Humanoid Whole-Body Manipulation

Bhaswanth Ayapilla[1,2] **Kshitij Madhav Bhat**[1,2]
[1]Robotics Institute, [2]School of Computer Science
Carnegie Mellon University
{bayapill, kmadhavb}@andrew.cmu.edu

Abstract: Humanoid robots hold promise for versatile locomotion and manipulation in human environments, yet learning to control their high-dimensional dynamics remains challenging. Recent benchmarks show that state-of-the-art reinforcement learning algorithms require millions of samples and struggle on complex tasks. We investigate FlowRL [1], a flow-based off-policy actor-critic method, on locomotion and manipulation tasks from HumanoidBench [2]. We propose two modifications: (1) replacing FlowRL's auxiliary behavior-optimal critic with an AWR-style exponential Q-weighting that eliminates two networks and a hyperparameter, and (2) upgrading all network components to a SimBa-inspired residual architecture [3] for more stable training at greater depth. Our experiments demonstrate that these modifications maintain or improve performance across humanoid locomotion tasks while substantially reducing algorithmic complexity. Please refer to the Project Website for more details.

Keywords: Humanoid, Robots, Reinforcement Learning, Flow Matching

1 Introduction

The recently introduced HumanoidBench [2] provides a comprehensive evaluation of RL algorithms on 27 humanoid tasks, revealing that state-of-the-art methods such as DreamerV3 [4], TD-MPC2 [5], SAC [6], and PPO [7] struggle significantly on high-dimensional control, particularly on tasks requiring coordinated locomotion and manipulation.

FlowRL [1] is a recently proposed flow-based actor-critic method that parameterizes the policy as a state-conditioned velocity field, generating actions via ODE integration from Gaussian noise. It derives a constrained policy search objective that jointly maximizes Q-values while bounding the Wasserstein-2 distance to a behavior-optimal policy implicit in the replay buffer, achieving competitive performance on DMControl and HumanoidBench with only a single flow step. However, FlowRL's constraint relies on two auxiliary networks, a behavior-optimal Q-network and V-network estimated via expectile regression, adding algorithmic complexity and an additional hyperparameter τ.

In this work, we investigate two modifications to FlowRL on HumanoidBench locomotion tasks. First, motivated by the AWR/AWAC family of methods [8, 9], we show that the behavior-optimal critic can be replaced by exponential reweighting of buffer actions using only the current Q-function, since the intractable partition function cancels in gradients. Second, motivated by SimBa [3], we replace the sequential network blocks with pre-norm residual feedforward units initialized near the identity, providing a simplicity bias that stabilizes training as network depth increases. Together, these modifications reduce the number of required networks from five to three while preserving the core flow-based policy improvement guarantee.

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

2 Environment

We use HumanoidBench, a MuJoCo-based simulation environment featuring a *Unitree G1* humanoid robot with two dexterous Shadow Hands. The robot has:

1. Observation space: 151D (51D body proprioception + 50D per hand)
2. Action space: 61D position control (19D body + 21D per hand) at 50 Hz
3. Physics: MuJoCo MJX with realistic contact dynamics, runs at $\sim$ 1000 FPS

3 Reward Function

We use *Unitree G1* humanoid robot for the `g1-window-v0` task, a whole-body manipulation scenario where the robot must grasp a window wiping tool and keep its tip parallel to a window by following a prescribed vertical velocity. Success requires precise whole-body coordination to maintain tool-surface contact while simultaneously adjusting posture to execute the wiping motion. The reward function drives this behavior through two primary components: a manipulation term that rewards the robot for maintaining a stable upright stance, keeping its hands close to the tool, and tracking a target vertical tool velocity of 0.5 m/s; and a strict contact term that is only active when the tool touches the glass, incentivizing the robot to keep five distinct contact points on the wiper flush against the window pane. **Please refer to Table 1 in the Appendix for the details of the reward function.**

4 Method

We implement FlowRL, a recent flow-based off-policy actor-critic method, and propose two modifications: an AWR-style critic weighting that eliminates two auxiliary networks, and a SimBa-inspired residual architecture for all network components.

4.1 Modification 1: FlowRL

Policy representation. FlowRL parameterizes the policy π_θ as a state-conditioned velocity field $v_\theta(t, s, a^t)$. Actions are generated by integrating the induced ODE from Gaussian noise $a^0 \sim \mathcal{N}(0, I)$:

$$a^1 = a^0 + \int_0^1 v_\theta(t, s, a^t)\, dt, \qquad a^t = ta^1 + (1-t)a^0. \tag{1}$$

The conditional target velocity simplifies to $u(t, a^t|a^1) = a^1 - a^0$, so the standard CFM loss becomes a regression onto buffer actions.

Constrained policy search. Policy improvement is framed as maximizing expected return while bounding the Wasserstein-2 distance to the *behavior-optimal* policy π_{β^*}— the best historical policy implicit in the replay buffer—where $Q^*(s,a) \geq Q^{\pi_{\beta^*}}(s,a) \geq Q^{\pi_\beta}(s,a)$. Via a tractable W_2 upper bound and Lagrangian relaxation, the practical objective is

$$\mathcal{L}_{\text{FlowRL}}(\theta) = \mathbb{E}_{s,a\sim\mathcal{D},\, a'\sim\pi_\theta}\Big[-Q^{\pi_\theta}(s,a') + \lambda\, f\big(Q^{\pi_{\beta^*}}(s,a) - Q^{\pi_\theta}(s,a')\big)\|v_\theta(s,a^t,t) - (a-a^0)\|^2\Big], \tag{2}$$

where $f(x) = \max(x, 0)$ up-weights buffer actions that outperform the current policy and λ is a fixed Lagrange multiplier.

Critic setup. Evaluating $f(\cdot)$ requires *two* separate critic systems: (i) a twin Q-network Q^{π_θ} trained with standard Bellman updates, and (ii) a twin $Q^{\pi_{\beta^*}}/V^{\pi_{\beta^*}}$ pair trained via expectile regression [10] to approximate the behavior-optimal value without explicit maximization over the

buffer:

$$\arg\min_{V^{\pi_{\beta^*}}} \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^\tau\big(Q^{\pi_{\beta^*}}(s,a) - V^{\pi_{\beta^*}}(s)\big)\right], \tag{3}$$

$$\arg\min_{Q^{\pi_{\beta^*}}} \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\big(r + \gamma V^{\pi_{\beta^*}}(s') - Q^{\pi_{\beta^*}}(s,a)\big)^2\right], \tag{4}$$

where $L_2^\tau(x) = |\tau - 1(x < 0)|x^2$ and $\tau \in (0.5, 1)$ is the expectile hyperparameter.

