A Systematic Ablation Study of Sensor and Actuator Robustness Methods for Quadruped Locomotion

Anand Patel
School of Computer Science and Applied Mathematics
University of the Witwatersrand, Johannesburg
Email: anand.patel@students.wits.ac.za

***Abstract*—Combining multiple robustness methods targeting different failure modes is widely assumed to provide additive benefits, yet this assumption remains empirically untested. We systematically evaluate whether Smooth Regularized RL and Domain Randomization, targeting sensor noise and actuator failures respectively, cooperate or interfere when combined for quadruped locomotion. Across 38,000 evaluation episodes, we find that combining these methods produces significant interference rather than synergy: the combined approach underperforms specialized domain randomization by 56-145% under stress conditions due to conflicting gradient signals during training. Surprisingly, in our experiments, domain randomization achieves best baseline performance through regularization effects, contradicting the assumption that robustness training sacrifices nominal performance. Our results challenge the multi-objective robustness paradigm and demonstrate that specialized training for the hardest failure mode provides superior performance across all conditions.**

***Index Terms*—Quadruped Locomotion, Robustness, Domain Randomization, Smooth Regularization, Reinforcement Learning, PPO, Ablation Study, Negative Synergy**

I. Introduction

Real-world robotic systems fail in predictable ways: sensors degrade and actuators malfunction. Reinforcement learning policies trained under ideal simulation conditions catastrophically fail when deployed on physical hardware experiencing sensor noise and actuator failures [1]–[5]. Two training paradigms address these failure modes: Smooth Regularized RL (SR2L) creates robustness to sensor noise through smoothness constraints [6], while Domain Randomization (DR) creates robustness to actuator failures through randomized exposure [4], [5].

The natural hypothesis is that complementary benefits are achieved, as these methods target distinct failure modes (sensor degradation vs. actuator malfunction); combining them should provide additive robustness. However, an alternative hypothesis is interference: SR2L demands policy insensitivity to observation changes while DR demands strong sensitivity to detect failures, creating conflicting training objectives. Prior multi-randomization work shows promise [7], yet no systematic study quantifies whether combining SR2L and DR produces synergy or interference. This work provides that empirical analysis.

We train four quadruped locomotion policies to test these hypotheses systematically: **M1** (baseline with PPO only and no robustness training), **M2** (SR2L only), **M3** (DR only), and **M4** (SR2L and DR combination). Each model undergoes identical evaluation across sensor noise tests, actuator failure tests, and combined stress conditions.

This ablation design directly answers whether combining methods provides complementary benefits or creates interference. Our evaluation of 38,000 episodes provides quantitative evidence in support of the interference hypothesis: specialized training consistently outperforms multi-objective training across all conditions.

Contributions. This work makes three key contributions:

1. Negative transfer in multi-objective robustness. We provide quantitative evidence that combining robustness methods targeting different failure modes creates interference rather than synergy, with the combined approach (M4) underperforming specialized training (M3) by 56-145% under stress conditions.

2. Observation normalization as critical enabler. Through controlled ablation, we identify that observation normalization provides a substantial robustness baseline (retaining 83% of performance under sensor noise), making specialized robustness training less critical than previously assumed.

3. Cross-distribution generalization. We demonstrate that sensor noise robustness training creates cross-distribution generalization, with policies maintaining high performance on noise types never encountered during training (Poisson, salt-and-pepper).

Our findings challenge the assumption that combining complementary robustness methods produces additive benefits, and provide actionable deployment guidance: train specialized DR for actuator failures, which also provides cross-robustness to sensor noise as a side benefit.

II. Background

Reinforcement learning addresses sequential decision-making under uncertainty, formalized as a Markov Decision Process (MDP) defined by tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$: state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $\mathcal{P}(s'|s,a)$, reward function $\mathcal{R}(s,a)$, and discount factor γ. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps observations to actions. The goal is to learn an optimal policy π^* that maximizes expected cumulative discounted reward: $J(\pi) = \mathbb{E}_{\tau\sim\pi}[\sum_{t=0}^{\infty}\gamma^t r_t]$. For quadruped locomotion, states are proprioceptive observations (joint positions, velocities, torso orientation), actions are motor commands, and rewards encourage forward motion.

