arXiv:2611.33102v1 [cs.LG] 24 Nov 2026

Carnegie Mellon University

强化学习可以从自身轨迹中恢复

Jiayi Sheng[*1] **Jingwen Gu**[*2] **Guanning Zeng**[1] **Zhaoyi Zhou**[1]
Stefano Viel[3] **Qianqian Wang**[4] **Fahim Tajwar**[1] **Andrea Zanette**[1]1卡内基梅隆大学 [2]伊利诺伊大学厄巴纳-香槟分校[3]TufaLabs [4]哈佛大学
{jsheng2,azanette}@andrew.cmu.edu, jgu17@illinois.edu

摘要

广义而言，强化学习（RL）后训练通过使用模型生成的数据进行训练，将计算资源转化为模型能力。通常，每次更新后都会丢弃轨迹，而强化学习期间获得的知识则保存在模型权重中。在这项工作中，我们探究这些知识有多少保留在轨迹本身中。一个自然的初步尝试是，在存档轨迹而非新生成的轨迹上重放原始强化学习算法，但这会导致灾难性崩溃。我们对这一现象给出了一个简单解释，并据此提出补救方法：对存档轨迹进行最大似然训练。令人惊讶的是，从同一个基础模型出发，按照轨迹生成的顺序重放这些轨迹，且不区分正确响应和错误响应，就足以恢复在线强化学习的性能。事实上，每个轨迹批次仅进行一次最大似然更新，就能紧密跟踪在线强化学习的整条学习曲线。此外，仅使用最近几个强化学习步骤中的轨迹进行最大似然训练，通常就足以恢复最终模型的性能。这表明，强化学习投入的大部分计算资源，可能都用于发现更优的采样分布，而非从样本中学习该分布。

1 引言

强化学习（RL）后训练通过使用模型生成的数据进行训练，将计算资源转化为模型能力（Lambert 等人，2025年；Guo 等人，2025年）。在整个训练过程中，模型会反复从当前分布中生成样本，并利用奖励反馈改进这一分布。由于语言模型具有自回归特性，生成这些样本的成本尤其高昂（Noukhovitch 等人，2025年）。然而，在标准的同策略后训练中，轨迹通常会在相应轮次的更新完成后被丢弃（Schulman 等人，2017年；Shao 等人，2024年；Ahmadian 等人，2024年；Tajwar 等人，2026年）。训练结束时，留下的只有模型权重，其中编码着强化学习过程中获得的知识。

在本文中，我们研究这种知识的另一种表征：由……生成的数据
训练过程中的模型。随着模型不断改进，其训练数据的采样分布也随之变化。这自然引出了一个问题：

模型在强化学习过程中学到的内容，有多少可以从它在此过程中生成的数据中恢复？

一个自然的初步尝试是重放存档轨迹，将它们视为离线强化学习数据集（Levine 等人，2020年）。原则上，从相同的基础模型出发，在相同数据上应用相同的更新，应当能够重现原始训练轨迹。然而，在我们的实验中，这一过程极不稳定，会导致灾难性崩溃，即使采用了我们评估的异策略校正（Degris 等人，2012年；Marek 和 Ryabinin，2026年）也是如此。我们发现了一种可能导致重放不稳定的机制：一旦重放模型偏离生成轨迹的模型，策略梯度基线就可能在连续更新中放大这种偏差。

*贡献相同。

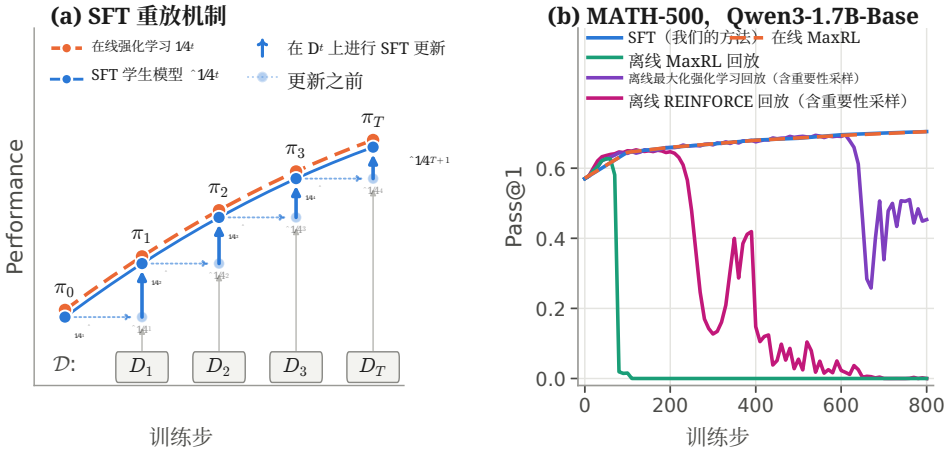


图 1：从轨迹中恢复在线强化学习。(a) 使用监督微调恢复在线强化学习。在线运行在每个步骤采样一个批次 D_t 。学生模型 $\hat{\pi}_t$ 按顺序对每个 D_t 进行一次监督微调更新，从而使其与在线检查点 π_t保持接近。(b) 在在线 MaxRL 运行的存档轨迹上训练的 Qwen3-1.7B-Base 在 MATH-500 上的 pass@1。标记为含重要性采样的重放会根据截断的重要性比率对每个词元重新加权。三种离线重放每 10 步评估一次，其他运行每 100 步评估一次。

这一观察提示了另一种方法。我们不必试图重现原始强化学习更新，而可以通过最大似然直接学习生成轨迹的分布，并采用监督微调（SFT）（Rusu 等人，2016年；Kim 和 Rush，2016年）。在总体层面上，这一目标函数会推动重放模型趋近于生成轨迹的模型，从而纠正两者之间的偏差，而非将其放大。

尽管这一观察在总体层面上直观易懂，但这种方法在经验上的有效性却令人惊讶。从同一个基础模型出发，按照轨迹生成的顺序，对每个批次进行一次最大似然更新，我们便能以惊人的精确度恢复在线强化学习的整条学习曲线。这一过程无需获取奖励，甚至无需区分正确响应和错误响应。我们观察到，在多种设置下都能高度还原轨迹，并且在不同模型规模、任务和强化学习算法中都能恢复最终性能。因此，无需重现模型在强化学习训练中获得这些能力所经历的优化过程，也能恢复其性能随训练演变的过程。

或许更令人意外的是，恢复最终模型的性能并不需要重放整个训练历史。在多种设置下，仅对单个强化学习后期步骤中的轨迹反复进行最大似然更新，就能恢复经过一千次在线强化学习更新后达到的性能。这表明，强化学习中的大量计算工作都用于发现改进后的采样分布，而不是从样本中学习该分布。

除了恢复性能之外，这些档案还支持进一步复用：可以将分别训练的专用模型生成的轨迹结合起来训练一个模型，也可以用一个模型生成的轨迹训练另一个模型。总体而言，这些发现表明，存档的强化学习轨迹能够保留模型性能的演进过程和最终水平，使其成为训练过程中的可复用资产。

2 预备知识

在线强化学习。 语言模型在提示词 x下对响应 y 定义了一个策略 $\pi_\theta(y \mid x)$，而验证器会为每个响应分配一个二元奖励 $r(x,\ y) \in \{0, 1\}$。给定提示分布 ρ，在线强化学习最大化期望奖励

$$J(\theta) = \mathbb{E}_{x\sim\rho}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[r(x, y)\big].$$

在总体层面，广泛使用的方法通过策略梯度更新模型

$$g_\theta = \mathbb{E}_{x\sim\rho}\Big[w(x)\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[\nabla_\theta \log \pi_\theta(y \mid x)\,\big(r(x, y) - b(x)\big)\big]\Big], \tag{1}$$

其中 基线 $b(x)$ 在不改变期望梯度的情况下降低方差，而权重 $w(x)$ 表示..... ts
多少 每个提示词都有贡献。在实践中，在每个步骤，模型会采样 G 个响应 $y_{i1}, \ldots, y$ iG

对每个 B 提示 x_i。将奖励写作 $r_{ij} = r(x_i, y_{ij})$，它通过以下方式估计方程（1）

$$\hat{g}_\theta = \frac{1}{BG} \sum_{i=1}^{B} w_i \sum_{j=1}^{G} \nabla_\theta \log \pi_\theta(y_{ij} \mid x_i) \big(r_{ij} - b_{ij}\big), \tag{2}$$

其中 w_i 和 b_{ij} 估计 $w(x_i)$以及 $b(x_i)$来自 G 的奖励 x_i。表 1列出了它们对广泛使用方法的选择。

表 1： 总体层面（方程 (1) 中的 $w(x)$ 和 $b(x)$）以及实际应用中（方程 (2) 中的 w_i 和 b_{ij}）对权重和基线的选择。 $\bar{r}_i$ 和 σ_i 分别是奖励 r_{i1}、……、 r_{iG}的均值和标准差，而 $\bar{r}_{i,\,-j}$ 不包括 r_{ij}。 $p = p(x)$ 是当前策略在 x上的成功率， $\bar{r}_i$ 对其进行估计。

	种群		实际应用中	
方法	$w(x)$	$b(x)$	w_i	b_{ij}
REINFORCE（Williams，1992）	1	p	1	$\bar{r}_i$
RLOO（Ahmadian 等人，2024年）	1	p	1	$\bar{r}_{i,\,-j}$
GRPO（Shao 等人，2024年）	$1/\sqrt{p\,(1-p)}$	p	$1/\sigma_i$	$\bar{r}_i$
MaxRL（Tajwar 等人，2026年）	$1/p$	p	$1/\bar{r}_i$	$\bar{r}_i$

保存轨迹。每次运行在每个步骤采样 T 步 t一个批次

$$D_t = \big\{(x_i, y_{i1}, \ldots, y_{iG})\big\}_{i=1}^{B}$$

从其当前策略 π_t，然后更新 π_t 至 π_{t+1} ，使用方程（2中的梯度估计量。应用更新后， D_t 通常会被丢弃。我们则按生成顺序保存所有批次（图2，上方），从而得到轨迹数据集

$$\mathcal{D} = (D_0, D_1, \ldots, D_{T-1}). \tag{3}$$

SFT 仅使用 $\mathcal{D}$中的提示词和响应。第3.1节介绍的离线强化学习回放还需要它们的奖励。

3 从轨迹中恢复在线强化学习

本节介绍如何仅凭保存的轨迹恢复一次在线强化学习运行，并从总体层面简要说明各方法在何种情况下有效。我们首先在第3.1 节讨论离线强化学习回放，并解释其为何可能失效，无论是否使用重要性采样。接着，第3.2 节介绍对轨迹进行 SFT，并说明在什么情况下只需最近几个批次；第3.3 节说明轨迹的顺序为何也很重要；第3.4节则将其应用于专精模型的组合和跨模型迁移。我们将在第4节检验这些预测。

3.1 离线强化学习重放

要从轨迹中恢复一次在线强化学习运行，最直接的方法就是离线重放其训练过程（Levine 等人，2020年）。从相同的基础模型出发，使用相同的数据、奖励和超参数，按照与在线运行相同的方式更新模型。形式上，一个新模型 $\hat{}\pi_0$ 从 π_0开始，并按照原始顺序对每个保存的批次 D_t 进行一次更新。我们先从总体层面分析这种重放，然后给出实践中使用的更新方法。

总体分析。 在步骤 t，响应来自 π_t，而正在更新的模型是 $\hat{}\pi_t$。因此，重放计算的是方程（1）的策略梯度，其中响应来自 π_t ，而非 $\hat{}\pi_t$，

$$g_t = \mathbb{E}_{x\sim\rho}\Big[w(x)\,\mathbb{E}_{y\sim\pi_t(\cdot|x)}\big[\nabla \log \hat{\pi}_t(y \mid x)\,\big(r(x,y) - b(x)\big)\big]\Big]. \tag{4}$$

对于每个提示词 x， g_t的基线项为 $-w(x)\,b(x)\;\mathbb{E}_{y\sim\pi_t}[\nabla \log \hat{}\pi_t(y \mid x)]$。当响应来自正在更新的模型时，即 $\hat{}\pi_t = \pi_t$， $\mathbb{E}_{y\sim\pi_t}[\nabla \log\ \pi_t(y \mid x)] = 0$，因此基线项会消失。然而，离线重放期间的微小数值误差（Qi 等人，2025年）可能导致 $\hat{}\pi_t$ 偏离 π_t。此时基线项不再消失，并会使更新产生偏差（Degris 等人，2012年）。

要了解这种有偏更新会产生什么影响，考虑在固定提示词 x 上使用 MaxRL。设 $p_t(x)$ 为 π_t 在 x上的成功率，设 π_t^+为在正确响应条件下的 π_t （假设 $p_t(x) > 0$），

$$p_t(x) = \mathbb{E}_{y\sim\pi_t(\cdot|x)}\big[r(x,y)\big], \qquad \pi_t^+(y \mid x) = \frac{\pi_t(y \mid x)\, r(x,y)}{p_t(x)}. \tag{5}$$

MaxRL 将策略向 π_t^+移动。在总体层面，它使用权重 $w(x) = 1/p_t(x)$ 以及基线 $b(x) = p_t(x)$ （表 1）。我们将方程（4）的内层期望记为 $g_t(x)$:

$$\begin{aligned} g_t(x) &= w(x)\, \mathbb{E}_{y\sim\pi_t}\big[\nabla \log \hat{\pi}_t(y \mid x)\, \big(r(x,y) - b(x)\big)\big] \\ &= \mathbb{E}_{y\sim\pi_t}\Big[\nabla \log \hat{\pi}_t(y \mid x)\Big(\frac{r(x,y)}{p_t(x)} - 1\Big)\Big] \\ &= \sum_y \frac{\pi_t(y \mid x)\, r(x,y)}{p_t(x)}\, \nabla \log \hat{\pi}_t(y \mid x) - \mathbb{E}_{y\sim\pi_t}\big[\nabla \log \hat{\pi}_t(y \mid x)\big]. \end{aligned}$$

根据方程（5），第一项中的分数为 $\pi_t^+(y \mid x)$，因此第一项是在 π_t^+下的期望。这给出

$$g_t(x) = \underbrace{\mathbb{E}_{y\sim\pi_t^+}\big[\nabla \log \hat{\pi}_t(y \mid x)\big]}_{\text{learning successful behavior}} - \underbrace{\mathbb{E}_{y\sim\pi_t}\big[\nabla \log \hat{\pi}_t(y \mid x)\big]}_{\text{baseline term}}. \tag{6}$$

在同策略情况下，当 $\hat{}\pi_t = \pi_t$时，基线项的期望为零。一旦 $\hat{}\pi_t$ 从 π_t发生漂移，它就不再为零。将 π_t 和 π_t^+ 视为固定分布，方程（6）就是对以下目标进行上升的梯度

$$-\mathrm{KL}(\pi_t^+ \,\|\, \hat{\pi}_t) + \mathrm{KL}(\pi_t \,\|\, \hat{\pi}_t). \tag{7}$$

