The *Idola Tribus* of AI: Large Language Models tend to perceive order where none exists

Shin-nosuke Ishikawa
Graduate School of Artificial Intelligence and Science, Rikkyo University
Strategic Digital Business Unit, Mamezou Co., Ltd.
shinnosuke-ishikawa@rikkyo.ac.jp

Masato Todo and **Taiki Ogihara**
Strategic Digital Business Unit,
Mamezou Co., Ltd.

Hirotsugu Ohba
Graduate School of Artificial Intelligence
and Science, Rikkyo University

Abstract

We present a tendency of large language models (LLMs) to generate absurd patterns despite their clear inappropriateness in a simple task of identifying regularities in number series. Several approaches have been proposed to apply LLMs to complex real-world tasks, such as providing knowledge through retrieval-augmented generation and executing multi-step tasks using AI agent frameworks. However, these approaches rely on the logical consistency and self-coherence of LLMs, making it crucial to evaluate these aspects and consider potential countermeasures. To identify cases where LLMs fail to maintain logical consistency, we conducted an experiment in which LLMs were asked to explain the patterns in various integer sequences, ranging from arithmetic sequences to randomly generated integer series. While the models successfully identified correct patterns in arithmetic and geometric sequences, they frequently over-recognized patterns that were inconsistent with the given numbers when analyzing randomly generated series. This issue was observed even in multi-step reasoning models, including OpenAI o3, o4-mini, and Google Gemini 2.5 Flash Preview Thinking. This tendency to perceive non-existent patterns can be interpreted as the AI model equivalent of *Idola Tribus* and highlights potential limitations in their capability for applied tasks requiring logical reasoning, even when employing chain-of-thought reasoning mechanisms.

1 Introduction

Recent achievements of large language models (LLMs) have raised expectations that they can perform well across a wide range of tasks in human activities, reducing labor and duties through automation enabled by artificial intelligence (AI) technologies (Minaee et al., 2024; Wan et al., 2024). In particular, the AI agent framework is a promising approach for handling complex tasks by integrating LLMs with external systems, enabling self-situation understanding and action planning (Xi et al., 2025; Wooldridge and Jennings, 1995). To develop autonomous systems capable of executing complex real-world tasks, a deep understanding of LLM behavior is essential.

A key advantage of LLMs over other AI systems is that their input and output are in human-understandable natural language. Users can provide task instructions through everyday conversation without strict formatting, much like assigning tasks to colleagues. There is ongoing discussion that LLMs exhibit behavior that can only be interpreted as genuine language understanding, making it increasingly difficult to dismiss the possibility that they truly comprehend natural language (Mitchell and Krakauer, 2023). Additionally, results can be received through a conversational interface, allowing users to ask follow-up questions. In this context, it is crucial to ensure that LLMs accurately interpret human-provided prompts and generate reasonable and reliable outputs, rather than merely producing "language-like" lists of information. To assess this aspect, (Kong et al., 2024) proposed a method for aligning LLMs with user objectives to enhance human-LLM communication. Similarly, the LCM team et al. (2024) introduced a new framework, the Large Concept Model, which aims to address the abstract nature of natural language communication by incorporating higher-level semantic representations called "concepts."

One widely recognized issue in LLM applications is hallucination, where LLM outputs contain untruthful information (Huang et al., 2025; Zhang et al., 2023). While one possible cause is the limited knowledge coverage in training data, studies have shown that hallucinations can occur even when an LLM possesses sufficient knowledge to generate correct answers (Simhi et al., 2024). To address knowledge gaps, various frameworks have been proposed to enhance LLMs with external information through input prompts, such as in-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 12714–12727

context learning (Brown et al., 2020) and retrieval-augmented generation (Gao et al., 2024).

Another critical issue with LLMs is ensuring a reasonable thinking process and self-consistency. Leaps in logic or unsupported conjectures can lead to false conclusions and may ultimately cause task execution failures. Chain-of-thought (CoT) prompting techniques have been proposed to address this issue by encouraging step-by-step reasoning (Wei et al., 2022; Kojima et al., 2022). Recently, major LLM developers have introduced "thinking" models, such as OpenAI o3, o4-mini (OpenAI, 2025b), and Google Gemini 2.5 Flash Preview Thinking (Deepmind, 2025), which incorporate built-in multi-step self-evaluation and modification mechanisms based on the CoT concept.

A reasonable thinking process involves not only deduction but also induction, both of which are crucial for executing real-world tasks. In practical societal settings, strict procedural definitions for tasks are often limited, making it necessary to formulate hypotheses and proceed accordingly. Effective hypothesis formation requires not only consistency with the provided information but also the ability to abstract key patterns and principles. Without proper abstraction, misunderstandings or biased assumptions can critically impact task execution.

In this paper, we present an experiment to investigate the capabilities of LLMs, including thinking models, in forming hypotheses through precise information understanding and pattern abstraction. We use a simple task of identifying regularities in number series, allowing us to evaluate LLMs' pattern recognition ability independently of hallucinations related to knowledge accuracy.

Francis Bacon, in his famous work *Novum Organum* (Bacon, 1620), identified the tendency of biased belief and referred to it as *Idola*, an inherent aspect of human nature. In particular, the tendency to over-recognize patterns in randomness is called *Idola Tribus* (Idols of the Tribe). The motivation of this research is to examine whether *Idola Tribus*, as described in the following citation, is a relevant concern for AI systems, particularly LLMs.

