

The GPT-Academic program cannot find abstract section in this paper.

O1 Replication Journey: A Strategic Progress Report -Part 1

Yiwei Qin^{1,4*} Xuefeng Li^{1,4*} Haoyang Zou^{4*} Yixiu Liu^{1,4*} Shijie Xia^{1,4*}

Zhen Huang⁴ Yixin Ye^{1,4} Weizhe Yuan² Hector Liu³ Yuanzhi Li³ Pengfei Liu^{1,4†}

¹ Shanghai Jiao Tong University, ² New York University,

³ MBZUAI, ⁴ Generative AI Research Lab (GAIR)

Abstract

This paper introduces a pioneering approach to artificial intelligence research, embodied in our O1 Replication Journey. In response to the announcement of OpenAI’s groundbreaking O1 model, we embark on a transparent, real-time exploration to replicate its capabilities while reimagining the process of conducting and communicating AI research. Our methodology addresses critical challenges in modern AI research, including the insularity of prolonged team-based projects, delayed information sharing, and the lack of recognition for diverse contributions. By providing comprehensive, real-time documentation of our replication efforts, including both successes and failures, we aim to foster open science, accelerate collective advancement, and lay the groundwork for AI-driven scientific discovery. Our research progress report diverges significantly from traditional research papers, offering continuous updates, full process transparency, and active community engagement throughout the research journey. Technologically, we proposed the “journey learning” paradigm, which encourages models to learn not just shortcuts, but the complete exploration process, including trial and error, reflection, and backtracking. With only 327 training samples and without any additional tricks, journey learning outperformed conventional supervised learning by over 8% on the MATH dataset, demonstrating its extremely powerful potential. We believe this to be the most crucial component of O1 technology that we have successfully decoded. We share valuable resources including technical hypotheses and insights, cognitive exploration maps, custom-developed tools, etc at <https://github.com/GAIR-NLP/O1-Journey>.

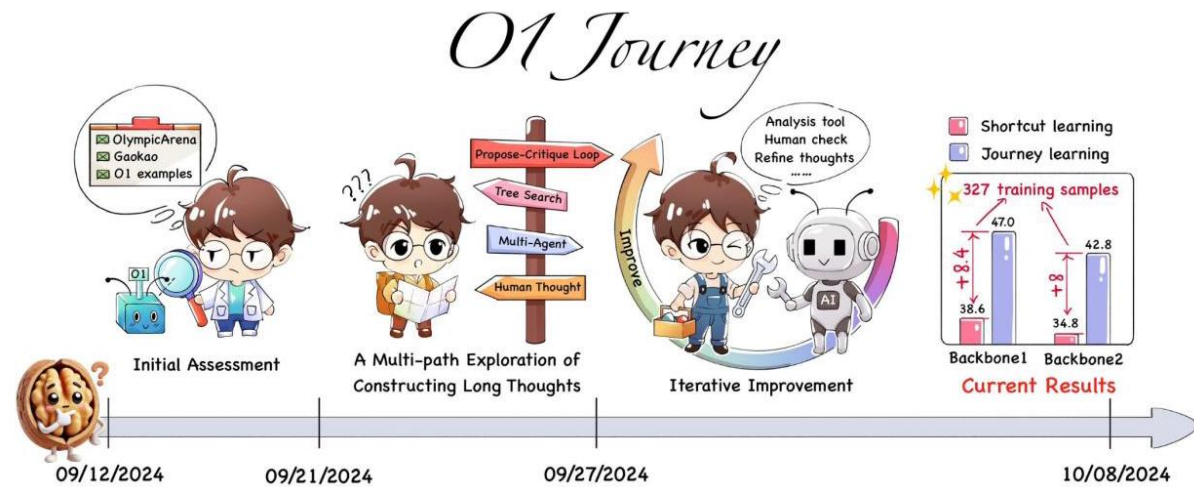


Figure 1: Illustration of our O1 replication journey from September 12 to October 8, 2024. It depicts four key stages: Initial Assessment, Multi-path Exploration, Iterative Improvement, and Current Results.

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

O1 Replication Journey: A Strategic Progress Report -Part 1

Yiwei Qin^{1,4*} Xuefeng Li^{1,4*} Haoyang Zou^{4*} Yixiu Liu^{1,4*} Shijie Xia^{1,4*}

Zhen Huang⁴ Yixin Ye^{1,4} Weizhe Yuan² Hector Liu³ Yuanzhi Li³ Pengfei Liu^{1,4†}

¹ 上海交通大学, ² 纽约大学,

³ MBZUAI, ⁴ 生成式人工智能研究实验室 (GAIR)

Abstract

本文介绍了一种开创性的人工智能研究方法，体现在我们的 O1 复制之旅中。针对 OpenAI 宣布的开创性 O1 模型，我们开始了透明、实时的探索，旨在复制其功能的同时，重新构想进行和交流 AI 研究的过程。我们的方法解决了现代 AI 研究中的关键挑战，包括长时间团队项目中的封闭性、信息共享的延迟以及对多样化贡献的认可不足。通过提供我们复制工作（包括成功和失败）的全面实时记录，我们旨在促进开放科学，加速集体进步，并为 AI 驱动的科学发现奠定基础。我们的研究进展报告与传统研究论文显著不同，提供持续更新、全过程透明以及研究过程中的积极社区参与。技术上，我们提出了“旅程学习”范式，鼓励模型不仅学习捷径，还要学习完整的探索过程，包括试错、反思和回溯。仅使用 327 个训练样本且没有任何额外技巧，旅程学习在 MATH 数据集上的表现比传统的监督学习高出超过 8%，展示了其极其强大的潜力。我们认为这是我们在 O1 技术中成功解码的最关键部分。我们在 <https://github.com/GAIR-NLP/O1-Journey> 分享了宝贵资源，包括技术假设和见解、认知探索地图、自定义开发的工具等。

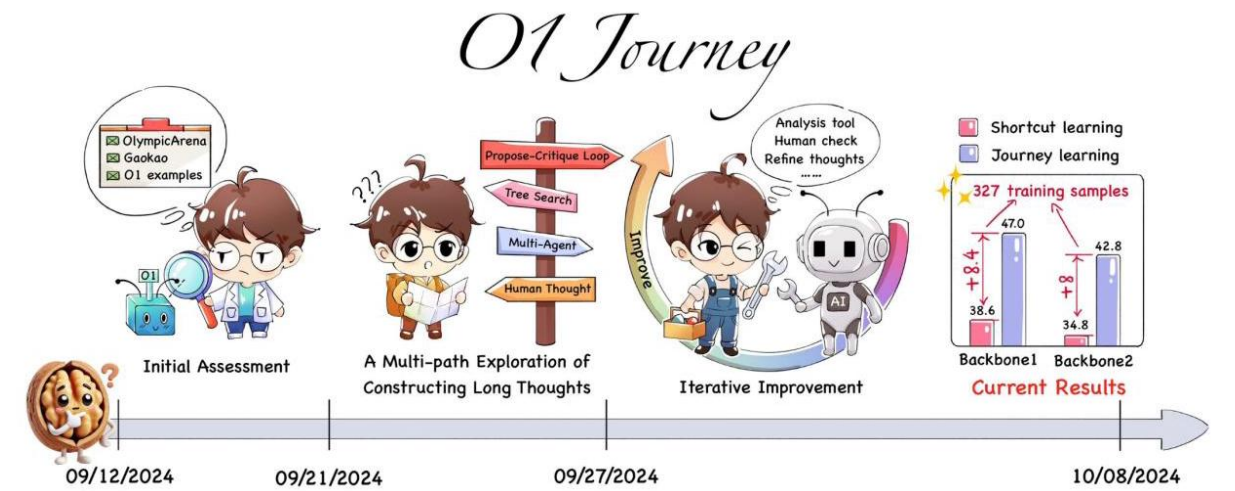
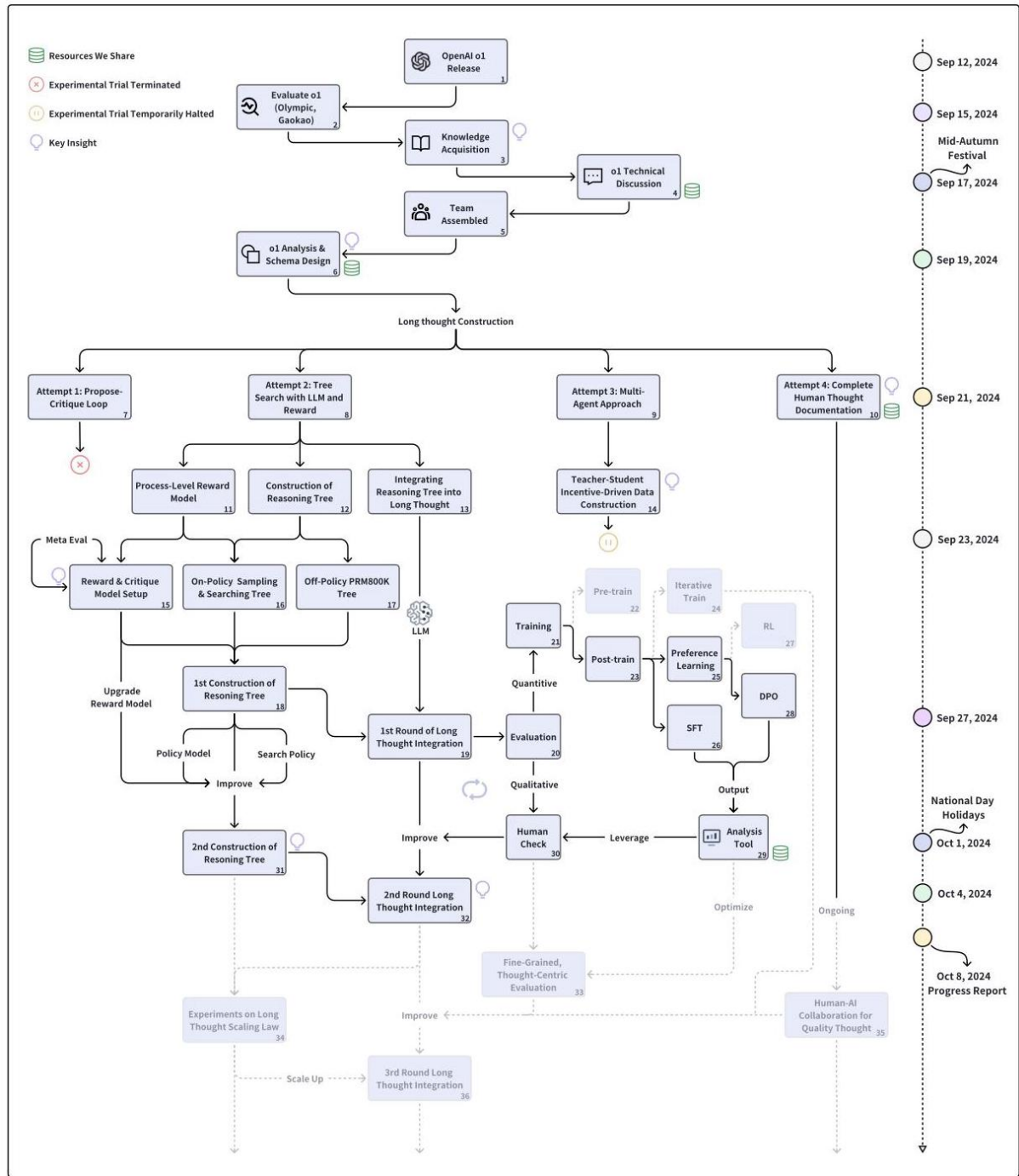


图 1: 2024 年 9 月 12 日至 10 月 8 日我们 O1 复制旅程的示意图。它描绘了四个关键阶段：初步评估、多路径探索、迭代改进和当前结果。这一旅程以我们新颖的“旅程学习”方法告终，该方法显著优于传统的“捷径学习”方法。在仅有 327 个训练样本的情况下，我们的旅程学习技术在 MATH500 (Lightman

The journey culminates in our novel “journey learning” approach, which significantly outperforms traditional “shortcut learning” methods. With only 327 training samples, our journey learning technique surpassed shortcut learning by 8.4% and 8.0% respectively on the MATH500 (Lightman et al., 2024). (c) denotes our Walnut Plan, which aims to revolutionize AI by developing systems capable of deep scientific thinking, ultimately enabling AI-driven breakthroughs in human knowledge and discovery.

1 Chronological Overview of the O1 Exploration Journey



* Co-first authors
* Corresponding author

等人，2024）上分别超过了捷径学习 8.4% 和 8.0%。（c）表示我们的核桃计划，该计划旨在通过开发能够进行深度科学思考的系统来革新 AI，最终实现 AI 驱动的人类知识和发现的突破。

1 Chronological Overview of the O1 Exploration Journey

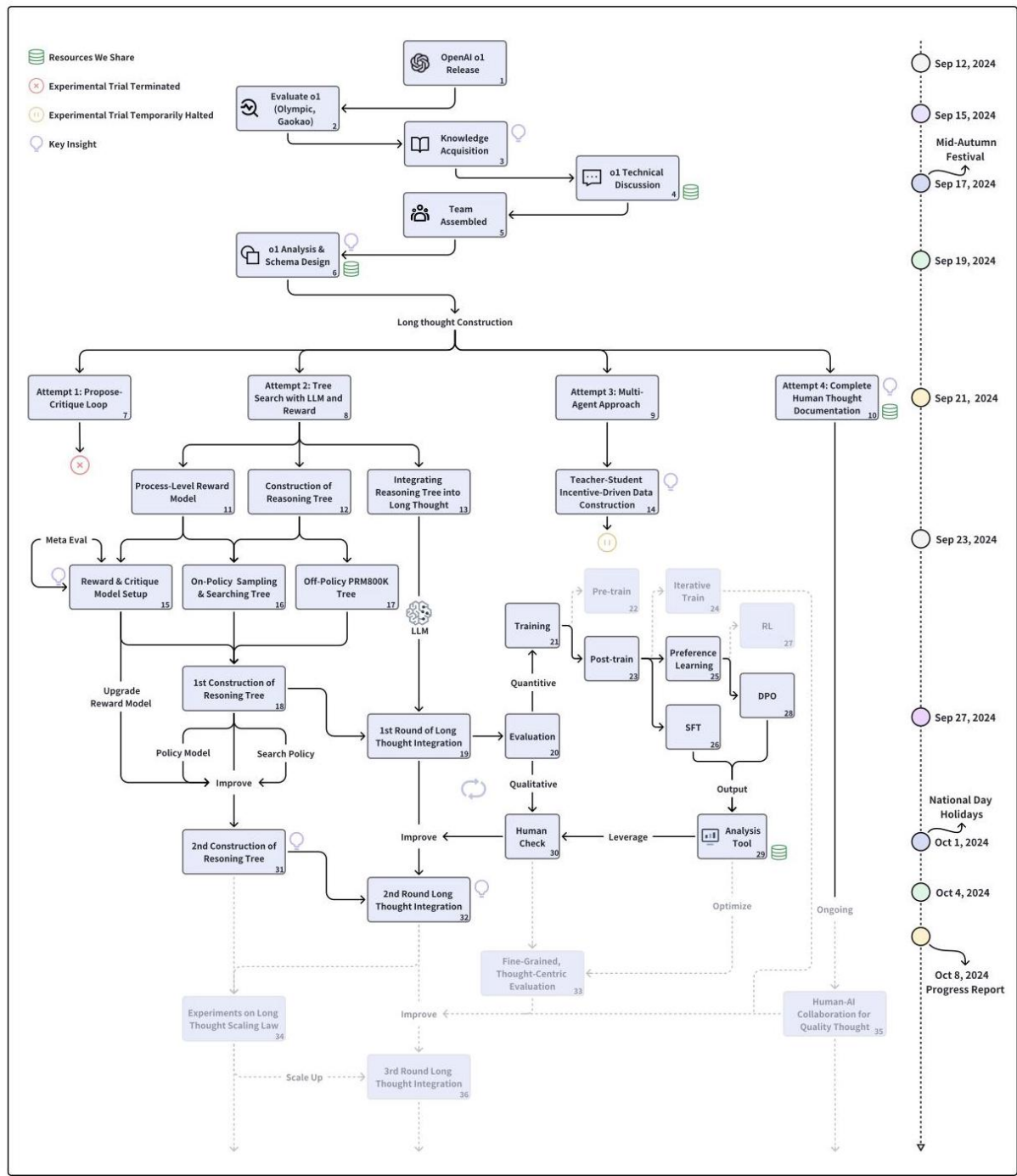


图 2：本图概述了我们从 2024 年 10 月 8 日开始探索 OpenAI 的 o1 技术的研究历程。时间线按时间

* 共同第一作者
* 通讯作者

Figure 2: This figure outlines our research journey exploring OpenAI’s o1 technology from its release through October 8, 2024. A timeline tracks our progress chronologically, with research activities flowing vertically in the main diagram. Following the o1 release, we progressed from initial evaluation and knowledge acquisition to team assembly and analysis. Our exploration then focused on four long thought construction attempts. The second attempt, our core exploration, splits into three tracks: Process-Level Reward Model, Construction of Reasoning Tree, and Integrating Reasoning Tree into Long Thought (detailed explanations of specific nodes can be found in Table 7). These converge in an iterative cycle of model improvement, including both quantitative and qualitative evaluation. The diagram’s right side illustrates our training pipeline, featuring pre-training, iterative training, and optimization techniques. Solid black elements represent completed paths and milestones, while gray dashed elements indicate planned future explorations. This visualization captures both our achievements and future research directions in o1 technology development.

2 Introduction

The landscape of artificial intelligence research has been dramatically altered by the announcement of OpenAI’s O1 model, a purportedly groundbreaking language model capable of complex reasoning tasks. Despite the excitement generated by this announcement, the AI community finds itself in a peculiar position: we know of O1’s existence and its claimed capabilities, but the details of its implementation, training data, and even its complete outputs remain shrouded in mystery. This lack of transparency not only hampers technological progress but also raises important questions about the open nature of scientific advancement in the AI field. It is within this context that our team embarked on the O1 Replication Journey. Our primary goal is not to achieve performance parity with OpenAI’s O1 - a task we acknowledge as extremely challenging given the limited information and resources available. Instead, our mission is to transparently document and share our exploration process, focusing on the fundamental questions we encounter, uncovering new scientific questions, and sharing our trial-and-error experiences with the broader AI community. By doing so, we aim to reduce the total collective cost of trial-and-error in the world and identify the key factors contributing to O1’s reported success.

This report’s structure marks a significant departure from traditional scientific publications, addressing key challenges in modern AI research. In an era of prolonged, team-based AI projects, we aim to combat information isolation and researcher burnout through enhanced transparency and real-time feedback. Additionally, this report represents a bold reimagining of AI research methodology. It aims not only to provide a valuable reference for current O1 replication efforts but also to establish a new paradigm for future AI research and broader scientific exploration. Through this innovative approach, we strive not only to achieve technological breakthroughs but also to actively shape a more open, collaborative, and responsible scientific culture. Simultaneously, we are accumulating invaluable learning materials for future AI systems capable of scientific discovery, thus laying the groundwork for the next generation of artificial intelligence in scientific research.

