Evaluating π_0 on Long-horizon Manipulation Tasks

Justin Yue Mingxuan Yan Evan Goldman
University of California, Riverside

Abstract

Empirical studies have shown that while vision-language-action models like π_0 excel at generalization, they struggle in long-horizon scenarios such as multi-step manipulation tasks and fail to recover from unexpected failure modes. In this project, we investigate the effect of lifelong learning capabilities on π_0's performance in long-horizon manipulation tasks. Using the comprehensive LIBERO benchmark—which provides diverse task suites for assessing both declarative and procedural knowledge transfer, we systematically evaluate π_0's performance under different tasks after fine-tuing. Our goal is to identify critical failure modes and potential solutions that future works can explore. Thus, the insights gained from this work aim to contribute toward building more robust, adaptive embodied agents capable of continuous learning in dynamic environments.

1 Introduction

The advent of foundation models has reinvigorated the development of embodied agents that are able to intelligently interact and navigate within their environment. For this to be possible, these agents must learn to be generalists. Two approaches come to mind: multitask and lifelong learning. Multitask learning [5] initializes an agent to be knowledgeable with set tasks, but achieving good performance across these tasks is difficult and is not flexible with learning new tasks. Instead, lifelong learning [10], also known as continual learning, allows agents to evolve over time in respond to changes in the environment and the tasks needed to address said changes. Due to lifelong learning's advantage in pragmatism, interest has grown in lifelong learning, leading to the creation of datasets and benchmarks such as the LIBERO dataset [9].

Another aspect of creating a generalized embodied AI agent is finding the right training recipe. π_0 [1] is one such work that seeks to address this challenge by developing in a novel vision-language-action (VLA) framework. In this framework, the authors propose several improvements such as incorporating flow matching with vision-language models (VLMs) and curating the largest robot learning dataset. While the π_0 model generalizes well and outperforms its baselines in the authors' evaluation, the authors note that π_0 may not perform well on certain embodiments and tasks. Some third-party evaluations even show that π_0 does not attempt to recover from these failure modes despite exhibiting this emergent behavior on tasks in its trained dataset.

In this work, we intend to better understand how we can improve π_0's generalization and long-horizon performance by finetuning on the LIBERO [9] dataset. Specifically, we use LIBERO-10 since this subset of the LIBERO dataset consists of long-horizon tasks and evaluate a fine-tuned π_0 checkpoint on this subset. We find that despite being fine-tuned, π_0's performance is not uniform across the tasks and struggles with one particular more than the others. From these rollouts, we analyze and provide future directions to explore and address this problem.

For EE 260: Trustworthy AI for Autonomous Systems

2 Related Works

2.1 Datasets

Compared to datasets for image classification and natural language processing, existing datasets for robot learning are scarce. One attempt to address this issue is the Open-X Embodiment Dataset [4], where multiple labs worked together to collect data on multiple tasks with different robot embodiments. The DROID dataset [6] is another effort that leverages the same robot platform throughout the data collection but focuses on more scene diversity.

2.2 Vision-language-agent (VLA)

RT-1 [3] was one of the first methods that explored the usage of the transformer architecture to create the ideal model architecture. Trained on data collected from many sources, RT-1 can output a goal pose for the robot. With the advent of large language models (LLMs), RT-2 [2] improved on its predecessor by incorporating a VLM, embuing itself with Internet-scale knowledge and allowing for better promptability. The addition of RT-2 saw the introduction of the term, vision-language-action (VLA) models. More recently, OpenVLA [7] became the first open-sourced VLA model that demonstrated that large amount of model parameters is not necessary for effective robot learning.

2.3 Flow Matching for Efficient Robot Control

Autoregressive models such as OpenVLA [7] tends to be slow due to performing inference dependent on the previous values each time. Flow matching [8], a variant of diffusion model, addresses this issue by solving the optimal transport between two distributions in a continuous manner using ordinary differential equations (ODE). This method is especially useful for action chunking, allowing for 50 Hz during runtime.

