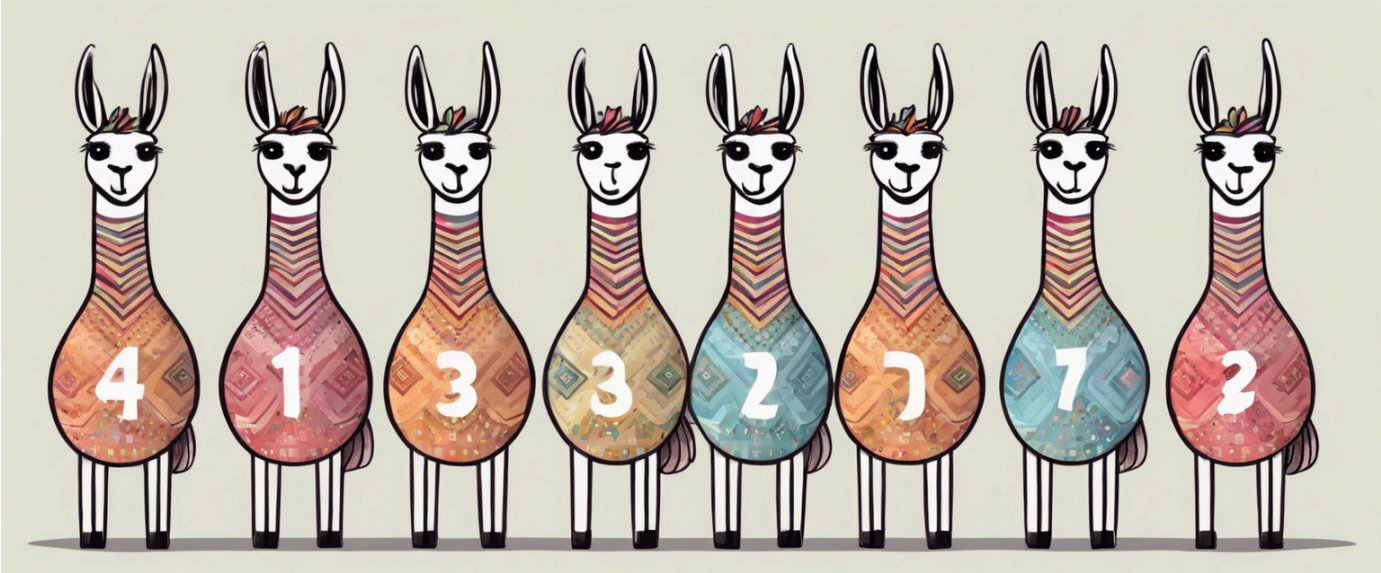
NumSeqBench: Benchmarking Inductive Reasoning in Language Models via Number Sequences



Ken Tsui

Code|Dataset

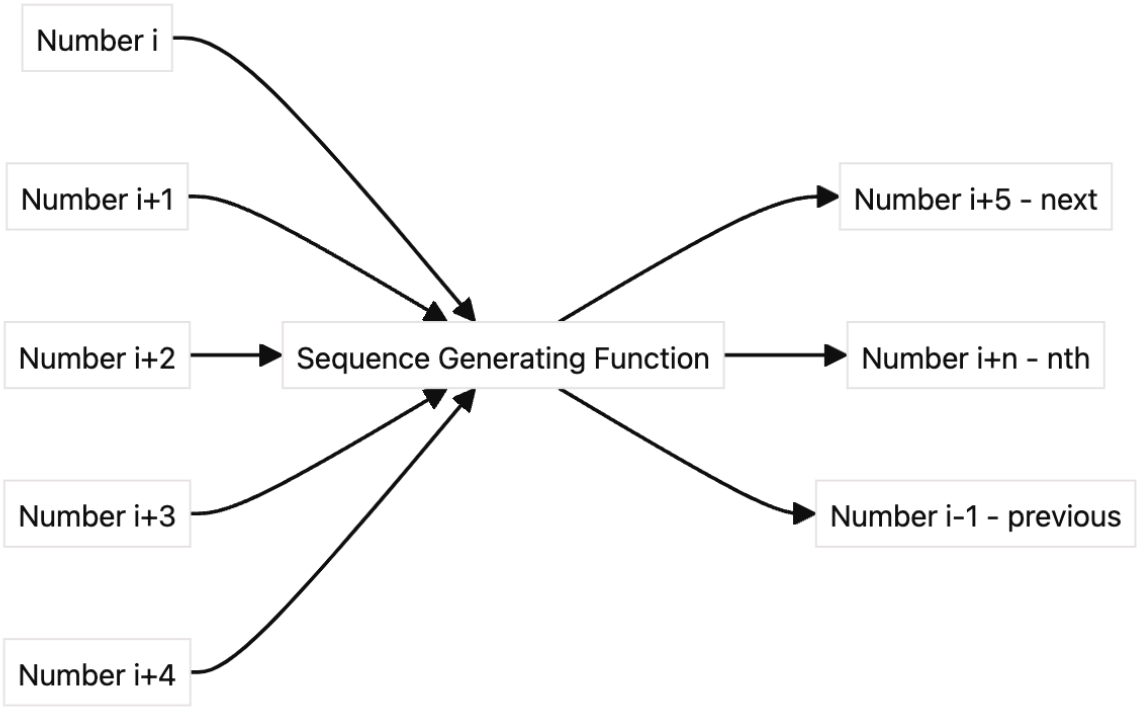
1. Introduction

Inductive reasoning is critical to human cognition. It is a form of logical thinking where we draw general conclusions from specific observations. The beautify of it is that it allows us to learn from experience, without being told the principle, and apply that knowledge to new situations.

Do LLM perform well in inductive reasoning? Few shot prompting finds that by adding more samples, task performance increases. It is an early sign that LLM can peform inductive reasoning.

