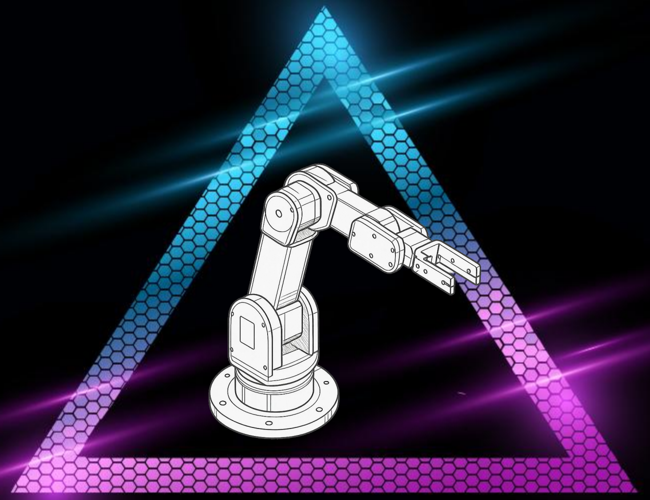
Analytics Vidhya | DATAHACK SUMMIT

Hack Session

From Language to Robotics: Practical Lessons Bridging LLMs, RL, and AI



Speaker

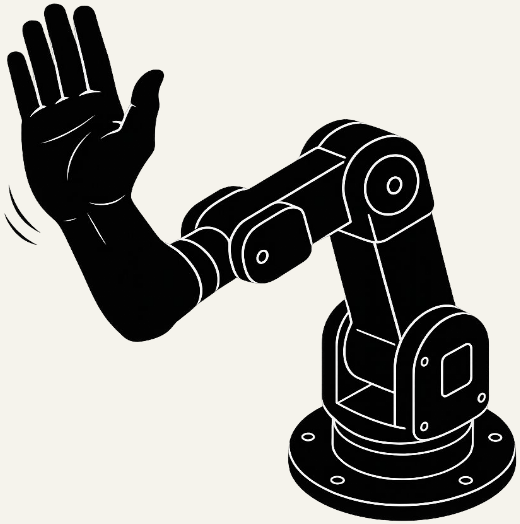
Logesh Kumar Umapathi

Machine Learning Consultant | Blackbox.ai

Hi 👋

I am Logesh Kumar Umapathi

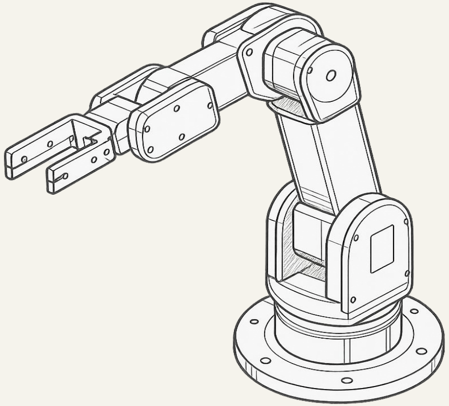
Machine Learning Consultant @ Blackbox.ai



Twitter: @logesh_umapathi
Linkedin: www.linkedin.com/in/logeshkumaru/
Website: logeshumapathi.com
Research : https://scholar.google.com/citations?user=eWwEqToAAAAJ&hl=en&authuser=1

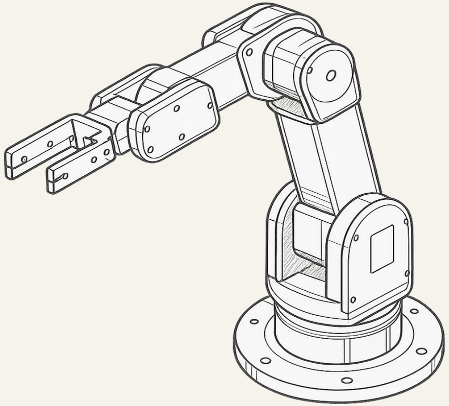
Robotics : One of my nine lives

Graham Weaver: https://www.grahamweaver.com/blog/stanford-graduate-business-school-last-lecture-2024



Agenda

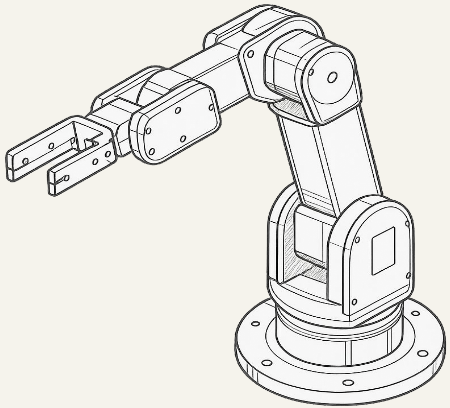
1. A Glimpse
2. Meet the Hardware - So100
3. Fundamentals of Imitation Learning
4. Popular Policies:
    a. ACT
    b. SmolVLA
    c. Pi0
5. How to record a dataset
6. How not to train the arm
7. Demo




LG
McCafé

Meet the Hardware

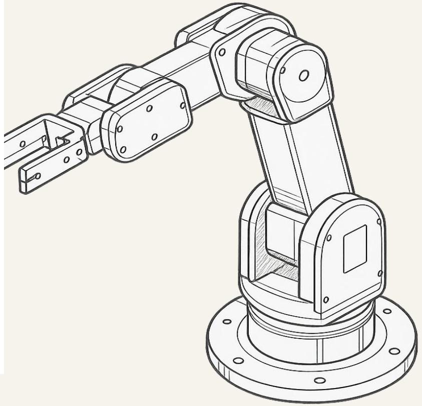
Spec	Value
Degrees of freedom	6 (+ 1 gripper)
Motors	6 serial bus servos
Build	Mostly 3-D printed PLA
Cost	~250 USD per Arm (India: INR 40000)
Strength	200–300 g payload (w/ 12 V 30 kg cm servos)
Reach	~40 cm, ±180° sweep