A. Proximal Policy Optimization

We use Proximal Policy Optimization (PPO) [8], a policy gradient method that balances sample efficiency with training stability. PPO updates the policy by maximizing a clipped surrogate objective:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \quad (1)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the probability ratio between new and old policies, $\hat{A}_t$ is the advantage estimate, and ϵ is the clipping parameter (typically 0.2). This clipping constrains policy changes per update, preventing destructive large updates common in vanilla policy gradient methods. PPO has become the standard for continuous control tasks including legged locomotion [1], [9], providing reliable convergence with moderate hyperparameter sensitivity.

B. Robustness Methods for Reinforcement Learning

Policies trained in ideal simulation conditions often fail when deployed on physical hardware due to observation noise and environmental variations. Two primary approaches address this sim-to-real gap: *observation robustness* methods like SR2L that train policies to tolerate sensor noise through smoothness regularization, and *environmental robustness* methods like domain randomization that expose policies to varied conditions during training.

SR2L augments the standard RL objective with a smoothness penalty:

$$L_{total} = L_{PPO} + \lambda \cdot \mathbb{E}_{s\sim\mathcal{D}}[\|\pi_\theta(s) - \pi_\theta(s+\delta)\|^2] \quad (2)$$

where $\delta \sim \mathcal{N}(0, \sigma^2 I)$ is Gaussian noise applied to observations, and λ controls the regularization strength. This penalizes policy sensitivity to observation perturbations.

Domain randomization trains on randomly varied environments without modifying the loss function; instead, it varies environment parameters (e.g., joint failures, masses, friction) across episodes. While both target robustness, they operate through different mechanisms: SR2L constrains policy smoothness through explicit regularization, while DR achieves robustness through diverse training exposure. Whether these complementary approaches cooperate or interfere when combined remains an open question that this work addresses.

C. Curriculum Learning in Reinforcement Learning

Curriculum learning [10] is a training strategy inspired by human education, where learning progresses from simple to complex tasks. Rather than training on the full task difficulty from the start, curriculum learning organizes training into stages of progressively increasing difficulty, allowing the agent to master fundamental skills before attempting more challenging variations.

In reinforcement learning, curriculum learning prevents early training failure on tasks where the full difficulty would overwhelm the untrained policy. For robustness training specifically, curriculum learning is particularly valuable when the target robustness conditions (e.g., high sensor noise or multiple simultaneous joint failures) would cause complete locomotion failure if applied from the beginning. By gradually introducing these challenges, the policy first learns stable baseline locomotion, then incrementally adapts to handle perturbations.

For domain randomization targeting actuator failures, curriculum learning typically involves phased training: early phases use clean environments or minimal randomization to establish basic locomotion, middle phases introduce single failure modes, and later phases increase failure frequency or combine multiple failures [11]. This structured approach is a valuable method of learning as it significantly improves both training success rates and final policy robustness compared to exposing agents to full randomization from initialization.

III. Methodology

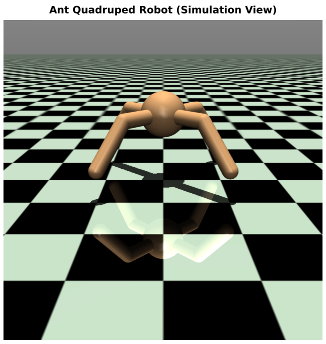
A. Experimental Platform and Environment

All experiments use the RealAnt-v0 quadruped environment, a custom robot extending Gymnasium's Ant-v4 [12] with latency modeling, noise injection, and joint failure support, simulated in MuJoCo physics engine [13]. The RealAnt has 8 actuated degrees of freedom (2 joints per leg: hip and ankle), organized as shown in Figure 1. The 29-dimensional observation space consists of: (1) torso linear velocities in x, y, z (3D), (2) torso z-position (1D), (3) $\sin$ and $\cos$ of torso Euler angles for roll, pitch, yaw (6D), (4) torso angular velocities (3D), (5) joint angular positions (8D), and (6) joint angular velocities (8D). The action space consists of 8 continuous angular position set-points for the joints.

These observations have vastly different numerical ranges: joint velocities span approximately $\pm$10 rad/s, while torso positions span approximately $\pm$0.05m, creating a 200:1 magnitude ratio. To prevent gradient domination by high-magnitude features, we apply observation normalization: maintaining online mean and standard deviation statistics during training, then normalizing each dimension to zero mean and unit variance before passing observations to the policy network.

