# Vision-Language-Action Models for Chemical Manipulation: A Practical Study with SmolVLA on the SO-101 Robotic Arm

Ferdinand Paar

Student Nr.: s1128559 Course: BKI209 (6 EC)
Primary Supervisor: Andrei Voinea Project Oversight: Dr. Mathijs Mabesoone
Radboud University, Faculty of Social Sciences, AI Department

June 2026

**Abstract**

Robots such as the Opentrons OT-2 work well when the deck is calibrated and nothing moves, but small physical changes still need human correction. In this project I tested whether a compact Vision-Language-Action (VLA) model can make one simple lab-style manipulation less dependent on fixed coordinates. I fine-tuned SmolVLA-450M on an SO-101 arm for a tube-in-hole task. For the final run, I trained on 50 episodes selected after more than 300 total recorded episodes during setup and testing, and fine-tuned the model on one A100 GPU. The model succeeded in 9 out of 10 trials when the scene matched the training setup, but only 5 out of 10 trials when the tube started in positions absent from the final training set. Moderate lighting changes had limited effect, while a small overhead-camera displacement caused complete failure. The main result is that the model can do the task under tightly controlled conditions, but it depends heavily on fixed cameras, enough spatial coverage in the demonstrations, and a gripper that can hold the tube reliably. Project files, including inference scripts, are available in a private GitHub repository: so101-smolvla-pipe-in-hole. The dataset and trained model are available on Hugging Face Hub: dataset and model.

## Contents

# 1 Introduction

Laboratory automation is already useful in chemistry, especially for repetitive pipetting and sample-transfer steps. Robots such as the Opentrons OT-2 can run these steps with good reproducibility when the deck is calibrated (Bolt et al., 2025). The problem is that this calibration also makes the system quite rigid. If a tube rack is nudged, a vial is tilted, or a consumable is placed a few millimetres away from its expected position, the robot normally does not notice. It follows the same programmed movement until a human catches the mistake and recalibrates the setup.

This matters most in a shared lab rather than in a fully closed industrial cell. In a shared lab, people move equipment, light changes during the day, and objects are often reset by hand. A robot that can use vision to deal with small layout changes would be more useful in that kind of setting. So the question is not only whether the robot can do the movement once, but whether it still works when the camera image changes.

Vision-Language-Action (VLA) models are relevant here because they predict robot actions from camera images and a task description (Brohan et al., 2023; Shukor et al., 2025; Zitkovich et al., 2023). Instead of programming every coordinate by hand, the user records demonstrations and the policy learns from them. This is still imitation learning, so the model depends strongly on what it has seen during training (Hussein et al., 2017; Ross et al., 2011).

For this project, this is the main uncertainty. A compact VLA may learn what a tube looks like, but it may also learn that the tube is usually in one part of the image. A small camera shift could also be enough to make the learned movement fail. I tested this on a simplified lab task: picking up a test tube and inserting it into a rack slot.

I fine-tuned SmolVLA-450M (Shukor et al., 2025), a compact open-source VLA designed for local inference, on an SO-101 6-axis robotic arm. The task was pipe-in-hole, used here as a simplified stand-in for tube-transfer operations on OT-2-style equipment. I then measured how the trained policy behaved under controlled changes to the object starting position, ambient lighting, and overhead-camera viewpoint.

Research question: Under a fixed two-camera SO-101 setup and 50 final training demonstrations, how does SmolVLA-450M performance on a tube-in-hole task change between training-matched conditions, unseen tube starting positions, altered lighting, and a displaced overhead camera?

I answer this with four experiments: a baseline under training-matched conditions (Section 4.1), a change in object starting position (Section 4.2), a lighting change (Section 4.3), and a camera-viewpoint change (Section 4.4). Together, these tests show where the current setup works and where it breaks.

The remainder of the report is organised as follows. Section 2 reviews robotic laboratories, imitation learning, and VLA control. Section 3 describes the hardware, task, data collection, training, and inference procedure. Section 4 reports the four experiments. Section 5 analyses the observed failure modes, and Section 6 discusses limitations and future work. Section 7 concludes.

# 2 Background

## 2.1 Robotic Laboratories and Low-Level Manipulation

Self-driving laboratories combine automated hardware with software that chooses and updates experiments from data (Tom et al., 2024). Examples include autonomous materials platforms (MacLeod et al., 2020) and mobile robotic chemistry systems that move between lab instruments (Burger et al., 2020). These systems are impressive, but they also show a practical bottleneck: the robot still has to handle real objects such as tools, vials, plates, and instruments.

Many current laboratory robots avoid this problem by using fixed deck positions. The OT-2 works well because each object location is calibrated in advance (Bolt et al., 2025). That is enough for

pipetting and fixed consumable layouts, but it does not solve the case where an object is slightly misplaced. This project focuses on that smaller problem. It does not try to build a full self-driving laboratory. It only asks whether a learned visuomotor policy can make one tube-transfer movement less dependent on fixed coordinates.