4.2 Modification 2: Eliminating $Q^{\pi_{\beta^*}}$ via AWR-Style Weighting (SNQF)

Motivation. FlowRL's constraint weight $f(Q^{\pi_{\beta^*}} - Q^{\pi_\theta})$ couples the exploitation signal to a costly auxiliary critic ($Q^{\pi_{\beta^*}}$, $V^{\pi_{\beta^*}}$, expectile hyperparameter τ). We observe that this machinery can be replaced by a *single* exponential reweighting of buffer actions using only the current critic Q^{π_θ}, following the AWR/AWAC/CRR family of methods [8, 9, 11].

Q-tilted behavior distribution. Define the per-state Q-tilted distribution

$$\tilde{\pi}_\alpha(a|s) = \frac{\pi_\beta(a|s)\exp\big(Q^{\pi_\theta}(s,a)/\alpha\big)}{Z_\alpha(s)}, \quad Z_\alpha(s) = \int_{\mathcal{A}} \pi_\beta(a|s)\exp\big(Q^{\pi_\theta}(s,a)/\alpha\big)da. \tag{5}$$

This reweights buffer actions by their exponentiated Q-values: high-Q actions are amplified, low-Q actions are suppressed, recovering the role of π_{β^*} without a separate network.

SNQF objective. These observations yield the *Self-Normalizing Q-weighted Flow* (SNQF) loss, which replaces Eq. (2):

$$\mathcal{L}_{\text{SNQF}}(\theta) = \underbrace{-\mathbb{E}_{s\sim\mathcal{D},\, a'\sim\pi_\theta}\big[Q^{\pi_\theta}(s,a')\big]}_{\text{Q-maximization}} + \lambda \underbrace{\mathbb{E}_{(s,a)\sim\mathcal{D},\, t, a^0}\Big[\exp\big(\bar{Q}(s,a)/\alpha\big)\cdot\|v_\theta(s,a^t,t) - (a - a^0)\|^2\Big]}_{\text{Q-weighted CFM (exploitation)}} \tag{6}$$

where gradients are stopped through $\exp(\bar{Q}/\alpha)$. SNQF requires only the standard twin Q-network; $Q^{\pi_{\beta^*}}$, $V^{\pi_{\beta^*}}$, and the expectile hyperparameter τ are eliminated entirely.

Compared to FlowRL's hard indicator weight f, SNQF's soft exponential weight provides smoother gradients via temperature α. The weight also depends solely on the buffer action a, decoupling exploitation from the current policy's sampled action a' and simplifying the optimization.

5 Experimental Results

The random policy baseline achieved a mean return of 2.87 ± 0.66 over 100 episodes on the `g1-window-v0` task. As expected for random actions, the performance remains consistent with no learning trend.

Figure 1: Episode returns for `g1-window-v0` task with random policy

5.1 Learning Curves

We evaluate two reinforcement learning approaches on the `g1-window-v0` task: PPO as the on-policy policy-gradient baseline and SAC as the off-policy Q-function-based baseline. We also include a random policy baseline (mean return of 2.87 ± 0.66 over 100 episodes) as a lower bound reference. All results report mean episode return versus environment steps. The PPO and SAC results are evaluated over three random seeds.

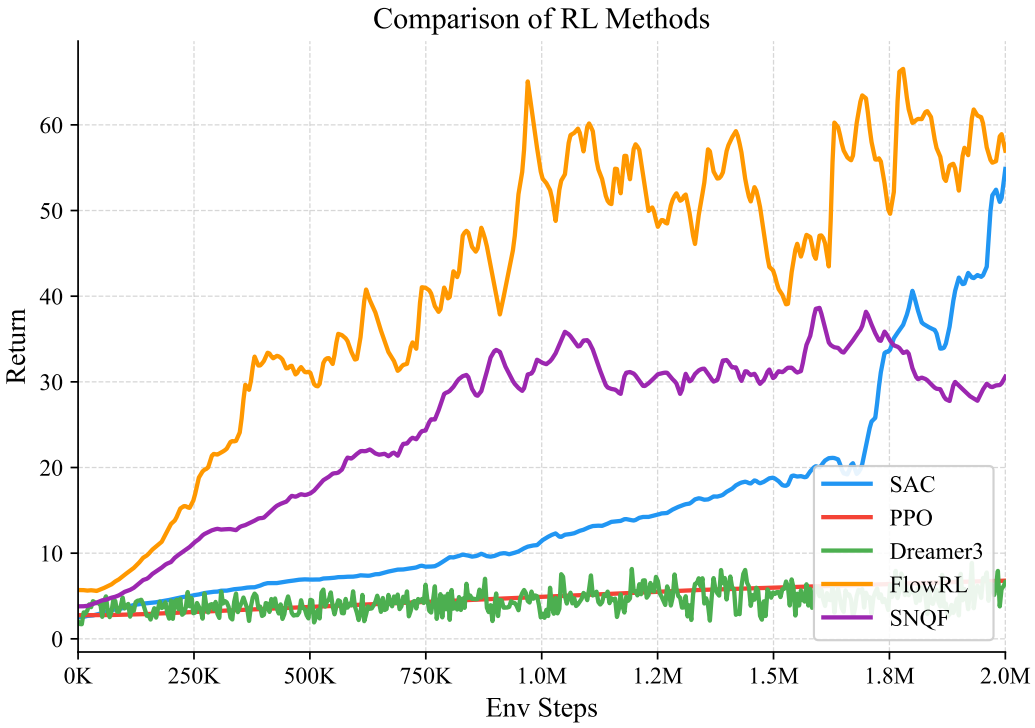

Figure 2: Evaluation returns for `g1-window-v0` task with all the methods

5.2 Analysis of Learning Curves

Figure 2 presents the episode returns for PPO, SAC, and the random policy baseline across three independent seeds each. The two methods tell fundamentally different stories about how and whether an agent can make progress on the `g1-window-v0` task.

5.2.1 PPO

PPO exhibits slow but remarkably consistent learning throughout the entire 20M step training budget. All three seeds start near the random baseline and climb steadily, converging to returns of 19-21 by the end of training, with inter-seed variance never exceeding 2-3 return units at any point (Figure 3). This reproducibility is a direct consequence of PPO's conservative on-policy updates, which constrain the policy from making large jumps between iterations.

The learning curve shows no clear sign of plateauing at 20M steps, though the rate of improvement slows noticeably after 14M steps. Gains of less than 2 return units over the final 6M steps suggest that PPO is approaching the limits of what its current policy parameterization can achieve without further architectural or algorithmic changes.

