# 深度强化学习的挑战

## 第七章

沈梓倩、赵玉俊、王骁越、项学文、韩成瑞

东南大学数学学院

2025 年 11 月 10 日

# 目录



## 汇报顺序

沈梓倩：1 引言；2 样本效率
赵玉俊：3 训练稳定性；4 灾难性遗忘
王骁越：5 探索问题；6 元学习与表示学习
项学文：7 多智能体强化学习；8 模拟到现实
韩成瑞：9 大规模强化学习；10 其他挑战；11 总结与展望

# 目录

# 深度强化学习面临的主要挑战

核心挑战：

1. 样本效率问题
2. 训练稳定性
3. 灾难性遗忘
4. 探索相关问题
5. 元学习与表示学习
6. 多智能体强化学习
7. 模拟到现实迁移
8. 大规模强化学习

**本章目标**

- 识别现有方法的缺陷
- 了解可能的解决方案
- 探索未来研究方向

# 目录

# 样本效率问题

## 定义

样本效率（Sample Efficiency）指算法从有限数据中学习有效策略的能力。样本高效算法能够用更少的环境交互次数达到相同或更好的性能。

问题的严重性：

- 人类学习者：Pong 游戏约 15 分钟（∼1000 帧）即可掌握
- DQN 算法：需要约 2 亿帧样本才能达到人类水平
- 差距：约 **20** 万倍的样本效率鸿沟

## 实际意义

- 现实世界数据采集成本高（时间、能源、硬件磨损）
- 安全关键应用中失败代价巨大（自动驾驶、医疗）
- 限制了强化学习在真实场景的部署

# 提高样本效率的方法（1/3）

**1.** 从专家示范中学习（**Imitation Learning**）

- 行为克隆：监督学习专家轨迹
- 逆强化学习（**IRL**）：推断专家奖励函数
- **GAIL**：生成对抗模仿学习
- 应用：AlphaGo 预训练、机器人操作

**2.** 基于模型的强化学习（**Model-Based RL**）

核心思想：

- 学习环境动力学模型
- 利用模型规划和想象
- 减少真实环境交互

模型组成：

- 状态转移：$s_{t+1} = f(s_t, a_t)$
- 奖励函数：$r_t = r(s_t, a_t)$

# 提高样本效率的方法（2/3）

案例：**PILCO** 算法 [1]

- 使用高斯过程（**Gaussian Process**）建模动力学
- 通过矩匹配（**Moment Matching**）进行策略优化
- Cart-Double-Pendulum 任务：仅需 20-30 次试验
- 对比：无模型方法（如 DQN）需要数千至数万次试验

## 局限性与挑战

- 模型偏差（**Model Bias**）：学到的模型不准确可能误导策略
- 复合误差（**Compounding Error**）：多步预测误差累积
- 计算复杂度：精确建模高维系统困难
- 可扩展性：难以应用于复杂视觉输入任务

# 提高样本效率的方法（3/3）

**3.** 改进算法本身

| 问题 | 解决方案 |
| --- | --- |
| 策略梯度方差大 | Actor-Critic 方法 |
| 小规模 大规模 | DQN（深度神经网络） |
| Q 值过估计 | Double DQN |
| 探索不足 | Noisy DQN, SAC |
| 离散 连续 | DDPG |
| DDPG 不稳定 | TD3（孪生延迟 DDPG） |
| 策略安全更新 | TRPO（信赖域方法） |
| TRPO 计算慢 | PPO（一阶近似） |
| 二阶优化加速 | ACKTR |

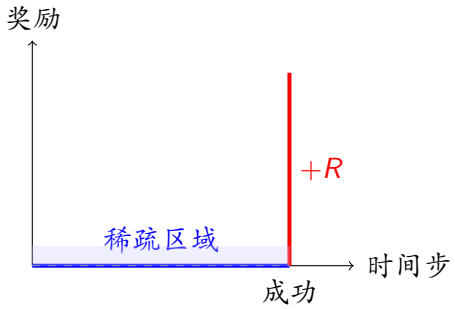
# 稀疏奖励问题

问题描述：

- 环境仅在任务完成时给予奖励
- 中间过程奖励值为 0 或常数
- 缺乏有效的学习信号引导探索
- 信用分配（Credit Assignment）困难

解决方案：

- 后见之明经验回放（**HER**）
- 分层强化学习（**HRL**）
- 内在动机（**Intrinsic Motivation**）
- 好奇心驱动（**Curiosity-Driven**）
- 课程学习（**Curriculum Learning**）




**典型场景**

机器人抓取、迷宫导航、游戏关卡通关

# 目录

# 学习稳定性问题

## 不稳定的表现

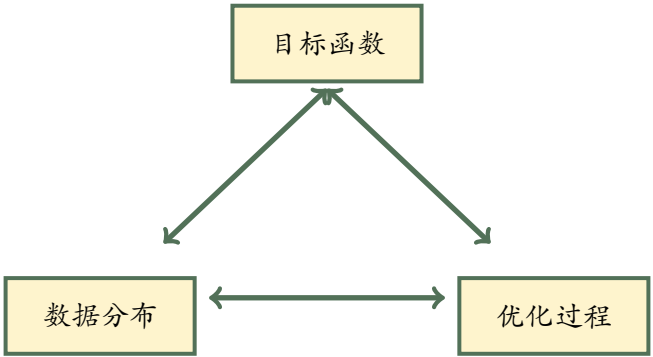
- 时间维度：学习曲线剧烈震荡、非单调性、突然崩溃
- 横向比较：不同随机种子结果差异巨大

