Demystifying Long Chain-of-Thought Reasoning in LLMs

Edward Yeo [*1] **Yuxuan Tong** [*2] **Morry Niu** [1] **Graham Neubig** [3] **Xiang Yue** [*3]

Abstract

Scaling inference compute enhances reasoning in large language models (LLMs), with long chains-of-thought (CoTs) enabling strategies like backtracking and error correction. Reinforcement learning (RL) has emerged as a crucial method for developing these capabilities, yet the conditions under which long CoTs emerge remain unclear, and RL training requires careful design choices. In this study, we systematically investigate the mechanics of long CoT reasoning, identifying the key factors that enable models to generate long CoT trajectories. Through extensive supervised fine-tuning (SFT) and RL experiments, we present four main findings: (1) While SFT is not strictly necessary, it simplifies training and improves efficiency; (2) Reasoning capabilities tend to emerge with increased training compute, but their development is not guaranteed, making reward shaping crucial for stabilizing CoT length growth; (3) Scaling verifiable reward signals is critical for RL. We find that leveraging noisy, web-extracted solutions with filtering mechanisms shows strong potential, particularly for out-of-distribution (OOD) tasks such as STEM reasoning; and (4) Core abilities like error correction are inherently present in base models, but incentivizing these skills effectively for complex tasks via RL demands significant compute, and measuring their emergence requires a nuanced approach. These insights provide practical guidance for optimizing training strategies to enhance long CoT reasoning in LLMs. Our code is available at: https://github.com/eddycmu/demystify-long-cot.

1. Introduction

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Anthropic, 2023; OpenAI, 2023) have demonstrated remarkable reasoning abilities in domains like mathematics (Cobbe et al., 2021) and programming (Chen et al., 2021). A key technique for enabling reasoning abilities in LLMs is chain-of-thought (CoT) prompting (Wei et al., 2022), which guides models to generate intermediate reasoning steps before arriving at a final answer.

Despite these advancements, LLMs still struggle with highly complex reasoning tasks, such as mathematical competitions (Hendrycks et al., 2021), PhD-level scientific QA (Rein et al., 2024), and software engineering (Jimenez et al., 2024), even with CoT. Recently, OpenAI's o1 models (OpenAI, 2024) have demonstrated significant breakthroughs in these tasks. A key distinguishing feature of these models is their ability to scale up inference compute with long CoTs, which include strategies such as recognizing and correcting mistakes, breaking down difficult steps, and iterating on alternative approaches, leading to substantially longer and more

[*]Project Lead. [1]IN.AI [2]Tsinghua University. Work started when interning at CMU. [3]Carnegie Mellon University. Correspondence to: Xiang Yue <xyue2@andrew.cmu.edu>.

揭秘大语言模型中的长链推理

Edward Yeo [*1] **Yuxuan Tong** [*2] **Morry Niu** [1] **Graham Neubig** [3] **Xiang Yue** [*3]

Abstract

*警告：该PDF由GPT-Academic开源项目调用大语言模型+Latex翻译插件一键生成，版权归原文作者所有。翻译内容可靠性无保障，请仔细鉴别并以原文为准。项目Github地址 https://github.com/binary-husky/gpt_academic/。项目在线体验地址 https://auth.gpt-academic.top/。当前大语言模型：Qwen2.5-72B-Instruct，当前语言模型温度设定: 0.3。为了防止大语言模型的意外谬误产生扩散影响，禁止移除或修改此警告。

扩展推理计算可以增强大型语言模型（LLMs）的推理能力，长链思考（CoTs）能够实现诸如回溯和错误纠正等策略。强化学习（RL）已经成为开发这些能力的关键方法，但长CoTs出现的条件仍然不清楚，RL训练需要精心的设计选择。在本研究中，我们系统地调查了长CoT推理的机制，确定了使模型能够生成长CoT轨迹的关键因素。通过广泛的监督微调（SFT）和RL实验，我们提出了四个主要发现：（1）虽然SFT不是严格必要的，但它简化了训练并提高了效率；（2）推理能力往往随着训练计算的增加而出现，但其发展不是必然的，奖励塑形对于稳定CoT长度的增长至关重要；（3）可验证的奖励信号的扩展对于RL至关重要。我们发现，利用带有过滤机制的噪声、网络提取的解决方案显示出强大的潜力，特别是在处理分布外（OOD）任务如STEM推理方面；（4）核心能力如错误纠正在基础模型中是固有的，但通过RL有效激励这些技能以应对复杂任务需要大量的计算，测量其出现需要细致的方法。这些见解为优化训练策略以增强LLMs中的长CoT推理提供了实用的指导。我们的代码可在以下地址获取：https://github.com/eddycmu/demystify-long-cot。

1. Introduction

大型语言模型（LLMs）(Brown et al., 2020; Touvron et al., 2023; Anthropic, 2023; OpenAI, 2023) 在数学 (Cobbe et al., 2021) 和编程 (Chen et al., 2021) 等领域展示了显著的推理能力。使 LLMs 获得推理能力的一个关键技术是链式思维（CoT）提示 (Wei et al., 2022)，它引导模型在得出最终答案之前生成中间推理步骤。

尽管取得了这些进展，LLMs 仍然在处理高度复杂的推理任务时遇到困难，例如数学竞赛 (Hendrycks et al., 2021)、博士级别的科学问答 (Rein et al., 2024) 和软件工程 (Jimenez et al., 2024)，即使使用了 CoT。最近，OpenAI 的 o1 模型 (OpenAI, 2024) 在这些任务中展示了显著的突破。这些模型的一个关键区别特征是它们能够通过长 CoT 扩展推理计算，其中包括识别和纠正错误、分解困难步骤以及迭代替代方法等策略，从而导致更长且更有结构的推理过程。

多项努力试图通过训练 LLMs 生成长 CoT 来复制 o1 模型的性能 (Qwen Team, 2024b; DeepSeek-AI, 2025; Kimi Team, 2025; Pan et al., 2025; Zeng et al., 2025)。大多数这些方法依赖于可验证的奖励，例如基于真

[*]Project Lead. [1]IN.AI [2]清华大学。工作始于在卡内基梅隆大学实习期间。[3]卡内基梅隆大学. Correspondence to: Xiang Yue <xyue2@andrew.cmu.edu>.

structured reasoning processes.

Several efforts have attempted to replicate the performance of o1 models by training LLMs to generate long CoTs (Qwen Team, 2024b; DeepSeek-AI, 2025; Kimi Team, 2025; Pan et al., 2025; Zeng et al., 2025). Most of these approaches rely on verifiable rewards, such as accuracy based on ground-truth answers, which helps to avoid reward hacking in reinforcement learning (RL) at scale. However, a comprehensive understanding of how models learn and generate long CoTs remains limited. In this work, we systematically investigate the underlying mechanics of long CoT generation. Specifically, we explore:

1) *Supervised fine-tuning (SFT) for long CoTs* – the most direct way to enable long CoT reasoning. We analyze its scaling behavior and impact on RL, finding that long CoT SFT allows models to reach higher performance and also facilitates easier RL improvements than short CoT.

2) *Challenges in RL-driven CoT scaling* – we observe that RL does not always stably extend CoT length and complexity. To address this, we introduce a cosine length-scaling reward with a repetition penalty, which stabilizes CoT growth while encouraging emergent reasoning behaviors such as branching and backtracking.

3) *Scaling up verifiable signals for long CoT RL* – Verifiable reward signals are essential for stabilizing long CoT RL. However, scaling them up remains challenging due to the limited availability of high-quality, verifiable data. To address this, we explore the use of data containing noisy, web-extracted solutions (Yue et al., 2024b). While these "silver" supervision signals introduce uncertainty, we find that, with an appropriate mixture in SFT and filtration in RL, they show promise, especially in out-of-distribution (OOD) reasoning scenarios such as STEM problem-solving.

4) *Origins of Long CoT Abilities and RL Challenges* – Core skills like branching and error validation are inherently present in base models, but effective RL-driven incentivization demands careful designs. We examine RL incentives on long CoT generation, trace reasoning patterns in pre-training data, and discuss nuances in measuring their emergence.

2. Problem Formulation

In this section, we define the notation, followed by an overview of SFT and RL methods for eliciting long CoTs.

Research Aim

Our goal is to *demystify long chain-of-thought reasoning* in LLMs. Through systematic analysis and ablations, we extract key insights and offer practical strategies to enhance and stabilize its performance.

2.1. Notation

Let x be a query, and let y be the output sequence. We consider a LLM parameterized by θ, which defines a conditional distribution over output tokens: $\pi_\theta(y_t \mid x, y_{1:t-1})$.

We denote by $\text{CoT}(y) \subseteq y$ the tokens in the generated output that constitute the *chain-of-thought*, which is often a reasoning trace or explanatory sequence. The final "answer" can be a separate set of tokens or simply the last part of y.

In this work, we use the term *long chain-of-thought* (*long CoT*) to describe an extended sequence of reasoning tokens that not only exhibits a larger-than-usual token length but also demonstrates more sophisticated behaviors such as:

1) Branching and Backtracking: The model systematically explores multiple paths (branching) and reverts to earlier points if a particular path proves wrong (backtracking).

2) Error Validation and Correction: The model detects inconsistencies or mistakes in its intermediate steps and takes corrective actions to restore coherence and

实答案的准确性，这有助于避免在大规模强化学习（RL）中的奖励欺骗。然而，对模型如何学习和生成长 CoT 的全面理解仍然有限。在这项工作中，我们系统地研究了长 CoT 生成的底层机制。具体来说，我们探讨了以下几点：

1) *用于长 CoT 的监督微调（SFT）* – 使长 CoT 推理成为可能的最直接方法。我们分析了其扩展行为及其对 RL 的影响，发现长 CoT SFT 使模型能够达到更高的性能，并且比短 CoT 更容易进行 RL 改进。

2) *RL 驱动的 CoT 扩展挑战* – 我们观察到 RL 并不总是稳定地扩展 CoT 的长度和复杂性。为了解决这个问题，我们引入了一个带有重复惩罚的余弦长度扩展奖励，这在鼓励分支和回溯等新兴推理行为的同时稳定了 CoT 的增长。

3) *扩展用于长 CoT RL 的可验证信号* – 可验证的奖励信号对于稳定长 CoT RL 至关重要。然而，由于高质量、可验证数据的可用性有限，扩展它们仍然具有挑战性。为了解决这个问题，我们探索了使用包含噪声的、从网络提取的解决方案的数据 (Yue et al., 2024b)。虽然这些"银色"监督信号引入了不确定性，但我们发现，通过在 SFT 中适当混合和在 RL 中过滤，它们在处理分布外（OOD）推理场景（如 STEM 问题解决）时显示出潜力。

4) *长 CoT 能力的起源和 RL 挑战* – 分支和错误验证等核心技能在基础模型中固有存在，但有效的 RL 驱动激励需要精心设计。我们研究了长 CoT 生成中的 RL 激励，追踪了预训练数据中的推理模式，并讨论了衡量其出现的细微差别。

2. Problem Formulation

在本节中，我们首先定义符号，然后概述用于引出长链思维（CoTs）的SFT和RL方法。

Research Aim

Our goal is to *demystify long chain-of-thought reasoning* in LLMs. Through systematic analysis and ablations, we extract key insights and offer practical strategies to enhance and stabilize its performance.

2.1. Notation

令 x 为查询，y 为输出序列。我们考虑一个由 θ 参数化的大型语言模型（LLM），它定义了输出标记的条件分布：$\pi_\theta(y_t \mid x, y_{1:t-1})$.

我们用 $\text{CoT}(y) \subseteq y$ 表示生成输出中构成 *思维链* 的标记，这通常是一个推理轨迹或解释序列。最终的"答案"可以是单独的一组标记，也可以仅仅是 y 的最后一部分。

在本工作中，我们使用 *长思维链* (*long CoT*) 一词来描述一个扩展的推理标记序列，它不仅表现出比平常更长的标记长度，还展示了更复杂的行为，例如：

1) 分支和回溯：模型系统地探索多条路径（分支），并在特定路径被证明错误时返回到早期点（回溯）。

2) 错误验证和纠正：模型检测其中间步骤中的不一致或错误，并采取纠正措施以恢复连贯性和准确性。

2.2. Supervised Fine-Tuning (SFT)

一种常见的做法是通过在数据集 $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i)\}_{i=1}^{N}$ 上使用 SFT (Lamb et al., 2016) 来初始化策略 π_θ，其中 y_i 可以是正常的或长的 CoT 推理标记。

2.3. Reinforcement Learning (RL)

在可选的SFT初始化之后，我们可以进一步使用强化学习优化长CoT的生成。

奖励函数。 我们定义一个标量奖励 r_t，旨在鼓励正确且可验证的推理。我们仅考虑最终答案的基于结

accuracy.

2.2. Supervised Fine-Tuning (SFT)

A common practice is to initialize the policy π_θ via SFT (Lamb et al., 2016) on a dataset $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i)\}_{i=1}^{N}$, where y_i can be normal or long CoT reasoning tokens.

2.3. Reinforcement Learning (RL)

After optional SFT initialization, we can further optimize the generation of long CoT with reinforcement learning.

Reward Function. We define a scalar reward r_t designed to encourage correct and verifiable reasoning. We only consider the outcome-based reward for the final answer produced, and do not consider process-based reward for the intermediate steps. We denote the term $r_{\text{answer}}(y)$ to capture the correctness of the final solution.

Policy Update. We adopted Proximal Policy Optimization (PPO) (Schulman et al., 2017) as the default policy optimization method in our experiments. We also briefly discuss REINFORCE (Sutton & Barto, 2018) method in subsection 7.3. We adopt a rule-based verifier as the reward function, which compares the predicted answer with the ground truth answer directly. The resulting updates push the policy to generate tokens that yield higher reward.

2.4. Training Setup

We adopt `Llama-3.1-8B` (Meta, 2024) and `Qwen2.5-7B-Math` (Qwen Team, 2024a) as the base models, which are representative general and math-specialized models respectively. For both SFT and RL, we use the 7,500 training sample prompt set of MATH (Hendrycks et al., 2021) by default, with which verifiable ground truth answers are provided. For SFT when ground truth answers are available, we synthesize responses by rejection sampling (Zelikman et al., 2022; Dong et al., 2023; Yuan et al., 2023; Gulcehre et al., 2023; Singh et al., 2023; Yue et al., 2024a; Tong et al., 2024). Specifically, we first sample a fixed number N of candidate responses per prompt and then filter by only retaining ones with final answers consistent with the corresponding ground truth answers. We also discuss data like WebInstruct (Yue et al., 2024b) that is more diverse but without gold supervision signals like ground truth answers in §5. We train the models with the OpenRLHF framework (Hu et al., 2024).

2.5. Evaluation Setup

We focus on four representative reasoning benchmarks: MATH-500, AIME 2024, TheoremQA (Chen et al., 2023), and MMLU-Pro-1k (Wang et al., 2024a). Given that our training data is primarily in the mathematical domain, these benchmarks provide a comprehensive framework for both in-domain (MATH-500 test set) and out-of-domain evaluations (AIME 2024, TheoremQA, MMLU-Pro-1k). By default, we generate from the models using a temperature of $t = 0.7$, a top-p value of 0.95, and a maximum output length of 16,384 tokens. Please refer to Appendix E.1 for further details on the evaluation setup.

3. Impact of SFT on Long CoT

In this section, we compare long and short CoT data for SFT and in the context of RL initialization.

3.1. SFT Scaling

To compare long CoT with short CoT, the first step is to equip the model with the corresponding behavior. The most straightforward approach is to fine-tune the base model on CoT data. Since short CoT is common, curating SFT data for it is relatively simple via rejection sampling from existing models. However, how to obtain high-quality long CoT data remains an open question.

Setup. To curate the SFT data, for long CoT, we distill from `QwQ-32B-Preview` (we discuss other long CoT data construction methods in §3.3). For short CoT, we distill from `Qwen2.5-Math-72B-Instruct`,

果的奖励，而不考虑中间步骤的基于过程的奖励。我们用 $r_{\text{answer}}(y)$ 表示最终解决方案的正确性。

策略更新。 在我们的实验中，我们采用了近端策略优化（PPO）(Schulman et al., 2017) 作为默认的策略优化方法。我们也在 subsection 7.3 中简要讨论了 REINFORCE (Sutton & Barto, 2018) 方法。我们采用基于规则的验证器作为奖励函数，该验证器直接将预测答案与真实答案进行比较。由此产生的更新推动策略生成能够获得更高奖励的标记。

2.4. Training Setup

我们采用 `Llama-3.1-8B` (Meta, 2024) 和 `Qwen2.5-7B-Math` (Qwen Team, 2024a) 作为基础模型，它们分别是代表性的通用模型和数学专业模型。对于 SFT 和 RL，我们默认使用 MATH (Hendrycks et al., 2021) 的 7,500 个训练样本提示集，这些样本提供了可验证的正确答案。在 SFT 中，当有正确答案可用时，我们通过拒绝采样 (Zelikman et al., 2022; Dong et al., 2023; Yuan et al., 2023; Gulcehre et al., 2023; Singh et al., 2023; Yue et al., 2024a; Tong et al., 2024) 合成响应。具体来说，我们首先为每个提示采样固定数量 N 的候选响应，然后仅保留最终答案与相应正确答案一致的响应。我们还在 §5 中讨论了类似 WebInstruct (Yue et al., 2024b) 的数据，这些数据更加多样化，但没有像正确答案这样的金标准监督信号。我们使用 OpenRLHF 框架 (Hu et al., 2024) 训练模型。

2.5. Evaluation Setup

我们关注四个具有代表性的推理基准测试：MATH-500、AIME 2024、TheoremQA (Chen et al., 2023) 和 MMLU-Pro-1k (Wang et al., 2024a)。鉴于我们的训练数据主要集中在数学领域，这些基准测试为领域内（MATH-500 测试集）和领域外评估（AIME 2024、TheoremQA、MMLU-Pro-1k）提供了一个全面的框架。默认情况下，我们使用温度 $t = 0.7$、top-p 值为 0.95 和最大输出长度为 16,384 个 token 从模型生成结果。有关评估设置的更多详细信息，请参阅附录 E.1。

3. Impact of SFT on Long CoT

在本节中，我们比较了SFT和RL初始化背景下长链和短链的CoT数据。

3.1. SFT Scaling

为了比较长链思维（CoT）与短链思维（CoT），第一步是使模型具备相应的行为。最直接的方法是在CoT数据上微调基础模型。由于短链思维较为常见，通过从现有模型中使用拒绝采样来整理SFT数据相对简单。然而，如何获得高质量的长链思维数据仍然是一个开放的问题。

设置。 为了整理SFT数据，对于长链思维，我们从`QwQ-32B-Preview`中提取（我们在§3.3中讨论了其他长链思维数据构建方法）。对于短链思维，我们从`Qwen2.5-Math-72B-Instruct`中提取，这是一个在数学推理方面表现良好的短链思维模型。具体来说，我们通过首先为每个提示采样N个候选响应，然后筛选出答案正确的响应来进行拒绝采样。对于长链思维，我们使用$N \in \{32, 64, 128, 192, 256\}$，而对于短链思维，我们使用$N \in \{32, 64, 128, 256\}$，为了效率跳过一个$N$。在每种情况下，SFT标记的数量与$N$成正比。我们使用基础模型`Llama-3.1-8B` (Meta, 2024)。关于SFT设置的更多详细信息，请参见附录E.3。

结果。 图1中的虚线表明，随着SFT标记的增加，长链思维SFT继续提高模型的准确性，而短链思维SFT则在较低的准确性水平上早期饱和。例如，在MATH-500上，长链思维SFT实现了超过70%的准确性，并且即使在3.5B标记时仍未达到平台期。相比之下，短链思维SFT在低于55%的准确性水平上收敛，SFT标记从大约0.25B增加到1.5B仅带来了约3%的绝对改进。

要点 3.1 SFT 扩展上限

具有长 CoT 的 SFT 可以扩展到比短 CoT 更高的性能上限。（图 1）

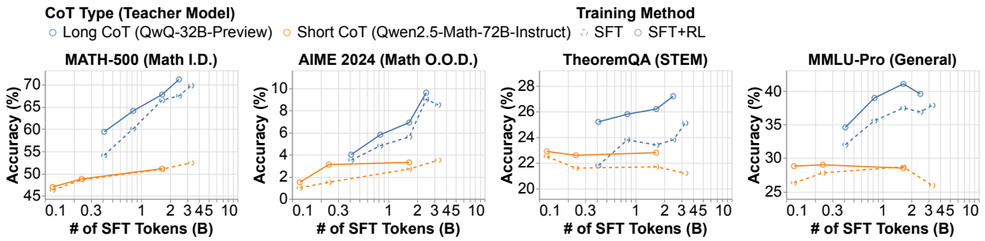


Figure 1. Scaling curves of SFT and RL on `Llama-3.1-8B` with long CoTs and short CoTs. SFT with long CoTs can scale up to a higher upper limit and has more potential to further improve with RL.

which is a capable short CoT model in math reasoning. Specifically, we perform rejection sampling by first sampling N candidate responses per prompt and then filtering for ones with correct answers. For long CoT, we use $N \in \{32, 64, 128, 192, 256\}$, while for short CoT, we use $N \in \{32, 64, 128, 256\}$, skipping one N for efficiency. In each case, the number of SFT tokens is proportional to N. We use the base model `Llama-3.1-8B` (Meta, 2024). Please refer to Appendix E.3 for more details about the SFT setup.

