

Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models

Zhi Rui Tam^{1*}, Cheng-Kuang Wu^{1*}, Yi-Lin Tsai¹, Chieh-Yen Lin¹, Hung-yi Lee², Yun-Nung Chen²

¹Appier AI Research, ²National Taiwan University

Abstract

Structured generation, the process of producing content in standardized formats like JSON and XML, is widely utilized in real-world applications to extract key output information from large language models (LLMs). This study investigates whether such constraints on generation space impact LLMs' abilities, including reasoning and domain knowledge comprehension. Specifically, we evaluate LLMs' performance when restricted to adhere to structured formats versus generating free-form responses across various common tasks. Surprisingly, we observe a significant decline in LLMs' reasoning abilities under format restrictions. Furthermore, we find that stricter format constraints generally lead to greater performance degradation in reasoning tasks. Our code and results are available online.¹

1 Introduction

The few-shot in-context learning (Brown et al., 2020) and instruction-following (Wei et al., 2021) capabilities of large language models (LLMs) have allowed them to solve downstream tasks out of the box. However, a major obstacle to incorporating LLMs into industrial applications is their lack of adherence to standardized output formats. This inconsistency complicates output parsing and undermines the reliability of these models.

One common approach to overcoming this obstacle is *structured generation*, which involves providing output in standardized formats like JSON or XML through *format restrictions*. These restrictions can be implemented in various ways, such as instructing LLMs to adhere to specified formats with *format-restricting instructions*, or using industrial solutions like JSON mode (OpenAI, 2024; Gemini, 2024), Instructor (Liu, 2024), or

Question:
Eliza's rate per hour for the first 40 hours she works each week is \$10. She also receives an overtime pay of 1.2 times her regular hourly rate. If Eliza worked for 45 hours this week, how much are her earnings for this week?

Standard Prompting

Instruct : Provide your output in the following text format:
Step by step reasoning: ...
Answer: The final answer is ...

.....
Step by step reasoning:
1. Eliza's regular rate per hour is \$10 for the first 40 hours.
2. For the 5 hours of overtime, her rate is 1.2 times her regular rate, which is $1.2 * \$10 = \12 per hour.
3. Eliza worked 40 regular hours and 5 overtime hours, totaling 45 hours.
4. Earnings for regular hours = 40 hours * \$10/hour = \$400
5. Earnings for overtime hours = 5 hours * \$12/hour = \$60
6. Total earnings for the week = \$400 (regular) + \$60 (overtime) = \$460

Answer: The final answer is 460.

