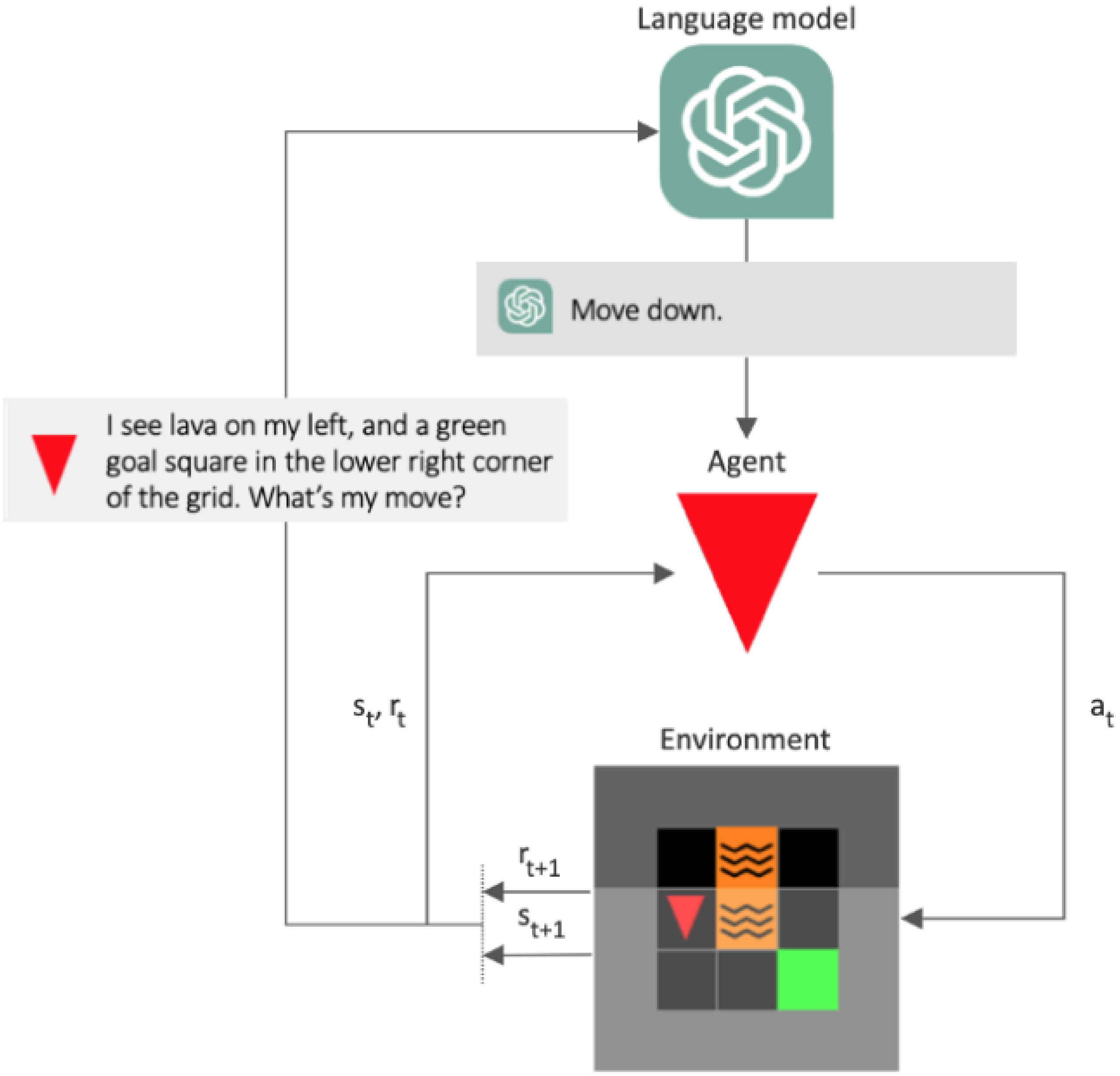
Integrating Large Language Models into Reinforcement Learning

Gregor Kajda, Jonatan Hoffmann Hanssen, Adrian Duric

Supervisors: Katrine Nergård, Kai Olav Ellefsen

Aim of the Project

- RL in large environments
  - Large state and action spaces
  - Poor sampling efficiency
- LLMs can make the agent try smarter actions
- Our goal: integrate an LLM into the RL framework



Guiding Pretraining in Reinforcement Learning with Large Language Models

Project 11

Integrating Large Language Models into Reinforcement Learning

Paper overview

Danijar Hafner 2021, *Crafter*. Screenshot by author.
MIT License

Paper overview

- Presents a method for using Large Language Models to explore a 2D environment more intelligently

Danijar Hafner 2021, *Crafter*. Screenshot by author.
MIT License

Paper overview

- Presents a method for using Large Language Models to explore a 2D environment more intelligently
- LLM suggests actions for the agent to take based on a description of the state

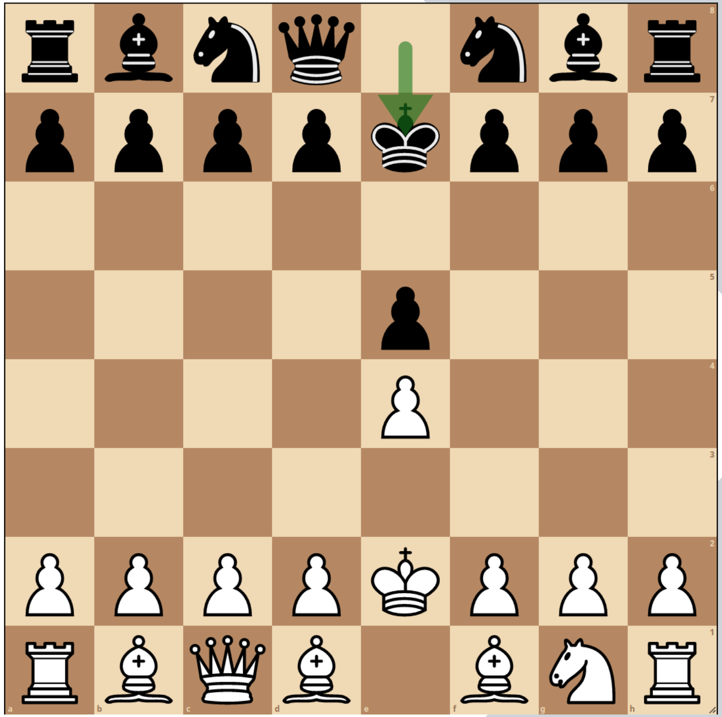
Danijar Hafner 2021, *Crafter*. Screenshot by author.
MIT License

Paper overview

- Presents a method for using Large Language Models to explore a 2D environment more intelligently
- LLM suggests actions for the agent to take based on a description of the state
- Improvements over other methods

