LeKiwi: Identify and Collect Empty Bottles

Bonnie Liu
UCLA Computer Science
bonnieliu2002@ucla.edu

Abstract

This project presents the design, construction, and evaluation of a low-cost mobile manipulator built using the LeKiwi platform from Hugging Face's LeRobot ecosystem. The robot was assembled from approximately $1,000 of 3D-printed and off-the-shelf components, integrated with onboard sensing and control, and teleoperated to collect a real-world demonstration dataset for a bottle-collection task. The task required the robot to visually identify an empty water bottle, grasp it without disturbing a nearby full bottle, and deposit it into a cart. A policy based on Action Chunking with Transformers (ACT) was trained on 50 human demonstrations and deployed on the physical robot. Across 25 evaluation trials, the policy achieved a 56% success rate, consistently identifying the correct bottle and completing the task when a stable grasp was obtained. The results show that meaningful end-to-end manipulation performance can be achieved with accessible, open-source robotics tools, while also highlighting the practical challenges and reliability tradeoffs inherent to operating on low-cost hardware.

1 Introduction

Hugging Face is a company that develops tools for machine learning, including tools for robotics research. Their project LeRobot provides a framework for building low-cost robots (about $1,000 in materials) alongside open-source software support. LeRobot currently supports multiple platforms such as the SO101 arm, LeKiwi, and Hope Junior [2].

This study focuses on LeKiwi, a mobile manipulator designed for low-cost real-world robotics research. The project demonstrates the feasibility of end-to-end learning for the combined perception and control task of picking up an empty water bottle and depositing it into a cart.

2 Background

One of the key inspirations behind LeRobot is ALOHA (Affordable Learning for Open-source Hardware Agents) [7], introduced in the paper "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware." ALOHA is a low-cost bimanual robot designed to make research in dexterous manipulation more accessible. The system can be built for under $20,000, significantly cheaper than industrial-grade robotic platforms that often cost several times more. ALOHA demonstrated that, with careful hardware design and imitation learning techniques, low-cost robots could achieve complex tasks such as threading, insertion, and fine motor coordination. Its success highlighted the feasibility of democratizing robotic research through affordable hardware and open-source software.

Building upon the insights from ALOHA, Hugging Face's LeRobot project has adopted Action Chunking with Transformers (ACT) as one of its primary policy architectures. ACT is a lightweight (80M parameters), data-efficient sequence modeling approach tailored for robotics, where the policy predicts short sequences of future actions ("chunks") rather than issuing step-wise commands [7].

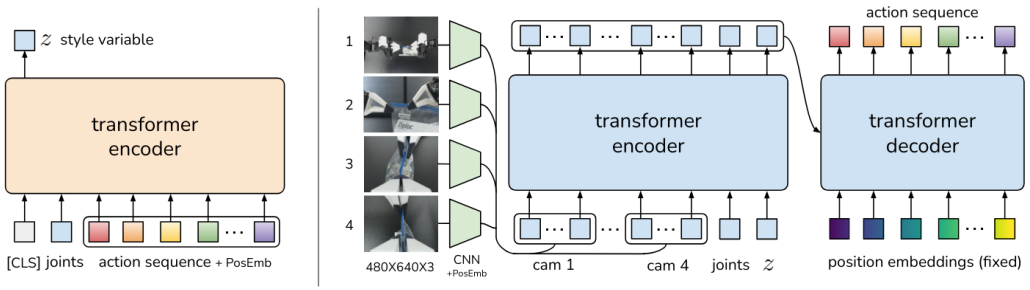

Figure 1: Architecture of Action Chunking with Transformers (ACT)

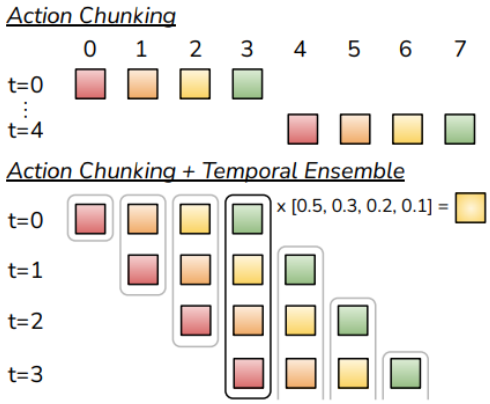

Figure 2: Action chunking combined with temporal ensembling

This formulation reflects the fact that many robot behaviors naturally unfold over multiple timesteps, and modeling these segments directly can simplify learning.