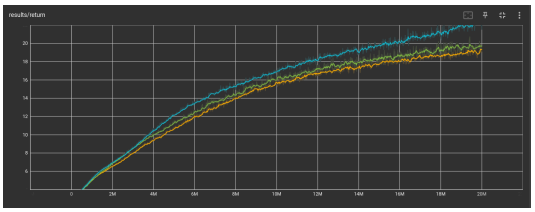

Figure 3: Episode returns for `g1-window-v0` task with PPO on 3 seeds

5.2.2 SAC

As shown in Figure 4, all three seeds rise sharply in the first 500k steps, already surpassing PPO's 5M-step performance before SAC has even used a quarter of PPO's sample budget. Two seeds reach peak returns of 60-75 around 2-2.5M steps, which is 3-4$\times$ higher than PPO's final performance. This reflects SAC's key advantage: the replay buffer enables aggressive reuse of past experience, and entropy regularization drives the broad exploration needed to discover the coordinated whole-body behaviors this task requires. However, none of the three seeds sustain these peak returns. The pink seed oscillates violently between 20 and 75 from 2M steps onward. The grey seed collapses almost entirely after 4.5M steps, falling to $\sim$10, barely above the random baseline, and never recovers. The purple seed is the most stable, maintaining returns in the 30-45 range through 5M steps, but with high variance throughout.

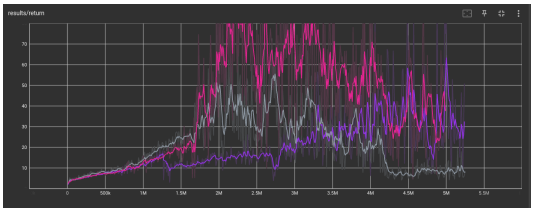

Figure 4: Episode returns for `g1-window-v0` task with SAC on 3 seeds

Taken together, SAC is far more sample-efficient but fails to consolidate its gains into a stable policy, while PPO is slow but monotonic and reproducible.

5.2.3 TD-MPC2

TD-MPC2 was evaluated on the `g1-window-v0` task across three random seeds up to 1.2M environment steps. Two of the three seeds (blue) exhibit meaningful learning, as shown in Figure 5, rising from near the random baseline ($\sim$5) to peak returns of 45–57 by 1.1M steps, though with substantial variance throughout. The third seed (orange) remains largely flat near the random baseline for the entire training run, never exceeding returns of $\sim$10. The episode length curves (Figure 17) mirror the return curves exactly: the two learning seeds grow from $\sim$10 steps to peaks of 120–170 steps by 1.2M steps, while the orange seed stays near 10–20 steps throughout. This parallel growth of return and episode length confirms that TD-MPC2's reward accumulation is driven by the agent learning to stay alive longer, not just achieving higher per-step rewards. Despite the high variance, the two learning seeds demonstrate that TD-MPC2 is considerably more sample-efficient than PPO, achieving SAC-level returns in roughly the same 500k–1M step window, but without the sharp collapse pattern observed in SAC. The upward trend in both learning seeds shows no sign of plateauing by 1.2M steps, suggesting continued improvement with a larger training budget.

5.2.4 Dreamer3

DreamerV3 was evaluated on the `g1-window-v0` task across a single seed up to 4.5M environment steps. As shown in Figure 6, the agent remains near the random baseline ($\sim$5) for the first 3M steps, with only marginal and highly noisy improvement. A modest upward trend begins around

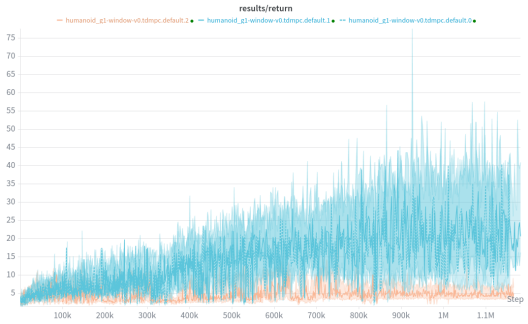

Figure 5: Reward Returns for `g1-window-v0` task with TD-MPC2 on 3 seeds

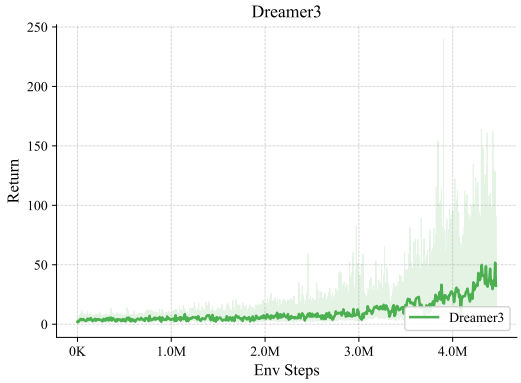

Figure 6: Episode returns for `g1-window-v0` task with Dreamer3

3M steps, eventually reaching a smoothed return of approximately 40 to 50 by 4.5M steps, though instantaneous returns spike as high as 150 due to extreme variance. Learning is both slow and unstable throughout: the confidence band is wide across the entire run, indicating high episode-to-episode variability rather than consistent policy improvement. These results are notably poor for a model-based method. DreamerV3's world model must simultaneously learn the high-dimensional proprioceptive dynamics of the G1 humanoid (77-dimensional observations, 19-dimensional actions) and a latent imagination policy, a significantly harder credit-assignment problem than model-free methods face. The window-cleaning task compounds this difficulty: rewards are sparse and contact-dependent, making it hard to learn an accurate reward predictor in latent space, which in turn corrupts the imagined rollouts used for policy optimization. Compared to SAC, which reaches returns of 60–75 within 2M steps by directly optimizing Q-values on real transitions, DreamerV3's model-learning overhead is a liability rather than an advantage on this task. Performance is roughly comparable to PPO, which also learns slowly but without the additional failure mode of a miscalibrated world model.

5.2.5 FlowRL

FlowRL was evaluated on the `g1-window-v0` task across a single seed up to 4.5M environment steps. As shown in Figure 7, the agent learns quickly relative to all other baselines, rising from the random baseline to a smoothed return of approximately 50 by 1M steps. Learning continues gradually, plateauing around 70 by 2M steps and remaining broadly stable thereafter, though with substantial instantaneous variance (spikes reaching 150–200). This plateau is the highest sustained return among all evaluated methods on this task. The strong sample efficiency of FlowRL can be attributed to its flow-based policy representation, which captures multimodal action distributions more expressively than Gaussian (SAC) or deterministic (TD3) policies. The constrained policy search objective, jointly maximizing Q-values while regularizing toward behavior-optimal actions in the

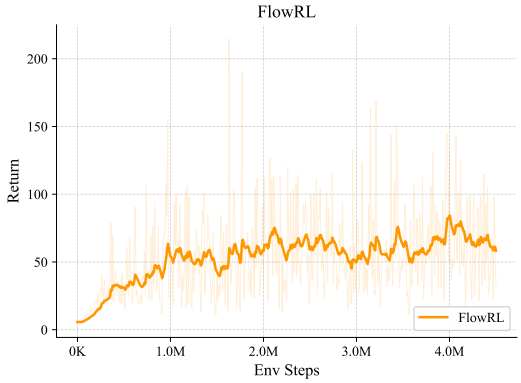

Figure 7: Episode returns for `g1-window-v0` task with FlowRL

buffer via a weighted CFM loss, provides a stable exploitation signal that prevents the policy collapse observed in SAC. The persistent variance despite the plateau suggests the policy continues to explore diverse action modes, a natural consequence of the stochastic ODE-based action generation, but without the catastrophic forgetting pattern seen in other methods.

