解构深度主动推理。

Th´eophile Champion TMAC3@KENT.AC.UK
英国坎特伯雷 *CT2 7NZ* 肯特大学计算机学院

Marek Grze´s M.GRZES@KENT.AC.UK
英国坎特伯雷 *CT2 7NZ* 肯特大学计算机学院

Lisa Bonheme LB732@KENT.AC.UK
肯特大学，计算机学院，坎特伯雷 *CT2 7NZ*，英国

Howard Bowman H.BOWMAN@KENT.AC.UK
伯明翰大学，心理学学院，伯明翰 *B15 2TT*，英国；肯特大学，计算机学院，坎特伯雷 *CT2 7NZ*，英国；伦敦大学学院，威康人类神经影像中心（荣誉职位），伦敦 *WC1N 3AR*，英国

arXiv:2303.01618v2 [cs.AI] 8 May 2023

编辑：待填写

摘要

主动推断是一种关于感知、学习和决策的理论，可应用于神经科学、机器人学、心理学和机器学习。近年来，人们一直在通过蒙特卡洛树搜索和深度学习扩展这一框架，并开展了大量研究。这些工作的最终目标是利用深度主动推理解决更加复杂的任务。首先，我们回顾现有文献；随后，逐步构建深度主动推理智能体，具体如下：(i) 实现变分自编码器（VAE）；(ii) 实现深度隐马尔可夫模型（HMM）；(iii) 实现深度临界隐马尔可夫模型（CHMM）；以及 (iv) 实现完整的深度主动推理智能体（DAI）。对于 CHMM 和 DAI 智能体，我们尝试了五种期望自由能的定义和三种不同的动作选择策略。根据我们的实验，能够解决 dSprites环境的模型都是最大化奖励的模型。最后，我们使用中心化卷积核对齐，比较了各种模型（例如深度Q网络、CHMM、DAI）的各层所学习表征的相似性。值得注意的是，最大化奖励的 CHMM 与最小化期望自由能的 CHMM 学习到了非常相似的表征，但评论家网络的最后一层除外（这反映了学习目标的差异），以及转移网络和编码器网络中的方差层。进一步检查这些（方差）层时，我们发现，最大化奖励的 CHMM 的转移网络比最小化期望自由能的 CHMM 的转移网络确定得多。更准确地说，最小化期望自由能的 CHMM 只有在执行向下动作时，才对世界状态转移有较高置信度。这表明，最小化期望自由能的 CHMM 总是选择向下动作，并未为其他动作收集足够的数据。相比之下，最大化奖励的 CHMM 持续选择向左和向右动作，从而成功解决任务。这两个 CHMM 之间唯一的差异在于认知价值，它旨在使转移网络和编码器网络的输出尽可能接近。因此，最小化期望自由能的 CHMM 会反复选择同一个动作（向下），并在选择该动作时成为预测未来的专家。这实际上使转移网络与编码器网络输出之间的 KL散度变小。此外，选择向下动作时，平均奖励为零，而对于其他所有动作，期望奖励均为负。因此，如果 CHMM 必须坚持选择单一动作以保持 KL散度较小，那么向下动作就是奖励最高的选择。由此，深度主动推理中认知价值的适当表述仍是一个悬而未决的问题。

能量会反复选择单一动作（向下），并在选择该动作时成为预测未来的专家。这实际上使转移网络与编码器网络输出之间的 KL散度变小。此外，选择向下动作时，平均奖励为零，而对于其他所有动作，期望奖励均为负。因此，如果 CHMM 必须坚持选择单一动作以保持 KL散度较小，那么向下动作就是奖励最高的选择。由此，深度主动推理中认知价值的适当表述仍是一个悬而未决的问题。

关键词： 深度学习、主动推断、贝叶斯统计、自由能原理、强化学习

1. 引言

主动推断是一个统一的感知、学习和规划框架，源自理论神经科学（Costa 等人，2020a；Champion等人，2021、2022b、a、c）。这一框架成功解释了广泛的脑现象（Friston等人，2016；Itti和Baldi，2009；Schwartenbeck等人，2018；FitzGerald等人，2015），并已应用于机器人学和人工智能中的大量任务（Fountas 等人，2020；Pezzato等人，2020；Sancaktar 等人，2020；Cˌatal 等人，2020；Cullen等人，2018；Millidge，2019）。

一个富有前景的研究方向，是扩展这一理论框架，以应对日益复杂的任务。朝这一目标开展的研究通常受到机器学习最新进展的推动。例如，变分自编码器（Doersch，2016；Higgins等人，2017；Kingma和Welling，2014；Rezende等人，2014）对于将深度神经网络整合到主动推断中发挥了关键作用（Sancaktar 等人，2020；Cˌatal 等人，2020；Millidge，2020），而蒙特卡洛树搜索算法（Browne 等人，2012；Silver等人，2016）已被用于改进规划效率（Fountas 等人，2020；Champion等人，2022b、a、c、d）。

另一个密切相关的领域是强化学习（Mnih 等，2013；van Hasselt等人，2015；Lample和Chaplot，2016），它研究的是同类任务：智能体必须与环境交互。该领域一个众所周知的挑战是连续样本之间存在相关性，这违背了大多数机器学习所依赖的标准独立同分布假设。为打破这种相关性，研究人员提出将智能体过去的经验存储在经验回放缓冲区中（Mnih 等，2013）。随后可以从缓冲区中随机重新采样这些经验，用于训练 Q网络，以近似 Q值。训练 Q网络的目标是最小化其输出与目标值之间的均方误差，该目标值定义如下：

$$y(o_t, a_t) = \mathbb{E}_{o_{t+1}\sim E(o_t,a_t)}\Big[r_t + \gamma \max_{a_{t+1}\in\mathcal{A}} \mathcal{Q}_{\theta_a}(o_{t+1}, a_{t+1})\Big],$$

其中，t 是当前时间步，$\mathcal{A}$ 是可用动作集合，$y(o_t, a_t)$是待预测的目标 Q 值，$\mathbb{E}$是关于从环境接收到的观测的期望，r_t 是智能体在状态[1] o_t下执行动作 a_t 时获得的奖励，o_{t+1} 是在状态 o_t下执行动作 a_t 后到达的状态，E 是对 o_{t+1} 进行采样的环境模拟器，γ 是用于折扣未来奖励的折扣因子，而 $\mathcal{Q}_{\theta_a}(o_{t+1}, a_{t+1})$是 Q网络的输出，即在状态 o_{t+1}下执行动作 a_{t+1} 的估计 Q 值。

遗憾的是，使用上述目标来训练 Q网络可能会导致训练不稳定。通常，$\hat{}$ 通过引入一个目标网络 $\mathcal{Q}_{\hat{\theta}_a}(o_{t+1}, a_{t+1})$来解决这一问题，该网络只是 Q 网络的一个副本。随后，每隔一段时间就将目标网络的权重与 Q 网络的权重同步

1. 请注意，我们使用符号说明 o_τ 表示（任意）时间步 τ处的（可观测）状态，而不是更常见的符号 s_τ。这是因为我们保留符号 s_τ，用于表示主动推断中出现的（不可观测）状态。

次（学习）迭代（Mnih 等，2013）。新的目标通过用目标网络替换 Q网络得到，即，

$$y(o_t, a_t) = \mathbb{E}_{o_{t+1} \sim E(o_t, a_t)} \Big[r_t + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{\mathcal{Q}}_{\hat{\theta}_a}(o_{t+1}, a_{t+1}) \Big].$$

在第2节中，我们回顾现有文献，并介绍：深度Q网络（DQN）智能体（Mnih 等，2013）；Fountas 等人（2020）提出的采用蒙特卡洛方法的深度主动推理智能体（DAI_{MC}）；Millidge（2020）提出的将深度主动推理作为变分策略梯度的方法（DAI_{VPG}）；Rood 等人（2020）提出的橡胶手错觉深度主动推理智能体（DAI_{RHI}）；Sancaktar 等人（2020）、Lanillos 等人（2020）和 Oliver 等人（2019）提出的人形机器人控制深度主动推理智能体（DAI_{HR}）；Ueltzh¨offer（2018）提出的基于自由动作目标的深度主动推理智能体（DAI_{FA}）；van der Himst 和 Lanillos（2020）提出的适用于部分可观测马尔可夫决策过程的深度主动推理智能体（DAI_{POMDP}）；以及其他一些在线上无法获取代码的方法。我们认为，尽管这些方法都揭示了实现深度主动推理智能体所涉及的重要问题，但迄今尚未有完整的实现方案发表。因此，为了系统地探索深度主动推理智能体的构建过程，我们在第3节中逐步构建这样一个智能体。我们首先从一个由编码器和解码器网络组成的简单变分自编码器（VAE）开始。接着，加入转移网络，构成深度隐马尔可夫模型（HMM）。然后，加入评论家网络以定义动作先验，从而得到评论家隐马尔可夫模型（CHMM）。最后，加入策略网络来近似动作后验，由此构成完整的深度主动推理（DAI）智能体。随后，在第4节中，我们讨论各个中间步骤的能力与局限性，并在本节分析和讨论各个中间模型所学习到的表征。最后，第6节将我们的发现置于相关背景中，并对本文作出总结。

2. 现有研究综述

本节将讨论强化学习文献中的 DQN 智能体、主动推断文献中六个可在线获取代码的智能体（DAI_{MC}, DAI_{VPG}, DAI_{RHI}, DAI_{HR},DAI_{FA} 和 DAI_{POMDP}），以及其他几个代码不可获取的深度主动推理智能体。最后，我们将说明如何使用中心化核对齐来比较智能体学到的表征。请注意，本节使用的符号汇总于附录 A。

2.1 DQN 智能体（Mnih 等，2013）

我们先从 DQN 智能体（Mnih 等，2013）开始介绍，其目标是最大化随时间获得的奖励总量。在每个时间步 τ，智能体都会观测一幅图像 o_τ，并被允许执行一个动作 $a_\tau \in \mathcal{A}$。在观测到 o_τ时执行 a_τ后，智能体会获得奖励 r_τ。Q学习算法（Sutton 等，1998）旨在通过计算每个状态-动作对（o_τ， a_τ）的Q值 $Q(o_\tau, a_\tau)$ 来最大化奖励。Q值表示在状态 o_τ下采取动作 a_τ 所能获得的预期奖励总量。对于Atari游戏 等基于图像的领域，这种方法难以实现，因为需要为每一幅可能的图像存储一个 Q值向量。因此，研究者开发了 DQN算法（如图 1 所示），利用深度神经网络 $\mathcal{Q}_{\theta_a}$ 来近似 Q值。更正式地说， $\mathcal{Q}_{\theta_a}$ 将任意观测映射为一个尺寸为 $\#A$ 的向量，其中包含每个可能动作的 Q值；当输入图像 o_τ时，我们用 $\mathcal{Q}_{\theta_a}(o_\tau, a_\tau)$ 表示 Q_{θ_a}预测的输出向量中位置 a_τ处的元素。正如我们在引言中讨论的那样，通过引入目标网络 $\mathcal{Q}_{\hat{\theta}_a}$可以提高 Q-ˆ 网络的训练稳定性。该网络在结构上与 Q网络完全相同，其权重每经过 K （学习）次迭代便与 Q网络的权重同步。随后，使用梯度下降优化 Q网络的权重，以最小化 Q网络输出与目标值之间的均方误差，即

Q网络的输出与目标值之间的均方误差，即 $\theta_a^* = \arg\min_{\theta_a} \text{MSE}[\mathcal{Q}_{\theta_a}(o_t, \cdot)\ y(o_t, \cdot)],$，其中 $y(o_t, a_t)$ 是每个状态-动作对对应的目标 Q 值，并且正如前文所强调的，其定义如下：

$$y(o_t, a_t) = \mathbb{E}_{o_{t+1} \sim E(o_t, a_t)} \Big[r_t + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{\mathcal{Q}}_{\hat{\theta}_a}(o_{t+1}, a_{t+1}) \Big].$$

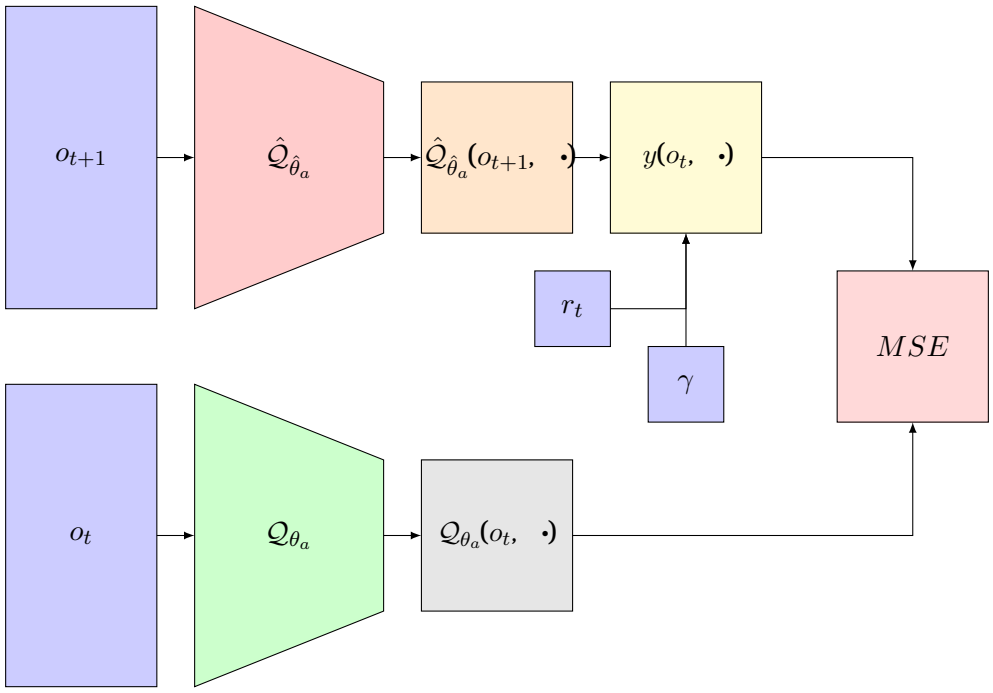


图 1：此图展示了 DQN 智能体。简而言之，图像 o_t 被输入 Q网络，图像 o_{t+1} 被输入目标网络。Q网络输出时间 t每个动作对应的 Q值，目标网络输出时间 $t+1$每个动作对应的 Q值。随后，利用时间 $t+1$ 每个动作对应的奖励、折扣因子和 Q值来计算目标值 $y(o_t, \cdot)$。最后，通过调整 Q网络的权重，使 Q网络的预测与目标值之间的均方误差最小化。

2.2 DAI_{MC} 智能体（Fountas 等人（2020））

本节回顾 Fountas 等人（2020）提出的 DAI_{MC} 智能体。该智能体代表了最雄心勃勃且最完整的深度主动推理智能体实现，因此为该领域引入了重要的新概念。相关代码可在以下网址获取：https://github.com/ zfountas/deep-active-inference-mc。 DAI_{MC} 智能体由四个深度神经网络组成，如图 2 和图 3 所示。编码器 $\mathcal{E}_{\phi_s}$ 以图像作为输入，输出隐状态上变分分布的均值和方差，即 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t;\ \mu, \sigma)$，其中 $\mu, \sigma = \mathcal{E}_{\phi_s}(o_t)$。解码器 $\mathcal{D}_{\theta_o}$ 以状态作为输入，输出一组伯努利分布的参数，这些参数可解释为预期的（重建）图像 $\hat{}o_t$，即，

$$P_{\theta_o}(o_t|s_t) = \mathcal{B}\text{ernoulli}(o_t; \hat{o}_t),$$

其中，$\hat{}o_t = \mathcal{D}_{\theta_o}(s_t)$ 是解码器预测的值，而 $\mathcal{B}\text{ernoulli}(o_t; \hat{}o_t)$ 是由伯努利分布构成的乘积分布，其定义如下：

$$\mathcal{B}\text{ernoulli}(o_t; \hat{o}_t) = \prod_{x,y} \text{Bernoulli}(o_t[x, y]; \hat{o}_t[x, y]),$$

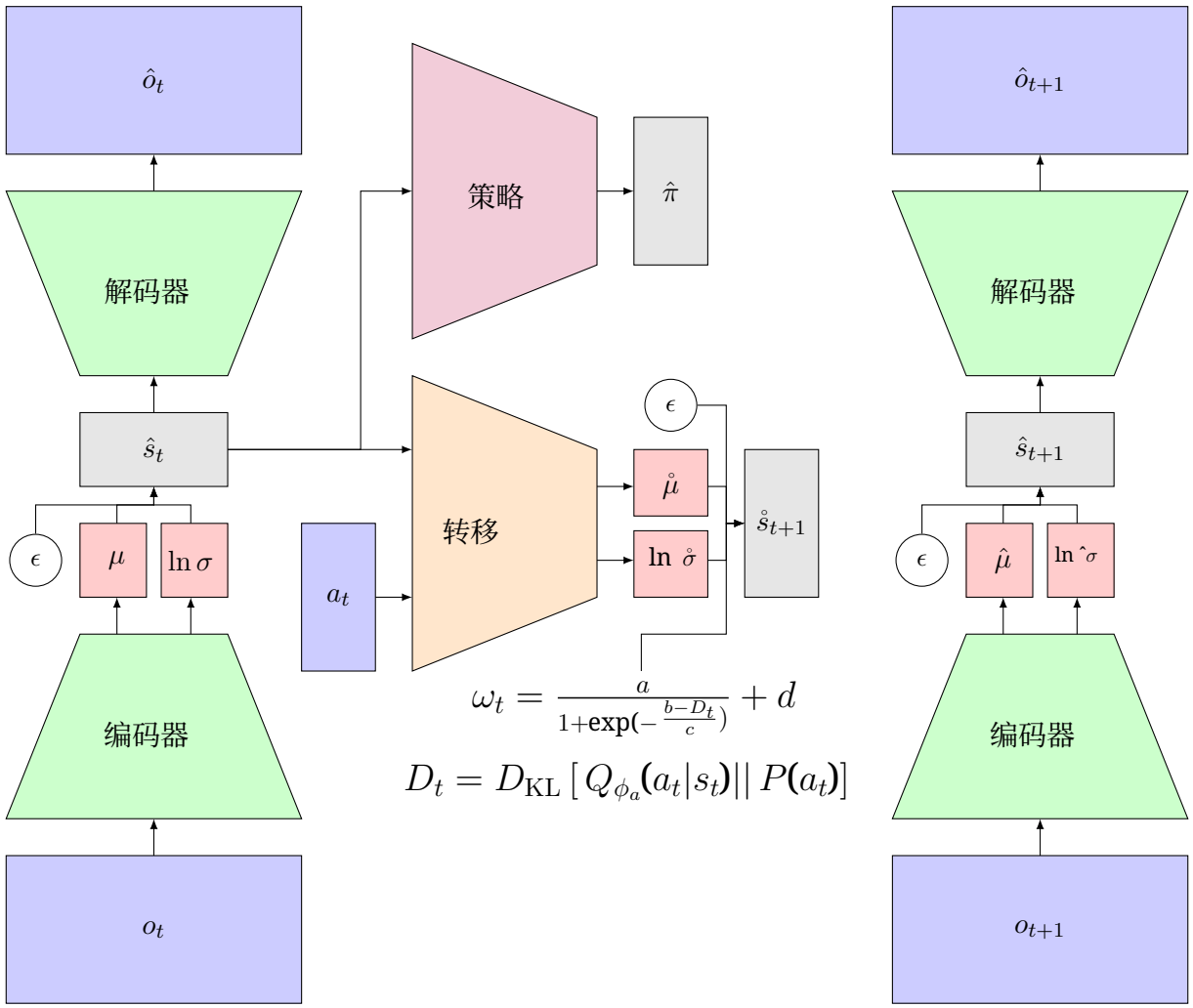


图 2：该图展示了 DAI_{MC} 智能体，它由一个编码器、一个解码器、一个转移网络和一个策略网络组成。图中重复展示了相同的变分自编码器（编码器和解码器），以反映连续的时间点。

其中，Bernoulli(·; ·) 是像素 $o_t[x, y]$ 所有可能取值上的伯努利分布，其参数由参数 $\hat{o}_t[x, y]$ 确定。转移网络 $\mathcal{T}_{\theta_s}$ 以状态-动作对作为输入，并输出隐状态上高斯分布的均值和方差，即 $P_{\theta_s}(s_{\tau+1}|s_\tau, a_\tau) = \mathcal{N}(s_{\tau+1}; \mathring{\mu}, \frac{\mathring{\sigma}}{\omega_t})$，其中 $\mathring{\mu}, \mathring{\sigma} = \mathcal{T}_{\theta_s}(s_\tau, a_\tau)$，而 ω_t 是自上而下注意力参数，用于调节转移映射的精度（见下文）。策略网络 $\mathcal{P}_{\phi_a}$ 以状态作为输入，并输出动作上的分布，即 $Q_{\phi_a}(a_t|s_t) = \mathrm{Cat}(a_t; \hat{\pi})$，其中 $\hat{\pi} = \mathcal{P}_{\phi_a}(s_t)$。最后，动作先验定义如下：

$$P(a_t) = \sum_{\pi \in \Pi} [\pi_t = a_t] P(\pi), \tag{1}$$

其中，Π是所有可能策略的集合，π_t 是策略 π 在时间 t规定的动作，方括号表示指示函数：若括号内的条件成立，则值为一；否则为零，而 $P(\pi)$ 是定义如下的策略先验：

$$P(\pi) = \sigma[-G(\pi)],$$

其中，$\sigma[\cdot]$ 是 softmax 函数，而 $G(\pi)$ 是策略 π的期望自由能（EFE），其定义如下：

$$G(\pi) = \sum_{\tau=t}^{T} G_\tau(\pi) = \sum_{\tau=t}^{T} \mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau, \theta|\pi) - \ln \tilde{P}(o_\tau, s_\tau, \theta|\pi)\Big], \tag{2}$$

其中，$\tilde{Q} = Q(o_\tau, s_\tau, \theta|\pi) = Q(\theta|\pi)Q(s_\tau|\theta, \pi)Q(o_\tau|s_\tau, \theta, \pi)$ 是预测后验，而 $\tilde{P}(o_\tau, s_\tau, \theta|\pi) = P(\theta|s_\tau, o_\tau, \pi)P(s_\tau|o_\tau, \pi)P(o_\tau|\pi)$ 是目标分布。不过，公式 2 需要重新整理才能在实践中计算[2]，，第 2.2.3 节将给出这一推导。最后，如图 2 所示，自上而下注意力参数的计算方式如下：

$$\omega_t = \frac{a}{1 + \exp(-\frac{b - D_t}{c})} + d,$$

其中，$D_t = D_{\mathrm{KL}}\left[Q_{\phi_a}(a_t|s_t)||P(a_t)\right]$，和 $\{a, b, c, d\}$ 是固定的超参数。直观而言，当（来自策略网络的）动作后验接近（由期望自由能得到的）动作先验时，ω_t 较高；当动作后验与动作先验相距较远时，ω_t 较低。这反过来意味着，当动作后验与动作先验差异很大时，转移映射中会引入额外的不确定性（见公式 1 前的段落）。最后请注意，计算动作先验（定义见公式 1）所需的项数会随着规划的时间范围增加而呈指数级增长。由于这在实践中难以处理，Fountas 等人（2020年）采用蒙特卡洛树搜索（MCTS）算法来评估每个动作的期望自由能（见下文）。最后，通过从以下分布中采样来执行动作选择：

$$\tilde{P}(a_t) = \frac{N(\hat{s}_t, a_t)}{\sum_{\hat{a}_t} N(\hat{s}_t, \hat{a}_t)},$$

其中，$\hat{s}_t$ 是环境的当前状态，而 $N(s_t, a_t)$表示在 MCTS 中，从状态 s_t 采取动作 a_t 的访问次数。

2.2.1 The Monte-Carlo tree search

本节将介绍 DAI_{MC}使用的规划算法，即蒙特卡洛树搜索（MCTS）。MCTS 通过让智能体向前展望未来来增强其规划能力。在每个动作-感知循环开始时，智能体会获得一幅图像 o_t。该图像可输入编码器，以得到潜在状态后验分布的均值向量 μ，即 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t; \mu, \sigma)$。由于 μ 是高斯后验分布的均值，因此可以将其解释为时间步 t时潜在状态的最大后验估计（MAP）。该最大后验估计将构成蒙特卡洛树搜索（MCTS）的根节点。

MCTS 的第一步是使用树的置信上界（UCT）准则，确定应扩展树中的哪个节点。将树的根节点$\hat{s}_t$ 称为当前节点，记作$\hat{s}_\tau$。如果当前节点没有子节点（即此前尚未从当前节点选择过动作），则选择该节点进行扩展。否则，UCT 准则值最高的子节点将成为新的当前节点，并重复这一过程，直至到达叶节点（即此前尚未从该节点选择过动作的节点）。与动作$\hat{a}_\tau$ 对应的$\hat{s}_\tau$子节点的 UCT 准则（Browne 等人，2012）如下：

$$UCT(\hat{s}_\tau, \hat{a}_\tau) = -\bar{G}(\hat{s}_\tau, \hat{a}_\tau) + C_{explore} \cdot \frac{Q_{\phi_a}(a_\tau = \hat{a}_\tau | s_\tau = \hat{s}_\tau)}{1 + N(\hat{s}_\tau, \hat{a}_\tau)},$$

2. 这里的“在实践中”是指“在代码中”，或者说“在实现该方法时”。

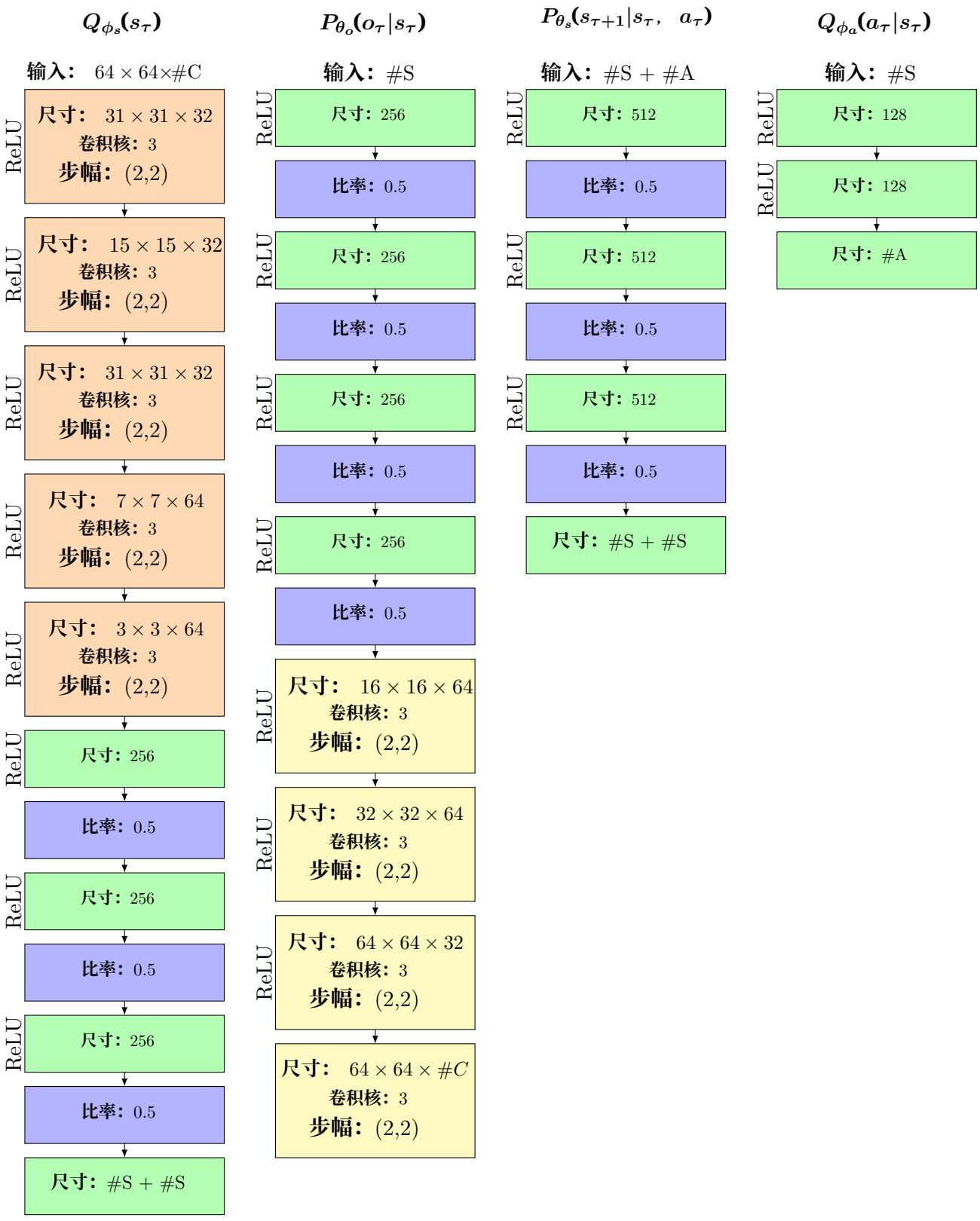


图3：DAI_{MC} 智能体的神经网络架构。橙色模块对应卷积层，绿色模块对应全连接层，蓝色模块对应丢弃层，黄色模块对应上卷积层。对于 dSprites环境，有四个动作（即 #A $= 4$）、十个状态（即 #S $= 10$）和一个通道（即 #C $= 1$）。对于 Animal-AI环境，有三个动作（即 #A $= 3$）、十个状态（即 #S $= 10$）和三个通道（即 #C $= 3$）。这7个智能体都经过训练，以最小化变分自由能。

其中，$G(\hat{s}_\tau, \hat{a}_\tau)$ 是在状态 $\hat{s}_\tau$ 下执行动作 $\hat{a}_\tau$ 的平均期望自由能，$C_{explore}$ 是探索常数，用于调节树级别的探索程度；$N(\hat{s}_\tau, \hat{a}_\tau)$ 是在状态 $\hat{s}_\tau$ 下动作 $\hat{a}_\tau$ 被访问的次数，而 $Q_{\phi_a}(a_\tau = \hat{a}_\tau | s_\tau = \hat{s}_\tau)$ 是策略网络预测的在状态 $\hat{s}_\tau$ 下动作 $\hat{a}_\tau$ 的后验概率。

令 $\mathring{s}_\tau$ 为上述选择过程选出的（叶）节点。随后，MCTS 会扩展 $\mathring{s}_\tau$ 的一个子节点。扩展过程使用转移网络计算 $P_{\theta_s}(s_{\tau+1} | s_\tau = \mathring{s}_\tau, a_\tau = \mathring{a}_\tau)$ 的均值 $\mathring{\mu}$，并将其视为时刻 $\tau + 1$ 状态的最大后验估计。接着，我们需要估计（虚拟）执行动作 $\mathring{a}_\tau$ 的代价。根据定义，该代价是式（2）给出的期望自由能，并可通过蒙特卡洛展开来改进其估计。规划迭代的最后一步是将新扩展的（虚拟）动作的代价沿树向上传播至根节点。形式上，更新可写为：

$$\forall s \in \mathbb{A}_{\mathring{s}_\tau} \cup \{\mathring{s}_\tau\}, \quad \boldsymbol{G}_s \leftarrow \boldsymbol{G}_s + \boldsymbol{G}_{\mathring{s}_\tau}, \tag{3}$$