> *"The human understanding is of its own nature prone to suppose the existence of more order and regularity in the world than it finds. And though there be many things in nature which are singular and unmatched, yet it devises for them parallels and conjugates and relatives which do not exist."*
>
> *— Francis Bacon, Novum Organum, Aphorism XLV. (1620)* [1]

2 Related Work

2.1 Evaluation and Enhancement of Logical Reasoning Capabilities

Various studies have been conducted to evaluate logical consistency in natural language processing, and several datasets have been introduced to assess the logical reasoning capabilities of LLMs. ReClor (Yu et al., 2020), LogiQA (Liu et al., 2020), and LogiQA2.0 (Liu et al., 2023) are representative datasets for evaluating self-consistent deductive reasoning. These datasets assess logical capabilities through predefined answer choices, where the correct answers are explicitly determined. While (Creswell et al., 2022) introduced inductive tasks to evaluate LLMs' logical reasoning capabilities, the correct answers in these tasks are still explicitly defined.

Our approach focuses on evaluating the inductive information abstraction capabilities of LLMs, making tasks without fixed answers more appropriate. In this context, we designed an experiment on regularity identification in number series, highlighting the contrast with previous studies.

In addition, several studies have aimed to enhance the logical reasoning abilities of LLMs. Dalvi et al. (2022) proposed a framework for explaining the reasoning process using entailment trees, while Pan et al. (2023) attempted to improve logical reasoning by integrating LLMs with symbolic solvers. These approaches could potentially contribute to enhancing the information abstraction capabilities evaluated in this paper.

2.2 Mathematical Capability

The task of identifying regularities in numerical series, as addressed in this paper, can also be interpreted as part of mathematical problem-solving. Various datasets, ranging from grade school to college-level mathematics, have been developed to assess LLMs' mathematical capabilities, including AQUA-RAT (Ling et al., 2017), MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), GSM-Plus (Li et al., 2024). Recently, it has been reported

[1] Original Latin edition (Bacon, 1620); English translation from (Bacon, 1900).

that an LLM-based system achieved a gold medalist performance in the Mathematical Olympiad (Chervonyi et al., 2025).

Basic number series problems, such as arithmetic and geometric sequences, are included in the MATH dataset. However, these problems are designed to be solved using explicit rules and are fundamentally different from the open-ended task presented in this paper. While this study is related to mathematical capability, its primary objective is to evaluate self-consistency free from biases, rather than focusing solely on the mathematical aspect. In addition, our task does not require the ability to convert real-world problem statements into mathematical expressions for computation.

2.3 Biases in LLMs

There are many indications of biases in LLMs, including gender (Wan et al., 2023), political (Rozado, 2024), and cultural (Tao et al., 2024) biases. Although this stems from a different perspective than the concept of *Idola Tribus*, which is the main focus of this study, LLMs have also been reported to exhibit behaviors analogous to human cognitive biases (Echterhoff et al., 2024; Shaikh et al., 2024). Since LLMs are trained on texts written by humans, these biases can be seen as a negative legacy inherited from human data. The findings of this study should be considered as part of these biases and represent an issue that needs to be addressed.

3 Method

We conducted an experiment in which LLMs identified regularities in a prepared list of number series and then evaluated whether their descriptions accurately explained the series to investigate *Idola Tribus*, over-recognition biases. The details of the regularity identification configuration and evaluation method are provided in the following subsections.

3.1 Configuration for Regularity Identification in Number Sequences

To evaluate how LLMs generalize provided information and identify patterns, as well as how they make hasty and inaccurate generalizations leading to false patterns, we prepared several categories of integer sequences, ranging from easily recognizable patterns to cases that are nearly impossible to define with a clear rule. Table 1 presents these categories along with descriptions, the number of series prepared, and examples. We fixed several sets of number series for each category and had multiple LLMs perform identical tasks using the same numbers. The total number of numerical series tested was 724. In general, we used positive integers up to 100 as the values in the series, except for the geometric series categories, to simplify the experimental settings and clearly assess whether LLM responses exhibit bias.

For the easier cases, we prepared arithmetic, geometric, and difference series with difference sequences. The arithmetic series were generated with first terms and common differences selected randomly from integers between 1 and 9. In the geometric series, the common ratio was chosen from integers between -5 and 5, excluding -1, 0, and 1, to prevent excessively large absolute values in the sequence. The first term was randomly selected from integers between 1 and 9, following the same setting as the arithmetic series. For the difference series, the first term, along with both the first term and the common difference of the difference sequence, were randomly selected from integers between 1 and 9. We also prepared quasi-ordered cases—arithmetic, geometric, and difference series with a single-term error of +1 or -1—to investigate whether LLMs recognize these errors and distinguish them from purely ordered series.

In addition to the number series categories described above, we prepared two categories: random-increasing and random. The random-increasing series consist of randomly generated numbers with the condition that each term is greater than the previous one, while the random series have no such constraint. These series generally do not exhibit clear regularity and are used to assess whether LLMs incorrectly perceive false patterns. To ensure distinct categorization, we designed these series to avoid overlap with other categories (e.g., random-increasing series are not arithmetic, geometric, or difference series, and random series do not belong to the random-increasing category). The random-increasing series are generated by adding random integers between 1 and 10 to the previous term, starting from an initial term randomly selected between 2 and 18. The random series are generated using random integers between 1 and 99.