Our initial explorations have already yielded intriguing insights into the potential mechanisms behind O1’s reported capabilities. A key breakthrough in our research at the current stage is the proposed ”journey learning” paradigm, which represents a fundamental shift in how we approach model training. This innovative method encourages models to learn not just shortcuts to solutions, but the complete exploration process, including trial and error, reflection, and backtracking (see Figure 3). The power of this approach is evident in its performance: with only 327 training samples and without any additional

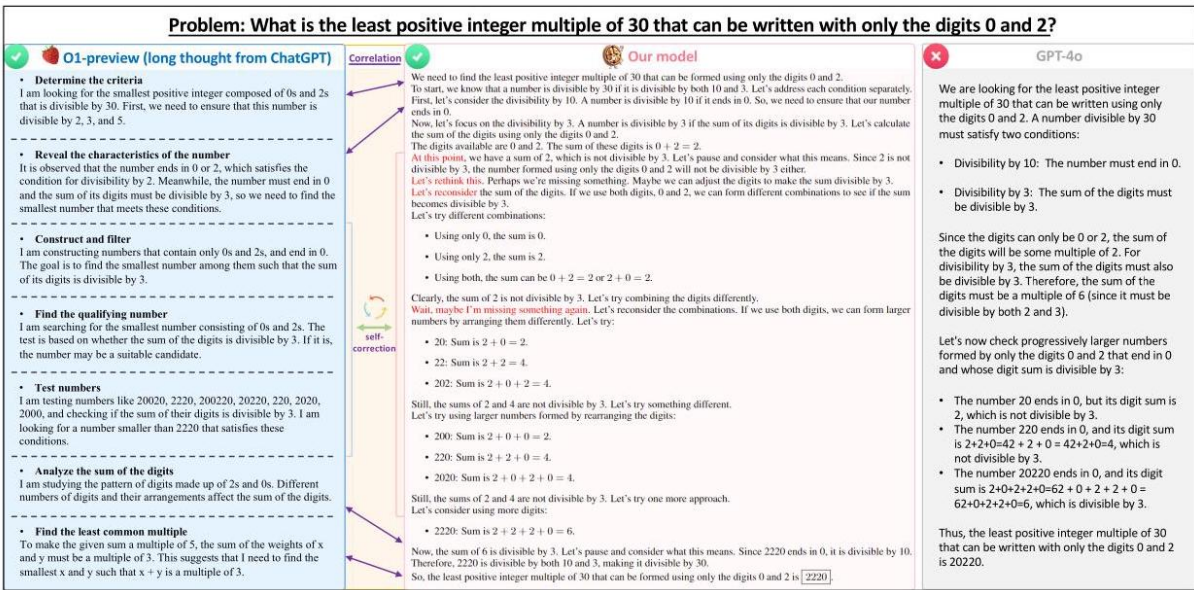
顺序追踪我们的进展，研究活动在主图中垂直流动。在 o1 发布后，我们从初步评估和知识获取进展到团队组建和分析。我们的探索随后集中在四次长时间思维构建尝试上。第二次尝试，即我们的核心探索，分为三个方向：过程级奖励模型、推理树的构建和将推理树整合到长时间思维中（具体节点的详细解释可在表 7 中找到）。这些方向在模型改进的迭代循环中汇聚，包括定量和定性评估。图的右侧展示了我们的训练管道，包括预训练、迭代训练和优化技术。实心黑色元素代表已完成的路径和里程碑，而灰色虚线元素表示计划中的未来探索。这一可视化既捕捉了我们在 o1 技术开发中的成就，也展示了未来的研究方向。

2 Introduction

人工智能研究的景观因 OpenAI 的 O1 模型的宣布而发生了巨大变化，该模型据称是一个能够完成复杂推理任务的突破性语言模型。尽管这一宣布引发了极大的兴奋，但 AI 社区发现自己处于一个特殊的位置：我们知道 O1 的存在及其声称的能力，但其实施细节、训练数据，甚至其完整的输出仍然笼罩在神秘之中。这种缺乏透明度不仅阻碍了技术进步，还引发了关于 AI 领域科学进步开放性质的重要问题。正是在这种背景下，我们的团队开始了 O1 复制之旅。我们的主要目标不是实现与 OpenAI 的 O1 性能的对等——鉴于可用的有限信息和资源，我们承认这是一个极其具有挑战性的任务。相反，我们的使命是透明地记录和分享我们的探索过程，专注于我们遇到的基本问题，揭示新的科学问题，并与更广泛的 AI 社区分享我们的试错经验。通过这样做，我们旨在减少世界上的总试错成本，并确定促成 O1 报告成功的關鍵因素。

本报告的结构与传统科学出版物有显著不同，旨在应对现代 AI 研究中的关键挑战。在长时间、团队合作的 AI 项目时代，我们希望通过增强透明度和实时反馈来对抗信息孤立和研究人员的倦怠。此外，本报告代表了对 AI 研究方法的大胆重新构想。它不仅旨在为当前的 O1 复制努力提供有价值的参考，还旨在为未来的 AI 研究和更广泛的科学探索建立新的范式。通过这种创新方法，我们不仅力求实现技术突破，还积极塑造更加开放、合作和负责任的科学文化。同时，我们正在为未来能够进行科学发现的 AI 系统积累宝贵的教育资源，从而为下一代人工智能在科学研究中的应用奠定基础。

我们的初步探索已经产生了关于 O1 报告能力背后潜在机制的有趣见解。目前阶段我们研究中的一个关键突破是提出的“旅程学习”范式，这代表了我们在模型训练方法上的根本转变。这种创新方法鼓励模型不仅学习解决问题的捷径，还包括完整的探索过程，包括试错、反思和回溯（见图 3）。这种方法的力量在其性能中显而易见：仅使用 327 个训练样本且没有任何额外技巧，旅程学习在 MATH 数据集上的表现比传统的监督学习高出超过 8%。我们相信这是迄今为止我们成功解码的 O1 技术中最关键的组成部分。



tricks, journey learning outperformed conventional supervised learning by over 8% on the MATH dataset, demonstrating its extremely powerful potential. We believe this to be the most crucial component of O1 technology that we have successfully decoded so far.

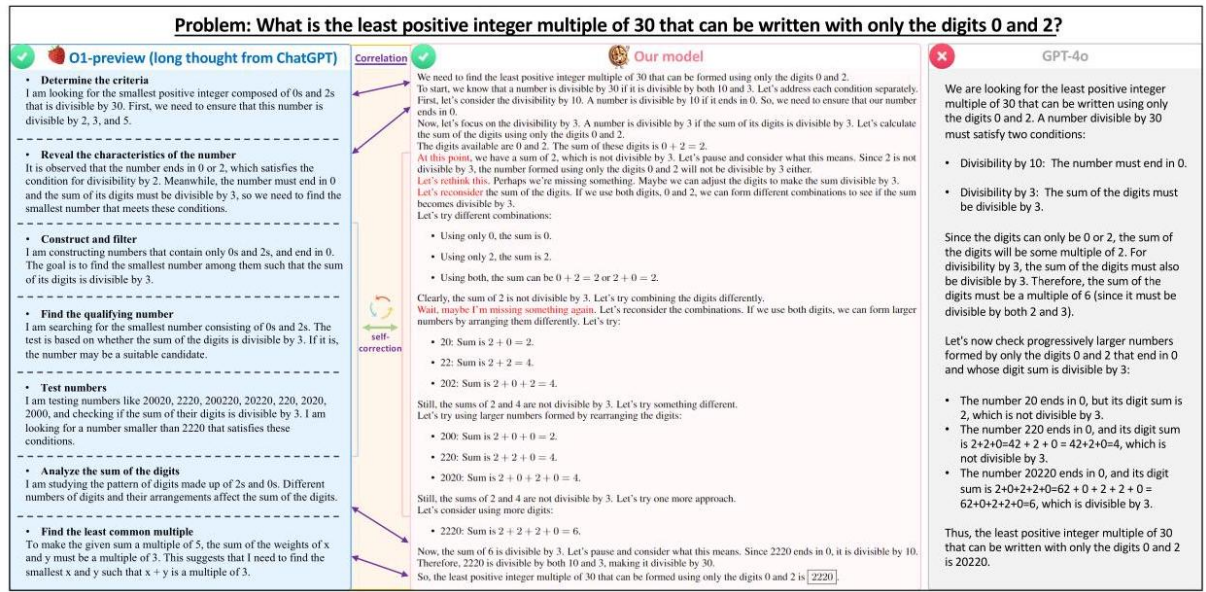


Figure 3: A case study comparing our model with OpenAI O1-preview and GPT-4o in solving math problems.

Through this journey, we anticipate a multifaceted impact on the field of AI research and development: (1) we expect to gain deeper insights into the fundamental principles underlying advanced language models, potentially uncovering key mechanisms that contribute to O1’s reported capabilities. (2) Moreover, by championing transparency and real-time sharing of our findings, we aim to foster a more open and collaborative AI research ecosystem, encouraging knowledge exchange and collective problem-solving. (3) Finally, by meticulously documenting our entire journey, including both successes and failures, we will create an invaluable dataset for training future AI systems in scientific discovery, laying the groundwork for the next generation of AI-driven research methodologies.

Aspects		Traditional Research Paper	Proposed Progress Report
Research Process	Timing of Publication	Published after research completion	Real-time updates throughout the research process
	Length of Research	Suitable for shorter-term projects	Designed for prolonged, team-based endeavors
	Handling of Setbacks	Typically not reported in detail	Candidly shared as valuable learning experiences
InformationSharing	Information Flow	Limited until publication	Continuous sharing of insights and findings
	Transparency	Focus on successful outcomes	Full disclosure of process, including failures
	Data Sharing	Often limited to final results	Includes interim data, tools, and methodologies
ImpactandValue	Impact on Motivation	Delayed gratification until publication	Ongoing feedback and recognition
	Reproducibility	Often challenging due to limited details	Enhanced through comprehensive documentation
	AI Training Potential	Limited to final outcomes	Rich dataset including full exploration process

Table 1: Comparison between ”Traditional Research Paper” and ”Proposed Progress Report”.
As part of our commitment to open science, we will be releasing numerous valuable resources throughout this journey. These include:

(1) Our detailed hypotheses regarding the O1 technical stack, along with a comprehensive map of our cognitive exploration path. This resource provides insight into our strategic thinking and decision-making processes throughout the replication attempt. (2) A collection of insights and positive outcomes derived from our trial-and-error experiences. This compilation offers valuable lessons learned and unexpected

图 3: 比较我们模型与 OpenAI O1-preview 和 GPT-4o 在解决数学问题上的案例研究。

通过这一过程，我们预期在人工智能研究和开发领域产生多方面的影响：(1) 我们期望能够更深入地了解高级语言模型的基本原理，可能揭示出对 O1 所报告的能力有贡献的关键机制。(2) 此外，通过倡导透明度和实时分享我们的发现，我们旨在促进一个更加开放和协作的人工智能研究生态系统，鼓励知识交流和集体问题解决。(3) 最后，通过详细记录我们的整个过程，包括成功和失败，我们将为训练未来的 AI 系统在科学发现方面创建一个无价的数据集，为下一代 AI 驱动的研究方法奠定基础。

Aspects		Traditional Research Paper	Proposed Progress Report
Research Process	Timing of Publication	Published after research completion	Real-time updates throughout the research process
	Length of Research	Suitable for shorter-term projects	Designed for prolonged, team-based endeavors
	Handling of Setbacks	Typically not reported in detail	Candidly shared as valuable learning experiences
InformationSharing	Information Flow	Limited until publication	Continuous sharing of insights and findings
	Transparency	Focus on successful outcomes	Full disclosure of process, including failures
	Data Sharing	Often limited to final results	Includes interim data, tools, and methodologies
ImpactandValue	Impact on Motivation	Delayed gratification until publication	Ongoing feedback and recognition
	Reproducibility	Often challenging due to limited details	Enhanced through comprehensive documentation
	AI Training Potential	Limited to final outcomes	Rich dataset including full exploration process

表 1: “传统研究论文”与“提议的进展报告”对比。

作为我们对开放科学的承诺的一部分，我们将在这一过程中发布许多宝贵的资源。这些资源包括：(1) 关于 O1 技术栈的详细假设，以及我们认知探索路径的全面地图。这一资源提供了对我们整个复制尝试过程中的战略思考和决策过程的洞察。(2) 从我们的试错经验中得出的见解和积极成果的集合。这一汇编提供了宝贵的经验教训和意外发现，可能对更广泛的 AI 研究社区有益。(3) 我们认知过程的广泛文档，包括讨论演示和头脑风暴会议。这些材料提供了对我们团队协作解决问题的方法和创意生成的透明视角。(4) 我们初步努力的初步结果和实验数据，以及对我们定制开发的注释平台的访问。这些资源展示了我们早期的进展，并为从事类似工作的研究人员提供了实用工具。

3 Why We Created Progress Report?

在快速发展的 AI 研究领域中，传统的方法和报告实践越来越不足以应对现代 AI 项目的复杂性和规模。本报告代表了一项开创性的努力，旨在重新构想进行和传达 AI 研究的过程。通过提供我们复制开创性的 O1 模型的旅程的全面实时记录，我们旨在解决当代 AI 研究中的关键挑战，促进开放科学，重新定义科学传播，为 AI 驱动的科学发现奠定基础，并促进负责任的 AI 发展。以下内容不仅是对我们发现的记录，而且是对 AI 时代科学探索和合作新范式的勇敢提议。

1. 应对现代 AI 研究的挑战：人工智能技术的快速发展开启了一个新的研究范式，其特点是长期的团队合作，通常持续六个月或更长。虽然这种转变有利于突破性创新，但也无意中给科学过程带来了新的挑战。长时间团队合作的内在封闭性经常导致向更广泛的科学界的信息流动减少。此外，这些项目的长期性质往往导致研究人员的回报延迟，可能在整个研究过程中引发焦虑和动力下降。此外，大规模团队项目的复杂性使得个人贡献的识别变得更加困难，可能侵蚀传统的学术激励结构。我们的进展报告方法旨在通过提高透明度、促进实时反馈和认可，以及鼓励对长期研究项目的持续承诺来应对这些新兴挑战。

2. 促进开放科学和集体进步：本着开放科学和集体进步的精神，本报告的主要动力是传播我们在复制 O1 模型过程中获得的宝贵见解、资源和教训。这种方法超越了仅仅分享一个训练好的模型；它涵盖了我们在探索过程中使用的所有工具、数据集和方法的全面记录。通过坦率地分享我们的挫折和失败尝试，我们旨在提供超越成功故事的教育价值。这种透明度旨在帮助其他研究人员避免潜在的陷阱，从而加速整个领域的进步。此外，通过阐明我们的思维过程和创新方法，我们希望激发社区内的创造力，促进新思想和方法的产生。

3. 为 AI 在科学发现中的应用奠定基础：我们对科学探索过程的详细记录具有深远的意义，特别是在 AI 能力迅速提升的背景下。通过记录我们探索过程的全部，包括成功和失败，我们正在培养一个独特

discoveries that may benefit the broader AI research community. (3) Extensive documentation of our cognitive processes, including discussion presentations and brainstorming sessions. These materials offer a transparent look into our team’s collaborative problem-solving approach and idea generation. (4) Preliminary results and experimental data from our initial efforts, as well as access to our custom-developed annotation platform. These resources showcase our early progress and provide practical tools for researchers engaged in similar endeavors.

3 Why We Created Progress Report?

In the rapidly evolving landscape of artificial intelligence research, traditional methodologies and reporting practices are increasingly proving inadequate to address the complexities and scale of modern AI projects. This report represents a pioneering effort to reimagine the process of conducting and communicating AI research. By providing a comprehensive, real-time account of our journey to replicate the groundbreaking O1 model, we aim to address critical challenges in contemporary AI research, foster open science, redefine scientific communication, lay the groundwork for AI-driven scientific discovery, and promote responsible AI development. What follows is not merely a documentation of our findings, but a bold proposition for a new paradigm in scientific exploration and collaboration in the AI era.

1. Addressing the Challenges of Modern AI Research: The rapid evolution of artificial intelligence technologies has ushered in a new era of research paradigms, characterized by prolonged, team-based endeavors that often span six months or more. This shift, while conducive to breakthrough innovations, has inadvertently introduced novel challenges to the scientific process. The inherent insularity of extended team collaborations frequently results in a diminished flow of information to the broader scientific community. Moreover, the protracted nature of these projects often leads to delayed gratification for researchers, potentially fostering anxiety and diminished motivation throughout the research journey. Additionally, the complexity of large-scale team projects complicates the recognition of individual contributions, potentially eroding the traditional academic incentive structures. Our progress report methodology aims to address these emergent challenges by enhancing transparency, facilitating real-time feedback and recognition, and encouraging sustained commitment to long-term research initiatives.

2. Fostering Open Science and Collective Advancement: In the spirit of open science and collective advancement, the primary impetus behind this report is to disseminate the invaluable insights, resources, and lessons gleaned from our endeavor to replicate the O1 model. This approach transcends the mere sharing of a trained model; it encompasses a comprehensive documentation of the tools, datasets, and methodologies employed throughout our exploratory process. By candidly sharing our setbacks and unsuccessful attempts, we aim to provide educational value that often surpasses that of mere success stories. This transparency is intended to assist other researchers in navigating potential pitfalls, thereby accelerating progress across the field. Furthermore, by elucidating our thought processes and innovative approaches, we aspire to catalyze creativity within the community, fostering the generation of novel ideas and methodologies.

3. Laying the Foundation for AI in Scientific Discovery: The meticulous documentation of our scientific exploration process holds profound significance, particularly in the context of rapidly advancing AI capabilities. By recording our exploration process in its entirety, including both successes and failures, we are cultivating a unique and invaluable dataset. This comprehensive record is crucial for training AI models that genuinely comprehend scientific methodologies, mirroring the approach validated by the O1 model. The success of O1 underscores the importance of AI systems learning not just outcomes, but the complete scientific exploration process, including trial and error. Our report captures not only technical

而宝贵的数据库。这一全面记录对于训练真正理解科学方法的 AI 模型至关重要，这得到了 O1 模型的验证。O1 的成功强调了 AI 系统不仅学习结果，还要学习完整的科学探索过程，包括试错的重要性。我们的报告不仅捕捉技术细节，还包括决策理由、灵感来源和思维过程。这些“人为因素”对于训练能够进行真实科学发现的 AI 模型至关重要。此外，这种方法具有跨学科的价值，提供了一个研究记录和知识共享的模板，可以促进各个科学领域的创新。

4. 促进负责任的 AI 发展：在追求技术突破的同时，我们对 AI 发展潜在的社会影响和伦理考虑保持高度警觉。通过详细记录我们的研究过程和决策，我们建立了透明度的高标准，这对于培养公众对 AI 研究的信任至关重要。我们的报告不仅超越了技术细节，还包括对潜在社会影响的持续讨论和反思，从而展示了在整个技术发展过程中融入伦理考虑。这种整体方法有助于培养一个更加负责任和伦理导向的 AI 研究文化。和认可，以及鼓励对长期研究项目的持续承诺。

4 旅程学习：从“捷径学习”到新范式的转变

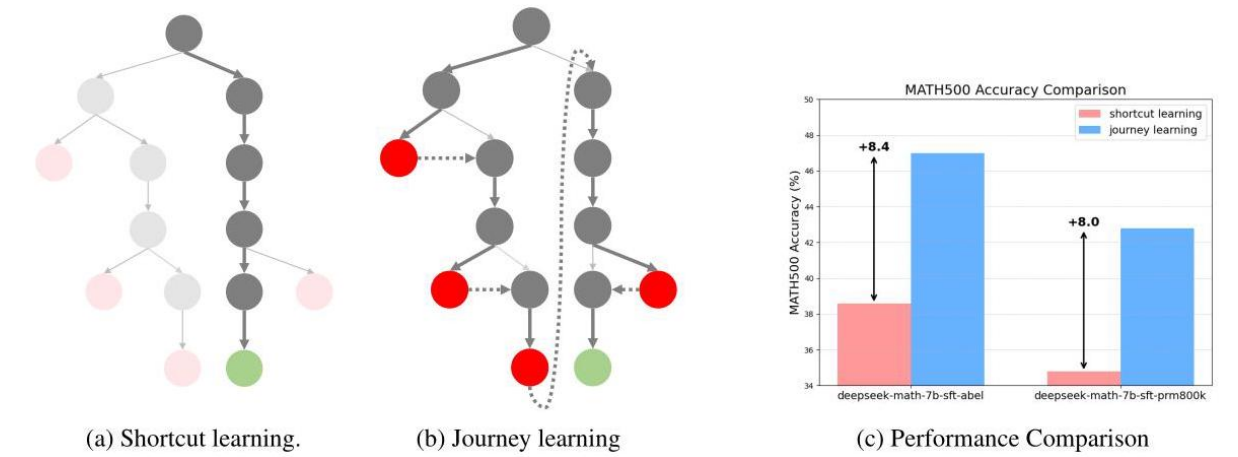


图 4：从“捷径学习”到“旅程学习”的范式转变。用于推理任务的搜索树。对于数学问题解决任务，根节点表示初始问题，而叶节点是最终结论。绿色节点表示正确答案，红色节点表示错误答案。传统上，学习集中在直接从根到叶的捷径路径的监督训练上。然而，本研究探讨了整个探索路径的监督学习，包括试错和纠正过程。(c) “捷径学习”和“旅程学习”在 MATH500 上的表现 (Lightman 等人, 2024)。基础模型是在 Abel 训练数据和 PRM800K 上分别微调的 deepseek-math-7b-base。

我们认为，大多数现有的机器学习或大型语言模型训练方法（例如，监督微调）可以被描述为“捷径学习”。虽然这种传统范式在特定、明确定义的任务中可能有效，但在面对复杂、动态和开放性问题时，显示出显著的局限性。捷径学习的几个关键特征包括：(1) 快速结果导向：它强调在短时间内实现特定的性能指标或完成特定任务。(2) 高度依赖数据：性能改进通常依赖于增加训练数据量，而不是增强学习算法本身。(3) 泛化能力有限：在训练数据分布之外的场景中，性能可能会急剧下降。(4) 缺乏自我纠正能力：这些系统通常缺乏识别和纠正自身错误的能力。虽然捷径学习推动了许多 AI 的进步，但在处理现实世界挑战的复杂性方面，它难以产生真正智能和可靠的 AI 系统。随着我们追求更高级形式的人工智能甚至超级智能，这种方法的局限性变得越来越明显。认识到这些不足，我们提出了一种新的范式，称为“旅程学习”。这种创新方法不仅仅是一种学习方法；它是 AI 开发的新范式。旅程学习旨在使 AI 系统能够通过学习、反思、回溯和适应不断进步，就像人类一样，从而表现出更高水平的智能。

details but also decision rationales, sources of inspiration, and thought processes. These "human factors" are essential for training AI models capable of authentic scientific discovery. Moreover, this approach has interdisciplinary value, offering a template for research documentation and knowledge sharing that can foster innovation across various scientific domains.

