Learning Sim-to-Real Humanoid Locomotion in 15 Minutes

Younggyo Seo* **Carmelo Sferrazza*** **Juyue Chen**
Guanya Shi **Rocky Duan** **Pieter Abbeel**

Amazon FAR (Frontier AI & Robotics)

Abstract

Massively parallel simulation has reduced reinforcement learning (RL) training time for robots from days to minutes. However, achieving fast and reliable sim-to-real RL for humanoid control remains difficult due to the challenges introduced by factors such as high dimensionality and domain randomization. In this work, we introduce a simple and practical recipe based on off-policy RL algorithms, i.e., **FastSAC** and **FastTD3**, that enables rapid training of humanoid locomotion policies in just 15 minutes with a single RTX 4090 GPU. Our simple recipe stabilizes off-policy RL algorithms at massive scale with thousands of parallel environments through carefully tuned design choices and minimalist reward functions. We demonstrate rapid end-to-end learning of humanoid locomotion controllers on Unitree G1 and Booster T1 robots under strong domain randomization, e.g., randomized dynamics, rough terrain, and push perturbations, as well as fast training of whole-body human-motion tracking policies. We provide videos and open-source implementation at: https://younggyo.me/fastsac-humanoid.

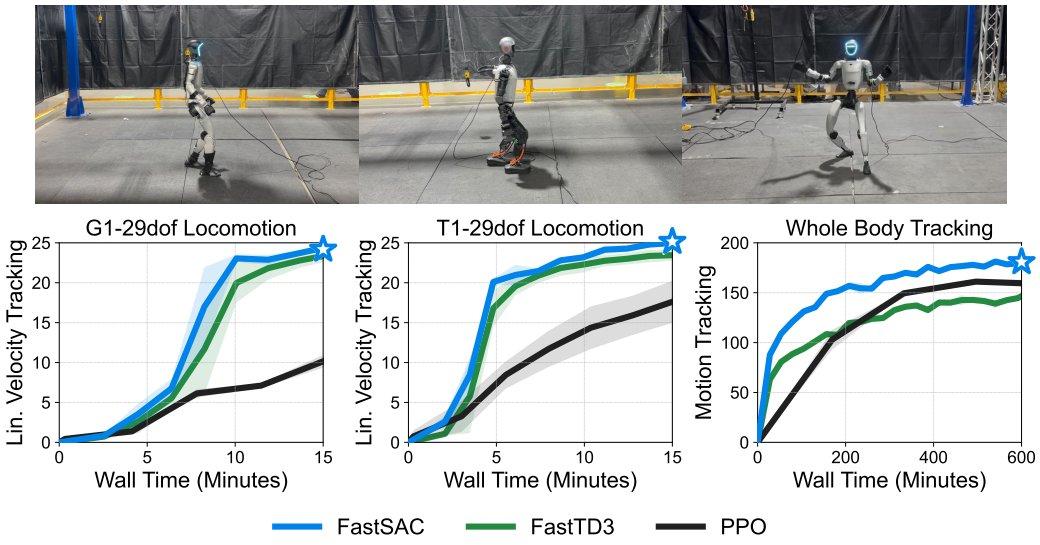


Figure 1: **Summary of results.** We introduce a simple recipe based on off-policy RL algorithms, i.e., FastSAC and FastTD3, that learns robust humanoid locomotion policies in 15 minutes on a single RTX 4090 GPU, with strong domain randomization including randomized dynamics, rough terrain, and push perturbations. We also show that our recipe based on off-policy RL algorithms is scalable and accelerates the training of whole-body tracking policies: trained with 4×L40s GPUs and 16384 parallel environments, FastSAC and FastTD3 learn to complete the full sequence of dancing motion much faster than PPO under the same condition. For sim-to-real deployment with Unitree G1 and Booster T1, we used the checkpoints saved at the points we marked as ★.

*Equal Contribution.

1 Introduction

In recent years, reinforcement learning (RL) has undergone a dramatic shift driven by the emergence of massively parallel simulation frameworks (Rudin et al., 2022; Kaufmann et al., 2023). By scaling environment throughput to thousands of environments, these frameworks have reduced wall-clock training time from many hours to mere minutes for a wide range of benchmark tasks (Makoviychuk et al., 2021; Mittal et al., 2023; Zakka et al., 2025). This shift has had an outsized impact on robotics, where sim-to-real development is inherently iterative: a policy is trained in simulation, deployed on hardware, and reveals mismatches such as unmodeled dynamics or sensing inaccuracies (Zhao et al., 2020). These discrepancies must then be corrected by improving the simulation environment, requiring the entire pipeline to be retrained (Chebotar et al., 2019). Because these cycles repeat until the policy is reliable, fast simulation becomes essential for making such iteration feasible.

Despite the speed offered by modern parallel simulators, these iterative cycles remain expensive in practice, especially for high-dimensional systems such as humanoids. Achieving robust transfer of policies to the real world typically requires expanding domain randomization (Sadeghi & Levine, 2016; Tobin et al., 2017; Peng et al., 2018b), randomizing terrain properties (Rudin et al., 2022), or shaping curricula that encourage low-effort, stable whole-body behavior. Such components complicate exploration and reduce sample efficiency, pushing training for humanoid locomotion or tracking back into the multi-hour regime. Thus, despite dramatic gains in raw throughput, achieving fast, reliable sim-to-real iteration for humanoid control remains a challenge.

