Jingfeng

大语言模型后训练



数据、偏好、强化学习与蒸馏
从目标函数走向可验证的训练闭环

Jingfeng（Codex 辅助编写与修订）· 2026 年 10 月 22 日修订

一个后训练实验至少包含四个对象：模型要改变的行为、用来改变行为的数据、产生梯度的目标，以及判断改变是否有用的独立证据。本书围绕这四者展开。读者应能够从一条损失曲线追问到它背后的采样分布，并把一个看似合理的训练配方改写成能够被检验的假设。

本次修订以 2026 年 9 月 3 日的讲义为基础，完成全文技术审读、算例验证和 Jingfeng 书籍排版。来源核验截至 2026 年 10 月 22 日；这表示实际查阅的日期，不表示已经穷尽该日期以前的全部论文。原稿和原版 PDF 在项目中独立保留。

本书提供的证据

公式推导说明结论成立的条件；论文引用只支持相应作者、版本和实验条件下的论断；本次 CPU 结果来自合成教学策略。真实大模型训练、真实模型裁判与分布式 GPU 吞吐如未执行，均明确写为实验方案，不与本次实测混用。

导读：把目标、分布与证据连起来

本书面向理解条件概率、求导、向量运算和基本 Python 的读者。需要补充的知识是对数似然、交叉熵、期望与蒙特卡洛估计；策略梯度在第 5 章给出必要推导。完整的马尔可夫决策过程证明可结合《强化学习》分册阅读；LoRA 与量化微调的部署操作可结合《现代小模型微调》。

阅读路线与章节衔接

先读第 1–3 章，建立统一符号、数据契约和最小 SFT。只要模板、mask 或切分仍有问题，继续比较算法就很难解释。随后第 4–7 章从反馈模型走到 DPO、PPO、RLOO 与 GRPO；第 8–9 章讨论教师信号、搜索和轨迹再利用。最后四个环节是系统、评测、安全和可复现实验，分别对应第 10–13 章。第 14 章把这些能力组织成研究路线。

学习者可以先跑第 13 章的 CPU 检查，再回读其对应推导；开展研究的读者应先固定第 11 章的预算与保留集，再选择目标；工程实现者可将第 3、10 章的数据和数值不变量作为接入检查。附录给出额外推导、阅读问题和术语速查。

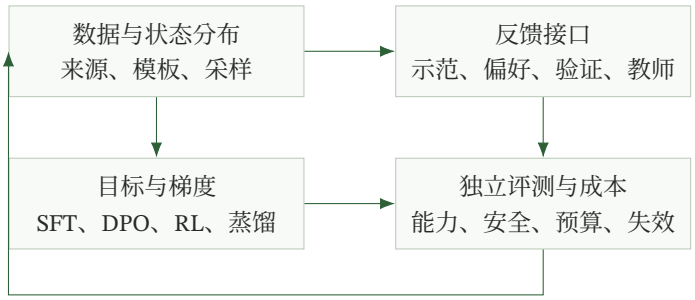


图 1 可组合的后训练闭环。箭头表示信息依赖；各方法可交替使用，不构成必须依次执行的阶段链。图由本书绘制。

全书符号和归一化约定

所有对数使用自然对数，信息量单位为 nat。若无另行说明，一个回答包含终止 EOS，序列对数概率为所有有效生成 token 对数概率之和；多轮工具返回是条件观测，不承担策略的动作似然。最大长度截断与模型主动输出 EOS 分开记录。

符号	含义与使用条件
$x \sim \mathcal{D}$	提示分布；训练、验证、测试和部署分布须分别命名。
$s_t = (x, y_{<t})$，$a_t = y_t$	无外部工具时的前缀状态与动作；工具任务另含环境观测。
π_θ，π_k	当前可训练策略、本轮冻结的采样快照；下标 k 是版本，不是时间步。
μ，d_μ	实际行为分布及其访问状态的分布；温度、top-p 可使 $\mu \neq \pi_k$。
π_{ref}，q_T	冻结参考模型、冻结教师；参考用于约束，教师用于提供监督，两者未必相同。
T_i，m_{it}	第 i 条完成的有效动作数、目标 token 的 loss mask；padding 不计。
R_i，r_t，$\hat{A}_{it}$	序列回报、单步奖励、优势估计；三者不能互换。
$\beta > 0$，$\tau > 0$	KL 系数、分布或偏好温度；有温度的 BT 模型在 DPO 中出现 β/τ。
G，ϵ	同提示组大小、比例裁剪范围；数值稳定常数另记 ε。
ρ_{it}	当前策略对行为动作概率的比率，必须写明分母与采样变换。

“序列平均”指先对一个完成求和或平均，再对完成平均；“token 平均”指全批有效 token 总损失除以其总数。每章在需要时明确二者。它们在变长数据上会给出不同目标，不能只用一个含混的 `mean()` 表达。

目录

导读：把目标、分布与证据连起来 **ii**

第 1 章 后训练的对象：行为、分布与目标 **1**
1.1 从“会续写”到任务策略 . . . 1
1.2 token、完成与环境轨迹 . . . 1
1.3 五种分布分别负责什么 . . . 2
1.4 KL 正则目标及其适用条件 . . . 2
1.5 KL 的方向、链式分解与长度 . . . 2

第 2 章 数据工程：把反馈变成可审计的样本 **4**
2.1 数据源是训练目标的一部分 . . . 4
2.2 schema 保存行为与测量过程 . . . 4
2.3 按题族切分，再对训练侧去污染 . . . 5
2.4 规则、模型和执行器分别测量什么 . . . 5
2.5 样本混合与 token 混合的换算 . . . 5

第 3 章 SFT：从条件似然到正确的训练张量 **7**
3.1 先定义哪个 token 承担损失 . . . 7
3.2 shift 与 mask 的逐 token 算例 . . . 7
3.3 chat template 是模型接口 . . . 7
3.4 packing 需要两种隔离 . . . 8
3.5 截断、加权与长度归一化 . . . 8
3.6 累积梯度时保持同一个分母 . . . 8
3.7 参数高效更新的边界 . . . 9

第 4 章 奖励、偏好与反馈测量 **10**
4.1 从成对比较到 Bradley–Terry 模型 . . . 10
4.2 软标签、排序与标注噪声 . . . 10
4.3 奖励模型、过程模型、验证器与裁判 . . . 11
4.4 训练奖励模型之前，先检验测量系统 . . . 11

第 5 章 策略梯度、GAE 与 PPO **13**
5.1 序列回报如何作用到每个 token . . . 13
5.2 冻结参考策略与 KL 奖励塑形 . . . 13
5.3 GAE：显式处理终止与时间截断 . . . 14
5.4 PPO：裁剪的是代理目标 . . . 14
5.5 RLOO：同一提示的留一 baseline . . . 15
5.6 可审计的训练循环与诊断 . . . 15

第 6 章 DPO 与离线偏好目标 **17**
6.1 第一步：解固定奖励下的 KL 正则问题 . . . 17
6.2 第二步：将奖励差代入偏好似然 . . . 17
6.3 第三步：从 token 得到完整序列概率 . . . 18
6.4 梯度、温度与标签噪声 . . . 18

6.5 从目标差异理解相邻方法 . . . 19
6.6 实现与实验应保存什么 . . . 19

第 7 章 可验证奖励与组相对优化 20
7.1 验证任务必须先给出契约 . . . 20
7.2 GRPO 的组内相对优势 . . . 20
7.3 写出一个与实现一致的代理目标 . . . 21
7.4 全同奖励组与长度偏置 . . . 21
7.5 KL 值的估计与 KL 梯度是两件事 . . . 22
7.6 采样分布、旧策略与 on-policy 条件 . . . 22
7.7 一致的训练伪代码与系统报告 . . . 23

第 8 章 蒸馏：在谁的前缀上学习什么 24
8.1 全称、接口与分布 . . . 24
8.2 先冻结前缀，再定义 token 目标 . . . 24
8.3 局部梯度与完整轨迹目标不能混同 . . . 25
8.4 词表、尾部概率与教师成本 . . . 26
8.5 在线偏好蒸馏：本书采用的具体闭环 . . . 27
8.6 何时值得做同策略蒸馏 . . . 27

第 9 章 合成数据、自训练与工具轨迹 28
9.1 筛选后再拟合：一个条件分布 . . . 28
9.2 Best-of-N、pass@N 与实际选择成功率 . . . 28
9.3 自训练、批评修订与闭环退化 . . . 29
9.4 工具与 Agent：训练对象是动作和恢复 . . . 29

第 10 章 训练系统：把概率契约落实到 rollout 与 learner 31
10.1 一条轨迹经过哪些组件 . . . 31
10.2 策略陈旧与采样变换 . . . 31
10.3 显存账本：权重只是其中一项 . . . 32
10.4 数值稳定性从不变量开始 . . . 32
10.5 恢复训练与成本日志 . . . 33

第 11 章 评测：固定问题、预算与测量系统 34
11.1 先写评测契约，再比较模型 . . . 34
11.2 指标覆盖的是哪一层能力 . . . 34
11.3 pass@k、搜索与实际选择 . . . 34
11.4 成对比较、tie 与统计单位 . . . 35
11.5 裁判要先校准，再用于归因 . . . 35
11.6 推理预算、费用与回归门槛 . . . 36

第 12 章 安全与规格：把代理奖励的边界纳入闭环 37
12.1 后训练安全是一项可测量的要求 . . . 37
12.2 把权限与拒答写进标注协议 . . . 37
12.3 reward hacking 是代理与目标的偏离 . . . 37
12.4 将红队发现变成回归样本 . . . 38
12.5 数据权利、隐私与发布证据 . . . 38

第 13 章 实验科学：从公式到可追踪证据 39
13.1 先规定要回答的问题 . . . 39

13.2 先用独立结果检查实现 39
13.3 实际运行的 CPU 训练消融 40
13.4 奖励与裁判偏差：检测器也需要真值 41
13.5 复现入口与真实模型升级方案 42

第 14 章 研究路线：把机制问题变成可证伪实验 43
14.1 从一个未知量开始 43
14.2 一个消融应改变哪件事 43
14.3 八周实践路线与交付标准 44
14.4 结论应能承受什么追问 44

附录 A 数学推导与估计器边界 45
A.1 KL 正则目标的闭式解与两个极限 45
A.2 序列 KL 的链式分解 45
A.3 KL 塑形为何可进入策略梯度 45
A.4 包括自身的组均值与留一均值 46
A.5 采样 KL 值与自动微分的反例 46
A.6 DPO 的重参数化与概率尺度 47

附录 B 阅读索引、术语与跨册指引 48
B.1 带着问题阅读原始资料 48
B.2 术语速查 48
B.3 本书与其他分册 49

参考文献 50

第 1 章

后训练的对象：行为、分布与目标

1.1 从“会续写”到任务策略

预训练学到下一个 token 的条件分布。后训练则用更有目的的数据与反馈，改变模型执行指令、回答问题、使用工具和结束任务的方式。继续预训练、监督微调、偏好优化、强化学习和蒸馏可以交替出现；阶段名称无法替代机制说明。InstructGPT 的示范、奖励模型和 PPO 流程是一个具体实现 [1]，并非所有模型都必须经过同一条流水线。

对一项后训练任务，先写下提示来自哪里、什么行为算成功、能取得什么反馈、允许多少生成与更新预算。代码任务可执行隐藏测试，开放问答可能只能获得带噪偏好，格式任务可以直接解析 schema。这些反馈的可重复性、覆盖范围和成本不同，会决定更合适的目标。

方法	数据与反馈	改变行为的方式及主要限制
SFT	固定示范 (x, y)	提高示范似然；不直接比较候选，依赖覆盖和模板。
拒绝采样后微调	生成候选与筛选器	对筛后文本做 SFT；训练分布被筛选事件重加权。
奖励建模	同提示的偏好对或评分	学习代理奖励；测量偏差会传到后续优化。
DPO 类	固定或周期刷新的偏好	直接优化相对似然；依赖偏好模型和参考支持。
在线 RL	策略轨迹与标量/过程奖励	用回报估计策略梯度；探索、方差、陈旧数据和投机需控制。
蒸馏	教师文本或概率分布	在指定前缀分布上迁移教师行为；容量与分词接口有约束。

1.2 token、完成与环境轨迹

给定提示 x，回答 $y = (y_1, \dots, y_T)$ 的自回归概率为

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}), \qquad \log \pi_\theta(y \mid x) = \sum_{t=1}^{T} \log \pi_\theta(y_t \mid s_t). \tag{1.1}$$

EOS 是一个动作，必须在模型主动结束的完成中计入。为避免无限长度的技术问题，可先固定有限词表和最大时域 H：提前 EOS 是终止，达到 H 是外部截断。截断样本所包含的前缀概率仍可计算，但它不是“模型在该处选择 EOS”的概率。训练和评测都要保存这一区别。

无工具时 $s_t = (x, y_{<t})$、$a_t = y_t$。有工具时，状态还含工具返回、环境版本和可见历史。环境观测不是模型采样的动作，不能把工具生成的长文本也加进策略似然。轨迹目标中只对策略实际决定的动作求和，工具执行耗时和失败则进入成本或回报。

在 softmax 模型中，大部分 token 原本就有正概率。文献里常说的“扩展支持集”通常指在有限采样预算内，让有用轨迹从几乎不可见变成可访问 [2]，不意味着从严格零概率凭空创造数学支持。对已经零概率且无法探索的动作，仅靠其缺席的回报不会提供学习信号。

1.3 五种分布分别负责什么

设当前学习器为 π_θ，本轮采样快照为 π_k，参考为 π_{ref}，教师为 q_T。真实行为策略 μ 还取决于采样器。若快照输出 logits z_k，温度为 τ_s，候选集合为 $S(s)$，则

$$\mu(a \mid s) = \frac{\mathbb{I}\{a \in S(s)\} \exp(z_k(a \mid s)/\tau_s)}{\sum_{v \in S(s)} \exp(z_k(v \mid s)/\tau_s)}. \tag{1.2}$$

没有裁剪支持、且 $\tau_s = 1$ 时才有 $\mu = \pi_k$。top-p、top-k、重复惩罚或工具约束都可能改变行为分布。存储“旧模型原始 log-prob”不必然等于存储真实采样 log-prob。

状态分布也要说明权重。令

$$\tilde{d}_\mu(s \mid x) = \mathbb{E}_{y \sim \mu} \sum_{t=1}^{T_y} \mathbb{I}\{s_t = s\}.$$

它的总质量为 $\mathbb{E}_\mu T$。若除以该值成为概率分布 d_μ，就对应跨轨迹 token 权重；先在每条轨迹内除以 T_y 再平均，会得到另一个状态分布。全书将局部期望所采用的权重与损失分母一起写出。

固定数据、学生前缀和教师前缀可以混合，例如在同一归一化约定下

$$d_k^{\text{mix}} = \lambda d_{\mu_k} + (1-\lambda) d_{\text{data}}, \quad 0 \le \lambda \le 1. \tag{1.3}$$

在线表示数据持续收集；*on-policy* 表示采样与被优化策略足够对应。异步数据虽在线，仍可能滞后。SFT 和离线蒸馏在固定数据上监督学习，可以具有分布偏移；不能仅因数据固定，就声称它们已采用某种 off-policy RL 校正。重要性采样是针对具体目标分布的推导，不是所有旧数据都应乘同一种 ratio。

1.4 KL 正则目标及其适用条件

考虑参考分布支持上的序列奖励 $r(x, y)$ 和 $\beta > 0$：

$$J(\pi) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi}[r(x, y)] - \beta \mathbb{E}_{x \sim \mathcal{D}} D_{\text{KL}}\big(\pi(\cdot \mid x) \| \pi_{\text{ref}}(\cdot \mid x)\big). \tag{1.4}$$

若归一化常数有限，且允许在参考支持上自由选择分布，则

$$\pi^*(y \mid x) = \frac{\pi_{\text{ref}}(y \mid x) e^{r(x,y)/\beta}}{Z(x)}, \quad Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) e^{r(x,y)/\beta}. \tag{1.5}$$

把目标改写成 $\beta \log Z(x) - \beta D_{\text{KL}}(\pi \| \pi^*)$ 即得证明。有限参数模型未必表示得出 π^*，优化也未必找到它，所以闭式解是理解约束的工具，而非训练收敛保证。

一个能手算的最优分布

参考对两个完成各给 1/2，奖励为 $(0, \log 3)$，$\beta = 1$。未归一化权重为 $(1/2, 3/2)$，故最优概率为 $(1/4, 3/4)$。奖励乘以 2 且 β 也乘以 2，不改变最优分布。若只改变其一，目标便改变。这解释了比较奖励归一化与 KL 系数时为何要一起记录尺度。

当奖励有限、可行集合固定，$\beta \to \infty$ 时最优分布趋近参考；$\beta \downarrow 0$ 时集中到参考支持中最高奖励的完成。一个代理奖励若偏爱无效长文，降低 β 也会更强地偏爱它。KL 约束控制漂移，并不能校正奖励的含义。

1.5 KL 的方向、链式分解与长度

定义 $D_{\text{KL}}(P\|Q) = \sum_z P(z) \log[P(z)/Q(z)]$。若 $P = (0.8, 0.2)$、$Q = (0.5, 0.5)$，则 $D_{\text{KL}}(P\|Q) = 0.192745$，反向为 0.223144，单位 nat。方向决定在哪个分布下平均，不能用一个未标方向的“KL”替代。

对同一提示、相同终止约定的策略，序列 KL 满足链式分解：

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \mathbb{E}_{y\sim\pi_\theta} \sum_{t=1}^{T_y} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s_t) \| \pi_{\mathrm{ref}}(\cdot \mid s_t)\big). \tag{1.6}$$

证明是将式 (1.1) 的对数比展开，再在前缀条件下对动作取期望。工具环境若在两种策略下有相同的条件转移核，该核在轨迹比值中抵消；环境或观测协议若变化，就不能只比较 token 概率。

式 (1.6) 是求和。把每条回答除以随机长度 T_y，优化的是长度加权的另一个目标；跨批 token 平均也不同于期望序列 KL。采样 log-ratio 的单点可以为负，而完整离散分布的 KL 非负。第 7 章和附录 A 将进一步区分 KL 数值估计与其梯度。

练习与提示

1. 参考为 $(0.9, 0.1)$，奖励仍为 $(0, \log 3)$，求 $\beta = 1$ 时最优策略。提示：先计算未归一化权重，而非直接对奖励做 softmax。
2. 构造两个长度分别为 1 和 3 的完成，说明序列和 token 平均如何改变它们的相对权重。提示：分别写出分母 2 和 4。
3. 若 top-k 采样删除了当前策略仍给正概率的动作，为什么动作 ratio 无法还原完整目标？提示：从重要性采样的支持覆盖条件出发。

第 2 章

数据工程：把反馈变成可审计的样本

2.1 数据源是训练目标的一部分

同一个 DPO 损失，在礼貌风格偏好、代码正确性偏好和安全拒答偏好上会学到不同的行为。数据不只是算法的输入，还定义了训练分布与“更好”的含义。Tülu 3 等公开后训练配方的价值之一，是把数据、目标和评测作为相互依赖的系统呈现 [3]；不能将某一数据配比脱离模型和任务直接当作常数使用。

示范可来自人工、授权历史交互、领域材料或教师生成。偏好可来自人类比较、模型裁判或规则验证。每个来源都要回答：谁有权提供，允许怎样使用，采集时采用什么协议，处理之后还剩什么选择偏差。公开网页能访问，只说明可获取，不能据此推断训练、再分发和商用许可。

证据对象	至少保存的字段	为什么影响训练
来源与权限	原链接/内部 ID、获取日期、许可文本版本、授权范围、再分发限制	同一站点或仓库不同部分可有不同许可；未知项须追查或排除。
生成过程	模型与 revision、模板 hash、采样设置、提示来源、生成时间	合成样本可能只覆盖教师风格，且候选分布决定筛选偏差。
质量与标签	标注协议、分歧、置信度、tie、裁判版本、过滤原因	把不确定标签硬化会造成虚假的确定性。
变换与切分	原始/规范化 hash、父样本 ID、group ID、split、变换版本	改写、翻译、同题多解应继承同一组，避免泄漏。

2.2 schema 保存行为与测量过程

SFT 数据保留消息角色、工具定义和轮次，而非只保存拼好的字符串。偏好数据保留两个原始候选的顺序和标签概率，避免在排序后丢失位置偏差。结果奖励保留验证器命令、版本、退出码和错误类型；过程反馈保留步骤边界及其可见上下文。工具轨迹另外记录权限、超时、重试和环境快照。

下面是一条合成教学样本，其中许可证仅指这条本地构造内容，不声称任何外部数据集使用该许可：

Listing 2.1 可追溯偏好记录的最小结构

```
{"id":"toy-17", "group_id":"sum-17", "split":"train",
 "prompt":"Compute 2+3.", "candidate_1":"5", "candidate_2":"6",
 "p_candidate_1_preferred":1.0, "tie":false,
 "feedback":{"kind":"exact_integer", "version":"toy-v1"},
 "origin":{"kind":"locally_generated", "license":"CC0-1.0"},
 "generation":{"policy_version":"toy-sft-v1", "temperature":1.0},
 "transforms":["integer_parser_v1"]}
```

