# 超越正确性：通过多步自动定理证明揭示推理中的大语言模型生成的逻辑缺陷

郑欣怡*
华中科技大学科学与技术中国武汉
xinyi_zheng@hust.edu.cn

Ningke Li*新加坡国立大学新加坡
ningke_li@u.nus.edu

栾晓坤北京大学北京，中国
luanxiaokun@pku.edu.cn

Kailong Wang†华中科技大学武汉，中国
wangkl@hust.edu.cn

Ling Shi南洋理工大学新加坡
lingshi@ntu.edu.sg

Meng Sun北京大学北京，中国
sunm@pku.edu.cn

Haoyu Wang
*Huazhong* 科技大学武汉，中国
haoyuwang@hust.edu.cn

arXiv:2512.23511v1 [cs.SE] 29 Dec 2025

## 摘要

大语言模型（LLM）展现出了令人瞩目的推理能力，因此被应用于医疗保健、法律和科学研究等高风险领域。然而，其推理往往包含由流畅语言掩盖的细微逻辑错误，这给关键应用带来了重大风险。尽管事实核查、自一致性方法和基于规则的验证等现有方法提供了部分解决方案，但它们无法检测多步推理中的复杂逻辑缺陷。

为应对这些挑战，我们提出了 MATP，这是一种通过多步自动定理证明系统验证大语言模型推理的评估框架。MATP 将自然语言推理转换为一阶逻辑（FOL），并应用自动定理证明器逐步评估逻辑有效性。该方法能够识别隐藏的逻辑错误，并对推理正确性进行细粒度分类。在一项基准测试中，我们使用来自 10 个大语言模型、涵盖 PrOntoQA-OOD、ProofWriter 和 FOLIO 等任务的 10,830 个推理实例进行了评估。结果表明，MATP 在推理步骤验证方面比基于提示的基线方法高出 42 个百分点以上。此外，MATP 还揭示了模型层面的差异：推理模型生成的输出在逻辑上比通用模型更加连贯。这些结果表明，MATP 有望提升大语言模型生成推理的可信度。

## 1 引言

大语言模型（LLM）在自然语言理解与生成方面取得了显著进展，展现出强大的推理能力 [2, 17, 36],推理 [48], 和决策能力 [10, 54]。它们的成功推动了其在医疗保健 [6], 法律 [21], 金融 [26],和科学研究 [57, 59], 等高风险领域的应用，在这些领域，可靠的推理至关重要。然而，安全部署仍然面临挑战：尽管能够生成流畅且看似连贯的输出，大语言模型却经常表现出细微的逻辑错误、矛盾或缺乏依据的推断 [25, 31, 35]，可能导致误诊、错误的法律判决或不准确的研究结论等严重后果。近期的相关工作旨在应对这些局限性的工作包括大型推理模型（LRM），其通过神经符号人工智能、强化学习和因果推断来增强大语言模型。然而，苹果公司近期的分析 [41] 表明，即使是大型推理模型也难以应对复杂任务，往往依赖浅层启发式方法，而非稳健的推理。此外，大语言模型的推理仍容易受到对抗攻击的影响，在这类攻击中，轻微的输入扰动就可能系统性地破坏推理链 [11, 51]。

现有研究主要关注检测大语言模型输出中的推理缺陷，但现有方法存在一些关键局限。事实核查方法 [3, 4, 44] 利用外部知识库验证事实准确性，却往往会忽略逻辑错误，尤其是在推理涉及新颖或隐式的事实组合时。自一致性方法 [50] 通过对多个输出进行多数投票来减少随机错误，但无法发现跨多次生成反复出现的系统性错误。基于规则和事后验证的方法 [20, 40, 53]可以标记简单的矛盾，但难以处理复杂的多步逻辑，因此只能有限地覆盖推理缺陷。

先前采用形式化方法增强大语言模型推理的研究 [32, 34, 56], 提供了宝贵的见解。我们认为，将形式化方法用于评估而非增强大语言模型推理质量，是开展严谨分析的一个有前景的方向。尽管这些方法在理论上颇具吸引力，仍有若干相互交织的挑战有待解决。**挑战#1：大语言模型生成的推理缺乏显式逻辑结构**。与形式化推理系统不同，大语言模型通过概率模式匹配而非结构化推理生成推理内容，因此推理链冗长且隐式。将这种非结构化推理形式化为精确的逻辑表示极具挑战，尤其是与 Logic-LM 相比 [34],，后者只对预先定义的推理任务进行形式化。**挑战#2：大语言模型中细微且不易察觉的推理错误**。单个错误或缺乏依据的推理步骤，就可能在不被察觉的情况下污染后续推理，得出看似合理却根本不可靠的结论。识别并隔离这类错误十分困难，因为这需要在整条推理链中精确追踪并验证逻辑依赖关系。

为应对上述挑战，我们提出多步自动定理证明（MATP），这是一种事后评估框架，可对大语言模型生成的推理进行细粒度分析。MATP 利用大语言模型的自然语言理解

*两位作者对本研究的贡献相同。
†Kailong Wang（wangkl@hust.edu.cn）是通讯作者。

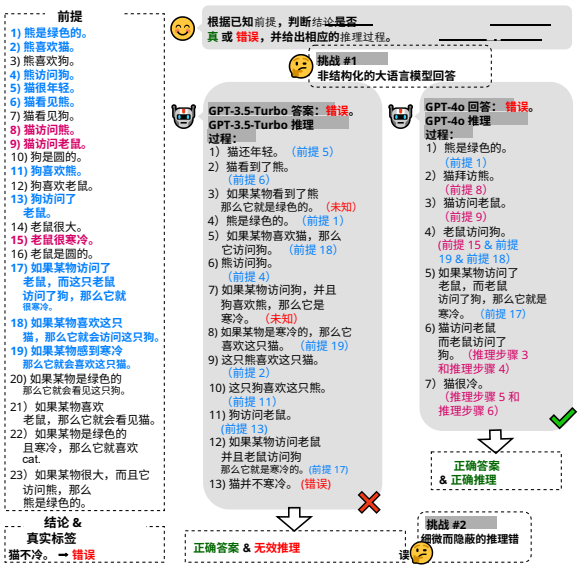


**图1：动机示例。GPT-3.5- Turbo 使用的前提以 亮蓝色标记；GPT-4o 包含这些前提，并添加了 深洋红色前提。GPT-3.5-Turbo 的推理无效，而 GPT-4o 利用前提8、9和15，通过有效的推理过程得出了正确答案。**

能力，在单次处理中将推理任务和模型生成的推理链都转换为统一的一阶逻辑（FOL）表示**挑战 #1**，从而能够借助自动定理证明器（ATP）进行结构化验证。随后，它会验证每个推理步骤的正确性，并在推理链中寻找一条有效证明路径，以逻辑上蕴含最终结论，从而区分真正的演绎推理与单纯的信息复述**挑战 #2**。根据验证结果，MATP 对推理链进行细粒度分类，判断其是完全有效、部分存在缺陷，还是逻辑上不相容。此分类综合考量答案正确性、步骤级有效性和证明路径的可靠性，超越二元准确率度量指标，从而深入洞察推理质量。通过这一设计，MATP 能够对大语言模型在复杂演绎任务中的推理表现进行更易解释、更可靠的评估。

**实验结果与发现**。我们通过在两个数据集上进行大量实验来评估MATP：(1) 一个基准测试集，包含来自PrOntoQA-OOD [39], ProofWriter [43], 和FOLIO [15], 的推理任务，以及来自 10 个主流大语言模型（包括五个通用模型和五个推理模型）的响应，共计 10,830 个实例；(2) 一个基于Pr OntoQA-OOD的分类数据集，通过对标准推理链施加受控扰动构建而成。在该基准数据集中，MATP 将自然语言句子转换为一阶逻辑表示，在PrOntoQA-OOD 和ProofWriter中语法较简单的任务上准确率超过 99%，同时在语言复杂度更高的FOLIO任务上也保持了相对出色的表现。在验证单个推理步骤时，MATP 始终比基于提示的基线方法高出 42 个百分点以上的宏平均 F1 分数。在分类数据集上，MATP 分类准确率达到 99%，显著优于所有基线方法，展现出区分细粒度推理类型的能力，尽管偶尔仍会出现误分类。

基于 MATP 对基准数据集中所有大语言模型生成响应的评估和分类，我们观察到，推理模型——尤其是 DeepSeek-R1——尽管答案准确率相近，却比通用模型产生了逻辑连贯性更高的推理链。此外，大多数评估失败源于自然语言到一阶逻辑翻译阶段的错误。我们进一步识别并总结了常见的失败模式，为指导未来改进大语言模型生成推理的评估提供了参考。

**贡献**。本文的主要贡献如下：

- **针对大语言模型生成推理的事后评估框架**。我们提出了 MATP，这是一种集成形式化方法的框架，通过验证步骤级正确性和整体逻辑一致性来评估大语言模型生成的推理。MATP 已在代码仓库 [61] 中开源。
- **新颖的细粒度推理链分类**。我们提出了一种新的分类方案，将推理链区分为可靠、部分存在错误和无效三类，从而能够更细致地评估大语言模型生成的推理质量。
- **对大语言模型生成推理行为的分析**。通过大规模评估和分类，我们揭示了特定于模型的推理模式和失败模式，并识别出常见的 NL2FOL 翻译错误，为未来的改进提供参考。

## 2 背景与动机

### 2.1 大语言模型中的逻辑推理

大语言模型执行的逻辑推理任务通常分为归纳、溯因和演绎推理。其中，演绎推理尤为关键，也是本文工作的核心。它强调基于明确陈述的前提进行严谨、相容且系统的推理，因此在形式化验证、法律推理和安全关键决策等高可信度领域至关重要。近期的神经符号方法已展示出将自然语言转换为形式化表示，以及实现大语言模型与定理证明器集成以进行形式化推理的可行性 [32, 34, 56]。受这些工作启发，我们提出了一种轻量级的分而治之框架，用于评估大语言模型生成的演绎推理链的逻辑有效性。

**一个启发性示例**。图1说明了为何有必要系统地验证大语言模型的推理。当面对来自 ProofWriter 的演绎推理任务时，GPT-3.5-Turbo 和 GPT-4o 都得出了正确的结论，但采用的推理过程却大相径庭。值得注意的是，这两个模型都以自然语言表达推理，却没有明确的逻辑结构，因此难以验证各个步骤及其相互联系是否正确。这凸显了 **挑战 #1**。GPT-3.5-Turbo 的推理存在严重缺陷：第 3 步和第 7 步引入了缺乏依据的信息，而第 13 步则与前提矛盾——但这些错误被流畅的自然语言掩盖了。这说明了识别细微却影响重大的推理错误所面临的 **挑战 #2** 。此外，即使各个步骤看似正确，GPT-3.5-Turbo 生成的整条推理链仍无法支持该结论，这表明答案可能只是偶然得出的。

## 2.2 形式化基础

我们概述了使用一阶逻辑（FOL）和自动定理证明器（ATP）对逻辑推理进行形式化和验证的技术基础。

**一阶逻辑** 是一种用于表示逻辑陈述并进行推理的形式语言。它通过引入谓词和量化变量扩展了命题逻辑，从而能够表达关于某个论域中对象的一般性陈述。一阶逻辑的核心组成部分包括：常量 （表示特定对象）、谓词 （表示属性或关系）、逻辑联结词 （如否定（¬）、合取（∧）、析取（∨）和蕴含（→），以及量词 （∀表示“对所有”， ∃ 表示“存在”）。

评估此类推理的核心标准是逻辑有效性。若且唯若结论 $C$ 在满足前提集合 $P = \{P_1, P_2, \ldots\ldots, P_n\}$ 中所有前提的每个模型（即对符号的每种可能解释）中都为真，则结论 $C$ 是该前提集合的逻辑结果，记作 $P \models C$。有效性这一性质确保不存在前提为真而结论不成立的反例，从而为正确性提供了严格标准。在一阶逻辑等可靠且完备的逻辑系统中，这种语义蕴涵（$\models$）等价于句法可证性（$\vdash$），这意味着有效的结论也可以通过有限个形式推理步骤从前提中推导出来。我们的工作旨在自动检查大语言模型生成的内容是否相对于给定前提满足这一有效性标准。

