



模仿学习概述

汇报人：李明谦

2025年6月28日

规格严格 功夫到家



目录

01

模仿学习概
述

02

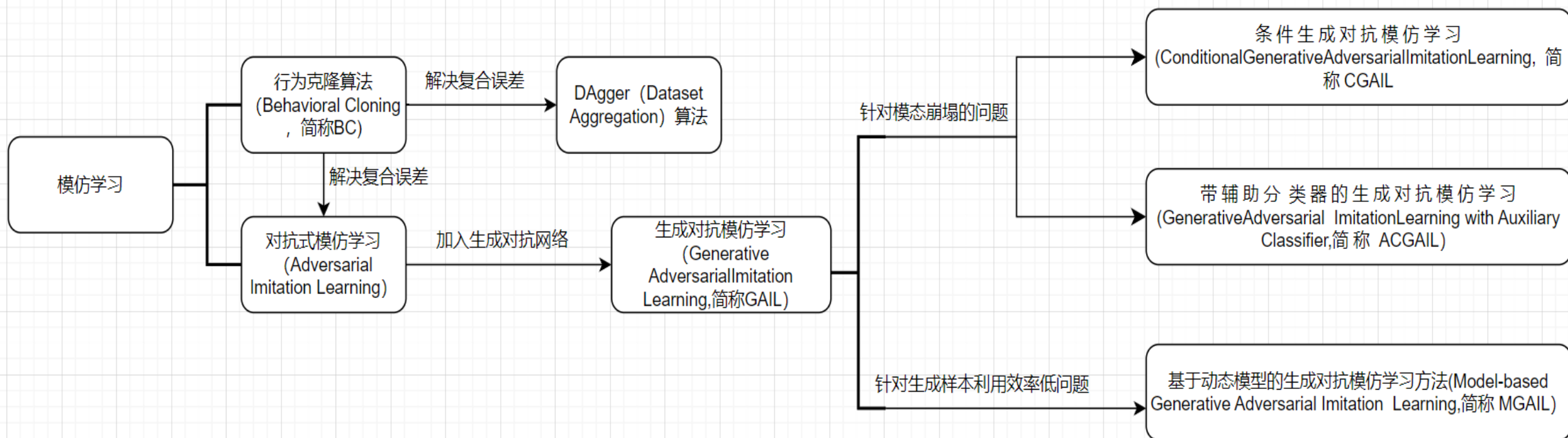
未来研究方
向

模仿学习概述

模仿学习（Imitation Learning）（IL）——从专家示例中学习——是一种让智能体（机器人）像人类专家一样能够进行智能决策的方法。

在模仿学习中,获取专家样本集合的方式主要有以下两种:

- (1)由人类专家示范而获得专家样本集合;
- (2)通过RL方法对专家手工定义的标准奖赏函数学习,得到贪婪策略,再由贪婪策略得到专家样本集合



行为克隆算法 (BC, Behavioral Cloning)

行为克隆的实现流程与典型的监督学习任务（如图像分类、回归预测）几乎完全一样。

步骤一：收集专家数据 (Data Collection)

我们记录下专家在每一个决策点所观察到的状态（Observation）和它所执行的动作（Action）。将这些数据整理成一个数据集 D ，其中包含了大量的状态-动作对： $D=\{(s_1,a_1),(s_2,a_2),(s_3,a_3),\dots,(s_N,a_N)\}$ 。

步骤二：构建监督学习问题 (Problem Formulation)

我们将这个数据集视为一个标准的监督学习任务：

我们的目标是学习一个策略模型 $\hat{\pi}_{\theta}(a|s)$ （通常是一个神经网络， θ 是网络参数），使得 $\hat{\pi}_{\theta}(a|s)$ 的输出尽可能地接近真实的专家动作 a 。

步骤三：模型选择与训练 (Model Selection & Training)

对于任意给定的一个需要估计的概率模型 $\hat{\pi}_{\theta}$ （ θ 是被估计的参数），一个数据样本 (s, a) 的似然可以表示为 $\hat{\pi}_{\theta}(a|s)$ 。因此，最大化对数似然模型可以表述为：

$$\begin{aligned} \max_{\hat{\pi}_{\theta} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}} \quad & \sum_{(s,a) \in \mathcal{D}} \log \pi(a|s) \\ \text{s.t.} \quad & \sum_{a \in \mathcal{A}} \hat{\pi}_{\theta}(a|s) = 1, \quad \forall s \in \mathcal{S}. \end{aligned}$$

