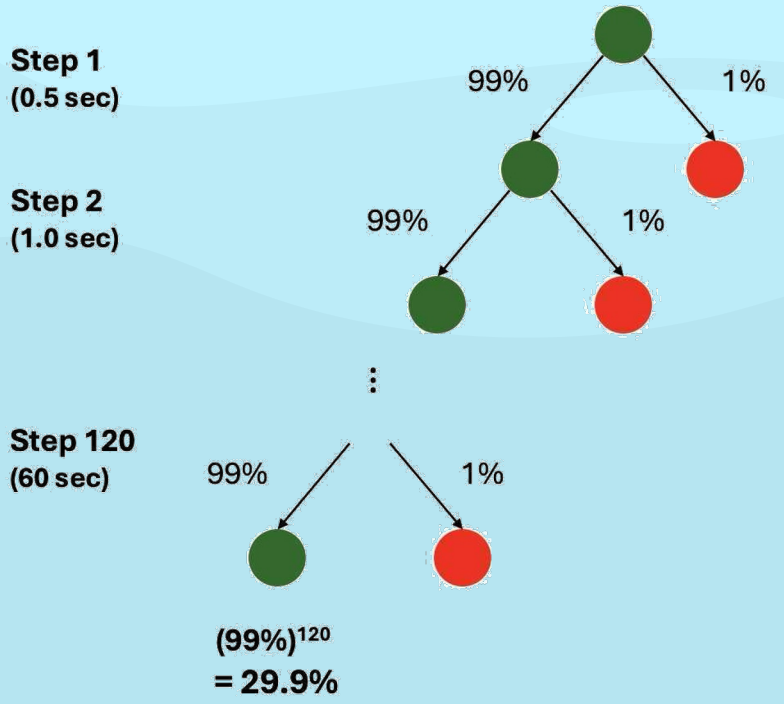
Agent Reasoning Pt 1: Neurosymbolic Tool Use

Prithviraj Ammanabrolu

Problem 3: Long Horizons

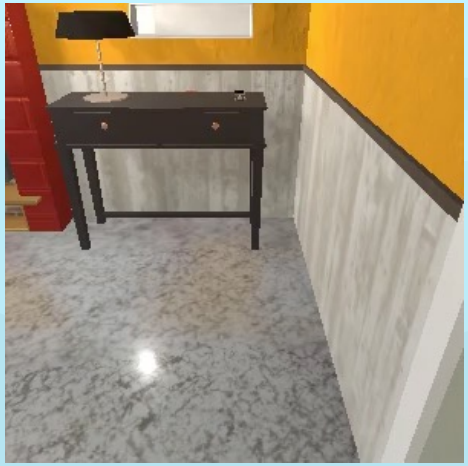
Optimizing across entire steps and sequences of thought

Local optimization, e.g. single step tool calls will not help reasoning

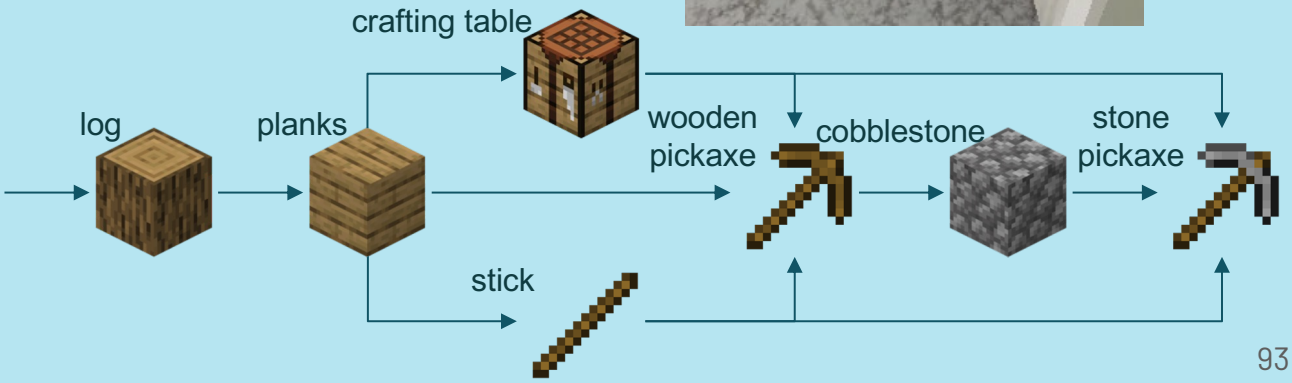


Challenges in Grounded Environments

Relatively Low Level Control:



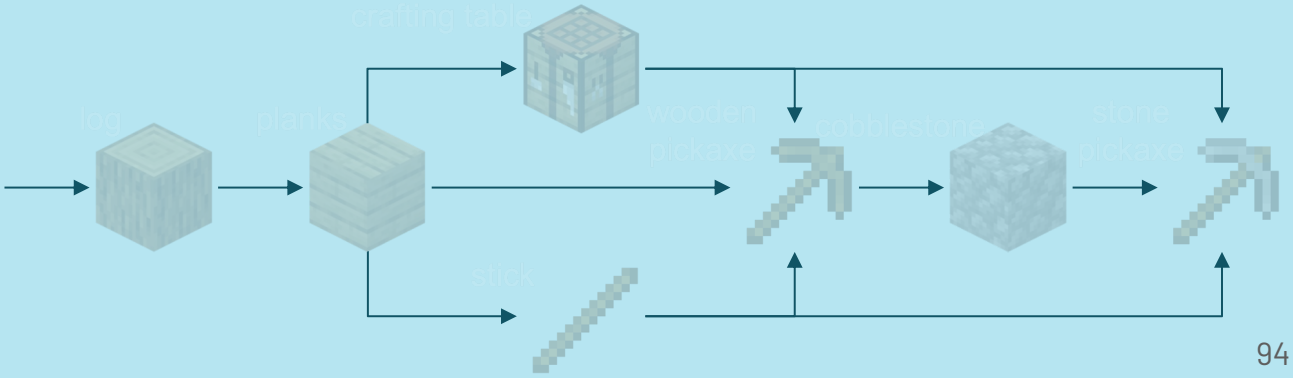
High Level Planning:



