思路痕迹：通过从大语言模型到小语言模型的推理蒸馏增强算术问题解决能力

Tyler McDonald
Department of Computer Science
Brock University
St. Catharines, Ontario
tmcdonald3@brocku.ca

Ali Emami
Department of Computer Science
Brock University
St. Catharines, Ontario
aemami@brocku.ca

Abstract

随着大型语言模型（LLMs）继续用于日常任务，提示工程在计算语言学中仍是一个活跃的贡献领域，特别是在需要专业知识的领域，如算术推理。虽然这些 LLMs 已针对多种任务进行了优化，但对于小型团队来说，其全面应用在计算或财务上可能会变得沉重。此外，完全依赖专有的、闭源的模型往往限制了定制和适应性，在研究和应用可扩展性中带来了显著挑战。相反，通过利用不超过 70 亿参数的开源模型，我们可以优化资源使用，同时仍然观察到对标准提示方法的显著改进。为了培养这一概念，我们引入了 Trace-of-Thought Prompting，这是一种简单的、零样本提示工程方法，指示 LLMs 通过关键问题解决来创建可观察的子问题，专为增强算术推理能力而设计。当应用于开源模型并与 GPT-4 结合使用时，我们观察到 Trace-of-Thought 不仅允许对问题解决过程的新见解，还在不超过 70 亿参数的语言模型中引入了高达 125 % 的性能提升。这一方法强调了开源计划在普及 AI 研究和提高高质量计算语言学应用可访问性方面的潜力。我们的代码提供了 这里 中包含的数据分析和测试脚本。

1 介绍

提示工程已成为自然语言处理 (NLP) 领域研究的关键，随着各种高性能方法迅速发表，逐渐在研究社区中受到好评，仅仅是性能在众多数据集上已成为主导标准。然而，这些方法将推理过程流行为一种“全有或全无”的过程，直到进行后验评估时才有洞察这些步骤。推理不透明意味着常见、容易解决的错误可能会在解决问题的过程中传播，降低性能，并可能鼓励贪婪解决方案，同时不给解决问题期间提供额外的机会去优化推理。(**?**) 。

此外，随着各种小型语言模型被微调用于特定任务的兴起，用户在自己的消费者硬件上部署模型来解决问题变得显然容易得多（**????** 等）。这种对一系列本地模型的控制带来了通过开源语言建模进入可访问时代的范式转变。虽然易于部署和访问，但这些语言模型在诸如文本和多模态算术推理等常见任务上可能会遇到困难，要求提高其关键问题解决能力。(**???**) 。

通过认识到混合部署的潜力，以及在多模态环境中同时使用大规模和小规模模型的优势，我们可以努力抵消每种方法的不足。虽然大规模的闭源语言模型计算密集，但我们可以将这些模型的关键推理能力提取出来，使小规模模型也具备相同的优势。通过利用这种提炼，我们可以将计算负荷移至商业可用的硬件上，而不是使用闭源 API，从而在部署模型时获得灵活性，同时通过直接访问基础设备来确保可靠的服务质量。

鉴于这些挑战和机遇，我们提出了一种新的提示策略——Trace-of-Thought，该策略将复杂问题的解决过程分解为可观察的推理步骤，直接回应了在自然语言处理领域中对更透明、高效和可访问问题解决机制的需求。图 1 展示了一个关于 GSM8K (**?**) 问题实例的比较示例，强调了简化和简洁的推理路径，使得 Trace-of-Thought 特别适合通常在复杂推理链上有困难的小型模型。

我们通过引入 Trace-of-Thought 所做的贡献有三个方面：

1. 透明性。Trace-of-Thought 提供了一个可观察的框架，其中一个问答模型要执行的步骤在执行之前由人类监督者所知。这不仅允许对当前问题进行更有见地的推理，还可以在无需等待事后分析的情况下，了解模型在执行任务时的优缺点。

2. 开源问题解决。通过使用大规模闭源模型的指令来增强开源模型，我们观察到性能的大幅提升，这展示了小型模型作为竞争性问题解决者的实用性。通过利用开源模型作为解决方案模型，我们观察到这些模型的性能比其他流行方法提高了超过 125 %。此外，像 WizardMath-7B 这样的模型超过了 GPT-3.5-Turbo Chain-of-Thought 超过 10 % ，当正确的模型被建立时，使用