造成不稳定的根本原因：

- 1. 非平稳性：策略改变导致数据分布持续变化
- 2. 违反 **i.i.d.** 假设：序列数据高度相关
- 3. 自举偏差：用估计值更新估计值
- 4. 高方差梯度：策略梯度估计器噪声大
- 5. 脆弱性：对超参数、初始化极度敏感

## 核心挑战

目标函数、数据分布、优化过程三者相互耦合、动态变化



# VIME 实验中的不稳定性

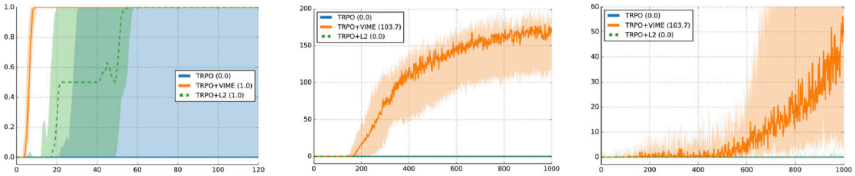



图 1: VIME 实验学习曲线（Houthooft et al., 2016）

- **MountainCar**: TRPO 曲线覆盖整个奖励范围 [0,1]
- **HalfCheetah**: TRPO+VIME 同样不稳定
- TRPO 已是相对稳定算法（二阶优化 + 信赖域）

# 影响稳定性的因素 [2]

**Henderson** 等人的研究结论：

- 1. 网络结构：对 TRPO 和 DDPG 结果有显著影响
- 2. 激活函数：ReLU 或 Leaky ReLU 表现最好
- 3. 奖励缩放：效果对不同环境不一致
- 4. 随机种子：5 个种子可能不足

**重要提示**

使用不同随机种子获得平均结果非常重要

# 提高稳定性的方法

| 方法 | 改进措施 |
|---|---|
| 价值函数拟合 | 降低 REINFORCE 算法的方差 |
| Actor-Critic | 结合 Q-Learning 和策略梯度 |
| DQN | 目标网络 + 延迟更新 + 经验回放池 |
| TD3 | 目标策略平滑正则化 + 双 Critic |
| TRPO | 二阶优化 + 信赖域限制 |

## 挑战仍存

不稳定性、随机性和对超参数的敏感性仍是强化学习社区的巨大挑战

# 目录

# 灾难性遗忘问题

## 定义

灾难性遗忘（Catastrophic Forgetting）[3]：神经网络在持续学习新任务时，对新数据的拟合导致已学知识的快速丢失，性能急剧下降。

在强化学习中的表现：

- 策略退化：学习新策略时忘记之前有效的行为模式
- 价值函数失真：对已访问状态的价值估计变得不准确
- 多任务干扰：多个任务之间的负迁移

为什么在 **RL** 中特别严重？

- 强化学习本质上是非平稳优化问题
- 策略改变 $\rightarrow$ 数据分布变化 $\rightarrow$ 价值函数更新 $\rightarrow$ 策略进一步改变
- 形成" 追逐移动目标" 的循环

# 解决灾难性遗忘的方法

**1.** 约束策略更新（**Constrained Policy Updates**）

- **TRPO**：KL 散度约束保证策略渐进改进
- **PPO**：裁剪目标函数限制更新幅度
- 数学形式：$\max_\theta \mathbb{E}[\cdots]$ s.t. $D_{KL}(\pi_{\theta_{old}} \| \pi_\theta) \leq \delta$

**2.** 经验回放机制（**Experience Replay**）

- 经验回放池：存储历史转移，打破时间相关性
- 优先经验回放（**PER**）：重点回放高 TD 误差样本
- 后见之明经验回放（**HER**）：稀疏奖励下的数据增强

**3.** 网络正则化技术

- 弹性权重巩固（**EWC**）：保护重要参数不被过度修改
- 渐进神经网络：为新任务添加新模块，冻结旧参数
- 知识蒸馏：用旧模型约束新模型的输出分布

# 目录

# 探索的挑战

探索困难的来源：

- 稀疏奖励：成功信号极少
- 欺骗性奖励：局部最优陷阱
- 高维动作空间：组合爆炸
- 长时间依赖：延迟奖励归因
- 安全约束：现实中探索风险

探索-利用困境（**Exploration-Exploitation Dilemma**）：

如何在利用已知最优策略和探索未知区域之间取得平衡？

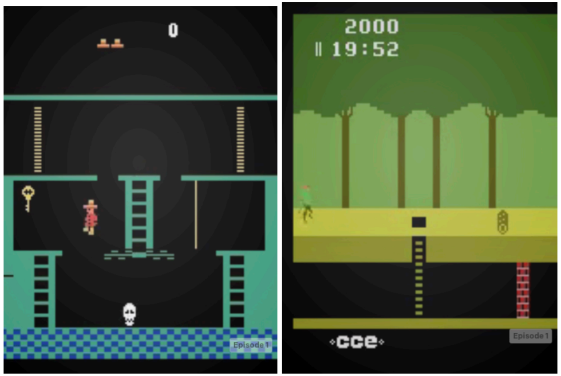



图 2: Montezuma's Revenge：探索极具挑战性的 Atari 游戏

# 大规模游戏的探索挑战

游戏复杂度对比 [4]

| | 雅达利游戏 | 围棋 | 《星际争霸》 |
|---|---|---|---|
| 信息类型 | 近完美 | 完美 | 不完美 |
| 动作空间 | 17 | 361 | $10^{26}$ |
| 每场游戏的活动次数 | 100/s | 100/s | 1000/s |
| 玩家数量 | 单个 | 两个 | 多个 |

