DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

immediate

Abstract

Mathematical reasoning poses a significant challenge for language models due to its complex and structured nature. In this paper, we introduce DeepSeekMath 7B, which continues pre-training DeepSeek-Coder-Base-v1.5 7B with 120B math-related tokens sourced from Common Crawl, together with natural language and code data. DeepSeekMath 7B has achieved an impressive score of 51.7% on the competition-level MATH benchmark without relying on external toolkits and voting techniques, approaching the performance level of Gemini-Ultra and GPT-4. Self-consistency over 64 samples from DeepSeekMath 7B achieves 60.9% on MATH. The mathematical reasoning capability of DeepSeekMath is attributed to two key factors: First, we harness the significant potential of publicly available web data through a meticulously engineered data selection pipeline. Second, we introduce Group Relative Policy Optimization (GRPO), a variant of Proximal Policy Optimization (PPO), that enhances mathematical reasoning abilities while concurrently optimizing the memory usage of PPO.

数学推理因其复杂性和结构性而对语言模型提出了重大挑战。在 MATH 基准测试中，最好的开源模型刚刚达到 30%+ 的 top-1 准确率，而大公司的封闭模型已经超过了 40% 的里程碑。在本文中，我们引入了 DeepSeekMath 7B，它继续使用来自 Common Crawl 的 120B 个数学相关 token 以及自然语言和代码数据对 DeepSeek-Coder-Base-v1.5 7B 进行预训练。DeepSeekMath 7B 在不依赖外部工具包和投票技术的情况下，在竞赛级 MATH 基准测试中取得了 51.7% 的令人印象深刻的分数，接近 Gemini-Ultra 和 GPT-4 的性能水平。来自 DeepSeekMath 7B 的 64 个样本的自一致性在 MATH 上达到 60.9%。DeepSeekMath 的数学推理能力归因于两个关键因素：首先，我们通过精心设计的数据选择管道，充分利用了公开可用的网络数据的巨大潜力。其次，我们引入了组相对策略优化 (GRPO)，这是近端策略优化 (PPO) 的一种变体，它可以增强数学推理能力，同时优化 PPO 的内存使用情况。

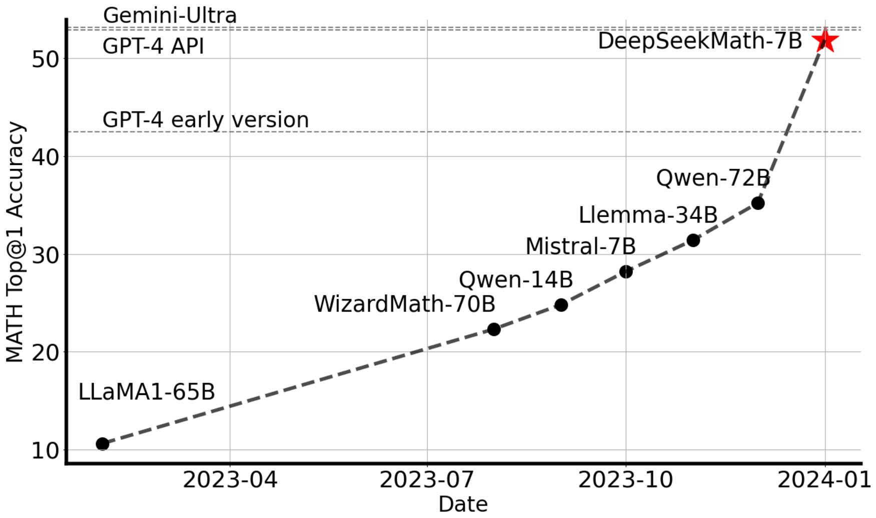

Figure 1 | Top1 accuracy of open-source models on the competition-level MATH benchmark (17) without the use of external toolkits and voting techniques.
在不使用外部工具包和投票技术的情况下，竞赛级 MATH 基准 (17) 上开源模型的 Top@1

1. Introduction

Large language models (LLM) have revolutionized the approach to mathematical reasoning in artificial intelligence, spurring significant advancements in both the quantitative reasoning benchmark (17) and the geometry reasoning benchmark (46). Moreover, these models have proven instrumental in assisting humans in solving complex mathematical problems (44). However, cutting-edge models such as GPT-4 (32) and Gemini-Ultra (1) are not publicly available, and the currently accessible open-source models considerably trail behind in performance.
大型语言模型 (LLM) 彻底改变了人工智能中的数学推理方法，推动了定量推理基准 (17) 和几何推理基准 (46) 的重大进步。此外，这些模型已被证明有助于帮助人类解决复杂的数学问题 (44)。然而，GPT-4 (32) 和 Gemini-Ultra (1) 等尖端模型尚未公开，目前可访问的开源模型在性能上远远落后。

In this study, we introduce DeepSeekMath, a domain-specific language model that significantly outperforms the mathematical capabilities of open-source models and approaches the performance level of GPT-4 on academic benchmarks. To achieve this, we create the DeepSeekMath Corpus, a large-scale high-quality pre-training corpus comprising 120B math tokens. This dataset is extracted from the Common Crawl (CC) using a fastText-based classifier (22). In the initial iteration, the classifier is trained using instances from OpenWebMath (34) as positive examples, while incorporating a diverse selection of other web pages to serve as negative examples. Subsequently, we employ the classifier to mine additional positive instances from the CC, which are further refined through human annotation. The classifier is then updated with this enhanced dataset to improve its performance. The evaluation results indicate that the large-scale corpus is of high quality, as our base model DeepSeekMath-Base 7B achieves 64.2% on GSM8K (9) and 36.2% on the competition-level MATH dataset (17), outperforming Minerva 540B (25). In addition, the DeepSeekMath Corpus is multilingual, so we notice an improvement in Chinese mathematical benchmarks (51; 61). We believe that our experience in mathematical data processing is a starting point for the research community, and there is significant room for improvement in the future.
在本研究中，我们引入了 DeepSeekMath，这是一个领域特定语言模型，其数学能力显著优于开源模型，在学术基准上接近 GPT-4 的性能水平。为了实现这一目标，我们创建了 DeepSeekMath 语料库，这是一个包含 1200 亿个数学 token 的大规模高质量预训练语料库。此数据集是使用基于 fastText 的分类器 (22) 从 Common Crawl (CC) 中提取的。在初始迭代中，使用来自 OpenWebMath (34) 的实例作为正例对分类器进行训练，同时结合各种其他网页作为负例。随后，我们使用分类器从 CC 中挖掘更多正例，并通过人工注释进一步细化。然后使用这个增强的数据集更新分类器以提高其性能。评估结果表明，大规模语料库质量较高，我们的基础模型 DeepSeekMath-Base 7B 在 GSM8K (9) 上达到 64.2%，在竞赛级 MATH 数据集 (17) 上达到 36.2%，优于 Minerva 540B (25)。此外，DeepSeekMath 语料库是多语言的，因此我们注意到中文数学基准 (51; 61) 有所改进。我们相信，我们在数学数据处理方面的经验是研究界的起点，未来还有很大的改进空间。

DeepSeekMath-Base is initialized with DeepSeek-Coder-Base-v1.5 7B (15), as we notice that starting from a code training model is a better choice compared to a general LLM. Furthermore, we observe the math training also improves model capability on MMLU (16) and BBH benchmarks (43), indicating it does not only enhance the model's mathematical abilities but also amplifies general reasoning capabilities.
DeepSeekMath-Base 使用 DeepSeek-Coder-Base-v1.5 7B (15) 初始化，因为我们注意到从代码训练模型开始比从一般的 LLM 开始是更好的选择。此外，我们观察到数学训练也提高了 MMLU (16) 和 BBH 基准 (43) 上的模型能力，这表明它不仅增强了模型的数学能力，而且还增强了一般推理能力。

After pre-training, we apply mathematical instruction tuning to DeepSeekMath-Base with chain-of-thought (50), program-of-thought (8; 13), and tool-integrated reasoning (14) data. The resulting model DeepSeekMath-Instruct 7B beats all 7B counterparts and is comparable with 70B open-source instruction-tuned models.
经过预训练后，我们使用思路链 (50)、思路程序 (8; 13) 和工具集成推理 (14) 数据对 DeepSeekMath-Base 进行数学指令调整。生成的模型 DeepSeekMath-Instruct 7B 击败了所有 7B 同类模型，可与 70B 开源指令调整模型相媲美。

Furthermore, we introduce the Group Relative Policy Optimization (GRPO), a variant reinforcement learning (RL) algorithm of Proximal Policy Optimization (PPO) (40). GRPO foregoes the critic model, instead estimating the baseline from group scores, significantly reducing training resources. By solely using a subset of English instruction tuning data, GRPO obtains a substantial improvement over the strong DeepSeekMath-Instruct, including both in-domain (GSM8K: 82.9% → 88.2%, MATH: 46.8% → 51.7%) and out-of-domain mathematical tasks (e.g., CMATH: 84.6% → 88.8%) during the reinforcement learning phase. We also provide a unified paradigm to understand different methods, such as Rejection Sampling Fine-Tuning (RFT) (57), Direct Preference Optimization (DPO) (37), PPO and GRPO. Based on such a unified paradigm, we find that all these methods are conceptualized as either direct or simplified RL techniques. We also conduct extensive experiments, e.g., online v.s. offline training, outcome v.s. process supervision, single-turn v.s. iterative RL and so on, to deeply investigate the essential elements of this paradigm. At last, we explain why our RL boosts the performance of instruction-tuned models, and further summarize potential directions to achieve more effective RL based on this unified paradigm.
此外，我们引入了组相对策略优化 (GRPO)，这是近端策略优化 (PPO) (40) 的变体强化学习 (RL) 算法。GRPO 放弃了批评模型，而是从组分数估计基线，从而大大减少了训练资源。通过仅使用英语教学调整数据的子集，GRPO 获得了比强大的 DeepSeekMath-Instruct 显着的改进，包括强化学习阶段的域内（GSM8K：82.9% → 88.2%，MATH：46.8% → 51.7%）和域外数学任务（例如，CMATH：84.6% → 88.8%）。我们还提供了一个统一的范式来理解不同的方法，例如拒绝采样微调 (RFT) (57)、直接偏好优化 (DPO) (37)、PPO 和 GRPO。基于这种统一的范式，我们发现所有这些方法都被概念化为直接或简化的 RL 技术。我们还进行了广泛的实验，例如在线与离线训练、结果与过程监督、单轮与迭代 RL 等，以深入研究该范式的基本要素。最后，我们解释了为什么我们的 RL 可以提高指令调整模型的性能，并进一步总结了基于这种统一范式实现更有效的 RL 的潜在方向。

1.1. Contributions

Our contribution includes scalable math pre-training, along with the exploration and analysis of reinforcement learning.
我们的贡献包括可扩展的数学预训练，以及强化学习的探索和分析。

Math Pre-Training at Scale
大规模数学预训练

- Our research provides compelling evidence that the publicly accessible Common Crawl data contains valuable information for mathematical purposes. By implementing a meticulously designed data selection pipeline, we successfully construct the DeepSeekMath Corpus, a high-quality dataset of 120B tokens from web pages filtered for mathematical content, which is almost 7 times the size of the math web pages used by Minerva (25) and 9 times the size of the recently released OpenWebMath (34).
  我们的研究提供了令人信服的证据，表明可公开访问的 Common Crawl 数据包含对数学目的有价值的信息。通过实施精心设计的数据选择管道，我们成功构建了 DeepSeekMath Corpus，这是一个高质量的数据集，包含 1200 亿个 token，这些 token 来自经过数学内容过滤的网页，其大小几乎是 Minerva (25) 使用的数学网页的 7 倍，是最近发布的 OpenWebMath (34) 的 9 倍。
- Our pre-trained base model DeepSeekMath-Base 7B achieves comparable performance with Minerva 540B (25), indicating the number of parameters is not the only key factor in mathematical reasoning capability. A smaller model pre-trained on high-quality data could achieve strong performance as well.
  我们的预训练基础模型 DeepSeekMath-Base 7B 实现了与 Minerva 540B (25) 相当的性能，表明参数数量并不是数学推理能力的唯一关键因素。使用高质量数据进行预训练的较小模型也可以实现出色的性能。

- We share our findings from math training experiments. Code training prior to math training improves models' ability to solve mathematical problems both with and without tool use. This offers a partial answer to the long-standing question: does code training improve reasoning abilities? We believe it does, at least for mathematical reasoning.
  我们分享了数学训练实验的发现。数学训练之前的代码训练提高了模型在使用和不使用工具的情况下解决数学问题的能力。这为长期存在的问题提供了部分答案：代码训练能提高推理能力吗？我们相信它能，至少对于数学推理而言。
- Although training on arXiv papers is common, especially in many math-related papers, it brings no notable improvements on all mathematical benchmarks adopted in this paper.
  尽管在 arXiv 论文上进行训练很常见，尤其是在许多与数学相关的论文中，但它并没有为本文采用的所有数学基准带来显著的改进。

Exploration and Analysis of Reinforcement Learning
强化学习的探索与分析

- We introduce Group Relative Policy Optimization (GRPO), an efficient and effective reinforcement learning algorithm. GRPO foregoes the critic model, instead estimating the baseline from group scores, significantly reducing training resources compared to Proximal Policy Optimization (PPO).
  我们引入了组相对策略优化 (GRPO)，这是一种高效且有效的强化学习算法。GRPO 放弃了批评模型，而是根据组分数估计基线，与近端策略优化 (PPO) 相比，显著减少了训练资源。
- We demonstrate that GRPO significantly enhances the performance of our instruction-tuned model DeepSeekMath-Instruct, by solely using the instruction-tuning data. Furthermore, we observe enhancements in the out-of-domain performance during the reinforcement learning process.
  我们证明，仅使用指令调整数据，GRPO 就能显著提高我们指令调整模型 DeepSeekMath-Instruct 的性能。此外，我们在强化学习过程中观察到域外性能的增强。
- We provide a unified paradigm to understand different methods, such as RFT, DPO, PPO, and GRPO. We also conduct extensive experiments, e.g., online v.s. offline training, outcome v.s. process supervision, single-turn v.s. iterative reinforcement learning, and so on to deeply investigate the essential elements of this paradigm.
  我们提供了一个统一范式来理解不同的方法，例如 RFT、DPO、PPO 和 GRPO。我们还进行了广泛的实验，例如在线与离线训练、结果与过程监督、单轮与迭代强化学习等，以深入研究该范式的基本要素。
- Based on our unified paradigm, we explore the reasons behind the effectiveness of reinforcement learning, and summarize several potential directions to achieve more effective reinforcement learning of LLMs.
  基于我们的统一范式，我们探索 RL 有效性的原因，并总结出实现更有效的 LLM - RL 的几个潜在方向。

1.2. Summary of Evaluations and Metrics **评估和指标摘要**

- English and Chinese Mathematical Reasoning: We conduct comprehensive assessments of our models on English and Chinese benchmarks, covering mathematical problems from grade-school level to college level. English benchmarks include GSM8K (9), MATH (17), SAT (3), OCW Courses (25), MMLU-STEM (16). Chinese benchmarks include MGSM-zh (41), CMATH (51), Gaokao-MathCloze (61), and Gaokao-MathQA (61). We evaluate models' ability to generate self-contained text solutions without tool use, and also the ability to solve problems using Python.
  英语和中文数学推理：我们根据英语和中文基准对我们的模型进行全面评估，涵盖从小学水平到大学水平的数学问题。英语基准包括 GSM8K (9)、MATH (17)、SAT (3)、OCW 课程 (25)、MMLU-STEM (16)。中文基准包括 MGSM-zh (41)、CMATH (51)、Gaokao-MathCloze (61) 和 Gaokao-MathQA (61)。我们评估模型在不使用工具的情况下生成自包含文本解决方案的能力，以及使用 Python 解决问题的能力。
  On English benchmarks, DeepSeekMath-Base is competitive with the closed-source Minerva 540B (25), and surpasses all open-source base models (e.g., Mistral 7B (21) and Llemma-34B (3)), regardless of whether they've undergone math pre-training or not, often by a significant margin. Notably, DeepSeekMath-Base is superior on Chinese benchmarks, likely because we don't follow previous works (3; 25) to collect English-only math pre-training data, and also include high-quality non-English ones. With mathematical instruction tuning and reinforcement learning, the resulting DeepSeekMath-Instruct and DeepSeekMath-RL demonstrate strong performance, obtaining an accuracy of over 50% on the competition-level MATH dataset for the first time within the open-source community.
  在英语基准测试中，DeepSeekMath-Base 与闭源的 Minerva 540B (25) 相媲美，并且超越了所有开源基础模型（例如 Mistral 7B (21) 和 Llemma-34B (3)），无论它们是否经过数学预训练，通常都领先显著。值得注意的是，DeepSeekMath-Base 在中文基准测试中更胜一筹，这可能是因为我们没有像之前的作品 (3; 25) 那样收集纯英语的数学预训练数据，而是包含了高质量的非英语数据。通过数学指令调整和强化学习，由此产生的 DeepSeekMath-Instruct 和 DeepSeekMath-RL 表现出色，在开源社区中首次在竞赛级 MATH 数据集上获得了超过 50% 的准确率。