This work introduces a simple and practical recipe that brings sim-to-real iteration time for humanoid robots back to the order of minutes. At the core of this recipe are FastSAC and FastTD3 (Seo et al., 2025), efficient variants of popular off-policy RL algorithms, i.e., Soft Actor-Critic (Haarnoja et al., 2018a) and TD3 (Fujimoto et al., 2018), which have been shown to learn humanoid control policies faster than on-policy RL algorithms such as PPO (Schulman et al., 2017). While Seo et al. (2025) demonstrated the first sim-to-real deployment of FastTD3 policies to real humanoid hardware, the results were limited to relatively simple controllers for humanoids with only a subset of joints. In this work, we show that with careful design choices and hyperparameters, FastSAC and FastTD3 can scale to full-body humanoid control, enabling rapid sim-to-real iterations for training locomotion policies with all joints or whole-body tracking policies that follow human motion.

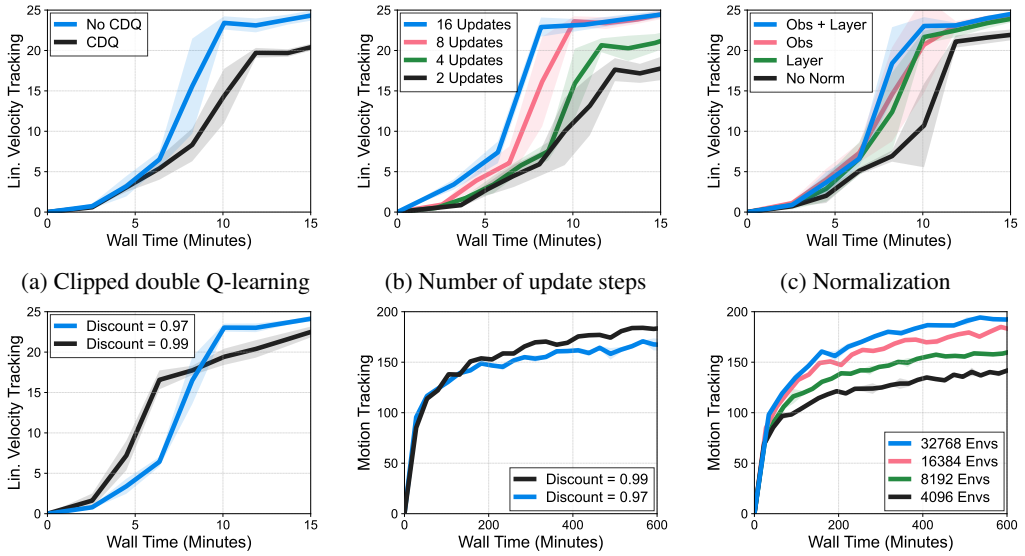
Another important aspect of our fast sim-to-real recipe is its simplicity in reward design. By adopting reward functions with essential terms, we can quickly sweep hyperparameters, isolate what matters for transfer, and avoid the brittle engineering often required in humanoid locomotion setups. With our recipe, we train a full-fledged humanoid locomotion policy with randomized dynamics, rough terrain, push perturbations, and an automatic action-rate curriculum, all end-to-end in 15 minutes on a single RTX 4090 GPU. The code for this recipe is available in the Holosoma repository (FAR et al., 2025) at `https://github.com/amazon-far/holosoma`.

2 Recipe

2.1 FastSAC and FastTD3: Off-Policy RL for Humanoid Control

Our recipe is based on off-policy RL algorithms tuned for large-scale training with massively parallel simulation, i.e., FastTD3 and FastSAC (Seo et al., 2025), instead of PPO (Schulman et al., 2017) that has been a standard algorithm for sim-to-real RL due to the ease of scaling up with parallel simulation. This is motivated by recent work that have demonstrated off-policy algorithms can also scale effectively and be faster than PPO in various benchmark tasks (Li et al., 2023; Raffin, 2025; Seo et al., 2025; Shukla, 2025) by effectively re-using the data from simulation. Notably, Seo et al. (2025) report the first sim-to-real deployment of humanoid control policies trained with off-policy RL to real humanoid hardware. However, its results were limited to humanoid robots with a subset of joints.

This section describes how we train FastTD3 and FastSAC to achieve full-body humanoid control, enabling rapid sim-to-real iterations for training locomotion policies with all joints or whole-body tracking policies that follow human motion. In particular, we improve the recipe of Seo et al. (2025) in training FastSAC, which learns how to explore environments from data with its maximum entropy learning scheme instead of deterministic policies with fixed noise schedules (FastTD3), mitigating the exploration challenges of training with strong domain randomization.



(a) Clipped double Q-learning (b) Number of update steps (c) Normalization

(d) Effect of γ (e) Effect of γ (WBT) (f) Number of environments (WBT)

Figure 2: **FastSAC: Analyses.** We investigate the effect of (a) Clipped double Q-learning, (b) number of update steps, (c) normalization techniques, and (d) discount factor γ on a Unitree G1 locomotion task with rough terrain. We further investigate the effect of (e) discount factor γ and (f) number of environments on a G1 whole-body tracking (WBT) task with a dancing motion. We use a single RTX 4090 GPU for locomotion experiments (a-d) and 4×L40s GPUs for whole-body tracking (e-f).

Scaling up off-policy RL with massively parallel simulation Similar to prior work (Li et al., 2023; Shukla, 2025; Raffin, 2025; Seo et al., 2025), we use massively parallel simulation for training FastSAC and FastTD3 agents. We find that the effect of using more environments is particularly visible in challenging whole-body tracking tasks (see Figure 2f). We also find that most of observations in Seo et al. (2025) with regard to scaling up off-policy RL also holds for full-body humanoid control as well. For instance, we find that using large batch size up to 8K consistently improves performance. We also find that taking more gradient steps per each simulation step usually ends up in a faster training, and slow simulation speed often becomes a bottleneck with more challenging setups such as training robots in non-flat terrains (see Figure 2b). This makes off-policy RL, which can re-use data from previous interactions instead of discarding it, a more attractive choice for fast training.