其中，$\mathring{s}_\tau$ 是被选中进行扩展的节点，$\boldsymbol{G}_s$ 是 s的期望自由能，$\mathbb{A}_{\mathring{s}_\tau}$ 是树中 $\mathring{s}_\tau$ 的所有祖先节点的集合。在反向传播过程中，我们还会按如下方式更新访问次数：

$$\forall s \in \mathbb{A}_{\mathring{s}_\tau} \cup \{\mathring{s}_\tau\}, \quad \boldsymbol{N}_s \leftarrow \boldsymbol{N}_s + 1. \tag{4}$$

如果令 $\boldsymbol{G}_s^{aggr}$表示任意节点 s 的聚合代价，该代价是在每次扩展后应用公式3得到的，那么现在就可以将 $\boldsymbol{G}_s$ 正式表示为：

$$\bar{\boldsymbol{G}}_s = \frac{\boldsymbol{G}_s^{aggr}}{\boldsymbol{N}_s}.$$

重要的是，如果节点 s 对应于通过执行动作$\hat{a}_\tau$从状态$\hat{s}_\tau$ 到达的状态，那么 $G(\hat{s}_\tau, \hat{a}_\tau) = \boldsymbol{G}_s$且 $N(\hat{s}_\tau, \hat{a}_\tau) = \boldsymbol{N}_s$。上述规划过程会在达到最大规划迭代次数时结束，或者在确定明显的胜出者时结束；也就是说，如果 $\max_{a_t} P(a_t) - \frac{1}{\#A} > T_{dec}$，其中#$A$ 是可能动作的数量，T_{dec} 是一个（阈值）超参数。

2.2.2 Derivation of the variational free energy

在本节中，我们推导 Fountas 等人（2020）使用的变分自由能。这一推导最初由 Millidge（2020）提出，并可加以调整，用于推导第3节所介绍模型的变分自由能。回顾一下，经典表述中的变分自由能旨在使近似后验分布 $Q_\phi(s_t, a_t)$ 尽可能接近真实后验分布[3]$P(s_t, a_t | o_t, s_{t-1}, a_{t-1})$，也就是说，

$$Q_\phi^*(s_t, a_t) = \underset{Q_\phi(s_t, a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P(s_t, a_t | o_t, s_{t-1}, a_{t-1})\right].$$

利用贝叶斯定理、期望的线性性，以及 $Q_\phi(s_t, a_t)$ 的积分为 1 这一事实：

$$\begin{aligned}
Q_\phi^*(s_t, a_t) &= \underset{Q_\phi(s_t, a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P(s_t, a_t | o_t, s_{t-1}, a_{t-1})\right] \\
&= \underset{Q_\phi(s_t, a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P(s_t, a_t, o_t, s_{t-1}, a_{t-1})\right] + \underbrace{\ln P(o_t, s_{t-1}, a_{t-1})}_{\text{Constant w.r.t } Q_\phi(s_t, a_t)} \\
&= \underset{Q_\phi(s_t, a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P(s_t, a_t, o_t, s_{t-1}, a_{t-1})\right].
\end{aligned}$$

3. 后验分布指的是一种条件分布，其中给定的变量是我们观测到特定值的变量。注意，s_{t-1} 的值未知，但可以从关于 s_{t-1} 的后验分布中采样（该分布来自上一个动作-感知循环）。

利用 d-分离准则（Koller 和 Friedman，2009），可以证明：

$$P(s_t, a_t, o_t, s_{t-1}, a_{t-1}) = P_{\theta_o}(o_t|s_t)P(a_t)P_{\theta_s}(s_t|s_{t-1}, a_{t-1})Q_\phi(s_{t-1}, a_{t-1}),$$

其中，$Q_\phi(s_{t-1},\ a_{t-1})$ 是通过前一时间步的推断过程得到的变分后验分布。在上述方程中，使用 $Q_\phi(s_{t-1},\ a_{t-1})$ 替代了 $P(s_{t-1},\ a_{t-1})$，即将 $Q_\phi(s_{t-1},\ a_{t-1})$ 用作经验先验。此外，由于 $Q_\phi(s_{t-1},\ a_{t-1})$ 相对于 $Q_\phi(s_t,\ a_t)$ 是常数，因此上述最小化问题可简化为：

$$\begin{aligned} Q^*_\phi(s_t, a_t) &= \underset{Q_\phi(s_t,a_t)}{\arg\min} \underbrace{D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P_{\theta_o}(o_t|s_t)P(a_t)P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right]}_{\text{variational free energy}} \\ &= \underset{Q_{\phi_a}(a_t|s_t)Q_{\phi_s}(s_t)}{\arg\min} \mathbb{E}_{Q_{\phi_s}(s_t)}\Big[D_{\mathrm{KL}}\left[\, Q_{\phi_a}(a_t|s_t) || \, P(a_t)\right]\Big] + D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right] \\ &\qquad - \mathbb{E}_{Q_{\phi_s}(s_t)}\Big[\ln P_{\theta_o}(o_t|s_t)\Big]. \end{aligned} \tag{5}$$

比较式 (5) 中的变分自由能与式 (2) 中的期望自由能，可以发现二者存在不一致之处。具体而言，在期望自由能的定义中，参数被视为潜变量，参见式 (2) 中的 θ；但在变分自由能中，参数则被视为神经网络的参数，参见式 5 中的 θ_s 和 θ_o。请注意，θ 不可能既是参数（即参数、向量或矩阵），又是随机变量；若 θ 是随机变量，就必须定义其概率密度，即 $P(\theta)$。此外，这种不一致也引出了一个问题：期望自由能是否真的是变分自由能的期望。总而言之，该 DAI_{MC} 智能体配备了四个深度神经网络，分别对 $Q_{\phi_a}(a_t|s_t)$、$Q_{\phi_s}(s_t)$、$P_{\theta_s}(s_t|s_{t-1},\ a_{t-1})$ 和 $P_{\theta_o}(o_t|s_t)$ 进行建模。这些网络的权重通过反向传播进行优化，以最小化式 (5) 给出的变分自由能。请注意，式 (5) 可分解为两个能够解析计算的 KL 散度项，而其中的期望可以通过蒙特卡洛估计近似得到。此外，由于 $P_{\theta_o}(o_t|s_t)$ 被建模为伯努利分布的乘积，$P_{\theta_o}(o_t|s_t)$ 的对数可化简为二元交叉熵。

2.2.3 Independence assumptions and the expected free energy

方程 (2) 中给出的期望自由能需要重新整理，因为它不易求值。因此，我们介绍 Fountas 等人（2020）提出的推导过程。接着，我们指出其中使用了两个独立性假设，即 $s_\tau \perp\!\!\!\perp \theta \mid \pi$ 和 $s_\tau \perp\!\!\!\perp \theta \mid \pi,\ o_\tau$，但并未明确给出证明。最后，我们提出一种不需要这两个假设、且结果更简单的替代推导。根据概率乘法法则，可以看出 $Q(s_\tau,\ \theta|\pi) = Q(\theta|s_\tau,\ \pi)Q(s_\tau|\pi)$ 和 $\tilde{}P(o_\tau,\ s_\tau,\ \theta|\pi) = P(o_\tau|\pi)P(s_\tau|o_\tau,\ \pi)P(\theta|s_\tau,\ o_\tau,\ \pi)$。利用这两个因式分解，式 (2) 中给出的期望自由能，即，

$$G_\tau(\pi) = \mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau, \theta|\pi) - \ln \tilde{P}(o_\tau, s_\tau, \theta|\pi)\Big],$$

$\tilde{}$其中 $Q = Q(o_\tau,\ s_\tau,\ \theta|\pi)$，并可重新排列如下：

$$\begin{aligned} G_\tau(\pi) = &- \mathbb{E}_{\tilde{Q}}\Big[\ln \tilde{P}(o_\tau|\pi)\Big] \\ &+ \mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln \tilde{P}(s_\tau|o_\tau, \pi)\Big] \\ &+ \mathbb{E}_{\tilde{Q}}\Big[\ln Q(\theta|s_\tau, \pi) - \ln \tilde{P}(\theta|s_\tau, o_\tau, \pi)\Big]. \end{aligned} \tag{6}$$

请注意，上述推导遵循 Fountas 等人（2020年）的工作。

根据 Fountas 等人（2020年）对方程 (6) 的第二项重新排列

首先，将方程（6）的第二项重新整理为存在解析解的熵项。在 Fountas 等人（2020年）的补充材料中，推导过程如下：

$$\begin{aligned}\mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln \tilde{P}(s_\tau|o_\tau,\pi)\Big] &\triangleq \mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln Q(s_\tau|o_\tau,\pi)\Big] \\ &= \mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)Q(o_\tau|s_\tau,\theta,\pi)}\Big[\ln Q(s_\tau|\pi) - \ln Q(s_\tau|o_\tau,\pi)\Big] \\ &= \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(s_\tau|\theta,\pi)}[\ln Q(s_\tau|\pi)] - \mathbb{E}_{Q(s_\tau|\theta,\pi)Q(o_\tau|s_\tau,\theta,\pi)}[\ln Q(s_\tau|o_\tau,\pi)]\Big] \\ &= \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(s_\tau|\theta,\pi)}[\ln Q(s_\tau|\pi)] - \mathbb{E}_{Q(o_\tau|\theta,\pi)Q(s_\tau,|o_\tau,\theta,\pi)}[\ln Q(s_\tau|o_\tau,\pi)]\Big],\end{aligned} \tag{7}$$

其中，在第一行中，一个分布被重新命名，即 $\tilde{P}(s_\tau|o_\tau,\pi) \triangleq Q(s_\tau|o_\tau,\pi)$。推导的下一步（参见 Fountas 等人（2020年）的补充材料）将这一最终表达式重新排列如下：

$$\mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln Q(s_\tau|o_\tau,\pi)\Big] = \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(o_\tau|\theta,\pi)}[H[Q(s_\tau|o_\tau,\pi)]] - H[Q(s_\tau|\pi)]\Big].$$

然而，上述方程假设 $s_\tau \perp\!\!\!\perp \theta \mid \pi$ 和 $s_\tau \perp\!\!\!\perp \theta \mid \pi, o_\tau$。换言之，部分关于 θ 的条件已被省略，即：

$$\begin{aligned}&\mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(s_\tau|\theta,\pi)}[\ln Q(s_\tau|\pi)] - \mathbb{E}_{Q(o_\tau|\theta,\pi)Q(s_\tau,|o_\tau,\theta,\pi)}[\ln Q(s_\tau|o_\tau,\pi)]\Big] \quad \text{(last expression of derivation 7)} \\ &\neq \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(s_\tau|\pi)}[\ln Q(s_\tau|\pi)] - \mathbb{E}_{Q(o_\tau|\theta,\pi)Q(s_\tau|o_\tau,\pi)}[\ln Q(s_\tau|o_\tau,\pi)]\Big] \\ &= \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(o_\tau|\theta,\pi)}\big[H[Q(s_\tau|o_\tau,\pi)]\big] - H[Q(s_\tau|\pi)]\Big].\end{aligned}$$

是否可以省略这些条件，取决于分布的因子分解。换言之，必须使用 d-分离准则检验这两个假设（即 $s_\tau \perp\!\!\!\perp \theta \mid \pi$ 和 $s_\tau \perp\!\!\!\perp \theta \mid \pi, o_\tau$）。然而，这很难做到，因为如前所述，参数 θ 在 (2) 中是潜变量，但在 (5) 中被视为参数，这使得图模型不明确。我们不尝试证明 $s_\tau \perp\!\!\!\perp \theta \mid \pi$ 和 $s_\tau \perp\!\!\!\perp \theta \mid \pi, o_\tau$，而是提出一种不需要此类独立性假设的替代推导。

方程 （6）第二项的另一种推导

从方程（6）的第二项重新出发，我们可以重新排列如下 s:

$$\begin{aligned}\mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln \tilde{P}(s_\tau|o_\tau,\pi)\Big] &\triangleq \mathbb{E}_{\tilde{Q}}\Big[\ln Q(s_\tau|\pi) - \ln Q(s_\tau|o_\tau,\pi)\Big] \\ &= \mathbb{E}_{Q(s_\tau|\pi)Q(\theta,o_\tau|s_\tau,\pi)}\Big[\ln Q(s_\tau|\pi)\Big] - \mathbb{E}_{Q(o_\tau|\pi)Q(s_\tau|o_\tau,\pi)Q(\theta|s_\tau,o_\tau,\pi)}\Big[\ln Q(s_\tau|o_\tau,\pi)\Big] \\ &= \mathbb{E}_{Q(s_\tau|\pi)}\Big[\ln Q(s_\tau|\pi)\Big] - \mathbb{E}_{Q(o_\tau|\pi)Q(s_\tau|o_\tau,\pi)}\Big[\ln Q(s_\tau|o_\tau,\pi)\Big] \\ &= \mathbb{E}_{Q(o_\tau|\pi)}\Big[H[Q(s_\tau|o_\tau,\pi)]\Big] - H[Q(s_\tau|\pi)],\end{aligned}$$

其中第一行将分布重新命名，即 $\tilde{P}(s_\tau|o_\tau,\pi) \triangleq Q(s_\tau|o_\tau,\pi)$时，从第一行到第二行使用了两种不同的因式分解，第一行与第二行之间运用了期望的线性性；而第二行到第三行省略了关于 θ 的期望，因为常数的期望就是常数本身。重要的是，上述推导没有作出任何独立性假设，并得出了更简单的结果。这种替代推理方式很有益，因为它得出了依赖更少假设的、更有力的推导。该推导得出的更简单结果也具有实际影响。事实上，关于 $Q(\theta|\pi)$ 的期望消失了，而关于 $Q(o_\tau|\theta,\pi)$ 的期望现在变成关于 $Q(o_\tau|\pi)$ 的期望。这两项变化表明，这一项需要采用不同的实现方式。

重新排列 Fountas 等人（2020年）方程 （6）的第三项

为求完整起见，我们现在聚焦于式（6）的第三项，它可以重新排列如下：

$$\begin{aligned}\mathbb{E}_{\tilde{Q}}\Big[\ln Q(\theta|s_\tau,\pi)-\ln\tilde{P}(\theta|s_\tau,o_\tau,\pi)\Big] &\triangleq \mathbb{E}_{\tilde{Q}}\Big[\ln Q(\theta|s_\tau,\pi)-\ln Q(\theta|s_\tau,o_\tau,\pi)\Big]\\ &= \mathbb{E}_{\tilde{Q}}\Big[\ln Q(o_\tau|s_\tau,\pi)-\ln Q(o_\tau|s_\tau,\theta,\pi)\Big],\end{aligned}$$

˜其中， $P(\theta|s_\tau,\ o_\tau,\ \pi)$ 被重命名为 $Q(\theta|s_\tau,\ o_\tau,\ \pi)$，并使用贝叶斯定理得到：

$$Q(\theta|s_\tau,\pi)=\frac{Q(\theta|s_\tau,o_\tau,\pi)Q(o_\tau|s_\tau,\pi)}{Q(o_\tau|s_\tau,\theta,\pi)}\Leftrightarrow\frac{Q(\theta|s_\tau,\pi)}{Q(\theta|s_\tau,o_\tau,\pi)}=\frac{Q(o_\tau|s_\tau,\pi)}{Q(o_\tau|s_\tau,\theta,\pi)}.$$

˜最后，回顾 $Q=Q(o_\tau,\ s_\tau,\ \theta|\pi)=Q(\theta|\pi)Q(s_\tau|\theta,\ \pi)Q(o_\tau|s_\tau,\ \theta,\ \pi)$，并利用期望的线性性，我们得到：

$$\begin{aligned}\mathbb{E}_{\tilde{Q}}\Big[\ln Q(\theta|s_\tau,\pi)-\ln Q(\theta|s_\tau,o_\tau,\pi)\Big] &= \mathbb{E}_{Q(o_\tau,s_\tau|\pi)}\Big[\ln Q(o_\tau|s_\tau,\pi)\Big]+\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big]\\ &= \mathbb{E}_{Q(o_\tau|s_\tau,\pi)Q(s_\tau|\pi)}\Big[\ln Q(o_\tau|s_\tau,\pi)\Big]+\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big]\\ &= -\mathbb{E}_{Q(s_\tau|\pi)}\Big[H\big[Q(o_\tau|s_\tau,\pi)\big]\Big]+\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big],\end{aligned}$$

其中，由于 $\ln\ Q(o_\tau|s_\tau,\ \pi)$ 相对于 θ 是常数，因此我们能够使用：

$$\mathbb{E}_{Q(o_\tau,\theta,s_\tau|\pi)}[\ln Q(o_\tau|s_\tau,\pi)]=\mathbb{E}_{Q(o_\tau,s_\tau|\pi)}[\ln Q(o_\tau|s_\tau,\pi)].$$

总而言之，该推导给出了一个基于以下两个熵项的表达式： $H[Q(o_\tau|s_\tau,\ \pi)]$和 $H[Q(o_\tau|s_\tau,\ \theta,\ \pi)]$,作者声称可以对其进行估计（更多详细信息请参见附录 B）。请注意，我们提出的 EFE 替代方案（见下文）使用了这一推导来处理方程（6）的第三项。

Fountas 等人（2020年）的期望自由能

如果遵循 Fountas 等人（2020年）提出的推导，那么期望自由能为：

$$\begin{aligned}G_\tau(\pi)=&-\mathbb{E}_{\tilde{Q}}\Big[\ln\tilde{P}(o_\tau|\pi)\Big]\\ &+\mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(o_\tau|\theta,\pi)}\big[H[Q(s_\tau|o_\tau,\pi)]\big]-H[Q(s_\tau|\pi)]\Big]\\ &+\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big]-\mathbb{E}_{Q(s_\tau|\pi)}\Big[H\big[Q(o_\tau|s_\tau,\pi)\big]\Big].\end{aligned}\tag{8}$$

请注意，在第 2.2 节中，我们重点介绍了 Fountas 等人（2020年）给出的方法，并进行了一些调整以保持一致性。附录 B 中详细介绍了方程 8 的实现，以及论文与代码之间存在的一些差异。

我们提出的期望自由能替代方案

如果遵循我们提出的替代推导，那么期望自由能为：

$$\begin{aligned}G_\tau(\pi)=&-\mathbb{E}_{\tilde{Q}}\Big[\ln\tilde{P}(o_\tau|\pi)\Big]\\ &+\mathbb{E}_{Q(o_\tau|\pi)}\Big[H[Q(s_\tau|o_\tau,\pi)]\Big]-H[Q(s_\tau|\pi)],\\ &+\mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big]-\mathbb{E}_{Q(s_\tau|\pi)}\Big[H\big[Q(o_\tau|s_\tau,\pi)\big]\Big].\end{aligned}$$

最后， DAI_{MC} 确实解决了 dSprites环境。

2.3 DAI_{VPG} 智能体（Millidge，2020年）

在本节中，我们将解释并讨论 Millidge（2020年）的方法。代码可在以下网址获取：https://github.com/BerenMillidge/DeepActiveInference。需要注意的是，尽管论文中的数学基于部分可观测马尔可夫决策过程（POMDP）的形式化框架，但代码并未实现编码器/解码器架构，这意味着代码实现的是完全可观测决策过程，即 MDP。此外，DAI_{VPG} 由三个神经网络组成，如图 4 和图 5 所示。第一个是转移网络，根据当前观测和动作预测未来观测，即 $\mathring{o}_{\tau+1} = \mathcal{T}_{\theta_o}(o_\tau, a_\tau)$。第二个是策略网络，用于对动作上的变分分布 $Q_{\phi_a}(a_\tau|o_\tau)$ 进行建模。第三个是评论家网络，根据当前观测预测每个动作的期望自由能。此外，Millidge（2020年）将动作先验定义如下：

$$P(a_\tau|o_\tau) = \sigma[-\zeta G(o_\tau, a_\tau)],$$

其中，ζ是动作先验的精度，$\sigma[\cdot]$是一个 softmax 函数，而 $G(o_\tau, a_\tau)$ 是在观测到 o_τ 时采取动作 a_τ 的期望自由能（期望自由能）。在论文中，数学推导基于 POMDP 形式体系。因此，$G(o_\tau, a_\tau)$ 记作 $G(s_\tau, a_\tau)$，定义如下：

$$G(s_\tau, a_\tau) = -r_\tau + \underbrace{D_{\mathrm{KL}}\left[\, Q(s_\tau) || \, Q(s_\tau|o_\tau)\right]}_{\text{intrinsic value}} + \hat{\mathcal{G}}_{\hat{\theta}_a}(a_{\tau+1}, s_{\tau+1}), \tag{9}$$

ˆ 其中，r_τ是智能体在时间步 τ 获得的奖励，而 $\mathcal{G}_{\hat{\theta}_a}(a_{\tau+1}, s_{\tau+1})$ 是目标网络（即评论家网络的副本，其权重每经过 K 次学习迭代便同步一次）。现在请记住，在该实现中，没有编码器 $Q(s_\tau)$，也没有解码器 $P(o_\tau|s_\tau)$。换言之，不存在隐状态 s_τ，因此内在价值是如何计算的并不确定。GitHub[4] 上以下网址提供的代码：https://github.com/BerenMillidge/ DeepActiveInference，文件 active_inference_with_Tmodel.jl（见第 51 行）表明使用了以下方程：

$$\text{intrinsic value} = \sum_i \left[o_{\tau+1}[i] - \mathring{o}_{\tau+1}[i]\right]^2, \tag{10}$$

其中，$o_{\tau+1}[i]$ 是时间步 $\tau + 1$收到的第 i 个观测，而 $\mathring{o}_{\tau+1}[i]$ 是转移网络预测的第 i 个观测（在时间步 $\tau + 1$）的（数值）值。更正式地说，上述对内在值的表述对应于两个均具有单位协方差矩阵的高斯分布之间的 KL 散度，即：

$$\text{intrinsic value} = D_{\mathrm{KL}}\left[\, Q(o_{\tau+1}) || \, P(o_{\tau+1}|o_\tau, a_\tau)\right] = \sum_i \left[o_{\tau+1}[i] - \mathring{o}_{\tau+1}[i]\right]^2,$$

其中 $P(o_{\tau+1}|o_\tau, a_\tau) = \mathcal{N}(o_{\tau+1}; \mathring{o}_{\tau+1}, I)$，而 $Q(o_{\tau+1})$ 是一个高斯分布，其均值向量为 $o_{\tau+1}$，协方差矩阵为单位协方差矩阵。不过请注意，(9) 是 POMDP 场景下期望自由能的定义。Costa 等人（2020b）解释道，MDP 场景下的期望自由能定义如下：

$$G(a_{t:T-1}, o_t) \approx \sum_{\tau=t+1}^{T} D_{\mathrm{KL}}\left[\, P(o_\tau|a_{t:T-1}, o_t) || \, P(o_\tau)\right],$$

4. 我们 指的是 2022年6月6日 GitHub 上提供的代码版本。

其中 $P(o_\tau)$ 是智能体的先验偏好（与强化学习中的奖励相关），而$P(o_\tau|a_{t:T-1},\ o_t)$ 是转移映射。重要的是，这一期望自由能的定义不像 (9) 那样分解为外在项和内在项。因此，（就目前而言），DAI_{VPG} 智能体的实现混合了 POMDP 和 MDP 场景：生成模型对应于 MDP，而期望自由能则改编自 POMDP 场景。

我们通过讨论转移网络、策略网络和评论家网络的训练过程来结束本节。如论文中所述，转移网络的训练目标是最小化变分自由能。此外，鉴于上文提到的高斯假设（采用单位协方差矩阵），KL 散度可简化为均方误差（MSE）。因此，转移网络会通过最小化智能体在时间 $\tau+1$ 观测到的内容与转移网络预测的观测 ($\mathring{o}_{\tau+1}$) 之间的 MSE 来更新，即:

$$\theta_o^* = \arg\min_{\theta_o} \text{MSE}\Big[o_{\tau+1}, \mathring{o}_{\tau+1}\Big],$$

其中 $\mathring{o}_{\tau+1} = \mathcal{T}_{\theta_o}(o_\tau,\ a_\tau)$。策略网络的训练目标是最小化动作变分后验 $Q_{\phi_a}(a_\tau|o_\tau)$ 与动作先验 $P(a_\tau|o_\tau)$ 之间的 KL 散度，即:

$$\phi_a^* = \arg\min_{\phi_a} D_{\text{KL}}\left[\, Q_{\phi_a}(a_\tau|o_\tau) || \, P(a_\tau|o_\tau)\right],$$

这会使变分自由能最小化。最后，评论家网络通过最小化式 (9) 中定义的目标期望自由能与评论家网络输出 $\mathcal{G}_{\theta_a}(o_\tau,\ \bullet)$ 之间的 MSE 来训练:

$$\theta_a^* = \arg\min_{\theta_a} \text{MSE}\Big[G(o_\tau, \bullet), \mathcal{G}_{\theta_a}(o_\tau, \bullet)\Big].$$

DAI_{VPG} 能够解决CartPole环境。

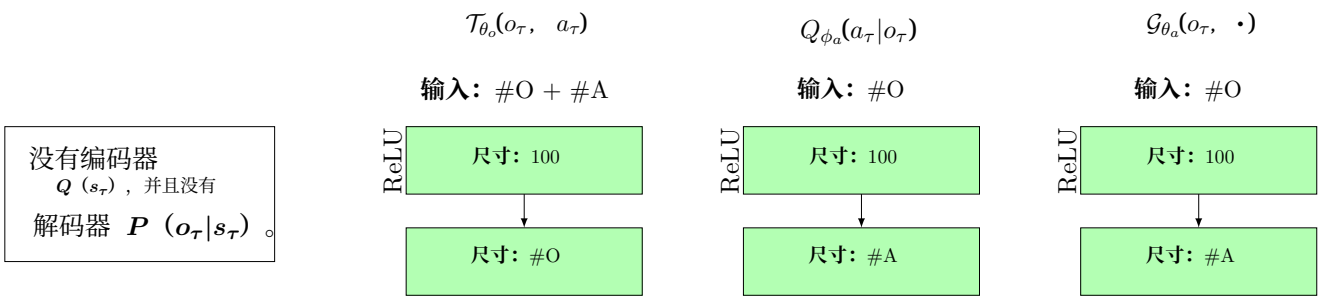


图4: DAI_{VPG} 智能体的神经网络架构。绿色模块对应全连接层。第一个神经网络是转移网络，它以时间步 τ 的观测和动作作为输入，并输出时间步 $\tau+1$ 观测的高斯分布均值。第二个神经网络是策略网络，用于建模动作的变分后验。第三个神经网络是评论家网络，它以观测作为输入，并输出每个动作的期望自由能。

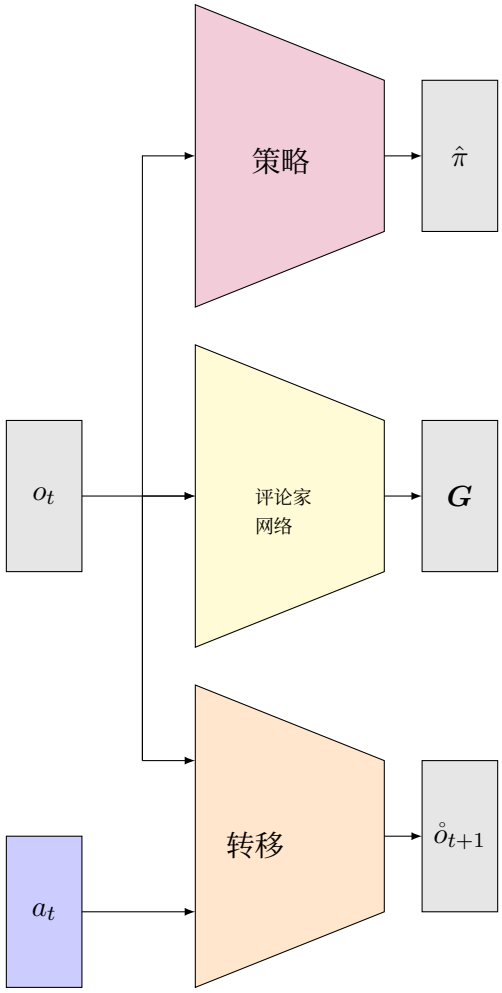


图 5：此图展示了 DAI_{VPG} 智能体。唯一新增的部分是策略网络，它以时间 t 的隐藏状态作为输入，并输出动作变分后验的参数 $\hat{}\pi$ 。重要的是， DAI_{VPG} 根据期望自由能（EFE）采取动作。

2.4 DAI_{RHI} 智能体（Rood 等人，2020年）

本节将解释并讨论 Rood 等人（2020）提出的方法。简而言之，本文提出了一种变分自编码器（VAE），能够解释在橡胶手错觉（RHI）实验中观察到的结果。在 Rood 等人（2020）的实验中，智能体（即人或计算机）能够在三维空间中移动手臂。然而，智能体无法观测到手臂的真实位置，而是看到一只位于不同位置的人造手。这可以通过虚拟现实（针对人类）或模拟器（针对计算机）来实现。由于 Rood 等人（2020）将研究范围限定在变分自编码器上，因此这种方法不能被视为深度主动推理的完整实现。更确切地说，其中缺少转移网络和评论家网络（或策略网络）。

2.5 DAI_{HR} 智能体（Sancaktar 等人，2020年；Lanillos 等，2020年；Oliver 等，2019 ）

在本节中，我们将解释并讨论以下方法：Sancaktar 等人（2020年）、Lanillos 等人（2020年）和 Oliver 等人（2019）。简而言之，这些论文提出了一种基于单个解码器网络的自由能最小化方案，分别用于控制 Nao、TIAGo 和 iCub 机器人。由于，

Sancaktar 等人（2020年）、Lanillos 等人（2020年）和 Oliver 等人（2019）将自身限制在单个解码器的背景下，因此，这种方法不能被视为深度主动推理的完整实现。更确切地说，其中缺少编码器、转移和评论家网络（或策略网络）。

2.6 DAI_{FA} 智能体（Ueltzh¨offer，2018）

本节回顾 Ueltzh¨offer（2018）提出的方法。本文的原始代码可在 GitHub 上通过以下网址获取：https://github.com/kaiu85/deepAI_paper。该方法由四个深度神经网络组成。编码器 $\mathcal{E}_{\phi_s}$ 将状态上的近似后验 $Q_{\phi_s}(s_t|s_{t-1}, o_t)$ 建模为高斯分布，即 $Q_{\phi_s}(s_t|s_{t-1}, o_t) = \mathcal{N}(s_t;\ \mu, \sigma)$，其中$\mu, \sigma = \mathcal{E}_{\phi_s}(s_{t-1}, o_t)$。解码器 $\mathcal{D}_{\theta_o}$将似然映射 $P_{\theta_o}(o_\tau|s_\tau)$ 建模为高斯分布，即 $P_{\theta_o}(o_\tau|s_\tau) = \mathcal{N}(o_\tau;\ \mu_o, \sigma_o)$，其中 $\mu_o, \sigma_o = \mathcal{D}_{\theta_o}(s_\tau)$。转移网络 $\mathcal{T}_{\theta_s}$将转移映射 $P_{\theta_s}(s_\tau|s_{\tau-1})$ 建模为高斯分布，即 $P_{\theta_s}(s_\tau|s_{\tau-1}) = \mathcal{N}(s_\tau;\ \mathring{\mu}, \mathring{\sigma})$，其中 $\mathring{\mu}$, $\mathring{\sigma} = \mathcal{T}_{\theta_s}(s_{\tau-1})$。请注意，转移网络仅以先前状态为条件。这是因为动作包含在解码器预测的观测中。具体而言，实验在 MountainCar 环境中进行，这意味着智能体观测的是汽车的 x 位置 o^x_τ。此外，根据本体感觉的理念，智能体会观测自身的动作，即 $o^a_{\tau-1} = a_{\tau-1}$，其中 $a_{\tau-1}$是智能体在时间 $\tau - 1$执行的动作。下文中，我们令 $o_\tau = (o^x_\tau, o^a_{\tau-1})$ 表示汽车的 x 位置与智能体采取的动作的拼接。重要的是，由于o_τ 包含 $o^a_{\tau-1}$，潜在空间必须（隐式地）对动作进行编码，解码器才能成功预测观测。最后，策略网络 $\mathcal{P}_{\theta_a}$ 将动作先验 $P_{\theta_a}(a_\tau|s_\tau)$ 建模为高斯分布，即 $P_{\theta_a}(a_\tau|s_\tau) = \mathcal{N}(a_\tau;\ \mu_a, \sigma_a)$，其中 $\mu_a, \sigma_a = \mathcal{P}_{\theta_a}(s_\tau)$。图 6 展示了这些深度神经网络的架构。随后，Ueltzh¨offer（2018）将自由动作目标定义为随时间累积的变分自由能：

$$FA(o_{1:T}, \phi, \theta) = \sum_{\tau=1}^{T} \Bigg[\overbrace{-\mathbb{E}_{Q_{\phi_s}(s_\tau|s_{\tau-1}, o_\tau)}\Big[\ln P_{\theta_o}(o_\tau|s_\tau)\Big]}^{\text{accuracy}} + \overbrace{D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_\tau|s_{\tau-1}, o_\tau) || \, P_{\theta_s}(s_\tau|s_{\tau-1})\right]}^{\text{complexity}} \Bigg]_{\mathrm{VFE}_\tau},$$