- Formal Mathematics: We evaluate DeepSeekMath-Base using the informal-to-formal theorem proving task from (20) on miniF2F (60) with Isabelle (52) chosen to be the proof assistant. DeepSeekMath-Base demonstrates strong few-shot autoformalization performance.
  形式数学: 我们使用 miniF2F (60) 上 (20) 的非形式化到形式化定理证明任务评估 DeepSeekMath-Base，并选择 Isabelle (52) 作为证明助手。
- Natural Language Understanding, Reasoning, and Code: To build a comprehensive profile of models' general understanding, reasoning, and coding capabilities, we evaluate DeepSeekMath-Base on the Massive Multitask Language Understanding (MMLU) benchmark (16) which encompasses 57 multiple-choice tasks covering diverse subjects, BIG-Bench Hard (BBH) (43) which consists of 23 challenging tasks that mostly require multi-step reasoning to solve, as well as HumanEval (7) and MBPP (2) which are widely used to evaluate code language models. Math pre-training benefits both language understanding and reasoning performance.
  我们使用 miniF2F (60) 上来自 (20) 的非形式化到形式化定理证明任务评估 DeepSeekMath-Base，并选择 Isabelle (52) 作为证明助手。DeepSeekMath-Base 展示了强大的少量自动形式化性能。
- **自然语言理解、推理和编码**：为了全面了解模型的一般理解、推理和编码能力，我们根据大规模多任务语言理解 (MMLU) 基准 (16)（包含 57 个涵盖不同主题的多项选择任务）、BIG-Bench Hard (BBH) (43)（包含 23 个具有挑战性的任务，大多需要多步推理才能解决）以及广泛用于评估代码语言模型的 HumanEval (7) 和 MBPP (2) 对 DeepSeekMath-Base 进行了评估。数学预训练有利于提高语言理解和推理性能。

2. Math Pre-Training

2.1. Data Collection and Decontamination

In this section, we will outline the process of constructing the DeepSeekMath Corpus from Common Crawl. As depicted in Figure 2, we present an iterative pipeline that demonstrates how to systematically gather a large-scale mathematical corpus from Common Crawl, starting with a seed corpus (e.g., a small but high-quality collection of math-related dataset). It's worth noting that this approach is also applicable to other domains, such as coding.
在本节中，我们将概述从 Common Crawl 构建 DeepSeekMath 语料库的过程。如图 2 所示，我们展示了一个迭代流程，演示了如何从 Common Crawl 系统地收集大规模数学语料库，从种子语料库（例如，一组规模较小但质量较高的数学相关数据集）开始。值得注意的是，这种方法也适用于其他领域，例如编码。

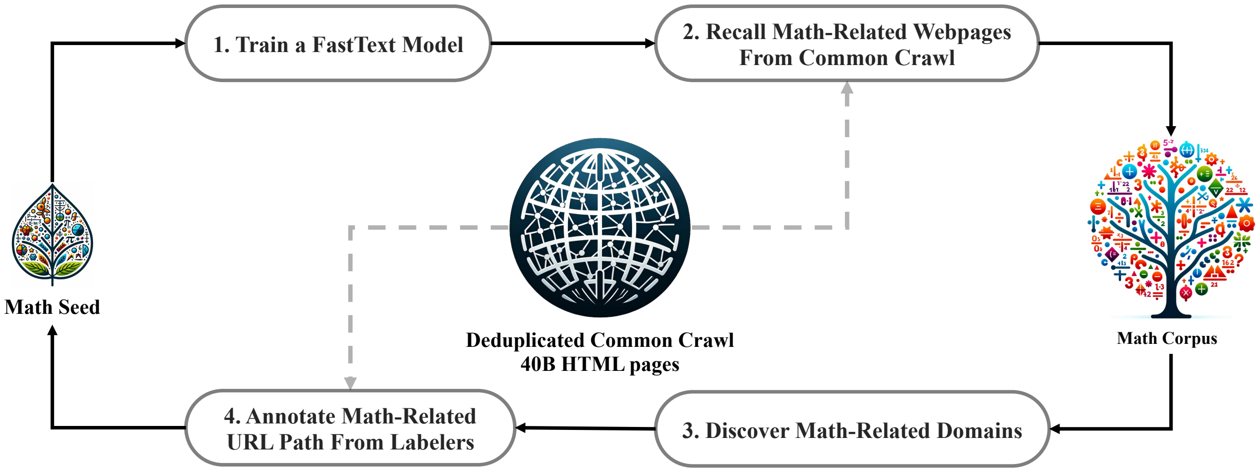
1. Train a FastText Model
2. Recall Math-Related Webpages From Common Crawl
Math Seed
Deduplicated Common Crawl 40B HTML pages
Math Corpus
4. Annotate Math-Related URL Path From Labelers
3. Discover Math-Related Domains

Figure 2 | An iterative pipeline that collects mathematical web pages from Common Crawl.

First, we choose OpenWebMath (34), a collection of high-quality mathematical web texts, as our initial seed corpus. Using this corpus, we train a fastText model (22) to recall more OpenWebMath-like mathematical web pages. Specifically, we randomly select 500,000 data points from the seed corpus as positive training examples and another 500,000 web pages from Common Crawl as negative ones. We employ an open-source library[1] for training, configuring the vector dimension to 256, learning rate to 0.1, the maximum length of word n-gram to 3, the minimum number of word occurrences to 3, and the number of training epochs to 3. To reduce the size of the original Common Crawl, we employ URL-based deduplication and near-deduplication techniques, resulting in 40B HTML web pages. We then recall mathematical web pages from deduplicated Common Crawl with the fastText model. To filter out low-quality mathematical content, we rank the collected pages according to their scores predicted by the fastText model, and only preserve the top-ranking ones. The volume of data preserved is assessed through pre-training experiments on the top 40B, 80B, 120B, and 160B tokens. In the first iteration, we choose to keep the top 40B tokens.
首先，我们选择 OpenWebMath (34)（一个高质量的数学网络文本集合）作为我们的初始种子语料库。利用这个语料库，我们训练了一个 fastText 模型 (22)，以召回更多类似 OpenWebMath 的数学网页。具体来说，我们从种子语料库中随机选择 500,000 个数据点作为正训练示例，从 Common Crawl 中随机选择另外 500,000 个网页作为负训练示例。我们使用一个开源库 [1] 进行训练，将向量维度配置为 256，学习率为 0.1，单词 n-gram 的最大长度为 3，单词出现的最小次数为 3，训练周期数为 3。为了减小原始 Common Crawl 的大小，我们采用了基于 URL 的重复数据删除和近似重复数据删除技术，最终生成了 40B 的 HTML 网页。然后，我们使用 fastText 模型从去重后的 Common Crawl 中调用数学网页。为了过滤掉低质量的数学内容，我们根据 fastText 模型预测的分数对收集到的页面进行排名，只保留排名靠前的页面。通过对前 40B、80B、120B 和 160B 个 token 进行预训练实验来评估保留的数据量。在第一次迭代中，我们选择保留前 40B 个 token。

After the first iteration of data collection, numerous mathematical web pages remain uncollected, mainly because the fastText model is trained on a set of positive examples that lacks sufficient diversity. We therefore identify additional mathematical web sources to enrich the seed corpus, so that we can optimize the fastText model. Specifically, we first organize the entire Common Crawl into disjoint domains; a domain is defined as web pages sharing the same base URL. For each domain, we calculate the percentage of web pages that are collected in the first iteration. Domains where over 10% of the web pages have been collected are classified as math-related (e.g., mathoverflow.net). Subsequently, we manually annotate the URLs associated with mathematical content within these identified domains (e.g., mathoverflow.net/questions). Web pages linked to these URLs, yet uncollected, will be added to the seed corpus. This approach enables us to gather more positive examples, thereby training an improved fastText model capable of recalling more mathematical data in the subsequent iteration. After four iterations of data collection, we end up with 35.5M mathematical web pages, totaling 120B tokens. In the fourth iteration, we notice that nearly 98% of the data has already been collected in the third iteration, so we decide to cease data collection.
在第一次数据收集迭代之后，仍有许多数学网页未被收集，这主要是因为 fastText 模型是在一组缺乏足够多样性的正例上训练的。因此，我们确定了其他数学网络资源来丰富种子语料库，以便我们可以优化 fastText 模型。具体来说，我们首先将整个 Common Crawl 组织成不相交的域；域被定义为共享相同基本 URL 的网页。对于每个域，我们计算在第一次迭代中收集的网页百分比。收集了超过 10% 网页的域被归类为与数学相关的域（例如，mathoverflow.net）。随后，我们手动注释这些已识别域中与数学内容相关的 URL（例如，mathoverflow.net/questions）。链接到这些 URL 的尚未收集的网页将被添加到种子语料库中。这种方法使我们能够收集更多正面示例，从而训练出改进的 fastText 模型，使其能够在后续迭代中调用更多数学数据。经过四次数据收集迭代后，我们最终得到了 3550 万个数学网页，总计 1200 亿个 token。在第四次迭代中，我们注意到在第三次迭代中已经收集了近 98

To avoid benchmark contamination, we follow (15) to filter out web pages containing questions or answers from English mathematical benchmarks such as GSM8K (9) and MATH (17) and Chinese benchmarks such as CMATH (51) and AGIEval (61). The filtering criteria are as follows: any text segment containing a 10-gram string that matches exactly with any sub-string from the evaluation benchmarks is removed from our math training corpus. For benchmark texts that are shorter than 10 grams but have at least 3 grams, we employ exact matching to filter out contaminated web pages.
为了避免基准污染，我们遵循 (15) 来过滤掉包含英语数学基准（例如 GSM8K (9) 和 MATH (17)）和中文基准（例如 CMATH (51) 和 AGIEval (61)）问题或答案的网页。过滤标准如下：任何包含与评估基准中的任何子字符串完全匹配的 10 元语法字符串的文本段都将从我们的数学训练语料库中删除。对于短于 10 元语法但至少有 3 元语法的基准文本，我们采用精确匹配来过滤掉受污染的网页。

[1] https://fasttext.cc

2.2. Validating the Quality of the DeepSeekMath Corpus

We run pre-training experiments to investigate how the DeepSeekMath Corpus is compared with the recently released math-training corpora
我们进行了预训练实验来研究 DeepSeekMath Corpus 与最近发布的数学训练语料库相比如何：

- MathPile (49): a multi-source corpus (8.9B tokens) aggregated from textbooks, Wikipedia, ProofWiki, CommonCrawl, StackExchange, and arXiv, with the majority (over 85%) sourced from arXiv;
一个多源语料库（89 亿个词条），汇集了教科书、维基百科、ProofWiki、CommonCrawl、StackExchange 和 arXiv，其中大部分（超过 85%）来自 arXiv；
- OpenWebMath (34): CommonCrawl data filtered for mathematical content, totaling 13.6B tokens;
CommonCrawl 数据经过过滤，包含数学内容，总计 136 亿个 token；
- Proof-Pile-2 (3): a mathematical corpus consisting of OpenWebMath, AlgebraicStack (10.3B tokens of mathematical code), and arXiv papers (28.0B tokens). When experimenting on Proof-Pile-2, we follow (3) to use an arXiv:Web:Code ratio of 2:4:1.
一个由 OpenWebMath、AlgebraicStack（10.3B 数学代码 token）和 arXiv 论文（28.0B token）组成的数学语料库。在对 Proof-Pile-2 进行实验时，我们遵循 (3) 使用 arXiv:Web:Code 比率 2:4:1。

2.2.1. Training Setting

We apply math training to a general pre-trained language model with 1.3B parameters, which shares the same framework as the DeepSeek LLMs (11), denoted as DeepSeek-LLM 1.3B. We separately train a model on each mathematical corpus for 150B tokens. All experiments are conducted using the efficient and light-weight HAI-LLM (18) training framework. Following the training practice of DeepSeek LLMs, we use the AdamW optimizer (28) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight_decay = 0.1, along with a multi-step learning rate schedule where the learning rate reaches the peak after 2,000 warmup steps, decreases to its 31.6% after 80% of the training process, and further decreases to 10.0% of the peak after 90% of the training process. We set the maximum value of learning rate to 5.3e-4, and use a batch size of 4M tokens with a 4K context length.
我们将数学训练应用于具有 1.3B 参数的通用预训练语言模型，该模型与 DeepSeek LLM (11) 共享相同的框架，称为 DeepSeek-LLM 1.3B。我们在每个数学语料库上分别训练一个模型，包含 150B 个 token。所有实验均使用高效、轻量级的 HAI-LLM (18) 训练框架进行。遵循 DeepSeek LLM 的训练实践，我们使用 AdamW 优化器 (28)，其中 $\beta_1 = 0.9$、$\beta_2 = 0.95$ 和 weight_decay = 0.1，以及多步学习率计划，其中学习率在 2,000 个预热步骤后达到峰值，在训练过程的 80% 后降至 31.6%，并在训练过程的 90% 后进一步降至峰值的 10.0%。我们将学习率的最大值设置为 5.3e-4，并使用 4M 个 token 的批处理大小和 4K 上下文长度。

Math Corpus	Size	English Benchmarks					Chinese Benchmarks		
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
No Math Training	N/A	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
MathPile	8.9B	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
OpenWebMath	13.6B	11.5%	8.9%	3.7%	31.3%	29.6%	16.8%	0.0%	14.2%
Proof-Pile-2	51.9B	14.3%	11.2%	3.7%	43.8%	29.2%	19.9%	5.1%	11.7%
DeepSeekMath Corpus	120.2B	23.8%	13.6%	4.8%	56.3%	33.1%	41.5%	5.9%	23.6%

Table 1 | Performance of DeepSeek-LLM 1.3B trained on different mathematical corpora, evaluated using few-shot chain-of-thought prompting. Corpus sizes are calculated using our tokenizer with a vocab size of 100K.

