标奖励值为−10, 碰到终点奖励值为+10. 每个回合的最大步长为 100, 若智能体到达两个红色目标, 并且躲开两个绿色目标, 最终到达终点, 则认为成功完成该任务.

4.1.2 质点控制

本文设计了一种面向稀疏奖励下的多目标导向强化学习的质点控制任务, 我们基于 DeepMind 公司的 MuJoCo 虚拟引擎[44]和 OpenAI 公司的 Gym 的强化学习的标准 API[45], 设计了一个连续动作环境, 智能体在 2 维连续空间内执行动作, 如图 5 所示.

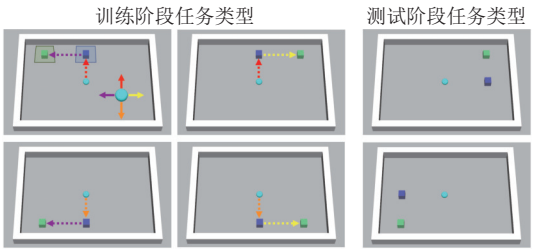

图 5 质点控制环境示意图

智能体需从中间的起点出发, 依次找到蓝色的目标 g_1 、绿色目标 g_2 . 训练环境具有 6 种任务类型, 图 5 左边列出了 4 种. 测试环境具有 2 种任务类型, 如图 5 右边所示. 每个回合, 各个任务的起点、 g_1 、 g_2 从图中正方形的浅色区域内生成. 智能体每走一步奖励值为−1, 按次序碰到 g_1 和 g_2 奖励值为 0. 在该类型的任务中, 我们期望智能体学到向上/下/左/右寻找目标点 4 个技能, 分别在图 5 中用不同颜色箭头表示, 在测试任务中, 是训练任务没有的任务类型.

4.2 基线方法与参数设置

4.2.1 基线方法

本文所提出的 GCSSD 方法主要针对多目标的强化学习任务, 并且可以结合在策略和离策略的强化学习算法, 处理离散和连续动作空间的任务. 因此, 采用在策略的 PPO 方法作为离散任务中的基线方法, 采用离策略的 SAC 方法作为连续任务中的基线方法. 并结合深度循环网络对其处理多任务能力进行增强, 此外, 在两种任务中引入 OC 算法, 作为目前技能学习的基线方法. 因此, 通过上述该方法与 GCSSD 进行比较, 能够展示其技能学习优势.

● PPO 是一种基于策略梯度的强化学习算法[15], 其核心是结合了策略梯度方法和剪切 (clipping) 技术, 从而能够有效地避免过度更新策略, 提高算法的稳定性和可靠性.

● RPPO 使用循环神经网络 (RNN) 来表示策略函数[46], 通过将历史状态和动作信息作为输入, 使得策略函数能够更好地处理长期依赖性, 是面向多任务环境改进 PPO 算法的常用方法.

● SAC 是一种高效、稳定且能够学习最大熵策略的强化学习算法[16]. 其核心思想是学习一个最大熵策略, 该策略在执行动作时尽可能地探索状态空间, 从而最大化总体回报.

● OC 是一种经典的基于抉择框架的技能强化学习算法[18]. 在 OC 中, 智能体可以选择执行一组原子动作的序列. 每个抉择由一个策略和一个价值函数组成. 在执行抉择时, 智能体可以在一段时间内执行该抉择中定义的原子动作序列, 以实现一定的目标.

● ROC 是对 OC 方法的一种改进, 与 RPPO 算法一样, 使用循环神经网络 (RNN) 来表示策略函数[46], 从而更好地处理长期依赖性, 是提升 OC 方法处理多任务环境的一种常用方式.

4.2.2 参数设置

● 在网格世界中, 将轨迹分为两段, 则训练阶段任务类型 $\mathcal{T}_1$ 的前半段需靠近红色目标两次, 后半段需躲避绿色目标两次; 训练阶段任务类型 $\mathcal{T}_2$ 的前半段需躲避绿色目标两次, 后半段需靠近红色目标两次. 则该两种任务类型中存在两种技能, 即靠近红色目标两次、躲避绿色目标两次. 同理, 任务类型 $\mathcal{T}_3$ 、$\mathcal{T}_4$ 也存在两种技能, 即先躲避绿色目标再靠近红色目标、先靠近红色目标再躲避绿色目标. 则训练任务的技能集合共 4 种. 另外, 测试阶段所用到的技能是训练阶段技能的子集.