行为克隆算法 (BC, Behavioral Cloning)

$$\begin{aligned} \max_{\hat{\pi}_{\theta} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}} \quad & \sum_{(s,a) \in \mathcal{D}} \log \pi(a|s) \\ \text{s.t.} \quad & \sum_{a \in \mathcal{A}} \hat{\pi}_{\theta}(a|s) = 1, \quad \forall s \in \mathcal{S}. \end{aligned}$$

可以验证上述公式对应的问题是一个凸优化问题，并且其最优解的形式对于有限状态和动作空间的马尔可夫决策过程问题很简单。具体而言，最优解可以用计数来求解：对于数据集出现里的状态 s ，我们令

$$\hat{\pi}_{\theta}(a|s) = \frac{\sum_{(s_i, a_i) \in \mathcal{D}} \mathbb{I}(s_i = s, a_i = a)}{\sum_{(s_i, a_i) \in \mathcal{D}} \mathbb{I}(s_i = s)},$$

这里 $\mathbb{I}(\cdot)$ 表示示例函数，即如果 $\cdot$ 为真，那么 $\mathbb{I}(\cdot) = 1$ ；否则 $\mathbb{I}(\cdot) = 0$ 。对于数据集里没有出现过的状态 s ，我们可以令其动作分布为一个均匀分布，也就说 $\pi(a|s) = 1/|\mathcal{A}|$ 。

行为克隆算法 (BC, Behavioral Cloning)

缺点：复合误差

when driving for itself, the network may occasionally stray from the center of road and so must be prepared to recover by steering the vehicle back to the center of the road

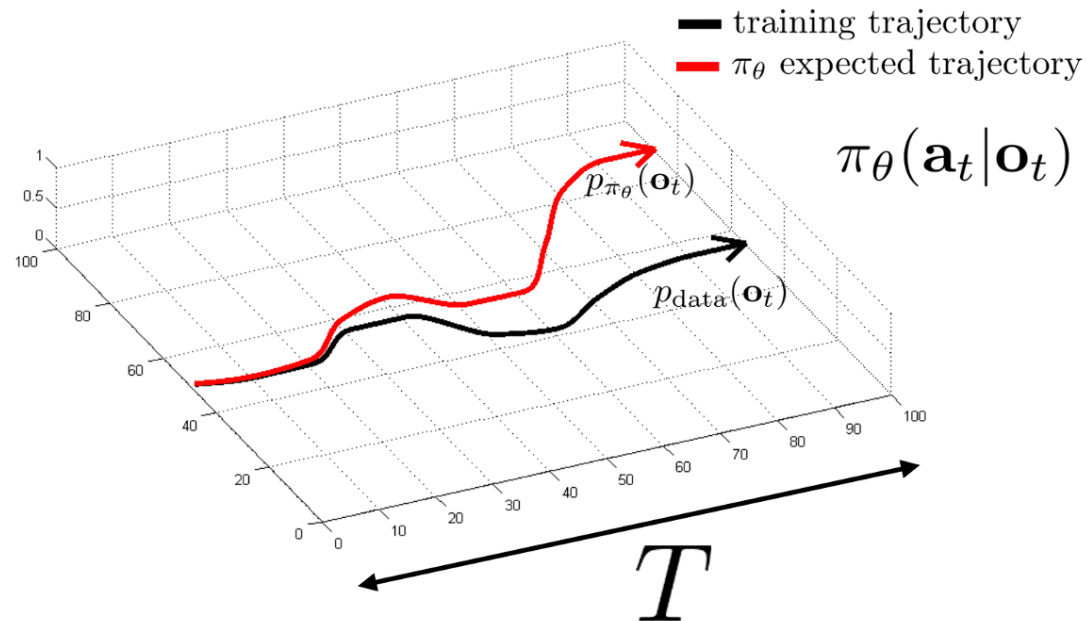
——Dean A Pomerleau. “Alvinn: An autonomous land vehicle in a neural network”. In: Advances in Neural Information Processing System. 1989

