Chapter7-MARL&Sim2Real

7.6 多智能体强化学习

1

在前面章节里，我们默认环境中只有一个智能体去与环境交互、收集回报并寻找最优策略，这属于典型的单智能体强化学习。

可是在真实世界里，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。
此时，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。

现代学习算法更多的是出色的受试者（Test-Takers），而非创新者。智能体的智能上限可能受到其所在环境的限制。因此，创新的产生成为人工智能（Artificial Intelligence，AI）中一个较热的话题。多智能体强化学习（Multi-Agent Reinforcement Learning，MARL）正是在这样的背景下提出，它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”——不是由环境设计者预设“如何击败对手或如何合作”，而是让这些能力在长期对局与协作中自发涌现。

举例来说，古老的围棋游戏的发明者从未定义什么策略能够击败对手，而对手通常也构成了动态环境的一部分。然而，在一代又一代人类玩家或人工智能体的自我演化过程中，先进的策略被发明出来，每个智能体作为其他人环境的一部分，而对自身的提高也构成他人的新挑战。不仅仅是围棋，星际争霸、交通编队、仓储调度与多机械臂协同抓取，都是这类问题的应用舞台。

2

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。这里对dec-pomdp模型进行简单介绍。

Dec-POMDP可以定义为一个八元组模型。其中N代表智能体集合，包含从1到n的所有智能体。S是全局状态空间，包含了环境所有可能的状态。每个智能体i都有自己的动作空间A_i，所有智能体的动作组合成联合动作a。同样，每个智能体i也有自己的观测空间O_i，所有智能体的观测组合成联合观测o。

3

接下来我们详细了解模型的动态过程。转移函数P定义了在给定当前状态和联合动作下，下一状态的概率分布。

观测函数O决定了在状态转移后，各个智能体获得局部观测的概率分布。回报函数R为系统提供评估标准，在协作场景下所有智能体共享同一回报。
“初始分布”回答的问题是：整个任务或故事开始时，环境处于什么状态？定义了在第一个时间步（t=0），整个多智能体系统所处的真实状态 s_0有多大的可能性是某一个特定状态。即初始状态 s_0 是从分布 ρ_0 中随机采样出来的。

4

在策略层面，每个智能体i基于自己的私有历史来选择动作，这个历史包含了该智能体过去的所有观测和动作。所有智能体的策略组合成联合策略π。
输入是智能体 i的**私有历史**$h_{i,t}$。这个历史包括它从开始到现在看到的所有观测值 o和采取过的所有动作 a。因为其他智能体的动作和环境的真实状态它都看不到，所以这是它做决策的**唯一依据**。输出不是一个具体的动作，而是一个在动作空间 A_i上的**概率分布**$\Delta(A_i)$。$\Delta(A_i)$表示所有可能动作的概率集合（每个动作概率≥0，且所有概率之和为1）。

整个系统的轨迹分布由初始状态分布、各智能体策略、状态转移函数和观测函数共同决定。

- **a) 智能体们做出动作**：`∏ π_i(a_{i,t} | h_{i,t})`
  - 在状态 `s_t`下，每个智能体 i根据自己的策略和历史，选择动作 `a_{i,t}`。这个动作被选中的概率就是 `π_i(a_{i,t} | h_{i,t})`。
  - 最前面的连乘 ∏表示**所有智能体的动作概率要乘在一起**，因为联合动作 `a_t`是所有智能体同时行动的结果。
- **b) 环境状态发生改变**：`P(s_{t+1} | s_t, a_t)`
  - 这是**状态转移函数**。在联合动作 `a_t`作用于当前状态 `s_t`后，环境以一定的概率转移到下一个状态 `s_{t+1}`。
- **c) 智能体们收到新的观测**：`O(o_{t+1} | s_{t+1}, a_t)`
  - 这是**观测函数**。环境进入新状态 `s_{t+1}`后，每个智能体会收到一个新的观测值 `o_{i, t+1}`。这个函数给出了收到某个联合观测 `o_{t+1}`的概率。

