计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202220970
61(12): 3108−3120, 2024

基于兴趣函数的多样化 Option-Critic 算法

栗军伟[1] 刘 全[1,2,3,4] 黄志刚[1] 徐亚鹏[1]

[1](苏州大学计算机科学与技术学院 江苏苏州 215006)

[2](软件新技术与产业化协同创新中心(南京大学) 南京 210023)

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

[4](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)

(Li_junwei2021@163.com)

A Diversity-Enriched Option-Critic Algorithm with Interest Functions

Li Junwei[1], Liu Quan[1,2,3,4], Huang Zhigang[1], and Xu Yapeng[1]

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

[2](*Collaborative Innovation Center of Novel Software Technology and Industrialization* (*Nanjing University*), *Nanjing* 210023)

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

[4](*Jiangsu Provincial Key Laboratory for Computer Information Processing Technology* (*Soochow University*), *Suzhou*, *Jiangsu* 215006)

Abstract As a common temporal abstraction method for hierarchical reinforcement learning, Option framework allows agents to learn strategies at different time scales, which can effectively solve sparse reward problems. In order to ensure that options can guide agents to access more state space, some methods improve the diversity of options by introducing mutual information in internal reward and termination functions. However, it will lead to slow algorithm learning speed and low knowledge transfer ability of internal strategy, which seriously affect algorithm performance. To address the above problems, diversity-enriched option-critic algorithm with interest functions(DEOC-IF) is proposed. Based on the diversity-enriched option-critic (DEOC) algorithm, the algorithm constrains the selection of the upper-level strategy on the internal strategy of Option by introducing the interest function, which not only ensures the diversity of the Option set, but also makes the learned internal strategies focus on different regions of the state space, which is conducive to improving the knowledge transfer ability of the algorithm and accelerating the learning speed. In addition, DEOC-IF introduces a new interest function update gradient, which is beneficial to improve the exploration ability of the algorithm. In order to verify the effectiveness and option reusability of the algorithm, the algorithm comparison experiments are carried out in four-room navigation task, Mujoco, and MiniWorld. Experimental results show that DEOC-IF algorithm has better performance and option reusability compared with other algorithms.

Key words reinforcement learning; temporal abstractions; Option framework; interest function; Option-Critic algorithm

摘 要 Option 框架作为分层强化学习的一种常用时序抽象方法,允许智能体在不同的时间尺度上学习策略,可以有效解决稀疏奖励问题.为了保证 Option 可以引导智能体访问更多的状态空间,一些方法通过引

收稿日期: 2022-11-23; **修回日期**: 2024-01-03

基金项目: 国家自然科学基金项目 (62376179, 61772355, 61702055, 61876217, 62176175); 江苏高校优势学科建设工程资助项目

This work was supported by the National Natural Science Foundation of China (62376179, 61772355, 61702055, 61876217, 62176175) and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

通信作者: 刘全(quanliu@suda.edu.cn)

入基于互信息的内部奖励和终止函数来提升 Option 内部策略的多样性. 但这会导致算法学习速度慢和内部策略的知识迁移能力低等问题，严重影响了算法性能. 针对以上问题，提出基于兴趣函数优化的多样化 Option-Critic 算法（diversity-enriched Option-Critic algorithm with interest functions, DEOC-IF）. 该算法在多样化 Option-Critic 算法（diversity-enriched Option-Critic, DEOC）的基础上，通过引入兴趣函数约束上层策略对 Option 内部策略的选择，既保证了 Option 集合的多样性，又使得学习到的内部策略可以关注状态空间的不同区域，有利于提高算法的知识迁移能力，加快学习速度. 此外，DEOC-IF 算法引入一种新的兴趣函数更新梯度，有利于提高算法的探索能力. 为了验证算法的有效性和知识迁移能力，分别在 4 房间导航任务、Mujoco 和 MiniWorld 实验环境中，将 DEOC-IF 算法与其他最新算法进行对比实验. 结果表明，DEOC-IF 算法具有更好的性能优势和策略迁移能力.

关键词 强化学习；时序抽象；Option 框架；兴趣函数；Option-Critic 算法

中图法分类号 TP391

强化学习[1]（reinforcement learning, RL）是机器学习研究领域的一个重要分支，主要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. 深度强化学习[2-3]（deep reinforcement learning, DRL）把深度学习（deep learning, DL）应用到强化学习中，使得学习过程同时具备深度学习的感知能力和强化学习的决策能力. DRL 可以用来解决强化学习中具有高维状态和动作空间的任务. 在深度 Q 网络[4]（deep q-network, DQN）提出之后，出现了多种改进的经典算法[5-8]，提升了 DRL 智能体的探索能力和算法鲁棒性. 然而，DRL 算法在面对稀疏奖励任务、长时序延迟奖励任务和具有分层结构的任务时，问题的求解会变得很困难，智能体很难学习到最优的策略. 分层强化学习[9]（hierarchical reinforcement learning, HRL）的抽象机制可以用来解决上述问题. 分层强化学习以半马尔可夫决策过程[10]（semi-Markov decision process, SMDP）为理论基础，通过分层抽象机制提高智能体的探索能力和策略的知识迁移能力. Kulkarni 等人[11]将深度学习与分层强化学习结合，提出了深度分层强化学习（deep hierarchical reinforcement learning, DHRL）. DHRL 形成了上层策略控制下层策略探索环境的问题求解方式，实现了更强的特征提取能力和策略学习能力. 近年来 DHRL 已经被成功应用于推荐系统[12]、自动驾驶[13]和自然语言处理[14]等领域，成为强化学习领域的研究热点之一. 关于 DHRL 的研究，其方向主要包括 2 种类型：1）基于子目标[15]. 该种方法利用特征提取把状态空间中部分状态位置作为子目标. 其解决任务的过程为上层策略产生子目标，下层策略负责指导智能体到达子目标，重复此过程直到智能体到达最终目标位置. 2）基于 Option[10]. 上层策略调用下层网络学习到的一组内部策略解决整个任务. 关于 Option 深度分层强化学习的研究热点，主要集中于 Option 集合的多样性，以及 Option 内部策略的可重复利用性. 本文提出的方法为基于 Option 分层强化学习.

Option 框架作为分层控制机制的一种，可以被视为一种特殊的技能，允许强化学习智能体使用长时序行为去表征和规划，从而改善策略的学习效率以及策略的知识迁移能力. 如何端到端地学习到好的下层 Option 内部策略一直以来是 Option 框架的一个挑战. Bacon 等人[16]提出了 OC（Option-Critic）框架，把策略梯度理论应用到 Option 学习上，从而可以通过梯度更新参数的方式端到端地学习 Option 内部策略、终止函数以及上层策略. 然而，在策略学习过程中，OC 框架中的下层 Option 内部策略集合会出现退化问题. 比如，下层 Option 内部策略频繁终止使得 Option 变为简单行为；一个 Option 内部策略控制了整个任务，其他 Option 内部策略得不到及时更新；以及下层 Option 内部策略集合的状态-动作分布变得非常相似. 上述问题严重影响了智能体的实验性能以及策略的知识迁移能力. 为了阻止 Option 内部策略频繁终止，Harb 等人[17]在 OC 框架的基础上通过引入思考成本（deliberation cost）η，提出了异步优势 Option-Critic（asynchronous advantage Option-Critic, A2OC）算法. A2OC 算法提出一种新的优化目标，在最大化奖励函数的同时，最小化下层 Option 内部策略的切换代价. 因此，A2OC 算法鼓励下层内部策略执行的步数更长，解决了 OC 框架的退化问题. Klissarov 等人[18]把近端策略优化[19]（proximal policy optimization, PPO）算法与 OC 框架相结合，提出了近端策略 Option-Critic（proximal policy Option-Critic, PPOC）算法，提高了算法学习速度和算法实验性能. 为了解决下层 Option 内部策略

集合相似的问题，Li 等人[20] 提出了一种基于异策略的软 Option 行动者-评论家(soft Option actor-critic, SOAC)算法. SOAC 算法基于最大熵模型，引入了基于信息论的内部奖励，提高了智能体对环境的探索能力，从而可以学习到多样化的 Option 集合. 为了学习到适合探索的多样化 Option 内部策略，Kanagawa 等人[21] 提出了信息最大化选项-评论家(infomax Option-Critic, IMOC)算法. IMOC 算法引入信息最大化终止目标，从而保证 Option 集合的多样性. 大多数关于学习多样化 Option 的研究文章使用状态区分和特殊化 Option 内部策略集合[22-24]. 与此不同，Kamat 等人[25] 根据 Option 内部策略的动作分布来评价其多样性，提出了多样化 Option-Critic(diversity-enriched Option-Critic, DEOC)算法，并且在 DEOC 算法的基础上，为了保证最大熵伪奖励函数产生作用，引入了新的终止函数，提出 TDEOC(termination-DEOC)算法. 在状态信息不够完整以及智能体旨在任务之间迁移知识时，TDEOC 算法可以获得很好的性能. 本文在 TDEOC 算法基础上，把 Option 初始集表示为兴趣函数[26]，不再假设每一个 Option 在每个状态都会被选择到，而是限制上层策略对 Option 的可选择性.

