Preference Optimization for Continuous Robotic Control

Karthik Pythireddi [*1] **Taylor Tam** [*2] **Jonathan Lu** [*3]

Abstract

Reward design for continuous robotic control is often brittle because small specification errors can produce unintended behaviors. Preference-based reinforcement learning addresses this challenge by learning from pairwise trajectory comparisons rather than the manually designed rewards. We study whether preference optimization methods originally developed for the language model alignment transfer effectively to continuous robotic control. Across LIBERO, RoboCasa, Meta-world and Robosuite, we compare the RLHF, PPO, DPO and a likelyhood-regularized DPO-P style objective policies initialized from behavioral cloning or pretrained foundation models. We find that the standard DPO and PPO frequently destabilize competent policies and can lead to unexpected task collapse, while RLHF was generally more robust across environments, and DPOP performed well in Meta-World.. Overall our results show that that the continuous control is highly selective to the policy displacement and successful preference optimization in robotics requires objectives that better preserve reference-policy behavior.

1. Introduction and Objective

Designing reward functions for robotic control remains a central challenge in reinforcement learning. Primarily, small errors in reward specification often produce unintended or unsafe behaviors. Preference-based Reinforcement Learning (PbRL) addresses this limitation by learning from pairwise trajectory comparisons instead of depending solely on manually designed reward functions (Hejna & Sadigh, 2023).

[*]Equal contribution [1]Department of Electrical Engineering, Stanford University, California, USA [2]Department of Computer Science, Stanford University, California, USA [3]Department of Physics, Stanford University, California, USA. Correspondence to: Karthik Pythireddi <karthik9@stanford.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

In this paper, we examine whether preference optimization originally developed for language model alignment extends effectively to continuous robotic control (Rafailov et al., 2024; Christiano et al., 2023). We compare RLHF, PPO, DPO, and a likelihood-regularized DPOP-style objective across several robotic manipulation benchmarks, including LIBERO, RoboCasa, Meta-World, and RoboSuite. We find that standard DPO can fail catastrophically in continuous control, while RLHF and DPOP are considerably more robust. Taken together, these results indicate that alignment methods successful in token-based domains do not transfer reliably to robotics, where even small likelihood shifts can induce trajectory-level errors and break task completion.

2. Background and Related Work

An early influential work on the preference-based reinforcement learning was introduced by Christiano et al. (Christiano et al., 2023), where the goals were defined in terms of the human preferences between a pair of trajectory segments. In this framework, trajectory comparisons are collected and used to train a reward model that predicts which behaviors are preferred. They used this approach to solve games like Atari and were able to demonstrate that we can train complex novel behaviors and tasks without access to the reward function. The learned reward model is then used as a surrogate to optimize a policy using reinforcement learning.

In the context of LLMS, there has been a more recent work by Stanford University and CZ Biohub (Rafailov et al., 2024) which does not use the reward model rather uses a classification loss to extract the optimal policy in the closed form. This method is defined as Direct Preference Optimization (DPO), which is computationally friendly and can fine tune the model to align with the human preferences as well or better than the existing methods.

While preference optimization has been widely studied in the token-based domains such as language modeling, its behavior in continuous robotic control remains an open question. Robotic manipulation policies operate in continuous action spaces where small changes in action likelihood accumulate over time and we signifiacntly drift from the desired trajectory over time. As a results, methods that work well in the discrete token domain many not transfer directly to the continuous robotics control.

3. Tasks and Data

We evaluate preference optimization methods on various continuous control environments and tasks, including RoboSuite (**?**), RoboCasa (Nasiriany et al., 2026), LIBERO (Liu et al., 2023), and Meta-World (Yu et al., 2021). These cover both single-stage and long horizon tasks, as well as varying levels of task complexity, horizon length, and control difficulty: Meta-World provides standardized manipulation environments, such as reaching and object placement; RoboCasa (Nasiriany et al., 2026) introduces realistic household scenes with a pretrained foundation model; Robosuite offers physics-based tabletop robotic environments with rich action dynamics; Finally, LIBERO introduces more complex, multi-step coordination tasks that capture structured robotic workflows.

