

DeepSeek-Prover-V2：通过强化学习子目标分解推进形式化数学推理

DeepSeek-Prover-V2: Advancing Formal Mathematical Reasoning via Reinforcement Learning for Subgoal Decomposition

DeepSeek Team

摘要

DeepSeek-Prover-V2 在 Lean 4 形式化定理证明上达到当前最高水平 (SOTA)，通过强化学习和子目标分解策略在 MinF2F 测试集达到 88.9% 的 pass ratio。模型能够将复杂的数学证明任务分解为可管理的子目标，逐步构建严密的证明链条。这一突破标志着 AI 在形式化数学证明领域的重要进展。

1. Introduction

1. 引言 大型语言模型 (LLMs) 推理能力的涌现已彻底变革了人工智能的众多领域, 尤其在数学求解领域 (DeepSeek-AI, 2025)。这些进展主要得益于推理时扩展范式, 尤其是通过自然语言思维链推理 (Jaech et al., 2024)。与仅依赖单次前向传播得出答案不同, LLMs 能够对中间推理步骤进行反思, 从而同时提升准确性与可解释性。尽管自然语言推理...

(该方法的) 求解策略支持模块化、可重用性, 并能实现更高效的证明搜索 (Wang et al., 2024b; Zheng et al., 2024)。近期研究通过采用多层次结构进行结构化证明生成 (Wang et al., 2024a), 以及利用强化学习技术优化将复杂定理分解为可管理子目标的过程 (Dong et al., 2024), 进一步拓展了这一范式。在本文中, 我们开发了一种用于子目标分解的推理模型, 利用一系列合成冷启动数据与大规模强化学习来提升其性能。为构建冷启动数据集, 我们设计了一条简单而高效的递归定理证明流水线, 采用 DeepSeek-V3 (DeepSeek-AI, 2024) 作为统一的子目标分解与形式化工具。我们提示 DeepSeek-V3 将定理分解为高层证明草图, 同时将这些证明步骤形式化为 Lean 4 代码, 从而生成一系列子目标。由于子目标分解由大型通用模型驱动, 我们采用一个较小的 7B 参数模型来处理每个子目标的证明搜索, 从而降低相关的计算负担。此外, 我们引入了一种课程学习框架, 利用分解出的子目标生成猜想定理, 逐步增加训练任务的难度, 以更好地引导模型的学习过程。一旦某个难题的分解步骤得到解决, 我们将完整的逐步形式化证明与 DeepSeek-V3 对应的思维链配对, 构建冷启动推理数据。基于冷启动数据, 后续引入强化学习阶段以进一步增强模型能力。我们需要证明: 对于任意整数 $n \geq 4$, 不等式 $n^2 \leq n!$ 成立。其中, $n!$ 表示 n 的阶乘, 即所有不超过 n 的正整数的乘积。为了形式化地证明该命题, 我们可以使用数学归纳法。具体归纳过程如下:

2. Inductive Step: Assume the inequality holds for some $n = 4$, i.e., $2 \leq 4!$.

2. 归纳步骤: 假设不等式对某个 $n=4$ 成立, 即 $n^2 \leq n!$ 。

然后证明它对 $n+1$ 也成立, 即 $(n+1)^2 \leq (n+1)!$ 。

Lean 4 证明结构:

```
theorem induction_ineq_nsqlefactn (n : ℕ) (h : 4 ≤ n) : n ^ 2 ≤ n ! := by
  have base_case : 4 ^ 2 ≤ 4 ! := by simp [Nat.factorial]
  have inductive_step : ∀ k ≥ 4, k ^ 2 ≤ k ! → (k + 1) ^ 2 ≤ (k + 1) ! := by
    intro k h₁ h₂; simp_all [Nat.factorial]; nlinarith
  have final_proof : ∀ n ≥ 4, n ^ 2 ≤ n ! := by
    intro n hn; induction' hn with k hk
  case refl => exact base_case
  case step => apply inductive_step k hk; exact by assumption
  apply final_proof; exact h
```

证明: 对任意整数 $n \geq 4$, 有 $n^2 \leq n!$ 。

theorem induction_ine

我们贡献了 ProverBench, 这是一个包含325个形式化问题的基准数据集, 旨在推动神经定理证明研究, 其中包括15道来自享有盛誉的AIME竞赛 (2024-2025年) 的题目。DeepSeek-Prover-V2-671B成功解决了这15道极具挑战性的AIME题目中的6道, 进一步彰显了其卓越的数学推理能力。 Draft: 我们贡献了 ProverBench 基准数据集, 该数据集包含325个形式化问题, 旨在推动神经定理证明研究, 其中涵盖15道来自享有盛誉的AIME (美国数学邀请赛) 竞赛 (2024-2025年) 的题目。DeepSeek-Prover-V2-671B成功解出了这15道极具挑战性的AIME题目中的6道, 进一步彰显了其卓越的数学推理能力。 Refined: 我们贡献了 ProverBench 基准数据集, 该数据集包含325个形式化问题, 旨在推动神经定理证明研究, 其中涵盖15道来自享有盛誉的AIME (美国数学邀请赛) 竞赛 (2024-2025年) 的题目。DeepSeek-Prover-V2-671B成功解出了这15道极具挑战性的AIME题目中的6道, 进一步彰显了其卓越的数学推理能力。