在本实验中, 状态 s 为智能体当前所处网格中位置的横坐标与纵坐标 (x,y), 目标向量为 $\vec{g}=(g_1,g_2,g_3,g_4,g_f)$, $g_i=(x_i,y_i)$, $i=1,2,3,4$ 和 f 分别表示 4 个目标及终点的横、纵坐标. 技能表征 $\vec{z}_i=(z_{i_1},z_{i_2})$, 其中 z_{i_1}、z_{i_2} 均设为 2 维. 动作 a 是使智能体向上/下/左/右移动一格. 技能策略 π_θ 采用 LSTM 网络, 输入维度为 16 (忽略 LSTM 网络中隐表示的维度), 输出维度为 4. 技能表征网络 e_η 采用全连接网络, 则输入维度为 40, 输出维度为 2. 实验所用参数如表 1 所示.

表 1 网格世界中模型的超参数

参数	值	参数	值	参数	值
回合最大步长	100	A_t 参数 λ	0.95	π_θ 隐层维度	64
训练回合数量	60 000	Clipping 截断率	0.1	π_θ 网络层数	2
批处理数据量	64	Adam 学习率	5×10^{-4}	e_η 隐层维度	64
折扣因子 γ	0.99	梯度下降更新次数	2	e_η 网络层数	2

● 在质点控制中, 将轨迹分为两段, 则训练阶段任务类型 $\mathcal{T}_1$ 的前半段需向上走找到 g_1, 后半段需向左走找到 g_2; 训练阶段任务类型 $\mathcal{T}_2$ 的前半段需向上走找到 g_1, 后半段需向右走找到 g_2. 其他任务也同理, 各需要 2 种技能组合. 则训练任务的技能集合为 4 种, 即向上/下/左/右寻找目标点. 同样, 测试阶段所用到的技能是训练阶段技能的子集.

实验中状态 s 为智能体所处网格的横、纵坐标以及横、纵方向速度 (x,y,v_x,v_y), 目标向量 $\vec{g}=(g_1,g_2)$, $g_i=(x_i,y_i)$, $i=1,2$ 表示 2 个目标的横、纵坐标. 技能表征 $\vec{z}_i=(z_{i_1},z_{i_2})$, 其中 z_{i_1}、z_{i_2} 均设为 2 维. 动作 a 是一个二维向量表示横、纵坐标的增减量, 每一维的范围在 $[-1,1]$. 技能策略 π_θ 采用全连接网络, 输入维度为 12, 输出维度为 2. 技能表征网络 e_η 采用全连接网络, 则输入维度为 40, 输出维度为 2. 实验所用参数如表 2 所示.

表 2 质点控制中模型的超参数

参数	值	参数	值	参数	值
回合最大步长	100	Adam 学习率	0.001	π_θ 隐层维度	256
训练总步长	1.75×10^5	梯度下降更新次数	1	π_θ 网络层数	2
批处理数据量	5 000	目标平滑系数	0.005	e_η 隐层维度	256
折扣因子 γ	0.99	目标更新间隔	1	e_η 网络层数	2
回放缓存数据量	10^6	—	—	—	—

4.3 结果分析

4.3.1 对比实验

● 在网格世界中, 首先, 为了证明分组轨迹对比学习得到的技能策略的效果, 我们将 GCSSD-2 (分段数 N =2)

与 RPPO 算法在训练和测试环境中进行了对比. 从图 6 和表 3 可以看出, GCSSD-2 在训练和测试环境上均高于基线方法. 其次, 为了证明利用分组轨迹进行技能学习对测试任务能够快速适应, 我们在训练和测试环境中对不使用分段而直接分组的 GCSSD-1 (分段数 N=1)、使用分段再分组的轨迹学习技能的 GCSSD-2 进行了比较. 从图 6 和表 3 可以看出, 两种方法在训练环境中的表现相当, 而在测试环境中, GCSSD-2 的结果明显高于 GCSSD-1. 最后, 我们还与技能发现方法 OC 进行了比较, 展示我们算法的优越性. 同样从图 6 和表 3 中可以看出, 我们的方法优于 OC 算法. 由于实验过程中, 我们发现使用 RNN 网络作为策略网络会大大提升实验效果, 因此我们也对其他方法是否使用 RNN 网络作为策略网络进行了对比.