For each environment, we begin from a reference policy, either a behavioral cloning (BC) policy trained on demonstrations or a pretrained foundation model. We then construct preference datasets from pairs of trajectories, where one rollout is preferred based on task success or return. These pairs are then used to finetune the reference policy using RLHF, DPO, or DPOP style objectives, while we include PPO as a policy optimization baseline when feasible. We can therefore isolate how differing preference optimization methods affect policy performance, and, in turn, determine whether they improve behavior or destabilize trajectories that were already successful under the reference policy.

4. Approach

4.1. Problem Formulation

We begin with a reference policy π_{ref}, obtained either from behavioral cloning demonstrations or from a pretrained foundation model depending on the benchmark. For each task, we collect trajectory rollouts from the reference policy and construct a preference dataset of trajectory pairs (τ^+, τ^-), where τ^+ is preferred over τ^- based on task success or trajectory return. Our goal is to use these pairwise preferences to improve the policy while preserving successful behavior already present in the reference policy.

4.2. RLHF

In the RLHF setting, we first train a reward model r_ϕ from pairwise trajectory preferences following the approach of (Christiano et al., 2023). Given a preferred trajectory τ^+ and a dispreferred trajectory τ^-, we train the reward model using a Bradley–Terry objective:

$$\mathcal{L}_{\text{RM}} = -\log \frac{\exp(r_\phi(\tau^+))}{\exp(r_\phi(\tau^+)) + \exp(r_\phi(\tau^-))}. \tag{1}$$

The learned reward model is then used as a surrogate reward function for policy optimization. This separates preference modeling from policy improvement and forms the standard RLHF pipeline used in our experiments.

4.3. Reward-Weighted Regression (RWR)

Rather than optimizing RLHF's learned reward model with PPO, we use Reward-Weighted Regression (Peters & Schaal, 2007). This changes policy improvement to a weighted supervised learning problem. Given rollout samples $\{(s_i, a_i, r_i)\}$, the policy parameters are updated as:

$$\theta_{k+1} = \left(\Phi^\top W \Phi\right)^{-1} \Phi^\top W \mathbf{a} \tag{2}$$

where $\Phi = [\phi(s_1), \ldots, \phi(s_n)]^\top$ are state feature vectors, $\mathbf{a} = [a_1, \ldots, a_n]^\top$ are the collected actions, and $W = \text{diag}(\tilde{w}_1, \ldots, \tilde{w}_n)$ is a diagonal matrix of normalized exponentiated rewards:

$$\tilde{w}_i = \frac{\exp(r_i/\beta)}{\sum_{j=1}^{n} \exp(r_j/\beta)} \tag{3}$$

RWR avoids the instability of PPO under sparse rewards and limited compute, making it well-suited to the continuous action spaces present in robotic manipulation.

We choose RWR over PPO to optimize the policy in RLHF because PPO requires large batch sizes and dense reward signals for stable advantage estimation. These conditions are rarely met in sparse-reward robotic manipulation (Schulman et al., 2017). RWR reframes policy improvement as a weighted supervised learning problem with a closed-form update, which makes it more numerically stable with the compute we use in our experiments. This is shown by the complete failure of standalone PPO across our LIBERO tasks (Section 6.1).

4.4. PPO Baseline

We include Proximal Policy Optimization (PPO) (Schulman et al., 2017) as a standard policy optimization baseline and also use it as the optimizer in the RLHF pipeline after reward-model training. PPO updates the policy using a clipped surrogate objective:

$$\mathcal{L}_{\text{PPO}} = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{4}$$

where

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \tag{5}$$

and $\hat{A}_t$ denotes the estimated advantage. Including PPO helps distinguish the effect of preference-based learning from standard policy optimization.

