

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



Advanced RL Algorithms for Agents

Improving how agents learn from experience

Carnegie Mellon University
Language Technologies Institute

Graham Neubig
Language Technologies Institute

1

智能体的高级强化学习算法

改进智能体从经验中学习的方式

Graham Neubig · 语言技术研究所 · 卡内基梅隆大学

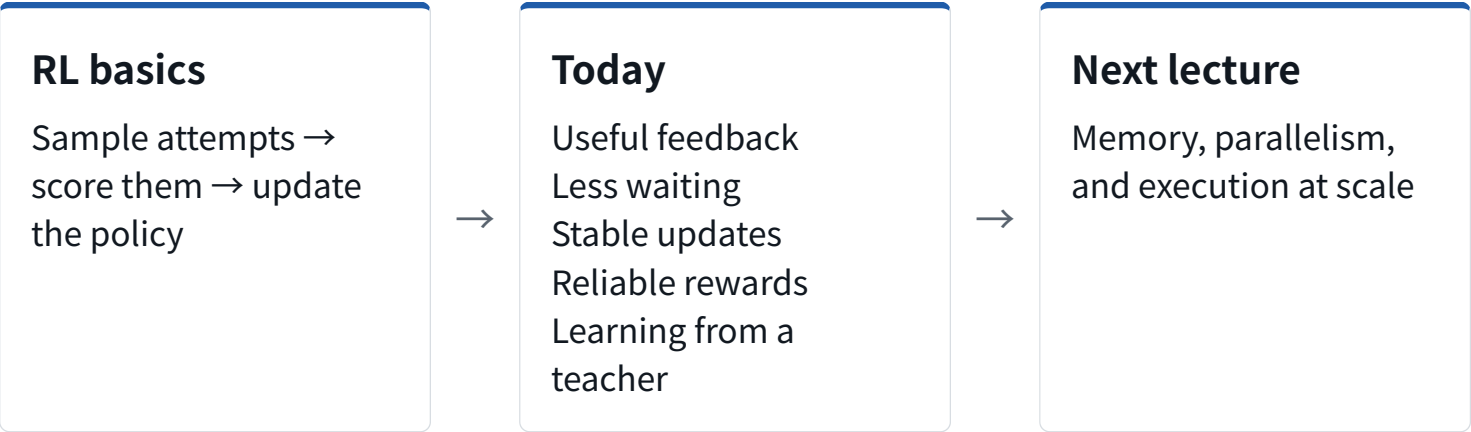
智能体的高级强化学习算法

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

RL lecture map



2

强化学习基础：采样尝试 → 为尝试打分 → 更新策略。

本讲：提供有用反馈、减少等待、稳定更新、获得可靠奖励、向教师模型学习。

下讲：记忆、并行化和大规模执行。

强化学习课程图谱

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

An RL task: fixing a bug

We'll train a coding agent to fix bugs, using tests to judge each attempt.

Prompt

Fix the retry setting: zero should disable retries, but the code returns three. Preserve the other cases.

Initial state: buggy code

```
def retry_count(config):  
    return config.get("retries") or 3
```

Verifier: tests

Input → expected output

Missing → 3	None → 3
0 → 0	2 → 2

[Bug-fix example · Lecture 6: Coding Agents \(slides\)](#)

3

我们将训练一个编码智能体修复错误，并用测试来评价每次尝试。

提示：修复重试设置：零应禁用重试，但当前代码返回三。保留其他情况。

初始状态：有缺陷的代码。

```
def retry_count(config):  
    return config.get("retries") or 3
```

验证器：测试。输入 → 预期输出：缺失 → 3；None → 3；0 → 0；2 → 2。

Bug-fix 示例 · 第6讲：编码智能体（幻灯片） ↗

译注 / 与原文翻译分开

代码与配置字段保留原样。

一个强化学习任务：修复错误

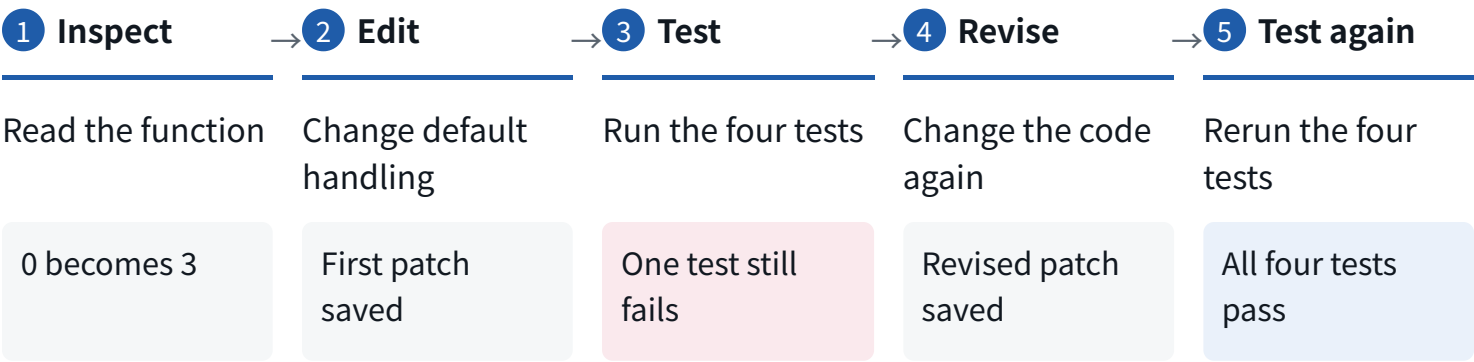
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A successful trajectory

A trajectory τ is one attempt: the agent’s actions and the observations they produce.



Final reward: $R(\tau) = 1$ if all four tests pass; 0 otherwise.

4

一条轨迹 τ 是一次尝试：智能体采取的动作及其产生的观测。

1 检查：读取函数；2 编辑：更改默认值处理；3 测试：运行四项测试；4 修改：再次更改代码；5 再次测试：重新运行四项测试。

过程：0 被变成 3；保存第一个补丁；仍有一项测试失败；保存修改后的补丁；四项测试全部通过。

最终奖励： $R(\tau) = 1$ 表示四项测试全部通过；否则为 0。

一条成功的轨迹

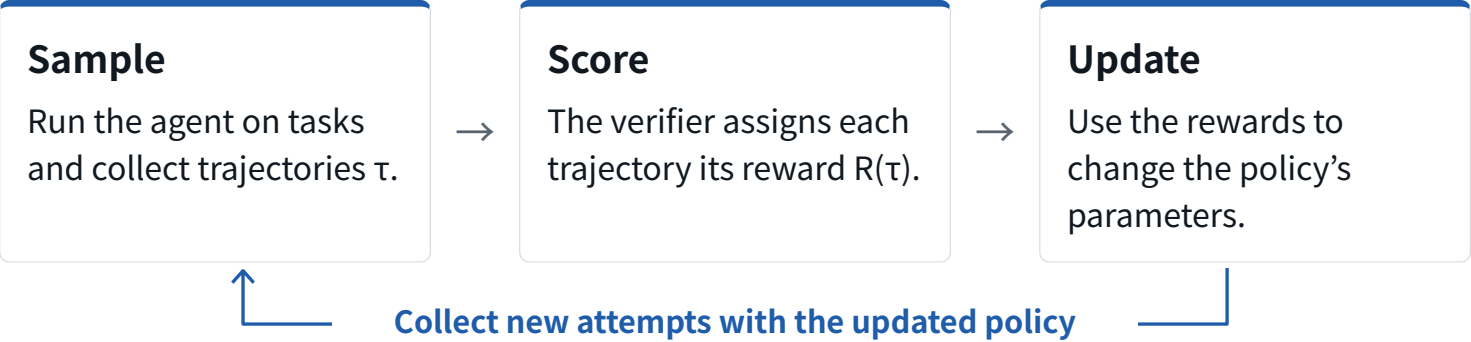
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

The RL training loop

Training repeats the loop to increase average reward over many attempts.



x is the task (initial state and prompt); θ is the policy’s parameters.
 $p_{\theta}(\tau \mid x)$ is the probability of trajectory τ given task x .

Goal $J(\theta)$: maximize average reward on task x

$$J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\cdot|x)}[R(\tau)]$$

5

训练反复执行该循环，以提高多次尝试的平均奖励。

采样：在任务上运行智能体并收集轨迹 τ 。→ 评分：验证器为每条轨迹分配奖励 $R(\tau)$ 。→ 更新：使用奖励改变策略参数。

使用更新后的策略收集新的尝试。

x 是任务（初始状态与提示）； θ 是策略参数。 $p_{\theta}(\tau \mid x)$ 是给定任务 x 时轨迹 τ 的概率。

目标 $J(\theta)$ ：最大化任务 x 上的平均奖励。
 $J(\theta) = \mathbb{E}_{\tau \sim p_{\theta}(\cdot|x)}[R(\tau)]$

译注 / 与原文翻译分开

变量与公式符号保留原样。

强化学习训练循环

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reward and the policy update

Plain REINFORCE uses the final reward to weight every action in an attempt.

Successful attempt

All four tests pass: $R(\tau) = 1$.

Increase the likelihood of its actions.

Failed attempt

At least one test fails: $R(\tau) = 0$.

This attempt contributes no update.

At step t , action a_t follows history h_t : the task, earlier actions, and observations. $\pi_\theta(a_t | h_t)$ is the policy’s probability of that action given the history.

Loss to minimize for one sampled trajectory

$$\mathcal{L}_{PG} = -R(\tau) \sum_t \log \pi_\theta(a_t | h_t)$$

How can a failed attempt still provide a learning signal?

6

基础 REINFORCE 使用最终奖励为一次尝试中的每个动作赋权。

成功尝试：四项测试全部通过， $R(\tau)=1$ ；提高该尝试中各动作的概率。

失败尝试：至少一项测试失败， $R(\tau)=0$ ；这次尝试不会产生更新。

在步骤 t ，动作 a_t 接续于历史 h_t ：任务、较早的动作和观测。 $\pi_\theta(a_t | h_t)$ 是策略在该历史下选择该动作的概率。

对单条采样轨迹要最小化的损失：
 $L_{PG} = -R(\tau) \sum_t \log \pi_\theta(a_t | h_t)$

失败的尝试怎样才能仍然提供学习信号？

译注 / 与原文翻译分开

公式符号保留原样。

奖励与策略更新

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Credit assignment and dense feedback

Which actions deserve credit?

7

信用分配与密集反馈。

哪些动作值得获得正向信用?

信用分配与密集反馈

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Comparing trajectories on the same task

Sample four attempts at the same bug-fix task, then compare their rewards.

Attempt	Final tests	Reward $R(\tau)$	Update weight: reward – baseline
1	All four pass	1	$1 - 0.25 = +0.75$
2	At least one fails	0	$0 - 0.25 = -0.25$
3	At least one fails	0	$0 - 0.25 = -0.25$
4	At least one fails	0	$0 - 0.25 = -0.25$

Use the group average as a baseline:
 $(1 + 0 + 0 + 0) / 4 = 0.25$.

Even failed attempts now contribute: their actions get a negative weight.

