Extended Abstract

Motivation The pursuit of creating versatile, autonomous robots has long been a central goal in artificial intelligence and robotics. While significant progress has been made in quadruped locomotion, achieving robust and adaptive locomotion in humanoid robots remains a formidable challenge. Humanoids, by their nature, can exhibit a much richer and more complex set of behaviors, from bipedal walking and running to crawling and climbing. Each of these behaviors demands a sophisticated level of coordination between the arms, legs, and torso that is not typically required for four-legged robots. This project undertakes a foundational step toward enabling such complex locomotion on the **ToddlerBot**, a small, 3D-printed, open-source humanoid robot designed for machine learning research Shi et al. (2025). Our central aim is to develop a stable and effective training pipeline capable of teaching the ToddlerBot to imitate complex reference motions, laying the groundwork for a unified, egocentric policy that can navigate diverse and unstructured environments.

Method Our methodology is rooted in the principles of example-guided deep reinforcement learning, inspired by the **DeepMimic** framework Peng et al. (2018), and implemented within the **Brax** physics engine (MJX). The training regimen is structured as a two-step process. First, the robot undergoes **pre-training in a fixed environment** (`fixed_env`), where it learns to follow a reference motion without external disturbances. This phase is critical for establishing a foundational motion prior. Second, the agent is moved to a **regular, interactable environment** for fine-tuning, where it must learn to maintain the reference poses while adapting to the challenges of balance and survival. This phase introduces domain randomization, random initializations, and terrain perturbations.

Implementation To guide the learning process, we implemented a suite of reward functions loyal to the original DeepMimic paper. These include: a `Reward_pose` that tracks the orientation of body parts using quaternion differences and an exponential reward function ($e^{-(k \cdot \text{error}^2)}$); a `Reward_end_effector_pos` for aligning the hands and feet to the reference trajectory; a `Reward_com_pos` to ensure torso alignment; and velocity-based rewards (`Reward_joint_lin_vel`, `Reward_ang_vel`) computed via finite differencing of positions and quaternions. These are implemented using MJX-compatible state variables and verified on ground-truth replay trajectories. We also explored action scale normalization, policy output rescaling, and learning rate sweeps during finetuning.

Results Our experiments revealed that successful pretraining was possible only when using a high **action scale of 2.0**, which led to reasonable motor tracking rewards (final error near -20). However, this high action range resulted in jittery arm motions and poor foot-ground alignment. Reducing the action scale below 1.5 caused the position tracking reward to plateau early, failing to learn the reference motion at all (final reward near -100). Finetuning in the interactable environment proved more difficult. At a low learning rate ($1 \cdot 10^{-5}$), the policy retained some motion prior but failed to survive. At a higher learning rate ($5 \cdot 10^{-5}$), survival rewards improved briefly but caused catastrophic forgetting: the policy abandoned its pre-trained reference behavior, and motor position rewards collapsed entirely. Additional changes—such as replacing motor tracking with pose-based rewards—showed promise in offline GT evaluation but still failed during online PPO training. We consistently observed instability across multiple runs.

Discussion Through extensive ablations, we concluded that reward structure alone is insufficient to overcome policy instability. One key insight is that survival and imitation objectives often compete in the optimization process, especially without proper observation scaling. Our current PPO setup lacks asymmetric observation handling, leading to poor generalization in dynamic environments. Despite implementing reward components that track pose, velocity, and end-effectors, the learning still fails without careful normalization and architecture-level adjustments. The failure modes point toward deep conflicts between precision tracking and robustness, and resolving this requires improved input representations and stable critic targets.

Conclusion To address these limitations, we plan to migrate to the latest version of Brax, which natively supports asymmetric actor-critic architectures with normalized observations. This should improve the stability of both imitation and survival training. Once the pipeline is stabilized, we aim to train a library of expert behaviors (e.g., crawling, walking, climbing) and integrate them into a unified policy via behavior stitching or Adversarial Motion Priors (AMP). This will enable whole-body control from egocentric vision in unstructured terrain, moving us closer to general-purpose humanoid locomotion on real hardware.

Toward Whole-Body Locomotion for Humanoid Robot

Tae Yang
Department of Computer Science
Stanford University
taeyang@stanford.edu

Group Member 2
Department of Electrical Engineering
Stanford University
name2@stanford.edu

Zhicong Zhang
Department of Mechanical Engneering
Stanford University
zhicong5@stanford.edu

Abstract

Achieving robust and adaptive locomotion in humanoid robots remains an open challenge due to the complexity of coordinating full-body motions across arms, legs, and torso. In this work, we take a step toward enabling whole-body behaviors such as crawling on **ToddlerBot**, a compact, 3D-printed humanoid platform designed for learning-based control. We develop a reinforcement learning pipeline inspired by the DeepMimic framework, implemented in the Brax (MJX) simulator. Our approach consists of two stages: (1) pretraining the agent in a fixed environment to follow handcrafted reference motions, and (2) fine-tuning in a domain-randomized environment with survival and balance constraints. We implement a suite of rewards including pose alignment via quaternion error, end-effector tracking, and joint velocity matching. While the agent successfully learns short-horizon crawling behaviors in the fixed environment, fine-tuning in dynamic settings reveals significant instability, including catastrophic forgetting and degraded tracking. We identify the lack of observation normalization and adversarial reward conflict as primary causes. To address these issues, we plan to adopt asymmetric actor-critic architectures and train multiple expert behaviors to be composed into a unified, vision-conditioned locomotion policy.

