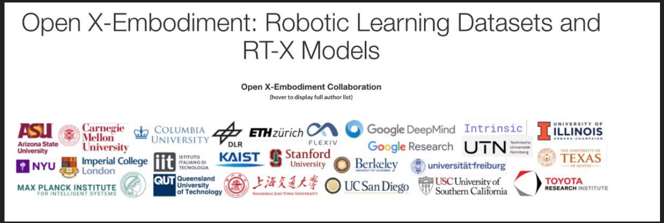
Remote Teleoperation Data Collection

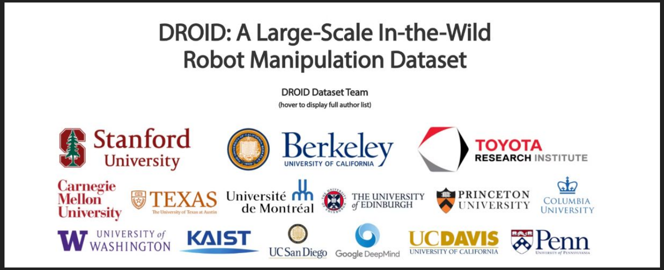
B.S.: Aaron Zheng, Akshaj Gupta, Kourosh Salahi, Ziteng (Ender) Ji

M.S.: Samuel Mankoff

ME206A Project Group 18

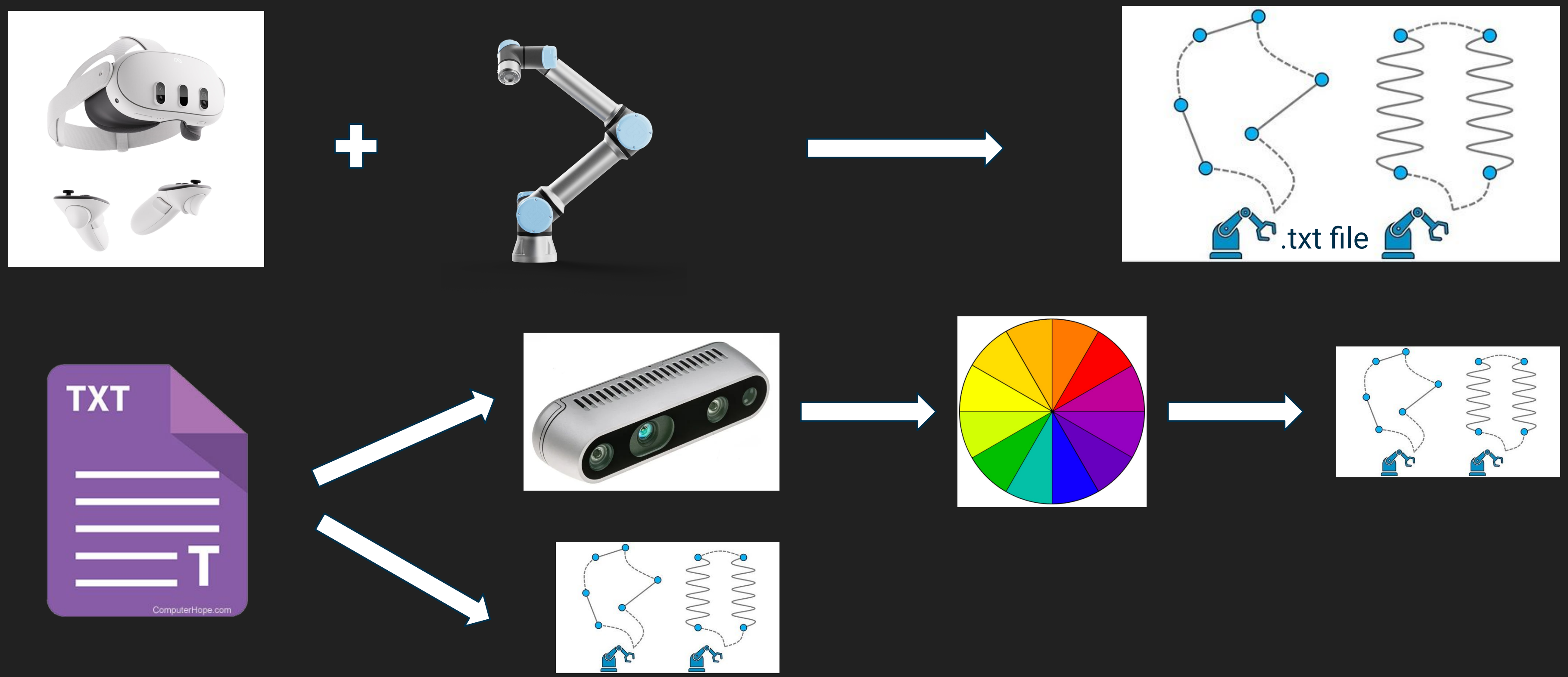
Why data collection is important?