2.4 π_0: VLM-Flow Model for General Robot Control

π_0 [1] is one of the SOTA VLAs. It leverages a pre-trained VLM as its backbone, enabling it to inherit semantic knowledge from large-scale internet data. This foundation allows the model to comprehend and process complex instructions and visual inputs. To translate this understanding into physical actions, π_0 incorporates a flow-matching architecture, which facilitates the generation of continuous and precise actions necessary for dexterous manipulation tasks.

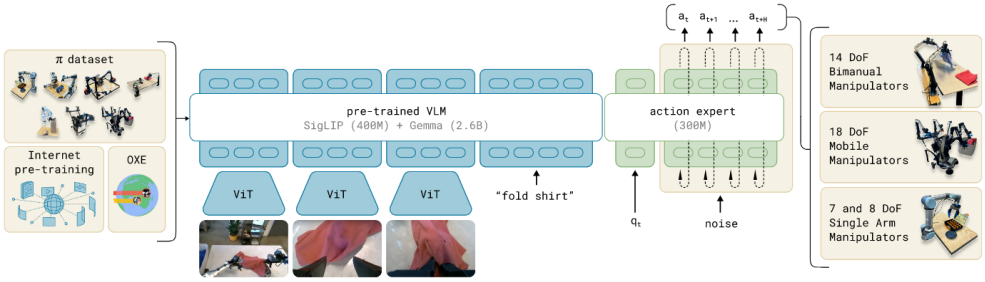

Figure 1: The architecture of π_0.

As shown in Fig. 1, in π_0 the VLM and the flow matching components are integrated through an "action expert" module. This module augments the VLM by producing flow-based outputs that are conditioned on the semantic representations derived from the vision and language inputs. This design ensures that the action generation is directly informed by the contextual understanding provided by the VLM.

Training π_0 involves a two-phase process: an initial pre-training phase using a diverse dataset collected from multiple robotic platforms—including single-arm robots, dual-arm robots, and mobile manipulators—followed by fine-tuning on specific tasks to enhance performance. This methodology enables the model to perform tasks in a zero-shot manner after pre-training and to adapt to new skills through fine-tuning.

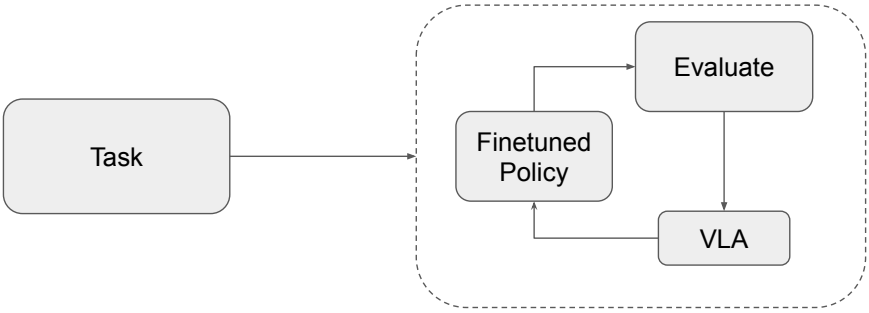

Figure 2: Evaluation Pipeline of π_0 on LIBERO-10

2.5 Lifelong Learning

In a lifelong robot learning problem, a robot sequentially learns over K tasks $\{T^1, \ldots, T^K\}$ with a single policy π [9]. Hopefully, prior knowledge of the agent can facilitate the learning of new tasks (forward transfer); the newly learned knowledge can reinforce the performance of previous tasks (backward transfer).

The main distinction between lifelong learning and multi-task learning is that the agent loses access to the data of previous $k-1$ tasks when optimizing T^k. It forces an agent to abstract the data samples and learn universal knowledge. It is even challenging when an appropriate task curriculum is impossible (known as robustness to task ordering [9]).

3 Task and Datasets

Future robots must adapt post-deployment through lifelong learning. To simulate this paradigm, LIBERO [9] (Lifelong Robot Learning Benchmark) is designed to address critical challenges in lifelong learning for decision-making (LLDM), focusing on both declarative knowledge (e.g., object relationships) and procedural knowledge (e.g., action sequences) transfer. It consists of 4 components: LIBERO-Object, LIBERO-Spatial, and LIBERO-goal that each have 10 tasks to test for positive transfer on object, spatial relationships, and task goals respectively. LIBERO-100 consists of 100 tasks and tests on entangled knowledge. LIBERO-100 can be split into LIBERO-90 for 90 short horizon tasks to be used for pretraining and LIBERO-Long for 10 long-horizon tasks to be used for evaluation. Besides these fixed task suites, LIBERO also provides a pipeline to generate more task suites in a procedural manner. In terms of data, the task suites provide RGB images from the environment and wrist cameras, proprioception data, language task specifications, and PDDL scene descriptions.