1 Introduction

Humanoid locomotion remains one of the most ambitious goals in robotics, demanding precise whole-body coordination and adaptability to diverse environments. Unlike quadrupeds, whose locomotion patterns largely rely on synchronized leg motion, humanoid robots can exhibit a wide spectrum of behaviors—such as walking, crawling, and climbing—each of which requires dynamic use of arms, legs, and torso for balance, propulsion, and contact reasoning. The ability to switch between such modes based on terrain context is essential for enabling versatile, real-world humanoid navigation.

Recent advancements in reinforcement learning (RL) have shown promise in enabling humanoid robots to learn complex locomotion behaviors. For instance, Radosavovic et al. introduced a transformer-based controller trained via RL that enables real-world humanoid locomotion across diverse terrains without fine-tuning Radosavovic et al. (2023). Similarly, Thibault et al. demonstrated velocity-based RL locomotion policies for the REEM-C robot using Brax/MJX for fast, parallel training Thibault et al. (2024).

This project investigates the problem of learning such whole-body locomotion behaviors on **ToddlerBot**, a compact, open-source, 3D-printed humanoid robot designed for learning-based control Shi et al. (2025). Our ultimate goal is to create a unified, egocentric locomotion policy that allows ToddlerBot to traverse complex environments using diverse motor skills. As a first step, we focus on training the robot to imitate short reference crawling motions using reinforcement learning in simulation, and analyze the challenges involved in transferring these skills to more realistic, perturbed environments.

We adopt a two-stage pipeline: (1) pretraining in a fixed environment to learn a reference motion from keyframed trajectories, and (2) fine-tuning in an interactable environment using Brax (MJX) with domain randomization and survival objectives. To guide the agent, we employ a DeepMimic-style reward formulation Peng et al. (2018) that tracks pose, velocity, end-effector alignment, and center-of-mass stability. Keyframe animations are manually engineered and tested in MuJoCo before being used as ground-truth reference trajectories (Figure 1).

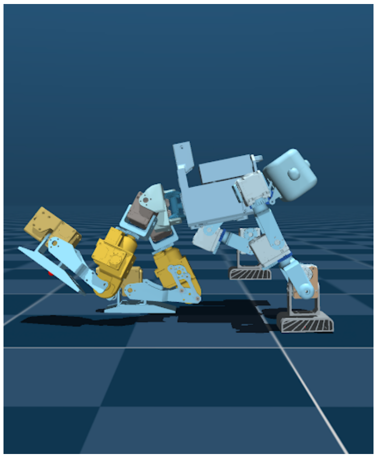
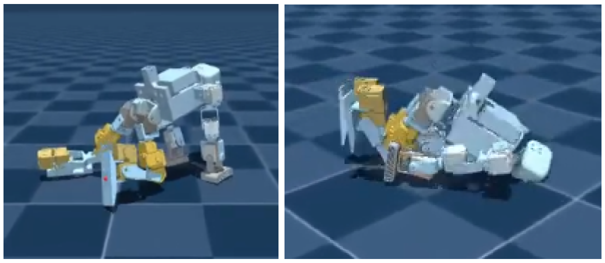


Figure 1: Interpolated keyframe-based crawling motion visualized in MuJoCo. These sequences form the reference trajectory for RL pretraining.

While the agent is able to reproduce short crawling motions in the fixed environment, we observe significant instability during the fine-tuning stage. For example, policies trained with a high action scale (2.0) achieve reasonable tracking performance (final error ≈ -20), but suffer from jittery and imprecise limb placement. On the other hand, reducing the action scale below 1.5 causes the learning process to stall entirely, with position tracking rewards plateauing above -100.

Additionally, as shown in Figure 2, fine-tuning in the regular environment reveals an adversarial interaction between survival and tracking objectives. PPO policies often forget the reference motion, failing to maintain torso stability or consistent reward signals over time. These instabilities suggest a systemic optimization issue rather than a reward misconfiguration.



(a) Catastrophic forgetting of reference motion.

(b) Collapse of motor position rewards in regular environment.

Figure 2: Training instability observed during fine-tuning.

We hypothesize that the root cause lies in a lack of proper observation normalization and architectural rigidity in the current PPO implementation. To address this, we propose migrating to the latest version of Brax, which supports asymmetric actor-critic architectures and observation normalization. This architectural shift is expected to improve stability by separating privileged reference inputs from learned policy observations.

Our longer-term plan is to extend this framework to multiple expert behaviors (e.g., crawling, walking, climbing) and combine them into a unified high-level controller via policy distillation or adversarial motion priors (AMP). This work lays a foundation for scalable whole-body motion learning in humanoids and emphasizes the importance of robust policy architecture and motion priors for general-purpose control.

2 Related Work

Deep Reinforcement Learning for Humanoid Locomotion. Deep reinforcement learning (DRL) has significantly advanced the capabilities of humanoid robots in performing complex locomotion tasks. Peng et al.'s **DeepMimic** framework Peng et al. (2018) introduced a modular reward design enabling physics-based characters to imitate motion capture data effectively. Building upon this, Radosavovic et al. Radosavovic et al. (2023) developed a transformer-based controller trained via RL, achieving real-world humanoid locomotion across diverse terrains without fine-tuning. Additionally, Figure AI demonstrated natural walking controllers learned purely in simulation using end-to-end RL, facilitating rapid engineering iterations for their humanoid robots AI (2025).

