

ACT vs. SmolVLA: A Controlled Comparison of Task Success, Training Time, and Generalization

Kaixuan Li
Pioneer Research

Abstract

This study compares the practical performance of Action Chunking with Transformers (ACT) and SmolVLA on the same robotic pick-and-place task. ACT was trained as a task-specific policy, while SmolVLA was fine-tuned from its released pre-trained checkpoint. Both models used the same 150-demonstration dataset and the same six-joint SO-101 robotic arm. Synchronized images from a wrist camera and an overhead camera were their only observation inputs; recorded motor states were not provided to either model. The final policies completed 120 autonomous physical trials divided equally across familiar, unfamiliar-object, unfamiliar-position, and unfamiliar-environment conditions. Overall, ACT succeeded in 41 of 60 trials (68.3%), while SmolVLA succeeded in 37 of 60 trials (61.7%). ACT performed better under the familiar and unfamiliar-object conditions, whereas SmolVLA achieved a slightly higher success rate under the unfamiliar-position condition. Both models achieved the same 66.7% success rate under darker environmental lighting. ACT reached its final checkpoint in approximately 12 hours, compared with 14.5 hours of fine-tuning for SmolVLA. However, the cell-level results revealed uneven spatial patterns for both models, showing that overall success rates can hide position-specific strengths and weaknesses. Within the limits of this single-arm and single-task experiment, ACT was more dependable when the task remained close to the training demonstrations, while SmolVLA retained more of its familiar performance under certain spatial and lighting changes. These results suggest that the variety and balance of robotic demonstrations may be as important as the total amount of training data when developing more adaptable embodied intelligence systems.

1. Introduction

Nowadays, the artificial intelligence people come into contact with exists on screens and in virtual reality: it is capable of generating images, forming texts, analyzing information, and even competing in games with real humans. However, when we first designed robots and artificial intelligence, most of the problems we intended to conquer lay in the physical world: repetitive work, dangerous conditions, and situations that are too dangerous or impossible for humans to handle. From small things like doing chores to matters of greater importance like rescue missions and even space exploration, these are the things that cannot be dealt with using the current intelligence of software alone. A robot must be able to feel its surrounding environment, make decisions, and execute those decisions with a physical body. This combination of artificial intelligence and robotics is often called embodied intelligence. It matters because it allows machines to expand human abilities beyond constraints like strength, endurance, safety, and accessibility.

Yet, intelligence and physical components are often not completely connected. Most robots we encounter in real life are hard-coded machines that might be able to act flawlessly in a carefully controlled environment where all their tasks are within the expected range of their written code but might fail completely when facing the slightest change in conditions or tasks. Such a system is not the connection between intelligence and physical components but is more like a sophisticated puppet manipulated through strings of code.

I have a great interest in this problem because I do not want to make a robot that simply repeats its pre-programmed steps. I want to understand how the model observes its own action through its cameras, reflects on itself, and makes changes to its next-step actions. Computer vision is the first step toward achieving this goal: if robots want to act individually in the real world, they must understand and observe their own surroundings and what changes their actions bring.

With this idea in mind, I prepared a set of robotic arms and cameras as shown in figure 0 and figure 1. These devices provided me with an efficient and practical platform for investigating this connection between code, vision, and action.

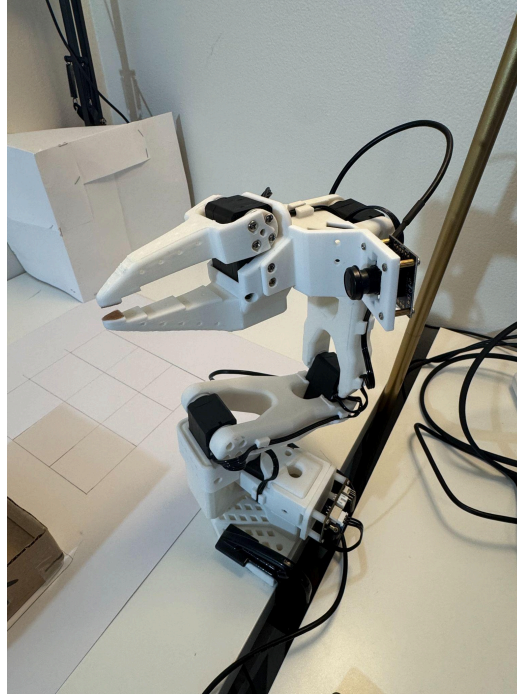


Figure 0. Robotic Arm Platform SO-101 Follower Arm

This project compares the output of two models to learning physical intelligence behaviors. Action Chunking with Transformers (ACT) is an imitation-learning policy that predicts sequences, or chunks, of actions from robot observations (Zhao et al., 2023). SmolVLA is a compact vision-language-action model designed to connect visual observations and language instructions with robot actions while remaining practical on relatively accessible hardware (Shukor et al., 2025). These two models use different methods to treat the same core problem: how can robots learn from demonstrations provided by humans? By putting them onto the same robotic arm and giving them an identical local demonstration dataset, I am able to compare their practical performance, training cost, and generalization capability.

Research Question:

When ACT is trained from scratch and pre-trained SmolVLA is fine-tuned using the same local robotic demonstration dataset, how do they differ in task success rate, training time, and ability to generalize to unfamiliar objects, positions, and environments on the same robotic arm?

To investigate this problem, I designed a physical task that allows me to reach a clear conclusion of success or failure, such as letting the robotic arm pick up an object and move it to the pre-determined destination area. I used my own robotic arms and cameras to record the human demonstrations (aka the dataset). ACT will be trained from scratch on this dataset, as it usually is trained with no previous checkpoints, while SmolVLA will be initialized from the released pre-trained checkpoint, since a base is needed for the model to comprehend tasks with a limited local dataset, and fine-tuned on the same set of demonstrations. The robotic arm, camera settings, task details, computer used for training, and local dataset will remain unchanged throughout the process. However, the initialization of the models will be different, so this

experiment will compare the two models in their selected practical configurations rather than isolate model architecture as the only variable. I will record ACT’s full training time and SmolVLA’s fine-tuning time separately.

The evaluation includes result outputs under conditions familiar to the model and result outputs under unfamiliar conditions. I will first test the models with items, positions, and visual environments that exist within the dataset, then change one variable at a time: different items, changes in initial and ending positions, or alterations to the background or lighting. Before the official testing, I will set up an objective standard for success and a time limit for each attempt to avoid non-intentional successes. The task success rate will be calculated as the number of successful trials divided by the total number of trials. I will have three different sets of reports for different items, positions, and background environments. I will also evaluate the failed attempts, such as being unable to pick up or locate objects. Doing so allows me to make better cross-comparisons between the two models and better explain the difference in performance between them.

2. Background

SmolVLA

SmolVLA is a compact, open-source Vision-Language-Action model, mainly used for robotic control (Aubakirova et al., 2025). VLA models can connect the visual data from the robotic component with instructions about how it should execute those instructions. SmolVLA receives images from multiple cameras, a human instruction, and information about the robot’s current physical state. Its vision-language component processes this information, while its action part predicts a continuous chunk of future robot actions. A chunk is a sequence of several future actions predicted together instead of one action at a time. In my configuration, SmolVLA generated 50 actions in each chunk. Each action contained the six commands needed to control the arm’s joints and gripper. Because I used synchronous inference, the robot first received the current camera images, generated and executed the chunk, and then observed the scene again when another chunk was required. It did not reconsider every action from a new camera frame while the current chunk was being executed. SmolVLA contains approximately 450 million parameters, making it small enough to train and run on consumer hardware. I also kept the text instruction exactly the same from one step to the next instead of varying its wording.

The SmolVLA developers had already pre-trained the released `smolvla_base` checkpoint on public, community-collected robotics datasets. In this experiment, I loaded that released checkpoint and fine-tuned it using my own 150 robotic demonstrations. Although SmolVLA is compact compared with many other VLA models, its approximately 450 million parameters were still too large for practical training from scratch on my 150-demonstration dataset. Fine-tuning the released checkpoint was therefore the more practical choice (Shukor et al., 2025; Aubakirova et al., 2025).

Action Chunking with Transformers (ACT)

Action Chunking with Transformers, also called ACT, is an imitation-learning model that learns robotic tasks from human demonstrations. In this experiment, ACT received synchronized images from the wrist and overhead cameras. A ResNet-18 backbone first converted these images into visual features. During training, a Transformer encoder learned a latent representation of the demonstrated movements, while a Transformer decoder used the visual features and this representation to predict a sequence, or chunk, of future actions. Although ACT can also use current joint positions as observations, they were not provided to the model in this experiment. The demonstrated motor commands were used only as the actions that the model learned to predict, not as observation inputs during training or inference (Zhao et al., 2023).

