arXiv:2305.01400v1 [cs.RO] 2 May 2023

回到这里：通过回归分布规划实现鲁棒模仿

GeoffreyCideron Baruch Tabanpour Sebastian Curi Google Research，Brain 团队；Google Research，Brain 团队；ETH

Sertan Girgin LeonardHussenot Gabriel Dulac-Arnold Google Research，Brain 团队；Google Research，Brain 团队；Google Research，Brain 团队

Matthieu Geist OlivierPietquin* **RobertDadashi***Google Research，Brain 团队Google Research，Brain 团队Google Research，Brain 团队

摘要：我们研究这样一种模仿学习（IL）设定：专家数据并非在实际部署环境中收集，而是在一个不同版本的环境中收集。为解决由此产生的分布偏移，我们将行为克隆（BC）与规划器相结合；当智能体偏离演示分布时，规划器负责将其带回专家曾访问过的状态。由此得到的算法 *POIR*可离线训练，并利用在线交互高效地微调其规划器，从而持续提升性能。我们在逼真的机器人操作模拟器中，对多种人工生成的操作示范进行了*POIR*测试，结果表明，学习到的策略能够适应不同的初始状态分布和含噪动力学。

关键词： 模仿学习、行为克隆、模型预测控制

1 引言

模仿学习（IL）是序贯决策中的一种范式，其中智能体利用离线专家轨迹来模仿专家的行为 [1]。强化学习（RL）需要额外的奖励信号，而在实践中往往难以指定；相比之下，模仿学习只需要专家轨迹，通常更易于收集。部分得益于其简单性，模仿学习已成功应用于多种现实世界任务，从机器人操作 [2, 3, 4] 到自动驾驶 [5, 6]。

然而，部署模仿学习的一个关键挑战在于，智能体可能会在最终部署环境中遇到离线专家未标注的状态 [7]。在医疗保健 [8, 9] 和机器人技术 [10, 11]，等应用中，在线实验可能存在风险（例如，在人类患者身上进行），或者标注成本高昂（例如，离策略机器人数据集可能需要数月才能收集完成）。因此，在本研究中，我们关注这样一类方法：它们不仅能够在训练期间未见过的状态下保持稳健，还能够在无法访问可查询专家的情况下进行在线适应。

典型的模仿学习方法，例如行为克隆（BC） [1]和对抗性模仿学习（AIL） [12] ，并非旨在同时实现数据高效，并能够抵御专家数据与最终部署环境之间的分布失配。BC 将 IL 视为一个监督学习问题，在专家的状态分布下最大化采取专家动作的似然 [1]。尽管 BC 能够从离线专家数据中生成有用的策略，但在训练期间未见过的状态下表现不佳 [7]：BC 不仅容易受到分布外（OOD）输入的影响，而且无法访问可查询专家，因此部署后无法适应新状态。相比之下，对抗性模仿学习方法采用逆强化学习（IRL）方法，首先学习一个奖励函数，使专家轨迹数据在该函数下达到最优，然后使用这一学习到的奖励函数训练策略 [13, 12, 14]。由于对抗性模仿学习方法直接在部署环境中训练强化学习策略，因此能够在专家数据中最初未出现的状态下学习策略。然而，对抗性模仿学习方法需要大量在线数据才能达到专家水平，并且

*共同贡献

a通常在实践中很难调优 [15]，而 BC 不使用在线数据，调优起来更简单。
本文提出利用行为克隆的数据效率和离线训练能力，同时发挥离线基于模型的方法的灵活性 [16, 17, 18]，使策略能够应对分布外状态，并在部署后对最终策略进行微调。*POIR*（基于离线模仿奖励的规划）是一种基于模型的模仿学习算法，可利用专家演示离线数据集（即无需任何奖励）学习鲁棒策略，并在部署后继续利用从环境中收集的数据提升性能。通过在规划过程中使用显式的模仿奖励，*POIR* 能够在进入专家支持集之外的状态时，返回专家数据中出现过的状态。

POIR 可在离线和在线设置下部署。在离线设置下，行为克隆策略和规划器使用的世界模型仅基于离线专家转移数据进行训练。在在线设置下，世界模型会利用规划器策略在部署环境中收集的新转移数据进行微调，从而提高其对离线转移数据未覆盖的部分状态空间的预测准确性；进而提升规划器策略在相对于离线专家数据而言的分布外状态下的表现。

我们这项工作的贡献有两点：首先，我们提出了*POIR*，一种基于模型的模仿学习方法，能够抵御初始状态分布扰动和部署环境中的随机性。我们在真实的 Robosuite [19] 模拟器上，通过一系列复杂的人类生成的机器人操作示范，展示了该方法相较于行为克隆和对抗性模仿学习方法的性能。其次，我们表明，*POIR* 可以像行为克隆一样进行离线训练和部署；上线后，它还能以远少于对抗性模仿学习方法所需的数据持续提升性能，同时避免强化学习训练算法和对抗损失带来的复杂性。

2 方法

我们将环境建模为回合式马尔可夫决策过程（MDP）(S, A, P, r, ρ_0, T) [20]，其中 S 表示状态空间，A表示动作空间，P 表示转移核，r 表示奖励函数，ρ_0 表示初始状态分布，T 表示任务时域。策略 π 是一个将状态映射为动作分布的函数；我们将所有策略构成的空间记为 Π。令 ρ_π 表示分布。

由策略 π 诱导的状态和动作上的。在模仿学习中，我们不知道真实奖励r。相反，我们会基于专家策略 π_e 诱导的专家演示 $D_e = \{(s_t, a_t, s_{t+1})_{t \in T}\}$，在离线环境下构建模仿奖励 r_{IL}。

模仿学习的目标是找到一个策略 π，使其诱导的状态-动作分布 ρ_π与专家的状态-动作分布 ρ_{π_E} [21]之间的某种散度 D 最小。换言之，目标是最小化：$\min_\pi D(\rho_\pi, \rho_{\pi_E})$。例如，GAIL [12] 的目标是最小化詹森-香农散度的一种正则化形式。

接下来，我们介绍*POIR*：一种基于模型的模仿学习算法，它利用人类专家演示的离线数据集进行稳健的模仿学习，并通过环境中收集的在线数据进一步改进。*POIR* 由三个主要部分组成：行为克隆策略先验、规划器和模仿奖励。

2.1 通用方法

我们的一般方法将行为克隆固有的优势与规划算法相结合，使其能够生成鲁棒动作并高效地在线适应。具体而言，我们将行为克隆策略用作先验，引导规划器中的轨迹展开。规划器利用学习得到的世界模型生成轨迹，然后根据奖励函数对动作进行排序并选择最佳动作；我们利用模仿奖励，促使智能体始终接近专家支持集。系统的所有组件最初均在专家轨迹数据集上训练，轨迹 $D_e = \{(s_t, a_t, s_{t+1})_{t \in T}\}$。关于 *POIR* 的三个核心组件（行为克隆先验策略、规划器和模仿奖励）的详细信息将在以下章节中介绍，并在图 1中进行说明。*POIR*的概览见算法 1。

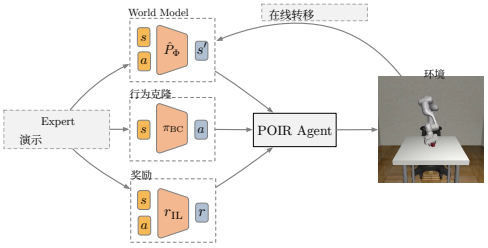


图 1：各类*POIR* 组件及其依赖关系概览。行为克隆策略和世界模型由专家数据训练而成，而模仿奖励则可以依据专家数据、世界模型的集成差异，或行为克隆策略的集成差异。

2.2 行为克隆策略先验

POIR 是一种基于规划的方法，它利用行为克隆作为先验来引导规划器 rollout。我们工作的一个关键方面是，行为克隆策略先验和世界模型（第 2.3 节）都采用自助集成实现 [22, 23, 24]。实际上，我们的行为克隆策略先验是由 K 个神经网络组成的集成模型，并按照第 2.3 节所述，在规划器中以轮询方式使用。不过，将行为克隆用作基线时，我们选择对 K 个神经网络的输出取平均值，以生成最终的动作输出，并将其称为集成行为克隆（EBC）。因此，我们的集成行为克隆策略形式如下：$\pi_{BC} = \frac{1}{K}\sum_1^K \pi_{\theta_k}(s)$，与单网络行为克隆相比，它显著提升了性能（见图3），此前的研究也观察到了这一点 [25]。我们使用完整的专家演示集训练 EBC，以用于一个

g给定任务，使动作预测误差最小化：$\theta_k = \text{argmin}_{\theta_k} \mathbb{E}_{D_e}[(\pi_{\theta_k}(s_t) - a_t)^2]$.