- Modern robot learning is **data-hungry**: powerful policies including imitation learning and diffusion models only perform as well as the demonstrations and interactions they're trained on.
- Large-scale datasets like **Open X-Embodiment** and **DROID** demonstrate the critical importance of real-world robot data by pooling demonstrations from many research labs and diverse robotic platforms.
- **Our contribution**: We introduce a novel approach to collect real-world manipulation data through intuitive teleoperation using Meta Quest 3 VR and hand controllers.

Project Pipeline



Project Components

VR-Based Control (Planning)

Enable intuitive control of the UR7e robotic arm using Meta Quest 3 headset and controllers for natural 6-DoF manipulation.

Data Collection & Storage

Capture high-fidelity trajectory data during teleoperation sessions and save demonstrations for training robot learning models.

Trajectory Replay (Actuation)

Replay recorded trajectories by loading saved text files and executing the stored motion commands on the physical robot.

Vision-Based Selection Sorting (Sensing)

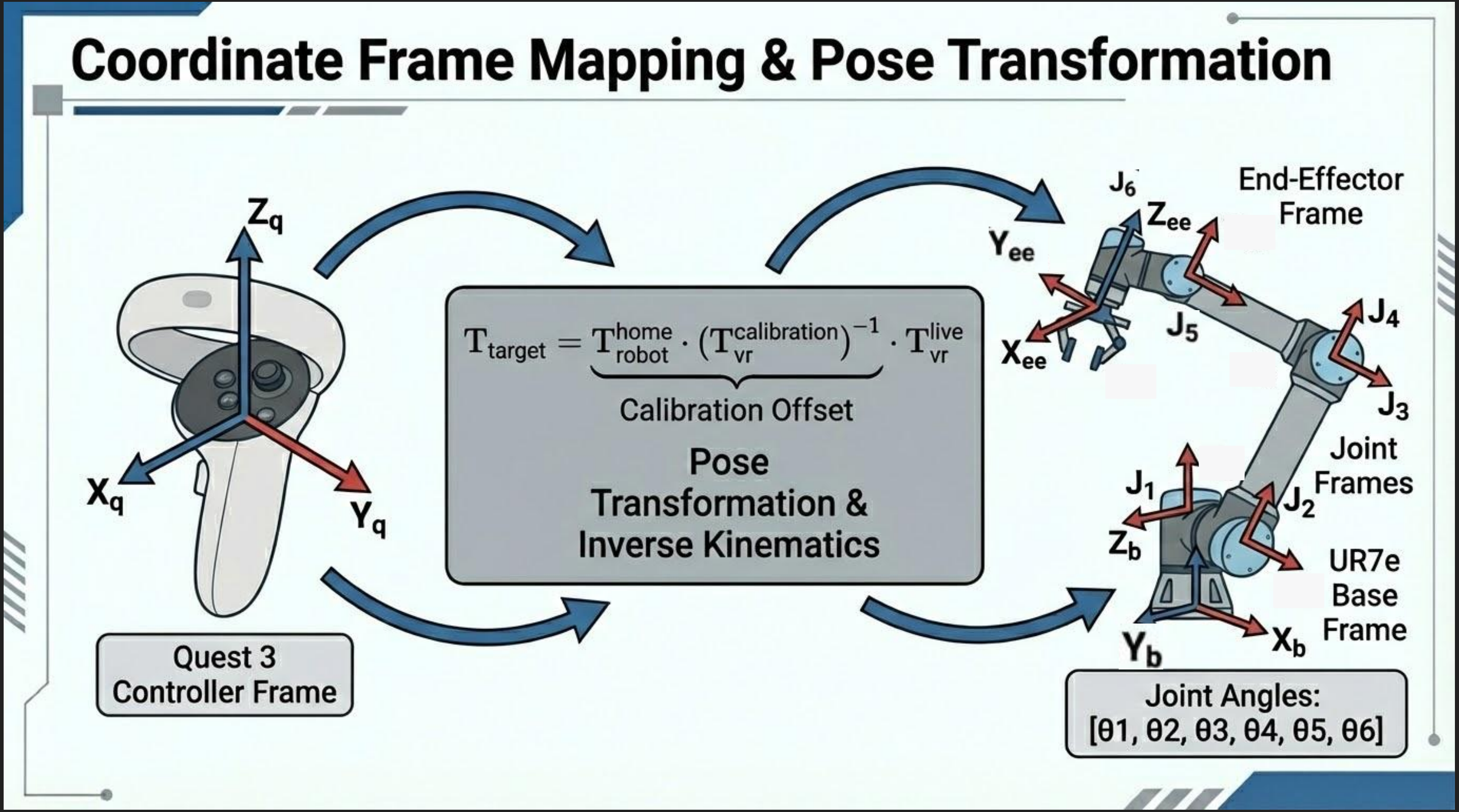
Intelligently select and replay specific trajectories based on visual input from a color sensor for conditional behavior.