真实数据还应保存可核对的许可凭证位置和处理日志；未知字段写成显式状态，不用一个笼统的 `clean=true` 代替审计。隐私处理的原始映射若必须保留，应放在受控位置，不随训练文本与模型包公开。

2.3 按题族切分，再对训练侧去污染

随机按行切分会把同一提示的两种答案分到两边，也会把同题翻译、教师改写和多轮续写分开。模型此时只需记住题目，就可能在测试集获得高分。正确的切分单位应接近希望检验的泛化单位：题目、文档、用户、项目或环境模板。

一个可执行的流程是先建立父子关系和近重复候选图，将同题多解、翻译和改写归入同组，再将组分给 train/validation/test。独立冻结保留集；比较训练数据与保留集的精确及近重复，将冲突从训练侧移除或隔离，记录删除数量。不要为了训练方便修改测试集组成，也不要让验证器的隐藏测试回流进训练文本。

精确 hash 只能发现相同规范化字符串。MinHash/LSH 或 n-gram 可筛选近重复，embedding 检索可补语义候选，但相似不等于泄漏：两道题可能共享固定格式，却要求不同推理。保存阈值与人工复核样本，分别报告误报与漏报的迹象。过强的规范化若去掉数字、变量和代码标点，还会把不同题错误合并。

时间切分也有条件。按文档发布日期分组，可减少后出材料回流；但抓取日期不等于内容首次发表日期，后出的汇编也可能包含旧题。模型预训练语料往往不可见，因此本书的后训练污染审计只能排查本次可见数据，不能证明基座从未见过测试题。

一个小型泄漏案例

原始记录 A 是“2+3 等于多少”，B 是它的英文翻译，C 是以 A 为用户首轮的多轮对话。三条记录的字符串 hash 不同，但都属于同一题族。若 A 在训练集、B 在测试集，所谓跨语言成功可能来自题目记忆。令三条记录共享 `group_id` 并一起切分，再用未出现的数字组合测试，才能更接近问题要求。

2.4 规则、模型和执行器分别测量什么

规则适合识别空文本、角色顺序错误、超长、乱码和明显重复；模型过滤适合相关性、语言或偏好评分；执行器适合可解析格式、单元测试和可验证答案。这些检查测量的属性不同，并不自动相互独立。两个同源裁判一致，可能只是共享了同一种偏差。

过滤应记录进入数、通过数、每个失败原因和交集。若先用奖励模型筛掉全部短回答，再声称训练集中的长度与质量高度相关，就把选择规则当成了证据。保留小份未过滤、分层人工复核数据，检查筛选器是否排除了少数语言、简洁正确答案或困难但有价值的轨迹。

过程监督还需记录标注者看到了什么。若步骤标签使用最终答案倒推，模型在仅有前缀时可能无法获得同样的信息；若验证器只检查最终值，就不能把通过样本中的每一步都标为正确。第 4 章会区分这些监督对象。

2.5 样本混合与 token 混合的换算

设源 j 的有效 token 总数为 n_j，可定义名义源采样概率

$$p_j = \frac{n_j^{\alpha}}{\sum_{\ell} n_{\ell}^{\alpha}}, \quad 0 \le \alpha \le 1. \tag{2.1}$$

这只是一个选择，$\alpha < 1$ 会提高小源相对权重。若按该概率抽样本，平均有效长度为 $\bar{T}_j$，实际 token 占比近似为

$$\omega_j = \frac{p_j \bar{T}_j}{\sum_{\ell} p_{\ell} \bar{T}_{\ell}}.$$

两个源平均长度为 100 和 1000，样本各占一半时，token 比例约为 1 ： 10。若希望 token 各占一半，样本概率应接近 10/11 和 1/11。截断、空标签和 packing 会进一步改变有效长度，最后应以训练器实际消费的 token 日志复核。

若使用先样本平均的目标，每个样本名义权重相同；若使用全批 token 平均，长样本通常贡献更多项，但实际梯度大小还取决于预测误差和相关性。不能仅凭 token 数断言精确的梯度贡献。建议同时保存源级样本数、有效 token、重复曝光次数、截断率和损失。

动态采样可把预算投入不确定或容易失败的提示，但这会改变数据分布。用训练侧诊断决定采样，验证集用于少量预先规定的选择；保留测试集不参与筛选。主动标注时还应保留容易样本和分层随机样本，用来估计整体表现，避免只在模型意见分歧的区域定义“真实偏好”。

练习与提示

1. 为一份包含翻译和教师改写的指令集定义题族切分。提示：先画父样本关系，再给每个连通组一个 split。
2. 某源名义占 20% 样本，平均有效长度是其他源的 4 倍，求 token 占比。提示：分子为 0.2×4，分母还包括 0.8×1。
3. 构造一个高质量过滤器错删简洁答案的案例，给出未过滤审计样本的用途。提示：质量标签不能由待检验的长度规则本身生成。

第 3 章

SFT：从条件似然到正确的训练张量

3.1 先定义哪个 token 承担损失

监督微调（supervised fine-tuning，SFT）在示范前缀上预测下一个 token。令 $z_{i,0:L_i-1}$ 为第 i 条已渲染对话，$m_{i,t}$ 表示位置 t 的 token 是否作为监督目标，则

$$\mathcal{L}_{\mathrm{SFT}} = -\frac{1}{N}\sum_{i}\sum_{t=1}^{L_i-1} m_{i,t}\log \pi_\theta(z_{i,t}\mid z_{i,<t}), \quad N=\sum_{i}\sum_{t=1}^{L_i-1} m_{i,t} > 0. \tag{3.1}$$

位置 0 没有前一个预测位置，通常是 BOS 或首个条件 token。completion-only 指只让指定回答 token 承担损失；在多轮对话中，可以监督所有 assistant 轮，也可以只监督最后一轮，二者应在数据卡中区分。user、system 和工具返回仍参与条件上下文，mask 为零不表示它们从注意力中消失。

给定前缀，负对数似然的 logits 梯度是 $p_v - \mathbb{I}\{v = z_t\}$。因此错误地把用户文本设为标签，会推动模型重现用户文字；错误地漏掉结束标记，会丢失主动结束的监督。loss 下降只证明模型拟合了实际送入的目标，不能证明 mask 表达了任务意图。

3.2 shift 与 mask 的逐 token 算例

考虑教学用序列，不对应任何特定模型的真实特殊 token：

位置 t	0	1	2	3	4	5
token	BOS	U	q	A	a	EOS
目标 mask m_t	0	0	0	0	1	1

预测位置 3 的 logits 预测位置 4 的 `a`；位置 4 的 logits 预测位置 5 的 EOS。所以计算张量是 `logits[:-1]` 对 `input_ids[1:]`，并乘 `mask[1:]`。把未平移的 mask 直接乘在预测位置上，会把一个回答目标错移到另一个 token。

若这两个正确目标的概率为 1/2 和 1/4，则

$$\mathcal{L}_{\mathrm{completion}} = \frac{-\log(1/2)-\log(1/4)}{2} = \frac{\log 8}{2} = 1.03972077084.$$

若误把概率为 0.1 的一个用户 token 也设为目标，平均损失变成 $\log 80/3 \approx 1.460675$，梯度方向也包含了拟合用户文本的要求。修正学习率不会消除这种任务污染。本书 CPU 检查直接使用这组概率及独立手算值，见第 13 章。

多轮时尤其要检查 assistant 结束 token、工具调用参数与工具返回的边界。工具调用由模型决定的参数通常可监督；工具执行器返回的内容是条件观测。某些模板把轮次结束和整段结束设计为不同 token，要以 tokenizer 与模板版本为准，不能一律插入一个通用 EOS。

3.3 chat template 是模型接口

消息结构到 token 串的映射由 chat template 决定。相同文本用不同的角色控制 token 渲染，模型看到的是不同输入。官方 Transformers 文档强调使用模型匹配的模板，并注意模板已经包含特殊 token 时避免重复添加 [4]。这是一项接口要求，不是所有模型共用同一串角色标记。

模板错误如何进入训练

假设训练样本在每个回答前加入 `A`，推理却把用户结尾直接接到模型输出；模型可能先生成角色标记，或者继续用户发言。另一个常见错误是先渲染含 BOS/EOS 的文本，再次分词时又自动添加同样的特殊 token。两种情况下 token 计数与 loss mask 都可能看起来合法，但上下文协议已经改变。应保存同一条消息的训练、评测渲染结果，并比较生成起始位置。

最小接入检查包括消息往返、目标可视化和边界反例。消息往返检查角色顺序和控制 token；目标可视化逐个打印 token ID、解码文本、角色、mask 和其预测位置；反例检查空回答、最后一轮只有工具返回、含模板分隔符的普通文本和超过上下文长度的样本。字符串解码会隐藏某些特殊 token 或规范化空格，因此还要直接比较 token ID，不能只看解码后的句子。

3.4 packing 需要两种隔离

packing 把多条样本放入一个物理序列，减少 padding。若希望保持“独立样本的 SFT”语义，至少要保证：第二条样本看不到第一条的内容；第一条末尾不承担预测第二条开头的目标；位置编码的重置或连续使用与所采用模型和实现保持一致。

仅插入 EOS 并不构成注意力隔离。普通因果 mask 仍允许第二条读到第一条。可用分段的块对角因果注意力或支持独立序列边界的等价内核。标签 mask 还要独立检查：切断注意力，并不会自动删除跨边界的 next-token 标签。

一个有效测试是固定样本 B，把与之同包的 A 替换成完全不同的 A′。在关闭 dropout、使用相同位置约定且隔离正确时，B 的 logits 和损失应在数值误差内不变。若实现主动采用跨样本上下文，目标已发生改变，应命名为该设置，并用匹配的推理环境验证。

3.5 截断、加权与长度归一化

截断可能保留完整提示却删除全部回答。这样的样本 $N_i = 0$，不应通过 `clamp_min(1)` 静默变成一次零损失更新；应记录并过滤，若全批无目标则跳过更新并报警。对部分回答截断，不要把外部截断伪装成真实 EOS；可保留有效前缀监督，也可按任务规则丢弃，但要报告比例。对代码和工具调用，截断还可能破坏语法与可执行性。

带样本权重 w_i、token 权重 u_{it} 和软目标 q_{itv} 的形式为

$$\mathcal{L} = -\frac{1}{N_w} \sum_{i,t} w_i u_{it} m_{it} \sum_{v} q_{itv} \log \pi_\theta(v \mid z_{i,<t}), \quad N_w = \sum_{i,t} w_i u_{it} m_{it} > 0. \tag{3.2}$$

label smoothing 可令 $q = (1 - \varepsilon_{\text{ls}})\delta_{z_t} + \varepsilon_{\text{ls}}/|\mathcal{V}|$，它改变监督分布，不是纯粹的数值稳定技巧。样本权重、温度混合与重复曝光也会共同改变目标，应避免无意重复加权。

设两个回答分别含 1 和 3 个有效 token，总 NLL 为 1 和 6。全批 token 平均为 $7/4 = 1.75$；先对样本平均再平均为 $(1 + 2)/2 = 1.5$。前者对应每个有效 token 同权，后者对应每条完成同权。DPO 的序列对数概率则必须求和，不能因为 SFT 用 token 平均就沿用同样的序列得分，详见第 6 章。

3.6 累积梯度时保持同一个分母

若各 microbatch 有效 token 数不同，把每个 microbatch mean 除以累积步数，会变成 microbatch 同权。要实现式 (3.1)，应以整个优化窗口的有效 token 总数归一化。单进程概念代码如下；大窗口可先统计长度或保存各分子梯度，再统一缩放，不要求同时保存全部计算图。

Listing 3.1 按优化窗口有效 token 平均的 SFT

```
for window in accumulation_windows(loader):
    N = sum(count_shifted_targets(b) for b in window)
    if N == 0:
        record_invalid_window(); continue
    optimizer.zero_grad()
    for b in window:
        logits = model(b.input_ids, attention_mask=b.block_mask)
        nll = cross_entropy(logits[:, :-1].transpose(1, 2),
                            b.input_ids[:, 1:],
                            reduction="none")
        numerator = (nll * b.target_mask[:, 1:]).sum()
        (numerator / N).backward()
    clip_grad_norm_(trainable_parameters, max_norm)
    optimizer.step(); scheduler.step()
```

在 W 个进程的 DDP 中，若梯度默认做进程平均，而目标是全局 token 平均，各进程的分子需乘 W/N_{global}。若通信采用求和，则不乘 W。这是归一化约定的推导，实际框架还可能自行缩放，接入时用不等长两批的独立参考梯度验证，避免重复缩放。

log-softmax 与长序列 log-prob 的归约可在 float32 中完成，避免先求概率乘积再取对数。bf16 仅在硬件与算子支持时使用；CPU 算例用 float64 做参考。SFT loss 下降后仍要看独立任务、结束率、长度、模板有效率和基础能力保留，不能用训练似然替代部署指标。

3.7 参数高效更新的边界

LoRA 冻结 W_0，训练低秩分量 [5]：

$$W = W_0 + \frac{\alpha}{r} BA, \quad A \in \mathbb{R}^{r \times d_{\text{in}}},\ B \in \mathbb{R}^{d_{\text{out}} \times r}. \tag{3.3}$$

它减少可训练参数、梯度和优化器状态，却仍需要基座前向与激活。QLoRA 将冻结基座量化存储，并用较高精度计算适配器更新；NF4、双重量化和 paged optimizer 是其具体配方的一部分 [6]。不能把“4 bit 存权重”等同于整个训练每参数只占半字节。

比较全参、LoRA 和 QLoRA，记录可训练模块、rank/缩放、量化格式、计算精度、有效 token 和 checkpoint 选择规则。学习率不必相同，但搜索预算应透明。更详细的低资源实现见《现代小模型微调》；本书接下来关注目标与反馈，显存账本则在第 10 章展开。

练习与提示

1. 为两轮对话画出所有 assistant token 的目标位置。提示：mask 标记的是被预测 token，logits 位于它前一位置。
2. 两个 microbatch 分别含 2、6 个有效 token，推导其 NLL 分子的归一化系数。提示：目标分母是 8，而非两个 microbatch 各自 mean 后再除以 2。
3. 设计能同时检测注意力泄漏和跨样本标签泄漏的两个测试。提示：前者替换同包前文，后者只检查边界处的目标索引。

第 4 章

奖励、偏好与反馈测量

SFT 告诉模型“这段文字值得模仿”，却没有直接回答两个候选谁更好。本章把反馈看成需要校准的测量接口：人类比较、神经奖励模型、过程标签、程序验证器和模型裁判，观察的对象与错误模式各不相同。学会写出它们的似然和误差，比先选择一个奖励网络更重要。

4.1 从成对比较到 Bradley–Terry 模型

对同一提示 x 的两个回答 y_a, y_b，设奖励模型为 $r_\psi(x, y)$。Bradley–Terry（BT）模型写成

$$p_\psi(a \succ b \mid x) = \sigma(h_\psi), \qquad h_\psi = \frac{r_\psi(x, y_a) - r_\psi(x, y_b)}{\tau_{\text{pref}}}, \qquad \sigma(z) = \frac{1}{1 + e^{-z}}. \tag{4.1}$$

$\tau_{\text{pref}} > 0$ 表示偏好随机性的尺度。常见实现取 $\tau_{\text{pref}} = 1$，把它吸收到奖励单位；这是一种约定，不是标注者毫无噪声。BT 只用一个标量描述每个回答的潜在效用，不能表达所有循环偏好、群体差异和上下文依赖。InstructGPT 与 DPO 均以这种比较模型构造偏好似然 [1, 7]。

令 $z \in \{0, 1\}$ 表示是否偏好 a，每对样本的负对数似然为

$$\ell_{\text{BT}}(\psi) = -z \log \sigma(h_\psi) - (1 - z) \log \sigma(-h_\psi). \tag{4.2}$$

将已排序的 (y^+, y^-) 总是放在同一顺序时，$z = 1$，损失为 $-\log \sigma(\Delta r_\psi / \tau_{\text{pref}})$；其中 $\Delta r_\psi = r_\psi(x, y^+) - r_\psi(x, y^-)$，在 $\tau_{\text{pref}} = 1$ 的单位约定下可省去温度。但数据应保留原始呈现顺序、标注者和交换实验结果，不能让“第一列”成为标签捷径。

奖励的识别有两个限制。第一，给同一提示的所有回答加 $c(x)$，偏好概率不变；第二，同时把 r_ψ 和 τ_{pref} 乘以同一个正数，概率也不变。因此原始奖励没有天然的零点或跨模型可比单位。把一个 checkpoint 的均分从 3 提高到 6，不能独立证明行为变好；要在固定比较集、固定尺度和独立任务上重新测量。

一个可以手算的偏好标签

若四名标注者中三人偏好 a，先把经验频率记为 $p = 0.75$。在单对、无正则且可自由拟合的情形，最优 BT logit 为 $h^* = \log[p/(1-p)] = \log 3 \approx 1.0986$。若 $\tau_{\text{pref}} = 2$，对应奖励差为 $2 \log 3$，但预测概率仍为 0.75。这个算例显示，偏好概率相同不意味着奖励差有共同单位。

4.2 软标签、排序与标注噪声

用软标签 $p \in [0, 1]$ 代替硬标签，可得

$$\ell_{\text{soft}}(h; p) = -p \log \sigma(h) - (1 - p) \log \sigma(-h), \qquad \frac{\partial \ell_{\text{soft}}}{\partial h} = \sigma(h) - p. \tag{4.3}$$

这里的 p 可以是多次独立比较的投票频率。模型自报“信心 0.9”未经校准，不能直接当作 90%正确概率；置信权重也不同于软标签：$w\ell(h; 1)$ 改变样本强度，$\ell(h; p)$ 改变目标概率。二者均需保存来源和计算方式。

平局还要分清“质量接近”“双方都差”和“无法判断”。只有确实表达等概率偏好时，$p = 0.5$ 才

是合适的 BT 标签；缺失或不适用标签应单独处理。若硬标签以概率 $\eta < 0.5$ 独立翻转，观测概率为

$$p_{\text{obs}} = \eta + (1 - 2\eta)p_{\text{true}}. \tag{4.4}$$

它把概率推向0.5, 系统性的长度或立场偏差却不满足这种对称噪声模型。使用一个“label smoothing”参数并不能自动修复所有标注偏差。

对 K 个回答的严格排序 σ，一种扩展是逐次选择的 Plackett–Luce 似然：

$$\ell_{\text{PL}} = -\sum_{j=1}^{K-1} \log \frac{\exp(r_\psi(x, y_{\sigma(j)})/\tau_{\text{pref}})}{\sum_{l=j}^{K} \exp(r_\psi(x, y_{\sigma(l)})/\tau_{\text{pref}})}. \tag{4.5}$$

最后一项概率为 1，故省去。把一个排序展开成全部成对标签更方便，但这些标签共享同一提示和同一次排序，统计分析时不能把它们当成完全独立样本。BT 和排序模型的对应关系可结合 DPO 的偏好建模部分阅读 [7]。

4.3 奖励模型、过程模型、验证器与裁判

接口	实际输出	能说明什么，以及不能说明什么
奖励模型（RM）	输入 (x, y) 后输出标量或分布	学到标注分布上的偏好代理；分数本身不保证事实正确。
结果奖励模型（ORM）	完整答案的预测分数	估计终局质量；长轨迹的信用分配通常稀疏。
过程奖励模型（PRM）	步骤或前缀的局部评分	提供过程监督；依赖步骤切分、标签定义与聚合规则。
程序验证器（verifier）	可执行检查的通过、失败或诊断	对写入规范的性质给出可重复反馈；不代表规范覆盖真实任务。
模型裁判（judge）	成对偏好、等级、评语或概率	执行 rubric 的另一模型；可能有位置、长度、风格和自偏好。

PRM 的代表性工作使用人类步骤标签来识别数学解答的局部正确性 [8]。原始过程分数如何组成序列奖励，必须再给一个定义。例如把方法所定义的逐步奖励 r_t 相加：

$$R(x, y) = \sum_{t=1}^{T} r_t(x, y_{\le t}) + r_{\text{terminal}}(x, y). \tag{4.6}$$

这只是奖励构造；不能默认把每步“正确概率”直接相加，就得到真实终局回报。十步各 0.9 的分数乘积是约 0.349，而一步 0.9 的乘积是 0.9；求和又分别为 9 和 0.9。两种聚合都引入长度效应。若局部事件相依，概率乘积还缺少所需的条件概率解释。应同时报告步骤切分规则、加总或最小值等聚合方式、长度切片及最终任务成功率。

验证器的边界同样具体。JSON 语法合法不等于字段含义正确；公开单元测试通过不等于隐藏测试通过；数学答案解析器可能接受夹带错误推导的最终数字。把解析、资源限制、异常处理、隐藏测试和人工抽查作为反馈契约的一部分，才能避免模型逐渐优化验证器的漏洞。