Academic Labs : Cost of building

Bill of Materials

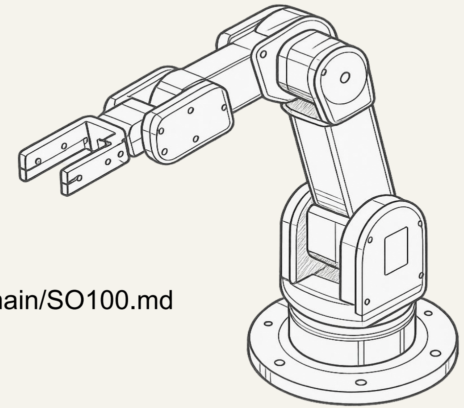
Part	Quantity	Link	Price (per unit)
Robots			
ViperX 300 Robot Arm 6DOF	2	https://www.trossenrobotics.com/viperx-300-robot-arm-6dof.aspx	$5,695.95
WidowX 250 Robot Arm 6DOF	2	https://www.trossenrobotics.com/widowx-250-robot-arm-6dof.aspx	$3,295.95
Robot Cage			
Standing Desk (48x30”, Black)	1	https://a.co/d/4JuSVhu	$199.99
Extrusion Bars Black, 10PCS 1220mm(48")	1	https://a.co/d/8ywG7Eu	$99.99
Extrusion Bars Black, 10PCS 1000mm(39.4")	1	https://a.co/d/8ywG7Eu	$83.99
Extrusion Bars Black, 4PCS 150mm(5.9")	2	https://a.co/d/8ywG7Eu	$11.99
Corner Bracket 20Set M5x8	3	https://a.co/d/9BCcU5S	$28.99
Corner bracket Black L-4 with Screw	4	https://a.co/d/ihrw6YW	$15.99
Rope Crimping Tool (15 inch for 3/64" to 1/8")	1	https://a.co/d/1t1EsyZ	$35.97



Aloha BOM

BOM

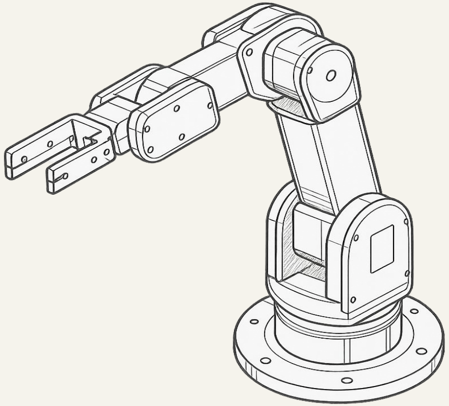
Item	Qty	Cost (INR)
12v ST3215 Servo	12	24000
Motor Control Board	2	1048
Top USB camera	1	1198
Wrist USB camera	1	3000
Table Clamp 4pcs	1	1101
Power Supply 12v	2	1390
3D Printing Parts		7600
Total		~**40,000** INR



BOM for rest of the world and Printing guide: https://github.com/TheRobotStudio/SO-ARM100/blob/main/SO100.md

Imitation Learning 101

1. Supervised learning on <image, state> → <next action> pairs
2. Advantages: simple, data-efficient, safe (no exploration)
3. Limitations: covariate shift & unseen states → compounding errors



What is a policy?

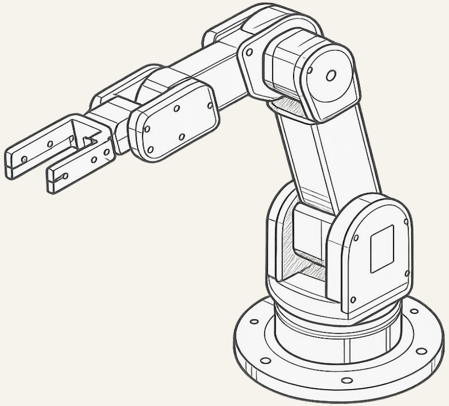
- A policy is a function / model (π) that maps the current state (S) of the robot to an action

  $\pi : S \rightarrow A$

  S - the state is usually the position of the robot, the cameras and sensors feed, and the text instructions.

  A - the actions the 6-DOF (degrees of freedom)

  cartesian position (x, y, z, rx, ry, rz), the angles of the joints…



https://docs.phospho.ai/learn/policies

Action Chunking Transformer

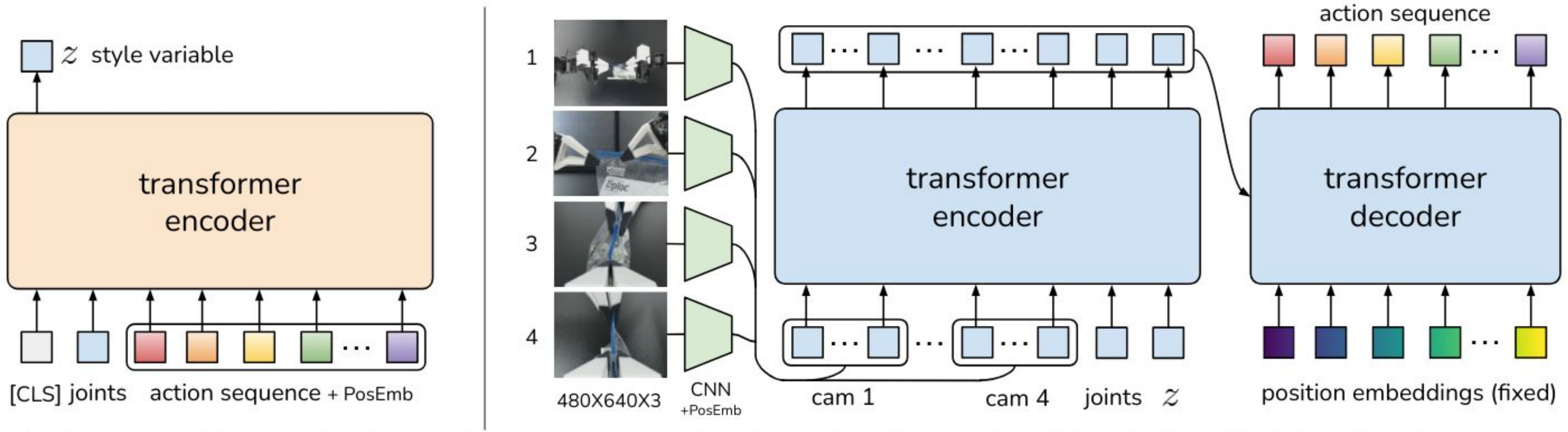
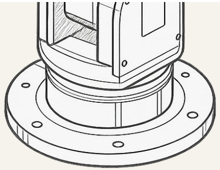
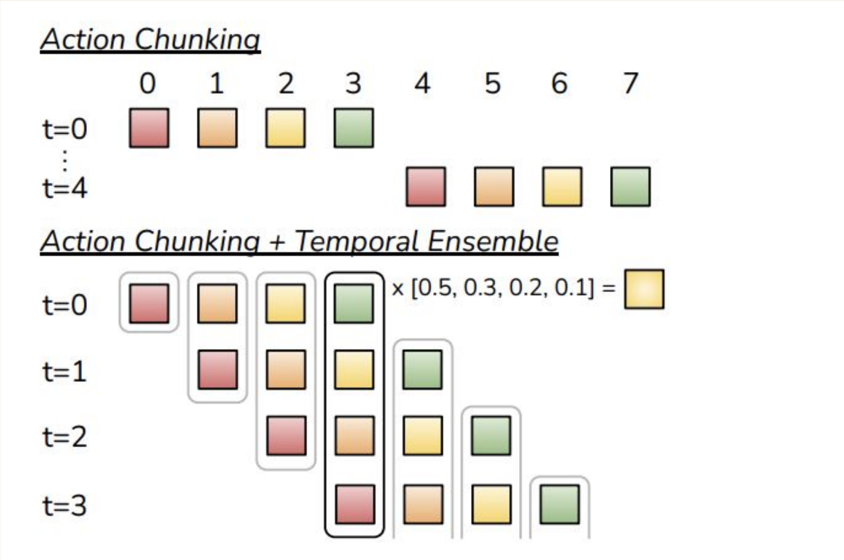


