arXiv:2301.07182v1 [cs.NE] 3 Jan 2023

# 基于奖励外推的遗传模仿学习

**Boyuan Zheng**[1] , **Jianlong Zhou**[1] 和**Fang Chen**[1]
[1]悉尼科技大学 Boyuan.Zheng-1@student.uts.edu.au,
Jianlong.Zhou@uts.edu.au

**摘要**

模仿学习在各种领域都展现出卓越的性能。然而，模仿学习也受到诸多先决条件的限制。研究界已开展大量研究来缓解这些限制，例如采用随机策略以避免遇到未见状态、消除对动作标签的需求，以及从次优示范中学习。受自然繁殖过程的启发，我们提出了一种名为GenIL 的方法，将遗传算法与模仿学习相结合。遗传算法通过繁殖具有不同回报的轨迹提高了数据效率，并帮助模型估计出更准确、更精简的奖励函数参数。我们在 Atari 和 MuJoCo 两个领域对 GenIL 进行了测试，结果表明，在输入数据有限时，GenIL 在外推准确率、鲁棒性和整体策略性能方面均成功超越了以往的外推方法。

## 1 引言

模仿学习利用来自其他来源的示范来复现目标行为，而无需手工设计奖励函数；近年来深度学习的进展进一步推动了模仿学习的发展。由于模仿学习具有“不依赖奖励函数”的特点，因此在设计奖励函数并不切实际的问题中发挥着关键作用，例如机械臂操作（Duan 等，2017年）和自动驾驶（Codevilla 等，2018年）。然而，现有的大量研究在开发方法时，都假设输入的示范是最优或近乎最优的。这一假设为智能体性能设定了上限，并在一定程度上加剧了次优示范对最终性能的负面影响（Zheng 等，2021年）。当输入示范为次优示范时，智能体性能最终会收敛至次优水平。此外，次优示范比最优示范更加丰富，也更容易收集，因此不加以利用是一种浪费。具体而言，由于模仿学习具有跨学科特性，来自其他研究领域的专家所提供的最优示范，获取成本可能很高，实际操作中也可能行不通。

值得庆幸的是，近期已有多项研究（Chang 等，2015年；Brown 等，2019年；Ding等，2019年）对此进行了探究，并尝试利用次优示范实现优于示范者的性能。基于对抗结构的模仿学习和外推是解决这一问题的两种主流方法。最先进的对抗方法 GAIL（Ho 和 Ermon，2016年）在处理次优输入时表现出相当强的鲁棒性。后续方法（如 RAIL（Zuo 等，2020年））构建了更成熟的框架，以更好地利用次优示范，但结构脆弱等缺点也源于对抗结构。另一方面，研究界也广泛探索了另一条研究方向：通过外推诱导出优于示范者的行为。除了次优示范外，输入通常还包含额外信息，例如人类偏好或不同程度的噪声。这些额外信息有助于模型建立次优示范与额外信息之间的关系，并推导出接近最优的策略。外推方法在 Atari 和 MuJoCo 领域都取得了相当出色的性能，但获取额外信息通常十分费力。

在这项工作中，我们提出了一种名为遗传模仿学习（GenIL）的模仿学习方法。该方法采用从观察中模仿（IfO）设定，将遗传算法的概念与模仿学习相结合，并利用次优示范实现更紧凑的外推。与以往工作不同，GenIL 摒弃了繁琐的流程，借鉴了遗传算法的思想。通过对输入轨迹进行交叉和变异，GenIL 能够生成质量各异的新轨迹。这些新生成的轨迹可作为额外信息，辅助外推过程。我们在多个 Atari 和 MuJoCo 领域中，将本方法与其他相关方法进行了比较。结果表明，与现有模仿学习算法相比，GenIL 能够提高数据效率，并为未见轨迹外推出更准确、更紧凑的奖励。在整体策略性能方面，GenIL 同样表现出竞争力，并优于以往方法。

我们的工作贡献包括：

• 我们提出了一种更优的奖励外推模型 GenIL，它通过整合遗传

算法与模仿学习。

• GenIL 能更好地利用次优数据，并且仅使用两条表现各异的轨迹作为输入来训练模型。

• 我们展示了一系列实验，并表明 GenIL 能够从次优示范中学习到有意义的策略，且优于以往方法。

## 2 相关工作

本节简要总结从次优示范中学习、从观察中模仿、奖励外推和遗传算法等相关主题的最新进展。

模仿学习旨在借助示范复现目标行为，在训练自主智能体方面发挥着关键作用。通常，模仿学习可分为行为克隆（Bain 和 Sammut，1999年）和逆向强化学习（Russell，1998年）。两者的区别在于是否还原奖励函数。通常，这些方法假设输入的示范是最优的（或近乎最优），并将从最优示范中批量采样的状态-动作对作为输入。随着模仿学习研究的不断深入，研究人员开始对这些假设提出质疑，其中包括对最优示范必要性的质疑。尤其是在 GAIL（Ho 和 Ermon，2016年）提出之后，研究界深入探讨了从次优示范中学习的可行性。一种常用技术是将事后经验回放（HER）（Andrychowicz 等，2017年）融入 GAIL。HGAIL（Liu 等，2019年）和 GoalGAIL（Ding 等，2019年）等研究利用 HER，沿着次优示范设定新目标，从而超越次优示范。在 RAIL（Zuo等，2020年）中，研究者除了扩展 HER，还提出了一种名为事后复制的技术，利用原始示范和事后获得的示范来加快学习过程。

