Low-Cost Robotic Arms and Large Language Models for Autonomous Scientific Laboratories

Pranav Sinha[1], Sumit Jha[2], Arvind Ramanathan[3] and Sunny Raj[1]

[1]Oakland University, U.S {pranavsinha,raj}@oakland.edu, [2]Florida International University, U.S [sumit.jha@fiu.edu],
[3]Argonne National Laboratory [ramanathana@anl.gov]

1. Abstract

The autonomous scientific laboratories paradigm envisions robots conducting trial-and-error experiments to eliminate uncertainties caused by human error, improving reproducibility. Argonne's Workflow Execution Interface (WEI) library helps integrate devices for complex workflows. However, current experiments use expensive robots, making them less practical for education. To address this, we propose using the affordable Arduino Braccio robot and a webcam for an object-moving experiment. A fine-tuned Llama 3 model converts English instructions into Python code for robot control, with a 75% failure rate to simulate real-world conditions. This setup offers accessible training for students and professionals.

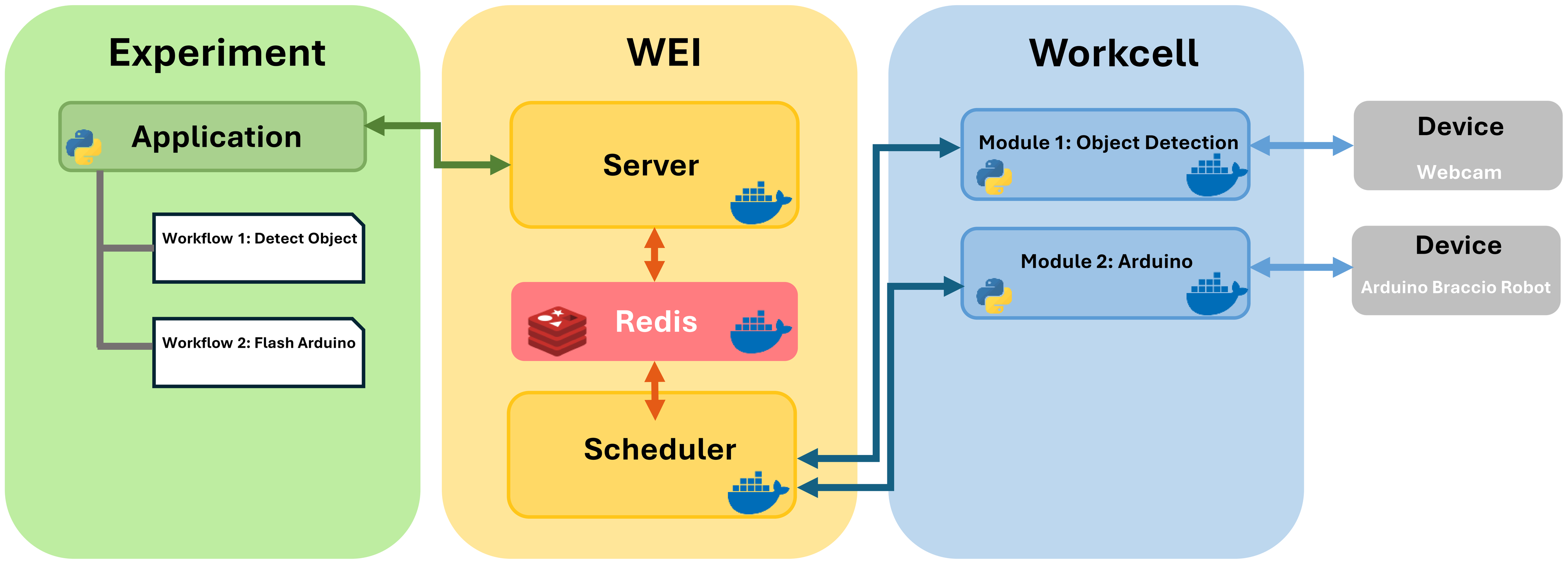
2. About WEI

The **Workflow Execution Interface (WEI)** is a set of tools and software which enables autonomous scientific discovery. It has the following components:

1. **Workcells**: A group of scientific instruments, robots, devices, and software that work together to execute scientific workflows, defined in WEI.
2. **Workflows**: A sequence of steps executed on a workcell, with each step specifying actions for a module, defined in WEI via a YAML file.
3. **Modules**: Software packages controlling scientific instruments, robots, or devices, adhering to WEI's Module Interface, and including the physical hardware, drivers, and necessary integrations.
4. **Experiments**: A collection of workflow runs and logic using WEI to control workcells.