由于模型早期的预测出现小错误，导致后续状态逐渐偏离专家轨迹，从而产生更大的预测误差，最终行为崩溃。

解决方法：

DAGger算法

对抗式模仿学习



DAgger 算法

为了解决复合误差问题，Stéphane Ross 等人提出了 DAgger (Dataset Aggregation) 算法。这是一个基于在线学习 (Online Learning) 的算法，其把行为克隆得到的策略与环境不断的交互，来产生新的数据。在这些新产生的数据上，DAgger 会向专家策略申请示例；然后在增广后的数据集上，DAgger 会重新使用行为克隆进行训练，然后再与环境交互...；这个过程会不断重复进行。由于数据增广和环境交互，DAgger 算法会大大减小未访问的状态的个数，从而减小复合误差。

Algorithm 1 DAgger 算法

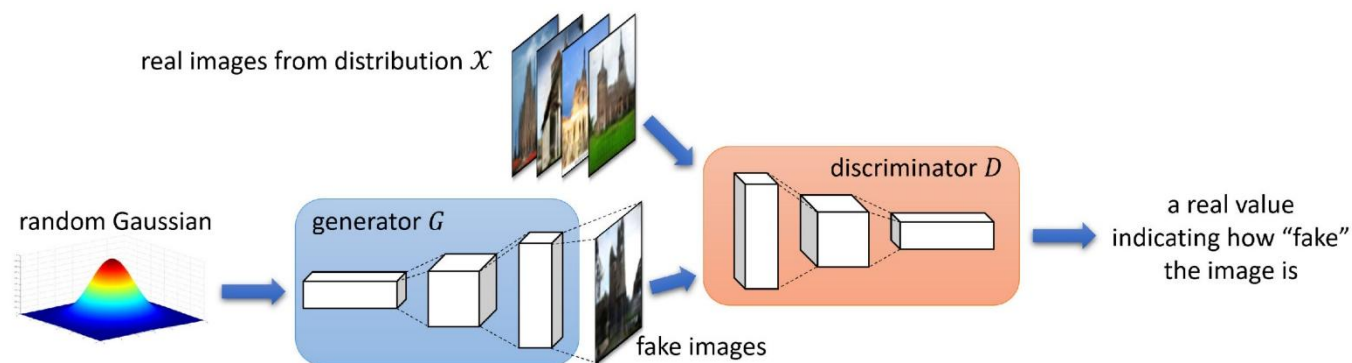
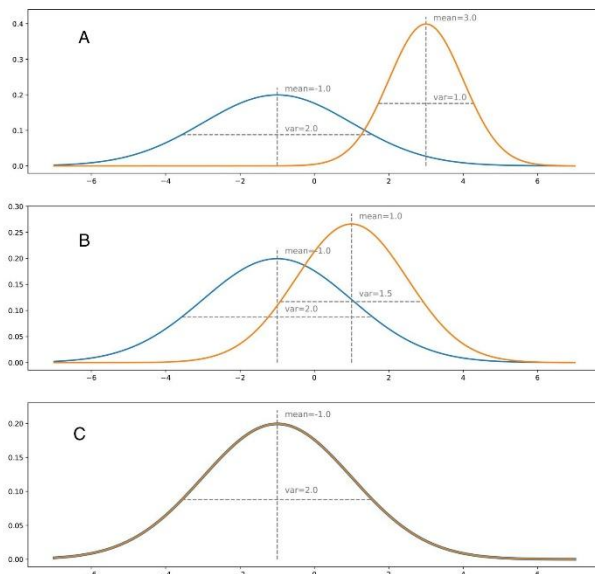
- 1: 初始化数据集 $\mathcal{D}$ 和策略 $\hat{\pi}_1$ 。
- 2: **for** episode $i = 1, 2, \dots, T$ **do**
- 3: 设置混合策略 $\pi_i = \beta_i \pi^E + (1 - \beta_i) \hat{\pi}_i$ 。
- 4: 让 π_i 与环境交互得到一条新的轨迹 tr 。
- 5: 向专家咨询示例动作得到新的数据集 $\mathcal{D}_i = \{(s, \pi^E(s)) : s \in \text{tr}\}$ 。
- 6: 增广数据集 $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$ 。
- 7: 在增广后的数据集上重新使用行为克隆训练策略 $\hat{\pi}_{i+1}$
- 8: **end for**