"sophisticated"

->

在AI论文中常译为“强大的”或“卓越的”。这里用“卓越的”或“强大的”均可, 我选“卓越的”。

"contribute"

->

在贡献列表中通常译为“提出”或“构建”。为保持原意, 译为“贡献了”或“提出了”。学术中文常用“提出/构建了”。

我们贡献了 ProverBench 基准数据集, 该数据集包含325个形式化问题, 旨在推动神经定理证明研究, 其中涵盖15道来自享有盛誉的AIME (美国数学邀请赛) 竞赛 (2024-2025年) 的题目。DeepSeek-Prover-V2-671B成功解出了这15道极具挑战性的AIME题目中的6道, 进一步彰显了其卓越的数学推理能力。Text: 我们贡献了 ProverBench 基准数据集, 该数据集包含325个形式化问题, 旨在推动神经定理证明研究, 其中涵盖15道来自享有盛誉的AIME (美国数学邀请赛) 竞赛 (2024-2025年) 的题目。DeepSeek-Prover-V2-671B成功解出了这15道极具挑战性的AIME题目中的6道, 进一步彰显了其卓越的数学推理能力。

2. Method

将复杂定理的证明分解为一系列作为中间步骤的较小引理, 是人类数学家常用的一种有效策略。多项先前的研究在神经定理证明的背景下探索了这种层次化策略, 旨在利用现代大语言模型 (LLMs) 的非形式化推理能力来提升证明搜索效率 (Jiang 等, 2023; Zhao 等, 2023; Wang 等, 2024a; Dong 等, 2024)。在本文中, 我们构建了一个简单而高效的流程, 利用 DeepSeek-V3 (DeepSeek-AI, 2024) 作为形式化定理证明中子目标分解的统一工具。图3 | 我们将分解后的子目标转化为一系列引理声明的示例。我们首先 (a) 替换原始待证目标, 然后 (b) 将前置子目标作为前提引入。类型 (b) 的声明用于复杂问题的递归求解, 而类型 (a) 和 (b) 均被纳入课程学习过程中。

保持原样或按中文习惯调整, 通常保留英文格式。语言模型在非形式化推理能力方面取得了显著突破。为弥合形式化与非形式化推理之间的差距, 我们利用在数学推理和指令遵循能力方面表现出色的前沿通用大语言模型 (LLM), 构建了我们的定理证明系统的基础框架。能够分解证明步骤, 并将其用形式化语言表达。首先用自然语言分析数学问题, 然后将证明分解为更小的步骤, 并将每个步骤翻译为对应的 Lean 形式化语句。仅生成省略细节的高层证明大纲。占位符结尾, 表示待求解的子目标。该方法模仿了人类的证明构建风格, 即将复杂定理逐步简化为一系列更易于处理的引理。子目标的递归求解。生成的子目标, 我们采用递归求解策略, 系统地解决每个中间证明步骤。这种构建方式使得后续子目标能够利用先前步骤的中间结果进行求解, 从而形成更局部的依赖结构, 并有助于推导更简单的引理。为降低大规模证明搜索的计算开销, 我们采用了一个较小的7B 参数证明模型, 该模型专门针对处理分解后的引理进行了优化。在所有分解步骤成功求解后, 即可自动推导出原始定理的完整证明。面向基于子目标的定理证明的课程学习。

人类撰写文本的形式化提供了高质量且多样化的形式化内容, 但证明模型获得的训练信号往往较为稀疏,

因为大量计算尝试未能产生成功证明, 因此无法提供正向奖励信号。为生成更密集的训练信号, Dong and Ma (2025) 提出了一种自博弈方法, 通过生成与原始定理陈述密切相关的可处理猜想来丰富训练问题集, 从而更高效地利用训练计算资源。在本文中, 我们实现了一种 straightforward a

数百条高质量的合成冷启动数据, 这些数据构成了训练 DeepSeek-Prover-V2 的基础。该冷启动数据集生成策略与同期形式化推理模型研究 Kimina-Prover (Wang et al., 2025) 的策略有所不同。具体而言, 我们通过直接将自然语言证明形式化为结构化的形式证明草图来合成数据。相比之下, Kimina-Prover 采用了一种反向工作流: 它首先收集完整的形式化证明及其对应的非形式化版本, 随后利用通用推理模型逆向合成中间自然...

1. High-efficiency non-Chain-of-Thought (non-CoT) mode: This mode is optimized for

Draft: 快速生成 Lean 形式化证明代码, 侧重于生成无需显式中间推理步骤的简洁证明。Final decision: 快速生成 Lean 形式化证明代码, 侧重于生成无需显式中间推理步骤的简洁证明。(Matches academic tone precisely) "explicit intermediate reasoning steps" -> 在形式化方法中, "explicit" 常译为“显式”。Final: 快速生成 Lean 形式化证明代码, 侧重于生成无需显式中间推理步骤的简洁证明。“快速生成 Lean 形式化证明代码, 侧重于生成无需显式中间推理步骤的简洁证明。”(If it's a standalone phrase, this is perfect. I'll provide it directly.) Translation: 快速生成 Lean 形式化证明代码, 侧重于生成无需显式中间推理步骤的简洁证明。