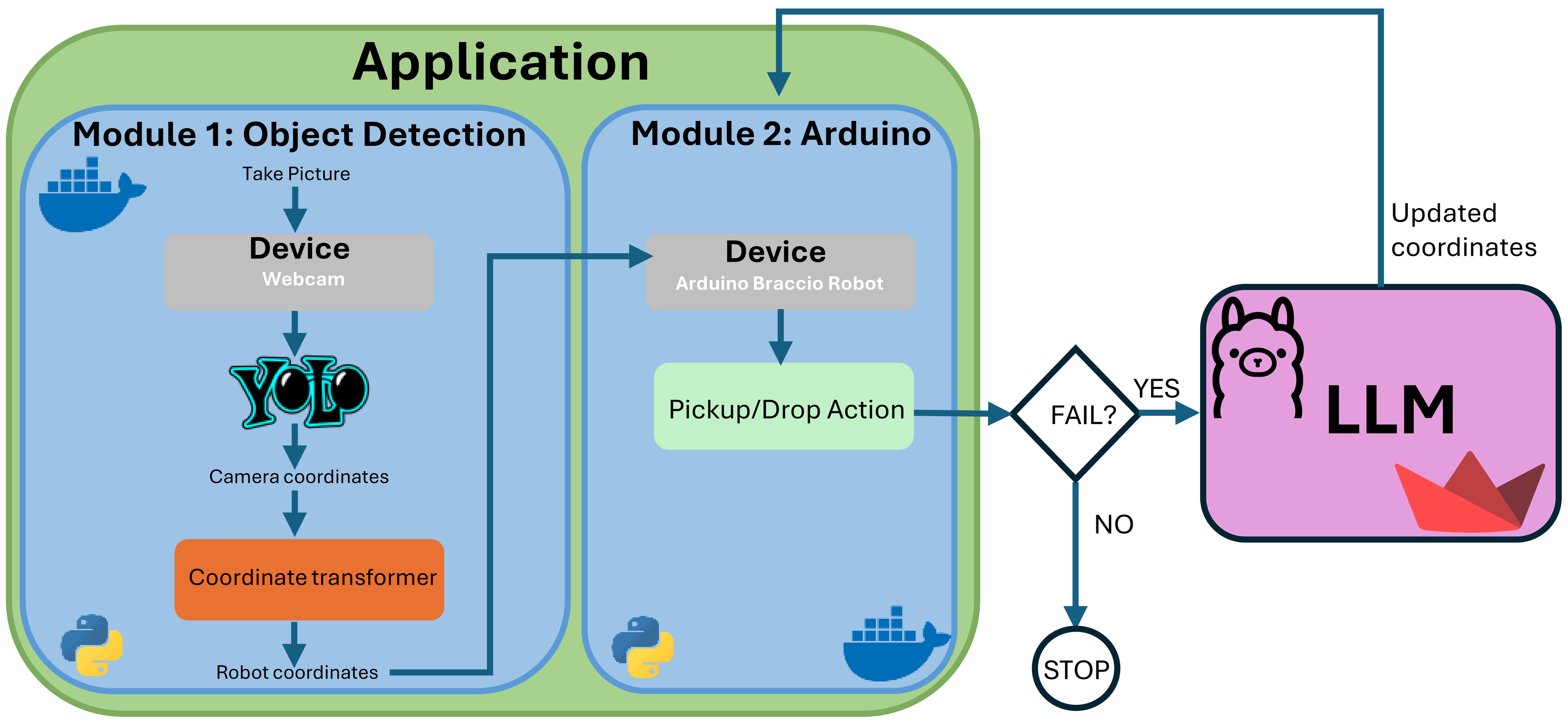
3. WEI Framework

This setup includes a workflow scheduler, a Python library for new devices, and a Python client to integrate workflows into experiments. Experiment applications enable users to run workflows on the WEI Workcell while integrating custom code and logic.



4. Method

The experiment involves a webcam, which takes the picture of the pickup and drop objects. The picture is sent through an image detection model, **YOLOv4-tiny** in this case. While effective in detecting objects, with the help of bounding boxes, the process uses a linear coordinate transformation function that does not account for camera depth, resulting in imprecise pickup and drop coordinates. Due to the inaccuracies in the initial robot coordinates generated by the image detection model, a trial-and-error approach is needed. By intentionally using this imperfect transformation, the system mimics real-world scenarios where conditions are not always ideal. The Large Language Model (LLM) is then tasked with adjusting these coordinates based on user input in natural language, enhancing accuracy through iterative adjustments. This process reflects a trial-and-error method, where each cycle of error correction helps improve the robot's movements, thereby simulating real-world experimentation dynamics.