2.3 规划器

我们算法方法的核心是利用规划器回到专家的状态分布。*POIR* 使用模型预测控制（MPC） [26]和基于 MPPI 的 [27] 轨迹优化器，并按照 [16]中的详细说明进行了修改，以支持集成模型和策略先验。直观而言，当策略相对于专家演示已经处于分布内时，规划器无法提供太多额外价值。然而，当策略严重偏离分布，或决定采取会导致其脱离该分布的动作时，规划器将提供更接近演示分布的候选动作轨迹。下面将进一步介绍所学习的世界模型和轨迹优化器。

世界模型 基于规划的方法的一个核心组成部分是底层世界模型。ˆ 我们使用一个由 ϕ参数化的确定性函数 $P_\phi : S \times A \to S$，该函数接收状态-动作对并输出下一状态。该函数在专家演示数据集 D_e上进行学习：$\text{argmin}_\phi \mathbb{E}_{(s_t,a_t,s_{t+1})\in D_e}[(\hat{P}_\phi(s_t, a_t) - s_{t+1})^2]$。如前所述，世界模型是一个ˆ自举集成模型；学习得到 K 个世界模型，其中每个 P_k 都在同一数据集上训练，但采用不同的权重初始化。对于确定性环境，已有研究表明，选择确定性模型还是随机模型对性能影响不大 [28]。尽管某些实验环境具有随机性，但示范数据本身是确定性的，而确定性模型在实践中已被证明适用于微调。

规划器轨迹 如上所述，我们的策略采用模型预测控制（MPC） [26], 这一简单而有效的控制策略 [29, 23]。在每个策略步骤中，MPC 都会调用轨迹优化器，为当前状态 s_t寻找一条最优轨迹 $A_H = \{a_1, \ldots, a_H\}$，该轨迹根据世界模型 P的动力学使某个目标函数 r 最大化。随后，MPC 将轨迹中的第一个动作 a_1应用于系统，并观测相应的下一状态s_{t+1}，算法再以此重复执行。*POIR* 采用了 MPPI 的一个修改版本 [30, 23, 16]，下面将详细介绍。

POIR 的轨迹优化方案包括生成 N 条并行轨迹，这些轨迹从由 π_{BC}生成的提议分布中采样。对于每条轨迹，采样一个动作，并加入高斯噪声进行扰动 $\sim a_t = \{\pi_{BC}(s_t) + \epsilon_i\}_{i=1}^N$，其中 $\epsilon_i \sim \mathcal{N}$为（0，$\sigma^2$）。随后，将扰动后的动作 $\tilde{a}_t$ 输入世界模型，以生成后续的$s_{t+1} = P(s_t, \tilde{a}_t)$。重复此过程 H次，以生成完整的动作轨迹 $A_H^n = \{\tilde{a}_1, \ldots, \tilde{a}_H\}$，并行处理这 N 条轨迹。同时计算相应的累积奖励，为每条 N 轨迹生成 $R^n = \sum_{h=1}^H r_{IL}(s_t^n, \tilde{a}_t^n)$。

按照模型预测控制的要求，最终返回的动作是具有最高奖励的轨迹中的第一个动作：$i_{max} = \text{argmax}_n R^n$ 奖励最高：$\pi_{POIR}(s_t) = A_{i_{max}}[0]$。完整算法详见算法2。

ˆ**集成模型的使用** 如前所述，策略先验 π_{BC} 和世界模型 P实际上都由 K 集成模型组成。我们通过将每条 N 轨迹分配给一个集成头来利用这些集成模型；策略先验和世界模型均采用这种方式。该分配在整个展开过程中保持不变，具体做法由 [23]和 [16], 执行，并使动作选择和模型预测都更加多样化。为简化符号，我们未在算法2中明确描述这一点；完整算法见附录中的算法3。

2.4 模仿奖励

POIR 的最后一个组成部分是模仿奖励，r_{IL}。在以下各节中，我们将构造若干奖励函数，其标准是：对于接近专家支持集的状态，奖励应较高；反之则应较低。对于我们提出的每种奖励，接近的概念各不相同。

L2奖励 构造此类奖励函数的一种简单方法，是计算当前状态与示范数据中所有状态之间的L2距离，然后取这些距离中的最小值。此奖励函数不需要预训练。对于状态 s，我们将L2奖励记为： $r_{L2}(s) = -\min_{s_e \in D_e} ||s - s_e||_2$。

对于较大的数据集，计算L2奖励可能会非常耗费计算资源，因为我们需要计算与专家数据中所有状态的距离。近似查找方法可以将复杂度从$O(n)$ 降低至 $O(log(n))$ [31, 32]。尽管L2奖励很简单，但先前的研究已证明它在机械臂操作任务中效果显著 [33]，并且可以通过学习嵌入扩展到更高维的状态空间 [33]。采用该奖励函数的*POIR* 变体称为 *POIR*-L2。

集成分歧奖励 r_{IL} 的另一种形式是考察在示范数据上训练的网络集成的预测分歧。两种知名方法分别利用行为克隆策略的分歧（由DRIL提出） [34], , 以及世界模型的分歧（由MoREL提出） [17]。前一种变体称为*POIR*-DRIL，后一种称为*POIR*-MOREL。与*POIR*-L2相比，这些方法能够很好地适应专家数据集规模和状态维度，因为在推理时只需进行一次前向传播，而无需对 D_e进行穷举搜索。奖励可表示为 $r_{DRIL}(s) = \hat{} - \max_{i,\ j} ||\pi_{\theta_i}(s) - \pi_{\theta_j}(s)||_2$ 以及 $r_{MOREL}(s,\ a) = -\max_{i,\ j} ||P_{\phi_i}(s,\ a) - P_{\phi_j}(s,\ a)||_2$。

算法1*POIR*

1: 输入： D_e：专家轨迹， L_{steps}：环境步数， H：规划时域， N：轨迹数量， σ：策略先验噪声尺度。 ˆ 2: π_{BC}, P, r_{IL} = 训练(D_e) 3: $D_a = \{\}$ 4: 对 $t = 1$ 到 L_{steps} 执行 5: 观察 6 a_t = ：选择动作(s_t, π_{BC}, ˆ $P, r_{IL}, H, N, \sigma, k$) 7: s_{t+1} = 环境步进 (a_t) 8: $D_a = D_a \cup \{(s_t, a_t, s_{t+1})\}$ˆ 9: 在 $D_e \cup D_a$ 上训练 P 10: 结束循环

算法 2*POIR*-选择动作

1: **选择动作**(s, π_{BC}, $\hat{P}, r_{IL}, H, N, \sigma$):
2: 设置$\mathbf{R} = \vec{0}_N$， $\mathbf{A} = \vec{0}_{N,H}$
3: **对于** $n = 1..N$ **执行**
4: $s_1 = s$
5: **对于** $t = 1..H$ **执行**
6: $\epsilon \sim \mathcal{N}(0,\ \sigma^2)$
7: $\tilde{a}_t = \pi_{BC}(s_t) + \epsilon$
8: $s_{t+1} = \hat{P}(s_t,\ \tilde{a}_t)$
9: $\mathbf{A}_{n,t} = \tilde{a}_t$
10: $R_n = R_n + r_{IL}((s_t, \tilde{a}_t, s_{t+1}))$
11: **结束循环**
12: **结束循环**
13: **返回** $a_1 = A_{\text{argmax } R,\ 1}$

2.5 在线微调

将 POIR 部署到环境后，我们可以对其进行在线微调。主要目标是使用与系统交互所获得的额外数据，对 P 进行微调。我们只需将来自智能体 D_a 的数据添加到原始数据集 D_e 中，并使用 $D = D_e \cup D_a$ 对 P 进行训练即可。在模仿奖励对 P 使用集成差异的情况下，我们会保留一个冻结版本 P，该版本仅使用 D_e 进行训练，以便计算模仿奖励。

3 实验

为探究 *POIR* 在分布失配情境下的优势，我们使用一系列真实机器人操作任务上的人类演示来训练 *POIR* ，并评估 *POIR*在原始环境的各种物理扰动版本中的表现。

3.1 环境与演示

环境我们研究了*POIR*在 Robosuite 仿真环境 [19] 中定义的一系列机器人操作任务上的表现。我们选择这些环境，是因为它们兼具几项关键特性：任务复杂、可获取大量由人类生成的演示，并且环境作者已证明，仿真器表现与机器人实机性能之间具有较强相关性 [35]。我们使用 7自由度Panda机械臂模型，聚焦于难度逐渐增加的三项任务：**Lift**：抓取方块并将其从桌子上抬起（10 维观测空间）；**PickPlaceCan**：抓取易拉罐并将其放入箱中（14 维观测空间）；**NutAssemblySquare**：抓取方形螺母并将其放置在杆状夹具上（14 维观测空间）。这三项任务都具有确定性，并且默认包含两个变化来源：

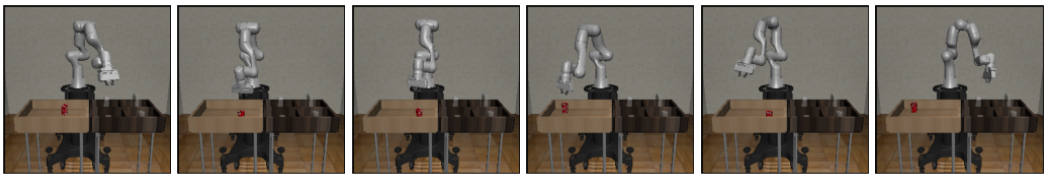


图 2：对于 $\sigma_{noise} = 0.4$ 所示的噪声，与默认噪声 $\sigma_{noise} = 0.2$相比，机械臂可以从距离桌子中心远得多的位置开始。所有环境的示例见附录 E。

min机械臂的初始位置以及 d
the 要抓取的物体的初始位置。

演示 我们使用来自 Robomimic [35]数据集的开源人工生成演示。该数据集是一个经过充分研究的人工生成演示数据集，有充分证据表明，在 Robosuite 模拟器中表现良好的方法，在相同任务的真实机器人版本中同样表现良好。对于每项任务，我们使用熟练人类操作员的 演示，这些演示由一名熟练的人类操作员使用 RoboTurk [36]收集，共 200 条。