4.4 训练奖励模型之前，先检验测量系统

奖励模型训练集与测试集应按提示及近重复簇切分，再保存候选的生成模型、采样温度、答案长度、角色模板和标注版本。只在同一 SFT 模型的输出上测试，不能说明 RM 能正确评价后续 RL 产生的域外文本。至少保留跨模型候选、长短匹配对、困难负例和新领域提示。

评价分两层。成对准确率衡量排序；若把输出解释为概率，还要报告 Brier 分数或可靠性图，并说明分桶和样本量。再用独立结果验证器、人类复核和反事实改写检查代理是否稳定。关注 reward–length 相关性很有用，但相关性本身不能证明因果：复杂任务本来就可能需要长答案，应在同一内容、同一提示的配对扰动上定位捷径。

模型裁判的常见偏差在 MT-Bench/Chatbot Arena 研究中已有分析；长度控制评估给出了另一种统计校正思路 [9, 10]。这些工作支持开展偏差检查，不保证一次位置交换或一次回归就消除了偏差。

三组反事实审计

固定正确性及核心内容，构造以下对照，并保留原文本：

1. 将同一对候选的 A/B 位置交换，再把输出映射回候选身份；比较两次偏好是否一致。
2. 给其中一个候选加入不改变答案的冗余解释，检查分数是否只因长度而上升。
3. 将相同内容改为列表、普通段落或带标题的格式，检查 rubric 是否受表面结构驱动。

同时随机化候选身份，固定 judge 版本与采样配置，并用独立正确性标签确认变换没有改变质量。第 13 章提供合成偏差接口的实测与真实裁判的可执行方案；二者的证据范围不同。

练习与提示

1. 为什么给 RM 加 $c(x)$ 不改变偏好？提示：先写同一提示下的奖励差，再说明跨提示奖励均值为何不由比较数据确定。
2. 设真实偏好概率为 0.8、独立标签翻转率为 0.1。计算观测概率与 logit。提示：先用式 (4.4)，结果为 0.74 而非 0.7。
3. 为代码任务选择 ORM、PRM 和隐藏测试的组合。提示：逐项列出观测对象、可复算日志和漏检风险，不以一个平均分代替三种接口。

下一章先固定反馈接口，再研究如何把标量回报变成策略更新。奖励测量错误与梯度估计错误必须分别诊断。

第 5 章

策略梯度、GAE 与 PPO

人类反馈强化学习（reinforcement learning from human feedback，RLHF）将偏好转成奖励，再用策略优化改变模型的生成分布。InstructGPT 展示了示范 SFT、偏好奖励建模、PPO 优化的三阶段流程 [1]。三阶段分别引入示范覆盖、反馈误差和优化误差；只有把这些对象分开，训练失败才可定位。本章从无裁剪的策略梯度开始，逐步加入 baseline、时序信用分配和 PPO 代理目标。

5.1 序列回报如何作用到每个 token

先考虑固定提示 x、与参数无显式关系的序列回报 $R(x,y)$，并令 $y \sim \pi_\theta$。由对数导数恒等式，

$$\nabla_\theta \mathbb{E}_{y\sim\pi_\theta}[R(x,y)] = \sum_y R(x,y)\nabla_\theta \pi_\theta(y \mid x) \tag{5.1}$$

$$= \mathbb{E}_{y\sim\pi_\theta}\left[R(x,y)\sum_{t=1}^{T_y}\nabla_\theta \log \pi_\theta(y_t \mid s_t)\right]. \tag{5.2}$$

完整答案的奖励乘在整条轨迹的 score function 上。终局奖励能影响早期 token，但没有指出到底哪一步造成成功，所以方差可能很大。若存在逐步奖励，可将 R 换成从 t 开始的 reward-to-go；动作之前已经确定的奖励，其 score 期望为零。

为降低方差，引入状态 baseline $b(s_t)$。只要给定 s_t 后它不依赖当前动作，并在计算策略梯度时停止对 baseline 反传，

$$\mathbb{E}_{a_t\sim\pi_\theta(\cdot|s_t)}[b(s_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)] = b(s_t)\nabla_\theta \sum_a \pi_\theta(a \mid s_t) = 0. \tag{5.3}$$

因此用 $R_t - b(s_t)$ 替换 R_t 不改变该期望。critic 可用另一项回归损失训练；若与策略共享参数，策略损失中的优势仍须 detach，否则会多出不属于上述恒等式的梯度。

“baseline 不依赖动作”是条件性要求，不是代码中变量名的要求。把同一回答自己的奖励直接加入 baseline，一般破坏条件独立；在同一 batch 上拟合且过拟合的 baseline，也不能仅凭 detach 就保证无偏。后文的留一设计避免最直接的自包含问题。

5.2 冻结参考策略与 KL 奖励塑形

第 1 章的 KL 正则目标，在当前策略采样时可写成

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r(x,y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}\right]. \tag{5.4}$$

这里 π_{ref} 是冻结参考模型，$\beta > 0$ 是奖励单位下的 KL 系数。若回报固定且支持条件满足，正则项中显式求导产生的 $-\beta\mathbb{E}[\nabla \log \pi_\theta]$ 为零，因此可用正则化回报乘 score 估计梯度。这不意味着任意缓存的 KL 数值都具有相同梯度。

实际训练先冻结采样快照 π_k，本节暂设行为策略 $\mu = \pi_k$。在采样 token 上缓存

$$\tilde{r}_t = r_t - \beta\left[\log \pi_k(y_t \mid s_t) - \log \pi_{\text{ref}}(y_t \mid s_t)\right], \tag{5.5}$$

通常将 RM 的序列奖励放在最后一个有效 completion token，其他 $r_t = 0$。缓存的塑形回报在本轮更新中视为固定。它在采样策略附近提供局部更新方向；重用很多轮后，不能继续把它称为当前策略完整目标的精确梯度。

式 (5.5) 中的单个 log-ratio 可能为负，完整词表 KL 则非负。序列 KL 对应含 EOS 的 token 和；先除以随机长度再平均，会改变统计量。参考策略用于约束相对原行为的漂移，π_k 用于描述谁生成了数据；它们即使初始权重相同，也不是同一个概念。

5.3 GAE：显式处理终止与时间截断

令 $V_\phi(s_t)$ 预测从当前状态开始的回报，c_t 表示本步之后是否允许继续 bootstrap：真实终止时为 0，否则为 1。以旧 value 快照计算

$$\delta_t = \tilde{r}_t + \gamma c_t V_{\text{old}}(s_{t+1}) - V_{\text{old}}(s_t), \tag{5.6}$$

$$\widehat{A}_t = \delta_t + \gamma\lambda c_t \widehat{A}_{t+1}, \qquad \widehat{A}_{T+1} = 0. \tag{5.7}$$

这就是广义优势估计（generalized advantage estimation，GAE）的反向递推 [11]。若一条回答在第 T 步 EOS 终止，$c_T = 0$；若只是采样时间窗结束且任务本可继续，则 $c_T = 1$，使用可用的 $V_{\text{old}}(s_{T+1})$。若任务规范将超长明确判为失败终止，则应据此设终止及失败奖励，不能在两种语义之间混用。

连续无终止段上的展开是 $\widehat{A}_t = \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}$，最后一步残差必须包含在内。常见文本任务取 $\gamma = 1$；若改用 $\gamma < 1$，长程奖励本身被折扣，已改变任务偏好。λ 在偏差与方差间折中，近似 critic、截断和 $\lambda < 1$ 可能带来估计偏差，GAE 并非自动无偏。

终止掩码的两步算例

取两步轨迹，$\gamma = \lambda = 1$、$\tilde{r} = (0, 1)$、$V_{\text{old}}(s_1) = 0.4$、$V_{\text{old}}(s_2) = 0.6$，第二步真实终止。则 $\delta_2 = 1 - 0.6 = 0.4$，$\delta_1 = 0.6 - 0.4 = 0.2$，故 $\widehat{A}_2 = 0.4$、$\widehat{A}_1 = 0.6$。两处 value target $\widehat{A}_t + V_{\text{old}}(s_t)$ 均为 1。若第二步实际是截断，后续 value 为 0.3，则两处优势与 target 都要再加 0.3。漏掉最后残差或误设终止，会直接改变第一步更新。

5.4 PPO：裁剪的是代理目标

PPO 使用采样时缓存的旧策略概率，构造 token 比率与裁剪代理 [12]：

$$\rho_t(\theta) = \frac{\pi_\theta(y_t \mid s_t)}{\pi_k(y_t \mid s_t)}, \tag{5.8}$$

$$L_{\text{clip}}(\theta) = \mathbb{E}_{(s_t, y_t)\sim\pi_k}\left[\min\left(\rho_t\,\text{sg}(\widehat{A}_t), \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\,\text{sg}(\widehat{A}_t)\right)\right], \tag{5.9}$$

其中 $\epsilon > 0$ 控制裁剪区间，sg 表示停止梯度。给定状态的动作比率并未校正整条状态访问分布；PPO 依赖局部更新与受控数据重用，不能据此声称已得到任意旧轨迹上的精确 on-policy 梯度。

正优势且 $\rho_t > 1 + \epsilon$ 时，这一 token 的正向激励被截平；负优势且 $\rho_t < 1 - \epsilon$ 时，继续降低其概率的激励被截平。但共享参数仍可改变这些概率，裁剪不是把所有比率强制锁在区间内，更不是全局 KL 上界。应监控实际比率、clip fraction、熵及新旧策略 KL，并设置过大偏移时提前结束 inner epochs 的规则。

critic 回归目标可取 $\widehat{V}_t = \text{sg}(\widehat{A}_t + V_{\text{old}}(s_t))$，总损失的一种约定为

$$\mathcal{L} = -L_{\text{clip}} + c_V \mathbb{E}_t[(V_\phi(s_t) - \widehat{V}_t)^2] - c_H \mathbb{E}_t[\mathcal{H}(\pi_\theta(\cdot \mid s_t))]. \tag{5.10}$$

本节已在回报中放入参考 KL；若再加一个 KL 损失，应说明它是额外正则而非无意重复。也可选择仅在损失中加 KL，但必须给出采样分布、估计器及梯度路径。熵正则、value clipping、advantage whitening、采样权重同样属于需要记录的实现选择。

5.5 RLOO：同一提示的留一 baseline

REINFORCE leave-one-out（RLOO）对同一提示独立采样 $G > 1$ 条序列，以其他序列的奖励均值作为 baseline[13]：

$$b_i = \frac{1}{G-1}\sum_{j\neq i} R_j, \qquad \widehat{A}_i = R_i - b_i = \frac{G}{G-1}(R_i - \overline{R}). \tag{5.11}$$

给定提示及采样策略，若其他候选与第 i 条的动作独立，b_i 不依赖该条动作，故满足 baseline 条件。若裁判对整组联合打分，使 R_j 本身依赖 y_i，或者组内通过共享搜索产生强耦合，不能直接沿用此无偏性论证。奖励和 baseline 在策略损失中均 detach。

若直接使用包括自身的组均值，则在独立同分布、固定奖励及无裁剪的简化情形，其梯度期望是原梯度的 $(G-1)/G$。留一均值修正这个自包含缩放；再除以依赖整组的随机标准差，会引入另一种估计器，不能只用“baseline 不影响期望”解释。

例如 $G = 4$、奖励为 $(1, 0, 0, 0)$，RLOO 优势为 $(1, -1/3, -1/3, -1/3)$。若四个奖励均为 1 或均为 0，RLOO 优势仍全为 0。增大组数可能增加发现异质结果的机会，但在一个已经全同的组上换 baseline，不能凭空恢复任务梯度。

5.6 可审计的训练循环与诊断

Listing 5.1 PPO 的局部更新循环；此处行为策略等于采样快照

```
for update_id in range(num_updates):
    snapshot = freeze(policy)
    seq, old_logp, end_flags = sample(snapshot, prompts)
    completion_mask = action_mask(seq, include_eos=True)
    old_values, bootstrap = frozen_value_predictions(seq, end_flags)
    raw_reward = reward_model_or_verifier(seq)
    ref_logp = reference.logprob(seq)
    shaped = put_terminal_reward(raw_reward, seq)
    shaped -= beta * (old_logp - ref_logp) * completion_mask
    adv, target = gae(shaped, old_values, end_flags,
                      bootstrap=bootstrap, mask=completion_mask)
    adv, target = stop_gradient(adv), stop_gradient(target)
    for inner in range(inner_epochs):
        new_logp, value = learner(seq)
        ratio = exp(new_logp - old_logp)
        policy_loss = -masked_mean(minimum(
            ratio * adv, clip(ratio, 1-eps, 1+eps) * adv),
            completion_mask)
        value_loss = masked_mean((value - target)**2, completion_mask)
        loss = policy_loss + value_coef * value_loss
        update(loss)
        if measured_old_new_kl() > early_stop_limit:
            break
    log(raw_reward, independent_success, reference_kl,
        entropy, clip_fraction, length, policy_lag, cost)
```

伪代码的 mask 仅覆盖有效 completion 动作，包含约定的 EOS，排除 prompt、padding 和工具返回文本；分布式归一化分母按全局有效 token 数计算。真实训练还需保存采样配置、旧概率、终止原因和 value 快照版本。若使用温度、top-p 等改变采样分布，行为策略不再是此处假定的 π_k，处理方法见第 7 章。

现象	先收集的证据	针对性处理
RM 奖励升、任务成功率降	独立验证、长度与格式切片、KL 与熵	修复反馈，限制漂移；不可只提高奖励权重。
策略更新几乎为零	全同奖励组比例、有效 mask、候选多样性	修复 mask 或探索/反馈；RLOO 不解决全同组。
value loss 或梯度爆炸	终止掩码、reward 尺度、ratio 与 dtype	先纠正 target 及数值问题，再调学习率。
离线正常、在线不稳	版本差、旧概率、采样分布、错误轨迹	限制陈旧度和数据重用，重收集有问题的轨迹。

练习与提示

1. 推导式 (5.3)，再构造一个依赖动作的 baseline 使结论失效。提示：二动作 bandit 中令 baseline 等于该动作奖励。
2. 将两步 GAE 算例改成 $\lambda = 0$。提示：优势只剩 TD 残差，value target 不再都等于完整蒙特卡洛回报。
3. 证明 RLOO 与去均值优势相差 $G/(G-1)$。提示：先把其他 $G-1$ 条的奖励和写成 $G\overline{R} - R_i$。

离线偏好优化可以省去这套在线采样与 critic 训练流程，但其目标依赖另一组明确假设。下一章逐步说明这些假设如何把奖励模型换成策略的 log-ratio。

第 6 章

DPO 与离线偏好目标

离线偏好优化从已经标注的候选中学习，更新内循环不必生成新回答。它的计算便利不意味着没有分布假设：候选来自哪种行为、哪些回答根本未被比较、参考模型如何约束相对概率，都影响结果。本章完整推导直接偏好优化（direct preference optimization，DPO），再说明相邻目标究竟改变了什么。

6.1 第一步：解固定奖励下的 KL 正则问题

对固定提示 x，考虑式 (1.4)。假设 $\beta > 0$、参考策略在相关回答上为正，且 $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) e^{r(x,y)/\beta} < \infty$。在完整条件序列分布上，最优解是

$$\pi^*(y \mid x) = \frac{\pi_{\text{ref}}(y \mid x) e^{r(x,y)/\beta}}{Z(x)}. \tag{6.1}$$

有限最大长度、含 EOS 的有限词表空间使求和容易定义；无界长度下必须额外保证归一化存在。这里先对所有允许分布求最优，并未保证某个有限参数神经模型一定能表示它。

取对数并移项，得到奖励的重参数化

$$r(x, y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x). \tag{6.2}$$

$\log Z(x)$ 只依赖提示。在同一提示的一对回答之间求差时，这个未知量恰好抵消。若把不同提示的回答强行组成一个 pair，这一步就不成立。

6.2 第二步：将奖励差代入偏好似然

采用第 4 章的 BT 模型，令

$$\Delta_\theta(x, y_a, y_b) = \log \frac{\pi_\theta(y_a \mid x)}{\pi_{\text{ref}}(y_a \mid x)} - \log \frac{\pi_\theta(y_b \mid x)}{\pi_{\text{ref}}(y_b \mid x)}, \qquad m_\theta = \frac{\beta}{\tau_{\text{pref}}} \Delta_\theta. \tag{6.3}$$

用参数化策略替换理想的 π^*，BT 预测概率变为 $\sigma(m_\theta)$。在 $\tau_{\text{pref}} = 1$ 的单位约定下，对固定偏好数据集 $\mathcal{D}_{\text{pref}}$ 最小化

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\text{pref}}} \log \sigma(\beta \Delta_\theta(x, y^+, y^-)). \tag{6.4}$$

这是 DPO 的 logistic 分类目标 [7]。推导消除了显式奖励网络，不需要对每个更新 step 计算 $Z(x)$，也不需要在线 rollout。但它没有消除 BT 模型假设、冻结 reference、偏好覆盖与函数类限制。

理论重参数化的证据范围

固定奖励的 KL 最优化有闭式解；把它代入 BT 似然，得到策略参数上的偏好最大似然目标。这两件事并不自动证明：有限、有噪声、分布偏移的数据上训练一个受限神经模型，就等价于已知真实奖励的全局 RL 最优解。DPO 也不能从没有出现、没有反馈的罕见错误中凭空获得监督。

6.3 第三步：从 token 得到完整序列概率

在本书约定下，$y = (y_1, \dots, y_{T_y})$ 包含真正终止的 EOS 或约定的 end-of-turn 动作。DPO 需要

$$\log \pi_\theta(y \mid x) = \sum_{t=1}^{T_y} \log \pi_\theta(y_t \mid x, y_{<t}). \tag{6.5}$$

实现中使用 completion mask 排除 prompt 和 padding，但包含答案的全部有效动作。若有多轮对话，应先声明优化的是哪个条件回答或哪些助手动作，工具观察只作为条件，不当作策略生成动作求概率。

把式 (6.5) 除以 T_y 得到平均 token log-prob，它通常不再是一个归一化序列分布的 log 概率。直接替换会改变 DPO 的隐式奖励与样本权重，应作为另一目标报告。长度归一化可能缓解某些长度捷径，也可能引入新的偏好；不能在理论写序列概率、代码写 token mean 后声称完全相同。

长度差异为何需要单独检查

设一对回答的每个有效 token 相对于 reference 的 log-ratio 都为 0.1，长度分别为 10 和 2。序列 sum 给出的相对差为 $1.0 - 0.2 = 0.8$；token mean 的差则为 0。这个合成算例并未说明长回答更正确，只说明两个目标确实不同。实际应报告 chosen/rejected 长度差、按长度匹配的 pair 结果，以及固定内容的冗长化扰动。

完整序列概率会随长度变化，但 DPO 比较的是两个 reference-relative log-ratio，而非只比较原始序列 log 概率。因此“sum 天然让 DPO 偏爱短答案”也过于简单：数据、reference、终止概率与标签中的长度相关性共同决定偏置。

6.4 梯度、温度与标签噪声

对一个硬偏好 pair，记 $m = \beta\Delta_\theta$，则

$$\nabla_\theta \ell = -\beta\sigma(-m)\left[\nabla_\theta \log \pi_\theta(y^+ \mid x) - \nabla_\theta \log \pi_\theta(y^- \mid x)\right]. \tag{6.6}$$

模型尚未满足的比较具有较大权重；正确分离后权重减小。这里没有要求每次更新后 chosen 的绝对概率都上升：共享参数和归一化可能让两个概率同时下降，只是 rejected 下降得更多。故除偏好 margin 外，还应监控 chosen/rejected 的 log-prob、一般能力与独立生成质量。

在“奖励 r 已固定”的理想 KL 问题中，较大 β 意味着更强约束。在 DPO 的数据拟合中，β 同时缩放分类 logit 与梯度，reference、优化步数和可表示函数类也在起作用，不能直接把这个理想结论当作训练曲线的单调定律。若软偏好为 $p \in (0, 1)$，单对自由拟合的最优条件为

$$\Delta_\theta^* = \frac{\tau_{\text{pref}}}{\beta} \log \frac{p}{1-p}. \tag{6.7}$$

这清楚地显示奖励单位、偏好噪声尺度和目标 log-ratio 之间的关系。硬标签可分离时，logistic 损失可能继续追求越来越大 margin；有限数据中的噪声和过拟合不能靠“没有 RL”自动消除。

软标签版本使用 $\ell(m; p) = -p \log \sigma(m) - (1-p) \log \sigma(-m)$。对称翻转模型下可按式 (4.4) 描述观测概率，但系统性裁判偏差需要反事实审计，不能仅通过一个噪声率修正。所谓保守 DPO 或 label smoothing 在不同实现中含义不一，报告时应直接写出这一损失及其 p 的来源。