Danijar Hafner 2021, *Crafter*. Screenshot by author. MIT License

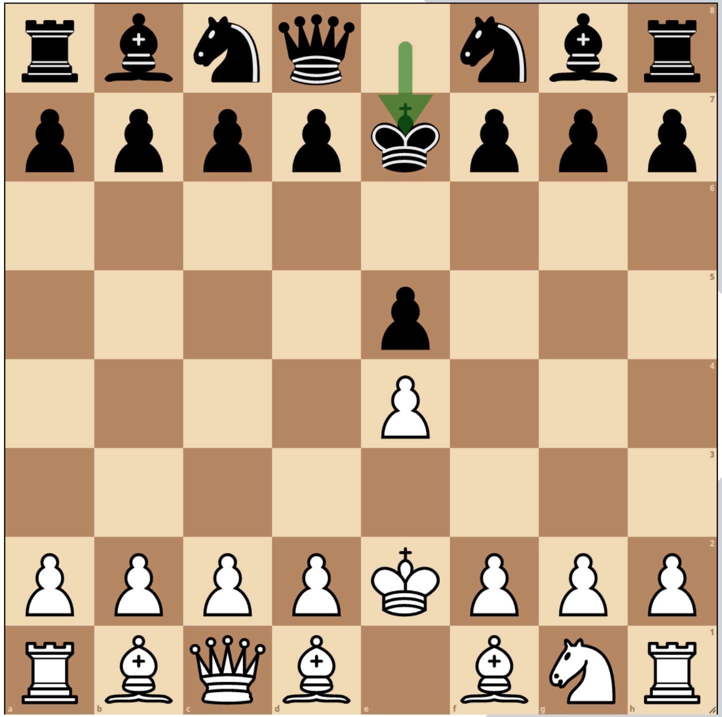
Motivation



Lichess 2023, *Double Bongcloud*. Screenshot by author

Motivation

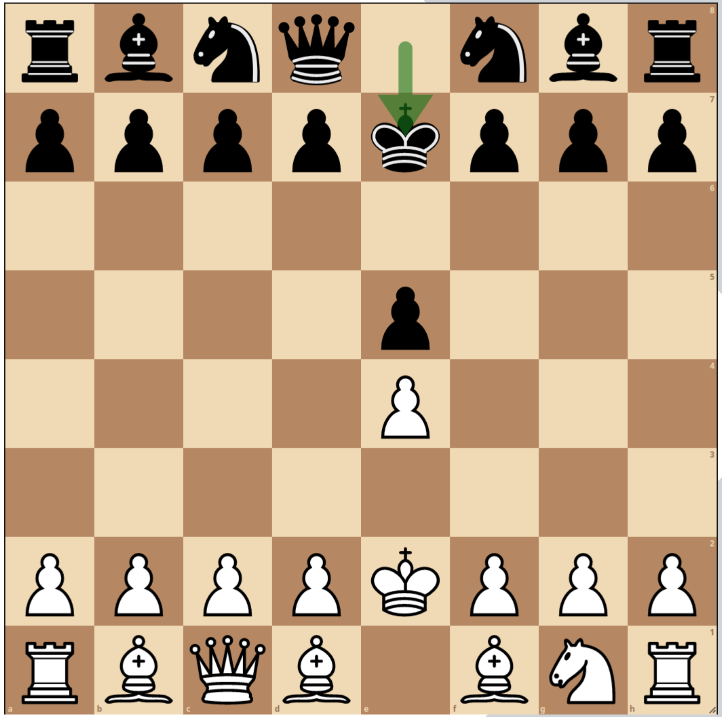
- A problem for RL is that rewards are often very rare and delayed



Lichess 2023, *Double Bongcloud*. Screenshot by author

Motivation

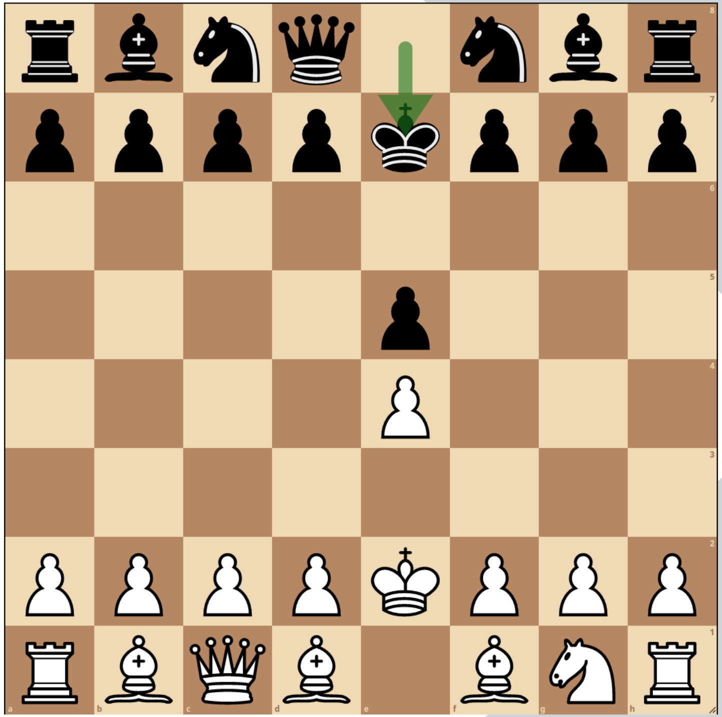
- A problem for RL is that rewards are often very rare and delayed
- Furthermore, many problems have huge state-action spaces



Lichess 2023, *Double Bongcloud*. Screenshot by author
APGL License

Motivation

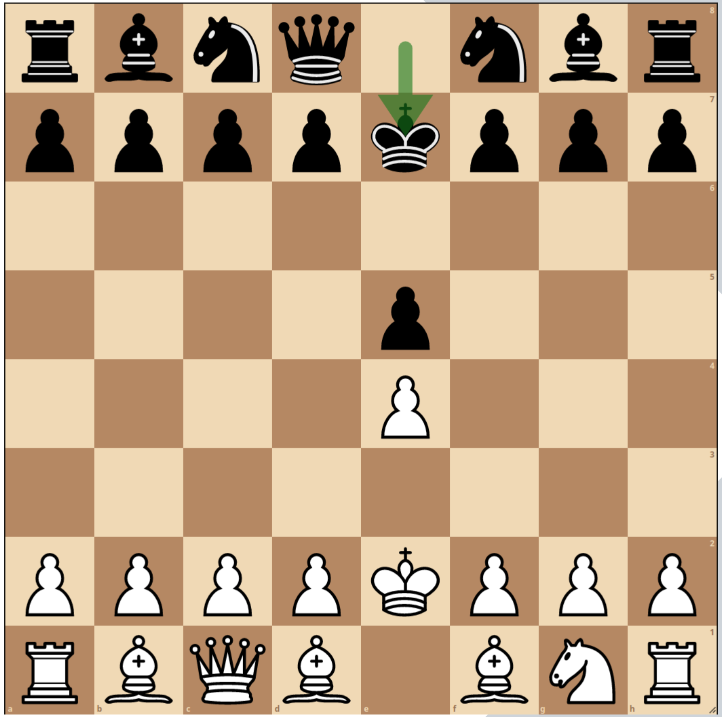
- A problem for RL is that rewards are often very rare and delayed
- Furthermore, many problems have huge state-action spaces
- Intrinsically Motivated RL attempts to solve this by rewarding:
  - Novelty of outcomes
  - Surprise



Lichess 2023, *Double Bongcloud*. Screenshot by author

Motivation

- A problem for RL is that rewards are often very rare and delayed
- Furthermore, many problems have huge state-action spaces
- Intrinsically Motivated RL attempts to solve this by rewarding:
  - Novelty of outcomes
  - Surprise
- “But not everything novel or unpredictable is useful”



