# E-PAL: Extension of Program-Aided Language Models

Claudia Lee, Jessica Yan, William Zhang
COS484 Final Project

## Introduction

Program-aided Language models (PAL) uses LLMs to decompose problems into runnable Python code and offloads the solution to a Python interpreter which then executes the program to produce the final answer. By decoupling reasoning (done by the LLM) from execution (done by the interpreter), PAL eliminates many of the arithmetic and logical slips that plague chain-of-thought solutions. We extend upon this work by testing PAL's ability to solve two new tasks from the BIG-Bench evaluation suite: **tracking shuffled objects** and **web of lies**.

## Approach

As our study focuses on evaluating tracking shuffled objects and web of lies on PAL, both tasks require logical reasoning compared to pattern matching. For each task, we designed an approach as follows:

1. Design prompt templates that demonstrate to the LLM how to translate the query into the Python snippets.
2. Use the python snippets and execute them to get the final answer.
   a. Model has to be able to handle the decomposition of the problem and we would evaluate how well it did on the code it produced.
3. Compare computed final answer to a baseline where we prompt GPT-4 and GPT4.1 nano on the same inputs, but do not provide any other instructions other than to provide a final answer for evaluation.
   a. Experiments were conducted on GPT-4 and GPT-4.1NANO with a temperature of 0 with no changes in other parameters.
4. Report the accuracy of GPT-4 with PAL and without PAL and use confusion matrices to provide some insight to where PAL falls short.

## Visual Diagram of PAL Approach

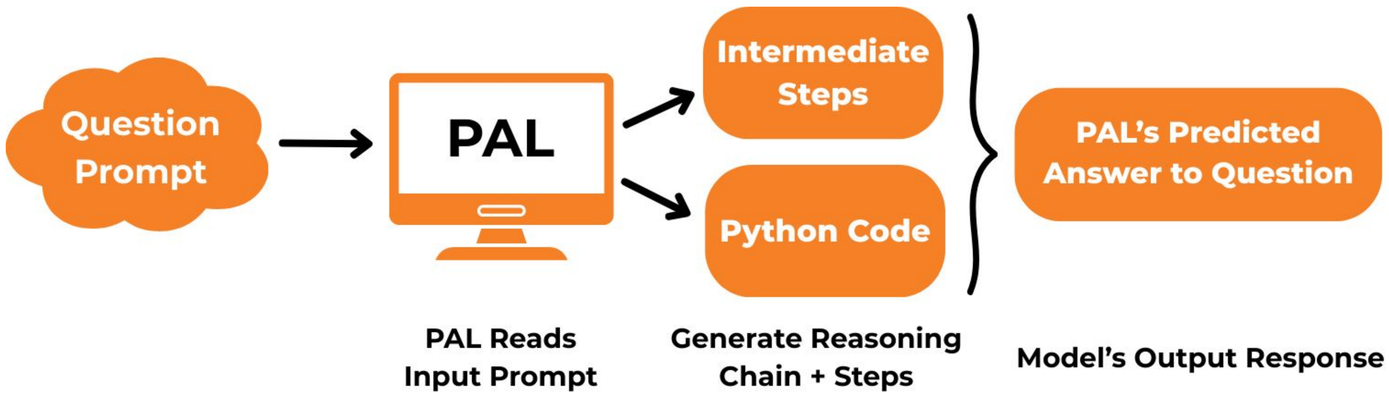


Figure 1: General Approach

## Results

| Prompt | GPT 4 w/o PAL | GPT 4 w/ PAL | 4.1 Nano w/o PAL | 4.1 Nano w/ PAL |
|---|---|---|---|---|
| Tracking Shuffled Objects | 0.519 | 0.608 | 0.366 | 0.612 |
| Web of Lies | 0.596 | 1.0 | 0.524 | 1.0 |