其中，$s_0 = (0, \ldots, 0)$ 是表示初始隐藏状态的零向量，T是时间范围，$P_{\theta_o}(o_\tau|s_\tau)$、$Q_{\phi_s}(s_\tau|s_{\tau-1}, o_t)$ 和 $P_{\theta_s}(s_\tau|s_{\tau-1})$ 分别使用高斯分布进行建模，其参数分别由解码器、编码器和转移网络预测。图7展示了自由动作目标的计算过程，以及智能体的动作-感知循环。第一个动作-感知循环开始于将初始隐藏状态 s_0 输入策略网络，策略网络输出动作高斯分布的参数。随后，从该高斯分布中采样动作 $\hat{}a_0$，并在环境中执行，从而得到新的观测 o^x_1。接着，将动作 $\hat{}a_0$ 与 o^x_1 拼接，形成 o_1。然后，将观测 o_1 和状态 s_0 输入编码器，编码器输出关于 $\hat{}s_1$的高斯分布参数。最后，从该高斯分布中采样一个状态，并将其作为下一次动作-感知循环的输入。该过程持续进行，直到达到时间范围。

在每个动作-感知循环中，都会计算该时间步的变分自由能。为了计算变分自由能$_\tau$，将状态 s_τ 输入转移网络和编码器。这两个网络都会输出关于 $s_{\tau+1}$的高斯分布参数。从编码器预测的分布中采样一个状态，并将其作为输入传给解码器；解码器会输出关于 $o_{\tau+1}$的高斯分布参数。最后，使用关于 $o_{\tau+1}$ 的高斯分布参数计算准确度项，并使用关于 $s_{\tau+1}$ 的两个高斯分布的参数计算复杂度项。

现在，我们将重点关注智能体的先验偏好。通常，先验偏好是期望自由能的一部分。然而，Ueltzh¨offer（2018）采用了不同的方法。回想一下，潜变量 s_τ 是

使用多元高斯分布建模的。 DAI_{FA} 智能体将潜在空间的第一个维度保留用于对先验偏好进行编码。具体而言，转移网络会预测一个关于潜在状态的多元高斯分布的均值向量和协方差矩阵的对角线（即另一个向量）。均值向量的第一个元素被固定为目标 x 位置，而方差向量的第一个元素被设为相对较小的值。这实际上会推动智能体朝目标位置移动。此外，编码器还会预测另一组均值向量和方差向量。编码器预测的均值向量的第一个元素被固定为智能体当前观测到的 x 位置，而方差向量的第一个元素被设为相对较小的值。需要注意的是，固定转移网络预测的均值向量和方差向量的第一个元素的值并无争议，即这只是生成模型的定义方式。然而，固定编码器预测的均值向量和方差向量的第一个元素的值则可能引发争议。具体而言，编码器本应预测变分分布，也就是对真实后验分布的近似。但固定编码器预测的均值向量和方差向量的第一个元素的值，很可能会使变分后验偏离真实后验更多。

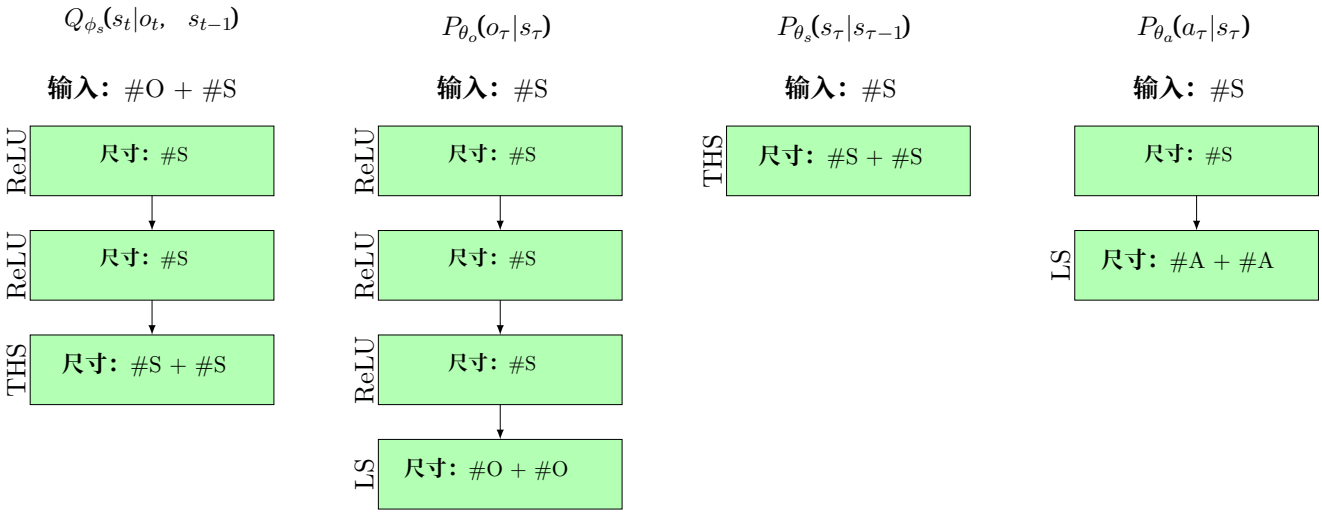


图 6： DAI_{FA} 智能体的神经网络架构。绿色模块对应全连接层。第一个神经网络是编码器，它以时间 $t-1$ 的状态和时间 t的观测作为输入，并输出时间 t的状态分布参数。第二个神经网络是解码器，它以时间 τ的状态作为输入，并输出时间 τ的观测分布参数。第三个是转移网络，它以时间步 $\tau-1$的状态作为输入，并输出时间步 τ的状态分布参数。第四个神经网络是策略网络，用于建模动作先验；也就是说，策略以时间 τ 的状态作为输入，并输出时间步 τ的动作分布参数。最后，THS 表示双曲正切和 Softplus，即前半部分神经元采用双曲正切激活函数，后半部分采用 Softplus 激活函数；LS 表示线性激活函数和 Softplus，即前半部分神经元采用线性激活函数，后半部分采用 Softplus 激活函数。

DAI_{FA} 智能体有几个重要特点。首先，它不使用期望自由能，而是通过训练智能体，使其随时间推移最小化累积变分自由能。其次，这种方法会沿时间展开部分可观测马尔可夫决策过程。换言之，代码会构建一个庞大的计算图，其中包含每个动作-感知循环对应的编码器、解码器、转移网络和策略网络。因此，这种方法的计算开销很大，当时间范围较长时，甚至会变得难以

处理。第三， DAI_{FA} 要求建模者将先验偏好编码到编码器和转移网络预测的分布中。这可能会限制该方法的适用性。事实上，如前文所述，对于 MountainCar 问题，可以将智能体的先验偏好编码在潜在空间的第一个维度中。

然而，在潜在空间中手动编码先验偏好有两个主要缺点。首先，模型需要针对不同环境进行修改。这是因为每个环境中，智能体的先验偏好都不相同。其次，对于某些环境，先验偏好该如何定义并不明确。例如，玩吃豆人时，智能体需要吃掉所有豆子，同时躲避幽灵。如何将这些要求编码到模型的潜在空间中？这尤其具有挑战性，因为智能体唯一获得的观测是游戏图像，也就是说，智能体无法直接获取吃豆人和幽灵的位置。

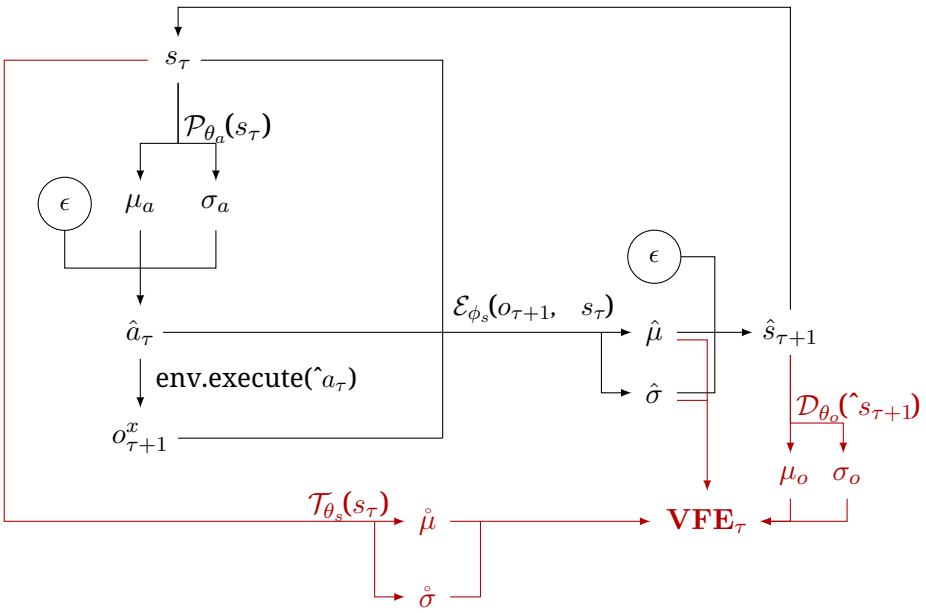


图7：行动—感知循环（黑色）和自由动作目标的估计（红色）。请注意， $\mathring{\mu}$、 $\mathring{\sigma}$、 $\hat{}\mu$ 和 $\hat{}\sigma$ 用于计算变分自由能的复杂度项，而 μ_o 和 σ_o 用于计算变分自由能的准确度项。

2.7 DAI_{POMDP} 智能体（van der Himst 和 Lanillos，2020年）

本节回顾 van der Himst 和 Lanillos（2020年）提出的方法。代码可在此处获取：https://github.com/Grottoh/Deep-Active-Inference-for-Partially-Observable-MDPs。 DAI_{POMDP} 智能体由五个深度神经网络组成。

解码器 $\mathcal{D}_{\theta_o}$ 将 $P_{\theta_o}(o_\tau|s_\tau)$ 建模为伯努利分布的乘积，因此： $P_{\theta_o}(o_\tau|s_\tau)$ =$\mathcal{B}$伯努利分布$(o_\tau; \hat{}o_\tau)$，其中 $\hat{}o_\tau = \mathcal{D}_{\theta_o}(s_\tau)$。转移网络 $\mathcal{T}_{\theta_s}$ 将 $P_{\theta_s}(s_{\tau+1}|s_\tau, a_\tau)$ 建模为高斯分布，即 $P_{\theta_s}(s_{\tau+1}|s_\tau, a_\tau) = \mathcal{N}(s_{\tau+1}|\mathring{\mu}, \mathring{\sigma})$，其中 $\mathring{\mu}$， $\ln \mathring{\sigma} = \mathcal{T}_{\theta_s}(s_\tau, a_\tau)$。评论家网络 $\mathcal{G}_{\theta_a}$ 输出一个向量，其中包含每个动作的预测期望自由能，并据此将动作先验定义为 $P_{\theta_a}(a_\tau|s_\tau) = \sigma[-\zeta\mathcal{G}_{\theta_a}(s_\tau, \bullet)],$，其中 $\sigma[\bullet]$ 是 softmax 函数， ζ是动作先验的精度，而 $\mathcal{G}_{\theta_a}(s_\tau, \bullet)$ 是评论家网络在状态 s_τ 作为输入时预测的每个动作的期望自由能。状态的变分后验 $Q_{\phi_s}(s_t)$ 是由编码器 $\mathcal{E}_{\phi_s}$建模的高斯分布，即 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t; \mu, \sigma)$，其中 μ， $\ln \sigma = \mathcal{E}_{\phi_s}(o_t)$。动作的变分后验 $Q_{\phi_a}(a_t|s_t)$ 是由策略网络 $\mathcal{P}_{\phi_a}$建模的分类分布，即 $Q_{\phi_a}(a_t|s_t)$ =分类分布$(a_t; \hat{}\pi)$，其中 $\hat{}\pi = \mathcal{P}_{\phi_a}(s_t)$。然后，智能体需要最小化定义如下的变分自由能

如下：

$$Q^*_\phi(s_t, a_t) = \underset{Q_\phi(s_t,a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_{\phi_a}(a_t|s_t)Q_{\phi_s}(s_t) || \, P_{\theta_o}(o_t|s_t)P_{\theta_s}(s_t|s_{t-1}, a_{t-1})P_{\theta_a}(a_t|s_t)\right]$$
$$= \underset{Q_\phi(s_t,a_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right] + D_{\mathrm{KL}}\left[\, Q_{\phi_a}(a_t|s_t) || \, P_{\theta_a}(a_t|s_t)\right] - \mathbb{E}_{Q_{\phi_s}(s_t)}[\ln P_{\theta_o}(o_t|s_t)].$$

然而，正如论文中所解释的，状态上的 KL 散度被均方误差（MSE）替代如下：

$$Q^*_\phi(s_t, a_t) = \underset{Q_\phi(s_t,a_t)}{\arg\min}\, \mathrm{MSE}(\mu, \mathring{\mu}) + D_{\mathrm{KL}}\left[\, Q_{\phi_a}(a_t|s_t) || \, P_{\theta_a}(a_t|s_t)\right] - \mathbb{E}_{Q_{\phi_s}(s_t)}[\ln P_{\theta_o}(o_t|s_t)],$$

其中，μ 和 $\mathring{\mu}$ 分别是由编码器和转移网络预测的均值向量。论文称，使用最大后验（MAP）估计来计算状态预测误差，而不是使用密度之间的 KL 散度，以此为这种替代提供依据。然而，状态预测误差与状态上的 KL 散度是两个不同的量，只有当状态上的两个密度均为具有单位协方差矩阵的高斯分布时，它们才相等。然而，编码器网络预测的分布并不具有单位协方差矩阵。

简单来说，在此情境下，均方误差与状态密度之间的 KL 散度不一定等价。因此，DAI_{POMDP} 智能体可能不会始终遵循自由能原理。

2.8 DAI_{SSM} 智能体（van der Himst 和 Lanillos，2020年）

由 (C.atal 等人，2020年) 提出的深度主动推理智能体基于状态空间模型，因此被称为 DAI_{SSM}。该方法的代码未在网上公开，但我们成功从作者处获取了代码。重要的是，DAI_{SSM} 这是一种离线方法，意味着模型首先在固定数据集上进行训练，该数据集通过在环境中执行随机动作或手动控制机器人来收集。模型训练完成后，可以利用想象展开估计不同动作序列的期望自由能，并在环境中执行具有最低期望自由能的策略的第一个动作。

2.9 机器人操作的主动探索（Schneider 等人，2022年）

我们评述的最后一篇论文（Schneider 等人，2022年）以强化学习视角为出发点，在该视角下，智能体旨在最大化奖励。然而，智能体并非贪婪地最大化奖励，而是同时最大化模型参数与预期状态和奖励之间的信息增益，即：

$$\max_\pi \quad \underbrace{\mathbb{E}_{P(\boldsymbol{r}|\pi)}\big[f(\boldsymbol{r})\big]}_{\text{Expected reward}} + \beta \underbrace{\mathbb{E}_{P(\boldsymbol{s},\boldsymbol{r}|\pi,s_t)}\big[D_{\mathrm{KL}}\left[\, P(\theta|\boldsymbol{s},\boldsymbol{r},\pi,s_t) || \, P(\theta)\right]\big]}_{\text{Information gain}}, \quad \text{where:} \quad f(\boldsymbol{r}) = \sum_{\tau=t+1}^{t+H} r_\tau \qquad (11)$$

其中，t 是当前时间步，H 是规划的时间范围，π 是智能体策略，θ 是模型的参数，$\boldsymbol{r} = \boldsymbol{r}_{t+1:t+H}$ 是时间步 $t+1$ 与 $t+H$之间的奖励序列，$\boldsymbol{s} = \boldsymbol{s}_{t+1:t+H}$ 是时间步 $t+1$ 与 $t+H$之间的状态序列，β 是用于调节信息增益影响的超参数。请注意，智能体按照策略 π行动时所获得的奖励序列 $\boldsymbol{r}$ 的分布，即 $P(\boldsymbol{r}|\pi)$，并非确定性的。实际上，由于转移映射具有随机性，同一策略可能产生不同的状态轨迹，而不同状态又可能产生不同的奖励。请注意，相对于 $P(\boldsymbol{r}|\pi)$ 的期望会遍历所有可能的奖励序列，并将每个序列传入 $f(\boldsymbol{r})$。随后，作者展示了公式11与期望自由能之间的关系。该关系在（Schneider, N.D.）中有更详细的解释，并基于以下假设：

1. 对于所有 $\tau \in \{t+1, ..., t+H\}$，状态 s_τ和奖励 r_τ都是潜变量

2. 生成模型包含用于表示状态 o_τ^s 和奖励 o_τ^r的观测变量

3. 似然映射，即 $P(o_\tau^r|r_\tau)$ 和 $P(o_\tau^s|s_\tau)$，是狄拉克分布，这实际上使潜变量完全可观测；但模型参数 θ 除外——这些参数在时间步零时按照狄拉克分布分布，并在之后的时间内保持不变

4. 转移映射 $P(s_{\tau+1}|s_\tau,\ a_\tau,\ \theta)$ 的方差不依赖于参数 θ，例如，$P(s_{\tau+1}|s_\tau,\ a_\tau,\ \theta) = \mathcal{N}(s_{\tau+1};\ \boldsymbol{\mu},\ \boldsymbol{\sigma})$，其中 $\boldsymbol{\mu} = f_\theta(s_\tau,\ a_\tau)$ 由具有参数 θ 的神经网络进行预测，而 $\boldsymbol{\sigma} = \sigma I$ 是一个对角矩阵，其对角元素均等于 σ。

重要的是，虽然上述假设使作者能够从公式11推导出期望自由能，但这些假设也对模型施加了许多限制。例如，如果似然映射不是狄拉克分布，该证明便不成立。在本文中，我们关注此类模型。总而言之，这种方法是对应用于机器人控制的强化学习的一项重要贡献，但不能被视为完整的深度主动推理智能体。该方法的代码可在以下 URL 获取：https://github.com/TimSchneider42/aerm。

2.10 使用中心化核对齐衡量表征相似性

顾名思义，表征相似性指标旨在衡量两个表征之间的相似性。在深度学习的语境下，这些表征对应于 $\mathbb{R}^{n\times p}$ 激活值矩阵，其中 n 表示数据样本的数量，p 表示某一层中神经元的数量。在本文中，我们旨在使用此类指标比较第3节所述深度学习模型学习到的表征，以及深度Q网络学习到的表征。

在我们的分析中，我们将使用中心化核对齐（CKA）（Cortes 等，2012；Cristianini 等，2002）。CKA 是希尔伯特–施密特独立性准则（HSIC）（Gretton 等，2005）的归一化版本，用于衡量两个表征的 $n \times n$ 核矩阵之间的对齐程度。Kornblith 等（2019）表明，对于深度学习应用，使用中心化层激活值的线性核效果良好。因此，我们将重点关注线性 CKA，即 RV系数（Robert 和 Escoufier，1976）。此外，研究表明，线性 CKA 得出的结果与其他表征相似性指标相近，同时计算速度更快（Bonheme 和 Grzes，2022）。

为简洁起见，在本文余下部分中，我们将线性 CKA简称为 CKA。现在，我们对 CKA 作更正式的定义。给定针对 n 个数据样本取得的中心化层激活值 $x \in \mathbb{R}^{n\times m}$ 和 $y \in \mathbb{R}^{n\times p}$，CKA 定义如下：

$$CKA(x, y) = \frac{\|y^T x\|_F^2}{\|x^T x\|_F \|y^T y\|_F},$$

其中，$\|\cdot\|_F$ 表示弗罗贝尼乌斯范数，其定义如下：

$$\|a\|_F = \sqrt{\mathrm{tr}(aa^T)} = \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{l}|a_{ij}|^2},$$

其中，$a \in \mathbb{R}^{k\times l}$ 是任意矩阵，a^T 是 a的转置。

CKA 的局限性虽然 CKA 在实践中能得出准确结果，但它可能对神经网络架构的差异过于敏感（Maheswaranathan 等人，2019），因此可能会低估来自不同类型层（例如卷积层和反卷积层）的激活值之间的相似性。因此，在分析此类情况时，我们只讨论相似性的变化。例如，如果 a 和 b 是卷积层，而 c 是线性层，我们就不会比较 $CKA(a,\ b)$ 和 $CKA(a,\ c)$。不过，我们会比较 $CKA(a,\ c)$ 和 $CKA(b,\ c)$。

3. 逐步构建深度主动推理智能体

我们介绍的所有深度主动推理模型都做出了重要贡献，展示了多种可能的实现方式。不过，我们认为这些方法都不是深度主动推理完整而定论性的实现。在介绍过程中，我们始终指出这些已发表方法的局限性。因此，在本文余下部分，我们将回归基本原理，逐个组件地“构建”一个智能体，以确定“自然的”深度主动推理框架中的哪些部分赋予其解决推理问题的能力，哪些部分可能导致其无法解决问题。此外，在这一逐个组件的考察过程中，我们会将由此形成的不同深度主动推理变体与一种标准（已有充分验证的方法）进行比较：深度Q网络（Mnih 等，2013）。因此，本节将逐步构建一个深度主动推理智能体。第3.1节介绍用于运行所有模拟的 dSprites环境。我们选择该环境，是为了检验主动推理智能体能否解决 dSprites 问题，正如 Fountas 等人（ 2020 ）所探讨的那样。第 3.2 节介绍本文后续所述智能体与环境交互的方式。接着，第3.3节介绍一个变分自编码器（VAE）智能体，第3.4节讨论一个深度隐马尔可夫模型（HMM）智能体，第3.5节介绍一个深度批判性隐马尔可夫模型（CHMM）智能体，最后，第3.6节介绍一个完整的深度主动推理智能体。请注意，本节使用的符号汇总于附录 A。

3.1 dSprites环境

dSprites环境基于 dSprites 数据集（Matthey 等人，2017），该数据集最初旨在分析变分自编码器所学习的潜在表征（Doersch，2016）。dSprites 数据集由正方形、椭圆和心形的图像组成。每张图像包含一个形状（正方形、椭圆或心形），并具有自身的尺度、方向以及（X，Y）位置。在 dSprites环境中，智能体可以通过执行四种动作（即向上、向下、向左、向右）来移动这些形状。为了使任务易于处理，在下一个动作-感知循环开始之前，智能体选择的动作会在环境中执行八次，即在时间步 t 和 $t+1$ 之间，X 或 Y 位置会增加或减少八个。智能体的目标是将所有正方形移向图像的左下角，并将所有椭圆和心形移向图像的右下角，参见图8。

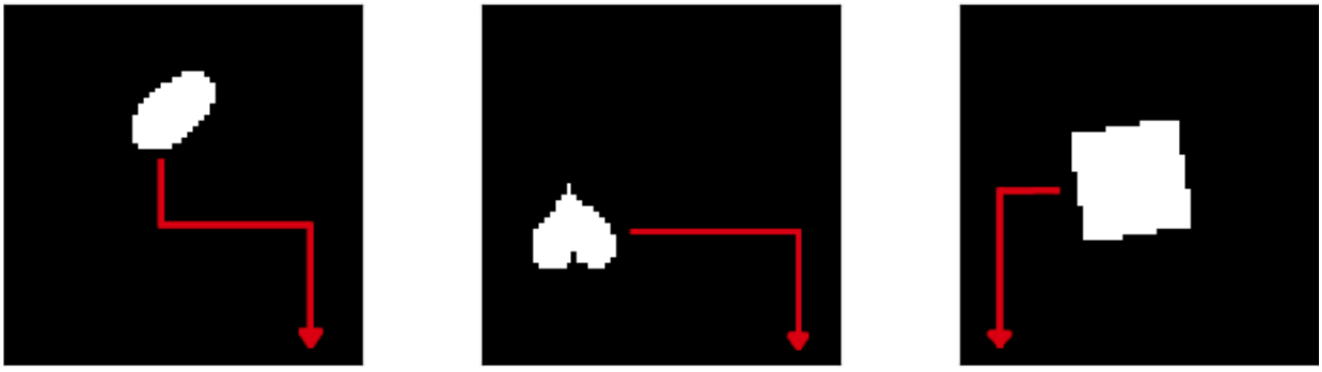


图8：此图展示了 dSprites 环境，智能体必须将所有正方形移向图像的左下角，并将所有椭圆和心形移向图像的右下角。红色箭头表示对智能体行为的预期。

3.2 智能体-环境交互

本节将介绍后续各节中所述的所有智能体如何与环境交互。每个智能体都经过了 $N = 500K$ 次迭代训练。每次试验开始时，环境都会重置为一个随机状态，智能体则接收一个观测[5] o_t。智能体利用 o_t选择一个动作 a_t，并在环境中执行该动作。随后，智能体会收到新的观测 o_{t+1}、一个奖励r_{t+1} 以及一个布尔值 $done$ ，用于说明试验是否结束。接着，新的经验（o_t， a_t，o_{t+1}， r_{t+1}， $done$）会被添加到经验回放缓冲区中，并从中抽取一个批次来训练智能体的各个神经网络。最后，如果试验已经结束，环境就会重置为一个随机状态，从而产生新的观测 o_t；否则， o_{t+1} 会成为新的 o_t ，结束动作-感知循环。算法1总结了智能体-环境交互过程。

算法1： 智能体与环境之间的交互。

输入： env 环境，
$agent$ 智能体，
$buffer$经验回放缓冲区，
N 训练迭代次数。

```
o_t =env.reset()                                  // 从环境获取初始观测
重复 N 次
    a_t ← 选择动作(o_t)                                          // 选择一个动作
    o_{t+1}, r_{t+1}, done ← 环境.execute(a_t)                   // 在环境中执行动作
    buffer.push new 经验(o_t, a_t, o_{t+1}, r_{t+1}, done)       // 将经验添加到
     经验回放缓冲区
    agent.learn(buffer)                                          // 执行一轮训练
    如果完成 == 为真 则
        o_t ← env.reset()                                        // 试验结束时重置环境
    else
        o_t ← o_{t+1}
    end
end
```

3.3 变分自编码器

在本节中，我们介绍第一个基于变分自编码器的智能体。该智能体由两个深度神经网络组成，即编码器和解码器。编码器 $\mathcal{E}_{\phi_s}$ 将图像 o_t作为输入，并输出变分后验 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t;\ \mu,\ \sigma)$ 的参数，其中， μ 是高斯分布的均值向量， σ 是协方差矩阵的对角元素。解码器 $\mathcal{D}_{\theta_o}$对似然映射 $P_{\theta_o}(o_t|s_t)$ 进行建模，该映射根据状态s_t为每个图像 o_t 赋予一个概率，其定义如下：

$$P_{\theta_o}(o_t|s_t) = \mathcal{B}\text{ernoulli}(o_t; \hat{o}_t),$$

其中，$\hat{}o_t = \mathcal{D}_{\theta_o}(s_t)$ 是解码器预测的值， $\mathcal{B}$ernoulli(o_t; $\hat{}o_t$) 是由伯努利分布构成的乘积，定义如下：

$$\mathcal{B}\text{ernoulli}(o_t; \hat{o}_t) = \prod_{x,y} \text{Bernoulli}(o_t[x,y]; \hat{o}_t[x,y]),$$

其中，伯努利分布(•；•) 是像素 $o_t[x,y]$ 所有可能取值上的伯努利分布，其参数 $\hat{}o_t[x,y]$ 由解码器网络预测。智能体的目标是最小化

5. 每个观测包含由三张图像构成的序列，即对应于环境当前状态的图像，以及在前两个时间步收集的两张图像。

变分自由能（VFE）：

$$\boldsymbol{F} = D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_o}(o_t, s_t)\right] = D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_o}(o_t|s_t)P(s_t)\right],$$

其中，$P(s_t) = \mathcal{N}(s_t; 0, I)$ 是方差为 1 的各向同性（多元）高斯分布。变分自由能可重新整理如下：

$$\boldsymbol{F} = D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P(s_t)\right] - \mathbb{E}_{Q_{\phi_s}(s_t)}[\ln P_{\theta_o}(o_t|s_t)],$$

其中，两个高斯分布之间的 KL 散度可以通过解析解计算，而 $P_{\theta_o}(o_t|s_t)$的对数的期望则通过使用单个样本 $\hat{}s_t \sim Q_{\phi_s}(s_t)$进行蒙特卡洛估计。样本 $\hat{}s_t$ 通过重参数化技巧获得，具体如下：$\hat{}s_t = \mu + \sigma \odot \hat{}\epsilon$，其中 $\odot$是两个向量的逐元素乘积，且 $\hat{}\epsilon \sim \mathcal{N}(\epsilon; 0, I)$。

总而言之，该智能体采取随机动作，并将其经验存储在经验回放缓冲区中（参见第3.2节）。随后，从经验回放缓冲区中采样批次的经验（o_t、 a_t、 o_{t+1}、 r_{t+1}、 $done$）。接着，将时间步 t 的观测输入编码器，编码器输出高斯分布的均值和对数方差 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t; \mu, \sigma)$。使用重参数化技巧从 $Q_{\phi_s}(s_t)$中采样一个潜在状态，然后将其作为输入提供给解码器，解码器输出伯努利分布的参数 $\hat{}o_t$。通过解析方法计算 $Q_{\phi_s}(s_t)$与 $P(s_t)$之间的 KL 散度；由于 $P_{\theta_o}(o_t|s_t)$是伯努利分布的乘积， $P_{\theta_o}(o_t|s_t)$的对数可简化为二元交叉熵（BCE）。接下来，通过从 KL 散度中减去二元交叉熵得到变分自由能，并使用反向传播更新编码器和解码器网络的权重。图 9 展示了本节介绍的 VAE智能体。请注意，该智能体采取随机动作。

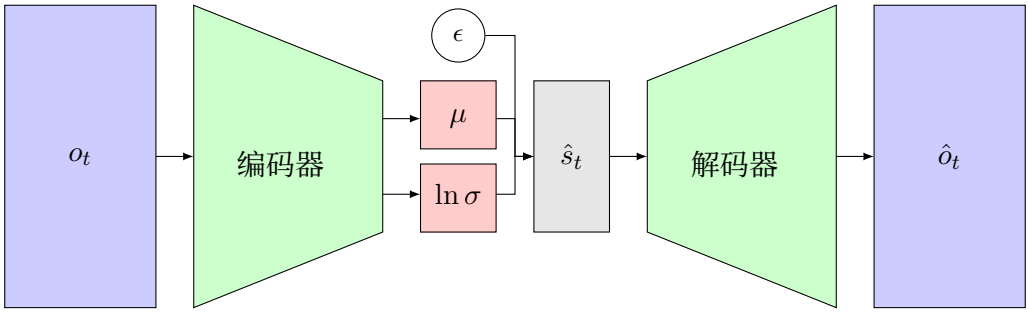


图 9：此图展示了 VAE智能体。从左到右依次为输入图像 o_t、编码器网络、均值 μ 和对数方差 $\ln\sigma$所在的层、用于重参数化技巧的 epsilon随机变量、潜在状态 $\hat{}s_t$、解码器网络，最后是重建图像 $\hat{}o_t$。请注意，该智能体的生成模型中没有动作。因此，VAE智能体会采取随机动作。