Format-Restricting Prompting

Instruct : Provide your output in the following valid JSON format:

```
```json
{
 "step_by_step_reasoning": "...",
 "answer": ...
}
```
```json
{
 "step_by_step_reasoning": "Calculate the earnings for \nthe first 40 hours at $10 per hour. Then calculate the \nearnings for the additional 5 hours at $10 * 1.2 per \nhour. Add both amounts to find the total earnings for \nthe week.",
 "answer": 490
}
```



Figure 1: GPT-3.5-turbo prompted with GSM8K math questions in standard natural language answered correctly, but failed when format restrictions were applied.

Guardrails (PrefectHQ, 2024). These strategies simplify parsing workflows and streamline the integration of LLMs into real-world applications.

Due to the growing demand for structured generation, the research community has shown increased interest in investigating LLMs' format-following abilities. For example, IFEval (Zhou et al., 2023), INFOBENCH (Qin et al., 2024), and FOFO (Xia et al., 2024) focus on evaluating LLMs' instruction-following capabilities, including format adherence. However, these studies do not address a critical

\*Equal contribution

<sup>1</sup><https://github.com/appier-research/structure-gen>

# 让我自由发言？格式限制对大型语言模型性能影响的研究

祁瑞棠, 吴承光, 蔡怡琳, 林杰彦, 李宏毅, 陈云农

<sup>1</sup> Appier AI Research, 台湾大学

## 摘要

结构化生成，即以 JSON 和 XML 等标准化格式生产内容的过程，在现实世界的应用中被广泛使用，以从大型语言模型中提取关键输出信息 (LLMs)。本研究探讨了此类对生成空间的限制是否会影响LLMs的能力，包括推理和领域知识理解。具体来说，我们评估了LLMs在受限于遵循结构化格式与生成自由格式响应的各种常见任务时的表现。令人惊讶的是，我们发现格式限制下LLMs的推理能力显著下降。此外，我们发现更严格的格式限制通常会导致推理任务中的性能下降更大。我们的代码和结果可在[网上获取](#)。

## 1 引言

少量样本情境学习 (Brown 等, 2020) 和指令遵循能力 (Wei 等, 2021) 的大语言模型 (LLMs) 使它们能够直接解决下游任务。然而，将LLMs纳入工业应用的一个主要障碍是它们缺乏对标准化输出格式的遵守。这种不一致性使得输出解析复杂化，并损害了这些模型的可靠性。

一种克服这一障碍的常见方法是结构化生成，它涉及通过格式限制提供标准格式 (如 JSON 或 XML) 的输出。这些限制可以通过多种方式实现，例如通过格式限制指令指导LLMs遵守特定格式，或者使用工业解决方案，如 JSON 模式 (OpenAI, 2024; Gemini, 2024)、Instructor (刘, 2024) 等。

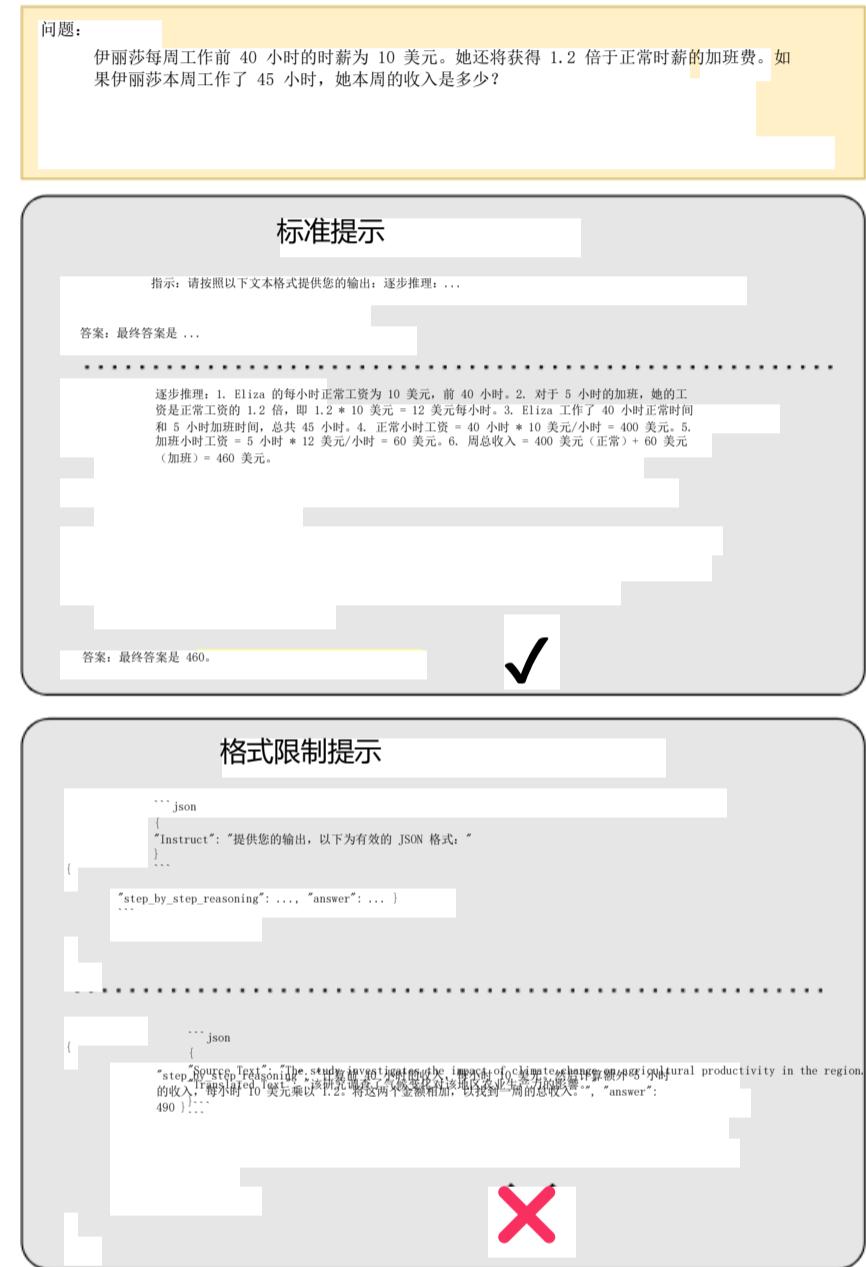


图 1: GPT-3.5-turbo 在标准自然语言中用 GSM8K 数学问题进行提示，回答正确，但在格式限制下失败。

围栏 (PrefectHQ, 2024)。这些策略简化了解析工作流程，并使LLMs与实际应用的集成更加顺畅。

由于对结构化生成的需求不断增长，研究界对调查LLMs格式跟随能力的兴趣日益增加。例如，IFEval (周等, 2023 年)、INFOBENCH (秦等, 2024 年) 和 FOFO (夏等, 2024 年) 专注于评估LLMs指令跟随能力，包括格式遵循。然而，这些研究并未解决一个关键

\* 同等贡献  
<https://github.com/appier-research/结构生成>

question for industrial applications: *Do format-restricting instructions affect the quality of LLMs’ generated content?* In other words, they fail to explore whether format restrictions degrade LLMs’ performance, which has great business impacts. This performance degradation is shown in Figure 1.

In this work, we address the aforementioned research question through extensive empirical experiments. We present a comprehensive analysis of the potential impacts of format-restricting instructions on LLMs’ performance across a wide range of tasks. The formats studied include commonly used schemas such as JSON, XML, and YAML. To the best of our knowledge, this is the first systematic investigation into the relationship between format-restricting instructions and the quality of generated content. Our contributions are twofold:

- We observe declines in LLMs’ reasoning abilities under format restrictions, with stricter constraints generally leading to greater performance degradation in reasoning tasks.
- We offer insights into why performance degrades due to format constraints and propose simple approaches to mitigate these issues, thereby achieving both consistent formats and optimal performance.
- We explore not only JSON but also other commonly used schemas like XML and YAML. Additionally, we test three different format-restricting strategies: constrained decoding, format-restricting instructions, and NL-to-Format, all of which are applicable to industrial settings.

## 2 Methodology for Structured Generation

To study different levels of format restrictions on downstream performance, we adopt the following three common methodologies in our experiments:

**Constrained Decoding (JSON-mode):** Constrained decoding is a technique that limits the output of LLMs by enforcing predefined token space during the generation process. Among mainstream LLM providers, **JSON mode** is a widely implemented instance of this technique, especially due to its extensive use in industrial settings. This mode, available as a hyperparameter flag in OpenAI and Gemini APIs, ensures the output is valid JSON. It is assumed that the implementation is similar to the constrained decoding methods described by

(Willard and Louf, 2023; Koo et al., 2024), and provided in Text-Generation-Inference<sup>2</sup>.

**Format-Restricting Instructions (FRI):** They direct the LLM to generate responses in standardized formats such as JSON, XML, and YAML, adhering to specified schemas. These instructions ensure that the generated output follows a structured format, facilitating the extraction and evaluation of the final answer. This approach is more relaxed than constrained decoding, as it does not enforce a predefined token space.

**NL-to-Format:** This two-step process first instructs the LLM to answer the question in natural language, and then instructs it to convert its response into the target format schema. As the most relaxed version of structured generation, this method decouples *content generation* from *format adherence*, aiming to maintain the performance of unrestricted natural language responses while still providing structured output.

## 3 Experiments

### 3.1 Datasets

We adopt datasets from various domains, categorized by the primary skills they assess:

#### 3.1.1 Reasoning Tasks

**GSM8K** (Cobbe et al., 2021): A collection of mathematical problems set in natural language contexts, reflecting daily life scenarios. This dataset challenges LLMs to generate necessary intermediate reasoning steps.

**Last Letter Concatenation** (Wei et al., 2022): This task requires LLMs to produce a string by concatenating the last letters of a sequence of words, testing their ability to perform symbolic reasoning.

**Shuffled Objects** (Ghazal et al., 2013): This evaluate set from BigBench evaluates the ability to infer the final state given an initial state and a sequence of shuffling events. We use the entire validation set in our experiments.

#### 3.1.2 Classification Tasks

**DDXPlus** (Tchango et al., 2022): A multiple-choice medical diagnosis dataset where LLMs must select the most appropriate diagnosis from 49 possible diseases based on a given patient profile. We use a subset provided by StreamBench (Wu et al., 2024) due to the extensive number of questions.

<sup>2</sup><https://github.com/huggingface/text-generation-inference>

这些研究没有解决工业应用中的一个关键问题：格式限制指令是否会影响LLMs生成的内容的品质？换句话说，它们未能探讨格式限制是否会降低LLMs的性能，这对商业影响巨大。这种性能下降在图1中有所体现。

在此工作中，我们通过广泛的实证实验来回答上述研究问题。我们展示了关于格式限制指令对LLMs在广泛任务中的性能潜在影响的全面分析。研究格式包括常用的模式，如JSON、XML和YAML。据我们所知，这是首次对格式限制指令与生成内容质量之间关系进行的系统研究。我们的贡献有两方面：

- 我们观察到在格式限制下，LLMs的推理能力有所下降，通常更严格的限制会导致推理任务中的性能下降更加严重。
- 我们深入分析了由于格式限制导致性能下降的原因，并提出简单的方法来缓解这些问题，从而实现格式一致性和性能最优。
- 我们不仅探索了JSON，还测试了其他常用模式，如XML和YAML。此外，我们还测试了三种不同的格式限制策略：约束解码、格式限制指令和NL-toFormat，这些策略都适用于工业环境。

## 2 结构化生成方法

为了研究不同级别的格式限制对下游性能的影响，我们在实验中采用了以下三种常见方法：约束解码（JSON模式）：约束解码是一种在生成过程中通过强制预定义的标记空间来限制LLMs输出的技术。在主流LLM提供商中，JSON模式是这种技术的广泛实施实例，尤其是在工业环境中的广泛应用。这种模式作为超参数标志存在于OpenAI和Gemini API中，确保输出是有效的JSON。

假设实现方式与（Willard和Louf, 2023; Koo等, 2024) 所描述的约束解码方法相似，并在格式限制指令中提供。

调整LLM以生成遵循指定模式的标准化格式响应，如JSON、XML和YAML。这些指令确保生成的输出遵循结构化格式，便于提取和评估最终答案。此方法比约束解码更为宽松，因为它不强制执行预定义的标记空间。

NL-to-Format：此两步过程首先指示LLM以自然语言回答问题，然后指示其将回答转换为目标格式架构。作为结构化生成的最宽松版本，此方法将内容生成与格式遵循解耦，旨在在保持不受限制的自然语言回答性能的同时，仍提供结构化输出。

## 3 实验

### 3.1 数据集

我们采用来自各个领域的数据集，按它们评估的主要技能进行分类：

#### 3.1.1 理由任务

GSM8K (Cobbe等, 2021)：一组在自然语言环境下设定的数学问题集合，反映日常生活场景。该数据集挑战LLMs生成必要的中间推理步骤。

#### 最后字母拼接 (Wei等, 2022年)：

此任务要求LLMs通过连接一系列单词的最后一个字母来生成一个字符串，以测试其进行符号推理的能力。Shuffled Objects (Ghazal等人, 2013)：此评估集来自BigBench，评估了在给定初始状态和一系列洗牌事件的情况下推断最终状态的能力。我们在实验中使用了整个验证集。

#### 3.1.2 分类任务

DDXPlus (Tchang等, 2022年)：一个多选题医学诊断数据集，其中LLMs必须根据给定的患者资料从49种可能的疾病中选择最合适的诊断。由于问题数量众多，我们使用了StreamBench (Wu等人, 2024年) 提供的子集。<sup>2</sup>  
<https://github.com/huggingface/>

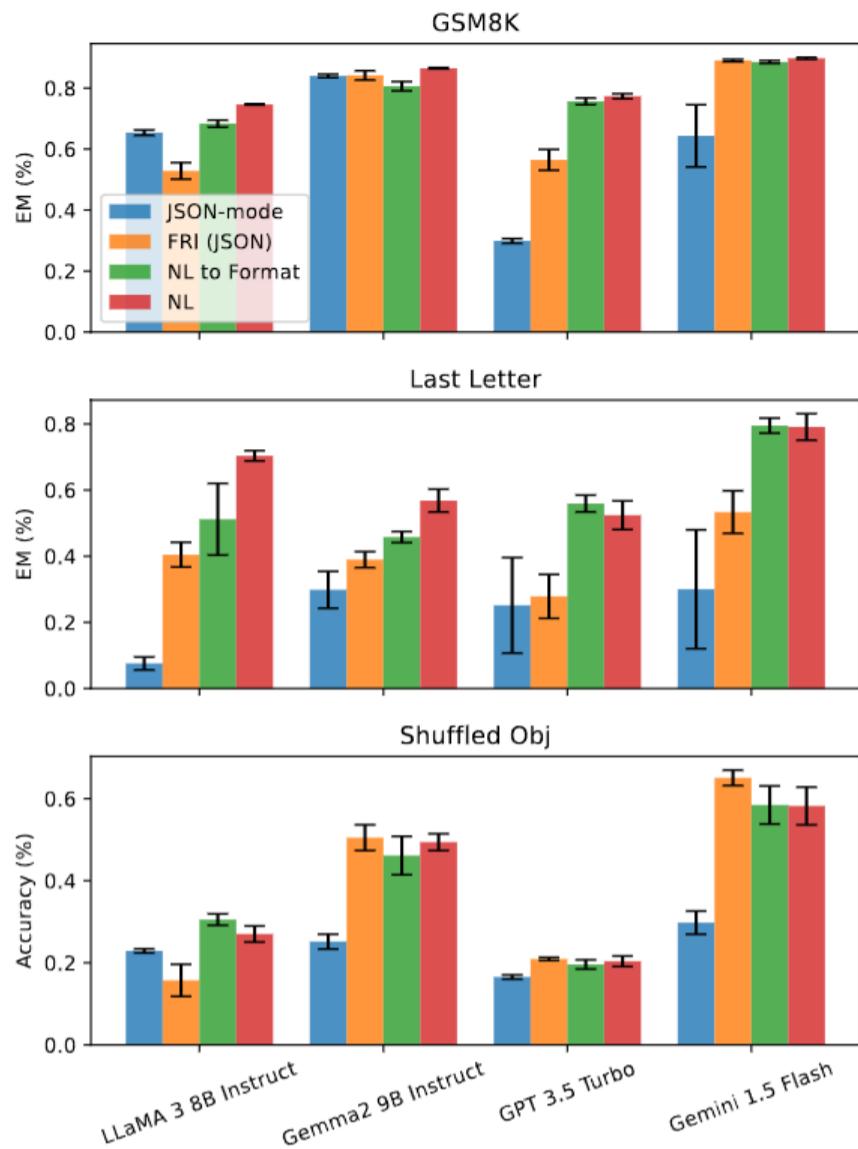


Figure 2: When comparing reasoning related task such as GSM8K, Last Letter and Shuffled Objects, we found more relaxed prompts typically yields better results as JSON-mode performs the worse in most case followed by FRI, NL to Format and Natural Language (NL)

**MultiFin** (Jørgensen et al., 2023): A multi-choice financial dataset that requires classifying a given paragraph into one of five categories.

**Sports Understanding** (Ghazal et al., 2013): This task from BigBench tests LLMs’ ability to determine whether an artificially constructed sentence relating to sports is plausible or implausible.

**NI - Task 280** (Mishra et al., 2022): A multiple-choice stereotype classification task based on a given paragraph. We included this task as it has been found to be sensitive to change in prompt formatting, with performance variations of up to 56% (Sclar et al., 2023).

### 3.2 Output Format

When designing the output format for each format, we wish to keep the schema simple; hence, we limit the number of key-value pairs for each dataset to 2: reasoning and answer fields. On top of this limitation, we permute the naming of the field names (e.g., "reasoning", "step-by-step reasoning").

While the outputs in our study may appear simplistic, converting Large Language Model (LLM)

responses to a desired format is not trivial in practice. LLMs’ output often deviates from instructions, necessitating complex parsing code to handle various response variations and edge cases, particularly when separating reasoning from the final answer. This problem is exacerbated when switching between different LLMs, as each model may have its own preferred output format, potentially breaking existing parser code. We have encountered this issue numerous times when building LLM applications, often resorting to instructing LLMs to respond in structured formats (e.g., JSON) to reduce the complexity of our parser code.

Our choice of simple output structures (one reasoning and one final answer field) was deliberate, allowing us to focus on the impact of structural bias on LLM reasoning ability, which is the primary aim of our work. We acknowledge that exploring LLM robustness with more complex output structures would be valuable. We have noted this as an important direction for future research.

### 3.3 Model

For all experiments, we compare *gpt-3.5-turbo-0125* (OpenAI, 2023), *claude-3-haiku-20240307* (Team, 2024a), *gemini-1.5-flash* (Team et al., 2023). For open weights model we use *LLAMA-3-8B-Instruct* (Team, 2024b) and *Gemma-2-9B-Instruct* (Team et al., 2024) inference using Text-Generation-Server for its support in **JSON mode**<sup>3</sup>.

### 3.4 Evaluation method

**Metrics.** To assess the performance of the models across the diverse range of tasks, we employ task-specific evaluation metrics. For the classification-based tasks (Sports Understanding, DDXPlus, Natural Instruction Task 280, and MultiFin), we use accuracy as the primary metric. For the Last Letter Concatenation and GSM8K, we utilize the exact match metric where the final answer must be the exact string match with the actual answer.

**Perfect Text Parser.** To disentangle format errors from the actual performance of the generated content, we use an LLM prompted to extract the final answer from the text, rather than relying on regex or string parsers. This approach acts as a perfect parser, minimizing errors introduced when switching between different models. Our ablation study, comparing different models, found that *claude-3-haiku-20240307* is the most consistent when using

<sup>3</sup><https://github.com/huggingface/text-generation-inference/pull/1938>

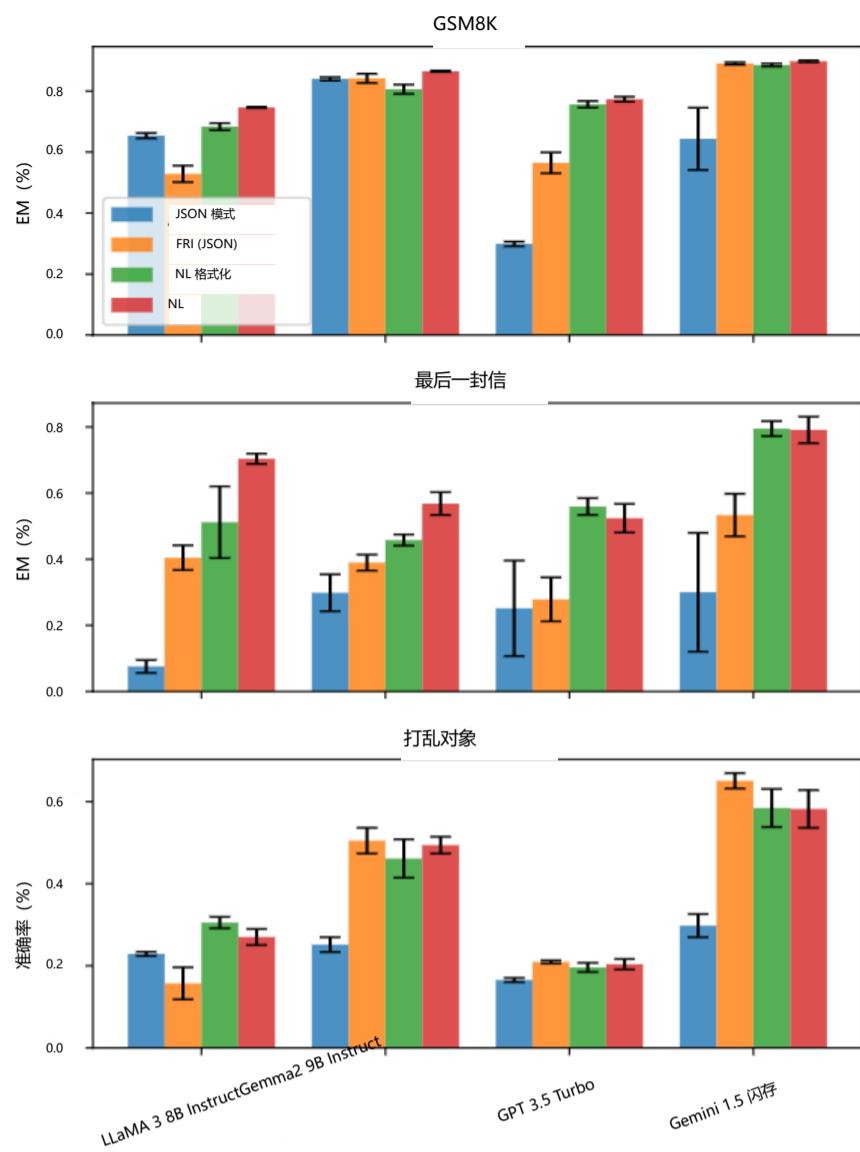


图 2：在比较与 GSM8K、最后字母和打乱的对象等相关的推理任务时，我们发现更宽松的提示通常会产生更好的结果，因为 JSON 模式在大多数情况下表现最差，其次是 FRI、NL to Format 和自然语言（NL）

多选金融数据集 (Jørgensen 等人, 2023)：需要将给定的段落分类到五个类别之一。

体育理解 (Ghazal 等人, 2013 年)：BigBench 中的这项任务测试了LLMs判断与体育相关的虚构句子是否合理或不合理的能力。

NI - 任务 280 (Mishra 等人, 2022 年)：基于给定段落的单选题型刻板印象分类任务。我们包括这个任务，因为它被发现对提示格式变化敏感，性能变化可达 56% (Sclar 等人, 2023 年)。

### 3.2 输出格式

当为每种格式设计输出格式时，我们希望保持模式简单；因此，我们将每个数据集的关键值对数量限制为 2：推理字段和答案字段。在此基础上，我们对字段名称进行排列（例如，“推理”，“逐步推理”）。

尽管我们研究中的输出可能看似简单，但将大型语言模型 (LLM) 转换为……

响应所需格式在实践中并非易事。LLMs 的输出往往偏离指令，需要复杂的解析代码来处理各种响应变体和边缘情况，尤其是在将推理与最终答案分离时。当在不同LLMs之间切换时，这个问题更加严重，因为每个模型可能都有自己的首选输出格式，这可能会破坏现有的解析代码。我们在构建LLM应用程序时多次遇到这个问题，通常不得不指示LLMs以结构化格式（例如 JSON）响应，以降低我们解析代码的复杂性。

我们的选择简单的输出结构（一个推理和一个最终答案字段）是有意为之，这使得我们能够专注于结构偏差对LLM推理能力的影响，这是我们工作的主要目标。我们承认，探索LLM鲁棒性在更复杂的输出结构中将是很有价值的。我们将此视为未来研究的一个重要方向。

### 3.3 模型

对于所有实验，我们比较了 gpt-3.5-turbo-0125 (OpenAI, 2023)，claude-3-haiku-20240307 (团队, 2024a)，gemini-1.5-flash (团队等, 2023)。对于开放权重模型，我们使用 LLaMA3-8B-Instruct (团队, 2024b) 和 Gemma-2-9BInstruct (团队等, 2024) 进行推理，利用 TextGeneration-Server 的 JSON 模式支持。

### 3.4 评估方法

**指标。**为了评估模型在各类任务中的性能，我们采用特定任务的评估指标。对于基于分类的任务（体育理解、DDXPlus、自然指令任务 280 和 MultiFin），我们使用准确率作为主要指标。对于最后字母拼接和 GSM8K，我们使用精确匹配指标，其中最终答案必须与实际答案完全字符串匹配。

**完美文本解析器。**为了将格式错误与生成内容的实际性能区分开来，我们使用LLM提示从文本中提取最终答案，而不是依赖于正则表达式或字符串解析器。这种方法充当完美的解析器，最小化了在不同模型之间切换时引入的错误。我们的消融研究，比较了不同的模型，发现 claude-3haiku-20240307 在使用时最为一致。

无法直接访问外部链接，因此无法提供翻译。请提供需要翻译的文本内容，我将为您翻译成简体中文。

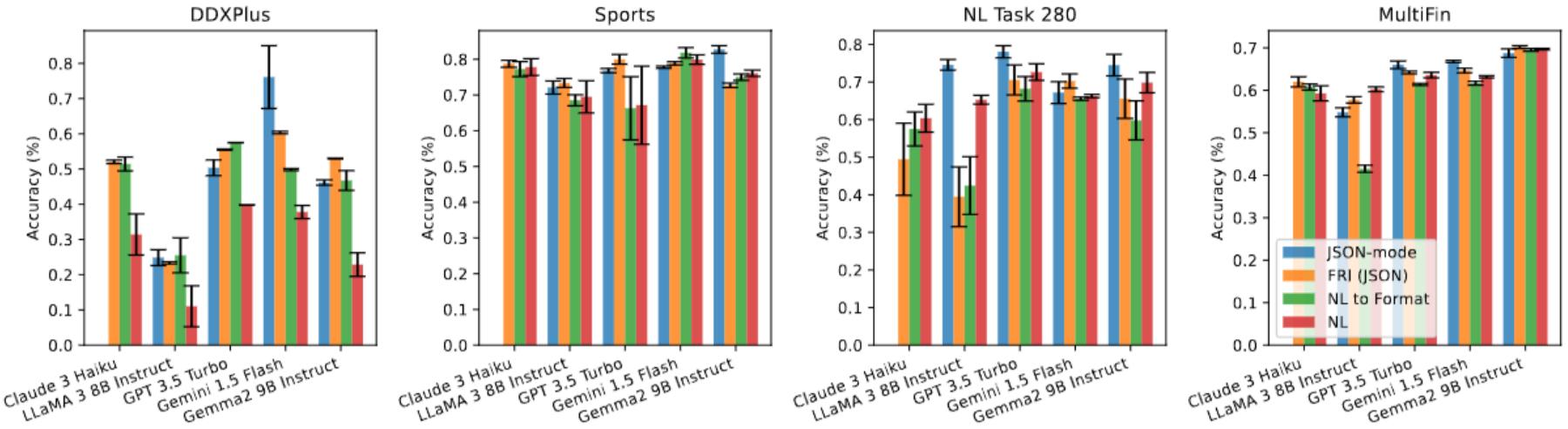


Figure 3: Classification related tasks on DDXPlus, Sports, Task280 and Multifin in different levels of format restriction.

*gpt-4-turbo* as a human reference, compared to four other low-cost APIs. Detailed comparison between *gpt-4-turbo* between human parsed answers as well as comparison of other LLMs can be found in Appendix B.

