Extended Abstract

Motivation Simulation-to-Simulation (Sim2Sim) transfer offers a practical way to evaluate how well reinforcement learning (RL) policies generalize across different physics engines, which is essential for reliable deployment in the real world. Despite significant progress in Sim2Real research, discrepancies between simulators such as Isaac Gym and MuJoCo still lead to unstable or failed policy transfers. Understanding and reducing this Sim2Sim gap not only provides a controlled testbed for studying generalization but also helps in identifying what makes a policy robust before attempting Sim2Real deployment.

Method We train locomotion policies for the Unitree Go2 and H1_2 robots using the PPO algorithm within the Isaac Gym simulator, leveraging the Legged Gym framework for efficient GPU-based parallel training. To improve transferability, we apply domain randomization by perturbing environment parameters such as friction, base mass, and external forces during training. After convergence, we evaluate the policies in MuJoCo without any fine-tuning. Our results show that domain randomization significantly improves zero-shot transfer for the Go2 quadruped, with more modest effects on the humanoid, suggesting the importance of task complexity and dynamic sensitivity in Sim2Sim generalization.

Implementation We use the Legged Gym framework built on Isaac Gym for high-throughput training of locomotion policies on the Unitree Go2 and H1_2 robots. Our policies are trained using PPO with a shared 3-layer MLP actor-critic architecture. During training, we apply domain randomization by varying physical parameters such as friction coefficient, base mass, and applying external lateral perturbations. All policies are trained for about 4 million environment steps on an RTX 4060 GPU. After convergence, we deploy the trained policies in the MuJoCo simulator without fine-tuning, to evaluate zero-shot Sim2Sim transfer performance.

Results Experimental results show that domain randomization improves transfer robustness for the Go2 quadruped. While training in Isaac Gym, policies with DR exhibit more variability and slightly lower episode rewards. However, when deployed in MuJoCo, the DR-trained policy achieves significantly higher reward (+495 vs. -849) and stable walking behavior. In contrast, the policy trained without DR fails to maintain balance. For the H1_2 humanoid, domain randomization shows minimal effect under our current setup, possibly due to task simplicity or limited DR coverage.

Discussion Our findings indicate that Domain Randomization (DR) significantly improves Sim2Sim transfer for the Unitree Go2 quadruped, but has limited impact on the H1_2 humanoid robot. This suggests that DR effectiveness depends on robot morphology and task complexity. DR improves generalization by reducing overfitting to the source simulator, although it slows convergence and increases training variance. These trade-offs highlight the need for task-specific DR strategies. For more complex systems, expanding DR to include observation noise, terrain perturbation, or adaptive randomization may be necessary for further gains.

Conclusion We show that domain randomization is a powerful tool for improving policy robustness in Sim2Sim transfer settings. Our experiments demonstrate that DR-trained policies generalize better across simulators without fine-tuning. While effective for dynamic quadrupeds, DR's benefits are less clear for simpler or less sensitive tasks. This points to the importance of customizing DR strategies. Our work offers insights into cross-simulator generalization and lays the groundwork for future efforts in bridging the Sim2Real gap using scalable, simulator-agnostic training pipelines.

Sim2Sim on Legged Robots

Jiaqi Shao
Department of Mechanical Engineering
Stanford University
jiaqis7@stanford.edu

Chenhao Zhu
Department of Mechanical Engineering
Stanford University
chenhzhu@stanford.edu

Yizhao Hou
Department of Mechanical Engineering
Stanford University
yzhou01@stanford.edu

Abstract

In this work, we investigate Simulation-to-Simulation (Sim2Sim) transfer in the context of robotic reinforcement learning (RL). While Sim2Real has received significant attention, Sim2Sim remains relatively under-explored despite offering an important and controlled setting to study generalization across environments. We focus on transferring locomotion policies trained in Isaac Gym to MuJoCo, two simulators with distinct dynamics and physics modeling. We apply Domain Randomization (DR) during training to expose the policy to diverse environmental configurations and assess its zero-shot transfer capability to the target simulator. Furthermore, we lay the groundwork for Active Domain Randomization (ADR), which intelligently selects challenging environment parameters during training. Our experiments with Unitree Go2 and H1_2 robots demonstrate that DR significantly enhances Sim2Sim performance for quadruped locomotion. We analyze key transfer bottlenecks, identify effective randomization dimensions, and propose future enhancements toward scalable simulator-agnostic policy learning.

