arXiv:2302.04250v2 [cs.LG] 4 May 2023

学习如何推断部分马尔可夫决策过程，以实现上下文内适应与探索

Chentian Jiang* 信息学，爱丁堡大学 爱丁堡，英国
chentian.jiang@ed.ac.uk

Nan Rosemary Ke
DeepMind 伦敦，英国
nke@deepmind.com

Hado van Hasselt
DeepMind 伦敦，英国
hado@deepmind.com

摘要

为了在不同任务之间实现泛化，智能体应从过去的任务中获取知识，以促进其在未来任务中的适应与探索。我们关注上下文内 适应与探索问题，在这一问题中，智能体仅依赖上下文，即状态、动作和/或奖励的历史记录，而非基于梯度的更新。后验采样（汤普森采样的扩展）是一种很有前景的方法，但它需要贝叶斯推断和动态规划，而这些过程往往涉及未知因素（例如先验分布）和高昂的计算成本。为应对这些困难，我们使用 Transformer 从训练任务中学习推断过程，并考虑由部分模型构成的假设空间。这些小型图结构表示马尔可夫决策过程，并支持低成本的动态规划。在符号炼金术基准的一个版本中，我们的方法在适应速度以及探索与利用之间的平衡方面接近精确后验采样预言机的表现。我们还表明，尽管部分模型排除了环境中的相关信息，但仍能导出良好的策略。

关键词

深度强化学习、基于模型、探索、适应、泛化、上下文内学习、汤普森采样

1 引言

为了实现良好的泛化，智能体应从过去的任务中获取知识，以帮助其在未来的任务中有效探索和适应。在深度强化学习（RL）中，人们对上下文内泛化的兴趣日益浓厚：进入新任务后，智能体的神经网络权重保持冻结，其策略仅根据实时上下文（即状态、动作和/或奖励的历史记录）进行探索和适应 [例如 3, 12, 33]。

受人类行为的启发 [例如， 25, 27], 上下文内泛化可以通过一种假设检验来实现：智能体可以从过去的任务中获取先验信念，了解任务通常如何运作，例如，物体之间交互通常受哪些动力学规律支配？面对一项陌生（但相关）的任务时，它可以定期从自己的信念中抽取一个有希望的假设，并依据该假设采取最优动作来检验它。由此产生的数据，即上下文，可以帮助智能体判断假设是否正确，并进一步完善其更广泛的后验信念。通过抽取并检验许多假设，智能体会尝试采取多样化的动作来探索新任务，并获取信息以调整其后验信念，使之更贴近任务的真实运作方式。随着智能体依据更准确的假设采取最优动作，其策略也会随之调整。

这一假设检验概念由一种称为后验采样 [20, 28], 的计算框架巧妙地概括；它将汤普森采样 [30] 从多臂老虎机扩展到了强化学习。然而，后验采样需要贝叶斯推断和动态规划，除玩具问题外，这些方法都难以处理，而且可能需要未知信息（例如先验分布）。即使通过神经网络近似后验采样来应对这些挑战 [，例如 1, 5, 13, 18, 19]„ 这些方法通常也是为在同一任务上进行训练和评估而设计，而非为了跨任务泛化。

我们用于近似后验采样的方法解决了上述所有难题：我们近似推断，而非依赖贝叶斯推断；并且针对由部分模型构成的假设空间进行近似——这些模型是以图表示的小型马尔可夫决策过程（MDP），适合通过动态规划高效求解。此外，我们将近似推断过程视为可泛化的知识，先从训练任务中学习（使用神经网络），再应用于留出测试任务。

为了说明什么是部分模型，可以想象桌上有一个空酒杯。再考虑一个小型马尔可夫决策过程模型，其动态仅捕捉到这样一点：推动酒杯会以很高的概率使酒杯破碎。这个模型是不完整的，因为它排除了用于预测酒杯是否会破碎的其他相关信息，例如酒杯在桌上的位置。然而，由于酒杯通常呈圆形，我们可能会认为，无论它位于桌上的什么位置，都很可能滚落桌面并摔碎。通过对我们的模型应用动态规划，我们可能无法找到能够考虑所有相关信息的最优策略，但仍然可以找到一个不错的策略，避免推动酒杯，并且计算速度更快。

为了学习部分模型的推断过程，我们的方法训练了一个 Transformer [31]，因为 Transformer 此前已展现出近似贝叶斯推断 [15, 35]并进行上下文内泛化的能力 [2, 6]。我们基于这样一种架构展开工作：该架构接收整个数据集——或者在我们的工作中接收完整的上下文/历史——而非单个数据点 [7, 10, 11, 15]。给定上下文后，我们的 Transformer 输出部分模型上的一个分布，并将其表示为小型图。在测试时，我们冻结 Transformer 的权重，并将其作为近似推理机嵌入后验采样框架中。

我们在符号炼金术基准的简化版本 [32] 上评估了我们的方法。该基准提供了一组任务分布，用于研究我们的泛化问题，即在一些任务上进行训练

*本研究是在 DeepMind 实习期间完成的。本研究也收录于 ICLR 2023 的强化学习再世化研讨会论文集。

神经符号人工智能智能体与多智能体系统研讨会论文集（*NeSyMAS 2023*），*Belle*、*Fisher*、*Huang*、*Mansouri*、*Mero˜no-Pe˜nuela*、*Natarajan*、*Tsamoura*（编），*2023* 年 *5* 月 *29—30* 日，英国伦敦。2023 年。

（这些任务具有不同的转移函数），并在留出任务上测量上下文内适应和探索性能。不同于大多数其他基准测试，Alchemy 为任务提供了良好的部分表示，本质上是为每个任务关联一个真实部分模型。该真实值被用作训练 Transformer 的监督信号，使我们能够暂时搁置部分模型应如何表示这一问题（例如，环境中的哪些内容应纳入或排除在部分模型的状态空间之外？）。相反，我们专注于学习如何推断训练任务中的部分模型，并测试在后验采样框架中使用部分模型是否具有价值。无论是在上下文内适应速度还是探索-利用平衡方面，我们的方法都接近于精确后验采样预言机的表现。我们还证实，部分模型能够促进向近最优奖励的适应。

2 问题设定

2.1 任务分布

环境的马尔可夫决策过程模型包含状态 $s \in \mathcal{S}$、动作 $a \in \mathcal{A}(s)$、转移概率函数 $p(s'|s,a)$、奖励函数 $r(s,a)$、折扣因子和初始状态分布。我们研究的设定包含一个任务分布 $p(k)$（以马尔可夫决策过程表示），其中转移函数和奖励函数均为确定性的，且可能因任务而异。通过从一组训练任务 $k\text{train} \sim p(k)$ 中学习，智能体应掌握可能的转移函数和奖励函数的总体结构，并能在测试任务 $k\text{test} \sim p(k)$ 上良好地泛化。我们将测试任务留出，使其在训练期间不会被遇到，并根据智能体在这些测试任务中进行上下文内适应和探索的表现来评估泛化能力。我们的问题设定与若干元强化学习研究类似，相关讨论见第 7 节。

2.2 定义上下文内适应与探索

我们针对本研究的目的定义适应和探索（但不声称这些定义涵盖相关研究）。我们将适应视为这样一个过程：(1) 推断出能够更准确描述环境某一部分的部分模型；(2) 产生更优的策略，以奖励作为衡量标准。重要的是，这种适应应当在上下文内发生；这里的上下文指动作和状态的历史记录：在测试时，随着条件化于更长的上下文（即更多状态和动作），模型和策略应有所改进——这类似于在贝叶斯推断中条件化于更多数据，而不是使用基于梯度的更新。我们将有意义的探索视为收集新的上下文，以获取有助于推断部分模型的信息，也就是区分不同的部分模型。作为近似，我们通过访问过的部分模型空间中的不同状态-动作对数量来衡量。

