用于适应开放世界新颖性的神经符号世界模型

Jonathan C. Balloch 计算学院 佐治亚理工学院 亚特兰大，美国
balloch@gatech.edu

Zhiyu Lin 计算学院 佐治亚理工学院 亚特兰大，美国 zhiyulin@gatech.edu

Robert Wright 佐治亚科技研究所 亚特兰大，美国
robert.wright@gtri.gatech.edu

Xiangyu Peng 计算学院 佐治亚理工学院 亚特兰大，美国 xpeng62@gatech.edu

Mustafa Hussain
计算学院
佐治亚理工学院
亚特兰大, USA
mustafa.hussain@gatech.edu

Aarun Srinivas
计算学院
佐治亚理工学院
亚特兰大, USA
asrinivas5@gatech.edu

Julia M. Kim，计算学院，佐治亚理工学院，美国亚特兰大，
julia.kim@gatech.edu

Mark O. Riedl，计算学院，佐治亚理工学院，美国亚特兰大，
riedl@cc.gatech.edu

arXiv:2301.06294v1 [cs.AI] 16 Jan 2023

摘要

开放世界中的新颖性——环境机制或属性的突然变化——在现实世界中十分常见。新颖性适应是指智能体在新颖性出现后提升其策略性能的能力。大多数强化学习（RL）方法都假设世界是一个封闭且固定的过程。因此，强化学习策略难以高效地适应新颖性。为此，我们提出了 WorldCloner，这是一种端到端可训练的神经符号世界模型，可实现快速的新颖性适应。WorldCloner 学习前新颖性环境状态转移的高效符号表示，并利用这一世界模型检测新颖性，以单次方式高效适应新颖性。此外，WorldCloner 通过基于想象的适应增强策略学习过程：世界模型模拟新颖性出现后的环境状态转移，帮助策略适应。将“想象”的状态转移与新颖性出现后的环境中的交互相结合，可以用更少的环境交互总次数恢复性能。通过使用专为研究序贯决策问题中新颖性而设计的环境，我们证明，与纯神经强化学习方法相比，符号世界模型能帮助其策略更高效地适应。

关键词

开放世界学习、新颖性适应、强化学习、世界模型

1 引言

大多数逼真的环境都是“开放世界”，其中很可能会遇到新颖且出乎意料的情况。请考虑以下示例：一扇门被重新配制了钥匙，因此之前的钥匙无法将其打开，需要使用另一把钥匙，从而改变了解；一种新技术使车辆能够驶过此前无法通行的地形，从而缩短了到目的地的距离；火山喷发突然改变了当地气候，导致本土作物歉收，使得养活社区更加困难。我们将这些变化称为“新颖性”，即环境的基本机制或属性发生了此前未曾出现过的变化 [6, 23]。新颖性可能会极大地改变序贯决策任务的解的分布。如果智能体——无论是人类还是人工智能——未能适应新颖性，而仍继续采取相同的策略，其任务表现可能会骤然下降。

快速且高效利用样本适应新颖性至关重要，例如涉及人身安全时。因此，人类进化出了出色的适应新颖性的能力 [7]。相比之下，深度强化学习（RL）智能体虽然在围棋 [30], 、Atari [2],和星际争霸 [33] 等游戏任务中展现出超越人类的性能，却特别容易在面对新颖性时失败。新颖性会导致环境动态发生分布偏移。因此，新颖性可以表述为在线任务迁移 [4]——在在线执行期间，智能体会突然发现自己身处一个具有全新转移动力学的环境中。传统的迁移方法是在新颖性出现后的环境中从头开始重新训练。也可以让智能体从已有模型参数出发，在新颖性出现后继续学习。然而，这种方法的样本效率可能很低，需要在部署期间与环境进行数十万次交互，才能收敛到新的最优策略。

我们提出了 WorldCloner，这是一种高效的世界模型强化学习系统，采用神经策略，并包含两项针对新颖性的改进，以增强标准深度强化学习执行循环。(1) 我们提出了一种简单、可快速更新的符号世界模型，用于学习转移函数的模型——即环境特征如何随时间变化，以及如何被改变。与可能需要对某种现象进行多次观察的神经世界模型不同，我们的符号世界模型只需一次

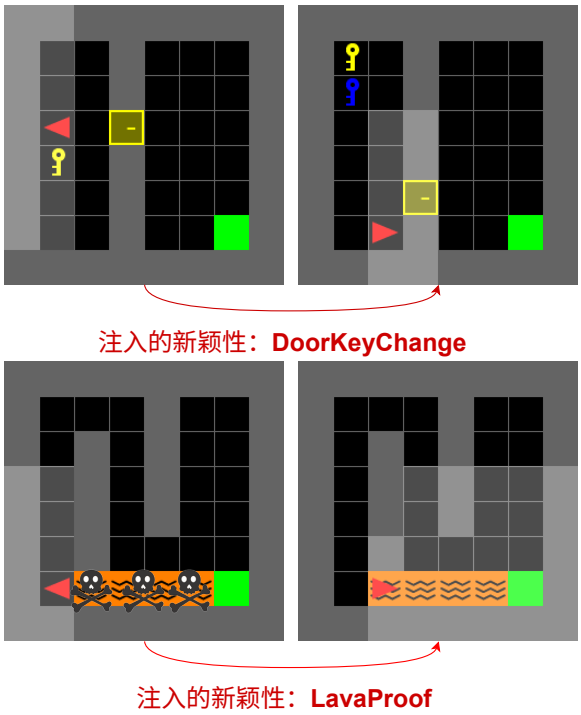

图 1：NovGrid 环境，智能体（红色三角形）必须到达目标（绿色方框）。新颖性无法直接观察到；智能体必须亲身体验新颖性才能察觉。上：前新颖性阶段，只有黄色钥匙能打开门；新颖性出现后，只有蓝色钥匙能打开门。下：前新颖性阶段，熔岩会带来 -1 奖励并使状态终止；新颖性出现后，熔岩可以安全行走。

新颖性出现后的观测即可完成更新，因此比神经世界模型更快地适应新颖性。(2) 我们提出了一种基于想象的适应方法，利用符号世界模型提高策略在部署时适应的效率。检测到新颖性后，智能体会更新其世界模型，并使用更新后的世界模型模拟新颖性出现后的环境中的环境转移，从而减少更新策略所需的真实环境交互次数。

我们在包含多种新颖性类型的 NovGrid 环境 [5] 中评估 WorldCloner 的样本效率。我们表明，与无模型强化学习和神经世界模型强化学习技术相比，使用 WorldCloner 进行新颖性出现后的适应所需的策略更新次数和环境交互次数更少。