3.2 分布失配

演示是在未施加扰动的环境版本中收集的，我们将其称为离线环境。我们在部署环境中评估*POIR* 的性能，这些环境在两个方面存在差异：初始状态分布的多样性，以及环境转移动力学中的随机性。

初始状态分布。 对于每项任务（Lift、PickPlaceCan 和 NutAssemblySquare），我们会修改部署环境中机器人手臂的初始状态分布。默认情况下，每个回合开始时，机械臂的初始位置会在一个均值位置附近随机变化，其高斯噪声的标准差为 $\sigma_{init} = 0.02$。默认噪声使机械臂靠近均值位置，如图 7所示。离线人类演示是在这种默认噪声下收集的。在实验中，我们逐步将部署环境中的 σ_{init} 从 0。02 增大至0。4。当 σ_{init} 值较大时，如图2所示，机器人手臂的初始位置可能位于桌子任一侧，而不再是桌子正上方的中央位置。其整体效果见附录 E及随附视频。对初始状态噪声的修改会导致演示中的默认初始位置与部署环境中的初始位置之间出现分布失配。

随机性。 我们在原本确定性的 Robosuite 环境中引入随机性。现实世界中，控制器或部署环境的自然变化都可能带来环境随机性，例如摩擦、光照和物体变形。我们使用标准差为 σ_{action} 的白高斯噪声，作为这些外部扰动的简单模型。每一步都会将高斯噪声添加到智能体执行的动作中。因此，环境变得具有随机性，因为两个智能体即使在相同状态下执行相同动作，也可能到达不同的下一状态。

3.3 实现细节

所有算法每经过 20 个回合 50，000 环境步数500，000步时评估一次，总计进行 5 个随机种子。我们报告平均环境成功率 $\in [0, 1]$ ，表示任务是否成功完成。对于每种算法，在所有环境中均使用相同的一组超参数。完整的超参数列表见附录C。

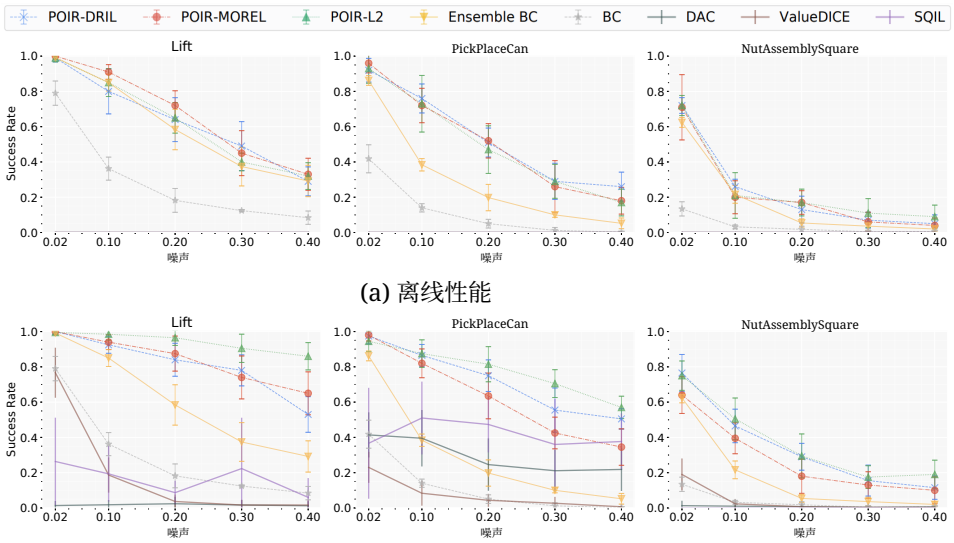
由于*POIR*在规划器中使用了行为克隆策略集成，因此行为克隆和 EBC 自然成为基线方法。我们对行为克隆、EBC 以及*POIR*中使用的行为克隆策略采用相同的网络架构（多层感知机）。对于*POIR* 和 EBC，我们使用 27 个集成网络。出于实现方面的考虑，集成网络的数量取 3 的倍数；我们发现数量超过 27 后性能没有提升。在规划器中，我们采样 $N = 4000$ 条轨迹。对于行为克隆动作提议周围的标准差噪声，我们使用 $\sigma = 0.2$ 。规划器轨迹的时域设为 $H = 5$，因为在初步超参数搜索中该设置表现最佳。

We 此外，我们还将算法与最先进的模仿学习算法进行比较：判别器演员 -

评论家（DAC） [15] 按照 [37], ValueDice [38], 和 SQIL [14]的指南进行调优。DAC、ValueDice 和 SQIL 是*POIR* 的自然基线方法，因为它们利用专家演示和从环境中收集的在线数据来学习最终策略。不过，DAC 和 SQIL 与*POIR* 还有一点不同：它们无法完全离线使用。

我们发现，状态归一化和动作归一化有助于*POIR*，但对行为克隆或 EBC 没有帮助。对于 DAC，我们发现状态归一化效果最佳。我们对离线专家演示进行归一化，使状态和动作的均值为 0、标准差为 1。在线转移则使用离线数据的均值和标准差参数进行归一化。

3.4 结果：初始状态分布不匹配的影响



(a) 离线性能

(b) 在线性能（经过 500,000 个环境步数后）

图 3： σ_{init} 对离线和在线场景下任务成功率的影响。

离线性能。 在完全离线的设置中，我们观察到 *POIR* 在各种环境和噪声水平下的性能均最佳。离线设置的结果见图 3 的上排。例如，在 PickPlaceCan 中，没有分布不匹配时，集成行为克隆和 *POIR*的性能几乎相同，但 *POIR* 的成功率提高了 $20\% - 30\%$，适用于 $\sigma_{init} = 0.1$。这表明，模仿奖励有助于规划器在演示中未见过的状态下找到比先验行为克隆策略提议更好的动作。

在线性能。 *POIR* 能够有效利用在线数据提升性能，如图 3 下排所示。经过 500,000 个环境步数的在线微调后，*POIR*-L2 的表现优于其他模仿奖励方法以及行为克隆/模仿学习基线方法。例如，在噪声值最大时，Lift 和 PickPlaceCan 任务中 *POIR*-L2 与 EBC 的成功率相差 50%（$\sigma_{init} = 0.4$）。这表明，利用收集到的数据改进规划器能够提升性能。这是 EBC 与 *POIR* 的一项关键区别，因为 EBC 无法利用这些新数据。引入一定程度的分布不匹配后，基于强化学习的方法（DAC、ValueDICE 和 SQIL）表现不佳；但在 PickPlaceCan 任务中，DAC 除外，其在 $\sigma_{init} = 0.4$ 时达到 20% 的成功率，这比 *POIR*-L2 在该配置下接近 60% 的成功率低三倍。

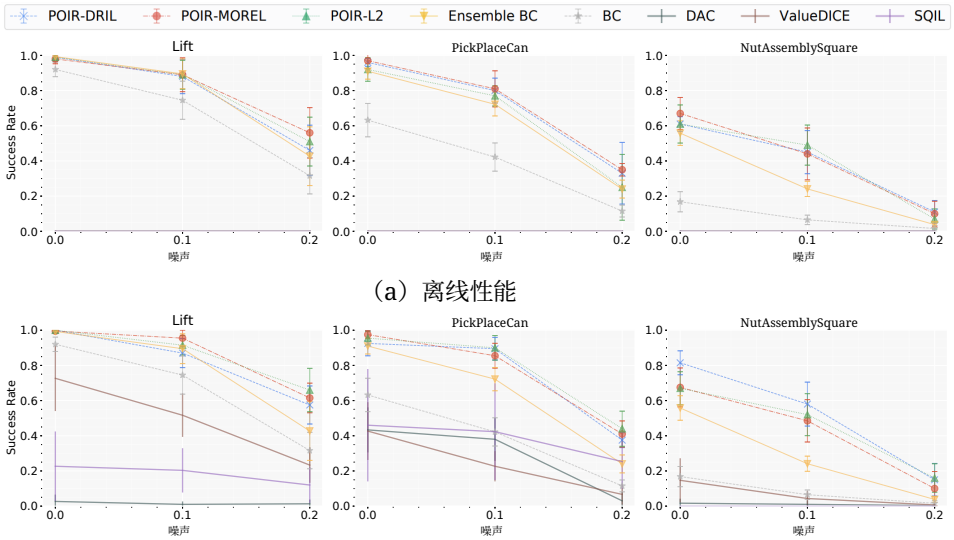
样本效率。POIR的主要优势之一是，其离线表现可与 *BC* 和 *EBC* 基线方法相媲美；此外，在环境中额外进行500,*000* 步微调后，*POIR* 的表现也能超过 DAC 这类最先进的模仿学习方法。相比之下，基于强化学习的方法（DAC、ValueDICE 和 SQIL）在 Lift 任务中经过 500,000 个环境步数后，仍无法达到 BC 的表现；而在 Nu- tAssemblySquare 任务中，经过 $5M$ 步后，成功率仍为0 。附录中的图5 展示了经过 $5M$个环境步数后的训练曲线。

