
APO: Human-assisted Robotic Policy Refinement via Action Preference Optimization

(Supplementary Material)

Anonymous Author(s)

Affiliation

Address

email

1 Supplementary Video

2 In this work, we propose the action preference optimization method to correct interaction failure
3 and achieve stable optimization for VLA models. In the supplementary video, we illustrate our
4 human-assisted interaction trajectories collection process as demonstrated in Figure 1. We also
5 provide comparison videos against other methods, highlighting the effectiveness of our approach in
6 both real-world and simulation scenarios.



Figure 1: The demonstration of our human-assisted interaction trajectory.

2 Human-assisted Collaboration Deployment

8 In this work, we propose a human-assisted collaboration deployment framework to support reliable
9 deployment and interaction trajectory collection. The **blue block** in Figure 1 illustrates the initial
10 deployment of the base policy for autonomous environment interaction. However, the base policy
11 is trained solely on expert demonstrations. When its predicted action causes failures, this model
12 struggles to recover from these failure states, as shown in the **red block**. To address this, we provide
13 human intervention to manually adjust the robotic arm’s movements for failure correction, as shown
14 in the **green blocks**.

15 Through this human-assisted approach, we ensure reliable deployment of the model in manipulation
16 tasks. Furthermore, we annotate these interaction trajectories for subsequent preference learning.
17 Specifically, we designate the last 10 actions before human intervention as undesirable data (representing failure actions), while the remaining trajectories serve as desirable data.

3 Implementation Details

20 In our work, we build the utility function v as below to estimate the relative gain on the robotic data:

$$v(o, \hat{a}) = \begin{cases} \lambda_D \sigma(r_\theta(o, \hat{a}) - z_0) & \text{if } \hat{a} \sim \hat{a}_{\text{desirable}} \\ \lambda_U \sigma(z_0 - r_\theta(o, \hat{a})) & \text{if } \hat{a} \sim \hat{a}_{\text{undesirable}}, \end{cases} \quad (1)$$

21 where $z_0 = KL(\pi_\theta || \pi_{ref})$ to guide the model to learn from preference pair data while simultaneously
22 preserving knowledge acquired from prior models. We compute the KL-divergence z_0 by leveraging

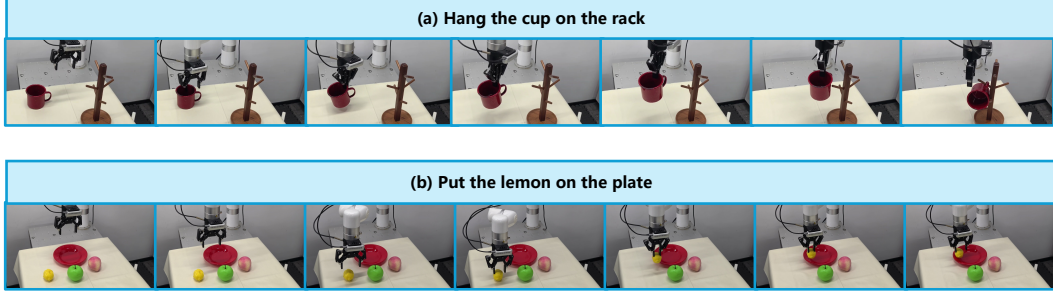


Figure 2: The demonstrations of real-world experiments.

the KTO [1] method, which leverages mismatched sample pairs for KL estimation. Further, we ignore the reject reward of the gripper action token to prevent erroneous rejection of the same gripper state.

Table 1: The results on π_0 -FAST model.

Methods	Square
Base policy	85%
Dagger	85%
TPO	90%
Ours	95%

Table 2: The results on real-world experiments.

Methods	Hang	Put
Base policy	70%	85%
Dagger	65%	85%
TPO	75%	80%
Ours	90%	100%

4 More Real-world Experiments

4.1 Generalization to various VLA models

In this section, we adopt our method to fine-tune the π_0 -FAST model. As shown in Table 1, the π_0 -FAST model achieves a higher success rate, benefiting from its action chunking prediction. Besides, our method could achieve consistent performance gains in real-world experiments. Because our method needs to decode to continuous action for adaptive reweighting, however, the π_0 -FAST model may fail to decode predicted action tokens into meaningful continuous actions, thus when the predicted action token sequences cannot be decoded to x , we would set the weight as 1 to promote the model focus on predicting correct action token sequences.

4.2 More real-world tasks

In this section, we provide two more real-world experiments as shown in Figure 2. For each task, we collect 100 expert demonstrations to train the base policy. Further, we deploy the base policy to interact with environments and collect 20 human-intervened trajectories. We mix the 20 human-intervened trajectories with 20 expert demonstrations for model preference optimization. As shown in Table 2, our method could achieve better performance compared with other behavior cloning and preference optimization methods.

References

- [1] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.