2.2.2. Evaluation Results

The DeepSeekMath Corpus is of high quality, covers multilingual mathematical content, and is the largest in size. DeepSeekMath **语料库质量很高，涵盖多语言数学内容，并且规模最大。**

- High-quality: We evaluate downstream performance on 8 mathematical benchmarks using few-shot chain-of-thought prompting (50). As shown in Table 1, there is a clear performance lead of the model trained on the DeepSeekMath Corpus. Figure 3 shows that the model trained on the DeepSeekMath Corpus demonstrates better performance than Proof-Pile-2 at 50B tokens (1 full epoch of Proof-Pile-2), indicating the average quality of DeepSeekMath Corpus is higher.
我们使用少样本思维链提示 (50) 评估了 8 个数学基准上的下游性能。如表 1 所示，在 DeepSeekMath Corpus 上训练的模型具有明显的性能领先优势。图 3 表明，在 DeepSeekMath Corpus 上训练的模型在 50B 个 token（Proof-Pile-2 的 1 个完整 epoch）时表现出比 Proof-Pile-2 更好的性能，表明 DeepSeekMath Corpus 的平均质量更高。

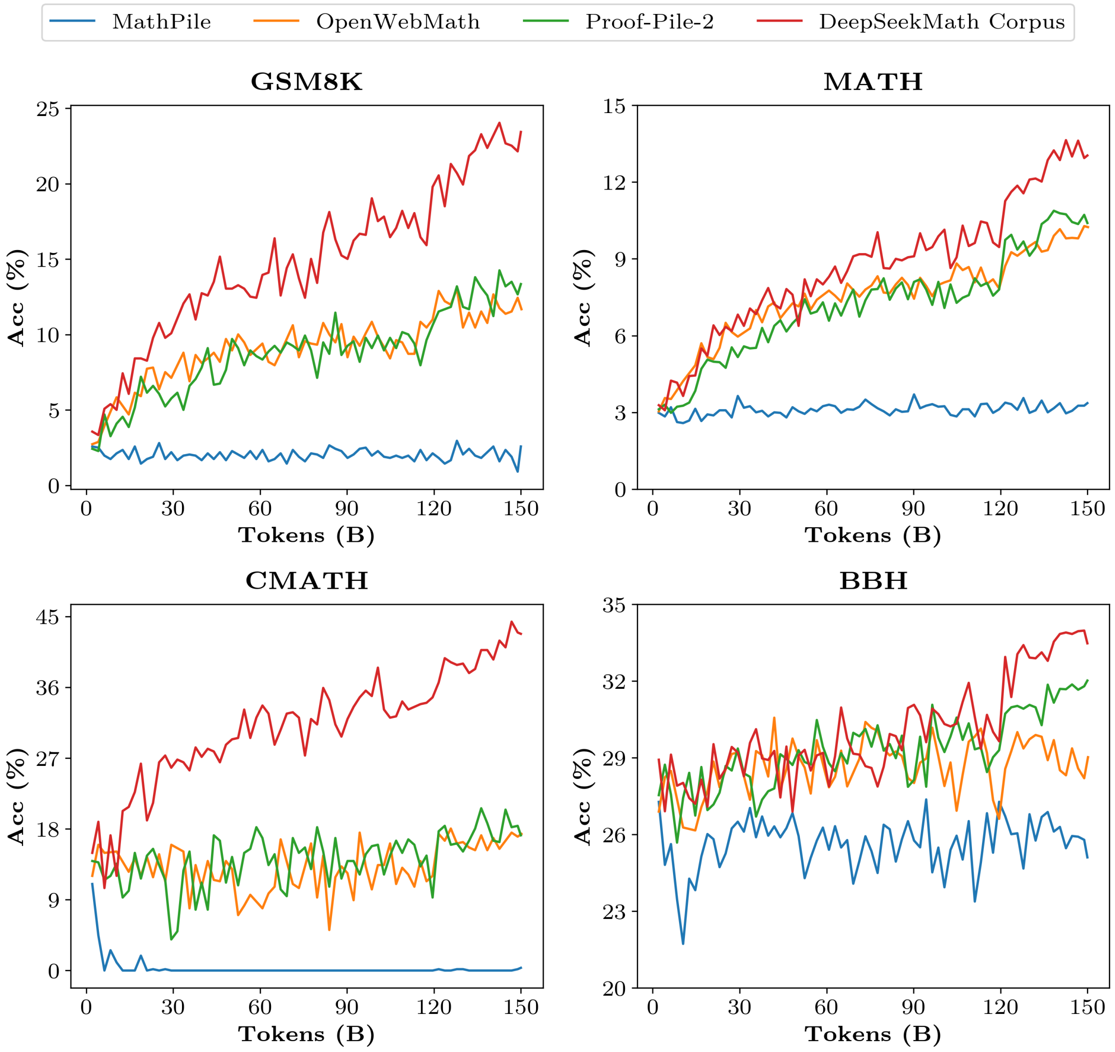

Figure 3 | Benchmark curves of DeepSeek-LLM 1.3B trained on different mathematical corpora.

- Multilingual: The DeepSeekMath Corpus encompasses data in multiple languages, predominantly featuring English and Chinese as the two most represented languages. As shown in Table 1, training on the DeepSeekMath Corpus enhances mathematical reasoning performance in both English and Chinese. In contrast, existing mathematical corpora, which are primarily English-centric, show limited improvement and may even hinder performance in Chinese mathematical reasoning.
  DeepSeekMath Corpus 包含多种语言的数据，主要以英语和中文为代表的两种语言。如表 1 所示，在 DeepSeekMath Corpus 上训练可提高英语和中文的数学推理性能。相比之下，现有的数学语料库主要以英语为中心，改进效果有限，甚至可能阻碍中文数学推理的表现。
- Large-scale: The DeepSeekMath Corpus is several times larger than existing mathematical corpora. As depicted in Figure 3, DeepSeek-LLM 1.3B, when trained on the DeepSeekMath Corpus, shows a steeper learning curve along with more lasting improvements. In contrast, the baseline corpora are much smaller, and have already been repeated multiple rounds during training, with the resulting model performance quickly reaching a plateau.
  DeepSeekMath Corpus 比现有的数学语料库大几倍。如图 3 所示，DeepSeek-LLM 1.3B 在 DeepSeek-Math Corpus 上进行训练时，学习曲线更陡峭，改进也更持久。相比之下，基线语料库要小得多，并且在训练过程中已经重复了多轮，最终的模型性能很快达到了稳定状态。

2.3. Training and Evaluating DeepSeekMath-Base 7B

In this section, we introduce DeepSeekMath-Base 7B, a base model with strong reasoning abilities, especially in mathematics. Our model is initialized with DeepSeek-Coder-Base-v1.5 7B (15) and trained for 500B tokens. The distribution of the data is as follows: 56% is from the DeepSeekMath Corpus, 4% from AlgebraicStack, 10% from arXiv, 20% is Github code, and the remaining 10% is natural language data from Common Crawl in both English and Chinese. We mainly adopt the training setting specified in Section 2.2.1, except that we set the maximum value of the learning rate to 4.2e-4 and use a batch size of 10M tokens.
在本节中，我们介绍了 DeepSeekMath-Base 7B，这是一个具有强大推理能力（尤其是在数学方面）的基础模型。我们的模型使用 DeepSeek-Coder-Base-v1.5 7B (15) 初始化，并针对 500B 个 token 进行训练。数据分布如下：56% 来自 DeepSeekMath Corpus，4% 来自 AlgebraicStack，10% 来自 arXiv，20% 是 Github 代码，其余 10% 是来自 Common Crawl 的英文和中文自然语言数据。我们主要采用 2.2.1 节中指定的训练设置，但我们将学习率的最大值设置为 4.2e-4，并使用 10M 个 token 的批处理大小。

We conduct a comprehensive assessment of the mathematical capabilities of DeepSeekMath-Base 7B, focusing on its ability to produce self-contained mathematical solutions without relying on external tools, solve mathematical problems using tools, and conduct formal theorem proving. Beyond mathematics, we also provide a more general profile of the base model, including its performance of natural language understanding, reasoning, and programming skills.
我们对 DeepSeekMath-Base 7B 的数学能力进行了全面评估，重点关注其在不依赖外部工具的情况下生成自包含数学解决方案、使用工具解决数学问题以及进行正式定理证明的能力。除了数学之外，我们还提供了基础模型的更一般概况，包括其自然语言理解、推理和编程技能的表现。

Mathematical Problem Solving with Step-by-Step Reasoning We evaluate DeepSeekMath-Base's performance of solving mathematical problems using few-shot chain-of-thought prompting (50), across eight benchmarks in English and Chinese. These benchmarks encompass quantitative reasoning (e.g., GSM8K (9), MATH (17), and CMATH (51)) and multiple-choice problems (e.g., MMLU-STEM (16) and Gaokao-MathQA (61)), covering diverse fields of mathematics from elementary to college-level complexity.
我们使用少样本思维链提示 (50) 在 8 个英文和中文基准上评估了 DeepSeekMath-Base 解决数学问题的性能。这些基准涵盖定量推理（例如，GSM8K (9)、MATH (17) 和 CMATH (51)）和多项选择题（例如，MMLU-STEM (16) 和 Gaokao-MathQA (61)），涵盖了从小学到大学水平的复杂程度的各种数学领域。

As shown in Table 2, DeepSeekMath-Base 7B leads in performance across all eight benchmarks among the open-source base models (including the widely-used general model Mistral 7B (21) and the recently released Llemma 34B (3) which underwent math training on Proof-Pile-2 (3)). Notably, on the competition-level MATH dataset, DeepSeekMath-Base surpasses existing open-source base models by over 10% absolute, and outperforms Minerva 540B (25), a closed-source base model 77 times larger which builds on PaLM (26) and is further trained on mathematical texts.
如表 2 所示，DeepSeekMath-Base 7B 在开源基础模型（包括广泛使用的通用模型 Mistral 7B (21) 和最近发布的在 Proof-Pile-2 (3) 上进行数学训练的 Llemma 34B (3)）的所有 8 个基准测试中均表现领先。值得注意的是，在竞赛级 MATH 数据集上，DeepSeekMath-Base 绝对超越现有开源基础模型 10% 以上，并优于 Minerva 540B (25)，后者是一个比其大 77 倍的闭源基础模型，它基于 PaLM (26) 构建，并在数学文本上进行进一步训练。

Mathematical Problem Solving with Tool Use We evaluate program-aided mathematical reasoning on GSM8K and MATH using few-shot program-of-thought prompting (8; 13). Models are prompted to solve each problem by writing a Python program where libraries such as math and sympy can be utilized for intricate computations. The execution result of the program is evaluated as the answer. As shown in Table 3, DeepSeekMath-Base 7B outperforms the prior state-of-the-art Llemma 34B.
我们使用少样本思维程序提示 (8; 13) 评估了 GSM8K 和 MATH 上的程序辅助数学推理。通过编写 Python 程序提示模型解决每个问题，其中可以使用 math 和 sympy 等库进行复杂的计算。程序的执行结果被评估为答案。如表 3 所示，DeepSeekMath-Base 7B 的表现优于之前最先进的 Llemma 34B。

Formal Mathematics Formal proof automation is beneficial to ensure the accuracy and reliability of mathematical proofs and enhance efficiency, with increasing attention in recent years. We evaluate DeepSeekMath-Base 7B on the task of informal-to-formal proving from (20) which is to generate a formal proof based on an informal statement, a formal counterpart of the statement, and an informal proof. We evaluate on miniF2F (60), a benchmark for formal Olympiad-level mathematics, and generate a formal proof in Isabelle for each problem with few-shot prompting. Following (20), we leverage models to generate proof sketches, and execute

Model	Size	English Benchmarks					Chinese Benchmarks		
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
Closed-Source Base Model									
Minerva	7B	16.2%	14.1%	7.7%	-	35.6%	-	-	-
Minerva	62B	52.4%	27.6%	12.0%	-	53.9%	-	-	-
Minerva	540B	58.8%	33.6%	17.6%	-	63.9%	-	-	-
Open-Source Base Model									
Mistral	7B	40.3%	14.3%	9.2%	71.9%	51.1%	44.9%	5.1%	23.4%
Llemma	7B	37.4%	18.1%	6.3%	59.4%	43.1%	43.4%	11.9%	23.6%
Llemma	34B	54.0%	25.3%	10.3%	71.9%	52.9%	56.1%	11.9%	26.2%
DeepSeekMath-Base	7B	64.2%	36.2%	15.4%	84.4%	56.5%	71.7%	20.3%	35.3%

Table 2 | Comparisons between DeepSeekMath-Base 7B and strong base models on English and Chinese mathematical benchmarks. Models are evaluated with chain-of-thought prompting. Minerva results are quoted from (25). DeepSeekMath-Base 7B 与强基模型在英语和中文数学基准上的比较。使用思路链提示来评估模型。Minerva 结果引自 (25)。

Model	Size	Problem Solving w/ Tools		Informal-to-Formal Proving	
		GSM8K+Python	MATH+Python	miniF2F-valid	miniF2F-test
Mistral	7B	48.5%	18.2%	18.9%	18.0%
CodeLlama	7B	27.1%	17.2%	16.3%	17.6%
CodeLlama	34B	52.7%	23.5%	18.5%	18.0%
Llemma	7B	41.0%	18.6%	20.6%	22.1%
Llemma	34B	64.6%	26.3%	21.0%	21.3%
DeepSeekMath-Base	7B	66.9%	31.4%	25.8%	24.6%

Table 3 | Few-shot evaluation of base models' ability to solve mathematical problems using tools and the ability to conduct informal-to-formal theorem proving in Isabelle.
对基础模型使用工具解决数学问题的能力以及在 Isabelle 中进行非正式到正式定理证明的能力进行少量评估。

the off-the-shelf automated prover Sledgehammer (35) to fill in the missing details. As shown in Table 3, DeepSeekMath-Base 7B demonstrates strong performance in proof autoformalization.
形式化证明自动化有利于保证数学证明的准确性和可靠性，提高证明效率，近年来受到越来越多的关注。我们在 (20) 的非形式化到形式化证明任务上评估了 DeepSeekMath-Base 7B，即基于非形式化语句、该语句的形式对应项和形式化证明生成形式化证明。我们在 miniF2F (60)（形式化奥林匹克数学的基准）上进行评估，并在 Isabelle 中为每个问题生成形式化证明，只需少量提示即可。按照 (20)，我们利用模型生成证明草图，并执行现成的自动证明器 Sledgehammer (35) 来填补缺失的细节。如表 3 所示，DeepSeekMath-Base 7B 在证明自动形式化方面表现出色。

Model	Size	MMLU	BBH	HumanEval (Pass@1)	MBPP (Pass@1)
Mistral	7B	62.4%	55.7%	28.0%	41.4%
DeepSeek-Coder-Base-v1.5†	7B	42.9%	42.9%	40.2%	52.6%
DeepSeek-Coder-Base-v1.5	7B	49.1%	55.2%	43.2%	60.4%
DeepSeekMath-Base	7B	54.9%	59.5%	40.9%	52.6%

Table 4 | Evaluation on natural language understanding, reasoning, and code benchmarks. DeepSeek-Coder-Base-v1.5† is the checkpoint right before learning rate decay, which is used to train DeepSeekMath-Base. On MMLU and BBH, we use few-shot chain-of-thought prompting. On HumanEval and MBPP, we evaluate model performance under the zero-shot setting and a few-shot setting, respectively.