**Consideration for Prompt Sensitivity.** Previous studies (Chen et al., 2023; Sclar et al., 2023; Zhu et al., 2023; Mizrahi et al., 2024) have shown that LLMs are sensitive to slight variations in prompts. To account for this, we evaluate our approach by nine prompt combinations: three task descriptions and three JSON, XML, and YAML schemas with slight variations in wording or format. For natural language prompting, we include three variations in text formats (e.g., *Give your reason first followed by your answers*). Details of the task description prompts and FRI prompts can be found in Appendix G.

## 4 Main Results

### 4.1 Impact of Format Restriction on Final Results

We investigate the effects of format restrictions on LLM performance by examining three progressively relaxed prompting approaches: JSON-mode, FRI, and NL-to-Format conversion.

We evaluate these approaches on datasets with exact match scores: GSM8K and Last Letter Concatenation presented in Figure 2. Surprisingly, JSON-mode performs significantly worse than FRI (JSON) on the Last Letter task. Upon inspection, we found that 100% of GPT 3.5 Turbo JSON-mode responses placed the "answer" key before the "reason" key, resulting in zero-shot direct answering instead of zero-shot chain-of-thought reasoning.

Comparing NL-to-Format with unrestricted Natural Language responses, we observe nearly identical performance across most models, as both de-

rive answers from the same initial natural language response. However, NL-to-Format occasionally introduces generation errors, leading to slightly lower performance for LLaMA 3 8B Instruct, while other models maintain consistent scores across both settings.

These findings suggest that the degree and implementation of format restrictions can significantly impact LLM performance, particularly in reasoning tasks. The order of keys in structured outputs and the decoupling of reasoning from format adherence emerge as important factors in maintaining LLM capabilities while providing structured responses.

When evaluating classification datasets, we observe a different trend compared to reasoning tasks, as illustrated in Figure 3. Notably, in the DDXPlus dataset, Gemini 1.5 Flash demonstrates a significant performance boost when JSON-mode is enabled. Across other classification datasets, JSON-mode performs competitively, and in some cases, surpasses the other three methodologies.

We hypothesize that JSON-mode improves classification task performance by constraining possible answers resulted in reducing errors in answer selection. Conversely, natural language responses may introduce distractions, leading to parsing errors. These findings suggest format restrictions' impact on LLM performance is task-dependent: stringent formats may hinder reasoning-intensive tasks but enhance accuracy in classification tasks requiring structured outputs.

## 5 Discussion

### 5.1 Impact on looser format restriction

To further investigate the effects of format restrictions, we examine a variation of the Soft Restrict setting where we remove the schema restriction from the prompt description. Instead of providing a

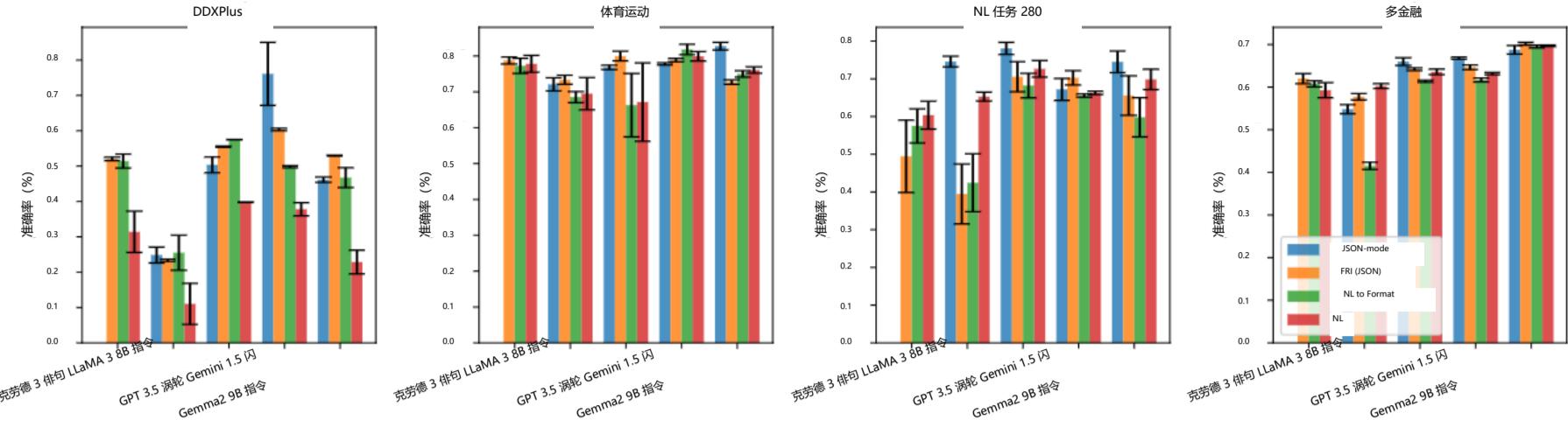


图 3: DDXPlus、体育、Task280 和 Multifin 在不同格式限制级别上的分类相关任务。

gpt-4-turbo 作为人类参考，与四个其他低成本 API 进行比较。gpt-4-turbo 与人类解析答案之间的详细比较以及其他LLMs的比较可以在附录 B 中找到。

考虑提示敏感性。先前的研究 (Chen 等, 2023; Sclar 等, 2023; Zhu 等, 2023; Mizrahi 等, 2024) 表明LLMs对提示的微小变化敏感。为了对此进行考虑，我们通过九种提示组合评估我们的方法：三种任务描述和三种具有轻微文字或格式变化的 JSON、XML 和 YAML 模式。对于自然语言提示，我们包括三种文本格式变化（例如，先给出你的理由，然后给出你的答案）。任务描述提示和 FRI 提示的详细信息可在附录 G 中找到。

由于两者都从相同的初始自然语言响应中得出答案。然而，NL-to-Format 偶尔会引入生成错误，导致 LLaMA 3 8B Instruct 的性能略有下降，而其他模型在两种设置中保持一致的分数。

这些发现表明，格式限制的程度和实施方式可以显著影响LLM的性能，尤其是在推理任务中。在结构化输出中键的顺序以及将推理与格式遵循分离成为在提供结构化响应的同时维持LLM能力的重要因素。

当评估分类数据集时，我们观察到与推理任务相比存在不同的趋势，如图 3 所示。值得注意的是，在 DDXPlus 数据集中，启用 JSON 模式时，Gemini 1.5 Flash 表现出显著的性能提升。在其他分类数据集中，JSON 模式表现具有竞争力，在某些情况下甚至超过了其他三种方法。

我们假设 JSON 模式通过限制可能的答案，从而减少答案选择中的错误，提高了分类任务的性能。相反，自然语言回答可能会引入干扰，导致解析错误。这些发现表明，格式限制对LLM性能的影响是任务依赖的：严格的格式可能会阻碍需要推理的任务，但在需要结构化输出的分类任务中可以提高准确性。

## 4 主要结果

### 4.1 格式限制对最终结果的影响

我们通过考察三种逐步放宽提示方法的性能，即 JSON 模式、FRI 和 NL 到格式转换，来研究格式限制对LLM性能的影响。

我们在这组具有精确匹配分数的数据集上评估这些方法：图 2 中展示的 GSM8K 和 Last Letter Concatenation。令人惊讶的是，JSON 模式在 Last Letter 任务上的表现显著低于 FRI (JSON)。经过检查，我们发现 100% 的 GPT 3.5 Turbo JSON 模式响应都将 "answer" 键放在 "reason" 键之前，导致零样本直接回答而不是零样本思维链推理。

比较 NL-to-Format 与无限制的自然语言响应，我们发现大多数模型在性能上几乎相同

## 5 讨论

### 5.1 对更宽松的格式限制的影响

为了进一步研究格式限制的影响，我们考察了软限制设置的一种变体，其中我们从提示描述中移除了模式限制。我们不是提供

<b>Model</b>	<b>Text</b>	<b>JSON</b>	<b>XML</b>	<b>YAML</b>
<i>gemini-1.5-flash</i>	89.33 (0.8)	<b>89.66</b> (0.3)	<b>89.26</b> (0.3)	<b>89.21</b> (0.4)
+ schema constraint	-	89.21 (1.5)	88.20 (2.2)	87.42 (3.7)
<i>claude-3-haiku</i>	86.51 (0.8)	<b>86.99</b> (0.2)	<b>86.96</b> (0.6)	<b>82.89</b> (5.7)
+ schema constraint	-	23.44 (22.9)	79.76 (7.0)	80.63 (2.8)
<i>gpt-3.5-turbo</i>	75.99 (3.1)	<b>74.70</b> (1.1)	<b>60.45</b> (7.2)	71.58 (3.0)
+ schema constraint	-	49.25 (12.0)	45.06 (19.9)	<b>73.85</b> (5.6)
<i>LLaMA-3-8B</i>	75.13 (0.9)	<b>64.67</b> (2.23)	<b>65.07</b> (0.56)	<b>69.41</b> (0.95)
+ schema constraint	-	48.90 (6.7)	56.74 (8.3)	46.08 (16.8)

Table 1: Comparing results without and with schema constraint, adding schema not only increase the sensitivity to prompt but also degrade in average performance.

specific schema (e.g., *"Reply your answer in JSON format with the following schema: { "reason": ..., "answer": ... }"*), we simply instruct the LLM to output in the target format language (e.g., *"Reply your answer in JSON format."*). Table 1 illustrates the effects of removing the schema restriction on the GSM8K dataset. We observe significant improvements in average scores and lower standard deviations across different prompt perturbations for Claude 3 Haiku, GPT-3.5 Turbo, and LLaMA 3 8B Instruct. These results suggest that while structured outputs can be beneficial for downstream processing, overly restrictive schemas may hinder LLM performance, particularly in reasoning-intensive tasks.

This finding suggests that a balance must be struck between the desire for easily parseable, structured outputs and the need to preserve the LLM’s inherent reasoning abilities. Practitioners may want to consider using looser format restrictions when dealing with complex reasoning tasks, while still maintaining some level of structure to facilitate downstream processing.

## 5.2 Comparison Across Different Formats

In this section we ablate the format language by comparing not just JSON but also XML and YAML format. Since all 3 language comes in different grammar syntax rules and restriction. We expect each models might perform differently for example Claude-3-Haiku uses XML for tool use schema.

On hindsight we do not see any structure format

which consistency stands out from others which generalized across all models in Figure 4. For Gemini model, we found JSON is more consistent however it does not always outperform other format for example Claude-3-Haiku.

In Table 11 we found in classification task JSON-mode performs much better than text due to the restriction on answer space. However in reasoning related task, JSON-mode failed to adhere to the order of reasoning first followed by answer causing a large drop in final performance.

## 5.3 Structure Format and Parsing Error Rates

We initially hypothesized that the performance gap between text and structured formats might be attributed to parsing errors during answer extraction. However, our analysis of error rates across different formats and models, as shown in Table 3, reveals that this is not the primary factor. In fact, Gemini 1.5 Flash and GPT 3.5 Turbo exhibit near zero parsing failures in all three formats. In the LLaMA 3 8B setting, the parsing error rate for the Last Letter task in JSON format is only 0.148%, yet there exists a substantial 38.15% performance gap as seen in Table 1.

This finding suggests that the performance differences between formats are not primarily due to parsing errors, but rather to the impact of format restrictions on the LLM’s reasoning and generation processes. However, we discovered that parsing errors, when present, can be effectively mitigated through a simple corrective step.

By prompting Claude-3-Haiku to reformat any output with parsing errors for both Claude 3 Haiku and LLaMA 3 8B (the two models with the highest percentage of parsing errors), we observed improved scores in JSON and YAML formats, as illustrated in Figure 5. This approach demonstrates the potential for enhancing the reliability of structured outputs without sacrificing the benefits of format-specific optimizations.

## 5.4 Study on Structure Generation with Context-free Grammars

A newer revision of the model *gpt-4o-mini-2024-07-18* now supports Context-free Grammars via a so-called Structure Output API. This API allows users to provide a predefined JSON schema, ensuring the response adheres to it with 100% guarantee. It’s important to note that this differs from the previously mentioned JSON-mode on OpenAI’s mod-

模型	文本	JSON	XML	YAML
Gemini-1.5 Flash	89.33	89.66	89.26	89.21
(0.3) (0.3) (0.4) + 架构约束	0.8			
束 -	89.21	88.20	87.42	
	-	(1.5)	(2.2)	(3.7)
克劳德-3-俳句	86.51	86.99	86.96	82.89
(0.2) (0.6) (5.7) + 架构约束	0.8			
束 -	23.44	79.76	80.63	
	-	(22.9)	(7.0)	(2.8)
gpt-3.5-turbo	75.99	74.70	60.45	71.58
(1.1) (7.2) (3.0) (3.1)				
架构约束 -	49.25	45.06		
73.85	-	(12.0)	(19.9)	(5.6)
LLaMA-3-8B	75.13	64.67	65.07	69.41
(2.23) (0.56) (0.95) + 架构约束	9			
约束 -	48.90	56.74	46.08	
	-	(6.7)	(8.3)	(16.8)

表 1：比较有无模式约束的结果，添加模式不仅增加了对提示的敏感性，而且降低了平均性能。

特定模式（例如，“请以 JSON 格式回复您的答案，格式如下：{ "reason": ..., "answer": ... }”），我们只需指示LLM以目标格式语言输出（例如，“请以 JSON 格式回复您的答案。”）。表 1 说明了移除模式限制对 GSM8K 数据集的影响。我们观察到，在 Claude 3 Haiku、GPT-3.5 Turbo 和 LLaMA 3 8B Instruct 的不同提示扰动下，平均分数显著提高，标准偏差降低。这些结果表明，虽然结构化输出可能对下游处理有益，但过于严格的模式可能会阻碍LLM的性能，尤其是在推理密集型任务中。  
这一发现表明，在易于解析的结构化输出和保留LLM固有推理能力之间必须取得平衡。实践者可能需要考虑在处理复杂推理任务时使用更宽松的格式限制，同时仍保持一定程度的结构以方便后续处理。

## 5.2 不同格式的比较

在这一节中，我们通过比较不仅仅是 JSON，还包括 XML 和 YAML 格式来消除格式语言。由于这三种语言都有不同的语法句法和限制。我们预计每个模型可能表现不同，例如 Claude-3-Haiku 使用 XML 作为工具使用模式。

回顾过去，我们没有看到任何结构格式

哪种一致性在图 4 中脱颖而出，它概括了所有模型。对于 Gemini 模型，我们发现 JSON 的一致性更高，但它并不总是优于其他格式，例如 Claude-3-Haiku。

在表 11 中，我们发现由于答案空间的限制，在分类任务中 JSONmode 的表现远优于文本。然而，在推理相关任务中，JSON 模式未能遵循先推理后回答的顺序，导致最终性能大幅下降。

## 5.3 结构格式与解析错误率

我们最初假设文本和结构化格式之间的性能差距可能归因于答案提取过程中的解析错误。然而，如表 3 所示，我们对不同格式和模型错误率的分析表明，这并非主要因素。实际上，Gemini 1.5 Flash 和 GPT 3.5 Turbo 在所有三种格式中均表现出近乎零的解析失败。在 LLaMA 3 8B 设置中，JSON 格式中 Last Letter 任务的解析错误率仅为 0.148%，但如表 1 所示，性能差距仍有 38.15%。

这一发现表明，不同格式之间的性能差异并非主要由于解析错误，而是由于格式限制对LLM推理和生成过程的影响。然而，我们发现，当存在解析错误时，可以通过简单的纠正步骤有效地减轻其影响。

通过提示 Claude-3-Haiku 重新格式化 Claude 3 Haiku 和 LLaMA 3 8B（两种模型中解析错误百分比最高的模型）的任何输出，我们发现 JSON 和 YAML 格式的分数有所提高，如图 5 所示。这种方法展示了在不牺牲格式特定优化优势的情况下，提高结构化输出可靠性的潜力。

## 5.4 研究基于上下文无关文法进行结构生成的方法

模型 gpt-4o-mini-202407-18 的新版本现在通过所谓的结构输出 API 支持上下文无关文法。该 API 允许用户提供预定义的 JSON 模式，确保响应 100% 符合该模式。需要注意的是，这与之前提到的 OpenAI 的 mod-的 JSON 模式不同。

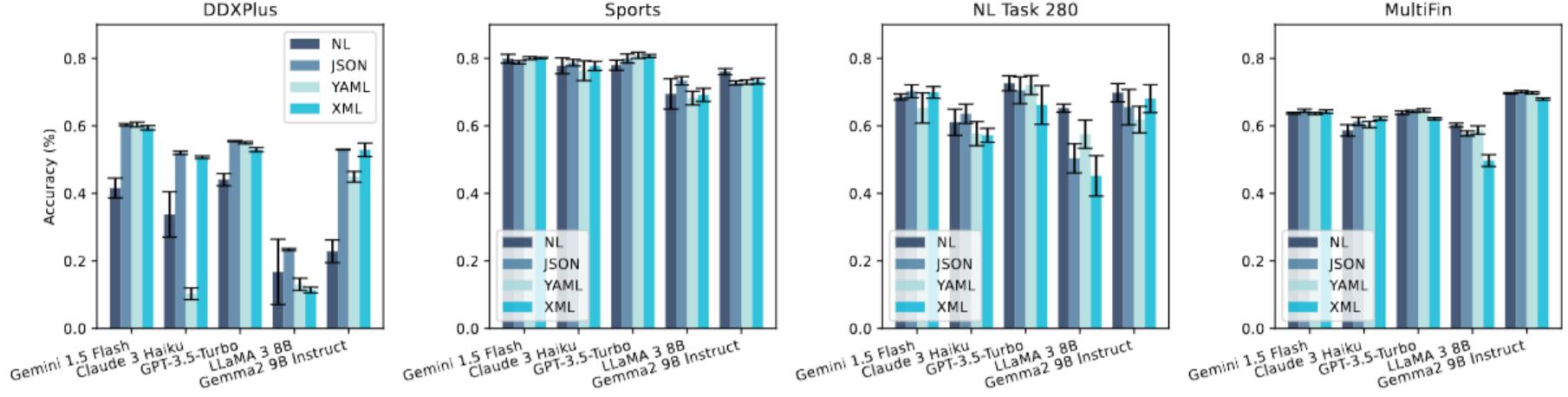


Figure 4: Comparison of different formats in classification related tasks on DDXPlus, Sports, Task280 and Multifin. NL=Natural Language. We showed the averaged accuracy for each format over 9 different prompts with standard deviation error.

Task	NL	FRI	JSON-Mode	JSON-Schema
GSM8K	<b>94.57</b> (3.95)	87.17 (4.43)	86.95 (1.36)	91.71 (0.68)
Shuffle Obj	<b>82.85</b> (5.67)	81.46 (3.71)	76.43 (9.74)	81.77 (6.86)
Last Letter	83.11 (3.54)	84.73 (2.99)	76.00 (6.69)	<b>86.07</b> (3.33)

Table 2: Performance of *gpt-4o-mini-2024-07-18* across tasks and formats. In 2 out of 3 reasoning datasets, NL (Natural Language) still performs slightly better than JSON-Schema.

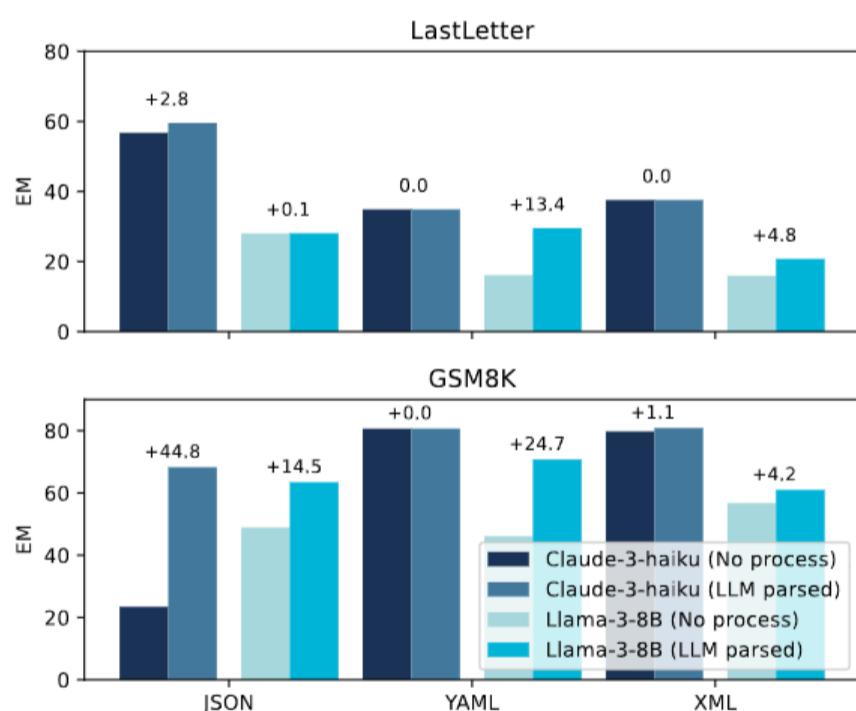


Figure 5: We found high parsing errors in Table 3 can be patched by calling a second prompt to fix any syntax error found in the previous response.

els, which uses the OpenAI function calling API. We conducted experiments on 3 reasoning datasets using *gpt-4o-mini*, denoting the newer structured output method as JSON-schema. Results are shown in Table 2.

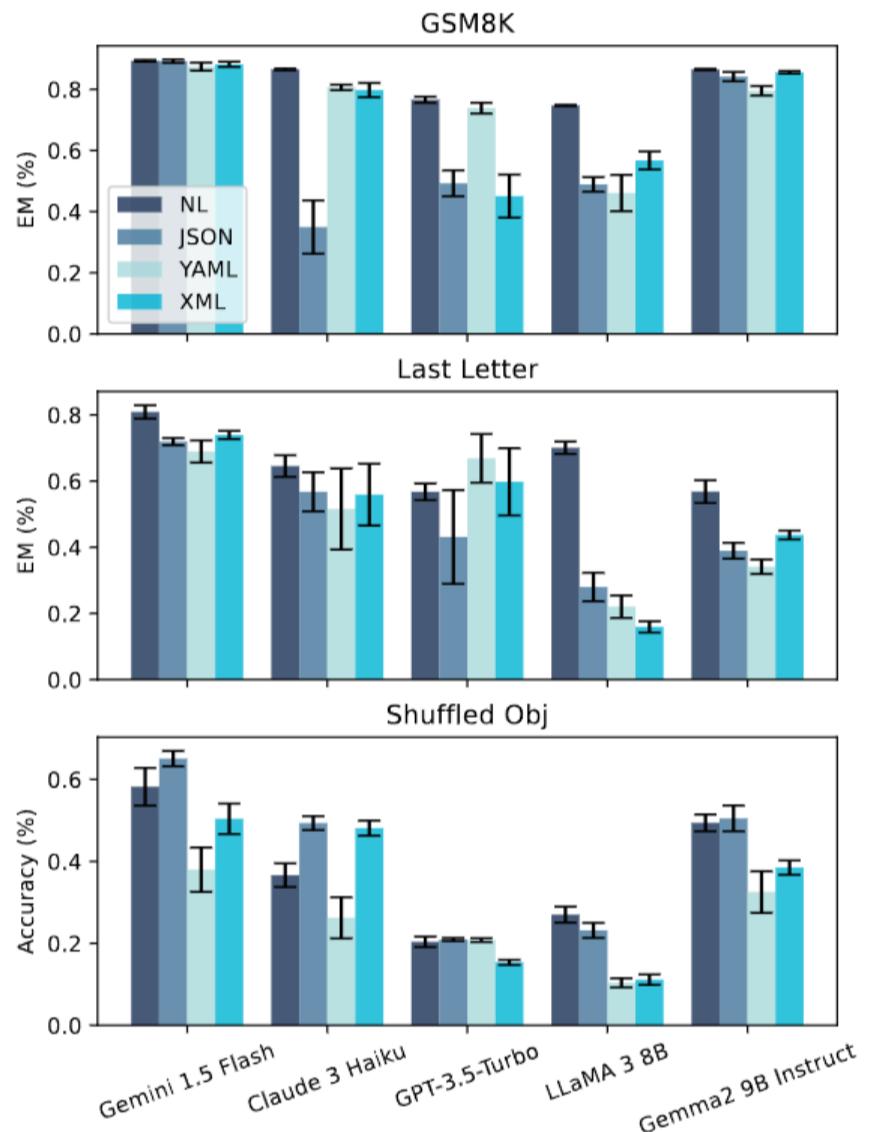


Figure 6: Comparison of JSON, YAML, XML with Natural Language (NL) response on reasoning related task. NL still performs better than other formats with the exception of GPT-3.5-Turbo.

## 6 Related Work

Our study can be summarized into two genres : reasoning ability of LLM and format following.

In study of LLMs reasoning ability, early work by (Kojima et al., 2022) found using "Think step-by-step" can elicit reasoning ability without few shot examples. Subsequent study (Jin et al., 2024) shows that the number of reasoning steps correlates with the final accuracy. Recent work by (Wang and Zhou, 2024) found Chain-of-Thought (CoT)

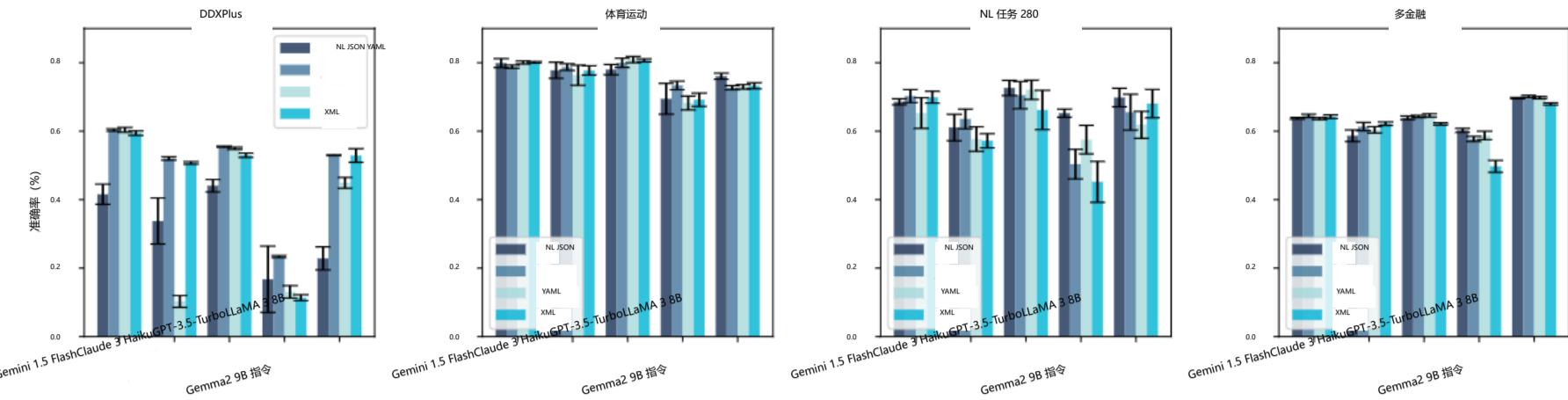


图 4: 在 DDXPlus、Sports、Task280 和 Multifin 的分类相关任务中不同格式的比较。NL=自然语言。我们展示了每种格式在 9 个不同提示下的平均准确率及标准误差。

Task	NL FRI JSON-Mode JSON-Schema		
	NL	JSON	YAML
GSM8K	94.57 (3.95)	87.17 (4.43)	86.95 (1.36)
打乱 Obj	82.85 (5.67)	81.46 (3.71)	76.43 (9.74)
最后信件	83.11 (3.54)	84.73 (2.99)	76.00 (6.69)
			86.07 (3.33)

表 2: gpt-4o-mini-2024-07-18 在不同任务和格式下的性能。在 3 个推理数据集中有 2 个，NL (自然语言) 的性能仍然略优于 JSON-Schema。

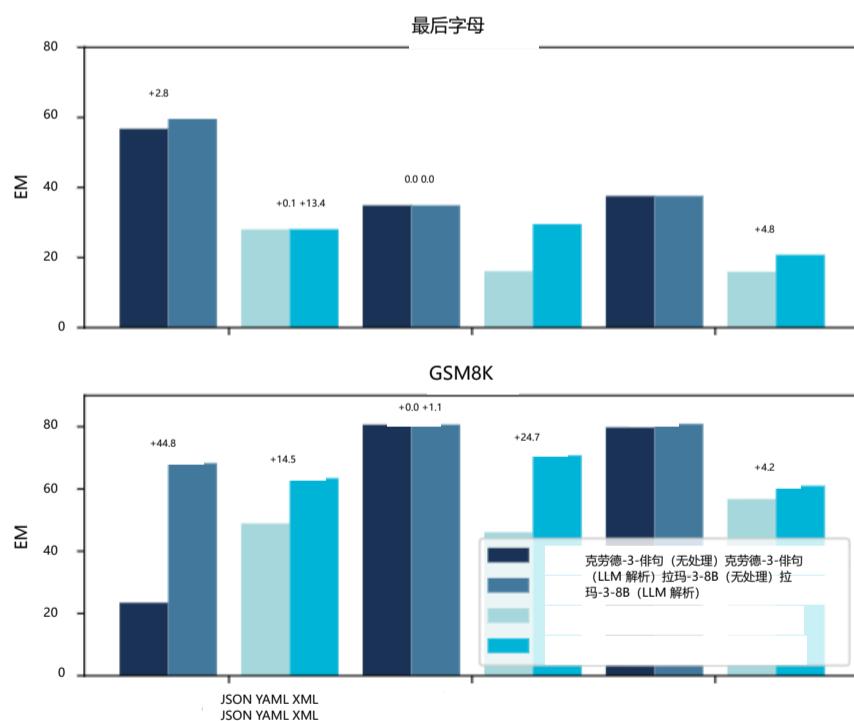


图 5: 我们发现表 3 中的高解析错误可以通过调用第二个提示来修复前一个响应中发现的任何语法规错误。

els，使用 OpenAI 函数调用 API。我们使用 gpt-4o-mini 在 3 个推理数据集上进行了实验，将较新的结构化输出方法表示为 JSON-schema。结果如表 2 所示。

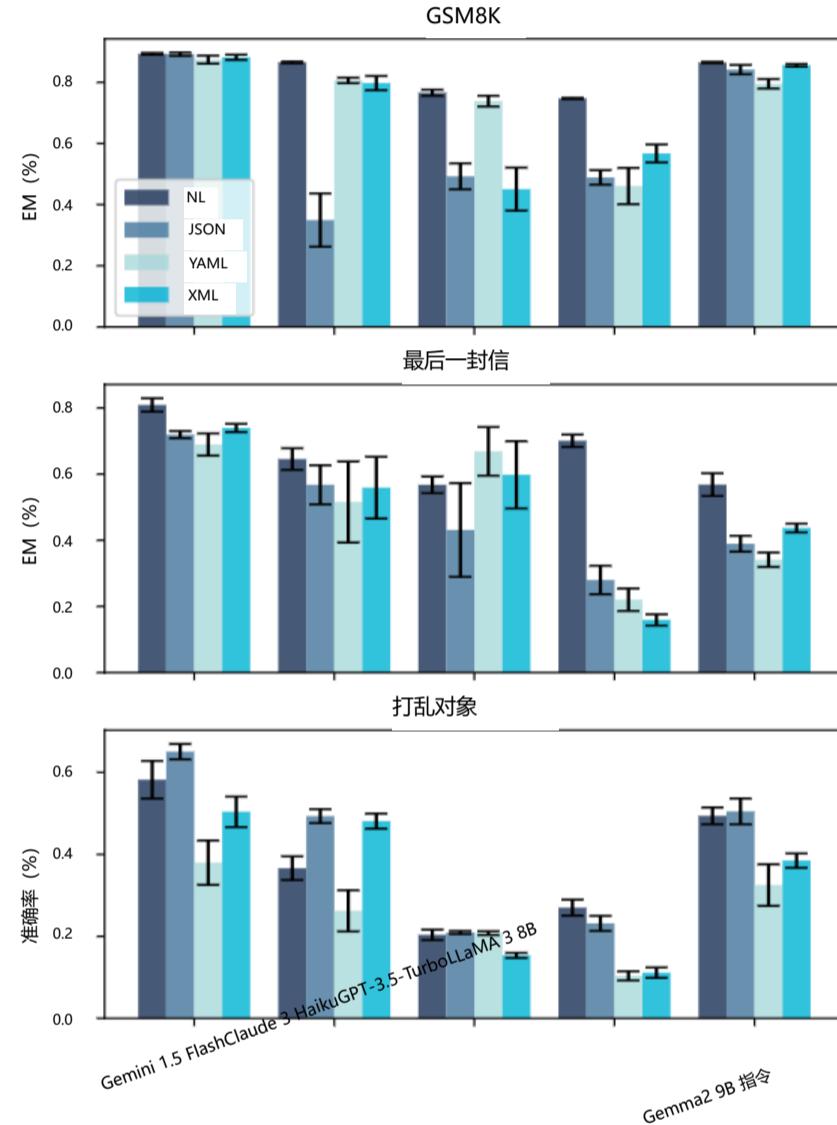


图 6: JSON、YAML、XML 与自然语言 (NL) 在推理相关任务中的比较。NL 的表现仍优于其他格式，除了 GPT-3.5-Turbo。

## 6 相关研究

我们的研究可以总结为两种类型：LLM的推理能力和格式遵循。

在研究LLMs推理能力时，(Kojima 等人, 2022)的早期研究发现使用“逐步思考”可以在没有少量示例的情况下激发推理能力。后续研究(金等人, 2024)表明推理步骤的数量与最终准确性相关。最近(王和周, 2024)的研究发现思维链 (CoT)

Table 3: Parsing error percentage across different models. We want to highlight that despite having near zero parsing error in Gemini-Flash XML and YAML, there’s still degradation in the final benchmark scores.