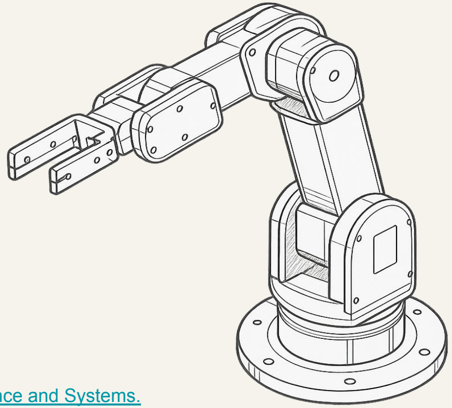
Fig. 4: *Architecture of Action Chunking with Transformers (ACT).* We train ACT as a Conditional VAE (CVAE), which has an encoder and a decoder. *Left:* The encoder of the CVAE compresses action sequence and joint observation into z, the style variable. The encoder is discarded at test time. *Right:* The decoder or policy of ACT synthesizes images from multiple viewpoints, joint positions, and z with a transformer encoder, and predicts a sequence of actions with a transformer decoder. z is simply set to the mean of the prior (i.e. zero) at test time.



Zhao et al., [Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. Robotics: Science and Systems.]

Action Chunking Transformer

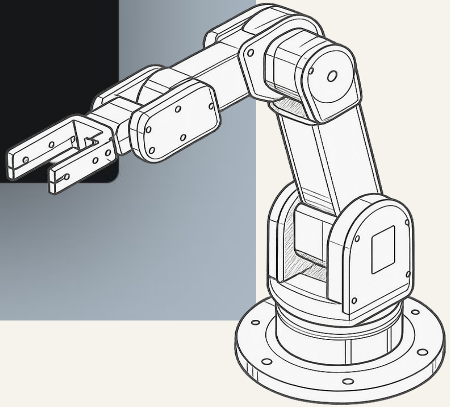




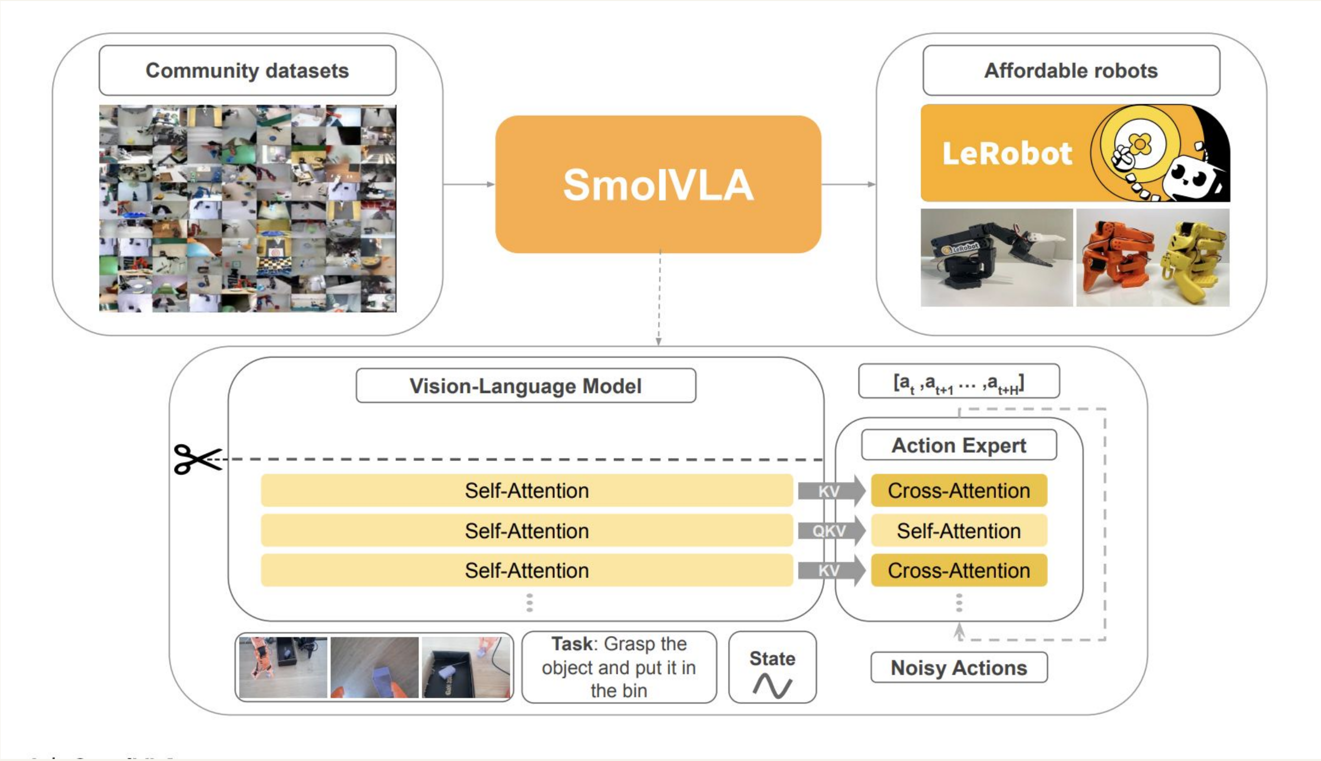
Zhao et al., [Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. Robotics: Science and Systems.]()