We introduced a benchmark dataset named NumSeqBench to evaluate LLM capability inductive reasoning in numeric sequence. "Find the next term" in a number sequence is common in most of the cognitive tests. However, we argue that it is alone is subject to two limitations:

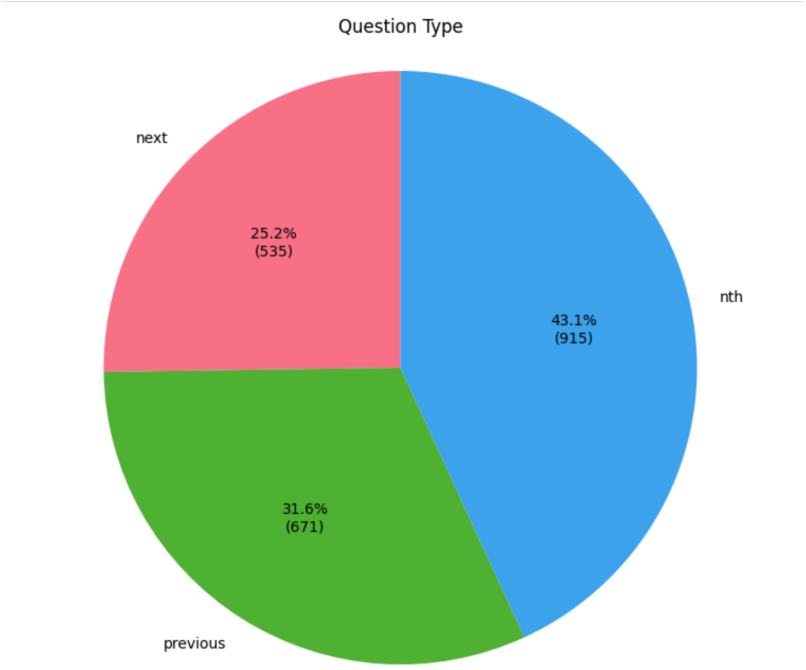
- it is not uncommon in web, heightening the data contamination risk
- find the next term is close to next token prediction pretraining objective, and as such might have an favorable performance. The short-term pattern recognition however does not necessarily mean the model understand the sequence generating function.

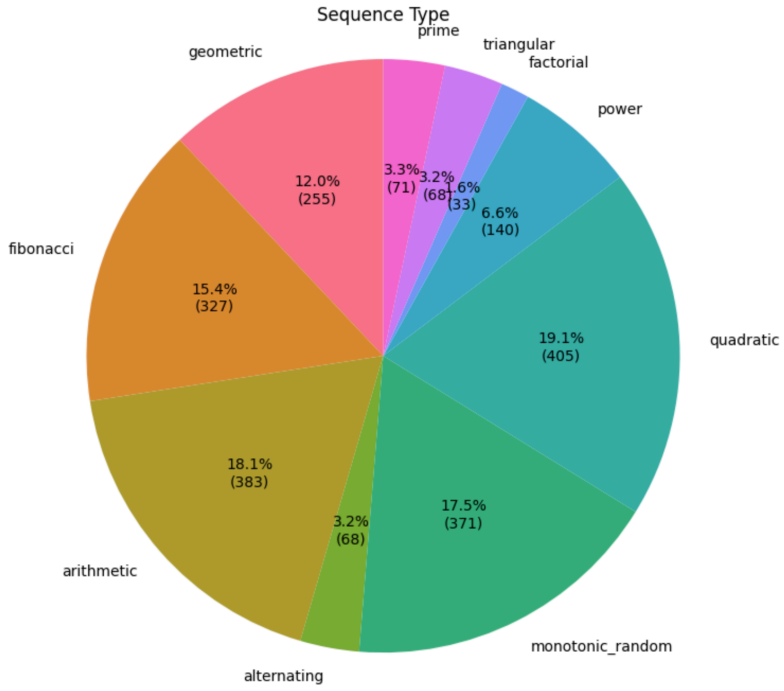


To address these limitations, we propose two extra tests - "next nth term (nth)" and "previous term (previous)" task to evaluate LLM ability to do long range inference, and backward inference. If a model can conclude the sequence generating function by induction, it can inference a term in any arbitrary position (next, nth, and the previous). Moreover, it can conclude whether such generating function exists. To test so, we added a novel test - inclusion of "monotonic random" sequence to evaluate if LLM can determine it is a random sequence and abstain from answering with a number.

2. Dataset Description

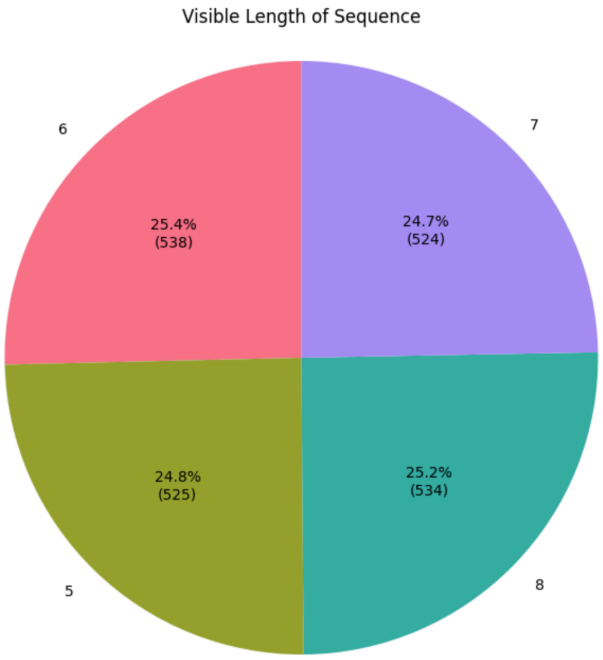
There are 2,121 unique questions. Below are the breakdown.





There are fewer prime, triangular, factorial,

alternating sequeunce because of less parameterisation in their functions.



We have avoided excessively large value of answer when parameterising different functions by restricting their range. See Code for the details.