3.4 深度隐马尔可夫模型

在本节中，我们介绍第二个基于隐马尔可夫模型的智能体。与 VAE 智能体类似，HMM 智能体由一个对 $Q_{\phi_s}(s_\tau)$ 建模的编码器网络，以及一个对 $P_{\theta_o}(o_\tau|s_\tau)$ 建模的解码器网络组成。不过，时间步 $t+1$ 的隐状态先验取决于时间步 t 的隐状态和动作。该先验由转移网络 $\mathcal{T}_{\theta_s}$ 建模，该网络用于预测高斯分布 $P_{\theta_s}(s_{t+1}|s_t, a_t) = \mathcal{N}(s_{t+1}; \mathring{\mu}, \mathring{\sigma})$ 的参数，其中 $\mathring{\mu}$ 是高斯分布的均值，而 $\mathring{\sigma}$ 是协方差矩阵的对角元素。回想一下，推断过程的目标是使近似后验 $Q_{\phi_s}(s_t)$ 拟合真实后验 $P(s_t|o_t, s_{t-1}, a_{t-1})$。形式上，这一优化可以写成最小化以下两者之间的 Kullback-Leibler散度：

近似后验与真实后验，即：

$$Q^*(s_t) = \underset{Q_{\phi_s}(s_t)}{\arg\min}\, D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P(s_t|o_t, s_{t-1}, a_{t-1})\right].$$

使用与第 2.2.2 节中所述几乎相同的推导，可以证明变分自由能为：

$$\begin{aligned} Q^*_{\phi_s}(s_t) &= \underset{Q_{\phi_s}(s_t)}{\arg\min} \underbrace{D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_o}(o_t|s_t) P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right]}_{\text{variational free energy}} \\ &= \underset{Q_{\phi_s}(s_t)}{\arg\min} D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right] - \mathbb{E}_{Q_{\phi_s}(s_t)}[P_{\theta_o}(o_t|s_t)]. \end{aligned} \tag{12}$$

公式 12 中的变分自由能可以用与 VAE 智能体类似的方式计算。简而言之，该智能体执行随机动作，并将其经验存入经验回放缓冲区（参见第3.2节）。然后，从经验回放缓冲区中采样经验批次 (o_{t-1}, a_{t-1}, o_t, r_t, $done$)。时间步$t-1$ 的观测被输入编码器，编码器输出高斯分布$Q_{\phi_s}(s_{t-1}) = \mathcal{N}(s_{t-1};\, \mu, \sigma)$的均值和对数方差。使用重参数化技巧从 $Q_{\phi_s}(s_{t-1})$中采样一个潜在状态，然后将其与动作 a_{t-1} 一同作为输入提供给转移网络。转移网络输出高斯分布 $P_{\theta_s}(s_t|s_{t-1}, a_{t-1}) = \mathcal{N}(s_t;\, \mathring{\mu}, \mathring{\sigma})$的参数。此外，时间步 t 的观测也可以输入编码器，以获得 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t;\, \hat{}\mu, \hat{}\sigma)$的参数。使用重参数化技巧从 $Q_{\phi_s}(s_t)$中采样，得到状态 $\hat{}s_t$ ；将其作为输入提供给解码器后，解码器会输出伯努利分布乘积 $\hat{}o_{t+1}$ 的参数。通过解析计算 $Q_{\phi_s}(s_t)$ 与 $P_{\theta_s}(s_t|s_{t-1}, a_{t-1})$之间的 KL 散度；由于 $P_{\theta_o}(o_t|s_t)$是伯努利分布的乘积，$P_{\theta_o}(o_t|s_t)$的对数可化简为二元交叉熵（BCE）。接下来，从 KL 散度中减去 BCE，得到变分自由能，并使用反向传播更新编码器、解码器和转移网络的权重。图 10 展示了 HMM 智能体。

3.5 深度批判性隐马尔可夫模型

本节将介绍我们的第三个智能体，它在上一节介绍的深度隐马尔可夫模型中加入了一个评论家网络。由此得到的模型称为深度 CHMM，如图 11 所示。CHMM 配备了一个编码器 $\mathcal{E}_{\phi_s}$，用于建模 $Q_{\phi_s}(s_\tau)$；一个解码器 $\mathcal{D}_{\theta_o}$，用于建模 $P_{\theta_o}(o_\tau|s_\tau)$；一个转移网络 $\mathcal{T}_{\theta_s}$，用于建模 $P_{\theta_s}(s_t|s_{t-1},\ a_{t-1})$；以及一个评论家网络 $\mathcal{G}_{\theta_a}$，用于预测每个动作的期望自由能（见下文）。随后，评论家网络用于定义动作先验：$P_{\theta_a}(a_t|s_t) = \sigma[-\zeta \mathcal{G}_{\theta_a}(s_t,\ \bullet)]$, 其中，$\zeta$是动作先验的精度，而 $\mathcal{G}_{\theta_a}(s_t,\ \bullet)$ 是评论家网络预测的在状态 s_t 下采取各个动作的期望自由能。编码器、解码器和转移网络的训练方式与之前相同，均以最小化变分自由能为目标。然而，评论家网络的训练目标是最小化其输出 $\mathcal{G}_{\theta_a}(s_t,\ \bullet)$ 与目标 G 值 $y(\bullet)$ 之间的平滑 L1 范数，即评论家网络最小化 $\mathrm{SL1}[\mathcal{G}_{\theta_a}(s_t,\ \bullet)\ y(\bullet)]$。请注意，选择 SL1 是因为它比均方误差更不易受离群值影响，其定义如下：

$$SL1[x, y] = \begin{cases} 0.5 \times \frac{(x-y)^2}{\beta} & \text{if } |x-y| < \beta \\ |x-y| - 0.5 \times \beta & \text{otherwise} \end{cases},$$

其中，β 是一个超参数，当 β 趋近于零时，SL1 损失会收敛到 L1 损失；当 β 趋近于 $+\infty$时，SL1 损失会收敛到恒为零的损失。直观来说，当逐元素绝对误差小于 β时，SL1 损失使用平方项，否则使用 L1 项。此外，目标 G 值定义如下：

$$y(a_t) = G_{t+1}(a_t) + \gamma \mathbb{E}_{Q_{\phi_s}(s_{t+1})}\Big[\max_{a_{t+1} \in \mathcal{A}} \hat{\mathcal{G}}_{\hat{\theta}_a}(s_{t+1}, a_{t+1})\Big],$$

其中，$Q_{\phi_s}(s_{t+1})$ 可通过将从经验回放缓冲区采样的图像 o_{t+1} 作为输入馈入编码器来计算；$G_{t+1}(a_t)$ 是采取动作 a_t 后在时间步 $t+1$ 的期望自由能（见下文），而 γ是

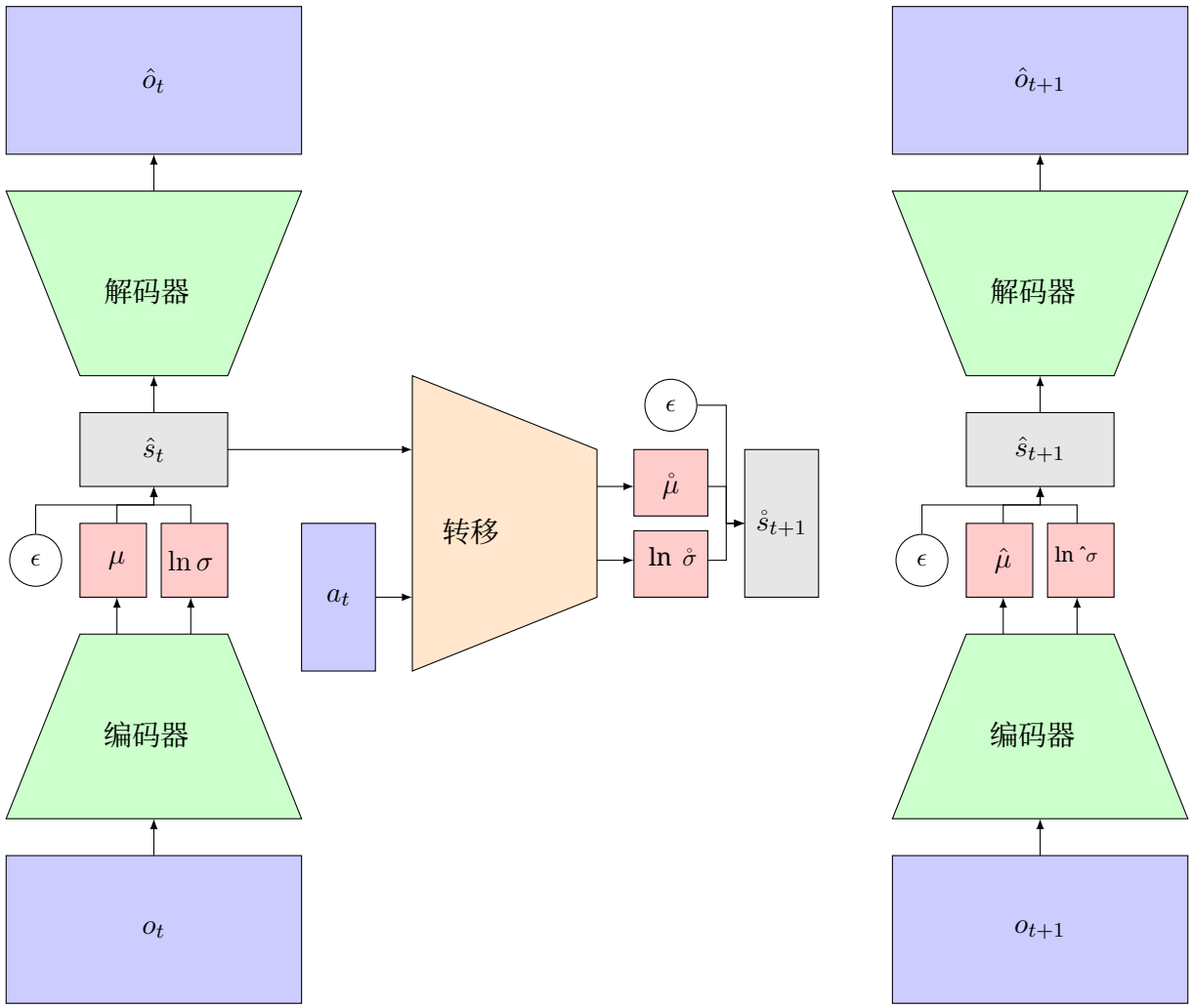


图 10：该图展示了隐马尔可夫模型智能体。左右两侧分别是两个自编码器，也就是说，一个位于时间步 t，另一个位于时间步 $t+1$。中间的转移网络将时间 t 的状态和动作作为输入，即 $(\hat{s}_t, a_t)$，并输出高斯分布的均值 $\mathring{\mu}$ 和对数方差 $\ln \mathring{\sigma}$。通过对潜变量 ϵ 进行采样并使用重参数化技巧，我们得到转移网络输出的潜在状态：$\mathring{s}_{t+1} = \mathring{\mu} + \mathring{\sigma} \odot \hat{\epsilon}$，其中 $\hat{\epsilon}$ 是从均值为零、方差为一的高斯分布中采样得到的。重要的是，模型似乎由两个互不相连的部分组成，但变分自由能会包含一个复杂度项，该项涉及转移网络和时间 $t+1$ 的编码器输出的高斯分布。请注意，该智能体采取随机动作。

折扣因子。请注意，上述方程是将贝尔曼方程（Bellman，1952）应用于期望自由能。此外，与 DQN 智能体一样，我们通过实现$\hat{}$一个目标网络 $\mathcal{G}_{\hat{\theta}_a}$ 来提高训练稳定性；该网络在结构上与评论家网络完全相同，其权重每经过 K（学习）次迭代便与评论家网络的权重同步。在探讨期望自由能这一主题之前，最后还需回答一个问题：CHMM 如何选择要在环境中执行的动作？

至少有四种可能性：（i）选择一个随机动作；（ii）根据评论家网络选择使期望自由能最大的动作，即 $a_t^* = \arg\max_{a_t} \mathcal{G}_{\theta_a}(s_t, a_t)$；（iii）从评论家网络输出的 softmax 函数中采样一个动作，即 $a_t^* \sim \sigma[\mathcal{G}_{\theta_a}(s_t, \cdot)]$，其中 $\sigma[\cdot]$ 是一个 softmax 函数；（iv）使用 $\mathring{\epsilon}$-贪心

指数衰减算法，即以概率 $\mathring{\epsilon}$ 选择一个随机动作，或以概率 $1-\mathring{\epsilon}$ 选择最佳动作，其中 $\mathring{\epsilon}$ 起始值较高，并以指数速度快速衰减。

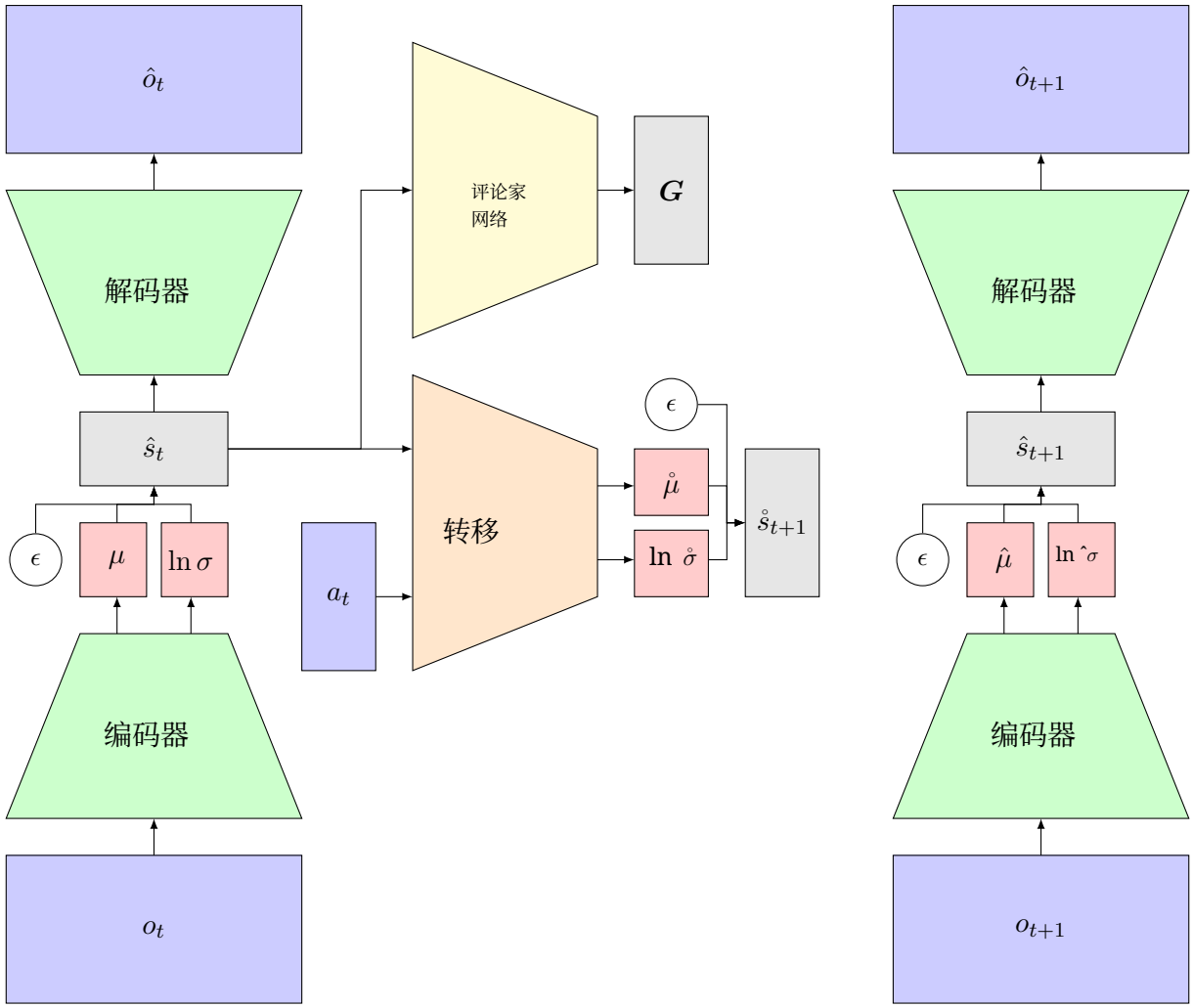


图 11：该图展示了 CHMM 智能体。唯一新增的部分是评论家网络，它以时间 t 的隐藏状态作为输入，并输出每个动作 G 的期望自由能。重要的是，CHMM 会根据期望自由能采取动作。

3.5.1 Expected free energy

本节将讨论期望自由能（EFE）的定义，然后探讨在深度主动推理中实现它的各种方法。最近，(Parr 和 Friston, 2019) 将期望自由能定义为：

$$G(\pi)=\sum_{\tau=t+1}^{T}G_\tau(\pi)=\sum_{\tau=t+1}^{T}\mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi)-\ln P(o_\tau,s_\tau)\big], \tag{13}$$

其中，$P(o_\tau|s_\tau)$ 是似然映射，$Q(s_\tau|\pi)$ 是变分分布；在文献中，$P(o_\tau,\ s_\tau)$ 被称为生成模型，但更恰当地说，它是对智能体先验偏好进行编码的目标分布。事实上，假设采用主动推断的标准生成模型（即

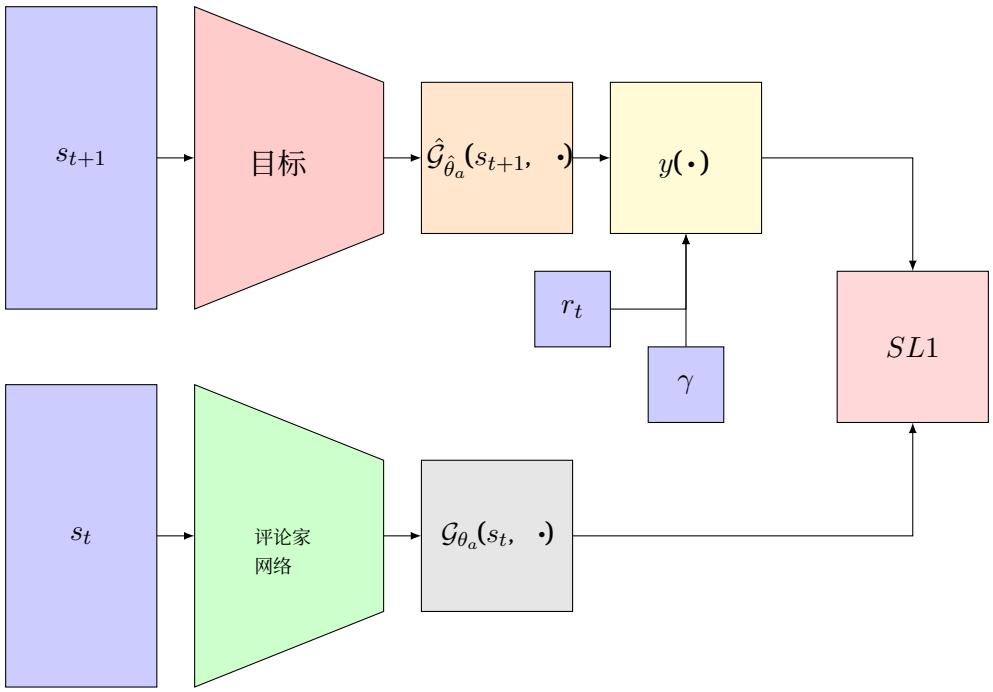


图12：该图展示了评论家网络仅最大化奖励时（即使用公式19计算期望自由能时）评论家的损失函数的计算过程。简而言之，状态 s_t 被输入评论家网络，状态 s_{t+1} 被输入目标网络。评论家网络输出时间 t 每个动作的 G 值，目标网络输出时间 $t+1$每个动作的 G 值。然后，利用奖励、折扣因子以及时间 $t+1$ 每个动作的 G 值计算目标值 $y(\cdot)$。最后，目标是通过调整评论家网络的权重，使评论家网络的预测与目标值之间的 SL1 最小化。

部分可观测马尔可夫决策过程），隐藏状态 s_τ 应取决于 $s_{\tau-1}$ 和 $a_{\tau-1}$。这一点很重要，因为它关系到 $P(o_\tau, s_\tau)$ 是否确实是生成模型，进而关系到期望自由能是否是变分自由能的期望。根据自由能原理，主动推理智能体必须最小化其（变分）自由能。然而，如果无法确立期望自由能与变分自由能之间的这种关系，就不能声称最小化期望自由能的智能体也会最小化其变分自由能。此外，我们需要重新整理式（13）中给出的期望自由能定义，以便纳入奖励：

$$
\begin{aligned}
G_\tau(\pi) &= \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi) - \ln P(o_\tau, s_\tau)\big] \\
&= \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi) - \ln P(s_\tau|o_\tau) - \ln P(o_\tau)\big] \\
&\approx \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi) - \ln Q(s_\tau) - \ln P(o_\tau)\big] \\
&= \underbrace{D_{\mathrm{KL}}\left[\, Q(s_\tau|\pi) || \, Q(s_\tau)\right]}_{\text{epistemic value}} - \underbrace{\mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln P(o_\tau)\big]}_{\text{extrinsic value}},
\end{aligned}
\tag{14}
$$

3.5.2 A principled estimate of the EFE at time $t+1$?

现在，问题是如何估计 (14)，我们将重点关注 $\tau = t+1$ 的情况。注意，由于$\tau = t+1$，策略 π 仅包含一个动作 a_t，即 $\pi = a_t$。在主动推断的表格版本中，变分分布由一个因子 $Q(s_\tau|\pi)$ 组成。然而，在深度主动推理文献中，

变分分布不包含这样的因子。通常使用如下蒙特卡洛估计：

$$Q(s_{t+1}|a_t) = \mathbb{E}_{Q_{\phi_s}(s_t)}\big[P_{\theta_s}(s_{t+1}|s_t, a_t)\big] \approx \frac{1}{N}\sum_{i=1}^{N} P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t^i, a_t), \tag{15}$$

其中 $\hat{s}_t^i \sim Q_{\phi_s}(s_t)$。重要的是，若要使期望自由能等于变分自由能的期望， $Q(s_{t+1}|a_t)$ 应当是变分分布的一个因子。然而，式（15）使用生成模型 $P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t^i,\ a_t)$ 的一个因子进行估计。这是一个概念性问题，与当前的深度主动推断方法有关，例如 Fountas 等人（2020）。接下来，我们使用 $N = 1$，得到这一估计的简化版本：

$$Q(s_{t+1}|a_t) \approx P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t). \tag{16}$$

请注意，在上面的方程中，$\hat{s}_t^i$ 记作 $\hat{s}_t$，因为只有一个样本，即 $N = 1$。至此，我们得到了 $Q(s_{t+1}|a_t)$ 的估计，而 $Q_{\phi_s}(s_t)$ 是变分分布。唯一缺少的部分是对外在价值的估计。在主动推断的表格版本中，智能体的偏好可以与强化学习文献中的奖励联系起来。本文遵循 Costa 等，2020b，并将先验偏好定义为：

$$P(o_\tau) = \frac{\exp(\psi r_\tau[o_\tau])}{\sum_{o_\tau}\exp(\psi r_\tau[o_\tau])},$$

其中， ψ是先验偏好的精度， $r_\tau[o_\tau]$是在进行观测o_τ时获得的奖励。对上式取对数可得：

$$\begin{aligned}\ln P(o_\tau) &= \psi r_\tau[o_\tau] - \ln\sum_{o_\tau}\exp(\psi r_\tau[o_\tau]) \\ &= \psi r_\tau[o_\tau] + C,\end{aligned} \tag{17}$$

其中，我们利用了对所有 o_τ 求和构成归一化项（即常数）这一事实。利用式（17），我们现在可以按如下方式估计外在价值：

$$\mathbb{E}_{P_{\theta_o}(o_\tau|s_\tau)Q(s_\tau|a_t)}\big[\ln P(o_\tau)\big] \approx \frac{1}{M}\sum_{i=1}^{M}\ln P(o_\tau = \hat{o}_\tau) = \frac{1}{M}\sum_{i=1}^{M}\psi r_\tau[o_\tau] + C,$$

其中，$\hat{o}_\tau \sim P_{\theta_o}(o_\tau|s_\tau = \hat{s}_\tau)$ 和 $\hat{s}_\tau \sim Q(s_\tau|a_t)$。下文中，我们使用 $M = 1$ 并舍去常数[6]，从而得到简化的估计式：

$$\mathbb{E}_{P_{\theta_o}(o_\tau|s_\tau)Q(s_\tau|a_t)}\big[\ln P(o_\tau)\big] \triangleq \psi r_\tau[o_\tau] \triangleq \psi r_\tau,$$

其中，我们简化了符号说明，将 $r_\tau[o_\tau]$ 记为 r_τ，本文的所有模拟均采用这一记法。总之，时间步 $\tau = t+1$的 EFE 估计如下：

$$\begin{aligned}G_{t+1}(a_t) &\approx D_{\mathrm{KL}}\left[\,Q(s_{t+1}|a_t)||\,Q_{\phi_s}(s_{t+1})\right] - \mathbb{E}_{P_{\theta_o}(o_{t+1}|s_{t+1})Q(s_{t+1}|a_t)}\big[\ln P(o_{t+1})\big] \\ &\approx D_{\mathrm{KL}}\left[\,P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t)||\,Q_{\phi_s}(s_{t+1})\right] - \psi r_{t+1},\end{aligned} \tag{18}$$

其中，$\hat{s}_t \sim Q_{\phi_s}(s_t)$、 $P_{\theta_s}(s_{t+1}|s_t,\ a_t)$ 已由生成模型给出， $Q_{\phi_s}(s_{t+1})$ 已由变分分布给出；KL 散度可通过解析解估计， ψ 是用于调节先验偏好精度的超参数，而 r_{t+1} 是在时间步$t+1$获得的奖励。如图12所示，时间步 $t+1$ 的奖励用于计算评论家网络必须预测的目标值。

6. 消去常数项 不会影响哪种策略最优。事实上， $\pi^* = \arg\max_\pi G(\pi) = \arg\max_\pi G(\pi) - C$.

3.5.3 Other definitions of the EFE at time $t+1$

在上一节中，我们介绍了一种可能有原则依据的期望自由能估计方法。正如本文后文将要讨论的，这种期望自由能估计在实证方面效果不佳。为了探索其他可能性，我们还尝试了以下定义，它们对认知价值项只做了相对微小的调整：

$$
\begin{aligned}
G^1_{t+1}(a_t) &= H[Q_{\phi_s}(s_{t+1})] - H[P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t)] - \psi r_{t+1}, \\
G^2_{t+1}(a_t) &= H[P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t)] - H[Q_{\phi_s}(s_{t+1})] - \psi r_{t+1}, \\
G^3_{t+1}(a_t) &= D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_{t+1}) || \, P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t)\right] - \psi r_{t+1},
\end{aligned}
$$

其中所有熵项和 KL 散度都是通过解析方法计算的。此外，我们还尝试直接预测（负的）期望未来奖励，如下所示：

$$
G^4_{t+1}(a_t) = -\psi r_{t+1}, \tag{19}
$$

这实际上使评论家网络的工作与深度Q网络智能体中的 Q网络完全相同（详情参见第 2.1 节）。更准确地说，两者完全相同，唯一的区别是：Q网络以观测作为输入，而评论家网络则以隐藏状态作为输入。

3.6 深度主动推理

本节将讨论图 13 所示的完整深度主动推理（DAI）智能体。简单来说，该智能体由五个深度神经网络组成，即编码器、解码器、转移网络、评论家网络和策略网络。解码器 $\mathcal{D}_{\theta_o}$ 将 $P_{\theta_o}(o_\tau|s_\tau)$ 建模为伯努利分布的乘积，即$P_{\theta_o}(o_\tau|s_\tau) = \mathcal{B}$伯努利分布$(o_\tau; \hat{}o_\tau)$，其中 $\hat{}o_\tau = \mathcal{D}_{\theta_o}(s_\tau)$。转移网络 $\mathcal{T}_{\theta_s}$将 $P_{\theta_s}(s_{\tau+1}|s_\tau,\ a_\tau)$ 建模为高斯分布，即 $P_{\theta_s}(s_{\tau+1}|s_\tau,\ a_\tau) = \mathcal{N}(s_{\tau+1}|\mathring{\mu},\ \mathring{\sigma})$，其中 $\mathring{\mu},\ \ln\mathring{\sigma} = \mathcal{T}_{\theta_s}(s_\tau,\ a_\tau)$。评论家网络 $\mathcal{G}_{\theta_a}$输出一个向量，其中包含每个动作的预测期望自由能，该向量用于将动作先验定义为 $P_{\theta_a}(a_\tau|s_\tau) = \sigma[-\zeta\mathcal{G}_{\theta_a}(s_\tau,\ \cdot)],$，其中 $\sigma[\cdot]$是 softmax 函数，ζ是动作先验的精度。基于此，智能体的完整生成模型如下：

$$
P_\theta(o_{o:T}, s_{o:T}, a_{o:T-1}) = P(s_0)\prod_{\tau=0}^{T} P_{\theta_o}(o_\tau|s_\tau)\prod_{\tau=0}^{T-1} P_{\theta_s}(s_{\tau+1}|s_\tau, a_\tau)P_{\theta_a}(a_\tau|s_\tau),
$$

其中 $P(s_0) = \mathcal{N}(s_0;\ \mu_0,\ \sigma_0)$ 是关于初始隐藏状态的高斯先验。令 t为当前时间步。DAI 智能体维护关于当前状态 s_t 和动作 a_t的后验信念。关于状态的变分后验 $Q_{\phi_s}(s_t)$ 是由编码器 $\mathcal{E}_{\phi_s}$建模的高斯分布，即 $Q_{\phi_s}(s_t) = \mathcal{N}(s_t;\ \mu,\ \sigma)$，其中 $\mu,\ \ln\sigma = \mathcal{E}_{\phi_s}(o_t)$。关于动作的变分后验 $Q_{\phi_a}(a_t|s_t)$ 是由策略网络 $\mathcal{P}_{\phi_a}$建模的分类分布，即 $Q_{\phi_a}(a_t|s_t) = \mathrm{Cat}(a_t; \hat{}\pi)$，其中 $\hat{}\pi = \mathcal{P}_{\phi_a}(s_t)$。因此，完整的变分分布定义为：

$$
Q_\phi(a_t, s_t) = Q_{\phi_a}(a_t|s_t)Q_{\phi_s}(s_t).
$$

DAI 智能体的变分自由能推导方式与第 2.2.2 节中的变分自由能类似，定义如下：

$$
\begin{aligned}
Q^*_\phi(s_t, a_t) &= \underset{Q_\phi(s_t,a_t)}{\arg\min}\ \underbrace{D_{\mathrm{KL}}\left[\, Q_\phi(s_t, a_t) || \, P_{\theta_o}(o_t|s_t)P_{\theta_s}(s_t|s_{t-1}, a_{t-1})P_{\theta_a}(a_t|s_t)\right]}_{\text{variational free energy}} \\
&= \underset{Q_\phi(s_t,a_t)}{\arg\min}\quad \mathbb{E}_{Q_{\phi_s}(s_t)}\left[D_{\mathrm{KL}}\left[\, Q_{\phi_a}(a_t|s_t) || \, P_{\theta_a}(a_t|s_t)\right]\right] + D_{\mathrm{KL}}\left[\, Q_{\phi_s}(s_t) || \, P_{\theta_s}(s_t|s_{t-1}, a_{t-1})\right] \\
&\qquad\qquad - \mathbb{E}_{Q_{\phi_s}(s_t)}[\ln P_{\theta_o}(o_t|s_t)].
\end{aligned} \tag{20}
$$