The robot's four legs are numbered by position: Joint 1 (front-left), Joint 2 (front-right, camera-facing), Joint 3 (rear-left), and Joint 4 (rear-right, camera-facing). Each leg contains two joints: a proximal hip joint closer to the torso and a distal ankle joint at the leg's end. References to specific joints throughout the paper use this numbering (e.g., "hip 1" refers to the front-left hip, "ankle 4" refers to the rear-right ankle). This anatomical layout is critical for interpreting joint failure results in Section IV-C, particularly the worst-case performance of camera-facing ankle joints.

The locomotion task requires the robot to walk forward as quickly as possible while maintaining stability. The reward



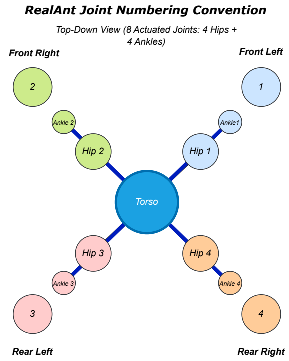


Fig. 1. **RealAnt quadruped robot.** Left: The robot simulated in MuJoCo physics engine. Right: Joint numbering schematic showing 8 actuated joints organized as 4 legs with 2 joints each: proximal hip joints (closer to torso) and distal ankle joints (end of leg). Joints are numbered 1-4 corresponding to front-left, front-right, rear-left, and rear-right legs respectively. Joints 2 and 4 are positioned on the right side when the robot walks left→right.

function is $r_t = 100 \cdot v_x^2$, where v_x is forward velocity, following velocity-based reward shaping [14]. Episodes terminate after 1000 timesteps or if the robot falls (torso height < 0.2m).

B. Model Training Configurations

We train four distinct models using Proximal Policy Optimization (PPO) [8] to enable systematic ablation analysis:

M1 (Baseline): Standard PPO trained in clean conditions with no robustness augmentation. Uses SuccessRewardWrapper only. Trained for 32M timesteps on RealAnt-v0. This establishes the performance ceiling under ideal conditions.

M2 (SR2L): PPO with Smooth Regularized Reinforcement Learning [6]. Augments the standard PPO loss with a smoothness regularization term:

$$L_{total} = L_{PPO} + \lambda \cdot \mathbb{E}_{s \sim \mathcal{D}}[\|\pi_\theta(s) - \pi_\theta(s + \delta)\|^2] \quad (3)$$

where $\delta \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.01$ applied only to joint observation dimensions (13-28), and $\lambda = 0.001$ (hyperparameter selection detailed below). Perturbations applied during training but not to the actual environment. Trained for 32M timesteps. While training uses Gaussian noise only, we evaluate generalization to Poisson and salt-and-pepper noise distributions to assess whether smoothness regularization transfers across noise types.

M3 (Domain Randomization): PPO with 3-phase curriculum domain randomization targeting joint failures. Training schedule:

- Phase 1 (0-10M steps): Clean training, no failures
- Phase 2 (10-20M steps): 50% episodes with exactly 1 random joint locked
- Phase 3 (20-32M steps): 60% episodes with 1-2 random joints locked

Joint locking implemented by setting affected joint torques to zero. Total training: 32M timesteps.

M4 (Combined): PPO with both SR2L smoothness regularization AND curriculum domain randomization. Identical curriculum schedule to M3, but with SR2L loss ($\lambda = 0.001$, $\sigma = 0.01$) applied throughout all phases. The combined training objective is:

$$L_{total} = L_{PPO}(s,a,r) + \lambda_{SR2L} \cdot \mathbb{E}[\|\pi_\theta(s) - \pi_\theta(s+\delta)\|^2] \quad (4)$$

where L_{PPO} is PPO's clipped surrogate loss and DR modifies the environment by randomizing joint failures during rollout collection. Total training: 32M timesteps.

C. SR2L Hyperparameter Selection

SR2L requires three key design choices: perturbation magnitude σ, regularization weight λ, and which observation dimensions to perturb. We determined these parameters empirically, following the general framework from the original SR2L work [6] and adapting it to the quadruped locomotion domain.

Perturbation magnitude ($\sigma = 0.01$): We selected $\sigma = 0.01$ through empirical tuning to represent realistic sensor noise levels while maintaining training stability. This magnitude is large enough to expose the policy to meaningful measurement errors but small enough to avoid overwhelming the learning signal during early training phases. The value balances between insufficient robustness training (at lower noise levels) and training instability (at higher noise levels).