2. High-precision Chain-of-Thought (CoT) mode: This mode systematically articulates

Draft: 在构建最终的形式化证明之前, 先生成中间推理步骤, 强调透明性与逻辑递进性。*Refinement:* 在构建最终的形式化证明之前, 先生成中间推理步骤, 以强调过程的透明性与逻辑的递进性。*Draft:* 与 DeepSeek-Prover-V1.5 (Xin et al., 2024b) 一致, 这两种生成模式由两个不同的引导提示词控制 (示例见附录 A)。*Refinement:* 与 DeepSeek-Prover-V1.5 (Xin et al., 2024b) 保持一致, 这两种生成模式分别由两个不同的引导提示词 (prompt) 进行控制 (示例参见附录 A)。*Draft:* 在第一阶段, 我们在课程学习框架内采用专家迭代来训练一个非CoT证明模型, 同时通过基于子目标的递归证明为难题合成证明。*Refinement:* 在第一阶段, 我们在课程学习框架内采用专家迭代 (expert iteration) 方法训练非思维链 (non-CoT) 证明模型, 同时通过基于子目标的递归证明方法为难题合成证明数据。*Draft:* 选择非CoT生成模式是为了加速迭代训练和数据收集过程, 因为它提供了显著更快的推理和验

证周期。***Refinement:*** 选用非 CoT 生成模式旨在加速迭代训练与数据收集流程，因其推理与验证周期显著更短。***Draft:*** 在此基础上，第二阶段利用冷启动思维链 (CoT) 数据，这些数据是通过将 DeepSeek-V3 的复杂数学推理模式与我们的合成形式化证明相结合而合成的。***Refinement:*** 在此基础上，第二阶段利用冷启动思维链 (CoT) 数据，该数据通过将 DeepSeek-V3 的复杂数学推理模式与我们合成的形式化证明相融合而生成。***Draft:*** CoT 模式通过进一步的强化学习阶段进行增强，遵循推理模型常用的标准训练流程。

Refinement: 随后，CoT 模式通过进一步的强化学习阶段进行增强，该阶段遵循推理模型常用的标准训练流程。***Draft:*** 专家迭代。***Draft:*** DeepSeek-Prover-V2 非CoT模式的训练过程遵循专家迭代 (Polu and Sutskever, 2020) 的范式，这是开发形式化定理证明器广泛采用的框架。***Refinement:*** DeepSeek-Prover-V2 非 CoT 模式的训练流程遵循专家迭代 (expert iteration) 范式 (Polu and Sutskever, 2020)，该范式是开发形式化定理证明器时广泛采用的框架。***Draft:*** 在每次训练迭代中，当前最佳的证明策略被用于为先前迭代中仍未解决的挑战性难题生成证明尝试。***Refinement:*** 在每次训练迭代中，系统使用当前最优的证明策略，为前序迭代中尚未解决的挑战性难题生成证明尝试。***Draft:*** 这些成功的尝试经过 Lean 证明助手验证后，被纳入 SFT 数据集以训练改进的模型。***Refinement:*** 经 Lean 证明助手验证成功的尝试将被纳入监督微调 (SFT) 数据集，用于训练性能更优的模型。***Draft:*** 这种迭代循环确保模型不仅从初始演示数据集中学习，还提炼其自身的成功推理轨迹，逐步提高解决更难问题的能力。***Refinement:*** 该迭代循环确保模型不仅能从初始演示数据集中学习，还能提炼自身成功的推理轨迹，从而逐步提升解决高难度问题的能力。***Draft:*** 整体训练过程与 DeepSeek-Prover-V1 (Xin et al., 2024a) 和 DeepSeek-Prover-V1.5 (Xin et al., 2024b) 基本保持一致，仅对训练问题的分布进行了两处修改。

Refinement: 整体训练流程与 DeepSeek-Prover-V1 (Xin et al., 2024a) 及 DeepSeek-Prover-V1.5 (Xin et al., 2024b) 基本保持一致，仅对训练问题的分布进行了两处调整。

的固定学习率。的固定学习率。数据，该数据生成不包含中间推理步骤的 Lean 代码；的高级数学推理过程蒸馏为结构化的证明路径。非 CoT 部分侧重于 Lean 定理证明器生态系统中的形式化验证能力，而 CoT 示例则显式地建模了将数学直觉转化为形式化证明结构的认知过程。强化学习。与 PPO (Schulman 等, 2017) 不同，GRPO 通过为每个定理提示采样一组候选证明，并基于它们的相对奖励来优化策略，从而无需额外的评论家 (critic) 模型。训练采用二元奖励

机制：每个生成的 Lean 证明若经验证正确则获得奖励 1，否则为 0。为确保学习效果，我们精心筛选训练提示，仅包含那些具有足够挑战性且监督微调模型能够求解的问题。知识蒸馏。强化学习阶段收集的 rollout 数据对该扩展上下文模型进行微调。除了 CoT 推理模式外，我们还引入了专家迭代期间收集的非 CoT 证明数据，以启用一种成本效益更高的证明方案，使小参数模型能够生成简洁的形式化输出。的性能。