第一项将学生模型吸引到成功行为上。第二项来自基线，将 $\hat{}\pi_t$ 推离 π_t，即生成轨迹的策略。这使得下一次更新的偏差进一步增大，直至发生崩溃。这直观地说明了为什么朴素的离线重放会变得不稳定：微小的偏离会不断累积。我们的实验确认，离线强化学习回放确实会发生崩溃（第 4.1节）。

E经验更新。实际上，重放会在保存的批次 D_t 上将 $\hat{}\pi_t$ 更新为 $\hat{}\pi_{t+1}$，并使用

$$\hat{g}_t = \frac{1}{BG}\sum_{i=1}^{B} w_i \sum_{j=1}^{G} \nabla \log \hat{\pi}_t(y_{ij} \mid x_i)\, \big(r_{ij} - b_{ij}\big), \tag{8}$$

其中$(x_i, y_{i1}, \ldots, y_{iG}) \in D_t$，而 w_i和 b_{ij}则是在线算法中的对应项（表1）。这与在线运行在方程（2）中使用的更新相同。唯一的区别在于，$\hat{}\pi_t$ 可能会偏离 π_t，也就是生成这些响应的策略。

针对这种漂移的一种标准修正方法是重要性采样（IS）（Degris 等人，2012年）。它根据每个词元在 $\hat{}\pi_t$ 下相较于在 π_t 下的可能性，对其重新加权。我们会在每个批次中存储 π_t 的词元概率。令 $y_{ij,l}$ 表示 y_{ij} 的第 l 个词元，并令 $\pi(y_{ij,l})$ 表示 $\pi(y_{ij,l} \mid x_i, y_{ij,<l})$，我们使用截断的词元级重要性比率

$$\rho_{ij,l} = \min\big(\hat{\pi}_t(y_{ij,l})/\pi_t(y_{ij,l}),\, C\big), \tag{9}$$

其中 $C = 2$。方程（8）中的更新随后变为

$$\hat{g}_t^{\mathrm{IS}} = \frac{1}{BG}\sum_{i,j} w_i \sum_{l=1}^{|y_{ij}|} \rho_{ij,l}\, \nabla \log \hat{\pi}_t(y_{ij,l})\, \big(r_{ij} - b_{ij}\big). \tag{10}$$

当 $\hat{}\pi_t = \pi_t$ 时，每个重要性比率都等于 1，方程 (10) 就简化为方程 (8)。然而，我们的实验表明，重要性采样只能延缓崩溃，无法阻止崩溃（第 4.1 节）。

3.2 通过监督微调恢复在线强化学习

接下来，我们不再重放在线更新，而是使用监督微调来恢复强化学习性能（图 2（a））。在步骤 t，通过最小化 D_t 中响应的负对数似然，将 $\hat{}\pi_t$ 更新为 $\hat{}\pi_{t+1}$。

$$\mathcal{L}_{\mathrm{SFT}}(\hat{\pi}_t; D_t) = -\frac{1}{N_t}\sum_{i,j}\sum_{l=1}^{|y_{ij}|} \log \hat{\pi}_t(y_{ij,l} \mid x_i, y_{ij,<l}), \tag{11}$$

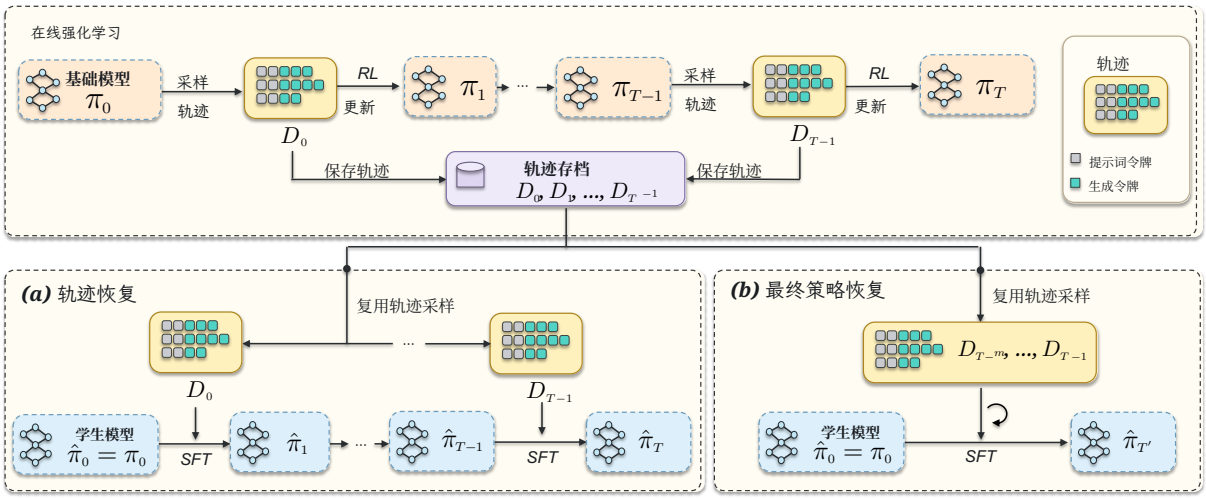


图 2：从保存的轨迹中恢复在线强化学习。 在线运行会保存每个步骤的轨迹 $D_0, \ldots, D_{T-1}$。(a) 对于轨迹恢复，学生模型按顺序对每个批次进行一次 SFT 更新。(b) 对于最终策略恢复，学生模型仅使用最后 m 步的轨迹进行 SFT 训练。

其中 $(x_i, y_{i1}, \ldots\ldots, y_{iG}) \in D_t$，$y_{ij,\,l}$ 是 y_{ij} 中的第 l 个词元，而 $N_t = \sum_{i,\,j} |y_{ij}|$ 是 D_t 中响应词元的数量。与方程（8）不同，该损失不使用奖励。最小化方程（11）对应于总体更新

$$\hat{\pi}_{t+1} = \arg\min_{q} \mathbb{E}_{x\sim\rho}\big[\mathrm{KL}\big(\pi_t(\cdot \mid x) \,\|\, q(\cdot \mid x)\big)\big]. \tag{12}$$

因此，每次 SFT 更新都会使 $\hat{\pi}_t$ 更接近 π_t。如果模型类能够表示 π_t，且优化达到全局最小值，方程 (12) 就会精确得到 $\hat{\pi}_{t+1} = \pi_t$。在训练过程中，学生模型因此会逐步跟随在线检查点 $\pi_0, \pi_1, \ldots, \pi_{T-1}$，一次前进一步（图 1(a)）。最终学生模型为 $\hat{\pi}_T = \pi_{T-1}$；如果最后一次强化学习更新幅度较小，它就会接近最终检查点 π_T。如果 $\hat{\pi}_t$ 偏离π_t，下一次更新仍会将它拉回在线运行的方向。在离线强化学习回放中，同样的漂移反而会使下一次更新产生更大的偏差。

如果我们只关心最终性能，就不必重放完整路径。根据上述分析，方程（12）的最小化解是 π_t。因此，原则上，仅在最后一个批次 D_{T-1}上对基础模型进行监督微调，就能恢复 π_{T-1}。然而在实践中，$\mathrm{KL}(\pi_{T-1} \,\|\, \pi_0)$ 很大。一个轨迹批次的数据太少，无法弥合这一差距，因此学生模型可能会对其过拟合。因此，我们使用更多数据，即最后m步的轨迹，D_{T-m}、……、D_{T-1}（图 2（b））。在总体层面，对这些数据进行监督微调可以求解

$$\min_{q} \frac{1}{m} \sum_{t=T-m}^{T-1} \mathbb{E}_{x\sim\rho}\big[\mathrm{KL}(\pi_t \,\|\, q)\big], \tag{13}$$

其最小化解是这些检查点的平均值。如果最后的 m 个检查点与 π_T 足够接近，那么该平均值也会接近 π_T，对它们的轨迹进行监督微调即可恢复最终策略。

3.3 轨迹的顺序很重要

根据第 3.2 节中的分析，SFT数据的顺序也很重要。我们比较已保存批次的三种顺序。按时间顺序回放采用在线运行中的顺序，D_0，D_1，……，D_{T-1}。反向回放采用 D_{T-1}，D_{T-2}，……，D_0。打乱顺序回放会随机排列 σ，并按 $\{0, \ldots, T-1\}$ 的顺序使用 $D_{\sigma(0)}$，$D_{\sigma(1)}$，……，$D_{\sigma(T-1)}$。在每种情况下，学生模型都从基础模型开始，并依次在每个批次上进行一次 SFT 更新。方程（12）中的每次精确更新都会使学生模型转向当前批次对应的策略，因此最终模型会接近最后呈现的策略。于是，这三种顺序最终得到的学生模型截然不同，

$$\underbrace{\hat{\pi}_T \approx \pi_{T-1}}_{\text{chronological}}, \quad \underbrace{\hat{\pi}_T \approx \pi_0}_{\text{reverse}}, \quad \underbrace{\hat{\pi}_T \approx \pi_{\sigma(T-1)}}_{\text{shuffled}}. \tag{14}$$

只有按时间顺序回放才能接近最终策略。我们在第 4.3 节验证这一分析。

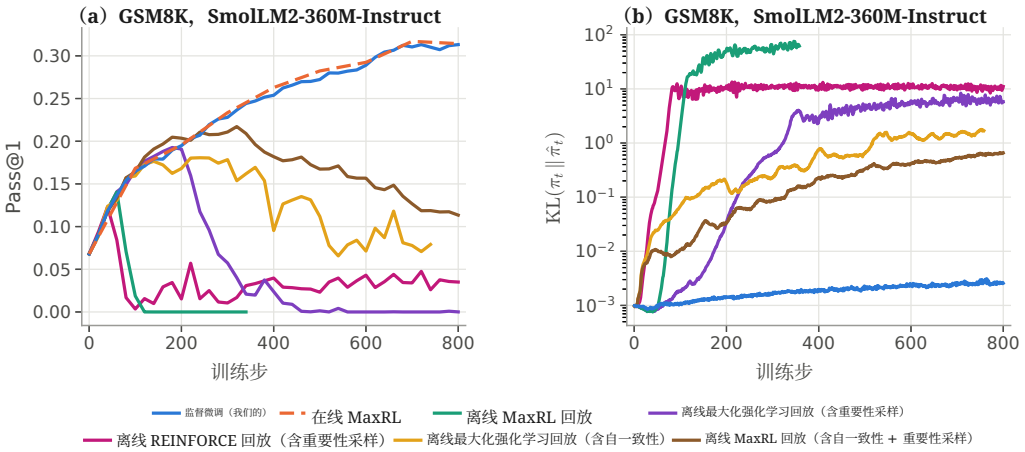


图3：离线强化学习回放会偏离生成这些轨迹的策略。（a）在在线 MaxRL 运行的轨迹上训练的 SmolLM2-360M-Instruct 的 GSM8K pass@1。（b）$\mathrm{KL}(\pi_t \| \hat{\pi}_t)$ 在 π_t 生成每个批次的策略与训练后模型$\hat{\pi}_t$之间，对于相同的运行结果。含 IS 会加入方程（10）中的重要性采样，含 SC 会进行分数中心化。

3.4 专用模型的组合与跨模型迁移

现在，我们将上述分析应用于保存的 rollout 的两种用途：将多个强化学习专家模型合并为一个模型，以及训练一个不同的模型。对于合并专用模型，设 π^1，……，π^M 是 M 个专用模型，它们以同一个基础模型为起点，分别在提示分布 ρ_1，……，ρ_M 上训练，且每个模型都有一个档案 $\mathcal{D}^k = (D_0^k, \ldots\ldots, D_{T-1}^k)$。学生模型按时间顺序重放这些档案，并在步骤 t 将 D_t^1，……，D_t^M 一起用于训练，其中任务 k 所占比例为 w_k。精确的监督微调随后可以解出

$$\hat{\pi}_{t+1} = \arg\min_q \sum_{k=1}^{M} w_k \, \mathbb{E}_{x\sim\rho_k}\big[\mathrm{KL}(\pi_t^k \,\|\, q)\big]. \tag{15}$$

当各任务没有共享提示词时，最小化解在任务 k 的每个提示词上都等于 π_t^k。因此，学生模型能够同时沿着每个专用模型的轨迹前进，在组合各任务的同时重现每个任务的训练曲线。权重合并则无法提供这样的保证，因为它合并的是参数，而非这些参数所产生的行为。

对于跨模型迁移，另一个学生模型从源运行的轨迹中学习，该源运行包含以下检查点：π_0、……、π_T。设 $\mathcal{Q}$ 为学生模型能够表示的策略集合。与方程（13）一样，学生模型使用最后 m 步的轨迹进行训练，D_{T-m}、……、D_{T-1}。由于 $\mathcal{Q}$ 可能不包含源策略，精确的监督微调求解

$$\min_{q\in\mathcal{Q}} \frac{1}{m} \sum_{t=T-m}^{T-1} \mathbb{E}_{x\sim\rho}\big[\mathrm{KL}(\pi_t \,\|\, q)\big], \tag{16}$$

其最小化解是投影到 $\mathcal{Q}$ 上的平均值 $\frac{1}{m}\sum_{t=T-m}^{T-1}\pi_t$。蒸馏求解相同的目标函数，但将 π_T 替换为每个 π_t，最终得到 π_T 在 $\mathcal{Q}$上的投影。当最后 m个检查点接近 π_T时，两个投影也会彼此接近。然而，蒸馏需要从最终策略采样新的响应。生成这些响应需要额外的时间和计算资源。我们的学生模型则重用已保存的 rollout，除了原始运行之外不产生额外成本，且表现与蒸馏相近（第4.5节）。

4 实验

现在转向对第 3 节分析的实证评估。我们首先展示，离线强化学习回放无法从轨迹中恢复在线强化学习运行（第 4.1 节），而监督微调则可以成功恢复（第 4.2 节）。第 4.3 节研究轨迹的顺序是否重要。最后，我们重用这些轨迹来组合专用模型（第 4.4 节），并训练另一个模型（第 4.5 节）。除非另有说明，否则每次在线运行都使用 MaxRL 训练 $T = 1000$ 步骤，每个批次包含 256 个提示、每个提示对应 16 条轨迹，并保存所有 rollout 批次。训练和评估详情见附录 A。

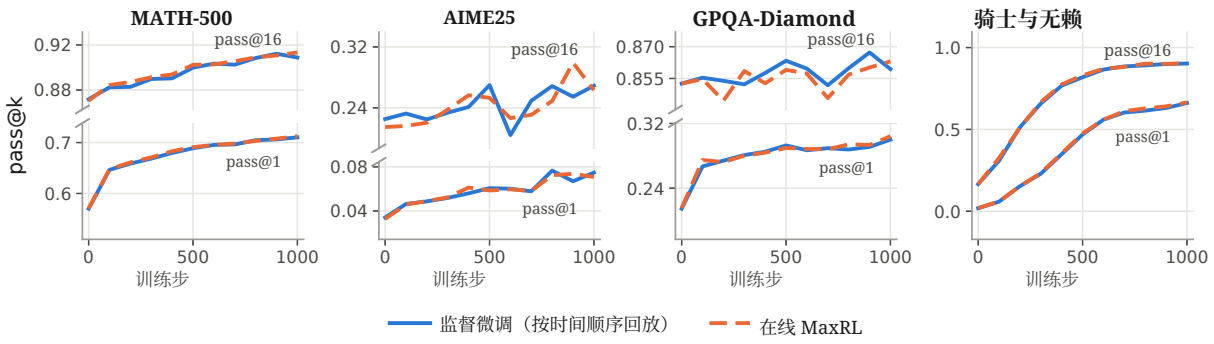


图4：对存档轨迹进行按时间顺序的监督微调，其学习曲线与在线 MaxRL 相符。 在 Qwen3-1.7B-Base 训练过程中，针对 MaxRL 运行的轨迹按原始顺序进行监督微调（蓝色实线）和在线 MaxRL（橙色虚线）的 pass@1 和 pass@16。两项指标差距较大处，y-轴采用了断轴处理。

