计算机科学
COMPUTER SCIENCE

基于符号知识的选项发现方法

王麒迪，沈立炜，吴天一

引用本文

王麒迪，沈立炜，吴天一. 基于符号知识的选项发现方法[J]. 计算机科学，2025，52(1)：277-288.

WANG Qidi, SHEN Liwei, WU Tianyi. Option Discovery Method Based on Symbolic Knowledge[J]. Computer Science, 2025, 52(1): 277-288.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240100221

基于符号知识的选项发现方法

王麒迪　沈立炜　吴天一
复旦大学计算机科学技术学院　上海 200438
(21210240038@m.fudan.edu.cn)

摘　要　基于选项(Option)的层次化策略学习是分层强化学习领域的一种主要实现方式。其中,选项表示特定动作的时序抽象,一组选项以多层次组合的方式可解决复杂的强化学习任务。针对选项发现这一目标,已有的研究工作使用监督或无监督方式从非结构化演示轨迹中自动发现有意义的选项。然而,基于监督的选项发现过程需要人为分解任务问题并定义选项策略,带来了大量的额外负担;无监督方式发现的选项则难以包含丰富语义,限制了后续选项的重用。为此,提出一种基于符号知识的选项发现方法,只需对环境符号建模,所得知识可指导环境中多种任务的选项发现,并为发现的选项赋予符号语义,从而在新任务执行时被重复使用。将选项发现过程分解为轨迹切割和行为克隆两阶段步骤:轨迹切割旨在从演示轨迹提取具备语义的轨迹片段,为此训练一个面向演示轨迹的切割模型,引入符号知识定义强化学习奖励评价切割的准确性;行为克隆根据切割得到的数据监督训练选项,旨在使选项模仿轨迹行为。使用所提方法在多个包括离散和连续空间的领域环境中分别进行了选项发现和选项重用实验。选项发现中轨迹切割部分的实验结果显示,所提方法在离散和连续空间环境中的切割准确率均高出基线方法数个百分点,并在复杂环境任务的切割中提高到20%。另外,选项重用实验的结果证明,相较于基线方法,赋予符号语义增强的选项在新任务重用上拥有更快的训练速度,并在基线方法无法完成的复杂任务中仍然得到良好收敛。

关键词:分层强化学习;演示学习;选项发现;马尔可夫决策过程
中图分类号　TP311

Option Discovery Method Based on Symbolic Knowledge

WANG Qidi,SHEN Liwei and WU Tianyi
School of Computer Science,Fudan University,Shanghai 200438,China

Abstract　Hierarchical strategy learning based on options is a prominent approach in the field of hierarchical reinforcement learning. Options represent temporal abstractions of specific actions,and a set of options can be combined in a hierarchical manner to tackle complex reinforcement learning tasks. For the goal of option discovery,existing research has focused on the discovery of meaningful options using supervised or unsupervised methods from unstructured demonstration trajectories. However,supervised option discovery requires manual task decomposition and option policy definition,leading to a lot of additional burden. On the other hand,options discovered through unsupervised methods often lack rich semantics,limiting the subsequent reuse of options. Therefore,this paper proposes a symbol-knowledge-based option discovery method that only requires modeling the symbolic knowledge of the environment. The acquired knowledge can guide option discovery for various tasks in the environment and assign symbolic semantics to the discovered options,enabling their reuse in new task executions. This method decomposes the option discovery process into two stages:trajectory segmentation and behavior cloning. Trajectory segmentation aims to extract semantically meaningful trajectory segments from demonstration trajectories. To achieve this,a segmentation model is trained specifically for demonstration trajectories,incorporating symbolic knowledge to define the accuracy of segmentation in reinforcement learning reward evaluation. Behavior cloning,on the other hand,supervises the training of options based on the segmented data,aiming to make the options mimic trajectory behaviors. The proposed method is evaluated in multiple domain environments,including both discrete and continuous spaces,for option discovery and option reuse experiments. In the option discovery experiments,the results of trajectory segmentation show that the proposed method achieves higher segmentation accuracy compared to the baseline method,with an improvement of several percentage points in both discrete and continuous space environments. Moreover,in complex environment tasks,the segmentation accuracy is further improved by 20%. Additionally,the results of the option reuse experiments demonstrate that options enriched with symbolic semantics exhibit faster training speed in adapting to new

到稿日期:2024-01-30　返修日期:2024-05-13
基金项目:上海市重大项目(2021SHZDZX0103)
This work was supported by the Shanghai Major Project(2021SHZDZX0103).
通信作者:沈立炜(shenliwei@fudan.edu.cn)

tasks compared to the baseline method. Furthermore, these symbolic semantics enhanced options show good convergence even in complex tasks that the baseline method fails to accomplish.

Keywords Hierarchical reinforcement learning, Demonstration learning, Option discovery, Markov decision process

1 引言

作为强化学习领域的一个重要分支，分层强化学习(Hierarchical Reinforcement Learning，HRL)旨在解决复杂任务中智能体面临的高维状态空间和大动作空间的问题。通过缩小子任务动作空间和使用层级学习策略的方式，HRL 有效地解决了长程任务问题，并在自动驾驶领域[1]、机器人连续操纵任务[2]等复杂的场景中实现突破。在 HRL 领域的研究中，基于选项(Option)[3]的层次化策略学习是一种主要的实现方式，该概念由 Sutton 等[4]提出。方法的核心是将策略分解为更小的子策略，这些子策略被称为选项。选项是一组动作的时序抽象，它由一系列基本动作组成，并能在一段时间内持续执行。依靠上层的元控制器调度，一组选项以多层次组合的方式解决复杂的强化学习任务。随着选项概念的引入，合理地设定选项成为了一个值得研究的问题。在早期，研究者通常使用人工定义的方式，这在简单的环境中可得到富有语义的选项，但当应用到更复杂多变的环境时，要得到富有语义的选项变得异常困难。与之相比，自动化设定选项是一种更好的方式，智能体可根据环境特征信息灵活地定义选项。

在自动化设定选项的众多方法中，选项发现是一种通过自主学习来发现具有语义的选项策略的方法。它旨在从非结构化的演示中发现选项，以避免人为定义选项的困难。已发现的选项可在同类型任务的执行中得到重用，从而提升智能体在任务中的学习效率。这一选项通常包括一个定义它何时开始和是否应当结束的函数，以及该选项实际的执行策略。

在选项发现中，如何从演示数据中将任务合理地分解为若干个选项是一个需要解决的问题。已有的研究可以分为两个方向。第一种采用监督学习的方式，通过提供尽可能完备的信息训练各种选项。它们或使用手工的方式分解演示轨迹[5]，或为特定的任务流程建模[6-7]。这种方式尽管可以训练出优秀的、语义丰富的选项策略，但在训练时所需的花费太过昂贵，难以有效地拓展到更广泛的任务类型和环境中。另一种则是以无监督的方式发现选项。该方法希望不提供额外信息，仅通过发现演示中蕴含的特征信息进行分解。研究者通常借助潜在变量来提取演示中的信息[8-9]，并以潜在变量表示学习到的选项策略。无监督的方式尽管在训练上只需要极少的花费，但得到的选项策略效果过于依赖鲜明的环境特征，这在复杂的环境中通常难以取得好的效果。由于使用的潜在变量(Latent)不具备良好的语义，因此发现的选项在新任务的重用上遇到困难。

针对上述问题，本文提出一种基于符号知识的选项发现方法，通过人为对环境符号建模，所得知识可指导环境中多种任务的选项发现，并为发现的选项赋予符号语义，从而在新任务执行时被重复使用。该方法在提供适当信息以便进行选项的训练和重用的同时，避免了为单一任务进行建模所带来的重复性工作。在技术方面，该方法将选项发现过程分解为轨迹切割和行为克隆两阶段步骤。轨迹切割旨在从演示轨迹中提取具备语义的轨迹片段。受到 Chen 等[10]应用强化学习方式解决程序合成问题的启发，本文所提方法将轨迹切割任务建模为马尔可夫决策模型，将一条演示轨迹的当前状态视为环境的状态表示，对轨迹执行的切割操作则是对这个环境的动作影响，以强化学习的方式训练一个切割模型。符号知识在轨迹切割中参与了强化学习过程的奖励塑造，为切割的好坏提供价值评判，指导切割策略参数的更新。行为克隆则根据切割得到的数据监督训练选项，使选项最大化模仿轨迹行为。方法将符号知识作用于定义选项的开始条件，并根据轨迹数据以监督学习的方式训练选项的执行策略和终止条件。符号形式的开始条件在选项的重用中提供了有效指导，在智能体选择动作时过滤掉不满足开始条件的选项，缩减了动作的探索空间。

本文最终在办公室[11]、建造[12]和导航[13]3 种实验环境中评测了所提方法，并将其与当前先进的基线方法 Compile[8]和 Taco[13]进行比较。办公室环境和建造环境分别是包含环境交互以及诸多资源依赖的相对简单和复杂的网格环境，而导航环境则是在二维空间中，以导航任务为主体的连续状态环境。整体实验分为选项发现和选项重用，前者评价基于符号知识的方法在选项发现任务上的效果，后者验证发现的选项在新任务的重用能力。实验表明，在导航环境和办公室环境下，基于符号知识的选项发现方法对演示轨迹的切割准确率均高出两种基线方法数个百分点，并在更复杂的建造环境任务中提高到 20%以上。在选项重用实验中，实验结果充分证明，受到符号语义增强的选项能够在新任务重用时彰显出明显优势，可以仅在 3000 次的步数上限设置下完成基线方法需要 10000 步才能完成的任务，并可以在基线方法无法完成的任务中保持良好的收敛速度。