8

针对同一个修复错误任务采样四次尝试，然后比较奖励。

尝试1：四项测试全通过，奖励 $R(\tau)=1$ ，更新权重 = 奖励 – 基线 = $1 - 0.25 = +0.75$ 。

尝试2：至少一项测试失败，奖励为0，更新权重 = $0 - 0.25 = -0.25$ 。

尝试3：至少一项测试失败，奖励为0，更新权重 = $0 - 0.25 = -0.25$ 。

尝试4：至少一项测试失败，奖励为0，更新权重 = $0 - 0.25 = -0.25$ 。

使用组内平均值作为基线： $(1 + 0 + 0 + 0) / 4 = 0.25$ 。

现在失败的尝试也会产生贡献：其中的动作获得负权重。

比较同一任务上的多条轨迹

第 11 讲 · 智能体的高级强化学习算法

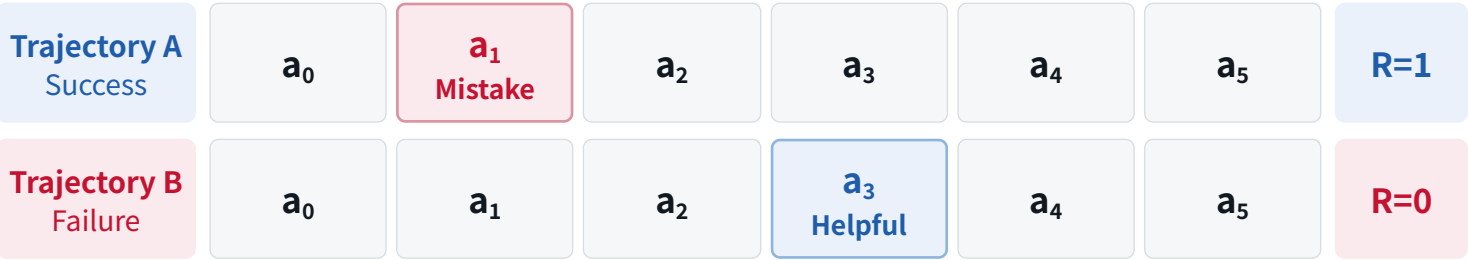
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Rewards for individual turns

A successful trajectory can contain a mistake; a failed trajectory can contain a helpful turn.

Two illustrative trajectories on the same task. Each action $a_0, \dots, a_5$ represents one agent turn.



The group weight is the same for every turn in a trajectory.

Turn-level rewards could tell us which turns to reinforce within each trajectory.

9

成功轨迹中也可能包含错误；失败轨迹中也可能包含有帮助的一轮。

图示为同一任务上的两条示例轨迹，每个动作 $a_0 \dots a_5$ 表示智能体的一轮操作。

轨迹 A：最终成功， $R=1$ ；错误出现在 a_1 。轨迹 B：最终失败， $R=0$ ；有帮助的一轮出现在 a_3 。

组内权重在一条轨迹的所有轮次上都相同。

轮次级奖励可以告诉我们：每条轨迹中哪些轮次值得加强。

为各轮分配奖励

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reward-to-go from one trajectory

With only a final reward, A receives zero at each step until the task succeeds.

History before	h_0	h_1	h_2	h_3	h_4	h_5
Action	$\mathbf{a_0}$	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$
Reward after	$r_1 = 0$	$r_2 = 0$	$r_3 = 0$	$r_4 = 0$	$r_5 = 0$	$r_6 = 1$

Starting just before a_3 , how much reward remains on this trajectory?

Add the three remaining rewards, counting each equally

$0 + 0 + 1 = 1$

This sum is the **reward-to-go R_3** : the rewards received after h_3 .

10

如果只有最终奖励，轨迹 A 在任务成功前的每一步奖励都为零。

历史： h_0 、 h_1 、 h_2 、 h_3 、 h_4 、 h_5 。动作： a_0 、 a_1 、 a_2 、 a_3 、 a_4 、 a_5 。奖励： $r_1=0$ 、 $r_2=0$ 、 $r_3=0$ 、 $r_4=0$ 、 $r_5=0$ 、 $r_6=1$ 。

从 a_3 之前开始，这条轨迹还剩多少奖励？

将剩余三项奖励相加，每项权重相同： $0 + 0 + 1 = 1$ 。

这个总和是后续回报 R_3 ：从 h_3 之后收到的奖励。

一条轨迹的后续回报

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Credit along two trajectories

B receives zero reward at every step. Compare its reward-to-go with A's.

Trajectory A Success	a_0 $R_0=1$	a_1 $R_1=1$	a_2 $R_2=1$	a_3 $R_3=1$	a_4 $R_4=1$	a_5 $R_5=1$	$R=1$
Trajectory B Failure	a_0 $R_0=0$	a_1 $R_1=0$	a_2 $R_2=0$	a_3 $R_3=0$	a_4 $R_4=0$	a_5 $R_5=0$	$R=0$

Add the remaining rewards: at every action, $R_t = 1$ on A and $R_t = 0$ on B.

How can we estimate which turns helped when we only observe the final reward?

11

轨迹 B 每一步奖励均为零。将其后续回报与轨迹 A 比较。

轨迹 A 最终成功， $R=1$ ：从 a_0 到 a_5 ，每一步的后续回报 R_t 都是1。轨迹 B 最终失败， $R=0$ ：从 a_0 到 a_5 ，每一步的后续回报 R_t 都是0。

将剩余奖励相加：轨迹 A 上每个动作的 $R_t=1$ ，轨迹 B 上每个动作的 $R_t=0$ 。

如果只能观察最终奖励，怎样估计哪些轮次提供了帮助？

沿两条轨迹分配信用

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Expected reward-to-go from a history

Imagine continuing ten times from B's history h_3 , using the same rollout policy μ .

Same h_3 Ten possible continuations →

Reward-to-go from each continuation (illustrative)

0

1

0

0

1

0

0

0

1

0

Three continuations have $R_3 = 1$. The average is $3 \div 10 = 0.3$.

Value: the average over all possible continuations under μ

$$V^\mu(h_3) = \mathbb{E}_\mu[R_3 \mid h_3]$$

The bar means “given h_3 .” Ten samples give an estimate of this average.

设想从轨迹 B 的历史 h_3 出发，使用相同的采样策略 μ 继续十次。

十种可能的后续轨迹示例，其后续回报分别为：0、1、0、0、1、0、0、0、1、0。

其中三次后续轨迹的 $R_3=1$ ，平均值为 $3 \div 10 = 0.3$ 。

价值：在 μ 下对所有可能后续轨迹取平均。

$V^\mu(h_3) = \mathbb{E}_\mu[R_3 \mid h_3]$

竖线表示 “给定 h_3 ” 。十个样本可用于估计这个平均值。

译注 / 与原文翻译分开

公式符号保留原样。

给定历史时的期望后续回报

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Learning the value function

Train a critic to predict reward-to-go from the history, using squared error.

Its prediction is $V_{\phi}(\mathbf{h})$, with learned parameters ϕ . Reuse the ten outcomes at h_3 : three 1s and seven 0s.

If $V_{\phi}(h_3) = 0.6$, its average squared error is

$$\frac{3(0.6 - 1)^2 + 7(0.6 - 0)^2}{10} = 0.30$$

The best prediction for these samples is their mean, **0.3** (squared error **0.21**).

Minimize over ϕ ; E_t averages histories from sampled rollouts

$$\mathcal{L}_V(\phi) = \mathbb{E}_t \left[(V_{\phi}(h_t) - R_t)^2 \right]$$

[Squared-error baseline training · Ranzato et al., ICLR 2016 · §3.2.1 ↗](#)

训练一个 critic，使用平方误差根据历史预测后续回报。

它的预测记为 $V_{\phi}(h)$ ，参数为 ϕ 。复用 h_3 处的十个结果：三个 1、七个 0。

若 $V_{\phi}(h_3)=0.6$ ，平均平方误差为 $[3(0.6-1)^2 + 7(0.6-0)^2] / 10 = 0.30$ 。

这些样本的最佳预测值是其均值0.3（平方误差0.21）。

对 ϕ 最小化； E 对采样轨迹中的历史取平均。

$L_V(\phi) = E_t[(V_{\phi}(h_t) - R_t)^2]$

平方误差基线训练 · Ranzato 等, ICLR 2016 · §3.2.1 ↗

译注 / 与原文翻译分开

critic 依术语表译为价值评估器（critic）。

学习价值函数

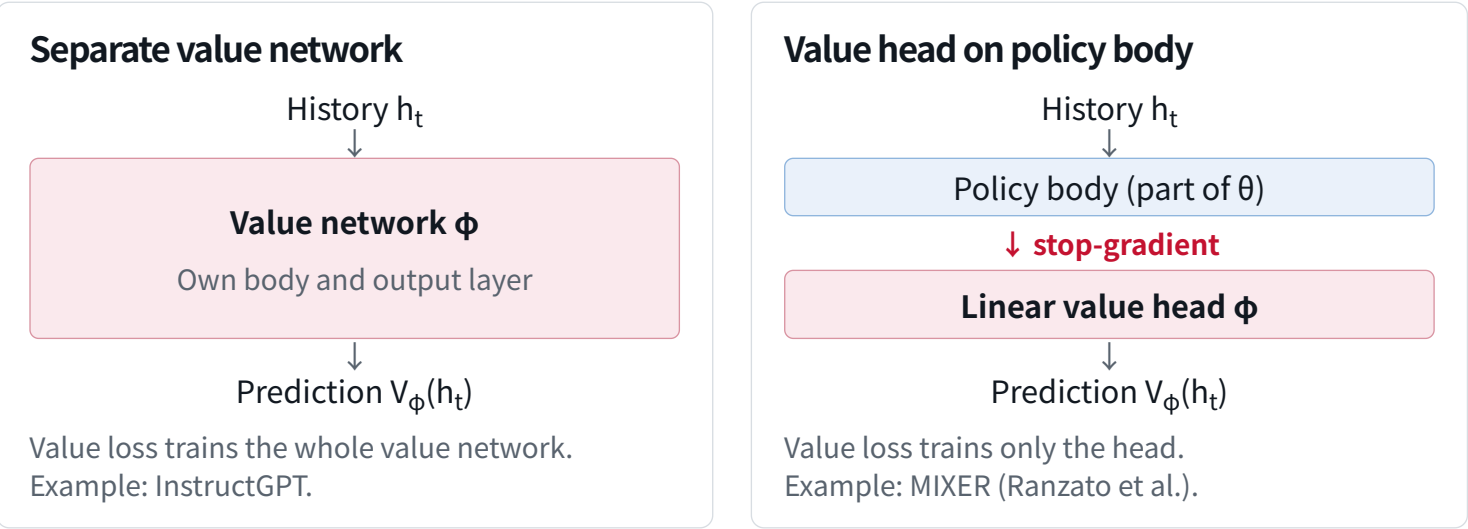
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Modeling the value function

A critic can use its own network or read the policy's hidden representation.



Stop-gradient blocks the value loss from updating the policy body. The policy loss still trains it.