Below shows the distribution of answer is in a reasonable range. |Percentile| Value| |-|-| |10|10| |25|17| |50|38| |75|729| |90|49,512| |95|2,032,340|

We use the prompt template:

```
Consider the following sequence: {X1, X2, X3, X4, X5}
[What is the next number in this sequence?|What is the {nth}th number in this sequence?|What is the p
Output your answer in JSON with key "answer". If you are not able to provide, answer "null"
```

3. Evaluation

LLM is prompted in zero shot setting, at temperature=0.0. No system prompt is set. Since this benchmark contains all open end questions, the random baseline is zero accuracy.

3.1 Metrics

There are two major metrics:

- `Accuracy` defined as the accuracy across different question types, namely, "next", "nth" and "previous".
- `Abstain F1` defined as the f1 score on "monotonic_random" problem.

We extracted LLM chat completion from "answer": [answer] to calculate accuracy, with a fallback to exact match the last number of the LLM generation. We added the fallback mechanism because 1) after our manual inspection of LLM

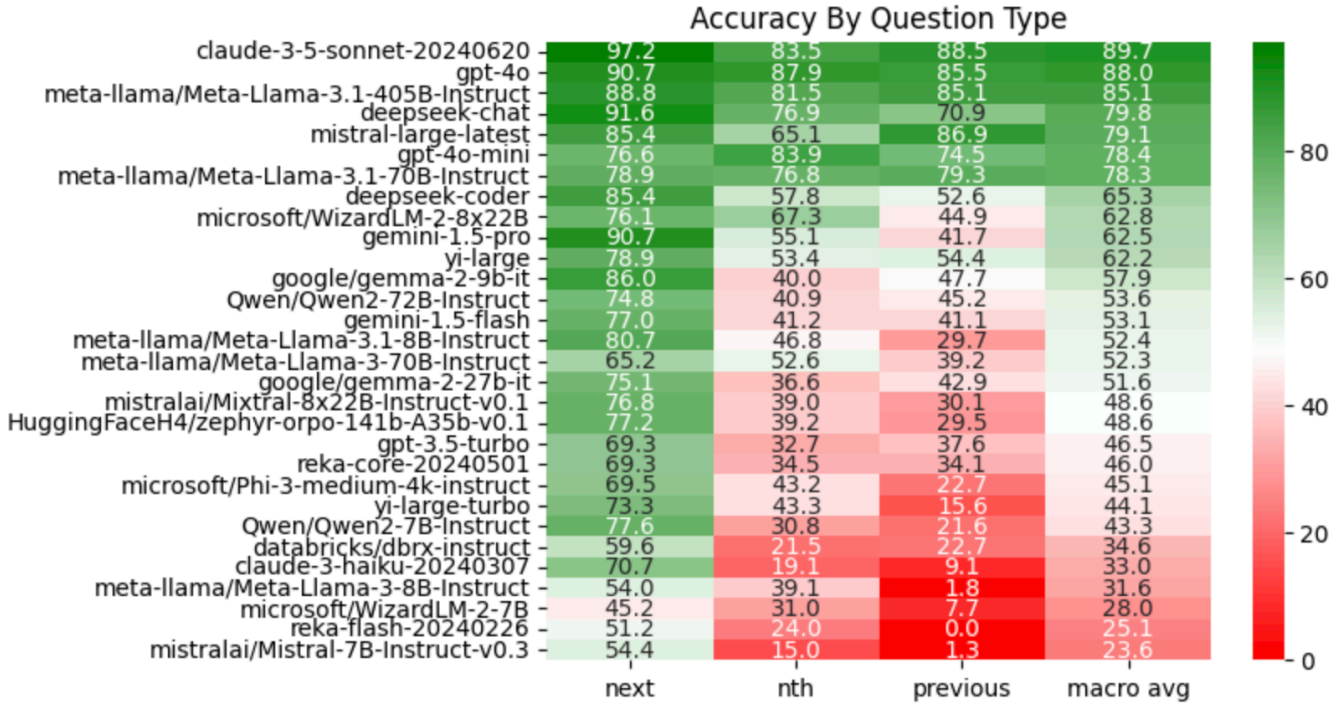
outputs that cannot be parsed, we found that their answers are correct, we do not want to miss correct answer. 2) The scope of this study is on inductive reasoning capability, but not on format following.

Nevertheless, We also show the accuracy factoring in the instruction following capabilities in the Appendix.