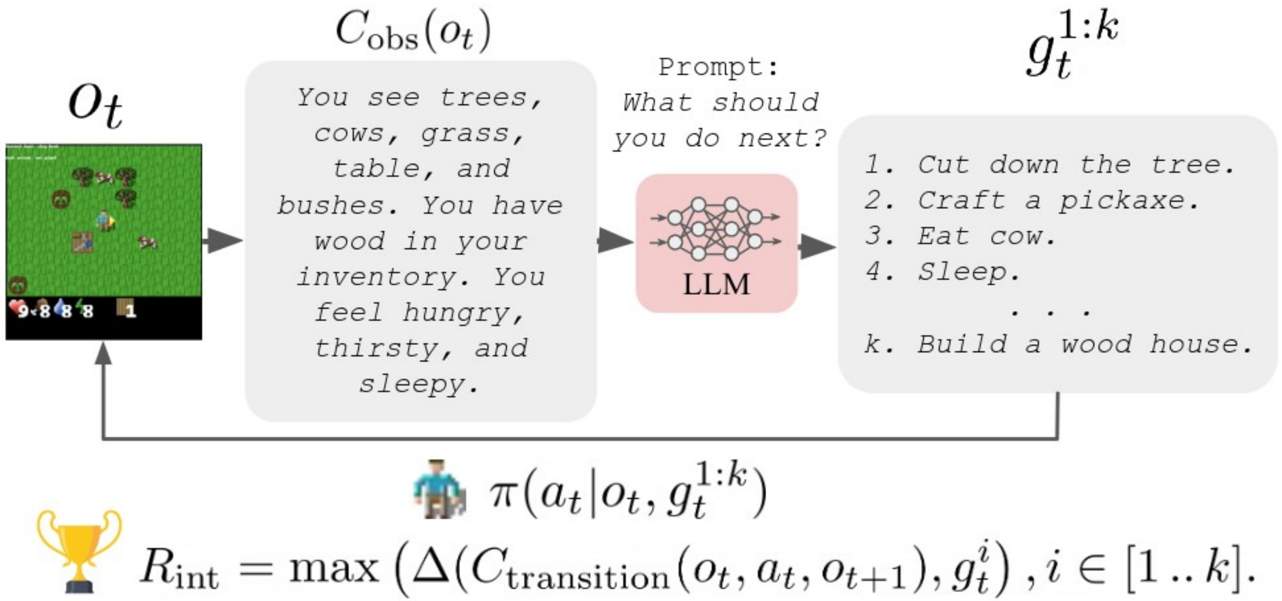
Lichess 2023, *Double Bongcloud*. Screenshot by author
APGL License

Exploration with LLMs (ELLM)

- Key insight: Humans do not explore uniformly
- We use intuition to explore *plausibly useful* behaviour first
- An LLM encodes information about human common-sense knowledge
- This can be used to make the agent explore more intelligently

Mossmouth 2013, *Mantrap*. Screenshot from https://spelunky.fandom.com/wiki/Mantrap_(HD)?file=XBLA_Mantrap.png

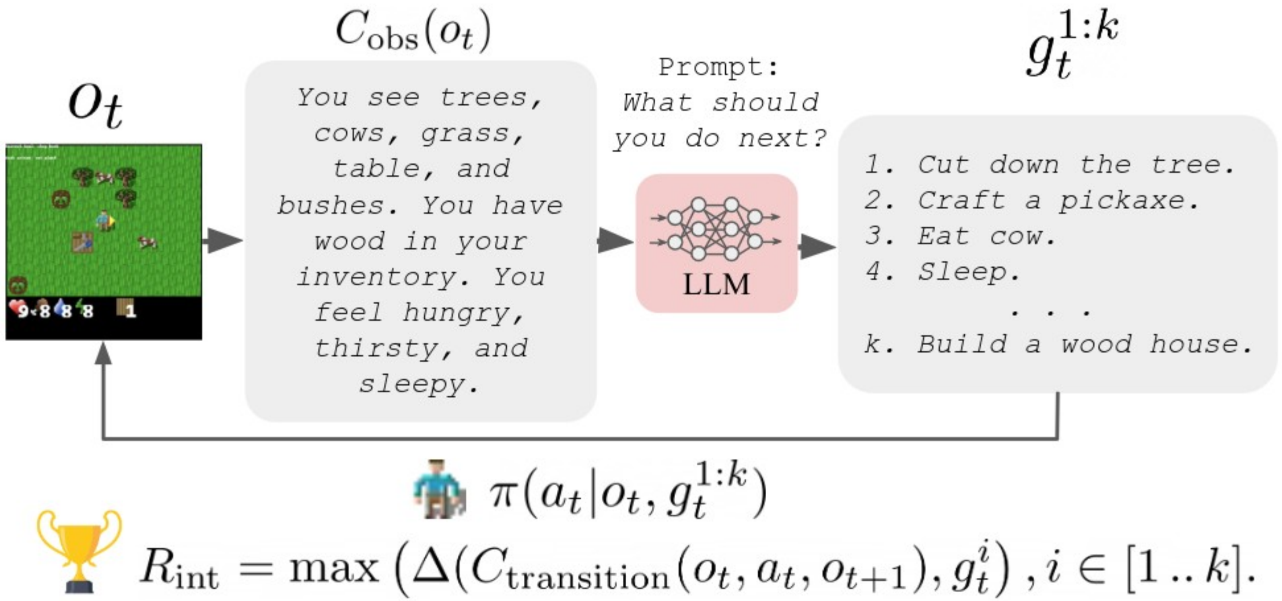
Implementation overview



Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

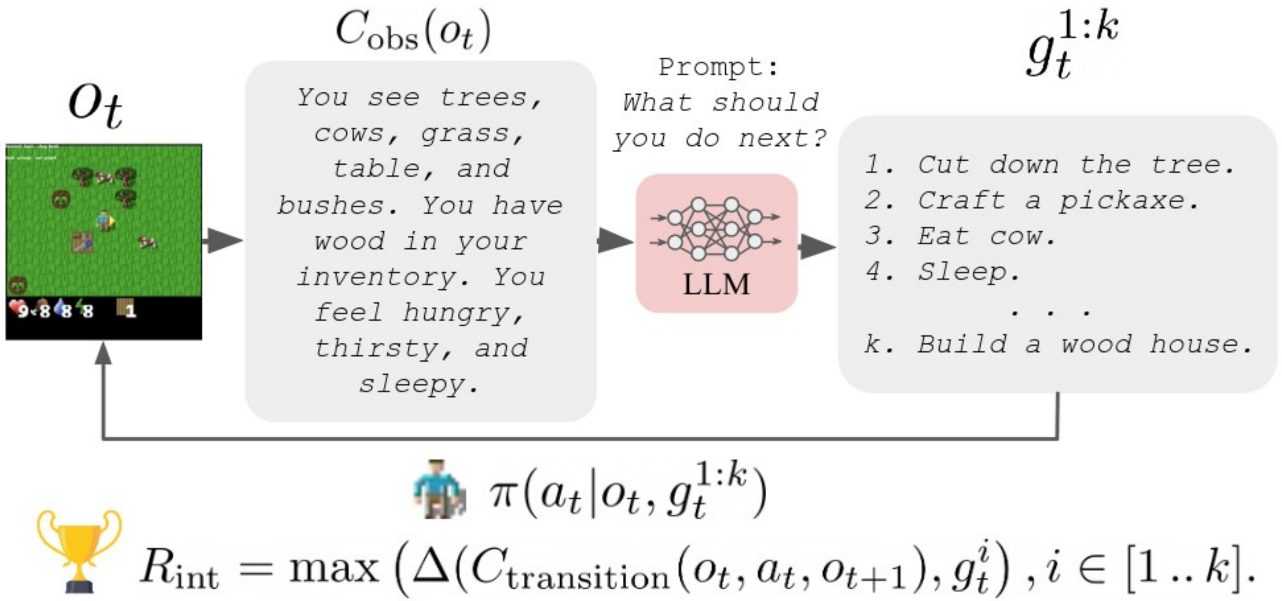
1)Observation captioned to natural language (C_{obs})



Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

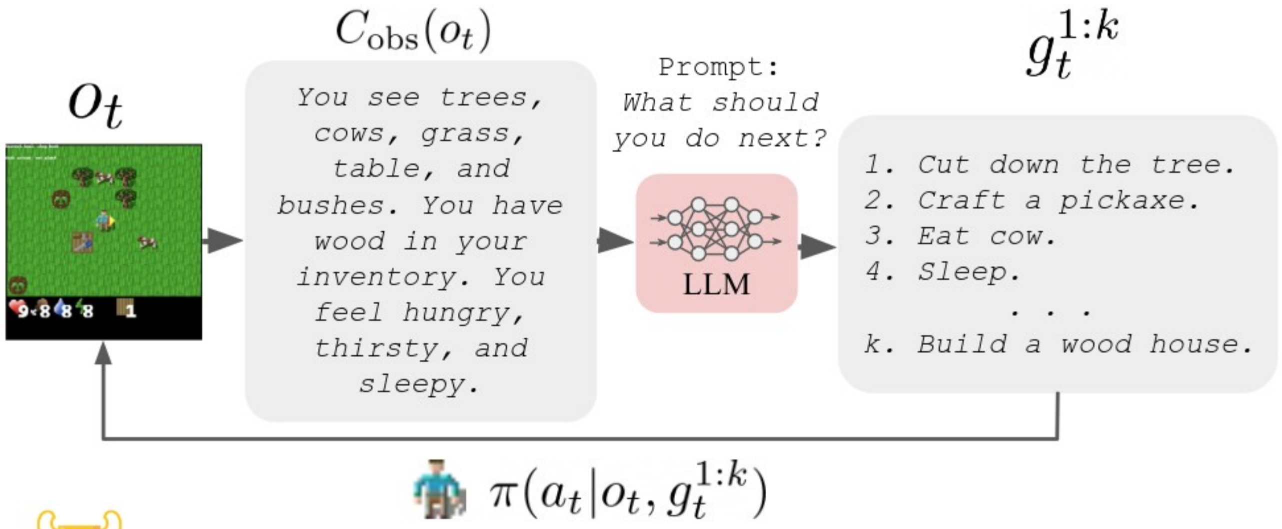
1)Observation captioned to natural language (C_{obs})
2)Text observation joined with LLM prompt



Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

1) Observation captioned to natural language (C_{obs})
2) Text observation joined with LLM prompt
3) LLM gives suggestions

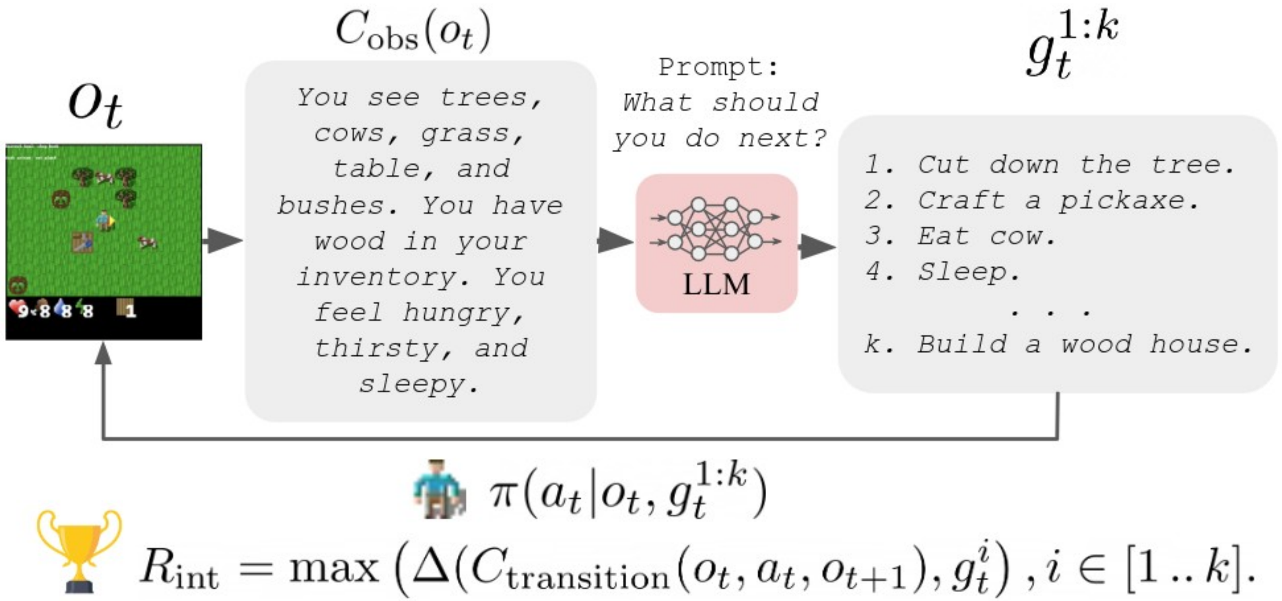


$$R_{\text{int}} = \max\left(\Delta(C_{\text{transition}}(o_t, a_t, o_{t+1}), g_t^i\right), i \in [1 .. k].$$

Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

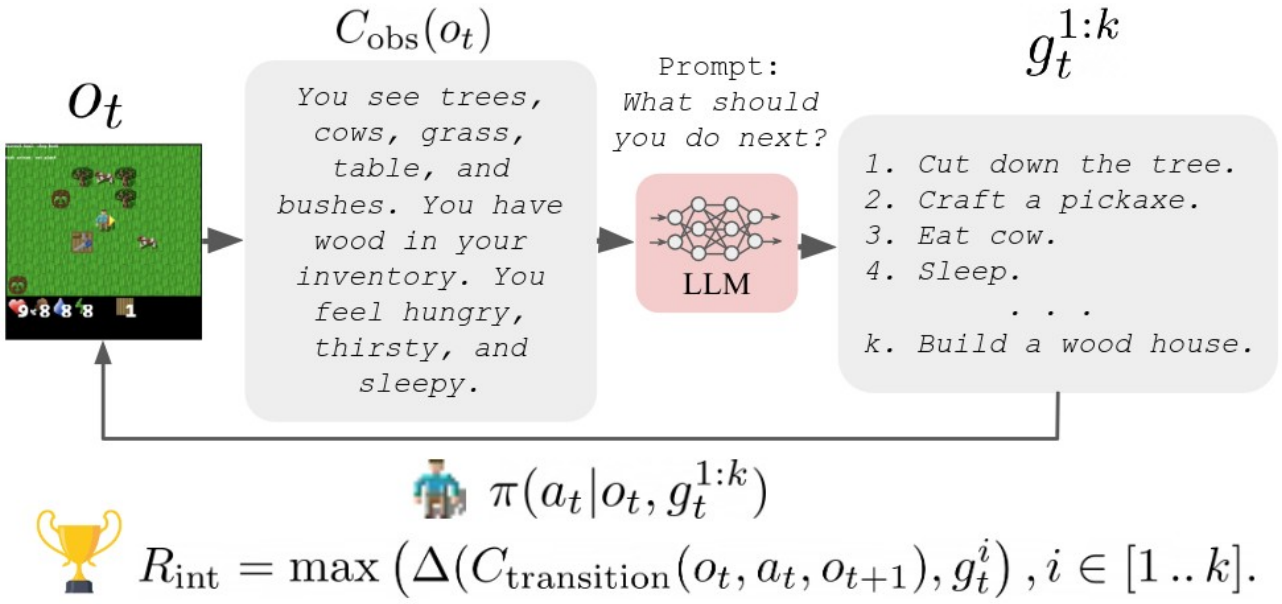
1)Observation captioned to natural language (C_{obs})
2)Text observation joined with LLM prompt
3)LLM gives suggestions
4)Agent does action



Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

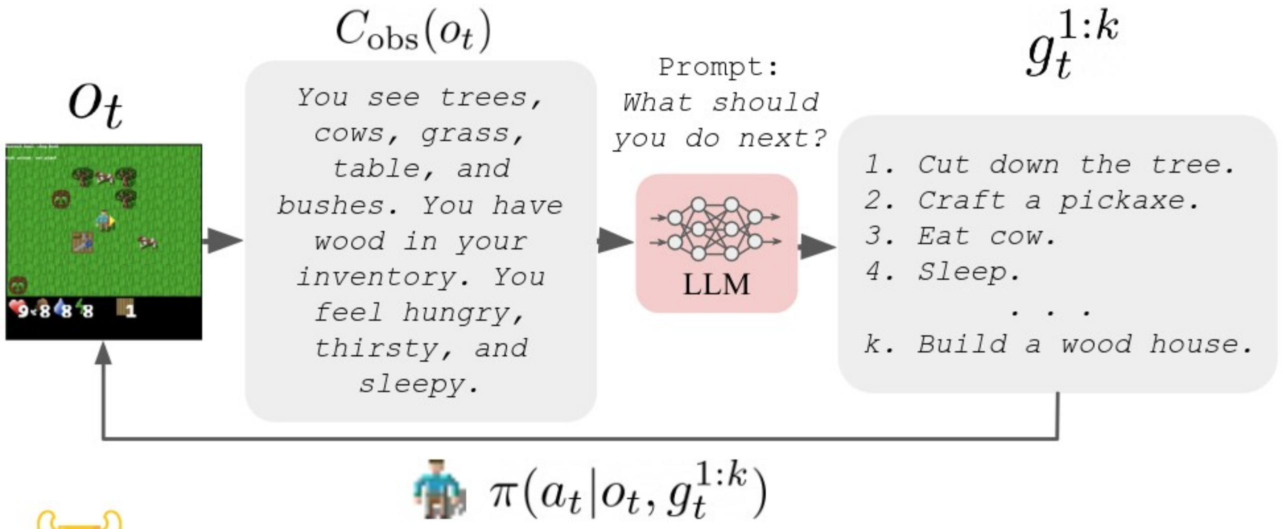
1)Observation captioned to natural language (C_{obs})
2)Text observation joined with LLM prompt
3)LLM gives suggestions
4)Agent does action
5)Action is also captioned ($C_{transition}$)



Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Implementation overview

1)Observation captioned to natural language (C_{obs})
2)Text observation joined with LLM prompt
3)LLM gives suggestions
4)Agent does action
5)Action is also captioned ($C_{transition}$)
6)Agent is rewarded if action caption is semantically similar to a suggested action



$$R_{\text{int}} = \max\left(\Delta(C_{\text{transition}}(o_t, a_t, o_{t+1}), g_t^i\right), i \in [1..k].$$

Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

Results

- Exploration with LLMs beats APT and RND, which are state of the art Intrinsically Motivated RL algorithms
- It also performs better on “downstream tasks”

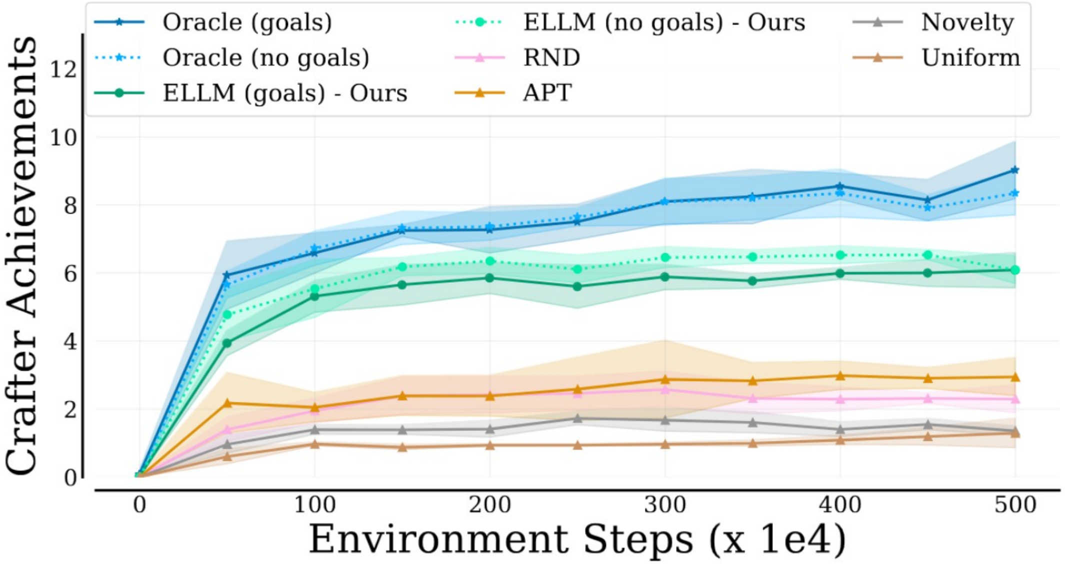
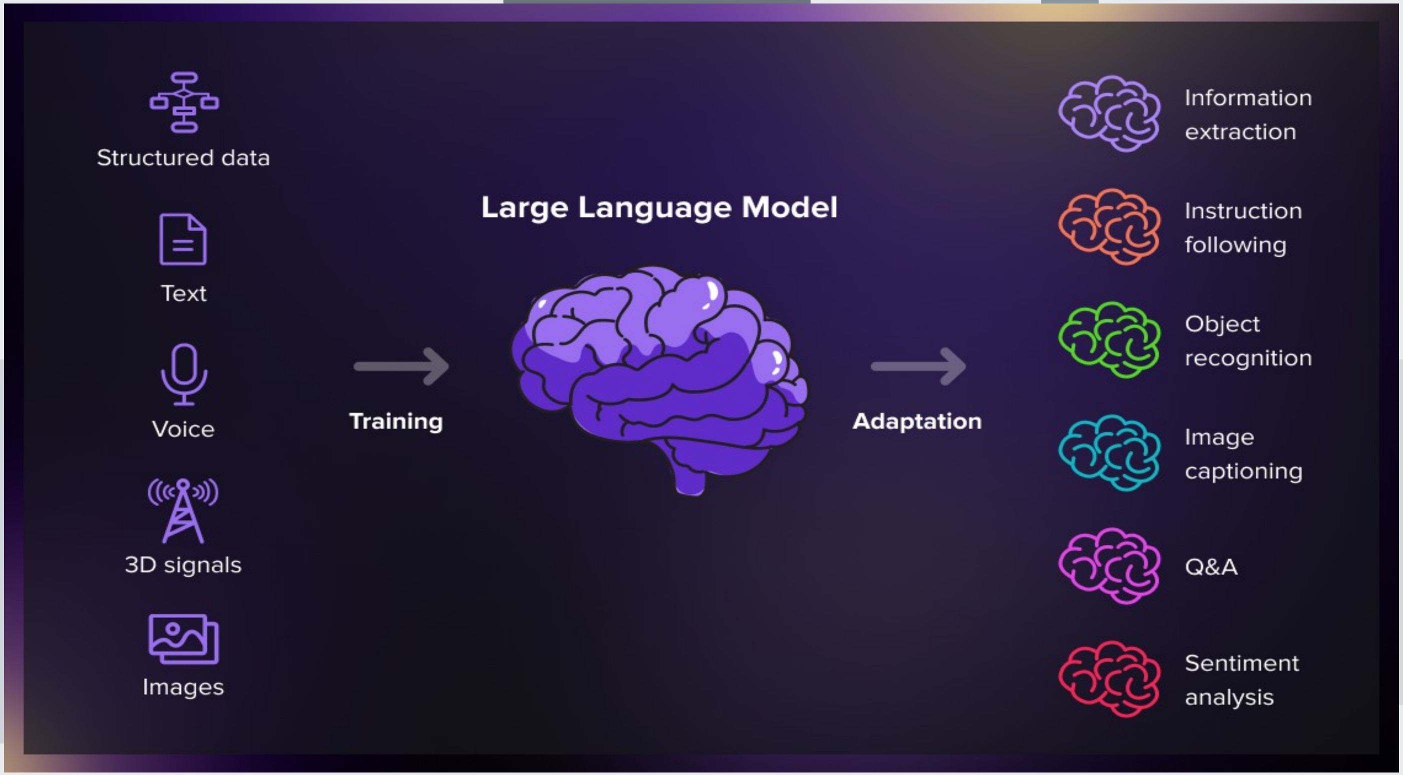