Joint-limit-aware action bounds One challenge in training off-policy RL algorithms such as SAC or TD3 is setting proper action bounds for its Tanh policy. For instance, Raffin (2025) observed that training often becomes unstable when trained in unbounded action space. To address this challenge, we introduce a simple technique that sets the action bounds based on the robots' joint limits when using PD controllers. In particular, we calculate the difference between each joint's limit and its default position, then use it as an action bound for each joint. We find that this effectively reduces the need to tune action bounds for training FastSAC and FastTD3[1].

Observation and Layer normalization Similar to Seo et al. (2025), we find that observation normalization is helpful for training. However, unlike Seo et al. (2025), we find that layer normalization (Ba et al., 2016) is helpful in stabilizing the performance in high-dimensional tasks (see Figure 2c). This is aligned with prior observations that find layer normalization is helpful for training SAC (Ball et al., 2023; Nauman et al., 2024) agents in challenging benchmark tasks.

[1]Interestingly, after we have fully stabilized the training with all the components, we find that we can achieve stable training of FastSAC and FastTD3 agents with an unbounded action space. Nonetheless, we still keep joint-limit-aware action bounds as we expect this scheme to be helpful in training off-policy RL agents for other robots or tasks. We recommend training agents in an unbounded action space when encountering a failure case where the robot is not generating enough torque due to restrictive action bounds.

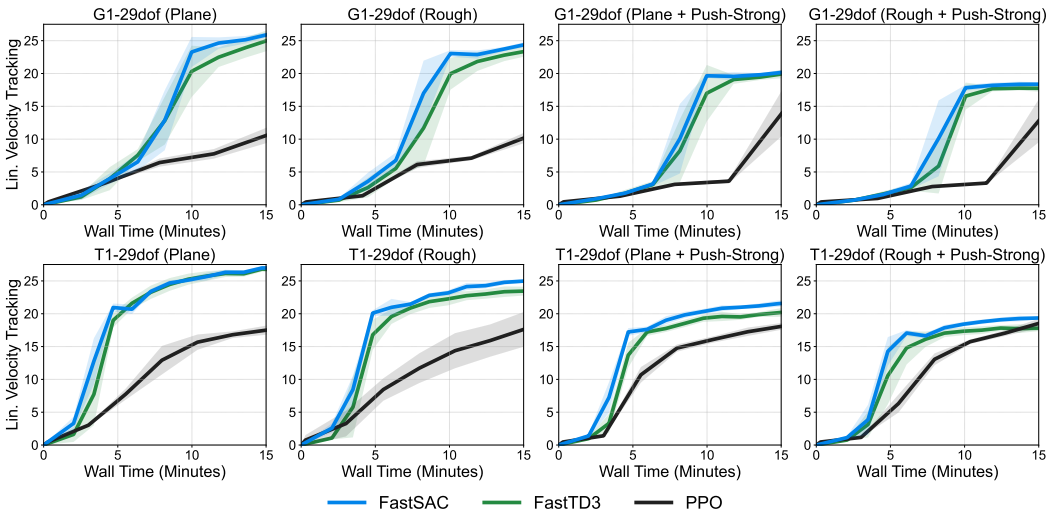


Figure 3: **Locomotion (velocity tracking) results.** FastSAC and FastTD3 enable fast training of G1 and T1 humanoid locomotion policies with strong domain randomization such as rough terrain or `Push-Strong` that applies push perturbations to humanoid robots every 1 to 3 seconds (max episode length is 20 seconds). For non-`Push-Strong` tasks, we apply push perturbations every 5 to 10 seconds. We use a single RTX 4090 GPU for all locomotion experiments.

Critic learning hyperparameters We find that using the average of Q-values improves FastSAC and FastTD3 performance over using Clipped double Q-learning (CDQ; Fujimoto et al. 2018) that uses the minimum (see Figure 2a). This aligns with the observation of Nauman et al. (2024) that shows CDQ is harmful when used with layer normalization. We find that low discount factor $\gamma = 0.97$ is helpful for simple velocity tracking tasks (see Figure 2d), while $\gamma = 0.99$ is helpful for challenging whole-body tracking tasks (see Figure 2e). Following prior work (Li et al., 2023; Seo et al., 2025), we also use a distributional critic, i.e., C51 (Bellemare et al., 2017). We find that distributional critic with quantile regression (Dabney et al., 2018) is too expensive in particular with large batch training.

FastSAC: Exploration hyperparameters A widely-used implementation of SAC bounds the standard deviation σ of the pre-tanh actions to be e^2 (Huang et al., 2022). However, we find that, when combined with large initial value of the temperature α, this sometimes causes instability due to excessive exploration. We instead set the maximum σ to be 1.0 and initialize α with the low value of 0.001. We also find that using auto-tuning for maximum entropy learning (Haarnoja et al., 2018b) consistently outperforms using the fixed alpha values. For the target entropy, we find that using 0.0 (for locomotion tasks) or $-|\mathcal{A}|/2$ (for whole-body tracking tasks) works best in practice.

FastTD3: Exploration hyperparameters Following prior work (Li et al., 2023; Seo et al., 2025), we use mixed noise schedule that randomly samples Gaussian noise standard deviation from the range $[\sigma_{\texttt{min}}, \sigma_{\texttt{max}}]$. We find that using low values, i.e., $(\sigma_{\texttt{min}}, \sigma_{\texttt{max}}) = (0.01, 0.05)$, performs the best.