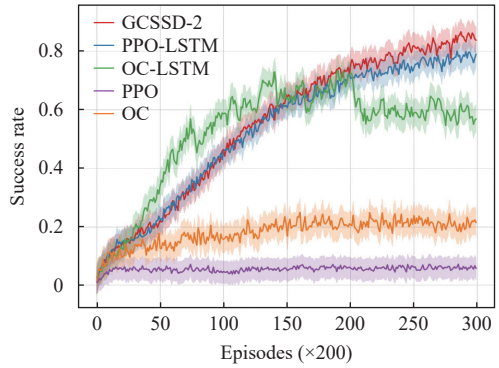

图 6 网格世界训练曲线

表 3 网格世界训练与测试成功率

算法	训练成功率	测试成功率
GCSSD-2 (N=2)	**0.840**	**0.251**
GCSSD-1 (N=1)	0.834	0.204
RPPO	0.778	0.180
PPO	0.059	0.046
ROC	0.719	0.226
OC	0.208	0.092

● 在质点控制中. 为了证明分组轨迹对比学习得到的技能策略在连续环境中也具有一定的作用, 我们将我们的方法与 SAC、PPO、OC 算法在训练和测试环境中进行了对比. 从图 7 和表 4 可以看出, 我们的方法比基线更快收敛到最优解, 并且在测试时达到更高的奖励值.

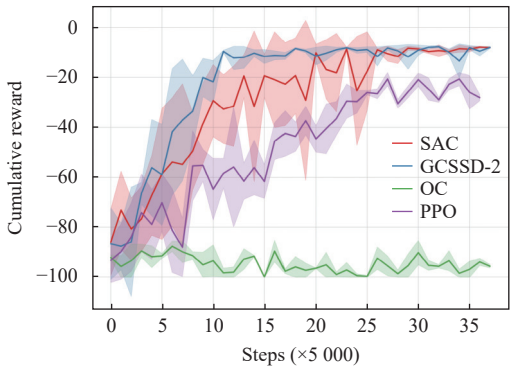

图 7 质点控制训练曲线

表 4 质点控制训练与测试奖励值

算法	训练奖励值	测试奖励值
GCSSD-2 (N=2)	**−8.519**	**−64.937**
SAC	−8.569	−73.878
PPO	−20.912	−84.456
OC	−90.562	−97.434

4.3.2 分段数量对实验的影响

在进行轨迹分段对比学习时, 可以根据实验设置来调整轨迹的分段数量, 我们在网格世界环境中将轨迹分为 2 段和 4 段并进行了对比实验. 实验结果发现, 当将轨迹分为 2 段时, 训练和测试的结果稍好于分为 4 段. 我们推测这是因为, 分为 4 段后, 子轨迹长度较短, 包含的信息量较少, 因此进行对比学习时, 区分度较低. 但即使这样, 分段为 4 时, 我们的实验结果仍然明显高于 RPPO 以及不使用分段 ($N=1$) 的方法. 实验结果如后文表 5 所示.

4.3.3 技能相似度 $w_{k,p}$ 对实验的影响

在采样阶段, 通过计算分段轨迹奖励均值的相似度 $w_{i_1,i_2}^{n_1,n_2}$ 划分 K 种技能, 重新计算第 k 、 p 种技能表征的相似度 $w_{k,p}$, 并且用该相似度来构建对比损失, 学习技能表征网络. 为了验证技能相似度 $w_{k,p}$ 对实验的影响, 我们对利

用 $w_{i_1,i_2}^{n_1,n_2}$ 对任务轨迹进行技能划分并以 $w_{k,p}$ 计算技能相似度和对任务轨迹进行随机技能划分在网格世界环境中进行了比较. 实验结果发现, 在随机技能划分下实验效果和 RPPO 方法接近, 低于我们的方法, 从而验证了我们方法的有效性. 实验结果如表 6 所示.

表 5　分段数量对结果的影响

算法	训练成功率	测试成功率
GCSSD-2 (N=2)	**0.840**	**0.251**
GCSSD-4 (N=4)	0.827	0.245
GCSSD-1 (N=1)	0.834	0.204
RPPO	0.778	0.180

表 6　技能划分对结果的影响

算法	训练成功率	测试成功率
按相似度划分技能	0.840	0.251
随机划分技能	0.795	0.179
RPPO	0.778	0.180

5 总　结

本文提出了一种基于分组对比学习的序贯感知技能发现方法 (GCSSD), 针对目前基于技能发现框架的强化学习方法存在的处理序贯技能组合能力不足的问题, 利用对比学习, 对分组后的轨迹进行嵌入表征学习. 并将表征嵌入至强化学习策略, 与其进行联合训练, 在测试任务中, GCSSD 利用已经学好的技能表征网络, 可以快速适应具有不同序贯技能组合的任务. 在实验中验证了所提算法的有效性. 未来的工作包括两方面, 其一是利用概率匹配进行轨迹分组, 其二是在更大规模的实验中验证所提算法的有效性.

