

# 设计合成：通过结构引导实现受控数据生成

Lei Xu<sup>1</sup>, Sirui Chen<sup>1,2</sup>, Yuxuan Huang<sup>1,3</sup>, Chaochao Lu<sup>1\*</sup>

<sup>1</sup>Shanghai Artificial Intelligence Laboratory, <sup>2</sup>Tongji University, <sup>3</sup>Shanghai Jiao Tong University  
{ xulei, chensirui, huangyuxuan, luchaochao } @pjlab.org.cn

## Abstract

由于复杂的逻辑和精确计算的需求，数学推理对大型语言模型（LLMs）来说仍然具有挑战性。现有的方法通过重新措辞问题来合成数据集以增强 LLM 的推理能力，但在生成质量和问题复杂性方面面临问题。为了解决这个问题，我们提出通过生成的问题解决代码从数学推理中提取结构信息，并通过结构化的解决方案指导数据生成。应用于 MATH 和 GSM8K，我们的方法生成了 39K 个问题及其标记的中间步骤，并提供了一个由 6.1K 个更高难度问题组成的基准。我们的基准测试结果显示，模型性能随着推理长度的增加而下降。此外，我们使用所提出的训练数据对一系列 LLMs 进行了微调实验，结果验证了我们数据集的有效性。我们希望所提出的方法和数据集能为增强 LLM 推理能力的未来研究提供贡献。我们的代码和数据可在 <https://github.com/solerxl/StructuralGeneration> 获得。

## 1 引言

近年来，大型语言模型（LLM）以快速的速度被开发出来，并在自然语言理解、生成和解题能力上表现出巨大的改进。它们的成功使其成为解决各种应用的热门选择，包括数学问题求解 (Zhou et al., 2023; Wang et al., 2024a; Gao et al., 2023)、代码生成 (Nijkamp et al., 2023; Chen et al., 2021) 和开放域对话 (OpenAI, 2024; Ouyang et al., 2022)。尽管它们的表现令人印象深刻，LLM 仍然在推理任务上表现欠佳 (Valmeekam et al., 2023; Zhou et al., 2023)。最具代表性的任务是解决数学问题，这需要复杂的多步骤推理能力，而 LLM 缺乏这些能力，常常会产生无意义或不准确的内容 (Lai et al., 2024)。为了应对这一挑战，后期训练已被证明在增强 LLM 的推理能力方面很有效。之前的研究 (Yue et al., 2024; Zeng et al., 2023) 表明，在高质量的链式推理 (CoT) 数据上微调预训练的 LLM，可以通过使模型更好地利

用预训练的知识来提高性能。一个显著的例子是 GPT-o1 (OpenAI, 2024b)，它通过延长 CoT 推理过程的长度引入了推理时间的扩展。尽管后期训练提供了有价值的密集监督，高质量的 CoT 推理数据稀缺仍然是一个主要的挑战。为此，诸如 PRM800K (Lightman et al., 2023) 之类的人类标注的数据集被引入。然而，这样的数据成本高昂且难以扩展，使得自动数据生成成为更实用的替代方案。提出了各种数据增强技术 (Yue et al., 2024; Tang et al., 2024a; Yu et al., 2024; Luo et al., 2025; Shao et al., 2024) 来提高 LLM 的推理能力。例如，MetaMath (Yu et al., 2024) 使用引导法从多个角度重新措辞数学问题。(Wang et al., 2024a) 将代码集成到推理过程中，使模型可以通过执行结果自我验证和纠正其推理。

如 (Saxton et al., 2019) 所指出的，解决数学问题依赖于两个核心能力：语言理解和数学推理。现有的数据生成方法主要关注前者，而对推理的结构性方面关注有限。结果，它们通常会面临以下问题：

1. 标签缺失：在无法访问中间计算过程的情况下，生成新问题的准确答案对于现有方法而言是具有挑战性的 (Wang et al., 2024a)，更不用说提供步骤级别的监督了。
2. 依赖于代理的质量：推理路径和答案的质量在很大程度上取决于代理，这使生成过程容易受到诸如幻觉 (Lai et al., 2024) 和奖励攻击 (Skalse et al., 2022; Luong et al., 2024) 等问题的影响。
3. 有限数据扩展：由于受制于代理的推理能力，现有方法难以生成超出代理能解决的问题。

在这项工作中，我们强调结构信息在数学推理中的作用，并利用它来生成结构化推理数据。如图 ?? 所示，从现有问题出发，我们将其推理过程转化为可执行的 Python 代码，这简化了语义并将推理限制在一个计算结构内。程

\*Corresponding author.

序由代码-推理对组成，其中每一行计算一个中间结果，并用相应的推理步骤进行注释。这既提供了真实数据又提供了步骤级监督（解决问题 1），同时也捕捉到步骤之间的依赖关系。利用这种结构，我们通过对代码进行最小改动来生成新问题。

所提出的方法将数学问题生成的任务分解为两个任务：首先，对代码进行局部干预，以确保中间步骤的正确性（解决问题 2）；其次，由代理进行翻译，将修改后的代码转换为连贯的问题和推理过程。通过获取上下文和答案，问题生成变为翻译任务而非推理任务，从而减少对模型推理能力的依赖，并允许问题难度超出代理的能力范围（解决问题 3）。

此外，通过利用结构信息，我们将解决方案分解为若干步骤，并通过步骤数量来量化问题的复杂性。尽管之前的研究 (Qi et al., 2025) 表明复杂性会影响模型性能，但很少有研究提供量化的方法来测量复杂性或提供带有此类注释的数据集。我们的方法通过分析和注释带有步骤信息的真实世界数据集来弥补这一空缺。

这项工作的贡献总结如下：

- 从方法论的角度来看，我们提出了一种新方法，利用结构信息来监控、指导和干预数学推理。该结构能够生成具有步骤级和结果级监督的干预数据。重要的是，它使我们能够通过步骤计数来量化问题的复杂性。
- 从数据的角度来看，我们基于 MATH 和 GSM8K 创建了更具挑战性的数据集。与现有方法不同的是，我们的数据集源自可执行代码，确保了更可靠的计算过程。
- 从模型的角度来看，我们在自己的数据集上微调了 Qwen-2-7b、Mistral-7b 和 LLaMA-3-8b 系列模型。在 MATH 和 GSM8K 上的实验结果显示了我们数据的优越性能。

为了提高大型语言模型 (LLMs) 的数学推理能力，提出了多种方法，通常分为两类：(1) 基于提示的方法和 (2) 基于微调的方法。基于提示的方法使用精心设计的提示来引导推理，而不改变模型参数。例如，链式思维 (CoT) 提示模型将问题分解为子步骤，生成通往答案的中间推理路径。基于微调的方法通过指令微调改进开源 LLMs，通常利用数据增广来创建有效的训练数据。拒绝采样微调 (RFT) 使用训练的奖励模型从候选项中选择最佳的推理路径。MetaMath 从多个角度重写问题以扩展训练数据。WizardMath 应用强化使用进化指令技术，

从 GSM8K 和 MATH 中生成新问题，然后将它们用于 SFT 和 PPO 训练以提高推理能力。现有方法通常遵循“问题-答案”工作流，这依赖于 LLM 来生成最终答案。然而，对于复杂问题，有限的推理能力可能导致不可靠的输出。相比之下，我们基于微调的方法采用逆向“答案-问题”工作流来生成可靠的逐步推理数据。

由于众所周知，大型语言模型在算术计算中不太可靠，因此一些研究尝试通过引入代码以提高准确性。例如，GSM8K 数据集在解决方案中嵌入代码以获得中间结果。CSV 方法使用基于提示的代码生成来计算和验证中间步骤。在此基础上，Wang 等人引入了 MathCodeInstruct 数据集，该数据集为每个问题包括自然语言、代码和执行输出，从而能够通过微调来改善推理。然而，很少有方法探索或利用数学问题中固有的结构信息。此外，Xie et al. (2024) 通过解析代码程序获得了这种结构信息，并使用抽象语法树结构化地生成对抗性示例以进行模型评估。在基于代码的方法基础上，我们的工作充分利用程序中的隐含依赖关系来生成准确的中间步骤，并通过扩展代码结构构建更具挑战性的问题。

我们基于 MATH (Hendrycks et al., 2021) 和 GSM8K (Cobbe et al., 2021) 数据集构建了我们的数据集。如图 ?? 所示，我们的方法分为两个阶段。首先，我们指导代理将原始推理过程转化为代码程序，其中相应的推理过程在每个代码块之前进行了注释。在此基础上，我们对代码进行干预，并将修改后的程序转化为新问题和推理过程。此外，我们引入了多级验证机制来控制生成数据的质量。

### 1.1 从推理过程提取结构

对于给定的问题  $q$  及其相应的解决过程  $s = (t_1, t_2, \dots, t_m)$ ，其中  $t_i$  表示自然语言步骤，我们要求代理将推理过程翻译成一个解决方案程序，该程序交织了逐步推理与代码块。代码生成的提示可以在附录 ?? 中找到。

基于此，我们解析程序并将其表示为树状的计算图<sup>1</sup>，如图 ?? 所示。在这个图中，叶子节点表示问题中提到的变量，根节点表示问题的最终答案，而中间节点与不同的推理步骤相关。

值得注意的是，尽管我们的计算结构类似于抽象语法树 (AST) (Xie et al., 2024)，但它通过用中间变量替换操作节点而有所不同。这个设计允许我们捕捉变量之间的依赖关系，并理解每个变量如何对最终结果产生影响。例如，在图 ?? 中，变量 `daises` 影响 `non\_roses` 和

<sup>1</sup>为了便于解释，我们使用树形类比，尽管这种结构允许在严格的树定义上有所偏差。

total。因此，修改 daises 会影响这些变量，并最终改变最后的输出。

## 1.2 结构干预

基于在第 1.1 节获得的结构，生成新问题的最简单方法是修改叶节点（问题中的变量），然后通过计算结构获得新的中间计算步骤和最终结果。值得注意的是，虽然 PAL (Gao et al., 2023) 以类似方式构建新的问题，我们的方法明确地将中间推理步骤与代码相关联，使我们能够通过模式匹配轻松生成新的计算步骤。

尽管上述干预方法产生了新的问题，它仍然在本质上依赖于原始的计算结构。先前的工作 (Saxton et al., 2019; Qi et al., 2024) 指出，长推理问题显著影响 LLM 的性能。从训练的角度来看，长推理问题也可以提高模型的泛化能力 (Saxton et al., 2019)。鉴于这些因素，我们建议利用计算结构来构建需要较长推理步骤的更难的数学问题。

