我们距离最佳推理效率有多远？

Jiaxuan Gao[1,2] **Shu Yan**[2,4] **Qixin Tan**[1,2] **Lu Yang**[1,2]
Shusheng Xu[1,2] **Wei Fu**[1,2] **Zhiyu Mei**[1,2] **Kaifeng Lyu**[3] **Yi Wu**[1,2,†]
[1] IIIS, Tsinghua University [2] Ant Research
[3] Simons Institute, UC Berkeley [4] Nanjing University
{ samjia2000, jxwuyi } @gmail.com

Abstract

大型推理模型（LRMs）通过延伸的链式推理（CoT）展现出非凡的问题解决能力，但往往会产生成过于冗长和重复的推理痕迹。这种低效性导致高推理成本并限制实际部署。虽然现有的微调方法旨在提高推理效率，但由于评估不一致，评估它们的效率增益仍然具有挑战性。在这项工作中，我们引入了推理效率前沿，即通过微调基础 LRMs（DeepSeek-R1-Distill-Qwen-1.5B/7B 和 Qwen3-4B/8B ）在多种方法和训练配置中获取的经验上限。基于这些前沿，我们提出了推理效率差距（REG），这是一种统一的指标，用于量化任何微调 L与这些前沿之间的偏差。在具有挑战性的数学基准上的系统评估揭露了当前方法的显著差距：它们要么牺牲了准确度以缩短长度，要么 在受限的令牌预算下仍然效率低下。 。为减少效率差距，我们提出了 REO-RL ，即 强化学习算法类别 ，通过针对稀疏的 token 预算集合。利用在策略性选定预算上的数值积分， REO-RL 以低误差近似完整的效率目标，使用一小组 token 预算。通过系统化的基准测试，我们证明了我们新的效率指标 REG 有效捕捉了准确度与长度的权衡，低 REG 方法在保持准确度的同时缩短了长度。我们的方法， REO-RL ，显著提高了推理效率，在评估的所有 L中降低了 REG，并且在 16K token 预算下匹配 Qwen3-4B/8B 的效率前沿，几乎没有准确度损失。消融研究确认了我们指数型 token 预算策略的有效性。最后，我们的研究结果强调，使 LRMs 完美地对齐于效率前沿的微调仍然是一个开放的挑战。我们将发布代码、数据和模型。[1] 。

1 引言

大型推理模型（LRMs）最近作为一类强大的模型出现，能够解决需要高级推理的复杂任务。前沿的 LRMs 如 OpenAI o1 [24] 和 DeepSeek R1 [8] 在包括数学推理和竞赛编程的广泛任务中获得了卓越的性能。这个成功背后的一个主要因素是它们能够通过扩展的思维链（CoT）过程来执行深入的、多步骤的推理。这些推理轨迹通常包括复杂的操作，如反思、验证和探索，均在一次推理过程中完成。

然而，长策略推理的强大能力是有代价的。LRMs 经常生成过于冗长和冗余的推理轨迹，这一现象被称为过度思考问题 [39, 33] 。最近的研究 [4, 33] 表明，即使是像 “2 + 3 = ?” 这样简单的问题，其结果可能会长达 900 个标记。这种冗余在推理时间上带来了显著的成本，并限制了实际部署。已经提出了几种微调方法以提高推理效率， 具有代表性的例子是使用 RL 进行长度缩减 [34, 18, 1, 2, 40, 31, 27, 38, 29, 10] ， 以及训练 LRM 自适应地选择思考或不思考模式的混合推理方法 [12, 17, 45, 15] 。然而，由于实验设置不一致，包括模型变化、评估基准和混杂的性能指标，这些方法的比较仍然很困难。目前尚不清楚现有方法在长度和准确性之间取得最佳平衡的程度如何。

在这项工作中，我们研究一个关键问题： 目前的方法距离达到最佳推理效率有多远？为了解答这个问题，我们使用一组 LRMs 和 深度搜索-蒸馏-Qwen-1.5B/7B 和 Qwen3-4B/8B ，在

[1] https://github.com/samjia2000/Optimal-Reasoning-Efficiency

Preprint. Under review.

一系列具有挑战性的数学推理基准上进行了全面的实证研究。我们引入了推理效率边界的概念，该概念来源于通过 各种 类型的算法和多样的训练配置来微调基础 LRMs。这些推理效率边界代表了在每个 token 预算下当前方法可实现的最佳奖励，提供了关于最优效率的实际下限。通过将当前方法与这些边界进行比较，我们发现了一个巨大的差距。现有方法往往在两方面之一方面不够理想，要么（1）为了缩短响应时间而牺牲了准确性，要么（2）达到高准确度的方法会消耗比达到中等准确度所需的更多的 tokens。为了量化这一差距，我们提出了推理效率差距（REG），这是一种统一指标，通过测量 LRM 的预算-精度曲线与效率边界之间的面积来捕捉准确性和响应长度。 具体来说，REG 被计算为所有标记预算中，效率前沿与 LRM 之间的准确性差距。REG 还量化所有准确性中浪费的标记的平均数量。 更重要的是，REG 提供了实用的见解，显示了还剩下多少改进空间。

我们进一步询问：如何对 LRM 进行微调以最大限度地减少这种效率差距？一个自然的方法是在强化学习训练期间优化所有可能的 token 预算的奖励。然而，这种方法导致了高昂的训练成本，因为需要评估所有 token 预算的奖励。为了克服这种密集奖励方法的低效性，我们引入了 Reasoning Efficiency Optimization with Reinforcement Learning （ REO-RL ）， 一种新颖的 RL 算法家族，用于提高 LRM 的推理效率。 。REO-RL 的关键见解是，可以通过对一小组有代表性的 token 预算进行数值积分来很好地近似所有 token 预算的总奖励。 我们介绍了 REO-RL 的两个安装方式，包括利用基于预言机的贪心方法根据估计的推理效率前沿来选择 token 预算的 REO-RL （Oracle），以及选择指数间隔 token 预算的 REO-RL （Exp）。

我们的消融研究进一步验证了指数代币预算选择策略的成功以及少量代币预算的有效近似。

2 相关工作

先前的研究表明，LRM 常常存在冗余推理现象。即使对于非常简单的问题，Frontier LRM 经常会生成跨越数千个标记的冗长响应。这种推理过程中的冗余在推理成本中带来了显著的开销。因此，提出了几项工作来使 LRM 推理更加简洁。[34, 18, 1, 2, 40, 31, 27, 38, 29, 10, 26] 探索了具有长度奖励设计的 RL 训练，主要集中在减少推理长度。一些工作应用 SFT 在具有可变长度推理轨迹的数据集上微调 LRM，以引出简洁的推理或可调节的长度控制。还有一些通过奖励模型引导解码、不确定性基础的动态推理和基于置信度的方法来提高推理效率。在这项工作中，我们研究了 LRM 的最佳推理效率，并集中于基于训练的方法以增强 LRM 推理效率。我们的方法旨在提高 LRM 在不同标记预算下的准确性，而不显式地激励更短的响应。

基于强化学习的 LRM 推理。 强化学习是激发和增强 LRM 推理能力的核心技术。前沿的 LRM，包括 OpenAI o1 [24] 和 DeepSeek R1 [8]，已表明在基础 LLM 上应用" 零 RL" 可以有效激发使用长 CoT 进行复杂推理的能力。一系列工作涌现出来，重点是从数据 [20, 28, 9, 14, 36]、算法 [8, 9, 20, 42, 44] 和训练框架 [32, 19, 28] 的角度提高 LRM 的 RL 训练效率。许多工作成功地在广泛的重推理领域应用零 RL 训练，包括多模态 [30, 46]、医学 [41, 3] 和金融 [16]。最近的工作也通过鼓励简洁推理与 RL [34, 18, 1, 2, 40, 31, 27, 38, 29, 10] 来探索效率增强。在这项工作中，我们着重于通过 RL 增强推理效率。

3 预备知识

LRM 推理。 在这项工作中，我们专注于数学推理任务。给定一个问题 x ，LRM 策略的目标是生成一个响应 y ，其中包含逐步推理以得出正确答案。我们假设可以访问一个验证器 $\mathcal{R}(x, y)$ ，它评估在给定问题 x 的情况下解决方案 y 的正确性。实际上，这样的验证器是通过匹配真实答案和模型生成的答案来实现的。LRM 是一个以 θ 参数化的策略 π_θ ，它以自回归方式生成推理标记序列。给定一个问题分布 $\mathcal{D}$ ，LRM 的目标是最大化产生正确响应的概率，

$$J(\mathcal{D}, \theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[\mathcal{R}(x, y)] \tag{1}$$

其中 θ 位于参数空间 Θ ，响应长度 $|y|$ 被限制为最大长度 L_{max} 。在实践中，θ 通常通过在基础 LRM 上应用如 RL 的微调方法获得。因此，我们假设存在一个基础 LRM θ_{base} ，并假设 Θ 为通过使用任何算法微调 θ_{base} 而可以获得的所有 LRM 的集合。

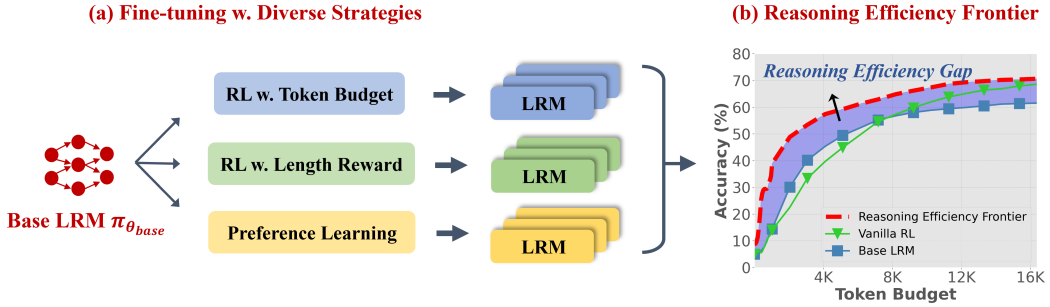


Figure 1: 推理效率前沿 & 推理效率差距。(a) 从一个基础 LRM $\pi_{\theta_{\text{base}}}$ 开始，我们应用多样的微调策略以获得大量的 LRMs。(b) 然后，我们计算在不同的 token 预算下可达到的最佳准确性，以获得推理效率前沿（方程 5）。推理效率差距 (REG) 是一个统一的指标，通过测量一个 LRM 的预算-准确率曲线与效率前沿之间的面积来同时捕捉准确性和长度。

4 理解高效推理的极限

4.1 限定代币推理中的最优性定义

为了评估最佳推理效率，我们必须评估固定令牌预算 L 下的 LRM π_θ 的表现。然而，仅仅在 L 个令牌后截断输出可能导致没有答案的不完整响应。为了解决这个问题，我们定义了一种沿用以往工作的回退机制 [22, 6]。如果推理轨迹 $y \sim \pi_\theta(\cdot|x)$ 超过 L 个令牌，模型将被提示从截断的轨迹 $y_{:L}$ 直接生成最终答案 a。

其中 $y_{:L}$ 表示推理轨迹的前 L 个标记，$\text{ExtractAnswer}(x, y)$ 从完整的轨迹中提取最终答案。此方法允许在不同预算之间进行一致评估，尽管在截断情况下会引入一些额外的标记成本。为简洁起见，我们使用 $r(x, y_{:L}; \theta)$ 来表示对响应 $y_{:L}$ 的 π_θ 的预期收益，即

$$r(x, y_{:L}; \theta) = \mathcal{R}(x, \text{Answer}(\pi_\theta, x, y_{:L})) \tag{2}$$

我们定义了模型 π_θ 在问答分布 $\mathcal{D}$ 上的长度受限奖励为当模型受到令牌预算 L 时获得的期望奖励，

$$J(\mathcal{D}, \theta, L) = \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x, y_{:L}; \theta)]] \tag{3}$$

然后，长度约束的最佳奖励捕捉了任何模型在相同预算下在参数空间 Θ 中能够达到的最佳可能奖励，

$$J_{\text{optimal}}(\mathcal{D}, \Theta, L) = \max_{\theta\in\Theta} J(\mathcal{D}, \theta, L) \tag{4}$$

4.2 推理前沿的实证估计

我们的目标是刻画最佳推理效率，记为 $J_{\text{optimal}}(\mathcal{D}, \Theta, L)$，其反映了在所有可能的模型参数 $\theta \in \Theta$ 下，在任何令牌预算 L 下可实现的最佳奖励。然而，在实际中精确计算这个最优边界是不可行的，因为这需要穷尽所有算法和训练配置进行探索。相反，我们通过使用现有方法微调多样化的一组模型来构建一个经验推理效率边界。令 $\hat{\Theta} = \{\theta_1, \ldots, \theta_m\} \subseteq \Theta$ 表示来自 m 微调模型的参数集合。基于这些，我们定义经验推理效率边界为，

Definition 4.1: Reasoning Efficiency Frontier

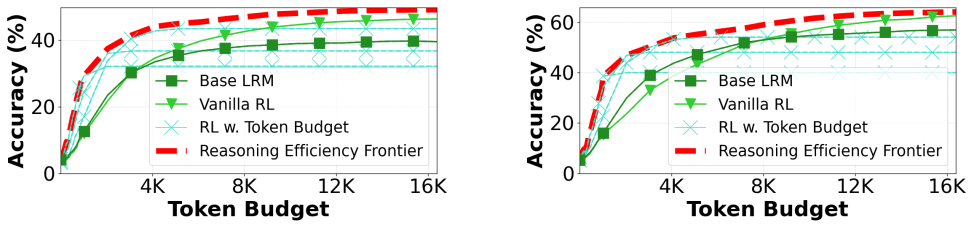
Given a parameter space Θ , a set of model parameters $\hat{\Theta} = \{\theta_1, \cdots, \theta_m\}$ and a question distribution $\mathcal{D}$, we define the reasoning efficiency frontier $\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$ as

$$\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L) = \max_{\theta\in\{\theta_1,\cdots,\theta_m\}} J(\mathcal{D}_t, \theta, L) \quad \forall L \in [1, L_{\text{max}}] \tag{5}$$