目前，关于 Option 集合的学习的主要难点包括：1)上层策略如何在特定状态选择合适的下层 Option 内部策略，保证学习到的 Option 具有知识迁移能力；2)终止函数如何确定恰当的终止位置，阻止 Option 集合的退化问题. 针对这 2 个问题，本文提出一种基于兴趣函数的多样化 Option-Critic 算法(diversity-enriched Option-Critic algorithm with interest functions, DEOC-IF). DEOC-IF 算法把 Option 的初始集合表示为参数化的兴趣函数，限制下层 Option 内部策略在特定状态的可选择性，不仅降低了 TDEOC 算法在 Option 选择时的性能开销，还提高了 Option 关于状态的特殊性，有利于提高算法学习速度和实验性能，以及下层策略的知识迁移能力.

本文的贡献主要包括 3 个方面：

1) 提出了一种新的学习多样化 Option 内部策略的 DEOC-IF 算法框架，该框架在 TDEOC 算法的基础上引入了兴趣函数，有利于提高算法学习速度和知识迁移能力；

2) 提出一种新的兴趣函数参数更新梯度，可以提高智能体对环境的探索能力，有效提高算法性能；

3) 通过基础实验和迁移实验，将 DEOC-IF 算法与常见的 DRL 算法和同类型的 DHRL 算法进行对比，验证所提算法的优越性.

1 背景知识

1.1 强化学习

强化学习是一种智能体通过不断地试错与环境进行交互的学习方式，然后利用环境的反馈信息通过反向传播改善策略网络. 强化学习训练模型通常以马尔可夫决策过程为理论基础，目标是最大化期望折扣累计奖励. 马尔可夫决策过程通常被定义为一个五元组 (S,A,γ,R,P)，其中 S 是所有环境状态的集合，A 是所有智能体可采取动作的集合，γ 是折扣因子. 奖励函数 $R:S\times A\to\mathbb{R}$ 是关于状态和动作的映射函数，P 表示状态转移函数. 在强化学习过程中，假设智能体当前观察状态为 $s_t\in S$，根据策略 π 选择动作 $a_t\in A$，与环境交互获得奖赏 $r\in\mathbb{R}$，并转移到下一状态 s_{t+1}，重复该过程，直至到达终止状态. 给定一个策略 π，状态值函数 V 被定义为

$$V_\pi(s)=E_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}|s_0=s\right]. \tag{1}$$

在状态 s 下采取动作 a 的动作值函数 Q 被定义为

$$Q_\pi(s,a)=E_\pi\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}|s_0=s,a_0=a\right]. \tag{2}$$

状态值函数 V_π 满足贝尔曼等式：

$$V_\pi(s)=\sum_a\pi(a|s)\left(r(s,a)+\gamma\sum_{s'}P(s'|s,a)V_\pi(s')\right). \tag{3}$$

强化学习的目标是找到一个最优策略 π^*，使得期望折扣累计奖赏最大，即策略满足 $V_{\pi^*}(s)=\max\limits_\pi V_\pi(s)$. 策略梯度理论[27] 提供了一种使用期望折扣累计回报优化随机策略 π_θ 的方法，期望折扣累计回报关于参数 θ 的梯度为

$$\frac{\partial L(\theta)}{\partial\theta}=E\left[\frac{\partial\log\pi(A_t|S_t)}{\partial\theta}A^\pi(S_t,A_t)\right], \tag{4}$$

其中 $A^\pi(S_t,A_t)=Q_\pi(S_t,A_t)-V_\pi(S_t)$ 是优势函数，S_t 和 A_t 分别为时刻 t 的状态和动作.

目前，深度强化学习融合了深度学习的感知能力和强化学习的决策能力，成为人工智能领域研究热点，并在很多方面得到应用. 在星际争霸游戏中，DeepMind 团队研发的 Alpha Star 击败了人类选手；谷歌把深度强化学习用于 YouTube 视频推荐，取得优异效果.

1.2 OC（Option-Critic）框架

Option 框架是分层强化学习中分层机制的一种. 一个 Option 内部策略 $o\in\Omega$ 由三元组 (I_o,π_o,β_o) 定义，其中 $I_o\subseteq S$ 是下层内部策略的初始状态集合，π_o 表示

Option 内部策略，终止条件$\beta_o: S \to [0,1]$是当前选择的 Option 内部策略在到达下一个状态s'后的终止概率. 在 Option 框架下，假设智能体当前所在状态为s，上层策略可以选择的下层 Option 内部策略集合为$\Omega(s)$. 则 Option 框架的执行过程为：上层策略$\pi_\Omega(s)$从集合$\Omega(s)$中选择一个 Option 内部策略o，智能体按照内部策略π_o执行动作，直到满足终止条件β_o. 然后上层策略重新选择一个下层 Option 内部策略o'，重复执行上述过程，直至完成整个任务.

OC 框架提出了一种学习下层 Option 内部策略和终止函数参数的梯度更新方法，假定了所有 Option 内部策略在每个状态都可以被上层策略选择. OC 框架使用$\pi_{o,\theta}$表示参数化后的 Option 内部策略，使用$\beta_{o,\vartheta}$表示参数化后的 Option 终止函数. 其中，θ和ϑ分别为π_o和β_o的参数，可以根据策略梯度定理进行参数优化. 在 OC 框架中，Option 值函数通常被定义为

$$Q_\Omega(s,o)=\sum_a \pi_{o,\theta}(a|s)Q_U(s,o,a), \tag{5}$$

其中$Q_U: S\times\Omega\times A\to\mathbb{R}$是指在状态$s$根据内部策略$o$选择动作$a$时的值函数. 可以表示为

$$Q_U(s,o,a)=r(s,a)+\gamma\sum_{s'}P(s'|s,a)U(o,s'). \tag{6}$$

$U:\Omega\times S\to\mathbb{R}$反映的是智能体选择内部策略$o$后，到达下一状态$s'$的价值函数，可定义为

$$U(o,s')=(1-\beta_{o,\vartheta}(s'))Q_\Omega(s',o)+\beta_{o,\vartheta}(s')V_\Omega(s'). \tag{7}$$

给定内部策略参数θ和初始状态策略对(s_0,o_0)，那么期望折扣累计回报关于参数θ的梯度为

$$\sum_{s,o}\mu_\Omega(s,o|s_0,o_0)\sum_a\frac{\partial\pi_{o,\theta}(a|s)}{\partial\theta}Q_U(s,o,a), \tag{8}$$

其中$\mu_\Omega(s,o|s_0,o_0)$表示从(s_0,o_0)到(s,o)轨迹的折扣权重. 给定终止函数参数ϑ和初始状态策略对(s_1,o_0)，则期望折扣累计回报关于参数ϑ的梯度为

$$-\sum_{s',o}\mu_\Omega(s',o|s_1,o_0)\frac{\partial\beta_{o,\vartheta}(s')}{\partial\vartheta}A_\Omega(s',o), \tag{9}$$

其中$\mu_\Omega(s',o|s_1,o_0)$表示从(s_1,o_0)到(s',o)轨迹的折扣权重.