3.2 Results and Analysis

3.2.1 Accuracy

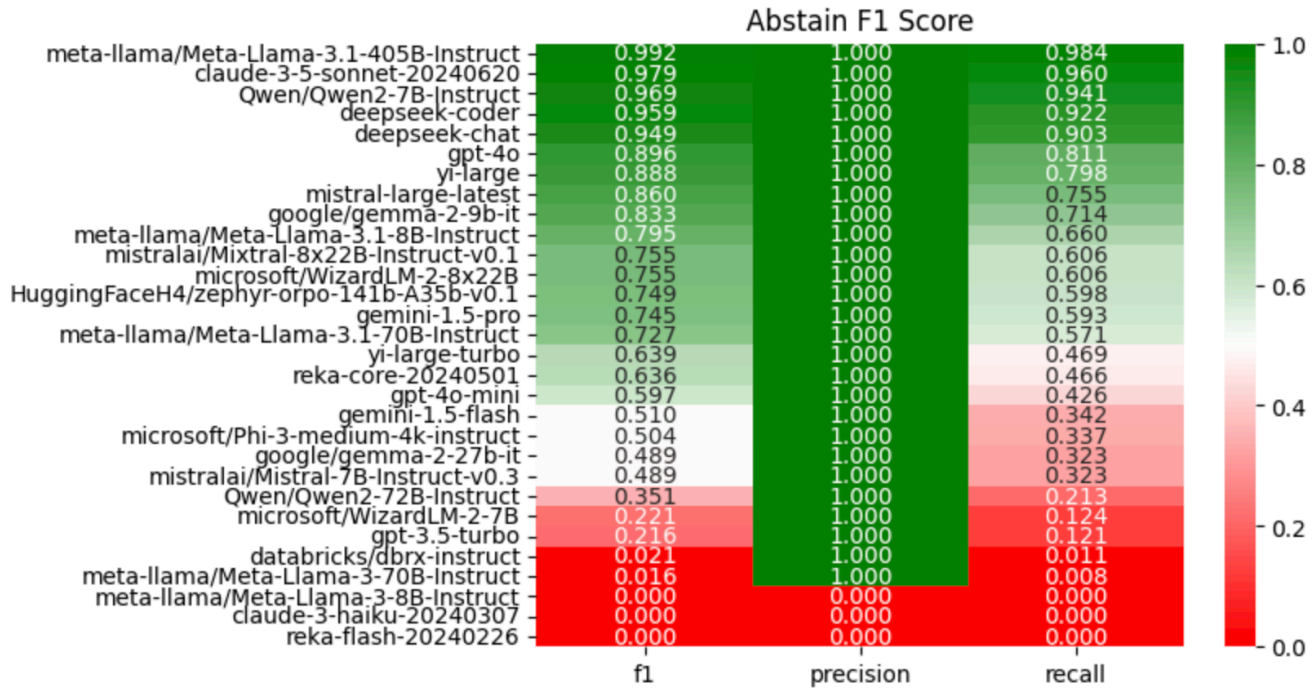
Overall, it is observed that some LLMs struggle in the benchmark, implying the difficulty of this benchmark, Claude 3.5 Sonnet tops the list with 89.7%, closely followed by gpt-4o (88.0%) and Llama3.1 405B (85.1%)).

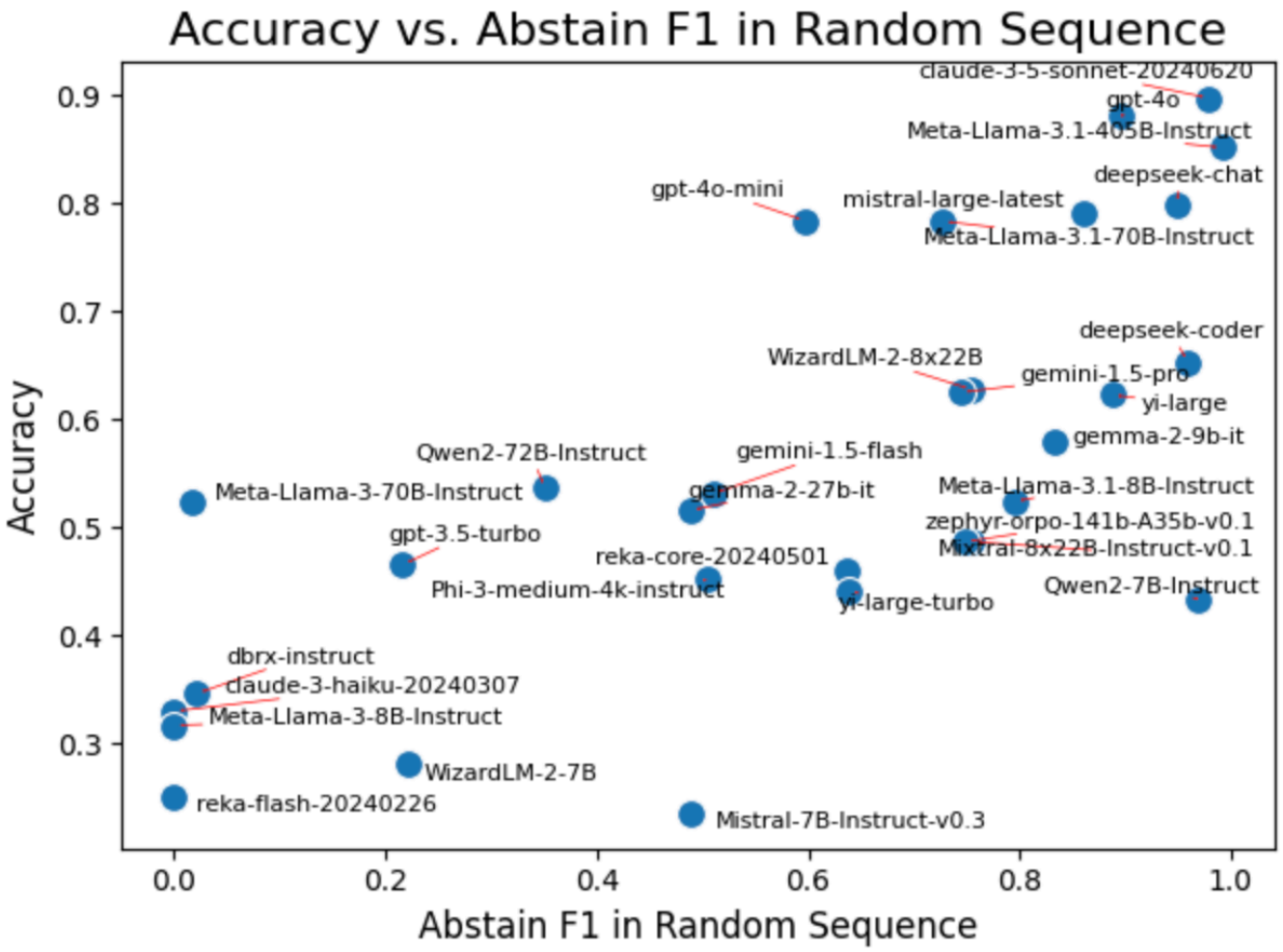


3.2.2 Abstain F1

Llama3.1 405B performed the best at abstaining from answering when given a random sequence, closely followed by Claude 3.5 Sonnet, Qwen2-7B-Instruct and DeepSeek coder and chat. Althougth there is a high correlation (0.709) between Accuracy and Abstain F1, not every strong model attained a high accuracy is able to abstain from answering,

