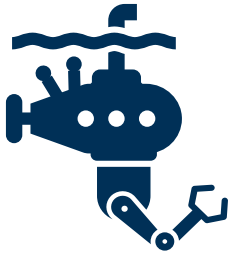
Say, can it run locally?

Hacker Role

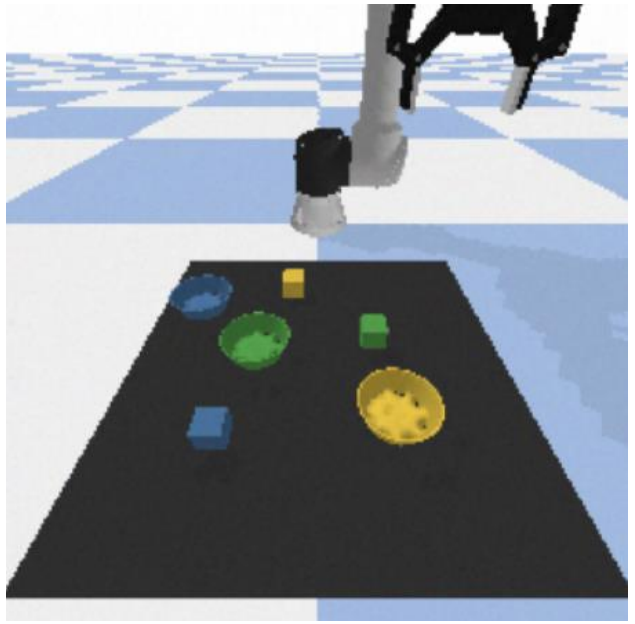
Sean Fish

Motivation

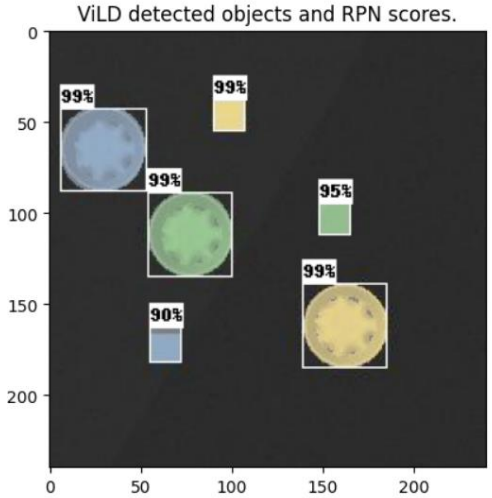
- Field Robotics
  - Aerial, surface, underwater vehicles
  - Not the strongest network connection
    - Lower bandwidth, high latency, intermittent
    - Sometimes not possible
- SayCan uses OpenAI models
  - Not useful without internet
  - Recent innovations in powerful local LLMs
  - I don't want to pay for OpenAI credits



Part 1: Run the Colab Demo

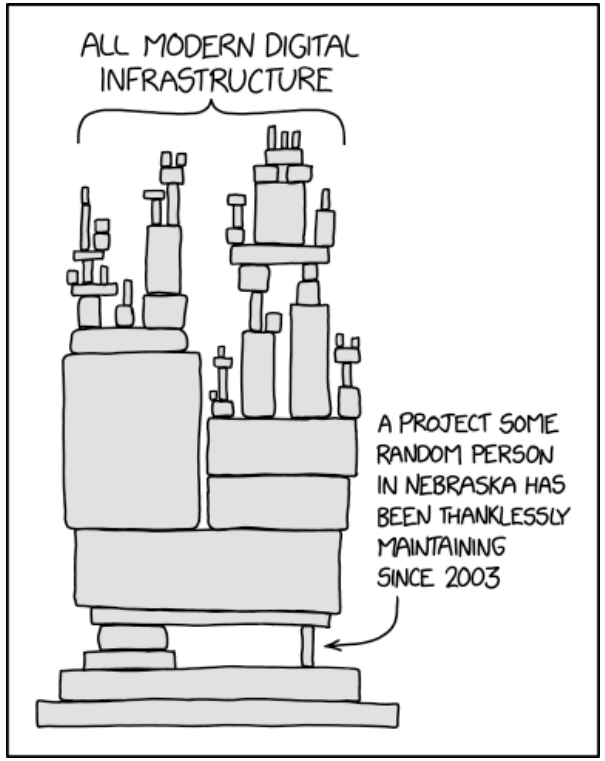


- Simulation environment
  - Robot arm
  - Blocks of various colors
  - Bowls of various colors
- LLM
  - From OpenAI
    - text-ada-001 by default
- Affordance Function
  - No RL affordance
  - CLIPort for actions
  - Object detection from ViLD
    - If found, 1
    - If not found, 0
    - If termination, .2