**自动定理证明器** 是专门设计的工具，用于自动判定 FOL 等形式系统中逻辑蕴涵的有效性。包括 Vampire 在内的最先进自动定理证明器 [19]（本研究使用了 Vampire）主要基于反驳证明原理运行。为验证逻辑有效性 $P \models C$，证明器会通过机械化方式尝试为 $P \cup \{\neg C\}$ 的不可满足性寻找句法证明，即搜索推导 $P \cup \{\neg C\} \vdash \bot$，其中 $\bot$ 表示矛盾。由于一阶逻辑的完备性，找到这样的证明即可对原始语义蕴涵 $P \models C$ 作出确凿确认。为实现互操作性，我们将一阶逻辑公式转换为标准化的定理证明器问题集（TPTP）格式 [42]。使用 Vampire 这样的高性能自动定理证明器，能够大规模、严格且自动地验证推理步骤。

## 3 方法

我们构建了一个细粒度评估框架，用于评估演绎任务中大语言模型生成的推理，不仅关注最终答案是否正确（§ 3.1）。具体而言，MATP 利用大语言模型将推理任务和模型生成的回答转换为形式化表示（§ 3.2），使用自动定理证明工具和专门设计的算法来验证推理步骤的正确性以及推理链的整体有效性（§ 3.3），并将推理链划分为细粒度类别（§ 3.4）。概览见图2。

### 3.1 任务形式化

给定一组前提和一个候选结论，系统会提示目标大语言模型判断该结论是否能从这些前提中逻辑推出，并生成推理以支持其判断。MATP 将问题陈述和生成的推理内容作为输入，执行形式化验证，以评估各个推理步骤的正确性以及推理链的整体充分性。形式上，MATP 的输入定义为 $D = \{P、C、L、S、A\}$，由五个部分组成：

**(1) 前提 ($P$):** 一组事实陈述 $P = \{P_1, P_2, \ldots, P_n\}$，其中每个 $P_i$ 都假定为真，供后续推理使用。**(2) 候选结论 ($C$):** 基于给定前提 $P$ 进行验证的陈述。按照设计，该陈述必然在逻辑上要么可由 $P$ 推出，要么与之矛盾。

**(3) 真实标签 ($L$):** 人工标注的标签 $L \in \{True, False\}$，表示给定前提 $P$ 下候选结论 $C$ 的正确性。具体而言， $True$ 表示 $P \vdash C$，即 $C$ 可由 $P$ 逻辑推出； $False$ 表示 $P \vdash \neg C$，即 $C$ 与 $P$ 矛盾。

**(4) 推理步骤 ($S$):** 目标大语言模型在推理过程中生成的一系列中间陈述 $S = (S_1, S_2, \ldots, S_m)$，其中每个 $S_i$ 都代表模型推理轨迹中为验证或驳斥结论而生成的陈述。

**(5) 预测答案（$A$）：** 目标大语言模型针对候选结论 $C$ 的正确性给出的最终判断 $A \in \{True, False\}$，代表模型基于推理步骤 $S$ 作出的决定。

基于输入 $D$，MATP 将前提 $P$、候选结论 $C$ 和推理步骤 $S$ 转换为一阶逻辑公式，并使用自动定理证明器进行逻辑验证。它会生成三个关键输出，以便进行全面评估：

**(1) 推理步骤正确性标注：** 对于每个推理步骤，MATP 会分配一个正确性标签 $StepCorrectness(S_i) \in \{True, False, Unknown\}$，表明根据前提 $P$， $S_i$ 是可证、可反驳还是不确定。**(2) 有效证明路径存在性：** MATP 会搜索一组正确的推理步骤 $S_{inProofPath} \subseteq S$，使其共同蕴含结论 $C$ 或其否定 $\neg C$。如果存在这样的有效证明路径，则表明该推理链在逻辑上是健全的。**(3) 推理链的细粒度分类**：通过分析最终预测答案的正确性、各推理步骤的正确性以及是否存在有效证明路径，MATP 将推理输出划分为不同的细粒度类别。

通过这一原则性工作流程，MATP 能够严谨且可解释地评估大语言模型的推理能力，超越表层正确性，揭示潜在的逻辑错误，并深入了解其底层推理结构。

#### 3.2 自然语言到一阶逻辑

该模块将给定的前提 $P$、候选结论 $C$ 和推理步骤 $S$ 转换为一阶逻辑表示，以便通过自动定理证明进行验证。

*3.2.1* 预处理。 为确保将自然语言准确、可靠地翻译为一阶逻辑，我们首先实施一系列预处理措施。**与大语言模型交互。** 为了为后续的形式化和验证生成推理链，我们采用结构化方式与大语言模型进行交互。给定前提 $P$ 和候选结论 $C$，我们提示模型评估 $C$ 是否能够通过明确生成的推理过程从 $P$ 逻辑推出。我们采用少样本提示策略，提供带注释的示例，以

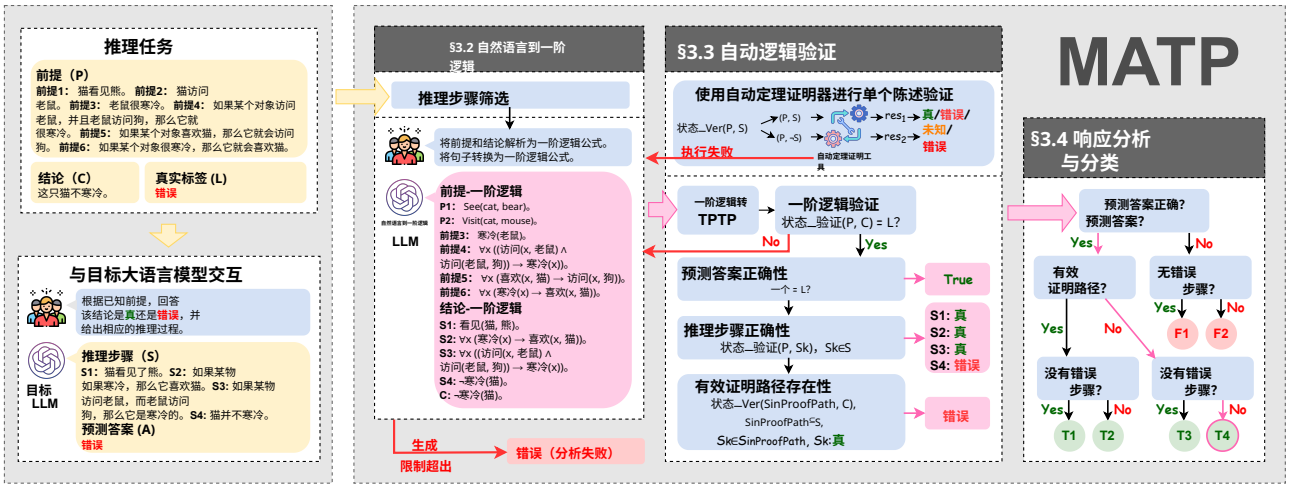


**图2：工作流 的 MATP。黄色节点表示输入，而粉色节点表示中间输出 s.**

**将自然语言转换为一阶逻辑的提示模板**

将给定的前提和结论解析为一阶逻辑公式。### 规则1. 每个句子必须转换为单个一阶逻辑公式。 2. 使用以下一阶逻辑语法：expr1 和 expr2 的逻辑合取： $expr1 \wedge expr2$ ... 3. 每个公式都必须在适用的情况下显式包含量词（∀x 或 ∃x）... 4. 当某个存在实体以新的属性再次出现时，所有属性（包括旧属性和新属性）都应在后续公式中置于单个存在量词的作用域内，并使用合取连接.....（其他更具体的规则）### 输出格式前提 / 结论：一阶逻辑公式 ::: （索引）引用。 ...### 示例前提：（1）奶牛很寒冷。（2）奶牛不是圆的。 ... 结论：（1）老鼠需要奶牛。 ... （9）老鼠吃老鼠。 ### 前提：Cold(cow) ::: （1）奶牛很寒冷。 ¬Round(cow) ::: （2）奶牛不是圆的。 ... 结论：Needs(mouse, cow) ::: （1）老鼠需要奶牛。 ... Eats(mouse, mouse) ::: （9）老鼠吃老鼠。### 你的任务仅返回一阶逻辑公式，不得包含其他文本。前提：{premises}结论：{conclusions}

**图3：将自然语言转换为一阶逻辑的提示模板。**

引导模型的行为模型首先仅根据 $P$生成逐步推理过程，随后再对真/错误 作出判断，确定是否蕴含 $C$。这种先推理后作答的格式将推理轨迹与最终判定分开，从而能够对两者进行细粒度评估。为确保推理忠实可靠，提示明确规定输出应避免不必要的复杂性、模糊表述或依赖外部知识。完整的提示设计见 [61]。

**进一步筛选**。生成推理过程后，输出会被拆分为单独的句子。随后，筛除不确定或推测性的陈述（例如，“Wren 不一定是 gorpus”）。受先前研究 [8], 启发，该研究利用与反思相关的关键词对推理内容进行分段，以识别反思阶段，我们对这种方法进行了扩展。我们通过检测表示歧义的表达（例如“可能”“矛盾”或“不一定”）来移除不确定或推测性的陈述，而不引入额外的筛选模型。剩余的推理步骤简洁明确，有助于后续处理。具体而言，由于难以直接比较自然语言句子，我们先将句子转换为一阶逻辑形式，再对重复陈述进行去重。

*3.2.2* 将自然语言转换为一阶逻辑。受 Logic-LM [34], 启发，该方法成功运用提示工程和 GPT-4 来形式化长文本推理任务。我们沿用其提示设计框架来实现自然语言到一阶逻辑转换。在保留其核心结构的同时，我们针对本任务的具体要求进行了若干调整。

**提示结构**。 如图3所示，提示结构包含五个组成部分： 任务定义、翻译规则、输出格式、示例演示和输入实例，具体如下。特别是，提示中的“结论”始终同时包含推理步骤 $S$ 和来自实际输入的候选结论 $C$。

任务定义确立了将前提和结论转换为相应一阶逻辑表示的主要目标。翻译规则规定了自然语言到一阶逻辑过程的基本约束。这些规则确保每个陈述都严格按照一阶逻辑语法进行翻译，同时遵循一系列质量保证策略。必要时，可根据具体情境灵活调整这些规则。输出格式要求每个生成的输出都包含一阶逻辑形式表达式、索引以及输入中对应的参考句。为确保完整并避免遗漏，前提和结论都按数字编列索引，以适应可能较长的输入。示例演示展示预期的输入与输出关系，从而提升清晰度，并强化对转换原则的正确运用。输入实例提供推理输入，其中“前提”对应前提 $P$，“结论”则包含推理步骤 $S$ 和候选结论 $C$。模型被要求仅输出所需的一阶逻辑公式，不得包含任何多余内容。

**质量保障策略**。在规则组件中，我们实施多种策略，以最大限度地减少常见错误并提升翻译质量。对于**量词的使用**，变量必须明确带有存在量词或全称量词。复杂陈述须使用括号，以界定量词辖域并避免歧义。对于由存在量词限定的实体，可以用多个陈述描述同一实体的不同属性，例如先为实体 $Tom$ 引入属性 $A$，之后再添加属性 $B$。后续翻译

应整合这两种属性，以保证语义完备性。为确保**存在量词实例化中的跨步骤连贯性**，我们通过规则约束，确保始终引用先前步骤中引入的实体及其谓词属性，避免逻辑矛盾。在单次处理中翻译前提、推理步骤和候选结论时，大语言模型的理解能力有助于保持谓词一致性。为进一步确保**谓词使用相容**，我们添加了一条规则，鼓励优先复用现有谓词，再引入新谓词。这可以减少冗余，防止谓词泛滥。为解决**大语言模型对**“异或”和“包含或”的混淆，我们加入有针对性的转换示例，以阐明两者的区别。这些策略的详细实现见完整提示 [61]。

