模仿综述笔记

> Imitation Learning: Progress, Taxonomies and Opportunities

1.模仿学习的简单分类：

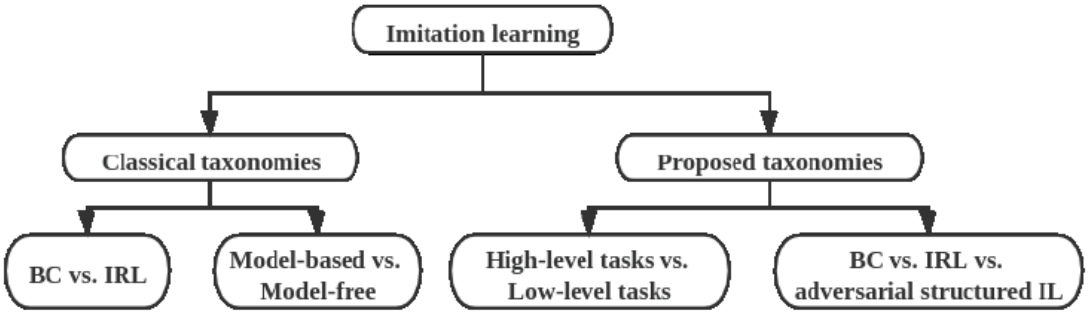

Fig. 2. Taxonomies in this review.

Table 1. Categorization of IL: BC vs. IRL

Classes	Examples and Publications
Behavioural Cloning	Few-shots learning[18]
	Input optimization[13]
	Latent policy learning[42]
	Real-world application[79]
Inverse Reinforcement Learning	Improving efficiency[9]
	Raw video as inputs[61]
	Adversarial structured[66]
	Sparse reward problem[44]

Table 2. Categorization of IL: Model-based vs. Model-free

Classes	Examples and Publications
Model-based IL	Forward model[19, 21]
	Inverse model[43]
Model-free IL	BC method[42]
	Reward engineering[9]
	Adversarial style[70]

2.一些框架

2.1 行为克隆vs逆强化学习

- **行为克隆**：
    1. 尝试学习得到状态->动作对的mapping，希望策略与数据中的mapping差距尽可能小。
    2. 在现实生活问题上得到应用。
    3. 成本低，实现简单，只需要学到(s,a)的mapping。
    4. 最近的BC研究(见表1)：
        1. 与元学习相结合，即：希望从更广泛的行为中预训练来学习。
        2. 与VR设备等其他技术结合。
- **逆强化学习**：
    1. 尝试从专家演示的数据中恢复一个奖励函数，从而转变为RL问题求解。
    2. 在虚拟环境上模拟算法性能。
    3. 需要的算力和技术更强，因为需要恢复得到唯一的奖励函数。
    4. 最近的IRL研究(见表1)：
        1. 用其他方法或问题设置扩展 GAIL[17]；
        2. 从原始视频中恢复奖励函数[5]；
        3. 通过使用强化学习的当前发展，如 TRPO[60] 和 HER[2]，开发更有效的基于模型的 IRL 方法。

2.2 Model-Base vs Model Free

另一种经典分类法将 IL 分为基于模型的方法和无模型的方法。 这两类之间的主要区别在于算法是否采用前向模型从环境上下文/动态中学习尽管从环境中学习听起来对各种方法都有好处，但对于给定的问题设置可能没有必要或不切实际。整合环境上下文/动态可以获得更多有用的信息，从而使算法能够实现数据效率和可行性，而缺点是学习模型成本高昂且具有挑战性。 例如，在机器人技术中，设备通常是精确的，空间位置、速度等参数很容易获得，系统动力学可能对再现行为的帮助相对较小。 另一方面，在自动驾驶汽车任务中，系统动力学对于避免撞到行人可能至关重要。 在这种情况下，选择无模型还是基于模型取决于任务。 表 2 列出了 IL 中最近的一些研究主题，分为基于模型和无模型。

2.3 Low-Level Tasks vs. High-Level Tasks

本小节介绍了一种新的分类法，它根据评估方法将 IL 分为操作任务和高级任务。模仿学习中并没有被广泛承认的benchmark，所以这种分类法可以告诉我们不同算法在应用场景上的倾向性。图3展示了IL上的一些经典benchmark。

