RELIC: Evaluating Compositional Reasoning via In-Context Language Recognition

Jackson Petty[λ], Michael Y. Hu[δ], Wentao Wang[δ],
Shauli Ravfogel[δ], William Merrill[δ], and Tal Linzen[δλ]

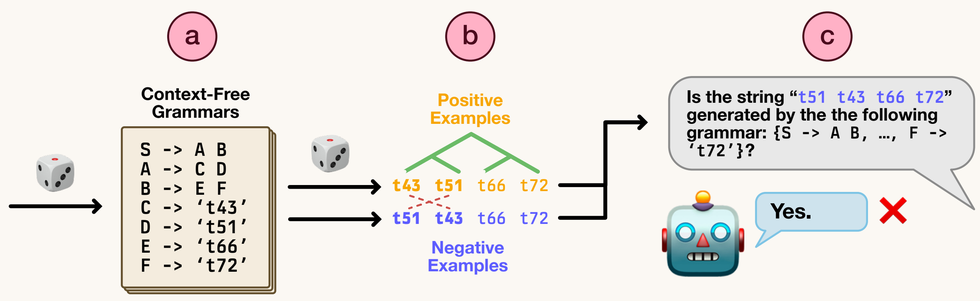
λ *Department of Linguistics*
δ *Center for Data Science*

NYU

How well can LLMs reason through highly-compositional problems as their complexity increases?

1. What is *language recognition*?

Given (a) a **Context-Free Grammar** (CFG) and (b) a **string**, determine whether (c) the grammar **generates** the string.



2. Why study language recognition?

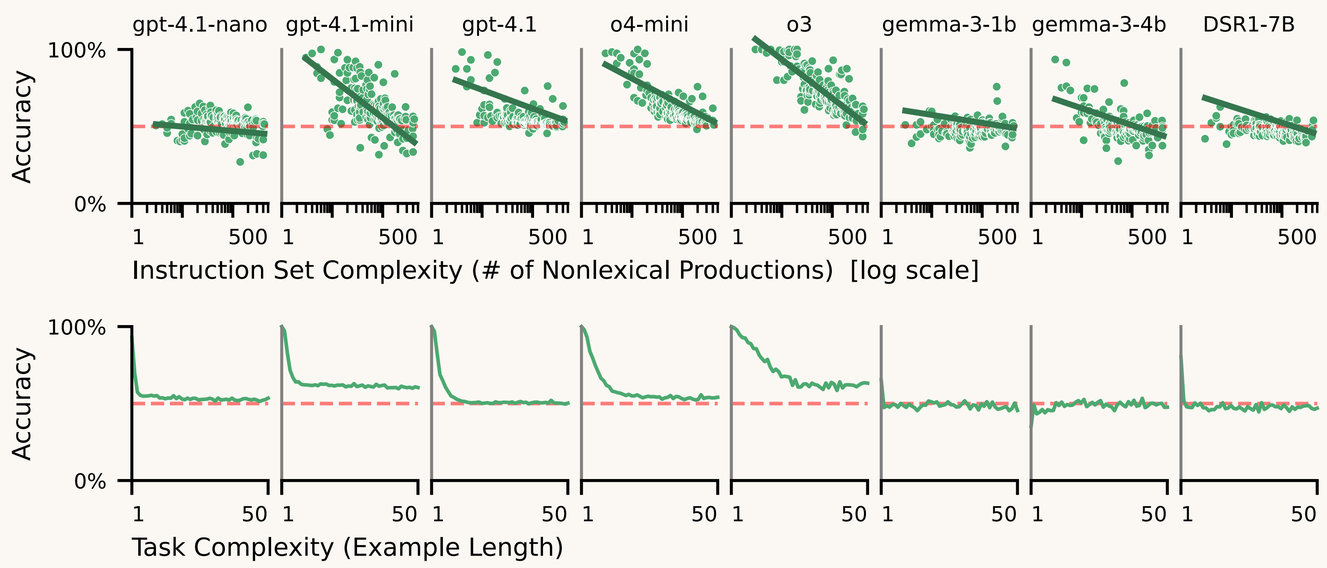
Hard Context-Use Evaluation:
- Non-linear retrieval order
- Recursive or mutually-referential retrieval
- Must search through possible compositions

Easy Formal Verification: Does the string parse?

Embeddable in "fuzzy," hard-to-verify tasks:
- Syntax verification
- Grammaticality judgement
- Generalized instruction following

3. LLMs succeed on simple cases, but fail on complex grammars & strings

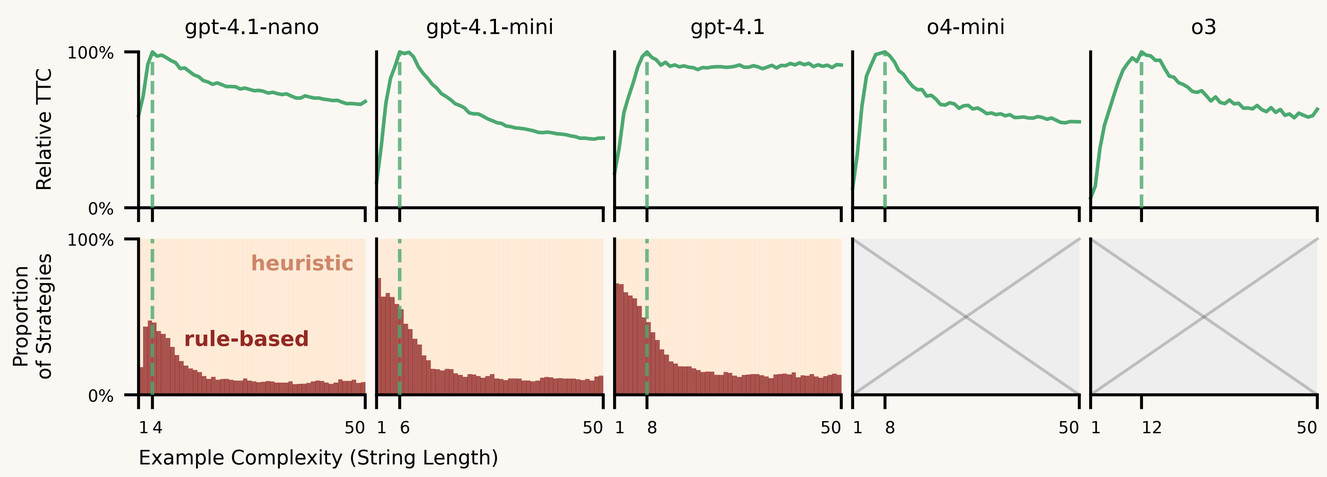
- Models can **succeed** when grammars are very **small** and strings are **short**
- But as **complexity increases** (grammar size and string length), performance falls sharply to **at-chance** accuracy
- LLMs display a **systematic inability** to generalize to more complex cases



4. Why do models get it wrong?

Decreased Compute Expenditure:
- As string complexity increases, TTC peaks and then falls off

Shifting to unsound heuristics:
- LLMs abandon sound rule-based strategies in favor of estimating answer via heuristics



Models "quiet quit" on hard problems, even though they know how to solve them!

Answer: LLMs robustly fail as complexity increases.

In-Context Language Recognition demonstrates how even "reasoning" LLMs fail to solve highly-compositional problems as their complexity increases.