Regularization weight ($\lambda = 0.001$): We use $\lambda = 0.001$ following the original SR2L paper's experimental setup [6] for continuous control tasks. This weight balances robustness and performance: values too large ($\lambda > 0.01$) cause excessive policy smoothing that degrades nominal locomotion, while values too small ($\lambda < 0.0001$) provide insufficient robustness incentive. Our experimental results (Section IV) confirm this weight provides measurable robustness benefits while maintaining competitive baseline performance.

Perturbation dimensions (13-28): We determined through preliminary experiments that perturbing only joint-specific measurements—positions (dimensions 21-28) and velocities (dimensions 13-20)—provided better training stability than full observation perturbation. This selective approach targets proprioceptive sensor noise in joint encoders, the primary sensor failure mode in quadruped systems. We exclude torso state dimensions (positions 0-12) because perturbing global pose measurements destabilized early training before the policy learned basic locomotion. This dimension selection focuses the smoothness regularization on joint sensors while maintaining stable learning dynamics.

D. PPO Training Configuration

All models use identical training configuration to ensure fair comparison. The policy architecture is a 2-layer multilayer perceptron (MLP) with [128, 128] hidden units and $\tanh$ activation, following standard practice in quadruped RL [1]. This moderate architecture size balances model expressiveness with sample efficiency. $\tanh$ activation is standard for continuous control as it naturally bounds policy outputs.

PPO hyperparameters were selected following established best practices [15] with minor empirical adjustments for the quadruped locomotion domain: learning rate 3×10^{-4} (Stable Baselines3 default [16]), batch size 1536, 8 epochs per update, clip range 0.15, GAE $\lambda = 0.97$, discount $\gamma = 0.995$. The clip range of 0.15 (more conservative than the default 0.2) prevents excessively large policy updates, improving training stability. We use 8 epochs per update (higher than the typical 4-10 range) to improve sample efficiency during the extended 32M-step training runs. The high discount factor ($\gamma = 0.995$) and GAE parameter ($\lambda = 0.97$) are appropriate for the long-horizon task (1000 timesteps per episode), enabling effective credit assignment over extended trajectories.

E. Observation Normalization

All models use running observation normalization to handle the vastly different numerical ranges in the observation space. During training, we maintain running statistics (mean μ and standard deviation σ) for each observation dimension, updating these with each environment step. Observations are normalized using $\hat{o}_i = (o_i - \mu_i)/\sigma_i$ before being passed to the policy network. During evaluation, statistics are frozen at their final training values to ensure consistent behavior. This normalization prevents gradient domination by high-magnitude features and is critical for learning in continuous control tasks [16], [17].

F. Evaluation Protocol

We evaluate all four models through systematic experiments testing baseline performance, sensor noise robustness, joint failure adaptation, and combined stress scenarios. Each experiment uses consistent evaluation protocols across models to enable direct comparison.

G. Performance Metrics

For each evaluation episode, we record:

- **Distance traveled**: Total forward distance over 1000 timesteps
- **Average velocity**: Mean forward velocity $\bar{v}_x$
- **Success rate**: Percentage of episodes achieving > 2m distance
- **Cumulative reward**: Total reward accumulated
- **Fall rate**: Percentage of episodes ending in termination
- **Retention percentage**: (Faulty performance / Baseline performance) $\times$ 100%
- **Recovery time**: Time to regain 50% baseline velocity after fault injection

IV. Experiments and Results

We present systematic experimental evaluation of all four models. Each experiment is described immediately before its results for clarity. All statistical tests use paired t-tests with Bonferroni correction ($\alpha = 0.05/n$) where n is the number of comparisons. Retention percentage is defined as (faulty performance / baseline performance) $\times$ 100%.

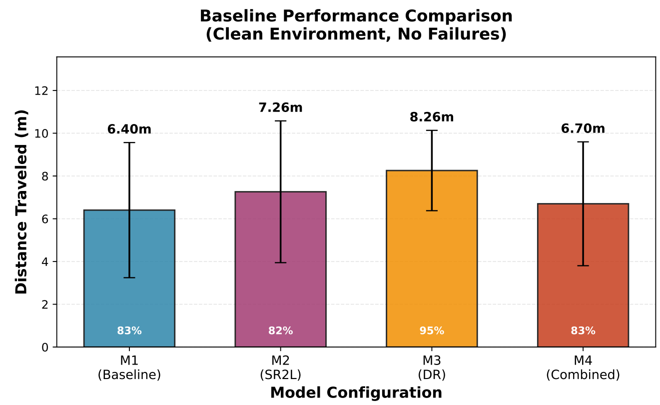


Fig. 2. **Baseline performance comparison.** M3 (DR) achieves best performance (8.26m), significantly outperforming M1 (6.40m, $p < 0.001$), M2 (7.26m, $p = 0.005$), M4 (6.70m, $p < 0.001$). Error bars: 95% CI. Bonferroni corrected, $n = 100$ episodes.