优化目标是最大化期望累积折扣回报，即让智能体在长期运行中获得最大的累积奖励。

- **期望** `E_{τ ~ p_π} [...]`：
  - 由于环境和决策的随机性，每次运行任务都会得到一条不同的轨迹 τ，从而得到不同的累计回报。
  - 我们不能只看某一次运行的运气，要看“平均表现”。所以，优化目标是计算**在策略 π下，所有可能轨迹的累计回报的平均值（期望值）**。
- **累计回报** `Σ γ^t R(s_t, a_t)`：
  - `R(s_t, a_t)`是**即时奖励**，衡量在时刻 t，团队在状态 `s_t`下执行联合动作 `a_t`的好坏。
  - Σ表示从开始 `t=0`到结束 `t=T−1`的奖励求和。
  - γ^t是**折扣因子**（0 < γ ≤ 1）。它意味着：
    - **眼前的奖励比未来的奖励更值钱**（γ < 1 时）。
    - 它确保了无限时长任务（T→∞）的总奖励不会趋于无穷大，从而可以比较。

5

为了处理部分可观测性，智能体会维护关于全局状态的信念，这个信念表示给定历史后对当前状态的概率估计。

信念状态会根据贝叶斯规则进行更新，结合之前的信念、状态转移和新的观测信息。基于这种信念，智能体可以制定更有效的策略。

为了评估策略的好坏，我们定义了多种值函数，包括状态值函数、状态-动作值函数和信念值函数，它们从不同角度衡量策略的长期性能。

6

一个非常重要且实用的范式是“集中训练、分散执行”（CTDE）：训练阶段允许使用全局信息、联合价值函数或中心化评论器，以稳定学习并处理非平稳；部署阶段则要求每个体只依赖自身的局部观测与策略独立执行，从而满足现实约束。

7

在方法层面，首先是自我博弈与对手采样。经典的Self-Play、Fictitious Self-Play（FSP）以及优先级FSP（PFSP）通过动态选择对手控制训练难度，避免策略陷入局部最优并维持学习有效性。
进一步的PSRO（Policy-Space Response Oracles）把策略学习提升到“策略集合”的层面，近似计算纳什混合，对每个体不断寻找对当前混合策略的最优回应，促使策略族朝均衡收敛。
DeepMind 的 AlphaStar 是这一思路的代表性系统：

8

如图4所示，它通过种群式的联盟训练（PBT）维持多样化的策略族，并用PFSP匹配对手难度，既保证探索广度，也保证训练稳定，最终达到职业水平。在这种框架里，优化的单位不再是单一策略，而是联盟中一组策略的整体表现，学习目标是“在多样化对手面前保持优势”的群体能力。

9

其次是“独立强化学习”这一强基线。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。

协作任务中，价值分解是提升可扩展性的关键路径：VDN把联合Q近似为个体Q的可加组合；QMIX通过单调性约束实现更灵活的可分解；QTRAN、QPLEX以及带注意力的Qatten进一步放宽或学习化分解结构，这些方法在团队协作、信用分配复杂的任务里表现突出。

10

策略梯度家族中，MADDPG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；
COMA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；
MAPPO、HATRPO、FACMAC等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。这些方法都通过CTDE架构有效处理了多智能体学习中的非平稳性和信用分配问题。

11

这些方法之所以被发明出来，是为化解MARL的核心挑战。
第一，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。
第二，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。
第三，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。
第四，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。

12

落地到案例，我们先看AlphaStar。它面临策略空间巨大、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。在更具学术可控性的环境里，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。

回到开头，现代学习系统长期以来更像优秀的“应试者”，而非“创新者”。多智能体社会性交互为我们提供了一条通向“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把这种共演过程系统化地纳入学习循环，结合博弈论与深度强化学习，在复杂、动态、非平稳的真实问题上逐步展现出强大的潜力。

7.7 模拟到现实

1

强化学习方法可以成功地解决大模拟环境中的任务，甚至在一些具体领域可以超过最好的人类表现，比如围棋游戏。然而，应用强化学习方法到现实任务上的挑战仍旧未被解决。

除了雅达利游戏、策略性计算机游戏、纸牌游戏，强化学习在现实世界中的潜在应用包括机器人控制、车辆自动驾驶、无人机自动控制等。这些涉及现实世界中硬件的任务通常对安全性和准确性有较高要求。

对于这些情况，一个误操作可能导致灾难性后果。当策略是通过强化学习方法学到的时候，这个问题就更加值得考虑，因为即便不考虑现实世界的采样效率，学习智能体的探索过程也会有巨大影响。