ACT and SmolVLA both predict multiple future actions at once, but they use different architectures and chunk settings. My ACT configuration predicted 100 actions per chunk, while SmolVLA predicted 50. Each ACT action contained commands for the arm's six motors. ACT also supports temporal ensembling, which combines overlapping predictions to produce smoother movements, but this feature was disabled in my experiment. I did not load a previously trained ACT manipulation policy. Instead, I followed ACT's common task-specific training setup and trained the policy using my local demonstrations. However, the ResNet-18 visual backbone still began with ImageNet weights. Therefore, when this paper describes ACT as trained from scratch, it means that no previously trained ACT policy was used, not that every part of the model began without prior weights.

3. Related Work

Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware

Zhao et al. (2023) introduced Action Chunking with Transformers, or ACT. Instead of predicting one movement at a time, ACT predicts an entire set of movements. The model uses temporal ensembling to combine overlapping predictions. By doing so, the robot's movement becomes much smoother, and unwanted errors from individual predictions are reduced. The researchers used a low-cost ALOHA robotic arm to test the ACT model on difficult manipulation tasks. The system reached a success rate of 80% to 90% using only 10 minutes of demonstration data (Zhao et al., 2023).

ACT was one of the models I compared in my experiment. Therefore, this paper was the main source of introduction to how ACT works. My experiment used a SO-101 single-arm robot, and I tested it on simpler pick-and-place tasks, so the success rate from the original paper cannot be compared with my own results. However, its methods help me understand how action chunks, visual feedback, and demonstration data affect ACT.

What Matters in Learning from Offline Human Demonstrations for Robot Manipulation

Mandlekar et al. (2022) researched six types of offline learning algorithms on five simulated tasks and three real-world robotic tasks. They used different qualities of demonstration

data and found out that the model’s performance is heavily affected by the quality of the demonstrations in the dataset, the selected models, and the rules used to stop training. Their results show that having a large dataset does not automatically guarantee accurate results from the models. If a demonstration in the dataset includes inconsistent, inaccurate, or failed actions, the model’s learning capability will be affected (Mandlekar et al., 2022).

This paper helped me to make clear standards and guidelines for data collection before I began running the human demonstrations. Every accepted demonstration should complete the task correctly within a time limit, avoid collisions or human correction, and contain clear camera footage. Both models will use the exact same local task-specific dataset, although SmolVLA will also begin with knowledge from its pre-trained checkpoint.

Data Scaling Laws in Imitation Learning for Robotic Manipulation

Lin et al. (2025) researched how the quantity and diversity of demonstrations in the dataset affect the model’s generalization capability. The researchers collected over 40,000 demonstrations and completed over 15,000 real-world robot tests. By doing so, they found out that increasing the variety of the training objects and background environments enhances the model’s generalization capability much better than collecting demonstrations with the exact same conditions every time. Their results suggest that dataset diversity is especially important when the robot is working with unfamiliar objects or unfamiliar environments (Lin et al., 2025).

Since my dataset contains only 150 samples, I did not attempt a large-scale experiment like the one shown in this paper. However, this research helps to explain why the two models may struggle outside the training distribution. It also supports the use of unfamiliar objects, different positions, and different environments as three separate conditions. Both models will receive the same local dataset, but SmolVLA will also benefit from its pre-training. Therefore, differences in the results cannot be attributed only to model structure. They may reflect the combined effects of architecture, initialization, and prior knowledge.

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

Zitkovich et al. (2023) introduced RT-2, one of the early vision-language-action models. RT-2 combines robotic demonstrations with knowledge from large-scale vision-language training. RT-2 represents robot actions as tokens, allowing vision information, language, and actions to be processed in the same format. Across approximately 6,000 trials, RT-2 demonstrated the capability to process unfamiliar objects, instructions, and simple reasoning tasks (Zitkovich et al., 2023).

RT-2 helped me to understand why VLA models might demonstrate different generalization capabilities than ACT. However, RT-2 used much more data and pre-training resources than the pre-trained SmolVLA used in my experiment, so the success rate presented in the text cannot be used as a direct comparison. I used this paper to explain the general ideas behind VLA models and why visual and language knowledge may help a robot to deal with situations not covered in the local training data.

OpenVLA: An Open-Source Vision-Language-Action Model

Kim et al. (2025) invented OpenVLA, a Vision-Language-Action model that has 7 billion parameters. It was trained using about 970,000 real-world demonstrations. OpenVLA combined a Llama 2 language model and the visual features from DINOv2 and SigLIP. Kim et al. used new backgrounds, item positions, item appearances, and human instructions to test the model and demonstrated how the model can be fine-tuned to fit on new robots and tasks (Kim et al., 2025).

Although I did not use OpenVLA, it helped me to understand Vision-Language-Action models better. The research paper separates generalization capability into vision, motion, physical, and semantic categories. This categorization helped clarify the generalization capability section of my research.

Diffusion-VLA: Generalizable and Interpretable Robot Foundation Model via Self-Generated Reasoning

Wen et al. (2025) introduced DiffusionVLA to the public. This model combines models based on language reasoning and models that are diffusion-based. The language portion explains what the robot should do, while the diffusion sector is responsible for generating continued movement. The researchers tested the model on object sorting, bin picking, and other real-world tasks. The tests also included unfamiliar items, new backgrounds, obstacle objects, and different lighting. In an experiment, the model successfully picked up 63.7% of 102 objects that were not used in the training phase (Wen et al., 2025).

This research paper demonstrated another method that combines computer vision, language models, and robotic movement. Because the research lasted for five weeks and that training three models (the other two being ACT and SmolVLA) would have been too difficult, I did not test DiffusionVLA. However, its experimental results supported my decision to use different items and changing background environments to test the generalization capability of the models. It also showed how pre-trained visual understanding might help the robot to process and deal with situations that are beyond its local dataset.

SmolVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics

Shukor et al. (2025) introduced SmolVLA. It is a model with 450 million parameters that is designed to run on hardware that is affordable to most. It utilizes camera vision, language commands, and the robot's current state to predict a set of actions. In the paper, SmolVLA was pre-trained with around 22,900 robot demonstrations. The model is smaller than many other vision-language-action models and can run on consumer hardware. The paper also compares pre-trained SmolVLA with ACT in simulation and on real SO100 and SO101 robotic arms (Shukor et al., 2025).

This paper is an important source for my project because it directly compares SmolVLA and ACT. In the SO101 task, SmolVLA's success rates in familiar and unfamiliar positions were 90% and 50%, while ACT's were 70% and 40%. Similar to the paper's experiment, my SmolVLA began from a pre-trained checkpoint while ACT was trained from scratch. Both models used the exact same local task-specific dataset, robotic arm, camera, and task. I recorded ACT's full training time and SmolVLA's fine-tuning time, and I stated that differences in performance may result from both model architecture and pre-training.

SmolVLA: Efficient Vision-Language-Action Model Trained on LeRobot Community Data

Aubakirova et al. (2025) explained SmolVLA in an official Hugging Face article. The article explained the model’s main structure and provided training commands. It also included a guide explaining how to fine-tune the released “smolvla base” and how to train the model from scratch. It also discussed asynchronous inference, which allows the robot to predict future actions while it is still moving (Aubakirova et al., 2025).

I used this article to load the released pre-trained “smolvla base” checkpoint and fine-tune it using my own dataset. I made this change because my early tests showed that SmolVLA performed too poorly when trained without the pre-trained checkpoint to support a useful comparison within the five-week project. I recorded the fine-tuning steps, hardware, checkpoint, and total fine-tuning time. I also used the same SmolVLA inference mode during every test. Since the article was written by the SmolVLA team and was not peer-reviewed, it is useful for setup instructions but is not an independent evaluation of the model.

Imitation Learning on Real-World Robots

The LeRobot guide explained how to train a physical robot through imitation learning. It covers calibration, teleoperation, camera setup, data recording, dataset visualization, model training, and evaluation. The recorded demonstrations will be saved using the standard “LeRobotDataset” format. The guide also provided instructions for training the ACT model and saving model checkpoints (Hugging Face, n.d.).

I used LeRobot as the shared workflow for ACT and SmolVLA. Both models used the same robot calibration, cameras, image settings, task instructions, and recorded local dataset. ACT began untrained, while SmolVLA began from the released pre-trained checkpoint. I saved the commands, configurations, starting and final checkpoints, ACT training time, and SmolVLA fine-tuning time. This reduced unwanted or accidental differences between the experiments while keeping the different initialization conditions transparent and repeatable.