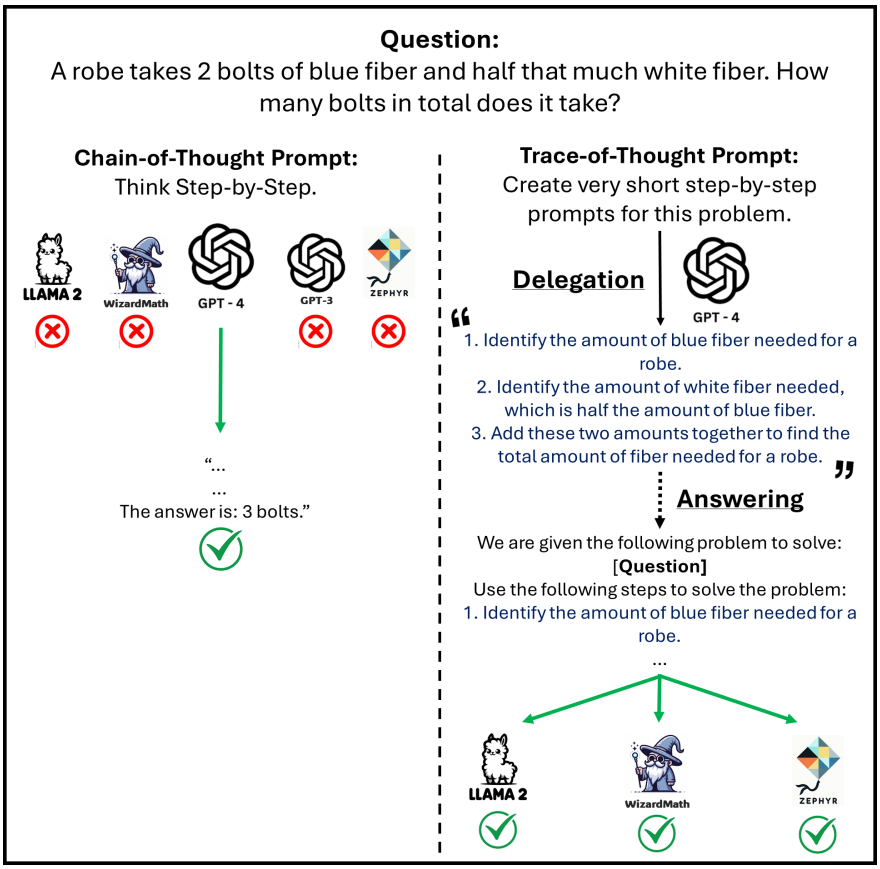


Figure 1: 标准零样本思维链提示与我们的思维痕迹提示策略在 GSM8K 问题实例上的比较。

Trace-of-Thought 是一个可行的替代选择。这种令人鼓舞的性能表明，小型开源语言模型可能会继续发展为闭源模型的合适替代方案。

3. 上下文知识蒸馏。通过使用由大型语言模型（如 GPT-4）生成的高质量指令，我们能够利用知识蒸馏的概念，并以上下文的方式应用它，而无需事先进行微调 (???) 。通过简单的委派和答复提示，我们利用 GPT-4 作为高级推理者的持续成功，并通过典型的人类交互将这种推理扩散到能力较弱的模型中 (?) 。这种方法并不是依赖大型闭源模型提供一刀切的解决方案，而是用于基于指令的推理和任务识别。

2 相关工作

开源语言建模。NLP 的民主化因开源语言模型的出现而得到了极大的推动。这些模型，例如 Zephyr-7B、Llama 2-7B Chat 和 WizardMath-7B，见证了该领域的进步，提供了特定领域的能力，挑战了大型专有模型的霸主地位 (???) 。我们的工作基于这一基础，通过实证评估展示了如何通过 Trace-of-Thought 提示进一步增强这些模型的能力，使其在计算成本的一小部分情况下接近其较大对手的效能。这不仅展示了它们作为独立解决方案的潜力，还强调了开源模型在促进一个具有竞争性和多样性的 NLP 生态系统中的重要性，特别是在利用这些开源替代方案的套件方面 (?) 。