现代工业中的机器控制仍旧严重依赖传统控制方法，而非最先进的机器学习或强化学习解决方案。然而，用一个聪明的智能体来控制这些物理机械仍旧是一个很好的追求，而大相关领域的研究人员正为之努力。

动机

2

近年来，深度强化学习被逐渐应用到越来越多的控制问题中。但是由于强化学习算法较高的样本复杂度以及其他一些物理限制，许多在模拟中展示的能力尚未在现实世界中复现。

指导性策略搜索（Guided Policy Search， GPS） (Levine et al., 2013) 是一种能够直接用真实机器人在有限时间内训练的算法。通过所学线性动态模型进行轨迹优化，这个方法能够以较少的环境交互学会复杂的操作技巧。研究人员也探索了用多个机器人进行并行化训练的方法 (Levine et al.,2018)。
文献 (Kalashnikov et al., 2018) 提出能同时在 7 个真实机器人上进行分布式训练的 QT-Opt算法，但是需要持续 4 个月的 800 个小时的机器人数据采样时间作为代价。他们成功示范了直接在现实世界部署的机器人学习，但是其时间消耗和资源上的要求一般是无法接受的。更进一步来说，直接在物理系统上训练策略的成功例子尚且只在有限的领域得到验证。

模拟到现实迁移（Sim-to-Real Transfer）则是可以替代直接在现实中训练深度强化学习智能体的方法，由于模拟性能的提升和一些其他原因，模拟到现实迁移的方法比之前受到更多注意。相比于直接在现实世界中训练，模拟到现实迁移可以通过在模拟中快速学习来实现。近年来，许多模拟到现实的方法成功将强化学习智能体部署到现实中 (Akkaya et al., 2019; Andrychowicz et al.,2018)。然而，相比于直接在现实环境中部署训练过程，模拟到现实的方法也有它本身的缺陷，这主要由模拟和现实环境的差异造成，称为现实鸿沟（Reality Gap）。