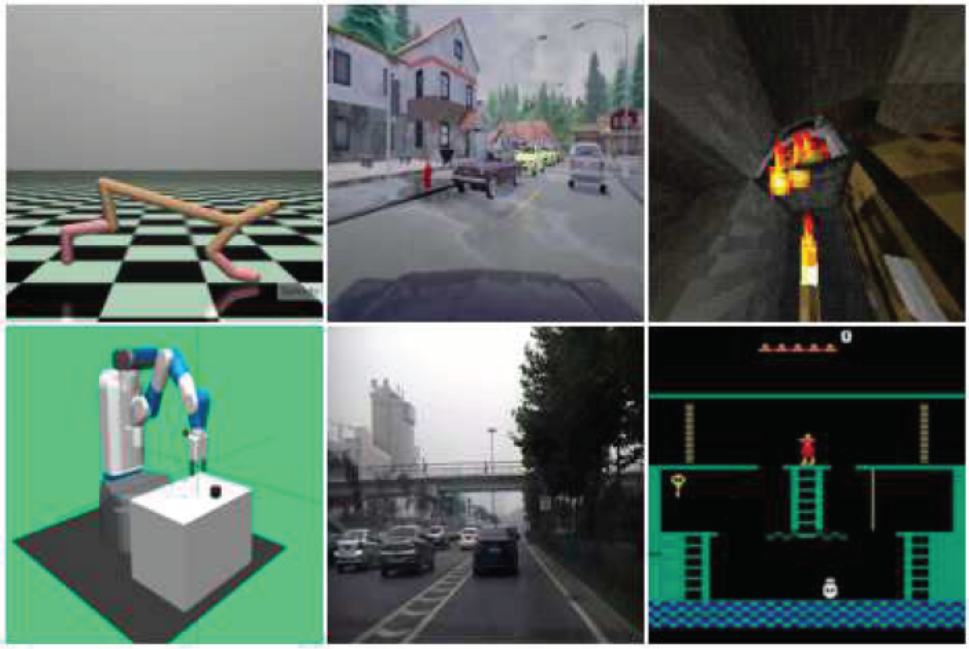

Fig. 3. Prevalent Tasks in IL. Top-left: HalfCheetah in Mujoco; Top-mid: CARLA simulator; Top-right: Minecraft scenario in MineRL dataset; Bottom-left: FetchPickAndPlace-v1 in OpenAI Gym; Bottom-mid: Driving scenario in Xi'an [80]; Bottom-right: Atari game–MontezumaRevenge

- **低层次任务：**
  1. 机械臂控制、机器人控制等。
  2. 低层次任务更多是复现底层的操作行为、例如移动手臂的距离、角度等。
- **高层次任务：**
  1. 自动驾驶、控制Agent玩游戏等。
  2. 高层次任务更多是规划动作的执行顺序，同时假定底层的动作能被正确执行，例如：自动驾驶任务中需要给定油门大小，方向盘角度，而不需要复现车的底层是如何加速、转向的。

  虽然一些算法都会在这两类任务上做实验，但是可以明显发现这两类任务的数量是不均衡的，也就是说算法在不同层次任务的表现是不同的，所以这种分类有助于提醒我们算法的目标域。

表 3 列出了该分类法下的一些最新研究。 当前的大多数模仿方法倾向于使用低级操作任务来评估所提出的方法，因为强化学习在游戏等高级控制器任务中的表现可以接受，而在奖励函数可能无法获得的低级操作任务上通常表现不佳。然而，高层次任务对IL来说非常困难，因为在诸如3D任务等状态和动作空间较大的任务中，IL需要额外大量的时间来处理大型的状态动作空间。

Table 3. Categorization of IL: Low-level Tasks vs. High-level Tasks

Classes	Examples and Publications
Low-level manipulation	Surgical assistance[49, 68]
	Vehicle manipulation[80]
	Robotic arm[61]
	VR teleoperation[79]
High-level tasks	2D gameplay[59]
	3D gameplay[4]
	Navigation[28]
	Sports analysis[78]

2.4 BC vs. IRL vs. Adversarial Structured IL

前面提到模仿可以分成BC和IRL两类，近年的GAIL算法的提出使得模仿增加了一类新的算法：对抗模仿学习。尽管ASIL和IRL密切相关，当ASIL并没有从专家数据中恢复奖励函数，因此ASIL被设置成独立于BC和IRL两类之外的算法。图5展示了这三种算法之间的区别：