5.2.6 SNQF

SNQF (Modification 1) was evaluated on the `g1-window-v0` task up to 2M environment steps. As shown in Figure 8, the agent learns steadily from the outset, rising from near zero to a smoothed return of approximately 30–35 by 1M steps and plateauing around 35 through 2M steps. Variance is high but comparable to FlowRL. The plateau is roughly half that of FlowRL on this task, indicating that removing the $Q^{\pi_{\beta^*}}$ machinery carries a performance cost on the challenging window-cleaning task. The gap relative to FlowRL is consistent with the theoretical difference between the two weighting schemes. FlowRL's weight $f(Q^{\pi_{\beta^*}} - Q^{\pi_\theta})$ provides a *relative* signal—it identifies buffer actions that are specifically better than what the current policy would do, whereas SNQF's exponential weight $\exp(\bar{Q}^{\pi_\theta}(s,a)/\alpha)$ identifies absolutely high-Q buffer actions without reference to the current policy's performance level. On a hard task like window cleaning where the current policy is far from optimal for most of training, the relative signal of FlowRL may provide a sharper exploitation target. The missing behavior policy networks thus appear to carry meaningful information on this task that the simpler reweighting cannot fully recover. Importantly, this gap does not hold uniformly. On easier locomotion tasks such as `h1-balance-simple-v0`, SNQF matches or outperforms FlowRL despite using one fewer critic system. This suggests that on tasks where the current Q-function is already a reliable signal of action quality, i.e., where the critic converges quickly, the exponential reweighting is sufficient and the auxiliary behavior-optimal critic adds noise rather than signal. The window-cleaning task, with its contact-dependent and spatially precise reward structure, likely keeps the critic noisier for longer, exposing the information advantage of the full FlowRL critic setup. Taken together, SNQF represents a favorable trade-off: substantially reduced algorithmic complexity at the cost of performance only on the most demanding tasks.

5.3 Success Rates

Neither PPO nor SAC achieves any measurable task success throughout training. Both `results/success` and `results/success_subtasks` remain flatlined at zero for the entire training duration across all seeds for both methods (Figure 16). However, the two methods fail at fundamentally different stages of the task pipeline.

5.3.1 PPO

For PPO, we do not see any plots for `hand_tool_proximity_reward`, `moving_wipe_reward`, and `stand_reward`. These metrics are never triggered because the robot does not meaningfully interact

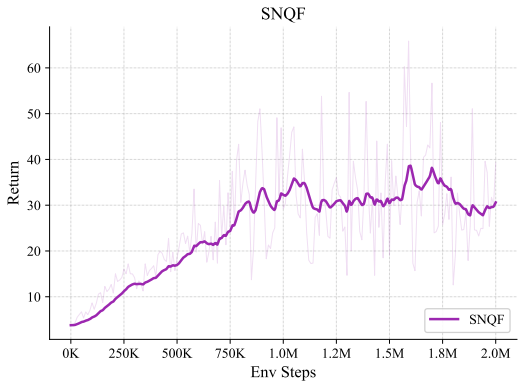

Figure 8: Episode returns for `g1-window-v0` task with SNQF

with the task at any point. It never gets near the tool, never makes wiper contact, and never achieves stable upright posture in a manipulation context. Over 20M steps, PPO learns to survive longer episodes, but it never once reaches the tool, let alone attempts to wipe the window. The task, from PPO's perspective, remains essentially unexplored.

5.3.2 SAC

The `hand_tool_proximity_reward` (Figure 9) rises from ∼0.4 to 0.7-0.85 by 2M steps, indicating that SAC does learn to bring its hands close to the tool. However, progress stalls there. The `moving_wipe_reward` (Figure 13) shows that it fluctuates noisily between 0.05 and 0.30 throughout training with no clear upward trend, meaning SAC makes intermittent wiper contact but never learns to execute the sustained, controlled wiping motion the task requires.

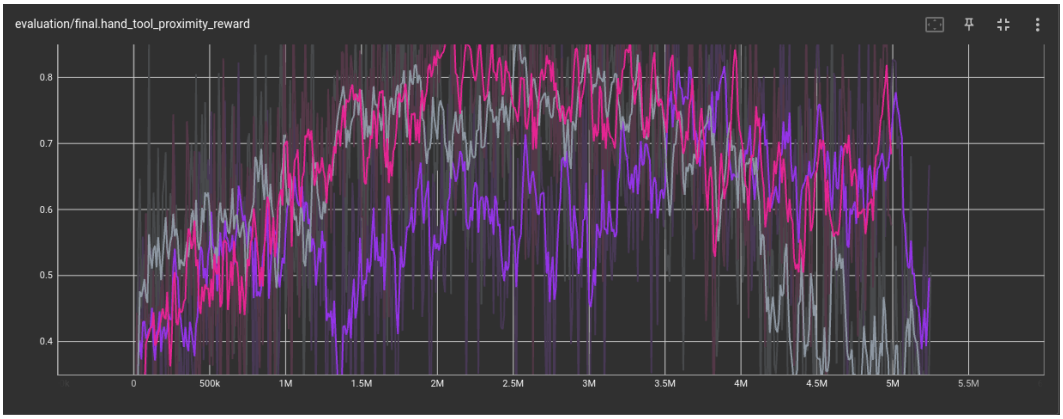

Figure 9: Hand-tool proximity reward for `g1-window-v0` task with SAC on 3 seeds

The `stand_reward` metric (Figure 12) remains low and highly variable throughout training, ranging between 0.04 and 0.19 with no improvement trend. For a task that fundamentally requires the robot to maintain stable upright posture while executing a coordinated arm motion, the inability to reliably satisfy the standing constraint is a critical failure. This directly explains the episode termination pattern discussed in Section 5.2.2: SAC is not collapsing because it forgets how to wipe, it is collapsing because it never reliably learns to stand while wiping in the first place.

5.3.3 TD-MPC2

TD-MPC2 achieves no measurable task success across any of the three seeds, with both `results/success` and `results/success_subtasks` flatlined at zero throughout training. De-

spite two seeds reaching competitive episode returns, the agent never simultaneously satisfies all conditions for a counted success. Sustained wiper contact with all five contact sites flush against the glass while maintaining stable upright posture. The high variance in returns is consistent with the agent occasionally achieving partial contact but failing to maintain it reliably enough to cross the success threshold.