3 学习如何推断部分模型

我们的方法从训练任务中学习一般知识，其形式是对部分模型进行推断的过程：给定上下文（状态和动作的历史记录），最能描述当前任务的部分转移函数和部分奖励函数的分布是什么？

3.1 部分马尔可夫决策过程

我们的部分马尔可夫决策过程模型仅描述环境的一部分：它们将完整状态空间 $\mathcal{S}$ 映射到更小的空间 $\mathcal{V}$（多对一），并在该空间上定义更简单的转移函数和奖励函数。我们将部分模型表示为图：状态是顶点 $v \in \mathcal{V}$，动作 $x \in \mathcal{X}(v)$ 是边遍历，转移函数表示为一个向量，其中包含每条边存在 $\mathbf{e} \in [0,1]^N$ 的概率；奖励函数则表示为一个向量，其中包含每个顶点-边组合获得正（=1）奖励 $\mathbf{r} \in [0,1]^M$ 的概率（否则，顶点-边组合的奖励为零）。

3.2 近似分布

不同于通常将转移函数和奖励函数表示为神经网络的做法 [（例如，8, 24],），我们分别将它们表示为边概率向量 $\mathbf{e}$ 和奖励概率向量 $\mathbf{r}$，它们是神经网络的输出，而非网络本身。我们的表示方法使本方法能够专注于学习一个小型图，而非环境动态中的所有细节。

我们的方法学习对推断过程 $p(\mathbf{e}, \mathbf{r}|\mathbf{C}_{1:t})$ 的近似。$\mathbf{e}$ 是部分模型图中的边概率向量，$\mathbf{r}$ 是奖励概率向量，而 $\mathbf{C}_{1:t}$ 是一个包含上下文的矩阵，即智能体截至时间 t 的状态与动作历史（s，a）。我们使用 Transformer f_θ 来定义该分布：

$$\hat{p}_\theta(\mathbf{e}, \mathbf{r}|\mathbf{C}_{1:t}) = \prod_n \mathrm{Ber}(e_n | f_{\theta,n}(\mathbf{C}_{1:t})) \times \prod_m \mathrm{Ber}(r_m | f_{\theta,m}(\mathbf{C}_{1:t}))$$

$e_n \in \{1, 0\}$ 是由 n 枚举的边概率，而 $r_m \in \{1, 0\}$ 是由 m 枚举的奖励概率。Ber 是伯努利分布，其概率参数是 Transformer 的 Sigmoid 函数输出，对应第 n 条边 $f_{\theta,\ n}(\mathbf{C}_{1:t})$，或第 m 个奖励位置（即第 m 个顶点-边）$f_{\theta,\ m}(\mathbf{C}_{1:t})$。为简单起见，我们假设所有伯努利分布彼此独立。

我们的 Transformer 还会输出一个 Softmax 函数，用于定义顶点上的分类分布 $\hat{}\ p_\theta(v_t|s_t)$（我们的 Transformer 接收整个上下文，但它应该学会 v_t 只取决于最近的状态 s_t）。

3.3 训练

对于每个训练任务 k，我们假设可以获得其转移函数部分表示的监督信号（以星号标记）$\mathbf{e}_k^*$ 和奖励函数 $\mathbf{r}_k^*$。对于任务中遇到的每个状态 s_t，我们假设可以获得该状态在部分模型空间中对应的顶点 v_t^*。这些监督信号为我们提供了良好的部分模型表示，使我们能够转而专注于：(1) 学习针对这类表示的推断过程 $\hat{}\ p_\theta(\cdot|\mathbf{C})$；(2) 测试尽管这些表示并非完整的马尔可夫决策过程，它们是否仍适合用于后验采样。我们的 Transformer 通过自回归地最小化以下目标进行训练

表 1：Transformer超参数

超参数	值
嵌入维度	32
编码器层数	4
注意力头数	4
编码器隐藏层大小	32
优化器	Adam（初始学习率：1e-4）

以下损失函数：

$$\mathcal{L}(\theta) = \mathbb{E}_{p(\mathbf{C}_{1:T},k)}\Bigg[\frac{1}{T}\sum_{t=1}^{T}\Big[-\ln(\hat{p}_\theta(\mathbf{e}_k^*,\mathbf{r}_k^*|\mathbf{C}_{1:t})) - \ln(\hat{p}_\theta(v_t^*|s_t))\Big]\Bigg]$$

其中，$p(\mathbf{C}_{1:T},k)$ 是一个联合分布，其分布对象为上下文（每个上下文包含 T 个时间步，即一个回合的长度）和任务。

我们生成离线训练数据，以表示来自$p(\mathbf{C}_{1:T},k)$的样本：对于每个训练任务 k，我们使用随机策略生成 M回合长度上下文 $\mathbf{C}_{1:T}$（即状态和动作的历史记录）。每个上下文都与相应任务 $\{\mathbf{e}_k^*,\mathbf{r}_k^*\}$ 的监督信号相关联，并与关于上下文中每个状态如何映射到顶点 v_t^* 的监督信号相关联。

3.4 Transformer架构与超参数

我们的 Transformer 架构基于过去几项研究中的思想，这些研究将整个数据集而非单个数据点作为输入 [7, 10, 11, 15]。在我们的情况下，我们使用整个上下文矩阵，而非单个状态-动作。我们的架构将在以下段落中介绍，经过调优的超参数汇总于表 1 中（调优使用第 5 节所述的验证任务完成）。

输入。我们的 Transformer 接收一个大小为 $T\times X$ 的上下文矩阵 $\mathbf{C}$，其中 T 是一个回合中的时间步数量，X 是经过独热编码的状态-动作中的变量数量（a_t，s_{t+1}）。

嵌入和位置编码。每个（a_t，s_{t+1}）行都通过多层感知机（MLP）嵌入为一个向量。随后沿时间步维度应用位置编码。

层。每一层都是一个自回归 Transformer 编码器层，由自回归多头自注意力机制（沿时间步维度运行）和一个全连接前馈网络组成。

输出。编码器层的输出会传递给多层感知机，以预测大小为 $T\times Y$ 的结果，其中 Y 是$\mathbf{e}$的长度加$\mathbf{r}$的长度，再加上可能顶点的数量。对于每个时间步 t，多层感知机的预测结果会分别经过多个 Sigmoid 函数和一个 Softmax 函数，从而得到$\mathbf{e}$和$\mathbf{r}$的伯努利参数，以及顶点的分类分布。

算法1 后验采样

要求： 训练好的 Transformer 参数 θ
  初始化空的上下文矩阵$\mathbf{C}$
  对于 时间步 t，在测试回合中 **执行**
    如果 采用完整方法**则**
      采样 $\mathbf{e},\mathbf{r} \sim \hat{} p_\theta(\mathbf{e},\mathbf{r}|\mathbf{C})$
    否则如果 消融**则**
      $\mathbf{e} \leftarrow \mathbb{E}\hat{}_{p_\theta(\mathbf{e}|\mathbf{C})}[\mathbf{e}]$; $\mathbf{r} \leftarrow \mathbb{E}\hat{}_{p_\theta(\mathbf{r}|\mathbf{C})}[\mathbf{r}]$
    结束如果
    使用 $\mathbf{e}$ 和 $\mathbf{r}$ 进行值迭代（转移函数和奖励函数在部分模型空间中）产生 π：$\mathcal{V}\to\mathcal{X}$
    $v_t \leftarrow \arg\max_{v_t} \hat{} p_\theta(v_t|s_t)$
    $x_t \leftarrow \pi(v_t)$
    将 x_t 映射为 a_t（真实环境中的动作）；观察 s_{t+1}
    将 $(a_t,\ s_{t+1})$ 作为一行追加到 $\mathbf{C}$
  结束循环