Part 1: Run the Colab Demo

- Problems on Colab
  - My free OpenAI Credits are expired
    - Even if they weren't, lots of API calls to score
    - OpenAI API has changed too
  - Dependencies have moved on
    - Colab Python version changed
    - Versions not specified
      - Ex: Jax API has changed



XKCD 2347

```
# Note for scoring model, due to limitations of the GPT-3 api, each option
# requires a separate call and can be expensive. Recommend iterating with ada.
```

Part 2: Run the Colab Demo (locally)

- Move to a local notebook
  - Figure out the dependencies
  - CUDA driver incompatibilities
    - Whoops, tensorflow for Python 3.8 does not want CUDA 12
      - Luckily this is not a huge bottleneck
- Figure out how to run an LLM locally
  - Checked out llamafile – start an OpenAI server with one command!
    - No logprobs…
  - Instead, let's use llama-cpp-python server with v1 API
    - Starting with TinyLLama 1.1B
  - Adjust OpenAI API calls for v1 API
    - llama.cpp server does not support lists of prompts!
    - Truncate output as calls require generating one token

The System

- OS: Fedora Silverblue 39
- CPU:
  - Intel Core i5-6600 (released 2015)
  - 4 Cores
  - RAM: 8GB
- GPU:
  - Nvidia GTX 1060 **6GB** (released 2016)
  - CUDA Cores **1280**
    - For reference, Nvidia Jetson AGX Orin has 2048
  - CUDA Version 12.4

Test Case

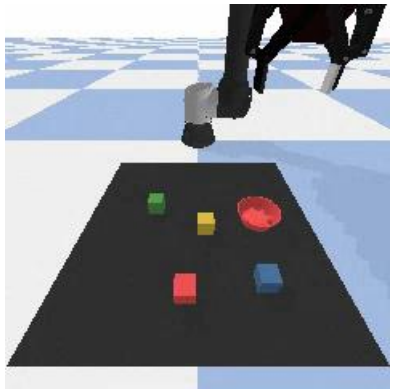
- Top 5 actions out of 53
- “To pick the blue block and put it on the red block, I should:”

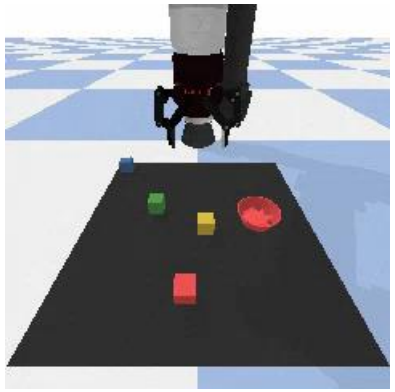
Logprobs	Action
-16.40082025527954	done()
-39.61580505082384	robot.pick_and_place(red block, blue block)
-40.683303487021476	**robot.pick_and_place(blue block, red block)**
-42.85621754499152	robot.pick_and_place(green block, red block)
-43.30987396882847	robot.pick_and_place(red block, green block)

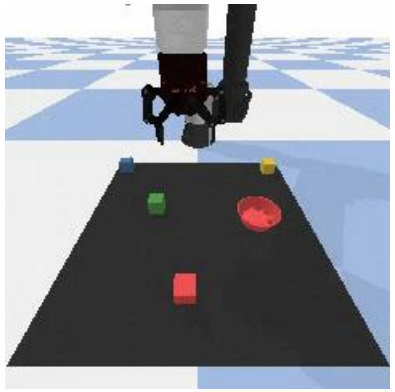
The Demo Task

- “put all the blocks in different corners”

