$\pi_{0.6}$ Model Card

Physical Intelligence

November 17, 2025

1 Introduction

We introduce $\pi_{0.6}$, our newest vision-language-action (VLA) model that builds on top of $\pi_{0.5}$ [2] and achieves stronger performance across tasks. $\pi_{0.6}$ preserves the hierarchical design of $\pi_{0.5}$, providing high-level subtask prediction and low-level action generation. It incorporates a few changes involving the pre-trained VLM backbone and prompt design (Section 2), as well as the training datasets (Section 3). Section 4 analyzes the improvement in performance of $\pi_{0.6}$ compared to $\pi_{0.5}$ on a range of tasks that require dexterity and generalization. These experiments evaluate each model out of the box, without finetuning. This model has also been adopted as the base model for $\pi^*_{0.6}$ [7] where it is further improved through real-world reinforcement learning.

2 Model Design

Similar to π_0 [1] and $\pi_{0.5}$, the $\pi_{0.6}$ architecture (Figure 1) generates action chunks based on both flow matching and tokenized discrete outputs. The vision-language backbone is initialized from the Gemma3 4B model [6], and the "action expert" has the same number of layers as the backbone and consists of about 860M parameters. During pre-training, the model receives up to four images as input, each having resolution 448×448, corresponding to a base camera, up to two wrist cameras, and an optional backward camera for mobile manipulators. The image tokens after the vision encoder are concatenated with the tokenized language prompt and tokenized proprioceptive states. We keep bidirectional attention among all of the image tokens (as in $\pi_{0.5}$) but use causal attention among the text tokens. Action tokens fed into the action expert use bidirectional attention. The model is trained with Knowledge Insulation [3]: the vision-language backbone predicts FAST action tokens [5] and co-training examples, such as multi-modal web data. The action expert predicts continuous actions, and the gradient from the action expert does not flow back to the main VLM backbone. In addition to the language command, $\pi_{0.6}$ can optionally take in conditioning metadata in the prompt that further modulates how the task is performed. With 5 denoising steps and 3 camera inputs, $\pi_{0.6}$ takes 63ms to produce an action chunk on a single H100 GPU.

3 Training data

$\pi_{0.6}$ largely inherits the training data composition used in $\pi_{0.5}$, which consists of cross-embodiment data collected in-house and external data sources, diverse mobile and non-mobile data collected in home environments, high-level subtask prediction, and multi-modal web datasets, including bounding box and keypoint prediction.

In previous models, we found additional task-specific fine-tuning using curated high-quality data, referred to as post-training, to be highly effective and sometimes necessary to achieve good performance. $\pi_{0.6}$ achieves significantly stronger performance across tasks without task-specific fine-tuning thanks to the diverse training data and rich metadata conditioning (Section 4).

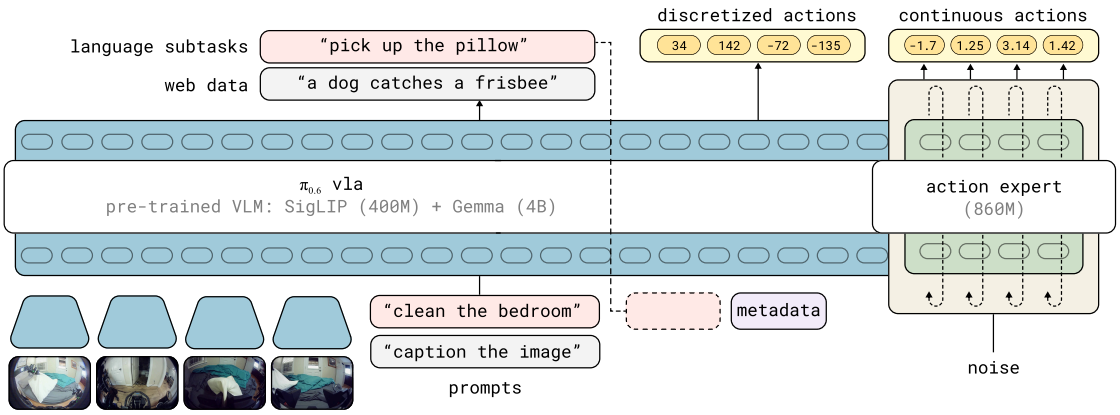

Figure 1: $\pi_{0.6}$ model architecture.