注意，$\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$ 作为最优边界的下界，因为 $\hat{\Theta}$ 是 Θ 的子集，

$$\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L) \leq J_{\text{optimal}}(\mathcal{D}, \Theta, L)$$

为了在方程 5 中获得对最优边界的近似，我们使用多种训练策略对模型进行微调，



(a) DeepSeek-R1-Distill-Qwen-1.5B (b) DeepSeek-R1-Distill-Qwen-7B

(c) Qwen3-4B (d) Qwen3-8B

Figure 2: DeepSeek-R1-Distill-Qwen-1.5B/7B 和 Qwen3-4B/8B 的推理效率边界。比较这两类模型，Qwen3 模型具有更高的准确性，并且比 DeepSeek-R1-Distill-Qwen 模型更接近效率边界。

- 在线 RL 与令牌预算：我们进行在线 RL 训练，令牌预算从 512 到 32k 不等。有效地用令牌预算对 LRM 进行微调，能够强制 LRM 在有限的令牌下进行推理 [37, 10] 。
- 在线强化学习与长度奖励。我们测试了多种奖励设计，以促进简洁而准确的推理，包括长度协调奖励 [18] 和长度组归一化奖励 [2] 。
- 偏好学习。我们应用 SimPO [21] ，结合通过 TOPS [39] 和 DAST [31] 等方法构建的偏好数据集。例如，我们对比简短的正确回答和较长的回答以促进简洁性 [23] 。

特别是在 DeepSeek-R1-Distill-Qwen-1.5B/7B 中，我们从经过 RL 微调的版本开始，并使用上述策略进一步微调它们。评估是在 一组具有挑战性的数学推理基准。 上进行的。更多训练和评估 的细节可以在 Sec. 6.1 和附录 C 中找到。

基于微调后的模型，我们能够构建 基础 LRMs，如图 2 所示。有趣的是，我们发现 DeepSeek-R1-Distill-Qwen 模型无论效率均远离前沿，存在明显的效率差距。相比之下，Qwen3 模型达到与效率前沿相似的准确性，并且更接近效率前沿，从而为推理效率的提升留出了有限的空间。 的推理效率前沿

给定估计的推理效率边界 $\hat{J}_{\text{optimal}}(\mathcal{D}, \Theta, L)$ ，我们引入推理效率差距（REG）这一指标，以量化任何 LRM 与达到最佳推理效率之间的距离。

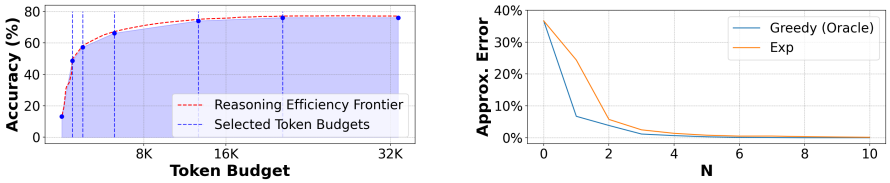
Definition 4.2: Reasoning Efficiency Gap (REG)

Given any LRM π_θ and the estimated reasoning efficiency frontier $\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$, we define Reasoning Efficiency Gap as,

$$d_{\text{REG}}(\theta, \mathcal{D}, \hat{\Theta}) = \sum_{L=1}^{L_{\max}} \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L) - J(\mathcal{D}, \theta, L) \tag{6}$$

注意，当 $\hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$ 和 $J(\mathcal{D}, \theta, L)$ 均为非递减时，REG 可以等效地表示所有准确性中浪费的标记的平均数。

REG 的优点。 作为一个单一指标，REG 统一了准确性和长度方面。同时，REG 定量评估了一个 LRM 与效率前沿之间推理效率的差距。大多数现有评估指标只关注准确性和长度方面中的一个。例如，Pu et al. [25] 提出通过计算最长和最短响应之间的长度差异来评估推理冗余，而不考虑生成响应的正确性。我们还注意到 Chen et al. [4] 引入了一个结果效率指标，



(a) (b)

Figure 3: 在 REO-RL 中的令牌预算选择。(a) 选择的令牌预算大致遵循指数模式。(b) 无论是 oracle 贪婪方法还是指数方法都能在少数令牌预算下实现较低的逼近误差。

测量首次达到正确答案的位置与正确响应的整体长度之间的比率。尽管这个结果效率指标同时考虑了准确性和长度，但如果 LRM 仅在响应末尾产生答案，则有被破解的风险。

5 方法论

5.1 通过优化长度约束奖励提升推理效率

为了最小化推理效率差距，一个直接的方法是优化在所有令牌预算下的长度约束奖励，以增强 $\pi_{\theta_{\text{base}}}$ 的推理效率，从而得到效率目标，

$$\mathscr{L}_{\text{Efficiency}}(\theta, \mathcal{D}) = \sum_{L=1}^{L_{\max}} J(\mathcal{D}, \theta, L) \tag{7}$$

，其中 $L_{\max}$ 是最大生成长度。然而，直接优化方程 7 在计算上是不切实际的。为每个预算 $L \in [1, L_{\max}]$ 评估 $J(\mathcal{D}, \theta, L)$ 需要单独的推断运行，以评估截断响应，如在 Sec. 4.1 中讨论的。因此，每个训练示例都需要多达 $L_{\max}$ 次额外的 LRM 生成，导致计算时间和内存使用增加。

5.2 Reasoning Efficiency Optimization with Reinforcement Learning

我们引入了 Reasoning Efficiency Optimization with Reinforcement Learning （ REO-RL ），这是一类有效的训练算法，它优化了方程 7 中目标的近似。在 REO-RL 中，我们没有在所有的 token 预算中优化长度受限的奖励，而是用一小部分选择的 token 预算 $L_1, \cdots, L_N$ 来近似目标，以确保较高的训练效率。具体来说，遵循数值积分中的梯形法则，我们可以用以下方式近似方程 7 中的目标，

$$\sum_{L=1}^{L_{\max}} J(\mathcal{D}, \theta, L) \approx \sum_{i=1}^{N} \frac{L_{i+1} - L_{i-1}}{2} J(\mathcal{D}, \theta, L_i) + \frac{L_1}{2} \cdot J(\mathcal{D}, \theta, 0) + \frac{L_{\max} - L_N}{2} \cdot J(\mathcal{D}, \theta, L_{\max}) \tag{8}$$

$$= f(\mathcal{D}, \theta, \{L_1, \cdots, L_N\}) \tag{9}$$

其中我们假设 $L_0 = 0$ 、$L_{N+1} = L_{\max}$ 和 $f(\mathcal{D}, \theta, \{L_1, \cdots, L_N\})$ 表示具有令牌预算 $L_1, \cdots, L_N$ 的近似目标。随着选择的令牌预算 N 的增加，$f(\mathcal{D}, \theta, \{L_1, \cdots, L_N\})$ 将越来越接近于方程 7 中的目标。

近似的目标 $f(\mathcal{D}, \theta, \{L_1, \cdots, L_N\})$ 可以等效地表示为具有密集奖励的 RL 目标，从而得到目标 REO-RL ，其中 $c_i = \frac{L_{i+1} - L_{i-1}}{2}$ 对应 $1 \le i \le N$ ，并且 $c_{N+1} = \frac{L_{\max} - L_N}{2}$ 是 i -th 令牌预算的系数，遵循方程 9 。

不同的标记预算集合 $L_1, \cdots, L_N$ 会导致等式 9 中产生不同的近似误差。理想情况下，所选标记预算引起的近似误差应该较低，以确保 REO-RL 与等式 5 中的原始目标一致。

注意，方程 5 中的原始目标受第 4.2 节中的理论推理效率边界约束，即 $\sum_{L=1}^{L_{\max}} J(\mathcal{D}, \theta, L) \le \sum_{L=1}^{L_{\max}} J_{\text{optimal}}(\mathcal{D}, \Theta, L)$ 。一个自然的想法是基于估计的推理效率边界确定最佳的代币预算选择方案。具体来说，对于任何代币预算集合 $L_1, L_2, \cdots, L_N$ ，可以通过类似方程 9 的方式来

逼近 $\sum_{L=1}^{L_{\max}} \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$ ，其中

$$f_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, \{L_1, \cdots, L_N\}) = \sum_{i=1}^{N} \frac{L_{i+1} - L_{i-1}}{2} \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L_i) + \frac{L_1}{2} \cdot \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, 0) \tag{10}$$

$$+ \frac{L_{\max} - L_N}{2} \cdot \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L_{\max}) \tag{11}$$

中 $f_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, \{L_1, \cdots, L_N\})$ 表示在给定代币预算 $L_1, \cdots, L_N$ 的情况下，$\sum_{L=1}^{L_{\max}} \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L)$ 的逼近值。

我们采用一种贪婪的方法，该方法迭代地选择近似误差最低的令牌预算。需要注意的是，由于推理效率的边界应该事先已知，这是一个 oracle 方法。这种贪婪选择方法导致了一个 oracle 算法， REO-RL （Oracle）。在 REO-RL （Oracle）中，令牌预算 $L_1, \cdots, L_N$ 根据以下方式选择，

$$L_i = \arg\min_{L'} \mid f_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, \{L_1, \cdots, L_{i-1}, L'\}) - \sum_{L=1}^{L_{\max}} \hat{J}_{\text{optimal}}(\mathcal{D}, \hat{\Theta}, L) \mid$$

。这种 oracle 贪婪方法可以产生一组具有低近似误差的令牌预算，如图 3(a) 所示。如图 3(b) 所示，随着更多令牌预算的增加，近似误差逐渐降低。值得注意的是，近似误差可能低于 1% 与 $N \geq 5$ 。

REO-RL（指数函数）。 然而，当推理效率前沿未知时，应用 REO-RL（预言机）是不可行的。我们观察到，通过贪婪选择方法选择的 token 预算大致遵循如图 3(a) 所示的指数间隔模式。因此，我们建议采用指数间隔方案进行 token 预算选择，提出算法 REO-RL （Exp），选择一组指数间隔的 token 预算，

$$L_i = L_{\min} \cdot (L_{\max}/L_{\min})^{\frac{i-1}{N}}$$

，其中 $L_{\min}/L_{\max}$ 是最小/最大 token 预算。

6 实验

6.1 实验设置

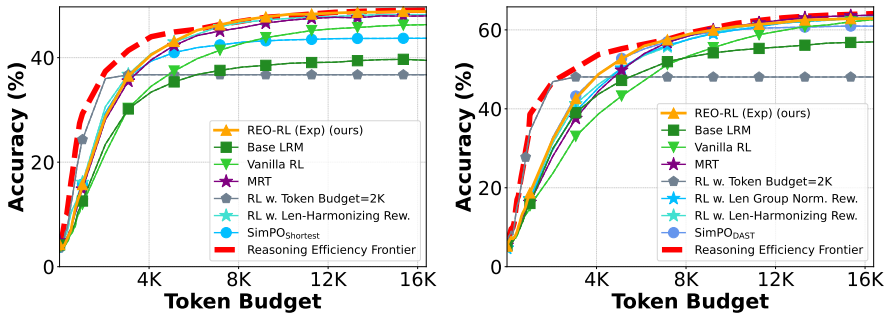
模型、数据集 & 指标。 我们使用 DeepSeek-R1-Distill-Qwen-1.5B & 7B 和 Qwen3-4B & 8B 作为基础 LRM。对于训练，我们采用由 DeepScaleR [20] 和 AReaL [28] 提供的 135k 个问题组合而成的训练数据混合物。对于评估，我们使用 几个具有挑战性的数学基准测试：AMC 2023，AIME 2024，& 2025，以及 Minerva Math [13] 。 我们报告在温度 $T = 0.6$ 和 top_p $= 0.95$ 下生成的 32 个响应的平均准确率，最大长度为 $L_{\max} = 32K$ 。主要结果是在三个基准上平均获得的。评估 REG 时，我们使用更小的长度 $L_{\max} = 16K$ 。

对于 REO-RL ，除了 REO-RL （Exp）和 REO-RL （Oracle），我们还进一步考虑了 REO-RL （Q-Spec），这是一种 REO-RL 的变体，旨在优化特定问题的最小推理标记预算 [2] 中的奖励。

- 基于长度奖励的在线强化学习：我们研究了基于长度奖励的在线强化学习，包括长度协调奖励 [18] 和长度组归一化奖励 [2] 。我们与元强化微调（MRT） [27] 进行比较，它通过最小化单步的遗憾来提高推理效率。此外，我们还采用了有硬性令牌预算的在线强化学习作为基线，这在最近的工作中也被采用 [10, 37] 。 我们注意到最近的一项工作 BRPO [26] 在线性分布的代币预算下优化奖励，这类似于在消融研究（Sec. 6.4 ）中研究的 REO-RL 变体。 [3]
- 混合推理。 我们使用 HGPO [12] 作为具有代表性的混合推理基线，该基线训练 LRM 自适应地选择思考模式和非思考模式之间。最近也有一些研究分享了类似的想法，即在非思考模式达到具有竞争力的准确度时，推动非思考模式 [5, 45, 35] 。