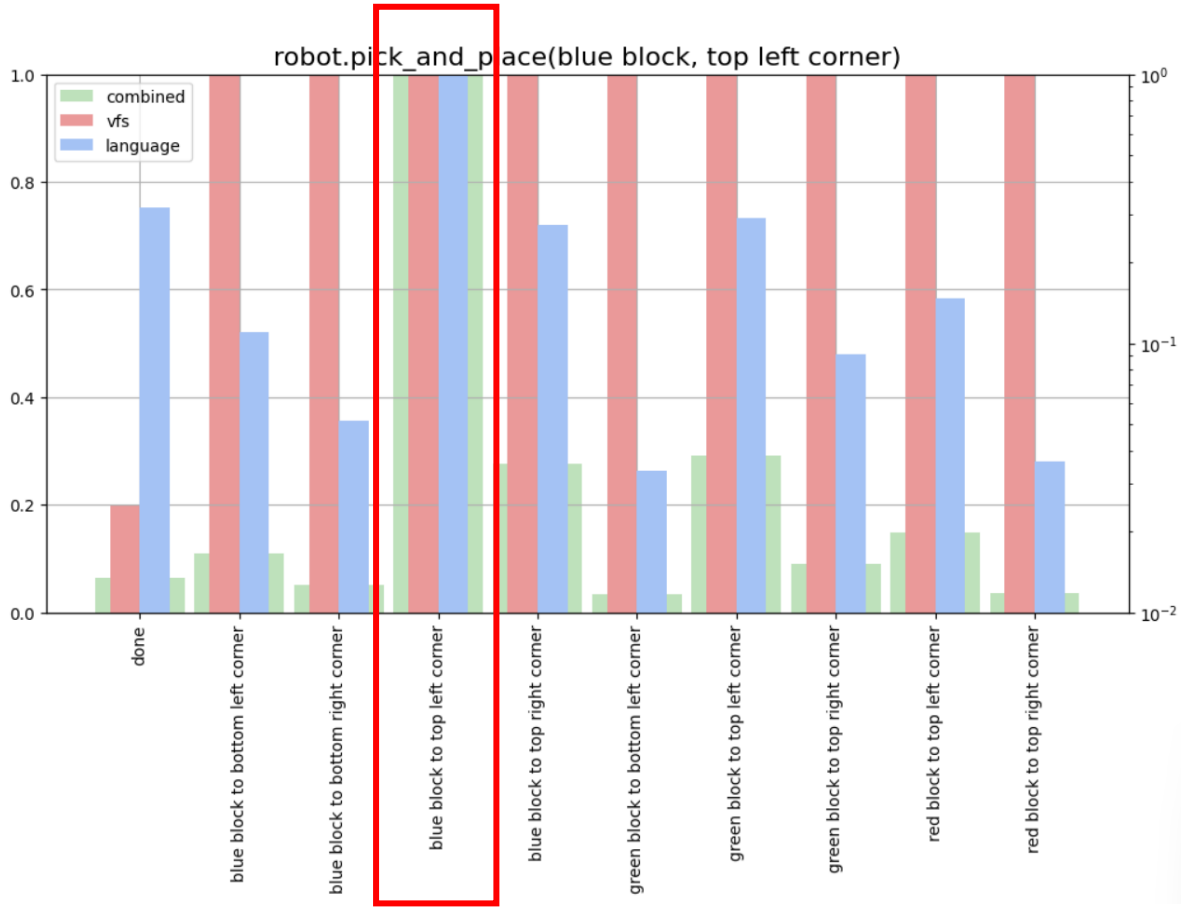
```
**** Solution ****
objects = [yellow block, green block, red bowl, blue
block, red block]
# put all the blocks in different corners.
Step 0: robot.pick_and_place(blue block, top left corner)
Step 1: robot.pick_and_place(red block, top left corner)
Step 2: robot.pick_and_place(green block, top left
corner)
Initial state:
GPT-3 says next step: Pick the blue block and place it on
the top left corner.
```