TD-MPC2 achieves no measurable task success across any of the three seeds, with both `results/success` and `results/success_subtasks` flatlined at zero throughout training. Despite two seeds reaching competitive episode returns, the agent never simultaneously satisfies all conditions for a counted success. Sustained wiper contact with all five contact sites flush against the glass while maintaining stable upright posture. The high variance in returns is consistent with the agent occasionally achieving partial contact but failing to maintain it reliably enough to cross the success threshold.

5.3.4 FlowRL and SNQF

Neither methods achieve a success rate of 1.0.

Episode duration and length both tell the same story: FlowRL (pink) reaches smoothed durations of $\sim$2.6s and episode lengths of $\sim$224 steps by 4.5M steps, while SNQF (purple) reaches only $\sim$1.9s and $\sim$101 steps at 2M steps—FlowRL sustains episodes roughly twice as long, consistent with its higher return plateau. This is expected: FlowRL's behavior-optimal critic $Q^{\pi_{\beta^*}}$ provides a stable, relative constraint that keeps the policy anchored to high-quality buffer trajectories, encouraging the agent to stay alive and on-task longer. SNQF's softer exponential reweighting offers a weaker exploitation anchor, so the policy is more prone to drifting into failure states and terminating early.

5.4 Failure Cases

PPO's failure is purely one of exploration. The agent never progresses beyond basic postural stability. None of the task-relevant metrics are ever triggered, and the reward signal is dominated entirely by the stability component. In a 61-DoF action space, PPO's conservative on-policy updates simply never discover the behavioral sequence needed to reach the tool, let alone execute the wiping motion.

SAC's failure is more structural. Despite learning approximate hand-tool proximity, it cannot simultaneously satisfy postural stability and wiper contact precision. The `stand_reward` (Figure 12) never improves meaningfully, and every attempt to interact with the tool destabilizes the robot's posture, causing early termination. This creates a damaging cycle: manipulation attempts destabilize posture, posture failures cause early termination, and early termination prevents the agent from accumulating the experience needed to learn stable manipulation. This cycle, combined by the Q-value overestimation discussed in Section 5.2.2, explains why SAC's performance oscillates rather than converges.

TD-MPC2's failure mode is seed-dependent and rooted in two compounding issues. First, one seed completely fails to learn: its episode length never exceeds $\sim$20 steps and its policy scale (`train/pi_scale`, Figure 10) plateaus near 3 throughout training, compared to 12–17 for the learning seeds by 1.2M steps. This divergence in `pi_scale` indicates the failed seed's policy never learns to commit to the large, coordinated actions the task requires. It stays in a low-confidence, near-random action regime from which the world model never provides a learning signal strong enough to escape. Second, for the two learning seeds, the persistently high variance across the entire training run indicates the agent cannot consolidate its learned behaviors into a stable policy. The most likely cause is compounding model error: the wiper-glass contact introduces sharp discontinuities in the dynamics that are difficult for a smooth neural world model to represent accurately, and policy updates derived from inaccurate imagined rollouts through these contact phases produce oscillating behavior rather than convergence. Unlike SAC's collapse, which is abrupt and tied to Q-value overestimation, TD-MPC2's instability is chronic and diffuse: the agent never fully falls apart, but also never fully stabilizes.

For FlowRL, The most visible failure mode in Figure 7 is the persistent high variance despite a stable mean return. Even after the plateau at $\sim$70, instantaneous returns span a range of nearly 150 points. This suggests the flow policy is sampling a wide mixture of action modes—some highly rewarding, others near-failure—without collapsing onto a single consistent behavioral strategy. The multimodal expressiveness of the flow representation, which is a strength during exploration, becomes a liability at convergence: the policy never fully commits to the high-return mode, and the $Q^{\pi_{\beta^*}}$ constraint may not be tight enough to suppress the low-return modes once a good solution has been found.

For SNQF, The most concerning pattern in Figure 8 is the slight downward drift after $\sim$1.5M steps, where the smoothed return declines from $\sim$38 back toward 27–30. This is consistent with the known instability of AWR-style exponential reweighting under a non-stationary critic: as the Q-function continues to update, the weights $\exp(\bar{Q}/\alpha)$ shift, potentially upweighting different buffer actions than those that produced the plateau. Without the relative anchor that $Q^{\pi_{\beta^*}}$ provides, the exploitation target drifts with the critic, and the policy can gradually unlearn behavior that was previously rewarded. In effect, SNQF may be more susceptible to the moving-target problem in the CFM loss, causing slow regression rather than stable convergence.

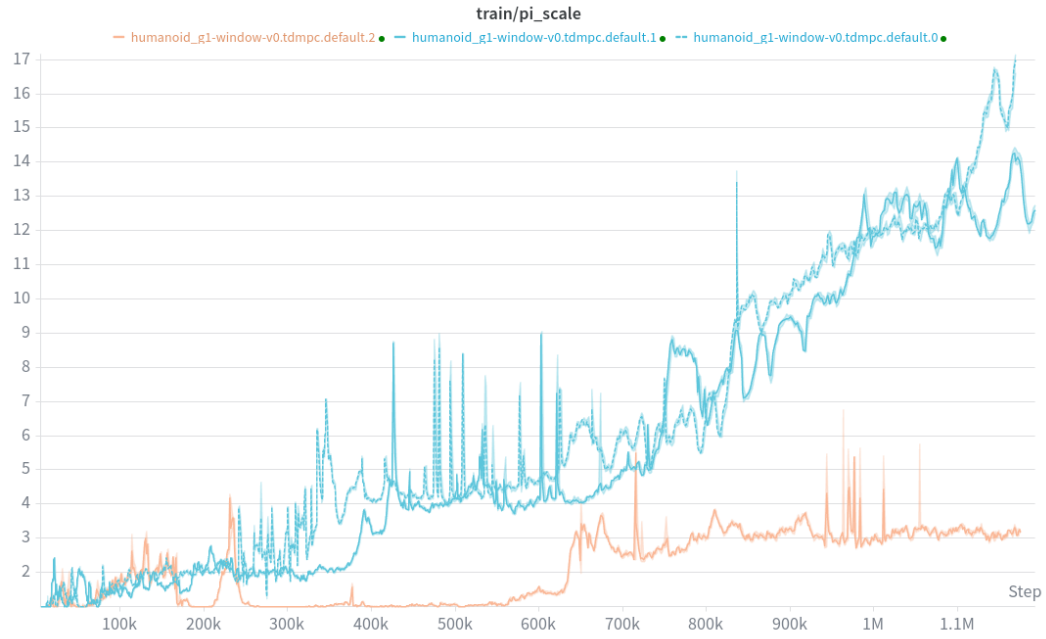

Figure 10: Pi scale for `g1-window-v0` task with TD-MPC2 on 3 seeds

6 Conclusion

We evaluated PPO and SAC on the g1-window-v0 whole-body manipulation task and found that both methods fail to achieve task success, albeit for different reasons. PPO is stable and reproducible but never progresses beyond basic postural stability, failing to discover any task-relevant behavior within a 20M step budget. SAC is far more sample-efficient and reaches returns 3-4$\times$ higher than PPO within 2M steps, but suffers from systematic policy instability driven by Q-value overestimation and an inability to simultaneously satisfy postural and contact precision constraints.

