paper

Created	@August 16, 2025 11:43 AM
Class	paper

▼ Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning

Def of Safe Learning

- 不仅仅关注收敛后的性能，而是保证学习的每一步都安全

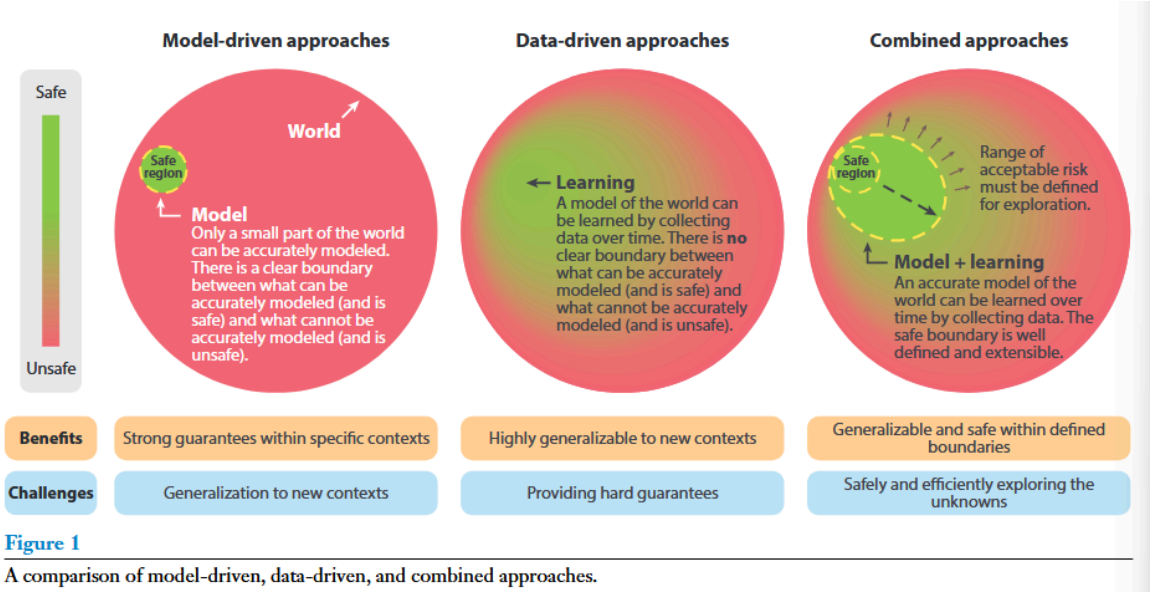


Figure 1

A comparison of model-driven, data-driven, and combined approaches.