Natural Language Understanding, Reasoning, and Code We evaluate model performance of natural language understanding on MMLU (16), reasoning on BBH (43), and coding capabilities on HumanEval (7) and MBPP (2). As shown in Table 4, DeepSeekMath-Base 7B exhibits significant enhancements in performance on MMLU and BBH over its precursor, DeepSeek-Coder-Base-v1.5 (15), illustrating the positive impact of math training on language understanding and reasoning. Additionally, by including code tokens for continual training, DeepSeekMath-Base 7B effectively maintains the performance of DeepSeek-Coder-Base-v1.5 on the two coding benchmarks. Overall, DeepSeekMath-Base 7B significantly outperforms the general model Mistral 7B (21) on the three reasoning and coding benchmarks.
我们评估了模型在 MMLU (16) 上的自然语言理解性能、在 BBH (43) 上的推理性能以及在 HumanEval (7) 和 MBPP (2) 上的编码能力。如表 4 所示，DeepSeekMath-Base 7B 在 MMLU 和 BBH 上的性能显著优于其前身 DeepSeek-Coder-Base-v1.5 (15)，这说明了数学训练对语言理解和推理的积极影响。此外，通过包含用于持续训练的代码 token，DeepSeekMath-Base 7B 在两个编码基准上有效地保持了 DeepSeek-Coder-Base-v1.5 的性能。总体而言，DeepSeekMath-Base 7B 在三个推理和编码基准上的表现明显优于通用模型 Mistral 7B (21)。

3. Supervised Fine-Tuning

3.1. SFT Data Curation

We construct a mathematical instruction-tuning dataset covering English and Chinese problems from different mathematical fields and of varying complexity levels: problems are paired with solutions in chain-of-thought (CoT) (50), program-of-thought (PoT) (8; 13), and tool-integrated reasoning format (14). The total number of training examples is 776K.
我们构建了一个数学指令调优数据集，涵盖不同数学领域和不同复杂程度的英语和中文问题：问题与解决方案以思路链 (CoT) (50)、思路程序 (PoT) (8; 13) 和工具集成推理格式 (14) 配对。训练示例总数为 776K。

- English mathematical datasets: We annotate GSM8K and MATH problems with tool-integrated solutions, and adopt a subset of MathInstruct (59) along with the training set of Lila-OOD (30) where problems are solved with CoT or PoT. Our English collection covers diverse fields of mathematics, e.g., algebra, probability, number theory, calculus, and geometry.
  我们用工具集成解决方案注释 GSM8K 和 MATH 问题，并采用 MathInstruct (59) 的子集以及 Lila-OOD (30) 的训练集，其中使用 CoT 或 PoT 解决问题。我们的英文合集涵盖了数学的不同领域，例如代数、概率、数论、微积分和几何。
- Chinese mathematical datasets: We collect Chinese K-12 mathematical problems spanning 76 sub-topics such as linear equations, with solutions annotated in both CoT and tool-integrated reasoning format.
  我们收集了中国 K-12 数学问题，涵盖 76 个子主题，例如线性方程，并以 CoT 和工具集成推理格式注释解决方案。

3.2. Training and Evaluating DeepSeekMath-Instruct 7B

In this section, we introduce DeepSeekMath-Instruct 7B which undergoes mathematical instruction tuning based on DeepSeekMath-Base. Training examples are randomly concatenated until reaching a maximum context length of 4K tokens. We train the model for 500 steps with a batch size of 256 and a constant learning rate of 5e-5.
在本节中，我们介绍了 DeepSeekMath-Instruct 7B，它基于 DeepSeekMath-Base 进行数学指令调整。训练示例被随机连接，直到达到 4K 个 token 的最大上下文长度。我们以 256 的批处理大小和 5e-5 的恒定学习率对模型进行 500 步训练。

We evaluate models' mathematical performance both without and with tool use, on 4 quantitative reasoning benchmarks in English and Chinese. We benchmark our model against the leading models of the time
我们根据 4 个英文和中文定量推理基准，评估模型在不使用工具和使用工具情况下的数学表现。我们将我们的模型与当时领先的模型进行对比：

- Closed-source models include: (1) the GPT family among which GPT-4 (32) and GPT-4 Code Interpreter [2] are the most capable ones, (2) Gemini Ultra and Pro (1), (3) Inflection-2 (19), (4) Grok-1 [3], as well as models recently released by Chinese companies including (5) Baichuan-3 [4], (6) the latest GLM-4 [5] from the GLM family (12). These models are for general purposes, most of which have undergone a series of alignment procedures.

[2] https://openai.com/blog/chatgpt-plugins#code-interpreter
[3] https://x.ai/model-card
[4] https://www.baichuan-ai.com
[5] https://open.bigmodel.cn/dev/api#glm-4

- Open-source models include: general models like (1) DeepSeek-LLM-Chat 67B (11), (2) Qwen 72B (4), (3) SeaLLM-v2 7B (31), and (4) ChatGLM3 6B (6), as well as models with enhancements in mathematics including (5) InternLM2-Math 20B [6] which builds on InternLM2 and underwent math training followed by instruction tuning, (6) Math-Shepherd-Mistral 7B which applys PPO training (40) to Mistral 7B (21) with a process-supervised reward model, (7) the WizardMath series (29) which improves mathematical reasoning in Mistral 7B and Llama-2 70B (45) using evolve-instruct (i.e., a version of instruction tuning that uses AI-evolved instructions) and PPO training with training problems primarily sourced from GSM8K and MATH, (8) MetaMath 70B (56) which is Llama-2 70B fine-tuned on an augmented version of GSM8K and MATH, (9) ToRA 34B (14) which is CodeLlama 34B fine-tuned to do tool-integrated mathematical reasoning, (10) MAmmoTH 70B (59) which is Llama-2 70B instruction-tuned on MathInstruct.

As shown in Table 5, under the evaluation setting where tool use is disallowed, DeepSeekMath-Instruct 7B demonstrates strong performance of step-by-step reasoning. Notably, on the competition-level MATH dataset, our model surpasses all open-source models and the majority of proprietary models (e.g., Inflection-2 and Gemini Pro) by at least 9% absolute. This is true even for models that are substantially larger (e.g., Qwen 72B) or have been specifically enhanced through math-focused reinforcement learning (e.g., WizardMath-v1.1 7B). While DeepSeekMath-Instruct rivals the Chinese proprietary models GLM-4 and Baichuan-3 on MATH, it still underperforms GPT-4 and Gemini Ultra.
如表 5 所示，在禁止使用工具的评估设置下，DeepSeekMath-Instruct 7B 表现出强大的逐步推理性能。值得注意的是，在竞赛级 MATH 数据集上，我们的模型至少以 9% 的绝对速度超越了所有开源模型和大多数专有模型（例如 Inflection-2 和 Gemini Pro）。即使对于大得多的模型（例如 Qwen 72B）或通过以数学为重点的强化学习（例如 WizardMath-v1.1 7B）进行了专门增强的模型，情况也是如此。虽然 DeepSeekMath-Instruct 在 MATH 上可与中国专有模型 GLM-4 和 Baichuan-3 相媲美，但它的表现仍然不及 GPT-4 和 Gemini Ultra。

Under the evaluation setting where models are allowed to integrate natural language reasoning and program-based tool use for problem solving, DeepSeekMath-Instruct 7B approaches an accuracy of 60% on MATH, surpassing all existing open-source models. On the other benchmarks, our model is competitive with DeepSeek-LLM-Chat 67B, the prior state-of-the-art that is 10 times larger.
在评估设置下，模型可以整合自然语言推理和基于程序的工具来解决问题，DeepSeekMath-Instruct 7B 在数学上的准确率接近 60%，超越了所有现有的开源模型。在其他基准测试中，我们的模型与 DeepSeek-LLM-Chat 67B 相媲美，后者是之前最先进的模型，规模是前者的 10 倍。

4. Reinforcement Learning

4.1. Group Relative Policy Optimization

Reinforcement learning (RL) has been proven to be effective in further improving the mathematical reasoning ability of LLMs after the Supervised Fine-Tuning (SFT) stage (29; 48). In this section, we introduce our efficient and effective RL algorithm, Group Relative Policy Optimization (GRPO).
强化学习 (RL) 已被证明能够有效地在监督微调 (SFT) 阶段 (29; 48) 之后进一步提高 LLM 的数学推理能力。在本节中，我们介绍了我们高效且有效的 RL 算法，即组相对策略优化 (GRPO)。

4.1.1. From PPO to GRPO

Proximal Policy Optimization (PPO) (40) is an actor-critic RL algorithm that is widely used in the RL fine-tuning stage of LLMs (33). In particular, it optimizes LLMs by maximizing the following surrogate objective
(40) 是一种演员-评论家 RL 算法，广泛应用于 LLM (33) 的 RL 微调阶段。具体来说，它通过最大化以下替代目标来优化 LLM：

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right], \quad (1)$$

where π_θ and $\pi_{\theta_{old}}$ are the current and old policy models, and q, o are questions and outputs sampled from the question dataset and the old policy $\pi_{\theta_{old}}$, respectively. ε is a clipping-related hyper-parameter introduced in PPO for stabilizing training. A_t is the advantage, which is computed by applying Generalized Advantage Estimation (GAE) (39), based on the rewards $\{r_{\geq t}\}$ and a learned value function V_ψ. Thus, in PPO, a value function needs to be trained alongside the policy model and to mitigate over-optimization of the reward model, the standard approach is to add a per-token KL penalty from a reference model in the reward at each token (33), i.e.,

[6] https://github.com/InternLM/InternLM-Math

Model	Size	English Benchmarks		Chinese Benchmarks	
		GSM8K	MATH	MGSM-zh	CMATH
Chain-of-Thought Reasoning					
Closed-Source Model					
Gemini Ultra	-	94.4%	53.2%	-	-
GPT-4	-	92.0%	52.9%	-	86.0%
Inflection-2	-	81.4%	34.8%	-	-
GPT-3.5	-	80.8%	34.1%	-	73.8%
Gemini Pro	-	86.5%	32.6%	-	-
Grok-1	-	62.9%	23.9%	-	-
Baichuan-3	-	88.2%	49.2%	-	-
GLM-4	-	87.6%	47.9%	-	-
Open-Source Model					
InternLM2-Math	20B	82.6%	37.7%	-	-
Qwen	72B	78.9%	35.2%	-	-
Math-Shepherd-Mistral	7B	84.1%	33.0%	-	-
WizardMath-v1.1	7B	83.2%	33.0%	-	-
DeepSeek-LLM-Chat	67B	84.1%	32.6%	74.0%	80.3%
MetaMath	70B	82.3%	26.6%	66.4%	70.9%
SeaLLM-v2	7B	78.2%	27.5%	64.8%	-
ChatGLM3	6B	72.3%	25.7%	-	-
WizardMath-v1.0	70B	81.6%	22.7%	64.8%	65.4%
DeepSeekMath-Instruct	7B	82.9%	46.8%	73.2%	84.6%
DeepSeekMath-RL	7B	88.2%	51.7%	79.6%	88.8%
Tool-Integrated Reasoning					
Closed-Source Model					
GPT-4 Code Interpreter	-	97.0%	69.7%	-	-
Open-Source Model					
InternLM2-Math	20B	80.7%	54.3%	-	-
DeepSeek-LLM-Chat	67B	86.7%	51.1%	76.4%	85.4%
ToRA	34B	80.7%	50.8%	41.2%	53.4%
MAmmoTH	70B	76.9%	41.8%	-	-
DeepSeekMath-Instruct	7B	83.7%	57.4%	72.0%	84.3%
DeepSeekMath-RL	7B	86.7%	58.8%	78.4%	87.6%

Table 5 | Performance of Open- and Closed-Source models with both Chain-of-Thought and Tool-Integrated Reasoning on English and Chinese Benchmarks. Scores in gray denote majority votes with 32 candidates; The others are Top1 scores. DeepSeekMath-RL 7B beats all open-source models from 7B to 70B, as well as the majority of closed-source models. Although DeepSeekMath-RL 7B is only further trained on chain-of-thought-format instruction tuning data of GSM8K and MATH, it improves over DeepSeekMath-Instruct 7B on all benchmarks.
英文和中文基准测试中，开源和闭源模型的思维链和工具集成推理性能。gray 中的分数表示 32 个候选者的多数票；其他分数为 Top1 分数。DeepSeekMath-RL 7B 击败了从 7B 到 70B 的所有开源模型，以及大多数闭源模型。虽然 DeepSeekMath-RL 7B 仅在 GSM8K 和 MATH 的思维链格式指令调优数据上进行了进一步训练，但它在所有基准测试中都比 DeepSeekMath-Instruct 7B 有所改进。

其中 π_θ 和 $\pi_{\theta_{old}}$ 分别是当前和旧的策略模型，q, o 分别是从问题数据集和旧策略 $\pi_{\theta_{old}}$ 中采样的问题和输出。ε 是 PPO 中引入的与裁剪相关的超参数，用于稳定训练。A_t 是优势，它是通过应用广义优势估计 (GAE) (39) 计算得出的，基于奖励 $\{r_{\geq t}\}$ 和学习到的价值函数 V_ψ。因此，在 PPO 中，需要与策略模型一起训练价值函数，为了减轻奖励模型的过度优化，标准方法是在每个 token (33) 的奖励中添加来自参考模型的每个 token KL 惩

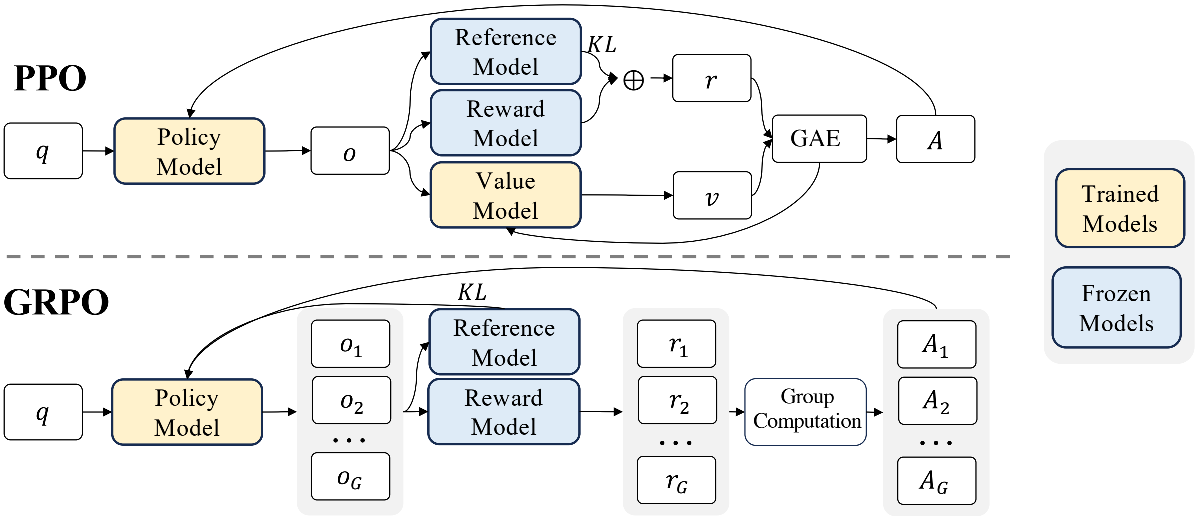

Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.
PPO 和我们的 GRPO 的演示。GRPO 放弃了价值模型，而是根据组分数估计基线，从而大大减少了训练资源。

罚，即

$$r_t = r_\varphi(q, o_{\le t}) - \beta \log \frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{ref}(o_t|q, o_{<t})}, \tag{2}$$

where r_φ is the reward model, π_{ref} is the reference model, which is usually the initial SFT model, and β is the coefficient of the KL penalty.
其中 r_φ 是奖励模型，π_{ref} 是参考模型，通常是初始 SFT 模型，β 是 KL 惩罚的系数。

As the value function employed in PPO is typically another model of comparable size as the policy model, it brings a substantial memory and computational burden. Additionally, during RL training, the value function is treated as a baseline in the calculation of the advantage for variance reduction. While in the LLM context, usually only the last token is assigned a reward score by the reward model, which may complicate the training of a value function that is accurate at each token. To address this, as shown in Figure 4, we propose Group Relative Policy Optimization (GRPO), which obviates the need for additional value function approximation as in PPO, and instead uses the average reward of multiple sampled outputs, produced in response to the same question, as the baseline. More specifically, for each question q, GRPO samples a group of outputs $\{o_1, o_2, \cdots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model by maximizing the following objective
由于 PPO 中使用的价值函数通常是与策略模型大小相当的另一个模型，因此它会带来大量的内存和计算负担。此外，在 RL 训练期间，价值函数被视为计算方差减少优势的基线。而在 LLM 上下文中，通常只有最后一个 token 才会被奖励模型分配奖励分数，这可能会使训练在每个 token 上都准确的价值函数变得复杂。为了解决这个问题，如图 4 所示，我们提出了组相对策略优化 (GRPO)，它消除了 PPO 中对额外价值函数近似的需求，而是使用针对同一问题产生的多个采样输出的平均奖励作为基线。更具体地说，对于每个问题 q，GRPO 从旧策略 $\pi_{\theta_{old}}$ 中采样一组输出 $\{o_1, o_2, \cdots, o_G\}$，然后通过最大化以下目标来优化策略模型：