总而言之，我们的贡献如下：

- 我们提出了 WorldCloner，这是一种用于新颖性检测与适应的神经符号世界模型。
- 我们提出了一种新的符号表示，并设计了高效的学习算法，以及利用该表示帮助世界模型适应新颖性的方法。
- 我们表明，WorldCloner 对新颖性的适应效率高于最先进的强化学习算法。

2 背景

新颖性处理。 新颖性是指数据集、数据流或环境基本要素发生的突发且此前未曾见过的变化 [6, 23]。新颖性处理可分为三项挑战：新颖性检测、新颖性表征和新颖性适应。针对序贯决策问题中具备新颖性感知能力的智能体，已有研究包括自适应混合连续—离散规划，以及结合演员-评论家强化学习技术使用知识图谱，以改进检测和适应能力 [17, 19, 22, 26]。对“隐藏”领域偏移的适应 [8] 也与此相关，尽管它最初并非以应对新颖性为目标。

对于新颖性适应而言，明确检测并刻画新颖性并非绝对必要。在线连续或终身学习 [31]以及在线学习 [16, 29], 中，学习器会假设世界过于复杂或难以预测，无法通过离线学习来掌握，因此在功能上将世界视为永远处于新颖状态。这些学习器不区分训练与部署，而是持续更新。采用这些技术时，无论是在新颖性出现之前还是之后，都几乎无法保证收敛。

从具备一定前新颖性知识的状态开始，即便朴素的在线学习方法最终能在新颖性出现后收敛，也可能引发灾难性推断 [20], , 导致智能体几乎无法迁移先前模型中的知识。这种遗忘现象会降低学习效率。迁移学习 [35]旨在通过将模型迁移到仅存在细微但实质性差异的环境中，克服这一局限。尽管这种方法与新颖性适应有相似之处，迁移学习技术通常假设智能体遇到新的数据集或环境时，可以返回离线训练阶段；而新颖性适应则必须在部署期间进行。

强化学习中的世界模型。 强化学习大致可分为基于模型和无模型 [32]两类。在基于模型的强化学习中，智能体会学习转移函数 的模型，并使用该模型通过蒙特卡洛树搜索 [30] 等迭代技术，或价值迭代等动态规划方法来制定策略。在无模型强化学习中，策略模型是直接学习得到的，无需转移函数，然后通过采样来确定智能体的动作。对于新颖性适应，基于模型的方法可能具有更高的样本效率，但与无模型方法相比，它们往往对建模误差更加敏感 [32]。

世界模型 会同时学习转移函数和策略，并结合二者来提升智能体的性能 [12, 27]。Dream- erV2 [14] 代表了基于世界模型的强化学习领域的最先进水平。在Dreamer方法中，只有世界模型直接从与环境的交互中学习，而策略则完全通过世界模型“想象”出的交互进行训练 [13]。这建立在先前的世界模型研究之上；那些研究利用想象来帮助训练标准强化学习模型 [12, 18, 25] 和多智能体模型 [24]。DreamerV2的样本效率高于大多数无模型强化学习智能体，但仍需要一个非常大的模型；它并非专为新颖性处理而设计。此前没有研究使用基于世界模型的强化学习来实现新颖性适应。

强化学习通常将环境建模为标准马尔可夫决策过程（MDP）：$M = S,\ A,\ R,\ P,\ \gamma$，其中 S

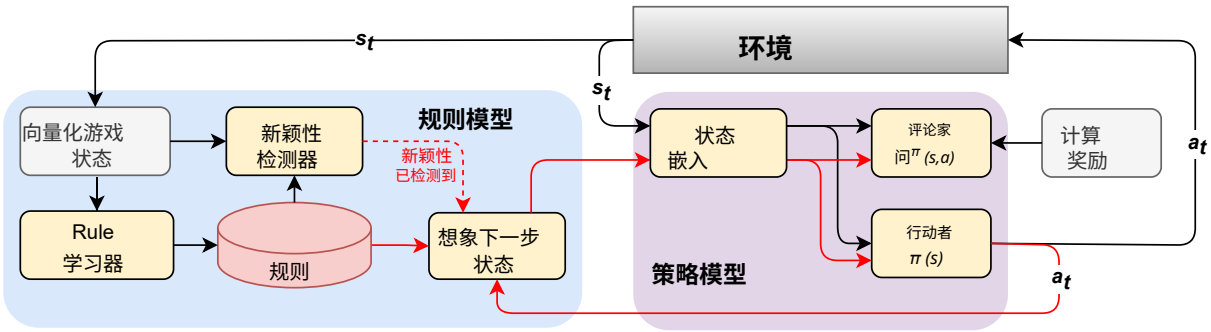

图 2：WorldCloner 架构。紫色模块和黑色弧线表示传统的强化学习执行循环，其中损失通过紫色模块中的黑色弧线向后反向传播。蓝色模块包含规则模型学习和新颖性检测。红色弧线表示新颖性出现后的环境中的信息流，使用学习到的规则模拟新环境。在新颖性出现后，损失沿着策略模型内的红色弧线和黑色弧线向后反向传播。

表示环境状态集合，A表示动作集合，R：$S\times A\to\mathbb{R}$是将状态和动作映射为标量奖励的函数，P：$S\times A\to\mathbb{S}$是状态间的转移函数，γ是折扣因子。在经过任意且先验未知数量的回合或游戏后t引入新颖性，构成了从原始环境或马尔可夫决策过程M到新环境或马尔可夫决策过程M_t的转变。除智能体的决策模型外，问题的所有方面都被视为环境的属性。这包括智能体形态、动作前置条件和效果。我们对问题作出几项限制。我们假设每个智能体的观测空间和动作空间在整个过程中保持一致。我们还假设智能体的任务T始终保持一致，因此外在奖励函数也始终保持一致。

3 WORLDCLONER

WorldCloner 是一种可端到端训练的神经符号世界模型，由两个组成部分构成：（1）神经策略和（2）符号世界模型。符号世界模型（我们也称之为“规则模型”）由一组规则 组成，这些规则共同近似环境的潜在状态转移函数。规则模型有两项核心功能。首先，规则模型会学习预测新颖性出现前的状态转移。因此，违反规则表明出现了新颖性，需要更新规则模型和策略。其次，在新颖性出现后的环境中，WorldCloner 使用规则模型模拟环境，从而生成轨迹展开，用于重新训练神经策略模型，减少与真实环境的交互次数。如图 2 所示，世界模型与策略之间的这种交互使 WorldCloner 能够在新颖性出现前信任其策略，并在新颖性出现后更多地依赖其世界模型。我们的世界模型算法经过专门设计，使规则模型独立于神经策略的实现，因此，只要策略框架使用与规则模型相同的数据输入，我们的方法就能与之兼容。在 WorldCloner 的实现中，我们采用基于优势演员-评论家（A2C）神经架构的近端策略优化（PPO）[28]。