ACT employs a Conditional Variational Autoencoder (CVAE) structure: a ResNet-18 vision backbone first encodes images from multiple camera viewpoints; a transformer encoder integrates visual features, joint positions, and a learned latent variable z; and a transformer decoder generates coherent action sequences using cross-attention. The policy conditions on multiple RGB observations, current robot joint states, and the latent variable (set to zero during inference), and outputs a sequence of k future actions.

A key refinement—temporal ensembling—smooths predictions by exponentially weighting past and current outputs, stabilizing execution on real hardware. In practice, ACT trains in only a few hours on a single GPU and often attains high success rates with as few as 50 demonstrations, making it an accessible starting point for newcomers to imitation learning. Prior results show that ACT performs well in manipulation tasks, particularly under noisy or multimodal demonstrations, positioning it as a strong candidate for real-world systems such as LeKiwi.

3 Hardware

At the outset of the project, the initial goal was to build XLeRobot, a dual-arm mobile robot [5]. However, because software support for this platform was not yet available at the time, the design was shifted toward a single-arm build compatible with the existing LeKiwi software stack. This required modifying the original XLeRobot design but allowed the use of Hugging Face's open-source control tools.

The physical assembly of the robot involved 3D printing and fitting together all structural components—most of which were fabricated using the UCLA Hill Makerspace's Bambu Lab P1S printers due to their higher print quality compared to the Original Prusa MK4S machines at the UCLA Boelter Makerspace—followed by wiring and calibration. A Raspberry Pi 5 served as the onboard computer,