Model	Format	Task		Reasoning				Classification			
		Last Letter	GSM8K	DDXPlus	Sports	Task280	MultiFin				
Gemini-Flash	JSON	0.0	0.03	0.37	0.0	0.0	0.0				
	XML	0.0	0.19	1.26	0.0	0.22	0.0				
	YAML	0.0	0.0	0.68	0.06	6.46	0.0				
Claude-3-Haiku	JSON	3.48	60.07	0.09	0.0	10.26	0.0				
	XML	0.0	1.85	0.48	0.0	0.41	0.0				
	YAML	0.0	0.0	86.66	1.02	0.13	0.0				
GPT-3.5-Turbo	JSON	0.0	0.13	0.0	0.0	0.0	0.0				
	XML	0.0	0.24	0.35	0.0	0.0	0.0				
	YAML	0.0	0.0	0.32	1.23	0.08	0.0				
LLaMA 3 8B	JSON	0.15	22.75	1.63	0.28	1.61	0.0				
	XML	17.93	7.62	32.45	6.54	22.04	5.78				
	YAML	32.40	33.18	34.40	7.16	2.19	0.14				

reasoning seed prompt (Kojima et al., 2022) can be removed with a carefully crafted CoT decoding schema.

The exploration of LLMs’ ability to follow instructions and produce responses in specified formats was first addressed by IFEval (Zhou et al., 2023) which aimed to evaluate the general instruction-following ability of LLMs, and it contains a subset of test instances specifically assessing format-following. INFOBENCH (Qin et al., 2024) introduces a broader coverage of instructions and conducts a more fine-grained analysis by decomposing the instructions into different categories, including format specifications. FOFO (Xia et al., 2024) is a benchmark solely focused on the format-following ability of LLMs. However, these works do not explore if format instruction interfere with downstream performance.

## 7 Conclusion

Our study reveals that structured generation constraints significantly impact LLM performance across various tasks. Format restrictions, particularly constrained decoding (JSON-mode), can hinder reasoning abilities while enhancing classification task accuracy. Looser format restrictions generally improve performance and reduce variance in reasoning tasks. Parsing errors, while not the primary cause of performance differences, can be mitigated through corrective prompting. These findings underscore the importance of balancing format adherence, reasoning capabilities, and cost efficiency in LLM applications. Given that our study focuses on reasoning-intensive tasks, future work should explore how reasoning tasks of vary-

ing difficulty, from intensive to simple, are affected by restrictive formats and LLMs. To mitigate the performance degradation of LLMs due to restrictive formats, future studies should include a wider range of training data that contains instructions in various restrictive formats in local LLMs.

## References

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Yulin Chen, Ning Ding, Xiaobin Wang, Shengding Hu, Haitao Zheng, Zhiyuan Liu, and Pengjun Xie. 2023. Exploring lottery prompts for pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15428–15444.
- 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*.
- Google Gemini. 2024. Generate json output with the gemini api. <https://ai.google.dev/gemini-api/docs/json-mode?lang=python>. Accessed on 2024-07-02.
- Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Croquette, and Hans-Arno Jacobsen. 2013. Bigbench: Towards an industry standard benchmark for big data analytics. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 1197–1208.

表 3：不同模型中的解析错误百分比。我们想强调的是，尽管在 Gemini-Flash XML 和 YAML 中解析错误接近零，但最终的基准分数仍然有所下降。

模型	Task 格式	推理			分类		
		最后字母	GSM8K	DDXPlus	体育任务	280	多鳍
Gemini-Flash	JSON	0.0	0.03	0.37	0.0	0.0	0.0
	XML	0.0	0.19	1.26	0.0	0.22	0.0
	YAML	0.0	0.0	0.68	0.06	6.46	0.0
克劳德-3-俳句	JSON	3.48	60.07	0.09	0.0	10.26	0.0
	XML	0.0	1.85	0.48	0.0	0.41	0.0
	YAML	0.0	0.0	86.66	1.02	0.13	0.0
GPT-3.5-Turbo	JSON	0.0	0.13	0.0	0.0	0.0	0.0
	翻译文本：GPT-3.5- <del>XML</del>	JSON	0.0	0.24	0.35	0.0	0.0
	YAML	0.0	0.0	0.32	1.23	0.08	0.0
LLaMA 3 8B	JSON	0.15	22.75	1.63	0.28	1.61	0.0
	翻译文本：LLaMA 3 8B	JSON	0.15	32.45	6.54	22.04	5.78
	YAML	0.14	32.40	33.18	34.40	7.16	2.19

推理种子提示（小岛等，2022）可以通过精心设计的 Cot 解码方案去除。

探索LLMs遵循指令并按指定格式生成响应的能力，首先由 IFEval（周等，2023）提出，旨在评估LLMs的通用指令遵循能力，其中包含专门评估格式遵循的测试实例子集。INFOBENCH（秦等，2024）引入了更广泛的指令覆盖范围，并通过将指令分解为不同类别进行更细致的分析，包括格式规范。FOFO（夏等，2024）是一个仅关注LLMs格式遵循能力的基准。然而，这些工作并未探讨格式指令是否会干扰下游性能。

未来工作应探讨从密集到简单不同难度的推理任务如何受到限制格式和LLMs的影响。为了减轻限制格式对LLMs性能下降的影响，未来的研究应包括包含各种限制格式指令的更广泛的本地LLMs训练数据。

## 参考文献

汤姆·布朗，本杰明·曼，尼克·赖德，梅拉妮·苏比亚，贾里德·D·卡普兰，普拉夫拉·达里瓦尔，阿夫林德·尼尔拉坎坦，普拉纳夫·希亚姆，吉里什·萨斯特里，阿曼达·阿斯凯尔，等人。2020。语言模型是少样本学习者的。神经信息处理系统进展，33：1877–1901。

陈玉林，丁宁，王晓斌，胡胜定，郑海涛，刘志远，谢鹏军。2023。

探索预训练语言模型的彩票提示模型。在《计算语言学协会第 61 届年度会议论文集》（第 1 卷：长篇论文），第 15428–15444 页。

卡尔·科贝，维奈特·科萨拉朱，穆罕默德·巴维里安，马克·陈，河宇俊，卢卡斯·凯撒，马蒂亚斯·普拉珀特，杰瑞·特沃雷克，雅各布·希尔顿，瑞伊基罗中野，克里斯托弗·赫塞，约翰·舒尔曼。2021。训练验证者解决数学文字问题。arXiv 预印本 arXiv:2110.14168。

谷歌 Gemini. 2024. 生成 JSON 输出使用 Gemini API。https://ai.google.dev/gemini-api/docs/json-mode?lang=python。访问日期：2024-07-02。

Ahmad Ghazal, Tilman Rabl, Mingqin Hu, Francois Raab, Meikel Poess, Alain Crolette, 和 Hans-Arno Jacobsen. 2013. Bigbench：迈向大数据分析行业标准的基准。在《Proceedings...》2013 年 ACM SIGMOD 国际数据管理会议论文集，第 1197–1208 页。

## 7 结论

我们的研究显示，结构化生成约束对各种任务中的LLM性能有显著影响。格式限制，尤其是受约束的解码（JSON模式），可能会阻碍推理能力，同时提高分类任务的准确性。较宽松的格式限制通常能提高性能并减少推理任务中的方差。虽然解析错误不是性能差异的主要原因，但可以通过纠正提示来减轻。这些发现强调了在LLM应用中平衡格式遵循、推理能力和成本效率的重要性。鉴于我们的研究专注于推理密集型任务

- Mingyu Jin, Qinkai Yu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, Mengnan Du, et al. 2024. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*.
- Rasmus Kær Jørgensen, Oliver Brandt, Mareike Hartmann, Xiang Dai, C. Igel, and Desmond Elliott. 2023. Multifin: A dataset for multilingual financial nlp. In *ACL Findings*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.
- Terry Koo, Frederick Liu, and Luheng He. 2024. Automata-based constraints for language model decoding. *arXiv e-prints*.
- Jason Liu. 2024. [instructor](#).
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *ACL*.
- Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.
- OpenAI. 2023. Gpt-4 technical report.
- OpenAI. 2024. Json mode. <https://platform.openai.com/docs/guides/text-generation/json-mode>. Accessed on 2024-07-02.
- PrefectHQ. 2024. [marvin](#).
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.
- Arsène Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. Ddxplus: a new dataset for automatic medical diagnosis. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 31306–31318.
- Anthropic Team. 2024a. Introducing the next generation of claude.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Meta LLaMA Team. 2024b. Introducing meta llama 3: The most capable openly available llm to date.
- Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *ArXiv*, abs/2402.10200.
- Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.
- Brandon T Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *arXiv e-prints*, pages arXiv–2307.
- Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung yi Lee. 2024. Streambench: Towards benchmarking continuous improvement of language agents.
- Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofo: A benchmark to evaluate llms’ format-following capability. *arXiv preprint arXiv:2402.18667*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.
- Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, et al. 2023. Prompt-bench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

## A Limitation

This study contains two primary limitations. First, due to cost constraints, we were unable to include results from more powerful language models such as LLaMA 70B or GPT-4o in our experiments. The inclusion of these models could potentially provide additional insights into how performance scales

金明宇, 余勤凯, 赵海燕, 华文月, 孟艳达, 张永锋, 杜梦娜, 等

2024. 推理步骤长度对大型的影响  
语言模型。arXiv 预印本 arXiv:2401.04925。

拉一斯穆斯·凯尔·约根森, 奥利弗·布兰特, 玛丽克·哈特曼, 向代, C.伊格尔, 以及德蒙特·埃利奥特。2023 年。  
Multifin: 多语言金融自然语言处理数据集。在 ACL  
发现中。

小岛隆史, 顾石祥, 里德, 松尾裕隆, 岩泽祐介。

2022。《大型语言模型是零样本推理者》。载于《神经  
信息处理系统进展》。

库泰瑞, 刘弗雷德里克, 和何鲁恒。2024。

自动机约束语言模型解码。arXiv 电子预印本。

刘杰森. 2024. 讲师。

斯瓦鲁普·米什拉, 丹尼尔·卡沙比, 奇塔·巴拉尔, 汉纳内赫·哈吉什尔齐。2022。《通过自然语言众包指令实现跨  
任务泛化》。

在 ACL。

莫兰·米兹拉希, 盖伊·卡普兰, 丹·马尔金, 罗特姆·德罗  
尔, 达夫娜·沙哈夫, 加百列·斯坦沃斯基。2024。《艺术  
现状? 呼吁多提示llm评估》。

《计算语言学协会汇刊》第 12 卷第 933-949 页。

OpenAI. 2023. Gpt-4 技术报告。

OpenAI. 2024. Json 模式。https://platform.  
openai.com/docs/guides/text-generation/json-  
mode. 访问于 2024-07-02。

PrefectHQ. 2024. marvin.

秦一伟, 宋凯强, 胡业波, 姚文林, 赵桑武, 王晓阳, 吴  
宣生, 费  
刘鹏飞, 刘, 和董宇。2024。Infobench: 评估大型  
语言模型的指令遵循能力。arXiv 预印本  
arXiv:2401.03601。

梅拉妮·斯克拉尔, 叶金·崔, 尤利亚·采特科夫, 阿兰妮·苏  
尔。2023。《量化语言模型对提示设计中虚假特征的敏  
感性: 或者我是如何开始担心提示格式的》。

在第十二届学习表示国际会议上

阿森·范西·尚戈, 里沙布·戈尔, 魏志, 朱利安·马尔特尔,  
朱奥马纳·戈斯恩。2022。《Ddxplus: 用于自动医学诊  
断的新数据集》。在《.....》(Proceedings.....) 中。  
第 36 届国际神经网络信息处理系统会议论文集, 第  
31306–31318 页。

Anthropic Team. 2024a. 介绍下一代 Claude。  
Gemini Team, Rohan Anil, Sebastian Borgeaud,  
Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu  
Soricut, Johan Schalkwyk, Andrew M Dai, Anja  
Hauth, 等人。2023. Gemini: 一个家族

高度能效的多模态模型。arXiv 预印本  
arXiv:2312.11805。

Gemma 团队, 托马斯·梅斯纳德, 卡西迪·哈丁, 罗伯特  
·达达希, 苏里亚·布帕蒂拉朱, 谢拉·帕塔克, 劳伦特·西  
弗雷, 莫尔甘·里维埃, 米希尔·桑杰·卡莱, 朱丽叶特·洛  
夫, 等。2024。Gemma: 基于 Gemini 研究和技术的  
开源模型。arXiv 预印本 arXiv:2403.08295。

元 LLaMA 团队。2024b. 介绍元 Llama 3: 迄今为止最  
强大的公开可用llm。

王学志, 周邓。2024。《无需提示的链式思维推理》。  
ArXiv, abs/2402.10200。

贾森·魏, 马滕·博斯马, 赵文星, 郭凯文, 魏宇, 布莱恩  
·莱斯特, 杜楠, 戴安德鲁, 吴国越。2021. 微调语言模  
型是零样本学习者。arXiv 预印本 arXiv:2109.01652。

贾森·魏, 王雪志, 戴尔·舒尔曼斯, 马滕·博斯马, 夏  
斐, 爱德·H·蔡, 吴国伟·莱, 周登尼, 等。2022。思维  
链提示引发大型语言模型的推理。载于《神经信息处理  
系统进展》。

布兰登·T·威拉德和雷米·卢夫。2023。大型语言模型的高  
效引导生成。arXiv 电子预印本, 页码 arXiv-2307。

Wu Cheng-Kuang, Tam Zhi Rui, Lin Chieh-Yen, Yun-  
Nung Chen, 和 Hung yi Lee. 2024。  
《Streambench: 迈向语言代理持续改进的基准测  
试》。

夏从英, 陈星, 杜江淑, 杨心怡, 冯一豪, 徐然, 尹文  
鹏, 熊才明。2024。《Fofo: 评估llms格式跟随能力的  
基准》。arXiv 预印本 arXiv:2402.18667。

周杰夫, 陆天健, 米什拉·斯瓦鲁普, 布拉马·悉达多, 巴  
苏·苏乔伊, 刘毅, 周登尼, 侯乐。2023。大型语言模  
型的指令遵循评估。arXiv 预印本 arXiv:2311.07911。  
朱凯杰, 王金东, 周佳恒, 王志晨, 陈浩, 王毅东, 杨  
琳怡, 叶伟, 岳

张, 尼尔·振强·贡, 等。2023. Promptbench: 评估大  
型语言模型在对抗性提示上的鲁棒性。arXiv 预印本  
arXiv:2306.04528。

## 一种局限性

这项研究存在两个主要局限性。首先, 由于成本限  
制, 我们无法在我们的实验中包含 LLaMA 70B 或  
GPT-4o 等更强大的语言模型的结果。这些模型的  
包含可能会提供关于性能如何缩放的额外见解。

Task	Examples	Accuracy (%)
Last Letter	100	97.0
Shuffle Obj	100	96.0
GSM8K	100	100.0
<b>Average</b>	300	97.7

Table 4: Alignment between GPT-4-Turbo and human annotations across different tasks.

with model size and architecture. Second, our evaluation dataset, while diverse, is limited in scope. A broader range of tasks and domains could offer a more comprehensive assessment of the proposed approach’s effectiveness and generalizability.

## B Choosing which LLMs as answer extraction

We first validate if existing LLMs such as *gpt-4-turbo* can be the perfect parser in answer extraction in reasoning tasks such as GSM8K, Last Letter Concatenation. We sampled 300 responses in total: 100 each from Last Letter, Shuffle Object, and GSM8K, each of the responses were independently parsed by human evaluators. We then compared the human-parsed answers with those extracted by GPT-4-turbo. The result shown in Table 4, shows *gpt-4-turbo* can indeed denote as a perfect parser in these 3 cases.

To select the best and low cost answer LLM parser, we select 200 samples from six datasets response in natural language format which a total of 1,200 samples. We then use *gpt-4-turbo* as best LLM answer parser as our reference and calculate the kappa cohen score with 3 LLMs candidates: *gemini-1.5-flash*, *claude-3-haiku-20240307* and *llama-3-8b-instruct* in Figure 7. Result shows *claude-3-haiku-20240307* has the highest agreement with *gpt-4-turbo* at 0.86 followed by *llama-3-8b-instruct*.

## C Cost Comparison Across Different Formats

An important consideration in deploying LLM applications in industry settings is the associated token cost. We analyzed the input and output tokens across our experiments for all models and formats. For brevity, we present the averaged results from all six datasets in Table 5. Our analysis reveals that text and YAML formats generally incur similar costs. Interestingly, we found that YAML is the most cost-effective format for LLaMA-3-8B,

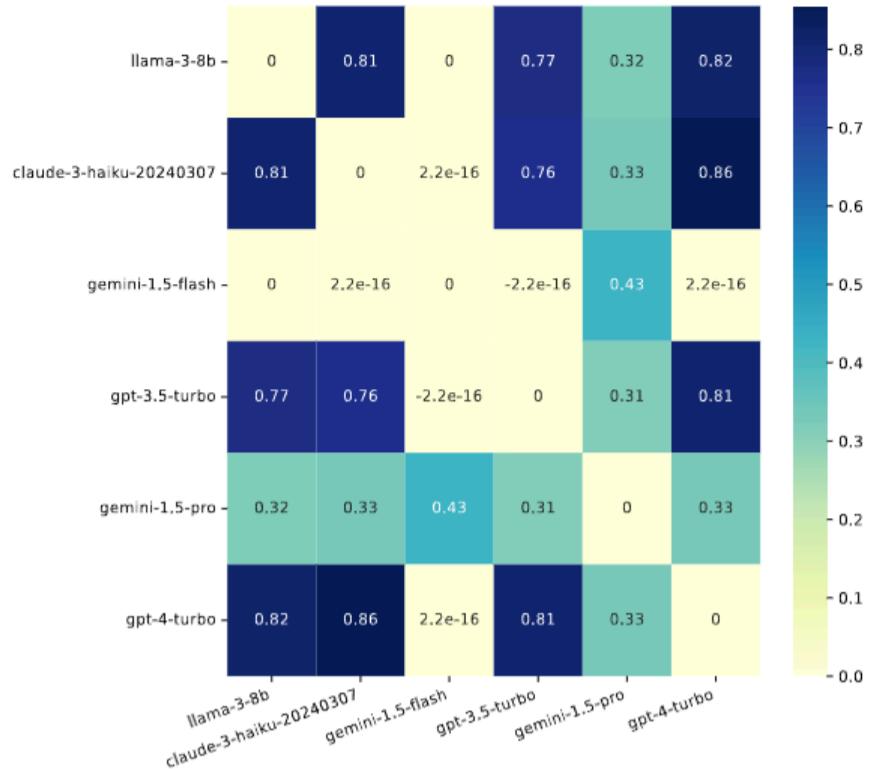


Figure 7: Agreement scores among all LLMs on the final extracted answers.

Model	text	json	xml	yaml
LLaMA-3-8b	0.11	0.09	0.09	0.08
Gemini-1.5-Flash	0.20	0.21	0.21	0.19
Claude-3-Haiku	0.20	0.30	0.30	0.29
GPT-3.5-Turbo	0.35	0.23	0.24	0.23

Table 5: Comparison of total costs (US dollar per 1000 entries) for different models and output formats. Numbers are averaged over all 6 datasets.

Gemini-1.5-Flash, and GPT-3.5-Turbo. Surprisingly, for Claude-3-Haiku, the lowest cost is associated with the text format, which is unexpected given the prevalence of XML examples in their documentation for tool use. The full cost breakdown for each dataset can be found in Table 6, providing a more detailed view for practitioners interested in fine-tuning their approach for specific use cases.

## D Additional models

We also tested additional models from Mistral and OpenAI : *Mistral-7b-v0.3*, *GPT-4o-mini-2024* on format prompt variation in GSM8K, Last Letter, Shuffled Object, Sports Understanding, MultiFin, NL Task 280 and DDXPlus. The result is visualized in Figure 8.

## E Comparison between using regex and LLM as answer parser in GSM8K

To answer the difference between using regex parser to extract the final strict match answer, we calculate the Exact Match score in GSM8K results using the prompt format template "The final answer is". Table 8 results reveal a significant gap

Task	示例	准确率 (%)	
最后信件 100	97.0	洗牌对象 100	96.0
GSM8K 100	100.0		
平均	300	97.7	

表 4: GPT-4-Turbo 与人类标注在不同任务中的对齐情况。

关于模型大小和架构。其次，我们的评估数据集虽然多样，但范围有限。更广泛的任务和领域范围可以提供对所提出方法有效性和泛化能力的更全面评估。

### 选择作为答案提取的LLMs

首先验证现有的LLMs如 gpt-4turbo 是否能在答案提取推理任务中如 GSM8K、最后字母拼接中成为完美的解析器。我们总共采样了 300 个回答：每个任务 100 个，每个回答都由人类评估者独立解析。然后我们比较了人类解析的回答与 GPT-4-turbo 提取的回答。表 4 中的结果显示，gpt-4turbo 确实可以在这 3 个案例中被视为完美的解析器。