4.5. DPO

Direct Preference Optimization (DPO) (Rafailov et al., 2024) removes the explicit reward-model stage and directly

optimizes the policy from preference comparisons. Given a preference pair (τ^+, τ^-) and reference policy π_{ref}, DPO encourages the policy to increase the relative likelihood of the preferred trajectory:

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \left[\log \frac{\pi_\theta(\tau^+)}{\pi_{\text{ref}}(\tau^+)} - \log \frac{\pi_\theta(\tau^-)}{\pi_{\text{ref}}(\tau^-)} \right] \right), \tag{6}$$

where β controls the sharpness of the preference update. This formulation is computationally attractive because it avoids explicit reward-model learning and the complicated fine-tuning that comes with it.

4.6. DPOP

To reduce harmful policy displacement (Pal et al., 2024), we also evaluate a likelihood-regularized DPOP-style objective. The objective modifies the standard DPO loss with a penalty on the log-likelihood of the preferred trajectory if it is outperformed by a baseline:

$$\begin{aligned} \mathcal{L}_{\text{DPOP}} &= \mathcal{L}_{\text{DPO}} \\ &\quad + \lambda \cdot \max\big(0,\ \log \pi_{\text{ref}}(\tau^+) - \log \pi_\theta(\tau^+)\big) \end{aligned} \tag{7}$$

where $\lambda > 0$ determines the strength of the regularization. The $\max(0, \cdot)$ hinge is one-sided. It will penalize the policy when $\log \pi_\theta(\tau^+) < \log \pi_{\text{ref}}(\tau^+)$, or when fine-tuning has *reduced* the likelihood of the preferred trajectory below the reference level. When the finetuned policy assigns higher likelihood to τ^+ than the reference, the penalty is zero so $\mathcal{L}_{\text{DPOP}} = \mathcal{L}_{\text{DPO}}$. DPOP addresses the likelihood displacement failure mode observed with standard DPO 4, while allowing the policy to improve in preferences where the reference is suboptimal.

5. Experimental Setup

5.1. Benchmarks

We evaluate on four continuous-control benchmarks spanning a range of task complexity and control difficulty.

LIBERO (Liu et al., 2023) is a long-horizon manipulation benchmark designed to evaluate compositional generalization and multi-step object interaction in household settings. We evaluate on four tasks from LIBERO_10 (Tasks 0, 3, 4, and 7).

RoboCasa (**?**) provides realistic household manipulation scenarios with richer scene layouts and task structure than standard tabletop environments. In our setting, it serves as a foundation-model benchmark using GROOT-N1.6 (Research, 2025) as the reference policy.

Meta-World (Yu et al., 2021) provides standardized tabletop manipulation tasks using a 7-DoF Sawyer arm. We focus on `bin-picking-v3` and `peg-insert-side-v3`, the latter being one of the most challenging tasks due to tight spatial tolerances.

RoboSuite offers physics-based tabletop environments with rich contact dynamics. We evaluate on the Lift task, which serves as a low-success baseline to test whether methods can recover useful behavior.

5.2. Preference Data Construction

For each environment, we collect trajectory rollouts from the reference policy and construct pairwise preference datasets. Each pair (τ^+, τ^-) is labeled using environment signals such as task success or cumulative return, with the higher-performing trajectory designated as preferred.

No human annotations are used. Preference datasets are constructed offline prior to fine-tuning and held fixed across all methods to ensure a controlled comparison. All methods are trained on identical preference datasets.

5.3. Environment-Specific Setup

5.3.1. LIBERO

We evaluate on four tasks from LIBERO_10 (Tasks 0, 3, 4, and 7) using a Franka Panda robot (7-DoF). Observations consist of RGB images (128×128) and proprioceptive state, while actions correspond to joint velocities and gripper control.

The reference policy is a behavioral cloning (BC) policy trained on expert demonstrations. Each method is evaluated over 20 episodes per task, with episode horizons ranging from 300 to 500 steps.

5.3.2. RoboCasa

RoboCasa experiments use the GROOT-N1.6 (3B) pre-trained foundation model (Research, 2025) as the reference policy, operating on a high-DoF humanoid embodiment (GR1). Observations consist of egocentric RGB inputs at 250×250 resolution.

