超越最后的答案：你的推理过程揭示的比你想象的更多

Hasan Abed Al Kader Hammoud* KAUST
Hani Itani KAUST
Bernard Ghanem KAUST

Abstract

大型语言模型（LLMs）通过逐步推理解决复杂问题。标准评估实践包括生成完整的推理轨迹并评估最终答案的正确性。在本文中，我们通过提出以下两个问题来挑战对最终答案的依赖：最终答案是否可靠地代表模型的最优结论？替代推理路径能否产生不同的结果？为了回答这些问题，我们分析称为子思考的中间推理步骤，并根据我们的发现提出一种方法。我们的方法包括根据语言提示将推理轨迹分段为连续的子思考。我们首先提示模型生成从每个中间子思考终点开始的延续。我们从源自不同子思考的每个完成的延续中提取一个潜在答案。我们发现，通过选择最频繁的答案（众数）来汇总这些答案，通常比仅依赖于原始完整轨迹得出的答案能显著提高准确性。分析从不同子思考得出的答案的一致性揭示了与模型的自信程度和正确性相关的特征，这表明有可能识别出不太可靠的答案。我们在各种大型语言模型和具有挑战性的数学推理数据集（AIME2024 和 AIME2025）上的实验显示了一致的准确性改进，增益分别达到 13 % 和 10 %。实施可在以下网址获得：https://github.com/hammoudhasan/SubthoughtReasoner 。

1 介绍

大型语言模型（LLMs）在被引导逐步阐述其推理过程中，表现出了解决复杂任务的非凡能力 [28] 。推理不仅需要通过扩大预训练规模来获得足够的知识基础，还需要在推理期间（测试时计算）增加计算资源。这使模型能够参与类似于人类“系统 2 思维”的深思熟虑的多步骤推理过程 [14] ，超越即时直观的“系统 1”反应 [14] 。像 OpenAI 的 o1 [12] 和 DeepSeek-R1 [8] 这样的模型证明了在测试时计算中扩展的重要性，即投入大量推理资源以生成详细的推理线索，然后再产生最终输出。对这类模型的标准评估协议通常仅关注最终输出，即模型生成一个推理线索并以最终答案结束，仅对此单一最终答案进行正确性评估。

然而，仅仅依赖最终答案可能会忽略隐藏在推理过程中的重要信息。这样隐含地假设单一生成路径代表模型最后确定的推理结果，同时忽略了思维过程中的微小变化可能会导致不同的、更准确的结论。这引发了一个基本问题：我们是否可以通过分析在整个推理过程中答案的演变和一致性，来确立一个更可靠的评估语言大模型推理能力的方法？

在本文中，我们提出了一种方法，通过测试 LLM 推理的内部一致性来研究这个问题。我们的核心思想是中断推理过程的中间点或" 子思考"，并检查从这些状态中得到的结论，如图 1 所示。具体来说，我们的方法包括：

1. 使用标准贪婪解码为给定问题生成一个初始的、完整的推理轨迹。
2. 将这一思路分割为一系列子思维，基于自然语言标记来进行分隔，这些标记通常指示推理中的转变或进展（例如，“等等”，“或者”，“嗯”）。
3. 提示同一模型从中间状态开始生成完整解决方案（即每个累积的子思维序列之后）。
4. 从这些生成的推测中提取最终的数字答案，产生一组潜在答案，以反映在初始推理结构内从不同点得出的结论。

*Correspondence at hasanabedalkader.hammoud@kaust.edu.sa

Preprint.

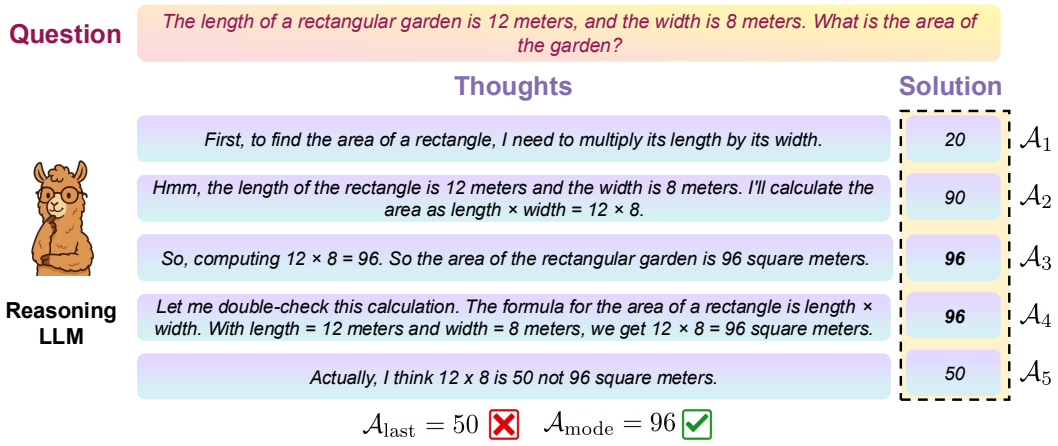


Figure 1: 子思考分析。我们展示了通过检查中间推理步骤及其对应的答案（$A_1, \ldots, A_5$），对这些答案（A_{mode}）取众数通常比仅使用最终答案（A_{last}）能够获得更好的性能，这也是通常的方法。此图说明了一个例子，其中 $A_{mode} = 96$ 是正确的，而 $A_{last} = 50$ 是不正确的。