Result. The dashed lines in Figure 1 illustrate that as we scale up the SFT tokens, long CoT SFT continues to improve model accuracy, whereas short CoT SFT saturates early at a lower accuracy level. For instance, on MATH-500, long CoT SFT achieves over 70% accuracy and has yet to plateau even at 3.5B tokens. In contrast, short CoT SFT converges below 55% accuracy, with an increase in SFT tokens from approximately 0.25B to 1.5B yielding only a marginal absolute improvement of about 3%.

Takeaway 3.1 for SFT Scaling Upper Limit

SFT with long CoT can scale up to a higher performance upper limit than short CoT. (Figure 1)

3.2. SFT Initialization for RL

Since RL is reported to have a higher upper limit than SFT, we compare long CoT and short CoT as different SFT initialization approaches for RL.

Setup. We initialize RL using SFT checkpoints from §3.1, and train for four epochs, sampling four responses per prompt. Our approach employs PPO (Schulman et al., 2017) with a rule-based verifier from the MATH dataset, using its training split as our RL prompt set. We adopt our cosine length scaling reward with the repetition penalty, which will be detailed in §4. Further details about our RL setup and hyperparameters can be found in Appendix E.4 & E.5.1 respectively.

Result. The gap between solid and dashed lines in Figure 1 shows that models initialized with long CoT SFT can usually be further significantly improved by RL, while models initialized with short CoT SFT see little gains from RL. For example, on MATH-500, RL can improve long CoT SFT models by over 3% absolute, while short CoT SFT models have almost the same accuracies before and after RL.

Takeaway 3.2 for SFT Initialization for RL

SFT with long CoTs makes further RL improvement easier, while short CoTs do not. (Figure 1)

3.3. Sources of Long CoT SFT Data

To curate long CoT data, we compare two approaches: (1) **Construct** long CoT trajectories by prompting short CoT models to generate primitive actions and sequentially combining them; (2) **Distill** long CoT trajectories from existing long CoT models that exhibit emergent long CoT patterns.

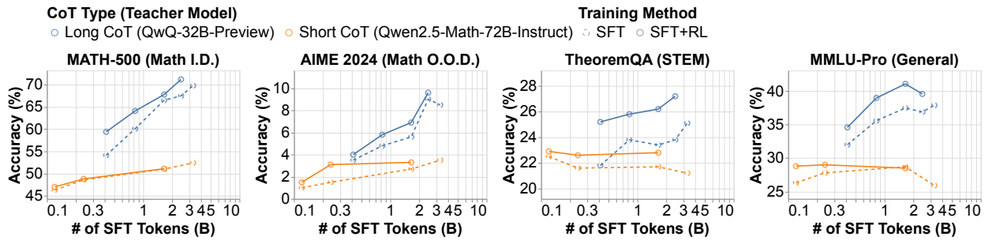


Figure 1. 在 `Llama-3.1-8B` 上，使用长链和短链的 SFT 和 RL 的扩展曲线。使用长链的 SFT 可以扩展到更高的上限，并且有更大的潜力通过 RL 进一步改进。

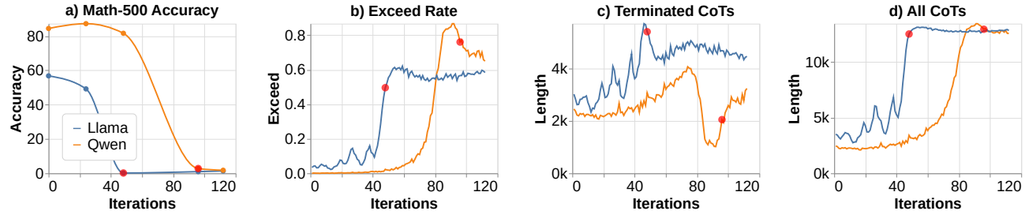


Figure 2. `Llama3.1-8B` 和 `Qwen2.5-Math-7B` 模型在使用经典奖励进行强化学习训练时，出现了超越上下文窗口大小的 CoT 长度扩展，导致 MATH-500 准确率下降。图表中的红点对应准确率降至接近零的迭代点。“Terminated CoTs” 指的是在上下文长度内结束的响应。

3.2. SFT Initialization for RL

由于据报道RL的上限比SFT更高，我们将长CoT和短CoT作为RL的不同SFT初始化方法进行比较。

设置. 我们使用§3.1中的SFT检查点初始化RL，并训练四个epoch，每个提示采样四个响应。我们的方法采用PPO (Schulman et al., 2017)和来自MATH数据集的基于规则的验证器，使用其训练集作为我们的RL提示集。我们采用了带有重复惩罚的余弦长度缩放奖励，这将在§4中详细说明。关于我们的RL设置和超参数的更多细节分别可以在附录E.4和E.5.1中找到。

结果. 图1中实线和虚线之间的差距表明，使用长CoT SFT初始化的模型通常可以通过RL显著改进，而使用短CoT SFT初始化的模型从RL中获益甚微。例如，在MATH-500上，RL可以将长CoT SFT模型的性能提高超过3%的绝对值，而短CoT SFT模型在RL前后几乎具有相同的准确性。

要点 3.2 对于用于 RL 的 SFT 初始化

带有长 CoTs 的 SFT 使进一步的 RL 改进变得更加容易，而短 CoTs 则不然。(图 1)

3.3. Sources of Long CoT SFT Data

为了整理长链思考（CoT）数据，我们比较了两种方法：(1) **构建**长链思考轨迹，通过提示短链思考模型生成基本动作并依次组合它们；(2) **提炼**长链思考轨迹，从展示出新兴长链思考模式的现有长链思考模型中提炼。

设置。 为了构建长链思考轨迹，我们开发了一个动作提示框架（附录 E.8），定义了以下基本动作：`clarify`，`decompose`，`solution_step`，`reflection` 和 `answer`。我们使用多步提示与短链思考模型（例如，`Qwen2.5-72B-Instruct`）来序列化这些动作，而更强的模型 `o1-mini-0912` 生成包含自我修正的反思步骤。对于提炼长链思考轨迹，我们使用 `QwQ-32-Preview` 作为教师模型。在这两种方法中，我们都采用了 MATH 训练集作为提示集，并应

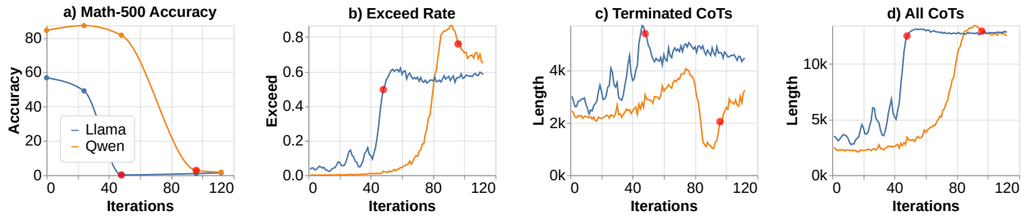


Figure 2. Both `Llama3.1-8B` and `Qwen2.5-Math-7B` models trained under RL with the Classic Reward manifested emergent CoT length scaling past the context window size, resulting in a decline in MATH-500 accuracy. The red points on the charts correspond to the iteration where the accuracy dropped to near zero. "Terminated CoTs" refer to responses that conclude within the context length.

Setup. To construct long CoT trajectories, we developed an Action Prompting framework (Appendix E.8) which defined the following primitive actions: `clarify`, `decompose`, `solution_step`, `reflection`, and `answer`. We employed multi-step prompting with a short CoT model (e.g., `Qwen2.5-72B-Instruct`) to sequence these actions, while a stronger model, `o1-mini-0912`, generates reflection steps incorporating self-correction. For distilling long CoT trajectories, we use `QwQ-32-Preview` as the teacher model. In both approaches, we adopt the MATH training set as the prompt set and apply rejection sampling. To ensure fairness, we use the same base model (`Llama-3.1-8B`), maintain approximately 200k SFT samples, and use the same RL setup as in §3.2.

Result. Table 1 shows that the model distilled from emergent long CoT patterns generalizes better than the constructed pattern, and can be further significantly improved with RL, while the model trained on constructed patterns cannot. Models trained with the emergent long CoT pattern achieve significantly higher accuracies on OOD benchmarks AIME 2024 and MMLU-Pro-1k, improving by 15-50% relatively. Besides, on the OOD benchmark TheoremQA, RL on the long CoT SFT model significantly improves its accuracy by around 20% relative, while the short CoT model's performance does not change. This is also why we conduct most of our experiments based on distilled long CoT trajectories.

Takeaway 3.3 for Long CoT Cold Start

SFT initialization matters: high-quality, emergent long CoT patterns lead to significantly better generalization and RL gains. (Table 1)

Table 1. Emergent long CoT patterns outperform constructed ones. All the models here are fine-tuned from the base model `Llama-3.1-8B` with the MATH training prompt set.

Training Method	Long CoT SFT Pattern	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k
SFT	Constructed	48.2	2.9	21.0	18.1
	Emergent	**54.1**	**3.5**	**21.8**	**32.0**
SFT+RL	Constructed	52.4	2.7	21.0	19.2
	Emergent	**59.4**	**4.0**	**25.2**	**34.6**

4. Impact of Reward Design on Long CoT

This section examines reward function design, with a focus on its influence on CoT length and model performance.

4.1. CoT Length Stability

Recent studies on long CoT (DeepSeek-AI, 2025; Kimi Team, 2025; Hou et al., 2025) suggest that models naturally improve in reasoning tasks with increased thinking time. Our experiments confirm that models fine-tuned on long CoT distilled from `QwQ-32B-Preview` tend to extend CoT length under RL training, albeit sometimes unstably. This instability, also noted by Kimi Team (2025); Hou et al. (2025), has been addressed using techniques based on length and repetition penalties to stabi-

用了拒绝采样。为了确保公平性，我们使用了相同的基模型（`Llama-3.1-8B`），保持了大约 200k 的 SFT 样本，并使用了与 §3.2 中相同的 RL 设置。

结果。 表 1 显示，从新兴长链思考模式提炼的模型比构建的模式具有更好的泛化能力，并且可以通过 RL 进一步显著提升，而基于构建模式训练的模型则不能。使用新兴长链思考模式训练的模型在 OOD 基准 AIME 2024 和 MMLU-Pro-1k 上的准确率显著提高，相对提高了 15-50%。此外，在 OOD 基准 TheoremQA 上，RL 在长链思考 SFT 模型上的准确率显著提高了约 20% 相对，而短链思考模型的性能没有变化。这也是为什么我们大多数实验都是基于提炼的长链思考轨迹进行的。

要点 3.3 长 CoT 冷启动

SFT 初始化很重要：高质量、新兴的长 CoT 模式显著提高了泛化能力和 RL 收益。（表 1）

Table 1. 出现的长CoT模式优于构建的模式。这里的所有模型都是从基础模型`Llama-3.1-8B`微调而来的，并使用了MATH训练提示集。

Training Method	Long CoT SFT Pattern	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k
SFT	Constructed	48.2	2.9	21.0	18.1
	Emergent	**54.1**	**3.5**	**21.8**	**32.0**
SFT+RL	Constructed	52.4	2.7	21.0	19.2
	Emergent	**59.4**	**4.0**	**25.2**	**34.6**

4. Impact of Reward Design on Long CoT

本节考察奖励函数设计，重点关注其对CoT长度和模型性能的影响。

4.1. CoT Length Stability

最近关于长链思考（CoT）的研究 (DeepSeek-AI, 2025; Kimi Team, 2025; Hou et al., 2025) 表明，模型在增加思考时间后，其推理任务的性能会自然提升。我们的实验确认，经过长链思考数据精调的模型在强化学习（RL）训练下倾向于延长链思考的长度，尽管有时会不稳定。这种不稳定性也被 Kimi

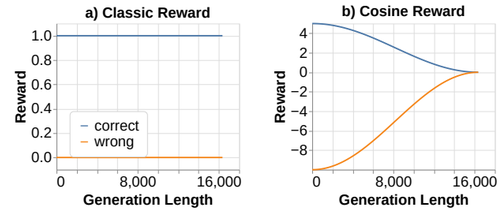


Figure 3. 经典奖励函数和余弦奖励函数。余弦奖励随生成长度变化。

Team (2025); Hou et al. (2025) 注意到，并通过基于长度和重复惩罚的技术来稳定训练。

设置。 我们使用了两种不同的模型，这些模型使用 MATH 训练集从 `QwQ-32B-Preview` 中提取的长链思考数据进行了精调，上下文窗口大小为 16K。这些模型是 `Llama3.1-8B` 和 `Qwen2.5-Math-7B`。我们使用了一个基于规则的验证器和一个简单的正确答案奖励 1。我们将这称为 *经典奖励*。更多详细信息可以在附录 E.5.2 中找到。

结果。 我们观察到，两个模型在训练过程中都增加了链思考的长度，最终达到了上下文窗口的限制。这导致了由于链思考超出允许的窗口大小而训练准确率下降。此外，不同的基础模型表现出不同的扩展行为。较弱的 `Llama-3.1-8B` 模型在链思考长度上的波动比 `Qwen-2.5-Math-7B` 更大，如图 2 所示。

我们还发现，链思考超出上下文窗口大小的速率在某个低于 1 的阈值处趋于平稳（图 2）。这表明，超出限制开始对链思考长度分布产生显著的向下压力，并突显了上下文窗口大小在隐式长度惩罚中的作用。值得注意的是，即使没有明确的超出长度惩罚，由于奖励或优势归一化，轨迹也可能受到惩罚，这两者在强化学习框架中都是标准做法。

要点 4.1 关于 CoT 长度稳定性

CoT 长度并不总是以稳定的方式增加。 (图 2)

lize training.

Setup. We used two different models fine-tuned on long CoT data distilled from `QwQ-32B-Preview` using the MATH train split, with a context window size of 16K. The models were `Llama3.1-8B` and `Qwen2.5-Math-7B`. We used a rule-based verifier along and a simple reward of 1 for correct answers. We shall refer to this as the *Classic Reward*. More details can be found in Appendix E.5.2.

Results. We observed that both models increased their CoT length during training, eventually reaching the context window limit. This led to a decline in training accuracy due to CoTs exceeding the allowable window size. Additionally, different base models exhibited distinct scaling behaviors. The weaker `Llama-3.1-8B` model showed greater fluctuations in CoT length compared to `Qwen-2.5-Math-7B`, as illustrated in Figure 2.

We also found that the rate at which CoTs exceeded the context window size leveled off at a certain threshold below 1 (Figure 2). This suggests that exceeding the limit started to apply significant downward pressure on the CoT length distribution, and highlights the context window size's role in implicit length penalization. Notably, a trajectory might be penalized even without an explicit exceed-length penalty due to reward or advantage normalization, both of which are standard in RL frameworks.

Takeaway 4.1 for CoT Length Stability

CoT length does not always scale up in a stable fashion. (Figure 2)

4.2. Active Scaling of CoT Length

We found that reward shaping can be used to stabilize emergent length scaling. We designed a reward function to use CoT length as an additional input and to observe a few ordering constraints. Firstly, correct CoTs receive higher rewards than wrong CoTs. Secondly, shorter correct CoTs receive higher rewards than longer correct CoTs, which incentivizes the model to use inference compute efficiently. Thirdly, shorter wrong CoTs should receive higher penalties than longer wrong CoTs. This encourages the model to extend its thinking time if it is less likely to get the correct answer.

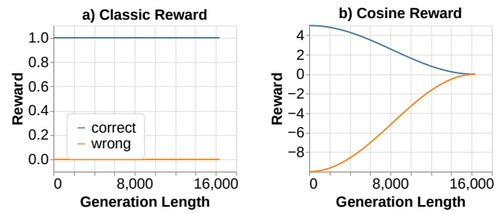


Figure 3. The Classic and Cosine Reward functions. The Cosine Reward varies with generation length.

We found it convenient to use a piecewise cosine function, which is easy to tune and smooth. We refer to this reward function as the *Cosine Reward*, visualized in Figure 3. This is a *sparse* reward, only awarded once at the end of the CoT based on the correctness of the answer. The formula of **CosFn** can be found in equation 1 in the appendix.

$$R(C, L_{\text{gen}}) = \begin{cases} \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^c, r_L^c), & \text{if } C = 1, \\ \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^w, r_L^w), & \text{if } C = 0, \\ r_e, & \text{if } L_{\text{gen}} = L_{\max}. \end{cases}$$

Hyperparameters:

r_0^c / r_0^w : Reward (correct/wrong) for $L_{\text{gen}} = 0$,

r_L^c / r_L^w : Reward (correct/wrong) for $L_{\text{gen}} = L_{\max}$,

r_e : Exceed length penalty,

Inputs:

C : Correctness (0 or 1),

L_{gen} : Generation length.

Setup. We ran experiments with the Classic Reward and the Cosine Reward. We used the `Llama3.1-8B` fine-tuned on long CoT data distilled from `QwQ-32B-Preview` using the MATH train split, as our starting point. For more details, see Appendix E.5.3.

4.2. Active Scaling of CoT Length

我们发现奖励塑形可以用于稳定出现的长度缩放。我们设计了一个奖励函数，使用CoT长度作为附加输入，并观察几个排序约束。首先，正确的CoTs比错误的CoTs获得更高的奖励。其次，较短的正确CoTs比较长的正确CoTs获得更高的奖励，这激励模型高效地使用推理计算。第三，较短的错误CoTs应比较长的错误CoTs获得更高的惩罚。这鼓励模型在不太可能得到正确答案时延长其思考时间。

我们发现使用分段余弦函数很方便，因为它易于调整且平滑。我们称这个奖励函数为*余弦奖励*，如图3所示。这是一个*稀疏*奖励，仅在CoT结束时根据答案的正确性一次性给予。**CosFn**的公式可以在附录中的方程1中找到。

$$R(C, L_{\text{gen}}) = \begin{cases} \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^c, r_L^c), & \text{if } C = 1, \\ \text{CosFn}(L_{\text{gen}}, L_{\max}, r_0^w, r_L^w), & \text{if } C = 0, \\ r_e, & \text{if } L_{\text{gen}} = L_{\max}. \end{cases}$$

Hyperparameters:

r_0^c / r_0^w : Reward (correct/wrong) for $L_{\text{gen}} = 0$,

r_L^c / r_L^w : Reward (correct/wrong) for $L_{\text{gen}} = L_{\max}$,

r_e : Exceed length penalty,

Inputs:

C : Correctness (0 or 1),

L_{gen} : Generation length.

设置. 我们使用了经典奖励和余弦奖励进行了实验。我们使用了在MATH训练集上从`QwQ-32B-Preview`蒸馏的长链数据微调的`Llama3.1-8B`作为起点。更多细节见附录E.5.3。

结果. 我们发现，余弦奖励显著稳定了模型在RL下的长度缩放行为，从而也稳定了训练准确率并提高了RL效率（图4）。我们还观察到模型在下游任务上的性能有所提高（图5）。

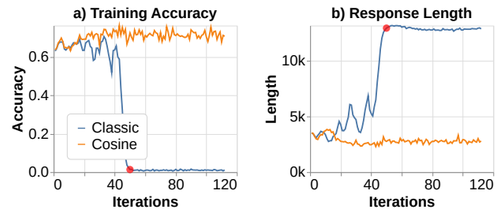


Figure 4. `Llama3.1-8B` 使用余弦奖励（Cosine Reward）进行长度整形训练，表现出更稳定的 (a) 训练准确率和 (b) 响应长度。这种稳定性导致了在下游任务上的性能提升（图 5）。图表中的红点表示训练准确率降至接近零的迭代。

要点 4.2 关于 CoT 长度的主动缩放

奖励塑形可以用于在提高准确性的同时稳定和控制 CoT 长度。(图 4，5)

4.3. Cosine Reward Hyperparameters

余弦奖励的超参数可以调整以不同方式塑造CoT的长度。

设置. 我们使用同一模型在从`QwQ-32B-Preview`蒸馏出的长CoT上进行微调，但在余弦奖励函数中使用了不同的超参数设置。我们调整了正确和错误奖励$r_0^c, r_L^c, r_0^w, r_L^w$，并观察了它们对CoT长度的影响。更多详细信息，请参见附录E.5.4。

结果. 从附录中的图9可以看出，如果正确答案的奖励随着CoT长度的增加而增加（$r_0^c < r_L^c$），CoT长度会急剧增加。我们还发现，正确奖励相对于错误奖励越低，CoT长度越长。我们将其解释为一种训练出的风险规避行为，其中正确和错误奖励的比例决定了模型对答案的置信度，以使其从终止CoT并给出答案中获得正的期望值。

要点 4.3 关于余弦奖励超参数

余弦奖励可以调整以激励不同的长度缩放行为。(附录图 9)

4.4. Context Window Size

我们知道，更长的上下文为模型提供了更多的探索空间，随着训练样本的增加，模型最终学会了利用

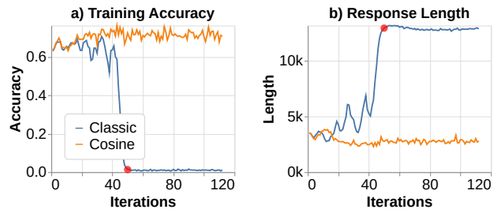


Figure 4. `Llama3.1-8B` trained with length shaping using the Cosine Reward exhibited more stable (a) training accuracy and (b) response length. This stability led to improved performance on downstream tasks (Figure 5). Red points on the charts indicate iterations where training accuracy dropped to near zero.