*3.2.3* 重新生成。 大语言模型最初生成的自然语言到一阶逻辑的翻译并不总是正确。如果生成的一阶逻辑在验证过程中出现执行错误，或者候选结论的验证结果与真实标签相矛盾，则会重试 NL2FOL 流程，直至达到固定的最大尝试次数。如果所有重试均失败，则返回错误，表示无法分析输入。重新生成期间不会提供反馈。由于难以定位翻译中的语义错误，我们无法提供有用且具体的反馈。将上一次生成的全部输出用作反馈也是一种选择，但§4.5中的实验表明，这种做法的影响微乎其微。

## 3.3 逻辑验证

该模块旨在使用自动定理证明器对输入陈述（$P$、 $C$、 $S$）的一阶逻辑表示进行逻辑验证。此过程会为每个推理步骤赋予正确性标签，并确定是否存在有效证明路径。

**一阶逻辑到 TPTP**。我们将一阶逻辑公式转换为 TPTP 格式，以兼容标准自动定理证明工具。此过程通过 Python 脚本完成，该脚本确保一阶逻辑与 TPTP 表示之间存在确定性的一一映射。我们会执行预处理步骤，使一阶逻辑公式符合 TPTP 语法，包括将 Unicode 符号替换为 ASCII 字符、规范化谓词和常量名称，以及处理不受支持的逻辑联结词（例如，排斥析取 $A \oplus B$ $\neg A \wedge B \vee A \wedge \overset{B}{\neg}$ 会被改写为 ( ) ( )）。脚本还会进行轻量级语法修复，例如补全缺失的括号或纠正误用的逻辑联结词，从而提高稳健性并减少后续验证中的解析错误。

每个公式都会转换为 TPTP 的一阶形式格式：“fof(名称, 角色, 公式)”，其中 名称 是逻辑陈述的唯一标识符，角色 指定其为公理还是猜想，而 公式 则是一阶逻辑表达式改写为 TPTP 语法后的形式。所有前提都被设为公理，候选结论则被设为猜想。唯一标识符遵循“前提_索引”和“结论”的模式。我们先将自然语言翻译为一阶逻辑，再转换为 TPTP，因为 TPTP 语法更加复杂且冗长，直接将自然语言转换为 TPTP 更容易出错。将一阶逻辑转换为 TPTP 有助于修正语法错误，并确保与定理证明器兼容，从而提高验证的可靠性。

**可证性验证**。 验证过程的一个核心部分是确定陈述 $S$ 是否可由一组前提 $P$证明。对于 $S$，有三种可能的结果：可证，这意味着 $S$ 被 $P$ 逻辑蕴含，并且只要 $P$ 为真；可反驳，这意味着 $S$ 与 $P$ 矛盾，并且当 $P$ 为真时始终为假；不确定，这意味着根据 $P$，既无法证明也无法反驳 $S$。为确定这一点，我们调用证明器，同时检查 $P \vdash S$ 或 $P \vdash \neg S$是否成立。如果其中任一成立，则相应地将$S$ 分类为真或错误。如果两者都不成立（即 $P \cup S$ 和 $P \cup \neg S$ 都相容），则将 $S$ 标记为未知。算法 1 总结了这一过程，其中输出真、错误和未知分别对应三种可证性结果。特别是，如果 $S$ 和 $\neg S$ 都可证，则表明 $P$内部存在矛盾，此时返回错误。

**算法1**使用自动定理证明器进行单个陈述验证

**要求：** 前提 $P = \{P_1, P_2, \ldots\ldots, P_n\}$，候选陈述 $S$
**确保：** 检查结果（**真**，**错误**，**未知**，或**错误**）

```
函数VerifySingleStatement(P, S)
    res_1 ←ATPRun(P, S) 且 res_2 ←ATPRun(P, ¬S)
    如果 res_1 中有反驳且 res_2 中有反驳，则
        返回错误                                  ▷ 中存在矛盾 P
    如果 res_1 中存在反驳，则
        返回真                                    ▷ S 已验证
    如果 res_2 中存在反驳，则
        返回错误                                  ▷ ¬S 已验证
    返回未知
```

**推理步骤验证**。我们首先将Ver-ifySingleStatement 应用于前提 $P$ 和候选结论 $C$， 以检查其是否与真实标签 $L$一致。如果不一致，可能是以下原因造成的：（1）语义上不正确但语法有效的自然语言到一阶逻辑翻译，导致自动定理证明器对 $C$ 作出与 $L$不相容的判断；或（2）真实标签标注有误。由于无法自动区分或纠正这两种情况，该流程会返回自然语言到一阶逻辑阶段重新生成。验证结论后，我们使用相同的前提 $S_i$ 评估每个中间推理步骤 $P$的正确性。如果某个步骤被证伪或判定为不确定，则将其记录为潜在的逻辑错误。算法2概述了这一流程。该算法同时返回预测答案是否正确，以及每个推理步骤的正确性标注。值得注意的是，某些错误步骤可能不会影响最终判断，而另一些则可能损害整个推理链。为区分这两种情况，我们会进一步分析推理步骤中是否存在有效证明路径。

**算法2**推理步骤验证

**要求：** 前提 $P = \{P_1, P_2, \ldots, P_n\}$，候选结论 $C$，真实标签 $L$，推理步骤 $S = \{S_1, S_2, \ldots, S_m\}$，预测答案 $A$
**确保：** 返回 $AnswerCorrectness$，步骤正确性标注 $S_{correctness}$

```
函数 推理步骤验证(P, C, L, S, A)
    如果 VerifySingleStatement(P, C) ≠ L 那么
        返回错误                                  ▷ 自然语言到一阶逻辑翻译错误或标注错误
    S_correctness ← [ ]                           ▷ 初始化步骤正确性列表
    对于 i ← 1 到 m，执行
        追加(S_correctness, VerifySingleStatement(P, S_i))
    AnswerCorrectness ← (L = A)                   ▷ 检查答案是否匹配
    返回 AnswerCorrectness, S_correctness
```

**推理证明路径的验证**。除了评估单个推理步骤之外，我们还会评估推理过程是否包含逻辑上支持结论标签所必需的全部步骤。换言之，我们要确定推理链是否包含一条通向正确结论分类的有效证明路径。为此，我们通过仅纳入那些引入了无法由此前已纳入步骤推导出的新事实的、标记为真的步骤，迭代构建候选证明路径。形式上，证明路径是集合 $S_{inProofPath} \subseteq S$ 的一个子集，其中 a

步骤 $S_i$ 当且仅当其被标记为 真 且无法由 $S_{inProofPath}$ 蕴含时才会被加入。处理完所有步骤后，我们会检查候选结论 $C$ 的真值是否与真实标签 $L$ 匹配，评估时使用证明路径 $S_{inProofPath}$。如果为真，则表明推理过程包含一条支持结论标签的有效证明路径。否则，该证明路径将被视为无效。该验证过程详见 算法 3。

**算法 3** 推理证明路径验证

**要求：** 候选结论 $C$、真实标签 $L$、推理步骤 $S = \{S_1, S_2, \ldots\ldots, S_m\}$，步骤正确性标注 $S_{correctness}$
**确保：** 返回 $HasValidProofPath$
函数 VerifyReasoningProofPath($C$, $L$, $S$, $S_{correctness}$)
  $S_{inProofPath} \leftarrow [\,]$
  对于 $i \leftarrow 1$ 到 $m$ 执行
    如果 $S_{correctness\_i}$ = 真 则
      如果 VerifySingleStatement($S_{inProofPath}$, $S_i$) = 未知 则
        Append($S_{inProofPath}$, $S_i$) ▷ 构建证明路径
  $HasValidProofPath \leftarrow$ (VerifySingleStatement($S_{inProofPath}$, $C$) = $L$)
  返回 $HasValidProofPath$

如果自动定理证明器遇到执行错误，表明生成的 一阶逻辑 内容存在语法缺陷，我们将返回自然语言到一阶逻辑阶段重新生成。

## 3.4 响应分析与分类

在前两个阶段之后，超过自然语言到一阶逻辑最大重试次数的输入会被归类为 **错误**，这表明我们的框架无法处理这些输入。对于成功处理的输入，本模块基于逻辑验证对推理链进行细粒度分类，以全面评估推理过程。我们不仅评估最终的预测答案，还分析各个推理步骤和整体推理链，以识别潜在的逻辑缺陷。这使我们能够区分那些偶然正确的响应与基于有效推理的响应。经过逻辑验证后，我们得到三个关键的评估维度：(1) 预测答案的正确性；(2) 每个推理步骤的正确性；以及 (3) 推理链中是否存在有效证明路径。基于这些维度，我们定义了六种推理类别，如 图2所示。

对于预测答案错误的响应，我们根据中间推理步骤是否包含错误，将其分为两类。任何标记为 *False* 或 *Unknown* 的推理步骤都被视为错误。**F1** 表示所有推理步骤均正确但最终答案错误的情况，这表明目标模型可能未能构建有效的证明路径。**F2** 涵盖最终答案和至少一个推理步骤均错误的情况，这表明中间错误可能会误导推理过程，或者模型未能形成连贯的证明。

对于预测答案正确的回答，我们评估每个步骤的正确性以及推理链的有效性。**T1** 表示所有步骤均正确且存在有效证明路径的理想情况。**T2**包括存在一些错误步骤，但证明路径仍然有效的情况，这表明模型具有一定的鲁棒性。**T3** 表示所有推理步骤均正确，但推理链并不构成有效证明；这说明模型正确推断出了答案，却未能提供逻辑完备的推理路径。**T4** 包括步骤层面存在错误且没有有效证明路径的回答，这引发了人们对正确答案是否只是偶然得出或源于有缺陷推理的担忧。

# 4 评估

我们的评估聚焦于以下研究问题：

- **研究问题 1（有效性）：** MATP 能否准确验证大语言模型生成的推理步骤是否正确?
- **研究问题2（推理链分类）：** MATP 能否有效地对不同类型的推理链进行分类?
- **研究问题3（模型推理分析）：** 不同大语言模型的推理质量如何？它们在生成推理链时分别表现出哪些特征?