4 Methodology

The LIBERO benchmark highlights task ordering as a significant challenge in lifelong learning. A key problem is developing a model that remains robust across various long-horizon task sequences, as real-world agents are likely to perform tasks with multiple steps.

To evaluate this robustness, we will conduct experiments as depicted in Fig. 2. We fine-tune π_0 on the LIBERO-10 dataset since this dataset consists of long-horizon manipulation tasks. Tasks can include interacting with multiple objects in the scene and requires multiple steps for successful completion. For example, available prompts include "put the black bowl in the bottom drawer of the cabinet and close it" and "put the yellow and white mug in the microwave and close it." After fine-tuning our π_0 policy, we deploy it on a Franka Emika Panda 7 degree-of-freedom(DoF) robot arm with a Schunk two-finger gripper. LIBERO-10 consists of 10 tasks where we select 10 random seeds per task and perform 10 rollouts for each seed. In total, we use 1000 rollouts. Another hyperparameter is the number of flow-matching steps, which we set to 8.

5 Experiments

The robot demonstrated a high success rate across the majority of tasks with a success rate of 80% or higher as shown in Fig. 7. In these successful tasks, π_0 is able to grasp both simple objects, such as boxes and cans that are simple to grasp, and complex objects, such as cups and pots with handles that are not symmetrical in shape and thus more challenging to grasp. Furthermore, π_0 exhibits good understanding of how these tasks require multiple steps. For instance, π_0 is able to perform consecutive pick & place actions for different objects. This observation of success also applies to tasks that require distinct actions as shown by the tasks that require pick & place actions before closing a cabinet. Fig. 3 illustrate some of these successful cases.

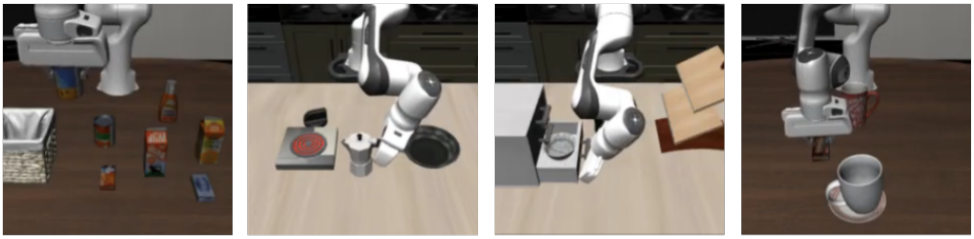

Figure 3: Examples of four different tasks successfully completed by the robot. From left to right: "place the alphabet soup and cream cheese box into the basket", "turn the stove on and place the pot", "place the lid into the cabinet and close it", "and place the cup onto the plate".

The robot successfully completed these tasks, including the less conventional task of placing the cup onto the plate. However, it faced significant challenges when dealing with duplicate objects. Notably, the results highlight a clear failure case for π_0, which achieved a success rate of only 20%, in stark contrast to the 80% or higher success rate observed for other tasks. While previous literature has mentioned attempted recovery without explicit training to do so, we do not observe any reasonable attempt to recover.

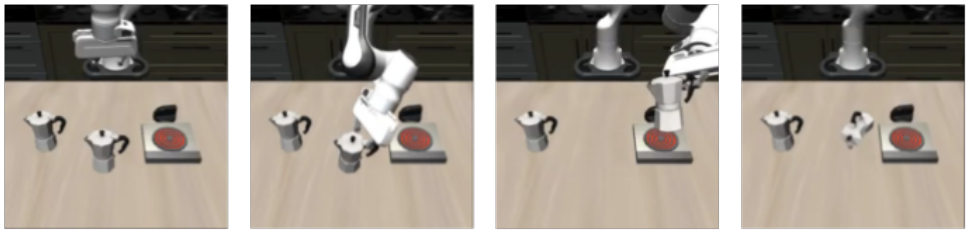

Figure 4: The robot nearly completes the task. It brings the pot close to the stove and then it loses track of what it was doing. The arm swings upwards and it drops the pot.

