In-the-Wild Evaluation of $\pi_{0.5}$-DROID on a Franka FR3 Robot

Wenqian Zhang
Department of Computer Engineering
University of California, Riverside
wzhan240@ucr.edu

June 11, 2026

Abstract

Vision-language-action (VLA) models aim to enable robots to follow natural language instructions in open-ended real-world environments. In this project, I evaluate a $\pi_{0.5}$/DROID checkpoint on a real Franka FR3 robot equipped with a Robotiq gripper, one wrist-mounted ZED camera, and one external ZED camera. The model is deployed zero-shot without task-specific fine-tuning. I test six tabletop and interaction-oriented manipulation tasks: placing bread on a plate, placing a pen on a plate, sequentially placing a pen and bread on a plate, placing all objects into a basket, placing a black pen into a red cup, and handing bread to an outstretched human hand. Across 60 total trials, the policy achieves perfect success on simple pick-and-place tasks and the flexible multi-object basket task, with 10/10 success on bread-to-plate, pen-to-plate, and all-objects-to-basket. Performance drops on more constrained tasks: the sequential pen-then-bread task achieves 8/10 success and often violates the specified order by attempting to grasp the bread before the pen, while the black-pen-to-red-cup task achieves only 3/10 autonomous success due to hovering, freezing, and precise grasping failures. The handover task fails in all 10 trials, suggesting that the checkpoint does not reliably understand or execute human-interaction tasks outside common tabletop placement. Overall, the results suggest that the $\pi_{0.5}$-DROID checkpoint transfers surprisingly well for many tabletop behaviors, but remains sensitive to small-object grasping, precise placement, camera resolution, long-horizon instruction grounding, and human-aware interaction.

1 Introduction

Generalist robot policies have recently become a promising direction for real-world manipulation. Instead of training a separate policy for every object, environment, and task, vision-language-action models attempt to map language instructions and visual observations directly into robot actions. This is attractive because it could reduce the need for task-specific data collection and engineering when deploying a robot in a new workspace.

However, real-world zero-shot deployment remains challenging. Even when a model is trained or fine-tuned on a robot platform similar to the target robot, small differences in embodiment, camera placement, gripper behavior, controller implementation, object appearance, lighting, and workspace layout can cause large changes in performance. Therefore, evaluating these models on real hardware is important, even when the evaluation is small-scale.

In this final project, I evaluate a $\pi_{0.5}$/DROID checkpoint on a Franka FR3 robot. The project is inspired by the "π_0 in the Wild" style of evaluation, where the emphasis is on testing a generalist policy on realistic, user-specified manipulation tasks rather than on a fixed simulation benchmark. My goal is to answer four questions:

1. Can the $\pi_{0.5}$/DROID checkpoint perform simple tabletop pick-and-place tasks on our Franka FR3 setup?
2. How does performance change as the task becomes more compositional or requires more precise placement?
3. Can the policy handle interaction-oriented goals such as handing an object to a human?
4. What are the dominant failure modes during real-world execution?

The main contribution of this report is an empirical case study of deploying an existing VLA checkpoint on a real Franka FR3 robot. Instead of proposing a new model, I focus on practical deployment behavior: whether the robot approaches the correct object, whether it closes the gripper at the right time, whether it lifts and transports the object, and whether it completes the requested placement. A second contribution is a set of qualitative observations about controller smoothness and out-of-distribution (OOD) tasks. With Polymetis control, the robot motion is surprisingly smooth and compliant, and the system can solve many basic manipulation tasks zero-shot. At the same time, performance degrades when the task requires precise contact geometry, strict temporal ordering, unfamiliar affordances, or human interaction.

2 Related Work

Vision-language-action models. Recent robot foundation models use vision-language backbones and large-scale robot data to produce low-level actions conditioned on natural language instructions. π_0 is a flow-matching VLA model designed for general robot control, using a pretrained vision-language model and an action expert to generate continuous action chunks for dexterous manipulation (Black et al., 2024). $\pi_{0.5}$ extends this direction by using co-training over heterogeneous sources, including multiple robots, semantic prediction, web data, object detections, and low-level actions, with the goal of improving open-world generalization (Physical Intelligence et al., 2025).