图 3: 对比不同的游戏

**挑战**

巨大的动作空间（$10^{26}$）与长控制序列

# 探索问题的解决方案

**1.** 模仿学习（**Imitation Learning**）

- 从专家演示中学习，快速接近可行策略空间

**2.** 内在动机（**Intrinsic Motivation**）

- 好奇心驱动：奖励预测误差 $r_i = \|\hat{s}_{t+1} - s_{t+1}\|^2$
- 计数探索：访问次数倒数作为探索奖励
- 信息增益：最大化状态-动作信息量

**3.** 分层强化学习（**Hierarchical RL**）

- 将复杂任务分解为子目标层次
- **Options** 框架：学习可重用的技能
- **Feudal Networks**：管理者-工作者结构

# 新兴探索方法

**1. Go-Explore 算法**

- 第一阶段：在确定性环境中系统性探索，记录所有访问状态
- 第二阶段：返回有价值的状态继续探索（" 回到感兴趣状态"）
- 第三阶段：鲁棒化训练，对环境随机性建立鲁棒策略
- 在 Montezuma's Revenge 上取得超人类表现

**2. 基于种群的训练（Population-Based Training, PBT）** [4]

- DeepMind 用于 AlphaStar 训练
- 多样性维持：智能体集合形成联盟（League）
- 自我对弈：不同策略之间相互竞争
- 在线进化：动态调整超参数和网络结构
- 充分探索策略空间，避免陷入局部最优

# 现实世界探索的安全性

## 问题示例：自动驾驶

- 需要从失败情况学习
- 实际车辆无法采集车祸样本
- 不能使用随机动作探索（可能导致灾难）

解决方案：模拟到现实转移（**Sim-to-Real Transfer**）

- 1. 先在模拟中进行训练
- 2. 再将策略转移到现实中
- 3. 适用于：机器人操作、机器人手术等

# 目录

# 元学习与表示学习

## 核心问题

如何让智能体利用多任务学习经验，实现快速适应新任务的能力？

元学习（**Meta-Learning**）

- " 学会学习"（Learning to Learn）
- 内循环：在具体任务上快速适应
- 外循环：跨任务优化学习算法
- 目标：few-shot 快速泛化

表示学习（**Representation Learning**）

- 学习任务无关的通用特征
- 降低原始感知维度
- 提取语义层次信息
- 促进跨任务知识迁移

## 关键价值

减少每个新任务所需的样本量，实现终身学习（Lifelong Learning）

# 模型无关的元学习（MAML）

**MAML** 算法 [5]

- 核心思想：寻找良好的参数初始化点 $\theta_0$
- 从 $\theta_0$ 出发，少量梯度步骤即可适应新任务
- 双层优化：
  - 内层：$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta)$
  - 外层：$\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}_{\mathcal{T}_i}(\theta_i')$

表示学习的前沿方向

- 对比学习：SimCLR、MoCo 自监督方法
- 世界模型：学习环境动力学隐表示
- 因果表示：识别状态因果变量

# 元学习与表示学习：研究趋势

## 主流范式

预训练大模型 + 下游任务微调（Pre-training + Fine-tuning）

代表性工作：

- 计算机视觉：自监督预训练（MoCo v3、MAE）
- 自然语言处理：GPT 系列、BERT 系列
- 强化学习：Decision Transformer、Gato（多任务智能体）

优势：

- 利用大规模无标注数据学习通用表示
- 显著降低下游任务的样本需求
- 实现跨任务、跨领域的知识迁移

# 目录

# 多智能体强化学习（MARL）

## 现实动机

- 真实世界里，其他参与者往往与我们同时存在并相互影响：对抗、协作以及竞争—协作的混合互动随处可见。
- 其他智能体本身就构成了我们所处的动态环境，使得环境分布随时间变化、呈现非平稳特性。
- 为我们研究群体智能、策略共演化、对手建模与机制设计提供了系统化的工具，也让智能体在社会性交互中催生出“创新”。

## 应用舞台

- 围棋/星际争霸、交通编队、仓储调度
- 多机械臂协同抓取、无人系统编队

# Dec-POMDP 模型

- 从建模角度看，MARL 通常用马尔可夫博弈来刻画：存在全局状态，若干智能体各自选择动作，形成联合动作并驱动联合转移，每个个体获得自己的回报。
- 很多场景是部分可观测的，智能体只能基于局部观测做决策。

## Decentralized Partially Observable Markov Decision Process

定义：Dec-POMDP 是一个八元组 $\mathcal{M} = \langle N, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, \{\mathcal{O}_i\}_{i=1}^n, P, O, R, \gamma \rangle$