4. Controlled ACT-SmolVLA Comparison

4.1 Overall Research Design

This research used a controlled robotic arm to compare two working models on the same pick-and-place task. ACT and SmolVLA used the same 150-demonstration dataset, SO-101 follower arm, two cameras, destination box, workspace, training and inference computers, task definition, and success standard. The main planned difference was the model itself and its initialization: ACT used a task-specific policy without a previously trained ACT manipulation checkpoint, while SmolVLA was fine-tuned from the released lerobot/smolvla_base checkpoint. The comparison therefore focused on the two systems in their practical configurations rather than trying to isolate architecture alone.

Each autonomous physical rollout was treated as one trial. The two experimental factors were the model and the evaluation condition: Familiar, Unfamiliar Object, Unfamiliar Position, and Unfamiliar Environment. The primary outcome was binary success or failure, and training or fine-tuning time was recorded separately as a measure of computational cost. Generalization was measured as each model’s change in success rate from its own familiar condition.

The official experiment contained 120 trials: 15 trials for each model under each of the four conditions, or 60 trials per model. Practice rollouts were completed first to confirm that each policy and the robotic arm were functional, and those practice results were excluded from the official totals.

4.2 Robot, Cameras, and Workspace

The experiment platform is a six-joint SO-101 follower robotic arm. The demonstration data are collected through the teleoperation between the SO-101 leader arm and the SO-101 follower arm. The follower arm’s six recorded state and action dimensions corresponded to shoulder pan, shoulder lift, elbow flexion, wrist flexion, wrist rotation, and gripper position. Before the data collection, both the leader and follower arms were calibrated through LeRobot’s standard calibration procedure. No additional calibration was performed during the official evaluation.

The entire project utilizes two RGB cameras. One hand-eye camera is installed on the left side of the follower arm’s wrist, and a fixed camera provides the view of the workspace from an overhead-front position. Both streams were captured at 640 x 480 pixels and 30 frames per second. The cameras were named “observation.images.handeye” and “observation.images.front” in the dataset and model configurations. These camera settings remained unchanged throughout data collection and all 120 official trials.

The workspace is constructed with a white surface, destination box, and a painted 3 x 3 starting grid. The grid is 7 x 7 inches (17.8 x 17.8 cm), making each square approximately 2.33 inches (5.9 cm) wide. The box footprint measured approximately 6.4 x 5.8 inches (16.3 x 14.7 cm) and remained in a fixed location to the left of the grid. Three lamps surround the workspace as constant light sources. These lamps remained in the familiar, unfamiliar-object, and unfamiliar-position conditions and were switched off only in the unfamiliar-environment condition.

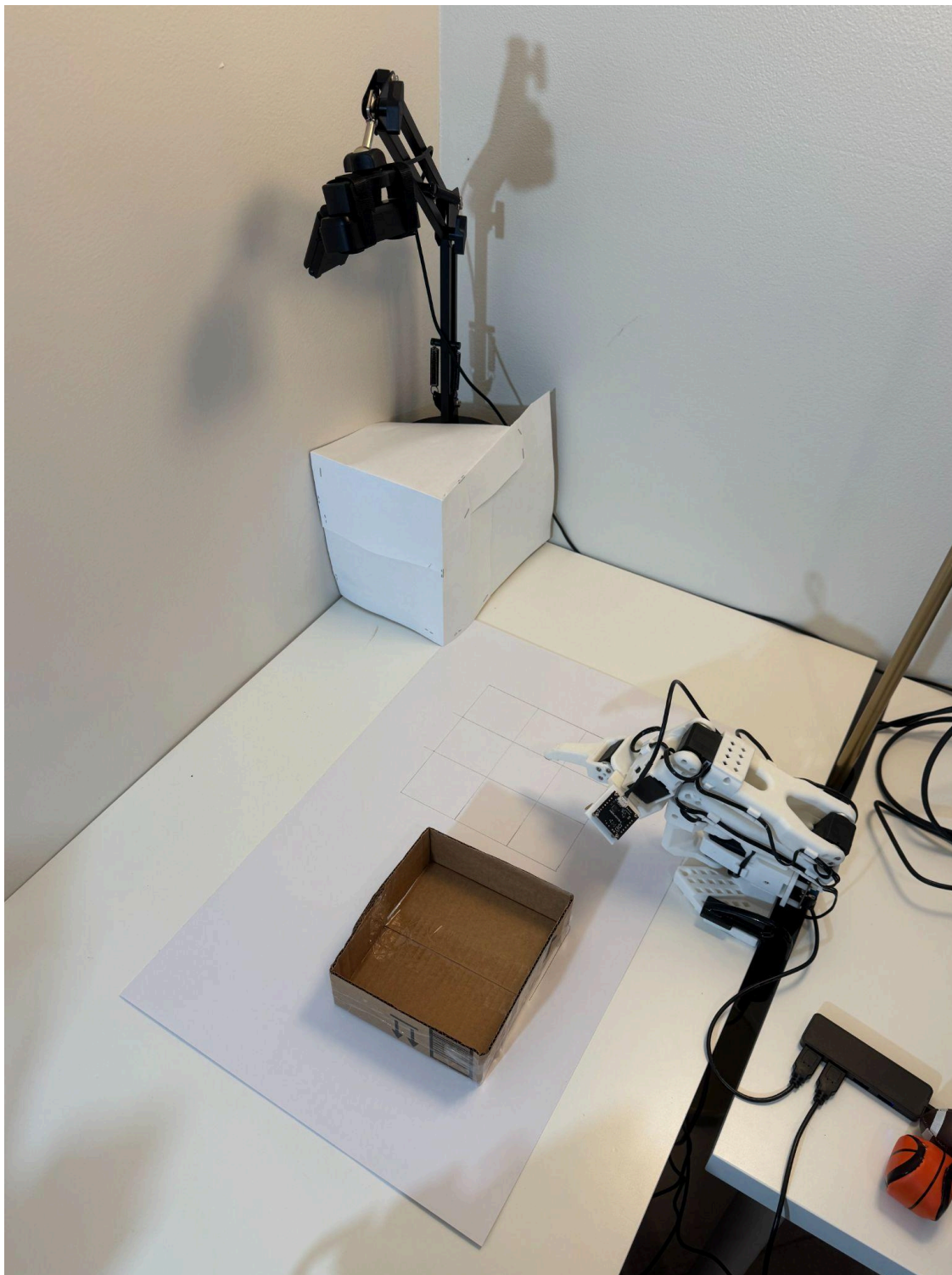


Figure 1. Physical evaluation workspace showing the SO-101 follower arm, fixed camera on the top of the picture, destination box, and 3×3 grid.



Figure 2. Physical evaluation workspace showing the position of the three lamps

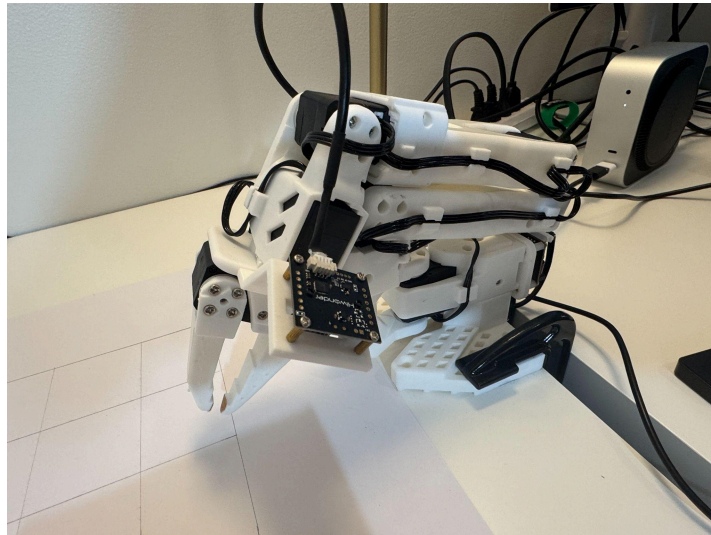


Figure 3. SO-101 follower arm and wrist-mounted hand-eye camera.

4.3 Manipulation Task and Outcome Definition

For every demonstration and official evaluation, the robotic arm's task is to pick up an

object from the 3×3 grid and place it completely inside the destination box. According to the arm's direction when in default position, the grid is labeled from left to right and top to bottom, with the first row being 1 to 3, the second row being 4 to 6, and the third row being 7 to 9. Since Positions 3, 4, 6, and 7 exist within the local demonstration dataset, they are defined as familiar positions, while Positions 1, 2, 5, 8, and 9 are defined as unfamiliar positions.