4 后验采样

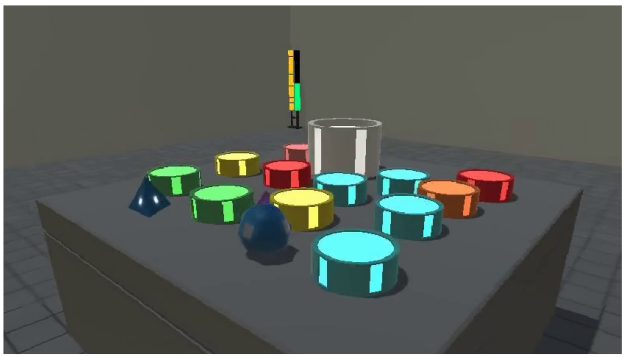
在测试时，我们的 Transformer 权重 θ 保持冻结。与离线训练不同，此时它会在后验采样框架（算法1）中进行在线 推断。

作为一种通用框架，后验采样会为环境可能的模型设定一个先验分布。随后，它从该分布中采样一个模型，并使用动态规划求解该模型的最优策略。该策略会执行若干时间步（在我们的工作中为 1 个时间步），从而生成用于更新模型后验分布的数据。接着，从后验分布中采样下一个模型，并重复这一过程。

在我们的工作中，我们近似推断过程，并使用部分模型而非完整模型。我们用 $\hat{} p_\theta(\mathbf{e},\ \mathbf{r}|\mathbf{C}_0)$ 替代先验，其中$\mathbf{C}_0$ 为空上下文。给定新数据，即状态和动作，我们将其附加到上下文$\mathbf{C}_{1:t}$，并用 $\hat{} p_\theta(\mathbf{e},\ \mathbf{r}|\mathbf{C}_{1:t})$ 替代后验更新。随后采样一个确定性部分模型，即$\mathbf{e}$ 和$\mathbf{r}$ 仅包含 1/0 概率的模型，并通过动态规划求解，尤其是使用值迭代，从而在部分模型空间中生成一个策略 π。

我们使用 Transformer 给出的当前顶点预测 v_t，在部分模型空间中查询一个动作 $x_t \leftarrow \pi(v_t)$。最后，我们假设部分模型动作 $\mathcal{X}$（边遍历）与环境中的真实动作之间存在已知的一一映射 $\mathcal{A}$。将 x_t 映射为 a_t 后，我们的方法在真实环境中执行 a_t，并观测下一个状态 s_{t+1}。然后将 $(a_t,\ s_{t+1})$ 附加到上下文$\mathbf{C}_{1:t+1}$中。

与最初提出的后验采样方案 [20, 28],不同，我们的方法在每个时间步都采样一个模型，而不是间隔较长时间（例如一个回合）才采样一次。在某些设置中，在采样模型下行动较长时间，可能会产生更多信息丰富的数据，有助于推理机区分不同模型 [22]。然而，在我们的环境中（第5节），即时的状态-动作数据对于区分部分模型可能极具信息量，因此我们倾向于在每个时间步更新部分模型信念并进行采样。若要将我们的方法改为采用最初的后验采样设置，只需延长采样间隔即可。



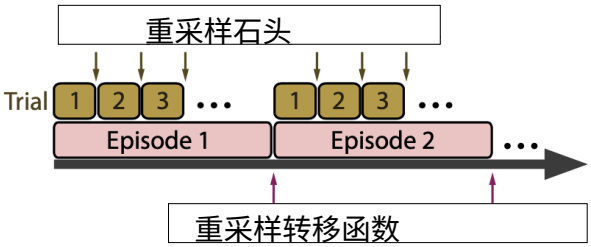


图1：Alchemy环境（改编自Alchemy论文中的图1 [32]）。上：石头（球体和金字塔物体）、药剂和坩埚（灰色圆柱体）的示意图。在我们的工作中，智能体观测到的是 Alchemy 的符号化（而非 3D）版本，其中只有1块石头和6瓶药剂。智能体的目标是将石头浸入药剂中，把它转化为奖励更高的石头。下：回合结构。

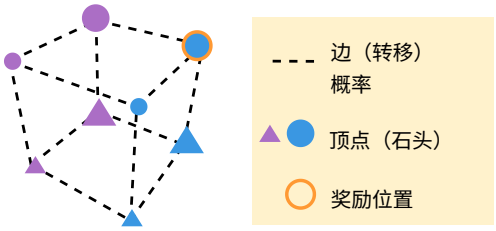


图2：我们简化炼金术环境的部分模型（也称为“立方体”）。

4.1 消融

我们构建了一种消融实验，在后验采样过程（算法1）中移除采样环节，从而测试采样的有无对适应和探索的影响。我们不再进行采样，而是使用e 和r的期望值，它们是Transformer直接输出的结果（即每条边的伯努利分布参数，以及每个奖励位置的伯努利分布参数）。这些期望值现在表示随机的 转移函数和奖励函数（而非通过采样得到的确定性函数）：边遍历以某个0到1之间的概率成功，顶点-边组合以某个0到1之间的概率产生正奖励。

5 实验设置

5.1 简化符号化 Alchemy环境

我们聚焦于符号炼金术基准 [32]（图1上方），因为它（1）提供任务的部分模型表示，可用于训练我们的 Transformer；（2）能够评估我们的方法在留出任务中的适应和探索效果。完整的 Symbolic Alchemy 环境是一个试验平台，可用于研究一系列问题，例如学习如何对多个对象采取行动，以及学习从潜在状态到感知观测的不同映射。我们选择跨任务泛化这一问题，并为简化起见，只关注奖励函数固定、转移函数各异的任务。我们对 Symbolic Alchemy 进行了简化，以便单独研究这一问题。

在我们的环境中，智能体会收到关于1块石头和6瓶药剂的符号化观测。它可以将石头浸入药剂中；药剂会因此耗尽，并可能改变石头的颜色、形状或大小。6瓶药剂各自具有独特效果，作用于一种特征（例如大小）并引发一种变化（即增加或减少）。石头特征的不同组合对应不同的奖励（-3、-1、1或15），智能体的目标是将初始石头特征转化为奖励更高的特征。石头的亮度值越高，表示奖励越高（图1中的小块区域）。要实际获得石头的奖励，智能体必须将石头放入坩埚中（图*1*中的圆柱形容器）。智能体必须在一场试验结束前完成此操作；每场试验包含10个时间步（每个回合有20场试验）。石头放入坩埚后便会消失，智能体必须等待下一场试验，才能获得一块特征随机采样的新石头。

5.1.1 部分模型。 我们的环境为每项任务关联一个部分模型，其中每项任务都有唯一的部分转移函数。这些模型以受约束图的形式表示，称为“立方体”（图2），其设计很大程度上基于 Alchemy 原有的部分模型表示。每个立方体有 8 个顶点和 12 条边；这些边只能沿着三维空间中绘出的立方体轮廓存在，其中每个维度代表石头的一种特征（颜色、形状、大小）。顶点表示单块石头可能具有的特征组合，且只有一个顶点与正奖励相关联，代表若放入坩埚便能获得最高奖励（15）的石头。边的存在表示石头是否可以在两个顶点（即两组特征）之间转换。如果存在一条边，石头就可以通过对某种药水执行动作来转换，前提是药水并非空的。对药水执行动作，表示在三维立方体空间中沿某个维度和方向移动（例如，黄色药水会使石头变大，即对应大小维度的递增方向）。由此，立方体表示一个小型马尔可夫决策过程，其中顶点是状态，边遍历是动作，与正奖励相关联的顶点表示奖励函数，而边的存在则表示转移函数。