随着新型模仿学习算法的出现，受限的输入形式也得到了突破。直到2018年，状态-动作对序列一直是模仿学习算法的输入数据。动作通常用作标签来训练模型。然而，2018年，Liu 等人提出了一种名为观测模仿（IfO）的新型模仿范式，改变了以往的传统。受人类和动物模仿方式的启发，IfO 不再需要动作标签，只利用观测来训练模型。IfO 采用监督学习方式，使用编码器-解码器结构从观测中提取特征，并通过测量真实值与预测观察之间的距离，取代传统的动作标签。IfO 的出现为无法获取动作标签的场景提供了解决方案，也扩大了可用训练资源。大量研究（Torabi 等人，2019年；Zhu 等人，2021年；Aytar等人，2018年）致力于将原始观察用作输入。例如，Sermanet 等人于2018年提出了一种自监督

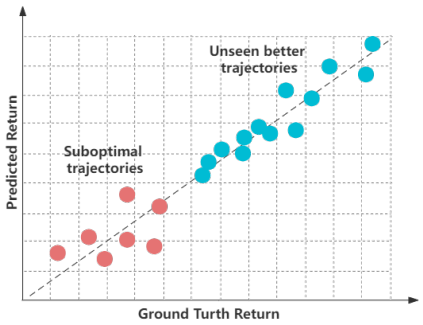


图 1：关于外推的图解。红色点表示用于训练的次优示范，蓝色点表示未见轨迹。

模仿学习方法——时间对比网络（TCN），利用无标签的多视角视频，帮助智能体学习关于任务的内在联合不变表征。

除了使用对抗结构解决次优示范问题外，研究界也在探索其他解决方案。从次优示范中外推奖励是一种合理的解决方案。外推是指利用附加信息或奖励信号，帮助模型建立次优示范与预期目标之间的联系。图 1 展示了外推的图解。与对抗结构化模仿学习相比，外推所需的环境交互更少，结构也更加稳定，但添加有意义的信息可能既费力又昂贵。Palan 等人，2019年提出了 DemPref，该方法利用人类的在线偏好来解决真实世界的机械臂场景中的示范质量问题。同样，Brown 等人，2019年指出，预先对数据集进行排序有助于智能体推断出优于示范者的行为。提出的 T-REX 将逐渐改进的演示数据集或人工排序的数据集作为标准逆向强化学习的输入，并在次优场景中优于最先进的方法。其后续方法 D-REX（Brown 等人，2020年）通过注入不同程度的噪声，省去了费力的流程并实现了排序。然而，噪声并不总是会损害智能体性能。如（Brown 等人，2020年）所示，在轨迹中引入适当的随机步骤有助于提升泛化能力，并防止对训练数据集分布过拟合。在这种情况下，本文更灵活地使用随机策略，在实现自动排序的同时帮助提升策略性能。

另一方面，本文也借鉴了遗传算法（GA）的思想。遗传算法是一种自适应随机优化算法，是进化算法的一种（Karakatič 和 Podgorelec，2015年）。遗传算法最初由（De Jong，1975年）提出，随后由（Holland，1992年；Goldberg，2006年）进一步改进，用于解决复杂的优化问题。研究界已在博弈策略学习（Eliya 和 Herrmann，2020年）等领域开展了将进化算法与模仿学习相结合的多项研究，

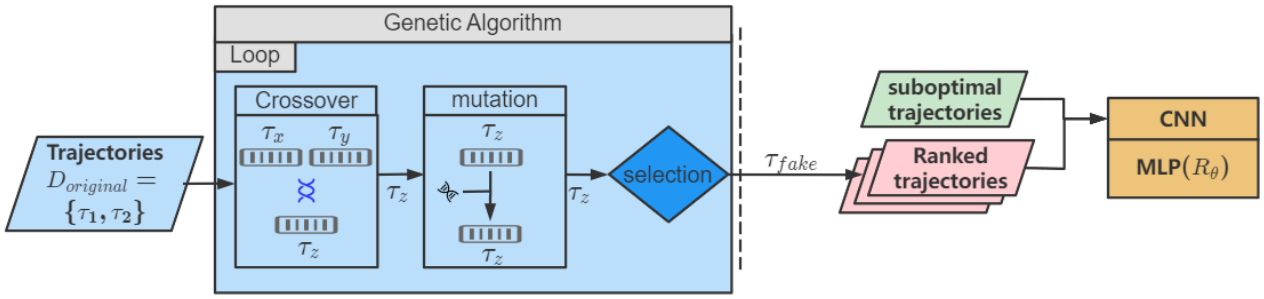


图 2：我们方法的示意图。次优示范 $\mathcal{D}$ 首先作为遗传算法部分中的个体染色体，用于生成“伪”轨迹 $\tau_{fake}$，随后将其与生成的数据集一同输入卷积神经网络。卷积神经网络提取的特征用于训练由多层感知机表示的奖励参数 $R_\theta$。