1 Introduction

One of the biggest challenges in deploying reinforcement learning (RL) in the real world is ensuring that policies trained in simulation can generalize to different settings. In practice, policies that perform well in one simulator often fail when transferred to another. This issue stems from differences in how various simulators model physical interactions, dynamics, and sensory feedback. For instance, contact models may differ in how they compute forces during collisions, whether they assume rigid or soft bodies, or how they implement friction and damping. On the dynamics side, simulators might use different numerical integrators (e.g., semi-implicit Euler versus Runge-Kutta), adopt different time-step sizes, or vary in how they deal with instability and constraint violations. Observation pipelines also vary widely, with differences in sensor latency, noise profiles, resolution, and update rates. These inconsistencies, while sometimes subtle, can have a significant impact on the performance and stability of learned policies when transferred between simulators.

Although much of the literature focuses on transferring policies from simulation to the real world (Sim2Real), the intermediate step of Simulation-to-Simulation (Sim2Sim) transfer is equally important. Sim2Sim allows researchers to study generalization in a more controlled environment, where differences between simulators are easier to isolate and analyze. For example, transferring a policy from Isaac Gym to MuJoCo or PyBullet provides insight into how sensitive the policy is to specific modeling assumptions. These cross-simulator experiments help diagnose failure cases early, before moving to the more unpredictable real world.

Stanford CS224R 2025 Final Report

In this project, we examine how Domain Randomization (DR) can help address the Sim2Sim transfer problem. DR involves training policies in environments where physical and sensory parameters are randomized at each episode. The idea is that exposure to a wider variety of conditions during training leads to more robust behaviors at test time. In addition to traditional DR, we also consider Active Domain Randomization (ADR), where a separate process learns to focus the randomization on the most challenging and informative regions of the environment space. By applying these methods to both quadruped (Unitree Go2, see Figure 1) and humanoid (Unitree H1_2, see Figure 2) robots, we aim to better understand which aspects of DR and ADR are most effective for enabling transfer between simulators.