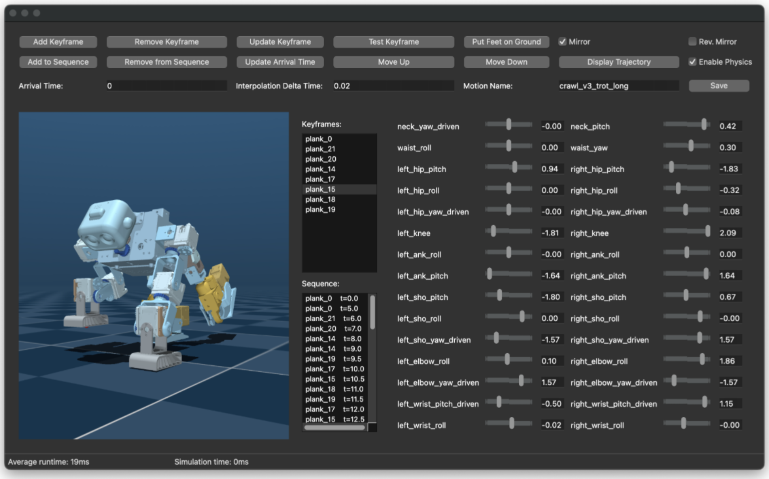
Whole-Body Humanoid Motion and Multi-Behavior Integration. Achieving whole-body coordination in humanoid robots necessitates integrating various motor skills. The **AMP** framework Peng et al. (2021) leverages adversarial motion priors to learn stylized physics-based character control, enabling the synthesis of diverse behaviors. Similarly, the **SMAP** framework Zhao et al. (2025) introduces self-supervised motion adaptation for physically plausible humanoid whole-body control, bridging the gap between human and humanoid action spaces. The **AMO** framework Li et al. (2025) combines sim-to-real RL with trajectory optimization for real-time adaptive whole-body control, demonstrating superior stability and an expanded workspace.

Simulation Frameworks for Scalable Training. Efficient simulation environments are crucial for training complex locomotion policies. **MJX**, a JAX-compatible, GPU-accelerated physics engine built on Brax, enables fast parallel rollout and backpropagation, facilitating large-scale training Google Research (2021). **Humanoid-Gym** Gupta et al. (2025) provides an easy-to-use RL framework based on Nvidia Isaac Gym, designed to train locomotion skills for humanoid robots with an emphasis on zero-shot transfer from simulation to real-world environments.

Vision-Based Control and Teleoperation. Integrating vision into control policies enhances the adaptability of humanoid robots. The **H2O** framework He et al. (2024) enables real-time whole-body teleoperation of a full-sized humanoid robot using only an RGB camera, employing a scalable "sim-to-data" process to filter and select feasible motions. This approach allows for dynamic whole-body motions in real-world scenarios, including walking, back jumping, and kicking.

Curriculum Learning and Gait Conditioning. Curriculum learning strategies have been employed to facilitate the acquisition of complex locomotion behaviors. The **DeepWalk** approach Rodriguez et al. (2021) utilizes a novel DRL method to enable omnidirectional locomotion for humanoid robots, introducing a curriculum that gradually increases task difficulty. Additionally, a unified gait-conditioned RL framework Chen et al. (2025) allows humanoid robots to perform standing, walking, running, and smooth transitions within a single recurrent policy, employing a compact reward routing mechanism to support stable multi-gait learning.

In summary, our work builds upon these advancements by adapting modular reward designs and scalable simulation frameworks to the ToddlerBot platform. By focusing on whole-body motion imitation and addressing the challenges of training stability and multi-behavior integration, we aim to contribute to the development of versatile and robust humanoid locomotion policies.



☑ Mirror ☐ Rev. Mirror ☑ Enable Physics

Arrival Time: 0 Interpolation Delta Time: 0.02 Motion Name: crawl_v3_trot_long

Keyframes:

```
plank_0
plank_21
plank_20
plank_14
plank_17
plank_15
plank_18
plank_19
```

Sequence:

```
plank_0   t=0.0
plank_0   t=5.0
plank_21  t=6.0
plank_20  t=7.0
plank_14  t=8.0
plank_14  t=9.0
plank_19  t=9.5
plank_17  t=10.0
plank_15  t=10.5
plank_18  t=11.0
plank_19  t=11.5
plank_17  t=12.0
plank_15  t=12.5
```

neck_yaw_driven	-0.00	neck_pitch	0.42
waist_roll	0.00	waist_yaw	0.30
left_hip_pitch	0.94	right_hip_pitch	-1.83
left_hip_roll	0.00	right_hip_roll	-0.32
left_hip_yaw_driven	-0.00	right_hip_yaw_driven	-0.08
left_knee	-1.81	right_knee	2.09
left_ank_roll	-0.00	right_ank_roll	0.00
left_ank_pitch	-1.64	right_ank_pitch	1.64
left_sho_pitch	-1.80	right_sho_pitch	0.67
left_sho_roll	0.00	right_sho_roll	-0.00
left_sho_yaw_driven	-1.57	right_sho_yaw_driven	1.57
left_elbow_roll	0.10	right_elbow_roll	1.86
left_elbow_yaw_driven	1.57	right_elbow_yaw_driven	-1.57
left_wrist_pitch_driven	-0.50	right_wrist_pitch_driven	1.15
left_wrist_roll	-0.02	right_wrist_roll	-0.00

Average runtime: 19ms Simulation time: 0ms

Figure 3: Key-frame editor built with PyQt.