中间推理。诸如算术推理领域中的任务需要慎重和计划好的步骤以创造出强有力的输出，而无法正确识别子问题可能会导致基本错误 (?) 。方法如零样本和小样本链式思维提示有助于这种深思熟虑的推理；然而，这些推理路径在运行时之前是隐藏的，并且不会对潜在的不正确推理提供任何见解 (?) 。同样，即使有强大的链式推理，随后的输出也可能不忠实于所使用的推理 (??) 。像最少到最多提示之类的方法确实允许对中间推理的一定程度的可见性，并鼓励忠实的表现，但可能会由于非常长的输入提示而在效率和可读性与表现之间进行权衡 (?) 。在实现思想轨迹时，我们希望严格约束问题解决步骤，以生成和指定给模型，同时在简洁提示中保持强大的表现。

问题分解。先前的工作已经通过使用递归提示或作为微调方法来学习中间步骤模式来探索问题分解 (???) 。然而，这些方法已经将问题分解作为训练数据的一个函数来处理，为了性

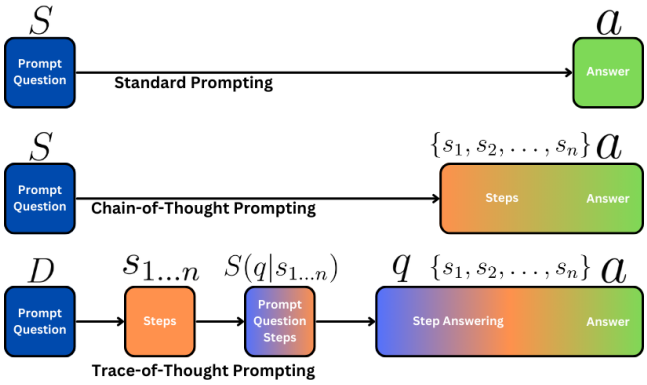


Figure 2: 与链式思维提示和标准提示的工作流程相比，思维踪迹提示的工作流程。

能提升而在问题的时间复杂度上进行权衡，或者需要额外的人类参与以取得成功 (**???**) 。与这些工作不同，思想轨迹作为一种高效的、就地的提示方法不同，因为完整的对话默认情况下不需要超过两个提示。此外，思想轨迹假设模型不需要在利用问题分解策略的训练数据上进行训练。通过从现有训练语料库中提取趋势和优势——在这里通过一系列广泛的模型演示——用户可以简单地附加提示前缀和后缀以在他们自己的工作中使用思想轨迹。

3 模型的混合性

像 GPT-4 这样的 LLMs 的使用已经在众多领域带来了变革，展示了处理复杂任务和细微推理的无与伦比的能力。尽管它们表现卓越，依赖于这样的闭源模型在可访问性、定制和长期可用性方面引入了限制。(**?**) 这些限制强调了在计算语言学中需要更民主的解决方案。

针对行业标准模型的专有特性，自然语言处理社区已见证了开源替代品开发和采用的激增。这些模型，如 WizardMath 和 DeepSeek-Math，不仅在算术推理基准上与 GPT-3.5 达成了平价，而且还正在缩小与 GPT-4 (**????**) 的差距。这种开源解决方案的趋势正在重塑格局，提供了可行的、具成本效益的替代方案，使得对高质量自然语言处理工具的访问更加民主化。

我们在此提出的模型混合概念，作为这个不断演变环境中的自然进化。此方法不仅仅是利用不同模型各自的优点以提高性能；更是在它们之间促进一种协同关系。模型混合将大语言模型（LLM）的批判性推理能力与小型开源模型的灵活性和可及性结合起来，旨在打造一个整体优于部分相加效果的组合 (**??**) 。

通过将模型视为合作伙伴，我们利用它们各自的独特优势——无论是对大型语言模型的深刻、细致理解，还是较小模型的专业知识和高效性。这种协作的集成不仅提升了性能，还确保了在快速变化的应用领域中的弹性和适应性。此外，通过打破将开发视为性能竞赛的观念，我们鼓励未来的工作专注于可访问且强大的语言建模，同时强调来自研究人员和开发者的社区合作的好处。

使用思想轨迹提示，我们寻求体现模型混合的原则，提出一种提示方法，该方法打破了传统线性和递归结构。此方法旨在最大化开源模型的效用，引导它们与从其更大对手中提炼出的推理能力协同工作。这也是朝向一个面向集成的未来迈出的一步，在这个未来中，不同模型的集体智能为计算语言学中创新问题解决策略铺平了道路。