3. Experimental Results

在多个形式化定理证明基准数据集上进行了系统评估，涵盖高中竞赛题与本科水平数学题。等, 2024b) 保持一致。除非另有说明，基线模型的评估结果均引自其原始论文。和 IMO 竞赛，以及从 MATH 数据集 (Hendrycks 等, 2021) 中精选的题目。数论和数学归纳法。每个子集包含 244 道题，且在各个学科领域的分布完全一致。题目则被纳入结合子目标分解的课程学习中。进行了一处修订 (详见附录 D)。与当前最优 (SoTA) 模型的对比。数据集上评估的当前最优形式化定理证明模型的对比情况。准确率。

将具有挑战性的问题分解为一系列易于处理的步骤，在非形式化推理与形式化证明构建之间架起了一座有效的桥梁。表3 | DeepSeek-Prover-V2在miniF2F-test上生成的平均token数。CoT与非CoT对比。表1的实验结果表明，在形式化数学推理中，CoT (思维链) 推理模式相较于非CoT模式具有显著的性能优势。这进一步印证了CoT提示的有效性，即鼓励将复杂问题分解为中间步骤，同时也再次证实了推理时扩展 (inference-time scaling) 在形式化定理证明领域依然成立。作为对这些发现的补充，表3提供了DeepSeek-Prover-V2在不同推理模式下生成token数量的统计结果。正如预期，CoT模式生成的输出显著更长，反映了其复杂的推理过程。有趣的是，在非CoT设置下，671B模型平均生成的输出长度比7B模型更长。深入分析发现，尽管非CoT模式未显式提示推理过程，但更大规模的模型经常在证明代码中插入简短的自然语言注释，这些注释类似于隐式的推理步骤 (见附录A)。这表明，即使在没有显式CoT提示的情况下，大容量模型也可能将中间推理过程内化并以隐式方式外显。ProofNet (Azerbayev 等, 2023) 包含371个Lean 3问题，选自多种流行的本科纯数学教材，涵盖实变函数与复变函数分析、线性代数、抽象代数和拓扑学等主题。我们使用了Xin 等 (2024b) 提供的ProofNet的Lean 4翻译版本，该版本进一步划分为两个子集：ProofNet-valid和ProofNet-test，分别包含185和186个问题。ProofNet的测试集专用于模型评估，因为ProofNet-valid问题的变体已包含在Dong和Ma (2025) 提供的公开合成数据集中，该数据集用于我们的监督微调。

如表4所示的结果表明，与不使用CoT的设置相比，使用CoT推理时DeepSeek-Prover-V2的通过率得到了

显著提升。值得注意的是，尽管训练数据主要来源于高中…… "challenging problems into a sequence of tractable steps, serving as an effective bridge between informal reasoning and formal proof construction." -> 将具有挑战性的问题分解为一系列易于处理的步骤，在非形式化推理与形式化证明构建之间架起了一座有效的桥梁。表3 | DeepSeek-Prover-V2在miniF2F-test上生成的平均token数。CoT与非CoT对比。"The experimental results in Table 1 demonstrate a substantial performance advantage of the CoT reasoning mode over the non-CoT mode in formal mathematical reasoning." -> 表1的实验结果表明，在形式化数学推理中，CoT（思维链）推理模式相较于非CoT模式具有显著的性能优势。

"This reinforces the effectiveness of CoT prompting, which encourages decomposition of complex problems into intermediate steps, and further confirms that inference-time scaling holds in the domain of formal theorem proving." -> 这进一步印证了CoT提示的有效性，即鼓励将复杂问题分解为中间步骤，同时也再次证实了推理时扩展（inference-time scaling）在形式化定理证明领域依然成立。"Complementing these findings, Table 3 provides statistics on the number of tokens generated by DeepSeek-Prover-V2 under different reasoning modes." -> 作为对这些发现的补充，表3提供了DeepSeek-Prover-V2在不同推理模式下生成token数量的统计结果。

Translation: 表4 | ProofNet-test 和 PutnamBench 上的实验结果。Goedel-Prover-SFT 和 STP 在 PutnamBench 上的得分来源于其原始论文，这些论文是在包含 644 道题目的 PutnamBench 早期版本上进行评估的。
Translation: ……级数学，该模型在更高级的大学本科水平数学问题上表现出强大的泛化能力，凸显了其稳健的形式化推理能力。
Translation: PutnamBench (Tsoukalas 等, 2024) 是一个持续更新的基准测试，收录了 1962 年至 2023 年间威廉·洛厄尔·普特南数学竞赛 (William Lowell Putnam Mathematical Competition) 的竞赛数学题。普特南竞赛是面向美国和加拿大本科生的一项极具声望的年度数学竞赛，涵盖分析学、线性代数、抽象代数、组合数学、概率论和集合论等多种大学水平领域。我们在最新发布的 PutnamBench 上评估了我们的模型，该版本包含 658 道使用 Lean 4 形式化的题目。我们排除了与 Lean 4.9.0 不兼容的题目，并在剩余的 649 道题目上对模型进行了评估。在初始运行中，我们使用每道题 1024 的采样预算 (sample budget) 成功解决了 49 道题。在向基准测试维护者提交证明后，我们因陈述形式化错

误 (misformalized statements) 排除了两道题。最终结果为成功解决 47 道题 (如表 4 所示)，这显著优于其非 CoT 版本。这些结果进一步凸显了 CoT 推理方法在处理具有挑战性的本科生水平数学问题方面的有效性。