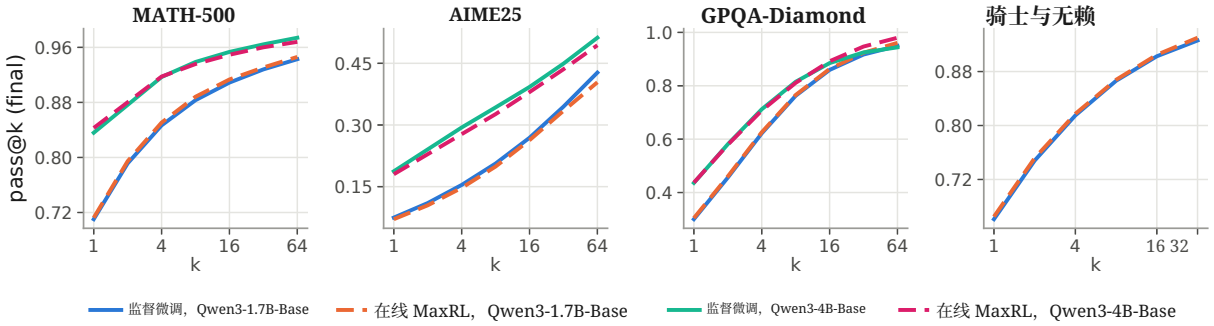


图 5：对存档轨迹进行按时间顺序的监督微调，可恢复在线 MaxRL 的最终检查点。针对 MaxRL 运行的轨迹按原始顺序进行监督微调（实线）和在线 MaxRL（虚线）的最终检查点 pass@k，其中 Qwen3-1.7B-Base（蓝色/橙色）和 Qwen3-4B-Base（绿色/红色）。

4.1 离线强化学习重放崩溃

我们首先使用方程（8），在 Polaris-53K（An 等，2025）上重放 Qwen3-1.7B-Base（Yang 等，2025）的在线 MaxRL 运行，分别测试不使用和使用重要性采样（IS，方程（10））的情况。我们还使用 IS 重放相同数据，但采用 REINFORCE 的权重和基线（表 1）。每次重放都从同一个基础模型开始，并每 10 个步骤在 MATH-500（Hendrycks 等，2021；Lightman 等，2024）上评估其 pass@1。我们在 GSM8K（Cobbe 等，2021）上训练的 SmolLM2-360M-Instruct（Ben Allal 等，2025）上重复这些重放实验，并额外与一种针对这一不匹配问题的近期修复方法——分数中心化（SC）（Marek 和 Ryabinin，2026）——进行比较。SC 从每个采样词元的分数中减去其在生成这些轨迹的策略下的期望值，并测试不使用和使用 IS 的情况。为了了解重放为何会崩溃，我们还跟踪 $\mathrm{KL}(\pi_t \,\|\, \hat{}\pi_t)$，即生成每个批次的策略与训练模型之间按词元计算的 KL 散度。所有重放都共享在线运行的代码，只将新的轨迹替换为已保存的轨迹。

图 1（b）显示，正如第 3.1 节所预测的那样，离线 MaxRL 重放在训练早期就会崩溃。重要性采样会延缓崩溃，但无法阻止它。采用重要性采样时，离线 MaxRL 重放在训练的大部分时间里都与在线运行保持一致，之后才崩溃；而离线 REINFORCE 重放则会早得多地崩溃。

图 3（a）显示，SmolLM2-360M-Instruct 上也呈现相同模式：所有离线重放都会崩溃，而分数中心化和重要性采样只能延缓崩溃。为了解其中原因，面板 (b) 跟踪了 KL 散度；对于每种离线重放，KL 散度都会增长数个数量级，直至重放崩溃。这就是第 3.1 节所说的漂移。重要性采样和分数中心化是为 π_t 与 $\hat{}\pi_t$ 之间的小幅偏差而设计的（Degris 等人，2012年；Marek 和 Ryabinin，2026年）。它们可以降低每次更新的偏差，但当训练后的模型偏离生成数据的策略太远时，仍可能失效（Bai 等人，2026年）。

要点一

用离线强化学习回放来恢复在线强化学习会导致崩溃，而重要性采样只能延缓崩溃。

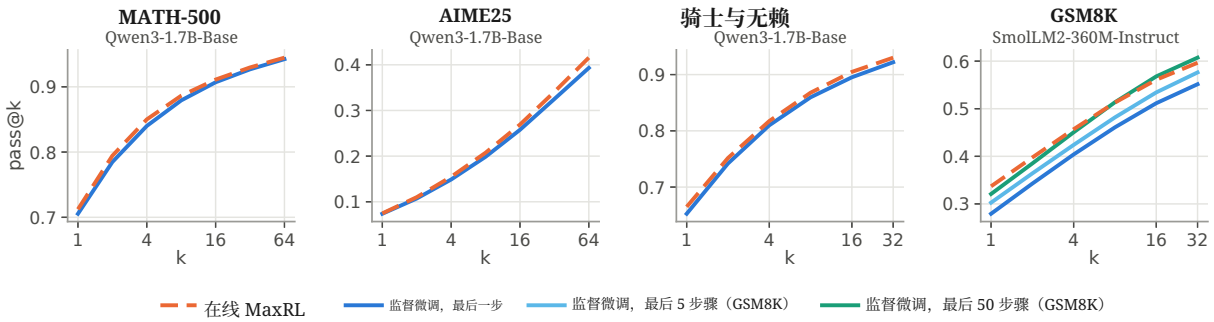


图6：从最后一个强化学习步骤的轨迹中进行端点恢复。在线 MaxRL（橙色虚线）的最终检查点 pass@k，以及基础模型在最后一个强化学习步骤的轨迹上进行监督微调后的 pass@k（蓝色）。对于 SmolLM2-360M-Instruct，监督微调还会使用最后 5 步（浅蓝色）和 50 步（绿色）的数据进行训练。

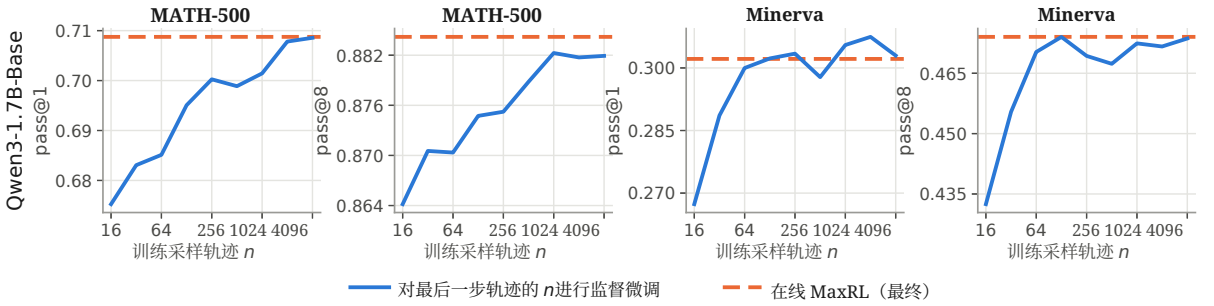


图 7：需要使用最后一步的多少轨迹。Polaris 上的 Qwen3-1.7B-Base：基础模型在最后一个强化学习步骤的 4096 条轨迹中随机抽取 n，进行监督微调并完成 100 次监督微调更新后，与在线 MaxRL 的最终检查点（橙色虚线）进行比较。

4.2 监督微调恢复在线强化学习

现在，我们使用保存的 rollout，通过监督微调来训练学生模型（第 3.2 节）。除了第 4.1 节中的 Qwen3-1.7B-Base 和 SmolLM2-360M-Instruct 运行外，我们还使用了 Qwen3-4B-Base 在 Polaris-53K 上以及 Qwen3-1.7B-Base 在骑士与无赖（KnK）上的在线运行（Xie 等人，2024年）。我们在 MATH-500、AIME25（美国数学协会，2025年）、GPQA-Diamond（Rein 等人，2024年）以及 KnK 谜题上进行评估。

我们定义两种恢复：轨迹恢复和端点恢复。轨迹恢复要求在整个训练过程中，每个 t 都满足 $\hat{}\pi_t$ 与 π_t 相匹配；而端点恢复只要求最终学生模型 $\hat{}\pi_T$ 与最终检查点 π_T 相匹配。用 $F_k(\pi)$ 表示策略在基准测试上的 pass@k，我们在固定集合 $k \in \mathcal{K} = \{1, 2, 4, \ldots, 32\}$ 上对 pass@k 的差值取平均。轨迹误差还会在经过评估的步骤集合 $\mathcal{T}$ 上取平均，

$$\epsilon_{\text{traj}} = \frac{1}{|\mathcal{T}|\,|\mathcal{K}|} \sum_{t \in \mathcal{T}} \sum_{k \in \mathcal{K}} \left| F_k(\hat{\pi}_t) - F_k(\pi_t) \right|, \tag{17}$$

端点误差仅比较最终学生模型与最终检查点，

$$\epsilon_{\text{end}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} \left| F_k(\hat{\pi}_T) - F_k(\pi_T) \right|. \tag{18}$$

作为基线，我们还计算了未经训练的基础模型的两种误差；该模型在每个步骤都用 π_0 替代 $\hat{}\pi_t$。其误差衡量在线运行偏离基础模型的程度。

轨迹恢复。学生模型按原始顺序对每个保存的批次执行一次监督微调更新。在第4.1节的运行中，监督微调紧随在线运行进行，而每次离线重放都会崩溃；在整个训练过程中，学生模型与生成各批次的策略之间的 KL 散度始终很小（图 1(b) 和3）。这是预期结果，因为监督微调直接最小化这一 KL 散度（方程（12））。且每次更新都会将学生模型拉回到在线运行所对应的状态。随后，我们每隔 100 步将学生模型与在线运行进行比较。图 4 比较了 Qwen3-1.7B-Base 在训练过程中 SFT 学生模型与在线 MaxRL 的 pass@1 和 pass@16。在全部四项基准测试中，两项指标的曲线都同步上升，几乎无法区分。

表 2： 监督微调相对于在线 MaxRL 的恢复误差。**(a)** 在图 4 所示基准测试上，对所有轨迹进行按时间顺序的监督微调。ϵ_{traj}（方程（17））针对步数 0、100、.....、1000，而 ϵ_{end}（方程（18））针对步数 1000。**(b)** 监督微调在所有轨迹上以及最近 ϵ_{end} 步数的轨迹上的端点误差。在这两个表中，基础模型在每一步都用未经训练的基础模型 π_0 替代学生模型。对于 Qwen3-4B-Base，K 省略 $k = 2$。

(a) 轨迹恢复与端点恢复

	Qwen3-1.7B-Base		Qwen3-4B-Base
基准测试	ϵ_{traj}	ϵ_{end}	ϵ_{end}
对所有轨迹进行监督微调			
MATH-500	0.002	0.004	0.004
AIME25	0.011	0.007	0.013
GPQA-Diamond	0.004	0.004	0.008
KnK	0.006	0.003	–
基础模型			
MATH-500	0.055	0.079	0.068
AIME25	0.025	0.042	0.060
GPQA-Diamond	0.043	0.059	0.050
KnK	0.490	0.719	–

(b) 最后 m 步的端点恢复

模型	基准测试	All 轨迹	最后 m 步数	Base 模型
Qwen3-1.7B-Base ($m = 1$)	MATH-500	0.004	0.007	0.079
	AIME25	0.007	0.008	0.042
	KnK	0.003	0.009	0.719
SmolLM2-360M ($m = 50$)	GSM8K	0.003	0.009	0.226

即使在 AIME25 上，两次运行的 pass@16 都有波动，学生模型也会随在线运行上下起伏；在 KnK 上，它则跟随在线 MaxRL，从接近零一路提升到最终准确率。表 ϵ_{traj}（方程（17））中的轨迹误差 2a 概括了这些曲线；对于 Qwen3-1.7B-Base，该误差在每项基准测试上都保持在约 0.01 或以下。此外，所有基准测试上的误差都低于基础模型，除 AIME25 外，在其他所有基准测试上都低了一个数量级。SFT 学生模型从未接触奖励，优化的目标函数也不同，却能以与在线运行相同的速度提升，因此仅凭保存的 rollout 就足以恢复在线强化学习。

端点恢复。 在所有轨迹上训练的学生模型也能恢复最终检查点。图 5展示了两次运行最终的 pass@k 结果，涵盖 k，上限为 64（KnK 上为 32）。对于 Qwen3-1.7B-Base，监督微调曲线在所有基准测试和每个 k上几乎都与在线曲线重合；Qwen3-4B-Base 在 MATH-500、AIME25 和 GPQA-Diamond 上也呈现相同模式，因此结果并不取决于模型规模。表 ϵ_{end}（方程（18））中的端点误差 2a 对于两个模型都很小；在大多数基准测试上，Qwen3-1.7B-Base 的端点误差比基础模型低一个数量级。同样的结论也适用于其他在线强化学习算法：对 Qwen3-1.7B-Base 的 GRPO 和 RLOO 运行轨迹进行监督微调，也能恢复每次运行最终的 pass@k 结果（附录 B.1）。各尺寸 KnK 谜题的结果见附录 B.2。

如果我们只关心最终性能，第 3.2 节表明，与其使用完整的归档采样轨迹，不如在最后 m个步骤的轨迹上进行训练（方程（13））。这几个批次远小于完整的归档采样轨迹，因此学生模型会多次遍历它们。它会依次对每个批次进行一次监督微调更新，从 D_{T-m} 到 D_{T-1}，然后从 D_{T-m}重新开始。我们在 Polaris-53K 和 KnK 上的 Qwen3-1.7B-Base 运行，以及第 4.1节的 SmolLM2-360M-Instruct 运行上测试了这一方法，并将学生模型与最终在线检查点进行比较（图 6）。详细设置见附录 A.1。对于 Qwen3-1.7B-Base，仅最后一个步骤（$m = 1$）就足够了；在 MATH-500、KnK 和 AIME25 上，无论 k取何值，监督微调曲线都与在线曲线保持接近。事实上，并不需要该步骤的所有轨迹，因为随机抽取其中一半，就已经能在 MATH-500 上达到与在线运行相当的结果；而在 Minerva 上，只需更少的轨迹即可（图 7 和附录 B.3）。