这一过程为原始问题产生了一个答案分布。我们分析这个分布有两个主要目的：首先，我们研究模型的答案在不同子思维阶段的演变情况。我们考察原始路径中的最终答案是否能一致地从较早的点到达。我们还探讨模型在最终正确回答与错误回答问题之间，答案分布的差异。我们假设，不同子思维序列中的答案不一致或高变异性可能表明困难或潜在错误，这可作为低置信度或幻觉的信号。

其次，根据这一分析的见解，我们探讨将收集到的答案进行聚合是否能得到更稳健的最终结果。具体来说，我们假设在所有生成的完成中，出现频率最高的答案（众数）代表了一个更可信的结论，反映了在略微扰动的推理轨迹中的收敛性。

我们使用七个开源权重的 LLM 在具有挑战性的数学推理数据集（AIME2024, AIME2025）上的实验验证了这些假设。我们观察到，正确解答和错误解答的问题在一致性模式上确实存在差异。此外，通过众数汇总答案相比仅使用初始思路中的最终答案显著提高了准确性，这证明了我们分析的实际效益。

我们的贡献是：

- 一种系统分析 LLM 推理的方法，通过生成和评估从中间子思维中得出的结论。
- 一项分析显示在推理过程中答案一致性如何演变，揭示正确解和错误解的不同模式，并根据答案分布特征（例如，熵）提出检测错误的潜力。
- 实证证据表明，通过汇集子思考完成的答案，特别是通过取众数，可以显著提高准确性，相较于标准的最终答案方法，在 AIME2024 上提高达 13 %，AIME2025 上提高达 10 %。
- 比较贪婪与非贪婪采样策略用于生成子思想补全，强调它们各自的优点。

我们相信这些发现为理解 LLM 推理的本质提供了有价值的见解，并表明超越单一最终答案的评估可以解锁对模型能力的更细致理解，并可能导致性能评估的改进以及新的推理策略。我们接下来将详细介绍我们的方法论，然后是实验设置和结果，最后讨论其影响。

2 相关工作

测试时间缩放和推理。 思维链（CoT）[28] 提示是在测试时间或推理时间计算的扩展中的一个关键工作。它明确要求大型语言模型（LLM）在得出最终答案之前生成一个结构化的推理链。自洽 CoT [27] 是一种 CoT 变体技术，它用基于采样的解码替代贪心解码，以便采样多个推理链并通过一致性聚合选择最佳答案。其他提示技术专注于构建引发推理的结构化提示 [19, 21, 23] 。搜索和计划提示技术将推理任务分解为一组子任务 [6, 15, 25, 31] 。这些方法可以归类为评估最终结果或推理过程的方法 [17] 。基于提示的测试时间扩展技术引导模型选择最佳推理链，而无需更新其参数。我们的方法利用从普通 CoT 提示生成的完整推理链。然后通过增加的子思维序列提示同一模型，以便在不同推理链长度时引出答案。然后

类似于自洽 CoT，将答案的分布与众数进行聚合。值得注意的是，我们的方法可以与任何生成显式子思维序列的测试时间扩展方法一起使用。

基于训练的推理。 基于训练的技术通过训练模型来增强其推理能力。这些方法的关键挑战在于缺乏人工注释的逐步推理链。该方向的研究集中在开发自动生成有效推理轨迹的技术或提出有效利用可用数据的训练技术。训练推理模型最简单的方法是通过监督微调 (SFT) 在推理轨迹 [10, 18] 上对模型进行微调。其他研究表明，偏好学习进一步提升了推理能力。[11, 13, 18] 皆探讨了 DPO [22] 。[16, 30] 研究了逐步 DPO 而不是结果级别。最近的大多数方法通过利用强化学习 (RL) 绕过了对注释推理链的需求。在这个方向上的一个特别成功是在 GRPO [24] 中，该研究表明，即使没有初始的监督微调步骤，RL 也足以产生复杂的推理能力。目前讨论的方法都使用显性自然语言推理轨迹。一系列近期工作探讨了使用隐性推理来隐式表示推理链。这些方法侧重于将自然语言链压缩成更少数量的标记 [4, 5] 。其他工作引入了可学习的标记，这些标记被认为使模型能够在输出答案标记之前执行额外的非语言步骤。[7, 26] 。更有效地，[9] 建议使用最后一层隐藏特征作为隐性推理标记，然后将其反馈给模型以自回归地生成下一个标记。我们的方法是一种测试时方法，不更新模型参数。它适用于任何在最终答案之前输出显性自然语言思维过程的推理模型。