2 相关工作

与本文相关的工作可分为演示学习、选项发现和符号表示结合深度学习 3 个领域。

演示学习(Learning From Demonstration)是一个广泛的领域，重点是基于已训练好的模型或人类专家产生的演示轨迹，学习解决对应的任务[14]。有多种方式可以实现这一点，如对演示的行为克隆[15]，或将演示用参数化的模型拟合[16]。模块化演示学习旨在从复杂的演示中提取可重用的不同层级的策略原语，以提高数据使用效率，并便于策略重用。Schaal 提出了运动源语这一概念[6]，并提供了一个比演示任务低一级的中级策略，运动源语将训练分层，这成为了模块化演示学习领域的一个开端。Niekum 等[17-18]在此基础上解决了运动源语的发现问题，采用贝叶斯非参数方法寻找演示轨迹中的重复结构；CST[7]则是将中级策略定义为能力(Skill)，用变更点检测方法将轨迹分割成不同的能力合并到能力树中，通过对任务问题特定状态建模完成能力的发现。以上的方法考虑

到了轨迹的分割以及源语的发现，但它们过于依赖演示轨迹的特征，使得在复杂演示数据下的效果受到显著影响，并且对单条演示任务建模会降低模型的拓展能力。本文的工作则考虑对多种不同任务所在的环境符号建模，以符号知识温和地指导模型分割演示轨迹并学习策略，且可以在较为复杂的环境信息中保持较好的效果。

最近有关模块化演示学习的工作更多地与分层强化学习问题相结合，被称为强化学习的选项发现（Option Discovery）[19]。由于传统强化学习存在的维度诅咒和稀疏奖励等问题，许多研究者考虑使用分层概念来解决这一类问题。Barto 等基于此提出了分层强化学习[20]，试图通过划分层级、缩小搜索空间的方式简化稀疏奖励任务的学习。在分层强化学习的多种方法实践中[21-22]，Sutton 等提出的选项（Option）这一时间抽象概念得到了普遍的认同[23]。选项表示一个时间连续的任务执行策略，由 3 部分组成：开始集合 I、终止条件 B 以及执行策略 π。将选项作为中级策略，可以将强化学习任务分层，上层训练选择选项的策略，下层训练选项内部的执行子策略。Krishnan 等在此基础上提出了一个深度选项发现框架 DDO[24]，以策略梯度的方式递归地发现参数化选项，并使用发现的选项扩增新任务智能体的动作空间来加速学习；他们后续在此基础上使用 DDCO[25] 解除了 DDO 必须要手动设置选项数量这一限制。Shiarlis 等提出了时间任务对齐方法 Taco[13]，它是一种共同学习演示分割和选项子策略的方法。该工作考虑将弱监督草图与演示轨迹在时间上对齐，从而依据草图信息分解演示轨迹。相比之前的工作，他们仅为演示数据提供弱监督草图信息，而无须对选项参数或任务问题做过多的假设和定义。本文工作和 Taco 的相似之处在于，都使用弱监督符号或草图取代元控制器的功能，但符号知识描述整体领域环境的信息，因而可以在同一环境下的不同任务中共享，且符号知识相比草图拥有更多的语义信息，这可以弥补草图在状态较为复杂的环境中表现不佳的缺陷。对于已发现选项的重用，这些符号知识也增强了学习到的选项的语义，便于后续任务的探索学习。

除此之外，也存在另一部分工作考虑引入潜在变量来帮助发现选项。潜在变量用于演示特征提取，存储已发现选项中的语义信息。Compile[8] 采用无监督的方式，通过变分自编码器（Variational Auto-Encoder，VAE）生成模型学习轨迹状态的特征信息，并获得潜在变量表示的选项。Shankar 等[26] 则使用因式分解的变分推理方式去推断选项，它去除了需要提前指定选项数量的限制。与潜在变量生成方法相比，本文提出的方法可以在更复杂的环境中有效学习选项，并且生成的选项具备符号知识语义，相较于潜在变量，更易被理解，因而拥有更好的重用能力。

符号表示结合深度学习也存在着不少的研究。Xie 等用符号时间知识来指导深度模型的训练[27]，将线性时序逻辑（Linear Temporal Logic，LTL）与模型预测结果一同嵌入到图（DFA）中，训练模型以满足逻辑知识给定的约束。Yang 等[28] 则使用符号规划指导分层强化学习上层策略，根据任务信息生成符号计划。符号计划可在上层元控制器选择选项时提供目标导向，帮助元控制器更好地学习收敛。本文提出的方法借鉴了 Yang 等利用符号知识指导强化学习的思路，将其用于离线强化学习的选项发现领域中，在符号知识的指导下，使用已有非结构化的演示数据有效地发现选项子策略并使其在后续任务中得到重用。

3 预备知识

本章根据研究的问题，介绍了符号知识和分层强化学习领域的有关预备知识，分为符号规划、强化学习、行为克隆以及选项发现 4 个部分。

3.1 符号规划

符号知识是一种高层的抽象的符号状态表示；符号规划则是在一个领域空间。对给定的问题规划出一个符号解决方案。当前存在许多描述性的符号语言，本文使用的是规划领域定义语言（Planning Domain Definition Language，PDDL）。一个规划领域空间 D 定义了这个空间内物体可能的状态以及智能体可以执行的动作，D 由一组命题 $P=\{p_1,p_2,\cdots,p_n\}$、一组符号动作 $C=\{c_1,c_2,\cdots,c_m\}$ 和一组符号状态 $H=\{h_1,h_2,\cdots,h_k\}$ 组成。命题是一个由人定义的，抽象的布尔状态变量，表示当前环境是否满足这一变量所描述的状态。如命题 $\mathrm{on}(cup_1,table_1)$ 表示杯子 1 在桌子 1 上，此时这一命题为真；当不存在这一命题，或存在 $\mathrm{not}(\mathrm{on}(cup_1,table_1))$ 时，则表示这一命题为假。一个符号状态 h 是由一组已赋值的命题组成的，这些命题共同定义了整个状态空间。

一个符号动作 c_i 可以表示为元组 $c_i=(args,precond_i,effect_i^+,effect_i^-)$。其中，$args$ 指符号动作的参数，通常是领域空间中的对象；$precond_i\subseteq P$ 是动作的前置条件，表示执行动作需要满足的条件，即只有当前状态可以使得 $precond_i$ 中的所有命题都为真时，才可以执行该符号动作；$effect_i^+\subseteq P$ 和 $effect_i^-\subseteq P$ 则分别是动作的正向后置条件和负向后置条件，$effect_i^+$ 表示该符号动作执行后改变值为真的命题集合，$effect_i^-$ 表示该符号动作执行后改变值为假的命题集合。

规划问题是在某一领域空间 D 下，定义了一个开始状态 $h_0\subseteq P$ 和目标状态 $h_g\subseteq P$，希望找到从状态 h_0 开始，最终达到目标状态 h_g 所需的有序符号动作序列，这一序列被称为计划 $Plan$。计划通常由一组串行有序的、包含参数的符号动作 $\{c_1\rightarrow c_2\rightarrow\cdots\rightarrow c_n\}$ 组成。一旦拿到该规划问题的计划，只需从初始状态开始，按顺序执行计划中的符号动作且保证每一个符号动作均执行成功，就可以达到预期的目标状态。规划问题如今一般依靠符号规划器求解。符号规划器通过先进的偏序搜索、剪枝等技术，可以高效地寻找到可用计划。当前常用的规划器有 POPF，FD（Fast Downward）等，本文方法采用 FD 作为默认的符号规划器。

3.2 强化学习

强化学习用于解决马尔可夫决策过程（Markov Decision Process，MDP）问题，其通常被定义为一个元组 $(S,A,Tr_{ss'}^a,r,\gamma)$。其中，$S$ 表示状态空间；A 表示动作空间；$Tr_{ss'}^a$ 是一个状态转移函数，表示在状态 $s\in S$ 时，执行动作 $a\in A$，到达状态 $s'\in S$ 的转移概率；$r(s,a):S\times A\rightarrow R$ 是奖励函数，R 表示实数集，奖励是环境在智能体每次执行动作改变状态之后发出的反馈，用于评价智能体当前动作的好坏；$\gamma:[0,1)$ 是折扣因子，

用于奖励的折损传递。强化学习通常使用一个策略 $\pi: S \rightarrow A$ 来解决 MDP 问题，这个策略 π 根据当前的状态选择最合适的动作。强化学习的训练希望通过不断地探索环境并收集奖励反馈来优化策略，使得执行任务获得的环境奖励最大化。

3.3 行为克隆

行为克隆是示教学习的一种实现方法，它将示教学习建模为一个监督学习问题。该问题的目标是训练一个策略 π，其可最大化克隆轨迹数据集 $F=\{\tau_1, \tau_2, \cdots, \tau_M\}$ 的行为。任意一条轨迹 τ_i 是由一组 T 个时间相关的状态-动作对组成的，$\tau_i=\{(s_1, a_1), (s_2, a_2), \cdots, (s_T, a_T)\}$，其中 $s \in S, a \in A$。将策略 π 用参数 θ 建模，则 $\pi_\theta(a \mid s)$ 表示在状态 s 时策略 π_θ 选择执行动作 a 的概率。

行为克隆整体的优化目标可由式(1)表示：

$$\theta^* = \arg\max_{\theta} E_{\tau \in F}\left[\sum_{t=1}^{T} \log \pi_\theta(a_t \mid s_t)\right] \tag{1}$$