The official experiment begins when the autonomous model execution starts and ends when the robotic arm places the object within the destination box, or the timer exceeds 45 seconds. The trial will be marked as a success only when the object enters the destination box within 45 seconds. Any other outcome will be marked as a failure, including the object being moved outside the 3×3 grid, the need for human assistance, or the timer passing 45 seconds. If the object only touches the destination box or has parts that remain outside the box, the trial will not be counted as a success until the object is fully within the box before the timer reaches 45 seconds. The researcher observes each trial in person and records the binary results immediately after each trial.

The destination box's position is fixed, but the orientation of the object is not fixed. The robotic arm is not always fully restored back to the exact starting position before each trial. Instead, the evaluation program's return-to-start behavior was allowed to move the arm back toward its initial position, and the next trial began after the arm had returned to an approximately consistent pose. This procedure was intended mainly to avoid manual repositioning during autonomous trials.

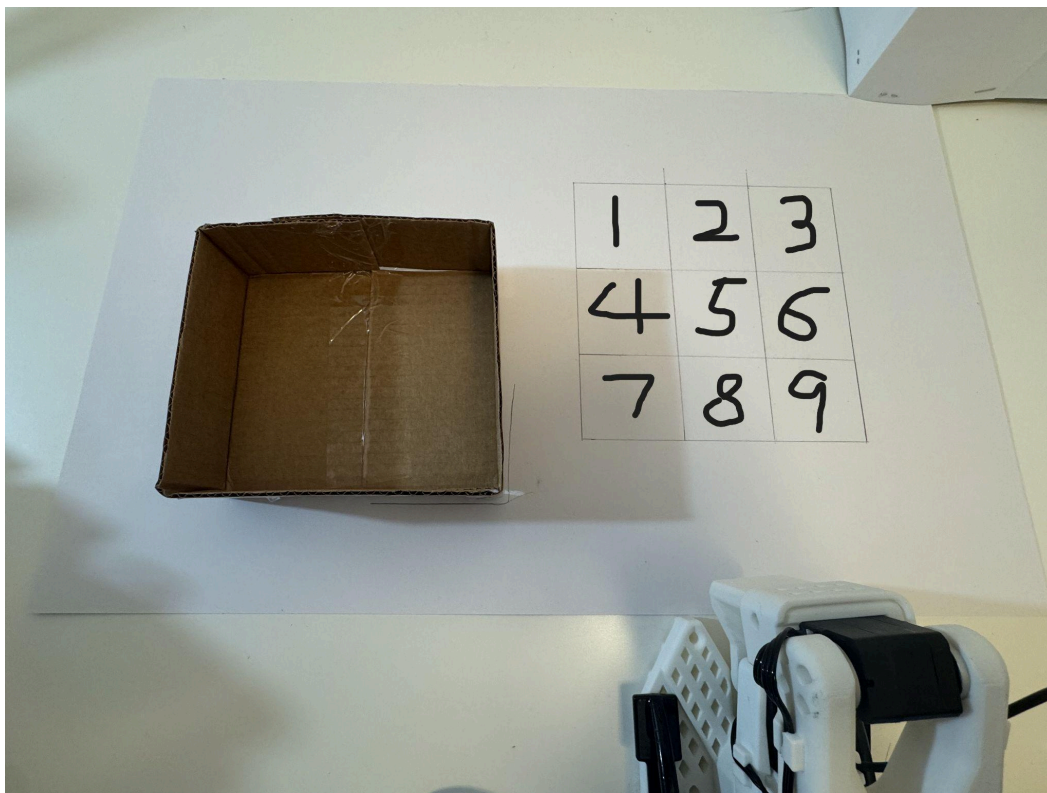


Figure 4. Numbering convention for the 3×3 starting grids.

4.4 Demonstration Dataset

The local dataset was recorded personally by the researcher through SO-101 leader-follower teleoperation and stored in the LeRobotDataset format. The final repository [ted88168/demov0805_merged](https://github.com/leobot/demos0805_merged) contains 150 complete demonstration episodes, 88,162 frames, two synchronized camera streams, six-dimensional robot states, and six-dimensional actions. All episodes were stored within a single training split covering episodes 0-149. No separate validation splits or filters for episodes were used. At 30 frames per second, the dataset represented approximately 49 minutes of robotic interaction in total. Because the local dataset contained only 150 episodes and I had decided in advance to evaluate the final 100,000-step checkpoint instead of choosing a checkpoint from validation results, I used all episodes for training. This gave both models the full local dataset, although it also meant that no held-out validation performance was measured.

The demonstration data use the three familiar objects shown in Figure 7: a small orange-black soft basketball, a small deep-brown soft American football with white stripes, and a small, hard 3 x 3 Rubik’s cube. The object’s initial position is randomly selected from familiar Positions 3, 4, 6, and 7. The destination box, cameras, work surface, and normal three-lamp illumination were kept consistent during collection. Object orientation was allowed to vary. When a teleoperation attempt does not correctly complete the task, the demonstration is removed from the dataset and re-recorded.

The dataset contained only one language task label, recorded exactly as “Grab the lego.” I kept this wording for every object because earlier practice tests showed that it produced the most reliable behavior, and changing the instruction would have introduced another variable. The text was used as SmolVLA’s command input, while ACT did not receive a natural-language instruction. Because the wording never changed, this experiment did not test language generalization. Both models used the same two camera streams and demonstrated action targets from the complete repository. The recorded motor states remained available as feedback in the dataset but were not used as observation inputs by either model. The full set of 150 episodes was used for both policies.

4.5 Policy Training and Checkpoint Selection

Both models were trained on the same Windows 11 computer using an NVIDIA GeForce RTX 3070 Ti Laptop GPU with 8 GB of video memory. The software environment utilized Python 3.12.13, LeRobot 0.5.2, PyTorch 2.7.1 with CUDA 11.8, and cuDNN 9.1. Both training configurations used a batch size of 16, a random seed of 1000, 100,000 optimization steps, and checkpoints saved every 20,000 steps. Image augmentation was disabled. The final 100,000-step checkpoint was used for evaluation for both models. Checkpoints were not selected after observing the official trial results.

Table 0. ACT and SmolVLA Training Configurations

Configuration	ACT	SmolVLA
---------------	-----	---------

Initialization	No prior ACT policy checkpoint	Released lerobot/smolvla_base checkpoint
Local Dataset	ted88168/demov0805_merged	ted88168/demov0805_merged
Training Steps	100,000	100,000
Batch Size and Random Seed	16 and 1000	16 and 1000
Optimizer	AdamW	AdamW
Learning Rate	1×10^{-5} , no scheduler	Peak 1×10^{-4} , cosine decay
Weight Decay	1×10^{-4}	1×10^{-10}
Action Chunk	100	50
Approximate Training or Fine-Tuning Time	12 hours	14.5 hours

- #### 4.5.1 ACT Configuration

The ACT robot-control policy was trained without loading a previously trained ACT policy checkpoint. Its configured visual encoder was a ResNet-18 initialized with ImageNet weights; therefore, trained from scratch in this study referred specifically to the ACT manipulation policy and did not mean that every visual component began from random weights. ACT received the two 640 x 480 RGB views and the current robot state. Mean-and-standard-deviation normalization was applied to images, states, and actions. The policy used a variational autoencoder with a latent dimension of 32 and a Kullback–Leibler weight of 10. The configured action chunk and number of executed action steps were both 100. Temporal ensembling was disabled. Optimization used AdamW with a learning rate of 1×10^{-5} , weight decay of 1×10^{-4} , and no learning-rate scheduler.

- #### 4.5.2 SmolVLA Configuration

SmolVLA was initialized from the released lerobot/smolvla_base checkpoint and fine-tuned using the same 150 episodes. Its vision-language backbone was HuggingFaceTB/SmolVLM2-500M-Video-Instruct. The raw 640 x 480 camera streams were resized with padding during preprocessing to the model’s 512 x 512 visual-processing size. Image normalization was configured as identity, while robot states and actions used mean-and-standard deviation normalization. The vision encoder was frozen; expert and action-related components and the state projection were trained, and parameter-efficient fine-tuning was not used. The action chunk and number of action steps were both 50, the flow-matching generation process used 10 steps, and key–value caching was enabled. Optimization used AdamW with a peak learning rate of 1×10^{-4} , 1,000 warm-up steps, 30,000 cosine-decay steps, a final learning rate of 2.5×10^{-6} , and

weight decay of 1×10^{-10} .

4.6 Inference Platform and Rollout Configuration

All official physical rollouts were executed on the same Mac mini rather than on the training computer. The computer used for inference was a Mac16,10 Mac mini with an Apple M4 processor, a 10-core CPU, 24 GB of unified memory, and an Apple GPU accessed through PyTorch’s Metal Performance Shaders backend. It ran macOS 26.5.2, Python 3.13.13, and LeRobot 0.5.2, with the two cameras providing 640 x 480 RGB frames at a nominal 30 frames per second.