3.1 基于区间的符号世界模型

在 WorldCloner 中，用于建模转移函数 P 的符号世界模型表示为一组形式为 c_s，c_a，e 的 K规则 $\{\rho_k\}$。在这种表示中，c_s 是状态前置条件，c_a 是动作前置条件（类似于 do演算 前置条件 do(a)），而 e 是效果。如果输入状态 s 和动作 a 与某条规则的前置条件相匹配，则判定该规则 ρ 适用。状态前置条件包含一组值，对应于状态特征 $\phi_1...\phi_m$ 的一个子集。当 $\{\rho\}$ 的状态前置条件和动作前置条件都得到满足时，$\{\rho\}$ 即适用，并可在被选中时执行。效果 e 是输入状态与预测状态之间的差异：$e=s'-s$。这种表述方式通过编码前置条件和效果，与 ADL 和 PDDL [21] 等逻辑演算框架有相似之处。我们的方法旨在通过学习而非人工设计来获得，与“游戏规则”学习 [11]类似。

WorldCloner 将前置条件独特地表示为一组轴对齐包围区间 （AABI），也称为特征空间中覆盖训练数据的超矩形或 n-正交多面体 [10]。AABI 是简单的 d 维凸几何体：给定一组需要归类的样本点 $x_1...x_n$，它们会确定每个维度上能够包围整个样本集的最小区间。无论区间大小如何，AABI 都可由两个d维点定义——一个下界和一个上界——因此在训练和推理时查询起来都非常高效。它们可以容纳连续变量与类别变量（非连续变量）的混合；这两类变量在符号方法中都很常见，而类别型 AABI 的取值就是完全匹配的值集合。比如，参见图 3底部，其中展示了 NovGrid 网格世界中一条用于开门的规则所对应的 AABI。在此例中，区间将动作限制在单个智能体位置（3，5）。图4展示了另一个例子，其中动作适用于一个位置区间。

前置条件的 AABI 不必相交；每条唯一规则都可以包含多个不相交的区间。我们的规则更新算法（见下一节）会将不同区间的数量降至最低。当动作根据当前状态产生不同效果时，同一个动作就会存在多条规则。例如，在网格世界中，当智能体前方没有障碍物时，前进动作会沿智能体面朝的方向改变状态中的 ϕ 位置特征；但如果智能体正前方有一堵墙，前进动作就不会对状态产生任何效果。

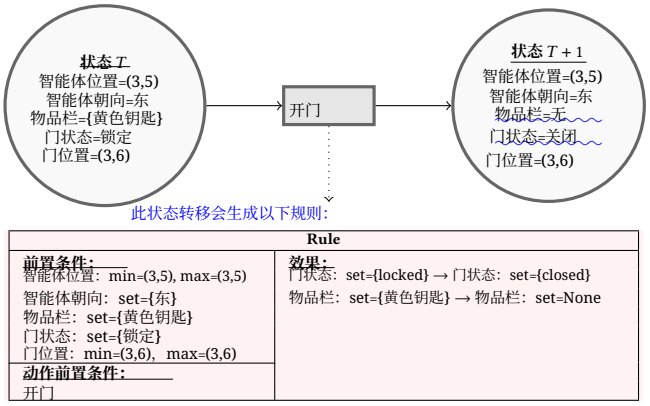

图3：上方展示传递给规则学习器的环境状态示例（变化部分已加下划线）。下方展示学习到的世界模型规则，描述钥匙如何开门。

算法1： 规则模型更新

输入： 先前状态 s_{t-1}，动作 a_{t-1}，下一状态 s，世界模型规则集 P
输出： 应用规则 ρ
状态变化 $\delta s = s - s_{t-1}$;
$RuleHit$ =假;
对于 规则 $\rho_k = \langle c_{s,k}, c_{a,k}, e_k \rangle$ 在 P **执行**
  如果 $a_{t-1} == c_{a,k}$ **则**
    如果 $\delta s == e_k$ **则**
      如果 $CollisionCheck(s_{t-1}, c_{s,k})$ **则**
        $RuleHit \leftarrow$ 真;
        返回 ρ;
      else
        $RuleRelaxation(\rho, s_{t-1})$;
        $RuleHit \leftarrow$ 真;
    else
      如果 $CollisionCheck(s_{t-1}, c_{s,k})$ **则**
        $RuleCollisionResolution(\rho, s_{t-1})$;
      else
        $RuleCreation(c_s = s_{t-1}, c_a = a_{t-1}, e = \delta s)$;
        $RuleHit \leftarrow$ 真;
如果 $RuleHit$ == 假**那么**
  $RuleCreation(c_s = s_{t-1}, c_a = a_{t-1}, e = \delta s)$;

这种功能还使我们能够处理概率转移；多条规则将具有相同的动作和状态前置条件，但具有不同的 效果分布。以使用钥匙打开锁着的门为例，锁可能会“卡住”，智能体可能需要尝试多次。于是，预测开锁的规则的效果将是一个分布，该分布涵盖具有相同前置条件但效果不同的规则。

3.2 规则学习

规则学习过程（算法1）构建一组紧凑、无冲突的规则，在最大限度覆盖状态-动作空间的同时，尽可能降低符号世界模型的复杂度。此外，这是一个在线更新过程；一旦学会某条规则，即可在无需了解过去观测的情况下更新该规则。

规则更新过程从将规则模型初始化为空集开始。在环境中执行一个动作后，

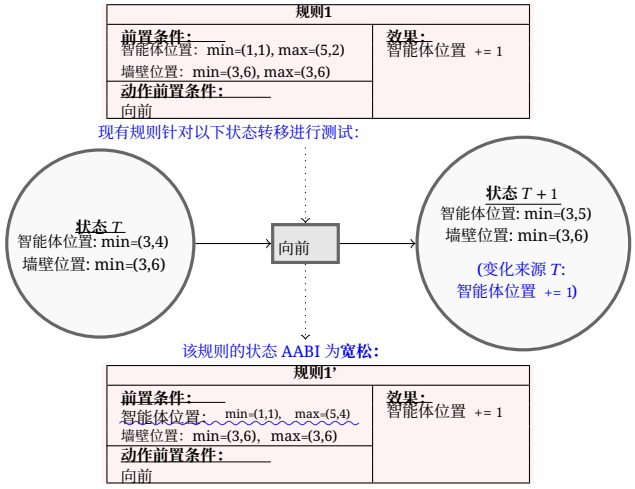

图4：规则放宽示例，其中，与智能体位置对应的蓝色下划线前置条件 AABI 已在修改后的规则1’中扩展，以纳入状态S中的智能体位置。

规则学习器接收环境的先前状态，以推导出前置条件；接收所执行的动作，以及环境的新状态，以推导出效果。更新算法将先前状态、动作和新状态，分别与模型中任何现有规则的状态前置条件、动作前置条件和效果进行比较，然后进入以下四种情况之一：