Train

```
python src/lerobot/scripts/train.py \
              --dataset.repo_id=repo_id/dataset_name
\             --policy.type=act \
              --output_dir=outputs/train/output_dir \
              --job_name=act_so100_test \
              --policy.device=mps \
              --wandb.enable=true \
              --policy.push_to_hub=False
```



SmolVLA



Shukor et al., SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics

SmolVLA

Policy	VLA pt.	Success Rate (%) — Real World			
		Pick-Place	Stacking	Sorting	Avg.
Single-task Training					
SmolVLA (0.45B)	No	55	45	20	40
Multi-task Training					
SmolVLA (0.45B)	No	80	40	35	51.7
SmolVLA (0.45B)	Yes	75	90	70	78.3

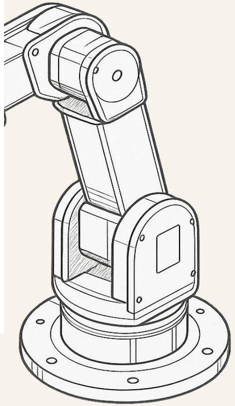
community data helps!

Policy	Success Rate (%) — Real World			
	Pick-Place	Stacking	Sorting	Avg.
Single-task Training				
ACT	70	50	25	48.3
Multi-task Training				
π_0 (3.5B)	100	40	45	61.7
SmolVLA (0.45B)	75	90	70	78.3

Table 3 | Real-world benchmarks (SO100). Success rate (%) across three tasks using policies trained in multi-task and single-task settings.

Policy	Success Rate (%) — Real World	
	In Distribution	Out of Distribution
Single-task Training		
ACT	70	40
SmolVLA (0.45B)	90	50

Table 4 | Real-world benchmark (SO101). Success rate (%) for the Pick-Place-Lego task using policies trained in single-task setting.

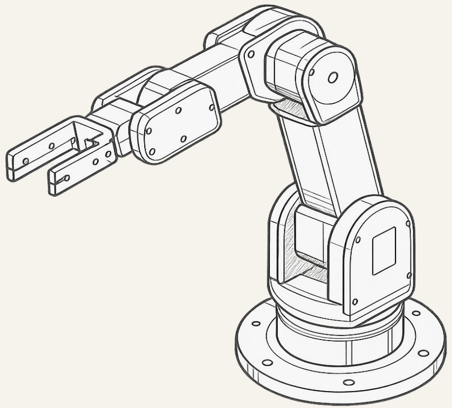


Shukor et al., SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics

SmolVLA

- Flow Matching Transformer (~100M parameters) : direct, non-autoregressive prediction of continuous actions

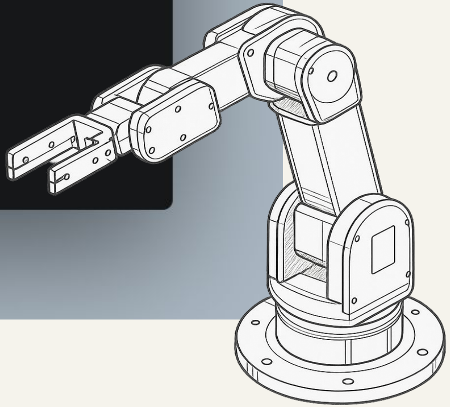
- Visual Token Reduction: 512×512 image is compressed into just 64 tokens, instead of 1024
-
- Faster Inference via Layer Skipping: action expert only attends to VLM features up to half the total layers

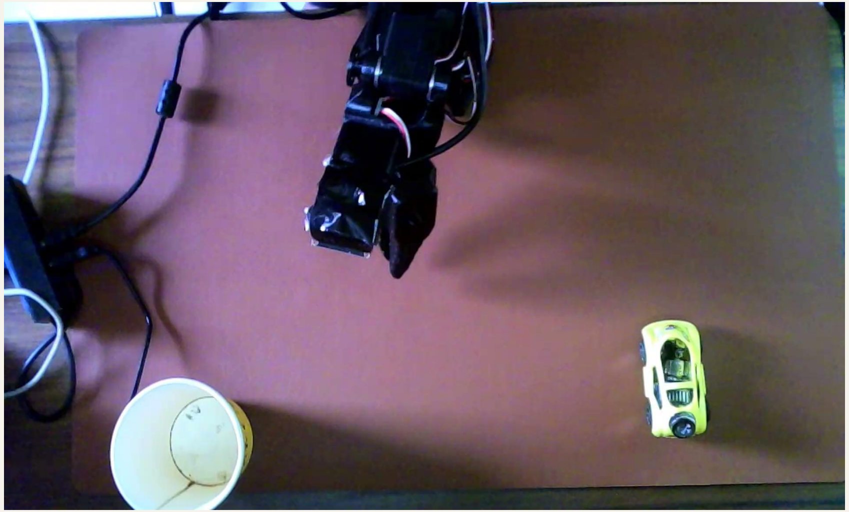


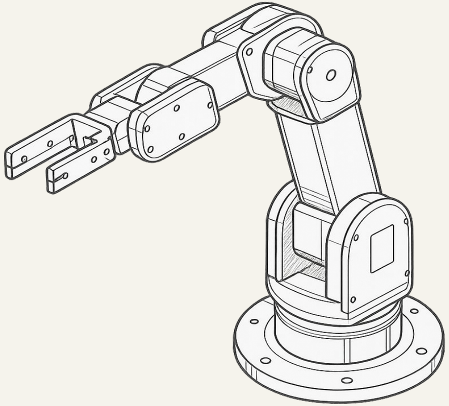
Shukor et al., SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics

Train

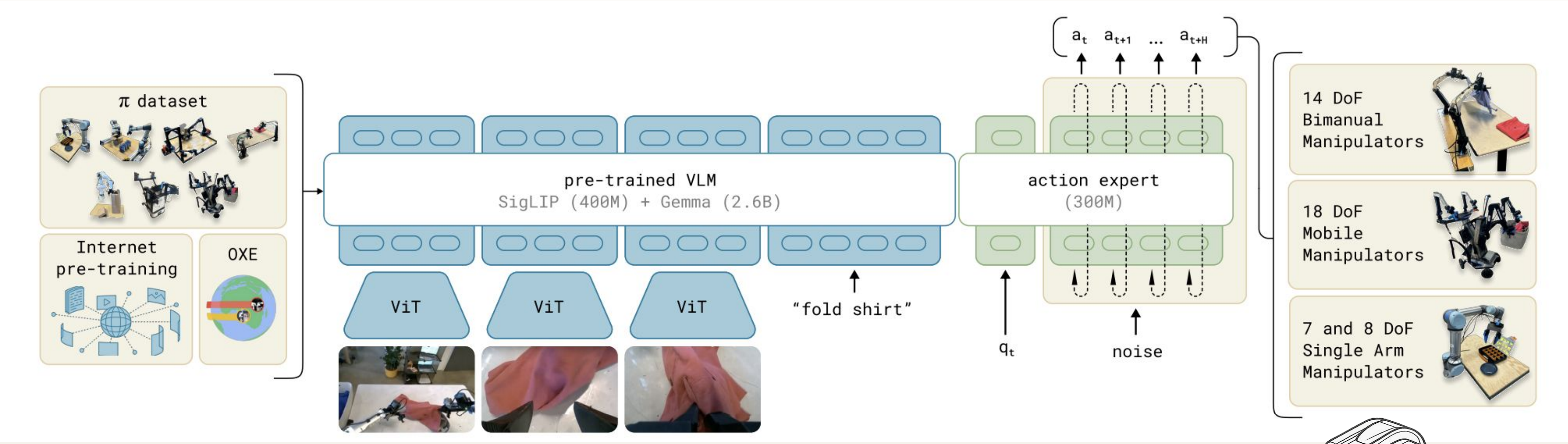
```
python src/lerobot/scripts/train.py \
              --policy.path=lerobot/smolvla_base \
              --dataset.repo_id=infinitylogesh/grab_toy_car \
              --batch_size=128 \
              --steps=20000 \
              --save_freq=10000 \
              --output_dir=outputs/train/grab_toy_car_2_cam_smolvla
\             --job_name=grab_toy_car_2_cam_smolvla \
              --policy.device=cuda \
              --policy.repo_id=infinitylogesh \
              --wandb.enable=true \
              --policy.push_to_hub=true
```



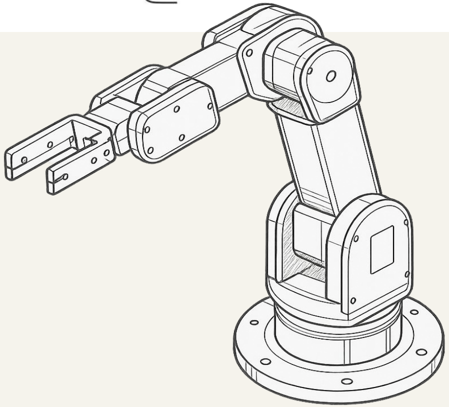




Pi0

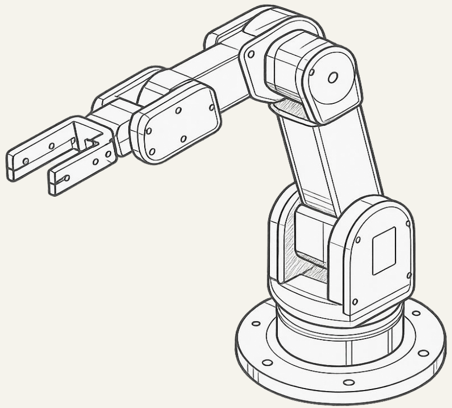


$$\mathbf{A}_t = [a_t, a_{t+1}, ..., a_{t+H-1}]$$
$$o_t = [I_t^1, ..., I_t^n, \ell_t, q_t]$$



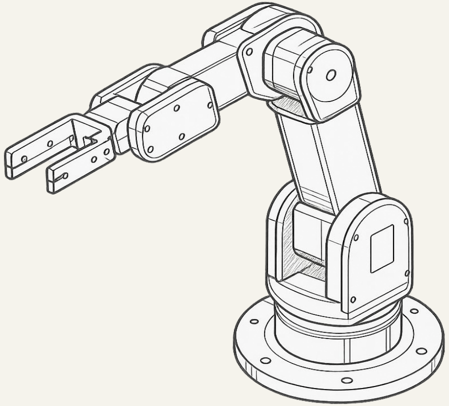
Black et al., π0: A Vision-Language-Action Flow Model for General Robot Control.