具体来说，给定一个图结构  $G(T, E)$ ，其中  $T = \{t_1, t_2, \dots, t_m\}$  代表对应于各个推理步骤的变量节点， $E$  代表变量关系节点，我们首先从叶节点和中间节点中随机选择一个节点  $t_i$ ，并生成一个代理节点  $t'_i$ ，该节点从  $t_i$  映射而来。即  $t'_i = f(t_i)$ ，其中函数  $f$  可以看作是对  $t_i$  的干预。如图 ?? 所示，我们引入一个代理变量 EXTRA\\_VAR，并用这个新变量替换 daises，用于后续计算。由于编程语言的顺序特性，所有涉及 daises 的计算都会出现在修改后的代码行之后。因此，这种干预可以通过在后续代码中将所有出现的  $t_i$  替换为  $t'_i$  来轻松实现。

为了获得映射函数  $f$ ，由于 GSM8K 问题仅涉及到小学算术，我们将  $f$  设置为简单的加法、减法或乘以一个随机值。需要注意的是，由于除法经常导致不直观或与事实相悖的结果（例如，“0.3 人”），因此在实验中被排除。由于 MATH 问题涉及到更为复杂和特定领域的数学运算，我们允许代理决定如何引入新的推理步骤。更多细节可以在附录 ?? 中找到。

基于修改后的结构，我们要求代理将其转化为一个新的推理过程，并生成一个相应的问题。在此过程中，原始问题和解决代码也作为上下文信息提供给代理。由于修改后的程序与原始程序只有轻微差异，代理只需调整修改的部分。为了指导准确计算，我们将每行代码的执行结果标注为真实值。推理和问题生成的提示在附录 .3 中提供。

## 1.3 质量控制

我们采用多层次验证方法来确保干预数据的质量，包括（1）本地代码执行和（2）代理评估。

| Dataset   | GSM8K $tr$ | MATH $tr$ | GSM8K $te$ | MATH $te$ |
|-----------|------------|-----------|------------|-----------|
| Code Gen. | 7.1K       | 6.0K      | 1.3K       | 3.9K      |
| Extension | 14.3K      | 11.6K     | 3.2K       | 2.9K      |
| Total     | 39.0K      |           | 11.3K      |           |

Table 1: 在所提议的数据集中代数的数量，其中  $tr$  表示训练集， $te$  表示测试集，“Code Gen”代表直接从原始推理过程翻译的样本，“Extension”代表在结构扩展后获得的样本。

对于局部验证，我们将生成代码的执行结果与原始文本中的真实情况进行比较，以验证在第 1.2 节中生成的代码的正确性。对于结构性干预，我们将修改后的代码中每个变量的执行结果与原始结果进行比较，并过滤掉出现数值符号反转（例如，正数变为负数）或数据类型更改（例如，整数变为浮点数）的情况。

尽管提供了推理过程的真实答案，大型语言模型 (LLMs) 仍可能生成反事实的输出或产生不匹配的问题和推理过程。为了缓解这一问题，在代理评估中，我们首先要求代理在生成过程期间进行自我评估。此外，为了防止其输出过于自信（即情境意识问题 (Berglund et al., 2023)）并减少单一模型引入的认知偏差，我们采用来自不同系列的外部模型来评估生成结果。只有通过自我评估和外部代理评估的数据才会被保留。在这项研究中，我们使用 Claude-3-5-Sonnet 模型进行数据生成，并使用 GPT-4o 进行评估。虽然我们的方法对所用模型没有严格要求，但我们发现 Claude-3-5-Sonnet 相较于 GPT-4o 生成了语义更丰富的问题。我们还进行了人工评估，以评估和审查生成数据的质量，详细信息在附录 ?? 中提供。

在本节中，我们首先介绍所提出数据集的详细信息。然后，我们在所提出的基准上测试最先进的大型语言模型 (LLMs)。最后，我们通过展示在 Mistral、Qwen 和 Llama 系列模型上的微调结果，证实我们数据的有效性。我们使用两个广泛使用的数学数据集生成我们的数据：(1) GSM8K，这是一个小学数学数据集，包含 7,473 个训练样本和 1,319 个测试样本；(2) MATH，它包含 7,500 个训练样本和 5,000 个测试样本，涵盖五个难度级别和七个科目的高中竞赛级问题。两个数据集都提供完整的解题过程，这促进了我们方法的代码生成过程。表格显示了从 GSM8K 和 MATH 生成的样本数量。我们通过仅在之前成功的生成基础上进行干预来应用渐进扩展。对于训练集 GSM8K 和 MATH，每个样本最多经历三轮干预，而对于测试集，我们允许最多八轮。如果在对一个问题进行干预后没有产生有效样本，我们就停止

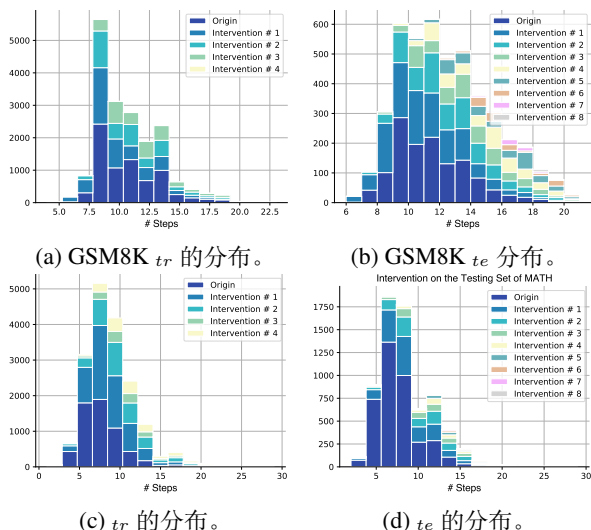


Figure 1: 不同数据组件中的步骤分布。

对该问题的进一步尝试。这个过程重复三次，结果总共生成了 5 万对带有中间推理步骤的问题和答案。其中，来自原始训练集的 3.9 万个样本（包括“代码生成”和“扩展”）用于模型训练，而来自测试集的 6,100 个“扩展”样本则用作测试基准，以评估模型在更具挑战性数据上的性能。我们将我们的基准称为“Math-Struct”和“GSM8K-Struct”。

**难度分布** 我们的方法将整体推理过程分解为单个步骤。正如在 (Qi et al., 2024) 中指出的，问题难度受任务的内在复杂性影响，这体现在所需推理步骤的数量上。基于此，我们通过测量每个问题中的步骤数量来分析原始样本和干涉样本的难度分布。

具体而言，在获得给定问题的解答代码后，我们统计第一级缩进的代码块数量作为求解步骤的近似值。使用这一衡量标准，我们绘制了在原始数据和每次干预后的干预数据中求解步骤数量的频率分布直方图，结果如图 1 所示。

总体而言，与原始分布相比，干预后的步长分布向右移动，表明数据集难度增加。对原始和干预分布的进一步比较显示，大多数干预后的数据点位于原始分布的右侧。这表明我们的干预方法生成了更多具有更长求解步骤的问题。

从干预轮次 #1 到干预轮次 #8 的趋势来看，我们观察到随着生成轮次的增加，问题变得更加复杂和困难。这种复杂性的增加使得模型更难理解这些问题，导致成功生成的数据量减少。为了清楚地了解各种模型在不同难度级别问题上的性能，我们使用最先进的模型对原始问题及其扩展版本进行了广泛的评

估。包括 GPT-o1<sup>2</sup> (OpenAI, 2024b)、GPT-4o (OpenAI, 2024a)、Claude-3-5-Sonnet (Anthropic, 2024)、Qwen2.5-MATH-Instruct (Yang et al., 2024)、Llama-3.3 (Llama team, 2024) 和 Mistral-Large-Instruct (Mistral AI team, 2024a)。我们评估了更多的模型，并在附录 ?? 中提供了它们的结果。

我们采用了 MathCoder<sup>3</sup> 使用的提示格式，要求模型在 `\boxed{}` 标签内包含它们的答案。所有模型都在零样本设置下进行测试，并使用基于单次生成的准确性作为评估指标。

图 2 显示了不同模型在不同难度级别问题上的表现整体而言，随着推理步骤的增加，所有模型的性能都有所下降，这支持了现有观点，即较长的推理步骤对应更具挑战性的问题。如图 1 所示，在 MATH 数据集中，超过 15 步的样本较少，这导致了在这个范围内模型性能的更大波动。

由于 Claude-3-5-Sonnet 和 GPT-4o 参与了数据生成过程，生成的数据不可避免地携带了一些它们的偏见。Claude-3-5-Sonnet 和 GPT-4o 在数据生成时倾向于保留它们能够理解的问题，这有助于它们整体上表现优异。然而，需要强调的是，在生成阶段，模型完全依赖于它们的理解和创造力，因为每一步都提供了监督信息。相比之下，在评估阶段，缺乏这样的指导信息，它们的表现取决于它们的推理能力。

值得注意的是，GPT-o1 模型的性能低于我们的预期。在附录 .2 中，我们检查了 GPT-o1 产生错误答案的样本，发现 GPT-o1 在长推理过程中往往忽略关键信息并出现简单的计算错误。由于据报道 o1 存在“变笨”的问题<sup>4</sup>，因此需要注意目前的评估结果可能无法准确反映 o1 的真实性能。

在附录 .1 中，我们评估了使用不同干预轮次生成的数据集上的模型性能。我们观察到，随着轮次增加，性能下降，这表明额外的干预导致问题变得更加具有挑战性。为了处理自动生成可能带来的质量问题，在附录 ?? 中，我们关注于由 Claude-3.5-Sonnet 和 GPT-4o 两者都进行正确推理的样本子集。然后，我们对这些经过验证的样本的合并进行所有模型的评估。正如附录 ?? 所示，这个子集上的性能趋势与图 4 中的趋势一致，进一步确认了我们数据的可靠性。

<sup>2</sup>为了节省成本，我们仅在扩展数据集上评估了 GPT-o1 的性能。

<sup>3</sup><https://github.com/mathllm/MathCoder/blob/main/src/inference.py>

<sup>4</sup><https://community.openai.com/t/is-anyone-elses-gpt-4o-and-o1-suddenly-acting-dumb>