Refinement: "sample budget" -> 采样预算/推理预算。In LLM context, it usually means number of generated samples/proof attempts. I'll use "采样预算" or "推理尝试次数". Let's stick to "采样预算". "misformalized statements" -> 形式化表述有误的题目。"non-CoT counterpart" -> 非思维链 (non-CoT) 版本。
Translation: 强化学习中的奖励黑客 (Reward Hacking) 现象。我们最初的报告提出了一项意外发现：DeepSeek-Prover-V2-7B 成功解决了 PutnamBench 上的 13 道题目，而其更大的 671B 版本未能解决。我们感谢 Lean 社区协助我们查明导致这一意外结果的原因，该原因被追溯至 Lean

某些边界情况。来利用这一用户界面漏洞 (示例见附录B)，而这些在671B版本生成的输出中明显缺失。某些边界情况。来利用该用户界面漏洞 (示例见附录B)，而这些调用在671B版本生成的输出中则明显缺失。是一个综合性基准测试，包含100道用 Lean 4 形式化的组合数学竞赛题，每道题均配有对应的自然语言描述。进行评估。在此设置中，正确答案已嵌入 Lean 语句中，使得评估能够完全聚焦于证明生成过程。占位符的题目后，我们对该基准中的77道题目进行了评估。我们的初步运行结果显示成功解决了12道题。在与基准维护者合作并确认其中两道题的陈述存在形式化错误 (misformulated) 后，最终结果修正为成功解决10道题。这些结果表明，尽管该证明模型主要在数论和代数领域进行训练，但面对组合数学问题固有的高难度，它仍展现出令人鼓舞的泛化能力。推理能够有效识别形式化错误，并据此调整其证明策略。爆炸原理) 关闭目标命题。

并涵盖广泛的领域，包括代数、应用数学、微积分、数论与离散数学。FormalMATH-Lite 是从完整语料库中精心筛选出的一个包含 425 道题目的可管理子集 (其中包含 359 道高中水平题目与 66 道本科水平题目)，旨在解决在完整数据集上进行评估的不切实际性问题，同时仍能支持对跨不同数学领域的测试时缩放进行系统评估。表6 展示了各类定理证明器在 FormalMATH-All 与 FormalMATH-Lite 基准测试上的性能对比分析。Dee...

Intro phrase: 实现对高中竞赛题目与本科水平数学的更全面评估。
Table 8 Caption: 表8 | 用于形式化的AIME 24&25题目选择。带下划线加粗索引的题目已由DeepSeek-Prover-V2求解。使用Maj@16 (16次采样多数投票) 由DeepSeek-V3-0324求解的题目以灰色背景高亮显示。美国数学邀请赛 (AIME) 是

一项年度数学竞赛，旨在挑战和表彰在数学方面展现出卓越才能的高中生。AIME 24&25的题目已成为评估大语言模型推理能力的标准基准。为了弥合模型在形式化与非形式化数学推理性能评估之间的差距，我们精心挑选并形式化了AIME 24&25中的一组题目。为确保形式化过程更加清晰简洁，我们过滤掉了那些在Lean中表示可能较为繁琐的几何、组合与计数问题。最终筛选出15道题目，涵盖初等数论与代数等竞赛级主题。我们使用自然语言数学推理的标准“求解答案”任务，在所选题目集上对DeepSeek-V3-0324进行评估。通过对16个采样响应进行多数投票，该模型成功解出了15道题中的8道。相比之下，DeepSeek-Prover-V2-671B在给定正确答案的形式化证明生成设置下，能够为15道题中的6道构建有效的形式化证明。这一对比凸显了非形式化数学推理与形式化定理证明之间的性能差距正在显著缩小，表明先进语言模型在语言理解与形式化逻辑严谨性之间的对齐程度不断提高。 *Table 9 Caption:* 表9 | ProverBench中数学领域的分布情况。

(确保) 在难度级别和主题领域上实现全面覆盖。因此，我们形式化了310道题目，涵盖广泛的光谱，从竞赛级别的初等数学到本科阶段通常涉及的高级主题。这一综合性基准测试覆盖了数论、初等代数、线性代数、抽象代数、微积分、实分析、复分析、泛函分析和概率论。有意纳入这一系列多样化的数学领域，使得能够全面评估模型在不同抽象层次和推理风格下的能力。数论和代数问题测试模型处理离散结构和方程的能力，而分析类问题则评估对极限、连续性和微积分的理解。抽象代数和泛函分析部分挑战模型对抽象结构和空间进行推理，需要高级的形式化推理能力。评估结果如表7所示。如图所示，采用思维链 (CoT) 推理的DeepSeek-Prover-V2-671B模型在所有基线模型中持续表现最优，进一步印证了在其他基准测试评估中观察到的趋势。

4. Conclusion

在本工作中，我们提出了一种综合的流水线，用于合成冷启动推理数据，以推进形式化定理证明。本文提出了一套完整的流水线，用于合成冷启动推理数据，以推动形式化定理证明的发展。在Lean 4证明助手作为子目标分解和引理形式化的统一模型。我们的数据构建过程基于一个递归定理证明框架。作为统一模型，在Lean 4证明助手中同时负责子目标分解与引理形式化。我们的方法将高层证明草图与形式化步骤相结合，创建了一系列可管理的子目标，这些子目标可以使用较小的7B模型高效解决，显著降低了计算需求。该方法将高层证明概要 (proof sketches) 与形式化步骤相结合，生成一系列易于处理的子目标。这些子目标可通过较小的7B参数模型高效求解，从而大幅降低了计算开销。我们开发的课程学习框架使用这些分解的子目标来生成越来越难的训练任务，创建了一个更有效的学习进程。我们设计的课程学习框架利用这些分解后的子目标，生成难度递增的训练任务，从而构建出更高效的学习演