Challenges in Grounded Environments

Low Level Control:

High Level Planning:



Key Idea:
Neurosymbolic Control

USE SYMBOLIC TOOLS/PLANNERS FOR LOW LEVEL CONTROL AS INTERFACE FOR HIGH LEVEL LLM

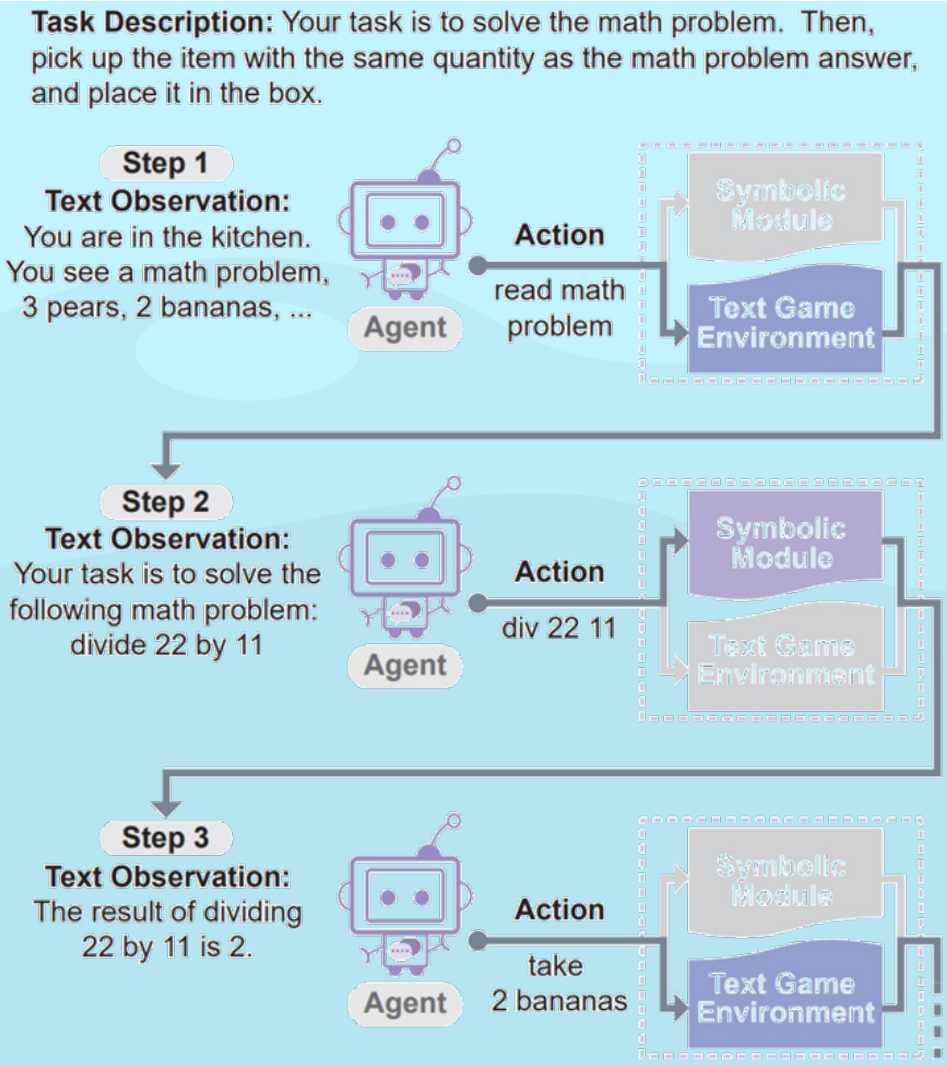
Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2022.

Key Idea:
Neurosymbolic Control

USE SYMBOLIC TOOLS/PLANNERS FOR LOW LEVEL CONTROL AS INTERFACE FOR HIGH LEVEL LLM

WHY TEACH AN LM TO ADD NUMBERS WHEN CALCULATORS EXIST???

Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2022.



Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2022.

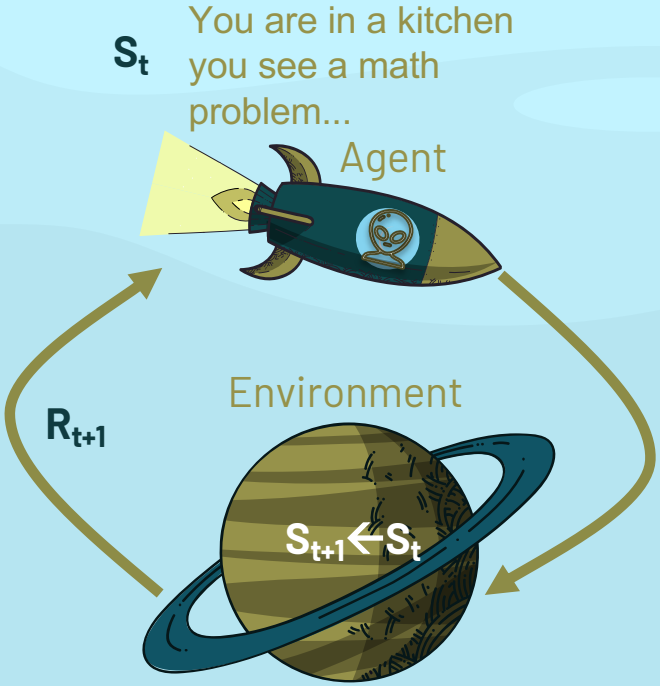
Simplification to step-level Markov Decision Process (MDP)