涌现式重试。从定性结果来看，我们发现行为克隆智能体会朝目标物体移动，尝试抓取它，然后无论是否成功抓取，都会继续表现得仿佛已经抓住了物体。与此相反，我们在实验中发现，*POIR* 即使最初的抓取尝试失败，也能够再次尝试抓取物体。我们在随附的视频文件以及附录*F*中展示了POIR的涌现式重试。

实时控制。先前的研究表明，基于 MPPI 的方法适用于实时控制 [30, 23]。POIR 在 MPPI 基础上的额外开销包括使用行为克隆先验，以及计算奖励函数；这两项分别通过对神经网络集成进行一次前向传播来完成（使用 L2奖励时还需额外进行一次查表）。我们通过实验发现，POIR-DRIL 和 POIR-MoREL 在 TPUv2 GCP机器上使用加速器的单个核心进行推理时，运行频率为 10 Hz，而 POIR-L2 的运行频率为 5 Hz。我们认为，这表明该方法可用于实时控制。

消融实验。为展示 *POIR*各组成部分的重要性，我们还运行了另外两种基线方法：不使用行为克隆先验的 POIR，以及 [39] 使用 DRIL奖励的 PPO；后者本质上就是 [34]不带辅助行为克隆损失的 DRIL算法。结果详见附录 B。这两种基线方法在所有实验中的成功率均为 0%。这些结果表明，行为克隆先验和规划器/模仿学习奖励组合都很重要。

3.5 结果：环境随机性不匹配的影响



（a）离线性能

（b）经过 500,000 个环境步数后的在线性能

图 4：σ_{action} 对离线和在线场景下任务成功率的影响。

如图4所示，*POIR* 在加入动作噪声随机性后，性能优于其他基线方法。我们还注意到，在线数据能够提升训练得到的策略。值得注意的是，在 PickPlaceCan 中，当动作噪声为 0.1 时，集成行为克隆与 *POIR*-L2 之间的成功率差距为 25% – 35%。对于 NutAssemblySquare，在没有动作噪声时，*POIR*-L2 与集成行为克隆之间的成功率差距从 0%变为在动作噪声为 15% 时近乎 0.1 的成功率提升。对于 Lift 和 NutAssemblySquare，DAC 在前 500,000 个环境步数内无法学到良好的策略。如图 6所示，即使经过 $5M$ 个环境步数，基于强化学习的方法在几乎所有设置下的表现仍然不如 EBC。

4 相关工作与局限

模仿学习会受到专家数据与策略 [1] 之间分布偏移的影响。 [40] 将这一概念形式化，并证明行为克隆在 T 中具有二次遗憾界。许多模仿学习方法旨在使策略对分布偏移更具鲁棒性 [7, 41, 13]。

具有可查询专家策略的模仿学习： [7] 提出了 DAGGER，在在线示范中查询专家策略，以改进行为克隆策略，并在 T 中实现线性遗憾。近期研究利用

策略不确定性来选择性地决定何时查询专家 [42, 43, 44, 45]。我们提出的模仿学习方法不需要可查询专家，也不需要明确获取在线示范来缓解分布偏移。

对抗性模仿学习：对抗性模仿学习方法通过学习一个能够区分专家转移与非专家转移的判别器，力求使策略的状态-动作分布与固定的专家数据相匹配 [12, 15]。无模型强化学习策略利用判别器作为奖励，在在线环境轨迹上进行训练。MGAIL [46] 使用额外的世界模型，在完整轨迹上基于判别器的精确梯度训练策略。对抗性模仿学习方法会生成一种在实践中难以调优、且在线数据采样效率不高的无模型控制器 [12]。我们在第3.4节中将方法与 DAC（一种最先进的对抗性模仿学习方法）进行比较。与 DAC 不同，我们的方法可离线部署，且在线采样效率更高。

带支持的模仿学习：近期的模仿学习方法会显式地利用策略与专家之间的分布失配，将其作为训练无模型控制器的奖励信号，而无需使用判别器[33, 34, 47, 48, 49]。这些方法还需要在线收集数据来训练强化学习策略，其采样效率与 DAC 相近。

离线强化学习：许多离线强化学习方法利用不确定性估计，使最终策略能够应对分布偏移。基于模型的离线强化学习方法利用世界模型或网络集成中的不确定性，使最终策略与专家保持接近 [18, 17, 50, 51, 16, 52]。离线强化学习与模仿学习设置不同，因为它假设可以获取环境奖励，而模仿学习中环境奖励未知。离线强化学习方法无法直接处理 POIR 使用的模仿学习奖励。例如，在演示数据集上，L2奖励恰好为 0。

基于模型的模仿学习（MBIL）：在其他场景中，人们已探索在模仿学习中使用世界模型 [53, 54]。IMPLANT [55] 使用 GAIL，利用在线数据学习策略和值函数，随后将其用于规划器策略。IMPLANT 表明，其规划器在简单的运动环境（Hopper、HalfCheetah 和 Walker2d）中的有噪声部署环境下具有鲁棒性；而我们的方法主要不同之处在于使用行为克隆而非 GAIL。这一区别使 *POIR* 能够离线部署，而 IMPLANT 则不能。Imitative Models [56] 学习专家数据上的密度模型，并将该密度模型用作目标条件开放环规划器中的奖励；研究者在自动驾驶模拟器环境中展示了对噪声具有鲁棒性的策略。后续工作 Robust Imitative Planning（RIP） [57]，使用网络集成，降低规划器中轨迹的不确定性，并通过查询专家策略 [58]在线调整策略。类似的方法 RMBIL [59] 在与专家分布不匹配的环境中部署时，也显示出优于行为克隆的效果。RMBIL 是一种纯离线方法，使用神经常微分方程控制器和条件变分自编码器来编码状态。与 Imitative Models、RIP、RMBIL 及其他类似工作 [60]，不同，我们的方法采用更简单的策略：结合开放环轨迹规划算法 [16] 的模型预测控制 [61]，并使用行为克隆先验；值得注意的是，我们表明*POIR* 既能在离线设置下运行，也能在线运行，且无需查询专家。

模仿学习/强化学习中的泛化：模仿学习和强化学习中的泛化旨在提升部署期间面对未见过的交互时的表现 [62]。近期的零样本和少样本模仿学习方法旨在适应新任务、物体和领域，采用的方法包括目标条件化 [63, 64, 65],元学习 [66, 67], 领域随机化 [68, 69, 70]。强化学习中也已展现出泛化能力的提升 [71, 72]。我们的方法探索的是这样一种情形：测试环境相对于专家数据呈分布外状态，但任务或物体并非新的。我们通过基于模型的控制器，在分布外测试环境中取得了优于行为克隆的表现，而非依赖目标条件化或数据增强。

局限性：我们预计该方法存在三项局限性：按照论文中的实现方式，我们在实验中使用的 L2奖励无法直接扩展到大型数据集；该方法在图像等高维状态空间中可能表现不佳；实验结果也未在实体机器人上验证。针对扩展到大型数据集这一问题，已有方法可以缓解这一局限，例如近似最近邻查询 [31, 32]。为处理像素等高维观测，可以采用循环状态空间模型 [73, 74] 或其他嵌入方法 [33] ，将观测嵌入潜在空间，从而生成更紧凑的表示。最后，尽管我们没有在真实机器人上测试该方法，但已有研究表明，在 Robosuite 模拟器上评估学习方法的结果与真实世界的机器人实验具有良好相关性，尤其是使用 robomimic 数据集 [35] 时，正如本文所做的那样。

5 结论

总之，我们提出了*POIR*，这是一种将行为克隆与规划器相结合的模仿学习算法。行为克隆策略为规划器生成候选动作，而规划器则选出最佳动作，驱动智能体返回专家支持集所覆盖的状态。与现有方法不同，*POIR*使用基于模型的控制器，可在离线或在线设置下部署，且无需查询专家。我们展示了该方法完全离线学习的能力，以及显著优于常见对抗性模仿学习方法的下游样本效率。我们还证明，与对抗式模仿学习/行为克隆相比，*POIR* 生成的动作能够更稳健地应对部署环境中的初始状态分布失配和随机动作。最后，我们通过实验证明，使用在线数据训练世界模型能够进一步提升策略性能，并增强其应对分布失配的鲁棒性。