Optimization hyperparameters We train FastSAC and FastTD3 using Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.0003. We find that weight decay 0.1, which Seo et al. (2025) uses, is a too strong regularization for high-dimensional control tasks and thus we use weight decay of 0.001. Similar to Zhai et al. (2023) where using low β_2 for Adam makes training stable with large batch sizes, we find that using $\beta_2 = 0.95$ slightly improves stability compared to using $\beta_2 = 0.99$.

Remark on additional techniques We expect that recent advances in improving off-policy RL (D'Oro et al., 2023; Schwarzer et al., 2023; Nauman et al., 2024; Lee et al., 2024; Sukhija et al., 2025; Lee et al., 2025; Obando-Ceron et al., 2025) will be helpful for further improving the performance and stability of FastSAC and FastTD3. However, this work aims to keep the recipe as simple as possible and we expect the research community to advance the state-of-the-art based on our recipe.

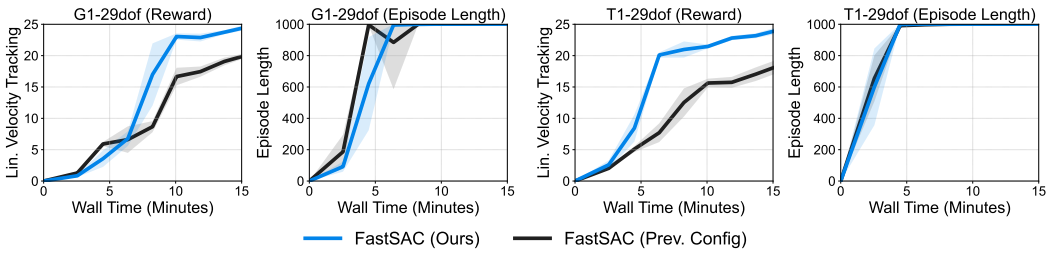


Figure 4: **Improvement from our FastSAC recipe.** While a version of FastSAC was previously considered as a baseline to FastTD3 (Seo et al., 2025) in the context of humanoid control, a straightforward implementation of FastSAC exhibited training instabilities. In this work, we have stabilized and improved FastSAC with a carefully tuned set of hyperparameters and design choices.

2.2 Simple Reward Design

Reward design for humanoid locomotion and whole-body control has traditionally depended on heavy reward shaping, often 20+ terms (Mittal et al., 2023; Lab, 2025), e.g., tracking rewards for kinematic quantities, detailed posture regularizer, penalties on joint configurations, foot placement constraints, and shaping terms that strictly prescribe how the robot should move. This complexity makes hyperparameter tuning difficult and often leads to brittle policy optimization.

Inspired by recent works that rely on much simpler reward functions (Zakka et al., 2025; Liao et al., 2025), we show that robust and natural behaviors can emerge from substantially simpler objectives (less than 10 terms). Specifically, we adopt a minimalist reward philosophy that only adds a reward term if necessarily needed, and aim to have a nearly identical set of rewards across algorithms and robots. Our goal is not to enforce a particular style, but to provide enough structure for robust locomotion and whole-body control while preserving behavioral richness. Fewer reward terms also simplify hyperparameter tuning, enabling rapid sweeps crucial for sim-to-real iteration.

Locomotion (velocity tracking) We use a compact set of reward terms that cover only the essential components needed for stable humanoid gait transferrable from simulation to the real-world:

- Linear and angular velocity tracking rewards to encourage the humanoid to follow commanded x-y speed and yaw rate. These are the main driver of emergent locomotion.
- A simple foot-height tracking term (Zakka et al., 2025; Shao et al., 2022) to guide swing motion.
- A default-pose penalty to avoid extreme joint configurations.
- Feet penalties to encourage parallel relative orientation and prevent foot crossing.
- A per-step alive reward that encourages remaining in valid, non-fallen states.
- Penalties that keep the torso near a stable upright orientation.
- A penalty on the action rate to smooth control outputs.

We terminate the episode on ground contact by the torso or other non-foot body parts. We also use symmetry augmentation (Mittal et al., 2024) to encourage symmetric walking pattern, which we also find to be helpful for faster convergence. All penalties above are subject to a curriculum that ramp up their weights over the course of training as the episode length increases (Lab, 2025), considerably simplifying exploration. We find that these terms are sufficient to produce robust locomotion across rough terrain, with randomized dynamics and external perturbations, without relying on extensive reward shaping or other carefully tuned heuristics, and are applicable across multiple robots (i.e., G1 and T1) and algorithms (i.e., FastSAC, FastTD3, and PPO).

Whole-body tracking For whole-body tracking, we follow the reward structure introduced in BeyondMimic (Liao et al., 2025), which already adheres to the same minimalist principles. These rewards are built around tracking goals with lightweight regularization, together with DeepMimic-style termination conditions (Peng et al., 2018a). We additionally find that introducing external disturbances in the form of velocity pushes further robustifies sim-to-real performance.