6-tuple of <S, A, T, R, γ, K>:

- S states = steps so far
- A words = step
- T transition fn
- R reward function
- γ discount factor
- K max sentence length

Objective: Find policy π_θ: S → A to maximize long term expected rewards

$$\mathbb{E}_\pi[\sum_{t=0}^{K} \gamma^t R(s_t, a_t)]$$



The result is 2. Next you need to find …

Tasks and Tools

Navigation (GPS)

Retrieval (Database lookup)

Arithmetic (Calculator)

Sorting

Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2022.

Results: Supervised Learning vs. Neurosymbolic Control

	Baseline		NeuroSymbolic	
Benchmark	Score	Steps	Score	Steps
MapReader	0.71	27	**1.00**	**10**
Arithmetic	0.56	5	**1.00**	**5**
Sorting	0.72	7	**0.98**	**8**
TWC	0.90	6	**0.97**	**3**
Average	0.72	11	**0.99**	**7**

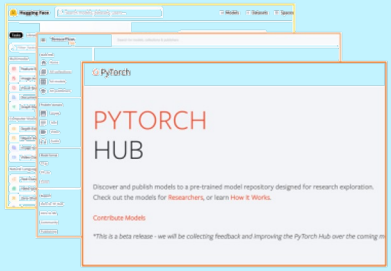
Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2022.

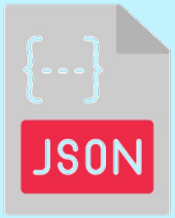
How to evaluate tool use?

- Did your function call match the ground truth function call?
- Did you function call(s) do the task it was supposed to?

How to evaluate tool use?

- Did your function call match the ground truth function call?
  - Some kind of matching with ground truth
- Did you function call(s) do the task it was supposed to?
  - Execution output match, needs env

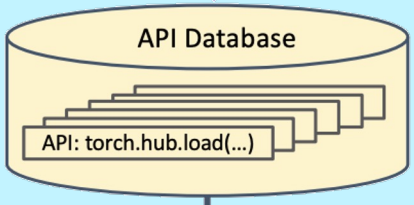




Dataset curation: 1,645 API calls. 94 from Torch Hub (exhaustive), 626 from TensorFlow Hub v2 (exhaustive) and 925 from HuggingFace (Top 20 in each domain).

Self-instruct with in-context examples to generate 16,450 {instruction,API} pairs

This is then used to train Gorilla-7B



"I want to see some cats dancing in celebration!"

Information Retriever

Input:
###Task: Generate image from text
###Reference API:
StableDiffusionPipeline.from_pretrained (...)

GORILLA

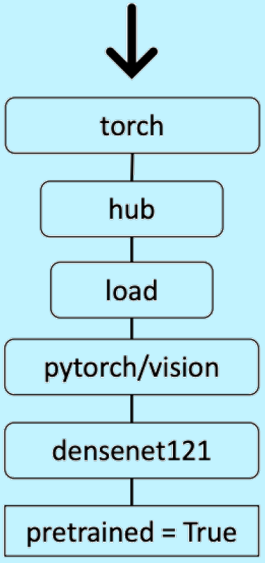
API: `StableDiffusionPipeline.from_pretrained(stabilityai/stable-diffusion-2-1)`

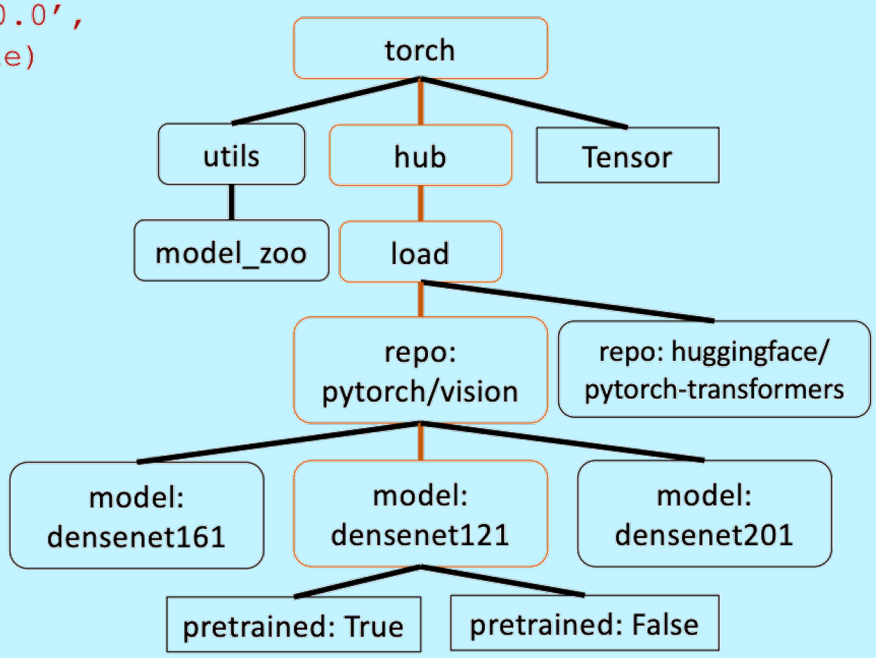
Execution Results!