We evaluate on four pick-and-place tasks involving cabinet interaction. The reference model is first adapted using supervised fine-tuning on expert demonstrations (1000 per task), followed by preference-based fine-tuning. Preference datasets consist of 200 trajectory pairs (50 per task). Each method is evaluated over 20 episodes per task.

5.3.3. Meta-World

We evaluate on `bin-picking-v3` and `peg-insert-side-v3`, using a 39-dimensional state observation space and a 4-dimensional continuous action space (Cartesian velocity + gripper control). The maximum episode length is 150 steps.

All policies are parameterized as MLPs with hidden dimension 256. Preference datasets are constructed at a ratio of 10 pairs per collected trajectory.

5.3.4. ROBOSUITE

We evaluate on the Lift task using a Panda robot with low-dimensional state observations. Rendering and camera observations are disabled, dense reward shaping is enabled, control frequency is set to 20 Hz, and random seed 0 is used throughout. Preferences are synthetically generated from true episodic returns and policies are evaluated deterministically using average return and success rate.

For RLHF, we use 60 training iterations with 64 episodes and 256 preference pairs per iteration, and 30 evaluation episodes. Policy, value, and reward networks are two-layer MLPs of size 256–256, with learning rates of 1×10^{-4}, 5×10^{-5}, and 1×10^{-4} respectively. The reward model is trained for 4 epochs with batch size 16. PPO uses $\gamma = 0.99$, GAE $\lambda = 0.95$, clip ratio 0.2, 4 PPO epochs, batch size 2048, and entropy coefficient 1×10^{-3}.

For DPO, we use 20 training iterations with 16 episodes and 64 preference pairs per iteration, and 5 evaluation episodes. The policy network is a 256–256 MLP with learning rate 3×10^{-4} and $\beta = 0.1$, trained for 6 DPO epochs with batch size 16. We warm-start with behavioral cloning using 40 trajectories (top 25% as demonstrations), 10 BC epochs, batch size 512, BC learning rate 1×10^{-3}, and a preference buffer of size 256.

5.4. Training Details

For DPO and DPOP, we use a learning rate of 1×10^{-5}, train for 20 epochs, and set the temperature parameter $\beta = 0.01$. For DPOP, the likelihood regularization strength is $\lambda = 0.5$.

For RLHF with Reward-Weighted Regression (RWR), we use a discount factor $\gamma = 0.99$ for computing returns during reward model training.

Batch sizes are kept small (e.g., 8) due to compute constraints. Experiments were conducted using a mix of local compute and GPU resources depending on the benchmark.

5.5. Evaluation Metrics

We report **task success rate** as the primary evaluation metric, defined as the fraction of evaluation rollouts in which the agent completes the task within the allotted horizon.

Evaluation is performed using deterministic rollouts of the fine-tuned policy. For LIBERO and RoboCasa, we evaluate using 20 episodes per task. For Meta-World, we evaluate using 50 goal configurations across 24 unseen seeds and report 95% confidence intervals.

Table 1. Overall Success Rate Across Robotic Benchmarks.

Environment	Model Used	Baseline	DPO	RLHF	PPO
LIBERO	BC policy	60.0%	65.0%	66.2%	0.0%
RoboCasa	Groot N1.6	52.5%	0.0%	1.2%	3.8%
Meta-World	BC policy	92.8%	0.0%	97.8%	N/A
RoboSuite	BC policy	0.0%	0.0%	6.7%	N/A

Table 2. Success Rate Comparison Across Manipulation Tasks of LIBERO

Task	Description	BC	DPO	RLHF	PPO
0	put both the alphabet soup and the can	65.0%	65.0%	70.0%	0.0%
3	put the black bowl in the bottom drawer	90.0%	95.0%	95.0%	0.0%
4	put the white mug on the left plate	60.0%	65.0%	50.0%	0.0%
7	put both the alphabet soup and the can	25.0%	35.0%	50.0%	0.0%
AVG		60.0%	65.0%	66.2%	0.0%

In addition to success rates, we qualitatively analyze rollout behavior to assess whether fine-tuning preserves or destabilizes the structure of the reference policy.