推理模型中的过度思考现象。 在推理模型中，过度思考现象发生在模型为相对简单的问题 [3] 生成过于详细和冗余的推理步骤时。这个现象削弱了推理模型的推断效率，并且在某些情况下导致错误的答案。最近的一些研究通过引入基于长度的奖励来控制 CoT 推理的长度，明确解决了计算效率和推理质量问题 [2, 29] 。s1 方法 [20] 引入了“预算强制”，通过目标提示修改有效地控制计算。同样地，L1 [1] 引入了长度控制策略优化（LCPO），精确管理推理复杂性。与预算推理技术相反，我们的方法在高计算范围内运行。

我们的方法受到这样一种观察的启发：过度思考可能导致错误答案。它分析了随着模型持续思考而进行的思维过程动态。它提取出一个自洽的答案，并通过测量模型答案的熵来提供关于正确性的见解。

3 方法学

我们引入了一个分析 LLM 推理的框架，通过检查初始推理过程中从中间步骤（“次想法”）推导出的结论。该过程包括：1）生成初始推理轨迹，2）根据语言线索对其进行分段，3）从这些中间点提示完成，4）提取所得答案，以及 5）分析这些答案的分布。

3.1 问题设置和初始轨迹生成

令 P 表示一个需要复杂推理的问题陈述（例如，数学证明或计算）。我们采用一种推理语言模型，记为 $\mathcal{M}$ ，来解决 P 。该过程首先是制定初始提示 $\Pi(P)$ ，旨在指导 $\mathcal{M}$ 提供在特定定界符内的逐步推理（例如，`<thought>...</thought>` ），随后以指定格式给出最终答案（例如，`\boxed{Answer}` ）。

使用这个提示，我们通过贪婪解码生成一个初始完整响应 R_{full} ，以获得模型最可能的推理路径：

$$R_{full} = \mathcal{M}(\Pi(P), \text{Params}_{greedy})$$

从这个完整响应 R_{full} 中，我们提取两个关键组件：

- 主要的推理痕迹 T ，通常被识别为最终 `<thought>...</thought>` 块中的内容。
- 最终答案 A_{last} ，提取自 R_{full} 的结尾部分，通常遵循 `\boxed{...}` 的格式。此提取是由一个专用的提取程序或模型完成，记为 $\mathcal{M}_{extract}$ 。

这个答案 A_{last} 作为比较的基准，它是标准的方法，将初始轨迹结束时产生的单一答案作为基线。

在我们方法的核心是将初始推理轨迹 T 分割成一系列有意义的中间步骤或子思想，记为 $(s_1, s_2, \ldots, s_n)$ 。这种分割旨在捕捉模型可能暂停、反思、改变方向或移动到其推理中的一个明确的下一步的点。

我们基于来自一个预定义集合 W 的词或短语的出现进行分割，我们将其称为子思考过渡标记。这些标记通常表示反思、修正、排序或选择的探索。我们实验中使用的集合 W 是：

Subthought Transition Markers (W)

```
"Wait", "Alternatively", "Another angle", "Another approach", "But wait",
"Hold on", "Hmm", "Maybe", "Looking back", "Okay", "Let me", "First", "Then",
"Alright", "Compute", "Correct", "Good", "Got it", "I don't see any
errors", "I think", "Let me double-check", "Let's see", "Now", "Remember",
"Seems solid", "Similarly", "So", "Starting", "That's correct", "That
seems right", "Therefore", "Thus"
```

我们利用从 W 导出的正则表达式来分割轨迹 T 。该模式确保来自 W 的转换标志通常表示新的子思考块 s_j 的开始（对于 $j > 1$ 而言），并且标志本身被包含在 s_j 的开头。如果在 T 内没有找到来自 W 的标志，则整个轨迹被视为单一子思考（$n = 1$ ）。令 $\oplus$ 表示字符串连接，原始轨迹可以重构为 $T = s_1 \oplus s_2 \oplus \cdots \oplus s_n$ 。

对于每个识别出的子思路边界 $i \in \{1, 2, \ldots, n\}$ ，我们构建一个累积的部分思维轨迹 T_i ，代表到子思路 s_i 结束的推理过程：