Zero-shot

How to evaluate tool use?

```
torch.hub.load('pytorch/vision:v0.10.0',
       'densenet121', pretrained=True)
```



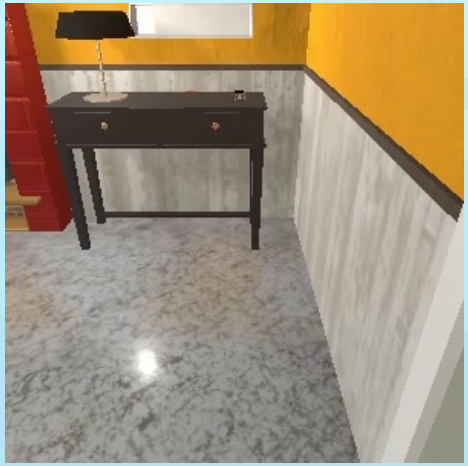


Issues with such evals

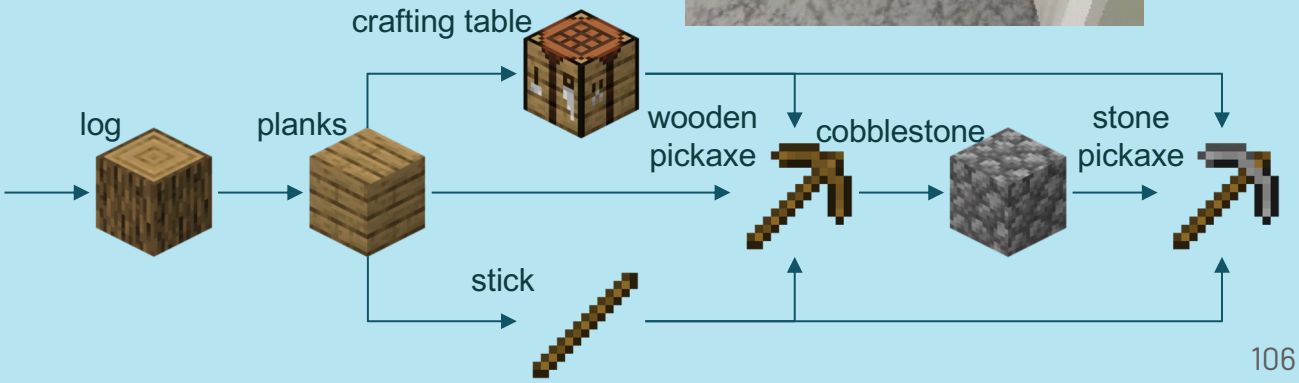
- BFCLv1/2 are one step – long horizon issue. Even v3 latest one is only ~4-5 steps long on average
- Sub-tree matching with ground truth only gets you so far. You need an env.
- Something like SWE Bench (Gym) is more realistic – tradeoff is that many realistic Github scenarios don't have verifiable solutions

Challenges in Grounded Environments

Relatively Low Level Control:



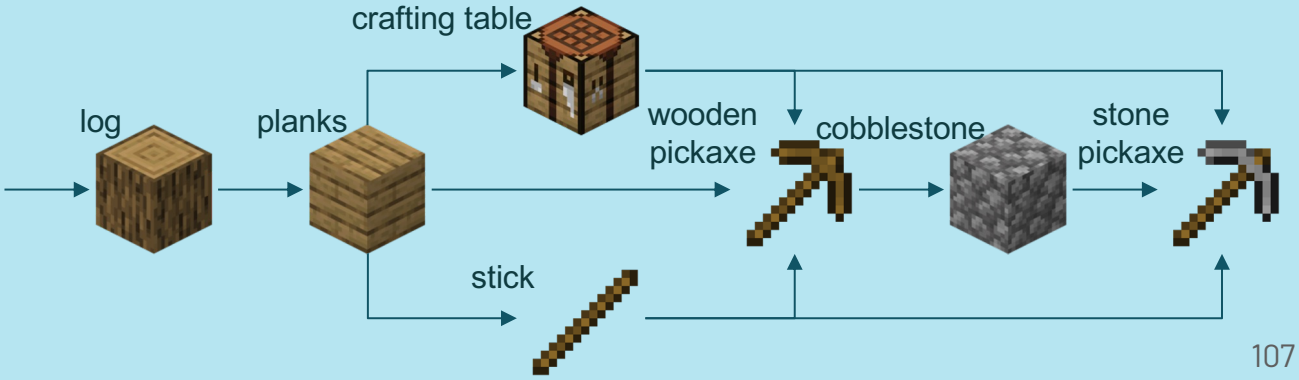
High Level Planning:



Challenges in Grounded Environments

Low Level Control

High Level Planning:



Using World Knowledge

How can agents intelligently interact with common objects and characters? E.g.:

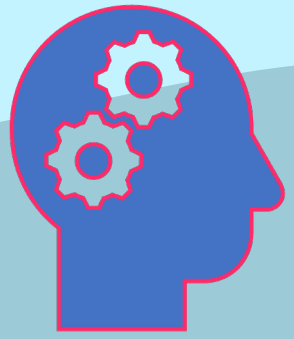
Affordance Extraction: given an object, identify reasonable verbs to apply to that object:

"open mailbox" vs "eat mailbox"

Social Commonsense Knowledge: given a character, identify reasonable utterances to apply to them. E.g. talking to a king:

"My lord" vs "My dude"

Ammanabrolu and Riedl *Transfer in Deep Reinforcement Learning with Knowledge Graphs. TextGraphs@EMNLP-19*
Ammanabrolu et al. *How to Motivate Your Dragon: Teaching Goal-Driven Agents to Act and Speak in Fantasy Worlds.* NAACL-21