Result. We found that the Cosine Reward significantly stabilized the length scaling behavior of the models under RL, thereby also stabilizing the training accuracy and improving RL efficiency (Figure 4). We also observed improvements in model performance on downstream tasks (Figure 5).

Takeaway 4.2 for Active Scaling of CoT Length

Reward shaping can be used to stabilize and control CoT length while improving accuracy. (Figure 4, 5)

4.3. Cosine Reward Hyperparameters

The Cosine Reward hyperparameters can be tuned to shape CoT length in different ways.

Setup. We set up RL experiments with the same model fine-tuned on long CoT distilled from `QwQ-32B-Preview`, but with different hyperparameters for the Cosine Reward function. We tweaked the correct and wrong rewards $r_0^c, r_L^c, r_0^w, r_L^w$ and observed their impact on the CoT lengths. For more details, see Appendix E.5.4.

Result. We see from Figure 9 in the Appendix that if the reward for a correct answer increases with CoT length ($r_0^c < r_L^c$), the CoT length increases explosively. We also see that the lower the correct reward relative to the wrong reward, the longer the CoT length. We interpret this as a kind of trained risk aversion, where the ratio of the correct and wrong rewards determines how confident the model has to be about an answer for it to derive a positive expected value from terminating its CoT with an answer.

Takeaway 4.3 for Cosine Reward Hyperparameters

Cosine Reward can be tuned to incentivize various length scaling behaviors. (Appendix Figure 9)

4.4. Context Window Size

We know that longer contexts give a model more room to explore, and with more training samples, the model eventually learns to utilize more of the context window. This raises an interesting question – are more training samples necessary to learn to utilize a larger context window?

Setup. We set up 3 experiments using the same starting model fine-tuned on long CoT data distilled from `QwQ-32B-Preview` with the MATH train split. We also used the latter as our RL prompt set. Each ablation used the Cosine Reward and repetition penalty with a different context window size (4K, 8K, and 16K). For more details, see Appendix E.5.5.

Result. We found that the model with a context window size of 8K performed better than the model with 4K, as expected. However, we observed performance was better under 8K than 16K. Note that all three experiments used the same number of training samples (Figure 6). We see this as an indication that models need more training compute to learn to fully utilize longer context window sizes, which is consistent with the findings of (Hou et al., 2025).

Takeaway 4.4 for Context Window Size

Models might need more training samples to learn to utilize larger context window sizes. (Figure 6)

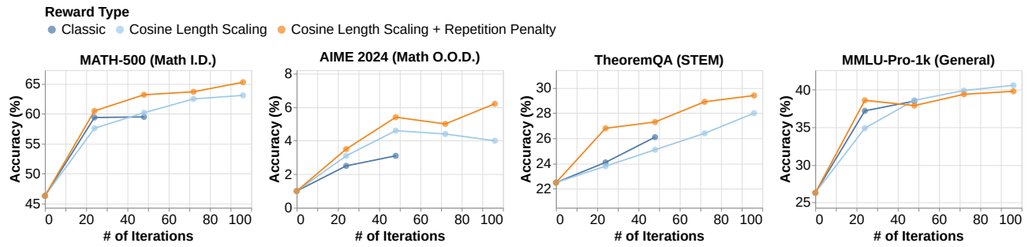


Figure 5. 不同奖励函数训练的`Llama-3.1-8B`在多种评估基准上的表现。

更多的上下文窗口。这引发了一个有趣的问题——是否需要更多的训练样本才能学会利用更大的上下文窗口？

设置。 我们使用相同的起始模型，在从`QwQ-32B-Preview`蒸馏的长CoT数据上进行了微调，并使用MATH训练集作为我们的RL提示集，设置了3个实验。每个消融实验使用了不同的上下文窗口大小（4K、8K和16K），并使用了余弦奖励和重复惩罚。更多详细信息，请参见附录E.5.5。

结果。 我们发现，上下文窗口大小为8K的模型表现优于4K的模型，这在意料之中。然而，我们观察到8K的性能优于16K。需要注意的是，所有三个实验都使用了相同数量的训练样本（图6）。我们认为这表明模型需要更多的训练计算资源才能学会充分利用更长的上下文窗口大小，这与 (Hou et al., 2025) 的发现一致。

要点 4.4 关于上下文窗口大小

模型可能需要更多的训练样本才能学会利用更大的上下文窗口大小。 (图 6)

4.5. Length Reward Hacking

我们观察到，当有足够的训练计算资源时，模型开始表现出奖励劫持的迹象，即通过重复而不是学习解决问题来增加其在难题上的CoT长度。我们还注意到模型的分支频率下降，我们通过计算CoT中关键词“`alternatively,`”出现的次数来估算这一频率（图10）。

我们通过实现一个简单的N-gram重复惩罚（算法1）来缓解这一问题。我们观察到，重复惩罚最有效地应用于重复的标记上，而不是作为整个轨迹的稀疏奖励。同样，我们发现，在计算回报时对重复惩罚进行折现是有效的。关于重复发生的具体反馈可能使模型更容易学习不这样做（详见§4.6）。

设置。 我们使用了在从`QwQ-32B-Preview`蒸馏出的长CoT数据上微调的`Llama3.1-8B`模型。我们进行了两次RL训练运行，均使用余弦奖励，但一次包含重复惩罚，一次不包含。更多详细信息，请参阅附录E.5.6。

结果。 重复惩罚导致了更好的下游任务性能，同时CoT也更短，这意味着推理计算资源的利用效率更高（图5）。

观察。 我们的实验揭示了重复惩罚、训练准确率和余弦奖励之间的关系。当训练准确率较低时，余弦奖励对CoT长度施加了更大的向上压力，导致通过重复进行的奖励劫持增加。这反过来又需要更强的重复惩罚。未来的工作可以进一步研究这些相互作用，并探索动态调优方法以实现更好的优化。

要点 4.5 关于长度奖励攻击

长度奖励在足够的计算资源下会被攻击（图10），但可以通过使用重复惩罚来缓解这一问题。（图5）

4.6. Optimal Discount Factors

我们假设，应用具有时间局部性的重复惩罚（即，较低的折扣因子）将最为有效，因为它为特定的违

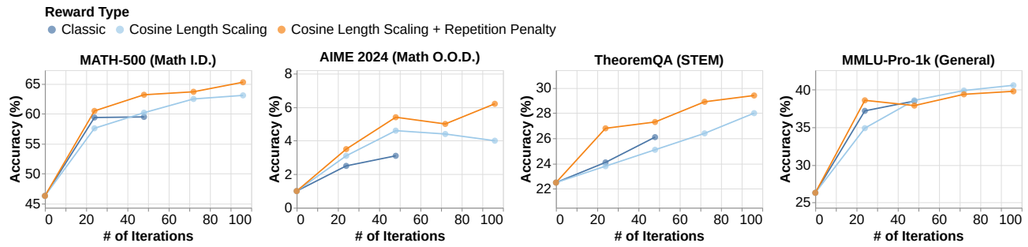


Figure 5. Performance of `Llama-3.1-8B` trained with different reward functions on a variety of evaluation benchmarks.

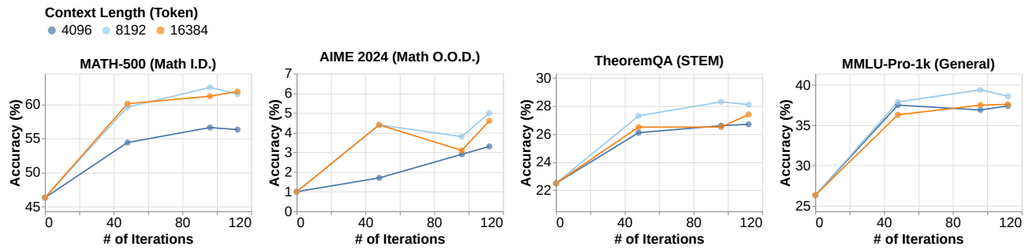


Figure 6. Performance of `Llama-3.1-8B` trained with different context window sizes. All experiments used the same number of training samples.

4.5. Length Reward Hacking

We observed that with enough training compute, the model started to show signs of reward hacking, where it increased the lengths of its CoTs on hard questions using repetition rather than learning to solve them. We also noted a fall in the branching frequency of the model, which we estimated by counting the number of times the pivot keyword "`alternatively,`" appeared in the CoT (Figure 10).

We mitigated this by implementing a simple N-gram repetition penalty (Algorithm 1). We observed that the penalty was most effectively applied on repeated tokens, rather than as a sparse reward for the entire trajectory. Similarly, we found that discounting the repetition penalty when calculating the return was effective. Specific feedback about where the repetition occurred presumably made it easier for the model to learn not to do it (see more in §4.6).

Setup. We used the `Llama3.1-8B` model fine-tuned on long CoT data distilled from `QwQ-32B-Preview`. We ran two RL training runs, both using the Cosine Reward, but with and without the repetition penalty. For more details, please refer to Appendix E.5.6.

Result. The repetition penalty resulted in better downstream task performance and also shorter CoTs, meaning there was better utilization of inference compute (Figure 5).

Observation. Our experiments revealed a relationship between the repetition penalty, training accuracy, and the Cosine Reward. When training accuracy was low, the Cosine Reward exerted greater upward pressure on CoT length, leading to increased reward hacking through repetition. This, in turn, required a stronger repetition penalty. Future work could further investigate these interactions and explore dynamic tuning methods for better optimization.

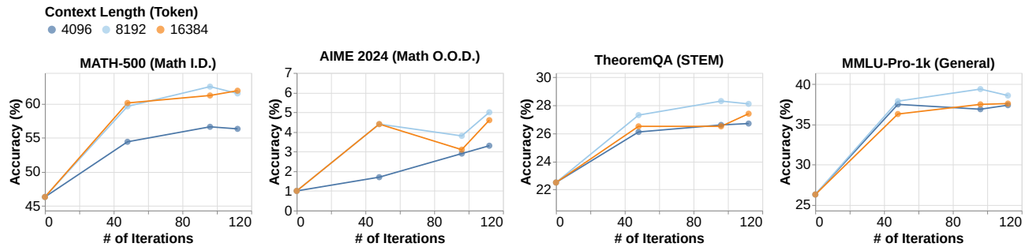


Figure 6. 使用不同上下文窗口大小训练的 `Llama-3.1-8B` 的性能。所有实验使用的训练样本数量相同。

规标记提供了更强的学习信号。然而，我们也观察到，当正确性（余弦）奖励的折扣因子过低时，性能会下降。

为了最佳地调整两种奖励类型，我们修改了PPO中的GAE公式，以适应多种奖励类型，每种类型都有自己的折扣因子γ：$\hat{A}_t = \sum_{l=0}^{L} \sum_{m}^{M} \gamma_m^l r_{m,t+l} - V(s_t)$。为了简化，我们设$\lambda = 1$，这被证明是有效的，尽管我们没有对这个参数进行广泛的调整。

设置。 我们使用相同的`Llama3.1-8B`模型在`QwQ-32B-Preview`蒸馏的长CoT数据上进行微调，运行了多次RL实验。我们使用了余弦奖励和重复惩罚，但采用了不同的折扣因子组合。更多细节，请参见附录E.5.7。

结果。 较低的折扣因子有效地执行了重复惩罚，而较高的折扣因子增强了正确性奖励和超长惩罚。较高的因子使模型在CoT早期选择正确答案时能够获得足够的奖励（图5）。

我们观察到一个相当有趣的现象，即降低正确性（余弦）奖励的折扣因子γ增加了模型CoT中的分支频率，使模型迅速放弃那些似乎不能立即导致正确答案的方法（图11，附录D中的摘录）。我们假设这种短期思维是由于正确答案前的相对较少的标记获得了奖励，这意味着通向正确答案的垫脚石被低估了。这种行为降低了性能（图5）。然而，我们认为这一定性结果可能对研究社区具有潜在的兴趣，因为它与延迟满足等行为与给予生物大脑的奖励分布之间的关系相似(Gao et al., 2021)。

要点 4.6 最优折扣因子

不同类型的奖励和惩罚具有不同的最优折扣因子。(图 5)

5. Scaling up Verifiable Reward

可验证的奖励信号，如基于真实答案的信号，对于稳定长链CoT强化学习以完成推理任务至关重要。然而，由于高质量的人工标注可验证数据在推理任务中的有限可用性，扩大此类数据规模非常困难。为了解决这一问题，我们尝试使用其他更易获得但噪声更大的数据，例如从网络语料库中提取的推理相关问答对。具体来说，我们使用WebInstruct数据集 (Yue et al., 2024b) 进行实验。为了提高效率，我们构建了WebInstruct-462k，这是一个通过MinHash (Broder et al., 1998) 方法去重后得到的子集。

5.1. SFT with Noisy Verifiable Data

我们首先探讨将这种多样化的数据添加到SFT中。直观上，尽管监督信号不太可靠，但多样化数据可能有助于模型在RL过程中的探索。

设置。 我们实验了三种设置，改变没有黄金监督信号的数据比例：0%，100%，和大约50%。我们通过从 `QwQ-32B-Preview` 蒸馏来进行长CoT SFT。对于有黄金监督信号的数据（MATH），使用真实答案进行拒绝采样。相比之下，对于来自WebInstruct的数据，虽然没有完全可靠的监督信号但规模大得多，我们从教师模型中为每个提示采样一个响应，不进行过滤。对于这里的RL，我们采用与 §3.2 中相同的设置，使用MATH训练集。

Takeaway 4.5 for Length Reward Hacking

Length rewards will be hacked with enough compute (Figure 10), but this can be mitigated using a repetition penalty. (Figure 5)

4.6. Optimal Discount Factors

We hypothesized that applying the repetition penalty with temporal locality (i.e., a low discount factor) would be most effective, as it provides a stronger learning signal about the specific offending tokens. However, we also observed performance degradation when the discount factor for the correctness (cosine) reward was too low.

To optimally tune both reward types, we modified the GAE formula in PPO to accommodate multiple reward types, each with its own discount factor γ: $\hat{A}_t = \sum_{l=0}^{L} \sum_{m}^{M} \gamma_m^l r_{m,t+l} - V(s_t)$. For simplicity, we set $\lambda = 1$, which proved effective, though we did not extensively tune this parameter.

Setup. We ran multiple RL experiments with the same `Llama3.1-8B` model fine-tuned on `QwQ-32B-Preview` distilled long CoT data. We used the Cosine Reward and repetition penalty but with different combinations of discount factors. For more details, please see Appendix E.5.7.

Result. A lower discount factor effectively enforces the repetition penalty, whereas a higher discount factor enhances the correctness reward and the exceed-length penalty. The higher factor allows the model to be adequately rewarded for selecting a correct answer earlier in the CoT (Figure 5).

We observed a rather interesting phenomenon where decreasing the discount factor γ of the correctness (cosine) reward increased the branching frequency in the model's CoT, making the model quickly give up on approaches that did not seem to lead to a correct answer immediately (Figure 11, Extract in Appendix D). We hypothesize that this short-term thinking was due to a relatively small number of tokens preceding the correct answer receiving rewards, which means stepping stones to the right answer are undervalued. Such behavior degraded performance (Figure 5). However, we think this qualitative result might be of potential interest to the research community, due to its similarity to the relationship between behaviors like delayed gratification and the distribution of rewards given to the biological brain (Gao et al., 2021).

Takeaway 4.6 for Optimal Discount Factors

Different kinds of rewards and penalties have different optimal discount factors. (Figure 5)

5. Scaling up Verifiable Reward

Verifiable reward signals like ones based on ground-truth answers are essential for stabilizing long CoT RL for reasoning tasks. However, it is difficult to scale up such data due to the limited availability of high-quality human-annotated verifiable data for reasoning tasks. As an attempt to counter this, we explore using other data that is more available despite more noise, like reasoning-related QA pairs extracted from web corpora. Specifically, we experiment with the WebInstruct dataset (Yue et al., 2024b). For efficiency, we construct WebInstruct-462k, a deduplicated subset derived via MinHash (Broder et al., 1998).

5.1. SFT with Noisy Verifiable Data

We first explore adding such diverse data to SFT. Intuitively, despite less reliable supervision signals, diverse data might facilitate the model's exploration during RL.

Setup. We experiment with three setups, varying the proportion of data without gold supervision signals: 0%, 100%, and approximately 50%. We conduct long CoT SFT by distilling from `QwQ-32B-Preview`. For data with gold supervision signals (MATH), ground truth answers are used for rejection sampling. In contrast, for data from WebInstruct without fully reliable supervision signals but with a much larger scale, we sample one

结果。 表 2 显示，加入银级监督数据提高了平均性能。将WebInstruct数据添加到长CoT SFT中，在MMLU-Pro-1k上的绝对准确率比仅使用MATH提高了5–10%。此外，混合MATH和WebInstruct数据在各个基准测试中实现了最佳的平均准确率。

Table 2. 添加带有银监督信号的数据通常是有益的。"WebIT"是 WebInstruct 的缩写。

Long CoT SFT Data	Training Method	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k	AVG
100% MATH	SFT	54.1	3.5	21.8	32.0	27.9
	SFT + RL	**59.4**	4.0	**25.2**	34.6	30.8
100% WebIT	SFT	41.2	0.8	21.9	41.1	26.3
	SFT + RL	44.6	1.9	22.5	**43.3**	28.1
50% MATH + 50% WebIT	SFT	53.6	**4.4**	23.5	41.7	30.8
	SFT + RL	57.3	3.8	25.1	42.0	**32.1**

要点 5.1 对于带有噪声可验证数据的 SFT

向 SFT 添加噪声但多样的数据可以导致在不同任务中实现平衡的性能。 (表 2)

5.2. Scaling up RL with Noisy Verifiable Data

我们比较了从嘈杂的可验证数据中获取奖励的两种主要方法：1) 提取简短答案并使用基于规则的验证器；2) 使用能够处理自由形式响应的基于模型的验证器。

这里的关键因素是问答对是否可以有简短答案。因此，我们还比较了数据集是否仅保留具有简短答案的样本。

设置。 我们通过提示 `Qwen2.5-Math-7B-Instruct` 使用原始参考解决方案来实现基于模型的验证器。为了提取简短答案，我们首先提示 `Llama-3.1-8B-Instruct` 从原始响应中提取答案，然后使用 `QwQ-32B-Preview` 进行拒绝采样。具体来说，我们从 WebInstruct-462k 中每条提示生成两个响应，并丢弃两个响应都不符合提取的参考答案的情况。这一过程产生了大约 189k 个响应，涉及 115k 个独特的提示。我们的案例研究表明，拒绝采样由于以下原因丢弃了许多提示：1) 许多 WebInstruct 提示缺乏我们的基于规则的验证器可以有效处理的简短答案， 2) 一些提示对于 `QwQ-32B-Preview` 来说也太难了。对于 SFT，我们在过滤后的数据集上训练 `Llama-3.1-8B` 作为强化学习 (RL) 的初始化。在 RL 阶段，我们在未过滤的设置中使用完整的 462k 提示集，在过滤的设置中使用 115k 子集，每次训练使用 30k 提示和每个提示 4 个响应。关于基于模型的验证器、答案提取和 RL 超参数的更多详细信息可以在附录 & E.5.8 & E.6 & E.7 中找到。

Table 3. 不同验证器和提示过滤方法下的RL性能。此处的所有模型都是从`Llama-3.1-8B`微调而来。"MATH Baseline"是在表2中仅使用MATH数据集通过SFT和RL训练的模型。其他模型则是通过从`QwQ-32B-Preview`蒸馏的SFT和不同设置的RL训练而来。

Prompt Set	Verifier Type	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k
MATH Baseline		59.4	4.0	25.2	34.6
SFT Initialization		46.6	1.0	23.0	28.3
Unfiltered	Rule-Based	45.4	3.3	25.9	35.1
	Model-Based	47.9	3.5	26.2	40.4
Filtered	Rule-Based	**48.6**	3.3	**28.1**	**41.4**
	Model-Based	47.9	**3.8**	26.9	**41.4**

结果。 Table 3 显示，在相同数量的 RL 样本下，基于规则的验证器在过滤后的简短答案提示集上使用 RL 达到了大多数基准测试中的最佳性能。这可能表明，在适当的过滤后，基于规则的验证器可以从嘈杂的可验证数据中生成最高质量的奖励信号。此外，与在人工标注的验证数据（MATH）上训练的模型相比，利用嘈杂但多样的可验证数据仍然显著提高了 O.O.D. 基准测试的性能，TheoremQA 的绝对增益高达 2.9%，MMLU-Pro-1k 的绝对增益高达 6.8%。相比之下，将基于规则的验证器应用于未过滤的数据会导致最差的性能。这可能是由于其在自由形式答案上的训练准确率较低，而基于模型的验证器则表现得更好。

从带有噪声的可验证数据中获取奖励信号的RL 5.2

为了从带有噪声的可验证数据中获得奖励信号，基于规则的验证器在过滤短答案的提示集后表现最佳。 (表 3)

response per prompt from the teacher model without filtration. For RL here, we adopt the same setup as in §3.2, using the MATH training set.

Result. Table 2 shows that incorporating silver-supervised data improves average performance. Adding WebInstruct data to long CoT SFT yields a substantial 5–10% absolute accuracy gain on MMLU-Pro-1k over using MATH alone. Furthermore, mixing MATH and WebInstruct data achieves the best average accuracy across benchmarks.

