*Jingfeng*

# 强化学习

从 MDP 与深度控制到语言模型和智能体反馈学习

Jingfeng · 2026 年 10 月 22 日修订



从一个可手算的问题，走到可检验的学习系统。

本书围绕同一个问题展开：当行为改变未来数据，怎样定义目标、估计价值、改进策略，又怎样判断获得的反馈值得相信？从四状态 MDP 出发，推导与实现并行推进，直到深度控制、离线数据、多智能体以及语言模型的反馈学习。

| | |
|---|---|
| 读者 | 具备基础概率、微积分、线性代数与 Python 知识，希望阅读论文并开展实验。 |
| 主线 | 建模与估计；策略与控制；数据与约束；反馈与证据。 |
| 本次实测 | 四状态规划与 TD、风险路径上的表格控制、短轨迹 GAE/PPO 边界核验。 |
| 版本 | 原稿知识快照为 2026-09-03；本次完成全文审订、证据核对与书籍模板迁移。 |

# 前言：把反馈变成可以讨论的对象

如果一个系统会根据过去的结果改变行为，那么“给它一个分数”只是学习问题的开始。我们还要说明分数来自什么过程，行为改变了哪些未来机会，训练中见过的数据能否支持新策略，以及估计误差是否会被反复优化放大。强化学习提供了描述这些问题的语言，也提供了能在具体假设下分析的算法。

本书从一个四状态问题开始。它足够小，能够手算最优值、策略值、一次 TD 更新和奖励变换；它也足够完整，能够展示即时奖励与长期回报的冲突。之后每次引入新机制，都先交代原来的方法在哪个环节受限：未知模型需要样本估计，巨大状态空间需要函数逼近，连续动作需要新的策略参数化，固定数据需要覆盖假设，而偏好与验证器则要求重新审视“奖励就是目标”这一方便却危险的简化。

## 需要哪些准备

数学上需要理解条件期望、向量与矩阵、导数和简单概率分布。遇到随机逼近、重要性采样或 KL 散度时，正文会说明它们在当前问题中的含义；本书并不以这些说明替代完整的概率论和优化课程。编程上需要能阅读数组更新和 Python 循环。随书实测仅使用 CPU 上的表格算法，不要求 GPU、机器人设备或语言模型账号。

## 三条阅读路线

第一次接触 RL，可顺读第 1–4 章，再到第 12 章亲手复核数据。阅读每个更新式时回答三个问题：目标值使用谁的数据、哪一项不反向传播、何时可以 bootstrap。这些问题比记住算法简称更能防止实现错误。

已经实现过 PPO 或 SAC 的读者，可重点看第 3、4 章与附录，将公式逐项对照自己的掩码、概率密度和损失函数，再选择离线学习、模型规划、多智能体或约束学习专题。专题之间不构成一条必须按算法年代走完的路线，数据获取方式和决策结构才是选择依据。

关注语言模型与工具智能体的读者，应先理解价值估计、策略梯度和离线覆盖，再读第 11 章。一个回答可以被看成单步动作，也可以展开为 token 级轨迹；两种建模有不同的信用分配和终止语义。它们与通用 RL 相通，同时仍受奖励模型和验证器误差限制。

## 怎样理解本书的证据

书中把四类内容分开写明：数学推导、明确给定的教学数据、文献中的机制或报告、此次实际运行的实验。四状态和短轨迹数字有独立参考计算；风险路径实验有逐种子数据、固定训练预算与评估协议。本次没有运行深度控制训练，也没有微调语言模型；相关章节给出目标、训练流程和可执行验证方案。

本次修订日期为 2026 年 10 月 22 日。文献核验以可获取的原论文、出版社/会议页面和官方技术文档为依据，并保存条目及版本证据。日期表示本书核验时间，不表示覆盖了这一日期以前的全部研究。本书不把预印本的作者报告当成独立复现，也不据此宣布跨任务的最优方法。

## 版本与分册关系

原稿由 Codex 辅助编写，知识快照标为 2026 年 9 月 3 日；本版署名 Jingfeng，完成全文组织、公式与边界条件修订、教学算例增补和 Jingfeng 书籍版式迁移。原始源码及 PDF 保存在主题目录中，逐章变化、来源核验和限制另见修订说明。

《具身智能》承接机器人动作接口与数据工程；《大语言模型后训练》承接完整的偏好数据和训练流水线；《LLM 智能体系统》承接执行器、记忆与工具系统；本书可独立阅读。

## 全书结构




每一层都沿用前面的目标和符号，但会增加新的信息条件。阅读专题时，先找到它改变了哪一项假设，再判断哪些结论仍能沿用。

# 目录

**前言：把反馈变成可以讨论的对象** **i**

**全书结构** **ii**

**符号与终止约定** **vii**

**第 1 章　决策问题、价值函数与统一符号** **1**

1.1　交互顺序与建模边界 . . . . . 1
1.2　回报、折扣与任务结束 . . . . . 2
1.3　贯穿全书的四状态 MDP . . . . . 2
1.4　价值函数与 Bellman 算子 . . . . . 3
1.5　折扣访问分布与策略改变的数据 . . . . . 4
1.6　部分可观测与记忆 . . . . . 4
1.7　奖励与真正目标的关系 . . . . . 5

**第 2 章　从规划到样本估计：DP、MC 与 TD** **6**

2.1　已知模型时的规划基准 . . . . . 6
2.2　Monte Carlo：用完整经历评估固定策略 . . . . . 7
2.3　TD：用一步自洽关系学习 . . . . . 7
2.4　多步回报与资格迹 . . . . . 8
2.5　离策略估计与重要性采样 . . . . . 8
2.6　从评估走向控制：SARSA 与 Q-learning . . . . . 9
2.7　收敛条件与适用边界 . . . . . 10

**第 3 章　从策略梯度到 Actor–Critic 与 PPO** **11**

3.1　折扣目标的梯度从哪里来 . . . . . 11
3.2　Baseline、优势与 stop-gradient . . . . . 11
3.3　Actor–Critic 与双掩码 GAE . . . . . 12
3.4　性能差异、TRPO 与 surrogate 的近似 . . . . . 13
3.5　PPO clipping 做了什么 . . . . . 14
3.6　熵奖励与连续动作的衔接 . . . . . 15

**第 4 章　深度价值与连续控制** **16**

4.1　共同骨架：回放、延迟目标与梯度边界 . . . . . 16
4.2　离散动作：从 DQN 到 Double DQN . . . . . 17
4.3　连续动作：用 actor 代替显式最大化 . . . . . 17
4.4　SAC：让随机策略与软价值使用同一个目标 . . . . . 18

4.5 动作变换：执行的动作必须对应所算的密度 . . . . . 19
4.6 把五种算法放进同一个训练循环 . . . . . 20
4.7 诊断、部分可观测性与扩展边界 . . . . . 21

**第 5 章 获得探索信号、时间抽象与示范 22**
5.1 从随机动作到有目的地收集经验 . . . . . 22
5.2 奖励塑形：何时保持原问题不变 . . . . . 23
5.3 HER：重用失败经验需要哪些条件 . . . . . 23
5.4 Options：把决策的时间尺度写进 Bellman 目标 . . . . . 24
5.5 专家示范：先学习动作，再补足部署分布 . . . . . 24
5.6 从匹配行为到推断奖励 . . . . . 25
5.7 偏好反馈与方法选择 . . . . . 25

**第 6 章 模型、规划与误差累积 27**
6.1 Dyna：把真实经验与规划接起来 . . . . . 27
6.2 为什么一步准确仍会长期失真 . . . . . 27
6.3 PETS：在每个真实时刻重新规划 . . . . . 28
6.4 MBPO：模型用于训练，执行时依赖策略 . . . . . 28
6.5 潜变量模型：从观测历史到想象训练 . . . . . 29
6.6 MuZero：为搜索学习任务相关模型 . . . . . 29
6.7 按样本预算与执行接口选路线 . . . . . 30

**第 7 章 固定数据学习与反事实评估 31**
7.1 固定数据、行为策略与支持 . . . . . 31
7.2 先建立行为克隆与行为约束基线 . . . . . 31
7.3 CQL：对数据外估值施加压力 . . . . . 32
7.4 IQL：在数据动作内估值，再提取策略 . . . . . 32
7.5 序列建模改变了训练形式，没有消除覆盖 . . . . . 33
7.6 OPE：用日志估计未执行策略的价值 . . . . . 33
7.7 置信区间与选择偏差比点估计更重要 . . . . . 34

**第 8 章 多智能体：信息、协调与共同学习 35**
8.1 从联合 MDP 到部分可观测博弈 . . . . . 35
8.2 CTDE：训练特权信息不能泄漏到执行 . . . . . 35
8.3 价值分解及 QMIX 的表达边界 . . . . . 36
8.4 MAPPO 与算法选择的实际依据 . . . . . 36
8.5 自博弈：胜率取决于对谁比赛 . . . . . 37
8.6 一份能比较的多智能体评测协议 . . . . . 37

**第 9 章 约束、尾部风险与鲁棒决策 38**

9.1 成本约束首先要定义单位和时域 . . . 38
9.2 Lagrangian：可以调节违约代价，不能凭空保证可行 . . . 38
9.3 左尾 CVaR：最差一部分回合有多差 . . . 39
9.4 稀有事件的证据需要足够样本 . . . 39
9.5 鲁棒 MDP：环境不确定性放进哪个集合 . . . 39
9.6 从优化目标到部署边界 . . . 40
9.7 练习：一个约束是否真是所需约束 . . . 40

**第 10 章 表示、泛化与迁移 41**

10.1 先定义任务族，再定义泛化 . . . 41
10.2 表示应保留哪种决策信息 . . . 41
10.3 Successor features：哪些变化可以线性复用 . . . 42
10.4 Meta-RL：适应本身也是评测对象 . . . 42
10.5 非平稳部署与统计单位 . . . 43

**第 11 章 语言模型与智能体的反馈学习 44**

11.1 回答级与 token 级的两种建模 . . . 44
11.2 从偏好标签到奖励模型 . . . 44
11.3 KL 正则目标与 DPO 的推导边界 . . . 45
11.4 RLAIF 与验证器：更大规模不等于更少假设 . . . 45
11.5 GRPO：组内相对反馈怎样进入策略梯度 . . . 46
11.6 Agentic RL：外部状态让信用分配更具体 . . . 47
11.7 统一报告：反馈质量与优化收益一起看 . . . 47
11.8 练习：先分辨问题，再选择优化器 . . . 48

**第 12 章 实验、诊断与可复现性 49**

12.1 先说明证据回答什么问题 . . . 49
12.2 实验一：同一 MDP 的解析值与计算值 . . . 49
12.3 实验二：逐项核验短轨迹与截断 . . . 50
12.4 实验三：固定预算下的风险路径控制 . . . 52
12.5 从可运行代码到可核查记录 . . . 54
12.6 把小实验的诊断方法带到深度强化学习 . . . 55

**第 13 章 从算法选择到可证伪的研究计划 57**

13.1 先由环境与预算缩小方法范围 . . . 57
13.2 把训练循环拆成可诊断的四个边界 . . . 58
13.3 调参应追踪机制，而非背诵默认数字 . . . 58

13.4 把开放问题写成可以失败的研究设计 . . . . . 59
13.5 十二周路线：每周留下一个可以核对的对象 . . . . . 60

**附录 A 推导补充与练习提示** **61**
A.1 从 Bellman residual 到价值误差 . . . . . 61
A.2 性能差异引理的望远镜证明 . . . . . 61
A.3 KL 正则最优分布与支持限制 . . . . . 62
A.4 有限片段的 GAE 与 $\lambda$ 回报 . . . . . 62
A.5 练习解题提示：从数字回到条件 . . . . . 63

**附录 B 实现边界与报告核对表** **64**
B.1 转移进入缓冲区前 . . . . . 64
B.2 不同损失允许哪些梯度 . . . . . 64
B.3 按信息与预算选择起点 . . . . . 65
B.4 一次运行应留下什么 . . . . . 65

**参考文献** **66**

# 符号与终止约定

除特别声明外，本书以有界奖励、$0 \le \gamma < 1$ 的有限折扣 MDP 为理论起点。连续空间、有限时域、平均奖励和部分可观测任务会在相应章节另列条件。奖励统一记为 $r_t$：它发生在 $s_t$ 选择 $a_t$ 并转移到 $s_{t+1}$ 的这一步。这与部分文献记为 $R_{t+1}$ 的约定只差下标，不改变过程。

| 符号 | 含义及约定 |
|---|---|
| $\mathcal{S}, \mathcal{A}$ | 状态与动作空间；可用动作可随状态变化。 |
| $P(s' \mid s,a), R(s,a,s')$ | 转移概率和条件奖励均值；$\bar{r}(s,a)$ 表示再对下一状态取期望。 |
| $d_0(s)$ | 初始状态分布；比较策略时固定同一 $d_0$。 |
| $o_t, h_t, \beta_t(s)$ | 观测、可用历史及 POMDP 的状态信念；信念与下文标量 bootstrap 掩码 $b_t$ 分开记。 |
| $\pi(a \mid s), \mu(s)$ | 随机策略与确定性策略；离策略章节中的 $\mu(a \mid s)$ 可另指行为策略，届时明确。 |
| $G_t = \sum_{k \ge 0} \gamma^k r_{t+k}$ | reward-to-go（从当前步起的折扣回报）；终止后奖励为零。 |
| $J(\pi) = \mathbb{E}[G_0]$ | 从 $d_0$ 出发的期望折扣回报。 |
| $V^\pi, Q^\pi, A^\pi$ | 状态价值、动作价值与优势，$A^\pi = Q^\pi - V^\pi$。 |
| $d_\gamma^\pi(s)$ | 归一化折扣状态占用分布：$(1-\gamma)\sum_t \gamma^t \Pr(s_t = s)$。 |
| $\mu_\gamma^\pi(s,a)$ | 联合状态动作占用：$d_\gamma^\pi(s)\pi(a \mid s)$；不与行为策略 $\mu(a \mid s)$ 混用。 |
| $\theta, \phi, \bar{\phi}$ | actor 参数、critic 参数、延迟的目标 critic 参数；DQN 单网络另说明。 |
| $\alpha, \eta, \beta$ | 表格学习率（或 SAC 熵温度）、一般步长、KL 正则系数；$\alpha$ 的角色在算法上下文给出，温度可写 $\alpha_{\text{ent}}$。 |
| $\delta_t, \lambda$ | TD residual 与 trace/GAE 衰减系数。 |
| $d_t^{\text{term}}, d_t^{\text{trunc}}$ | 真正任务终止与外部时间限制截断标记。 |
| $b_t = 1 - d_t^{\text{term}}$ | bootstrap 掩码：真终止不接下一价值，外部截断通常仍接最终观测价值。 |
| $c_t$ | GAE 递推连接掩码：仅在下一条 residual 存在且属于同一回合时为 1；reset 或批末取 0。 |
| $\rho_t$ | 重要性比率；一般是目标/行为策略，PPO 中为新/旧策略的当前动作概率比。 |
| $D_{\text{KL}}(p\|q), \mathcal{H}(p)$ | KL 散度及熵；连续动作时使用相应密度与微分熵。 |
| $\text{sg}(x)$ | stop-gradient：数值等于 $x$，对优化变量的梯度为零。 |

**两个不同的“结束”问题**

价值目标问的是“此任务之后还有没有未来奖励”；GAE 的递推问的是“下一条样本还能否接在这条轨迹后面”。外部时间限制截断时，前者可能为是，后者因 reset 为否。有限时域任务如果把剩余时间放入状态，到时则是任务定义内的终止，不能机械地当作外部截断。见第 1、3 章和实际边界测试。

# 第 1 章　决策问题、价值函数与统一符号

强化学习首先是一个决策问题，然后才是一组更新公式。若动作会改变未来遇到的数据，单步预测准确并不足以说明策略好：一个调度器可以立即清空短作业，却让长作业无限等待；一个智能体可以反复触发计分器，却没有完成任务。本章先明确环境、信息和目标，再用同一个小型问题建立后文的价值估计、策略改进与实验参照。

阅读本章需要条件概率、期望、向量和无穷级数。遇到条件期望时，可以先在有限表格上把它读成加权平均；后面的神经网络只改变这些量的表示和估计方式，并不替代问题定义。基本定义和经典结果以 Sutton 与 Barto 的教材为基础 [1]。

## 1.1　交互顺序与建模边界

本书统一采用如下时间顺序：

$$s_t \longrightarrow a_t \sim \pi(\cdot \mid s_t) \longrightarrow (r_t, s_{t+1}).$$

$r_t$ 是动作 $a_t$ 引发的这一次转移的奖励，不是进入 $s_t$ 之前的奖励。轨迹可记为

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots).$$

其他资料经常把同一奖励写为 $R_{t+1}$；读论文或移植代码时，要先对齐时间索引。

在可完全观测、有限状态动作的默认设定下，折扣马尔可夫决策过程（Markov decision process, MDP）为

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, d_0, \gamma).$$

状态为 $s \in \mathcal{S}$，可用动作为 $a \in \mathcal{A}(s)$，初始状态分布为 $d_0$。转移核是 $P(s' \mid s, a)$；定义

$$R(s, a, s') = \mathbb{E}[r_t \mid s_t = s, a_t = a, s_{t+1} = s'], \qquad \bar{r}(s, a) = \sum_{s'} P(s' \mid s, a) R(s, a, s').$$

即使奖励本身有随机性，Bellman 方程也可以用其条件均值表达。写下 $P$ 和 $R$ 不意味着学习者已经知道它们：已知模型时可以规划，未知模型时需要用交互样本估计。

马尔可夫性要求

$$\Pr(s_{t+1}, r_t \mid s_0, a_0, \ldots, s_t, a_t) = \Pr(s_{t+1}, r_t \mid s_t, a_t).$$

它是关于所选状态是否包含预测未来所需信息的假设，而不是关于真实世界有没有记忆。例如队列长度相同，但剩余服务时间不同，下一时刻的完成情况也不同；只保留队列长度就可能不满足这一假设。改变状态表示会改变后续价值函数的含义。

| 问题 | 可考虑的状态与动作 | 必须写清的建模选择 |
|---|---|---|
| 控制 | 位置、速度；控制输入 | 传感器是否充分、碰撞是否终止 |
| 调度 | 队列与容量；分配或路由 | 等待成本、预算、未完成任务如何计分 |
| 推荐 | 用户上下文与历史；展示项目 | 点击之外的长期指标、用户状态变化 |
| 语言智能体 | 对话与工具历史；文本或工具操作 | 可见信息、奖励检查、任务结束条件 |

## 1.2 回报、折扣与任务结束

### 1.2.1 同一回报定义下的三类任务

若一个回合在 $T$ 次转移后结束，回报是

$$G_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k}, \qquad J(\pi) = \mathbb{E}_{s_0 \sim d_0, \pi}[G_0]. \tag{1.1}$$

对持续任务，将上限替换为无穷。本书涉及收缩性与无限和的默认理论条件为：有限状态动作、$0 \le \gamma < 1$、奖励几乎处处满足 $|r_t| \le R_{\max} < \infty$。因此 $|G_t| \le R_{\max}/(1-\gamma)$，期望、求和以及后面使用的算子有明确意义。

折扣改变目标，不能仅当作数值稳定参数。相隔 $k$ 步的同一奖励，其当前权重是 $\gamma^k$；$1/(1-\gamma)$ 给出一个常用的尺度参照，但不等于所有任务的真实规划长度。平均回报

$$\limsup_{T \to \infty} \frac{1}{T} \mathbb{E}_\pi \left[ \sum_{t=0}^{T-1} r_t \right]$$

则是另一类目标，需要遍历性等额外条件，本书不把折扣算法的证明直接移植过去。

有限时域任务可以取 $\gamma = 1$，但价值通常必须写成 $V_t(s)$，或把剩余步数加入状态。在这种情形，有限次后向递推有意义，却不能再用“折扣收缩”解释收敛。一个时间上限若属于任务定义，例如“十步内完成”，剩余时间就是决策信息；若只是采样器为了便于训练而中断一个持续任务，它并没有使环境价值突然归零 [2]。

### 1.2.2 真终止与截断使用不同语义

真终止（termination）表示任务按模型结束，例如到达目标或发生吸收失败。可引入吸收状态 $s_\perp$，满足

$$P(s_\perp \mid s_\perp, a) = 1, \qquad R(s_\perp, a, s_\perp) = 0, \qquad V(s_\perp) = 0.$$

这样有限回合的回报可以等价写成无限和。截断（truncation）表示数据采集停止，环境中的决策过程可能仍会继续。记

$$d_t^{\text{term}} \in \{0, 1\}, \qquad b_t = 1 - d_t^{\text{term}}, \qquad y_t = r_t + \gamma b_t V(s_{t+1}). \tag{1.2}$$

这里 $s_{t+1}$ 是该转移真正产生的最后观测。向量环境若自动 reset，接口返回的观测可能已经来自下一回合；必须另外取得 final observation，不能把新回合的初始价值接到旧回合末端。

**一个边界有两种问题**

“下一状态是否仍有未来价值”决定 bootstrap 掩码 $b_t$；“缓冲区里的下一条 TD residual 是否属于同一条连续轨迹”决定第 3 章的递推掩码 $c_t$。外部时间上限处通常 $b_t = 1$，但因下一条数据已经 reset，$c_t = 0$。真终止处两者都为零。

这一区分也有适用条件。若截断后的状态不可取得，不能假装已正确 bootstrap；应修复采样接口，或明确记录该估计的缺失。若时间上限本就是任务终点，则它应在扩展状态的任务模型中作为终止处理。

## 1.3 贯穿全书的四状态 MDP

考虑三个非终端状态 $s_0, s_1, s_2$ 与一个终端状态 $s_\perp$。初始分布集中在 $s_0$，转移和奖励均确定。仅 $s_0$ 有选择，其余状态只有一个动作。

| 当前状态 | 动作 | 下一状态 | 奖励 |
|---|---|---|---|
| $s_0$ | direct | $s_\perp$ | 2 |
| $s_0$ | via | $s_1$ | 0 |
| $s_1$ | next | $s_2$ | 1 |
| $s_2$ | finish | $s_\perp$ | 3 |

参考策略在 $s_0$ 以 1/2 概率选择 via。取 $\gamma = 0.9$，从后向前得到

$$\begin{aligned} V^\pi(s_2) &= 3, \qquad V^\pi(s_1) = 1 + 0.9 \times 3 = 3.7, \\ Q^\pi(s_0, \text{direct}) &= 2, \qquad Q^\pi(s_0, \text{via}) = 0.9 \times 3.7 = 3.33, \\ V^\pi(s_0) &= \tfrac{1}{2} \times 2 + \tfrac{1}{2} \times 3.33 = 2.665. \end{aligned} \tag{1.3}$$

因此最优策略选择绕行，$V^*(s_0) = 3.33$。它没有在第一步取得奖励，却获得更高的折扣总回报。若将 $\gamma$ 改为 0.5，绕行值变成 $0.5 + 3(0.5)^2 = 1.25$，直接结束反而更好。两条路径的优劣在 $\gamma + 3\gamma^2 = 2$，即 $\gamma = 2/3$ 时交换。

这个算例区分了三件事：固定策略的价值为 2.665，最优价值为 3.33，某条直接结束样本的回报只有 2。不能把某一次样本、策略期望和最优解混为一谈。第 2 章在这张表上执行规划和采样更新，第 12 章保存实际计算与独立参考检查。

## 1.4 价值函数与 Bellman 算子

固定平稳策略 $\pi$，定义

$$V^\pi(s) = \mathbb{E}_\pi[G_t \mid s_t = s], \qquad Q^\pi(s,a) = \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a].$$

由 $G_t = r_t + \gamma G_{t+1}$ 和全期望公式，得到

$$V^\pi(s) = \sum_a \pi(a \mid s) \left[ \bar{r}(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V^\pi(s') \right], \tag{1.4}$$

$$Q^\pi(s,a) = \bar{r}(s,a) + \gamma \sum_{s'} P(s' \mid s,a) \sum_{a'} \pi(a' \mid s') Q^\pi(s',a'). \tag{1.5}$$

式中已经包含零奖励吸收状态；样本实现可以等价地使用式 (1.2)，显式屏蔽真终止转移。定义优势 $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$，则 $\sum_a \pi(a \mid s) A^\pi(s,a) = 0$。在算例的 $s_0$，两个优势分别为 $-0.665$ 和 0.665。

对任意候选函数 $V$，定义期望算子和最优算子

$$(\mathcal{T}^\pi V)(s) = \sum_a \pi(a \mid s) \left[ \bar{r}(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s') \right], \tag{1.6}$$

$$(\mathcal{T}^* V)(s) = \max_a \left[ \bar{r}(s,a) + \gamma \sum_{s'} P(s' \mid s,a) V(s') \right]. \tag{1.7}$$

在本章的有限、有界、$\gamma < 1$ 条件下，两者都将有界函数映射为有界函数。对最优算子，利用 $|\max_a x_a - \max_a y_a| \le \max_a |x_a - y_a|$，有

$$\begin{aligned} |(\mathcal{T}^* V)(s) - (\mathcal{T}^* W)(s)| &\le \max_a \gamma \left| \sum_{s'} P(s' \mid s,a)(V(s') - W(s')) \right| \\ &\le \gamma \|V - W\|_\infty. \end{aligned}$$

取状态最大值便得到 $\gamma$ 收缩；期望算子将最大值换成策略加权平均即可同样证明。完备空间上的收缩映射各有唯一不动点，分别是 $V^\pi$ 与 $V^*$，并且

$$\|(\mathcal{T}^*)^k V_0 - V^*\|_\infty \le \gamma^k \|V_0 - V^*\|_\infty. \tag{1.8}$$

有限动作保证最大值能取到；一般连续空间需要额外的可测性与存在性条件。若 $\gamma = 1$，上述不等式至多给出非扩张性，不能继续声称有唯一不动点或同样的误差界。神经网络拟合还引入投影与优化误差，收缩性也不自动传递给完整训练算法。

## 1.5 折扣访问分布与策略改变的数据

价值函数从一个状态看未来，访问分布则从整个策略看会访问哪些状态。包括吸收扩展在内，定义归一化折扣状态分布

$$d_\gamma^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr_\pi(s_t = s), \qquad \mu_\gamma^\pi(s,a) = d_\gamma^\pi(s)\pi(a \mid s). \tag{1.9}$$

由于每一时刻的状态分布和为 1，$\sum_s d_\gamma^\pi(s) = 1$。若只保留未终止状态，这个和一般小于 1，不能仍把它当成已归一化分布。由定义直接得到

$$J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\mu_\gamma^\pi}[\bar{r}(s,a)]. \tag{1.10}$$

因子 $1/(1-\gamma)$ 来自归一化，后面的策略梯度和性能差异公式也使用同一约定。

四状态参考策略的非终端折扣质量依次为

$$d_\gamma^\pi(s_0) = 0.1, \quad d_\gamma^\pi(s_1) = 0.045, \quad d_\gamma^\pi(s_2) = 0.0405.$$

余下的 0.8145 在吸收状态。折扣单步奖励期望为 $0.1\times(0.5\times 2) + 0.045\times 1 + 0.0405\times 3 = 0.2665$，除以 0.1 正好回到 $J(\pi) = 2.665$。

令 $P^\pi(s' \mid s) = \sum_a \pi(a \mid s)P(s' \mid s, a)$，分布满足流量方程

$$d_\gamma^\pi(s') = (1-\gamma)d_0(s') + \gamma\sum_s d_\gamma^\pi(s)P^\pi(s' \mid s).$$

策略改变既改变当前动作比例，也改变未来的状态分布。只在旧数据上比较两个策略的单步动作收益，会漏掉第二个变化；第 3 章的 surrogate 近似和离线学习中的数据支持问题都从这里产生。

## 1.6 部分可观测与记忆

部分可观测 MDP（POMDP）把真实状态与观测分开：学习者看到 $o_t \sim O(\cdot \mid s_t)$，可执行的策略依赖历史 $h_t = (o_0, a_0, r_0, \dots, a_{t-1}, r_{t-1}, o_t)$，不能偷偷访问评估时不存在的真实状态。若模型已知，belief $\beta_t(s) = \Pr(s_t = s \mid h_t)$ 是一个充分统计量；当观测不含额外奖励信息时，其更新具有形式

$$\beta_{t+1}(s') \propto O(o_{t+1} \mid s')\sum_s P(s' \mid s, a_t)\beta_t(s).$$

更一般地，应把实际观察到的奖励也纳入观测似然。完整 belief 空间通常连续且难以精确维护。

帧堆叠、状态估计器、循环网络和 Transformer 记忆都试图保留有用历史，但有限记忆不自动成为精确 belief。一个简单反例是：两个隐藏状态显示完全相同的画面，却要求相反动作；任何无记忆观测策略都无法同时正确。工程上应先检查历史长度、reset 后隐状态、训练与评估的可见信息是否一致，再归因于优化器失效。

## 1.7 奖励与真正目标的关系

奖励是对目标的形式化承诺。把完成、时间、能耗、违规等项相加前，应说明每项的单位和量级，并同时记录原始指标。若某个要求是预算或约束，改成一个固定大惩罚后得到的是不同问题；即使训练回报更高，也不代表满足约束。

在四状态算例中，只给终点奖励可以表达“最终完成”，加上每步成本会改变对路径长度的偏好。在回合结束时停止计分的任务里，对所有非终端转移加常数也可能改变最优路径，因为两条路径获得常数的次数不同。相反，持续折扣任务中给包括吸收自环在内的每一步都加同一常数，只会给所有策略的回报加上相同的 $c/(1-\gamma)$。奖励变换是否保策略，取决于完整任务约定。

潜势型塑形 $\gamma\Phi(s_{t+1})-\Phi(s_t)$ 是一种有条件的策略不变构造，第 5 章会用望远镜和式说明终端边界。这里先保留一个实践原则：训练奖励与独立成功判据要分别报告。日志还应能够辨认重复计分、拖延结束、操纵观测等反例，而不能只保存一条总回报曲线。

**检查建模是否真的闭合**

把四状态算例的 $\Pr(\text{via}\mid s_0)$ 改为 $p$。证明 $J(p)=2+p(\gamma+3\gamma^2-2)$，再求 $p=1/2,\gamma=0.5$ 时的参考价值。提示：先分别写出两条完整路径的回报；结果应为 1.625。接着解释为什么一次在 $s_1$ 外部截断的样本不应把 $V(s_1)$ 改成零。最后检查：若任务真的限定最多一次转移，原问题是否还拥有相同的状态定义？