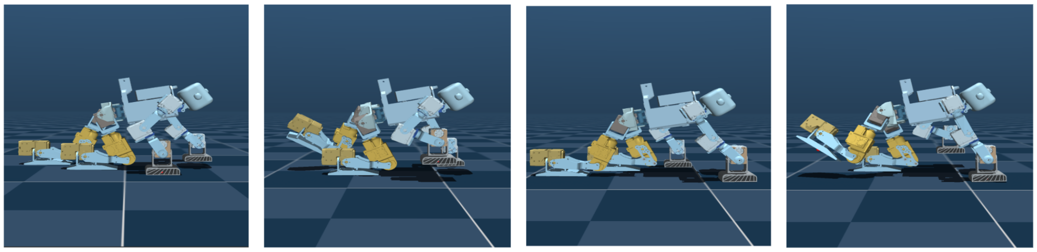


Figure 4: The four key-frames that created the crawling motion.

3 Method

3.1 Reference-Motion Acquisition

We developed a MuJoCo key-frame editor with PyQt to create the crawling motion (see Fig. 3). Poses are first hand-placed, then interpolated with cubic splines, and exported as NumPy arrays of joint angles, velocities and root quaternions for every 10 ms step. These files serve simultaneously as imitation targets and as privileged inputs for the critic.

3.2 Reinforcement-Learning Formulation

We pose whole-body crawling as a continuous-control task in Brax (MJX) Google Research (2021). The actor takes the observation $\mathbf{o}_t$ and produces bounded joint-position targets. These are rescaled and blended with the reference command:

$$\mathbf{a}_t = s_a \tanh\big(f_\theta(\mathbf{o}_t)\big) + \mathbf{a}_t^{\text{ref}}, \tag{1}$$

where $s_a = 2.0$ during pre-training and f_θ is a shared MLP policy. Attempts to lower s_a below 1.5 caused learning to stall, confirming the sensitivity to actuation range.

We adopt Brax's default PPO implementation. Advantage estimates are computed with Generalised Advantage Estimation (GAE):

$$\hat{A}_t = \sum_{l=0}^{T-1-t} (\gamma\lambda)^l \, \delta_{t+l}, \quad \delta_t = r_t + \gamma V_{\theta_{\text{old}}}(s_{t+1}) - V_{\theta_{\text{old}}}(s_t). \tag{2}$$

The actor is updated with the clipped surrogate objective

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(\rho_t(\theta)\,\hat{A}_t,\ \text{clip}\big(\rho_t(\theta), 1-\varepsilon, 1+\varepsilon\big)\,\hat{A}_t\big)\Big], \quad \rho_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}, \tag{3}$$

while the critic minimises

$$L^V(\phi) = \tfrac{1}{2}\,\mathbb{E}_t\Big[\big(V_\phi(s_t) - V_t^{\text{target}}\big)^2\Big]. \tag{4}$$

Observations are standardised online via an exponential moving average.

3.3 Reward Functions

We adopt three exponential tracking terms from DeepMimic and add three robot-specific terms:

1. **Pose**: tracks body-part orientations via quaternion error.
2. **Joint velocity**: tracks linear and angular joint velocities.
3. **End-effector position**: tracks hand and foot positions.
4. **Torso**: tracks torso position and orientation.
5. **Velocity tracking**: tracks overall body velocity.
6. **Motor regularisation**: penalises excessive torque and acceleration.

Each term has the exponential form

$$r_{k,t} = \exp\big(-\kappa_k\,\|e_{k,t}\|_2^2\big), \tag{5}$$

and the total step reward is

$$r_t = \sum_k w_k\, r_{k,t} + r_{\text{survive}}, \tag{6}$$

where w_k are term-specific weights and $r_{\text{survive}} = 1$ if the torso height exceeds 7 cm.

3.4 Two-Stage Training Pipeline

Stage 1 – Pre-training (`fixed_env`). The policy is trained for 100 PPO updates on a flat plane, tracking the reference motion only.

Stage 2 – Fine-tuning (`regular_env`). Domain randomisation is enabled—random pushes, variable friction, randomised initial poses and Perlin-noise terrain. Because the actor output is already `tanh` bounded, we leave network weights untouched and tune only the learning rate.

3.5 Stereo Depth Estimation

Using ToddlerBot's fisheye stereo cameras, we implemented depth estimation with FoundationStereo Wen et al. (2025) (Fig. 5). To meet the 10fps budget on an NVIDIA Jetson Orin Nano, we compiled a TensorRT FP16 engine at 96×128 resolution. Figure 6 compares simulated and real-world depth on YCB objects Calli et al. (2015).