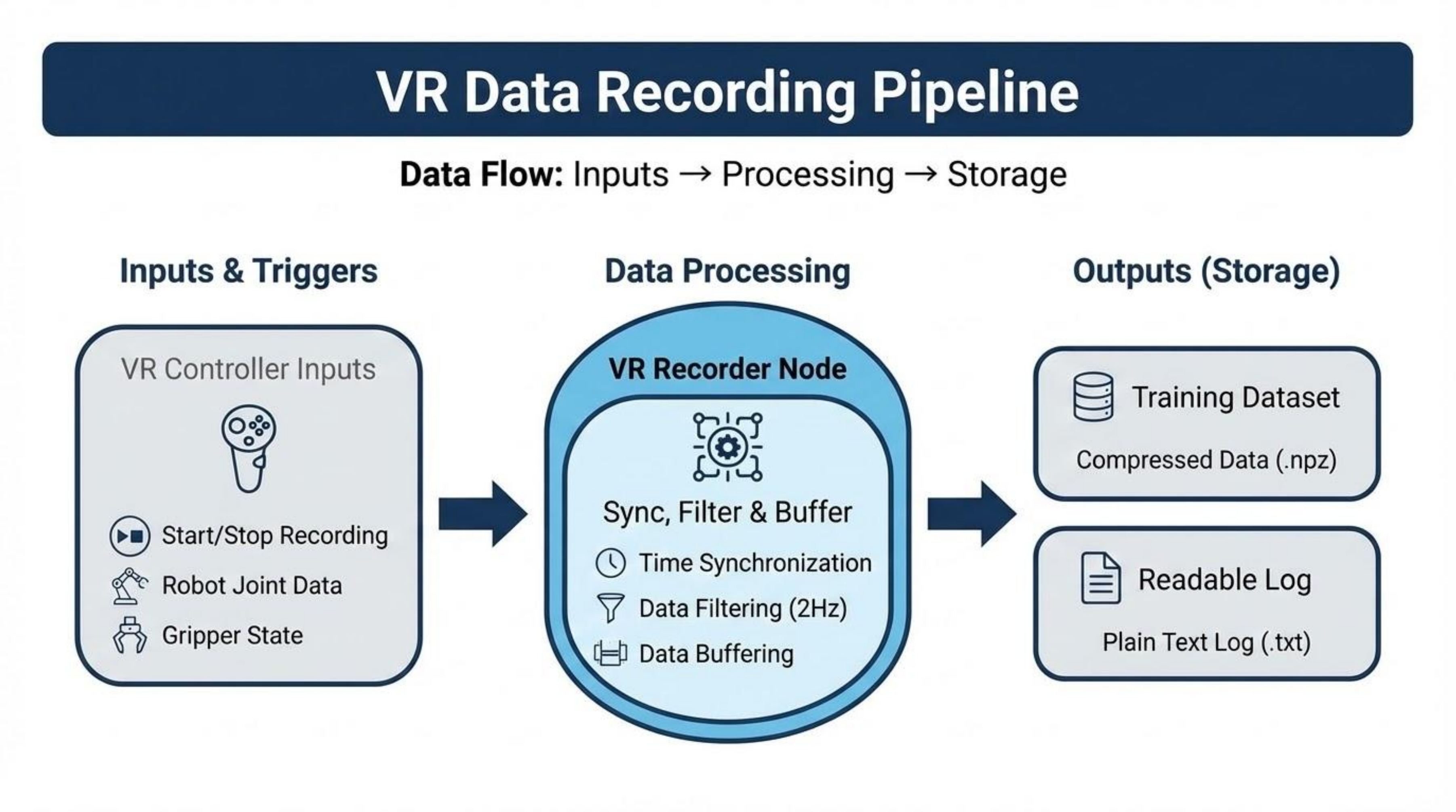
VR Based Control

SteamVR,
ALVR

replay,
ur7e_coms
moveit
ros2

VR Based Control

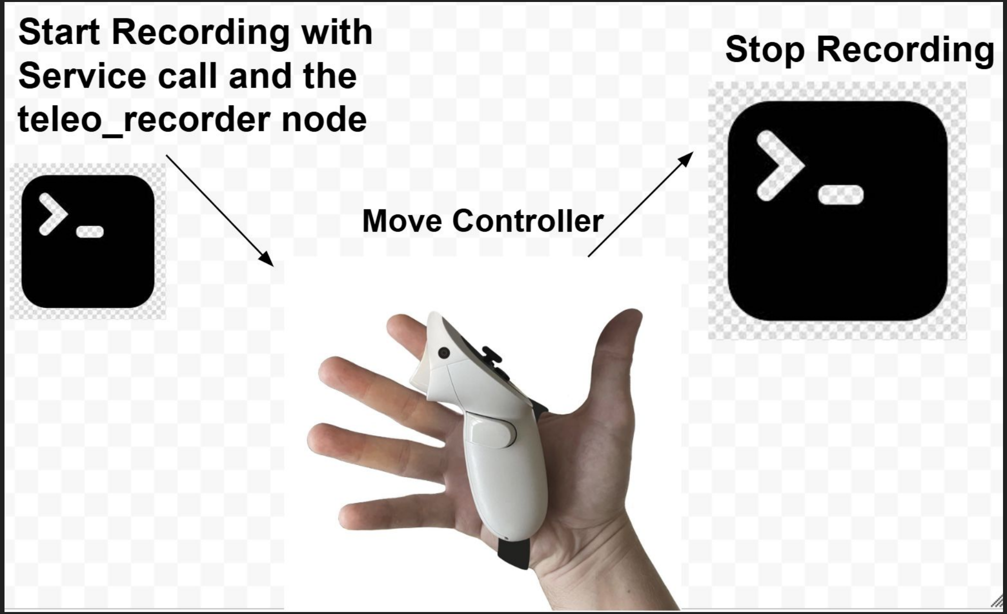



VR Data Recording Pipeline
Data Flow: Inputs → Processing → Storage
Inputs & Triggers
Data Processing
Outputs (Storage)
VR Controller Inputs
Start/Stop Recording
Robot Joint Data
Gripper State
VR Recorder Node
Sync, Filter & Buffer
Time Synchronization
Data Filtering (2Hz)
Data Buffering
Training Dataset
