

1 Policy

DexGraspNet: 合成数据 (Synthetic Data) + 深度学习

- 1. 场景理解: 预测每个点 **抓取可能性 (Graspness)**, 是否是 **物体 (Objectness)**
- 2. 局部特征: 不用全局特征 (关联性弱、泛化性差), 选择 Graspness 高的地方附近的点云, 提取局部特征 (**几何信息**)
- 3. 条件抓取生成模块: 条件生成处理 (T, R) 多峰分布, 然后采样后直接预测手指形态 θ

仅处理包覆式抓取 (Power Grasp), 没处理指尖抓取 (Precision Grasp); 主要使用力封闭抓取; 折射 (Refraction) / 镜面反射 (Specular Reflection), 导致点云质量差。

ASGrasp: 深度修复, 合成数据 + 监督学习。域随机化、多模态立体视觉、立体匹配 (Stereo Matching)。

可供性 (Affordance): 指一个物体所能支持或提供的交互方式或操作可能性, 哪个区域、何种方式进行交互。

Where2Act: 大量随机尝试 + 标注。学习从视觉输入预测交互点 a_p 、交互方向 $R_{z|p}$ 和成功置信度 $s_{R|p}$ 。 **VAT-Mart**: 预测一整条操作轨迹。

利用视觉输入进行预测:

- **物体位姿 (Object Pose)**: 需要模型、抓取标注。
- **抓取位姿 (Grasp Pose)**: 直接预测抓取点和姿态, 无模型或预定义抓取。
- **可供性 (Affordance)**

运动规划 (+ 启发式规则, 如预抓取 Pre-grasp, 到附近安全位置再闭合, 避免碰撞)

- 1. **操作复杂度有限**: 难以处理复杂任务, 受启发式规则设计限制。
- 2. **开环执行 (Open-loop)**: 规划一次, 执行到底, 闭眼做事。高频重规划可近似闭环。

策略学习: 学习 $\pi(a_t|s_t)$ 或 $\pi(a_t|o_t)$, 实现 **闭环控制**。

BC: 将 $D = \{(s_i, a_i^*)\}$ 视为监督学习任务, 学习 $\pi_\theta(s) \approx a^*$ 。

分布偏移: 策略 π_θ 的错误会累积, 导致访问训练数据中未见过的状态 ($p_\pi(s)$ 与 $p_{\text{data}}(s)$ 不匹配), 策略失效。

- 1. 改变 $p_{\text{data}}(o_t)$: **Dataset Aggregation (Dagger)**

训练 $\pi_i \Rightarrow$ 用 π_i **执行 (Rollout)** 收集新状态 \Rightarrow 查询专家在此状态下的 $a^* \Rightarrow D \leftarrow D \cup \{(s, a^*)\} \Rightarrow$ 重新训练 π_{i+1} 。但是出错才标注, 也会影响准确性。

- 2. 改变 $p_\pi(o_t)$ (**更好拟合**): 从 (传统算法) 最优解中获取; 从教师策略中学习 (有 **特权信息 Privileged Knowledge**)

遥操作数据 (Teleoperation): 贵, 也存在泛化问题。

非马尔可夫性: 引入历史信息, 但可能过拟合, 因果混淆 (Causal Confusion)。

目标条件化 (Goal-conditioned): $\pi(a|s, g)$, 共享数据和知识。但 g 也有分布偏移问题。

IL 局限性: 依赖专家数据、无法超越专家、不适用于需要精确反馈的高度动态 / 不稳定任务。

离线学习 (Offline Learning): 指学习过程无法干预数据的产生过程。我们只能使用一个预先收集好的、固定的数据集进行学习。模仿学习中的 BC 就是典型的离线学习。

在线学习 (Online Learning): 指智能体在学习过程中可以主动与环境交互, 实时产生新的数据, 并利用这些新数据更新自己的策略。强化学习通常可以在线进行。

策略梯度定理: 采样 N 条轨迹 $\tau^{(i)} \sim p_\theta(\tau)$, 然后计算:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \nabla_\theta \log p_\theta(\tau^{(i)}) R(\tau^{(i)})$$

请注意, 这个梯度表达式中并没有出现奖励函数 $R(\tau)$ 关于 θ 的梯度 $\nabla_\theta R(\tau)$ 。 $\nabla_\theta \log p_\theta(\tau) = \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t)$ (马尔科夫性拆开简化)

环境模型: 包括状态转移概率 $p(s_{t+1}|s_t, a_t)$ 和奖励函数 $r(s_t, a_t)$, 真实世界一般都拿不到。

- **Model-Free**: 不需要知道环境的模型
- **Model-Based**: 利用神经网络学习环境的模型

基础策略梯度算法 (REINFORCE): 按照整条轨迹的总回报 $R(\tau^{(i)})$ 加权, On-Policy。

初始化 θ , 然后循环

- 1. 用 π_θ 采样 N 条轨迹 $\{\tau^{(i)}\}$, 计算 $R(\tau^{(i)})$
- 2. $\hat{g} = \frac{1}{N} \sum_{i=1}^N \left[\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t^{(i)}|s_t^{(i)}) \right) R(\tau^{(i)}) \right]$
- 3. 更新 $\theta \leftarrow \theta + \alpha \hat{g}$

On-Policy: 数据来自当前策略。通常效果好, **样本效率低**, 每次都得重新采样。

Off-Policy: 数据可来自不同策略。**样本效率高**, 可能不稳定。