4. Promoting Responsible AI Development: In our pursuit of technological breakthroughs, we remain acutely aware of the potential societal impacts and ethical considerations associated with AI development. Through detailed documentation of our research processes and decision-making, we establish a high standard of transparency, which is crucial for cultivating public trust in AI research. Our report goes beyond technical specifics, incorporating ongoing discussions and reflections on potential societal impacts, thereby demonstrating the integration of ethical considerations throughout the technological development process. This holistic approach contributes to the cultivation of a more responsible and ethically-minded AI research culture and recognition, and encouraging sustained commitment to long-term research initiatives.

4 Journey Learning: A New Paradigm Shift from "Shortcut Learning"

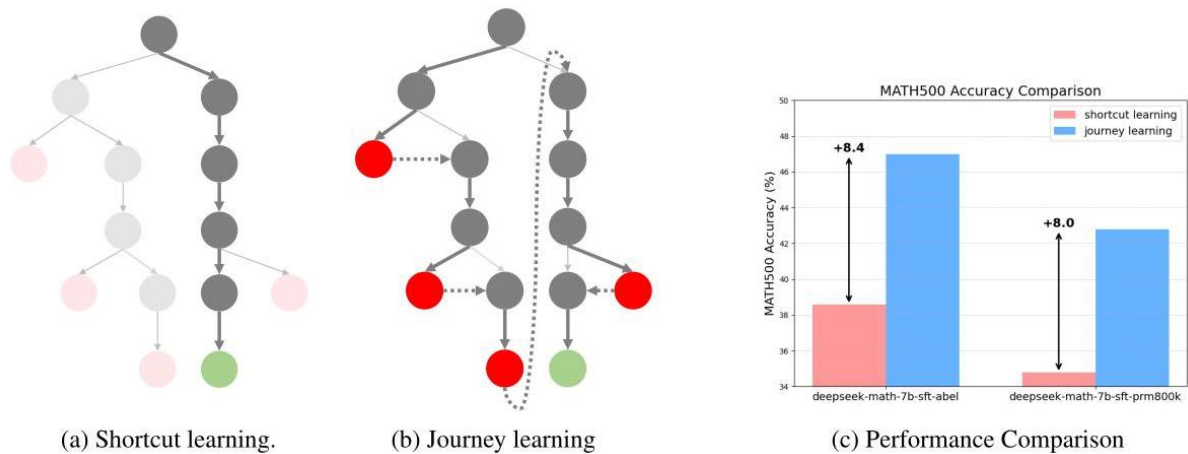


Figure 4: A paradigm shift from "shortcut learning" to "journey learning". A searching tree for reasoning tasks. For the math problem-solving task, the root node represents the initial problem, while the leaf nodes are final conclusions. Green nodes indicate correct answers, and red nodes incorrect ones. Traditionally, learning focused on supervised training of a direct root-to-leaf shortcut path. This work, however, explores supervised learning of the entire exploration path, encompassing trial-and-error and correction processes. (c) Performance of "shortcut learning" and "journey learning" on MATH500 (Lightman et al., 2024). The base models are deepseek-math-7b-base fine-tuned on Abel training data and PRM800K separately.

We claim that most existing approaches to machine learning or large language model training (e.g., supervised fine-tuning) can be characterized as "shortcut learning." This traditional paradigm, while potentially effective in specific, well-defined tasks, shows significant limitations when faced with complex, dynamic, and open-ended problems. Shortcut learning is defined by several key characteristics: (1) Quick results orientation: It emphasizes achieving specific performance metrics or completing particular tasks in a short time frame. (2) Heavy data dependency: Performance improvements often rely on increasing the volume of training data rather than enhancing the learning algorithms themselves. (3) Limited generalization: Performance can deteriorate dramatically in scenarios outside the distribution of the training data. (4) Lack of self-correction: These systems typically lack the ability to identify and correct their own errors. While shortcut learning has driven many advances in AI, it struggles to produce truly

Characteristic	Shortcut Learning	Journey Learning
Learning Depth	Surface features and simple correlations	Deep causal relationships and underlying principles
Reasoning Ability	Limited, struggles with complex reasoning	Powerful, demonstrates human-like reasoning
Self – Improvement	Lacks self-correction mechanisms	Continuous self-assessment and improvement
Generalization	Limited, easily affected by data distribution changes	Strong, can handle new situations
Innovation Capacity	Limited, struggles to solve new problems	High, can generate innovative solutions
DataDependency	Highly dependent on large training datasets	More focused on quality and learning strategies
Interpretability	Poor, often seen as a "black box"	Better, can track internal reasoning processes
Ethical Considerations	May unintentionally amplify data biases	Easier to implement ethical constraints and adjustments
Security	Vulnerable to adversarial attacks	More robust, able to identify potential threats
Long – termValue	Quick results in specific tasks	Paves the way for AGI development
Human Analogy	Exam-oriented education, crash courses	Comprehensive education, lifelong learning

表 2: 捷径学习与旅程学习的比较。

旅程学习在捷径学习的基础上有了显著的进步。虽然捷径学习在复杂、动态的环境中往往表现不佳，但旅程学习专门设计用于在这种场景中茁壮成长。它旨在创建的 AI 系统不仅仅是狭窄的任务特定工具，而是能够处理现实世界挑战的细微差别和复杂性的适应性强、具备推理能力的实体。这一新范式有望带来更强大、更适应、更像人类的 AI，能够更好地在各个领域为人类服务和互动。随着我们继续开发和完善旅程学习范式，我们期待它在 AI 研究和应用中开辟新的可能性，有可能彻底改变我们对人工智能及其在社会中角色的看法。

5 Background

过程级奖励模型过程奖励模型 (PRMs) 用于对 LLMs (Lightman et al., 2024; Uesato et al., 2022; Xia et al., 2024) 的响应进行细粒度评估，特别是在数学推理领域。通过准确评估每一步的正确性，PRMs 可以提高训练后的质量 (Wang et al., 2024c; Sun et al., 2024) 并通过各种搜索方法在推理过程中提高准确性 (Luo et al., 2024; Wang et al., 2024a)。实施 PRMs 可以涉及使用具有高级提示技术的专有模型 (Hao et al., 2024) 或使用步骤级监督数据进行训练 (Xia et al., 2024; Wang et al., 2024c)。后一种方法具有挑战性，因为它需要高质量的标注数据 (Xia et al., 2024)。这导致了对使用强化学习原理的兴趣，这些原理将多步推理过程建模为马尔可夫决策过程 (MDP)，并使用诸如蒙特卡洛树搜索 (Silver et al., 2016) 等技术来估计每一步的价值，无论是在线 (Chen et al., 2024) 还是离线 (Wang et al., 2024c)。

链式思维理论链式思维 (CoT) 提示显著提升了 LLMs 的推理能力。基础研究表明，提供中间推理步骤可以提高在复杂任务（如算术和常识推理）中的表现 (Wei et al., 2022)。此外，理论研究揭示了 CoT 通过启用本质上串行的计算来增强仅解码器的变压器，而这种计算在低深度变压器中尤其缺乏 (Li et al., 2024b)。最近的研究还表明，即使常规模型大小的自回归变压器也可以通过 CoT 推导解决复杂的任务，如算术和决策，这使用了电路复杂性理论 (Feng et al., 2024)。最近的工作强调在预训练阶段整合“错误校正”数据以提高推理准确性，表明这种数据可以在不需要多轮提示的情况下提高准确性 (Ye et al., 2024)。总体而言，这些发现强调了 CoT 提示在增强 LLMs 在复杂推理任务中的性能和可访问性方面的关键作用。内部思维人工智能模型中内部思维的探索已经发展，研究人员强调模型需要反思其推理并改进其输出。早期的工作如 STaR (Zelikman et al., 2022) 提出了通过让模型生成解释其决策的理由来引导推理，使模型能够通过迭代改进在复杂任务中提高性能。在此基础上，Quiet-STaR (Zelikman et al., 2024a) 通过训练语言模型在每个标记后生成理由，帮助它们更有效地预测和解释未来的文本。Zhang et al. (2024) 进一步扩展了这一研究方向，通过在每个训练实例中嵌入反思，鼓励模型审查其决策并考虑替代推理路径。RISE (Qu et al., 2024) 引入了一种递归内省的方法，模型在检测到错误后迭代调整其响应，旨在通过多次尝试实现自我改进。这些发展表明，越来越多的注意力集中在使 AI 模型能够参与反思和自我纠正的过程，增强其处理更复杂推理任务的能力。

推理时间扩展最近的研究表明，扩展推理时间可以比传统的扩展方法（如增加模型参数或训练数据量）更有效地提高模型性能 (Sardana and Frankle, 2023; Snell et al., 2024)。虽然参数扩展一直是提高模

intelligent and reliable AI systems capable of handling the complexities of real-world challenges. As we pursue more advanced forms of artificial intelligence or even superintelligence, the limitations of this approach become increasingly apparent. Recognizing these shortcomings, we propose a new paradigm called "journey learning." This innovative approach represents more than just a learning method; it's a new paradigm for AI development. Journey learning is designed to enable AI systems to progress continuously through learning, reflection, backtracking, and adaptation, much like humans do, thereby exhibiting higher levels of intelligence.

Characteristic	Shortcut Learning	Journey Learning
Learning Depth	Surface features and simple correlations	Deep causal relationships and underlying principles
Reasoning Ability	Limited, struggles with complex reasoning	Powerful, demonstrates human-like reasoning
Self – Improvement	Lacks self-correction mechanisms	Continuous self-assessment and improvement
Generalization	Limited, easily affected by data distribution changes	Strong, can handle new situations
Innovation Capacity	Limited, struggles to solve new problems	High, can generate innovative solutions
DataDependency	Highly dependent on large training datasets	More focused on quality and learning strategies
Interpretability	Poor, often seen as a "black box"	Better, can track internal reasoning processes
Ethical Considerations	May unintentionally amplify data biases	Easier to implement ethical constraints and adjustments
Security	Vulnerable to adversarial attacks	More robust, able to identify potential threats
Long – termValue	Quick results in specific tasks	Paves the way for AGI development
Human Analogy	Exam-oriented education, crash courses	Comprehensive education, lifelong learning

Table 2: Comparison between Shortcut Learning and Journey Learning.

Journey learning represents a significant advancement over shortcut learning. While shortcut learning often falters in complex, dynamic environments, journey learning is specifically designed to thrive in such scenarios. It aims to create AI systems that are not just narrow, task-specific tools, but adaptable, reasoning entities capable of handling the nuances and complexities of real-world challenges. This new paradigm holds the promise of more capable, adaptable, and human-like AI that can better serve and interact with humans across various domains. As we continue to develop and refine the journey learning paradigm, we expect it to open new possibilities in AI research and applications, potentially revolutionizing our approach to artificial intelligence and its role in our society.

5 Background

Process-level Reward Model Process reward models (PRMs) are used to provide fine-grained evaluations of responses from LLMs (Lightman et al., 2024; Uesato et al., 2022; Xia et al., 2024), especially in the area of mathematical reasoning. By accurately assessing the correctness of each step, PRMs can enhance post-training quality (Wang et al., 2024c; Sun et al., 2024) and improve accuracy during inference through various search methods (Luo et al., 2024; Wang et al., 2024a). Implementing PRMs can involve using proprietary models with advanced prompting techniques (Hao et al., 2024) or training with step-level supervision data (Xia et al., 2024; Wang et al., 2024c). The latter approach is challenging because it requires high-quality annotated data (Xia et al., 2024). This has led to interest in using reinforcement learning principles, which model the multi-step reasoning process as a Markov Decision Process (MDP) and use techniques like Monte Carlo Tree Search (Silver et al., 2016) to estimate the value of each step, either online (Chen et al., 2024) or offline (Wang et al., 2024c).

COT Theory Chain-of-thought (CoT) prompting has significantly advanced the reasoning capabilities of LLMs. Foundational studies demonstrate that providing intermediate reasoning steps enhances performance on complex tasks, such as arithmetic and commonsense reasoning (Wei et al., 2022). Additionally, theoretical investigations reveal that CoT empowers decoder-only transformers by enabling inherently serial computation, which is otherwise lacking, particularly in low-depth transformers (Li et

al., 2024). This approach leverages the model's ability to generate intermediate reasoning steps, which can be used to evaluate the correctness of the final answer. The use of CoT has been shown to improve performance on a wide range of tasks, including mathematical reasoning, commonsense reasoning, and question answering. The improvement is attributed to the model's ability to "show its work," which allows for better error detection and correction. The use of CoT also helps to reduce the model's tendency to "hallucinate" or generate incorrect reasoning steps. The use of CoT is a promising direction for improving the reasoning capabilities of LLMs. The use of CoT has been shown to improve performance on a wide range of tasks, including mathematical reasoning, commonsense reasoning, and question answering. The improvement is attributed to the model's ability to "show its work," which allows for better error detection and correction. The use of CoT also helps to reduce the model's tendency to "hallucinate" or generate incorrect reasoning steps. The use of CoT is a promising direction for improving the reasoning capabilities of LLMs.

Large Language Models (LLMs) have shown remarkable progress in natural language processing tasks, but their reasoning capabilities remain a challenge. This paper explores the potential of LLMs in reasoning tasks by analyzing their performance on various benchmarks and identifying key factors that influence their reasoning ability. We propose a framework for evaluating LLM reasoning performance, which includes a set of metrics and a series of experiments. Our results show that LLMs can perform well on reasoning tasks, but their performance is highly dependent on the quality of the input and the complexity of the task. We discuss the implications of our findings for the development of LLM reasoning systems and provide suggestions for future research.

6 Exploration Journey

This section represents the core of our O1 replication work. This section explores the exploration process through a series of key questions, reflecting the complex path shown in the research timeline diagram. From using OlympicArena (Huang et al., 2024) dataset for O1 initial evaluation, to the complex "long-term thinking construction" stage, our journey is filled with multiple attempts, continuous iteration, and a deep exploration of the O1 capabilities.

In this chapter, we explore the issues not only reflecting the progress of research, but also reflecting our deep exploration of the O1 cognitive process. We first explore the O1 thinking structure, then deeply explore the mechanism and construction of long-term thinking——this is what is shown in the diagram "long-term thinking construction" stage. Our exploration extends to the development of the reward model, the construction of the strategy-based reasoning tree, and the integration of these elements into a coherent long-term thinking, reflecting the complex interconnected process in the research timeline. As shown in the figure, our method involves multiple iterations and parallel investigation flows. This

al., 2024b). Recent studies also reveal CoT prompting enhances LLMs by showing that even constant-sized autoregressive Transformers can solve complex tasks like arithmetic and decision-making through CoT derivations, using circuit complexity theory (Feng et al., 2024). Recent work emphasizes the integration of "error-correction" data into the pretraining stage to enhance reasoning accuracy, showing that such data can lead to higher accuracy without the need for multi-round prompting (Ye et al., 2024). Overall, these findings underscore the pivotal role of CoT prompting in enhancing LLM performance and accessibility in complex reasoning tasks.

Internal Thought The exploration of internal thought in AI models has evolved as researchers emphasize the need for models to reflect on their reasoning and refine their outputs. Early work like STaR (Zelikman et al., 2022) proposed bootstrapping reasoning by having models generate rationales that explain their decisions, allowing them to improve their performance on complex tasks through iterative refinement. Building on this, Quiet-STaR (Zelikman et al., 2024a) generalizes the approach by training language models to generate rationales after each token, helping them predict and explain future text more effectively. Zhang et al. (2024) further expanded this line of work by embedding reflection within each training instance, encouraging models to review their decisions and consider alternative reasoning paths. RISE (Qu et al., 2024) introduced a method for recursive introspection, where models iteratively adjust their responses after detecting errors, aiming for self-improvement over multiple attempts. These developments illustrate the growing focus on enabling AI models to engage in reflective, self-correcting processes, enhancing their ability to handle more complex reasoning tasks.

Inference Time Scaling Recent studies have demonstrated that scaling inference time can provide more efficient improvements in model performance (Sardana and Frankle, 2023; Snell et al., 2024) compared to traditional scaling approaches such as increasing model parameters or training data volume. While parameter scaling has been a dominant paradigm in advancing model capabilities (Kaplan et al., 2020; Brown et al., 2020; Chowdhery et al., 2022), it often leads to diminishing returns and significant computational overhead. In contrast, allowing models more time to process and refine their outputs during inference has emerged as a promising alternative scaling dimension (Madaan et al., 2023). Inference time scaling offers several advantages: 1) Resource efficiency, utilizing existing model capacities more thoroughly; 2) Adaptable computation, allocating more processing time to complex tasks; and 3) Improved reasoning through step-by-step problem solving or iterative refinement (Yao et al., 2023; Cobbe et al., 2021). Empirical evidence suggests that doubling inference time often yields performance improvements comparable to those achieved by significantly increasing model size, but at a fraction of the computational cost (Snell et al., 2024). Some successful implementations include deliberation mechanisms and iterative refinement protocols (Huang et al., 2022; Miao et al., 2023), which have shown particular promise in tasks requiring complex reasoning or creative generation.