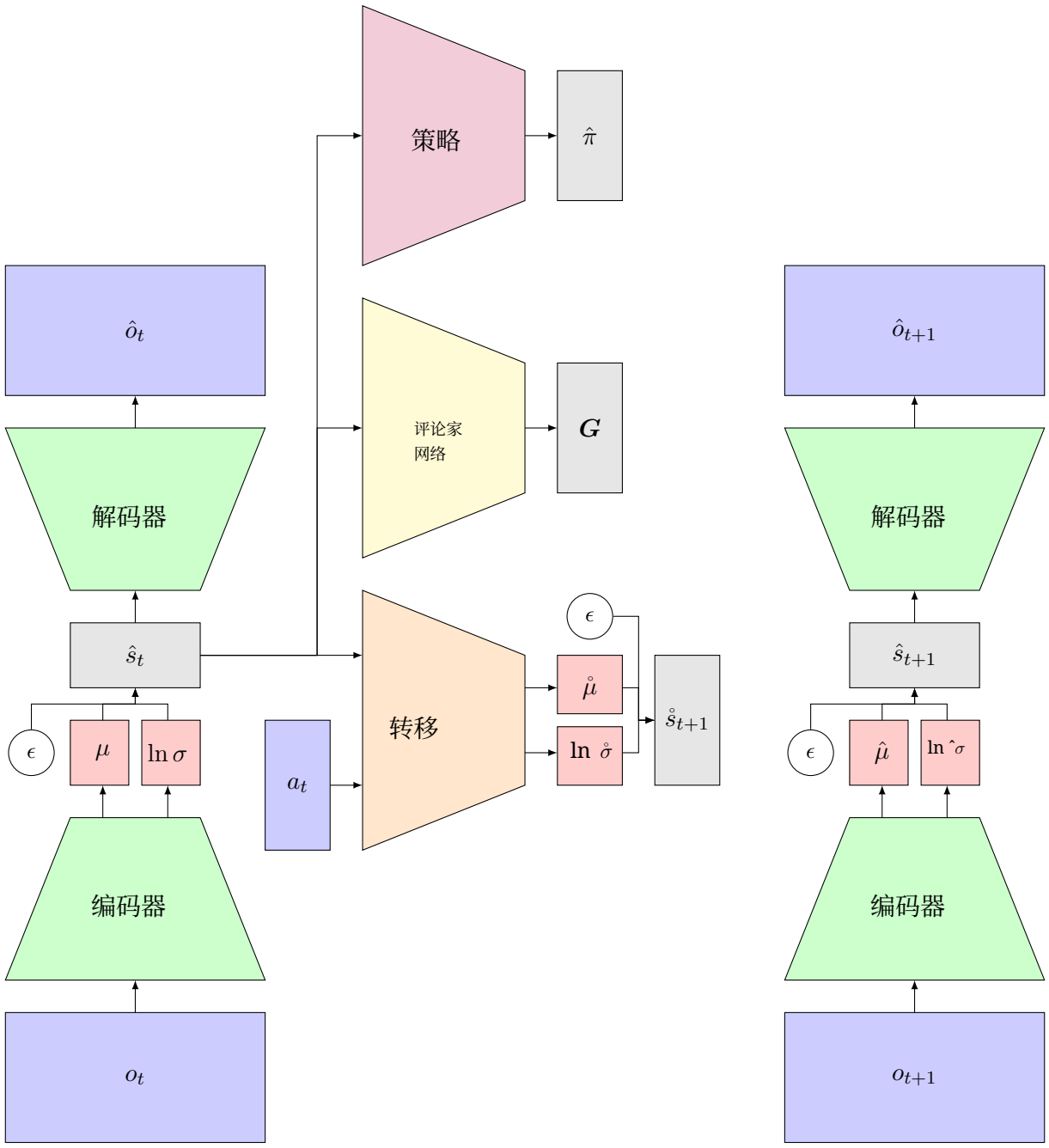


图 13：该图展示了 DAI 智能体。唯一新增的部分是策略网络，它以时间 t 的隐藏状态作为输入，并输出动作变分后验的参数 $\hat{}\pi$ 。重要的是，DAI 会基于期望自由能（EFE）采取动作。

因此，变分自由能是 s_{t-1}、a_{t-1} 和 o_t 的函数。a_{t-1} 和 o_t 均可从经验回放缓冲区中获取；当观测 o_{t-1} 作为输入时，可以从编码器网络预测的变分分布中采样得到 s_{t-1} 。此外，KL 散度可以通过解析方法计算；关于 $Q_{\phi_s}(s_t)$ 的期望可以用蒙特卡洛估计近似；由于 $P_{\theta_o}(o_t|s_t)$ 是伯努利分布的乘积，似然映射的对数可简化为二元交叉熵。因此，所有变分自由能项均可估计，而编码器、解码器、转移和策略

网络可以通过梯度下降进行训练，以最小化 VFE。评论家网络的权重按照第3.5节中的方法，通过梯度下降进行优化，以最小化评论家网络输出$\mathcal{G}_{\theta_a}(s_t, \cdot)$与目标 G 值 $y(\cdot)$之间的平滑 L1 范数，即 $\mathrm{SL1}[\mathcal{G}_{\theta_a}(s_t, \cdot)\ y(\cdot)]$, 其中，目标 G 值定义如下：

$$y(a_t) = G_{t+1}(a_t) + \gamma \mathbb{E}_{Q_{\phi_s}(s_{t+1})}\Big[\max_{a_{t+1}\in\mathcal{A}} \hat{\mathcal{G}}_{\hat{\theta}_a}(s_{t+1}, a_{t+1})\Big],$$

其中，$Q_{\phi_s}(s_{t+1})$可以通过将从经验回放缓冲区中采样的图像 o_{t+1} 作为输入馈送给编码器来计算，γ 是折扣因子，而 $G_{t+1}(a_t)$是在时间步 $t+1$ 采取动作 a_t后获得的期望自由能，即：

$$G_{t+1}(a_t) \approx D_{\mathrm{KL}}\left[\, P_{\theta_s}(s_{t+1}|s_t = \hat{s}_t, a_t) || \, Q_{\phi_s}(s_{t+1})\right] - \psi r_{t+1}, \tag{21}$$

其中，$\hat{}s_t \sim Q_{\phi_s}(s_t)$，$\psi$ 是用于调节先验偏好精度的超参数，而 r_{t+1} 是在时间步 $t+1$获得的奖励。请注意，我们还尝试了第3.5.3节中介绍的时间 $t+1$ 处 EFE 的其他定义。最后，就 DAI 智能体执行的动作选择而言，至少有四种可能性：(i) 选择一个随机动作；(ii) 根据策略网络选择后验概率最高的动作，即 $a_t^* = \arg\max_{a_t} Q_{\phi_a}(a_t|s_t)$；(iii) 从动作后验中采样一个动作，即 $a_t^* \sim Q_{\phi_a}(a_t|s_t)$；以及 (iv) 使用带指数衰减的 $\mathring{\epsilon}$-贪心算法，即以 $\mathring{\epsilon}$ 的概率选择随机动作，或以 $1-\mathring{\epsilon}$ 的概率选择最佳动作。

4. 结果

本节讨论 DQN 智能体以及第3节介绍的各个模型在解决 dSprites 问题时取得的结果。可用于复现所有实验的代码可在 GitHub 上通过以下网址获取：https://github.com/ChampiB/Challenges_Deep_Active_Inference。第4.1节介绍 DQN 智能体取得的结果。第4.2节介绍 VAE智能体得到的变分自由能，并展示 VAE 生成的重建图像。第4.3节展示 HMM 智能体的变分自由能以及生成的图像序列。第4.4节展示 CHMM 得到的变分自由能，以及该模型采用不同动作选择方案和不同期望自由能定义时获得的奖励。最后，第4.5节讨论 DAI 智能体得到的变分自由能，以及该模型获得的奖励。请注意，在以下各节中，每次使用 CKA 时，我们都会抽取 5K 个数据样本，并用它们计算所有 CKA分数。

4.1 DQN 智能体

在本节中，我们报告深度Q网络 智能体获得的结果。如图14所示，深度Q网络能够累计获得约50K的总奖励。该结果验证了我们实现的正确性，并为评估CHMM和DAI 智能体的性能提供了一个基线。为了更好地理解深度Q网络学习到的表征，我们计算了其各层激活值之间的CKA分数。从图15可以看出，尽管靠近输入的层之间仍保留一定的相似性，但最后两层学习到了高度特异的表征。

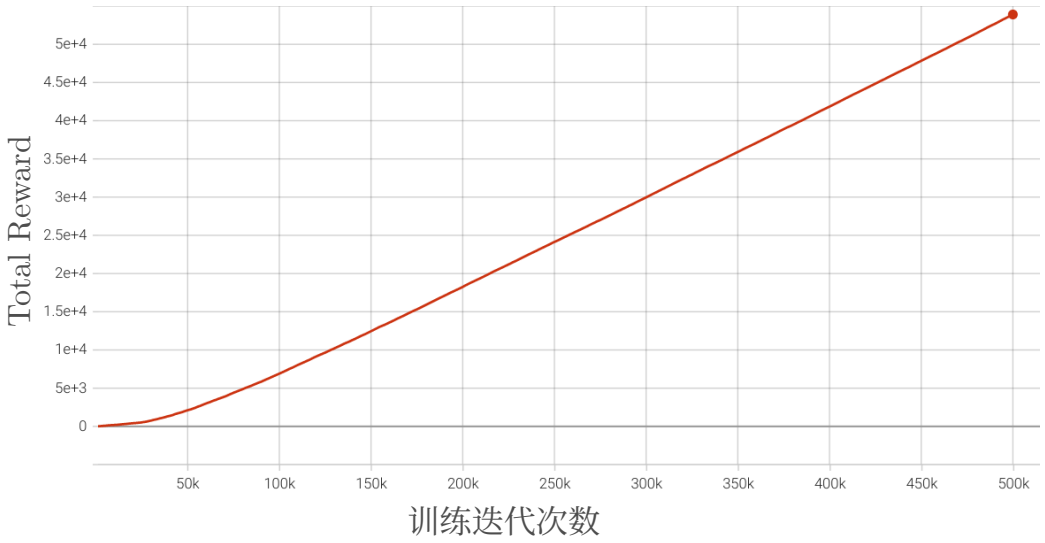


图14：该图展示了 DQN 智能体在 50万次训练迭代期间获得的累积奖励。

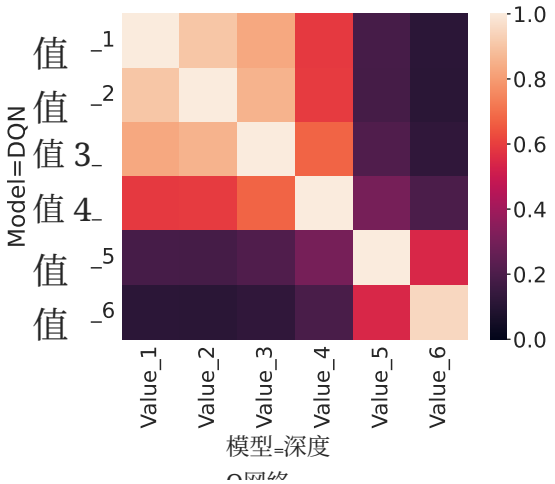


图15：值 X 表示价值网络的第 X 层，即值 1 最接近输入，值 6 是输出层。我们可以看到，DQN 的前三层学习到的表征非常相似（CKA 接近 1）。第四层学习到的表征开始出现分歧（CKA 较低），而最后两层学习到的表征具有很强的特异性，与前面的层大不相同（CKA 接近 0），但彼此之间略有相似。

4.2 VAE智能体

在本节中，我们报告VAE智能体获得的结果。如图16所示，随着训练的进行，变分自由能逐渐降低。此外，在完成50万次训练迭代后，变分自编码器能够正确重建图像，参见图18。除此之外，由于变分自编码器在环境中采取随机动作，该智能体无法完成任务，累计获得的总奖励约为-7K。这些结果表明我们的实现是正确的，并为dSprites环境中随机策略下获得的奖励数量提供了一个基线。

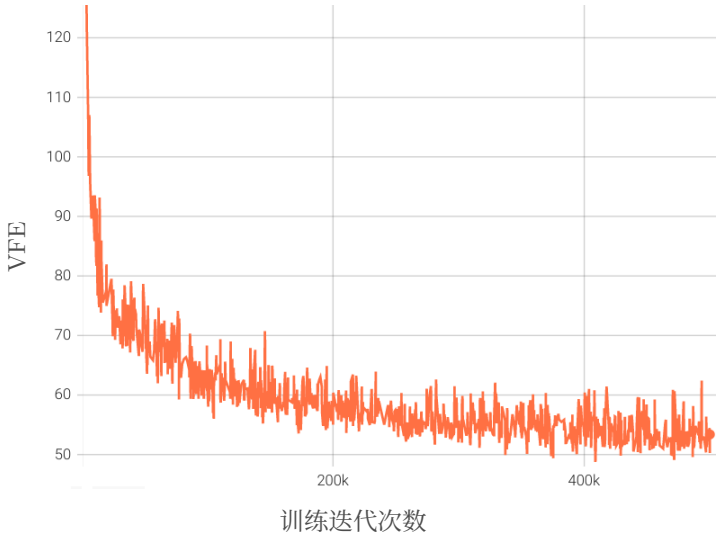
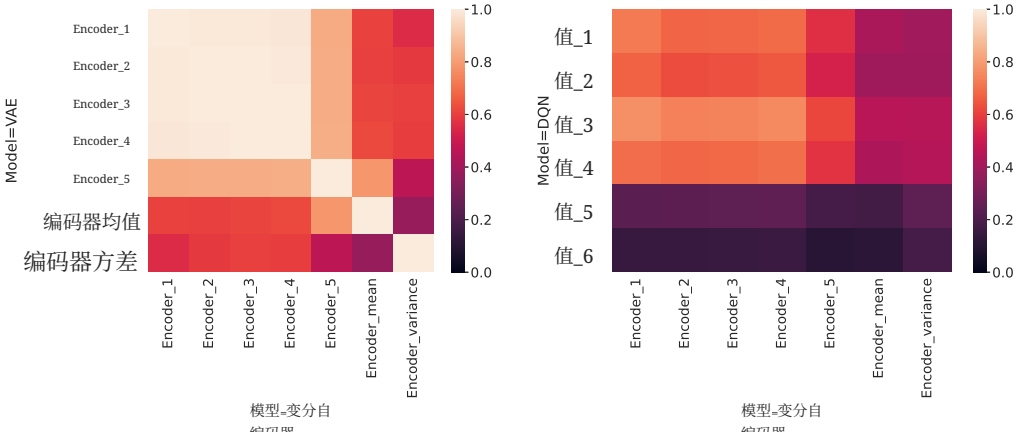


图16：此图展示了 VAE智能体在训练的50万次迭代过程中的变分自由能。



(a) (b)

图17：编码器X表示编码器网络的第X层，即编码器1最靠近输入，而编码器5位于均值和方差层之前。(a)展示了VAE编码器不同层所学习表征之间的相似性。(b)展示了深度Q网络和VAE所学习表征之间的相似性。请注意，VAE和深度Q网络都以图像作为输入，并需要对其进行处理，以分别学习紧凑表征并重建图像，或预测累积奖励。因此，两者都会在第一层学习表征边缘及其组合。

为了观察变分自编码器学习到的表征，我们计算了编码器不同层激活值之间的 CKA 分数。如图17a所示，除输出端的均值和方差表征（最后两层）外，所有层的表征都高度相似，这与 Bonheme和Grzes（2022年）的观察结果相同，因此也在预期之中。如图17b所示，这些表征总体上与深度Q网络在早期层学习到的表征相似，但最后两层的表征有明显差异，反映了变分自编码器与深度Q网络学习目标的不同。

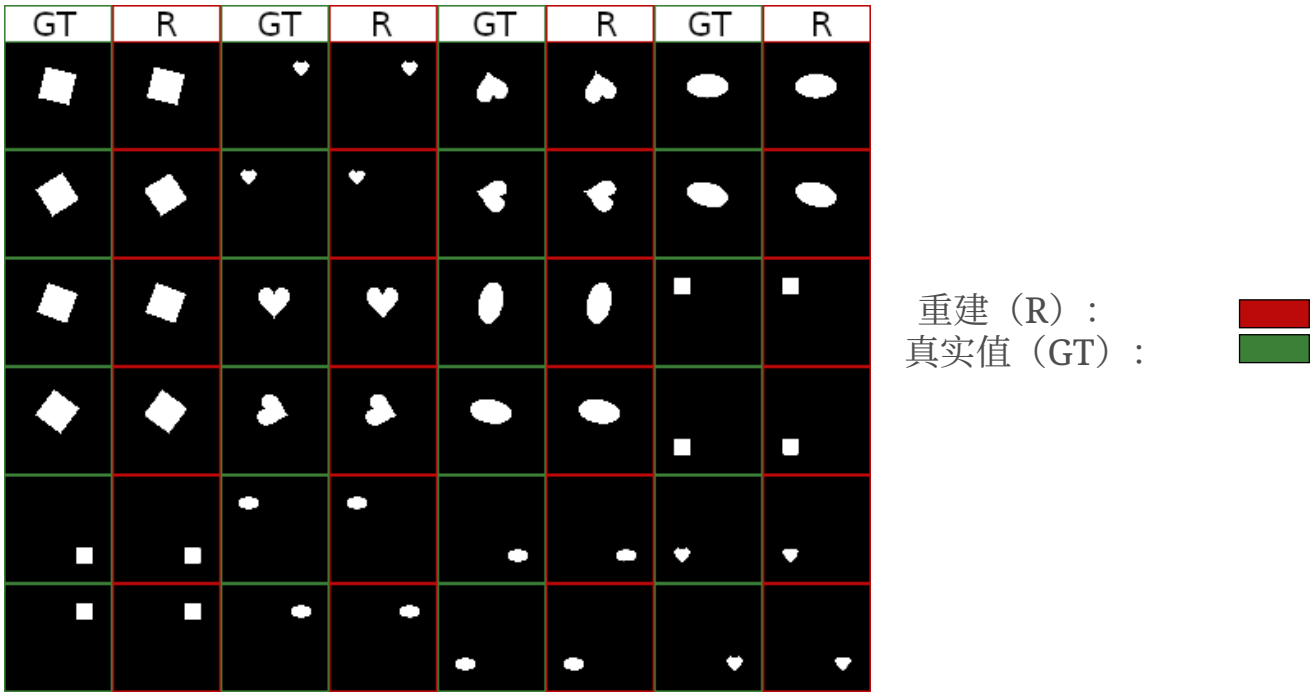


图18：该图展示了变分自编码器经过50万次训练迭代后生成的重建图像。各列交替显示输入图像和重建图像。

4.3 HMM 智能体

本节报告 HMM 智能体得到的结果。如图21所示，随着训练推进，变分自由能逐渐降低。比较图16和图21可以看出，隐马尔可夫模型的变分自由能低于变分自编码器。这是因为智能体对先验的设定更加灵活；也就是说，两个模型的对数似然相同，但隐马尔可夫模型的复杂度项小于变分自编码器。此外，在50万次训练迭代结束时，隐马尔可夫模型能够正确生成图像序列，参见图21。另外，由于 HMM 在环境中采取随机动作，智能体未能完成任务，累计奖励总额约为 -7K。这些结果表明我们的实现是正确的，并验证了在 dSprites环境中随机行动时获得的奖励基线。

与VAE类似，我们关注于观察HMM的编码器以及其转移网络所学习到的表征。HMM的编码器所学习到的表征与VAE所学习到的表征遵循相同的趋势，其中HMM的对数方差与该模型学习到的其他表征之间的差异更加明显，如图 20a和图 20b所示。我们还可以从图 20a中观察到，转移网络在其前三层中学习到了与均值表征（HMM编码器均值）相似的表征，而最后一层所学习到的表征（转移方差）与转移网络或编码器网络学习到的任何其他表征都不相似。我们还可以从图 20b中看到，HMM学习到的均值和方差表征（编码器均值和编码器方差）不同于VAE学习到的表征，这可能表明转移网络对这两个表征产生了影响。与VAE类似，从图 20c中可以观察到，编码器和转移网络的方差层所学习到的表征（编码器方差和转移方差）与DQN学习到的表征非常不同。相比之下，编码器的前四层（编码器1至编码器4）与DQN前几层学习到的表征（值 1至值 4）相似，但与最后两层（值 5和值 6）

与深度Q网络前几层（值 1 到值 4）学到的表征相似，但与最后两层（值 5 和值 6）非常不同。

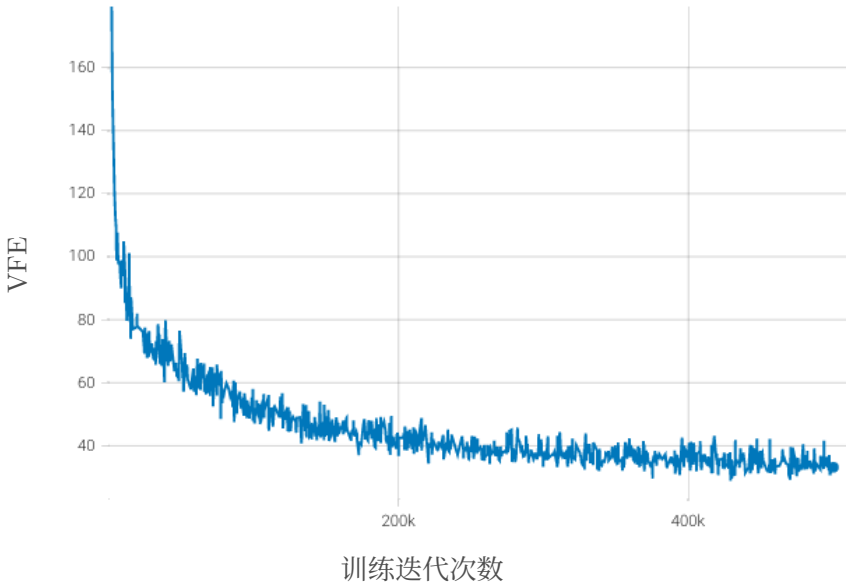
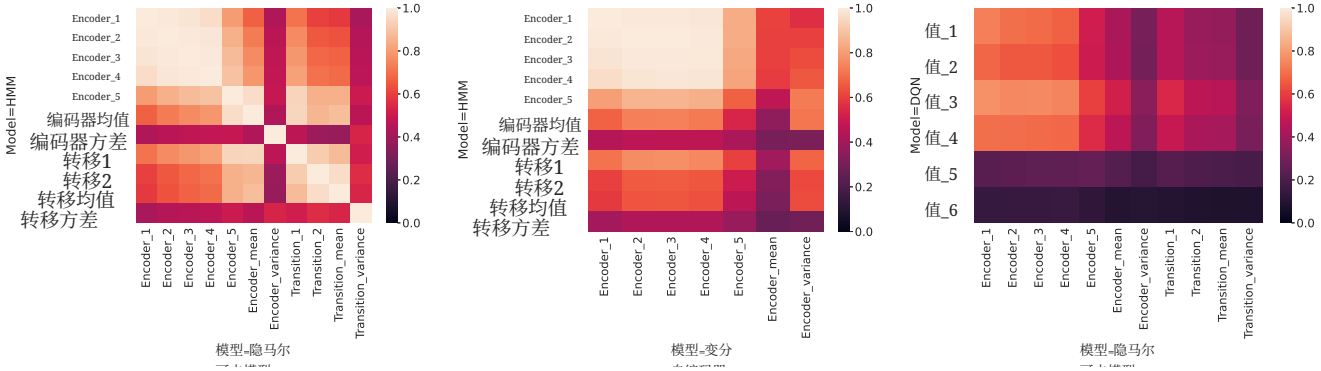


图20：该图展示了 HMM 智能体在训练的 50万次迭代期间的变分自由能。



(a) (b) (c)

图20：转移 X 表示转移网络的第 X 层，即转移 1 是最接近输入的层，而转移 2 是紧邻均值和方差层之前的层。转移均值和转移方差分别是转移网络的均值层和方差层。(a) 展示了 HMM 的编码器和转移网络中不同层所学习的表征之间的相似性。(b) 展示了 HMM 和变分自编码器所学习的表征之间的相似性。(c) 展示了深度Q网络和 HMM 所学习的表征之间的相似性。

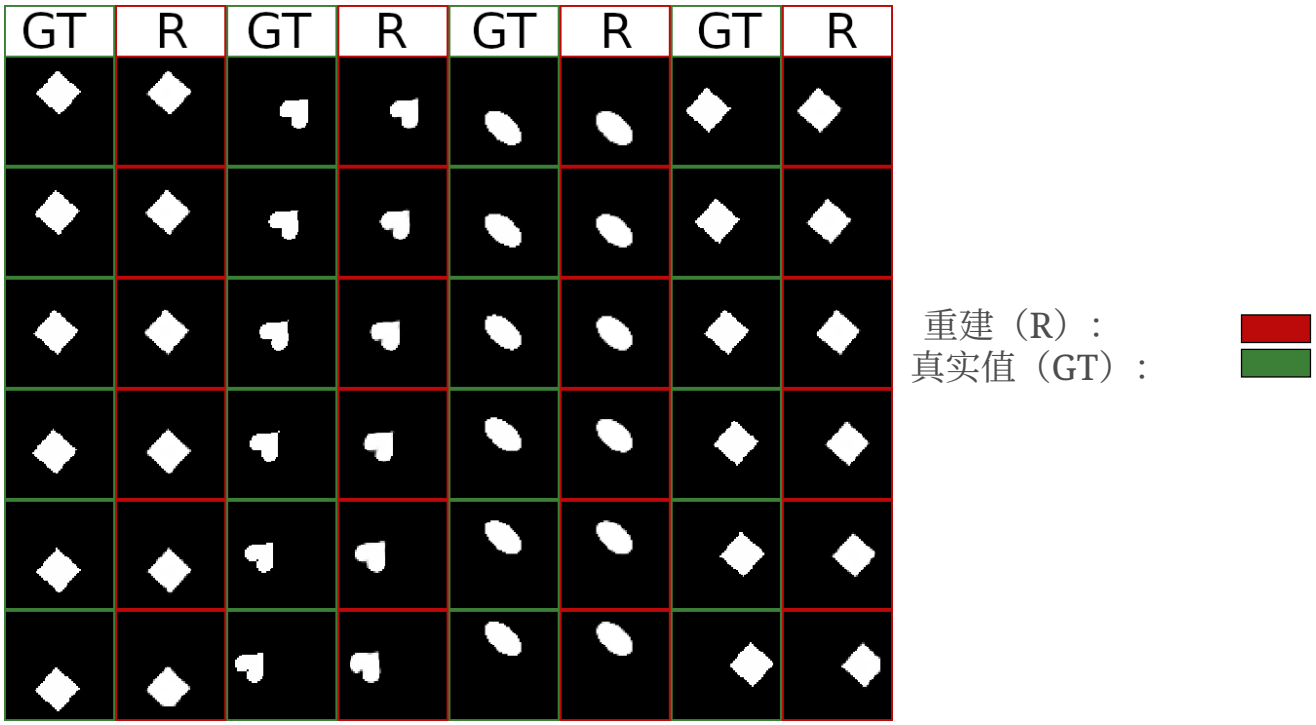


图21：该图展示了隐马尔可夫模型经过50万次训练迭代后生成的重建图像序列。各列交替呈现真实图像和重建图像。时间沿垂直方向流逝（从上到下），每列中的同一动作会被重复执行。

4.4 CHMM 智能体

本节报告 CHMM 智能体在采用不同动作选择策略以及不同期望自由能定义时得到的结果。图 22 展示了 CHMM 智能体使用 $\acute{\epsilon}$- 贪心算法进行动作选择时获得的累积奖励，以及深度 Q 网络智能体获得的总奖励。CHMM 智能体的评论家网络经过训练，能够预测第3.5节提出的五种期望自由能定义。请注意，深度Q网络智能体的表现优于所有 CHMM 智能体，而唯一能够完成任务的智能体是仅最大化奖励的 CHMM 智能体，也就是说，它不考虑任何信息增益。图 23 展示了与图 22 相同的实验，但 CHMM 智能体采用 softmax采样进行动作选择。在这种设置下，没有任何 CHMM 智能体能够完成任务。最后，图 24 再次展示了相同的实验，但这次 CHMM 智能体根据评论家网络选择最佳动作。在这种设置下，只有最大化奖励的 CHMM 智能体能够完成任务。此外，对比图 22 和图 24 可以清楚看出，使用 $\acute{\epsilon}$-贪心算法的 CHMM 智能体表现优于根据评论家网络选择最佳动作的 CHMM 智能体。简单来说，后者缺乏探索，导致学习速度减慢。

此外，图25展示了使用 $\acute{\epsilon}$-贪心算法的 CHMM 智能体的变分自由能。除橙色曲线所代表的智能体外，所有智能体都成功最小化了变分自由能；该智能体的变分自由能突然变为 NaN（即非数值）。该智能体最小化的是 G^1 所定义的期望自由能。请注意， G^1 既不是奖励，也不是“有原则的”期望自由能； G^1 是我们为探索其他定义而尝试的定义之一。此外，

未展示使用 softmax采样和最佳动作选择的 CHMM 智能体的变分自由能，因为其结果与图25所示结果非常相似。

最后，图26展示了 CHMM（以奖励最大化为目标）训练后得到的预测轨迹示例。通过与图21对比可以看出，CHMM 对环境动态的理解不如 HMM 智能体。这表明智能体的两个目标之间存在冲突，即最大化奖励[7]（或期望自由能）与学习世界模型。更准确地说，图21表明，采取随机动作的 HMM 智能体能够收集种类丰富的训练样本，并出色地学习环境动态，但无法完成任务。相比之下，以奖励最大化为目标的 CHMM 能够完成任务，却只学到了一个较差的环境模型，参见图26。