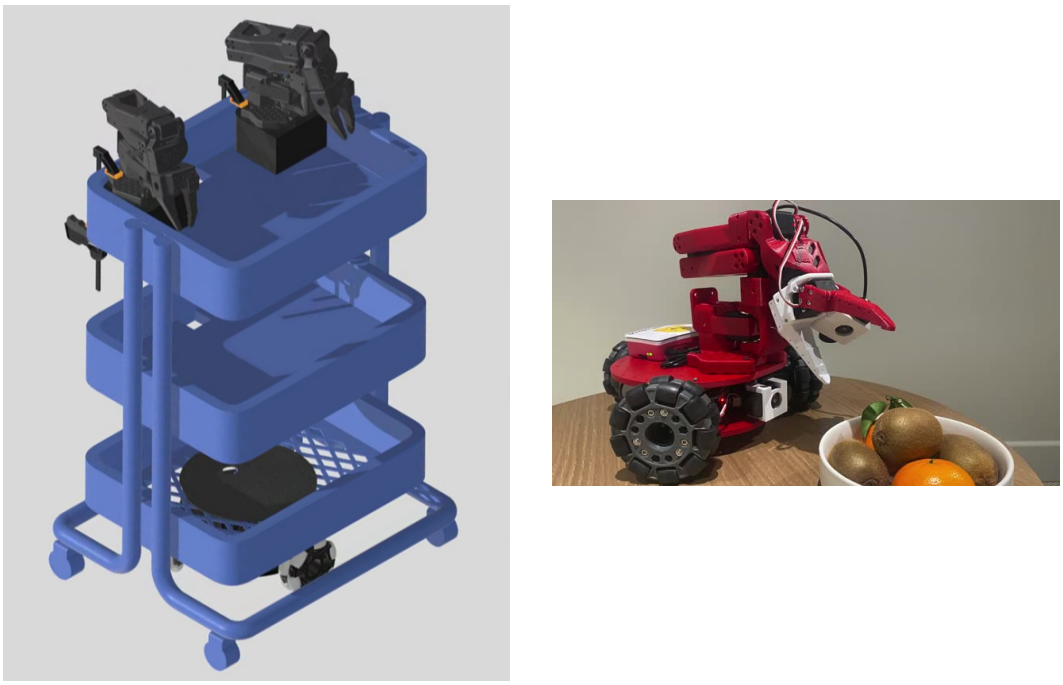

Figure 3: *Left:* XLeRobot, a bimanual mobile manipulator designed for dexterous dual-arm tasks. *Right:* LeKiwi, a simplified single-arm variant compatible with the current LeRobot software stack.

with SSH access enabling remote control and debugging. While this provided a convenient workflow for teleoperation, it also introduced challenges when paired with the robot's demanding power requirements. The USB-C to DC converter cables initially employed incompatible protocols, which caused voltage instability and occasional power surges that shut down the system mid-operation.

The motors presented another recurring issue. During experiments, the follower arm motors frequently overheated and failed, forcing repeated disassembly, replacement or resetting of faulty motors, and recalibration. These failures slowed progress and highlighted the fragility of using low-cost components.

4 Experiments

The primary task for the LeKiwi robot was collecting empty bottles. This task was chosen because it combines perception, manipulation, and basic planning, making it a useful benchmark for evaluating robot policies. The robot needed to recognize and distinguish between empty and full bottles and then reliably execute grasping and placement actions.

The task can be divided into three subtasks:

1. **Identify empty bottle.** Distinguish an empty bottle from a nearby full one based on visual input.
2. **Grasp empty bottle.** Align the arm and successfully grasp the empty bottle without disturbing the full one.
3. **Drop off empty bottle in cart.** Transport and release the empty bottle into a designated cart.

4.1 Data Collection

A controlled setup was created with one full and one empty water bottle, initialized randomly along a 15 cm line (at least 6 cm apart), similar to prior experiment setups [7]. 50 human teleoperated episodes

were recorded, each lasting 30 seconds at 30 FPS. A wrist-mounted camera provided egocentric observations for the model.

Figure 4: The positions of the empty water bottle and full water bottle were randomly initialized along a 15 cm line (at least 6 cm apart).

4.2 Training

The ACT model was trained with the following configuration:

- Steps: 100,000
- Batch size: 8
- Chunk size: 48
- Temporal ensemble coefficient: 0.2

Chunk size refers to the number of future actions the model predicts at once, and temporal ensemble coefficient refers to the weighting rate used for the exponential smoothing of predictions. Together, they shape how the policy trades off between responsiveness to new observations and smooth, temporally consistent action sequences.

4.3 Results

In this study, success was defined as the robot completing all three subtasks:

- Correctly identifying the empty water bottle.
- Grasping the empty bottle without disturbing the full one.
- Dropping the bottle into the cart.

Across 25 evaluation trials, the ACT policy achieved a 56% success rate.

A key observation from the experiments was that grasping emerged as the most difficult subtask. While the robot reliably distinguished between full and empty bottles using visual input, it often failed at securely grasping the empty bottle. This was typically due to small alignment errors or unstable grasps, given that the only reliable grasping point was the bottle cap. Once the robot successfully executed the grasp, however, the remaining subtasks—lifting and placing the bottle into the cart—were completed smoothly and consistently.

Trial	Empty Bottle Position	Full Bottle Position	Throughput (actions/sec)	Successful?
1	0	12	25.11	No
2	12	0	24.36	No
3	6	0	25.86	No
4	5	11	25.20	Yes
5	2	11	24.96	No
6	12	6	25.69	No
7	2	10	25.18	No
8	8	15	25.26	Yes
9	8	0	24.98	Yes
10	15	2	25.25	Yes
11	3	15	23.60	Yes
12	15	9	23.24	Yes
13	11	1	25.11	No
14	12	0	25.47	No
15	5	14	24.81	Yes
16	14	8	23.09	Yes
17	10	3	24.08	No
18	7	13	23.51	Yes
19	10	4	24.39	Yes
20	4	10	25.21	Yes
21	1	7	25.04	Yes
22	10	0	24.26	No
23	5	11	25.05	No
24	14	3	24.54	Yes
25	5	13	24.46	Yes

Table 1: Evaluation results for 25 trials of the empty bottle collection task.