对抗式模仿学习

对抗式模仿学习基于逆向强化学习的模仿学习(Imitation Learning via Inverse Reinforcement Learning, 简称IRL-IL)方法得来。

IRL-IL假设专家策略等价于由未知的真实奖赏函数推导出的最优策略.从字面上理解,逆向强化学习是强化学习的逆向过程,它根据给定的专家样本求解未知的奖赏函数.基于解得的奖赏函数,IRL-IL通过RL方法求解最优策略的方式,间接地还原专家策略.这种模仿专家的方式使IRL-IL具备了长远规划的能力.因此,IRL-IL能有效解决BC的级联误差问题并表现出更强的泛化性、鲁棒性

在IRL-EL的基础上加入生成对抗网络得到对抗式模仿学习方法。原始的生成对抗网络Generative Adversarial Nets,简称GANs)由生成模型(又称生成器)和判别模型(又称判别器)这两个相对抗的网络模型共同构成。其中,生成模型能够产生符合期望的样本输出的模型,如根据噪声输入产生高维图片,判别模型则用于判断生成模型所输出的结果真假



How can generative adversarial networks (GANs) learn real-life distributions easily?

最早出现且最具代表性的对抗式模仿学习方法是Ho等人于2016年提出的生成对抗模仿学习方法(Generative Adversarial Imitation Learning,简称GAIL)。如果把策略表征为从状态输入到动作输出的生成模型,那么模仿学习根据专家样本学习策略的过程其实就是生成模型的训练过程.在GAIL中,根据输入状态输出动作的策略可类比为生成器,而根据输入专家样本或生成样本输出奖赏值的奖赏函数可类比为判别器.从而,GAIL将求解奖赏函数的过程类比作判别器的训练过程,将策略的学习过程类比作生成器的训练过程.

Algorithm 2 生成对抗式模仿学习

Input: 专家示例 $\mathcal{D} = \{(s, a)\} \stackrel{i.i.d.}{\sim} d^{\pi^E}$, 将策略参数和判别器参数分别初始化为 θ_0, w_0 。

- 1: **for** $i = 1, 2, \dots, N$ **do**
- 2: 使用策略 π_{θ_i} 在环境中采样, 得到数据集 $\mathcal{D}_i = \{(s_t, a_t)_t\}$ 。
- 3: 通过下面的梯度公式来更新判别器参数 $w_i \rightarrow w_{i+1}$:

$$\sum_{(s,a) \in \mathcal{D}} \nabla_w \log(D_w(s, a)) + \sum_{(s,a) \in \mathcal{D}_i} \nabla_w \log(1 - D_w(s, a)).$$

- 4: 设置奖赏函数 $r(s, a) = -\log(1 - D_{w_{i+1}}(s, a))$, 利用信赖域策略优化算法 [53] 来更新策略参数 $\theta_i \rightarrow \theta_{i+1}$ 。
- 5: **end for**

Output:

优点:

基于生成对抗网络框架, GAIL的策略和奖赏函数模型可运用神经网络来自动抽取样本的抽象特征。因此,GAIL具有更强的表征能力。并且, GAIL直接将策略作为学习的目标, 它运用高效的策略梯度方法训练策略模型。从而,GAIL能避开IRL-IL需消耗大量计算资源的内部计算过程, 具有更高效的计算能力。已有工作表明, GAIL能够在如自动驾驶、仿真及真实机器人操控等复杂的大规模问题中表现出优异的性能。

缺点:

1、模态崩塌问题

模态崩塌指的是生成器只学会生成数据分布中的某几个“模式”（或者甚至一个模式），导致生成的样本缺乏多样性。

发生模态崩塌的原因:

- 生成器发现某个模式比较容易骗过判别器，就“卡”在这个模式上不动，反复生成类似的样本。
- 判别器的反馈不够强或不够全面，无法推动生成器去探索和生成其他模式。
- 训练过程中梯度消失或不稳定，导致生成器难以学习更全面的分布