## 4.1 实验设置

**基准数据集。** 我们从三个侧重推理的数据集中构建基准测试集，这些数据集均专为自然语言推理任务设计。PrOntoQA-OOD [39] 和 ProofWriter [43]专注于结构化本体中的逻辑推理，着重考察类别、属性和基于规则的推理；FOLIO [15]则侧重于真实场景下的语义推理，涉及关于人物、地点、数量和事件的知识。这些数据集中的每个测试用例均由前提、候选结论和真实标签（*True*、*False*或 *Unknown*）组成。PrOntoQA-OOD 和 ProofWriter 中的测试用例使用结构化自然语言，而 FOLIO 中的测试用例则采用更贴近日常人类语言的表达。

- **PrOntoQA-OOD** [39]（选取了 384 个测试用例）是一个合成数据集，其中所有结论都标记为 *True*。任务根据所需的最少推理步骤数进行分类；我们从 4 跳子集中抽样，以确保更高的复杂度。

- **ProofWriter** [43]（选取了 344 个测试用例）使用更自然的语言和更复杂的逻辑。我们从其 5 跳测试集中抽样，并排除带有 *Unknown* 标签的任务，因为其蕴含关系不明确。
- **FOLIO** [15]（选取了355个测试用例）是一个基于现实世界事实构建的人工编写的数据集。我们从训练集中进行抽样，仅保留 *True*/*False* 个案例，并舍弃结论只是简单重复某个前提的实例。

为平衡数据集规模并聚焦于能够 eliciting 更丰富模型响应的高跳数实例，我们从 PrOntoQA-OOD 和 ProofWriter 的最大跳数子集中随机抽取测试用例，而不是使用所有可用数据。对于缺乏基于跳数划分的 FOLIO，我们从其训练集中随机抽样。对于每个选定的测试用例，我们使用10个具有代表性的大语言模型生成推理链和预测答案（详见下一节）。用于生成的提示模板见 [61]。由此共得到10830个基准实例。每个实例都包含测试用例提供的前提、候选结论和真实标签，以及目标大语言模型生成的推理链和预测答案。该基准用于评估 MATP 的有效性（研究问题 1），并分析不同模型之间的推理链类别（研究问题3）。

**受测模型。** 为确保评估全面，我们使用 MATP 测试了 10 个大语言模型，包括五个通用模型和五个推理模型。通用模型包括 GPT-3.5- turbo、GPT-4o [1, 18], LLaMA3.1-8B-chat-hf、LLaMA3.1-70B-chat- hf [13] 和 Qwen2.5-7B-Instruct [46]。推理模型包括 QwQ-32B [37], DeepSeek-R1-Distill-Qwen-7B、DeepSeek-R1- Distill-Qwen-32B[1], DeepSeek-R1 [14] 和 GPT-o4-mini [33]。

[1]为简洁起见，我们用 DS-R1-Qwen7B/32B 指代 DeepSeek-R1-Distill-Qwen-7B/32B。

**表 1：对基准测试集上的 MATP 进行评估：自然语言到一阶逻辑翻译准确率和推理步骤正确性分类性能。自然语言到一阶逻辑准确率（§ 4.2.1）采用四项度量指标进行评估——ER（执行率）、EA（执行准确率）、FB（一阶逻辑 BLEU）和 LE（逻辑等价性）；推理步骤正确性（§4.2.2）则通过宏平均 F1 分数衡量。所有数值均以百分比（%）表示，分数越高表示性能越好。在度量指标行中，GPT-4o 和 DS-R1 分别指使用 GPT-4o 和 DeepSeek-R1 实现的基于提示的基线方法。**

| 数据集 / 目标大语言模型 | PrOntoQA-OOD（384） | | | | | | | ProofWriter（344） | | | | | | | FOLIO（355） | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ER | EA | FB | LE | 宏平均 F1 分数 MATP | GPT-4o | DS-R1 | ER | EA | FB | LE | 宏平均 F1 分数 MATP | GPT-4o | DS-R1 | ER | EA | FB | LE | 宏平均 F1 分数 MATP | GPT-4o | DS-R1 |
| GPT-3.5 | 99.22 | 100.0 | 100.0 | 100.0 | 87.13 | 54.33 | 39.94 | 99.71 | 99.71 | 83.88 | 99.91 | 100.0 | 43.03 | 38.39 | 96.34 | 69.59 | 36.98 | 88.37 | 91.93 | 57.11 | 43.21 |
| GPT-4o | 98.96 | 100.0 | 100.0 | 100.0 | 100.0 | 49.75 | 47.92 | 99.71 | 99.42 | 84.14 | 100.0 | 100.0 | 33.08 | 33.33 | 96.06 | 70.09 | 37.11 | 87.78 | 100.0 | 42.23 | 31.06 |
| LLaMA3.1-8B | 99.22 | 100.0 | 100.0 | 100.0 | 100.0 | 38.53 | 49.74 | 99.71 | 100.0 | 83.88 | 99.91 | 100.0 | 46.77 | 40.00 | 95.49 | 70.50 | 36.84 | 88.52 | 89.87 | 68.01 | 47.08 |
| LLaMA3.1-70B | 99.22 | 99.74 | 100.0 | 100.0 | 55.42 | 49.59 | 69.69 | 99.71 | 97.38 | 83.69 | 99.91 | 98.61 | 46.10 | 43.05 | 94.37 | 71.64 | 36.46 | 88.20 | 93.51 | 38.53 | 42.12 |
| Qwen2.5-7B | 98.96 | 100.0 | 100.0 | 100.0 | 100.0 | 44.43 | 32.80 | 99.42 | 99.42 | 84.14 | 100.0 | 96.34 | 52.80 | 47.29 | 97.18 | 72.46 | 36.35 | 88.15 | 79.69 | 49.34 | 38.40 |
| QwQ-32B | 99.22 | 100.0 | 100.0 | 100.0 | 100.0 | 49.37 | 49.79 | 99.13 | 99.41 | 84.28 | 100.0 | 100.0 | 72.70 | 49.27 | 95.49 | 71.09 | 36.63 | 88.16 | 88.75 | 37.29 | 36.55 |
| DS-R1-Qwen7B | 99.22 | 99.74 | 100.0 | 100.0 | 100.0 | 44.27 | 39.96 | 99.42 | 98.25 | 84.14 | 100.0 | 98.84 | 62.44 | 36.66 | 96.34 | 70.76 | 37.81 | 88.65 | 72.08 | 41.72 | 39.66 |
| DS-R1-Qwen32B | 99.22 | 99.74 | 100.0 | 100.0 | 100.0 | 49.58 | 83.13 | 99.42 | 99.42 | 84.28 | 100.0 | 66.47 | 32.52 | 32.42 | 96.62 | 71.43 | 37.29 | 88.92 | 72.82 | 38.53 | 33.70 |
| DeepSeek-R1 | 99.22 | 99.74 | 100.0 | 100.0 | 100.0 | 49.82 | 100.0 | 99.42 | 99.42 | 84.21 | 100.0 | 52.16 | 49.19 | 49.35 | 97.18 | 70.72 | 37.06 | 87.82 | 61.33 | 32.82 | 31.33 |
| GPT-o4-mini | 98.18 | 100.0 | 100.0 | 100.0 | 100.0 | 48.19 | 100.0 | 99.71 | 100.0 | 84.07 | 100.0 | 100.0 | 54.96 | 54.49 | 95.49 | 70.50 | 36.58 | 88.45 | 94.24 | 46.13 | 32.40 |
| 平均值 | 99.06 | 99.90 | 100.0 | 100.0 | 94.26 | 47.79 | 61.30 | 99.53 | 99.24 | 84.07 | 99.97 | 91.24 | 49.36 | 42.43 | 96.06 | 70.88 | 36.91 | 88.30 | 84.42 | 45.17 | 37.55 |

**模型配置。** 为获得稳定且相容的推理输出，我们将温度 设为 0，并将*top-p* = 1 和*top-k* = 0 设为较高概率词元提供倾向。GPT-o4-mini 是个例外，由于模型限制，其温度固定为 1。

**基线。** 按照 [58]，我们将自身方法与基于提示的基线进行比较，使用 GPT-4o 和 DeepSeek-R1（研究问题 1 和研究问题2）。提示会引出两种输出：(1) 对推理步骤的逐步判断（*True*、*False*或*Unknown*）；(2) 关于推理是否支持预测答案（*Yes*或*No*）的最终判断；两者均附有解释。完整提示见 [61]。

**实现。** 我们基于 MetaGPT [16]，（一种用于多智能体系统的开发框架，可协调工具调用并管理流程阶段），整合现有组件，将评估框架实现为自动化工具链。具体而言，我们使用 GPT-4o 进行自然语言到一阶逻辑翻译，利用其处理长篇复杂自然语言任务的出色表现，并将每个输入的生成尝试次数限制为三次。对于自动验证，我们采用广泛使用的自动定理证明器 Vampire [19]。所有实现细节见 [61]。

**运行环境。** 所有实验均在一台运行 Ubuntu 22.04 的服务器上进行，该服务器配备两颗 64 核 AMD EPYC 7713 CPU、512 GB 内存和两块 NVIDIA A100 PCIe 80GB GPU。

## 4.2 研究问题 1：有效性

我们在基准数据集上进行评估，以考察 MATP 的有效性。评估聚焦于两个方面：自然语言到一阶逻辑翻译的准确率，以及识别各个推理步骤正确性的能力。

*4.2.1* 自然语言到一阶逻辑翻译的准确率。 我们从整体和句子两个层面评估自然语言到一阶逻辑的翻译。

**整体层面的度量指标。** 遵循 Logic-LM [34]，，我们使用执行率（ER）和执行准确率（EA）来评估整个基准测试集上的自然语言到一阶逻辑翻译。ER 衡量 Vampire 能够成功执行的一阶逻辑输出所占比例，反映语法正确性。EA 衡量执行成功时 Vampire 的判定与真实标签相符的频率，表明语义是否得到保留。

**句子级度量指标。**我们采用 LogicLLaMA [55] 中的一阶逻辑 BLEU（FB）和逻辑等价性（LE）来评估句子级转换。FB 使用针对一阶逻辑的专用分词方法计算相似度分数，而 LE 则判定生成的一阶逻辑公式与参考一阶逻辑公式在逻辑上是否等价。对于包含参考一阶逻辑公式的数据集（FOLIO），我们对所有前提计算 FB 和 LE。对于其他数据集（PrOntoQA-OOD、ProofWriter），我们为每个数据集的 300 条前提人工标注参考一阶逻辑公式，用于度量指标评估，并由三位作者交叉核验以确保一致性。

**结果。** 如 表 1所示，PrOntoQA-OOD 和ProofWriter 上的 ER 和 EA 均接近 100%，表明语法和语义对齐效果良好。相比之下，FOLIO 的 ER 较高（96.06%），但 EA 较低（70.88%），这表明在保留复杂陈述的语义方面仍面临挑战。在句子层面，PrOntoQA-OOD 上的 FB 和 LE 评分，以及ProofWriter 上的 LE 评分均接近 100%。然而，ProofWriter的 FB 降至 84.07%，主要是因为模型输出与基本形式标注在形式上不匹配（例如，“喜欢”与“喜欢”），这凸显了 FB 对表层差异的敏感性。在 FOLIO上，FB 评分降至 36.91%，而 LE 仍相对较高（88.30%），反映出尽管文本相似度较低，逻辑上仍等价。这种差异可以理解，因为对于复杂句子，谓词或常量名称的选择不太可能始终与预期形式一致。

为更好地理解 FOLIO 上较低的执行准确率，我们人工审查了 MATP 对候选结论给出的标签与真实标签不一致的案例。导致这一问题的关键因素将在§4.4.3中详细分析。

*4.2.2* 识别各推理步骤正确性的能力。本评估关注 MATP 是否为每个推理步骤正确标注，以及其表现与基线相比如何。

**指标。**我们将该任务表述为一个三分类问题（*True*、*False*、*Unknown*），并使用宏平均 F1 分数进行评估。鉴于标注成本较高，我们针对每个数据集，从十个模型中各抽取 20 个 ER 贡献任务，共计 600 个样本。三位作者对这些样本中的每个推理步骤的正确性进行人工标注，并根据汇总后的标注计算最终的宏平均 F1 分数。

**结果。** 详细结果见 表 1。 MATP 在推理步骤正确性分类任务中始终优于 GPT-4o 和 DeepSeek-R1 基线方法。在 PrOntoQA-OOD 和 ProofWriter上，MATP 分别取得了 94.26% 和 91.24% 的 F1 值，显著高于 GPT-4o（47.79%、49.36%）和 DeepSeek-R1（61.30%、

42.43%）。即使在形式化程度较低的 FOLIO 数据集上，MATP 仍保持了 84.42% 的 F1 值，而基线方法的 F1 值则降至 46% 以下。我们进一步分析了基线方法表现不佳的原因。具体而言，当推理是隐含的或需要多跳推理时，基线方法经常将正确的步骤误标为 *False* 或*Unknown*。它们还难以区分 *False* 和*Unknown*，导致二者之间频繁混淆。

研究问题 1 的回答

MATP 通过将自然语言准确转换为一阶逻辑并识别其正确性来验证推理步骤，在所有数据集上的表现均优于基于提示的基线方法。它在 PrOntoQA-OOD 和 ProofWriter 上表现良好，但由于语言模式更加复杂，在 FOLIO 上表现欠佳。