Figure 2: Model Accuracy Metrics with and without PAL for Each Prompt

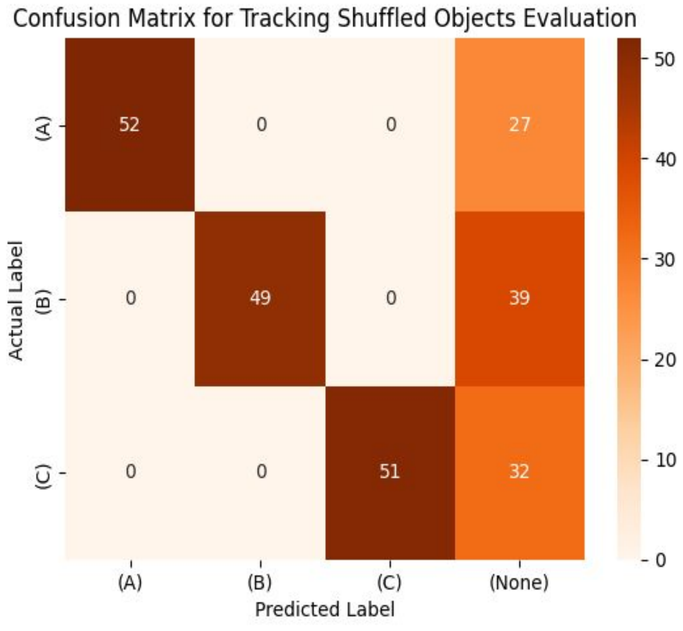


Figure 3: Confusion Matrix for GPT 4 w/ PAL on Tracking Shuffled Objects Prompt

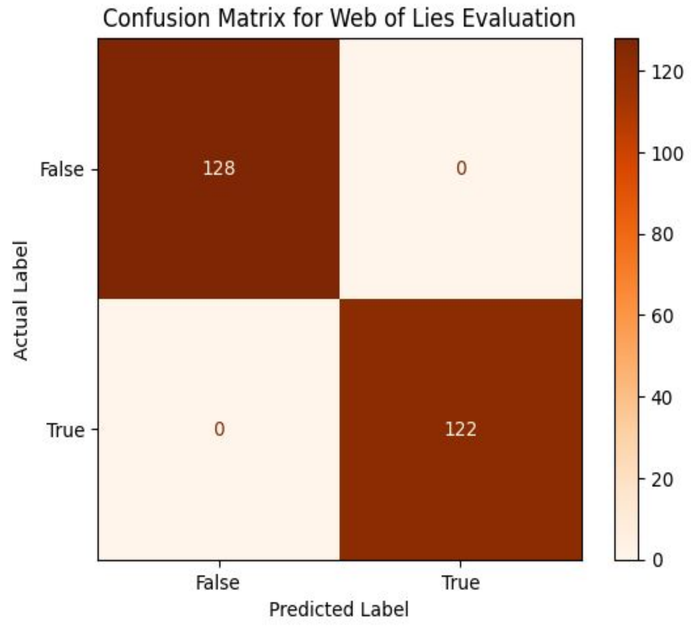


Figure 4: Confusion Matrix for GPT 4 w/ PAL on Web of Lies Prompt

## Conclusions

PAL consistently reduced the logical errors.
For tracking shuffled objects:
- Accuracy boosted from 51.9% to 60.8%
- For nano, accuracy increased 36.6% to 61.2%
- In the confusion matrix, we can see that the mistakes commonly within the None section which means there may have been errors with how the Python code was created by GPT-4 and 4.1 nano.

For web of lies:
- Accuracy was 100% vs baseline of 59.6%/52.4% (nano).
- In the confusion matrix, there was no issues as PAL was able to fully resolve all the inputs with no errors.

These results demonstrate the improvement of LLMs through offloading the execution of the final answer to something like Python interpreter. The approach presented provides a way for new datasets to be added and for other similar logical problems that can be solved by GPT-4 with PAL.

## Future Directions

Through this project, there are various possible next steps, outlined below:
- Potentially improve PAL's performance on tracking objects through associative memory
- Examine how a LLM responds to an action of calling itself through a different prompt.
  - Allows us to better understand how effective it may be for a model to call itself when responding to a question.
- Apply PAL to other multimodal inputs like image or video-based prompts/tasks
- Test on newer reasoning models like DeepSeek R1 and evaluate its performance.
- Try asking GPT to generate in other languages like C. (Python is close to English)
- Identify subsets of prompts that PAL may underperform on and iteratively improve the model in those areas through self-feedback.

## Acknowledgments

We would like to thank the following people for their guidance throughout this project:
- Adithya Bhaskar as our adviser
- Rest of COS484 Staff for allowing a rewarding research exploration