以及机器人运动（Park等，2008年）和自动驾驶（Jalali 等，2019年）。这些研究利用进化算法来实现离线数据集扩充或优化网络参数。遗传算法的过程模仿基因的生成与进化方式，以实现高效搜索和优化。在现实世界中，基因交换相对稀疏，只会在产生新世代时发生。遗传算法遵循生物学的三项基本原则：繁殖、自然选择和个体多样性（达尔文，1859年），同时跳过其他生物活动。它将问题的不同特征表示为染色体或基因型，并通过遗传算子迭代地产生新的个体。例如，交叉会交换两条染色体上的基因并生成新的后代，而变异则会改变染色体上的基因片段。每个个体代表一种可能的解，并参与自然选择过程，其表现由适应度函数评估。适应度函数会筛除较弱的个体，从而实现优化。除了需要合适的适应度函数外，要高效地获得最优解，个体多样性也是一个关键影响因素。当个体的特征分布较为分散时，遗传算法更有可能有效地找到最优全局解。

在这项工作中，我们将遗传算法引入模仿学习，并提出一种名为 GenIL 的方法。GenIL 对次优轨迹应用遗传算子——交叉和变异——生成具有不同回报水平的新样本，以外推奖励函数参数。

## 3 问题定义

在这项工作中，我们假设 GenIL 是在马尔可夫决策过程（MDP）框架下建模的。MDP 是一种满足如下性质的过程：在任意时刻 $t$，下一状态 $s_{t+1}$ 只取决于当前状态 $s_t$。通常，MDP 定义为一个元组 $(\mathcal{S},\mathcal{A},\mathcal{P},\gamma,\mathcal{D},\mathcal{R})$，其中 $\mathcal{S}$ 是有限状态集，$\mathcal{A}$ 是相应的动作集，$\mathcal{P}$是状态转移概率集，后继状态 $s_{t+1}$ 从该转移模型中抽取，即 $s_{t+1} = P(\cdot|s_t, a_t)$，$\gamma \in [1, 0)$ 是折扣因子，$\mathcal{D}$ 是示范轨迹 $\mathcal{D}_n = \{\tau_1, \tau_2, ..., \tau_n\}$，而 $\mathcal{R}$ 是奖励函数 $\mathcal{S} \mapsto \mathcal{R}$。轨迹 $\tau_x$ 的期望累积奖励为 $G(\tau_x) = \sum_{t=0}^{T} \gamma^t R(s^t)$。

与其他模仿学习方法类似，我们假设无法获取最优奖励函数。输入示范 $\mathcal{D}_{original}$ 由两条轨迹组成：一条良好（或近乎最优）的轨迹 $\tau_{good}$ 和一条次优示范 $\tau_{bad}$。在这里，我们希望近似一个奖励函数参数 $\theta$，使其尽可能准确地反映预期的最优策略。为表示轨迹的优劣，我们提出用一串相同的数值替代原始步奖励序列，即 $R'(S) = \{rank\}^m$，其中 $\sum R'_{\tau_{good}}(S) > \sum R'_{\tau_{bad}}(S)$。在 T-REX（Brown 等，2019年）等先前工作中，秩由专家手动指定。排序通过监督学习作为训练标签。本工作沿用先前的设定，同时免去人类专家繁琐的排序工作，改由遗传算法生成排序。通过执行交叉和变异步骤，原始轨迹相互混合并生成伪轨迹$\tau_{fake}$，其伪步奖励 $\sum R'_{\tau_{good}}(S) > \sum R'_{\tau_{fake}}(S) > \sum R'_{\tau_{bad}}(S)$具有不同的数值。根据累积伪步奖励，这些伪轨迹被划分为不同的秩，并与原始数据集共同构成排序后的输入。基于这个组合数据集训练的网络预期能够对包含更多良好示范片段的轨迹预测出更高的回报，即$\mathbb{E}_\pi[\sum \gamma R_\theta(\tau_a)] > \mathbb{E}_\pi[\sum \gamma R_\theta(\tau_b)]$，若 $R'(\tau_a) > R'(\tau_b)$。

## 4 方法

如引言部分所述，GenIL 倡导采用 IfO 设定，同时利用遗传算法。我们假设遗传算法能够为奖励推断实现有意义的排序。与其他 IfO 方法一样，GenIL 使用状态信息来训练模型。给定一个由两条表现各异的轨迹 $\mathcal{D}_{original} = \{\tau_1, \tau_2\}$ 组成的演示数据集，目标是训练一个由神经

**算法 1**GenIL

**输入**：示范 $\mathcal{D}$、后代数量 K、变异率 $p_{mut}$、交叉率 $p_{crx}$
**输出**：估计的奖励参数 $R_\theta$

```
随机初始化奖励参数 R_θ 。
使用初始排名D = {重新标注演示数据(τ_1, Rank_1), (τ_2, Rank_2)}。
当后代数量 < K 时
    τ_x, Rank_x, τ_y, Rank_y = 选择父代(D)
    后代 τ_fk, Rank_fk = 交叉(p_crx, τ_x, Rank_x, τ_y, Rank_y)
    后代 τ_fk, Rank_fk = 变异(p_mut, τ_fk, Rank_fk)
    如果 Rank_fk 满足选择规则 则
        将后代 (τ_fk, Rank_fk) 添加到 D_rank
        D → D ∪ D_rank
    结束如果
结束 while 循环
运行 T-REX 以获得 R_θ
使用奖励参数 R_θ通过强化学习获得策略 π_θ 。
```