4 Experimental Setup

We conduct all experiments using the Brax simulator with the MJX physics backend. The robot model used is the ToddlerBot, a 30-active-degree-of-freedom quadruped approximately 0.56 meters in height. Control signals are issued at 100 Hz, and each training episode is 500 steps long, which matches the upper bound of the exponential reward terms used in the imitation loss. This duration

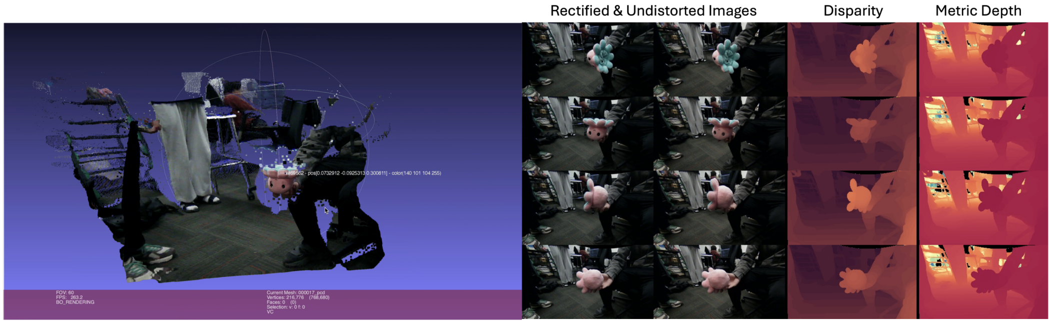


Figure 5: Depth estimation pipeline. Raw fisheye images are rectified and undistorted before inference.

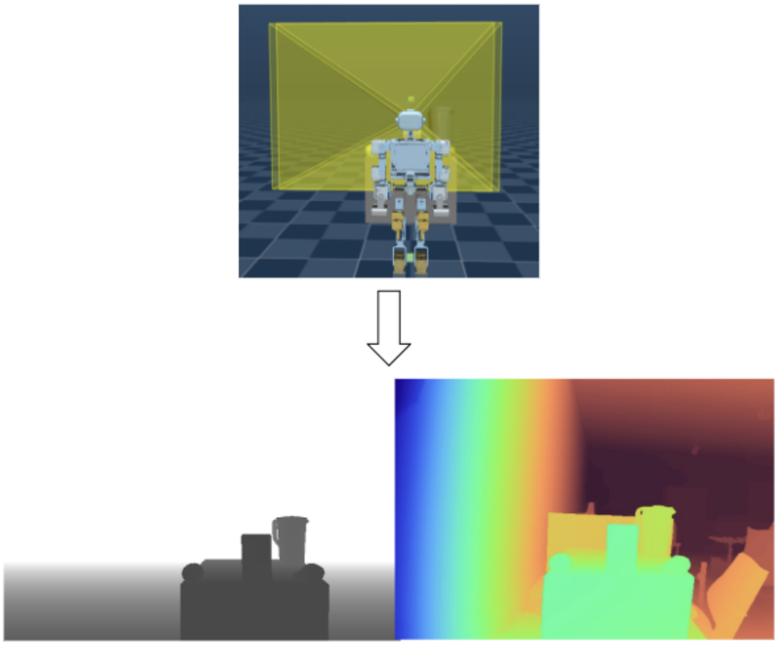


Figure 6: Left: simulated depth; right: estimated depth from FoundationStereo.

ensures that the agent experiences the full temporal context of the reference motion during training and evaluation.

Two environments are used throughout training. The first, `fixed_env`, contains a flat ground plane and no external disturbances. This environment is used exclusively for pre-training the policy to track the reference motion without needing to worry about balance or perturbations. The second environment, `regular_env`, introduces a variety of perturbations: Perlin-noise terrain, randomized initial root poses, randomized friction coefficients drawn from the interval $[0.8, 1.2]$, and random lateral pushes of up to ± 5 Newtons. This environment is used for fine-tuning, where the agent must not only imitate the motion but also survive under domain randomization and environmental variability.

Optimization is performed using PPO with Brax's default settings. We experiment with two learning rates during fine-tuning: 1×10^{-5}, which preserves the crawling behavior but fails to adapt to disturbances, and 5×10^{-5}, which improves survival but degrades imitation fidelity. No other PPO parameters were changed, and the same actor and critic architectures were maintained across experiments. The actor network outputs joint position targets, bounded by a `tanh` layer; the action scale is fixed at 2.0 throughout, as lower values were found to prevent convergence in the pre-training stage.

For evaluation, we track imitation quality using the negative mean squared error between the agent's joint angles and those in the reference trajectory. Additionally, we measure survival time and provide a reward breakdown across components to evaluate the agent's robustness and policy generalization. These metrics are logged and visualized across multiple random seeds to assess consistency and performance variance. All experiments are conducted with the goal of isolating the effects of fine-tuning under increasing environmental complexity while preserving the quality of motion learned during imitation pre-training.