(1) 无变化：先前状态位于现有规则的状态前置条件 AABI 内，且该规则的动作和效果匹配。(2) 规则创建：不存在动作前置条件得到满足或状态差异与效果匹配的规则。此时会创建一条新的“点”规则，精确描述先前状态。(3) 规则放宽：存在一条规则，其动作前置条件得到满足，且状态差异与效果匹配，但先前状态不在该规则现有的状态前置条件 AABI 覆盖范围内。通过扩展 AABI 来“放宽”该规则。(4) 规则冲突消解：存在一条规则，其动作前置条件和状态前置条件 AABI 均得到满足，但效果不同。使用最小割（min-cut）操作对现有规则的 AABI 进行拆分。

当智能体执行一个动作时，规则学习器会观察时间 T 的先前状态、已执行的动作以及时间 $T+1$ 的下一状态。如果存在满足状态转移的动作前置条件、状态前置条件或效果的规则，我们会先尝试通过规则放宽和规则冲突消解来修改现有规则。只有在规则冲突消解和规则放宽均不适用时，才需要创建规则。对于智能体执行的动作，我们最初只考虑满足动作前置条件的规则。接着，我们判断动作执行前的状态是否包含在这些现有规则中任一规则的 AABI 内。这通过几何超平面分离定理（也称为分离轴定理） [3, 15]来实现。从几何角度而言，对于所有特征 $\phi_d \in \Phi$，如果存在某个特征，使得该点小于某一区间的最小值或大于其最大值，那么

就存在一个分离超平面。该定理简单来说是指，如果特征空间中该点与几何形状之间存在一个超平面，则二者不会发生碰撞，且该点位于AABI 之外。具体而言，对于先前状态 s_{t-1} 和 AABI $I = [I_{min}, I_{max}]$:

$$s_{t-1} \notin I \iff \exists \phi \; s.t. \; [s_{t-1,\phi} > I_{max,\phi} \cup s_{t-1,\phi} < I_{min,\phi}]. \quad (1)$$

规则创建很简单：在观测到的状态-动作-状态转移中，先前状态与动作的组合不属于任何现有规则的 AABI。此时会向规则模型中添加一条新规则，其中连续状态特征的 AABI 的最小值和最大值均设为其当前值，而类别状态特征值则设为其所属特征的单元素成员。同样，规则的动作前置条件设为状态转移中的动作，效果则等于 s_{t-1} 与 s之间的差值。图3展示了这一过程。我们将每个特征都限定为单一取值集合的规则称为“点”规则。

规则放宽旨在扩展现有规则的 AABI，使其匹配新遇到的动作前置条件和效果。操作很简单：对于先前状态 s_{t-1} 以及由点 I_{min}和 I_{max}表示的 AABI I，放宽后的最小点和最大点分别为 $I^*_{min} = \min(s_{t-1}, I_{min})$ 和 $I^*_{max} = \max(s_{t-1}, I_{max})$。图4展示了规则放宽。此处，已有一条先前学到的规则，用于描述前进动作导致的智能体位置变化。该规则与动作前置条件和效果匹配，但与状态前置条件不匹配，可能只是因为在环境的那个区域尚未观测到这种情况。因此，AgentPosition AABI 会扩展以涵盖该观测。

扩展后，会再次检查新的 AABI I^*，这次是使用上述相同的几何超平面分离定理，检查它是否与其他现有规则发生碰撞。不过，在比较区间时，我们通过比较区间的最大值与最小值，检查$I*$ 与另一条规则 ρ_k 的 AABI I^k 之间是否存在超平面。给定最大值点$[I^*_{min}, I^*_{max}]$和最小值点 $[I^k_{min}, I^k_{max}]$，如果存在一个特征 ϕ使得$I^*_{min} > I^k_{max}$，或反之，则不会发生碰撞。如果确实发生碰撞，我们不会尝试在两条规则之间折中，而是执行规则冲突消解。

规则冲突消解会找到现有规则状态前置条件 AABI 的最小割划分。将 AABI 视为一个图，其中图节点是边界超平面（即超矩形的“面”），边是连接相对超平面的线段，其权重为线段长度。这个 AABI 的最小割就是沿着与先前状态相交的最大特征轴进行划分。将划分后得到的新 AABI 添加到现有规则的前置条件中，移除原有 AABI，并评估先前状态是否需要通过规则创建或规则放宽来加以容纳，因为它可能不属于任一拆分结果。图5展示了一种情况：观察到向前规则未能正确预测向前发生的状态转移结果，因为智能体撞上了一堵墙。由于描述该状态转移的规则具有被现有向前规则涵盖、但效果不同的前置条件，因此发生了规则冲突。通过拆分规则来解决该冲突。由于先前状态现在位于拆分规则的 AABI 之外，因此可以创建一条新的点规则。

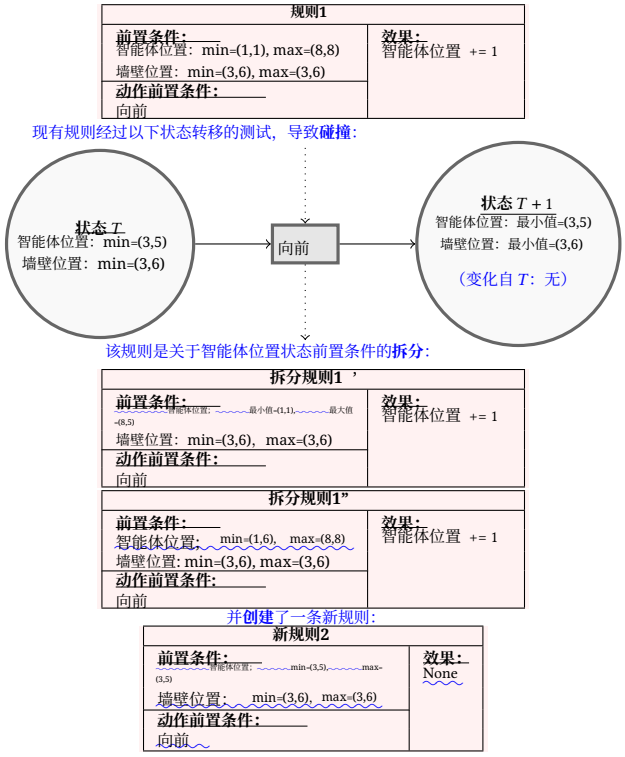

图5：规则冲突，以及由此产生的规则拆分和创建。新拆分出的规则1’和规则1”中带蓝色下划线的前置条件，标示了原始规则1被拆分所依据的特征维度。新创建的规则2解释了导致其与原始规则1发生冲突的状态转移。