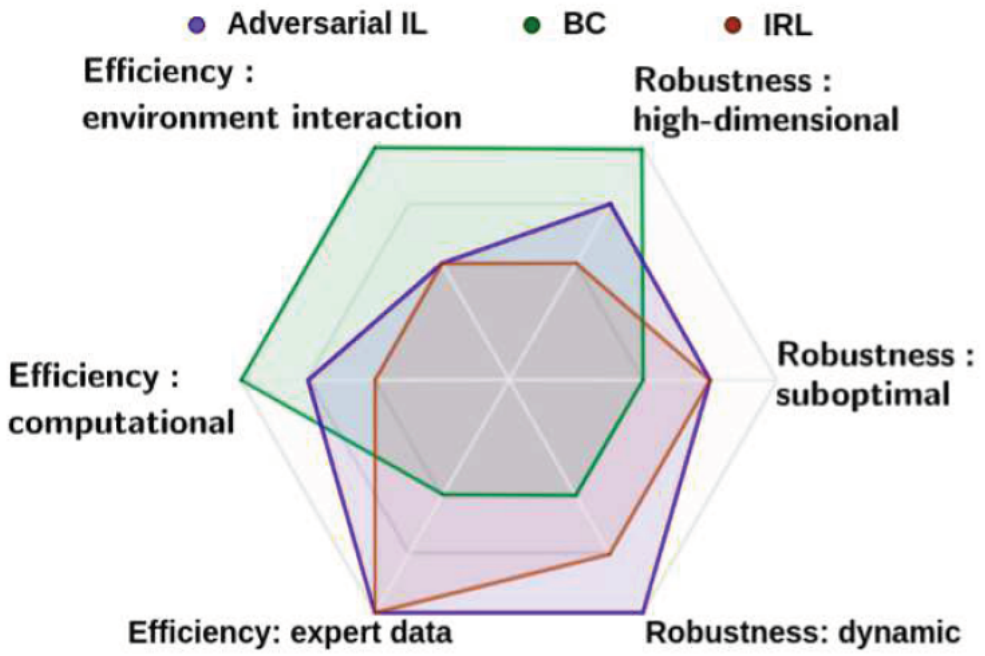

Fig. 5. Web plot for taxonomy: BC vs. IRL vs. Adversarial Structured IL. We collected 6 popular evaluation criteria from the research and empirically ranked them into three levels based on research consensus. The outer the point, the higher the ranking, which means that it scores higher in the evaluation from the empirical perspective.

在鲁棒性方面，主要关注高维空间的鲁棒性，演示次优时的鲁棒性（包括演示中对噪声的考虑），以及动态系统的鲁棒性。 BCmethods通常在高维空间具有更好的性能，因此它们在机器人上得到了广泛的评估，而在动态环境和次优数据集下的性能有限； IRL 方法优化其内循环中的参数，这成为限制其在高维空间中性能的负担，但恢复的奖励函数将有利于智能体在动态系统中进行预测。 由于对抗性结构化 IL 方法通常源自 GAIL，因此它们继承了 GAIL 的优点：在高维空间和分布发生变化时的鲁棒性。 由于最近在 IRL 和 Adversarial 结构化 IL 中的研究在次优演示问题上取得了进展，所以在次优演示的鲁棒性评估中给予它们相同的排名。

3.主要研究主题和方法

3.1行为克隆

行为克隆通过利用专家/oracle提供的演示将状态/上下文直接映射到动作/轨迹。 在生成控制输入或轨迹后，损失函数 L 将根据问题公式进行设计，并以监督学习的方式进行优化。 最先进的行为克隆使用负对数似然损失来更新策略，例如：

$$\arg\min_{\pi} \mathcal{L}(\pi) = -\frac{1}{N}\sum_{k=1}^{N} \log \pi(a_K \mid s_k) \tag{1}$$

Algorithm 1 Basic behavioural cloning method

1: Collect expert demonstration into dataset $\mathcal{D}$;
2: Select policy representation π_θ and loss function $\mathcal{L}$;
3: Use $\mathcal{D}$ to optimize the loss function $\mathcal{L}$ based on policy representation π_θ;
4: **return** optimized policy representation π_θ;

