01

3 POSSIBLE WAYS GROCERY ROBOT

Hard-coded + Planning	open-loop
Vision + DoF Motion Solving	decomposed, closed-loop enhanced
VLA/VLM + Isaac Lab + IL/RL + MoE	frontier, resource-consuming

Goal-Why we care (for groceries)

- User Stories and Requirements
  - Pick-up robustness
  - Have ability to communicate with environment
  - Cost

- Closed-loop robustness to small layout/pose/lighting shifts
- Scales to open-set SKUs and language commands.
- Develop a low-cost, adaptable robotic arm solution.

Previous Project: Augmenting Low-Cost Robotic Arms with YOLO-Based Perception and Vision-Language-Action Policies

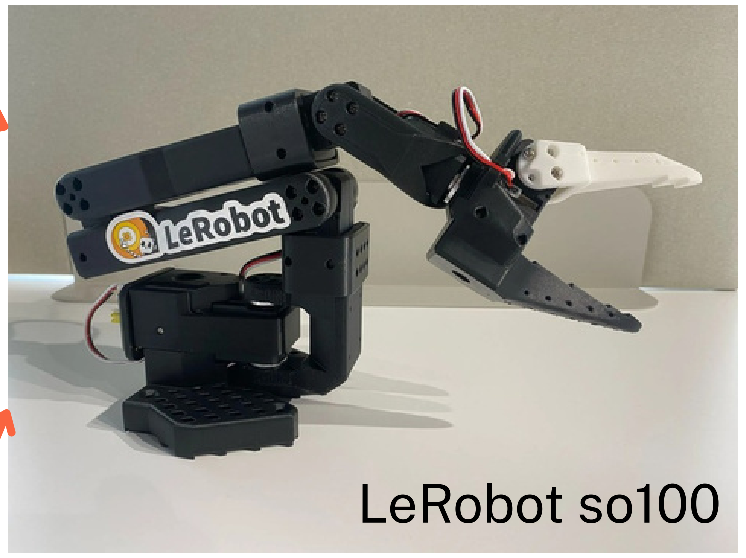
Workspace clutter and messy

Costly Industrial robotic solutions

Prof Introduced in class

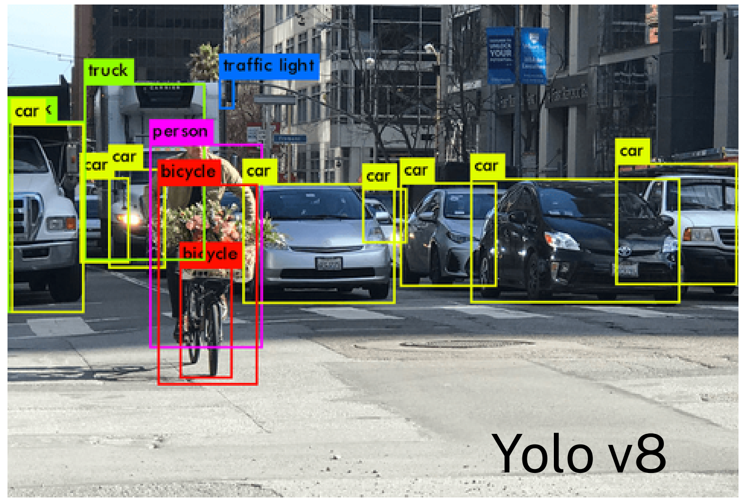
3D printed

assembly easy



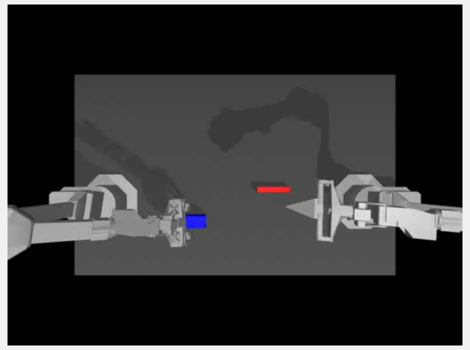
LeRobot so100

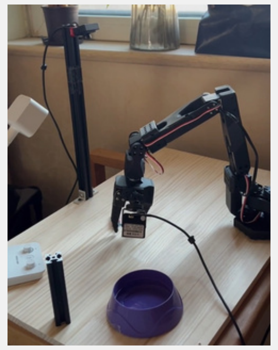
+



Yolo v8

Improve Performance on more generalized cases





- ACT vs Diffusion Policy
- Generalization Capability Limited

Goal

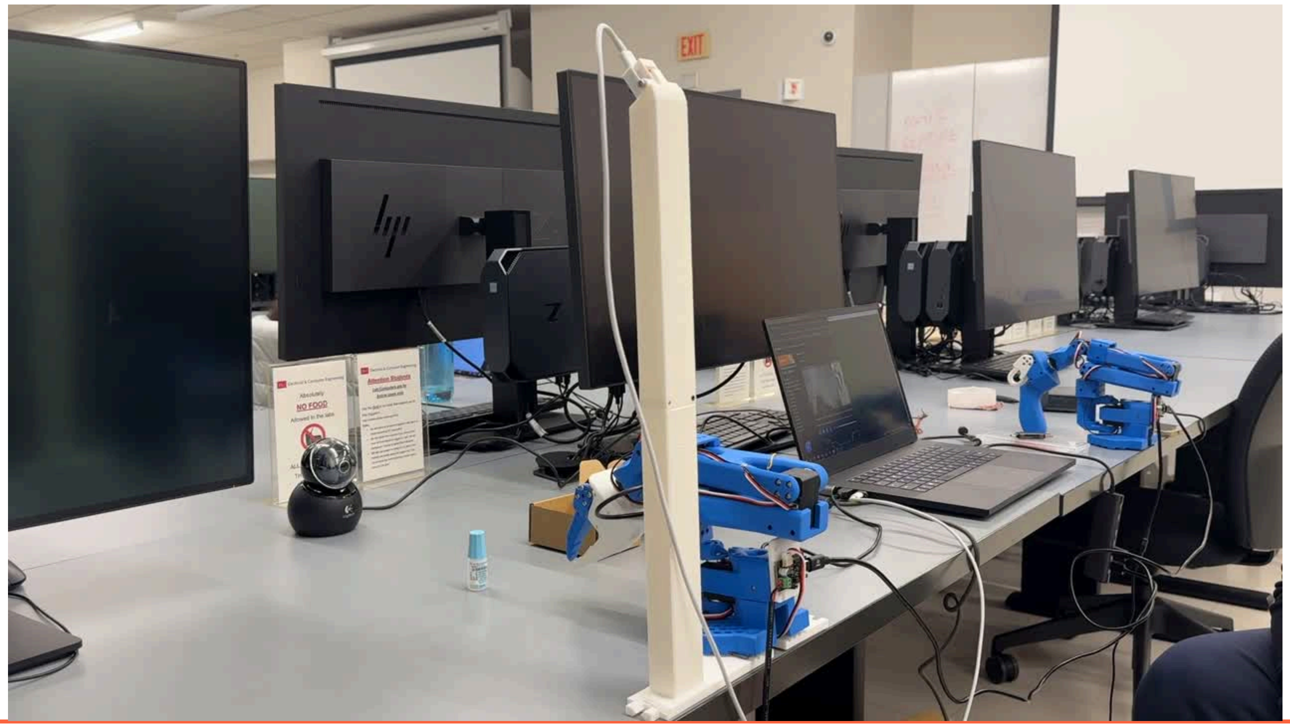
- Achieve generalization without retraining
- Integrate YOLOv8 for object detection and vision-language-action diffusion policies.
- Develop a low-cost, adaptable robotic arm solution.

Success Rate

Back to First Page

1 out of 2

50.00%



03

3 Paths in detail

•

Back to First Page

Path A: Hard-coded + Planning (open-loop)

- State machine + IK/trajectory (MoveIt/OMPL).