The final 100,000-step checkpoints for ACT and smolvla_base_so101_0805_merged were loaded for all the official trials. Both models used normal or synchronous inference, not asynchronous inference. In synchronous operation, the system captured the current images, preprocessed the inputs, generated an action chunk, transferred the processed actions to the chip on the SO-101 robotic arm, executed that chunk, and then generated another chunk when required. The target control rate was 30 Hz, although the actual loop timing also depended on model computation, preprocessing, device transfer, and serial communication. Both models’ inference settings remained unchanged across their 60 official trials.

4.7 Evaluation Conditions and Trial Allocation

The experiment includes one familiar condition and three forms of distribution shift. Only the targeted factor is changed in each unfamiliar condition. Whenever a trial uses a familiar object, a random object is selected from basketball, American football, and Rubik’s cube. The starting position is also randomly selected from the permitted positions under that specific condition. Random selection is performed individually for each trial.

Table 1. Evaluation Conditions and Intended Changes

Condition	Object Set	Allowed Positions	Lighting	Intended Change
Familiar	Random Familiar Object	3, 4, 6, or 7	Three workspace lamps on	None. Trained condition
Unfamiliar Object	Purple Rigid Block	3, 4, 6, or 7	Three workspace lamps on	Object identity, shape, color, and rigidity
Unfamiliar Position	Random Familiar Object	1, 2, 5, 8, or 9	Three workspace lamps on	Starting position
Unfamiliar Environment	Random Familiar Object	3, 4, 6, or 7	Three workspace lamps off	Visual illumination

The unfamiliar object is a purple toy block with a rectangular or cubic base and a large cylindrical stud on top. It did not appear in the local demonstration dataset. In the unfamiliar position condition, the object, box, camera, workspace, and light source remain unchanged. For

the unfamiliar environment condition, the three lamps in Figure 2 were turned off. Other conditions remain unchanged. Before each trial, a random-number generator selected the object, permitted starting grid position, and object orientation. The destination box remained fixed.

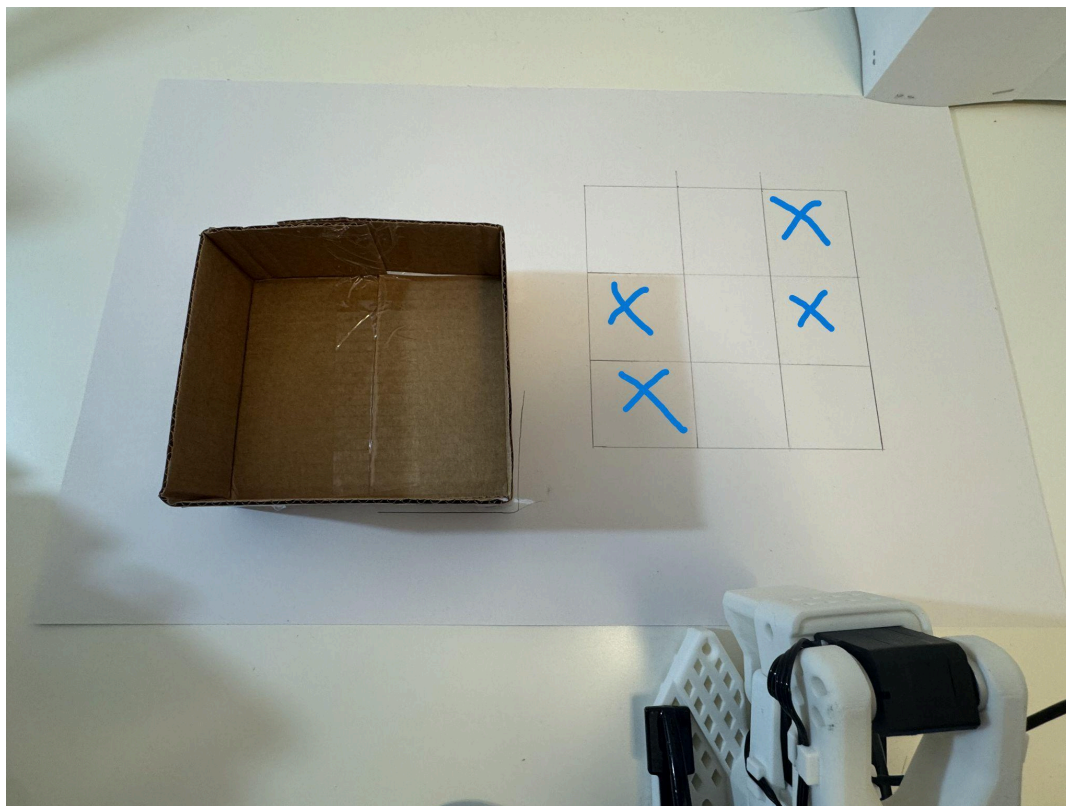


Figure 5. Familiar starting Positions 3, 4, 6, and 7.

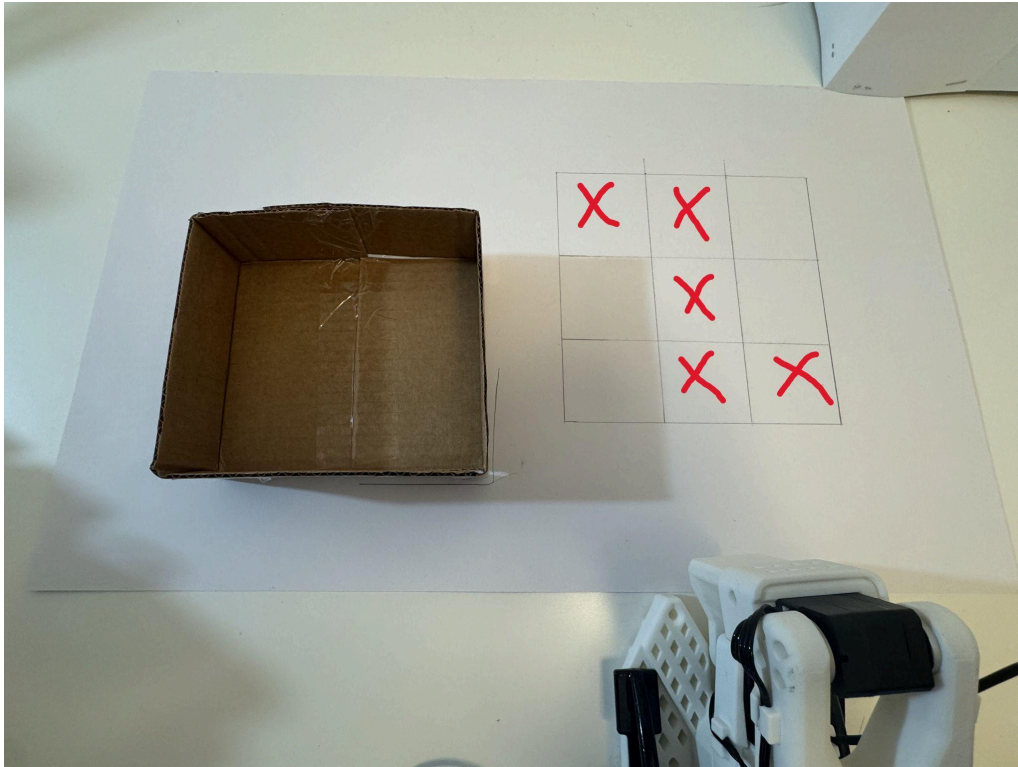


Figure 6. Unfamiliar starting Positions 1, 2, 5, 8, and 9.



Figure 7. Familiar object set used during demonstration collection and applicable evaluation conditions.