References:

[1] Li JW, Monroe W, Ritter A, Jurafsky D, Galley M, Gao JF. Deep reinforcement learning for dialogue generation. In: Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics, 2016. 1192–1202. [doi: 10.18653/v1/D16-1127]

[2] Kwan WC, Wang HR, Wang HM, Wong KF. A survey on recent advances and challenges in reinforcement learning methods for task-oriented dialogue policy learning. Machine Intelligence Research, 2023, 20(3): 318–334. [doi: 10.1007/s11633-022-1347-y]

[3] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright CL, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, Schulman J, Hilton J, Kelton F, Miller L, Simens M, Askell A, Welinder P, Christiano P, Leike J, Lowe R. Training language models to follow instructions with human feedback. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 2011.

[4] Rudin N, Hoeller D, Reist P, Hutter M. Learning to walk in minutes using massively parallel deep reinforcement learning. In: Proc. of the 5th Conf. on Robot Learning. London: PMLR, 2021. 91–100.

[5] Yu C, Dong YZ, Guo X, Feng YH, Zhuo HK, Zhang Q. Structure-motivated interactive deep reinforcement learning for robotic control. Ruan Jian Xue Bao/Journal of Software, 2023, 34(4): 1749–1764 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6708.htm [doi: 10.13328/j.cnki.jos.006708]

[6] Wang JY, Huang ZQ, Yang DY, Huang XW, Zhu Y, Hua GY. Spatio-lock synchronous constraint guided safe reinforcement learning for autonomous driving. Journal of Computer Research and Development, 2021, 58(12): 2585–2603 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2021.20211023]

[7] Xuan SZ, Ke LJ. Study on attack-defense countermeasure of UAV swarms based on multi-agent reinforcement learning. Radio Engineering, 2021, 51(5): 360–366 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-3106.2021.05.004]

[8] Teh YW, Bapst V, Czarnecki WM, Quan J, Kirkpatrick J, Hadsell R, Heess N, Pascanu R. Distral: Robust multitask reinforcement learning. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 4499–4509.

[9] Hessel M, Soyer H, Espeholt L, Czarnecki W, Schmitt S, van Hasselt H. Multi-task deep reinforcement learning with popart. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 3796–3803.

[10] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1126–1135.

[11] Kumar A, Zhou A, Tucker G, Levine S. Conservative Q-learning for offline reinforcement learning. In: Proc. of the 34th Conf. on Neural Information Processing Systems. 2020. 1179–1191.

[12] Parisotto E, Ba LJ, Salakhutdinov R. Actor-mimic: Deep multitask and transfer reinforcement learning. In: Proc. of the 4th Int'l Conf. on

Learning Representations. San Juan, 2016. 1–16.

[13] Abel D, Jinnai Y, Guo SY, Konidaris GD, Littman ML. Policy and value transfer in lifelong reinforcement learning. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 20–29.

[14] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence. Phoenix: AAAI, 2016. 2094–2100. [doi: 10.1609/aaai.v30i1.10295]

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

[16] 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 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 1861–1870.

[17] 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]

[18] Bacon PL, Harb J, Precup D. The option-critic architecture. In: Proc. of the 31st AAAI Conf. on Artificial Intelligence. San Francisco: AAAI, 2017. 1726–1734. [doi: 10.1609/aaai.v31i1.10916]

[19] Eysenbach B, Gupta A, Ibarz J, Levine S. Diversity is all you need: Learning skills without a reward function. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019. 1–22.

[20] Sharma A, Gu SX, Levine S, Kumar V, Hausman K. Dynamics-aware unsupervised discovery of skills. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020. 1–21.

[21] Frans K, Ho J, Chen X, Abbeel P, Schulman J. Meta learning shared hierarchies. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018. 1–11.

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

[23] Kim J, Park S, Kim G. Unsupervised skill discovery with bottleneck option learning. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 5572–5582.

[24] Nachum O, Gu SX, Lee H, Levine S. Data-efficient hierarchical reinforcement learning. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Montréal: Curran Associates Inc., 2018. 3307–3317.

[25] Levy A, Konidaris GD, Platt Jr R, Saenko K. Learning multi-level hierarchies with hindsight. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019. 1–16.

[26] Li AC, Florensa C, Clavera I, Abbeel P. Sub-policy adaptation for hierarchical reinforcement learning. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020. 1–15.