Figure 4: Ground truth achievements unlocked per episode across pretraining, mean±std across 5 seeds.

Du et Al. *https://arxiv.org/pdf/2302.06692.pdf*

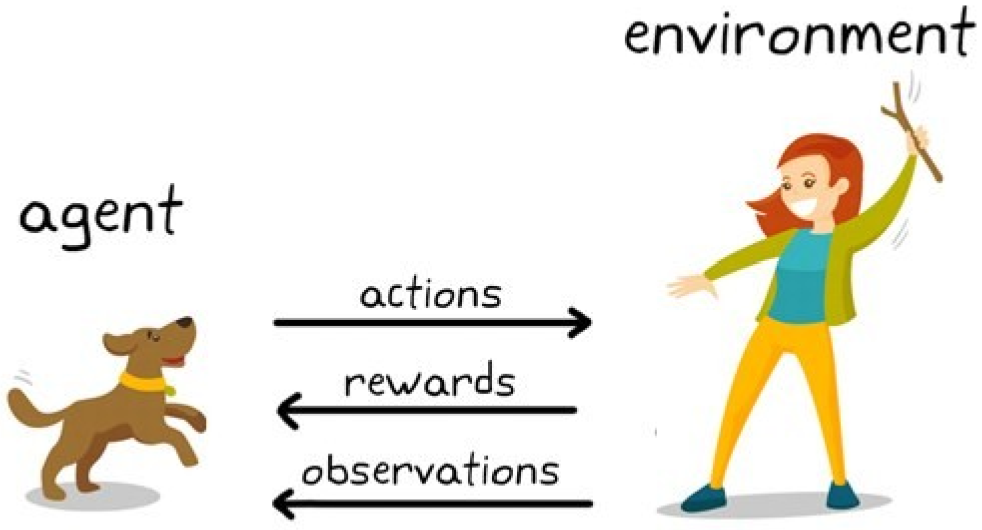


Pre-Trained Language Models for Interactive Decision-Making

Universitetet i Oslo
September 28th, 2023

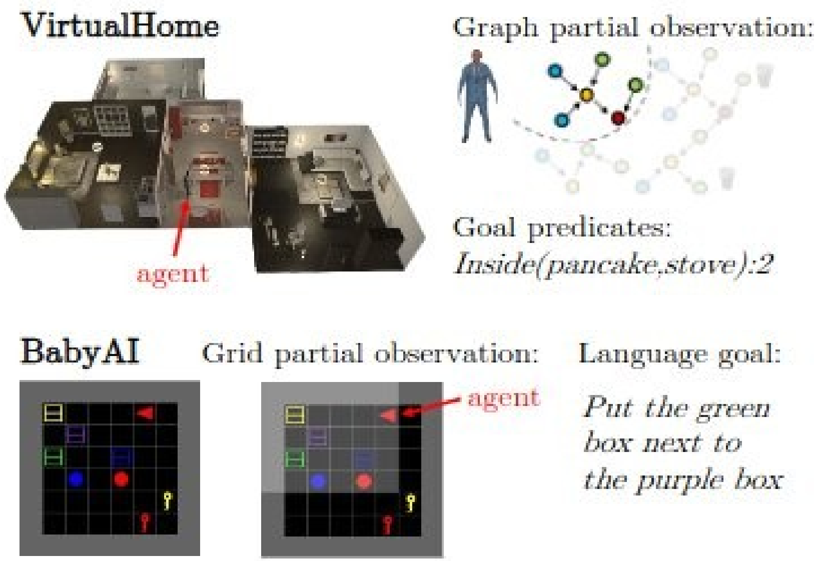
Introduction

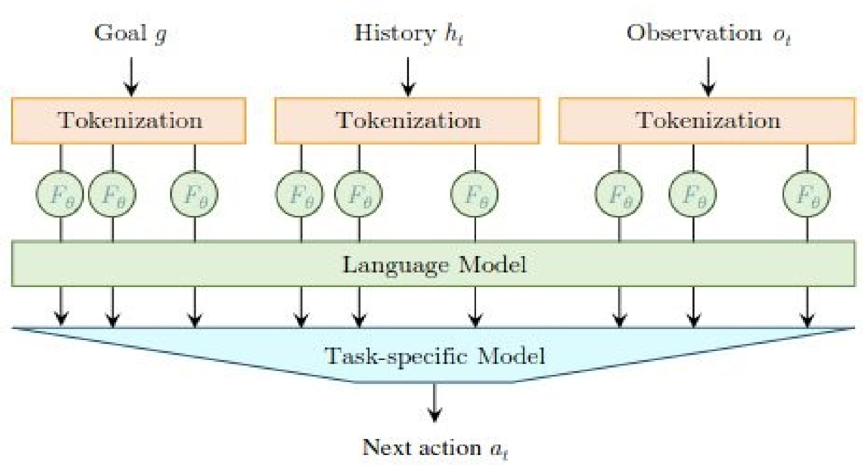
- Reinforcement Learning
- Non-trivial planning and reasoning capabilities
- LM-based policy
- Active Data Gathering
- Why do LM perform so much better?



- LID → Pre-Trained Language Model for Interactive Decision-Making
- LID integrated into the policy network
- Convert goal, history and observations to text, and feed it to LM/LID.
- Receive “contextualized token” represantation, which is averaged and used to predict next action.