网络构成的奖励推断模型，使其能够从未见过的轨迹对中区分表现更好的轨迹，即当$\sum R'(\tau_x) > \sum R'(\tau_y)$时，$\sum_{s\in\tau_x}\mathcal{R}_\theta(s) < \sum_{s\in\tau_y}\mathcal{R}_\theta(s)$。研究证明，要恢复正确的奖励函数，必须引入额外信息（Brown 等，2020年）。许多先前的研究（Palan 等，2019年；Brown 等，2019年；Castro 等，2019年）要么利用人类偏好，要么注入噪声，以提供排序所需的额外信息。对样本进行适当排序可视为序数回归问题的一个实例，从而减少歧义（Castro 等，2019年；Brown 等，2020年）。本文沿用以往的框架，并提出一种新颖的学习范式 GenIL：利用遗传算法作为额外信息来实现排序，并辅助奖励推断。图 2 展示了 GenIL 的工作流程。

GenIL 由两个部分组成。(1) 遗传算法部分对从数据集中随机采样的两条轨迹执行交叉和变异步骤，并输出一条“伪造”轨迹，即 $\tau_{fake} = \{s_i | s_i \in \tau_1 \cup \tau_2 \cup S_{mut}\}$ 和

$$\sum_{s_i\in\tau_{fake}} R'(s_i) = \sum_{s_j\in S_j\subset\tau_1} R'(s_j) + \sum_{s_k\in S_k\subset\tau_2} R'(s_k) + \sum_{s_n\in S_{mut}} R'(s_n)$$

其中，$S_{mut}$ 是可用的变异样本集，变异集中的状态是从所有已访问状态中批量采样得到的。同时，为了保持状态之间的随机性和连续性，我们将交叉步长设定在小于 10 的随机区间内，而变异步骤会在随机状态上提供随机排名。至于选择部分，我们采用简单的选择标准：如果生成的后代的平均排名处于预定义区间内，则将该后代添加到相应的排序数据集中。选择步骤完成后，排序数据集与原始数据集合并为已排序数据集，即$D_{ranked} = D_{original} + D_{fake}$。在实验中，我们将生成后代的数量视为超参数，并通过反复试验将其设为 12。在生成的数据集上应用遗传算法，可以更充分地利用数据，并在数据效率方面优于先前的方法。(2) 逆强化学习部分从生成的已排序数据集中学习奖励推断模型。为了从帧中提取特征，我们首先实现了一个四层卷积神经网络，随后通过多层感知机（多层感知机）将其映射为参考奖励。该网络以轨迹为输入，并输出给定轨迹的预期奖励。与先前的工作类似，该模型采用监督学习方式进行训练，使用改编自（Brown等，2019年）的成对排序损失：

$$\mathcal{L}(\theta) \approx -\sum_{\tau_i,\tau_j} \log \frac{\exp\sum_{s\in\tau_j} R_\theta(s)}{\exp\sum_{s\in\tau_i} R_\theta(s) + \exp\sum_{s\in\tau_j} R_\theta(s)}$$

其中 $\sum_{s\in\tau_j} R'(s) > \sum_{s\in\tau_i} R'(s)$。该损失函数采用 Plackett-Luce 模型的形式（Plackett，1975年；Luce，2012年），该模型已被证明能够有效地通过神经网络从排序实例中学习模型（Christiano 等，2017年；Mohlin 等，2020年）。所得奖励函数参数 $\theta$ 可进一步供其他强化学习算法使用，以获得策略。GenIL 的算法描述见算法 1。

## 5 实验

上述算法 GenIL 在从观察中模仿的范式下实现了一种基本的遗传算法。我们假设，遗传算法的实现能够更好地利用次优数据，在实现数据效率和鲁棒性的同时取得有竞争力的性能。本节将介绍用于验证这一假设的实验设计，并讨论实验结果。

### 5.1 设置

我们在图形处理器 NVIDIA Quadro RTX 5000 上，使用五个常见基准对 GenIL 进行评估。其中包括 MuJoCo 模拟器（Todorov 等，2012年）在 OpenAI Gym（Brockman 等，2016年）中的机器人任务：Hopper、HalfCheetah；以及三个 Atari 任务：Breakout、Beamrider 和 SpaceInvaders。

我们采用一个三层神经网络来表示奖励参数，每层包含 256 个神经元并使用 ReLU 激活函数。对于 Mujoco任务，我们直接使用模拟器提供的状态信息来训练模型。而对于 Atari任务，我们使用原始帧作为输入，因此在