算法1展示了BC算法的基本框架，因为BC算法与MDP的联系较少，所以BC算法的效率很高，但是**缺点在于数据集D的好坏会影响BC算法的性能，尤其是当Agent转移到数据集中未出现的状态时，往往只能随机选择动作。**损失函数可以在不同的问题上设计成不同的形式，例如可以使用KL散度来作为损失函数来最小化专家分布q_π和策略分布：

$$\pi^* = \arg\min_{\pi} D_{KL}(q(\pi_E \| q(\pi))) \tag{2}$$

BC 可以细分为无模型 BC 和基于模型的 BC 方法。 主要区别在于该方法是否学习了一个前向模型来估计系统动力学。 由于无模型 BC 方法不考虑上下文，因此无模型 BC 方法在可以使用精确控制器且专家可以控制和修改机器人关节的行业应用中表现良好。 然而，无模型 BC 方法通常难以预测未来状态，并且无法保证在没有精确控制器的环境下输出的可行性。 在这种“不完美”的环境下，智能体对系统动力学的信息有限，并且通常会由于“compounding error”而陷入看不见的场景。 虽然基于模型的 BC 方法利用环境信息并迭代学习动态以产生可行的输出，但代价是基于模型的 BC 方法通常具有更大的时间复杂度，因为过程中设计迭代学习。

一个经典BC算法就是Dagger，Dagger通过策略π^*(某个固定策略)与新策略$\hat{\pi}_i$的概率混合来构建下一个策略$\hat{\pi}_{i+1} = \beta_i \pi^* + (1-\beta_i)\hat{\pi}_i$，它先通过当前策略来采样一批数据(状态动作对)，由于这些轨迹中会出现之前数据集中未出现的状态，因此Agent表现会很差，**为了修复未出现的状态，Agent把新产生的数据交给专家标注，并将标注后的数据加入数据集后再次训练新策略。**通过这样的方式，数据集覆盖的状态空间不断迭代增加，从而缓解compounding error，解决了传统BC算法在未见过的状态上表现不佳的情况。**然而Dagger涉及与专家的频繁交互，带来了大量的成本代价**，因此应用场景比较受限；Dagger的另一个问题是忽略了动作的成本，它是在动作成本相同的视频游戏上进行评估的，然而动作成本在许多现实问题上都是不同的。

3.2 逆强化学习

与BC不同，IRL迭代地从专家演示中恢复和评估奖励函数，而不是建立从状态到动作的映射。选择BC或IRL取决于问题设置。IRL试图从专家演示中学习“意图”，即reward function。BC更适合专家演示十分精确和丰富的任务，往往简单而高效；而IRL相反，它更适合从不完美的专家演示中推断“意图”。算法3展示了经典IRL算法：

Algorithm 3 Classic feature matching IRL method

Require: The set of demonstrated trajectories $\mathcal{D}$;

1: Initialize reward function parameter ω and policy parameter θ;
2: **repeat**
3: Evaluate current policy π_θ state-action visitation frequency u;
4: Evaluate loss function $\mathcal{L}$ w.r.t. u and the dataset $\mathcal{D}$ distribution;
5: Update the reward function parameter ω based on the loss function;
6: Update the policy parameter θ in the inner loop RL method using the updated reward parameter ω;
7: **until**
8: **return** optimized policy representation π_θ;

IRL的基本流程为：

1. 初始化由参数w控制的奖励函数$R_w(\tau)$和策略θ。
2. 根据当前策略π_θ来评估(状态,动作)对的访问频率u。
3. 选定关于u和数据集$\mathcal{D}$的Loss funciton.
4. 根据Loss function来更新奖励函数参数w。
5. 使用RL算法根据奖励函数R_w来更新π_θ
6. 不断重复2-5的流程。

为了使得状态能被直接用于计算reward，在IRL中通常使用ϕ来将状态映射到由[0,1]组成的特征向量上，R_w的具体表示如下：

$$\begin{aligned} &\phi(s): s \rightarrow [0,1]^k \\ &\phi(\tau) = E[\sum_t \gamma^t \phi(s_t)] \\ &R_w(\tau) = w^T * \phi(\tau) \end{aligned} \tag{3}$$

一个经典的IRL算法是最大熵算法，其基于最大熵来更新R_w：

$$w^* = \arg\max_w \sum_{\tau \in D} p(\tau|w) \tag{4}$$

其中$p(\tau|w)$的概率可以通过一个优化问题来求解，具体来说：该算法认为专家演示中的特征访问频率与恢复得到的奖励函数中特征访问频率的期望相同，即:

$$E_{\pi_\theta}[\phi(\tau)] = E_{\pi_E}[\phi(\tau)] \tag{5}$$

因此我们可以得到一个优化问题：

$$\begin{aligned} &max \; -p\log p \\ s.t. &\sum_{\tau \in D} P_{\pi_\theta}(\tau)\phi(\tau) = E_{\pi_E}[\phi(\tau)] \\ &\sum_{\tau \in D} P(\tau) = 1 \end{aligned} \tag{6}$$