3

在实践中有大因素会导致现实鸿沟，而这由具体系统而定。举例来说，系统动力学过程的差异将导致模拟和现实的动力学鸿沟，

念一遍

4

如图 5所示是一个例子。

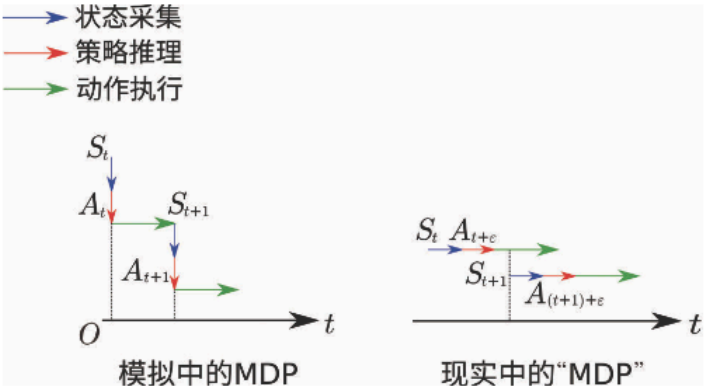

图 7.4　图片展示了模拟和现实中 MDP 的差异，它是由状态采集和策略推理过程产生的时间延迟造成的，这是造成现实鸿沟的可能因素之一（见彩插）

下面按“左图-右图-差异与含义-一个直观例子”来仔细解读图 7.4。

- 左侧：理想的“模拟中的 MDP”
- 纵向蓝/红/绿三箭头几乎在同一点发生，表示每个控制周期内三个过程“同时、无延迟”完成：
- 蓝：状态采集 S_t 发生在时刻 t。
- 红：策略推理立刻用 S_t 计算出 A_t。
- 绿：动作执行立刻把 A*t 作用到环境，环境转移到 S*{t+1}。
- 含义：在离散时间步的理想仿真里，观测、决策、执行被视为瞬时完成，策略等价于 π(A_t | S_t)。
- 右侧：现实中的“MDP”（带延迟的闭环）
- 三种颜色的箭头都有明显的“水平位移”，表示各环节存在时间延迟 ε：
- 蓝：获得状态/观测并不在 t 瞬时，而是要到 t+ε 才得到与 S_t 对应的信息（感知/通信/构建观测耗时）。
- 红：策略推理需要计算时间，导致得到动作变为 A_t 在 t+ε 才产生。
- 绿：执行器也有响应/传输/机械滞后，使真正作用到环境的是 A*t 在 t+ε 时刻才生效，下一步 A*{t+1} 也是在 (t+1)+ε 才执行。
- 含义：现实里智能体做决策时，用到的是“滞后观测”，动作也“迟到”才被执行。等效为带总延迟 δ 的控制链路：δ = δ*obs（观测）* + δ_pol（推理） + δ_act（执行），策略更像 π(A_t | O{t−δ})，而非 π(A_t | S_t)。
- 这幅图要表达的“现实鸿沟”核心
- 仿真默认零延迟，现实存在非零延迟，造成“时序错位（phase lag）”。同样的控制信号在现实与仿真产生不同轨迹，从而出现性能下降、振荡甚至不稳定，这正是模拟到现实迁移的关键障碍之一。

总结：图 7.4 用颜色区分“采集-推理-执行”三个环节，并通过水平位移直观展示“现实中的延迟”。左图零延迟的理想 MDP与右图带延迟的现实闭环之间的时序错位，就是“现实鸿沟”的重要来源。

现实应用中的现实鸿沟可以在某种程度上用文献 (Jeong et al., 2019b) 中的图 6 来理解，该图展示了机器人上模拟轨迹和现实轨迹的差异，以及模拟和参考信号的差异。对于强化学习进行机器人控制任务来说，参考信号是发送给智能体的控制信号，从而在机械臂的关节角度上获得预期的行为。由于延迟、惯性和其他动力学上的不准确性，模拟和现实中的轨迹都会与参考信号有显著差异。

此外，现实中的轨迹与模拟中的不同就是现实鸿沟。图中的系统识别（System Identification）是一种确认系统中动力学参数值的方法，可以用在策略或者模拟器中来缩模拟动力学过程和现实的差异。泛化力模型（Generalized Force Model，GFM）是一个在论文 (Jeong et al., 2019b) 中新提出的方法，可以用额外的力来校正模拟器，从而生成与现实更接近的模拟轨迹。然而，即使使用了识别和校正的方法，现实鸿沟依然可能存在，从而影响策略从模拟到现实中迁移。

图 7.5 在一个简单的关节角度控制过程中，机器人控制的参考信号、模拟和现实中的差异。图片改编自文献 (Jeong et al., 2019b)（见彩插）

6

除了由于不同动力学过程导致的一个时间步上模拟现实轨迹的差异，现实鸿沟也有其他来源。比如，在连续的现实世界控制系统中，有系统响应时间延迟或系统观察构建过程耗时，而这些在有离散时间步的理想模拟情况下可能都不存在。

如前面的图5所示，在模拟环境或传统强化学习设置下，状态采集和策略推理过程都认为是始终没有时间损耗的，而在现实情况下，这两个过程都可能需要相当的时间，这使得智能体总是根据先前动作执行时的先前状态产生的滞后观察来进行动作选择。

上面的问题也会使得模拟和现实的轨迹展现出不同的模式，**如图 7 所示。** 考虑一个物体操作任务，即使我们假定有很快的神网络前向过程（Forward Process）而忽略策略推理的时间消耗，现实世界中物体位置也可能需要一个摄机来捕捉并用一些定位技术来追踪，而这需要相当的时间来处理。这个观察构建的过程会引入时间延迟，从而即使在完全相同的控制信号下，现实轨迹和模拟轨迹的对比图上也会展示出时间间隙。

图 7 图片展示了物体观察状态（位置）在同一控制信号下的时间延迟。由于现实中额外的观察量构建过程，现实世界轨迹（下方）相比于模拟轨迹（上方）有一定延迟。不同的线体现了多次测试结果，加粗的线为均值（见彩插）

这类延迟观察量使得现实世界中的强化学习智能体只能够接受先前观察 O_{t-1} 来对当前步做出动作选择 A_t，而非直接根据当前状态 S_t。因此实践中的策略根据时间延迟 δ通常会有形式 $\pi(A_t|O_{t-\delta})$，而这不同于模拟中根据实时观察训练的策略，从而会产生较差的现实表现。