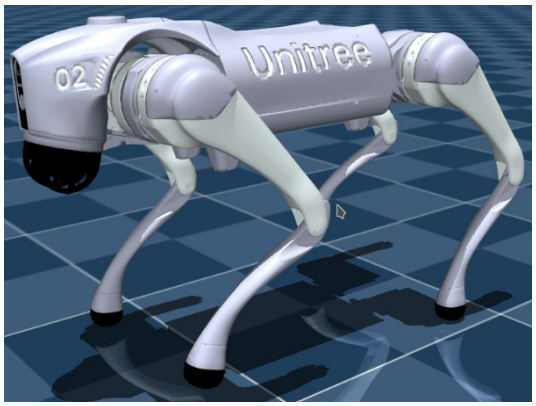

Figure 1: Unitree Go2 robot

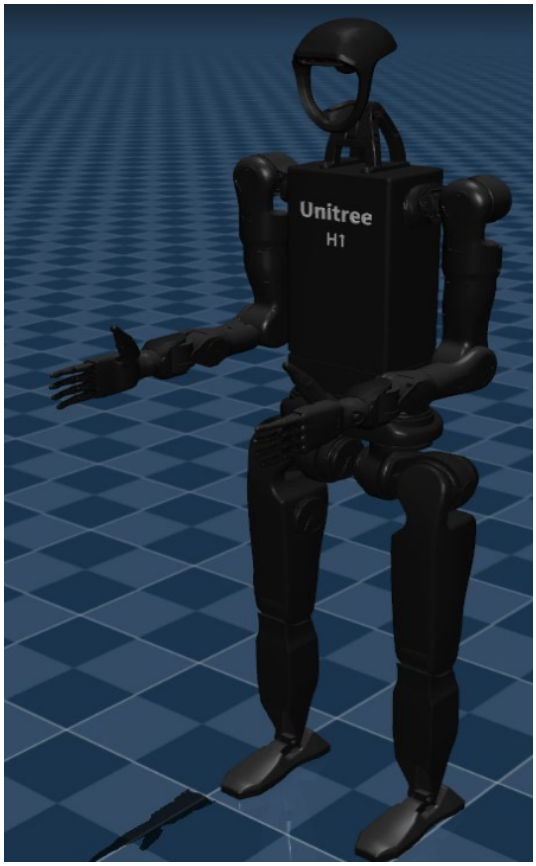

Figure 2: Unitree H1_2 humanoid robot

2 Related Work

Our research draws from several core areas in reinforcement learning and robotic simulation, with a particular focus on policy optimization, simulation engines, and domain generalization.

Proximal Policy Optimization (PPO) Schulman et al. (2017) has emerged as a reliable and efficient reinforcement learning algorithm, especially for continuous control tasks. It introduces a clipped surrogate objective that restricts how much a new policy can deviate from the old one during an update, which helps maintain learning stability. PPO strikes a balance between trust region methods and vanilla policy gradients, making it widely applicable in robotics tasks like locomotion and manipulation. Its robustness and simplicity have made it a go-to choice for policy training in many simulation-based robotics studies, including our own.

Simulation platforms are another key component of our work. Isaac Gym Makoviychuk et al. (2021) is a high-performance physics simulator that leverages GPU acceleration to enable massive parallelization. Its integration with PyTorch and ability to simulate thousands of environments simultaneously significantly speeds up training. These features are particularly useful when applying domain randomization, as they allow for a broader range of variability in less time. Meanwhile, MuJoCo Todorov et al. (2012) has become a standard in the field for its accurate modeling of multi-body dynamics and contact physics. Its solver architecture is well-suited for simulating complex robot interactions with the environment, offering a good benchmark for assessing how well policies trained in a faster simulator like Isaac Gym transfer to a more precise one.

Our experiments are built upon the Legged Gym framework Rudin et al. (2022), which provides a clean and flexible setup for training locomotion controllers using Isaac Gym. It includes ready-to-use environments for legged robots, reward function templates, and support for curriculum learning and domain randomization. The framework has shown that it is possible to train stable walking policies in minutes using GPU-based simulation, and its modular design makes it easy to test transfer scenarios like those in our project.

We also rely on theoretical guidance from the domain randomization literature. Chen et al. Chen et al. (2022) present a formal framework that treats simulators as parameterized Markov Decision Processes (MDPs). They analyze how discrepancies between training and test environments affect policy performance and offer theoretical bounds on the generalization gap. Their work supports the intuition that policies trained with sufficient variability—e.g., through randomized physical properties or observation noise—are more likely to succeed when transferred to new environments.

Taken together, these tools and insights shape our approach to studying Sim2Sim transfer. By combining scalable training with Isaac Gym, realistic evaluation in MuJoCo, structured experimentation through Legged Gym, and theoretical grounding from domain randomization research, we aim to better understand what makes a policy robust across simulators and where current methods fall short.

3 Method

Sim2Sim transfer addresses the challenge of deploying reinforcement learning (RL) policies across simulation environments that differ in physics engines, observation models, and rendering fidelity. In our project, we focus on transferring walking policies trained in NVIDIA Isaac Gym to MuJoCo. These simulators differ substantially in contact modeling, numerical integration, and physics approximations. We explore Domain Randomization (DR) as the primary method to bridge this gap and improve policy generalization. Our pipeline integrates standard DR techniques and sets the foundation for future Active Domain Randomization (ADR) (Mehta et al., 2019).