Compressed Data (.npz)
Readable Log
Plain Text Log (.txt)

VR Recording & Saving

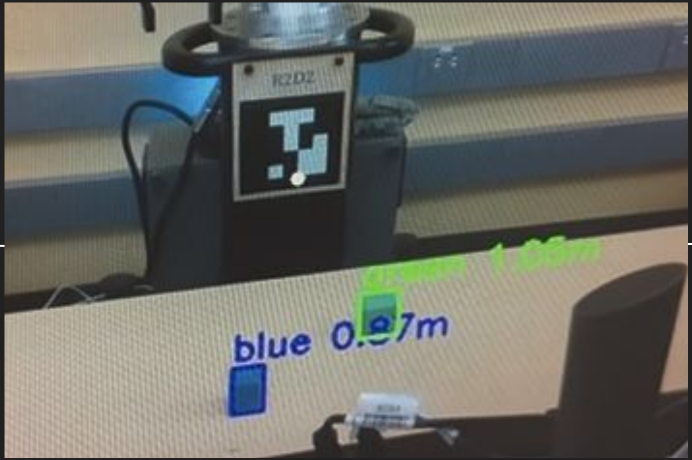
VR recorder, detailed steps:

1) Run teleo_recorder node
2) Run the corresponding service call (/start_recording) to start
3) While performing teleoperation, this service will sample ur7e joint state at 2Hz (with timer)
   a) Samples every 0.5s for smooth replay
4) Run (/stop_recording) to end and save joint states + gripper states in .txt file