## 4.3 研究问题2：推理链分类

为了评估 MATP 能否准确分类推理链，我们在一个分类数据集上进行了评估，并将其表现与基线方法进行比较。

**分类数据集**。 我们基于 200 个 PrOntoQA-OOD 任务构建了一个分类数据集。以 PrOntoQA-OOD 提供的标准推理链（T1 类）为基础，我们进行有针对性的变异，以获得不同类型的推理链：为生成 T2 类推理链，我们插入一个否定前提，以引入错误的推理步骤；对于 T3 类推理链，我们移除关键步骤并注入无关内容，以破坏逻辑流程；对于 T4 类推理链，我们同时应用 T2 类和 T3 类变异。每种变体均针对 50 个任务生成，以确保四种类型之间的均衡覆盖。

**结果**。 图4 展示了 MATP 与基线方法相比的分类准确率。如图所示，MATP 在 T1–T3 上实现了 100% 的准确率，在 T4 上达到 96%。对两个 T4 误分类案例的分析表明，这些错误源于轻微的自然语言到一阶逻辑翻译偏差，这表明只要翻译正确，MATP就能始终准确地对推理链进行分类。相比之下，基于 GPT-4o 的基线方法准确率较低，在 T1–T4 上分别为 78%、72%、84% 和 90%。错误分析揭示了两个主要问题：(1) 54 个正确步骤被误判为错误，导致 14 个 T1/T3 案例被分类为 T2/T4；(2) 该方法难以评估各步骤是否充分支持结论，导致 20 个 T1/T2 案例被误分类为 T3/T4，以及 11 个 T3/T4 案例被误分类为 T1/T2。基于 DeepSeek-R1 的基线方法表现更好，准确率达到 96%、90%、
准确率分别为 92% 和 82%。大多数误分类源于未能评估各步骤是否充分支持结论——6 个 T1/T2 案例被误判为 T3/T4，13 个 T3/T4 案例被误判为 T1/T2。这主要是因为其指令遵循能力较弱，常常默认采用全局前提与结论匹配。

虽然通过完美翻译可以保证 100% 的分类准确率，但在实践中，我们无法确保自然语言到一阶逻辑翻译毫无瑕疵。因此，MATP 无法确保其在现实应用中的绝对正确性。尽管如此，其准确率仍显著高于基线方法。我们还记录了所有工具执行每项任务的平均处理时间。GPT-4o 每项任务约需 6 秒，DeepSeek-R1 慢得多，需要 96 秒，而 MATP 的速度相近，每项任务约需 9 秒。

研究问题2的答案在自然语言到一阶逻辑翻译准确的情况下，MATP 能够实现完美分类，明显优于基于提示的基线方法。

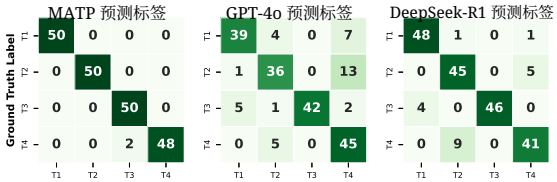


**图4：MATP 与基线方法的推理链分类性能热力图。**

## 4.4 研究问题3：模型推理分析

我们利用 MATP 生成的推理链分类结果，评估不同大语言模型在各项任务中的表现，重点关注推理质量和特定模型的模式。首先，我们比较模型在不同数据集上的表现，找出其优势、局限和推理倾向。接着，我们分析各类别中的结构模式，以揭示常见行为和失败模式。最后，我们考察 MATP 无法评估的案例，并总结其典型原因。这项分析揭示了推理中的关键瓶颈，并为改进大语言模型推理提供了方向。

*4.4.1* 模型性能分析。总体类别分布如图5所示。错误部分表示 MATP 未能进行评估的案例。MATP 分析案例在 PrOntoQA-OOD、ProofWriter和FOLIO上的平均失败率分别为 1.04%、1.22% 和 31.92%。FOLIO 上明显较高的失败率表明，MATP 对复杂的自然语言结构更为敏感，这主要是由于自然语言到一阶逻辑翻译面临挑战。具体失败原因将在§ 4.4.3中讨论。这些失败案例不纳入后续分类分析。

在 PrOntoQA-OOD 这类简单数据集上，推理模型与通用模型的答案准确率相近，但实际推理质量存在显著差异。推理模型（例如 DeepSeek-R1、QwQ-32B）始终能生成有效的推理链（T1–T2）；其中，DeepSeek-R1 在 PrOntoQA-OOD 上的 T1 比例超过 96%。相比之下，通用模型（例如 LLaMA3.1–8B）往往依赖浅层或不相容的逻辑，即使准确率很高，也缺乏有效的证明路径。通用模型还存在冗余逻辑（LLaMA）或过度使用无关前提（GPT-3.5）等问题，因此给出的答案虽然正确，却缺乏有力论证。即使是推理模型，在 ProofWriter 这类具有挑战性的数据集上也难以应对，仍会出现不完整或无效的推理链（T3/T4）。在 FOLIO 上，所有模型的 T2 和 T4 占比均有所增加。这部分源于评估工具难以处理真正的自然语言，另一部分则是因为该数据集依赖显式前提中未必包含的背景知识。模型规模也会影响性能。与较小的模型相比，LLaMA3.1–70B 等较大的模型通常具有更高的 T1 比例和更少的错误，但在逻辑连贯性方面仍落后于专门针对推理优化的模型。

总而言之，仅凭答案准确率无法反映推理能力的强弱。我们的分类策略揭示了逻辑有效性和稳健性方面的关键差异，凸显了评估推理过程而非仅评估最终答案的必要性。

*4.4.2* 推理链类别分析。我们进一步分析各个类别，以识别常见模式，并在表2中提供代表性示例。

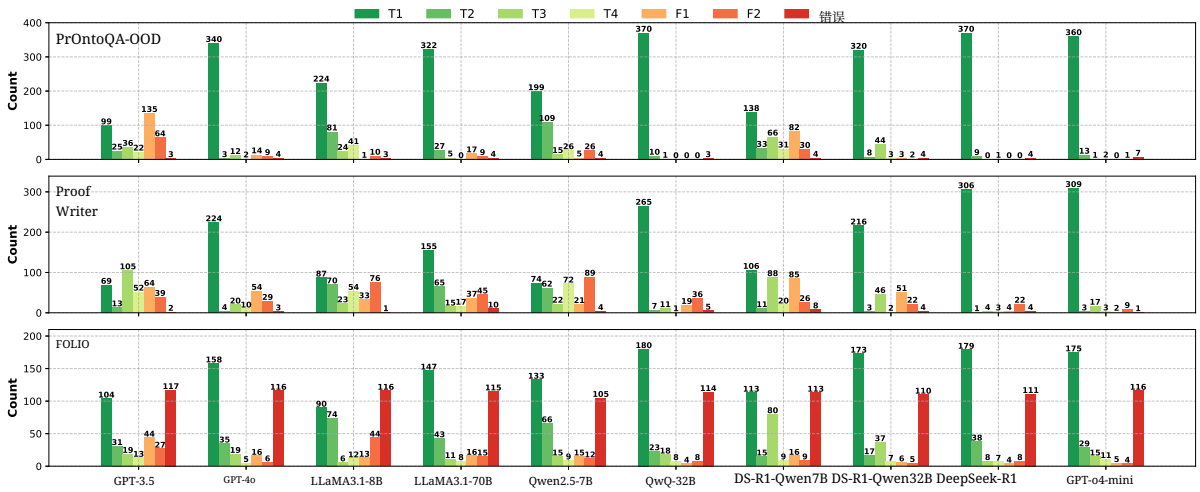


**图5：基准测试集数据中大语言模型生成的推理链类别分布** **ts.**

**表2：推理链分类中的特殊情况及代表性示例。在推理回答中，标为橙色的句子被视为 *Unknown* 相对于任务前提，而黑色句子则 *True*。**

| 描述 | 推理回答 | 任务名称 |
|---|---|---|
| **T2：** 如果第 *Unknown* 步被排除，这个回答就是一个标准的回应。 | **GPT-4o：** 1）老虎吃狗。2）如果有人吃狗，那么狗就需要熊。3）狗吃狗。4) 狗需要熊。5) 有证据可以得出狗需要熊。 | **任务 1：** ProofWriter-RelNeg-OWA-D5-226-Q3 |
| **T3：** 需要证明的结论是“Fiona不高大”。然而，模型正在证明“Anne很高大”。 | **GPT-3.5：** ... 5）如果Anne是白色的，那么Anne就很高大。6）有证据可以得出结论：Anne很高大。 | **任务 2：** ProofWriter-AttNeg-OWA-D5-1003-Q5 |
| **T4：** 模型引入了 *Unknown* 信息以捏造构造一条有效证明路径。 | **DS-R1-Qwen7B：** 1）Sally 是一只 dumpus。2）Sally 是一只 gorpus。3）每只 gorpus 都是 zumpus。4）因此，Sally 是一只 zumpus。5）每只 zumpus 都是 romrus。6）因此，Sally 是一只 romrus。 | **任务 3：** PrOntoQA-OOD-4跳-组合-随机-无相邻项-69-4 |
| **F1 值：** 模型的推理过程包含足够的信息来推导结论，然而它错误地回答说它做不到。 | **QwQ-32B:**1）Fiona 很安静。2）安静的事物都是绿色的。3）Fiona 必定是绿色的。4）Fiona 很粗糙。.....9）因此，Fiona 不是绿色的。 | **任务 4：** ProofWriter-AttNeg-OWA-D5-630-Q4 |
| **F1 值：** 模型的推理步骤完全正确，但最终答案错误。 | **GPT-4o：** 1) Bob 很安静。2) Bob 很聪明。3) 所有聪明的事物都是绿色的。4) Bob 是绿色的。5) 绿色的事物都是白色的。6) Bob 是白色的。7) 有证据可以得出结论：Bob 是白色的。 | **任务 5：** ProofWriter-AttNeg-OWA-D5-327-Q5 |
| **F1 值：** 模型在推理过程中陷入循环，导致推理过程停滞。 | **LLaMA3.1-8B：** 1）每个 lorpus 都是一个 numpus。2）每每个 lorpus 都是一个 numpus。.....20）每个 lorpus 都是一个 numpus。..... | **任务 6：** PrOntoQA-OOD-4跳-组合-随机-无形容词-22-7 |

**T1** 推理链在步骤级逻辑和整体一致性方面都完全正确，代表了理想的输出。尽管如此，即使满足我们的分类标准，仍存在一些值得进一步讨论的小问题，例如冗余步骤——虽然正确但无关紧要，以及被跳过的步骤——某一步依赖于多个未陈述前提。GPT-4o 在图1中的输出展示了这类情况（步骤1和步骤4）。

**T2** 类回答与 T1 类似，但包含一个或多个不会影响最终结论的错误步骤。删除这些步骤后，剩余步骤便构成完全正确且有效的证明（即任务1）。

**T3** 链由正确的步骤组成，但无法形成有效的证明路径。这通常源于隐式推理，即遗漏了关键前提；或源于偶然正确，即证明了一个无关的结论，而该结论碰巧具有相同的真值（即任务 2）。

模型不确定时往往会默认回答 *False*，这也可能导致答案碰巧正确。

**T4** 类回答存在与 T3 类似的问题，但还包含错误的推理步骤。这些输出通常杂乱无章、缺乏连贯性。有些情况甚至会引入不存在的事实来编造看似合理的推理链（即任务3）。

**F1** 类和**F2** 类回答都会得出错误的最终预测答案。F1 中的推理链仅包含正确步骤，而 F2 中则至少包含一个错误步骤。常见问题包括由于前提缺失或整合不当而导致推理不完整（即任务4）。我们还观察到一些特殊的失败模式：模型可能得出正确结论，却给出错误答案（即任务5）；也可能陷入重复循环，导致推理过程停滞（即任务6）。