[InstructGPT · Ouyang et al., 2022 · App. C.4 ↗](#) [MIXER · Ranzato et al., 2016 · § 3.2.1 ↗](#)

14

价值评估器可以使用独立网络，也可以读取策略的隐藏表示。

独立价值网络：历史 $h_t \rightarrow$ 价值网络 ϕ （自身网络主体与输出层） $\rightarrow$ 预测 $V_\phi(h_t)$ 。价值损失训练整个价值网络。示例：InstructGPT。

策略主体上的价值头：历史 $h_t \rightarrow$ 策略主体（属于 θ ） $\rightarrow$ stop-gradient $\rightarrow$ 线性价值头 $\phi \rightarrow$ 预测 $V_\phi(h_t)$ 。价值损失只训练价值头。示例：MIXER（Ranzato 等）。

stop-gradient 会阻止价值损失更新策略主体；策略损失仍会训练策略主体。

InstructGPT · Ouyang 等, 2022 · 附录 C.4 ↗ · MIXER · Ranzato 等, 2016 · §3.2.1 ↗

译注 / 与原文翻译分开

变量和 stop-gradient 算子名称保留原样。

价值函数的建模方式

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

One-step change in value

Here, rewards are zero. Compare the critic’s value before and after each action.

Trajectory B · final three actions

a₃		
Before	$V_{\phi}(h_3) = 0.3$	
Reward	$r_4 = 0$	
Next	$V_{\phi}(h_4) = 0.6$	
$0 + 0.6 - 0.3 = \mathbf{+0.3}$		
a₄		
Before	$V_{\phi}(h_4) = 0.6$	
Reward	$r_5 = 0$	
Next	$V_{\phi}(h_5) = 0.4$	
$0 + 0.4 - 0.6 = \mathbf{-0.2}$		
a₅		
Before	$V_{\phi}(h_5) = 0.4$	
Reward	$r_6 = 0$	
Next	$V_{\phi}(h_6) = 0.0$	
$0 + 0.0 - 0.4 = \mathbf{-0.4}$		

B ends after a₅, so no future reward remains at h₆: $V_{\phi}(h_6) = 0$.

Temporal-difference (TD) error, with no reward discounting

$$\delta_t = r_{t+1} + V_{\phi}(h_{t+1}) - V_{\phi}(h_t)$$

15

此处所有奖励均为零。比较 critic 在每个动作前后的价值。

轨迹 B 的最后三步：a₃，动作前 $V_{\phi}(h_3)=0.3$ ，奖励 $r_4=0$ ，之后 $V_{\phi}(h_4)=0.6$ ，差值 $0+0.6-0.3=+0.3$ 。

a₄：动作前 $V_{\phi}(h_4)=0.6$ ，奖励 $r_5=0$ ，之后 $V_{\phi}(h_5)=0.4$ ，差值 $0+0.4-0.6=-0.2$ 。

a₅：动作前 $V_{\phi}(h_5)=0.4$ ，奖励 $r_6=0$ ，之后 $V_{\phi}(h_6)=0.0$ ，差值 $0+0.0-0.4=-0.4$ 。

轨迹 B 在 a₅ 后结束，因此 h₆ 时没有未来奖励： $V_{\phi}(h_6)=0$ 。

不使用奖励折扣的时序差分（TD）误差：
 $\delta_t = r_{t+1} + V_{\phi}(h_{t+1}) - V_{\phi}(h_t)$

译注 / 与原文翻译分开

critic 依术语表译为价值评估器（critic）。

价值的一步变化

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Generalized advantage estimation

How should later errors affect the credit given to B's a_3 ?

Trajectory B · final three actions

a_3

$\delta_3 = +0.3$

Weight: 1

a_4

$\delta_4 = -0.2$

Weight: 0.5

a_5

$\delta_5 = -0.4$

Weight: 0.25

Choose $\lambda = 0.5$: each later error gets half the weight of the previous one.

Estimated advantage at a_3 : combine the three errors

$$\hat{A}_3 = 0.3 + 0.5(-0.2) + 0.25(-0.4) = 0.1$$

GAE repeats this backward: $\hat{A}_t = \delta_t + \lambda \hat{A}_{t+1}$. At the end, no later errors remain.

GAE · Schulman et al., ICLR 2016 ↗

16

之后的误差应如何影响对 B 轨迹中 a_3 的信用分配?

轨迹 B 最后三个动作: a_3 的 $\delta_3 = +0.3$, 权重1; a_4 的 $\delta_4 = -0.2$, 权重0.5; a_5 的 $\delta_5 = -0.4$, 权重0.25。

取 $\lambda = 0.5$: 每个后续误差的权重是前一个误差的一半。

a_3 的估计优势: $\hat{A}^3 = 0.3 + 0.5(-0.2) + 0.25(-0.4) = 0.1$ 。

GAE 以此方式从后往前递推: $\hat{A}_t = \delta_t + \lambda \hat{A}_{t+1}$ 。到轨迹末尾后, 不再有后续误差。

GAE · Schulman 等, ICLR 2016 ↗

广义优势估计

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A surrogate: learning value from annotations

A process reward model (PRM) learns step correctness as a surrogate for value.

The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to $\frac{2}{5}$, what is the numerator of the fraction? (Answer: 14)

☹️

😐

😊

Let's call the numerator x .

☹️

😐

😊

So the denominator is $3x-7$.

☹️

😐

😊

We know that $\frac{x}{(3x-7)} = \frac{2}{5}$.

☹️

😐

😊

So $5x = 2(3x-7)$.

☹️

😐

😊

$5x = 6x - 14$.

☹️

😐

😊

So $x = 7$.

Here, $5x = 6x - 14$ implies $x = 14$. The annotator marks “ $x = 7$ ” as incorrect.

[Let's Verify Step by Step · Lightman et al., 2023 · Figure 1 ↗](#)

过程奖励模型（PRM）学习每一步的正确性，作为价值的替代信号。

这里， $5x = 6x - 14$ 可推出 $x = 14$ 。标注者将 “ $x = 7$ ” 标记为错误。

Let's Verify Step by Step · Lightman 等, 2023 · 图1 ↗

替代信号：从标注中学习价值

同步与异步强化学习

18

个人学习译文 · 非 CMU 官方译本 · 译注独立标记 · 原始课件来自 cmu-agents.com

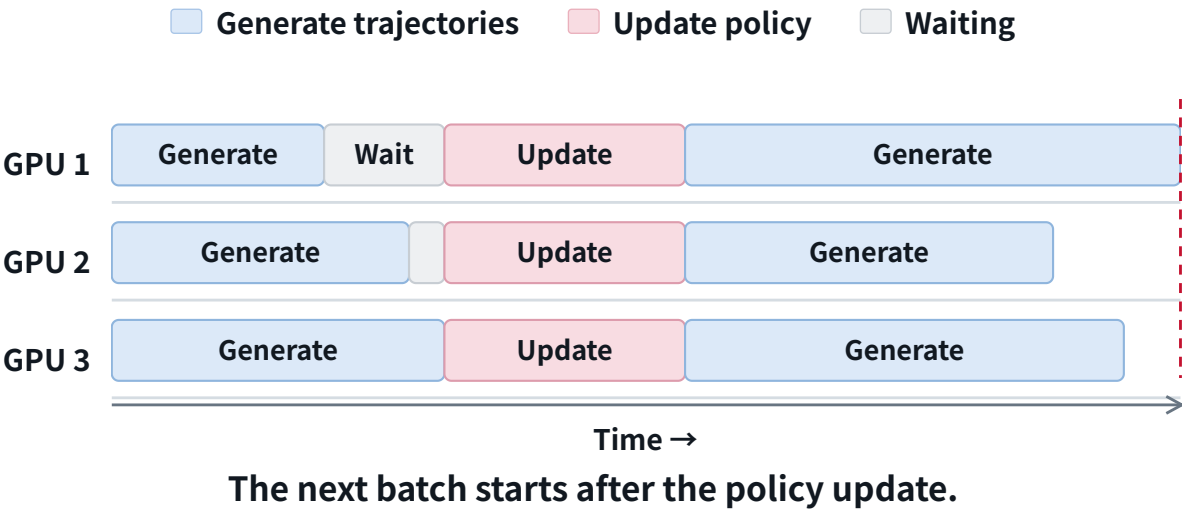
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Waiting in synchronous RL

Every trajectory in a batch must finish before the policy update starts.



[AReaL · Fu et al., 2026 · § 3.2 ↗](#)

批次中的每条轨迹都必须结束，策略更新才能开始。

时间顺序：生成轨迹 → 更新策略 → 等待。GPU 1–3 在轨迹生成与策略更新之间会有等待；下一批要在策略更新后才能开始。

AReaL · Fu 等, 2026 · §3.2 ↗

同步强化学习中的等待

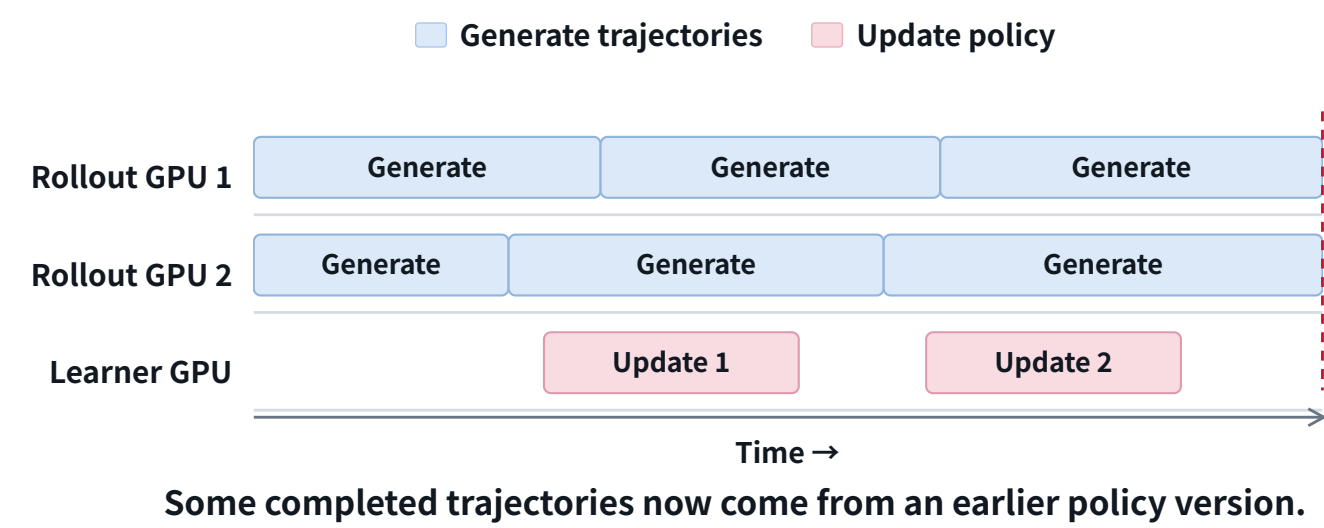
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Overlapping rollouts and updates

Async RL lets rollout workers keep generating while the learner updates from completed trajectories.