TD-MPC2 partially outperforms both PPO and SAC: two of three seeds reach competitive returns more sample-efficiently than PPO, and without SAC's catastrophic collapse, but one seed fails entirely and no seed achieves task success or converges to a stable policy.

References

[1] L. Lv, Y. Li, Y. Luo, F. Sun, T. Kong, J. Xu, and X. Ma. Flow-based policy for online reinforcement learning. *arXiv preprint arXiv:2506.12811*, 2025.

[2] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation, 2024. URL https://arxiv.org/abs/2403.10506.

[3] H. Lee, D. Hwang, D. Kim, H. Kim, J. J. Tai, K. Subramanian, P. R. Wurman, J. Choo, P. Stone, and T. Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. *arXiv preprint arXiv:2410.09754*, 2024.

[4] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

[5] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 47376–47405, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/cf73d57b6dcda32b293df7c2d5341f49-Paper-Conference.pdf.

[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290.

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[8] X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019. URL http://arxiv.org/abs/1910.00177.

[9] A. Nair, M. Dalal, A. Gupta, and S. Levine. Accelerating online reinforcement learning with offline datasets. *CoRR*, abs/2006.09359, 2020. URL https://arxiv.org/abs/2006.09359.

[10] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning. *CoRR*, abs/2110.06169, 2021. URL https://arxiv.org/abs/2110.06169.

[11] Z. Wang, A. Novikov, K. Zolna, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Y. Siegel, J. Merel, Ç. Gülçehre, N. Heess, and N. de Freitas. Critic regularized regression. *CoRR*, abs/2006.15134, 2020. URL https://arxiv.org/abs/2006.15134.

7 Appendix

7.1 Analysis: SimBa-Style Residual Networks

Motivation. FlowRL's original networks use a sequential `LayerNorm–Dense` stack (three dense layers per block) with no skip connections. Motivated by SimBa [3], we hypothesize that a *simplicity bias*— initializing each block near the identity—enables stable training with more parameters by ensuring the network learns only the corrections it needs.

Residual block. We replace each value block with a pre-norm residual feedforward unit:

$$\mathbf{x} \leftarrow \mathbf{x} + W_2\,\mathrm{SiLU}(W_1\,\mathrm{LN}(\mathbf{x})), \tag{7}$$

where W_2 is initialized with scale 0.1 so the residual branch starts near zero. The block is thus near-identity at initialization, and the network accumulates nonlinear corrections only where needed.

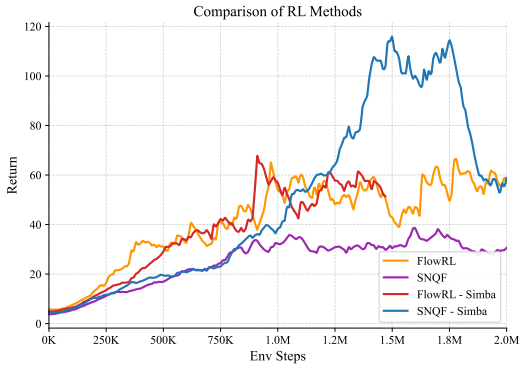

Figure 11: Deep network analysis

Network architectures. All three networks (twin Q-critic, value network, flow policy) follow the same blueprint:

$$\text{input} \;\to\; \underbrace{\text{Dense}}_{\text{projection}} \;\to\; \underbrace{[\text{ResBlock}]_{i=1}^{n}}_{\text{residual backbone}} \;\to\; \underbrace{\text{LN} \;\to\; \text{Dense}}_{\text{post-norm head}},$$

with $n = 3$ blocks for the critics and $n = 2$ for the actor and value network. A SiLU activation replaces the ELU/GELU activations used in the original FlowRL. The skip connections allow gradients to flow cleanly through deep stacks, reducing vanishing-gradient issues that can destabilize online RL training with larger networks.

Expected benefit. Near-identity initialization prevents early training from being dominated by random nonlinear transformations, while the residual pathway lets additional depth contribute without degrading the optimization landscape. This is especially beneficial in online RL, where value estimates are non-stationary and the loss surface evolves throughout training.

7.2 Additional Analysis on PPO

The episode length curve (Figure 15) tells the same story. Episodes grow from $\sim$20 steps to $\sim$65 steps at roughly the same rate as returns, and the three seeds remain nearly indistinguishable throughout. The parallel growth of return and episode length strongly suggests that PPO's reward accumulation is driven primarily by the agent learning to maintain balance and avoid early termination, rather than by discovering any meaningful window-cleaning behavior.

7.3 Additional Analysis on SAC

The episode length curves for SAC (Figure 18) mirror the return curves: every spike and collapse in return corresponds directly to a spike and collapse in episode length. This correspondence shows that SAC's performance drops are not caused by the agent adopting a worse within-episode strategy, but by episodes terminating earlier, most likely due to the robot losing balance or dropping the tool. The `per_timestep_reward` metric (Figure 14) supports this interpretation: even during return collapses, per-step reward remains relatively stable between 0.10 and 0.25, indicating that the agent's behavior quality when alive has not degraded, but it is simply staying alive for far fewer steps. This pattern of sharp rise followed by oscillatory collapse is consistent with Q-value overestimation in off-policy methods, where the critic learns to overestimate returns for certain state-action pairs, the policy exploits these overestimates to adopt physically aggressive behaviors that initially score well but are dynamically unstable, and performance degrades as a result. The consistency of this collapse across all three seeds at roughly the same training stage ($\sim$2M steps) points to a systematic issue rather than seed-specific bad luck.