We selected the latest high-performance LLMs, widely used across various applications, for evaluation: Open AI GPT-4.1 (OpenAI, 2025a), o3, o4-mini (OpenAI, 2025b) and Google Gemini

#	Count	Category	Description	Example
1	81	arithmetic	Arithmetic series with positive integer common differences ranging from 1 to 9. The number of patterns is 81, consisting of 9 common differences (1 to 9) multiplied by 9 first terms (1 to 9).	$8, 9, 10, 11, 12$,...
2	81	geometric	Geometric series with integer common ratios from -5 to 5, excluding 0 and 1. The number of patterns is 81, consisting of 9 common ratio (-5 to -1 and 2 to 5) multiplied by 9 first terms (1 to 9).	$3, -6, 12, -24, 48$,...
3	100	difference	Number series in which the differences between consecutive terms form arithmetic sequences with positive integer common differences ranging from 1 to 9.	$4, 7, 11, 16, 22$,...
4	81	quasi-arithmetic	Almost arithmetic series with the same conditions as #1, but with one term deviating by 1 from the expected pattern.	$8, 10, 10, 11, 12$, ...
5	81	quasi-geometric	Almost geometric series with the same conditions as #2, but with one term deviating by 1 from the expected pattern.	$3, -6, 11, -24, 48$,...
6	100	quasi-difference	Almost difference series with the same conditions as #3, but with one term deviating by 1 from the expected pattern.	$4, 8, 11, 16, 22$,...
7	100	random-increasing	Randomly generated increasing integer sequences. Not applicable to #1–#6.	$17, 25, 33, 43, 50$,...
8	100	random	Randomly generated integer sequences. Not applicable to #1–#7.	$54, 1, 78, 7, 49$,...
Total	724			

Table 1: List of number series categories for the regularity identification test, ranging from easily identifiable arithmetic series to random series with no clear order.

2.5 Flash Preview Thinking (Gemini Team, 2023; Deepmind, 2025) (abbreviated as "Gemini2.5" in the tables). The o3, o4-mini, and Gemini 2.5 Flash Preview Thinking models incorporate the latest multi-step CoT reasoning techniques to enhance logical consistency. Additionally, we included Llama 3.3 (Meta, 2025) as a representative of high-performance open models. The versions used were gpt-4.1-2025-04-14 for GPT-4.1, o3-2025-04-16 for o3, o4-mini-2025-04-16 for o4-mini, and gemini-2.5-flash-preview-04-17-thinking for Gemini 2.5 Flash Preview Thinking.

Figure 1 shows an input prompt with a simple instruction for the regularity identification task. Each prompt included five values from a given series. Regularity becomes easier to explain when only four or fewer values are provided, such as when fitting a third-order polynomial function, while randomness becomes clearer when six or more values are provided in the random series. The same prompt was used across all models. The output is restricted to a single sentence via the prompt for ease of evaluation. We standardized the approach by providing instructions solely through user prompts, without using system prompts.

Prompt for identifying regularities

role: user

content: Please explain the regularity in the following number series in one sentence, ensuring that all the numbers are not included together in your answer:
2, 33, 9, 25, 51, ...

Figure 1: Prompt for the regularity identification task, shown for a sample case of a random series: 2, 33, 9, 25, 51, ...

3.2 Evaluation Method for Inferred Regularities

As the next step, we evaluated the regularity descriptions obtained using the method described in

the previous subsection to determine how many number series were successfully generalized and abstractly described correctly. While the validated cases confirm that the LLMs performed well, analyzing the failed cases is crucial for understanding their tendencies. Therefore, the evaluation has two main objectives: assessing regularity identification capabilities and analyzing the patterns in failure cases.

We emphasize that there is no perfect autonomous evaluation method with flawless accuracy for determining the validity of regularity descriptions. However, maintaining quality and consistency across $724 \times 5 = 3{,}620$ regularity descriptions through human evaluation alone is challenging in terms of reproducibility and consistency. Therefore, we chose to use LLMs as evaluators to ensure the experiment remains reproducible and consistent, even if their accuracy is limited. The concept of using LLMs as evaluators, known as "LLM-as-a-Judge," has become increasingly common (Gu et al., 2025). Zheng et al. (2023) demonstrated that the agreement between LLM-as-a-Judge and human annotators is comparable to inter-annotator agreement, indicating that LLMs have the capability to evaluate LLM-generated descriptions.

The prompt for LLM evaluation of the regularity descriptions is shown in Fig. 2. We designed the evaluation prompt not only to assess the validity of regularity descriptions but also to analyze their characteristics, tendencies, and the potential to avoid invalid outputs. To support this, we distinguished between valid descriptions that align with the preset category and those that do not. Additionally, we included an evaluation option for descriptions that state the series is random. Since the dataset includes a random series category, models are not required to invent plausible regularities for these cases. We analyzed how often the outputs correctly identify such series as random. In summary, we defined four evaluation options (Fig. 2): (1) correct explanation aligning with the preset category, (2) correct explanation not aligning with the preset category, (3) incorrect explanation, and (4) statement that the series is random.

We calculated the success rate based on the evaluation results to assess how well the target LLMs can generalize the provided numerical information and describe the regularities. In this calculation, outputs corresponding to options 1 and 2 are both considered successful explanations, regardless of whether they align with the preset category. For categories other than arithmetic, geometric, and difference series, we also treat a statement that the series is random (option 4) as a successful explanation, as these categories do not have clearly describable regularities.

Prompt for evaluating inferred regularities

role: user
content: The number series 2, 33, 9, 25, 51, ... is generated based on the rule of {Generation rule}.
Is the following description an explanation of the regularity in the series?
"{Output from the prompt for guessing regularities}"

Please choose one of the following options and present it at the end of your output in the format "result={}":

1: Correct explanation aligning with the generation rule
2: Correct explanation but not aligning with the generation rule
3: Incorrect explanation
4: Statement that it is a random series

Figure 2: Prompt for regularity description evaluation, shown for a sample case of a random series: 2, 33, 9, 25, 51, ... corresponding to the case in Fig. 1.

For the evaluation models, we used the o3 model. This selection was based on a preliminary experiment using author-annotated descriptions, where all candidate models performed adequately, but o3 showed the best performance within the test setting. Details of the preliminary experiment are provided in Appendix A.