A. Experiment 1: Baseline Performance and Training Dynamics

Protocol. Each model was evaluated for 100 episodes in clean conditions (no sensor noise or faulty joints) to establish baseline performance.

Baseline results (Figure 2): M3 (DR) achieved best performance at 8.26m, significantly outperforming M1 (6.40m, $p < 0.001$), M2 (7.26m, $p = 0.005$), and M4 (6.70m, $p < 0.001$). Domain randomization acts as effective regularization, improving locomotion quality even in clean test conditions: M3 outperforms M1 by 29% despite both being tested on failure-free environments. M4 underperforms M3 despite identical DR training, showing early signs of SR2L+DR interference.

Training dynamics (Figure 3): Multi-seed training curves (n=5 seeds) reveal the importance of proactive evaluation. M3 achieves highest training rewards and best evaluation (8.26m), demonstrating effective learning. M1 shows moderate training rewards but worst evaluation (6.40m), demonstrating that high training rewards do not guarantee high-quality locomotion without regularization. M4 exhibited catastrophic collapse at 30M steps (rewards $\rightarrow$ 0) when SR2L smoothness and DR failures were simultaneously active, providing direct evidence of gradient conflict.

Key finding: For this locomotion task, domain randomization acts as regularization, providing baseline improvement over unaugmented training (M3: 8.26m vs M1: 6.40m). M3 achieves both highest training rewards and best evaluation performance, while M1's moderate training rewards do not translate to high-quality locomotion—demonstrating that DR regularization improves policy quality beyond what standard PPO achieves.

B. Experiment 2: Sensor Noise Robustness

Protocol. Models evaluated across 12 escalating Gaussian noise levels (0$\times$ to 30$\times$ M2's training noise, 100 episodes per level). Observation normalization ablation conducted by training M1/M2/M3/M4 without normalization.

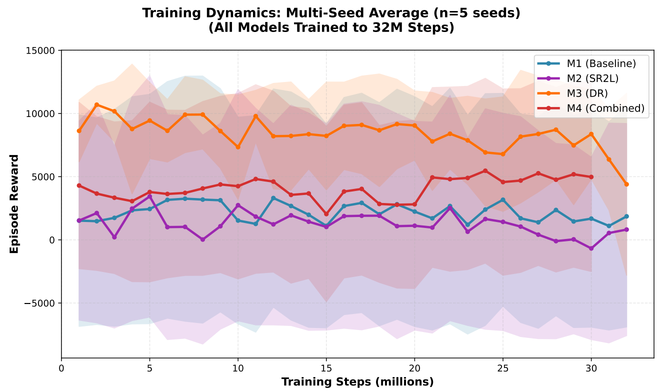


Fig. 3. **Training dynamics show training-evaluation disconnect (multi-seed average, n=5 seeds).** M1: worst evaluation (6.40m, lacks regularization). M3: stable training performance, best evaluation (8.26m, regularization effect). M4: 30M steps (SR2L+DR gradient conflict). Error bands show ±1 SD across seeds. Checkpoints evaluated every 1M steps, 10 episodes per checkpoint.

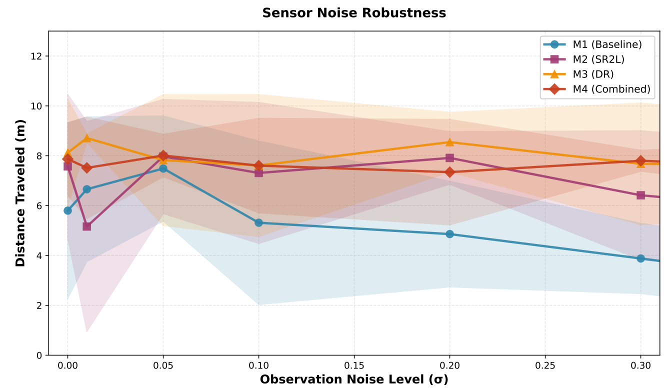


Fig. 4. **Sensor noise robustness.** All models maintain >90% retention at 10× training noise ($\sigma = 0.10$). M2 shows 100.7% retention. Shaded regions represent ±1 standard deviation across 100 episodes per noise level, reflecting stochastic variation in environment dynamics and policy execution. $n = 100$ episodes per noise level.