### 2.2 Imitation Learning and VLA Control

Imitation learning trains a policy from expert demonstrations rather than from hand-written control rules (Hussein et al., 2017). It is useful for small manipulation tasks because a human can show the movement directly with teleoperation. The downside is distribution shift: once the learned policy makes a small mistake, it may enter states that were rare or absent in the demonstrations, and errors can build up from there (Ross et al., 2011). For this reason, the diversity of demonstrations is not just a detail. It is part of the method.

Vision-Language-Action models extend this idea by conditioning the action policy on both images and language. RT-1 used image observations and task instructions to predict real-robot actions at scale (Brohan et al., 2023). RT-2 connected vision-language pretraining with robot action prediction (Zitkovich et al., 2023). SmolVLA follows the same general direction, but in a smaller open-source model intended for lower-cost robotics setups (Shukor et al., 2025).

In this project, the VLA receives two RGB image streams, the current robot state, and the fixed task label `"Pipe in hole"`. I use the language input as task conditioning, not as free-form conversation. The model is not asked to interpret a laboratory protocol or reason about chemistry. It is only asked to execute one learned motor behaviour when the same task label is supplied.

Image understanding in VLA systems is also different from text-only language modelling. The model needs a vision encoder and projection layers that convert images into representations that can be combined with the language/task input and then mapped to robot actions (Shukor et al., 2025; Zitkovich et al., 2023). This matters for the present experiments because changes in object position and camera viewpoint directly change the visual representation on which the policy depends.

### 2.3 SmolVLA-450M

SmolVLA-450M (Shukor et al., 2025) is an open-source VLA model developed for the LeRobot ecosystem. It is small enough to run on consumer hardware, which made it suitable for this project because inference had to run locally beside the robot.

The key advantage is that it predicts chunks of future actions, about 50 timesteps, rather than single actions. This makes robot movements smoother because the policy does not need to decide every motor command independently. For this project, the main constraint was inference. Training could run on the A100 cluster, but deployment had to run beside the robot on my MacBook M1 Pro. The cluster would only solve this bottleneck if I had a low-latency inference tunnel working: the MacBook would send RGB images and joint positions to the cluster, and the cluster would send back the next motor command in time. I did not get that tunnel working, so the model still had to be small enough for local inference.

## 3 Experimental Setup

### 3.1 Hardware

#### SO-101 Robotic Arm System

The SO-101 is an open-source 6-axis robotic arm designed for imitation-learning research (Figure 1). It has these parts:

- **Leader arm:** A passive, low-friction arm moved by hand to record demonstrations during data collection. It was placed next to the operator and outside the task workspace, so it did not

appear in the camera view used by the policy.

- **Follower arm:** The active arm that mirrors the leader during collection and operates autonomously during inference.
- **Feetech STS3215 servos:** Magnetic absolute encoders provide high-precision joint-position feedback at each of the six degrees of freedom.
- **Dual-camera setup:** One fixed overhead camera for the workspace view and one wrist-mounted camera on the follower arm, providing the two visual streams required by SmolVLA.

Figure 1: The SO-101 follower arm used in this project.

### Compute

Inference was run on a MacBook M1 Pro (16 GB unified memory). Fine-tuning was launched on a compute node with four NVIDIA A100 40 GB GPUs; the final training run used one of those GPUs.

### Home-Lab Environment

I set up the SO-101 with dual cameras and a test-tube rack as a home-lab testbed (Figure 2). It mimics an OT-2 deck: small working area, different-sized objects, and a target slot that the robot has to find visually.

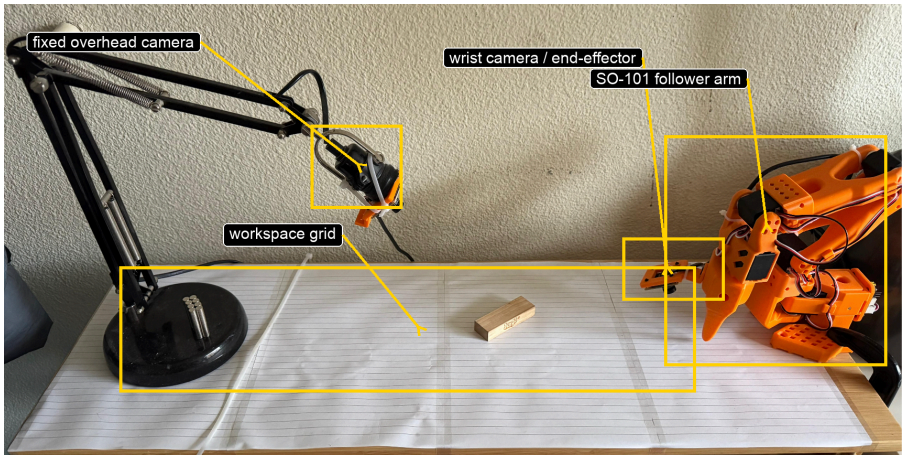



Figure 2: Home-lab testbed. The black stand on the left holds the fixed overhead camera, the orange SO-101 follower arm is on the right, and the paper grid marks the task workspace. The wrist camera is mounted on the follower arm and moves with the end-effector.