4 Result

We successfully obtained 3,620 regularity descriptions for 724 number series shown in Table 1 using the five regularity identification LLMs, along with evaluation results for all descriptions. A summary of the success rates, aggregated based on the evaluation results, is presented in Table 2.

All five models correctly identified the regularities in all arithmetic series, achieving a 100% success rate, and performance was also high for the geometric series. For the difference series, o3, o4-mini, Gemini 2.5 Flash Preview Thinking and Llama 3.3 maintained strong performance, while the success rate declined for GPT-4.1. For the quasi-arithmetic, quasi-geometric, and quasi-difference series, the success rate decreased across all LLMs, reflecting the fact that these series do not exhibit clear regularities. In particular, the non-thinking models GPT-4.1 and Llama 3.3 performed worse than the self-iterative reasoning models o3, o4-mini, and Gemini 2.5 Flash Preview Thinking. In the random-increasing and random categories,

Category	GPT-4.1	o3	o4-mini	Gemini2.5	Llama3.3	Average
arithmetic	**100%**	**100%**	**100%**	**100%**	**100%**	100%
geometric	**100%**	98.8%	**100%**	**100%**	95.1%	98.8%
difference	36.0%	99.0%	99.0%	**100%**	93.0%	85.4%
quasi-arithmetic	16.0%	51.9%	56.8%	**63.0%**	21.0%	41.7%
quasi-geometric	11.1%	55.6%	25.9%	**65.4%**	12.3%	34.1%
quasi-difference	4.0%	64.0%	64.0%	**70.0%**	7.0%	41.8%
random-increasing	7.0%	**66.0%**	56.0%	44.0%	4.0%	35.4%
random	8.0%	**52.0%**	43.0%	31.0%	0.0%	26.8%
Total	33.0%	**73.1%**	67.8%	70.6%	39.9%	56.9%

Table 2: Success rates of regularity identification for each category and LLM, based on the described evaluation method. Boldface indicates the highest success rate achieved by an individual model for each category.

the trend of thinking models outperforming non-thinking models remained consistent with the quasi-ordered series cases. Overall, the success rate was high for the thinking models, with comparable performance among them. However, the success rate for the random series remained notably low across all LLMs.

The fact that LLMs provide incorrect explanations for random series clearly demonstrates their tendency to overestimate regularities when interpreting information—specifically, the numerical values in the series used in the experiment—indicating the presence of *Idola Tribus* in LLMs. As a result, our findings reveal a tendency to perceive ordered patterns that are inconsistent with the provided information—clear evidence that these AI models exhibit *Idola Tribus*.

To investigate whether LLMs tend to force a pattern even when they fail to find a plausible one, we conducted an additional experiment by modifying the regularity identification prompt to explicitly allow the series to be random (Fig. 3). Table 3 shows a comparison of the rate at which each model explains that the series is random, using the original prompt (Fig. 1) and the random-allowing prompt (Fig.3). We observe that the rate of random-series explanations increases significantly with GPT-4.1, o3, o4-mini, and LLaMA 3.3 in categories other than arithmetic, geometric, or difference series. This suggests that prompting with the option to declare a series as random—indicating no significant regularity—encourages these models to state more confidently when they find no clear pattern. In contrast, Gemini 2.5 Flash Preview Thinking does not show a similar change.

The success rates with the random-allowing prompt configuration are shown in Table 4. Com-

Prompt for identifying regularities (random allowed)

role: user

content: Please explain the regularity in the following number series in one sentence, without listing all the numbers together in your answer. If you find no significant regularity, you may respond that it is a random series: 2, 33, 9, 25, 51, ...

Figure 3: Prompt for the regularity identification task, shown for a sample case of a random series: 2, 33, 9, 25, 51, ...

pared to Table 2, the improvements correspond to the increase in random-series explanations reported in Table 3. The o3 and o4-mini models showed comparable performance, achieving the highest success rates among the models. Nevertheless, success rates for quasi-ordered series remain significantly lower than those for their corresponding ordered cases across all models. Gemini 2.5 Flash Preview Thinking did not show a notable improvement, as the number of random-series explanations did not increase substantially.

5 Discussion

As shown in the previous section, we presented a clear and comprehensive analysis—demonstrating for the first time—that LLMs occasionally generate false patterns that contradict the information they are given. When a clear regularity cannot be identified, the model should explicitly state this, rather than providing potentially inaccurate explanations. The lack of random-series explanations under the original prompt clearly illustrates this issue, where models incorrectly assert the presence of a pattern instead of acknowledging the absence of an identifiable structure.

One notable finding is that the tendency to over-

Category	GPT-4.1	o3	o4-mini	Gemini2.5	Llama3.3	Average
quasi-arithmetic	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
quasi-geometric	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
quasi-difference	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
random-increasing	1.0%	**9.0%**	6.0%	8.0%	0.0%	4.8%
random	5.0%	**28.0%**	14.0%	17.0%	0.0%	12.8%
Total	1.3%	**8.0%**	4.3%	5.4%	0.0%	3.8%

Category	GPT-4.1	o3	o4-mini	Gemini2.5	Llama3.3	Average
quasi-arithmetic	18.5%	13.6%	**30.9%**	0.0%	2.5%	13.1%
quasi-geometric	**13.6%**	3.7%	9.9%	0.0%	6.2%	6.7%
quasi-difference	6.0%	33.0%	**59.0%**	0.0%	0.0%	19.6%
random-increasing	38.0%	56.0%	**66.0%**	5.0%	17.0%	36.4%
random	83.0%	87.0%	**93.0%**	30.0%	86.0%	75.8%
Total	33.1%	41.1%	**54.3%**	7.6%	23.8%	32.0%

Table 3: Rate of explanations stating that the target is a random series (evaluation result option 4). Upper panel: results with the original prompt (Fig. 1); lower panel: results with the random-allowing prompt (Fig. 3). Boldface indicates the highest rate by an individual model for each category.