通过拉格朗日法求解可得：

$$p(\tau) = \frac{1}{Z}\exp\left(R_w\right) \tag{7}$$

将式(7)带入式(4)可得最大熵逆强化学习的完整优化目标。

最大熵IRL也可以从另一个角度来理解：

$$\underset{c\in\mathcal{C}}{\text{maximize}}\left(\min_{\pi\in\Pi}-H(\pi)+\mathbb{E}_{\pi}[c(s,a)]\right)-\mathbb{E}_{\pi_E}[c(s,a)] \tag{8}$$

其中$H(\pi)\triangleq\mathbb{E}_{\pi}[-\log\pi(a\mid s)]$，$E_{\pi_E}$通过专家样本来估计。最大熵IRL通过寻找cost function $c\in C$给专家策略更低的coust，给其他策略更高的cost。专家策略可以通过确定的RL算法获得：

$$RL(c)=\arg\min_{\pi\in\prod}-H(\pi)+E_{\pi}[c(s,a)] \tag{9}$$

即寻找一个cost funciton给最高熵策略，并最小化期望累计成损失。

IRL的两个问题：

1. 框架内置的强化学习算法来更新策略参数θ，这在高维问题上是耗时且低效的。
2. 学习得到的reward funciton可能不唯一，许多不同的reward funciton可能会导致相同的动作，因此这要求IRL方法具有更强的表达能力和更高效的框架。

3.3生成对抗模仿学习(GAIL)

GAIL综合了GAN的对抗结构和IRL的思路，由生成式模型G和判别式模型D组成，其中G生成数据分布ρ_{π}并将其与样本数据分布ρ_{π_E}混合来迷惑D，如果D无法从中甄别出ρ_{π}，则说明G生成的轨迹与样本中的轨迹十分接近。算法4展示了GAIL的抽象流程。

Algorithm 4 GAIL [25]

Require: Expert trajectories $\tau_E\sim\pi_E$, initial policy and discriminator parameter θ_0, ω_0
  for $i=0,1,2,\ldots$ **do**
    Sample trajectories $\tau_i\sim\pi_{\theta_i}$.
    Update the discriminator parameters ω_i to ω_{i+1}.
    Update the policy parameter θ_i to θ_{i+1}.
  end for

形式化的来说GAIL的优化目标为：

$$\min_{\pi}\max_{D\in(0,1)^{\mathcal{S}\times\mathcal{A}}}\hat{E}_{\tau_i}[\log(D_w(s,a))]+\hat{E}_{\tau_E}[\log(1-D_w(s,a))] \tag{10}$$

其中D_w为判别式对数据的评分：如果D_w越接近0，那么判别式D认为该数据越可能是专家产生；若越接近于1，则D认为越可能是生成式G生成。**因此根据优化目标(10)简单来说是：而D希望尽可能的提高分类的准确性，即希望(10)左项中对G生成数据的判别概率越高，希望(10)右项对专家生成的数据判别概率越低；G希望产生更多的D无法正确识别的轨迹来最小化D的分类正确率；**

表4展示了GAIL近年来的改进：

Table 4. Different Kinds of Derivative on GAIL

GAILs	Methods
Make further improvement	MGAIL[7], InfoGAIL[35]
Apply to other research question	MAGAIL[63], GAIfO[70]
Other generative model	Diverse GAIL[76], GIRL[77]

3.4 从观测中模仿(IFO)

之前介绍的IL算法的数据输入形式为(状态，动作)对序列，而将诸如图像、视频帧等原始观测形式整理为(s,a)对序列往往是耗时耗力的，而且可能会遗漏信息。IFO算法试图直接通过视频来得到rewrad function，在IFO算法中，它尝试去给不同的上下文构建联系从而使用VAE方法对其编码。

目标模型包含4个组件：

1. $Enc_1(o_t^i)$从原始观测中提取特征向量z_1。
2. $Enc_2(o_0^j)$从目标观测中提取特征向量z_2。
3. 一个转换器z_3。
4. 一个目标上下文编码器$Dec(z_3)$。