1.3 多样化 Option-Critic 算法

OC 框架以一种无模型的、端到端的方式学习长时序动作策略，改善了智能体策略学习和动作规划的效率. 然而，这种方式学习到的下层 Option 内部策略集合会出现退化问题. 比如 Option 内部策略之间的区分度较低，以及 Option 内部策略频繁终止等. 这些退化问题严重影响了 OC 框架的实验性能. Kamat 等人[25] 提出了 TDEOC 算法，可以有效地解决上述问题. 大多数研究工作根据状态学习到多样化的 Option 内部策略集合，而 TDEOC 算法根据 Option 内部策略的动作分布评价其多样性. 当环境的状态空间为部分可观测或智能体旨在多个任务之间迁移知识时，这种方法非常合适. 为了保证 Option 内部策略集合的多样性，TDEOC 算法使用信息熵构造伪奖励. 在实验环境中，TDEOC 算法使用 2 个 Option 内部策略. 因此，引入的伪奖励函数可表示为

$$R_{\text{bonus}}=H(A^{\pi_{o_1}}|S)+H(A^{\pi_{o_2}}|S)+H(O^{\pi_\Omega}|S)+H(A^{\pi_{o_1}};A^{\pi_{o_2}}|S), \tag{10}$$

其中H表示底数为 e 的香农熵，A表示动作分布. 式(10)等号右侧的前 2 项是为了增加下层内部策略选择动作时的随机性；第 3 项是为了增加上层策略在选择下层内部策略时的随机性；第 4 项是为了鼓励不同的 Option 内部策略彼此之间动作分布的多样性. 把上述伪奖励函数引入到标准强化学习框架中，得到增广奖励函数：

$$R_{\text{aug}}(S_t,A_t)=(1-\tau)R(S_t,A_t)+\tau R_{\text{bonus}}(S_t), \tag{11}$$

其中，τ是一个超参数，控制伪奖励函数在增广奖励函数中的相对重要程度. 当$\tau\to 0$时，增广奖励函数趋近于标准强化学习目标. 通过最大化香农熵的奖励函数目标，可以鼓励下层内部策略动作分布的多样性. 实验表明，增广奖励函数阻止策略快速收敛到局部最优，表现出了更强的探索能力以及策略的鲁棒性.

但是，当一个 Option 内部策略正在被执行时，集合中其他的 Option 内部策略只能保持休眠. 这会导致没有被选择到的 Option 内部策略无法得到及时的更新，最大熵目标就没有发挥出真正的作用. 因此，TDEOC 算法提出新的终止函数：

$$L(\theta_\beta)=E[\beta(S_t,O_t)D(S_t)], \tag{12}$$

其中$D(S_t)$表示 Option 内部策略集合在给定状态的相对多样性，被定义为

$$D(S_t)=\frac{R_{\text{bonus}}(S_t)-\mu_{R_{\text{bonus}}}}{\sigma_{R_{\text{bonus}}}}, \tag{13}$$

其中，$\mu_{R_{\text{bonus}}},\sigma_{R_{\text{bonus}}}$分别为$R_{\text{bonus}}$的平均值和标准差. 与 OC 框架的终止函数相比，终止函数不再最大化期望折扣累计回报，而是关注 Option 内部策略集合的动作多样性. 当未被选择的 Option 内部策略在当前状态展现出不同的动作时，被选择到的 Option 内部策略则倾向于终止在此状态.

2 基于兴趣函数优化的 Option-Critic 算法

本节主要阐述 DEOC-IF 算法的具体架构以及理论

分析. 其中2.1节介绍兴趣Option-Critic(interest-Option-Critic, IOC)算法框架[26], 2.2节介绍DEOC-IF算法的整体框架.

2.1 IOC框架

人类常会利用已掌握的经验知识或者技能来尝试解决新的任务. 即使存在多种任务的解决方案, 人类也会考虑根据效率、兴趣等因素去除一些低满意度的解决方案, 从剩下的一组方案中选择最优的解决方案. Option框架借鉴这一思想, 用了一个类似的概念, 即下层Option内部策略的初始集. 在数学中, 指示函数$1_I: S \to \{0,1\}$是定义在集合S上的函数, 表示其中有哪些元素属于某一子集I, 即

$$1_I(s)=\begin{cases}1, & 若s\in I,\\ 0, & 若s\notin I.\end{cases} \tag{14}$$

Khetarpal等人[26]把指示函数推广到兴趣函数$I_o: S\times\Omega\to\mathbb{R}_+$来表示下层内部策略$o$的初始状态集合. 若下层内部策略$o$可以在状态$s$处被初始化, 则$I_o(s)>0$. $I_o(s)$的值越大, 表示在状态s处上层策略对于下层内部策略o的兴趣值越大. 为了学习优化I_o, I_o被表示为参数化的函数$I_{o,z}$, 可通过策略梯度方法更新其参数z. IOC算法定义兴趣上层策略为

$$\pi_{I_{o,z}}(o|s)=I_{o,z}(s)\pi_\Omega(o|s)\Big/\sum_{o'}I_{o',z}(s)\pi_\Omega(o'|s). \tag{15}$$

假定o_t在时刻t被初始化或者被启动, 那么从元组(s_t,o_t)转移到元组(s_{t+1},o_{t+1})的一步折扣概率为

$$P_\gamma^{(1)}(s_{t+1},o_{t+1}|s_t,o_t)=\sum_a\pi_{o_t,\theta}(a|s_t)\gamma P(s_{t+1}|s_t,a)\Big((1-\beta_{o_t,\vartheta}(s_{t+1}))1_{o_t=o_{t+1}}+\beta_{o_t,\vartheta}(s_{t+1})\Big(I_{o_{t+1},z}(s_{t+1})\pi_\Omega(o_{t+1}|s_{t+1})\Big/\sum_o I_{o,z}(s_{t+1})\pi_\Omega(o|s_{t+1})\Big)\Big). \tag{16}$$

通过迭代, k步折扣概率可以被表示为

$$P_\gamma^{(k)}(s_{t+k},o_{t+k}|s_t,o_t)=\sum_{s_{t+1}}\sum_{o_{t+1}}\Big(P_\gamma^{(1)}(s_{t+1},o_{t+1}|s_t,o_t)P_\gamma^{(k-1)}(s_{t+k},o_{t+k}|s_{t+1},o_{t+1})\Big). \tag{17}$$

则期望折扣回报关于参数z的梯度为

$$\frac{\partial Q_\Omega(s,o)}{\partial z}=\frac{\partial}{\partial z}\left\{\sum_a\pi_{o,\theta}(a|s)Q_U(s,o,a)\right\}. \tag{18}$$

使用式(6)(7)对式(8)展开, 有

$$\frac{\partial Q_\Omega(s,o)}{\partial z}=\sum_a\pi_{o,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\Big((1-\beta_{o,\vartheta}(s'))\frac{\partial Q_\Omega(s',o)}{\partial z}+\beta_{o,\vartheta}(s')\frac{\partial V_\Omega(s')}{\partial z}\Big). \tag{19}$$

又$V_\Omega(s)=\sum_o\pi_{I_{o,z}}(o|s)Q_\Omega(s,o)$, 将其和式(15)代入式(19), 得

$$\frac{\partial Q_\Omega(s,o)}{\partial z}=\sum_a\pi_{o,\theta}(a|s)\sum_{s'}\gamma P(s'|s,a)\sum_{o'}\beta_{o,\vartheta}(s')\times\Big(\frac{\partial\pi_{I_{o,z}}(o'|s')}{\partial z}Q_\Omega(s',o')\Big)+\sum_{s'}\sum_{o'}\times\Big(\sum_a\pi_{o,\theta}(a|s)\gamma P(s'|s,a)\Big((1-\beta_{o,\vartheta}(s'))1_{o=o'}+\beta_{o,\vartheta}(s')\pi_{I_{o,z}}(o'|s')\Big)\Big)\frac{\partial Q_\Omega(s',o')}{\partial z}. \tag{20}$$

把式(16)代入式(20)并推广到k步, 可得

$$\frac{\partial Q_\Omega(s,o)}{\partial z}=\sum_{k=0}^{\infty}\sum_{s',o'}P_\gamma^{(k)}(s',o'|s,o)\times\Big(\beta_{o,\vartheta}(s')\frac{\partial\pi_{I_{o,z}}(o'|s')}{\partial z}Q_\Omega(s',o')\Big), \tag{21}$$

故期望折扣回报关于兴趣函数参数z的梯度为

$$\sum_{s',o'}\hat{\mu}_\Omega(s',o'|s,o)\beta_o(s')\frac{\partial\pi_{I_{o,z}}(o'|s')}{\partial z}Q_\Omega(s',o'), \tag{22}$$

其中$\hat{\mu}_\Omega(s',o'|s,o)=\sum_{k=0}^{\infty}P_\gamma^{(k)}(s',o'|s,o)$表示根据兴趣上层策略$\pi_{I_{o,z}}$从$(s,o)$到$(s',o')$轨迹采样的折扣权重.