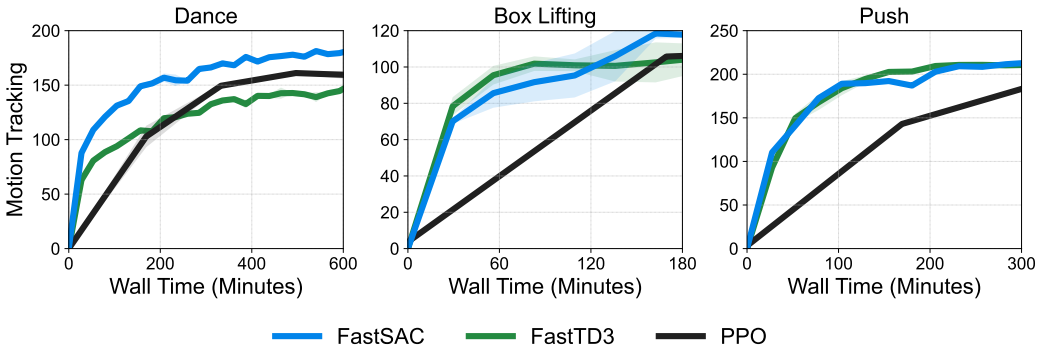


Figure 5: **Whole-body tracking results.** We show that FastSAC and FastTD3 are competitive or superior to PPO in whole-body motion tracking tasks. See Figure 6 for the sim-to-real deployment of FastSAC policies to real hardware. We use 4×L40s GPUs for all whole-body tracking experiments.

3 Experiments

3.1 Locomotion (Velocity Tracking)

Setup For locomotion tasks, we train RL policies to maximize the sum of reward as we described in Section 2.2, i.e., by training the robots to achieve the target linear and angular velocities while minimizing several penalty terms. Throughout training, we randomly sample target velocity commands every 10 seconds. When sampling the target commands, we randomly set the target velocities to zero with 20% probability, so that the robot learns to stand instead of making it constantly walk on its position. Unless otherwise specified, we train all robots on a mix of flat and rough terrains, which stabilizes robot walking in sim-to-real deployment. We apply various domain randomization techniques to further robustify sim-to-real deployment: push perturbations, action delay, PD-gain randomization, mass randomization, friction randomization, and center of mass randomization (only for G1). We report linear velocity tracking reward in all learning curves.

Results Figure 3 shows that FastSAC and FastTD3 quickly train G1 and T1 humanoid robots to track velocity commands in 15 minutes, significantly outperforming PPO in terms of wall-time clock. We emphasize this is achieved in the existence of strong domain randomization: our humanoids learn to stand and walk in rough terrain, with consistent push perturbations, action delay, center of mass randomization, etc. In particular, we observe that FastSAC and FastTD3 enables fast training of locomotion policies with strong domain randomization such as `Push-Strong` that applies push perturbations to robots every 1 to 3 seconds, while PPO struggles with such strong perturbations. We also find that FastSAC slightly outperforms FastTD3 in several locomotion setups, which we hypothesize to be due to efficient exploration through its maximum entropy exploration scheme.

FastSAC improvement over previous configuration Figure 4 shows how our recipe for training FastSAC improves the performance over the previous version of FastSAC trained with configuration from Seo et al. (2025). Specifically, we find that the use of layer normalization (Seo et al., 2025), disabling CDQ (Fujimoto et al., 2018), careful tuning of exploration and optimization hyperparameters is important for performance improvement (see Section 2 for details).

3.2 Whole-Body Tracking

Setup For whole-body tracking, we mostly follow the setup in BeyondMimic (Liao et al., 2025). We train RL policies to maximize the sum of rewards as we described in Section 2.2. We report the sum of tracking rewards in all learning curves. Throughout training, we randomly sample motion segments for each episode. Unlike BeyondMimic that minimizes the use of domain randomization for motion tracking, we find that using various domain randomization techniques stabilizes the behavior at deployment time. In particular, we randomize friction, center of mass, joint position bias, body mass, PD gains, and also apply push perturbations.

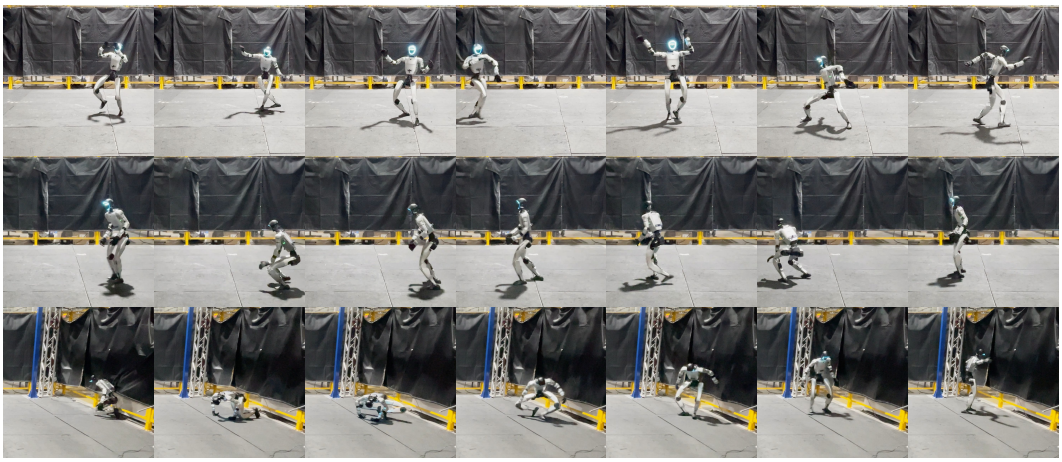


Figure 6: **Whole-body tracking examples.** We demonstrate the sim-to-real deployment of whole-body tracking controllers for Unitree G1 trained with FastSAC (Top: Dance, Middle: Box Lifting, Bottom: Push). Videos are available at `https://younggyo.me/fastsac-humanoid`.