Category	GPT-4.1	o3	o4-mini	Gemini2.5	Llama3.3	Average
arithmetic	**100%**	**100%**	**100%**	**100%**	**100%**	100%
geometric	**100%**	**100%**	98.8%	**100%**	**100%**	99.8%
difference	45.0%	**100%**	**100%**	**100%**	97.0%	88.4%
quasi-arithmetic	40.7%	66.7%	**82.7%**	54.3%	24.7%	53.8%
quasi-geometric	24.7%	**69.1%**	45.7%	66.7%	17.3%	44.7%
quasi-difference	9.0%	84.0%	**86.0%**	69.0%	11.0%	51.8%
random-increasing	46.0%	85.0%	**91.0%**	40.0%	20.0%	56.4%
random	83.0%	94.0%	**97.0%**	30.0%	86.0%	78.0%
Total	55.0%	87.7%	**88.3%**	68.9%	56.6%	71.3%

Table 4: Success rates of regularity identification for each category and LLM using the random-allowing prompt (Fig. 3). Boldface indicates the highest success rate achieved by an individual model for each category.

recognize patterns also appeared in the thinking models, despite their higher success rates compared to non-thinking models (Table 2). This suggests either that the thinking models lack sufficient capability to verify regularity hypotheses or that they assert false regularities even when they recognize them as incorrect. This question can be examined using the o3 results, as o3 was used for both regularity identification and evaluation. We found that o3 judged its own identified regularities as valid in only 52 out of 100 cases within the random category. This indicates that the o3 model generated false patterns even in cases where it itself recognized them as incorrect. The failure to reflect self-evaluation results may be related to fundamental differences between the CoT process and human thinking (Bao et al., 2024).

By comparing correct regularity explanations that align and do not align with the preset categories (evaluation result options 1 and 2), we found that in the arithmetic, geometric, and difference series, almost all correct explanations aligned with the preset categories. Since the regularities in these categories are simple and well-defined, it is reasonable that LLMs did not attempt to explain them using alternative patterns. In contrast, for the quasi-ordered categories, which include one deviation from a clear regularity, there were comparable numbers of correct explanations both aligning and not aligning with the preset categories. This suggests that the evaluated models demonstrate strong performance in their ability to identify valid regularities, even when those regularities were not explicitly intended in the series generation process.

For the o3 and o4-mini models, a greater variety of highly creative ideas were generated compared to other models, including interpretations based on atomic numbers, football players, piano, tarot, the Holy Bible, and telephone country codes—even though these were inconsistent with the given series. This level of creativity could be a significant advantage if the model's internal mechanism for logical self-validation functioned reliably. At present, however, both o3 and o4-mini tend to validate ideas they are not well-equipped to evaluate, leading to misinterpretations and false confirmations. It has been reported that LLMs often struggle with tasks such as recognizing characters in words, counting words, or identifying letter sequences(Qin et al., 2023)—weaknesses that may contribute to the observed failures. In such cases, the models' wide-ranging ideas ultimately lead to lower-quality outputs.

Based on the behavioral changes observed with the random-allowing prompt, it is plausible that this bias stems from an implicit compulsion in LLMs to always provide an answer in response to a given instruction. This tendency can lead to false outputs resembling confabulation or improvisation—similar to hallucinations. In this context, frameworks designed to mitigate hallucinations may offer useful insights. Several studies have explored such approaches, including prompting LLMs to state explicitly when they do not have certain knowledge (Zhang et al., 2024), or even to explain why they cannot answer a given question (Deng et al., 2024). If LLMs were able to state that they do not know the regularities in a number series—or to explain why they cannot identify them—the impact of this phenomenon could be significantly reduced. Further investigation is needed to determine whether these concepts can be applied to cases of logical reasoning, rather than purely knowledge-based tasks.

The fact that the models with the best performance under the random-allowing prompt, o3 and o4-mini, achieve higher success rates for random series than for quasi-ordered series is notable. Quasi-ordered series may appear to contain a regularity, as the deviation involves only a single value in the series. There are cases in which LLMs persist in trying to explain a pattern rather than recognizing the absence of a clear rule. This tendency provides evidence that the behavior aligns with the characteristics of *Idola Tribus*, suggesting that the phenomenon observed in LLMs shares the same nature. Although the number of target LLMs is limited to five, we observed this tendency across all five models. This aspect also aligns with the concept of *Idola Tribus*, which refers to shared tendencies among humans, suggesting a comparable tendency within the "tribe" of LLMs.

One possible cause of this phenomenon might be a tendency to process information efficiently during training or instruction tuning. Providing additional prompts to help LLMs recognize this bias may reduce over-interpretation, although it could also limit their willingness to engage with more complex reasoning tasks.

To overcome this bias through approaches other than fine prompt adjustments, fine-tuning with strategies proposed in previous studies for enhancing logical reasoning capabilities could also be effective in addressing the pattern recognition biases identified in this paper (Zelikman et al., 2022; Morishita et al., 2023). Ideally, developing a bias-free model through fine-tuning would be preferable, as relying on a biased model requires constant vigilance to mitigate bias through prompting. Although the fine-tuning configurations discussed in these studies are primarily designed to improve deductive reasoning, the core principle of maintaining logical consistency with all available information is equally important for addressing inductive biases. It has also been noted that model performance depends not only on the quantity of training data but, more importantly, on its quality (Ye et al., 2025). In this context, exploring effective fine-tuning strategies is a key next step in tackling the issue.