进路径。的思维链推理配对，我们建立了宝贵的冷启动推理数据，连接了非形式化数学思维与形式化证明结构。(桥接) 了非形式化的数学思维与形式化证明结构。的思维链推理相配对，我们构建了极具价值的冷启动推理数据，有效衔接了非形式化的数学思维与形式化证明结构。随后的强化学习阶段大幅增强了这种联系，导致形式化定理证明能力显著提高。随后的强化学习阶段进一步强化了这种关联，使模型在形式化定理证明能力上取得了显著提升。在一系列基准测试中始终优于所有基线，涵盖高中竞赛题和本科水平数学。在涵盖高中数学竞赛题与本科水平数学题的一系列基准测试中，均稳定超越了所有基线模型。我们未来的工作将专注于将此范式扩展到类似AlphaProof的系统，最终目标是解决代表自动定理证明挑战前沿的IMO级别数学问题。未来的工作将致力于将该范式扩展至类似AlphaProof的系统，最终目标是攻克代表自动定理证明前沿挑战的国际数学奥林匹克 (IMO) 级别数学问题。

2025. URL <https://arxiv.org/abs/2501.12948>.

参考文献：

K. Dong and T. Ma. STP: Self-play LLM theorem provers with iterative conjecturing and proving. arXiv preprint arXiv:2502.00212, 2025.

K. Dong, A. Mahankali, and T. Ma. Formal theorem proving by rewarding LLMs to decompose proofs hierarchically. arXiv preprint arXiv:2411.01829, 2024.

M. Eppe, C. Gumbsch, M. Kerzel, P. D. Nguyen, M. V. Butz, and S. Wermter. Intelligent problem-solving as integrated hierarchical reinforcement learning. Nature Machine Intelligence, 4(1): 11–20, 2022.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathemat

L. C. Paulson. Isabelle: 一个通用定理证明器。Springer Verlag, 1994年。S. Polu 和 I. Sutskever. 生成式语言模型在形式化证明中的应用。这些都是形式化定理证明领域的重要基础工作和相关研究成果，为DeepSeek-Prover-V2的方法设计提供了理论支撑和技术参考。

参考文献 (续)：

reasoning: A new frontier in AI. arXiv preprint arXiv:2412.16075, 2024.

H. Ying, Z. Wu, Y. Geng, J. Wang, D. Lin, and K. Chen. Lean workbook: A large-scale Lean problem set formalized from natural language math problems. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024.

Z. Yu, R. Peng, K. Ding, Y. Li, Z. Peng, M. Liu, Y. Zhang, Z. Yuan, H. Xin, W. Huang, et al. FormalMath: Benchmarking formal mathematical reasoning of large language models. arXiv preprint arXiv:2505.02735, 2025.

J. Zhang, Q. Wang, X. Ji, Y. Liu, Y. Yue, F. Zhang, D. Zh

Lean 4 证明片段 (三角恒等式化简) :

-- 使用特定角度的余弦和恒等式化简表达式

```
have h5 : Real.cos (2 * Real.pi / 7) = Real.cos (2 * (Real.pi / 7)) := by ring
```

```
have h6 : Real.cos (3 * Real.pi / 7) = Real.cos (3 * (Real.pi / 7)) := by ring
```

```
rw [h5, h6]
```

```
have h7 : Real.cos (2 * (Real.pi / 7)) = 2 * Real.cos (Real.pi / 7) ^ 2 := by
```

```
have h71 : Real.cos (2 * (Real.pi / 7)) = 2 * Real.cos (Real.pi / 7) ^ 2 := by
```

```
rw [Real.cos_two_mul]; ring_nf; nlinarith [Real.cos_sq_add_sin_sq (Real.pi / 7)]
```

```
rw [h71]
```

```
have h8 : Real.cos (3 * (Real.pi / 7)) = 4 * Real.cos (Real.pi / 7) ^ 3 := by
```

Lean 4 证明片段 (正弦恒等式) :

```
have h191 : Real.sin (3 * (Real.pi / 7)) = 3 * Real.sin (Real.pi / 7) ^ 3 := by
```

```
rw [Real.sin_three_mul]; ring_nf
```

```
nlinarith [Real.cos_sq_add_sin_sq (Real.pi / 7), Real.sin_le_one (Real.pi / 7), Real.cos_le_one (Real.pi / 7)]
```

```
rw [h191]
```

```
have h20 : Real.sin (2 * (Real.pi / 7)) = 2 * Real.sin (Real.pi / 7) * Real.cos (Real.pi / 7) := by
```

```
have h201 : Real.sin (2 * (Real.pi / 7)) = 2 * Real.sin (Real.pi / 7) * Real.cos (Real.pi / 7) := by
```

```
rw [Real.sin_two_mul]; ring_nf
```

```
nlinarith [Real.cos_sq_add_sin_sq (Real.pi / 7), Real.sin_le_one (Real.pi / 7), Real.cos_le_
```