然而，单个批次并不适用于每个模型。对于 GSM8K 上的 SmolLM2-360M-Instruct，在最后一步的轨迹上进行监督微调，其表现始终低于在线运行 k；而最后 5 步的轨迹会缩小差距，最后 50 步的轨迹则能弥合其中大部分差距。对于这两个模型，使用这些较晚阶段的轨迹时，端点误差都接近于在所有轨迹上进行监督微调的结果，并且远低于基础模型的结果（表2b）。这些结果表明，最终策略在很大程度上已经包含于最新的轨迹中，因此无需额外花费时间或计算资源，从最终策略中采样新的响应即可恢复该策略。

要点 2

按原始顺序对存档轨迹进行监督微调，可以完整复现在线强化学习的整个学习曲线。仅使用最后几个强化学习步骤的轨迹，就能恢复其最终策略。

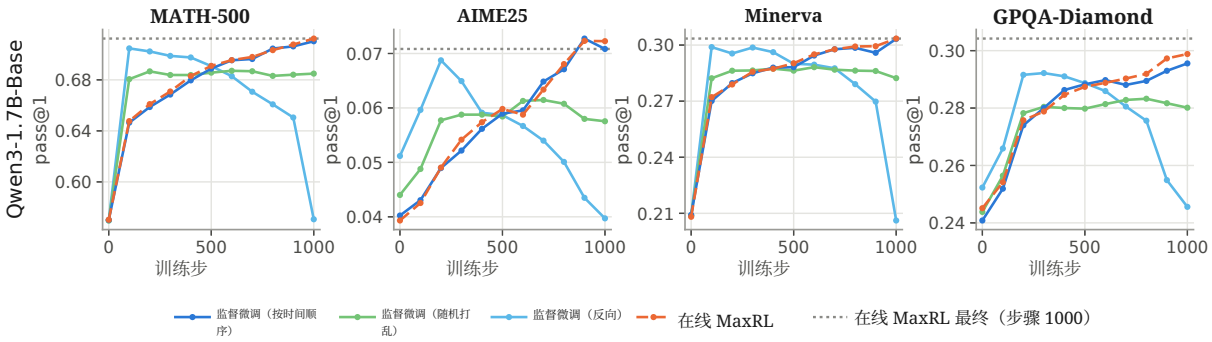


图 8：回放顺序干预。在 Polaris 上的 Qwen3-1.7B-Base：按时间顺序、随机打乱和逆序对相同的已存档 MaxRL 采样轨迹进行监督微调，并与在线 MaxRL 进行比较。展示训练过程中的 pass@1。灰色虚线表示在线 MaxRL 的最终值；AIME25 和 GPQA-Diamond 的数据经过 3 次评估平滑处理。

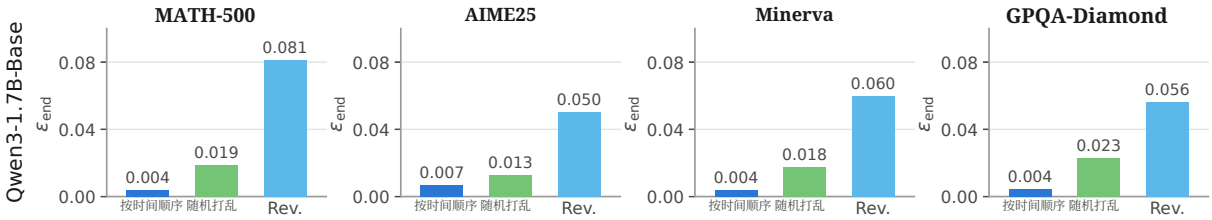


图 9：各回放顺序的端点误差。端点误差 ϵ_{end} （方程（18）），即对图8 中的轨迹按时间顺序、随机打乱和逆序进行监督微调后，在第 1000 步时的误差。

4.3 按时间顺序、逆序和随机打乱回放

我们按照第 3.3 节中的三种顺序，重放 Qwen3-1.7B-Base 在 Polaris-53K 上 MaxRL 运行的相同轨迹，并报告训练期间的 pass@1 和最终学生模型的端点误差 ϵ_{end} （方程（18））。

如图 8所示，只有按时间顺序回放才能恢复训练曲线。在 MATH-500 上，反向回放在训练早期接近最终在线策略，但随后又退回到基础模型的水平；打乱顺序回放则停滞在在线 MaxRL 之下，而按时间顺序回放始终紧随在线 MaxRL。Minerva（Lewkowycz 等人，2022年）和 GPQA-Diamond 也呈现出相同的模式。图 9 中的最终检查点证实了所有基准测试上的这一排序：按时间顺序回放的端点误差最小，打乱顺序回放的误差较大，反向回放的误差最大，正如方程（14）所预测的那样。

要点 3

只有按时间顺序回放才能追踪在线强化学习。轨迹的顺序决定了学生模型的演进路径。

4.4 组合专精模型

强化学习专家模型通常会分别训练，因为不同任务适合不同的超参数、提示分布和训练时长。它们通常通过合并权重来组合，方法包括权重平均（Wortsman 等人，2022年）、任务算术（Ilharco 等人，2023年）和 TIES（Yadav 等人，2023年）。这些方法可能会受到任务特定更新之间干扰的影响。我们则使用所有专精模型的轨迹进行监督微调（第 3.4 节），训练单个学生模型，并将这种方法称为数据空间组合。

在所有轨迹上训练。我们研究了两个使用 MaxRL 训练的 Qwen3-1.7B-Base 专用模型，一个针对 Polaris，另一个针对 KnK。按照方程（15），学生模型从相同的基础模型开始，并按时间顺序重放两个档案。在每个强化学习步骤 t，它都会进行一次 SFT 更新，使用 D_t^1，然后再使用D_t^2进行一次更新，因此它看到的是两个专用模型在同一训练阶段产生的轨迹。我们每隔 100 个强化学习步骤，就在 MATH-500、AIME25 和 Minerva 上评估其数学能力，并在各种规模的 KnK 谜题上进行评估；随后将其与各任务的在线专用模型进行比较（图 10）。尽管它同时针对两个任务进行训练，

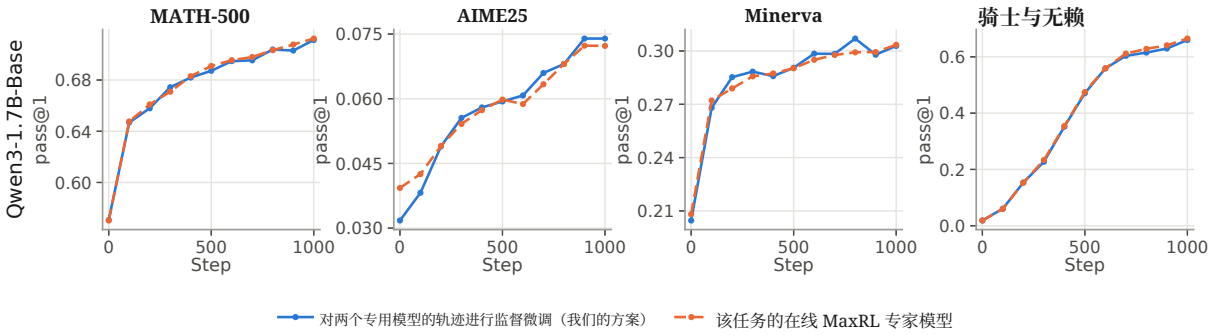


图 10：对两种强化学习专家模型的所有轨迹进行监督微调，会同时遵循两条训练曲线。 Qwen3-1.7B-Base：按时间顺序交替进行一次更新，在 Polaris 和 KnK MaxRL 专家模型的轨迹上通过监督微调训练一个学生模型，并比较其在强化学习训练过程中的 pass@1 与每个任务的在线专用模型。AIME25 的结果取 3 次评估的平滑值。

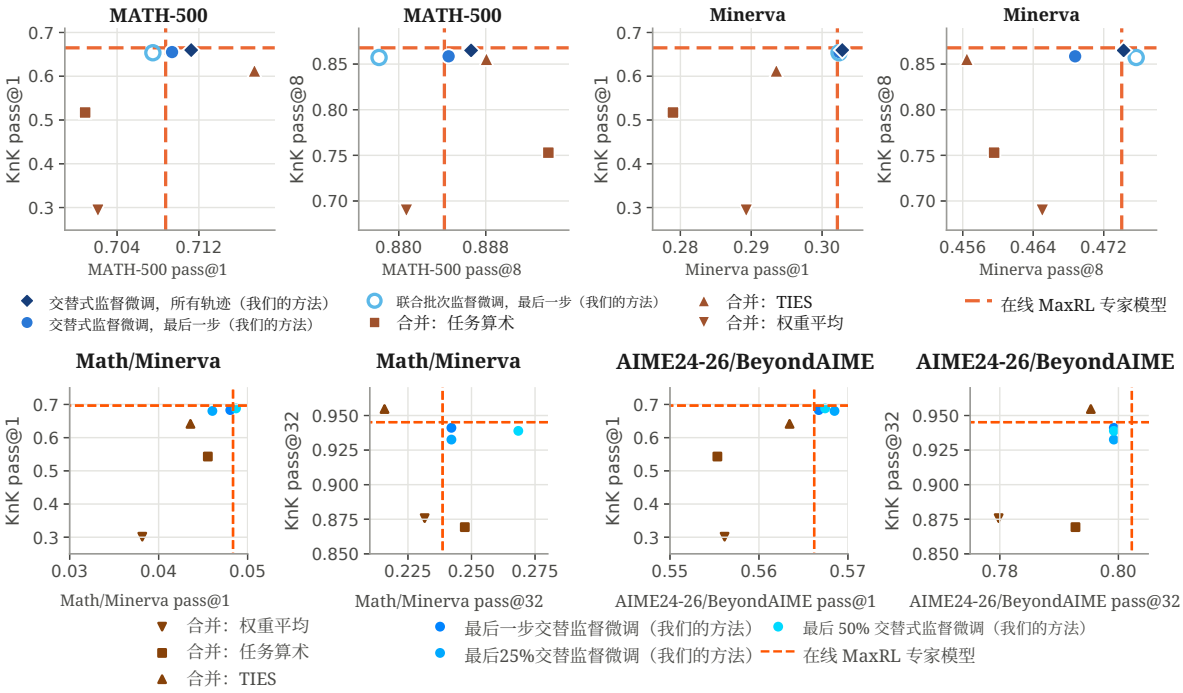


图 11：组合两个强化学习专家模型。 Qwen3-1.7B-Base：在两个 Polaris 基准测试 MATH-500 和 Minerva 上的 pass@k，对比 Knights-and-Knaves 的 pass@k，其中 $k = 1$ 和 8。数据空间组合通过按时间顺序对两个专家模型的所有轨迹进行监督微调（深蓝色），或对它们的最后一步展开进行监督微调（300 次更新）来训练。该方法与合并两个最终检查点的权重空间合并进行比较。虚线表示每个专家模型在其自身任务上的表现，因此右上角为最佳。每种方法的计算资源和词元预算见附录 D.2。

学生模型在各自任务上的表现始终沿着对应专用模型的训练曲线，从基础模型一直到最终检查点。这正是方程（15）的预测结果，因为这两个任务没有共享提示词。

在最后一步展开上进行训练。 如第 4.2节所述，如果只关注最终策略，学生模型也可以改为在每个专用模型最后一个强化学习步骤的轨迹上训练 300 次更新。我们考虑两种合并这两组轨迹的方式。 交替式监督微调 依次将更新分配给两个任务，因此每次更新只使用一个专用模型的轨迹。 联合批次监督微调 则将两个专用模型的轨迹放入同一次更新中。我们将二者分别与上述三种权重合并进行比较，并在附录 D中介绍详细的训练方案。每个模型都会在两个任务上进行评估：对于数学任务，使用 MATH-500 和 Minerva 上的 pass@k；对于 KnK，则使用所有规模谜题上的 pass@k。图11 绘制了 $k = 1$ 和 $k = 8$之间的对应关系，同时给出了每个专用模型在其自身任务上的得分，以及在所有轨迹上训练得到的最终学生模型的得分。

两个最后一步学生模型都接近两条虚线的交点；在该交点处，模型将同时匹配两个专用模型，而使用所有轨迹训练的学生模型几乎正好位于交点上。这一现象在两个数学基准测试和 k 的两个取值下都成立，两个监督微调（SFT）变体的表现也相近。合并后的模型则

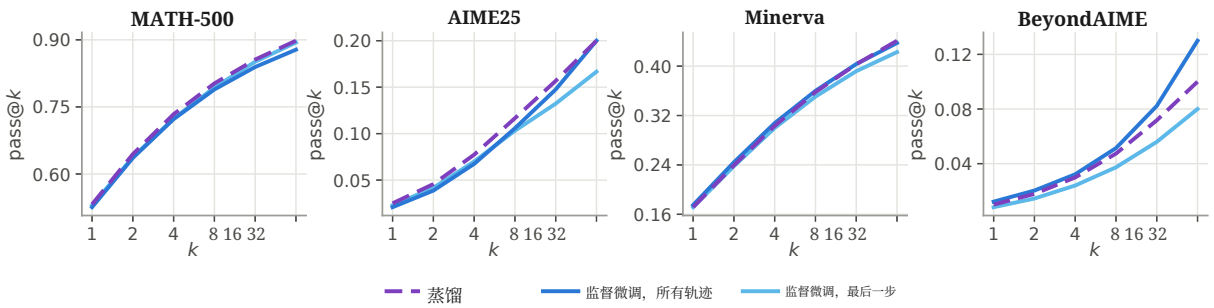


图12：从 Qwen3-1.7B-Base 迁移到 Qwen3-0.6B-Base。在 Polaris-53K 上使用 Qwen3-1.7B-Base 的在线 MaxRL 运行的最终策略（蒸馏）、所有轨迹或最后一步的轨迹中的样本进行监督微调后，Qwen3-0.6B-Base 的 Pass@k 。

离交点更远，这主要是因为它们损失了大量 KnK 准确率（更多结果见附录 D ）。

在后期 rollout 上训练。 在这两种设置之间，我们分别使用每个专用模型强化学习训练最后 50% 和最后 25% 的轨迹进行交替式监督微调，并将两者与三种合并方式以及使用最后一步展开进行的交替式监督微调进行比较。对于数学任务，我们报告了 MATH-500、Minerva、AIME24–26 和 BeyondAIME 上 $k = 1$ 与 $k = 32$ 的 pass@k 。通常，使用更大比例的轨迹能够提升组合学生模型的表现。在 MATH-500 和 Minerva 上，使用最后 50% 轨迹训练的学生模型在 pass@1 和 pass@32 两项指标上甚至超过了 MaxRL 专精模型；在 AIME24–26 和 BeyondAIME 上，它在 pass@1 上也超过了 MaxRL 专精模型。三个回放训练的学生模型都比任何一种权重合并更接近两个专精模型的交集，唯一的例外是 KnK 的 pass@32 指标上使用的 TIES。

要点 4

对独立专精模型的所有轨迹进行监督微调，可以得到一个模型，使其在各自任务上的训练曲线都遵循对应专精模型的训练曲线。即使仅使用最后一步展开，也保留了各专精模型所学内容的大部分，而且效果优于权重合并。