- Pros: reliable in fixed shelves/finite SKUs; predictable.
- Cons: poor generalization; heavy rule maintenance.

Possible realization

- Traditional labeling and motion/trajectory work
- Bad generalization task, might not be practical

Path B: Vision + DoF Motion Solving (decomposed, closed-loop enhanced)

- YOLO/seg → 6D pose → IK/trajectory + visual servo; small policy bits (grasp scoring).
- Pros: moderate robustness; moderate data; debuggable.
- Cons: module error accumulation; longer horizons brittle.

Possible realization

- Light weight training that is guranteed to be finished on the single GPU or So
- Better Robustness but may need a lot of tunning on the parameters on-site
- hard to simulate
- RL-Nav+YOLO

Path C: VLA/VLM + Isaac Lab + IL/RL + MoE (frontier)

- Millions of sim episodes + dozens of real demos for alignment.
- MoE: experts for navigation / seek-and-grasp / alignment / placing; LLM routes; planner/IK fallback for safety.
- Pros: higher ceiling; closed-loop robustness.
- Risks: trial-and-error; compute; sim2real.

Possible realization

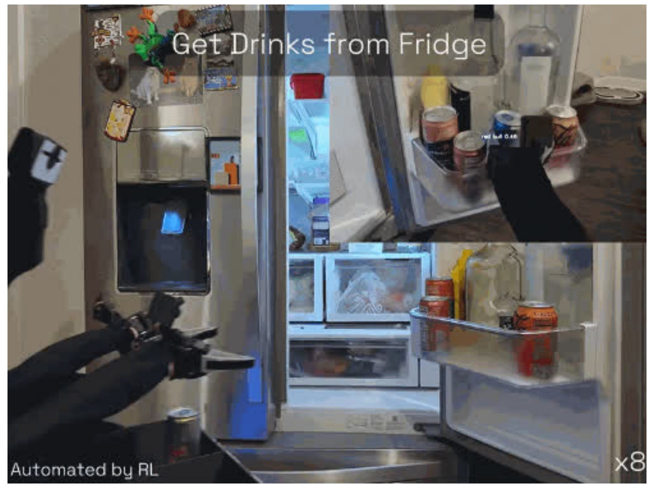
- Usage of Nvidia Issac Sim software
- Might need a large of computing resource
- Does not have way to explain the mechnism
- Novel, new, exploration on physical intelligence/Embodied AI
- MoE structure can be implemented
- LLM/VLM router+ Expert model on motion

Existing Frameworks

xLeRobot with Lerobot framework

low-cost mobile manipulator platform (sim + real teleop path).