Sensing & Replay

Ur7e_Executor



‘blue’, ‘green’, ‘none’

detected_color

camera_detection

full_execute

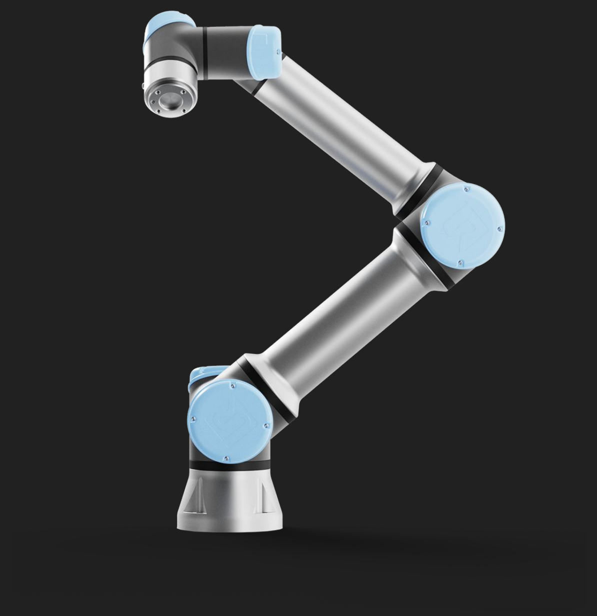
‘blue’, ‘green’

Populate
Job Queue

Replay
Trajectory

Biggest Challenge

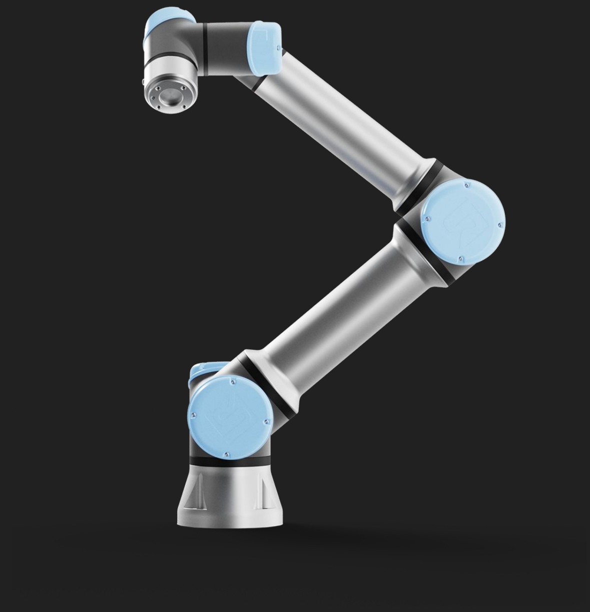
- Hardware
    - Connecting Quest headset and controller to the same local network as the lab computer
- Software
    - Smooth trajectory replay; aligning recording sampling rate and replay speed



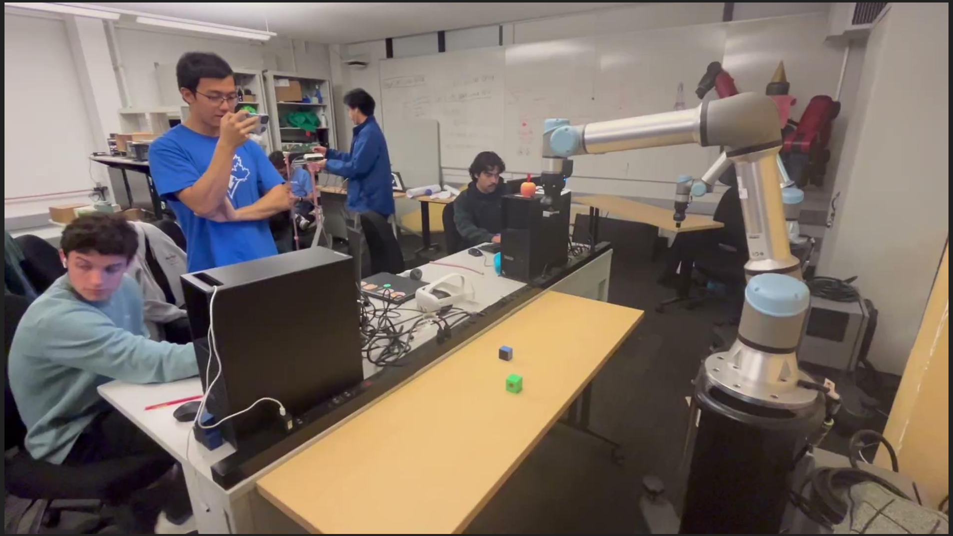
Future Considerations

Controls

- We experimented with controls
  - PID controls per joint
  - Speed limiting
  - Reduces noise
- Additionaly safety Robot bounding box
  - UR7e Proximity



Video Demo



Video Demo