出现新颖性后，在规则冲突消解中拆分规则，可能导致其中一条拆分规则的前置条件包含一个具有空区间或空类别集合的特征。如果发生这种情况，则舍弃这个“空拆分”。

3.3 新颖性检测

按照上述方法创建符号世界模型后，前新颖性神经策略会收敛，此时关闭二者的学习；这样既能节省计算，也能让策略专注于利用。智能体在环境中执行任务时，会寻找与世界模型中规则不一致的状态-动作-状态转移，并在以下两种情况之一发生时触发适应。（1）同一规则连续被违反 n 次。若观察到的环境后续状态与规则预期的状态效果不符，但状态和动作前置条件均匹配，则定义为一次违反。（2）同一观察状态导致预测失败，且连续访问次数超过 n 次。 n 是用于控制对新颖性敏感度的超参数，在我们的实验中设为 $n = 2$。

一旦检测到新颖性，神经策略和符号世界模型便再次开始在线更新。新颖性后规则更新过程与新颖性前规则学习（算法1）完全相同。因此，只需执行一次规则学习算法迭代，规则模型便可得到更新，

算法 2： 基于想象的适应

输入： 新颖性出现前的策略 π、世界模型 P、混合比例 η
$PostNov \leftarrow False$;
$s_t \leftarrow$ 初始观测;
当 $true$ **时**
  选择动作 $a_t = \pi(s_t)$;
  预测下一个状态 $\hat{}s_{t+1} = P(s_t, a_t)$;
  在 Env 中执行 a_t 并观察下一个状态和奖励 s_{t+1}, r_t;
  $PostNov \leftarrow [PostNov$ **或** $DetectNovelty(P, \hat{}s_{t+1}, s_{t+1})]$;
  如果 $PostNov$ **则**
    将 $\langle s_t, a_t, s_{t+1}, r_t \rangle$ 添加到 $UpdateBuffer$;
    $P \leftarrow RuleModelUpdate(s_t, a_t, s_{t+1})$;
    将 $ImagineRollouts(P)$ 每 η 添加到 $UpdateBuffer$;
    定期使用 $UpdateBuffer$ 更新 π;
  $s_t \leftarrow s_{t+1}$

并保证下一状态预测得到改进。

3.4 基于想象的策略适应

新颖性出现后，符号世界模型中更新后的规则反映了智能体对新颖性出现后状态转移函数的认知。智能体现在使用该规则模型来“想象”一系列动作将如何展开——我们将其称为“基于想象的模拟”——并在不与真实环境交互或执行动作的情况下更新其策略。具体而言，如算法 2 所示，我们使用规则模型模拟状态-动作-状态转移，并将其填入智能体的更新缓冲区——即用于训练策略的数据。策略训练算法根据从更新缓冲区抽取的样本计算损失，并通过策略模型反向传播损失。图 2 展示了标准的演员-评论家神经网络架构如何与基于想象的模拟协同工作，以提供真实和想象的状态观测。

智能体在想象环境中遵循其策略，并反复体验第一条规则变化所带来的后果，获得降低（或提高）的预期奖励，从而促使策略远离（或趋向于）受影响的动作。随着符号世界模型检测到新的差异，并更准确地表征新颖性出现后的环境，策略或许能够“想象”体验并适应新颖性，而无需在环境中大量接触该新颖性。

继续以重新配钥匙的门锁为例（图 1，上方），智能体执行了新颖性出现前的策略，先前往黄色钥匙处，再前往黄色门处，却发现门已无法打开。当智能体到达状态 智能体在门前，智能体持有黄色钥匙, 门已锁定并执行解锁动作时，它预期 $DoorUnlocked$ 变为真。然而，由于门锁已重新配钥匙，解锁动作没有带来任何变化。发生这种情况时，就出现了规则冲突；解决方法是创建一条具有 $\{\emptyset\}$ 效果增量的规则，并将旧规则“拆分”。在此例中，拆分后得到的状态具有空 AABI，因此该规则会被删除。规则冲突被检测为新颖性，智能体开始更新其策略。具体而言，智能体不再获得穿过门的效用，策略更新也反映出直接位于门前的效用降低。随着时间推移，该状态的效用会降低到一定程度，使智能体倾向于选择其他状态，从而增加探索，并最终找到蓝色钥匙。这种从利用转向探索的过程，因智能体能够反复设想自己到达门前并选择其他动作而加速，因为不存在任何能使门打开的有效规则。

最终，通过探索，智能体会来到手持蓝色钥匙的门前：⟨门前的智能体，携带蓝色钥匙的智能体，门已锁定⟩ 这一次，尝试用蓝色钥匙打开门将产生 $DoorUnlocked$ 变为 $DoorClosed$的效果。该效果与解锁动作的前置条件不匹配任何现有规则，于是便会创建一条新规则，此时想象将促进更快速的策略学习。再次获得门后状态的效用和奖励后，智能体将收敛到一项涉及蓝色钥匙的新策略。

为考虑世界模型误差，我们也会使用真实环境交互来训练策略。混合比例参数用于控制策略更新缓冲区中想象环境与真实环境轨迹展开的比例；在真实环境中每执行 t 步，智能体就会在想象环境中执行 $\frac{t}{\eta}$ 步。这种混合实际上为策略更新引入了噪声，有助于推动策略回到“探索”模式，在该模式下，策略会更随机地选择动作。

4 实验

我们在 NovGrid [5] 环境中进行实验。该环境扩展了 MiniGrid [9] 环境，加入了可在测试期间任意时刻注入的新颖性。NovGrid 支持针对标准新颖性开展可控、可复现的实验。我们使用两个 8x8 MiniGrid 环境：(1) DoorKey，智能体必须拾起钥匙、打开一扇门，并前往门后的目标；(2) LavaShortcutMaze，智能体必须穿过一个迷宫，而迷宫靠近目标的一侧沿线分布着一池熔岩。在所有情况下，我们都使用 MiniGrid 默认的稀疏奖励。

我们测试了方法在三种新颖性类型上的性能：捷径、变化和屏障 [5]。LavaProof 是一种捷径新颖性，它使 LavaShortcutMaze 环境中的熔岩对智能体无害（在前新颖性阶段，熔岩会摧毁智能体），从而提供一条通往目标的更短路径。DoorKeyChange 是一种变化新颖性，它改变 DoorKey 环境中两把钥匙中哪一把可以开门；这不会改变抵达目标的难度，但需要采用长度和复杂度相近的不同状态—动作序列。LavaHurts 是一种屏障新颖性，也是 LavaProof 的逆向变化：它改变 LavaShortcutMaze 中熔岩的效果，使熔岩会摧毁智能体（在前新颖性阶段，熔岩无害），从而消除通往目标的熔岩捷径。DoorKeyChange 和 LavaProof 这两种新颖性在图 1 中作了概念性展示。