为了选择最佳且成本低的LLM解析器，我们从六个以自然语言格式响应的数据集中选取了 200 个样本，总计 1,200 个样本。然后，我们将 gpt-4-turbo 作为最佳LLM解析器作为参考，并与 3 个候选者：gemini-1.5-flash、claude-3-haiku-20240307 和 llama-3-8b-instruct 计算 kappa cohen 分数，如图 7 所示。结果显示，claude-3-haiku-20240307 与 gpt-4-turbo 的吻合度最高，为 0.86，其次是 llama-38b-instruct。

### C 不同格式的成本比较

在工业环境中部署LLM应用程序时，一个重要的考虑因素是相关的代币成本。我们分析了所有模型和格式的输入和输出代币。为了简洁，我们在表 5 中展示了所有六个数据集的平均结果。我们的分析显示，文本和 YAML 格式通常产生相似的成本。有趣的是，我们发现 YAML 是 LLaMA-3-8B 最经济的格式。

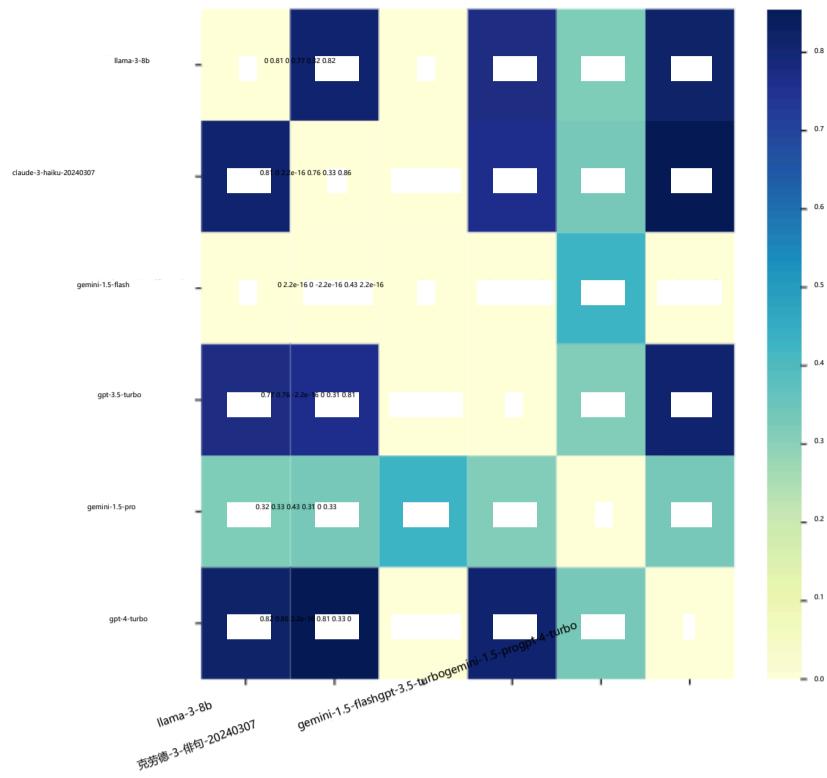


图 7: 所有LLMs在最终提取的答案中的协议得分。

模型	文本	JSON	XML	YAML
LLaMA-3-8b	0.11	0.09	0.09	0.08
Gemini-1.5-Flash	0.20	0.21	0.21	0.19
Claude-3-Haiku	0.20	0.30	0.30	0.29
GPT-3.5-Turbo	0.35	0.23	0.24	0.23

表 5: 不同模型和输出格式的总成本（每 1000 条记录的美元）比较。数值为所有 6 个数据集的平均值。

Gemini-1.5-Flash 和 GPT-3.5-Turbo。令人惊讶的是，对于 Claude-3-Haiku，最低成本与文本格式相关联，这在他们的工具使用文档中 XML 示例普遍存在的情况下是出乎意料的。每个数据集的完整成本分解可在表 6 中找到，为对特定用例进行方法微调感兴趣的实践者提供了更详细的视角。

### D 其他模型

我们还在 Mistral 测试了额外的模型 OpenAI: Mistral-7b-v0.3, GPT-4o-mini-2024 GSM8K 格式提示变化，最后一个字母，打乱的对象，体育理解，MultiFin，NL 任务 280 和 DDXPlus。结果如图 8 所示。

### E 使用正则表达式与LLM作为GSM8K答案解析器的比较

为了回答使用正则表达式解析器提取最终严格匹配答案的差异，我们使用提示格式模板“最终答案是”计算 GSM8K 结果中的精确匹配得分。表 8 的结果揭示了一个显著的差距

Dataset	Format	gemini-1.5-flash			llama-3-8b			claude-3-haiku			gpt-3.5-turbo		
		In	Out	Tot	In	Out	Tot	In	Out	Tot	In	Out	Tot
lastletter	text	0.04	0.09	0.12	0.02	0.02	0.04	0.03	0.12	0.15	0.05	0.07	0.12
	json	0.04	0.10	0.14	0.02	0.03	0.05	0.03	0.17	0.21	0.06	0.05	0.11
	xml	0.04	0.10	0.14	0.02	0.03	0.05	0.03	0.15	0.18	0.06	0.07	0.13
	yaml	0.04	0.09	0.13	0.02	0.02	0.05	0.03	0.14	0.18	0.06	0.09	0.14
gsm8k	text	0.05	0.13	0.18	0.03	0.03	0.06	0.04	0.23	0.27	0.07	0.16	0.23
	json	0.05	0.14	0.20	0.03	0.03	0.07	0.04	0.29	0.33	0.08	0.12	0.19
	xml	0.06	0.14	0.19	0.03	0.03	0.07	0.05	0.27	0.32	0.08	0.12	0.20
	yaml	0.05	0.13	0.18	0.03	0.03	0.06	0.04	0.28	0.32	0.08	0.14	0.22
multifin	text	0.05	0.01	0.06	0.03	0.00	0.03	0.03	0.02	0.05	0.07	0.02	0.09
	json	0.05	0.02	0.07	0.03	0.00	0.03	0.04	0.05	0.09	0.07	0.02	0.09
	xml	0.05	0.02	0.07	0.03	0.01	0.04	0.04	0.04	0.08	0.08	0.03	0.10
	yaml	0.05	0.01	0.06	0.03	0.00	0.03	0.04	0.02	0.06	0.07	0.01	0.08
sports	text	0.04	0.04	0.08	0.02	0.01	0.03	0.03	0.10	0.13	0.05	0.05	0.10
	json	0.04	0.06	0.10	0.02	0.01	0.04	0.03	0.11	0.15	0.06	0.07	0.12
	xml	0.04	0.07	0.11	0.02	0.02	0.04	0.03	0.14	0.17	0.06	0.08	0.14
	yaml	0.04	0.05	0.08	0.02	0.01	0.04	0.03	0.12	0.15	0.05	0.06	0.11
task280	text	0.04	0.05	0.09	0.03	0.01	0.03	0.03	0.05	0.08	0.06	0.04	0.11
	json	0.05	0.04	0.08	0.03	0.01	0.03	0.04	0.07	0.11	0.07	0.04	0.11
	xml	0.05	0.04	0.09	0.03	0.01	0.04	0.04	0.08	0.11	0.07	0.05	0.12
	yaml	0.04	0.03	0.07	0.03	0.01	0.03	0.04	0.05	0.09	0.06	0.03	0.10
ddxplus	text	0.26	0.15	0.41	0.15	0.04	0.18	0.19	0.20	0.38	0.38	0.21	0.59
	json	0.22	0.18	0.41	0.13	0.06	0.19	0.19	0.33	0.52	0.34	0.15	0.48
	xml	0.23	0.19	0.42	0.14	0.06	0.19	0.19	0.37	0.56	0.34	0.18	0.51
	yaml	0.22	0.15	0.37	0.13	0.05	0.18	0.19	0.31	0.50	0.33	0.15	0.48

Table 6: Performance comparison of different models across various datasets and formats. Values represent processing times in seconds for Input (In), Output (Out), and Total (Tot).

between regex match and LLM as final answer parser in EM score across various language models, highlighting the limitations of using only one strict regex matching for different models. For example, GPT-3.5-Turbo shows a 31.8 percentage point improvement from regex match (43.7%) to overall accuracy (75.5%), while Gemini-1.5-Flash exhibits an even larger 43.5 point difference. This pattern is consistent across all models, with mistral-7b demonstrating the most dramatic 42 point increase.

These disparities underscore the value of using LLMs as answer parsers, as they can understand and evaluate responses beyond literal string matching, accounting for paraphrases and contextual understanding, thus providing a more nuanced and accurate assessment in text-based tasks.

Just to be safe we also assess the reliability of GPT-4-turbo as a parser, we conducted a manual validation study:

- We sampled 300 responses in total: 100 each from Last Letter, Shuffle Object, and GSM8K
- These responses were independently parsed by human evaluators.

- We then compared the human-parsed answers with those extracted by GPT-4-turbo.

The results of this validation are shown in Table 7. These findings demonstrate an average alignment of 97.7% between GPT-4-turbo and human-parsed answers, supporting our characterization of GPT-4-turbo as a near-perfect parser for this task.

Task	GPT-4-Turbo correctness
Last Letter	97/100
Shuffle Obj	96/100
GSM8K	100/100

Table 7: Alignment between GPT-4-turbo and human-parsed answers. In general we found GPT-4-turbo is very close to perfect parser which serves as a versatile parser to all kinds of task.

## F Averaged numbers for all datasets

### F.1 Zero shot prompting comparing Text, JSON, XML, YAML

Table (10, 9) shows all the number with standard deviation on all 4 format (NL, JSON, XML, YAML) in classification and reasoning tasks.

	gemini-1.5-flash	llama-3-8b	克劳德-3-俳句	gpt-3.5-turbo										
数据集格式	输入 文本 yaml	输出 0.04 0.09 0.13 总计 0.12 0.02 0.02 0.04 0.03 0.12 0.15 0.05 0.07 0.12 json 0.04 0.10 0.14 0.02 0.03 0.05 0.03 0.17 0.21 0.06 0.05 0.11 xml 0.04 0.10 0.14 0.02 0.03 0.05 0.03 0.15 0.18 0.06 0.07 0.13 yaml 0.04 0.09 0.13 0.02 0.02 0.05 0.03 0.14 0.18 0.06 0.09 0.14	输入 文本 yaml	输出 0.05 0.13 0.18 0.03 0.03 0.06 0.04 0.23 0.27 0.07 0.16 0.23 json 0.05 0.14 0.20 0.03 0.03 0.07 0.04 0.29 0.33 0.08 0.12 0.19 xml 0.06 0.14 0.19 0.03 0.03 0.07 0.05 0.27 0.32 0.08 0.12 0.20 yaml 0.05 0.13 0.18 0.03 0.03 0.06 0.04 0.28 0.32 0.08 0.14 0.22	输入 文本 yaml	输出 0.05 0.13 0.18 0.03 0.03 0.06 0.04 0.23 0.27 0.07 0.16 0.23 json 0.05 0.14 0.20 0.03 0.03 0.07 0.04 0.29 0.33 0.08 0.12 0.19 xml 0.06 0.14 0.19 0.03 0.03 0.07 0.05 0.27 0.32 0.08 0.12 0.20 yaml 0.05 0.13 0.18 0.03 0.03 0.06 0.04 0.28 0.32 0.08 0.14 0.22	输入 文本 yaml	输出 0.05 0.01 0.06 0.03 0.00 0.03 0.03 0.02 0.05 0.07 0.02 0.09 json 0.05 0.02 0.07 0.03 0.00 0.03 0.04 0.05 0.09 0.07 0.02 0.09 xml 0.05 0.02 0.07 0.03 0.01 0.04 0.04 0.04 0.08 0.08 0.03 0.10 yaml 0.05 0.01 0.06 0.03 0.00 0.03 0.04 0.02 0.06 0.07 0.01 0.08	输入 文本 yaml	输出 0.04 0.04 0.08 0.02 0.01 0.03 0.03 0.10 0.13 0.05 0.05 0.10 json 0.04 0.06 0.10 0.02 0.01 0.03 0.04 0.11 0.15 0.06 0.07 0.12 xml 0.04 0.07 0.11 0.02 0.02 0.04 0.03 0.14 0.17 0.06 0.08 0.14 yaml 0.04 0.05 0.08 0.02 0.01 0.04 0.03 0.12 0.15 0.05 0.06 0.11	输入 文本 yaml	输出 0.04 0.05 0.09 0.03 0.01 0.03 0.03 0.05 0.08 0.06 0.04 0.11 json 0.05 0.04 0.08 0.03 0.01 0.03 0.04 0.11 0.15 0.07 0.04 0.11 xml 0.05 0.04 0.09 0.03 0.01 0.04 0.04 0.08 0.11 0.07 0.05 0.12 yaml 0.04 0.03 0.07 0.03 0.01 0.03 0.04 0.05 0.09 0.06 0.03 0.10	输入 文本 yaml	输出 0.26 0.15 0.41 0.15 0.04 0.18 0.19 0.20 0.38 0.38 0.21 0.59 json 0.22 0.18 0.41 0.13 0.06 0.19 0.19 0.33 0.52 0.34 0.15 0.48 xml 0.23 0.19 0.42 0.14 0.06 0.19 0.19 0.37 0.56 0.34 0.18 0.51 yaml 0.22 0.15 0.37 0.13 0.05 0.18 0.19 0.31 0.50 0.33 0.15 0.48

表 6：不同模型在不同数据集和格式下的性能比较。数值表示输入 (In)、输出 (Out) 和总计 (Tot) 的处理时间 (秒)。

在 EM 分数中，将正则表达式匹配与 LLM 作为最终答案解析器应用于各种语言模型之间，突显了仅使用严格正则表达式匹配对不同模型的局限性。例如，GPT-3.5-Turbo 从正则表达式匹配（43.7%）到整体准确率（75.5%）提高了 31.8 个百分点，而 Gemini-1.5-Flash 则表现出更大的 43.5 点差异。这种模式在所有模型中都是一致的，其中 mistral7b 展示了最显著的 42 点增长。

这些差异凸显了使用LLMs作为答案解析器的重要性，因为它们能够理解并评估超出字面字符串匹配的回应，考虑到释义和语境理解，从而在基于文本的任务中提供更细腻和准确的评估。

- 总共采样 300 个回答：每个实验 100 个，分别来自 Last Letter、Shuffle Object 和 GSM8K
  - 这些回应由人类评估者独立解析。

- 然后，我们将人工解析的答案与 GPT-4-turbo 提取的答案进行了比较。

结果如表 7 所示。这些发现表明，GPT-4-turbo 与人工解析答案的平均一致性为 97.7%，支持我们将 GPT-4-turbo 描述为这项任务中近乎完美的解析器。

任务 GPT-4-Turbo 正确性 最后一个字母 97/100 洗牌 Obj 96/100 GSM8K 100/100

表 7: GPT-4-turbo 与人类解析答案的对应关系。总体而言，我们发现 GPT-4-turbo 非常接近完美的解析器，它作为一种通用的解析器适用于各种任务。

为确保安全，我们还评估了 GPT-4-turbo 作为解析器的可靠性，我们进行了一项手动验证研究：

## 所有数据集的平均数

## F.1 零样本提示比较文本、JSON、XML、YAML

表 (10, 9) 显示了在分类和推理任务中所有 4 种格式 (NL、JSON、XML、YAML) 的所有数值及其标准差。

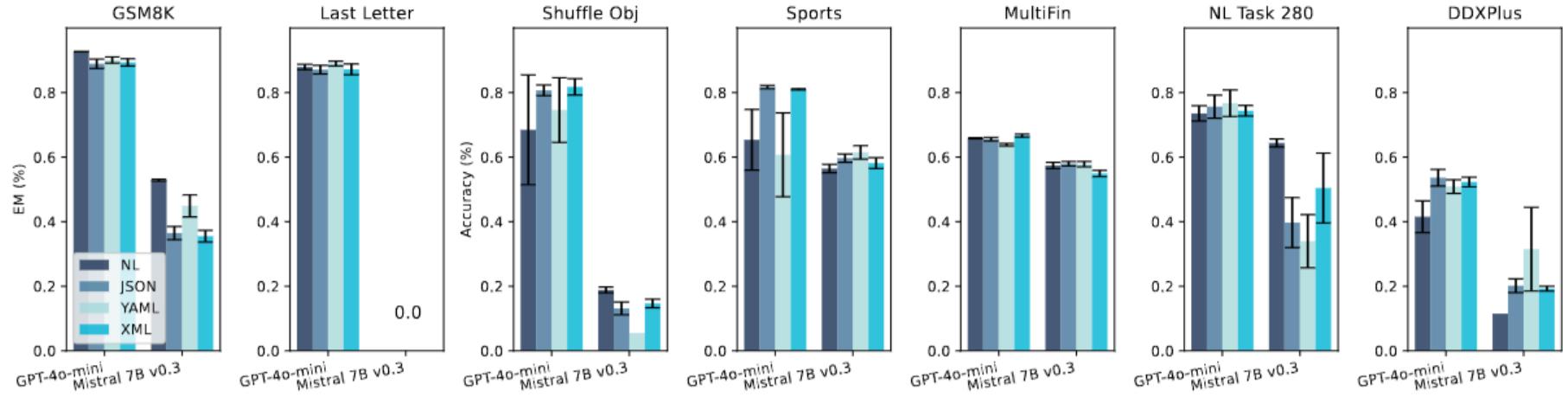


Figure 8: Exact Match scores on GSM8K and Last Letter on reasoning related datasets. Classification related tasks on Shuffled Object, Sports Understanding, MultiFin, NL Task 280 and DDXPlus in different levels of format restriction. In general, we found GPT-4o is quite consistent on adding format restriction. In the Last Letter task, the exact match scores of Mistral-7B-v0.3 across all 4 formats are very close to 0%, which are not explicitly shown in the figure.

Model	Regex Match	LLM Match
GPT-3.5-Turbo	43.7	75.5
Gemini-1.5-Flash	25.8	69.3
Claude-3-Haiku	67.4	85.8
Gemma2-9b	82.5	86.0
LLaMA-3-8b	46.9	55.7
Mistral-7b-v0.3	10.4	52.4

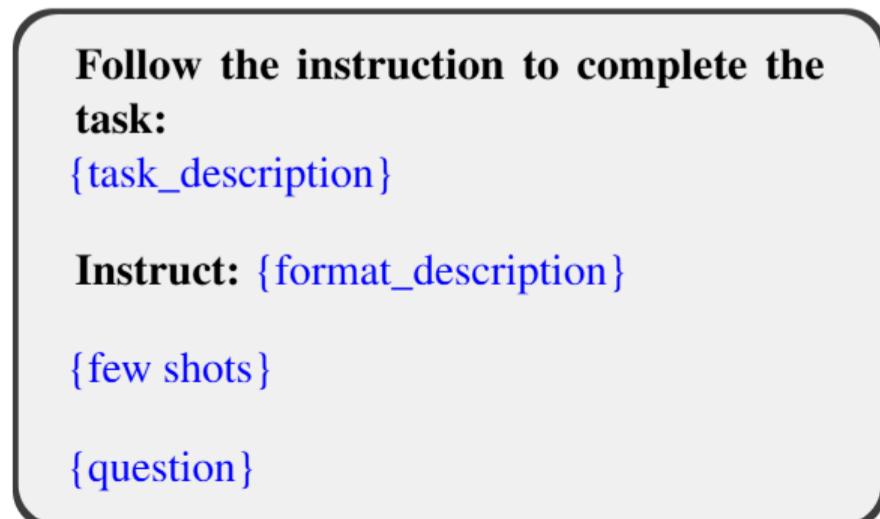
Table 8: Comparison of model performance on regex match "*The final answer is (\d+)*" accuracy and using Claude-3-Haiku as answer parser.

The JSON-mode scores for GPT 3.5 turbo, Gemini 1.5 Flash and LLaMA 3 8B are presented in Table 11. This table shows the performance of these three models on six different datasets when using JSON-mode.

## G Prompt

### G.1 Prompt Format

For each task we fix the same template and only swapping the task description, format description, few shots example and question text.



**Task Description** A task description describes the task and the final goal of the task.

**Format Description** A format description includes the target format (ie JSON, XML or YAML) and

Table 9: Zero shot prompting results for gemini-1.5-flash, gpt-3.5-turbo, claude-3-haiku, llama-3-8B, and gemma2-9B-IT averaged on 3 reasoning tasks with standard deviation in reasoning related task.

	Last Letter	GSM8K	ShuffleObj
Gemini-1.5-Flash			
Text	65.4 (3.1)	89.3 (0.8)	58.2 (13.0)
JSON	77.0 (7.3)	89.2 (1.5)	65.1 (5.3)
XML	74.2 (10.4)	88.2 (2.2)	50.4 (10.5)
YAML	71.4 (20.3)	87.4 (3.7)	34.3 (17.1)
GPT-3.5 Turbo			
Text	56.7 (7.1)	76.6 (2.8)	20.4 (3.6)
JSON	25.2 (29.1)	49.3 (12.0)	20.9 (1.1)
XML	22.3 (27.8)	45.1 (19.9)	15.4 (1.8)
YAML	66.9 (22.0)	73.9 (5.6)	20.8 (1.3)
Claude 3 Haiku			
Text	57.7 (21.1)	86.5 (0.8)	36.6 (8.2)
JSON	56.7 (16.7)	23.4 (22.8)	49.3 (4.8)
XML	33.8 (31.5)	79.8 (7.0)	48.1 (5.2)
YAML	31.6 (32.4)	80.6 (2.8)	18.1 (14.7)
LLaMA 3 8B			
Text	70.1 (5.3)	74.7 (0.6)	27.0 (5.5)
JSON	28.0 (12.2)	48.9 (6.7)	15.7 (11.0)
XML	15.9 (4.8)	56.7 (8.3)	11.1 (3.6)
YAML	16.1 (10.4)	46.1 (16.8)	9.6 (3.6)
Gemma2 9B IT			
Text	56.8 (9.8)	86.5 (0.6)	49.4 (5.8)
JSON	39.0 (6.8)	84.2 (3.7)	50.5 (8.9)
XML	43.7 (3.8)	85.6 (0.6)	38.5 (5.0)
YAML	23.4 (15.7)	79.5 (4.1)	23.0 (16.4)

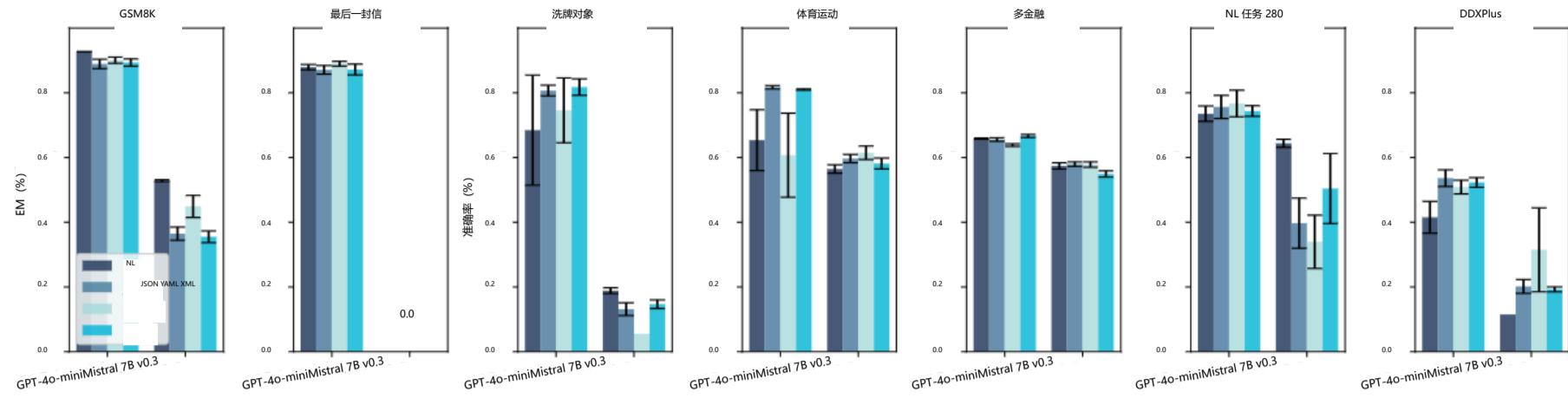


图 8: GSM8K 和推理相关数据集上的精确匹配分数以及最后字母。在 Shuffled Object、Sports Understanding、MultiFin、NL Task 280 和 DDXPlus 的不同格式限制级别上的分类相关任务。总的来说，我们发现 GPT-4o 在添加格式限制方面相当一致。在最后字母任务中，Mistral-7B-v0.3 在所有 4 种格式上的精确匹配分数都非常接近 0%，这在图中没有明确显示。

模型 正则匹配 LLM 匹配	
GPT-3.5-Turbo	43.7 75.5 Gemini-1.5-Flash
25.8 69.3 Claude-3-Haiku	67.4 85.8 Gemma2-9b 82.5 86.0 LLaMA-3-8b 46.9 55.7 Mistral-7b-v0.3 10.4 52.4

表 8: 模型在正则表达式匹配 "The final answer is (\d+)" 准确率和使用 Claude-3-Haiku 作为答案解析器上的性能比较。

表 11 展示了 GPT 3.5 turbo、Gemini 1.5 Flash 和 LLaMA 3 8B 在 JSON 模式下的分数。该表显示了这三个模型在六个不同数据集上使用 JSON 模式时的性能。

## G 提示

### G.1 提示格式

对于每个任务，我们固定相同的模板，仅交换任务描述、格式描述、少量示例和问题文本。

遵循指示完成任务：

{task\_description}

指令：{格式描述}

{少量样本}

{问题}

任务描述 任务描述用于描述任务及其最终目标。

格式描述 格式描述包括目标格式（例如 JSON

XML 或 YAML）和表 9: gemini-1.5flash、gpt-3.5-turbo、claude-3-haiku、llama-3-8B 和 gemma2-9B-IT 在 3 个推理任务上的零样本提示结果及推理相关任务的平均值和标准差。

最后一封信 GSM8K ShuffleObj	
Gemini-1.5-Flash	
文本	65.4 (3.1) 89.3 (0.8) 58.2 (13.0) JSON 77.0 (7.3)
	89.2 (1.5) 65.1 (5.3) XML 74.2 (10.4) 88.2 (2.2) 50.4
	(10.5) YAML 71.4 (20.3) 87.4 (3.7) 34.3 (17.1)
GPT-3.5 Turbo	
文本	56.7 (7.1) 76.6 (2.8) 20.4 (3.6) JSON 25.2
	(29.1) 49.3 (12.0) 20.9 (1.1) XML 22.3 (27.8) 45.1
	(19.9) 15.4 (1.8) YAML 66.9 (22.0) 73.9 (5.6) 20.8
	(1.3)
克劳德 3 俳句	
文本	57.7 (21.1) 86.5 (0.8) 36.6 (8.2) JSON 56.7
	(16.7) 23.4 (22.8) 49.3 (4.8) XML 33.8 (31.5) 79.8
	(7.0) 48.1 (5.2) YAML 31.6 (32.4) 80.6 (2.8) 18.1 (14.7)
LLaMA 3 8B	
翻译文本	0.1 LLaMA 3 8B (0.6) 27.0 (5.5) JSON 28.0
	(12.2) 48.9 (6.7) 15.7 (11.0) XML 15.9 (4.8)
	56.7 (8.3) 11.1 (3.6) YAML 16.1 (10.4) 46.1
	(16.8) 9.6 (3.6)
Gemma2 9B IT	
文本	56.8 (9.8) 86.5 (0.6) 49.4 (5.8) JSON 39.0 (6.8)
	84.2 (3.7) 50.5 (8.9) XML 43.7 (3.8) 85.6 (0.6) 38.5
	(5.0) YAML 23.4 (15.7) 79.5 (4.1) 23.0 (16.4)

Table 10: Zero shot prompting results for gemini-1.5-flash, gpt-3.5-turbo, claude-3-haiku, llama-3-8B, and gemma2-9B-IT averaged on 4 classification tasks with standard deviation in classification related task

	DDXPlus	Sports	Task280	MultiFin
Gemini-1.5-Flash				
Text	41.6 (6.6)	79.9 (3.2)	68.6 (2.5)	63.5 (0.3)
JSON	60.3 (0.8)	78.9 (1.3)	70.3 (5.4)	65.2 (1.1)
XML	59.4 (1.4)	80.2 (0.7)	70.0 (4.9)	64.5 (1.6)
YAML	60.4 (1.6)	80.1 (1.2)	65.3 (12.7)	64.1 (0.4)
GPT-3.5 Turbo				
Text	44.1 (3.2)	67.2 (26.8)	72.7 (6.3)	63.0 (0.5)
JSON	55.5 (0.4)	80.0 (3.3)	70.6 (11.2)	64.0 (0.9)
XML	53.0 (1.4)	80.7 (1.1)	66.2 (16.2)	62.2 (1.1)
YAML	55.0 (0.8)	80.9 (2.3)	72.1 (8.0)	65.4 (0.9)
Claude 3 Haiku				
Text	33.8 (13.5)	77.8 (5.8)	61.1 (11.0)	62.0 (1.9)
JSON	52.0 (1.1)	78.7 (2.8)	49.5 (27.2)	63.7 (1.3)
XML	50.8 (0.8)	77.8 (3.8)	45.0 (25.0)	62.4 (1.1)
YAML	6.9 (5.3)	76.4 (8.3)	44.5 (24.2)	61.8 (1.7)
LLaMA 3 8B				
Text	12.04 (15.2)	69.49 (12.7)	65.28 (3.4)	60.26 (1.4)
JSON	23.37 (0.7)	73.38 (3.5)	39.46 (22.4)	57.74 (2.0)
XML	11.35 (1.9)	69.20 (5.5)	35.36 (22.5)	58.77 (3.2)
YAML	13.08 (4.1)	68.25 (5.7)	45.42 (24.4)	49.74 (4.2)
Gemma2 9B IT				
Text	22.9 (5.8)	76.1 (2.3)	69.8 (7.7)	70.0 (0.4)
JSON	53.0 (0.2)	72.7 (1.6)	65.6 (11.7)	70.2 (0.7)
XML	52.9 (2.8)	73.3 (2.4)	68.1 (11.7)	68.0 (0.7)
YAML	44.9 (2.2)	73.0 (1.7)	60.5 (11.0)	69.8 (0.7)

Dataset	GPT3.5T	Gemini1.5F	LLaMA3 8B
LastLetter	1.78 (0.3)	0.67 (0.5)	<b>7.56 (2.7)</b>
GSM8K	29.87 (0.8)	47.78 (3.1)	<b>65.38 (1.3)</b>
MultiFin	66.00 (1.3)	<b>66.79 (0.4)</b>	54.82 (1.5)