- $N = \{1, \ldots, n\}$：智能体集合
- $\mathcal{S}$：全局状态空间
- $\mathcal{A}_i$：智能体 $i$ 的动作空间，联合动作 $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{A} = \prod_{i=1}^n \mathcal{A}_i$
- $\mathcal{O}_i$：智能体 $i$ 的观测空间，联合观测 $\mathbf{o} = (o_1, \ldots, o_n) \in \mathcal{O} = \prod_{i=1}^n \mathcal{O}_i$

# Dec-POMDP：转移、观测与回报

转移函数：

$$P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S}), \quad s_{t+1} \sim P(\cdot \mid s_t, \mathbf{a}_t)$$

观测函数：

$$O : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{O}), \quad \mathbf{o}_{t+1} \sim O(\cdot \mid s_{t+1}, \mathbf{a}_t)$$

每个智能体 $i$ 获得局部观测 $o_{i,t+1}$，通常 $O_i(o_i \mid s, \mathbf{a})$ 表示智能体 $i$ 的观测分布。

回报函数：

$$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, \quad r_t = R(s_t, \mathbf{a}_t)$$

协作场景下所有智能体共享同一回报；一般场景下可定义 $R_i(s, \mathbf{a})$。

初始分布： $s_0 \sim \rho_0(\cdot)$，折扣因子 $\gamma \in [0, 1)$

# Dec-POMDP：策略与目标

策略形式：

- 智能体 $i$ 的私有历史：$h_{i,t} = (o_{i,0}, a_{i,0}, o_{i,1}, a_{i,1}, \ldots, o_{i,t})$
- 策略：$\pi_i : \mathcal{H}_i \to \Delta(\mathcal{A}_i)$，其中 $\mathcal{H}_i$ 为历史空间
- 联合策略：$\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$
- 动作采样：$a_{i,t} \sim \pi_i(\cdot \mid h_{i,t})$

轨迹分布：

$$p_{\boldsymbol{\pi}}(\tau) = \rho_0(s_0) \prod_{t=0}^{T-1} \left[ \prod_{i=1}^{n} \pi_i(a_{i,t} \mid h_{i,t}) \right] P(s_{t+1} \mid s_t, \mathbf{a}_t) O(\mathbf{o}_{t+1} \mid s_{t+1}, \mathbf{a}_t)$$

优化目标：

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\tau \sim p_{\boldsymbol{\pi}}} \left[ \sum_{t=0}^{T-1} \gamma^t R(s_t, \mathbf{a}_t) \right]$$

# Dec-POMDP：信念状态与值函数

信念状态（**Belief State**）：

- 智能体 $i$ 维护关于全局状态的信念：$b_{i,t}(s) = P(s_t = s \mid h_{i,t})$
- 信念更新（Bayes 规则）：

$$b_{i,t}(s) \propto O_i(o_{i,t} \mid s, \mathbf{a}_{t-1}) \sum_{s'} P(s \mid s', \mathbf{a}_{t-1}) b_{i,t-1}(s')$$

- 基于信念的策略：$\pi_i(a_i \mid b_{i,t})$ 或 $\pi_i(a_i \mid h_{i,t})$

值函数：

- 状态值函数：$V^{\pi}(s) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t) \mid s_0 = s]$
- 状态-动作值函数：$Q^{\pi}(s, \mathbf{a}) = R(s, \mathbf{a}) + \gamma \sum_{s'} P(s' \mid s, \mathbf{a}) V^{\pi}(s')$
- 信念值函数：$V^{\pi}(b) = \sum_s b(s) V^{\pi}(s)$

# CTDE 范式：集中训练、分散执行

- 训练期（集中）：允许访问 $s_t$ 与 $\mathbf{a}_t$，使用中心化评论器/联合价值
- 部署期（分散）：个体策略 $\pi_i(a_i \mid o_{i,t})$ 独立执行，满足现实约束
- 代表方法：**VDN/QMIX/QPLEX**（值分解），**MADDPG/COMA/MAPPO**（策略梯度）

## 收益

缓解非平稳与信用分配，提高稳定性与可扩展性

# 自我博弈与策略族：Self-Play / PSRO

- **Self-Play/FSP/PFSP**：动态选择对手，控制训练难度
- **PSRO**：在策略集合层面近似纳什混合，持续寻找最优回应
- 联盟/种群训练（**PBT/League**）：维持多样化策略生态与稳定梯度