**表 3： MATP 对 DeepSeek-R1 在 folio-train-57-168 上回答的误分类。红色 表示 MATP的错误输出，而绿色的 则标示预期的正确答案。**

| 任务信息和推理回答 | GPT-4o 生成的 Step-FOL | MATP 输出 |
|---|---|---|
| **结论：** Leo 是动物。（真）<br>**筛选后的推理步骤：**<br>1) 所有宠物都是动物。<br>2) Charlie 有一只名叫 Leo 的调皮宠物狗。<br>3) 宠物狗是一种宠物。<br>4) 由于所有宠物都是动物，Leo 是宠物，这意味着 Leo 是动物。 | 1) $\forall x(pet(x) \rightarrow animal(x))$<br>2) $pet(leo) \wedge dog(leo) \wedge naughty(leo) \wedge haspet(charlie, leo)$<br>3) $\forall x(dog(x) \rightarrow pet(x))$<br>3) $\forall x(pet(x) \wedge dog(x) \rightarrow pet(x))$<br>4) $\forall x((pet(x) \wedge animal(x)) \rightarrow (pet(leo) \rightarrow animal(leo)))$ | S1：真<br>S2：真<br>S3：未知<br>S3：真<br>S4：真<br>类别：T2<br>类别：T1 |

总之，T1 和 T2 的推理最为可靠，而 T3 和 T4 则不太稳定。F1 有改进潜力，F2 的问题最多。总体推理质量排序如下：**T1 > T2 > T3 > T4 > F1 > F2**。人工检查还发现了少数分类错误。如表3所示，某一步中的自然语言到一阶逻辑翻译错误导致 T1 被误标为 T2。虽然分类并非完全准确，但它能可靠地捕捉总体推理模式并指出潜在的错误步骤；与基于提示的基线方法相比，它具有更清晰的一致性和可解释性。

*4.4.3* 错误案例分析。 大多数评估失败源于自然语言到一阶逻辑翻译错误。表4 展示了具有代表性的案例。尽管对 PrOntoQA-OOD 和 ProofWriter 的翻译总体准确，但仍偶有问题，例如量词辖域错误（即案例 ❶），或将“或”误解为合取

**表4：自然语言到一阶逻辑翻译中的典型错误案例。错误部分以红色突出显示。**

| 描述 | 自然语言句子 | 一阶逻辑表示 | 数据集 |
|---|---|---|---|
| 一阶逻辑中的全称量词是错误的。 | 如果某人聪明且不善良，那么他们年轻。 | $\forall x(smart(x) \wedge \neg kind(x)) \rightarrow young(x)$ | 案例❶ 来自 ProofWriter |
| 错误地使用 ∧ 而非将 ∨用于"或"。 | Alex 是一个 gorpus，一个 sterpus，一个 numpus，或一个 gorpus. | $Gorpus(alex) \wedge Sterpus(alex) \wedge Numpus(alex) \wedge Gorpus(alex)$ | 案例❷ 来自 PrOntoQA-OOD |
| $BoyBand(btob4u)$，适用于推理候选结论被省略。 | 《Show Your Love》是由韩国南部记载的-人男子组合BtoB 4U。 | $Song(showyourlove) \wedge RecordedBy(showyourlove, btob4u)$ | 案例❸ 来自 FOLIO |
| 谓词使用不当——地，同一个词"裁判观察员"在上下文中由不同的谓词表示。 | Brian Winter 被指派为一名裁判观察员服务器在他退役后。一些足球裁判成为裁判观察。 | $AppointedObserver(brian) \wedge \exists x(FootballReferee(x) \wedge RefereeObserver(x))$ | 案例❹ 来自 FOLIO |
| "要么……要么……"表示异或，这与或不同，并且应使用以下符号表示 ⊕. | 显示的所有内容集合是或者一个植物或动物。 | $\forall x(Displayed(x) \rightarrow (Plant(x) \vee Animal(x)))$ | 情况❺ 来自 FOLIO |
| 直接使用 > 是不符合语法，应表达为 $MoreRevenue(airbus, boeing)$。 | 空中客车制造了更多修订版-收入比波音去年。 | $Revenue(airbus, lastyear) > Revenue(boeing, lastyear)$ | 案例❻ 来自 FOLIO |

（在连接多个实体时，即案例❷）。相比之下，由于自然语言丰富多样，FOLIO 引入了更多样且复杂的错误。我们总结了以下常见错误模式。

**信息遗漏**。当句子包含多个事实时，关键细节会被遗漏（案例❸）。

**谓词使用不当**。同一概念可能被映射到不同的谓词，导致语义不一致（即案例❹）。**异或与或混淆**。涉及排他性选择的表达式（如"要么……要么……"）常被误表示为析取（∨），而非异或（⊕），即使提示中明确要求如此（即案例❺）。

**处理领域特定表达式不充分**。涉及比较级、最高级、数学或时间的领域特定表达式往往处理不当，导致语法不通——

数学 FOL（即，案例❻）。

**标注错误**。除了上述提取问题外，我们还在人工检查期间观察到 FOLIO 数据集中存在标注错误。图 6 展示了一个此类错误标注的示例。我们发现，任务中第四个前提的 FOL 真实标注表示被错误地写为"$\forall x((Professional(x) \wedge HighWinRatio(x)) \rightarrow Big3(x))$"，这可能是导致标注错误的原因。在 folio- train-63-186、167-480、167-479、302-746、470-1359 和 470-1360 等任务中也发现了类似问题。

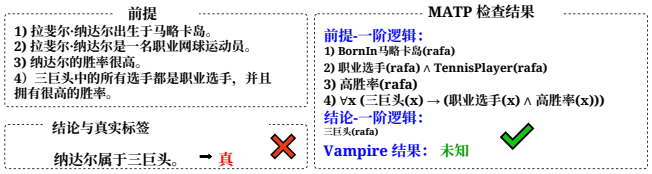


**图 6：FOLIO 中标注错误的示例。任务 ID 为 folio-train-34-99。**

研究问题3的答案

MATP 揭示了不同大语言模型在推理行为上的明显差异。尽管答案准确率相近，推理模型生成的逻辑比通用模型更连贯，这凸显了评估推理质量时不能只关注最终答案。

## 4.5 消融研究

我们进行了消融研究，比较了自然语言到一阶逻辑的再生成过程中有无反馈时 MATP 的表现。

**设置**。如图5所示，大多数分析失败都发生在FOLIO上。因此，我们评估了两个模型 GPT-4o 和 DeepSeek-R1 在FOLIO的 355 个任务上的回答。由于难以确定具体错误，我们将此前生成的全部内容作为反馈。我们在图3 中的原始提示下添加了一个反馈字段，位于输入实例下方，其中包含先前存在转换错误的自然语言到一阶逻辑翻译。该字段有助于模型在重新生成时避免类似错误。同样，重新生成的最大尝试次数设为三次。我们重点关注有反馈和无反馈两种设置下的失败次数。

**结果**。如表5所示，使用反馈的效果很小，有时无反馈设置反而能取得更好的结果。此外，反馈会增加资源消耗。因此，在 MATP 的实际实现中不采用反馈。

**表5：反馈对 MATP 失败的影响。**

| 模型 | 有反馈 | 无反馈 |
|---|---|---|
| **GPT-4o** | 118 | 116 |
| **DeepSeek-R1** | 119 | 111 |

# 5 讨论

## 5.1 未来改进

本文提出了：(1) 一种将非正式的大语言模型推理转化为形式逻辑并使用自动定理证明工具进行验证的框架；(2) 一种超越二元答案正确性的多层级分类策略。得益于 MATP 的模块化设计，我们可以独立升级各个组件，使其能够适用于需要严格推理验证的高风险场景。未来仍有若干值得探索的方向：

**增强形式化**。MATP 的准确率取决于可靠的自然语言到一阶逻辑翻译。当前基于提示的方法在处理复杂自然语言输入时仍存在局限。未来的工作可以采用更稳健或领域适配的翻译工具。

**开放域知识需求**。某些映射（例如 .g. ，"低"到"¬ 高"）需要前提中未提供的外部知识。为确保自然语言与形式语言之间严格的一一映射，同时处理隐含的常识，模型需要在翻译之前或过程中自主引入必要的知识（例如，"低"等于"不高"）。

**改进评估指标**。对提取的一阶逻辑进行评估通常依赖黄金标准标注和基于相似度的指标，而这些指标往往无法反映语义忠实度和逻辑有效性 [47]。这凸显了对更具原则性且自动化的评估方法的需求。

**重构推理图**。未来的工作可以将推理步骤建模为节点，将自动定理证明器检测到的推理建模为有向边，从大语言模型生成的推理输出中构建推理图。该结构支持细粒度验证，并有助于识别超越简单路径存在性检查的冗余、缺失或冲突逻辑。

**扩展推理支持**。除正向蕴涵外，支持逆向推理也能扩大适用范围。

## 5.2 响应分析优化

为提高效率并减少人工工作量——尤其是在自然语言到一阶逻辑评估中，我们利用大语言模型的内部特征，使用熵和詹森-香农（JS）散度 [30]来识别不确定或可能存在缺陷的推理步骤。如图7所示，在ProofWriter数据集上评估 Llama3.1-8B 系列模型时，有缺陷的推理通常与较高的熵或 JS 散度相关。请注意，涵盖不同数据集和大语言模型的完整图表见 [61]。

基于此，我们采用基于阈值的过滤策略：将熵低于 7.0 且 JS 散度低于 0.065 的推理输出视为可能正确，从而将分析重点放在不确定性更高的案例上。这一步预过滤有效减轻了评估负担；在不同数据集和模型上，正确推理的覆盖率为 14.7% 至 40.5%。尽管有些错误回答被误过滤掉（例如，Llama3.1-8B 在 ProofWriter 和 FOLIO 上分别为 8.5% 和 6.7%），但其他配置的假阴性率接近零，验证了该方法的可靠性。总体而言，该方法通过优先处理不确定性较高的推理步骤，提高了评估效率。未来的工作将探索更紧密地整合大语言模型的内部信号，以实现自动过滤和半监督推理质量评估。

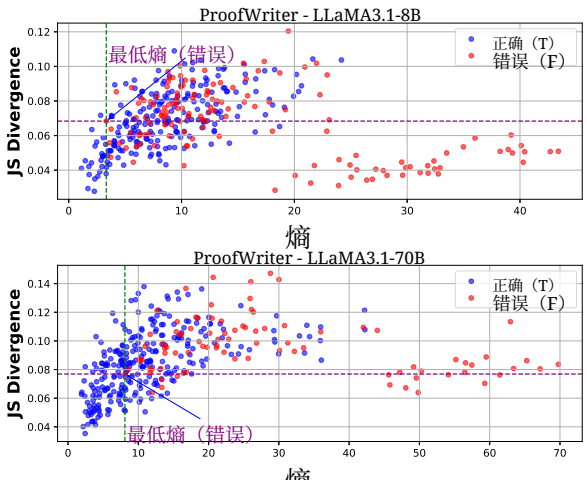


**图 7：ProofWriter 上 Llama3.1 系列模型正确（T）与错误（F）推理回答的熵和 Jensen–Shannon 散度分布对比。**

## 6 相关工作

**自动生成形式化规范。**大语言模型日益广泛地用于自动生成形式化规范，以及将自然语言转换为逻辑表达式。SpecGen [29] 采用基于对话和变异的方法生成程序规范，无需依赖模板。NL2CTL [60]通过提示工程构建 NL-CTL 数据集，并微调 T5 模型，将需求转换为计算树逻辑（CTL）。SAFE [5] 面向 Rust，结合数据合成、微调和符号验证，以自动生成证明，准确率超过 70%。NL2FOL [22] 将自然语言转换为一阶逻辑，并使用 SMT 求解器进行谬误检测和解释。NL2SPEC [7] 支持交互式地将文本转换为时序逻辑，并允许通过迭代优化来消除歧义。