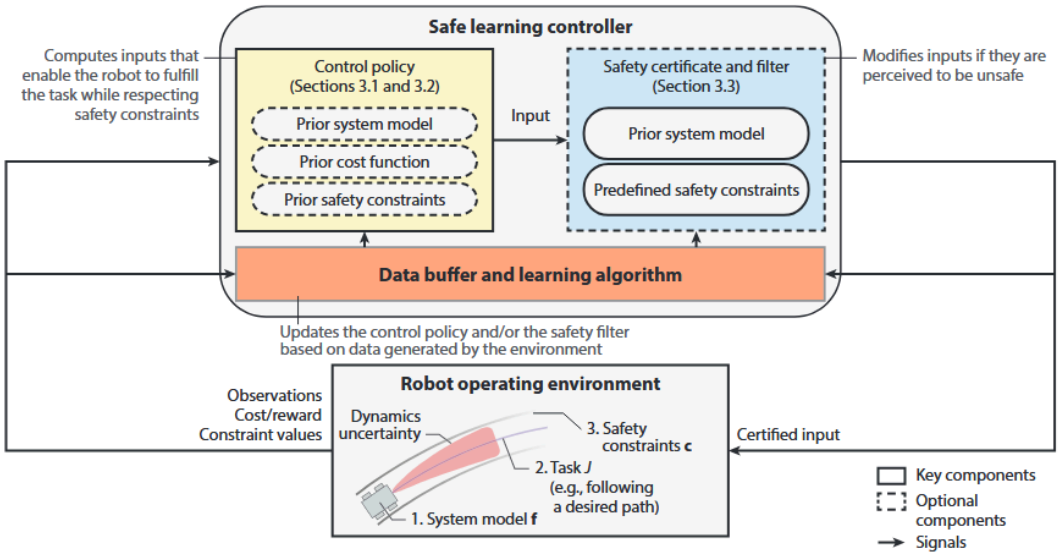


Figure 2

Block diagram representing the safe learning control approaches reviewed in this article. The three main components of the safe learning control problem are the cost function J, the system model $\mathbf{f}$, and the constraints $\mathbf{c}$, all of which may be initially unknown. Data are used to update the control policy (see Sections 3.1 and 3.2) or the safety filter (see Section 3.3).

Safety level

- 3: constraint satisfaction guaranteed
- 2: constraint satisfaction with high probability
- 1: constraint satisfaction encouraged

Control theory perspective

- 模型预测控制（MPC）
- 控制屏障函数（CBFs）
- robust MPC
- constrain exploration

RL Perspective

- constrained RL
- reward shaping
- risk sensitive RL
- safe exploration

Combination

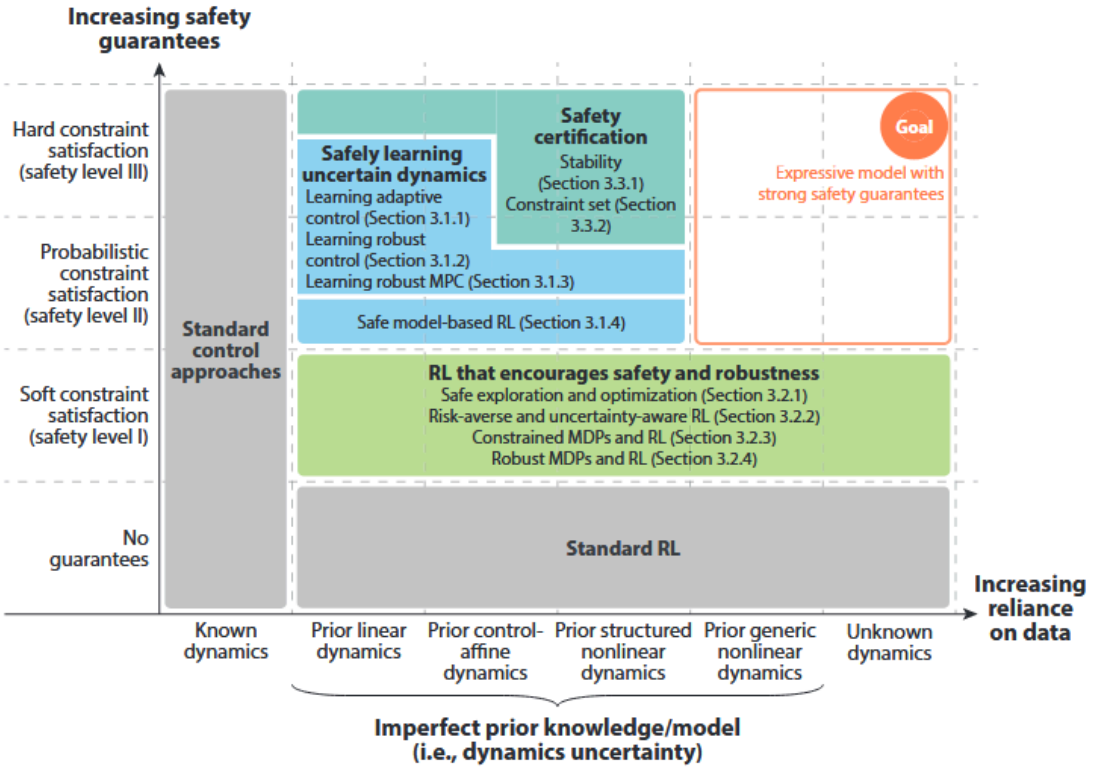


Figure 4

Summary of the safe learning control approaches reviewed in Section 3. Abbreviations: MDP, Markov decision process; MPC, model predictive control; RL, reinforcement learning.