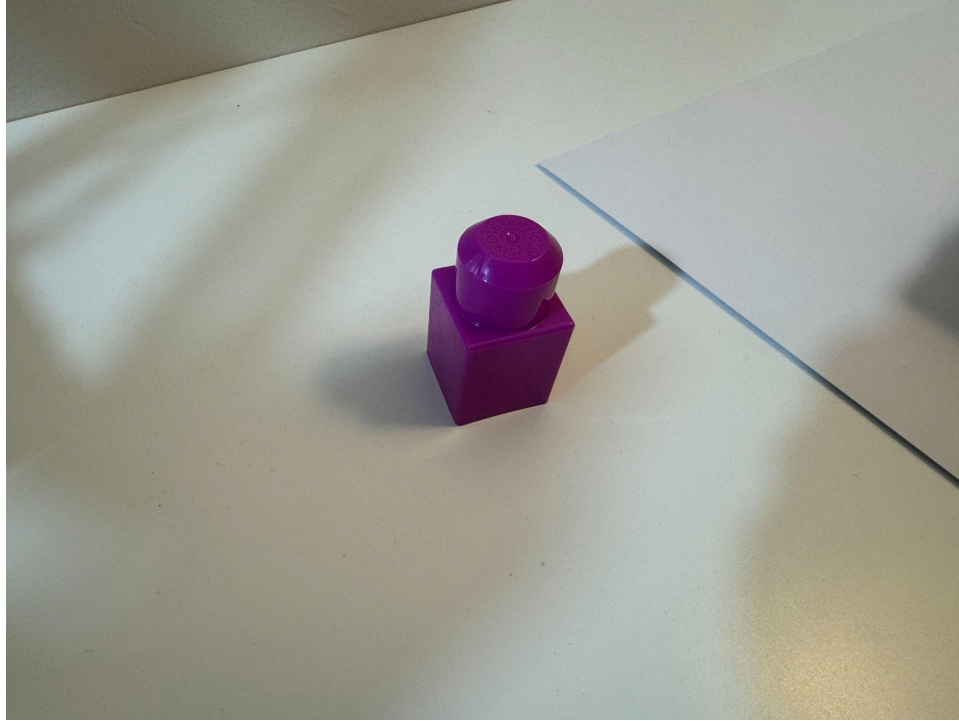


Figure 8. Purple rigid block used only in the unfamiliar-object condition.

The experiments were conducted in blocks instead of alternating. All 60 ACT trials were completed first, followed by 60 SmolVLA trials.

4.8 Trial Procedure and Data Recording

Before each trial, the object from the previous trial was removed, and the researcher waited for the robotic arm's autonomous return-to-start program to move it back toward its approximate starting pose. The researcher then placed the selected object on the selected grid cell with a varied orientation, while the destination box remained fixed. The appropriate final checkpoint remained loaded, and synchronous autonomous execution began. No human correction was applied while either model controlled the arm during any official attempt. The complete sequence used for both models is shown in Figure 9.

The reproducible trial sequence was:

```

for model in [ACT, SmoVLA]:
    load the model's final 100,000-step checkpoint
    for condition in [familiar, unfamiliar object, unfamiliar position, unfamiliar
environment]:
        repeat 15 times:
            randomly select an allowed object
            randomly select an allowed grid position
            reset the box/object scene
            wait for the arm to return approximately to its start pose
            place the object in the selected grid
            begin synchronous autonomous execution and the 45-second limit
            if (the object is fully inside the box within 45 seconds):
                record success = 1
            else:
                record success = 0
            record the trial

```

Figure 9. Pseudocode of the reproducible trial sequence

All 120 official trials were recorded without any replacement, re-recording, or deletion. During SmoVLA’s unfamiliar environment condition trials, ENV-22 and ENV-28 experienced temporary disconnection due to component overheating after both the attempted motion and the outcome assessment had been completed. Their original outcomes were recorded and kept for that reason. The trials were not repeated, and the robotic arm was restarted before the evaluation continued. The complete set of recorded trial videos is available at [ted88168/datasets](https://ted88168.com/datasets).

4.9 Outcome Measures and Analysis

For each model-condition pair, task success rate was calculated as

Success Rate (%) = Number of Success Trials / 15 x 100

Results were recorded both as the number of successful trials out of the total number of trials and as percentages. Performance under familiar conditions was used as the control group for both models. Every unfamiliar condition’s change in success rate was used as the primary variable to evaluate the generalization capability of the models under that specific condition. The equation for evaluation was “Success rate under unfamiliar condition - Success rate under familiar condition.” This calculation made it possible to distinguish absolute unfamiliar-condition performance from the size of the decline relative to each policy’s own control-group success rate. A positive value would mean that the model performed better under the unfamiliar condition, while a negative value would mean that it performed worse; the size of the value shows the size of the change.

This experiment’s analysis will be descriptive because the research represents one robot, one dataset, one trained checkpoint per model, and 15 repeated physical trials per

model-condition cell. Approximate training time was reported separately for the two training procedures. Because ACT and SmolVLA differed in initialization, architecture, preprocessing, action-chunk configuration, and optimization procedure, observed performance differences are interpreted as differences between the two selected practical systems instead of as the isolated effect of architecture alone.

5. Comparative Analysis

5.1 Quantitative Results

The final result dataset contains 120 non-repeating trial codes and binary outputs of 0 (failure) and 1 (success). Each of the eight model-condition combinations contained 15 trials, so every recorded trial was retained in the condition-level calculations. Across all four conditions, ACT succeeded in 41 of 60 trials (68.3%), while SmolVLA succeeded in 37 of 60 trials (61.7%).

Table 2. Condition-Level Task Success by Model

Test Conditions	ACT Success	ACT Success Rate	SmolVLA Success	SmolVLA Success Rate	Difference (Percentage Points)
Familiar Condition	13/15	86.7%	11/15	73.3%	+13.3
Unfamiliar Object	10/15	66.7%	7/15	46.7%	+20.0
Unfamiliar Position	8/15	53.3%	9/15	60.0%	-6.7
Unfamiliar Environment	10/15	66.7%	10/15	66.7%	0.0
All conditions	41/60	68.3%	37/60	61.7%	+6.7

Note: Difference = ACT Success Rate - SmolVLA Success Rate. Percentages are rounded to one decimal place.

The success rates for both models across all four conditions are shown in Figure 10. Under familiar conditions, ACT's success rate was 13.3 percentage points higher than SmolVLA. When using the unfamiliar purple object, ACT's success rate was 20.0 percentage points higher than SmolVLA. Under unfamiliar position conditions, SmolVLA was one trial higher than ACT, making a 6.7 percentage point difference. The two models had the exact same success rate under unfamiliar environment conditions. Therefore, the relative ordering in terms of success rate changed with the types of tests presented instead of remaining constant across all

four conditions.

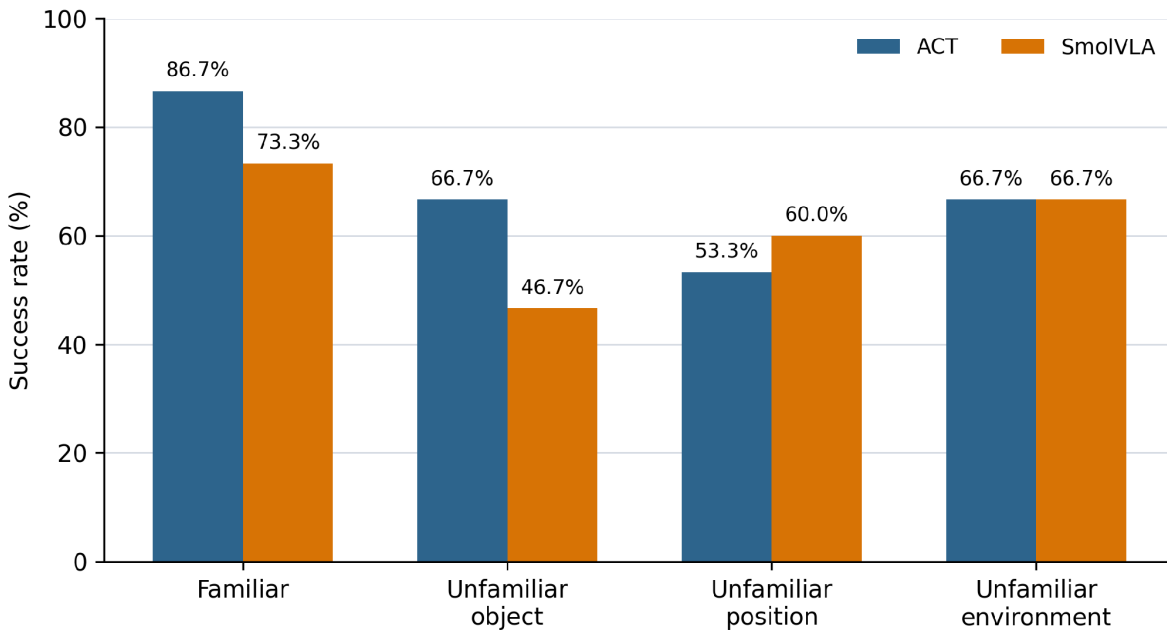


Figure 10. Task success rate by model and evaluation condition (15 trials per bar).

5.2 Generalization Relative to the Familiar Condition

This research aims to evaluate the generalization capabilities of the models by comparing each model's success rate under unfamiliar conditions with its success rate under familiar conditions. ACT's greatest decrease in success rate was under the unfamiliar-position condition, where its success rate dropped by 33.3 percentage points, while SmolVLA's greatest decrease in success rate was under the unfamiliar-object condition, where its success rate dropped by 26.7 percentage points.