Table 2. Adding data with a silver supervision signal is often beneficial. "WebIT" is the abbreviation of WebInstruct.

Long CoT SFT Data	Training Method	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k	AVG
100% MATH	SFT	54.1	3.5	21.8	32.0	27.9
	SFT + RL	**59.4**	4.0	**25.2**	34.6	30.8
100% WebIT	SFT	41.2	0.8	21.9	41.1	26.3
	SFT + RL	44.6	1.9	22.5	**43.3**	28.1
50% MATH + 50% WebIT	SFT	53.6	**4.4**	23.5	41.7	30.8
	SFT + RL	57.3	3.8	25.1	42.0	**32.1**

Takeaway 5.1 for SFT with Noisy Verifiable Data

Adding noisy but diverse data to SFT leads balanced performance across different tasks. (Table 2)

5.2. Scaling up RL with Noisy Verifiable Data

We compare two main approaches to obtain rewards from noisy verifiable data: 1) to extract short-form answers and use a rule-based verifier; 2) to use a model-based verifier capable of processing free-form responses.

Here a key factor is whether the QA pair can have a short-form answer. So we also compare whether the dataset is filtered by only retaining samples with short-form answers.

Setup. We implement the model-based verifier by prompting `Qwen2.5-Math-7B-Instruct` with the raw reference solution. To extract short-form answers, we first prompt `Llama-3.1-8B-Instruct` to extract from the raw responses and then apply rejection sampling with `QwQ-32B-Preview`. Specifically, we generate two responses per prompt from WebInstruct-462k and discard cases where neither response aligns with the extracted reference answers. This process yields approximately 189k responses across 115k unique prompts. Our case studies show that the rejection sampling drops many prompts due to: 1) many WebInstruct prompts lack short-form answers that our rule-based verifier can process effectively, and 2) some prompts are too difficult even for `QwQ-32B-Preview`. For SFT we train `Llama-3.1-8B` on the filtered dataset as initialization for reinforcement learning (RL). In the RL stage, we use the full 462k prompt set in the unfiltered setup and the 115k subset in the filtered setup, training with 30k prompts and 4 responses per prompt. Further details about the model-based verifier, the answer extraction and the RL hyperparameters can be found in Appendix & E.5.8 & E.6 & E.7 respectively.

Table 3. Performance of RL with different verifiers and prompt filtering methods. All the models here are fine-tuned from `Llama-3.1-8B`. The "MATH Baseline" is the model trained with SFT and RL on MATH only in Table 2. The other models are trained with SFT by distillation from `QwQ-32B-Preview` and RL with different setups.

Prompt Set	Verifier Type	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k
MATH Baseline		59.4	4.0	25.2	34.6
SFT Initialization		46.6	1.0	23.0	28.3
Unfiltered	Rule-Based	45.4	3.3	25.9	35.1
	Model-Based	47.9	3.5	26.2	40.4
Filtered	Rule-Based	**48.6**	3.3	**28.1**	**41.4**
	Model-Based	47.9	**3.8**	26.9	**41.4**

Result. Table 3 shows that RL with the rule-based verifier on the filtered prompt set with short-form answers achieves the best performance across most benchmarks under the same number of RL samples. This might indicate that rule-based verifier after appropriate filtration can produce the highest-quality reward signals from noisy verifiable data. Moreover, compared to the model trained on human-annotated verified data (MATH), leveraging noisy yet diverse verifiable data still significantly boosts

6. Exploration on RL from the Base Model

DeepSeek-R1 (DeepSeek-AI, 2025) 已经证明，通过在基础模型上扩大强化学习的计算规模，可以产生长链推理。最近的研究 (Zeng et al., 2025; Pan et al., 2025) 尝试通过运行相对较少的强化学习迭代来观察长链推理行为的出现（例如，"啊哈时刻" (DeepSeek-AI, 2025)，这是一种能够实现自我验证和纠正等关键功能的突发性认识）。在本节中，我们还探讨了从基础模型进行强化学习的方法。

6.1. Nuances in Analysis Based on Emergent Behaviors

自我验证行为有时被模型的探索标记为新兴行为或"恍然大悟"时刻，因为这种模式在短链思考（CoT）数据中很少见。然而，我们注意到有时自我验证行为已经存在于基础模型中，通过强化学习（RL）加强这些行为需要严格的条件，例如一个强大的基础模型。

设置。 我们遵循 Zeng et al. (2025) 的设置，使用基于规则的验证器通过 PPO 训练 `Qwen2.5-Math-7B`，大约在 8k MATH 3-5 级问题上进行训练，但使用我们自己的基于规则的验证器实现。对于推理，我们采用温度 $t = 0$（贪婪解码），因为我们的初步实验显示，对于直接从 `Qwen2.5-Math-7B` 通过 RL 获得的模型，$t = 0$ 通常显著优于 $t > 0$。我们使用最大输出长度为 4096 个 token，考虑到训练上下文长度为 4096 个 token。请注意，我们使用零样本提示基础模型，以避免引入输出模式的偏差。我们从之前工作的长链思考案例中选择了五个具有代表性的关键词，"wait"（等待）、"recheck"（重新检查）、"alternatively"（或者）、"retry"（重试）和"however"（然而）(OpenAI, 2024; DeepSeek-AI, 2025; Pan et al., 2025; Zeng et al., 2025)，并计算它们的频率，以量化模型进行自我验证的程度。关于 RL 超参数的更多详细信息可以在附录 E.5.9 中找到。

结果。 图 7 显示，我们的从 `Qwen2.5- Math-7B` 模型中进行的强化学习（RL）有效地提高了准确性，但并没有增加基础模型输出中已存在的"复查"模式的频率，也没有有效地激励其他反思模式，如"重试"和"另选方案"。这表明，尽管显著提高了性能，但从基础模型进行的强化学习并不一定会激励反思模式。有时这些行为存在于基础模型的输出中，而强化学习并没有实质性地增强它们。因此，我们可能需要更加谨慎地识别新兴行为。

6.2. Nuances in Analysis Based on Length Scaling

长度扩展被认为是模型有效探索的另一个重要特征。然而，我们注意到，有时长度扩展可能伴随着KL散度的减少，这提出了长度可能受到KL惩罚的影响，只是回到基础模型的较长输出，而不是反映获得长链思考能力的可能性。

设置。 设置与§6.1相同。除了输出标记长度外，我们还计算了"编码率"。如果模型的输出包含"```python"，则将其分类为"编码"，因为`Qwen2.5-Math-7B`使用自然语言和编码来解决数学问题。请注意，这里的"编码"输出实际上是自然语言输出的一种特殊形式，其中的代码不会被执行，代码的输出是由模型生成的。

结果。 图8 (1)显示，输出标记的长度在初始下降后增加，但从未超过基础模型的初始长度。

Zeng et al. (2025)建议，初始下降可能是由于模型从生成长编码输出转变为生成较短的自然语言输出。然而，图8 (2)表明，自然语言输出实际上比编码输出更长，且长度的初始下降发生在两种类型的输出中。此外，图8 (3)显示，编码率随后再次增加，这表明编码与自然语言之间的区别可能不会显著影响优化过程。

此外，我们怀疑随后的长度扩展不是来自模型的探索，因为当长度扩展时，策略相对于基础模型的KL散度下降，如图8 (4)所示。这可能表明是KL惩罚影响了长度。如果是这种情况，策略输出长度超过基础模型的潜力很小，因为探索受到KL约束的限制。

performance on O.O.D. benchmarks, with absolute gains of up to 2.9% on TheoremQA and 6.8% on MMLU-Pro-1k. In contrast, applying a rule-based verifier to unfiltered data results in the worst performance. This might be caused by its low training accuracy on free-form answers, while the model-based verifier achieves much better performance.

Takeaway 5.2 for RL with Noisy Verifiable Data

To obtain reward signals from noisy verifiable data, the ruled-based verifier after filtering the prompt set for short-form answers works the best. (Table 3)

6. Exploration on RL from the Base Model

DeepSeek-R1 (DeepSeek-AI, 2025) has demonstrated that long chain-of-thought reasoning can emerge by scaling up reinforcement learning compute on a base model. Recent studies (Zeng et al., 2025; Pan et al., 2025) have attempted to replicate this progress by running a relatively small number of RL iterations to observe the emergence of long CoT behavior (e.g., the "aha moment" (DeepSeek-AI, 2025), an emergent realization moment that enables critical functions like self-validation and correction). We also explore the method of RL from the base model in this section.

6.1. Nuances in Analysis Based on Emergent Behaviors

Self-validation behaviors are sometimes flagged as emergent behaviors or "aha-moment" by the model's exploration, since such patterns are rare in short CoT data. However, we notice that sometimes self-validation behaviors already exist in the base model and reinforcing them through RL requires strict conditions, such as a strong base model.

Setup. We follow the setup from Zeng et al. (2025) to train `Qwen2.5-Math-7B` using PPO with a rule-based verifier on approximately 8k MATH level 3-5 questions, but we use our own rule-based verifier implementation. For inference, we adopt temperature $t = 0$ (greedy decoding), as our preliminary experiments show that $t = 0$ usually significantly outperforms $t > 0$ for models obtained by direct RL from `Qwen2.5-Math-7B`. We use the maximum output length of 4096 tokens considering the training context length of 4096 tokens. Note that we use zero-shot prompting for the base model to avoid introducing biases to the output pattern. We select five representative keywords, "wait", "recheck", "alternatively", "retry" and "however" from long CoT cases in previous works (OpenAI, 2024; DeepSeek-AI, 2025; Pan et al., 2025; Zeng et al., 2025), and calculate their frequencies to quantify the extent to which the model does self-validation. Further details about the RL hyperparameters can be found in Appendix E.5.9.

Result. Figure 7 shows that our RL from `Qwen2.5-Math-7B` effectively boosts the accuracies, but does not increase the frequency of the "recheck" pattern existing in the output of the base model, nor effectively incentivize other reflection patterns such as "retry" and "alternatively". This indicates that RL from the base model does not necessarily incentivize reflection patterns, though significantly boosting the performance. Sometimes such behaviors exist in the base model's output and RL does not substantially enhance them. So we might need to be more careful about recognizing emergent behaviors.

6.2. Nuances in Analysis Based on Length Scaling

The length scaling up is recognized as another important feature of the effective exploration of the model. However, we notice that sometimes length scaling up can be accompanied by the KL divergence decreasing, which raises the possibility that the length is influenced by the KL penalty and is just reverting back to the base model's longer outputs, rather than reflecting the acquisition of long CoT ability.

Setup. The setup is the same as in §6.1. Besides the

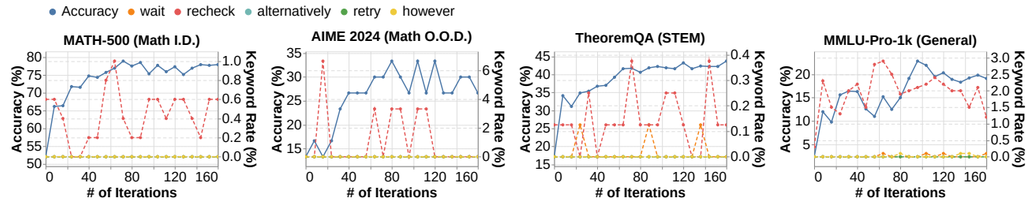


Figure 7. 在从基础模型 `Qwen2.5-Math-7B` 进行强化学习 (RL) 期间，不同基准上的准确性和反射关键词率的动态变化。尽管准确性稳步提高，但我们没有看到"wait"、"alternatively"和"recheck"这些关键词率在RL训练过程中有显著提升。

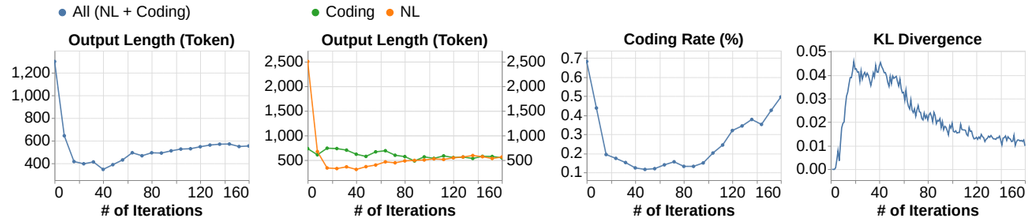


Figure 8. 在 `Qwen2.5-Math-7B` 的强化学习过程中，MATH-500 上的输出令牌长度和编码率的动态变化以及 MATH Lv3-5（训练数据）上策略相对于基础模型的 KL 散度。

6.3. Potential Reasons Why Emergent Behavior is Not Observed with `Qwen2.5-Math-7B`

我们对来自`Qwen2.5-Math-7B`的RL的详细分析，如在§6.1和§6.2中所呈现的，表明它未能完全复制`DeepSeek-R1`的训练行为。我们确定了以下潜在原因：1) 基础模型相对较小（7B参数），可能缺乏在受到激励时快速发展这种复杂能力的容量。2) 模型可能在（持续）预训练和退火过程中过度暴露于类似MATH的短指令数据，导致过拟合并阻碍了长链推理行为的发展。

6.4. Comparison between RL from the Base Model and RL from Long CoT SFT

我们比较了从基础模型和从长CoT SFT中获得的RL的性能，发现从长CoT SFT中获得的RL通常表现更好。

设置。 我们使用基础模型 `Qwen2.5- Math-7B` 进行比较。从基础模型中获得的RL结果来自 §6.1 中训练的模型。对于从长CoT SFT中获得的RL，我们采用了类似于 §3.2 的设置。具体来说，我们选择了7.5k MATH训练集作为提示集，通过使用 `QwQ-32B-Preview` 对每个提示进行32个候选响应的拒绝采样来整理SFT数据，并使用我们的余弦长度缩放奖励和重复惩罚以及基于规则的验证器进行PPO训练，每个提示采样8个响应，训练8个epoch。为了使 `Qwen2.5-Math-7B` 适应长CoT SFT和RL，该模型的预训练上下文长度仅为4096个token，我们将其RoPE (Su et al., 2024) θ 增加了10倍。我们不报告从长CoT SFT中获得的RL的经典奖励结果，因为它崩溃了。在评估中，我们采用了 §2.5 中的默认温度采样设置来评估从长CoT SFT中获得的RL，并采用了 §6.1 中的贪婪解码设置来评估从基础模型中获得的RL，以获得最佳性能。关于蒸馏、SFT超参数和RL超参数的更多细节分别可以在附录 E.2 & E.3 & E.5.9 中找到。

结果。 表 4 显示，在 `Qwen2.5-Math-7B` 上，从长 CoT SFT 模型初始化的 RL 显著优于从基础模型初始化的 RL，并且进一步改进了长 CoT SFT 本身。具体来说，使用我们余弦奖励的长 CoT SFT 的 RL 平均比从基础模型初始化的 RL 高出显著的 8.7%，并比 SFT 初始化提高了 2.6%。值得注意的是，仅应用从 `QwQ-32B-Preview` 蒸馏的长 CoT 的 SFT 已经能够产生强大的性能。

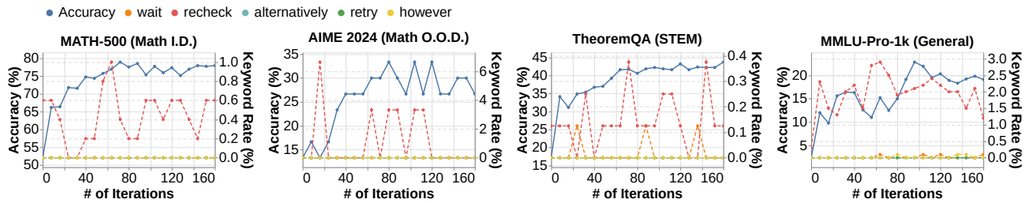


Figure 7. Dynamics of accuracies and reflection keyword rates on different benchmarks during our RL from the base model `Qwen2.5-Math-7B`. We do not see the keyword rates of "wait", "alternatively", and "recheck" get significantly improved during the RL training even though the accuracy is steadily increasing.

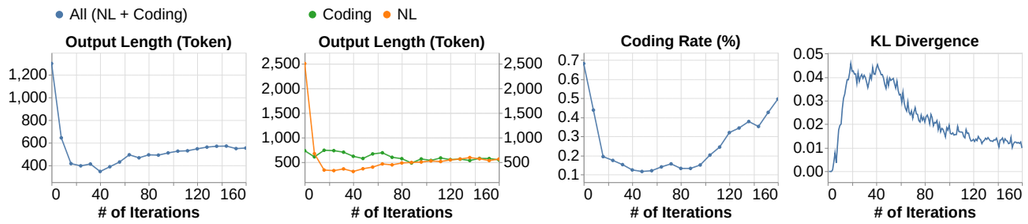


Figure 8. Dynamics of the output token lengths and the coding rate on MATH-500 and the KL divergence of the policy over the base model on MATH Lv3-5 (training data) during our RL from `Qwen2.5-Math-7B`.

output token length, we also calculate the "coding rate". We classify the model's output as "coding" if it contains the "```python", since `Qwen2.5-Math-7B` uses both natural language and coding to solve mathematical problems. Note that the "coding" output here is actually a special form of natural language output, where the code in it is not executed, and the code's output is generated by the model.

Result. Figure 8 (1) shows that the length of the output token increases after an initial drop, but never exceeds the initial length of the base model.

Zeng et al. (2025) suggest that the initial drop may be due to the model transitioning from generating long coding outputs to shorter natural language outputs. However, Figure 8 (2) indicates that natural language outputs are actually longer than coding outputs, and the initial drop in length occurs in both types of output. Furthermore, Figure 8 (3) shows that the coding rate subsequently increases again, suggesting that the distinction between coding and natural language may not significantly impact the optimization process.

Moreover, we suspect that the subsequent length scaling up is not from the model's exploration, since when the length scales up, the KL divergence of the policy over the base model drops, as shown in Figure 8 (4). This might indicate that it is the KL penalty influencing length. If that is the case, there is little potential for the policy output length to exceed the base model's since the exploration is limited by the KL constraint.

6.3. Potential Reasons Why Emergent Behavior is Not Observed with `Qwen2.5-Math-7B`

Our detailed analysis of RL from `Qwen2.5-Math-7B`, as presented in §6.1 and §6.2, suggests that it fails to fully replicate the training behavior of `DeepSeek-R1`. We identify the following potential causes: 1) The base model, being relatively small (7B parameters), may lack the capacity to quickly develop such complex abilities when incentivized. 2) The model might have been over-exposed to MATH-like short instruction data during (continual) pre-training and annealing, leading to overfitting and hindering the development of long CoT behaviors.

Table 4. 基于 `Qwen2.5-Math-7B` 的不同模型的性能。这里的 SFT 数据是从 `QwQ-32B-Preview` 通过拒绝采样蒸馏得到的。

Setup	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k	AVG
Base (0-shot)	52.0	13.3	17.1	2.4	21.2
(Direct) RL	77.4	23.3	43.5	19.7	41.0
SFT	84.0	24.4	42.2	38.5	47.3
SFT + RL	**85.9**	**26.9**	**45.4**	**40.6**	**49.7**

6.5. Long CoT Patterns in Pre-training Data

基于§6.1中的结果，我们假设，诸如模型重新审视其解决方案等激励行为可能已经在预训练期间部分习得。为了检验这一点，我们采用了两种方法来调查此类数据是否已经存在于网络上。

首先，我们使用了生成式搜索引擎Perplexity.ai来识别明确包含从多个角度解决问题或在提供答案后进行验证的网页。我们使用的查询和识别的示例见附录F.1。

其次，我们使用`GPT-4o`生成了一系列具有"恍然大悟"特征的短语（附录F.2.1），然后使用MinHash算法(Broder, 1997)在OpenWebMath(Paster et al., 2023)中进行搜索，该数据集是从CommonCrawl(Rana, 2010)中过滤出来的，常用于预训练。我们发现，在讨论论坛的帖子中存在大量匹配项，其中多个用户之间的对话显示出与长链思考（CoT）的相似性，讨论了多种方法，并伴随有回溯和错误纠正（附录F.2.2）。这提出了一个有趣的假设，即长链思考可能源自人类对话，尽管我们也应该注意到讨论论坛是OpenWebMath中的常见数据来源。

基于这些观察，我们假设强化学习（RL）主要指导模型重新组合其在预训练期间已经内化的技能，以形成新的行为，从而提高在复杂问题解决任务中的表现。鉴于本文的广泛范围，我们留待未来的工作对这种行为进行更深入的调查。

7. Discussions and Future Work

在本工作中，我们揭示了长链思维推理在大语言模型中的奥秘。在本节中，我们概述了潜在的未来方向。

7.1. Scaling up Model Size

我们相信模型大小是限制 subsection 6.1 中观察到的行为出现的主要因素。Hyung Won Chung (Chung, 2024) 最近分享了类似的观点，认为较小的模型可能难以发展高级推理能力，而是依赖于基于启发式的模式识别。未来的研究可以调查使用更大基础模型的强化学习。

7.2. RL 基础设施仍处于初级阶段

在尝试扩大模型规模时，我们在扩展到 32B 时遇到了显著的挑战，最终确定所需的 GPU 数量太大而无法继续。我们观察到，开源的强化学习框架（例如 OpenRLHF (Hu et al., 2024)）通常协调多个针对不同训练和推理工作负载优化的系统，导致模型参数的多个副本被存储在内存中。此外，像 PPO 这样的算法在这些工作负载之间同步且顺序地交替，进一步限制了效率。这些因素导致硬件利用率低，特别是在长 CoT 情景中，由于 CoT 长度的更高方差，这在推理过程中导致了落后者 (Kimi Team, 2025)。我们期待机器学习和系统研究的进展能够帮助克服这些限制，加速长 CoT 建模的进展。

7.3. REINFORCE Is More Tricky to Tune than PPO

我们还探索了REINFORCE++ (Hu, 2025)作为PPO的更快替代方案，以扩展数据规模。然而，我们发现REINFORCE++比PPO显著更不稳定，导致训练准确率较低（图13）。由于这种不稳定性可能是由于未调优的设置（附录E.5.10），我们避免对算法做出一般性结论。我们将其作为一个可能对社区有用的观察结果呈现出来。

6.4. Comparison between RL from the Base Model and RL from Long CoT SFT

We compare the performance of RL from the base model and RL from long CoT SFT and find that RL from long CoT SFT generally performs better.