[27] Zhang J, Yu HN, Xu W. Hierarchical reinforcement learning by discovering intrinsic options. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2021. 1–19.

[28] Gregor K, Rezende DJ, Wierstra D. Variational intrinsic control. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017. 1–15.

[29] Hénaff OJ, Srinivas A, De Fauw J, Razavi A, Doersch C, Eslami SMA, van Den Oord A. Data-efficient image recognition with contrastive predictive coding. In: Proc. of the 37th Int'l Conf. on Machine Learning. JMLR.org, 2020. 391.

[30] He KM, Fan HQ, Wu YX, Xie SN, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9726–9735. [doi: 10.1109/CVPR42600.2020.00975]

[31] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: Proc. of the 37th Int'l Conf. on Machine Learning. JMLR.org, 2020. 149.

[32] Chen XL, He KM. Exploring simple Siamese representation learning. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 15745–15753. [doi: 10.1109/CVPR46437.2021.01549]

[33] Grill JB, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, Doersch C, Pires BA, Guo ZD, Azar MG, Piot B, Kavukcuoglu K, Munos R, Valko M. Bootstrap your own latent a new approach to self-supervised learning. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 1786.

[34] Kaelbling LP. Learning to achieve goals. In: Proc. of the 13th Int'l Joint Conf. on Artificial Intelligence. Chambéry: Morgan Kaufmann, 1993. 1094–1099.

[35] Schaul T, Horgan D, Gregor K, Silver D. Universal value function approximators. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 1312–1320.

[36] Pong V, Gu SX, Dalal M, Levine S. Temporal difference models: Model-free deep RL for model-based control. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[37] Zhao R, Sun XD, Tresp V. Maximum entropy-regularized multi-goal reinforcement learning. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 7553–7562.

[38] Eysenbach B, Salakhutdinov R, Levine S. C-Learning: Learning to achieve goals via recursive classification. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2021.

[39] Andrychowicz M, Wolski F, Ray A, Schneider J, Fong R, Welinder P, McGrew B, Tobin J, Abbeel P, Zaremba W. Hindsight experience replay. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 5055–5065.

[40] Nasiriany S, Pong VH, Lin S, Levine S. Planning with goal-conditioned policies. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 1329.

[41] Burda Y, Edwards H, Storkey AJ, Klimov O. Exploration by random network distillation. In: Proc. of the 7th Int'l Conf. on Learning Representations. New Orleans: OpenReview.net, 2019.

[42] Ecoffet A, Huizinga J, Lehman J, Stanley KO, Clune J. First return, then explore. Nature, 2021, 590(7847): 580–586. [doi: 10.1038/s41586-020-03157-9]

[43] Jiang YD, Liu EZ, Eysenbach B, Kolter JZ, Finn C. Learning options via compression. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: Curran Associates Inc., 2022. 1540.

[44] Todorov E, Erez T, Tassa Y. MuJoCo: A physics engine for model-based control. In: Proc. of the 2012 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems. Vilamoura-Algarve: IEEE, 2012. 5026–5033. [doi: 10.1109/IROS.2012.6386109]

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

[46] Neil D, Segler MHS, Guasch L, Ahmed M, Plumbley D, Sellwood M, Brown N. Exploring deep recurrent models with reinforcement learning for molecule design. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

附中文参考文献:

[5] 余超, 董银昭, 郭宪, 冯旸赫, 卓汉逵, 张强. 结构交互驱动的机器人深度强化学习控制方法. 软件学报, 2023, 34(4): 1749–1764. http://www.jos.org.cn/1000-9825/6708.htm [doi: 10.13328/j.cnki.jos.006708]

[6] 王金永, 黄志球, 杨德艳, Huang XW, 祝义, 华高洋. 面向无人驾驶时空同步约束制导的安全强化学习. 计算机研究与发展, 2021, 58(12): 2585–2603. [doi: 10.7544/issn1000-1239.2021.20211023]

[7] 轩书哲, 柯良军. 基于多智能体强化学习的无人机集群攻防对抗策略研究. 无线电工程, 2021, 51(5): 360–366. [doi: 10.3969/j.issn.1003-3106.2021.05.004]

杨尚东(1990－), 男, 博士, 讲师, 主要研究领域为强化学习, 多智能体系统, 机器学习.

陈兴国(1984－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为强化学习, 游戏人工智能, 机器学习.

余淼盈(1998－), 女, 硕士生, 主要研究领域为强化学习, 机器学习, 数据挖掘.

陈蕾(1975－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为机器学习, 模式识别.