然后，我们基于原始提示 $\Pi(P)$ 创建一个修改后的提示 P_i 。该提示 P_i 包含原始问题描述，但将完整的推理轨迹 T 替换为部分轨迹 T_i 。P_i 被格式化为在适当的推理分隔符（例如，`<thought>...</thought>` ）内出现，并以一种促使模型 $\mathcal{M}$ 从该特定状态继续推理过程的方式结束。令 $\text{Format}(\Pi(P), T_i)$ 表示此格式化功能：

$$P_i = \text{Format}(\Pi(P), T_i)$$

每个部分提示 P_i 然后被反馈到相同的推理模型 $\mathcal{M}$ 中以生成一个补全 C_i 。连接 $R_i = P_i \oplus C_i$ 形成一个完整的响应，从中间状态 T_i 开始。这一过程对每个 i 从 1 到 n 重复，生成 n 个完整的响应变体。

我们尝试了两种不同的采样策略来生成这些补全 C_i ：

- 贪婪子思路完成（Params_{greedy} ）：使用温度 =0.0 和 top-p=1.0。这一策略迫使模型沿着由 T_i 定义的状态的确定性、最高概率的推理路径继续。
- 非贪婪次思想完成 ($\text{Params}_{diverse}$): 使用的温度为 1.0，顶端概率为 0.95。这鼓励随机性，并允许模型探索从 T_i 延伸出来的替代的、潜在不太可能但仍可行的推理路径。

重要的是要注意，用于分割的初始轨迹 T 始终使用 Params_{greedy} 生成。贪心和非贪心策略的选择仅适用于从部分提示 P_i 生成 n 补全 C_i 的过程中。

对于每个生成的响应变化 $R_i = P_i \oplus C_i$ ，从分思维 $s_1, \ldots, s_n$ 后开始进行补全，我们提取最终的数值答案 A_i 。这一提取使用的是与获取 A_{last} 相同的程序或模型 $\mathcal{M}_{extract}$ ：

$$A_i = \mathcal{M}_{extract}(R_i)$$

。$\mathcal{M}_{extract}$ 被设计用来强健地识别并隔离最终的数值答案，解析特定格式如 `\boxed{...}` ，或者如果预期格式不存在，则识别最显著的数值结果。这一过程为原问题 P 生成了一组 n 个潜在答案：

$$\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$$

。这个集合 $\mathcal{A}$ 捕捉了模型在被迫从不同的中间阶段完成推理过程时所达成的结论。

3.2 分析与聚合指标

答案集合 $\mathcal{A}$ 构成了我们分析的基础。如结果部分所详述，我们首先分析该集合的性质，比如答案 $(A_1, \ldots, A_n)$ 的演变及其分布（例如，使用一致性度量或熵），以理解稳定性如何与正确性相关。

基于这一分析，我们评估汇聚这些答案的有效性。令 A_{true} 为问题 P 的真实答案。我们定义一个正确性的指示函数：如果答案 A 与 A_{true} 匹配则为 $\text{IsCorrect}(A, A_{true}) = 1$ ，否则为 0 。我们使用两个主要指标比较表现，并以问题的数据集计算平均值：

我们的核心假设，通过实验探讨的是，分析集合 $\mathcal{A}$ 可以提供有价值的见解。具体来说，通过模式聚合响应，在贪婪和非贪婪完成策略两者中，相较于基线有显著改善 $\text{Acc}_{MostFreq} \geq \text{Acc}_{Last}$ 。

4 实验

4.1 实验设置。

我们考虑两个数据集 AIME2024 和 AIME2025，它们是基于美国邀请数学考试的。已知这两个数据集都需要具备推理能力才能成功解决。

为了验证我们的假设，我们考虑了七个开源模型：DeepScaleR-1.5B-Preview、DeepSeek-R1-Distill-Qwen-1.5B、DeepSeek-R1-Distill-Qwen-14B、EXAONE-Deep-7.8B、Light-R1-7B-DS、QwQ-32B，以及 Skywork-OR1-Math-7B。对于答案提取（$\mathcal{M}_{extract}$），我们始终使用了 `Qwen/Qwen2.5-14B-Instruct`，明确地提示以提取最终的数值答案，格式为 `\boxed{...}`，或者识别出最可能的最终数值结果。

为了实现高效推理，我们基于 VLLM 库构建了我们的流程。在每次 LLM 调用时，我们将新生成的标记的最大数量限制为 8192。

4.2 跨子思想的回答演变分析

我们首先研究当 i 从 1 增加到 n 时，如何从完成 i -th 子思维 (T_i) 的推理中得出最终答案 A_i 的演变。图 2 展示了在使用贪心子思维完成策略对 AIME2024 中选定问题的不同模型时，此演变过程。每个图展示了答案序列 $A_1, \ldots, A_n$（y 轴）相对于子思维索引 i（x 轴，标记为“候选索引”）。图中还标记了真实答案 (A_{true})、原始完整追踪中的最终答案 (A_{last}) 和序列中出现最频繁的答案 (A_{mode})。