4.5 跨模型迁移

对于跨模型迁移（第 3.4 节），我们使用某次模型 B 运行的轨迹进行监督微调，训练一个模型 A。我们将 Qwen3-1.7B-Base 在 Polaris-53K 上的在线 MaxRL 运行作为源模型（A），将 Qwen3-0.6B-Base 作为学生模型（B）。我们采用第 4.2 节中的两种方式，使用保存的轨迹训练学生模型：要么按照原始顺序，对 1000 个 rollout 批次中的每个批次分别进行一次 SFT 更新；要么对最后一个强化学习步骤的轨迹进行 200 次更新。由于不能期待较小的学生模型匹配较大的最终策略，因此我们将其与从最终策略 π_T A （方程（16））进行蒸馏的结果进行比较，而不是在 A 上进行在线强化学习。我们在 MATH-500、AIME25、Minerva 和 BeyondAIME（ByteDance-Seed，2025年）上评估全部三个学生模型，并报告 k 从 k 1 到 32 的 pass@。

如图12所示，在所有基准测试上，使用保存的 rollout 训练的两个学生模型都紧密遵循蒸馏的 pass@k 曲线，因此，在训练不同模型时，保存的 rollout 可以取代从最终策略获取的新样本。

要点 5

强化学习运行产生的轨迹，既可以用于训练不同的模型，也可以像从其最终策略进行蒸馏那样发挥作用。

5 相关工作

从自生成与重放数据中学习。自训练方法会使用模型自身生成的样本对模型进行微调，通常只保留奖励较高的样本，并在每一轮从不断改进的模型重新生成数据（Zelikman 等人，2022年；Singh 等人，2024年；Dong 等人，2023年；Yuan 等人，2023年）。在基于价值的强化学习中，经验回放会重复利用存储的状态转移（Mnihet 等人，2015年）；而对来自其他策略的数据使用策略梯度则需要进行校正（Degris 等人，2012年），这正是在线 MaxRL

目标函数在离线运行时会失效。近期的大语言模型系统在在线训练中能够容忍略微陈旧的轨迹（Noukhovitch 等人，2025年；Sun 等人，2025年）。与我们的工作最接近的是 ROSS（Zhang 等人，2026年），它使用历史轨迹中筛选出的续写对较晚的检查点进行微调。与之不同，我们重放来自基础模型的固定、未经筛选、按步骤索引的存档，不生成任何新轨迹，并探究它能否重建完整的在线轨迹。

蒸馏和监督微调与强化学习。 在教师模型的样本上进行监督微调属于序列级蒸馏（Kim 和 Rush，2016年），而将训练好的强化学习策略蒸馏到学生模型中，则可追溯至策略蒸馏（Rusu 等人，2016年）。同策略蒸馏（Agarwal et al.，2024年）指出，仅使用教师模型的输出进行训练会导致训练与推理不匹配，这与我们的跨模型迁移有关。以往的蒸馏使用单个最终教师模型；我们则蒸馏强化学习运行过程中一系列策略，而重放顺序的影响使我们的研究与课程学习相关（Bengio etal.，2009年）。我们的结果也与关于强化学习相较于监督微调有何增益的争论有关。Chuet al. (2025年) 报告称，监督微调会导致记忆，而强化学习能够实现泛化；但另一些研究发现，强化学习主要是重新调整基础模型中已有行为的权重（Yue et al.，2025年），只要数据合适，监督微调也能实现这一过程（Huang etal.，2025年），而且强化学习产生的行为与其同策略数据密切相关（Shenfeldet al.，2026年）。我们的重建结果支持这一以数据为中心的观点：在强化学习自身的轨迹上进行监督微调，可以复现其学习曲线。

模型合并与多教师整合。 权重空间方法通过对微调模型取平均值（Wortsman 等人，2022年）、添加任务向量（Ilharco 等人，2023年）或解决参数干扰问题（Yadav 等人，2023年）来组合模型，但经过强化学习训练的模型在此类合并后性能会显著下降（Wu 等人，2026年）。教师空间方法将多个在线专精模型蒸馏到一个学生模型中，并使用学生模型自身的轨迹（Ma 等人，2026年；Gao 等人，2026年），而联合多领域强化学习则在所有任务上训练单一策略（Wang 等人，2026年）。数据空间组合提供了第三种选择：它既不需要专用模型的权重，也不需要在线教师，只需每个专用模型的存档轨迹。

6 结论

在这项工作中，我们探讨模型在强化学习后训练期间学到的内容有多少保留在其轨迹中。在保存的轨迹上重放原始强化学习算法会导致训练崩溃，因为相对于生成这些轨迹的策略出现的微小偏差会使更新产生偏差，并在训练过程中不断累积。在相同轨迹上进行最大似然训练则可以避免这一反馈回路。从相同的基础模型出发，并按照生成轨迹的顺序，每个轨迹批次进行一次监督微调更新，就能紧密追踪在线强化学习的整个学习曲线，无需奖励，也无需区分正确响应和错误响应。这一结论适用于不同的模型规模、任务和强化学习算法；而且，最后几个强化学习步骤生成的轨迹往往就足以恢复最终模型。这些结果表明，强化学习所投入的大部分计算资源，可能用于发现更优的采样分布，而非从样本中学习该分布。将轨迹视为数据也开辟了新的用途。轨迹的顺序决定了学生模型的学习路径；多个独立专用模型的轨迹组合成一个模型，效果优于合并它们的权重；而一个模型的轨迹训练较小模型的效果，也不逊于从其最终策略进行蒸馏。我们的实验仅涉及参数规模最大为 4B 的模型，目标是恢复在线强化学习的效果，而非超越它。将方法扩展到更大的模型，以及重用轨迹以超越在线强化学习，都是值得探索的未来方向。

参考文献

Rishabh Agarwal、Nino Vieillard、Yongchao Zhou、Piotr Stanczyk、Sabela Ramos、Matthieu Geist 和 Olivier Bachem。同策略蒸馏语言模型：从自生成错误中学习。载于国际学习表征会议（*ICLR*），2024年。

Arash Ahmadian、Chris Cremer、Matthias Gallé、Marzieh Fadaee、Julia Kreutzer、Olivier Pietquin、Ahmet Üstün 和 Sara Hooker。回归基础：重新审视用于从人类反馈中学习的大语言模型中的 REINFORCE 风格优化。载于第 *62* 届计算语言学协会年会论文集（第 *1* 卷：长篇论文），第 12248–12267 页，泰国曼谷，2024年。计算语言学协会。doi：10.18653/v1/2024.acl-long.662。

Chenxin An、Zhihui Xie、Xiaonan Li、Lei Li、Jun Zhang、Shansan Gong、Ming Zhong、Jingjing Xu、Xipeng Qiu、Mingxuan Wang 和 Lingpeng Kong。《POLARIS：一种用于扩展高级推理模型强化学习的后训练方案》。https://hkunlp.github.io/blog/2025/Polaris，2025年。

Hao Bai、Chenlu Ye、Xuanchang Zhang、Aviral Kumar 和 Tong Zhang。《正则化分数中心化》。*arXiv* 预印本 *arXiv:2611.21791*，2026年。Loubna Ben Allal、Anton Lozhkov、Elie Bakouch、Gabriel Martín Blázquez、Guilherme Penedo、Lewis Tunstall 等。《SmolLM2：小模型如何实现大作为——小型语言模型的以数据为中心的训练》。*arXiv* 预印本 *arXiv:2502.02737*，2025年。Yoshua Bengio、Jérôme Louradour、Ronan Collobert 和 Jason Weston。《课程学习》。载于第*26*届国际机器学习会议（*ICML*）论文集，第41–48页，2009年。doi：10.1145/1553374.1553380。ByteDance-Seed。BeyondAIME。https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME，2025年。Tianzhe Chu、Yuexiang Zhai、Jihan Yang、Shengbang Tong、Saining Xie、Dale Schuurmans、Quoc V. Le、Sergey Levine 和 Yi Ma。《监督微调会记忆，强化学习能泛化：基础模型后训练的比较研究》。载于第*42*届国际机器学习会议论文集，第267卷，机器学习研究论文集，第10818–10838页，2025年。

Karl Cobbe、Vineet Kosaraju、Mohammad Bavarian、Mark Chen、Heewoo Jun、Lukasz Kaiser、Matthias Plappert、Jerry Tworek、Jacob Hilton、Reiichiro Nakano、Christopher Hesse 和 John Schulman。训练验证器解决数学文字题。*arXiv* 预印本 *arXiv:2110.14168*，2021年。

Thomas Degris、Martha White 和 Richard S. Sutton。离策略行动者-评论家。载于第*29*届国际机器学习会议论文集，2012年。

Jia Deng、Wei Dong、Richard Socher、Li-Jia Li、Kai Li 和 李飞飞。ImageNet：大规模层次化图像数据库。载于*IEEE*计算机视觉与模式识别会议，2009年。

Hanze Dong、Wei Xiong、Deepanshu Goyal、Yihan Zhang、Winnie Chow、Rui Pan、Shizhe Diao、Jipeng Zhang、Kashun Shum 和 Tong Zhang。RAFT：面向生成式基础模型对齐的奖励排序微调。机器学习研究汇刊，2023年。

Huan-ang Gao、Haohan Chi、Yong Yan、Shiyuan Feng、Hanlin Wu、Zheng Jiang、Bingxiang He、Wei- Ying Ma、张亚勤和Hao Zhou。Open-mopd：诊断并修复多教师同策略蒸馏中的能力失衡。*arXiv*预印本 *arXiv:2608.19098*，2026年。

Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Peiyi Wang、Qihao Zhu、RunxinXu等。DeepSeek-R1通过强化学习激励大语言模型进行推理。*Nature*，645(8081):633–638，2025年。doi：10.1038/s41586-025-09422-z。arXiv:2501.12948。

何恺明、Xiangyu Zhang、Shaoqing Ren和Jian Sun。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议，2016年。

Dan Hendrycks、Collin Burns、Saurav Kadavath、Akul Arora、Steven Basart、Eric Tang、Dawn Song 和 Jacob Steinhardt。使用 MATH 数据集 测量数学问题求解。载于神经信息处理系统进展数据集与基准赛道，2021。

Audrey Huang、Adam Block、Dylan J. Foster、Dhruv Rohatgi、Cyril Zhang、Max Simchowitz、Jordan T. Ash 和 Akshay Krishnamurthy。语言模型的自我改进：锐化机制。载于国际学习表征会议 *(ICLR)*，2025年。

Gabriel Ilharco、Marco Tulio Ribeiro、Mitchell Wortsman、Suchin Gururangan、Ludwig Schmidt、Hannaneh Hajishirzi 和 Ali Farhadi。使用任务算术编辑模型。载于国际学习表征会议 *(ICLR)*，2023年。arXiv:2212.04089。

Yoon Kim 和 Alexander M. Rush。序列级知识蒸馏。载于*2016*年自然语言处理实证方法会议论文集，第 1317–1327 页，2016年，得克萨斯州奥斯汀。计算语言学协会。doi：10.18653/v1/D16-1139。

Nathan Lambert、Jacob Morrison、Valentina Pyatkin、Shengyi Huang、Hamish Ivison、Faeze Brahman、Lester James V. Miranda、Alisa Liu、Nouha Dziri、Shane Lyu、Yuling Gu、Saumya Malik、Victoria Graf、Jena D. Hwang、Jiangjiang Yang、Ronan Le Bras、Oyvind Tafjord、Chris Wilhelm、Luca Soldaini、Noah A. Smith、Yizhong Wang、Pradeep Dasigi 和 Hannaneh Hajishirzi。《Tülu 3：推动开放语言模型后训练的前沿》。载于语言建模会议（*COLM*），2025年。arXiv:2411.15124。

Sergey Levine、Aviral Kumar、George Tucker 和 Justin Fu。《离线强化学习：教程、综述及开放问题展望》。*arXiv* 预印本 *arXiv:2005.01643*，2020年。

Aitor Lewkowycz、Anders Andreassen、David Dohan、Ethan Dyer、Henryk Michalewski、Vinay Ramasesh、Ambrose Slone、Cem Anil、Imanol Schlag、Theo Gutman-Solo 等。使用语言模型解决定量推理问题。载于《神经信息处理系统进展》，2022年。

Hunter Lightman、Vineet Kosaraju、Yura Burda、Harri Edwards、Bowen Baker、Teddy Lee、Jan Leike、John Schulman、Ilya Sutskever 和 Karl Cobbe。让我们逐步骤进行验证。载于国际学习表征会议，2024年。

Wenhan Ma、Jianyu Wei、Liang Zhao、Hailin Zhang、Bangjun Xiao、Lei Li、Qibin Yang、Bofei Gao、Yudong Wang、Rang Li、Jinhao Dong、Zhifang Sui 和 Fuli Luo。Mopd：用于大语言模型后训练能力整合的多教师同策略蒸馏。*arXiv* 预印本 *arXiv:2606.30406*，2026年。

Martin Marek 和 Max Ryabinin。分数中心化可稳定离策略强化学习。*arXiv* 预印本 *arXiv:2609.20807*，2026年。

美国数学协会。美国数学邀请赛（AIME）。https://maa.org/maa-invitational-competitions/，2025年。

Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A. Rusu、Joel Veness、Marc G. Bellemare、Alex Graves、Martin Riedmiller、Andreas K. Fidjeland、Georg Ostrovski、Stig Petersen、Charles Beattie、Amir Sadik、Ioannis Antonoglou、Helen King、Dharshan Kumaran、Daan Wierstra、Shane Legg 和 Demis Hassabis。通过深度强化学习实现人类水平的控制。*Nature*，518(7540):529–533，2015年。doi: 10.1038/nature14236。

Michael Noukhovitch、Shengyi Huang、Sophie Xhonneux、Arian Hosseini、Rishabh Agarwal 和 Aaron Courville。《异步RLHF：面向语言模型的更快速、更高效的离策略强化学习》。发表于国际学习表征会议，2025年。

Penghui Qi、Zichen Liu、Xiangxin Zhou、Tianyu Pang、Chao Du、Wee Sun Lee 和 Min Lin。《通过半精度浮点数解决训练-推理不匹配问题》。*arXiv* 预印本 *arXiv:2510.26788*，2025年。

David Rein、Betty Li Hou、Asa Cooper Stickland、Jackson Petty、Richard Yuanzhe Pang、Julien Dirani、Julian Michael 和 Samuel R. Bowman。《GPQA：研究生级谷歌防作弊问答基准》。发表于语言建模会议，2024年。

Andrei A. Rusu、Sergio Gomez Colmenarejo、Çağlar Gülçehre、Guillaume Desjardins、James Kirkpatrick、Razvan Pascanu、Volodymyr Mnih、Koray Kavukcuoglu 和 Raia Hadsell。《策略蒸馏》。发表于国际学习表征会议 *(ICLR)*，2016年。

John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。*arXiv* 预印本 *arXiv:1707.06347*，2017年。

Zhihong Shao、Peiyi Wang、Qihao Zhu、Runxin Xu、Junxiao Song、Xiao Bi、Haowei Zhang、Mingchuan Zhang、Y. K. Li、Y. Wu 和 Daya Guo。DeepSeekMath：推动开放语言模型数学推理能力的极限。*arXiv* 预印本 *arXiv:2402.03300*，2024年。

