Can Large Language Models reason?

Testing LLMs on CTF Challenges

Aleksandra Taneva
sashita77377@gmail.com
Supervisor: Dr. Z. Erkin
Examiner: Dr. M. Olsthoorn

INTRODUCTION

The problem:
State of the art AI models have scored great on some math benchmarks, beating humans. Recently, they scored worse than human experts on the AICrypto[1] benchmark. We go on to investigate why.

Complex CTF Challenges:
They often include an obscure mathematical protocol, require creative reasoning and precise numerical output. AI needs to be capable of coming up with a strategy, following it exactly, and have an overview of the entire process of solving it.

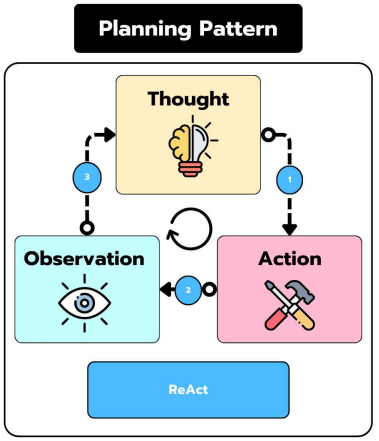
The research question:
- How does access to tools and context within a locally-hosted, open-source agentic framework improve performance on the AICrypto benchmark compared to baseline SOTA LLMs?
- How do these LLMs perform when given access to tools, within a ReAct framework?

THE EXPERIMENT

Base models: Qwen3-32B, Qwen2.5-72B wrapped in a ReAct (Reason–Act–Observe) framework
Our Tools:
- Python
- SageMath
- I/O tools
- Submit_flag

Hardware:
- DelftBlue cluster (2 GPU nodes)

The experiment was conducted on a subset of the AICrypto benchmark.



RESULTS

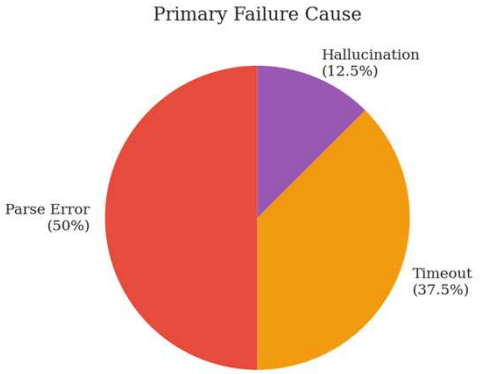
Key Findings
- 0% solve
- Recognizes strategies, can't apply
- Failure modes are codependent, not isolated:
  - *Hallucinated logic*
  - *Tool misuse*
  - *Parsing issues*
  - *Strategic loops*
- Size increase (32B → 72B) doesn't lead to improvements.
- ReAct is too heavy for small models.
- SOTA models use more tools
- Data contamination
- LLMs often bluff [2]

Insight into reasoning:
The small LLMs can't reason in a multi-step, context-aware sense. They don't react to the observed outcomes, they don't self-correct, and further, they can't follow the proposed ReAct framework as we expected.

Knowledge vs Application:
Despite sometimes naming the correct strategy for solving, mapping that to fully functioning code is not possible. It doesn't apply the strategy to code or exhaust all the options.

The models aren't fine-tuned for:
- Tool-usage
- Multi-turn reasoning
- Self-correcting
- Adhering to an output format

Not all models are made for tool usage or complicated improvements. The potential to improve smaller models beyond fine-tuning is underexplored.



FUTURE WORK

Check if these limitations apply to bigger open-source models.
Augment bigger models with In-context learning (RAG), and see if that helps.
Fine-tune a model on a collected database of CTF challenges.

Open Questions:
- Can LLMs truly reason?
- What hides behind the benchmarks' success?

REFERENCES

[1] Yu Wang et al. Aicrypto: A comprehensive bench-mark for evaluating cryptography capabilities of largelanguage models, 2025.
[2] Ivo Petrov et al. Proof or bluff? evaluating llms on 2025 usa math olympiad, 2025.