我们观察到不同的模式：

值得注意的是，在我们的实验中，我们没有观察到 A_{last} 正确但 A_{mode} 错误的情况。

4.3 答案分布熵和正确性

图 2 中的视觉模式表明，$\mathcal{A} = \{A_1, \ldots, A_n\}$ 中答案的分布携带了关于模型推理过程的信息。为了量化该分布中的多样性或不确定性，我们计算每个问题的香农熵：

$$H(\mathcal{A}) = -\sum_{a \in \text{Unique}(\mathcal{A})} p(a) \log_2 p(a)$$

其中 $p(a) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{I}(A_j = a)$ 是答案 a 在序列 $\mathcal{A}$ 中的频率。较高的熵表示不同答案的分布更广（一致性较低），而较低的熵表示趋向于一个或少数几个答案的汇聚（一致性较高）。

图 3 比较了最终回答正确（使用 A_{last} 作为最终答案）的问题与回答错误的问题在 AIME2024 上使用贪心完成的三个不同模型中的答案分布的平均熵。

在所有模型中，我们观察到一个明显的趋势：正确回答问题的平均熵显著低于错误回答问题的平均熵。这定量地证实了一个视觉观察，即成功的推理路径往往在子思维中表现出较高的内部一致性，而不成功的路径通常以高变异性和探索各种（错误）结论为特征。未来，这可以作为一个指标提供给用户，以指示模型答案正确的可能性。

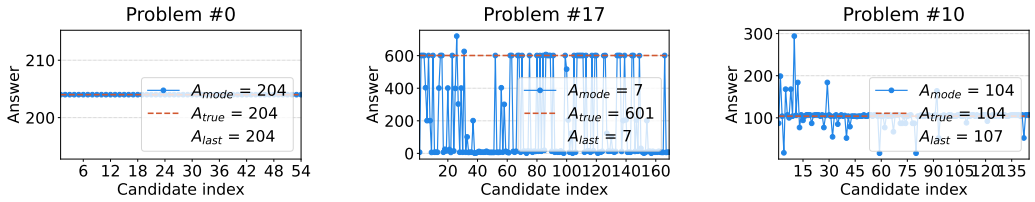
4.4 子思维聚合提升准确性

基于定性见解，我们现在定量评估我们的中心假设：使用最频繁的答案（A_{mode}）是否比使用最后一个答案（A_{last}）能带来更高的准确性？我们在所有测试的模型和两个数据集（AIME2024, AIME2025）之间比较 $\text{Acc}_{MostFreq}$ 与基准 Acc_{Last}。同时，我们通过报告 Greedy 和 Non-Greedy 完成的结果来研究分思想完成策略的影响。

图 4 显示了主要的准确性结果。该图比较了基线最后答案准确性（Acc_{Last}，蓝色条）与最频繁答案准确性（$\text{Acc}_{MostFreq}$，橙色条）在四种条件下的表现：AIME2024 使用 Greedy 完成（左上），AIME2024 使用 Non-Greedy 完成（右上），AIME2025 使用 Greedy 完成（左下），以及 AIME2025 使用 Non-Greedy 完成（右下）。条形图上方的数字表示绝对准确度百分比点差（$\text{Acc}_{MostFreq} - \text{Acc}_{Last}$）。

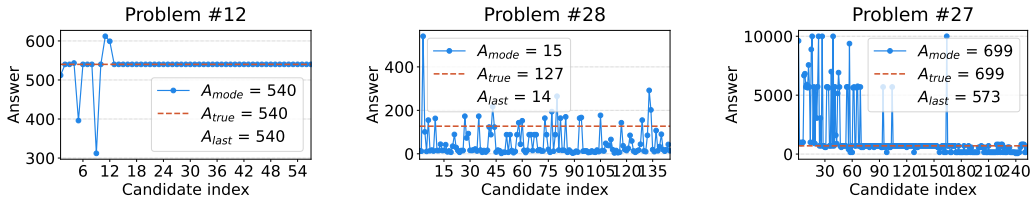
结果强烈支持我们的假设。使用众数（$\text{Acc}_{MostFreq}$）汇总答案在几乎所有模型、数据集和完成策略中均始终优于或与基准准确率（Acc_{Last}）相匹配。改进可能是显著的：