Table 1: Reward Function Specification for `g1-window-v0`

Component	Weight	Equation / Logic	Parameters (G1)
Total Reward	1.0	$r_{\text{total}} = 0.5 \cdot r_{\text{manip}} + 0.5 \cdot r_{\text{contact}}$	
1. Manipulation Term (r_{manip})			
Sub-weight sum		$r_{\text{manip}} = 0.2 \cdot r_{\text{stable}} + 0.4 \cdot r_{\text{move}} + 0.4 \cdot r_{\text{prox}}$	
a. Stability Composite (r_{stable})	0.10	$r_{\text{stable}} = r_{\text{stand}} \cdot r_{\text{ctrl}} \cdot r_{\text{head}}$	
i. Standing Height	-	$r_{\text{stand\_h}} = \text{tol}(h_{\text{head}})$	bounds $= (1.28, \infty)$, margin $= 0.32$
ii. Upright Torso	-	$r_{\text{upright}} = \text{tol}(u_{\text{torso}})$	bounds $= (0.9, \infty)$, margin $= 1.9$
iii. Actuator Control	-	$r_{\text{ctrl}} = 0.8 + 0.2 \cdot \text{mean}(\text{tol}(F_{\text{act}}))$	bounds $= (0, 0)$, margin $= 10$, sigmoid=quad
iv. Head Dist. Constraint	-	$r_{\text{head}} = \text{tol}(\|p_{\text{head}} - p_{\text{init}}\|)$	bounds $= (0.4, 0.4)$, margin $= 0.1$
b. Tool Velocity (r_{move})	0.20	$r_{\text{move}} = \text{tol}(\lvert v_z^{\text{tool}} \rvert)$	bounds $= (0.5, 0.5)$, margin $= 0.5$
c. Hand-Tool Proximity (r_{prox})	0.20	$r_{\text{prox}} = \min(\text{tol}(d_{\text{left}}), \text{tol}(d_{\text{right}}))$	bounds $= (0, 0.2)$, margin $= 0.5$
2. Window Contact Term (r_{contact})			
Sub-weight sum		$r_{\text{contact}} = \mathbb{I}_{\text{collision}} \cdot r_{\text{surface}}$	
a. Contact Filter ($\mathbb{I}_{\text{collision}}$)	Mask	$\mathbb{I} = 1$ if $\text{geom}_{\text{pane}} \cap \text{geom}_{\text{wipe}} \neq \emptyset$	Binary mask (0 or 1)
b. Surface Alignment (r_{surface})	0.50	$r_{\text{surface}} = \min_{i \in \{a..e\}} \text{tol}(x_{\text{site}_i})$ *(alignment of 5 wiper sites to glass x-plane)*	bounds $= (0.92, 0.92)$, margin $= 0.4$

Note: $\text{tol}(v)$ refers to the `dm_control` tolerance function. Weights in the second column represent the effective contribution to the total reward. For G1, the stand height is set to 1.28m.