下一章暂时假定状态可以枚举。这样可以把“模型已知时怎样计算”和“只有样本时怎样估计”分开，并用解析答案检查每一次更新。

# 第 2 章　从规划到样本估计：DP、MC 与 TD

Bellman 方程给出了价值的自洽关系，但“知道方程”还不等于“知道答案”。本章按信息来源组织方法：有完整模型时计算条件期望；只有完整回合时平均回报；只有短片段时用当前估计补上未来。最后再让估计影响动作选择，得到控制算法。这个顺序把价值误差、采样误差和策略变化分开，便于判断一次实验究竟出了什么问题。

## 2.1　已知模型时的规划基准

### 2.1.1　策略评估与策略改进

固定策略的矩阵形式是

$$V^\pi = r^\pi + \gamma P^\pi V^\pi, \qquad V^\pi = (I - \gamma P^\pi)^{-1} r^\pi.$$

这里包含零奖励吸收状态，或等价地删除该状态并固定其价值为零。$\gamma < 1$ 时，逆矩阵存在，可由几何级数 $\sum_{k\ge 0} \gamma^k (P^\pi)^k$ 理解。显式求逆在大问题上既昂贵又没有必要；迭代 $V_{k+1} = \mathcal{T}^\pi V_k$ 就能逐步逼近。

在四状态 MDP 中，从全零函数开始做同步更新，即一轮内所有右侧都使用同一个 $V_k$，得到下表。$V_k^\pi$ 表示策略评估迭代的中间值，$U_k$ 表示价值迭代的中间值；它们不是每轮都已经准确的真实价值。

| 轮次 $k$ | $V_k^\pi(s_0)$ | $V_k^\pi(s_1)$ | $V_k^\pi(s_2)$ | $U_k(s_0)$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 3 | 2 |
| 2 | 1.45 | 3.7 | 3 | 2 |
| 3 | 2.665 | 3.7 | 3 | 3.33 |

价值信息每轮向前传播一步：第二轮 $s_1$ 已经看到终点奖励，但 $s_0$ 还只看到第一轮的 $s_1$。这个无环例子三轮就精确；一般有环 MDP 通常是渐近收敛。原地更新可以更快传播，但中间结果依赖状态遍历顺序，不能拿同步表逐项比较。

策略改进选择对 $V^\pi$ 贪心的策略 $\pi'$。由 $\mathcal{T}^{\pi'} V^\pi \ge \mathcal{T}^\pi V^\pi = V^\pi$，再利用算子单调性和重复评估，可得 $V^{\pi'} \ge V^\pi$。策略迭代交替执行评估和改进；价值迭代则每轮直接使用 $\mathcal{T}^*$。前者每次改策略前估得更充分，后者把评估和改进交织起来。

**Listing 2.1**　有限 MDP 的同步价值迭代

```
V = zeros(number_of_states)
while True:
    old = V.copy()
    for s in nonterminal_states:
        values = []
        for a in actions[s]:
            values.append(sum(p * (r + gamma * old[next_s])
                              for p, next_s, r in model[s, a]))
        V[s] = max(values)
    V[terminal_state] = 0.0
    if max_abs(V - old) <= tolerance:
        break
```

程序的停止阈值需要解释。若已计算 Bellman residual $\|\mathcal{T}^*V - V\|_\infty \le \varepsilon$，则由收缩性

$$\|V - V^*\|_\infty \le \frac{\varepsilon}{1-\gamma}.$$

相同的数值阈值在 $\gamma = 0.9$ 和 0.999 时对应不同的价值误差上界。这个界依赖完整模型与所有状态上的最大残差，不能用一个小样本批次的平均 TD error 代替。

## 2.2 Monte Carlo：用完整经历评估固定策略

Monte Carlo（MC）以完整回合回报为目标：

$$V(s_t) \leftarrow V(s_t) + \alpha_n(s_t)\big[G_t - V(s_t)\big]. \tag{2.1}$$

在固定策略、正确奖励与终止语义下，给定当前状态的完整回报满足 $\mathbb{E}[G_t \mid s_t = s] = V^\pi(s)$。这解释了样本目标的含义，并不意味着任何有限样本平均、任意停样规则或任意变动策略都无偏。

First-visit MC 每个回合只使用某状态的首次访问，every-visit MC 使用所有访问。回合内多次访问是相关样本；两者都需要相应的大数条件，不能把每次访问当作独立实验。对本书的有界折扣任务，回报有界；要实际得到完整回报，还需要回合最终结束。持续任务没有自然终局，有限片段直接求和会漏掉尾部。

四状态参考策略从 $s_0$ 出发的回报只可能是 2 或 3.33，概率各半。因此

$$\mathbb{E}[G_0] = 2.665, \qquad \mathrm{Var}(G_0) = 0.665^2 = 0.442225.$$

若独立运行 $N$ 个回合，样本均值方差为 $0.442225/N$。这是该确定性环境中由随机动作造成的回报方差；环境转移随机时还会增加其他来源。MC 不使用未来价值估计，代价是必须等到奖励到达，并承受整条后续轨迹的随机性。

## 2.3 TD：用一步自洽关系学习

时间差分（temporal difference，TD）把下一个状态的当前估计代入 Bellman 方程：

$$\delta_t = r_t + \gamma b_t V(s_{t+1}) - V(s_t), \qquad V(s_t) \leftarrow V(s_t) + \alpha_n(s_t)\delta_t. \tag{2.2}$$

若 $V$ 尚不准确，给定 $s_t = s$ 的 TD 目标期望是 $(\mathcal{T}^\pi V)(s)$，不是直接等于 $V^\pi(s)$。这种以现有估计构造目标的做法称为 bootstrap。它使信息可以在线传播，也把当前估计误差带入目标；但在适当条件下，表格 TD 的不动点仍是 $V^\pi$，不能据此说 TD 的最终答案必然有偏。

取算例中的转移 $s_1 \to s_2$，奖励 1。若更新前 $V(s_1) = 1, V(s_2) = 2$，$\alpha = 0.5$，则

$$y = 1 + 0.9 \times 2 = 2.8, \quad \delta = 2.8 - 1 = 1.8, \quad V_{\text{new}}(s_1) = 1.9.$$

解析值 3.7 没有在一次更新中凭空出现：下一状态的低估和步长都限制了这一次变化。反之，使用已准确的 $V(s_2) = 3$，目标才是 3.7。

“MC 方差高、TD 方差低”是常见倾向，不能作为脱离条件的定理。若后续回报本来确定、而 bootstrap 网络误差很大，TD 目标也可能更不可靠。选择方法时要看奖励延迟、模型随机性、估计误差和可获得的片段长度，而不是只看一个偏差方差标签。

## 2.4 多步回报与资格迹

### 2.4.1 有限片段中的 $n$ 步与 $\lambda$ 回报

设从 $t$ 开始最多有 $m$ 条连续转移，到达真终止或采样边界。对 $1 \le n \le m$，且前 $n$ 步没有更早终止，定义

$$G_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k r_{t+k} + \gamma^n B_{t+n} V(s_{t+n}). \tag{2.3}$$

$B_{t+n}$ 在第 $n$ 步真正终止时为零，否则为一。若轨迹提前终止，应停止累计并去掉尾部，而不是继续读取 reset 后数据。若 $m$ 只是 rollout 边界，$G_t^{(m)}$ 仍需要尾部价值。

在同一个固定价值函数下，有限片段的前向 $\lambda$ 回报为

$$G_t^{\lambda} = (1-\lambda) \sum_{n=1}^{m-1} \lambda^{n-1} G_t^{(n)} + \lambda^{m-1} G_t^{(m)}. \tag{2.4}$$

权重和等于 1。最后一项吸收了剩余权重；直接把一个无限几何混合在终点截断，会丢掉权重。$\lambda = 0$ 给出一步 TD，$\lambda = 1$ 给出该片段最长的回报。只有片段到达真终止，后者才是完整 MC。

绕行轨迹在 $s_0$ 的三个多步目标分别为

$$G_0^{(1)} = 0.9V(s_1), \quad G_0^{(2)} = 0.9 + 0.81V(s_2), \quad G_0^{(3)} = 0.9 + 0.81 \times 3 = 3.33.$$

当初值为零时，一步目标暂时看不到奖励，二步看到 0.9，三步看到整条路径。增加步数让远端奖励更快进入一次更新，同时也纳入更多采样随机性。

### 2.4.2 后向观点：哪些过去状态应一起更新

表格累积资格迹（eligibility trace）为每个状态保留一个衰减权重：

$$e_t(s) = \gamma\lambda e_{t-1}(s) + \mathbf{1}\{s = s_t\}, \tag{2.5}$$

$$V(s) \leftarrow V(s) + \alpha\delta_t e_t(s) \quad \text{对所有 } s. \tag{2.6}$$

一个较晚到达的 TD residual 因而能够更新刚访问过的多个状态，不必为每个起点显式保存所有多步回报。回合开始将迹清零；真终止或外部截断/reset 后，也不能把旧迹带到新回合。

前向混合与后向迹的等价性需要仔细说明：若一段数据的价值参数固定，累计的离线更新可以相互对应；普通在线 TD($\lambda$) 在每一步修改参数后，后续 residual 也随之变化，有限步长下不再逐项等于那个固定参数的前向表达。真在线方法处理了这种差异。深度 Actor–Critic 常先固定采样时的价值估计，再离线计算优势，正好便于采用明确的前向定义。

## 2.5 离策略估计与重要性采样

行为策略 $\mu$ 收集数据，目标策略 $\pi$ 是要评估的对象。若 $\pi$ 可能选的动作在 $\mu$ 下概率为零，数据无法直接支持对应反事实。重要性采样（importance sampling，IS）至少要求：对目标可达历史和动作，$\pi(a \mid s) > 0$ 蕴含 $\mu(a \mid s) > 0$，且环境动力学与奖励机制相同。

对一条终止回合，从状态 $s_t$ 起的轨迹比率为

$$\rho_{t:T-1} = \prod_{j=t}^{T-1} \frac{\pi(a_j \mid s_j)}{\mu(a_j \mid s_j)}.$$

在相应可积条件下，$\mathbb{E}_\mu[\rho_{t:T-1}G_t \mid s_t = s] = V^\pi(s)$。若评估 $Q^\pi(s_t, a_t)$，当前动作已经条件固定，比率应从 $t+1$ 开始。逐决策形式把每个奖励只乘上影响它的前缀比率：

$$V^\pi(s) = \mathbb{E}_\mu\left[\sum_{k=0}^{T-t-1} \gamma^k \rho_{t:t+k} r_{t+k} \,\middle|\, s_t = s\right].$$

它避免让很晚的动作概率给较早奖励增加无关噪声。

普通 IS 用 $N^{-1}\sum_i \rho_i G_i$；自归一化或加权 IS 用 $\sum_i \rho_i G_i / \sum_i \rho_i$，分母为零时必须另行处理。后者一般存在有限样本偏差，在适当条件下仍一致，往往降低极端权重带来的不稳定。长轨迹上的乘积权重可能迅速膨胀；截断权重可降低方差，却改变估计目标或引入偏差。离线评估章将进一步讨论数据覆盖和置信区间。

在四状态 MDP 中，假设行为策略以 1/2 走绕行，而目标总是绕行。从 $s_0$ 评估时，绕行样本的比率为 2，直接结束样本为 0。普通 IS 的单样本估计于是为 6.66 或 0，期望为 3.33，但方差比仅沿目标策略采样大得多。支持条件成立并不等于估计精确。

## 2.6 从评估走向控制：SARSA 与 Q-learning

SARSA 的名字来自五元组 $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$。用实际行为的下一动作构造目标：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_t + \gamma b_t Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right]. \tag{2.7}$$

它是 on-policy 更新：下一动作从当前用于交互的策略采样。真终止后不需要采样动作；外部截断若仍需 bootstrap，必须能够在真正最后状态计算相应下一动作或策略期望。

Q-learning 将下一步换成贪心值：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_t + \gamma b_t \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right]. \tag{2.8}$$

数据可以来自探索策略，目标却对应最优 Bellman 算子，所以是 off-policy 控制 [3]。它不需要给这个一步最大值显式乘上整条轨迹 IS 权重；这是特定 Bellman 更新的结构，不表示任意离策略多步回报都能忽略分布差异。

**Listing 2.2** 表格控制中目标动作的区别

```
action = epsilon_greedy(Q[state], epsilon)
next_state, reward, terminated, truncated = step(action)
if terminated:
    target = reward
elif algorithm == "sarsa":
    next_action = epsilon_greedy(Q[next_state], epsilon)
    target = reward + gamma * Q[next_state, next_action]
else:
    target = reward + gamma * max(Q[next_state])
Q[state, action] += alpha * (target - Q[state, action])
# Reset after terminated or truncated; use the final state above.
# In a full SARSA loop, reuse next_action for the next real action.
```

在有危险边缘的路径任务中，持续 $\epsilon$ 探索会让靠近危险区域的行为承担跌落成本。SARSA 的目标包含这一探索行为，Q-learning 的贪心目标则不包含下一步随机探索。由此可能出现路径差异；但 SARSA 没有显式优化风险度量，更没有逐步安全保证。探索强度、奖励、训练预算与评估时是否探索都能改变结论。第 12 章实际对照两者的训练回报、评估回报和路径选择，不预设胜负。

## 2.7 收敛条件与适用边界

对每个被更新的状态或状态动作对，以其本地访问次数 $n$ 计步长。经典随机逼近条件是

$$\sum_n \alpha_n = \infty, \qquad \sum_n \alpha_n^2 < \infty.$$

前一条避免过早停止学习，后一条控制累计噪声。固定学习率适合追踪变化，却不满足这个几乎处处收敛条件。

固定策略的表格 TD(0) 需要充分访问待估状态、合适步长以及有界奖励和折扣等条件，收敛到该策略的价值。表格 Q-learning 需要每个相关状态动作对无限更新，且采样、噪声和步长满足定理要求，才收敛到 $Q^*$。表格 SARSA 的最优收敛通常进一步采用 GLIE：无限探索，同时在极限变得贪心。只说“$\epsilon$ 随时间下降”不够；若下降太快，一些动作可能永远没有再访问的机会。状态本来不可达，也不能靠调整 $\epsilon$ 使其得到数据。

| 方法 | 目标所依赖的信息 | 首先检查的条件 |
|---|---|---|
| DP | 完整转移与奖励期望 | 模型是否正确、同步或原地更新 |
| MC | 真实完整回报 | 策略是否固定、回合是否真正结束 |
| TD 与多步 | 样本加尾部价值 | bootstrap 误差、终止与片段边界 |
| SARSA | 实际下一动作价值 | 行为策略、探索日程与访问次数 |
| Q-learning | 下一状态贪心价值 | 所有动作覆盖、最大值估计误差 |

函数逼近、bootstrap 与离策略数据共同出现时，表格收缩分析不再足以保证训练稳定。第 4 章的回放与目标网络缓和了若干误差来源，但不构成一般收敛证明。先让一个实现通过小 MDP 的解析检查，比直接投入大型环境更容易发现索引和掩码错误。

**把更新规则变成可检查的计算**

先从零值函数计算绕行轨迹在 $\lambda = 1/2$ 下的 $G_0^\lambda$。提示：三个权重为 $1/2, 1/4, 1/4$，得到 $0.25 \times 0.9 + 0.25 \times 3.33 = 1.0575$。再用 $\alpha = 1/2$ 执行一次从 $V(s_0) = 0$ 开始的更新。最后构造一个行为策略永远直接结束的例子，说明为什么无论收集多少这样的数据，都不能仅凭 IS 估出绕行奖励。建议用第 12 章的独立路径枚举结果核对自己的代码。

# 第 3 章　从策略梯度到 Actor–Critic 与 PPO

价值型控制先估计每个动作的后果，再选择动作；策略梯度直接调整动作分布的参数。当动作连续、策略必须随机化，或想对一个已有策略做局部改进时，这个视角尤其自然。本章沿着一个问题展开：怎样把一次轨迹的反馈变成一个目标明确、方差可控、更新幅度可诊断的梯度？

## 3.1　折扣目标的梯度从哪里来

设策略 $\pi_\theta(a \mid s)$ 可微，环境与初始分布不依赖 $\theta$。先考虑有限轨迹，并假定策略支持与参数微分满足交换微分和期望所需的条件。轨迹密度为

$$p_\theta(\tau) = d_0(s_0) \prod_t \pi_\theta(a_t \mid s_t) P(s_{t+1}, r_t \mid s_t, a_t).$$

对本书的折扣目标 $J(\theta) = \mathbb{E}[\sum_k \gamma^k r_k]$，对数导数恒等式给出

$$\nabla_\theta J = \mathbb{E}\left[\left(\sum_k \gamma^k r_k\right) \sum_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right].$$

动作 $a_t$ 不影响之前已发生的奖励。条件化于过去，score 的期望为零，因此可以去掉每个 score 之前的奖励：

$$\nabla_\theta J = \mathbb{E}\left[\sum_t \gamma^t G_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]. \tag{3.1}$$

无限轨迹还需支配性等额外条件保证上述交换合法。把条件回报换为 $Q^{\pi_\theta}$，并采用第 1 章的归一化 occupancy，就得到策略梯度定理的形式

$$\nabla_\theta J = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\gamma^{\pi_\theta}, a \sim \pi_\theta}\left[Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a \mid s)\right]. \tag{3.2}$$

REINFORCE 用样本回报代替动作价值 [4]；策略梯度定理说明不必显式对状态分布求导，但状态分布的影响已经包含在右侧的访问分布中 [5]。

**不要混用两种时间权重**

若以完整回合估计 $J = \mathbb{E}[\sum_t \gamma^t r_t]$，式 (3.1) 中有外层 $\gamma^t$，不能因为 $G_t$ 内部已有折扣就再删掉它。若直接从 $d_\gamma^\pi$ 抽状态，这个权重已进入采样分布。很多连续任务实现把 rollout 中的时刻等权使用，这是对实际采样分布构造的常用近似，不能再无条件声称它就是前述 episodic 折扣目标的无偏梯度。

在四状态 MDP 中，令 $p = \pi_\theta(\text{via} \mid s_0) = \text{sigmoid}(\theta)$。前章已经得到 $J(p) = 2 + 1.33p$，故

$$\frac{\partial J}{\partial \theta} = 1.33 p(1-p).$$

在 $p = 1/2$ 时梯度是 0.3325。用 score 直接计算也应得到同一结果：绕行 score 为 $1-p$，直接结束 score 为 $-p$，于是 $p \times 3.33(1-p) + (1-p) \times 2(-p) = 1.33p(1-p)$。这是检查策略梯度符号和采样权重的独立参照。

## 3.2　Baseline、优势与 stop-gradient

### 3.2.1　为什么减去状态函数不改期望梯度

对任何不依赖当前动作的状态函数 $b(s)$，

$$\mathbb{E}_{a \sim \pi_\theta(\cdot \mid s)}[b(s) \nabla_\theta \log \pi_\theta(a \mid s)] = b(s) \nabla_\theta \sum_a \pi_\theta(a \mid s) = 0.$$

因此可以用 $G_t - b(s_t)$ 替代回报。常用 $b = V_\phi$，以“比该状态下的通常结果好多少”代替绝对分数。四状态参考策略在 $s_0$ 的优势为 $\pm 0.665$；两条路径都获正奖励，但直接结束应降低概率。

状态价值是方便且有效的 baseline，却不是一般意义上方差最小的标量 baseline。令 $g = \nabla_\theta \log \pi_\theta(a \mid s)$，只考虑固定状态下单个向量估计 $(G - b)g$ 的方差迹。因为其均值与 $b$ 无关，最小化二阶矩得到

$$b^*(s) = \frac{\mathbb{E}[G\|g\|^2 \mid s]}{\mathbb{E}[\|g\|^2 \mid s]}, \tag{3.3}$$

分母非零。只有当相关权重满足相应条件时，它才等于 $\mathbb{E}[G \mid s]$。完整轨迹还有跨时刻协方差，不能把这个局部结论再夸大成整个梯度估计的通用最优解。

### 3.2.2 梯度应该经过哪条路径

actor 的最小化损失可写为

$$L_\pi = -\sum_t \gamma^t \log \pi_\theta(a_t \mid s_t)\ \mathrm{sg}(\hat{A}_t).$$

sg 表示数值保持不变、求导时视为常数。若优势包含 critic，甚至与 actor 共享编码器，也不应从策略损失对优势本身求导，否则多出不属于上述 score 估计的项。critic 则可以通过自己的回归损失更新共享编码器。共享参数没有取消两种损失各自的数学角色。

同理，存储的旧策略 log probability、采样时的旧价值和训练目标都应冻结。critic 损失例如

$$L_V(\phi) = \frac{1}{2}\mathbb{E}_t\left[V_\phi(s_t) - \mathrm{sg}(\hat{R}_t)\right]^2.$$

baseline 若含参数，其作为 baseline 出现在策略损失中时仍需 stop-gradient；“减 baseline 不改变期望”的恒等式并没有要求去微分 baseline。

## 3.3 Actor–Critic 与双掩码 GAE

Actor–Critic 用 critic 估计未来，用 actor 改策略 [6]。一步优势估计是

$$\delta_t = r_t + \gamma b_t V_{\mathrm{old}}(s_{t+1}) - V_{\mathrm{old}}(s_t), \qquad b_t = 1 - d_t^{\mathrm{term}}.$$

若 critic 恰好等于 $V^\pi$，给定 $(s_t, a_t)$ 的 residual 期望等于 $A^\pi(s_t, a_t)$；critic 有误时，一步目标也随之有误。策略在变、数据有限、函数逼近不准确，使 actor 与 critic 形成耦合，而不是两个完全独立的监督学习问题。

广义优势估计（generalized advantage estimation，GAE）把连续 TD residual 按 $\gamma\lambda$ 加权 [7]。实现时定义第二个掩码：仅当缓冲区中存在同一回合、时间相邻的下一条 residual 时，$c_t = 1$；真终止、时间限制后的 reset，以及没有下一条 residual 的 rollout 末端均取 $c_t = 0$。后向递推是

$$\hat{A}_t = \delta_t + \gamma\lambda c_t \hat{A}_{t+1}, \qquad \hat{R}_t = \hat{A}_t + V_{\mathrm{old}}(s_t). \tag{3.4}$$

bootstrap 值来自该转移的 final observation，递推则只连接真正连续的数据。$c_t = 0$ 并不自动使 $b_t = 0$。

| 转移边界 | $b_t$ | $c_t$ | 原因 |
|---|---|---|---|
| 普通中间转移 | 1 | 1 | 未来有值，也有下一条 residual |
| 真终止 | 0 | 0 | 任务结束，随后数据不可接入 |
| 外部时间上限并 reset | 1 | 0 | 保留尾部价值，切断轨迹连接 |
| 普通 rollout 末端 | 1 | 0 | 在最后状态 bootstrap，当前数组不再递推 |

若跨 rollout 连续保存了下一段并联合计算，也可以连接末端；关键是下一条 residual 真实可用且仍属同一回合。重排 mini-batch 后不能再按数组下标计算 GAE，必须先在时间有序的 rollout 上完成它。

**Listing 3.1** 时间与环境两维数组上的双掩码 GAE

```
# values and next_values were saved with the old critic.
# next_values use final observations, never reset observations.
with no_grad():
  delta = rewards + gamma * bootstrap_mask * next_values - values
  advantages = zeros_like(values)
  carry = zeros(number_of_environments)
  for t in reversed(range(rollout_length)):
    carry = delta[t] + gamma * lam * continuation_mask[t] * carry
    advantages[t] = carry
  returns = advantages + values
```

### 3.3.1 一条短轨迹的手算

对四状态的完整绕行轨迹，取冻结估计 $V_{\text{old}}(s_0, s_1, s_2) = (2.5, 3.5, 2.8)$，$\gamma = 0.9, \lambda = 0.8$。逐项计算为

| $t$ | $r_t$ | $G_t$ | $\delta_t$ | $\hat{A}_t$ | $\hat{R}_t$ |
|---|---|---|---|---|---|
| 0 | 0 | 3.33 | 0.65 | 0.76808 | 3.26808 |
| 1 | 1 | 3.7 | 0.02 | 0.164 | 3.664 |
| 2 | 3 | 3 | 0.2 | 0.2 | 3 |

例如 $\hat{A}_1 = 0.02 + 0.72 \times 0.2 = 0.164$。若在 $s_1 \to s_2$ 后外部截断，只有前两条数据，正确做法是 $b_1 = 1, c_1 = 0$，于是 $\hat{A}_1 = 0.02$，$\hat{A}_0 = 0.6644$。若错误地把截断当作真终止，则 $\delta_1 = 1 - 3.5 = -2.5$，得到 $\hat{A}_0 = -1.15$，连 actor 更新方向也会反转。

$\lambda = 0$ 只用一步 residual；$\lambda = 1$ 在完整回合上望远镜相消，给出 $G_t - V_{\text{old}}(s_t)$。后者即使 baseline 不准确，完整 MC 策略梯度仍可在相应采样条件下无偏；但有截断尾部时仍依赖 bootstrap。若 critic 精确，各 $\lambda$ 都可以具有正确的条件优势期望；一般 critic 下，选择 $\lambda$ 是对多步随机性与尾部估计误差的折中，不能保证某个固定值在所有任务最优。

## 3.4 性能差异、TRPO 与 surrogate 的近似

旧数据上的梯度只在旧策略附近可信。性能差异引理精确给出两策略之间的联系：

$$J(\pi') - J(\pi) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\gamma^{\pi'}, a \sim \pi'} [A^\pi(s, a)]. \tag{3.5}$$

它可以通过沿新策略轨迹求和 $r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$ 并作望远镜相消来证明。实际却没有大量新策略数据，于是把 $d_\gamma^{\pi'}$ 替换为 $d_\gamma^\pi$，构造

$$L_\pi(\pi') = J(\pi) + \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_\gamma^\pi, a \sim \pi} \left[ \frac{\pi'(a \mid s)}{\pi(a \mid s)} A^\pi(s, a) \right].$$

比率修正动作分布，不能同时修正未来状态分布。这就是局部 surrogate 与真实性能之间的主要缺口。

TRPO 在这个 surrogate 上加入平均 KL 约束 [8]：

$$\begin{aligned}\max_{\theta} \quad & \mathbb{E}_{s\sim d_\gamma^{\pi_{\text{old}}},a\sim\pi_{\text{old}}}[\rho_\theta(s,a)\hat{A}(s,a)]\\ \text{使} \quad & \mathbb{E}_{s\sim d_\gamma^{\pi_{\text{old}}}}[D_{\text{KL}}(\pi_{\text{old}}(\cdot\mid s)\|\pi_\theta(\cdot\mid s))]\le\delta,\\ & \rho_\theta(s,a)=\pi_\theta(a\mid s)/\pi_{\text{old}}(a\mid s).\end{aligned}$$

局部二阶近似把 KL 写成参数步长的 Fisher 二次型；共轭梯度近似求解相应方向，再通过回溯检查经验 KL 和 surrogate。理论性能下界中的最坏状态策略差异与准确优势，和实现中的平均 KL、有限样本、近似 critic 并不相同。因此应理解其控制分布变化的思路，同时保留实际诊断，不能把近似算法描述为无条件单调改进。

## 3.5 PPO clipping 做了什么

PPO 用更容易进行多轮小批量优化的样本目标 [9]：

$$L^{\text{clip}}(\theta)=\mathbb{E}_t\left[\min\left(\rho_t\hat{A}_t,\text{clip}(\rho_t,1-\epsilon,1+\epsilon)\hat{A}_t\right)\right],\qquad \rho_t=\exp(\log\pi_\theta-\log\pi_{\text{old}}). \tag{3.6}$$

这里期望的时间权重应与所选采样目标一致；若要对应式 (3.2)，状态按旧策略的折扣 occupancy 加权。省略正的全局常数不改最优方向，省略随时间变化的权重则是另一项近似。

当 $\hat{A}_t>0$，增大 $\rho_t$ 超过 $1+\epsilon$ 不再增加这一个样本的目标；当 $\hat{A}_t<0$，减小到 $1-\epsilon$ 以下也不再增加目标。与此相反方向的变化仍会受到惩罚。它限制的是这个样本的优化激励，并没有把概率比投影回区间，更没有硬性限制所有状态上的 KL。

四状态参考策略的精确优势为 $\pm0.665$。取 $\epsilon=0.2$，新策略令绕行概率从 0.5 变为 $p$，两动作比率为 $2p$ 和 $2(1-p)$。当 $p\ge0.6$，在 $s_0$ 上对旧动作分布求平均的两个样本项都进入平台：

$$\frac{1}{2}(1.2\times0.665)+\frac{1}{2}(0.8\times(-0.665))=0.133.$$

在这个平台上，$p=0.6$ 和 $p=0.99$ 的 clipped 目标相同，而旧到新的 KL 随 $p\to1$ 发散。这直接说明 clipping 不是 KL 约束。神经网络还会让不同状态共享参数，某个样本进入平台后，其概率仍可能被其他样本推动。

### 3.5.1 一次更新应保存哪些量

完整 PPO 更新先冻结一版策略采样，按时间顺序计算优势和目标，再打散做小批量优化。示意如下：

**Listing 3.2** PPO 优化中的冻结量和时间权重

```
old_logp = rollout.logp.detach()
adv = advantages.detach()
value_target = returns.detach()
weight = rollout.discount_weight.detach()
# weight is gamma**t for the episodic discounted estimator.
# Equal rollout weights instead define the common sampled-state surrogate.
for epoch in range(number_of_epochs):
  for batch in minibatches(rollout):
    logp, entropy, value = model.evaluate(batch.obs, batch.action)
    ratio = exp(logp - old_logp[batch.index])
    a = adv[batch.index]
    gain = minimum(ratio * a, clamp(ratio, 1-eps, 1+eps) * a)
    w = weight[batch.index]
```

```
    policy_loss = -(w * gain).sum() / w.sum()
    value_loss = mean((value - value_target[batch.index])**2) / 2
    loss = policy_loss + value_coef * value_loss - entropy_coef * mean(entropy)
    optimizer.zero_grad()
    loss.backward()
    clip_gradient_norm(model.parameters(), max_norm)
    optimizer.step()
  if measured_kl_on_full_rollout(model, rollout) > kl_limit:
    break
```

这个代码块是接口层伪代码，不声称包含某个环境的全部训练配置。旧 log probability 在多轮 epoch 中保持不变；若每次更新后重算“旧概率”，比率就失去意义。循环网络还需按序列构造批次并维护初始隐状态，不能任意打散每一帧。

