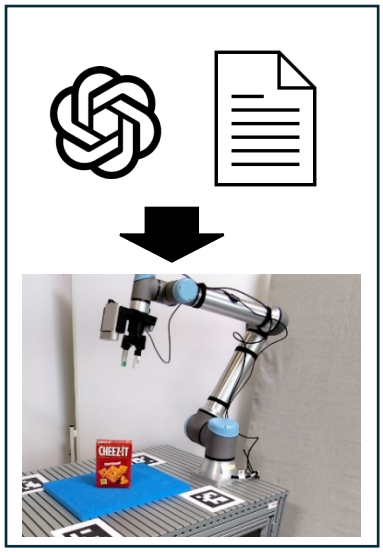
Bachelorarbeit

Foundation Models for Code-Based Robot Manipulation

Problem formulation

Programming robots to perform manipulation tasks typically requires expert knowledge, predefined task logic, or manually designed robot programs. Recent large language models and vision-language models show strong capabilities in understanding natural-language instructions, reasoning about task structure, and generating executable code. This raises the question of whether such models can be used as an intuitive interface for programming robot manipulation tasks. This thesis investigates how reliably foundation models can generate executable robot programs from task descriptions and, optionally, visual scene information. The focus is not on low-level control generation, but on using a predefined set of robot skills or API functions to compose manipulation programs for tasks such as pick-and-place, object reorientation, or simple stacking. The goal is to evaluate the capabilities, limitations, and robustness of current language and vision-language models for code-based robot task execution.

Task definition



The student will develop a framework in which a large language model or vision-language model receives a manipulation task description and generates robot code using a predefined robot API. The available functions may include operations such as moving to a pose, grasping or releasing an object, querying object poses, or executing relative motions. The generated programs will be executed first in a safe simulation or sandboxed environment and, where feasible, on a real robot platform. Several manipulation tasks of increasing complexity will be considered, for example pick-and-place, object reorientation, and simple multi-step manipulation. Different model configurations and prompting strategies may be compared, such as language-only versus vision-language input, zero-shot prompting versus prompts with examples or API documentation, and optional program repair after execution errors. The system will be evaluated with respect to task success, code validity, API misuse, robustness to scene changes, and the number of required correction or regeneration steps.

Requirements

- Solid programming skills in Python.
- Basic knowledge of robotics and robot manipulation.
- Experience with ROS2, simulation environments, or robot programming is beneficial.

Betreuer: M.Sc. Edgar Welte
Betr. Hochschullehrerin: Prof. Dr.-Ing. Rania Rayyes
Bearbeitungszeitraum: SS 2027
Bearbeiter: offen