We build our experiments on the Legged Gym framework, which is based on Isaac Gym and provides optimized environments for training legged robots. We train policies for two platforms: the Unitree Go2 quadruped and the Unitree H1_2 humanoid. Policy learning is handled by Proximal Policy Optimization (PPO), using a shared 3-layer multilayer perceptron (MLP) for both the actor and critic, with layer sizes of 256, 128, and 64, and tanh activations. The input observations consist of joint positions and velocities, the robot base orientation, gravity vector projections, and velocity commands. The action space consists of 12 joint position targets that are updated at every control step. The reward function encourages forward walking while discouraging high energy usage, slipping, and unstable base orientations.

To promote generalization across simulators, we apply domain randomization during training by modifying environmental parameters at the start of each episode. Specifically, we:

- Randomize the friction coefficient in the range [0.5, 1.5].
- Vary the base mass of the robot by $\pm 10\%$.
- Apply random external lateral pushes during rollout.

These changes introduce a range of physically plausible disturbances that encourage the policy to learn behaviors that are robust to variability.

We train each policy twice: once with domain randomization and once without, keeping all other parameters fixed. Once trained in Isaac Gym, each policy is directly evaluated in MuJoCo without any further tuning. For the Go2 robot, we find a clear difference in performance: policies trained without DR tend to collapse or fail to make progress. In contrast, policies trained with DR are able to walk stably. For the H1_2 humanoid, we observe less pronounced differences, likely due to the relatively simple nature of flat-ground walking or the limited variety in training disturbances.

These results highlight the value of domain randomization for transferring policies across simulators, particularly for systems with sensitive dynamics like quadrupeds. As next steps, we plan to introduce additional sources of variation such as observation noise, control latency, and terrain irregularities. We also aim to implement ADR, where a teacher policy selects environment parameters that are maximally challenging for the student policy. By combining DR and ADR, we hope to push toward policies that not only transfer well between simulators, but are also better prepared for eventual deployment in the real world.

4 Experimental Setup

To study the effects of domain randomization (DR) on Sim2Sim transfer in robot locomotion, we design a controlled experimental pipeline with two simulation platforms: Isaac Gym and MuJoCo. As shown in Figure 3, our goal is to train walking policies in Isaac Gym and evaluate their generalization in MuJoCo under varying domain randomization conditions.

4.1 Training Environment: Isaac Gym with Legged Gym

We use the Legged Gym framework (Rudin et al., 2022), which provides modular and scalable tools for training locomotion policies on legged robots. The training is conducted within NVIDIA Isaac Gym (Makoviychuk et al., 2021), a GPU-accelerated physics engine that enables massive parallelism—allowing us to simulate up to 4096 environments in parallel. This high-throughput simulation is critical for training robust policies efficiently.

We train two types of robots:

- **Unitree Go2**: a quadruped robot with 12 actuated joints.
- **Unitree H1_2**: a humanoid robot with more complex dynamics and higher instability.

We use the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) with a 3-layer multilayer perceptron (MLP) network with layer sizes of 256, 128, and 64. Tanh activations are applied at each layer. The architecture shares parameters between the actor and critic heads.

4.2 Observation and Action Spaces

The observation space includes:

- Joint positions and velocities,
- Base orientation,
- Projected gravity vector
- Commanded linear and angular velocities,.

The action space consists of target joint positions for all 12 motors, updated at every control timestep using the output of the PPO policy.

4.3 Reward Design

The reward function used in training is adapted from the default Legged Gym setup and includes components for:

- Forward velocity tracking,
- Torque and foot slip penalties,
- Base stabilization (e.g., limiting excessive pitch/roll),
- Foot contact regularity.

These reward components are designed to encourage stable, energy-efficient locomotion that aligns with the target motion command.