### 3.5.2 优势归一化与监控边界

批内标准化 $\tilde{A} = (\hat{A} - \operatorname{mean} \hat{A})/(\operatorname{std} \hat{A} + \varepsilon)$ 常能改善数值尺度，但不是有限样本上的恒等变换。同一批动作参与了均值估计，且中心化可能改变优势符号，进而改变 clipping 的激活区间。缩放也改变固定学习率下的步幅，以及策略、价值、熵三种损失的相对作用。应保存原始优势分布，明确采用全 rollout 还是每个 mini-batch 的统计量，并把这个选择纳入复现配置。

记录 KL、进入截断区间的比率比例、熵、价值误差与梯度范数。用旧动作样本估计 $\mathbb{E}[\log \pi_{\text{old}} - \log \pi_\theta]$ 时，有限样本结果甚至可能为负；若能计算完整条件动作分布的 KL，则含义更直接。KL early stop 只是经验控制，不能为未覆盖状态提供保证。增大 epoch 数也不只是“多训练一会儿”：旧数据与新策略的失配会逐渐增大。

## 3.6 熵奖励与连续动作的衔接

最大熵目标写为

$$J_{\text{ent}}(\pi) = \mathbb{E}_\pi \left[ \sum_t \gamma^t \big( r_t + \alpha \mathcal{H}(\pi(\cdot \mid s_t)) \big) \right], \quad \mathcal{H}(\pi(\cdot \mid s)) = -\mathbb{E}_{a \sim \pi} \log \pi(a \mid s).$$

它改变了优化目标；较高熵也不等于探索到了任务需要的稀有状态。PPO 中一个局部 entropy bonus 与完整的最大熵动态规划仍有区别。第 4 章的 SAC 从软价值目标同时推导 critic 和 actor，而非只在普通 Q 更新后额外加一项。

连续策略必须明确动作分布与执行动作的一致性。若先从无界高斯采样再硬裁剪，边界上的实际分布含有集中质量，不能把裁剪后的动作继续当作普通高斯密度。可逆的 tanh 变换能够保留可计算密度，但要加入 Jacobian；下一章给出稳定计算和动作单位变换。PPO 的新旧策略必须评价同一动作及同一种变换约定，否则 ratio 不是概率密度比。

**用小问题检查梯度和边界**

在四状态 MDP 的 $p = 1/2$ 处，用两条完整路径枚举计算带 $V^\pi$ baseline 的期望策略梯度，应仍为 0.3325。再将短轨迹中的 $\lambda$ 改为 1，核对 $\hat{A} = G - V_{\text{old}}$。最后只保留前两条转移，并分别用“任务确实在此结束”与“外部采样截断”解释两套目标；提示：同一组观测和奖励若缺边界语义，无法唯一确定正确 bootstrap。

# 第 4 章　深度价值与连续控制

表格方法为每个状态动作单独存一个数；神经网络让相近输入共享参数，也让一次更新同时改变许多未被采样的位置。它因此能处理图像和连续状态，却不再自动继承表格 Q-learning 的收敛结论。函数逼近、bootstrap 与离策略数据相互作用时，估计甚至可能发散。本章沿着同一个训练循环比较 DQN、Double DQN、DDPG、TD3 和 SAC：数据如何保存，下一动作由谁选择，目标由谁计算，以及梯度允许走到哪里。

默认环境是充分观测、动作边界固定的折扣任务。连续空间还需要适当的可测性、可积性与可微性条件；网络可训练不等于这些条件或全局最优性已经成立。部分可观测任务在章末另行讨论。

## 4.1　共同骨架：回放、延迟目标与梯度边界

### 4.1.1　一条转移应保留什么

经验回放池（replay buffer）至少保存

$$(s, a, r, s', d^{\text{term}}, d^{\text{trunc}}), \qquad b = 1 - d^{\text{term}}.$$

$a$ 是环境实际执行的动作，$s'$ 是该转移的后继观测。自动 reset 的向量环境可能直接返回下一回合初始观测；若发生外部截断，应从接口的最终观测字段取出原回合的 $s'$，不能拿 reset 后的观测 bootstrap。真正终止时目标为 $r$；任务定义外的时间上限通常仍接下一价值。若时间预算本身就是任务的一部分，应把剩余时间并入状态，到时按真终止处理。

回放将相邻样本打散，并允许一条经验被多次使用，但没有创造新的环境证据，也没有使全部样本独立同分布。池内通常混合多个旧策略的数据；其容量、抽样规则和数据年龄决定训练分布。旧经验可能扩大覆盖，也可能在环境变化后变得失效。每次环境转移对应多少次梯度更新称为更新比率，须连同 batch 大小、启动学习前的采样量一起报告；仅比较相同 episode 数并不足以控制预算。

### 4.1.2　目标网络为何有用，又没有保证什么

本章用 $Q_\phi$ 表示在线 critic，$Q_{\bar{\phi}}$ 表示目标 critic；确定性 actor 为 $\mu_\theta$。critic 最小化

$$L_Q(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\frac{1}{2}\big(Q_\phi(s,a) - y\big)^2\right], \qquad y = \operatorname{sg}\big(r + \gamma b\, v_{\text{next}}\big). \tag{4.1}$$

sg 表示整个目标不参与此次反向传播。它是 TD 的半梯度更新，不是对一个随 $\phi$ 改变的 Bellman residual 求完整梯度。DQN 也常用 Huber 损失抑制大残差的梯度；损失选择与奖励缩放都应写入配置。

初始化时先令目标网络等于在线网络，再采用硬复制或软更新。例如本书约定

$$\bar{\phi} \leftarrow (1-\tau)\bar{\phi} + \tau\phi, \qquad 0 < \tau \le 1.$$

这里 $\tau$ 越小，目标变化越慢；有些代码把保留旧参数的系数命名为 polyak，其数值含义相反。硬复制则每隔约定的学习步数令 $\bar{\phi} \leftarrow \phi$。目标更新在优化步骤之外执行，不通过计算图学习。网络若含运行均值等缓冲区，也应声明其同步方式。延迟目标缓解相互追逐，代价是滞后；它不构成非线性离策略学习的收敛证明。

## 4.2 离散动作：从 DQN 到 Double DQN

DQN 用网络输出每个离散动作的价值，行为通常采用 $\epsilon$-greedy，训练目标为 [10, 11]

$$y_{\text{DQN}} = \text{sg}\left[r + \gamma b \max_{a'} Q_{\bar{\phi}}(s', a')\right]. \tag{4.2}$$

同一 batch 中只回归实际执行动作的 $Q_\phi(s, a)$，不把全部动作输出都拉向同一个目标。行为探索、回放抽样和下一状态的最大化是三个不同环节；下一动作不必等于当时行为策略会采取的动作。

### 4.2.1 最大化为何可能放大估计噪声

若两个动作的真实值都为 5，估计误差分别独立地以相同概率取 $-1$ 或 1，则最大估计以 1/4 概率为 4、以 3/4 概率为 6，均值为 5.5。虽然单个估计无偏，取最大值仍产生正偏差。这个算例说明一种机制，不表示所有训练误差都无偏，也不表示实际 DQN 必然处处高估。

Double DQN 将选择与评估分开 [12]：

$$a^* = \arg\max_{a'} Q_\phi(s', a'), \qquad y_{\text{Double}} = \text{sg}\left[r + \gamma b Q_{\bar{\phi}}(s', a^*)\right]. \tag{4.3}$$

动作索引来自在线网络，数值来自目标网络，二者都不通过 $y$ 回传梯度。若评估噪声与选择完全独立且无偏，上述教学例子的评估均值可回到 5；实际两个网络有关联，因此 Double DQN 是减少一种偏差的构造，并非无偏性或总回报的普遍保证。

### 4.2.2 表示、抽样和回报分布分别改变什么

Dueling 网络用

$$Q(s, a) = V(s) + A(s, a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s, a')$$

共享状态信息，并通过中心化消除分解的平移不唯一性 [13]。它改变表示结构；其中的 $V, A$ 不会仅因命名就等于真实策略值与优势。

优先经验回放（PER）按 $p_i \propto (|\delta_i| + \varepsilon)^\kappa$ 抽样，使大残差经验更常出现 [14]。若池内有 $N$ 条样本，对固定目标的均匀池损失，无归一化权重 $w_i = 1/(Np_i)$ 能修正此抽样分布差异。常见实现用 $w_i = (Np_i)^{-\beta_{\text{IS}}}$ 并退火 $\beta_{\text{IS}}$；小于 1 时是部分修正，额外归一化还会改变步幅。权重通常不参与求导。它修正的是回放抽样，不能消除行为策略失配、数据外动作误差或任意多步目标的离策略偏差。

回报分布方法估计随机回报 $Z(s, a)$，而非只估计其均值：C51 用固定支撑上的概率表示多峰与尾部 [15]。若动作仍按 $\mathbb{E}[Z]$ 最大化，控制目标仍是期望回报；有了尾部表示并不会自动变成风险敏感算法，也没有自动获得探索保证。Rainbow 组合 Double、Dueling、PER、多步目标、分布表示和 NoisyNet[16, 17]。组合实验应控制交互量与更新量，并检查各部件的额外假设，不能把某个基准上的增益当成全部任务的配方。

## 4.3 连续动作：用 actor 代替显式最大化

连续动作无法逐项枚举 $\max_a Q(s, a)$。DDPG 学习确定性 actor，令它提供 critic 认为好的动作，同时用行为噪声收集经验 [18]：

$$y_{\text{DDPG}} = \text{sg}\left[r + \gamma b Q_{\bar{\phi}}\big(s', \mu_{\bar{\theta}}(s')\big)\right], \tag{4.4}$$

$$L_\mu(\theta) = -\mathbb{E}_{s\sim\mathcal{D}} Q_\phi\big(s, \mu_\theta(s)\big). \tag{4.5}$$

critic 用旧行为的实际动作回归；actor 用当前动作改善价值。这是回放状态分布上的代理目标，不能不加条件地称为初始分布下 $J(\pi)$ 的无偏梯度。其链式法则为

$$\nabla_\theta L_\mu = -\mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q_\phi(s,a)\big|_{a=\mu_\theta(s)}\nabla_\theta \mu_\theta(s)\right].$$

更新 actor 时冻结 critic 的参数，但保留 critic 对动作输入的导数。把整个 Q 输出 detach 会同时切断这条必要路径。更新 critic 时则相反：整个后继目标、目标 actor 和目标 critic 都不反传。DDPG 为 actor 与 critic 各维护目标副本。

采样时可执行 $a = \mathrm{clip}(\mu_\theta(s)+\epsilon, a_{\min}, a_{\max})$；噪声用于获得数据，不能因此把 critic 的训练动作误换成无噪声均值。也不能默认部署前撤掉探索就一定改善行为，应独立评估。DDPG 的单个 critic 可能被 actor 利用，特别是在回放很少覆盖的动作区域。

### 4.3.1 TD3 的三个改动各作用在哪里

TD3 同时学习两个 critic，并在后继动作上平滑目标 [19]：

$$\begin{aligned}\epsilon &= \mathrm{clip}(\xi, -c_\epsilon, c_\epsilon), \qquad \xi \sim \mathcal{N}(0, \sigma^2_{\mathrm{targ}} I),\\ \tilde{a}' &= \mathrm{clip}\big(\mu_{\bar{\theta}}(s') + \epsilon, a_{\min}, a_{\max}\big),\\ y_{\mathrm{TD3}} &= \mathrm{sg}\left[r + \gamma b \min_{j=1,2} Q_{\bar{\phi}_j}(s', \tilde{a}')\right].\end{aligned} \tag{4.6}$$

两位 critic 都回归同一个 $y$。取较小值抑制正误差被反复选中的效应，也可能带来低估；相关的双网络不是置信区间，最小值不是经校准的安全下界。目标平滑减少 actor 对狭窄价值尖峰的利用；它只发生在目标动作上，与真实采样时的行为噪声应分开配置。

每做 $d$ 次 critic 更新再更新一次 actor，使用 $-\mathbb{E}Q_{\phi_1}(s,\mu_\theta(s))$；典型 TD3 实现也在这个延迟时刻更新 actor 与 critic 的目标副本。延迟让 actor 面对相对充分更新的 critic，并不保证 critic 已准确。噪声标准差和裁剪阈值要明确是在归一化动作坐标还是环境单位下指定。

## 4.4 SAC：让随机策略与软价值使用同一个目标

SAC 将熵纳入目标。以下软 Bellman 关系用于非终端决策状态；真终止后的软价值固定为零，不再累计策略熵。定义软动作价值时不含当前动作的熵，把当前熵放入软状态价值：

$$V^\pi_{\mathrm{soft}}(s) = \mathbb{E}_{a\sim\pi}[Q^\pi_{\mathrm{soft}}(s,a) - \alpha\log\pi(a\mid s)], \qquad Q^\pi_{\mathrm{soft}}(s,a) = \mathbb{E}[r + \gamma b V^\pi_{\mathrm{soft}}(s')].$$

这样与上一章的奖励加熵目标相接，也解释了下一状态为何减去 log density。温度 $\alpha > 0$ 决定奖励与熵的相对尺度。原始 SAC 包含独立的 V 网络；以下采用后续论文中常用的双 Q、无独立 V 网络与自动温度版本 [20, 21]：

$$y_{\mathrm{SAC}} = \mathrm{sg}\left[r + \gamma b\left(\min_j Q_{\bar{\phi}_j}(s',a') - \alpha\log\pi_\theta(a'\mid s')\right)\right], \quad a'\sim\pi_\theta(\cdot\mid s'). \tag{4.7}$$

两位 critic 拟合该目标。这里后继动作由当前随机 actor 采样，通常不再维护一个目标 actor；整个右侧仍在无梯度环境下计算。真终止会去掉后继价值与后继熵，不能仅给 Q 项加掩码而留下熵项。

### 4.4.1 随机 actor 仍可沿动作路径求导

若 $a = f_\theta(s,\xi)$，其中 $\xi$ 的分布不依赖 $\theta$，重参数化 actor 损失为

$$L_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D},\xi}\left[\alpha\log\pi_\theta(f_\theta(s,\xi)\mid s) - \min_j Q_{\phi_j}(s, f_\theta(s,\xi))\right]. \tag{4.8}$$

critic 参数和温度在这一步固定，动作及其 log probability 保持梯度。最小值的激活分支提供动作导数；不可将其整体 detach。相反，critic 目标和温度目标中的 log probability 应 detach。actor 采样使用能保留路径梯度的接口；仅调用一个返回无梯度样本的采样函数，会丢掉 Q 对动作的学习信号。

### 4.4.2 熵温度的符号与梯度尺度

令 $h_\theta = \mathbb{E}[-\log \pi_\theta(a \mid s)]$，希望在所用状态分布上满足平均熵下界 $h_\theta \geq h_*$。固定策略时，对偶温度目标可最小化 $\alpha(h_\theta - h_*)$。令 $\ell = \log \alpha$，两种常见写法为

$$\begin{aligned} L_\alpha(\ell) &= -\mathbb{E}\left[e^\ell \,\mathrm{sg}(\log \pi_\theta(a \mid s) + h_*)\right], & \partial_\ell L_\alpha &= \alpha(h_\theta - h_*), \\ L_\ell^{\text{proxy}}(\ell) &= -\mathbb{E}\left[\ell \,\mathrm{sg}(\log \pi_\theta(a \mid s) + h_*)\right], & \partial_\ell L_\ell^{\text{proxy}} &= h_\theta - h_*. \end{aligned} \tag{4.9}$$

第一式是将正温度参数化后得到的对偶损失；第二式是常用代理损失，省掉了梯度中的 $\alpha$ 因子。它们有相同零点与方向，却有不同的优化尺度，学习率不能不加说明地通用。熵低于目标时梯度为负，梯度下降提高 $\ell$ 和 $\alpha$，从而增强 actor 的熵激励。这是检查符号的直接方法。

连续变量使用微分熵，可以为负。常见的 $h_* = -n_a$ 是归一化动作坐标下的启发式，$n_a$ 是动作维度，不是连续动作集合的大小。目标可能与动作变换或任务不匹配；自动调温不保证探索覆盖或训练稳定。奖励整体缩放也会改变合适的温度尺度，须同时检查奖励单位、实际熵与 $\alpha$ 曲线。

## 4.5 动作变换：执行的动作必须对应所算的密度

设各维边界有限且 $a_{\max,i} > a_{\min,i}$，记

$$m_i = \frac{a_{\max,i} + a_{\min,i}}{2}, \qquad c_i = \frac{a_{\max,i} - a_{\min,i}}{2} > 0.$$

令 $\xi_i$ 独立服从 $\mathcal{N}(0,1)$，对角高斯策略可取 $u_i = \mu_{\theta,i}(s) + \sigma_{\theta,i}(s)\xi_i$、$z_i = \tanh u_i$，执行 $a_i = m_i + c_i z_i$。由变量变换公式，环境动作坐标下的联合 log density 是

$$\log \pi_a(a \mid s) = \sum_{i=1}^{n_a} \left[\log \mathcal{N}(u_i; \mu_{\theta,i}, \sigma_{\theta,i}^2) - \log c_i - \log(1 - \tanh^2 u_i)\right]. \tag{4.10}$$

log probability 要沿动作维求和，留下每个 batch 样本一个标量。标准差可由有界的 log standard deviation 指数化，以避免数值溢出；实际边界必须记录。不要从已饱和为 $\pm 1$ 的浮点动作反求 arctanh；采样时保留 $u$ 更稳妥。Jacobian 可用恒等式

$$\log(1 - \tanh^2 u) = 2\big(\log 2 - u - \mathrm{softplus}(-2u)\big)$$

稳定计算。直接在对数中加 $\varepsilon$ 是数值近似，过大的 $\varepsilon$ 会改变边界附近的密度。

以一维 $u = 0, \mu = 0, \sigma = 1, c = 2$ 为教学检查，$a = m$，应有 $\log \pi_a = -\frac{1}{2}\log(2\pi) - \log 2 \approx -1.6121$。漏掉缩放项会得到 $-0.9189$。常数 $-\log c_i$ 不改变固定温度下 actor 的直接密度梯度，却会改变熵数值和自动温度的目标；若改用归一化坐标，应一致采用 $\log \pi_z$，并满足 $h_{*,a} = h_{*,z} + \sum_i \log c_i$ 的换算，而非混用两种单位。熵坐标改变还相当于每步增加一个常数；若策略能影响回合长度，该常数可能改变结束时机的激励，不能把两套约定直接视为相同任务。

**硬裁剪与可逆变换不是同一种分布**

先从无界高斯采样再硬裁剪，会在动作边界形成概率质量，不能继续用普通高斯密度描述执行结果。SAC 的熵项因此会算错；PPO 的新旧概率比也可能失真。DDPG/TD3 的行为动作可以含裁剪噪声，因为相应 critic 只使用已执行动作，并不据此计算高斯策略熵。若接口在 tanh 缩放之外再次显著裁剪动作，仍须检查实际执行分布。

## 4.6 把五种算法放进同一个训练循环

下表中的后继目标都代入式 (4.1) 并整体 stop-gradient。算法名背后的差别主要位于目标动作、价值选择与 actor 更新，而非是否使用相同的优化器。

| 算法 | 后继动作与价值 | actor | 目标副本与节奏 |
|---|---|---|---|
| DQN | 目标 Q 上取最大值 | 无独立 actor | 一个目标 Q，定期复制 |
| Double DQN | 在线 Q 选，目标 Q 评 | 无独立 actor | 一个目标 Q，定期复制 |
| DDPG | 目标 actor，单目标 Q | 最大化在线 Q | actor/Q 均软更新 |
| TD3 | 平滑目标动作，双 Q 取小 | 最大化第一位 Q，延迟更新 | actor/双 Q 随延迟节奏更新 |
| SAC | 当前随机动作，双 Q 取小并减熵项 | 重参数化的软目标 | 目标双 Q；通常无目标 actor |

**Listing 4.1** 回放型算法的共同接口层伪代码

```
initialize_online_networks()
copy_online_to_targets()
obs = env.reset()
for env_step in range(interaction_budget):
  action = behavior_action(obs)  # random warmup or exploration policy
  next_obs, reward, terminated, truncated, info = env.step(action)
  final_obs = transition_successor(next_obs, info)  # before any reset
  replay.add(obs, action, reward, final_obs, terminated, truncated)
  obs = reset_or_continue(next_obs, terminated, truncated, info)
  if not ready_to_learn(replay, env_step):
    continue
  for _ in range(updates_per_environment_step):
    batch = replay.sample(batch_size)
    with no_grad():
      b = 1.0 - batch.terminated
      # DQN, Double, DDPG, TD3, or SAC formula for next value
      y = masked_bellman_target(batch, b, algorithm)
    optimize_critics(batch.obs, batch.action, y)
    if has_actor(algorithm) and actor_update_due(algorithm):
      freeze_critic_parameters()  # retain derivative wrt action
      optimize_actor(batch.obs, detached_temperature())
      unfreeze_critic_parameters()
      if algorithm == "SAC":
        optimize_log_temperature(detached_logp(batch.obs))
    update_targets_when_due(algorithm)
    log_update_counts_and_diagnostics()
```

伪代码省略了优化器、优先级刷新、评估与 checkpoint 等接口配置，不是已执行的深度训练结

果。目标计算最好只对 $b = 1$ 且后继有效的样本查询网络，再回填终止样本目标；用 $0 \times \text{NaN}$ 并不能得到零。若用多步 replay，还须定义跨终止的累积奖励、$\gamma^n$ 和离策略修正，不能只把一步目标中的 $r$ 换成任意长回报。

## 4.7 诊断、部分可观测性与扩展边界

先记录原始回报与训练奖励，再同时观察 Q 值分位数、TD residual、actor/critic 梯度、目标滞后、回放年龄和更新比率。Q 值变大可能来自奖励尺度，也可能来自 bootstrap 错误；TD 损失很小只说明拟合了当前目标。对 DDPG/TD3 记录行为噪声与 actor 动作的饱和比例，对 SAC 另报 log density、熵目标和温度；评估随机策略与固定动作评估都应明确命名，不能择优混报。观测归一化统计量在评估时冻结。

若 $o_t$ 不是马尔可夫状态，循环网络可以输入历史，但逐帧打散 replay 会破坏其含义。应按序列采样，保存回合边界，用 burn-in 重建隐状态，并声明是否对 burn-in 反向传播；真终止、reset 与外部截断仍须分开。增加记忆容量并不自动恢复被遗漏的信息。

分布式 actor 与 learner 还会产生策略滞后。A3C 用异步工作者采集并更新；IMPALA 用 actor–learner 分离和 V-trace 控制部分离策略效应 [22, 23]。这不是把任意 replay 接到 PPO 上就能得到的修正。吞吐收益应连同队列滞后、环境步数、学习步数、设备和总墙钟报告，不能用训练器每秒样本数替代端到端成本。

**训练大网络之前的四个可核验问题**

用合成输入令后继 Q 极大：真终止目标必须仍等于 $r$，外部截断则应保留 bootstrap。固定 critic $Q(s,a) = -(a-0.3)^2$ 与 actor $\mu_\theta = \theta$，在 $\theta = 0.1$ 处最小化 actor 损失的导数应为 $-0.4$；若为零，检查动作路径是否被 detach。再核对式 (4.10) 的一维密度与数值 Jacobian，并用熵低于/高于目标的两组数检查温度更新方向。以上都是实现边界检查；它们通过后，仍需第 12 章的多种子评估来判断实际学习行为。

# 第 5 章 获得探索信号、时间抽象与示范

前几章假设回放或轨迹已经包含足够有用的经验。然而，若到达目标需要先连续做对几十个动作，准确的 TD 目标也不能凭空创造一次成功。探索解决“收集什么数据”，信用分配解决“如何从数据判断动作的作用”，模仿与偏好则引入环境奖励之外的证据。这些问题会相互影响，却不能用同一个技巧代替。

本章先讨论探索信号，再用潜势塑形说明“更容易学”与“仍然是原任务”的区别，最后把时间抽象和专家信息接入同一个学习循环。默认 $0 \le \gamma < 1$；有限回合中，$r_t$ 始终是从 $s_t$ 执行 $a_t$ 到 $s_{t+1}$ 的奖励。

## 5.1 从随机动作到有目的地收集经验

一个 $\epsilon$-greedy 策略以概率 $\epsilon$ 随机选动作，其余时间选择当前最优动作。它在表格任务中便于分析，但逐步独立的随机选择很难穿过长走廊：若每一步都必须选中两个动作中的一个，连续 $L$ 步成功的概率可以低至 $2^{-L}$。增加每步噪声不等于增加整段行为的多样性。按回合采样参数噪声、保留随机价值函数或选择一个持续执行的目标，都可能产生更连贯的探索。

探索方案首先要回答两个问题。第一，不确定的是环境规律，还是环境本来就随机？前者称为认知不确定性，可通过数据减少；后者称为偶然不确定性，即使模型已知也会存在。第二，新增信号是否改变了最终优化目标？例如训练时加上新颖性奖励，会改变当时的策略偏好。只有在最后仍按外在回报评测，并明确奖金是否退火，才能判断它帮助了任务还是仅帮助了得分。

有限状态空间中，可以用访问计数 $N_t(s,a)$ 构造奖金

$$b_t(s,a) = \frac{c}{\sqrt{N_t(s,a)+1}}, \qquad \tilde{r}_t = r_t^{\text{ext}} + b_t(s_t,a_t).$$

它表达“少见的状态动作值得尝试”，但这里的公式只是教学性启发式；要获得遗憾界，还需要相应置信集、访问规则和环境假设。高维观测中几乎每张图都不同，计数必须依赖离散化或表示。同一房间的光照变化若被当成新状态，奖金就会被背景消耗。

ICM 在学习的特征空间中预测下一状态，利用预测误差鼓励探索；RND 则固定随机目标网络 $f$，训练预测网络 $g_\theta$，以

$$r_t^{\text{int}} = \|f(o_{t+1}) - g_\theta(o_{t+1})\|_2^2 \tag{5.1}$$

作为新颖性信号 [24, 25]。RND 的目标是固定的，不需要预测随机环境的下一帧，但观测噪声、表示分辨率和预测器容量仍会影响奖金。任何把预测误差当探索价值的方法，都应检查“噪声电视”式失败：智能体不断观看不可预测变化，却不接近目标。

**判断探索是否有效**

分别绘制外在回报、内在奖励和任务相关覆盖率。后者可用已到达房间、完成子目标或可达区域中的状态计数衡量，定义必须预先固定。若内在奖励上升而任务覆盖不变，先检查观测噪声与特征；若覆盖扩大而成功率不变，再检查目标可达性和信用分配。评测阶段关闭奖金时还要检查策略是否依赖奖金输入。

## 5.2 奖励塑形：何时保持原问题不变

稀疏奖励下，人们自然会奖励“离目标更近”。问题是，局部距离可能与最终任务不一致：反复接近目标再退后，也可能获得高分。潜势塑形提供了一个可验证的构造 [26]。给定状态函数 $\Phi$，定义

$$F(s_t, s_{t+1}) = \gamma\Phi(s_{t+1}) - \Phi(s_t), \qquad \tilde{r}_t = r_t + F(s_t, s_{t+1}). \tag{5.2}$$

对长度 $T$ 的轨迹，额外回报满足望远镜求和

$$\sum_{t=0}^{T-1} \gamma^t F(s_t, s_{t+1}) = -\Phi(s_0) + \gamma^T \Phi(s_T). \tag{5.3}$$

无限折扣任务若 $\Phi$ 有界，最后一项趋于零。同一初始状态下，所有策略的价值都减去同一常数 $\Phi(s_0)$，故动作排序不变。固定初始分布下，目标也只差 $-\mathbb{E}_{s_0}[\Phi(s_0)]$。这正是结论成立的原因，不是“密集奖励通常不影响最优策略”。

有限回合在终止时截断求和，必须处理式 (5.3) 的残余项。把所有真正终端状态设为 $\Phi = 0$ 是常用做法。若允许时间依赖潜势，应使用 $\gamma\Phi_{t+1}(s_{t+1}) - \Phi_t(s_t)$，并把剩余时间作为状态的一部分；不能在有真实截止时间的任务里继续沿用无限时域推导。若时间上限只是数据收集的截断，则还需按原继续任务 bootstrap，而非把截断观测擅自设成零势能终端。

### 5.2.1 在贯穿 MDP 上检查终端残余

使用前文的四状态例子：$s_0$ 的 direct 获得 2 后终止，via 获得 0 到达 $s_1$，随后依次获得 1、3 后终止，$\gamma = 0.9$。两条路径的原回报分别为 2 和 3.33。取

$$\Phi(s_0) = 0, \quad \Phi(s_1) = \Phi(s_2) = 1, \quad \Phi(s_\perp) = 0.$$

via 路径的塑形后逐步奖励是 $0.9, 0.9, 2$，折扣回报仍为

$$0.9 + 0.9 \times 0.9 + 0.9^2 \times 2 = 3.33.$$

direct 的回报仍为 2。反过来，若把终端势能错误地设成 10，并在终止后直接停止累计，短路径多得 $0.9 \times 10 = 9$，长路径多得 $0.9^3 \times 10 = 7.29$；两者变成 11 和 10.62，最优选择被反转。即使所有终端的势能相等，变长回合中的非零终端势能也不能随意保留。

## 5.3 HER：重用失败经验需要哪些条件

目标条件策略写作 $\pi(a \mid s, g)$。HER 的想法是：一次未完成目标 $g$ 的轨迹，可能已经完成另一个目标 $g'$；将轨迹中的实际到达状态选为 $g'$，重算奖励，再用 off-policy 算法学习 [27]。机器人没有把物体推到指定位置，却推到了另一位置，这仍是一条关于“如何到那里”的经验。

重标记可复用同一转移的基本条件是，目标只改变奖励或成功判据，不改变物理转移：

$$P(s' \mid s, a, g) = P(s' \mid s, a).$$

此外，必须保存足够信息以计算 $r(s, a, s', g')$，并按新目标重新检查成功与终止语义。旧目标已触发环境重置后，不能伪造一段新目标下从未执行的后续轨迹。如果某个目标会打开门、改变工具可用性或修改动作动力学，直接替换 $g$ 已不再是同一个转移样本。

即使动力学与目标无关，随机环境中也要小心事后选择。若把随机抛掷的实际落点当目标，重标记样本看起来次次成功，但给定这个目标重新执行时未必成功；按结果选样改变了条件转移分布。