2.2 DEOC-IF算法描述

虽然兴趣函数能够限制上层策略根据兴趣值的大小去选择恰当的下层Option内部策略最大化期望折扣回报, 但是并没有保证Option内部策略集合的多样性. OC框架中出现的Option集合关于状态-动作分布非常相似的退化问题仍然会出现, 从而影响最终的算法性能. TDEOC算法假设在每一个状态下层所有Option内部策略都会被选择到, 并没有考虑到Option和状态的相关性, 会使得上层策略选择到不恰当的Option, 从而影响了策略的知识迁移能力. 而且, TDEOC算法为了保证下层内部策略的多样性, 引入的伪奖励会导致算法整体的学习速度变慢. 针对以上问题, 本文提出了基于兴趣函数优化的多样化Option-Critic算法, 即DEOC-IF算法. DEOC-IF算法继承了TDEOC算法中引入的伪奖励函数和新的终止函数, 保证了Option集合的多样性. 同时, DOEC-IF算法把Option元组中的初始集合推广为兴趣函数$I_{o,z}$, 在上层策略选择下层Option内部策略时, 就会更关注Option内部策略的适用程度. 例如, 一个在红绿灯路口指导智能体执行直行和停步动作的Option内部策略, 只有遇到下一个红绿灯路口才会被上层策略重新选择. 这样, 学习到的Option集合更具有解释性, 有利于迁移学习. 但是, 为了进一步提升智能体对环

境的探索能力，当上层策略在选择下层内部策略时，不仅考虑兴趣函数值的大小，还考虑下层内部策略的探索能力. 这里用随机性来衡量下层内部策略对环境的探索能力. 为实现此目的，修改兴趣函数的梯度为

$$\frac{\partial L(z)}{\partial z}=E\left[\beta_{o,\vartheta}(s')\frac{\partial \pi_{I_{o,z}}(o'|s')}{\partial z}\left(Q_{\Omega}(s',o')+cH(\pi_{o'}(\cdot|s'))\right)\right], \tag{23}$$

其中c是正则项系数. 当该系数非常小时，正则项的引入对于性能没有危害，反而会鼓励上层策略选择探索能力强的下层内部策略以提升实验性能，后文会通过消融实验进行证明.

当下层内部策略采用正态分布的形式表示时，正则项的信息熵可以推导为

$$\begin{aligned}H=&-\int p(x)\ln p(x)\mathrm{d}x=-\int_{-\infty}^{\infty}(2\pi\sigma^2)^{-\frac{1}{2}}\mathrm{e}^{-(x-\mu)^2/2\sigma^2}\times\\&\ln\left[(2\pi\sigma^2)^{-\frac{1}{2}}\mathrm{e}^{-(x-\mu)^2/2\sigma^2}\right]\mathrm{d}x=\frac{1}{2}\ln(2\pi\sigma^2)\int_{-\infty}^{\infty}(2\pi\sigma^2)^{-\frac{1}{2}}\times\\&\mathrm{e}^{-(x-\mu)^2/2\sigma^2}\mathrm{d}x+\frac{1}{2\sigma^2}\int_{-\infty}^{\infty}(2\pi\sigma^2)^{-\frac{1}{2}}(x-\mu)^2\mathrm{e}^{-(x-\mu)^2/2\sigma^2}\mathrm{d}x=\\&\frac{1}{2}(\ln(2\pi\sigma^2)+1).\end{aligned} \tag{24}$$

故正态分布的信息熵仅与方差σ有关，可以表示为方差的函数.

基于兴趣函数优化的多样化 Option-Critic 算法模型的整体框架如图 1 所示. 智能体根据兴趣上层策略选择一个下层 Option 策略o，然后根据策略o选取动作作用于环境进入下一个状态. DEOC-IF 算法根据环境给予的奖励求出梯度值并反向优化网络参数. 重复此过程，直至满足优化目标.

算法 1 介绍了 DEOC-IF 算法的整个过程，其是对图 1 的详细描述. 与 TDEOC 算法不同，上层策略不再完全依赖于π_{Ω}来选取下层内部策略，而是根据兴趣上层策略$\pi_{I_{o,z}}$选取合适的下层内部策略o，如行②所示. 行③~⑮是策略评估过程，行⑯~⑱是策略优化过程，即对各参数的更新优化. 其中，行⑥是引入的增广奖励函数，行⑯是对内部策略o的参数更新，行⑰是对终止函数参数的更新，行⑱是对兴趣函数参数的更新. 算法 1 整体采用表格式内部 Option Q-learning 更新方式.

算法 1. 基于兴趣函数优化的多样化 Option-Critic 算法.

输入：超参数τ和c，折扣率γ，Option 内部学习率α_{θ}，终止函数学习率α_{ϑ}，兴趣函数学习率α_z；

输出：网络参数.

① 初始化：$\pi_{\Omega},\pi_o,\beta_o$以及关于参数$z$的兴趣函数$I_{o,z}$；

② 根据$\pi_{I_{o,z}}$选择一个o；

③ Repeat:

④ 根据策略$\pi_{o,\theta}(a|s)$选择一个动作a；

⑤ 执行动作a，观测状态s'，得到奖赏r；

⑥ $r'=(1-\tau)r+\tau r_{\mathrm{bonus}}(s)$；

⑦ if o在状态s'终止

⑧ 根据$\pi_{I_{o,z}}(\cdot|s')$选择一个新的o'；

⑨ else

⑩ $o'=o$；

⑪ end if

⑫ 根据$r_{\mathrm{bonus}}(s)$正则化采样得到$D(s)$；

⑬ $\delta\leftarrow r'-Q_U(s,o,a)$；

⑭ $\delta\leftarrow\delta+\gamma(1-\beta_o(s'))Q_{\Omega}(s',o')+\gamma\beta_o(s')\max\limits_{o'}Q_{\Omega}(s',o')$；

⑮ $Q_U(s,o,a)\leftarrow Q_U(s,o,a)+\alpha\delta$；

⑯ $\theta\leftarrow\theta+\alpha_{\theta}\dfrac{\partial\log\pi_{o,\theta}(a|s)}{\partial\theta}Q_U(s,o,a)$；

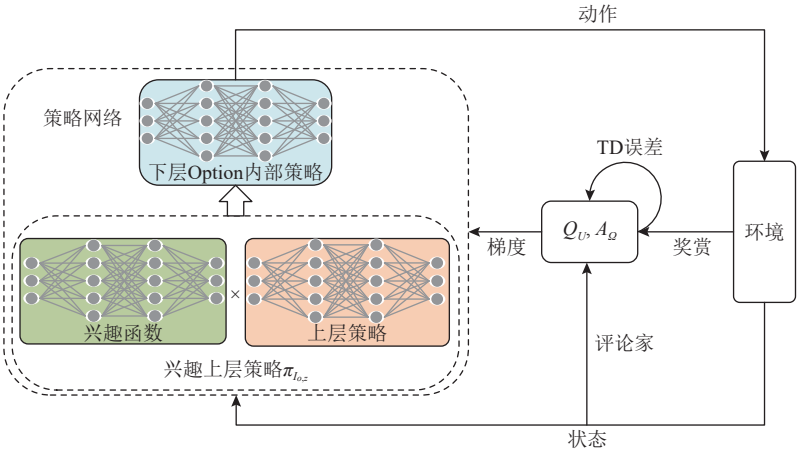


Fig. 1 Illustration of DEOC-IF algorithm

图 1 DEOC-IF 算法示意图

⑰ $\vartheta \leftarrow \vartheta - \alpha_\vartheta \frac{\partial \beta_{o,\vartheta}(s')}{\partial \vartheta}(Q_\Omega(s',o) - V_\Omega(s'))$;

⑱ $z \leftarrow z + \alpha_z \beta_{o,\vartheta}(s') \frac{\partial \pi_{I_{o,z}}(o'|s')}{\partial z}(Q_\Omega(s',o') + cH(\pi_{o'}(\cdot|s')))$;

⑲ Until s'是终止状态.