DeepSeek-R1-Distill-Qwen-14B



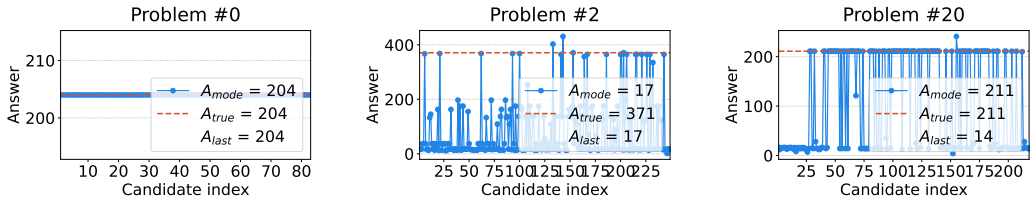
(a) 一致正确　(b) 波动不正确　(c) 模式最后更正

Light-R1-7B-DS



(d) 一致正确　(e) 波动不正确　(f) 模式最后纠正

QwQ-32B



(g) 一致正确　(h) 波动不正确　(i) 模态最后修正

Figure 2: AIME2024 上的答案演变（贪婪完成）。每一行对应一个不同的模型。

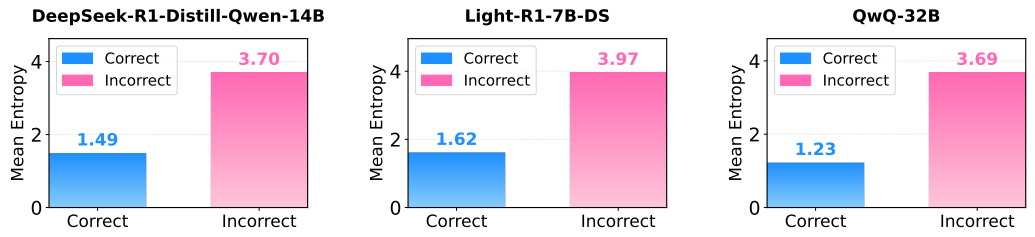


Figure 3: 在 AIME2024（贪婪完成）中，次思维答案分布的平均熵。对比正确回答的问题（$A_{last} = A_{true}$）和错误回答的问题（$A_{last} \neq A_{true}$）。较低的熵与正确答案相关。

- 在 AIME2024 上，收益最高可达 +13.33 %（Light-R1-7B-DS, Non-Greedy），并且经常超过 +6 %，多个模型实现了 +10 % 的收益（例如，DeepSeek-R1-Distill-Qwen-14B, Skywork-OR1-Math-7B, QwQ-32B 在 Non-Greedy 下）。
- 在 AIME2025 上，收益最高可达 +10.0 %（DeepSeek-R1-Distill-Qwen-14B Greedy，Skywork-OR1-Math-7B Non-Greedy），并且多个模型的收益超过 +6 %。
- 即使增益为 0 %，我们的方法通常也不会显著影响性能。观察到的少数细微下降（如 AIME2024 Greedy 上的 DeepScaleR 减少了 6.66 %；AIME2025 Non-Greedy 上的 QwQ-32B

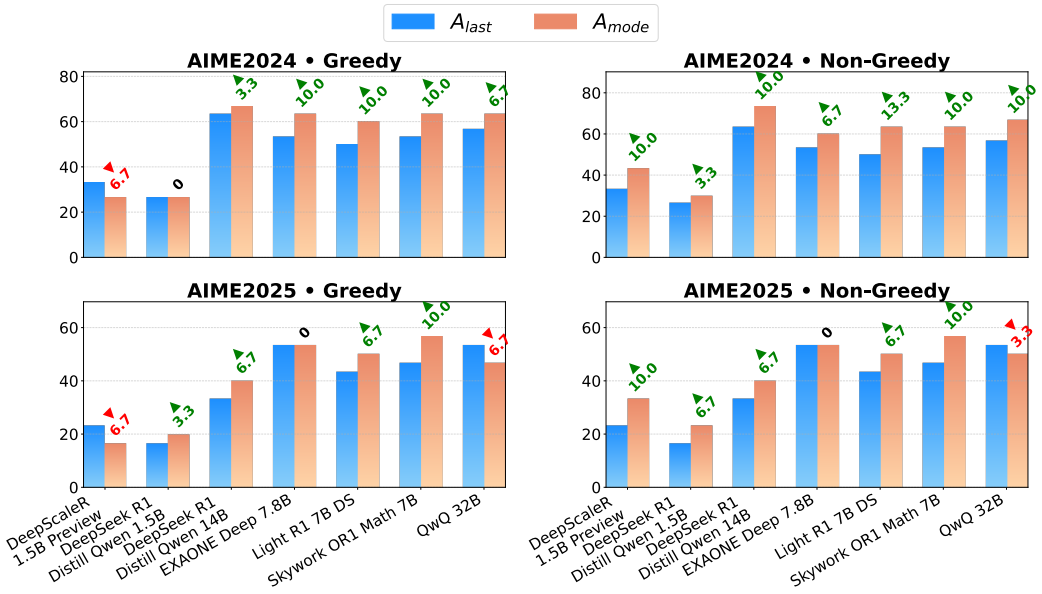


Figure 4: 准确性比较：最后答案与最频繁答案。通过在不同模型和 AIME 数据集上使用贪心和非贪心子思维完成，比较最后答案的准确性（Acc_{Last}，蓝色）和最频繁答案的准确性（$\text{Acc}_{MostFreq}$，橙色）。柱状图上方的数字显示绝对增益（$\text{Acc}_{MostFreq} - \text{Acc}_{Last}$）。绿色向上的三角形表示改进，红色向下的三角形表示下降。我们的方法始终能提高或匹配基线准确性。