原始 HER 在常见近确定性目标任务中很实用，却不是任意随机 MDP 的无偏重标记定理。实现中应保留原目标样本，报告重标记比例，并用固定的原始目标分布测成功率。

信用分配与探索也应分开诊断。更长的 $n$-step target 或 eligibility trace 可以更快传播已经出现的奖励，却不能保证第一次成功发生。HER 增加可学习的成功标签，但不自动把所有目标都变得可达；目标课程需要同时关注能力边界与最终测试分布。

## 5.4 Options：把决策的时间尺度写进 Bellman 目标

一个 option $o = (\mathcal{I}_o, \pi_o, \beta_o)$ 包含可启动状态集合、内部动作策略和终止概率。高层在 $s_t \in \mathcal{I}_o$ 时选择 $o$，内部策略执行 $\tau \geq 1$ 步后返回控制权。它将原始 MDP 转成半马尔可夫决策过程（SMDP），因为下一次决策间隔是随机的 [28]。

令 option 执行期间的折扣奖励为

$$R_t^o = \sum_{k=0}^{\tau-1} \gamma^k r_{t+k}.$$

高层 Q-learning 的一个样本目标是

$$y_t^o = R_t^o + \gamma^\tau b_t \max_{o' \in \mathcal{O}(s_{t+\tau})} Q(s_{t+\tau}, o'), \tag{5.4}$$

其中 $b_t = 0$ 表示任务真正终止，正常的 option 终止则 $b_t = 1$。若只用 $\gamma$ 而不是 $\gamma^\tau$，长 option 的未来价值会被系统性高估。若记录的是未折扣奖励和，也不能直接放进此式。

高层并非在每个原始步都重新选 option；若允许中途打断，打断规则本身就是另一种策略，需重新说明数据如何生成。低层不断更新时，同一个 option 的转移规律也会变化，使高层回放陈旧。零步终止、所有 option 永不结束、子目标不可达，都是比网络结构更先需要检查的边界。

Option-Critic 联合学习内部策略和终止函数 [29]，但梯度可学不代表学到的抽象有用。一个简单实验应记录每个 option 的持续时间、启动分布、终止位置和使用比例，再检查它是否跨起点或任务复用。若所有 option 都执行同样行为，层次结构只是增加了参数；若只奖励完成手工子目标，还应确认它没有改变整体任务的最优解。

## 5.5 专家示范：先学习动作，再补足部署分布

行为克隆（BC）直接用专家数据 $\mathcal{D}_E$ 拟合

$$L_{\mathrm{BC}}(\theta) = -\mathbb{E}_{(s,a)\sim\mathcal{D}_E} \log \pi_\theta(a \mid s).$$

连续动作也可在明确噪声模型后做回归。它的优势是训练稳定、无需反复在线试错；困难在于训练状态来自专家，而执行状态来自学习策略。一次小偏差可能进入专家从未访问的状态，随后误差进一步扩大。离线验证上的动作准确率因此不足以证明闭环控制成功。

DAgger 把“部署会访问哪里”带回标注循环 [30]：

1. 用现有聚合数据训练策略 $\pi_i$；
2. 在环境中执行 $\pi_i$ 或声明好的专家混合策略；
3. 对实际访问的状态请求专家动作，把新标签加入聚合数据；
4. 重训并在独立任务上闭环评测。

专家给出的是这些状态的正确动作，不只是接管后最终走过的轨迹。理论中的无遗憾在线学习、可查询专家和损失条件，都不是现实部署的免费资源。专家不能观察某些状态、接管会改变状态分布、

查询危险状态有代价时，应把这些限制写进协议。DAgger 也不是仅凭已有固定示范即可完成的离线算法。

示范是否次优决定了下一步。在纯 BC 后加入环境奖励，策略可能超越示范，但也可能忘掉罕见的恢复行为。应保留 BC 基线，并区分“更多在线交互”与“更好的示范利用”带来的收益。报告专家成功率、标注次数、交互次数和失败示范比例，比只写示范条数更有解释力。

## 5.6 从匹配行为到推断奖励

GAIL 不逐动作复制，也不要求恢复唯一奖励，而是匹配专家与策略的状态动作占用 [31]。用本书归一化折扣联合占用 $\mu_\gamma^\pi(s,a)$，令判别器 $D$ 输出“来自当前策略”的概率，可写为

$$\min_\pi \max_D \mathbb{E}_{\mu_\gamma^\pi} \log D(s,a) + \mathbb{E}_{\mu_\gamma^{\pi_E}} \log(1 - D(s,a)) - \lambda \mathcal{H}_\gamma(\pi), \tag{5.5}$$
$$\mathcal{H}_\gamma(\pi) = \mathbb{E}_{(s,a)\sim\mu_\gamma^\pi}[-\log \pi(a \mid s)].$$

这个约定下，策略可以使用 $-\log D(s,a)$ 作为奖励；若实现将 $D$ 定义为专家概率，符号必须对应改变。判别器学到的是当前可区分的特征，不是外部安全证明。专家与策略轨迹长度、终止状态处理或观测预处理不同，判别器可能仅靠这些捷径分类。

逆强化学习（IRL）进一步询问“什么奖励能解释专家行为”。最大熵方法以解释数据为约束，避免对未约束行为过度确定 [32]。在确定性动力学、固定有限时域、已知可行轨迹集合的教学设定下，可用

$$p_\omega(\tau) = \frac{q_0(\tau)\exp\left(\sum_t r_\omega(s_t,a_t)\right)}{Z(\omega)}$$

表达相对轨迹基测度 $q_0$ 的指数族模型。随机动力学下，环境随机性必须保持外生，不能让归一化隐式选择有利的随机结果；最大因果熵表述对策略的因果条件分布建模，不能仅把所有轨迹当可自由控制的序列。

奖励有不可辨识性：正比例变换和适当的潜势变换可保留最优行为，示范覆盖之外的奖励也缺少约束。IRL 恢复的因而是一个与行为及建模假设相容的奖励，而非唯一的真实意图。若研究目标是迁移，应在动力学或初始状态改变后再检查恢复奖励；只在原示范分布上得到高分，无法区分奖励恢复和行为记忆。

## 5.7 偏好反馈与方法选择

当逐动作标签难以提供，标注者可以比较两个片段 $\sigma^A, \sigma^B$。常用成对模型为

$$P_\omega(\sigma^A \succ \sigma^B) = \frac{\exp R_\omega(\sigma^A)}{\exp R_\omega(\sigma^A) + \exp R_\omega(\sigma^B)}, \qquad R_\omega(\sigma) = \sum_{(s,a)\in\sigma} r_\omega(s,a).$$

用偏好交叉熵训练奖励，再用 RL 优化它，形成闭环 [33]。片段长度不同、奖励是否折扣、平局和分歧如何编码，都会改变模型含义。仅凭片段比较，长程成功、安全底线和未展示的副作用可能不可识别。主动选择模型分歧大的片段可以提高查询效率，但仍要为稀有失败、代表性样本和独立审查留出预算。

| 已有条件 | 优先建立的基线 | 先验证的边界 |
|---|---|---|
| 奖励可得、交互便宜 | 简单随机探索与原算法 | 奖励是否可达，训练与评测奖励是否一致 |
| 目标奖励可重算 | 目标条件学习与 HER | 目标是否改变动力学，终止与随机选择偏差 |
| 可靠专家轨迹 | BC，再决定是否在线改进 | 闭环偏移、专家质量和查询成本 |
| 只有片段偏好 | 奖励学习与保留测试偏好 | 标注一致性、覆盖和奖励过优化 |
| 任务含重复长行为 | 先验证固定 options，再学习抽象 | 持续时间折扣、可达性及 option 退化 |

这些路线可以组合，但组合之前应能解释每一项增加了什么证据、消耗了多少真实交互。下一章讨论另一类证据：从已有转移学习模型，并用它进行规划或产生训练目标。第 7 章则将收紧条件，完全禁止训练中的新交互。

**练习与提示**

把四状态 MDP 的 $\Phi(s_0)$ 改成 2，其他潜势保持本章的正确设置，分别计算两条路径的塑形回报，并解释动作排序为何不变。再构造一个包含随机落点的目标任务，比较原目标样本和“实际落点作目标”的样本中成功事件的频率。第二题要求解释条件选择，不要求运行深度网络。

# 第 6 章 模型、规划与误差累积

如果一条真实转移很昂贵，能否从它学习环境规律，再在模型中练习许多次？模型型强化学习的吸引力来自这种复用，也来自显式比较未来方案的能力。但多生成一百万条模型转移并没有多获得一百万次环境证据；如果策略利用了模型错误，训练分数越高反而可能偏离真实任务。

本章区分模型的三种用途：提供价值更新的模拟数据，在执行时搜索动作，以及在潜空间内训练策略。它们可以组合，却对模型精度、计算预算和部署接口提出不同要求。

## 6.1 Dyna：把真实经验与规划接起来

在完全可观测 MDP 中，可以学习转移 $\hat{P}_\psi(s' \mid s,a)$ 和条件期望奖励 $\hat{r}_\psi(s,a)$。若奖励依赖下一状态，应学习相应联合分布或条件函数；若任务会终止，还要建模真正的终止事件。Dyna 将直接学习与模型规划放在同一个循环中 [34]：

1. 在真实环境执行动作，保存 $(s_t, a_t, r_t, s_{t+1}, \text{terminated})$；
2. 用真实转移更新价值函数，并更新模型；
3. 从已有状态动作中选择规划起点，向模型查询模拟后继；
4. 用模拟转移执行若干次价值更新，然后继续真实交互。

表格确定性环境可直接记忆转移；随机环境则需要估计转移分布。规划起点的采样也重要：均匀采样、按近期访问频率采样和优先更新价值变化大的前驱，会把同样的计算花在不同位置。

一条模拟 TD 目标可以写为

$$\hat{y} = \hat{r}(s,a) + \gamma \hat{b} \max_{a'} Q_{\bar{\theta}}(\hat{s}', a'), \qquad \hat{s}' \sim \hat{P}_\psi(\cdot \mid s,a).$$

这里的 $\hat{b}$ 表示模型预测的继续事件。训练时必须区分真实环境的终止、模型的终止预测和人为设定的模拟 rollout 长度。短 rollout 到头通常是截断，需要尾值；把每个模拟末端都当作终局，会悄悄修改学习问题。

模型不是越全面越好。控制任务可能只需要准确预测奖励、接触、可达性和关键动力学，而不需要还原背景纹理。反过来，若模型遗漏了制动延迟，即使整体状态均方误差很低，也可能在真正重要的动作上出错。

## 6.2 为什么一步准确仍会长期失真

考虑确定性动力学 $s_{k+1} = f(s_k, a_k)$ 与近似模型 $\widehat{f}$，暂时给定同一动作序列。若有关区域内 $f$ 对状态是 $L$-Lipschitz，且一步模型误差至多 $\epsilon$，则

$$\Delta_{k+1} \le L\Delta_k + \epsilon, \qquad \Delta_0 = 0.$$

故当 $L \neq 1$ 时，$\Delta_k \le \epsilon(L^k - 1)/(L-1)$，$L = 1$ 时为 $k\epsilon$。闭环策略还会因不同状态选出不同动作，所需稳定性条件更强。该推导不是神经网络模型的自动保证；它展示了误差为什么可能随视野增加而放大。

另一种视角是价值误差。对有限状态动作空间、同一固定策略、$|r|, |\hat{r}| \le R_{\max}$，假设全部状态动作上满足

$$|r(s,a) - \hat{r}(s,a)| \le \epsilon_r, \qquad \sum_{s'} |P(s' \mid s,a) - \hat{P}(s' \mid s,a)| \le \epsilon_P.$$

把两个 Bellman 方程相减并取最大范数，可得

$$\|V^\pi - \hat{V}^\pi\|_\infty \le \frac{\epsilon_r}{1-\gamma} + \frac{\gamma R_{\max}\epsilon_P}{(1-\gamma)^2}. \tag{6.1}$$

这里的 $\epsilon_P$ 是 $L^1$ 距离；若改用总变差距离，常数需要对应调整。奖励界与一致误差界都不可省略。训练集平均预测误差小，不能推出此式的前提；最优策略又会主动搜索模型认为高价值的区域，更容易碰到训练数据外的错误。

**模型验证不能只随机拆转移**

同一轨迹相邻帧高度相关，随机把转移分到训练和测试会低估泛化误差。至少按轨迹拆分，并另外保留策略更新后访问的数据。分别测一步与多步预测、奖励、终止、关键事件和候选动作排序；再检查“模型判优、真实变差”的样本。集成分歧低也可能是所有模型共享了同一个偏差。

## 6.3 PETS：在每个真实时刻重新规划

PETS 用概率动力学集成表示模型不确定性，并用采样轨迹比较动作序列 [35]。每个模型可输出下一状态差分的均值与方差：模型内方差表达近似的环境噪声，模型之间的分歧表达近似的认知不确定性。两者的含义依赖拟合和校准，不能把“方差大”直接等同于某个已证明的置信区间。

模型预测控制（MPC）在当前状态求解有限视野问题，例如

$$a^*_{t:t+H-1} \in \arg\max_{a_{t:t+H-1}} \mathbb{E}_{\hat{P}}\left[\sum_{k=0}^{H-1}\gamma^k \hat{r}_{t+k} + \gamma^H \hat{V}(s_{t+H})\right], \tag{6.2}$$

并只执行第一个动作，获得新的真实观测后重新规划。尾值可以省去，此时相当于将 $\hat{V}$ 设为零，但有限视野的偏好会改变。交叉熵方法等采样优化器可以反复保留高分候选，更新动作序列分布；其计算不是免费资源，必须计入每步延迟。

重新规划使预测错误有机会被真实观测纠正，但不消除模型偏差。若一次动作就会不可逆地越过危险边界，下次重规划已经来不及。对模型预测施加约束，也只有在误差和执行偏差有适当界时才能转化成真实约束。

PETS 式路线适合真实样本昂贵、状态较低维、短视野内动力学可学且允许在线规划的任务。如果执行器每毫秒都需要动作，大量候选 rollout 可能无法满足时限。此时可考虑离线蒸馏控制器，或直接在模型里训练快速策略，但这是另一个计算与泛化问题。

## 6.4 MBPO：模型用于训练，执行时依赖策略

MBPO 从真实回放中的状态启动较短模型 rollout，将合成数据交给 off-policy 策略优化 [36]。它与 MPC 的关键区别是：前者用模型改进参数，部署时可只运行 actor；后者通常在每个执行时刻搜索动作。

短 rollout 的动机不是“模型在前几步一定正确”，而是控制偏离真实数据分布的程度。模拟步数为 $k$ 时，大致存在三种相互竞争的影响：真实经验复用更多、模型偏差积累更多、尾值依赖仍然存在。模型太差时，$k = 0$ 的无模型基线完全可能更好；模型改善后，较长 rollout 才可能合算。

一个可复现的实现应保存两个来源标签明确的缓冲区，至少说明：模型训练只用了哪些真实数据，rollout 起点如何采样，长度如何调整，真实与合成数据的混合比例，以及 actor/critic 每个真实步更新多少次。在同样真实交互预算下增加十倍梯度更新，可能提高样本效率而降低墙钟效率，两者都应报告。

离线模型学习更困难：没有新交互来纠正策略主动找到的模型漏洞。MOPO 等方法用模型不确定性惩罚合成奖励，MOReL 用保守的已知区域构造规划问题 [37, 38]。这些方法将假设转移到“不确定性是否识别未知区域”，并没有消除第 7 章的数据支持问题。

## 6.5 潜变量模型：从观测历史到想象训练

图像观测常是部分可观测的。单帧编码 $z = f(o)$ 未必保留速度、遮挡物和历史动作；世界模型因此常结合循环确定性状态 $h_t$ 与随机潜变量 $z_t$。一种教学性的结构为

$$h_t = f_\psi(h_{t-1}, z_{t-1}, a_{t-1}), \qquad q_\psi(z_t \mid h_t, o_t), \qquad p_\psi(z_t \mid h_t).$$

后验 $q$ 在真实观测到来后校正状态；先验 $p$ 支持没有未来观测的想象。模型通过观测、奖励、继续事件的预测损失以及先验后验之间的正则学习。不同论文的变量、梯度停止与权重并不相同，不能把这一简图当作可直接复现的完整实现。

World Models 展示了学习紧凑动态表示并在其中控制的路线；Dreamer 进一步在潜空间想象轨迹上训练 actor 和 critic[39, 40]。对给定的潜状态序列，一段 $H$ 步回报目标应是奖励和一次尾值，例如

$$\widehat{G}_0^{(H)} = \sum_{k=0}^{H-1} \gamma^k m_k \hat{r}_k + \gamma^H m_H V(\hat{z}_H), \qquad m_0 = 1, \quad m_{k+1} = m_k \hat{c}_k, \tag{6.3}$$

其中 $\hat{c}_k$ 是该转移继续的预测概率。不能在每步奖励中直接再加一个完整未来价值，否则会重复计算。用 $\lambda$-return 时，相应递推为

$$\widehat{G}_k^\lambda = \hat{r}_k + \gamma \hat{c}_k \big[(1-\lambda) V(\hat{z}_{k+1}) + \lambda \widehat{G}_{k+1}^\lambda\big], \qquad \widehat{G}_H^\lambda = V(\hat{z}_H).$$

继续概率与折扣的角色要区分；实现若已经把 $\gamma \hat{c}_k$ 合成 discount，就不应再次乘 $\gamma$。

DreamerV3 强调一组跨任务复用的建模与优化选择，包括离散潜变量、尺度处理和稳健的训练目标 [41]。其跨域报告不能被解释为在任意新领域无须检查模型或超参数。对本书读者，更有用的复现顺序是先确认真实观测后验、纯先验 rollout 和行为策略的输入一致，再检查奖励尺度与继续预测，最后考虑复杂正则。

潜空间的诊断也应面向控制。将背景换掉后重建变差但控制不变，未必是坏事；漏掉小障碍却重建大部分像素，可能很危险。可从同一真实历史启动多条想象，比较未来奖励、成功与失败事件、动作排序和真实执行回报。策略在模型里找到高分行为后，必须回到环境验证，而不能只增加想象时长。

## 6.6 MuZero：为搜索学习任务相关模型

AlphaZero 将已知游戏规则、策略价值网络和树搜索结合，通过自博弈生成改进策略 [42]。MuZero 的不同点是学习表示、潜状态动态以及奖励/策略/价值预测，不要求重建原始观测 [43]：

$$z_t^0 = h_\theta(o_{\le t}), \qquad (\hat{r}_t^k, z_t^{k+1}) = g_\theta(z_t^k, a_{t+k}), \qquad (p_t^k, v_t^k) = f_\theta(z_t^k).$$

实际历史中的动作把模型展开若干步，奖励目标来自真实奖励，价值目标来自适当的多步回报，策略目标来自搜索访问分布。搜索因而同时是动作选择工具和训练目标生成器。

这种模型追求的是决策相关预测一致性，不是像素层面的完整世界模拟。它不意味着环境规则、合法动作、终止处理或搜索工程都可以忽略。原方法的任务、观测接口与计算预算有明确条件；随机环境中如何表示随机转移，也需要额外设计，不能把单一路径潜动态直接当成完整随机生成器。

还应区分搜索改善与策略学习改善。如果评测时给算法更多搜索次数，性能提升可能来自额外在线计算。公平实验至少同时给出每步搜索预算、网络前向次数、动作延迟以及无搜索或固定搜索预算下的结果。自博弈任务还涉及对手分布，不能只报告对当前版本的胜率；第 8 章会继续讨论。

## 6.7 按样本预算与执行接口选路线

| 路线 | 模型的主要用途 | 执行时开销 | 首先检查 |
|---|---|---|---|
| Dyna | 产生价值更新样本 | 可只查价值/策略 | 规划起点与模型偏差 |
| PETS/MPC | 比较候选动作序列 | 每步规划 | 预测视野与动作时限 |
| MBPO | 产生短合成 rollout | 可只运行 actor | 真实分布起点、混合比 |
| Dreamer | 潜空间训练行为 | 编码、状态更新与 actor | 后验/先验差异及终止 |
| MuZero | 支持搜索并生成目标 | 搜索与网络计算 | 决策预测、搜索预算 |

上述不是性能排行榜。低维连续控制、视觉探索、棋类和语言工具环境有不同的动作接口与数据结构，不能把各自论文的归一化分数横向比较。先固定真实交互、模型训练、策略更新与部署计算预算，再比较有模型和无模型基线，才能知道模型究竟节约了什么。

**一个不需要训练大模型的验证**

在四状态 MDP 中把模型的 finish 奖励从 3 错写成 2，手算模型认为的 via 回报，并比较它与真实值。再逐步增大误差，找到模型会选择 direct 的阈值。这个例子说明：重要的不只是预测误差大小，还包括动作价值间隔。式 (6.1) 是最坏情况界，不会告诉我们每个问题实际何时选错动作。

# 第 7 章 固定数据学习与反事实评估

Off-policy 描述学习策略与收集策略不同，并不意味着算法在完全不能补数据时仍然可靠。在线 Q-learning 选错动作后，环境至少可能提供纠正；离线强化学习只有固定日志，策略却仍会倾向于选择估值最高的动作。估值误差与动作选择相互放大，成为与监督学习不同的关键难点。

本章先划清从数据中能识别什么，再说明 CQL、IQL 如何约束改进，最后讨论部署前的离线策略评估（off-policy evaluation，OPE）。这些方法都不能从不存在的证据恢复任意反事实。

## 7.1 固定数据、行为策略与支持

设数据 $\mathcal{D}$ 由行为策略 $\mu$ 收集，包含转移、轨迹边界、终止类型，最好还包含动作概率和收集策略版本。目标是用数据学习策略 $\pi$，训练期间不新增环境交互 [44]。若日志来自多种策略，“行为策略”可以是依赖历史与版本的混合机制，不必是单一平稳网络。

在同一个 MDP 和初始分布中，必要的支持条件通常写为

$$\mu_\gamma^\pi(s,a) > 0 \implies \mu_\gamma^\mu(s,a) > 0.$$

这里上标 $\mu$ 指行为策略，下标 $\gamma$ 的 $\mu_\gamma$ 仍是联合占用测度。这仅保证理论上有覆盖。有限样本还关心密度比多大：若目标策略把大量概率放在行为策略极少访问的区域，数据量可能远不足以估计其价值。假设 $\mu_\gamma^\pi/\mu_\gamma^\mu \le C$ 的集中系数界，正是在量化这种困难；实际连续高维问题中，$C$ 往往无法精确知道。

### 7.1.1 一个不可识别例子

设初始状态只有动作 $a$ 和 $b$，执行一次后终止。日志只记录 $a$ 的奖励为 1。两个环境都与全部日志相容：一个环境里 $b$ 的奖励为 100，另一个为 $-100$。任何只读这些日志的算法都无法区分二者。因此，若新策略总选 $b$，其真实价值无法由数据唯一确定。一个神经网络给出自信的 $Q(s,b)$，并没有增加关于环境的证据。

这也解释离线 Bellman 备份的危险：即使只在日志里的 $(s,a)$ 上计算平方损失，目标

$$r + \gamma \max_{a'} Q(s',a')$$

仍会查询日志外动作。数据内 loss 下降可以与策略表现变差同时发生。目标策略还会访问新的状态，形成更长的分布偏移链。

**第一份报告应该描述数据**

按轨迹统计来源策略、成功与失败比例、回报分布、长度、截断、动作范围和缺失字段。明确奖励是否重标记、是否存在重复轨迹、日志是否保留真实时间顺序。只给“若干百万转移”无法说明数据质量，更无法说明目标策略的覆盖。

## 7.2 先建立行为克隆与行为约束基线

行为克隆只学习日志动作分布，不做远离数据的 Bellman 最大化。当数据大多来自高质量行为、动作标签充分、闭环偏移可控时，它往往是强基线。若日志包含次优与优秀轨迹混合，BC 可能平均掉重要选择；若所有轨迹都失败，也不能指望简单复制获得成功。

BCQ 先生成接近数据的候选动作，再做受限价值选择；BEAR 用分布距离控制策略偏离 [45, 46]。这些方法的共同动机是减少没有数据支撑的动作查询，而不是证明行为策略处处最优。约束太紧会

阻止有证据的改进，约束太松则重新暴露外推误差。动作维度高时，判断两个分布是否接近本身也是估计问题。

模型型离线方法将风险转移到模型上。MOPO 在不确定区域扣减奖励，MOReL 对已知区域之外采取保守建模 [37, 38]。惩罚有效的前提是模型不确定性确实反映了策略将要遇到的错误，而非仅反映观测噪声。

## 7.3 CQL：对数据外估值施加压力

离散动作下，CQL 的一种常用形式为 [47]

$$\begin{aligned} L_Q(\phi) = \frac{1}{2}\mathbb{E}_{\mathcal{D}}\left[Q_\phi(s,a) - \mathrm{sg}(y)\right]^2 \\ + \alpha \mathbb{E}_{s\sim\mathcal{D}}\left[\log \sum_{a'} \exp Q_\phi(s,a') - \mathbb{E}_{a\sim\hat{\mu}(\cdot|s)} Q_\phi(s,a)\right], \\ y = r + \gamma b\, \mathbb{E}_{a'\sim\pi(\cdot|s')} Q_{\bar{\phi}}(s',a'). \end{aligned} \tag{7.1}$$

其中 $b$ 是真正终止的 bootstrap 掩码，sg 表示停止梯度；$\hat{\mu}$ 可由数据动作的经验采样实现。熵正则版本的备份还需加入相应策略熵项，此处为解释保守项而省去它。

对一个状态，$\log \sum_a \exp Q(s,a)$ 的梯度是 softmax 权重，倾向于压低当前高值动作；减去数据动作的平均 Q 则相对抬高数据内动作。因此它不是把全部 Q 乘上一个小于一的系数，也不是只惩罚低频动作的手工表。$\alpha$ 过大时仍可能将有价值的动作差异抹平，过小时又不能控制外推。

论文中的保守价值结论有采样、函数类与优化等条件，不能把任意有限 batch 的神经网络训练结果称为真实价值的逐状态下界。连续动作时，求和改为相对于指定测度的积分，通常需要提议分布和重要性校正；直接从少数 actor 动作做 log-sum-exp 并不是精确的全动作积分。奖励单位、动作维度与提议分布也会影响正则尺度，故不能跨任务机械复用 $\alpha$。

## 7.4 IQL：在数据动作内估值，再提取策略

IQL 避免在 critic 备份中显式查询新策略动作 [48]。它学习三个对象：数据动作的 Q、状态价值 V 和由优势加权的 actor。令 $\tau_e \in (0,1)$ 为 expectile 参数，定义非对称平方损失

$$\ell_{\tau_e}(u) = |\tau_e - \mathbf{1}\{u<0\}|u^2.$$

三个训练目标可写为

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \ell_{\tau_e}\big(\mathrm{sg}(Q_{\bar{\phi}}(s,a)) - V_\psi(s)\big), \tag{7.2}$$

$$L_Q(\phi) = \mathbb{E}_{\mathcal{D}}\left[Q_\phi(s,a) - \mathrm{sg}\big(r + \gamma b V_\psi(s')\big)\right]^2, \tag{7.3}$$

$$L_\pi(\theta) = -\mathbb{E}_{(s,a)\sim\mathcal{D}}\left[w(s,a)\log \pi_\theta(a\mid s)\right], \tag{7.4}$$

$$w(s,a) = \min\big\{w_{\max}, \exp\big[\beta\ \mathrm{sg}(Q_{\bar{\phi}}(s,a) - V_\psi(s))\big]\big\}.$$

这里 $\beta$ 为逆温度，越大越强调高优势样本；也有实现用温度 $T = 1/\beta$ 表达，不能混用调参方向。目标 Q 可用双 Q 的较小值，具体延迟更新与权重截断需记录。

当 $\tau_e = 0.5$，expectile 是条件均值；$\tau_e > 0.5$ 更重视数据中高 Q 动作，逐渐接近数据支持内的较好选择。它不是对随机回报左尾风险的估计，也不是把环境随机奖励取最大值。IQL 特意先对动作价值做 expectile，再用普通期望回归 Q，以免把动力学随机性误当可控选择。

优势加权并未让 actor 在严格意义上拥有数据支持保证：连续密度模型可能在动作之间插值，也可能在未覆盖状态输出任意动作。若 $\tau_e$ 接近一、$\beta$ 很大，少数高估样本会支配拟合。应监控权重分布、有效加权样本量、BC loss 和动作分布变化，而不只看 Q loss。

**一个局部的 IQL 手算**

同一状态的数据动作有等频的两个 Q 值 0 和 2。令 $0 < V < 2$，最小化 $\frac{1}{2}[(1-\tau_e)V^2 + \tau_e(2-V)^2]$ 得 $V = 2\tau_e$。当 $\tau_e = 0.75$ 时 $V = 1.5$。这只是固定 Q 下的动作分布 expectile，不是最终策略价值的独立证明。

## 7.5 序列建模改变了训练形式，没有消除覆盖

Decision Transformer 将目标回报、状态和动作组成因果序列，按监督学习方式预测动作；Trajectory Transformer 则对轨迹变量建模，并可通过搜索生成候选行为 [49, 50]。它们绕开显式 Q 最大化，但仍依赖日志提供可学的行为及结果关系。

原始 Decision Transformer 常用未折扣的 return-to-go

$$R_t^{\text{cond}} = \sum_{k=t}^{T-1} r_k$$

作为条件变量。它与本书默认折扣回报不是同一个量；复现时应沿用并明确相应定义。推理中根据已获奖励更新目标条件，不等于模型可以实现任意大目标。超出数据范围的目标回报可能引发分布外行为。更长上下文能记住历史，却不会自行纠正目标不可达、隐藏混杂或成功轨迹缺失。

若数据能由片段拼接形成更好策略，Bellman 方法和序列条件方法利用这种结构的方式不同。比较时应检查具体数据是否包含可拼接状态、历史是否影响转移，以及测试目标是否落在训练条件范围，而非笼统声称某种架构能自动解决长程信用分配。

## 7.6 OPE：用日志估计未执行策略的价值

以下先用固定有限时域 $H$ 说明。真正终止后补零奖励并保持吸收状态，目标为

$$J_H(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{H-1} \gamma^t r_t\right].$$

若初始分布也变化，还需额外处理初始状态密度比。仅重加权动作不能纠正不同的任务人群。若日志未记录影响动作与结果的隐变量，常见的无混杂与 Markov 假设也可能不成立。