4 思维痕迹提示

考虑一种情境，其中一个复杂问题不是由单一实体解决，而是通过协作努力解决，利用各参与者的专业能力。这就是思维痕迹提示的本质——一种将问题解决过程分成两个透明、可管理阶段的方法，利用不同语言模型的优势。

让我们定义一个语言模型 D ，它接收一个可分解的问题 q ，这个查询可以被解构为一系列更简单的中间子问题。这个模型 D 在单一上下文中生成一系列推理步骤 $s_{1...n}$ ：

$$D(q) \rightarrow s_1, s_2, \ldots, s_n \tag{1}$$

关键在于，$s_{1...n}$ 不是孤立的输出，而是按顺序生成的，提供了一条清晰的、逐步的解决路径。

接下来，一个求解器 S 利用这些中间步骤来计算最终答案 a ：

$$S(q|s_1, s_2, \ldots, s_n) \rightarrow a \tag{2}$$

重要的是，求解器 S 可以是不同于 D 的模型或实例，这为问题解决过程提供了灵活性。

思维痕迹方法包括两个不同的阶段：

1. 委托（$D(q)$ ）：这个阶段涉及生成一组简洁的、分步的解决问题的说明。在进入回

Prompt Type	Template
Standard	“<question>.”
Chain-of-Thought	“<question>. Think step-by-step. ”
Trace-of-Thought - Delegation	“ Create very short step-by-step prompts for the following problem: <question>. Format as a list. Do not solve the problem. ”
Trace-of-Thought - Solution	“ We are given the following problem: <question>. Use the following steps to solve the problem: <steps>.”

Table 1: 实验评估中使用的提示模板。

答阶段之前，用户可以手动或使用验证模型来验证和完善这些步骤。这个明确的分段确保在早期阶段即可纠正任何不准确之处 (**?**) 。

2. 回答（$S(q|s_1, s_2, \ldots, s_n)$）：在第二阶段，解算器模型使用委派的步骤来解决原始问题。模型仅专注于所提供的步骤，不会受到提示而进行额外的推理，从而严格遵循制定的计划。图 2 比较了这种简化过程与之前方法可能更复杂的路径。

。思维痕迹提示不仅增强了推理过程的可见性，还使用户能够利用一系列模型的专业能力，特别是在算术推理等领域。这促进了资源的更有效利用，并为模型根据其优势 (**?**) 作为专注的问题解决者或熟练的委派者铺平了道路。

通过遵循模型混合的原则，思维轨迹提示主张建立一个多样化、合作式的问题解决环境。它预示着向实现集成导向方法迈进的一步，其中不同模型的优势融合带来强大、适应性强且能力卓越的自然语言处理系统。

对本文的目的，我们考虑两个算术推理任务：从这些数据集中，我们选择前 200 个问题，无论是从数据集的主体部分还是从测试分割部分（如果可用的话）。我们考虑了多种封闭源代码和开源模型进行比较：

GPT-4（未公开大小，闭源）：由 OpenAI 创建的闭源模型。使用于 06/13 快照。

GPT-3.5-Turbo（未公开大小，闭源）：由 OpenAI 创建的 GPT-4 的前身系列的一部分，在 06/13 快照中使用。

WizardMath-7B（7B，开源）：由 WizardLM 生产的开源模型套件，基于 Mistral 训练，并在 GSM8K 数据集上进行微调。使用人类反馈强化学习（RLHF）的新扩展技术，强化进化指令（RLEIF）进行训练。