[AReal · Fu et al., 2026 · § 4.1 ↗](#)

20

异步 RL 允许采样轨迹工作进程在学习器使用已完成轨迹更新时继续生成轨迹。

时间示意：Rollout GPU 1、2 持续生成；Learner GPU 执行 Update 1、Update 2。

此时一些已完成的轨迹来自较早版本的策略。

AReal · Fu 等, 2026 · §4.1 ↗

重叠执行采样轨迹和更新

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

稳定策略更新



稳定策略更新

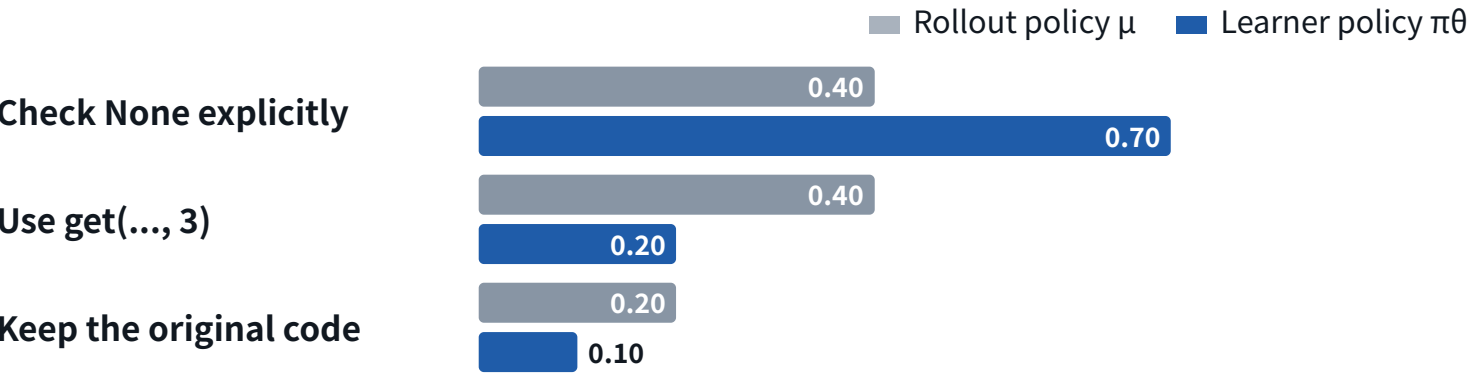
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Rollouts from an older policy

A rollout can finish after the learner has updated to a newer policy.



μ : sampled the recorded edit. $\pi\theta$: is being updated now.

The rollout is stale: its actions came from μ , while the learner uses $\pi\theta$.

22

学习器更新为较新的策略后，某条采样轨迹仍可能才刚完成。

采样策略 μ 与学习器策略 $\pi\theta$ 的概率对照：明确检查 None： $\mu=0.40$ ， $\pi\theta=0.70$ ；使用 `get(..., 3)`： $\mu=0.40$ ， $\pi\theta=0.20$ ；保留原始代码： $\mu=0.20$ ， $\pi\theta=0.10$ 。

μ 生成了记录中的编辑。 $\pi\theta$ 正在更新。

该采样轨迹已经陈旧：其中的动作来自 μ ，而学习器当前使用 $\pi\theta$ 。

译注 / 与原文翻译分开

变量、代码片段和概率按原样保留。

来自旧策略的采样轨迹

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Importance sampling

At a fixed history h , reweight a sampled action a by how likely it is under the current policy.

At a fixed history h

$$\mathbb{E}_{a \sim \pi_{\theta}}[R(a)] = \mathbb{E}_{a \sim \mu} \left[\frac{\pi_{\theta}(a \mid h)}{\mu(a \mid h)} R(a) \right]$$

Correct edit

Behavior: 0.40

Current: 0.70

Weight: **0.70 / 0.40 = 1.75**

Incomplete edit

Behavior: 0.40

Current: 0.20

Weight: **0.20 / 0.40 = 0.50**

$R(a)$ is the final reward after taking action a in this example.

23

在固定历史 h 下，根据动作 a 在当前策略下的可能性，对采样动作重新加权。

在固定历史 h 下：

$$\mathbb{E}_{\{a \sim \pi_{\theta}\}}[R(a)] = \mathbb{E}_{\{a \sim \mu\}}[(\pi_{\theta}(a|h) / \mu(a|h)) R(a)]$$

正确编辑：行为策略概率0.40，当前策略概率0.70，权重 $0.70 / 0.40 = 1.75$ 。

不完整编辑：行为策略概率0.40，当前策略概率0.20，权重 $0.20 / 0.40 = 0.50$ 。

在此示例中， $R(a)$ 是采取动作 a 后得到的最终奖励。

译注 / 与原文翻译分开

公式和变量保留原样。

重要性采样

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

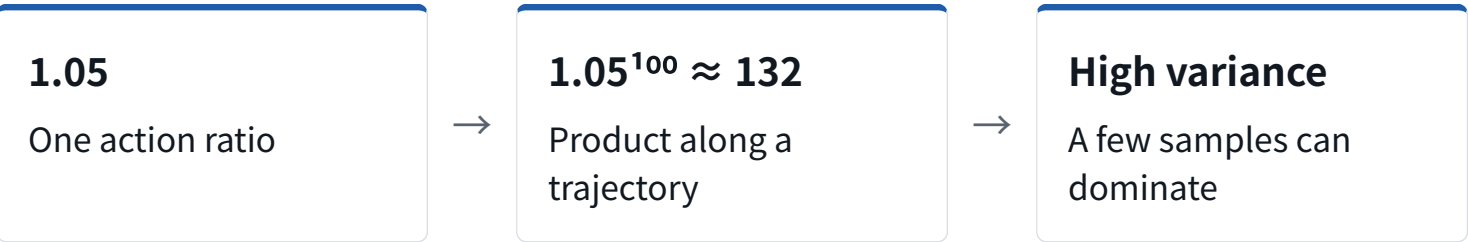
ZH / 中文译文 · 第 1/1 部分

Long trajectories and stale data

The whole-trajectory ratio w multiplies local mismatches along an agent interaction.

Action a_t follows history h_t ; environment dynamics are unchanged

$$w(\tau \mid x) = \frac{p_{\theta}(\tau \mid x)}{p_{\mu}(\tau \mid x)} = \prod_t \frac{\pi_{\theta}(a_t \mid h_t)}{\mu(a_t \mid h_t)}$$



CTPO · Zhang et al., 2026 · § 3.2 ↗

24

整条轨迹的比率 w 会将智能体交互过程中局部不匹配累乘起来。

动作 a_t 接续历史 h_t ; 环境动态保持不变。

$$w(\tau|x) = p_{\theta}(\tau|x) / p_{\mu}(\tau|x) = \prod_t [\pi_{\theta}(a_t|h_t) / \mu(a_t|h_t)]$$

单个动作比率为1.05 → 沿一条轨迹连乘: $1.05^{100} \approx 132$ → 高方差: 少数样本可能主导更新。

CTPO · Zhang 等, 2026 · §3.2 ↗

译注 / 与原文翻译分开

公式和变量保留原样。

长轨迹与陈旧数据

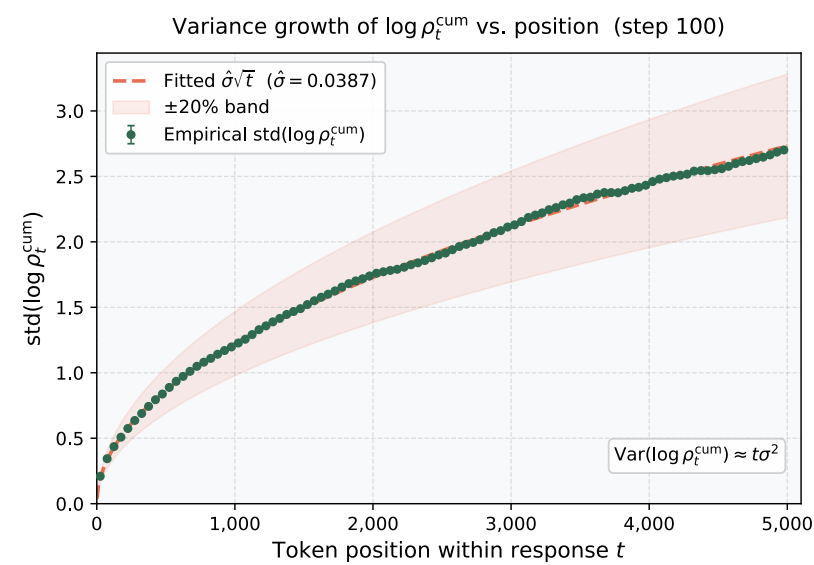
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Mismatch grows along a response

The cumulative ratio ρ_t^{cum} multiplies action ratios through token t .



Its spread grows at later positions in tool-using math rollouts.

[CTPO · Zhang et al., 2026 · Fig. 1 ↗](#)

25

累积比率 ρ_t^{cum} 会将直到词元 t 为止的动作比率相乘。

图表：第100步时， $\log \rho_t^{cum}$ 的方差随响应中词元位置增加；横轴为响应内词元位置 t (0–5,000)，纵轴为 $\text{std}(\log \rho_t^{cum})$ (0–约3)。图例：拟合的 $\hat{\sigma}\sqrt{t}$ ($\hat{\sigma}=0.0387$)； $\pm 20\%$ 范围；经验标准差 $\text{std}(\log \rho_t^{cum})$ 。图中还标出 $\text{Var}(\log \rho_t^{cum}) \approx t\sigma^2$ 。横轴刻度：0、1000、2000、3000、4000、5000；纵轴刻度：0.0、0.5、1.0、1.5、2.0、2.5、3.0。

在使用工具的数学采样轨迹中，越靠后的词元位置，其分布范围越大。

CTPO · Zhang 等, 2026 · 图1 ↗

译注 / 与原文翻译分开

纵轴末端读数约为2.7，按图表刻度转录。

不匹配会沿响应不断累积

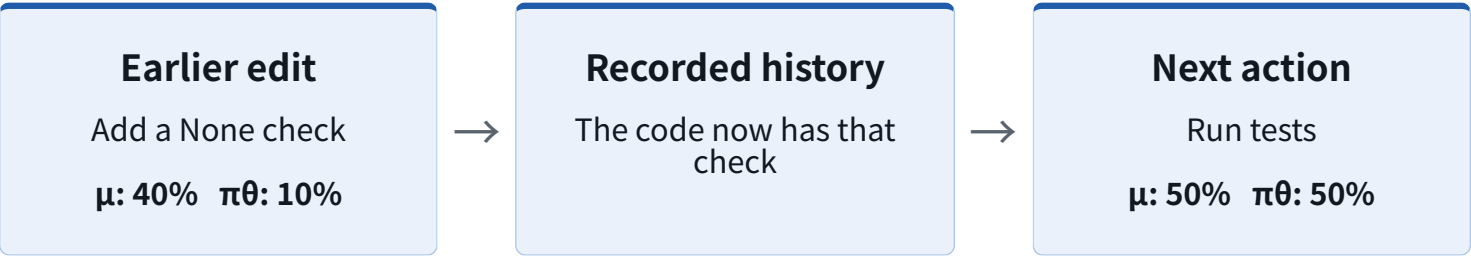
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Earlier actions change later histories

The current policy may rarely reach a history that the rollout policy visited.



The next-action ratio is 1, but reaching this history is **0.10 / 0.40 = 0.25** as likely.