Search-to-thought In recent years, the shift from traditional search-based methods to implicit reasoning approaches has significantly advanced AI research (Ruoss et al., 2024). Classic systems like Deep Blue (Campbell et al., 2002) relied heavily on explicit search algorithms such as alpha-beta pruning and Monte Carlo Tree Search to achieve superhuman performance (Silver et al., 2017). However, with the advent of deep learning, Chain of Thought (CoT) (Wei et al., 2022) reasoning has gained significant attention for its ability to improve model performance by generating intermediate reasoning steps without search. Implicit Chain-of-Thought Reasoning (Deng et al., 2023) bypass the need for generating explicit reasoning steps by leveraging the internal hidden states of models. This method distills knowledge from a teacher model trained to generate intermediate steps, allowing student models to solve tasks more efficiently by reasoning vertically through their internal layers. Similarly in chess AI, a 270M parameter transformer model can achieve grandmaster-level play without any explicit search by learning action-values through

方法在我们对评估方法和训练策略的讨论中得到了体现，展示了我们如何通过定量和定性评估的循环来验证假设和改进技术，包括人工检查和专门的分析工具。

通过围绕这些关键问题构建本章，我们不仅提供了我们技术旅程的清晰叙述，还展示了探索未知 AI 技术的系统方法。这种以问题为导向的格式与我们的“旅程学习”范式相一致，强调整个学习和探索过程的重要性，而不仅仅是最终结果。随着我们逐步解答每个问题，读者将了解我们的决策过程、面临的挑战以及开发的创新解决方案。通过研究时间线中所示的透明分享我们的思维过程、尝试甚至失败，我们旨在为 AI 社区提供宝贵见解，并促进该领域的集体进步。

通过本节，我们邀请读者跟随我们的探索旅程，不仅了解我们对 O1 的发现，还了解我们如何在信息有限的情况下，应对复制这一突破性 AI 模型的艰巨任务。我们充满好奇心、坚持不懈和创新的旅程，证明了开放、协作的 AI 研究在推动人工智能边界方面的力量。

6.1 Q1: What does O1’s Thought Look Like?

表 3 是基于 OpenAI 提供的 O1 的思想示例的详细分析创建的，¹ 其中包括八个解决复杂任务的推理步骤，或“思想”实例。每个示例都经过仔细检查，以提取相关的特征，如标记数量、行数和关键词。这些示例被分类为不同类型的问题，每个问题类型都与从简单的英语阅读理解到复杂的多步骤数学推理任务的难度级别相关联。我们的分析显示了一个趋势：随着难度的增加，响应长度（无论是标记还是行数）往往会成比例增长。这表明，难度较高的问题涉及更多的推理步骤。

除了标记和行数的计数，我们还进行了关键词频率分析，以识别可能表征推理过程的重复出现的术语。除了常见的连接词如“and”和“so”之外，我们的分析还突出了几个出现频率较低但非常重要的关键词。诸如“consider”、“if”和“possible”等关键词频繁出现，通常标志着推理过程中的分支，其中考虑了多个路径。这些关键词在复杂度较高的问题中的频率显著较高，表明模型在这些情况下探索了不同的解决方案路径。像“wait”和“Alternatively”这样的关键词是模型能够进行反思和自我修正的重要指标。这表明模型不仅遵循线性路径，还能够基于反思重新考虑和改进其方法，显示出更深层次的理解和更细致的推理方法。

为了理解 OpenAI 的 O1 的思维过程，我们咨询了数学系的两名博士生，仔细审查了 O1 在解决数学问题时所采用的推理过程。通过他们的详细检查，他们提取了反映 O1 如何处理和推理复杂方程的潜在思维链。这个结构化的思维图如图 5 所示。经过这些探索，我们确定了我们构建的长思维数据应具有以下特征：

- 迭代问题解决：该模型首先定义函数，然后逐步探索相关表达式，将复杂的方程分解为更简单的组件，体现了结构化和系统化的方法。
- 关键思想指标：使用诸如“因此”来表示结论，“或者”来探索不同路径，“等一下”来反思，以及“让我计算一下”来过渡到计算，这些都突显了模型的推理阶段。
- 递归和反思方法：该模型经常重新评估和验证中间结果，使用递归结构来确保一致性，这是严谨的数学推理中常见的做法。
- 假设探索：模型测试不同的假设，随着收集到更多信息，调整其方法，展示了其推理过程的灵活性。
- 结论与验证：最后，模型求解方程并验证结果，强调在完成之前验证结论的重要性。

¹ <https://openai.com/index/learning-to-reason-with-llms/>

supervised training on a large dataset of games (Ruoss et al., 2024). These approaches highlight a trend where models are increasingly able to generalize complex reasoning and decision-making processes internally, thus reducing reliance on computationally expensive search algorithms while maintaining high performance in domains like mathematical reasoning and game playing.

Self-improvement in LLM Self-improvement methods for LLMs aim to enhance model performance by enabling them to learn from their own outputs with minimal human intervention. These approaches typically involve supervised fine-tuning (SFT) on high-quality outputs generated by the models (Zelikman et al., 2024b; Li et al., 2024a; Wang et al., 2024d) or preference optimization, where the model learns from pairs of good and bad responses it generated to a query (Xu et al., 2024; Yuan et al., 2024; Pang et al., 2024; Wu et al., 2024a). In general instruction-following tasks, the quality of model outputs is often determined by an external reward system-this can be a trained reward model (Xu et al., 2024), human evaluators (Ziegler et al., 2019), or the LLMs themselves through techniques like LLM-as-a-Judge prompting (Zheng et al., 2023). In mathematical domains, however, output quality is primarily judged by whether the model reaches the correct answer (Zelikman et al., 2024b; Pang et al., 2024). For more fine-grained evaluation, step-level rewards for mathematical reasoning tasks may be assigned by human annotators or a trained process reward model (Lightman et al., 2024). Iterative self-improvement techniques have shown promise across a range of tasks, from instruction-following (Xu et al., 2024; Yuan et al., 2024) to more complex reasoning-based challenges (Zelikman et al., 2024b; Pang et al., 2024), highlighting their potential for driving further advancements in LLM capabilities. However, recent findings suggest that LLM-generated texts often exhibit truncated "tails", meaning that the distribution of generated outputs lacks the variability found in human-generated content, particularly in the less common, outlier responses (or "tails" of the distribution) (Shumailov et al., 2024; Dohmatob et al., 2024). This reduced variability can lead to a phenomenon known as model collapse, where the model converges toward a narrower range of behaviors, ultimately harming performance (Shumailov et al., 2024). This issue has been observed in tasks like language modeling (Shumailov et al., 2024) and iterative preference optimization for mathematical reasoning (Wu et al., 2024b). To mitigate the risk of model collapse, researchers recommend maintaining a balanced mix of clean, human-authored data alongside LLM-generated content during training (Shumailov et al., 2024; Dohmatob et al., 2024; Gerstgrasser et al., 2024). This approach helps preserve diversity and prevents the model from degrading in performance over time.

6 Exploration Journey

This section represents the core of our O1 replication endeavor. This section systematically unfolds our exploration process through a series of pivotal questions, mirroring the complex pathway illustrated in our research timeline diagram. From the initial evaluation of O1 using OlympicArena (Huang et al., 2024) datasets to the intricate "Long Thought Construction" phase, our journey has been marked by multiple attempts, continuous iterations, and deep dives into the essence of O1's capabilities.

The questions we address in this chapter not only reflect the progression of our research but also embody our profound inquiry into the nature of O1's cognitive processes. We begin by examining the structure of O1's thoughts, then delve into the mechanics and construction of long thoughts - a concept central to our "Long Thought Construction" phase depicted in the diagram. Our exploration extends to the development of reward models, the construction of on-policy reasoning trees, and the integration of these elements into cohesive long thoughts, mirroring the complex web of interconnected processes in our research timeline. Our methodology, as visualized in the diagram, involves multiple iterations and

6.2 Q2: How does Long Thought Work?

这是一个我们认为重要的问题。然而，在我们目前的进展阶段，我们只是提出我们的假设。我们并不认为我们有足够的实证证据来验证它们的准确性。O1 的长期思考方法之所以取得显著成功，可以归因于我们在第 4 节中介绍的旅程学习。与传统的捷径学习不同，旅程学习使模型能够探索整个决策轨迹，模仿人类的问题解决过程。这种全面的探索使 O1 能够考虑多种解决方案路径，从错误中学习，并理解完整的问题解决过程。通过体验正确和错误的路径，模型发展出强大的错误处理和自我纠正能力，增强其对新挑战的适应性。这种方法促进了对问题领域的更深入理解，不仅仅知道正确答案，而是理解为什么和如何得出正确答案。旅程学习过程紧密模拟人类的认知过程，包括试错、反思和调整。这导致了更高的可解释性，因为 O1 可以提供详细的解决方案步骤，并解释其推理过程，包括它是如何从错误中恢复的。因此，O1 的长期思考过程，基于旅程学习，不仅仅是延长计算时间，而是代表了一种彻底的、类似人类的推理探索。这种方法使 O1 能够处理更复杂的问题，提供更可靠和可解释的答案，并在面对新挑战时表现出更大的适应性，从而解释了它在各种任务中的卓越表现。

6.3 Q3: How to Construct Long Thoughts?

构建长思考的过程涉及到反思和回溯等行为，这是旅程学习的核心部分。为了实现这一点，我们进行了一系列尝试。

尝试 1: 基于 LLM 和奖励的树搜索根据我们在第 6.1 节对长思考的观察，其最显著的特征是在推理导致错误或无用节点时尝试反思和回溯。这类似于在问题的推理树上进行搜索，在错误节点处回溯，直到找到正确的解决方案路径。为了实现这一点，我们需要构建一个推理树，其中根节点代表问题，每个其他节点代表一个推理步骤。从根节点到任何节点的路径表示从问题到该结论的推理过程。此外，回溯和反思必须基于错误的推理步骤，这需要一个更细粒度的奖励模型（即过程级）来指示树中每个节点的正确性。通过在具有过程级奖励的推理树上执行搜索算法，我们可以将错误步骤整合到思考链中，从而构建包含回溯和反思等行为的长思考。

尝试 2: 提出-批评循环尝试 1 通过基于预定义规则在树上执行搜索来构建长思考，但这限制了回溯和反思等行为的自由度。因此，我们允许模型选择其当前行为。我们构建了一个提出-批评循环，其中我们预定义了一些模型可能采取的行为（即继续、回溯、反思、终止），并让模型选择行为来构建推理树。如果树没有达到最终答案，模型可以收到这一负面信号，引导其反思并纠正其方法。

尝试 3: 多代理方法在基于推理树构建长思考的过程中存在多个挑战，包括许多无效节点不有助于构建长思考，以及由不依赖于反思行为的推理步骤引起的逻辑不一致问题。为了解决这些问题，我们设计了一种利用多代理辩论的算法，其中一个代理作为策略模型，持续进行推理，而另一个代理作为批评模型，指示策略模型是否应继续当前推理或执行回溯等行为。两个代理进行持续对话，当找到正确答案时，自然构建了一个长思考数据集。

尝试 4: 完整的人类思考过程注释当人类解决推理问题时，他们通常不会一直进行正向推理，直到解决问题或失败；相反，当他们无法继续时，他们会反思、回溯并重写推理。这种行为与长思考的特征非常接近。因此，我们可以忠实地、全面地记录人类解决推理任务的过程，从而生成高质量的长思考。

OpenAI o1 在数学推理中的思考结构

```
# 问题: 解方程  $p(1/x) = x^2$ 
## 步骤 1: 定义  $q(x) = p(1/x) - x^2$ 
- 分析  $q(x)$  的根:
  - 因此:  $q(k) = 0$  对于  $k = \pm 1, \pm 2, \dots, \pm n$ 
- 让我考虑: 构建  $s(x) = x^{2n}q(x)$ 
```

parallel streams of investigation. This approach is reflected in our discussion of evaluation methods and training strategies, showcasing how we validate hypotheses and refine our techniques through cycles of quantitative and qualitative assessments, including human checks and specialized analysis tools.

By structuring this chapter around these key questions, we not only provide a clear narrative of our technical journey but also demonstrate a systematic approach to exploring unknown AI technologies. This question-driven format aligns with our "journey learning" paradigm, emphasizing the importance of the entire learning and exploration process, not just the final outcomes. As we progress through each question, readers will gain insights into our decision-making process, the challenges we faced, and the innovative solutions we developed. This transparent sharing of our thought processes, attempts, and even failures, as illustrated in our research timeline, aims to contribute valuable insights to the AI community and foster collective advancement in the field.

Through this section, we invite readers to traverse our exploration journey, understanding not just what we discovered about O1, but how we approached the daunting task of replicating a groundbreaking AI model with limited information. Our journey, marked by curiosity, persistence, and innovation, serves as a testament to the power of open, collaborative AI research in pushing the boundaries of what's possible in artificial intelligence.

6.1 Q1: What does O1's Thought Look Like?

The Table 3 is created based on a detailed analysis of O1's thought examples provided by OpenAI,¹ which includes eight instances of reasoning steps, or "thoughts," for solving complex tasks. Each example in this is meticulously examined to extract relevant features such as the number of tokens, lines, and keyword. These examples are categorized into different problem types, each associated with a difficulty level ranging from simple English reading comprehension to complex multi-step math reasoning tasks. Our analysis demonstrates a trend: as the difficulty increases, the response length (both tokens and lines) tends to grow proportionally. This suggests that higher difficulty problems involve more reasoning steps.

In addition to token and line counts, we conducted a keyword frequency analysis to identify recurring terms that may characterize the reasoning process. In addition to commonly observed connective words like "and" and "so", our analysis highlights several less frequently occurring but highly significant keywords. Keywords such as "consider", "if" and "possible" appear frequently, often signaling branching in the reasoning process where multiple paths are considered. The frequency of these keywords was notably higher in problems with higher complexity, indicating the model's exploration of different solution paths in these scenarios. Keywords like "wait" and "Alternatively" are crucial indicators of the model's ability to engage in reflection and self-correction. This suggests a deeper understanding and a more nuanced approach to reasoning, as the model is not just following a linear path but is capable of reconsidering and refining its approach based on reflection.

To understand the thought process of OpenAI's O1, we consult two PhD candidates from the mathematics department to carefully review the reasoning process employed by OpenAI's O1 in solving mathematical problems. Through their detailed examination, they extracted the underlying thought chain that reflects how O1 approaches and reasons through complex equations. This structured thought graph is illustrated in Figure 5. After these explorations, we determined that the long thought data we need to construct should have the following characteristics:

- Iterative Problem-Solving: The model starts by defining functions and gradually explores related expressions, breaking down complex equations into simpler components, reflecting a structured and methodical approach.

- 分析 $s(x)$ 的性质:
 - * 因此: $s(x)$ 是一个多项式
- 或者: 考虑 $s(x) = x^{2n}p(1/x) - x^{2n+2}$
 - * 让我解释: 展开 $p(1/x)$ 的表达式。
- 因式分解 $s(x) = m(x)h(x)$
 - * 定义 $m(x) = \prod (x^2 - k^2)$ * 假设 $h(x)$ 的形式:
 - 考虑: $h(x)$ 必须是一个二次多项式
 - 尝试: $h(x) = -x^2 + bx + c$
 - 等待: 我们需要确定 b 和 c 的值
 - 让我计算: 确定 $h(x)$ 的系数

步骤 2: 或者, 构造 $t(x) = x^{2n}p(1/x) = p^*(x)$

- 分析 $p^*(x)$ 的性质:
 - 因此: $p^*(x)$ 是 $p(x)$ 的互反多项式
- 考虑 $s(x) = t(x) - x^{2n+2}$
 - 因此: $s(x) = p^*(x) - x^{(2n+2)}$
- 让我考虑 $(p^*(x))$ 的系数
 - 分析常数项: $p^*(0) = 1$
 - 分析 x^{2n} 项的系数: $a_0 = c$

步骤 3: 让我计算: 展开 $m(x)h(x)$

- 分析高阶项: $-x^{2n+2} + cx^{2n}$
- 与 $s(x) : -x^{2n+2} + [\text{低阶项}]$ 比较
 - 因此: 确认 $h(x) = -x^2 + c$

步骤 4: 等待: 让我们重新考虑 $m(x)$ 在 $x=0$ 时的值

- 计算: $m(0) = (-1)^n(n!)^2$
- 考虑: $s(0) = m(0)h(0) = 1$
- 因此: $c = \frac{(-1)^n}{(n!)^2}$
 - 验证: 这与前面的结果一致

步骤 5: 因此: 解 $h(x) = 0$

- $-x^2 + c = 0$
- $x^2 = c = \frac{1}{(n!)^2}$
- 因此: 额外的解是 $x = \pm 1/n!$
- 等待: 我们需要考虑 n 为偶数的情况
 - 当 n 为偶数时: $c > 0$ - 验证: 这确保了解是实数

图 5: OpenAI O1 在数学推理中的思维结构。

案例 1: 多项式乘法代数问题的解 # 问题: $(3t^2 + 5t + a)$ 和 $(4t^2 + bt - 2)$ 的乘积是 $12t^4 + 26t^3 - 8t^2 - 16t + 6$ 。求 $a + b$ 的值。# 我们提出的模型回答: 首先, 我们需要展开乘积 $(3t^2 + 5t + a)(4t^2 + bt - 2)$ 。当我们这样做时, 应该仔细地乘以每一项:

- $3t^2$ 和 $4t^2$ 的乘积给我们 $12t^4$ 。
- $3t^2$ 和 bt 的乘积给我们 $3bt^3$ 。
- $3t^2$ 与 -2 的乘积为我们提供了 $-6t^2$ 。
- $5t$ 和 $4t^2$ 的乘积给我们 $20t^3$ 。

- Key Thought Indicators: The use of terms like "Therefore" for conclusions, "Alternatively" for exploring different paths, "Wait" for reflection, and "Let me compute" for transitioning into calculations highlights the model's reasoning stages.
- Recursive and Reflective Approach: The model frequently reassesses and validates intermediate results, using a recursive structure to ensure consistency, which is typical in rigorous mathematical reasoning.
- Exploration of Hypotheses: The model tests different hypotheses, adjusting its approach as it gathers more information, demonstrating flexibility in its reasoning process.
- Conclusion and Verification: Finally, the model solves the equations and verifies the results, emphasizing the importance of validating conclusions before finishing.

6.2 Q2: How does Long Thought Work?

This is a question we consider important. However, at our current stage of progress, we are merely putting forward our hypotheses. We don't believe we have sufficient empirical evidence to verify their accuracy. The remarkable success of O1's long-thought approach can be attributed to journey learning, which we have introduced in §4. Unlike traditional shortcut learning, journey learning allows the model to explore the entire decision trajectory, mimicking human problem-solving processes. This comprehensive exploration enables O1 to consider multiple solution paths, learn from errors, and understand the complete problem-solving process. By experiencing both correct and incorrect paths, the model develops robust error-handling and self-correction capabilities, enhancing its adaptability to new challenges. This approach fosters a deeper understanding of the problem domain, going beyond merely knowing the correct answer to comprehending why and how to arrive at it. The journey learning process closely simulates human cognitive processes, incorporating trial-and-error, reflection, and adjustment. This results in enhanced explainability, as O1 can provide detailed solution steps and explain its reasoning, including how it recovers from mistakes. Consequently, O1's long thought process, grounded in journey learning, is not simply about extended computation time but represents a thorough, human-like reasoning exploration. This methodology equips O1 to handle more complex problems, offer more reliable and interpretable answers, and demonstrate greater adaptability when faced with novel challenges, thus explaining its exceptional performance across various tasks.

6.3 Q3: How to Construct Long Thoughts?

Constructing long thoughts with actions such as reflection and backtracking is the core part of journey learning. To achieve this, we undertook a series of attempts.

