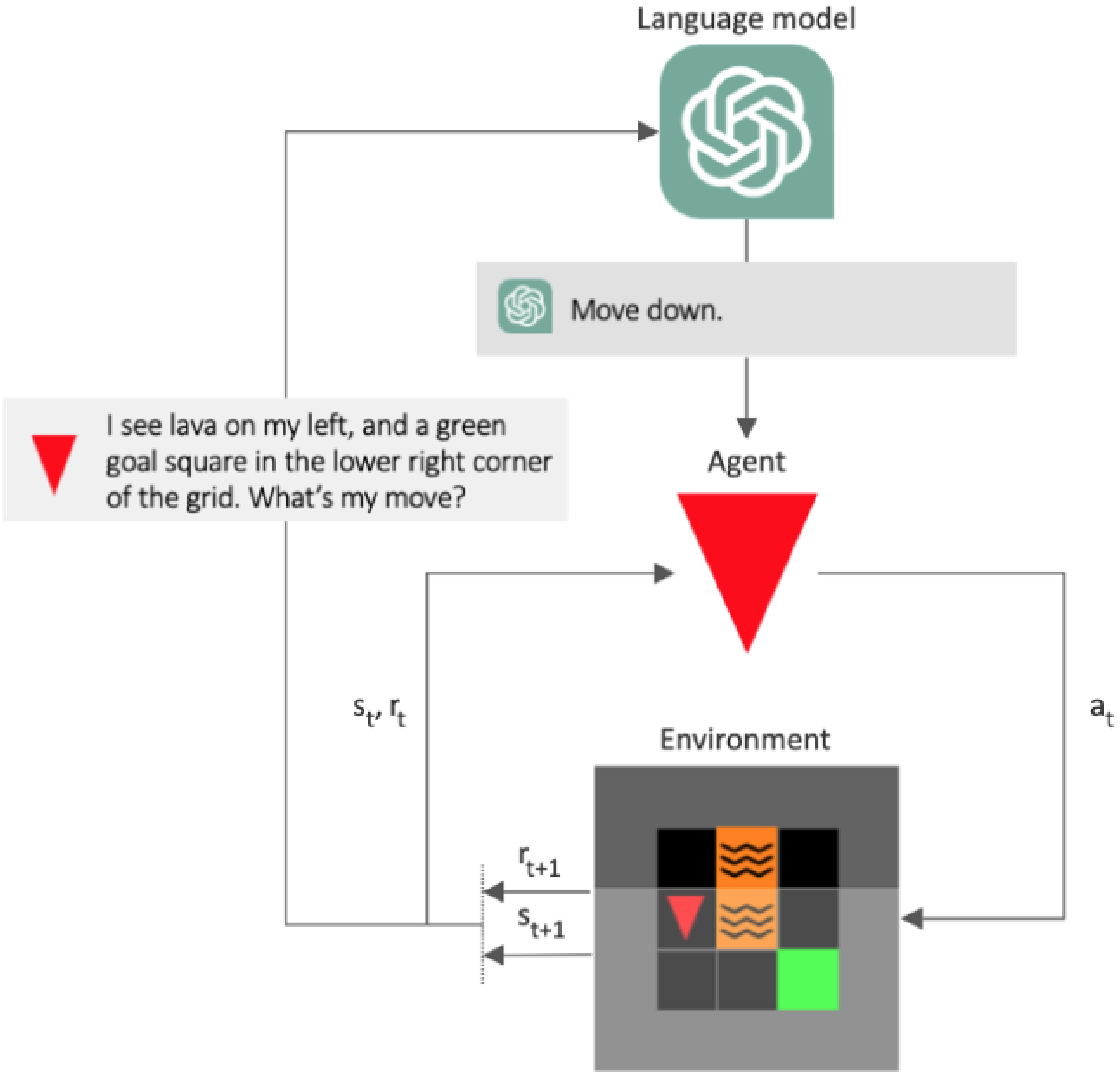
Integrating Large Language Models into Reinforcement Learning

Gregor Kajda, Jonatan Hoffmann Hanssen, Adrian Duric

Supervisors: Katrine Nergård, Kai Olav Ellefsen

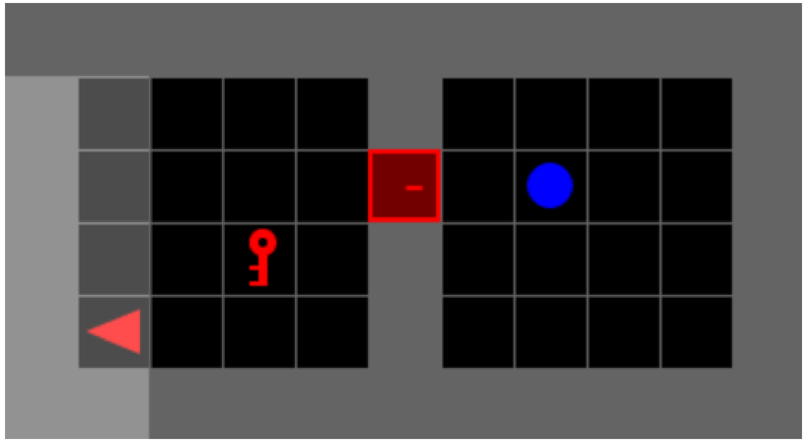
Aim of the Project

- 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



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?