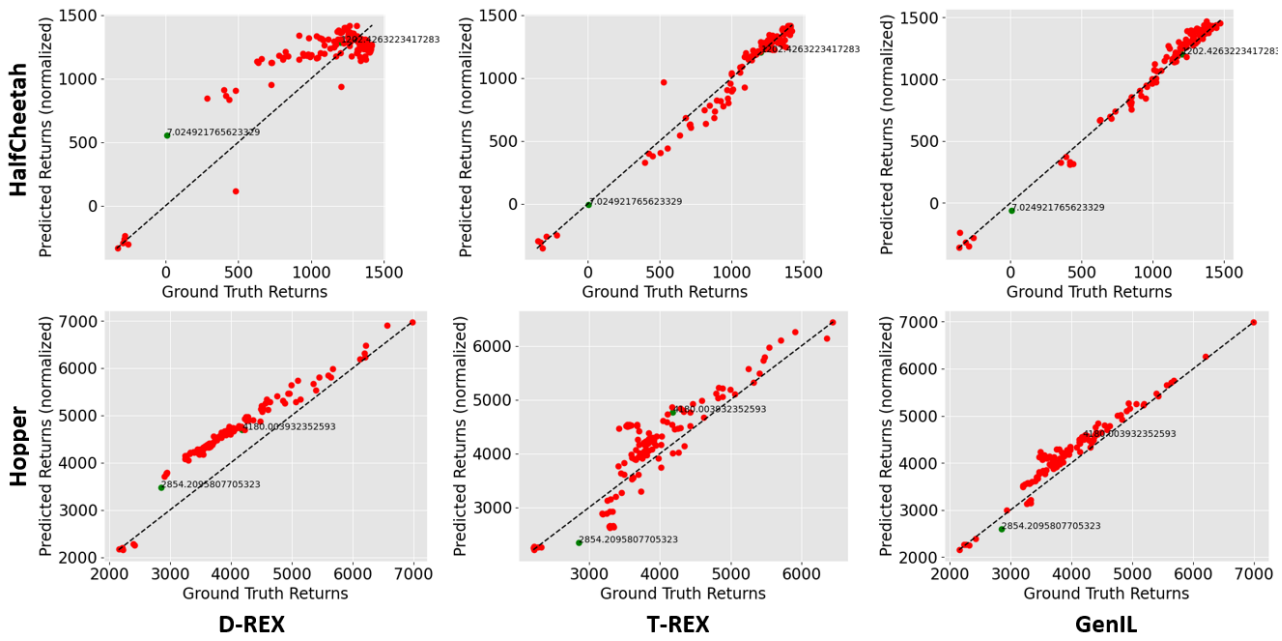


图3：外推图形比较。绿色点是用于训练的数据，红色点是用于预测的未见轨迹。预测回报和真实回报被归一化到相同范围内，以便进行比较。

全连接层前添加一个四层卷积神经网络，每层使用 Leaky-ReLU 激活函数。为避免“因果混淆”问题（de Haan 等，2019年）导致模型与无关特征建立错误关联，我们会屏蔽分数和剩余寿命等指标部分。

参考近期方法，我们使用由 OpenAI Baselines（Dhariwal 等，2017年）实现的近端策略优化（PPO）（Schulman 等，2017年），采用默认超参数和奖励函数生成数据集。PPO 训练每进行 20 个训练步骤便保存一个检查点。为了体现智能体性能的差异，我们恢复两个智能体：一个来自训练早期阶段的检查点，另一个来自接近训练结束阶段的检查点。对于 Atari任务，我们使用检查点 400 和 1000；而在 MuJoCo 领域，我们选择检查点 40 和 220 作为输入。记录恢复的智能体与任务环境之间的交互，形成轨迹。为确保评估公平，记录初始轨迹，并将其作为其他基线方法的输入。结合遗传算法生成的轨迹，将这些轨迹重新采样为约 5000 个片段，长度各不相同，范围为 100 至 300，然后输入网络。对于遗传算法的超参数，我们将固定的变异率和交叉率分别设为 0.05 和 0.9。为了从外推得到的奖励参数中获取策略，我们在 MuJoCo 领域对策略进行 480 个训练步骤的训练；而在 Atari 领域，我们使用 3000 个训练步骤评估策略。

我们比较了 GenIL 与行为克隆、T-REX 和 D-REX 等其他方法在 20 次试验中的平均表现，每次试验使用不同的随机种子训练五个模型。具体而言，由于利用外推为模仿学习学习奖励函数的相关研究相对较少，我们从准确率和鲁棒性等方面比较了外推性能与 T-REX。

## 5.2 结果

我们假设，借助基础遗传算法，GenIL 可在以下两个方面实现更好的外推：（1）未见轨迹上的外推准确率，我们使用归一化预测回报与真实值的比率，即 $average\left(\frac{R_\theta(\tau_t)}{R^*(\tau_t)}\right)$，来衡量。（2）训练后的智能体在目标领域中的整体表现，该表现通过模拟获得的真实值回报来衡量。

图3展示了在 MuJoCo 领域中，所提出的 GenIL 与先前的外推算法 T-REX 和 D-REX 之间的外推效果对比。绿色点表示由偏离的检查点生成的两条训练轨迹。红色点表示可能更好甚至更差的未见轨迹，我们旨在为这些未见轨迹外推出紧凑的预测奖励。x 轴表示不同演示水平对应的真实回报，y 轴表示外推模型预测的回报。为方便比较，期望回报和真实回报被缩放至相同范围。虚线用于表示预测回报与真实回报之间的偏差。