How to record a dataset



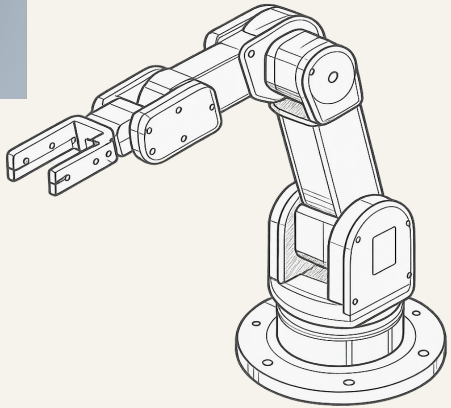
Assemble

Follow this guide:
https://huggingface.co/docs/lerobot/en/so100



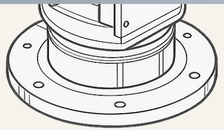
Calibration

```
python -m lerobot.calibrate \
    --teleop.type=so100_leader \
    --teleop.port=/dev/tty.usbmodem59591125071
\   --teleop.id=my_leader
```



Teleoperation

```
python -m lerobot.teleoperate \
    --robot.type=so100_follower \
    --robot.port=/dev/tty.usbmodem58FA0958621 \
    --robot.id=my_follower \
    --robot.cameras="{ wrist: {type: opencv, index_or_path: 0, width: 1280, height: 720, fps: 30},top: {type: opencv, index_or_path: 1, width: 1280, height: 720, fps: 30}}" \
    --teleop.type=so100_leader \
    --teleop.port=/dev/tty.usbmodem59591125071 \
    --teleop.id=my_leader \
    --display_data=true
```



Controlling the Arm ✋→🤖

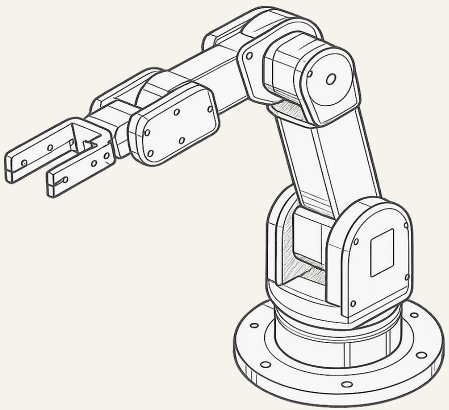
1. Leader–Follower Teleoperation

- Two identical arms
- Encoders on leader stream joint angles → follower mirrors
- Intuitive, latency-free, but doubles hardware cost

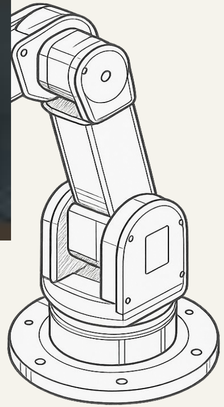
2. Game-pad Teleoperation (PS4 Controller)

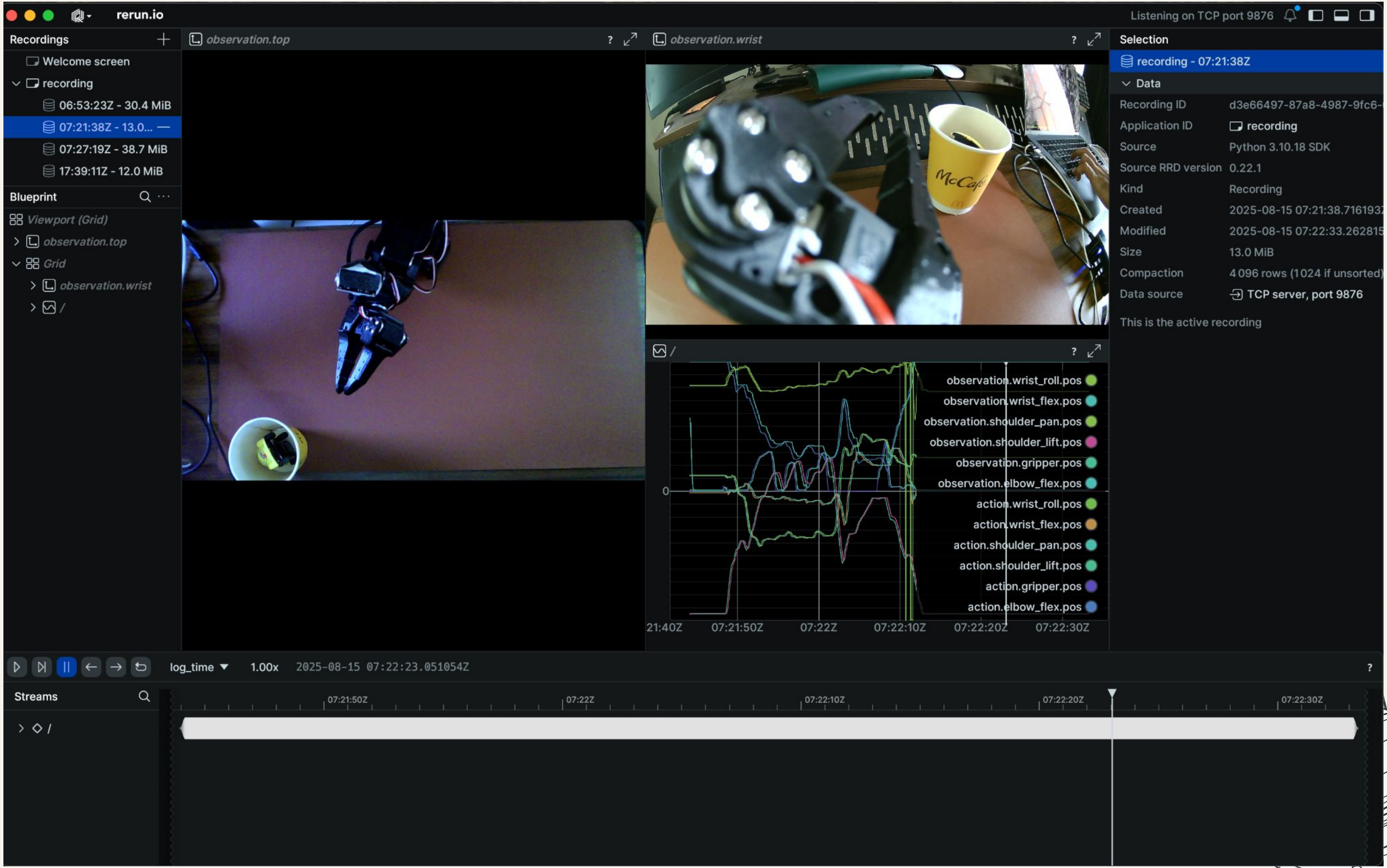
- Map sticks/buttons to joints & gripper
- Haptic & LED feedback for joint limits
- Preset poses & accelerometer control

> Design rule: Match interfaces to your data-collection needs.