### 7.6.1 重要性采样：正确概率与长程方差

已知行为动作概率时，定义

$$\rho_{0:t}^{(i)} = \prod_{k=0}^{t} \frac{\pi(a_k^{(i)} \mid s_k^{(i)})}{\mu(a_k^{(i)} \mid s_k^{(i)})}.$$

若策略依赖历史，分子分母都必须改成相应历史条件概率。逐决策重要性采样（PDIS）估计为

$$\hat{J}_{\text{PDIS}} = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{H-1} \gamma^t \rho_{0:t}^{(i)} r_t^{(i)}. \tag{7.5}$$

在支持成立、行为概率正确、轨迹独立且期望存在时，这个有限时域估计无偏。长乘积仍可能造成巨大方差。连续确定性目标策略与连续行为密度之间常不满足所需绝对连续关系，不能随意把动作距离代入概率比。

轨迹级普通 IS 使用完整 $\rho_{0:H-1}$ 乘整段回报。自归一化版本再除以权重之和，通常牺牲有限样本无偏性以改善稳定性；截断权重也引入偏差。行为概率由数据拟合时，额外估计误差会影响这些性质。对权重 $w_i$ 可报告

$$\mathrm{ESS} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2},$$

但 ESS 高不证明覆盖充分：若从未观测到的关键失败区域权重根本无法计算，统计量不会主动报警。

### 7.6.2 FQE：固定策略的 Bellman 回归

Fitted Q evaluation 不改进目标策略，反复用固定 $\pi$ 拟合 [51]

$$y_j = r_j + \gamma b_j \mathbb{E}_{a' \sim \pi(\cdot|s'_j)} Q_{k-1}(s'_j, a'), \tag{7.6}$$

$$Q_k \approx \arg\min_{Q \in \mathcal{F}} \sum_{j \in \mathcal{D}} (Q(s_j, a_j) - y_j)^2. \tag{7.7}$$

最后在目标初始分布上平均 $\mathbb{E}_{a\sim\pi} Q_k(s_0, a)$。有限时域也可按时间反向拟合 $Q_t$。FQE 不需要显式行为概率，却需要数据覆盖目标访问区域、函数类能表达相关 Bellman 备份，以及回归与迭代误差可控。它仍然会在目标动作或新状态上外推；更低的验证 TD loss 不是价值准确性的充分证据。

### 7.6.3 Doubly robust：模型残差与权重结合

令 $\hat{q}_t$ 为固定的有限时域动作价值估计，$\hat{v}_t(s) = \mathbb{E}_{a\sim\pi}\hat{q}_t(s, a)$，并令 $\hat{v}_H = 0$。一种序贯 doubly robust（DR）表达是 [52]

$$\hat{J}_{\mathrm{DR}} = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{v}_0(s_0^{(i)}) + \sum_{t=0}^{H-1}\gamma^t \rho_{0:t}^{(i)}\left(r_t^{(i)} + \gamma b_t^{(i)}\hat{v}_{t+1}(s_{t+1}^{(i)}) - \hat{q}_t(s_t^{(i)}, a_t^{(i)})\right)\right]. \tag{7.8}$$

其含义是先使用模型预测，再用加权 TD 残差校正。对固定、在独立数据上得到的价值模型，正确重要性比可校正模型误差；在相应条件下，正确价值模型也可抵消行为模型误差。所谓“双重稳健”不能理解为两个模型都差仍可靠，也不能免除支持和有限矩条件。若在同一轨迹上拟合并评价，残差相关性会破坏直接套用的性质，应使用样本拆分或交叉拟合。

## 7.7 置信区间与选择偏差比点估计更重要

OPE 的置信区间需要明确统计单位。独立收集的轨迹可以整条重采样；同一连续日志的强时间依赖、共享用户或对手变化，需要相应分块或分层处理。不能把同一轨迹的几千个转移当成几千次独立策略试验。bootstrap 对重尾权重、小样本稀有事件和完全缺失支持并无自动修复作用。

如果反复用同一评估集选择超参数和候选策略，最高估值会带有选择偏差。保留只在最终使用的测试集，报告候选数量和选择规则；若要声称高置信度优于行为策略，还需考虑同时比较与置信下界的构造。不同估计器一致也可能源自共享数据缺口，不应以简单多数投票代替覆盖分析。

**可执行的离线评估协议**

按轨迹划分训练、验证和最终测试，冻结数据与奖励版本。训练 BC 和一个改进方法，用验证集选超参数；在独立日志上运行 FQE，行为概率可用时补 IS/DR。报告回报估计、区间、权重、支持与估计器差异。在线验证须预先限定预算，并从纯离线结果中单列。

离线学习应列出结论所依赖的日志、目标策略与覆盖假设，再据此决定补哪些数据、是否保守部署，或承认证据不足。

# 第 8 章 多智能体：信息、协调与共同学习

多个决策者共处一个环境，并不只是把单智能体代码并行运行。其他人的动作影响转移，其他人的学习又影响未来的数据分布；局部观测、团队奖励和通信限制决定了哪些联合策略实际上能够执行。算法名称之前，必须先说明游戏和信息结构。

## 8.1 从联合 MDP 到部分可观测博弈

一个 $n$ 智能体 Markov game 包含状态 $s$、联合动作 $\mathbf{a} = (a_1, \dots, a_n)$、转移 $P(s' \mid s, \mathbf{a})$ 以及各自奖励 $r_i(s, \mathbf{a})$。完全合作时奖励可共享，零和双人游戏满足 $r_1 = -r_2$，一般和游戏则既可能合作也可能冲突。优化团队回报与寻找博弈均衡是不同问题。

局部观测环境中，智能体 $i$ 的可用历史可写为

$$h_t^i = (o_0^i, a_0^i, o_1^i, \dots, a_{t-1}^i, o_t^i), \qquad a_t^i \sim \pi_i(\cdot \mid h_t^i).$$

这里 $h_t^i$ 是信息历史，不是上一章的潜状态符号。若执行时没有通信或公共随机信号，常见联合策略按局部策略乘积分解。若允许通信、公共计划或相关随机性，则策略类更大，必须说明消息何时到达、带宽及代价。部署给一个智能体读取所有局部观测，已经改变了问题。

即使每个智能体都做 Q-learning，局部环境也未必平稳。固定其他策略时，可以将其边缘化为

$$P_i(s' \mid s, a_i) = \sum_{\mathbf{a}_{-i}} P(s' \mid s, a_i, \mathbf{a}_{-i}) \pi_{-i}(\mathbf{a}_{-i} \mid \mathbf{h}_{-i}).$$

其他策略更新后，这个有效转移也改变了；部分可观测时，$s$ 的简化表达还需相应信念或历史条件。旧回放混合不同对手版本，不能直接套用平稳单智能体表格收敛结论。

## 8.2 CTDE：训练特权信息不能泄漏到执行

集中训练、分散执行（CTDE）允许训练时 critic 访问全局状态、其他动作或联合历史，actor 执行时仍只使用声明的局部信息。它利用集中数据降低估计难度，并没有取消执行约束。MADDPG 用集中动作价值评估各自 actor；COMA 则针对合作场景设计反事实 baseline[53, 54]。

以 COMA 为例，共享团队 Q 时，智能体 $i$ 的优势可写为

$$A_i(s, \mathbf{a}) = Q(s, \mathbf{a}) - \sum_{a_i'} \pi_i(a_i' \mid h_i) Q(s, (a_i', \mathbf{a}_{-i})). \tag{8.1}$$

它固定其他人的已选动作，平均掉自己的动作，得到与当前 $a_i$ 无关的 baseline，因而可用于减少策略梯度方差。它是在当前 critic 和条件下的贡献比较，不等于已经识别出长期社会因果效应，也不自动处理公平分配。

执行信息边界应逐项检查。actor 的输入、循环隐状态、动作合法性掩码、观测归一化和通信模块，都可能携带训练时才存在的信息。例如，局部视野外的敌人若只通过全局动作掩码透露位置，即使网络输入没有全局状态，仍发生信息泄漏。训练时用共享统计量并不必然违规，但部署是否可获得及更新这些统计量必须一致。

**一次有效的信息边界测试**

冻结模型后，让每个 actor 在只提供自身输入的独立进程中执行；全局状态只进入评价器与训练 critic。检查断开通信、增加延迟和缺失队友观测时的行为。若必须增加额外信息才能复现结果，

应把它作为新的执行协议，而不是隐含实现细节。

## 8.3 价值分解及 QMIX 的表达边界

合作任务中，联合动作空间随智能体数指数增长。VDN 将联合价值近似为局部值之和，QMIX 用全局状态条件的单调混合函数 [55]：

$$Q_{\text{tot}}(s,\mathbf{a}) = f_\psi(s, Q_1(h_1,a_1),\dots,Q_n(h_n,a_n)), \qquad \frac{\partial f_\psi}{\partial Q_i} \geq 0. \tag{8.2}$$

固定状态和历史，且可行联合动作集合是各局部集合的笛卡尔积时，逐个选择 $\arg\max_{a_i} Q_i$，就能最大化这个学得的混合价值。训练时混合器读取全局状态不影响局部 actor 的贪心执行。

这是一种一致性结构，不是关于真实最优值的普适表达定理。考虑只有一步的两人合作矩阵：

| | 智能体 2：A | 智能体 2：B |
|---|---|---|
| 智能体 1：A | 0 | 1 |
| 智能体 1：B | 1 | 0 |

当第二人选 A，第一人应更偏好 B；当第二人选 B，第一人应更偏好 A。固定局部历史时，$Q_1(A)$ 与 $Q_1(B)$ 只有一个排序，单调混合不能精确表达完整的联合值表。它仍可能学会其中一个最优角色分配，例如第一人 A、第二人 B；反例说明的是完整非单调价值结构不能被精确表示，并非断言任何最优行为都绝无可能学到。

此外，若联合动作受共享资源约束，例如两个智能体不能同时占同一槽位，各自局部最优动作的组合可能不可行。单调性不负责解决这个耦合约束。通信、集中分配器或另一种策略表示可能必要，但也会改变执行开销与假设。

## 8.4 MAPPO 与算法选择的实际依据

MAPPO 将 PPO 的局部策略与集中价值估计用于合作任务 [56]。它的吸引力是实现路径清晰，离散与连续 actor 都可沿用；代价是依赖新轨迹，且多个策略同步改变时，样本重复利用和大更新更需谨慎。PPO 的 clipped surrogate 在这里同样不是硬 KL 或联合性能单调保证。

参数共享也不是无条件收益。同类智能体共享网络能复用数据，角色不同或动作空间不同则需要额外建模。若输入智能体身份以打破对称，应在未见数量和身份上测试；若不用身份，完全相同观测下的确定性共享策略可能无法自发形成不同角色。循环状态有助于部分可观测，却要求每个环境、每个智能体的终止与重置掩码正确，不能将某人的记忆传给另一个人。

| 路线 | 适合先尝试的条件 | 主要诊断 |
|---|---|---|
| 独立 Q/Actor–Critic | 耦合较弱、快速基线 | 对手变化、回放版本与协调失败 |
| MADDPG | 连续动作、可用集中 critic | critic 维度、局部 actor 信息和对手覆盖 |
| COMA | 离散合作、团队信用困难 | 反事实 Q 精度与计算规模 |
| QMIX | 合作离散、局部贪心可执行 | 单调结构偏差、联合约束 |
| MAPPO | 有新交互预算、合作环境 | 更新幅度、循环掩码与样本效率 |

该表只描述选择依据。不同论文中的地图、人数、观测范围和训练预算不同，分数不构成跨协

议排名。先给出独立学习或简单协调基线，再解释为何集中 critic 或价值分解增加了收益，比一次堆叠所有稳定化模块更容易诊断。

## 8.5 自博弈：胜率取决于对谁比赛

只对当前版本自博弈可能出现循环。石头剪刀布中，一代打败上一代并不代表逐渐接近稳健策略；它可能只是在三种纯策略之间轮转。历史策略池、混合对手采样与种群训练能扩大对手覆盖，但也引入对手选择和维护成本。

双人零和场景中，若第一人最大化收益 $J(\pi,\nu)$、第二人最小化，一个有用的差距为

$$\mathrm{Gap}(\pi,\nu)=\max_{\pi'}J(\pi',\nu)-\min_{\nu'}J(\pi,\nu').$$

小型已知游戏可精确求最好回应；大型环境通常只能训练近似回应。近似攻击未找到漏洞，只说明这次搜索未找到，不能据此给出真实可利用性的上界。一般和任务还可能有多个均衡和不同社会福利，不能照搬单个零和差距解释全部表现。

合作任务的对应问题是对队友过拟合。固定团队的高回报可能来自隐含约定，与未见队友配合时崩溃。应做跨训练种子的交叉配对、固定队友池和未见策略测试，区分团队共同适应与可复用协作能力。平均团队回报还可能掩盖某个成员承担全部风险，需要补充每个角色的收益、失败和资源使用。

## 8.6 一份能比较的多智能体评测协议

评测至少固定环境版本、地图/任务集、人数与角色、观测、通信、合法动作信息、终止规则和对手/队友采样。交互预算要说明是环境联合步数还是所有智能体动作数：$n$ 个智能体执行一百万联合步，通常包含 $n$ 百万个个体动作记录。墙钟还受并行环境和集中网络开销影响。

统计单位通常是独立训练种子与独立游戏，而不是把同一局中的 $n$ 个智能体当成 $n$ 次独立实验。胜率应说明平局处理，团队任务报告成功率与回报分布；竞争任务报告对手矩阵和固定对手集上的结果，合作任务报告未见队友与人数变化。学习算法、执行信息或通信预算任一项不同，都应在表中显式分开。

**练习与解题方向**

先在本章的二维矩阵中假设 $Q_1(A)>Q_1(B)$，证明单调混合无法同时满足两列的真实排序；再讨论身份输入能否让共享 actor 选择某个最优角色分配。前一问讨论价值函数表达，后一问讨论策略表示，二者不能混成同一个“不可能”结论。

# 第 9 章 约束、尾部风险与鲁棒决策

“平均得分高”没有回答训练时能否承受失败、最差一批回合会发生什么，或者环境变化后是否仍然可用。本章将这些要求分成三个数学问题：成本约束控制期望预算，风险度量关心回报分布，鲁棒优化关心环境模型的不确定性。三者可以结合，但任何一个优化目标都不等于整套部署系统的安全保证。

## 9.1 成本约束首先要定义单位和时域

约束 MDP（CMDP）在奖励之外定义成本 $c_i$，例如能耗、碰撞指示或资源使用。使用本书的未归一化折扣目标：

$$\max_{\pi} J_R(\pi), \qquad J_{C_i}(\pi) = \mathbb{E}_{\pi}\left[\sum_{t\geq 0} \gamma^t c_{i,t}\right] \leq d_i. \tag{9.1}$$

如果改用归一化联合占用表示，$J_{C_i} = \mathbb{E}_{\mu_\gamma^\pi}[c_i]/(1-\gamma)$；阈值也必须按同样比例换算。有限回合总能耗、长期平均功率和折扣能耗是不同约束，不能共用数值相同的 $d_i$。

更重要的是，期望成本上限不等于“每步都安全”。高成本灾难若足够稀少，仍可能满足较小的平均预算；折扣还会降低晚期事故的权重。若需求是一次回合发生碰撞的概率不超过 $\delta$，应定义并估计相应事件概率；若需求是每个时刻状态属于安全集合，就需要逐步约束和额外可控性假设。数学目标应从真实容错需求推导，而非把容易优化的指标命名为安全。

## 9.2 Lagrangian：可以调节违约代价，不能凭空保证可行

引入非负乘子，构造

$$\mathcal{L}(\pi, \lambda) = J_R(\pi) - \sum_i \lambda_i (J_{C_i}(\pi) - d_i), \qquad \lambda_i \geq 0.$$

理想对偶问题是 $\min_{\lambda\geq 0} \max_{\pi} \mathcal{L}(\pi, \lambda)$。实践中交替提高策略的惩罚后回报，并更新

$$\lambda_i \leftarrow \left[\lambda_i + \eta_\lambda (\hat{J}_{C_i} - d_i)\right]_+. \tag{9.2}$$

成本超标时乘子增大，奖励单位不变而成本惩罚更强。这个方向常被写反；如果违反约束反而降低惩罚，应首先检查符号。

有限 MDP 的占用测度表述在适当可行性条件下有清楚的凸优化解释。然而，神经策略的非凸参数化、有限 batch 成本估计、近似 critic 和交替更新，都可能产生振荡或不可行结果。若任务本身没有可行策略，再大的乘子也不能制造可行性；它可能只会让学习崩溃。成本信号稀少时，乘子长期接收零观测，随后一次事故又引起剧烈更新，应报告这种动态。

CPO 以局部近似同时处理回报改进、成本约束与信赖域 [57]。其分析中的模型、优势和距离控制条件不等于实际深度实现处处成立。样本估计的成本约束满足，也不保证真实期望满足，更不保证每条训练轨迹无事故。比较算法时应同时绘制回报、成本、乘子和累计违约，不能只在训练结束挑一个可行 checkpoint。

**训练安全与最终策略安全要分开报告**

一个最终可行策略可能经历了大量训练事故；另一个方法最终回报较低，却在整个学习过程中遵守了限制。若真实交互有代价，应报告训练累计成本、最严重事件、接管次数和可行策略出现时间。环境重置、人工筛选或丢弃失败轨迹都属于协议的一部分。

## 9.3 左尾 CVaR：最差一部分回合有多差

令随机回报 $G = \sum_t \gamma^t r_t$，较大更好。期望目标可能接受低概率灾难。奖励的左尾条件风险价值可用无原子歧义的形式定义 [58]：

$$\mathrm{CVaR}_\alpha^{\mathrm{lower}}(G) = \sup_{\eta\in\mathbb{R}} \left\{ \eta - \frac{1}{\alpha}\mathbb{E}[(\eta - G)_+] \right\}, \qquad 0 < \alpha \le 1. \tag{9.3}$$

它衡量最差 $\alpha$ 比例回报的平均水平；$\alpha$ 越小，关注的尾部越极端。连续分布的适当条件下可写成分位点以下的条件均值，但分位点有概率质量时必须只取所需的一部分质量。成本越大越坏时，应转为右尾定义；把奖励左尾和成本右尾混用会优化相反方向。

### 9.3.1 一个含原子的手算例子

策略 A 恒得 0.2；策略 B 以 0.99 概率得 2，以 0.01 概率得 $-100$。期望分别为 0.2 和 0.98，期望最大化偏好 B。对 $\alpha = 0.05$，B 的最差 5% 包含全部 1% 灾难和 4% 正常结果，故

$$\mathrm{CVaR}_{0.05}^{\mathrm{lower}}(G_B) = \frac{0.01(-100) + 0.04(2)}{0.05} = -18.4,$$

而 A 的 CVaR 仍为 0.2。如果错误使用 $\mathbb{E}[G \mid G \le q_{0.05}]$，由于 $q_{0.05} = 2$，会把整个分布都纳入，得到 0.98，完全漏掉尾部定义。

其他风险度量也有不同偏好。方差惩罚会同时惩罚意外高回报与灾难，不专门关注左尾；指数效用依赖尺度和风险参数；分布式 critic 能提供回报分布信息，却不会在没有风险目标时自动变成风险敏感控制。

尾部优化还带来时间一致性问题：在起点优化整段回报的静态 CVaR，未必与每个后继状态重新优化 CVaR 相容。若要求动态风险度量，应明确嵌套结构或增广状态，不能直接把普通 Bellman 最优算子中的期望替换成一次 CVaR 就宣称解决了原静态目标。

## 9.4 稀有事件的证据需要足够样本

经验 CVaR 只依赖最差的一小部分样本，估计波动往往比均值大。$n$ 次独立回合中，最差 $\alpha$ 尾部平均只有约 $n\alpha$ 个样本；若事故概率更低，实际事故次数更少。几条成功视频无法证明低事故率。

一个简单的二项例子很有用。若 $n$ 次独立同分布测试未观察到任何事故，事故概率的单侧 $1-\delta$ 精确上界为

$$p_{\mathrm{upper}} = 1 - \delta^{1/n}.$$

当 $n = 100, \delta = 0.05$ 时，仍约为 2.95%。这不是证明事故率为零；也不是在环境漂移、相关回合或挑选测试条件后仍成立的保证。安全评测要预先规定事件定义、测试分布、样本量和停测规则。

若通过重要性采样或对抗测试增强稀有事件，应区分发现漏洞与估计自然发生率。故意集中采样困难场景可以有效发现失败，却不能把这个失败频率直接当成真实部署概率；恢复概率解释需要已知采样机制与相应校正。

## 9.5 鲁棒 MDP：环境不确定性放进哪个集合

设真实转移属于集合 $\mathcal{U}$，鲁棒目标为

$$\max_{\pi} \min_{P\in\mathcal{U}} J(\pi; P).$$

“最坏情况”只对所选集合有意义。集合太窄会遗漏真实变化，太宽会让策略为不可能的灾难牺牲全部效用。集合的来源可以是物理参数范围、统计置信集或明确的扰动预算；应说明它是数据推导、先验指定还是人为压力测试。

在有限状态动作、折扣和有界奖励下，若不确定性满足适当的状态动作矩形性，即每个 $(s,a)$ 的转移可从 $\mathcal{U}_{s,a}$ 独立选择，典型鲁棒 Bellman 算子为 [59]

$$(\mathcal{T}_{\mathrm{rob}}V)(s)=\max_{a}\left[r(s,a)+\gamma\min_{p\in\mathcal{U}_{s,a}}\sum_{s'}p(s')V(s')\right]. \tag{9.4}$$

内层最小值允许对手为每个状态动作选择最不利转移。若未知量是一台机器整段回合固定的同一个质量参数，转移之间存在耦合，任意逐状态选择不同参数可能过度保守，也可能不再对应原先的非矩形问题。此时不能未经证明套用这个动态规划式。

域随机化则通常优化 $\mathbb{E}_{P\sim q}[J(\pi;P)]$。它增加训练分布的多样性，却一般是平均目标，不等于集合内最坏情况最优。对抗训练优化某个受限攻击者；攻击者没找到失败，也不表示更强攻击者不存在。报告中应明确区分随机扰动均值、指定集合最差值、有限搜索找到的最差值和已证明的界。

## 9.6 从优化目标到部署边界

实际执行常在策略之外增加动作过滤、备份控制器或运行时监测。例如，先由策略给候选动作，再由 shield 检查是否会违反约束，不可接受时转交备份。其保证取决于安全集合、模型误差界、可恢复性和求解时限。若传感器缺失或约束集合已经无可行动作，过滤器也未必能够保护系统。

过滤后的实际执行策略与原 actor 不同。记录数据时要保存建议动作、实际动作和接管理由，否则 critic 可能把备份控制器的成功错误归因给原策略。评测也要报告介入率；一个几乎每步被接管的高成功率策略，不能被描述为已经自主学会安全控制。

从仿真转真实系统时，本书只处理决策层原则：在保留参数上检查零样本表现，校准观测与延迟，再按明确预算进行少量适应。机器人硬件、执行器接口与数据工程由《具身智能》承接。对本书的通用算法来说，关键是不要把“模拟器里未出现事故”当成真实环境的结论。

**一张联合报告表**

对每个候选策略，在同一测试分布上报告回报均值及区间、成本均值与阈值、违约回合比例、指定 $\alpha$ 的左尾 CVaR、最严重事件、干预率以及执行延迟。另用独立扰动集测试鲁棒性，记录训练是否见过这些扰动。表格中的空项应写明未测，不能用一个加权总分隐藏它们。

## 9.7 练习：一个约束是否真是所需约束

设碰撞成本仅在碰撞发生的那一步为 1，其他为零。比较“不折扣的回合首次碰撞指示期望”“折扣碰撞次数期望”和“任意时刻绝不碰撞”三个要求。提示：一次事故发生得越晚，第二个指标越小，但第一个指标不变；第三个指标还排除了所有正概率事故。再说明成本阈值按 $1-\gamma$ 归一化时，哪些数值必须同时变化。

# 第 10 章 表示、泛化与迁移

一个策略在训练环境中得到高回报，只说明它适应了见过的数据与交互协议。换一幅背景、一个奖励权重或一组工具规则，可能改变完全不同的数学对象。本章先把“换了什么、允许学习多少”写清，再讨论哪些知识有理由复用。第 9 章的鲁棒目标要求在声明的不确定集合中表现可靠；这里讨论的泛化首先是一个训练与测试协议，不能仅凭方法名称获得保证。

## 10.1 先定义任务族，再定义泛化

用 $\xi$ 标识一个任务实例，可包含动力学参数、奖励、初始状态分布、观测映射和终止规则。记相应问题为 $\mathcal{M}_\xi$，回报为 $J_\xi(\pi)$。一种零样本测试目标是

$$J_{\text{test}}(\pi) = \mathbb{E}_{\xi \sim p_{\text{test}}}[J_\xi(\pi)],$$

其中 $p_{\text{train}}$ 决定学习时可接触什么，$p_{\text{test}}$ 决定结论要推广到哪里。二者可以相同，也可以有意不同；都要说明任务生成规则和有限样本如何抽取。即使它们相同，测试仍须使用未参与训练与选择的新实例。在同一个训练地图上更换动作采样种子，只测到该地图内的随机性，不能代替新地图测试。

“分布外”不是一个统一难度等级。至少应分别记录以下变化：

- **观测变化：**状态转移和奖励相同，背景、照明或传感器噪声改变；
- **动力学变化：**动作的后果改变，例如质量、转移概率或工具响应延迟改变；
- **目标变化：**动力学保持，奖励权重、目标位置或允许的成本改变；
- **结构变化：**状态空间、地图拓扑、对象数或可用动作改变；
- **时间变化：**部署过程中的用户、对手或环境机制持续漂移。

这些变化可能组合出现。训练区间内的新参数值是插值测试；超出区间是参数外推；把各自见过的因素重新组合，是组合测试。它们都不能由训练回报推出。报告“泛化提高”时，应写成对哪一种变化、相对哪个基线、在何种预算下提高。

### 10.1.1 训练、验证与测试承担不同工作

训练集用于更新参数，验证集用于选超参数、checkpoint 和停止规则，测试集用于最终报告。如果观察测试结果后继续挑模型，这一测试集就承担了验证作用，需另留新的最终测试集。切分单位应对应预期的新信息：同一轨迹的相邻片段不应分到训练与测试两侧；同一地图的轻微改色也不一定构成独立任务。若目标是跨用户或跨场景迁移，应按用户或场景分组切分，而非随机打散 transition。

归一化器、词表、状态编码器、奖励尺度和任务采样器都可能泄漏信息。例如使用全部测试观测拟合归一化统计量，是一种利用未标注测试数据的适应，可以作为明确声明的协议研究，却不应再称为完全冻结的零样本评估。评估环境种子、训练种子和任务实例标识应分开保存。

## 10.2 表示应保留哪种决策信息

设观测历史为 $h_t$，编码器产生 $z_t = f_\eta(h_t)$。一个控制表示是否充分，取决于它是否保留预测奖励、终止和未来决策所需的信息。在有限 MDP 的精确状态聚合中，若同一组内的状态具有相同的可用动作集合、对每个动作具有相同的期望奖励，并且转移到每个聚合组的概率相同，就可以在聚合空间写出一致的 Bellman 递推。像素相似不蕴含这个条件：外观相同的门可能一扇可开、一扇上锁；外观差异很大的两个画面也可能对应完全相同的控制状态。

重建、动态预测、对比学习和奖励预测都能为表示提供训练信号，但每种信号都有取舍。重建可能花容量记录背景；逆动力学可能忽略动作无法改变、却决定危险的因素；数据增强若删去目标标记，会强迫策略对本应区分的状态不敏感。因此先声明希望不变的因素，再检查这种不变性是否保持动作含义、奖励与终止。在 POMDP 中，仅用当前图像未必足够，增加网络容量也不能恢复已经丢弃的历史信息。

**一次能区分机制的表示实验**

固定策略骨干、交互预算和调参预算，比较无辅助目标与一个辅助目标。分别测试背景变化和动力学变化，保留原始回报、成功率与失败位置。再冻结编码器，只重新训练小型策略头：若只有端到端重训受益，证据支持的是整个训练过程，而不能单独归因于可迁移表示。线性探针能检验某些信息是否可解码，但不能单独证明策略会利用它完成任务。

## 10.3 Successor features：哪些变化可以线性复用

假设一族任务共享状态、动作、转移核、折扣与特征语义，只有奖励权重改变。令每次转移的特征为 $\varphi(s,a,s') \in \mathbb{R}^k$，奖励满足

$$R_w(s,a,s') = \varphi(s,a,s')^\top w。$$

若任务会真正终止，吸收自环上的特征统一设为零；“步数”只统计终止前的真实转移，到达终点的那次转移仍可具有非零事件特征。这样无限吸收扩展与有限回合记分一致。对固定策略 $\pi$，定义 successor features（后继特征）

$$\psi^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \varphi(s_t,a_t,s_{t+1}) \,\middle|\, s_0 = s, a_0 = a\right]。 \tag{10.1}$$

在有界特征和 $\gamma < 1$ 下可交换求和与期望，因此

$$Q_w^\pi(s,a) = \psi^\pi(s,a)^\top w。$$

这个分解把“固定行为将访问哪些特征”与“新任务如何给特征定价”分开 [60]。例如在第 1.3 节的 MDP 中，可令特征分别统计步数、到达目标和风险事件；改动这些事件的奖励权重时，可以对同一策略的后继特征重新加权，这一代数构造无需重新估计该策略的全部访问过程。

分解给出的是固定策略的价值，并不说明旧策略对新奖励仍最优。有若干旧策略时，可据它们在新奖励下的动作价值选择行为，形成广义策略改进的思路；近似误差和旧策略覆盖范围决定了可期待的改进程度。若新任务改变转移、可达性、动作含义或终止规则，$\psi^\pi$ 通常也要改变。若特征表示不足以表达新奖励，重估 $w$ 无法弥补缺失信息。不能把对新奖励的线性重估概括为对任意新环境的即时迁移。

## 10.4 Meta-RL：适应本身也是评测对象

元强化学习从任务分布中学习一种适应机制。将任务内的适应数据记为 $D_{\text{adapt}}$，适应算子记为 $\mathcal{U}(\theta, D_{\text{adapt}})$，可写出示意目标

$$\max_\theta \ \mathbb{E}_{\xi\sim p_{\text{train}}} \mathbb{E}_{D_{\text{adapt}}\sim \mathcal{M}_\xi}\left[J_\xi\left(\pi_{\mathcal{U}(\theta,D_{\text{adapt}})}\right)\right]。$$