6. Results

We evaluated preference optimization methods on multiple robotic manipulation benchmarks, such as LIBERO, RoboCasa, Meta-World, and RoboSuite. Our comparisons evaluate how different fine-tuning methods affect policies that already exhibit some degree of competent behavior, either from behavioral cloning or from a pretrained foundation model. Across several environments, we find that standard DPO is the least reliable method: while RLHF and DPOP often preserve or slightly improve baseline performance, standard DPO can catastrophically degrade control behavior in continuous-action settings.

Table 1 summarizes the overall success rates across the robotic benchmarks evaluated in this study.

6.1. LIBERO

We evaluated BC, DPO, RLHF, and PPO on four LIBERO manipulation tasks. These tasks involve multi-step object interaction and coordination, providing a useful measure of whether preference optimization can improve already competent behavioral cloning policies.

A rollout video was recorded for Task 7 demonstrating a PPO failure case. PPO failed under our training setup, likely due to GPU memory constraints that forced small batch sizes and made sparse-reward optimization ineffective in practice. RLHF and DPO both made modest improvements over the BC baseline, but the gains were limited and inconsistent between tasks. RLHF achieved the strongest average performance, while PPO completely failed. Overall, the

LIBERO results suggest that preference-based fine-tuning can sometimes help, but improvements remain modest when the starting BC policy is already reasonably strong.

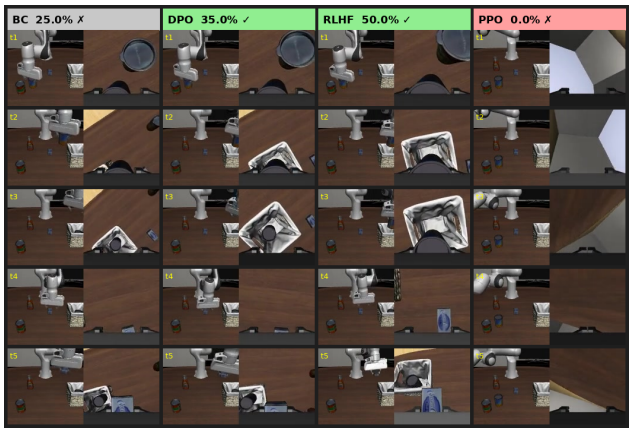

Figure 1. Task 7 rollout comparison across all methods. Green = successful episode, Red/Gray = failure. RLHF achieves the highest success rate (50%) while PPO fails entirely (0%).

6.2. RoboCasa

RoboCasa produced the clearest failure cases in our study. Using the GROOT-N1.6 foundation model as a baseline on a pick-and-place cabinet-closing task, the unmodified policy achieved a success rate of 52.5%. However, after fine-tuning, standard DPO collapsed completely with 0.0% success. RLHF and PPO also performed substantially worse than the baseline, achieving only 1.2% and 3.8% success, respectively. Therefore, when the initial policy already captures the correct action structure, preference optimization can easily become destructive rather than beneficial. In particular, standard DPO failed to preserve the successful behavior of the reference policy in this continuous-control setting.

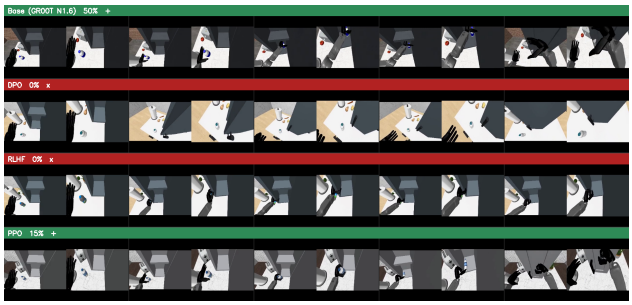

Figure 2. Humanoid Bottle to CabinetClose rollout comparison across all methods. Green = successful episode, Red/Gray = failure. RLHF achieves the highest success rate (52.5%) while DPO fails entirely (0%) and PPO and RWR has very small success.