表1：GenIL 与先前方法 T-REX 的策略性能比较。表中数值是通过使用不同随机种子的 20 次试验得出的。

| 任务 | GenIL | | T-REX | | D-REX | | BC | |
|---|---|---|---|---|---|---|---|---|
| | Avg | Std | Avg | Std | Avg | Std | Avg | Std |
| **HalfCheetah** | **1765.9** | 383.7 | 1397.3 | 408.7 | 378.4 | 542.1 | -364.3 | 2.0 |
| **Hopper** | **1693.0** | 657.3 | 1342.1 | 473.8 | 482.1 | 315.2 | 800.6 | 30.9 |
| **Beamrider** | **927.6** | 647.2 | 825.4 | 265.5 | 843.7 | 176.3 | 477.2 | 155.5 |
| **Breakout** | **17.5** | 6.3 | 14.1 | 5.9 | 11.8 | 2.9 | 2.1 | 4.1 |
| **SpaceInvaders** | **478.1** | 192.5 | 364.3 | 151.2 | 299.3 | 87.6 | 127.3 | 85.4 |

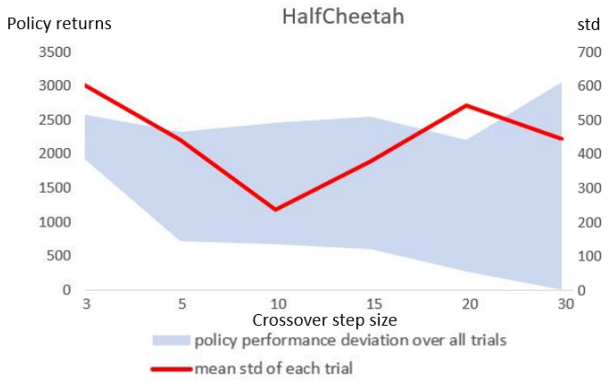


图4：策略性能随交叉步长变化的情况。蓝色区域表示在不同随机种子下，每种交叉步长进行10次试验所得到的策略性能。红线表示每次试验中使用相同交叉步长的五个训练模型的性能标准差平均值。

从图3可以看出，与 Hopper 相比，这三种外推方法在 HalfCheetah 上的表现都更好，预测值也更接近实际值。在 HalfCheetah 中，未见轨迹密集分布在图的右上方，这表明原始 PPO 策略逐渐收敛至 1500 的真实奖励。相比之下，在 Hopper 中，大多数未见轨迹集中在 4000 的真实奖励附近。至于不同方法之间的比较，对于真实奖励相近的数据点，GenIL 的预测垂直偏差较小，这意味着 GenIL 能够利用生成的虚假轨迹，为未见轨迹做出更紧凑、更稳健的预测。

表1比较了 GenIL 与以往方法的策略性能。表中的数值是根据使用不同随机种子的 20 次试验得出的。从表1可以看出，GenIL 在所有任务中的表现都优于以往方法。与以往使用有限示范的方法相比，所得策略在不同任务中均保持了有竞争力的性能。输入数量似乎对 GenIL 和 T-REX 的影响较小。相比之下，示范质量严重限制了行为克隆和 D-REX。行为克隆无法在大多数任务中学到有意义的策略，这些示范为行为克隆的性能设定了上限。与 T-REX 和 GenIL 相比，D-REX 性能的标准差最小，这表明 D-REX 比 T-REX 和 GenIL 更稳定。嵌入式BC过程会在每个排名中生成足够的示范，从而使生成的策略更加稳定。然而，行为克隆过程限制了其性能。这或许可以解释为何行为克隆和 D-REX 会同时在某些任务中表现不佳。比较两类任务的性能可知，GenIL 在离散型 Atari任务中的性能偏差远大于其在连续型 Mujoco任务中的偏差。

对于Mujoco任务，行为克隆（BC）未能在 HalfCheetah中学到有意义的策略，因为次优示范可能不足以建立状态和动作之间的可执行映射。同时，BC的性能也受到“累积误差”（Ross和Bagnell，2010）的限制，因此学到的策略无法从未见状态中恢复。D-REX也受到BC局限性的影响，未能外推出良好的奖励参数。不过，大量由BC生成的轨迹所产生的影响表明，性能偏差显著取决于训练样本的数量，即使这些样本远非最优。T-REX在HalfCheetah和Hopper中取得了良好的策略性能，略优于次优示范，但在所有评估方法中，其策略性能偏差最大。从表1可以看出，与其他评估方法相比，GenIL在HalfCheetah和Hopper中也取得了更好的性能；尤其是在HalfCheetah中，GenIL的表现大幅超过示范和T-REX，这与图3的结果一致：GenIL对未见数据的奖励预测更紧凑、更稳定。

在整个实验过程中，我们发现遗传算法的内在参数会影响学习性能。例如，当初始输入轨迹之间的差异显著时，增加秩的数量可以实现更准确的外推、更好的策略性能以及更小的方差。然而，当初始输入轨迹之间的差异较小时，较少的秩也能取得良好性能，同时节省寻找末端秩候选项的时间。生成的伪轨迹数量对外推同样起着关键作用。在训练中使用更多伪后代时，未见数据的预测结果聚集得更紧密，这与 D-REX 呈现出相似的特征。GenIL 中另一个重要参数可能是交叉步长与子采样规模之比。