3 实 验

为了评估 DEOC-IF 算法的性能优势和策略的知识迁移能力, 分别在离散环境、连续环境以及迁移环境下进行实验. 同时, 为了验证正则项的引入对实验性能的益处, 在 3.5 节做了消融实验. 本文选择 PPO 作为基础算法, 因为 PPO 不仅可以应用在连续控制任务中, 还可以应用在离散控制任务中, 并获得很好的性能表现. 而且, OC 框架已经在 PPO 上得到了扩展, 即 PPOC 算法, 在 Mujoco 实验上取得了优异效果. 本节实验旨在评估 DEOC-IF 算法的 3 个方面:

1) 兴趣函数的引入是否可以提高算法学习速度和实验性能;

2) 学习到的 Option 集合的知识迁移能力是否提高;

3) 带正则项的兴趣函数更新方式是否有益于实验性能.

3.1 4 房间导航任务

本文首先在经典的 4 房间导航任务[10] 中测试 DEOC-IF 算法的性能以及 Option 集合的知识迁移能力.

3.1.1 实验环境及设置

4 房间环境是分层强化学习常用的离散环境. 与经典 4 房间环境不同的是本文提高了环境的难度, 并增加了迁移实验部分. 起初, 智能体从不包含目标位置的状态空间中随机选择一个状态作为起始位置, 目标位置为房间的右面过道处, 如图 2 所示. 实验经过 1 000 个情节之后, 在右下方房间随机选择一个新的位置作为目标状态. 如果智能体到达了目标位置, 会获得+1 奖赏, 到达其他位置获得的奖赏为 0.

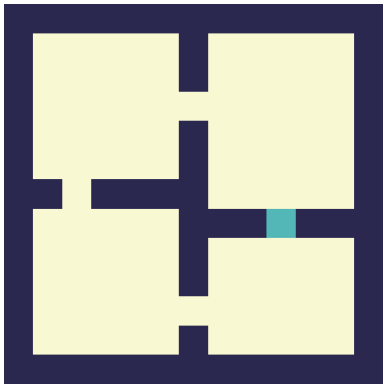


Fig. 2 Four-room navigation task environment

图 2 4 房间导航任务环境

下层 Option 内部策略采用的是内部 Option Q-learning 方式进行参数更新, 所有权重均被初始化为 0. 为了与对比算法保持一致, 对于 DEOC-IF 算法, 本实验使用 4 个下层 Option 内部策略, 一个情节的最大步长数为 1 000, 正则项系数为 0.001. 其他超参数如表 1 所示.

Table 1 Hyper-Parameters of Four-Room Navigation Task

表 1 4 房间导航任务超参数

超参数	参数值
α_ϑ	0.25
α_θ	0.25
γ	0.99
α_z	0.15

3.1.2 实验结果分析

本节把 DEOC-IF 算法和 TDEOC 算法、OC 算法进行实验对比. 这 3 种算法都可以达到目标位置, 使用对应情节下的累积奖赏值作为纵坐标不易看出这 3 种算法的区别. 本文使用智能体到达目标位置所需要的时间步数作为纵坐标, 可以明显看出 3 种算法的差异. 实验效果如图 3 所示, 每个算法的实验性能取自 300 次运行的平均值. 在开始的时候, 智能体的移动方向是不确定的, 在训练中逐渐改变移动方向的概率. 在前 1 000 个情节中, 3 种算法几乎有相同的收敛速度. 在改变目标状态的位置之后, 与 TDEOC 算法和 OC 算法相比, DEOC-IF 算法能更快使得智能体到达改变后的目标位置, 且有更好的收敛效果. TDEOC 算法引入伪奖励, 并且采用了新的终止函数, 提高了算法的探索能力, 从而能够保证学习到的 Option

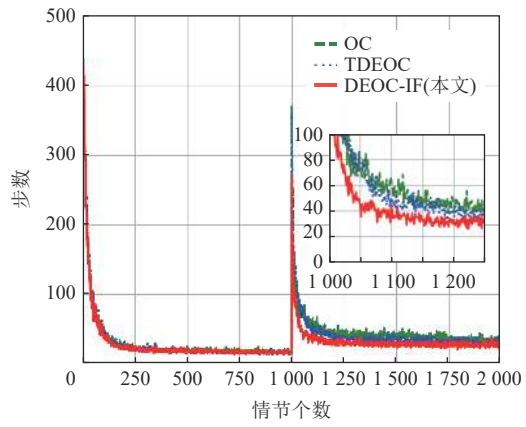


Fig. 3 Learning curve in four-room environment

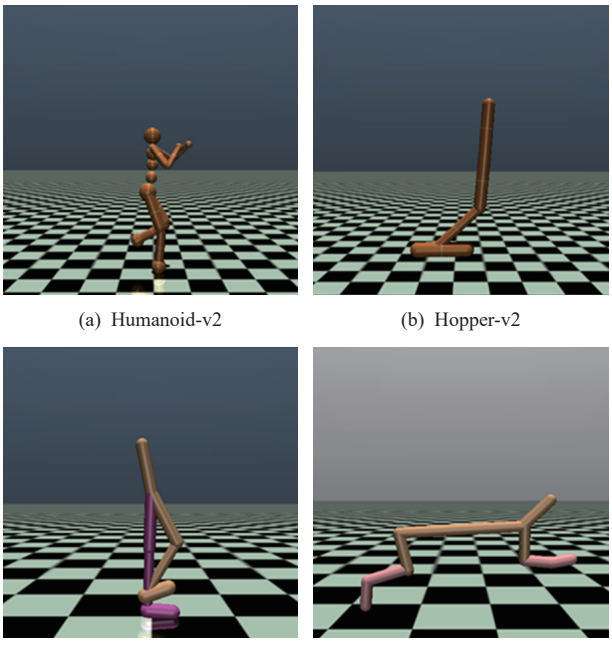
图 3 4 房间环境学习曲线

集合具有多样性，导致 TDEOC 算法的性能优于 OC 算法的性能. 而 DEOC-IF 算法在 TDEOC 算法的基础上引入了兴趣函数，保证学习到的不同 Option 内部策略关注房间的不同区域，例如直行处和拐弯处. 而且，DEOC-IF 算法的上层策略在选择下层内部策略时，不仅考虑兴趣函数值的大小，同时考虑下层内部策略的随机程度，因此具有更好的实验效果. DEOC-IF 算法相比于 TDEOC 算法和 OC 算法，实验性能提升了 20 步左右.

3.2 Mujoco 环境实验

3.2.1 实验环境和设置

OpenAI Gym[28] 是强化学习中常用的实验仿真平台，其中包含很多常用的环境测试接口，例如 Box2D，Atari 等游戏环境以及 Mujoco 物理引擎等. 为了有效评估 DEOC-IF 算法和其他对比算法在连续控制任务中的性能差异，本文选取 Mujoco 物理引擎中的连续控制任务 Humanoid-v2, Hopper-v2, Walker2d-v2, HalfCheetah-v2 作为实验环境，如图 4 所示.



(a) Humanoid-v2 (b) Hopper-v2

(c) Walker2d-v2 (d) HalfCheetah-v2

Fig. 4 Mujoco experimental environments

图 4 Mujoco 实验环境

4 个实验环境的状态空间及动作空间的具体介绍如表 2 所示. 4 个实验环境的任务目标为：Humanoid-v2 训练一个 3 维双腿机器人向前走. Hopper-v2 训练一个 2 维单腿机器人在不摔倒的前提下尽可能快地向前跳. Walker2d-v2 训练一个 2 维双腿机器人在不摔倒的前提下尽可能快地向前走. HalfCheetah-v2 训练一个豹形机器人在不摔倒的前提下快速奔跑.