6.3. Meta-World

Table 3. Success Rate and Mean Reward on Meta-World `bin-picking-v3` (24 seeds, 50 episodes each, 95% CI).

Method	Success Rate	Mean Reward
BC	$92.75\% \pm 1.47\%$	450.58 ± 17.73
DPO	$0.00\% \pm 0.00\%$	1.66 ± 0.05
DPOP	$97.08\% \pm 0.95\%$	433.96 ± 23.35
RLHF w/ RWR	$\mathbf{97.83\% \pm 0.82\%}$	$\mathbf{484.54 \pm 14.02}$

Table 4. Success Rate and Mean Reward on Meta-World `peg-insert-side-v3` (24 seeds, 50 episodes each, 95% CI).

Method	Success Rate	Mean Reward
BC	$82.58\% \pm 2.15\%$	597.54 ± 30.46
DPO	$0.00\% \pm 0.00\%$	1.88 ± 0.07
DPOP	$85.50\% \pm 1.99\%$	571.13 ± 32.41
RLHF w/ RWR	$\mathbf{87.67\% \pm 1.86\%}$	$\mathbf{587.49 \pm 34.88}$

Our main Meta-World experiment was conducted on the bin-picking-v3 and peg-insert-side-v3 tasks. In this environment the BC baseline already performed strongly, achieving respectively 92.8% and 82.58% success rates. However, when fine-tuned with standard DPO, performance collapsed to 0.0% in both tasks, indicating a complete loss of task-completing behavior. A variant of likelihood-regularized DPOP reached respective success rates of 97.08% and 85.50%. RLHF achieved the best performance among all methods with respective success rates of 97.83% and 87.67%.

Figure 3. Bin picking demonstrated a performance collapse on finetuned DPO, while RLHF preserved the baseline and DPOP remained stable.

This result is one of the strongest findings in our study. Our results suggest that the key issue may not be preference

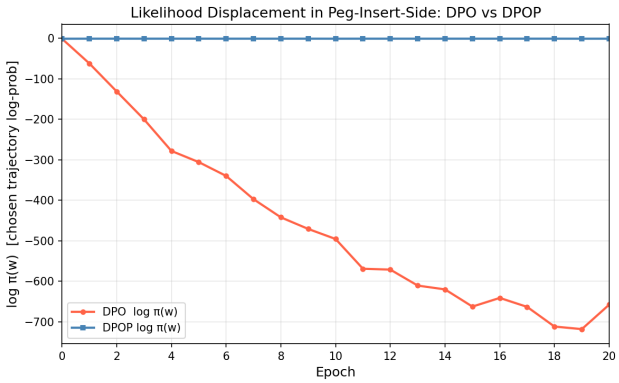

Figure 4. DPO drives the log-probability of chosen trajectories from near zero to below -700 over 20 epochs, while DPOP's likelihood-regularization penalty keeps it stable throughout. The same pattern holds on bin-picking-v3.

optimization itself, but rather the tendency of standard DPO to move the policy too far from successful reference behavior, a phenomenon known as likelihood displacement (Figure 4). In continuous control, even small shifts in action likelihood can alter the full trajectory and disrupt task execution. Our DPOP result supports this interpretation by explicitly demonstrating that constraining harmful policy displacement can prevent the collapse observed with standard DPO.

6.4. RoboSuite

We also evaluated methods in RoboSuite, a setting in which all methods performed at relatively low success rates: both the BC baseline and DPO achieved 0.0% success, while RLHF improved slightly to 6.67%. Although absolute success rates are low, the qualitative trend is consistent with the rest of our experiments: RLHF is more likely than standard DPO to preserve or recover usable control behavior, even when overall task performance remains limited.