为确保交叉有效，我们将交叉步长设为小于 10，而子采样规模约为 200。在这种情况下，交叉后伪轨迹中会包含足够多来自初始轨迹的小片段；事实上，来自优质初始轨迹的小片段数量成为衡量伪轨迹强度的指标。以 HalfCheetah 领域为例，我们评估了不同交叉步长下的策略性能（见图4）。

在图4中，x 轴表示不同的交叉步长，左侧 y 轴表示策略性能，右侧 y 轴反映每次试验中五个模型的标准差。根据图4，我们可以看到，交叉步长对策略性能上限没有明显影响，但会显著影响策略性能的一致性。当交叉步长相对于子采样大小过大时，交叉效果会显著减弱，秩之间的差异也会变得更加模糊。从每次试验的策略性能和标准差可以看出，交叉步长过大时，训练出的智能体差异更为显著，行为也更加不稳定。另一方面，当交叉步长过小时，不同试验间的策略性能会更加一致，但每次试验的标准差会增大。我们推测，这一问题的原因在于无法很好地保持动态转移的一致性，从而导致虚假轨迹出现碎片化。

## 6 局限性

在模仿学习中更好地利用遗传算法仍面临一些挑战。一个重要问题是需要进行更动态的超参数选择。遗传算法的引入为模仿学习增加了更多超参数。在本研究中，超参数是通过反复试验选定的。未来还需进一步研究，以实现更好的优化。动态超参数选择可以避免密集的交叉操作和变异操作导致生成的轨迹支离破碎且随机化。如何平衡遗传算法在模仿学习中带来的泛化与碎片化，可能是一个值得进一步研究的有趣课题。由于 GenIL 对一条轨迹中的所有观测采用相同的排序，这种做法可能意义不大。未来研究可以利用观测访问频率或动作频率等统计数据来区分不同片段，并提高更有意义部分的权重。未来工作还可以在 GenIL 中采用更先进的选择规则和学习范式，以提升推断效果和整体性能。

## 7 结论

本文介绍了 GenIL，这是一种结合遗传算法与逆向强化学习的奖励外推方法。GenIL 利用交叉和变异，重现示范从远非最优逐步改进的过程，并推断奖励通过逆强化学习，从由此产生的排序数据集中获得函数参数。这种外推技术使我们能够更好地利用次优示范，并免去专家繁琐的偏好流程。实验表明，GenIL 不仅在外推过程中波动较小，而且整体策略表现也优于以往方法。

## 参考文献


Marcin Andrychowicz、Filip Wolski、Alex Ray、Jonas Schneider、Rachel Fong、Peter Welinder、Bob McGrew、Josh Tobin、OpenAI 的 Pieter Abbeel 和 Wojciech Zaremba。事后经验回放。载于神经信息处理系统进展，第 5048–5058 页，2017 年。

YusufAytar、Tobias Pfaff、David Budden、Tom Le Paine、Ziyu Wang 和 Nando de Freitas。通过观看 YouTube 玩转高难度探索游戏。*arXiv* 预印本 *arXiv:1805.11592*，2018年。

Michael Bain 和 Claude Sammut。行为克隆框架。载于《机器智能》第*15*卷，第103–129页。牛津大学出版社，1999年。

Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 Wojciech Zaremba。OpenAI Gym。*arXiv* 预印本 *arXiv:1606.01540*，2016年。

Daniel Brown、Wonjoon Goo、Prabhat Nagarajan 和 Scott Niekum。通过基于观测的逆向强化学习，从次优示范中进行外推。载于国际机器学习大会，第783–792页。机器学习研究论文集，2019年。

Daniel S Brown、Wonjoon Goo 和 Scott Niekum。通过自动排序的示范实现优于示范者的模仿学习。载于机器人学习大会，第330–359页。机器学习研究论文集，2020年。

Pablo Samuel Castro、Shijian Li 和 Daqing Zhang。利用多个排序专家进行逆向强化学习。*arXiv*预印本 *arXiv:1907.13411*，2019年。

Kai-Wei Chang、Akshay Krishnamurthy、Alekh Agarwal、Hal Daum´e III 和 John Langford。学习如何比老师搜索得更好。载于国际机器学习大会，第2058–2066页。机器学习研究论文集，2015年。Paul Christiano、Jan Leike、Tom B Brown、Miljan Mar- tic、Shane Legg 和 Dario Amodei。基于人类偏好的深度强化学习。*arXiv*预印本 *arXiv:1706.03741*，2017年。Felipe Codevilla、Matthias M¨uller、Antonio L´opez、Vladlen Koltun 和 Alexey Dosovitskiy。通过条件模仿学习实现端到端驾驶。载于*2018*年*IEEE*国际机器人与自动化会议（*ICRA*），第4693–4700页。IEEE，2018年。查尔斯·达尔文。《通过自然选择的物种起源》。Pub One Info，1859年。

Pim de Haan、Dinesh Jayaraman 和 Sergey Levine。模仿学习中的因果混淆。神经信息处理系统进展，32:11698–11709，2019年。