Hypothesis

AGENTS WITH INTERNAL WORLD MODELS PRODUCE LANGUAGE NEEDED FOR INTERACTION MORE ACCURATELY

Key Idea:
Abstract World Model

USE WORLD KNOWLEDGE IN LLM TO KICK START AN AGENT'S EXPLORATION OF THE WORLD

BREAK DOWN GOAL INTO SMALLER SUBGOALS THAT ARE EASIER TO SOLVE

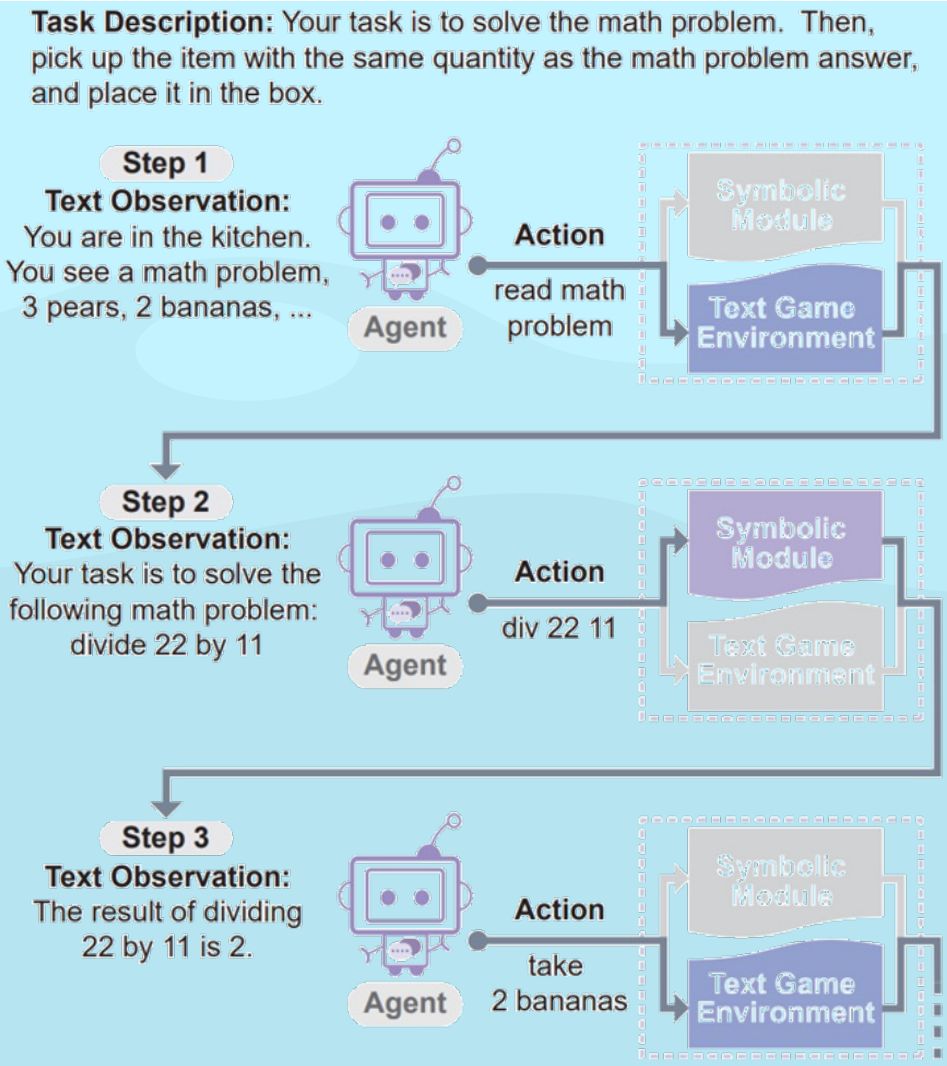
Nottingham, Ammanabrolu, Suhr, Hajishirzi, Choi, Singh, Fox. *Do Embodied Agents Dream of Pixelated Sheep?: Embodied Decision Making using Language Guided World Modelling.* ICML 2023.

Grounding Language

When there's no human feedback, you learn from **Environmental Feedback**!

High level planning through an LLM as an **Abstract World Model**

Low level control through **Neurosymbolic Planning**



Wang, Jansen, Alexandre-Cote, Ammanabrolu. *Behavior Cloned Transformers are Neurosymbolic Reasoners.* EACL 2021.

RL vs Planning

Reinforcement Learning:

- The environment is initially unknown
- The agent interacts with the environment
- The agent improves its policy

Planning:

- An (estimated) model of the environment (T) is known
- The agent performs computations with its model (without any external interaction)
- The agent improves its policy a.k.a. deliberation, reasoning, introspection, pondering, thought, search

(Classical) Planning

Can be with Search i.e. BFS, DFS, A*

Can be with Dynamic Programming

- Requires overlapping subproblems with optimal compositionality.
- The more your problem has these properties, the more “reasoning” will help

None of these scale

(Reinforcement) Learning + Planning

- Model based RL – Explore, learn transition function from rollouts, T: S x A → S
- Do planning on that
- World models, Alpha Zero, etc etc all belong here
- But wait LLMs kinda are their own T (and also policy)

Tool Use Reasoning

- LLMs are their own Transition + policy. But policy changes a bit with tools conceptually
- In reality, it doesn't matter because you can just give LLMs descriptions of tool <input, output> and it should be able to anticipate outputs
- Big question is what are good intermediate state representations for a given task? And how good are LLMs at anticipating outputs?

PDDL Recap

- Standard encoding for classic planning tasks
- Many specific languages for creating simulations have similarities with PDDL
- Syntax of the language isn't as important as the core concepts (most good LLMs can take care of syntactic sugar)

What's in a PDDL task?

- Objects: Things in the world that interest us.
- Predicates: Properties of objects that we are interested in; can be true or false.
- Initial state: The state of the world that we start in.
- Goal specification: Things that we want to be true.
- Actions/Operators: Ways of changing the state of the world.