Setup. We compare using the base model `Qwen2.5-Math-7B`. The results of RL from the base model are from the model trained in §6.1. For RL from long CoT SFT, we adopt a setup similar to §3.2. Specifically, we choose the 7.5k MATH training set as the prompt set, curate the SFT data by rejection sampling with 32 candidate responses per prompt using `QwQ-32B-Preview`, and perform PPO using our cosine length-scaling reward with repetition penalty and our rule-based verifier, sampling 8 responses per prompt and training for 8 epochs. To adapt `Qwen2.5-Math-7B` with a pre-training context length of only 4096 tokens to long CoT SFT and RL, we multiply its RoPE (Su et al., 2024) θ by 10 times. We don't report the results of RL with classic reward from long CoT SFT since it collapses. For evaluation, we adopt our default temperature sampling setup for RL from long CoT SFT as in §2.5 and greedy decoding setup for RL from the base model as in §6.1 for the best performance. Further details about the distillation, SFT hyperparameters and RL hyperparameters can be found in Appendix E.2 & E.3 & E.5.9, respectively.

Table 4. Performance of different models based on `Qwen2.5-Math-7B`. The SFT data here is distilled with rejection sampling from `QwQ-32B-Preview`.

Setup	MATH 500	AIME 2024	Theo. QA	MMLU Pro-1k	AVG
Base (0-shot)	52.0	13.3	17.1	2.4	21.2
(Direct) RL	77.4	23.3	43.5	19.7	41.0
SFT	84.0	24.4	42.2	38.5	47.3
SFT + RL	**85.9**	**26.9**	**45.4**	**40.6**	**49.7**

Result. Table 4 shows that, on `Qwen2.5-Math-7B`, RL initialized from the long CoT SFT model significantly outperforms RL from the base model and further improves upon the long CoT SFT itself. Specifically, RL from long CoT SFT with our cosine reward surpasses RL from the base model by a substantial 8.7% on average and improves over the SFT initialization by 2.6%. Notably, simply applying SFT with long CoT distilled from `QwQ-32B-Preview` already yields strong performance.

6.5. Long CoT Patterns in Pre-training Data

Based on the results in §6.1, we hypothesize that incentivized behaviors, such as the model revisiting its solutions, may have already been partially learned during pre-training. To examine this, we employed two methods to investigate whether such data are already present on the web.

Firstly, we used a generative search engine Perplexity.ai to identify webpages explicitly containing problem-solving steps that approach problems from multiple angles or perform verification after providing an answer. The query we used and the examples we identified are in Appendix F.1).

Secondly, we used `GPT-4o` to generate a list of phrases that are characteristic of the "aha moment" (Appendix F.2.1), then used the MinHash algorithm (Broder, 1997) to search through OpenWebMath (Paster et al., 2023), a dataset filtered from the CommonCrawl (Rana, 2010) frequently used in pre-training. We found that there was a significant number of matches in discussion forum threads, where the dialogue between multiple users showed similarity to long CoT, with multiple approaches being discussed along with backtracking and error correction (Appendix F.2.2). This raises the intriguing possibility that long CoT originated from human dialogue, although we should also note that discussion forums are a common source of data in OpenWebMath.

Based on these observations, we hypothesize that RL primarily guides the model to recombine skills it already internalized during pre-training towards new behaviors to

7.4. Scaling up Verification

虽然我们的研究结果表明，将基于规则的验证器与提示集过滤相结合非常有效，但在不同领域设计这些规则和整理提示集仍然是一项劳动密集型任务。更根本的是，这种方法将人类设计的启发式方法嵌入到强化学习（RL）环境中，反映了我们的思维方式，而不是允许出现性学习。正如《苦涩的教训》[1]所强调的，手动编码人类直觉往往是一种低效的长期策略。这引发了一个有趣的问题：如何有效地扩展验证信号？在设计RL环境的背景下，是否存在类似于预训练的方法？我们期待未来在银色监督信号和强化学习验证中自监督方法的潜力方面的研究。

7.5. Latent Capabilities in Base Models

推理是基础模型中的一种潜在能力，最近才被解锁。我们的分析表明，这种新兴思维的一个可能来源是互联网讨论论坛上的人类对话。这引发了一个更广泛的问题：还有哪些能力存在于预训练数据中，等待着从人类知识和经验的庞大库中被激发出来？我们期待更详细的分析，将模型行为追溯到其数据来源，这可能会带来新的见解，并帮助揭示基础模型中的隐藏能力。

Impact Statement

本文旨在提供关于扩展推理计算和训练策略的见解，以实现大型语言模型中的长链推理。这项工作的更广泛影响主要与在各个领域中增强推理和解决问题的能力的潜力有关，其中能够进行可解释和可验证推理的模型可以推动创新并改善决策。我们的研究结果强调了确保稳健的训练数据准备、稳定性和与可验证事实的一致性的重要性。我们鼓励未来的研究积极开发确保这些能力负责任使用的保护措施。这包括精心设计奖励塑造和训练协议，以最小化意外后果，同时最大化社会利益。

[1]http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Acknowledgment

作者要感谢李远志在这一主题上的深入讨论。作者还要感谢SimpleRL团队，特别是曾伟豪和何俊贤，感谢他们分享训练经验和实验观察。此外，作者感谢陈文虎、任晓义、李超、马子乔、潘佳怡、王兴耀和金胜一在项目早期或最终阶段提供的宝贵意见和讨论。最后，作者要感谢DeepSeek-R1和Kimi-k1.5团队发布的技术报告，这些报告启发了本文的几个额外实验设计。这项工作部分得到了卡内基·博世研究所对岳翔的奖学金支持。

参考文献

Anthropic. Introducing claude, 2023. URL https://www.anthropic.com/index/introducing-claude.

Broder, A. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pp. 21–29, 1997. doi: 10.1109/SEQUEN.1997.666900.

Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. Min-wise independent permutations. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 327–336, 1998.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain,

improve performance on complex problem-solving tasks. Given the broad scope of this paper, we leave a more in-depth investigation of this behavior to future work.

7. Discussions and Future Work

In this work, we demystify long CoT reasoning in LLMs. In this section, we outline potential future directions.

7.1. Scaling up Model Size

We believe that model size is the primary factor limiting the emergence of the behavior observed in subsection 6.1. Hyung Won Chung (Chung, 2024) recently shared a similar perspective, suggesting that smaller models may struggle to develop high-level reasoning skills and instead rely on heuristic-based pattern recognition. Future research could investigate RL using a larger base model.

7.2. RL Infrastructure Is Still in Its Infancy

While attempting to scale up the model size, we encountered significant challenges in expanding to 32B, ultimately determining that the required number of GPUs was too large to proceed. We observe that open-source RL frameworks (e.g., OpenRLHF (Hu et al., 2024)) often coordinate multiple systems optimized for different training and inference workloads, leading to multiple copies of model parameters being stored in memory. Additionally, algorithms like PPO alternate between these workloads synchronously and sequentially, further limiting efficiency. These factors contribute to low hardware utilization, an issue that is particularly exacerbated in long CoT scenarios due to the higher variance in CoT length, which leads to stragglers during inference (Kimi Team, 2025). We look forward to advancements in machine learning and systems research that will help overcome these limitations and accelerate progress in long CoT modeling.

7.3. REINFORCE Is More Tricky to Tune than PPO

We also explored REINFORCE++ (Hu, 2025) as a faster alternative to PPO for scaling up data. However, we found it to be significantly more unstable than PPO, leading to lower training accuracies (Figure 13). As this instability may be due to an untuned setup (Appendix E.5.10), we refrain from making general claims about the algorithm. We present this as an observation that may be useful to the community.

7.4. Scaling up Verification

While our findings demonstrate that combining rule-based verifiers with prompt set filtering is highly effective, designing such rules and curating prompt sets across different domains remains labor-intensive. More fundamentally, this approach embeds human-designed heuristics into the RL environment, reflecting how we think rather than allowing for emergent learning. As highlighted in The Bitter Lesson[1], manually encoding human intuition tends to be an inefficient long-term strategy. This raises an intriguing question: how can verification signals be scaled effectively? Is there an equivalent of pretraining in the context of designing RL environments? We look forward to future research on silver supervision signals and the potential for self-supervised approaches in RL verification.

7.5. Latent Capabilities in Base Models

Reasoning is a latent capability in base models that has only recently been unlocked. Our analysis suggests that one possible source of this emergent thinking is human dialogue on Internet discussion forums. This raises a broader question: what other abilities exist, waiting to be elicited from the vast reservoir of human knowledge and experience embedded in pre-training data? We look forward to more detailed analyses tracing model behaviors back to their data origins, which could yield new insights

[1] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.

Chen, W., Yin, M., Ku, M., Lu, P., Wan, Y., Ma, X., Xu, J., Wang, X., and Xia, T. TheoremQA: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Chung, H. W. Don't teach. incentivize. Presentation slides, 2024. URL https://t.co/2sjhynKxzJ. Slide 48.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dao, T. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., KaShun, S., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

Feng, X., Wan, Z., Wen, M., McAleer, S. M., Wen, Y., Zhang, W., and Wang, J. Alphazero-like tree-search can guide large language model decoding and training, 2023.

Gao, Z., Wang, H., Lu, C., Lu, T., Froudist-Walsh, S., Chen, M., Wang, X.-J., Hu, J., and Sun, W. The neural basis of delayed gratification. *Science Advances*, 7 (49):eabg6611, 2021. doi: 10.1126/sciadv.abg6611.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Hou, Z., Lv, X., Lu, R., Zhang, J., Li, Y., Yao, Z., Li, J., Tang, J., and Dong, Y. Advancing language model reasoning through reinforcement learning and inference scaling, 2025.

Hu, J. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Hu, J., Wu, X., Zhu, Z., Xianyu, Wang, W., Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework, 2024.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. R. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025.

Lamb, A. M., ALIAS PARTH GOYAL, A. G., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural*

and help uncover hidden capabilities within base models.

Impact Statement

This paper aims to provide insights into scaling inference compute and training strategies to enable long chain-of-thought reasoning in large language models. The broader impacts of this work primarily relate to the potential for enhanced reasoning and problem-solving capabilities across various domains, where models capable of interpretable and verifiable reasoning could drive innovation and improve decision-making. Our findings emphasize the importance of ensuring robust training data preparation, stability, and alignment with verifiable ground truths. We encourage future research to actively develop safeguards that ensure these capabilities are used responsibly. This includes careful design of reward shaping and training protocols to minimize unintended consequences while maximizing societal benefits.

Acknowledgment

The authors would thank Yuanzhi Li for insightful discussions on this topic. The authors would also thank the SimpleRL team, particularly Weihao Zeng and Junxian He, for sharing their training experiences and experimental observations. Additionally, the authors appreciate Wenhu Chen, Xiaoyi Ren, Chao Li, Ziqiao Ma, Jiayi Pan, Xingyao Wang, and Seungone Kim for their valuable comments and discussions during the early or final stages of the project. Finally, the authors would acknowledge the DeepSeek-R1 and Kimi-k1.5 teams for their technical report releases, which inspired several additional experiment designs of this paper. This work was supported in part by a Carnegie Bosch Institute Fellowship to Xiang Yue.

参考文献

Anthropic. Introducing claude, 2023. URL https://www.anthropic.com/index/introducing-claude.

Broder, A. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pp. 21–29, 1997. doi: 10.1109/SEQUEN.1997.666900.

Broder, A. Z., Charikar, M., Frieze, A. M., and Mitzenmacher, M. Min-wise independent permutations. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 327–336, 1998.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code, 2021.

Chen, W., Yin, M., Ku, M., Lu, P., Wan, Y., Ma, X., Xu, J., Wang, X., and Xia, T. TheoremQA: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Chung, H. W. Don't teach. incentivize. Presentation slides, 2024. URL https://t.co/2sjhynKxzJ. Slide 48.

Information Processing Systems, volume 29. Curran Associates, Inc., 2016.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing frontiers in open language model post-training, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. The flan collection: Designing data and methods for effective instruction tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22631–22648. PMLR, 23–29 Jul 2023.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts, 2017.

Meta. Introducing meta llama 3: The most capable openly available llm to date., 2024. URL https://ai.meta.com/blog/meta-llama-3.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Pan, J., Zhang, J., Wang, X., and Yuan, L. Tinyzero, 2025. URL https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

Paster, K., Santos, M. D., Azerbayev, Z., and Ba, J. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

Qwen Team. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024a.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024b. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Rana, A. Common crawl – building an open web-scale crawl using hadoop, 2010. URL https://www.slideshare.net/hadoopusergroup/common-crawlpresentation.

Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dao, T. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., KaShun, S., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

Feng, X., Wan, Z., Wen, M., McAleer, S. M., Wen, Y., Zhang, W., and Wang, J. Alphazero-like tree-search can guide large language model decoding and training, 2023.

Gao, Z., Wang, H., Lu, C., Lu, T., Froudist-Walsh, S., Chen, M., Wang, X.-J., Hu, J., and Sun, W. The neural basis of delayed gratification. *Science Advances*, 7 (49):eabg6611, 2021. doi: 10.1126/sciadv.abg6611.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Hou, Z., Lv, X., Lu, R., Zhang, J., Li, Y., Yao, Z., Li, J., Tang, J., and Dong, Y. Advancing language model reasoning through reinforcement learning and inference scaling, 2025.

Hu, J. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Hu, J., Wu, X., Zhu, Z., Xianyu, Wang, W., Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework, 2024.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. R. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025.

Lamb, A. M., ALIAS PARTH GOYAL, A. G., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. Professor forcing: A new algorithm for training recurrent networks. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tulu 3: Pushing frontiers in open language model post-training, 2024.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Singh, A., Co-Reyes, J. D., Agarwal, R., Anand, A., Patil, P., Liu, P. J., Harrison, J., Lee, J., Xu, K., Parisi, A., et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Tong, Y., Zhang, X., Wang, R., Wu, R., and He, J. DART-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a.

Wang, Z., Li, Y., Wu, Y., Luo, L., Hou, L., Yu, H., and Shang, J. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision, 2024b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuan, Z., Yuan, H., Li, C., Dong, G., Tan, C., and Zhou, C. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Yue, X., Zheng, T., Zhang, G., and Chen, W. Mammoth2: Scaling instructions from the web. *NeurIPS 2024*, 2024b.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Zeng, W., Huang, Y., Liu, W., He, K., Liu, Q., Ma, Z., and He, J. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason, 2025. Notion Blog.

Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. The flan collection: Designing data and methods for effective instruction tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22631–22648. PMLR, 23–29 Jul 2023.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts, 2017.

Meta. Introducing meta llama 3: The most capable openly available llm to date., 2024. URL https://ai.meta.com/blog/meta-llama-3.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022.

Pan, J., Zhang, J., Wang, X., and Yuan, L. Tinyzero, 2025. URL https://github.com/Jiayi-Pan/TinyZero. Accessed: 2025-01-24.

Paster, K., Santos, M. D., Azerbayev, Z., and Ba, J. Openwebmath: An open dataset of high-quality mathematical web text, 2023.

Qwen Team. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024a.

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, 2024b. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Rana, A. Common crawl – building an open web-scale crawl using hadoop, 2010. URL https://www.slideshare.net/hadoopusergroup/common-crawlpresentation.

Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506, 2020.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Singh, A., Co-Reyes, J. D., Agarwal, R., Anand, A., Patil, P., Liu, P. J., Harrison, J., Lee, J., Xu, K., Parisi, A., et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zheng, L., Yin, L., Xie, Z., Sun, C., Huang, J., Yu, C. H., Cao, S., Kozyrakis, C., Stoica, I., Gonzalez, J. E., Barrett, C., and Sheng, Y. SGLang: Efficient execution of structured language model programs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Tong, Y., Zhang, X., Wang, R., Wu, R., and He, J. DART-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., and Chen, W. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a.

Wang, Z., Li, Y., Wu, Y., Luo, L., Hou, L., Yu, H., and Shang, J. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision, 2024b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., ichter, b., Xia, F., Chi, E., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuan, Z., Yuan, H., Li, C., Dong, G., Tan, C., and Zhou, C. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yue, X., Qu, X., Zhang, G., Fu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. MAmmoTH: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Yue, X., Zheng, T., Zhang, G., and Chen, W. Mammoth2: Scaling instructions from the web. *NeurIPS 2024*, 2024b.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Zeng, W., Huang, Y., Liu, W., He, K., Liu, Q., Ma, Z., and He, J. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. https://hkust-nlp.notion.site/simplerl-reason, 2025. Notion Blog.

Zheng, L., Yin, L., Xie, Z., Sun, C., Huang, J., Yu, C. H., Cao, S., Kozyrakis, C., Stoica, I., Gonzalez, J. E., Barrett, C., and Sheng, Y. SGLang: Efficient execution of structured language model programs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

A. Related Work

复杂的推理和链式思维提示。 大型语言模型（LLMs）在各种自然语言处理任务中展示了显著的能力，包括复杂推理。提高LLM推理能力的一个重要进展是实施链式思维（CoT）提示 (Wei et al., 2022)。这项技术涉及引导模型生成中间推理步骤，从而提高它们在需要逻辑演绎和多步骤问题解决的任务上的表现。初步研究 (Lambert et al., 2024; Wei et al., 2022; Longpre et al., 2023; Yu et al., 2024) 专注于短CoT，即模型生成简洁的推理路径以得出解决方案。尽管对简单问题有效，但短CoT在处理需要更深入思考的复杂任务时可能具有局限性。OpenAI的o1系列模型 (OpenAI, 2024) 首次通过增加CoT推理过程的长度引入了推理时间扩展。这种方法通过将复杂问题分解为更细的步骤并在解决问题时进行反思，帮助LLMs应对复杂问题，从而得出更准确和全面的解决方案。在本研究中，我们通过识别使模型能够表现出这种行为的关键因素来探索长CoT，鼓励高级推理能力。

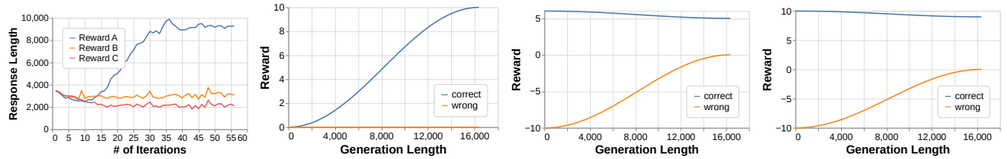
用于LLM的强化学习。 强化学习（RL）已被证明在提高LLM在各个领域的性能方面非常有效。RL技术，如基于人类反馈的强化学习（RLHF），使模型输出与人类偏好对齐，提高连贯性 (Ouyang et al., 2022)。最近的研究 (Kimi Team, 2025; DeepSeek-AI, 2025; Lambert et al., 2024) 利用RL使LLMs能够自主探索复杂问题的推理路径。DeepSeek-R1 (DeepSeek-AI, 2025) 在数学、编程和推理任务中表现出色，而无需依赖训练好的奖励模型 (Lightman et al., 2024; Wang et al., 2024b) 或树搜索 (Feng et al., 2023; Snell et al., 2024)。值得注意的是，即使在没有监督微调的基础模型中，这种能力也会出现，尽管以牺牲输出可读性为代价。同样，Kimi K1.5 (Kimi Team, 2025) 通过RL增强了通用推理能力，重点关注多模态推理和控制思维过程的长度。这些研究突显了RL在优化推理中的作用，特别是在中间步骤难以监督且只有最终结果可验证的情况下。我们的研究采用了类似的设置，但在不同训练条件和初始化策略下，更详细地探讨了不同模型行为的出现。

A. Related Work

Complex reasoning and chain of thought prompting. Large Language Models (LLMs) have demonstrated remarkable capabilities in various natural language processing tasks, including complex reasoning. A significant advancement in improving LLM reasoning ability is the implementation of Chain of Thought (CoT) prompting (Wei et al., 2022). This technique involves guiding models to generate intermediate reasoning steps, thereby improving their performance on tasks that require logical deduction and multistep problem solving. Initial studies (Lambert et al., 2024; Wei et al., 2022; Longpre et al., 2023; Yu et al., 2024) focused on short CoT, where models produce concise reasoning paths to arrive at solutions. Although effective for straightforward problems, short CoT can be limiting when addressing more intricate tasks that necessitate deeper deliberation. OpenAI' s o1 (OpenAI, 2024) series models were the first to introduce inference-time scaling by increasing the length of the CoT reasoning process. This approach helps LLMs tackle complex problems by breaking them into finer steps and reflecting during problem-solving, leading to more accurate and comprehensive solutions. In this work, we explore long CoT by identifying key factors that enable models to exhibit this behavior, encouraging advanced reasoning capabilities.