其中，E 表示以演示轨迹作为监督样本、策略 π_θ 根据状态选择执行动作的期望。

3.4 选项发现

选项发现是模块化示教学习和分层强化学习交融产生的问题，该问题是在 Sutton 提出的选项框架的基础上，期望从已有的演示数据中学习到可重用的子策略选项。一个选项 o 一般由 3 个部分组成：一个执行策略 $\pi: S \times A \rightarrow [0,1]$，与强化学习智能体策略意义相似；一个开始集合 $I: I \subseteq S$；终止条件 $\beta: S \rightarrow [0,1]$。只有当环境状态 $s_i \in I$ 时，一个选项 (π, I, β) 才可以被元控制器选择开始执行。选项之后，会根据执行策略 π 对当前环境状态选择一系列动作，直到终止条件 β 被满足，选项终止，并将控制权交回上层元控制器。给定一个选项集合 $OS=\{o_1, o_2, \cdots, o_L\}$，基于完全监督的选项发现，需要首先将数据集 F 中的每个轨迹 τ 分段并归于 L 类 F_L，之后对每一个选项 o_l 使用对应轨迹段 F_l 执行行为克隆。因此，第 l 个选项的执行策略 π_θ 优化目标可以表示为：

$$\theta^*_{l=1,\cdots,L} = \arg\max_{\theta_l} E_{F_l}\left[\sum_{t=1}^{T_F} \log \pi_{\theta_l}(a_t \mid s_t)\right] \tag{2}$$

然而，上述方式需要人为地划分每一条轨迹，并将每一段切割后的子轨迹对应到不同的子策略，这将会耗费非常多的资源。因此，如何根据轨迹数据将轨迹自动切割是一个值得研究的问题。Taco 方法采用了弱监督的方式，引入了草图。草图是一段动作序列，包含动作执行的前后顺序，其中每一个动作对应着一个选项。Taco 将轨迹切割定义为时间对齐任务，目的是找到轨迹中每一个状态动作对 (s_i, a_i) 应当属于哪一个选项。本文借鉴了 Shiarlis 的思路，同样采用符号知识指导的方式辅助切割轨迹。不同的是，本文方法以强化学习的方式训练一个策略，该策略可根据输入的轨迹动作序列，输出对应的切割点集；而且提出的方法引入的符号知识并不同于草图仅针对某单一任务，而是对整个任务环境空间建模，因此可以允许多任务的多种轨迹数据共享符号知识，将轨迹切割问题拓展至同一领域空间的多种任务轨迹切割。

4 使用符号知识指导选项发现问题

本章提出了使用符号知识指导选项发现问题的方法。首先，定义了符号知识的形式，并定义了基于符号知识的选项发现问题；然后，从问题出发，分别从轨迹切割和行为克隆这两个方面介绍本文的方法；最后，介绍了将发现选项用于重用的方法。

4.1 问题定义

本方法采用 PDDL 符号语言作为符号知识的描述方式。针对演示轨迹所在的环境，通过人工构建的方式，为环境建模了对应的领域空间 $D=\{C, P, Abstract\}$。领域空间 D 包括一组符号动作 $C=\{c_1, c_2, \cdots, c_L\}$、一个命题集合 $P=\{p_1, p_2, \cdots, p_k\}$ 和一个抽象函数 $Abstract$。符号动作可以表示为元组 $c_i=(args, precond_i, effect_i^+, effect_i^-)$，其元素分别表示符号动作的参数、前置条件和后置条件。抽象函数 $Abstract: S \rightarrow H$ 负责底层状态空间 S 到高层符号状态空间 H 的映射，一个高层符号状态 h 是由一组带有真假值的命题 p_k 组成的。

本方法定义了一个基于符号知识的选项发现问题，考虑包含多种任务的一组专家演示轨迹集 $F=\{\tau_1, \tau_2, \cdots, \tau_M\}$，其中任意一条时间段为 T 的轨迹 τ_m 由一组状态动作对 (s, a) 组成，即 $\tau_m=\{(s_1, a_1), (s_2, a_2), \cdots, (s_T, a_T)\}$。方法中同样定义了一组需要学习的选项集合 $OS=\{o_1, o_2, \cdots, o_L\}$ 以及一个领域空间 $D=\{C, P, Abstract\}$。因此，基于符号知识的选项发现问题是根据轨迹数据集 F 中的每一条轨迹 τ_m，借助符号知识的引导，学习一组与符号动作对应的选项，其执行语义应当与对应符号动作一致。

4.2 基于符号知识的选项发现

选项发现过程主要可拆解为轨迹切割和行为克隆两个步骤。轨迹切割的目标是将演示轨迹切割成可用于训练单个选项的子轨迹数据；行为克隆则是根据切割得到的子轨迹数据，分别训练各个选项的策略网络。符号知识对选项发现的指导体现在对轨迹切割效果的评判和对行为克隆选项开始条件的设置上。本节将分别从轨迹切割和行为克隆这两个方面介绍方法内容。

4.2.1 基于符号知识的轨迹切割

轨迹切割任务是根据所给的演示轨迹数据集，将其中的每一条轨迹进行切割，提取具备语义的轨迹片段，以期望这些子轨迹可以满足所给约束条件。现有的轨迹切割方法大致分为两个方向：以 Compile 和 Shankar 等[9]为代表的方法采取无监督的方式切割轨迹，他们采用 VAE 的方式，使用潜在变量表达一段子轨迹，并以潜在变量重建原轨迹的特征作为训练目标；而以 Taco 方法为首的则采用弱监督的方式，通过定义草图，以时间任务对齐的方法，将轨迹中的数据与草图实现对应，从而帮助切割轨迹。

与上述方向不同，本文借助符号知识指导轨迹切割任务，通过将轨迹切割任务定义为马尔可夫决策过程，以强化学习的方式训练一个切割策略网络。其中，符号知识以评价切割得到的子轨迹对符号约束满足性的方式定义了环境奖励。本小节后续将详细介绍方法的内容。

基于符号知识的轨迹切割任务的目标是最大化切割得到的子轨迹对符号知识约束的满足性。本文通过符号知识的指导，训练一个神经网络，完成轨迹切割的主要任务。因此，轨迹切割任务形式化定义为：训练一个网络 ρ_θ，根据轨迹数据集 F 中的每一条轨迹 τ_m，得到一组切割点 $B=\{b_1, b_2, \cdots, b_N\}$，

$N \ll T$，使得每一个切割后得到的子轨迹$\tau_m^{b_n,b_{n-1}}$都对应于一个选项o_l，并满足领域空间 D 对该选项符号定义约束的最大似然。

符号知识对切割网络的指导主要体现在对其切割效果的评判能力上。借助领域空间 D 中的抽象函数 $Abstract$ 将一段轨迹τ_m的首个底层状态s_1和最后一个底层状态s_T分别映射到高层符号状态$h_1=Abstract(s_1)$ 和$h_T=Abstract(s_T)$中。之后，将h_1和h_T分别视为轨迹τ_m所对应任务的开始状态和目标状态，从而构建一个符号规划问题 $Problem=(D,h_1,h_T)$，并借助符号规划器 FD 规划得到由一组符号动作顺序编排而成的计划 $Plan=\{c_1\rightarrow c_2\rightarrow\cdots\rightarrow c_N\}$，通过顺序执行计划中的每个符号动作，可以顺利地从开始状态到达目标状态。因而每一段切割点b_n都对应 $Plan$ 中的一个符号动作c_n，可以借助符号动作 c_n评判切割点b_n选择的好坏。

然而，若以监督学习的方式训练切割网络ρ_θ，在借助符号知识计算切割网络ρ_θ的损失 $Loss$ 时，一个符号动作中描述的是该动作执行前后符号状态的变化，其中的约束也指对动作执行前后符号状态的约束，它在轨迹中对应着一段子轨迹，而切割点则只对应轨迹中的一个点，难以直观地通过符号动作c_n的约束衡量切割点b_n的对错。因此，本方法将切割点的对错转变为相邻切割点切割得到的子轨迹的好坏，通过衡量切割后的子轨迹$\tau_m^{b_n}$与对应的符号动作c_n的前置条件$precond_n$和后置条件$effect_n$的满足性，间接地评价切割点选择的好坏。这一指标被称为切割的准确性 r。其中，$\tau_m^{b_n,b_{n-1}}$表示第 m 条轨迹中被切割点b_{n-1}和b_n切割出的子轨迹，每一条子轨迹是由一组状态－动作序列构成的，形式如下：

$$\{(s_{b_{n-1}},a_{b_{n-1}}),(s_{b_{n-1}+1},a_{b_{n-1}+1}),\cdots,(s_{b_n},a_{b_n})\}$$

因此，切割网络ρ_θ的训练可以表示为：

$$\theta^*=\arg\max_\theta \prod_{m=1}^{M}\left[\rho_\theta(b_1,b_2,\cdots,b_N\mid\tau_m)\sum_{n=1}^{N}r(c_n,\tau_m^{b_n,b_{n-1}})\right] \tag{3}$$

由于公式中对切割网络ρ_θ输出的切割点b_n的评价是间接的，监督学习所必需的损失函数难以建立，因此本文尝试将整个切割过程建模为 MDP，使用 r 表示切割动作得到的奖励，并以强化学习的方式训练切割网络。