Approaches that address the problem of safe learning control at two stages:

1. online adaptation or learning
    a. online data are used to adjust paras of the controller, the robot dynamic model, cost function, constrain functions, ...
2. offline learning
    a. recorded collected data and use to update model

Safe learning control approaches

- learning uncertain dynamics to safe improve performance
    - integrating machine and adaptive control
    - learning-base robust control

    - reducing conservatism in robust model predictive control with learning and adaption
    - safe model-based reinforcement learning with a priori dynamics
- encouraging safety and robustness in RL
    - safe exploration and optimization
    - risk-averse reinforcement learning and uncertainty-aware reinforcement learning
    - constrained Markov decision processes and reinforcement learning
    - robust Markov decision processed and reinforcement learning
- certifying learning-based control under dynamics uncertainty
    - stability certification
    - constraint set certification

4.BENCHMARKS

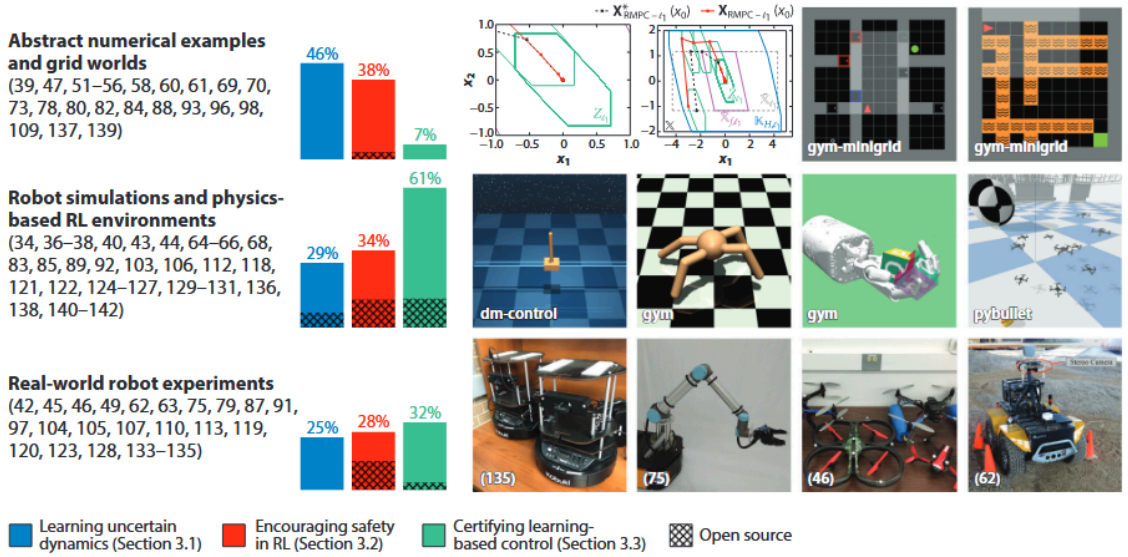


Figure 5

Summary of the environments used for evaluation. With increasing complexity, they can be classified as abstract numerical examples and grid worlds, robot simulations and physics-based RL environments, and real-world robot experiments. The histograms show the prevalence of each category in Sections 3.1–3.3, as well as the fraction whose code is open source. Abbreviation: RL, reinforcement learning.