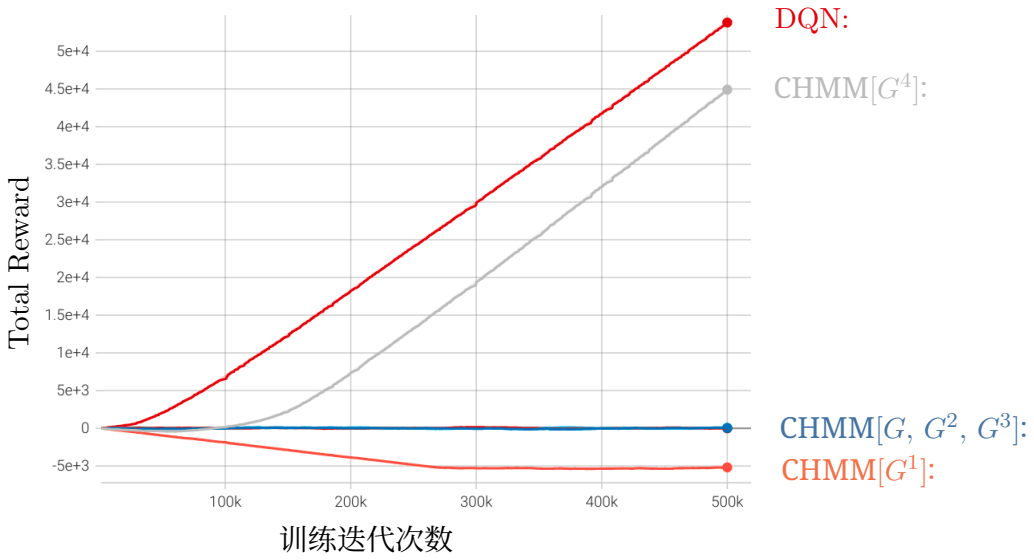


图 22：此图展示了 CHMM 智能体在 50万次训练迭代期间获得的奖励总量（采用 $\mathring{\epsilon}$-贪心动作选择）。只有两个模型成功完成了任务：以奖励最大化为目标（不考虑信息增益）的模型，即红色的 DQN 智能体，以及灰色的 CHMM，其评论家网络只预测奖励。

7. 如图12所示，奖励用于计算评论家网络必须预测的目标值。

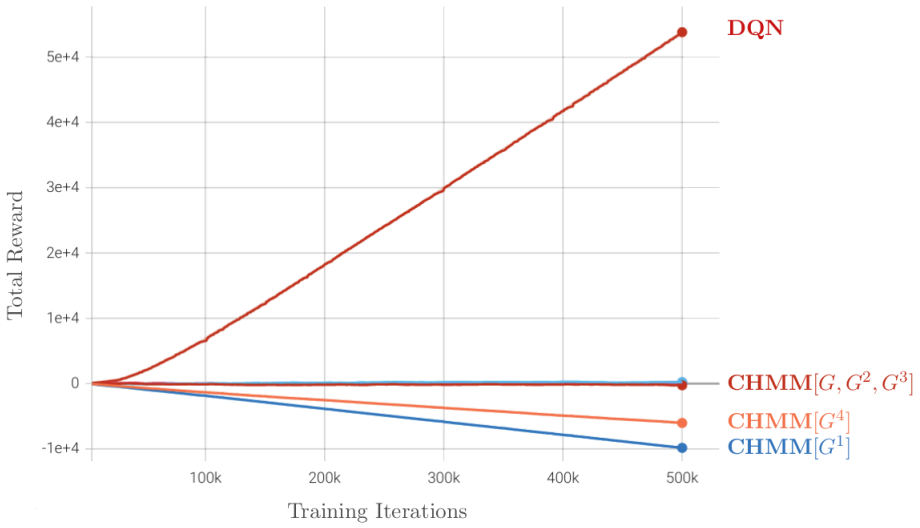


图23：该图展示了 CHMM 智能体（采用 softmax采样）在50万次训练迭代期间获得的奖励总量。唯一能够解决该任务的模型是红色的 DQN 智能体，所有 CHMM 智能体均告失败。

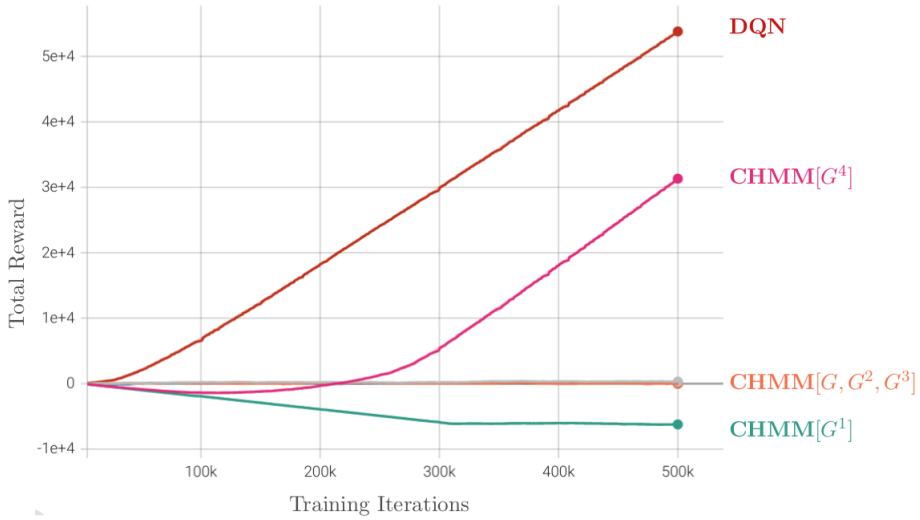


图 24：该图展示了 CHMM 智能体（采用最佳动作选择）在50万次训练迭代期间获得的奖励总量。唯一能够解决该任务的两个模型，是最大化奖励（不考虑信息增益）的模型，即红色的 DQN 智能体，以及粉色的 CHMM（其评论家网络仅预测奖励）。

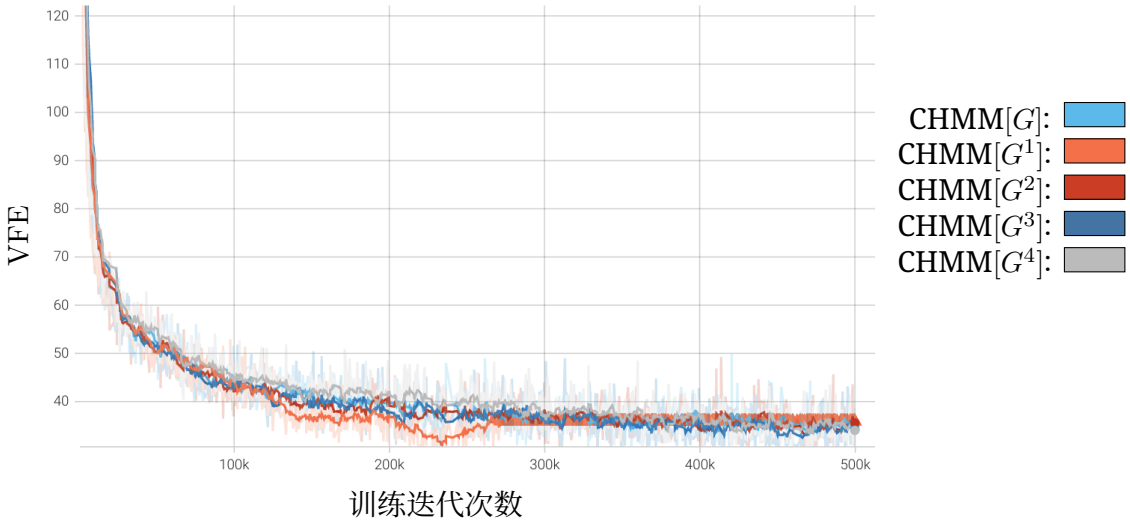


图 25：该图展示了 CHMM 智能体在 50万次训练迭代期间的变分自由能。除橙色所示的智能体发生崩溃外，所有智能体都能够最小化其变分自由能；该智能体按照 G^1 的定义最小化期望自由能。更准确地说，变分自由能取值为“非数值”（NaN），这一点可从 270K 至 500K 训练迭代之间的粗水平线看出。

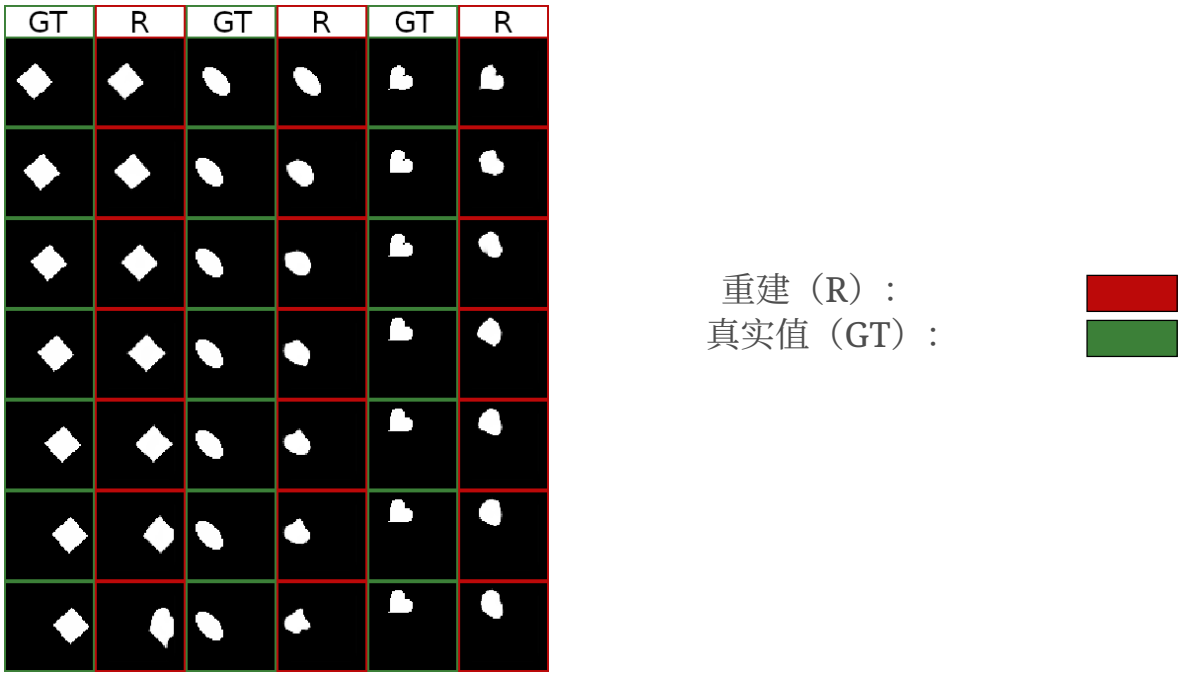
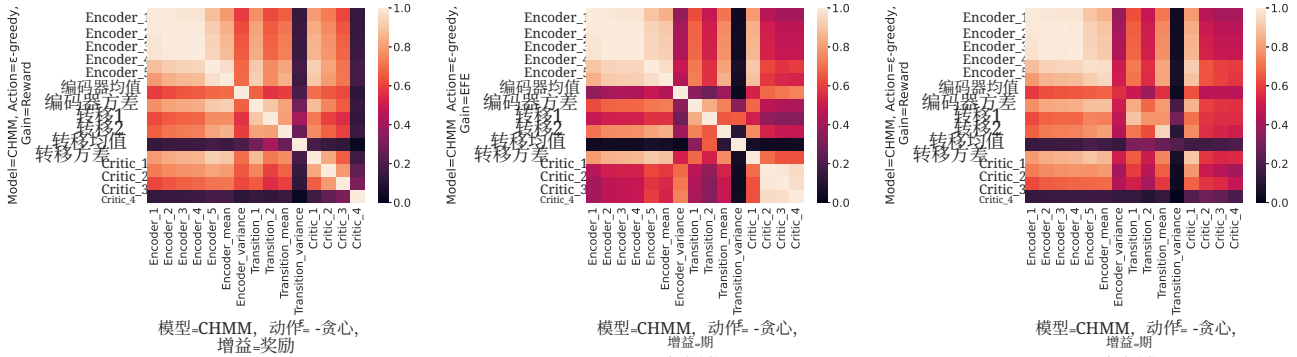


图 26：该图展示了 CHMM（最大化奖励）在 50万次训练迭代后生成的重建图像序列。各列交替显示真实图像和重建图像。时间沿垂直方向推移（从上到下），并且在每一列中都重复执行相同的动作。

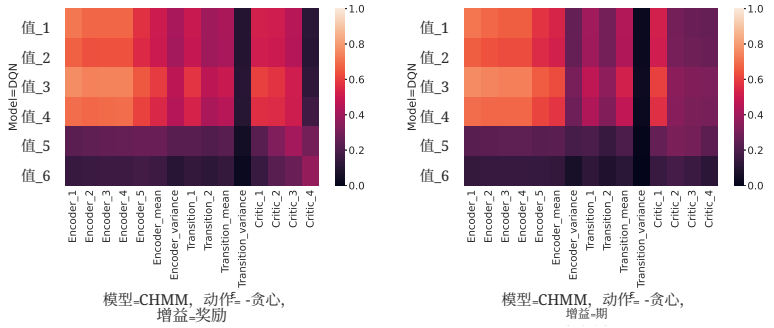
4.4.1 How do CHMMs learn?

采用 $\hat{\epsilon}$-贪心动作选择 如图 22 所示，只有评论家网络最大化奖励的 CHMM 才能完成该任务。因此，我们可以预期，该 CHMM 学到的表征会比其他 CHMM 学到的表征更接近深度Q网络学到的表征。我们可以从图 28 中看到，最大化奖励的 CHMM 的评论家网络最后两层，确实比在评论家网络最小化期望自由能时学到的表征更接近深度Q网络的最后两层表征（参见矩阵右下角，值 5 和值 6 与评论家 3 和评论家 4 的交集）。然而，两个 CHMM 的评论家网络学到的表征仍与深度Q网络的最后两层有很大差异（同样参见矩阵右下角，CKA 低于 0.4）。有趣的是，最大化奖励的 CHMM 的前四层与深度Q网络较早的层仍具有较高相似度（4×4 左上区域），这表明不同模型之间存在一些共同表征。我们还可以从图 27a 中看到，编码器、转移网络和评论家网络之间的 CKA 分数大于或等于 0.6（转移网络的方差层和评论家网络的最后一层除外），这表明最大化奖励的 CHMM 的转移网络和评论家网络保留了编码器的一些信息。如图 27b 所示，当 CHMM 最小化期望自由能时，评论家网络前三层保留的信息要少得多。



(a) (b) (c)

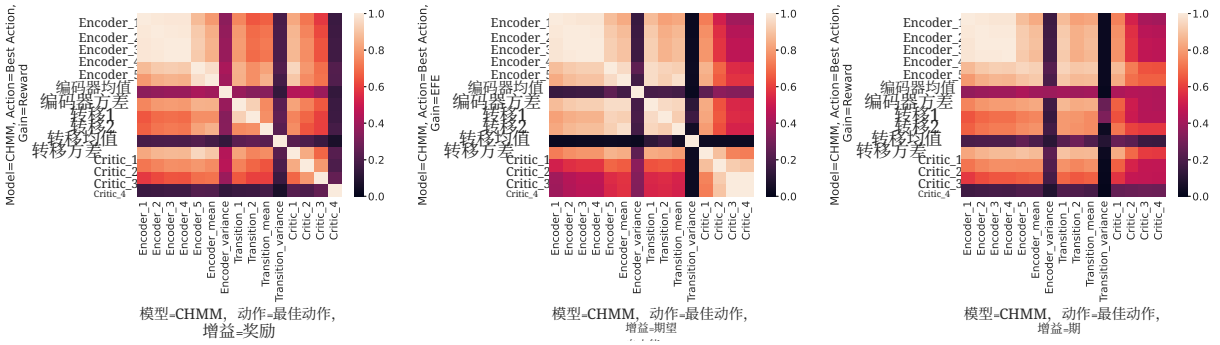
图 27：（a）展示了一个评论家网络最大化奖励（采用 $\hat{\epsilon}$-贪心选择）的 CHMM 中，编码器、转移网络和评论家网络各层学习到的表征之间的相似性。（b）展示了一个评论家网络最小化期望自由能（采用 $\hat{\epsilon}$-贪心选择）的 CHMM 中，编码器、转移网络和评论家网络各层学习到的表征之间的相似性。（c）展示了两个 CHMM 之间学习到的表征的相似性，其中一个的评论家网络优化期望自由能，另一个优化奖励（二者均采用 $\hat{\epsilon}$-贪心选择）。



(a) (b)

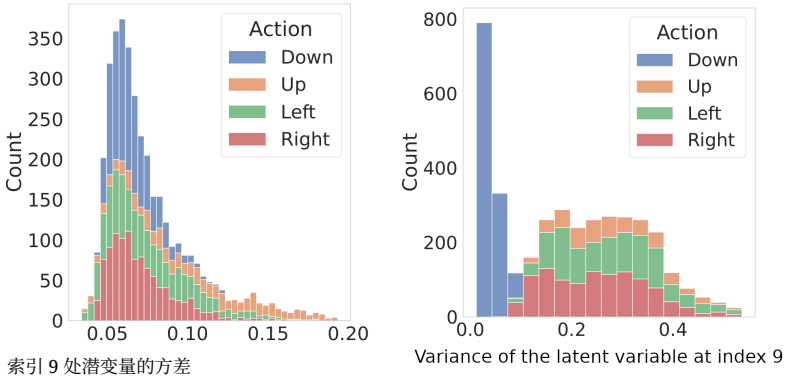
图 28：（a）展示了一个评论家网络最大化奖励（采用 $\hat{\epsilon}$-贪心选择）的 CHMM 与一个深度Q网络之间学习到的表征的相似性；（b）展示了一个评论家网络最小化期望自由能（采用 $\hat{\epsilon}$-贪心选择）的 CHMM 与一个深度Q网络之间学习到的表征的相似性

采用最佳动作选择 如图 24 所示，只有评论家网络最大化奖励的 CHMM 才能完成该任务。然而，从图 29 中可以看出，评论家网络最小化期望自由能的 CHMM，在大多数层学到的表征都与最大化奖励的 CHMM 相似，但编码器和转移网络的方差层（编码器方差和转移方差）除外。为了更好地了解两个模型的方差层所学表征之间的差异，我们将 5K 个状态-动作对输入转移网络，并展示了转移网络输出的方差分布。这项分析表明，最大化奖励时，转移网络（方差层）的方差非常小且变化不大；而最小化期望自由能时，方差更大且变化更多，如图 30 所示。这反映出最小化期望自由能的 CHMM 对转移具有更高的不确定性。具体而言，最小化期望自由能的 CHMM 似乎对向下动作很有把握，但对所有其他动作都非常不确定，这表明该 CHMM 总是选择向下动作，且没有为其他动作收集足够的数据。



(a) (b) (c)

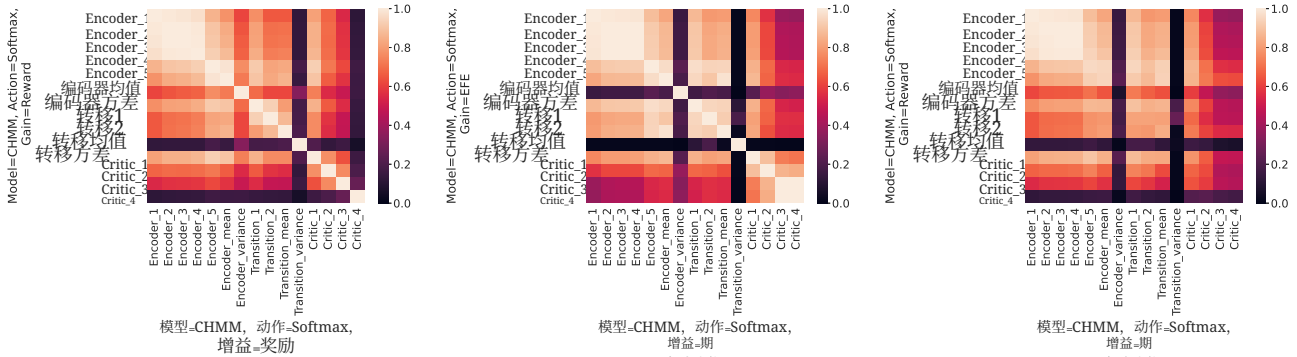
图 29：(a) 展示了一个以奖励最大化为目标的 CHMM（采用最佳动作选择）中，编码器、转移网络和评论家网络各层所学习表征之间的相似性。(b) 展示了一个以期望自由能最小化为目标的 CHMM（采用最佳动作选择）中，编码器、转移网络和评论家网络各层所学习表征之间的相似性。(c) 展示了两个 CHMM 所学习表征之间的相似性：一个以期望自由能优化为目标，另一个以奖励优化为目标（二者均采用最佳动作选择）。



(a) (b)

图 30：(a) 展示了以奖励最大化为目标的 CHMM 的转移网络方差层中的一个潜在维度。(b) 展示了以期望自由能最小化为目标的 CHMM 的转移网络方差层中的一个潜在维度。这两幅图均体现了两个模型中方差激活值的典型分布。请注意，对于以期望自由能最小化为目标的 CHMM，只有向下动作的方差较低。这表明该 CHMM 总是选择向下动作，且没有为其他动作收集足够的数据。

采用 softmax 动作选择的 CHMM 如图23所示，采用 softmax 动作选择的 CHMM 均未能完成任务。再次可以看到，评论家网络最小化期望自由能的 CHMM 与评论家网络最大化奖励的 CHMM，其转移网络的方差层方差存在显著差异（参见图31c 中两个转移方差的交点）。此外，如图32所示，在优化期望自由能时，转移网络输出的不确定性也呈现出相同的趋势。虽然这或许可以解释为何优化期望自由能的模型无法完成任务，却无法说明为何最大化奖励的模型也无法完成任务；我们可以推测，这些结果可能归因于 softmax 动作选择。更准确地说，如果评论家网络预测的值彼此非常接近，那么使用 softmax 采样的智能体可能会执行随机动作。



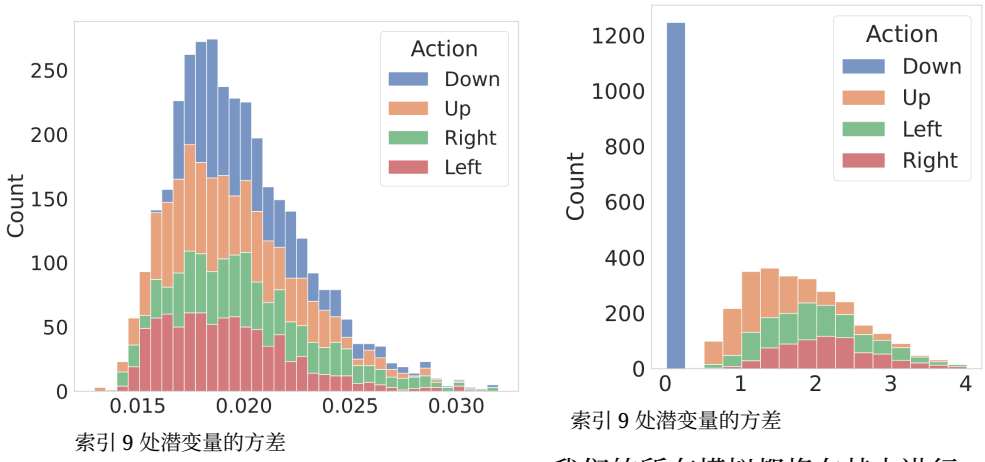
(a) (b) (c)

图31：(a) 展示了一个评论家网络最大化奖励的 CHMM（采用 softmax 动作选择）中，编码器、转移网络和评论家网络不同层所学习表征之间的相似性。(b) 展示了一个评论家网络最小化期望自由能（EFE）的 CHMM（采用 softmax 动作选择）中，编码器、转移网络和评论家网络不同层所学习表征之间的相似性。(c) 展示了两个 CHMM 所学习表征之间的相似性：一个的评论家网络优化 EFE，另一个优化奖励（二者均采用 softmax 动作选择）。

4.4.2 DEGENERATE BEHAVIOUR WITH THE EXPECTED FREE ENERGY?

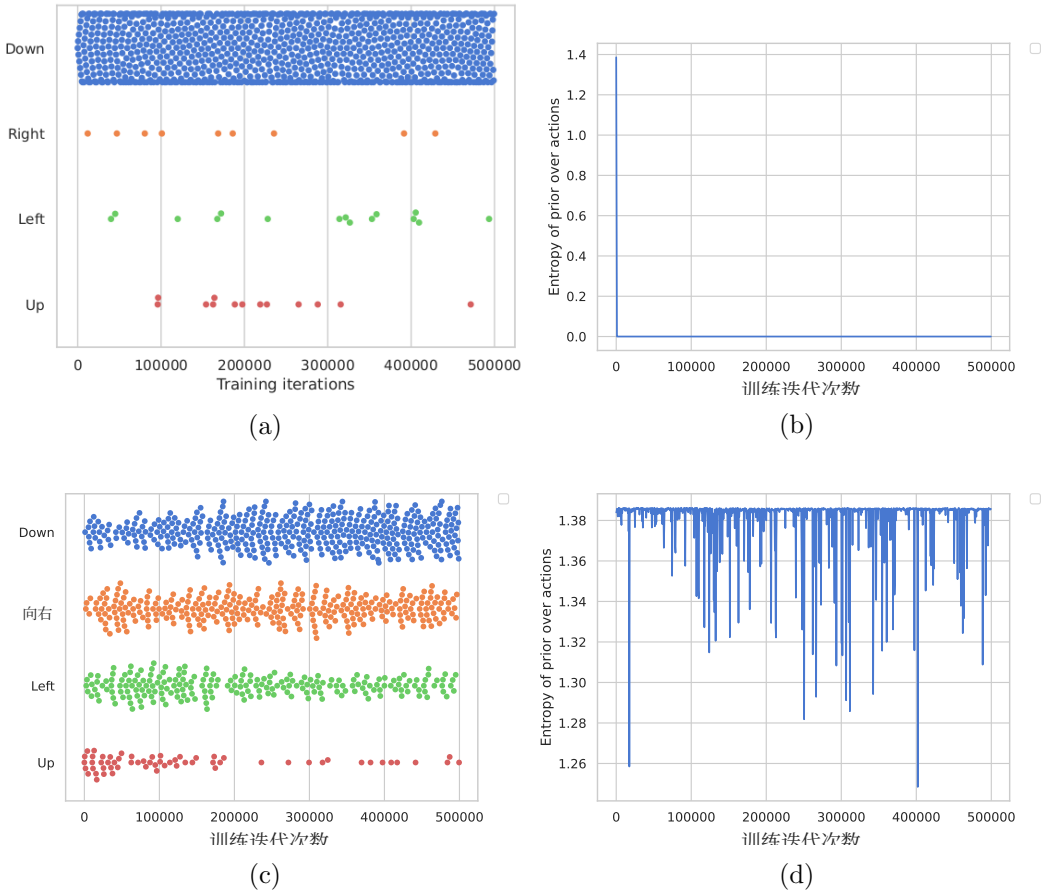
到目前为止，我们看到，最小化期望自由能（EFE）的 CHMM 无法完成任务。此外，我们发现，对于向上、向左和向右动作，转移网络存在不确定性，这表明最小化 EFE 的 CHMM 总是执行向下动作。图33证实了这一点。事实上，图33a 显示，最小化 EFE 的智能体几乎只选择向下动作；图33b 则显示，动作先验的熵很快就收敛到零。

相比之下，最大化奖励的 CHMM 会不断选择向右和向左的动作，从而将图形拖向相应的角落（见图33c）。此外，如图33d所示，动作先验的熵仍远高于零。请注意，最小化期望自由能的 CHMM 与最大化奖励的 CHMM 之间的唯一区别是信息增益，其定义为转移网络与编码器网络输出之间的 KL散度。由于期望自由能被最小化，这两个网络的输出需要尽可能接近。



(a) (b)

图32: （a）显示了奖励最大化的 CHMM 转移网络方差层的一个潜在维度。（b）显示了期望自由能最小化的 CHMM 转移网络方差层的一个潜在维度。两幅图均代表了两个模型中方差激活值的分布。请注意，对于期望自由能最小化的 CHMM，只有向下动作的方差较低。这表明，期望自由能最小化的 CHMM 总是选择向下动作，因此无法为其他动作收集足够的数据。



(a) (b) (c) (d)

图33: （a）显示了期望自由能最小化的 CHMM 在每次规划迭代中采取的动作。（b）显示了期望自由能最小化的 CHMM 的动作先验熵。（c）显示了奖励最大化的 CHMM 在每次规划迭代中采取的动作。（d）显示了奖励最大化的 CHMM 的动作先验熵。

这表明，最小化期望自由能的 CHMM 会选择单一动作（向下），并在选择该动作时成为预测未来的专家。这实际上使转移网络和编码器网络的输出之间的 KL散度变小。此外，选择向下动作时，平均奖励为零，因为在 dSprites 数据集中，图像左侧和右侧的形状数量相同；而当智能体穿过底线时，随着它接近某个角落，获得的奖励会线性增加（或减少），并且在图像中心处为零。对于所有其他动作，由于智能体在未穿过底线的情况下经过 50 个动作-感知循环后，试验就会中断，并且智能体会收到 -1 的奖励，因此期望奖励为负。由此，如果 CHMM 必须坚持只选择一个动作来保持较小的 KL散度，那么它能选择的最佳动作就是向下，即向下动作具有最高的期望奖励。

此外，图34展示了将 X% 的信息增益加入目标函数所产生的影响：智能体起初只最大化奖励（参见公式19），经过 200K 次训练迭代后，则最小化奖励加上 X% 信息增益。可以看出，即使只加入 1% 的信息增益，所获得的奖励也会大幅减少。

总之，同样是这种旨在赋予最小化期望自由能的智能体探索行为的信息增益，也会阻碍智能体解决 dSprites环境。这是因为智能体被限制为只能选择一个动作，从而导致策略次优。

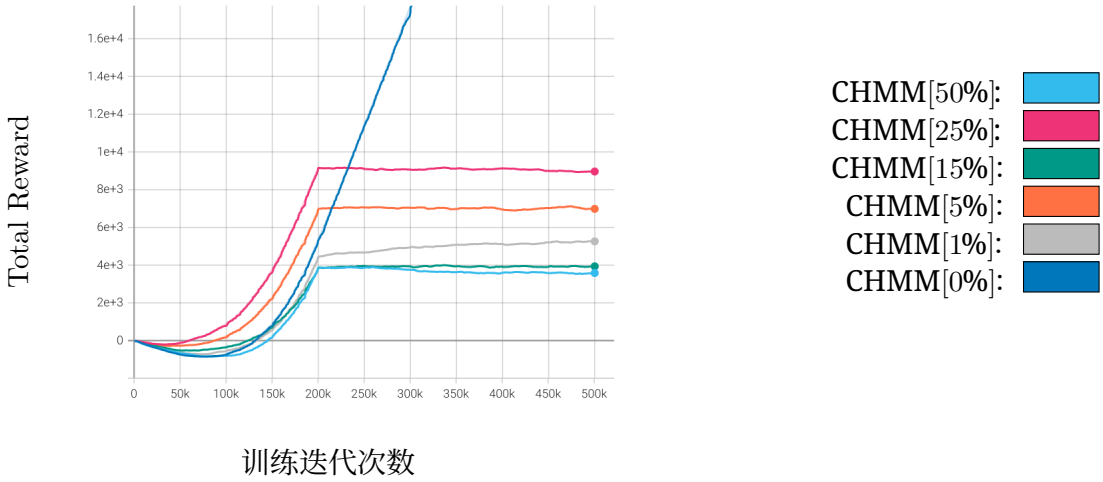


图34：该图展示了 CHMM 智能体在50万次训练迭代中累积的总奖励。所有智能体最初都只最大化奖励；经过20万次训练迭代后，信息增益的 X% 被加入目标函数。请注意，即使只加入1%的信息增益，也足以大幅降低智能体累积的总奖励。20万次迭代之前轨迹的差异是任意的，源于随机初始化的差异。

4.5 DAI 智能体

本节报告 DAI 智能体采用不同动作选择策略和不同期望自由能定义时所得到的结果。首先，15 个 DAI 智能体中，大多数因数值不稳定性而崩溃，也就是说，变分自由能突然变成了“非数值”。唯一存活下来（即未崩溃）的 DAI 智能体，在进行动作选择时采用 softmax采样，同时最大化奖励。图38显示，DAI 智能体成功地最小化了变分自由能；但如图37所示，它并未完成任务，表现与随机智能体相当。最后，图35展示了 DAI 智能体经过50万次训练迭代后生成的图像序列。