Throughput was also measured as a proxy for inference speed. The best-performing ACT model achieved an average of 24.69 actions per second when run on an Apple M3 Mac Pro, with predicted actions transmitted over SSH to the robot.

Key Takeaways:

- Visual identification of the empty bottle was robust across trials. This indicates that the ResNet-18 vision backbone was effective at extracting the visual features needed to distinguish between empty and full bottles.
- The bottleneck in performance was grasping, which accounted for most failures. This made sense since the wrist camera was a 640x480 RGB camera that cost $30, not an advanced depth camera.
- Once grasping was successful, the subsequent actions (lifting and depositing) almost always succeeded. This was expected given that, once a stable grasp was achieved, the remaining trajectory required only coarse positional control rather than precise end-effector alignment.

5 Limitations and Future Work

5.1 Hardware Improvements

5.1.1 Depth Perception

Current wrist camera lacks depth sensing, making grasping unstable. Future iterations could mount an Intel RealSense camera either directly on the wrist (for fine manipulation) or above the arm (for global perception).

5.1.2 Motor Reliability

Frequent failures of Feetech motors severely limited experimentation. Possible solutions include switching to Dynamixel motors, which have robust Hugging Face software support [2]. However, this would likely require custom CAD design and 3D-printed adapters.

5.2 Policy Development

5.2.1 Task Factorization

Current ACT policy controls both locomotion and grasping within one model. These behaviors are largely independent, so training separate locomotion and grasping policies may yield better results and more efficient training.

5.2.2 Alternative Policies

While ACT is powerful, it is computationally heavy. Other members of the LeRobot community have noted that simpler policies—such as Decoder-Only Transformers (DOT)—can match or even outperform ACT while offering substantially faster inference [3]. DOT models follow the standard autoregressive transformer architecture, predicting each action token sequentially without the conditional variational components used in ACT, making them both lighter and more efficient at deployment.

Additionally, exploring lightweight diffusion models could provide additional robustness in stochastic or highly multimodal settings, where capturing uncertainty is essential for reliable control. Diffusion-based policies generate actions through iterative denoising steps, which enables them to model complex action distributions and recover gracefully from perturbations. However, despite recent progress in reducing model size and accelerating sampling, diffusion methods still require multiple inference steps per action. This iterative sampling process introduces latency that can be prohibitive for real-time control on physical robots, particularly those with high-frequency actuation requirements. As a result, diffusion models remain promising for robustness but currently face practical limitations when deployed on embedded hardware or time-critical systems [6].

5.3 Simulation and Hybrid Training

5.3.1 Simulation Platforms

Although existing simulation frameworks do not yet provide native support for mobile platforms such as LeKiwi, related tooling offers a foundation for scalable pretraining. In particular, gym-hil is a recently introduced suite of Gymnasium-compatible environments designed for human-in-the-loop reinforcement learning and integrated directly into the LeRobot ecosystem [2]. The framework currently focuses on manipulator-centric tasks—most notably a MuJoCo-based simulation of a Franka Panda arm—enabling teleoperation, demonstration collection, and policy learning within a standardized LeRobot dataset format.

Despite this compatibility, gym-hil does not presently include environments for mobile-base robots. Consequently, applying its human-guided simulation capabilities to LeKiwi would require extending the framework with custom environments that model mobile locomotion and navigation. Nevertheless, gym-hil provides a coherent interface, data pipeline, and training workflow that could serve as a robust foundation for developing pretraining procedures tailored to mobile robot platforms.