Large-scale robot manipulation datasets. Large and diverse robot datasets are a key ingredient for generalist policies. The DROID dataset contains large-scale in-the-wild manipulation demonstrations collected with a standardized Franka-based setup across many scenes and tasks (Khazatsky et al., 2024). Since the DROID platform uses Franka arms and multiple camera views, DROID-tuned checkpoints are especially relevant for testing on Franka-style lab setups. However, even similar hardware can differ in camera viewpoint, calibration, controller stack, gripper dynamics, and workspace layout, which motivates direct deployment experiments.

In-the-wild robot policy evaluation. Instead of evaluating only on fixed benchmarks, recent work has also tested generalist policies in real environments with user-generated instructions. Penn-PAL's π_0 in-the-wild evaluation tested a DROID-tuned model across many realistic tasks and highlighted several practical phenomena: sensible behavior does not always imply task success, prompt phrasing can significantly change performance, and failure modes include freezing, collision issues, and fine-grained manipulation errors (Penn PAL Lab, 2025). My project follows this evaluation philosophy but focuses on a smaller but controlled set of tasks on our Franka FR3 setup.

3 Method

3.1 Robot Setup

The experiments are conducted on a Franka FR3 robot equipped with a Robotiq parallel gripper. The visual input is provided by two ZED cameras: one wrist-mounted ZED camera attached near the robot end-effector, and one external ZED camera observing the tabletop workspace. This follows the observation structure used by the $\pi_{0.5}$-DROID inference pipeline, where the policy is conditioned on one external camera view and one wrist camera view.

The robot operates in a tabletop manipulation workspace containing common household-like objects such as bread, pens, a plate, a basket, and a cup. The robot receives a free-form natural language instruction for each rollout, observes the scene through the two RGB camera streams, and executes low-level actions through the robot control stack. In our deployment, the Franka FR3 is controlled through the same action interface expected by the DROID policy: joint velocity commands for the 7 robot joints and a gripper position command for the Robotiq gripper.

3.2 Policy Checkpoint and Inference Interface

I use the `pi05_droid` checkpoint from the openpi framework. This checkpoint is a $\pi_{0.5}$ model fine-tuned on the DROID robot manipulation dataset and is used directly for inference without collecting additional demonstrations or

Table 1: Hardware and deployment setup.

Component	Description
Robot arm	Franka FR3 7-DoF robot arm
Gripper	Robotiq parallel gripper
Wrist camera	ZED camera mounted near the end-effector, used as the wrist-view RGB input
External camera	ZED camera observing the tabletop scene, used as the external RGB input
Workspace	Tabletop manipulation environment with bread, two pens, plate, basket, cup, and small distractor objects
Policy checkpoint	$\pi_{0.5}$-DROID checkpoint from openpi
Control interface	7-D joint velocity action plus 1-D gripper position action
Controller	Polymetis-based control stack
Evaluation mode	Zero-shot real-world deployment without task-specific fine-tuning

performing task-specific fine-tuning. Therefore, this project evaluates the transfer behavior of an existing generalist VLA policy under a new real-world lab setup.

The policy outputs an action chunk rather than a single action. In the DROID interface, the predicted action chunk has shape $(10, 8)$, corresponding to 10 future action steps. Each action step is 8-dimensional: 7 dimensions for robot joint velocity control and 1 dimension for the gripper position command. During execution, the robot runs part of the predicted chunk open-loop before querying the policy again.

Formally, each inference call can be written as:

$$a_{t:t+H} = \pi_\theta(I_t^{\text{ext}}, I_t^{\text{wrist}}, q_t, g_t, l), \tag{1}$$

where I_t^{ext} is the external camera image, I_t^{wrist} is the wrist camera image, q_t is the robot joint position, g_t is the gripper position, l is the language instruction, and $a_{t:t+H}$ is the predicted action chunk. In this deployment, $H = 10$ for the raw policy output, and each action has dimension 8.

3.3 Controller and Deployment Observation

A practical observation from this project is that the control stack has a large effect on the perceived quality of policy execution. In our setup, running the $\pi_{0.5}$-DROID checkpoint through the Polymetis-based control stack produced surprisingly smooth robot motion. The arm did not move in a jerky or unstable way during successful rollouts. Instead, the behavior appeared compliant and well-damped, which made the robot safe and natural to observe during tabletop manipulation. Qualitatively, the system exhibited very good impedance-like behavior in real-world execution.