Idan Shenfeld、Jyothish Pari 和 Pulkit Agrawal。强化学习剃刀：为什么在线强化学习遗忘较少。发表于国际学习表征会议（*ICLR*），2026年。arXiv:2509.04259。

Avi Singh、John D. Co-Reyes、Rishabh Agarwal、Ankesh Anand、Piyush Patil、Xavier Garcia、Peter J. Liu、James Harrison、Jaehoon Lee、Kelvin Xu、Aaron Parisi、Abhishek Kumar、Alex Alemi、Alex Rizkowsky、Azade Nova、Ben Adlam、Bernd Bohnet、Gamaleldin Elsayed、Hanie Sedghi、Igor Mordatch、Isabelle Simpson、Izzeddin Gur、Jasper Snoek、Jeffrey Pennington、Jiri Hron、Kathleen Kenealy、Kevin Swersky、Kshiteej Mahajan、Laura Culp、Lechao Xiao、Maxwell L. Bileschi、Noah Constant、Roman Novak、Rosanne Liu、Tris Warkentin、Yundi Qian、Yamini Bansal、Ethan Dyer、Behnam Neyshabur、Jascha Sohl-Dickstein 和 Noah Fiedel。超越人类数据：利用语言模型扩展问题求解自训练。机器学习研究汇刊，2024年。

Yifan Sun、Jingyan Shen、Yibin Wang、Tianyu Chen、Zhendong Wang、Mingyuan Zhou 和 Huan Zhang。通过难度定向在线数据选择和轨迹重放提高大语言模型强化微调的数据效率。载于神经信息处理系统进展，2025年。

Fahim Tajwar、Guanning Zeng、Yueer Zhou、Yuda Song、Daman Arora、Yiding Jiang、Jeff Schneider、Ruslan Salakhutdinov、Haiwen Feng 和 Andrea Zanette。最大似然强化学习。载于第*43*届国际机器学习会议论文集，第306卷机器学习研究论文集，第118190–118245页，2026年。网址：https://proceedings.mlr.press/v306/tajwar26a.html。

Haoqing Wang、Xiang Long、Ziheng Li、Yilong Xu、Tingguang Li 和 Yehui Tang。《混合还是合并：面向大语言模型的多领域强化学习》。发表于语言建模会议（*COLM*），2026年。arXiv:2602.12566。

Yubo Wang、Xueguang Ma、Ge Zhang、Yuansheng Ni、Abhranil Chandra、Shiguang Guo、Weiming Ren、Aaran Arulraj、Xuan He、Ziyan Jiang 等。《MMLU-Pro：更稳健、更具挑战性的多任务语言理解基准》。发表于神经信息处理系统进展数据集与基准赛道，2024年。

Ronald J. Williams。《用于联结主义强化学习的简单统计梯度跟随算法》。机器学习，8(3-4)：229–256，1992年。doi：10.1007/BF00992696。

Mitchell Wortsman、Gabriel Ilharco、Samir Yitzhak Gadre、Rebecca Roelofs、Raphael Gontijo-Lopes、Ari S. Morcos、Hongseok Namkoong、Ali Farhadi、Yair Carmon、Simon Kornblith 和 Ludwig Schmidt。模型汤：对多个微调模型的权重进行平均，可在不增加推理时间的情况下提升准确率。发表于国际机器学习会议（*ICML*），2022年。arXiv:2203.05482。

Chenrui Wu、Zexi Li、Jiajun Bu、Jiangchuan Liu 和 Haishuai Wang。稀疏性诅咒：通过模型合并理解 RLVR 模型参数空间。发表于*ACM SIGKDD*知识发现与数据挖掘会议（*KDD*），2026年。arXiv:2606.18521。

Chulin Xie、Yangsibo Huang、Chiyuan Zhang、Da Yu、Xinyun Chen、Bill Yuchen Lin、Bo Li、Badih Ghazi 和 Ravi Kumar。关于大语言模型在逻辑推理中的记忆。*arXiv* 预印本 *arXiv:2410.23123*，2024年。

Prateek Yadav、Derek Tam、Leshem Choshen、Colin Raffel 和 Mohit Bansal。《TIES-Merging：解决模型合并时的干扰》。载于神经信息处理系统进展，2023年。

An Yang、Anfeng Li、Baosong Yang、Beichen Zhang、Binyuan Hui、Bo Zheng、Bowen Yu、Chang Gao、Chengen Huang、Chenxu Lv 等。《Qwen3 技术报告》。*arXiv* 预印本 *arXiv:2505.09388*，2025年。

Zheng Yuan、Hongyi Yuan、Chengpeng Li、Guanting Dong、Keming Lu、Chuanqi Tan、Chang Zhou 和 Jingren Zhou。《使用大语言模型学习数学推理的规模关系》。*arXiv* 预印本 *arXiv:2308.01825*，2023年。

Yang Yue、Zhiqi Chen、Rui Lu、Andrew Zhao、Zhaokai Wang、Yang Yue、Shiji Song 和 Gao Huang。《强化学习真的能在基础模型之外激发大语言模型的推理能力吗？》载于神经信息处理系统进展（*NeurIPS*），2025年。

Eric Zelikman、Yuhuai Wu、Jesse Mu 和 Noah D. Goodman。STaR：通过推理自举推理。载于神经信息处理系统进展，第35卷，2022年。

Zhiwei Zhang、Huayu Deng、Fei Zhao、Jiayan Fu、Bin Liang、Kam-Fai Wong 和 Chuan Mu。Ross：通过选择性监督，从自生成轨迹中重新学习。*arXiv* 预印本 *arXiv:2609.35954*，2026年。

A 实验设置

所有在线运行和重放均在 verl 框架中实现。除非另有说明，监督微调会最小化所有归档响应（无论正确与否）的标记级负对数似然；它从与在线运行相同的初始化开始，并按照原始顺序，对每个归档的 RL 步骤执行一次监督微调更新。在使用 MaxRL 优势时，奖励为 r 的响应具有优势 $(r-\hat{}p)/(\hat{}p+10^{-6})$，其中 $\hat{}p$ 是针对同一提示词（或图像）采样的轨迹的平均奖励。

A.1 语言模型设置

表 3 列出了图 4、5和 6背后的设置。Polaris 模型在 MATH-500、AIME25、GPQA-Diamond、Minerva（Lewkowycz 等人，2022年）以及 BeyondAIME（ByteDance-Seed，2025年）上进行评估。除非另有说明，所有语言模型评估均以 0.6 的温度和 top-p 0.95 进行采样，并报告无偏 pass@k 。

在最后 m 步进行监督微调（第 4.2 节）时，学生模型按顺序遍历批次 D_{T-m}、……、D_{T-1}，每个批次进行一次监督微调更新，然后从 D_{T-m} 开始，在 D_{T-1} 之后重新循环。更新次数达到表 3 中所列数量后停止。

表 3： 语言模型设置。“步骤”指一次在线强化学习更新；一次监督微调更新会使用一个归档步骤的轨迹。

设置	在线 MaxRL	重放	评估
Qwen3-1.7B-Base, Polaris-53K	1000 步数，256 个提示 × 每步骤 16 条轨迹 $=4096$ 。学习率 10^{-6}，恒定	监督微调：AdamW，学习率 10^{-6}，恒定，权重衰减 0.01，梯度裁剪 1.0，max 长度 6144；1000 次更新。最后仅步骤：第 1000 步的 4096 条轨迹，200 次更新。离线 MaxRL：相同的轨迹和学习率，采用在线 MaxRL 优势 $(r-\hat{}p)/(\hat{}p+10^{-6})$ 计算得出，基于每条轨迹的 16 个存储的奖励提示词，每个存储项更新一次 step	图 4和 5：128 样本 per 问题，每100步。Figure 6：512个样本
Qwen3-4B-Base, Polaris-53K	1000 步数，256 个提示 × 每个步骤 16 条轨迹 $=4096$ ；学习率 10^{-6}，恒定	对存档轨迹进行监督微调，位于顺序：AdamW，学习率 10^{-6}，恒定，权重衰减 0.01，梯度裁剪 1.0，max 长度 6144；1000 次更新	图 5：1024 个样本每个问题，最终检查点
Qwen3-1.7B-Base, 骑士 and Knaves	1000 步数，256 个谜题 × 每个步骤 16 条轨迹。学习率 10^{-6}，恒定	监督微调：AdamW，学习率 10^{-6}，恒定，权重衰减 0.01，最大长度 8192；1000 次更新。 仅最后一步：第 1000 步的轨迹，已评估经过 500 次更新后	监督微调：每道谜题 32 个样本。在线：每道谜题 32 个采样，每 100 步，最终为 128 个检查点
SmolLM2-360M-Instruct，GSM8K	1500 步，256 个提示 × 每个步骤 128 条轨迹；AdamW，学习率 10^{-5}，恒定；响应长度 2048	监督微调：AdamW，学习率 10^{-5}，恒定，在最后 1、5 或 50 个步骤；全部在 420 次更新后进行比较	每个问题 32 个样本；对于 $k>1$，pass@k 是由 verl 计算的自助法估计计算所得

A.2 迷宫

策略经过训练以解决 17×17 迷宫。在线 MaxRL 运行和所有重放都从一个在示范上进行 1500 步 SFT 后的预热检查点开始。表 4 列出了图 17（左）的设置。全轨迹运行会将 1000 个强化学习步骤中的每一步重放 8 次，而最后一次k 运行会将最后 k 个步骤 $8000/k$ 重放若干次，因此两者都大约进行 8000 次 SFT 更新。对于 $k>1$，pass@k 是 verl 根据评估样本计算出的自助法估计。

表 4： 迷宫设置。

	在线 MaxRL	SFT 回放
初始化	迷宫_ckpt1500	same
每步数据	256 个迷宫 $\times$ 32 条轨迹 $= 8192$	一个归档 RL 步骤（8192 条轨迹）
长度（提示词 / 回复）	320 / 180 个词元	总计 512 个词元
目标函数	MaxRL 优势，不含 KL 散度	所有轨迹上的负对数似然
优化器	AdamW，学习率 10^{-4}，恒定	AdamW，学习率 3×10^{-5}，恒定，权重衰减 0.01，梯度裁剪 1.0
轨迹采样	温度 1.0	—
重放步数	–	全部：每隔 1–1000 个步骤，$8\times$；最近 k 个：步数 $1001 - k$ 至 1000，$8000/k\times$ 每个
评估	强化学习步骤 1000，每个迷宫 2048 个样本	最后一个检查点，每个迷宫 32 个样本
评估采样	温度 1.0，top-p 1.0	same

A.3 ImageNet

使用在线 MaxRL 在 ImageNet 上训练 ResNet-50，并使用 SFT、离线 REINFORCE 和离线 MaxRL 重放其存档轨迹（图17，右）。表5列出了相关设置。

表 5： ImageNet 设置。

	在线 MaxRL	离线重放
初始化	ResNet-50	same
每步数据	256 张图像 $\times$ 1024 轨迹 $= 262144$	一个归档 RL 步骤（262144 条轨迹）
输入分辨率	224×224 (来自随机调整大小裁剪的 RandomResizedCrop 256×256)	相同（224×224 评估时采用中心裁剪）
目标函数	MaxRL 优势，不使用 KL 散度	监督微调：对所有轨迹计算负对数似然；离线 REINFORCE：仅对正确轨迹计算负对数似然；离线 MaxRL：所有轨迹，按……加权 MaxRL 优势 $(r - \hat{p})/(\hat{p} + 10^{-6})$
优化器	随机梯度下降，学习率 0.1，动量 0.9，权重衰减 10^{-4}，采用余弦衰减，并进行 2 个 epoch 的预热，梯度裁剪 10^5	same
轨迹采样	softmax 温度 1.0，多项式采样有放回地，$K = 1024$ 每次 image	–
重放步数	–	全部：训练轮次 1–20，每个在线步骤的轨迹重放一次
评估	每 1000 步，解析式 pass@k，用于 $k = 1, 2, 4, \ldots\ldots, 1024$，基于 ImageNet 验证集	每 1000 步，以及每个重放训练轮次结束时，使用相同的指标
评估采样	–（pass@k，根据 softmax 概率解析计算得出，无需重新采样）	same

B 其他恢复结果

本节是对第 4节的补充，涵盖其他在线强化学习算法的轨迹、不同规模的 KnK 谜题、所需的最后一步轨迹数量，以及超越语言建模的任务。

B.1 其他在线强化学习算法

除了 MaxRL，我们还通过按时间顺序重放从训练循环中采样的轨迹，重现在线 GRPO 和 RLOO 的性能。对于在线强化学习和监督微调，我们采用附录 A.1中的相同超参数和实验设置，唯一的区别是所选的在线强化学习算法。图13 表明，按时间顺序的监督微调也能恢复 GRPO 和 RLOO 的在线运行情况。具体而言，对于这三种在线强化学习算法，我们观察到，在 AIME25-26 和 MATH500 上，SFT 回放在较大的 k 下，其 pass@ k 超过了原始在线运行。因此我们得出结论，恢复

这是各种在线强化学习算法中的普遍现象，而非 MaxRL 算法所特有的现象。

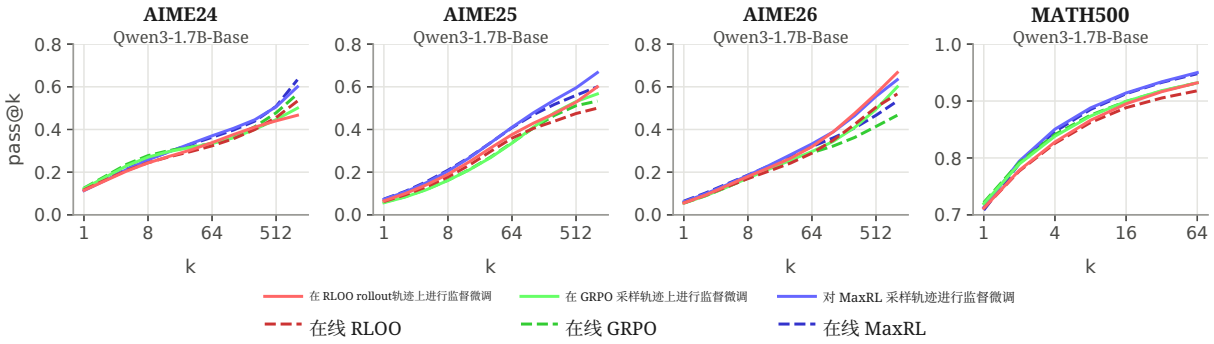


图 13：按时间顺序的监督微调能够恢复不同在线强化学习算法的最终检查点。在 MaxRL、GRPO 和 RLOO 运行过程中存档的轨迹上微调 Qwen3-1.7B-Base 后得到的最终检查点，其 pass@k 表现与相应的在线运行进行对比。