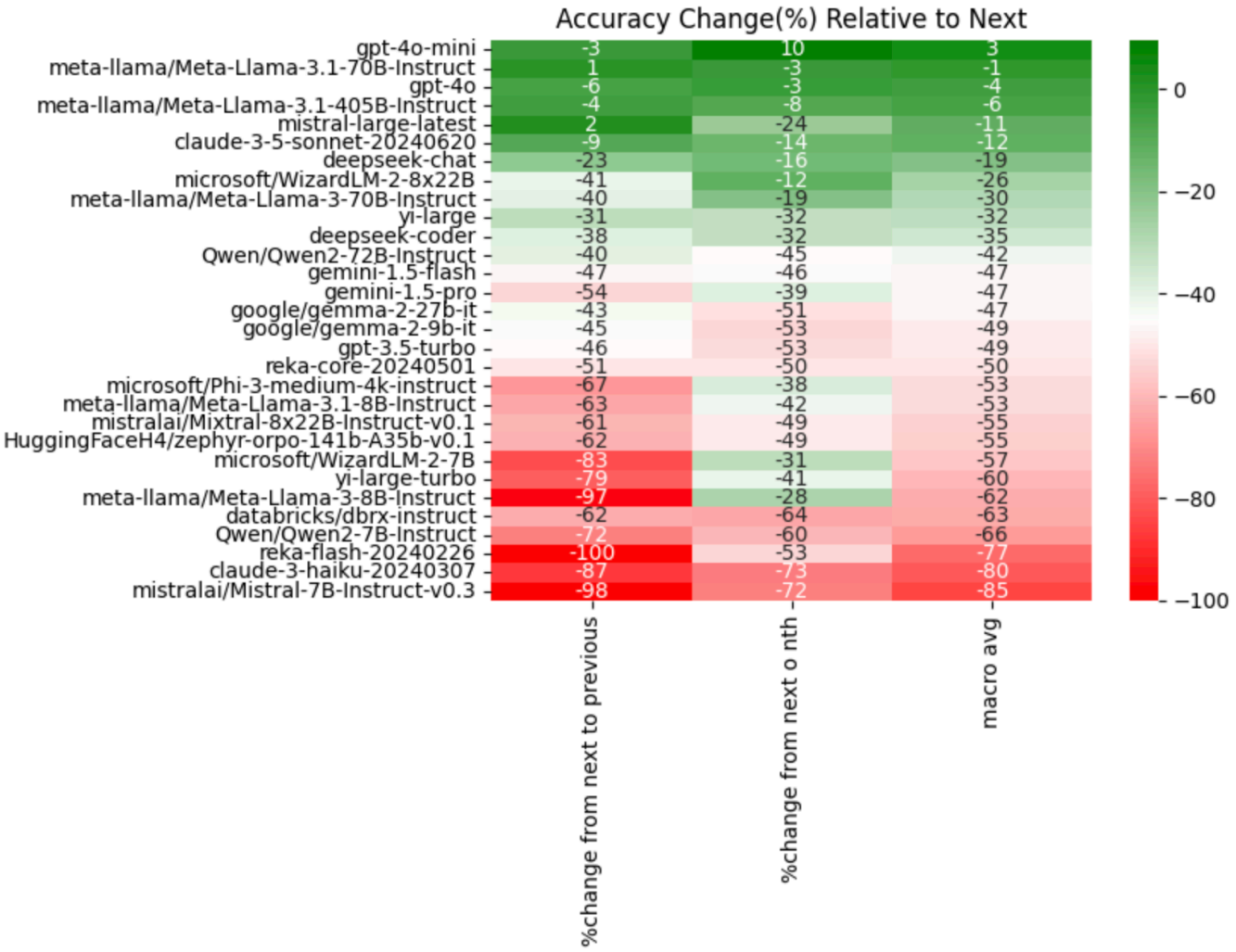
for example, gpt-4o-mini only obtain a F1 score of 0.42 while it achieved an Accuracy of 78.4%





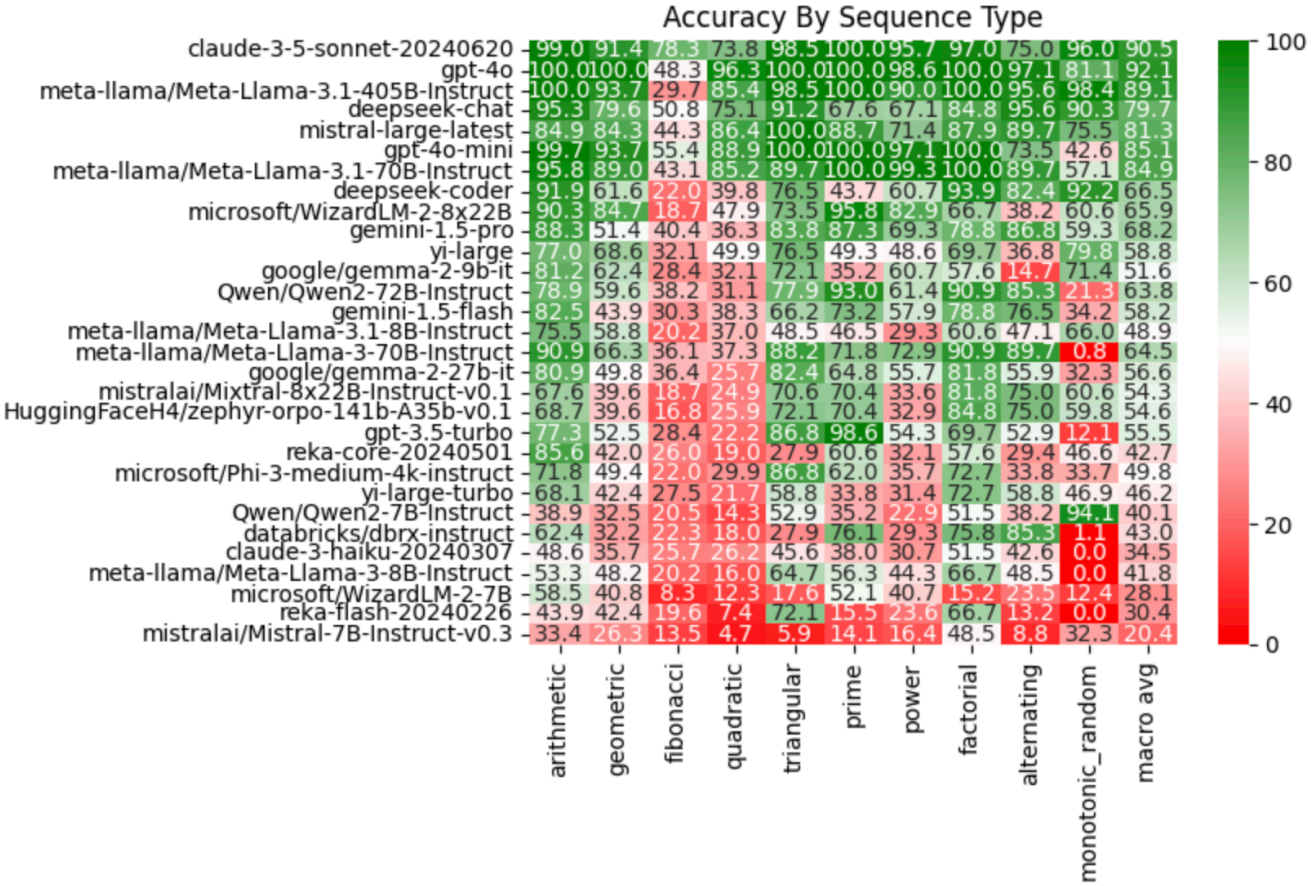
3.2.3 Robustness Across Next vs Nth vs Previous