2、生成样本利用效率低

原因1：无模型策略学习方式

这是最主要、最根本的原因。GAIL通常使用无模型的强化学习算法（如TRPO,PPO）来优化其生成器（策略）。无模型方法不试图学习环境的动态模型（即状态转移函数 $P(s'|s,a)$ ）。智能体不知道在状态 s 执行动作 a 后会确切地发生什么，它只能通过大量的实际尝试（Trial-and-Error）来直接学习一个从状态到动作的映射（策略 $\pi(a|s)$ ）。为了评估一个策略的好坏，智能体必须在真实环境中运行该策略，收集成千上万条轨迹。它无法像“有模型”方法那样，在内部“想象”或“规划”不同动作序列的后果。在GAIL中，学习信号的传递链条很长：

智能体与环境交互→生成轨迹→轨迹喂给判别器→判别器更新→判别器输出奖励信号→智能体根据奖励更新策略

同时由于GAIL的稳定算法（如TRPO,PPO）都是On-Policy（同策略）的，On-Policy的核心要求：用来更新当前策略的数据，必须是由当前这个策略本身收集的。后果就是每当策略（生成器）更新一点点，之前花费巨大代价收集到的所有样本就必须被丢弃，智能体需要用它这个微小更新后的新策略，重新去环境中收集一批全新的数据。这造成了极大的样本浪费。



GAIL

原因2.随机性策略（Stochastic Policy）的加剧效应

在强化学习中，策略通常是随机的，即对于同一个状态 s ，策略 $\pi(a|s)$ 会输出一个动作的概率分布，而不是一个确定的动作。这在探索（Exploration）中至关重要，但在模仿学习中却带来了问题。

因为策略是随机的，即使在完全相同的状态下，智能体这次和下次的行为也可能不同。这导致它生成的轨迹千差万别，方差极大。

对判别器的影响：判别器 D 面临一个艰巨的任务。它不仅要区分专家的（通常是低方差、确定性强的）轨迹，还要从一大堆由智能体随机策略产生的、质量参差不齐的“噪音”轨迹中学习。这使得判别器的学习变得困难且不稳定。

GAIL中的随机策略，使得智能体在探索过程中会产生大量“不像专家”的行为。这些行为虽然有助于探索环境，但对于“欺骗”判别器这个核心任务来说，它们是低质量的负样本，拖慢了模仿学习的收敛速度。智能体需要花费大量样本去尝试那些最终会被证明是“错误”的动作。

总结：两大因素的协同作用

无模型框架决定了GAIL必须通过海量的、昂贵的真实世界试错来学习，这是其效率低下的结构性原因。

随机性策略则为这个本已低效的过程中注入了大量的噪声和不确定性。它使得每一次试错（样本）所能提供的信息价值降低，因为奖励信号本身变得不稳定，导致策略的更新方向不明确，收敛速度进一步放缓。

因此，我们可以这样理解：

无模型决定了GAIL“必须吃很多饭才能学饱”（需要大量样本），而随机性策略则决定了“每一口饭的营养都很低”（每个样本的信息价值不高），最终导致GAIL成为一个“大胃王”，样本利用效率极低。



针对GAIL的改进

解决模态崩塌:

1、条件生成对抗模仿学习 (Conditional Generative Adversarial Imitation Learning, 简称CGAIL)

从专家样本中活的模态的标签数据, 并将这些模态标签数据可作为条件约束来构建策略和奖赏函数模型。

2、带辅助分类器的生成对抗模仿学习 (Generative Adversarial Imitation Learning with Auxiliary Classifier, 简称ACGAIL)

新的辅助网络用来对样本所属的模态类别进行分类, 从而帮助GAIL重构关于模态的条件信息, 通过引入辅助的网络模型, ACGAIL不仅能监督地对模态数据学习, 还能充分利用额外的模态标签数据学习样本中的抽象特征

解决生成样本利用效率低:

基于动态模型的生成对抗模仿学习方法 (Model-based Generative Adversarial Imitation Learning, 简称MGAIL)

MGAIL在原始的GAIL中引入了一个前向模型来对随机的动态环境建模。并且, MGAIL用重参数化技巧对策略中随机采样动作的过程建模。从而, MGAIL通过新引入的前向模型构建了端到端可微分的梯度计算流图。它使策略能够根据反向传播的链式梯度法则来更新奖赏函数模型中的参数信息。