Reinforcement learning for LLM. Reinforcement Learning (RL) has proven effective in enhancing LLM performance across domains. RL techniques, such as Reinforcement Learning from Human Feedback (RLHF), align model outputs with human preferences, improving coherence (Ouyang et al., 2022). Recent studies (Kimi Team, 2025; DeepSeek-AI, 2025; Lambert et al., 2024) leverage RL to enable LLMs to explore reasoning paths autonomously for complex problems. DeepSeek-R1 (DeepSeek-AI, 2025) achieves strong performance in mathematics, coding, and reasoning tasks without relying on a trained reward model (Lightman et al., 2024; Wang et al., 2024b) or tree search (Feng et al., 2023; Snell et al., 2024). Notably, this capability emerges even in base models without supervised fine-tuning, albeit at the cost of output readability. Similarly, Kimi K1.5 (Kimi Team, 2025) enhances general reasoning with RL, focusing on multimodal reasoning and controlling thought process length. These works highlight RL' s role in optimizing reasoning when intermediate steps are hard to supervise, and only final outcomes are verifiable. Our research share a similar setup but with more detail on disentangling how different model behaviors emerge under varying training conditions and initialization strategies.

B. Figures and Tables



(a) Response lengths under different Cosine Rewards (b) Reward A (c) Reward B (d) Reward C

Figure 9. (a) 调整余弦奖励的超参数会导致不同的长度缩放行为。注意，奖励A由于模型在上下文窗口内停止的能力降低，导致下游任务的性能下降。 (b) 奖励A: $r_0^c = 0, r_L^c = 10, r_0^w = r_L^w = 0$， (c) 奖励B: $r_0^c = 6, r_L^c = 5, r_0^w = -10, r_L^w = 0$ (d) 奖励C: $r_0^c = 10, r_L^c = 9, r_0^w = -10, r_L^w = 0$.

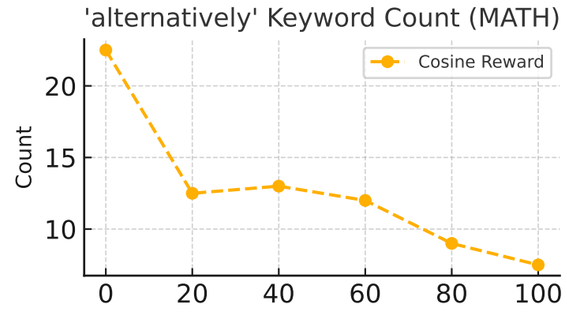
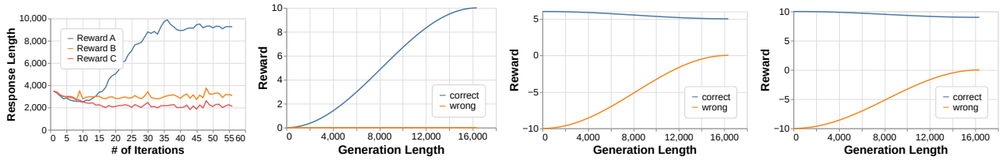


Figure 10. CoT 分支频率，通过枢轴词 "alternatively," 的关键词计数估计，在更多的训练计算下，随着余弦奖励的使用而减少。我们将此现象以及重复增加归因于奖励劫持。

Figure 11. 在不同 γ_c 值下的 CoT 分支频率。降低折扣因子增加了分支频率，导致模型更快地放弃问题解决方法。

B. Figures and Tables



(a) Response lengths under different Cosine Rewards (b) Reward A (c) Reward B (d) Reward C

Figure 9. (a) Tuning the hyperparameters of the Cosine Reward results in different length scaling behavior. Note that Reward A results in some performance degradation on downstream tasks due to the model's reduced ability to stop within the context window. (b) Reward A: $r_0^c = 0, r_L^c = 10, r_0^w = r_L^w = 0$, (c) Reward B: $r_0^c = 6, r_L^c = 5, r_0^w = -10, r_L^w = 0$ (d) Reward C: $r_0^c = 10, r_L^c = 9, r_0^w = -10, r_L^w = 0$.

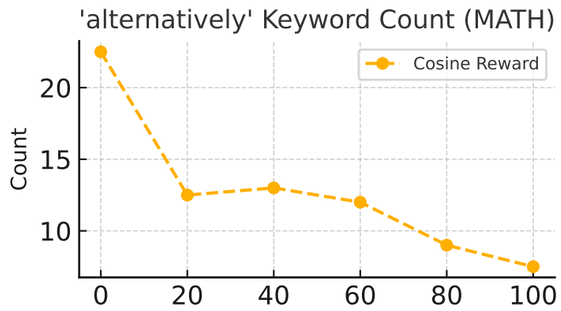


Figure 10. CoT branching frequency, estimated by the keyword count of the pivot word "alternatively,", decreased under the Cosine Reward with more training compute. We attributed this, along with increased repetition, to reward hacking.

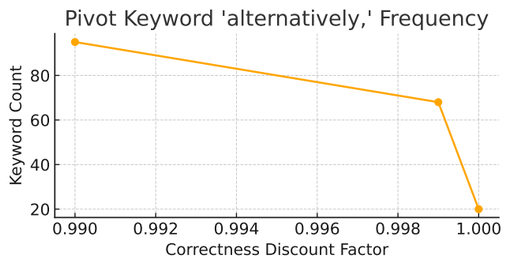


Figure 11. Branching frequency in CoT at different γ_c values. Lowering the discount factor increased branching frequency, causing the model to abandon problem-solving approaches more quickly.

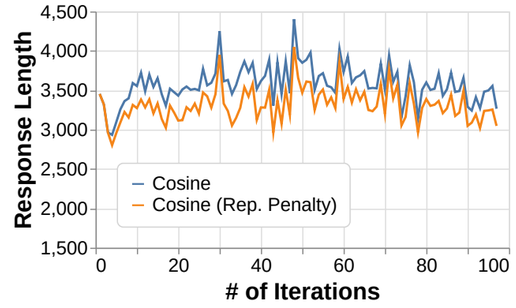


Figure 12. 使用余弦奖励训练的模型的响应长度，有和没有重复惩罚的情况。我们发现重复惩罚减少了响应长度。

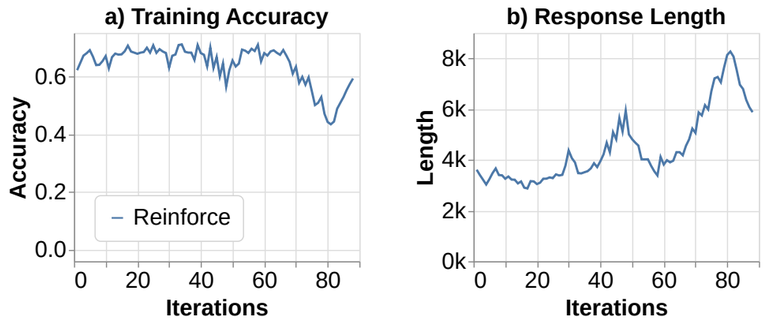


Figure 13. 使用经典奖励的强化学习显示出训练不稳定性。

Table 5. 使用不同折扣因子训练模型在正确性（余弦）奖励和重复惩罚下的性能。我们发现不同类型的奖励具有不同的最优值。

Correctness Discount	Repetition Discount	MATH -500	AIME 2024	Theo. QA	MMLU -Pro-1k
SFT		50.4	3.5	20.6	32.4
1.000	1.000	55.7	**5.0**	25.7	34.5
	0.999	**58.0**	4.6	**26.0**	**36.5**
	0.99	57.8	3.8	24.5	33.3
0.999	0.999	53.5	2.1	19.5	30.7
	0.99	55.2	1.7	18.5	32.0
0.99	0.99	47.9	0.2	15.6	25.5

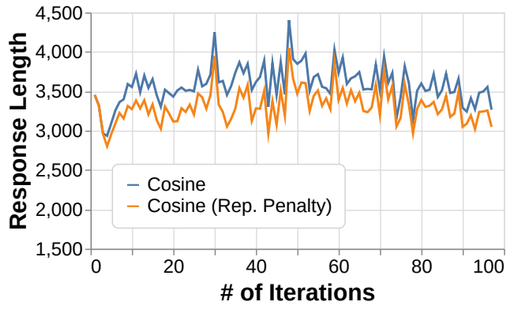


Figure 12. Training response length of models trained with Cosine Reward with and without repetition penalty. We see that repetition penalty reduced the length.

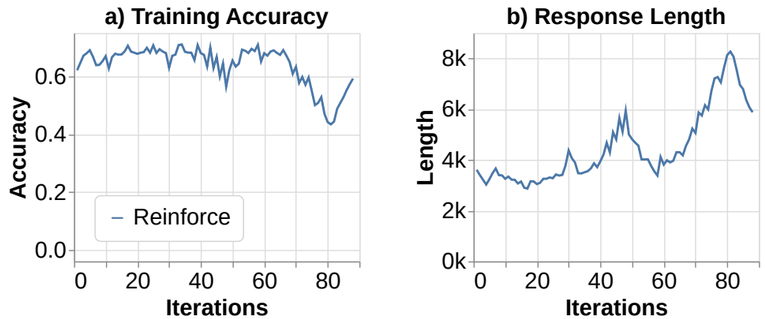


Figure 13. Reinforce with classic reward shows signs of training instability.

Table 5. Performance of model trained with different discount factors for the correctness (cosine) reward and repetition penalty. We see that different reward types have different optimal values.

Correctness Discount	Repetition Discount	MATH -500	AIME 2024	Theo. QA	MMLU -Pro-1k
SFT		50.4	3.5	20.6	32.4
1.000	1.000	55.7	**5.0**	25.7	34.5
	0.999	**58.0**	4.6	**26.0**	**36.5**
	0.99	57.8	3.8	24.5	33.3
0.999	0.999	53.5	2.1	19.5	30.7
	0.99	55.2	1.7	18.5	32.0
0.99	0.99	47.9	0.2	15.6	25.5

C. Algorithms and Formulas

C.1. Cosine Reward Formula

$$\textbf{CosFn}(t, T, \eta_{min}, \eta_{max}) = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{t\pi}{T})) \tag{1}$$

上述公式通常在梯度下降优化过程中用作学习率调度。它由(Loshchilov & Hutter, 2017)引入。

C.2. N-gram Repetition Penalty

Algorithm 1 N-gram Repetition Penalty

Input:
  s : sequence of tokens
  l : sequence length
  N : n-gram size
  P : penalty value
  m : maximum sequence length
Output: $r \in \mathbb{R}^m$
$seq \leftarrow s[1:l]$ {Extract subsequence of length l}
$ngrams \leftarrow \emptyset$ {Set of observed n-grams}
$r \leftarrow \vec{0} \in \mathbb{R}^m$ {Initialize reward vector}
for $j \leftarrow 1$ to $|seq| - N + 1$ **do**
  $ng \leftarrow (seq[j], seq[j+1], ..., seq[j+N-1])$ {Current n-gram}
  if $ng \in ngrams$ **then**
    for $t \leftarrow j$ to $j + N - 1$ **do**
      $r[t] \leftarrow P$ {Apply penalty}
    end for
  end if
  $ngrams \leftarrow ngrams \cup \{ng\}$
end for
Output: r

C. Algorithms and Formulas

C.1. Cosine Reward Formula

$$\textbf{CosFn}(t, T, \eta_{min}, \eta_{max}) = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + \cos(\frac{t\pi}{T})) \tag{1}$$

The formula above is commonly used as the learning rate schedule during gradient descent optimization. It was introduced by (Loshchilov & Hutter, 2017).

C.2. N-gram Repetition Penalty

Algorithm 1 N-gram Repetition Penalty

Input:
  s : sequence of tokens
  l : sequence length
  N : n-gram size
  P : penalty value
  m : maximum sequence length
Output: $r \in \mathbb{R}^m$
$seq \leftarrow s[1:l]$ {Extract subsequence of length l}
$ngrams \leftarrow \emptyset$ {Set of observed n-grams}
$r \leftarrow \vec{0} \in \mathbb{R}^m$ {Initialize reward vector}
for $j \leftarrow 1$ to $|seq| - N + 1$ **do**
  $ng \leftarrow (seq[j], seq[j+1], ..., seq[j+N-1])$ {Current n-gram}
  if $ng \in ngrams$ **then**
    for $t \leftarrow j$ to $j + N - 1$ **do**
      $r[t] \leftarrow P$ {Apply penalty}
    end for
  end if
  $ngrams \leftarrow ngrams \cup \{ng\}$
end for
Output: r

D. Extracts

Short-Term Thinking ($\gamma_c = 0.99$)

But in this case, we saw that the fixed points don't satisfy the equation

$$p(p(x)) = x.$$

Wait, maybe the solutions to $p(p(x)) = x$ are not the fixed points of p, but something else.
Alternatively, perhaps I can consider that $p(p(x)) = x$ implies that p is a bijection, but since $p(x)$ is quadratic, it's not necessarily invertible over all real numbers.
Wait, but the problem says that $p(p(x)) = x$ has four real solutions, so perhaps there are four values of x that satisfy this equation.
Alternatively, perhaps I need to find the roots of

$$p(p(x)) - x = 0,$$

and solve for x. But that seems complicated. Maybe there's a better way.

D. Extracts

Short-Term Thinking ($\gamma_c = 0.99$)

But in this case, we saw that the fixed points don't satisfy the equation

$$p(p(x)) = x.$$

Wait, maybe the solutions to $p(p(x)) = x$ are not the fixed points of p, but something else.
Alternatively, perhaps I can consider that $p(p(x)) = x$ implies that p is a bijection, but since $p(x)$ is quadratic, it's not necessarily invertible over all real numbers.
Wait, but the problem says that $p(p(x)) = x$ has four real solutions, so perhaps there are four values of x that satisfy this equation.
Alternatively, perhaps I need to find the roots of

$$p(p(x)) - x = 0,$$

and solve for x. But that seems complicated. Maybe there's a better way.

E. Experimental Setup

E.1. Evaluation Setup

基准测试 以下是我们的评估基准测试的详细信息：

- **MATH-500** (Hendrycks et al., 2021): 一个领域内的数学推理基准。MATH 包含了来自美国高中数学竞赛的 12,500 个问题。为了效率，我们采用了 MATH-500，这是其测试分割的一个广泛使用的 i.i.d. 子集。
- **AIME 2024**：一个超出领域范围的数学推理基准，包含2024年美国邀请数学考试（AIME）的30个问题。
- **TheoremQA** (Chen et al., 2023)：一个跨领域的STEM推理基准，包含800个样本。它涵盖了数学、电子工程与计算机科学、物理和金融等领域的350多个定理。
- **MMLU-Pro-1k** (Wang et al., 2024a): 一个领域外的通用推理基准。MMLU-Pro 包含来自学术考试和教科书的 12,000 多个问题，涵盖了包括生物学、商业、化学、计算机科学、经济学、工程学、健康、历史、法律、数学、哲学、物理、心理学和其他在内的 14 个不同领域。为了效率，我们采用了其测试集的一个 1,000 个样本的 i.i.d. 子集，称为 MMLU-Pro-1k。我们尽量保持分布与原始分布相同。图 14 显示了下采样前后的分布。

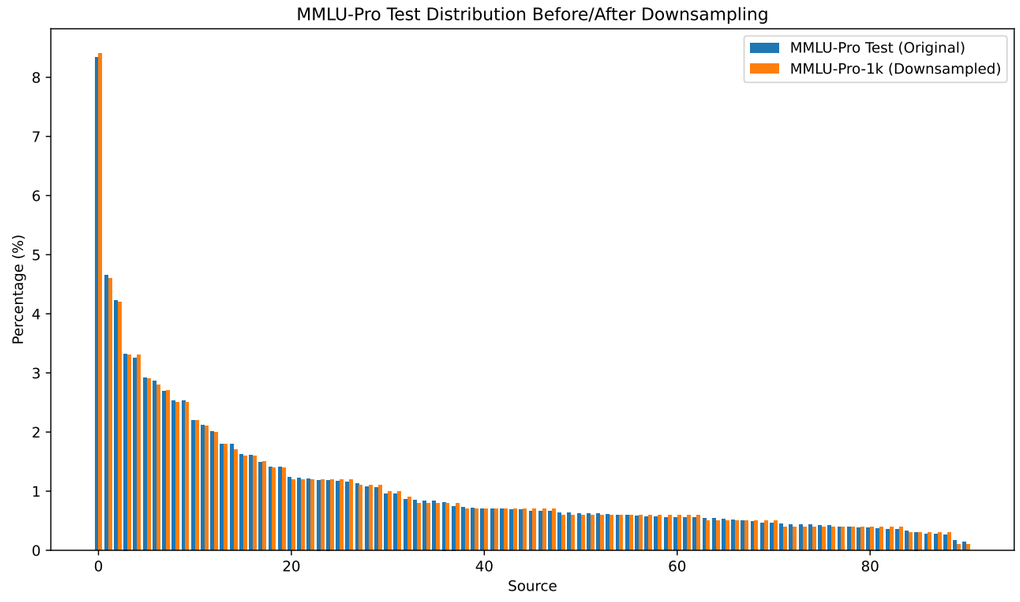


Figure 14. MMLU-Pro 测试集在下采样前/后的分布，针对 MMLU-Pro-1k 子集。该子集与完整数据集是独立同分布的。

统计指标 我们使用至少4个随机种子计算平均准确率。为了控制AIME 2024小样本量导致的方差，我们每个提示采样16个响应。

E. Experimental Setup

E.1. Evaluation Setup

Benchmarks Below are details of our evaluation benchmarks:

- **MATH-500** (Hendrycks et al., 2021): an in-domain mathematical reasoning benchmark. MATH consists of 12,500 problems from American high school math competitions. For efficiency, we adopt MATH-500, a widely-used i.i.d. subset of its test split.
- **AIME 2024**: an out-of-domain mathematical reasoning benchmark consisting of the 30 problems from American Invitational Mathematics Examination (AIME) 2024.
- **TheoremQA** (Chen et al., 2023): an out-of-domain STEM reasoning benchmark consisting of 800 samples. It covers 350+ theorems spanning across Math, EE&CS, Physics and Finance.
- **MMLU-Pro-1k** (Wang et al., 2024a): an out-of-domain general reasoning benchmark. MMLU-Pro comprises over 12,000 questions from academic exams and textbooks, spanning 14 diverse domains including Biology, Business, Chemistry, Computer Science, Economics, Engineering, Health, History, Law, Math, Philosophy, Physics, Psychology, and Others. For efficiency, we adopt an 1,000-sample i.i.d. subset of its test split, called MMLU-Pro-1k. We tried to keep the distribution identical to the original one. Figure 14 shows the distribution before/after the downsampling.

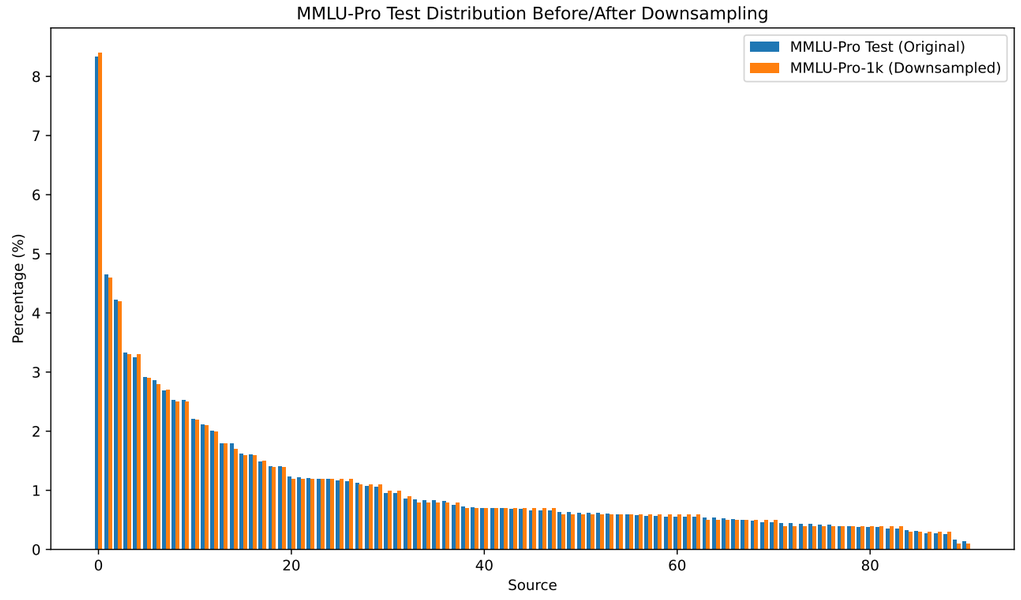


Figure 14. MMLU-Pro test distribution before/after downsampling for the MMLU-Pro-1k subset. The subset is i.i.d. to the full set.