参考文献

[1] D. A. Pomerleau. 面向自主导航的人工神经网络高效训练。神经计算，3(1):88–97，1991年。[2] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. 利用虚拟现实遥操作对复杂操作任务进行深度模仿学习。载于*2018*年 *IEEE*机器人与自动化国际会议（*ICRA*），第5628–5635页。IEEE，2018年。[3] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, 等。传输网络：为机器人操作重排视觉世界。*arXiv*预印本 *arXiv:2010.14406*，2020年。[4] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. 隐式行为克隆。载于机器人学习会议，第158–168页。PMLR，2022年。[5] D. A. Pomerleau. Alvinn：一种基于神经网络的自主陆地车辆。技术报告，CMU，1989年。[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, 等。自动驾驶汽车的端到端学习。*arXiv*预印本 *arXiv:1604.07316*，2016年。[7] S. Ross, G. Gordon, and D. Bagnell. 将模仿学习和结构化预测转化为无悔在线学习。载于第十四届国际人工智能与统计学会议论文集，第627–635页。JMLR研讨会与会议论文集，2011年。[8] P. S. Thomas, B. C. da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. 防止智能机器的不良行为。《科学》，366(6468):999–1004，2019年。[9] O. Gottesman, F. Johansson, J. Meier, J. Dent, D. Lee, S. Srinivasan, L. Zhang, Y. Ding, D. Wihl, X. Peng, 等。在观察性健康场景中评估强化学习算法。*arXiv*预印本 *arXiv:1805.12298*，2018年。[10] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, 等。利用仿真和领域自适应提升深度机器人抓取的效率。载于*2018*年 *IEEE*机器人与自动化国际会议（*ICRA*），第4243–4250页。IEEE，2018年。[11] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, 等。Qt-opt：用于视觉机器人操作的可扩展深度强化学习。*arXiv*预印本 *arXiv:1806.10293*，2018年。[12] J. Ho and S. Ermon. 生成对抗模仿学习。神经信息处理系统进展，29:4565–4573，2016年。[13] P. Abbeel and A. Y. Ng. 通过逆强化学习实现学徒学习。载于第二十一届国际机器学习会议论文集，第1页，2004年。[14] S. Reddy, A. D. Dragan, and S. Levine. SQIL：通过稀疏奖励的强化学习实现模仿学习。*arXiv*预印本 *arXiv:1905.11108*，2019年。[15] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. 判别器－演员－评论家：解决对抗性模仿学习中的样本利用效率低和奖励偏差问题。*arXiv*预印本 *arXiv:1809.02925*，2019年。[16] A. Argenson and G. Dulac-Arnold. 基于模型的离线规划。载于国际学习表征会议，2020年。[17] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. MOReL：基于模型的离线强化学习。载于*NeurIPS*，2020年。

[18] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. MOPO：基于模型的离线策略优化。《神经信息处理系统进展》，33:14129–14142，2020年。[19] Y. Zhu, J. Wong, A. Mandlekar, and R. Mart´ín-Mart´ín. Robosuite：面向机器人学习的模块化仿真框架与基准测试。载于 arXiv预印本 arXiv:2009.12293，2020年。[20] R. S. Sutton and A. G. Barto. 强化学习：导论。麻省理工学院出版社，2018年。[21] S. K. S. Ghasemipour, R. Zemel, and S. Gu. 从散度最小化视角看模仿学习方法。载于机器人学习会议，第1259–1277页。PMLR，2020年。[22] B. Lakshminarayanan, A. Pritzel, and C. Blundell. 使用深度集成进行简单且可扩展的预测不确定性估计。载于神经信息处理系统进展，第6402–6413页，2017年。[23] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. 用于学习灵巧操作的深度动力学模型。载于机器人学习会议，第1101–1112页，2020年。[24] K. Chua, R. Calandra, R. McAllister, and S. Levine. 利用概率动力学模型，通过少量试验实现深度强化学习。载于神经信息处理系统进展，第4754–4765页，2018年。[25] F. Sasaki and R. Yamashina. 从含噪演示中进行行为克隆。载于国际学习表征会议，2020年。[26] J. Rault, A. Richalet, J. Testud, and J. Papon. 模型预测启发式控制：在工业过程中的应用。Automatica，14(5):413–428，1978年。[27] G. Williams, A. Aldrich, and E. Theodorou. 使用协方差变量重要性采样的模型预测路径积分控制。arXiv预印本 arXiv:1509.01149，2015年。[28] M. Lutter, L. Hasenclever, A. Byravan, G. Dulac-Arnold, P. Trochim, N. Heess, J. Merel, and Y. Tassa. 为模型预测智能体学习动力学模型。arXiv预印本 arXiv:2109.14311，2021年。[29] Y. Tassa, T. Erez, and E. Todorov. 通过在线轨迹优化合成并稳定复杂行为。载于2012 IEEE/RSJ智能机器人与系统国际会议，第4906–4913页。IEEE，2012年。[30] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. 用于基于模型的强化学习的信息论模型预测控制。载于2017年 IEEE机器人与自动化国际会议（ICRA），第1714–1721页。IEEE，2017年。[31] P. Indyk and R. Motwani. 近似最近邻：迈向消除维数灾难。载于第三十届 ACM 计算理论年度研讨会论文集，第604–613页，1998年。[32] R. Guo, P. Sun, E. Lindgren, Q. Geng, D. Simcha, F. Chern, and S. Kumar. 利用各向异性向量量化加速大规模推理。载于国际机器学习会议，第3887–3896页。PMLR，2020年。[33] R. Dadashi, L. Hussenot, M. Geist, and O. Pietquin. 原始形式的Wasserstein模仿学习。载于ICLR 2021－第九届国际学习表征会议，2021年。[34] K. Brantley, W. Sun, and M. Henaff. 分歧正则化模仿学习。载于国际学习表征会议，2019年。[35] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Mart´ín-Mart´ín. 离线人类演示对机器人操作学习有何重要性。arXiv预印本 arXiv:2108.03298，2021年。[36] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, 等。RoboTurk：通过模仿进行机器人技能学习的众包平台。载于机器人学习会议，第879–893页。PMLR，2018年。

[37] M. Orsini、A. Raichuk、L. Hussenot、D. Vincent、R. Dadashi、S. Girgin、M. Geist、O. Bachem、O. Pietquin 和 M. Andrychowicz。对抗性模仿学习中什么最重要？*arXiv*预印本 *arXiv:2106.00672*，2021年。[38] I. Kostrikov、O. Nachum 和 J. Tompson。通过离策略分布匹配进行模仿学习。*arXiv*预印本 *arXiv:1912.05032*，2019年。[39] J. Schulman、F. Wolski、P. Dhariwal、A. Radford 和 O. Klimov。近端策略优化算法。*arXiv*预印本 *arXiv:1707.06347*，2017年。[40] S. Ross 和 D. Bagnell。模仿学习的高效归约。载于第十三届国际人工智能与统计学会议论文集，第 661–668 页。JMLR研讨会与会议论文集，2010年。[41] A. Y. Ng、S. J. Russell 等。逆强化学习算法。载于国际机器学习会议，第 1 卷，第 2 页，2000年。[42] K. Lee、K. Saigol 和 E. A. Theodorou。用于模型预测控制的安全端到端模仿学习。*ArXiv*，abs/1803.10231，2018年。[43] J. Silv´erio、Y. Huang、F. J. Abu-Dakka、L. Rozo 和 D. G. Caldwell。使用核化运动基元的不确定性感知模仿学习。载于*2019*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第 90–97 页。IEEE，2019年。[44] Y. Cui、D. Isele、S. Niekum 和 K. Fujimura。用于深度模仿学习的不确定性感知数据聚合。载于*2019*年机器人与自动化国际会议（*ICRA*），第 761–767 页。IEEE，2019年。[45] N. Di Palo 和 E. Johns。Safari：结合想象的安全主动机器人模仿学习。*arXiv*预印本 *arXiv:2011.09586*，2020年。[46] N. Baram、O. Anschel、I. Caspi 和 S. Mannor。端到端可微对抗模仿学习。载于国际机器学习会议，第 390–399 页。PMLR，2017 年。[47] K.-E. Kim 和 H. S. Park。通过核均值嵌入进行模仿学习。载于第三十二届*AAAI*人工智能会议，2018年。[48] R. Wang、C. Ciliberto、P. V. Amadori 和 Y. Demiris。随机专家蒸馏：通过专家策略支持估计进行模仿学习。载于国际机器学习会议，第 6536–6544 页。PMLR，2019年。[49] M. Kelly、C. Sidrane、K. Driggs-Campbell 和 M. J. Kochenderfer。HG-DAGGER：与人类专家进行交互式模仿学习。载于*2019*年机器人与自动化国际会议（*ICRA*），第 8077–8083 页。IEEE，2019年。[50] T. Yu、A. Kumar、R. Rafailov、A. Rajeswaran、S. Levine 和 C. Finn。COMBO：保守的离线基于模型的策略优化。载于强化学习自监督研讨会-*ICLR 2021*，2021年。[51] R. Rafailov、T. Yu、A. Rajeswaran 和 C. Finn。利用潜在空间模型从图像进行离线强化学习。载于动力学与控制学习会议，第 1154–1168 页。PMLR，2021年。[52] X. Zhan、X. Zhu 和 H. Xu。基于轨迹剪枝的基于模型的离线规划。*arXiv*电子预印本，第 arXiv–2105 页，2021年。[53] T. Osa、J. Pajarinen、G. Neumann、J. A. Bagnell、P. Abbeel、J. Peters 等。模仿学习的算法视角。机器人学基础与趋势，7(1-2)：1–179，2018年。[54] P. Englert、A. Paraschos、M. P. Deisenroth 和 J. Peters。概率式基于模型的模仿学习。《适应性行为》，21(5)：388–403，2013年。[55] C. Qi、P. Abbeel 和 A. Grover。快慢结合的模仿：通过决策时规划从演示中实现鲁棒学习。*arXiv*预印本 *arXiv:2204.03597*，2022年。