gpt-4o-mini, Llama3.1 70B, gpt-4o and Llama3.1 450B are the most robust when it comes to long range inference and backward inference. Some model accuracy drops to almost 100% when it is asked to find the previous term. We believe it is combination of a lack of such training data, and also the next token objective in training naturally limits LLM from backward induction in particular.



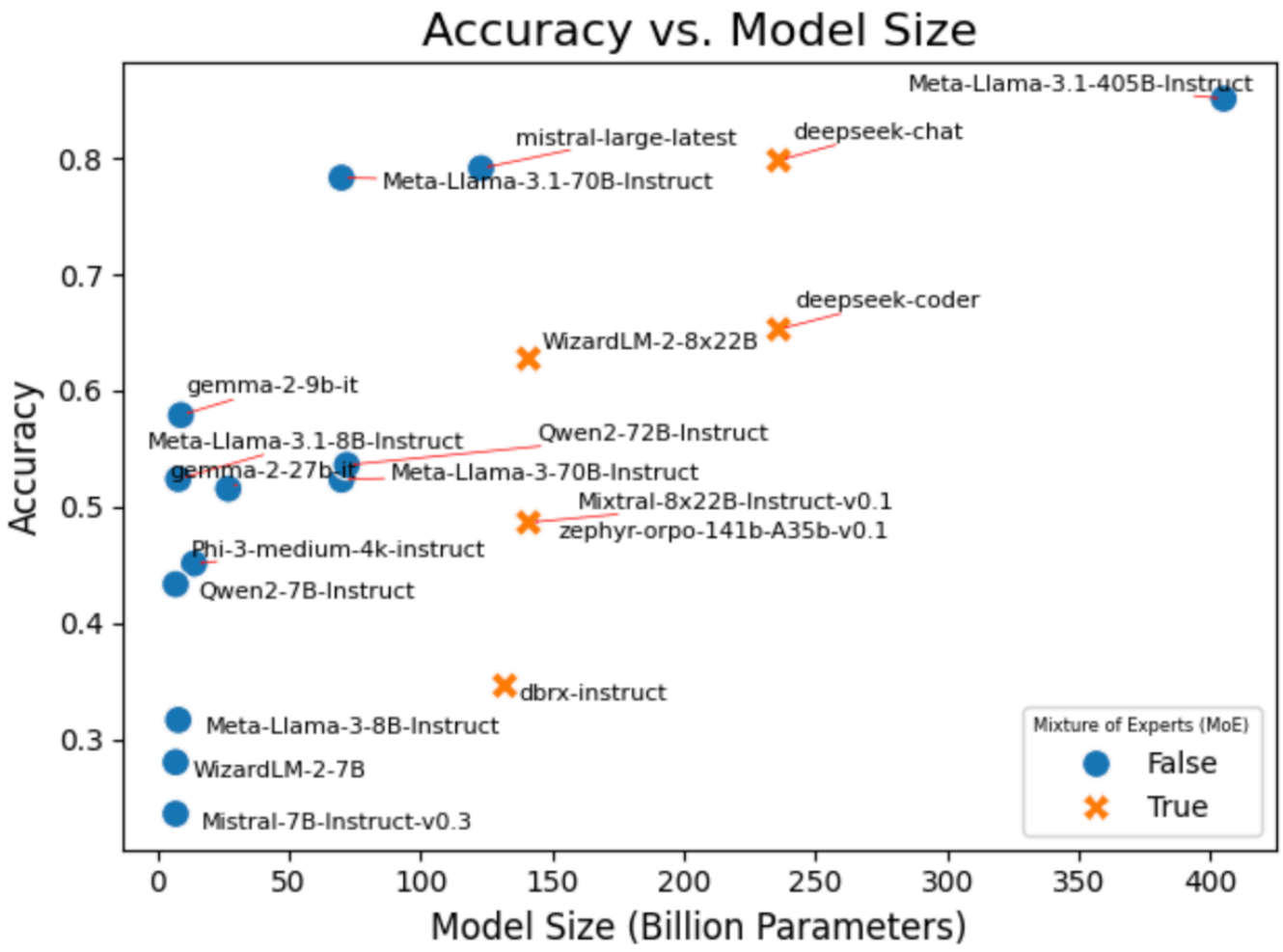
3.2.4 Different Functions

The expectation is that a strong model performs robustly across different functions, but in reality every LLM has its weakbess, Fibonacci sequence is the most difficult one because its value depends on the previous value and the value before i. Only Claude 3.5 Sonnet can perform reasonably well. On the other hand, factorial function value depends on all the previous values. But perhaps its pattern is more obvious and there is not much variable, it is easy to be caught by LLM. Prime is an another interesting question type. Despite the difficulty in computation, its accuracy is relatively high. Does LLM memorise it like human, without actually computing it?