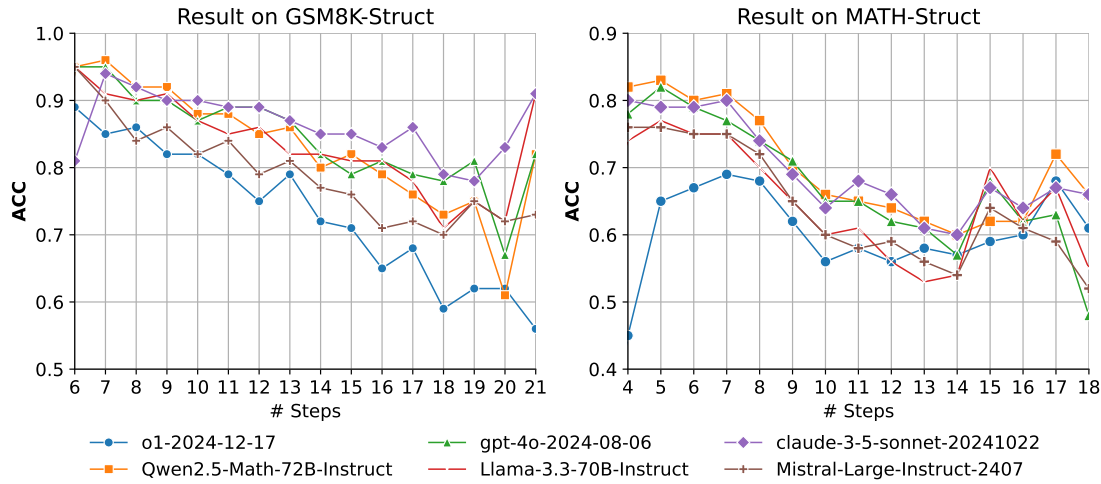


Figure 2: 不同步骤的基准测试结果。

在这一部分，我们使用生成的数据进行模型训练，以验证生成的数据在提高模型推理性能方面的有效性。根据表格 1，我们从  $\text{GSM8K}_{tr}$  和  $\text{MATH}_{tr}$  数据集里选择了总共 39K 的数据，包括代码生成和扩展任务，用于模型训练。我们比较了 MathCoder (Wang et al., 2024a)、MetaMath (Yu et al., 2024)、Math-Shepherd (Wang et al., 2024b)、MammoTH (Yue et al., 2024) 和 MathScale (Tang et al., 2024b) 数据集。为了公平比较，我们从每个数据集中选择了 39K 的数据以与我们的数据集尺寸匹配。由于 MetaMath 包含了四类：‘AnsAug’，‘FOBAR’，‘Rephrase’，‘Self-Verification’，对于每种类别，我们平均选择了 4875 个样本。对于其他数据集，我们随机选择了 39K 个样本。

我们选择了四个基线 SFT 模型：Mistral-7b-v0.1 (Jiang et al., 2023)、Qwen-2-7b (Qwen Team, 2024) 和 Meta-Llama-3-8b (Llama team, 2024)。使用 LlamaFactory 训练框架 (Zheng et al., 2024)，我们在 8 个 A100 GPUs 上训练这些模型。我们将序列长度设置为 2048，并对每个模型进行两轮的全参数微调。对于所有模型，批量大小设置为 256。所有模型的学习率从集合  $\{1e-5, 2e-5, 2e-6\}$  中选择。由于 MathCoder 和我们的数据集中都包含 Python 代码，我们引入了两个特殊的标记，`<start\_code>` 和 `<end\_code>`，以便在使用这些数据集进行微调时清楚地区分代码和文本内容。对于其他模型，我们根据原始论文 (Yu et al., 2024) 采用 Alpaca 风格的 (Taori et al., 2023) 提示进行训练。训练提示在附录 .5 中提供。

#### 1.4 原始数据集上的结果

我们首先使用原始的 GSM8K 和 MATH 数据集评估了在不同数据集上训练的模型，如表 ?? 所示。我们的数据集实现了最佳的总体性能，大多数情况下排名前两名。由于指令跟随能力有限，Mistral-7B 和 Llama-3-8B 基线表现不佳，但在微调后显示出显著的提升。我们观察到，在大多数数据集上微调 Qwen-2-7B 会导致性能下降，这可能是由于灾难性遗忘所致。我们在附录 A 中对此问题进行了详细分析。

我们进一步评估了微调模型在所提出基准上的表现，并在图 ?? 中展示了各模型在不同难度问题上的表现分布。与图 2 类似，所有方法在问题的推理长度增加时表现均有所下降。然而，整体上，使用我们数据微调的模型在所有难度级别的问题上表现更好。值得注意的是，对于 Mistral-7b 和 Qwen-2-7b 模型，MetaMath、MammoTH 和我们的方法在较容易的问题上表现出相当的推理性能，但当问题难度增加时，它们的性能迅速下降。这表明，我们的数据有助于提高大规模语言模型的长程推理能力。

#### 1.5 框架的多功能性

鉴于 GSM8K 和 MATH 在数据生成研究中的广泛使用，并包含大量具有逐步推理过程的问答数据，我们将我们的数据生成基于这两个数据集。然而，我们强调我们的方法不仅限于 GSM8K 和 MATH。为了展示其更广泛的适用性，我们将我们的生成框架应用于更难的数学问题，并在下面展示了 AIME\_2024 (MAA, February 2024) 数据集上的一个生成示例。由于空间限制，我们仅展示原始问题和生成问题之间的区别。

```
Origin: ...She wins a prize if at least
        two of her numbers were          $2$
        of the randomly chosen
        numbers, and wins the grand prize if
        all four of her numbers were the
        randomly chosen numbers...
```

```
New: ...She wins a small prize if
      exactly two of her numbers were
      among the randomly chosen numbers, a
      medium prize if exactly three
      numbers match, and the grand prize
      if all four of her numbers match...
```

此外，我们的方法不仅限于数学问题，还可以应用于任何可以通过代码表示推理过程的任务。为了说明这一多功能性，我们展示了一个基于 MBPP (Austin et al., 2021) 数据集并通过我们的干预框架生成的编程问题：

```
Origin: Write a function to find minimum
        of three numbers.
```

```
New: Write a function that finds the
      minimum and maximum of three numbers
      and returns a tuple containing: (1)
      the minimum value, (2) the count of
      how many times this minimum value
      appears, (3) the maximum value, and
      (4) the count of how many times this
      maximum value appears among the
      three numbers. Additionally, if the
      minimum and maximum are the same
      number (meaning all three numbers
      are equal), return (-1, -1, -1, -1).
```

欲获取上述示例的完整问题和解决方案，请参阅附录 ??。

在这篇论文中，我们探讨了结构信息在数学推理中的作用，并展示了其各种应用和重要性。基于结构数据，我们可以通过计算推理步骤来量化问题的难度，在中间步骤获取监督，并干预结构以生成更具挑战性的样本。据我们所知，目前尚无研究利用结构信息对推理问题的难度进行量化评估和干预。我们强调，结构信息的效用不仅仅限于这些应用。通过使用代码作为媒介，推理结构形成了一个比自然语言更狭窄且更可预测的通道，使我们能够追踪当前变量如何影响后续计算。此外，计算结构固有的有向无环性质为推理步骤提供了拓扑顺序，可以用于指导和限制结构干预。我们计划在未来的工作中进一步探索这种拓扑结构的潜力。

## 局限性

与其他数据生成方法相比，我们研究的主要局限性是生成过程效率较低，这很大程度上是由于使用了多级验证策略来确保数据质量。由于生成的数据可能包含个体被分配过多工作量或食物的情况，虽然这不会影响算术本身，但常常被 GPT-4o 基于常识判断标记为不正确。因此，只有 51.4 % 的 GSM8K 和 35.4 % 的 MATH 样本被保留。虽然成功生成了大约 42 % 的 MATH 生成结果和 21 % 的 GSM8K 生成结果，但在 GPT-4o 评估后被舍弃。为了确保数据质量，我们在分析中没有使用这些被丢弃的样本。然而，值得注意的是，由于我们的方法潜在地确保了计算过程的正确性，这些被丢弃的数据仍包含具有正确计算的监督过程。正如在 (Yu et al., 2024) 中指出的那样，错误的推理过程仍可能有助于提高模型的推理能力。为了验证这一点，我们结合失败和成功的生成以训练 LLM 在混合数据集上。结果和分析在附录 B 中提供。结果显示，包含错误数据似乎可以提高对数学问题的性能，这表明在未来工作中有进一步探索的潜力。

另一个导致生成效率低的因素是我们的过程是增量式的而非重复性的：第  $(k + 1)$  次生成步骤是基于第  $k$  次干预的样本，逐步增加了问题的难度。虽然我们的生成过程减缓了模型的推理能力，但随着推理步骤的增加，模型逐渐难以理解所呈现的问题和推理步骤，使生成过程更容易失败。虽然反复干预可能有助于增加数据量，如我们之前提到的，这会导致大量数据被丢弃。考虑到我们生成的数据涉及与原始问题不同且难度更高的推理过程，一个更好的方法可能是将其与基于释义的现有方法结合，从而更有效地增加数据量。这将是未来研究的方向。

## References

- Anthropic. 2024. [Claude 3.5 sonnet](#).
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. 2023. [Taken out of context: On measuring situational awareness in llms](#). *Preprint*, arXiv:2309.00667.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- 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 preprint arXiv:2110.14168*.
- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [Pal: Program-aided language models](#). *Preprint*, arXiv:2211.10435.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. [Step-dpo: Step-wise preference optimization for long-chain reasoning of llms](#). *Preprint*, arXiv:2406.18629.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*.
- Llama team. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei

- Zhang. 2025. [Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct](#). *Preprint*, arXiv:2308.09583.
- Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. [Left: Reasoning with reinforced fine-tuning](#). *Preprint*, arXiv:2401.08967.
- MAA. February 2024. [American invitational mathematics examination - aime](#). In *American Invitational Mathematics Examination - AIME 2024*.
- Mistral AI team. 2024a. [Au large](#).
- Mistral AI team. 2024b. [Mistral nemo](#).
- Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. [Codegen: An open large language model for code with multi-turn program synthesis](#). In *The Eleventh International Conference on Learning Representations*.
- OpenAI. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- OpenAI. 2024a. [Hello gpt-4o](#).
- OpenAI. 2024b. [Learning to reason with llms](#).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.
- Zhenting Qi, Hongyin Luo, Xuliang Huang, Zhuokai Zhao, Yibo Jiang, Xiangjun Fan, Himabindu Lakkaraju, and James Glass. 2024. [Quantifying generalization complexity for large language models](#). *Preprint*, arXiv:2410.01769.
- Zhenting Qi, Hongyin Luo, Xuliang Huang, Zhuokai Zhao, Yibo Jiang, Xiangjun Fan, Himabindu Lakkaraju, and James R. Glass. 2025. [Quantifying generalization complexity for large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Qwen Team. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. [Analysing mathematical reasoning abilities of neural models](#). *Preprint*, arXiv:1904.01557.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.
- Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. [Defining and characterizing reward hacking](#). *Preprint*, arXiv:2209.13085.
- Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024a. [Mathscale: Scaling instruction tuning for mathematical reasoning](#). *Preprint*, arXiv:2403.02884.
- Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024b. [Mathscale: Scaling instruction tuning for mathematical reasoning](#). In *Forty-first International Conference on Machine Learning*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. [Stanford alpaca: An instruction-following llama model](#). [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca).
- Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. 2023. [On the planning abilities of large language models \(a critical investigation with a proposed benchmark\)](#). *Preprint*, arXiv:2302.06706.
- Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2024a. [Math-coder: Seamless code integration in LLMs for enhanced mathematical reasoning](#). In *The Twelfth International Conference on Learning Representations*.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. [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, Bangkok, Thailand. Association for Computational Linguistics.
- Roy Xie, Chengxuan Huang, Junlin Wang, and Bhuwan Dhingra. 2024. [Adversarial math word problem generation](#). *Preprint*, arXiv:2402.17916.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. [Metamath: Bootstrap your own mathematical questions for large language models](#). *Preprint*, arXiv:2309.12284.

- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. 2024. [MAmmoTH: Building math generalist models through hybrid instruction tuning](#). In *The Twelfth International Conference on Learning Representations*.
- Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. [Agenttuning: Enabling generalized agent abilities for llms](#). Preprint, arXiv:2310.12823.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. [Llamafactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.
- Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023. [Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification](#). Preprint, arXiv:2308.07921.

包括来自 GPT 系列的模型 ( GPT-o1, GPT-4o) (OpenAI, 2024b,a) , Claude 系列 ( Claude-3-5-Sonnet) (Anthropic, 2024) , Qwen 系列 ( Qwen2.5-MATH-Instruct, Qwen-2.5-Instruct) (Yang et al., 2024) , Llama-3 系列 ( Llama-3.3, Llama-3.1 (Llama team, 2024) ), 以及 Mistral 系列 ( Mistral-Large-Instruct, Mistral-Nemo-Instruct) (Mistral AI team, 2024a,b) 。评估结果如图 3 所示。在同一系列的模型中比较, 我们观察到较大的模型通常在需要较长推理步骤的问题上表现更好。这表明拥有更多参数的模型具有更强的长步骤推理能力。

### 1.1 不同干预次数的结果

请注意, 我们使用一种渐进的方法生成数据, 其中每个后续干预都是基于前一个生成的数据。在本节中, 我们展示了不同模型在每个阶段干预数据上的表现。如图 4 所示, 随着干预阶段的推进, 模型性能普遍下降, 这表明生成数据的难度随着每个阶段的增加而增加。

考虑到在模型生成过程中可能仍然会发生错误, 我们通过 Claude-3-5-Sonnet 和 GPT-4o 的性能过滤数据, 进一步验证评估结果的可靠性。具体来说, 我们假设 Claude-3-5-Sonnet 和 GPT-4o 正确回答的问题是正确。在此假设的基础上, 我们选择被 Claude-3-5-Sonnet 和 GPT-4o 正确回答的问题, 并将两者的并集作为过滤后的正确数据子集。然后我们观察所有模型在该子集上的性能变化, 如图 5 所示。与图 2 相比, 我们可以看到过滤后数据集上的性能趋势与完整数据集的趋势相似。这进一步表明, 推理步骤数量的增加提高了问题的难度, 导致模型性能逐渐下降。

为了进一步验证我们数据集的质量, 我们进行了人为评估。我们从 MATH-Struct 和 GSM8K-Struct 中随机抽取了 30 个实例。为确保评估质量, 我们邀请了两两位志愿标注者评估数据集, 具体包括: (1) 合格的标注者: 两位标注者都非常合格, 在计算机科学领域有超过四年的经验。(2) 标注者之间的一致性: 标注者们通过合作解决分歧并确保一致性, 以保持高度的一致性和可靠性。

对于每个问题, 我们提供相应的标准答案和推理过程。然后, 注释人员必须确定问题是否:

1. 有效问题 (VQ): 问题明确、无歧义, 并且可以合理解决。
2. 不清楚或模糊的问题 (UAQ): 问题含糊不清或可以有多种解释。

3. 不完整问题 (IQ): 问题缺乏关键信息, 因此无法解决。

4. 多重可能答案 (MPA): 由于缺少约束条件, 该问题允许多个正确的解决方案。

5. 与真实值不匹配 (MGT): 该问题与给定的真实值不一致, 使得评估不可靠。

我们的人工评估结果如图 6 所示。可以观察到, 有效问题 (VQ) 几乎占到 90.0 % , 这进一步证实了我们数据集的高质量。

从图 7 到图 11 , 我们展示了 Claude-3-5-Sonnet 和 GPT-4o 无法正确回答的例子, 这表明我们的方法可以生成超出代理能力的问题。

为了证明其更广泛的适用性, 我们将我们的框架应用于更复杂的数学问题 (AIME\\_{2024} (MAA, February 2024) ) 并在图 12 中展示了一个生成的示例。在生成的问题中, 添加的步骤 `istotal\_medium\_or\_better = ways\_three\_match + ways\_four\_match` 修改了条件概率计算, 以仅考虑至少有中等奖 (3 或 4 个匹配) 的情形, 这不同于原始问题, 其中包括了所有获奖情形 (2、3 或 4 个匹配)。

此外, 我们的方法适用于任何可以通过代码表示推理过程的任务。为了展示这种多功能性, 我们将其应用于编程问题。图 13 展示了我们框架在 MBPP (Austin et al., 2021) 数据集上的输出。

### 2 GPT-o1 的定性错误分析

图 14 到 17 包括一些由 OpenAI 的 GPT-o1 模型给出的错误答案的典型示例。

图 14 显示该模型未能处理短语 “但改变了主意, 决定不再购买任何餐具”, 导致总价计算错误。这表明在理解否定和情境变化方面存在不足, 暗示该模型在处理对先前信息的修改时存在困难。提高其识别和适应此类情境变化的能力将提升类似任务的准确性。

从图 15 到图 17 的问题突出了一个关键弱点: 模型在处理大型数值计算时挣扎, 从而导致错误。这表明它在准确处理复杂算术方面存在局限性。

我们在图 18 中展示了代码生成的提示。

### 3 代码翻译提示

我们在图 19 中展示了用于 GSM8K 数据集的代码生成提示, 并在图 20 中展示了用于 MATH 数据集的代码生成提示。由于 MATH 中问题的专业性和复杂性更高, 我们选择让代理来确定具体的干预措施, 从而生成更合理的问题。