实现 我们采用vLLM库加速推理，并使用SymEval[2]，这是一个能够处理复杂数学对象（如矩阵和函数）的精细答案评分器，与我们在强化学习设置中的采样和奖励实现保持一致。注意，一些强化学习实验是使用评分器的早期版本进行的，导致性能上存在细微差异。

E.2. Details about Distillation

为了从 `QwQ-32B-Preview` 中提取长的 CoT 轨迹，我们采用温度 $t = 1.0$，top-p 值为 0.95 以及最大输出长度为 8192 个 token。我们的初步实验表明，8192 个 token 在 MATH-500 上的准确率与 16384 个 token 几乎相同，但所需时间显著减少。

为了从 `Qwen2.5-Math-72B-Instruct` 中提取短的 CoT 轨迹，我们采用温度 $t = 0.7$，top-p 值为 0.95 以及最大输出长度为 4096 个 token，因为 `Qwen2.5-Math-72B-Instruct` 的上下文限制为 4096 个 token，并且我们的初步实验观察到使用 $t = 1.0$ 时会产生相当比例的无意义输出。

注意，数据是使用 SGLang (Zheng et al., 2024) 和我们代码的早期版本提取的。

在应用拒绝采样时，我们采用 SymEval 验证器作为评分器。

E.3. Details abour SFT Setup

我们使用OpenRLHF (Hu et al., 2024) 进行我们的SFT实验。默认情况下，我们采用表 6 中的SFT超参数。

为了提高效率，我们利用了基于DeepSpeed库 (Rasley et al., 2020) 的Flash Attention 2 (Dao, 2024) 和 ZeRO (Rajbhandari et al., 2020)。我们统一将微批次大小设置为1，因为增加它时我们没有观察到加速效果。

Table 6. SFT Hyperparameters

Batch Size	Context Length	LR	Epochs
256	128K	5e-6	2

E.4. Details about RL Setup

我们使用OpenRLHF (Hu et al., 2024) 进行我们的强化学习实验。在描述超参数时，我们采用了与OpenRLHF相同的命名约定。

E.5. Experiment Hyperparameters

请注意，下面的 `BS` 列同时指代 `rollout_batch_size`（在采样-训练迭代中使用的提示数量）和 `train_batch_size`（在训练更新中使用的样本数量），因为在我们的大多数强化学习设置中，这两个超参数采用相同的数值。此外，`Samples` 列指代每个提示的样本数量。

E.5.1. Details of Section 3.2 (SFT Initialization for RL)

SFT 数据：从 `QwQ-32B-Preview` 或 `Qwen2.5-Math-72B-Instruct` 中蒸馏出的 CoT 数据，使用 MATH 训练集的不同候选响应数量。

[2] https://github.com/tongyx361/symeval

Statistical Metrics We calculate the average accuracy with at least 4 random seeds. To tame the variance caused by the small size of AIME 2024, we sample 16 responses per prompt.

Implementation We adopt the vLLM library to accelerate the inference and SymEval[2], an elaborate answer grader capable of processing complex mathematical objects like matrices and functions, keeping consistent with the sampling and reward implementation in our RL setup. Note that a few RL experiments are carried out with an earlier version of the grader, causing nuanced performance differences.

E.2. Details about Distillation

To distill long CoT trajectories from `QwQ-32B-Preview`, we adopt the temperature $t = 1.0$, the top-p value of 0.95 and the maximum output length of 8192 tokens. Our preliminary experiments show that 8192 tokens show almost the same accuracy with `QwQ-32B-Preview` on MATH-500 as 16384 tokens, while costing significantly less time.

To distill short CoT trajectories from `Qwen2.5-Math-72B-Instruct`, we adopt the temperature $t = 0.7$, the top-p value of 0.95 and the maximum output length of 4096 tokens, since `Qwen2.5-Math-72B-Instruct` has a context limit of 4096 tokens and our preliminary experiments observe a non-negligible ratio of nonsense output when using $t = 1.0$.

Note the data is distilled with SGLang (Zheng et al., 2024) with an early version of our code.

When applying rejection sampling, we adopt the SymEval verifier as the grader.

E.3. Details abour SFT Setup

We use OpenRLHF (Hu et al., 2024) for our SFT experiments. By default, we adopt the SFT hyperparameters in Table 6.

For efficiency, we utilize Flash Attention 2 (Dao, 2024) and ZeRO (Rajbhandari et al., 2020) based on the DeepSpeed library (Rasley et al., 2020). We uniformly set the micro batch size as 1 since we don't observe acceleration when increasing it.

Table 6. SFT Hyperparameters

Batch Size	Context Length	LR	Epochs
256	128K	5e-6	2

E.4. Details about RL Setup

We use OpenRLHF (Hu et al., 2024) for our RL experiments. When describing hyperparameters, we adopt the same naming conventions as OpenRLHF.

[2] https://github.com/tongyx361/symeval

Table 7. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma = 1$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.2. Details of Section 4.1 (CoT Length Stability)

SFT 数据：从 `QwQ-32B-Preview` 中提取的长 CoT 数据，使用 MATH 训练集分割。

Table 8. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Qwen2.5-Math-7B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01

E.5. Experiment Hyperparameters

Note that the `BS` column below refers to both `rollout_batch_size` (the number of prompts used in a sampling-training iteration) and `train_batch_size` (the number of samples used in a training update) because we adopt the same number for these two hyperparameters in most of our RL setups. Also, the `Samples` column refers to the number of samples per prompt.

E.5.1. Details of Section 3.2 (SFT Initialization for RL)

SFT Data: CoT data distilled from `QwQ-32B-Preview` or `Qwen2.5-Math-72B-Instruct` with the MATH train split with different number of candidate responses per prompt.

Table 7. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma = 1$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.2. Details of Section 4.1 (CoT Length Stability)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 8. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Qwen2.5-Math-7B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01

E.5.3. Details of Section 4.2 (Active Scaling of CoT Length)

SFT 数据：从 `QwQ-32B-Preview` 中提炼的长 CoT 数据，使用 MATH 训练集分割。

Table 9. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.3. Details of Section 4.2 (Active Scaling of CoT Length)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 9. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Llama3.1-8B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.4. Details of Section 4.3 (Cosine Reward Hyperparameters)

SFT 数据：从 `QwQ-32B-Preview` 中提取的长 CoT 数据，使用 MATH 训练集。

Table 10. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = 0$ $r_L^c = +10$ $r_0^w = 0$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +6$ $r_L^c = +5$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +10$ $r_L^c = +9$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.4. Details of Section 4.3 (Cosine Reward Hyperparameters)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 10. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = 0$ $r_L^c = +10$ $r_0^w = 0$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +6$ $r_L^c = +5$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +10$ $r_L^c = +9$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.5. Details of Section 4.4 (Context Window Size)

SFT 数据：从 `QwQ-32B-Preview` 中提取的长 CoT 数据，使用 MATH 训练集。

Table 11. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 2048	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 6144	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.6. Details of Section 4.5 (Length Reward Hacking)

SFT 数据：从 `QwQ-32B-Preview` 中提炼的长 CoT 数据，使用 MATH 训练集分割。

E.5.5. Details of Section 4.4 (Context Window Size)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 11. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 2048	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 6144	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.6. Details of Section 4.5 (Length Reward Hacking)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 12. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.7. Details of Section 4.6 (Optimal Discount Factors)

SFT 数据：从 `QwQ-32B-Preview` 中蒸馏出的长 CoT 数据，使用 MATH 训练集。

Table 12. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma = 1$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	8	16	512	2	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.7. Details of Section 4.6 (Optimal Discount Factors)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 13. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 1$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.999$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 0.999$ $\gamma_p = 0.999$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 0.999$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 0.99$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

Table 13. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 1$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.999$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 0.999$ $\gamma_p = 0.999$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 0.999$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01
Llama3.1-8B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$	$\lambda = 1$ $\gamma_c = 0.99$ $\gamma_p = 0.99$	4	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6	0.01

E.5.8. Details of Section 5.2 (RL with Noisy Verifiable Data)

SFT 数据：从 `QwQ-32B-Preview` 中通过 WebInstruct 蒸馏出的 462k 条长 CoT 数据中筛选出 115k 条。

Table 14. Hyperparameters

Base Model	RL Prompt Set Verifier	Rewards	GAE	Episodes Instances	Samples	BS	Epochs	Context Length	LR KL
Llama3.1-8B	Unfiltered (30k sampled) Symeval	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Unfiltered (30k sampled) LLM-as-a-judge	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Filtered (30k sampled) Symeval	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Filtered (30k sampled) LLM-as-a-judge	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01

E.5.8. Details of Section 5.2 (RL with Noisy Verifiable Data)

SFT Data: 115k filtered from 462k instances of long CoT data distilled from `QwQ-32B-Preview` with WebInstruct.

Table 14. Hyperparameters

Base Model	RL Prompt Set Verifier	Rewards	GAE	Episodes Instances	Samples	BS	Epochs	Context Length	LR KL
Llama3.1-8B	Unfiltered (30k sampled) Symeval	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Unfiltered (30k sampled) LLM-as-a-judge	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Filtered (30k sampled) Symeval	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01
Llama3.1-8B	Filtered (30k sampled) LLM-as-a-judge	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma_c = 1$ $\gamma_p = 0.99$	1 30k instances	4	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 9e-6 KL: 0.01

E.5.9. Details of Section 6 (Exploration on RL from the Base Model)

Table 15. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Qwen2.5-Math-7B	Correct: $+1$ Wrong: -0.5 No Answer: -1	$\lambda = 0.95$ $\gamma = 1$	20	8	1024 (Train: 128)	1	Prompt: 1024 Gen: 3072	Actor: 5e-7 Critic: 9e-6	0.01
Qwen2.5-Math-7B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Qwen2.5-Math-7B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01

E.5.10. Details of Section 7.3 (REINFORCE 比 PPO 更难调参)

SFT 数据：从 `QwQ-32B-Preview` 中提炼的长 CoT 数据，使用 MATH 训练集。

Table 16. Hyperparameters

Base Model	Rewards	Gamma	Episodes	Samples	BS	Epochs	Context Length	LR	KL	Clip
Llama3.1-8B	Correct: $+1$	$\gamma = 1$	8 (stopped early)	8	512	1	Prompt: 2048 Gen: 14336	5e-7	0.01	0.1

E.6. Implementation of the Model-Based Verifier

我们使用了`Qwen2.5-7B-Instruct`作为基于模型的验证器。它被提供了参考答案和长CoT的后缀。我们截断了长CoT以避免混淆验证器。我们使用了以下提示。

E.5.9. Details of Section 6 (Exploration on RL from the Base Model)

Table 15. Hyperparameters

Base Model	Rewards	GAE	Episodes	Samples	BS	Epochs	Context Length	LR	KL
Qwen2.5-Math-7B	Correct: $+1$ Wrong: -0.5 No Answer: -1	$\lambda = 0.95$ $\gamma = 1$	20	8	1024 (Train: 128)	1	Prompt: 1024 Gen: 3072	Actor: 5e-7 Critic: 9e-6	0.01
Qwen2.5-Math-7B	Correct: $+1$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01
Qwen2.5-Math-7B	Cosine: $r_0^c = +2$ $r_L^c = +1$ $r_0^w = -10$ $r_L^w = 0$ $r_e = -10$ Rep. Penalty: $P = -0.05$ $N = 40$	$\lambda = 1$ $\gamma = 1$	8	8	512	1	Prompt: 2048 Gen: 14336	Actor: 5e-7 Critic: 4.5e-6	0.01

E.5.10. Details of Section 7.3 (REINFORCE is more tricky to tune than PPO)

SFT Data: Long CoT data distilled from `QwQ-32B-Preview` with the MATH train split.

Table 16. Hyperparameters

Base Model	Rewards	Gamma	Episodes	Samples	BS	Epochs	Context Length	LR	KL	Clip
Llama3.1-8B	Correct: $+1$	$\gamma = 1$	8 (stopped early)	8	512	1	Prompt: 2048 Gen: 14336	5e-7	0.01	0.1

E.6. Implementation of the Model-Based Verifier

We used `Qwen2.5-7B-Instruct` as our model-based verifier. It was provided with both the reference answer and the suffix of the long CoT. We truncated the long CoT to avoid confusing the verifier. We used the following prompt.

Prompt Template for Model-Based Verifier

```
Given the following last 20 lines of the LLM response to a math question
and the reference solution to that question, evaluate if the LLM response is correct
    based only on the LLM's final answer.

LLM response (last 20 lines):
...
{out}

Reference solution:
{ref}

Explain your thought process step-by-step before responding with 'Judgement: <
   correct/wrong/not_found>'
```

E.7. Implementation of Short-Form Answer Extraction

我们使用 `Llama-3.1-8B-Instruct` 模型从 WebInstruct 中的 QA 对中提取简短答案，使用以下提示模板：

Prompt Template for Short-Form Answer Extraction

```
Problem: {Problem}

Solution: {Solution}

Based on the Problem and the Solution, extract a short final answer that is easy to
   check.
Provide the short final answer in the format of "The final answer is $$
\boxed{...}
$$"
- If the answer is a mathematical object, write it in LaTeX, e.g., "The final answer
    is $$
\boxed{\frac{1}{2}}
$$"
- If the answer is a boolean, write it as "True" or "False", e.g., "The final answer
    is $$
\boxed{True}
$$"
- If the Problem can't be answered in a short form, respond with "" like "The final
   answer is $$
\boxed{}
$$"
```

对于生成参数，我们使用温度 $t = 0$（贪婪解码）并将最大输出长度设置为512个标记。

Prompt Template for Model-Based Verifier

```
Given the following last 20 lines of the LLM response to a math question
and the reference solution to that question, evaluate if the LLM response is correct
     based only on the LLM's final answer.

LLM response (last 20 lines):
...
{out}

Reference solution:
{ref}

Explain your thought process step-by-step before responding with 'Judgement: <
    correct/wrong/not_found>'
```

E.7. Implementation of Short-Form Answer Extraction

We use the `Llama-3.1-8B-Instruct` model to extract short-form answer from QA pairs in WebInstruct, with the following prompt template:

Prompt Template for Short-Form Answer Extraction

```
Problem: {Problem}

Solution: {Solution}

Based on the Problem and the Solution, extract a short final answer that is easy to
    check.
Provide the short final answer in the format of "The final answer is $$
\boxed{...}
$$"
- If the answer is a mathematical object, write it in LaTeX, e.g., "The final answer
     is $$
\boxed{\frac{1}{2}}
$$"
- If the answer is a boolean, write it as "True" or "False", e.g., "The final answer
     is $$
\boxed{True}
$$"
- If the Problem can't be answered in a short form, respond with "" like "The final
    answer is $$
\boxed{}
$$"
```

For generation parameters, we use temperature $t = 0$ (greedy decoding) and set the maximum output length as 512

生成后，我们简单地从 `\boxed{...}` 中提取简短答案。

E.8. Action Prompting Framework

我们研究了公开发布的 `o1-preview` 的 CoTs，并确定其思维可以归类为几种类型的动作（如下所列）。为了构建长的 CoTs，我们为每种动作设计了提示，并实现了一个多步骤提示框架来对它们进行排序。该框架将 CoT 的控制流交给了 LLM，由 LLM 做出分支或循环决策，而框架则更被动地作为一个状态机对 LLM 的输出做出反应。框架负责围绕构建 CoT 的样板代码，使用仅追加的日志，并管理所有的编排。

- `clarify`: 针对问题进行一些观察，以便确定解决问题的方法。
- 分解: 将当前问题分解为更小、更容易解决的子问题。
- `solution_step`: 计算解决方案中的单个步骤。在数学的背景下，这可能是进行一些算术运算或符号操作。
- `reflection`: 评估当前方法和部分解决方案，以查看是否犯了任何错误，是否实现了任何子目标，或者是否应考虑替代方法。请注意，我们为 `reflection` 行动使用了一个强大的教师模型 `o1-mini`，因为该行动更难正确响应，需要自我纠正。
- `answer`: 以最终答案回应并终止链式思考。

E.8.1. Control Flow

框架与LLM之间交互的简化描述。

tokens.

After generation, we simply extract the short-form answer from within the `\boxed{...}`.

E.8. Action Prompting Framework

We studied the publicly released CoTs of `o1-preview` and identified that its thoughts could be categorized into a few types of actions (listed below). To construct long CoTs, we designed prompts for each of these actions and implemented a multi-step prompting framework to sequence them. The framework ceded control flow of the CoT to the LLM, with the LLM making branching or looping decisions while the framework acted more passively as a state machine reacting to the LLM outputs. The framework took care of the boilerplate around constructing the CoT with an append-only log and managed all of the orchestration.

- `clarify`: Making some observations about the problem in order to identify an approach to solve it.
- `decompose`: Breaking the current problem down into smaller and easier sub-problems to solve.
- `solution_step`: Computing a single step in the solution. In the context of math, this could be doing some arithmetic or symbolic manipulation.
- `reflection`: Evaluating the current approach and partial solution to see if any mistakes were made, any sub-goals were achieved, or if alternative approaches should be considered instead. Note that we used a strong teacher model `o1-mini` for the `reflection` action as that one was a more difficult prompt to respond to correctly as it requires self-correction.
- `answer`: Responding with a final answer and terminating the CoT.

E.8.1. Control Flow

Simplified description of the interaction between the framework and LLM.

Algorithm 2 Action Prompting State Machine

Input: $prompt$
Output: $chain\_of\_thought$ sequence
$chain\_of\_thought \leftarrow [prompt]$ {Initialize singleton chain of thought sequence from prompt}
$state \leftarrow$ "clarify"
while True **do**
  if $state =$ "clarify" **then**
    $output \leftarrow$ prompt_action_clarify()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "decompose" **then**
    $output \leftarrow$ prompt_action_decompose()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "solution_step" **then**
    $output \leftarrow$ prompt_action_solution_step()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "reflection" **then**
    $output \leftarrow$ prompt_action_reflection()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "answer" **then**
    $output \leftarrow$ prompt_action_answer()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
    return $chain\_of\_thought$ {Terminate after answer action}
  end if
end while

Algorithm 2 Action Prompting State Machine

Input: $prompt$
Output: $chain\_of\_thought$ sequence
$chain\_of\_thought \leftarrow [prompt]$ {Initialize singleton chain of thought sequence from prompt}
$state \leftarrow$ "clarify"
while True **do**
  if $state =$ "clarify" **then**
    $output \leftarrow$ prompt_action_clarify()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "decompose" **then**
    $output \leftarrow$ prompt_action_decompose()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "solution_step" **then**
    $output \leftarrow$ prompt_action_solution_step()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "reflection" **then**
    $output \leftarrow$ prompt_action_reflection()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
  else if $state =$ "answer" **then**
    $output \leftarrow$ prompt_action_answer()
    $(state, thought) \leftarrow$ parse$(output)$
    $chain\_of\_thought$.append$(thought)$
    return $chain\_of\_thought$ {Terminate after answer action}
  end if
end while

E.8.2. Action Prompting Templates

Action: Clarify

```
You are a very talented mathematics professor.
In a few sentences, VERY CONCISELY rephrase the problem to clarify its meaning and
    explicitly state what needs to be solved. Highlight any assumptions, constraints
     and potential misinterpretations.
Do NOT attempt to solve the problem yet -- you are just clarifying the problem in
    your mind.

<problem>
{goal}
</problem>

Answer in the following format:

<clarification>
Problem clarification as instructed above
</clarification>
<goal>
Summarize the problem into a single statement describing the goal, e.g. Find the
    value of the variable w.
</goal>
```

E.8.2. Action Prompting Templates

Action: Clarify

```
You are a very talented mathematics professor.
In a few sentences, VERY CONCISELY rephrase the problem to clarify its meaning and
    explicitly state what needs to be solved. Highlight any assumptions, constraints
     and potential misinterpretations.
Do NOT attempt to solve the problem yet -- you are just clarifying the problem in
    your mind.

<problem>
{goal}
</problem>

Answer in the following format:

<clarification>
Problem clarification as instructed above
</clarification>
<goal>
Summarize the problem into a single statement describing the goal, e.g. Find the
    value of the variable w.
</goal>
```


Action: Decompose

```
You are a talented mathematics professor.
You already have a partial solution to a problem.
In a single sentence, propose candidates for the next subgoal as the next step of
    the partial solution that will help you make progress towards the current goal.
Do not repeat any subgoal, we don't want any infinite loops!
Do not suggest using a computer or software tools.

<current goal>
{current_goal}
</current goal>
<parent goal>
{parent_goal}
</parent goal>
<partial solution>
{solution}
</partial solution>

Format your answer as follows:

<thinking>
step-by-step thinking of what the next possible subgoal should be, as well as some
    other alternatives that might also work
remember, we want to solve the parent goal WITHOUT repeating the subgoals that are
    already DONE.
do not suggest verification or checking.
{parent_goal}
</thinking>
<sentence>
single sentence describing the subgoal
phrase it as if you were thinking to yourself and are considering this as a
    hypothesis (don't express too much certainty)
</sentence>
<sentence>
single sentence describing an *ALTERNATIVE* subgoal, without repeating previous ones
start off with "Alternatively,"
</sentence>
<sentence>
single sentence describing an *ALTERNATIVE* subgoal, without repeating previous ones
start off with "Alternatively,"
</sentence>
```