### 3.2 Task Design and Data Collection

#### Task: Pipe-in-Hole

The target task was pipe-in-hole: pick up a test tube from a small stand and insert it into a larger hole in a rack (Figure 3). This task has the key challenges of lab manipulation: grasping a small cylinder, placing it accurately in a slot, and finding it under different lighting. The task label was `"Pipe in hole"`.

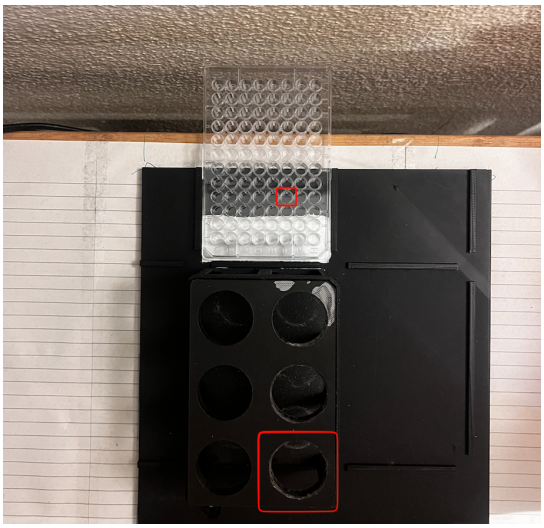


Figure 3: Top view of the experimental setup. The upper red square marks the tube starting position, and the larger lower red square marks the goal position for the tube.

#### Data Collection

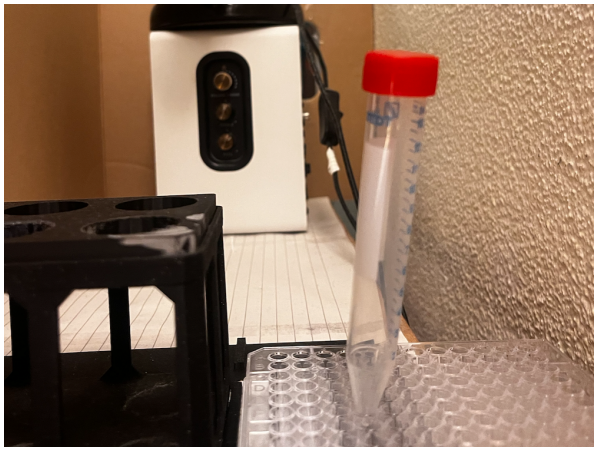
I recorded more than 300 episodes in total while building and debugging the setup. Many of these recordings were learning runs in which I adjusted camera placement, lighting, reset timing, gripper approach, and the episode workflow. The final training run used 50 episodes recorded with the LeRobot `lerobot-record` script and the task label `"Pipe in hole"`.

In practice, each episode was collected as follows. First, the tube was reset by hand to the start region and the follower arm was placed in its initial pose. I then moved the separate leader arm by hand; the follower arm mirrored this movement and the LeRobot script recorded robot state plus both camera streams. During recording I watched the camera display rather than the physical arm. This was important because the learned policy would later receive only the camera views and robot state, not an external view from the human operator. Successful demonstrations were saved immediately after the tube was inserted; failed attempts were discarded and repeated.

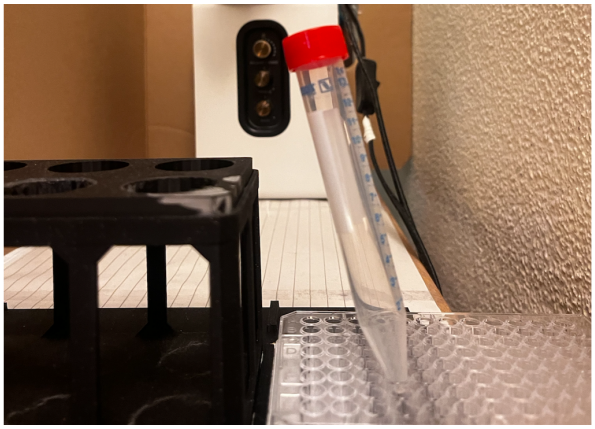
| Data-collection setting | Value used in the final run |
|---|---|
| Robot | SO-101 follower arm controlled by SO-101 leader arm |
| Cameras | Two OpenCV cameras at 640 by 480 pixels and 30 fps |
| Camera names during recording | `front` and `side` |
| Task label | `"Pipe in hole"` |
| Final dataset | `mundgelenk/so101_50_pipe_in_hole` |
| Final episode count | 50 successful demonstrations |
| Total recorded during development | More than 300 episodes, including discarded setup and debugging runs |

Table 1: Main data-collection settings. The exact recording commands are included in the project repository.

Figures 4a and 4b show natural variation across final demonstrations. Tube angle, approach direction, and tube rotation differed between episodes. This variation was useful, but it did not cover every possible tube starting position, which becomes important in Section 4.2.



(a) Episode: one tube orientation.



(b) Episode: a different tube orientation.