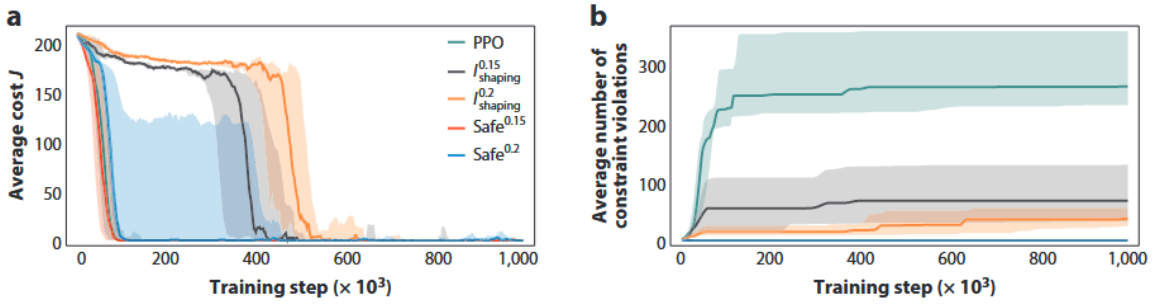


Figure 7

Total (*a*) cost and (*b*) constraint violations during learning for PPO, PPO with cost shaping (for two parameterizations), and PPO with safe exploration (for two slack variable values). Plotted are medians with upper and lower quantiles over 10 seeds. Abbreviation: PPO, proximal policy optimization.

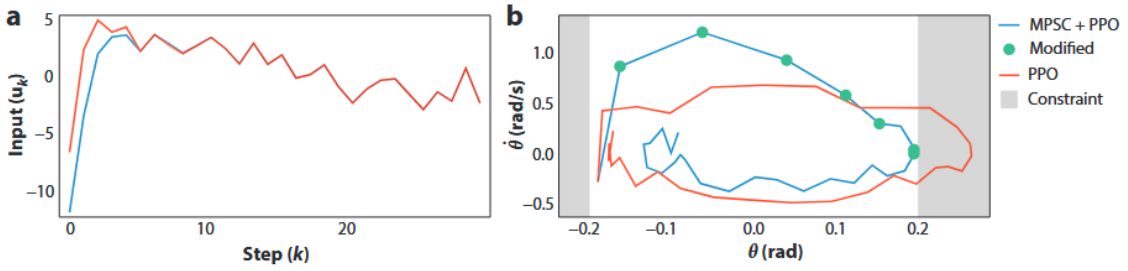


Figure 8

(*a*) Plot of uncertified PPO input (*red*) against certified MPSC + PPO input (*blue*). (*b*) Cart–pole state diagram (θ and $\dot{\theta}$) comparing the MPSC + PPO certified trajectory (*blue*) and the uncertified PPO trajectory (*red*). Green dots show when the MPSC modified the learning controller's input; the MPSC is most active when the system is about to leave the constraint boundary (*gray*) or the set of states from which the MPSC can correct the system. Abbreviations: MPSC, model predictive safety certification; PPO, proximal policy optimization.

1. 安全层 / Safety Filter（动作修正）

- 机器人RL策略输出一个动作 *uRL* 。
- 在执行前，**安全层会检查是否满足约束**（比如关节力矩不超限、姿态不失稳）。
- 如果不安全，就替换为最接近的安全动作 *usafe* 。
- 实现方式：
  - **CBF（控制屏障函数）**：实时解优化问题，强制状态保持在安全集内。
  - **QP（Quadratic Programming）安全投影**：解 min ||*u* – *uRL*||2 约束在安全范围。

2. 安全探索 / Safe Exploration

- RL 训练时不是盲目尝试，而是**限制动作空间**：

- 初期在一个已知的**安全集**里探索（比如慢走、低速转向）。
- 随着数据积累，逐渐扩大安全集。

- 常见方法：
  - **基于高斯过程（GP）的SafeOpt**：只在模型置信区间内探索。
  - **Lyapunov-based RL**：策略更新时强制保证Lyapunov稳定条件。

---

3. 结合控制理论的 RL 更新

- 用 **MPC + RL** 或 **RL + 控制律**：
  - RL 学习高层策略（如步态参数），底层由MPC或PD控制保证稳定。
  - RL 更新基于真机数据，但安全性由底层控制保证。
- 或者在 RL 更新时引入 **约束 MDP（CMDP）**：
  - 在优化时同时考虑奖励和安全约束。
  - 如果新策略可能违反约束 → 不更新或退回上个安全策略。