一种解决这个问题的方式是修改模拟器，使其有相同的时间延迟，从而训练智能体去学习。然而，这会导致其他的问题，比如如何精确地表示和测模拟和现实中的时间延迟，如何保证基于延迟观察学习的智能体的表现等。近来，文献 (Ramstedtet al., 2019) 提出了实时强化学习方法，文献 (Xiao et al., 2020) 提出了“边运动边思考（ThinkingWhile Moving）”的方法，在连续时间

MDP 设置下轻了强化学习对于实时环境中延迟观察和并发动作选（Concurrent Action Choices）的问题，使得在现实世界中的控制轨迹更加平滑。

8

如上所述，从强化学习角度来看模拟到现实迁移的主要问题在于：在模拟中训练得到的策略由于现实鸿沟不能在现实世界中始终正常使用，这个现实鸿沟即模拟和现实的差异。由于这个模型的差异，模拟环境中的成功策略无法很好地迁移到相应的现实中。

9

经过上面的分析 **核心问题**

总体来说，解决模拟到现实迁移的方法可以分为至少两个大类：零样本（Zero-Shot）方法和自适应学习方法。

将控制策略从模拟迁移到现实的问题可以被看作是域自适应（Domain Adaption）的一个例子，即将一个在源域（Source Domain）中训练的模型迁移到新的目标域（Target Domain）。这些方法背后的一个关键假设是不同的域有公共的特征，从而在一个域中的表征方式和行为会对其他域有用。
域自适应要求新的域中的数据适应预训练的策略。在新的域中获取数据的复杂性或困难程度，比如在现实世界中收集样本，这种自适应学习方法因而需要有较高的效率。元学习 (Arndt et al., 2019; Nagabandi et al., 2018)、残差策略学习（Residual Policy Learning） (Johannink et al., 2019; Silver et al., 2018b) 和渐进网络（Progressive Networks） (Rusu et al., 2016a,b) 等方法被用于这些情形。

10

零样本 （Zero-Shot）迁移是一个与域自适应互补一类技术，它尤其适用于在模拟中学习。这意味着在迁移过程中没有任何基于现实世界数据的进一步学习过程。
域随机化（Domain Randomization）是零样本迁移中典型的一类方法。通过域随机化，源和目标域的差异被建模为源域中的随机性。通过域随机化可以学到更普适的策略，而非过拟合到具体模拟器设置的特征策略。根据具体的应用，随机化可以被施加到不同的特征上。
举例来说，对于机器人操作任务，摩擦力和质的大小、力矩和速度的误差在实际机器人上都会影响到控制的精度。因此，在模拟器中这些参数可以被随机化，从而用强化学习训练一个更鲁棒的策略 (Peng et al., 2018)，这个过程称为**动力学随机化（Dynamics Randomization）**。在**视觉域下的随机化**可以用于直接将基于视觉的策略从模拟迁移到现实，而不需要任何现实的图来训练 (Sadeghi et al., 2016; Tobin et al., 2017)。可能的视觉随机化的特征变包括纹理、光照条件和物体位置等。

11

现实鸿沟通常是依赖于具体任务的，它可能由动力学参数或者动力学过程的定义不同造成。除了动力学随机化 (Peng et al., 2018) 或视觉特征（观察）随机化，还有一些其他方法来跨越现实鸿沟。

系统识别 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

利用系统识别（System Identification）来学习一个对动力学敏感（Dynamics-Aware）的策略 (Yu et al., 2017; Zhou et al., 2019) 是一个有希望的方向，它试图学习一个以系统特征为条件的策略，这些系统特征包括动力学参数或者轨迹的编码。

也有一些方法来最小化模拟与现实的差异，比如之前介绍的 GFM 方法用于进行力校正，等等。

12

通过模拟到模拟（Sim-to-Real via Sim-to-Sim）(James et al., 2019) 是另一个跨过现实鸿沟的方法，它使用随机到标准自适应网络（Randomized-to-Canonical Adaptation Networks, RCANs）来将随机的或现实世界图转化成它们同等的非随机的标准型，而与模拟环境中的类似。
渐进网络 (Rusu et al., 2016a) 也可以用于模拟到现实迁移 (Rusu et al., 2016b)，这是一个普适的框架，重复利用任何低级视觉特征到高级策略中，从而迁移到新的任务上，它以一种组合式但是简单的方法来构建复杂技能。

13

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。