标签翻转的独立参考值

取 $\beta = 0.5$、reference-relative log-ratio 为 0.4 和 -0.2，则 $m = 0.3$。正向损失 $\log(1 + e^{-0.3}) \approx 0.554355$，翻标后为 0.854355，相差 0.3；策略等于 reference 时均为 $\log 2$。

6.5 从目标差异理解相邻方法

下面只比较有明确原始定义的代表方法。方法名不能替代对应代码版本中的公式和归一化设置。

方法	目标或数据的关键变化	与 DPO 不能混同的假设
IPO	理论序列形式对 Δ_θ 做有限目标的平方拟合	不同偏好建模与正则推导；典型目标为 $(\Delta_\theta - 1/(2\beta_{\text{IPO}}))^2$，其系数约定须单列 [14]。
KTO	使用单条 desirable/undesirable 标签及参照点效用	数据不必成对；正负比例、效用权重及 KL 参照点估计影响目标 [15]。
ORPO	chosen 的 SFT 项与相对 odds 惩罚联合优化	没有独立冻结 reference，不等于把 DPO 公式中的 reference 删掉 [16]。
SimPO	用长度平均 log-prob 构造无 reference 的 margin 并加入目标间隔	明确改变序列 DPO 目标；目标 margin、长度与 SFT 初始化需要分别控制 [17]。

离线偏好学习适合已有可靠偏好、在线反馈昂贵且候选覆盖尚可的第一轮实验。若错误只在当前策略生成的罕见前缀出现，需刷新候选、补充数据或转入可验证反馈训练；这仍不保证某一方法普遍胜出。在线刷新 pair 后继续做 DPO，与用标量奖励做策略梯度是两个闭环，第 8 章再比较它们与蒸馏的关系。

6.6 实现与实验应保存什么

Listing 6.1 采用完整 completion 序列 sum 的 DPO

```
for batch in preference_loader:
    chosen_lp = sequence_logprob(policy, batch.chosen, sum_tokens=True)
    rejected_lp = sequence_logprob(policy, batch.rejected, sum_tokens=True)
    # Cache is valid only for this exact reference, tokenizer and template.
    chosen_ref, rejected_ref = frozen_reference_logprob(batch)
    margin = beta * ((chosen_lp - chosen_ref) -
                     (rejected_lp - rejected_ref))
    loss = softplus(-margin).mean()  # hard chosen labels
    # If adding SFT, state its token/sample normalization and coefficient.
    update(loss)
    log(margin, chosen_lp, rejected_lp, batch.length_gap)
```

缓存 reference 概率必须绑定 checkpoint、tokenizer、chat template、EOS 和截断规则的 hash；改变任意一项都要重算。chosen 与 rejected 使用同一提示条件与一致模板。禁止只对 chosen 包含 EOS 或用不同最大长度截断而不记录。训练指标之外，固定生成预算评估独立保留集的成功率、偏好、安全回归及长度；用配对提示比较，防止标签和训练 loss 彼此自证。

练习与提示

1. 推导式 (6.3)，解释 $Z(x)$ 何时抵消。提示：同一提示、共同 reference、同一奖励单位缺一不可。
2. $p = 0.75$、$\beta = 0.5$、$\tau_{\text{pref}} = 1$ 时求最优 Δ_θ。提示：答案为 $2\log 3$，不要把 BT 奖励差与策略 log-ratio 差混为一谈。
3. 构造一个“更长”的标签捷径，再设计一个内容固定的反事实审计。提示：长度匹配数据和改变损失归一化是两种不同干预，应分开消融。

第 7 章

可验证奖励与组相对优化

可验证奖励强化学习（reinforcement learning with verifiable rewards，RLVR）用程序、规则或环境检查提供反馈。它描述反馈来源，并不指定优化器：同一验证器可配合 REINFORCE、PPO、RLOO 或 GRPO。反过来，GRPO 也可以使用神经奖励模型。把“反馈接口”和“梯度估计器”分开，才能讨论一种改进究竟解决了什么。

7.1 验证任务必须先给出契约

数学最终答案、代码测试、形式化证明、结构化输出与工具调用都可构成验证任务，但其可信范围不同。对每个任务先声明输入输出、解析规则、允许资源、终止与超时、测试可见性和失败分类。

接口	可重复检查的性质	必须独立审计的缺口
数值答案解析	输出符合容差或等价表达式	解析歧义、单位、多个答案、碰巧猜对与错误过程。
代码执行	在指定沙箱内通过测试	覆盖不足、测试泄漏、超时与访问外部资源。
证明检查器	证明项满足形式系统	形式化命题是否符合自然语言任务、是否允许不当公理。
JSON 或工具 schema	格式与参数类型合法	参数语义、权限、工具副作用和真实任务完成。

可重复执行不代表不能被投机。若奖励只看答案字符串，模型可学会重复候选数字；若只看 schema，可输出合法但无用的调用。训练验证器与保留集检查应尽量解耦：保留隐藏测试、反事实任务、人工抽样及失败 trace，区分解析失败、真实错误和基础设施错误。不能把网络超时悄悄记成模型推理错误，也不能无记录地丢弃失败样本。

7.2 GRPO 的组内相对优势

组相对策略优化（Group Relative Policy Optimization，GRPO）由 DeepSeekMath 提出，以同一提示的多条回答估计相对回报，通常省去独立 critic[18]。先固定采样快照 π_k，暂设 $y_i \overset{\text{iid}}{\sim} \pi_k(\cdot \mid x)$，每组大小 $G > 1$。对终局奖励 R_i，定义

$$\overline{R}_x = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad \sigma_x = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(R_i - \overline{R}_x)^2}, \tag{7.1}$$

$$\widehat{A}_i = \frac{R_i - \overline{R}_x}{\sigma_x + \varepsilon_{\text{norm}}}. \tag{7.2}$$

$\varepsilon_{\text{norm}} > 0$ 是标准化的数值稳定项，区别于 ratio 裁剪阈值 ϵ。本章使用总体方差分母 G；若实现用 $G-1$ 或把稳定项加在平方根内，必须相应记录。结果监督下，序列内所有有效动作共享 $\widehat{A}_i$；过程监督需要另行定义逐步 return，不能把两者公式混用。

减去组均值去掉提示特有的奖励平移，除标准差大致消去正比例尺度；有限稳定项使尺度不变性只近似成立。与此同时，每组按自己的随机标准差缩放，改变了不同提示和不同难度的相对权重。

组均值包括自身，标准差也依赖自身，故这不是“只减一个动作独立 baseline”的无偏性结论。

对于 $(1,0,0,0)$，忽略稳定项时 $\overline{R}=1/4$、$\sigma=\sqrt{3}/4$，标准化优势为 $(\sqrt{3},-1/\sqrt{3},-1/\sqrt{3},-1/\sqrt{3})$。同一组的 RLOO 优势则为 $(1,-1/3,-1/3,-1/3)$。两者有相同符号，但缩放机制不同；在不同组间，比例也不同。

7.3 写出一个与实现一致的代理目标

记第 i 条回答长度 T_i 包含 EOS，token 比率为

$$\rho_{it}(\theta)=\frac{\pi_\theta(y_{it}\mid s_{it})}{\pi_k(y_{it}\mid s_{it})}. \tag{7.3}$$

沿用 DeepSeekMath 结果监督的每序列平均结构，可写局部代理

$$L_{\mathrm{GRPO}}(\theta)=\mathbb{E}_{x\sim\mathcal{D},\,y_{1:G}\sim\pi_k^{\otimes G}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\Big\{\min\big(\rho_{it}\,\mathrm{sg}(\widehat{A}_i),\,\mathrm{clip}(\rho_{it},1-\epsilon,1+\epsilon)\,\mathrm{sg}(\widehat{A}_i)\big) - \beta\,\hat{k}_{it}(\theta)\Big\}\Bigg]. \tag{7.4}$$

本章将 reference 在整个对照阶段固定；DeepSeekMath 的迭代算法还讨论了外层刷新 reference 及继续训练奖励模型。刷新规则是额外算法选择，应记录版本，不能把它与每个 batch 刷新的采样快照混淆。组内奖励和优势缓存后停止梯度。原方法以采样 token 上的非负 KL 表达式构造惩罚；下节说明其数值与梯度含义。式 (7.4) 是旧轨迹上的裁剪代理，不能直接等同于第 1 章当前策略的完整序列 KL 目标。

长度归一化有真实后果。对每条序列先除 T_i，每条回答总权重接近相等，长回答中单个 token 权重较小。全 batch token 平均则为

$$\frac{\sum_i\sum_{t=1}^{T_i}\ell_{it}}{\sum_i T_i},\qquad \text{不同于}\qquad \frac{1}{G}\sum_i\frac{\sum_t\ell_{it}}{T_i}. \tag{7.5}$$

例如两条序列长度为 1 和 3，每 token 代理值分别为 $+1$ 与 -1，每序列平均得到 0，全 token 平均得到 -0.5。这只是合成算例，却足以证明两种代码并非同一个损失。固定最大长度分母、每提示平均、每序列平均和全 token 平均，都应在对照中明确区分。DAPO 专门讨论了动态采样与 token 级归一化等工程选择 [19]；不能只换算法名而省去这些差异。

7.4 全同奖励组与长度偏置

若同组奖励全相同，式 (7.2) 的分子全零，因此任务优势全零；添加稳定项避免 NaN，并不会创造学习方向。RLOO 对该组也全零。若仍有 KL、熵或 SFT replay 项，参数可以更新，但不能把这种更新算作本组任务回报提供的信号。

对独立的二值结果、单次成功概率 p，组内全同的概率为

$$P_{\text{same}}=p^G+(1-p)^G. \tag{7.6}$$

因此非常容易或非常困难的提示都可能没有组内方向。增大 G 可增加得到混合结果的机会，但固定总 rollout 预算时会减少覆盖的提示数。过滤全同组并重采样会改变有效提示分布，且额外采样有成本；应记录过滤前后难度、重采样次数和总生成 token，不只记录进入 learner 的 token。

可以选择保留零优势组并如实报告，或在预先规定的预算内改探索、改课程、增加有根据的过程信号或 SFT 示范。若验证器错误、任务完全不可达或所有采样都因模板失败，单纯调 baseline 与组大小无济于事。

长度偏置还来自奖励定义。逐步分数求和可能奖励冗长，平均可能偏爱少做步骤，超长一律记零可能误伤接近正确的长轨迹。报告长度分桶成功率、超长率、各长度段的优势及 EOS 概率；用等内容的长短变体检查反馈，再讨论损失分母，避免把不同机制统称“长度问题”。

7.5 KL 值的估计与 KL 梯度是两件事

对固定状态 s，令 $P = \pi_\theta(\cdot \mid s)$、$Q = \pi_{\text{ref}}(\cdot \mid s)$。若两者在有限词表上严格为正且 $a \sim P$，两个采样量为

$$k_1(a) = \log P(a) - \log Q(a), \tag{7.7}$$

$$k_3(a) = e^{d(a)} - 1 - d(a), \qquad d(a) = \log Q(a) - \log P(a). \tag{7.8}$$

k_1 单点可负，k_3 逐点非负；二者在此采样条件下的期望都等于 $D_{\text{KL}}(P\|Q)$。k_3 的恒等式来自 $\mathbb{E}_P[e^d] = \sum_a Q(a) = 1$。这不表示所有状态、所有旧轨迹、所有反传方式都等价。

1. 若 $a \sim \pi_k$ 或其他行为策略，而 P 已改变，未重加权的 k_1, k_3 通常不再无偏估计当前 $D_{\text{KL}}(P\|Q)$。
2. 在固定采样 token 上直接对 k_3 反传，只保留被采样表达式的显式导数，漏掉采样分布随 P 变化的 score 项。因此即使值的期望相等，其梯度一般不等于完整词表 KL 的梯度。
3. 对同一旧前缀计算全词表 KL，可以核对局部数值并得到该固定状态的精确梯度；它仍未自动校正当前策略访问状态的分布。

例如 $P = (0.8, 0.2)$、$Q = (0.5, 0.5)$，$D_{\text{KL}}(P\|Q) \approx 0.192745$；以第一动作 logit 为参数，完整 KL 梯度为 $0.16 \log 4 \approx 0.221807$，固定采样权重下直接求 k_3 导数的期望却为 0.3。第 13 章保存了独立数值检查，附录 A 推导这一差异。实现若用 k_3 作为代理惩罚，应如实标记；改为全词表 KL 或加入完整 score 项，是另一种梯度设定，需要单独比较。

7.6 采样分布、旧策略与 on-policy 条件

若按 π_k 的原始 softmax、温度 1、不截断词表采样，可令行为策略 $\mu = \pi_k$。若用温度 τ_s、top-p 或 top-k，实际行为变为

$$\mu(a \mid s) \propto \exp(z_k(a \mid s)/\tau_s)\, \mathbb{I}\{a \in \mathcal{S}(s)\}. \tag{7.9}$$

因此“从当前模型生成”还不足以证明样本来自当前未变换策略的目标分布。必须保存温度、过滤集合、归一化后的真实行为 log-prob，以及用于损失的未变换 log-prob。

若目标仍是未变换的 π_θ，条件动作校正需考虑 π_θ/μ，而非忽略温度的 π_θ/π_k。仅校正当前动作还不能校正旧前缀分布；完整序列重要性比率是 $\prod_t \pi_\theta(y_t \mid s_t)/\mu(y_t \mid s_t)$，长序列下方差可能极大。top-$p$ 把目标仍可能生成的动作概率置零，破坏支持覆盖，不能靠有限重要性权重恢复缺失部分。

另一选择是把变换后的分布明确作为待优化策略，但此时模型概率、比率、参考与目标也要一致定义；随参数变化的截断集合还会带来额外问题。教学基准先使用温度 1、全支持采样，便于核对；使用启发式近似的生产实现应直接说明近似及其诊断。在线收集、近似 on-policy 和严格符合目标采样分布，是不同层面的描述。

7.7 一致的训练伪代码与系统报告

Listing 7.1 每序列平均的 GRPO 局部代理；全支持、温度 1 采样

```
for prompts in prompt_loader:
    snapshot = freeze(policy)
    groups, old_logp = sample(snapshot, prompts, G,
                              temperature=1, top_p=1)
    rewards = verifier(groups)
    adv = stop_gradient(group_normalize(rewards))
    ref_logp = reference.logprob(groups)  # frozen
    for inner in range(inner_epochs):
        new_logp = policy.logprob(groups)
        ratio = exp(new_logp - old_logp)
        gain = minimum(ratio * adv,
                       clip(ratio, 1-eps, 1+eps) * adv)
        d = ref_logp - new_logp
        k3 = expm1(d) - d  # proxy gradient, see text
        token_objective = (gain - beta * k3) * completion_mask
        per_sequence = token_objective.sum(-1) / valid_lengths
        loss = -per_sequence.mean()  # equal G per prompt
        update(loss)
    log(all_equal_group_rate, resample_cost, raw_reward,
        independent_success, reference_kl, entropy, length)
```

稳定实现还需对 long sequence 的 log-prob 使用 float32 累加，并对 e^d 的非有限值报警；简单把它裁小会改变目标，必须记录。若组大小不同或进行了动态过滤，最后的 mean 应重新核对是否仍给每提示、每组或每序列所声称的权重。数据版本、旧策略版本、参考版本和验证器版本共同决定一批轨迹的含义。

DeepSeekMath 展示了数学数据、SFT 与 GRPO 的组合；DeepSeek-R1 报告了规则反馈探索，以及后续拒绝采样、SFT 和蒸馏阶段 [18, 20]。这些系统报告支持研究多阶段闭环，不足以推出“任意基座只要套 GRPO 就会获得可靠推理”，也不单独证明一条可读推理链的每一步都正确。比较时同时固定提示、生成预算、更新预算与独立验证口径。

练习与提示

1. 对 $G = 8$、$p = 0.01$ 估算全同组概率。提示：$(0.99)^8$ 约为 0.923，不能把困难提示上的无方向误判为程序一定有 bug。
2. 证明 $\mathbb{E}_P[k_3] = D_{\mathrm{KL}}(P\|Q)$，再说明支持或采样分布改变时哪一步失效。提示：先化简 $\mathbb{E}_P[e^d]$，检查剩余求和是否等于 1。
3. 在固定总生成 token 下设计 $G = 4, 8, 16$ 的比较。提示：同时报告不同提示数、接受组数和重采样成本，不能只比较每 step 奖励。

第 8 章

蒸馏：在谁的前缀上学习什么

强化学习用回报告诉学生哪些行为值得增加概率；蒸馏则把教师的输出作为监督。二者都能使用学生生成的数据，但反馈和目标不同。本章先固定状态分布与局部损失，再讨论如何刷新数据。这样才能分清教师文本、教师概率、偏好标签和任务奖励分别传递了什么信息。

8.1 全称、接口与分布

本书把 *on-policy distillation* 译为同策略蒸馏，缩写 OPD；这里的 policy 是生成前缀的学生策略。它通常先让学生生成，再让冻结教师在这些前缀上提供监督。*Online preference distillation* 则译为在线偏好蒸馏，本章用这一全称描述“生成候选、教师比较、偏好更新”的闭环，不把它视为一个统一、标准化的 OPD 算法名称。若论文使用其他展开，阅读时应保留其原全称和目标。

路线	前缀与反馈	训练对象及边界
离线软标签蒸馏	固定数据前缀、教师词表概率	逐 token 散度；需要可对齐的词表与概率接口。
离线序列蒸馏	固定教师回答、硬标签	对教师文本做 SFT；不保留完整的教师不确定性。
同策略 token 蒸馏	学生生成前缀、教师概率	在学生访问的状态上匹配局部分布；刷新与散度方向需另外指定。
在线偏好蒸馏	学生候选、教师或裁判比较	对新收集的偏好对做 DPO-like 更新；不是 token KL。
自训练与筛选	自身候选、规则或模型筛选	对接受样本重新拟合；没有新教师知识也可能改变数据分布。
在线 RL	策略轨迹、标量或过程回报	策略梯度；必须另外处理采样策略、优势和重要性比率。

蒸馏可以迁移到较小模型，也可以在同规模模型之间进行。[21] 给出经典软标签蒸馏视角；GKD（Generalized Knowledge Distillation）把学生生成的前缀和多种散度用于自回归模型 [22]。这些路线不要求把教师当作真值：教师会在陌生或已经不可恢复的前缀上犯错。

8.2 先冻结前缀，再定义 token 目标

第 k 轮从学生快照 π_k 出发，用实际采样器 μ_k 收集轨迹，得到固定的有效前缀集合 $\mathcal{S}_k$。当生成温度为 1、无 top-p/top-k 截断且停止规则匹配时，$\mu_k = \pi_k$；否则应保存真实采样配置，不能只保存原始模型名称。令 d_k 是在这批前缀上按有效 token 计数归一化的经验分布，冻结教师记为 q_{T}。一个明确的局部目标是

$$\begin{aligned}\mathcal{L}_{\mathrm{fKD}}^{(k)}(\theta) &= \mathbb{E}_{s\sim d_k}\left[-\sum_{v\in\mathcal{V}} q_{\mathrm{T}}(v \mid s)\log \pi_\theta(v \mid s)\right] \\ &= \mathbb{E}_{s\sim d_k}\left[D_{\mathrm{KL}}(q_{\mathrm{T}}(\cdot \mid s)\|\pi_\theta(\cdot \mid s)) + H(q_{\mathrm{T}}(\cdot \mid s))\right].\end{aligned} \tag{8.1}$$

教师熵与 θ 无关，因此最小化软交叉熵等价于最小化这里的前向 KL。前缀、有效 mask 和教师概率都在该轮固定；内循环只对学生在这些前缀上的 logits 求导。若先按回答取均值再按样本取均值，则

每个回答等权；它与本式按 token 加权的目标不同，需单独记录。

同一固定状态分布上也可以用反向 KL：

$$\mathcal{L}_{\mathrm{rKD}}^{(k)}(\theta) = \mathbb{E}_{s\sim d_k} \sum_{v\in\mathcal{V}} \pi_\theta(v \mid s) \log \frac{\pi_\theta(v \mid s)}{q_{\mathrm{T}}(v \mid s)}. \tag{8.2}$$

若完整词表概率可用，式 (8.2) 的局部词表求和可以直接微分，并不必然需要高方差采样。困难在于大词表成本、教师访问接口，以及状态分布是否也随学生改变。有限容量下，前向 KL 对遗漏教师高概率区域惩罚较强；反向 KL 对学生给教师低概率区域分配质量惩罚较强。这是优化倾向，不是任何数据和参数化下都成立的性能排序。

8.2.1 温度缩放究竟改了什么

若使用经典 softened KD，明确令

$$q_{\mathrm{T}}^{(\tau)} = \mathrm{softmax}(z_{\mathrm{T}}/\tau), \qquad p_\theta^{(\tau)} = \mathrm{softmax}(z_\theta/\tau), \qquad \ell_\tau = -\tau^2 \sum_v q_{\mathrm{T}}^{(\tau)}(v) \log p_\theta^{(\tau)}(v). \tag{8.3}$$