## 案例

AlphaStar 采用 PFSP+PBT 的联盟训练，面向“在多样对手前保持优势”的群体能力

# AlphaStar 训练机制

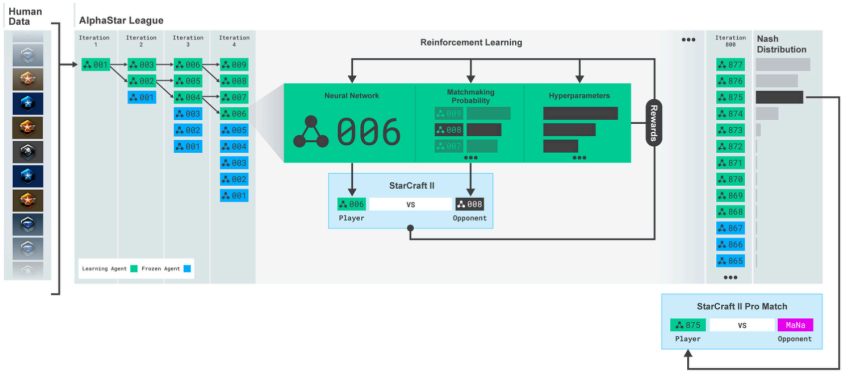



图 4: AlphaStar 的训练机制：每个小方块表示联盟中的一个智能体

- 联盟中的每个智能体用不同颜色方块表示
- 不同智能体探索不同策略区域

# 独立强化学习与价值分解

## 独立强化学习

独立 DQN、独立 PPO（IPPO）把其他智能体当作环境的一部分来学，简单好用、易扩展，但容易受到非平稳性的影响，是很好的起点，却往往需要进一步的稳定化设计。

## 价值分解

- **VDN**：把联合 $Q$ 近似为个体 $Q$ 的可加组合，即 $Q_{tot} = \sum_i Q_i$
- **QMIX**：通过单调性约束实现更灵活的可分解
- **QTRAN**、**QPLEX**：进一步放宽或学习化分解结构
- **Qatten**：带注意力的可学习分解结构

## 策略梯度方法

**MADDPG**（**Multi-Agent Deep Deterministic Policy Gradient**）：使用中心化评论器和去中心化策略兼顾稳定性与可部署性。训练时，每个智能体的评论器可以访问全局状态和所有智能体的动作，从而获得更准确的 Q 值估计；执行时，每个智能体仅基于自身局部观测独立决策，满足实际部署需求。这种方法有效缓解了多智能体环境中的非平稳性问题。

**COMA**（**Counterfactual Multi-Agent Policy Gradients**）：通过反事实优势刻画" 去掉某个体贡献后" 的增量收益，缓解信用分配难题。COMA 为每个智能体计算反事实基线，即假设该智能体采取默认动作时的期望回报，从而准确评估个体动作的边际贡献。这种方法在协作任务中能够有效区分每个智能体的实际贡献。

**MAPPO**、**HATRPO**、**FACMAC**：在多智能体设定下改造单体的稳定算法，使之更易调、更抗非平稳。MAPPO 将 PPO 扩展到多智能体场景，通过中心化价值函数和去中心化策略实现稳定训练；HATRPO 引入信任域方法保证策略更新的安全性；FACMAC 结合了 actor-critic 框架和值分解思想，在连续动作空间中表现优异。

## MARL 的核心挑战

**1**、非平稳性：队友与对手都在同时学习，经验分布持续漂移。解决思路包括 CTDE 架构、使用目标网络或阶段性冻结对手、显式对手建模与种群混合对手采样，以及分布鲁棒或最坏情况优化。

**2**、信用分配：个体贡献难以从全局回报中分离。反事实优势（如 COMA）、差分回报、价值分解（QMIX/QPLEX 等）以及可学习的分配权重（Qatten）是常用手段。

**3**、探索与可扩展性：联合动作空间指数级爆炸。层级控制、目标条件化探索、内在奖励与多样性驱动、以及分层规划都能改善样本效率与收敛质量。

**4**、强局部可观测与通信代价：需要在带宽、时延和鲁棒性之间折中，采用可学习通信协议、注意力选择通道与事件触发的稀疏通信能减少冗余而保持关键协调。

# 案例与未来展望

## AlphaStar

- AlphaStar 面临策略空间巨大、对手多样且环境先天非平稳的挑战，通过联盟训练维持策略多样性，用 PFSP 控制对手难度与训练梯度质量，并以混合策略减少过拟合单一路径，最终达到职业水平。
- 在其他环境里，如协作编队与猎捕任务（MPE/SMAC），价值分解与反事实优势在信用分配与多体协同上取得了系统性收益，使得智能体能够更好地协作与对抗。

## 从“应试者”到“创新者”

多智能体的社会性交互提供通向“创新”的路径：智能体在对抗与协作中相互推动，既把对手当作环境的一部分，也把自身的进步变为他人的新挑战。MARL 将这种共演纳入学习循环，结合博弈论与深度强化学习，正在复杂、动态、非平稳的真实问题上展现强大潜力。

# 目录

# 为什么需要模拟到现实（Sim-to-Real）

## 现实任务与挑战

- 深度强化学习在大规模模拟中已达或超越人类（如围棋），但现实任务仍具挑战。
- 现实应用：机器人控制、自动驾驶、无人机、工业控制等，安全性与准确性要求极高。
- 强化学习的探索在现实中代价大且有风险；工业仍大量依赖传统控制。
- 追求：用“聪明的智能体”操控物理系统，这一方向正被学术界与工业界积极推进。

## 动机

模拟训练成本低、速度快、可并行且安全；直接在现实训练通常效率低、成本高且存在安全隐患，因此需要“先模拟、再迁移”。

# 现实尝试与 Sim-to-Real 总览

## 直接在现实训练的代表

- **Guided Policy Search（GPS）**：用学习到的线性动态模型做轨迹优化，以较少交互学复杂技能（Levine et al., 2013；2018 并行化）。
- **QT-Opt** 抓取：7 台真实机器人分布式训练，约 **4** 个月**/800** 小时采样，展示了现实端到端学习的可行性与成本（Kalashnikov et al., 2018）。