Sports	76.82 (0.9)	<b>77.79 (0.4)</b>	72.08 (2.6)
Task 280	<b>78.07 (2.3)</b>	67.19 (4.1)	74.57 (2.0)
DDXPlus	51.87 (2.8)	<b>84.92 (2.1)</b>	22.59 (0.1)

Table 11: Averaged scores for JSON-mode to all 6 datasets, performance varies significantly across tasks and models, suggesting that different models may have strengths in different areas when using JSON-mode.

a targeted schema we intend the LLM response to adhere to.

For each description slot, we create 3 variations each which results in 9 prompt combinations. Each variation must retain the original meaning with slight change in wording, order of instruction. For each model we prompt all 9 prompts to calculate the sensitivity and variance of the final result.

If the current task requires reasoning, we include the zero shot chain-of-thought prompting : "Think step-by-step" in task description and ensures the LLM response to generate reasoning before giving the final answer.

## G.2 Prompt Variations

Our study employs a range of prompt variations across multiple tasks to assess the robustness and generalizability of language models. We developed three distinct task description variations for each of the following datasets:

- GSM8K (Figure 9)
- Last Letter (Figure 10)
- Shuffle Object (Figure 11)
- DDXPlus (Figure 12)
- Sports Understanding (Figure 13)
- Natural Language - Task 280 (Figure 14)
- MultiFin (Figure 15)

For tasks involving chain-of-thought reasoning (GSM8K, Last Letter, Shuffle Object Tracking, DDXPlus, Sports Understanding, and NL-Task 280), we implemented three prompt format variations. These are illustrated in Figures 19, 20, and 21.

Additionally, we created three answering format variations for both reasoning-based tasks and those

表 10: gemini-1.5-flash、gpt-3.5-turbo、claude-3-haiku、llama-3-8B 和 gemma29B-IT 在 4 个分类任务上的零样本提示结果及分类相关任务的标准差

DDXPlus 体育任务 280 多鳍								
<b>Gemini-1.5-Flash</b>								
文本	41.6 (6.6)	79.9 (3.2)	68.6 (2.5)	63.5 (0.3)	JSON	60.3 (0.8)	78.9 (1.3)	70.3 (5.4)
					XML	59.4 (1.4)	80.2 (0.7)	70.0 (4.9)
					YAML	60.4 (1.6)	80.1 (1.2)	65.3 (12.7)
						64.1 (0.4)		
<b>GPT-3.5 Turbo</b>								
文本	44.1 (3.2)	67.2 (26.8)	72.7 (6.3)	63.0 (0.5)	JSON	55.5 (0.4)	80.0 (3.3)	70.6 (11.2)
					XML	53.0 (1.4)	80.7 (1.1)	66.2 (16.2)
					YAML	55.0 (0.8)	80.9 (2.3)	72.1 (8.0)
						65.4 (0.9)		
<b>克劳德 3 俳句</b>								
文本	33.8 (13.5)	77.8 (5.8)	61.1 (11.0)	62.0 (1.9)	JSON	52.0 (1.1)		
					XML	50.8 (0.8)	77.8 (3.8)	45.0
					YAML	6.9 (5.3)	76.4 (8.3)	44.5 (24.2)
						61.8 (1.7)		
<b>LLaMA 3 8B</b>								
翻译文本	12.04 (1.15)	44.23 (0.89)	49 (12.7)	65.28 (3.4)	60.26 (1.4)	JSON	23.37 (0.7)	73.38 (3.5)
					XML	11.35 (1.9)	69.20 (5.5)	39.46 (22.4)
					YAML	13.08 (4.1)	68.25 (5.7)	57.74 (2.0)
						45.42 (24.4)		35.36 (22.5)
						49.74 (4.2)		
<b>Gemma2 9B IT</b>								
文本	22.9 (5.8)	76.1 (2.3)	69.8 (7.7)	70.0 (0.4)	JSON	53.0 (0.2)	72.7 (1.6)	
					XML	52.9 (2.8)	73.3 (2.4)	68.1 (11.7)
					YAML	44.9 (2.2)	73.0 (1.7)	60.5 (11.0)
						69.8 (0.7)		

数据集 GPT3.5T Gemini1.5F LLaMA3 8B
最后字母 1.78 (0.3) 0.67 (0.5) 7.56 (2.7) GSM8K 29.87 (0.8) 47.78 (3.1) 65.38 (1.3)
多芬 66.00 (1.3) 66.79 (0.4) 54.82 (1.5) 体育 76.82 (0.9) 77.79 (0.4) 72.08 (2.6) 任务 280 78.07 (2.3) 67.19 (4.1) 74.57 (2.0) DDXPlus 51.87 (2.8) 84.92 (2.1) 22.59 (0.1)

表 11: JSON 模式对所有 6 个数据集的平均分数，性能在任务和模型之间差异显著，表明在使用 JSON 模式时，不同的模型可能在不同的领域具有优势。

一个我们希望LLM响应遵守的目标架构。对于每个描述槽，我们创建 3 个变体，共产生 9 种提示组合。每个变体必须保留原始意义，同时略有措辞和指令顺序的改变。对于每个模型，我们使用所有 9 个提示来计算最终结果的敏感性和方差。如果当前任务需要推理，我们在任务描述中包含零样本思维链提示：“逐步思考”，并确保在给出最终答案之前，LLM响应生成推理。

## G.2 提示变异

我们的研究在多个任务中采用了一系列提示变体，以评估语言模型的鲁棒性和泛化能力。我们为以下数据集开发了三种不同的任务描述变体：

- GSM8K (图 9)
- 最后的信件 (图 10)
- 打乱对象 (图 11)
- DDXPlus (图 12)
- 体育理解 (图 13)
- 自然语言 - 任务 280 (图 14)
- 多功能 (图 15)

对于涉及思维链推理的任务 (GSM8K、Last Letter、Shuffle Object Tracking、DDXPlus、Sports Understanding 和 NL-Task 280)，我们实现了三种提示格式变体。这些变体在图 19、20 和 21 中展示。此外，我们还为基于推理的任务和那些任务创建了三种回答格式变体。

requiring direct answers. These "direct answer prompts" are presented in Figures 16, 17, and 18.

**Task description variation1:**

You are a math tutor who helps students of all levels understand and solve mathematical problems.

Read the last question carefully and think step by step before answering, the final answer must be only a number.

**Task description variation2:**

Read the last question carefully and think step by step before answering, the final answer must be only a number. You are a math tutor who helps students of all levels understand and solve mathematical problems.

**Task description variation3:**

Mathematical problem-solving task:

- Given: A mathematical question or problem
- Required: A numerical answer only
- Role: You are a math tutor assisting students of all levels
- Process: Think step by step to solve the problem

Note: Read the question carefully before beginning your analysis.

Figure 9: GSM8K Task Description Variations

**Task description variation1:**

You are given a string of words and you need to take the last letter of each words and concate them.

Read the last question carefully and think step by step before answering.

**Task description variation2:**

Read carefully for each of the last question and think step by step before answering. You are given a string of words and you need to take the last letter of each words and concatenate them.

**Task description variation3:**

String manipulation task:

- Given: A sequence of words
- Required: A new string made from the last letter of each word
- Process: Think step by step to solve this challenge

Note: Ensure you've read the question thoroughly before beginning.

Figure 10: Last Letter Task Description Variations

我们为基于推理的任务和需要直接回答的任务创建了三种回答格式变体。这些“直接回答提示”在图 16、17 和 18 中展示。

**任务描述变体 1:**

您是一名数学家教，帮助所有水平的学生理解和解决数学问题。仔细阅读最后一题，在回答之前一步一步思考，最终答案必须仅是一个数字。

**任务描述变体 2:**

仔细阅读最后一个问题，在回答之前逐步思考，最终答案必须仅是一个数字。你是一名数学辅导老师，帮助所有水平的学生理解和解决数学问题。

**任务描述变体 3:**

**数学问题解决任务：** • 已知：一个数学问题或题目 • 要求：仅提供一个数值答案 • 角色：您是一位辅导所有水平学生的数学导师 • 流程：逐步思考解决问题 • 注意：在开始分析之前仔细阅读问题。

**任务描述变体 1:**

您被给出一个单词序列，需要取出每个单词的最后一个字母并将它们拼接起来。仔细阅读最后一个问题，在回答之前逐步思考。

**任务描述变体 2:**

仔细阅读最后一个问题，并在回答前逐步思考。你将得到一串单词，需要取每个单词的最后一个字母并将它们连接起来。

**任务描述变体 3:**

**字符串操作任务：** • 给定：一系列单词 • 需求：由每个单词的最后一个字母组成的新字符串 • 流程：逐步思考以解决这个挑战 注意：在开始之前务必仔细阅读问题。

图 9: GSM8K 任务描述变化

图 10: 最后字母任务描述变化

**Task description variation1:**

In this task, you are tasked to answer the following commonsense knowledge task.  
Read carefully for each of the last question and think step by step before answering.  
Make sure the answer only contain one of these four choice : A, B, C, D, E, F, G

**Task description variation2:**

Read carefully for each of the last question and think step by step before answering.  
Make sure the answer only contain one of these four choice : A, B, C, D, E, F, G  
In this task, you are tasked to answer the following commonsense knowledge task.

**Task description variation3:**

Context understanding assessment:

- Given: A story related to many person in the same place
- Required: Determine if the person who is in the end of the story
- Process: Think step by step to analyze the context
- Output: Answer the correct answer and only contain one of these seven choice : A, B, C, D, E, F, G

**Task description variation1:**

Extract the following RESPONSE final answer, your answer should be the one which match any of these valid diagnoses:

- Possible NSTEMI / STEMI
- Spontaneous rib fracture
- Pulmonary embolism
- Pulmonary neoplasm

...

- Scombroid food poisoning

RESPONSE:

**Task description variation2:**

Act as a medical doctor and diagnose the patient based on the given patient profile  
All possible valid diagnoses for you to choose from are as follows:

- Possible NSTEMI / STEMI
- Spontaneous rib fracture
- Pulmonary embolism
- Pulmonary neoplasm

...

- Scombroid food poisoning

**Task description variation3:**

Medical diagnosis task:

- Given: A patient profile
- Required: Diagnose the patient based on the provided information
- Process: Think step by step to analyze the patient's symptoms and history
- Output: Select one diagnosis from the provided list of valid options

Note: Carefully review the patient profile and the list of possible diagnoses before making your determination. Do not answer "Insufficient information" - you must choose from the given options.

Valid diagnoses (select one):

- Possible NSTEMI / STEMI
- Spontaneous rib fracture
- Pulmonary embolism
- Pulmonary neoplasm

...

- Scombroid food poisoning

Figure 11: Shuffle object Task Description Variations

Figure 12: DDXPlus Task Description Variations

### 任务描述变体 1:

在这个任务中，你被要求回答以下常识知识问题。仔细阅读每个最后一个问题，并在回答前逐步思考。确保答案只包含以下四个选项之一：A、B、C、D、E、F、G

任务描述变体 2：仔细阅读每个最后一个问题，并在回答前逐步思考。确保答案只包含以下四个选项之一：A、B、C、D、E、F、G

在这个任务中，你被要求回答以下常识知识问题。

### 任务描述变体 3:

情境理解评估：• 给定：一个与同一地点许多人物相关的故事 • 要求：确定故事结尾的人物 • 过程：逐步思考以分析情境 • 输出：给出正确答案，且只包含以下七个选项之一：A、B、C、D、E、F、G

### 任务描述变体 1：

提取以下响应的最终答案，你的答案应与以下任何有效诊断相匹配：

- 可能的非 ST 段抬高型心肌梗死/ST 段抬高型心肌梗死  
自发性肋骨骨折  
肺栓塞  
肺肿瘤
- ...
- 鲑鱼中毒反应：

### 任务描述变体 2：

扮演一名医生，根据提供的患者档案对患者进行诊断。您可以选择的所有可能的诊断如下：

- 可能的非 ST 段抬高型心肌梗死/ST 段抬高型心肌梗死  
自发性肋骨骨折  
肺栓塞  
肺肿瘤
- ...
- 鲑中毒食物中毒任务描述变体 3：

医学诊断任务：• 给定：患者档案 • 要求：根据提供的信息对患者进行诊断 • 流程：逐步思考分析患者的症状和病史 • 输出：从提供的有效选项列表中选择一个诊断结果 注意：在做出决定前，仔细审查患者档案和可能的诊断列表。不要回答“信息不足”——必须从给定选项中选择。有效诊断（选择一个）：

图 11：洗牌对象任务描述变化

- 可能的非 ST 段抬高型心肌梗死/ST 段抬高型心肌梗死  
自发性肋骨骨折  
肺栓塞  
肺肿瘤
- ...
- 鲑鱼中毒食物中毒

图 12：DDXPlus 任务描述变化

**Task description variation1:**

You are given a sentence and your task is to determine whether a sentence relating to sports is plausible or implausible

Read carefully for each of the last question and think step by step before answering.

Answer yes if its plausible, no if implausible

**Task description variation2:**

You are given a sentence and your task is to determine whether a sentence relating to sports is plausible or implausible. Read carefully for each of the last question and think step by step before answering. Answer yes if its plausible, no if implausible

**Task description variation3:**

Sentence plausibility assessment:

- Given: A sentence related to sports
- Required: Determine if the sentence is plausible or implausible
- Process: Think step by step to analyze the sentence
- Output: Answer "yes" if plausible, "no" if implausible

Figure 13: Sports Task Description Variations

**Task description variation1:**

In this task, you are given a short passage that conveys stereotype or anti-stereotype about a specific target. A stereotype is an over-generalized belief about a particular group of people. An anti-stereotype is an idea that goes against a common stereotype. You are expected to classify the passage into four types of stereotype or anti-stereotype: gender, profession, race, and religion.

**Task description variation2:**

You are expected to classify the passage into four types of stereotype or anti-stereotype: gender, profession, race, and religion.

In this task, you are given a short passage that conveys stereotype or anti-stereotype about a specific target. A stereotype is an over-generalized belief about a particular group of people. An anti-stereotype is an idea that goes against a common stereotype.

**Task description variation3:**

Sentence stereotype assessment:

- Given: A passage related to stereotype or anti-stereotype
- Required: Determine if the paragraph is one of these four category : gender, profession, race, and religion
- Output: Answer only one of the four category

Figure 14: Task 280 Task Description Variations

**任务描述变体 1:**  
你被给出一个句子，你的任务是判断一个与体育相关的句子是否合理。仔细阅读每个最后一个问题，在回答之前逐步思考。如果合理，回答“是”，如果不合理，回答“否”。

**任务描述变体 2:**  
你被给出一个句子，你的任务是判断一个与体育相关的句子是否合理。仔细阅读每个最后一个问题，并在回答前逐步思考。如果合理，回答“是”，如果不合理，回答“否”。

**任务描述变体 3:**  
句子可信度评估：  
• 给定：与体育相关的句子  
• 要求：判断句子是否可信或不可信  
• 过程：逐步思考分析句子  
• 输出：如果可信回答“是”，如果不可信回答“否”

图 13：体育任务描述的变体

**任务描述变体 1：**  
在这个任务中，你将得到一段简短的文字，其中包含关于特定目标的刻板印象或反刻板印象。刻板印象是对特定人群的一种过度概括的信念。反刻板印象是一种与常见刻板印象相悖的观点。你被期望将这段文字归类为四种刻板印象或反刻板印象类型：性别、职业、种族和宗教。

**任务描述变体 2：**  
您需要将这段文字分类为四种刻板印象或反刻板印象类型：性别、职业、种族和宗教。在这个任务中，您将得到一段简短的文字，它传达了对特定目标的刻板印象或反刻板印象。刻板印象是对特定人群的一种过度概括的信念。反刻板印象是一种与常见刻板印象相悖的观点。

**任务描述变体 3：**  
句式刻板印象评估：  
• 给定：与刻板印象或反刻板印象相关的段落  
• 要求：确定该段落是否属于以下四个类别之一：性别、职业、种族和宗教  
• 输出：仅回答四个类别中的一个

图 14：任务 280 任务描述变化

**Task description variation1:**

Act as a finance expert and assign the content based to the valid category

All possible valid category for you to choose from are as follows (one category per line, in the format of <category>):

- Finance
- Technology
- Tax and Accounting
- Business and Management
- Government and Controls
- Industry

Your answer MUST based on the above options, do not answer Insufficient information

**Task description variation2:**

Act as a finance expert and assign the content based to the valid category

Your answer MUST based on the above options, do not answer Insufficient information

All possible valid category for you to choose from are as follows (one category per line, in the format of <category>):

- Finance
- Technology
- Tax and Accounting
- Business and Management
- Government and Controls
- Industry

**Task description variation3:**

Act as a finance expert and assign the content based to the valid category

All possible valid category for you to choose from are as follows (one category per line, in the format of <category>):

Finance  
Technology  
Tax and Accounting  
Business and Management  
Government and Controls  
Industry

Your answer MUST based on the above options, do not answer Insufficient information

**DA prompt description variation 1:****Natural language:**

Derive the most likely category to answer key. Provide your output in the following valid text format:

Answer: ...

**JSON:**

Derive the most likely category to answer key. Provide your output in the following valid JSON format:

```
““json
{
 "answer": "..."
} ““
```

**YAML:**

Derive the most likely category to answer key. Provide your output in the following valid YAML format:

```
““yaml
answer: ...
““
```

**XML:**

Derive the most likely category to answer block Provide your output in the following valid XML format:

```
““xml
<root>
<answer>...</answer>
</root>
““
```

Figure 15: MultiFin Task Description Variations

Figure 16: Variation 1 for direct Answering format with only answer field in all 4 format.

**任务描述变体 1:**

扮演财务专家，根据内容分配到有效类别。您可以选择的所有有效类别如下（每行一个类别，格式为<类别>）：

金融

- 技术
- 税务与会计
- 商业与管理
- 政府 和 控制
- 行业 您的回答必须基于上述选项，不要回答信息不足

**任务描述变体 2:**

扮演财务专家并根据内容分配到有效类别，您的答案必须基于上述选项，不要回答信息不足。您可以选择的所有有效类别如下（每行一个类别，格式为<类别>）：

金融

- 技术
- 稽核与会计
- 商业与管理
- 政府 和 控制

**任务描述变体 3:**

扮演财务专家并根据以下有效类别分配内容  
(每行一个类别，格式为<类别>)：金融  
技术  
税务与会计  
商业与管理  
政府与控制  
工业  
您的回答必须基于以上选项，不要回答信息不足

**DA 提示描述变体 1：自然语言：**

Answer: 推导出最有可能回答关键问题的类别。

**JSON:**