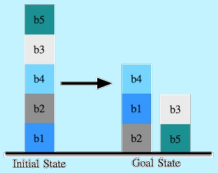
2 .pddl files, domain and problem

LLM Planning

- LLMs (no thinking) are rather bad at planning
- Intuition why is
  - that they have been trained with no n-step DP style look ahead
  - transition matrix size is Huge, anticipating next world states is hard

A Failure Example of GPT-4 in Planning

Problem (P1): You have 5 blocks. One cannot place more than one block on another block. b5 is on top of b3. b4 is on top of b2. b2 is on top of b1. b3 is on top of b4. b1 is on the table. b5 is clear. Your arm is empty.
Your goal is to move the blocks.
b1 should be on top of b2.
b3 should be on top of b5.
b4 should be on top of b1.



GPT-4:

```
1. Move b5 from b3 to the table.
2. Move b3 from b4 to b5.
3. Move b4 from b2 to the table.
4. Move b2 from b1 to b5.
5. Move b1 from the table to b2.
6. Move b4 from the table to b1.
```

Liu et al. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. 2023.

The Transition Matrix is B I G

Back of envelope calc:

Obs: 100000^{512} (512 tokens x 50k vocab)

Action: 1000^{5} (5 tokens x 1000 vocab)

Naïve lower bound of dim of T is $O(S^2A) = (100000^{1024})(1000^{5})$

$= 10^{5135}$

Ammanabrolu and Riedl *Playing Text-Adventure Games with Graph-based Deep Reinforcement Learning.* NAACL-19

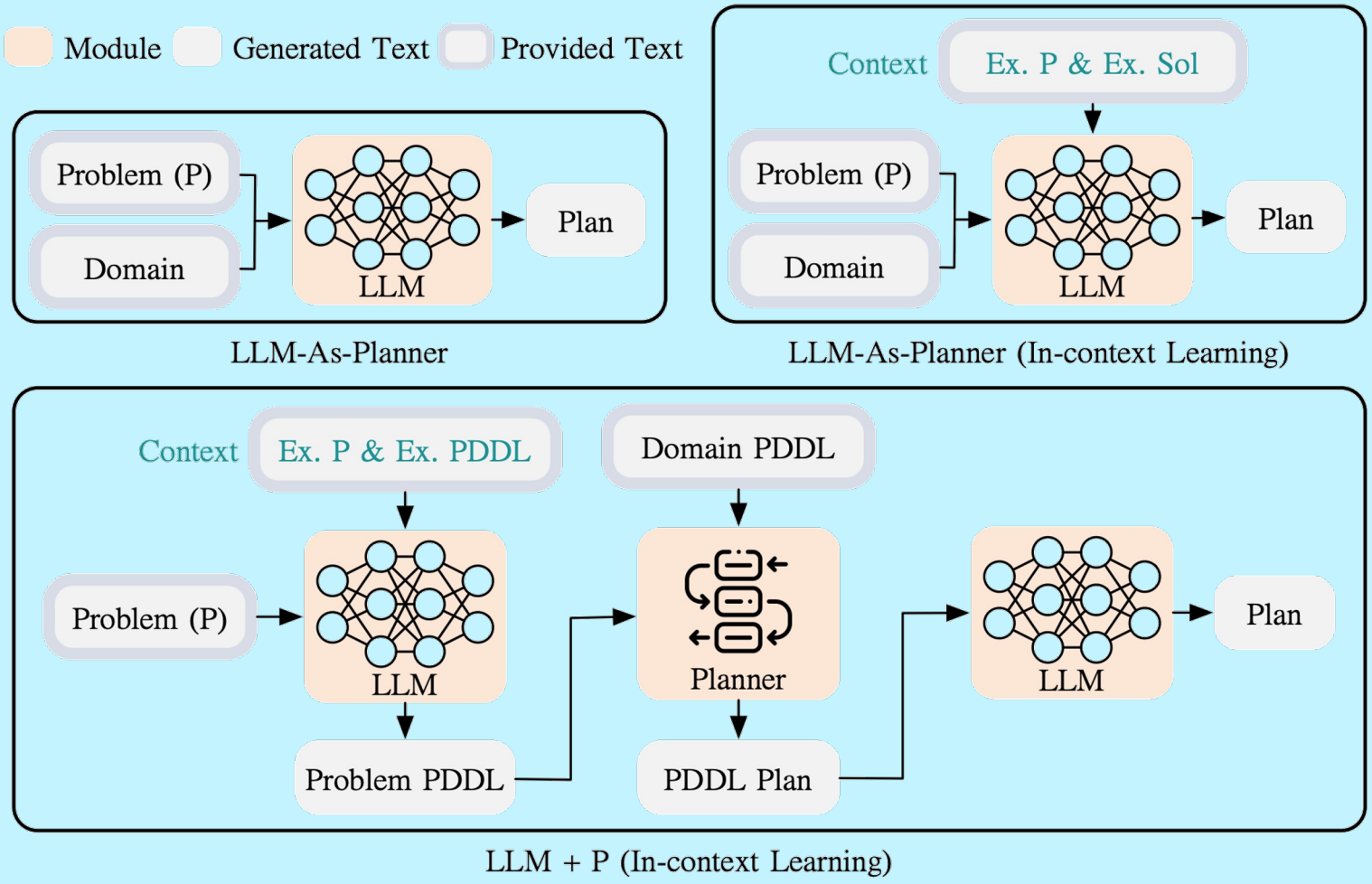
Outside the Real Estate Office

A grim little cul-de-sac, tucked away in a corner of the claustrophobic tangle of narrow, twisting avenues that largely constitute the older portion of Anchorhead. Like most of the streets in this city, it is ancient, shadowy, and leads essentially nowhere. The lane ends here at the real estate agent's office, which lies to the east, and winds its way back toward the center of town to the west. A narrow, garbage-choked alley opens to the southeast.