[2] 关于 REO-RL (Q-Spec) 的更多细节，请参阅附录 E

[3] 附录中提供了有关 BRPO 和 REO-RL 之间的更详细讨论。 A



(a) DeepSeek-R1-Distill-Qwen-1.5B (b) DeepSeek-R1-Distill-Qwen-7B

(c) Qwen3-4B (d) Qwen3-8B

Figure 4: REO-RL (Exp) 与代表性的基线方法 在 16K 标记预算内 的性能比较。 REO-RL (Exp) 在推理效率方面优于基线。在 DeepSeek-R1-Distill-Qwen 模型中，在紧张的 token 限制下仍然存在较大的性能差距。对于 Qwen3 模型，尽管 REO-RL 在 16K token 预算下匹配效率前沿，但准确率损失仍然存在（见 Tab. 1 ）。结果平均在 AMC 2023, AIME 2024 & 2025 和 Minerva Math 上。

- 监督微调：对于训练数据中的每个问题，我们生成多个响应并在最短的正确响应上执行 SFT。数据生成采用两种策略。第一种策略是利用问题作为输入直接生成 [23] 。第二种策略遵循 TOPS，通过不同层次的推理努力来提示 LRM [39] 。
- 偏好学习：我们在各种偏好数据集上应用 SimPO [21] 。在 $SimPO_{shortest}$ 中，我们使用最短的正确回答和最长的回答作为偏好对。我们还遵循 TOPS [39] 和 DAST [31] 构建偏好数据集。

我们也尝试了基于增强学习的长度控制方法，例如 [1, 37] ，但发现这些方法仅在低代币预算下实现了成功的长度控制，并与增强学习结合代币预算产生了相似的长度-准确性折衷。关于基线和我们其他研究的更多细节可以在附录中找到。 D 。

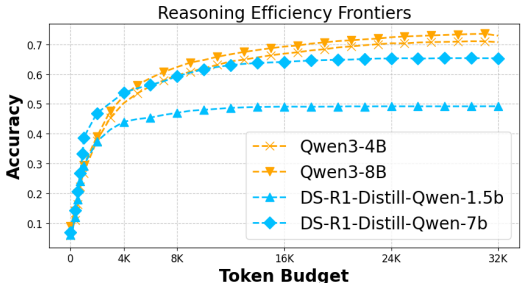


Figure 5: DeepSeek-R1-Distill-Qwen-1.5B/7B 和 Qwen3-4B/8B 的推理效率前沿。Qwen3 模型获得了更高的准确性。有趣的是，在 4K 令牌预算下，DeepSeek-R1-Distill-Qwen-7B 比 Qwen3 模型的推理效率更高。

对于 REO-RL 和所有基线，我们在 AReaL 框架 [28] 的基础上实现了所有方法。对于 DeepSeek-R1-Distill-Qwen-1.5B/7B，我们对相应的 RL 训练版本进行微调。具体而言，我们

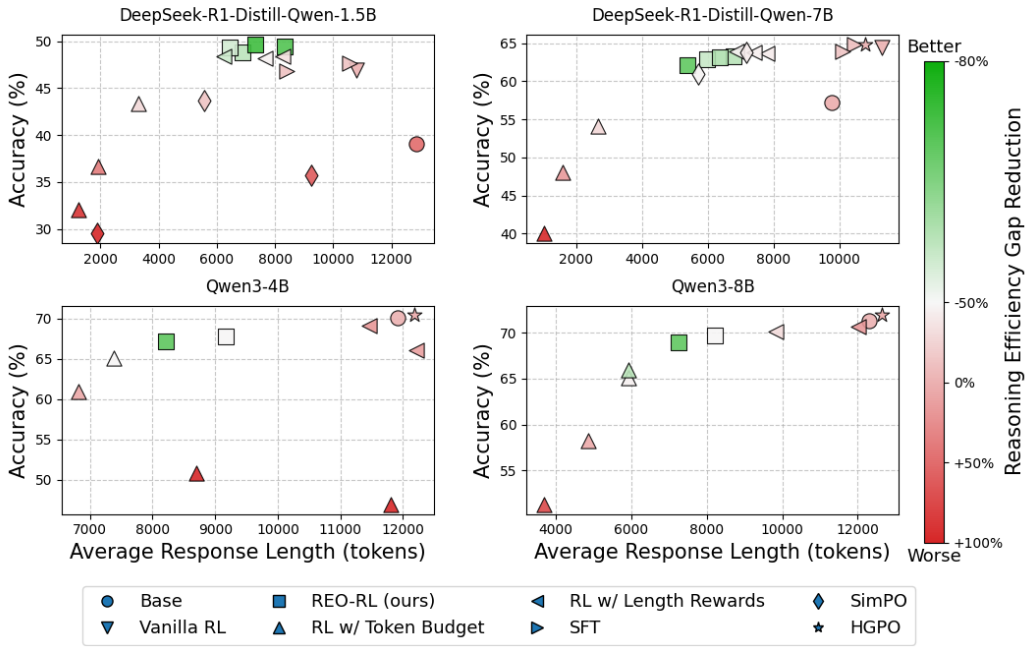


Figure 6: REG 有效地捕捉了准确性与响应长度之间的权衡。要实现低 REG，需要具有竞争力的准确性和较短的响应长度。通过最小化效率差距， REO-RL 在推理效率方面优于基准。

使用 AReaL-Boba-RL-1.5B [28] 和 SkyWork-OR1-Math-7B [9] 作为 1.5B 和 7B 实验进一步微调的起始点。对于 对于 Qwen3 模型，我们直接在基础 LRMs 上应用 RL 训练。（Exp）和 REO-RL （Oracle），我们使用 $N = 5$ 的令牌预算，因为 $N = 5$ 已经如图 3(b) 所示获得了足够准确的近似。更多训练细节，请参阅附录。 C 。

6.2 主要结果

图 4 展示了使用 REO-RL (Exp) 微调的 LRM 的预算-准确率曲线，以及一系列具有代表性的基线方法。[4] 图 6 进一步说明了不同方法类别的准确性、响应长度和 REG 减少。 全部定量结果详见表 1 。我们强调几个关键结论，

现有方法和前沿之间存在根本性的差距。如图 4 所示，在给定充足的令牌预算的情况下，几个基线可以接近甚至达到推理效率的前沿。然而，为了达到中等水平的准确性，它们可能需要显著多于效率前沿的令牌数量。另一方面，一些方法，如具有令牌预算的 RL，在严格的令牌限制下表现良好，但总体准确性明显较低。 值得注意的是，我们提出的方法 REO-RL 在低 token 预算情况下可以匹配 Qwen3 模型的效率前沿，但整体准确率仍有轻微下降，如表 1 所示。 我们的基准测试结果表明，优化基础 LRM 以精确匹配推理效率前沿仍然是一个开放性问题。

REG 有效地捕捉了准确性和响应长度之间的权衡。如图 6 所示，实现低 REG 需要在高准确性与简洁输出之间取得平衡。例如， REO-RL 在生成紧凑响应的同时保持了较强的准确性，导致 REG 显著降低。相比之下，仅优化其中一个方面的方法，如以牺牲准确性为代价来最小化长度的方法，无法实现低 REG。尽管 RL with Token Budgets 产生了较短的响应，但准确性却遭受了重大损失。

最后， REO-RL 在推理效率方面始终优于基线。与基线相比， REO-RL 的结果更接近推理效率的前沿，表现出在 REG 方面的更大减少。 REO-RL 在生成较短输出的同时保持高精度。 REO-RL （Exp）在所有基础 LRMs 中始终将 REG 减少至少 50 %。对于 Qwen3 模型，应用基于长度的奖励的 RL 基线无法始终如一地缩小效率差距。事实上，我们发现 RL 与长度协调的奖励甚至使 Qwen3-4B/8B 模型的效率差距扩大。