Llama 2-7B Chat (7B, 开源)：是 Meta 公司生产的一套开源模型的一部分，并针对对话目的进行了微调。

Zephyr-7B（7B，开源）：由 HuggingFace 的 H4 团队开发的开源模型，使用直接偏好优化 (**?**) 在合成数据上训练。

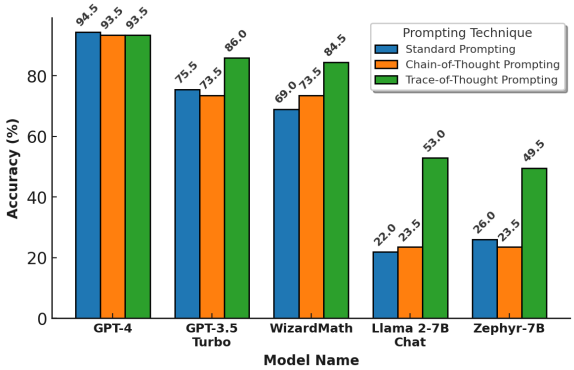


Figure 3: 在 GSM8K 上使用三种提示方法的各种模型的准确性。

我们在所有数据集上比较三种主要的提示方法，所有都是在零样本环境下进行的——这种提示环境被选择用于以最公平的形式模拟典型的人机交互。

标准提示：模型使用数据集的准确内容进行提示，未作修改。

链式思维提示：模型被给予一个问题，并被指示“逐步思考”，而没有之前的上下文示例。

思想追踪提示：选择一个委托模型和解决模型，然后我们执行思想追踪的委托 & 答题工作流。同样的，没有提供之前的上下文例子。在每个思想追踪的实例中，我们都使用 GPT-4。[1]

为了说明这些提示方法之间的操作差异，图 2 提供了一个比较工作流程图，可以参考以直观地理解每种方法的结构。有关使用的提示模板的详细比较，请参见表 1 。

4.1 评估

答案通过与数据集标签进行精确匹配来评估准确性；“接近”的答案不被考虑，除非由模型四舍五入。结果，包括问题、模型和数据集答案，以及对于思路跟踪，步骤和答案，均记录在 CSV 文件中。每个回应由内部注释员标记为真或假。

[1]我们使用 GPT-4 作为委托模型是基于其已验证的推理能力和在我们研究中的便利性，并不是因为模型需要大规模或闭源。我们鼓励未来的工作研究开源的 Trace-of-Thought 替代方案，以提高语言建模的成本效益。

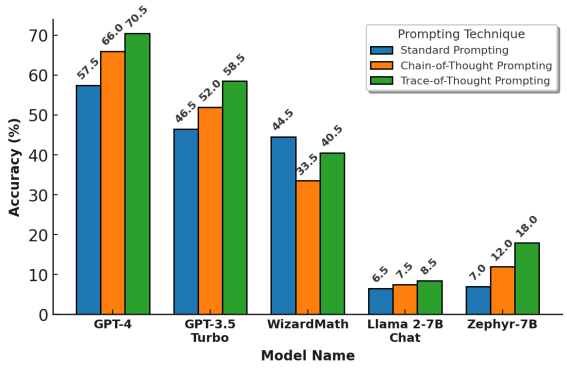


Figure 4: 三种提示方法下各模型在 MATH 上的准确性。

如果模型得到正确答案，但添加了额外信息，如不必要的附加结果或后续问题，我们将其标记为正确答案，但记录该情况以便于进一步分析。此外，在一些模棱两可的情况下，如果模型正确推理出答案，但在答案格式化过程中可能返回了不同答案，我们会将该问题标记为正确，并注明需要进一步分析。例如，如果一个模型在推理过程中得出结果为 5，但错误地说“答案是：6”，我们会认可推理得到的 5，而不是格式化后的答案 6。

4.2 GSM8K

图 3 展示了每个模型在 GSM8K ($n = 200$) 上，使用所有三种方法的表现。当模型大小减小时，我们观察到通过使用思想轨迹提示能获得更大且在统计上显著的相对性能提升。虽然 GPT-4 整体上仅有轻微的损失，但 GPT-3.5-Turbo 的准确率提高了 17 %，Llama 2-7B Chat 的提升超过了 125 %，而其他模型的准确率提升范围在 14 % 到 50 % 之间。类似地，在表格 2 和 3 中，我们看到更紧密的准确率范围，更高的平均值并且准确率提升非常一致，这表明所有模型的准确性总体上呈现出令人鼓舞的改善趋势。当结合 WizardMath 和思想轨迹使用时，我们观察到的提升接近 GPT-4，这是利用 GPT-4 的推理能力和 WizardMath 的算术微调的有力例证。GSM8K 的表格结果和统计显著性测试分别见附录表格 **??** 和 5 。