Uses a standard LM, GPT-2, to process the input sequence rather than to predict future tokens





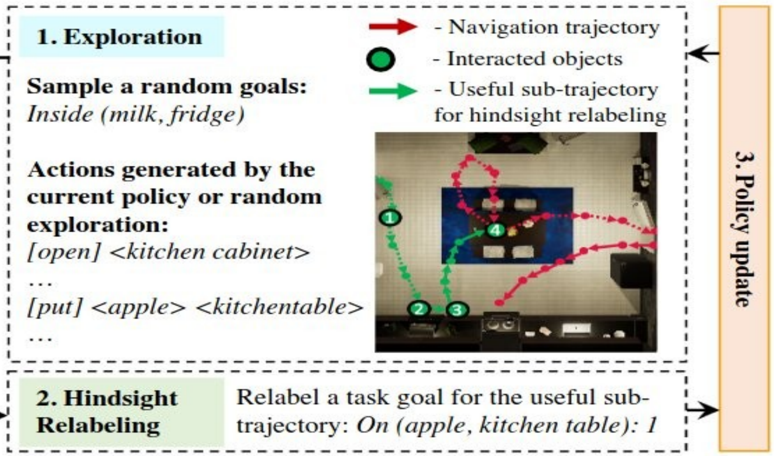


Figure 2: **LID with the active data gathering procedure.** By iteratively repeating the exploration, hindsight relabeling, and policy update, LID with active data gathering can learn an effective policy without using pre-collected expert data.

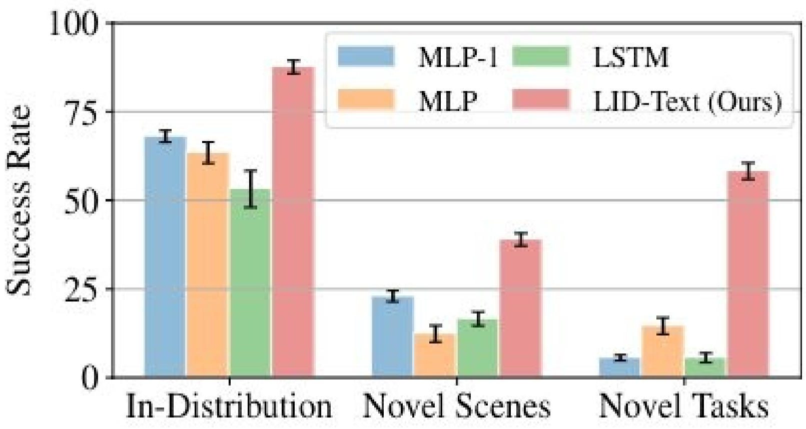
Tasks	Methods	Number of Demos				
		100	500	1K	5K	10K
GoToRedBall	BabyAI-Ori [16]	81.0	96.0	99.0	99.5	99.9
	LID-Text (Ours)	**93.9**	**99.4**	**99.7**	**100.0**	**100.0**
GoToLocal	BabyAI-Ori [16]	55.9	84.3	98.6	**99.9**	99.8
	LID-Text (Ours)	**64.6**	**97.9**	**99.0**	99.5	99.5
PickupLoc	BabyAI-Ori [16]	28.0	58.0	93.3	97.9	**99.8**
	LID-Text (Ours)	**28.7**	**73.4**	**99.0**	**99.6**	**99.8**
PutNextLocal	BabyAI-Ori [16]	**14.3**	16.8	43.4	81.2	97.7
	LID-Text (Ours)	11.1	**93.0**	**93.2**	**98.9**	**99.9**

	In-Distribution	Novel Scenes	Novel Tasks
Random	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
Goal-Object	0.8 ± 0.5	0.0 ± 0.0	0.4 ± 0.4
PPO	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
DQN+HER	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
LID-ADG (Ours)	$\mathbf{46.7 \pm 2.7}$	$\mathbf{32.2 \pm 3.3}$	$\mathbf{25.5 \pm 4.1}$

Table 2: **Comparisons of methods without using expert data on VirtualHome.** LID-*ADG (Ours)* is the only successful approach.

	In-Distribution	Novel Scenes	Novel Tasks
LID-ADG (Ours)	46.7 ± 2.7	$\mathbf{32.2 \pm 3.3}$	25.5 ± 4.1
PPO (LID-ADG Init)	$\mathbf{53.7 \pm 3.5}$	30.2 ± 3.4	$\mathbf{27.8 \pm 2.7}$
DT (LID-ADG Data)	42.4 ± 1.5	21.6 ± 2.48	16.8 ± 1.0

Table 3: The proposed method with active data gathering, LID-ADG (Ours), can be used as an policy initializer for online RL or a data provider for offline RL.



Favorable Weight Initialization (I.E. Pre-Trained LM)

> “Why do Language models perform so well at generaliztion to embodied decision-making problems?”

~~Input Encoding Scheme~~

Sequential Input Representation

OMNI: Open-endedness via Models of human Notions of Interestingness

JENNY ZHANG, JOEL LEHMAN, KENNETH STANLEY, JEFF CLUNE

Training Reinforcement Learning (RL) Agents in Large Environments

- Large environment → Large search space
- → Infinitely many possible tasks
  - Even when we only count tasks that the agent is able to learn

How do we choose which tasks to learn first?

- Large Language Models (LLMs) contain human knowledge
  - Humans know which tasks are interesting
- → An LLM could tell an RL agent which tasks to learn first

FIGURE 1 Minecraft – an example of an extremely large environment, with an infinitely large action space. Mojang 2011, *Minecraft*. Screenshot from https://minecraft.fandom.com/wiki/Gameplay

Method

PROMPT

```
You are a player in a game. You want to learn as many skills as possible.
You can do these tasks well: <tasks done well>.
Suggest whether the given tasks are interesting: <tasks to be determined>.
```

ALGORITHM

Algorithm 1 Mechanism to partition the task set into interesting and boring sets.

1: Sort the tasks based on the evaluated task success rates.
2: Create two empty sets, one to track the interesting tasks and one to track the boring tasks.
3: Identify the task with highest success rate and not in any of the sets. Add it to the interesting set.
4: Prompt the LM to determine if any of the remaining tasks are boring, contexted on the current set of interesting tasks. Tasks in the interesting set are input as `<tasks done well>` and tasks yet to be categorized are input as `<tasks to be determined>` in the LM prompt (above).
5: Update the boring set with tasks that the LM has determined as boring.
6: Repeat steps 3 - 5 until all tasks are in either set.

FIGURE 2 Algorithm as presented in [1].

Usage in Practice

- Algorithm tested in Crafter
- RL agent trained using Proximal Policy Optimization (PPO)
    - State-of-the-art «standard» RL method
- OMNI's role: Suggest tasks for agent to perform
    - Interesting tasks will be chosen more often
    - Influences policy of RL agent (choosing an action)
- «Boring» tasks were added to show LLM's decision-making ability