请注意，虽然智能体没有完成该任务，但它相当了解环境动态。不过，智能体难以处理代表心形的图像。

比较图 21、26 和 35 可见，采用 softmax采样的 DAI 智能体，其重建效果优于使用 $\hat{\epsilon}$-贪心算法的 CHMM 智能体（见图 26）。相较之下，DAI 智能体对图像序列的重建效果不如执行随机动作的 HMM 智能体（见图 21）。

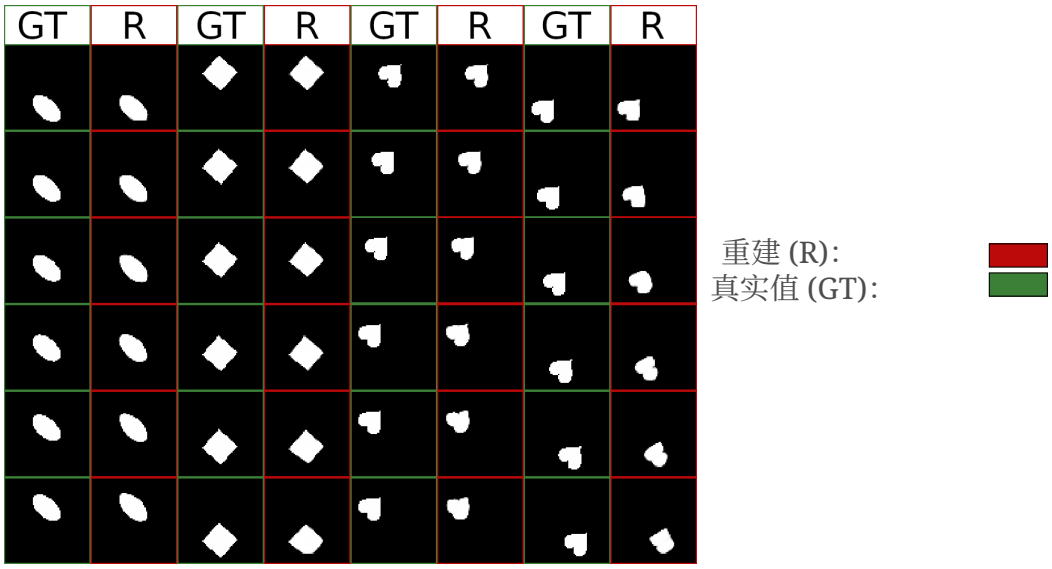
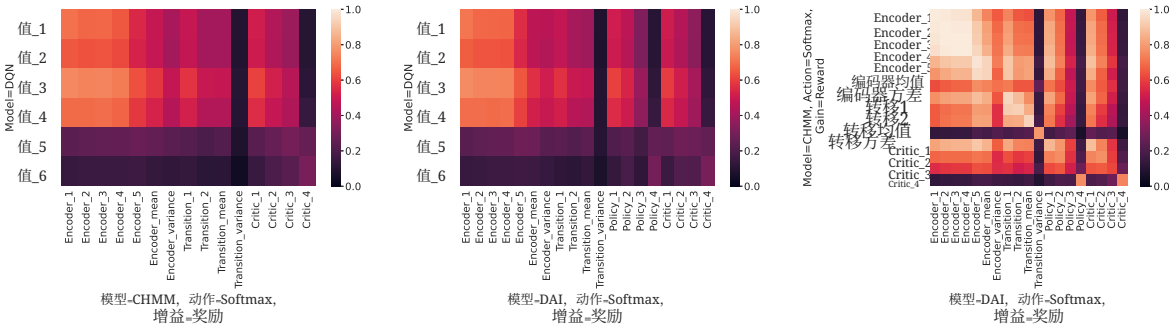


图35：此图展示了 DAI 在经过 50万次训练迭代后生成的重建图像序列。各列依次交替显示真实图像和重建图像。时间沿垂直方向流逝（从上到下），且每列中都重复执行相同的动作。



(a) (b) (c)

图36：（a）显示了通过最大化奖励并采用 softmax动作选择学习到的深度Q网络与 CHMM 的表征之间的相似性。（b）显示了通过最大化奖励并采用 softmax动作选择学习到的深度Q网络与 DAI 的表征之间的相似性。（c）显示了 CHMM 与 DAI 的表征之间的相似性。两者均最大化奖励并采用 softmax动作选择。

如前所述，训练期间唯一没有崩溃的 DAI 使用了 softmax动作选择，且其评论家网络以最大化奖励为目标。从图36b和图36a可以看出，该 DAI 与深度Q网络之间的表征相似性，与使用相同动作选择方式并以最大化奖励为目标的深度Q网络和 CHMM 之间的表征相似性非常接近。图36c 中 CHMM 与 DAI 模型的比较进一步证实了这一点。有趣的是，我们可以看到策略网络和评论家网络学习到了相似的表征，这表明策略网络正在正确地学习。然而，我们之前推断 softmax动作选择可能并非最优；鉴于 DAI 无法解决该任务，这一结论似乎同样适用于 DAI。

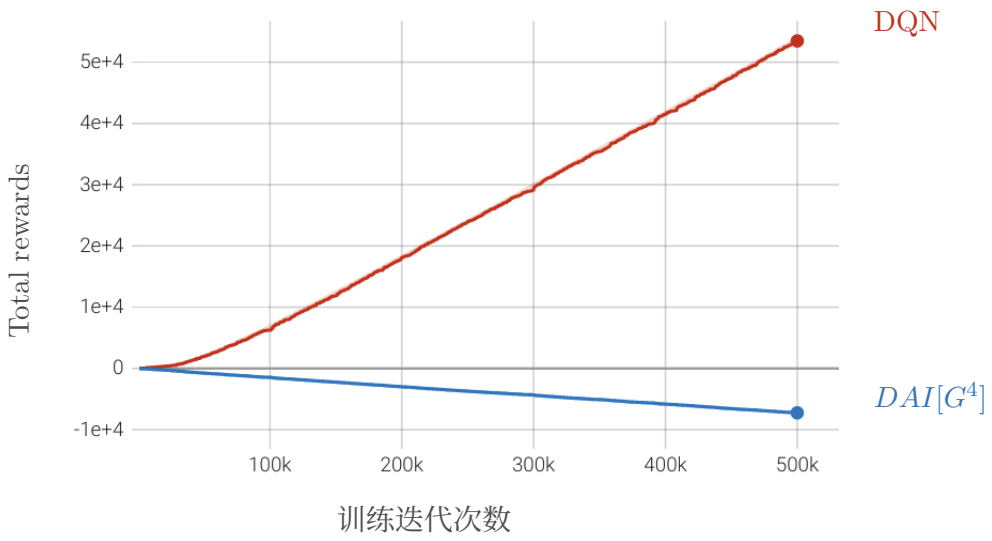


图37：此图展示了 DAI 智能体在训练的 50万次迭代期间获得的奖励总量。该智能体在从策略网络输出的 softmax 函数中采样动作的同时，最大化奖励。简而言之，DAI 智能体无法完成任务，其表现与随机智能体相当。

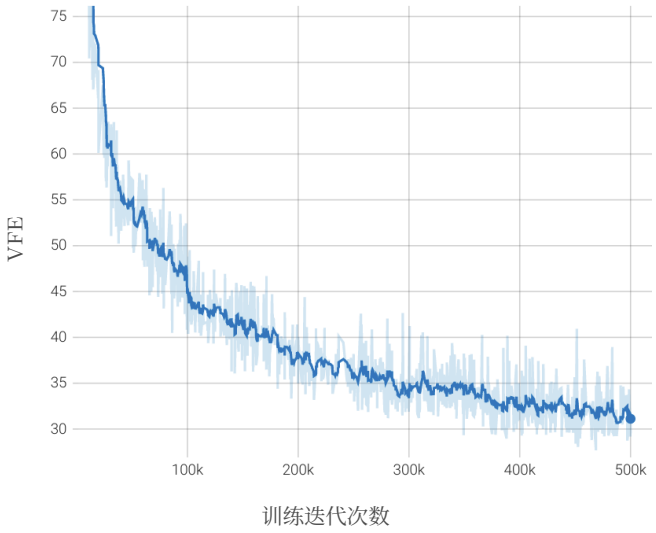


图38：此图展示了 DAI 智能体在训练的 50万次迭代期间的变分自由能。该智能体在从策略网络输出的 softmax 函数中采样动作的同时，最大化奖励。该智能体能够最小化其变分自由能。

5. 认知价值的讨论

在本节中，我们将讨论如何将期望自由能分解为认知价值和外在价值。更具体地说，鉴于深度主动推断智能体（例如 CHMM 和 DAI）的结果不佳，我们试图理解认知价值对性能造成的不利影响。假设认知价值的定义如下：

$$\boldsymbol{EV} = \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln P(s_\tau|o_\tau) - \ln Q(s_\tau|\pi)\big],$$

我们设置了两个实验（详情请参见附录 C），其中所有分布都存储在表格中。换句话说，以下分类分布使用矩阵表示： $P(o_\tau|s_\tau)$ 、 $Q(s_\tau|\pi)$ 和 $P(s_\tau|o_\tau)$ 。

在第一个实验中，状态 $P(s_\tau)$ 的先验保持不变，而似然 $P(o_\tau|s_\tau)$ 变得越来越均匀。在此过程中，真实后验 $P(s_\tau|o_\tau)$ 变得越来越接近近似后验 $Q(s_\tau|\pi)$ ，并且认知价值逐渐降低；参见图 39 的左侧面板。这是预期的行为，在这种情况下，认知价值会促进探索。

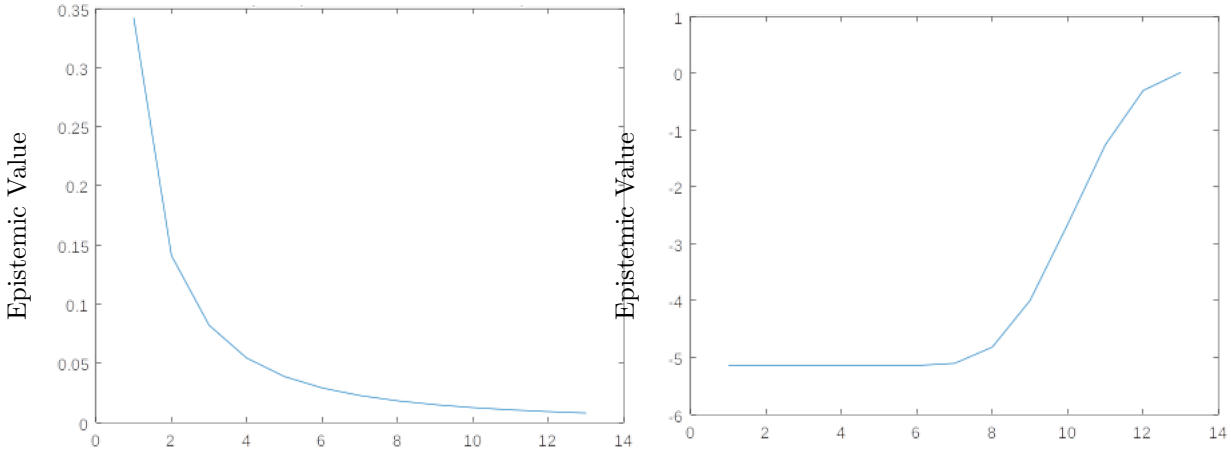
在第二个实验中，似然 $P(o_\tau|s_\tau)$ 是固定的，并且熵相当高；与此同时，状态上的先验 $P(s_\tau)$ 沿状态轴向一个方向偏移。因此，真实后验$P(s_\tau|o_\tau)$ 与近似后验 $Q(s_\tau|\pi)$ 变得更加不同，但这会导致认知价值降低；见图 39 右图。这是一种退化行为，因为在这种设置下，认知价值会阻碍探索。

总而言之，正如 Parr 和 Friston（2019）的公式（10）所示，将期望自由能分解为认知价值和外在价值似乎会表现出两种截然不同的行为，具体取决于分布的定义方式，参见图 39。这一点在深度主动推理文献中尤为重要，因为该文献以此方程为基础。例如，第 4.4.2 节中的图表表明，最小化期望自由能的 CHMM 智能体几乎完全专注于“向下”这一动作，即它没有进行探索，从而导致性能不佳。

6. 结论

本文质疑了“深度主动推理已是一个解决了的问题”这一普遍假设，并指出了该领域要取得进展仍需解决的挑战。我们回顾了八种实现深度主动推理的方法：(a) DAI_{MC}Fountas 等人（2020）提出的方法，(b) DAI_{VPG}Millidge（2020）提出的方法，(c) DAI_{RHI} Rood 等人（2020）提出的方法，(d) DAI_{HR} Sancaktar 等人（2020）提出的方法，(e) DAI_{FA} Ueltzh¨offer（2018）提出的方法，(f) DAI_{POMDP}van der Himst 和 Lanillos（2020）提出的方法，(g) DAI_{SSM}C¸atal 等人（2020）提出的方法，以及 (h) Schneider 等人（2022）提出的方法，其代码未能在网上获取。

总体而言，这些方法带来了许多有趣的想法，例如：使用深度神经网络预测相关分布的参数，在主动推断中使用蒙特卡洛树搜索进行规划，以及使用期望自由能的自助法估计来训练评论家网络。然而，我们难以复现其中一些声称的结果，例如在 Animal-AI环境中训练 DAI_{MC} ；有些情况下，我们也无法获取代码。理想情况下，未来研究应借鉴开放科学框架的做法，将产生所声称结果的代码开源。此外，不同论文对期望自由能的定义各不相同。这表明，还需要进一步研究，以阐明表格型主动推断和深度主动推断中期望自由能的定义及其合理性（参见第5节和附录 C）。总之，近期关于深度主动推理的研究（Fountas 等人，2020年；Millidge，2020年；Rood 等人，2020年；Sancaktar 等人，2020年；Ueltzh¨offer，2018；van der Himst 和 Lanillos，2020年；C¸atal 等人，2020年）朝着构建完整的深度主动推理智能体迈出了重要的第一步，但仍有许多细节需要完善，例如期望自由能的定义与推导，以及研究的可复现性。有关更多详情，请参阅第2节。



后验分布的相似度： $P(s_\tau|o_\tau)$ 和 $Q(s_\tau|\pi)$　　后验分布的相似度： $P(s_\tau|o_\tau)$ 和 $Q(s_\tau|\pi)$

图 39：最左侧的图展示了第一个实验的结果，其中似然逐渐变得越来越均匀。在这种情况下，认知价值会鼓励探索，因为最大化认知价值会使真实后验分布 $P(s_\tau|o_\tau)$ 与近似后验$Q(s_\tau|\pi)$ 尽可能不同，即在 x 轴上向左移动。然而，最右侧的图展示了第二个实验的结果，其中状态的先验发生了变化。在这种情况下，认知价值不会促进探索，因为最大化认知价值会使真实后验分布 $P(s_\tau|o_\tau)$ 与近似后验 $Q(s_\tau|\pi)$ 尽可能相似，即在 x 轴上向右移动。事实上，在这种情况下，最大化认知价值会导致信息丢失。

在回顾现有研究后，我们尝试逐步实现一个深度主动推理智能体。首先，我们构建了一个采取随机动作的变分自编码器智能体。该智能体能够学习到任务的有用（潜在）表征。然而，由于该智能体采取随机动作，因此无法完成任务。随后，我们添加了转移网络，构建了隐马尔可夫模型（HMM）智能体，该智能体同样采取随机动作。该智能体能够学习到任务及其动态特性的良好表征，但无法完成任务。

接下来，我们尝试将评论家网络纳入该方法，从而得到评论家隐马尔可夫模型（CHMM）。在此背景下，我们对期望自由能的几种可能实现进行了实验。我们还尝试移除信息增益，仅预测奖励。此外，我们实现了三种动作选择策略，即：根据期望自由能选择最佳动作、softmax采样以及ε-贪婪。

使用ε-贪婪算法时，只有奖励最大化智能体能够完成任务。然而，与简单的深度Q网络相比，该智能体需要更多的训练迭代次数才能学会正确的行为。这可能是因为，CHMM不仅需要学习如何完成任务，还需要学习环境动态。使用softmax采样时，所有智能体都未能完成任务。其中一个智能体的表现甚至比选择随机动作的智能体还差。最后，在选择最佳动作时，只有奖励最大化智能体能够完成任务。重要的是，根据我们的实验，使用ε-贪婪算法的智能体获得了最高的累积奖励，并且最快学会完成任务。此外，奖励最大化智能体能够正确完成任务，但其潜在表征的质量不如HMM智能体。这可能是因为，在执行奖励最大化动作时，可用于学习环境模型的数据缺乏多样性，即探索不足。

接下来，我们尝试将策略网络融入该方法，从而构建出完整的深度主动推理智能体（DAI）。与 C HMM 智能体一样，我们尝试了几种可能的实现方式

来计算期望自由能，尝试去除信息增益、仅预测奖励，并实现了三种动作选择策略，分别是：根据期望自由能选择最佳动作、softmax采样和ε-贪婪算法。使用ε-贪婪算法或选择最佳动作时，所有智能体都未能解决任务。使用softmax采样时，大多数智能体出现数值不稳定并崩溃，其余智能体也未能解决任务。

最后，我们使用中心化核对齐比较了不同模型（例如深度Q网络、CHMM、DAI等）各层所学习表征的相似性。结果显示，DQN在最初几层学到了通用特征，而在最后两层学到的特征则非常专门化。VAE在最初几层学到的特征与DQN相似，但在最后两层与DQN有所不同，这反映了学习目标的差异。类似地，HMM在最初几层学到的特征与DQN相似，但在最后两层，其学习到的表征则与DQN不同。此外，HMM学习到的均值表征和方差表征与VAE中相应的表征不同，这表明转移网络会影响模型的潜在表征。

此外，使用 ϵ-贪心算法进行动作选择时，最大化奖励的CHMM学到的表征比最小化期望自由能的CHMM学到的表征更接近DQN。重要的是，最大化奖励的CHMM中的评论家网络保留了比最小化期望自由能的CHMM更多来自编码器的信息。选择最佳动作（根据评论家网络判断）时，最大化奖励的CHMM与最小化期望自由能的CHMM学到的表征非常相似，只有转移网络和编码器网络的方差层例外。进一步检查这些（方差）层后，我们发现，最大化奖励的CHMM中的转移网络比最小化期望自由能的CHMM中的转移网络确定得多。更准确地说，最小化期望自由能的CHMM只有在执行向下动作时，才对世界状态转移有信心。这表明，最小化期望自由能的CHMM总是选择向下动作，因此没有为其他动作收集足够的数据。随着训练的推进，可视化所选动作的分布也印证了这一点：最小化EFE的智能体几乎只选择向下动作。相比之下，最大化奖励的CHMM会持续选择向左和向右动作，从而成功完成任务。这两个CHMM的唯一区别在于认知价值，其目标是尽可能缩小转移网络和编码器网络的输出之间的差异。因此，最小化期望自由能的CHMM会选择单一动作（向下），并逐渐成为预测执行该动作后未来情况的专家。这实际上使转移网络和编码器网络输出之间的KL散度变小。此外，选择向下动作时，平均奖励为零，而选择其他所有动作时，期望奖励都为负。因此，如果CHMM必须坚持选择单一动作以保持KL散度较小，那么向下动作就是奖励最高的动作。

关于方差层的相同观察也适用于使用 softmax采样进行动作选择的连续隐马尔可夫模型。虽然这或许能解释为什么优化期望自由能的模型无法完成任务，但无法解释为什么最大化奖励的模型也无法完成任务；我们可以假设，这些结果可能是由 softmax动作选择造成的。更确切地说，如果评论家网络预测的值彼此非常接近，那么使用 softmax采样的智能体可能会执行随机动作。请注意，提高增益参数可能有助于智能体区分彼此接近的值。

此外，DAI（通过 softmax采样最大化奖励）与深度Q网络之间的表征相似性，与采用连续隐马尔可夫模型的深度Q网络（通过 softmax采样最大化奖励）的表征相似性非常接近。此外，DAI的策略网络和评论家网络学习到了相似的表征，这表明策略网络正在正确学习。因此，DAI（通过 softmax采样最大化奖励）在 dSprites环境中失败，很可能是由于 softmax动作选择，而不是模型所学习的表征。

最后，我们对深度主动推理中常用的期望自由能（Fountas 等人，2020年）的研究表明，在某些情况下，退化行为可能由此产生。这或许可以解释为何

在我们的规划目标中加入认知价值，似乎会对智能体探索其环境并获取信息的能力产生如此有害的影响。在深度学习语境下使用这一定义的期望自由能，可能正是我们难以实现深度主动推理的根本原因之一，也可能解释了深度主动推理文献中存在的一些表述不确定性（例如添加负号）。

总而言之，深度主动推理领域受益于强化学习和深度学习文献中的大量不同思想。未来，如果能够提出一种满足以下五项要求的方法，将会很有价值：（i）该方法是完整的，即由编码器、解码器、转移网络、策略网络以及（可选的）评论家网络组成；（ii）该方法所依据的数学理论没有错误，并且与自由能原理一致；（iii）实现与数学理论一致；（iv）代码公开可用，从而能够验证实现的正确性并复现结果；（v）该方法能够解决输入空间较大的任务，例如基于图像的任务。我们相信，这样的方法将通过提供一个强大且可复现的基线，使未来的研究能够以此进行基准测试，从而促进深度主动推理领域的发展。

致谢

我们感谢 Karl J. Friston、Thomas Parr、Lancelot Da Costa 和 Zafeirios Fountas 就本文开展的有益讨论并提供反馈，也感谢他们向我们推荐了重要资源，例如他们的新书和若干论文。

参考文献

Bellman R（1952年）论 动态规划理论。《美国国家科学院院刊》38(8)：716–719

Bonheme L，Grzes M（2022年）变分自编码器如何学习？来自表征相似性的洞见。arXiv 预印本 arXiv:2205.08399，2205.08399

Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, Perez D, Samothrakis S, Colton S (2012) 蒙特卡罗树搜索方法综述。《IEEE计算智能与AI游戏汇刊》4(1):1–43

C¸atal O, Wauthier S, De Boom C, Verbelen T, Dhoedt B (2020年) 学习用于主动推断的生成式状态空间模型。《计算神经科学前沿》14:574,372

Champion T, Grze´s M, Bowman H (2021) 在变分消息传递中实现主动推断：结果盲确定性寻求者。《神经计算》33(10):2762–2826, DOI 10.1162/neco a 01422, URL https://doi.org/10.1162/neco_a_01422, https://direct.mit.edu/neco/article-pdf/33/ 10/2762/1963368/neco_a_01422.pdf

Champion T, Bowman H, Grze´s M（2022a）分支时间主动推断：实证研究与复杂性类别分析。神经网络 152：450–466，DOI https://doi.org/10.1016/j.neunet.2022.05.010，URL https://www.sciencedirect.com/science/article/pii/S0893608022001824

Champion T, Da Costa L, Bowman H, Grze´s M（2022b）分支时间主动推断：理论及其一般性。神经网络 151：295–316，DOI https://doi.org/10.1016/j.neunet.2022.03.036，URL https://www.sciencedirect.com/science/article/pii/S0893608022001149

Champion T, Grze´s M, Bowman H（2022c）《分支时间主动推断与贝叶斯滤波》。神经计算 34(10):2132–2144，DOI 10.1162/neco a 01529，网址 https://doi.org/10.1162/

neco_a_01529, https://direct.mit.edu/neco/article-pdf/34/10/2132/2042425/neco_a_01529. pdf

Champion T, Grze´s M, Bowman H (2022d) 多模态多因素分支时间主动推断。DOI 10.48550/ARXIV.2206.12503，URL https://arxiv.org/abs/2206.12503 Cortes C, Mohri M, Rostamizadeh A (2012) 基于中心化对齐的卷积核学习算法。J Mach Learn Res 13(1):795–828 Costa LD, Parr T, Sajid N, Veselic S, Neacsu V, Friston K (2020a) 离散状态空间上的主动推断：综合。2001.07203 Costa LD, Sajid N, Parr T, Friston K, Smith R (2020b) 动态规划与主动推断之间的关系：离散有限时域情形。2009.08111

Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola JS (2002) 关于核-目标对齐。载于：Dietterich TG, Becker S, Ghahramani Z (编) 神经信息处理系统进展，第14卷，麻省理工学院出版社，温哥华，加拿大，第367–373页，URL http://papers.nips.cc/paper/1946-on-kernel-target-alignment.pdf

Cullen M、Davey B、Friston KJ、Moran RJ（2018）OpenAI Gym中的主动推断：一种针对精神疾病的计算研究范式。《生物精神病学：认知神经科学与神经影像学》3(9):809 – 818，DOI https://doi.org/10.1016/j.bpsc.2018.06.010，URL http://www.sciencedirect.com/science/article/pii/S2451902218301617，《精神病学中的计算方法与建模》

Doersch C（2016）变分自编码器教程。1606.05908

FitzGerald THB、Dolan RJ、Friston K（2015）多巴胺、奖励学习与主动推断。《计算神经科学前沿》9:136，DOI 10.3389/fncom.2015.00136，URL https://www.frontiersin. org/article/10.3389/fncom.2015.00136

Fountas Z, Sajid N, Mediano PAM, Friston K (2020年) 使用蒙特卡洛方法的深度主动推断智能体。2006.04176

Friston K, FitzGerald T, Rigoli F, Schwartenbeck P, Doherty JO, Pezzulo G (2016) 主动推断与学习。《神经科学与生物行为评论》68:862 –879，DOI https://doi.org/10.1016/j.neubiorev. 2016.06.022

van Hasselt H, Guez A, Silver D (2005) 使用希尔伯特-施密特范数测量统计依赖性。载于：Jain S, Simon HU, Tomita E (编)《算法学习理论》，Springer Berlin Heidelberg，第63–77页

van Hasselt H, Guez A, Silver D (2015) 使用双重 Q 学习的深度强化学习。1509.06461 Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-变分自编码器：在受约束的变分框架下学习基本视觉概念。载于：第5届国际学习表征会议，ICLR 2017，土伦，法国，2017年4月24日至26日，会议论文集，OpenReview.net，网址 https://openreview.net/forum?id=Sy2fzU9gl

van der Himst O, Lanillos P（2020年）用于部分可观测马尔可夫决策过程的深度主动推理。CoRR abs/2009.03622，URL https://arxiv.org/abs/2009.03622，2009.03622

Itti L, Baldi P (2009) 贝叶斯惊奇会吸引人类注意力。Vision Research 49(10):1295 – 1306，DOI https://doi.org/10.1016/j.visres.2008.09.007，URL http://www.sciencedirect.com/science/article/pii/S0042698908004380，视觉注意：心理物理学、电生理学与神经影像

Kingma DP, Welling M (2014) 自动编码变分贝叶斯。载于：国际机器学习会议，班夫，加拿大，第 2 卷，URL http://arxiv.org/abs/1312.6114

Koller D, Friedman N (2009) 概率图模型：原理与技术。麻省理工学院出版社

Kornblith S, Norouzi M, Lee H, Hinton G (2019) 重新审视神经网络表征的相似性。载于：Chaudhuri K, Salakhutdinov R (编) 第36届国际机器学习会议论文集，PMLR，美国长滩，《机器学习研究会议录》，第 97 卷，第 3519–3529 页，URL http://proceedings.mlr.press/v97/kornblith19a.html

Lample G、Chaplot DS（2016）使用深度强化学习玩第一人称射击游戏。1609.05521

Lanillos P、Cheng G 等（2020年）机器人自我/他者区分：主动推断与神经网络在镜像中的学习相遇。arXiv 预印本 arXiv:200405473

Maheswaranathan N、Williams A、Golub M、Ganguli S、Sussillo D（2019）大规模循环网络群体中神经动力学的普适性与个体性。载于：Wallach H、Larochelle H、Beygelzimer A、d'Alch´e-Buc F、Fox E、Garnett R（编），《神经信息处理系统进展》，Curran Associates, Inc.，第32卷，网址：https://proceedings.neurips.cc/paper/2019/file/5f5d472067f77b5c88f69f1bcfda1e08-Paper.pdf

Matthey L、Higgins I、Hassabis D、Lerchner A（2017）dSprites：解耦测试精灵数据集。https://github.com/deepmind/dsprites-dataset/

Millidge B（2019）结合主动推断与分层预测编码：教程式介绍与案例研究。URL https://doi.org/10.31234/osf.io/kf6wc

Millidge B（2020年）深度主动推理作为变分策略梯度。《数学心理学杂志》96:102,348，DOI https://doi.org/10.1016/j.jmp.2020.102348，URL http://www.sciencedirect.com/science/article/pii/S0022249620300298

Mnih V、Kavukcuoglu K、Silver D、Graves A、Antonoglou I、Wierstra D、Riedmiller M（2013）使用深度强化学习玩 Atari。1312.5602

Oliver G、Lanillos P、Cheng G（2019）人形机器人中的主动推断、身体感知与动作。arXiv预印本 arXiv:190603022

Parr T、Friston KJ（2019）广义自由能与主动推断。《生物控制论》113（5-6）：495–513

Pezzato C、Hernandez C、Wisse M（2020）机器人学中用于反应式动作规划与执行的主动推断和行为树。2011.09756

Rezende DJ、Mohamed S、Wierstra D（2014）深度生成模型中的随机反向传播与近似推断。载于：Xing EP、Jebara T（编）《第31届国际机器学习会议论文集》，PMLR，中国北京，《机器学习研究会议录》，第32卷，第1278–1286页，网址：http://proceedings.mlr.press/v32/rezende14.html

Robert P、Escoufier Y（1976年）线性多元统计方法的统一工具：RV系数。《英国皇家统计学会会刊C辑（应用统计）》25(3)：257–265，网址：http://www.jstor.org/stable/2347233

Rood T, van Gerven M, Lanillos P（2020年）橡胶手错觉的深度主动推理模型。载于：Verbelen T, Lanillos P, Buckley CL, De Boom C（编），《主动推断》，施普林格国际出版，尚，页码 84–91

Sancaktar C, van Gerven M, Lanillos P（2020年）用于身体感知和动作的端到端基于像素的深度主动推理。2001.05847

Schneider T（无日期）用于机器人操作的主动推断，未发表

Schneider T, Belousov B, Abdulsamad H, Peters J（2022年）用于机器人操作的主动推断。arXiv 预印本 arXiv:220610313

Schwartenbeck P, Passecker J, Hauser TU, FitzGerald THB, Kronbichler M, Friston K（2018）好奇心与目标导向探索的计算机制。bioRxiv DOI 10.1101/411272，网址 https://www.biorxiv.org/content/early/2018/09/07/411272，https://www.biorxiv.org/content/early/2018/09/07/411272.full.pdf

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, NhamJ, Kalchbrenner N, Sutskever I, Lillicrap TP, Leach M, Kavukcuoglu K, Graepel T, Hassabis D（2016）利用深度神经网络和树搜索掌握围棋。《自然》529(7587):484–489，DOI 10.1038/nature16961，URL https: //doi.org/10.1038/nature16961

Sutton RS, Barto AG, 等（1998年）强化学习导论。麻省理工学院出版社，剑桥

Ueltzh¨offer K（2018）深度主动推理。Biol Cybern 112(6):547–573，DOI 10.1007/s00422-018-0785-7，URL https://doi.org/10.1007/s00422-018-0785-7