4.3 数学

图 4 展示了每个模型在 MATH ($n = 200$) 上使用三种方法的表现。虽然对于像 GPT-4 和 GPT-3.5 这样的闭源模型的提升缺乏统计显著性，但它们的改进，特别是 GPT-4 的提升，因为它可以由自身提示受益，标志着使用 Trace-of-Thought 作为通用解决方案带来的准确性优势的强劲趋势，即使在大型闭源模型上亦是如此。此外，虽然 Trace-of-Thought 未能在 WizardMath-7B 上与其他方法取得等效，但我们在历史上表现不佳的模型 Llama 2 Chat 和 Zephyr 上继续看到收益。MATH 数据集的使用维持了类似的趋势，即精确度范围更紧密以及性能提升，如表 2 和 4 所示。Trace-of-Thought 的这一初始迭代能够应用的程度仍有待观察，但这些结果坚定地支持至少在部分上提升算术问题解决能力在困难问题数据集上的概念。MATH 的表格结果和统计显著性测试可以分别在附录表 **??** 和 6 中找到。

在使用小规模语言模型的思维轨迹时，我们观察到了两个与模型增强提供步骤的自身推理相关的有趣趋势。首先，如图 **??** 所示，开源语言模型识别并随后忽略了冗余步骤，保留了透明性，同时保留了推理并减少了幻觉的可能性。通过识别不必要的过程或净收益为零的过程，模型表现出比单纯的指令跟随更深入的问题解决，而是使用步骤作为监督框架的问题解决。此外，当模型遇到一个可以根据所需信息进一步分割的步骤时——无论是来自先前的步骤还是从观察问题上下文中获得的——它可能会选择通过进一步的推理链来增强提供的步骤，类似于递归推理链，如图 **??** 中所示。通过进一步引入局部推理，模型填补了空白，或添加了其认为必要的问题上下文。虽然先前的工作已经研究了类似的步骤评估作为事后方法，但这种现象利用了答案生成发生的相同上下文；通过分析和修正现有步骤，无需进行额外的事后验证，而模型可以在必要时进行修正。

在将 GPT-4 用作 MATH 数据集上的委托模型和求解模型时，我们观察到准确性的显著提高，尽管在该系列测试中没有使用其他模型（图 4 ）。这种反身使用 GPT-4 虽然在统计上没有显著性，但说明了无论任务中使用哪种模型，深思熟虑推理的好处。

这种我们称之为反身思维轨迹的方法，更少强调模型集成的能力，而是利用了以前在思维树等方法中应用的深思熟虑问题解决的方法。通过使用反馈思维轨迹取得的渐进成功提出了进一步考虑使用更大、开源模型进行委托和问题解答任务的可能性。这种从闭源建模解决方案的转移在计算语言学中是一个关键范式，使研究公司能够开发不仅用于解决问题，还用于深思熟虑推理和指令生成的模型。

虽然现有的方法如“思维链”提示鼓励逐步推理，但很难将这些步骤正确地分割为整个解决方案的独立部分。我们发现“思维轨迹”提示鼓励模型遵循定义的步骤结构，包括使其解决方案与步骤编号对齐以便于更好地观察，即使步骤是冗余的或缺失的。图 5 （附录）等案例展示了在利用 GPT-3.5-Turbo 时两种方法在

Model	Highest Alternative	Trace-of-Thought	% Change
GPT-4	94.5	93.5	-1.06 %
GPT-3.5-Turbo	75.5	86	+ 13.91 %
WizardMath-7B	73.5	84.5	+ 14.97 %
Llama 2-7B Chat	23.5	53	+ 125.53 %
Zephyr-7B	12	18	+ 50 %

Model	Highest Alternative	Trace-of-Thought	% Change
GPT-4	66	70.5	+ 6.82 %
GPT-3.5-Turbo	75.5	86	+ 12.5 %
WizardMath-7B	44.5	40.5	-8.99 %
Llama 2-7B Chat	7.5	8.5	+ 13.33 %
Zephyr-7B	26	49.5	+ 90.38 %

Table 2: 在 GSM8K 和 MATH（$n = 200$）上测试的模型之间的准确性增益比较。

Approach	Min	Max	Mean
Standard Prompting	22	94.5	57.4
Chain-of-Thought	23.5	93.5	57.5
Trace-of-Thought	49.5	93.5	73.3