减少了 3.33 %）可能归因于噪声，特别是对于较小的模型而言，或者是特定问题交互导致的，而不是系统性缺陷。

在比较子思维完成策略时，非贪婪完成往往比贪婪完成带来稍大或更频繁的改进，尤其是在 AIME2024 的结果中表现明显（例如，对比 Light-R1、Skywork、QwQ-32B 的左上和右上面板）。这表明通过采样探索替代推理路径（非贪婪）通常在揭示模型的稳健一致答案方面比简单地从中间状态加强最可能路径（贪婪）更有效。尽管如此，贪婪完成也在基准方法上提供了一致的好处。

重要的是，这种改进模式在一组多样化的模型（参数范围从 1.5B 到 32B）以及两个具有挑战性的 AIME 数据集上仍然适用。这种一致性突显了分析子思维稳定性的一般适用性和稳健性，并且使用模式答案比单一的最终答案更可靠地指示模型的推理结果。

5 结论

我们证明了仅基于推理过程的最终答案来评估大型语言模型可能不是最优的。通过分析单个推理过程中的中间子想法，并汇总从完成这些部分想法中得出的答案，我们展示了以下内容：

Conclusions:

1. 模式聚合增强了准确性：在源自中间子思路的补全中选择最频繁的答案（A_{mode}），相比于仅依靠初始轨迹的最终答案（A_{last}），显著提升了准确性。在各种模型上，观察到在 AIME2024 上达到 +13 % 以及在 AIME2025 上达到 +10 % 的增益。
2. 答案一致性信号可靠性：从子思维生成的答案的分布提供了一个有价值的信号。高一致性（低熵）与基线正确解（A_{last}）强烈相关，而高波动（高熵）是错误解决方案或模型挣扎的特征。这表明可以使用分布指标进行置信度估计或错误检测的潜力。

3. 非贪婪完成通常能最大化收益：尽管贪婪和非贪婪子思维完成都通过模式聚合提高了准确性，但非贪婪采样（T=1.0, top-p=0.95）经常会带来更大的提升，可能是因为它更好地探索了初始路径段周围的推理空间。

我们的研究结果强调了超越最后一步来更好地理解和评估 LLM 推理能力的重要性，提供了一种简单但有效的方法来从现有的推理过程中提取更可靠的答案。未来的工作可以探索将这些一致性信号整合到解码或训练中。

6

致谢

这项工作由 KAUST 生成性人工智能卓越中心在奖项编号 5940 下支持。计算资源由 IBEX 提供，该资源由 KAUST 的超级计算核心实验室管理。

References

[1] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

[2] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

[3] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

[4] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024.

[5] Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

[6] Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. Successive prompting for decomposing complex questions. *arXiv preprint arXiv:2212.04092*, 2022.

[7] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

[10] Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024.

[11] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

[12] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[13] Fangkai Jiao, Chengwei Qin, Zhengyuan Liu, Nancy F Chen, and Shafiq Joty. Learning planning-based reasoning by trajectories collection and process reward synthesizing. *arXiv preprint arXiv:2402.00658*, 2024.

[14] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[15] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*, 2022.

[16] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, 2023.

[18] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*, 2024.

[19] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*, 2021.

[20] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[21] Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Luke Zettlemoyer, and Hannaneh Hajishirzi. Prompting contrastive explanations for commonsense reasoning tasks. *arXiv preprint arXiv:2106.06823*, 2021.

[22] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[23] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[25] Mirac Suzgun and Adam Tauman Kalai. Meta-prompting: Enhancing language models with task-agnostic scaffolding. *arXiv preprint arXiv:2401.12954*, 2024.

[26] Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. *arXiv preprint arXiv:2310.05707*, 2023.

[27] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022.

[29] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[30] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

[31] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

A 熵图

图 5 和 6 显示了 AIME2024 和 AIME2025 的次级答案分布的平均熵。我们在第 4.3 节中观察到，正确解决的问题的平均熵总是低于错误解决的问题，突显了模型在面对难以正确解决的问题时，往往会提供多样的次级答案选择。