Results Figure 5 shows that FastSAC and FastTD3 can also quickly train G1 humanoid robots to track human motion, being competitive or superior to PPO. We find that FastSAC outperforms FastTD3 in the `Dance` task, which is a longer motion compared to the other tasks we considered. We hypothesize better exploration through maximum entropy RL enables faster learning on more challenging tasks. Further investigation into the performance difference between FastSAC and FastTD3 in more diverse types of tasks is an interesting future direction.

Sim-to-real deployment In Figure 6, we further demonstrate the sim-to-real deployment of FastSAC whole-body tracking policies to real Unitree G1 humanoid hardware. We find that FastSAC policies can complete several motions including the long motion like `Dance` that lasts more than 2 minutes. These results show that FastSAC not only enables faster training in simulation but also actually learns deployable robust full-body humanoid control policies.

4 Related Work

Reinforcement learning with massively parallel simulation Massively parallel simulation has significantly reduced the wall-clock time required to train RL policies. Early work primarily relied on CPU-based parallelization by launching simulations across multiple processes (Heess et al., 2017; Akkaya et al., 2019; Stooke & Abbeel, 2018; Espeholt et al., 2018; Radosavovic et al., 2024). While policy learning often leveraged multiple GPUs, overall throughput remains bottlenecked by the process-management overhead and inherently slow simulation speed. To address these limitations, the idea of using GPU-based parallel environments have been proposed (Liang et al., 2018; Makoviychuk et al., 2021; Mittal et al., 2023; Authors, 2024; Zakka et al., 2025), scaling up environment throughput to thousands of environments. This has been the key driver of recent successes in training controllers for diverse robots with impressive capabilities (Rudin et al., 2022; Agarwal et al., 2023; Cheng et al., 2024; Singh et al., 2024; Zhuang et al., 2024; Li et al., 2025; He et al., 2025b,a). Building on this trend, our work focuses on accelerating sim-to-real iterations by combining massively parallel simulation with off-policy RL algorithms tuned for large-scale training regimes.

Algorithms for sim-to-real reinforcement learning Proximal policy optimization (PPO; Schulman et al. 2017) has been the de-facto standard algorithm for sim-to-real RL, and is often the only supported algorithm in widely used learning frameworks (Makoviychuk et al., 2021; Mittal et al., 2023; Zakka et al., 2025; Schwarke et al., 2025), largely due to the ease of scaling up on-policy RL with massively parallel environments. However, recent works have started to demonstrate that off-policy RL methods can also scale effectively in such large-scale training regimes (Li et al., 2023; Raffin, 2025; Shukla, 2025; Seo et al., 2025). Notably, Seo et al. (2025) report the first sim-to-real deployment of humanoid control policies trained with FastTD3, an efficient variant of TD3 (Fujimoto et al., 2018) optimized

Algorithm 1 FastSAC: Pseudocode (distributional critic is omitted for simplicity)

Initialize actor π_θ, two critics Q_{ϕ_1}, Q_{ϕ_2}, entropy temperature α, replay buffer $\mathcal{B}$
Initialize target critics $Q_{\phi_1^{\texttt{target}}}, Q_{\phi_2^{\texttt{target}}}$ with $\phi_1^{\texttt{target}} \leftarrow \phi_1$ and $\phi_2^{\texttt{target}} \leftarrow \phi_2$
for each environment step **do**
    Sample $a \sim \pi_\theta(o)$ given the current observation o, and take action a
    Observe next state o' and reward r'
    Store transition $\tau = (o, a, o', r')$ in replay buffer $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau\}$
    for $j = 1$ **to** `num_updates` **do**
        Sample mini-batch $B = \{\tau_k\}_{k=1}^{|B|}$ from $\mathcal{B}$
        Compute target Q-value via average:
        $$y = r' + \frac{\gamma}{2} \sum_{i=1}^{2} \Big(Q_{\phi_i^{\texttt{target}}}(o', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|o') \Big) \text{ with } \tilde{a}' \sim \pi_\theta(\cdot|o')$$
        Update critic:
        $$\phi_i \leftarrow \phi_i - \nabla_{\phi_i} \frac{1}{|B|} \sum_{\tau_k \in B} \Big(Q_{\phi_i}(o, a) - y \Big)^2 \text{ for } i \in \{1, 2\}$$
        Update actor with reparameterization trick:
        $$\theta \leftarrow \theta + \nabla_\theta \frac{1}{2|B|} \sum_{\tau_k \in B} \sum_{i=1}^{2} \Big(Q_{\phi_i}(o, \tilde{a}) - \alpha \log \pi_\theta(\tilde{a}|o) \Big) \text{ with } \tilde{a} \sim \pi_\theta(\cdot|o)$$
        Update entropy temperature:
        $$\alpha \leftarrow \alpha - \nabla_\alpha \frac{1}{|B|} \sum_{\tau_k \in B} (\mathcal{H}^{\texttt{target}} - \mathcal{H}(o)) \cdot \alpha$$
        Update target critic $\phi_i^{\texttt{target}} \leftarrow \rho \phi_i^{\texttt{target}} + (1 - \rho)\phi_i$ for $i \in \{1, 2\}$
    end for
end for

for large-batch training with parallel simulation. However, their results were limited to humanoid controllers only with a subset of joints. In this work, we further push this direction by developing a sim-to-real RL recipe based on FastSAC and FastTD3 that achieve full-body humanoid control that controls all joints for locomotion or follows human motion. While doing so, we have also stabilized and improved the performance of FastSAC, which has been shown to exhibit training instabilities for humanoid control in prior work (Seo et al., 2025), with careful design choices.