## Sim-to-Real 方案与现实鸿沟

- **Sim-to-Real**：先在高效模拟中学习，再部署到现实（如 Akkaya et al., 2019；Andrychowicz et al., 2018）。
- 现实鸿沟（**Reality Gap**）：模拟与现实在动力学、感知、时序与噪声等方面存在系统差异，导致性能退化、失效或安全风险。

# 现实鸿沟的主要来源

## 1. 动力学与执行

- 摩擦/阻尼/弹性参数不准，接触与碰撞建模偏差
- 执行器非线性、饱和、死区与延迟

## 2. 感知与时序

- 传感器噪声、光照/遮挡、标定误差
- 观测构建/推理/通信/执行链路的累计延迟

## 3. 环境与不确定性

- 未建模扰动、几何/材质变化、长尾极端场景
- 任务/安全约束差异，现实操作规程限制

# 现实鸿沟：时间延迟导致的 MDP 差异

## 图 5

- 模拟中默认“零延迟”：状态采集、策略推理、动作执行同一时间步完成。
- 现实中存在感知/推理/执行链路延迟，导致观测滞后与动作迟到。
- 等效为带总延迟 $\delta$ 的闭环：实际策略更像 $\pi(A_t \mid O_{t-\delta})$。
- 同一控制信号会出现相位滞后与性能退化，是“现实鸿沟”的重要来源。

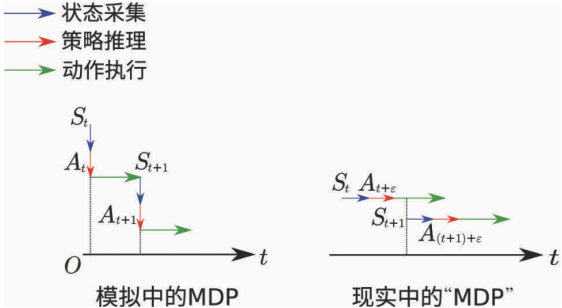



图 5: 时间延迟导致的 MDP 差异

# 现实鸿沟：机器人控制轨迹差异

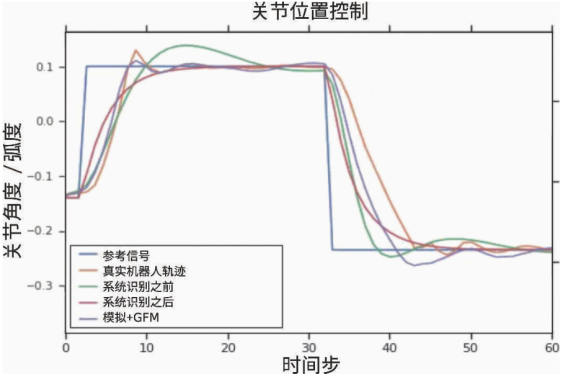



图 6: 机器人控制中的轨迹差异

## 图 6

- 展示参考信号（控制命令）、模拟轨迹与现实轨迹的差异。
- 延迟、惯性与动力学不准使得两者均偏离参考；模拟与现实的偏差即现实鸿沟。
- 系统识别（**SI**）：估计动力学参数以缩小差异，可用于策略或模拟器。
- **GFM**：向模拟器注入额外力校正，使模拟轨迹更贴近现实；但鸿沟仍可能残留。

# 现实鸿沟：轨迹分析

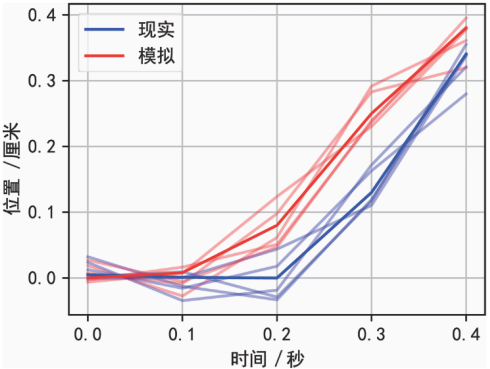



图 7: 物体位置的时间延迟

## 图 7

- 视觉定位/追踪需要时间，产生观测构建延迟，现实轨迹相对模拟出现时间间隙。
- 决策基于滞后观测 $O_{t-1}$：现实策略近似 $\pi(A_t \mid O_{t-\delta})$ 而非 $\pi(A_t \mid S_t)$。
- 可能的对策：在模拟器注入并随机化延迟；实时 RL 与“边运动边思考”提升连续时间下的平滑与稳定。

# 现实鸿沟：策略形式差异

## 策略形式差异

模拟：$\pi(A_t|S_t)$　　现实：$\pi(A_t|O_{t-\delta})$

- 模拟器中：策略基于当前精确状态 $S_t$
- 现实世界：策略基于延迟观测 $O_{t-\delta}$
- 时间延迟 $\delta$ 导致控制性能下降
- 需要考虑传感器延迟和执行延迟