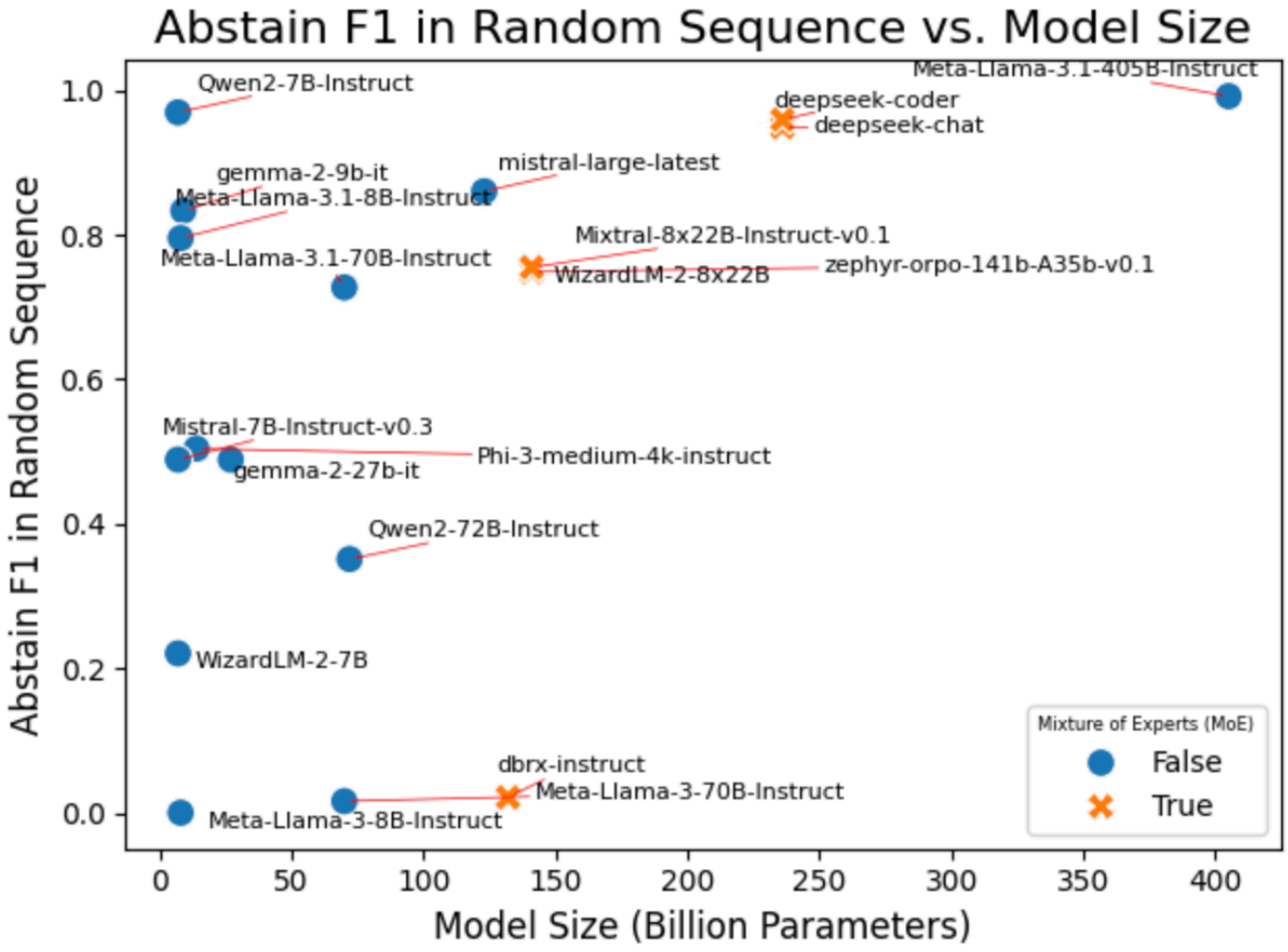
3.2.5 Model Size

Only open source model is studied in this subsection, as the model size of close source model is unknown. Llama3.1 405B performs the best. Despite a moderate correlation (0.66) between accuracy and model size, it does not necessarily mean that accuracy increases with model size. For example, there are large variation of performance of model <= 10B, where gemma-2-9b-it performs the best given its model size.



Similarly, there is no strong relationship (correlation of 0.44) present in Abstrain F1 score. In some cases, small model

perform better than large model.



3.2.6 Data Contamination

While the whole test is and can be programmeticaly generated without reference to web data. The factorial, prime and triangular sequence might be subject to a higher contamination risk because of less parameterisation of these functions.

4. Conclusion

We introduced NumSeqBench to evaluate SOTA LLMs inductive reasoning capability in numeric sequence, including their ability to abstain from answering given a random sequeunce. The SOTA model (Claude 3.5 Sonnet) tops the list with 89.7%, closely followed by gpt-4o (88.0%) and Llama3.1 405B (85.1%). Despite the SOTA performance, it remains a challenging task. In particular, we observe a lot of SOTA models achieved high accuracy in finding the next term, but its accuracy significantly dropped when asked to find next nth term, and previous term. These include models that are top in the open llm leaderboard. We believe it is combination of a lack of such training data, and also the next token objective in training naturally limits LLM from backward induction and long-term induction. We are optimistic that NumSeqBench can help researchers and practitioners to assess and advance inductive reasoning in language model further.