This was an important part of the deployment experience. Before running these experiments, I expected zero-shot VLA deployment on real hardware to be highly brittle. However, with the Franka FR3, Robotiq gripper, ZED camera observations, and Polymetis control, the robot was able to complete many basic manipulation tasks without any task-specific fine-tuning. This was surprising because the system was not trained on our exact robot workspace, objects, camera placement, or gripper configuration. The result suggests that the $\pi_{0.5}$-DROID checkpoint provides useful manipulation priors, while the controller helps convert the predicted action chunks into smooth real-world execution.

At the same time, these experiments should not be interpreted as showing fully general robot intelligence. The smoothness of motion and the success on common pick-and-place tasks do not guarantee that the model understands arbitrary objects or arbitrary physical goals. The following experiments therefore focus not only on success rates, but also on where the policy begins to fail under more precise, sequential, interaction-oriented, or out-of-distribution instructions.

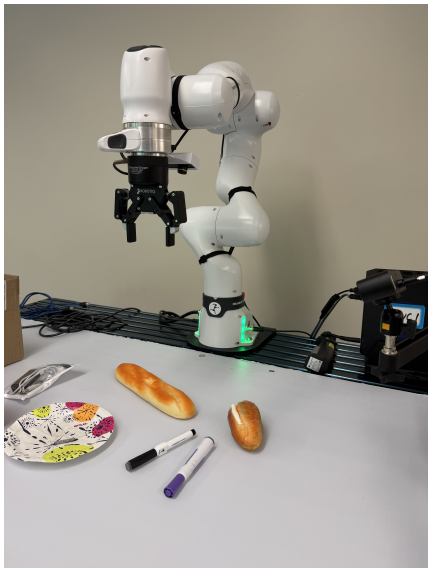

Figure 1: Experimental setup. The system consists of a Franka FR3 robot arm, a Robotiq parallel gripper, a wrist-mounted ZED camera, and an external ZED camera observing the tabletop workspace.

Table 2: Evaluation tasks.

Task ID	Language prompt	Skill type
T1	Pick up the bread and place it on the plate	Single-object pick-and-place
T2	Pick up the pen and place it on the plate	Single-object pick-and-place
T3	Pick up the pen and place it on the plate, then pick up the bread and place it on the plate	Sequential pick-and-place
T4	Pick up all the objects into the basket	Multi-object collection
T5	Place the black pen into the red cup	Fine-grained object-target placer
T6	Hand the bread to the outstretched hand	Human-interaction / handover

3.4 Task Design

I evaluate the policy on six tasks with increasing difficulty and different structure. The first two are simple single-object pick-and-place tasks. The third and fourth require longer-horizon or multi-object behavior. The fifth requires precise small-object manipulation and target alignment. The sixth tests human-interaction behavior.

3.5 Evaluation Protocol

For each task, I run 10 real-world trials, resulting in 60 total rollouts. Each rollout starts from a similar tabletop configuration with small variations in object position and orientation. The robot is given a natural language instruction and is allowed to execute the policy until the task is completed, the robot stops making progress, or human intervention becomes necessary for safety.

I use three evaluation labels:

- **Success:** the robot completes the full task autonomously.
- **Partial success:** the robot clearly grounds the correct object or target and makes meaningful progress, but fails to complete the task without intervention.
- **Failure:** the policy does not make useful progress, repeatedly freezes, or interacts with the wrong object or target.

Table 3: Task-level results over 10 trials per task.

Task	Success	Partial	Failure	Main observation
Bread → plate	10/10	0/10	0/10	Reliable large-object pick-and-place.
Pen → plate	10/10	0/10	0/10	Reliable despite the pen being thinner than the bread.
Pen → plate, then bread → plate	8/10	2/10	0/10	Often completes the task, but sometimes violates the specified order and attempts the bread first.
All objects → basket	10/10	0/10	0/10	Robust when the target is large and the instruction does not require a strict order.
Black pen → red cup	3/10	7/10	0/10	Often grounds the pen but hovers or freezes before grasping; precise placement is difficult.
Bread → outstretched hand	0/10	0/10	10/10	The robot appears confused and does not reliably understand the human handover goal.

4 Experiments

4.1 Overall Results

Table 3 summarizes the results. The policy performs very well on simple pick-and-place tasks and on the flexible multi-object basket task. It achieves 10/10 success for placing bread on a plate, placing a pen on a plate, and placing all objects into a basket. However, the success rate decreases for tasks that require either strict temporal ordering or precise object-target placement. The sequential pen-and-bread task reaches 8/10 success, while the black-pen-to-red-cup task reaches only 3/10 autonomous success. The handover task fails in all 10 trials.