$\tau > 0$ 是蒸馏温度；τ^2 是常用的梯度尺度补偿，仍需与其他损失权重共同设定。只软化教师而保持学生温度为 1 也是合法目标，但它不是同一个公式。生成温度、蒸馏温度、偏好噪声温度和 KL 系数 β 分属不同接口，配置文件应分开命名。

8.2.2 一个两 token 算例

在同一前缀上，教师分布为 $q_{\mathrm{T}} = (0.9, 0.1)$，学生为 $p = (0.6, 0.4)$。自然对数下，

$$\begin{aligned} D_{\mathrm{KL}}(q_{\mathrm{T}} \| p) &= 0.9 \log(1.5) + 0.1 \log(0.25) \approx 0.2263, \\ D_{\mathrm{KL}}(p \| q_{\mathrm{T}}) &= 0.6 \log(2/3) + 0.4 \log 4 \approx 0.3112. \end{aligned} \tag{8.4}$$

两个数都非负，但大小和梯度不同。对前向 CE，学生 logits 的梯度是 $p - q_{\mathrm{T}} = (-0.3, 0.3)$。若把学生自己采到的 token 当作未筛选的硬标签，在采样策略与当前学生相同时，其期望 logits 梯度为零；仅“让学生生成再拟合自己”不能自动获得教师纠错信息。

8.3 局部梯度与完整轨迹目标不能混同

式 (8.1) 在一个刷新周期内固定 d_k。若改写成

$$\mathcal{L}(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[F_\theta(y)], \tag{8.5}$$

并声称求这个完整目标的梯度，则在可交换求和与微分等通常条件下，

$$\nabla_\theta \mathcal{L} = \mathbb{E}_{y\sim\pi_\theta}\left[\nabla_\theta F_\theta(y) + F_\theta(y) \nabla_\theta \log \pi_\theta(y \mid x)\right]. \tag{8.6}$$

第二项来自轨迹分布随参数变化。离散采样通常不能直接反向传播，但这一 score-function 项有明确定义；忽略它得到的是固定样本上的局部更新，不能称为完整梯度。局部更新配合不断刷新的学生前缀，是一种需要独立评测的训练过程。

局部梯度甚至可以与完整梯度异号

考虑两个二元 token。第一步学生生成 1 的概率为 $p = \sigma(\theta)$，教师为 0.5；第二步学生生成 1 的概率在第一步为 0、1 时分别为 0.2、0.8，教师对应为 0.8、0.6，第二步概率本身不依赖 θ。取两步 CE 之和，$\theta = 0.4$ 时，固定前缀的局部梯度为 0.09869，状态分布项为 −0.13323，完整梯度为 −0.03454。随书标准库脚本 `evidence/sources/check_distillation.py` 用精确枚举与中

心差分核对此算例。这是合成教学计算，不是大模型训练结果；它说明必须先说清楚优化的是哪一个目标。

对相同词表、相同终止约定且满足支持条件的教师和学生，完整序列反向 KL 还有链式分解：

$$D_{\mathrm{KL}}(\pi_\theta(y \mid x) \| q_{\mathrm{T}}(y \mid x)) = \mathbb{E}_{y \sim \pi_\theta} \sum_{t=1}^{T_y} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t) \| q_{\mathrm{T}}(\cdot \mid s_t)) . \tag{8.7}$$

这里是每条序列的 token 求和，再对序列求期望，包含 EOS。它不等于可变长度下任意的 token 平均。MiniLLM 从完整序列反向 KL 出发，构造同策略优化与方差控制 [23]；不能仅把式 (8.2) 换个名称就宣称复现了它的全部算法。

前缀来自学生，不等于每个更新都严格同策略

快照 π_k 的旧前缀在更新多轮后不再来自当前 π_θ。局部监督蒸馏仍可以使用它们，但若要解释成当前策略上的完整目标，就需要重新分析分布差异。应记录刷新频率、内循环步数、采样配置和策略版本，比较旧前缀与新前缀上的 KL、错误率及长度。

8.4 词表、尾部概率与教师成本

逐 token KL 要求两边在同一动作空间上定义概率。教师 token 的索引 100 不一定对应学生索引 100，甚至一个教师 token 可能对应多个学生 token。词表不兼容时，应使用有明确概率语义的映射，或退回教师文本的序列蒸馏；不能直接按索引相减。黑盒教师若只返回文本，也不能被描述成提供了完整 logits。

若接口只返回教师 top-k token，直接在这 k 项重新归一化会丢弃尾部质量，训练的是截断后的教师分布。若能得到真实的 top-k 概率及学生在同一集合的概率，可另设一个尾部桶：

$$q_\perp = 1 - \sum_{v \in K} q_{\mathrm{T}}(v), \qquad p_\perp = 1 - \sum_{v \in K} \pi_\theta(v). \tag{8.8}$$

在 $K \cup \{\perp\}$ 上计算的是合并类别后的 KL，它是完整 KL 的下界，仍不恢复尾部内部的差异。保存 top-k mass，并在一小批可访问完整分布的样本上比较误差，比把缺失概率当零可靠。数值计算应使用稳定的 log-softmax；若教师真正给学生可采动作零概率，反向 KL 可能无穷，平滑会改变目标，必须显式记录。

教师可在一条完整学生轨迹上做一次因果前向，取得每个位置对应的条件分布；它看到的仍是学生前缀。成本因此同时报告教师处理 token 数、上下文长度、词表输出量、调用数、缓存命中率和时间。只比调用数会掩盖批量大小及长上下文差异。

Listing 8.1 固定快照前缀上的 token 蒸馏：目标与分布分开保存

```
for refresh in range(num_refreshes):
    snapshot = freeze(student)
    traj = generate(snapshot, prompts, actual_sampling_config)
    prefixes, mask = make_student_prefixes(traj, include_eos=True)
    targets = teacher_logprobs(prefixes)  # frozen teacher
    weights = frozen_confidence_weights(targets, verifier(traj))
    for step in range(inner_steps):
        logp = student_logprobs(prefixes)
        per_token = forward_ce(targets, logp)
        loss = weighted_token_mean(per_token, mask, weights)
        update(student, loss)
    evaluate_new_prefixes(student, independent_holdout)
```

置信门控会改变训练分布：教师高熵可能意味着错误，也可能意味着多种合法答案。低熵错误又可能被过度保留。应按任务正确性、长度和难度检查被过滤样本，并把门控作为单独消融。教师冻结、一般能力 replay、限定内循环和定期刷新也都是可检验的设计选择，不能代替独立的任务反馈。

8.5 在线偏好蒸馏：本书采用的具体闭环

令 $x \sim \mathcal{D}$，从实际采样快照独立生成有序候选 $(y_1, y_2) \sim \mu_k^{\otimes 2}(\cdot \mid x)$。教师或裁判输出 $p_J \in [0,1]$，表示在固定 rubric 下 y_1 优于 y_2 的概率，以及质量权重 $w_J \geq 0$。定义

$$
\begin{aligned}
h_\theta &= \log \frac{\pi_\theta(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)} - \log \frac{\pi_\theta(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)}, \qquad z_\theta = \beta h_\theta, \\
\mathcal{L}^{(k)}_{\text{online-pref}} &= \mathbb{E}\left[w_J\{-p_J \log \sigma(z_\theta) - (1-p_J)\log \sigma(-z_\theta)\}\right].
\end{aligned}
\tag{8.9}
$$

本式把偏好温度吸收到奖励单位，按同一 prompt 的完整 completion 对数概率求和；它是本书明确定义的在线 DPO-like 实例，不代表所有同名方法。严格标签可令 $p_J = 1$ 或 0；平局可保留为软标签或按事先规则丢弃。若先把候选排序为 chosen/rejected，就要同步转换概率，不能让排序与标签方向冲突。

μ_k 决定候选来源，π_{ref} 是冻结的偏好锚点，两者作用不同。每轮更新参考模型会改变算法。这里在固定 pair 上做监督式偏好梯度，不自动需要 PPO 的 old-policy ratio；若改用标量回报和策略梯度，就是另一个在线偏好 RL 目标。偏好对数量增加也不保证额外信息：候选重复、裁判位置偏差及同一 rubric 的共适应都可能让闭环失效。

一轮在线偏好蒸馏的可复核记录

保存 prompt 与 split、两条原始候选、采样策略版本、生成配置、裁判版本与 rubric、两个位置顺序的判定、平局/解析失败、最终 p_J, w_J、冻结参考版本和更新步数。报告 pair 接受率、去重率、策略陈旧程度、judge token 成本，以及独立 verifier 或人工复核结果。

8.6 何时值得做同策略蒸馏

如果错误主要发生在学生自己生成的早期前缀之后，教师在这些状态上的纠错可能比继续增加教师成功轨迹更直接。但若教师在错误前缀上不可靠，或任务已经不可恢复，局部概率匹配可能强化错误。2026 年关于 OPD 的研究讨论教师与学生行为模式重叠、教师信息增量和配方条件 [24]；本书将这些视为该研究设置下的证据，不据此宣称任意模型对都受益。

可以提出一个更具体的假设：在固定学生更新 token、教师处理 token、提示集合和最大输出长度时，学生前缀蒸馏比固定教师前缀蒸馏降低长答案的早期错误率。除最终成功率外，还应报告按位置错误、教师在错误前缀上的正确率、KL、输出熵、长度和成本。若只看到更长回答带来更高 pass@N，尚不足以证明前缀纠错机制。

练习与提示

1. 用式 (8.6) 写出一个两步生成模型的完整梯度，并指出固定第一步样本时漏掉哪项。提示：分别对轨迹概率和局部损失求导。
2. 设教师 top-2 总质量只有 0.7，构造两个尾部分配不同但 top-2 一致的学生，说明重新归一化的 top-2 KL 无法区分什么。提示：保留尾部桶仍会丢掉尾部内部差异。
3. 在有序 pair 中交换 y_1, y_2，证明同时令 $p_J \mapsto 1 - p_J$ 不改变式 (8.9)。提示：z_θ 变号。
4. 设计一组教师与学生分词不兼容的字符串，说明为什么文本一致不能推出 token 分布可直接比较。提示：检查空格、汉字和多 token 数字。

第 9 章

合成数据、自训练与工具轨迹

上一章的教师可以输出完整分布，也可以只输出文本或比较标签。若只能获得候选和筛选信号，最直接的路线是把通过检查的行为变成新训练数据。本章关注这一闭环改变了什么分布，以及如何分开衡量模型本身、搜索过程和选择器的贡献。

9.1 筛选后再拟合：一个条件分布

设候选来自行为采样器 $\mu(y \mid x)$，接受概率为 $a(x,y) \in [0,1]$。在同一 prompt 下独立采样并按此规则接受，若

$$Z_a(x) = \sum_y \mu(y \mid x) a(x,y) > 0, \tag{9.1}$$

则一个被接受的候选服从

$$p_{\text{sel}}(y \mid x) = \frac{\mu(y \mid x) a(x,y)}{Z_a(x)}. \tag{9.2}$$

确定性验证器对应 $a(x,y) = \mathbf{1}\{R(x,y) \geq c\}$。在接受样本上做 SFT，常被称为筛选微调或拒绝采样微调（rejection fine-tuning，RFT）；本书每次使用 RFT 都指这一具体流程，避免与其他同缩写方法混同。它通过筛选改变经验分布，没有自动变成在线策略梯度。

式 (9.2) 只描述独立接受的理想情况。每题最多留一条、去重、跨题配额、难度分层和 best-of-N 都会进一步改变分布。难题接受率低时，如果直接合并所有通过样本，容易把训练数据改成“简单题较多”的集合。因此应同时记录候选侧与接受侧的题目数、每题候选数、通过率、长度和难度分布。

筛选过程的手算检查

一个 prompt 只有三种候选，$\mu = (0.6, 0.3, 0.1)$；验证器只接受后两种，则接受率为 0.4，接受分布为 $(0, 0.75, 0.25)$。若每天为每题固定采样 N 次，至少接受一条的概率为 $1 - 0.6^N$，但这不会使第一种错误候选变得正确。若验证器误接收它，拟合也会写入相应错误。

筛选只能放大已经采到且被接受的区域。基础策略给正确行为的概率极低时，有限候选可能一次也见不到；若确实为零，独立筛选不能创造该行为。这里说的“有效支持集扩展”应理解为让有限预算下罕见行为更常出现，不能替代概率支持的严格含义。

9.2 Best-of-N、pass@N 与实际选择成功率

令每个独立候选的真实成功概率为 p。在这一简化条件下，N 个候选中至少一个成功的概率为

$$P(\text{至少一次成功}) = 1 - (1-p)^N. \tag{9.3}$$

它描述候选覆盖，不描述选择器能否找到成功候选。实际 best-of-N 按打分器选出 $\arg\max_i \hat{R}(x, y_i)$；打分器可能把错误但更长或格式更漂亮的回答选到第一。若只有完美 oracle 才知道谁正确，不能把候选中的 pass@N 当作实际部署的选中成功率。

Self-consistency 对多个推理输出的最终答案投票，其作用取决于正确与错误答案的概率质量分布。同一提示下独立采样可以条件独立，但即使如此，一个常见错误答案拥有最大概率质量时，增

加候选仍可能使投票更稳定地选错。若候选来自共享搜索前缀或相互依赖的修订，独立采样假设还会失效；不能仅凭轨迹文字不同就认定独立。比较时应分别报告单次生成成功率、候选覆盖、实际选择成功率、无效回答和投票平局。

把搜索改善写回模型后仍需重新测 pass@1

一个系统可能依靠 32 次采样和昂贵教师筛选取得高分。将这些答案做 SFT 后，学生的单次质量是否提高，是另一个需要测量的问题。教师候选预算、训练 token 和部署候选预算都应列入成本表；不能把增加推理预算与提高基础策略混成同一结论。

若最终只需一个答案，可固定最大总生成 token 和选择器预算，比较“更多短候选”与“更少长候选”。若某方法使用执行器筛选而另一方法没有，应该把选择器差异作为处理条件，而不是只在脚注中写“采样参数略有不同”。

9.3 自训练、批评修订与闭环退化

自训练（self-training）用模型生成的结果作为后续监督。生成器、批评者、修订器和筛选器可以由同一模型的不同提示、不同快照或独立模型承担；这些名称不保证它们拥有独立信息。有效的新信号可能来自环境执行、外部材料、不同模型的互补知识，或对高质量区域的明确选择。

没有筛选或外部约束时，对自身分布重新做极大似然，其总体最优点仍是原分布；有限样本和优化误差却可能导致熵下降、错误放大或长尾丢失。因此，自训练有收益不能仅归因于“模型学会反思”。应检查哪种数据选择、验证器或搜索预算带来了变化。

Listing 9.1 生成、验证、分层选择与再训练的完整闭环

```
for round_id in range(num_rounds):
    snapshot = freeze(policy)
    candidates = generate(snapshot, train_prompts, budget, config)
    checks = verifier(candidates, version=pinned_verifier)
    selected, rejected = stratified_select(
        candidates, checks, quota_per_prompt=quota,
        length_and_difficulty_buckets=True, deduplicate=True)
    audit_selection(candidates, selected, rejected)
    train_data = merge(selected, fixed_general_replay)
    policy = sft_update(policy, train_data, fixed_update_budget)
    evaluate(policy, untouched_holdout, independent_checks)
    stop_if_quality_gate_fails()
```

将拒绝样本保存下来，有助于检查验证器误判和偏好对构造；它不意味着把错误回答当作正例训练。对 critique–revise 流程，分别评估“原回答正确且被改坏”“原回答错误且被修好”和“修改后仍错误”。只统计最终正确率会隐藏无效修订。对 self-play，应固定环境规则和评测对手，防止双方共同适应出只在内部有效的协议。

DeepSeek-R1 技术报告区分纯 RL 探索、多阶段训练、拒绝采样和向较小模型蒸馏 [20]；Tülu 3 给出数据、偏好与可验证反馈阶段的公开配方 [3]。本书借此说明阶段可以组合，不把某个系统的最终分数归因于其中一个算法名称，也不将其具体收益外推到未测量的模型和任务。

9.4 工具与 Agent：训练对象是动作和恢复

工具任务的轨迹交替包含模型动作与环境观测，可记为

$$\xi = (s_1, a_1, o_2, r_1, s_2, a_2, o_3, r_2, \ldots), \qquad s_{t+1} = f(s_t, a_t, o_{t+1}). \tag{9.4}$$

这里 a_t 可以是一段工具调用 token、一个环境动作或终止；o_{t+1} 是外生工具结果。使用 ξ 表示轨迹，避免与温度 τ 混淆。语言模型的策略概率只乘入它生成的动作 token，通常不把工具返回文本当作策略采样动作。

例如一段“assistant 调用计算器 → tool 返回结果 → assistant 回答”的串行化数据，可以训练调用参数和最终回答的 token；工具结果只作为条件输入。若 loss mask 把结果也设为 1，目标就变成部分模拟工具观测；在离线环境建模中这可以是有意设计，但不等于同一个 Agent 策略目标。RL 的 log-prob 与重要性比率也必须使用动作 mask。

事件	必须保存的证据	训练与评测含义
工具调用与返回	schema 版本、参数、结果摘要、时间、调用标识	分清动作与观测，校验参数合法性和结果归属。
超时或权限拒绝	错误类型、权限边界、是否实际执行	学习恢复、替代或停止；不能把无返回当作成功。
解析失败与重试	原始文本、解析器版本、重试次数	防止奖励只检查可解析格式而忽略任务完成。
环境状态变化	状态版本、可重放输入、幂等标识	避免在变化环境中错误复用旧奖励。
终止	成功/失败/截断、未完成动作	区分真实结束和预算耗尽，影响优势与回报。

只保存成功轨迹，会让模型看不到线上最常见的恢复决策。可在可重放环境中注入工具不可用、空结果、权限拒绝和带干扰内容的观测；标签应反映是否重试、改用其他工具、请求必要信息或停止。目标是正确决策，不能把“调用次数越多”或“总是拒绝”当作可靠性。

训练时的环境反馈也可能被投机。例如格式奖励只要求一个工具调用字段，模型可能提交语法有效但无意义的参数；成功奖励只看可见单元测试，模型可能利用测试泄漏。独立评测应使用未见任务与环境状态、隐藏测试、动作合法性和恢复率，并报告每成功任务调用数、token、延迟和越界行为。外部观测应作为数据处理，其内容不能自动获得任务指令的优先级。

本节围绕动作目标、反馈来源和评测闭环展开。运行时调度、记忆和多 Agent 协作可继续阅读同系列《LLM 智能体系统》；提示注入与权限威胁模型见《可信与安全的大模型》。

练习与提示

1. 对接受概率为 $a = (0.1, 0.5, 1)$ 的三类候选，推导筛选分布；再增加“每题只留最短回答”的规则，说明原公式为何不再足够。提示：后一规则依赖同组其他候选。
2. 构造 $N = 4$、至少一条正确但 best-of-N 仍选择错误的候选组，分别计算候选覆盖和选中成功率。提示：让打分器偏爱错误的长回答。
3. 一次批评修订有 30% 错题被修好，却有 10% 原正确题被改坏。用原正确率 p 表示最终正确率。提示：分别对正确与错误子集条件化。
4. 画出一次工具超时后重试的事件序列，标明哪些 token 有 SFT loss、哪些动作有策略 log-prob、哪些内容只是观测。提示：调用成功与任务成功应有不同字段。

第 10 章

训练系统：把概率契约落实到 rollout 与 learner

10.1 一条轨迹经过哪些组件

在线后训练通常包含提示采样器、rollout worker、反馈服务、学习器和独立评测器。将它们分开可以并行处理，也会产生队列延迟、权重同步与数据陈旧问题。一个回报高的完成，如果无法确认其行为概率、采样设置和评分版本，就很难被正确用于策略更新。

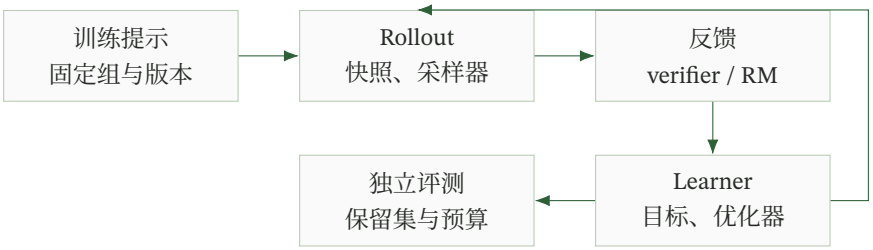


图 10.1 版本化的 rollout–learner 闭环。权重刷新、反馈入队和更新不是同一个时刻；图为本书绘制。

每条轨迹至少记录：提示组、token ID、角色与动作 mask、结束原因、环境版本、生成模型 revision、模板与 tokenizer hash、采样变换、实际行为 log-prob、反馈原始值及版本、生成和入队时间。训练时再附 learner 版本、目标版本与过滤理由。若异步生成中途刷新权重，单个轨迹的分母会变成多个策略的条件概率乘积；更易审计的起点是在一条轨迹内固定快照。