B.2 按谜题规模划分的 KnightsandKnaves

图14 和 15 按谜题规模细分了第 4 节中的 KnK 结果。

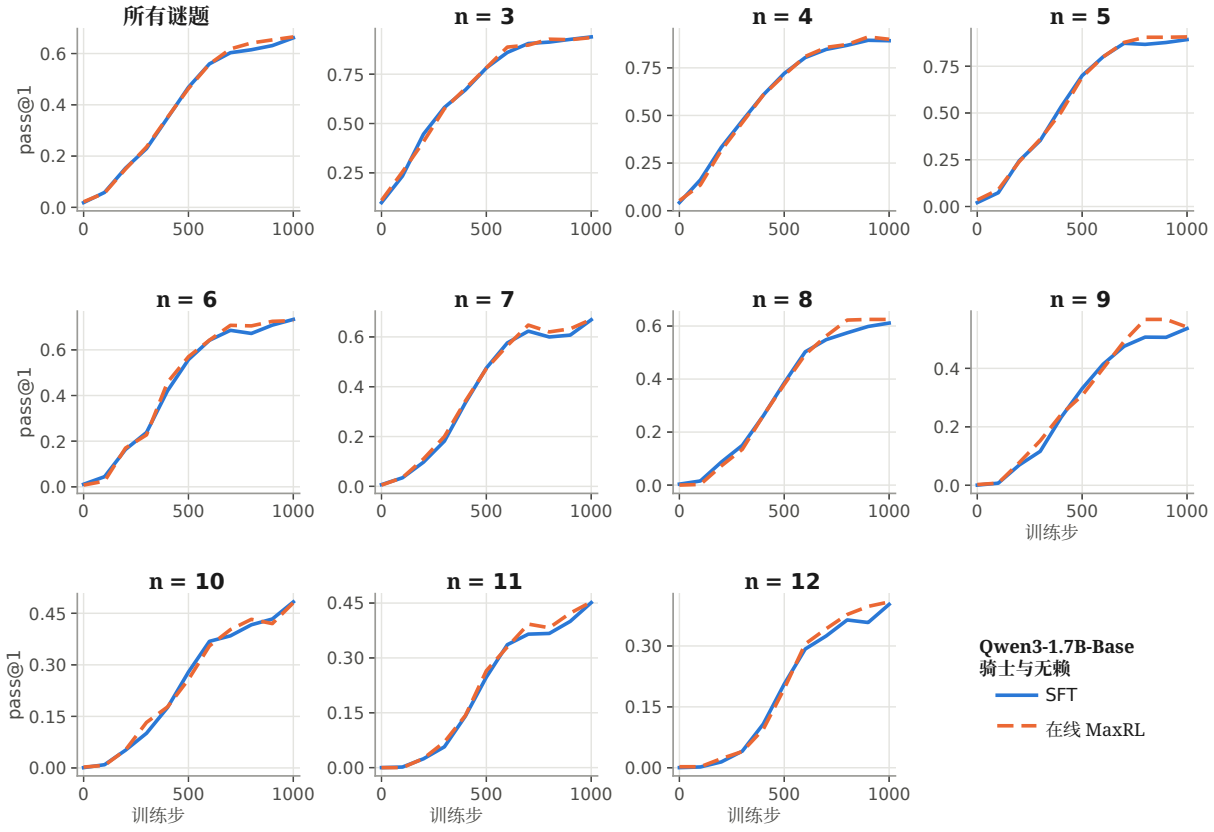


图14：骑士与无赖轨迹重放。在线 MaxRL（橙色虚线）与按时间顺序对其存档轨迹进行监督微调（蓝色）的训练期间 pass@1 表现，涵盖所有谜题以及各个谜题规模 $n = 3, \ldots\ldots, 12$。在线 MaxRL 在训练期间每道谜题使用 4 个样本进行评估，在最终检查点使用 128 个样本；监督微调使用 32 个样本。在训练后期，部分规模较大的谜题上出现了小幅差距。

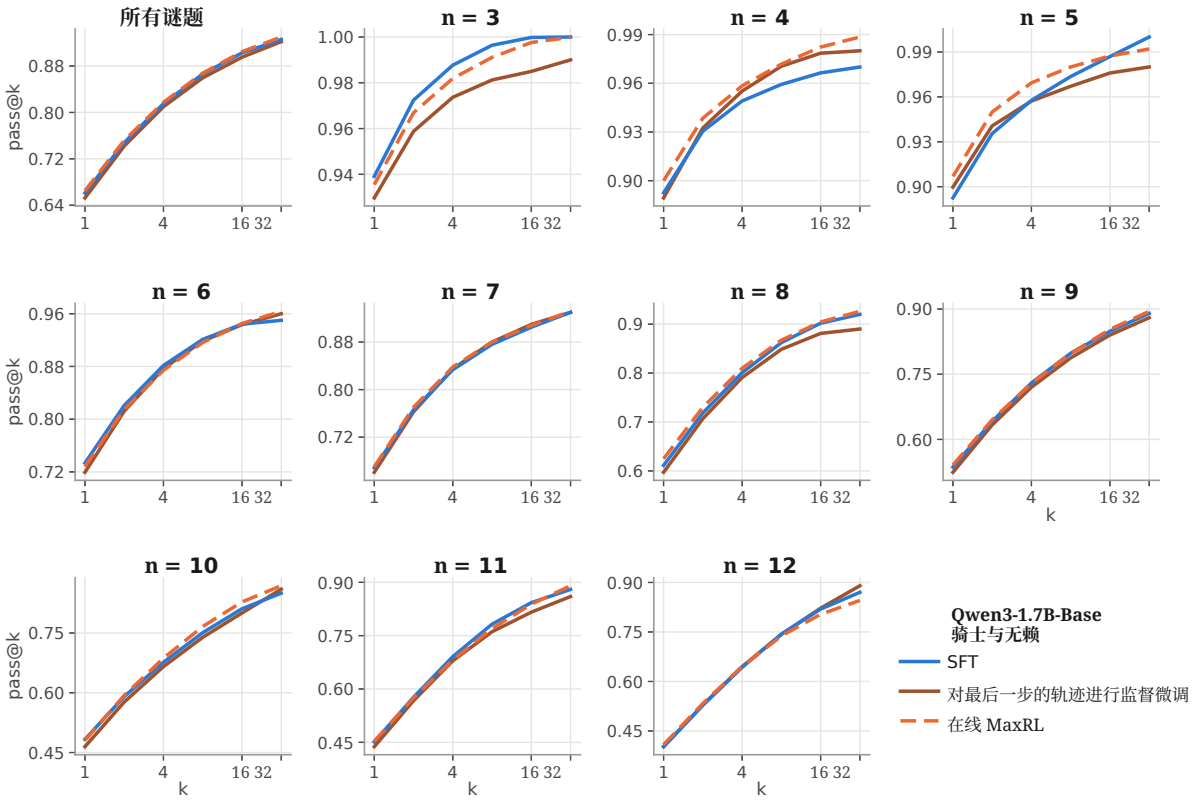


图 15：KnK 端点重建。所有谜题以及每种谜题规模 n 下，在线 MaxRL 的最终检查点 pass@k （橙色、虚线）、对全部 1000 个强化学习步骤的轨迹进行监督微调的结果（蓝色），以及仅对最后一步的轨迹进行监督微调的结果（棕色）。

B.3 最后一步需要多少条轨迹?

对于 Qwen3-1.7B-Base，最后一个强化学习步骤的 4096 条轨迹可以恢复最终检查点（第 4.2 节）。为了解所需的轨迹数量，我们在随机子集上训练基础模型，子集大小为 $n = 16$、32、.....、4096条轨迹。每次 SFT 更新都将整个子集作为单个批次，因此所有运行都进行相同的 100 次更新，唯一的区别在于它们看到的数据量（图7）。

我们将每个子集与使用全部 4096 条轨迹进行的 SFT 进行比较，这与在线 MaxRL 相对应。在 MATH-500 上，与完整批次相比，pass@1 的差距随着 n从0。033 （16 条轨迹时）稳步缩小，0。007 （1024 条轨迹时），直到 2048 条轨迹时才完全弥合（0。708 相比于0。709）。pass@8 的要求较低，从 1024 条轨迹开始便与完整批次相匹配。Minerva 更早达到饱和：从 64 条轨迹开始，pass@1 和 pass@8 与完整批次的差距都保持在0。006 以内。训练更长时间无法弥补数据量不足，而随着更新次数增加，部分曲线还会出现下降（图16）。

因此，在 MATH-500 上，要匹配最终策略，需要使用最后一个步骤的大部分轨迹，约为 4096 条中的 2048 条；而在 Minerva 上则少得多。

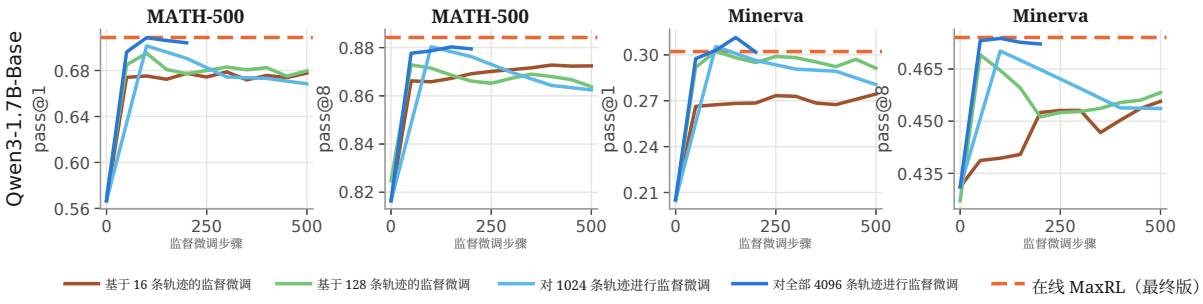


图 16：SFT 与轨迹数量的重复对比。图7 中的运行结果基于 SFT 训练，分别使用最后一步的 16、128、1024 条轨迹以及全部 4096 条轨迹，并与最终的在线 MaxRL 检查点（橙色虚线）进行比较；pass@8 取 3 次评估的平滑结果。4096 条轨迹的运行仅训练了 200 个步骤；其曲线未在该点之后进行外推。

B.4 语言建模之外的任务

关于强化学习资产的论断不应仅建立在语言模型推理之上。因此，我们在两个性质不同的任务上重复进行重放实验：导航 17×17 迷宫，以及使用 ResNet-50（He 等人，2016年）在 ImageNet（Deng 等人，2009年）上进行图像分类（图17）。对于每个任务，我们使用一次在线 MaxRL 运行中所有步骤的轨迹进行离线训练；对于迷宫任务，还使用其最后 k 步的轨迹进行训练，并与最终在线检查点进行比较。在迷宫任务中，监督微调会在进入下一个步骤之前，对每个步骤的轨迹训练 8 次（图18 比较了 1、2、4 和 8 次重复）；一次最后的k 运行会将其中每个 k步骤 $8000/k$ 重复训练若干次，因此每次迷宫运行都会进行约 8000 次监督微调更新。设置详见附录A。

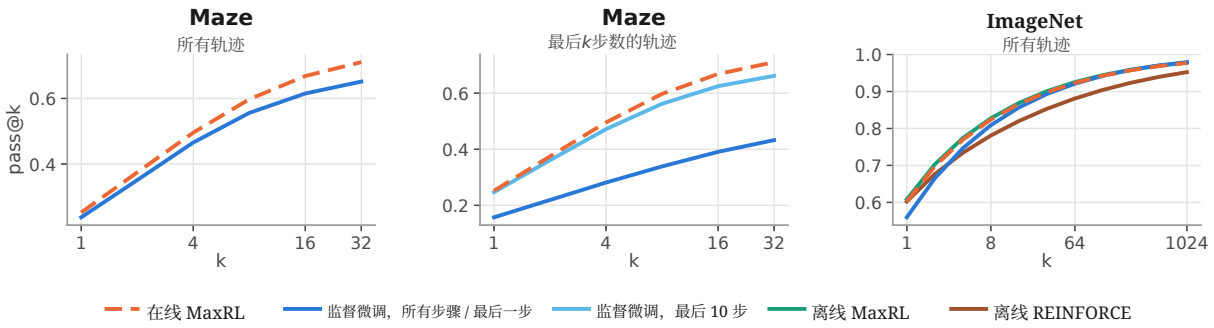


图17：语言任务之外的重建。 在线 MaxRL（橙色虚线）的最终检查点 pass@k，以及在其存档轨迹上进行离线训练所得的结果，涵盖迷宫和 ImageNet 上的 ResNet-50。第一个迷宫面板和 ImageNet 面板使用所有强化学习步骤的轨迹，第二个迷宫面板则仅使用最后 k 步的轨迹。

观察 1

重放也能在迷宫和 ImageNet 上恢复在线 MaxRL；在迷宫任务中，只需一个较短的后期轨迹窗口就足够了。

在迷宫任务中，使用全部 1000 个强化学习步骤的轨迹进行监督微调，达到的 pass@1 为 0。238，而在线结果为 0。252（图 17，左）。只用最后一步的轨迹进行训练，效果较差（0。157），但在相同的监督微调预算下，仅用最后 10 步的轨迹就已达到 0。247。

在 ImageNet 上，离线 MaxRL 与在线 MaxRL 的差距对每个 0。006 都不超过 $k \leq 1024$（图 17，右）。监督微调的起点较低（pass@1 为 0。56，而在线结果为 0。60），但从 0。01 起，差距已不超过 $k = 16$；而离线 REINFORCE 在更大的 k时则落后，最多相差 0。046。

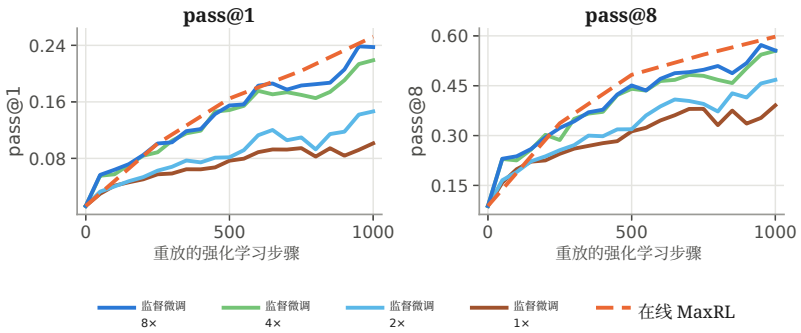


图18：每个已保存的强化学习步骤执行重复 SFT 更新时的迷宫轨迹重建。在进入下一步骤之前，SFT 对 1000 个在线 MaxRL 步骤中每一步的轨迹重放 1、2、4 或 8 次，并与在线 MaxRL（橙色虚线）进行比较。此实验同时改变了优化强度和重放调度，因此无法单独确定相邻策略之间的漂移所起的作用。