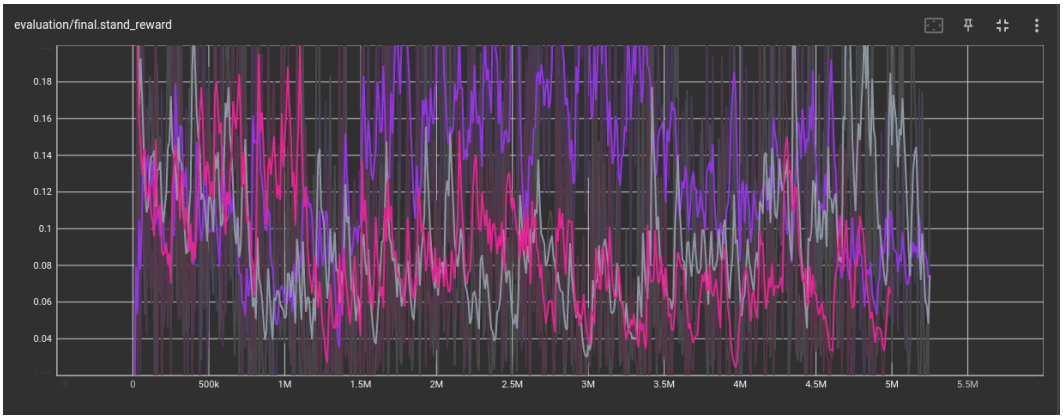

Figure 12: Stand reward for `g1-window-v0` task with SAC on 3 seeds

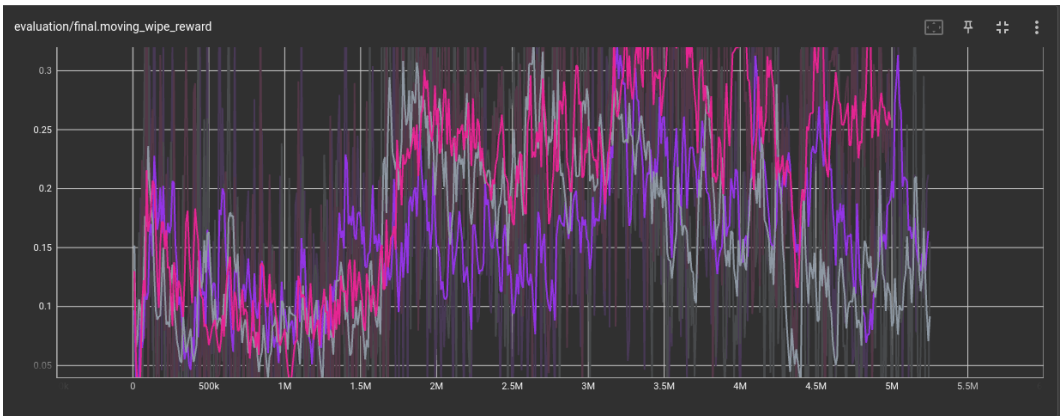

Figure 13: Moving wipe reward for `g1-window-v0` task with SAC on 3 seeds

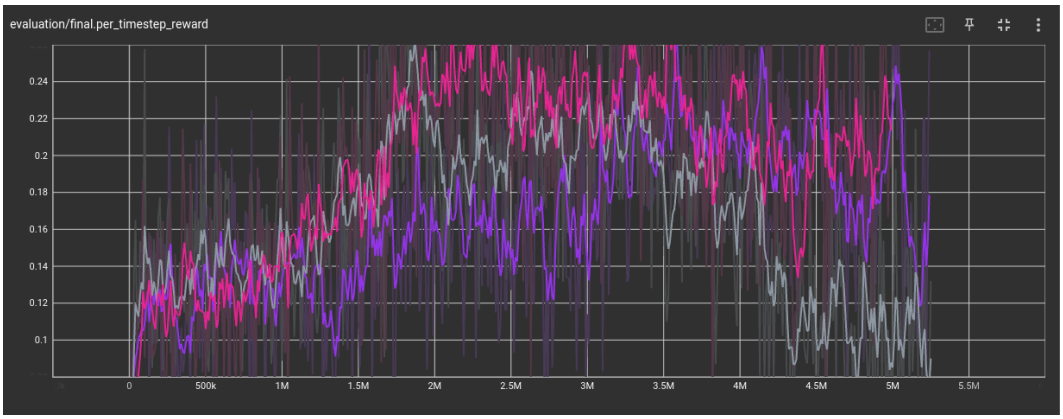

Figure 14: Per-timestep reward for `g1-window-v0` task with SAC on 3 seeds

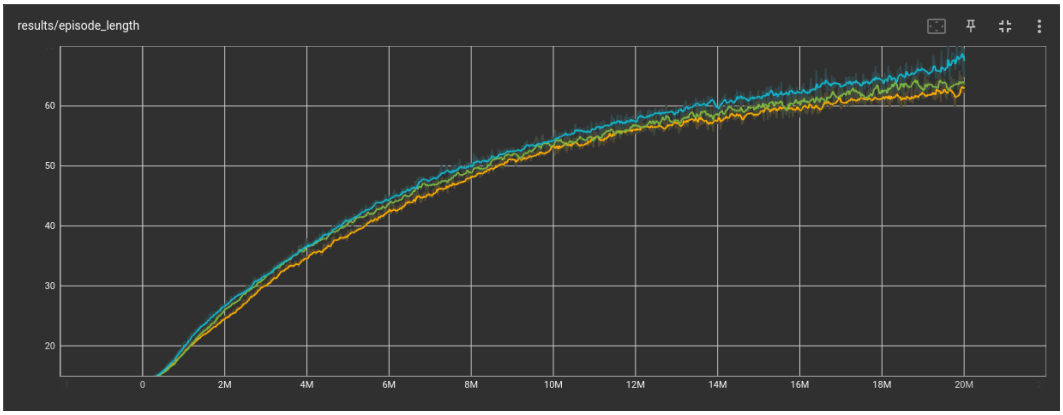

Figure 15: Episode lengths for `g1-window-v0` task with PPO on 3 seeds

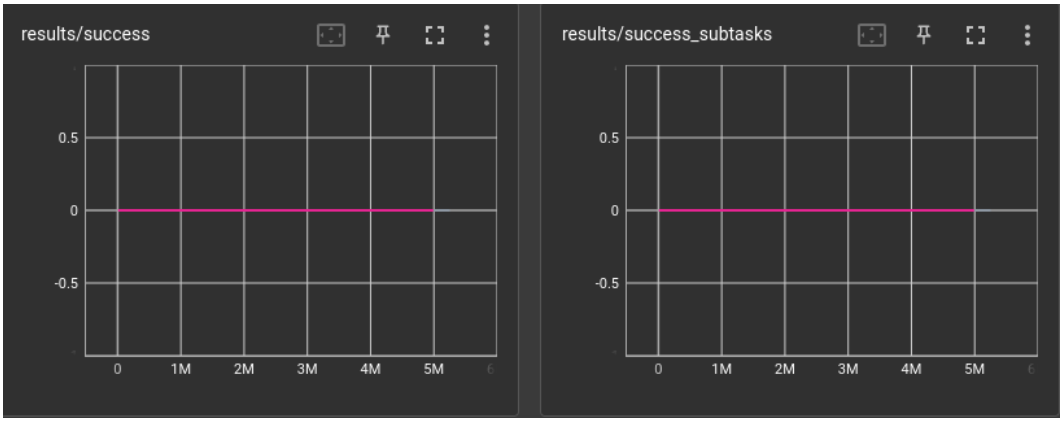

Figure 16: Task success and success subtasks for `g1-window-v0` task with PPO and SAC on 3 seeds each

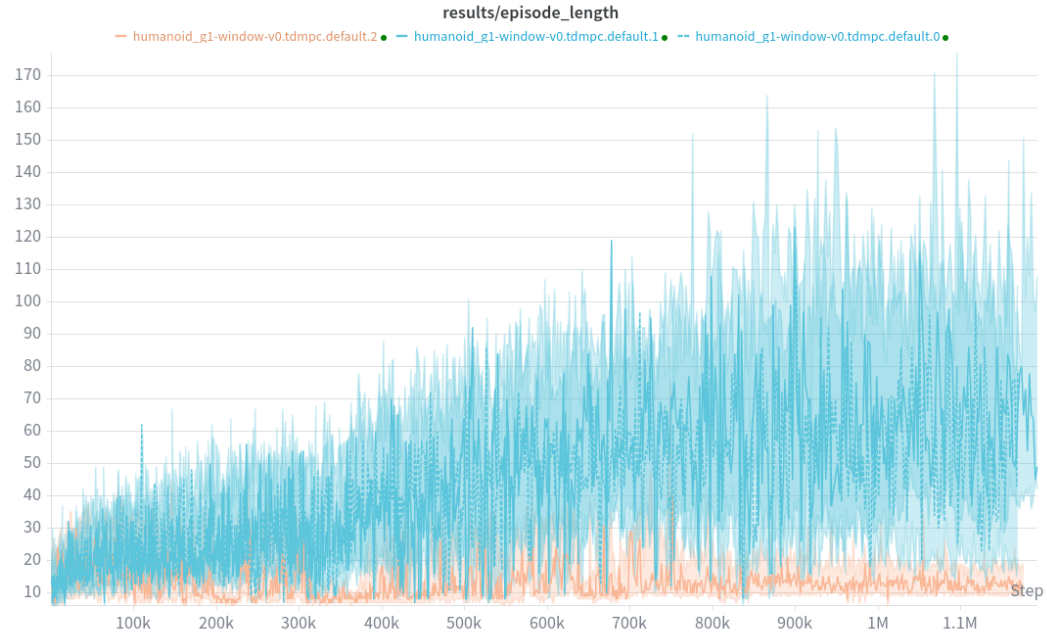

Figure 17: Episode lengths for `g1-window-v0` task with TD-MPC2 on 3 seeds

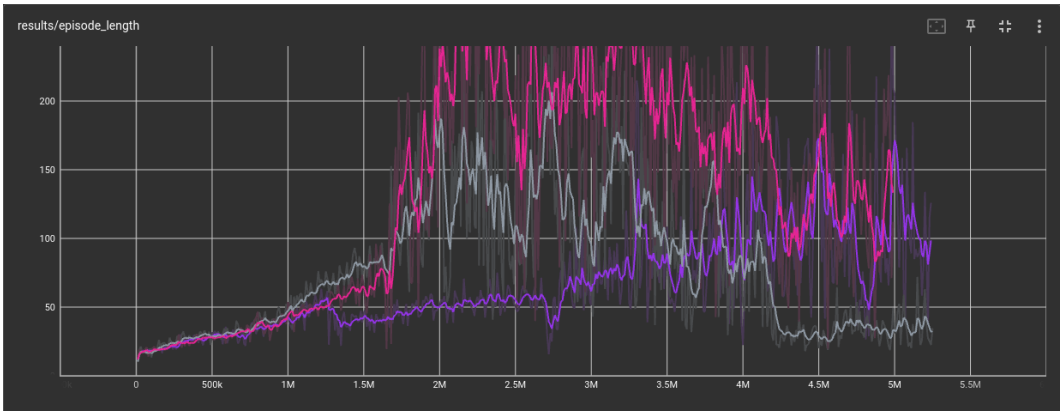

Figure 18: Episode lengths for `g1-window-v0` task with SAC on 3 seeds