Action: Decompose

```
You are a talented mathematics professor.
You already have a partial solution to a problem.
In a single sentence, propose candidates for the next subgoal as the next step of
    the partial solution that will help you make progress towards the current goal.
Do not repeat any subgoal, we don't want any infinite loops!
Do not suggest using a computer or software tools.

<current goal>
{current_goal}
</current goal>
<parent goal>
{parent_goal}
</parent goal>
<partial solution>
{solution}
</partial solution>

Format your answer as follows:

<thinking>
step-by-step thinking of what the next possible subgoal should be, as well as some
    other alternatives that might also work
remember, we want to solve the parent goal WITHOUT repeating the subgoals that are
    already DONE.
do not suggest verification or checking.
{parent_goal}
</thinking>
<sentence>
single sentence describing the subgoal
phrase it as if you were thinking to yourself and are considering this as a
    hypothesis (don't express too much certainty)
</sentence>
<sentence>
single sentence describing an *ALTERNATIVE* subgoal, without repeating previous ones
start off with "Alternatively,"
</sentence>
<sentence>
single sentence describing an *ALTERNATIVE* subgoal, without repeating previous ones
start off with "Alternatively,"
</sentence>
```


Action: Solution Step

```
You are an extremely PEDANTIC mathematics professor who loves to nitpick.
You already have a partial solution to a problem. Your task is to solve *only* the
    current goal.
You should include symbols and numbers in every sentence if possible.

<current goal>
{current_goal}
</current goal>
<partial solution>
{solution}
</partial solution>

BE VERY CONCISE. Include calculations and equations in your response if possible,
    and make sure to solve them instead of just describing them.
DO NOT SOLVE THE WHOLE QUESTION, JUST THE CURRENT GOAL: {current_goal}
Do not repeat any calculations that were already in this prior step:
{prior_step}
```

Action: Solution Step

```
You are an extremely PEDANTIC mathematics professor who loves to nitpick.
You already have a partial solution to a problem. Your task is to solve *only* the
    current goal.
You should include symbols and numbers in every sentence if possible.

<current goal>
{current_goal}
</current goal>
<partial solution>
{solution}
</partial solution>

BE VERY CONCISE. Include calculations and equations in your response if possible,
    and make sure to solve them instead of just describing them.
DO NOT SOLVE THE WHOLE QUESTION, JUST THE CURRENT GOAL: {current_goal}
Do not repeat any calculations that were already in this prior step:
{prior_step}
```


Action: Reflection

```
You are a talented mathematics professor.
You already have a partial solution to a math problem.
Verify whether the current subgoal has been achieved.

<current goal>
{current_goal}
</current goal>
{parent_goal}
<partial solution>
{solution}
</partial solution>

Format your answer as follows:

<verification>
Come up with a quick, simple and easy calculation to double check that the solution
    is correct.
This calculation should not re-compute the solution in the same way, as that would
    defeat the purpose of double-checking.
Use one of the following strategies:
- An easier, alternative method to arrive at the answer
- Substituting specific values into equations and checking for consistency
- Working backwards from the answer to derive the given inputs and then checking for
     consistency
Be consise. Do not suggest using a computer.
At the end of your verification, restate the answer from the current solution. Do
    not calculate it if it hasn't been solved.
Phrase it as if you are reflecting as you solve the problem.
</verification>
<current_goal_achieved>
true or false, depending on whether the solution is correct and the current goal has
     been achieved: {current_goal}
</current_goal_achieved>
<parent_goal_achieved>
true or false, depending on whether the parent goal has been achieved:
{parent_goal.target}
</parent_goal_achieved>
<new_goal>
If the solution is not correct or the current goal has not been achieved, suggest an
     alternative current goal here in a single sentence.
Start off with "Alternatively,"
Your goal should be sufficiently different from subgoals that have been solved or
    that have timed out:
{parent_goal_tree}
</new_goal>
```

Action: Reflection

```
You are a talented mathematics professor.
You already have a partial solution to a math problem.
Verify whether the current subgoal has been achieved.

<current goal>
{current_goal}
</current goal>
{parent_goal}
<partial solution>
{solution}
</partial solution>

Format your answer as follows:

<verification>
Come up with a quick, simple and easy calculation to double check that the solution
    is correct.
This calculation should not re-compute the solution in the same way, as that would
    defeat the purpose of double-checking.
Use one of the following strategies:
- An easier, alternative method to arrive at the answer
- Substituting specific values into equations and checking for consistency
- Working backwards from the answer to derive the given inputs and then checking for
     consistency
Be consise. Do not suggest using a computer.
At the end of your verification, restate the answer from the current solution. Do
    not calculate it if it hasn't been solved.
Phrase it as if you are reflecting as you solve the problem.
</verification>
<current_goal_achieved>
true or false, depending on whether the solution is correct and the current goal has
     been achieved: {current_goal}
</current_goal_achieved>
<parent_goal_achieved>
true or false, depending on whether the parent goal has been achieved:
{parent_goal.target}
</parent_goal_achieved>
<new_goal>
If the solution is not correct or the current goal has not been achieved, suggest an
     alternative current goal here in a single sentence.
Start off with "Alternatively,"
Your goal should be sufficiently different from subgoals that have been solved or
    that have timed out:
{parent_goal_tree}
</new_goal>
```


Action: Answer

```
Extract the final answer, making sure to obey the formatting instructions.
Solution:
{solution}

Formatting instructions:
{format}
```

Action: Answer

```
Extract the final answer, making sure to obey the formatting instructions.
Solution:
{solution}

Formatting instructions:
{format}
```


F. Long CoT Patterns in Pre-training Data

F.1. Snapshot of webpages

以下两个示例展示了如何在回答问题后进行显式验证可以自然地存在于网页上。

Explicit verification

$x + 7 = 10$

This problem can be solved by subtracting 7 from each side.

$x + 7 - 7 = 10 - 7$

$x = 3$

Once the problem is solved, the solution can be verified by rewriting the problem with 3 substituted for x.

$3 + 7 = 10$

$10 = 10$

Both sides are equal, verifying that $x = 3$ is a valid solution.

Explicit verification that found an error

$x + 7 = 10$ A student rushing through her homework might mistakenly write $x = 2$ as the solution to this problem. If she takes a moment to rework the equation with her answer, she will realize the answer is incorrect.

$x + 7 = 10$

$2 + 7 = 10$

$9 = 10$

Since $9 \neq 10$, the student knows she needs to go back and find a different solution to the problem.

F. Long CoT Patterns in Pre-training Data

F.1. Snapshot of webpages

Source: brilliant.org

The following two examples demonstrate how explicit verification after answering a question can naturally exist on a webpage.

Explicit verification

$x + 7 = 10$

This problem can be solved by subtracting 7 from each side.

$x + 7 - 7 = 10 - 7$

$x = 3$

Once the problem is solved, the solution can be verified by rewriting the problem with 3 substituted for x.

$3 + 7 = 10$

$10 = 10$

Both sides are equal, verifying that $x = 3$ is a valid solution.

Explicit verification that found an error

$x + 7 = 10$ A student rushing through her homework might mistakenly write $x = 2$ as the solution to this problem. If she takes a moment to rework the equation with her answer, she will realize the answer is incorrect.

$x + 7 = 10$

$2 + 7 = 10$

$9 = 10$

Since $9 \neq 10$, the student knows she needs to go back and find a different solution to the problem.

来源: kidswholovemath.substack.com

Attempt the question from different perspective

The Double Check Game

Regardless of the scenario, we can play the double check game!

The game is simple: we try to solve the problem in as many different ways as possible.

Elementary School Example

Math problem is: $78 - 57 = ?$

To play the game, we try to solve the problem in as many different ways as possible.

The first solution:

$? = 78 - 57$

Break apart the 57:

$? = 78 - 50 - 7$

$? = 28 - 7$

$? = 21$

A second solution:

$? = 78 - 57$

Subtract an easier number from 78:

$? = 78 - 60 + 3$

$? = 18 + 3$

$? = 21$

A third solution:

$? = 78 - 57$

Subtract 57 from an easier number:

$? = 80 - 57 - 2$

$? = 23 - 2$

$? = 21$

...

Source: kidswholovemath.substack.com

Attempt the question from different perspective

The Double Check Game

Regardless of the scenario, we can play the double check game!

The game is simple: we try to solve the problem in as many different ways as possible.

Elementary School Example

Math problem is: $78 - 57 = ?$

To play the game, we try to solve the problem in as many different ways as possible.

The first solution:

$? = 78 - 57$

Break apart the 57:

$? = 78 - 50 - 7$

$? = 28 - 7$

$? = 21$

A second solution:

$? = 78 - 57$

Subtract an easier number from 78:

$? = 78 - 60 + 3$

$? = 18 + 3$

$? = 21$

A third solution:

$? = 78 - 57$

Subtract 57 from an easier number:

$? = 80 - 57 - 2$

$? = 23 - 2$

$? = 21$

...

F.2. OpenWebMath

F.2.1. Queries

我们使用GPT-4o生成了在长链思考（CoT）中常见的典型枢纽关键词的示例。这些关键词用于在OpenWebMath中查找具有长链思考轨迹特征的有趣属性的文档。

”Aha” Phrases

```
"Let's think step by step."
"Let's go through this one step at a time."
"Breaking it down step by step..."
"Thinking about it logically, first..."
"Step 1: Let's figure out the starting point."
"If we follow the steps carefully, we get..."
"To solve this, let' s analyze it piece by piece."
"Going through this systematically, we have..."
"Okay, let' s solve this gradually."
"Does that make sense?"
"Is this correct?"
"Wait, does that check out?"
"Am I missing something?"
"Hmm⋯ does that work?"
"Let me verify that."
"That makes sense, right?"
"Hold on, is this right?"
"Let' s double-check this."
"Wait, actually..."
"Oh, hold on..."
"Wait a second..."
"Actually, let me rethink that."
"Hmm, let me go back for a moment."
"I might need to check this again."
"Let's pause and reassess."
"Let' s check by doing the reverse."
"Let's verify by working backward."
"Can we check this by reversing the process?"
"To confirm, let's undo the steps."
"A good way to verify is by reversing it."
"If we undo the operations, do we get the same result?"
...
```

F.2. OpenWebMath

F.2.1. Queries

We used `GPT-4o` to generate examples of typical pivot keywords found in long CoT. These were used to find documents in OpenWebMath that have interesting properties characteristic of long CoT trajectories.

”Aha” Phrases

```
"Let’s think step by step."
"Let’s go through this one step at a time."
"Breaking it down step by step..."
"Thinking about it logically, first..."
"Step 1: Let’s figure out the starting point."
"If we follow the steps carefully, we get..."
"To solve this, let’ s analyze it piece by piece."
"Going through this systematically, we have..."
"Okay, let’ s solve this gradually."
"Does that make sense?"
"Is this correct?"
"Wait, does that check out?"
"Am I missing something?"
"Hmm⋯ does that work?"
"Let me verify that."
"That makes sense, right?"
"Hold on, is this right?"
"Let’ s double-check this."
"Wait, actually..."
"Oh, hold on..."
"Wait a second..."
"Actually, let me rethink that."
"Hmm, let me go back for a moment."
"I might need to check this again."
"Let’s pause and reassess."
"Let’ s check by doing the reverse."
"Let’s verify by working backward."
"Can we check this by reversing the process?"
"To confirm, let’s undo the steps."
"A good way to verify is by reversing it."
"If we undo the operations, do we get the same result?"
...
```


F.2.2. Matches

以下讨论发生在概率编程框架MC Stan的消息板上。用户Tiny对如何解释某些数据有疑问，多名其他用户作出了回应。我们可以看到，典型的长链思考（CoT）关键词（以**粗体**突出显示）包括分支、自我修正，甚至对方法可行性的评估。

Discussion on message board

So the question is then to find the right prediction task, looking at your setup, those may include:

...

48 For a hypothetical future serial drawn from the same “population” as the observed serials. (i.e. include the varying intercept via a new level and “sample$_{n}ew_{l}evels =$,$uncertainty$ßÆ)$For the ø true Æ or ø average Æ underlying system (i.e. ignore the varying intercept) In the experiments you actually observed$What is the expected difference in some of the constants (or anything else) between two future experiments? All of those (and more) should be

F.2.2. MATCHES

Source: MC Stan Discussion Forum

The discussion below took place on a message board for the probabilistic programming framework MC Stan. The user Tiny has a question about how to interpret some data and multiple other users are responding. We can see the usual pivot keywords (highlighted in **bold**) characteristic of long CoT, including branching, self-correction and even an assessment of the feasibility of an approach.

Discussion on message board

So the question is then to find the right prediction task, looking at your setup, those may include:

...

48 For a hypothetical future serial drawn from the same “population” as the observed serials. (i.e. include the varying intercept via a new level and “sample$_n ew_l evels =$,$uncertainty$ßÆ)$For the øtrue Æor øaverage Æunderlying system (i.e. ignore the varying intercept) In the experiments you actually observe
$What is the expected difference in some of the constants (or anything else) between two future experiments? All of those (and more) should be

源: physicsforums.com

以下讨论发生在物理论坛上。用户Songoku请求作业帮助，而用户BvU尝试在不直接透露答案的情况下提供帮助。我们看到了一些常见的关键词，表明自我反思、表达不确定性和假设的形成。

Source: physicsforums.com

The discussion below took place on a physics forum. The user Songoku is asking for help with homework and another user BvU is trying to assist without revealing the solution directly. We see the usual pivot keywords indicating self-reflection, expression of uncertainty and formulation of hypotheses.

Discussion on a physics forum

Cylinder in 3 D
1. Dec 13, 2017

songoku
1. The problem statement, all variables and given/known data
Let r be a positive constant. Consider the cylinder x2 + y2 <= r2, and let C be the part of the cylinder that satisfies
0 <= z <= y.
(1) Consider the cross section of C by the plane x = t (-r <= t <= r), and express its area in terms of r, t.
(2) Calculate the volume of C, and express it in terms of r.

...

5. Dec 13, 2017
BvU
Simple case: x = 0. So -1 <= y <= 1. In the yz plane 0 <= z <= y is a triangle.
What about y ?

6. Dec 13, 2017
songoku
I think I am missing something here because I feel I can't really grasp the hint given.
Let me start from the basic again:
1. Let the x - axis horizontal, y - axis vertical and z - axis in / out of page. I imagine there is circle on xy plane with radius r then it extends out of page (I take out of page as z+) to form 3 D cylinder. **Is this correct?**
2. Plane x = t is like the shape of a piece of paper hold vertically with the face of paper facing x - axis (I mean x - axis is the normal of the plane). **Is this correct?**
Thanks

7. Dec 14, 2017
BvU
Yes

8. Dec 14, 2017
songoku

"Consider the cross section of C by plane x = t" means plane x = t cuts the cylinder ?
And the intersection will be rectangle?
...

Discussion on a physics forum

Cylinder in 3 D
1. Dec 13, 2017

songoku
1. The problem statement, all variables and given/known data
Let r be a positive constant. Consider the cylinder x2 + y2 <= r2, and let C be the part of the cylinder that satisfies
0 <= z <= y.
(1) Consider the cross section of C by the plane x = t (-r <= t <= r), and express its area in terms of r, t.
(2) Calculate the volume of C, and express it in terms of r.

...

5. Dec 13, 2017
BvU
Simple case: x = 0. So -1 <= y <= 1. In the yz plane 0 <= z <= y is a triangle.
What about y ?

6. Dec 13, 2017
songoku
I think I am missing something here because I feel I can't really grasp the hint given.
Let me start from the basic again:
1. Let the x - axis horizontal, y - axis vertical and z - axis in / out of page. I imagine there is circle on xy plane with radius r then it extends out of page (I take out of page as z+) to form 3 D cylinder. **Is this correct?**
2. Plane x = t is like the shape of a piece of paper hold vertically with the face of paper facing x - axis (I mean x - axis is the normal of the plane). **Is this correct?**
Thanks

7. Dec 14, 2017
BvU
Yes

8. Dec 14, 2017
songoku

"Consider the cross section of C by plane x = t" means plane x = t cuts the cylinder ?
And the intersection will be rectangle?
...

用户Baymax正在求助于一个概率问题，我们看到他与另一位用户Lulu的对话。我们注意到他们之间的快速交流类似于长链思考中灵活的分支行为，其中多个解决方案被迅速评估和考虑。我们还看到了一种顿悟的表达，这可以很容易地在长链思考中重新表述为自我验证。

Discussion on Stack Exchange

probability that we stop flipping after exactly ten flips in a biased coin flipping?

...

I thought that let us fix of getting a third head at last that is at 10th flip, so that we would stop there, and the remaining - getting two heads can be accommodated in the 9 trials. so there are $$9$$ choose 2 ways of getting two heads so the probability that we stop flipping after exactly ten flips is $$^9C_{2}$$. $$\frac{1}{4}^3$$.$$\frac{3}{4}^7$$. **Is this correct?**

EDIT - Now the probability of getting exactly 3 heads? I got it to be $$^{10} C_{3} \frac{1}{4}^3 \frac{3}{4}^7$$. Should we get the same as the previous one? any reason why they should/should not be same?

48 I think you switched $P(H), P(T)$ |**but the approach is good.**| { lulu Oct 1 '18 at 16:13• |**oh i see now!**| thanks! { BAYMAX Oct 1 '18 at 16:13• @lulu please see the edit { BAYMAX Oct 1 '18 at 16:30• Your probability for exactly 3 heads is right as well. It should be obvious why the results have to be different. In the first case the outcome of the last flip is fix and in the second case the outcome of the last flip is not fix. { callculus Oct 1 '18 at 16:31...

Source: StackExchange

The user Baymax is asking for help on a probability problem and we see dialogue with another user Lulu. We see that the quick back-and-forth between them is similar to the kind of nimble branching behavior in long CoT where multiple solutions are quickly assessed and considered. We also see an expression of realization which can be easily re-cast as self-verification in a long CoT.

Discussion on Stack Exchange

probability that we stop flipping after exactly ten flips in a biased coin flipping?

...

I thought that let us fix of getting a third head at last that is at 10th flip, so that we would stop there, and the remaining - getting two heads can be accommodated in the 9 trials. so there are $$9$$ choose 2 ways of getting two heads so the probability that we stop flipping after exactly ten flips is $$^9C_{2}$$. $$\frac{1}{4}^3$$.$$\frac{3}{4}^7$$. **Is this correct?**

EDIT - Now the probability of getting exactly 3 heads? I got it to be $$^{10} C_{3} \frac{1}{4}^3 \frac{3}{4}^7$$. Should we get the same as the previous one? any reason why they should/should not be same?

48 I think you switched $P(H), P(T)$ |**but the approach is good.**| { lulu Oct 1 '18 at 16:13• |**oh i see now!**| thanks! { BAYMAX Oct 1 '18 at 16:13• @lulu please see the edit { BAYMAX Oct 1 '18 at 16:30• Your probability for exactly 3 heads is right as well. It should be obvious why the results have to be different. In the first case the outcome of the last flip is fix and in the second case the outcome of the last flip is not fix. { callculus Oct 1 '18 at 16:31...

StackExchange

User88 与多位其他用户互动。请注意，他们相互帮助澄清疑问，这让人联想到长链自我修正轨迹中的自我纠正。

Discussion on Stack Exchange

Choosing units for drug testing

Here's a third puzzle that I found in a book, slightly paraphrased because I don't entirely remember the format of the original.

...

How can he arrange the dosage amounts so that he ends up using all 25 test packages, and the total units of dosage used in the tests are as low as possible?

The book had the answer, but one, it didn't explain how the answer was arrived at, and two, I don't remember what the answer was and no longer have that book with me.

48 |**Am I missing something**|, or is the goal just to find 25 coprime numbers from 25 to 50? { Aza May 20 '14 at 4:33• They don't have to be coprime. There just can't be any two where one is a factor of the other. And the range is from 1 to 50, not 25 to 50. { Joe Z. May 20 '14 at 4:34• |**Wouldn't a single test**| of 1 unit technically satisfy the requirement? Or |**am I missing something? Ah, I guess you have to**| perform exactly 25 tests. { arshajii May 20 '14 at 14:28• |**Yea.**| Wouldn't 1 win? { awesomepi May 20 '14 at 19:24• You have to use all 25 tests. { Joe Z. May 20 '14 at 19:31By logically starting from 26-50 and trying to shrink them one by one you can easily show:
$8, 12, 14, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 29, 30, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49$Which equals 711...

Source: StackExchange

User88 interacts with multiple other users. Observe that they are helping to clarify each others' doubts, which is reminiscent of self-correction in long CoT trajectories.

Discussion on Stack Exchange

Choosing units for drug testing

Here's a third puzzle that I found in a book, slightly paraphrased because I don't entirely remember the format of the original.

...

How can he arrange the dosage amounts so that he ends up using all 25 test packages, and the total units of dosage used in the tests are as low as possible?

The book had the answer, but one, it didn't explain how the answer was arrived at, and two, I don't remember what the answer was and no longer have that book with me.

48 |**Am I missing something**|, or is the goal just to find 25 coprime numbers from 25 to 50? { Aza May 20 '14 at 4:33• They don't have to be coprime. There just can't be any two where one is a factor of the other. And the range is from 1 to 50, not 25 to 50. { Joe Z. May 20 '14 at 4:34• |**Wouldn't a single test**| of 1 unit technically satisfy the requirement? Or |**am I missing something? Ah, I guess you have to**| perform exactly 25 tests. { arshajii May 20 '14 at 14:28• |**Yea.**| Wouldn't 1 win? { awesomepi May 20 '14 at 19:24• You have to use all 25 tests. { Joe Z. May 20 '14 at 19:31By logically starting from 26-50 and trying to shrink them one by one you can easily show:
$8, 12, 14, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 29, 30, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49$Which equals 711...