实时数据更新流程（实践）

在机器人真机上常见的安全 RL loop 大概长这样：

1. **传感器采集数据**
2. **策略产生动作**。
3. **安全层检查/修正动作**（CBF/QP/MPC）
4. **执行动作并收集数据**（state, reward, constraint violation）
5. **后台异步更新RL参数**
6. **安全验证**：新参数只在验证通过后才上线执行

可能的实现方法：

1. PPO/SAC + **动作安全层**（QP投影/CBF）
2. RL只学步态参数，高频稳定性由MPC或LQR控制保证.

▼ SERL

Components

RLPD

- **高样本效率**：RLPD是一种离策略（off-policy）的Actor-Critic算法，基于Soft Actor-Critic (SAC)改进而来 。它支持高“更新-数据比率”（UTD），即每与环境交互一次，就在内部进行多次梯度更新，从而充分利用来之不易的真实世界数据 。
- **融合先验数据**：该算法能够轻松地将人类演示等先验数据整合进训练过程中 。具体做法是在每次训练时，从先验数据（如演示）和在线采集的数据中各采样一半，组成一个训练批次 。这使得策略可以从一个较高的起点开始学习，极大加速了收敛。

奖励函数设计

- **二元分类器奖励**：用户只需收集少量任务成功和失败状态的图像，训练一个二元分类器 。在RL训练中，该分类器会判断当前状态是否成功，并输出一个概率，奖励函数则被设为该成功概率的对数（$r(s) = \log p(e|s)$），从而引导策略学习 。
- **对抗性训练（VICE）**：为了防止策略找到“欺骗”分类器的捷径（即看起来成功但实际未成功的状态），SERL支持使用VICE方法 。该方法会将策略访问过的所有状态都作为负样本来更新分类器，形成一种策略（生成器）和奖励分类器（判别器）之间的对抗性训练

免重置的持续训练框架

系统同时训练两个独立的RL智能体：

- **Forward Policy**：学习完成指定任务（例如，将物体从A点移动到B点） 。
- **Backward Policy**：学习将任务“撤销”，即把环境恢复到初始状态（例如，将物体从B点移回A点） 。

通过交替执行这两个策略，机器人可以在无人干预的情况下持续进行“尝试-重置”的循环，实现7x24小时的自主训练

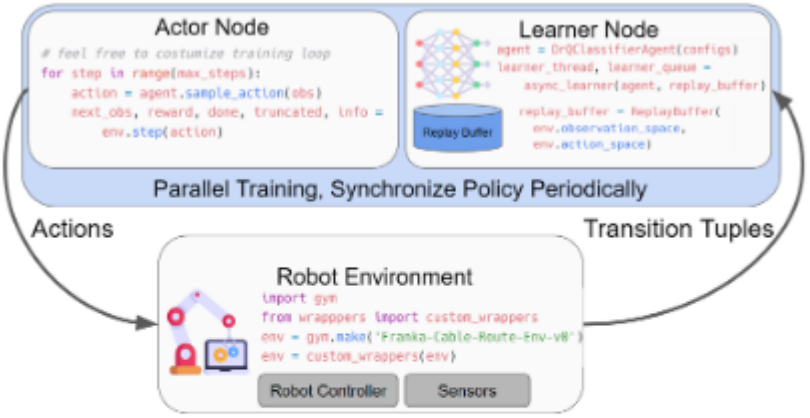


Figure 2: Software architecture and real-world robot training example code. SERL runs three parallel processes, consisting of the actor, which chooses actions, and the learner node, which actually runs the training code, and the robot environment, which executes the actions from the actor and contributes data back to the learner.