[4]为了更清晰地可视化，我们仅包含一组具有代表性的方法，并绘制它们在有限的令牌预算下的性能。

Method	AIME 2024			AIME 2025			AMC 2023			Minerva Math			Average		
	Acc (%) ↑	Len ↓	REG ↓	Acc (%) ↑	Len ↓	REG ↓	Acc (%) ↑	Len ↓	REG ↓	Acc (%) ↑	Len ↓	REG ↓	Acc (%) ↑	Len ↓	REG ↓
DeepSeek-R1-Distill-Qwen-1.5B															
Base LRM	29.2	16757.1	2239.0	23.5	16577.6	1485.9	71.6	9956.8	2425.1	32.0	8212.0	970.7	39.1	12875.9	1780.2
Vanilla RL	42.2	12902.2	1266.2	28.2	13984.8	1077.7	81.9	8044.4	1742.6	35.2	8305.8	641.3	46.9	10809.3	1181.9
REO-RL (Exp) (ours)	$42.9_{\uparrow 0.7}$	8780.3	↓ 46.6%	$31.9_{\uparrow 3.6}$	8630.5	↓ 64.4%	$84.5_{\uparrow 2.7}$	4990.8	↓ 63.6%	$36.1_{\uparrow 0.8}$	5094.7	↓ 51.7%	$48.8_{\uparrow 2.0}$	6874.1	↓ 57.6%
REO-RL (Oracle) (ours)	$42.5_{\uparrow 0.3}$	8443.8	↓ 30.0%	$34.7_{\uparrow 6.5}$	8288.7	↓ 85.4%	$84.4_{\uparrow 2.5}$	4626.6	↓ 53.1%	$35.8_{\uparrow 0.5}$	4427.8	↓ 56.5%	$49.3_{\uparrow 2.5}$	6446.7	↓ 54.7%
REO-RL (Q-Spec) (ours)	$46.6_{\uparrow 4.4}$	9162.3	↓ 90.7%	$32.7_{\uparrow 4.5}$	8832.4	↓ 81.4%	$84.1_{\uparrow 2.2}$	5338.5	↓ 66.6%	$35.4_{\uparrow 0.2}$	5979.7	↓ 45.0%	$49.7_{\uparrow 2.8}$	7328.2	↓ 73.5%
RL w. Token Budget=1K	$17.3_{\downarrow 24.9}$	1487.4	↑ 170.4%	$13.5_{\downarrow 14.7}$	1282.8	↑ 140.7%	$65.6_{\downarrow 16.2}$	1111.3	↑ 43.1%	$32.0_{\downarrow 3.2}$	1108.8	↓ 24.1%	$32.1_{\downarrow 14.8}$	1247.6	↑ 90.3%
RL w. Token Budget=2K	$22.4_{\downarrow 19.8}$	2423.9	↑ 114.2%	$17.5_{\downarrow 10.7}$	1966.0	↑ 90.6%	$73.1_{\downarrow 8.8}$	1547.1	↓ 13.6%	$33.9_{\downarrow 1.3}$	1671.8	↓ 57.0%	$36.7_{\downarrow 10.1}$	1902.2	↑ 38.5%
RL w. Token Budget=4K	$31.4_{\downarrow 10.8}$	3991.0	↑ 25.8%	$26.1_{\downarrow 2.1}$	3770.8	↓ 17.3%	$80.9_{\downarrow 0.9}$	2561.2	↓ 60.3%	$35.2_{0.0}$	2942.8	↓ 65.5%	$43.4_{\downarrow 3.5}$	3316.5	↓ 28.2%
RL w. Len Group Norm. Rew.	$44.4_{\uparrow 2.2}$	10486.0	↓ 37.5%	$30.7_{\uparrow 2.5}$	10599.6	↓ 33.7%	$82.7_{\uparrow 0.9}$	5911.5	↓ 30.7%	$35.8_{\uparrow 0.6}$	6057.2	↓ 34.6%	$48.4_{\uparrow 1.5}$	8263.6	↓ 33.3%
RL w. Len-Harmonizing Rew.	$41.5_{\downarrow 0.7}$	9119.2	↓ 31.3%	$31.6_{\uparrow 3.3}$	8113.9	↓ 60.0%	$85.1_{\uparrow 3.2}$	3993.2	↓ 65.1%	$35.4_{\uparrow 0.2}$	3721.0	↓ 69.6%	$48.4_{\uparrow 1.5}$	6236.8	↓ 55.5%
MRT	$42.9_{\uparrow 0.7}$	9797.8	↓ 45.3%	$29.8_{\uparrow 1.6}$	9823.0	↓ 41.1%	$84.9_{\uparrow 3.0}$	5607.8	↓ 54.2%	$35.0_{\downarrow 0.2}$	5445.0	↓ 26.1%	$48.2_{\uparrow 1.3}$	7668.4	↓ 45.0%
$SFT_{Shortest}$	$43.2_{\uparrow 1.0}$	12898.1	↓ 11.1%	$29.9_{\uparrow 1.7}$	14221.8	↓ 22.3%	$82.5_{\uparrow 0.6}$	7513.7	↓ 9.6%	$35.2_{\downarrow 0.1}$	7641.9	↓ 11.5%	$47.7_{\uparrow 0.8}$	10568.9	↓ 13.2%
SFT_{TOPS}	$41.8_{\downarrow 0.4}$	10823.9	↓ 8.4%	$30.7_{\uparrow 2.5}$	11720.5	↓ 33.7%	$82.7_{\uparrow 0.9}$	6454.1	↓ 17.5%	$31.8_{\downarrow 3.4}$	4666.3	↑ 30.4%	$46.8_{\downarrow 0.1}$	8416.2	↓ 12.3%
$SimPO_{DAST}$	$22.9_{\downarrow 19.3}$	14029.4	↑ 112.1%	$19.4_{\downarrow 8.9}$	11161.7	↑ 70.0%	$68.1_{\downarrow 13.8}$	5955.5	↑ 42.2%	$32.6_{\downarrow 2.6}$	5906.4	↑ 2.1%	$35.7_{\downarrow 11.1}$	9263.3	↑ 61.8%
$SimPO_{Shortest}$	$35.8_{\downarrow 6.4}$	7422.1	↑ 0.5%	$26.5_{\downarrow 1.8}$	7380.9	↑ 0.1%	$77.7_{\downarrow 4.2}$	4020.8	↓ 17.3%	$34.8_{\downarrow 0.4}$	3437.0	↓ 51.5%	$43.7_{\downarrow 3.2}$	5565.2	↓ 13.2%
$SimPO_{TOPS}$	$15.6_{\downarrow 26.6}$	2647.4	↑ 195.5%	$18.3_{\downarrow 9.9}$	2412.2	↑ 81.3%	$58.5_{\downarrow 23.4}$	1497.8	↑ 115.5%	$25.7_{\downarrow 9.5}$	1022.8	↑ 129.4%	$29.5_{\downarrow 17.3}$	1895.1	↑ 131.0%
DeepSeek-R1-Distill-Qwen-7B															
Base LRM	55.3	13062.1	1887.4	39.7	14241.9	1527.0	90.9	6177.3	1611.4	43.1	5575.8	855.7	57.2	9764.3	1470.4
Vanilla RL	66.2	14264.7	1579.0	52.9	16305.3	1453.1	93.9	7259.8	1722.2	44.6	7300.7	945.2	64.4	11282.6	1424.9
REO-RL (Exp) (ours)	$64.0_{\downarrow 2.3}$	7671.5	↓ 60.7%	$48.8_{\downarrow 4.2}$	8361.1	↓ 57.3%	$93.4_{\downarrow 0.5}$	4144.6	↓ 48.4%	$45.5_{\uparrow 0.9}$	3687.0	↓ 59.4%	$62.9_{\downarrow 1.5}$	5966.0	↓ 55.9%
REO-RL (Oracle) (ours)	$63.9_{\downarrow 2.4}$	9348.6	↓ 50.6%	$49.0_{\downarrow 4.0}$	9189.6	↓ 66.5%	$94.7_{\uparrow 0.8}$	4444.1	↓ 57.9%	$45.5_{\uparrow 0.9}$	4132.9	↓ 57.8%	$63.3_{\downarrow 1.2}$	6778.8	↓ 58.0%
REO-RL (Q-Spec) (ours)	$63.9_{\downarrow 2.4}$	8407.8	↓ 62.0%	$51.4_{\downarrow 1.6}$	9298.5	↓ 75.3%	$93.4_{\downarrow 0.5}$	4230.6	↓ 54.5%	$43.9_{\downarrow 0.7}$	3581.0	↓ 38.9%	$63.1_{\downarrow 1.3}$	6379.4	↓ 59.3%
RL w. Token Budget=1K	$26.7_{\downarrow 39.6}$	1242.4	↑ 176.4%	$19.7_{\downarrow 33.2}$	1168.0	↑ 137.4%	$73.0_{\downarrow 20.9}$	960.4	↑ 59.5%	$40.6_{\downarrow 3.9}$	690.7	↓ 31.1%	$40.0_{\downarrow 24.4}$	1015.4	↑ 98.0%
RL w. Token Budget=2K	$36.5_{\downarrow 29.8}$	1929.6	↑ 91.2%	$27.6_{\downarrow 25.3}$	1831.6	↑ 61.0%	$84.3_{\downarrow 9.6}$	1427.5	↓ 36.6%	$43.9_{\downarrow 0.7}$	1210.1	↓ 74.8%	$48.1_{\downarrow 16.3}$	1599.7	↑ 17.3%
RL w. Token Budget=4K	$48.3_{\downarrow 17.9}$	3480.9	↑ 0.8%	$35.4_{\downarrow 17.5}$	3345.2	↓ 2.0%	$89.1_{\downarrow 4.8}$	2109.2	↓ 59.7%	$43.5_{\downarrow 1.1}$	1772.8	↓ 55.3%	$54.1_{\downarrow 10.3}$	2677.0	↓ 27.5%
RL w. Len Group Norm. Rew.	$64.2_{\downarrow 2.1}$	10608.1	↓ 28.8%	$50.8_{\downarrow 2.1}$	11486.8	↓ 47.9%	$94.5_{\uparrow 0.5}$	4788.9	↓ 43.7%	$45.2_{\uparrow 0.6}$	4309.4	↓ 45.5%	$63.7_{\downarrow 0.7}$	7798.3	↓ 40.9%
RL w. Len-Harmonizing Rew.	$65.5_{\downarrow 0.7}$	9167.7	↓ 37.0%	$51.1_{\downarrow 1.8}$	10394.6	↓ 46.1%	$94.5_{\uparrow 0.5}$	4307.9	↓ 52.7%	$44.4_{\downarrow 0.2}$	3617.9	↓ 42.2%	$63.9_{\downarrow 0.5}$	6872.0	↓ 44.9%
MRT	$66.1_{\downarrow 0.1}$	9210.6	↓ 42.4%	$50.5_{\downarrow 2.4}$	10559.3	↓ 55.9%	$93.8_{\downarrow 0.2}$	4986.6	↓ 38.6%	$44.7_{\downarrow 0.1}$	4852.7	↓ 32.4%	$63.8_{\downarrow 0.6}$	7402.3	↓ 43.0%
$SFT_{Shortest}$	$67.7_{\uparrow 1.5}$	13197.2	↓ 13.7%	$53.6_{\uparrow 0.7}$	15208.6	↓ 11.1%	$93.6_{\downarrow 0.3}$	6728.2	↓ 4.0%	$44.6_{\downarrow 0.0}$	6623.5	↓ 9.4%	$64.9_{\uparrow 0.5}$	10439.4	↓ 9.4%
SFT_{TOPS}	$66.0_{\downarrow 0.2}$	13204.1	↓ 9.0%	$52.0_{\downarrow 0.9}$	15170.8	↓ 11.7%	$93.3_{\downarrow 0.6}$	6272.5	↓ 11.9%	$44.2_{\downarrow 0.4}$	5759.7	↓ 30.4%	$63.9_{\downarrow 0.6}$	10101.8	↓ 14.1%
$SimPO_{DAST}$	$60.1_{\downarrow 6.1}$	7500.5	↓ 39.0%	$46.2_{\downarrow 6.7}$	7916.9	↓ 47.9%	$92.0_{\downarrow 1.9}$	4051.7	↓ 46.4%	$45.5_{\uparrow 0.9}$	3402.6	↓ 61.8%	$61.0_{\downarrow 3.4}$	5717.9	↓ 47.3%
$SimPO_{Shortest}$	$65.5_{\downarrow 0.7}$	9348.1	↓ 38.5%	$50.8_{\downarrow 2.1}$	10292.3	↓ 45.7%	$93.4_{\downarrow 0.5}$	4793.1	↓ 36.2%	$45.4_{\uparrow 0.8}$	4276.4	↓ 46.7%	$63.8_{\downarrow 0.6}$	7177.5	↓ 41.0%
HGPO	$68.1_{\uparrow 1.9}$	13556.8	↓ 4.6%	$53.1_{\uparrow 0.2}$	15609.8	↑ 2.3%	$93.4_{\downarrow 0.5}$	7077.3	↑ 2.7%	$45.0_{\uparrow 0.4}$	6873.4	↓ 15.6%	$64.9_{\uparrow 0.5}$	10779.3	↓ 0.4%
Qwen3-4B															
Base LRM	72.1	14808.7	614.3	64.7	17618.7	335.0	97.0	8397.1	955.8	46.5	6864.3	654.3	70.1	11922.2	639.9
REO-RL (Exp) (ours)	$70.0_{\downarrow 2.1}$	10799.0	↓ 87.8%	$59.9_{\downarrow 4.8}$	13063.4	↓ 45.6%	$93.8_{\downarrow 3.2}$	5307.7	↓ 74.4%	$45.2_{\downarrow 1.3}$	3653.3	↓ 40.7%	$67.2_{\downarrow 2.8}$	8205.9	↓ 65.2%
REO-RL (Q-Spec) (ours)	$67.9_{\downarrow 4.2}$	12066.1	↓ 21.7%	$62.1_{\downarrow 2.6}$	14755.0	↓ 48.5%	$95.1_{\downarrow 2.0}$	5662.4	↓ 63.6%	$46.0_{\downarrow 0.5}$	4203.5	↓ 39.7%	$67.8_{\downarrow 2.3}$	9171.7	↓ 45.5%
RL w. Token Budget=1K	$25.9_{\downarrow 46.1}$	22992.1	↑ 576.6%	$23.5_{\downarrow 41.1}$	21087.8	↑ 815.4%	$73.6_{\downarrow 23.4}$	19078.4	↑ 139.6%	$43.5_{\downarrow 3.0}$	10106.0	↑ 27.9%	$41.6_{\downarrow 28.4}$	18316.1	↑ 304.4%
RL w. Token Budget=2K	$39.5_{\downarrow 32.6}$	12996.9	↑ 311.4%	$30.1_{\downarrow 34.6}$	12546.4	↑ 575.7%	$85.6_{\downarrow 11.4}$	7827.0	↓ 0.8%	$46.0_{\downarrow 0.6}$	5542.0	↓ 34.5%	$50.3_{\downarrow 19.8}$	9728.1	↑ 141.0%
RL w. Token Budget=4K	$52.6_{\downarrow 19.5}$	7909.6	↑ 77.3%	$37.3_{\downarrow 27.4}$	8809.0	↑ 370.7%	$90.8_{\downarrow 6.2}$	5316.3	↓ 62.6%	$47.3_{\uparrow 0.8}$	4302.1	↓ 66.5%	$57.0_{\downarrow 13.1}$	6584.2	↑ 26.7%
RL w. Token Budget=8K	$64.1_{\downarrow 8.0}$	10240.9	↓ 3.7%	$54.2_{\downarrow 10.5}$	11042.7	↑ 64.7%	$93.0_{\downarrow 4.0}$	6050.0	↓ 23.2%	$47.4_{\uparrow 0.8}$	5067.9	↓ 40.7%	$64.7_{\downarrow 5.4}$	8100.4	↓ 11.5%
RL w. Len Group Norm. Rew.	$70.0_{\downarrow 2.1}$	15022.9	↑ 51.0%	$63.1_{\downarrow 1.6}$	16825.9	↑ 104.7%	$95.6_{\downarrow 1.4}$	7682.7	↓ 7.5%	$47.5_{\uparrow 1.0}$	6322.9	↓ 23.7%	$69.1_{\downarrow 1.0}$	11463.6	↑ 17.1%
RL w. Len-Harmonizing Rew.	$67.7_{\downarrow 4.4}$	15265.5	↓ 0.4%	$56.2_{\downarrow 8.4}$	18466.5	↑ 183.1%	$95.0_{\downarrow 2.0}$	7380.8	↓ 41.9%	$45.5_{\downarrow 1.1}$	7771.8	↑ 15.9%	$66.1_{\downarrow 4.0}$	12221.1	↑ 12.3%
HGPO	$72.4_{\uparrow 0.3}$	15123.8	↓ 16.4%	$65.4_{\uparrow 0.7}$	17951.8	↑ 49.6%	$96.2_{\downarrow 0.9}$	8858.2	↑ 16.1%	$47.7_{\uparrow 1.2}$	6797.4	↓ 11.7%	$70.4_{\uparrow 0.3}$	12182.8	↑ 5.6%
Qwen3-8B															
Base LRM	75.1	15221.2	1402.2	67.2	17682.8	656.1	94.8	9050.7	1026.9	48.3	7281.1	682.0	71.3	12308.9	941.8
REO-RL (Exp) (ours)	$73.6_{\downarrow 1.5}$	9285.4	↓ 69.7%	$59.0_{\downarrow 8.2}$	11343.1	↓ 68.2%	$95.1_{\uparrow 0.3}$	4699.7	↓ 98.6%	$47.8_{\downarrow 0.5}$	3671.8	↓ 62.1%	$68.9_{\downarrow 2.5}$	7250.0	↓ 75.9%
REO-RL (Q-Spec) (ours)	$73.2_{\downarrow 1.9}$	10852.9	↓ 37.1%	$64.4_{\downarrow 2.8}$	12645.2	↓ 84.8%	$94.6_{\downarrow 0.2}$	5416.1	↓ 71.5%	$46.8_{\downarrow 1.6}$	3952.7	↓ 36.1%	$69.7_{\downarrow 1.6}$	8216.7	↓ 54.6%
RL w. Token Budget=1K	$32.3_{\downarrow 42.8}$	5170.2	↑ 201.8%	$22.7_{\downarrow 44.5}$	5291.7	↑ 439.9%	$67.6_{\downarrow 27.2}$	2042.4	↑ 212.6%	$39.6_{\downarrow 8.8}$	1419.3	↑ 77.6%	$40.5_{\downarrow 30.8}$	3480.9	↑ 223.7%
RL w. Token Budget=2K	$52.2_{\downarrow 22.9}$	5151.0	↑ 33.1%	$35.1_{\downarrow 32.1}$	4696.1	↑ 203.0%	$84.8_{\downarrow 9.9}$	3840.4	↑ 8.8%	$47.4_{\downarrow 0.9}$	3034.1	↓ 56.2%	$54.9_{\downarrow 16.5}$	4180.4	↑ 39.9%
RL w. Token Budget=4K	$62.3_{\downarrow 12.8}$	6905.4	↓ 18.2%	$45.3_{\downarrow 21.9}$	6920.0	↑ 49.2%	$89.0_{\downarrow 5.8}$	4274.1	↓ 32.6%	$48.1_{\downarrow 0.2}$	3426.4	↓ 62.0%	$61.2_{\downarrow 10.2}$	5381.5	↓ 18.3%
RL w. Token Budget=8K	$71.7_{\downarrow 3.4}$	7363.9	↓ 84.8%	$51.4_{\downarrow 15.8}$	8359.4	↑ 9.5%	$92.1_{\downarrow 2.7}$	4437.3	↓ 73.4%	$48.4_{\uparrow 0.0}$	3534.8	↓ 74.2%	$65.9_{\downarrow 5.5}$	5923.8	↓ 63.4%
RL w. Len Group Norm. Rew.	$74.2_{\downarrow 0.9}$	12841.8	↓ 20.2%	$64.8_{\downarrow 2.4}$	14991.5	↓ 63.3%	$94.5_{\downarrow 0.2}$	6366.9	↓ 48.0%	$46.7_{\downarrow 1.6}$	5138.1	↓ 10.9%	$70.1_{\downarrow 1.3}$	9834.6	↓ 33.6%
RL w. Len-Harmonizing Rew.	$74.8_{\downarrow 0.3}$	15489.7	↑ 17.7%	$65.8_{\downarrow 1.4}$	18779.6	↑ 54.8%	$95.3_{\uparrow 0.5}$	8655.2	↓ 1.1%	$47.9_{\downarrow 0.5}$	7707.5	↑ 17.2%	$71.0_{\downarrow 0.4}$	12658.0	↑ 18.9%
HGPO	$75.3_{\uparrow 0.2}$	15547.2	↑ 10.6%	$68.0_{\uparrow 0.8}$	18112.5	↑ 7.6%	$95.3_{\uparrow 0.5}$	9292.2	↑ 9.7%	$48.9_{\uparrow 0.5}$	7605.6	↓ 5.9%	$71.9_{\uparrow 0.5}$	12639.4	↑ 6.8%