**增强大语言模型的推理能力。**增强大语言模型的推理能力对于高风险任务至关重要。在模型内部，DeepSeek- R1 [14] 结合监督微调和强化学习，使模型具备思维链和自验证等推理能力。在模型外部，Logic-of-Thought [28] 和 Chain-of-Knowledge [49] 等提示策略通过向提示中注入结构化逻辑或知识，提升逻辑完备性和事实连贯性。工具增强方法将大语言模型用作语义解析器，把自然语言转换为形式语言，再由外部符号工具处理，从而提高推理准确率。LINC [32] 将自然语言转换为一阶逻辑，以进行逻辑定理证明。SatLM [56] 将自然语言转换为声明式任务规范，并针对数学和逻辑推理任务使用 Z3 SMT求解器 [9],。Logic-LM [34] 涵盖四类形式语言及其对应工具，并针对不同类型的推理任务选择合适的工具。我们的工作采用了类似策略，但重点在于评估大语言模型对推理任务的回答是否正确且符合逻辑一致性。

**评估大语言模型的推理能力。**评估大语言模型的推理能力已成为研究重点。以往研究在多种任务中评估模型表现，例如在函数 [27] 和软件工程 [45] 层面评估代码理解的正确性，以及在真实世界推理场景中进行事实性幻觉检测 [24]。随着推理能力不断进步，评估目标已从整体任务正确性扩展到逐步推理分析。综述 [23] 总结了评估推理步骤的四个关键维度：事实性、有效性、连贯性和实用性。尽管已有进展，大多数方法仍依赖劳动密集型的黄金标准推理链 [12, 38] ，或基于大语言模型的概率评估 [52, 58], ，这两者都会削弱逻辑严谨性。ROSCOE [12] 提出了若干度量指标，通过基于嵌入的计算，评估推理步骤、参考步骤与问题之间的关系。MR-GSM8K [58] 通过提示大语言模型检查推理的正确性，并输出解答正确性、首个错误步骤和错误分析等结构化字段。ReasonEval [52]利用高质量标注数据微调大语言模型，为每个推理步骤的有效性和冗余性打分。我们的工作将自动定理证明器引入这一研究方向，以评估大语言模型生成的推理步骤。

## 7 结论

本文介绍了 MATP，这是一种评估框架，通过自动定理证明弥合了非正式大语言模型推理与形式化验证之间的差距。MATP 将自然语言推理系统地转换为一阶逻辑，并在每一步进行严格验证，从而以前所未有的方式揭示大语言模型生成的推理链的逻辑有效性。我们的广泛评估表明，MATP 不仅在一阶逻辑翻译和推理验证方面达到较高准确率，还能进行细粒度分类，区分真正的演绎推理与肤浅的模式匹配或偶然正确。模型之间显著的差异凸显了在高风险应用中进行形式化验证的关键必要性。尽管处理复杂语言结构仍面临挑战，MATP 仍为确保安全关键领域中的 AI 推理可靠、可验证奠定了基础；在这些领域，基于逻辑健全性的可信度至关重要。

# 参考文献


[1] Josh Achiam、Steven Adler、Sandhini Agarwal、Lama Ahmad、Ilge Akkaya、Florencia Leoni Aleman、Diogo Almeida、Janko Altenschmidt、Sam Altman、Shyamal Anadkat 等。2023年。GPT-4 技术报告。arXiv 预印本 arXiv:2303.08774（2023年）。[2] Janice Ahn、Rishu Verma、Renze Lou、Di Liu、Rui Zhang 和 Wenpeng Yin。2024年。用于数学推理的大语言模型：进展与挑战。载于第 18 届计算语言学协会欧洲分会会议论文集：学生研究研讨会，Neele Falk、Sara Papi 和 Mike Zhang（编）。计算语言学协会，马耳他圣朱利安，225–237。https://aclanthology.org/2024.eacl-srw.17/[3] Isabelle Augenstein、Timothy Baldwin、Meeyoung Cha、Tanmoy Chakraborty、Giovanni Luca Ciampaglia、David Corney、Renee DiResta、Emilio Ferrara、Scott Hale、Alon Halevy、Eduard Hovy、Heng Ji、Filippo Menczer、Ruben Miguez、Preslav Nakov、Dietram Scheufele、Shivam Sharma 和 Giovanni Zagni。2024年。大语言模型时代的事实性挑战与事实核查机遇。《自然·机器智能》6，8（2024年），852–863。https://doi.org/10.1038/s42256- 024-00881-z[4] Shiqi Chen、Yiran Zhao、Jinghan Zhang、I-Chun Chern、Siyang Gao、Pengfei Liu 和 Junxian He。2023年。FEL M：大语言模型事实性评估基准测试集。载于神经信息处理系统第三十七届会议数据集与基准测试集专题。https://openreview.net/forum?id=jSO7Vgolc6[5] Tianyu Chen、Shuai Lu、Shan Lu、Yeyun Gong、Chenyuan Yang、Xuheng Li、Md Rakib Hossain Misu、Hao Yu、Nan Duan、Peng Cheng 等。2024年。通过自我进化为 Rust 代码自动生成证明。arXiv 预印本 arXiv:2410.15756（2024年）。[6] Jan Clusmann、Fiona R. Kolbinger、Hannah Sophie Muti、Zunamys I. Carrero、Jan-Niklas Eckardt、Narmin Ghaffari Laleh、Chiara Maria Lavinia Löffler、Sophie-Caroline Schwarzkopf、Michaela Unger、Gregory P. Veldhuizen、Sophia J. Wagner 和 Jakob Nikolas Kather。2023年。医学领域大语言模型的未来图景。《通讯·医学》3，1（2023年），141。https://doi.org/10.1038/s43856-023-00370-1[7] Matthias Cosler、Christopher Hahn、Daniel Mendoza、Frederik Schmitt 和 Caroline Trippel。2023年。NL2SPEC：利用大语言模型将非结构化自然语言交互式地转换为时序逻辑。载于计算机辅助验证国际会议。Springer，383–396。[8] Yu Cui 和 Cong Zuo。2025年。针对推理型大语言模型的实用推理中断攻击。arXiv 预印本 arXiv:2505.06643（2025年）。[9] Leonardo De Moura 和 Nikolaj Bjørner。2008年。Z3：一种高效的 SMT 求解器。载于系统构建与分析工具和算法国际会议。Springer，337–340。[10] Eva Eigner 和 Thorsten Händler。2024年。大语言模型辅助决策的决定因素。arXiv 预印本 arXiv:2402.17385（2024年）。[11] Esther Gan、Yiran Zhao、Liying Cheng、Mao Yancan、Anirudh Goyal、Kenji Kawaguchi、Min-Yen Kan 和 Michael Shieh。2024年。大语言模型对抗性排版错误的推理鲁棒性。载于2024年自然语言处理实证方法会议论文集，Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen（编）。计算语言学协会，美国佛罗里达州迈阿密，10449–10459。https://doi.org/10.18653/v1/2024.emnlp-main.584[12] Olga Golovneva、Moya Chen、Spencer Poff、Martin Corredor、Luke Zettlemoyer、Maryam Fazel-Zarandi 和 Asli Celikyilmaz。2022年。ROSCOE：用于逐步推理评分的一套度量指标。arXiv 预印本 arXiv:2212.07919（2022年）。[13] Aaron Grattafiori、Abhimanyu Dubey、Abhinav Jauhri、Abhinav Pandey、Abhishek Kadian、Ahmad Al-Dahle、Aiesha Letman、Akhil Mathur、Alan Schelten、Alex Vaughan 等。2024年。Llama 3 模型系列。arXiv 预印本 arXiv:2407.21783（2024年）。[14] Daya Guo、Dejian Yang、Haowei Zhang、Junxiao Song、Ruoyu Zhang、Runxin Xu、Qihao Zhu、Shirong Ma、Peiyi Wang、Xiao Bi 等。2025年。DeepSeek-R1：通过强化学习激发大语言模型的推理能力。arXiv 预印本 arXiv:2501.12948（2025年）。[15] Simeng Han、Hailey Schoelkopf、Yilun Zhao、Zhenting Qi、Martin Riddell、Wenfei Zhou、James Coady、David Peng、Yujie Qiao、Luke Benson、Lucy Sun、Alex Wardle-Solano、Hannah Szabo、Ekaterina Zubova、Matthew Burtell、Jonathan Fan、Yixin Liu、Brian Wong、Malcolm Sailor、Ansong Ni、Linyong Nan、Jungo Kasai、Tao Yu、Rui Zhang、Alexander R. Fabbri、Wojciech Kryscinski、Semih Yavuz、Ye Liu、Xi Victoria Lin、Shafiq Joty、Yingbo Zhou、Caiming Xiong、Rex Ying、Arman Cohan 和 Dragomir Radev。2024年。FOLIO：基于一阶逻辑的自然语言推理。arXiv:2209.00840 [cs.CL] https://arxiv.org/abs/2209.00840[16] Sirui Hong、Mingchen Zhuge、Jonathan Chen、Xiawu Zheng、Yuheng Cheng、Jinlin Wang、Ceyao Zhang、Zili Wang、Steven Ka Shing Yau、Zijuan Lin、Liyang Zhou、Chenyu Ran、Lingfeng Xiao、Chenglin Wu 和 Jürgen Schmidhuber。2024年。MetaGPT：多智能体协作框架的元编程。载于第十二届国际学习表征会议。https://openreview.net/forum?id=VtmBAGCN7o