6 Conclusion

We conducted an experiment in which LLMs identified regularities in various types of number series, including randomly generated ones, and discovered a tendency for LLMs to over-recognize patterns that do not fully explain the provided information. This tendency can be regarded as the LLM equivalent of *Idola Tribus* in humans. LLMs tend to force themselves to explain patterns even when they do not find a plausible one, unless explicitly instructed to acknowledge the absence of regularities. This tendency has been shown to share the same characteristic found in *Idola Tribus* in humans, particularly in its greater likelihood to appear when the series seems ordered.

Future research is expected to focus on mitigating the impact of this bias, and further model development will be necessary to address it. Until

this tendency is properly controlled, it is essential to remain aware of the issue when applying LLMs to practical tasks, just as humans must be cautious to avoid biases in general, in order to prevent unintended errors.

Limitation

In this paper, we conducted an evaluation experiment on only five major LLMs to confirm the tendency for false pattern recognition. However, this does not guarantee that the same tendency exists in all current and future LLMs.

We have presented results using only two variations for regularity identification, as shown in Figs. 1 and 3. Improved results may be achievable with more optimized prompts. Our intention was not to suggest that such tendencies are unavoidable, but rather to emphasize that users should be cautious of these incorrect recognitions.

Additionally, the evaluation method using LLMs selected in this study is not perfectly precise. As a result, the success rate values presented in this paper may not be highly accurate in absolute terms, though the overall trend and main findings remain unchanged.

Ethics Statement

We investigated the tendencies of selected large language models and did not handle any personal information. This research focuses solely on logical biases, excluding considerations of gender, political, or cultural biases. Therefore, it does not raise any ethical concerns.

Acknowledgements

We thank Eiiti Hanyuda for his suggestion and idea to assess the logical reasoning ability of LLMs using number series. We also thank Masao Yasui for valuable discussions on the application of LLMs to real-world tasks. This work was supported by JSPS KAKENHI Grant Number 24K15077. We utilized ChatGPT as an assistant to edit the text, aiming to improve the English expressions to make them more appropriate.

References

Francis Bacon. 1620. Novum organum.

Francis Bacon. 1900. *The Works of Francis Bacon*, volume 8. Houghton, Mifflin and Company, Boston. Publication year uncertain; often cataloged as [190-?]. American reprint of the Spedding–Ellis–Heath edition.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, Linyi Yang, and Yue Zhang. 2024. How likely do llms with cot mimic human reasoning?

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yuri Chervonyi, Trieu H. Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V. Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2.

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*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2022. Explaining answers with entailment trees.

Google Deepmind. 2025. Start building with gemini 2.5 flash. `https://developers.googleblog.com/en/start-building-with-gemini-25-flash/` Accessed: May 20, 2025.

Yang Deng, Yong Zhao, Moxin Li, See-Kiong Ng, and Tat-Seng Chua. 2024. Don't just say "I don't know"! self-aligning large language models for responding to unknown questions with explanations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13652–13673, Miami, Florida, USA. Association for Computational Linguistics.

Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. Cognitive bias in decision-making with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12640–12653, Miami, Florida, USA. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

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*, volume 35, pages 22199–22213. Curran Associates, Inc.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. 2024. Aligning large language models with representation editing: A control perspective.

LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, Jo ao Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, Safiyyah Saleem, and Holger Schwenk. 2024. Large concept models: Language modeling in a sentence representation space.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *CoRR*, abs/1705.04146.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Meta. 2025. Llama. https://www.llama.com/ Accessed: May 20, 2025.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey.

Melanie Mitchell and David C. Krakauer. 2023. The debate over understanding in ai's large language models. *Proceedings of the National Academy of Sciences*, 120(13).

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. 2023. Learning deductive reasoning from synthetic corpus based on formal logic.

OpenAI. 2025a. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/ Accessed: May 20, 2025.

OpenAI. 2025b. OpenAI o3 and o4-mini system card. https://openai.com/index/o3-o4-mini-system-card/ Accessed: May 20, 2025.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver?

David Rozado. 2024. The political preferences of llms.

Ammar Shaikh, Raj Abhijit Dandekar, Sreedath Panat, and Rajat Dandekar. 2024. Cbeval: A framework for evaluating and interpreting cognitive biases in llms.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. 2024. Distinguishing ignorance from error in llm hallucinations.

Yan Tao, Olga Viberg, Ryan S Baker, and René F Kizilcec. 2024. Cultural bias and cultural alignment of large language models. *PNAS Nexus*, 3(9).

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Michael Wooldridge and Nicholas R. Jennings. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, Qi Zhang, and Tao Gui. 2025. The rise and potential of large language model based agents: a survey. *Science China Information Sciences*, 68(2):121101.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

A Preliminary Experiment for Evaluation Method Investigation

We conducted an experiment using four number series to investigate the evaluation method. Five LLMs from the main experiment were tasked with identifying regularities, and the validity of their descriptions was annotated by the authors as ground truth. The series were selected from the random, difference, and quasi-arithmetic categories, as descriptions for random and quasi-regular series are more challenging to evaluate. These four test cases were excluded from the main experiment.

Next, we tested the evaluation prompt on 21 selected descriptions, excluding vague or overly similar ones and including some results from the random-allowing prompt (Fig. 3), using the five models as candidate LLM-as-a-judge evaluators. Table 5 the 21 descriptions used for testing. The expected evaluation results were annotated by the authors and are listed in Table 5. We note that it is not necessary to distinguish between annotations 1 and 4 for random series, as a correct description based on the generation rule (option 1) and a statement that the series is random (option 4) are effectively equivalent.