内层数据与适应后评价数据应按协议分开。若适应期间的试探行为也有成本，则应同时报告它们的回报和违约，不能只呈现适应完成后的最好成绩。

三条代表路线改变的对象不同：

- **MAML**：学习适合少量梯度更新的参数初值。测试时使用任务内数据计算梯度，必须说明更新步数、步长、数据量和是否使用奖励 [61]。
- **Duan 等的 RL$^2$**：跨任务训练循环策略，测试时主要通过隐状态累积观测、动作和奖励来适应，外层权重可保持冻结。隐状态在同一任务的多个回合间是否保留、在新任务何时重置，是算法定义的一部分 [62]。该引用对应 *RL$^2$: Fast Reinforcement Learning via Slow Reinforcement Learning*，不与 Wang 等的 *Learning to reinforcement learn* 混用。
- **PEARL**：根据任务上下文推断潜变量的近似后验，并让策略依赖该潜变量，利用离策略数据进行元训练。测试适应通常更新上下文后验并采样任务潜变量，而不必对策略权重做梯度更新；上下文采样、潜变量更新与探索协议共同决定表现 [63]。

“网络参数不更新”不等于零适应：隐状态、潜变量、检索记忆和归一化统计量的改变都可能利用测试信息。相反，允许梯度更新也不必然比上下文推断获得更多环境交互。公平比较应统一可访问的信息、交互次数、适应计算量和评价时点。

### 10.4.1 将测试适应预算写成可执行协议

一个可复核的协议可写为：每个新任务先收集 $B$ 次交互，最多执行 $K$ 次参数更新，再在独立回合评估；任务切换时重置指定状态。$B = 0$ 且不利用测试数据更新任何状态统计量时，才是这里的冻结零样本协议。如果任务标识直接提供给策略，应与需要从奖励推断任务的设置分开。多回合适应还须说明环境是否重置、潜在任务是否保持，以及历史是否跨回合保留。测试时增加搜索、采样或工具调用也会增加预算，应在同一质量 – 成本曲线上比较。

## 10.5 非平稳部署与统计单位

任务之间相互独立，与一个任务在时间上持续漂移，是两种不同设定。对后者，可将环境写为 $\mathcal{M}_{\xi_t}$，声明变化是突变、渐变还是对策略有响应。若测试目标是未来表现，随机打散时间会把未来信息泄漏到过去，应按时间切分或滚动回测。变化检测与上下文策略可以作为方案，但检测延迟、误报与恢复成本都必须计入结果。旧 replay 是有用记忆还是过时证据，取决于动力学和奖励变化的具体部分。

持续学习需同时报告新任务学习速度和旧任务遗忘，并声明旧任务是否会再次出现、是否允许保留原始数据、记忆容量有多大。只展示最后一个时间窗口的回报，无法区分“适应成功”与“忘掉了旧任务”。

统计汇总也要尊重层次：同一训练 seed 下的多个测试回合共享一套参数，同一地图的多个观测共享环境结构。应分别给出训练 seed、测试任务与回合数，并以声明的任务权重汇总。比较两种算法时，可以对相同测试任务做配对分析；区间估计应保留任务或训练运行的聚类结构，不能把成千上万条相关 transition 当成独立训练证据 [64, 65]。

**练习与解题提示**

设训练任务只改变四状态 MDP 的目标奖励，测试任务还改变一条转移概率。判断哪些量可以复用，哪些必须重新估计；再分别写出冻结零样本与允许 100 次测试交互的协议。提示：奖励权重重估不改变已缓存的后继特征，而转移概率改变了特征的未来访问分布。最后为一个循环元学习器列出任务切换时必须重置的所有状态，检查是否把测试任务的信息带到了下一任务。

# 第 11 章　语言模型与智能体的反馈学习

前面各章暂时把奖励看成环境给出的量。语言模型任务揭示了另一个层次：人类偏好、AI 评审、单元测试和真实工具结果都能提供反馈，但它们并不测量同一个目标。算法可以认真地优化一个有偏的分数；因此本章同时讨论策略目标和反馈的生成过程。

## 11.1　回答级与 token 级的两种建模

设提示 $x \sim p_X$，回答 $y = (y_1, \dots, y_T)$。回答级建模把 $x$ 当上下文、整个 $y$ 当一次动作，构成 contextual bandit（上下文赌博机）。没有外部工具时，输出 token 之间的状态转移只是把已生成内容接到前缀中；用 token 级 MDP 描述同一过程则令 $s_t = (x, y_{<t})$、$a_t = y_t$，并有

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{11.1}$$

如果只在末尾评价回答，前面的 token 奖励为零，结尾获得 $r(x, y)$。通常使用无折扣的有限长度目标；这不属于前面无限时域 $\gamma < 1$ 的收缩性定理，需要由有限生成长度保证回报可积。

这两种描述可以有相同的回答分布，却产生不同的估计器。回答级 REINFORCE 的样本梯度为

$$g_{\text{seq}} = r(x, y)\, \nabla_\theta \log \pi_\theta(y \mid x).$$

token 级 actor–critic 可学习前缀价值以分配信用。后者可能降低方差，也引入价值估计偏差和额外计算。不能从“没有单独 critic”推出“不存在长期信用分配”，也不能从使用 token 序列推出任务必然需要完整通用 RL。

EOS 是策略选择的结束动作；生成长度上限的含义则取决于任务。若部署规定超过预算就是失败，预算应纳入任务状态和奖励。若只是采样器提前停止而未完成任务，就缺失了真实后续；给它零分、丢弃或接一个价值估计都会改变估计问题，必须分别记录。用 GRPO 时没有现成的 critic 负责修复这一缺失。

## 11.2　从偏好标签到奖励模型

经典 RLHF（人类反馈强化学习）流程先用示范做监督微调，再收集同提示回答对偏好，拟合奖励模型，最后优化策略并约束其偏离参考模型的程度 [33, 66, 67]。这是一种代表性流程，不是 RLHF 的唯一操作定义。监督微调、奖励建模和策略优化解决不同问题，改变任一步都可能改变最终表现。

### 11.2.1　Bradley–Terry 模型究竟假设了什么

给定严格偏好 $y^+ \succ y^-$，一个常用模型为

$$\Pr(y^+ \succ y^- \mid x) = \sigma\big(r_\phi(x, y^+) - r_\phi(x, y^-)\big), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}. \tag{11.2}$$

奖励模型最小化负对数似然。它只通过差值被识别：给同一提示的全部回答奖励加常数，偏好概率不变。若不固定标签噪声温度，奖励尺度也会与温度混淆。式 (11.2) 描述二元严格偏好，没有自动建模平局。对平局需要另行定义标签似然、软标签或排除规则，不能在差值里任意减去一个含义不明的“ties”。

以贯穿全书的 MDP 为例，$\gamma = 0.9$ 时两条完整轨迹回报分别为 2 和 3.33。若把它们按回报差放入式 (11.2)，则“绕行更好”的模型概率为 $\sigma(1.33) \approx 0.791$。这是给定噪声模型的教学计算，不是人

类标注者的实测一致率。若标注者只看即时奖励，偏好可能反过来；这说明学习到的标量依赖反馈协议，不能仅用优化损失判断它是否忠实于长期目标。

### 11.2.2 对奖励误差做分布检查

奖励模型通常在旧策略回答上拟合，新策略却会主动寻找高分区域。原分布上的排序准确率不能保证新分布上的有效性。至少按提示领域、长度、格式、语言、是否拒答和难度分层检查；保留与训练评审独立的审查样本，并记录标注者分歧，而不是用一个平均准确率吞掉分歧。当两个回答的偏好差异很小，重复标注可能比再训练更大的奖励模型更有信息。

优化过程还可能把长度、语气和格式当成质量的代理。应联合画出奖励分数、独立任务分数与回答长度，并检查分数提高时的失败样例。若奖励上升而独立检验下降，继续扩大策略更新通常只会加强对代理的利用。

## 11.3 KL 正则目标与 DPO 的推导边界

对固定奖励模型和提示分布，考虑

$$\max_{\pi} \ \mathbb{E}_{x\sim p_X,y\sim\pi}[r(x,y)] - \beta \mathbb{E}_{x\sim p_X} D_{\mathrm{KL}}\big(\pi(\cdot\mid x)\|\pi_{\mathrm{ref}}(\cdot\mid x)\big), \qquad \beta>0. \tag{11.3}$$

在参考策略具有正支持、分配函数有限、允许优化任意回答分布的条件下，逐提示的最优解为

$$\pi^*(y\mid x)=\frac{\pi_{\mathrm{ref}}(y\mid x)e^{r(x,y)/\beta}}{Z(x)}, \quad r(x,y)=\beta\log\frac{\pi^*(y\mid x)}{\pi_{\mathrm{ref}}(y\mid x)}+\beta\log Z(x). \tag{11.4}$$

这不是对有限神经网络、有限数据和不完整优化过程的精确保证。参考概率为零的回答不可能通过有限 KL 获得正概率；参考模型中的数据覆盖因此仍然重要。

将式 (11.4) 中的奖励差代入 Bradley–Terry 模型，$\log Z(x)$ 相消，得到 DPO 损失 [68]：

$$\Delta_\theta(x,y^+,y^-)=\log\frac{\pi_\theta(y^+\mid x)}{\pi_{\mathrm{ref}}(y^+\mid x)}-\log\frac{\pi_\theta(y^-\mid x)}{\pi_{\mathrm{ref}}(y^-\mid x)}, \tag{11.5}$$

$$\mathcal{L}_{\mathrm{DPO}}(\theta)=-\mathbb{E}_{\mathcal{D}_{\mathrm{pref}}}\log\sigma\big(\beta\Delta_\theta(x,y^+,y^-)\big). \tag{11.6}$$

这解释了 DPO 与 KL 正则奖励最优化的关系，也说明它的假设。偏好应当由同一类可比较标量差解释，数据需覆盖所关心回答，参考策略及 $\beta$ 的角色要固定。离线训练不需要每一步在线采样，也不需要显式奖励网络，但并没有消除反馈误差或数据支持不足。

实现时 $\pi(y\mid x)$ 是回答 *token* 对数概率之和，提示和 padding 不计入损失。EOS 是否计入、截断样本如何处理要保持一致。若改成每个回答 token 的平均对数概率，就改变了式 (11.6) 中的建模目标，不能仍声称是同一推导。参考模型冻结；在不需要其梯度时以无梯度方式计算。应先在两条人工回答上检查损失对 $\Delta_\theta$ 的导数符号，再上真实数据。

**一个无需训练模型的概率检查**

设胜者在新/参考策略中的回答概率比为 2，败者为 1/2。则 $\Delta_\theta=\log 4$，当 $\beta=1$ 时，DPO 模型的胜者概率为 4/5，单对损失为 $-\log(4/5)$。交换标签后损失为 $-\log(1/5)$。这只测试概率和标签方向，不检验真实反馈是否正确。

## 11.4 RLAIF 与验证器：更大规模不等于更少假设

RLAIF 用 AI 反馈代替或补充人的偏好；Constitutional AI 的代表流程利用原则驱动的批评、修订和 AI 评审 [69]。反馈成本可以下降，但规范选择、评审提示、位置偏差、模型同源性和拒答偏好

仍是学习系统的一部分。评审结果应保存模型版本和提示；同一模型既出题、作答又评审时，需要额外的独立验证。

可验证奖励 RL（RLVR）把程序测试、答案比较器或形式检查器用作反馈来源。验证器提供可重复执行的判据，却不自动证明“它检查的就是用户真正想要的东西”。程序通过有限单元测试，不等于对所有输入正确；格式可解析，不等于推理可靠；数学终值相同，也可能来自数据泄漏或解析器漏洞。

将潜在目标记为 $z$、验证输出为 $v$，则应同时检查假阳性 $\Pr(v=1 \mid z=0)$、假阴性 $\Pr(v=0 \mid z=1)$ 和未覆盖区域。实际中没有完整的 $z$，可用保留测试、人工反例与不同实现的验证器估计局部错误。这类估计不能被宣传为全域正确性。

过程奖励模型评价中间步骤。Lightman 等的工作为过程监督提供了实验依据 [70]；这里援引的是该论文的监督设计和任务内比较，不能推出“中间过程打分高就能保证最后答案正确”。过程切分、标注协议和最终验证应分别记录。

## 11.5 GRPO：组内相对反馈怎样进入策略梯度

DeepSeekMath 的 GRPO 从同一提示采样 $K$ 个回答，以组内奖励统计量构造相对优势，省去独立的价值网络 [71]。对只给终局奖励的版本，可写为

$$\bar{r} = \frac{1}{K}\sum_{i=1}^{K} r_i, \qquad \hat{A}_i = \frac{r_i - \bar{r}}{s_r + \varepsilon_{\text{num}}}. \tag{11.7}$$

应说明 $s_r$ 使用总体还是样本标准差，$\varepsilon_{\text{num}}$ 只避免除零，不凭空产生信息。若 $K=1$ 或组内全部同奖，所有相对优势为零；KL 正则可能仍有梯度，但该组没有奖励区分信号。

定义每个 token 的新旧概率比 $\rho_{i,t}$，一个代表性的回答等权 surrogate 是

$$L = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{T_i}\sum_{t=1}^{T_i}\left[\min\{\rho_{i,t}\hat{A}_i, \operatorname{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\} - \beta k_{i,t}\right]. \tag{11.8}$$

这里 $k_{i,t}$ 表示指定的 KL 惩罚或其估计，必须随实现说明采样分布和梯度处理。优势、旧概率和奖励均从采样批固定下来；不能让 autograd 通过奖励均值和标准差回传到采样决策。逐 token 截断不是完整回答重要性采样的无偏修正，也不是硬 KL 约束。

把 $\sum_t / T_i$ 改成对全部有效 token 等权，会改变长短回答的权重。把组内标准差移除，也会改变不同难度提示的相对梯度尺度。组均值包含样本自身，有限组的基线估计与随机尺度都影响估计器；因此不能直接把式 (11.7) 称为真实优势的无偏估计。Dr. GRPO 对长度与归一化偏差的分析，以及 DAPO 对采样、token 损失和超长处理的设计，提醒我们应逐项比较目标与实现 [72, 73]。本书不把这些具体实验条件下的改进写成普适优胜结论。

**概率比、KL 与冻结对象各有参照**

PPO/GRPO 的分母是采样时的旧策略；KL 的参照通常是固定参考策略。二者不可混用。单次采样的 $\log \pi_\theta(a \mid s) - \log \pi_{\text{ref}}(a \mid s)$ 可以为负，只有按 $\pi_\theta$ 对动作取期望才是该条件分布的 KL。若样本来自旧策略，所用 KL 估计与梯度修正还需另行审查。

DeepSeek-R1 报告了 R1-Zero 的 RL 路线及包含冷启动和多阶段训练的 R1 路线 [74]。必须区分两者，不能将“某一实验未用人工推理轨迹冷启动”概括为整个训练过程中没有监督数据。本章只讨论公开材料支持的训练反馈结构，未复现其模型训练，也没有将其报告的性能推广到任意开放式任务。

## 11.6 Agentic RL：外部状态让信用分配更具体

当动作包含检索、运行代码和修改工具状态时，环境转移不再只是字符串拼接。用可用历史 $h_t$ 决策更稳妥；短上下文、外部记忆和隐藏工具状态使任务常表现为 POMDP。轨迹应保存输入观测、动作、工具返回、权限与版本、耗时、奖励来源和最终状态，否则无法分辨成功来自策略、初始环境差异还是评测器异常。

一个代码修复任务可有最终隐藏测试奖励和中间可见测试反馈。中间测试作为观测能够帮助决策；若也把它计入奖励，需检查重复运行是否能无限得分。对修改、提交、付款等不可逆动作，失败后重新生成不能被当成无成本的环境 reset。训练环境需要状态隔离与明确终止，评估应记录恢复成本与副作用。

**Listing 11.1** 工具反馈任务的最小采集结构（流程示意，未运行模型训练）

```
task, obs = env.reset(task_id, seed)
history = [obs]
while not env.finished:
  action, old_logp = policy.sample(history)
  next_obs, reward, terminated, truncated, info = env.step(action)
  store(history, action, old_logp, reward,
      terminated, truncated, info["final_observation"])
  history = history + [action, next_obs]
  if terminated or truncated:
    break
verify_final_state_with_held_out_checks(task, env.state)
```

长程任务的策略更新还需明确回答级、操作级或 token 级信用单位、重试预算，以及旧轨迹回放与 on-policy 估计的兼容性。完整执行系统见《LLM 智能体系统》，数据与训练流水线见《大语言模型后训练》。

## 11.7 统一报告：反馈质量与优化收益一起看

| 报告对象 | 最少需要的信息 |
|---|---|
| 任务与数据 | 提示来源、划分、去重、可用工具、环境初始状态、污染检查与版本。 |
| 奖励与验证 | 人工/AI/规则来源、模型和提示版本、覆盖与失败反例、奖励尺度和截断处理。 |
| 采样与优化 | 回答数与 token 数、组大小、温度、有效 token 掩码、损失归一化、KL 方向、更新轮次。 |
| 质量与成本 | 独立成功率、pass@1 与 pass@k 的采样协议、回答长度、训练及推理墙钟、硬件和总 token 预算。 |
| 统计与失败 | 训练随机种子、共享评估提示、按提示/任务聚类的区间、奖励投机、工具副作用与恢复成本。 |

pass@k 随 $k$ 增大而改变测试时预算，不能直接与一次采样成功率比较。同一道题上的多个回答也不是多个独立训练随机种子。若独立奖励检验只覆盖一小部分提示，应同时报告抽样规则和样本量。RL 与语言模型综述可用来寻找问题和资料 [75, 76]，但实际实验仍需回到明确版本的原始方法和评测协议。

## 11.8 练习：先分辨问题，再选择优化器

1. 对四状态两条轨迹，若 $\gamma$ 从 0.9 改为 0.5，按回报给偏好会怎样变化？提示：长路径回报变为 $0 + 0.5 + 0.25 \times 3 = 1.25$，偏好与目标定义一起变化。
2. 组内奖励为 $(1, 1, 1, 1)$ 时，增大数值稳定常数会产生奖励学习信号吗？提示：分子仍全零；应区分“没有区分信号”和“优化器没有任何梯度”。
3. 一个验证器只检查输出最后一行。构造可得高分但不完成任务的回答，再设计独立检验。说明新检验仍有哪些输入没有覆盖。
4. DPO 中将对数概率和改为均值，为什么不只是实现上的数值稳定化？提示：比较两条不同长度回答在式 (11.6) 中的权重。

# 第 12 章 实验、诊断与可复现性

一个更新式能够运行，并不意味着它回答了原来的决策问题。奖励是否来自当前转移，时间上限是否误写成终止，评估时有没有撤掉训练所依赖的探索，都会改变最后的结论。本章把前面的数学定义落到三个小实验上：四状态 MDP 检查 Bellman 与 TD；短轨迹检查 GAE、概率比率和截断；风险路径任务比较两种表格控制算法。全部数值来自随书代码实际运行或明确列出的手算输入，没有深度网络训练结果。

## 12.1 先说明证据回答什么问题

强化学习的实验至少要区分四个量：最终策略质量、环境交互成本、计算成本和失败风险。同样的回报可能来自更多交互、更慢的推理，或者把少数严重失败平均掉。报告之前应先写出任务、数据收集策略、目标策略和预算单位，再决定如何汇总。

本章的四状态与短轨迹输入是教学构造，其解析值能够独立核验；悬崖任务是本次 CPU 实测；后文的 PPO 与 SAC 扩展是尚未运行的方案。论文中的大规模结果属于各自的环境、实现与预算，不能把本章小实验当成对那些结果的复现。实验日志可以证明本次发生了什么，不能单独证明方法在另一个任务上更好。

| 问题 | 应保留的量 | 常见误读 |
| --- | --- | --- |
| 最终质量 | 固定检查点的独立评估回报、成功率、失败分布 | 只展示训练时最高回报 |
| 样本效率 | 环境转移数；真实交互另报时长或费用 | 把 episode 数当成相同交互预算 |
| 计算效率 | 训练、评估和总墙钟；设备与并行度 | 不说明模拟器或数据生成成本 |
| 可靠性 | 每个种子的结果、截断率、跌落与恢复行为 | 只比较均值，遗漏循环或长尾 |

多种子不是把一条运行切成许多独立样本。一条训练得到的策略评估 200 次，主要减小的是该策略条件下的评估噪声；它没有产生 200 次独立训练。跨任务汇总还要说明任务权重、归一化基准和统计单位。随机种子不多、分布重尾时，均值及其区间尤其需要配合原始点和失败案例阅读 [64, 65]。

## 12.2 实验一：同一 MDP 的解析值与计算值

沿用第 1 章的四状态任务。$s_0$ 的 direct 动作以奖励 2 到达终止状态 $s_\perp$；via 以奖励 0 到达 $s_1$。随后 $s_1$ 以奖励 1 到达 $s_2$，$s_2$ 以奖励 3 到达 $s_\perp$。所有转移确定，$V(s_\perp) = 0$；配套代码以 T 表示这个终止状态。固定策略在 $s_0$ 以各 1/2 的概率选择两条路线。

### 12.2.1 从闭式结果核对两种 Bellman 计算

对这个固定策略，先从后向前算出

$$V^\pi(s_2) = 3, \qquad V^\pi(s_1) = 1 + 3\gamma, \qquad V^\pi(s_0) = 1 + \frac{\gamma}{2}(1 + 3\gamma).$$

最优值则把第一步的策略平均换成最大值：

$$V^*(s_0) = \max\{2, \gamma(1 + 3\gamma)\}.$$

程序一方面求解固定策略的线性方程 $(I - \gamma P_\pi)V = r_\pi$，另一方面从零执行同步策略评估和最优价值迭代。三个方法的作用不同：线性求解与解析式核对策略评估，最优算子核对动作选择，不能把固定策略的 2.665 当成最优值。

**表 12.1** 四状态 MDP：闭式值与实际计算一致，最大绝对误差小于 $10^{-12}$。

| $\gamma$ | $V^\pi(s_0)$ | $V^\pi(s_1)$ | $V^\pi(s_2)$ | $V^*(s_0)$ |
|---|---|---|---|---|
| 0 | 1 | 1 | 3 | 2 |
| 0.5 | 1.625 | 2.5 | 3 | 2 |
| 0.9 | 2.665 | 3.7 | 3 | 3.33 |

在 $\gamma = 0.9$ 时，同步策略评估的 $s_0$ 值依次为 $0, 1, 1.45, 2.665$；价值迭代的对应值为 $0, 2, 2, 3.33$。前几轮看起来不动，不意味着没有学习：$s_2$ 的信息还没有传播到 $s_0$。本例最长只有三条转移，三次备份后已经达到精确值，随后一次不变备份触发停止。这种有限步现象来自无环结构，不能推广为一般 MDP 的迭代次数保证。

### 12.2.2 一次 TD 更新没有获得完整的价值函数

令更新前 $\hat{V}(s_1) = 1, \hat{V}(s_2) = 2$，采到 $s_1 \xrightarrow{r_t=1} s_2$，取 $\gamma = 0.9, \alpha = 0.5$。一步计算得到

$$y_t = 1 + 0.9 \times 2 = 2.8, \quad \delta_t = 2.8 - 1 = 1.8, \quad \hat{V}_{\text{new}}(s_1) = 1 + 0.5 \times 1.8 = 1.9.$$

解析值 $V^\pi(s_1) = 3.7$ 没有因此变成 1.9。TD 使用当前的 $\hat{V}(s_2)$ 作为目标的一部分，目标还不准确，且这里只观察了一条转移。这一检查验证的是时序、目标和更新位置，不是收敛定理。

当 $\gamma = 0$ 时，所有后继价值都应消失；真正终止时，后继价值也必须消失。时间限制则是另一种情形：如果它只是采集器停止记录而任务本身仍会继续，就需要在实际末观测处 bootstrap。若有限时域本身就是任务定义，应把剩余时间纳入状态并按任务终止处理。不能只因为程序重置了环境，就推断后继价值等于零。

## 12.3 实验二：逐项核验短轨迹与截断

### 12.3.1 把 reward-to-go、TD residual 和 GAE 放在同一张表里

取三个转移的奖励 $(r_0, r_1, r_2) = (1, 0, 2)$，估值 $(V_0, V_1, V_2, V_3) = (0.5, 0.8, 0.4, 0)$，最后一条转移真正终止，$\gamma = 0.9, \lambda = 0.8$。这里的 reward-to-go 是到真实终点的奖励和，没有额外价值 bootstrap。

例如 $G_0 = 1 + 0.9 \times 0 + 0.9^2 \times 2 = 2.62$，而 $\delta_0 = 1 + 0.9 \times 0.8 - 0.5 = 1.22$。由于 $\gamma\lambda = 0.72$，有

$$\hat{A}_1 = -0.44 + 0.72 \times 1.6 = 0.712, \qquad \hat{A}_0 = 1.22 + 0.72 \times 0.712 = 1.73264.$$

本例的 $V_t + \hat{A}_t$ 分别为 $2.23264, 1.512, 2$，并不全部等于 $G_t$。$\lambda < 1$、bootstrap 值不精确时，这正是不同估计器应有的区别。

程序用逆向递推计算 GAE，核验程序另用分数算术直接求有限和，两者没有共享同一个递推实现。另测 $\lambda = 0$ 时优势应等于一步 residual，$\gamma = 0$ 时优势应等于 $r_t - V_t$。这些边界比只检查输出数组形状更能发现语义错误。

**表 12.2** 人工构造短轨迹的精确算例；小数不是训练性能。

| $t$ | $r_t$ | $G_t$ | $\delta_t$ | $\hat{A}_t^{\mathrm{GAE}}$ |
|---|---|---|---|---|
| 0 | 1 | 2.62 | 1.22 | 1.73264 |
| 1 | 0 | 1.8 | −0.44 | 0.712 |
| 2 | 2 | 2 | 1.6 | 1.6 |

### 12.3.2 两个掩码解决两个不同问题

令 $b_t = 1 - \mathbf{1}\{\mathrm{terminated}_t\}$；令 $c_t$ 表示下一条 residual 在当前缓存中可用，且与当前转移属于同一回合。程序使用

$$\delta_t = r_t + \gamma b_t V(s_{t+1}^{\mathrm{final}}) - V(s_t), \qquad \hat{A}_t = \delta_t + \gamma\lambda c_t \hat{A}_{t+1}.$$

真实终止把 $b_t$ 和 $c_t$ 都置零；外部时间限制保留 $b_t = 1$，但重置前必须使 $c_t = 0$。缓存恰好用尽时，没有下一条 residual，也应使 $c_t = 0$，同时保留对实际后继状态的 bootstrap。这里的 $s_{t+1}^{\mathrm{final}}$ 是这一次转移的末观测；自动重置包装器若返回了新回合观测，需要从其接口取得末观测并单独保存。

把短轨迹的最后一步改为外部时间限制，末观测估值改为 1.5，则最后一步目标变成

$$y_2 = 2 + 0.9 \times 1.5 = 3.35, \qquad \delta_2 = 3.35 - 0.4 = 2.95.$$

因此该回合前三项 GAE 变成 $(2.43248, 1.684, 2.95)$。它们可以依赖末观测，不能依赖重置后的轨迹。测试故意把下一回合的估值和奖励改成很大的数，前三项仍必须不变。如果把重置观测的估值 10 用作末观测，错误目标会是 11；如果把截断当作终止，错误目标会是 2。两种 bug 的方向相反，却都可能暂时产生看似平滑的训练曲线。

### 12.3.3 PPO 的 min 不能写成统一的概率裁剪

下面另构造四个策略样本，直接给出采样动作的新旧概率和优势，取 clipping 参数 $\epsilon = 0.2$。最后两项的负优势是用于核验符号的输入，并不是上表正优势的训练结果。记 $\rho = p_{\mathrm{new}}/p_{\mathrm{old}}$，每项计算

$$\min\{\rho A, \mathrm{clip}(\rho, 0.8, 1.2)A\}.$$

**表 12.3** 概率比率与 PPO surrogate 的逐项核验。

| $p_{\mathrm{old}}$ | $p_{\mathrm{new}}$ | $A$ | $\rho$ | $\rho A$ | min 结果 |
|---|---|---|---|---|---|
| 0.2 | 0.3 | 1.73264 | 1.5 | 2.59896 | 2.079168 |
| 0.5 | 0.45 | 0.712 | 0.9 | 0.6408 | 0.6408 |
| 0.4 | 0.2 | −1.6 | 0.5 | −0.8 | −1.28 |
| 0.4 | 0.6 | −1.6 | 1.5 | −2.4 | −2.4 |

四项平均为 −0.240008；若优化器最小化 policy loss，符号要再取负。负优势且 $\rho = 0.5$ 时，min 选中更负的 −1.28；负优势且 $\rho = 1.5$ 时，却保留未裁剪的 −2.4。这说明 clipping 修改的是样本 surrogate，既不是把所有比率硬约束在区间内，也不是性能单调改进保证。实际更新还要保存采样时的旧 log-probability，并把优势、回报目标和旧策略统计视为固定数据。

## 12.4 实验三：固定预算下的风险路径控制

### 12.4.1 先公开环境与比较协议

任务采用 $4 \times 12$ 网格，行号从上至下为 $0, 1, 2, 3$。起点为 $(3, 0)$，终点为 $(3, 11)$，底行中间十格是悬崖。动作是上下左右，越界保持原位。普通转移奖励 $-1$；跌落奖励 $-100$ 并回到起点，但**仍处于同一回合**；到达终点才真正终止。每回合最多 500 步，时间限制触发重置，同时保留未观测 bootstrap。这是受经典 cliff walking 启发的自包含实现，不声称与某个第三方环境版本逐位一致。

SARSA 和 Q-learning 都从全零 Q 表开始，以 $\epsilon = 0.1$ 的固定 $\epsilon$-greedy 策略收集数据，最优动作并列时均匀抽样。两者均用 $\alpha = 0.5, \gamma = 0.99$，种子固定为 $0, \ldots, 19$，**每种算法、每个种子恰好使用 40 000 次训练转移**。不按 episode 数停止，不因失败种子追加训练，也没有选择最好检查点。学习率和探索率不递减，所以这个有限预算实验不满足表格渐近收敛论证中的全部条件。