4 Evaluations

We present evaluation results comparing $\pi_{0.6}$ with the improved version of $\pi_{0.5}$ trained with Knowledge Insulation (open-sourced at openpi [4]). Neither models underwent task-specific fine-tuning. We refer to such evaluation setting where the models are not post-trained to perform well on specific task as out-of-the-box evaluation. The evaluation metrics include task success rate or progress, and throughput defined as the number of successes per hour. Standard error is shown through the error bars in the figures.

Static tasks. Figure 2 shows the results on tasks including shirt folding (shirt initially flat on the table), laundry folding (T-shirts and shorts initially in a basket), box assembly, and table bussing, performed with static robot platforms. These tasks were introduced in the π_0 paper. Across tasks, $\pi_{0.6}$ shows significant improvement over $\pi_{0.5}$ in speed and often success rates. The biggest differences lie in laundry folding and box assembly — previously these two tasks require fine-tuning with high-quality data to achieve non-zero success rates. $\pi_{0.6}$ can out-of-the-box fold laundry reliably, and fully assemble the box 20% of the time.

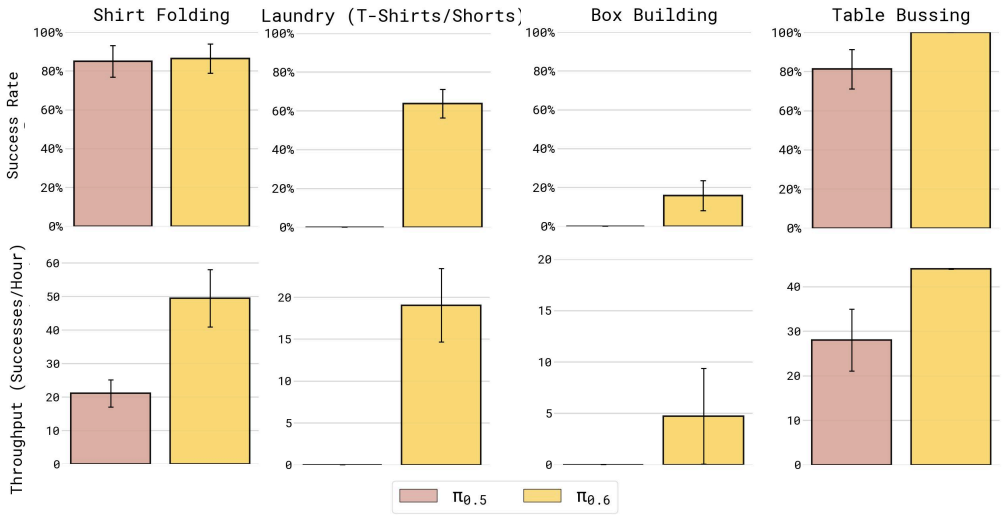

Figure 2: Out-of-the-box evaluation results in static tasks. $\pi_{0.6}$ improves speed in all tasks and is able to fold laundry consistently and sometimes fully assemble the box.

Mobile tasks. Figure 3 shows the results on four tasks performed with mobile bimanual robots: picking up laundry and placing it into a basket, tidying bed, putting dishes in the sink, putting items into the drawer. They are the main evaluation tasks in the $\pi_{0.5}$ paper. Across tasks, $\pi_{0.6}$ improves throughput over $\pi_{0.5}$ when the average task progress is saturated or otherwise improves both performance and throughput.