Table 6 presents the results of the preliminary experiment. We found that all candidate models performed as evaluators with a certain level of accuracy, with o3 achieving the best performance. For o3, if we treat options 1 and 2 as equivalent—since both represent acceptable cases—there was only one error (Exp. P10), involving a discrepancy between acceptable and unacceptable judgments compared to the annotation. This was the fewest among the five candidates; therefore, we selected o3 as the LLM-as-a-judge evaluator for the main experiment.

While this experiment might not be not sufficient to fully characterize the accuracy of the evaluation method, it suggests that the approach is capable of capturing general trends for each combination of regularity identifier LLM and number series category.

B Examples of outputs

Successful examples of valid regularity identification across all eight categories are shown in Table 7. These examples were selected from the descriptions evaluated as valid, representing typical cases that clearly align with human-confirmed results. In the arithmetic, geometric, and difference series categories, the expected mathematical regularities are successfully identified. In the quasi-ordered categories, the $+1$ or -1 errors in comparison to the simple ordered series were correctly recognized, and the regularities were described while account-

Exp ID	Category	Number series	Regularity identifier LLM	Regularity description	Author annotation
P1	random	$2, 33, 9, 25, 51$,...	GPT-4.1	The sequence alternates between adding 31 and subtracting 24 to generate the next number.	3
P2	random	$2, 33, 9, 25, 51$,...	o4-mini	After the initial 2, the sequence alternates between multiples of three whose other factor is a successive odd prime ($33 = 3\times11$, $51 = 3 \times 17$, ...) and perfect squares of consecutive odd integers ($9 = 3^2$, $25 = 5^2$, ...).	3
P3	random	$2, 33, 9, 25, 51$,...	o3	Treat the list as two interlaced sequences — one made by repeatedly adding successive even squares ($+4$, $+16$, $+36$, $+64$, ...) starting from 2, and the other by successively subtracting the corresponding odd squares (-9, -25, -49, ...); when these two strands are woven together they yield the given numbers in the stated order.	3
P4	random	$2, 33, 9, 25, 51$,...	Gemini2.5	The sequence starts with 2 and 33; for n greater than two, term n equals the square of (2n-3), adding two if n is odd and five or greater.	2
P5	random	$2, 33, 9, 25, 51$,...	o3	No clear arithmetic, geometric, digit-manipulation or indexing rule links each element to its successor, so the list should be treated as a random assortment of numbers.	1 or 4
P6	random	$2, 33, 9, 25, 51$,...	Gemini2.5	There is no significant regularity in this series.	1 or 4
P7	random	$85, 1, 3, 34, 1$,...	GPT-4.1	The series alternates between subtracting 84 and then multiplying the result by the next odd number in sequence (1, 3, 5,...).	3
P8	random	$85, 1, 3, 34, 1$,...	o3	Begin with a two-digit number; write the GCD of its two digits, write the absolute difference of those digits, then make the next two-digit member by using that difference as the tens digit and the digital root of the digits' sum as the units digit, and repeat.	2
P9	random	$85, 1, 3, 34, 1$,...	o4-mini	Each term after the initial two-digit number is generated by alternately summing its digits (to produce two separate one-digit entries) and then taking the two most recent one-digit entries to form the next two-digit term by writing their product followed by their sum.	3
P10	random	$85, 1, 3, 34, 1$,...	Gemini2.5	The series begins with eighty-five, and subsequent terms cycle through the values obtained by taking Fibonacci numbers at index one, four, and nine.	2
P11	random	$85, 1, 3, 34, 1$,...	Llama3.3	The series appears to alternate between a large number and the number 1, with the large numbers increasing in a specific pattern, while the small number remains constant at 1.	3
P12	random	$85, 1, 3, 34, 1$,...	Llama3.3	The series appears to be random, with no discernible pattern or regularity, as the numbers seem to be unrelated and do not follow a consistent sequence or mathematical operation.	1 or 4
P13	quasi-arithmetic	$4, 13, 22, 32, 40$,...	GPT-4.1	The pattern in the series is that the numbers increase alternately by 9 and then by 8.	3
P14	quasi-arithmetic	$4, 13, 22, 32, 40$,...	o3	Beginning with 4, each term is obtained by adding 9 to the previous one, so every member of the sequence is a number that leaves a remainder of 4 when divided by 9.	3
P15	quasi-arithmetic	$4, 13, 22, 32, 40$,...	o4-mini	Each consecutive gap cycles through adding nine, then nine again, then ten, and then eight.	2
P16	quasi-arithmetic	$4, 13, 22, 32, 40$,...	Gemini2.5	Each term after the first is found by adding an amount from a repeating four-step cycle.	2
P17	difference	$5, 14, 29, 50, 77$,...	GPT-4.1	In this series, each term increases by consecutive odd numbers multiplied by three: specifically, the difference between each consecutive pair rises by 6 each time (i.e., 9, 15, 21, 27, ...), so each term adds six more than the previous difference.	1
P18	difference	$5, 14, 29, 50, 77$,...	o3	The difference between consecutive terms rises by six each time (9, 15, 21, 27, ...), so the nth term is given by $3n^2 + 2$.	1
P19	difference	$5, 14, 29, 50, 77$,...	o4-mini	Each term equals three times the square of its position in the sequence plus two.	2
P20	difference	$5, 14, 29, 50, 77$,...	Gemini2.5	The regularity is that the difference between consecutive terms increases by a fixed value.	1
P21	difference	$5, 14, 29, 50, 77$,...	Llama3.3	The series appears to be formed by adding 9, then 15, then 21, and so on, increasing the increment by 6 each time, which suggests a quadratic or second-order relationship between the terms.	1

Table 5: Descriptions to be tested for the regularity description evaluator model, with author-provided annotations. The annotated numbers correspond to the options shown in Fig. 2. The descriptions in P5, P6, and P12 were obtained using the random-allowing prompt (Fig. 3).