依据式(3)，切割网络ρ_θ需要根据输入的每一条轨迹 τ 输出一个切割序列$\{b_1,b_2,\cdots,b_n\}$。在以马尔可夫决策过程建模整个切割过程后，切割网络应当迭代地根据当前轨迹状态输出对应的切割点b_n。本方法定义$q_{m,n}$表示第 m 条轨迹τ_m在第 n 次切割后的状态，为了方便后续描述且不失一般性，将$q_{m,n}$简写为q_n，并将子轨迹$\tau_m^{b_n,b_{n-1}}$简写为τ^{b_n}。本方法也定义了 Q 表示轨迹状态q_n的空间，用 B 表示切割动作b_n的空间，则切割网络ρ_θ可以表示为$\rho_\theta:Q\rightarrow B$。为了便于轨迹状态的流转，本方法定义了一个轨迹状态的切割操作 $CUT:(q_n,b_n)\rightarrow q_{n+1}$，切割操作将轨迹 τ 的前b_n段状态动作对去除。例如，一段长度为 10 的轨迹序列$\{(s_1,a_1),(s_2,a_2),\cdots,(s_{10},a_{10})\}$，经过 $b=3$ 的切割后，变为$\{(s_4,a_4),(s_5,a_5),\cdots,(s_{10},a_{10})\}$。

因此，基于符号知识的 MDP 形式的轨迹切割问题可以被看作一个元组$\langle Q,B,C,CUT,r,\gamma\rangle$。本文使用上文的切割准确率 r 作为 MDP 中的奖励函数，接收当前的轨迹状态q_n、切割点b_n和下一轨迹状态q_{n+1}，依据对应的符号动作c_n得到实值奖励，即 $r:Q\times B\times Q\times C\rightarrow R$，其中 R 表示实数空间，γ 表示折扣因子。

由于输入网络的不同轨迹的长度是可变的，且单条轨迹的多次迭代切割之间被认为存在一定的联系，因此选择 LSTM 模型实现切割网络ρ_θ。为了便于奖励的塑造，本文构建了一个从符号状态到符号动作的映射网络$M_\mu:H\rightarrow C$，表示当前这个高层符号状态 h 最可能属于某一个符号动作 c，网络 M 与ρ 参数独立，共同训练学习。

整个算法的框架结构如图 1 所示，框架整体是强化学习的交互机制，将使用符号知识表示的符号领域空间 D 和映射网络 M 一同作为强化学习探索环境，将整个演示轨迹 τ 视为环境当前状态。在每一次迭代中，智能体探测环境状态q_n，并通过策略网络ρ_θ产生与环境交互的切割点b_n。领域空间 D 根据轨迹 τ 生成对应的计划 $Plan$，并借助映射网络M_μ得到对应的符号动作c_i。环境根据切割点切割演示轨迹，使用计划和符号动作对切割得到的子轨迹进行评价，得到环境迭代奖励r_n。最后，智能体依靠(q_n,b_n,q_{n+1},r_n)更新切割网络参数。

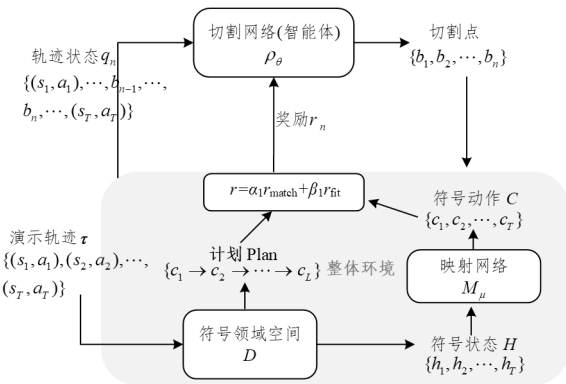


图 1 基于符号知识 MDP 形式的轨迹切割框架

Fig. 1 Trajectory cutting framework of MDP form based on symbolic knowledge

本文方法使用符号知识辅助定义 MDP 过程的奖励函数 r。奖励函数由匹配奖励r_{match}和融洽奖励r_{fit}两部分组成，形式化定义为 $r(q_n,b_n)=\alpha_1 r_{\text{match}}+\beta_1 r_{\text{fit}}$，其中$\alpha_1$和$\beta_1$是超参数，表示对两种奖励的配比。匹配奖励表示为$r_{\text{match}}=Match(M(Abstract(\tau^{b_n})),c_n)$。切割后的子轨迹$\tau^{b_n}$先经过抽象函数映射为高层符号状态$h_m$，再经过网络 M 映射得到符号动作c_m。r_{match}定义为c_m和子轨迹所对应 $Plan$ 中的符号动作c_n的匹配度，具体计算时计算两个符号动作中所有符号状态的匹配个数占总符号状态的比值。融洽奖励r_{fit}则定义为$r_{\text{fit}}=Fit(precond_n,effect_n,Abstract(\tau^{b_n}))$，表示切割后的子轨迹$\tau^{b_n}$对于对应的 Plan 中的符号动作$c_n$的前置条件$precond_n$和后置条件$effect_n$的符合程度。记$\tau_0^{b_n}$和$\tau_{-1}^{b_n}$分别表示切割得到子轨迹的第一个状态和最后一个状态，r_{fit}期望$\tau_0^{b_n}$所对应的高层符号状态 $Abstract(\tau_0^{b_n})$应当满足符号动作c_n的前置条件$precond_n$的约束，$\tau_{-1}^{b_n}$对应的高层符号状态则应满足后置条件$effect_n$的约束。在具体实现上，由于高层符号状态是由一组实例化的命题组成的，而前置与后置条件同样是由一组命题组成的，因此本方法将两者通过 one-hot 编码，形成两个

向量，并采用余弦相似度计算两者的差异，以此作为融洽奖励的值。

根据上文定义，切割网络ρ_θ的奖励塑造依赖于映射网络M_μ，因此接下来将介绍映射网络M_μ的训练过程。映射网络采用监督学习的方式进行训练。训练所需的损失函数L与ρ_θ的奖励函数r结构类似，同样由两部分组成，构建为$L=\alpha_2 L_{\text{match}}+\beta_2 L_{\text{rec}}$，$\alpha_2$和$\beta_2$是超参数。其中，$L_{\text{match}}$与$\rho_\theta$中的奖励$r_{\text{match}}$的定义类似，表示子轨迹$\tau^{b_n}$经过抽象函数*Abstract*和映射网络$M_\mu$得到的符号动作$c_M$与该子轨迹在计划*Plan*中对应的符号动作$c_n$的匹配度；$L_{\text{rec}}$则表示一种重建损失，即一个高层符号状态$h_i$经过$M_\mu$映射为符号动作空间$C$的分布，再根据分布对符号动作空间中每个符号动作$c_i$的前置条件$precond_i$和后置条件$effect_i$进行加权，得到一个重建的高层符号状态$h_{\text{rec}}$，$L_{\text{rec}}$则为$h_i$和$h_{\text{rec}}$的差异，即$L_{\text{rec}}=Reconstruction(h_i,h_{\text{rec}})$。映射网络$M_\mu$接收切割网络$\rho_\theta$切割后的子轨迹，经过*Abstract*函数抽象后得到高层符号状态序列送入网络，以最小损失函数L为目标，监督学习的方式训练网络参数μ。

对于切割网络ρ_θ的训练，为了解决MDP形式的轨迹切割问题，使用近端策略优化方法(Proximal Policy Optimization，PPO)作为强化学习的框架，这是一种actor-critic的策略梯度算法。Actor-critic需要同时训练一个策略网络ρ_θ(同切割网络)和一个价值网络V_φ，两个网络参数的更新满足式(4)和式(5)。

$$\theta_{k+1}=\arg\max_\theta \frac{1}{|F_k|T}\sum_{\sigma\in F_k}\min\left(\frac{\rho_\theta(b_t|q_t)}{\rho_{\theta_k}(b_t|q_t)}Adv^{\rho_{\theta_k}}(q_t,b_t),\ g(\epsilon,Adv^{\rho_{\theta_k}}(q_t,b_t))\right) \tag{4}$$

$$\varphi_{k+1}=\arg\min_\varphi \frac{1}{|F_k|T}\sum_{\sigma\in F_k}\sum_{t=0}^{T}(V_\varphi(q_t)-r_t)^2 \tag{5}$$

其中，F_k表示执行策略$\rho_k=\rho_{\theta_k}$收集到的轨迹σ的集合，$Adv^{\rho_{\theta_k}}$表示对策略网络ρ_θ的优势估计。g函数则依据如下定义：

$$g(\epsilon,Adv)=\begin{cases}(1+\epsilon)Adv, & Adv\geqslant 0\\(1-\epsilon)Adv, & Adv<0\end{cases} \tag{6}$$

由于两个网络的奖励r和损失函数L都依赖彼此的输出结果，因此选择使用交替训练的方式共同训练切割网络ρ_θ和映射网络M_μ。整个训练流程如算法1所示，当训练切割网络ρ_θ时，算法会固定映射网络M_μ的参数，以便获得准确的奖励，直到更新切割网络参数后，才会解除映射网络的固定，反之亦然。调整轮次中训练切割和映射网络的比例，优化训练。在实际训练时，为避免两个网络参数初始化问题而影响训练的速度和最终效果，本文方法通过在训练前对映射网络做少量预先训练的方式优化其初始参数。具体地，由于一条轨迹的第一个和最后一个状态一定分别对应于计划*Plan*的第一个和最后一个符号动作，因此可以使用每段轨迹中包含的这些数据少量预先训练映射网络，以提升其在实际训练时的效果。

算法1 基于符号知识的轨迹切割

输入：轨迹数据集F=$\{\tau_1,\tau_2,\cdots,\tau_M\}$，符号领域空间D=$\{C,P,Abstract\}$，切割网络$\rho_\theta$，映射网络$M_\mu$，总训练轮次time0，切割网络训练轮次time1，映射网络训练轮次time2