为了评估智能体在这些测试环境中的性能，我们采用 Balloch 等人基于 Alspector 提出的两项新颖性适应指标。 [5] [1]。(1) 渐近适应性能 是智能体在新颖性出现后的环境中的最终性能，相对于随机基线而言，数值越高越好。该指标用于观察一种方法是否已完全适应新颖性出现后的环境。(2) 自适应效率 是智能体在真实的新颖性出现后的环境中达到渐近适应性能所需的时间步数，步数越少越好。在我们的工作中，当 10步移动平均法达到渐近

	自适应效率 @0.95（步数）↓	前新颖性性能 ↑	渐近性能 ↑	更新效率（策略更新）↓
DoorKeyChange 新颖性				
PPO	2.25E6	0.973	0.971	2.25E6
DreamerV2	5.3E5	0.971	0.973	3.82E8
Ours	9.8E5	0.972	0.970	**1.63E6**
LavaProof 新颖性				
PPO	1.39E5	0.972	0.991	1.39E5
DreamerV2	适应失败	0.965	适应失败	适应失败
Ours	8.3E4	0.972	0.991	**1.38E5**
LavaHurts 新颖性				
PPO	2.08E6	0.992	0.971	2.08E6
DreamerV2	1.05E6	0.992	0.968	7.56E8
Ours	1.07E6	0.992	0.972	**1.78E6**

表1：三次运行的平均新颖性指标结果。DreamerV2 未能适应 LavaProof 中的新颖性。

适应性能的 95% 时，即视为达到该标准。我们还增加了第三项指标，更新效率，即新颖性出现后达到渐近适应性能所需的策略更新次数，同样是次数越少越好。

我们将 WorldCloner 在出现新颖性后的性能与两个基线进行比较。第一个是采用近端策略优化（PPO）的标准强化学习智能体 [28], ，这是我们 WorldCloner 方法中的神经策略所使用的同一种无模型强化学习方法。第二个基线是 DreamerV2 [14],，这是一种最先进的世界建模智能体，能够学习端到端的神经世界模型。在训练期间，没有向这些智能体提供任何关于新颖性的信息。所有方法最多训练 1000 万个时间步，以确保收敛；每种方法的结果均取三次运行的平均值。在第 50k 个回合时引入新颖性，此时所有智能体早已收敛。由于基线智能体不具备新颖性检测能力，我们在评估期间保持其学习过程开启，使智能体能够在新颖性引入后立即做出反应。

所有智能体的策略架构均采用卷积神经网络特征提取器和两个全连接输出网络，一个用于估计价值，另一个用作智能体的策略函数。DreamerV2 的所有超参数和架构均与原始论文保持一致 [14], ，而 PPO 的所有超参数均与 MiniGrid 推荐的超参数一致 [34]。对于 WorldCloner，我们采用 60% 真实轨迹采样与 40% 想象轨迹采样的混合比例。该比例是通过实证权衡渐近适应性能与自适应效率确定的。想象占比更高时，智能体无法恢复到新颖性出现后的完整性能。WorldCloner 世界模型使用 MiniGrid 的符号特征进行规则学习，包括智能体和物体的对象类型、颜色和位置，智能体朝向、智能体的物品栏，以及门是锁定、未锁定还是打开状态。

5 结果

我们在表1中记录了这些评估结果，并在图6中进一步展示了我们方法的适应过程。表中显示，正如预期，在新颖性出现之前，三种方法在所有三种新颖性情境下都收敛到几乎相同的性能。这意味着所有方法都能在所有环境中找到长度相同、通往目标的解决方案序列。

对于 DoorKeyChange 的“变化”新颖性，DreamerV2 的自适应效率略高于 WorldCloner，而两者的表现都显著优于 PPO。从这一结果可以看出，想象对新颖性出现后的适应大有裨益。然而，自适应效率并不能说明全部情况。DreamerV2 仅利用想象来更新策略，这意味着每次更新世界模型时，DreamerV2 都必须通过多轮想象迭代来更新策略。因此，DreamerV2 的更新效率要求的策略更新次数几乎比 WorldCloner 多两个数量级。

对于 LavaProof 的“捷径”新颖性，WorldCloner 的自适应效率显著优于 PPO，而 DreamerV2 始终无法发现这一捷径新颖性。Dreamer 在新颖性出现后的环境中进行了 250 万步训练，并在多次运行中得到相同结果。我们认为，DreamerV2 失败的原因在于其策略学习器依赖世界模型的准确性这一独特机制。遗憾的是，世界模型仍然会预测熔岩会造成负面后果。由于较长的前新颖性解法仍能抵达目标，Dreamer 的世界模型因此保持不变，想象过程也始终没有变化，策略也就不会更新。因此，如果策略每次都生成相同的动作序列，世界模型就很容易过拟合。PPO 和 WorldCloner 不会遇到这个问题，因为策略会出现微小变化，从而触发策略更新，使其能够更敏锐地应对捷径新颖性。不过，这些方法也并非完美，如图6所示。对于捷径新颖性注入，两种方法都需要超过 25,000 个环境步才能作出反应；而对于屏障和增量新颖性则不然。这表明捷径新颖性带来了独特挑战：当新颖性出现前的最优路径仍然存在时，已经收敛且不再探索的智能体将难以察觉新颖性。正如 [4], 所指出的，未来仍需研究新颖性感知探索技术。

最后，LavaHurts 中“屏障”新颖性与 DoorKeyChange 新颖性相似：WorldCloner 和 DreamerV2 的自适应效率非常接近，且都大幅优于 PPO，但在所有情况下都需要很长时间才能收敛。此外，在 LavaHurts 和 DoorKeyChange 新颖性中，DreamerV2 和 WorldCloner 都表现出较高的方差，这表明在某些试验中，这两种方法都在适应过程的早期找到了新解决方案，但未能稳定地收敛到该方案。迁移学习告诉我们，这两项结果都是预料之中的，因为