Table 3: 使用 GSM8K ($n = 200$) 对每种提示方法的准确性进行集中趋势测量。

Approach	Min	Max	Mean
Standard Prompting	6.5	57.5	32.4
Chain-of-Thought	7.5	66	34.2
Trace-of-Thought	8.5	70.5	39.2

Table 4: 每种提示方法的准确性集中趋势的测量，使用 MATH ($n = 200$)。

透明度上的差异。重要的是，“思维轨迹”提示鼓励遵循所提供的步骤指南进行推理。虽然“思维链”在呈现答案时开发了这个推理链，但“思维轨迹”对解决方案模型设定了明确的期望，以遵守结构和预期的步骤，即使这些步骤可以被修改。这种清晰的推理窗口不仅允许判断解决方案模型在某些目标任务上的效力，还可以对委托模型生成某些步骤的必要性提供一些见解。结合解决方案模型有时会完全省略这些冗余步骤的能力，这是一个鼓励细致且精细调整委托过程的强烈趋势。我们介绍了思维轨迹提示，这是一种新颖的提示方法，旨在将问题分解为委托和解答步骤，我们分析了其在使用 GPT-4 进行委托的小规模语言模型上应用时的表现。思维轨迹提示旨在推动向易于访问和开源的语言建模的范式转变，并建议像 GPT-4 这样的大型语言模型未来可能作为委托者而不是问题解决者。我们的研究结果敦促社区考虑可访问、包容和民主化研究的优势，并在开发未来的方法时优先考虑开源模型。

5

限制

提示敏感性。忽略委托或答案提示中的某些部分可能会导致性能下降，这尚未被充分探索。我们鼓励对此类提示结构进行彻底的显著性研究，以防止这种性能下降。

抽象推理。诸如 ARC 和 ACRE 之类的数据集探索抽象推理任务，其目标是在极少的数据 (**??**) 上推广和应用一个共同的模式。由于它们的抽象特性，以及原始问题缺乏可观察的划分，目前尚不清楚 Trace-of-Thought 是否是执行诸如一般模式识别或常识推理等任务的有效和有用的框架。

财务影响。由于解决方案模型生成步骤所需的额外提示，以及为达到此效果所需的较长输入提示，在主要闭源的环境中应用 Trace-of-Thought 可能会导致额外的成本和增加受限资源（如 API 额度以应对速率限制）的使用。

从委托模型中解的扩散。类似于提示的敏感性，用户应该注意，如果过于强烈地提示其委托模型以鼓励解的产生，可能会导致步骤的生成是基于先前的工作，而不是解模型的方向，从而导致后续结果被污染。

模型依赖性和可扩展性。Trace-of-Thought 的有效性在很大程度上依赖于所选模型的能力，这可能限制其在模型缺乏特定知识或推理技能的领域的应用。此外，该方法的计算和资源需求可能在资源有限的环境中带来可扩展性挑战。

Two Sample Z-Test for Proportions - significant results are bolded, and th

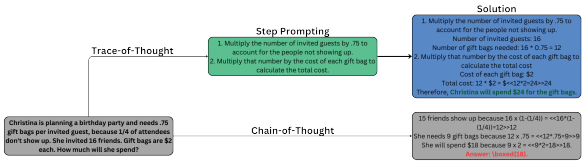
Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative
GPT-4	93.5	
GPT-3.5-Turbo	86	
WizardMath-7B	84.5	
Llama 2-7B Chat	52.5	
Zephyr-7B	56	

Table 5: 使用两样本比例 Z 检验比较 Trace-of-Thought 在 GSM8K 数据集上的表现与各模型中表现最好的替代方法进行统计比较。

Two Sample Z-Test for Proportions - significant results are bolded, and th

Model	Trace-of-Thought($\bar{x}_{ToT}$)	Highest Performing Alternative
GPT-4	70.5	
GPT-3.5-Turbo	58.5	
WizardMath-7B	40.5	
Llama 2-7B Chat	8.5	
Zephyr-7B	18	

Table 6: 使用两样本比例 Z 检验，在 MATH 数据集上比较 Trace-of-Thought 表现与模型中表现最佳的替代方法之间的统计差异。



PT-3.5-Turbo 使用链式思维与思维痕迹来解决 GSM8K 问题——思维痕迹提供了可观察的推理