5 Conclusion

By combining FastSAC and FastTD3, scalable off-policy RL algorithms, with a streamlined training pipeline, our recipe closes the gap between the promise of high-throughput parallel simulation and the practical demands of sim-to-real humanoid learning. In particular, we show that off-policy RL algorithms can be scaled effectively to reduce sim-to-real iteration time for learning whole-body humanoid controllers. In this work, we have intentionally maintained a simple, minimalist design that other researchers can easily build upon. We expect that incorporating recent advances in off-policy RL and humanoid learning into this recipe will push the state-of-the-art even further. To support such progress, we provide an open-source implementation of our recipe (FAR et al., 2025). We hope this report serves as a blueprint for researchers aiming to rapidly iterate on humanoid policies.

References

Agarwal, Ananye, Kumar, Ashish, Malik, Jitendra, and Pathak, Deepak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, 2023.

Akkaya, Ilge, Andrychowicz, Marcin, Chociej, Maciek, Litwin, Mateusz, McGrew, Bob, Petron, Arthur, Paino, Alex, Plappert, Matthias, Powell, Glenn, Ribas, Raphael, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Authors, Genesis. Genesis: A generative and universal physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Ball, Philip J, Smith, Laura, Kostrikov, Ilya, and Levine, Sergey. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, 2023.

Bellemare, Marc G, Dabney, Will, and Munos, Rémi. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, 2017.

Chebotar, Yevgen, Handa, Ankur, Makoviychuk, Viktor, Macklin, Miles, Issac, Jan, Ratliff, Nathan, and Fox, Dieter. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

Cheng, Xuxin, Shi, Kexin, Agarwal, Ananye, and Pathak, Deepak. Extreme parkour with legged robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Dabney, Will, Rowland, Mark, Bellemare, Marc, and Munos, Rémi. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

D'Oro, Pierluca, Schwarzer, Max, Nikishin, Evgenii, Bacon, Pierre-Luc, Bellemare, Marc G, and Courville, Aaron. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *International Conference on Learning Representations*, 2023.

Espeholt, Lasse, Soyer, Hubert, Munos, Remi, Simonyan, Karen, Mnih, Vlad, Ward, Tom, Doron, Yotam, Firoiu, Vlad, Harley, Tim, Dunning, Iain, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, 2018.

FAR, Amazon, Abbeel, Pieter, Chen, Juyue, Duan, Rocky, Escontrela, Alejandro, Gandhi, Manan, Gundry, Samuel, Huang, Xiaoyu, Kanazawa, Angjoo, Lewicki, Tomasz, Li, Jiaman, Liu, Karen, Rosenthal, Clay, Seo, Younggyo, Sferrazza, Carlo, Shi, Guanya, Shih, Linda, Tseng, Jonathan, Wu, Zhen, Yang, Lujie, Yi, Brent, and Zhang, Yuanhang. Holosoma, 2025. URL https://github.com/amazon-far/holosoma.

Fujimoto, Scott, Hoof, Herke, and Meger, David. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

Haarnoja, Tuomas, Ha, Sehoon, Zhou, Aurick, Tan, Jie, Tucker, George, and Levine, Sergey. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018a.

Haarnoja, Tuomas, Zhou, Aurick, Hartikainen, Kristian, Tucker, George, Ha, Sehoon, Tan, Jie, Kumar, Vikash, Zhu, Henry, Gupta, Abhishek, Abbeel, Pieter, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

He, Tairan, Gao, Jiawei, Xiao, Wenli, Zhang, Yuanhang, Wang, Zi, Wang, Jiashun, Luo, Zhengyi, He, Guanqi, Sobanbab, Nikhil, Pan, Chaoyi, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. In *Robotics: Science and Systems*, 2025a.

He, Tairan, Xiao, Wenli, Lin, Toru, Luo, Zhengyi, Xu, Zhenjia, Jiang, Zhenyu, Kautz, Jan, Liu, Changliu, Shi, Guanya, Wang, Xiaolong, et al. Hover: Versatile neural whole-body controller for humanoid robots. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025b.

Heess, Nicolas, Tb, Dhruva, Sriram, Srinivasan, Lemmon, Jay, Merel, Josh, Wayne, Greg, Tassa, Yuval, Erez, Tom, Wang, Ziyu, Eslami, SM, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Huang, Shengyi, Dossa, Rousslan Fernand Julien, Ye, Chang, Braga, Jeff, Chakraborty, Dipam, Mehta, Kinal, and Araújo, João G.M. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Kaufmann, Elia, Bauersfeld, Leonard, Loquercio, Antonio, Müller, Matthias, Koltun, Vladlen, and Scaramuzza, Davide. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Lab, CMU LeCAR. Humanoidverse: A multi-simulator framework for humanoid robot sim-to-real learning. https://github.com/LeCAR-Lab/HumanoidVerse, 2025.

Lee, Hojoon, Hwang, Dongyoon, Kim, Donghu, Kim, Hyunseung, Tai, Jun Jet, Subramanian, Kaushik, Wurman, Peter R, Choo, Jaegul, Stone, Peter, and Seno, Takuma. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. In *International Conference on Learning Representations*, 2024.

Lee, Hojoon, Lee, Youngdo, Seno, Takuma, Kim, Donghu, Stone, Peter, and Choo, Jaegul. Hyperspherical normalization for scalable deep reinforcement learning. In *International Conference on Machine Learning*, 2025.

Li, Zechu, Chen, Tao, Hong, Zhang-Wei, Ajay, Anurag, and Agrawal, Pulkit. Parallel q-learning: Scaling off-policy reinforcement learning under massively parallel simulation. In *International Conference on Machine Learning*, pp. 19440–19459. PMLR, 2023.