Exp ID	GPT-4.1	o3	o4-mini	Gemini2.5	Llama3.3	Annotation
P1	**3**	**3**	**3**	**3**	**3**	3
P2	1	**3**	4	2	1	3
P3	**3**	**3**	4	**3**	**3**	3
P4	1	1	4	1	3	2
P5	**4**	**4**	**1**	**1**	2	1 or 4
P6	**1**	**4**	**1**	**1**	**1**	1 or 4
P7	**3**	**3**	**3**	**3**	**3**	3
P8	**2**	1	1	1	**2**	2
P9	**3**	**3**	**3**	**3**	**3**	3
P10	1	3	3	**2**	3	2
P11	**3**	**3**	**3**	**3**	**3**	3
P12	**4**	**4**	**1**	**1**	**1**	1 or 4
P13	**3**	**3**	**3**	**3**	**3**	3
P14	2	**3**	**3**	**3**	**3**	3
P15	1	**2**	**2**	1	1	2
P16	3	**2**	1	1	3	2
P17	**1**	**1**	**1**	**1**	**1**	1
P18	**1**	**1**	**1**	**1**	**1**	1
P19	1	1	1	1	1	2
P20	**1**	**1**	**1**	**1**	**1**	1
P21	**1**	**1**	**1**	**1**	**1**	1
Correct count	14	**17**	14	15	14	
Accuracy	66.7%	**81.0%**	66.7%	71.4%	66.7%	

Table 6: Result of the preliminary experiment to select the evaluation model for the main experiment, tested with the dataset shown in Table 5. Boldface in the option numbers shows correct evaluations. Boldface in the correct count and accuracy show the best case.

ing for these errors. In the random-increasing and random cases, although no clear patterns are apparent, the regularity descriptions successfully explain all the values in the series. As mentioned in the main text, we do not require the rule descriptions to be capable of reproducing all numbers in the series. Instead, a description is considered valid as long as it does not contradict the given values. Thus, the descriptions are deemed valid.

Examples of invalid regularity descriptions are listed in Table 8. We did not include cases from the arithmetic series, as all instances were correctly identified. In the geometric series, the common ratio is a fixed factor, not an increasing one. For the difference series, the change from 14 to 23 is 9, which is not a prime number. The explanations for the quasi-arithmetic, quasi-geometric, and quasi-difference series correspond to the pure forms of each series type, without accounting for deviations. In the random-increasing explanation, only the first three terms are addressed, and the explanation is inconsistent with the remaining two terms. The explanation for the random case only accounts for the difference between the first two terms.

Category	Number series	Model	Result
arithmetic	$1, 9, 17, 25, 33, ...$	Llama3.3	The series is formed by adding 8 to the previous term, starting from 1, to generate a sequence of numbers that increase by a consistent difference.
geometric	$7, -21, 63, -189, 567, ...$	GPT-4.1	Each term in the series is obtained by multiplying the previous term by -3.
difference	$9, 10, 15, 24, 37, ...$	o3	Beginning with nine, each subsequent term is produced by adding consecutive odd increments that themselves rise by four (1, then 5, then 9, then 13, and so on).
quasi-arithmetic	$6, 14, 23, 30, 38, ...$	o4-mini	Each term is generated from the previous one by cyclically adding 8, then 9, then 7, and repeating that cycle.
quasi-geometric	$1, -5, 25, -125, 626, ...$	Gemini2.5	The series generally follows a pattern of powers of -5, but the fifth term listed is one greater than this rule suggests.
quasi-difference	$2, 11, 20, 32, 44, ...$	Llama3.3	The regularity in the series is obtained by adding 9, then 9, then 12, and then 12, and so on, increasing the increment by 3 after every two steps.
random-increasing	$20, 30, 33, 41, 51, ...$	GPT-4.1	Each term in the series alternately increases by 10 and then by 3 or 8 in succession.
random	$86, 24, 59, 48, 32, ...$	o3	The sequence consists of two intertwined arithmetic progressions: the numbers in the odd positions fall by twenty-seven each time, whereas those in the even positions rise by twenty-four on every move.

Table 7: Examples of successful regularity descriptions, confirmed by human evaluation, selected from cases deemed valid.

Category	Number series	Model	Result
geometric	$4, 20, 100, 500, 2500, ...$	Llama3.3	The series progresses by multiplying the previous term by a increasing factor, starting with 5, to generate the next term.
difference	$9, 14, 23, 36, 53, ...$	GPT-4.1	In this series, each term (starting from the second) is obtained by adding consecutive prime numbers to the previous term.
quasi-arithmetic	$1, 9, 17, 25, 32, ...$	o3	It is simply an arithmetic progression in which every new element is produced by adding eight to the one before it.
quasi-geometric	$8, -16, 33, -64, 128, ...$	o4-mini	Each term alternates in sign and its absolute value doubles that of the term before it.
quasi-difference	$7, 15, 30, 52, 80, ...$	Gemini2.5	The difference between consecutive terms increases by seven.
random-increasing	$5, 8, 14, 24, 31, ...$	GPT-4.1	Each number in the series is formed by alternately adding 3 and then 6 to the previous number.
random	$54, 74, 24, 5, 23, ...$	o3	Every succeeding term is obtained from the preceding one by reversing its digits and then adding or subtracting 29 alternately.

Table 8: Examples of unsuccessful regularity descriptions, confirmed by human evaluation, selected from cases deemed invalid.