Reference

Reproducibility

Citation

```
@misc{numseqbench2024,
    title={NumSeqBench: Benchmarking Inductive Reasoning in Language Models via Number Sequences},
    author={Ken Tsui},
    url={https://huggingface.co/blog/kenhktsui/numseqbench},
    year={2024}
}
```

Last but Not Least

We will continue to construct more programmatically generated benchmark in the future to evaluate different aspects of reasoning capability of LLMs. Stay tuned!

Appendix

A. List of Models Tested

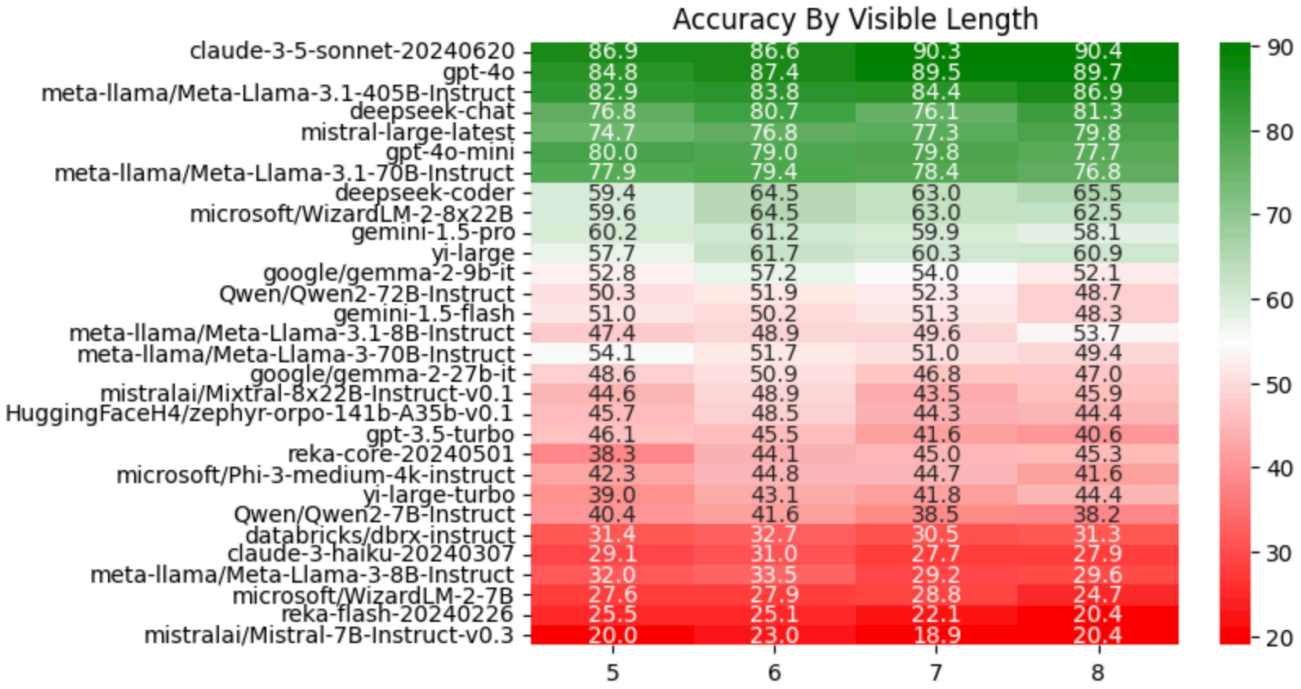
Model Name	Parameters (B)	Open/ Close Model Weight	API Provider Used
claude-3-5-sonnet-20240620	?	Close	Anthropic
claude-3-haiku-20240307	?	Close	Anthropic
databricks/dbrx-instruct	132 (MoE)	Open	DeepInfra
deepseek-chat	236 (MoE)	Open	DeepSeek
deepseek-coder	236 (MoE)	Open	DeepSeek
gemini-1.5-flash	?	Close	Google
gemini-1.5-pro	?	Close	Google
google/gemma-2-27b-it	27	Open	DeepInfra
google/gemma-2-9b-it	9	Open	DeepInfra
gpt-3.5-turbo	?	Close	OpenAI
gpt-4o	?	Close	OpenAI
gpt-4o-mini	?	Close	OpenAi

Model Name	Parameters (B)	Open/ Close Model Weight	API Provider Used
HuggingFaceH4/zephyr-orpo-141b-A35b-v0.1	141 (MoE)	Open	DeepInfra
meta-llama/Meta-Llama-3-70B-Instruct	70	Open	DeepInfra
meta-llama/Meta-Llama-3-8B-Instruct	8	Open	DeepInfra
meta-llama/Meta-Llama-3.1-405B-Instruct	405	Open	DeepInfra
meta-llama/Meta-Llama-3.1-70B-Instruct	70	Open	DeepInfra
meta-llama/Meta-Llama-3.1-8B-Instruct	8	Open	DeepInfra
microsoft/Phi-3-medium-4k-instruct	14	Open	DeepInfra
microsoft/WizardLM-2-7B	7	Open	DeepInfra
microsoft/WizardLM-2-8x22B	141 (MoE)	Open	DeepInfra
mistralai/Mistral-7B-Instruct-v0.3	7	Open	DeepInfra
mistralai/Mixtral-8x22B-Instruct-v0.1	141 (MoE)	Open	DeepInfra
mistral-large-latest	123	Open	Mistral AI
Qwen/Qwen2-72B-Instruct1	72	Open	DeepInfra
Qwen/Qwen2-7B-Instruct	7	Open	DeepInfra
reka-core-20240501	?	Close	Reka AI
reka-flash-20240226	?	Close	Reka AI
yi-large	?	Close	01.ai
yi-large-turbo	?	Close	01.ai

B. Sample Size

C. Instruction Following

D. Accuracy vs. Visible Length Of Sequence



E. Accuracy vs. Inference Length