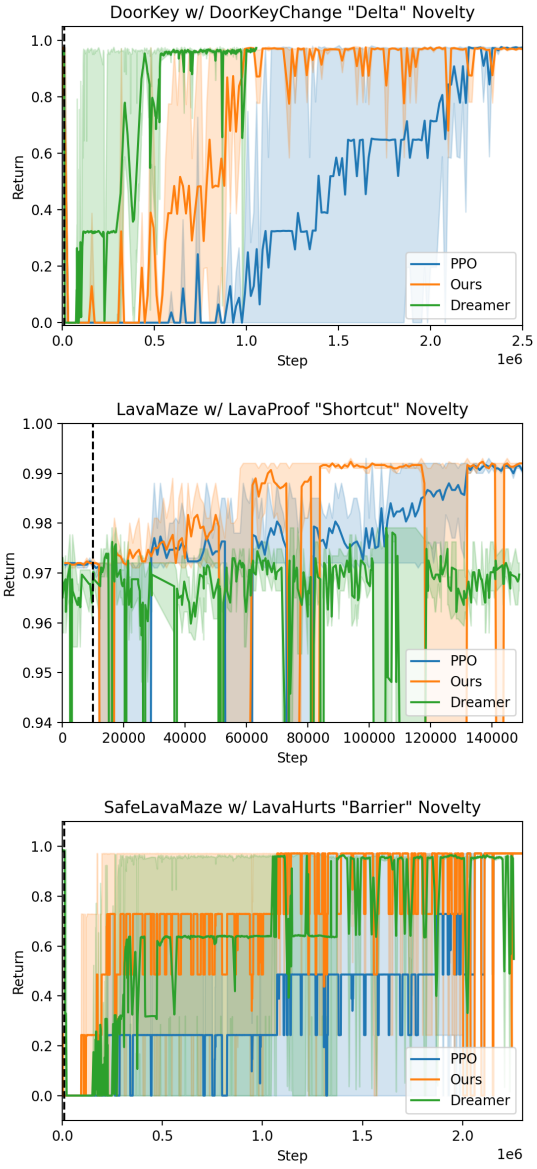

图6：这些曲线展示了新颖性出现后的自适应性能。每种方法都在环境步数达到10,000时遭遇新颖性注入（以垂直黑色虚线标示）。上图展示了DoorKey环境中DoorKeyChange“变化”新颖性的响应。中图展示了LavaShortcutMaze环境中LavaProof“捷径”新颖性的响应。注意：Dreamer没有找到捷径。下图展示了LavaShortcutMaze环境中LavaHurts“屏障”新颖性的响应；在新颖性出现之前，熔岩对智能体是安全的。

简单的前新颖性解决方案并未让策略为更复杂的新颖性出现后的环境做好准备。

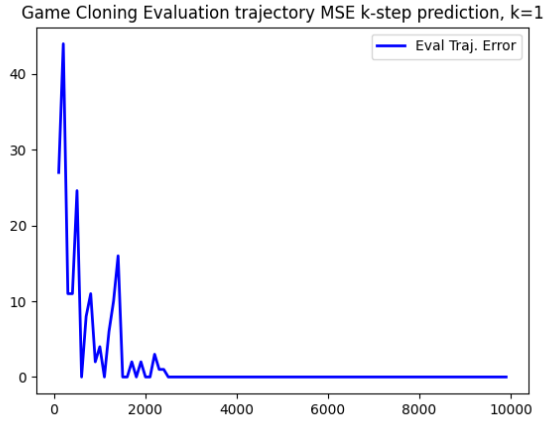

图7：这展示了WorldCloner 在Empty MiniGrid环境中进行规则学习器训练时，1步预测误差随环境步数的变化。

总体而言，与DreamerV2相比，我们的模型能够更高效地利用更新。由于DreamerV2只在想象中训练策略，因此每执行一次世界模型更新，就会在想象中进行数百次策略更新。如表1最后一列所示，通过融合环境交互数据与想象数据，WorldCloner所需的策略更新次数显著减少。

我们还评估了规则学习器的效率和准确性（结果如图7所示）。每经过100个训练步骤，我们就通过运行随机策略1000步来验证模型的准确性，并测量世界模型的平均1步预测误差。前新颖性规则学习器只需约2000个训练步骤，便可收敛到近乎完美的预测准确率。网格世界是简单且确定性的环境，因此这一结果并不令人意外。关键在于，与神经世界模型学习技术相比，它能够快速收敛，并且具有较高的样本效率 [13]。

这些结果表明，仅使用40%的想象，WorldCloner就能在所有新颖性上提升适应效率，相比之下，神经策略没有世界模型。此外，在这些新颖性上，WorldCloner的表现可与DreamerV2相媲美，并且能够通过捷径实现适应，而这些捷径是DreamerV2未能检测到的。最有趣的是，我们可以从表1的最后一列看到，WorldCloner以少于PPO和DreamerV2的策略更新次数取得了这些结果。不过，这将取决于WorldCloner向策略学习器注入的想象量，也应成为未来研究的主题。

6 结论

当自主智能体被部署在开放世界决策情境中时，就需要采用专门设计来处理新颖性的技术。这些新颖性可能包括：学习使用新钥匙开门、发现能够缩短行程时间的新捷径，或避开在以往安全路径上出现的新危险。

为此，我们展示了如何利用符号世界模型，更高效地使强化学习智能体策略适应新颖性。这些符号世界模型（1）能够快速更新；（2）能够模拟轨迹展开，并将其加入策略学习过程，从而减少与新颖性出现后的环境进行直接交互的次数。具体而言，我们的结果表明，WorldCloner 在适应效率方面可与最先进的神经世界建模技术相媲美，同时所需的策略更新次数仅为其一小部分。这表明，对于使强化学习智能体适应新颖性而言，符号表示是神经表示的有效补充。

为了有效实现新颖性适应，未来的技术应考虑如何利用世界模型，以及如何同时学习转移函数和策略模型。我们对基于世界模型、能够适应新颖性的智能体进行的观察表明，探索是高效实现新颖性适应的关键。即使能够使用世界模型，几乎所有强化学习智能体仍假设可以贪心地遵循已收敛的新颖性出现前的策略。这会导致一个问题重重的反馈回路：通过想象训练的策略只能访问世界模型已知的内容，而这些内容可能不包含新颖性；但与此同时，这些策略又决定了与环境的交互。因此，除非新颖性出现在旧的最优策略所经过的路径上，否则智能体可能永远无法遇到这种新颖性，而训练它的世界模型也可能永远无法适应。要在更可信的“开放世界”环境中运作，新的探索技术需要考虑如何利用世界模型识别新颖性，以及如何在这种“新颖性探索”和遵循智能体策略之间进行权衡。