>*go southeast*

Alley

This narrow aperture between two buildings is nearly blocked with piles of rotting cardboard boxes and overstuffed garbage cans. Ugly, half-crumbling brick walls to either side totter oppressively over you. The alley ends here at a tall, wooden fence. High up on the wall of the northern building there is a narrow, transom-style window.



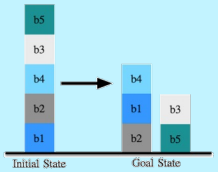
Liu et al. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. 2023.

LLM Planning

- LLMs (no thinking) are rather bad at planning
- Intuition why is
  - that they have been trained with no n-step DP style look ahead
  - transition matrix size is Huge, anticipating next world states is hard
    - Make rank of T smaller

A Failure Example of GPT-4 in Planning

Problem (P1): You have 5 blocks. One cannot place more than one block on another block. b5 is on top of b3. b4 is on top of b2. b2 is on top of b1. b3 is on top of b4. b1 is on the table. b5 is clear. Your arm is empty.
Your goal is to move the blocks.
b1 should be on top of b2.
b3 should be on top of b5.
b4 should be on top of b1.



GPT-4:

```
1. Move b5 from b3 to the table.
2. Move b3 from b4 to b5.
3. Move b4 from b2 to the table.
4. Move b2 from b1 to b5.
5. Move b1 from the table to b2.
6. Move b4 from the table to b1.
```

Liu et al. LLM+P: Empowering Large Language Models with Optimal Planning Proficiency. 2023.

The Transition Matrix is B I G

Back of envelope calc:

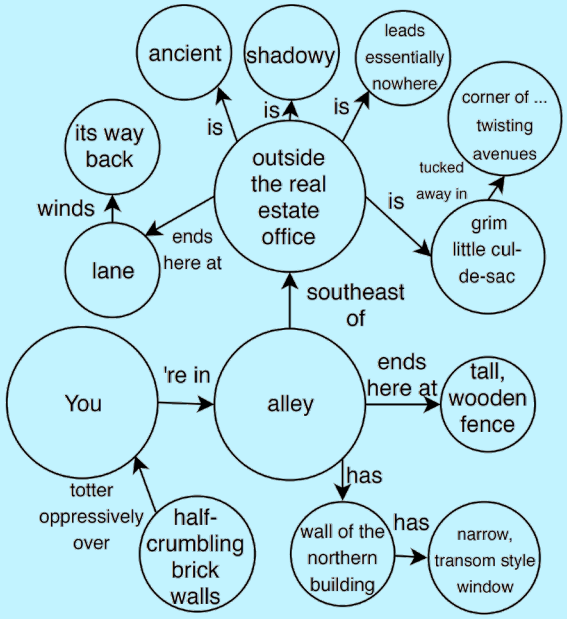
Obs: 100000^{512} (512 tokens x 50k vocab)

Action: 1000^{5} (5 tokens x 1000 vocab)

Naïve lower bound of dim of T is $O(S^2A) = (100000^{1024})(1000^{5})$

$= 10^{5135}$

Ammanabrolu and Riedl *Playing Text-Adventure Games with Graph-based Deep Reinforcement Learning*. NAACL-19



Outside the Real Estate Office

A grim little cul-de-sac, tucked away in a corner of the claustrophobic tangle of narrow, twisting avenues that largely constitute the older portion of Anchorhead. Like most of the streets in this city, it is ancient, shadowy, and leads essentially nowhere. The lane ends here at the real estate agent's office, which lies to the east, and winds its way back toward the center of town to the west. A narrow, garbage-choked alley opens to the southeast.