输出：切割网络ρ_θ

```
1. foreach i=1,2,…,time0 do
2. foreach j=1,2,…,time1 do
3.     训练切割网络ρθ，固定映射网络参数 μ；
4.     cuttingNetwork.train(F,D,ρθ,Mμ)；
5.   end for
6.   for each j=1,2,…,time 2 do
7.     训练映射网络Mμ，固定映射网络参数 μ；
8.     mapping Network.train(F,D,ρθ,Mμ)；
9.   end for
10. end for
```

4.2.2 基于符号知识的行为克隆

在轨迹切割任务后，轨迹数据集F被切分为了L份，每一份数据F_l都包含了一个选项策略训练所需的子轨迹数据集。因此，行为克隆任务根据第l个选项训练需要的子轨迹数据集F_l，以克隆的方式训练出一个选项。该任务的目标是最大化各选项策略的似然，如式(7)所示。最终训练得到一个选项集$OS=\{o_1,o_2,\cdots,o_L\}$，选项集中每一个选项$o_l$拥有自己的执行策略$\pi_{\eta_l}$。

$$\eta^*=\arg\max_\eta \prod_{l=1}^{L}\left(\sum_{\tau\in F_l}\left[\prod_{i=1}^{len(\tau)}\pi_{\eta_l}(a_i|s_i)\right]\right) \tag{7}$$

首先介绍学习选项的开始集合。现有工作通常只判断当前环境状态是否满足选项的开始集合，而本文则额外考虑了符号层面的状态，将选项的开始集合从符号状态和环境状态两个角度定义。根据上节轨迹切割方法的描述可知，一个选项o_l对应了一个符号动作c_l，方法将该符号动作c_l的前置条件$precond_l$作为选项o_l在符号状态层面的开始状态集合，只要环境满足这一前置条件，则表示可以执行该选项。在环境状态层面，本方法为每一个选项o_l训练一个开始状态判定网络$Init_l$，该网络根据输入的环境状态，输出一个0～1的分布，这一分布表示该环境状态有多大的概率满足开始状态。为训练该网络，对每一个选项的子轨迹数据集进行分类处理。对于每一条子轨迹，将子轨迹τ^{b_n}最靠前的状态视为最满足该选项的开始状态集，因此将子轨迹的前几个状态数据的标记置为1，而将子轨迹最后几个状态数据置为0。剩余的轨迹中间部分，按照从前到后标记值从1到0逐级递减的方式完成标记。之后，使用标记的数据对网络进行监督学习训练，得到用于判定环境开始状态的网络。

然后，根据子轨迹数据集学习每一个选项的执行策略。本方法为每一个选项o_l训练一个策略网络π_{η_l}，该网络根据当前环境状态输出最应执行的动作。将每一条子轨迹中包含的状态-动作序列作为数据集的输入和标记输出，以监督学习的方式训练执行策略π_{η_l}。为了防止训练出现过拟合情况，在监督数据中加入微小正则。

最后，为每一个选项o_l的终止条件训练了一个终止判定网络β_l。该网络接收当前的环境状态，输出一个0～1的分布，表示该状态下是否应当停止选项。在训练数据上，将一条子轨迹最靠后的几组状态-动作对视为距离选项终止最接近的状态，因此将其标记为1。而对于子轨迹的其他状态-动作

序列，则统一以 0 作为标记。在直接采用数据集训练时，由于标记为 1 的数据量明显少于标记为 0 的数据，为防止网络学习效果不佳，方法中选择对标记为 1 的终止训练数据进行增强。对每一条终止训练数据，将其复制为多份，以缩小终止与非终止的比例差异，并通过添加噪声的方式对其进行数据增强。最终，为每一个选项训练得到负责终止判定的网络β_l。

4.3 基于符号知识的选项重用

选项重用的目标是根据已发现和学习的选项，在后续的任务上重复使用选项，以减少任务的探索时间，加快任务的完成速度。在选项重用部分，本文借鉴了 Fox 等[24]对选项的重用技巧，将发现的选项添加到智能体在新任务的动作空间中。对于新的任务，智能体可以选择直接在环境中执行底层动作（如上下左右移动，与环境交互等），也可以选择调用选项执行已训练好的子任务策略，直到该选项执行完成。简单来说，选项重用方法将智能体选择动作的范围从原动作空间＝{ 底层动作 }拓展到增量动作空间＝{ 底层动作，已发现的选项 }。这样的方式看似增大了动作探索的空间，但其实由于选项是一个连续的时间抽象，一个选项的策略通常会输出多个有序的底层动作，因此在整个任务执行的时间周期中，引入选项的重用会显著地减少任务执行的总动作数（此时将一个选项也看作一个动作），从而加速强化学习的探索。

在此基础上，本文方法充分利用了符号知识对选项赋予的语义，并将其用于新任务的选项选择上。智能体在增量动作空间中选择选项时，会根据当前的环境状态，依据每一个选项o_1对应符号动作c_1的前置条件过滤掉不满足的选项，从而将增量动作空间从{ 底层动作，已发现的选项 }削减为{ 底层动作，已发现的满足条件的选项 }，有效地缩小了探索的动作空间，进一步加速了强化学习的探索。

5 实验

为了准确衡量本文方法的效果，需要回答两个问题。

Q1：基于符号知识的选项发现方法发现的选项的准确性如何？

Q2：由符号知识增强语义的选项在新任务的复用能力如何？

为了回答这两个问题，实验部分分别设计了选项发现实验和选项重用实验。在选项发现实验中，需要针对不同任务的演示轨迹数据集训练一个良好的切割网络ρ_θ，并利用带有切割边界标记的测试集测试切割网络的性能；然后，需要利用切割得到的轨迹数据集，以行为克隆方式学习选项，得到各个策略。在重用实验中，使用发现的选项来扩大新任务智能体的动作空间，并从中添加符号知识对选项选择的指导，测试符号增强的选项的重用能力。

5.1 实验设置

为实验设置了两个基线：Compile 方法[8]和 Taco 方法[13]。Compile 是一种无监督的选项发现方法，它以潜在变量的形式描述发现的选项，并使用 VAE 从演示轨迹中对其进行学习；Taco 则是借助动作序列形式的弱监督草图帮助轨迹的切割和选项的发现，它将轨迹切割和选项的行为克隆综合了起来，达到共同的最优。表 1 列出了基于符号知识的方法和基线在训练方式、支持的实验类型方面的区别。其中，基于符号知识的方法和 Taco 方法都支持对多任务的演示数据做切割，因此切割得到的选项可以共同用于重用实验对比；而 Compile 不支持这一点，所以对于相同的演示数据，Compile 会切割得到远多于前两种方式的选项数量（因为它不会合并两个任务中包含的相同选项），因此不对其做重用实验对比。

表 1 基线实验设置

Table 1 Baseline experiment settings

实验方法	训练方式	支持多任务	支持切割实验	支持重用实验
Taco	弱监督	是	是	是
Compile	无监督	否	是	否
基于符号知识的选项发现	强化学习	是	是	是

在实验中，基于符号知识方法的切割网络ρ_θ包含 Actor 网络和 Critic 网络，它们都是一个 2 层的 LSTM 网络；映射网络M_μ则是一个 3 层的 MLP。每一个选项中包含一个策略网络，根据当前状态选择合适的执行动作以及一个终止网络，并根据当前状态判断是否终止选项。此外，所有网络的隐藏层都包含 64 个隐藏单元。

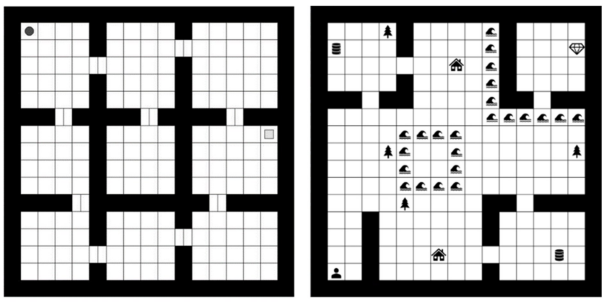
对于 Taco 基线方法，实验中使用了与原作者基本一致的参数和模型，仅修改了演示数据的维度以及测试集的数量。Taco 训练所需的草图，一齐生成演示数据，且演示数据只用于 Taco 方法的训练中。对于 Compile 基线方法，针对本文实验的环境和演示数据，将原作者提供的示例代码进行重写（示例代码仅支持一维的轨迹数据，而实验环境的轨迹数据都为二维），并修改了代码中的环境维度、预定分割数量等参数。其中，分割数量设置为同一段轨迹内，Taco 所需草图中包含的动作数。

5.2 实验环境

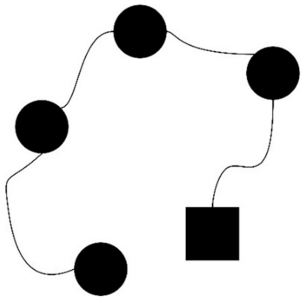
本文将在 3 个环境中评估基于符号知识方法的性能：办公室（Officeworld）环境、建造（Minecraft）环境和连续空间的导航环境（Nav-world）。其中，办公室环境和建造环境是离散状态下复杂程度不同的两个环境，而导航环境则是连续状态空间下的实验环境。这些环境涉及各种有挑战的低级控制：智能体必须学会避开障碍物，并学习如何与环境的各个物体交互。这些环境的奖励也都较为稀疏，通常只有完全完成了目标任务后，才可以获得整个任务的奖励，这给任务的探索带来了挑战。