```
JSON: { "answer": "推导出最有可能的类别以回答关键问题。" } "
```

**YAML:**

```
'yaml answer: 推导出最有可能回答关键问题的类别。'
```

**XML:**

推导最有可能的类别以回答阻塞问题。请以下有效的 YAML 格式提供输出：

```
" 'xml' 译为: " 'xml'"
<root> <answer>...</answer> </root>
"
```

图 15: MultiFin 任务描述变化

图 16: 直接回答格式的变体 1，所有 4 种格式中仅包含答案字段。

**DA prompt description variation 2:****Natural language:**

Provide your output in the following text format:

Step by step reasoning: ...

Answer: The final answer is ...

**JSON:**

Provide your output in the following valid JSON format:

```
““json
{
 "step_by_step_reasoning": "...",
 "answer": ...
}
““
```

**YAML:**

Provide your output in the following valid YAML format:

```
““yaml
step_by_step_reasoning: |
...
answer: ...
““
```

**XML:**

Provide your output in the following valid XML format:

```
““xml
<root>
 <step_by_step_reasoning>...
 </step_by_step_reasoning>
 <answer>...</answer>
</root>
““
```

**DA prompt description variation 3:****Natural language:**

Provide your output in the following text format:

Answer: <think step by step>. The final answer is <answer>

**JSON:**

Provide your output in the following valid JSON format:

```
““json
{
 "reason": "<think step by step>",
 "answer": <answer>
}
““
```

**YAML:**

Provide your output in the following valid YAML format:

```
““yaml
reasoning: |
<think step by step>,
answer: <answer>
““
```

**XML:**

Provide your output in the following valid XML format:

```
““xml
<root>
 <reason><think step by step></reason>
 <answer><answer></answer>
</root>
““
```

Figure 17: Variation 2 for direct Answering format with only answer field in all 4 format.

Figure 18: Variation 3 for direct Answering format with only answer field in all 4 format.

DA 提示描述变化 2：自然语言：

提供您的输出以下文本格式：逐步推理：...

答案：最终答案是 ...

JSON：

提供您的输出，以下为有效的 JSON 格式：

```
"json { "step_by_step_reasoning": ...,
"answer": ... }"
```

YAML：

提供您的输出，以下为有效的 YAML 格式：

```
"yaml step_by_step_reasoning: | ..."
```

回答：...

" "

XML：

提供您的输出，以下为有效的 XML 格式：

```
"xml "
```

DA 提示描述变体 3：自然语言：

提供您的输出，按照以下文本格式：答案：<逐步思考>。最终答案是：<答案>

JSON：

```
JSON: { "reason": "<逐步思考>", "answer":
<答案> } "
```

YAML：

提供您的输出，以下为有效的 YAML 格式：

```
"yaml reasoning: | <逐步思考>, answer:
<答案> "
```

XML：

提供您的输出，以下为有效的 XML 格式：

```
"xml [逐步思考] [答案] "
```

图 17：直接回答格式的变体 2，所有 4 种格式中仅包含答案字段。

图 18：仅包含答案字段的所有 4 种格式的直接回答格式变体 3。

**CoT prompt description variation 1:****Natural language:**

Provide your output in the following text format:

Answer: <reasoning first>. The final answer is <answer>

**JSON:**

Provide your output in the following valid JSON format:

```
““json
{
 "reason": ...,
 "answer": ...
}
““
```

**YAML:**

Provide your output in the following valid YAML format:

```
““yaml
reasoning: |
...
answer: ...
““
```

**XML:**

Provide your output in the following valid XML format:

```
““xml
<root>
<reason>...</reason>
<answer>...</answer>
</root>
““
```

**CoT prompt description variation 2:****Natural language:**

Provide your output in the following text format:

Step by step reasoning: ...

Answer: The final answer is ...

**JSON:**

Provide your output in the following valid JSON format:

```
““json
{
 "step_by_step_reasoning": ...,
 "answer": ...
}
““
```

**YAML:**

Provide your output in the following valid YAML format:

```
““yaml
step_by_step_reasoning: |
...
answer: ...
““
```

**XML:**

Provide your output in the following valid XML format:

```
““xml
<root>
<step_by_step_reasoning>...
</step_by_step_reasoning>
<answer>...</answer>
</root>
““
```

Figure 19: Reasoning response prompt - Variation 1

Figure 20: Reasoning response prompt - Variation 2

CoT 提示描述变体 1：自然语言：

提供您的输出应遵循以下文本格式：答案：<推理首先>。最终答案是 <答案>

JSON：

JS请以以下有效的 JSON 格式提供您的输出：`'json { "reason": ..., "answer": ... }'`

YAML：

提供您的输出，以下为有效的 YAML 格式：  
`"yaml reasoning: | ..."`

回答：...

“ ”

XML：

提供您的输出，以下为有效的 XML 格式：  
`"xml <原因>... <答案>..."`

CoT 提示描述变化 2：自然语言：

提供您的输出以下文本格式：逐步推理：...

答案：最终答案是 ...

JSON：

提供您的输出，以下为有效的 JSON 格式：  
`"json { "step_by_step_reasoning": ..., "answer": ... }"`

YAML：

提供您的输出，以下为有效的 YAML 格式：  
`"yaml step_by_step_reasoning: | ..."`

回答：...

“ ”

XML：

提供您的输出，以下为有效的 XML 格式：  
`"xml ... ... "`

图 19：推理响应提示 - 变体 1

图 20：推理响应提示 - 变体 2

**CoT prompt description variation 3:****Natural language:**

Provide your output in the following text format:

Answer: <think step by step>. The final answer is <answer>

**JSON:**

Provide your output in the following valid JSON format:

```
```json
{
  "reason": "<think step by step>",
  "answer": <answer>
}
```

```

**YAML:**

Provide your output in the following valid YAML format:

```
```yaml
reasoning: |
<think step by step>,
answer: <answer>
```

```

**XML:**

Provide your output in the following valid XML format:

```
```xml
<root>
<reason><think step by step></reason>
<answer><answer></answer>
</root>
```

```

Figure 21: Reasoning response prompt - Variation 3

CoT 提示描述变化 3：自然语言：

提供您的输出，按照以下文本格式：答案：<逐步思考>。最终答案是：<答案>

JSON：

```
JSON: { "reason": "<逐步思考>", "answer":
<答案> } "
```

YAML：

提供您的输出，以下为有效的 YAML 格式：  
“ ‘yaml reasoning: | <逐步思考>, answer:  
<答案> ’ ”

XML：

提供您的输出，以下为有效的 XML 格式：  
“ ‘xml [逐步思考] [答案] ’ ”

图 21：推理响应提示 - 变体 3