[56] N. Rhinehart、R. McAllister 和 S. Levine。用于灵活推理、规划和控制的深度模仿模型。*arXiv*预印本 *arXiv:1810.06544*，2018年。[57] P. Tigas、A. Filos、R. McAllister、N. Rhinehart、S. Levine 和 Y. Gal。鲁棒模仿规划：在不确定性下从演示进行规划。载于*NeurIPS 2019*自动驾驶机器学习研讨会，2019年。[58] A. Filos、P. Tigkas、R. McAllister、N. Rhinehart、S. Levine 和 Y. Gal。自动驾驶车辆能否识别分布偏移、从中恢复并适应这种偏移？载于国际机器学习会议，第 3145–3153 页。PMLR，2020年。[59] H. Lin、B. Li、X. Zhou、J. Wang 和 M. Q.-H. Meng。无需交互：使用神经常微分方程进行鲁棒的基于模型的模仿学习。*arXiv*预印本 *arXiv:2104.01390*，2021年。[60] A. Wu、A. Piergiovanni 和 M. S. Ryoo。利用未来图像相似性学习的基于模型的行为克隆。载于机器人学习会议，第 1062–1077 页。PMLR，2020年。[61] J. Richalet、A. Rault、J. Testud 和 J. Papon。模型预测启发式控制。*Automatica*（国际自动控制联合会期刊），14(5)：413–428，1978年。[62] R. Kirk、A. Zhang、E. Grefenstette 和 T. Rockt¨aschel。深度强化学习泛化研究综述。*arXiv*预印本 *arXiv:2111.09794*，2021年。[63] C. Lynch、M. Khansari、T. Xiao、V. Kumar、J. Tompson、S. Levine 和 P. Sermanet。从游戏交互中学习潜在计划。载于机器人学习会议，第 1113–1132 页。PMLR，2020年。[64] E. Jang、A. Irpan、M. Khansari、D. Kappler、F. Ebert、C. Lynch、S. Levine 和 C. Finn。BC-Z：利用机器人模仿学习实现零样本任务泛化。载于第五届机器人学习年会，2021年。[65] S. James、M. Bloesch 和 A. J. Davison。用于少样本模仿学习的任务嵌入控制网络。载于机器人学习会议，第 783–795 页。PMLR，2018年。[66] C. Finn、T. Yu、T. Zhang、P. Abbeel 和 S. Levine。通过元学习实现单样本视觉模仿学习。载于机器人学习会议，第 357–368 页。PMLR，2017年。[67] T. Yu、C. Finn、A. Xie、S. Dasari、T. Zhang、P. Abbeel 和 S. Levine。通过领域自适应元学习，从观察人类行为中实现单样本模仿。*arXiv*预印本 *arXiv:1802.01557*，2018年。[68] J. Tobin、R. Fong、A. Ray、J. Schneider、W. Zaremba 和 P. Abbeel。领域随机化：将深度神经网络从仿真迁移至现实世界。载于*2017*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第 23–30 页。IEEE，2017年。[69] B. Mehta、M. Diaz、F. Golemo、C. J. Pal 和 L. Paull。主动领域随机化。载于机器人学习会议，第 1162–1176 页。PMLR，2020年。[70] E. Xing、A. Gupta、S. Powers 和 V. Dean。KitchenShift：评估模仿式策略学习在领域偏移下的零样本泛化。载于*NeurIPS 2021*分布偏移研讨会：连接方法与应用，2021年。[71] Y. Chebotar、K. Hausman、Y. Lu、T. Xiao、D. Kalashnikov、J. Varley、A. Irpan、B. Eysenbach、R. Julian、C. Finn 等。可执行模型：机器人技能的无监督离线强化学习。*arXiv*预印本 *arXiv:2104.07749*，2021年。[72] K. Cobbe、O. Klimov、C. Hesse、T. Kim 和 J. Schulman。量化强化学习中的泛化。载于国际机器学习会议，第 1282–1289 页。PMLR，2019年。[73] D. Hafner、T. Lillicrap、J. Ba 和 M. Norouzi。梦境控制：通过潜在想象学习行为。*arXiv*预印本 *arXiv:1912.01603*，2019年。[74] D. Hafner、T. Lillicrap、I. Fischer、R. Villegas、D. Ha、H. Lee 和 J. Davidson。学习潜在动力学，以便从像素进行规划。载于国际机器学习会议，第 2555–2565 页，2019年。

[75] D. P. Kingma 和 J. Ba。Adam：一种随机优化方法。*arXiv* 预印本 *arXiv:1412.6980*，2014年。[76] M. Hoffman、B. Shahriari、J. Aslanides、G. Barth-Maron、F. Behbahani、T. Norman、A. Abdol- maleki、A. Cassirer、F. Yang、K. Baumli 等。Acme：用于分布式强化学习的研究框架。*arXiv* 预印本 *arXiv:2006.00979*，2020年。

A 学习曲线

本节展示截至 500 万环境步数的学习曲线。

A.1 初始状态分布失配。

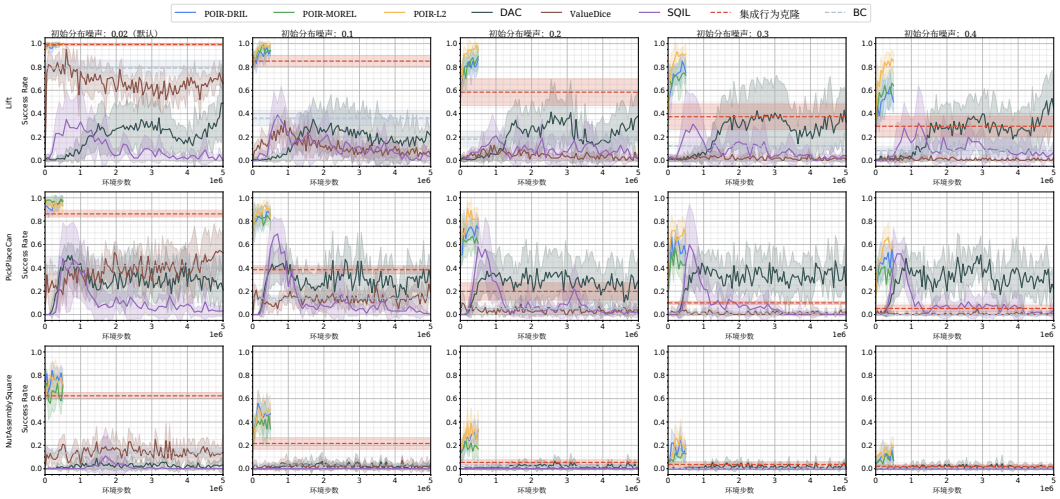


图5： σ_{init} 对 *POIR* 成功率的影响：与行为克隆和 DAC 基线方法相比，*POIR* 在所有环境和初始噪声水平下都能保持更好的表现。随着分布失配加剧，集成行为克隆与 *POIR* 之间的性能差距也随之扩大。阴影区域表示均值上下一个标准差的范围。

A.2 动作噪声。

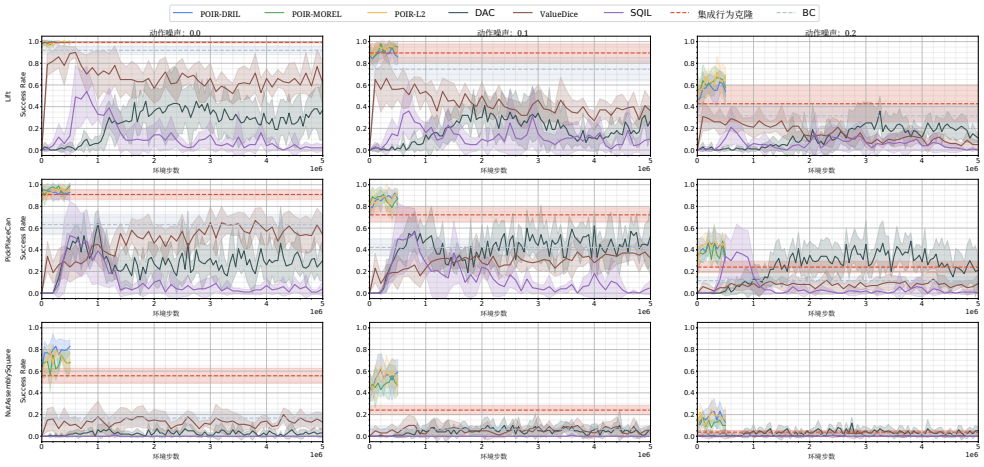


图6： σ_{action} 对 *POIR* 成功率的影响：该图展示了动作噪声对所评估的各种方法的影响。可以观察到，*POIR* 方法始终优于其他基线方法。阴影区域表示均值上下一个标准差的范围。

B 消融实验

我们进行了多项消融实验，以体现该方法各个组成部分的重要性：行为克隆先验、规划器和模仿学习奖励。表1-2-3的消融实验表明，每个组成部分对于 POIR 的性能都至关重要：

