Automatic Domain Randomization for Robust Sim-to-Real Transfer in Locomotion Tasks: Implementation and Parameter Relevance Analysis

Marc'Antonio Lopez
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
s336362@studenti.polito.it

Luigi Marguglio
Department of Control and Computer Engineering
Politecnico di Torino
Torino, Italy
s332575@studenti.polito.it

***Abstract*—Transferring reinforcement learning policies from simulation to the real world remains a fundamental challenge due to the mismatch between simulated and physical dynamics, commonly known as the reality gap. In this work, we implement Automatic Domain Randomization (ADR) for the MuJoCo Hopper locomotion task and conduct a comprehensive two-part study. Part 1 evaluates ADR against baseline and Uniform Domain Randomization (UDR), finding that UDR achieves $+3.9\%$ positive transfer while ADR 10M achieves near-perfect stability (-0.4% gap). Part 2 presents a systematic ablation study analyzing the contribution of individual parameters (mass, damping, friction) to transfer performance. Our key finding is that friction randomization alone achieves a remarkable $+154.6\%$ transfer gap, outperforming all other configurations. Statistical analysis reveals that friction contributes $+68.7\%$ marginally (p=0.074), while mass shows a negative contribution (-15.7%). Importantly, we discover strong antagonistic interaction effects: combining mass with friction reduces its positive effect by 94.5%. These results suggest that selective parameter randomization based on relevance analysis outperforms uniform randomization of all parameters.**

***Index Terms*—domain randomization, sim-to-real transfer, reinforcement learning, locomotion, ablation study, parameter relevance**

I. Introduction

Training robots in simulation offers significant practical advantages: unlimited data collection, massively parallel environments, and complete elimination of hardware damage risk. However, policies trained purely in simulation frequently fail when deployed on real hardware. This phenomenon, known as the reality gap, arises from the inevitable discrepancy between the idealized physics of simulation and the complex dynamics of the real world.

Domain Randomization addresses this challenge by varying simulation parameters during training. The underlying principle is straightforward: if an agent learns to perform well across a distribution of simulated environments, it should generalize to the real world, which can be viewed as just another sample from a sufficiently broad distribution. The robot effectively learns behaviors that are invariant to parameter changes, making it robust to the unknown real-world dynamics.

Traditional Uniform Domain Randomization (UDR) samples parameters from fixed ranges defined a priori. While this approach has achieved notable successes, it suffers from a fundamental limitation: selecting appropriate ranges requires careful manual tuning. Ranges that are too narrow may not encompass the real-world parameters, while ranges that are too wide can generate physically implausible scenarios, leading to a phenomenon called learned helplessness where the agent gives up learning anything useful.

Automatic Domain Randomization (ADR), introduced by OpenAI for dexterous manipulation, elegantly solves this problem by adapting randomization ranges based on agent performance. When performance exceeds a high threshold, the environment becomes "too easy" and ranges expand to increase difficulty. Conversely, when performance drops below a low threshold, ranges contract to make the task more manageable.

Contributions. This paper makes two main contributions:

1) **Part 1 - ADR Evaluation:** Systematic comparison of ADR against baseline and UDR across multiple training durations.
2) **Part 2 - Parameter Relevance Analysis:** Ablation study determining which parameters (mass, damping, friction) contribute most to transfer performance, including interaction effect analysis.

II. Related Work

The sim-to-real transfer problem has been extensively studied in the robotics and reinforcement learning communities. Domain randomization was popularized by Tobin et al. [2] for visual tasks, demonstrating that policies trained on randomized synthetic images could transfer successfully to real cameras. The approach was subsequently extended to dynamics randomization by Peng et al. [3].

Muratore et al. [8] introduced SPOTA (Simulation-based Policy Optimization with Transferability Assessment), providing a principled stopping criterion for domain randomization training based on transferability estimation. Ramos et al. [9] proposed BayesSim, a Bayesian framework that computes posterior distributions over simulator parameters, enabling adaptive domain randomization that outperforms uniform priors.

For locomotion specifically, Tan et al. [4] achieved successful sim-to-real transfer for quadruped robots by randomizing friction coefficients and introducing latency randomization during training. Hwangbo et al. [10] demonstrated that neural network policies trained in simulation could enable ANYmal robots to perform agile and dynamic motor skills, with careful attention to contact dynamics. Lee et al. [11] extended this work, showing that proprioceptive controllers trained in simulation achieve zero-shot generalization to challenging natural terrains including mud, snow, and vegetation.