该模型将两组观测值源观测值：$D_i = [o_t^i]_{t=0}^T$和目标观测值：$D_j = [o_t^j]_{t=0}^T$作为输入，然后在假设源观察和目标观察是时间对齐的情况下，使用这两个集合来预测目标上下文中的未来观察。转换器z_3 将$Enc_1(o_t^i)$产生的 z_1 特征转换成$Enc_2(o_0^j)$产生的 z_2 的上下文，即: $z_3 = T(z_1, z_2)$，然后把z_3通过$Dec(z_3)$解码到目标观测$\hat{o}_t^j$中。

该模型使用监督学习，损失函数为$L_{trans} = \|(\hat{o}_t^j) - o_t^j\|_2^2$。为了提高性能，所提出模型的最终目标与 VAE 重建的损失和时间对齐的损失相结合，即：

$$L = \sum_{(i,j)} (L_{trans} + \lambda_1 L_{rec} + \lambda_2 L_{align}) \tag{11}$$

其中λ_1和λ_2为预先确定好的超参，最终输出的reward funciton由两部分组成：一是平方欧几里得距离上的偏差惩罚，它衡量目标观测的编码特征和被转换到目标上下文的专家观测特征的距离，例如：

$$\hat{R}_{feat}\left(0_t^l\right) = -\left\| Enc_1\left(o_t^l\right) - \frac{1}{n}\sum_{t=0}^{T} T\left(o_t^i, o_0^l\right) \right\|_2^2 \tag{12}$$

二是确保当前观测与转换观测保持相似：

$$\hat{R}_{img}\left(0_t^l\right) = -\left\| o_t^l - \frac{1}{n}\sum_{t=0}^{T} M\left(o_t^i, o_0^l\right) \right\|_2^2 \tag{13}$$

其中 M 是完整的观测转移模型。提出的奖励函数可以应用于任何强化学习算法，例如TRPO。

4 挑战与机遇

- **Diverse behavior learning:**

  当前的 IL 方法通常使用特定于任务的训练数据集**来学习重现单个行为**。虽然有研究通过结合对抗性结构和变分自编码器提出了多样化的模仿学习，但这仍然是一个困难的挑战。可以采用其他方法来优化 IL，例如迁移学习可以帮助Agent从类似的任务中学习，从而使训练过程更加高效。

- **Sub-optimal demonstration for training:**

  当前的 IL 方法通常需要一组高质量的演示来进行训练。 但是，高质量演示的数量可能有限且获得成本高昂。 **现有的研究已经表明使用次优演示进行训练的可能性，但依然有待开发。**

- **Imitation not just from observation:**

  当前的 IfO 方法通常使用原始视频和观察偏差来恢复奖励函数。 **但观测不仅仅是视频，比如也有可能是语音输入。** 这在IFO中可能是一个待开发的方向。

- **Better representation:**

  良好的策略表示可以有利于训练过程，以实现数据效率和计算效率。 寻找更好的政策表示仍然是 IL 的一个活跃研究课题。 除了策略表示之外，如何表示演示是 IL 中的另一个问题。 示范的表现需要更有效率和表现力。目前的演示都局限于(s,a)对或者视频帧输入，如何压缩数据的表示或做类似特征工程的研究是一大方向。

- **Find globally optimal solution：**

  大多数研究是基于演示寻找局部最优解决方案，这可能为代理性能设定上限。 未来的方向可能是为特定任务找到全局最优解，这需要代理理解行为的意图，而不是简单的模范克隆。 目前有研究成功超越了演示器的性能，但找到全局最优仍然需要努力。

5 总结

模仿学习在广泛的问题中都取得了出色的表现，从解决困难的探索 Atari 游戏到通过机械臂在避开障碍物的同时实现对象操纵。 不同种类的模仿学习方法对这一重大发展做出了贡献，例如 BC 方法更直观地复制行为，而环境参数可以很容易地获得； 当问题更多地关注环境动态而不关心训练时间时，IRL 方法可以实现数据效率和未来行为预测； 对抗性结构化 IL 方法在训练过程中消除了专家交互，并提供了足够的处理高维问题的能力。 在 IL 方法不断发展和发展的同时，IL 也在寻求在设置上的突破，例如 IfO 方法通过取代需要来简化输入 当输入演示是原始视频时的动作标签。 尽管 IL 取得了成功，但挑战和机遇依然存在，例如多样化的行为学习、利用次优演示和语音指导、更好的表示以及最终找到全局最优解。 未来的工作有望解开 IL 及其实际应用。