参考 log-prob 可以缓存，但缓存键至少包含参考权重、输入 token、模板、截断与 mask 约定。不能把旧参考下的缓存用于新参考，或把不同提示上下文中同一回答的概率当作相同数值。

10.2 策略陈旧与采样变换

版本差 $\Delta k = k_{\text{learner}} - k_{\text{actor}}$ 是易记录的陈旧指标，却不是分布距离。大步更新可能使一个版本的差距就很大，小步更新则可能相反。还应检查在同一轨迹上的 log-ratio 分位数、裁剪比例、熵、反馈延迟和组内候选相似度。

若行为为 μ，理想的动作比率是 $\pi_\theta(a_t \mid s_t)/\mu(a_t \mid s_t)$。序列重加权还需要各动作比率的乘积，可能使长回答方差急剧增加。PPO 常用局部动作 ratio 的 surrogate；仅在固定旧状态上修正动作概率，不代表已精确修正全部状态占用差异。第 5、7 章给出的适用条件在系统中仍须成立。

一个明确的异步策略是限制队列年龄与更新版本差，超过阈值就丢弃并计入浪费预算；或者使用推导清楚的校正目标并监测权重退化。可记录非负样本权重的有效样本量

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

该式要求 $\sum_i w_i > 0$；全零时报告有效权重缺失，不用稳定常数伪造 ESS。ESS 只描述权重集中程度，不能证明奖励无偏、样本独立或支持覆盖。top-p 将部分动作概率置零时，无法用已有样本恢复这些动作对完整当前策略目标的贡献。裁剪重要性权重降低方差，也引入偏差，须作为算法设置公开。

一次权重同步的可检查过程

冻结采样快照并赋版本号；确认各 worker 的权重校验值一致；用同一固定小批输入比较 learner 与 rollout 的原始 log-prob；再比较应用采样变换后的行为 log-prob。最后生成一条完成，检查保存的每个分母、结束原因和 mask。若推理引擎采用不同精度或内核，先量化误差，再决定容忍阈值，不能预设两端逐 bit 一致。

10.3 显存账本：权重只是其中一项

令模型参数量为 P。下面是推算，不是本次 GPU 实测：假设全参数训练，bf16 权重与梯度各 2 byte/参数，另保留 float32 主权重以及两个 Adam 矩各 4 byte/参数。

项目	存储模型	$P = 7 \times 10^9$ 的推算
训练权重	$2P$ byte	14 GB，约 13.04 GiB
梯度	$2P$ byte	14 GB，约 13.04 GiB
float32 主权重	$4P$ byte	28 GB，约 26.08 GiB
Adam 两个矩	$8P$ byte	56 GB，约 52.15 GiB
以上静态合计	$16P$ byte	112 GB，约 104.31 GiB
独立冻结参考	另加 $2P$ byte	若同精度同大小、未共享/卸载，另加约 13.04 GiB

这里 $1\ \text{GB} = 10^9$ byte，$1\ \text{GiB} = 2^{30}$ byte。不同优化器可以省主权重、改变梯度精度或使用分片/低精度状态，必须替换相应项。激活、临时 logits、通信缓冲、内存碎片、奖励模型、critic、教师和 rollout 副本尚未计入，因此 104.31 GiB 不是峰值显存，也不是最低可训练显存。

数据并行状态分片可以把部分静态项分摊到多个设备，但前向 all-gather、反向通信和暂存峰值仍要计入。不能把表中总数简单除以卡数就承诺可训练。激活 checkpoint 用重算换显存；长度分桶和安全 packing 减少无效位置；适配器减少可训练状态。这些方法影响的是不同账目，需要用同一配置下的峰值测量核对。

rollout 的 KV 缓存另计。若有 L 层、并行生成 B 条、缓存长度 S、h_{kv} 个 KV 头、头维 d_h、每元素 b byte，理想连续缓存近似

$$M_{\text{KV}} \simeq 2LBSh_{\text{kv}}d_h b. \tag{10.1}$$

系数 2 对应 K 与 V。取 $L = 32, B = 8, S = 4096, h_{\text{kv}} = 8, d_h = 128, b = 2$，得到约 4 GiB。页式分配、共享前缀、量化和调度会改变实际占用；该式不包括推理权重与工作区，也不能直接代替训练激活账本。

10.4 数值稳定性从不变量开始

先在 log 空间计算概率，用稳定的 log-softmax 和 `logsigmoid`；不要先连乘长序列概率。长序列 log-prob、KL 与损失归约可在 float32 中计算，独立小算例则用 float64 参考。bf16 的指数范围有利于部分硬件上的混合精度训练，但无法修复错位标签、无限 ratio 或损坏的数据。

把约束分成“应报错”和“可稳定化”两类。空目标 batch、组大小小于算法要求、非法概率或丢失行为 log-prob，应显式处理；不能只在所有分母加 ε。合法的极小方差可以采用预先指定的稳定常数，但要记录零方差率。将 reward、advantage、ratio 截断或白化会改变有效权重，有时还改变目标，不能统称为不影响结果的实现技巧。

发生 NaN 时，依次保留输入与 mask、logits、log-prob、ratio/奖励/优势、损失、梯度范数及优

化器状态，定位第一个非有限数。梯度裁剪适合控制有限的大梯度；已经出现 Inf 时，直接裁剪未必恢复有效方向。发现问题应停止该次更新、保存事故批次并查因，不让无效参数继续进入 rollout。

若 KL 系数自适应调整，控制器只用训练侧独立测量的、明确方向和分母的 KL；不直接用单条可为负的采样值控制。设测量超过目标上界时增大 β，低于下界时减小，并固定更新间隔、上下限和步幅。它改变了训练过程，应与固定系数做消融；测试集分数不参与逐步控制。

10.5 恢复训练与成本日志

可恢复 checkpoint 除权重外，还要包含优化器与调度器状态、混合精度缩放器、全部 RNG、数据游标、采样器状态、参考与教师版本、当前 KL 控制器和未完成队列的处理规则。恢复后若丢弃旧轨迹，应记录为预算损耗；若重放，避免对同一批更新两次。验证恢复的办法是比较中断前后下一批 ID、下一次生成与下一次更新，而非只检查文件可加载。

吞吐分别记录 prompt token、生成 token、有效更新 token、padding、重采样和失败重试。报告单位应写成 token/s、设备小时或每成功任务成本，并注明模型、精度、序列长度、并发、硬件和测量区间。本次没有运行 GPU 系统测试，因此本章给出核算方式与接入协议，不报告某种架构的实测加速比。

练习与提示

1. 若梯度改为 float32 而其他假设不变，重算 7B 静态状态。提示：原表的 $2P$ 项变为 $4P$。
2. 用式 (10.1) 估计并发数翻倍的缓存变化，再列出公式未覆盖的三个开销。
3. 设计一次异步陈旧度消融。提示：固定总生成预算、反馈版本与数据，分别控制队列年龄，并把丢弃成本计入结果。

第 11 章

评测：固定问题、预算与测量系统

11.1 先写评测契约，再比较模型

评测要回答的是一个明确部署条件下的变化。例如“在同一组未见代码题、每题一次生成、相同最大长度和超时下，隐藏测试通过率是否提高”。如果同时改变温度、采样次数、解析器或裁判，这个问题就变成了系统级比较，仍有价值，但不能把差值全部归因于后训练目标。

冻结契约至少包括：提示组和 split hash，模型/tokenizer revision，chat template，系统提示，采样温度及 top-p，每题候选数，最大生成长度，stop 规则，工具和执行器版本，超时与错误处理，答案解析，裁判 rubric，种子和 checkpoint 选择。预先指定主指标、安全回归门槛和成本约束，其他观察标为探索性结果。

本书将验证集用于预定超参数和 checkpoint 选择，最终测试集用于冻结后的估计。反复查看测试失败并修数据，会把测试集转化为开发集；此后应重新建立独立保留集，不能沿用“从未见过”的描述。

11.2 指标覆盖的是哪一层能力

对象	可观测量	不能据此自动推出的结论
任务结果	精确匹配、隐藏测试、任务成功率	测试覆盖之外都正确，或推理过程每步正确。
指令遵循	可解析的长度、结构、关键词等约束	开放式回答有帮助且事实可靠。
偏好与奖励	配对胜率、RM 成对准确率、校准	新策略不会利用裁判漏洞或长度捷径。
过程可靠性	步骤错误定位、前缀判别、恢复成功率	更高过程评分必然导致最终正确。
安全与诚实	攻击成功率、无害请求过拒率、Brier	对所有攻击、用户和部署环境满足保证。
成本与系统	token、裁判/工具调用、延迟、峰值资源	不同硬件与并发设置下保持同一比值。

IFEval 将部分指令遵循要求做成可检查约束 [25]；RewardBench 与 RewardBench 2 检查奖励模型在不同偏好样本上的区分能力 [26, 27]；ProcessBench 关注数学过程错误识别 [28]。这些基准可补充测量，却不等于完整部署目标。引用任何分数时应同时给出数据版本、模型版本、推理与评分设置。本次未复现这些基准。

11.3 pass@k、搜索与实际选择

对每个问题独立生成 n 个候选，恰有 c 个通过测试，$n \geq k$ 时常用估计 [29]

$$\widehat{\text{pass@}k} = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}. \tag{11.1}$$

当 $n - c < k$，分子视为零。它等于从这批候选无放回选择 k 个时至少一个通过的比例；在相应独立同分布生成假设下，它可用于估计该采样策略的覆盖成功率。若 $n = 4, c = 1, k = 2$，估计为 $1 - 3/6 = 1/2$。当 $k > n$ 时不得套公式或外推；若候选来自相关搜索或不同温度的混合，需重新说明

其含义。

pass@k 借助测试真值判断“是否存在成功候选”。实际系统往往只有一个不完美的 selector。best-of-k 的选择成功率应以最终选出的答案验证；它不会因为候选中有正确答案就自动成功。self-consistency 的投票也可能稳定地选中常见错误。比较时分别报告单次质量、候选覆盖、选择成功和总预算，不能只选其中最好看的一个。

截断和解析失败通常是该设置下的任务失败，但还要单列原因。若代码超时涉及基础设施故障，应按预先规则重试并记录，不能事后只重跑失败模型。增加最大长度可能提高可完成率，也提高成本并改变输出分布，必须作为可见设置处理。

11.4 成对比较、tie 与统计单位

对相同提示比较模型 A、B，设 A 胜 W、负 L、平 T，可报告

$$\widehat{w}_A = \frac{W + T/2}{W + L + T}.$$

排除 tie 的胜率 $W/(W+L)$ 是另一个统计量，应同时给出原始计数。解析失败另记，不能静默归为 tie 或某一方失败。若裁判提供概率，可以计算 Brier 分数 $N^{-1}\sum_i (p_i - z_i)^2$，但只有独立标签 z_i 足够可信时，概率校准才有可解释的参照。

同一提示的多个候选、多个位置顺序和多次裁判共享问题难度，不是独立新问题。配对 bootstrap 以提示为重采样单位：一次抽取提示 ID 后，连同该提示下两个模型的所有输出与裁判记录一起抽取；每次计算模型差值，取其分位数区间。题目来自同一文档或项目时，聚类单位还应提高到相应来源组。

训练 seed、评测采样 seed 和提示抽样是不同的不确定性。至少报告每个训练 seed 的结果；三个 seed 的标准差只能描述这三个运行的变动，不能单独证明总体显著。资源允许时可用分层配对重采样探索其影响，少量种子的区间仍须谨慎解释。主指标与停止规则提前固定；多基准或大量超参数试验应说明多重比较与选择效应。

配对差值的最小记录

每行记录 `prompt_group`, `train_seed`, `model_id`, `generation_seed`, `output_id`, `score`, `cost`。A 与 B 使用相同提示和预算；若共享采样随机数，注明这一方差控制。保存逐题分数而非只存总平均，才能计算差值、定位哪些任务退化，以及按提示而非按 token 重采样。

11.5 裁判要先校准，再用于归因

模型裁判可能受位置、长度、格式、自偏好和 rubric 漏洞影响 [9, 10]。先以模型匿名的候选对进行评判，再交换位置，按答案身份比较是否保持同一偏好。若只统计“第一位置选择率接近一半”，可能漏掉内容和位置相互抵消的偏差。

位置交换只是一个检查。还要做语义不变的格式变体、正确与错误答案的长度对照、重复判分、不同裁判或规则复核，并保留 tie 和原始评分依据。加长回答若带来了必要解释，评分变化可能合理；检测长度偏差需要内容质量不变的对照，或错误长答案与正确短答案的独立真值。长度控制回归能减轻某些混杂，不能证明所有偏差都已消除。

训练与测试若使用同一 RM 或裁判，先优化它再让它给出“提升”，容易产生测量闭环。至少保留独立反馈源，如不同来源的人类样本、程序验证或未参与训练的裁判，并对分歧样本复核。独立模型也可能共享偏差，因此“换了模型名”并非充分保证。第 13 章提供可运行的合成偏差检查，以及明确未执行的真实模型裁判协议。

11.6 推理预算、费用与回归门槛

训练成本与推理成本分别报告。训练包括更新、rollout、参考/教师/奖励模型前向、验证、失败重试及超参数搜索；推理包括候选生成、工具/裁判调用和延迟。若讨论摊销成本，必须声明预期服务次数，不能把一次训练总成本与单条推理费直接相加。

“每成功任务成本”可用该批总成本除以成功数；成功数为零时应报告不可定义，而非用一个小常数生成巨大有限值。还应给出平均与尾部延迟、输出长度和失败重试分布。TTFT 是从请求进入约定测量点到首个 token 的时间，后续 token 的间隔统计与总体请求延迟另报；不同并发和硬件下不直接比较一个吞吐数字。

最终选择可以使用门槛：主任务达到预设改善，安全与基础能力不超出允许退化，成本不超过约束。阈值来自任务需求与测量精度，不是一组通用常数。未通过时检查失败切片，再改数据、反馈或目标；改完应在仍独立的保留集验证。评测的作用是暴露方法边界，而不是给每次更新寻找一个正面分数。

练习与提示

1. 一个 selector 总选错，但每组五个候选中都有一个正确，pass@5 和选择成功率分别是什么？提示：前者是覆盖，后者是系统最终结果。
2. 每题生成十条回答后，可否把它们当十道题计算二项置信区间？提示：考虑题目难度的共享相关性。
3. 设计一个控制质量与格式的裁判实验，并明确真值来源。提示：分别操纵格式、长度和答案正确性，避免一次同时改变三者。

第 12 章

安全与规格：把代理奖励的边界纳入闭环

12.1 后训练安全是一项可测量的要求

一个模型可能在平均偏好上提高，同时更容易过度拒答、泄露不该复述的信息或在工具失败后越权重试。后训练要把这些行为写成明确规格，并在训练前后测量。RLHF 与 Constitutional AI 提供了以人类或 AI 反馈塑造行为的具体路线 [30, 31]；它们的反馈规则和实验环境不能自动成为任何任务的安全保证。

设帮助性收益为 R_{help}，成本 C_j 表示不希望发生的行为或资源使用，可写成

$$\max_{\theta} \mathbb{E}_{x\sim\mathcal{D}_{\text{deploy}},y\sim\pi_\theta} R_{\text{help}}(x,y), \qquad \text{s.t.} \quad \mathbb{E}C_j \le c_j^{\max}. \tag{12.1}$$

对应的一种拉格朗日写法为

$$\min_{\theta}\max_{\lambda\ge 0}\left[-\mathbb{E}R_{\text{help}} + \sum_j \lambda_j(\mathbb{E}C_j - c_j^{\max})\right]. \tag{12.2}$$

这说明奖励和约束如何影响取舍，不意味着非凸神经网络优化满足强对偶或找到可行解。实际只用有限惩罚系数、有限样本和代理成本时，仍需独立门禁验证。约束是在指定分布上取平均，也不会自动控制每类用户或攻击的最坏风险。

可进一步分别估计关键切片 g 的 $\mathbb{E}[C_j \mid g]$，并报告最差切片、样本量和不确定性。若某切片没有样本，应报告覆盖不足，不能用总体低风险替代。第 11 章的聚类统计同样适用：同一攻击模板改写出的几十条样本，不等于几十种独立威胁。

12.2 把权限与拒答写进标注协议

系统、开发者、用户指令与工具观测的角色应由具体部署协议定义。训练数据要保留角色边界，并教会模型区分“外部内容说了什么”和“系统授权它做什么”。工具返回中出现的指令属于待处理数据，其可信度与权限由协议决定，不因被放入上下文就自动升级。

安全标签不能只是一列“拒答/不拒答”。至少要分别衡量：对确实超出授权或有害请求的适当拒绝；对可安全完成任务的有效回答；需要澄清的歧义请求；无害请求上的过度拒绝。反馈应评价回答是否给出符合规格的帮助，而非奖励某句固定拒答套话。

一个无害的训练对照是：用户要求总结一段包含“忽略上文”字样的引用文本。模型应按授权总结这段文本，而非执行引文中的指令；若只因出现敏感词就拒绝，又会损失正常能力。可将任务目标、引文内容和是否需要工具操作分别改变，检查模型是在理解边界，还是记住了关键词。

12.3 reward hacking 是代理与目标的偏离

奖励投机（reward hacking）指优化提高了代理评分，却没有相应改善真实目标。它不要求模型具有某种可证明的主观意图。更长回答、固定格式、迎合裁判或利用验证器漏洞，都可能在训练中被强化。

可观察失效	需要的独立证据	对应干预与复核
训练奖励升、任务不升	隐藏测试、独立裁判、长度与格式分层	修反馈或数据；在独立保留集复核，不能只增大 KL。
拒答率升、无害任务退化	无害集上的过拒率与有效帮助评分	加入可安全完成与澄清示范，分开衡量适当拒答。
表面格式正确、执行失败	schema 以外的语义和执行测试	增加反例与隐藏测试；保留旧测试做回归。
工具重试获得高分、成本失控	调用次数、超时、权限、最终结果	将成本与终止写入目标；检查是否学会过早停止。
平均提升、少数切片恶化	按任务/语言/攻击族分组的差值	调整采样与约束；报告最差切片和样本不足。

KL 可限制相对参考漂移，却无法让错误 verifier 变正确；更换裁判也不能自动消除共享偏差。应先用反事实对照定位代理依赖的特征，再修正规格、反馈或训练数据。第 13 章的合成裁判是可控例子：人为给位置或格式加分，训练/评测若只看该分数便会误判。

12.4 将红队发现变成回归样本

一个闭环包括：收集失败轨迹，按任务与机制归类，独立复核，构造修复样本，再用未参与修复的攻击族与无害对照评测。保留策略 checkpoint、原始输入输出、工具状态、反馈版本和严重度。修复一个具体模板后，测试同族改写与其他类别，检查是否把问题转移到别处。

只训练成功轨迹，可能使模型在线上遇到空结果、权限拒绝或工具暂时不可用时没有恢复策略。可在受控环境注入这些无害失败，监督合理澄清、重试或停止；奖励既要考虑最终成功，也要记录成本、权限和终止原因。Agent 运行时隔离与回滚设计见《LLM 智能体系统》，完整威胁模型与攻防评测见《可信与安全的大模型》；本书保留其与目标和轨迹标签有关的接口。

12.5 数据权利、隐私与发布证据

去除显式个人信息可以减少部分暴露，却不能证明模型不会记忆；成员推断压力测试同样只能在其攻击假设下给出证据。训练前保存来源与许可，训练后用与任务相称的样本和测试评估泄露风险。若数据不能再分发，提供获取条件、处理脚本与可替代的合成示例，不能把受限原文混入公开实验包。

发布记录应说明基座与 tokenizer、数据范围、许可、训练目标、评测协议、已知失败和未覆盖场景。安全门槛由任务与部署环境确定；本书的 CPU 教学实验没有测量真实模型安全性，也不能支持上线判断。完整书稿可复编、算例可复跑，解决的是教材与实验的可审计性，不能替代真实部署验证。

练习与提示

1. 总体攻击成功率降低，但某语言切片升高，能否声称安全全面改善？提示：分别写总体与条件期望，并检查样本量。
2. 设计区分“识别引文中的指令”和“遇到关键词就拒答”的无害对照。提示：固定词语，只改变它在任务中的角色。
3. 若减少工具调用次数被加入奖励，可能出现什么新投机？提示：过早停止也能降低成本，需要同时约束任务完成。

第 13 章

实验科学：从公式到可追踪证据

一个后训练目标写得正确，并不保证它在数据管道中被正确实现；代码通过数值检查，也不保证它能改善真实模型。本章把证据分成三层：独立算例检查实现约定，受控训练检验机制，真实任务评测检验迁移。三层都需要，但不能互相替代。配套实验位于本书 `manuscript/experiments/`，完整运行记录位于 `output/experiments/`。