环境模仿学习

基于模型的强化学习 (Model-Based Reinforcement Learning)

前面介绍的模仿学习算法，是用来模仿专家策略。在智能体与环境交互的过程中，除了智能体的策略之外，另一个重要的成分是环境的奖赏和转移函数。基于模型的强化学习则是学到近似的奖赏和转移函数，完成后就在这个近似的环境中进行策略学习，从而降低样本复杂度。

基于模型的强化学习是一类在虚拟的马尔可夫决策过程中进行学习策略的算法。具体而言，基于模型的强化学习首先从与真实环境交互得到的样本中，学习到一个经验环境模型 $\widehat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \widehat{\rho}, \widehat{P}, \widehat{r}, \gamma)$ 这个经验模型包含近似的奖赏函数 $\widehat{r}$ 和转移函数 $\widehat{P}$ 和初始状态分布 $\widehat{\rho}$ 之后智能体便在这个虚拟的环境模型上进行采样和策略的学习。在基于模型的强化学习中，由于智能体可以在经验环境模型上完成采样，从而降低了在真实环境的样本复杂度

Algorithm 4 基于模型的强化学习

Input: 初始化策略 π ，奖赏函数 $\widehat{r}$ ，转移函数 $\widehat{P}$ ，初始状态分布 $\widehat{\rho}$ ，和空数据集 $\mathcal{D} = \emptyset$ 。

1: **for** $i = 1, 2, \dots, N$ **do**

2: 使用策略 π 在环境中采样，得到数据集 $\mathcal{D} = \mathcal{D} \cup \{(s_t, a_t, s'_{t+1})\}$ 。

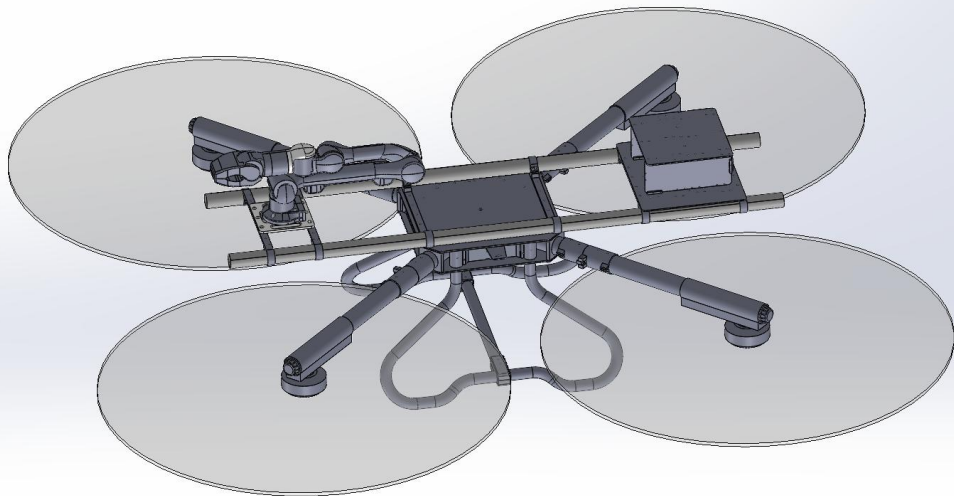
3: 通过数据集 $\mathcal{D}$ 训练奖赏函数 $\widehat{r}$ 和转移函数 $\widehat{P}$ 和初始状态分布 $\widehat{\rho}$ 。

4: 在经验环境模型中优化策略: $\pi \leftarrow \operatorname{argmax}_{\pi'} V_{\widehat{\mathcal{M}}}^{\pi'}$ 。

5: **end for**

Output:

现有的研究成果和未来研究方向



在Isaac lab中搭建大四轴的仿真环境

在大四轴的机械臂上复现一些模仿学习的算法，通过Isaac lab仿真出来



The background is a light gray with several abstract geometric shapes. There are four circles: a large pink one on the left, a medium blue one at the top center, a large gray one at the top right, and a medium pink one at the bottom right. There are also four triangles: a small pink one at the top center, a small gray one to the right of the center, a small dark gray one on the left, and a small blue one on the right.

THANK YOU