OpenAI's work on solving Rubik's cube with a robot hand [1] brought ADR to prominence by demonstrating that automatic difficulty adjustment could achieve unprecedented dexterity through an emergent curriculum. Mehta et al. [7] proposed Active Domain Randomization, which learns a parameter sampling strategy to prioritize informative environment variations rather than relying on uniform sampling.

Recent work by Gang et al. [6] investigated the impact of static friction on sim-to-real transfer, finding that friction parameters are often the most critical for locomotion tasks. Our ablation study provides empirical support for this finding.

III. Method

A. Environment and Task

Our experiments use the MuJoCo [12] Hopper environment, a standard benchmark for locomotion control. The agent observes an 11-dimensional state vector and outputs 3-dimensional continuous torques. We define a source environment with misspecified dynamics (1kg torso mass offset) and a target environment with correct dynamics.

B. Automatic Domain Randomization

Our ADR implementation maintains state $\mathcal{S} = \{\delta_m, \delta_d, \delta_f\}$ representing randomization ranges for mass, damping, and friction. Each $\delta \in [0, 1]$ specifies the fractional variation. The update rule: if mean reward $\bar{R} \geq R_{high} = 1200$, ranges increase by $\epsilon = 0.05$; if $\bar{R} < R_{low} = 600$, ranges decrease; otherwise unchanged.

C. Ablation Study Design

To analyze parameter relevance, we trained 10 configurations testing all combinations of mass (M), damping (D), and friction (F) randomization:

All configurations used 2.5M timesteps and seed 42 for reproducibility.

Note: Part 1 and Part 2 are *independent* experiments. Part 1 compares ADR at different durations (2.5M, 5M, 10M) against separately trained baseline/UDR models. Part 2 focuses on parameter ablation with a unified training framework, including its own baseline and UDR configurations for fair comparison within the ablation context.

TABLE I
ABLATION STUDY CONFIGURATIONS

Config	Mass	Damping	Friction
baseline	×	×	×
adr_none	×	×	×
adr_mass	✓	×	×
adr_damp	×	✓	×
adr_fric	×	×	✓
adr_mass_damp	✓	✓	×
adr_mass_fric	✓	×	✓
adr_damp_fric	×	✓	✓
adr_all	✓	✓	✓
udr	✓	✓	✓

IV. Part 1: ADR vs Baseline vs UDR

A. Training Dynamics

Figure 1 shows the ADR range expansion over training. Both the 2.5M and 5M runs achieved a final range of $\pm 60\%$, while the 10M run plateaued at $\pm 40\%$. This suggests that longer training does not necessarily lead to higher ADR ranges.

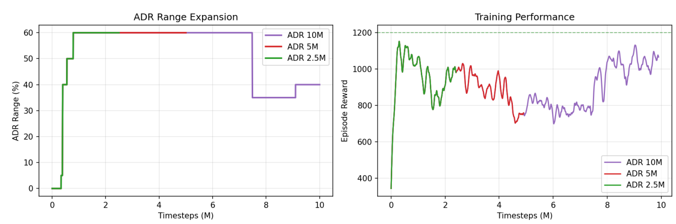

Fig. 1. ADR range expansion over training for different training durations.

B. Comparative Results

Table II presents the main evaluation results from Part 1.

TABLE II
PART 1: COMPARATIVE RESULTS (50 EVALUATION EPISODES, SEED=42)

Method	Source	Target	Gap
Baseline	1778 ± 65	1169 ± 95	-34.2%
UDR	1660 ± 10	$\mathbf{1725 \pm 34}$	$\mathbf{+3.9\%}$
ADR 2.5M	1567 ± 7	1533 ± 133	-2.1%
ADR 5M	1013 ± 224	781 ± 139	-22.9%
ADR 10M	1462 ± 39	1457 ± 145	-0.4%

Key findings:

- **Baseline** exhibits severe reality gap (-34.2%)
- **UDR** achieves best transfer ($+3.9\%$) with low variance
- **ADR 10M** achieves near-perfect stability (-0.4%)
- **ADR 5M** shows poor performance despite high range

V. Part 2: Ablation Study and Parameter Relevance

A. Transfer Performance by Configuration

Table III presents results from all 10 ablation configurations.

TABLE III
PART 2: ABLATION STUDY RESULTS (2.5M TIMESTEPS, SEED=42)

Config	Source	Target	Gap
adr_fric	642 ± 98	$\mathbf{1634 \pm 2}$	$\mathbf{+154.6\%}$
adr_all	1088 ± 115	1241 ± 258	$+14.1\%$
adr_damp	1631 ± 301	1761 ± 31	$+8.0\%$
adr_damp_fric	1530 ± 8	1638 ± 36	$+7.1\%$
udr	1724 ± 10	1711 ± 104	-0.8%
adr_mass_fric	1631 ± 3	1558 ± 85	-4.5%
adr_mass	1180 ± 181	973 ± 93	-17.5%
adr_mass_damp	949 ± 52	648 ± 29	-31.7%
adr_none	875 ± 208	303 ± 222	-65.4%
baseline	933 ± 213	314 ± 224	-66.4%

The most striking result is that `adr_fric` (friction-only randomization) achieves $+154.6\%$ transfer gap, vastly outperforming all other configurations including the full randomization approaches.