Video source : https://www.youtube.com/shorts/zNo3OOIlxPs




rerun.io
Listening on TCP port 9876
Recordings
Welcome screen
recording
06:53:23Z - 30.4 MiB
07:21:38Z - 13.0...
07:27:19Z - 38.7 MiB
17:39:11Z - 12.0 MiB
Blueprint
Viewport (Grid)
observation.top
Grid
observation.wrist
/
observation.top
observation.wrist
McCafé
/
observation.wrist_roll.pos
observation.wrist_flex.pos
observation.shoulder_pan.pos
observation.shoulder_lift.pos
observation.gripper.pos
observation.elbow_flex.pos
action.wrist_roll.pos
action.wrist_flex.pos
action.shoulder_pan.pos
action.shoulder_lift.pos
action.gripper.pos
action.elbow_flex.pos
0
21:40Z
07:21:50Z
07:22Z
07:22:10Z
07:22:20Z
07:22:30Z
Selection
recording - 07:21:38Z
Data
Recording ID
d3e66497-87a8-4987-9fc6-
Application ID
recording
Source
Python 3.10.18 SDK
Source RRD version
0.22.1
Kind
Recording
Created
2025-08-15 07:21:38.716193
Modified
2025-08-15 07:22:33.262815
Size
13.0 MiB
Compaction
4 096 rows (1024 if unsorted)
Data source
TCP server, port 9876
This is the active recording
log_time
1.00x
2025-08-15 07:22:23.051054Z
Streams
/
07:21:50Z
07:22Z
07:22:10Z
07:22:20Z
07:22:30Z

Action Chunking Transformer

```python
[4] from datasets import load_dataset
    import pandas as pd

    # Load the 'lerobot/pusht' dataset
    dataset = load_dataset("lerobot/svla_so101_pickplace", split="train")

    # Convert to pandas DataFrame for easier exploration
    df = pd.DataFrame(dataset)

    # Display the first 5 rows
    display(df.head())
```

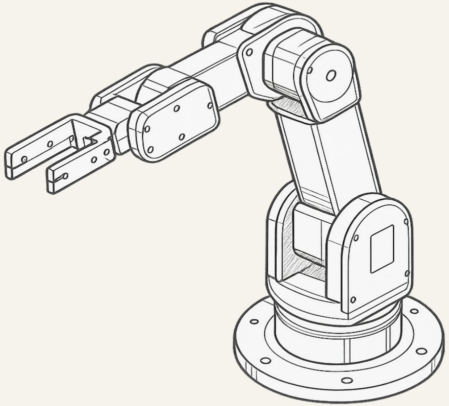
	action	observation.state	timestamp	frame_index	episode_index	index	task_index
0	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.000000	0	0	0	0
1	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.033333	1	0	1	0
2	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.066667	2	0	2	0
3	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.100000	3	0	3	0
4	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.133333	4	0	4	0

```python
df["observation.state"][0]
```

```
[1.9560878276824951,
 -98.74372100830078,
 98.92424774169922,
 74.81983947753906,
 -51.4529914855957,
 1.4093959331512451]
```

```python
df["action"][0]
```

```
[1.9117647409439087,
 -99.41053009033203,
 99.54483032226562,
 74.84062957763672,
 -48.522586822509766,
 1.6228748559951782]
```



https://docs.phospho.ai/learn/policies

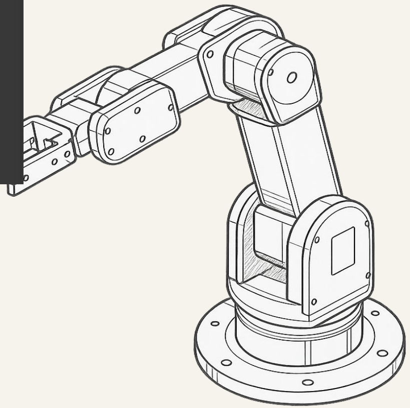
Readings from an episode

```
# first episode

display(df[df.episode_index==0])
```

	action	observation.state	timestamp	frame_index	episode_index	index	task_index
0	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.000000	0	0	0	0
1	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.033333	1	0	1	0
2	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.066667	2	0	2	0
3	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.100000	3	0	3	0
4	[1.9117647409439087, -99.41053009033203, 99.54...	[1.9560878276824951, -98.74372100830078, 98.92...	0.133333	4	0	4	0
...	...	...	...	...	...	...	...
298	[4.485294342041016, -99.41053009033203, 99.453...	[4.5109782218933105, -98.82746887207031, 98.83...	9.933333	298	0	298	0
299	[4.485294342041016, -99.41053009033203, 99.453...	[4.5109782218933105, -98.82746887207031, 98.83...	9.966666	299	0	299	0
300	[4.485294342041016, -99.41053009033203, 99.453...	[4.5109782218933105, -98.82746887207031, 98.83...	10.000000	300	0	300	0
301	[4.485294342041016, -99.41053009033203, 99.453...	[4.5109782218933105, -98.82746887207031, 98.83...	10.033334	301	0	301	0
302	[4.485294342041016, -99.41053009033203, 99.453...	[4.5109782218933105, -98.82746887207031, 98.83...	10.066667	302	0	302	0

303 rows × 7 columns



Total episode duration: 10 secs
Reading frequency : 30 times per sec (30hz)
Total number of readings : 302

Practical Teleop & Dataset Tips

1. Don’t show the leader arm in any training camera view. Otherwise the policy learns to imitate the controller, not the task.