4.2 Simple Pick-and-Place Tasks

The first two tasks evaluate basic object grounding, grasping, transport, and placement. The bread-to-plate task succeeds in all 10 trials. This task is relatively easy because the bread is visually salient, physically large, and tolerant to small grasp pose errors, while the plate provides a forgiving placement target.

The pen-to-plate task also achieves 10/10 success. Although the pen is thinner and requires more precise gripper alignment than the bread, the plate still makes the final placement stage forgiving. Together, these results suggest that the $\pi_{0.5}$-DROID checkpoint transfers well to common tabletop pick-and-place behaviors on the Franka FR3 setup.

4.3 Sequential and Multi-Object Tasks

The sequential task asks the robot to pick up the pen, place it on the plate, then pick up the bread and place it on the plate. The policy succeeds in 8/10 trials, but sometimes violates the specified order by attempting the bread first. One likely reason is that the pen is harder to grasp than the bread; after one or two unsuccessful pen grasp attempts, the policy tends to switch toward the easier bread object. This suggests that the model often understands the overall task structure, but does not always preserve the temporal order specified in language.

The all-objects-into-basket task achieves 10/10 success. Although this task is also long-horizon, it is more flexible because the instruction does not specify an object order and the basket is a large target. Compared with the strict sequential task, this allows the policy to choose an easier order and tolerate less precise placement, which likely explains its higher reliability.

Figure 2: Representative rollout visualization for the sequential task: "pick up the pen and place it on the plate and pick up the bread and place it on the plate." The policy often completes both subtasks, but in some failures it violates the requested order and attempts to grasp the bread before the pen.

4.4 Fine-Grained Placement: Black Pen into Red Cup

The most difficult tabletop task is placing the black pen into the red cup, with only 3/10 autonomous success. In many partial trials, the robot moves toward the correct pen and hovers near it, indicating that language grounding is often correct. However, it frequently freezes or remains slightly above the pen instead of moving down far enough for a stable grasp. Interestingly, when a human manually pushes the arm slightly downward during this hovering phase, the policy can often grasp the pen and complete the task. This suggests that the main bottleneck is not object recognition, but precise depth estimation and final grasp execution.

This task is harder than pen-to-plate because the black pen is small and thin, and the cup requires much more precise target alignment than a plate or basket. Since the policy input images are resized to 224×224, and the exterior view includes padding, small objects occupy limited image area. This may reduce effective visual resolution and make centimeter-level grasping and insertion more difficult.

4.5 Human Handover: Bread to an Outstretched Hand

I also evaluate a human-interaction task: handing the bread to an outstretched human hand. This task fails in all 10 trials. This contrasts sharply with the 10/10 bread-to-plate result using the same bread object, suggesting that the failure is not caused by the object itself. Instead, the policy does not reliably recognize the human hand as a valid target or execute a handover behavior.

This highlights an important limitation of zero-shot VLA deployment. Although "hand me the bread" is semantically simple for humans, it requires interaction-specific concepts such as recognizing the recipient, approaching the hand safely, and releasing the object at the right time. The $\pi_{0.5}$-DROID checkpoint appears much more reliable for object-to-object tabletop manipulation than for object-to-human interaction.

4.6 Failure Mode Analysis

The results reveal several recurring failure modes.

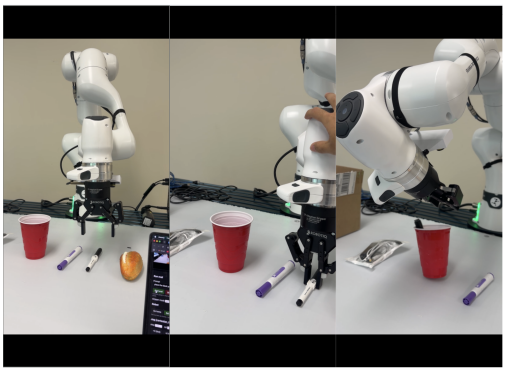

Figure 3: Representative rollout visualization for the black-pen-to-red-cup task. The policy often localizes the pen correctly but hovers above it or freezes before establishing a stable grasp, suggesting that precise depth estimation and final grasp execution are the main bottlenecks.