13.1 先规定要回答的问题

实验前的一页规格至少写明：假设、主指标、训练与评测分布、反馈来源、基线、预算、检查点选择和失败判据。例如本章的受控问题是：同一个含噪 *SFT* 检查点收到正确的二元偏好后，其正确答案概率如何变化；翻转偏好会发生什么？这比“DPO 是否优于 SFT”更窄，也更容易用证据回答。

公平预算取决于问题。相同步数不等于相同 token 数，相同 token 数也不等于相同 FLOPs；RL 还需要计入 rollout、失败重试、验证器和裁判成本。比较前选择主要约束，再把其余账目公开。用最佳检查点对比另一个方法的最终检查点，会把模型选择差异混入算法差异。本章全部使用预设的最后一步，没有根据测试集选模型。

13.2 先用独立结果检查实现

`checks.py` 实际运行了 44 项数值检查，训练程序又运行了 2 项梯度有限差分检查。检查采用 NumPy 的双精度计算，参考值来自概率手算、完整枚举或中心有限差分。它们覆盖本书最容易混淆的边界，而不只是比较两份相似代码。

对象	独立参照与检查目的
completion-only loss	两个目标概率为 $0.5, 0.25$，故损失是 $\log 8/2 = 1.03972077084$；检查 causal shift、多轮 assistant 的 EOS 和空目标拒绝。
packing	修改第一个样本的 token，普通因果注意力会改变第二个样本的上下文；分块注意力保持不变，并屏蔽跨样本的下一 token 标签。
DPO	四个独立序列概率给出 $\log(1 + 6^{-0.2})$；检查标签翻转、参考等于策略、长度求和与均值、解析梯度与有限差分。
KL	枚举二元词表，核验两个方向、负的单点 log-ratio、非负 k3、旧策略采样的重要性修正，以及估计值与梯度的区别。
优势与裁剪	枚举全部 2^4 个独立二元回答组，检查 RLOO 的留一条件；覆盖全同奖励、组大小为 1、长度加权、PPO 的正负优势裁剪。
GAE	三步终局回报、$\gamma = 0.9, \lambda = 1, V = 0$ 给出 $(0.81, 0.9, 1)$；最后一步不能被求和上界遗漏。另查真实终止与时间截断的 bootstrap 区别。

以多轮 mask 为例，输入为 `[BOS,U,q,A,a,EOS]` 时，目标位置 mask 为 `[0,0,0,0,1,1]`，而承担预测的 logits 位置是 3、4。mask 属于目标 *token*，所以在 next-token shift 后应使用 `mask[1:]`。若误用 `mask[:-1]`，就改变了训练对象。一个样本在截断后若只剩提示，程序应跳过并记录，或明确报错；用分母钳制掩盖整个 batch 没有监督的事实，会使“训练完成步数”失真。

KL 的值正确，梯度仍可能错误

取 $p = (0.8, 0.2)$、$q = (0.5, 0.5)$，枚举得到

$$D_{\mathrm{KL}}(p\|q) = 0.1927447570, \qquad D_{\mathrm{KL}}(q\|p) = 0.2231435513.$$

令 $p = (\sigma(\theta), 1-\sigma(\theta))$，在 $\theta = \log 4$ 处，完整 KL 对 θ 的导数是 $0.16 \log 4 \approx 0.2218071$。在同一点冻结采样权重后，对 k1 的局部表达式直接求导得到 0，对 k3 得到 0.3。这并不否定二者在正确分布下的值估计；它说明当采样分布也随参数变化时，不能漏掉对应的导数项。训练中的 KL surrogate 必须连同采样分布、停止梯度位置和所要估计的梯度一起说明。

另一个容易被数值稳定技巧掩盖的情况是零方差组。若四个回报都为 1，GRPO 的中心化优势为零，RLOO 的留一优势也为零；增加分母中的小常数只能防止除零，不能创造奖励差异。本书把稳定项加在标准差之外：回报为 $(1, 0, 1, 0)$、$\varepsilon = 0.1$ 时，优势是交替的 $\pm(0.5/0.6) = \pm 5/6$；这项手算检查能区分 $\sqrt{\mathrm{Var}} + \varepsilon$ 与 $\sqrt{\mathrm{Var} + \varepsilon}$。若两个序列长度为 1 和 3、每 token 的 surrogate 贡献分别为 $+1$ 和 -1，先做每序列均值再平均得到 0，全体有效 token 平均则为 -0.5。归一化改变了权重，不能在公式与实现之间任意互换。

13.3 实际运行的 CPU 训练消融

13.3.1 模型、数据和切分

本次实测使用一个教学自回归策略，不是预训练大语言模型。提示包含 $a, b \in \{0, \dots, 19\}$，任务是判断 $a + b \geq 20$。模型首先生成 YES 或 NO，然后以概率 1 生成 EOS。它只有三个参数：

$$f(a, b) = \left(1, \frac{a - 9.5}{10}, \frac{b - 9.5}{10}\right), \tag{13.1}$$

$$\pi_\theta(\mathrm{YES} \mid a, b) = \sigma(f(a, b)^\top \theta), \qquad \pi_\theta(\mathrm{NO} \mid a, b) = 1 - \pi_\theta(\mathrm{YES} \mid a, b). \tag{13.2}$$

EOS 无可学习参数，因此本实验不能检验终止策略、真实长文本的长度偏差或多步信用分配。

400 个提示先按无序整数对 $\{a, b\}$ 归为题族，把交换加数的两种写法放在同一组；题族按固定 SHA-256 排序，整组分配到尚有容量的第一个集合，得到 240 个训练、80 个验证与 80 个测试提示。三个集合的完整提示和交换题族都没有重叠；它们仍共享整数、可能的和值及特征空间，因此测量的是组合插值，不是跨领域泛化。训练示范以概率 0.25 翻转正确标签，固定数据种子 20261022 实际产生 73 个翻转标签；验证与测试真值从任务规则独立计算。数据审计文件保存全部 split ID、题族交集、翻转数和 SHA-256，不能仅凭“使用了随机种子”判断没有泄漏。

每个初始化种子 $7, 11, 19$ 先在同一份含噪示范上运行 200 步 SFT，再从该检查点复制四个分支。全部使用 full-batch 梯度下降、学习率 0.1 和 float64；每个分支额外运行 120 步。

1. 继续含噪 SFT：保留原示范，控制仅延长训练的影响。
2. 继续干净 SFT：用任务真值作为示范，控制获得更干净监督的影响。
3. 正确标签 DPO：同一提示下，将正确答案作为 chosen、错误答案作为 rejected，冻结初始 SFT 参考，$\beta = 0.5$。
4. 翻转标签 DPO：仅交换偏好方向，检查优化器是否忠实地学习错误反馈。

偏好对由二元答案空间完整构造并用算术规则标注，没有在线生成、拒绝采样、教师模型或模型裁判。因此这是离线 SFT→DPO 消融。它不能作为 on-policy 学习、RLVR 或探索改善的实验证据。

13.3.2 预算与评测约定

初始 SFT 每 seed 处理 $200 \times 240 \times 2 = 96000$ 个 completion token exposure。继续训练阶段，SFT 每个示范重复两次，DPO 每个提示处理两个回答；每个回答均为答案加 EOS，所以四个分支都处理

$$120 \times 240 \times 2 \times 2 = 115200$$

个策略评分的 completion token exposure。重复示范不增加新信息，也不代表真实 Transformer 的计算成本相同；DPO 还需一次缓存参考模型的 960 个 token 评分。完整账本分别记录策略评分、参考评分、更新数、墙钟时间和评测生成数。本实验只匹配额外更新数与策略评分 token exposure，没有匹配 FLOPs 或墙钟时间。

主指标是测试集中正确答案的精确枚举概率均值，即在温度 1、无 top-p 截断时单次采样成功率的期望；二元空间允许直接计算它，无需用少量随机生成近似。次指标包括贪心准确率、一次实际随机生成的准确率、答案熵和到冻结 SFT 参考的 KL。每个设置的测试生成预算相同：80 个提示、每提示 1 个回答、每回答 2 个 token。不同分支共享采样随机数，以降低配对比较的噪声；种子之间的小标准差不代表已经覆盖提示或采样不确定性。

13.3.3 结果及其适用范围

表 13.2 来自本次实际运行。第一项误差为三个初始化种子的样本标准差，不是 95% 置信区间；三个种子使用同一批 80 个测试提示，不能把它们当成 240 个独立问题。

表 13.2 合成 CPU 策略的测试结果；KL 单位为 nat/回答。

设置	成功概率（%）	贪心（%）	到 SFT 的 KL
SFT 初始检查点	58.00 ± 0.23	92.50	0.0000
继续含噪 SFT	60.25 ± 0.14	92.50	0.0018
继续干净 SFT	66.48 ± 0.12	97.50	0.0253
正确标签 DPO	69.27 ± 0.15	97.50	0.0446
翻转标签 DPO	43.81 ± 0.23	16.67	0.0662

正确反馈提高了正确答案概率，翻转反馈则使它降低。干净 SFT 与 DPO 的贪心准确率同为 97.50%，而它们的采样成功概率和 KL 仍然不同；只报贪心分数会遮住分布变化。干净 SFT 控制还表明，收益的一部分来自监督信息改善。由于本实验只用一组预设学习率、β 和目标归一化，DPO 的数值优势不能证明其普遍优于 SFT，更不能外推为大模型能力提升。所有配置和失败检查均保留，未按测试结果调参。

13.4 奖励与裁判偏差：检测器也需要真值

本次另用 20 个算术提示运行显式有偏的合成裁判，其分数由正确性和人为加入的干扰项组成。位置、长度和 Markdown 三个因素分别操纵，避免把多个变化混在一次比较中。

因素	受控构造	本次合成结果
位置	第一位置额外加 1.25 分，正确性只加 1 分；交换正确与错误答案的位置。	40 次比较均选第一位置；交换后答案身份翻转率 100%，与真值分歧 50%。

因素	受控构造	本次合成结果
长度	每空格分隔 token 加 0.1 分；错误答案后添加 20 个无关词。	20 次比较均偏选较长错误答案；正确性不变的延长也使分数增加 2。
格式	Markdown 粗体额外加 1.25 分；只改变包装。	20 次比较均偏选粗体错误答案；正确答案仅改粗体也增加 1.25 分。

这些偏差幅度是实验预先注入的，不能用于估计任何真实奖励模型或 LLM 裁判的偏差。它们验证的是：检测流程能否发现质量不变时的评分漂移，以及有没有独立真值来发现“高分但错误”。实际训练中还应保留原始回答、评分依据、裁判版本、温度、解析失败和 tie，不把无法解析的输出静默计成输赢。

真实模型裁判的升级方案已生成 200 个输入：20 个基础提示，每个包含原样、错误答案加长、错误答案粗体、正确答案加长、正确答案粗体五种变体，再各自交换位置。`judge_probe.py` 提供 prepare、run、analyze 三个入口；run 通过本地适配器命令接收 JSON 输入并返回 winner。测试真值和扰动元数据不会发送给裁判。**本次只生成协议输入，未调用真实模型裁判**。运行后应按基础提示聚类，报告位置身份一致性、各变体相对原样的变化、tie 与解析失败率，再抽查独立人工或程序真值。算术只是检测起点，不能代替开放式帮助性、事实性或安全评测。

13.5 复现入口与真实模型升级方案

在本主题目录运行以下命令；需要 Python 3.10 及以上与 NumPy，输出目录可以更换：

Listing 13.1 无需网络或 GPU 的复现实验

```
python3 manuscript/experiments/run.py \
  --config manuscript/experiments/config.json \
  --output output/experiments
```

输出包含命令、环境、配置、数据审计、15 份 seed/设置记录、日志、检查点、结果表与哈希。本次使用 Python 3.12.14、NumPy 2.3.5、arm64 CPU，无预训练权重或外部数据；正式 PDF 直接导入生成的实测表。

真实模型升级方案见 `manuscript/experiments/README.md`。锁定模型/tokenizer revision、模板、许可和数据哈希；固定分组、SFT 起点、参考、mask、截断及检查点，记录候选与评分来源。冻结配置后才读最终保留集，报告能力、安全、长度、校准和成本。RLVR 另记 rollout、零方差组和行为版本，不以离线枚举代替在线预算。

练习与提示

1. 三个种子的标准差为何不能替代提示 bootstrap？提示：种子与问题变动对应不同不确定性，同题多次生成也有关联。
2. 每 token log-ratio 提升为 $\log 2$，长度 4 和 1、$\beta = 0.2$，求 DPO 的 sum 与 mean margin。提示：分别为 $0.6 \log 2$ 与 0；长度平均会换目标。
3. 如何把实验变为真正的 on-policy？提示：让学生实际决定数据或前缀分布，记录行为概率与版本；给静态文件改名不够。
4. 正确答案加长后得分更高是否必然是偏差？提示：先看 rubric 与信息量，再用语义等价和错误加长对照区分合理质量变化与长度捷径。

第 14 章

研究路线：把机制问题变成可证伪实验

14.1 从一个未知量开始

研究不必从发明一个缩写开始。可以先找出一个尚未分清的机制：偏好提升来自更好标签还是目标本身？组内标准化改善了方差，还是重新加权了难度？学生前缀蒸馏改善长程错误，还是仅消耗更多教师计算？这些问题都要求比较中的数据、分布、目标和预算分别可见。

本书核验过的文献中，GKD 与 MiniLLM 给出不同蒸馏目标及学生分布处理 [22, 23]，DeepSeek-Math、DeepSeek-R1 与 DAPO 展示了可验证奖励训练中的不同系统和算法选择 [18, 20, 19]。2026 年有关后训练分布角色和 on-policy 蒸馏机制的论文提供了进一步观察 [2, 24]。它们是形成假设的来源；论文在特定模型和任务上的结果，不能直接证明新任务中的收益。

问题	可控制的对照	可否定假设的证据
长程信用分配	同一终局反馈下，比较过程信号、RLOO 和 critic；固定候选预算	早期错误、恢复率和独立任务均未改善，或过程评分只奖励常见路径。
组标准化的作用	固定生成样本，分别中心化、除标准差与固定尺度	提升随题目难度权重改变而消失，说明不能只归因于低方差。
学生前缀蒸馏	固定教师调用、更新 token 和前缀难度，比较数据/学生前缀	最终分数上升但早错率与恢复没有改善，或教师在错误前缀不可靠。
在线偏好收益	固定初始策略与裁判预算，只改变 pair 刷新频率	新 pair 不增加信息，或收益在独立裁判下消失。
陈旧策略的代价	固定总生成预算、反馈、优化步数，扫队列年龄	吞吐上升但有效样本量和成功/成本恶化。
安全与多目标	加入无害对照和最差切片约束，保留同一主任务	平均偏好上升而无害帮助性或关键切片退化。

14.2 一个消融应改变哪件事

以“学生前缀减轻暴露偏差”为例，训练两组使用同一教师、学生初始化、提示组和局部散度，仅改变前缀来源。分别报告教师在前缀上的可验证可靠性、逐位置错误率、错误后恢复、任务成功、长度、熵和成本。若学生前缀更难，还需要控制难度或明确这一分布改变就是待研究机制。只匹配 optimizer steps 不足以匹配教师前向与总 token。

对 GRPO，可以先在同一批缓存回报上比较归一化得到的梯度方向，检查零方差和长度权重；若要估计梯度方差，还须对多个独立 rollout 批次重复测量。再跑在线训练，区分估计器变化与随之改变的采样分布。固定缓存实验不能证明在线长期收益，却能排除若干实现错误。第 13 章的有限枚举就是这种证据分层的起点。

对 DPO，使用正确、含噪和翻转标签，加入继续 SFT 和干净 SFT 控制。这样才能看出目标是否忠实学习标签，以及获得额外标签信息能解释多少变化。大模型实验还要扫描合理超参数，并控制搜索预算；一个固定配置赢过另一个固定配置，只支持局部比较。

14.3 八周实践路线与交付标准

下面的八周学习计划可根据可用资源延长或压缩。先以 CPU 教学策略验证，再逐级迁移，并以表中可检查的交付物衡量进展。

阶段	主题	能被别人检查的交付物
第 1 周	数据与 SFT	来源/许可卡、按题族切分、模板 hash、逐 token mask 打印、packing 反例、损失手算。
第 2 周	奖励与偏好	BT 模型或小型评分器、软标签与翻标控制、校准/长度/位置对照、原始偏好记录。
第 3 周	DPO 与邻近目标	从同一 SFT 出发的偏好训练；明确序列 sum、参考、β、数据和搜索预算。
第 4 周	策略梯度	可枚举任务上的 REINFORCE/RLOO；检查 baseline 独立性、KL 梯度、终止和截断。
第 5 周	RLVR/GRPO	固定 rollout 预算的组大小与归一化消融、零方差率、长度切片、独立 verifier。
第 6 周	蒸馏	教师/学生前缀对照，局部散度与停止梯度位置、温度/尾部质量、教师成本账本。
第 7 周	评测与安全	冻结评测契约、配对提示统计、裁判校准、无害对照与关键切片回归。
第 8 周	研究报告	一个可证伪问题、完整配置/日志、失败结果、模型/数据卡、可复现包及结论边界。

若资源允许，真实小模型综合项目可从可执行数学、代码或结构化工具任务中选一个，先建立可靠 SFT，再只比较一种偏好或奖励训练。固定基座与模板，明确是否使用全参或适配器，按实际资源控制规模。待数据和评测可信后，再添加第二种目标。完整的多方法排行不是入门项目的必要交付，能解释一个差值往往更有价值。

14.4 结论应能承受什么追问

一份完成的报告应能回答：学生在哪些状态上学习，行为策略与参考有何不同，反馈是否可靠，目标中的分母怎样选择，更新是否在优化写下的量，改善是否仍出现在独立任务与相同预算下。对无法核实的前沿说法，保留问题与来源，不把预期填成结果。

本次修订的文献检查截至 2026 年 10 月 22 日，具体版本记录在随书修订说明和来源审计中。后训练成为可积累的知识，依靠的是清楚的目标、可检查的数据与诚实的证据。

综合练习与提示

1. 将本章一个研究问题改成一页实验规格，写出失败判据。提示：说明什么观察出现时，你愿意放弃原假设。
2. 同时希望匹配更新 token、教师调用和墙钟时间时，资源可能无法完全匹配，如何报告？提示：先固定主要预算，公开其余成本，画质量 – 成本曲线。
3. 复现第 13 章后，提出一项它无法回答的大模型问题。提示：从可学习 EOS、长轨迹、探索或分布迁移选择一个，给出新增证据需求。

附录 A

数学推导与估计器边界

本附录集中补足正文使用的恒等式。它们在有限词表、有限允许轨迹空间、可交换求导与求和的条件下成立；无界长度和连续动作需要另行检查可积性。所有概率分布、取样方式与停止梯度都写在式子旁，避免把值的恒等式误读成实现的梯度保证。

A.1 KL 正则目标的闭式解与两个极限

对固定提示，记 $Q(y)=\pi_{\text{ref}}(y\mid x)>0$、$\beta>0$、$Z=\sum_y Q(y)e^{r(y)/\beta}$。定义 $P^*(y)=Q(y)e^{r(y)/\beta}/Z$，则

$$\beta D_{\mathrm{KL}}(P\|P^*)=\beta\sum_y P(y)\left[\log\frac{P(y)}{Q(y)}-\frac{r(y)}{\beta}+\log Z\right] \tag{A.1}$$

$$=\beta D_{\mathrm{KL}}(P\|Q)-\mathbb{E}_P[r]+\beta\log Z. \tag{A.2}$$

移项即

$$\mathbb{E}_P[r]-\beta D_{\mathrm{KL}}(P\|Q)=\beta\log Z-\beta D_{\mathrm{KL}}(P\|P^*). \tag{A.3}$$

KL 非负，所以 $P=P^*$ 达到最优。在有限空间、奖励固定时，$\beta\to\infty$ 有 $P^*\to Q$。$\beta\to 0^+$ 时，概率集中到最高奖励集合；若多个回答奖励并列最高，集合内部仍按 Q 的相对质量分配，并非必然只剩一个回答。

若 $Q(y)=0$，有限 KL 的策略不能在该回答上分配正概率。将一个零支持行为称为“RL 可以随意探索出来”不符合这个受约束问题。实际 softmax 通常数学上全正，但截断采样、硬掩码和数值下溢都可能改变有效支持，应分别讨论。

A.2 序列 KL 的链式分解

令 $P(y)=\prod_t P(y_t\mid y_{<t})$、$Q$ 同理，真实终止后不再增加动作。对完整回答取期望，

$$D_{\mathrm{KL}}(P(y)\|Q(y))=\mathbb{E}_{y\sim P}\left[\sum_{t=1}^{T_y}\log\frac{P(y_t\mid y_{<t})}{Q(y_t\mid y_{<t})}\right] \tag{A.4}$$

$$=\sum_t \mathbb{E}_{s_t\sim d_{P,t}}\left[D_{\mathrm{KL}}(P(\cdot\mid s_t)\|Q(\cdot\mid s_t))\right]. \tag{A.5}$$