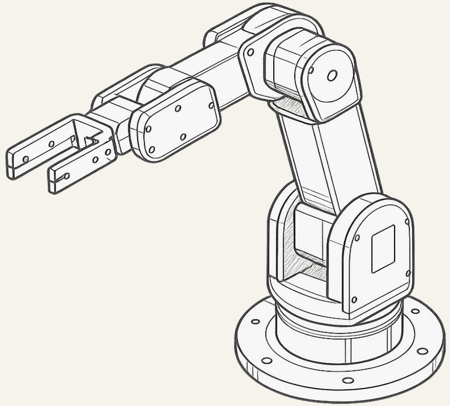
2. Use contrast backgrounds & anti-slip mats to reduce reflection & drift.

3. Diversify positions—cover edges & extremes of the workspace.

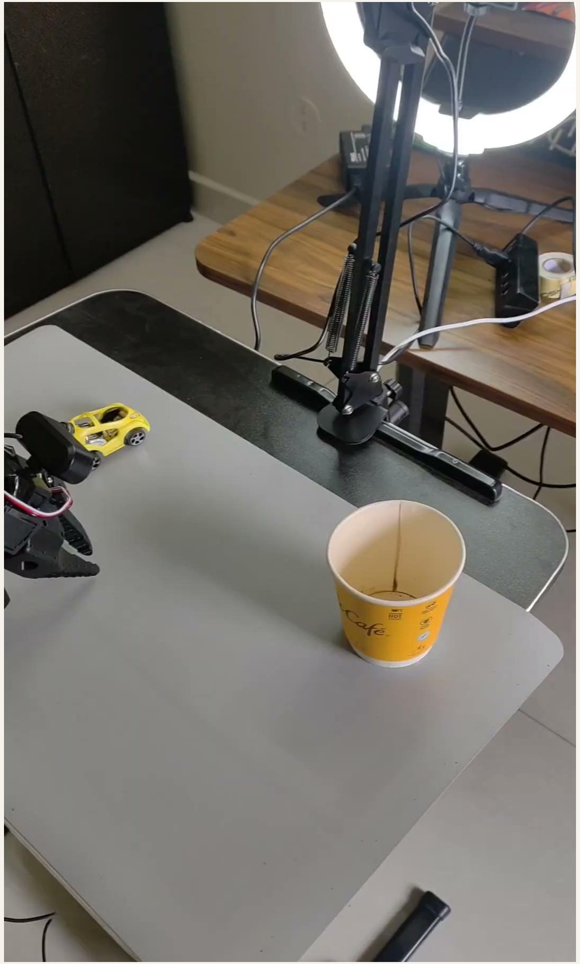
4. 50–60 demos ≈ enough for a simple pick-and-place skill.

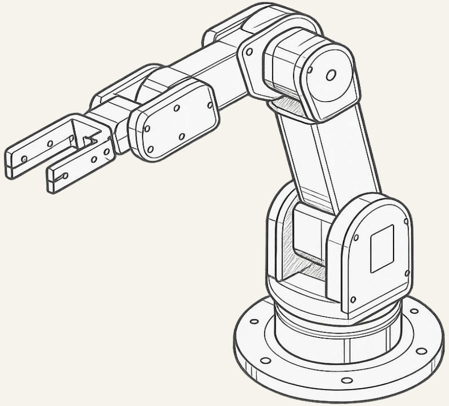
5. Use a wrist camera and train on a stable surface.

6. Good to have a constant and stable source of light.

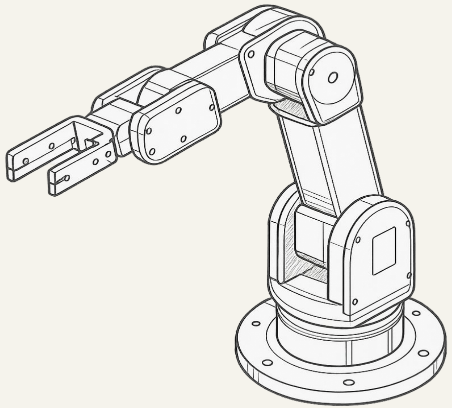


How not to record a dataset

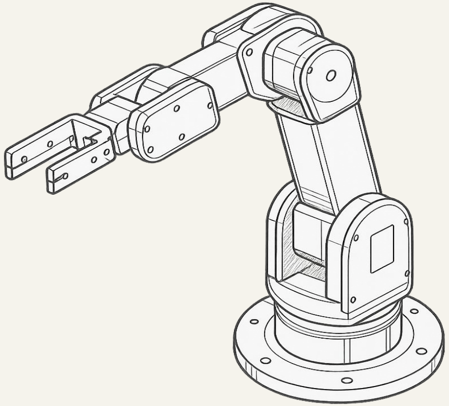




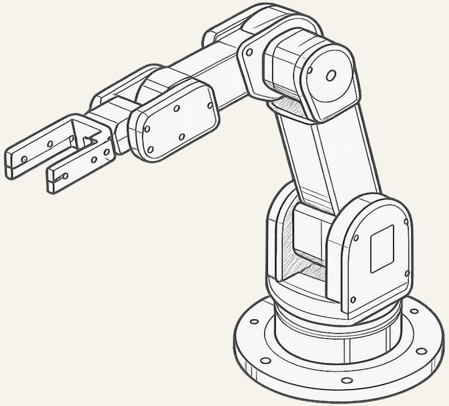
Demo | Recording a dataset



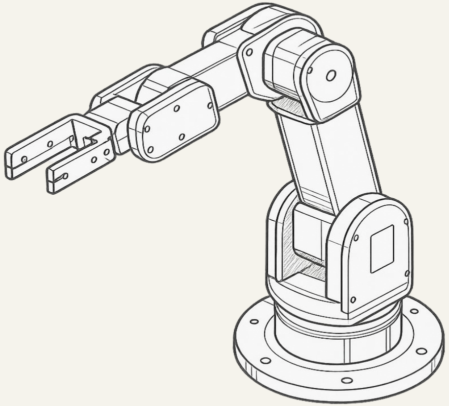
Demo | Training and Inference - trained policy



Demo | Calibrating the Arms



What’s the big deal ?



Thank you

Slides & code:

https://128.pl/1NLbm

Feedback & Questions

Twitter: @logesh_umapathi
Linkedin: www.linkedin.com/in/logeshkumaru/