CTPO · Zhang et al., 2026 · § 3.1 ↗

当前策略可能很少到达采样策略曾访问过的历史。

较早的编辑：添加 None 检查 → 记录的历史：代码现在包含该检查 → 下一动作：运行测试。

较早动作概率：μ=40%，πθ=10%；下一动作概率：μ=50%，πθ=50%。

下一动作的比率为1，但到达该历史的可能性只有 0.10 / 0.40 = 0.25。

CTPO · Zhang 等, 2026 · §3.1 ↗

较早的动作会改变后续历史

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Which actions enter the weight?

For a response with T actions, ρ_t is the probability ratio for action t . Methods use different parts of the response.

Method	Recorded actions →					Weight uses
PPO / GRPO	a_0	a_1	a_t	a_{t+1}	a_{T-1}	Current action
CTPO	a_0	a_1	a_t	a_{t+1}	a_{T-1}	Actions through t
Full ratio	a_0	a_1	a_t	a_{t+1}	a_{T-1}	All actions
GSPO	a_0	a_1	a_t	a_{t+1}	a_{T-1}	All, length-normalized

DAPO and CISPO also use the current-action ratio; their clipping rules differ.
[CTPO · Zhang et al., 2026 · Table 1 ↗](#) [DAPO · Yu et al., 2025 ↗](#) [CISPO · MiniMax-M1, 2025 ↗](#)

27

对于含 T 个动作的响应， ρ_t 是动作 t 的概率比率。不同方法使用响应中的不同部分。

PPO / GRPO：记录动作 a_0 、 a_1 、 a_t 、 a_{t+1} ... a_{T-1} ；权重只使用当前动作。

CTPO：权重使用截至 t 的动作。

完整比率：权重使用全部动作。

GSPO：权重使用全部动作，并按长度归一化。

DAPO 与 CISPO 也使用当前动作比率，但裁剪规则不同。

CTPO · Zhang 等, 2026 · 表1 ↗ · DAPO · Yu 等, 2025 ↗ · CISPO · MiniMax-M1, 2025 ↗

哪些动作会进入权重计算？

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

裁剪策略更新

Clipping policy updates

28

裁剪策略更新

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Why limit an update?

Stale rollouts can get large weights, letting a few samples dominate an update.

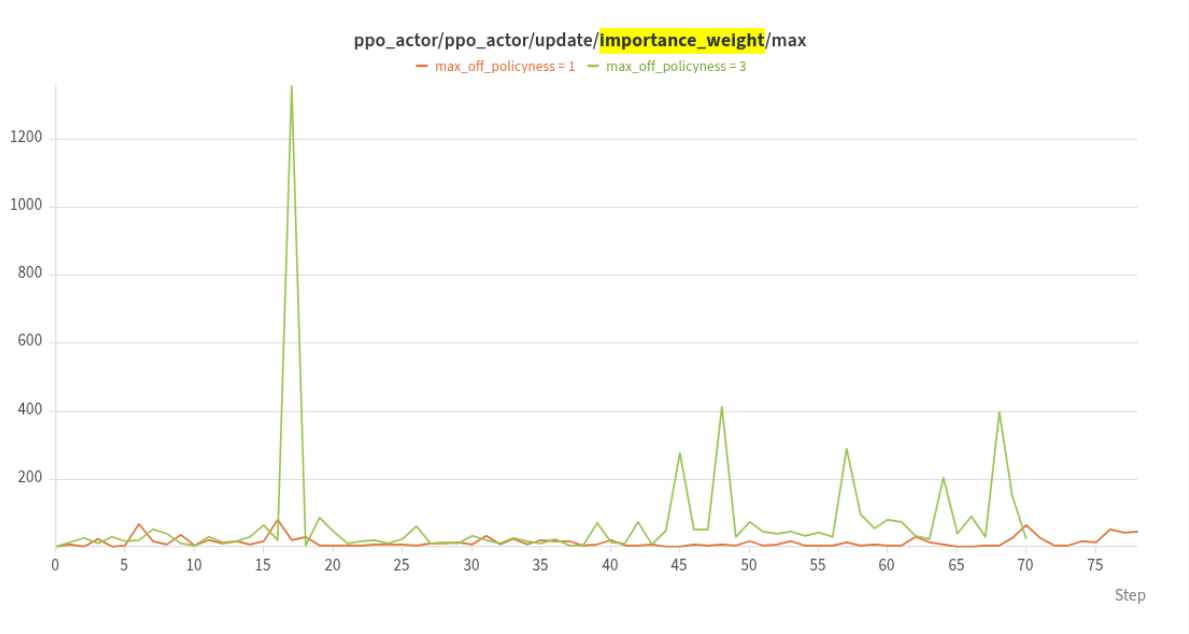


Image Credit: Aditya Soni ↗

29

陈旧的采样轨迹可能获得很大的权重，使少数样本主导一次更新。

图片来源: Aditya Soni ↗

为什么要限制一次更新？

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Clipping a probability ratio

The clip function replaces a ratio outside the chosen bounds with the nearest bound.

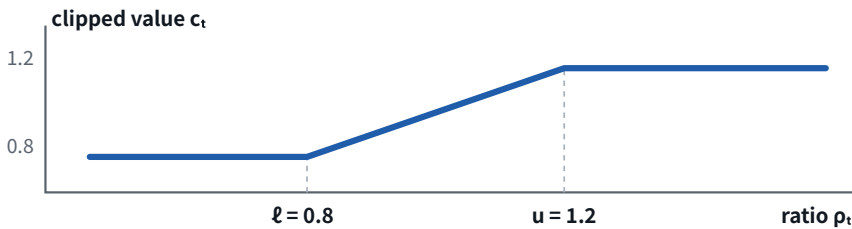
Here each action is a generated token u_t ; $u_{<t}$ is its earlier context.

Ratio for one sampled token

$$\rho_t = \frac{\pi_{\theta}(u_t \mid u_{<t})}{\mu(u_t \mid u_{<t})}$$

Clipped value: ℓ is lower, u is upper

$$c_t = \text{clip}(\rho_t, \ell, u)$$



PPO · Schulman et al., 2017 ↗ CISPO · MiniMax-M1, 2025 ↗

30

裁剪函数会将所选范围之外的比率替换为最近的边界值。

这里，每个动作都是生成的词元 u_t ； $u_{<t}$ 是它之前的上下文。

一个采样词元的比率： $\rho_t = \pi_{\theta}(u_t \mid u_{<t}) / \mu(u_t \mid u_{<t})$ 。

裁剪值： ℓ 为下界， u 为上界； $c_t = \text{clip}(\rho_t, \ell, u)$ 。

图表展示 clipped value c_t 对 ratio ρ_t 的关系：低于下界时取 $\ell=0.8$ ，高于上界时取 $u=1.2$ ，中间区间保持原比率。

PPO · Schulman 等, 2017 ↗ · CISPO · MiniMax-M1, 2025 ↗

译注 / 与原文翻译分开

公式和符号保留原样。

裁剪概率比率

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

PPO and CISPO policy objectives

PPO puts c_t inside a minimum; CISPO uses it as a fixed weight.

Shared clipped ratio	$c_t = \text{clip}(\rho_t, \ell, u)$
PPO policy objective to maximize	$J_{\text{PPO}}(\theta) = \mathbb{E}_t \left[\min \left(\rho_t \hat{A}_t, c_t \hat{A}_t \right) \right]$
CISPO policy objective to maximize	$J_{\text{CISPO}}(\theta) = \mathbb{E}_t \left[\text{sg}(c_t) \hat{A}_t \log \pi_{\theta}(u_t \mid u_{<t}) \right]$

$\mathbb{E}_t$ averages sampled tokens; $\hat{A}_t$ estimates whether a token helped. sg holds c_t fixed for gradients. A loss to minimize is $-J$.

PPO · Schulman et al., 2017 ↗ CISPO · MiniMax-M1, 2025 ↗

31

PPO 将 c_t 放入最小值运算；CISPO 将其作为固定权重。

共同的裁剪比率： $c_t = \text{clip}(\rho_t, \ell, u)$ 。

要最大化的 PPO 策略目标：

$$J_{\text{PPO}}(\theta) = \mathbb{E}_t[\min(\rho_t \hat{A}_t, c_t \hat{A}_t)]$$

要最大化的 CISPO 策略目标：

$$J_{\text{CISPO}}(\theta) = \mathbb{E}_t[\text{sg}(c_t) \hat{A}_t \log \pi_{\theta}(u_t \mid u_{<t})]$$

$\mathbb{E}_t$ 对采样词元取平均； $\hat{A}_t$ 估计词元是否带来帮助。sg 在梯度计算中固定 c_t 。
要最小化的损失为 $-J$ 。

PPO · Schulman 等, 2017 ↗ · CISPO · MiniMax-M1, 2025 ↗

译注 / 与原文翻译分开

公式和变量保留原样。

PPO 与 CISPO 的策略目标

第 11 讲 · 智能体的高级强化学习算法

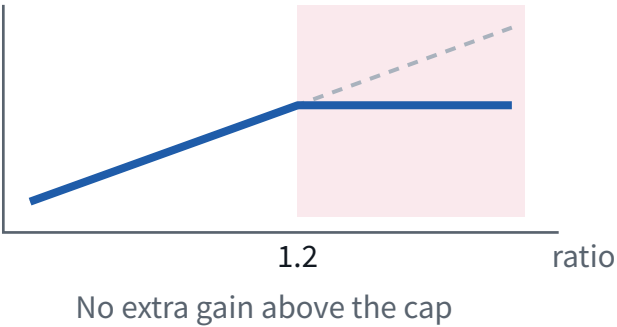
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

PPO’s clipped objective

PPO stops rewarding increases above the cap and decreases below the floor.

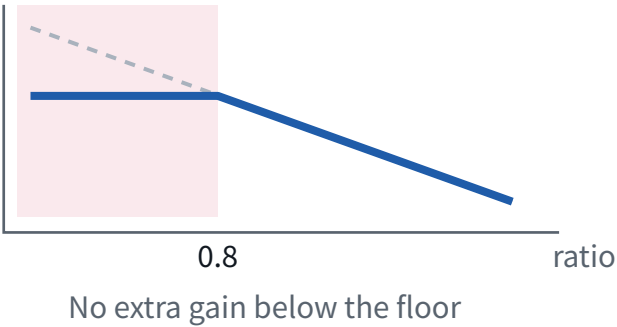
Positive advantage: $A = +1$



For a helpful action, stop rewarding a ratio above $1 + \epsilon$.

In this plot, $\epsilon = 0.2$ is the clipping margin.

Negative advantage: $A = -1$



For an unhelpful action, stop rewarding a ratio below $1 - \epsilon$.

Adapted from PPO · Schulman et al., 2017 · Figure 1 ↗

32

PPO 停止奖励超过上限的比率上升，也停止奖励低于下限的比率下降。

正优势 $A = +1$ ：比率高于上限后不再获得额外收益。负优势 $A = -1$ ：比率低于下限后不再获得额外收益。

对有帮助的动作，当比率高于 $1 + \epsilon$ 时停止增加奖励；对没有帮助的动作，当比率低于 $1 - \epsilon$ 时停止增加奖励。

本图中 $\epsilon = 0.2$ ，是裁剪幅度。

改编自 PPO · Schulman 等，2017 · 图1 ↗

PPO 的裁剪目标

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reference-model KL

A KL penalty discourages the policy from drifting far from a fixed reference model.