5.3.2 Human-in-the-Loop Learning

Incorporating HIL-SERL (Sample-Efficient Reinforcement Learning with human feedback) presents a promising direction for accelerating policy improvement directly on physical hardware. HIL-SERL integrates real-time human interventions, corrective actions, or evaluative feedback into the learning loop, enabling policies to recover from suboptimal trajectories and reduce the number of required on-robot interactions [4]. If adapted to LeKiwi, this framework could substantially mitigate the data-efficiency challenges typically associated with on-policy learning in mobile robots, while providing a structured mechanism for leveraging human expertise during early training phases.

5.3.3 Scripted vs. Human Data

Prior work has shown that scripted demonstrations—those generated programmatically or through carefully engineered controllers—often yield higher-quality training data than raw human demonstrations [1]. Scripted trajectories tend to be consistent, noise-free, and perfectly aligned with task objectives, which helps policies converge more quickly and reduces the variance introduced by suboptimal human actions. In contrast, human-collected data, while richer and more diverse, can be noisy, inconsistent, and occasionally incorrect, leading to slower or less stable learning when used in isolation.

However, combining both modalities can provide complementary benefits. Scripted demonstrations supply clean exemplars of ideal behavior, establishing a strong baseline from which the policy can learn precise task structure. Human data introduces variability, edge cases, and alternative strategies that scripted controllers may not capture—improving robustness and generalization. Blending scripted precision with human flexibility can therefore produce policies that not only perform well under nominal conditions but also adapt effectively to real-world deviations and uncertainties.

5.4 Future Robotic Tasks

Building on the bottle collection task, future experiments could include:

- Table-to-trash navigation and disposal, broken into subtasks: approach table, pick up trash, navigate to trash can, and dispose.
- XLeRobot Expansion: Once software support matures, extending to bimanual manipulation (e.g., holding and opening a container) will broaden the scope of tasks achievable under low-cost robotics.

6 Acknowledgments

Thank you to the members of the UCLA Neural Engineering and Computation Lab for their support and feedback throughout this project. I am especially grateful to Professor Jonathan Kao for his guidance and for providing the funding that made this work possible. I also thank Xu Yan and John Zhou for their insightful discussions and mentorship.

Thank you as well to the following people who contributed significantly to the construction and development of the robot: Sarah Darzacq, Ali Shreif, Matthew Li, Michael Yuan, and Lime Yao. I additionally thank Jason Chan for helpful discussions regarding the LeRobot software. I am also grateful to Ilia Larchenko, Pepijn Kooijmans, and other members of the LeRobot Discord community for their valuable advice on working with the LeRobot framework.

References

[1] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 80375–80395. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/fe692980c5d9732cf153ce27947653a7-Paper-Conference.pdf`.

[2] Hugging Face. Lerobot documentation. Online documentation, 2024. URL `https://huggingface.co/docs/lerobot/index`. Accessed: 2025-08-20.

[3] Ilia Larchenko. Lekiwi: 3d-printed mobile manipulator robot | teleoperate, train policies, and control with ai. YouTube, 2025. URL https://www.youtube.com/watch?v=lJ6PH--el54.

[4] Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2410.21845*, 2024. URL https://hil-serl.github.io/static/hil-serl-paper.pdf.

[5] XLeRobot Project. Xlerobot documentation. Online documentation, 2025. URL https://xlerobot.readthedocs.io/en/latest/.

[6] Rosa Wolf, Yitian Shi, Sheng Liu, and Rania Rayyes. Diffusion models for robotic manipulation: A survey. *arXiv preprint arXiv:2504.08438*, 2025. URL https://arxiv.org/pdf/2504.08438.

[7] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. doi: 10.15607/RSS.2023.XIX.016. URL https://arxiv.org/abs/2304.13705.

7 Appendix

A detailed guide for building and setting up the LeKiwi robot can be found here: (https://docs.google.com/document/d/1IN3tzRw-3-lilND3QcM1UuEyitI-AkRCHnf5MkvUZ6c/edit?usp=sharing).

Video demonstrations of the robot in action can be found here: (https://bonnieliu2002.github.io/lekiwi.html).