- 集成行为克隆：该算法可视为我们方法的一项消融，因为它仅依赖行为克隆先验。结果表明，在没有分布偏移的情况下，集成行为克隆表现出色，但随着偏移增大，其性能会迅速下降。因此，在专家支持集上，行为克隆先验能够选择足以完成任务的动作，但无法从未见状态中恢复。
- 不使用行为克隆先验的 POIR：在此版本中，规划器不依赖行为克隆来选择候选动作，而是以均匀随机的方式采样动作。因此，该方法在所有环境中的成功率均为 0%，这表明行为克隆先验至关重要。
- PPO + DRIL 奖励：在此版本中，我们使用 Acme[2] 算法的公开实现训练 PPO 算法，以最大化所提出的奖励之一（DRIL）。该算法在所有环境中的成功率均为 0%。这表明，即使采用在线强化学习算法，仅凭模仿学习奖励的信号也不足以完成任务。然而，POIR 特有的组合（行为克隆先验 +规划器 + 模仿学习奖励）能够利用奖励信号，找到使智能体保持在专家支持集上的动作；在该区域内，行为克隆先验表现出色。

这些消融实验表明，单独使用任何一个组件都不足以解决分布失配问题：行为克隆无法从未见状态中恢复；规划器在计算资源有限的情况下，只能利用合理的候选动作；单独使用模仿学习奖励，其信号太弱，不足以引导任务完成。

幅度	0.02	0.1	0.2	0.3	0.4
POIR-DRIL（在线）	1	0.9	0.78	0.65	0.52
POIR-DRIL（离线）	1	0.77	0.65	0.4	0.22
集成行为克隆	1	0.85	0.60	0.38	0.3
PPO-DRIL	0	0	0	0	0
POIR-DRIL（不含行为克隆先验）	0	0	0	0	0

表1：Lift 上 POIR算法各组件的消融

幅度	0.02	0.1	0.2	0.3	0.4
POIR-DRIL（在线）	0.92	0.83	0.65	0.5	0.4
POIR-DRIL（离线）	0.96	0.75	0.5	0.33	0.25
集成行为克隆	0.87	0.38	0.2	0.1	0.05
PPO-DRIL	0	0	0	0	0
POIR-DRIL（无行为克隆先验）	0	0	0	0	0

表2：POIR算法在 PickPlaceCan 上各组件的消融实验。

[2]https://git hub.com/deepmind/acme/tree/master/acme/agents/jax /ppo

幅度	0.02	0.1	0.2	0.3	0.4
POIR-DRIL（在线）	0.72	0.38	0.22	0.15	0.08
POIR-DRIL（离线）	0.72	0.33	0.17	0.08	0.03
集成行为克隆	0.62	0.22	0.05	0.03	0.02
PPO-DRIL	0	0	0	0	0
POIR-DRIL（不使用行为克隆先验）	0	0	0	0	0

表 3：POIR算法各组成部分在 NutAssemblySquare 上的消融。

C 超参数与实现细节

本节概述*POIR*、行为克隆和 DAC 的全部超参数。

C.1 行为克隆、EBC 和*POIR*的细节。

离线训练。对于行为克隆策略和世界模型的离线训练，我们采用500 个训练轮次，每个轮次都使用批量大小为256，完整遍历一次离线专家轨迹。*POIR*-DRIL 和 *POIR*-MOREL 分别在离线数据上通过50000 个梯度步数训练模仿学习奖励策略和世界模型，批量大小为256。我们使用 Adam 优化器 [75]，学习率为 1e-4，$\beta_1 = 0.9$，并且 $\beta_2 = 0.999$ 适用于所有神经网络。

***POIR* 在线训练**。对于*POIR* 在线训练，我们使用回放缓冲区存储专家数据和在线规划器策略数据的混合数据 $D = D_e \cup D_a$。我们将从缓冲区采样的在线数据与离线数据的比例记为 $mixture\ ratio$。在线微调开始时，我们使用 $mixture\ ratio = 0.0$，并且每经过 100 个梯度步数，就以 0.05 为增量线性提高该比例，直至达到最大值 0.5。在线训练最多进行 $500k$ 个环境步数。

规划器 以往基于 MPPI [30, 23, 16] 的方法会根据各动作轨迹的回报进行 Softmax 平均，但在回报较大时可能不稳定。我们观察到，实验中 Top-k 平均的效果相当，同时数值稳定性更好。

超参数。 行为克隆和世界模型使用多层感知机，其层数为 5 层， $relu$ 激活函数为 300 个隐藏单元。行为克隆使用单个网络，EBC 和世界模型则使用由 27个神经网络组成的集成。我们按照桌子 4所示对参数进行了搜索范围设定，最终值显示在最右列。对于行为克隆和 EBC，归一化状态和动作并无益处。对于*POIR*，我们同时对状态和动作进行归一化。

参数	搜索范围	最终值
学习率	{1e-4, 5e-4}	1e-4
隐藏单元	{100, 300, 500}	300
层数	{1, 3, 5}	5
网络数量	{1, 3, 9, 27}	27
激活函数	$\{relu, tanh\}$	$relu$
规划器：轨迹数量 N	{1000, 4000}	4000
规划器：动作噪声 σ	{0.1, 0.15, 0.2}	0.2
规划器：时域 H	{1, 5, 10, 20}	5
POIR：观测值归一化	{真，假}	True
POIR：动作归一化	{是，否}	True
行为克隆：观测值归一化	{是，否}	否
行为克隆：动作归一化	{是，否}	假

表 4：EBC、*POIR* 中的世界模型以及规划器的超参数。BC 使用相同的网络超参数，但仅使用一个网络。BC 和 EBC 均不使用规划器。

C.2 基线方法的详细信息（DAC、ValueDICE、SQIL、PPO+DRIL）。

超参数选择。 对于每个基线和超参数集，我们在 Lift 上使用 3 种初始化噪声 $\sigma \in \{0.02, 0.2, 0.4\}$ 运行算法。表现最佳的超参数随后用于在所有环境和噪声扰动值上的评估。

C.2.1 DAC。

对于 DAC，我们参考了 Orsini 等人提供的超参数配置。 [37]并在此基础上进行了调优。完整的超参数列表见表5。不同超参数的详细信息见此处：
https://github.com/deepmind/acme/tree/master/acme/agents/jax/ail。

参数	搜索范围	最终值
策略多层感知机深度	2	2
策略多层感知机宽度	256	256
评论家 MLP 深度	3	3
评论家 MLP 宽度	512	512
激活函数	$relu$	$relu$
折扣因子	0.97	0.97
批量大小	256	256
强化学习算法	SAC	SAC
SAC 学习率	{1e-4, 3-e4}	3e-4
SAC 熵维度	-0.5	-0.5
n 步回报	{1, 3}	3
回放缓冲区大小	3e6	3e6
减去 logpi	假	假
吸收状态	True	True
判别器输入	$(s,\ a)$	$(s,\ a)$
判别器多层感知机深度	2	2
判别器多层感知机宽度	128	128
判别器激活函数	$relu$	$relu$
梯度惩罚系数	{1, 10}	10
观测值归一化	{假，真}	True
学习率	3e-5	3e-5
奖励函数	$ln(D) - ln(1 - D)$	$ln\ (D) - ln\ (1 - D)$

表 5：DAC 的超参数。

C.2.2 ValueDice

我们使用了 Acme 库中的参考实现 [76]。超参数搜索范围和选定的超参数见表 6。不同超参数的详细信息请见：
https://github.com/deepmind/acme/tree/master/acme/agents/jax/value_dice。

参数	搜索范围	最终值
nu 学习率	{0.001, 0.03, 0.05}	0.01
nu 学习率	{0, 3, 10}	10
策略学习率	{1e-5, 3e-5, 1e-4}	1e-5
策略正则化系数	{0, 1e-5, 1e-4, 1e-3}	0.0001
α	{0, 0.05}	0.05
批量大小	256	256
策略多层感知机深度	2	2
策略多层感知机宽度	256	256
nu 多层感知机深度	2	2
nu 多层感知机宽度	256	256

表 6：ValueDICE 的超参数。

C.2.3 SQIL

我们使用了 Acme 库中的参考实现 [76]。超参数搜索和选定的超参数见桌子 7。不同超参数的详细信息见此处：
https://github.com/deepmind/acme/tree/master/acme/agents/jax/sac。

参数	搜索范围	最终值
策略多层感知机深度	2	2
策略多层感知机宽度	256	256
评论家 MLP 深度	3	3
评论家 MLP 宽度	512	512
激活函数	$relu$	$relu$
折扣因子	0.97	0.97
批量大小	256	256
强化学习算法	SAC	SAC
SAC 学习率	{1e-4、3-e4、1e-3}	1e-4
SAC 熵维度	-0.5	-0.5
n 步回报	{1, 3}	3
回放缓冲区大小	1e6	1e6

表 7：SQIL 的超参数。

C.2.4 PPO+DRIL

我们使用了 Acme 库中的参考实现 [76]。超参数搜索和选定的超参数见桌子 8。PPO 的不同超参数的详细信息见此处：
`https://github.com/deepmind/acme/tree/master/acme/agents/jax/ppo`。由于每种配置的成功率均为 0，我们无法区分不同超参数的影响。