4.4 Evaluation Environment: MuJoCo

After training, the learned policy is directly deployed in the MuJoCo simulator (Todorov et al., 2012), a well-established physics engine with a different numerical integration scheme and contact model compared to Isaac Gym. This provides a natural Sim2Sim transfer benchmark.

The policies are evaluated in MuJoCo without any fine-tuning. We monitor:

- Average cumulative episode reward,
- Locomotion stability (e.g., whether the robot falls or maintains balance),
- Visual inspection of foot placement and body motion.

4.5 Evaluation Criteria

We compare the performance of policies trained:

1. **Without Domain Randomization (No DR)**: trained in idealized conditions in Isaac Gym.
2. **With Domain Randomization (Full DR)**: trained with the above randomization parameters applied.

Performance is primarily measured using the cumulative reward per episode in MuJoCo, as well as qualitative metrics like walking stability and recovery behavior under perturbations.

4.6 Hardware and Runtime

All training experiments were conducted on NVIDIA RTX 4060 GPUs with parallel simulation enabled. Training each policy to convergence required approximately about 4 million environment steps, corresponding to 1–2 hours of wall-clock time due to GPU acceleration.

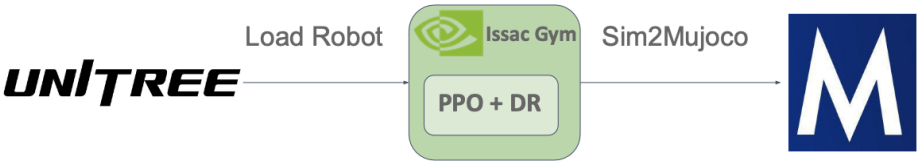

Figure 3: Pipeline of Sim-to-Sim Transfer

5 Results

We evaluate the impact of domain randomization (DR) on Sim2Sim transfer by comparing policies trained with and without DR in Isaac Gym, and tested in MuJoCo. Both the Go2 and the H1_2 robots are tested. However, the H1_2 humanoid robot shows little difference with and without the domain randomization method. That is, the humanoid robot walks successfully and steadily in the Mujoco environment even without domain randomization when trained in Isaac Gym. Noticeably, when we decrease the number of training epochs to downgrade the training performance, it still cannot distinguish the effect with and without domain randomization: the robots either fall together or be able to walk together under the same number of training epochs for both DR and non-DR cases.

Therefore, in the following analysis, we focus only on the experimental results of the Go2 robot, which exhibits a clear performance gap between the two settings. Our evaluation is divided into quantitative and qualitative results to better illustrate the impact of DR on policy transferability.

5.1 Quantitative Evaluation

Figure 4 and Figure 5 present the training curves of the Go2 robot with and without Domain Randomization (DR) in Isaac Gym. The Figure 4 shows the average number of steps per episode during training, and the Figure 5 shows the mean reward per episode.

We observe that:

- The policy trained without DR (blue curve) is more stable when reach convergence and reaches a marginally higher final reward. This suggests better overfitting to the specific training conditions in Isaac Gym.
- The policy trained with DR (orange curve) shows higher variance of episodes and lower rewards throughout training, indicating greater exploration and exposure to diverse dynamics. Despite slower convergence, it achieves comparable final performance in Isaac Gym.

However, this trend reverses during evaluation in MuJoCo. The DR-trained policy generalizes better to the target simulator, achieving a significantly higher reward (495 vs. -849), which highlights the benefit of introducing variability during training to enhance policy robustness.