立方体所表示的马尔可夫决策过程是任务的部分模型，因为它省略了一些信息。例如，它无法体现

药剂在使用后会变空（之后便无法再转化石头），也无法体现某些石头虽然不是最高奖励，却能带来少量正奖励（+1）。我们的方法在这类部分模型上执行动态规划（即值迭代），也就是寻找从当前顶点／石头到奖励最高的顶点／石头的最短路径。由此得到的策略可能并非最优；它可能会尝试使用已空的药剂，也可能不会把奖励较低（但为正）的石头放入坩埚（而是陷入零奖励的境地，因为剩余的药剂无法制造出奖励最高的石头）。尽管如此，我们认为立方体是有用的部分模型，因为它们包含了完整环境中转移的关键信息。此外，动态规划可能会产生这样的策略：每种药剂最多只需使用一次，就能获得奖励最高的石头，从而实现最优试验奖励，而无需了解药剂变空或低奖励石头的信息。因此，我们假设，在这些部分模型上执行动态规划，往往能产生带来高奖励的策略。

Alchemy 让研究人员能够访问其任务所依托的立方体，从而使我们能够选取一个真实部分模型，根据该模型生成任务，并使用随机策略收集轨迹数据。真实部分模型和轨迹数据分别成为训练我们的 Transformer 的监督信号和上下文。在 109 个可能的立方体中，我们使用 88 个进行训练、11 个进行验证（即用于选择超参数），以及 10 个进行测试。

5.1.2 评估。 我们的回合分为 20 次试验，每次试验包含 10 个时间步（图1底部）。在一个回合内，真实部分模型保持不变。同一回合中的各次试验之间，石头会重置为一组采样得到的特征（相当于在立方体上采样一个顶点），所有药剂也会重新恢复为满载状态。不同回合之间，真实部分模型中边的有无会发生变化，这意味着决定药剂如何改变石头的转移也会随之改变。为了获得高奖励，智能体必须能够很好地泛化到新的回合：探索新的石头与药水之间的转移，并调整策略，使其朝奖励最高的石头转移。

我们在评估中使用一个包含 10 个 Alchemy 立方体的留出集。对于每个立方体，我们生成 20 个回合，每个回合中的石头重置序列都通过不同方式采样。我们在单个回合（每回合 $= 200$ 个时间步）内评估我们的方法，每个回合进行 20 次试验。在每个评估回合开始时，我们的方法和消融版本都使用相同的已训练 Transformer，并且上下文为空。下文介绍的所有指标，均报告为 10×20=200 个评估回合的平均值。

我们将策略适应度量为智能体每次试验获得的最优奖励百分比。原始奖励除以通过求解（使用值迭代）真实且完整的马尔可夫决策过程得到的最优策略所获得的奖励。如果根据试验开始时的石头和药剂能够制成最佳石头，那么最优策略在该次试验中的奖励为 15，否则为 1。我们还将模型适应度量为以下两者之间的平均绝对误差（MAE）：边存在的预测概率（即在完整方法或消融实验中由我们的 Transformer 返回的伯努利参数），以及 Alchemy 立方体中边实际存在与否（概率为 1/0）。最后，我们将探索量度量为在部分模型空间中访问过的不同状态-动作对数量。我们选择这一指标，是因为与完整环境中的状态-动作对相比，它更能体现智能体为改进其部分模型信念而收集有用信息的程度。

5.2 在 Alchemy 中实现我们的方法

回想一下，对于我们的环境，部分模型图被限制为一个包含 8 个顶点和 12 条边的立方体（图2）。顶点仅表示 3 种石头特征（颜色、大小、形状），边仅表示是否可以使用某种药剂转换石头的某种特征（假设药剂始终装满）。这些部分模型的奖励函数表示 $\mathbf{r}$ 仅为一个顶点赋予正奖励（=1），用于编码将哪块石头放入坩埚时奖励最高（所有其他顶点的奖励均为零）。转移函数表示为一个大小为 12 的向量，其中包含 12 条边 $\mathbf{e} \in [0,1]^{12}$的概率。

此处上下文 $\mathbf{C}_{1:t}$是智能体截至时间 t的状态-动作历史矩阵。状态 s由以下向量表示：石头的 4 个二元指标（颜色、大小、形状、是否在坩埚中）、指示每种药剂是否为空的 6 个二元指标，以及一个表示石头亮度的 4 类变量。动作 a是 1-8 的整数，分别表示空操作、将石头放入坩埚，以及使用 6 种药剂中的一种。

我们针对 Alchemy 环境实现了我们的方法，这意味着我们的 Transformer 经过训练，可以根据智能体与石头和药剂交互的上下文/历史$\mathbf{e}$推断立方体边的概率$\mathbf{C}$。由于奖励函数是固定的，我们不使用 Transformer 学习$\mathbf{r}$，而是将其提供给值迭代过程。在测试时，Transformer 推断出的边概率通过值迭代得出一个策略，将顶点映射到边遍历，或直接映射到空操作/“放入坩埚”动作。最后，边遍历会被映射回完整环境中应尝试的药剂。

我们在 Alchemy环境中生成离线训练数据：对于 88 个真实值训练立方体（每个立方体代表一个不同的任务），我们使用随机策略为每个立方体生成 500 个回合长度上下文 $\mathbf{C}_{1:T}$，其中 $T = 200$ 是一个回合中的时间步数。每个上下文都与相应任务 $\mathbf{e}_k^*$的监督信号相关联，即真实的立方体边，并且与有关上下文中每个状态如何映射到一个顶点 v_t^*的监督信号相关联，即该状态下真实的石头（颜色、大小、形状）。11 个真实值立方体用于验证，即选择 Transformer 的超参数；最后 10 个立方体（总共 109 个 Alchemy 立方体中的）用于生成在线测试回合，以评估我们的方法。

5.3 参考点比较

我们与 2 个预言机进行比较：精确后验采样 表示一种基于部分模型进行后验采样的精确实现。与我们的方法不同，它已知包含 109 种可能 Alchemy 立方体（包括测试立方体）的假设空间，并且知道如何在该空间上进行完美的贝叶斯推理（从均匀先验开始）。它代表了适应速度的上界，以及后验采样框架中理想的探索—利用平衡。真实部分模型 表示后验分布已收敛至真实部分模型时的后验采样，为奖励（使用部分模型可达到的奖励）和部分模型准确率设定了固定上界。

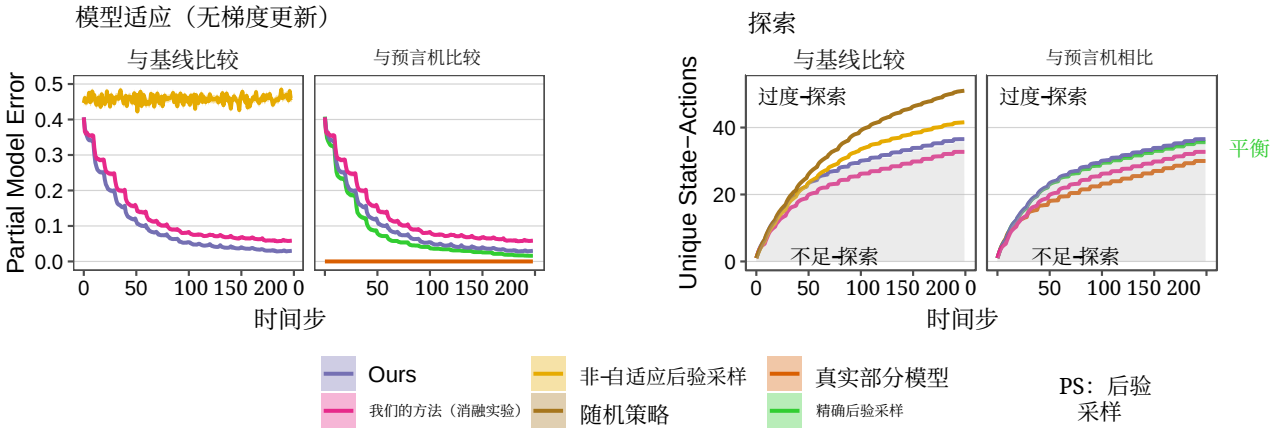


图 3：留出测试任务中的模型适应与探索。所有曲线均为留出测试回合的平均值，阴影带表示标准误差（有时不可见）。左：我们的方法适应其部分模型推断的速度和准确度接近精确后验采样预言机（精确后验采样）。右：探索量以在部分模型空间内访问过的不同状态-动作对的累计数量衡量。我们的方法实现了接近 精确后验采样 **（绿色曲线）的探索-利用平衡。我们的方法在**上下文内**完成适应与探索，无需基于梯度的更新。**