5 Results

5.1 Pre-training in Fixed Environment

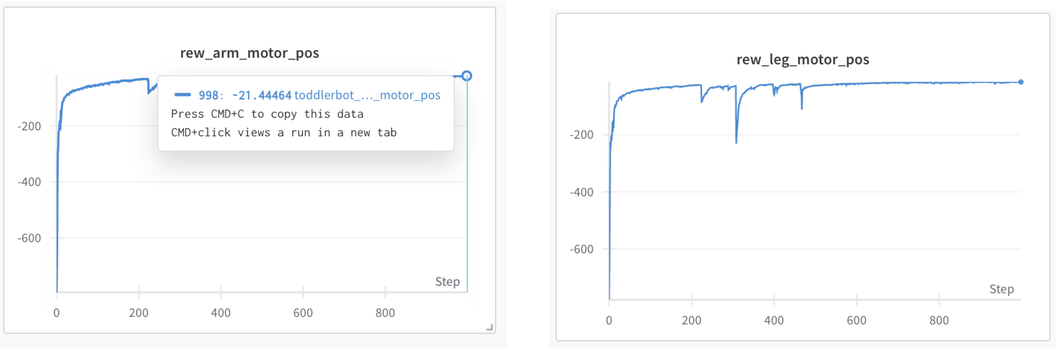
We apply a negative mean-squared-error between target and actual joint positions as the motor tracking reward, and set the action scale to 2.0 to ensure sufficient control amplitude.

Quantitative Results

- **Action Scale:** 2.0
- **Arm MSE Reward:** converges to approximately -21.4 by 1000 steps, indicating accurate arm joint tracking (see Fig. 7b).
- **Leg MSE Reward:** also reduces steadily, plateauing at a similar order of magnitude (see Fig. 7a), though with higher variance across seeds.

Qualitative Observations

- Despite adding torque and energy penalties to smooth controls, the resulting motion is still noticeably jittery.
- End-effectors (hands and feet) do not align precisely with the reference trajectory, suggesting that position-only supervision is insufficient for full pose fidelity.



(a) Leg motor position MSE reward

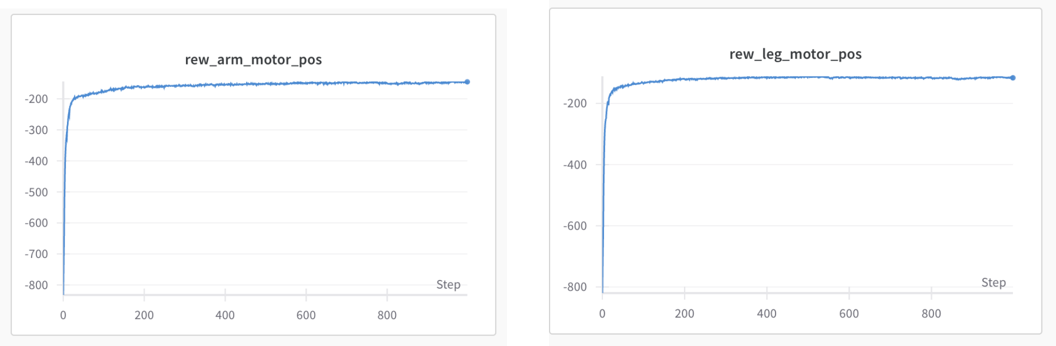
(b) Arm motor position MSE reward

Figure 7: Negative MSE joint-tracking rewards during pre-training (action scale $= 2.0$).

Failure under Low Action Scale To assess sensitivity to action magnitude, we repeated the negative-MSE pre-training with action scales below 1.5. In this regime, both arm and leg tracking rewards flatten out at very high losses (worse than –100) and never improve over 400 steps, indicating that the policy entirely fails to learn the reference crawl when control amplitudes are too small. (see Fig 8)

5.2 Fine-Tuning in the Interactable Environment

Starting from the 1000 steps pre-trained checkpoint (action scale = 2.0), we fine-tune the policy in the full Brax environment with an added survival objective. Because the actor's final layer remains tanh–activated, we did not alter the network output directly but instead reduced the learning rate to encourage stable adaptation.



(a) Leg motor position MSE reward

(b) Arm motor position MSE reward

Figure 8: Negative MSE joint-tracking rewards during pre-training (action scale < 1.5).

Learning Rate = 1×10^{-5}

- **Survival reward:** Remains at zero throughout fine-tuning, showing that the agent fails to acquire any survival behavior (Fig. 9).
- **Leg motor position reward:** After an initial transient increase, it steadily declines over 350 steps, indicating the policy abandons the reference gait when pressured by survival loss (Fig. 9).
- **Reference motion retention:** Qualitatively, the agent still recalls the pre-trained crawling pattern to some extent but cannot execute it stably in the interactive environment—likely due to the overly small learning rate.

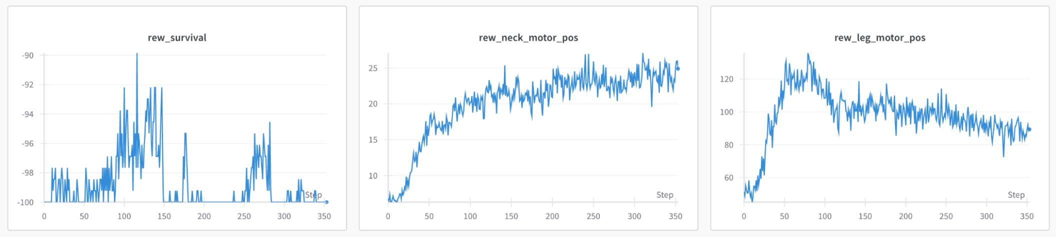


Figure 9: Fine-tuning dynamics in the interactable environment at lr = 1×10^{-5}.