区别只在 bootstrap：SARSA 使用同一行为策略在后继状态抽到的动作；Q-learning 使用后继 Q 的最大值。前者的目标包含继续探索造成的后果，后者试图估计贪心目标策略的值。二者都不自动拥有约束安全保证。

训练完成后冻结 Q 表，独立评估两种策略：纯 greedy，以及保留 $\epsilon = 0.1$ 的策略。每个种子、每种评估策略运行 200 回合；评估期间不更新参数，随机流与训练分开。评估交互另行计数，不计入 40 000 次训练预算。环境本身确定，纯 greedy 若没有并列动作，其重复回合没有新的环境随机性；真正的训练重复单位仍然是 20 个种子。

### 12.4.2 报告单位、学习曲线与分布

回报均为每回合**未折扣奖励和**，而更新内部仍使用 $\gamma = 0.99$；这两个量分别用于描述行为和定义学习目标。训练末期指标使用最后 10 000 次转移内结束的完整回合，包含因 500 步上限而结束的回合；整次预算结束时尚未完成的回合单独记录，不悄悄补齐。跨窗口开始的回合会按结束位置归入窗口，因此学习曲线另报覆盖所有转移的窗口奖励与跌落数。

每个种子先计算自己的均值，再对 20 个种子等权平均。置信区间对种子级指标重采样 5 000 次，报告 percentile 95% 区间，重采样种子为 20261021。它没有把同一策略的 200 个评估回合当成 200 次独立训练；也不校正多重比较，不能解释为普适性能保证。

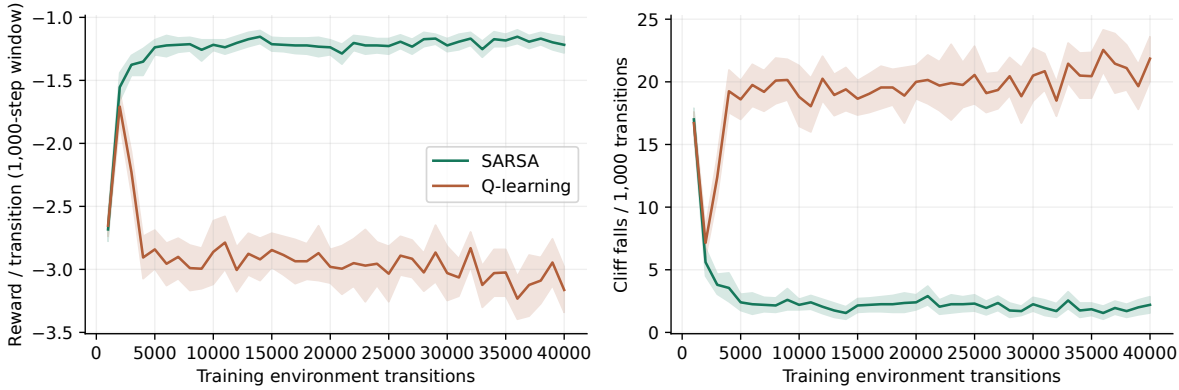



**图 12.1** 本次实测的训练曲线。每点覆盖不重叠的 1 000 次环境转移，实线为 20 个种子的均值，阴影为种子级 bootstrap 95% 点态区间；没有额外曲线平滑。跌落率的分母是转移数。

训练期间，SARSA 末期每千次转移跌落 1.92 次，Q-learning 为 20.84 次；窗口每步奖励分别为 $-1.1901$ 和 $-3.0627$。末期回报的种子配对差 SARSA 减 Q-learning 为 24.20，95% 区间 $[20.71, 26.94]$。

**表 12.4** 固定预算的实际回报结果。括号中为种子级 95% 区间，数值越大越好；测试策略列没有改变训练配置。

| 回报口径 | SARSA | Q-learning |
|---|---|---|
| 训练末 10 000 步 | −28.19<br>[−31.14, −25.87] | −52.40<br>[−53.88, −51.10] |
| 冻结后 greedy | −186.25<br>[−282.95, −89.65] | −13.00<br>[−13.00, −13.00] |
| 冻结后 $\epsilon = 0.1$ | −63.38<br>[−104.42, −33.82] | −51.31<br>[−53.06, −49.42] |

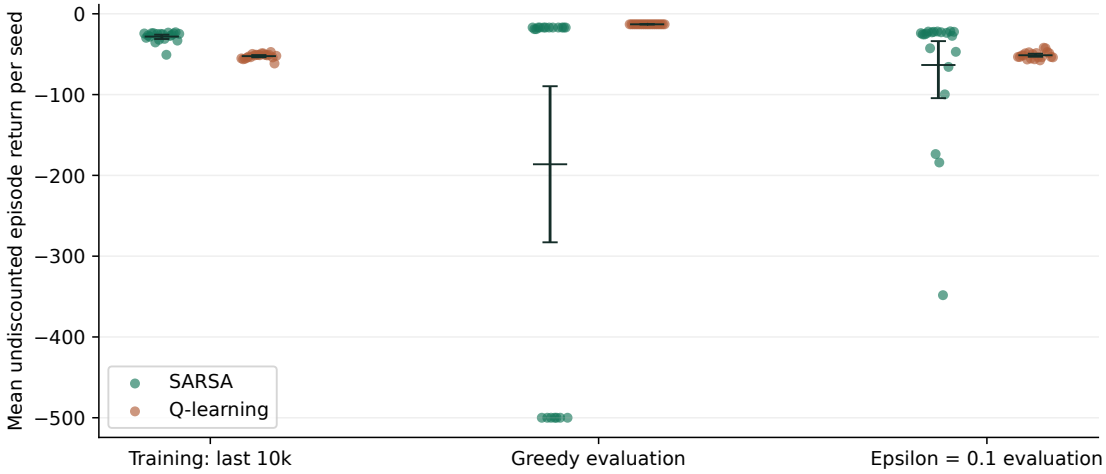



**图 12.2** 每个小点代表一次独立训练种子的回报均值；横向错开只为避免重叠。深色横线为种子均值，误差线为种子级 95% 区间。SARSA 的 greedy 长尾不是绘图异常：7 个种子循环至时间上限。

这些结果支持在本配置的探索训练过程中，SARSA 少付了悬崖代价。它们没有说明冻结部署时谁总是更好。

### 12.4.3 失败种子改变了看似熟悉的故事

冻结后，Q-learning 的全部 20 个种子都学到了 13 步贴近悬崖的 greedy 路线，回报 −13，未跌落。SARSA 的 13 个种子能够到达终点，成功路线长度为 17 或 19；另外 7 个种子在纯 greedy 下停留或循环到 500 步上限。所有 greedy 回合的 SARSA 跌落率同样为零，但成功率只有 65%。**没有跌落不等于完成任务**，仅报告跌落率就会遮住这一失败。

例如种子 5 的 SARSA 路线到达 $(0, 2)$ 后反复选择撞向上边界的动作；种子 10 在 $(1, 0)$ 附近停留。它们的 −500 回报来自耗尽时间，并非跌落惩罚。原始路线与最终 Q 表均保存，可以直接检查当时的 argmax，而不需要凭训练曲线猜测。常数步长下估值仍在波动，SARSA 又针对带探索的后继动作学习；把探索立即撤掉，既改变了评估策略，也可能移除了逃离循环的随机动作。这是与本次记录一致的解释，尚未通过步长或探索消融确立唯一因果。

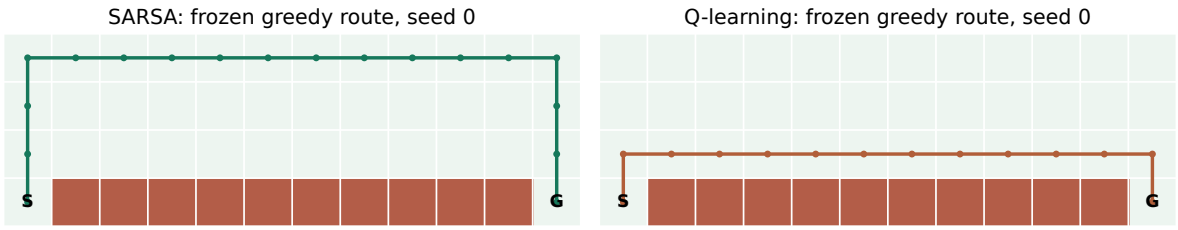



**图 12.3** 按预先固定的首个种子 0 展示两条冻结 greedy 路线。红色格为悬崖，S/G 为起终点。这张图展示路线差异，不代表全部种子；失败种子也完整保存在日志中。

在保留探索的评估中，SARSA 成功率为 97.65%，Q-learning 为 100%；每千次转移的平均跌落率分别约为 1.50 和 20.25。然而 SARSA 的少数策略仍然走得很慢，其种子平均回报从 −21.70 到 −348.29，中位数为 −25.00；Q-learning 从 −42.02 到 −57.61，中位数为 −52.49。SARSA 减 Q-learning 的均值差为 −12.07，配对 95% 区间 $[-52.41, 18.44]$。均值、中位数和跌落率回答了不同问题，不能挑其中一个写成必然优胜的结论。

日志还定义了贴边路线指标：一个回合只要访问过第 2 行、列 1 至 10 中任一格，就记为贴边；它不是跌落事件，也不是安全距离的完整分布。Q-learning 的 greedy 回合全部贴边，SARSA 为 5%，但这 5% 的分母也包含未完成回合。因此该指标必须和成功率、步数及具体路线一起看。

## 12.5 从可运行代码到可核查记录

### 12.5.1 文件、命令与运行成本

实现位于 manuscript/experiments/。默认命令在书籍的 manuscript/ 目录下执行，解释器需已安装 NumPy 与 Matplotlib：

```
python3 experiments/run_minimal_experiments.py --seeds 20 --steps 40000
python3 experiments/validate_reference.py
```

完整绝对路径命令、解释器、工作目录和源码 SHA-256 保存在 run_metadata.json，因此读者可以区分示意命令与本次真正执行的命令。结果默认进入本书 output/experiments/ 下以新版 PDF 同名命名

的目录；图保存在 manuscript/figures/。实验 README 给出独立虚拟环境与固定版本的安装方法。

本次使用 Apple M5 CPU、单个 Python 进程，无 GPU；Python 3.12.14、NumPy 2.3.5、Matplotlib 3.10.7。合计训练 1 600 000 次转移，独立评估 1 102 994 次转移；评估步数由冻结策略的实际回合长度决定，不人为补成相同数量。按本次运行元数据，训练与评估墙钟合计约为 6.5 秒和 4.1 秒，总执行约 11.3 秒，包含统计与制图，不含首次安装依赖。墙钟受机器负载和绘图库缓存影响，只用于说明本次成本，不能当成算法速度基准。

| 记录 | 用途 |
|---|---|
| config.json | 地图、预算、种子、超参数、回报及区间定义 |
| four_state_mdp.json | 各轮 DP、线性解、一次 TD 的输入与结果 |
| trajectory_checks.json | RTG、residual、双掩码、PPO 正负优势核验 |
| raw/ | 每算法、每种子的回合记录、窗口记录、评估回合与最终 Q 表 |
| seed_summaries.csv | 每个训练种子的汇总及训练、评估墙钟 |
| summary.json | 均值、标准差、中位数、范围、区间及种子配对差 |
| reference_validation.json | 独立算术、边界用例和原始账目验证 |

### 12.5.2 独立检查应能推翻错误实现

核验程序不测试预期的算法排名。它用分数算术核对四状态闭式值与短轨迹有限和，检查真实终止、时间截断、缓存用尽、$\gamma = 0$ 和 $\lambda = 0$，并向重置后的回合注入极大 residual，确认优势不跨回合泄漏。环境边界另测撞墙、跌落回起点和到达终点。

原始日志还有一条独立的账目恒等式。若回合长度为 $L$、跌落 $F$ 次，由普通奖励 $-1$ 和跌落奖励 $-100$ 可得

$$G = -L - 99F.$$

每个训练窗口和每个评估回合都必须满足它；完整回合步数加上最后未完成回合步数必须等于 40 000。这样的检查能发现漏记跌落、把一次环境重置当作额外转移，或者在最后一个回合偷偷超预算等错误。它无法证明真实任务建模正确，仍需要读者审视环境定义。

## 12.6 把小实验的诊断方法带到深度强化学习

开始训练大网络前，应先确认一个小环境能区分机制。本章暴露的失败提示了合理的下一步：在保持交互预算的条件下，分别改变步长递减、探索退火和评估策略，检验循环是否减少；预先保留同一组种子与额外测试种子，避免只修好已经看见的失败。这样的消融尚未在本次运行中执行。

深度价值学习首先记录 Q、target 与 TD error 的分布，再看 target network、replay age 与状态动作覆盖。策略梯度除回报外，记录未标准化优势尺度、entropy、近似 KL、clip fraction、value loss 和终止原因。若 Q 值上涨而回报不变，应先核对目标、奖励尺度及终止语义；若熵迅速变小，应检查动作分布与探索；若多个种子只有个别成功，应先调查失败轨迹，再把平均曲线画得更平滑。

### 12.6.1 尚未执行的 PPO 与 SAC 扩展方案

本次**未执行 PPO、SAC 或其他深度算法的训练**。随书短轨迹验证的是其目标构件，不是神经网络优化效果。读者若扩展为新的实测，应另存日志与配置，不能把方案中的预算或预期曲线写成本书已有结果。

PPO 可先在低维离散控制任务实现无 baseline 的 REINFORCE，再依次加入 baseline、GAE、优势标准化和 clipping。每次只引入一个机制，固定总环境转移和评估协议；记录新旧概率比、策略 KL、clip fraction、value target 与真实回报。明确 rollout 的时间边界和最终观测，保存行为 log-probability；一旦未裁剪与裁剪目标的数值检查不通过，应在训练之前停止定位。

SAC 可在低维连续任务先核验动作范围、tanh 变换的 log-probability、双 Q 最小值、温度梯度和 replay 的终止标志，再比较固定温度与自动温度。其预算要同时报告环境转移和梯度更新次数，因为更高 replay ratio 可能在相同交互下消耗更多计算。冻结策略的确定性与随机性评估分别记录；若用外部限时，和本章一样保留未观测，不把 reset observation 放进 TD 目标。

涉及离线数据、语言模型反馈或智能体工具调用时，还需记录数据支持、奖励模型或验证器版本、工具失败、token 数、标注费与墙钟。目标是让读者能够从某个结论追到具体指标，再追到运行和原始记录。没有这一链条，更多方法名称和更大的训练预算都不会使结论更可靠。

## 练习与核验提示

1. 用四状态 MDP 求使 via 与 direct 同值的折扣因子。提示：解 $3\gamma^2 + \gamma - 2 = 0$，并只保留 $[0, 1)$ 中的根。
2. 将截断案例的末观测价值改为零，再把 reset observation 的价值改成 1 000。哪些结果应该改变？提示：只允许前者影响当前回合的 bootstrap。
3. 从原始 CSV 重算种子 5 的 greedy 回报，解释为什么跌落数为零却只有 $-500$。不要删掉该种子再报告总体均值。
4. 设计一个区分“探索撤除”与“常数步长估计波动”的实验矩阵，保持训练交互预算一致。提示：训练配置和评估配置应成为两个独立轴；此题要求方案，现有日志不能代替新实验。

# 第 13 章　从算法选择到可证伪的研究计划

前面已经给出算法的目标、数据依赖和失败边界。实际项目仍要回答三个顺序问题：在预算内能取得什么证据，哪种方法的假设最贴近环境，哪一个最小实验能区分候选解释。本章整合算法选择、训练组织和研究路线。表格中的方法是起点，并不构成跨任务性能排名；后半章的选题与十二周安排均为**尚未执行的学习或研究方案**。本次实际运行的范围只包括第 12 章记录的最小实验。

## 13.1　先由环境与预算缩小方法范围

先写状态与信息、动作、奖励与成本、终止、数据来源和评测任务。如果一项决策只影响当前收益，且不改变后续状态，bandit 或监督学习可能已经足够；若环境模型已知且状态动作可枚举，动态规划是应先建立的参考解。这些判断比先选一个复杂优化器更能减少不确定性。

| 问题结构与资源 | 可先建立的基线 | 选择依据与先验证的边界 |
|---|---|---|
| 有限、可枚举且模型已知 | 策略评估、价值迭代 | 建立解析或高精度参照；若不能枚举，需说明近似发生在哪里。 |
| 离散动作、在线交互可行 | 表格方法或 Double DQN；可并行时比较 PPO | Q 的动作最大化必须可计算；回放提高数据复用，on-policy 更新则依赖持续取得新轨迹。 |
| 连续动作、交互昂贵 | 随机/示范基线，再比较 SAC 与 TD3 | 回放可反复使用真实数据；仍要检验 critic 外推、探索覆盖、动作边界和温度等具体机制。 |
| 环境交互便宜、并行充足 | PPO 与简单策略基线 | 吞吐可能补偿较低的数据复用；应同时比较环境步数和墙钟，而非预设某一项更好。 |
| 模型可学且真实试验有限 | Dyna、短模型 rollout 或 MPC | 分别核算模型训练和规划开销；用真实留出轨迹检查多步误差，避免策略利用模型漏洞。 |
| 只有固定日志 | BC，再比较 IQL/CQL 与 OPE | 不存在靠在线探索补覆盖的机会；目标动作的数据支持和评价可识别性先于算法调参。 |
| 稀疏目标奖励、目标可重标记 | 简单探索，满足条件时加入 HER | 先确认目标不改变动力学、奖励可重算与终止一致；内在奖励需分开记录外在目标表现。 |
| 局部信息的合作智能体 | 独立学习，再按动作结构选 MAPPO/QMIX | 训练全局信息是否可用、执行通信是否允许，决定 CTDE 能否成立；参数共享也要检查身份信息。 |
| 有约束或稀有灾难 | 可审计的保守基线，再比较约束或风险目标 | 明确期望成本、逐步违约和尾部风险的区别；优化器名称不能替代安全证据。 |
| 偏好或可验证语言任务 | SFT/离线偏好基线，再考虑在线反馈学习 | 需要新策略数据时才引入相应闭环；同预算比较反馈质量、生成量、验证成本与最终成功。 |

“交互少”还需给出单位。一个像素环境的一步、一个完整回答和一次工具事务，成本并不相同。可预先设定环境交互上限 $N_{\text{env}}$、学习器更新上限 $N_{\text{grad}}$、墙钟上限 $T_{\text{wall}}$ 和验证预算 $N_{\text{eval}}$，记录先触及哪个上限。算法 A 在相同步数时更好、在相同时间时更差，是可以同时成立的结果。在线环境、模型生成数据和人工标注应分别计数，不能合并为一个没有成本含义的“样本数”。

## 13.2 把训练循环拆成可诊断的四个边界

一个可维护的训练程序应分清采集、估计、更新和评估。对任一算法，首先让一条 transition 的含义可以从日志还原：观测是否已经归一化，动作是变换前还是变换后，奖励是原始值还是训练值，下一观测是否来自 reset，终止属于任务还是采样器。若这些字段不清楚，再详细的网络配置也不足以复现实验。

**Listing 13.1** 通用训练组织示意；算法目标由前文定义

```
config = freeze_config()
for step in range(config.environment_budget):
    action, policy_info = behavior_policy(observation)
    transition = env.step(action)
    # Keep the transition's final observation before an auto-reset.
    storage.add(observation, action, transition, policy_info)
    observation = next_interaction_observation(transition)

    if update_is_due(storage, step):
        batch = sample_according_to_algorithm(storage)
        targets = construct_targets(batch)  # includes boundary masks
        update_parameters(batch, targets)
        update_targets_or_policy_version()

    if evaluation_is_due(step):
        evaluate_separate_environments(frozen_policy_snapshot())
        save_config_logs_and_checkpoint()
```

这段伪代码只规定组织边界，不能直接替代 PPO 或 SAC 实现。PPO 缓冲区保存采样策略版本、旧 log-prob、价值和两个边界掩码；优势和目标在当前更新阶段固定，ratio 对同一已采样动作计算。多轮优化不重新把当前策略概率写成“旧概率”。SAC 的 replay 则保存真实终止、截断及最终下一观测，critic 目标停止梯度，actor 优化时保留动作到 Q 的梯度，但不把这次反向传播当成 critic 的参数更新。温度优化对 log-prob 的梯度路径应按所用目标固定，target critic 再按声明频率更新。这些要求分别对应第 3、4 章的目标式。

评估环境不参与训练数据更新，评估时冻结预处理统计量。确定性均值动作与随机策略可能给出不同结果，应声明采用哪一种。checkpoint 除权重外还应保存优化器、随机数状态、归一化统计量、目标网络和采样步数；恢复训练时检查环境状态是否也能恢复，不能恢复则如实记录新的回合边界。

## 13.3 调参应追踪机制，而非背诵默认数字

折扣由任务目标和有效时域决定，不能只为提高分数改变；奖励缩放会改变 critic 量级，也可能改变固定熵系数下的相对目标。GAE 的 $\lambda$、多步回报长度和模型 rollout 长度都涉及信息传播与估计误差的权衡，但它们改变的是不同目标，不能用同一条“越长越好”的经验概括。学习率、批量、更新频率和 replay ratio 共同决定每份数据被使用多少次。比较时保留环境步数、梯度步数和策略版本，才能判断收益来自机制还是额外优化。

| 观察到的现象 | 需要区分的解释 | 一个最小诊断 |
| --- | --- | --- |
| critic 数值持续增大 | 奖励尺度变了、终止掩码错了，或 bootstrap 外推失控 | 在固定小批量核对手算目标；同时画原始奖励、目标值和 Q 分位数。 |

| 观察到的现象 | 需要区分的解释 | 一个最小诊断 |
| --- | --- | --- |
| PPO 更新后回报骤降 | 大策略漂移、优势符号/比率错误，或评价噪声 | 固定 rollout 检查 ratio 初值为 1；逐轮记录 KL、clip fraction 和未归一化优势。 |
| SAC 温度或动作饱和 | 熵目标不合适、Jacobian 漏项，或动作尺度不一致 | 用已知高斯样本核对变换密度；对照环境实际收到的动作范围。 |
| 模型中收益高、真实收益低 | 模型误差被策略利用，或训练/部署输入不一致 | 固定同一起点与动作序列，比真实和模型多步奖励、终止与状态误差。 |
| 离线训练损失低、部署差 | 数据外动作高估、覆盖不足，或评价脚本不一致 | 对行为动作和策略动作分别统计 Q；给出 BC 基线与支持不足反例。 |
| 平均回报提高、失败更多 | 长尾分布或约束被平均掩盖 | 展示独立训练点、失败率和尾部回报，并检查结论是否由少数 seed 决定。 |

每次诊断只需先改变一个足以区分解释的因素。若固定输入仍不能复现一次更新，应先修复数据与梯度路径；若这些核对通过，再考虑增加任务数或训练预算。调参预算是比较协议的一部分：给新方法更多搜索，再只比较最好一次运行，测到的是方法加搜索流程的组合。

## 13.4 把开放问题写成可以失败的研究设计

有用的研究问题应指出哪项假设可能错误，并预先定义什么观察会削弱它。以下研究方案围绕前文讨论的反馈、模型、离线支持和迁移机制展开。它们尚未执行；资源数值由实际项目的先导运行决定，正式比较前应固定预算和停止规则。

### 13.4.1 验证器覆盖是否比优化强度更限制收益

假设：在一个具有可控隐藏测试的代码或结构化任务中，增加训练验证器覆盖，比在固定验证器上增加策略更新，更能提高独立测试成功率。固定初始模型、训练任务和总生成预算，设置“增加验证覆盖”与“增加优化轮次”两个因素，保留相同成本的监督或重采样基线。训练验证器与最终隐藏测试独立；分别报告训练奖励、隐藏成功、长度、采样次数和验证耗时。若训练奖励提高而隐藏成功不变，证据支持反馈被利用；若在覆盖变化后收益仍仅随计算量变化，原假设受到削弱。不能将隐藏测试失败直接归为优化失败，应先人工审查代表性反例。

### 13.4.2 模型分歧能否预测策略将遇到的误差

假设：集成模型的分歧能为当前候选策略识别不可靠 rollout 区域。先在可重复采样的有限环境或小型模拟器中固定真实数据预算，训练同容量模型，对留出起点和预先固定的动作序列比较分歧、多步奖励误差和终止误差。随后只改变 rollout 长度或惩罚规则，比较真实环境回报和模型训练/规划时间。应包含随机惩罚或固定短 rollout 的基线，以排除仅减少模型使用量的效果。若高分歧不能区分高误差，或校准关系在策略更新后消失，则不能用这次分歧作为可靠性证据，即使某条回报曲线偶然提高。

### 13.4.3 固定日志下的改进是否来自可支持的动作

假设：保守估值在特定覆盖不足下减少错误改进。在小 MDP 中控制日志对某些动作的覆盖，保持数据量、行为策略质量和评估预算可追踪。对照 BC、一个普通价值学习基线与一种保守方法，用

已知模型的真实策略值核对 OPE，分别画覆盖、估计误差和策略回报。若惩罚只是在充分覆盖区间压制改进，应报告代价；若完全未观测动作也被稳定判为高收益，则应检查是否泄漏了模型信息。这类可控实验不能代替真实日志评测，但能独立检查方法所依赖的机制。

### 13.4.4 测试适应的收益究竟由什么预算产生

假设：某种表示或元学习初始化能在固定新任务交互预算下加快适应。按第 10 章切分训练、验证和测试任务，比较从头训练、普通预训练与适应方法，分别给出 $B = 0$ 和多个预先选定的 $B > 0$ 点。统一测试任务信息、参数更新次数和适应时间，报告适应阶段累计回报。若优势只在额外数据或更长计算后出现，结论应缩小为该预算下的质量改进，不能称为更快适应；若只在验证任务成立，应检查任务选择过拟合。

## 13.5 十二周路线：每周留下一个可以核对的对象

这份学习方案可按验收对象调整进度；完成本周验收后再进入下一步。CPU 路线优先使用小表格环境和短轨迹，拥有 GPU 或集群时再把已核对的机制扩展到神经网络。深度控制、偏好学习和多智能体训练所需环境、许可证及计算预算应单独准备。

| 周次 | 主线 | 可核对的交付物与进入下一步的依据 |
|---|---|---|
| 1 | MDP、回报与 Bellman | 重算四状态例题；改变折扣和终止后解释价值变化，保留独立线性方程参照。 |
| 2 | MC、TD 与表格控制 | 重跑风险路径实验；从原始日志重建分布，解释 SARSA 和 Q-learning 的目标差异。 |
| 3 | 策略梯度与 baseline | 对短轨迹核对 reward-to-go 和梯度符号；在小环境比较有无 baseline 的估计方差。 |
| 4 | GAE、PPO 与边界 | 核对两个掩码、初始 ratio 和 clipping 符号；若训练，记录 KL 与全部 seed。 |
| 5 | DQN 与离散控制 | 在可计算参考值的环境比较普通与 Double 目标，确认动作选择和评估网络分离。 |
| 6 | 连续动作与 SAC/TD3 | 先核对动作变换密度与梯度路径；再按预算运行连续控制，分开记录步数和墙钟。 |
| 7 | 探索、塑形与示范 | 给一个稀疏任务设置随机/BC 基线；构造终端势能残余反例或检验 HER 的适用条件。 |
| 8 | 模型与规划 | 在留出动作序列上画多步误差；比较短模型 rollout 与真实交互，记录规划成本。 |
| 9 | 固定日志与 OPE | 控制覆盖并核对真实策略值、IS/FQE 或 DR；在选择策略前固定数据切分。 |
| 10 | 约束或多智能体 | 二选一深入：定义成本单位与违约，或声明局部信息/通信；避免只做方法名巡览。 |
| 11 | 偏好与可验证反馈 | 建立一个小型反馈任务，冻结独立评测；比较训练分数与真实成功并审查奖励漏洞。 |
| 12 | 独立研究计划 | 写明一个假设、一个反证条件、预算匹配对照和分析脚本；先导实验通过后再扩展。 |

每次实验写明观察、机制解释、反证条件和下一次最小实验，并归档代码、配置、数据与环境版本、种子、原始结果及失败记录。排除错误假设本身也是进展，不能只保存最好曲线。

# 附录 A 推导补充与练习提示

本附录把正文中容易混用的恒等式放回同一套条件：有限状态动作、有界奖励、固定初始分布 $d_0$ 和 $0 \le \gamma < 1$。真终止后采用零奖励吸收扩展。有限时域 $\gamma = 1$ 的问题可以后向求解，但下述含 $1/(1-\gamma)$ 的结论不适用。

## A.1 从 Bellman residual 到价值误差

第 1 章证明了 $\mathcal{T}^\pi$ 与 $\mathcal{T}^*$ 的 $\gamma$ 收缩性。设 $\mathcal{T}$ 是其中任一个算子，$V_\star$ 是其不动点。对任意候选 $V$，三角不等式给出

$$\begin{aligned}\|V - V_\star\|_\infty &\le \|V - \mathcal{T}V\|_\infty + \|\mathcal{T}V - \mathcal{T}V_\star\|_\infty \\ &\le \|V - \mathcal{T}V\|_\infty + \gamma\|V - V_\star\|_\infty.\end{aligned}$$

移项便得到

$$\|V - V_\star\|_\infty \le \frac{\|V - \mathcal{T}V\|_\infty}{1-\gamma}.$$

它检查的是全状态上的 Bellman residual。在样本回归中，经验平均损失很小仍可能漏掉没有数据的状态；在函数逼近中，投影也未必保持这个范数下的收缩。因此不能把网络的一项训练损失直接代入该界。

还可以把估值误差连接到贪心策略。若 $\|V - V^*\|_\infty \le \varepsilon$，并令 $\pi_V$ 对 $V$ 贪心，则

$$\begin{aligned}\|V^* - V^{\pi_V}\|_\infty &\le \|\mathcal{T}^*V^* - \mathcal{T}^*V\|_\infty + \|\mathcal{T}^{\pi_V}V - \mathcal{T}^{\pi_V}V^*\|_\infty \\ &\quad + \|\mathcal{T}^{\pi_V}V^* - \mathcal{T}^{\pi_V}V^{\pi_V}\|_\infty \\ &\le 2\gamma\varepsilon + \gamma\|V^* - V^{\pi_V}\|_\infty.\end{aligned}$$

故贪心策略的损失至多为 $2\gamma\varepsilon/(1-\gamma)$。这里最大值计算精确、动作全覆盖，且 $V$ 的误差是在所有状态上有界的；缺少这些条件，公式不能充当部署证书。