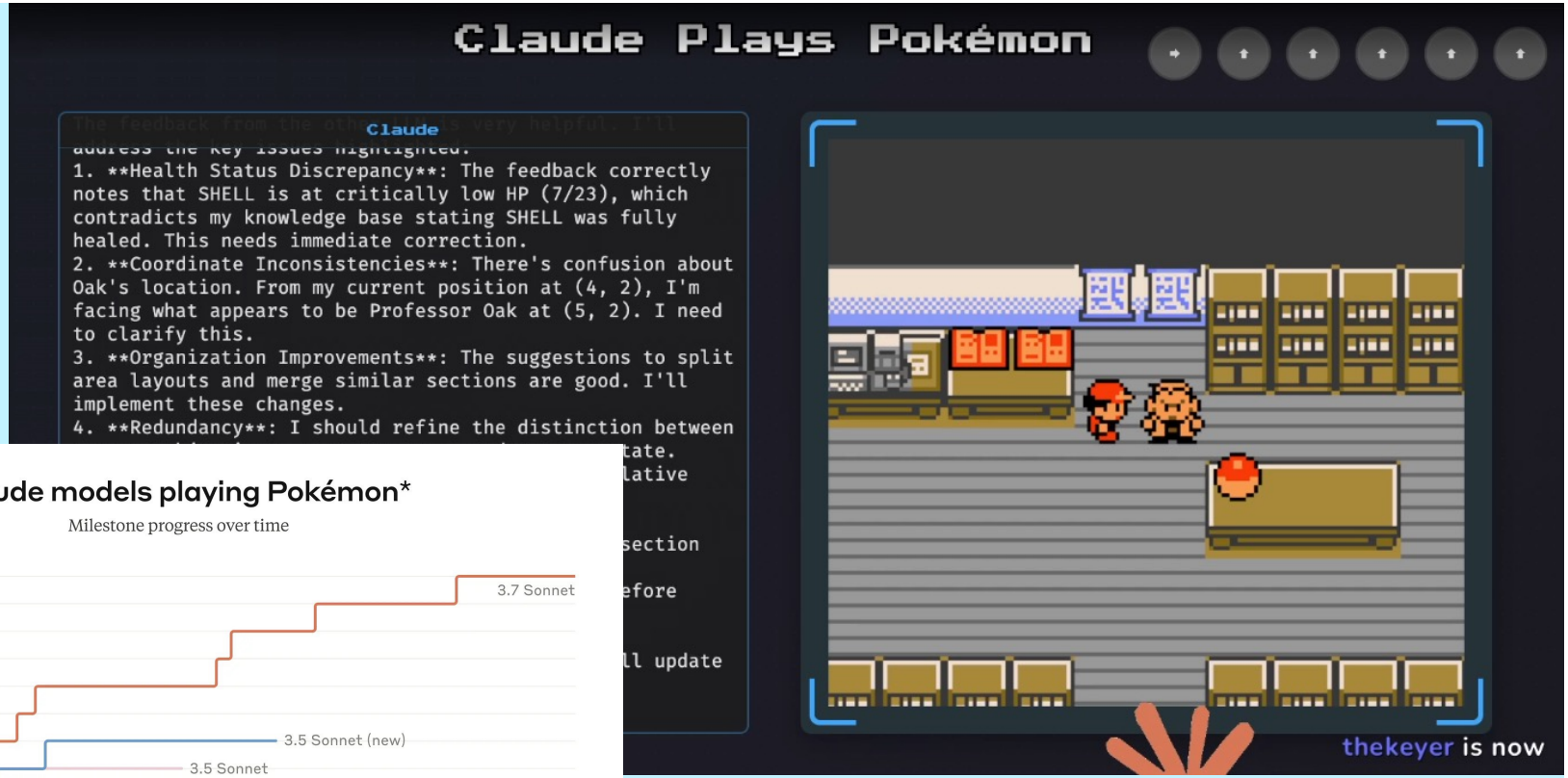
>go southeast

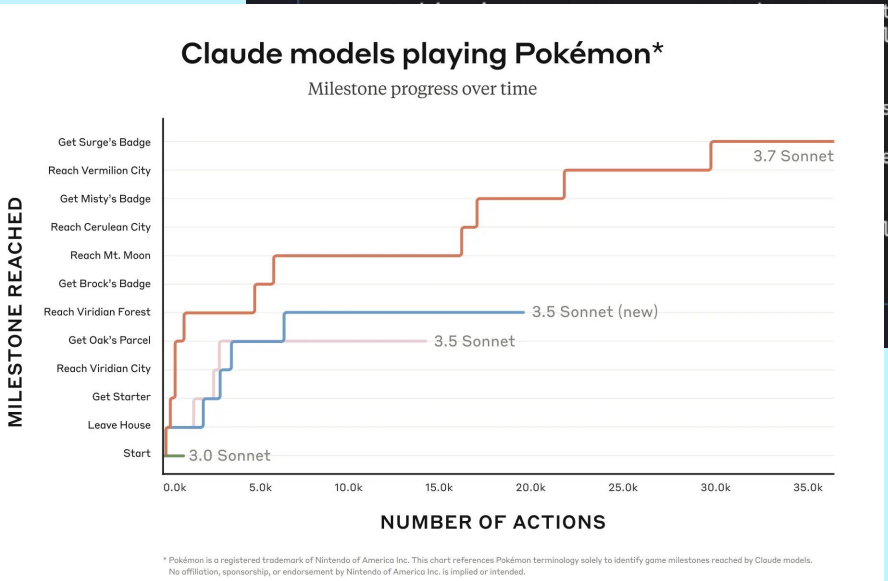
Alley

This narrow aperture between two buildings is nearly blocked with piles of rotting cardboard boxes and overstuffed garbage cans. Ugly, half-crumbling brick walls to either side totter oppressively over you. The alley ends here at a tall, wooden fence. High up on the wall of the northern building there is a narrow, transom-style window.

Alternate Reasoning Approaches

- Are there other (possibly complementary) methods to try to bake this “look ahead” into LLMs?
- Yes! Next time. Inference time scaling or “thinking”!


Claude Plays Pokémon
Claude
1. **Health Status Discrepancy**: The feedback correctly notes that SHELL is at critically low HP (7/23), which contradicts my knowledge base stating SHELL was fully healed. This needs immediate correction.
2. **Coordinate Inconsistencies**: There's confusion about Oak's location. From my current position at (4, 2), I'm facing what appears to be Professor Oak at (5, 2). I need to clarify this.
3. **Organization Improvements**: The suggestions to split area layouts and merge similar sections are good. I'll implement these changes.
4. **Redundancy**: I should refine the distinction between
thekeyer is now


Claude models playing Pokémon*
Milestone progress over time
MILESTONE REACHED
Get Surge's Badge
Reach Vermilion City
Get Misty's Badge
Reach Cerulean City
Reach Mt. Moon
Get Brock's Badge
Reach Viridian Forest
Get Oak's Parcel
Reach Viridian City
Get Starter
Leave House
Start
3.7 Sonnet
3.5 Sonnet (new)
3.5 Sonnet
3.0 Sonnet
0.0k
5.0k
10.0k
15.0k
20.0k
25.0k
30.0k
35.0k
NUMBER OF ACTIONS
* Pokémon is a registered trademark of Nintendo of America Inc. This chart references Pokémon terminology solely to identify game milestones reached by Claude models. No affiliation, sponsorship, or endorsement by Nintendo of America Inc. is implied or intended.