C 补充重放顺序结果

图 19 重复了第 4.3 节针对 Qwen3-4B-Base 的打乱顺序实验。图 20 报告了重放顺序运行在更多基准测试上的最终 pass@k 。

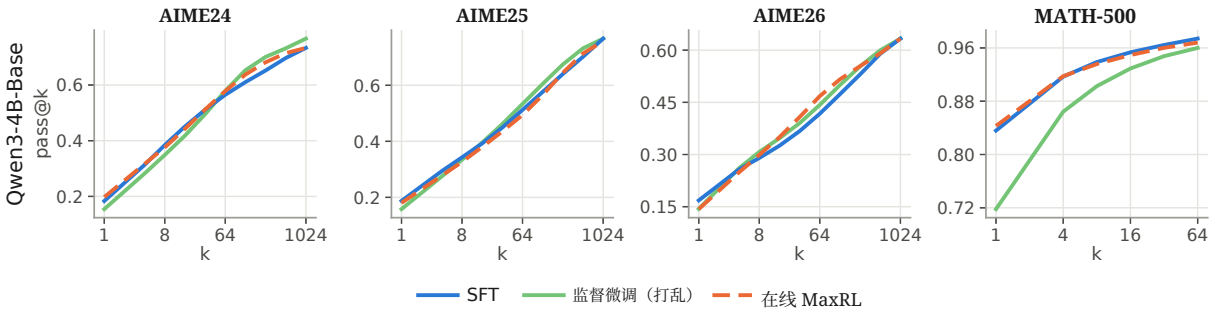


图19：Polaris 上 Qwen3-4B 的打乱顺序监督微调。 在线 MaxRL 的最终检查点 pass@k（橙色虚线），以及对其存档轨迹按时间顺序（蓝色）和打乱顺序（绿色）进行监督微调的结果。

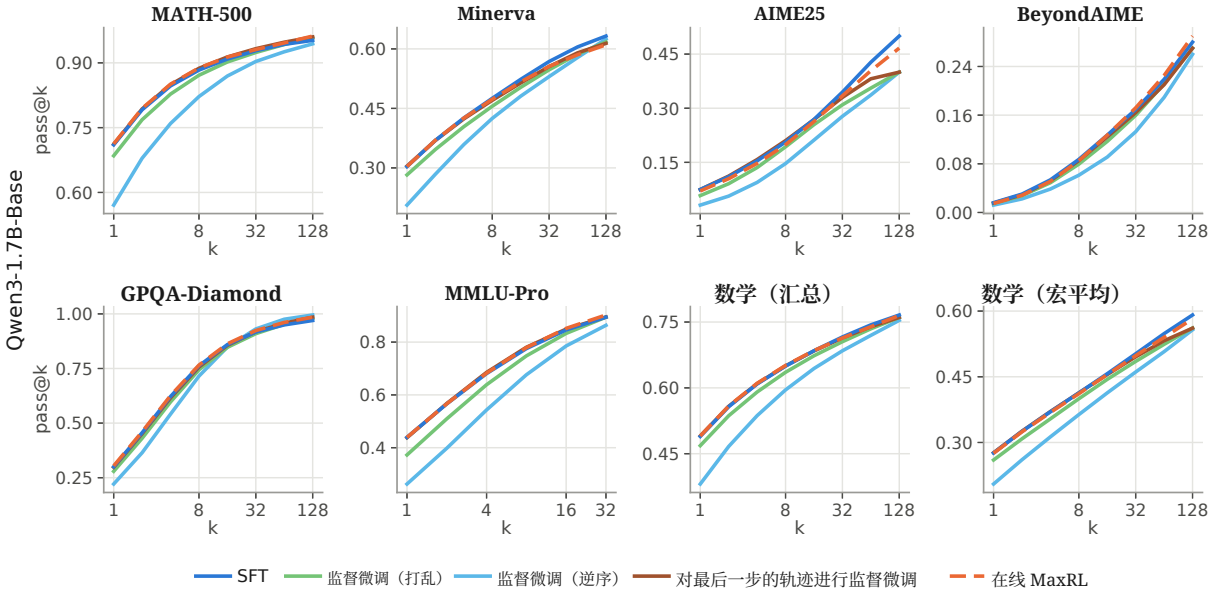


图20：域外评估。 图 k 的重放顺序运行，以及对最后一步的轨迹进行监督微调后，在最终检查点（步骤 1000）上的 pass@8，包括 GPQA-Diamond 和 MMLU-Pro（Wang 等人，2024年）。大多数基准测试每个问题使用 128 个样本进行评估；MMLU-Pro 则在 10% 的子集上每个问题使用 32 个样本进行评估，因此其曲线在 $k = 32$ 处结束。

D 组合细节

D.1 数据空间中的组合

给定 档案 $\mathcal{D}^1$、……、$\mathcal{D}^M$ 个从同一基础模型 π_0 训练而来的 M 专用模型 π^1、……、π^M，数据 a- 空间组合 位置会用所有这些内容训练一个学生模型，

$$\pi_0 \xrightarrow{\text{SFT on } \mathcal{D}^1,\ldots,\mathcal{D}^M} \hat{\pi}^{\text{comp}}. \tag{19}$$

由于组合后的对象是数据，组合过程会明确做出权重合并中隐含的选择。我们将其分解为三个组成部分。

选择。对于一次运行的每个存档 $\mathcal{D}^m = (D_0^m$、……、$D_{T_m-1}^m)$，其中有 T_m 步数，会选取其中一部分步骤。$\mathcal{S}_m \subseteq \{0, \ldots, T_m - 1\}$ 自然的选择包括完整轨迹、最后 K 步，或单个后期批次。第4.2 节表明，合适的选择取决于任务：对于 Polaris，单个后期批次足以实现端点恢复；而 SmolLM2-360M 则需要一个近期批次窗口。

调度。选定的数据通过两种方式呈现给学生模型：一种是联合批次监督微调，每次优化器更新都包含来自所有任务的样本；另一种是交替式监督微调，其中连续的更新会

按轮询顺序分配给不同任务。使用完整轨迹时，也可以按强化学习步骤配对批次，使学生模型并行跟随所有源轨迹。

分配。 每项任务获得训练信号中的一份 w_m，其份额为 $\sum_m w_m = 1$。任务占比必须按 词元衡量，而不能按响应数量或更新次数衡量：在方程（11）中的词元归一化损失下，响应较长的任务每次采样会获得更大的梯度权重。因此，当不同任务的响应长度不同时，相同的响应数量可能导致组成严重失衡。

评估。 组合后的学生模型会针对每位专精模型各自的任务进行评估。令$F_{m,k}(\pi)$表示策略 π 在任务 m的基准测试上的 pass@k 。组合后的学生模型$\hat{}\pi$ 在任务 m 上的保留率为

$$R_{m,k}(\hat{\pi}) = \frac{F_{m,k}(\hat{\pi})}{F_{m,k}(\pi^m)}, \tag{20}$$

因此，$R_{m,k} = 1$ 表示学生模型达到专精模型 π^m的水平，而$R_{m,k} > 1$ 表示其表现超越该专精模型。我们报告平均保留率$\bar{R}_k = \frac{1}{M}\sum_m R_{m,k}$和最差保留率 $\min_m R_{m,k}$。最差情况揭示了是否有任何任务为了其他任务而被牺牲。我们还会展示不同方法及 k取值下各任务的帕累托前沿。

D.2 实验方案与基线方法

所有专用模型都基于同一个基础模型，通过在线强化学习进行训练，其轨迹按照方程（3）进行存档。除非另有说明，专用模型均为使用 MaxRL 训练的 Qwen3-1.7B-Base 模型。我们选择的任务在领域、响应格式和响应长度方面各不相同：数学（Polaris）、逻辑谜题（骑士与无赖、KnK）以及代码（PIE）。我们研究 Polaris 与 KnK 这一对任务。

我们将自己的数据空间组合与三种权重空间基线方法进行比较：权重平均、任务算术和 TIES-Merging（Yadav 等人，2023年）。表6 报告了第4.4节组合实验的完整结果。

表6： ComposingMath 和 KnK 任务：六项数学基准测试、其加权平均汇总（全部数学）以及骑士与无赖任务上的 pass@1/8/32（%）。在六个合并/蒸馏检查点中（即不包括基础模型和两个在线最大强化学习专用模型），**粗体** 表示每列在每个 k中的最佳值，下划线表示次佳值，斜体表示第三佳值；并列值使用相同标记。阴影单元格的数值超过相同 k下对应的在线最大强化学习专用模型（数学列为 Polaris-53K，KnK 列为 KnK）。

检查点	k	MATH500	Minerva	BeyondAIME	AIME24	AIME25	AIME26	全部数学	KnK
基础模型与在线最大强化学习最终检查点									
Qwen3-1.7B-Base	pass@1	57.4	21.1	1.2	4.0	3.1	1.7	36.2	2.1
	pass@8	82.3	43.1	6.4	14.9	15.4	7.1	56.8	12.3
	pass@32	89.2	52.2	14.0	26.4	29.3	10.0	64.6	27.4
在线 MaxRL (Polaris-53K)	pass@1	70.8	30.5	1.6	11.6	7.4	6.4	46.4	3.4
	pass@8	88.5	47.9	8.7	25.9	20.9	18.6	62.5	16.2
	pass@32	93.3	56.2	17.0	33.1	33.8	27.6	69.1	31.3
在线 MaxRL（KnK）	pass@1	67.4	27.4	1.5	8.8	5.4	3.8	43.5	69.7
	pass@8	86.1	45.1	9.0	25.3	18.3	14.0	60.2	88.9
	pass@32	92.6	54.8	20.0	33.3	33.3	23.3	68.5	94.5
权重空间组合									
平均（50% 数学）	pass@1	70.1	29.0	1.4	8.4	6.3	4.6	45.4	30.2
	pass@8	87.7	45.4	8.1	22.7	*19.7*	17.3	61.1	71.3
	pass@32	92.0	52.2	*17.0*	30.0	33.3	*26.7*	67.2	87.6
任务算术	pass@1	70.4	28.2	1.3	*11.2*	6.9	**6.3**	45.5	54.3
	pass@8	**88.9**	45.4	7.4	25.4	**21.0**	**23.4**	62.0	78.6
	pass@32	93.2	53.7	*17.0*	30.0	**36.7**	**33.3**	*68.5*	86.9
TIES 合并（前 20%）	pass@1	71.2	29.1	*1.5*	10.7	6.2	5.5	*46.1*	64.1
	pass@8	88.6	*46.4*	7.9	**27.0**	16.1	*19.1*	61.9	88.4
	pass@32	*92.8*	*55.1*	16.0	30.0	23.3	30.0	68.1	**95.5**
数据空间组合（我们的方法）									
最后一步交替 监督微调（第 200 个训练周期）	pass@1	70.8	**30.7**	**1.7**	11.7	6.9	**6.3**	46.4	68.3
	pass@8	88.2	**47.9**	9.2	*26.0*	18.2	20.0	**62.4**	*88.1*
	pass@32	*92.8*	56.2	*17.0*	**33.3**	*30.0*	**33.3**	68.9	94.1
最后25%的步骤 交替式监督微调	pass@1	**71.4**	*30.1*	1.6	10.6	**7.2**	6.2	**46.5**	*68.0*
	pass@8	*88.5*	47.0	*8.9*	25.2	20.5	18.8	*62.2*	87.9
	pass@32	**93.6**	54.8	18.0	**33.3**	33.3	*26.7*	68.9	93.3
最后50%的步骤 交替式监督微调	pass@1	*71.1*	30.3	**1.7**	**12.5**	*6.8*	*6.0*	**46.5**	**68.8**
	pass@8	88.4	47.0	**10.0**	26.8	19.4	17.2	62.3	**88.7**
	pass@32	92.6	**56.6**	**23.0**	**33.3**	*30.0*	30.0	**69.4**	*93.9*

我们还对数学和 KnK 展开池之间的训练令牌分配进行了消融研究。在存档的后期数据中，KnK 轨迹的长度约为 Polaris 轨迹的两倍，因此混合相同数量的响应会使 KnK 获得显著更高的梯度权重。与此一致，按数量均衡混合会降低数学性能，这促使我们按词元进行分配。我们比较了词元分配方案 w_{Polaris}：w_{KnK} 1:1 和 2:1，以及在训练过程中进行 w_m 动态调整的方案。对于采用最后一步展开、最后 25% 展开和最后 50% 展开的交替式监督微调，我们在表7中报告了 1:1 和 2:1 两种词元分配方案的结果；后一种方案会针对每个 KnK 训练批次，重放两次数学训练批次。我们观察到，两个展开池之间的训练词元分配对恢复没有显著影响，并且两种分配方案都能可靠地恢复专精模型的 pass@k 。

表 7： 使用均衡词元组合数学与KnK任务：六个数学基准测试、其加权平均汇总（全部数学）以及骑士与无赖上的 pass@1/8/32（%）。在六个合并/蒸馏检查点中（即不包括基础模型和两个在线最大强化学习专用模型），**粗体** 表示每个 k列中各最佳 值，下划线表示第二佳值，斜体表示第三佳值；并列值使用相同标记。阴影单元格的值超过同一 k下相应的在线最大强化学习专用模型（数学列为 Polaris-53K，KnK 列为 KnK）。

检查点	k	MATH500	Minerva	BeyondAIME	AIME24	AIME25	AIME26	全部数学	KnK
最后一步									
最后一步交替 SFT	pass@1	70.8	30.7	*1.7*	11.7	*6.9*	6.3	46.4	*68.3*
	pass@8	88.2	47.9	9.2	26.0	18.2	**20.0**	*62.4*	*88.1*
	pass@32	*92.8*	*56.2*	17.0	**33.3**	*30.0*	**33.3**	*68.9*	**94.1**
最后一步交替 监督微调（令牌均衡）	pass@1	70.4	*30.5*	1.8	11.9	6.2	5.7	46.2	68.0
	pass@8	87.9	**48.0**	*9.3*	24.2	18.1	19.4	62.1	87.8
	pass@32	*92.8*	54.8	*18.0*	30.0	23.3	**33.3**	68.3	93.9
最后25%的步骤									
最后25%的步骤 交替式监督微调	pass@1	71.4	30.1	1.6	10.6	7.2	*6.2*	*46.5*	68.0
	pass@8	88.5	*47.0*	8.9	25.2	20.5	*18.8*	62.2	87.9
	pass@32	**93.6**	54.8	*18.0*	**33.3**	33.3	*26.7*	*68.9*	93.3
最后25%的步骤 交替式监督微调 （令牌均衡）	pass@1	*71.3*	*30.5*	**2.2**	*11.8*	6.8	6.0	46.7	67.4
	pass@8	**88.6**	**48.0**	**11.5**	24.6	17.7	17.8	**62.7**	87.7
	pass@32	93.2	**57.4**	22.0	30.0	26.7	*26.7*	**69.5**	**94.1**
最后50%的步骤									
最后50%的步骤 交替式监督微调	pass@1	71.1	30.3	*1.7*	**12.5**	6.8	6.0	*46.5*	**68.8**
	pass@8	*88.4*	*47.0*	10.0	**26.8**	*19.4*	17.2	62.3	**88.7**
	pass@32	92.6	56.6	**23.0**	**33.3**	*30.0*	30.0	69.4	93.9
最后50%的步骤 交替式监督微调 （令牌均衡）	pass@1	**71.5**	**30.9**	1.4	*11.8*	**8.8**	**6.7**	**46.9**	68.5
	pass@8	88.3	47.9	8.1	*25.8*	**25.0**	*18.8*	62.5	88.4
	pass@32	92.4	56.6	17.0	30.0	**36.7**	30.0	68.8	*93.7*