Actor-learner: Actor（执行器）节点负责与机器人环境交互并生成动作，而Learner（学习器）节点负责在后台进行策略网络的更新计算 。

这种分离设计确保了机器人的控制频率可以稳定在固定值，这对于需要快速响应的动态任务至关重要 。同时，它也减少了因计算等待造成的总训练时间。

For contact-rich tasks

对于PCB插入这类需要与环境进行精密接触的任务，SERL提出并集成了一种特别有效的**阻抗控制器设计**。

- **分层控制**：RL策略以较低频率（如10Hz）输出目标位姿，而底层的实时控制器（文中是PID）则以高频（如1kHz）跟踪该目标 。
- **实时层误差限制**：传统方法可能会因RL策略输出的目标离当前位置太远而产生巨大的接触力，导致硬件损坏 。SERL的关键创新在于，它**不在低频的RL策略层限制动作大小，而是在高频的实时控制器层限制目标位姿与当前位姿的误差** e 。

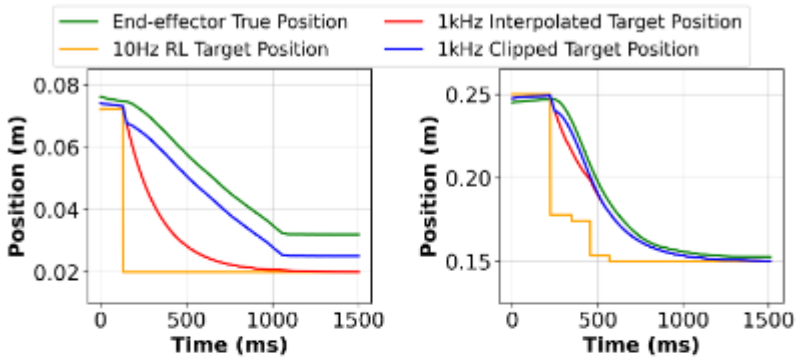


Figure 3: Visualization of controller logs from the robot when commanded with different movements, for the z-axis of the end-effector. The orange line is the commanded target (the output of RL), red is the smoothed target sent to the real-time controller, blue is the clipped target, and green is the robot position after executing this controller. **Left:** The robot end-effector was commanded to move into contact with a hard surface and continue the movement despite the contact. The reference limiting mechanism clipped the target to avoid a hard collision. **Right:** The command is a fast free-space movement, which our reference limiting mechanism does *not* block, allowing fast motion to the target.