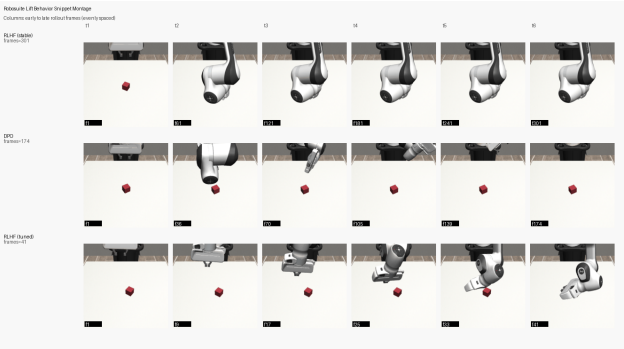

Figure 5. Robosuite Lift tasks across DPO, RLHF, and tuned RLHF.

Overall, these results show that preference optimization in robotics is highly sensitive to policy displacement. In language models, modest changes in sequence likelihood may still preserve useful behavior. In robotic control, however, even small changes in policy likelihood can alter the full action trajectory and cause task failure. Across our benchmarks, standard DPO is the least reliable method, while RLHF and DPOP are generally more robust.

7. Discussion

Our results highlight several important properties of preference optimization in continuous robotic control. When the reference policy is already strong, preference-based fine-tuning often yields modest gains. In such settings, demonstrations or pretraining may already capture most of the relevant task structure, leaving limited room for further improvement through preference supervision.

More importantly, our experiments show that preference optimization methods do not behave uniformly in continuous control. Across benchmarks, standard DPO was the least reliable method and, in multiple cases, catastrophically degraded policies that were already capable of completing the task. This failure mode was especially pronounced in RoboCasa and Meta-World, where relatively small deviations from the reference policy were sufficient to destroy task-completing behavior. These findings suggest that the main challenge is not simply learning from preferences, but doing so without destabilizing the action structure of an already competent controller.

This sensitivity appears to be a distinctive feature of continuous control. In language modeling, moderate shifts in sequence likelihood may still preserve semantically useful outputs. However, in robotic manipulation, small changes in action likelihood can compound over time, alter the resulting trajectory, and break task execution. This helps explain why methods that are effective in token-based alignment settings may transfer poorly when applied directly to continuous control policies.

Our results suggest that the core issue is not preference optimization itself, but rather the absence of sufficient constraints on policy updates. In Meta-World, standard DPO collapsed completely, whereas the likelihood-regularized DPOP variant remained stable and preserved strong performance. This indicates that explicitly limiting harmful policy displacement is critical for maintaining control fidelity in continuous action spaces. RLHF was also generally more robust than standard DPO, although its gains were often modest and benchmark-dependent. PPO performed poorly in our setting, particularly under sparse-reward conditions and limited compute, where optimization was unstable and often ineffective.

Taken together, these findings suggest that successful pref-

erence optimization in robotics requires objectives that are sensitive not only to preference satisfaction, but also to the preservation of the trajectory distribution of the reference policy. Methods that ignore this constraint risk improving the preference objective while degrading the behavior that matters for task completion. Interestingly, Table 3 and Table 4 suggest that methods which improve task success rate like DPOP and RLHF do not always recover the execution efficiency of the original BC policy. DPOP improves success on both Meta-World tasks yet achieves lower mean reward than BC. This suggests that preference optimization may shift behavior toward task completion without preserving the smooth, expert-like motion captured in the original demonstrations. We will leave a study of this efficiency-success trade-off to future work.

8. Future Work

There are several promising directions for future work. First, our experiments should be expanded across additional random seeds and a broader set of tasks in order to better characterize the consistency and statistical robustness of the observed trends. In particular, it would be valuable to determine whether the collapse of standard DPO is systematic across continuous-control benchmarks or concentrated in environments with especially strong reference policies.

Second, future work should examine the effect of noisy or imperfect preference labels. Our current study primarily focuses on whether preference optimization methods preserve or degrade competent behavior under clean preference signals. A natural next step is to test how robust RLHF, DPO, and DPOP are when preferences are ambiguous, inconsistent, or only weakly informative, as would be expected in realistic human-feedback settings.

