Manish Kumar

Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents

Introduction

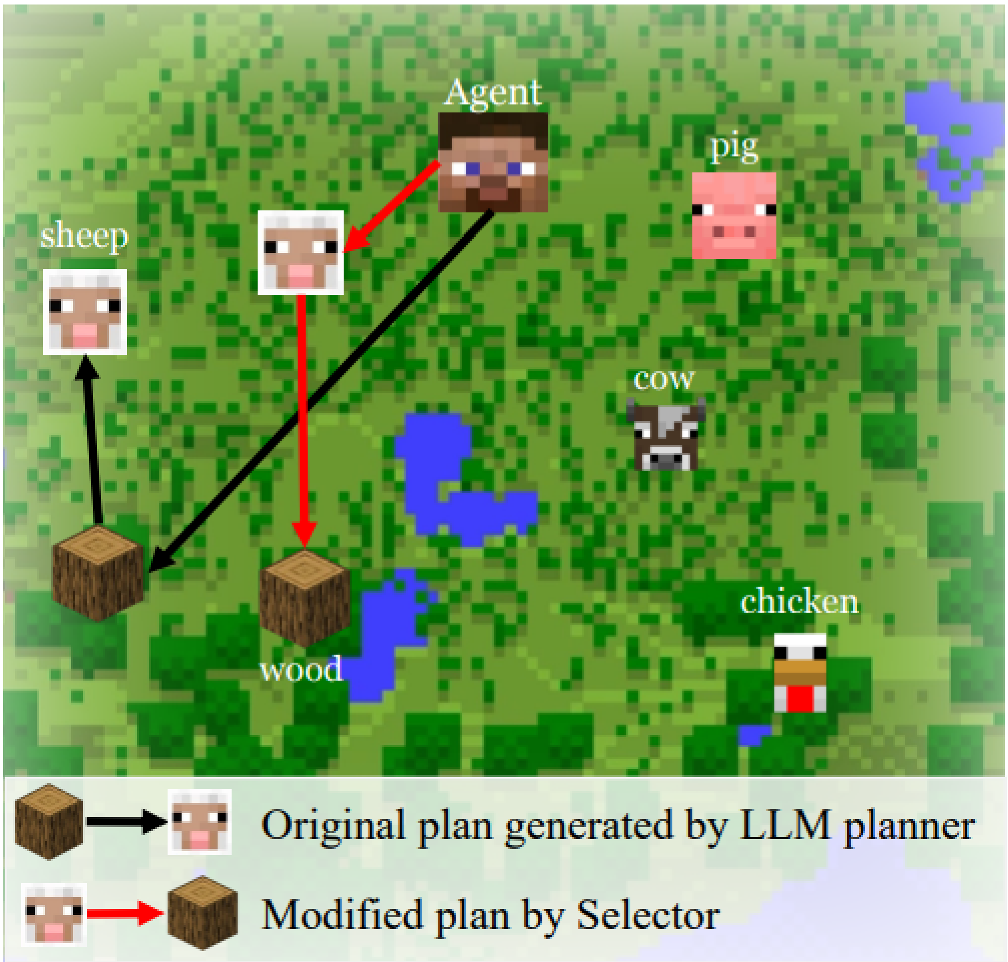
This paper aim to learn an agent that can solve "arbitrary long horizon" (long sequence of goals such as in retrosynthesis) and goal-reaching tasks with image observation and language goals. Specifically, it focuses on Minecraft which is an open-world video game where users can mine the ground and their surroundings for resources to build absolutely anything they want. People have built all sorts of things, from a simple sleeping bed to entire cities from scratch. As per the paper, developing multi-task agents that can accomplish a vast and diverse suite of tasks in an open-ended world has been viewed as one of the key milestones towards generally capable artificial intelligence.

Below is a 2 minute video that shows an AI agent trying to find a resource "diamond" by mining the earth.