Attempt 1: Tree Search with LLM and Reward Based on our observations of long thought in §6.1, its most prominent feature is the attempt to reflect and backtrack when reasoning leads to an incorrect or unhelpful node. This resembles searching on a reasoning tree for a problem, backtracking at erroneous nodes, until the correct solution path is found. To achieve this, we need to construct a reasoning tree where the root node represents the problem, and each other node represents a reasoning step. The path

¹ <https://openai.com/index/learning-to-reason-with-llms/>

- $5t$ 和 bt 的乘积给我们 $5bt^2$ 。
- $5t$ 和 -2 的乘积给我们 $-10t$ 。
- a 与 $4t^2$ 的乘积给我们 $4at^2$ 。
- a 和 bt 的乘积给我们 abt 。
- a 与 -2 的乘积为我们提供了 $-2a$ 。

Now, let's combine like terms:

- t^4 项是 $12t^4$ ，所以我们有 $12t^4$ 。
- t^3 项是 $3bt^3$ 和 $20t^3$ ，它们合并为 $(3b + 20)t^3$ 。
- t^2 项是 $-6t^2$, $5bt^2$ 和 $4at^2$ ，它们合并为 $(-6 + 5b + 4a)t^2$ 。
- t 项是 $-10t$ 和 abt ，它们合并为 $(-10 + ab)t$ 。
- The constant term is $-2a$ 。

让我们暂停并考虑到目前为止我们所知道的。我们需要这个展开的表达式与给定的多项式 $12t^4 + 26t^3 - 8t^2 - 16t + 6$ 匹配。

现在，让我们将同类项的系数相等：

- 对于 t^4 项，我们有 $12 = 12$ ，这是一致的。
- 对于 t^3 项，我们得到 $26 = 3b + 20$ 。解得 $b = 2$ 。
- 对于 t^2 项，我们有 $-8 = -6 + 5b + 4a$ 。代入 $b = 2$ ，我们得到 $-8 = -6 + 10 + 4a$ ，这简化为 $4a = -14$ 和 $a = -3.5$ 。
- 对于 t 项，我们有 $-16 = -10 + ab$ 。代入 $a = -3.5$ 和 $b = 2$ ，我们得到 $-16 = -10 - 7$ ，这是一致的。
- 对于常数项，我们有 $6 = -2a$ 。代入 $a = -3.5$ ，我们得到 $6 = 7$ ，这不成立。

等一下，让我们再检查一下我们的计算。常数项的计算应该是 $6 = -2a$ ，这给出 $a = -3$ 。让我们回去纠正这一点。

当 $a = -3$ 和 $b = 2$ 时，我们发现 $a + b = -3 + 2 = -1$ 。

因此， $a + b$ 的正确值是-1。

图 6: 我们提出的模型在数学推理中生成的思考。

案例 2: 余数问题的解

问题: 求

$$f(x) = x^{10} + 5x^9 - 8x^8 + 7x^7 - x^6 - 12x^5 + 4x^4 - 8x^3 + 12x^2 - 5x - 5$$

被 $x^2 - 1$ 除，而不使用长除法。

我们提出的模型回答:

from the root to any node signifies the reasoning process from the problem to that conclusion. Moreover, backtracking and reflection must be based on incorrect reasoning steps, necessitating a more fine-grained reward model (i.e., process-level) to indicate the correctness of each node in the tree. By executing a search algorithm on a reasoning tree with process-level rewards, we can integrate erroneous steps into a chain of thought, thereby constructing long thought that encompasses actions like backtracking and reflection.

Attempt 2: Propose-Critique Loop Attempt 1 constructs long thought by executing searches on the tree based on predefined rules, but this limits the freedom of actions like backtracking and reflection. Therefore, we allow the model to choose its current actions. We constructed a Propose-Critique Loop, where we pre-define some possible actions for the model (i.e., continue, backtracking, reflection, terminate) and let the model select actions to build the reasoning tree. If the tree does not reach the final answer, the model can be informed of this negative signal, guiding it to reflect and correct its approach.

Attempt 3: Multi-Agent Approach Building long thought on the foundation of a reasoning tree presents several challenges, including the presence of numerous ineffective nodes that do not contribute to constructing Long Thought, as well as issues of logical inconsistency caused by reasoning steps that do not depend on the reflection behavior. To address this, we designed an algorithm utilizing multi-agent debate, where one agent acts as the policy model, continuously reasoning, while another agent serves as the critique model, indicating whether the policy model should continue with the current reasoning or perform actions like backtracking. The two agents engage in ongoing dialogue, naturally constructing a long thought dataset when the correct answer is found.

Attempt 4: Complete Human Thought Process Annotation When humans tackle reasoning problems, they typically do not engage in constant forward reasoning until they either solve the problem or fail; instead, they reflect, backtrack, and rewrite reasoning when they can no longer proceed. This behavior closely aligns with the characteristics of long thought. Thus, we can faithfully and comprehensively document the process by which humans solve reasoning tasks, resulting in high-quality long thought.

The Thought Structure of OpenAI o1 in Mathematical Reasoning

#Problem: Solving the equation $p(1/x) = x^2$

##Step 1: Define $q(x) = p(1/x) - x^2$

- Analyze the roots of $q(x)$:
 - Therefore: $q(k) = 0$ for $k = \pm 1, \pm 2, \dots, \pm n$
- Let me consider: Construct $s(x) = x^{2n}q(x)$
 - Analyze the properties of $s(x)$:
 - * Therefore: $s(x)$ is a polynomial
 - Alternatively: Consider $s(x) = x^{2n}p(1/x) - x^{2n+2}$
 - * Let me explain: Expand the expression for $p(1/x)$.
 - Factorize $s(x) = m(x)h(x)$
 - * Define $m(x) = \prod (x^2 - k^2)$
 - * Hypothesize the form of $h(x)$:
 - Consider: $h(x)$ must be a quadratic polynomial
 - Try: $h(x) = -x^2 + bx + c$
 - Wait: We need to determine the values of b and c
 - Let me compute: Determine the coefficients of $h(x)$

##Step 2: Alternatively, construct $t(x) = x^{2n}p(1/x) = p^*(x)$

首先, 我们需要找到 $f(x)$ 被 $x^2 - 1$ 除时的余数。为此, 我们可以使用余数定理, 该定理指出, 如果我们将多项式 $f(x)$ 除以 $x - a$, 则余数为 $f(a)$ 。

现在, 让我们将其应用到我们的问题中。我们需要找到 $f(x)$ 被 $x^2 - 1$ 除时的余数。这表明我们应该将 $x = 1$ 和 $x = -1$ 代入 $f(x)$ 以找到余数。

让我们从 $x = 1$ 开始。将此代入 $f(x)$, 我们得到:

$$f(1) = 1^{10} + 5(1)^9 - 8(1)^8 + 7(1)^7 - (1)^6 - 12(1)^5 + 4(1)^4 - 8(1)^3 + 12(1)^2 - 5(1) - 5$$

Calculating this step-by-step:

$$1^{10} = 1, 5(1)^9 = 5, -8(1)^8 = -8, 7(1)^7 = 7, -(1)^6 = -1, -12(1)^5 = -12, 4(1)^4 = 4, \dots$$

Adding these values together, we find:

$$f(1) = 1 + 5 - 8 + 7 - 1 - 12 + 4 - 8 + 12 - 5 - 5 = -10$$

因此, 当 $f(x)$ 被 $x - 1$ 除时的余数是 -10。

现在, 让我们尝试 $x = -1$:

$$f(-1) = (-1)^{10} + 5(-1)^9 - 8(-1)^8 + 7(-1)^7 - (-1)^6 - 12(-1)^5 + 4(-1)^4 - 8(-1)^3 + 12(-1)^2 - 5(-1) - 5$$

Calculating each term:

$$(-1)^{10} = 1, 5(-1)^9 = -5, -8(-1)^8 = -8, 7(-1)^7 = -7, -(-1)^6 = -1, -12(-1)^5 = 12, \dots$$

Adding these values, we get:

$$f(-1) = 1 - 5 - 8 - 7 - 1 + 12 + 4 + 8 + 12 + 5 - 5 = 16$$

因此, 当 $f(x)$ 被 $x + 1$ 除时, 余数是 16。

让我们暂停并考虑我们目前所知道的。我们已经找到了 $f(x)$ 被 $x - 1$ 和 $x + 1$ 除时的余数。

现在, 我们需要利用这些结果来确定 $f(x)$ 被 $x^2 - 1$ 除时的余数。

由于 $x^2 - 1$ 可以分解为 $(x - 1)(x + 1)$, 因此 $f(x)$ 被 $x^2 - 1$ 除时的余数应该是一个线性表达式 $ax + b$ 。

让我们将 $f(x)$ 表达为: $f(x) = (x^2 - 1)q(x) + ax + b$, 其中 $q(x)$ 是商。

此时, 我们知道:

- 当 $f(x)$ 被 $x - 1$ 除时, 余数是 -10。
- 当 $f(x)$ 被 $x + 1$ 除时, 余数是 16。

让我们利用这些信息来求 a 和 b 。

首先, 考虑余数定理:

- 代入 $x = 1$, 我们有: $f(1) = a(1) + b = -10$
- 代入 $x = -1$: $f(-1) = a(-1) + b = 16$

让我们同时解这些方程:

从第一个方程: $a + b = -10$

从第二个方程: $-a + b = 16$

将这些方程相加得到: $2b = 6 \Rightarrow b = 3$

- Analyze the properties of $p^*(x)$:
 - Therefore: $p^*(x)$ is the reciprocal polynomial of $p(x)$
- Consider $s(x) = t(x) - x^{2n+2}$
 - Thus: $s(x) = p^*(x) - x^{2n+2}$
- Let me consider the coefficients of $(p^*(x))$
 - Analyze the constant term: $p^*(0) = 1$
 - Analyze the coefficient of the x^{2n} term: $a_0 = c$

##Step 3: Let me compute: Expand $m(x)h(x)$

- Analyze the higher-order terms: $-x^{2n+2} + cx^{2n}$
- Compare with $s(x) : -x^{2n+2} + [lower\ degree\ terms]$
 - Thus: Confirm $h(x) = -x^2 + c$

##Step 4: Wait: Let us reconsider the value of $m(x)$ at $x = 0$

- Compute: $m(0) = (-1)^n (n!)^2$
- Consider: $s(0) = m(0)h(0) = 1$
- Therefore: $c = \frac{(-1)^n}{(n!)^2}$
 - Verify: This is consistent with the previous result

##Step 5: So: Solve $h(x) = 0$

- $-x^2 + c = 0$
- $x^2 = c = \frac{1}{(n!)^2}$
- Therefore: Additional solutions are $x = \pm 1/n$!
- Wait: We need to consider the case when n is even
 - When n is even: $c > 0$
 - Verify: This ensures that the solutions are real

Figure 5: The Thought Structure of OpenAI O1 in Mathematical Reasoning.

将 $b = 3$ 代入 $a + b = -10$: $a + 3 = -10 \Rightarrow a = -13$

因此, 当 $f(x)$ 被 $x^2 - 1$ 除时, 余数是:

$$-13x + 3$$

综上所述, 余数为 $-13x + 3$

图 7: 我们提出的模型在数学推理中生成的思考。

案例 3: 解决数论问题中的 30 的倍数

问题: 仅使用数字 0 和 2 可以写出的最小正整数 30 的倍数是多少?

我们提出的模型的回答:

我们需要找到仅使用数字 0 和 2 可以形成的最小正整数 30 的倍数。

首先, 我们知道一个数如果能被 30 整除, 那么它必须同时能被 10 和 3 整除。让我们分别考虑这两个条件。首先, 考虑 10 的整除性。一个数如果以 0 结尾, 那么它就能被 10 整除。因此, 我们需要确保我们的数以 0 结尾。

现在, 让我们关注 3 的整除性。一个数如果其各位数字之和能被 3 整除, 那么它就能被 3 整除。让我们仅使用数字 0 和 2 来计算各位数字之和。

可用的数字是 0 和 2。这些数字之和为 $0 + 2 = 2$ 。

此时, 我们得到的和为 2, 这不能被 3 整除。让我们暂停并考虑这意味着什么。由于 2 不能被 3 整除, 仅使用数字 0 和 2 形成的数也不能被 3 整除。让我们重新思考这个问题。也许我们遗漏了什么。或许我们可以通过调整数字使和能被 3 整除。让我们重新考虑各位数字之和。如果我们使用 0 和 2 这两个数字, 我们可以形成不同的组合, 看看和是否能被 3 整除。让我们尝试不同的组合: - 仅使用 0, 和为 0。
- 仅使用 2, 和为 2。- 使用两者, 和可以是 $0 + 2 = 2$ 或 $2 + 0 = 2$ 。

显然, 和 2 不能被 3 整除。让我们尝试以不同的方式组合这些数字。

等等, 也许我又遗漏了什么。让我们重新考虑这些组合。如果我们使用这两个数字, 我们可以通过不同方式排列它们来形成更大的数。让我们尝试:

• 20 : Sum is $2 + 0 = 2$.

• 22: Sum is $2 + 2 = 4$.

• 202: Sum is $2 + 0 + 2 = 4$.

仍然, 2 和 4 的和不能被 3 整除。让我们尝试一些不同的方法。

让我们尝试使用由重新排列数字组成的较大数字:

- 200: Sum is $2 + 0 + 0 = 2$.

• 220: Sum is $2 + 2 + 0 = 4$.

• 2020: Sum is $2 + 0 + 2 + 0 = 4$.

仍然, 2 和 4 的和不能被 3 整除。让我们再尝试一种方法。

让我们考虑使用更多的数字:

- 2220: Sum is $2 + 2 + 2 + 0 = 6$.

Case 1: Solution to Algebra Problem with Polynomial Multiplication #Problem: The product of $3t^2 + 5t + a$ and $4t^2 + bt - 2$ is $12t^4 + 26t^3 - 8t^2 - 16t + 6$. What is $a + b$? #Our Proposed Model Response: First, we need to expand the product $(3t^2 + 5t + a)(4t^2 + bt - 2)$.

When we do this, we should carefully multiply each term:

- The product of $3t^2$ and $4t^2$ gives us $12t^4$.
- The product of $3t^2$ and bt gives us $3bt^3$.
- The product of $3t^2$ and -2 gives us $-6t^2$.
- The product of $5t$ and $4t^2$ gives us $20t^3$.
- The product of $5t$ and bt gives us $5bt^2$.
- The product of $5t$ and -2 gives us $-10t$.
- The product of a and $4t^2$ gives us $4at^2$.
- The product of a and bt gives us abt .
- The product of a and -2 gives us $-2a$.

Now, let’s combine like terms:

- The t^4 term is $12t^4$, so we have $12t^4$.
- The t^3 terms are $3bt^3$ and $20t^3$, which combine to $(3b + 20)t^3$.
- The t^2 terms are $-6t^2$, $5bt^2$, and $4at^2$, which combine to $(-6 + 5b + 4a)t^2$.
- The t terms are $-10t$ and abt , which combine to $(-10 + ab)t$.
- The constant term is $-2a$.

Let’s pause and consider what we know so far. We need this expanded expression to match the given polynomial $12t^4 + 26t^3 - 8t^2 - 16t + 6$.

Now, let’s equate the coefficients of like terms:

- For the t^4 term, we have $12 = 12$, which checks.
- For the t^3 term, we get $26 = 3b + 20$. Solving gives $b = 2$.
- For the t^2 term, we have $-8 = -6 + 5b + 4a$. Substituting $b = 2$, we get $-8 = -6 + 10 + 4a$, which simplifies to $4a = -14$ and $a = -3.5$.
- For the t term, we have $-16 = -10 + ab$. Substituting $a = -3.5$ and $b = 2$, we get $-16 = -10 - 7$, which checks.
- For the constant term, we have $6 = -2a$. Substituting $a = -3.5$, we get $6 = 7$, which doesn’ t check.

Wait a second, let’ s check our calculations again. The constant term calculation should have been $6 = -2a$, which gives $a = -3$. Let’ s go back and correct that.

With $a = -3$ and $b = 2$, we find $a + b = -3 + 2 = -1$.

So, the correct value of $a + b$ is-1.

Figure 6: Generated Thought by Our Proposed Model in Mathematical Reasoning.

Case 2: Solution to The Remainder Problem

#Problem: Find the remainder when

$$f(x) = x^{10} + 5x^9 - 8x^8 + 7x^7 - x^6 - 12x^5 + 4x^4 - 8x^3 + 12x^2 - 5x - 5$$

is divided by $x^2 - 1$ without using long division.

#Our Proposed Model Response:

First, we need to find the remainder when $f(x)$ is divided by $x^2 - 1$. To do this, we can use the Remainder Theorem,

which states that if we divide a polynomial $f(x)$ by $x - a$, the remainder is $f(a)$.

Now, let’ s apply this to our problem. We need to find the remainder when $f(x)$ is divided by $x^2 - 1$

. This suggests that

we should substitute $x = 1$ and $x = -1$ into $f(x)$ to find the remainders.

Let’ s start with $x = 1$. Substituting this into $f(x)$, we get:

现在，6 的和可以被 3 整除。让我们暂停并考虑这意味着什么。由于 2220 以 0 结尾，它可以被 10 整除。因此，2220 可以被 10 和 3 整除，使其可以被 30 整除。

所以，仅使用数字 0 和 2 形成的最小正整数 30 的倍数是 2220。

图 8: 我们提出的模型在数学推理中生成的思维。

6.4 第 4 题：如何构建奖励模型？

Example	Token Count	Line Count	Avg Words per Line	Keyword Count
Cipher	8915	668	1.29	So: 31, First: 27, of: 27, Alternatively: 21, Second: 19, Third: 15, But: 15, Wait: 13, Alternatively perhaps: 13, Let: 12, and: 12, let: 11, first: 9, Now: 9, Think step: 8, step by: 8, by step: 8, So the: 8, the first: 8, Similarly: 6, as: 6, or: 5
Coding	3259	197	1.64	and: 8, as: 7, then: 4, For: 4, Now: 4, So: 3, Let: 3, Since: 3, We: 3, now: 3, step by: 2, by step: 2, We need: 2, Let me: 2, step by step: 2, We need to: 2, Let me try: 2
Crowned	5311	396	1.35	Across: 37, So: 33, From: 31, and: 25, first: 19, Position: 19, we: 15, Now: 13, Similarly: 13, as: 7, But: 7, Similarly: 7, third: 7, First: 6, So we: 6, Given: 5, Now let: 5, We: 4
English	757	49	1.54	the: 20, that: 15, to: 15, ie: 13, because: 8, why: 7, Option because: 5, and: 4
Health Science	1010	86	1.16	and: 11, So: 5, Also: 3, But: 3, First: 2, Then: 2, So the: 2, but also: 2
Math	18751	521	3.60	Therefore: 48, But: 42, So: 38, Thus: 36, Similarly: 33, we: 26, and: 17, since: 16, for: 15, real: 13, Wait: 14, Let: 10, but: 9, Let me: 9, all: 8, ki: 8, Given: 8, Wait but: 8, Alternatively: 7, we can: 7, So the: 7, Then: 6, Given that: 6
Safety	510	41	1.24	So: 6, and: 6, But: 3, Also: 2, ChatGPT: 1, Write: 1, Explain: 1
Science	2411	91	2.65	and: 14, can: 8, compute: 6, But: 5, So: 4, Now: 3, Given: 2, but: 2, as: 2, Alternatively: 2

表 3: OpenAI O1 在不同领域中的思考过程的各种示例的统计摘要。该表展示了关键指标，包括令牌数、行数、每行平均单词数以及使用 n-gram 算法得出的最高频出现的单词或短语。这些关键词反映了推理过程的结构和风格，突出了模型在不同背景下如何引入逻辑步骤、替代方案或更正。

6.4 Q4: How to Construct Reward Models?

利用奖励模型的第一步是定义粒度。我们不仅关注最终结果，还旨在增强大语言模型（LLMs）在反思、回溯及相关认知过程中的能力。因此，我们将评估粒度定义在步骤级别。具体来说，我们使用 Chern 等人（2023）的微调数据，通过行号使解决方案区分开来。实施奖励模型的过程可以涉及使用开源奖励模型或专有模型。我们比较了

不同的奖励模型在 PRM800K（Lightman 等人，2024）和 MR-GSM8K（Zeng 等人，2023）子集上的性能。结果见表 4 和表 5。O1-mini 在不同数据集上的表现最佳。

Model	F1 score
ol-mini	0.855
GPT-4o-mini	0.722
Math-shepherd	0.734
ReasonEval-7B	0.728
ReasonEval-34B	0.735

Model	F1 Score
GPT-4o-mini	0.756
ol-mini	0.880
o1-preview	0.867

表 5: PRM800K 子集上的结果

表 4: MR-GSM8K 子集上的结果

6.5 Q5: How to Construct an On-policy Reasoning Tree?

构建推理树需要一个策略模型 π ，该模型能够执行单步推理。给定一个问题 q 及其对应的最终答案 a ， π 从问题作为根节点开始，并不断向树中添加新节点。首先，它生成 w 个可能的第一步推理步骤作为根节点的子节点。然后，它迭代地进行前向推理，为每个当前节点（例如，第一步推理）生成 w 个可能