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) &= \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\},\end{aligned} \tag{3}$$

where ε and β are hyper-parameters, and $\hat{A}_{i,t}$ is the advantage calculated based on relative rewards of the outputs inside each group only, which will be detailed in the following subsections. The group relative way that GRPO leverages to calculate the advantages, aligns well with the comparative nature of rewards models, as reward models are typically trained on datasets of comparisons between outputs on the same question. Also note that, instead of adding KL penalty in the reward, GRPO regularizes by directly adding the KL divergence between the trained policy and the reference policy to the loss, avoiding complicating the calculation of $\hat{A}_{i,t}$. And different from the KL penalty term used in (2), we estimate the KL divergence with the following unbiased estimator (38)
其中 ε 和 β 是超参数，$\hat{A}_{i,t}$ 是基于每个组内输出的相对奖励计算得出的优势，将在以下小节中详细介绍。GRPO 利用组相对方式计算优势，这与奖励模型的比较性质非常吻合，因为奖励模型通常在同一问题的输出比较数据

Algorithm 1 Iterative Group Relative Policy Optimization

```
Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models $r_\varphi$; task prompts $\mathcal{D}$; hyperparameters $\varepsilon$, $\beta$, $\mu$
policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, ..., I do
    reference model $\pi_{ref} \leftarrow \pi_\theta$
    for step = 1, ..., M do
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample $G$ outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{r_i\}_{i=1}^G$ for each sampled output $o_i$ by running $r_\varphi$
        Compute $\hat{A}_{i,t}$ for the $t$-th token of $o_i$ through group relative advantage estimation.
        for GRPO iteration = 1, ..., $\mu$ do
            Update the policy model $\pi_\theta$ by maximizing the GRPO objective (Equation 21)
    Update $r_\varphi$ through continuous training using a replay mechanism.
Output $\pi_\theta$
```

集上进行训练。还要注意，GRPO 不是在奖励中添加 KL 惩罚，而是通过将训练策略和参考策略之间的 KL 散度直接添加到损失中进行正则化，从而避免使 $\hat{A}_{i,t}$ 的计算复杂化。与 (2) 中使用的 KL 惩罚项不同，我们使用以下无偏估计量 (38) 来估计 KL 散度：

$$\mathbb{D}_{KL}\left[\pi_\theta || \pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1, \tag{4}$$

which is guaranteed to be positive.
保证是正的。

4.1.2. Outcome Supervision RL with GRPO

Formally, for each question q, a group of outputs $\{o_1, o_2, \cdots, o_G\}$ are sampled from the old policy model $\pi_{\theta_{old}}$. A reward model is then used to score the outputs, yielding G rewards $\mathbf{r} = \{r_1, r_2, \cdots, r_G\}$ correspondingly. Subsequently, these rewards are normalized by subtracting the group average and dividing by the group standard deviation. Outcome supervision provides the normalized reward at the end of each output o_i and sets the advantages $\hat{A}_{i,t}$ of all tokens in the output as the normalized reward, i.e., $\hat{A}_{i,t} = \widetilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$, and then optimizes the policy by maximizing the objective defined in equation (3).
正式来说，对于每个问题 q，从旧策略模型 $\pi_{\theta_{old}}$ 中抽取一组输出 $\{o_1, o_2, \cdots, o_G\}$。然后使用奖励模型对输出进行评分，从而相应地产生 G 个奖励 $\mathbf{r} = \{r_1, r_2, \cdots, r_G\}$。随后，通过减去组平均值并除以组标准差来将这些奖励标准化。结果监督在每个输出 o_i 结束时提供归一化奖励，并将输出中所有 token 的优势 $\hat{A}_{i,t}$ 设为归一化奖励，即 $\hat{A}_{i,t} = \widetilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$，然后通过最大化方程 (3) 中定义的目标来优化策略。

4.1.3. Process Supervision RL with GRPO

Outcome supervision only provides a reward at the end of each output, which may not be sufficient and efficient to supervise the policy in complex mathematical tasks. Following (48), we also explore process supervision, which provides a reward at the end of each reasoning step. Formally, given the question q and G sampled outputs $\{o_1, o_2, \cdots, o_G\}$, a process reward model is used to score each step of the outputs, yielding corresponding rewards: $\mathbf{R} = \{\{r_1^{index(1)}, \cdots, r_1^{index(K_1)}\}, \cdots, \{r_G^{index(1)}, \cdots, r_G^{index(K_G)}\}\}$, where $index(j)$ is the end token index of the j-th step, and K_i is the total number of steps in the i-th output. We also normalize these rewards with the average and the standard deviation, i.e., $\widetilde{r}_i^{index(j)} = \frac{r_i^{index(j)} - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$. Subsequently, the process supervision calculates the advantage of each token as the sum of the normalized rewards from the following steps, i.e., $\hat{A}_{i,t} = \sum_{index(j) \geq t} \widetilde{r}_i^{index(j)}$, and then optimizes the policy by maximizing the objective defined in equation (3).
结果监督仅在每次输出结束时提供奖励，这可能不足以有效地监督复杂数学任务中的策略。继 (48) 之后，我们还探索了过程监督，它在每个推理步骤结束时提供奖励。正式地，给定问题 q 和 G 个采样输出 $\{o_1, o_2, \cdots, o_G\}$，使用过程奖励模型对输出的每个步骤进行评分，得到相应的奖励：$\mathbf{R} = \{\{r_1^{index(1)}, \cdots, r_1^{index(K_1)}\}, \cdots, \{r_G^{index(1)}, \cdots, r_G^{index(K_G)}\}\}$，其中 $index(j)$ 是第 j 步的结束 token 索引，K_i 是第 i 个输出中的总步骤数。我们还用平均值和标准差对这些奖励进行归一化，即 $\widetilde{r}_i^{index(j)} = \frac{r_i^{index(j)} - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$。随后，流程监督将每个 token 的优势计算为以下步骤中归一化奖励的总和，即 $\hat{A}_{i,t} = \sum_{index(j) \geq t} \widetilde{r}_i^{index(j)}$，然后通过最大化方程 (3) 中定义的目标来优化策略。

4.1.4. Iterative RL with GRPO

As the reinforcement learning training process progresses, the old reward model may not be sufficient to supervise the current policy model. Therefore, we also explore the iterative RL with GRPO. As shown in Algorithm 1, in iterative GRPO, we generate new training sets for the reward model based on the sampling results from the policy model and continually train the old reward model using a replay mechanism that incorporates 10% of historical data. Then, we set the reference model as the policy model, and continually train the policy model with the new reward model.
随着强化学习训练过程的进展，旧的奖励模型可能不足以监督当前的策略模型。因此，我们还探索了使用 GRPO 的迭代强化学习。如算法 1 所示，在迭代 GRPO 中，我们根据策略模型的采样结果为奖励模型生成新的训练集，并使用包含 10% 历史数据的重放机制不断训练旧的奖励模型。然后，我们将参考模型设置为策略模型，并使用新的奖励模型不断训练策略模型。

4.2. Training and Evaluating DeepSeekMath-RL

We conduct RL based on DeepSeekMath-Instruct 7B. The training data of RL are chain-of-thought-format questions related to GSM8K and MATH from the SFT data, which consists of around 144K questions. We exclude other SFT questions to investigate the impact of RL on benchmarks that lack data throughout the RL phase. We construct the training set of reward models following (48). We train our initial reward model based on the DeepSeekMath-Base 7B with a learning rate of 2e-5. For GRPO, we set the learning rate of the policy model as 1e-6. The KL coefficient is 0.04. For each question, we sample 64 outputs. The max length is set to 1024, and the training batch size is 1024. The policy model only has a single update following each exploration stage. We evaluate DeepSeekMath-RL 7B on benchmarks following DeepSeekMath-Instruct 7B. For DeepSeekMath-RL 7B, GSM8K and MATH with chain-of-thought reasoning can be regarded as in-domain tasks and all the other benchmarks can be regarded as out-of-domain tasks.
我们基于 DeepSeekMath-Instruct 7B 进行 RL。RL 的训练数据是来自 SFT 数据、与 GSM8K 和 MATH 相关的思路链格式问题，其中包含约 144K 个问题。我们排除其他 SFT 问题，以调查 RL 对整个 RL 阶段缺乏数据的基准的影响。我们按照 (48) 构建奖励模型的训练集。我们基于 DeepSeekMath-Base 7B 训练我们的初始奖励模型，学习率为 2e-5。对于 GRPO，我们将策略模型的学习率设置为 1e-6。KL 系数为 0.04。对于每个问题，我们抽样 64 个输出。最大长度设置为 1024，训练批次大小为 1024。策略模型在每个探索阶段后仅进行一次更新。我们在遵循 DeepSeekMath-Instruct 7B 的基准上评估 DeepSeekMath-RL 7B。对于 DeepSeekMath-RL 7B，GSM8K 和具有思路链推理的 MATH 可视为域内任务，而所有其他基准可视为域外任务。

Table 5 demonstrates the performance of open- and closed-source models with both chain-of-thought and tool-integrated reasoning on English and Chinese benchmarks. We find that: 1) DeepSeekMath-RL 7B attains accuracies of 88.2% and 51.7% on GSM8K and MATH, respectively, utilizing chain-of-thought reasoning. This performance surpasses that of all open-source models in the 7B to 70B range, as well as the majority of closed-source models. 2) Crucially, DeepSeekMath-RL 7B is only trained on chain-of-thought-format instruction tuning data of GSM8K and MATH, starting from DeepSeekMath-Instruct 7B. Despite the constrained scope of its training data, it outperforms DeepSeekMath-Instruct 7B across all evaluation metrics, showcasing the effectiveness of reinforcement learning.
表 5 展示了在英语和中文基准上采用思路链推理和工具集成推理的开源和闭源模型的性能。我们发现：1）DeepSeekMath-RL 7B 利用思路链推理在 GSM8K 和 MATH 上分别达到 88.2% 和 51.7% 的准确率。这一性能超过了 7B 到 70B 范围内的所有开源模型以及大多数闭源模型。2）至关重要的是，DeepSeekMath-RL 7B 仅在 GSM8K 和 MATH 的思路链格式指令调优数据上进行训练，从 DeepSeekMath-Instruct 7B 开始。尽管其训练数据范围有限，但它在所有评估指标上都优于 DeepSeekMath-Instruct 7B，展示了强化学习的有效性。

5. Discussion

In this section, we will share our findings in pre-training and RL experiments.
在本节中，我们将分享我们在预训练和 RL 实验中的发现。

5.1. Lessons Learnt in Pre-Training

We first share our experience in pre-training. Unless otherwise specified, we will adhere to the training settings outlined in Section 2.2.1. It is worth noting that, when referring to the DeepSeekMath Corpus in this section, we use an 89B-token dataset from the second iteration of the data collection process.
我们首先分享我们在预训练方面的经验。除非另有说明，否则我们将遵守 2.2.1 节中概述的训练设置。值得注意的是，在本节中引用 DeepSeekMath 语料库时，我们使用来自数据收集过程第二次迭代的 89B-token 数据集。

Training Setting	Training Tokens			w/o Tool Use			w/ Tool Use	
	General	Code	Math	GSM8K	MATH	CMATH	GSM8K+Python	MATH+Python
No Continual Training	–	–	–	2.9%	3.0%	12.3%	2.7%	2.3%
Two-Stage Training								
Stage 1: General Training	400B	–	–	2.9%	3.2%	14.8%	3.3%	2.3%
Stage 2: Math Training	–	–	150B	19.1%	14.4%	37.2%	14.3%	6.7%
Stage 1: Code Training	–	400B	–	5.9%	3.6%	19.9%	12.4%	10.0%
Stage 2: Math Training	–	–	150B	21.9%	15.3%	39.7%	17.4%	9.4%
One-Stage Training								
Math Training	–	–	150B	20.5%	13.1%	37.6%	11.4%	6.5%
Code & Math Mixed Training	–	400B	150B	17.6%	12.1%	36.3%	19.7%	13.5%

Table 6 | Investigation of how code affects mathematical reasoning under different training settings. We experiment with DeepSeek-LLM 1.3B, and evaluate its mathematical reasoning performance without and with tool use via few-shot chain-of-thought prompting and few-shot program-of-thought prompting, respectively.
研究代码在不同训练环境下如何影响数学推理。我们使用 DeepSeek-LLM 1.3B 进行实验，并分别通过少量思维链提示和少量思维程序提示来评估其不使用和使用工具的数学推理性能。

5.1.1. Code Training Benefits Mathematical Reasoning

A popular yet unverified hypothesis suggests that code training improves reasoning. We attempt to offer a partial response to this, particularly within the mathematical domain: code training improves models' ability to do mathematical reasoning both with and without tool use.
一个流行但未经证实的假设表明，代码训练可以提高推理能力。我们试图对此做出部分回应，特别是在数学领域：代码训练提高了模型在使用和不使用工具的情况下进行数学推理的能力。

To study how code training affects mathematical reasoning, we experimented with the following two-stage training and one-stage training settings
为了研究代码训练如何影响数学推理，我们尝试了以下两阶段训练和单阶段训练设置：

Two-Stage Training

- Code Training for 400B Tokens → Math Training for 150B Tokens: We train DeepSeek-LLM 1.3B for 400B code tokens followed by 150B math tokens;
  针对 400B **个** token **进行代码训练** → **针对** 150B **个** token **进行数学训练**: 我们针对 400B 个代码 token 训练 DeepSeek-LLM 1.3B，然后针对 150B 个数学 token 进行训练；
- General Training for 400B Tokens → Math Training for 150B Tokens: As a control experiment, we also experiment with general tokens (sampled from a large-scale general corpus created by DeepSeek-AI) instead of code tokens in the first stage of training, in an attempt to investigate the advantages of code tokens over general tokens in improving mathematical reasoning.
  400B Tokens **通用训练** → 150B Tokens **数学训练**: 作为对照实验，我们还在训练的第一阶段用通用 token（从 DeepSeek-AI 创建的大规模通用语料库中采样）代替代码 token 进行实验，试图探究代码 token 在提高数学推理能力方面相对于通用 token 的优势。

One-Stage Training

- Math Training for 150B Tokens: We train DeepSeek-LLM 1.3B for 150B math tokens;
  针对 150B **个** token **进行数学训练**: 我们针对 150B 个数学 token 训练 DeepSeek-LLM 1.3B；
- Training on a mixture of 400B Code Tokens and 150B Math Tokens: Math training following code training degrades coding performance. We investigate whether code tokens, when mixed with math tokens for one-stage training, would still improve mathematical reasoning and also alleviate the problem of catastrophic forgetting.
  使用 400B **代码** token **和** 150B **数学** token **进行混合训练**：代码训练之后进行数学训练会降低编码性能。我们调查当将代码 token 与数学 token 混合进行单阶段训练时，是否仍能改善数学推理并缓解灾难性遗忘的问题。

Results Table 6 and Table 7 demonstrate the downstream performance under different training settings.
表 6 和表 7 展示了不同训练设置下的下游性能。

Code training benefits program-aided mathematical reasoning, both under the two-stage training and one-

Training Setting	Training Tokens			MMLU	BBH	HumanEval (Pass@1)	MBPP (Pass@1)
	General	Code	Math				
No Continual Training	–	–	–	24.5%	28.1%	12.2%	13.0%
Two-Stage Training							
Stage 1: General Training	400B	–	–	25.9%	27.7%	15.2%	13.6%
Stage 2: Math Training	–	–	150B	33.1%	32.7%	12.8%	13.2%
Stage 1: Code Training	–	400B	–	25.0%	31.5%	25.0%	40.0%
Stage 2: Math Training	–	–	150B	36.2%	35.3%	12.2%	17.0%
One-Stage Training							
Math Training	–	–	150B	32.3%	32.5%	11.6%	13.2%
Code & Math Mixed Training	–	400B	150B	33.5%	35.6%	29.3%	39.4%

Table 7 | Investigation of how different settings of code and math training affect model performance of language understanding, reasoning, and coding. We experiment with DeepSeek-LLM 1.3B. We evaluate the models on MMLU and BBH using few-shot chain-of-thought prompting. On HumanEval and MBPP, we conduct zero-shot and few-shot evaluations, respectively.