Figure 4: Variance across training demonstrations. Tube position and gripper approach angle differed between episodes, introducing natural noise into the dataset.

### 3.3 Training

Fine-tuning was performed with the LeRobot `lerobot-train` script. I did not run a full hyperparameter search. The settings were chosen to match the SmolVLA/LeRobot training workflow, fit reliably on one A100 40 GB GPU, and provide enough steps for the loss curve to flatten.

| Training setting | Value |
|---|---|
| Base policy | `lerobot/smolvla_base` |
| Dataset revision | `mundgelenk/so101_50_pipe_in_hole`, branch `main` |
| Output policy | `mundgelenk/smolvla_so101_pipe_in_hole` |
| Batch size | 16 |
| Training steps | 30,000 |
| Checkpoint frequency | Every 2,500 steps |
| Mixed precision | Enabled with AMP |
| Camera remapping | `front` to `camera1`; `side` to `camera2` |

Table 2: Fine-tuning configuration used for the final SmolVLA run.

I uploaded the dataset and trained model to Hugging Face Hub: dataset repository and model repository.

I logged training to Weights & Biases; the individual metric curves are reproduced in Appendix A. The main signals indicated a clean, stable run:

- **Training loss** (Figure 7) decreased from roughly 0.08 to about 0.01 and then flattened.
- **Gradient norm** (Figure 7) stayed bounded, which suggests numerically stable optimisation.
- **Learning rate** (Figure 7) followed the expected warm-up-then-decay schedule.
- **Resource use** (Figures 8 to 10) stayed stable during the run.

### 3.4 Inference and Reproducibility Artefacts

Inference ran on the MacBook next to the robot. The trained model was downloaded from Hugging Face, two configuration fields that pointed to the original training environment were removed, and the same LeRobot recording script was used with `--policy.path` set to the local model directory. During evaluation, the dataset output was local only and was not pushed to Hugging Face.

The private GitHub repository contains the report source and the runnable scripts needed by future students:

| Script | Purpose |
|---|---|
| `imitation/teleoperate.sh` | Check robot ports, leader-follower control, camera indices, and lighting before recording. |
| `imitation/record_dataset.sh` | Record and upload the final imitation-learning dataset. |
| `imitation/resume_recording.sh` | Continue a partially recorded dataset after interruption. |
| `imitation/train_smolvla.sh` | Fine-tune SmolVLA with the parameters in Table 2. |
| `imitation/download_model.sh` | Download the trained model and clean the local config for inference. |
| `imitation/run_inference.sh` | Run a short local inference session beside the robot. |
| `imitation/eval_10_runs.sh` | Run the 10-episode evaluation setting used for the reported success rates. |

Table 3: Project scripts included in the private GitHub repository.

# 4 Results

I tested the model on the MacBook M1 Pro. Each condition used 10 trials. A trial counted as successful only if the tube was fully inserted into the target hole without manual correction.

| Condition | Change from training setup | Reason for testing |
|---|---|---|
| Baseline | Same camera placement, same start region, same target hole, and similar room lighting | Measures whether the trained policy can reproduce the demonstrated task. |
| Unseen start positions | Tube placed in rack locations not present in the final 50 training demonstrations | Tests whether the policy localises the tube or depends on familiar image regions. |
| Lighting changes | Room and desk lighting changed by switching lamps on/off and moving light sources | Represents ordinary variation in a shared lab. Lighting intensity was not measured, so this condition is qualitative. |
| Camera displacement | Overhead camera moved by a few centimetres and rotated by a few degrees | Represents accidental camera movement during setup changes. Displacement was not precisely measured, so this is a stress test rather than a calibrated perturbation. |

Table 4: Evaluation conditions and their relevance to a shared laboratory setting.

## 4.1 Baseline: Training-Matched Conditions

When I used the same lighting, camera position, and tube placement as in training, the model succeeded 9 out of 10 times.[1] The one failure happened because the gripper slipped slightly during approach, which is a known problem with the angular gripper design (Section 5.2).

This baseline result suggests that VLA-based tube transfer is possible on local consumer hardware when the inference scene closely matches the training scene.

## 4.2 Sensitivity to Object Starting Position

I tested what happens when the tube starts in positions absent from the final training data. Several slots in the test rack, shown in yellow in Figure 5, did not appear in any of the 50 final training episodes.

[1] Example: https://youtu.be/NdbsRoq-Wh8.