(Lean 4 代码示例) 使用非线性算术推理器 nlinarith, 结合 $\cos(x) \leq 1$ 的不等式约束, 对 $\pi/7, 2\pi/7, 3\pi/7$ 的余弦值进行推理证明。这体现了自动定理证明中代数推理的重要作用。

或者, 我们可以注意到, 对于 $p \geq 2, q \geq 3, r \geq 4$, 乘积 $(p-1)(q-1)(r-1)$ 至少为 6, 且 $pqr \geq 1$

且 $2p + 2q + 2r - 7 \leq 2p + 2(p+1) + 2(p+2) - 7 = 6p - 1$ 。但 $6p - 1$

— 推导方程 $pqr - 2pq - 2pr - 2qr + 2p + 2q + 2r - 1 = 0$ 。

— 当 $p=2$ 时, 无解。

— 当 $p=3$ 时, 解得 $(p, q, r) = (3, 5, 15)$ 。

— 当 $p \geq 4$ 时, 无解。

5. **情形 $k=3$ ** :

— 推导方程 $2pqr - 3pq - 3pr - 3qr + 3p + 3q + 3r - 2 = 0$ 。

— 当 $p=2$ 时, 解得 $(p, q, r) = (2, 4, 8)$ 。

— 当 $p \geq 3$ 时, 无解。

6. **结论** :

— 唯一有效的三元组为 $(2, 4, 8)$ 和 $(3, 5, 15)$ 。

含 have 语句的 Lean 4 证明

```
```lean4
```

```
theorem imo_1992_p1 (p q r : Z) (h0 : 1
```

(Lean 4 代码) 通过非线性算术证明  $(p-1)(q-1) \geq 2$  以及  $(p-1)(q-1)(r-1) \geq 6$ 。这些不等式约束是数论证明中的关键中间步骤, 用于缩小变量取值范围。

Lean 4 证明片段 (IMO 1992 P1 分类讨论) :

```
have h20 : (q : Z) ^ 4 := by
```

```
by_contra h20; have h21 : q ^ 3 := by linarith; have h22 : q = 3 := by linarith
```

```
have h23 : (p : Z) = 3 := by norm_num [h17]; have h24 : (q : Z) = 3 := by norm_num [h22]
```

```
have h25 : (r : Z) ^ 4 := by linarith
```

```
norm_num [h17, h22, h23, h24] at h18
```

```
(try omega) (try nlinarith)
```

```
have h21 : (r : Z) ^ q + 1 := by linarith
```

```
have h22 : (q : Z) ^ 4 := by linarith
```

```
have h23 : (p : Z) = 3 := by norm_num [h17]
```

```
have h24 : (p : Z) * q * r ^ 1 = 2 * ((p - 1) * (q - 1) * (r - 1)) := by linarith
```