我们还与 2 个基线进行比较：非自适应后验采样 表示使用正确假设空间（109 种可能的 Alchemy 立方体）进行后验采样，但不进行推理，即始终从部分模型中均匀采样。随机策略 完全不考虑模型，而是均匀采样动作。它们体现了在部分模型信念不适应的情况下所能获得的奖励和探索量。

6 实验

6.1 模型适应

我们的方法能够在上下文内适应其推断出的部分模型，速度和准确度几乎与精确实现的后验采样（精确后验采样）相当（图 3；附录 A 中使用另外两个训练随机种子进行了定性复现）。这些结果表明，我们的 Transformer 在更广泛的部分模型假设空间（$2^{12} = 4096$ 个立方体）上学到的推断能力，其速度和准确度已接近在仅含 109 个 Alchemy 立方体的已知空间上进行完美贝叶斯推断的水平。在 200 个时间步内，即使从未见过测试部分模型中的边组合，我们方法的模型误差也几乎降至真实部分模型预言机的 0 误差。我们方法的消融版本没有进行后验采样，而是使用其后验分布的期望；它的适应效果同样良好，但速度较慢、准确度较低，这表明采样有助于实现更好的模型适应。图 4 展示了我们的完整方法如何学会根据部分模型中边存在与否的新证据来更新其后验信念。

6.2 探索

在后验采样框架中，平衡的探索程度由精确后验采样预言机表示：精确后验采样只在有助于找到奖励时才收集新信息策略。一开始，它通过以更均匀的概率采样不同的部分模型，并针对这些样本执行最优策略来进行更多探索（图 3）。随着它的信念变得更加确定，并逐渐接近真实部分模型，精确后验采样的策略也变得不那么多样，逐渐接近真实部分模型的最优策略，从而有效减少探索，并更加专注于利用其可靠的信念。另一方面，随机策略和非自适应后验采样会随机收集信息，并在部分模型空间中探索更多独特的状态-动作对，即独特的顶点-边。真实部分模型只会在执行其对完美部分模型的信念所对应的最优策略时观察新的顶点-边；它无需探索。

我们的结果表明，我们的方法在探索与利用之间取得了良好的平衡（图 3）：其探索程度仅略高于精确后验采样，低于随机基线（随机策略 和非自适应 *PS*），高于不进行探索的真实部分模型 预言机。我们的消融实验表现出类似的探索行为，但其探索程度低于精确后验采样预言机和我们完整的方法，这可以解释为什么它对部分模型信念的适应效果不如完整方法。我们的消融实验在部分模型空间中探索到的独特状态-动作对较少，这可能限制了 Transformer 可利用上下文来改进其模型信念的程度。另一方面，我们的完整方法收集了更多独特的状态-动作对，从而使同一个 Transformer 能够做出更好的推断。

6.3 策略适应

我们的方法在奖励和适应速度方面的上下文策略适应表现也接近精确后验采样 预言机（图 5；附录 A 中使用另外两个训练随机种子定性地复现了这一结果）。鉴于我们的方法在

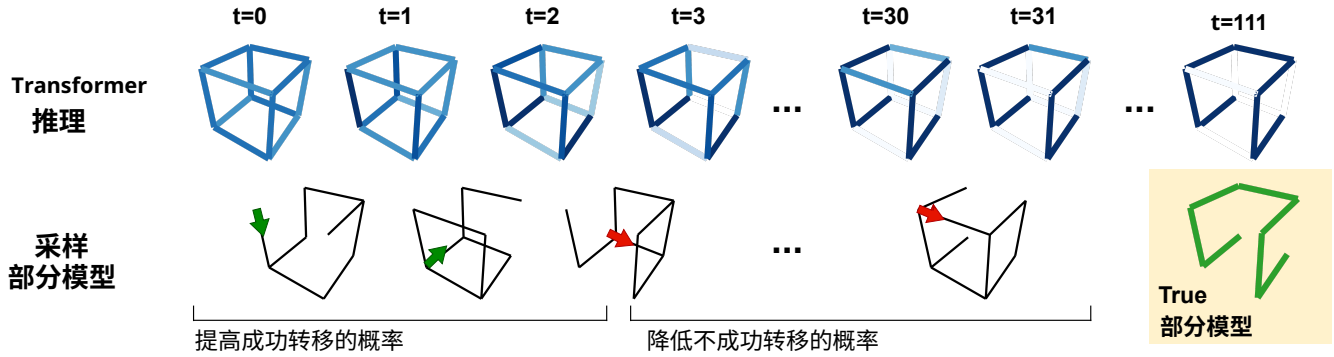


图 4：我们的方法进行推断和采样的示例。上：我们的 Transformer 对部分模型中边概率的推断，颜色越深表示概率越高。下：根据 Transformer 推断结果采样得到的部分模型。我们的方法在这些样本上执行值迭代，以选出一条待遍历的边，然后将其映射为一种药水，在完整的 Alchemy 环境中进行尝试。成功和失败的边转移分别以绿色和红色箭头表示；在完整环境中，观测结果体现为药水是否使石头发生变化。接收到这些观测并将其纳入上下文后（且不进行梯度更新），我们的 Transformer 会推断相应边的概率更高或更低。尽管训练期间从未见过真实的部分模型，它的信念仅用 111 个时间步就接近了真实值。

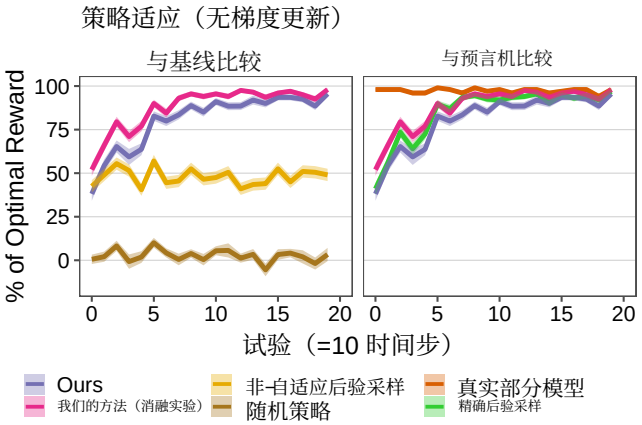


图 5：留出测试任务中的策略适应。线条和阴影带的含义与图 3 相同。我们的方法在上下文内 适应策略，使其获得约 90% 的最高可能奖励，适应速度接近精确的后验采样预言机（精确后验采样）。

适应部分模型信念方面表现出色，这些结果是合理的：随着这些信念迅速而准确地改善，采样模型就能更好地描述测试任务（的一部分），这意味着在该模型上进行值迭代会产生回报更高的策略。有趣的是，在较早的时间步中，我们的消融实验取得的奖励略高于完整方法（但在回合结束时两者表现相近）；这可能是由于它们利用 Transformer 中信息的方式不同。