Table 3. Success-Rate Changes Relative to the Familiar Condition

Testing Condition	ACT Success Rate	ACT Change in Success Rate (Percentage Points)	SmolVLA Success Rate	SmolVLA Change in Success Rate (Percentage Points)
Unfamiliar Object	66.7%	-20.0	46.7%	-26.7
Unfamiliar Position	53.3%	-33.3	60.0%	-13.3

Unfamiliar Environment	66.7%	-20.0	66.7%	-6.7
------------------------	-------	-------	-------	------

Note: Negative values indicate that there is a decrease in success rate under an unfamiliar condition when compared with the success rate under the familiar condition.

Under the unfamiliar-position and unfamiliar-environment conditions, SmolVLA had a smaller decrease in success rate compared with ACT, while under the unfamiliar-object condition, ACT's decrease in success rate was smaller. However, the results from the unfamiliar environment required particular attention: both models achieved a success rate of 66.7%, but SmolVLA had a smaller decrease in success rate because it also had a lower success rate under the familiar condition. The relative changes therefore complement, rather than replace, the absolute success rates. One possibility is that if a larger or more varied dataset helped SmolVLA learn the basic task more strongly, it might keep this smaller drop under some changes while also raising its absolute success rate. The present experiment cannot confirm that possibility, but it gives future work a clear question to test.

5.3 Spatial Pattern in the Unfamiliar-Position Condition Trials

The cell labels show that the similar condition totals of ACT (eight successes in 15 trials, 53.3%) and SmolVLA (nine successes in 15 trials, 60.0%) were results of different spatial patterns. From the arm's position and viewpoint, Cells 1 to 3 form the furthest row, with Cells 7 to 9 being the closest. ACT succeeded in 0/3 trials in Cell 1, 1/3 in Cell 2, 5/5 in Cell 5, 2/2 in Cell 8, and 0/2 in Cell 9. SmolVLA succeeded in 2/3, 3/3, 4/4, 0/3, and 0/2 trials in those same cells, respectively. Therefore, ACT's successes were mainly concentrated on Cell 5 and Cell 8, while SmolVLA's successes were mainly concentrated on Cell 1, Cell 2, and Cell 5.

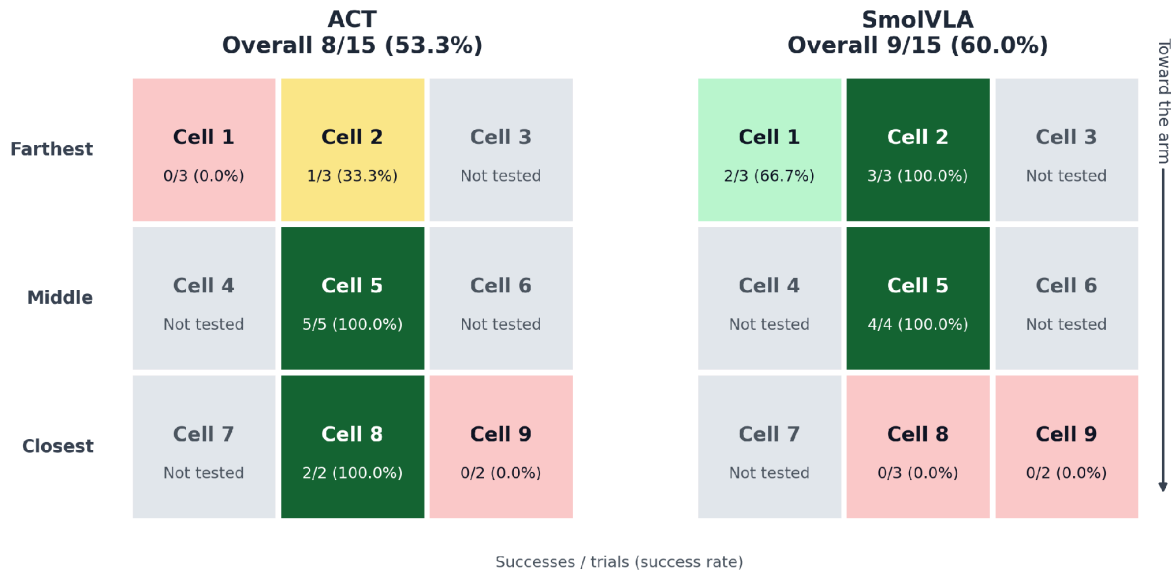


Figure 11. Cell-level success by model in unfamiliar-position trials, viewed from the arm's position. The robotic arm is positioned right below Cell 8. Cells 1 to 3 are the farthest, and Cells 7 to 9 are the closest.

Both models succeeded in every trial in Cell 5 and failed in every trial in Cell 9. However, they demonstrated significant differences in other cells. In the farthest test Cells 1 and 2, SmolVLA succeeded in five out of six trials (83.3%), while ACT only succeeded in one out of six trials (16.7%). However, in Cell 8, ACT succeeded in two out of two trials, while SmolVLA failed all three attempts. The one-trial condition-level advantage for SmolVLA therefore did not reflect uniform superiority across the workspace, since each model was more successful in different parts of the 3 x 3 grid.

Since reaching some cells required more accurate coordination among a greater number of the robotic arm's joints and motors, position-specific movement demands remain a plausible contributor to the differences in success rates among cells. The fact that SmolVLA failed every attempt in Cells 8 and 9 (zero successes in five trials) is consistent with that hypothesis. However, ACT's result of two successes in two trials in Cell 8, followed by zero successes in two trials in adjacent Cell 9, demonstrates that the distance between the cell and the robotic arm does not fully explain the entire pattern. Camera angle, occlusion, and trajectory-specific action errors could all have contributed to the final results.

These explanations remain tentative since the camera feed was the only source of observation input, motor status was not provided to the models, and this research does not include joint kinematic or movement trajectory analysis. In addition, only two to five trials were performed in each cell for each model, and the randomly assigned number of trials per cell is not identical across the two models. Therefore, the cell-level percentages are descriptive rather than statistically conclusive.

5.4 Model Training and Fine-Tuning Time Comparison

ACT training required around 12 hours to reach the 100,000-step checkpoint, while SmolVLA fine-tuning required around 14.5 hours on the same computer. Therefore, in the recorded wall-clock time, ACT finished around 2.5 hours sooner.

Table 4. Approximate wall-clock training and fine-tuning time

Procedure	Initial Status	Optimization Steps	Approximate Time
ACT Training	No prior ACT policy checkpoint. ImageNet-initialized ResNet-18 encoder	100,000	12 hours
SmolVLA Fine-Tuning	Released lerobot/smolvla_base checkpoint	100,000	14.5 hours

The 2.5-hour difference, however, should not be interpreted as an isolated measure of architectural efficiency between the two models because the two models used different initialization conditions, learning-rate schedules, action-chunk settings, preprocessing procedures, and trainable components. In particular, the ACT model did not have a prior ACT checkpoint, while SmolVLA was fine-tuned with a released pre-trained base. The timing result therefore compares the two selected practical training procedures, not two otherwise identical optimization processes.

5.5 Discussion of Findings

The condition-level results show that the difference between the two models depended heavily on the type of information changed by the test. ACT achieved the higher absolute success rates under the familiar and unfamiliar-object conditions, indicating greater stability on the repeated local task and the object change used in this experiment. SmolVLA was slightly higher under unfamiliar positions and tied ACT under unfamiliar lighting. The completed grid records further show that SmolVLA's unfamiliar-position advantage was not uniform across the workspace; it came mainly from stronger results in Cells 1 and 2, whereas ACT performed better in Cell 8. Similar overall totals can therefore be produced by different local strengths and do not, by themselves, measure stability across the entire workspace.

Since the two camera feeds were the only source of data input to the models, every generalization condition changed the visual information that the motion-prediction process directly relies on: the unfamiliar-object condition changed the objects' color, geometric shape, and hardness; the unfamiliar-position condition changed the visual-to-motion relationship; and switching off the three lamps changed image brightness and shadows. Without motor status feedback as an additional observation, both models had to infer the relevant scene and arm configuration from images alone.

Relative to its own success rate under familiar conditions, SmolVLA retained more of its familiar performance under position and lighting changes compared to ACT. However, the

absolute success rate must be taken into consideration alongside the relative success rate, since SmolVLA has a lower familiar condition success rate compared to ACT. Its lower relative success rate result with the purple block also demonstrates that pre-training did not guarantee improved generalization capability for every task.

Several system differences between the ACT and SmolVLA models could impact the observed pattern, including pre-training, visual preprocessing, action-chunk length, training objective, and optimization settings. ACT might have fitted the repeated, limited-range local tasks especially well because this matched ACT’s intended design, while SmolVLA’s pre-trained visual representation might have helped it to retain more performance under visual and spatial alterations.