Policy objective with a reference penalty

$$\max_{\theta} J_{\text{policy}} - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}})$$

Rollout policy μ

Generates the training batch and supplies the denominator in importance ratios.

Reference model π_{ref}

Stays fixed as an anchor while the policy learns.

J_{policy} is the policy objective; D_{KL} measures distribution difference; β sets the penalty strength.

[DeepSeekMath · Shao et al., 2024 · GRPO objective ↗](#)

33

KL 惩罚会阻止策略偏离固定参考模型太远。

带参考惩罚的策略目标： $\max_{\theta} J_{\text{policy}} - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}})$ 。

采样策略 μ ：生成训练批次，并提供重要性比率中的分母。

参考模型 π_{ref} ：作为锚点保持固定，供策略学习时参照。

J_{policy} 是策略目标； D_{KL} 衡量分布差异； β 控制惩罚强度。

DeepSeekMath · Shao 等, 2024 · GRPO 目标 ↗

译注 / 与原文翻译分开

数学符号与变量保留原样。

参考模型 KL 惩罚

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Entropy for exploration

An entropy bonus rewards a broader next-token distribution.

Policy objective with an entropy bonus

$$\max_{\theta} J_{\text{policy}} + \alpha \mathcal{H}(\pi_{\theta})$$

H measures next-token uncertainty; α sets the bonus strength.

- A broader distribution can help the agent try other actions.
- It does not guarantee that those actions will be useful.

Proximal Policy Optimization · Schulman et al., 2017 · § 5 ↗

34

熵奖励会促进更广的下一词元分布。

带熵奖励的策略目标： $\max_{\theta} J_{\text{policy}} + \alpha H(\pi_{\theta})$ 。

H 衡量下一词元的不确定性； α 控制奖励强度。

更广的分布可能帮助智能体尝试其他动作。

但这不保证这些动作有用。

Proximal Policy Optimization · Schulman 等, 2017 · §5 ↗

用于探索的熵

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A summary of popular algorithms

Compare how each method assigns credit, weights sampled tokens, and limits updates.

Algorithm	Weight from reward	Learned critic?	Importance ratio	Clipping
REINFORCE	Reward-to-go R_t	No	None: fresh rollouts	None
PPO	GAE from a critic	Yes	Current token p_t	Minimum of raw and clipped terms
GRPO	Group comparison	No	Current token p_t	Same as PPO
CISPO	Group comparison	No	Current token p_t	Clipped ratio as a fixed weight
GSPO	Group comparison	No	Whole response, length-normalized	PPO minimum, on the response ratio
DAPO	Group comparison	No	Current token p_t	PPO minimum; higher upper bound

Group comparison: subtract the group’s mean reward, then divide by its reward spread.

[PPO ↗](#) [GRPO ↗](#) [CISPO ↗](#) [GSPO ↗](#) [DAPO ↗](#)

35

比较各方法如何分配信用、为采样词元加权并限制更新。表头：算法 | 奖励带来的权重 | 是否使用学习得到的价值评估器（critic）？ | 重要性比率 | 裁剪。

REINFORCE：后续回报 R_t ；否；不使用比率，采用最新采样轨迹；不裁剪。

PPO：由价值评估器（critic）计算 GAE；是；当前词元 p_t ；取原始项与裁剪项的最小值。

GRPO：组内比较；否；当前词元 p_t ；与 PPO 相同。

CISPO：组内比较；否；当前词元 p_t ；将裁剪比率作为固定权重。

GSPO：组内比较；否；整条响应的比率，按长度归一化；对响应比率使用 PPO 最小值。

DAPO：组内比较；否；当前词元 p_t ；PPO 最小值，使用较高的上界。

组内比较：先减去组内平均奖励，再除以奖励的离散程度。

PPO ↗ · GRPO ↗ · CISPO ↗ · GSPO ↗ · DAPO ↗

常见算法概览

Task selection and reward reliability

36

任务选择与奖励可靠性

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Uninformative rollout groups

Sample four trajectories per task. Each group below shows their final rewards.

All attempts failed

0

0

0

0

Mean reward: 0
Subtract the mean:
0, 0, 0, 0

Mixed outcomes

0

1

0

1

Mean reward: 0.5
Subtract the mean:
-0.5, +0.5, -0.5, +0.5

All attempts passed

1

1

1

1

Mean reward: 1
Subtract the mean:
0, 0, 0, 0

Recall group advantages: subtract the group's mean reward. Equal rewards all become zero.

An all-zero group gives no outcome comparison. Why did every trajectory fail?

37

每项任务采样四条轨迹。每组下方给出它们的最终奖励。

全部失败：0、0、0、0；平均奖励0；减去平均值后为0、0、0、0。

结果混合：0、1、0、1；平均奖励0.5；减去平均值后为-0.5、+0.5、-0.5、+0.5。

全部通过：1、1、1、1；平均奖励1；减去平均值后为0、0、0、0。

回顾组优势：减去组内平均奖励。奖励相同的轨迹，其优势都变成零。

全零组无法比较结果。为什么每条轨迹都失败了？

信息量不足的轨迹组

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Why did every trajectory fail?

Four failed trajectories give the same rewards: 0, 0, 0, 0. Inspect what happened before choosing a remedy.

Verifier quality

A valid solution might be rejected.

Check: the requirements and failed tests.

Current capability

The model may not yet be able to solve the task.

Check: whether stronger-model demonstrations succeed.

Lack of exploration

All attempts may repeat the same approach.

Check: whether the trajectories differ.

38

四条失败轨迹的奖励相同：0、0、0、0。决定补救方案前，先检查发生了什么。

验证器质量：有效解法可能被拒绝。检查要求和失败的测试。

当前能力：模型可能还不能解决任务。检查更强模型的演示能否成功。

缺乏探索：所有尝试可能都重复同一种方法。检查轨迹之间是否存在差异。

为什么所有轨迹都失败了？

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

False negatives in the verifier

Return to the bug-fix task. This candidate produces every required output.

Missing → 3 ✓	None → 3 ✓
0 → 0 ✓	2 → 2 ✓
Candidate behavior All four cases are correct. Its implementation uses different source code.	Verifier rule Require the exact phrase value is None . The candidate lacks it: reject.

This is a **false negative**: a valid solution is rejected. Check behavior instead of an incidental wording choice.

[Related task/test review: SWE-bench Verified · 2024 ↗](#)

39

回到修复错误任务。这个候选方案会产生所有要求的输出。

缺失 → 3 ✓; None → 3 ✓; 0 → 0 ✓; 2 → 2 ✓。

候选行为：四种情况都正确，但实现代码不同。

验证器规则：要求精确出现短语 “value is None” 。候选方案中没有这个短语，因此被拒绝。

这是假阴性：有效解法被拒绝。应检查行为，而非偶然的措辞选择。

相关任务 / 测试审查：SWE-bench Verified · 2024 ↗

验证器的假阴性

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Result of false negatives

Reward noise

Fitting to spurious changes in the output.

Equivalent solutions receive different rewards because of wording or implementation details.

Benchmark saturation

Scores can flatten below 100% when the evaluator rejects valid solutions.

Benchmark	Plateau / limit	Evidence from the audit
SWE-bench Verified	≈81%	Progress slowed near 80.9%; many remaining failures had flawed tests.
τ-bench Airline	≈70%	Inconsistent tasks limited achievable scores.
τ-bench Retail	≈92%	Annotation errors limited achievable scores.

Historical versions, audited in 2025–26. These are reported plateaus or limits; τ-bench has since corrected the tasks.

[SWE-bench audit · OpenAI, Feb. 2026 ↗](#) [SABER · Cuadron et al., 2025 · § 5.1 ↗](#) [τ-bench fixes · Feb. 2026 ↗](#)

40

奖励噪声：模型会拟合输出中的伪相关变化。等价解法会因为措辞或实现细节不同而得到不同奖励。

基准饱和：评估器拒绝有效解法时，分数可能会在低于100%的位置停滞。

SWE-bench Verified：平台期约81%；进展在80.9%附近放缓，审计发现许多剩余失败来自有缺陷的测试。

τ-bench Airline：平台期约70%；任务不一致限制了可达分数。

τ-bench Retail：平台期约92%；标注错误限制了可达分数。

这些是历史版本，审计时间为2025–26年。列出的是报告的平台期或上限；τ-bench 此后已修正相关任务。

SWE-bench 审计 · OpenAI, 2026年2月 ↗ · SABER · Cuadron 等, 2025 · §5.1 ↗ · τ-bench 修复 · 2026年2月 ↗

假阴性的后果

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

False positives and reward hacking

Keep the same task requirements, but suppose training tests only the input retries=0.

```
def retry_count(config):  
    return 0
```

Candidate: always return zero.

Training verifier
The one tested case passes.
Reward: **1**

Input	Required output	Candidate output
0	0	0 ✓
Missing / None	3	0 ✗
2	2	0 ✗

A **false positive** rewards an invalid solution. Training can learn to exploit this gap.

41

保持任务要求不变，但假设训练测试只检查 retries=0。

```
def retry_count(config):  
    return 0
```

候选方案：始终返回零。训练验证器：唯一被测情况通过，奖励为1。

输入0：要求输出0，候选输出0 ✓。输入缺失 / None：要求输出3，候选输出0 ✗。
输入2：要求输出2，候选输出0 ✗。

假阳性会奖励无效解法。训练可能学会利用这一漏洞。

译注 / 与原文翻译分开

代码与配置字段保留原样。

假阳性与奖励投机

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Reward hacking in coding agents

A passing score can come from a shortcut that the benchmark was meant to exclude.

Available in the environment		Shortcut
Internet access	→	Find the published solution online.
Git history, including future commits	→	Find and copy the later bug fix.
A model API key	→	Call a different, stronger model for answers or training data.
Access to tests or the test runner	→	Make the tests pass without fixing the bug.

Documented API example: a PostTrainBench agent used a grading key to generate training data, despite an explicit restriction.

[Web, Git, and test exploits · MAI-Thinking-1 · § 3.3.1, p.43 ↗](#) [API misuse · PostTrainBench · § 5.4, Fig.7 ↗](#)

42

基准原本要排除的捷径，也可能得到及格分数。

环境中提供互联网访问 → 在网上查找已发表的解答。

环境中提供 Git 历史（包括未来提交）→ 找到并复制后续的错误修复。

环境中提供模型 API key → 调用另一个更强的模型获取答案或训练数据。

环境中提供测试或测试运行器的访问权限 → 让测试通过，却不修复错误。

有记录的 API 示例：PostTrainBench 智能体使用评分 key 生成训练数据，尽管明确规定禁止这样做。

利用网页、Git 与测试 · MAI-Thinking-1 · §3.3.1，第43页 ↗ · API 滥用 · PostTrainBench · §5.4，图7 ↗

编码智能体中的奖励投机

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Independent checks of progress

The constant-zero patch earned training reward 1, but passed only one of the four required cases.

Training reward Measures success on the training verifier	Independent success Check unseen tasks and behavior the training verifier missed	Costs and failures Inspect tool calls, regressions, and repeated actions
--	---	---

If reward rises without independent improvement, inspect trajectories for shortcuts.