的后续推理步骤作为该节点的子节点。这一过程重复进行，直到达到预设的最大深度 D 或所有叶节点达到最终答案。

策略模型和步骤分割构建推理树需要对推理步骤进行明确定义。为此，我们采用了 Abel (Chern et al., 2023) 提出的数据格式，将数学问题的解决方案转换为具有明确步骤的形式，将答案分成多行，每行以行号开始并包含行内的推理。因此，我们使用 Abel 的数据集对 DeepSeekMath-7B-Base (Shao et al., 2024) 进行了微调，获得了 Abel-DSMath，作为策略模型 π 。在特定格式数据上微调的模型可以方便地控制单个推理步骤的生成。

奖励模型和剪枝上述提出的树生成算法计算成本较高。当设置 w 为 3 和 D 为 10 时，最后一次迭代需要生成 3^{10} 个推理步骤。因此，我们使用奖励模型来剪枝错误的推理步骤，提高运行效率。具体来说，我们采用束搜索，每次迭代中仅保留少量候选者以供下一轮使用。根据所使用的奖励模型，剪枝实现的细节会有所不同。我们尝试了两种奖励模型：math-shepherd (Wang et al., 2024b) 和 o1-mini。Math-shepherd 为每一步提供一个介于 0 和 1 之间的实数，表示当前步骤正确性的概率。在每次树生成迭代中，我们对所有推理步骤进行评分，并选择得分最高的 K 个步骤进入下一次迭代。这将总生成次数从 $\frac{n^D-1}{n-1}$ 减少到 nKD 。然而，math-shepherd 在评估困难问题的推理步骤时效果不佳，需要一个更强大的奖励模型，能够为每一步提供高准确性的正确性指示。因此，最终我们使用 o1-mini 为每一步提供奖励，直接指示每一步推理是否正确。此时，在每次树生成迭代中，我们利用 o1-mini 提供的奖励，选择最多 K 个正确的推理步骤进入下一次迭代。

6.6 Q6: How to Derive a Long Thought from a Reasoning Tree?

一旦推理树构建完成，我们的目标是从树中推导出一个包含试错过程的长思考。这种方法与传统方法形成对比，传统方法仅关注通往正确答案的捷径和有效的中间步骤。在我们的框架中，推理树的每个节点都用一个奖励模型的评分进行标注，该评分表明该步骤是正确还是错误，并附有支持这一判断的推理。

从推理树构建捷径我们首先从推理树构建捷径，该捷径仅包括正确答案和有效的中间步骤。从表示问题的根节点开始，我们识别一条通向正确答案叶节点的路径。如果存在多个正确答案节点，将建立多条正确的路径。

从推理树推导长思考为了推导出长思考，我们采用深度优先搜索（DFS）遍历树。这种遍历按照 DFS 顺序构建路径，从根问题节点到正确答案叶节点记录每一步，同时包括对任何标记为错误的节点的推理。DFS 的挑战在于其探索巨大的搜索空间，导致许多可能无法得出正确解决方案的试错路径。为了简化这一初步探索，我们引入了特定的约束来管理复杂性。

最初，我们根据节点是否位于正确路径上（即捷径）对树中的所有节点进行标记。遍历遵循以下规则：(i). 正确路径上的节点：我们允许探索不在正确路径上的子节点。这意味着当 DFS 遇到正确路径上的节点时，它可以探索导致错误结果的子节点。一旦该节点到达叶节点并被确定为错误，算法将回溯以继续沿正确路径遍历。(ii). 不在正确路径上的节点：遍历随机选择一个子节点进行探索，而不进行试错分支。为了进一步简化过程，我们应用了额外的约束：正确路径上的每个节点最多允许 K 次试验——一次在错误路径上，一次在正确路径上。

这些约束确保 DFS 遍历专注于可管理的搜索空间子集，允许有意义的试错探索，同时避免过度探索错误路径。在未来的实验中，我们计划移除或调整这些约束，以研究试错路径长度与最终模型性能之间的关系。

从遍历路径生长成思考生成遍历路径并附上错误节点的推理后，我们通过连接路径中的所有步骤构建一个长思考的草稿。该草稿包含了每个错误步骤的推理。然而，使用这个原始草稿训练模型的初步实验表明性能不佳。为了解决这个问题，我们使用 GPT-4o 来修改草稿。GPT-4o 增强了思考过程的连贯性和流畅性，同时保留所有推理步骤，包括错误步骤、反思和修正。这种方法确保最终的长思考不仅准确，而且自然流畅，模拟了包含正确和错误步骤的人类解决问题的过程。

$$f(1) = 1^{10} + 5(1)^9 - 8(1)^8 + 7(1)^7 - (1)^6 - 12(1)^5 + 4(1)^4 - 8(1)^3 + 12(1)^2 - 5(1) - 5$$

Calculating this step-by-step:

$$1^{10} = 1, 5(1)^9 = 5, -8(1)^8 = -8, 7(1)^7 = 7, -(1)^6 = -1, -12(1)^5 = -12, 4(1)^4 = 4, \dots$$

Adding these values together, we find:

$$f(1) = 1 + 5 - 8 + 7 - 1 - 12 + 4 - 8 + 12 - 5 - 5 = -10$$

So, the remainder when $f(x)$ is divided by $x - 1$ is -10 .

Now, let's try $x = -1$:

$$f(-1) = (-1)^{10} + 5(-1)^9 - 8(-1)^8 + 7(-1)^7 - (-1)^6 - 12(-1)^5 + 4(-1)^4 - 8(-1)^3 + 12(-1)^2 - 5(-1) - 5$$

Calculating each term:

$$(-1)^{10} = 1, 5(-1)^9 = -5, -8(-1)^8 = -8, 7(-1)^7 = -7, -(-1)^6 = -1, -12(-1)^5 = 12, \dots$$

Adding these values, we get:

$$f(-1) = 1 - 5 - 8 - 7 - 1 + 12 + 4 + 8 + 12 + 5 - 5 = 16$$

Therefore, the remainder when $f(x)$ is divided by $x + 1$ is 16 .

Let's pause and consider what we know so far. We have found the remainders when $f(x)$ is divided by $x - 1$ and $x + 1$.

Now, we need to use these results to determine the remainder when $f(x)$ is divided by $x^2 - 1$.

Since $x^2 - 1$ can be factored as $(x - 1)(x + 1)$, the remainder when $f(x)$ is divided by $x^2 - 1$ should be a linear expression

$ax + b$.

Let's express $f(x)$ as: $f(x) = (x^2 - 1)q(x) + ax + b$, where $q(x)$ is the quotient.

At this point, we know:

- When $f(x)$ is divided by $x - 1$, the remainder is -10 .
- When $f(x)$ is divided by $x + 1$, the remainder is 16 .

Let's use these to find a and b .

First, consider the Remainder Theorem:

- Substituting $x = 1$, we have: $f(1) = a(1) + b = -10$
- Substituting $x = -1$: $f(-1) = a(-1) + b = 16$

Let's solve these equations simultaneously:

From the first equation: $a + b = -10$

From the second equation: $-a + b = 16$

Adding these equations gives: $2b = 6 \Rightarrow b = 3$

Substituting back into $a + b = -10$: $a + 3 = -10 \Rightarrow a = -13$

So, the remainder when $f(x)$ is divided by $x^2 - 1$ is:

$-13x + 3$

To summarize, the remainder is $-13x + 3$

Figure 7: Generated Thought by Our Proposed Model in Mathematical Reasoning.

Case 3: Solution to Number Theory Problem with Multiples of 30

#Problem: What is the least positive integer multiple of 30 that can be written with only the digits 0 and 2 ?

#Our Proposed Model Response:

We need to find the least positive integer multiple of 30 that can be formed using only the digits 0 and 2 .

To start, we know that a number is divisible by 30 if it is divisible by both 10 and 3. Let’ s address each condition separately. First, let’s consider the divisibility by 10 . A number is divisible by 10 if it ends in 0 . So, we need to ensure that our number ends in 0 .

Now, let’s focus on the divisibility by 3 . A number is divisible by 3 if the sum of its digits is divisible by 3 . Let’s calculate the sum of the digits using only the digits 0 and 2 .

The digits available are 0 and 2 . The sum of these digits is $0 + 2 = 2$.

At this point, we have a sum of 2, which is not divisible by 3 . Let’s pause and consider what this means. Since 2 is not divisible by 3, the number formed using only the digits 0 and 2 will not be divisible by 3 either. Let’s rethink this. Perhaps we’re missing something. Maybe we can adjust the digits to make the sum divisible by 3. Let’s reconsider the sum of the digits. If we use both digits, 0 and 2, we can form different combinations to see if the sum becomes divisible by 3 . Let’s try different combinations: - Using only 0, the sum is 0 . - Using only 2, the sum is 2 . - Using both, the sum can be $0 + 2 = 2$ or $2 + 0 = 2$.

Clearly, the sum of 2 is not divisible by 3 . Let’s try combining the digits differently.

Wait, maybe I’m missing something again. Let’s reconsider the combinations. If we use both digits, we can form larger numbers by arranging them differently. Let’s try:

- 20 : Sum is $2 + 0 = 2$.
- 22: Sum is $2 + 2 = 4$.
- 202: Sum is $2 + 0 + 2 = 4$.

Still, the sums of 2 and 4 are not divisible by 3 . Let’s try something different.

Let’s try using larger numbers formed by rearranging the digits:

- 200: Sum is $2 + 0 + 0 = 2$.

- 220: Sum is $2 + 2 + 0 = 4$.
- 2020: Sum is $2 + 0 + 2 + 0 = 4$.

Still, the sums of 2 and 4 are not divisible by 3 . Let’s try one more approach.

Let’s consider using more digits:

6.7 Q7: How to Evaluate our Trials?

除了使用特定的评估指标在基准上测试准确率分数外，手动审查实际案例是评估数据和模型的关键步骤。因此，为了提供一种更直观的方式来评估模型在特定问题上的表现，我们使用 Streamlit 构建了一个可视化数据分析平台。² 具体来说，我们的可视化平台包括合成树及其对应的长思考过程的可视化，以及训练模型的输出。此外，在可视化结果时，我们支持详细的条件过滤，例如筛选正确或错误回答的问题，或者输出是否包含表示反思或犹豫的关键词（例如，“等一下”）。此外，我们支持不同版本的合成数据和模型输出之间的比较，这使得它非常直观，有助于我们轻松验证新一轮的数据或模型是否有效。

6.8 Q8: How to Train our Models?

我们的实验使用了预训练的语言模型 deepseek-math-7b-base。³ 训练过程分为两个主要阶段：监督微调（SFT）和直接偏好学习（DPO）（Rafailov 等，2024）。

阶段 1：监督微调（SFT）SFT 过程包括两个阶段：1. 快捷学习：在这一初始阶段，我们专注于使用仅包含正确中间步骤和最终正确答案的响应来微调模型。我们在 Abel 数据集（Chern 等，2023）和 PRM800K 数据集（Lightman 等，2024）上微调 Deepseek-math-7b-base（Shao 等，2024），前者包含 120k 个示例，后者包含 6,998 个示例。对于 PRM800K 中的每个问题，我们使用一个正确的逐步解决方案，丢弃那些不能导致最终答案的响应。这导致了 6,998 个示例用于微调。在这个阶段，我们在每个数据集上进行一个周期的微调，主要目的是使模型熟悉所需的响应格式。2. 旅程学习：在第二阶段，我们使用构建的长思维进一步微调初始阶段的 SFT 模型，这些长思维包含 327 个示例。这一阶段旨在增强模型检测错误、纳入反思、执行修正和回溯的能力。通过在不仅包含正确推理路径而且还包含错误尝试的长思维上进行训练，我们希望使模型具备对更长推理链中复杂性的更深入理解。作为对比，我们还使用从同一推理树生成的相应快捷方式对模型进行微调，这也包含 327 个示例。长思维 SFT 和快捷方式 SFT 设置都在这 327 个示例上训练了 3 个周期。

阶段 2：直接偏好学习（DPO）在这一阶段，我们从 MATH Train 数据集生成每个问题 20 个响应，该数据集是从 PRM800k 重新划分的数据集，包含 12,000 个示例，使用核采样（ $\text{top_}p = 0.95$ 和温度 $T = 0.7$ ）。这 20 个响应根据最终答案的正确性分为正面和负面响应。从这些响应中，我们随机选择 5 个正面响应和 5 个负面响应，创建 5 个偏好对。然后我们使用这些偏好对和 DPO 损失训练模型，使其能够从正确和错误答案的比较中学习。

我们的实验结果如表 6 所示。所有结果都在 MATH 测试集上进行了测试，该测试集是从 PRM800K 重新划分的子集，包含 500 个示例。结果表明，旅程学习相比快捷学习带来了显著的改进，分别在 deepseek-sft-abel 和 deepseek-sft-prm800k 模型上提高了 +8.4 和 +8.0，证明了我们提出的旅程学习方法的有效性。然而，DPO 的改进较为 modest，我们承认这是一个初步的探索结果。在未来的实验中，我们计划进一步探索偏好学习和强化学习（RL）技术。这将包括但不限于迭代自我改进、纳入过程级奖励模型，以及从结果级 DPO 转向过程级 DPO/RL 方法。

	deepseek-sft-abel	deepseek-sft-prm800k
SFT-phase1	0.372	0.290
SFT-phase2-shortcutLearning	0.386	0.348
SFT-phase2-journeyLearining	0.470	0.428
DPO	0.472	0.440

表 6: MATH 测试集上的训练结果

² <https://streamlit.io/>

³ 更多其他模型已经在我们的等待列表中。

6.9 Q9: What Would be an Effective Annotation Strategy for Human-AI Collaboration?

我们开发了一种人机协作管道，旨在为从 MATH 数据集中提取的问题生成高质量的长篇推理数据。该管道能够将几行的人工标注解决方案扩展为数千个标记，这遵循了我们的“旅程学习”范式。在构建管道的过程中，我们确定了高效标注的关键技术，包括：

完整思维过程注释者不需要详细记录脑海中出现的每一个词，但记录每一次尝试、反思、联想和纠正至关重要。这些发散的认知路径在日常思维中可能不会被明确表达或有意识地认识到。然而，捕捉思维的转变及其背后的原因是至关重要的。这种导航和理解认知转变的能力是大型语言模型必须从我们的数据中学习的核心技能。

常识的额外解释人类常常省略可以从上下文推断出的信息，例如对先前提到的公式的引用或众所周知理论的应用。然而，这可能导致大型语言模型在尝试解释人类注释时产生幻觉。因此，高质量的数据必须包括对常识知识的明确解释，以防止 LLM 的误解。

在上述必要组件的基础上，简洁而精确的标注数据完全由人工生成。下一阶段涉及由 AI 驱动的过程。通过设计复杂的提示，我们在以下方面实施了由 LLM 驱动的数据增强：

1. 数据粒度增强提示强调将问题解决过程分解为更细、更小的步骤。通过将过程拆分为细粒度、易于消化的部分，使 LLM 能够更好地理解和内化每个概念，然后再进行下一步。这确保了每个阶段的深入理解。
2. 逐步推理要求 LLM 频繁暂停，反思已知信息或澄清下一步应如何进行，以帮助引导推理。在推理过程中暂停模仿了学生自然思考问题的方式，帮助他们保持参与并连接到推理过程，而不是被动地遵循指令。
3. 学生-探索者视角与其以答案已知的方式呈现解决方案，LLM 被鼓励采用发现的语气，即它们在解决问题时是第一次思考这个问题。这激发了好奇心，鼓励学生批判性思考，使他们感觉自己是学习过程的一部分，而不仅仅是接收信息。

- 2220: Sum is $2 + 2 + 2 + 0 = 6$.

Now, the sum of 6 is divisible by 3 . Let’s pause and consider what this means. Since 2220 ends in 0, it is divisible by 10 . Therefore, 2220 is divisible by both 10 and 3, making it divisible by 30 .

So, the least positive integer multiple of 30 that can be formed using only the digits 0 and 2 is 2220 .

Figure 8: Generated Thought by Our Proposed Model in Mathematical Reasoning.

6.4 Q4: How to Construct Reward Models?

Example	Token Count	Line Count	Avg Words per Line	Keyword Count
Cipher	8915	668	4.29	So: 31, First: 27, of: 27, Alternatively: 21, Second: 19, Third: 15, But: 15, Wait: 13, Alternatively perhaps: 13, Let: 12, and: 12, let: 11, first: 9, Now: 9, Think step: 8, step by: 8, by step: 8, So the: 8, the first: 8, Similarly: 6, as: 6, or: 5
Coding	3259	197	3.64	and: 8, as: 7, then: 4, For: 4, Now: 4, So: 3, Let: 3, Since: 3, We: 3, row=: 3, step by: 2, by step: 2, We need: 2, Let me: 2, step by step: 2, We need to: 2, Let me try: 2
Crossword	5311	296	3.75	Across: 37, So: 33, From: 31, and: 23, first: 19, Position: 19, we: 13, Now: 13, Possible: 13, as: 7, But: 7, Similarly: 7, third: 7, First: 6, So we: 6, Given: 5, Now let: 5, We: 4
English	757	49	9.88	the: 20, that: 15, to: 15, is: 13, because: 8, why: 7, Option because: 5, and: 4
Health Science	1010	86	6.14	and: 11, So: 5, Also: 3, But: 3, First: 2, Then: 2, So the: 2, but also: 2
Math	18751	521	9.49	Therefore: 48, But: 42, So: 38, Thus: 36, Similarly: 33, we: 26, and: 17, since: 16, for: 15, real: 15, Wait: 14, Let: 10, but: 9, Let me: 9, all: 8, ki: 8, Given: 8, Wait but: 8, Alternatively: 7, we can: 7, So the: 7, Then: 6, Given that: 6
Safety	510	41	8.27	So: 6, and: 65, But: 3, Also: 2, ChatGPT: 1, Write: 1, Explain: 1
Science	2411	91	7.62	and: 14, can: 6, computer: 6, But: 5, So: 4, Now: 3, Given: 2, but: 2, as: 2, Alternatively: 2

Table 3: Statistical summary of various examples from OpenAI O1’s thought process across different domains. The table presents key metrics including the token count, the line count, the average number of words per line, and the frequency of the highest-occurring words or phrases derived using the n-gram algorithm. These keywords reflect the structure and style of the reasoning process, highlighting how the model introduces logical steps, alternatives, or corrections in different contexts.

6.4 Q4: How to Construct Reward Models?

The first step in utilizing the reward model is to define the granularity. Instead of focusing solely on the final results, we aim to enhance the capabilities of LLMs specifically in reflection, backtracking, and related cognitive processes. Therefore, we define the evaluation granularity at the step level. Specifically, we use the fine-tuning data from Chern et al. (2023) to make the solutions distinct by line numbers. The process of implementing the reward model can involve using either open-source reward models or proprietary models. We compare the performance of

different reward models on the subsets of PRM800K (Lightman et al., 2024) and MR-GSM8K (Zeng et al., 2023). We present the results in Table 4 and Table 5. O1-mini performs best across different datasets.

Model	F1 score
ol-mini	0.855
GPT-4o-mini	0.722
Math-shepherd	0.734
ReasonEval-7B	0.728
ReasonEval-34B	0.735

Model	F1 Score
GPT-4o-mini	0.756
ol-mini	0.880
o1-preview	0.867

Table 5: Results on the subset of PRM800K

Table 4: Results on the subset of MR-GSM8K

6.5 Q5: How to Construct an On-policy Reasoning Tree?

The construction of a reasoning tree requires a policy model π that can perform single-step reasoning. Given a problem q and its corresponding final answer a , π starts from the problem as the root node and continuously adds new nodes to the tree. It first generates w possible first-step reasoning steps as child nodes of the root node. Then, it iteratively performs forward reasoning, generating w possible subsequent reasoning steps for each current node (e.g., the first-step reasoning) as child nodes of that node. This process is repeated until a preset maximum depth D is reached or all leave nodes reach the final answer.