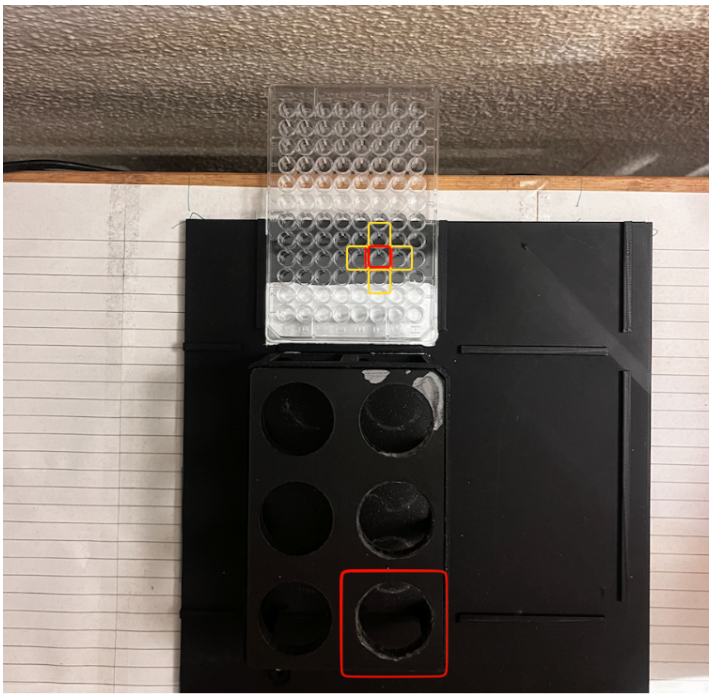


Figure 5: Yellow fields indicate tube starting positions absent from the training data. Placing the tube in these slots reduced success from 9/10 to 5/10.

Accuracy dropped to 5 out of 10. The arm often reached toward the area where the tube usually appeared during training, even when I placed it somewhere else. This does not mean the model learned nothing about the tube; it more likely learned a mix of tube recognition and a strong bias toward familiar image locations.

In practice, the model struggled with spatial generalisation. The final 50 training episodes did not cover enough different regions of the rack, so the action policy stayed too closely tied to the positions it had seen. Section 5.1 discusses this limitation.

### 4.3 Robustness to Lighting Changes

I tested lighting changes by switching lamps on and off and moving light sources around the workspace. Lighting did not strongly affect performance as long as the tube and rack remained clearly visible in both camera streams. When lighting became poor enough to make the camera display difficult for a human operator to interpret, the model also became unreliable.

This result is limited because I did not measure illumination in lux or fix the lamp positions quantitatively. I would not claim full lighting robustness from this test. The narrower result is that moderate visible lighting changes were less damaging than changes in object position or camera viewpoint in this setup. A future laboratory evaluation should use documented LED settings or a light booth so that lighting can be reproduced.

### 4.4 Sensitivity to Camera Position

I moved the overhead camera by a small amount, approximately a few centimetres and a few degrees, to test sensitivity to accidental camera displacement. The model failed in all trials. It did not reliably locate the target slot or initiate a successful grasp (Figure 6).

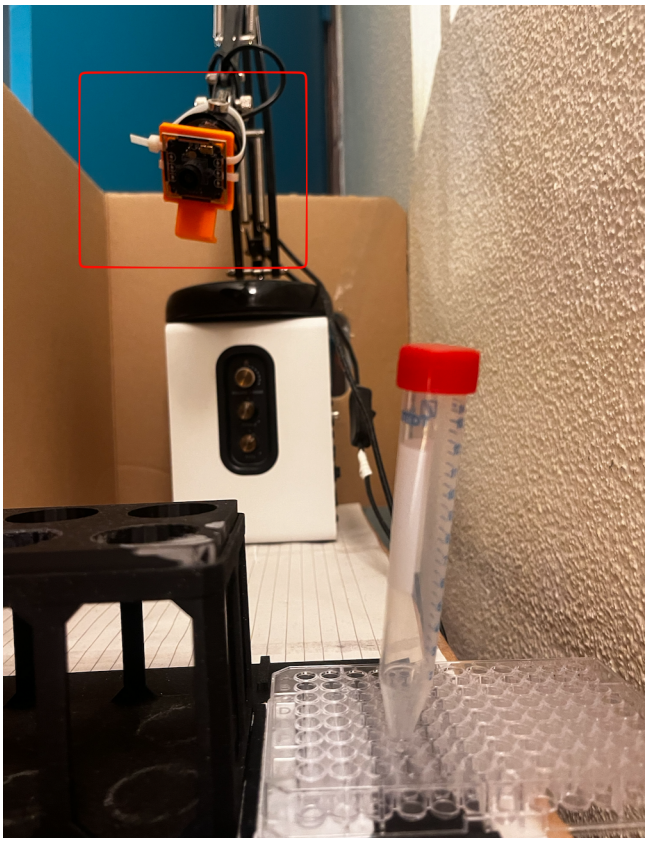


Figure 6: Overhead camera after a small deliberate displacement. Even this minor viewpoint shift caused the VLA to fail entirely.

This result shows that the trained policy was strongly viewpoint-dependent. In a controlled deployment this may be acceptable if the camera is rigidly mounted before data collection and left fixed during use. In a shared laboratory, however, camera displacement is a realistic failure mode because other users may move equipment, clean the bench, or reconfigure the workspace.

# 5 Analysis and Deployment Implications

## 5.1 Dataset Coverage and Spatial Generalisation

The main failure in this project was not baseline execution. Under matched conditions, the model usually performed the task. The main failure was generalisation to tube positions that were missing from the final 50 demonstrations. This is consistent with a standard problem in imitation learning: the policy is trained on the distribution of states shown by the demonstrator, but at inference time it must act under the distribution created by its own observations and actions (Ross et al., 2011). If the visual state is outside the demonstrated distribution, performance can drop sharply.