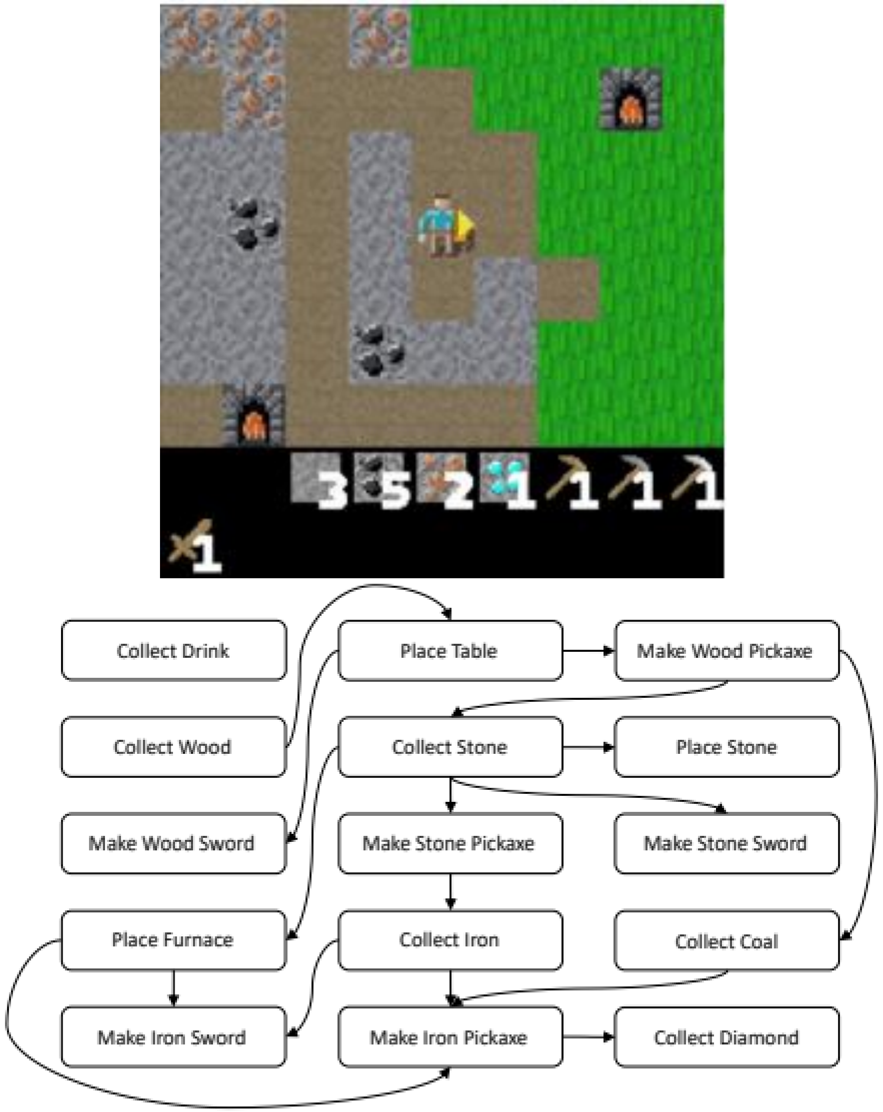


FIGURE 3 Above: Danijar Hafner 2021, *Crafter*. Screenshot from [1]. Below: Example of actions considered interesting, and the order in which they should be completed.

Relevance to Our Project

- We also want to choose relevant actions
- Generalized algorithm
  - It may be used even in different environments
- Other ways of using LLMs also possible
  - For reward shaping, instead of policy
- Interpretation of «interestingness»
  - Interesting = action with highest success rate?
  - Interesting = action most similar to other interesting actions?
    - OMNI algorithm assumes the two above
  - Interesting = (performed) action most similar to goal?

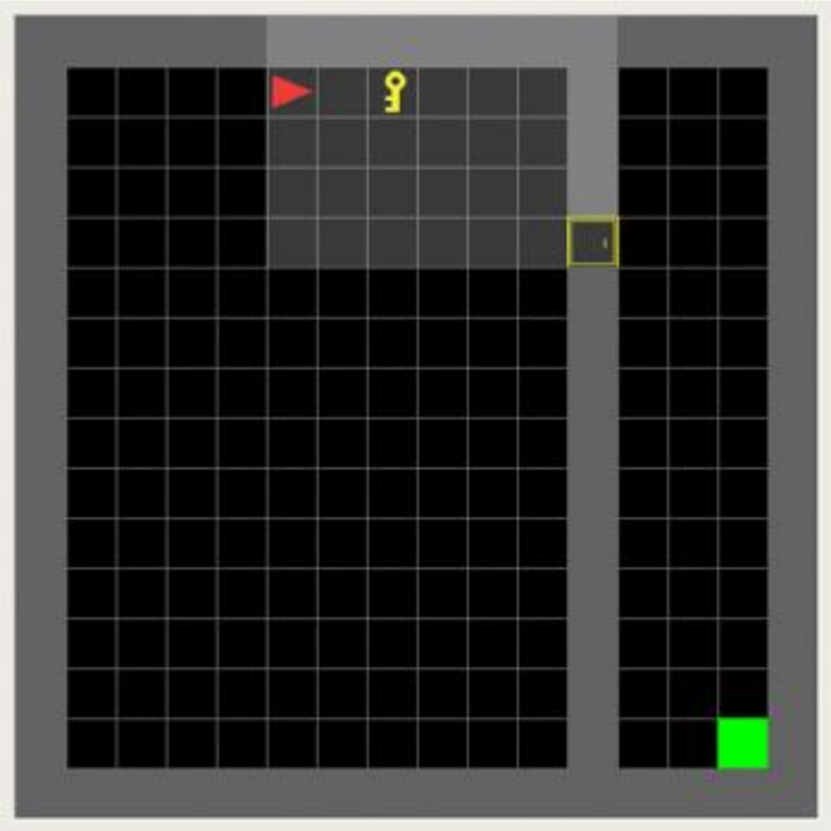


FIGURE 4 Minigrid, the testing environment we use in our project [2]. Screenshot from https://minigrid.farama.org/

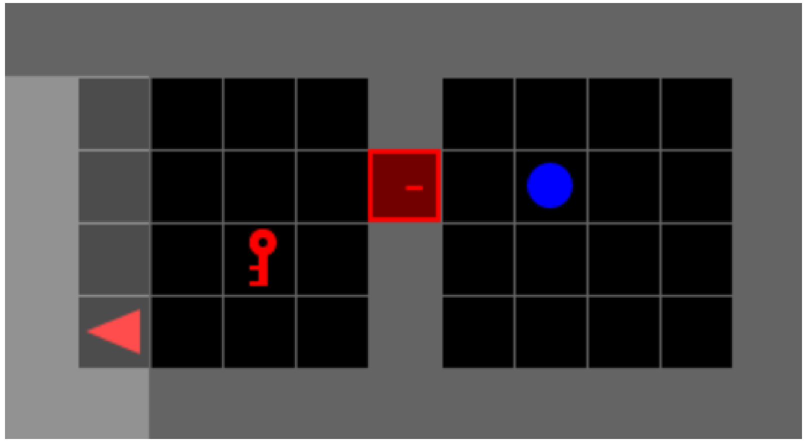
References

[1] J. Zhang et al., «OMNI: Open-endedness via Models of human Notions of Interestingness». https://arxiv.org/abs/2306.01711

[2] M. Chevalier-Boisvert et al., «Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks». https://arxiv.org/abs/2306.13831

Our Approach

- LLM as policy
  - LLM gets state prompt
  - Answer becomes RL agent policy
- LLM as reward
  - LLM gets state prompt
  - Returns recommended action
  - Similar actions to recommended one are rewarded



You are a player playing a videogame. It is a top down turn based game, where each turn you can move in one of the four cardinal directions. You can see a red key 4 squares north and 2 squares east, and a red door 3 squares south of your location. What move should you do? Please only answer a single cardinal direction, without elaborating on you choice. For example: given a description such as this, you could respond with the singular word "East".

North

Above: Farama 2023, *Minigrid*. Screenshot by author. Below: Example prompt to LLM, and LLM response.

Current State (!) of the Project

Achieved so far

- LLM can control agent directly in Minigrid environment
- Soon implemented conventional RL baseline (PPO)
- Can reward similarity between observation and LLM recommendation

To be improved

- LLM (Llama 2) is... not smart
- Agent still not actually trained by LLM actions
- Final architecture not decided upon yet

Where We're Headed

Establish conventional RL baseline

- Finalize Proximal Policy Optimization (PPO)
- Measure results

Integrate LLM into Architecture

- Decide: LLM as policy or reward?
- Automate communication between LLM and RL agent
- Fit into RL framework

Testing and evaluation

- Our results vs. PPO only?
- Sampling efficiency improved?