Table 2 Introduction to Mujoco Experimental Environments

表 2 Mujoco 实验环境介绍

环境名称	动作空间维度	状态空间维度
Humanoid-v2	17	376
Hopper-v2	3	11
Walker2d-v2	6	17
HalfCheetah-v2	6	17

本文提出的 DEOC-IF 算法在 TDEOC 算法的基础上引入了参数化的兴趣函数，并使用 3 个独立的网络分别对应于终止函数、兴趣函数、上层策略 π_Ω 与 Option 内部策略. 每个网络包含 2 个隐层，隐层神经元个数都为 64. 终止函数和兴趣函数网络都使用 sigmoid 函数作为激活函数，而上层策略使用 softmax 函数作为激活函数. 对于 Mujoco 任务，实验对比曲线都取自 5 个不同随机种子的平均值. 其他超参数如表 3 所示.

Table 3 Hyper-Parameters of Mujoco Experimental Environment

表 3 Mujoco 实验环境超参数

超参数	Humanoid-v2	Hopper-v2	Walker2d-v2	HalfCheetah-v2
τ	0.4	0.2	0.2	0.7
c	0.01	0.01	0.01	0.01
α_θ	5E–7	5E–7	5E–7	5E–7
α_θ	3E–4	1E–4	1E–4	1E–4
α_z	1E–4	1E–4	1E–4	5E–4

3.2.2 实验结果分析

在本实验中，把 DEOC-IF 算法与 TDEOC 算法、OC 算法以及 PPO 算法进行性能对比，其学习曲线如图 5 所示. DEOC-IF 算法相较于其他算法的优势是充分考虑了每个 Option 内部策略在每个状态的特殊性，同时保证了在策略学习过程中具有更强的探索能力和稳定性，在所有对比实验中 DEOC-IF 算法相比于 TDEOC, OC 和 PPO 算法均有不同程度的性能提升.

从图 5 中可以看出，OC 算法很快收敛到局部最优情况，TDEOC 算法性能持续提升. 这是因为 OC 算法探索能力低，导致任务相关的 Option 个数减少. 在 1 个 Option 终止之后，由于噪声等因素上层策略选择了次优的 Option. 而 TDEOC 算法引入伪奖励，并提出了新的终止函数，从而能够阻止 1 个 Option 解决整个任务的情况. DEOC-IF 算法是对 TDEOC 算法的改

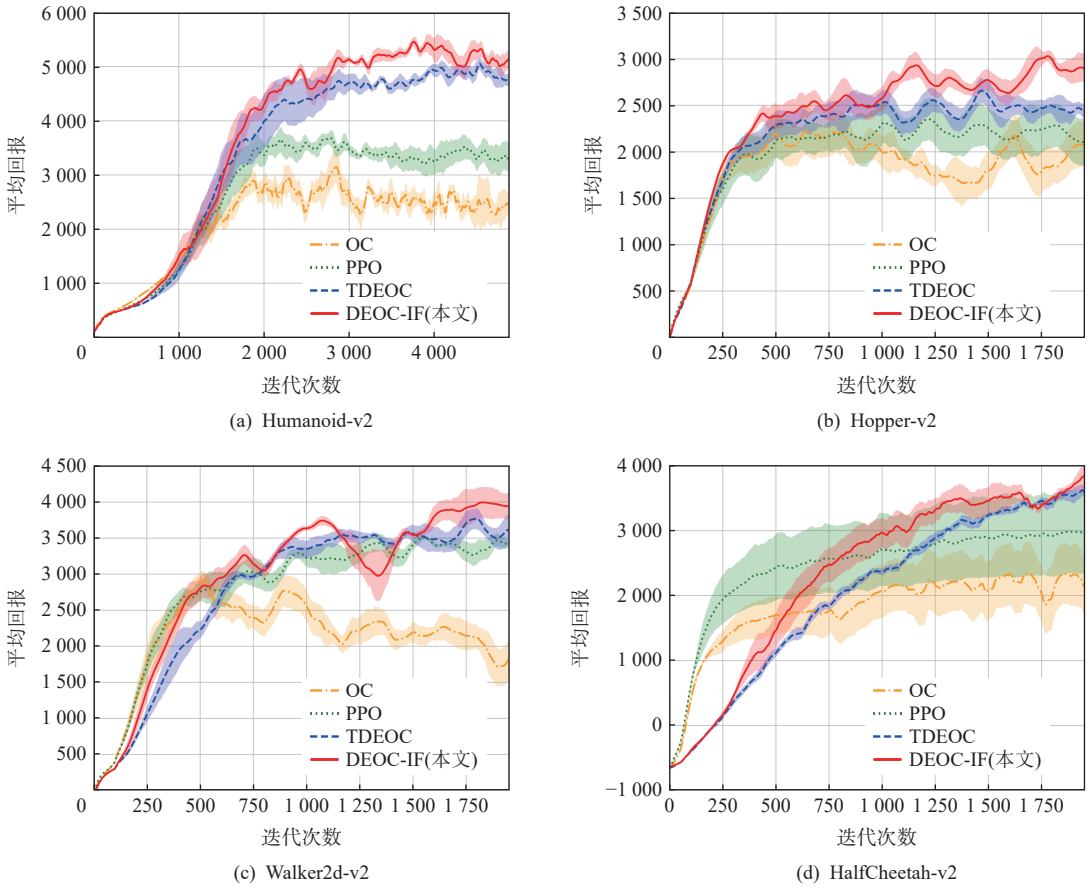


Fig. 5 Learning curve of algorithms in Mujoco environments

图 5 Mujoco 环境中算法学习曲线

进，引入了兴趣函数且充分考虑了下层内部策略的探索能力，从而能够加快学习速度以及提高实验性能. 在 Walker2d-v2 环境中，DEOC-IF 算法出现了波动，这是因为该环境的任务是训练 2 维双足智能体尽可能快地向前走，与其他环境相比环境更复杂.

3.3 迁移实验

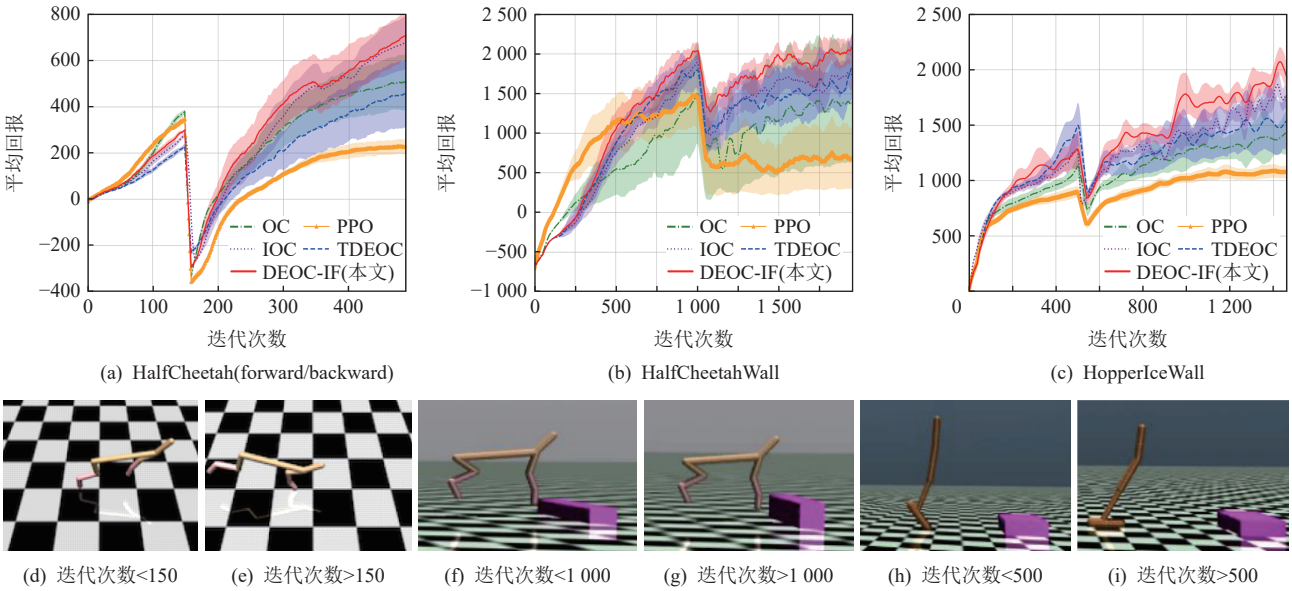
Option 内部策略学习的一个关键优势是学习到的技能可以在类似的任务中被重复使用. 在 3.1 节，评估了 DEOC-IF 算法在离散环境中的知识迁移能力. 本节主要评估 DEOC-IF 算法在连续环境中的知识迁移能力，分别在变种 HalfCheetah，HalfCheetahWall，HopperIceWall 任务上与 TDEOC，IOC，PPO，OC 等算法进行对比. 实验环境和对比曲线如图 6 所示. 在迁移环境中，DEOC-IF 算法的性能明显优于另外 4 个算法. 这表明 DEOC-IF 算法学习到的 Option 集合具有更高的可重复利用性.