The drop from 9/10 to 5/10 when the tube started in unseen positions suggests that the model learned a mixture of tube recognition and familiar image-location priors. This does not prove that it memorised pixels only. A more cautious interpretation is that the final dataset did not force the policy to separate "what the tube looks like" from "where the tube usually appears in this camera view."

Larger VLA work shows that more data diversity, larger models, and broader task coverage can improve generalisation (Brohan et al., 2023; Zitkovich et al., 2023). At the same time, small fine-tuned models can still suffer interference between behaviours when tasks or contexts are added without enough balanced data. Recent continual-learning work on VLA models suggests that forgetting and interference depend on model size, pretraining, replay, adapters, and data balance rather than on a

simple fixed number of tasks (Liu et al., 2026; Römer et al., 2026). For my results, the practical interpretation is simple: the final training set covered the demonstrated start region well enough for baseline success, but not enough for reliable position generalisation.

| Issue | Interpretation for this project |
|---|---|
| Spatial coverage | The final 50 episodes did not include several tube start regions, which likely explains the 5/10 result in unseen positions. |
| Visual variation | Lighting varied naturally during recording, which may explain why moderate lighting changes were less damaging than position changes. |
| Viewpoint coverage | The overhead camera was effectively fixed during training, so the policy had no reason to learn viewpoint-invariant features. |
| Model capacity | No fixed task limit is supported by the cited literature. Capacity depends on model size, pretraining, fine-tuning method, replay, and data balance. |

Table 5: How the observed failures relate to coverage in the final training data.

For my setup, I would try two fixes first. The first is to collect more demonstrations from different rack regions and assign prompts that describe those regions, such as `"Pipe in hole from left slot"`, `"Pipe in hole from centre slot"`, and `"Pipe in hole from right slot"`. This uses language conditioning to mark different contexts while keeping the overall task the same. The second is to add image augmentation during training, such as random crops, small zooms, and colour shifts, to reduce dependence on exact pixel locations.

### 5.2 Gripper Design as a Confound

One consistent problem across all tests was the gripper design. The SO-101 has an angular or scissor gripper where the fingers pivot around a pin. For a cylindrical tube, this means it grabs with just a single line of contact instead of squeezing evenly around the tube.

| Property | Angular (Scissor) | Parallel |
|---|---|---|
| Mechanism | Fingers pivot around a pin | Fingers translate linearly |
| Ideal object | Irregular shapes, tight spaces | Cylinders, rectangular blocks |
| Contact area | Single line along object | Full surface contact |
| Force distribution | Uneven; tube can slip or rotate | Even; secure, predictable grip |
| OT-2 relevance | Problematic for tubes/vials | Standard for lab consumables |

Table 6: Angular vs. parallel gripper comparison for cylindrical objects.

This gripper design is a problem because the tube slips or rotates when grabbed.[2] This makes failures harder to interpret: a failed trial may be caused by the policy, but it may also be caused by unstable contact between the gripper and the tube. Switching to a parallel gripper, where fingers close from opposite sides, would make the task closer to standard laboratory handling of cylindrical objects.

### 5.3 Why These Failure Modes Matter for Laboratory Use

These tests matter differently depending on the deployment setting. If the robot is installed in a closed, calibrated cell where cameras, lights, and objects are fixed, then the camera-displacement and

[2]Examples: https://youtube.com/shorts/hJA40Uhh7Cs.

lighting tests are less central. In that case, the baseline result is the most relevant finding: the system can perform the trained movement when the scene is reproduced.

For a shared chemistry lab, the perturbations matter more. Users may move a camera mount, shift a rack during cleaning, change lighting, or reset a tube by hand in a slightly different location. The results indicate that the model can tolerate some lighting variation, but it cannot be trusted after camera movement and it is only partly robust to unseen object starts. For my setup, that means three things are needed before deployment: rigid camera mounting before data collection, broader prompted demonstrations across the working envelope, and a gripper that holds cylindrical tubes consistently.

# 6 Limitations and Future Work

## 6.1 Limitations

The first limitation is dataset coverage. The final 50 demonstrations were sufficient for the training-matched baseline, but they did not cover enough tube start positions to support reliable spatial generalisation. The more than 300 total recordings helped refine the setup, but only the final 50 episodes formed the dataset used for training.

The second limitation is measurement precision. The lighting and camera-displacement tests were useful as stress tests, but they were not measured with laboratory instruments. Lighting was not recorded in lux, lamp positions were not fixed numerically, and the camera displacement was described only approximately. I treat these results as qualitative indicators of sensitivity rather than calibrated robustness measurements.

The third limitation is the gripper. The angular gripper caused tube slip and rotation, which makes failures harder to interpret. A failed trial could reflect a perception or policy error, but it could also reflect unstable contact between the gripper and the tube.