首先介绍办公室环境[11]。整个环境是一个网格世界，由可以走过的道路、可以开关的门和无法穿过的墙组成，墙和门将整个世界分隔成若干个房间，相邻房间之间通过门来连接。如图 2(a)所示，黑色区域表示墙体，白色区域表示道路，黑色圆圈表示玩家所在的位置，|形表示可以开关的门，正方形则表示任务需要到达的目的地。在这个环境中，智能体可以选择向 4 个方向移动，不能穿过一扇关闭的门，也无法穿越一堵墙。智能体在环境中的任务通常是，从给定的起点出发，将一杯咖啡送到某一房间的特定位置中，其通常需要经过一系列的中间房间才能到达目标房间。在实验中，定义整个办公室环境为一个 $M * N$ 的网格地图，每次生成环境时会随机设置

门和宝物的位置(合理的位置)。环境状态由一个 15 维的向量表示,向量中包括智能体所在的位置信息、环境中门和宝物的位置信息等。



(a)办公室环境 (b)建造环境



(c)导航环境

图 2 实验环境

Fig. 2 Experimental environment

建造环境的灵感则源于一个游戏《我的世界》,但它是实现在一个 2D 的网格世界中[12]。在这个世界中,除了包含可以行走的道路、阻碍前进的墙以及暂时无法通行,只能依靠搭桥连通的河流,还拥有诸如铁锭、树木、铁斧、木桥、钻石等资源以及铁斧和木桥建造台等工具箱。如图 2(b)所示,智能体在左下角,地图上包含一些随机生成的树木、铁矿,以及商店形状的建造台。任务的目标可能是拿到右上角的钻石,但通往钻石的路被河流阻挡了,因此它需要首先建造木桥,使用木桥才可以跨越河流,而木桥的建造又依赖其他资源的获取。智能体可以上下左右移动,还可以与当前位置的物体进行交互。例如,对树木的交互称为伐木,可以获得一块木头;对建造台的交互可能是建造一把铁斧或木桥等。智能体在环境中的任务通常是去拿到某一种资源,这种资源可能是可以直接通过交互获取的,也可能是需要通过建造台建造获得的。一些资源的获取还存在着对其他资源的依赖,如获取木头需要已经拥有铁斧。最复杂的目标还需要智能体构建工具来连通环境中最初不可达的区域。建造环境地图生成具有随机性,在生成环境时会随机生成树木、铁矿、钻石和建造台等物品在环境中的位置。实验中将建造环境的信息用一个 18 维的向量表示,其中包括智能体当前拥有的物品,以及智能体、树、铁矿、钻石和木桥所在的位置信息等。

导航环境(见图 2(c))是一个二维的连续空间环境,由智能体和若干个目标点组成,它们的初始位置都是在一定范围内随机设置的。任务的目标通常是使智能体按照某一顺序到达各个目标点。导航环境的状态信息由长度为 8 的向量表示,向量中包含了智能体距离其他目标点的距离,智能体可以在二维空间中向任意方向移动,其动作空间是二维的(v_x, v_y),v_x 和 v_y 分别表示在 x 和 y 方向上的速度。

5.3 选项发现实验

在该组实验中,主要探究选项发现过程中轨迹切割部分方法的实验效果。

首先,在不同环境中获得足够数量的针对不同任务的演示轨迹。在办公室环境中,针对 5 个任务(主要区别在于起始点和目标终点不同)获得了 1000 条演示轨迹;在建造环境中,则设立了 7 个任务(获得的物品不同,以及智能体初始拥有的物品不同),共获得了 1400 条演示轨迹。由于奖励稀疏,在获取演示轨迹时,本文为其设置了数个子目标奖励以便于加快探索的进展,设置这些子目标奖励的目的只是生成专家演示轨迹,不会用于后续的实验中。

然后,在两个环境上分别建立了符号领域空间,这包括从底层状态到高层状态的映射函数 *Abstract* 和一个符号动作集合。实验中将符号动作的前置条件和后置条件编码,以便于后续的计算。在办公室环境中,将符号状态定义为一个 49 维的稀疏向量,向量中只包含 0 和 1,以表示这一位所对应的谓词是否为真。而在建造环境中,谓词数量更加多样,符号状态则由 122 维向量表示。由于一个符号动作的前置条件和后置条件本质也是一组包含参数的谓词组成,因此当符号动作的参数确定(或称为实例化)时,它的前置条件和后置条件也可由符号状态向量来表示。但这种符号状态并不完整,因为前置条件和后置条件只考虑该符号动作影响的谓词,例如抓取一个苹果动作的前置条件可能只包含表示机械臂是空闲的谓词 empty(arm_1)和表示苹果在桌子上的谓词 on($apple_1$, $table_1$),而与抓取这个动作无关的诸如杯子在地上 on(cup_1, $ground_1$)就不会被关注和提及。因此,本文方法针对该问题设置了前置条件和后置条件掩码,以掩盖环境状态中与前置条件和后置条件无关的位,这将在计算当前符号状态与该符号动作前置条件和后置条件满足度时用到。

最后,使用演示轨迹训练基于符号知识方法的网络,并从演示轨迹中抽取部分用作测试集。测试集中包含一条轨迹中每个状态映射到的符号动作标签,而网络每进行一次切割,就将切割下的这一段子轨迹中的每个状态都映射到根据这一轨迹对应生成的计划 *Plan* 中对应的符号动作中。例如,轨迹 τ_i 对应的计划 $Plan_i=\{10,1,5\}$,其中 10,1,5 分别表示符号动作的编号,那么轨迹 τ_i 第一次切割出的子轨迹 $\tau_i^{b_0}$ 中的所有状态都对应到符号动作 10。因此本文将切割网络的准确率定义为:一条轨迹经过切割后,里面的每个状态对应的符号动作与给定的符号动作标签相比的准确率。每训练 k 个轮次,就测试一次切割的准确率并记录,同时一起记录到的数据还有切割网络的平均奖励和映射网络的平均损失。整体训练中采用批量训练的方式,批数量设置为 100。测试集中的符号动作标签只用于测量切割准确率,不会用于训练中。

在切割实验中,根据表 1 中的设置,将基于符号知识的方法与 Taco 和 Compile 两种基线方法做比较:通过在相同的演示轨迹数据集上训练 3 种方法的切割网络,来评判各自的轨迹切割能力。其中,Taco 训练所需的草图则同样包含在演示轨迹数据集中,但只有 Taco 会在训练中使用到它。对于准确率

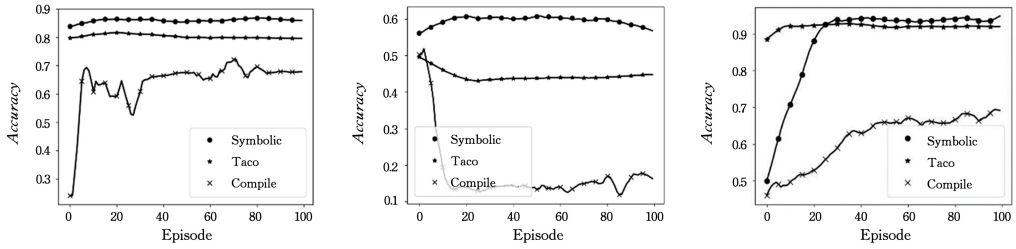
评测，实验在3种方法中选取上述同种方式检测各自的切割准确率并进行对比。

由图3(a)可知，在办公室环境中，Compile方法在训练初期准确率较低，之后逐渐提升到60%～70%；Taco和基于符号知识的方法则在一开始就很快学习到有效的切割策略。本文认为这样的差异表现是因为Compile方法是无监督的，其在训练初期无法获得有效的信息指导，只能从数据特征本身挖掘，因此一开始的切割准确率较低，后续在多个迭代之后，模型逐渐学习到演示数据特征，因此准确率逐步提升。但由于没有提供额外信息，因此Compile最终的准确率不如其他两种方法。而符号知识指导方法和Taco方法则分别是有额外信息提供的强化学习和弱监督学习方法，模型在已有知识的指导下学习的效率更高，因此在两三个迭代内就达到很好的效果。对于基于符号知识的方法和Taco的差异，因为符号知识不仅提供了任务中动作的序列，还包括每个动作的参数和前后置条件，这些信息可以有效地评价切割操作的好坏，因此其拥有比草图更好的指导性。基于符号知识的方法的准确率略高于Taco的81%，达到了89%左右，这也证明了符号知识对选项发现确实提供了有效的帮助。

而当环境切换到更为复杂的建造环境中时，如图3(b)所示，情况发生了变化。Compile方法很难在复杂的状态表示中提取特征用于学习，甚至会因为提取到错误的特征而收敛到较差的情况(这种情况指对于所有轨迹都只选择切割前几个点，因此准确率仅为前几个点与总长度的比值)。依靠草图辅助方法的效果在复杂环境中也受到了制约，Taco在训练中也会出现下降趋势，最终成功率大致收敛到45%。基线工作都因为复杂的环境而不能学习到有效信息，基于符号知识的方法则利用符号能力有效地弥补了这一点，准确率在训练的前几个轮次就达到了60%以上，最终在62%～66%区间波动。