[17] Jie Huang 和 Kevin Chen-Chuan Chang。2023年。《迈向大语言模型推理：一项综述》。载于《计算语言学协会论文集：ACL 2023 论文集》，Anna Rogers、Jordan Boyd-Graber 和 Naoaki Okazaki（编）。计算语言学协会，加拿大多伦多，1049–1065。https://doi.org/10.18653/v1/2023.findings-acl.67[18] Aaron Hurst、Adam Lerer、Adam P Goucher、Adam Perelman、Aditya Ramesh、Aidan Clark、AJ Ostrow、Akila Welihinda、Alan Hayes、Alec Radford 等。2024年。GPT-4o 系统卡。arXiv 预印本 arXiv:2410.21276（2024年）。[19] Laura Kovács 和 Andrei Voronkov。2013年。一阶定理证明与 Vampire。载于《计算机辅助验证》，Natasha Sharygina 和 Helmut Veith（编）。Springer Berlin Heidelberg，德国柏林，1–35。[20] Satyapriya Krishna、Jiaqi Ma、Dylan Z Slack、Asma Ghandeharioun、Sameer Singh 和 Himabindu Lakkaraju。2023年。语言模型的事后解释可以改进语言模型。载于《神经信息处理系统第三十七届会议》。https://openreview.net/forum?id=3H37XciUEv [21] Jinqi Lai、Wensheng Gan、Jiayang Wu、Zhenlian Qi 和 Philip S. Yu。2024年。法律领域的大语言模型：综述。AI Open 5（2024年），181–196。https://doi.org/10. 1016/j.aiopen.2024.09.002[22] Abhinav Lalwani、Lovish Chopra、Christopher Hahn、Caroline Trippel、Zhijing Jin 和 Mrinmaya Sachan。2024年。NL2FOL：将自然语言转换为一阶逻辑以检测逻辑谬误。arXiv 预印本 arXiv:2405.02318（2024年）。[23] Jinu Lee 和 Julia Hockenmaier。2025年。评估逐步推理轨迹：综述。arXiv 预印本 ar Xiv:2502.12289（2025年）。[24] Ningke Li、Yuekang Li、Yi Liu、Ling Shi、Kailong Wang 和 Haoyu Wang。2024年。Drowzee：用于检测大语言模型中事实冲突幻觉的蜕变测试。《ACM程序设计语言汇刊》8，OOPSLA2（2024年），1843–1872。[25] Yanda Li、Dixuan Wang、Jiaqing Liang、Guochao Jiang、Qianyu He、Yanghua Xiao 和 Deqing Yang。2024年。从谬误中推理：通过理解逻辑谬误增强大语言模型的逻辑推理。载于《计算语言学协会论文集：NAACL 2024 论文集》，Kevin Duh、Helena Gomez 和 Steven Bethard（编）。计算语言学协会，墨西哥墨西哥城，3053–3066。https://doi.org/10.18653/v1/2024.findings- naacl.192[26] Yinheng Li、Shaofei Wang、Han Ding 和 Hang Chen。2023年。金融领域的大语言模型：综述。载于《第四届ACM人工智能金融国际会议论文集》（美国纽约州布鲁克林）（ICAIF '23）。计算机协会，美国纽约州纽约市，374–382。https://doi.org/10.1145/3604237.3626869[27] Ziyu Li 和 Donghwan Shin。2024年。用于评估大语言模型代码理解能力的基于变异的一致性测试。载于《IEEE/ACM第三届人工智能工程——人工智能软件工程国际会议论文集》。150–159。[28] Tongxuan Liu、Wenjiang Xu、Weizhe Huang、Yuting Zeng、Jiaxing Wang、Xingyu Wang、Hailong Yang 和 Jing Li。2024年。Logic-of-thought：将逻辑注入上下文，以实现大语言模型的完整推理。arXiv 预印本 arXiv:2409.17539（2024年）。[29] Lezhi Ma、Shangqing Liu、Yi Li、Xiaofei Xie 和 Lei Bu。2024年。SpecGen：通过大语言模型自动生成形式化程序规范。arXiv 预印本 arXiv:2401.08807（2024年）。[30] M.L. Menéndez、J.A. Pardo、L. Pardo 和 M.C. Pardo。1997年。Jensen-Shannon散度。《富兰克林研究所学报》334，2（1997年），307–318。https: //doi.org/ 10 1016 0016 0032 96 00063 4 . /S - ( ) - Niels Mündler、Jingxuan He、[31] Slobodan Jenko 和 Martin Vechev。2024年。大语言模型的自相矛盾幻觉：评估、检测与缓解。载于《第十二届国际学习表征会议》。https://openreview.net/forum?id=EmQSOi1X2f [32] Theo Olausson、Alex Gu、Ben Lipkin、Cedegao Zhang、Armando Solar-Lezama、Joshua Tenenbaum 和 Roger Levy。2023年。LINC：结合语言模型与一阶逻辑证明器的神经符号逻辑推理方法。载于《2023年自然语言处理实证方法会议论文集》，Houda Bouamor、Juan Pino 和 Kalika Bali（编）。计算语言学协会，新加坡，5153–5176。https://doi.org/10. 18653 1 2023 313 / v / .emnlp-main. OpenAI。。推出 OpenAI o3 和 o4-mini。[33] 2025 https://openai.com/index/ introducing-o3-and-o4-mini/。访问日期：2025-07-13。[34] Liangming Pan、Alon Albalak、Xinyi Wang 和 William Yang Wang。2023年。Logic-lm：借助符号求解器赋能大语言模型，实现可靠的逻辑推理。arXiv 预印本 arXiv:2305.12295（2023年）。[35] Amirreza Payandeh、Dan Pluth、Jordan Hosier、Xuesu Xiao 和 Vijay K. Gurbani。2024年。大语言模型容易受到逻辑谬误影响吗？载于《2024年计算语言学、语言资源与评估联合国际会议论文集（LREC-COLING 2024）》，Nicoletta Calzolari、Min-Yen Kan、Veronique Hoste、Alessandro Lenci、Sakriani Sakti 和 Nianwen Xue（编）。ELRA 和 ICCL，意大利都灵，8276–8286。https://aclanthology.org/2024.lrec-main.726/[36] Shuofei Qiao、Yixin Ou、Ningyu Zhang、Xiang Chen、Yunzhi Yao、Shumin Deng、Chuanqi Tan、Fei Huang 和 Huajun Chen。2023年。通过语言模型提示进行推理：综述。载于第61届年度会议论文集

计算语言学协会（第1卷：长篇论文），Anna Rogers、Jordan Boyd-Graber 和 Naoaki Okazaki（编）。计算语言学协会，加拿大多伦多，5368–5393。https://doi.org/10.18653/v1/2023.acl- long.[37] Qwen团队。2025年。QwQ-32B：拥抱强化学习的力量。https://qwenlm.github.io/blog/qwq-32b/。访问日期：2025-07-13。[38] Abulhair Saparov 和 He He。2023年。语言模型是贪婪推理者：对思维链进行系统的形式化分析。载于《第十一届国际学习表征会议》。https://openreview.net/forum?id=qFVVBzXxR2V[39] Abulhair Saparov、Richard Yuanzhe Pang、Vishakh Padmakumar、Nitish Joshi、Seyed Mehran Kazemi、Najoung Kim 和 He He。2023年。使用分布外考试样例测试大语言模型的一般演绎推理能力。CoRR abs/2305.15269（2023年）。https://doi.org/10.48550/arXiv.2305.15269 arXiv:2305.15269[40] Sergio Servantez、Joe Barrow、Kristian Hammond 和 Rajiv Jain。2024年。逻辑链：基于规则的大语言模型推理。载于《计算语言学协会论文集：ACL 2024 论文集》，Lun-Wei Ku、Andre Martins 和 Vivek Srikumar（编）。计算语言学协会，泰国曼谷，2721 2733 10 18653 1 2024 159 – 。https://doi.org/ . /v / .findings-acl. Parshin Shojaee、Iman Mirzadeh、Keivan Alizadeh、Maxwell Horton、Samy Bengio 和 Mehrdad Farajtabar。2025年。思考的幻象：从问题复杂度视角理解推理模型的优势与局限。https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf[42] Geoff Sutcliffe。2017年。TPTP问题库及其配套基础设施。《自动推理杂志》59，4（2017年12月），483–502。https://doi.org/10.1007/s10817-017-9407-7[43] Oyvind Tafjord、Bhavana Dalvi Mishra 和 Peter Clark。2021年。ProofWriter：针对自然语言生成蕴含、证明和溯因陈述。arXiv:2012.13048 [cs.CL] https://arxiv.org/abs/2012.13048[44] Derek Tam、Anisha Mascarenhas、Shiyue Zhang、Sarah Kwan、Mohit Bansal 和 Colin Raffel。2023年。通过新闻摘要评估大语言模型的事实一致性。载于《计算语言学协会论文集：ACL 2023 论文集》，Anna Rogers、Jordan Boyd-Graber 和 Naoaki Okazaki（编）。计算语言学协会，加拿大多伦多，5220–5255。https://doi.org/ 10 18653 1 2023 322 . /v / .findings-acl. Minaoar Hossain Tanzil、Junaed Younus Khan 和 Gias Uddin。2024年。软件评审中的 ChatGPT 错误检测。载于《IEEE/ACM第46届国际软件工程会议论文集》。1–12。[46] Qwen团队。2024年。Qwen2.5：基础模型家族。https://qwenlm.github.io/blog/qwen2.5/。访问日期：2025-07-13。[47] Ramya Keerthy Thatikonda、Wray Buntine 和 Ehsan Shareghi。2025年。评估一阶逻辑接近度指标与人类判断的一致性。arXiv 预印本 arXiv:2501.08613（2025年）。[48] Zhongwei Wan、Xin Wang、Che Liu、Samiul Alam、Yu Zheng、Jiachen Liu、Zhong- nan Qu、Shen Yan、Yi Zhu、Quanlu Zhang、Mosharaf Chowdhury 和 Mi Zhang。2024年。高效大语言模型：综述。《机器学习研究汇刊》（2024年）。https://openreview.net/forum?id=bsCCJHbO8A 综述认证。[49] Jianing Wang、Qiushi Sun、Xiang Li 和 Ming Gao。2024年。通过知识链提示提升语言模型推理能力。载于《计算语言学协会第62届年会论文集》（第1卷：长篇论文），Lun-Wei Ku、Andre Martins 和 Vivek Srikumar（编）。计算语言学协会，泰国曼谷，4958–4981。https://doi.org/ 10 18653 1 2024 271 . /v / .acl-long. Xuezhi Wang、Jason Wei、Dale Schuurmans、Quoc V Le、Ed H. Chi、Sharan Narang、Aakanksha Chowdhery 和 Denny Zhou。2023年。自洽性提升语言模型中的思维链推理。载于《第十一届国际学习表征会议》。https://openreview.net/forum? id=1PL1NIMMrw[51] Junda Wu、Tong Yu、Xiang Chen、Haoliang Wang、Ryan Rossi、Sungchul Kim、Anup Rao 和 Julian McAuley。2024年。DeCoT：通过因果干预，为大语言模型知识密集型任务中的思维链去偏。载于《计算语言学协会第62届年会论文集》（第1卷：长篇论文），Lun-Wei Ku、Andre Martins 和 Vivek Srikumar（编）。计算语言学协会，泰国曼谷，14073 14087 10 18653 1 2024 758 – 。https://doi.org/ . /v / .acl-long. Shijie Xia、Xuefeng Li、Yixin Liu、Tongshuang Wu 和 Pengfei Liu。2025年。超越准确率评估数学推理。载于《AAAI人工智能会议论文集》，第39卷。27723–27730。[53] Tian Xie、Zitian Gao、Qingnan Ren、Haoming Luo、Yuqian Hong、Bryan Dai、Joey Zhou、Kai Qiu、Zhirong Wu 和 Chong Luo。2025年。Logic-RL：通过基于规则的强化学习释放大语言模型的推理能力。arXiv 预印本 arXiv:2502.14768（2025年）。[54] Sherry Yang、Ofir Nachum、Yilun Du、Jason Wei、Pieter Abbeel 和 Dale Schuurmans。2023年。用于决策的基础模型：问题、方法与机遇。arXiv 预印本 arXiv:2303.04129（2023年）。[55] Yuan Yang、Siheng Xiong、Ali Payani、Ehsan Shareghi 和 Faramarz Fekri。2023年。Harnessing the Power of Large Language Models for Natural Language to First- Order Logic Translation。*arXiv* 预印本 *arXiv:2305.15541*（2023年）。[56] Xi Ye、Qiaochu Chen、Isil Dillig 和 Greg Durrett。2023年。SatLM：使用声明式提示的可满足性辅助语言模型。载于神经信息处理系统进展，A. Oh、T. Naumann、A. Globerson、K. Saenko、M. Hardt 和 S. Levine（编），第36卷。Curran Associates, Inc.，45548–45580。https://proceedings.neurips.cc/paper_files/paper/2023/file/ 8e9c7d4a48bdac81a58f983a64aaf42b-Paper-Conference.pdf[57] Junqi Yin、Sajal Dash、Feiyi Wang 和 Mallikarjun Shankar。2023年。FORGE：面向科学的开放基础模型预训练。载于国际高性能计算、网络、存储与分析会议论文集（美国科罗拉多州丹佛）（*SC '23*）。计算机协会，美国纽约州纽约市，文章 81, 13 10 1145 3581784 3613215 页。[58] https://doi.org/./.Zhongshen Zeng、Pengguang Chen、Shu Liu、Haiyun Jiang 和 Jiaya Jia。2024年。MR-GSM8K：用于大语言模型评估的元推理基准。arXiv：2312.17080 [cs.CL] https://arxiv.org/abs/2312.17080[59] Yu Zhang、Xiusi Chen、Bowen Jin、Sheng Wang、Shuiwang Ji、Wei Wang 和 Jiawei Han。2024年。科学大语言模型及其在科学发现中的应用综述。载于*2024*年自然语言处理实证方法会议论文集，Yaser Al-Onaizan、Mohit Bansal 和 Yun-Nung Chen（编）。计算语言学协会，美国佛罗里达州迈阿密，8783–8817。https://doi.org/10.18653/v1/2024.emnlp- main.498[60] Mengyan Zhao、Ran Tao、Yanhong Huang、Jianqi Shi、Shengchao Qin 和 Yang Yang。2024年。NL2CTL：通过大语言模型自动生成形式化需求规范。载于形式化方法与软件工程，Kazuhiro Ogata、Dominique Mery、Meng Sun 和 Shaoying Liu（编）。Springer Nature Singapore，新加坡，1–17。[61] zxyhp。2025年。MATP。https://github.com/zxyhp/MATP。