Table 1: 所有方法的准确性、生成长度和推理效率差距（REG）。对于准确性和 REG，报告了相对于基础 RL 基线或基础 LRM 的相对变化。 REO-RL 可以显著减少从 LRM 到推理效率边界的差距，同时几乎不降低甚至不降低准确性。

除了这些主要发现之外，我们还有以下观察，

基础强化学习在不同预算中没有带来一致的改进。虽然基础强化学习相对于基础 LRM 提高了整体准确率，但无法在不同的 token 预算中提供稳固的提升。在 7B 实验中，当 token 预算低于 8K 时，其准确率低于基础 LRM。

HGPO 不一定通过自适应思维模式切换来提高推理效率。 我们发现 HGPO 可以提高整体准确性，但不一定能提高推理效率和减少响应长度。在实验中，我们观察到 LRM 在无思考模式下的训练过程中能够获得更高的奖励，同时响应长度也在增加。这表明即使在无思考提示的情况下，LRM 也会恢复到思考模式，即在 “<think>...</think>” 标签之外执行长链式思考推理。

6.3 与前沿 LRM 的比较

	Claude Sonnet 3.7 (Thinking)	DeepSeek R1	Qwen3-4B	Qwen3-8B	Qwen3-32B	Vanilla RL - 7B	REO-RL (Exp) - 7B (ours)
Length	17478.87	5156.39	8631.61	8898.57	7755.99	7732.11	4524.02
Accuracy	90.0 %	98.2 %	95.8 %	94.9 %	97.2 %	93.5 %	93.5 %

Table 2: 将响应长度与前沿 LRM 进行比较。

为了进一步评估使用 REO-RL （Exp）训练的学习关系模型（LRMs）在推理效率方面与先进 LRMs 的对比，我们进行了一项专注于正确响应长度的对照分析。由于前沿 LRMs 和那些用 REO-RL （Exp）微调的模型在整体准确性上存在显著差异，我们从测试集中构建了一个由 71 个问题组成的平衡子集，这些问题中所有模型的准确性均超过 50 %。对于每个模型，我们计算每个问题正确推理路径的平均长度。然后，通过对 71 个问题的正确响应长度求平均来获得每个模型的最终长度指标。如表 2 所示，通过使用 REO-RL （Exp）微调 DeepSeek-R1-Distill-Qwen-7B 获得的模型表现出比前沿 LRMs 更简洁的推理模式，且响应长度短得多。

6.4 REO-RL 的消融研究

Method	AIME 2024			AIME 2025			AMC 2023			Minerva Math			Average		
	Acc ↑	Length ↓	REG	Acc ↑	Length ↓	REG	Acc ↑	Length ↓	REG	Acc ↑	Length ↓	REG	Acc ↑	Length ↓	REG
Base LRM	55.3	13062.1	1887.4	39.7	14241.9	1527.0	90.9	6177.3	1611.4	43.1	5575.8	855.7	57.2	9764.3	1470.4
Vanilla RL	66.2	14264.7	1579.0	52.9	16305.3	1453.1	93.9	7259.8	1722.2	44.6	7300.7	945.2	64.4	11282.6	1424.9
REO-RL (Exp)	$64.0_{\uparrow 2.3}$	7671.5	↓ 60.7%	$48.8_{\uparrow 4.2}$	8361.1	↓ 57.3%	$93.4_{\uparrow 0.5}$	4144.6	↓ 48.4%	$45.5_{\uparrow 0.9}$	3687.0	↓ 59.4%	$62.9_{\uparrow 1.5}$	5966.0	↓ 55.9%
REO-RL (Oracle)	$63.9_{\uparrow 2.4}$	9348.6	↓ 50.6%	$49.0_{\uparrow 4.0}$	9189.6	↓ 66.5%	$94.7_{\uparrow 0.8}$	4444.1	↓ 57.9%	$45.5_{\uparrow 0.9}$	4132.9	↓ 57.8%	$63.3_{\uparrow 1.2}$	6778.8	↓ 58.0%
REO-RL (Task-Specific)	$63.9_{\uparrow 2.4}$	8407.8	↓ 62.0%	$51.4_{\uparrow 1.6}$	9298.5	↓ 75.3%	$93.4_{\uparrow 0.5}$	4230.6	↓ 54.5%	$43.9_{\downarrow 0.7}$	3581.0	↓ 38.9%	$63.1_{\uparrow 1.3}$	6379.4	↓ 59.3%
REO-RL (Task-Specific-Hard)	$61.5_{\downarrow 1.8}$	6804.2	↓ 53.5%	$47.9_{\uparrow 2.0}$	7092.5	↓ 79.8%	$92.0_{\downarrow 2.0}$	3253.7	↓ 60.7%	$44.9_{\uparrow 0.3}$	2314.7	↓ 72.8%	$61.5_{\downarrow 2.9}$	4866.3	↓ 65.6%
REO-RL (Linear)	$63.5_{\downarrow 2.7}$	8982.0	↓ 46.1%	$49.9_{\uparrow 4.0}$	9001.6	↓ 58.8%	$93.9_{\ 0.0}$	4290.9	↓ 48.4%	$44.9_{\uparrow 0.3}$	3733.6	↓ 47.5%	$63.1_{\uparrow 1.3}$	6502.0	↓ 50.3%
REO-RL (Exp) - K=10	$64.0_{\uparrow 2.3}$	9352.7	↓ 45.2%	$48.9_{\uparrow 4.1}$	9804.9	↓ 47.8%	$94.1_{\uparrow 0.2}$	4779.6	↓ 52.0%	$44.9_{\uparrow 0.3}$	4151.8	↓ 46.3%	$63.0_{\uparrow 1.5}$	7022.3	↓ 48.1%
REO-RL (Exp) - Coef=1	$62.6_{\downarrow 3.6}$	7218.5	↓ 70.0%	$47.2_{\downarrow 5.7}$	7525.9	↓ 73.0%	$93.1_{\downarrow 0.8}$	3736.4	↓ 63.4%	$45.4_{\uparrow 0.8}$	3052.7	↓ 73.4%	$62.1_{\downarrow 2.3}$	5383.4	↓ 69.3%

Table 3: DeepSeek-R1-Distilled-Qwen-7B 在所有基准测试上的消融研究结果

我们使用 DeepSeek-R1-Distill-Qwen-7B 模型对 REO-RL 的设计选择进行了消融研究，以更好地理解其灵活性和性能特征。我们证明， REO-RL 在不同配置下能够保持竞争性能。

令牌预算选择策略。我们评估采用 oracle 贪婪策略的 REO-RL (Oracle)。这个变体获得了比 REO-RL (Exp) 略好的性能，反映在较低的 REG 中。我们还研究了线性分布的令牌预算，这导致了不太理想的效率和更高的 REG。

REO-RL 目标中的系数 c_i 。将所有系数 c_i 统一设置为 1 会使模型更接近效率前沿并缩短响应长度。然而，这种方法会以整体准确性降低为代价。

选择的代币预算数量 N 。我们增加选择的代币预算数量至 $N = 10$ ，以探索更细的粒度是否能提高性能。在实践中，我们观察到使用更多的代币预算会导致更慢的收敛速度和更高的训练不稳定性。因此，这种配置整体上产生了较弱的结果。

特定问题的 Oracle 预算。我们研究了一种特定问题的 Oracle 策略，该策略为每个问题分配两个代币预算，即通过前沿估计实验（第 4.2 节）得出的推理代币最小预算和完整预算 L_{max} 。这一 Oracle 方法在 DeepSeek-R1-Distill-Qwen 模型上的表现优于 REO-RL (Exp)，但在 Qwen3 模型上的 REG 减少不及 REO-RL (Exp)。

7 结论

在这项工作中，我们研究了 LRM 的高效推理。我们引入了推理效率前沿，它表征了 LRM 的响应长度和准确性之间的经验最佳权衡。为了量化经过微调的 LRM 的推理效率，我们引入了推理效率差距（REG），这是一种统一的指标，能够同时捕捉准确性和长度。我们对现有方法进行了基准测试，揭示了当前微调方法和效率前沿之间的巨大差距。我们提出的方法是 REO-RL ， 能够大幅提升推理效率，并成为接近于最佳长度-准确性权衡的选择 。

References

[1] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

[2] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

[3] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024. URL https://arxiv.org/abs/2412.18925.

[4] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

[5] Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think, 2025. URL https://arxiv.org/abs/2505.13379.

[6] Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. Efficiently serving llm reasoning programs with certaindex. *arXiv preprint arXiv:2412.20993*, 2024.

[7] Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025. URL https://openreview.net/forum?id=wpK4IMJfdX.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Yang Liu, and Yahui Zhou. Skywork open reasoner series. https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680, 2025. Notion Blog.

[10] Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

[11] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

[12] Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models, 2025. URL https://arxiv.org/abs/2505.14631.

[13] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

[14] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025. URL https://arxiv.org/abs/2502.11886.

[15] Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. Thinkswitcher: When to think hard, when to think fast, 2025. URL https://arxiv.org/abs/2505.14183.

[16] Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, Chao Li, Sheng Xu, Dezhi Chen, Yun Chen, Zuo Bai, and Liwen Zhang. Fin-r1: A large language model for financial reasoning through reinforcement learning, 2025. URL https://arxiv.org/abs/2503.16252.

[17] Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.11896.

[18] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

[19] Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025. Notion Blog.

[20] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[21] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[22] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

[23] Tergel Munkhbat, Namgyu Ho, Seohyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*, 2025.

[24] OpenAI. Learning to reason with llms. urlhttps://openai.com/index/learning-to-reason-with-llms/. Accessed: 15 March 2025.

[25] Xiao Pu, Michael Saxon, Wenyue Hua, and William Yang Wang. Thoughtterminator: Benchmarking, calibrating, and mitigating overthinking in reasoning models, 2025. URL https://arxiv.org/abs/2504.13367.

[26] Penghui Qi, Zichen Liu, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Optimizing anytime reasoning via budget relative policy optimization, 2025. URL https://arxiv.org/abs/2505.13438.

[27] Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*, 2025.

[28] Ant Research RL Lab. Areal: Ant reasoning rl. https://github.com/inclusionAI/AReaL, 2025.

[29] Jianshu She, Zhuohao Li, Zhemin Huang, Qi Li, Peiran Xu, Haonan Li, and Qirong Ho. Hawkeye:efficient reasoning with model collaboration, 2025. URL https://arxiv.org/abs/2504.00424.

[30] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

[31] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*, 2025.

[32] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[33] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

[34] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[35] Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl, 2025. URL https://arxiv.org/abs/2505.10832.

[36] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025. URL https://arxiv.org/abs/2504.20571.

[37] Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. Scalable chain of thoughts via elastic reasoning, 2025. URL https://arxiv.org/abs/2505.05315.

[38] Junjie Yang, Ke Lin, and Xing Yu. Think when you need: Self-adaptive chain-of-thought learning, 2025. URL https://arxiv.org/abs/2504.03234.

[39] Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for llm reasoning, 2025. URL https://arxiv.org/abs/2502.18080.

[40] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[41] Hongzhou Yu, Tianhao Cheng, Ying Cheng, and Rui Feng. Finemedlm-o1: Enhancing the medical reasoning ability of llm from supervised fine-tuning to test-time training, 2025. URL https://arxiv.org/abs/2501.09213.

[42] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

[43] Zhaojian Yu, Yinghao Wu, Yilun Zhao, Arman Cohan, and Xiao-Ping Zhang. Z1: Efficient test-time scaling with code, 2025. URL https://arxiv.org/abs/2504.00810.

[44] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025. URL https://arxiv.org/abs/2504.05118.

[45] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think, 2025. URL https://arxiv.org/abs/2505.13417.

[46] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025.

[47] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs, 2024. URL `https://arxiv.org/abs/2312.07104`.