参考文献

[1] Joshua Alspector。2021。《表示编辑距离作为新颖性的度量》。*arXiv* 预印本 *arXiv:2111.02770*（2021）。[2] A Badia、B Piot、S Kapturowski、P Sprechmann、A Vitvitskyi、D Guo 和 C Blun- dell。2020。Agent57：超越人类 Atari 基准。在《第*37*届国际机器学习会议论文集》，线上，*PMLR*，第119卷。2020。[3] Keith Ball 等。1997。《现代凸几何入门》。几何风格 31，1-58（1997），26。[4] Jonathan C Balloch、Julia Kim、Jessica L Inman 和 Mark O Riedl。2022。探索在强化学习任务迁移中的作用。*IROS* 高级认知与推理技能终身学习研讨会。[5] Jonathan C Balloch、Zhiyu Lin、Mustafa Hussain、Aarun Srinivas、Xiangyu Peng、Julia Kim 和 Mark Riedl。2022。NovGrid：用于评估智能体对新颖性响应的灵活网格世界。在*AAAI* 研讨会《为开放世界设计人工智能》论文集。[6] Terrance Boult、Przemyslaw Grabowicz、Derek Prijatelj、Roni Stern、Lawrence Holder、Joshua Alspector、Mohsen M Jafarzadeh、Toqueer Ahmad、Akshay Dhamija、Chunchun Li 等。2021。迈向新颖性形式理论的统一框架。在*AAAI* 人工智能会议论文集，第35卷。15047–15052。[7] Daniel A Braun、Ad Aertsen、Daniel M Wolpert 和 Carsten Mehring。2009。在不可预测的运动任务中学习最优适应策略。《神经科学杂志》29，20（2009），6472–6478。[8] Yining Chen、Haipeng Luo、Tengyu Ma 和 Chicheng Zhang。2021。具有隐藏变化域的主动在线学习。arXiv: 2006 14481 [ ]。cs.LG [9] 2018 Maxime Chevalier-Boisvert、Lucas Willems 和 Suman Pal。。面向 OpenAI Gym 的极简网格世界环境。https://github.com/maximecb/gym- minigrid。[10] Harold Scott Macdonald Coxeter。1973。正多胞体。Courier Corporation。[11] Matthew Guzdial 和 Mark O. Riedl。2017。从视频中进行游戏引擎学习。在*2017*年国际人工智能会议论文集。http: //www.cc.gatech.edu/~riedl/pubs/ijcai17.pdf [12] David Ha 和 Jürgen Schmidhuber。2018。循环世界模型促进策略演化。在《神经信息处理系统进展》，S. Bengio、H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi 和 R. Garnett（编），第31卷。Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf [13] Danijar Hafner、Timothy Lillicrap、Jimmy Ba 和 Mohammad Norouzi。2019。《梦境到控制：通过潜在想象学习行为》。载于《国际学习表征会议》。[14] Danijar Hafner、Timothy P Lillicrap、Mohammad Norouzi 和 Jimmy Ba。2021。《利用离散世界模型精通雅达利游戏》。载于《国际学习表征会议》。[15] Trevor Hastie、Robert Tibshirani、Jerome H Friedman 和 Jerome H Friedman。2009。《统计学习要素：数据挖掘、推断与预测》。第2卷。施普林格。[16] Elad Hazan。2016。在线凸优化导论。《优化基础与趋势》2，3-4（2016），157–325。[17] Matthew Klenk、Wiktor Piotrowski、Roni Stern、Shiwali Mohan 和 Johan de Kleer。2020。《面向开放世界人工智能的基于模型的新颖性适应》。载于《国际诊断原理研讨会》（DX）。[18] Zhixuan Lin、Yi-Fu Wu、Skand Peri、Bofeng Fu、Jindong Jiang 和 Sungjin Ahn。2020。改进以对象为中心的世界模型中的生成式想象。载于《国际机器学习会议》。PMLR，6140–6149。[19] Bryan Loyall、Avi Pfeffer、James Niehaus、Tyler Mayer、Paola Rizzo、Alex Gee、Sanja Cvijic、William Manning、Mary Kate Skitka、McCoy Becker 等。2022。利用概率编程和新颖性感知规划实现开放世界中的新颖性在线适应的集成架构。载于AAAI 研讨会《为开放世界设计人工智能》论文集。[20] Michael McCloskey 和 Neal J Cohen。1989。联结主义网络中的灾难性干扰：序列学习问题。载于《学习与动机心理学》。第24卷。爱思唯尔，109–165。[21] Drew McDermott。2000。1998年人工智能规划系统竞赛。《AI Mag.》21（2000），35–55。[22] Xiangyu Peng、Jonathan C Balloch 和 Mark O Riedl。2021。《游戏中的新颖性检测与适应》。载于AAAI游戏强化学习研讨会。[23] Marco AF Pimentel、David A Clifton、Lei Clifton 和 Lionel Tarassenko。2014。新颖性检测综述。《信号处理》99（2014），215–249。[24] Arnu Pretorius、Scott Cameron、Andries Petrus Smit、Elan van Biljon、Lawrence Francis、Femi Azeez、Alexandre Laterre 和 Karim Beguir。2020。通过想象与基于模型的深度多智能体强化学习学习通信。[25] Sébastien Racanière、Theophane Weber、David Reichert、Lars Buesing、Arthur Guez、Danilo Jimenez Rezende、Adrià Puigdomènech Badia、Oriol Vinyals、Nico- las Heess、Yujia Li、Razvan Pascanu、Peter Battaglia、Demis Hassabis、David Silver 和 Daan Wierstra。2017。《用于深度强化学习的想象增强型智能体》。载于《神经信息处理系统进展》第30卷。神经信息处理系统基金会（NIPS），5690–5701。http://papers.nips.cc/paper/7152-imagination-augmented-agents-for-deep-reinforcement-learning.pdf[26] Vasanth Sarathy、Daniel Kasenberg、Shivam Goel、Jivko Sinapov 和 Matthias Scheutz。2021。SPOTTER：通过目标强化学习扩展符号规划算子。载于《第20届国际自主智能体与多智能体系统会议论文集》。1118–1126。[27] Jürgen Schmidhuber。1990。反应式环境中的动态强化学习与规划在线算法。载于1990 IJCNN国际神经网络联合会议。电气电子工程师学会，253–258。[28] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017。近端策略优化算法。arXiv 预印本 arXiv:1707.06347（2017）。[29] Shai Shalev-Shwartz 等。2012。在线学习与在线凸优化。《机器学习基础与趋势》4，2（2012），107–194。[30] David Silver、Julian Schrittwieser、Karen Simonyan、Ioannis Antonoglou、Aja Huang、Arthur Guez、Thomas Hubert、Lucas Baker、Matthew Lai、Adrian Bolton 等。2017 550，7676 2017 354 359。无需人类知识精通围棋。《自然》（），–。[31] Daniel L Silver、Qiang Yang 和 Lianghao Li。2013。终身机器学习系统：超越学习算法。载于2013 AAAI春季研讨会系列。[32] Richard S Sutton 和 Andrew G Barto。2018。强化学习：导论。麻省理工学院出版社。[33] Oriol Vinyals、Igor Babuschkin、Wojciech M Czarnecki、Michaël Mathieu、Andrew Dudzik、Junyoung Chung、David H Choi、Richard Powell、Timo Ewalds、Petko Georgiev 等。2019。利用多智能体强化学习达到《星际争霸II》宗师级水平。《自然》575，7782（2019），350–354。[34] Lucas Willems。2020。lcswillems/torch-ac：深度强化学习演员-评论家算法 A2C 和 PPO 的循环式多进程 PyTorch 实现。https://github.com/lcswillems/torch-ac [35] Zhuangdi Zhu、Kaixiang Lin 和 Jiayu Zhou。2021。《深度强化学习中的迁移学习：综述》。arXiv:2009.07888 [cs.LG]