$d_{P,t}$ 是第 t 步仍存活前缀的未归一化访问质量，包含终止概率；也可在固定最大长度上引入吸收终止动作来写同一恒等式。这里是 token 和，不是先对每条随机长度取平均。$\mathbb{E}_P[T_y^{-1}\sum_t\log(P/Q)]$ 一般不等于序列 KL 除以 $\mathbb{E}_P[T_y]$，更不等于原序列 KL。若评测报告 token 平均 KL，应将其明确标成另一统计量。

A.3 KL 塑形为何可进入策略梯度

设 $r(y)$ 及 Q 固定。对 $J(\theta)=\sum_y P_\theta(y)[r(y)-\beta\log(P_\theta(y)/Q(y))]$ 求导，得到

$$\nabla J=\mathbb{E}_{P_\theta}\left[(r-\beta\log(P_\theta/Q))\nabla\log P_\theta\right]-\beta\mathbb{E}_{P_\theta}[\nabla\log P_\theta] \tag{A.6}$$

$$=\mathbb{E}_{P_\theta}\left[(r-\beta\log(P_\theta/Q))\nabla\log P_\theta\right]. \tag{A.7}$$

最后一步使用归一化恒等式 $\mathbb{E}_P[\nabla \log P] = 0$。因此在当前策略采样点，将该正则化回报 detach 后乘 score，可得到正确的期望梯度；但换成陈旧策略取样或裁剪比率后，是另一个代理估计问题。若奖励网络也与策略共享并更新参数，还需声明是否有意包含奖励的显式参数梯度。

A.4 包括自身的组均值与留一均值

固定提示，令独立同分布的 $y_i \sim \pi_\theta$，记序列 score $g_i = \nabla \log \pi_\theta(y_i \mid x)$、固定奖励 $R_i = r(y_i)$。因 $\mathbb{E}[g_i] = 0$，对 $j \neq i$ 有 $\mathbb{E}[R_j g_i] = \mathbb{E}[R_j]\mathbb{E}[g_i] = 0$。于是

$$\mathbb{E}[(R_i - \overline{R})g_i] = \mathbb{E}[R_i g_i] - \frac{1}{G}\sum_j \mathbb{E}[R_j g_i] \tag{A.8}$$

$$= \left(1 - \frac{1}{G}\right)\mathbb{E}[R_i g_i]. \tag{A.9}$$

若改为 $b_i = (G-1)^{-1}\sum_{j\neq i} R_j$，则 $\mathbb{E}[b_i g_i] = 0$，恢复原期望。这个结论依赖同提示独立采样与交叉项可分解；联合打分、共享搜索和依赖当前候选的 baseline 不自动满足条件。

若再除以随机组标准差，分母也与 y_i 及 g_i 相关，一般不能从期望中提出。GRPO 的组标准化可以是有用的代理设计，但其理由不能偷换成动作独立 baseline 的不变性。全同奖励组的所有中心化优势都为零，这一点与是否做留一修正无关。

A.5 采样 KL 值与自动微分的反例

对固定状态令 $P = P_\theta$、Q 冻结且在有限动作空间全正，$a \sim P$。令 $d(a) = \log Q(a) - \log P(a)$，则

$$\mathbb{E}_P[k_1] = \mathbb{E}_P[-d] = D_{\mathrm{KL}}(P\|Q), \tag{A.10}$$

$$\mathbb{E}_P[k_3] = \mathbb{E}_P[e^d - 1 - d] = \sum_a Q(a) - 1 + D_{\mathrm{KL}}(P\|Q) = D_{\mathrm{KL}}(P\|Q). \tag{A.11}$$

非负性来自 $e^d \geq 1 + d$。若从 $\mu \neq P$ 采样，没有相应权重时上述等式失效。若采样覆盖完整支持，可用 $P(a)/\mu(a)$ 恢复固定状态期望，但会改变方差，且不能独自恢复前缀状态分布。

接着保持已采样动作固定，只对表达式自动微分：

$$\nabla_\theta k_3(a) = \left(1 - \frac{Q(a)}{P(a)}\right)\nabla_\theta \log P(a). \tag{A.12}$$

其期望为 $-\mathbb{E}_Q[\nabla \log P]$，通常不同于 $\nabla D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_P[\log(P/Q)\nabla \log P]$。完整期望的导数还包含分布项：

$$\nabla_\theta \mathbb{E}_{a\sim P_\theta}[k_3(a)] = \mathbb{E}_P[\nabla_\theta k_3(a) + k_3(a)\nabla_\theta \log P(a)]. \tag{A.13}$$

对 k_1 同样如此：固定动作的显式导数期望为 0，但 KL 梯度通常不为 0。stop-gradient 的位置决定代码计算了哪个导数。

令 $P = (p, 1-p)$、$p = \sigma(z)$、$Q = (1/2, 1/2)$，则

$$\frac{\mathrm{d}}{\mathrm{d}z} D_{\mathrm{KL}}(P\|Q) = p(1-p)\log\frac{p}{1-p}, \tag{A.14}$$

$$\mathbb{E}_P\left[\frac{\partial k_3(a)}{\partial z}\right] = p - \frac{1}{2}. \tag{A.15}$$

在 $p = 0.8$ 处，两者分别为 $0.16\log 4 \approx 0.221807$ 和 0.3。这是一个明确反例：值的无偏性不能推出固定样本 autograd 梯度的无偏性。核对实现时，应以完整词表求和或有限差分作为独立参考，而非只比较两个同源函数的输出。

A.6 DPO 的重参数化与概率尺度

由式 (6.2)，同提示奖励差满足

$$\frac{r(x,y_a)-r(x,y_b)}{\tau_{\text{pref}}}=\frac{\beta}{\tau_{\text{pref}}}\left[\log\frac{\pi^*(y_a\mid x)}{\pi_{\text{ref}}(y_a\mid x)}-\log\frac{\pi^*(y_b\mid x)}{\pi_{\text{ref}}(y_b\mid x)}\right]. \tag{A.16}$$

代入 BT 的 Bernoulli 似然即可得到 DPO。对软标签 p，一对样本的 logit 最优值为 $\log[p/(1-p)]$，所以相对策略 log-ratio 目标为 $\tau_{\text{pref}}\log[p/(1-p)]/\beta$。奖励尺度、偏好温度与正则强度若未统一，两个所谓相同 β 的实验可能根本没有相同含义。

将序列 log 概率改成 token 平均，会替换这个 log-ratio 定义；换 reference、每轮刷新 reference 或把模板中的角色 token 纳入概率，也都会改变目标。理论核验与实验复现的共同起点，应是一份能精确重建这些对象的配置，而不是方法缩写。

附录 B

阅读索引、术语与跨册指引

B.1 带着问题阅读原始资料

主题	原始资料	先核对的问题
示范与 RLHF	InstructGPT[1]；Helpful/Harmless[30]	哪些数据生成 RM，哪些策略生成 rollout，独立评测是什么？
AI 反馈与安全	Constitutional AI[31]	反馈规则由谁定义，监督修订与强化学习阶段分别做什么？
偏好目标	DPO[7]；IPO[14]	BT 假设如何进入 DPO；IPO 为何改变有限数据下的拟合行为？
非成对与无参考	KTO、ORPO、SimPO[15, 16, 17]	标签结构、归一化、anchor 和 margin 分别改变了什么？
优势估计	PPO、GAE、RLOO[12, 11, 13]	哪个量冻结，baseline 条件独立于什么，终止如何处理？
可验证奖励	DeepSeekMath、R1、DAPO[18, 20, 19]	组归一化、长度权重、过滤、KL 及系统设置是否相同？
知识蒸馏	Hinton 等 [21]；GKD、MiniLLM[22, 23]	温度作用在哪个分布，前缀从哪里来，词表是否相同？
训练/部署前缀	Scheduled Sampling[32]	干预前缀如何改变目标；它为何不能直接等同于 OPD 或无偏 MLE？
推理提示与训练	Chain-of-Thought[33]	提示中给推理示范和更新参数是不同干预，比较是否混淆？
公开配方与实现	Tülu3、LlamaFactory[3, 34]	数据和评测是否齐全，框架 revision 的默认 mask/归一化是什么？
评测	MT-Bench、长度控制 AlpacaEval[9, 10]	人类一致性、位置/长度偏差在什么样本上测得？

引用键只是项目内部标识。例如含“2023”的键可以指向原始提交于 2023、后来实质修订的论文。应以参考文献中标注的初稿和所读版本为准。arXiv 的版本变化可能连题名与目标描述一起改变，本书在来源审计中保留实际核验响应，避免将旧记忆当成当前元数据。

B.2 术语速查

术语	定义、条件或边界
SFT	固定示范前缀上的条件似然训练；需要明确目标 mask 和归一化。
行为策略/采样快照	μ 是实际采样分布，π_k 是冻结模型；采样变换可使二者不同。
参考/教师	π_{ref} 提供漂移约束，q_T 提供监督分布；不必相同，也不等于行为策略。
RM、ORM、PRM	奖励模型、结果奖励模型、过程奖励模型；它们都是测量接口，不自动是真值。

术语	定义、条件或边界
Verifier/模型裁判	前者按可执行规范验证特定属性；后者按 rubric 估计质量，均可能不完整或有错。
RLHF/RLAIF	使用人类/AI 反馈的训练路线；常含奖励建模和策略优化，也需说明具体阶段。
DPO	参考相对序列 log-ratio 的偏好似然目标，其推导依赖 KL 正则和偏好模型。
RLVR	强化学习使用可验证奖励；可验证的属性不一定覆盖全部任务质量。
GRPO/RLOO	组相对策略优化/留一回报 baseline；零方差组不因换一种中心化便产生信号。
On-policy distillation	学生访问前缀上的教师分布监督；本书用固定快照局部目标说明实现。
在线偏好蒸馏	在线生成候选、获取偏好、更新学生的闭环；不是所有论文统一定义的 OPD 算法。
拒绝采样微调	对筛选后的生成文本做监督训练，选择器改变条件分布。
Policy lag	生成与学习之间的策略陈旧；版本差之外还需概率与队列诊断。
Reward hacking	代理奖励提高而真实目标未相应提高的行为偏离，不以主观意图为定义。

B.3 本书与其他分册

需要 MDP、价值函数和策略梯度的完整基础证明时，阅读《强化学习》；需要低资源适配器、量化微调与实际设备操作时，阅读《现代小模型微调》。工具运行时、状态与恢复设计属于《LLM 智能体系统》，可信边界、隐私和系统威胁模型属于《可信与安全的大模型》。这些主题各有重心，本书在需要处解释最小前置知识，并以目标、数据、优化和证据为自己的主线。

参考文献

[1] L. Ouyang et al. *Training language models to follow instructions with human feedback*. 所用版本 1，修订日期 2022-03-04. Mar. 4, 2022. DOI: 10.48550/arXiv.2203.02155. arXiv: 2203.02155v1 [cs.CL]. URL: https://arxiv.org/abs/2203.02155v1 (visited on 10/22/2026).

[2] S. Zhao et al. *Large Language Model Post-Training: A Unified View of Off-Policy and On-Policy Learning*. 所用版本 2，修订日期 2026-04-16. Apr. 9, 2026. DOI: 10.48550/arXiv.2604.07941. arXiv: 2604.07941v2 [cs.CL]. URL: https://arxiv.org/abs/2604.07941v2 (visited on 10/22/2026).

[3] N. Lambert et al. *Tulu 3: Pushing Frontiers in Open Language Model Post-Training*. 所用版本 5，修订日期 2025-04-14. Nov. 22, 2024. DOI: 10.48550/arXiv.2411.15124. arXiv: 2411.15124v5 [cs.CL]. URL: https://arxiv.org/abs/2411.15124v5 (visited on 10/22/2026).

[4] Hugging Face. *Chat templates: Transformers documentation*. Live documentation snapshot archived with this revision. URL: https://huggingface.co/docs/transformers/chat_templating (visited on 10/22/2026).

[5] E. J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685v2 [cs.CL]. URL: https://arxiv.org/abs/2106.09685v2 (visited on 10/22/2026).

[6] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. *QLoRA: Efficient Finetuning of Quantized LLMs*. 所用版本 1，修订日期 2023-05-23. May 23, 2023. DOI: 10.48550/arXiv.2305.14314. arXiv: 2305.14314v1 [cs.LG]. URL: https://arxiv.org/abs/2305.14314v1 (visited on 10/22/2026).

[7] R. Rafailov et al. *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. 所用版本 3，修订日期 2024-07-29. May 29, 2023. DOI: 10.48550/arXiv.2305.18290. arXiv: 2305.18290v3 [cs.LG]. URL: https://arxiv.org/abs/2305.18290v3 (visited on 10/22/2026).

[8] H. Lightman et al. *Let's Verify Step by Step*. 所用版本 1，修订日期 2023-05-31. May 31, 2023. DOI: 10.48550/arXiv.2305.20050. arXiv: 2305.20050v1 [cs.LG]. URL: https://arxiv.org/abs/2305.20050v1 (visited on 10/22/2026).

[9] L. Zheng et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 所用版本 4，修订日期 2023-12-24. June 9, 2023. DOI: 10.48550/arXiv.2306.05685. arXiv: 2306.05685v4 [cs.CL]. URL: https://arxiv.org/abs/2306.05685v4 (visited on 10/22/2026).

[10] Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto. *Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators*. 所用版本 2，修订日期 2025-03-10. Apr. 6, 2024. DOI: 10.48550/arXiv.2404.04475. arXiv: 2404.04475v2 [cs.LG]. URL: https://arxiv.org/abs/2404.04475v2 (visited on 10/22/2026).

[11] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. "High-Dimensional Continuous Control Using Generalized Advantage Estimation". In: *International Conference on Learning Representations*. arXiv v6, 2018-10-20; first preprint 2015-06-08. 2016. arXiv: 1506.02438v6. URL: https://arxiv.org/abs/1506.02438v6 (visited on 10/22/2026).

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. *Proximal Policy Optimization Algorithms*. 所用版本 2，修订日期 2017-08-28. July 20, 2017. DOI: 10.48550/arXiv.1707.06347. arXiv: 1707.06347v2 [cs.LG]. URL: https://arxiv.org/abs/1707.06347v2 (visited on 10/22/2026).

[13] A. Ahmadian et al. *Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs*. arXiv v2, 2024-02-26. 2024. arXiv: 2402.14740v2. URL: https://arxiv.org/abs/2402.14740v2 (visited on 10/22/2026).

[14] M. G. Azar et al. *A General Theoretical Paradigm to Understand Learning from Human Preferences*. 所用版本 2，修订日期 2023-11-22. Oct. 18, 2023. DOI: 10.48550/arXiv.2310.12036. arXiv: 2310.12036v2 [cs.AI]. URL: https://arxiv.org/abs/2310.12036v2 (visited on 10/22/2026).

[15] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela. *KTO: Model Alignment as Prospect Theoretic Optimization*. 所用版本 5，修订日期 2026-09-08. Feb. 2, 2024. DOI: 10.48550/arXiv.2402.01306. arXiv: 2402.01306v5 [cs.LG]. URL: https://arxiv.org/abs/2402.01306v5 (visited on 10/22/2026).

[16] J. Hong, N. Lee, and J. Thorne. *ORPO: Monolithic Preference Optimization without Reference Model*. 所用版本 2, 修订日期 2024-03-14. Mar. 12, 2024. DOI: 10.48550/arXiv.2403.07691. arXiv: 2403.07691v2 [cs.CL]. URL: https://arxiv.org/abs/2403.07691v2 (visited on 10/22/2026).

[17] Y. Meng, M. Xia, and D. Chen. *SimPO: Simple Preference Optimization with a Reference-Free Reward*. 所用版本 3, 修订日期 2024-11-01. May 23, 2024. DOI: 10.48550/arXiv.2405.14734. arXiv: 2405.14734v3 [cs.CL]. URL: https://arxiv.org/abs/2405.14734v3 (visited on 10/22/2026).

[18] Z. Shao et al. *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*. 所用版本 3, 修订日期 2024-04-27. Feb. 5, 2024. DOI: 10.48550/arXiv.2402.03300. arXiv: 2402.03300v3 [cs.CL]. URL: https://arxiv.org/abs/2402.03300v3 (visited on 10/22/2026).

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

[20] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 所用版本 2, 修订日期 2026-01-04. Jan. 22, 2025. DOI: 10.48550/arXiv.2501.12948. arXiv: 2501.12948v2 [cs.CL]. URL: https://arxiv.org/abs/2501.12948v2 (visited on 10/22/2026).

[21] G. Hinton, O. Vinyals, and J. Dean. *Distilling the Knowledge in a Neural Network*. 所用版本 1, 修订日期 2015-03-09. Mar. 9, 2015. DOI: 10.48550/arXiv.1503.02531. arXiv: 1503.02531v1 [stat.ML]. URL: https://arxiv.org/abs/1503.02531v1 (visited on 10/22/2026).

[22] R. Agarwal et al. *On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes*. 所用版本 3, 修订日期 2024-01-17；ICLR 2024；初稿 2023. June 23, 2023. DOI: 10.48550/arXiv.2306.13649. arXiv: 2306.13649v3 [cs.LG]. URL: https://arxiv.org/abs/2306.13649v3 (visited on 10/22/2026).

[23] Y. Gu, L. Dong, F. Wei, and M. Huang. *MiniLLM: On-Policy Distillation of Large Language Models*. 所用版本 6, 修订日期 2026-01-31；ICLR 2024；题名按所用 v6 记录. June 14, 2023. DOI: 10.48550/arXiv.2306.08543. arXiv: 2306.08543v6 [cs.CL]. URL: https://arxiv.org/abs/2306.08543v6 (visited on 10/22/2026).

[24] Y. Li et al. *Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe*. 所用版本 2, 修订日期 2026-04-15. Apr. 14, 2026. DOI: 10.48550/arXiv.2604.13016. arXiv: 2604.13016v2 [cs.LG]. URL: https://arxiv.org/abs/2604.13016v2 (visited on 10/22/2026).

[25] J. Zhou et al. *Instruction-Following Evaluation for Large Language Models*. 2023. arXiv: 2311.07911v1 [cs.CL]. URL: https://arxiv.org/abs/2311.07911v1 (visited on 10/22/2026).

[26] N. Lambert et al. *RewardBench: Evaluating Reward Models for Language Modeling*. 所用版本 2, 修订日期 2024-06-08. Mar. 20, 2024. DOI: 10.48550/arXiv.2403.13787. arXiv: 2403.13787v2 [cs.LG]. URL: https://arxiv.org/abs/2403.13787v2 (visited on 10/22/2026).

[27] S. Malik et al. *RewardBench 2: Advancing Reward Model Evaluation*. 所用版本 2, 修订日期 2026-04-23. June 2, 2025. DOI: 10.48550/arXiv.2506.01937. arXiv: 2506.01937v2 [cs.CL]. URL: https://arxiv.org/abs/2506.01937v2 (visited on 10/22/2026).

[28] C. Zheng et al. *ProcessBench: Identifying Process Errors in Mathematical Reasoning*. 所用版本 4, 修订日期 2025-05-26. Dec. 9, 2024. DOI: 10.48550/arXiv.2412.06559. arXiv: 2412.06559v4 [cs.AI]. URL: https://arxiv.org/abs/2412.06559v4 (visited on 10/22/2026).

[29] M. Chen et al. *Evaluating Large Language Models Trained on Code*. 2021. arXiv: 2107.03374v2 [cs.LG]. URL: https://arxiv.org/abs/2107.03374v2 (visited on 10/22/2026).

[30] Y. Bai et al. *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback*. 所用版本 1, 修订日期 2022-04-12. Apr. 12, 2022. DOI: 10.48550/arXiv.2204.05862. arXiv: 2204.05862v1 [cs.CL]. URL: https://arxiv.org/abs/2204.05862v1 (visited on 10/22/2026).

[31] Y. Bai et al. *Constitutional AI: Harmlessness from AI Feedback*. 所用版本 1, 修订日期 2022-12-15. Dec. 15, 2022. DOI: 10.48550/arXiv.2212.08073. arXiv: 2212.08073v1 [cs.CL]. URL: https://arxiv.org/abs/2212.08073v1 (visited on 10/22/2026).

[32] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer. *Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks*. 所用版本 3, 修订日期 2015-09-23. June 9, 2015. DOI: 10.48550/arXiv.1506.03099. arXiv: 1506.03099v3 [cs.LG]. URL: https://arxiv.org/abs/1506.03099v3 (visited on 10/22/2026).

[33] J. Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 所用版本 6，修订日期 2023-01-10. Jan. 28, 2022. DOI: 10.48550/arXiv.2201.11903. arXiv: 2201.11903v6 [cs.CL]. URL: https://arxiv.org/abs/2201.11903v6 (visited on 10/22/2026).

[34] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo. "LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Ed. by Y. Cao, Y. Feng, and D. Xiong. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 400–410. DOI: 10.18653/v1/2024.acl-demos.38. URL: https://aclanthology.org/2024.acl-demos.38/ (visited on 10/22/2026).