A 附加讨论

连接到 MRT [26] 和 BRPO [26]。 与我们的方法类似， REO-RL ， MRT 和 BRPO 都有在有限的标记预算下优化奖励的想法。然而， REO-RL 与这些之前或同时进行的方法之间存在关键差异。

First, the most important difference is the way to evaluate a partial response, especially the prompt and generation configuration to force the LRM to produce a plausible answer. REO-RL requires the LRM to directly output the final answer with a generation budget and a prompt similar to "The Final Answer is \ boxed { ". MRT and BRPO employs a looser strategy that stops the thinking process with "</think>" and allows the model to produce a summarization within a moderate generation budget. Within the summarization phase, the model is able to perform lightweight thinking, which gives the model an additional try beyond the pre-specified token budget. Crucially, through RL training, this design has the risk of allowing the LRM to learn to perform budget-aware reasoning within the summarization phase, which would be infeasible in practice since the optimal token budget for a question is unknown in advance, as also discussed in MRT [27]. Therefore, we follow s1 [22] to force the LRM to produce a plausible answer with minimal additional reasoning efforts.

其次，在这些工作中选择部分响应的方式各不相同。MRT 依赖于通过关键词将模型生成的响应划分为语义完整的步骤。BRPO 选择一小组线性间隔的 token 预算。 REO-RL 引入了多种类型的 token 预算选择策略，例如比线性方式更能实现更好结果的指数间隔 token 预算，并应用于特定问题的 token 预算。最后，MRT 优化单步奖励，而 BRPO 和 REO-RL 都采用密集奖励训练，并在广泛的 token 预算范围内优化总奖励。

B 可重复性

我们将在 https://github.com/samjia2000/Optimal-Reasoning-Efficiency 中提供我们的代码。请参阅第 F 节了解关于推理效率前沿和推理效率差距的详细信息，以及第 C 节了解实现细节。

C 实现细节

对于训练数据，我们整合了来自 DeepScaleR [20] 和 AReaL [28] 的数据。对于至少 4B 大小的模型，我们使用 AReaL-Boba-RL-7B [28] 的训练数据。对于 1.5B，我们采用 DeepScaleR [20] 和 AReaL [28] 的混合训练数据，并去除重复的问题。

我们使用 AReaL 框架 [28] 实现训练算法，该框架支持 SGLang [47] 用于展开生成。下面我们详细介绍每个训练算法的实现细节。

在线强化学习训练。 我们使用 PPO 作为默认的在线 RL 算法。根据针对 LLM 推理的 RL 训练中的标准实践 [42, 11] ，我们不使用价值模型和 KL 正则化。PPO 训练的默认训练设置和超参数列在表中 4 。

对于 REO-RL 和基线方法，我们并不直接从基础 LRM 进行 RL 训练，因为这会导致训练时间过长和收敛速度变慢。相反，我们对经过 RL 训练的 1.5B 和 7B 设置版本进行进一步微调。具体来说，我们采用 AReaL-Boba-RL-1.5B [28] 和 Skywork-OR1-Math-7B [9] 作为进一步微调的起点。按照表 4 中的集群配置，每个实验可以在 4K GPU 小时内完成。

监督微调 SFT 的默认训练配置和超参数列在表 5 中。

偏好学习。 我们在 AReaL 框架 [28] 中实现了 SimPO [21] 。SimPO 的默认训练配置和超参数列在 Tab. 5 中。

D 基线

在我们的在线强化学习训练中，受限于 token 预算，我们在每个训练阶段控制生成的最大长度。我们采用一种渐进的长度缩减策略，而不是直接在固定的 token 预算上微调目标语言模型。我们从 16K 的 token 预算开始训练。当该层次的强化学习训练收敛后，我们将预算减少到 8K 并继续训练。这个过程持续进行，每次将 token 预算减半，直到我们达到最小的 512

Table 4: PPO 的默认训练配置和超参数。

Training Configuration	
Batch size (number of prompts)	128
Rollouts per prompt	16
Random seed	1
Cluster Config	8×8 H800 (for 1.5B) / 16×8 H800 (others)
PPO Parameters	
PPO Minibatches	4
Clipping ϵ	0.2
Advantage normalization	True
Discount factor γ	1.0
GAE λ	1.0
Epochs	2.0
Optimizer Parameters	
Optimizer	Adam
Learning rate	5×10^{-6}
Weight decay	0.05
β_1	0.9
β_2	0.95
Adam ϵ	1×10^{-5}
Gradient norm clipping	1.0
Learning rate scheduler	constant
Warmup steps proportion	0.001
Generation Parameters	
Temperature	1.0
Top-p	1.0
Top-k	-1
Max prompt length	1024
Min generation length	0
Max generation length	24376 (for 1.5B) / 32768 (others)

token 预算。这个分阶段的方法使得模型能够在保持推理性能的同时，逐步适应较短的生成长度。

在“使用长度组标准化奖励的 RL”基线中 [2]，对于每个问题 x 及其对应的采样响应集 $y^1, \cdots, y^m$，响应 y^i 的奖励被计算为，

$$r(x, y^i) = \mathbb{I}\{y^i \text{ is correct}\}(1 - \alpha f(|y^i|))$$

其中函数 f 根据正确响应的长度对 $|y^i|$ 进行标准化并应用一个 sigmoid 函数。具体来说，

$$f(|y^i|) = \sigma\left(\frac{|y^i| - \texttt{MEAN}(x)}{\texttt{STD}(x)}\right)$$

其中

$$\texttt{MEAN}(x) = \mathbb{E}_{y \sim \pi(\cdot|x), s.t. y \text{ is correct}}[|y|]$$
$$\texttt{STD}(x) = \sqrt{\text{Var}_{y \sim \pi(\cdot|x), s.t. y \text{ is correct}}[|y|]}$$

在“RL with Length-Harmonizing Rewards”基线 [18] 中，对于每个问题 x 和相应的一组采样响应 $y^1, \cdots, y^m$，响应 y^i 的奖励被计算为，

$$r(x, y^i) = \frac{\overline{L}_{ref}(x)}{|y|} - 1 + \gamma \cdot (\mathbb{I}\{y \text{ is correct}\} - \overline{A}_{ref}(x))$$

Table 5: 用于 SFT 的默认训练配置和超参数。

Training Configuration	
Batch size (number of prompt-answer pairs)	512
Cluster Config	16×8 H800
SFT Parameters	
Epochs	10
Save Frequency Steps	100
use_bf16	True
Max Seq Length	32768
Optimizer Parameters	
Optimizer	Adam
Learning rate	1×10^{-5}
Weight decay	0.05
β_1	0.9
β_2	0.95
Adam ϵ	1×10^{-5}
Gradient norm clipping	1.0
Learning rate scheduler	constant
Warmup steps proportion	0.03

Table 6: SimPO 的默认训练配置和超参数。

Training Configuration	
Batch size (number of preference pairs)	128
Cluster Config	16×8 H800
SimPO Parameters	
Epochs	2
Save Frequency Steps	10
use_bf16	True
Max Seq Length	32768
SimPO Coefficient β	1/2
SimPO Coefficient γ	1.2/1.4
Optimizer Parameters	
Optimizer	Adam
Learning rate	1×10^{-5} (for 1.5B) / 3×10^{-6} (or 7B)
Weight decay	0.05
β_1	0.9
β_2	0.95
Adam ϵ	1×10^{-5}
Gradient norm clipping	1.0
Learning rate scheduler	constant
Warmup steps proportion	0.03

其中 $\overline{L}_{ref}(x)$ 是参考模型在接受 x 作为输入时的平均响应长度，$\overline{A}_{ref}(x)$ 是参考模型的平均准确率。在我们的实现中，我们将作为 RL 训练起点的模型设定为参考模型。

在 MRT 基线中 [27]，与原始论文中仅使用离线收集的响应前缀进行单步优化不同，我们实现了在线强化学习训练版本，并为 MRT 设置了密集的奖励。对于每个问题 x 及其对应的一组采样响应，我们将每个响应分为几个步骤 $y = (y_1, \cdots, y_s)$。在每个训练步骤中，模型通

过计算以下目标的策略梯度来进行更新，

$$\mathbb{E}_{x,y=(y_1,\cdots,y_s)\sim\pi_{\theta'}(\cdot|x)}[\sum_{i=1}^{s}\mathbb{E}_{y'_i\sim\pi_\theta(\cdot|x,y_{:i-1})}[\mathcal{R}(x,\text{Answer}(\pi_{\theta'},x,[y_{:i-1};y']))$$
$$-\mathcal{R}(x,\text{Answer}(\pi_{\theta'},x,y_{:i-1}))+\alpha\cdot\mathcal{R}(x,y)]]$$

其中 α 是整体准确性的权重，并设置为 0.2 。

混合推理。 我们用 “$\backslash n$ ” 和 “$\backslash n\ \backslash n$ </think>” 作为开始思考和不思考的标记。我们首先在思考和不思考模式下对模型生成的回复运行 SFT，以确保模型在不思考模式下初始具有非零的概率。然后，我们应用 HGPO 训练，并遵循 [12] 以 0.2 为边界。

在 $\text{SFT}_{\text{Shortest}}$ 中，我们为训练数据集中的每个问题生成 16 个输出。然后，我们为每个问题选择最短的正确回答以构建 SFT 数据集。在 SFT_{TOPS} 中，我们根据 [39] 提示 LRM 生成具有三种不同推理努力类型的响应。我们严格遵循 [39] 中使用的提示。对于每种类型的推理努力，我们生成 16 个响应。为了构建 SFT 数据集，从所有 48 个响应中收集最短的正确响应。

偏好学习。 我们采用三种策略来构建偏好数据集。在 $\text{SimPO}_{\text{Shortest}}$ 中，我们采用为 $\text{SFT}_{\text{Shortest}}$ 生成的响应，并选择最短正确响应和最长响应作为每个问题的偏好对。在 $\text{SimPO}_{\text{TOPS}}$ 中，我们使用与 $\text{SimPO}_{\text{Shortest}}$ 相同的偏好构建策略，但应用于通过 TOPS 生成的响应，这些响应包含具有不同推理努力的推理痕迹。最后，在 $\text{SimPO}_{\text{DAST}}$ 中，我们再次利用为 $\text{SFT}_{\text{Shortest}}$ 生成的响应，但采用不同的偏好构建策略。在 $\text{SimPO}_{\text{DAST}}$ 的偏好数据集中，每对落入两种情况之一：它要么包含两个正确响应，其中正样本比负样本短得多，要么包含两个错误响应，其中正样本比负样本长得多。

我们也尝试了 Z1 [43] ，它构建基于代码的推理轨迹，以及 L1 [1] ，它微调 LRMs 以通过 RL 遵循令牌预算指令。我们发现 Z1 在数学推理任务上的表现较差，其准确性显著低于基础 LRMs，如表 7 所示。对于 L1，我们发现 L1 主要适用于紧张的令牌预算，即小于 $4K$ 。当我们将 L1 的训练方法扩展到更大的上下文长度时，即 1.5B 模型的 24K，我们发现很难通过 RL 使 LRM 学习遵循严格的令牌预算指令。因此，结果模型 L1-Exact-24K-1.5B 和 L1-Max-24K-1.5B 无法遵循令牌预算指令，如表 8 所示。

Method	AIME24		MATH500		GPQA	
	Accuracy (%)	Length	Accuracy (%)	Length	Accuracy (%)	Length
Base LRM	31.5	16747.6	83.6	5633.1	44.6	10325.2
Z1	10.0	15106.0	63.6	4904.1	61.6	9004.1

Table 7: Z1 在 DeepSeek-R1-Distill-Qwen-1.5B 上的结果。 [43]

Instructed Token Budget	AMC23		AIME24		AIME25	
	Accuracy (%)	Length	Accuracy (%)	Length	Accuracy (%)	Length
2048	63.9	20267.7	38.6	18067.4	27.2	19657.7
4096	63.5	20142.6	38.4	17961.3	26.9	19711.1
8192	62.9	20191.6	38.4	17868.7	26.8	19673.5

Table 8: L1-Exact-24K-1.5B 的结果。

E REO-RL

E.1 实现 REO-RL

在 REO-RL 的生成阶段，有两轮 LRM 生成。在第一轮中，为训练批次中的每个问题生成多个响应。在第二轮中，为了计算每个响应及跨所有选定的 token 预算的长度约束奖励，我们选择所有截断响应 $y_{:L_i}$ ，并应用一个提示以强制 LRM 在给定不完整的推理轨迹时生成最终答案，即 $a=\pi_\theta(\cdot|x,y_{:L},[\text{The Final Answer is}])$. 。我们遵循 [6] 和 [7] 采用的提示。

Prompt for Forcing LRM to Produce Answer

...
Oh, I suddenly got the answer to the whole problem. **Final Answer**:
[\ boxed {

稠密奖励强化学习。 REO-RL 通过强迫 LRM 在各种 token 预算下生成答案来获得密集奖励。 REO-RL 的目标如下所示，

$$REO-RL: \quad \mathscr{L}_{REO-RL}(\theta, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\sum_{i=1}^{N+1} c_i \mathcal{R}(x, \text{Answer}(\pi_\theta, x, y_{:L_i}))\right]\right]$$

其中 $c_i = \frac{L_{i+1}-L_{i-1}}{2}$ 对于 $1 \le i \le N$ 和 $c_{N+1} = \frac{L_{\max}-L_N}{2}$ 是 i -th token 预算的系数。

为了执行策略更新，我们计算两个连续的 token 预算 L_i 和 L_{i+1} 之间每个部分的回报。对于 $1 \le i \le N$ ，我们计算，

$$\text{Return}_\text{i}(\text{x}, \text{y}) = \sum_{j=i}^{N+1} c_j \mathcal{R}(x, \text{Answer}(\pi_\theta, x, y_{:L_j}))$$

由于在训练过程中我们禁用了价值模型，因此计算出的回报直接用作损失计算的优势。 在实际操作中，对于每个提示，我们在策略更新步骤之前对每个 $i \in [1, N]$ 应用组归一化，即 $\text{Adv}_i(x, y^j) = \frac{\text{Return}(\text{x},\text{y}^\text{j}) - \frac{1}{M}\sum_{j'}\text{Return}(x, y^{j'})}{\sqrt{\text{Var}_{j'}[\text{Return}(x, y^{j'})]}}$ 。然后 $\text{Adv}_i(x, y)$ 将被用作优势来更新令牌 $y_{L_i:L_{i+1}}$ 。