Third, the strong performance difference between standard DPO and DPOP suggests that regularization is a central design axis for preference optimization in robotics. Further work should investigate alternative trust-region and likelihood-constrained objectives that explicitly bound harmful policy displacement while still allowing beneficial adaptation. More broadly, developing objectives tailored to continuous action spaces may prove more effective than directly importing alignment methods from language modeling.

Finally, an important longer-term direction is to evaluate these methods beyond simulation, including on longer-horizon manipulation problems and real robotic platforms. Simulation results provide strong evidence that continuous control presents a distinct alignment challenge, but real-world deployment would test whether these methods can preserve stability, robustness, and task fidelity under the additional noise and uncertainty present in physical systems.

9. Conclusion

We studied whether the preference optimization methods developed for language model alignment transfer effectively to continuous robotic control. Across four robotic benchmarks – LIBERO, RoboCasa, Meta-World and Robosuite. We compared RLHF, DPO, PPO and a likelihood regularized DPOP-P style objective starting from either behavioral cloning policies or pretrained foundation models.

Our Results show that the preference optimization does not transfer uniformly to the continuous robotic control. In particular standard DPO was often unstable and, in several cases, catastrophically degraded policies that were already competent.By contrast, RLHF and likelihood-regularized DPOP have seen to perform well. Finally, our findings suggest that the successful preference optimization in continuous control requires objectives that explicitly limit policy divergence from the intended behavior.

Author Contributions

Karthik Pythireddi: Conceived the project idea in collaboration with the project mentor (Rahul) and our teammates liked this idea. Led the development of the project proposal, milestone report, poster, and final report. Conducted experiments on the LIBERO and RoboCasa benchmarks. Coordinated discussions with the mentor during office hours to address open questions and guide the project direction.
https://github.com/karthikpythireddi/policylearning_from_trajectory_preferences
https://github.com/karthikpythireddi/cs234_robocasa
https://github.com/karthikpythireddi/Isaac-GR00T
Jonathan Lu: Led the Meta-World experiments, including implementation and evaluation of all methods on bin-picking-v3 and peg-insert-side-v3. Independently implemented the DPOP and conducted the likelihood displacement analysis shown in Figure 4. Identified the efficiency-success trade-off between mean reward and task success rate across preference optimization methods, as reported in Table 3 and Table 4. Code available at https://github.com/fozziejonathan/cs234-finalproject
Taylor Tam: Led the RoboSuite Lift experiments end-to-end, including implementation, training, evaluation, and qualitative rollout analysis. Established that in this low-success setting, both the BC baseline and standard DPO remained at 0.0% success, while RLHF recovered nonzero task-completing behavior at 6.67% success, reinforcing the paper's broader finding that RLHF is more robust than standard DPO in continuous-control robotics. In parallel,

drafted major project deliverables, including the proposal, milestone report, poster, and final report.

Supplemental Materials

Video demonstrations of our policies across environments are available at https://drive.google.com/drive/folders/1Fp4hVoTv0v7cIIi_3s_G7QAJhWdW21Pl?usp=sharing

References

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences, 2023. URL https://arxiv.org/abs/1706.03741.

Hejna, J. and Sadigh, D. Inverse preference learning: Preference-based rl without a reward function, 2023. URL https://arxiv.org/abs/2305.15363.

Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning, 2023. URL https://arxiv.org/abs/2306.03310.

Nasiriany, S., Nasiriany, S., Maddukuri, A., and Zhu, Y. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots, 2026. URL https://arxiv.org/abs/2603.04356.

Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. Smaug: Fixing failure modes of preference optimisation with dpo-positive, 2024. URL https://arxiv.org/abs/2402.13228.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pp. 745–750, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273590. URL https://doi.org/10.1145/1273496.1273590.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Research, G. . N. Gr00t n1.5: An improved foundation model for generalist humanoid robots, 2025. URL https://research.nvidia.com/labs/gear/gr00t-n1_5/. Technical report.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Yu, T., Quillen, D., He, Z., Julian, R., Narayan, A., Shively, H., Bellathur, A., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. URL https://arxiv.org/abs/1910.10897.