3种方法对办公室环境和建造环境切割表现出的差异，体现了问题复杂性对方法准确率的影响。无监督方法太过依赖数据自身的特征，因此当数据特征变得复杂、不明显(这在建造环境中体现在状态表示中包含智能体拥有的物品，以及树木和铁矿等资源的位置直接或间接地影响任务能否完成)时，这一方法无法有效区分轨迹中不同状态的特征差异。草图可以提供一些有用的信息，这在简单环境中已经足够训练出有效的策略；但对于复杂的环境，智能体在地图中行走的路线可能有交叉和重复，此时仅靠草图提供的动作序列关系，已经难以帮助到策略的学习。基于符号知识的方法，尽管在复杂环境中准确率同样出现了下降(这可能与复杂环境演示数据中轨迹长度有所增加有关)，但仍然可以学习到相对有效的策略，这与本文方法使用强化学习的训练方法有关。因为符号知识是高层的状态和动作，所以可能无法判断出轨迹中每个环境状态所属的符号动作，但它可以判断某一环境状态是否满足了符号动作的前后置条件。因此，以强化学习的方式训练切割网络，可以更有效地利用符号知识。

如图3(c)所示，在基于连续空间的导航环境时，Compile方法的切割准确率逐步上升到65%并保持相对稳定，而Taco方法和符号知识指导方法的准确率则逐步提升到90%以上。这一差异主要源于Compile方法依赖更少的信息，导航环境中的轨迹存在相近甚至交叉的情况，这对于仅靠数据特征分解的方法存在一定的困难。其次，在接近收敛时，基于符号知识方法的准确率达到了95%左右，略高于Taco方法的92%，这体现了符号知识方法的优势，其可以更准确地切割轨迹。



(a)办公室环境 (b)建造环境 (c)导航环境

图3 轨迹切割实验

Fig. 3 Trajectory cutting experiment

从上述环境比较可以看出，基于符号知识的方法在网格环境中相比基线方法准确率均有提升，且该方法受到环境复杂性的影响更小，在连续空间环境中也能保持较高的切割准确率，这也证明了本方法的普适性和广泛性。从图3中的数据可以看出，基于符号知识训练至收敛所需的次数要多于Taco方法，本文认为这是两种方法的训练方式不同导致的。基于符号知识的方法采取强化学习的方式训练切割策略，为了防止无法达到全局最优，对策略更新的幅度较小；而Taco则采取弱监督方法，通过定义损失函数更新策略。另外，基于符号知识的方法呈现的结果大多具有震荡性，即难以像Taco一样保持准确率的稳定趋势。我们认为这与网络的训练方式有关，基于符号知识的方法交替训练了切割网络和映射网络，尽管这两个网络都朝着奖励最大化和损失最小化的方向更新，但是交替的方式决定了网络完全收敛的困难性，因此最终两个网络更可能是在最优值附近震荡，体现在测试结果上即为准确率的上下微小浮动。对于建造环境中，基于符号知识方法的正确率只达到65%而没有继续提升，本文认为原因是其符号知识的局限性，底层状态抽象到高层状态必定会产生信息损失，如相邻的几个底层状态可能对应到同一个高层状态，这就导致了符号知识对切割网络的指导是模糊的、受损的，因此在复杂环境中难以获得更好的效果。如何减小抽象带来的信息损失，也是本方法未来可能的工作方向。

在完成轨迹切割部分后，利用训练出的网络模型，将演示轨迹切割为对应于每一个选项的子轨迹，同时采用上文介绍的行为克隆方法训练每一个选项的开始条件、策略网络和终止网络并保存模型，完成选项发现的整体工作。

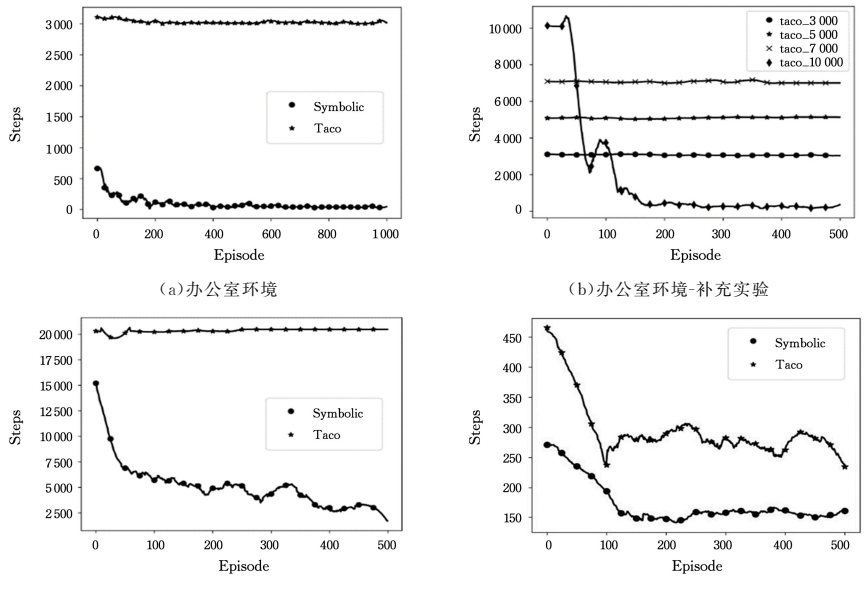
5.4 选项重用实验

重用实验部分，在3个实验环境上随机生成新的任务，并将奖励设置为完成任务为1、其余为0的稀疏奖励。如表1所列，由于Compile方法不支持在不同任务上发现共有的选项，因而对于包含不同任务的数据集，Compile训练出的选项中很可能包含同一子任务的多个重复选项，这给重用选项带来了很大的困难，因此该实验仅设置一个基线Taco。实验将选项的重用能力量化为在新任务中探索达到成功所需的步数和达到收敛所需的迭代次数，需要的步数和迭代次数越少，证明选项的重用能力越强。为了防止无限探索，在每次任务中设置了探索上限次数，并为选项内部尝试设置了最大步数。

在办公室环境中，实验设置的参数为：选项内部尝试的最大步数为200，单次任务总探索步数的上限为3000。由图4(a)可以看出，基于符号知识的方法以较快的速度达到收敛，单次任务探索的步数从一开始的1000降到200以下；而Taco方法则一直没有完成任务，步数始终处于探索上限3000。因此，为了探究Taco在办公室环境的实际重用能力，进行了补充实验，逐步提高Taco的探索步数上限，对比不同探索上限中Taco方法的重用能力。最终结果如图4(b)所示，当探索步数逐渐提高到10000步时，Taco才能有效地学习到策略，这表示在图上则是随着迭代次数的增加，探索步数逐渐下降至收敛。对于两种方法在重用实验上的表现，本文认为是符号知识对智能体选择选项的指导造成了这种差异。由于基于符号知识的方法在发现选项时将选项与一个符号动作绑定，选项依靠符号动作丰富了自身的语义，这可以体现在新任务智能体选择动作时，根据当前状态是否满足已发现选项所对应符号动作的前置条件，提前过滤掉不满足条件的选项，即智能体的动作空间从{原子动作，已发现的选项}缩小到{原子动作，已发现的满足条件的选项}，从而有效提升探索的效率。

在更复杂的建造环境中，实验将选项内部尝试的最大步数设置为500，单次任务总探索步数上限设置为20000。然而，对于目标为探索钻石的长程复杂任务(智能体需要寻找矿石，建造铁斧，砍倒树木，搭建木桥，最终才可找到钻石)，两种方法都难以完成。本文认为原因有两点：1)任务过于复杂而奖励过于稀疏，即便有重用选项帮助探索，智能体也难以在规定步数内找到解决方案；2)两种方法在建造环境的切割实验中准确率都不是很高，这意味着切割数据训练的选项也不一定包含足够良好的策略，从而会削弱重用选项对探索的帮助。因此，实验中调整了任务目标，设置为建造木桥或砍倒树木，降低了任务的难度。此时，由图4(c)可知，基于符号知识的方法有效地学习到了策略，探索步数从15000逐渐下降到2500以下；而Taco方法尽管在开始时曾探索到成功路线，但由于动作空间过大，完成任务的概率过小，最终仍然难以学习到有效的策略。

在连续空间导航环境中，实验将选项内部上限步数设置为200，单次任务总上限步数为3000。实验结果如图4(d)所示，基于符号知识的方法和Taco都较快地探索到了有效的策略，步数从450步分别下降到150和250步。本文认为，这是由于在该环境中，两种方法切割实验的结果均较好，取得了90%以上的准确率，因此都得到了优秀的选项策略，这有助于智能体在新任务中通过重用选项达到任务目标。其次，导航环境中发现的选项数量不是很多，仅与子目标点数量相同，因此，基于符号知识的选项重用方法对选项过滤的作用相对有限。而基于符号知识的方法重用效果仍优于Taco方法，这证明了在连续空间中，符号知识对选项重用的作用仍可帮助智能体更好地完成任务。



(a)办公室环境 (b)办公室环境-补充实验

(c)建造环境 (d)导航环境

图4 选项重用实验

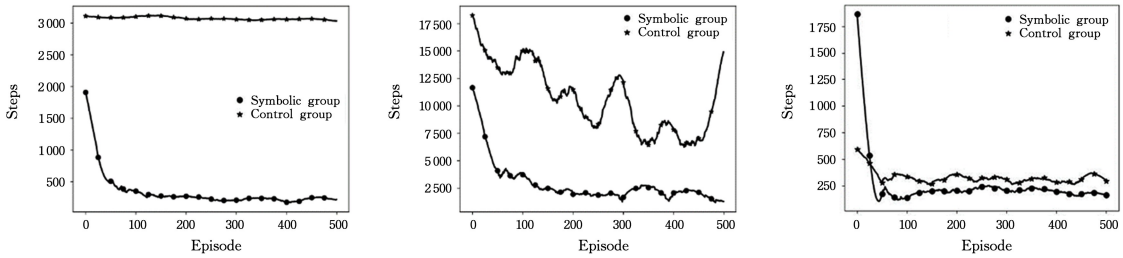
Fig. 4 Option reuse experiment

5.5 消融实验

根据以上实验，本文可以证明基于符号知识的方法在两个环境的重用中都更有优势。然而，在切割实验中Taco的准确率明显低于基于符号知识的方法，而切割的效果也会影响后续的重用能力，所以难以确定符号知识增强选项语义是造成重用实验中两种方法结果差异的主要原因。因此，