Learning Rate = 5×10^{-5}

- **Survival reward:** Stays at zero throughout fine-tuning, confirming that the agent never acquires any survival behavior at this learning rate (Fig. 10).
- **Leg motor position reward:** Exhibits a brief uptick before continuously decaying to near zero by step 350, indicating that under survival-driven gradients the pretrained gait is progressively unlearned (Fig. 10).
- **Reference gait stability:** Although the agent initially echoes the pretrained crawling pattern, it quickly loses stability and cannot execute the motion reliably (Fig. 10).

5.3 Different tracking rewards

Additionally, we test different types of motoy tracking reward in fixed environment:

MSE-based Motor Tracking Reward Using a negative mean-squared-error tracking reward, the learned policy exhibits:

- **Leg motor position reward:** Fails to converge, plateauing near -90 (versus -15 when using action scale $= 2.0$), showing poor leg tracking under MSE loss (Fig. 11).

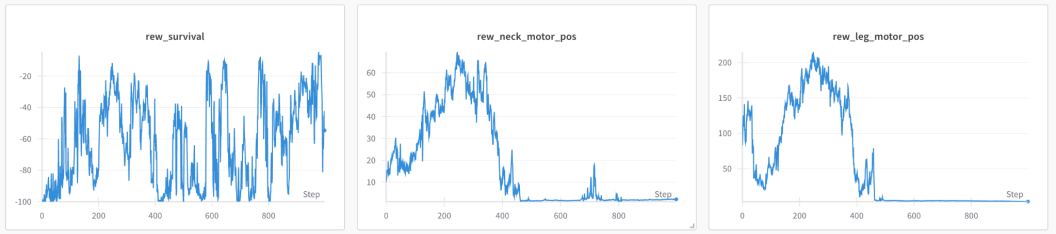


Figure 10: Fine-tuning dynamics in the interactable environment at lr = 5×10^{-5}.

- **Training stability:** Suffers from large oscillations, reflecting unstable learning dynamics in the static environment.

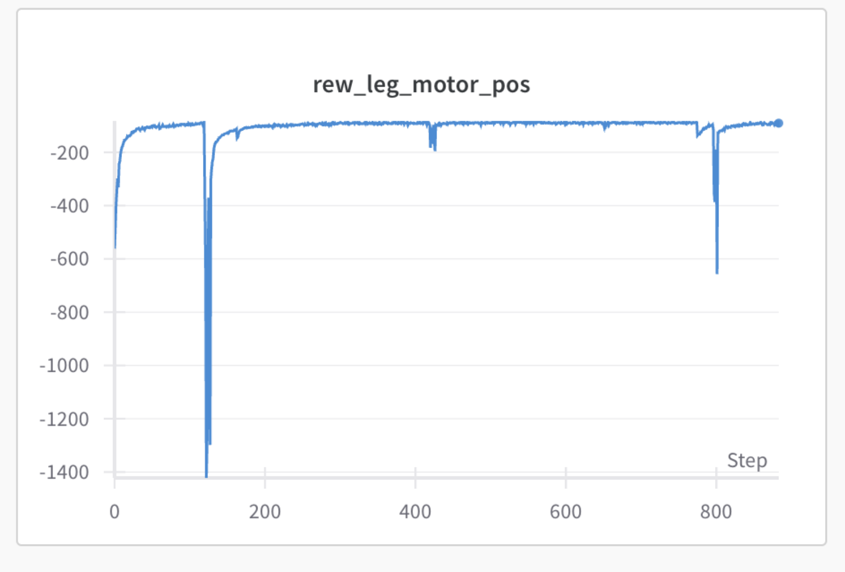


Figure 11: Arm and leg motor tracking rewards under MSE reward in the `fixed_env`.

Exponential-form Motor Tracking Reward Switching to an exponential-shaped tracking reward (episode length = 500, so max possible reward = 500), we observe:

- **Leg motor position reward:** Stagnates near 80 with intermittent catastrophic drops, indicating highly unstable leg tracking even in this non-interactive setting (Fig. 12).

6 Discussion

Our ablations reveal that simply reshaping the reward structure fails to stabilize our PPO policy: although multi-component motor-tracking signals (pose error, end-effector/COM alignment, velocity matching) reliably converge on static reference trajectories, they collapse under dynamic perturbations—and neither tuning action scales nor adjusting learning rates prevents torque collapse. Furthermore, survival and imitation objectives inherently conflict when processed under the same observation scaling: privileged reference inputs overwhelm the policy's native observations, inducing adversarial training dynamics and episodic reward collapse. Finally, forcing both precision- and robustness-focused signals through a single encoder exacerbates these tensions, as conflicting gradient directions derail convergence and prevent generalization to dynamic environments.