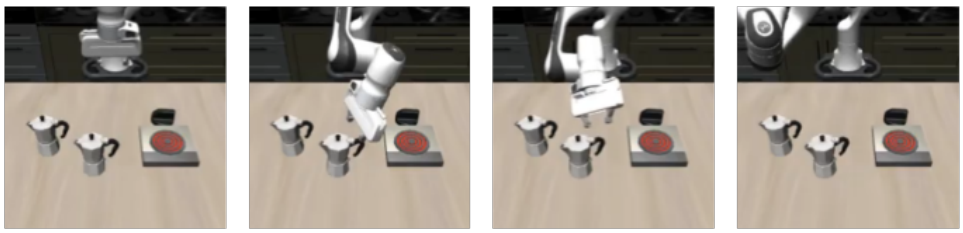

Figure 5: The robot failed to grab the pot on its first try, causing it to become confused and forget how to perform the task. This example shows a lack of error recovery.

However, π_0's performance on Task 7 was much worse with a success rate of about 20%. In Fig. 4, we observe that π_0 is able to grasp the pot but during the placing part, it begins to take erratic motion

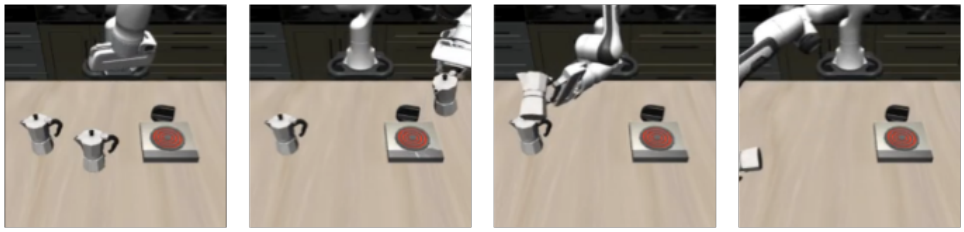

Figure 6: The robot partially completes the task, but it does not let go of the first pot before continuing on to the next pot. It then swings the pot, knocking over the other one and causing more confusion. It also loses track of both itself and the task it was doing.

as if it lost track of itself during placement. Fig. 5 illustrates another failure mode where π_0 fails to grasp the pot and instead of re-attempting, it begins to exhibit the same erratic motion in Fig. 5. Lastly, Fig. 6 shows where π_0 does successfully pick up the first pot, forgets to place it, and attempts to pick up the second pot. In doing so, it seems to become confused and exhibits erratic motions again.

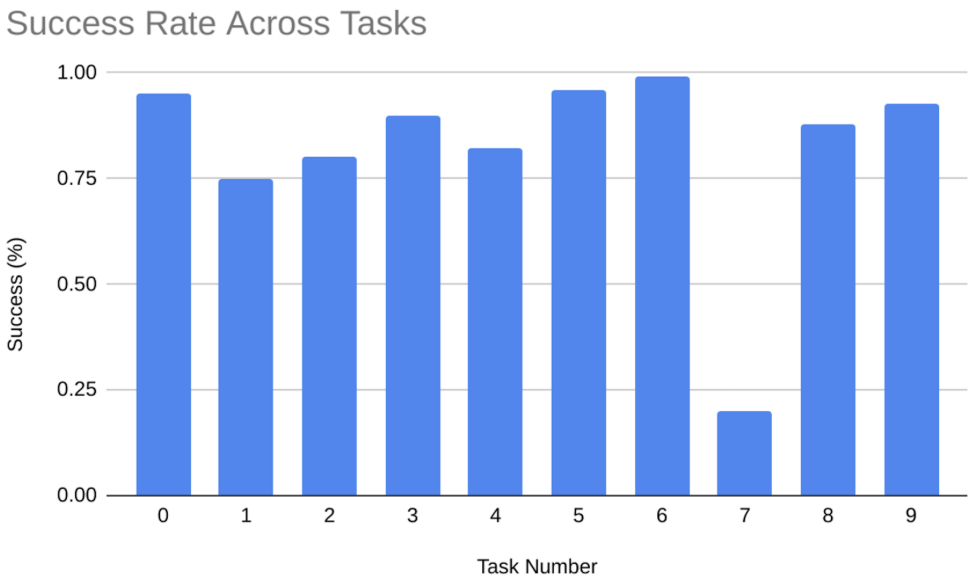

Figure 7: Success rate across different tasks.