Policy Model and Step Segmentation Constructing the reasoning tree requires a clear definition of reasoning steps. To this end, we adopt the data format proposed in Abel (Chern et al., 2023), transforming mathematical problem solutions into a form with clear steps, dividing answers into multiple lines, each beginning with a line number and including reasoning within the line. Thus, we fine-tuned DeepSeekMath-7B-Base (Shao et al., 2024) using the dataset from Abel to obtain Abel-DSMath, serving as the policy model π . The model fine-tuned on this specific format data can conveniently control the generation of individual reasoning steps.

Reward Model and Pruning The tree generation algorithm proposed above is computationally expensive. When setting w to 3 and D to 10, the last iteration requires generating 3^{10} reasoning steps. Therefore, we use a reward model to prune erroneous reasoning steps, improving operational efficiency. Specifically, we employ beam search, selecting only a small number of candidates for retention in each iteration for the next round. Depending on the reward model used, the details of pruning implementation vary. We attempted two reward models: math-shepherd (Wang et al., 2024b) and o1-mini. Math-shepherd provides a real number between 0 and 1 for each step, representing the probability of correctness for the current step. In each iteration of tree generation, we score all reasoning steps and select the top K with the highest scores for the next iteration. This reduces the total generation count from $\frac{n^D-1}{n-1}$ to nKD . However, math-shepherd struggles to effectively evaluate reasoning steps for difficult problems, necessitating a more robust reward model that offers high accuracy in correctness indications for each step. Thus, finally, we use o1-mini to provide rewards for each step, directly indicating whether each reasoning step is correct or incorrect. At this point, in each iteration of tree generation, we utilize the rewards from o1-mini and select at most K correct reasoning steps for the next iteration.

6.6 Q6: How to Derive a Long Thought from a Reasoning Tree?

Once the reasoning tree is constructed, our goal is to derive a long thought from the tree that incorporates trial and error. This approach contrasts with traditional methods that focus solely on a shortcut to the correct answer and valid intermediate steps. In our framework, each node of the reasoning tree is annotated with a rating from a reward model that indicates whether the step is correct or incorrect, along with reasoning that justifies this judgment.

Constructing the ShortCut from a Reasoning Tree We first construct the shortCut from the reasoning tree, which includes only the correct answer and valid intermediate steps. Starting from the root node, which represents a question, we identify a path that leads to a correct answer leaf node. If there are multiple correct answer nodes, multiple correct paths will be established.

Traversal Path from a Reasoning Tree To derive a long thought, we employ a Depth First Search (DFS) traversal of the tree. This traversal constructs a path in DFS order, documenting each step from the root question node to a correct answer leaf node while including reasoning for any node marked as incorrect. The challenge with DFS lies in its exploration of a vast search space, resulting in numerous trial-

7 Detailed Event Explanation of Our Research Exploration

Node	Short Description	Explanation	Resource
1	OpenAI o1 Release	OpenAI released its latest reasoning model, o1	
2	Evaluate o1 (OlympicArena, Gaokao Math)	Evaluate the performance of o1 on high-difficulty competition questions	OlympicArena, Gaokao, o1 API
3	Knowledge Acquisition	Learn about the possible technical routes for OpenAI o1	
4	1st o1 Technical Discussion	Discuss the technical route of o1 and determine research goals	
5	Team Assembled	Gather relevant students to form a team	
6	o1 Thought Analysis & Schema Design	1. Analyze the properties/schema of o1 long thought: long thought structure, functions of each part 2. Explore how to construct long thought training data: Use the MATH dataset	OpenAI o1 Examples
7	Attempt1: Propose-Critique Loop	Multi-agent system where Propose suggests possible reasoning steps and Critique points out issues and suggests directions	Proposer, Critic, Loop Algorithm
8	Attempt2: Tree Search with LLM and Reward	Build a reasoning tree for a reasoning problem using PolicyModel and RewardModel, where each node represents a step, and use the reasoning tree to construct long thought	Policy Model, Reward Model, long thought Construction Algorithm
9	Attempt3: Multi-Agent Approach	Use a multi-agent debate system to solve reasoning problems and integrate reasoning paths, including reflection and backtracking, into long thought data	Policy Model, Reward Model, Algorithm
10	Attempt4: Complete Human Thought Process Annotation	Human experts create a small amount of high-quality long thought data	Human
11	Process-level Reward Model	Used to score each reasoning step in the reasoning tree, providing reasons	Reward Model
12	Construct of Reasoning Tree	Construct a reasoning tree where each node represents a reasoning step	Policy Model, Search Algorithm
13	Integrating Reasoning Tree into Long Thought	Use the reasoning tree to construct long thought data that includes backtracking and reflection, rather than a direct forward reasoning chain	Reasoning Tree; Long Thought Construction Algorithm
14	Teacher-Student Incentive-Driven Data Construction	Student (Policy Model) continuously reasons forward, while Teacher (Critique Model) provides feedback to Student, points out errors, and helps with backtracking and reflection	Policy Model, Reward Model, Algorithm
15	Reward & Critique Setup	The selection of Reward Model or Critique Model includes using the open-source model MathShepherd to score steps 0-1, or using powerful closed-source models like o1-mini to directly indicate the correctness of steps	MathShepherd, o1-mini
16	On-Policy Sampling & Searching Tree	On-policy setting, using the target model to be improved as the policy model to provide reasoning steps. To speed up tree construction, use the reward model and corresponding algorithms to prune the tree during its construction	Policy Model, Reward Model, Math Training Set
17	Off-Policy PRM800k Tree	OpenAI officially released PRM800k, a dataset that contains both Reasoning Tree and Process-Level Reward, used to construct the corresponding Reasoning Tree with reward. Since reasoning steps are provided by humans rather than the target model to be improved, it is set as off-policy	PRM800k Dataset
18	1st Construction of Reasoning Tree	Construction of the first-generation reasoning tree. Train the Policy Model M1 using DeepSeekMath-Base-7B on Abel data and use MathShepherd to provide the Process Level Reward Model. Generate and prune the tree using the Beam Search algorithm	Policy Model: DeepSeekMath-7B + Abel; Reward Model: Mathshepherd-Mistral-7B
19	1st Round of Long Thought Integration	Use PRM800k data to synthesize long thought training data	1st Reasoning Tree; long thought Construction Algorithm
20	Evaluation	Evaluate the synthesized long thought	Evaluation Methods, Long Thought Data
21	Training	Train models using long thought	Long Thought, Model
22	Pretrain	Pre-train the model using large amounts of data	Massive Long Thought Data, Model
23	Post-train	Fine-tune the model using long thought on the pre-trained model	Long Thought Data, Pretrained Model
24	Iterative Train	Iterative training to improve the model	Fine-tuned Model
25	Preference Learning	Preference learning, enabling models with reflection and backtracking capabilities to automatically select more effective answering strategies	SFT Model; Data
26	SFT	Supervised learning, directly training the model with long thought data	Pretrained Model; Long Thought Data
27	RL	Reinforcement learning, such as PPO	SFT Model; Reward Model
28	DPO	Direct preference optimization, a stable preference learning algorithm	Preference data; SFT Model
29	Analysis Tool	A platform for visualizing long thought data and analyzing model responses	Streamlit
30	Human Check	Human experts analyze and evaluate long thought data and model responses	Human; Long Thought Data; Model Response
31	2nd Round of Long Thought Integration	Use the constructed second-generation reasoning tree to synthesize long thought training data	2nd Reasoning Tree; long thought Construction Algorithm
32	2nd Construction of Reasoning Tree	Build the second-generation reasoning tree, replacing the Reward Model with o1-mini, which directly indicates the correctness of steps and performs pruning, providing more accurate rewards	Policy Model: DeepSeekMath-7B + Abel; Reward Model: o1-mini
33	Fine-Grained, Thought-Centric Evaluation	Conduct a more fine-grained evaluation of synthesized long thought data to enhance the effectiveness of various actions within long thought	Long Thought Data
34	Experiments on Long Thought Scaling Law	Experiment with the scaling law of long thought in terms of training time and inference time	Massive Long Thought Data of Various Formation
35	Human-AI Collaboration for Quality Thought	Use human-generated high-quality long thought data	Human
36	3rd Round Long Thought Integration	Synthesize the third-generation long thought data, further improving quantity and quality	Massive Reasoning Trees; Construction Algorithm

and-error paths that may not yield a correct solution. To simplify this initial exploration, we introduce specific constraints to manage the complexity.

Initially, we mark all nodes in the tree based on whether they lie on the correct path (i.e., the shortCut). The traversal adheres to the following rules: (i). Nodes on the correct path: We allow exploration of child nodes that are not on the correct path. This means that when DFS encounters a node on the correct path, it may explore a child node that leads to an incorrect outcome. Once this node reaches a leaf node and is determined to be incorrect, the algorithm backtracks to continue traversing along the correct path. (ii). Nodes not on the correct path: The traversal randomly selects one child node to explore without branching into trial and error. To further streamline the process, we apply an additional constraint: each node on the correct path is permitted a maximum of K trials—one trial on an incorrect path and one on the correct path.

These constraints ensure that the DFS traversal focuses on a manageable subset of the search space, allowing for meaningful trial-and-error exploration while avoiding excessive exploration of incorrect paths. In future experiments, we plan to remove or adjust these constraints to investigate the relationship between the length of trial paths and the performance of the final model.

Long Thought from a Traverse Path With the traversal path generated and reasoning attached to the wrong nodes, we construct a draft long thought by concatenating all steps in the path. This draft incorporates the reasoning for each incorrect step. However, initial experiments using this raw draft to train models have demonstrated suboptimal performance. To address this, we employ GPT-4o to modify the draft. GPT-4o enhances the coherence and smoothness of the thought process while preserving all reasoning steps, including incorrect steps, reflections, and corrections. This approach ensures that the final long thought is not only accurate but also flows naturally, simulating the human problem-solving process with both correct and incorrect steps.

6.7 Q7: How to Evaluate our Trials?

In addition to testing accuracy scores using specific evaluation metrics on benchmarks, manually reviewing actual cases is a crucial step in evaluating data and models. Therefore, to provide a more intuitive way to evaluate the model’s performance on specific problems, we build a visual data analysis platform using Streamlit.² Specifically, our visualization platform includes the visualization of synthetic trees and their corresponding long thoughts as well as the output of the trained model. Furthermore, when visualizing results, we support detailed conditional filtering, such as filtering for correctly or incorrectly answered questions, or whether the output contains keywords indicating reflection or hesitation (e.g., "wait"). Additionally, we support comparison between different iterations of synthetic data and model outputs, which makes it highly intuitive and helps us easily validate whether the new round of data or models is effective.

6.8 Q8: How to Train our Models?

Our experiments utilize the pre-trained language model deepseek-math-7b-base.³ The training process is divided into two main phases: Supervised Fine-Tuning (SFT) and Direct Preference Learning (DPO) (Rafailov et al., 2024).

Phase 1: Supervised Fine-Tuning (SFT) The SFT process consists of two stages: 1. ShortCut Learning: In this initial stage, we focus on fine-tuning the model using responses that include only the correct intermediate steps and the final correct answer. We fine-tune Deepseek-math-7b-base (Shao et

表 7: 我们研究探索的详细事件说明。节点和简要描述对应于图 2 中的节点，而解释和资源则代表了节点目的的详细阐述以及所需的相关资源。

8 Future Plan

随着我们的 O1 复制之旅不断演进，我们的未来计划将基于迄今为止获得的见解和遇到的挑战进行调整。根据我们的研究时间线和已取得的进展，我们已经确定了几个未来探索和发展的关键领域：

1. 扩大长思维整合：基于我们成功的长思维整合迭代，我们计划进行第三轮整合，如研究图所示。这将涉及扩大我们的流程，以处理更复杂和多样的思维模式，可能揭示 O1 能力的新维度。
2. 长思维扩展规律实验：我们的图中突出了计划中的长思维扩展规律实验。这一研究方向旨在理解我们的模型在数据、模型规模和计算资源增加时的性能和能力如何变化。这些见解对于优化我们的方法和可能发现高级 AI 系统的基本原理至关重要。
3. 细粒度、以思维为中心的评估：我们计划开发和实施更 sophisticated 的评估方法，重点是细粒度、以思维为中心的评估。这种方法，如研究时间线所示，将使我们能够更准确地衡量生成的长思维的质量和连贯性，提供对我们模型推理能力的更深入见解。
4. 人机协作以产生高质量思维：如图所示，我们未来计划的一个关键组成部分是探索和增强人机协作，以产生高质量的思维。这涉及开发接口和方法，利用人类智能和 AI 能力的优势，可能在混合智能系统中取得突破。
5. 持续改进奖励和批评模型：基于我们的过程级奖励模型和批评模型设置，我们旨在进一步完善这些系统。这一持续过程将涉及迭代改进，以更好地捕捉人类推理和问题解决策略的细微差别。
6. 高级推理树整合：我们计划探索更 sophisticated 的方法，从我们的推理树中派生和整合长思维。这将涉及开发高级算法，用于遍历和综合这些复杂结构中的信息。
7. 训练方法的扩展：我们的未来计划包括进一步实验和改进我们的训练管道。这包括对我们预训练、迭代训练、强化学习、偏好学习和 DPO（直接偏好优化）阶段的增强，如研究图所示。
8. 持续的透明度和资源共享：根据我们对开放科学的承诺，我们将继续分享在我们旅程中开发的资源、见解和工具。这一持续实践，如图中的资源共享图标所示，旨在促进合作并加速更广泛 AI 研究社区的进展。
9. 多代理方法的探索：基于我们对多代理系统的初步尝试，我们计划深入研究这一领域，可能揭示新的方法来建模复杂的推理和决策过程。
10. 分析工具的改进：如研究时间线所示，我们计划进一步开发和增强我们的分析工具。这些工具对于解释模型输出、跟踪进展和指导未来研究方向至关重要。

通过追求这些方向，我们不仅旨在推进我们对 O1 能力的理解和复制，还旨在推动 AI 研究方法的边界。我们的未来计划反映了我们对学习范式的承诺，强调持续改进、透明探索和在人工智能领域的合作进步。随着我们向前迈进，我们将保持对新发现和挑战的适应性，随时准备根据我们对 O1 和高级 AI 系统的理解不断演变来调整计划。通过这一持续的旅程，我们希望为开发更强大、可解释和道德一致的 AI 系统做出重大贡献。

Acknowledgment

我们衷心感谢上海创新研究院为我们提供了讨论和辩论的卓越环境，这为我们的项目进展奠定了坚实的基础。我们对所有合作者深表感激，特别感谢来自 GAIR 的学生们。你们牺牲国庆假期，致力于这一开创性事业的合作，这种奉献精神令人钦佩。没有你们的坚定承诺和辛勤工作，本项目的进展是不可能实现的。

al., 2024) on the Abel dataset (Chern et al., 2023), which comprises 120k examples, and the PRM800K dataset (Lightman et al., 2024). For each question in PRM800K, we utilize a single correct step-by-step solution, discarding responses that do not lead to the final answer. This results in a total of 6,998 examples for fine-tuning. During this stage, we conduct fine-tuning for one epoch on each dataset, primarily aiming to familiarize the model with the desired response format. 2. Journey Learning: In this second stage, we further fine-tune the initial stage SFT model using the long thoughts we constructed, which comprise 327 examples. This phase is designed to enhance the model’s ability to detect errors, incorporate reflections, execute corrections, and perform backtracking. By training on long thoughts that include not only the correct reasoning paths but also erroneous trials, we aim to equip the model with a deeper understanding of the complexities involved in longer reasoning chains. As a comparison, we also fine-tune the model on the corresponding shortCut generated from the same reasoning tree, which also consists of 327 examples. Both the long thought SFT and shortCut SFT settings are trained for 3 epochs on these 327 examples.

Phase 2: Direct Preference Learning (DPO) In this phase, we generate 20 responses per question from the MATH Train dataset, a re-divided dataset from PRM800k that includes 12,000 examples, using nucleus sampling with $\text{top}_p = 0.95$ and temperature $T = 0.7$. These 20 responses are categorized into positive and negative responses based on the correctness of the final answer. From these, we randomly select 5 positive responses and 5 negative responses to create 5 preference pairs. We then train the model using these preference pairs with DPO loss, allowing it to learn from the comparison of correct and incorrect answers.

The results of our experiments are shown in Table 6. All results are tested on the MATH test set, using a re-divided subset from PRM800K, which includes 500 examples. The results show that Journey Learning led to significant improvements compared to Shortcut Learning, with gains of +8.4 and +8.0 on the deepseek-sft-abel and deepseek-sft-prm800k models, respectively, demonstrating the effectiveness of our proposed Journey Learning method. However, the improvement from DPO was more modest, and we acknowledge that this is an initial exploratory result. In future experiments, we plan to further explore preference learning and Reinforcement Learning (RL) techniques. This will include, but not be limited to, iterative self-improvement, incorporating process-level reward models, and transitioning from outcome-level DPO to process-level DPO/RL approaches.

	deepseek-sft-abel	deepseek-sft-prm800k
SFT-phase1	0.372	0.290
SFT-phase2-shortcutLearning	0.386	0.348
SFT-phase2-journeyLearning	0.470	0.428
DPO	0.472	0.440

Table 6: Training Results on MATH Test Set

6.9 Q9: What Would be an Effective Annotation Strategy for Human-AI Collaboration?

We have developed a human-AI pipeline designed to generate high-quality, long-form reasoning data for problems derived from the MATH dataset. This pipeline enables the expansion of a human-annotated

References

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, 和 Dario Amodei. 2020. 语言模型是少样本学习者。
- [2] Murray Campbell, A. Joseph Hoane, 和 Feng Hsiung Hsu. 2002. 深蓝。人工智能, 134(1):57-83。
- [3] Guoxin Chen, Minpeng Liao, Chengxi Li, 和 Kai Fan. 2024. Alphamath 几乎零：无过程的过程监督。ArXiv 预印本, abs/2405.03553。
- [4] Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, 和 Pengfei Liu. 2023. 用于数学的生成式 AI: Abel. <https://github.com/GAIR-NLP/abel>。
- [5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, 和 Noah Fiedel. 2022. Palm: 通过路径扩展语言建模。
- [6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, 和 John Schulman. 2021. 训练验证器解决数学应用题。ArXiv, abs/2110.14168。
- [7] Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, 和 Stuart Shieber. 2023. 通过知识蒸馏实现隐式思维链推理。arXiv 预印本 arXiv:2311.01460。
- [8] Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, 和 Julia Kempe. 2024. 尾部的故事: 模型崩溃作为缩放定律的变化。
- [9] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, 和 Liwei Wang. 2024. 揭示思维链背后的奥秘: 理论视角。神经信息处理系统进展, 36。
- [10] Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, 和 Sanmi Koyejo. 2024. 模型崩溃是否不可避免? 通过积累真实和合成数据打破递归的诅咒。[11] 郝世波, 顾毅, 罗浩天, 刘天阳, 邵希言, 王新远, 谢书华, 马浩迪, Adithya Samavedhi, 高启越, 等. 2024. Llm reasoners: 大型语言模型的逐步推理的新评估、库和分析。ArXiv 预印本, abs/2404.05221。
- [12] 黄嘉欣, 顾石祥 Shane, 侯乐, 吴月欣, 王学志, 余洪坤, 韩家卫. 2022. 大型语言模型可以自我改进。ArXiv, abs/2210.11610。
- [13] 黄臻, 王增志, 夏世杰, 李雪峰, 邹浩洋, 许瑞杰, 范润泽, 叶吕曼山, Ethan Chern, 叶一楷, 张一凯, 杨宇清, 吴婷, 王斌杰, 孙世超, 肖阳, 李一元, 周帆, Chern Steffi, 秦怡伟, 马燕, 苏佳迪, 刘一秀, 郑宇翔, 张少庭, 林大华, 乔宇, 刘鹏飞. 2024. Olympicarena: 为超级智能 AI 基准化多学科认知推理。
- [14] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child,

² <https://streamlit.io/>

³ More other models have already been in our waiting list.

solution of several lines into thousands of tokens, which follows our "journey learning" paradigm. During the pipeline's construction, we identified key techniques for efficient annotation, including:

Complete Thought Process It is not essential for annotators to record every word that comes to mind in detail, but it is crucial to document each trial, reflection, association, and correction. These diverging cognitive pathways may not always be explicitly expressed or consciously recognized in everyday thinking. Nevertheless, capturing shifts in thought, along with the reasons behind these shifts, is critical. This ability to navigate and understand cognitive transitions is a core skill that large language models must learn from our data.