Results (Figure 4): Although SR2L introduces theoretical sensor noise robustness, our evaluation reveals its effect is largely redundant once observation normalization is applied. All robustness-trained models maintain >90% performance at 10× training noise (σ=0.10), with M2 showing slight improvement (100.7% retention, consistent with stochastic resonance). M3 maintains 92% retention despite never training with noise, indicating regularization-induced robustness from joint failure adaptation. M1 shows 83% retention, revealing that normalization alone provides substantial robustness.

Normalization prerequisite: Without observation normalization, all models exhibited >99% performance collapse (M1/M2/M3/M4: <0.1m), confirming normalization is a fundamental prerequisite—not an implicit robustness mechanism—for continuous control with high-dimensional observations spanning vastly different numerical ranges [17].

Key finding: Observation normalization alone provides substantial sensor noise robustness (M1: 83% retention). SR2L training adds only marginal improvement (+17.7%), while M3 achieves 92% retention through regularization-induced robustness despite never training with noise. Implication: specialized noise training may be unnecessary for robustness from Gaussian Noise.

C. Experiment 3: Joint Failure Robustness

Protocol. Each of 8 joints locked individually for entire episodes (100 episodes per joint), simulating catastrophic single-joint failures.

Results (Figure 5): M3 achieved highest average distance under joint failures (3.80m), significantly outperforming M4 (3.39m, $p < 0.01$), M2 (2.23m, $p < 0.001$), and M1 (1.71m, $p < 0.001$). Joint failure location significantly impacts recovery difficulty: ankle joint failures cause greater performance degradation than hip joint failures across all models.

Key finding: M3 (DR) dominates joint failure robustness with 3.80m average performance, 2.2× better than baseline M1. Joint failure location matters: ankle failures are consistently harder to recover from than hip failures across all models.

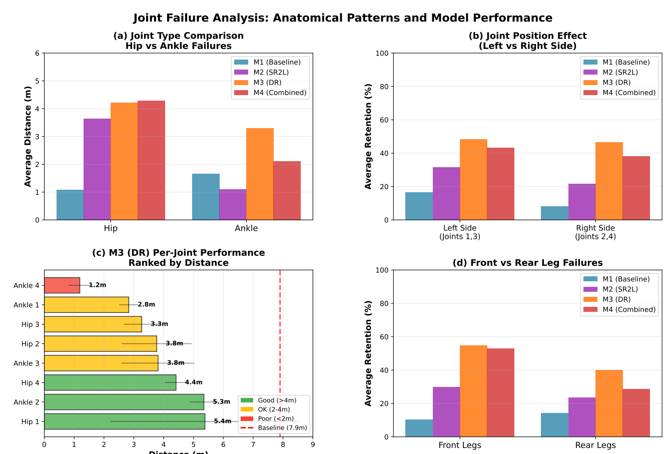


Fig. 5. **Joint failure robustness.** M3 (DR) achieves best average performance (3.80m), outperforming M4 (3.39m), M2 (2.23m), and M1 (1.71m). M3 shows consistent advantage across all 8 individual joint failures. Ankle failures cause greater performance degradation than hip failures. Ankle 4 is the universal worst-case across all models. $n = 100$ episodes per joint.

D. Experiment 4: Robustness Method Comparison

Protocol. We evaluate 6 combined stress scenarios crossing joint failures with sensor noise (100 episodes per scenario). The most severe condition combines dual cross-body joint failures (hip 4 + ankle 3) with high sensor noise (σ=0.10, representing 10× M2's training noise level).

Results (Figure 6): M3 consistently outperformed M4 across all scenarios. Under the most severe combined stress condition (dual joint failure + σ=0.10), M3 achieved 2.38m versus M4's 0.97m (145% advantage, $p < 0.001$). Under dual failure without noise, M3 achieved 4.40m versus M4's 2.81m (57% advantage). This confirms specialized DR training outperforms combined SR2L+DR training even under conditions requiring both noise tolerance and failure adaptation.