43

常量零补丁获得训练奖励1，但四项要求中只通过了一项。

训练奖励：衡量训练验证器上的成功。

独立成功：检查未见过的任务和训练验证器漏掉的行为。

成本与失败：检查工具调用、回归问题和重复动作。

如果奖励上升而独立评测没有改善，应检查轨迹是否走了捷径。

独立检查进展

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

DAPO: dynamic sampling

Identical rewards give zero group advantages. DAPO keeps mixed groups and samples until the batch is full.

Sample four trajectories per task

Each circle is a reward: 0 = failure, 1 = success.

Task A

0

0

0

0

Skip group

Identical rewards

Task B

0

1

0

1

Keep group

Mixed rewards

Task C

1

1

1

1

Skip group

Identical rewards

Task D

0

0

1

0

Keep group

Mixed rewards

New group



Training batch

Example target: 2 groups

2 / 2 groups collected

Task B

0

1

0

1

Task D

0

0

1

0

Batch full → update the policy

Keep every trajectory in a mixed group, including its failures.

DAPO · Yu et al., 2025 · § 3.2 Dynamic Sampling ↗

44

相同奖励会产生零组优势。DAPO 保留结果混合的轨迹组，并持续采样直到批次填满。

每个任务采样四条轨迹；每个圆圈表示奖励：0=失败，1=成功。

Task A：0、0、0、0，跳过该组（奖励相同）。Task B：0、1、0、1，保留该组（奖励混合）。Task C：1、1、1、1，跳过该组（奖励相同）。Task D：0、0、1、0，保留该组（奖励混合；作为新组采样）。

训练批次示例目标为2组：已收集2/2组，包含 Task B 与 Task D 的奖励；批次填满 → 更新策略。

保留混合组中的每条轨迹，包括失败轨迹。

DAPO · Yu 等, 2025 · §3.2 Dynamic Sampling ↗

DAPO：动态采样

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Partial and auxiliary rewards

Partial credit is an auxiliary reward: extra feedback for progress before full success.

Illustrative task: pass all 4 checks. Add 0.25 reward per check passed.

Four trajectories on the same task	τ_1	τ_2	τ_3	τ_4
Checks passed	0 / 4	1 / 4	2 / 4	1 / 4
Success reward	0	0	0	0
Subtract the mean: 0	0	0	0	0
Success + partial reward	0	0.25	0.50	0.25
Subtract the mean: 0.25	-0.25	0	+0.25	0

Different partial scores can reduce zero-advantage groups and batches.

45

部分得分是一种辅助奖励：在完全成功前，为进展提供额外反馈。

示例任务：通过全部4项检查。每通过一项检查，增加0.25奖励。

同一任务上的四条轨迹 τ_1 、 τ_2 、 τ_3 、 τ_4 ：通过检查数分别为0/4、1/4、2/4、1/4。成功奖励为0、0、0、0；减去均值0后仍为0、0、0、0。

成功奖励 + 部分奖励：0、0.25、0.50、0.25。减去均值0.25后：-0.25、0、+0.25、0。

部分得分不同，可以减少整组或整个批次都没有优势信号的情况。

部分奖励与辅助奖励

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Risks of auxiliary rewards

The model may learn to increase the added reward without improving the task.

Intended behavior	What the reward favored	Observed behavior
Useful tool use	A bug gave credit for superficial web-tool calls.	GPT-5.1 used the browser as a calculator and acted as though it had searched.
A playful “Nerdy” persona	Metaphors with “goblin” or “gremlin” received higher scores.	Creature metaphors became more common, including without the persona prompt.

[Calculator hacking · OpenAI, Dec. 2025 ↗](#) [Where the goblins came from · OpenAI, Apr. 2026 ↗](#)

模型可能学会增加额外奖励，却没有改进任务表现。

预期行为：有用地使用工具。奖励偏向：一个错误给表面化的网页工具调用加分。
实际行为：GPT-5.1 把浏览器当计算器用，并假装自己搜索过。

预期行为：活泼的 “Nerdy” 角色。奖励偏向：包含 “goblin” 或 “gremlin” 的比喻得分更高。实际行为：生物比喻变得更常见，即使没有角色提示也是如此。

Calculator hacking · OpenAI, 2025年12月 ↗ · Where the goblins came from · OpenAI, 2026年4月 ↗

辅助奖励的风险

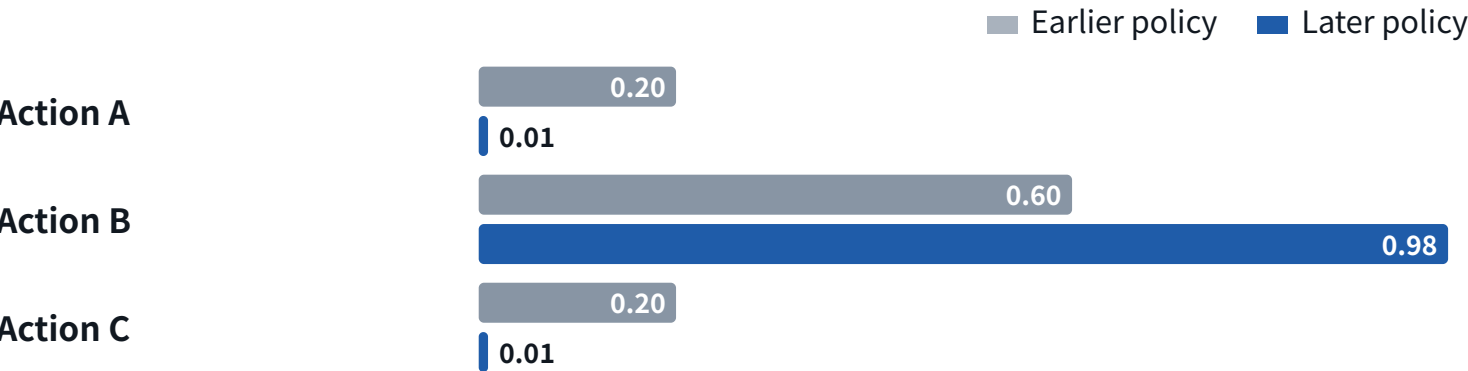
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Exploration collapse

At one history, training changes the probabilities of three possible actions.



- Almost every sample now chooses B, so alternatives are rarely tried.
- An entropy bonus encourages alternatives; check whether trajectories differ.

[DAPO · Yu et al., 2025 · exploration and entropy collapse ↗](#)

47

在一个历史状态下，训练改变三个可能动作的概率。

较早策略 → 较晚策略：Action A, 0.20 → 0.01；Action B, 0.60 → 0.98；Action C, 0.20 → 0.01。

现在几乎所有样本都会选择 B，因此很少尝试其他选项。

熵奖励会鼓励尝试其他选项；还要检查轨迹是否存在差异。

[DAPO · Yu 等, 2025 · 探索与熵崩塌 ↗](#)

探索崩塌

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

蒸馏：匹配教师的动作概率。

Learning from a teacher

Distillation: match a teacher's action probabilities

48

向教师模型学习

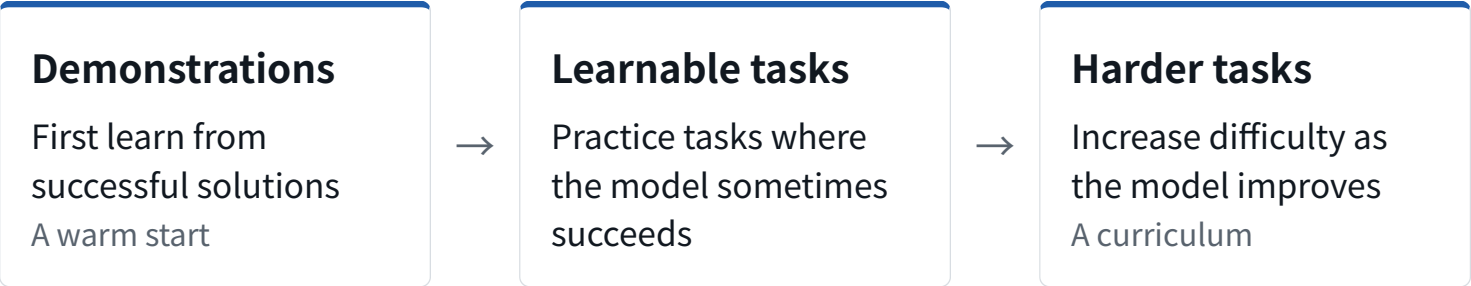
第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Curricula and warm starts

When the student rarely succeeds, a teacher can provide successful trajectories to learn from.



Keep earlier tasks in the mix, and use a fixed evaluation set to measure improvement.

49

学生模型很少成功时，教师模型可以提供成功轨迹供其学习。

示范：先从成功解法中学习——热启动。

可学习的任务：练习模型有时能成功的任务。

更难的任务：随着模型进步增加难度——课程学习。

将早期任务继续混入训练，并用固定评测集衡量进步。

课程学习与热启动

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Distillation

Train a student to match a teacher’s action probabilities on a fixed set of trajectories.

Saved trajectory from the teacher



At this saved history, the teacher assigns **80% to a_3** and **20% to the other action**. Call this distribution $q(\cdot | h_3)$.

When the student acts, its own mistakes can lead to histories missing from this training set.

Knowledge distillation · Hinton et al., 2015 · § 2 ↗ Distillation for language models · Agarwal et al., 2024 · § 3 ↗

在一组固定轨迹上训练学生模型，使其动作概率与教师模型匹配。

教师保存的轨迹： $h_0 \rightarrow a_0 \rightarrow h_1 \rightarrow a_1 \rightarrow h_2 \rightarrow a_2 \rightarrow h_3$ （在此处选择下一个动作）。

在这段保存的历史中，教师给 a_3 分配80%概率，给另一个动作分配20%。将此分布记为 $q(\cdot | h_3)$ 。

学生采取动作时，自身错误可能使它到达训练集中未包含的历史。

Knowledge distillation · Hinton 等, 2015 · §2 ↗ · Distillation for language models · Agarwal 等, 2024 · §3 ↗

蒸馏

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

On-policy distillation

Let the student generate the training histories, then ask the teacher for targets at those histories.

Student rollout on trajectory B



Example teacher targets at this student history: **80% for a_3** and **20% for the other action.**

Train the student to match these probabilities. “On-policy” means the histories came from the student's own rollouts.

[On-Policy Distillation of Language Models · Agarwal et al., ICLR 2024 ↗](#)

51

让学生模型生成训练历史，再在这些历史上向教师模型请求目标分布。

学生在轨迹 B 上采样： $h_0 \rightarrow a_0 \rightarrow h_1 \rightarrow a_1 \rightarrow h_2 \rightarrow a_2 \rightarrow h_3$ （在此处选择下一个动作）。

学生到达这段历史时，示例教师目标为： a_3 的概率80%，另一个动作的概率20%。

训练学生匹配这些概率。“在策略”表示这些历史来自学生自己的采样轨迹。

On-Policy Distillation of Language Models · Agarwal 等, ICLR 2024 ↗

在策略蒸馏

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

A dense distillation objective

At the same h_3 , suppose the student assigns equal probability to the two actions.