Model	Size	ArXiv Corpus	English Benchmarks					Chinese Benchmarks		
			GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
DeepSeek-LLM	1.3B	No Math Training	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
		MathPile	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
		ArXiv-RedPajama	3.3%	3.4%	4.0%	9.4%	9.0%	7.4%	0.8%	2.3%
DeepSeek-Coder-Base-v1.5	7B	No Math Training	29.0%	12.5%	6.6%	40.6%	38.1%	45.9%	5.9%	21.1%
		MathPile	23.6%	11.5%	7.0%	46.9%	35.8%	37.9%	4.2%	25.6%
		ArXiv-RedPajama	28.1%	11.1%	7.7%	50.0%	35.2%	42.6%	7.6%	24.8%

Table 8 | Effect of math training on different arXiv datasets. Model performance is evaluated with few-shot chain-of-thought prompting.

ArXiv Corpus	miniF2F-valid	miniF2F-test
No Math Training	20.1%	21.7%
MathPile	16.8%	16.4%
ArXiv-RedPajama	14.8%	11.9%

Table 9 | Effect of math training on different arXiv corpora, the base model being DeepSeek-Coder-Base-v1.5 7B. We evaluate informal-to-formal proving in Isabelle.

stage training settings. As shown in Table 6, under the two-stage training setting, code training alone already significantly enhances the ability to solve GSM8K and MATH problems using Python. Math training in the second stage yields further improvements. Interestingly, under the one-stage training setting, mixing code tokens and math tokens effectively mitigates the issue of catastrophic forgetting that arises from two-stage training, and also synergizes coding (Table 7) and program-aided mathematical reasoning (Table 6).
无论是在两阶段训练还是单阶段训练设置下，代码训练都有利于程序辅助数学推理。如表 6 所示，在两阶段训练设置下，仅代码训练就已经显著提高了使用 Python 解决 GSM8K 和数学问题的能力。第二阶段的数学训练带来了进一步的改进。有趣的是，在单阶段训练设置下，混合代码 token 和数学 token 可以有效缓解两阶段训练带来的灾难性遗忘问题，并协同编码（表 7）和程序辅助数学推理（表 6）。

Code training also improves mathematical reasoning without tool use. Under the two-stage training setting, the initial stage of code training already results in moderate enhancements. It also boosts the efficiency of the subsequent math training, eventually leading to the best performance. However, combining code tokens and math tokens for one-stage training compromises mathematical reasoning without tool use. One conjecture is that DeepSeek-LLM 1.3B, due to its limited scale, lacks the capacity to fully assimilate both code and mathematical data simultaneously.
代码训练也能提高不使用工具的数学推理能力。在两阶段训练设置下，代码训练的初始阶段已经取得了适度的提升。它还提高了后续数学训练的效率，最终达到最佳性能。然而，将代码 token 和数学 token 结合起来进行一阶段训练会损害不使用工具的数学推理能力。一种猜测是，由于规模有限，DeepSeek-LLM 1.3B 缺乏同时完全吸收代码和数学数据的能力。

5.1.2. ArXiv Papers Seem Ineffective in Improving Mathematical Reasoning

ArXiv papers are commonly included as a component of math pre-training data (3; 25; 36; 49). However, detailed analysis regarding their impact on mathematical reasoning has not been extensively conducted. Perhaps counter-intuitively, according to our experiments, arXiv papers seem ineffective in improving mathematical reasoning. We experiment with models of different sizes, including DeepSeek-LLM 1.3B and DeepSeek-Coder-Base-v1.5 7B (15), using arXiv corpora that underwent varied processing pipelines
ArXiv 论文通常被纳入数学预训练数据 (3; 25; 36; 49) 的组成部分。然而，关于它们对数学推理的影响的详细分析尚未得到广泛开展。也许与直觉相反，根据我们的实验，arXiv 论文似乎对提高数学推理无效。我们使用经过不同处理流程的 arXiv 语料库，对不同大小的模型进行了实验，包括 DeepSeek-LLM 1.3B 和 DeepSeek-Coder-Base-v1.5 7B (15)：

- MathPile (49): an 8.9B-token corpus developed with cleaning and filtering heuristic rules, over 85% of which are scientific arXiv papers;
  使用清理和过滤启发式规则开发的 8.9B token 语料库，其中 85
- ArXiv-RedPajama (10): the entirety of arXiv LaTeX files with preambles, comments, macros, and bibliographies removed, totaling 28.0B tokens.
  删除了序言、注释、宏和参考书目的全部 arXiv LaTeX 文件，总计 280 亿个 token。

In our experiments, we separately train DeepSeek-LLM 1.3B for 150B tokens and DeepSeek-Coder-Base-v1.5 7B for 40B tokens on each arXiv corpus. It seems that arXiv papers are ineffective in improving mathematical reasoning. When trained on a arXiv-only corpus, both models display no notable improvements or even deterioration across various mathematical benchmarks of different complexities employed in this study. These benchmarks include quantitative reasoning datasets like GSM8K and MATH (Table 8), multiple-choice challenges like MMLU-STEM (Table 8), and formal mathematics like miniF2F (Table 9).
在我们的实验中，我们分别在每个 arXiv 语料库上训练 DeepSeek-LLM 1.3B（150B 个 token）和 DeepSeek-Coder-Base-v1.5 7B（40B 个 token）。看来 arXiv 论文在提高数学推理能力方面效果不佳。在仅使用 arXiv 的语料库进行训练时，这两个模型在本研究中使用的各种复杂程度的数学基准上均未显示出明显的改善，甚至有所恶化。这些基准包括定量推理数据集（如 GSM8K 和 MATH）（表 8）、多项选择挑战（如 MMLU-STEM）（表 8）和形式数学（如 miniF2F）（表 9）。

However, this conclusion has its limitations and should be taken with a grain of salt. We have not yet studied
然而，这一结论有其局限性，应谨慎对待。我们尚未研究：

- The impact of arXiv tokens on specific math-related tasks not included in this research, such as informalization of theorems which is to convert formal statements or proofs to their informal versions;
  arXiv token 对本研究中未包括的特定数学相关任务的影响，例如定理的非形式化，即将正式的陈述或证明转换为其非正式版本；
- The effect of arXiv tokens when combined with other types of data;
  arXiv token 与其他类型的数据结合时的效果；
- Whether the benefits of arXiv papers would manifest themselves at a larger model scale.
  arXiv 论文的优势是否会在更大的模型规模上体现出来。

Thus, further exploration is required, which we leave for future studies.
因此，还需要进一步探索，我们留待未来的研究。

5.2. Insights of Reinforcement Learning

5.2.1. Towards to a Unified Paradigm

In this section, we provide a unified paradigm to analyze different training methods, such as SFT, RFT, DPO, PPO, GRPO, and further conduct experiments to explore the factors of the unified paradigm. Generally, the gradient with respect to the parameter θ of a training method can be written as
本节我们给出一个统一的范式来分析不同的训练方法，如 SFT、RFT、DPO、PPO、GRPO，并进一步进行实验探索统一范式的影响因素。一般来说，一个训练方法关于参数 θ 的梯度可以写成：

$$\nabla_{\theta} \mathcal{J}_{\mathcal{A}}(\theta) = \mathbb{E}[\underbrace{(q,o) \sim \mathcal{D}}_{Data\ Source}] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \underbrace{GC_{\mathcal{A}}(q,o,t,\pi_{rf})}_{Gradient\ Coefficient} \nabla_{\theta} \log \pi_{\theta}(o_t|q,o_{<t}) \right). \tag{5}$$

There exist three key components: 1) Data Source $\mathcal{D}$, which determines the training data; 2) Reward Function π_{rf}, which is the source of the training reward signal; 3) Algorithm $\mathcal{A}$: which processes the training data and

Methods	Data Source	Reward Function	Gradient Coefficient
SFT	$q, o \sim P_{sft}(Q, O)$	-	1
RFT	$q \sim P_{sft}(Q)$, $o \sim \pi_{sft}(O\|q)$	Rule	Equation 10
DPO	$q \sim P_{sft}(Q)$, $o^+, o^- \sim \pi_{sft}(O\|q)$	Rule	Equation 14
Online RFT	$q \sim P_{sft}(Q)$, $o \sim \pi_\theta(O\|q)$	Rule	Equation 10
PPO	$q \sim P_{sft}(Q)$, $o \sim \pi_\theta(O\|q)$	Model	Equation 18
GRPO	$q \sim P_{sft}(Q)$, $\{o_i\}_{i=1}^G \sim \pi_\theta(O\|q)$	Model	Equation 21

Table 10 | The data source and gradient coefficient of different methods. P_{sft} denotes the data distribution of supervised fine-tuning datasets. $\pi_{\theta_{sft}}$ and π_θ denote the supervised fine-tuned model and the real-time policy model during the online training process, respectively.

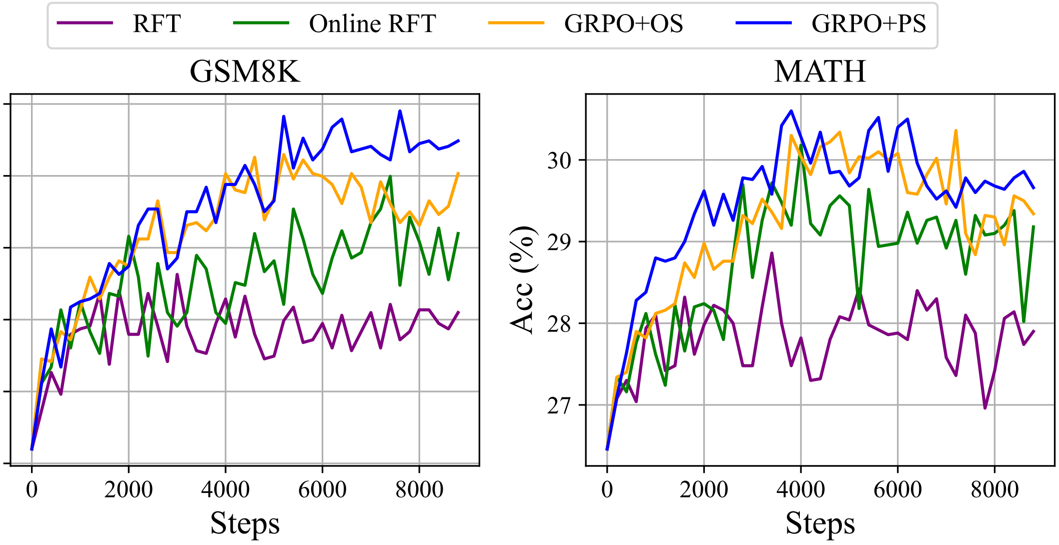

Figure 5 | Performance of the DeepSeekMath-Instruct 1.3B model, which was further trained using various methods, on two benchmarks.

the reward signal to the gradient coefficient *GC* that determines the magnitude of the penalty or reinforcement for the data. We analyze several representative methods based on such a unified paradigm:

- Supervised Fine-tuning (SFT): SFT fine-tunes pretrained model on human selected SFT data.
- Rejection Sampling Fine-tuning (RFT): RFT further fine-tunes the SFT model on the filtered outputs sampled from the SFT model based on SFT questions. RFT filters the outputs based on the correctness of their answers.
- Direct Preference Optimization (DPO): DPO further refines the SFT model by fine-tuning it on augmented outputs sampled from the SFT model, using pair-wise DPO loss.
- Online Rejection Sampling Fine-tuning (Online RFT): Different from RFT, Online RFT initiates the policy model using the SFT model and refines it by fine-tuning with the augmented outputs sampled from the real-time policy model.
- PPO/GRPO: PPO/GRPO initializes the policy model using the SFT model and reinforces it with the outputs sampled from the real-time policy model.

We summarize the components of these methods in Table 10. Please refer to Appendix A.1 for a more detailed derivation process.

Observation about Data Source We divide the data source into two categories, online sampling, and offline sampling. Online sampling denotes that the training data is from the exploration results of the real-time training policy model, while offline sampling denotes that the training data is from the sampling results of the initial SFT model. RFT and DPO follow the offline style, while Online RFT and GRPO follow the online style.
我们将数据来源分为两类，在线采样和离线采样。在线采样表示训练数据来自实时训练策略模型的探索结果，

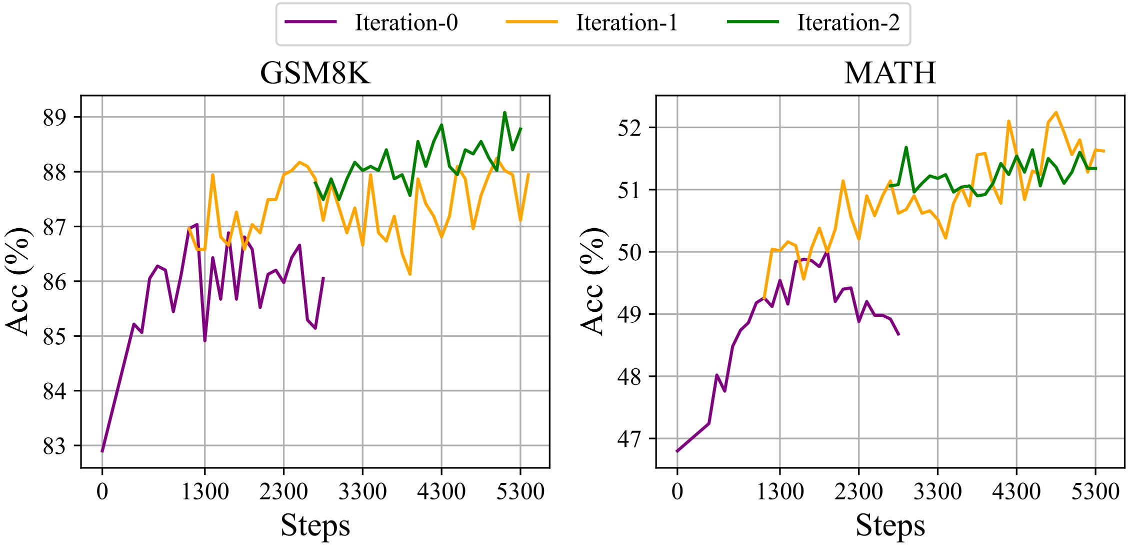

Figure 6 | Performance of iterative reinforcement learning with DeepSeekMath-Instruct 7B on two benchmarks.