如图，因为高频控制器+误差严格限制的关系，可以做到柔顺并不牺牲整体运动效率

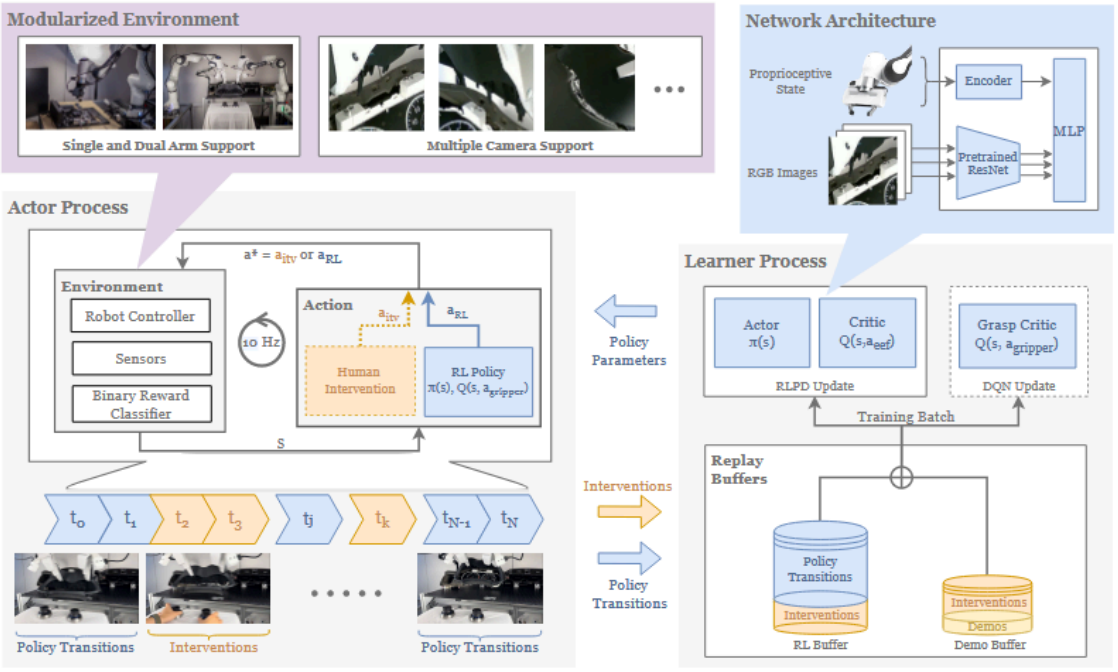
相对坐标系

相对观测：机器人的本体感知信息（如末端执行器位姿）是相对于**本回合初始时刻**的末端位姿来表示的，而不是相对于固定的世界坐标系 。

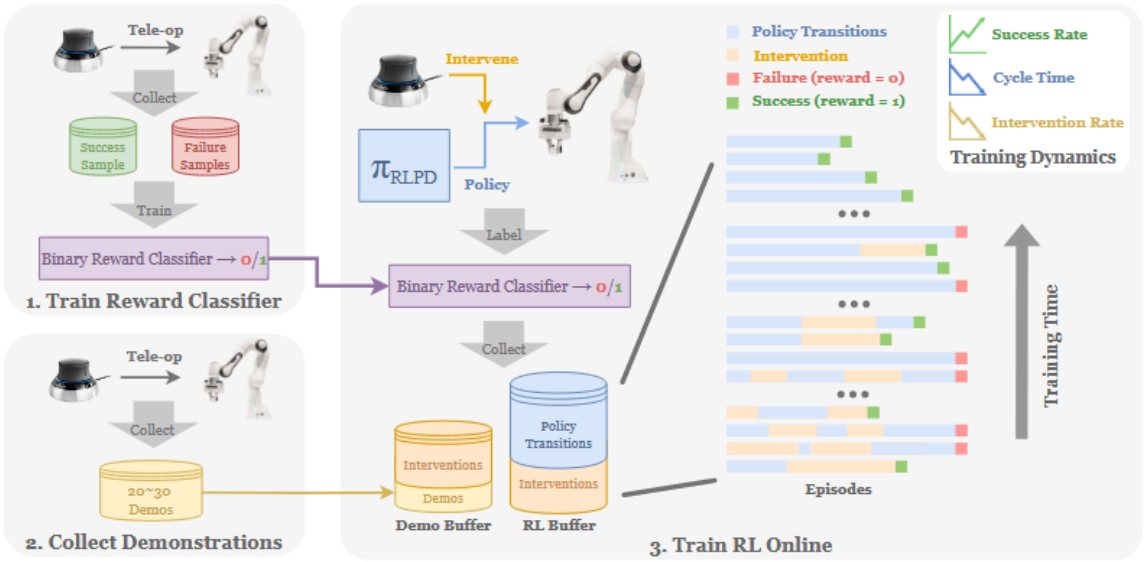
1. **相对动作**：策略输出的动作（如6D速度）是相对于**当前时刻**的末端执行器坐标系 。
2. **效果**：这种“以自我为中心”的表示方法，相当于在训练中模拟了目标的移动，使得策略学会的是一种相对运动关系，从而在测试时即使目标位置被扰动，策略也能成功完成任务 。

▼ Precise and Dexterous Robotic Manipulation via Human-in-the-Loop Reinforcement Learning

Overview of HIL-SERL



Process for training HIL-SERL



分布式架构，异步通信：

- Actor Process
  - 直接与机器人、环境交互的部分，加载策略并执行， 同时回传采集到的数据到Replay Buffer. 此过程可由人类操作手通过SpaceMouse等设备随时介入。
- Learner Process

    - 从经验回放池中均匀采样数据，使用RLPD算法更新策略网络和价值网络的参数
- Replay Buffer
    - Demo Buffer
        - 存放20-30条人类专家演示数据，以及干扰数据
    - RL Buffer
        - 存放机器人自主执行策略时产生的数据
    - learner将在两个池中等量采样数据进行训练