E.2 REO-RL（Q-Spec）

REO-RL (Q-Spec)。 或者，我们注意到，对于每个问题 x ，应该存在一个特定的最小令牌预算 L^x 。这个最小令牌预算反映了基本 LRM 能够学习达到与在全令牌预算下经过良好训练的对应物相同准确度的最低令牌预算。

为了推导出针对具体问题的最小代币预算，我们采用了一种实际的估算方法，该方法从第 4.2 节中不同配置的训练过程中生成的推演中提取。对于一个问题 x ，第 4.2 节的强化学习实验生成了大量由不同微调模型与相应长度约束奖励关联的推演，即 $\mathcal{D}_{\text{rollouts}} = \{(\hat{\theta}, x, y^1, \cdots, y^M, r(x, \cdot; \hat{\theta}))\}$ 其中 $\hat{\theta}$ 表示一个微调的 LRM，并且 $y^1, \cdots, y^M$ 是由 $\hat{\theta}$ 针对问题 x 生成的 M 推演。从 $\mathcal{D}_{\text{rollouts}}$ 中，我们计算出满足的 L^x 。

$$L^x = \min\left\{L | L \in [1, L_{\max}] \text{ s.t. } \max_{(\hat{\theta}, x, y^1, \cdots, y^M)\in\mathcal{D}_{\text{rollout}}} \frac{\sum_i r(x, y^i_{:L}; \hat{\theta})}{M} \ge \max_{(\hat{\theta}, x, y^1, \cdots, y^M)\in\mathcal{D}_{\text{rollout}}} \frac{\sum_i r(x, y^i_{:L_{\max}}; \hat{\theta})}{M}\right\} \tag{12}$$

在实际中，对所有历史模型 $\hat{\theta}$ 、历史展开和所有可能的 L 进行 $r(x, y^i_{:L}; \hat{\theta})$ 的评估可能会很昂贵，因为需要额外的推理运行来评估如方程 **??** 中的截断响应。在计算最小令牌预算时，我们直接评估部分响应 $y^i_{:L}$ 的正确性，而不进行额外的操作来生成答案。

在 REO-RL（特定于问题）中，我们为每个问题 x 设定了 $N = 1$ 和 $L_1 = L^x$ ，形成了目标，

$$REO-RL(Q-Spec): \quad \mathscr{L}_{REO-RL\text{ (Q-Spec)}}(\theta, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[c_1 \cdot r(x, y_{:L^x}; \theta) + c_2 \cdot r(x, y_{:L_{\max}}; \theta)\right]\right] \tag{13}$$

其中 $c_1 = \frac{L^x}{2}, c_2 = \frac{L_{\max}-L^x}{2}$ 满足公式 13 。

F 推理效率前沿 & 推理效率差距

推理效率前沿 下面的两个框分别记录了 DeepSeek-R1-Distill-Qwen-1.5B 和 DeepSeek-R1-Distill-Qwen-7B 的估计推理效率边界上的点的详细长度和准确性。

Reasoning Efficiency Frontier for Qwen3-4B

x, y = [0, 64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9216, 10240, 11264, 12288, 13312, 14336, 15360, 16384, 17408, 18432, 19456, 20480, 21504, 22528, 23552, 24576, 25600, 26624, 27648, 28672, 29696, 30720, 31744, 32768], [0.05864545036764705, 0.07844286151960785, 0.08253484987745098, 0.09157475490196078, 0.09366000306372549, 0.10081380208333333, 0.10913181678921569, 0.12006548713235293, 0.14239813112745098, 0.16250957414215686, 0.18545879289215683, 0.20630935968137254, 0.22248008578431372, 0.23359183517156862, 0.24719860600490196, 0.25932329963235295, 0.26839958639705885, 0.37697610294117645, 0.4535807291666667, 0.5027018229166667, 0.5349207261029412, 0.5595760569852941, 0.5795496323529412, 0.5964384191176471, 0.6082050398284314, 0.6200099571078431, 0.6309244791666666, 0.6428308823529412, 0.6487189797794117, 0.6562059589460785, 0.6632372089460784, 0.6686408547794117, 0.6734145220588235, 0.6795764399509804, 0.6835171568627451, 0.6894416360294119, 0.693035768995098, 0.6970358455882353, 0.7003484987745099, 0.7034734987745098, 0.7032781862745098, 0.7061714920343137, 0.7074371936274509, 0.7081533394607843, 0.7090647977941177, 0.7103381587009804, 0.7101064644607843, 0.707257199754902]

Reasoning Efficiency Frontier for Qwen3-8B

x, y = [0, 64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9216, 10240, 11264, 12288, 13312, 14336, 15360, 16384, 17408, 18432, 19456, 20480, 21504, 22528, 23552, 24576, 25600, 26624, 27648, 28672, 29696, 30720, 31744, 32768], [0.09000842524509803, 0.10032935049019608, 0.09675245098039215, 0.11996208639705883, 0.1271771599264706, 0.1278818167892157, 0.14018267463235295, 0.16002795649509804, 0.1714862898284314, 0.2006414675245098, 0.21497970281862744, 0.24749348958333334, 0.2608475030637255, 0.2683632046568627, 0.2772250306372549, 0.28248697916666665, 0.29307789522058825, 0.39007352941176476, 0.4767539828431373, 0.5257927389705882, 0.5632065716911765, 0.5868183210784313, 0.6074180453431373, 0.6239028033088235, 0.6389916513480391, 0.6473824295343137, 0.6582471660539215, 0.666616881127451, 0.6744542738970589, 0.6808746936274509, 0.6869829963235294, 0.6924479166666666, 0.695980775122549, 0.7023954503676471, 0.7065142463235294, 0.7109911151960785, 0.7135799632352942, 0.7167777267156862, 0.7191789215686275, 0.7239296109068628, 0.726351868872549, 0.7280005361519608, 0.7298598345588236, 0.7314587162990196, 0.7330135569852941, 0.7345473345588236, 0.7354587928921568, 0.7288985906862746]

Reasoning Efficiency Frontier for DeepSeek-R1-Distill-Qwen-1.5B

x, y = [0, 64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9216, 10240, 11264, 12288, 13312, 14336, 15360, 16384, 17408, 18432, 19456, 20480, 21504, 22528, 23552, 24576, 25600, 26624, 27648, 28672, 29696, 30720, 31744, 32768], [0.06101600796568627, 0.05281671262254902, 0.06075367647058823, 0.07656441482843138, 0.09422679227941178, 0.09907322303921567, 0.11950635723039214, 0.14104051776960785, 0.16222426470588236, 0.17982153799019607, 0.19957299325980393, 0.22226179534313725, 0.23991076899509806, 0.25617723651960783, 0.2744064031862745, 0.2836722579656863, 0.29287109375, 0.3738262101715686, 0.413882506127451, 0.44010225183823526, 0.4490253523284314, 0.4542604932598039, 0.4631778492647059, 0.4701803768382353, 0.4771963082107843, 0.48008961397058825, 0.48356885723039217, 0.48675130208333334, 0.4884727328431372, 0.48968098958333334, 0.49039713541666663, 0.4909466911764706, 0.4908088235294118, 0.4905771292892157, 0.4908662683823529, 0.49117647058823527, 0.4919289981617647, 0.4914445465686274, 0.4919002757352941, 0.4920955882352941, 0.4919002757352941, 0.4919289981617647, 0.4919289981617647, 0.4919577205882353, 0.4919002757352941, 0.4919289981617647, 0.4920955882352941, 0.49215303308823527]

Reasoning Efficiency Frontier for DeepSeek-R1-Distill-Qwen-7B

x, y = [0, 64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192, 9216, 10240, 11264, 12288, 13312, 14336, 15360, 16384, 17408, 18432, 19456, 20480, 21504, 22528, 23552, 24576, 25600, 26624, 27648, 28672, 29696, 30720, 31744, 32768], [0.06838809742647059, 0.07453469669117647, 0.08459520526960784, 0.09630629595588236, 0.09808900122549019, 0.11145450367647058, 0.14210707720588237, 0.16989123774509804, 0.1799383425245098, 0.2068627450980392, 0.22867455575980392, 0.24759880514705881, 0.26799938725490197, 0.2935891544117647, 0.31299019607843137, 0.33319546568627456, 0.38615196078431374, 0.4691272212009804, 0.50302734375, 0.5375919117647059, 0.5522518382352941, 0.5640356924019608, 0.5759803921568628, 0.5929974724264706, 0.6051547181372549, 0.61650390625, 0.6240272671568627, 0.6303423713235294, 0.63447265625, 0.63720703125, 0.6392252604166667, 0.64111328125, 0.64345703125, 0.6455403645833333, 0.6470377604166666, 0.6482747395833333, 0.6487955729166667, 0.6512044270833334, 0.6522460937500001, 0.6525065104166667, 0.6532877604166667, 0.6522460937500001, 0.6527669270833334, 0.6535481770833333, 0.6535481770833333, 0.6535481770833333, 0.6532877604166667, 0.6532877604166667]

为实际评估 REG，而不是严格遵循公式 6 ，我们通过在一个代币预算集上分别进行数值积分，类似于公式 9 ，来获得 $\sum_{L=1}^{L_{\max}} \hat{J}_{\text{optimal}}$ 和 $\sum_{L=1}^{L_{\max}} J(\mathcal{D}, \theta, L)$ 的近似值。我们选择 $\{L_1, \cdots, L_N\} = \{64i | 0 \leq i < 16\} \cup \{1024i | 1 \leq i \leq 16\}$ 。注意，我们设置 $L_{\max} = 16K$ 而不是 $L_{\max} = 32K$ ，以便在较低代币预算下关注效率差异。

G

NeurIPS 论文清单 该清单旨在鼓励负责任的机器学习研究的最佳实践，解决再现性、透明度、研究伦理和社会影响等问题。不要移除清单：未附带核对清单的论文将被直接拒稿。清单应放在参考文献和（可选）补充材料之后。清单不计入页数限制。

请仔细阅读清单指南，以获取如何回答这些问题的信息。对于清单中的每一个问题：

- 你应该回答 [Yes] 、[No] 或 [NA] 。
- [NA] 意味着该问题不适用于该特定论文，或者相关信息不可用。
- 请在答案之后提供简短的（1-2 句）理由说明（即使是 NA）。

它们对评审员、领域主席、高级领域主席和伦理评审员是可见的。在最终修订后，您还需要将其包含在论文的最终版本中，并且其最终版本将与论文一起发布。

审稿人在评估您的论文时会将这个检查列表作为一个因素。虽然" [Yes] " 通常比" [No] " 更可取，但如果给出合理的理由（例如，“误差条未报告因为计算代价太高”或“我们找不到我们使用的数据集的许可”），回答" [No] " 是完全可以接受的。通常，回答" [No] " 或" [NA] " 并不是被拒稿的理由。尽管这些问题是以二元方式表述的，我们承认真实的答案通常更为复杂，因此请根据您的最佳判断进行回答，并写出理由进行详细说明。所有的支持性证据可以出现在主论文或附录提供的补充材料中。如果您在一个问题上回答了 [Yes] ，请在理由中指明可以找到相关材料的章节。

重要，请注意：

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist"** ，
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers** 。

1. **Claims**

   问题：摘要和介绍中提出的主要论点是否准确反映了论文的贡献和范围？

   回答：[Yes]

   理由：我们认为我们的引言和摘要部分都清楚地说明了本文所做的贡献。

   指南：

   - 答案 NA 意味着摘要和引言不包括论文中的主张。
   - 摘要和/或引言应清楚地陈述提出的主张，包括本文所做的贡献以及重要的假设和限制。评审员对该问题的否定或不适用的答案会有负面看法。
   - 所提出的主张应与理论和实验结果相符，并反映结果在多大程度上可以推广到其他情境。
   - 在文中加入具有启发性的目标作为动机是可以的，只要明确这些目标并没有在论文中实现即可。

2. **Limitations**

   问题：论文是否讨论了作者所做工作的局限性？

   回答：[Yes]

   理由：请参见第 7 节。

   指南：

   - 答案 NA 表示论文没有限制，而答案 No 表示论文有限制，但这些限制在论文中未被讨论。
   - 我们鼓励作者在他们的论文中创建一个单独的“局限性”部分。
   - 论文应指出任何强假设以及当这些假设被违反时结果的稳健性（例如，独立性假设、无噪音环境、模型设定完备性、渐近近似仅在局部成立）。作者应反思这些假设在实际中可能如何被违反以及这将产生什么影响。
   - 作者应反思所提出结论的适用范围，例如，如果该方法仅在少数数据集上测试过或仅进行了少数几次运行。通常，实证结果往往依赖于隐含的假设，这些假设应该被明确表达。

- 作者应当反思影响该方法性能的因素。例如，当图像分辨率较低或在弱光条件下拍摄时，面部识别算法可能表现不佳。或者，由于无法处理专业术语，语音转文本系统可能无法可靠地用于为在线讲座提供字幕。
- 作者应该讨论所提出算法的计算效率以及它们随着数据集规模的变化情况。
- 如果适用，作者应讨论其方法在解决隐私和公平性问题时可能存在的局限性。
- 虽然作者可能担心如实陈述限制会被审稿人用作拒稿的理由，但更糟糕的结果可能是审稿人发现了论文中没有承认的限制。作者应使用其最佳判断力，认识到有利于透明化的个人行为在形成维护群体诚信的规范中起到重要作用。审稿人将被特别指示不要因为诚实陈述限制而进行惩罚。