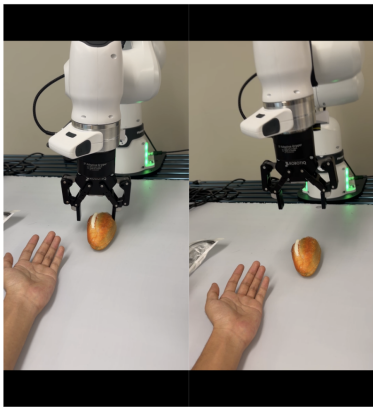

Figure 4: Representative rollout visualization for the handover task: "hand the bread to the outstretched hand." Unlike plate or basket placement, the policy does not reliably understand the outstretched human hand as a valid target and often appears confused.

Object grounding is often correct. Even in the difficult black-pen-to-red-cup task, the robot often moves toward the correct pen. This suggests that the model has meaningful language-conditioned visual grounding. The failures usually occur after the robot has already reached the correct object region.

Precise grasping is the main bottleneck. The most common failure occurs when the gripper hovers above a small object or repeatedly makes small adjustments without closing at the correct pose. This is especially visible for the black pen. Since human downward intervention can sometimes allow the policy to recover and finish the task, the failure appears to be related to final grasp pose accuracy rather than high-level task understanding.

Strict temporal ordering is not always preserved. In the sequential task, the policy sometimes attempts the bread before the pen even though the instruction asks for the pen first. The robot seems to prefer the easier object after pen grasping becomes uncertain. This suggests that the policy may optimize for task completion or local action feasibility rather than strictly following the specified language order.

Human-interaction goals do not transfer automatically. The complete failure on the handover task suggests that success on tabletop manipulation does not automatically transfer to human-interaction settings. The policy does not

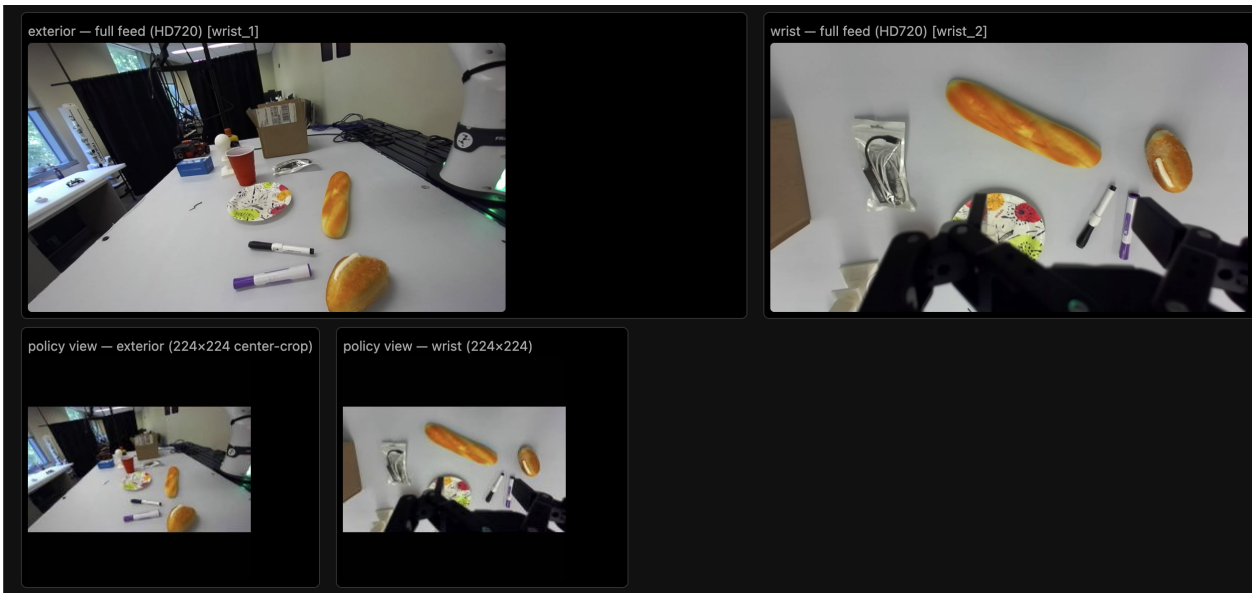

Figure 5: Camera and policy observation dashboard used during deployment. The top row shows the original HD720 exterior and wrist camera feeds, while the bottom row shows the resized 224×224 policy inputs. Although the full-resolution camera feed contains rich visual information, small objects such as pens occupy only a limited number of pixels after resizing and cropping, which may make precise grasping and depth estimation more difficult.

appear to treat the outstretched human hand as a familiar or valid placement goal.