### 4 用于 GPT 评估的提示

我们在图 21 中提供了 GPT-4o 的评估提示。

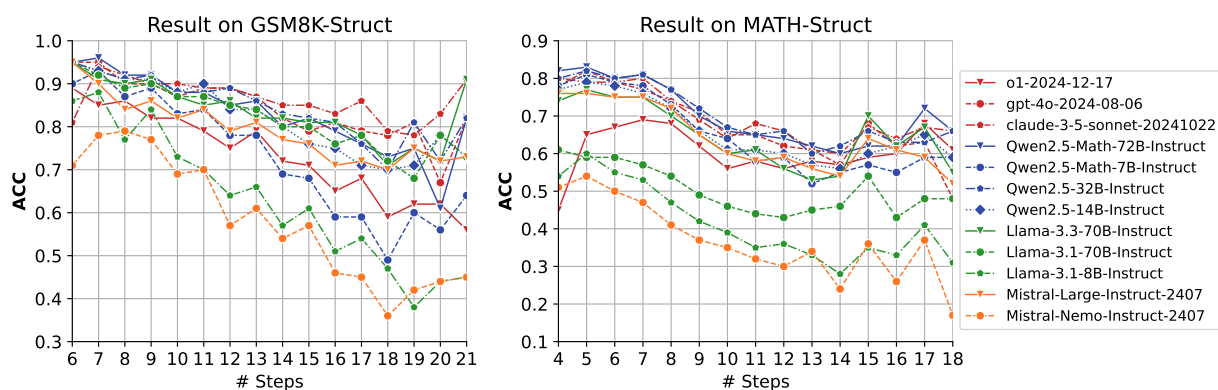


Figure 3: 额外的基准测试结果。

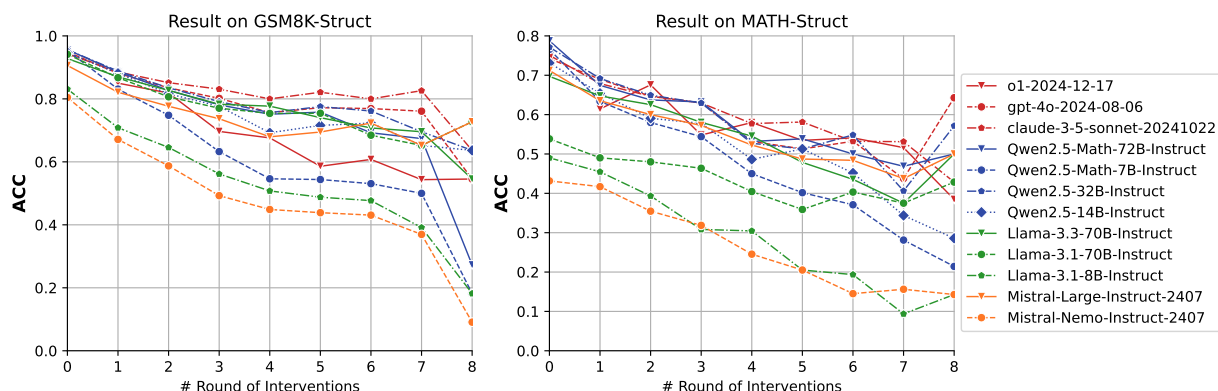


Figure 4: 不同干预次数的基准测试结果。

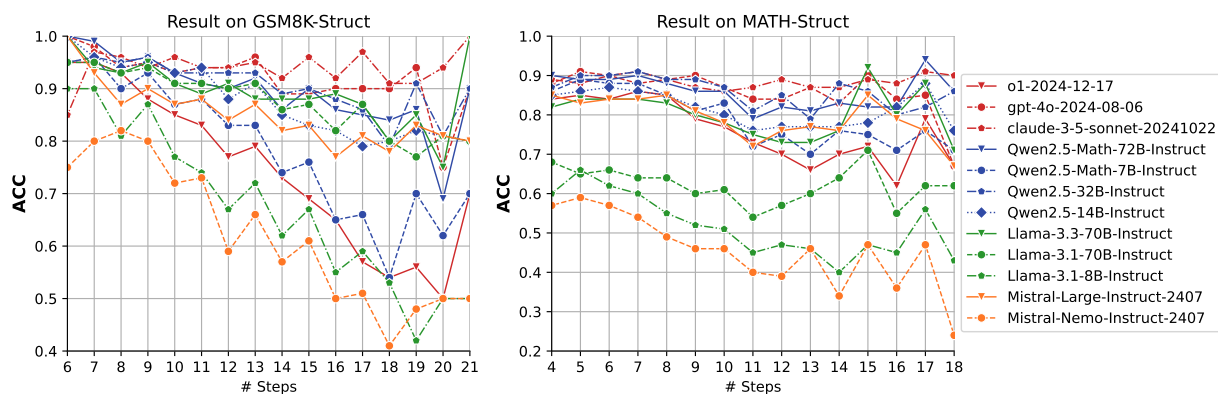


Figure 5: 筛选子集在不同推理步骤上的基准测试结果。

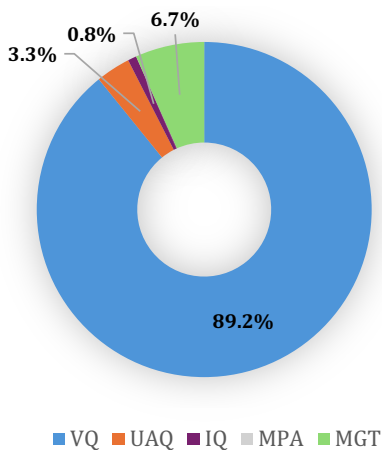


Figure 6: 人工评估结果。

## .5 用于训练和推理的提示

我们使用相同的提示格式进行训练和推理，如图 22 所示。

### A 灾难性遗忘分析

在表 ?? 中，我们观察到 Qwen-2-7B 在大多数数据集上进行微调时性能下降，我们将其归因于灾难性遗忘。为了验证这一点，我们测量了由预训练模型正确回答但在使用我们数据集进行微调后回答错误的 MATH 和 GSM8K 样本比例。如表 2 所示，大约 9.1 % 的 MATH 样本从正确变为错误。值得注意的是，预训练模型在 MATH 上的准确性达到 46.2 %，比微调模型高出 6.5 % (39.7 %)，这低于观察到的遗忘率。这支持了我们的假设，即性能损失主要是由于微调过程中的知识遗忘。

| GAM8K | MATH  |
|-------|-------|
| 2.5 % | 9.1 % |

Table 2: 在微调之后，MATH 和 GSM8K 数据集集中的误导性答案比例。

为了进一步支持这一结论，我们展示了预训练和微调的 Qwen-2-7B 模型在同一问题上的输出。如图 23 所示，预训练模型往往会产生更长的回答，并积极利用其背景知识来解决问题。相比之下，微调模型提供的回答更受限，仅专注于眼前的问题。鉴于解决数学问题通常需要检索并应用数学知识，我们假设微调抑制了这种检索能力，从而导致了观察到的性能下降。

### B 通过失败生成进行训练

为了探索错误数据的潜在价值，我们结合了失败和成功的生成来训练 LLM，并评估它们在

MATH 和 GSM8K 数据集上的表现。如表 3 所示，尽管结合的数据略微降低了 GSM8K 的性能，但却改善了 MATH 的数据表现，这表明错误数据可能在更复杂的推理任务中仍然提供有用的学习信号。这些结果表明，引入错误数据可以进一步增强 MATH 中的专门数学知识，从而提高模型性能。

|            | Dataset | Success (39K) | Success+Failure (100k) |
|------------|---------|---------------|------------------------|
| Mistral-7b | GSM8K   | 68.31 %       | 67.70 % ↓              |
|            | MATH    | 16.90 %       | 18.62 % ↑              |
| Llama-3-8b | GSM8K   | 71.34 %       | 70.51 % ↓              |
|            | MATH    | 19.70 %       | 22.66 % ↑              |

Table 3: 使用失败生成进行微调。

### Example from GSM8K

Original Question:

A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

Answer:

#### Original Solution Code

```
def solution():
    # <reason>The robe takes 2 bolts of blue fiber.</reason>
    blue_fiber_bolts = 2 # 2
    # <reason>The white fiber needed is half of the blue fiber, so it takes  $2/2=1$  bolt of white fiber.</reason>
    white_fiber_bolts = blue_fiber_bolts / 2 # 1.0
    # <reason>The total amount of fabric needed is  $2+1=3$  bolts.</reason>
    total_bolts = blue_fiber_bolts + white_fiber_bolts # 3.0
    # <reason>The result is 3 bolts in total.</reason>
    result = total_bolts # 3.0
    return result
```

New Question:

A grand ceremonial robe takes 2 bolts of blue fiber. The design first needs to be made 5 times larger. Due to a rare celestial event, the design must be made 9 times more intricate before being made 6 times more elaborate for the winter festival. The white fiber needed is still half of the blue fiber amount. Due to an unexpected surge in festival attendance, the grand master needs 162 extra bolts. Furthermore, due to a special royal decree, an additional 810 bolts are required for ceremonial banners. How many more bolts in total are needed compared to the original estimate of 18 bolts?

Answer:

#### New Solution Code

```
def solution():
    # <reason>The basic robe design starts with 2 bolts of blue fiber.</reason>
    blue_fiber_bolts = 2 # 2
    # <reason>First, the design is made 5 times larger, requiring  $2 * 5 = 10$  bolts.</reason>
    larger_design_bolts = blue_fiber_bolts * 5 # 10
    # <reason>Due to the celestial event, the design must be made 9 times more intricate.</reason>
    op_var = 9 # 9
    # <reason>The intricate design now requires  $10 * 9 = 90$  bolts.</reason>
    extra_var = larger_design_bolts * op_var # 90
    # <reason>Then, for the festival version, we need 6 times as much blue fiber as the intricate design, so  $90 * 6 = 540$  bolts of blue fiber.</reason>
    festival_blue_fiber_bolts = extra_var * 6 # 540
    # <reason>The white fiber needed is half of the blue fiber, so it takes  $540/2 = 270$  bolts of white fiber.</reason>
    white_fiber_bolts = festival_blue_fiber_bolts / 2 # 270.0
    # <reason>The new design requires a total amount of  $540 + 270 = 810$  bolts.</reason>
    total_bolts = festival_blue_fiber_bolts + white_fiber_bolts # 810.0
    # <reason>The initial difference between the new design requirement (810 bolts) and the original estimate (18 bolts) is  $810 - 18 = 792$  bolts.</reason>
    bolts_difference = total_bolts - 18.0 # 792.0
    # <reason>With the 162 extra bolts needed for increased attendance, the total additional requirement becomes  $792 + 162 = 954$  bolts more than the original estimate.</reason>
    additional_bolts_needed = bolts_difference + 162.0 # 954.0
    # <reason>Adding the 810 bolts required by the royal decree for ceremonial banners, we need  $954 + 810 = 1764$  more bolts than originally estimated.</reason>
    total_additional_bolts = additional_bolts_needed + 810.0 # ?
    # <reason>The final result is 1764 bolts more than originally estimated.</reason>
    result = total_additional_bolts # ?
    return result
```

Figure 7: 来源于 GSM8K 的扩展问题。

## Example from GSM8K

### Answer by Claude-3-5-Sonnet

Answer:

Let me help solve this step by step.

1) Let's first establish the base amount:

- Original estimate = 18 bolts

2) Calculate how the design changes affect the amount:

- 5 times larger  $\times$  9 times more intricate  $\times$  6 times more elaborate
- $5 \times 9 \times 6 = 270$  times the original amount

3) Calculate new amount after design changes:

- $18 \times 270 = 4,860$  bolts

4) Add extra bolts needed:

- Additional due to surge: 162 bolts
- Additional for banners: 810 bolts
- Total extra:  $162 + 810 = 972$  bolts

5) Calculate total new amount:

- $4,860 + 972 = 5,832$  bolts

6) Find difference from original estimate:

- $5,832 - 18 = 5,814$

Therefore, 5,814 more bolts are needed compared to the original estimate. 

### Answer by GPT-4o

Answer:

To solve the problem step by step, let's break down each part of the problem and make calculations as necessary:

1. Initial Blue Fiber Requirement:

- Initial requirement for the robe is 2 bolts of blue fiber.

2. Design Enlargement:

- The design needs to be made 5 times larger. If the base requirement is 2 bolts, then  $2 \times 5 = 10$  bolts are needed just for the size increase.

3. Increased Intricacy:

- The design must be made 9 times more intricate. Hence, multiplying the enlarged design's requirement by 9 gives us  $10 \times 9 = 90$  bolts.

4. Elaboration for the Winter Festival:

- The design must then be made 6 times more elaborate. Therefore, multiplying the requirement by 6 yields:  $90 \times 6 = 540$  bolts of blue fiber.

5. White Fiber Requirement:

- The white fiber required is half of the blue fiber amount. Therefore, if the blue fiber needed is 540 bolts, then the white fiber requirements would be  $\frac{540}{2} = 270$  bolts.

6. Total Fiber Requirement (excluding extras):

- Blue fiber: 540 bolts
- White fiber: 270 bolts
- Total without extras:  $540 + 270 = 810$  bolts

7. Additional Bolt Requirements:

- An additional 162 bolts are needed due to a surge in festival attendance.
- An additional 810 bolts are required due to a special royal decree for ceremonial banners.

8. Total Additional Bolts:

- Additional from festival attendance: 162 bolts
- Additional from royal decree: 810 bolts
- Total additional bolts:  $162 + 810 = 972$  bolts

9. Total Bolts Needed:

- Total from the enlarged, intricate, and elaborate design plus additional requirements:  $810 + 972 = 1782$  bolts

10. Comparison with the Original Estimate:

- The original estimate is 18 bolts.

11. Difference in Requirement:

- Total more bolts needed compared to the original estimate:  $1782 - 18 = 1764$ .

Therefore, the total number of additional bolts needed compared to the initial estimate is  $\boxed{1764}$ .



Figure 8: 由 Claude-3-5 Sonnet 和 GPT-4o 生成的答案。

### Example from MATH

Original Question:

If  $5x - 3 = 12$ , what is the value of  $5x + 3$ ?

Answer:

#### Original Solution Code

```
def solution():
    # <reason>From the equation  $5x - 3 = 12$ , we can solve for  $5x$  by adding 3 to both sides.</reason>
    right_hand_side = 12
    constant_term = 3
    # <reason>Adding 3 to both sides gives  $5x = 12 + 3 = 15$ .</reason>
    five_x = right_hand_side + constant_term
    # <reason>Now for the expression  $5x + 3$ , we take the value of  $5x$  which is 15 and add 3.</reason>
    # <reason>Therefore,  $5x + 3 = 15 + 3 = \boxed{18}$ .</reason>
    result = five_x + constant_term
    return result
```

New Question:

If  $5x - 3 = 12$ , what is the value of  $(5x + 3)^5 - (5x - 3)^5 + (5x)^5$ ?