6 Limitations

6.1 Memory

We observed that π_0 sometimes loses track of its objective, struggling to remember what it was doing. This limitation becomes particularly evident when multiple objects and tasks are involved, leading to confusion and incomplete executions. Since π_0 lacks memory, it cannot retain past actions or prioritize tasks effectively, resulting in inconsistent performance.

6.2 Spatial Reasoning

A significant challenge observed in π_0's performance is its difficulty in maintaining spatial awareness, particularly while in motion. The manipulator tends to lose track of its own position relative to the environment, leading to inaccuracies in task execution. This issue becomes particularly problematic

Figure 8: Wrist Images

when attempting to grasp objects, as π_0 struggles to correctly align its end-effector with the target, resulting in failed or imprecise grasps.

7 Future Work

7.1 Memory

To overcome the memory limitation, π_0 requires a mechanism to track its current objective. Potential solutions include:

- Implementing a short-term memory system to store and recall previous actions, allowing for better sequencing of operations.
- Developing algorithms that enable π_0 to determine the optimal execution order based on contextual factors such as resource availability and dependencies.

By integrating memory and improved task management strategies, π_0's performance can be significantly enhanced, making it more reliable in complex execution scenarios. Future research should also explore the trade-offs between memory integration and computational efficiency to achieve an optimal balance for real-world applications.

7.2 Spatial Reasoning

One of the key limitations identified in π_0's performance is its difficulty in maintaining spatial awareness, leading to errors in object manipulation and grasping. To address this, future research should focus on improving π_0's ability to accurately perceive and interpret its surroundings by aligning image and depth data more effectively.

Potential areas for improvement include:

- Fusion of Image and Depth Data: Implementing advanced sensor fusion techniques to combine RGB images with depth maps, point clouds, or other spatial representations. This would provide a more comprehensive understanding of the environment, enabling π_0 to detect objects and obstacles with greater accuracy.
- Point Cloud Processing for 3D Understanding: Utilizing point cloud data from LiDAR or structured light sensors to build more detailed 3D maps of the environment. This would improve π_0's ability to localize itself and plan precise movements, reducing grasping errors.

By aligning and optimizing the integration of depth and image data, π_0's spatial reasoning capabilities can be significantly enhanced. Future work should also explore how real-time processing of spatial data can be achieved without introducing excessive computational overhead, ensuring efficient and reliable operation in real-world environments.

8 Conclusion

Although π_0 is a highly capable generalist model, our evaluation revealed several failure cases that highlight its current limitations. Using the LIBERO dataset, we assessed π_0's performance on long-horizon tasks and systematically analyzed the scenarios where it struggled. Our findings suggest

that two primary factors contribute to these failures: memory constraints and deficiencies in spatial reasoning.

A significant observation was π_0's inability to retain and recall past actions, which led to inefficiencies in executing multi-step tasks. Due to its memorylessness, the model frequently lost track of its objectives, particularly in environments requiring sequential decision-making or task prioritization. This often resulted in incomplete actions or redundant operations, ultimately affecting its overall efficiency. Future work should explore the integration of short-term memory mechanisms or reinforcement learning techniques to enable π_0 to adapt and retain information across different stages of execution.

Additionally, we found that π_0 struggled with spatial awareness, particularly when dealing with multiple objects or when precise manipulation was required. The model exhibited difficulty in aligning image and depth data, leading to miscalculations in object positioning and failed grasp attempts. Addressing this issue requires enhancements in sensor fusion, real-time depth estimation, and 3D spatial modeling to improve π_0's perception and interaction capabilities in complex environments.

References

[1] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[4] Embodiment Collaboration. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[5] Michael Crawshaw. Multi-task learning with deep neural networks: A survey, 2020.

[6] Alexander Khazatsky et al. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[7] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[8] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[9] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[10] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2024.