3. **Theory assumptions and proofs**

   问题：对于每一个理论结果，论文是否提供了完整的假设集和完整（且正确）的证明？

   回答：[NA]

   理由：不适用

   指南：

   - 答案 NA 表示论文不包含理论结果。
   - 文章中的所有定理、公式和证明都应编号并进行交叉引用。
   - 在任何定理的陈述中，所有假设都应被清楚地陈述或引用。
   - 证明可以出现在主文中或补充材料中，但如果出现在补充材料中，建议作者提供简短的证明概述以提供直观理解。
   - 相反，论文核心部分提供的任何非正式证明都应由附录或补充材料中提供的形式证明补充。
   - 证明所依赖的定理和引理应当被正确引用。

4. **Experimental result reproducibility**

   问题：论文是否完全披露了重新生成主要实验结果所需的所有信息，以充分影响论文的主要论点和/或结论（无论是否提供代码和数据）？

   回答：[Yes]

   理由：我们在附录 C 、附录 D 和附录 E 中提供了重新生成实验结果的完整指南。

   指南：

   - 答案 NA 表示论文不包含实验。
   - 如果论文包含实验，对于这个问题的否定答案在评审者看来是不好的：无论是否提供代码和数据，使论文具有可重复性都是重要的。
   - 如果贡献是一组数据集和/或模型，作者应该描述所采取的步骤，以使他们的结果可重现或可验证。
   - 根据贡献的不同，可通过多种方式实现可复现性。例如，如果贡献是一种新颖的架构，则充分描述该架构可能就足够了；如果贡献是特定的模型和实证评估，则可能需要让他人能够通过相同的数据集复制模型，或者提供对模型的访问。通常，发布代码和数据是实现这一目标的一种好方法，但也可以通过详细说明如何复制结果、提供托管模型的访问（例如，在大型语言模型的情况下）、发布模型检查点，或其他适合所进行研究的方式来提供可复现性。
   - 尽管 NeurIPS 不要求公开代码，但会议要求所有提交必须提供某种合理的方法以确保可重复性，这可能取决于所作贡献的性质。例如
     (a) 如果贡献主要是一个新算法，论文应该明确说明如何重现该算法。
     (b) 如果本文的贡献主要是一个新的模型架构，则应清晰完整地描述该架构。
     (c) 如果所贡献的是一个新模型（例如，一个大型语言模型），那么应有一种可以访问此模型以重现结果的方法，或者一种可以重现该模型的方法（例如，提供一个开源数据集或构建该数据集的说明）。
     (d) 我们认识到，在某些情况下，可重复性可能会有些棘手，在这种情况下，欢迎作者描述他们提供可重复性的具体方式。在闭源模型的情况下，可能会以某种方式限制对模型的访问（例如，仅限注册用户），但其他研究人员应该可以通过某种路径来重现或验证结果。

5. **Open access to data and code**

6. 问题：论文是否提供了开放获取数据和代码的权限，并附有充分的说明，以便如补充材料中所述，能够忠实地重现主要实验结果？

   答案：[Yes]

   理由：我们在附录中通过匿名链接提供了我们的训练代码。B 。我们的训练数据来自开源项目，我们已经适当引用。我们还在附录中提供了估计的推理效率前沿，这对于计算引入的指标 REG 是必要的。F 。

   指导方针：

   - 答案 NA 意味着该论文不包括需要代码的实验。
   - 详情请参阅 NeurIPS 代码和数据提交指南（`https://nips.cc/public/guides/CodeSubmissionPolicy` ）。
   - 虽然我们鼓励发布代码和数据，但我们理解这可能并不总是可行的，所以“否”是可以接受的答案。论文不能仅仅因为没有包含代码而被拒绝，除非这对于贡献是核心的（例如，对于一个新的开源基准测试）。
   - 说明中应包含运行以重现结果所需的确切命令和环境。有关详细信息，请参阅 NeurIPS 代码和数据提交指南（ `https://nips.cc/public/guides/CodeSubmissionPolicy` ）。
   - 作者应提供有关数据访问和准备的说明，包括如何访问原始数据、预处理数据、中间数据和生成的数据等。
   - 作者应提供脚本以重现所有新提出的方法和基线的实验结果。如果只有部分实验可以重现，他们应说明哪些实验被省略以及原因。
   - 在提交时，为了保持匿名性，作者应发布匿名版本（如果适用）。
   - 建议在补充材料（附在论文后面）中尽可能提供详细的信息，但允许包含数据和代码的 URL。

7. **Experimental setting/details**

   问题：论文是否详细说明了理解结果所需的所有训练和测试细节（例如，数据划分、超参数、它们的选择方式、优化器类型等）？

   答案: [Yes]

   理由：请参阅 Sec. 6.1 和附录. C 。

   指南：

   - 答案 NA 表示论文不包括实验。
   - 实验设置应在论文的核心部分展示，并达到一个详细程度，以便能够理解和解析结果。
   - 完整的细节可以随代码一起提供，或在附录中提供，或者作为补充材料提供。

8. **Experiment statistical significance**

   问题：论文是否适当地和正确地定义误差棒或其他关于实验统计显著性的信息？

   答案：[No]

   理由：我们没有为端到端实验包含误差线，因为本文包括大量昂贵的实验。我们在不同设置的单次试验中展示结果。

   指南:

   - 答案 NA 表示论文不包含实验。
   - 如果结果附有误差条、置信区间或统计显著性检验，至少在支持论文主要论点的实验中，作者应该回答“是”。
   - 误差条所捕获的变异因素应被清楚地说明（例如，训练/测试的拆分、初始化、某些参数的随机抽取，或在给定实验条件下的总体运行）。
   - 应该解释误差条的计算方法（闭合形式公式、调用库函数、自举法等）。
   - 应该给出所作的假设（例如，误差服从正态分布）。
   - 应该明确误差线是标准差还是平均值的标准误差。
   - 可以报告 1sigma 误差条，但需要说明。作者最好报告 2sigma 误差条，而不是在误差正态性假设没有被验证的情况下说明他们有一个 96 % CI。
   - 对于非对称分布，作者应谨慎避免在表格或图形中展示对称误差线，因为这可能会导致结果超出范围（例如，出现负误差率）。

- 如果在表格或图表中报告了误差条，作者应在文中解释它们是如何计算的，并在文中引用相应的图或表。

9. **Experiments compute resources**

   问题：对于每个实验，论文是否提供了足够的信息关于计算资源（计算工作者的类型、内存、执行时间），以便重现实验?

   答案：[Yes]

   理由：请参阅附录。C 。

   指南：

   - 答案 NA 表示论文不包含实验。
   - 文章应指明计算工作者的类型，CPU 或 GPU，内部集群或云服务提供商，包括相关的内存和存储。
   - 论文应提供每个单独实验运行所需的计算量，并估算总的计算量。
   - 论文应披露整个研究项目是否需要比论文中报道的实验更多的计算（例如，未被纳入论文的初步或失败的实验）。

10. **Code of ethics**

    问题：本论文中进行的研究在各个方面是否符合 NeurIPS 伦理规范 `https://neurips.cc/public/EthicsGuidelines` ?

    答案: [Yes]

    正当性：不适用

    指南：

    - 答案 NA 意味着作者尚未审查 NeurIPS 伦理守则。
    - 如果作者回答“否”，他们应解释需要偏离伦理规范的特殊情况。
    - 作者应确保保持匿名性（例如，如果由于所在司法区域的法律或法规有特殊考虑）。

11. **Broader impacts**

12. 问题：这篇论文是否讨论了所完成工作的潜在正面社会影响和负面社会影响?

    答案: [NA]

    论证：本文研究训练算法，其社会影响有限。

    指南：

    - 回答 NA 意味着所进行的工作没有社会影响。
    - 如果作者回答 NA 或 No，他们应解释为何他们的工作没有社会影响，或者为何论文不涉及社会影响。
    - 负面社会影响的例子包括潜在的恶意或非预期用途（例如，虚假信息、生成假档案、监控），公平性考虑（例如，部署可能对特定群体产生不公正影响的技术），隐私考虑和安全考虑。
    - 会议预期许多论文将是基础研究，并非与特定应用相关，更不用说部署了。然而，如果有任何直接通向负面应用的路径，作者应指出。例如，指出生成模型质量的提升可以被用于生成用于假消息传播的深度伪造是合理的。另一方面，没有必要指出一个用于优化神经网络的通用算法可能使人们能够更快地训练生成深度伪造的模型。
    - 作者应考虑在技术按照预期使用且正常运行时可能产生的危害，在技术按照预期使用但给出错误结果时可能产生的危害，以及技术被（有意或无意地）误用所导致的危害。
    - 如果存在负面的社会影响，作者也可以讨论可能的缓解策略（例如，限制模型的发布，除了攻击之外还提供防御措施，监控滥用的机制，监测系统从反馈中随着时间的学习情况的机制，提高机器学习的效率和可访问性）。

13. **Safeguards**

    问题：论文是否描述了为了负责任地发布具有高误用风险的数据或模型（例如，预训练语言模型、图像生成器或抓取的数据集）而已实施的安全措施?

    答案: [NA]

理由：本文不构成此类风险。

指南：

- 答案 NA 表示论文没有这样的风险。
- 发布具有高滥用或双重使用风险的模型时，应附带必要的保障措施，以允许对模型的受控使用，例如要求用户遵守使用指南或限制访问模型，或实施安全过滤器。
- 从互联网抓取的数据集可能存在安全风险。作者应描述他们如何避免发布不安全的图像。
- 我们认识到提供有效的保障措施是具有挑战性的，尽管许多论文并不要求这样做，但我们鼓励作者考虑这一点并尽最大诚意努力。

14. **Licenses for existing assets**

问题：论文本中使用的资源（例如：代码、数据、模型）的创造者或原始所有者是否得到了适当的致谢，且是否明确提及和合适遵循了其许可和使用条款?

答案：[Yes]

15. 理由：我们确保对现有资产进行适当引用。

指南：

- 答案 NA 意味着论文没有使用现有资产。
- 作者应引用生成代码包或数据集的原始论文。
- 作者应说明所使用的资产版本，并在可能的情况下包含一个 URL。
- 每个资产都应包括许可的名称（例如，CC-BY 4.0）。
- 对于来自特定来源（例如网站）的抓取数据，应提供该来源的版权和服务条款。
- 如果发布资产，则应提供包中的许可证、版权信息和使用条款。对于流行的数据集，`paperswithcode.com/datasets` 已经为一些数据集整理了许可证。他们的许可指南可以帮助确定数据集的许可证。
- 对于重新包装的现有数据集，应提供原始许可和派生资产的许可（如果已更改）。
- 如果这些信息无法在线获取，建议作者联系资产的创建者。

16. **New assets**

问题：论文中引入的新资产是否有充分的文档记录，并且文档是否与资产一起提供?

答案：[Yes]

理由：我们发布了实现 REO-RL 的代码以及我们估计的效率前沿的确切数值。我们还将在未来发布必要的模型。

指南：

- 答案“NA”表示论文未发布新的资产。
- 研究人员应通过结构化模板在提交时传达数据集/代码/模型的详细信息。这包括有关训练、许可证、限制等的详细信息。
- 论文应该讨论是否以及如何获得那些资产被使用的人的同意。
- 在提交时，请记得匿名化您的资源（如果适用）。您可以创建一个匿名的 URL 或包含一个匿名化的 zip 文件。

17. **Crowdsourcing and research with human subjects**

问题：对于众包实验和涉及人类受试者的研究，论文是否包括提供给参与者的完整说明文本和截图（如果适用），以及有关补偿（如果有）的详细信息?

答案: [NA]

理由：该论文不涉及众包或者以人为研究对象的研究

指导方针：

- 答案 NA 表示论文不涉及众包或涉及人类受试者的研究。
- 将此信息包含在补充材料中是可以的，但如果论文的主要贡献涉及人类受试者，则应在主论文中尽可能多地包含详细信息。
- 根据 NeurIPS 伦理守则，参与数据收集、整理或其他工作的人员应至少获得数据收集者所在国家的最低工资。

18. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

19. 问题：论文是否描述了研究参与者可能遇到的风险，这些风险是否已告知参与者，以及是否获得了机构审查委员会（IRB）批准（或根据您所在国家或机构的要求获得的同等批准/审查）？

    答案: [NA]

    理由：本文不涉及众包或以人为研究对象的研究

    指导方针：

    - 答案 NA 意味着本文不涉及众包或关于人类受试者的研究。
    - 根据研究进行的国家，任何涉及人类受试者的研究可能需要获得 IRB 批准（或同等批准）。如果您获得了 IRB 批准，您应在论文中明确说明这一点。
    - 我们认识到，不同机构和地区的程序可能会有很大差异，我们期望作者遵守 NeurIPS 伦理规范以及其机构的指导方针。
    - 对于最初的提交，不要包含任何会破坏匿名性的的信息（如果适用），例如进行审查的机构。

20. **Declaration of LLM usage**

21. 问题：如果 LLM 是该研究核心方法中一个重要的、原创的或非标准的组成部分，论文是否描述了 LLM 的使用？请注意，如果 LLM 仅用于写作、编辑或格式化，并不影响研究的核心方法、科学严谨性或原创性，则不需要声明。

    答案: [NA]

    理由：这项工作没有使用 LLMs 作为任何重要、原创或非标准的组成部分

    指南：

    - 答案“NA”表示，本研究的核心方法开发不涉及将 LLMs 作为任何重要、原创或非标准的组成部分。
    - 有关应描述或不应描述的内容，请参阅我们的 LLM 政策（`https://neurips.cc/Conferences/2025/LLM` ）。