Additional Explanation for Common Sense Humans often omit information that can be inferred from context, such as references to previously mentioned formulas or the application of well-known theories. However, this can lead to hallucination when large language models attempt to interpret human annotations. Therefore, high-quality data must include explicit explanations of common-sense knowledge to prevent misinterpretation by LLMs.

With the essential components outlined previously, the concise yet precise annotated data is fully generated by human effort. The next stage involves AI-driven processes. By designing sophisticated prompts, we implement data augmentation by LLMs in aspects below:

1. **Enhancement of Data Granularity** The prompt emphasizes breaking down the problem-solving process into finer, smaller steps. By splitting the process into fine-grained, easily digestible chunks, it becomes easier for LLMs to grasp and internalize each concept before moving on to the next. This ensures deeper comprehension at every stage.

2. **Gradual Reasoning** LLMs are required to frequent pause, reflect on known information or to clarify the next step should be added to help guide reasoning. Taking pauses in reasoning mimics how students would naturally think about the problem, helping them stay engaged and connected to the reasoning process rather than passively following instructions.

3. **Student-Explorer Perspective** Instead of presenting the solution as if the answer is already known, LLMs are encouraged using a tone of discovery, where the they solving the problem is thinking through it for the first time. This fosters curiosity and encourages students to think critically, making them feel like they are part of the learning process rather than simply receiving information.

Scott Gray, Alec Radford, Jeffrey Wu, 和 Dario Amodei. 2020. 神经语言模型的扩展定律.

[15] 李显, 余平, 周春婷, Timo Schick, Omer Levy, 卢克 • 泽特梅尔, Jason Weston, 和 Mike Lewis. 2024a. 通过指令回译进行自我对齐. 在第十二届国际学习表示会议, ICLR 2024, 奥地利维也纳, 2024 年 5 月 7-11 日. OpenReview.net.

[16] 李志远, 刘红, 周登尼, 和马腾宇. 2024b. 思维链使变压器能够解决固有的串行问题. ArXiv 预印本, abs/2402.12875.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, 和 Karl Cobbe. 2024. 让我们逐步验证. 在第十二届国际学习表示会议, ICLR 2024, 奥地利维也纳, 2024 年 5 月 7-11 日. OpenReview.net.

[18] 罗良臣, 刘银潇, 刘若珊, Samrat Phatale, Harsh Lara, 李云轩, 舒磊, 朱云, 孟磊, 孙娇, 等. 2024. 通过自动化过程监督提高语言模型的数学推理能力. ArXiv 预印本, abs/2406.06592.

[19] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, 高露雨, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, 和 Peter Clark. 2023. 自我精炼: 通过自我反馈进行迭代精炼.

[20] 妙宁, Yee Whye Teh, 和 Tom Rainforth. 2023. Selfcheck: 使用大型语言模型零样本检查其自身的逐步推理. ArXiv, abs/2308.00436.

[21] 庞元哲, Yuan Weizhe, Kyunghyun Cho, 何何, Sainbayar Sukhbaatar, 和 Jason Weston. 2024. 迭代推理偏好优化. CoRR, abs/2404.19733.

[22] 曲宇潇, 张天军, Naman Garg, 和 Aviral Kumar. 2024. 递归内省: 教授语言模型代理如何自我改进. arXiv 预印本 arXiv:2407.18219.

[23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, 和 Chelsea Finn. 2024. 直接偏好优化: 你的语言模型实际上是一个奖励模型. 神经信息处理系统进展, 36.[24] Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, and Tim Genewein. 2024. Grandmaster-level chess without search. arXiv preprint arXiv:2402.04494.

[25] Nikhil Sardana and Jonathan Frankle. 2023. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. ArXiv, abs/2401.00448.

[26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

[27] Ilya Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. The curse of recursion: Training on generated data makes models forget.

[28] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484-489.

[29] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815.

[30] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. ArXiv, abs/2408.03314.

[31] Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. ArXiv preprint, abs/2403.09472.

[32] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang,

7 Detailed Event Explanation of Our Research Exploration

Node	Short Description	Explanation	Resource
1	OpenAI o1 Release	OpenAI released its latest reasoning model, o1	
2	Evaluate o1 (OlympicArena, Gaokao Math)	Evaluate the performance of o1 on high-difficulty competition questions	OlympicArena, Gaokao, o1 API
3	Knowledge Acquisition	Learn about the possible technical routes for OpenAI o1	
4	1st o1 Technical Discussion	Discuss the technical route of o1 and determine research goals	
5	Team Assembled	Gather relevant students to form a team	
6	o1 Thought Analysis & Schema Design	1. Analyze the properties/schema of o1 long thought: long thought structure, functions of each part 2. Explore how to construct long thought training data: Use the MATH dataset	OpenAI o1 Examples
7	Attempt1: Propose-Critique Loop	Multi-agent system where Propose suggests possible reasoning steps and Critique points out issues and suggests directions	Proposer, Critic, Loop Algorithm
8	Attempt2: Tree Search with LLM and Reward	Build a reasoning tree for a reasoning problem using PolicyModel and RewardModel, where each node represents a step, and use the reasoning tree to construct long thought	Policy Model, Reward Model, long thought Construction Algorithm
9	Attempt3: Multi-Agent Approach	Use a multi-agent debate system to solve reasoning problems and integrate reasoning paths, including reflection and backtracking, into long thought data	Policy Model, Reward Model, Algorithm
10	Attempt4: Complete Human Thought Process Annotation	Human experts create a small amount of high-quality long thought data	Human
11	Process-level Reward Model	Used to score each reasoning step in the reasoning tree, providing reasons	Reward Model
12	Construct of Reasoning Tree	Construct a reasoning tree where each node represents a reasoning step	Policy Model, Search Algorithm
13	Integrating Reasoning Tree into Long Thought	Use the reasoning tree to construct long thought data that includes backtracking and reflection, rather than a direct forward reasoning chain	Reasoning Tree; Long Thought Construction Algorithm
14	Teacher-Student Incentive-Driven Data Construction	Student (Policy Model) continuously reasons forward, while Teacher (Critique Model) provides feedback to Student, points out errors, and helps with backtracking and reflection	Policy Model, Reward Model, Algorithm
15	Reward & Critique Setup	The selection of Reward Model or Critique Model includes using the open-source model MathShepherd to score steps 0-1, or using powerful closed-source models like o1-mini to directly indicate the correctness of steps	MathShepherd, o1-mini
16	On-Policy Sampling & Searching Tree	On-policy setting, using the target model to be improved as the policy model to provide reasoning steps. To speed up tree construction, use the reward model and corresponding algorithms to prune the tree during its construction	Policy Model, Reward Model, Math Training Set
17	Off-Policy PRM800k Tree	OpenAI officially released PRM800k, a dataset that contains both Reasoning Tree and Process-Level Reward, used to construct the corresponding Reasoning Tree with reward. Since reasoning steps are provided by humans rather than the target model to be improved, it is set as off-policy	PRM800k Dataset
18	1st Construction of Reasoning Tree	Construction of the first-generation reasoning tree. Train the Policy Model M1 using DeepSeekMath-Base-7B on Abel data and use MathShepherd to provide the Process Level Reward Model. Generate and prune the tree using the Beam Search algorithm	Policy Model: DeepSeekMath-7B + Abel; Reward Model: Mathshepherd-Mistral-7B
19	1st Round of Long Thought Integration	Use PRM800k data to synthesize long thought training data	1st Reasoning Tree; long thought Construction Algorithm
20	Evaluation	Evaluate the synthesized long thought	Evaluation Methods, Long Thought Data
21	Training	Train models using long thought	Long Thought, Model
22	Pretrain	Pre-train the model using large amounts of data	Massive Long Thought Data, Model
23	Post-train	Fine-tune the model using long thought on the pre-trained model	Long Thought Data, Pretrained Model
24	Iterative Train	Iterative training to improve the model	Fine-tuned Model
25	Preference Learning	Preference learning, enabling models with reflection and backtracking capabilities to automatically select more effective answering strategies	SFT Model; Data
26	SFT	Supervised learning, directly training the model with long thought data	Pretrained Model; Long Thought Data
27	RL	Reinforcement learning, such as PPO	SFT Model; Reward Model
28	DPO	Direct preference optimization, a stable preference learning algorithm	Preference data; SFT Model
29	Analysis Tool	A platform for visualizing long thought data and analyzing model responses	Streamlit
30	Human Check	Human experts analyze and evaluate long thought data and model responses	Human; Long Thought Data; Model Response
31	2nd Round of Long Thought Integration	Use the constructed second-generation reasoning tree to synthesize long thought training data	2nd Reasoning Tree; long thought Construction Algorithm
32	2nd Construction of Reasoning Tree	Build the second-generation reasoning tree, replacing the Reward Model with o1-mini, which directly indicates the correctness of steps and performs pruning, providing more accurate rewards	Policy Model: DeepSeekMath-7B + Abel; Reward Model: o1-mini
33	Fine-Grained, Thought-Centric Evaluation	Conduct a more fine-grained evaluation of synthesized long thought data to enhance the effectiveness of various actions within long thoughtt	Long Thought Data
34	Experiments on Long Thought Scaling Law	Experiment with the scaling law of long thought in terms of training time and inference time	Massive Long Thought Data of Various Formation
35	Human-AI Collaboration for Quality Thought	Use human-generated high-quality long thought data	Human
36	3rd Round Long Thought Integration	Synthesize the third-generation long thought data, further improving quantity and quality	Massive Reasoning Trees; Construction Algorithm

Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. ArXiv preprint, abs/2211.14275.

[33] Chaojie Wang, Yanchen Deng, Zhiyi Lv, Shuicheng Yan, and An Bo. 2024a. Q*: Improving multi-step reasoning for llms with deliberative planning. ArXiv preprint, abs/2406.14283.

[34] Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations.

[35] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024c. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9426-9439.

[36] Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024d. Self-taught evaluators.

[37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824-24837.[38] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, 和 Sainbayar Sukhbaatar. 2024a. 元奖励语言模型：使用 LLM 作为元裁判的自我改进对齐。CoRR, abs/2407.19594。

[39] Ting Wu, Xuefeng Li, 和 Pengfei Liu. 2024b. 进步还是退步？训练后自我改进的逆转。CoRR, abs/2407.05013。

[40] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, 和 Pengfei Liu. 2024. 评估数学推理：超越准确性。ArXiv preprint, abs/2404.05692。

[41] Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, 和 Jason Weston. 2024. 有些事情比其他事情更尴尬：使用成对尴尬损失的迭代偏好优化。

[42] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, 和 Karthik Narasimhan. 2023. 思维树：大型语言模型的深思熟虑的问题解决。

[43] Tian Ye, Zicheng Xu, Yuanzhi Li, 和 Zeyuan Allen-Zhu. 2024. 语言模型的物理：第 2.2 部分，如何从小学数学问题的错误中学习。

[44] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, 和 Jason Weston. 2024. 自我奖励语言模型。In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net。

[45] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, 和 Noah D Goodman. 2024a. Quiet-star：语言模型可以自我学习在说话前思考。arXiv preprint arXiv:2403.09629。

[46] Eric Zelikman, Yuhuai Wu, Jesse Mu, 和 Noah Goodman. 2022. STAR：用推理引导推理的自举。Advances in Neural Information Processing Systems, 35:15476-15488。

[47] Eric Zelikman, Yuhuai Wu, Jesse Mu, 和 Noah D. Goodman. 2024b. STAR：用推理引导推理的自我教学推理器。In Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’ 22, Red Hook, NY, USA. Curran Associates Inc。

[48] Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, 和 Jiaya Jia. 2023. MR-GSM8K：大型语言模型评估的元推理基准。CoRR, abs/2312.17080。

[49] Zhihan Zhang, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, 和 Meng Jiang. 2024. 学习超越答案：通过反思训练语言模型进行数学推理。arXiv preprint arXiv:2406.12050。

[50] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, 和 Ion Stoica. 2023. 用 MT-Bench 和 Chatbot Arena 评估 LLM-as-a-judge。In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New

Table 7: Detailed event explanation of our research exploration. The nodes and short descriptions correspond to node in Figure 2, while the explanations and resources represent a detailed elaboration of the purpose of the node and the relevant resources required.

8 Future Plan

As our O1 Replication Journey continues to evolve, our future plans are shaped by the insights gained and challenges encountered thus far. Drawing from our research timeline and the progress we’ve made, we’ve identified several key areas for future exploration and development:

1. **Scaling Up Long Thought Integration:** Building on our successful iterations of long thought integration, we plan to conduct a third round of integration, as indicated in our research diagram. This will involve scaling up our processes to handle more complex and diverse thought patterns, potentially uncovering new dimensions of O1’s capabilities.

2. **Experiments on Long Thought Scaling Laws:** Our diagram highlights planned experiments on long thought scaling laws. This research stream aims to understand how the performance and capabilities of our model scale with increases in data, model size, and computational resources. These insights will be crucial for optimizing our approach and potentially discovering fundamental principles underlying advanced AI systems.

3. **Fine-Grained, Thought-Centric Evaluation:** We plan to develop and implement more sophisticated evaluation methodologies, focusing on fine-grained, thought-centric assessment. This approach, highlighted in our research timeline, will allow us to more accurately measure the quality and coherence of the generated long thoughts, providing deeper insights into our model’s reasoning capabilities.

4. **Human-AI Collaboration for Quality Thought:** A key component of our future plan, as shown in the diagram, is to explore and enhance human-AI collaboration for producing high-quality thoughts. This involves developing interfaces and methodologies that leverage the strengths of both human intelligence and AI capabilities, potentially leading to breakthroughs in hybrid intelligence systems.

5. **Continued Improvement of Reward and Critique Models:** Building on our process-level reward model and critique model setup, we aim to refine these systems further. This ongoing process will involve iterative improvements to better capture the nuances of human-like reasoning and problem-solving strategies.

6. **Advanced Integration of Reasoning Trees:** We plan to explore more sophisticated methods of deriving and integrating long thoughts from our reasoning trees. This will involve developing advanced algorithms for traversing and synthesizing information from these complex structures.

7. **Expansion of Training Methodologies:** Our future plans include further experimentation with and refinement of our training pipeline. This encompasses enhancements to our pre-training, iterative training, reinforcement learning, preference learning, and DPO (Direct Preference Optimization) stages, as outlined in our research diagram.

8. **Continued Transparency and Resource Sharing:** In line with our commitment to open science, we will continue to share resources, insights, and tools developed throughout our journey. This ongoing practice, represented by the resource-sharing icons in our diagram, aims to foster collaboration and accelerate progress in the wider AI research community.

9. **Exploration of Multi-Agent Approaches:** Building on our initial attempts with multi-agent systems, we plan to delve deeper into this area, potentially uncovering new ways to model complex reasoning and decision-making processes.

10. **Refinement of Analysis Tools:** We aim to further develop and enhance our analysis tools, as

Orleans, LA, USA, December 10 - 16, 2023。

[51] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, 和 Geoffrey Irving. 2019. 从人类偏好中微调语言模型。CoRR, abs/1909.08593。

indicated in our research timeline. These tools will be crucial for interpreting model outputs, tracking progress, and guiding future research directions.

By pursuing these avenues, we aim to not only advance our understanding and replication of O1’s capabilities but also to push the boundaries of AI research methodologies. Our future plans reflect our commitment to the journey learning paradigm, emphasizing continuous improvement, transparent exploration, and collaborative advancement in the field of artificial intelligence. As we move forward, we remain adaptable to new discoveries and challenges, ready to adjust our plans as our understanding of O1 and advanced AI systems continues to evolve. Through this ongoing journey, we hope to contribute significantly to the development of more capable, interpretable, and ethically aligned AI systems.

Acknowledgment

We would like to express our sincere gratitude to the Shanghai Innovation Institute for providing an exceptional environment for our discussion and debate, which has laid a solid foundation for the advancement of our project. We deeply indebted to all co-authors, with special appreciation extended to the students from GAIR. Your dedication in sacrificing your National Day holiday to collaborate on this groundbreaking endeavor is truly commendable. The progress of this project would not have been possible without your unwavering commitment and hard work.

References

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.
- [2] Murray Campbell, A.Joseph Hoane, and Feng hsiung Hsu. 2002. Deep blue. *Artificial Intelligence*, 134(1):57- 83.
- [3] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: process supervision without process. *ArXiv preprint*, abs/2405.03553.
- [4] Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. <https://github.com/GAIR-NLP/abel>.
- [5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

- [6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. ArXiv, abs/2110.14168.
- [7] Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. arXiv preprint arXiv:2311.01460.
- [8] Elvis Dohmatob, Yunzhen Feng, Pu Yang, Francois Charton, and Julia Kempe. 2024. A tale of tails: Model collapse as a change of scaling laws.
- [9] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. Advances in Neural Information Processing Systems, 36.
- [10] Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data.
- [11] Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyao Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, et al. 2024. Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. ArXiv preprint, abs/2404.05221.
- [12] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. ArXiv, abs/2210.11610.
- [13] Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai.
- [14] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.
- [15] Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2024a. Self-alignment with instruction backtranslation. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.
- [16] Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. 2024b. Chain of thought empowers transformers to solve inherently serial problems. ArXiv preprint, abs/2402.12875.
- [17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.
- [18] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. ArXiv preprint, abs/2406.06592.
- [19] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.
- [20] Ning Miao, Yee Whye Teh, and Tom Rainforth. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. ArXiv, abs/2308.00436.

- [21] Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. CoRR, abs/2404.19733.
- [22] Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. arXiv preprint arXiv:2407.18219.
- [23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36.
- [24] Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, and Tim Genewein. 2024. Grandmaster-level chess without search. arXiv preprint arXiv:2402.04494.
- [25] Nikhil Sardana and Jonathan Frankle. 2023. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. ArXiv, abs/2401.00448.
- [26] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Yu Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.
- [27] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2024. The curse of recursion: Training on generated data makes models forget.
- [28] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. nature, 529(7587):484-489.
- [29] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815.
- [30] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. ArXiv, abs/2408.03314.
- [31] Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. ArXiv preprint, abs/2403.09472.
- [32] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. ArXiv preprint, abs/2211.14275.
- [33] Chaojie Wang, Yanchen Deng, Zhiyi Lv, Shuicheng Yan, and An Bo. 2024a. Q*: Improving multi-step reasoning for llms with deliberative planning. ArXiv preprint, abs/2406.14283.
- [34] Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024b. Math-shepherd: Verify and reinforce llms step-by-step without human annotations.
- [35] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024c. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9426-9439.
- [36] Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024d. Self-taught evaluators.
- [37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824-24837.
- [38] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason

Weston, and Sainbayar Sukhbaatar. 2024a. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. CoRR, abs/2407.19594.

[39] Ting Wu, Xuefeng Li, and Pengfei Liu. 2024b. Progress or regress? self-improvement reversal in post-training. CoRR, abs/2407.05013.

[40] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy. ArXiv preprint, abs/2404.05692.

[41] Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. 2024. Some things are more cringe than others: Iterative preference optimization with the pairwise cringe loss.

[42] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

[43] Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2024. Physics of language models: Part 2.2, how to learn from mistakes on grade-school math problems.

[44] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net.

[45] Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D Goodman. 2024a. Quiet-star: Language models can teach themselves to think before speaking. arXiv preprint arXiv:2403.09629.

[46] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems, 35:15476-15488.

[47] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2024b. Star: self-taught reasoner bootstrapping reasoning with reasoning. In Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’ 22, Red Hook, NY, USA. Curran Associates Inc.

[48] Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. 2023. Mr-gsm8k: A meta-reasoning benchmark for large language model evaluation. CoRR, abs/2312.17080.

[49] Zhihan Zhang, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024. Learn beyond the answer: Training language models with reflection for mathematical reasoning. arXiv preprint arXiv:2406.12050.

[50] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

[51] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. CoRR, abs/1909.08593.