Camera resolution and viewpoint may limit fine manipulation. The policy receives 224×224 RGB observations from the exterior and wrist cameras. In the exterior view, padding and downsampling reduce the effective resolution of small objects. The wrist camera provides local visual information, but the final approach still depends on accurate depth and gripper-object alignment. These observation limitations likely contribute to the hovering behavior observed in the black-pen-to-red-cup task.

‘‘‘

4.7 Additional Qualitative Probes with Out-of-Distribution Objects

Beyond the main six evaluation tasks, I also tested several informal qualitative prompts with objects that were likely outside the common tabletop manipulation distribution. For example, I tried using a 3D-printed light bulb object and also asked the robot to push a small toy car. These trials were not included in the main quantitative table because they were less standardized, but they provided useful observations about the boundary of the checkpoint's zero-shot behavior.

In these out-of-distribution cases, the robot often appeared uncertain or failed to produce a meaningful strategy. This is different from the bread, pen, plate, basket, and cup tasks, where the robot often executed recognizable pick-and-place behaviors. One likely explanation is that the policy has learned strong priors for common manipulation patterns from the DROID data distribution, but does not necessarily know how to interact with unusual objects or less common goals. A 3D-printed light bulb may not correspond to a familiar object category or manipulation affordance, and pushing a toy car requires a different behavior pattern from grasping and placing.

These qualitative probes suggest that the model's zero-shot ability is impressive but still distribution-dependent. When the instruction, object, and target resemble common robot demonstration data, the policy can be highly effective. When the task requires an unfamiliar affordance or an object outside the training distribution, the policy may fail even if the instruction is simple from a human perspective.

4.8 Discussion

Overall, the $\pi_{0.5}$-DROID checkpoint transfers surprisingly well to the Franka FR3 setup for many tabletop manipulation tasks. The 10/10 success rates on bread-to-plate, pen-to-plate, and all-objects-to-basket show that the model can perform meaningful zero-shot manipulation using our Robotiq gripper and ZED camera setup. This is encouraging because the policy was not fine-tuned on our specific robot, objects, camera placement, or workspace.

At the same time, the results show that high success on simple tasks does not imply robust general manipulation. The model struggles when the task requires precise small-object grasping, insertion into a narrow target, or strict adherence to an ordered instruction. The complete failure on the handover task further suggests that success on tabletop manipulation does not automatically transfer to human-interaction settings.

The most surprising finding is how capable the system is under the right conditions. With the $\pi_{0.5}$-DROID checkpoint and Polymetis control, the robot motion is smooth and the policy can solve many basic tabletop tasks zero-shot. However, the failures on handover, precise cup insertion, and out-of-distribution objects show that this capability is not uniform. The model appears strongest when the task matches common pick-and-place patterns, and weakest when the task requires human interaction, precise contact geometry, unfamiliar object affordances, or behavior beyond standard grasp-and-place manipulation.

5 Conclusion

In this project, I evaluated a $\pi_{0.5}$-DROID checkpoint on a real Franka FR3 robot with a Robotiq gripper, a wrist-mounted ZED camera, an external ZED camera, and a Polymetis-based control stack. Across six tasks and 60 total trials, the policy showed strong zero-shot performance on simple and tolerant tasks, including 10/10 success on bread-to-plate, pen-to-plate, and all-objects-to-basket. However, performance decreased for more constrained tasks. The sequential pen-and-bread task achieved 8/10 success but sometimes violated the requested object order, while the black-pen-to-red-cup task achieved only 3/10 autonomous success due to hovering, freezing, and precise grasping failures. The handover task failed in all 10 trials, indicating that human-interaction behaviors do not transfer as reliably as standard tabletop manipulation.

These results suggest that current VLA policies already contain useful priors for real-world tabletop manipulation, especially when paired with a smooth control stack such as Polymetis. At the same time, robust deployment still depends heavily on the details of the hardware and observation pipeline. In particular, small-object manipulation, precise placement, strict temporal instruction following, unfamiliar affordances, and human-aware interaction remain challenging. Future work could evaluate more prompt variants, improve camera placement and preprocessing, add local grasp correction, or fine-tune the policy on a small number of demonstrations from the target Franka FR3 setup.

References

K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Penn PAL Lab. Evaluating π_0 in the wild: Strengths, problems, and the future of generalist robot policies. https://penn-pal-lab.github.io/Pi0-Experiment-in-the-Wild/, 2025. Accessed June 2026.

Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.