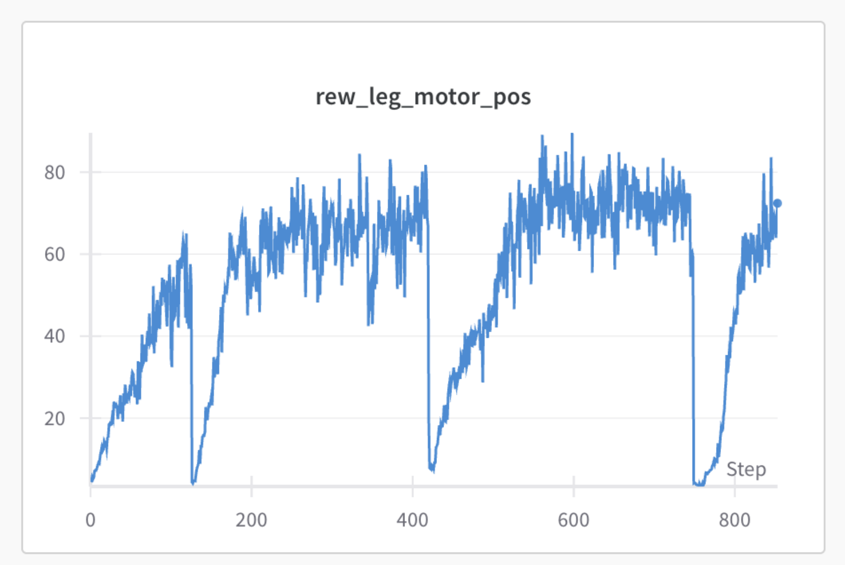


Figure 12: Arm and leg motor tracking rewards under exponential reward in the `fixed_env`.

7 Conclusion

Based on our experiments with ToddlerBot in Brax, the two-stage DeepMimic-inspired pipeline, pretraining on fixed reference motions followed by fine-tuning under survival and domain randomization—yields accurate motion tracking but suffers significant instability and catastrophic forgetting when faced with dynamic perturbations . These failures stem from competing imitation vs. survival objectives, a monolithic actor–critic architecture, and the absence of decoupled observation scaling. To overcome these limitations, we will migrate to the latest Brax release, leveraging its native support for asymmetric actor–critic architectures and built-in observation normalization. This will allow us to separate privileged reference inputs from policy observations and stabilize critic targets. Once the core training pipeline is robust, we will train a library of specialized expert behaviors (e.g., crawling, walking, climbing) and integrate them into a unified, egocentric vision-conditioned locomotion policy via behavior stitching or Adversarial Motion Priors (AMP). This approach paves the way toward versatile whole-body control of humanoid robots in unstructured environments and sets the stage for real-hardware deployment on the ToddlerBot platform.

8 Team Contributions

- **Tae Yang:** Assisted with reinforcement learning training and environment setup.
- **Daniel Jinag:** Developed the stereo depth estimation pipeline and camera calibration setup.
- **Zhicong Zhang:** Simulation environment setup and experiment implementation.

References

Figure AI. 2025. Natural and Reliable Humanoid Locomotion via End-to-End Reinforcement Learning. *Technical Report* (2025). https://figure.ai/blog/engineering-natural-and-reliable-humanoid-locomotion.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. 2015. Benchmarking in Manipulation Research: The YCB Object and Model Set and Benchmarking Protocols. *IEEE Robotics & Automation Magazine* 22, 3 (2015), 36–52. `https://doi.org/10.1109/MRA.2015.2448951` arXiv:1502.03143 [cs.RO].

Ming Chen, Zeyu Yang, Yifan Wu, and C. Karen Liu. 2025. A Gait-Conditioned RL Framework for Multi-Modal Humanoid Locomotion. *arXiv preprint arXiv:2505.08415* (2025).

Google Research. 2021. Brax: A Differentiable Physics Engine for Large-Scale RL. `https://github.com/google/brax`.

Manan Gupta et al. 2025. Humanoid-Gym: Training Realistic Locomotion for Humanoids in Isaac Sim. *arXiv preprint arXiv:2505.06612* (2025).

Yanchao He, Shen Tao, Tao Yang, and C. Karen Liu. 2024. Learning Real-Time Whole-Body Teleoperation of a Humanoid Robot with RGB Camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Zhen Li, Tao Han, Jun Wu, and C. Karen Liu. 2025. AMO: Adaptive Motion Optimization for Real-Time Humanoid Locomotion. *arXiv preprint arXiv:2504.07654* (2025).

Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. 2018. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

Xue Bin Peng, Tingwu Ma, and Sergey Levine. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–17.

Ilya Radosavovic, Tianhe Xiao, Bowen Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. 2023. Real-World Humanoid Locomotion with Reinforcement Learning. *arXiv preprint arXiv:2303.03381* (2023).

Juan Rodriguez, Lucas Perez, and Sandra Hirche. 2021. DeepWalk: Omnidirectional Humanoid Locomotion Using Deep Reinforcement Learning. *IEEE Robotics and Automation Letters* 6, 3 (2021), 4708–4715.

Haochen Shi, Weizhuo Wang, Shuran Song, and C Karen Liu. 2025. ToddlerBot: Open-Source ML-Compatible Humanoid Platform for Loco-Manipulation. *arXiv preprint arXiv:2502.00893* (2025).

William Thibault, William Melek, and Katja Mombaur. 2024. Learning Velocity-based Humanoid Locomotion: Massively Parallel Learning with Brax and MJX. *arXiv preprint arXiv:2407.05148* (2024).

Bowen Wen, Matthew Trepte, Joseph Aribido, Jan Kautz, Orazio Gallo, and Stan Birchfield. 2025. FoundationStereo: Zero-Shot Stereo Matching. *CVPR* (2025).

Yiyi Zhao, Ye Liu, Tao Yang, Yuke Li, and C. Karen Liu. 2025. SMAP: Self-Supervised Motion Adaptation for Physically Plausible Whole-Body Control of Humanoids. *arXiv preprint arXiv:2503.01234* (2025).