Next-action probabilities	a_3	Other action
Student π_θ	0.5	0.5
Teacher q	0.8	0.2

Reverse KL measures probability mismatch (natural logarithms)

$$0.5 \log(0.5/0.8) + 0.5 \log(0.5/0.2) \approx 0.223$$

Average this KL mismatch over student histories (distribution d_μ), then minimize:

$$L_{OPD} = \mathbb{E}_{h \sim d_\mu} [D_{KL}(\pi_\theta(\cdot | h) \| q(\cdot | h))]$$

GKD · Agarwal et al., ICLR 2024 · choice of divergence ↗

在同一 h_3 处，假设学生对两个动作分配相同概率。

下一动作概率：学生策略 π_θ 对 a_3 和其他动作分别为0.5、0.5；教师 q 分别为0.8、0.2。

反向 KL 衡量概率差异（使用自然对数）： $0.5 \log(0.5/0.8) + 0.5 \log(0.5/0.2) \approx 0.223$ 。

对学生历史（分布 d_μ ）上的 KL 差异取平均，然后最小化：
 $L_{OPD} = \mathbb{E}_{\{h \sim d_\mu\}} [D_{KL}(\pi_\theta(\cdot | h) \| q(\cdot | h))]$

GKD · Agarwal 等, ICLR 2024 · 散度选择 ↗

译注 / 与原文翻译分开

公式与变量保留原样。

密集蒸馏目标

第 11 讲 · 智能体的高级强化学习算法

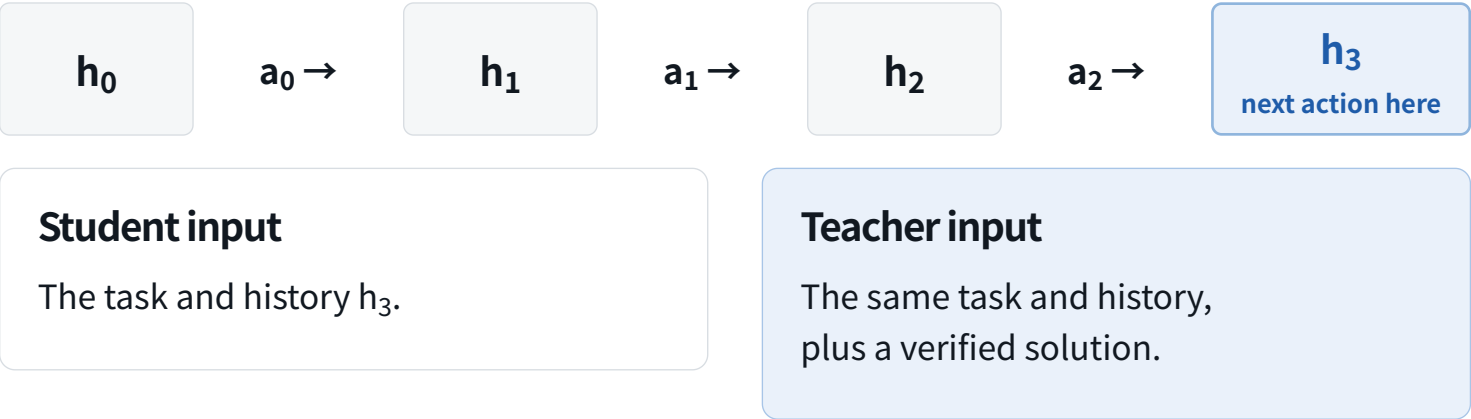
EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

On-policy self-distillation

A verified solution can help a teacher give better targets at the student's own history.

Student rollout on trajectory B



OPSD uses a frozen copy of the starting model as teacher. Its next-action probabilities train the student as before.

[Self-Distilled Reasoner · Zhao et al., 2026 · OPSD ↗](#)

53

已验证的解法可以帮助教师模型在学生自己的历史上提供更好的目标分布。

学生在轨迹 B 上采样： $h_0 \rightarrow a_0 \rightarrow h_1 \rightarrow a_1 \rightarrow h_2 \rightarrow a_2 \rightarrow h_3$ （在此处选择下一个动作）。

学生输入：任务与历史 h_3 。教师输入：相同的任务与历史，再加上已验证的解法。

OPSD 使用起始模型的一份冻结副本作为教师。教师对下一动作的概率与之前一样用于训练学生。

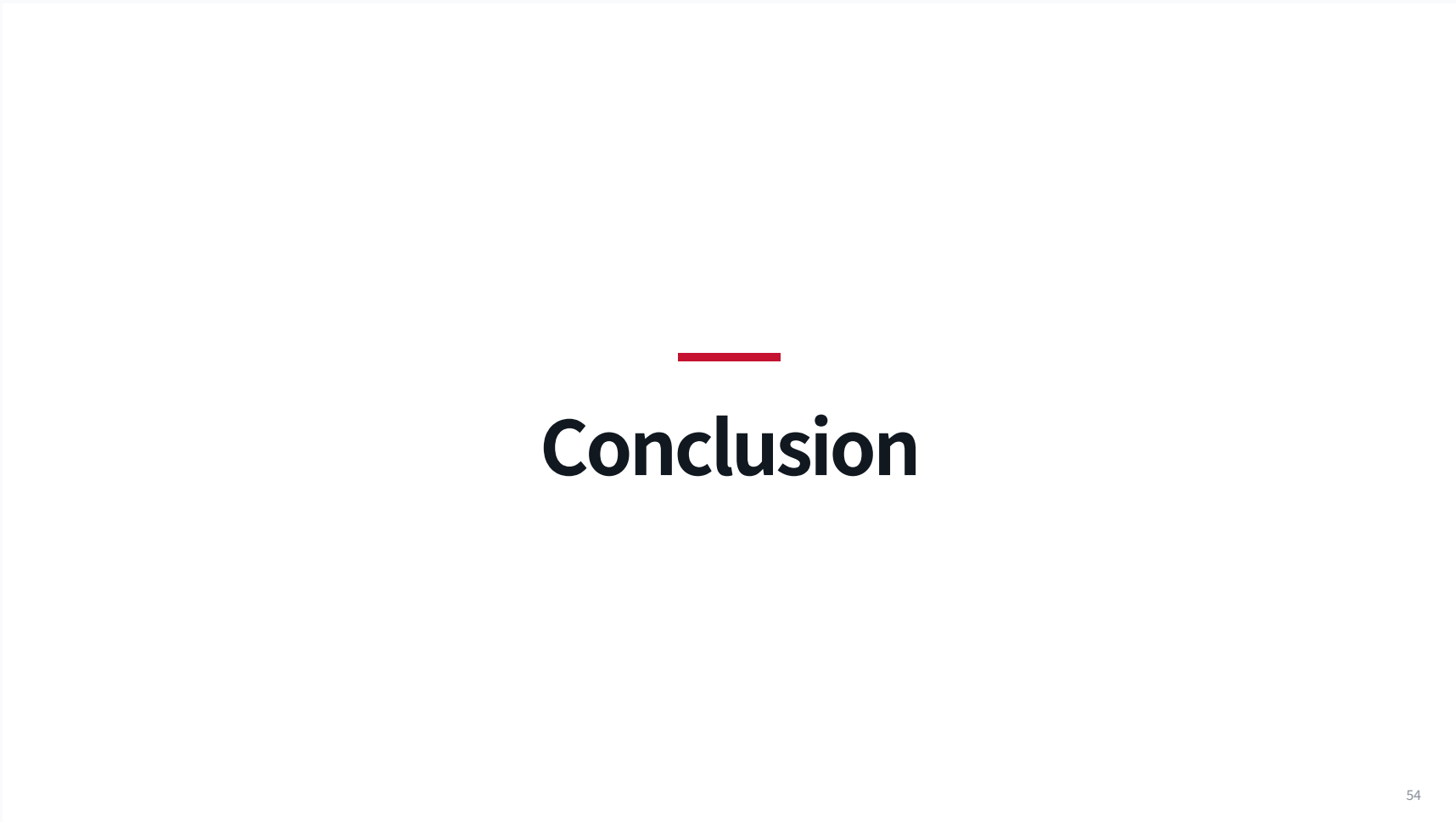
Self-Distilled Reasoner · Zhao 等, 2026 · OPSD ↗

在策略自蒸馏

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



结论

结论

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

RL takeaways

Useful feedback

Use group comparisons, a critic, or process rewards to assign credit.

Stable updates

Know the behavior policy; control how strongly samples change the model.

Reliable rewards

Match task difficulty and verify the behavior you actually want.

55

有用的反馈：使用组内比较、价值评估器或过程奖励来分配信用。

稳定的更新：了解行为策略；控制样本对模型的影响强度。

可靠的奖励：匹配任务难度，并验证实际想要的行为。

强化学习要点

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分



Questions?

Next Class: RL Systems

56

问题与答疑？

下节课：强化学习系统

问题与答疑

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Notation: the rollout

Symbol	Meaning
x, τ, T	Task; sampled trajectory; number of actions
h_t, a_t	History before step t ; action at that step
$u_t, u_{<t}$	Generated token at position t ; earlier context
π_θ, μ	Current policy; policy that generated the rollout
$p_\theta(\tau \mid x), p_\mu(\tau \mid x)$	Probability of the whole trajectory under each policy
$R(\tau), r_{t+1}, R_t$	Final reward; reward after a_t ; reward-to-go from h_t

57

记号 | 含义。

x 、 τ 、 T ：任务；采样轨迹；动作数量。

h_t 、 a_t ：步骤 t 前的历史；该步骤的动作。

u_t 、 $u_{<t}$ ：位置 t 生成的词元；此前上下文。

π_θ 、 μ ：当前策略；生成该采样轨迹的策略。

$p_\theta(\tau \mid x)$ 、 $p_\mu(\tau \mid x)$ ：在各自策略下整条轨迹的概率。

$R(\tau)$ 、 r_{t+1} 、 R_t ：最终奖励；采取 a_t 后的奖励；从 h_t 开始的后续回报。

译注 / 与原文翻译分开

记号保留原样。

记号：采样轨迹

第 11 讲 · 智能体的高级强化学习算法

EN / 英文官方原页 · 保持原样

ZH / 中文译文 · 第 1/1 部分

Notation: learning signals

Symbol	Meaning
$\hat{A}_t$	Estimated advantage: how an action compares with expectation
ρ_t, c_t	One-token policy ratio; its clipped value
$\rho_t^{cum}, w(\tau \mid x)$	Ratio through token t; ratio for the whole trajectory
$V^\mu(h), V_\phi(h)$	Expected reward-to-go under μ ; the critic's estimate
δ_t	Reward plus next value, minus current value
γ, λ	Reward discount (examples use $\gamma = 1$); weight on later errors in GAE
q, d_μ	Teacher's target; histories collected by the student

58

记号 | 含义。

$\hat{A}_t$: 估计优势，表示一个动作相对于预期的表现。

ρ_t, c_t : 单词元策略比率；裁剪后的比率值。

$\rho_t^{cum}, w(\tau \mid x)$: 截至词元 t 的比率；整条轨迹的比率。

$V^\mu(h), V_\phi(h)$: 在 μ 下的期望后续回报；价值评估器的估计值。

δ_t : 奖励 + 下一时刻的价值 - 当前价值。

γ, λ : 奖励折扣（示例中 $\gamma=1$ ）；GAE 中较晚误差的权重。

q, d_μ : 教师的目标分布；由学生收集的历史。

译注 / 与原文翻译分开

记号与变量保留原样。

记号：学习信号