The fourth limitation is compute placement. Training could run on GPU hardware, but inference had to run locally beside the robot. A larger model such as NVIDIA GR00T may be worth testing in a laboratory with GPU inference available (Bjorck et al., 2025), but that would require a workstation, embedded GPU, or reliable low-latency connection from the robot computer to a GPU server.

## 6.2 Future Work

The most direct next step is a more controlled and reproducible data-collection setup:

- Use 3 rigidly mounted cameras at documented positions. This addresses viewpoint rigidity and gives the policy more spatial information.
- Record 150-200 demonstrations from planned grid regions and use prompts that encode the start region. This targets the 9/10 to 5/10 drop without requiring every possible slot to be recorded separately.
- Replace the angular gripper with a parallel gripper. This would reduce tube slip and make policy failures easier to diagnose.
- Control and record lighting. A light booth or fixed LED panels with documented intensity would make the lighting condition reproducible.
- Test GPU-based inference. This would make it possible to compare SmolVLA with larger VLA models while keeping the same dataset and task.

For a full laboratory system, the VLA should be treated as a low-level execution policy rather than as the complete automation system. A separate scheduler or protocol-level controller would still be needed to decide which lab action should happen next, check whether the workspace is safe, and decide

how to recover from failures. The VLA would then execute bounded skills such as tube transfer, while the higher-level controller would handle the experimental protocol.

## 7 Conclusion

This project asked how SmolVLA-450M performance on a tube-in-hole task changes between training-matched conditions, unseen tube starting positions, altered lighting, and a displaced overhead camera. In my setup, the model worked well only when the scene stayed close to the training setup. Under training-matched conditions, it succeeded in 9 out of 10 trials. With unseen tube starting positions, success fell to 5 out of 10, which shows limited spatial generalisation from the final 50 demonstrations. Moderate visible lighting changes were less damaging, but I would treat that result carefully because lighting was not measured quantitatively. A small overhead-camera displacement caused complete failure.

The main takeaway is that compact VLA control can perform a simple lab-relevant manipulation when the physical scene is reproduced closely, but this version is not ready for a busy shared-lab environment. The next version should use rigid camera mounting, broader prompted demonstrations across the working area, measured lighting conditions, and a parallel gripper that can hold cylindrical tubes consistently. These changes would make the method easier to reproduce and would separate actual policy failures from avoidable setup and hardware failures.

## References


Bjorck, J., Castaneda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., Jang, J., Jiang, Z., Kautz, J., Kundalia, K., Lao, L., Li, Z., Lin, Z., Lin, K., Liu, G., … Zhu, Y. (2025). *GR00T N1: An open foundation model for generalist humanoid robots*. arXiv. https://doi.org/10.48550/arXiv.2503.14734

Bolt, R. R. A., Shorthouse, D., & Cook, M. T. (2025). A 3D-printed plate handler for automated handling of well plates on the Opentrons OT-2. *Journal of Open Hardware*, *9*(1). https://doi.org/10.5206/joh.v9i1.22756

Brohan, A., et al. (2023). *RT-1: Robotics transformer for real-world control at scale*. arXiv. https://doi.org/10.48550/arXiv.2212.06817

Burger, B., Maffettone, P. M., Gusev, V. V., Aitchison, C. M., Bai, Y., Wang, X., Li, X., Alston, B. M., Li, B., Clowes, R., Rankin, N., Harris, B., Sprick, R. S., & Cooper, A. I. (2020). A mobile robotic chemist. *Nature*, *583*(7815), 237–241. https://doi.org/10.1038/s41586-020-2442-2

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys*, *50*(2), 1–35. https://doi.org/10.1145/3054912

Liu, H., Kim, C., Liu, B., Liu, M., & Zhu, Y. (2026). *Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning*. arXiv. https://doi.org/10.48550/arXiv.2603.03818

MacLeod, B. P., Parlane, F. G. L., Morrissey, T. D., Häse, F., Roch, L. M., Dettelbach, K. E., Moreira, R., Yunker, L. P. E., Rooney, M. B., Deeth, J. R., Lai, V., Ng, G. J., Situ, H., Zhang, R. H., Aspuru-Guzik, A., Hein, J. E., & Berlinguette, C. P. (2020). Self-driving laboratory for accelerated discovery of thin-film materials. *Science Advances*, *6*(20), eaaz8867. https://doi.org/10.1126/sciadv.aaz8867

Römer, R., Zhang, Y., & Schoellig, A. P. (2026). *CLARE: Continual learning for vision-language-action models via autonomous adapter routing and expansion*. arXiv. https://doi.org/10.48550/arXiv.2601.09512

Ross, S., Gordon, G., & Bagnell, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, *15*, 627–635.

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*. arXiv. https://doi.org/10.48550/arXiv.2506.01844

Tom, G., Schmid, S. P., Baird, S. G., Cao, Y., Darvish, K., Hao, H., Lo, S., Pablo-García, S., Rajaonson, E. M., Skreta, M., Yoshikawa, N., Corapi, S., Akkoc, G. D., Strieth-Kalthoff, F., Seifrid, M., & Aspuru-Guzik, A. (2024). Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, *124*(16), 9633–9732. https://doi.org/10.1021/acs.chemrev.4c00055

Zitkovich, B., et al. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. *Proceedings of the 7th Conference on Robot Learning*, *229*, 2165–2183.