## A.2 性能差异引理的望远镜证明

固定两个策略 $\pi, \pi'$。在新策略 $\pi'$ 的轨迹上，用旧价值构造

$$\delta_t^\pi = r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t).$$

给定 $(s_t, a_t)$，其期望为 $A^\pi(s_t, a_t)$。将前 $H$ 步按 $\gamma^t$ 加权后，价值项相消：

$$\sum_{t=0}^{H-1} \gamma^t \delta_t^\pi = \sum_{t=0}^{H-1} \gamma^t r_t - V^\pi(s_0) + \gamma^H V^\pi(s_H).$$

$V^\pi$ 有界，故最后一项在 $H \to \infty$ 时趋零。取期望并使用相同的 $d_0$，有

$$\begin{aligned}J(\pi') - J(\pi) &= \sum_{t\ge 0} \gamma^t \mathbb{E}_{\pi'}[A^\pi(s_t, a_t)] \\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\gamma^{\pi'}, a\sim\pi'}[A^\pi(s,a)].\end{aligned}$$

这是精确等式。换成旧访问分布 $d_\gamma^\pi$ 才产生 surrogate 近似；对旧动作乘 $\pi'/\pi$ 只换了条件动作分布，没有换整条轨迹的状态分布。若旧策略对新动作没有支持，该动作比率本身也不可用。

四状态问题提供一个核对：旧策略绕行概率为 $p$，新策略为 $p'$，则

$$J(p') - J(p) = (p' - p)(\gamma + 3\gamma^2 - 2).$$

在这个问题中，只有初始状态有决策，其访问质量始终为 $1-\gamma$，所以替换状态分布不会在这里造成误差。这是小例子的特殊性，不能推广到行动会改变后续决策机会的一般任务。

## A.3 KL 正则最优分布与支持限制

设动作集合有限、$q(a) > 0$，$\beta > 0$，价值 $Q(a)$ 有限。优化

$$F(p) = \sum_a p(a)Q(a) - \beta \sum_a p(a) \log \frac{p(a)}{q(a)}$$

并约束 $\sum_a p(a) = 1$。对拉格朗日函数求导得

$$Q(a) - \beta \left( \log \frac{p(a)}{q(a)} + 1 \right) + \eta = 0.$$

归一化后

$$p^*(a) = \frac{q(a)e^{Q(a)/\beta}}{Z}, \qquad Z = \sum_a q(a)e^{Q(a)/\beta}.$$

代回目标还能得到更直接的最优性证明：

$$F(p) = \beta \log Z - \beta D_{\mathrm{KL}}(p \| p^*).$$

KL 非负，且等于零当且仅当分布相同，因此 $p^*$ 是唯一最优分布。若参考 $q(a) = 0$，有限 KL 要求 $p(a) = 0$；公式只在参考的支持集上使用。连续分布还需要基测度与有限分配函数。神经网络参数族未必包含 $p^*$，样本优化也未必找到它。

回答级 RLHF 取 $Q(a) = r(x, y)$、$q = \pi_{\mathrm{ref}}$。奖励差中的 $\beta \log Z(x)$ 在同提示偏好比较中消掉，才得到第 11 章的 DPO 形式。这个推导并没有把错误偏好、数据缺口或优化误差变没。

## A.4 有限片段的 GAE 与 $\lambda$ 回报

先考虑从 $t$ 到 $K-1$ 的连续片段，中间没有 reset。冻结同一价值函数，尾部的 $V(s_K)$ 在真终止时设零，在外部截断时保留。令 $\delta_j = r_j + \gamma V(s_{j+1}) - V(s_j)$。有限和展开得到

$$V(s_t) + \sum_{\ell=0}^{K-t-1} (\gamma\lambda)^\ell \delta_{t+\ell} = (1-\lambda) \sum_{n=1}^{K-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{K-t-1} G_t^{(K-t)}.$$

内部价值项的系数通过相邻 residual 合并，末项吸收全部剩余权重。这就是前向有限 $\lambda$ 回报与 GAE target 的联系；$\lambda = 1$ 时右侧只剩最长回报，若该片段没有真正终止，仍包含尾部估值。

实际数组把多个回合连在一起，必须使用

$$\delta_t = r_t + \gamma b_t V(s_{t+1}) - V(s_t), \qquad \hat{A}_t = \delta_t + \gamma\lambda c_t \hat{A}_{t+1}.$$

外部限时/reset 边界的 $b_t = 1, c_t = 0$；无下一条 residual 的普通批末也取 $c_t = 0$。在第 3 章的短轨迹中，若截断于第二次转移后，$\delta_1 = 0.02$，故 $\hat{A}_0 = 0.65 + 0.72 \times 0.02 = 0.6644$。把下一回合的 residual 改成任意大值，这个结果仍应不变。随书验证程序专门测试了这项不变量。

## A.5 练习解题提示：从数字回到条件

第 1 章取 $p = 1/2, \gamma = 0.5$ 时，固定策略价值为 $2 + \frac{1}{2}(0.5 + 3 \times 0.25 - 2) = 1.625$。最优动作排序在 $\gamma = 2/3$ 处交换；$\gamma = 1$ 的有限无环问题可以精确求值，但不是折扣收缩定理的案例。

第 2 章从零估计做 $\lambda = 1/2$ 混合，得到目标 1.0575；若步长为 1/2，更新后 $V(s_0) = 0.52875$。这是沿一条绕行轨迹的单次更新，既不是参考策略期望值，也不是收敛结果。

第 3 章的 score 检查在 $p = 1/2$ 处给出 0.3325。加入准确 baseline 后，两条路径的优势分别为 $0.665, -0.665$，score 为 $1/2, -1/2$，期望仍为 $\frac{1}{2}(0.665)(\frac{1}{2}) + \frac{1}{2}(-0.665)(-\frac{1}{2}) = 0.3325$。

第 5 章的塑形只要满足终端势能为零，便给每条轨迹减去同一个初始势能。若 $\Phi(s_0) = 2$，$\gamma = 0.9$ 时两条完整轨迹的塑形回报为 0 与 1.33，动作排序不变。若任意保留非零终端势能而在终止时停止记分，会留下与回合长度相关的 $\gamma^T \Phi(s_T)$，保证随之失效。

第 7 章的支持反例是识别问题：两个环境在日志所有状态动作上相同，却在未执行动作上不同。只用同一日志的任何估计器都无法区分两者；增加网络大小不会提供缺失的观测。需要新数据、可检验的结构模型，或承认结论仅对覆盖区域成立。

第 9 章的三类碰撞要求应分别报告。首次碰撞指示的期望是回合事故概率；折扣碰撞次数还受事故时刻和重复次数影响；逐步绝不碰撞是更强的可达性要求。一个拉格朗日乘子不能在没有模型和可行性条件时把三者变成同一指标。

最后，书中的算法和方案都应能回答一个反向问题：什么观测会表明当前解释不成立？算例可以核对公式，日志可以核对一次运行；对未知部署环境的可靠性，仍需另外的证据。

# 附录 B 实现边界与报告核对表

本附录用于阅读实现和准备实验报告。它不能代替正文中的条件，也不把某个算法名称当作通用推荐。先写清任务，再沿数据路径检查每一个目标。

## B.1 转移进入缓冲区前

每条转移至少保存状态或可用历史、执行动作、原始奖励、训练奖励、真正的后继观测、真终止与截断标记。自动 reset 接口还需保存 final observation；循环策略需另存序列起点与隐状态边界。若有动作缩放、过滤器或人工接管，建议动作和实际执行动作必须分开。

| 检查点 | 正常语义 | 能暴露错误的边界 |
| --- | --- | --- |
| 奖励时序 | $r_t$ 属于当前 $s_t, a_t \to s_{t+1}$ | 只在一次指定转移给非零奖励，检查是否偏移一格 |
| 真终止 | bootstrap 系数为零 | 任意改变终端价值，当前目标保持不变 |
| 外部时间限制 | 保留 final observation 的价值 | 把 reset 状态价值设为极大，旧回合目标不变 |
| GAE 递推 | 只接同回合下一条 residual | 改下一回合奖励，旧回合优势不变 |
| 动作变换 | 密度对应实际执行动作 | 接近上下界时 log-prob 有限，检查缩放 Jacobian |
| 回放抽样 | 记录采样概率和更新比 | 过度抽样稀有样本后检查权重及有效样本量 |

## B.2 不同损失允许哪些梯度

| 更新 | 需要梯度的量 | 固定量及常见错误 |
| --- | --- | --- |
| DQN/连续 critic | 当前 $Q_\phi(s,a)$ | target 的动作、后继 Q 和温度均冻结；不能让目标跟着预测一起移动 |
| REINFORCE/PPO | 当前动作 log-prob | 回报、优势、旧概率和价值 target 固定；不能每 epoch 重算分母 |
| DDPG/TD3 actor | $Q_\phi(s,\mu_\theta(s))$ 对动作的导数 | critic 参数不更新，但不能 detach 动作后再声称有 actor 梯度 |
| SAC actor | 重参数化动作及其 log-prob | $\alpha$ 固定，critic 参数固定；保留 Q 对动作的梯度 |
| SAC 温度 | $\log\alpha$ 或 $\alpha$ | 当前 log-prob/熵目标固定；区分两种参数化的梯度尺度 |
| DPO | 新策略回答 token 的 log-prob 和 | 参考模型固定；提示、padding 排除，EOS 与截断协议一致 |

共享编码器可被多种损失更新，但每条损失的求导路径仍应与其估计器一致。小型有限差分检查可以验证某个梯度的数值与方向；它不能证明估计器对应真实任务，后者还需目标和采样条件。

## B.3 按信息与预算选择起点

| 环境条件 | 可建立的比较基线 | 先保存的诊断 |
| --- | --- | --- |
| 离散在线、动作可枚举 | 表格基线或 DQN/Double DQN，另比较 PPO | Q/target 分布、覆盖、探索与真终止 |
| 连续在线、回放可用 | TD3 与 SAC | 双 Q、动作饱和、温度、log-prob、每转移更新次数 |
| 目标奖励可重算 | 原算法与 HER | 重标记比例、目标动力学条件、真实成功率 |
| 真实交互昂贵、有可学模型 | 无模型基线、短 rollout 或 MPC | 真实/模型步数分别计、闭环误差和规划墙钟 |
| 只有固定日志 | BC，覆盖允许时比较 IQL/CQL | 支持缺口、行为概率、OPE 区间及数据切分 |
| 合作多智能体 | 独立策略与 CTDE 基线 | 训练特权信息、执行通信、队友/对手分布 |
| 高风险决策 | 约束/风险目标与可解释备份 | 回报、违约概率、尾部风险、介入率和假设失效 |
| 偏好与验证器反馈 | 监督基线、DPO 或明确 RL 目标 | 独立验证器、奖励误差、长度、token 和工具成本 |

## B.4 一次运行应留下什么

任务说明写出状态信息、动作单位、奖励各项、终止原因和预算含义；配置保存全部随机源、环境版本、代码 hash、依赖、硬件和并行度。交互预算之外还要记录梯度更新次数、模拟数据、评估转移、token、标注或工具调用成本。日志同时保留完整种子和失败运行，避免只留下最好检查点。

报告按独立训练种子或独立任务集汇总；同一策略的重复评估回合不能冒充独立训练。画曲线时写明横轴、窗口、平滑、阴影定义；给终局指标同时列成功、超时与严重失败。挑选配置使用验证集，最终测试集不再充当调参反馈。多个测试任务共享设备、用户或地图族时，应保留相关结构并按正确的群组重采样。

本书实测具备逐种子原始回合、固定训练转移预算、独立评估及账目恒等式检查。读取记录的入口是第 12 章及配套实验 README。其余深度训练和研究矩阵均为明确标记的未运行方案；把方案变成结果时，应新增一次完整运行记录和相应局限说明。

# 参考文献


[1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html (visited on 10/22/2026).

[2] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev. "Time Limits in Reinforcement Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4045–4054. URL: https://proceedings.mlr.press/v80/pardo18a.html (visited on 10/22/2026).

[3] C. J. C. H. Watkins and P. Dayan. "Q-learning". In: *Machine Learning* 8.3–4 (May 1992), pp. 279–292. DOI: 10.1007/bf00992698. URL: https://doi.org/10.1007/bf00992698 (visited on 10/22/2026).

[4] R. J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3–4 (May 1992), pp. 229–256. DOI: 10.1007/bf00992696. URL: https://doi.org/10.1007/bf00992696 (visited on 10/22/2026).

[5] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html (visited on 10/22/2026).

[6] V. R. Konda and J. N. Tsitsiklis. "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, 1999. URL: https://proceedings.neurips.cc/paper_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (visited on 10/22/2026).

[7] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. Version 6. June 8, 2015. arXiv: 1506.02438v6. URL: https://arxiv.org/abs/1506.02438v6 (visited on 10/22/2026). Version revised 2018-10-20.

[8] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. *Trust Region Policy Optimization*. Version 5. Feb. 19, 2015. arXiv: 1502.05477v5. URL: https://arxiv.org/abs/1502.05477v5 (visited on 10/22/2026). Version revised 2017-04-20.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. Version 2. July 20, 2017. arXiv: 1707.06347v2. URL: https://arxiv.org/abs/1707.06347v2 (visited on 10/22/2026). Version revised 2017-08-28.

[10] V. Mnih et al. *Playing Atari with Deep Reinforcement Learning*. Version 1. Dec. 19, 2013. arXiv: 1312.5602v1. URL: https://arxiv.org/abs/1312.5602v1 (visited on 10/22/2026).

[11] V. Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 25, 2015), pp. 529–533. DOI: 10.1038/nature14236. URL: https://doi.org/10.1038/nature14236 (visited on 10/22/2026).

[12] H. van Hasselt, A. Guez, and D. Silver. *Deep Reinforcement Learning with Double Q-learning*. Version 3. Sept. 22, 2015. arXiv: 1509.06461v3. URL: https://arxiv.org/abs/1509.06461v3 (visited on 10/22/2026). Version revised 2015-12-08.

[13] Z. Wang et al. *Dueling Network Architectures for Deep Reinforcement Learning*. Version 3. Nov. 20, 2015. arXiv: 1511.06581v3. URL: https://arxiv.org/abs/1511.06581v3 (visited on 10/22/2026). Version revised 2016-04-05.

[14] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. *Prioritized Experience Replay*. Version 4. Nov. 18, 2015. arXiv: 1511.05952v4. URL: https://arxiv.org/abs/1511.05952v4 (visited on 10/22/2026). Version revised 2016-02-25.

[15] M. G. Bellemare, W. Dabney, and R. Munos. *A Distributional Perspective on Reinforcement Learning*. Version 1. July 21, 2017. arXiv: 1707.06887v1. URL: https://arxiv.org/abs/1707.06887v1 (visited on 10/22/2026).

[16] M. Hessel et al. *Rainbow: Combining Improvements in Deep Reinforcement Learning*. Version 1. Oct. 6, 2017. arXiv: 1710.02298v1. URL: https://arxiv.org/abs/1710.02298v1 (visited on 10/22/2026).

[17] M. Fortunato et al. *Noisy Networks for Exploration*. Version 3. June 30, 2017. arXiv: 1706.10295v3. URL: https://arxiv.org/abs/1706.10295v3 (visited on 10/22/2026). Version revised 2019-07-09.

[18] T. P. Lillicrap et al. *Continuous control with deep reinforcement learning*. Version 6. Sept. 9, 2015. arXiv: 1509.02971v6. URL: https://arxiv.org/abs/1509.02971v6 (visited on 10/22/2026). Version revised 2019-07-05.

[19] S. Fujimoto, H. van Hoof, and D. Meger. *Addressing Function Approximation Error in Actor-Critic Methods*. Version 3. Feb. 26, 2018. arXiv: 1802.09477v3. URL: https://arxiv.org/abs/1802.09477v3 (visited on 10/22/2026). Version revised 2018-10-22.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. Version 2. Jan. 4, 2018. arXiv: 1801.01290v2. URL: https://arxiv.org/abs/1801.01290v2 (visited on 10/22/2026). Version revised 2018-08-08.

[21] T. Haarnoja et al. *Soft Actor-Critic Algorithms and Applications*. Version 2. Dec. 13, 2018. arXiv: 1812.05905v2. URL: https://arxiv.org/abs/1812.05905v2 (visited on 10/22/2026). Version revised 2019-01-29.

[22] V. Mnih et al. *Asynchronous Methods for Deep Reinforcement Learning*. Version 2. Feb. 4, 2016. arXiv: 1602.01783v2. URL: https://arxiv.org/abs/1602.01783v2 (visited on 10/22/2026). Version revised 2016-06-16.

[23] L. Espeholt et al. *IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures*. Version 3. Feb. 5, 2018. arXiv: 1802.01561v3. URL: https://arxiv.org/abs/1802.01561v3 (visited on 10/22/2026). Version revised 2018-06-28.

[24] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. *Curiosity-driven Exploration by Self-supervised Prediction*. Version 1. May 15, 2017. arXiv: 1705.05363v1. URL: https://arxiv.org/abs/1705.05363v1 (visited on 10/22/2026).

[25] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. *Exploration by Random Network Distillation*. Version 1. Oct. 30, 2018. arXiv: 1810.12894v1. URL: https://arxiv.org/abs/1810.12894v1 (visited on 10/22/2026).

[26] A. Y. Ng, D. Harada, and S. Russell. "Policy invariance under reward transformations: Theory and application to reward shaping". In: *Proceedings of the International Conference on Machine Learning (ICML-99)*. 1999. URL: https://people.eecs.berkeley.edu/~russell/papers/icml99-shaping.pdf (visited on 10/22/2026).

[27] M. Andrychowicz et al. *Hindsight Experience Replay*. Version 3. July 5, 2017. arXiv: 1707.01495v3. URL: https://arxiv.org/abs/1707.01495v3 (visited on 10/22/2026). Version revised 2018-02-23.

[28] R. S. Sutton, D. Precup, and S. Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artificial Intelligence* 112.1–2 (Aug. 1999), pp. 181–211. DOI: 10.1016/s0004-3702(99)00052-1. URL: https://doi.org/10.1016/s0004-3702(99)00052-1 (visited on 10/22/2026).

[29] P.-L. Bacon, J. Harb, and D. Precup. *The Option-Critic Architecture*. Version 2. Sept. 16, 2016. arXiv: 1609.05140v2. URL: https://arxiv.org/abs/1609.05140v2 (visited on 10/22/2026). Version revised 2016-12-03.

[30] S. Ross, G. Gordon, and D. Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. Proceedings of Machine Learning Research. PMLR, 2011, pp. 627–635. URL: https://proceedings.mlr.press/v15/ross11a.html (visited on 10/22/2026).

[31] J. Ho and S. Ermon. *Generative Adversarial Imitation Learning*. Version 1. June 10, 2016. arXiv: 1606.03476v1. URL: https://arxiv.org/abs/1606.03476v1 (visited on 10/22/2026).

[32] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. "Maximum Entropy Inverse Reinforcement Learning". In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. 2008, pp. 1433–1438. URL: https://cdn.aaai.org/AAAI/2008/AAAI08-227.pdf (visited on 10/22/2026).

[33] P. Christiano et al. *Deep reinforcement learning from human preferences*. Version 4. June 12, 2017. arXiv: 1706.03741v4. URL: https://arxiv.org/abs/1706.03741v4 (visited on 10/22/2026). Version revised 2023-02-17.

[34] R. S. Sutton. "Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming". In: *Proceedings of the Seventh International Conference on Machine Learning*. Morgan Kaufmann, 1990, pp. 216–224. URL: http://incompleteideas.net/papers/sutton-90.pdf (visited on 10/22/2026).

[35] K. Chua, R. Calandra, R. McAllister, and S. Levine. *Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models*. Version 2. May 30, 2018. arXiv: 1805.12114v2. URL: https://arxiv.org/abs/1805.12114v2 (visited on 10/22/2026). Version revised 2018-11-02.

[36] M. Janner, J. Fu, M. Zhang, and S. Levine. *When to Trust Your Model: Model-Based Policy Optimization*. Version 3. June 19, 2019. arXiv: 1906.08253v3. URL: https://arxiv.org/abs/1906.08253v3 (visited on 10/22/2026). Version revised 2021-11-29.

[37] T. Yu et al. *MOPO: Model-based Offline Policy Optimization*. Version 6. May 27, 2020. arXiv: 2005.13239v6. URL: https://arxiv.org/abs/2005.13239v6 (visited on 10/22/2026). Version revised 2020-11-22.

[38] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. *MOReL : Model-Based Offline Reinforcement Learning*. Version 3. May 12, 2020. arXiv: 2005.05951v3. URL: https://arxiv.org/abs/2005.05951v3 (visited on 10/22/2026). Version revised 2021-03-02.

[39] D. Ha and J. Schmidhuber. *World Models*. Version 4. Mar. 27, 2018. arXiv: 1803.10122v4. URL: https://arxiv.org/abs/1803.10122v4 (visited on 10/22/2026). Version revised 2018-05-09.

[40] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. *Dream to Control: Learning Behaviors by Latent Imagination*. Version 3. Dec. 3, 2019. arXiv: 1912.01603v3. URL: https://arxiv.org/abs/1912.01603v3 (visited on 10/22/2026). Version revised 2020-03-17.

[41] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. *Mastering Diverse Domains through World Models*. Version 2. Jan. 10, 2023. arXiv: 2301.04104v2. URL: https://arxiv.org/abs/2301.04104v2 (visited on 10/22/2026). Version revised 2024-04-17.

[42] D. Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. Version 1. Dec. 5, 2017. arXiv: 1712.01815v1. URL: https://arxiv.org/abs/1712.01815v1 (visited on 10/22/2026).

[43] J. Schrittwieser et al. *Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model*. Version 2. Nov. 19, 2019. arXiv: 1911.08265v2. URL: https://arxiv.org/abs/1911.08265v2 (visited on 10/22/2026). Version revised 2020-02-21.

[44] S. Levine, A. Kumar, G. Tucker, and J. Fu. *Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems*. Version 3. May 4, 2020. arXiv: 2005.01643v3. URL: https://arxiv.org/abs/2005.01643v3 (visited on 10/22/2026). Version revised 2020-11-01.

[45] S. Fujimoto, D. Meger, and D. Precup. *Off-Policy Deep Reinforcement Learning without Exploration*. Version 3. Dec. 7, 2018. arXiv: 1812.02900v3. URL: https://arxiv.org/abs/1812.02900v3 (visited on 10/22/2026). Version revised 2019-08-10.

[46] A. Kumar, J. Fu, G. Tucker, and S. Levine. *Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction*. Version 2. June 3, 2019. arXiv: 1906.00949v2. URL: https://arxiv.org/abs/1906.00949v2 (visited on 10/22/2026). Version revised 2019-11-25.

[47] A. Kumar, A. Zhou, G. Tucker, and S. Levine. *Conservative Q-Learning for Offline Reinforcement Learning*. Version 3. June 8, 2020. arXiv: 2006.04779v3. URL: https://arxiv.org/abs/2006.04779v3 (visited on 10/22/2026). Version revised 2020-08-19.

[48] I. Kostrikov, A. Nair, and S. Levine. *Offline Reinforcement Learning with Implicit Q-Learning*. Version 1. Oct. 12, 2021. arXiv: 2110.06169v1. URL: https://arxiv.org/abs/2110.06169v1 (visited on 10/22/2026).

[49] L. Chen et al. *Decision Transformer: Reinforcement Learning via Sequence Modeling*. Version 2. June 2, 2021. arXiv: 2106.01345v2. URL: https://arxiv.org/abs/2106.01345v2 (visited on 10/22/2026). Version revised 2021-06-24.

[50] M. Janner, Q. Li, and S. Levine. *Offline Reinforcement Learning as One Big Sequence Modeling Problem*. Version 4. June 3, 2021. arXiv: 2106.02039v4. URL: https://arxiv.org/abs/2106.02039v4 (visited on 10/22/2026). Version revised 2021-11-29.

[51] H. Le, C. Voloshin, and Y. Yue. "Batch Policy Learning under Constraints". In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 3703–3712. URL: https://proceedings.mlr.press/v97/le19a.html (visited on 10/22/2026).

[52] N. Jiang and L. Li. "Doubly Robust Off-policy Value Evaluation for Reinforcement Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR, 2016, pp. 652–661. URL: https://proceedings.mlr.press/v48/jiang16.html (visited on 10/22/2026).

[53] R. Lowe et al. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. Version 4. June 7, 2017. arXiv: 1706.02275v4. URL: https://arxiv.org/abs/1706.02275v4 (visited on 10/22/2026). Version revised 2020-03-14.

[54] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. *Counterfactual Multi-Agent Policy Gradients*. Version 3. May 24, 2017. arXiv: 1705.08926v3. URL: https://arxiv.org/abs/1705.08926v3 (visited on 10/22/2026). Version revised 2024-12-11.

[55] T. Rashid et al. *QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning*. Version 2. Mar. 30, 2018. arXiv: 1803.11485v2. URL: https://arxiv.org/abs/1803.11485v2 (visited on 10/22/2026). Version revised 2018-06-06.

[56] C. Yu et al. *The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games*. Version 4. Mar. 2, 2021. arXiv: 2103.01955v4. URL: https://arxiv.org/abs/2103.01955v4 (visited on 10/22/2026). Version revised 2022-11-04.

[57] J. Achiam, D. Held, A. Tamar, and P. Abbeel. *Constrained Policy Optimization*. Version 1. May 30, 2017. arXiv: 1705.10528v1. URL: https://arxiv.org/abs/1705.10528v1 (visited on 10/22/2026).

[58] R. T. Rockafellar and S. Uryasev. "Optimization of conditional value-at-risk". In: *The Journal of Risk* 2.3 (2000), pp. 21–41. DOI: 10.21314/jor.2000.038. URL: https://doi.org/10.21314/jor.2000.038 (visited on 10/22/2026).

[59] G. N. Iyengar. "Robust Dynamic Programming". In: *Mathematics of Operations Research* 30.2 (May 2005), pp. 257–280. DOI: 10.1287/moor.1040.0129. URL: https://doi.org/10.1287/moor.1040.0129 (visited on 10/22/2026).

[60] A. Barreto et al. *Successor Features for Transfer in Reinforcement Learning*. Version 2. June 16, 2016. arXiv: 1606.05312v2. URL: https://arxiv.org/abs/1606.05312v2 (visited on 10/22/2026). Version revised 2018-04-12.

[61] C. Finn, P. Abbeel, and S. Levine. *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. Version 3. Mar. 9, 2017. arXiv: 1703.03400v3. URL: https://arxiv.org/abs/1703.03400v3 (visited on 10/22/2026). Version revised 2017-07-18.

[62] Y. Duan et al. *$RL^2$: Fast Reinforcement Learning via Slow Reinforcement Learning*. Version 2. Nov. 9, 2016. arXiv: 1611.02779v2. URL: https://arxiv.org/abs/1611.02779v2 (visited on 10/22/2026). Version revised 2016-11-10.

[63] K. Rakelly, A. Zhou, D. Quillen, C. Finn, and S. Levine. *Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables*. Version 1. Mar. 19, 2019. arXiv: 1903.08254v1. URL: https://arxiv.org/abs/1903.08254v1 (visited on 10/22/2026).

[64] P. Henderson et al. *Deep Reinforcement Learning that Matters*. Version 3. Sept. 19, 2017. arXiv: 1709.06560v3. URL: https://arxiv.org/abs/1709.06560v3 (visited on 10/22/2026). Version revised 2019-01-30.

[65] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare. *Deep Reinforcement Learning at the Edge of the Statistical Precipice*. Version 4. Aug. 30, 2021. arXiv: 2108.13264v4. URL: https://arxiv.org/abs/2108.13264v4 (visited on 10/22/2026). Version revised 2022-01-05.

[66] N. Stiennon et al. *Learning to summarize from human feedback*. Version 3. Sept. 2, 2020. arXiv: 2009.01325v3. URL: https://arxiv.org/abs/2009.01325v3 (visited on 10/22/2026). Version revised 2022-02-15.

[67] L. Ouyang et al. *Training language models to follow instructions with human feedback*. Version 1. Mar. 4, 2022. arXiv: 2203.02155v1. URL: https://arxiv.org/abs/2203.02155v1 (visited on 10/22/2026).

[68] R. Rafailov et al. *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. Version 3. May 29, 2023. arXiv: 2305.18290v3. URL: https://arxiv.org/abs/2305.18290v3 (visited on 10/22/2026). Version revised 2024-07-29.

[69] Y. Bai et al. *Constitutional AI: Harmlessness from AI Feedback*. Version 1. Dec. 15, 2022. arXiv: 2212.08073v1. URL: https://arxiv.org/abs/2212.08073v1 (visited on 10/22/2026).

[70] H. Lightman et al. *Let's Verify Step by Step*. Version 1. May 31, 2023. arXiv: 2305.20050v1. URL: https://arxiv.org/abs/2305.20050v1 (visited on 10/22/2026).

[71] Z. Shao et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. Version 3. Feb. 5, 2024. arXiv: 2402.03300v3. URL: https://arxiv.org/abs/2402.03300v3 (visited on 10/22/2026). Version revised 2024-04-27.

[72] Z. Liu et al. *Understanding R1-Zero-Like Training: A Critical Perspective*. Version 2. Mar. 26, 2025. arXiv: 2503.20783v2. URL: https://arxiv.org/abs/2503.20783v2 (visited on 10/22/2026). Version revised 2025-10-06.

[73] Q. Yu et al. *DAPO: An Open-Source LLM Reinforcement Learning System at Scale*. Version 2. Mar. 18, 2025. arXiv: 2503.14476v2. URL: https://arxiv.org/abs/2503.14476v2 (visited on 10/22/2026). Version revised 2025-05-20.

[74] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. Version 2. Jan. 22, 2025. arXiv: 2501.12948v2. URL: https://arxiv.org/abs/2501.12948v2 (visited on 10/22/2026). Version revised 2026-01-04.

[75] K. Liu et al. *Reinforcement Learning Meets Large Language Models: A Survey of Advancements and Applications Across the LLM Lifecycle*. Version 1. Sept. 20, 2025. arXiv: 2509.16679v1. URL: https://arxiv.org/abs/2509.16679v1 (visited on 10/22/2026).

[76] R. Yu et al. *Reward Models in Deep Reinforcement Learning: A Survey*. Version 1. June 18, 2025. arXiv: 2506.15421v1. URL: https://arxiv.org/abs/2506.15421v1 (visited on 10/22/2026).