而离线采样表示训练数据来自初始 SFT 模型的采样结果。RFT 和 DPO 遵循离线风格，而 Online RFT 和 GRPO 遵循在线风格。

As shown in Figure 5, we find that the Online RFT significantly outperforms RFT on two benchmarks. Specifically, Online RFT is comparable to RFT in the early stage of training but gains an absolute advantage in the later stage, demonstrating the superiority of online training. This is intuitive, as in the initial stage, the actor and the SFT model exhibit close resemblance, with the sampled data revealing only minor differences. In the later stage, however, the data sampled from the actor will exhibit more significant differences, and real-time data sampling will offer greater advantages.
如图 5 所示，我们发现 Online RFT 在两个基准测试中的表现都显著优于 RFT。具体来说，Online RFT 在训练初期与 RFT 相当，但在后期占据绝对优势，体现了在线训练的优越性。这是很直观的，因为在初始阶段，参与者和 SFT 模型表现出非常相似的特征，采样数据只显示出很小的差异。然而，在后期，从参与者采样的数据会表现出更明显的差异，实时数据采样将具有更大的优势。

Observation about Gradient Coefficient The algorithm processes the reward signal to the gradient coefficient to update the model parameter. We divide the reward function as ‘Rule’ and ‘Model’ in our experiments. Rule refers to judging the quality of a response based on the correctness of the answer, and Model denotes that we train a reward model to score each response. The training data of the reward model is based on the rule judgment. Equations 10 and 21 highlight a key difference between GRPO and Online RFT: GRPO uniquely adjusts its gradient coefficient based on the reward value provided by the reward model. This allows for differential reinforcement and penalization of responses according to their varying magnitudes. In contrast, Online RFT lacks this feature; it does not penalize incorrect responses and uniformly reinforces all responses with correct answers at the same level of intensity.
算法将奖励信号处理为梯度系数，以更新模型参数。我们在实验中将奖励函数分为 ‘规则’ 和 ‘模型’。规则是指根据答案的正确性判断反应的质量，模型表示我们训练一个奖励模型来对每个反应进行评分。奖励模型的训练数据基于规则判断。方程式10和21突出了 GRPO 和 Online RFT 之间的一个关键区别：GRPO 根据奖励模型提供的奖励值独特地调整其梯度系数。这允许根据反应的不同幅度进行差异化强化和惩罚。相比之下，Online RFT 缺乏这一特性；它不会惩罚错误的反应，而是以相同的强度均匀地强化所有正确答案的反应。

As demonstrated in Figure 5, GRPO surpasses online RFT, thereby highlighting the efficiency of altering positive and negative gradient coefficients. In addition, GRPO+PS shows superior performance compared to GRPO+OS, indicating the benefits of using fine-grained, step-aware gradient coefficients. Furthermore, we explore the iterative RL, in our experiments, we conduct two rounds of iteration. As shown in Figure 6, we notice that the iterative RL significantly improves the performance, especially at the first iteration.
如图 5 所示，GRPO 超越了在线 RFT，从而凸显了改变正负梯度系数的效率。此外，GRPO+PS 的性能优于 GRPO+OS，这表明使用细粒度、步进感知梯度系数的好处。此外，我们探索了迭代 RL，在我们的实验中，我们进行了两轮迭代。如图 6 所示，我们注意到迭代 RL 显著提高了性能，尤其是在第一次迭代时。

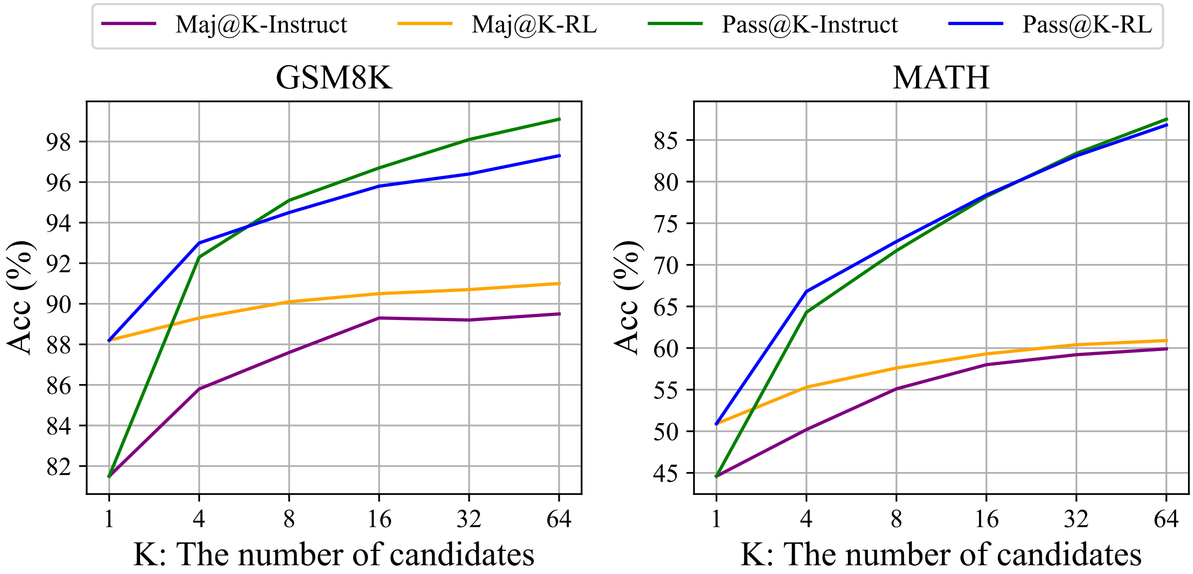

Figure 7 | The Maj@K and Pass@K of SFT and RL DeepSeekMath 7B on GSM8K and MATH (temperature 0.7). It was noted that RL enhances Maj@K but not Pass@K.
SFT 和 RL DeepSeekMath 7B 在 GSM8K 和 MATH（温度 0.7）上的 Maj@K 和 Pass@K。值得注意的是，RL 增强了 Maj@K，但没有增强 Pass@K。

5.2.2. Why RL Works?

In this paper, we conduct reinforcement learning based on a subset of instruction tuning data, and it achieves significant performance enhancement upon the instruction tuning model. To further explain why reinforcement learning works. We evaluate the Pass@K and Maj@K accuracy of the Instruct and RL models on two benchmarks. As shown in Figure 7, RL enhances Maj@K’ s performance but not Pass@K. These findings indicate that RL enhances the model’s overall performance by rendering the output distribution more robust, in other words, it seems that the improvement is attributed to boosting the correct response from TopK rather than the enhancement of fundamental capabilities. Similarly, (47) identified a misalignment problem in reasoning tasks within the SFT model, showing that the reasoning performance of SFT models can be improved through a series of preference alignment strategies (42; 47; 58).
本文基于指令调优数据子集进行强化学习，在指令调优模型上取得了显著的性能提升。为了进一步解释强化学习的工作原理，我们在两个基准上评估了 Instruct 和 RL 模型的 Pass@K 和 Maj@K 准确率。如图 7 所示，RL 提高了 Maj@K 的性能，但没有提高 Pass@K 的性能。这些发现表明，RL 通过使输出分布更加稳健来提高模型的整体性能，换句话说，**似乎这种改进归因于增加了 TopK 的正确响应，而不是基本能力的增强**。同样，(47) 在 SFT 模型的推理任务中发现了 **错位问题**，表明可以通过一系列偏好对齐策略来提高 SFT 模型的推理性能 (42; 47; 58)。

5.2.3. How to Achieve More Effective RL?

We demonstrate RL works pretty well in mathematical reasoning tasks. We also provide a unified paradigm to understand different representative training methods. Within this paradigm, all methods are conceptualized as either direct or simplified RL techniques. As summarized in Equation 5, there exist three key components: Data Source, Algorithm, and Reward Function. We provide some potential future directions about the three components.
我们证明了强化学习在数学推理任务中表现良好。我们还提供了一个统一的范式来理解不同的代表性训练方法。在这个范式中，所有方法都被概念化为直接或简化的强化学习技术。如公式 5 中总结的那样，存在三个关键组成部分：数据源、算法和奖励函数。我们提供了有关这三个组成部分的一些潜在未来方向。

Data Source   Data source is the raw material of all training methods. In the context of RL, we specifically refer to the data source as the unlabeled questions with the outputs sampled from the policy model. In this paper, we only use the questions from the instruction tuning stage and a naive nucleus sampling to sample outputs. We think this is a potential reason that our RL pipeline only improves the Maj@K performance. In the future, we will explore our RL pipeline on out-of-distribution question prompts, in conjunction with advanced sampling (decoding) strategies, like those based on tree-search methods (55). Also, the efficient inference techniques (23; 24; 53; 54), which determines the exploration efficiency of policy models, also play an exceedingly important role.
数据源是所有训练方法的原材料。在强化学习的背景下，我们特指数据源为未 token 的问题，其输出从策略模型中采样。在本文中，我们仅使用来自指令调整阶段的问题和简单的核心采样来采样输出。我们认为这是我们的强化学习管道仅提高 Maj@K 性能的潜在原因。未来，我们将结合 **高级采样（解码）策略**（例如基于树搜索方法 (55) 的策略），探索我们在分布外问题提示上的强化学习管道。此外，决定策略模型探索效率的 **高效推理技术** (23; 24; 53; 54) 也发挥着极其重要的作用。

Algorithms   Algorithms process the data and reward signal to the gradient coefficient to update the model parameter. Based on Equation 5, to some extent, all methods now fully TRUST the signal of the reward function to increase or decrease the conditional probability of a certain token. However, it is impossible to ensure the reward signal is always reliable, especially in extremely complex tasks. For example, even the PRM800K datasets (27), which have been carefully annotated by well-trained annotators, still contain approximately 20% of incorrectly annotations[7]. To this end, we will explore the reinforcement learning algorithm that is robust against noisy reward signals. We believe such WEAK-TO-STRONG (5) alignment methods will bring a fundamental change to the learning algorithms.
算法将数据和奖励信号处理到梯度系数中，以更新模型参数。基于方程5，在某种程度上，所有方法现在都完全**信任**奖励函数的信号，以增加或减少某个 token 的条件概率。然而，不可能确保奖励信号始终可靠，特别是在极其复杂的任务中。例如，即使是经过训练有素的注释者精心注释的 PRM800K 数据集 (27)，仍然包含大约 20% 的错误注释 [7]。为此，我们将探索对嘈杂奖励信号具有鲁棒性的强化学习算法。我们相信这种 WEAK-TO-STRONG (5) 对齐方法将给学习算法带来根本性的改变。

Reward Function   Reward function is the source of the training signal. In RL, the reward function is usually the neural reward model. We think there exist three important directions for reward models: 1) How to enhance the generalization ability of the reward model. The reward model must be effectively generalized to handle out-of-distribution questions and advanced decoding outputs; otherwise, reinforcement learning may merely stabilize the distribution of LLMs rather than improve their fundamental capabilities; 2) How to reflect the uncertainty of reward model. The uncertainty could potentially act as a linking bridge between the weak reward model and the weak-to-strong learning algorithms; 3) How to efficiently build high-quality process reward models that can provide fine-grained training signals for the reasoning process (27; 48).
奖励函数是训练信号的来源。在强化学习中，奖励函数通常是神经奖励模型。我们认为奖励模型有三个重要方向：1）**如何增强奖励模型的泛化能力**。奖励模型必须能够有效地泛化以处理分布外的问题和高级解码输出；否则，强化学习可能仅仅稳定了 LLM 的分布，而不是提高其基本能力；2）**如何反映奖励模型的不确定性**。不确定性可能成为弱奖励模型和弱到强学习算法之间的桥梁；3）**如何高效构建高质量的过程奖励模型**，为推理过程提供细粒度的训练信号 (27; 48)。

[7] https://github.com/openai/prm800k/issues/12#issuecomment-1728491852

6. Conclusion, Limitation, and Future Work

We present DeepSeekMath, which outperforms all open-source models on the competition-level MATH benchmark and approaches the performance of closed models. DeepSeekMath is initialized with DeepSeek-Coder-v1.5 7B and undergoes continual training for 500B tokens, with a significant component of the training data being 120B math tokens sourced from Common Crawl. Our extensive ablation study shows web pages offer significant potential for high-quality mathematical data, while arXiv may not as beneficial as we expected. We introduce Group Relative Policy Optimization (GRPO), a variant of Proximal Policy Optimization (PPO), which can notably improve mathematical reasoning capabilities with less memory consumption. The experiment results show that GRPO is effective even if DeepSeekMath-Instruct 7B has reached a high score on benchmarks. We also provide a unified paradigm to understand a series of methods and summarize several potential directions for more effective reinforcement learning.
我们提出了 DeepSeekMath，它在竞赛级 MATH 基准上的表现优于所有开源模型，并且接近封闭模型的性能。DeepSeekMath 使用 DeepSeek-Coder-v1.5 7B 初始化，并持续训练 500B 个 token，其中训练数据的很大一部分是来自 Common Crawl 的 120B 个数学 token。我们广泛的消融研究表明，网页为高质量数学数据提供了巨大的潜力，而 arXiv 可能没有我们预期的那么有益。我们引入了组相对策略优化 (GRPO)，它是近端策略优化 (PPO) 的一种变体，它可以显着提高数学推理能力，同时减少内存消耗。实验结果表明，即使 DeepSeekMath-Instruct 7B 在基准测试中取得了高分，GRPO 仍然有效。我们还提供了一个统一的范式来理解一系列方法，并总结了更有效的强化学习的几个潜在方向。

Although DeepSeekMath achieves impressive scores on quantitative reasoning benchmarks, its capability on geometry and theorem-proof are relatively weaker than closed models. For instance, in our dry run, the model cannot handle problems related to triangles and ellipses, which may indicate data selection bias in pre-training and fine-tuning. In addition, restricted by the model scale, DeepSeekMath is worse than GPT-4 on few-shot capability. GPT-4 could improve its performance with few-shot inputs, while DeepSeekMath shows similar performance in zero-shot and few-shot evaluation. In the future, we will further improve our engineered data selection pipeline to construct more high-quality pre-trained corpus. In addition, we will explore the potential directions (Section 5.2.3) for more effective reinforcement learning of LLMs.
虽然 DeepSeekMath 在定量推理基准上取得了令人印象深刻的成绩，但其在几何和定理证明方面的能力与封闭模型相比相对较弱。例如，在我们的试运行中，该模型无法处理与三角形和椭圆相关的问题，这可能表明在预训练和微调中存在数据选择偏差。此外，受模型规模的限制，DeepSeekMath 在少样本能力上不如 GPT-4。GPT-4 可以通过少样本输入提高其性能，而 DeepSeekMath 在零样本和少样本评估中表现出相似的性能。未来，我们将进一步改进我们工程化的数据选择流程，以构建更多高质量的预训练语料库。此外，我们将探索更有效的 LLM 强化学习的潜在方向（第 5.2.3 节）。

References

[1] R. Anil, S. Borgeaud, Y. Wu, J. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, D. Silver, S. Petrov, M. Johnson, I. Antonoglou, J. Schrittwieser, A. Glaese, J. Chen, E. Pitler, T. P. Lillicrap, A. Lazaridou, O. Firat, J. Molloy, M. Isard, P. R. Barham, T. Hennigan, B. Lee, F. Viola, M. Reynolds, Y. Xu, R. Doherty, E. Collins, C. Meyer, E. Rutherford, E. Moreira, K. Ayoub, M. Goel, G. Tucker, E. Piqueras, M. Krikun, I. Barr, N. Savinov, I. Danihelka, B. Roelofs, A. White, A. Andreassen, T. von Glehn, L. Yagati, M. Kazemi, L. Gonzalez, M. Khalman, J. Sygnowski, and et al. Gemini: A family of highly capable multimodal models. CoRR, abs/2312.11805, 2023. doi: 10.48550/ARXIV.2312.11805. URL https://doi.org/10.48550/arXiv.2312.11805.

[2] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, et al. Program synthesis with large language models. arXiv preprint arXiv:2108.07732, 2021.

[3] Z. Azerbayev, H. Schoelkopf, K. Paster, M. D. Santos, S. McAleer, A. Q. Jiang, J. Deng, S. Biderman, and S. Welleck. Llemma: An open language model for mathematics. arXiv preprint arXiv:2310.10631, 2023.

[4] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

[5] C. Burns, P. Izmailov, J. H. Kirchner, B. Baker, L. Gao, L. Aschenbrenner, Y. Chen, A. Ecoffet, M. Joglekar, J. Leike, et al. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. arXiv preprint arXiv:2312.09390, 2023.

[6] ChatGLM3 Team. Chatglm3 series: Open bilingual chat llms, 2023. URL https://github.com/THUDM/ChatGLM3.

[7] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. CoRR, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

[8] W. Chen, X. Ma, X. Wang, and W. W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. CoRR, abs/2211.12588, 2022. doi: 10.48550/ARXIV.2211.12588. URL https://doi.org/10.48550/arXiv.2211.12588.

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

[10] T. Computer. Redpajama: an open dataset for training large language models, Oct. 2023. URL https://github.com/togethercomputer/RedPajama-Data.

[11] DeepSeek-AI. Deepseek LLM: scaling open-source language models with longtermism. CoRR, abs/2401.02954, 2024. doi: 10.48550/ARXIV.2401.02954. URL https://doi.org/10.48550/arXiv.2401.02954.

[12] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang. Glm: General language model pretraining with autoregressive blank infilling. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320–335, 2022.