Answer:

#### New Solution Code

```
def solution():
    # <reason>From the equation  $5x - 3 = 12$ , we can solve for  $5x$  by adding 3 to both sides.</reason>
    right_hand_side = 12
    constant_term = 3
    # <reason>Adding 3 to both sides gives  $5x = 12 + 3 = 15$ .</reason>
    five_x = right_hand_side + constant_term
    # <reason>Now for the first expression  $5x + 3$ , we take the value of  $5x$  which is 15 and add 3.</reason>
    expression1 = five_x + constant_term
    # <reason>Therefore,  $5x + 3 = 15 + 3 = 18$ .</reason>
    # <reason>For the second expression  $5x - 3$ , we already know it equals 12.</reason>
    expression2 = right_hand_side
    # <reason>For the third expression  $5x$ , we know it equals 15.</reason>
    expression3 = five_x
    # <reason>For fifth powers, we can use the difference of fifth powers formula:  $a^5 - b^5 = (a - b)(a^4 + a^3b + a^2b^2 + ab^3 + b^4)$ .</reason>
    a = expression1
    b = expression2
    # <reason>First calculate  $a - b = 18 - 12 = 6$ .</reason>
    difference = a - b
    # <reason>Now we can compute  $(5x + 3)^5 - (5x - 3)^5 = 18^5 - 12^5 = (18 - 12)(18^4 + 18^3 * 12 + 18^2 * 12^2 + 18 * 12^3 + 12^4)$ .</reason>
    result1 = difference*(a**4 + a**3*b + a**2*b**2 + a*b**3 + b**4)
    # <reason>Finally, add  $(5x)^5 = 15^5$  to get the final result.</reason>
    result = result1 + expression3**5
    return result
```

Figure 9: 扩展自数学的问题。

### Example from MATH

#### Answer by Claude

Answer:

Let me solve this step by step.

1) First, from  $5x - 3 = 12$

- Let's solve for  $5x$

- $5x = 15$

- Therefore,  $x = 3$

2) Now, let's substitute  $5x = 15$  into each expression:

- $5x + 3 = 18$

- $5x - 3 = 12$

- $5x = 15$

3) Our expression is  $(5x + 3)^5 - (5x - 3)^5 + (5x)^5$

- This becomes  $18^5 - 12^5 + 15^5$

4) Let's calculate each term:

- $18^5 = 1,889,568$

- $12^5 = 248,832$

- $15^5 = 759,375$

5) Now let's do the final calculation:

- $1,889,568 - 248,832 + 759,375$

- $1,889,568 + 759,375 - 248,832$

- $2,648,943 - 248,832$

- $2,400,111$

Therefore,  $\boxed{2,400,111}$  ✓

Figure 10: Claude-3-5-Sonnet 的正确答案。

## Example from MATH

### Answer by GPT-4o

Answer:

To solve the problem, we start by determining the value of  $x$  from the equation  $5x - 3 = 12$ .

Step 1: Solve for  $x$

$$5x - 3 = 12$$

Add 3 to both sides to isolate the term with  $x$ :

$$5x = 12 + 3$$

$$5x = 15$$

Divide both sides by 5 to solve for  $x$ :

$$x = \frac{15}{5}$$

$$x = 3$$

Step 2: Find the values of  $5x + 3$ ,  $5x - 3$ , and  $5x$ , and then use these to find  $(5x + 3)^5 - (5x - 3)^5 + (5x)^5$   
For  $x = 3$ :

$$5x = 5 \times 3 = 15$$

$$5x + 3 = 15 + 3 = 18$$

$$5x - 3 = 15 - 3 = 12$$

Now compute:

$$(5x + 3)^5 = 18^5$$

$$(5x - 3)^5 = 12^5$$

$$(5x)^5 = 15^5$$

We want to find:

$$(5x + 3)^5 - (5x - 3)^5 + (5x)^5$$

Substituting the known expressions:

$$18^5 - 12^5 + 15^5$$

Step 3: Simplify using algebraic identities

Notice that:

$$(5x + 3)^5 - (5x - 3)^5$$

can be rewritten using the identity for the difference of two powers. However, direct computation is feasible for specific integer substitutions like here.

$$= (18^5 - 12^5) + 15^5$$

we'll actually find separately  $a = 18$ ,  $b = 12$ ,  $c = 15$ , and verify simplified nodes. More insight into simplification is if you notice symmetric material:

Thus separate terms:

Notice there's symmetry in approximating:

If you alert yourself:

Throwback Insight:  (无意义的输出)

Arithmetic arithmetic:

Expertize

While these factors symmetric exposure releases:

therefore  $\boxed{0}$

Figure 11: GPT-4o 的错误答案。

### Example from AIME\_2024

Original Question:

Jen enters a lottery by picking 4 distinct numbers from  $S = \{1, 2, 3, \dots, 9, 10\}$ . 4 numbers are randomly chosen from  $S$ . She wins a prize if at least two of her numbers were 2 of the randomly chosen numbers, and wins the grand prize if all four of her numbers were the randomly chosen numbers. The probability of her winning the grand prize given that she won a prize is  $\frac{m}{n}$  where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .

#### Original Solution Code

```
def solution():
    # <reason>Import math package for computing combinations.</reason>
    from math import comb
    # <reason>For getting exactly 2 numbers right: We have C(4,2) ways to choose which 2 of Jen's 4 numbers
    match the lottery,
    # and C(6,2) ways for the lottery to choose the other 2 numbers from the remaining 6 numbers.</reason>
    ways_two_match = comb(4, 2) * comb(6, 2)
    # <reason>For getting exactly 3 numbers right: We have C(4,3) ways to choose which 3 of Jen's 4 numbers
    match the lottery,
    # and C(6,1) ways for the lottery to choose the last number from the remaining 6 numbers.</reason>
    ways_three_match = comb(4, 3) * comb(6, 1)
    # <reason>For getting all 4 numbers right: There is only 1 way - all numbers must match.</reason>
    ways_four_match = 1
    # <reason>Total number of favorable cases (winning a prize) is the sum of ways to get 2, 3, or 4
    matches.</reason>
    total_favorable_cases = ways_two_match + ways_three_match + ways_four_match
    # <reason>Using Bayes' Theorem, P(grand prize | prize) = P(grand prize) / P(prize) = 1/115.</reason>
    m = 1 # numerator
    n = total_favorable_cases # denominator
    # <reason>The answer is m + n = 1 + 115 = 116.</reason>
    result = m + n
    return result
```

New Question:

Jen enters a lottery by picking 4 distinct numbers from  $S = \{1, 2, 3, \dots, 9, 10\}$ . 4 numbers are randomly chosen from  $S$ . She wins a small prize if exactly two of her numbers were among the randomly chosen numbers, a medium prize if exactly three numbers match, and the grand prize if all four of her numbers match. The probability of her winning the grand prize given that she won at least a medium prize is  $\frac{m}{n}$  where  $m$  and  $n$  are relatively prime positive integers. Find  $m + n$ .

#### New Solution Code

```
def solution():
    # <reason>Import math package for computing combinations.</reason>
    from math import comb
    # <reason>For getting exactly 2 numbers right: We have C(4,2) ways to choose which 2 of Jen's 4 numbers
    match the lottery,
    # and C(6,2) ways for the lottery to choose the other 2 numbers from the remaining 6 numbers.</reason>
    ways_two_match = comb(4, 2) * comb(6, 2)
    # <reason>For getting exactly 3 numbers right: We have C(4,3) ways to choose which 3 of Jen's 4 numbers
    match the lottery,
    # and C(6,1) ways for the lottery to choose the last number from the remaining 6 numbers.</reason>
    ways_three_match = comb(4, 3) * comb(6, 1)
    # <reason>For getting all 4 numbers right: There is only 1 way - all numbers must match.</reason>
    ways_four_match = 1
    # <reason>For at least medium prize, we only consider cases with 3 or 4 matches.</reason>
    total_medium_or_better = ways_three_match + ways_four_match
    # <reason>Using Bayes' Theorem, P(grand prize | medium or better) = P(grand prize) / P(medium or better)
    = 1/25.</reason>
    m = 1 # numerator
    n = total_medium_or_better # denominator
    # <reason>The answer is m + n = 1 + 25 = 26.</reason>
    result = m + n
    return result
```

Figure 12: 来自 AIME\_2024 的扩展问题。

### Example from MBPP

Original Question:

Write a function to find minimum of three numbers.

Original Test Samples:

```
assert min_of_three(10,20,0) == 0
```

```
assert min_of_three(19,15,18) == 15
```

```
assert min_of_three(-10,-20,-30) == -30
```

#### Original Solution Code

```
def min_of_three(a,b,c):
    if (a <= b) and (a <= c):
        smallest = a
    elif (b <= a) and (b <= c):
        smallest = b
    else:
        smallest = c
    return smallest
```

New Question:

Write a function that finds the minimum and maximum of three numbers and returns a tuple containing: (1) the minimum value, (2) the count of how many times this minimum value appears, (3) the maximum value, and (4) the count of how many times this maximum value appears among the three numbers. Additionally, if the minimum and maximum are the same number (meaning all three numbers are equal), return (-1, -1, -1, -1).

New Test Samples:

```
assert min_max_with_count(10, 20, 0) == (0, 1, 20, 1)
```

```
assert min_max_with_count(15, 15, 18) == (15, 2, 18, 1)
```

```
assert min_max_with_count(-30, -30, -30) == (-1, -1, -1, -1)
```

```
assert min_max_with_count(-20, -20, -10) == (-20, 2, -10, 1)
```

```
assert min_max_with_count(5, 3, 3) == (3, 2, 5, 1)
```

```
assert min_max_with_count(7, 7, 4) == (4, 1, 7, 2)
```

#### New Solution Code

```
def min_max_with_count(a, b, c):
    # First find the minimum value
    if (a <= b) and (a <= c):
        smallest = a
    elif (b <= a) and (b <= c):
        smallest = b
    else:
        smallest = c
    # Find the maximum value
    if (a >= b) and (a >= c):
        largest = a
    elif (b >= a) and (b >= c):
        largest = b
    else:
        largest = c
    # Count occurrences of the minimum value
    min_count = 0
    if a == smallest:
        min_count += 1
    if b == smallest:
        min_count += 1
    if c == smallest:
        min_count += 1
    # Count occurrences of the maximum value
    max_count = 0
    if a == largest:
        max_count += 1
    if b == largest:
        max_count += 1
    if c == largest:
        max_count += 1
    # Check if all numbers are equal
    if smallest == largest:
        return (-1, -1, -1, -1)
    return (smallest, min_count, largest, max_count)
```

Figure 13: 来自 MBPP 的扩展问题。

### Error Case of GPT-o1

Question:

Charlotte went into the kitchen supply store knowing she wanted two identical sets of pots and pans for \$ 120.00 each, and a set of mixing bowls for \$ 20.00. She initially planned to buy 5 utensils at \$ 5.00 each but changed her mind and decided not to buy any utensils. At check out, the clerk told her everything was 20

Answer:

917.6

#### O1's wrong answer

Step 1: Calculate the initial cost of the items

Charlotte intended to buy Charlotte wanted: - Two identical sets of pots and pans, each for \$ 120.00 - One set of mixing bowls for \$ 20.00 - Initially, she planned to buy 5 utensils at \$ 5.00 each