Kenneth Alan De Jong。 对一类遗传适应系统行为的分析。密歇根大学，1975年。

Prafulla Dhariwal、Christopher Hesse、Oleg Klimov、Alex Nichol、MatthiasPlappert、AlecRadford、JohnSchulman、Szymon Sidor、Yuhuai Wu 和 Peter Zhokhov。《OpenAI Baselines》，2017年。

Yiming Ding、Carlos Florensa、Mariano Phielipp 和 Pieter Abbeel。目标条件模仿学习。*arXiv* 预印本 *arXiv:1906.05838*，2019年。Yan Duan、Marcin Andrychowicz、Bradly C Stadie、Jonathan Ho、Jonas Schneider、Ilya Sutskever、Pieter Abbeel 和 Wojciech Zaremba。单样本模仿学习。*arXiv* 预印本 *arXiv:1703.07326*，2017年。Roy Eliya 和 J Michael Herrmann。进化式选择性模仿：无需示范者的模仿学习，打造可解释智能体。*arXiv* 预印本 *arXiv:2009.08403*，2020年。David E Goldberg。遗传算法。Pearson Education India，2006年。

Jonathan Ho 和 Stefano Ermon。生成对抗式模仿学习。神经信息处理系统进展，29:4565–4573，2016年。

John H Holland。 自然系统与人工系统中的适应：一项关于生物学、控制和人工智能应用的导论性分析。麻省理工学院出版社，1992年。

Seyed Mohammad Jafar Jalali、Parham M Kebria、Abbas Khosravi、Khaled Saleh、Darius Nahavandi 和 Saeid Nahavandi。通过深度模仿学习和神经进化实现最优自动驾驶。载于*2019*年 *IEEE* 国际系统、人类与控制论会议（*SMC*），第1215–1220页。IEEE，2019年。

Saˇso Karakatiˇc 和 Vili Podgorelec。用于求解多配送中心车辆路径问题的遗传算法综述。《应用软计算》，27:519–532，2015年。

YuXuan Liu、Abhishek Gupta、Pieter Abbeel 和 Sergey Levine。从观察中模仿：通过上下文转换学习从原始视频中模仿行为。发表于*2018*年 *IEEE* 国际机器人与自动化会议（*ICRA*），第 1118–1125 页。IEEE，2018年。

Naijun Liu、Tao Lu、Yinghao Cai、Boyao Li 和 Shuo Wang。后见生成对抗模仿学习。*arXiv* 预印本 *arXiv:1903.07854*，2019年。

R Duncan Luce。 个体选择行为：理论分析。Courier Corporation，2012年。

¨Erik Mohlin、Robert Ostling 和 Joseph Tao-yi Wang。在赢家通吃博弈中通过相似度加权模仿进行学习。《博弈与经济行为》，120:225–245，2020年。

Malayandi Palan、Nicholas C Landolfi、Gleb Shevchuk 和 Dorsa Sadigh。通过整合人类示范与偏好来学习奖励函数。*arXiv*预印本 *arXiv:1906.08928*，2019年。GaLam Park、Syungkwon Ra、ChangHwan Kim 和 JaeBok Song。使用遗传算法进行机器人运动模仿学习。*IFAC*会议论文集，41(2):730–735，2008年。Robin L Plackett。排列分析。《英国皇家统计学会杂志：*C*辑（应用统计）》，24(2):193–202，1975年。

St´ephane Ross 和 Drew Bagnell。模仿学习的高效归约。载于第十三届人工智能与统计国际会议论文集，第661–668页。JMLR研讨会与会议论文集，2010年。Stuart Russell。面向不确定环境的学习型智能体。载于第十一届计算学习理论年会论文集，第101–103页，1998年。John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。*arXiv*预印本 *arXiv:1707.06347*，2017年。Pierre Sermanet、Corey Lynch、Yevgen Chebotar、Jasmine Hsu、Eric Jang、Stefan Schaal、Sergey Levine 和 Google Brain。时间对比网络：从视频中进行自监督学习。载于*2018*年*IEEE*国际机器人与自动化会议（*ICRA*），第1134–1141页。IEEE，2018年。Emanuel Todorov、Tom Erez 和 Yuval Tassa。MuJoCo：用于基于模型的控制的物理引擎。载于*2012*年*IEEE/RSJ*智能机器人与系统国际会议，第5026–5033页。IEEE，2012年。Faraz Torabi、Garrett Warnell 和 Peter Stone。利用本体感觉从视频中进行模仿学习。*arXiv*预印本 *arXiv:1905.09335*，2019年。Boyuan Zheng、Sunny Verma、Jianlong Zhou、Ivor Tsang 和 Fang Chen。模仿学习：进展、分类体系与机遇。*arXiv*预印本 *arXiv:2106.12177*，2021年。Zhuangdi Zhu、Kaixiang Lin、Bo Dai 和 Jiayu Zhou。基于观测的离策略模仿学习。*arXiv*预印本 *arXiv:2102.13185*，2021年。Guoyu Zuo、Qishen Zhao、Kexin Chen、Jiangeng Li 和 Daoxiong Gong。针对低质量示范机器人任务的离策略对抗模仿学习。《应用软计算》，97:106795，2020年。