Training Process

- **First**, we select cameras that are most suitable for the task. For all cameras, we perform image cropping to focus on the area of interest and resize the images to 128×128 for the neural network to process.
- **Next**, we collect data to train the reward classifier. Additionally, we may collect extra data to address any false negative and false positive issues with the reward classifier. The trained reward classifier generally achieves an accuracy of greater than 95% in the evaluation data set. （训练奖励二元分类器）
- We **then** collect 20-30 trajectories of human demonstrations solving the tasks and use them to initialize the offline demo replay buffer. For each task, we either script a robot reset motion or let the human operator manually reset the task at the beginning of each trajectory, such as the USB pick-insertion task. （填充演示经验池）
- **Finally**, we start the policy training process. During this phase, human interventions may be provided to the policy if necessary, until the policy converges. It’s also important to note that we should avoid persistently providing long sparse interventions that lead to task successes. Such an intervention strategy will cause the overestimation of the value function, particularly in the early stages of the training process; which can result in unstable training dynamics. （在线人机强化学习）

Summary

设计理念：结合模仿学习和强化学习的优点

- 通过初始的人类演示和训练过程中的实时纠正，为机器人策略提供高质量的初始引导和关键时刻的指导，减少机器人进行无效探索所需的时间 。
- **强化学习的优化**：利用RL算法强大的优化能力，使策略不仅能学习人类的操作，还能通过自主试错和奖励驱动，探索出比人类演示更优、更快、更稳定的解决方案 。

真实物理世界运行使用技术

二元的稀疏奖励函数

- RL传统的reward设计方法对不同task的复用性差，而且设计时间经验成本高。为了解决这个问题，这篇文章首先通过遥操作收集少量成功和失败状态的图像数据（约5分钟），训练一个二元分类器 。在训练中，只有当任务完成时，分类器才会给予一个正奖励，否则奖励为零 。

预训练视觉主干网络

- 采用了在ImageNet上预训练好的ResNet-10模型作为视觉特征提取器 。这使得系统可以更快地学习到与任务相关的视觉特征，显著提升了样本效率和稳定性 。

Core: 人机循环的干预与学习机制

- 在训练过程中，人类操作员监控机器人的行为。当机器人陷入困境或执行错误动作时，操作员可以随时介入并提供纠正性操作 。这些干预数据会被同时存入演示池和RL池，用于后续的离策略学习 。随着策略的改进，人类干预的频率会逐渐降低，最终趋近于零 。

Ego-centric Formulation

- 从本质上讲，在每个训练事件的开始时，机器人的最终效果的姿势在工作区的预定区域内均匀地随机分配。

阻抗控制器

分离的抓取器控制

- 机械爪开合的任务，系统并没有将离散的抓取动作（开、合、保持）与连续的手臂运动统一建模 。而是为抓取动作单独训练了一个评判家网络（Critic Network），使用DQN算法来决定在当前状态下应该执行哪种抓取动作 。

分类

- 异步安全RL架构：安全探索+离线和在线结合 的方法
- 限制探索：人类在危险时刻接管
- 底层安全层：阻抗控制
- reward shaping via demonstrations: 解决reward engineering 难的问题
- Representation choices for safe generalization: 减少策略对绝对坐标的依赖
- Modular architectures for safety: 把末端爪子的抓取策略分开

机器狗相关

机器狗控制的本质

- 闭环控制，需要传感器实时反馈，调整动作平衡（Reactive Policy）

可能的方法

- 分层控制：高层（例如10Hz）的RL策略负责决策，输出目标；底层（例如1000Hz）的实时控制器（PID）负责执行和跟踪 ,并在底层限制误差 e
- 人类干预机制：当RL策略导致可能摔倒时，人类接管控制器relax（不知道机器狗任务应该怎样有效的接管）
- 免重置的策略：只让狗在地形中行走一小段路，就用回向策略倒退，重复学习（但是对于翻倒情况，这个可能不适用，需要人类干预翻倒自救）
- 相对观测：局部决策