https://www.youtube.com/watch?v=GHo8B4JMC38

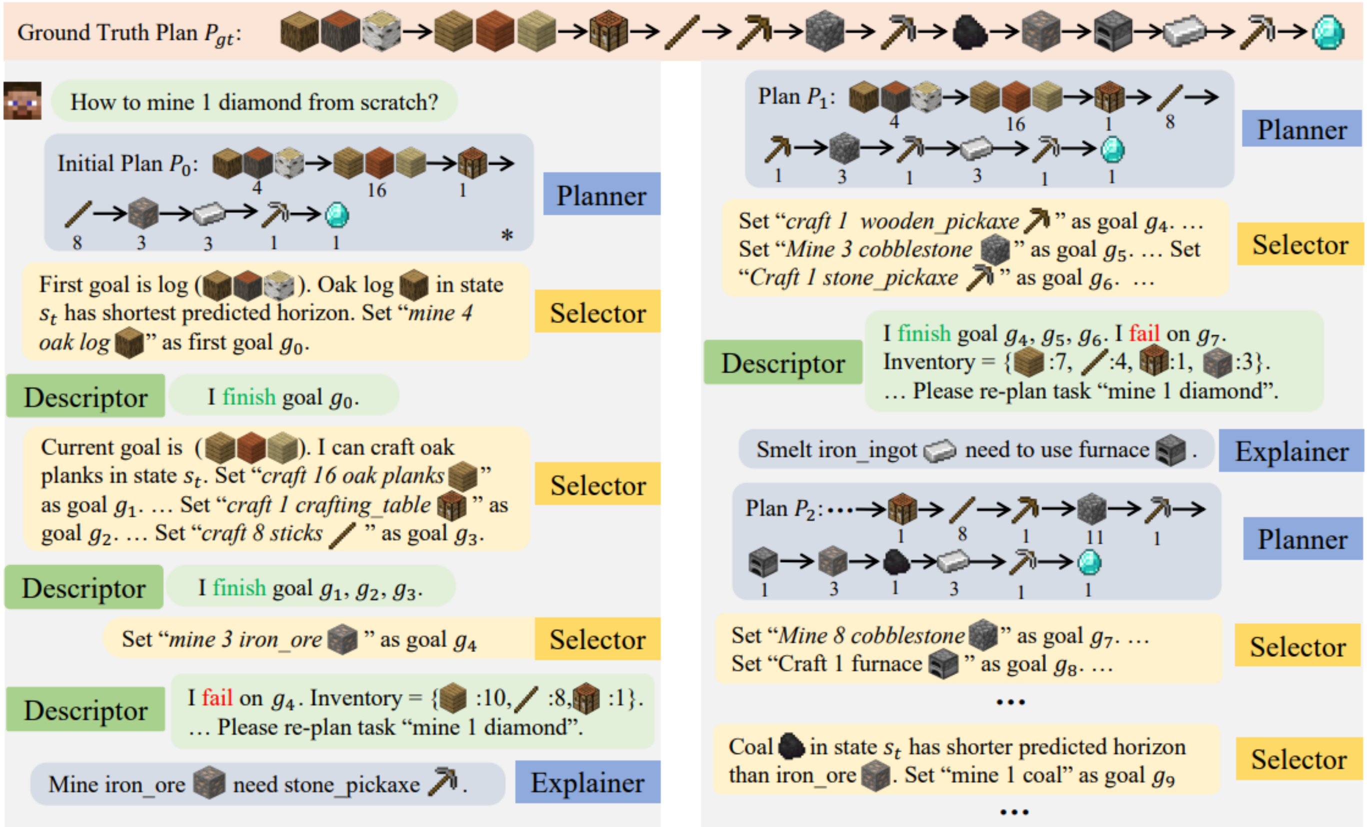
This paper identifies two primary challenges of planning in these environments:

1. **Long-term planning:** First, many tasks in Minecraft can be complex, as they usually comprise multiple sub-goals to be completed, e.g. the task `build a bed` includes 7 sub-goals (`mine wood blocks` for wood, `kill sheeps` for bedding, etc.) and therefore demand significantly longer reasoning steps of the `planner`. Meanwhile, many of these sub-goals have to be planned precisely with the exact object name and quantities, e.g. `mine 3 woods` , `kill 3 sheep`; otherwise, the subsequent sub-goals won't be executed due to failed preconditions.
2. **Planning efficiency:** Second is the efficiency of the produced plans, which is illustrated in the below figure. The `planners` do not consider the current proximity of the sub-goals to the agent when devising the plans, thereby producing inefficient plans. In the below figure, the initial plan (in black) was to `mine wood` first, then `kill sheep`. However, other way around is more efficient (in red). The `selector` part modifies the plan from the `planner`.



To this end, the authors propose “Describe, Explain, Plan and Select” (DEPS), an interactive planning approach based on Large Language Models (LLMs) to alleviate the aforementioned issues in most open-world environments. Whenever a failure happens when executing the current plan, a `descriptor` will summarize the current situation as text and send it back to the LLM-based `planner`. The LLM-based planner will then be prompted as `explainer` to locate the errors in the previous plan. Finally, the planner will re-plan the task to obtain a correct plan. This allows the feedback from the agents to be better handled by the planner and increases the overall success rate on Minecraft tasks by 52.74%. Additionally, the `selector` will modify the plan depending on which sub-tasks are most accessible based on the proximity to the agent.

Below figure shows an example of the whole process



A ground truth plan is given to mine a diamond. You ask the LLM model "How do I mine 1 diamond from scratch?" The planner comes with a set of goals, then selector decides which goals to execute first. If it's finished, tell the LLM that it's finished and it'll come up with next goals. If a goal can't be achieved (task failed), write a text explanation to the LLM and it'll come back with an updated plan.

Relevance to retrosynthesis

Retrosynthesis have something of a similar process. There's a goal of identifying an optimal synthesis path (like mining a diamond) and to achieve it, multiple sub-goals have to be completed (liking mining for wood). Sometimes, a path might not be feasible (goal failed) so we can ask the LLM (explainer) to come up with a new plan. Basically, an interactive retrosynthesis planning.