附录 A：符号说明

符号	含义
s_τ, o_τ, r_τ, a_τ	分别表示时间步 τ 的状态、观测、奖励和动作。
o^r_τ, $\hat{s}^r_\tau$, $\mathring{o}^r_\tau$	如图 40 所述，将奖励编码后的观测，隐藏的 将 o^r_τ 作为输入时，从编码器采样得到的状态，以及重建的观测 分别由解码器根据转移网络的输出重建。
$\hat{s}_\tau$, $\hat{o}_\tau$	将 o_τ 作为输入时，在时间步 τ 从编码器采样得到的状态，以及图像 分别是在时间 τ 将$\hat{s}_\tau$ 作为输入时由解码器重建的。
$\mathring{o}_\tau$	转移网络预测的时间步 τ 的观测。
$s_{i:j}$, $o_{i:j}$, $a_{i:j}$	分别表示时间步 i 至 j （含）之间的状态、观测和动作集合。
π, π_τ, π'	一个策略，即动作序列；该策略在时间步 τ规定的动作；以及 另一个尺寸小于或等于 π 尺寸的策略。
$\mathcal{A}$, Π	可能动作集合和可能策略集合。
$\mathbb{A}_s$	节点 s 的所有祖先节点集合。
$\#A$,$\#C$, $\#S$, $\#O$	动作、通道、状态和观测的数量。
$\mathcal{Q}_{\theta_a}$, $\hat{\mathcal{Q}}_{\hat{\theta}_a}$	由 θ_a参数化的Q网络，以及由 $\hat{\theta}_a$
$\mathcal{G}_{\theta_a}$, $\hat{\mathcal{G}}_{\hat{\theta}_a}$	由 θ_a 参数化的评论家网络，以及由 $\hat{\theta}_a$.
$\mathcal{E}_{\phi_s}$, $\mathcal{D}_{\theta_o}$	由 ϕ_s 参数化的编码器网络，以及由 θ_o 参数化的解码器网络。
$\mathcal{P}_{\phi_a}$	由 ϕ_a参数化的策略网络。
$\mathcal{T}_{\theta_s}$, $\mathcal{T}_{\theta_o}$	分别由 θ_s 或 θ_o 参数化的转移网络。
θ, ϕ	生成模型和变分分布的所有参数。
a, b, c, d	计算 ω_t 时涉及的四个超参数。
$N(s_\tau, a_\tau)$	在状态 s_τ中，动作 a_τ 被探索的次数。
t, γ	当前时间步和折扣因子。
ζ, ψ	动作先验和先验偏好的精度。
ϵ, $\hat{\epsilon}$	重参数化技巧中使用的随机变量，以及 epsilon 的一个样本。
$\mathring{\epsilon}$	使用 $\mathring{\epsilon}$-贪心算法时，选择随机动作的概率。
T_{dec}	一个超参数，用于定义蒙特卡洛树搜索中对应于明显胜出的阈值。
$G(\pi)$, $G_\tau(\pi)$	策略 π 的期望自由能（期望自由能），以及在时间 步 τ 遵循策略 π 时收到的期望自由能，分别为。
$\bar{\boldsymbol{G}}_s$, $\boldsymbol{G}^{\text{aggr}}_s$, $\boldsymbol{G}_s$, $\boldsymbol{N}_s$	平均期望自由能、聚合期望自由能、期望自由能以及访问次数 分别对应节点 s。
μ_o, σ_o, μ_a, σ_a	DAI_{FA}的编码器和策略网络预测的均值和方差向量。
μ, σ	编码器预测的关于 s_t 的高斯分布的均值和方差。
μ, σ	转移预测的关于 s_{t+1} 的高斯分布的均值和方差。
$\hat{\mu}$, $\hat{\sigma}$	编码器预测的 s_{t+1} 高斯分布的均值和方差。
$\hat{\pi}$	策略网络预测的 a_t 分类分布的参数。
ω_t	调节转移映射精度的自上而下注意力参数。
[条件]	一个指示函数：如果条件成立，则取值为 1；否则取值为 0。
$\sigma[\bullet]$	softmax 函数。
Cat(x; ϕ_x)	关于 x 、参数化为 ϕ_x的分类分布。
伯努利分布(x; ϕ_x)	关于 x 、参数化为 ϕ_x的伯努利分布。
Bernoulli(x; ϕ_x)	关于 x 、参数化为 ϕ_x的伯努利分布乘积。
$\mathcal{N}(x; \mu_x, \sigma_x)$	x 上的多元高斯分布，其参数为均值向量 μ_x，以及 对角协方差矩阵，其对角元素为 σ_x。
$X \overset{i}{\to} Y$, $X \overset{m}{\to} Y$	X 作为输入传入 Y，而 X 预测的分布的均值作为输入传入 Y。
$X \overset{s}{\to} Y$, $X \to Y$	将 X 预测的分布中的一个 样本 作为输入馈送给 Y， X 输出 Y。

表1：第2节和第3节的符号说明。

附录 B： DAI_{MC} 论文与代码之间的差异

本节聚焦于作者在 GitHub 上提供的 DAI_{MC} 实现：https://github.com/zfountas/deep-active-inference-mc/。首先，根据与其中一位作者的个人交流，GitHub 上的代码（截至2022年6月6日）与论文实验所用的代码并不相同。下文将介绍论文与代码之间的差异。例如，论文中 ω_t 的计算方式如下：

$$\omega_t = \frac{a}{1+\exp(-\frac{b-D_t}{c})} + d,$$

而代码使用以下公式：

$$\omega_t = a \times \left(1 - \frac{1}{1+\exp\left(-\frac{D_t-b}{c}\right)}\right) + d.$$

此外，论文指出，训练期间会使用蒙特卡洛树搜索来计算策略先验。然而，代码中仅在测试模型时使用蒙特卡洛树搜索；也就是说，训练智能体时不使用蒙特卡洛树搜索。此外，论文指出，动作是通过从以下分布中采样来选择的：

$$\tilde{P}(a_t) = \frac{N(\hat{s}_t, a_t)}{\sum_{\hat{a}_t} N(\hat{s}_t, \hat{a}_t)}.$$

然而，代码会从树中的根节点递归地选取整个动作序列 $\pi = (\mathring{a}_t,\ \mathring{a}_{t+1},\ \ldots\ldots,\ \mathring{a}_{t+n})$。在递归的每一步中，都会选取访问次数 $\mathring{a}_\tau$ 最高的节点。随后，从序列中移除相互抵消的动作，例如，如果 $a_\tau = LEFT$ 且 $a_{\tau+1} = RIGHT$，则这两个动作都会从序列中移除。此过程会生成一个长度相同或更短的新动作序列 π'。最后，在环境中执行整个动作序列 π'。这样可以避免在每个动作-感知循环中重复规划过程（节省计算时间），但同时也需要领域知识（以移除相互抵消的动作）。

此外，论文中的实验既在 dSprites环境上进行，也在 Animal-AI环境上进行。然而，代码无法复现 Animal-AI环境上的结果，也就是说，处理 Animal-AI环境的代码已被移除。此外，期望自由能的评估并非易事（见下文），论文中也没有讨论相关细节。在解释期望自由能各项的计算方式之前，我们先介绍一些符号，以便简洁地表示这些计算步骤。例如，我们记作：

$$o_t^r \xrightarrow{i} \text{Encoder} \xrightarrow{s} \text{Transition} \xrightarrow{m} \mathring{s}_{t+1}^r,$$

这意味着将 o_t^r 用作编码器的输入（$\xrightarrow{i}$），然后从编码器预测的分布中采样（$\xrightarrow{s}$）一个状态，并将其用作转移网络的输入。最后，将转移网络预测的分布的均值（$\xrightarrow{m}$）用作 $\mathring{s}_{t+1}^r$ 的最大后验估计。请注意，转移网络有两个输入（即一个状态和一个动作）；使用简洁符号时，我们隐含地假设策略[8] π 指定的动作作为输入提供给转移网络。此外，对于每个时间步 τ，智能体获得的奖励 r_τ 会编码在图像 o_τ 的像素中，如图 40 所示，从而得到一幅新图像 o_τ^r。如图 41 右侧所示，编码器/解码器网络经过训练，用于预测生成的图像 o_τ^r。方程 (8) 中第一项的计算过程如图 41 左侧所示。简而言之，我们有：

$$o_t^r \xrightarrow{i} \text{Encoder} \xrightarrow{s} \text{Transition} \xrightarrow{s} \text{Decoder} \xrightarrow{m} \mathring{o}_{t+1}^r.$$

8. π 是正在计算期望自由能的策略 r 期望自由能正在被计算的情境 计算对象。

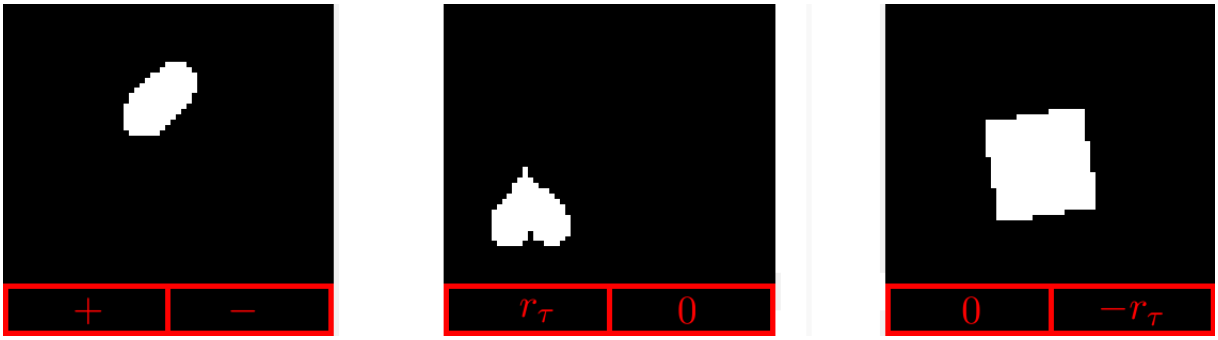


图 40：此图展示了奖励 $r_\tau \in [-1, 1]$ 如何编码到图像 o_τ中。向左，正负号分别标示奖励为正或负时，它在图像中的编码位置。中间，正奖励被编码在图像左侧。向右，负奖励被编码在图像右侧。

接下来，将一个编码了智能体可获得的最大奖励的矩阵（$\mathring{r}_{t+1}$）用作伯努利分布的参数，以计算重建图像 $\mathring{o}^r_{t+1}$ 前三行的概率的对数（即 $\boldsymbol{L}$）。请注意，如图 40 所述，前三行包含在时间 $t+1$ 获得的预测奖励。最后，计算 $\boldsymbol{L}$ 的均值，再将其乘以十，得到 $\mathbb{E}_{\tilde{Q}}[\ln P(o_\tau|\pi)]$。类似地，$H[Q(s_\tau|\pi)]$ 的计算过程如下：

$$o^r_t \xrightarrow{i} \text{Encoder} \xrightarrow{s} \text{Transition} \rightarrow \mathring{\mu}, \ln \mathring{\sigma},$$

其中，$Q(s_\tau|\pi)$等同于 $\mathcal{N}(s_\tau;\ \mathring{\mu},\ \mathring{\sigma})$，并使用解析解计算$Q(s_\tau|\pi)$的熵。接下来，$H[Q(o_\tau|s_\tau,\ \pi)]$ 的计算过程如下：

$$o^r_t \xrightarrow{i} \text{Encoder} \xrightarrow{s} \text{Transition} \xrightarrow{s} \text{Decoder} \xrightarrow{m} \mathring{o}^r_{t+1},$$

其中，观测 $\mathring{o}^r_{t+1}$等同于伯努利分布 $Q(o_\tau|s_\tau,\ \pi)$的参数，并使用解析解计算 $H[Q(o_\tau|s_\tau,\ \pi)]$。令人惊讶的是，另一个以完全相同方式采样的观测 $\mathring{o}^r_{t+1}$等同于伯努利分布 $Q(o_\tau|s_\tau,\ \theta,\ \pi)$的参数，并使用相同的解析解计算 $H[Q(o_\tau|s_\tau,\ \theta,\ \pi)]$。最后，将$\mathring{o}^r_{t+1}$反馈至编码器，以获得高斯分布 $Q(s_\tau|o_\tau,\ \pi)$的均值和对数方差，从而计算 $H[Q(s_\tau|o_\tau,\ \pi)]$；并使用高斯分布熵的解析解计算 $H[Q(s_\tau|o_\tau,\ \pi)]$。

总而言之，两个样本 $\mathring{o}^r_{t+1}$（如图 40 所述进行采样）已分别与两个不同分布的参数相对应，即 $Q(o_\tau|s_\tau,\ \theta,\ \pi)$ 和 $Q(o_\tau|s_\tau,\ \pi)$。此外，第三个样本$\tilde{}$取自 $\mathring{o}^r_{t+1}$（采用相同方式进行采样），也被用作分布 $P(o_\tau|\pi)$ 的输入。最后，尽管 Fountas 等人（2020）将期望自由能定义为式（8）所示的形式，但代码将加号变成了减号，从而得到以下期望自由能定义：

$$\begin{aligned} G_\tau(\pi) = &- \mathbb{E}_{\tilde{Q}}\Big[\ln \tilde{P}(o_\tau|\pi)\Big] \\ &- \mathbb{E}_{Q(\theta|\pi)}\Big[\mathbb{E}_{Q(o_\tau|\theta,\pi)}\big[H[Q(s_\tau|o_\tau,\pi)]\big] - H[Q(s_\tau|\pi)]\Big] \\ &+ \mathbb{E}_{Q(\theta|\pi)Q(s_\tau|\theta,\pi)}\Big[H[Q(o_\tau|s_\tau,\theta,\pi)]\Big] - \mathbb{E}_{Q(s_\tau|\pi)}\Big[H\big[Q(o_\tau|s_\tau,\pi)\big]\Big], \end{aligned}$$

其中，红色的减号原本应为加号。

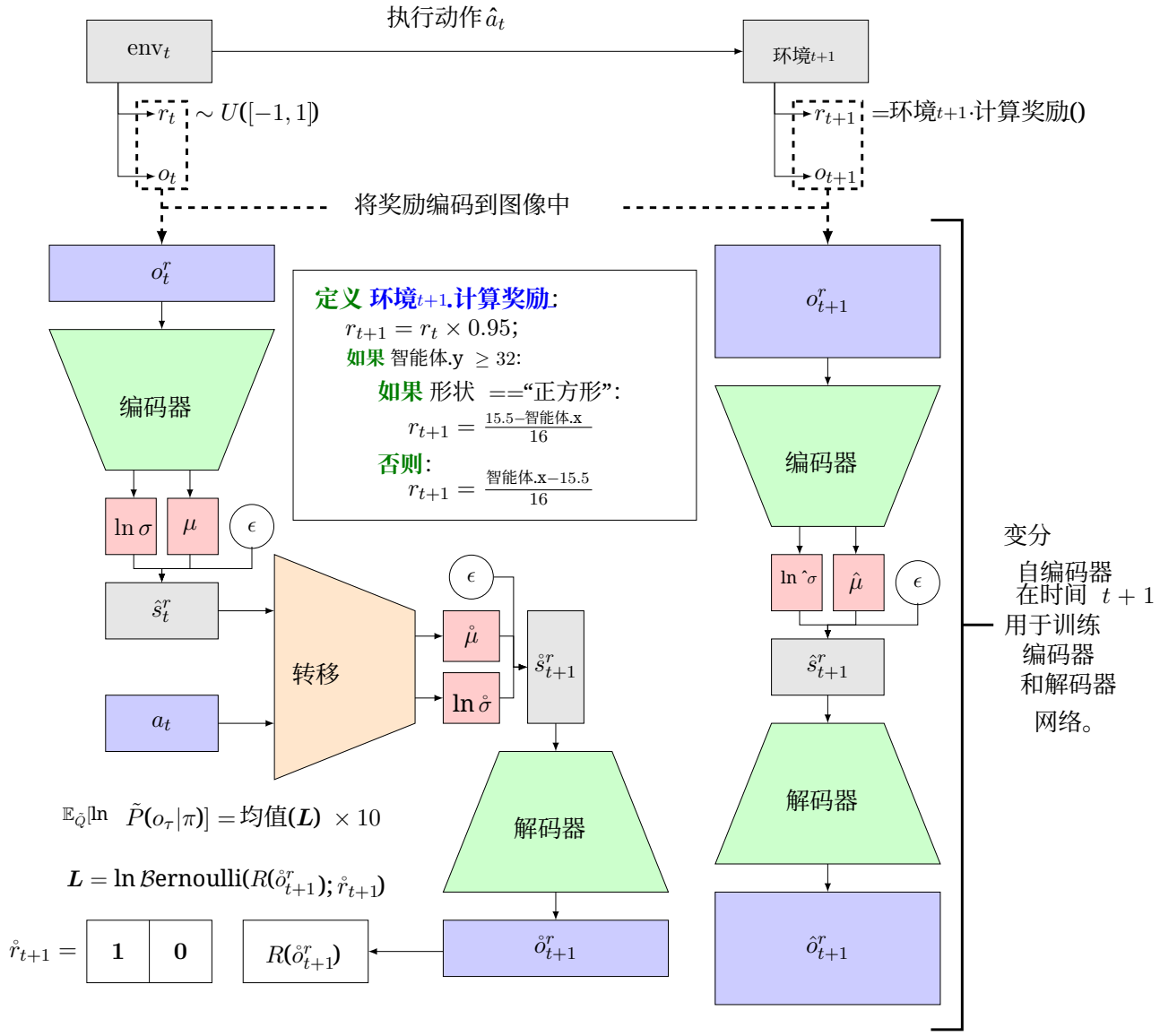


图41：此图展示了 Fountas 等人（2020）的（代码中） $\mathbb{E}_{\tilde{Q}}[\ln P(o_\tau|\pi)]$ 的计算过程。在时间 t，环境向智能体提供一张图像 o_t 以及一个从区间 $[-1$随机采样的奖励 r_t； $1]$。随后，动作 $\hat{a}_t$ 在环境中执行，智能体观测到一张图像 o_{t+1}和一个奖励 r_{t+1}，其中 r_{t+1} 根据图像中心所示的函数计算得出。接下来，时间 t 的奖励和 $t+1$ 分别编码在时间 t 和 $t+1$收到的图像中；有关编码的详情，请参见图 40。然后，将时间 $t+1$ 的编码图像（即 o_{t+1}^r）输入编码器，再使用重参数化技巧从变分后验分布中采样一个状态。此状态输入解码器，解码器尝试重建输入编码器的图像。计算出 $\hat{o}_{t+1}^r$ 后，使用反向传播学习编码器和解码器的权重。另一方面，时间 t 的编码图像（即 o_t^r）用于计算 $\mathbb{E}_{\tilde{Q}}[\ln P(o_\tau|\pi)]$。具体而言，将 o_t^r输入编码器，并从变分后验分布 $Q_{\phi_s}(s_t)$中采样一个状态。随后，将此状态与智能体在时间 t执行的动作 π （即 a_t）一起作为输入，送入转移网络。接着，可以从转移网络预测的分布中采样时间 $t+1$ 的状态。然后将此状态输入解码器，解码器输出 $\mathring{o}_{t+1}^r$。接下来，将一个编码智能体可获得的最大奖励的矩阵（即 $\mathring{r}_{t+1}$）用作伯努利分布的参数，以计算 $\hat{o}_{t+1}^r$前三行的概率的对数（即 $\boldsymbol{L}$），即 $R(\mathring{o}_{t+1}^r)$。随后计算 $\boldsymbol{L}$的均值，并将其乘以十，得到 $\mathbb{E}_{\tilde{Q}}[\ln P(o_\tau|\pi)]$。

附录 C：认知价值分析

在本附录中，我们研究 Parr 和 Friston（2019）公式（10）所给出的期望自由能分解为认知价值和外在价值。之所以特别关注这一表述，是因为它是对深度主动推理方法影响最大的期望自由能版本之一，例如 Fountas 等人（2020）所采用的版本。从期望自由能的定义开始，见正文中的公式（14）：

$$\begin{aligned} G_\tau(\pi) &= \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi) - \ln P(o_\tau, s_\tau)\big] \\ &= \mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln Q(s_\tau|\pi) - \ln P(s_\tau|o_\tau) - \ln P(o_\tau)\big] \\ &= \underbrace{-\mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln P(s_\tau|o_\tau) - \ln Q(s_\tau|\pi)\big]}_{\text{Epistemic value}} \underbrace{-\mathbb{E}_{P(o_\tau|s_\tau)Q(s_\tau|\pi)}\big[\ln P(o_\tau)\big]}_{\text{Extrinsic value}} \end{aligned}$$

在本附录的其余部分，我们将聚焦于认知价值，并报告两个实验。在第一个实验中，状态的先验（参见图 45 最左侧的图）等于近似后验（参见图 42），而似然（参见图 43）变得越来越接近均匀分布（参见图 44）。请注意，如果似然变为均匀分布，那么真实后验 $P(s_\tau|o_\tau)$ 将等于状态的先验 $P(s_\tau)$，即：

$$P(s_\tau|o_\tau) \propto P(o_\tau|s_\tau)P(s_\tau) = \frac{1}{|o_\tau|}P(s_\tau),$$

其中，$|o_\tau|$是时间步 τ的观测数量，并且在重新归一化后，$P(s_\tau|o_\tau) = P(s_\tau)$。图 39最左侧的图表显示，随着似然变得更加均匀分布，认知价值会降低；也就是说，随着真实后验 $P(s_\tau|o_\tau)$与近似后验 $Q(s_\tau|\pi)$变得更加相似，认知价值会降低。这种行为是意料之中的，因为真实后验 $P(s_\tau|o_\tau)$与近似后验 $Q(s_\tau|\pi)$差异越大，认知价值就越高。因此，最大化认知价值将促进探索和信息增益。

在第二个实验中，似然的熵较高（参见图 44最右侧的图表），状态的先验正从左向右移动（参见图 45）。当状态的先验 $P(s_\tau)$与近似后验 $Q(s_\tau|\pi)$不同时，联合分布 $P(o_\tau|s_\tau)Q(s_\tau|\pi)$会比真实后验 $P(s_\tau|o_\tau)$更接近近似后验 $Q(s_\tau|\pi)$。事实上，由于似然几乎均匀分布，真实后验 $P(s_\tau|o_\tau)$几乎等于状态的先验 $P(s_\tau)$。与此同时，由于似然几乎均匀分布，它对联合分布$P(o_\tau|s_\tau)Q(s_\tau|\pi)$不会产生太大影响，而近似后验 $Q(s_\tau|\pi)$将占主导地位。总之，当状态的先验 $P(s_\tau)$与近似后验 $Q(s_\tau|\pi)$不同时：

- 真实后验将几乎等于状态的先验 $P(s_\tau|o_\tau) \approx P(s_\tau)$
- 联合分布　　布 $P(o_\tau|s_\tau)Q(s_\tau|\pi)$ 将比 $P(s_\tau|o_\tau)$ 更接近 $Q(s_\tau|\pi)$

因此，当期望中的差值为负时，联合分布 $P(o_\tau|s_\tau)Q(s_\tau|\pi)$ 往往会较大（参见图 46）。事实上，由于联合分布与近似后验相似，这意味着当近似后验较大而真实后验较小时，联合分布会较大。这表明，真实后验的对数将非常负，而近似后验的对数则不那么负，即更接近零。随后，将近似后验的对数从真实后验的对数中减去，也就是说，将一个较小的正数加到一个非常负的数上。因此，结果将为负。

图 39 最右侧的图显示，随着先验 $P(s_\tau)$ 与近似后验 $Q(s_\tau|\pi)$ 越来越相似，真实后验 $P(o_\tau|s_\tau)$ 也随之越来越相似于近似后验 $Q(s_\tau|\pi)$，认知价值不断增加。但不应出现这种情况，因为真实后验 $P(s_\tau|o_\tau)$ 与近似后验 $Q(s_\tau|\pi)$ 越相似，认知价值就越大。因此，最大化认知价值并不会促进探索。事实上，它会建议将注意力集中在单个动作上，这一点与图33中的观察结果一致。

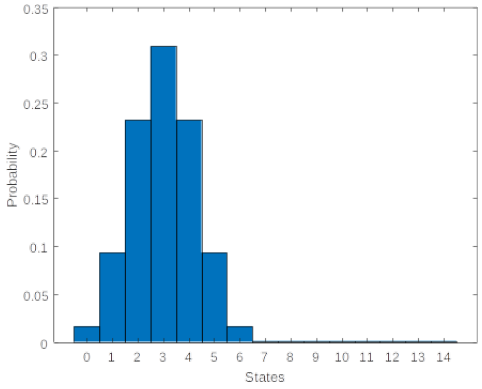


图 42：此图展示了近似后验分布 $Q(s_\tau|\pi)$，它服从二项分布，对应 6 次试验，成功概率为 0.5。

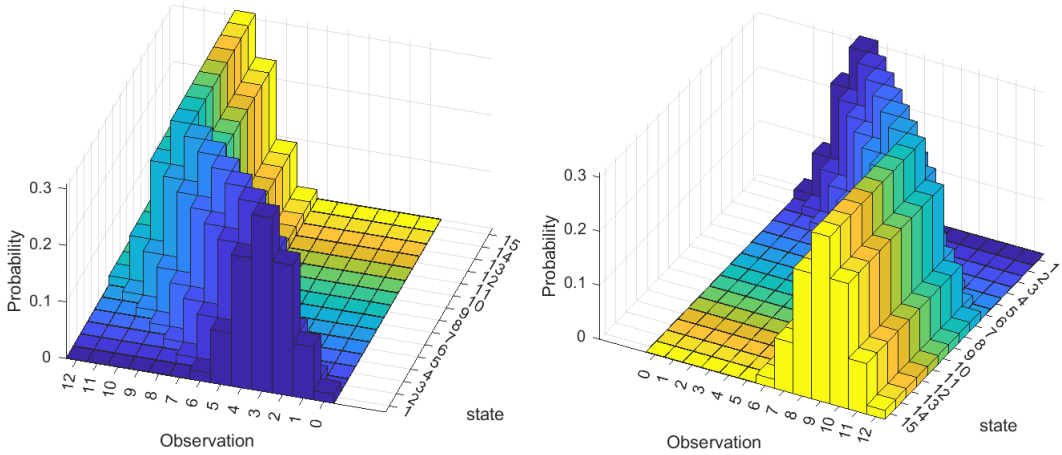


图 43：此图从两个视角展示了同一个似然映射 $P(o_\tau|s_\tau)$，即从前方和后方观察。该似然映射是通过沿观测轴滑动一个二项分布（对应 6 次试验，成功概率为 0.5）创建的，每当状态增加 1，就滑动一次。最后，当二项分布到达最极端的位置后，在剩余状态下保持不变。

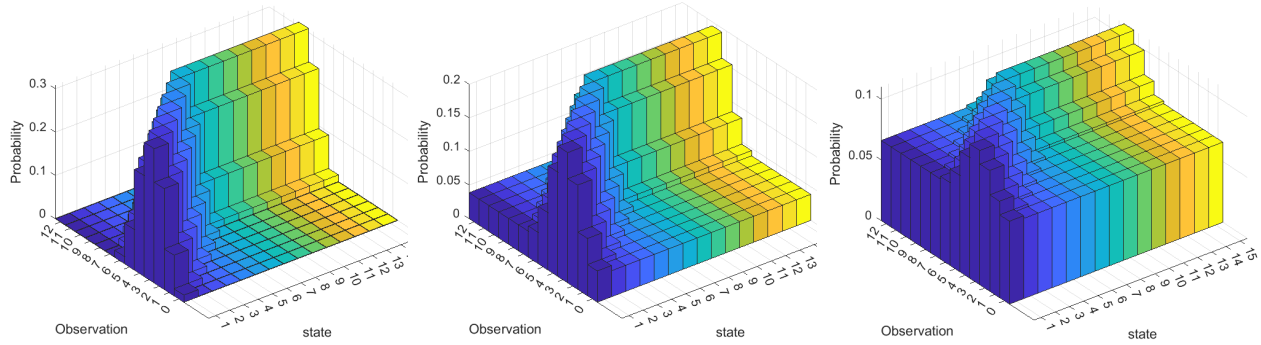


图 44：在第一次实验中，我们通过逐渐使似然映射 $P(o_\tau|s_\tau)$ 变得更加均匀来对其进行改变。图中从向左到向右展示了这一变化。

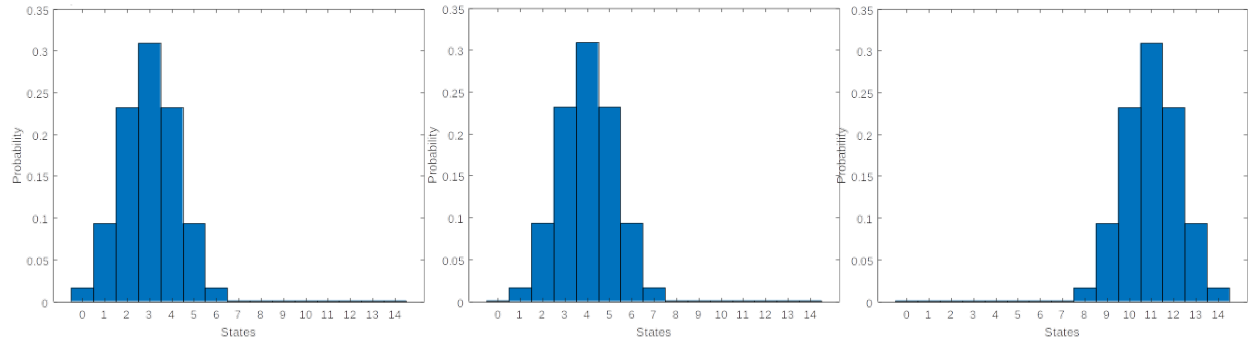


图 45：在第二次实验中，我们通过让状态的先验分布 $P(s_\tau)$ 与近似后验 $Q(s_\tau|\pi)$ 之间的差异越来越大，改变了该先验分布。图中从向左到向右展示了这一变化。

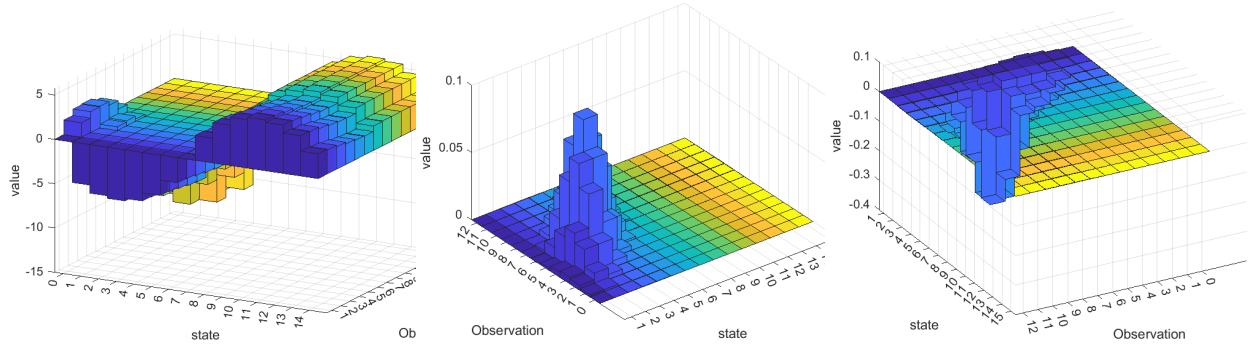


图 46：为了计算认知价值，我们首先计算了真实后验分布的对数 $\ln P(s_\tau|o_\tau)$ 与近似后验的对数 $\ln Q(s_\tau|\pi)$ 之差，针对观测 o_τ 取到的所有值进行计算（参见最左侧的图）。然后，我们计算了用于期望的联合分布 $P(o_\tau|s_\tau)Q(s_\tau|\pi)$（参见中间的图）。接着，我们计算了最左侧图和中间图所示矩阵的逐元素乘积（参见最右侧的图）。最后，将逐元素乘积的所有元素相加，即可得到认知价值。在此例中，认知价值将显著为负。