进一步做了消融实验。采用基于符号知识的方法切割演示数据并得到训练后的选项，A组是符号知识组，在重用任务中使用符号知识帮助缩小动作空间；B组则是对照组，不借助符号能力，采用正常强化学习方法。

根据图5(a)中的实验结果，在办公室环境中，符号知识组和上文实验一样，学习到了有效的任务执行策略，任务步数很快减少到200以下，而对照组仍然没有探索出有效的策略。而在图5(b)的建造环境中，符号知识组和对照组都学习到了恰当的策略，但符号知识组整体训练的收敛速度明显快于对照组，且对照组最终在迭代结束时，探索完成步数仍然在3000～7500步之间震荡，这显然是学习到的策略未能收敛，而符号知识组则较为稳定地保持在2500以下。如图5(c)所示，在连续空间导航环境中，两组都较快地学习到了优秀的策略，但符号知识组最终收敛时的步数小于对照组。这是由于符号知识筛选不满足的选项，从而减小了动作探索空间，加快了智能体探索速度。而导航环境中两组别实验差异不如办公室环境和建造环境大，本文认为这主要是由于导航空间学习的选项数量不是很多，因此符号知识过滤的效果不够明显。经过实验数据对比，可以明显地看出，符号知识对选项语义的增强可以有效辅助选项的重用，缩小智能体动作空间，不仅可以加快智能体探索的速度，而且可以提高智能体策略的稳定性，使其更好地完成收敛。因此，本文证明了符号知识在选项重用中可以发挥重要的作用。



(a)办公室环境 (b)建造环境 (c)导航环境

图5 消融实验

Fig. 5 Ablation experiment

5.6 局限性

基于符号知识的方法也存在许多局限和有待拓展的方面。由于本文提出的方法依赖符号知识对模型的指导，对于不同环境，需要人为地定义领域空间、符号动作以及底层到高层的符号抽象函数，这一点通常需要依赖开发者对环境的全面掌握和对符号知识的熟练运用，这也是本工作的主要局限。其次，符号知识作为高层抽象表示，本身存在着信息的缺损，因此利用符号知识指导下层的任务难以做到精确和极致。在本文的工作中，这一问题体现在符号知识对切割效果的奖励塑造略为粗糙，可能连续的几个底层动作对应于同一个高层状态，这就为切割的准确性带来了考验。

结束语 本文介绍了一种基于符号知识的强化学习选项发现方法。通过对环境符号建模，所得知识可指导环境中多种任务的选项发现，并为发现的选项赋予符号语义，从而在新任务执行时被重复使用。本文已经证明，在符号知识的指导下，智能体可以更好地依照选项语义发现和重用选项，这比目前已有的基线都拥有更高的准确率和更强的复用能力。通过构建跨任务共享子策略的智能体，基于符号知识的方法可以在稀疏和延迟奖励的任务中取得成功。最后，本文的工作表明，高维度的符号知识可以在一定程度上指导从演示中学习策略的任务。相比为每一个训练数据提供样本，为整个环境构建通用的符号知识总是显得更加容易。

本文也思考了未来工作可能的几个方向。首先，对于符号知识定义困难问题，期望今后可以借助大语言模型，通过输入自然语言的环境信息，自动生成环境的符号领域知识和符号动作的自动编码，这将涉及到提示学习的内容。而对于符号抽象函数，构建网络来拟合函数并通过少样本监督学习可能也是一个更好的途径。其次，对于符号知识的信息受损问题，一方面是更细化符号知识的粒度，以减小信息损失；另一方面则是将高层符号知识与底层环境状态相结合，共同参与切割网络的奖励塑造。

参 考 文 献

[1] ZHOU X, BAI T, GAO Y, et al. Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning[J]. Sensors, 2019, 19(7): 1576.

[2] JAIN D, ISCEN A, CALUWAERTS K. Hierarchical reinforcement learning for quadruped locomotion[C]// 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 7551-7557.

[3] YIN C, YANG R, ZHU W, et al. Survey on multi-agent hierarchical reinforcement learning[J]. CAAI Transactions on Intelligent Systems, 2020, 15(4): 646-655.

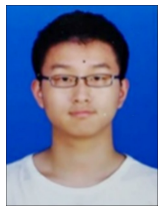
[4] SUTTON R S, PRECUP D, SINGH S. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning[J]. Artificial Intelligence, 1999, 112(1/2): 181-211.

[5] HOVLAND G E, SIKKA P, MCCARRAGHER B J. Skill acquisition from human demonstration using a hidden markov model [C]// Proceedings of IEEE International Conference on Robotics and Automation: volume 3. IEEE, 1996: 2706-2711.

[6] SCHAAL S. Dynamic movement primitives—A framework for motor control in humans and humanoid robotics[M]// Adaptive Motion of Animals and Machines. Berlin: Springer, 2006: 261-280.

[7] KONIDARIS G, KUINDERSMA S, GRUPEN R, et al. Robot learning from demonstration by constructing skill trees[J]. The

International Journal of Robotics Research, 2012, 31(3): 360-375.
[8] KIPF T, LI Y, DAI H, et al. Compile: Compositional imitation learning and execution[C]// International Conference on Machine Learning. PMLR, 2019: 3418-3428.
[9] SHANKAR T, TULSIANI S, PINTO L, et al. Discovering motor programs by recomposing demonstrations[C]// 8th International Conference on Learning Representations(ICLR). 2020.
[10] CHEN Y, WANG C, BASTANI O, et al. Program synthesis using deduction-guided reinforcement learning[C]// Computer Aided Verification: 32nd International Conference(CAV 2020). Springer, 2020: 587-610.
[11] ICARTE R T, KLASSEN T, VALENZANO R, et al. Using reward machines for high-level task specification and decomposition in reinforcement learning[C]// International Conference on Machine Learning. PMLR, 2018: 2107-2116.
[12] ANDREAS J, KLEIN D, LEVINE S. Modular multitask reinforcement learning with policy sketches[C]// International Conference on Machine Learning. PMLR, 2017: 166-175.
[13] SHIARLIS K, WULFMEIER M, SALTER S, et al. Taco: Learning task decomposition via temporal alignment for control [C]// International Conference on Machine Learning. PMLR, 2018: 4654-4663.
[14] ARGALL B D, CHERNOVA S, VELOSO M, et al. A survey of robot learning from demonstration[J]. Robotics and Autonomous Systems, 2009, 57(5): 469-483.
[15] ESMAILI N, SAMMUT C, SHIRAZI G. Behavioural cloning in control of a dynamic system[C]// 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century: volume 3. IEEE, 1995: 2904-2909.
[16] PETERS J, KOBER J, MÜLLING K, et al. Towards robot skill learning: From simple skills to table tennis[C]// Machine Learning and Knowledge Discovery in Databases: European Conference(ECML PKDD 2013). Springer, 2013: 627-631.
[17] NIEKUM S, OSENTOSKI S, KONIDARIS G, et al. Learning and generalization of complex tasks from unstructured demonstrations[C]// 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012: 5239-5246.
[18] NIEKUM S, OSENTOSKI S, KONIDARIS G, et al. Learning grounded finite-state representations from unstructured demonstrations[J]. The International Journal of Robotics Research, 2015, 34(2): 131-157.
[19] ZHU Y, STONE P, ZHU Y. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation [J]. IEEE Robotics and Automation Letters, 2022, 7(2): 4126-4133.
[20] ARTO A G, MAHADEVAN S. Recent advances in hierarchical reinforcement learning[J]. Discrete Event Dynamic Systems, 2003, 13(1/2): 41-77.
[21] PARR R, RUSSELL S. Reinforcement learning with hierarchies of machines[M]// Advances in Neural Information Processing Systems 10. The MIT Press, 1997: 1043-1049.
[22] DAYAN P, HINTON G E. Feudal reinforcement learning[M]// Advances in Neural Information Processing Systems 5. Morgan Kaufmann, 1992: 271-278.
[23] SUTTON R S, PRECUP D, SINGH S. Intra-option learning about temporally abstract actions[C]// ICML: volume 98. 1998: 556-564.
[24] FOX R, KRISHNAN S, STOICA I, et al. Multi-level discovery of deep options[J]. arXiv: 1703.08294, 2017.
[25] KRISHNAN S, FOX R, STOICA I, et al. Ddco: Discovery of deep continuous options for robot learning from demonstrations [C]// Conference on Robot Learning. PMLR, 2017: 418-437.
[26] SHANKAR T, GUPTA A. Learning robot skills with temporal variational inference[C]// International Conference on Machine Learning. PMLR, 2020: 8624-8633.
[27] XIE Y, ZHOU F, SOH H. Embedding symbolic temporal knowledge into deep sequential models[C]// 2021 IEEE International Conference on Robotics and Automation(ICRA). IEEE, 2021: 4267-4273.
[28] YANG F, LYU D, LIU B, et al. Peorl: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making[C]// Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence(IJCAI). 2018: 4860-4866.



WANG Qidi, born in 2000, postgraduate. His main research interests include reinforcement learning and program synthesis.

SHEN Liwei, born in 1982, associate professor. His main research interests include man-machine and object fusion system software, and robot software engineering.

(责任编辑:柯颖)