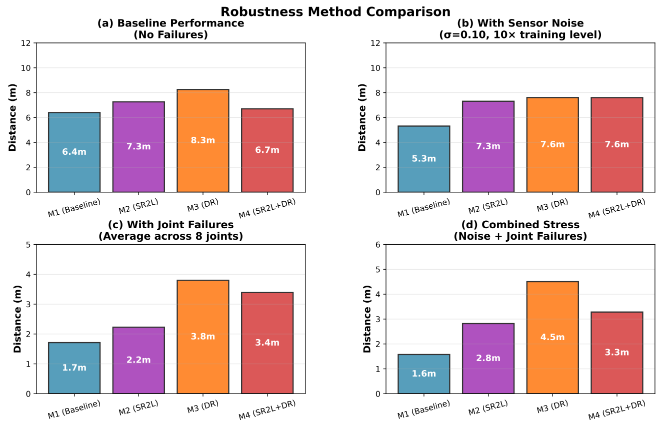


Fig. 6. **Robustness method comparison.** M3 (DR) outperforms M4 (Combined) across all combined stress scenarios. Maximum combined stress (dual joint failure + σ=0.10): M3 2.38m vs M4 0.97m (145% advantage, $p < 0.001$). M2 shows specialization—excellent noise retention but poor failure adaptation. M4 shows no robustness advantage over M3, showing further evidence of training interference. $n = 100$ episodes per scenario.

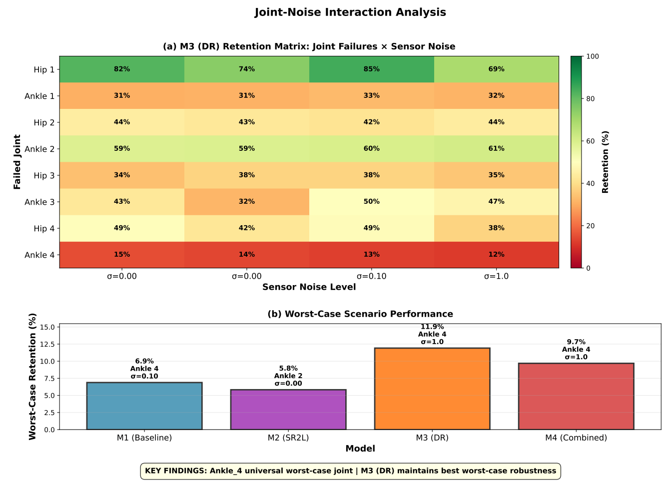


Fig. 7. **Joint-noise interaction.** Retention heatmaps across 8 joints $\times$ 4 noise levels. Ankle_4 universal worst-case: M3 best (13.7%) vs M1 (7.7%), M2 (6.5%), M4 (7.8%) at highest noise. Degradation approximates product of individual retention values (additivity not synergy), confirming distinct failure subsystems. $n = 100$ episodes per condition, 3,200 total.

M2 (SR2L) maintained high noise retention but showed poor joint failure adaptation (2.24m average), demonstrating clear specialization.

Key finding: No synergy from combining robustness methods—M4 (SR2L+DR) consistently worse than M3 (DR alone) by 57-145% across all combined stress scenarios. Specialized training outperforms multi-objective training.

E. Experiment 5: Joint-Noise Interaction Effects

Protocol. We conduct systematic ablation of joint failure $\times$ sensor noise interactions using a full factorial design. We cross all 8 individual joint failures with 4 sensor noise levels ($\sigma \in \{0.00, 0.05, 0.10, 1.0\}$), creating 32 distinct conditions per model. This tests whether joint failures and sensor noise combine additively or synergistically. Each model is evaluated for 100 episodes per condition.

Results (Figure 7): Ankle 4 emerged as universal worst-case joint under extreme combined stress (highest noise $\sigma = 1.0$), with M3 maintaining best worst-case retention (13.7%) compared to M1 (7.7%), M2 (6.5%), and M4 (7.8%). Performance degradation from combined stressors approximated the product of individual retention values (additivity, not synergy), confirming that sensor noise and joint failures affect distinct locomotion subsystems with no synergistic interaction. Ankle joints showed greater noise sensitivity than hip joints across all models.

Key finding: Joint failures and sensor noise combine additively (not synergistically), confirming distinct failure subsystems with no interaction effects. Ankle 4 is universal worst-case across all models. M3 maintains best worst-case robustness.

V. Discussion

A. Mechanisms of Training Interference

Our results demonstrate that combining SR2L and DR produces interference rather than synergy (M3 outperforms M4 by 56-145% across combined stress scenarios, all $p < 0.001$). We propose that this stems from fundamental incompatibility between their training objectives, creating conflicting gradient signals during optimization.