参数	搜索范围	最终值
策略多层感知机深度	5	5
策略多层感知机宽度	256	256
激活函数	$relu$	$relu$
PPO 学习率	{1e-4, 3-e4, 1e-3}	1e-4
PPO 熵成本	{0, 3e-4}	
PPO 训练轮数	2	2
PPO 小批次数	8	8
PPO 展开长度	8	8
PPO 价值损失系数	1	1
DRIL 神经网络数量	27	27
DRIL 批量大小	256	256
DRIL 训练轮数	10	10

表 8：PPO+DRIL 的超参数。

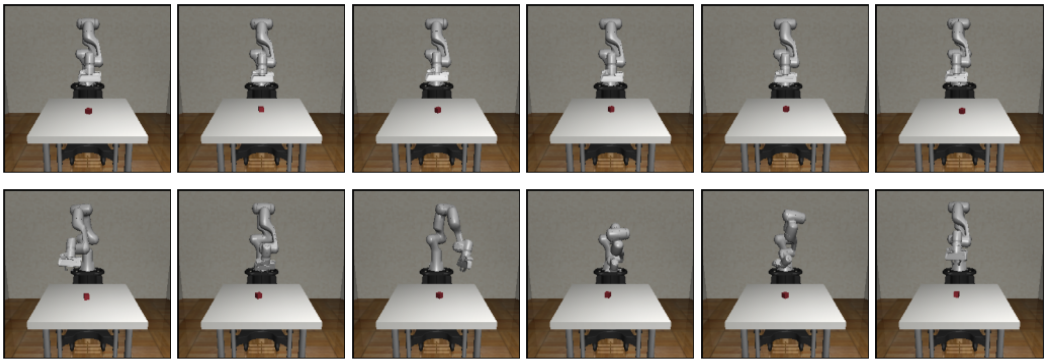
D 完整轨迹优化算法

算法 3*POIR*-选择动作

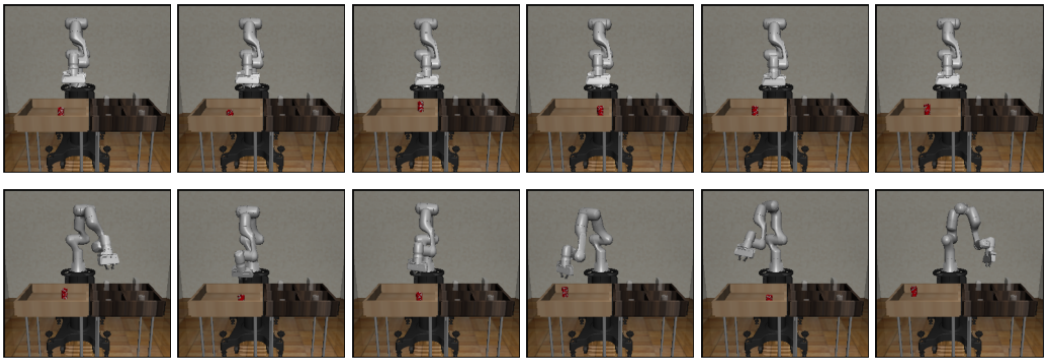
选择动作$(s, \pi_{BC}, \hat{P}, r_{IL}, H, N, \sigma, k, K)$:
设置 $\mathbf{R}^N = \vec{0}_N$ 此变量存储我们的 N 个轨迹回报。
设置 $\mathbf{A}_H^N = \vec{0}_H^N$ 此变量存储我们长度为 H 的 N 条动作轨迹。
对于 $n = 1..N$ **执行**
    $l = n$ 模 K 在整个轨迹中使用一致的集成头。
    $\hat{s}_1 = s, a_0 = T_0, R = 0$
    对于 $t = 1..H$ **执行**
        $\epsilon \sim \mathcal{N}(0, \sigma^2)$
        $\tilde{a}_t = \pi_{BC}^l(\hat{s}_t) + \epsilon$ 使用行为克隆策略采样当前动作。
        $\hat{s}_{t+1} = \hat{P}^l(\hat{s}_t, \sim a_t)$ 从环境模型中采样下一状态。
        $R^n = R^n + r_{IL}(\hat{s}_t, \sim a_t)$ 对所有集成成员的平均奖励取平均值。
        $\mathbf{A}_t^n = \tilde{a}_t$
    结束循环
结束循环
按照 R^N 降序排列A_H^N。
$a_1 = 1/k \sum_{n=1}^{k} A_1^n$
回报 a_1

E 环境中初始化噪声的影响

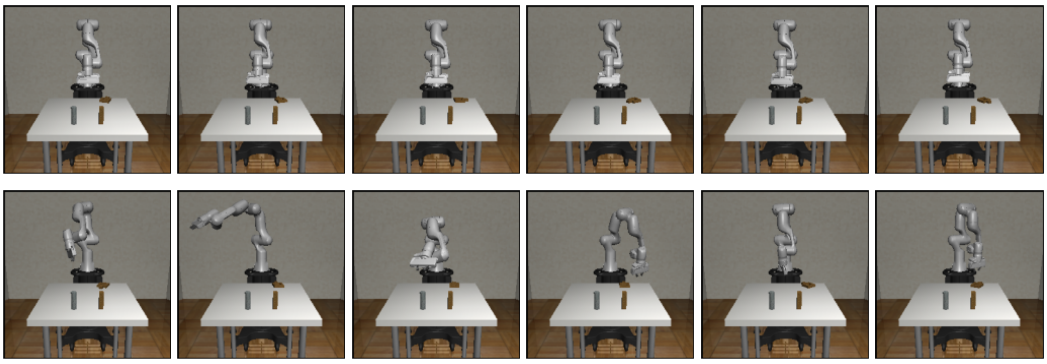
我们展示了初始化噪声对 Robomimic 任务中机器人手臂初始状态的影响。噪声值 σ_{init}用于缩放施加到机器人各初始关节位置上的高斯随机噪声的标准差。默认值 σ_{init} 为 0.02，而在我们的实验中，我们将其增大到 0。4 σ_{init}。其影响如图 7所示：对于默认噪声设置（$\sigma_{init} = 0.02$），我们为每个环境采样 6 个随机初始状态；并对 $\sigma_{init} = 0.4$ 设置也进行了采样。



(a) Lift：上方为默认初始化噪声 0.02，下方为初始化噪声 0.4。



(b) PickPlaceCan：上方为默认初始化噪声 0.02，下方为初始化噪声 0.4 .



(c) NutAssemblySquare：上方为默认初始化噪声 0.02，下方为初始化噪声 0.4。

图 7：Robosuite 环境中随机采样的初始状态。对于所有图，顶行显示初始化噪声默认为 0.02 时的初始状态，底行显示初始化噪声为 0.4 时的初始状态。请注意，在图 7a中，初始化噪声较高时，机器人手臂较不容易从方块正上方的位置开始。

F 涌现式重试与视频

图 8 展示了 POIR 智能体的重试行为。我们还附上了 POIR-L2 智能体在三个环境中的多段视频（经过 500k 步微调后）。即使初始位置与演示中所见的位置差异很大，POIR-L2 也能完成任务。这些视频还展示了 POIR 不断重试任务直至完成的能力。相比之下，集成行为克隆无法重试任务；如其中一段附带视频所示，当初始位置与演示中的位置相差较大时，它无法完成任务。我们还附上了一段 DAC 在 Lift 环境中运行的视频（经过 5M 步训练后），以展示 DAC 的一种失败模式：它不断重试任务，却始终没有触碰物体。

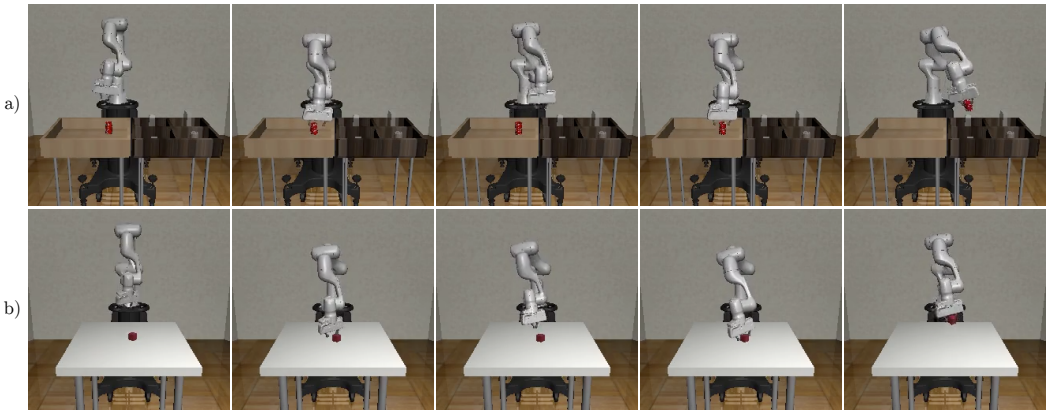


图 8：可视化 *POIR* 智能体在 a) PickPlaceCan 和 b) Lift 中的重试。我们在每个环境中选取一条轨迹，并按时间顺序展示其中部分帧。请注意，机械臂尝试抓取物体，未抓住物体便向上移动，随后返回并成功抓住物体。行为克隆（BC）不会呈现这种行为。