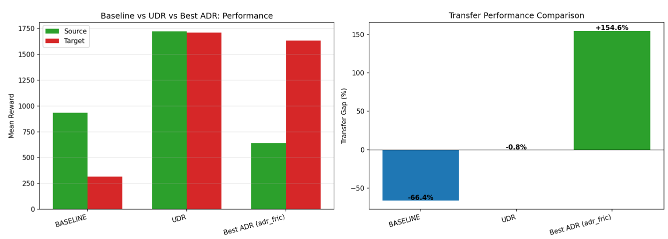

Fig. 2. Direct comparison of Baseline, UDR, and Best ADR configuration. Left: absolute performance on source and target. Right: transfer gap comparison.

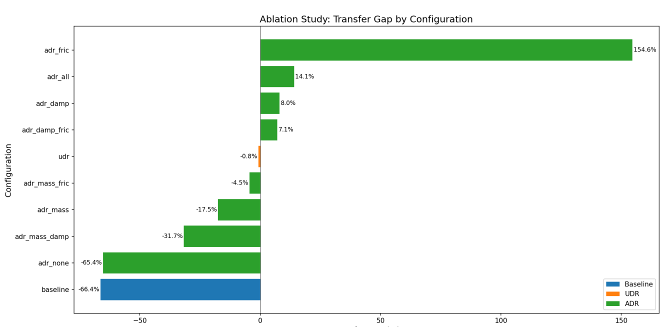

Fig. 3. Transfer gap by configuration. Positive values (green) indicate successful transfer; negative values (red) indicate reality gap.

B. Statistical Analysis of Parameter Contributions

We computed the marginal contribution of each parameter using a factorial analysis approach. For each parameter P, we measured:

$$\text{Contribution}(P) = \bar{G}_{with\_P} - \bar{G}_{without\_P} \tag{1}$$

where $\bar{G}$ is the mean transfer gap across relevant configurations.

Key Finding: Friction is the only parameter with a positive marginal contribution, and it is marginally significant at

TABLE IV
PARAMETER MARGINAL CONTRIBUTIONS

Rank	Param	Contrib.	p-value	Sig.
1	FRICTION	$+68.70\%$	0.0744	*
2	MASS	-15.68%	0.7120	–
3	DAMPING	-0.86%	0.9840	–

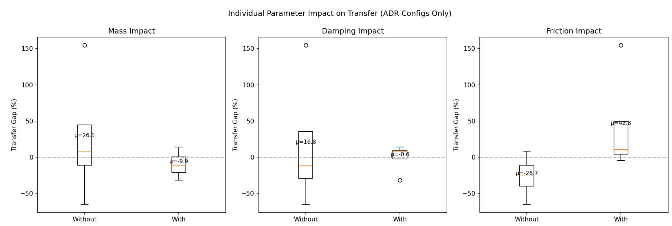

Fig. 4. Box plots showing the impact of each parameter on transfer gap. Configurations with friction enabled show substantially higher transfer performance.

$p < 0.10$. Mass shows a *negative* contribution, meaning adding mass randomization tends to hurt transfer performance.

C. Interaction Effects

We analyzed whether parameter combinations produce synergistic or antagonistic effects beyond their individual contributions.

TABLE V
PARAMETER INTERACTION EFFECTS

Interaction	Expected	Actual	Effect
M × D	-11.1%	-6.1%	$+5.0\%$ (syn.)
M × F	$+97.5\%$	$+2.9\%$	$\mathbf{-94.5}\%$ (ant.)
D × F	$+113.0\%$	$+6.8\%$	$\mathbf{-106.2}\%$ (ant.)

Critical Discovery: Strong antagonistic interactions exist. Adding mass to friction reduces friction's positive effect by 94.5%. Similarly, adding damping to friction reduces its effect by 106.2%. This explains why `adr_fric` (friction only) dramatically outperforms `adr_all` (all parameters).