1) HalfCheetah(forward/backward)

在迁移实验之前，环境的设置与 OpenAI Gym 中的 HalfCheetah 一样，即智能体受到奖励驱动，尽可能快地向前跑，如图 6(d)所示. 在 150 个迭代(大约 30 万步数)之后，调整奖励函数，驱动智能体尽可能快地向后跑[29]，如图 6(e)所示. 从图 6(a)中可以看出，迁移实验之前，PPO 算法优于其他 4 个算法，150 个迭代后改变了奖励函数，DEOC-IF 算法的性能明显优于另外 4 个算法，表明 DEOC-IF 算法学习到的 Option 集合具有更强的知识迁移能力.

2) HalfCheetahWall

通过这个实验，评估智能体对于路径上障碍物属性改变时的反应能力. 迁移实验之前，HalfCheetah-Wall 环境[30] 中障碍物的高度为 0.12 个单位，如图 6(f)所示. 在 1 000 个迭代(大约 200 万步数)之后，障碍物的高度被调整到 0.2 个单位，如图 6(g)所示. 从图 6(b)可以看出，一开始 PPO 算法的实验性能是最高的，500 个迭代之后，DEOC-IF 算法和 TDEOC 算法的性能都比 PPO 算法高，而且 DEOC-IF 算法的性能始终比 TDEOC 算法的性能更优，说明兴趣函数的引入使得上层策略能根据状态选择更恰当的 Option 内部策略. DEOC-IF 算法的阴影部分最少，说明 DEOC-IF 算法具有更好的稳定性. 在改变障碍物的高度之后，DEOC-IF 算法中的上层策略在选择下层内部策略时



(a) HalfCheetah(forward/backward) (b) HalfCheetahWall (c) HopperIceWall

(d) 迭代次数<150 (e) 迭代次数>150 (f) 迭代次数<1 000 (g) 迭代次数>1 000 (h) 迭代次数<500 (i) 迭代次数>500

Fig. 6 Learning curves of algorithms and environments in transfer tasks

图 6 迁移任务中算法的学习曲线和环境

充分考虑了下层内部策略在当前状态的适用程度和探索能力. 因此 DEOC-IF 算法的实验性能明显优于其他对比算法.

3) HopperIceWall

此环境与 OpenAI Gym 中 Hopper-v2 环境不同之处是在智能体前方 2.8 个单位处放置了一个障碍物, 长宽高分别为 0.25, 0.4, 0.12 个单位[30], 如图 6(h)所示. 在 500 个迭代(大约 100 万步数)之后, 把智能体和障碍物之间的距离由 2.8 增加至 3.3, 如图 6(i)所示. 实验效果如图 6(c)所示, 起初, DEOC-IF 算法和 TDEOC 算法几乎有相同的学习速度, 在改变环境之后, DEOC-IF 算法具有更好的学习性能.

3.4 OneRoom 实验

本文仍然使用了 MiniWorld 框架[31] 中更加复杂的 3D 视觉导航环境——OneRoom 任务. 在 OneRoom 任务中, 智能体需要在一个封闭的房间内到达一个随机产生的盒子的位置. 这需要智能体环顾整个房间, 发现盒子并能到达盒子位置, 如图 7 所示.

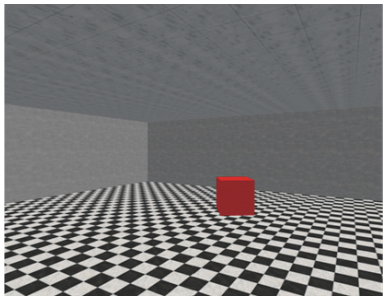


Fig. 7 OneRoom environment

图 7 OneRoom 环境

OneRoom 房间的观测空间是一个 3 通道 RGB 图片, 行为空间包含 8 个离散行为. 每个情节开始时, 盒子的位置是随机的. 如果智能体到达盒子位置或者 180 个时间步被执行完时, 情节终止. 实验使用深度卷积神经网络作为函数逼近器, CNN 结构与 DQN 保持一致. 兴趣函数和每个 Option 的终止函数使用 sigmoid 函数作为激活函数, 其他网络架构与 TDEOC 算法一致. 基础算法是 PPOC 算法, 实验性能取自 5 个独立种子的平均值, 如图 8 所示.

从图 8 中可以看出, OC 算法的性能是最低的. 引入带正则项的兴趣函数之后, DEOC-IF 算法的性能收敛速度比 TDEOC 算法快, 比 PPO 算法的性能更优. 实验表明, 引入兴趣函数使得上层策略具有更好的方向感, 能够选择出更好的 Option, 指导智能体快速到达盒子位置.

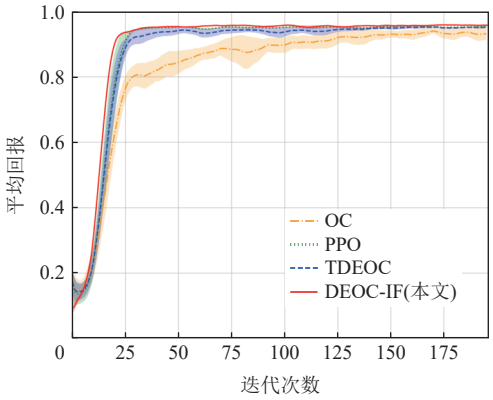


Fig. 8 Learning curve of algorithms in OneRoom environment

图 8 OneRoom 环境中算法学习曲线

3.5 消融实验

为了验证正则项的引入可以提升智能体对环境的探索能力，从而提升实验性能，本节在 Hopper-v2 环境中把带正则项的 DEOC-IF 算法和不带正则项的 DEOC-IF 算法进行了对比实验. 实验结果表明当正则项系数很小时，可以提升实验性能，如图 9 所示.

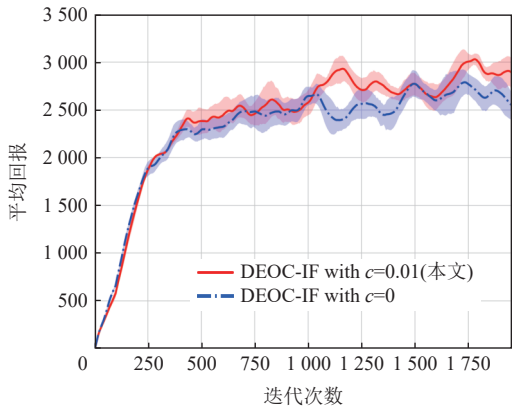


Fig. 9 Ablation experiment

图 9 消融实验

3.6 算法运行时间与符号表

DEOC-IF 算法与其他对比算法的运行时间如表 4 所示，文中所用的主要数学符号如表 5 所示. 从表 4 中可以看出 TDEOC 算法因为使用伪奖励保证内部策略的多样性，所以策略的训练速度缓慢. DEOC-IF 算法因为引入兴趣函数可以约束上层策略选择下层策略时的开销，因此可以加快学习速度.