而我们的消融使用 Transformer 的所有信息（即部分模型中边的期望值）来构建随机转移函数；我们的完整方法则通过采样 1/0 边概率来构建确定性转移函数，从而丢弃了其中一部分信息。考虑以下情况：我们的 Transformer 对其信念十分确定，也就是说，为边分配的概率极高或极低。对随机转移函数进行值迭代，会始终得到一项策略，沿着高概率的边前进（即使用 Alchemy药水），到达奖励最高的顶点（即 Alchemy石头）。在我们的完整方法中，采样得到的转移函数可能包含概率较低（但为 > 0）的边，也可能排除概率较高（但为 < 1）的边。对于相同的 Transformer 信念，我们的完整方法可能生成不同的转移函数，因此值迭代会得到波动更大的策略，无法始终到达奖励最高的顶点（石头）。

另一方面，策略更加稳定意味着我们的消融方法探索效果不如完整方法（如上文所述，并如图 3 所示），代价是模型适应效果也不如完整方法。我们强调，有效探索测试任务并由此准确地进行模型适应，是我们工作的核心关注点。获得测试任务的精确模型，对于增强我们对所得策略的信心至关重要。

6.4 部分模型能够导向高回报策略

我们曾假设，尽管部分模型只捕捉了完整环境中部分转移和奖励，但使用部分模型进行后验采样仍能产生高回报策略（见第 5 节）。后验采样涉及动态规划（在我们的工作中为值迭代），而动态规划通常假设可以访问完整模型，而非部分模型。图 5 显示，如果我们拥有良好的任务部分模型表示（Alchemy 就是如此），那么使用这些模型进行后验采样便能产生良好的策略：如果智能体对某项任务的信念已收敛到真实部分模型，那么它获得的奖励几乎与求解真实完整模型时相同，即几乎达到最优奖励的 100%（真实部分模型）。如果智能体仍不确定，但可以使用完美的贝叶斯推理机，它

就能调整策略，在 100 个时间步内获得几乎 100% 的最优奖励（精确后验采样）。如果智能体使用的不是完美推理机，而是近似推断过程（我们的 Transformer），则可以调整策略，在 100 个时间步内获得约 90% 的最优奖励（我们的方法）。

7 相关工作

7.1 元强化学习

我们的工作可以被视为一种元学习方法，它由一个“外层”学习算法（通过监督学习更新我们的 Transformer 权重）定义，该算法学习一个“内层”学习算法（我们的 Transformer 权重实现了一种学习算法，可根据上下文更新关于部分模型的分布）[9]。本质上，我们的方法“学会了如何学习”部分模型。通过将部分模型信念与规划（值迭代）相结合，我们方法的内层算法就成为一种基于模型的强化学习算法。内层强化学习算法是元强化学习方法的一项特征，而我们的工作与那些已展示出上下文内适应能力的元强化学习方法最为相关。关于其他有趣的元强化学习类别，请参阅 Hospedales 等人（2022年）的综述。

与我们的方法最相似的是 Rakelly 等人（2019年）提出的 PEARL。它学习一种内层算法，从上下文中推断潜变量，并像后验采样那样对该变量进行采样；另一个是 Laskin 等人（2022年）提出的算法蒸馏，它学习一种由 Transformer 权重编码的内层强化学习算法，并使用离线训练数据。然而，这些方法 [以及其他类似方法（如：3、33、36] ）与我们的方法并不能直接比较，因为它们适应的是策略（上下文内），没有中间的模型适应和规划过程。它们通常还使用强化学习或模仿学习进行训练，而且与我们的方法不同，无法利用我们的监督学习信号。

对于那些确实会在上下文内调整模型并进行规划的元强化学习方法，它们与我们的不同之处在于：它们要么不使用神经网络 [例如， 23] ，要么通常训练网络来预测观测，而不是更新模型上的分布 [例如，16, 34]。相反，我们专注于训练能够生成模型分布以进行后验采样的神经网络。另一个区别是，我们学习的是小型部分模型，而非更常见的学习完整马尔可夫决策过程各组成部分的方法；我们的方法可以降低规划成本。

我们的训练数据也不同于许多元强化学习研究，这些研究通常依赖经过策略改进的训练数据（在线或离线）。我们的工作表明，即使来自随机策略的数据也可能很有价值，并促使人们进一步研究如何从朴素策略中学习。

7.2 抽象马尔可夫决策过程

我们的部分模型也可以视为抽象马尔可夫决策过程，包含：(1) 一个抽象状态空间（即部分模型图上的顶点），用于压缩完整状态空间中的信息；(2) 定义在抽象状态上的转移函数和奖励函数（即图中的边和奖励位置）。这两个组成部分都是若干基于模型的强化学习研究的特点 [8, 14]。许多研究使用神经网络损失函数，促使模型学习能够包含足够信息、以重建原始观测的抽象状态。一个细微但有趣的区别是，我们的损失函数会有意鼓励排除观测中的信息（例如，在我们的环境中，抽象状态应舍弃有关药剂是否已用尽的信息）。与为了重建完整观测而进行优化的模型相比，我们的抽象模型需要编码的信息更少。值得注意的是，所编码的少量信息对于通过规划找到良好策略非常有用（图 5）。

通过价值等价方法 [例如， 4, 17, 24, 26, 29]，也实现了这样一种思路：学习排除完整马尔可夫决策过程中的部分信息、但仍有助于规划的抽象模型。价值等价是指，学习抽象模型时采用一种损失函数，以鼓励准确预测价值函数（而不试图重建原始观测）。就价值函数而言，使用抽象模型进行规划应当等同于在完整马尔可夫决策过程中进行规划。据我们所知，这些方法尚未解决我们的问题：在新任务中利用上下文来调整抽象马尔可夫决策过程。

8 结论

智能体如何才能在上下文内泛化到新任务？我们的方法基于后验采样 [20, 28, 30] ，并对通常未知或难以处理的组件进行近似：我们不再像精确贝叶斯推理那样明确指定先验分布和后验更新，而是使用 Transformer 从训练任务中学习这些内容。我们不再为大型马尔可夫决策过程模型计算代价高昂的解，而是考虑由较小的部分模型构成的假设空间，这些模型可以通过动态规划以更低的成本求解。在 Symbolic Alchemy 的一个版本中，我们评估了该方法对具有新颖转移函数的留出任务的泛化能力。我们的方法在适应速度和探索-利用平衡方面接近精确的后验采样预言机，并且仅依靠上下文即可实现这一点，也就是说，它利用状态和动作的历史，而非梯度更新。

局限性与未来工作。本文利用了有关如何将任务表示为部分模型的监督信号，因此可以专注于学习推断过程，并检验在后验采样框架中使用部分模型是否合理。我们的结果表明，能够良好表示任务的部分模型可能会排除寻找最优策略所需的信息，但基于这些模型进行规划仍能得到非常好的策略。未来的工作或许可以借鉴价值等价方法的思路，在没有监督信号的情况下学习部分模型表示，同时确保这些表示有助于规划。

致谢

我们感谢 Zita Marinho、TomSchaul、Jane Wang、Alaa Saade、Michael King 和 Sam Ritter 提供有益的讨论和反馈。我们也感谢 Lukas Schäfer 和 Mhairi Dunion 提供宝贵的写作反馈。

参考文献

[1] Kamyar Azizzadenesheli 和 Animashree Anandkumar。2019年。通过贝叶斯深度Q网络高效探索。arXiv。https://doi.org/10.48550/ arXiv.1802.04412 arXiv:1802.04412 [cs,stat][2] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda

Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever 和 Dario Amodei。2020年。语言模型是少样本学习者。载于神经信息处理系统进展，第33卷。Curran Associates, Inc.，1877–1901。[3] Yan Duan、John Schulman、Xi Chen、Peter L. Bartlett、Ilya Sutskever 和 Pieter Abbeel。2016年。RL^2: 通过慢速强化学习实现快速强化学习。arXiv。https://doi.org/ 10 48550 1611 02779 1611 02779 [ ] 。/arXiv. 。[4] arXiv: 。cs, stat Gregory Farquhar、Tim Rocktäschel、Maximilian Igl 和 Shimon Whiteson。2018年。TreeQN 和 ATreeC：用于深度强化学习的可微树结构模型。载于国际学习表征会议。[5] Meire Fortunato、Mohammad Gheshlaghi Azar、Bilal Piot、Jacob Menick、Matteo Hessel、Ian Osband、Alex Graves、Volodymyr Mnih、Remi Munos、Demis Hassabis、Olivier Pietquin、Charles Blundell 和 Shane Legg。2018年。用于探索的噪声网络。载于国际学习表征会议。[6] Shivam Garg、Dimitris Tsipras、Percy Liang 和 Gregory Valiant。2022年。Transformer 能在上下文内学到什么？简单函数类的案例研究。载于神经信息处理系统进展。[7] Anirudh Goyal、Abram Friesen、Andrea Banino、Theophane Weber、Nan Rosemary Ke、Adrià Puigdomènech Badia、Arthur Guez、Mehdi Mirza、Peter C. Humphreys、Ksenia Konyushova、Michal Valko、Simon Osindero、Timothy Lillicrap、Nicolas Heess 和 Charles Blundell。2022年。检索增强强化学习。载于第*39*届国际机器学习会议论文集。PMLR，7740–7765。[8] David Ha 和 Jürgen Schmidhuber。2018年。世界模型。*arXiv*（2018年3月）。https://doi.org/ 10 5281 1207631 1803 10122 [ ] 。/[9] zenodo。arXiv: 。cs, stat Timothy Hospedales、Antreas Antoniou、Paul Micaelli 和 Amos Storkey。2022年。神经网络中的元学习：综述。*IEEE*模式分析与机器智能汇刊44，9（2022年9月），5149–5169。https://doi.org/ 10.1109/TPAMI.2021.3079209[10] Nan Rosemary Ke、Silvia Chiappa、Jane Wang、Jorg Bornschein、Theophane Weber、Anirudh Goyal、Matthew Botvinic、Michael Mozer 和 Danilo Jimenez Rezende。2022年。学习诱导因果结构。arXiv:2204.04875[11] Jannik Kossen、Neil Band、Clare Lyle、Aidan N Gomez、Thomas Rainforth 和 Yarin Gal。2021年。数据点之间的自注意力：超越深度学习中的单个输入输出对。载于神经信息处理系统进展，第34卷。Curran Associates, Inc.，28742–28756。[12] Michael Laskin、Luyu Wang、Junhyuk Oh、Emilio Parisotto、Stephen Spencer、Richie Steigerwald、D. J. Strouse、Steven Hansen、Angelos Filos、Ethan Brooks、Maxime Gazeau、Himanshu Sahni、Satinder Singh 和 Volodymyr Mnih。2022年。通过算法蒸馏实现上下文内强化学习。arXiv。arXiv: 2210 14215 [ ] 。cs Zachary Lipton、Xiujun Li、Jianfeng Gao、[13] Lihong Li、Faisal Ahmed 和 Li Deng。2018年。BBQ-Networks：面向任务的对话系统深度强化学习中的高效探索。*AAAI*人工智能会议论文集 32, 1 2018 10 1609 32 1 11946 （4月）。[14] https://doi.org/ 。/aaai.v i 。Thomas M. Moerland、Joost Broekens、Aske Plaat 和 Catholijn M. Jonker。2022年。基于模型的强化学习：综述。arXiv。https://doi.org/10.48550/ arXiv.2006.16712 arXiv:2006.16712 [cs, stat][15] Samuel Müller、Noah Hollmann、Sebastian Pineda Arango、Josif Grabocka 和 Frank Hutter。2022年。Transformer 能够进行贝叶斯推断。载于国际学习表征会议。[16] Anusha Nagabandi、Ignasi Clavera、Simin Liu、Ronald S. Fearing、Pieter Abbeel、Sergey Levine 和 Chelsea Finn。2019年。通过元强化学习，在真实动态环境中学习适应。载于国际学习表征会议。[17] Junhyuk Oh、Satinder Singh 和 Honglak Lee。2017年。价值预测网络。载于神经信息处理系统进展，第30卷。Curran Associates, Inc。[18] Ian Osband、Charles Blundell、Alexander Pritzel 和 Benjamin Van Roy。2016年。通过自举DQN进行深度探索。载于神经信息处理系统进展，第 29 卷。Curran Associates, Inc.[19] Ian Osband、Benjamin Van Roy、Daniel J. Russo 和 Zheng Wen。2019年。通过随机化价值函数进行深度探索。机器学习研究杂志20，124（2019年），1–62。[20] Ian Osband、Daniel Russo 和 Benjamin Van Roy。2013年。（更）高效的强化学习：通过后验采样实现。载于神经信息处理系统进展，第 26 卷。Curran Associates, Inc.[21] Kate Rakelly、Aurick Zhou、Chelsea Finn、Sergey Levine 和 Deirdre Quillen。2019年。通过概率上下文变量实现高效离策略元强化学习。载于第*36*届国际机器学习大会论文集。PMLR，5331–5340。[22] Daniel Russo、Benjamin Van Roy、Abbas Kazerouni、Ian Osband 和 Zheng Wen。2020年。汤普森采样教程。arXiv。https://doi.org/10.48550/arXiv。 1707 02038 1707 02038 [ ] 。arXiv：。cs[23] Steindór Sæmundsson、Katja Hofmann 和 Marc Peter Deisenroth。2018年。利用潜变量高斯过程进行元强化学习。载于人工智能不确定性会议（*UAI*）。arXiv:1803.07551[24] Julian Schrittwieser、Ioannis Antonoglou、Thomas Hubert、Karen Simonyan、Laurent Sifre、Simon Schmitt、Arthur Guez、Edward Lockhart、Demis Hassabis、Thore Graepel、Timothy Lillicrap 和 David Silver。2020年。通过基于学习模型的规划，精通雅达利、围棋、国际象棋和将棋。《自然》588，7839（2020年12月），604 609 10 1038 41586 020 03051 4 –。https://doi.org/。/s--- Eric[25] Schulz、Emmanouil Konstantinidis 和 Maarten Speekenbrink。2015年。上下文多臂老虎机任务中的学习与决策。载于认知科学学会年会。[26] David Silver、Hado van Hasselt、Matteo Hessel、Tom Schaul、Arthur Guez、Tim Harley、Gabriel Dulac-Arnold、David Reichert、Neil Rabinowitz、Andre Barreto 和 Thomas Degris。2017年。预测器网络：端到端学习与规划。载于第*34*届国际机器学习大会论文集。PMLR，3191–3199。[27] Maarten Speekenbrink 和 Emmanouil Konstantinidis。2015年。不休眠老虎机问题中的不确定性与探索。认知科学专题7，2（2015年），351 367 10 1111 12145 –。https://doi.org/。/[28] tops。Malcolm Strens。贝叶斯框架下的强化学习。载于国际机器学习大会。[29] Aviv Tamar、YI WU、Garrett Thomas、Sergey Levine 和 Pieter Abbeel。2016年。价值迭代网络。载于神经信息处理系统进展，第 29 卷。Curran Associates, Inc.[30] William R. Thompson。1933年。根据两组样本的证据，一个未知概率大于另一个的可能性。《生物计量学》25，3/4（1933年），285–294。https://doi.org/10.2307/2332286[31] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。注意力机制即你所需。载于神经信息处理系统进展，第 30 卷。Curran Associates, Inc.[32] Jane X. Wang、Michael King、Nicolas Pierre Mickael Porcel、Zeb Kurth-Nelson、Tina Zhu、Charlie Deck、Peter Choy、Mary Cassin、Malcolm Reynolds、H. Francis Song、Gavin Buttimore、David P. Reichert、Neil Charles Rabinowitz、Loic Matthey、Demis Hassabis、Alexander Lerchner 和 Matthew Botvinick。2021年。Alchemy：元强化学习智能体的基准测试与分析工具包。载于神经信息处理系统第三十五届大会数据集与基准测试专题。[33] Jane X. Wang、Zeb Kurth-Nelson、Dhruva Tirumala、Hubert Soyer、Joel Z. Leibo、Remi Munos、Charles Blundell、Dharshan Kumaran 和 Matt Botvinick。2016年。学习强化学习。arXiv。arXiv: 1611 05763 。Ari Weinstein 和[34] 2017 Matthew M. Botvinick。运动控制中的结构学习：一种深度强化学习模型。arXiv。https://doi.org/10.48550/arXiv。 1706 06827 1706 06827 [ ] 。arXiv：。[35] cs Sang Michael Xie、Aditi Raghunathan、Percy Liang 和 Tengyu Ma。2022年。将上下文内学习解释为隐式贝叶斯推断。载于国际学习表征会议。[36] Luisa Zintgraf、Sebastian Schulze、Cong Lu、Leo Feng、Maximilian Igl、Kyriacos Shiarlis、Yarin Gal、Katja Hofmann 和 Shimon Whiteson。2021年。VariBAD：通过元学习实现变分贝叶斯自适应深度强化学习。机器学习研究杂志22，289（2021年），1–39。