[13] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig. PAL: program-aided language models. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA, volume 202 of Proceedings of Machine Learning Research, pages 10764–10799. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

[14] Z. Gou, Z. Shao, Y. Gong, Y. Shen, Y. Yang, M. Huang, N. Duan, and W. Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. CoRR, abs/2309.17452, 2023. doi: 10.48550/ARXIV.2309.17452. URL https://doi.org/10.48550/arXiv.2309.17452.

[15] D. Guo, Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. K. Li, F. Luo, Y. Xiong, and W. Liang. Deepseek-coder: When the large language model meets programming – the rise of code intelligence, 2024.

[16] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300, 2020.

[17] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874, 2021.

[18] High-flyer. Hai-llm: 高效且轻量的大模型训练工具, 2023. URL https://www.high-flyer.cn/en/blog/hai-llm.

[19] Inflection AI. Inflection-2, 2023. URL https://inflection.ai/inflection-2.

[20] A. Q. Jiang, S. Welleck, J. P. Zhou, W. Li, J. Liu, M. Jamnik, T. Lacroix, Y. Wu, and G. Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. arXiv preprint arXiv:2210.12283, 2022.

[21] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

[22] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext. zip: Compressing text classification models. arXiv preprint arXiv:1612.03651, 2016.

[23] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[24] Y. Leviathan, M. Kalman, and Y. Matias. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, pages 19274–19286. PMLR, 2023.

[25] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. Advances in Neural Information Processing Systems, 35:3843–3857, 2022.

[26] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, and V. Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

[27] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023.

[28] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

[29] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. arXiv preprint arXiv:2308.09583, 2023.

[30] S. Mishra, M. Finlayson, P. Lu, L. Tang, S. Welleck, C. Baral, T. Rajpurohit, O. Tafjord, A. Sabharwal, P. Clark, and A. Kalyan. LILA: A unified benchmark for mathematical reasoning. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022, pages 5807–5832. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.392. URL https://doi.org/10.18653/v1/2022.emnlp-main.392.

[31] X. Nguyen, W. Zhang, X. Li, M. M. Aljunied, Q. Tan, L. Cheng, G. Chen, Y. Deng, S. Yang, C. Liu, H. Zhang, and L. Bing. Seallms - large language models for southeast asia. CoRR, abs/2312.00738, 2023. doi: 10.48550/ARXIV.2312.00738. URL https://doi.org/10.48550/arXiv.2312.00738.

[32] OpenAI. GPT4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[33] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35:27730–27744, 2022.

[34] K. Paster, M. D. Santos, Z. Azerbayev, and J. Ba. Openwebmath: An open dataset of high-quality mathematical web text. CoRR, abs/2310.06786, 2023. doi: 10.48550/ARXIV.2310.06786. URL https://doi.org/10.48550/arXiv.2310.06786.

[35] L. C. Paulson. Three years of experience with sledgehammer, a practical link between automatic and interactive theorem provers. In R. A. Schmidt, S. Schulz, and B. Konev, editors, Proceedings of the 2nd Workshop on Practical Aspects of Automated Reasoning, PAAR-2010, Edinburgh, Scotland, UK, July 14, 2010, volume 9 of EPiC Series in Computing, pages 1–10. EasyChair, 2010. doi: 10.29007/TNFD. URL https://doi.org/10.29007/tnfd.

[36] S. Polu and I. Sutskever. Generative language modeling for automated theorem proving. CoRR, abs/2009.03393, 2020. URL https://arxiv.org/abs/2009.03393.

[37] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. 2023.

[38] J. Schulman. Approximating kl divergence, 2020. URL http://joschu.net/blog/kl-approx.html.

[39] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438, 2015.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[41] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei. Language models are multilingual chain-of-thought reasoners. In The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net, 2023. URL https://openreview.net/pdf?id=fR3wGCk-IXp.

[42] F. Song, B. Yu, M. Li, H. Yu, F. Huang, Y. Li, and H. Wang. Preference ranking optimization for human alignment. arXiv preprint arXiv:2306.17492, 2023.

[43] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. arXiv preprint arXiv:2210.09261, 2022.

[44] T. Tao. Embracing change and resetting expectations, 2023. URL https://unlocked.microsoft.com/ai-anthology/terence-tao/.

[45] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. CoRR, abs/2307.09288, 2023. doi: 10.48550/arXiv.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

[46] T. H. Trinh, Y. Wu, Q. V. Le, H. He, and T. Luong. Solving olympiad geometry without human demonstrations. Nature, 625(7995):476–482, 2024.

[47] P. Wang, L. Li, L. Chen, F. Song, B. Lin, Y. Cao, T. Liu, and Z. Sui. Making large language models better reasoners with alignment. arXiv preprint arXiv:2309.02144, 2023.

[48] P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. CoRR, abs/2312.08935, 2023.

[49] Z. Wang, R. Xia, and P. Liu. Generative AI for math: Part I - mathpile: A billion-token-scale pretraining corpus for math. CoRR, abs/2312.17120, 2023. doi: 10.48550/ARXIV.2312.17120. URL https://doi.org/10.48550/arXiv.2312.17120.

[50] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[51] T. Wei, J. Luan, W. Liu, S. Dong, and B. Wang. Cmath: Can your language model pass chinese elementary school math test?, 2023.

[52] M. Wenzel, L. C. Paulson, and T. Nipkow. The isabelle framework. In O. A. Mohamed, C. A. Muñoz, and S. Tahar, editors, Theorem Proving in Higher Order Logics, 21st International Conference, TPHOLs 2008, Montreal, Canada, August 18-21, 2008. Proceedings, volume 5170 of Lecture Notes in Computer Science, pages 33–38. Springer, 2008. doi: 10.1007/978-3-540-71067-7_7. URL https://doi.org/10.1007/978-3-540-71067-7_7.

[53] H. Xia, T. Ge, P. Wang, S.-Q. Chen, F. Wei, and Z. Sui. Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation. In H. Bouamor, J. Pino, and K. Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 3909–3925, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.257. URL https://aclanthology.org/2023.findings-emnlp.257.

[54] H. Xia, Z. Yang, Q. Dong, P. Wang, Y. Li, T. Ge, T. Liu, W. Li, and Z. Sui. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. arXiv preprint arXiv:2401.07851, 2024.

[55] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601, 2023.

[56] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. CoRR, abs/2309.12284, 2023. doi: 10.48550/ARXIV.2309.12284. URL https://doi.org/10.48550/arXiv.2309.12284.

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

[58] Z. Yuan, H. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. Rrhf: Rank responses to align language models with human feedback without tears. arXiv preprint arXiv:2304.05302, 2023.

[59] X. Yue, X. Qu, G. Zhang, Y. Fu, W. Huang, H. Sun, Y. Su, and W. Chen. Mammoth: Building math generalist models through hybrid instruction tuning. CoRR, abs/2309.05653, 2023. doi: 10.48550/ARXIV.2309.05653. URL https://doi.org/10.48550/arXiv.2309.05653.

[60] K. Zheng, J. M. Han, and S. Polu. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. arXiv preprint arXiv:2109.00110, 2021.

[61] W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, and N. Duan. AGIEval: A human-centric benchmark for evaluating foundation models. CoRR, abs/2304.06364, 2023. doi: 10.48550/arXiv.2304.06364. URL https://doi.org/10.48550/arXiv.2304.06364.

A. Appendix

A.1. Analysis of Reinforcement Learning

We provide the detailed derivation of the data source and gradient coefficient (algorithm and reward function) across various methods, including SFT, RFT, Online RFT, DPO, PPO, and GRPO.
我们提供了跨各种方法的数据源和梯度系数（算法和奖励函数）的详细推导，包括 SFT、RFT、Online RFT、DPO、PPO 和 GRPO。

A.1.1. Supervised Fine-tuning

The objective of Supervised Fine-tuning is maximizing the following objective
监督微调的目标是最大化以下目标：

$$\mathcal{J}_{SFT}(\theta) = \mathbb{E}[q, o \sim P_{sft}(Q, O)] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \log \pi_\theta(o_t | q, o_{<t}) \right). \tag{6}$$

The gradient of $\mathcal{J}_{SFT}(\theta)$ is:

$$\nabla_\theta \mathcal{J}_{SFT} = \mathbb{E}[q, o \sim P_{sft}(Q, O)] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \nabla_\theta \log \pi_\theta(o_t | q, o_{<t}) \right). \tag{7}$$

Data Source: The dataset employed for SFT. Reward Function: This can be regarded as human selection. Gradient Coefficient: always set to 1.
数据来源：SFT 使用的数据集。奖励函数：这可以视为人为选择。梯度系数：始终设置为 1。

A.1.2. Rejection Sampling Fine-tuning

Rejection Sampling Fine-tuning first samples multiple outputs from the supervised fine-tuned LLMs for each question, and then trains LLMs on the sampled outputs with the correct answer. Formally, the objective of RFT is to maximize the following objectives
拒绝抽样微调首先针对每个问题从监督微调 LLM 中抽取多个输出，然后使用正确答案在抽样输出上训练 LLM。正式来说，RFT 的目标是最大化以下目标：

$$\mathcal{J}_{RFT}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_{sft}(O|q)] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \mathbb{I}(o) \log \pi_\theta(o_t | q, o_{<t}) \right). \tag{8}$$

The gradient of $\mathcal{J}_{RFT}(\theta)$ is:

$$\nabla_\theta \mathcal{J}_{RFT}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_{sft}(O|q)] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \mathbb{I}(o) \nabla_\theta \log \pi_\theta(o_t | q, o_{<t}) \right). \tag{9}$$

Data Source: question in SFT dataset with outputs sampled from SFT model. Reward Function: Rule (whether the answer is correct or not). Gradient Coefficient:

$$GC_{RFT}(q, o, t) = \mathbb{I}(o) = \begin{cases} 1 & \text{the answer of o is correct} \\ 0 & \text{the answer of o is incorrect} \end{cases} \tag{10}$$

A.1.3. Online Rejection Sampling Fine-tuning

The only difference between RFT and Online RFT is that the outputs of Online RFT are sampled from the real-time policy model π_θ, rather than from the SFT model $\pi_{\theta_{sft}}$. Therefore, the gradient of online RFT is
RFT 和 Online RFT 唯一的区别在于，Online RFT 的输出是从实时策略模型 π_θ 中采样的，而不是从 SFT 模型 $\pi_{\theta_{sft}}$ 中采样的。因此，Online RFT 的梯度为：

$$\nabla_\theta \mathcal{J}_{OnRFT}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_\theta(O|q)] \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \mathbb{I}(o) \nabla_\theta \log \pi_\theta(o_t | q, o_{<t}) \right). \tag{11}$$

A.1.4. Direct Preference Optimization (DPO)

The objective of DPO is:

$$\mathcal{J}_{DPO}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o^+, o^- \sim \pi_{sft}(O|q)] \log \sigma \left(\beta \frac{1}{|o^+|} \sum_{t=1}^{|o^+|} \log \frac{\pi_\theta(o_t^+|q, o_{<t}^+)}{\pi_{\text{ref}}(o_t^+|q, o_{<t}^+)} - \beta \frac{1}{|o^-|} \sum_{t=1}^{|o^-|} \log \frac{\pi_\theta(o_{<t}^-|q, o_{<t}^-)}{\pi_{\text{ref}}(o_{<t}^-|q, o_{<t}^-)} \right) \tag{12}$$

The gradient of $\mathcal{J}_{DPO}(\theta)$ is:

$$\begin{aligned} \nabla_\theta \mathcal{J}_{DPO}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o^+, o^- \sim \pi_{sft}(O|q)] & \left(\frac{1}{|o^+|} \sum_{t=1}^{|o^+|} GC_{DPO}(q, o, t) \nabla_\theta \log \pi_\theta(o_t^+|q, o_{<t}^+) \right. \\ & \left. - \frac{1}{|o^-|} \sum_{t=1}^{|o^-|} GC_{DPO}(q, o, t) \nabla_\theta \log \pi_\theta(o_t^-|q, o_{<t}^-) \right) \end{aligned} \tag{13}$$

Data Source: question in SFT dataset with outputs sampled from SFT model. Reward Function: human preference in the general domain (can be ‘Rule’ in mathematical tasks).
数据来源：SFT 数据集中的问题，其输出取自 SFT 模型。奖励函数：一般领域中的人类偏好（在数学任务中可以是“规则”）。

Gradient Coefficient 梯度系数：

$$GC_{DPO}(q, o, t) = \sigma \left(\beta \log \frac{\pi_\theta(o_t^-|q, o_{<t}^-)}{\pi_{\text{ref}}(o_t^-|q, o_{<t}^-)} - \beta \log \frac{\pi_\theta(o_t^+|q, o_{<t}^+)}{\pi_{\text{ref}}(o_t^+|q, o_{<t}^+)} \right) \tag{14}$$

A.1.5. Proximal Policy Optimization (PPO)

The objective of PPO is:

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})}, 1 - \varepsilon, 1 + \varepsilon \right) A_t \right]. \tag{15}$$

To simplify the analysis, it is assumed that the model only has a single update following each exploration stage, thereby ensuring that $\pi_{\theta_{old}} = \pi_\theta$. In this case, we can remove the $\min$ and clip operation:
为了简化分析，假设模型在每个探索阶段之后只进行一次更新，从而确保 $\pi_{\theta_{old}} = \pi_\theta$。在这种情况下，我们可以删除 $\min$ 和 clip 操作：

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \frac{\pi_\theta(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} A_t. \tag{16}$$

The gradient of $\mathcal{J}_{PPO}(\theta)$ is:

$$\nabla_\theta \mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P_{sft}(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} A_t \nabla_\theta \log \pi_\theta(o_t|q, o_{<t}) \tag{17}$$

Data Source: question in SFT dataset with outputs sampled from policy model. Reward Function: reward model. Gradient Coefficient:

$$GC_{PPO}(q, o, t, \pi_{\theta_{rm}}) = A_t, \tag{18}$$

where A_t is the advantage, which is computed by applying Generalized Advantage Estimation (GAE) (39), based on the rewards $\{r_{\geq t}\}$ and a learned value function V_ψ.
其中 A_t 是优势，它是通过应用广义优势估计 (GAE) (39) 计算得出的，基于奖励 $\{r_{\geq t}\}$ 和学习到的价值函数 V_ψ。

A.1.6. Group Relative Policy Optimization (GRPO)

The objective of GRPO is (assume $\pi_{\theta_{old}} = \pi_\theta$ for simplified analysis):

$$\begin{aligned} \mathcal{J}_{GRPO}(\theta) = \mathbb{E}&[q \sim P_{sft}(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left[\frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})} \hat{A}_{i,t} - \beta \left(\frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - \log \frac{\pi_{ref}(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} - 1 \right) \right]. \end{aligned} \tag{19}$$

The gradient of $\mathcal{J}_{GRPO}(\theta)$ is:

$$\begin{aligned} \nabla_\theta \mathcal{J}_{GRPO}(\theta) = \mathbb{E}&[q \sim P_{sft}(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \\ & \frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left[\hat{A}_{i,t} + \beta \left(\frac{\pi_{ref}(o_{i,t}|o_{i,<t})}{\pi_\theta(o_{i,t}|o_{i,<t})} - 1 \right) \right] \nabla_\theta \log \pi_\theta(o_{i,t}|q, o_{i,<t}). \end{aligned} \tag{20}$$

Data Source: question in SFT dataset with outputs sampled from policy model. Reward Function: reward model.
数据来源：SFT 数据集中的问题，其输出取自策略模型。奖励函数：奖励模型。Gradient Coefficient 梯度系数:

$$GC_{GRPO}(q,o,t,\pi_{\theta_{rm}}) = \hat{A}_{i,t} + \beta\left(\frac{\pi_{ref}(o_{i,t}|o_{i,<t})}{\pi_{\theta}(o_{i,t}|o_{i,<t})} - 1\right), \tag{21}$$

where $\hat{A}_{i,t}$ is computed based on the group reward scores.
其中 $\hat{A}_{i,t}$ 是根据群体奖励分数计算的。