# A Training Curves

This appendix reproduces the individual training metrics from the Weights & Biases dashboard, summarised in Section 3.3. The final training run was launched on a compute node with four NVIDIA A100 40 GB GPUs, with the process using one GPU.

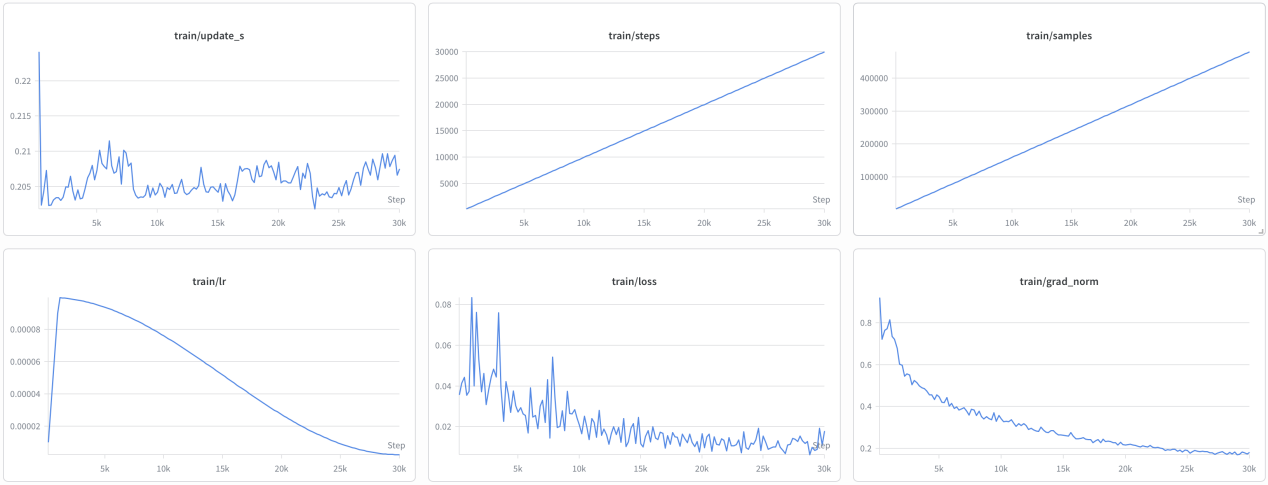



Figure 7: Training dashboard overview showing update time, steps, samples, learning rate, training loss, and gradient norm.


Figure 8: Epoch, episode, data-loading, and system-memory metrics during training.

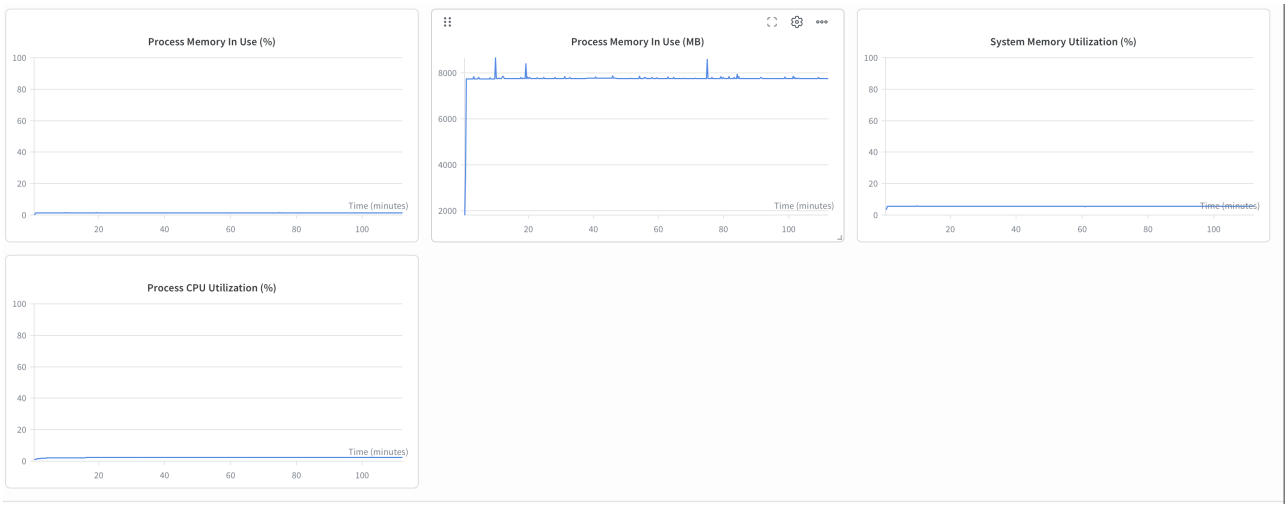



Figure 9: Process memory and CPU utilisation during training.


Figure 10: Network, disk, thread, and memory-availability counters from the training node.