Table 4 Running Time of Algorithm in Mujoco Environments

表 4 Mujoco 环境中算法运行时间 h

环境	OC	PPO	TDEOC	DEOC-IF(本文)
Hopper-v2	8.7	6.9	10.2	9.5
Walker2d-v2	13.6	10.3	17.2	15.1
Humanoid-v2	19.1	16.2	23.7	21.3
HalfCheetah-v2	8.3	7.1	9.4	8.9

Table 5 The Table of Mathematical Symbols

表 5 数学符号表

符号	符号含义	符号	符号含义
S	状态集合	Ω	Option 集合
A	动作集合	o	Option 下层策略
γ	奖励折扣系数	θ	下层策略参数
r	奖励	U	到达值函数
P	状态转移概率	μ	轨迹折扣权重
π	行为策略	R_{bonus}	伪奖励函数
V	状态值函数	H	香农熵
Q	动作值函数	τ	伪奖励超参数
I_o	Option 初始状态集合	c	正则项系数
β_o	Option 终止函数	z	兴趣函数参数

4 结 论

本文提出了一种结合 TDEOC 算法和兴趣函数的 DEOC-IF 算法. 该算法不再假设每个 Option 的初始状态集为整个状态空间，而是把 Option 框架中的初始集推广为一种兴趣函数，可以通过策略梯度更新参数的方式进行优化. DEOC-IF 算法既保证了 Option 集合的多样性，又限制了 Option 框架中初始集的大小，降低了上层策略选择 Option 内部策略时的性能开销，从而提高算法学习速度. 通过实验表明，兴趣函数类似于一种注意力机制，可以实现更快的学习速度，并且获得更好的稳定性以及策略迁移能力. 本文通过离散任务、连续控制任务、3D 导航任务以及迁移实验验证了 DEOC-IF 算法具有更优异的效果.

本文提出的兴趣函数的参数和终止函数的参数是完全独立的，但它们共同决定了整个算法的好坏程度. 因此，下一步的研究重点是探究兴趣函数和终止函数的关系，在保证 Option 集合多样性的同时，进一步提高算法的性能.

作者贡献声明：栗军伟提出论文思路，负责实验和撰写论文；刘全负责论文审核；黄志刚负责全文修订及部分实验图的绘制；徐亚鹏负责部分实验的完善.

参 考 文 献

[1] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[M]. Cambridge, MA: MIT, 2018

[2] Liu Quan, Zhai Jianwei, Zhang Zongzhang, et al. A survey on deep reinforcement learning[J]. Chinese Journal of Computers, 2018, 41(1): 1−27 (in Chinese)
(刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1−27)

[3] Liu Jianwei, Gao Feng, Luo Xionglin. Survey of deep reinforcement learning based on value function and policy gradient[J]. Chinese Journal of Computers, 2019, 42(6): 1406−1438 (in Chinese)
(刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述[J]. 计算机学报, 2019, 42(6): 1406−1438)

[4] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529−533

[5] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep

reinforcement learning[J]. arXiv preprint, arXiv: 1509.02971, 2015

[6] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods[C]//Proc of the 35th Int Conf on Machine Learning. New York: ACM, 2018: 1582−1591

[7] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//Proc of the 32nd Int Conf on Machine Learning. New York: ACM, 2015: 1889−1897

[8] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[C]//Proc of the 35th Int Conf on Machine Learning. New York: ACM, 2018: 1861−1870

[9] Lai Jun, Wei Jingyi, Chen Xiliang. Overview of hierarchical reinforcement learning[J]. Computer Engineering and Applications, 2021, 57(3): 72−79 (in Chinese)
(赖俊, 魏竞毅, 陈希亮. 分层强化学习综述[J]. 计算机工程与应用, 2021, 57(3): 72−79)

[10] Sutton R S, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1/2): 181−211

[11] Kulkarni T D, Narasimhan K, Saeedi A, et al. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation[C]//Proc of the 29th Advances in Neural Information Processing Systems. Cambridge, MA: MIT, 2016: 3675−3683

[12] Zhao Dongyang, Zhang Liang, Zhang Bo, et al. Mahrl: Multi-goals abstraction based deep hierarchical reinforcement learning for recommendations[C]//Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 871−880

[13] Duan Jingliang, Li Shengbo, Guan Yang, et al. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data[J]. IET Intelligent Transport Systems, 2020, 14(5): 297−305

[14] Liu Jianfeng, Pan Feiyang, Luo Ling. Gochat: Goal-oriented chatbots with hierarchical reinforcement learning[C]//Proc of the 43rd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2020: 1793−1796

[15] Levy A, Konidaris G, Platt R, et al. Learning multi-level hierarchies with hindsight[J]. arXiv preprint, arXiv: 1712.00948, 2017

[16] Bacon P L, Harb J, Precup D. The option-critic architecture[C]//Proc of the 31st AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2017: 1726−1734

[17] Harb J, Bacon P L, Klissarov M, et al. When waiting is not an option: Learning options with a deliberation cost[C]//Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2018: 3165−3172

[18] Klissarov M, Bacon P L, Harb J, et al. Learnings options end-to-end for continuous action tasks[J]. arXiv preprint, arXiv: 1712.00004, 2017

[19] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint, arXiv: 1707.06347, 2017

[20] Li Chenghao, Ma Xiaoteng, Zhang Chongjie, et al. Soac: The soft option actor-critic architecture[J]. arXiv preprint, arXiv: 2006.14363, 2020

[21] Kanagawa Y, Kaneko T. Diverse exploration via infomax options[J]. arXiv preprint, arXiv: 2010.02756, 2020

[22] Eysenbach B, Gupta A, Ibarz J, et al. Diversity is all you need: Learning skills without a reward function[J]. arXiv preprint, arXiv: 1802.06070, 2018

[23] Gregor K, Rezende D J, Wierstra D. Variational intrinsic control[J]. arXiv preprint, arXiv: 1611.07507, 2016

[24] Harutyunyan A, Dabney W, Borsa D, et al. The termination critic[C]//Proc of the 22nd Int Conf on Artificial Intelligence and Statistics. New York: PMLR, 2019: 2231−2240

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

[26] Khetarpal K, Klissarov M, Chevalier-Boisvert M, et al. Options of interest: Temporal abstraction with interest functions[C]//Proc of the 34th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2020: 4444−4451

[27] Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Proc of the 14th Advances in Neural Information Processing Systems. Cambridge, MA: MIT, 2000: 1057−1063

[28] Brockman G, Cheung V, Pettersson L, et al. OpenAI gym[J]. arXiv preprint, arXiv: 1606.01540, 2016

[29] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//Proc of the 34th Int Conf on Machine Learning. New York: ICML, 2017: 1126−1135

[30] Henderson P, Chang W D, Shkurti F, et al. Benchmark environments for multitask learning in continuous domains[J]. arXiv preprint, arXiv: 1708.04352, 2017

[31] Chevalier-Boisvert M, Bolun D, Mark T, et al. Gym-miniworld environment for OpenAI Gym[EB/OL]. 2018[2022-11-22]. https://github.com/maximecb/gym-miniworld



Li Junwei, born in 1998. Master candidate. His main research interests include reinforcement learning and hierarchical reinforcement learning.

栗军伟, 1998 年生. 硕士研究生. 主要研究方向为强化学习、分层强化学习.



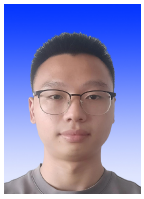
Liu Quan, born in 1969. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include intelligence information processing, automated reasoning, and machine learning.

刘　全, 1969 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为智能信息处理、自动推理、机器学习.



Huang Zhigang, born in 1993. PhD candidate. His main research interests include reinforcement learning, deep reinforcement learning, and hierarchical reinforcement learning.

黄志刚, 1993年生. 博士研究生. 主要研究方向为强化学习、深度强化学习、分层强化学习.



Xu Yapeng, born in 1996. Master candidate. His main research interests include deep reinforcement learning and hierarchical reinforcement learning.

徐亚鹏, 1996年生. 硕士研究生. 主要研究方向为深度强化学习、分层强化学习.