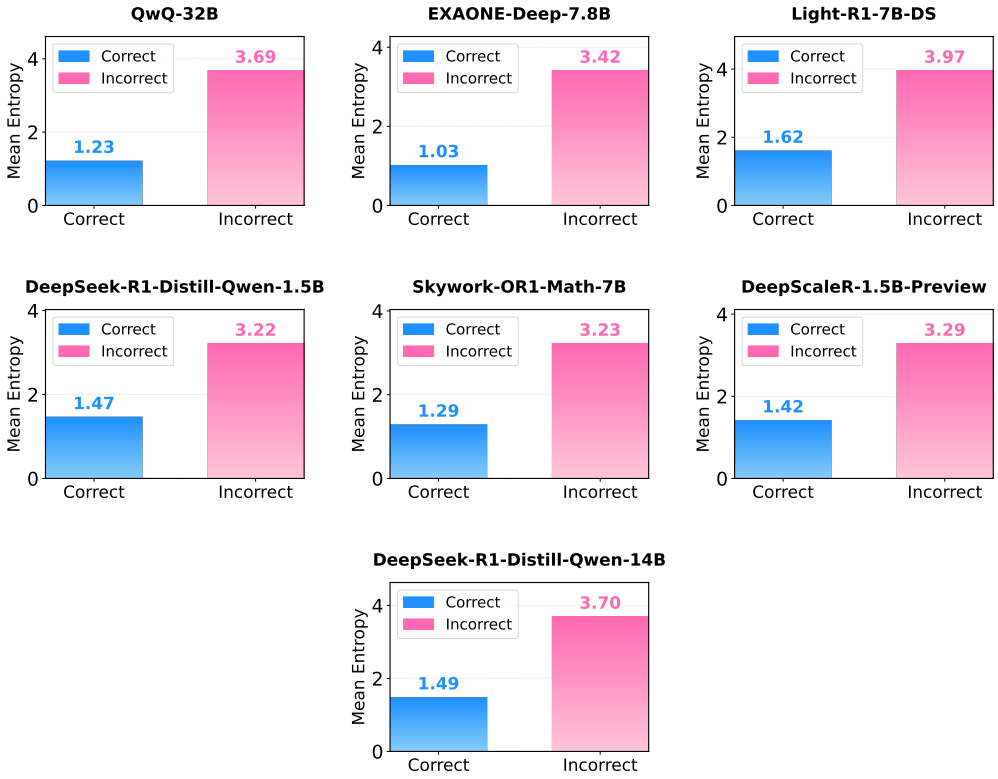


Figure 5: 在 AIME2024 上子思想回答分布的平均熵（贪心完成）。比较正确回答的问题（$A_{last} = A_{true}$）和错误回答的问题（$A_{last} \neq A_{true}$）。较低的熵与正确答案相关。

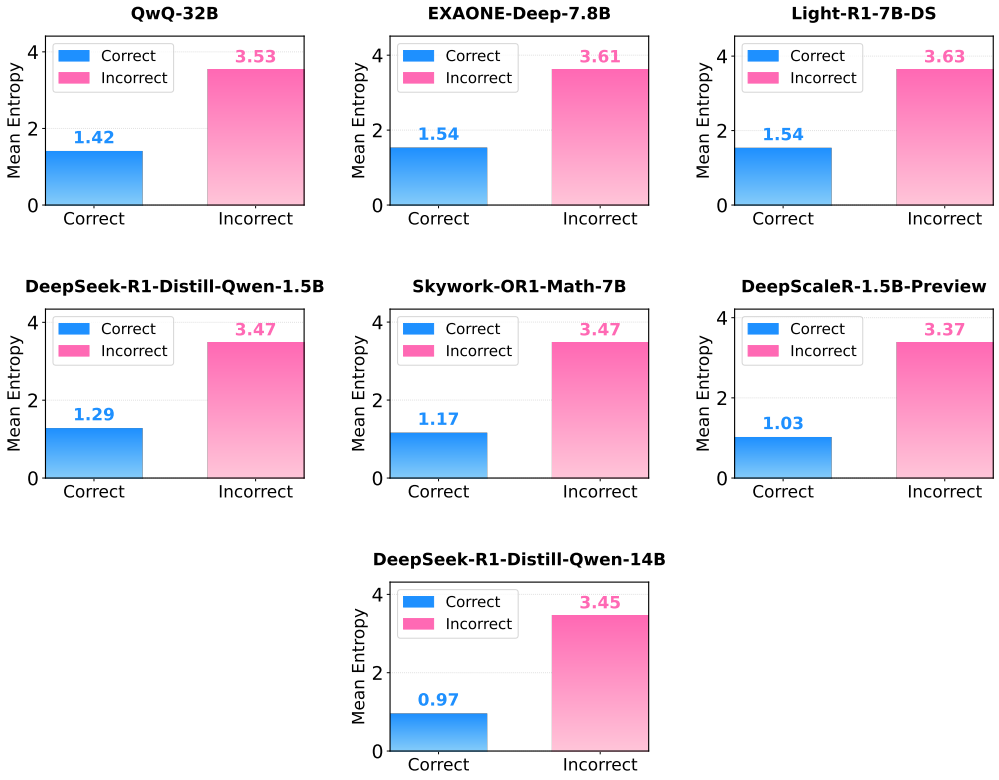


Figure 6: 在 AIME2025（贪心完成）上的子思维答案分布的平均熵。正确回答问题（$A_{last} = A_{true}$）和错误回答问题（$A_{last} \neq A_{true}$）之间的比较。较低的熵与正确答案相关。