5.6 Limitations and Data Irregularities

This experiment only involved one SO-101 robotic arm, one pick-and-place task, one dataset containing 150 episodes of demonstrations, one final checkpoint per model, and 15 trials per model-condition combination. Objects and positions were randomized independently rather than matched trial-for-trial between models. In addition, all ACT trials were completed before all SmolVLA trials, and the four conditions were tested in a fixed order within each model. Changes over time in hardware, temperature, or setup could therefore be mixed with model or condition effects.

After each trial, the researcher waited for the robotic arm to reposition itself before placing the next object. However, the robotic arm could return only to approximately the same position, not exactly the same position in every case. The final results were recorded as binary results: success or failure. Task completion time was not recorded other than ensuring all trials were within 45 seconds.

Two SmolVLA trials under unfamiliar environment conditions ENV-22 and ENV-28 experienced component overheating and disconnection after motion and outcome determination. However, since the results for both trials were recorded before the malfunction happened, both trials were kept and not re-recorded. In summary, this analysis remains descriptive and does not make statistical significance or universal performance claims.

6. Future work

The most direct extension would be to increase and balance the test trials and evaluation scale. Each model should receive more repeated trials in the 3 x 3 grid under different environments and with different objects, and both models should use the exact same randomized locations, object order, orientation, and environment setup during the trials. The order of testing for the two models can also be alternated or randomized to decrease the unwanted impact of hardware temperature, component wear, and the passage of time. A larger balanced sample would also make the cell-level differences more reliable and support more confident conclusions or statistical tests.

Future research can also record more detailed process data. The camera feed can remain the only source of input information for model training or fine-tuning and inference, but researchers can record the movement trajectory of each joint, motor status, end-effector paths, and individual task completion times. Failures can also be classified into groups such as “unable to locate,” “unable to pick up,” “unable to drop into the destination box,” and “overtime.” This information would help test whether the closer, higher-numbered cells require coordination among more motors and would help distinguish visual-estimation errors from action-execution errors.

Lastly, the research can be extended to greater variety in colors, shapes, materials, and backgrounds, as well as more complicated manipulation tasks, with multiple random seeds and training checkpoints rather than one final policy per model. Training time could be divided into data processing, forward and backward computation, checkpoint saving, and inference latency, while memory use and energy consumption could also be recorded. If resources permit, pre-trained and scratch-trained SmolVLA runs could be compared while other settings are controlled, providing a more direct test of how much its pre-trained base contributes to the overall generalization capability of the model.

7. Conclusions

When I first began this project, I was interested in one simple but crucial question: a robotic arm might be able to repeat a familiar action, but when the world around it changes, can it still complete that same task? Therefore, what I wanted to research was not just whether ACT and SmolVLA would be able to pick up a block and place it within the destination box, but rather how the models would respond when the object’s appearance changed, its position altered, or it was seen under darker lighting. After collecting 150 demonstrations and completing 120 official trials, the answer turned out to be more complicated than finding one clear winner: Both models learned the basic task, yet they succeeded and failed in noticeably different ways.

Overall, when the task was consistent with the familiar demonstrations, ACT was the more stable model: it achieved a higher absolute success rate, performed better under familiar conditions, and spent less time reaching its final checkpoint during the training process. However, when more variables such as changes in position and lighting were introduced into the tasks, SmolVLA retained more of its original performance, demonstrating that its pre-trained database might have helped it adjust to certain visual differences. To my surprise, however, this pattern was not consistent across every condition or every part of the workspace: For example, ACT succeeded in Cell 8 but failed in the neighboring Cell 9, while SmolVLA performed well in the farther Cells 1 and 2 but failed every attempt in Cell 8. These uneven spatial patterns became one of the most interesting results of the experiment, as they demonstrated that a single overall success rate can hide many smaller, specific strengths and weaknesses.

As I looked beyond the final totals, this research also changed my understanding of the dataset itself: At first, 150 demonstrations seemed like a clear measurement of how much experience the models would receive, but what I did not realize at the time was that each demonstration represented only one example of what the physical world could look like. If most of those examples contained similar objects, positions, backgrounds, and lighting, then the model

could become very capable of repeating that narrow situation without becoming equally capable of handling any degree of change. Therefore, increasing the number of demonstrations may not be enough by itself. The quality, variety, and balance of those demonstrations may be just as important as the total quantity, and a smaller but carefully varied dataset may sometimes prepare a robot for unfamiliar conditions better than a larger dataset filled with nearly identical scenes.

Of course, these conclusions still contain clear limitations: This experiment involved only one robotic arm, one pick-and-place task, one local dataset, and one final checkpoint for each model, while each model-condition combination contained only 15 trials. In addition, because camera images were the models' only observation input, both policies had to estimate the object's location and the arm's current configuration through visual information alone, and when that visual information is interrupted by the arm's own position or movement, uncontrolled variables become an unavoidable part of the experiment. Also, when the robotic arm missed an object, moved toward the wrong position, or failed to complete the placement, I could record what happened, but I could not always determine exactly why it happened. The error may have come from an inaccurate visual estimate, an unsuitable movement path, or the more difficult coordination required among several motors. Future experiments could record joint trajectories, motor status, task-completion time, and more detailed failure types for analysis, while keeping that recorded feedback separate from the information given to the models.

Looking forward, the next step should not be simply gathering more demonstrations but collecting better and more varied demonstrations that expose the robot to a wider and less predictable physical world, just like the real world. Future research could include more objects, positions, materials, backgrounds, lighting conditions, repeated training runs, and more balanced evaluation of trials. Most importantly, this brings the paper back to the larger problem introduced at the beginning: the gap between intelligence on a screen and intelligence that can act through a physical system. This small experiment did not close that gap, but it made the specific challenges within that gap much more visible. A useful robot must do more than memorize one successful movement: It must notice when something has changed, adjust its next action, and continue even when the environment is no longer perfectly familiar. As these models and datasets continue to improve, more adaptable and affordable robots may eventually assist with exhausting repetitive work, enter dangerous environments, support people who need physical help, and explore places where humans cannot safely go. To me, this project is only one small step toward that future, but it has shown me what must happen before artificial intelligence can truly move beyond the screen, become part of the physical world, and develop into true embodied intelligence.

8. References

- Aubakirova, D., Marafioti, A., Noyan, M., Gosthipaty, A. R., Capuano, F., Ben Allal, L., Cuenca, P., Shukor, M., & Cadene, R. (2025, June 3). *SmolVLA: Efficient vision-language-action model trained on LeRobot community data*. Hugging Face. <https://huggingface.co/blog/smolvla>
- Hugging Face. (n.d.). *Imitation learning on real-world robots*. LeRobot documentation. Retrieved August 2, 2026, from https://huggingface.co/docs/lerobot/il_robots
- Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Sanketi, P. R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., & Finn, C. (2025). OpenVLA: An open-source vision-language-action model. *Proceedings of Machine Learning Research*, 270, 2679–2713. <https://proceedings.mlr.press/v270/kim25c.html>
- Lin, F., Hu, Y., Sheng, P., Wen, C., You, J., & Gao, Y. (2025). Data scaling laws in imitation learning for robotic manipulation. *International Conference on Learning Representations*. https://proceedings.iclr.cc/paper_files/paper/2025/hash/88b7b2c896506daabc8d3fd587055167-A-bstract-Conference.html
- Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., & Martín-Martín, R. (2022). What matters in learning from offline human demonstrations for robot manipulation. *Proceedings of Machine Learning Research*, 164, 1678–1690. <https://proceedings.mlr.press/v164/mandlekar22a.html>
- Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., Alibert, S., Cord, M., Wolf, T., & Cadene, R. (2025). *SmolVLA: A vision-language-action model for affordable and efficient robotics* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2506.01844>
- Wen, J., Zhu, Y., Zhu, M., Tang, Z., Li, J., Zhou, Z., Liu, X., Shen, C., Peng, Y., & Feng, F. (2025). DiffusionVLA: Scaling robot foundation models via unified diffusion and autoregression. *Proceedings of Machine Learning Research*, 267, 66558–66574. <https://proceedings.mlr.press/v267/wen25g.html>
- Zhao, T. Z., Kumar, V., Levine, S., & Finn, C. (2023). Learning fine-grained bimanual manipulation with low-cost hardware. *Proceedings of Robotics: Science and Systems XIX*. <https://roboticsconference.org/2023/program/papers/016/>
- Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P. R., Salazar, G., ... Han, K. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. *Proceedings of Machine Learning Research*, 229, 2165–2183. <https://proceedings.mlr.press/v229/zitkovich23a.html>