Li, Zhongyu, Peng, Xue Bin, Abbeel, Pieter, Levine, Sergey, Berseth, Glen, and Sreenath, Koushil. Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control. *The International Journal of Robotics Research*, 2025.

Liang, Jacky, Makoviychuk, Viktor, Handa, Ankur, Chentanez, Nuttapong, Macklin, Miles, and Fox, Dieter. Gpu-accelerated robotic simulation for distributed reinforcement learning. In *Conference on Robot Learning*, 2018.

Liao, Qiayuan, Truong, Takara E, Huang, Xiaoyu, Tevet, Guy, Sreenath, Koushil, and Liu, C Karen. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241*, 2025.

Makoviychuk, Viktor, Wawrzyniak, Lukasz, Guo, Yunrong, Lu, Michelle, Storey, Kier, Macklin, Miles, Hoeller, David, Rudin, Nikita, Allshire, Arthur, Handa, Ankur, and State, Gavriel. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

Mittal, Mayank, Yu, Calvin, Yu, Qinxi, Liu, Jingzhou, Rudin, Nikita, Hoeller, David, Yuan, Jia Lin, Singh, Ritvik, Guo, Yunrong, Mazhar, Hammad, Mandlekar, Ajay, Babich, Buck, State, Gavriel, Hutter, Marco, and Garg, Animesh. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

Mittal, Mayank, Rudin, Nikita, Klemm, Victor, Allshire, Arthur, and Hutter, Marco. Symmetry considerations for learning task symmetric robot policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7433–7439. IEEE, 2024.

Nauman, Michal, Ostaszewski, Mateusz, Jankowski, Krzysztof, Miłoś, Piotr, and Cygan, Marek. Bigger, regularized, optimistic: scaling for compute and sample-efficient continuous control. In *Advances in Neural Information Processing Systems*, 2024.

Obando-Ceron, Johan, Mayor, Walter, Lavoie, Samuel, Fujimoto, Scott, Courville, Aaron, and Castro, Pablo Samuel. Simplicial embeddings improve sample efficiency in actor-critic agents. *arXiv preprint arXiv:2510.13704*, 2025.

Peng, Xue Bin, Abbeel, Pieter, Levine, Sergey, and Van de Panne, Michiel. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 2018a.

Peng, Xue Bin, Andrychowicz, Marcin, Zaremba, Wojciech, and Abbeel, Pieter. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810. IEEE, 2018b.

Radosavovic, Ilija, Kamat, Sarthak, Darrell, Trevor, and Malik, Jitendra. Learning humanoid locomotion over challenging terrain. *arXiv preprint arXiv:2410.03654*, 2024.

Raffin, Antonin. Getting sac to work on a massive parallel simulator: An rl journey with off-policy algorithms. *araffin.github.io*, Feb 2025. URL https://araffin.github.io/post/sac-massive-sim/.

Rudin, Nikita, Hoeller, David, Reist, Philipp, and Hutter, Marco. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 91–100. PMLR, 2022. URL https://proceedings.mlr.press/v164/rudin22a.html.

Sadeghi, Fereshteh and Levine, Sergey. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

Schulman, John, Wolski, Filip, Dhariwal, Prafulla, Radford, Alec, and Klimov, Oleg. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schwarke, Clemens, Mittal, Mayank, Rudin, Nikita, Hoeller, David, and Hutter, Marco. Rsl-rl: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Schwarzer, Max, Ceron, Johan Samir Obando, Courville, Aaron, Bellemare, Marc G, Agarwal, Rishabh, and Castro, Pablo Samuel. Bigger, better, faster: Human-level atari with human-level efficiency. In *International Conference on Machine Learning*, 2023.

Seo, Younggyo, Sferrazza, Carmelo, Geng, Haoran, Nauman, Michal, Yin, Zhao-Heng, and Abbeel, Pieter. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

Shao, Yecheng, Jin, Yongbin, Liu, Xianwei, He, Weiyan, Wang, Hongtao, and Yang, Wei. Learning free gait transition for quadruped robots via phase-guided controller. volume abs/2201.00206, 2022. URL https://arxiv.org/abs/2201.00206.

Shukla, Arth. Speeding up sac with massively parallel simulation. *https://arthshukla.substack.com*, Mar 2025. URL https://arthshukla.substack.com/p/speeding-up-sac-with-massively-parallel.

Singh, Ritvik, Allshire, Arthur, Handa, Ankur, Ratliff, Nathan, and Van Wyk, Karl. Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024.

Stooke, Adam and Abbeel, Pieter. Accelerated methods for deep reinforcement learning. *arXiv preprint arXiv:1803.02811*, 2018.

Sukhija, Bhavya, Coros, Stelian, Krause, Andreas, Abbeel, Pieter, and Sferrazza, Carmelo. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. In *International Conference on Learning Representations*, 2025.

Tobin, Josh, Fong, Rachel, Ray, Alex, Schneider, Jonas, Zaremba, Wojciech, and Abbeel, Pieter. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Zakka, Kevin, Tabanpour, Baruch, Liao, Qiayuan, Haiderbhai, Mustafa, Holt, Samuel, Luo, Jing Yuan, Allshire, Arthur, Frey, Erik, Sreenath, Koushil, Kahrs, Lueder A, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025.

Zhai, Xiaohua, Mustafa, Basil, Kolesnikov, Alexander, and Beyer, Lucas. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Zhao, Wenshuai, Queralta, Jorge Peña, and Westerlund, Tomi. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, 2020.

Zhuang, Ziwen, Yao, Shenzhe, and Zhao, Hang. Humanoid parkour learning. *arXiv preprint arXiv:2406.10759*, 2024.