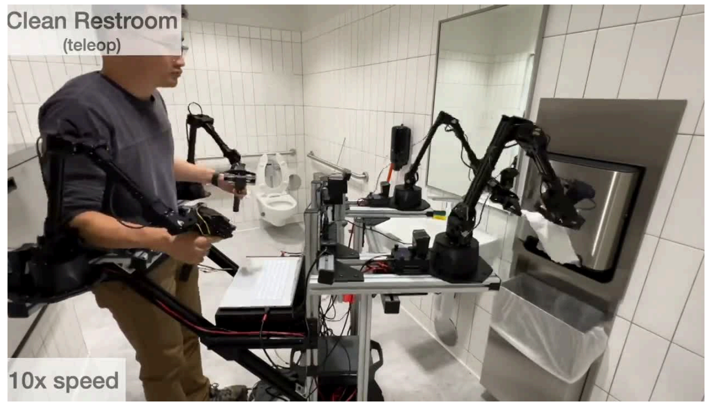
https://github.com/Vector-Wangel/XLeRobot



Stanford Mobile ALOHA

ong-horizon household tasks; sim/real data synergy; few dozen real demos can go far.

https://mobile-aloha.github.io/

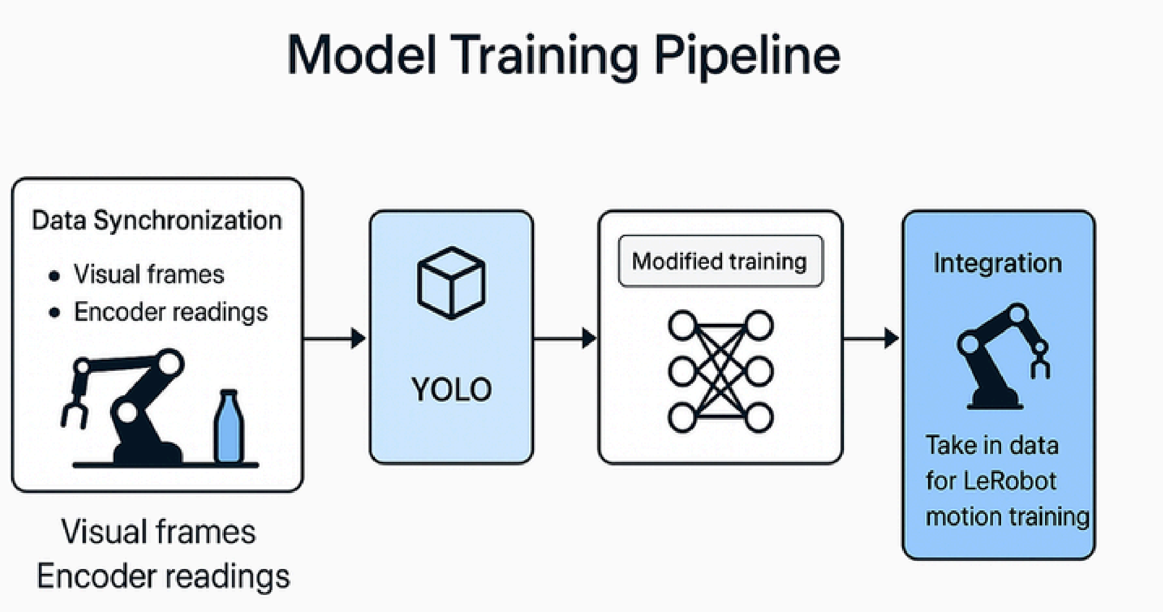


Pi0 & Pi0.5

large-scale sim + RL/IL workflows, potentially need school SCC cluster support

05

Potential Chanllenges



User Stories

- As a store associate, I want the robot to pick a known SKU from a fixed shelf and place it in a basket, so that it can fulfill simple orders. (UberEats market pickup)
- As a stocker, I want to restock from a tote to a shelf slot, so inventory is replenished. (Shop owner)
- As a safety officer, I want dynamic obstacle replanning so the robot yields to people/carts.
- As an operator, I want mis-grasp detection and up to two auto-retries with pose offsets, so success improves without human help.

Technical Questions

- “For the final demo, how should we weight reliability vs. novelty?”
- “Is a fixed shelf with N SKUs acceptable as baseline? Any must-have tasks?”
- “ if C underperforms B, should we keep C as research only?”
- “Do we have access to 1–2 GPUs / cloud? Any limits on data collection?”

15

Thank you!