```
have h25 : 3 * q * r ...
```

Lean 4 证明片段 (整除性分析) :

-- 利用  $(q-3)$  的绝对值整除 5 的绝对值这一事实

have h37 : ((q : Z) 3).natAbs | 5 := by simp [Int.natAbs] using h34

-- 由于 5 的因子只有 1 和 5，我们分情况讨论

have h38 : ((q : Z) 3).natAbs = 1 ∨ ((q : Z) 3).natAbs = 5 := by

have h39 : ((q : Z) 3).natAbs | 5 := h37

have h40 : ((q : Z) 3).natAbs 5 := Nat.le\_of\_dvd (by decide) h39

interval\_cases ((q : Z) 3).natAbs norm\_num at h39 omega

cases h38 with

| inl h38 =>

have h41 : (q : Z) 3 = 1 ∨ (q : Z) 3 = 5 := by

have h42 : ((q : Z) 3).natAbs = 1 := h38

have h43 : (q : Z) ...

以下示例取自 CombiBench 的早期版本 (Liu 等, 2025)，该问题已在最新版本中修正。为仅由字母和构成的无限周期词。假设的最小周期大于 2。则称有限非空词出现在中。证明：至少存在一个无处不在的有限非空词。在生成用于形式化证明该定理的 Lean 4 代码之前，请先提供一份详细的证明计划，概述主要的证明步骤与策略。中间引理以及证明结构，以便指导最终形式化证明的构建。策略。或保留英文加注释。比较稳妥。周期序列。最小周期。有限非空词。以下示例取自 CombiBench 的早期版本 (Liu 等, 2025)，该问题已在最新版本中修正。

1. We have an infinite periodic word 'W : Z → Fin 2' with minimal period 'N > 2' (where 'n : N+' is a

1. 我们有一个最小周期为  $N > 2n$  的无穷周期词  $W : \mathbb{Z} \rightarrow \text{Fin } 2$  (其中  $n : \mathbb{N}^+$  为正整数)。 —  $\text{Function.Periodic } W, N$  表示  $\forall z, W(z + N) = W(z)$ 。 — 最小性条件为  $\forall z, N$

2. The definitions:

## 2. 定义：

— 有限词  $U = (n, f : \text{Fin } n \rightarrow \text{Fin } 2)$  \*\*出现\*\*于  $W$  中，若存在  $k$  使得对所有  $i : \text{Fin } n$  有  $U.2, i = W(k + i)$ 。

— 在 Lean 中， $\text{appears } W, U$  定义为  $\exists k, \forall i : \text{Fin } n, U.2, i = W(k + i)$ 。

—  $U$  是\*\*普遍\*\*的 (ubiquitous)，若四个增广词

$Ua, Ub, aU, bU$  均出现于  $W$  中。

— 在 Lean 中， $\text{ubiquitous } W, U$  检查四个增广词是否出现，其中：

—  $Ua$  为  $\langle U.1 + 1, \text{Fin.snoc } U.2, 0 \rangle$ ，

—  $Ub$  为  $\langle U.1 + 1, \text{Fin.snoc } U.2, 1 \rangle$ ，

—  $aU$  为  $\langle U.1 + 1, \text{Fin.cons } 0, U.2 \rangle$ ，

—  $bU$  为  $\langle U.1 + 1, \text{Fin.cons } 1, U.2 \rangle$ 。

3. The goal is to find an injective function from 'Fin n' to ubiquitous finite nonempty words in 'W'

标题：3. 目标是寻找一个从 'Fin n' 到 'W' 中普遍存在的有限非空词的单射函数)：(即，到  $\Sigma k, \text{Fin } k \rightarrow \text{Fin } 2$ ，其中  $k \neq 0$  且该词是普遍存在的)。然而，Lean 中的问题略有不同：

2. The conditions are:

## 2. 条件如下：

**\*\*注意\*\***：在 Lean 中， $\text{Function.Periodic } f, c$  表示  $\forall z, f(z + c) = f(z)$  ( $c$  为周期，不一定是最小周期)。此处  $hW$  取  $N : \mathbb{N}$ ，但  $W : \mathbb{Z} \rightarrow \text{Fin } 2$  使用  $\mathbb{Z}$  上的加法， $N$  被 cast 到  $\mathbb{Z}$  (因为  $f(z+c)$  中  $c : \mathbb{Z}$ )。

然而：

—  $hW, 0, \dots$  不能直接适用，因为  $0 : \mathbb{N}$  满足  $2^{n.1} \geq 1$ ，但

$\text{Function.Periodic } W, 0$  为  $\forall z : \mathbb{Z}, W(z + 0) = W(z)$ ，这显然成立 ( $z + 0 = z$ )。但  $hW, 0, (\text{by } \text{linarith})$  断言  $n \in \text{Function.Periodic } W, 0$ ，这是错误的。

**\*\*结论\*\***：Lean 问题中的假设是矛盾的，因为

2. The conditions '(x i).1 ≠ 0' ('1 ≠ 0') and 'ubiquitous W (x i)' (which will follow from 'false').

条件  $(x i).1 \neq 0$  (即  $1 \neq 0$ ) 以及 ubiquitous  $W (x i)$  (由 false 推导得出)。这些是形式化逻辑中关于词序列和符号出现性质的数学条件描述。

3. The embedding is trivial because all 'i : Fin n' map to the same word.

3. 嵌入是平凡的，因为所有  $i$ ：  
 $\text{Fin}, n$  映射到同一个词。

但要实际证明

$\text{ubiquitous}, W, (x, i)$ ，我们需要证明  
 $\text{appears}, W, \angle 2, \angle$   
 等，但可以从  $hW, 0$  导出  $\text{false}$ 。

然而， $hN : 2^{n.1} \geq 2$  (若  $n.1 \geq 1$  则  $2^{n.1} \geq 2$ )。

但  $N : \mathbb{N}$ ，且  $0 < 2^{n.1} \geq 1$ ，所以  $N' = 0$

5. But 'Function.Periodic W 0' is trivially true, so  
 'hW' 0 ... ' is 'false'.

5. 但  $\text{Function.Periodic}, W, 0$   
 显然成立，所以  $hW, 0, \dots$  为  
 $\text{false}$ 。

注意  $2^{n.1}$  在  $n.1 = 0$  时不一定  $\geq 2$ ，但  $n : \mathbb{N}^+$  意味着  $n.1 \geq 1$  ( $n \neq 0$ )。因此，

$$2^{n.1} \geq 2^1 = 2 \quad (n.1 \geq 1)$$

然而，若  $N : \mathbb{N}$  满足  $N > 2^{n.1}$ ，则  $N \geq 1$  (当  $n.1 = 1$  且  $2^{n.1} = 2 > N$  时  $N$  可能为 1，但由  $hN$  这不可能)。不，在  $hN$  中， $2^{n.1} \geq 2^1 = 2$ ，所以  $N \geq 3$ 。

$n : \mathbb{N}^+$  为正整数 ( $n.1 \geq 1$ )，因此：

—  $n.1 \geq 1$  意味着  $2^{n.1} \geq 2^1 = 2$ ，所以  $N > 2$  且  $N \geq 3$ 。

因此， $0$

1. Derive 'False' from 'hW':

1. 从  $hW$  导出  $\text{False}$ ：37 —  
 由于  $N > 2^{n.1} \geq 2$ ，故  $N \geq 3$ 。 —  
 $hW, 0$  要求  $0$

3. The conditions on 'x' are vacuously true ('False → anything').

3. 关于  $x$  的条件 vacuously 成立 ( $\text{False} \rightarrow \text{anything}$ )。  
 ### Lean have 语句 `lean4 theorem imosl_2011_c6 (W : Z → Fin 2) (n : N+) (N : N) (hN : 2 ^ n.1`