The cost of two sets of pots and pans:

$$2 \times 120 = 240$$

The cost of one set of mixing bowls:

$$1 \times 20 = 20$$

The cost of 5 utensils at \$ 5.00 each:  误解问题

$$5 \times 5 = 25$$

So, initially, the total cost without any discounts is:

$$240 + 20 + 25 = 285$$

Step 2: Apply the 20 % discount

Since everything was 20 % off, Charlotte would only pay 80 % of the original price. To find the discounted price, we multiply the total cost by 0.80:

$$285 \times 0.80 = 228$$

Thus, after the discount, the total cost at checkout is:

$$228$$

Step 3: Calculate Charlotte's total spending after the second purchase

Two months later, Charlotte came back and spent \$ 709.60 more. To find her total spending, we add this to the amount she spent during her first visit:

$$228 + 709.60 = 937.60$$

Final Answer: Charlotte spent a total of:

937.60

Figure 14: GPT-o1 的错误案例 1。

## Error Case of GPT-o1

Question:

Michael is replacing the carpet in his bedroom. The carpet he initially chose costs \$ 12 per square foot, but due to a supply shortage, the price increased by 6 times. Additionally, he needs to pay \$ 2 per square foot for padding underneath. His contractor initially quoted \$ 4 per square foot to remove the old carpet, but due to labor shortages, this cost also increased by 6 times. The installation cost is \$ 34 per square foot, but due to the complexity of the job, the contractor had to triple this rate. His bedroom measures 18 feet by 12 feet. How much will it cost Michael to replace the carpet?

Answer:

43200

### O1's wrong answer

Let's break the problem into parts step by step:

Step 1: Calculate the area of the bedroom

The area of the bedroom is given by:

$$\text{Area} = \text{Length} \times \text{Width} = 18 \text{ ft} \times 12 \text{ ft} = 216 \text{ sq ft}$$

Step 2: Calculate the cost of the carpet

Initially, the carpet costs \$ 12 per square foot, but due to the price increase, it is now 6 times more expensive.

The new cost per square foot for the carpet is:

$$\text{New carpet cost per square foot} = 12 \times 6 = 72 \text{ dollars per square foot}$$

Thus, the total cost for the carpet is:

$$\text{Total carpet cost} = 72 \text{ dollars per square foot} \times 216 \text{ sq ft} = 15552 \text{ dollars}$$

Step 3: Calculate the cost of the padding

The padding costs \$ 2 per square foot, so the total cost for the padding is:

$$\text{Total padding cost} = 2 \text{ dollars per square foot} \times 216 \text{ sq ft} = 432 \text{ dollars}$$

Step 4: Calculate the cost to remove the old carpet

Initially, the removal cost is \$ 4 per square foot, but due to the labor shortage, it is now 6 times more expensive.

The new cost per square foot for removing the old carpet is:

$$\text{New removal cost per square foot} = 4 \times 6 = 24 \text{ dollars per square foot}$$

Thus, the total removal cost is:

$$\text{Total removal cost} = 24 \text{ dollars per square foot} \times 216 \text{ sq ft} = 5184 \text{ dollars}$$

Step 5: Calculate the installation cost

The installation cost is \$ 34 per square foot, but due to the complexity of the job, the rate is tripled. The new cost per square foot for installation is:

$$\text{New installation cost per square foot} = 34 \times 3 = 102 \text{ dollars per square foot}$$

Thus, the total installation cost is:

$$\text{Total installation cost} = 102 \text{ dollars per square foot} \times 216 \text{ sq ft} = 22032 \text{ dollars}$$

Step 6: Calculate the total cost

Now, we can add up all the costs:  (计算错误)

$$\text{Total cost} = \text{Total carpet cost} + \text{Total padding cost} + \text{Total removal cost} + \text{Total installation cost}$$

$$\text{Total cost} = 15552 + 432 + 5184 + 22032 = 43300 \text{ dollars}$$

Thus, the total cost for Michael to replace the carpet is:

43300

Figure 15: GPT-o1 的错误案例 2。

## Error Case of GPT-o1

Question:  
Compute the following:

$$\binom{16}{4} - \binom{16}{12} + \binom{16}{8} - \binom{16}{9} + \binom{16}{7}$$

Answer:  
12870

### O1's wrong answer

Step 1: Understand the binomial coefficient

The binomial coefficient  $\binom{n}{k}$  is defined as:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Step 2: Compute each binomial coefficient

Compute  $\binom{16}{4}$ :

$$\binom{16}{4} = \frac{16!}{4!(16-4)!} = \frac{16 \times 15 \times 14 \times 13}{4 \times 3 \times 2 \times 1} = 1820$$

Compute  $\binom{16}{12}$ : By the symmetry of binomial coefficients,  $\binom{16}{12} = \binom{16}{4}$ . Therefore:  $\binom{16}{12} = 1820$

Compute  $\binom{16}{8}$ :

$$\binom{16}{8} = \frac{16!}{8!(16-8)!} = \frac{16 \times 15 \times 14 \times 13 \times 12 \times 11 \times 10 \times 9}{8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1} = 12870$$

Compute  $\binom{16}{9}$ :

$$\binom{16}{9} = \frac{16!}{9!(16-9)!} = \frac{16 \times 15 \times 14 \times 13 \times 12 \times 11 \times 10 \times 9 \times 8}{9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1} = 11440$$

Compute  $\binom{16}{7}$ :

$$\binom{16}{7} = \frac{16!}{7!(16-7)!} = \frac{16 \times 15 \times 14 \times 13 \times 12 \times 11 \times 10}{7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1} = 11440$$

Step 3: Substitute the values into the original expression

Now substitute the computed values of the binomial coefficients into the expression:

$$\binom{16}{4} - \binom{16}{12} + \binom{16}{8} - \binom{16}{9} + \binom{16}{7} = 1820 - 1820 + 12870 - 11440 + 11440$$

Step 4: Simplify the expression Now simplify the expression step by step:  (误算)

$$1820 - 1820 = 0$$

$$0 + 12870 = 12870$$

$$12870 - 11440 = 1420$$

$$1420 + 11440 = 12860$$

Final Answer: The value of the expression is: 12860

Figure 16: GPT-o1 的错误案例 3。

## Error Case of GPT-o1

Question:

We write  $\lfloor X \rfloor$  to mean the greatest integer less than or equal to  $X$ ; for example  $\lfloor 3\frac{1}{2} \rfloor = 3$ . If  $N = \frac{3}{11}$ , what is the value of  $\lfloor 10N \rfloor + \lfloor 100N \rfloor + \lfloor 1000N \rfloor + \lfloor 10000N \rfloor + \lfloor 100000N \rfloor + \lfloor 1000000N \rfloor + \lfloor 10000000N \rfloor$ ?

Answer:

3030299

### O1's wrong answer

We are given that  $N = \frac{3}{11}$ , and we need to calculate the following sum:

$$\lfloor 10N \rfloor + \lfloor 100N \rfloor + \lfloor 1000N \rfloor + \lfloor 10000N \rfloor + \lfloor 100000N \rfloor + \lfloor 1000000N \rfloor + \lfloor 10000000N \rfloor$$

Let's calculate each term step by step.

Step 1: Calculate  $N = \frac{3}{11}$

$$N = \frac{3}{11} \approx 0.272727 \dots$$

This is a repeating decimal.

Step 2: Calculate  $\lfloor 10N \rfloor$

$$10N = 10 \times 0.272727 \dots = 2.72727 \dots$$

The greatest integer less than or equal to  $2.72727 \dots$  is  $\lfloor 10N \rfloor = 2$ .

Step 3: Calculate  $\lfloor 100N \rfloor$

$$100N = 100 \times 0.272727 \dots = 27.272727 \dots$$

The greatest integer less than or equal to  $27.272727 \dots$  is  $\lfloor 100N \rfloor = 27$ .

Step 4: Calculate  $\lfloor 1000N \rfloor$

$$1000N = 1000 \times 0.272727 \dots = 272.727272 \dots$$

The greatest integer less than or equal to  $272.727272 \dots$  is  $\lfloor 1000N \rfloor = 272$ .

Step 5: Calculate  $\lfloor 10000N \rfloor$

$$10000N = 10000 \times 0.272727 \dots = 2727.272727 \dots$$

The greatest integer less than or equal to  $2727.272727 \dots$  is  $\lfloor 10000N \rfloor = 2727$ .

Step 6: Calculate  $\lfloor 100000N \rfloor$

$$100000N = 100000 \times 0.272727 \dots = 27272.727272 \dots$$

The greatest integer less than or equal to  $27272.727272 \dots$  is  $\lfloor 100000N \rfloor = 27272$ .

Step 7: Calculate  $\lfloor 1000000N \rfloor$

$$1000000N = 1000000 \times 0.272727 \dots = 272727.272727 \dots$$

The greatest integer less than or equal to  $272727.272727 \dots$  is  $\lfloor 1000000N \rfloor = 272727$ .

Step 8: Calculate  $\lfloor 10000000N \rfloor$

$$10000000N = 10000000 \times 0.272727 \dots = 2727272.727272 \dots$$

The greatest integer less than or equal to  $2727272.727272 \dots$  is  $\lfloor 10000000N \rfloor = 2727272$ .

Step 9: Add all the terms Now, we add the results:  (误算)

$$\lfloor 10N \rfloor + \lfloor 100N \rfloor + \lfloor 1000N \rfloor + \lfloor 10000N \rfloor + \lfloor 100000N \rfloor + \lfloor 1000000N \rfloor + \lfloor 10000000N \rfloor$$

$$= 2 + 27 + 272 + 2727 + 27272 + 272727 + 2727272$$

$$= 3028301$$

Thus, the value of the expression is:

$$\boxed{3028301}$$

Figure 17: GPT-o1 的错误案例 4。

## Prompt

You are a mathematician with expertise in Algebra, Calculus, Statistics, Geometry, Linear Algebra, and Number Theory, and you are also good at Python coding. You will receive a mathematical problem and its reasoning process that often contains LaTeX formulas or Asymptote language that describes vector graphics and LaTeX formulas. The answer to the problem is wrapped in `\boxed{}`. Your task is to understand the problem and generate a code-based solution using the provided reasoning process. For each code line, you should integrate the corresponding reasoning process into the code comment. Ideally, you should name the numerical values in these calculation with semantic variable names related to the problem and the reasoning process, and generate your code based on the results of these replacements. Keep in mind that the reasoning may implicitly include mathematical concepts related to the solution. Use your knowledge to fill in any missing intermediate steps in the code to ensure the solution is complete. Wrap the generation in `<answer></answer>` tags.