A 训练随机种子间的可复现性

我们的结果 i 图 3 和图 5 所示结果在定性上可通过另外两个训练随机种子复现，如图 6–9 所示。

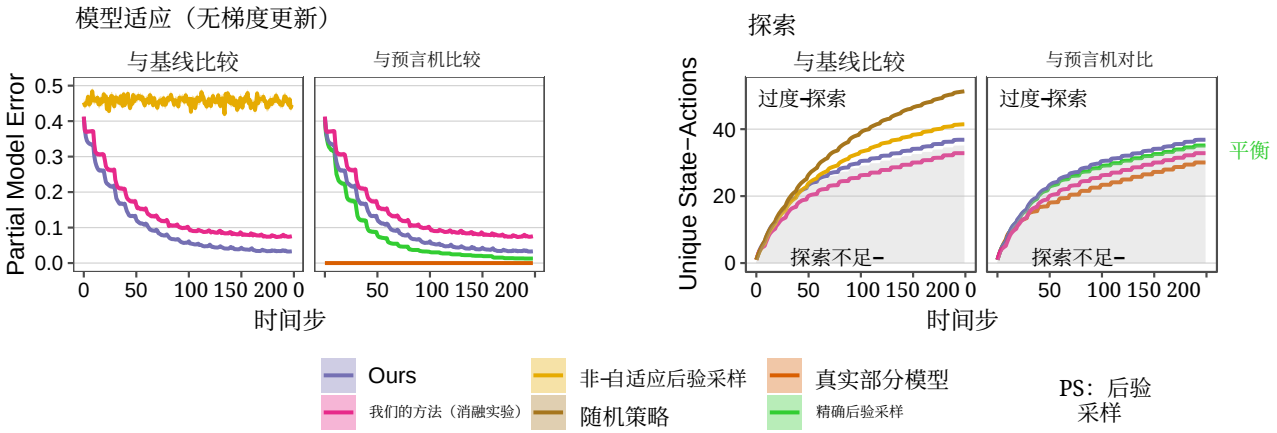


图 6：使用第二个训练随机种子复现图 3。

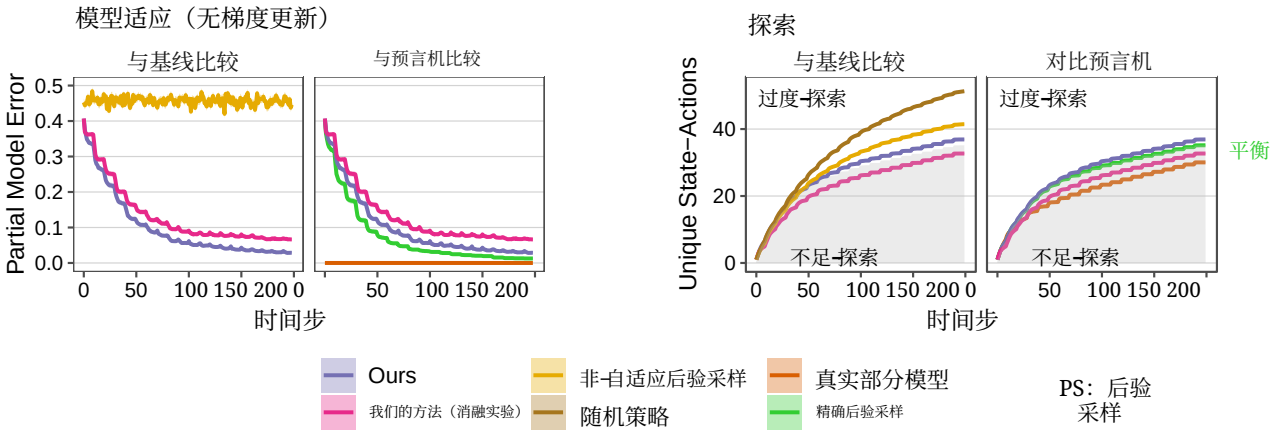


图 7：使用第三个训练随机种子复现图 3。

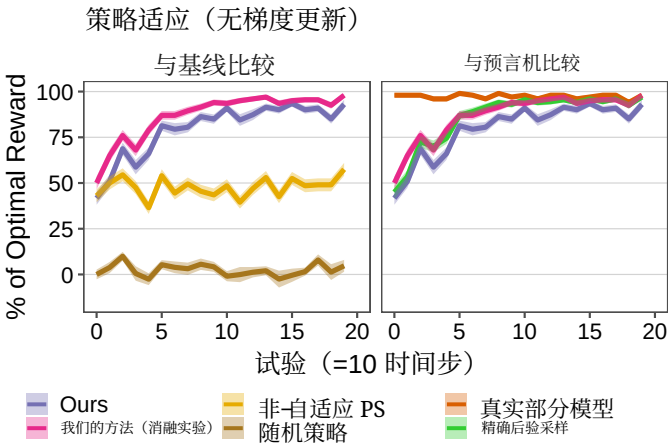


图 8：使用第二个训练随机种子复现图 5。

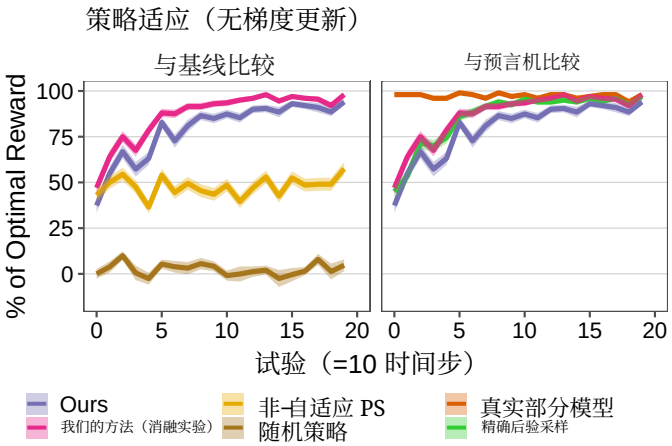


图 9：使用第三个训练随机种子复现图 5。