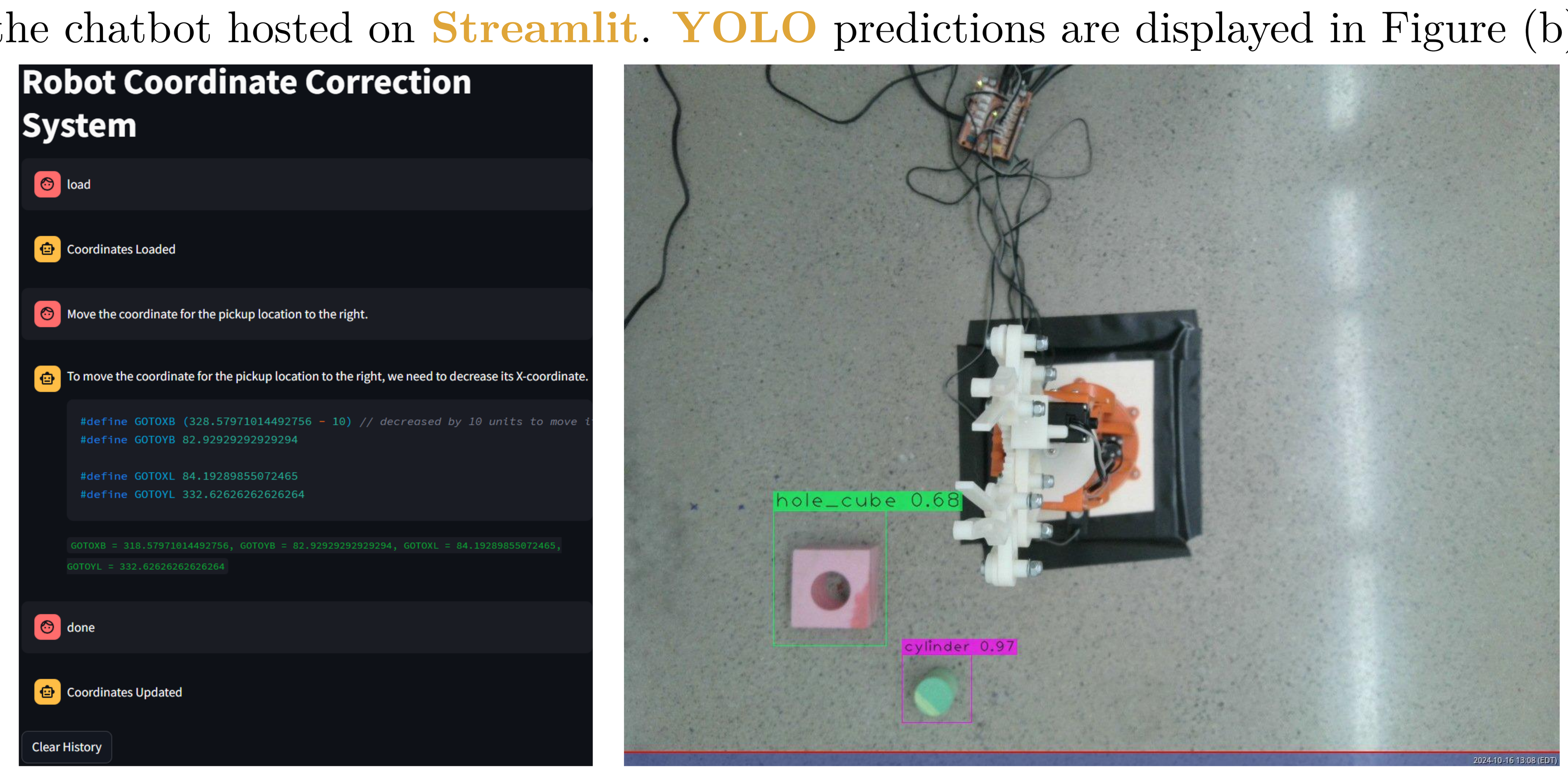
5. Prompt Design

This prompt is carefully crafted to position the LLM as a mathematician specializing in coordinate plane adjustments. This helps refine user-provided pickup and drop coordinates for a Braccio Robot, improving accuracy for smoother robot movement and object handling.

Example Prompt	
System	You are an expert mathematician with specialization in coordinate plane. The user will provide four robot coordinates for a Braccio Robot. Two for pickup location (GOTOXB, GOTOYB) and two for drop location (GOTOXL, GOTOYL). The coordinates are inaccurate and need slight modification. Based on the user input, adjust those coordinates.
User	Here are the coordinates: #define GOTOXB 328.57971014492756 #define GOTOYB 82.92929292929294 #define GOTOXL 84.19289855072465 #define GOTOYL 332.62626262626264 Move the pickup location to the right.

6. Interactive Coordinate Adjustment

The Meta Llama-3.1-8B model allows users to interact with the Braccio Robot using natural language. The LLM is tailored to understand and adjust robot coordinates for pickup and drop tasks. Positioned as an AI mathematician specializing in coordinate plane adjustments, it processes user instructions to improve the robot's movement precision. Enabling iterative modifications, the chatbot ensures smoother operation, making it easier to fine-tune the robot's performance for real-world applications. The **load** keyword loads the initial, approximate pickup and drop-off coordinates from **YOLO** predictions. Once the user is satisfied with the adjustments, the **done** keyword finalizes the coordinates, which are then sent to the robot for testing. Figure (a) illustrates the LLM interaction through the chatbot hosted on **Streamlit**. **YOLO** predictions are displayed in Figure (b).



a) LLM interaction b) YOLO predictions

7. Video demonstration