# Sim-to-Real：问题与方法分类

## 核心问题

在模拟中训练得到的策略，因现实鸿沟（模拟与现实的系统差异）无法稳定用于现实，导致性能退化甚至失效。

## 两大类方法

**1)** 零样本（**Zero-Shot**）与 **2)** 自适应学习（**Domain Adaptation**）。

- 域自适应视角：源域（模拟）→ 目标域（现实）；假设跨域存在可共享特征，需高效利用少量或零现实数据。
- 代表方向（自适应）：元学习（Arndt'19; Nagabandi'18）、残差策略（Johannink'19; Silver'18b）、渐进网络（Rusu'16a,b）。

## 零样本：域随机化（Domain Randomization）

将源—目标差异视为源域中的随机性，学到对多种扰动鲁棒的通用策略：

- 动力学随机化（Peng'18）：随机化质量、摩擦、阻尼、力矩/速度噪声等。
- 视觉随机化（Sadeghi'16; Tobin'17）：随机纹理、光照、相机、物体布局等，支持纯视觉策略零样本落地。

## 前提与动机

现实鸿沟依赖具体任务：可能由动力学参数/过程定义差异引起，单纯随机化有时不足，需要识别 + 条件化策略 + 校准的组合。

## 系统识别与动力学敏感策略

- 系统识别（**SI**）：用少量真实轨迹估计质量、惯性、摩擦、关节刚度、执行器时延等参数；更新模拟器或策略上下文（Yu'17; Zhou'19）。
- **Dynamics-Aware** 策略：学习以系统特征为条件的策略 $\pi(a \mid o, c)$，其中 $c$ 由 SI 或轨迹编码器（RNN/Transformer）提取；部署时可 **few-shot** 自适应更新 $c$，或与残差策略叠加做小步校正。
- **GFM** 力校正：在仿真中添加校正力 $f_{corr}(s, a; \theta)$ 以最小化模拟—现实轨迹差异（Jeong'19b），与 SI 协同进一步缩小鸿沟。

## 跨域表征与结构迁移

- **Sim-to-Sim（RCANs）**：随机/现实图像 → 标准型（**canonical**）渲染域，感知模块只在标准域工作；训练可用无配对对抗/一致性约束，部署时输入现实图像、输出标准域特征（James'19）。
- 渐进网络（**Progressive Nets**）：通过冻结旧列 + 侧向连接复用低层视觉/控制特征，再叠加新列适配新域/任务，避免灾难性遗忘；将模拟中学到的低级表征迁移到现实或新任务，组合式构建复杂技能（Rusu'16a,b）。

## 意义

当今的计算框架利用离散的基于二值运算的计算过程，因此在某种程度上，我们应当始终承认模拟和现实世界的差异。这是因为后者在时间和空间上是连续的（至少在经典物理系统中）。只要学习算法不足够高效而能够直接人脑一样应用于现实世界（或者即便可以实现），在模拟环境中得到一些预训练模型也总是有用的。如果模型在一定程度上有对现实环境的泛化能力就会更好，而这是模拟到现实迁移算法的意义。换句话说，模拟到现实迁移算法提供了始终考虑到在现实鸿沟下的学习模型方法论，而无关于模拟器本身有多精确。

# 目录

# 大规模强化学习的挑战

可扩展性（**Scalability**）问题

大规模应用案例：

- AlphaStar（星际争霸 II）
- OpenAI Five（刀塔 2）

关键技术：

- 模仿学习
- 种群训练
- 自我博弈框架

**现实**

当前算法对大规模任务仍不够高效

需要结合其他技术：

- 预训练策略
- 监督学习
- 多智能体协作

并行训练框架

| 算法 | 特点 |
| --- | --- |
| A3C | 异步优势 Actor-Critic |
| DPPO | 分布式近端策略优化 |
| R2D2 | 循环缓存分布式 DQN |
| IMPALA | 重要性加权行动者-学习者结构 |
| SEED | 可扩展高效深度强化学习 |

**QT-Opt** 案例 [6]

- 7 个机器人并行采样
- 在线 + 离线数据的经验回放缓存
- 分布式训练工作者
- 关键：平衡不同计算设备（CPU/GPU）

# 目录

# 其他重要挑战

**1.** 可解释性与可信 **AI**

- 黑盒问题：决策过程不透明
- 注意力可视化：理解策略关注特征
- 因果推理：区分相关性与因果性

**2.** 安全强化学习（**Safe RL**）

- 约束满足：训练中保证安全约束
- 风险敏感决策：考虑方差和尾部风险
- 保护屏障：运行时安全监督
- 分布外检测：识别异常状态

**3.** 理论基础

- 收敛性、样本复杂度理论保证

# 目录

# 痛苦的教训（The Bitter Lesson）

## Richard Sutton 的核心观点（2019）

利用计算能力的通用方法最终胜过利用人类知识的专用方法

两个可无限扩展的方法：

- 1. 搜索：系统性探索可能性空间
- 2. 学习：从数据中提取模式

## 历史证据

- 国际象棋：深度搜索击败专家规则（1997）
- 围棋：MCTS+ 深度学习（AlphaGo, 2016）
- 计算机视觉：深度学习超越手工特征（2012+）

# 元方法的重要性

## 第二个教训

" 构建人类思维方式的内容到 AI 中在短期有效，但长期会形成障碍。"