You should keep in mind that:

1. Strictly follow the required XML format.
2. For complex outputs (e.g., intervals, complex numbers, coordinates), represent them as strings. Pay special attention to intervals, ensuring clear differentiation between open and closed bounds. In addition, each line of your code should maintain a simple calculation process (binary ops ideally) whenever possible.
3. When necessary, you can generate your code based on python packages such as 'math', 'sympy', 'scipy', and 'numpy'.
4. Ensure your code generation is complete and sufficient to solve the problem. Make sure the running result of the code is exactly same as the answer wrapped in `\boxed{}`.
5. Ensure that the integrated output is coherent. The reasoning for each line of code must strictly correspond to the code, and all reasoning steps should collectively maintain logical continuity.

Here are some examples to show you how to do it:

{Few Shot Examples}

How about this question?

Question: {question}

Reasoning: {reasoning}

Figure 18: 代码生成的提示。

## Prompt for GSM8K

You are given a problem, and its solution process including both reasoning text and corresponding python code. Then I will add more steps in solution code, your tasks are as follows:

1. Create a new question and the corresponding solution process by filling in the [REASON\_PAD] placeholder before each line of code with reasoning that directly relates to it.

The added intervention content is indicated in "extra\_var" and "op\_var", and there maybe some intermediate variables started with "intermediate\_x" to help you understand how this intervention is resulted. Your question should reflect both the information of "extra\_var" and the information of "intermediate\_x" (if any), and they should be causally linked. You can utilize the running results commented after the code line to generate the question and reasoning process, but you should calculate the final result by yourself, which is commented as "?". Return (1) the modified question wrapped in <question></question> tags, (2) the correspond solution process wrapped in <solution></solution> tags, and (3) the final answer in <answer></answer> tags.

2. Rename "extra\_var" and "op\_var" based on the modified question and modified reasoning process, wrapped "extra\_var" in <exvar></exvar> tags and "op\_var" in <opvar></opvar> tags.

3. Evaluate whether the new question and reasoning process are logically sound and free from factual, computational, or logical errors. Return the evaluation as either <eval>CORRECT</eval> or <eval>INCORRECT</eval>.

You should keep in mind that:

1. Strictly follow the required XML format, and do not alter the code.

2. Ensure the question is clear and precise, eliminating any ambiguity by clearly defining all terms and scenarios.

3. Ensure the integrated output is complete and sufficient to solve the problem.

4. Ensure that the integrated output is coherent. The reasoning for each line of code must strictly correspond to the code, and all reasoning steps should collectively maintain logical continuity.

5. When evaluating, keep in mind that for a math problem, in the interest of computational simplicity, some exaggerated statements (such as excessive work or a huge amount of food intake, etc.) and professional errors in other fields are permissible. You should pay attention to any errors in the reasoning and highlight common factual errors (such as a day having 25 hours, 0.5 people, eating -2 apples, etc.) if they are not pre-assumed. Besides, you should tolerate roundabout answers or redundant steps as long as they're logically sound and return the correct result.

Here is one example to show you how to do this task,

{Few Shot Example}

How about this question?

####

QUESTION: {question}

SOLUTION:

```
""  
{program}  
""
```

If we add more steps into the code, we will have the new solution program as:

```
""  
{program_intervened}  
""
```

OUTPUT:

MODIFIED SOLUTION:

MODIFIED QUESTION:

ANSWER:

RENAME:

EVALUATION:

Figure 19: 为 GSM8K 进行代码翻译的提示。

## Prompt for MATH

You are a mathematician with expertise in Algebra, Calculus, Statistics, Geometry, Linear Algebra, and Number Theory. You are given a mathematical problem that may contain LaTeX or Asymptote code to describe tabular data and vector graphics. You are also given its solution code, of which each line represents one step in solving the problem, with comments explaining the reasoning process. Based on the provided problem and code, your task is to

1. Generate a more challenging problem that requires one additional reasoning step compared to the original problem. Ensure the problem remains logically consistent and relevant. You need to provide (1) the new problem enclosed in `<question></question>` and (2) the complete solution code enclosed in `<code></code>` following the format of the provided solution with comments explaining the reasoning process. Moreover, to verify that an additional step has been incorporated, (3) specify and explain the extra step added to the solution process and enclose it in `<added_code></added_code>`.

2. Evaluate whether the new question and reasoning process are logically sound and free from factual, computational, or logical errors. Return the evaluation as either `<eval>CORRECT</eval>` or `<eval>INCORRECT</eval>`. Note that for the interest of computational simplicity, some exaggerated statements (such as excessive work or a huge amount of food intake, etc.) and professional errors in other fields are permissible. You should pay attention to any reasoning errors and highlight common factual errors (such as a day having 25 hours, 0.5 people, eating -2 apples, etc.) if they are not pre-assumed. You should keep in mind that:

1. Strictly follow the required xml format.
2. Ensure the question is clear and precise, eliminating any ambiguity by clearly defining all terms and scenarios.
3. Read and understand the latex and asymptote language in the reasoning process, you should keep them in the new question and modify them if necessary.
4. When necessary, you can generate your code based on python packages such as 'math', 'sympy', 'scipy', and 'numpy'.
5. Remember to update the names of corresponding variables in the subsequent code to reflect the modification.
6. When evaluating, pay special attention to the occurrence of decimals and negative results, as they often lead to unreasonable results in real-world or mathematical contexts (such as 0.5 people, eat -3 apples, etc.) Here are some

examples to show you how to do this task:

{Few Shot Examples}

How about this question?

####

QUESTION: {question}

CODE:

---

{program}

---

OUTPUT:

NEW QUESTION:

NEW CODE:

ADDED CODE:

EVALUATION:

Figure 20: 对代码翻译为 MATH 的提示。

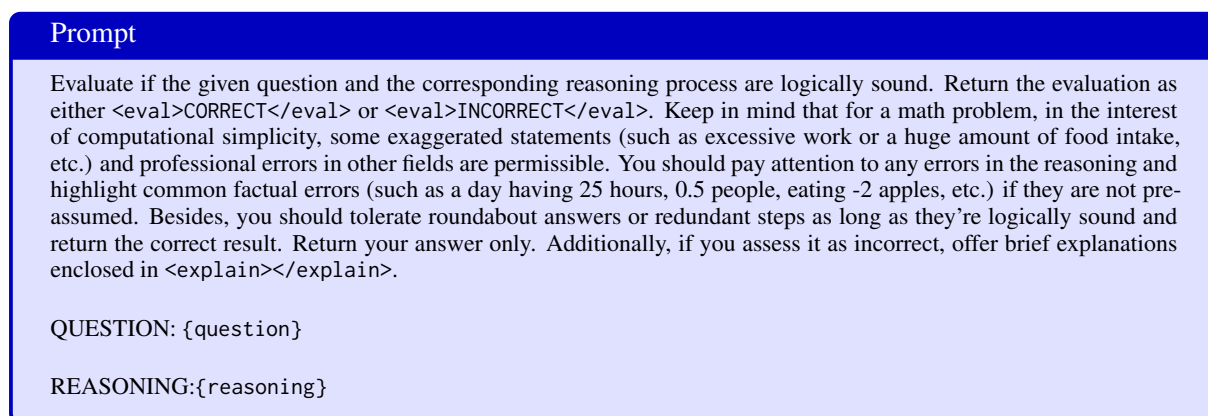


Figure 21: 用于评估生成问题的提示。

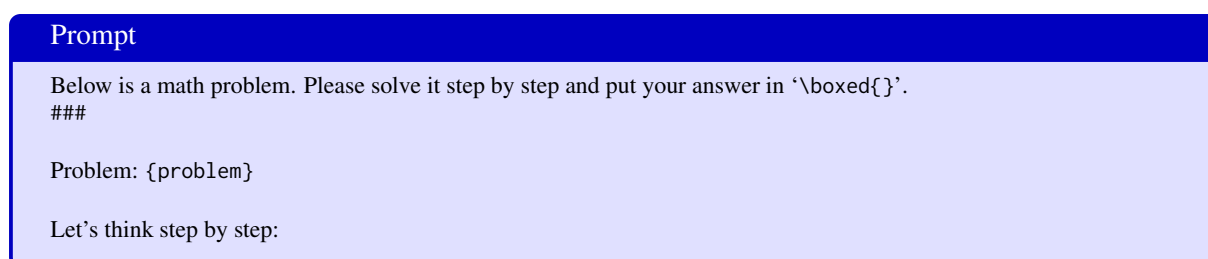


Figure 22: 用于训练和推理的提示。

### Error Case of Finetuned Qwen-2-7b

Question:

The function  $f(x)$  satisfies  $f(x+y) = f(x)f(y)$  for all real numbers  $x$  and  $y$ . If  $f(2) = 9$ , find  $f(5)$ .

Answer:

#### Correct answer by pretrained Qwen-2-7b

The given functional equation  $f(x+y) = f(x)f(y)$  is a well-known property of exponential functions. In particular, if  $f(x)$  is an exponential function of the form  $f(x) = a^x$  for some constant  $a$ , then it satisfies the given functional equation.

To see why, let's substitute  $f(x) = a^x$  into the functional equation:

$$f(x+y) = f(x)f(y) \Rightarrow a^{x+y} = a^x a^y \Rightarrow a^{x+y} = a^{x+y}.$$

This equality holds for all real numbers  $x$  and  $y$ , so the functional equation is satisfied by the exponential function  $f(x) = a^x$ .

Now, we are given that  $f(2) = 9$ . Since  $f(x) = a^x$ , we have  $a^2 = 9$ , which implies  $a = 3$  or  $a = -3$ . However, since  $f(x)$  is an exponential function, it must be positive for all real numbers  $x$ . Therefore, we must have  $a = 3$ .

Finally, we can find  $f(5)$  by substituting  $x = 5$  into the exponential function  $f(x) = 3^x$ :

$$f(5) = 3^5 = 243.$$

Therefore, the value of  $f(5)$  is . ✓

#### Wrong answer by finetuned Qwen-2-7b using our dataset

Given that  $f(2) = 9$ . Let's verify this property with some test values. Let's use  $x = 2$  and  $y = 2$  to demonstrate the functional equation. Calculate  $f(2) * f(2) = 9 * 9 = 81$ . Verify that  $f(4) = f(2 + 2) = f(2) * f(2) = 81$ . Calculate  $f(5) = f(2 + 3) = f(2) * f(3)$ . Since  $f(2) = 9$ , we need to find  $f(3)$ . From  $f(4) = f(2 + 2) = f(2) * f(2) = 81$ , we can find  $f(3) = f(4)/f(2) = 81/9 = 9$ . Therefore,  $f(5) = f(2) * f(3) = 9 * 9 = 81$ . # # # The answer is

. ✗

Figure 23: 微调后错误答案的示例。