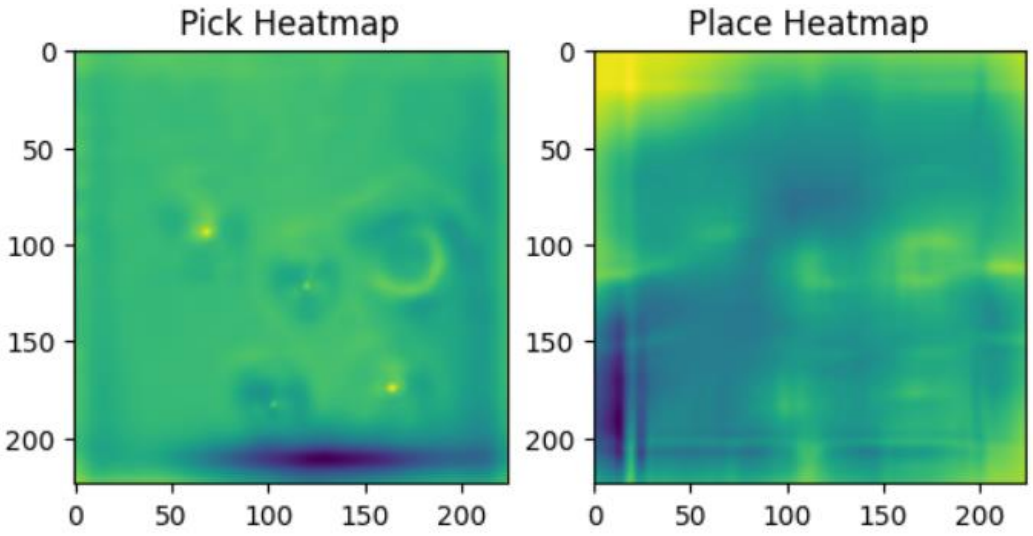






The Demo Task (More Outputs)





It worked!

- What did I learn from running locally?
  - Don't run out of vRAM
  - It takes forever (unless logprob scores are cached)
  - But it does seem to work (somewhat)
- Limitations
  - Might be related to the LLM
  - The affordance functions are another big limiting factor
    - Overly rely on LLM to discriminate between actions
  - I did not work with the Socratic Model section

Part 3: Compare some local models

- What should we look at?
  - Time
    - From start to finish (without cache)
  - Performance
    - Number of blocks in corners
    - Number of blocks in unique corners
- Which models?
  - TinyLlama 1.1B
  - Rocket-3B
  - Phi-2

Local LLMs

TinyLlama 1.1B Q5_K_M

Parameters: 1.1B
Size: 0.78GB
Max RAM: 3.28GB
Datasets:

- starcoderdata
- SlimPajama
- oasst
- UltraChat
- UltraFeedback

Rocket-3B Q5_K_M

Parameters: 3B
Size: 1.99GB
Max RAM: 4.49GB
Datasets:

- A mix of datasets
- UltraFeedback
- JudgeLM

Phi-2 Q5_K_M

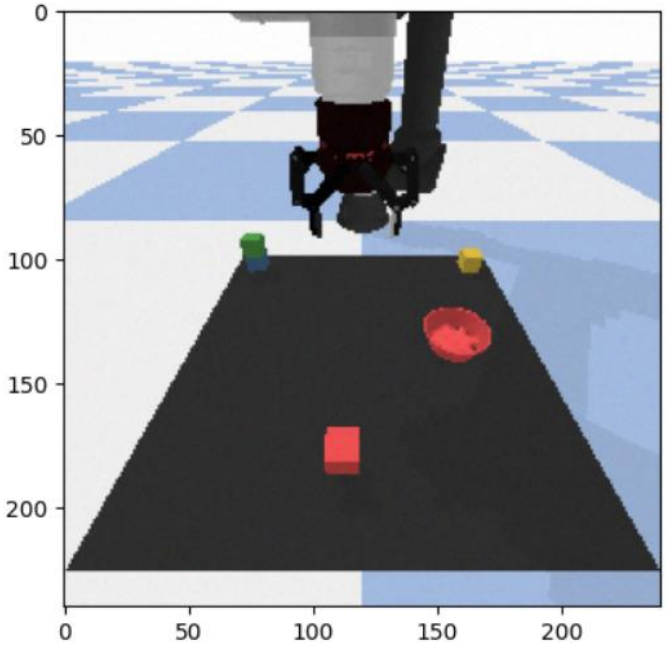
Parameters:
Size: 2.07GB
Max RAM: 4.57GB
Datasets:

- Textbooks
- Falcon RefinedWeb
- SlimPajama

Local LLMs

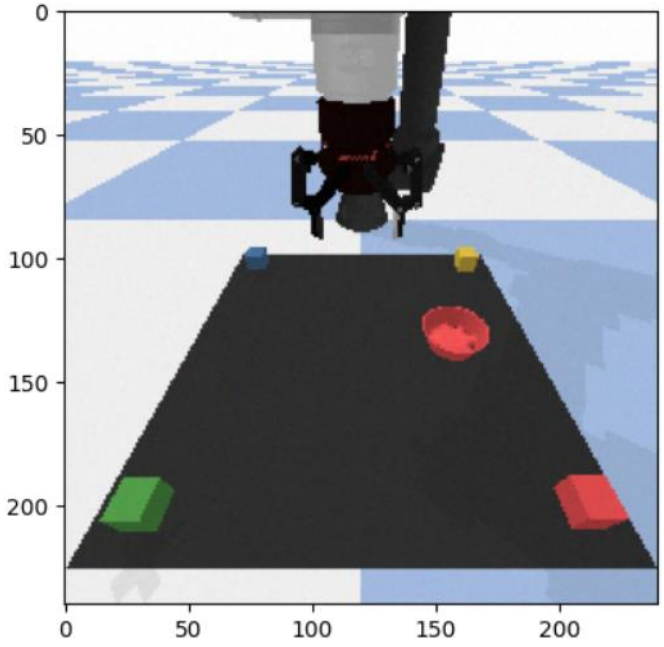
TinyLlama 1.1B Q5_K_M

Mean time/step: 9m38s
No. of steps: 4
No. blocks in corners: 3
No. corners w/ blocks: 2



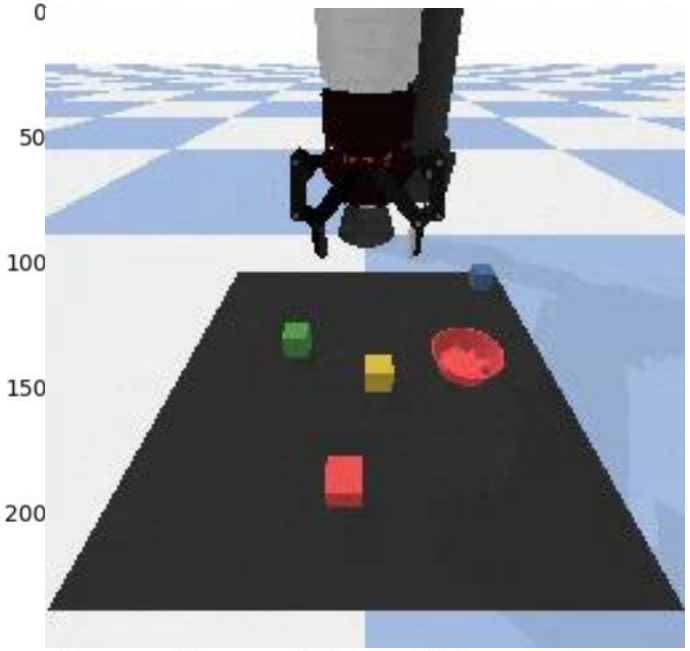
Rocket-3B Q5_K_M

Mean time/step: 16m48s
No. of steps: 5
No. blocks in corners: 4
No. corners w/ blocks: 4



Phi-2 Q5_K_M

Mean time/step: 16m31s
No. of steps: 4
No. blocks in corners: 2
No. corners w/ blocks: 1



Conclusions

- Can it run locally? Yes
- Does it run in a reasonable time frame? Not necessarily
  - This is an area to investigate for low resource systems
- Which model to use?
  - Apparently Rocket-3B – but this is a limited experiment
- Get more vRAM
- GitHub
  - github.com/seantfish/CS8803DLM-HackSayCan
  - Don't leave your OpenAI key here
    - OpenAI finds out very fast

Resources and Citations

- TheBloke/TinyLlama-1.1B-Chat-v1.0-GGUF · Hugging Face
- TheBloke/rocket-3B-GGUF · Hugging Face
- TheBloke/phi-2-GGUF · Hugging Face
- Phi-2: The surprising power of small language models - Microsoft Research
- google-research/saycan at master · google-research/google-research (github.com)