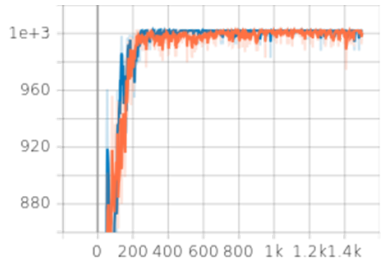

Figure 4: Training Mean Episode for Go2 With DR (Orange) and Without (Blue) DR

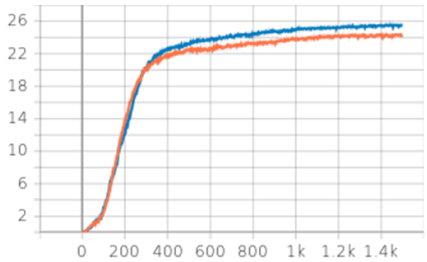

Figure 5: Training Mean Reward for Go2 With DR (Orange) and Without (Blue) DR

5.2 Qualitative Analysis

The qualitative results from MuJoCo evaluation are illustrated in Figure 6 and Figure 7. These screenshots show a stark contrast between policies trained with and without DR.

Without domain randomization (Figure 6), the Go2 robot quickly loses balance and turns over after deployment in MuJoCo. The lack of environmental perturbation during training results in a brittle policy that cannot adapt to slight dynamics discrepancies between simulators.

In contrast, the policy trained with DR (Figure 7) is able to walk stably and maintain balance throughout the episode. The robustness learned through exposure to randomized friction, base mass, and external perturbations allows the policy to transfer successfully despite simulator differences.

These qualitative results support our hypothesis that domain randomization improves Sim2Sim generalization and provides a practical pathway toward robust sim-to-real deployment.

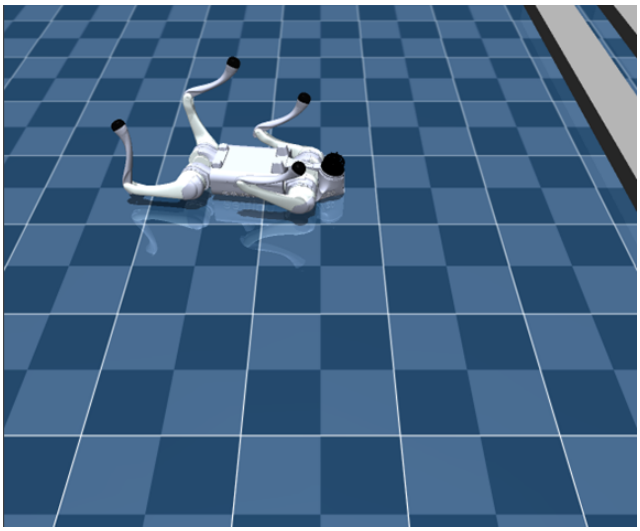

Figure 6: Testing result screenshot for Go2 without DR. Go2 robot soon turns over in Mujoco environment and cannot recover

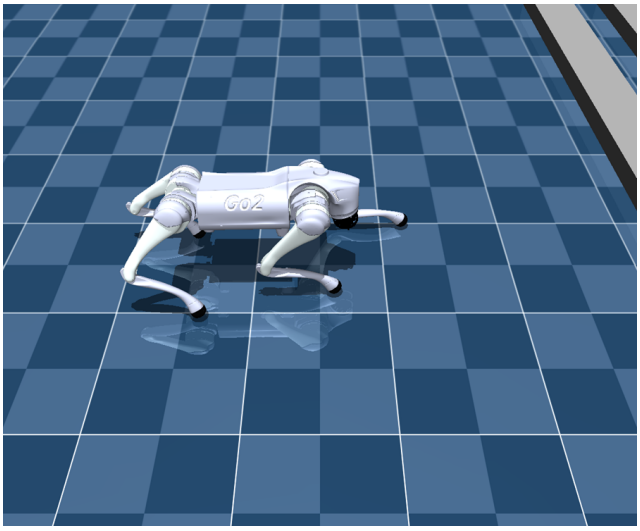

Figure 7: Testing result screenshot for Go2 with DR. The robot is able to walk stably in Mujoco environment

6 Discussion

Our experiments reveal that **Domain Randomization (DR)** can substantially enhance policy transferability in Sim2Sim settings, but its effectiveness is highly *task- and morphology-dependent*. Specifically, DR significantly improved the performance of the Unitree Go2 quadruped robot. By injecting random perturbations such as friction variation, mass changes, and lateral pushes during training in Isaac Gym, the policy acquired greater robustness and exhibited stable walking behavior in MuJoCo. These results suggest that DR acts as an effective regularizer to reduce overfitting to the source simulator and enhance generalization to unseen environments.