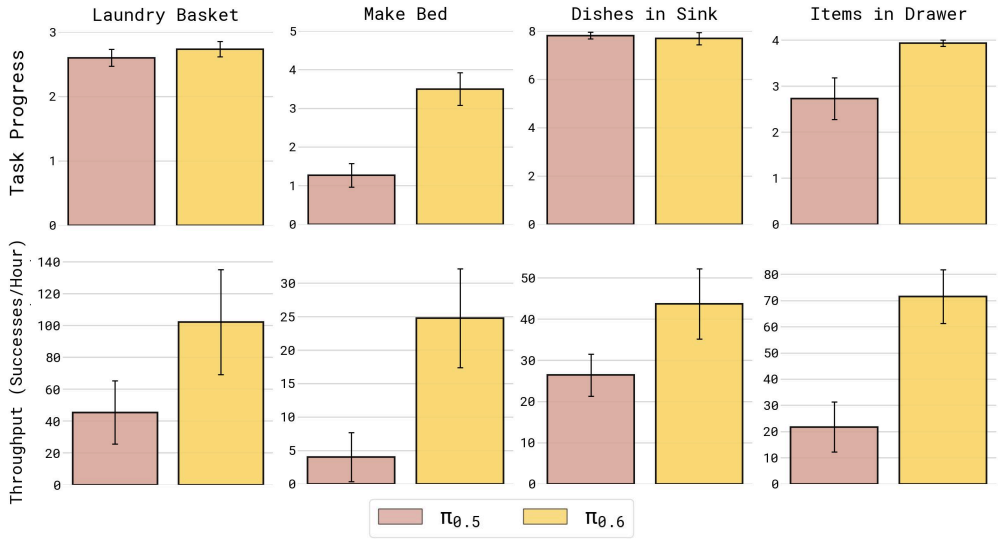

Figure 3: Out-of-the-box evaluation results in mobile tasks. The highest y-tick value in task progress is the maximum possible value for the task. $\pi_{0.6}$ improves speed in all tasks and also the average task progress if it is not saturated yet.

Generalization tasks. Figure 4 shows the results on a series of mobile and static tasks that require either within-distribution or out-of-distribution generalization in terms of language following (e.g., "pick up the third fruit from the left" or "move to where the fresh milk is kept") and novel skills (e.g., "wipe the spill with the bread" or "hang the shorts into oven handle"). The majority of the language instructions and the objects used during evaluation are not seen in training . Each of the four suite of tasks involves three levels of difficulty and a total of 12 to 18 instructions. Across settings, $\pi_{0.6}$ demonstrates healthy improvements over $\pi_{0.5}$. The mobile settings are generally more challenging as the tasks are often longer in horizon and the environments contain more task-irrelevant distractors.

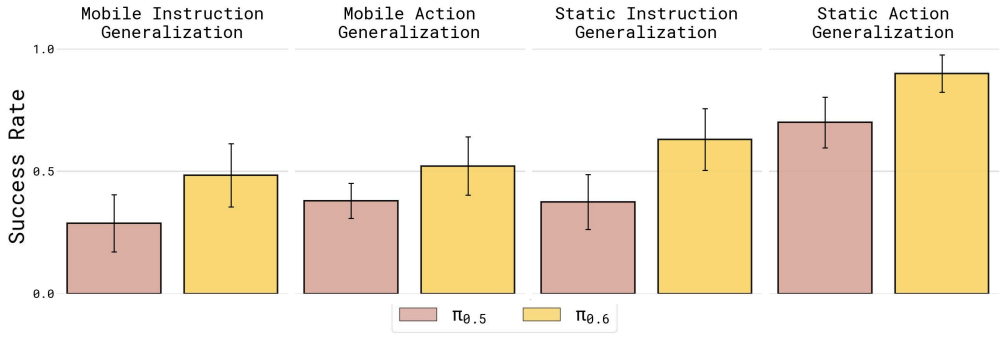

Figure 4: Out-of-the-box evaluation results in generalization-focused tasks. $\pi_{0.6}$ shows healthy improvement across settings over $\pi_{0.5}$.

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. In *Robotics: Science and Systems*, 2024.

[2] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y Galliker, et al. $\pi_{0.5}$: a Vision-Language-Action Model with Open-World Generalization. In *9th Annual Conference on Robot Learning*, 2025.

[3] Danny Driess, Jost Tobias Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Z Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge Insulating Vision-Language-Action Models: Train Fast, Run Fast, Generalize Better. In *NeurIPS*, 2025.

[4] Physical Intelligence. Openpi, 2025. URL https://github.com/Physical-Intelligence/openpi.

[5] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. *Robotics: Science and Systems*, 2025.

[6] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[7] Physical Intelligence team. $\pi^*_{0.6}$: a VLA That Learns From Experience. 2025.