应避免的做法：

- 硬编码特定领域知识
- 预定义的对称性和不变性

应追求的方向：

- 元学习：学习如何学习
- 架构搜索：自动设计神经网络

## 启示

投资于可扩展的计算和学习方法

# 本章总结

八大核心挑战：

1. 样本效率
2. 训练稳定性
3. 灾难性遗忘
4. 探索问题
5. 元学习与表示学习
6. 多智能体强化学习
7. 模拟到现实
8. 大规模强化学习

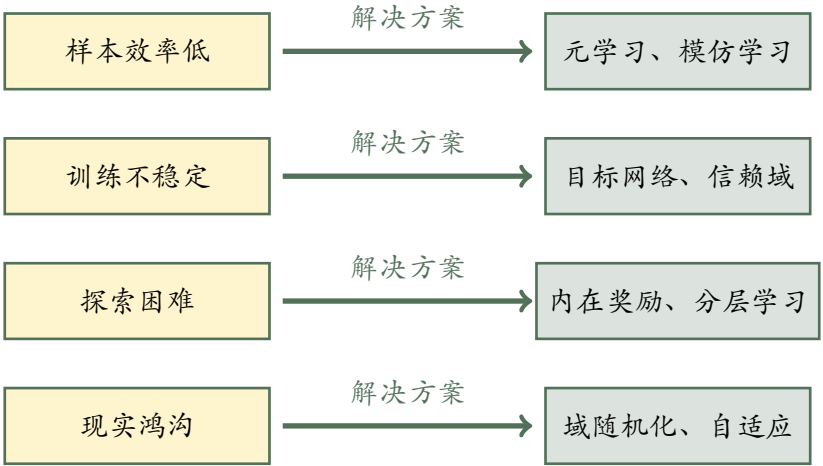
解决思路：

- 算法改进
- 架构创新
- 辅助技术
- 理论突破

## 未来方向

- 通用型方法
- 元学习机制
- 安全可靠
- 高效可扩展

# 挑战与机遇并存


样本效率低
解决方案
元学习、模仿学习
训练不稳定
解决方案
目标网络、信赖域
探索困难
解决方案
内在奖励、分层学习
现实鸿沟
解决方案
域随机化、自适应

# 研究展望

## 短期目标（1-3 年）

- 提高样本效率到实用水平
- 增强训练稳定性和可重复性
- 改进探索机制
- 缩小现实鸿沟

## 长期愿景（5-10 年）

- 通用强化学习智能体
- 真正的终身学习系统
- 安全可靠的现实世界部署
- 人机协同智能

# 参考文献 I


[1] Deisenroth, M., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. ICML.

[2] Henderson, P., Islam, R., Bachman, P., et al. (2018). Deep reinforcement learning that matters. AAAI.

[3] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., et al. (2017). Overcoming catastrophic forgetting in neural networks. PNAS.

[4] Vinyals, O., Babuschkin, I., Czarnecki, W. M., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature.

[5] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. ICML.

[6] Kalashnikov, D., Irpan, A., Pastor, P., et al. (2018). QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv:1806.10293.

# 参考文献 II


[7] Houthooft, R., Chen, X., Duan, Y., et al. (2016). VIME: Variational information maximizing exploration.

[8] Andrychowicz, M., Wolski, F., Ray, A., et al. (2017). Hindsight experience replay. NIPS.

[9] Berner, C., Brockman, G., Chan, B., et al. (2019). Dota 2 with large scale deep reinforcement learning. arXiv:1912.06680.

[10] Espeholt, L., Soyer, H., Munos, R., et al. (2018). IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures.

# 附录：主要算法对比 I

| 算法 | 类型 | 优势 | 劣势 |
|---|---|---|---|
| DQN | 离线策略 | 稳定性好 | 仅适用离散动作 |
| DDPG | 离线策略 | 连续动作 | 训练不稳定 |
| TD3 | 离线策略 | 改进 DDPG 稳定性 | 计算开销大 |
| PPO | 在线策略 | 简单有效 | 样本效率低 |
| TRPO | 在线策略 | 稳定更新 | 计算复杂 |
| SAC | 离线策略 | 鲁棒性好 | 超参数敏感 |
| A3C | 在线策略 | 并行训练 | 收敛慢 |

# 附录：主要算法对比 II

选择建议：

- 离散动作空间：DQN 系列
- 连续动作空间：SAC、TD3、PPO
- 需要稳定性：PPO、TRPO、SAC
- 样本效率优先：SAC、TD3
- 分布式训练：A3C、IMPALA、SEED
- 简单易用：PPO

# 附录：实践建议

**1.** 提高样本效率

- 使用经验回放（Replay Buffer）
- 考虑模仿学习预训练
- 尝试基于模型的方法

**2.** 增强训练稳定性

- 使用多个随机种子
- 调整学习率和批大小
- 采用目标网络和软更新

**3.** 改进探索

- 设计好的奖励函数（避免稀疏奖励）
- 使用内在奖励机制
- 考虑分层强化学习

# 附录：调试技巧

常见问题与解决方案：

**1.** 学习曲线不上升
- 检查奖励函数设计
- 降低学习率
- 增加探索

**2.** 训练过程崩溃
- 使用梯度裁剪
- 减小更新步长
- 检查网络初始化

**3.** 性能不稳定
- 增大批大小
- 使用目标网络
- 调整折扣因子

# 感谢观看！

深度强化学习

**Deep RL**