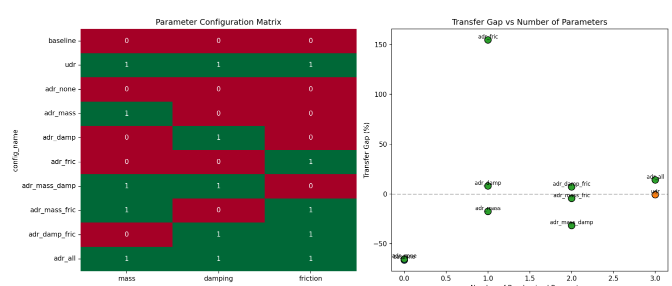

Fig. 5. Left: Parameter configuration matrix. Right: Transfer gap vs number of randomized parameters. Note the lack of monotonic relationship.

VI. Discussion

Our two-part study reveals several important insights:

1. Friction Dominates Transfer Performance. The exceptional performance of friction-only ADR ($+154.6\%$) can be attributed to:

- Ground contact is critical for locomotion, as demonstrated by Hwangbo et al. [10] and Lee et al. [11]
- Friction is often the most misspecified parameter in simulation
- Single-parameter ADR allows focused robustness learning

2. Mass Randomization Can Hurt. Counter-intuitively, mass randomization shows negative contribution (-15.7%). This may be because:

- The source environment already has a 1kg torso mass offset
- Additional mass randomization interferes with this built-in mismatch
- Over-robustification leads to conservative behaviors, consistent with findings by Ramos et al. [9] on parameter selection

3. More Parameters $\neq$ Better Transfer. Strong antagonistic interactions mean that randomizing all parameters can be worse than selective randomization. This challenges the common assumption that broader randomization is always beneficial.

4. ADR Range $\neq$ Transfer Quality. ADR 5M achieved $\pm 60\%$ range but -22.9% gap, while ADR 10M achieved only $\pm 40\%$ range but -0.4% gap. The ablation study explains this: what matters is *which* parameters are randomized, not just how much.

A. Practical Recommendations

Based on our empirical evidence:

TABLE VI
RECOMMENDED CONFIGURATIONS BY USE CASE

Goal	**Recommendation**
Maximum transfer	`adr_fric` (friction only)
Balanced robustness	`adr_damp_fric`
Conservative approach	`udr` (fixed $\pm 30\%$)
Avoid	`adr_mass`, `adr_mass_damp`

VII. Conclusion

We presented a comprehensive study of domain randomization for locomotion, encompassing both method comparison (Part 1) and parameter relevance analysis (Part 2). Our key findings:

1) **Friction is king:** Friction-only ADR achieves $+154.6\%$ transfer gap, outperforming all other configurations.
2) **Mass hurts:** Mass randomization contributes -15.7% to transfer, likely due to interference with built-in dynamics mismatch.
3) **Interactions matter:** Strong antagonistic effects reduce combined parameter benefits by up to 106%.
4) **Selective randomization wins:** Data-driven parameter selection outperforms uniform randomization of all parameters.

These results have practical implications: practitioners should conduct ablation studies to identify relevant parameters rather than assuming that more randomization is better.

Acknowledgment

This work was conducted as part of the Robot Learning course at Politecnico di Torino, under the supervision of the VANDAL laboratory.

References

[1] OpenAI et al., “Solving Rubik’s Cube with a Robot Hand,” arXiv:1910.07113, 2019.
[2] J. Tobin et al., “Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World,” IROS, 2017.
[3] X. B. Peng et al., “Sim-to-Real Robot Learning from Pixels with Progressive Nets,” CoRL, 2018.
[4] J. Tan et al., “Sim-to-Real: Learning Agile Locomotion For Quadruped Robots,” RSS, 2018.
[5] J. Schulman et al., “Proximal Policy Optimization Algorithms,” arXiv:1707.06347, 2017.
[6] S. Gang et al., “Impact of Static Friction on Sim2Real in Robotic Reinforcement Learning,” 2025.
[7] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, “Active Domain Randomization,” Proc. Conference on Robot Learning (CoRL), PMLR, 2020.
[8] F. Muratore, F. Treede, M. Gienger, and J. Peters, “Domain Randomization for Simulation-Based Policy Optimization with Transferability Assessment,” Proc. Conference on Robot Learning (CoRL), PMLR 87:700-713, 2018.
[9] F. Ramos, R. C. Possas, and D. Fox, “BayesSim: Adaptive Domain Randomization via Probabilistic Inference for Robotics Simulators,” Robotics: Science and Systems (RSS), 2019.
[10] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, “Learning Agile and Dynamic Motor Skills for Legged Robots,” Science Robotics, vol. 4, no. 26, 2019.
[11] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning Quadrupedal Locomotion over Challenging Terrain,” Science Robotics, vol. 5, no. 47, 2020.
[12] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A Physics Engine for Model-Based Control,” IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), 2012.