In contrast, DR had limited benefits for the humanoid H1_2 robot. We hypothesize that this may be due to the relatively low complexity of the walking task or the insensitivity of the humanoid model to the specific types of randomization applied. This observation underscores the need for *task-aware and morphology-specific randomization strategies*. For example, quadrupeds with more dynamic and underactuated behaviors benefit more from DR applied to dynamics parameters, whereas humanoids may require randomization over higher-level task constraints, terrain variation, or observation noise to achieve similar gains.

We also observed that DR introduces a trade-off between convergence speed and robustness. Policies trained with DR tend to converge more slowly and display greater reward fluctuations, due to increased environment variability. However, this variability improves generalization, especially in zero-shot transfer settings. This aligns with prior findings that robustness often comes at the cost of stable optimization.

Finally, while standard DR improves transferability, it may not be sufficient when the simulator gap is large. In such cases, more sophisticated techniques such as *Active Domain Randomization (ADR)* or representation-level adaptation may be required to further enhance policy robustness across simulators with divergent physical and numerical characteristics.

7 Conclusion

In this project, we investigated the role of **domain randomization** in improving **Sim2Sim transfer** of RL policies for legged locomotion. Using Isaac Gym for training and MuJoCo for evaluation, we explored how injecting variability in simulation can enhance policy robustness.

Our results demonstrate that domain randomization significantly enhances transfer performance for dynamic systems such as quadruped robots. The randomized training process enables policies to handle variations in dynamics and external disturbances, leading to successful zero-shot deployment in different simulators. However, the effect of DR is not uniform across all robot types and tasks, suggesting the importance of targeted DR design that considers the structure of the policy and complexity of the environment.

This study provides a systematic evaluation of Sim2Sim policy transfer and contributes insight into how simulator discrepancies affect learning and generalization. Our findings point to three future directions: (1) expanding the domain randomization space to include observation-level and temporal perturbations; (2) implementing adaptive training strategies such as ADR; and (3) quantifying the simulator mismatch through state and trajectory divergence analysis.

Ultimately, this work serves as a stepping stone toward reliable **Sim2Real** transfer, where simulation-trained policies can be confidently deployed in physical robotic systems.

8 Team Contributions

- **Chenhao Zhu:** Responsible for implementing domain randomization policies and combining them with PPO algorithms. Also responsible for supervising policy training in the Isaac Gym simulator.
- **Yizhao Hou:** Lead the design and execution of experiments, including zero-point transfer evaluations and ablation studies. Responsible for analyzing results and compiling results into reports and presentations.
- **Jiaqi Shao:** Manage the setup and configuration of target simulation environments on platforms MuJoCo. Lead the deployment and evaluation of trained strategies in target simulators.

Changes from Proposal Due to time limit, we decide move the active domain randomization research and application experiments into future work.

References

Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. 2022. Understanding Domain Randomization for Sim-to-real Transfer. arXiv:2110.03239 [cs.LG] https://arxiv.org/abs/2110.03239

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. 2021. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. arXiv:2108.10470 [cs.RO] https://arxiv.org/abs/2108.10470

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, and Liam Paull. 2019. Active Domain Randomization. arXiv:1904.04762 [cs.LG] https://arxiv.org/abs/1904.04762

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. 2022. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. arXiv:2109.11978 [cs.RO] https://arxiv.org/abs/2109.11978

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG] https://arxiv.org/abs/1707.06347

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5026–5033. https://doi.org/10.1109/IROS.2012.6386109