As described in Section III, M4's combined loss includes both PPO's policy gradient objective and SR2L's smoothness regularization, while DR modifies the environment by randomizing joint failures during rollout collection. During gradient descent, these components generate conflicting signals:

$$\begin{aligned} \nabla_\theta L_{PPO} &: \text{Adapt policy to environment state} \\ \nabla_\theta L_{SR2L} &: \text{Minimize policy sensitivity} \end{aligned} \tag{5}$$

The Conflict. When a joint fails, the robot's observations change drastically (zero velocity/position for the failed joint). Compensating for this failure requires the policy to react strongly to these observation changes—adjusting weight distribution and gait based on which joints remain functional. However, SR2L explicitly penalizes such sensitivity: it minimizes $\|\pi_\theta(s) - \pi_\theta(s + \delta)\|^2$, forcing the policy to produce similar actions despite observation changes. DR demands "adapt aggressively to failures," while SR2L demands "ignore observation perturbations." These objectives fundamentally conflict.

Empirical Evidence. M4's catastrophic collapse at 30M steps (Figure 3)—where locomotion capability temporarily vanishes—occurs when both SR2L penalties and DR failures are simultaneously active. During this period, training rewards drop from 30K to -8.6K (mean: 5.5K $\pm$18.5K), with value loss collapsing from 0.01 to 0.0001 (100$\times$ decrease) while entropy loss remains stable at -19.5, indicating the policy becomes deterministic but ineffective. Neither M2 (SR2L alone) nor M3 (DR alone) exhibit such instability, confirming the collapse stems from their interaction. After partial recovery, M4 achieves 6.70m baseline and 2.81m under combined stress, underperforming M3 (8.26m, 4.40m) in both metrics.

Implications. This finding contradicts the common assumption that combining robustness methods yields additive benefits. Multi-objective training can produce negative transfer when objectives conflict at the gradient level [18]. Practitioners should analyze whether robustness methods target independent failure modes (potentially compatible) or require opposing policy characteristics (incompatible). For quadruped locomotion, discrete structural adaptation (DR) fundamentally opposes continuous smoothness (SR2L).

B. Domain Randomization as Regularization

M3 (domain randomization) achieved best clean-environment performance (8.26m), significantly outperforming unaugmented baseline M1 (6.40m, $p < 0.001$), contradicting the assumption that robustness training sacrifices baseline performance. Multi-seed training curves reveal M3 achieves highest training rewards and best evaluation, while M1 shows moderate training rewards but severe overfitting—domain randomization prevents overfitting through continuous exposure to varied conditions, enabling both effective learning and robust generalization.

Mechanism. Domain randomization acts as implicit regularization by preventing the policy from memorizing specific environment configurations. By training under randomly varied joint failure patterns, M3 is forced to learn generalizable locomotion strategies that work across diverse conditions. This regularization effect parallels data augmentation in supervised learning [19], where exposing models to varied training examples prevents overfitting to training distribution specifics. In our experiments, this regularization benefit (M3: 8.26m vs M1: 6.40m, 29% improvement) outweighs any performance cost from training under perturbed conditions.

Implications for Deployment. This finding fundamentally changes the cost-benefit analysis of robustness training. Rather than accepting baseline performance sacrifice for robustness, practitioners deploying domain randomization for fault tolerance receive a "free" baseline performance improvement through overfitting prevention. This makes M3-style DR training the recommended approach even for environments expected to have low actuator failure rates, as the regularization benefit justifies the training complexity.

C. Model Behavior and Specialization

M3's stable multi-seed training performance reflects effective adaptation to diverse joint failure conditions, enabling superior robustness (3.80m average under joint failures, 8.26m baseline) compared to M4's incomplete recovery (3.39m failures, 6.70m baseline). M2 demonstrates clear specialization: excellent noise robustness (100.7% retention at 10$\times$ training noise) but poor joint failure adaptation (30.7% average retention). The smoothness constraint prevents reactive, noise-sensitive behaviors required for failure compensation—the interference mechanism observed in M4. Joint $\times$ noise multiplicative degradation indicates sensor noise and joint failures stress different locomotion subsystems, yet combining their training methods produces gradient conflict rather than modular robustness.

D. Implicit and Explicit Robustness Mechanisms

Observation normalization ablation revealed normalization is a prerequisite for learning (not implicit robustness): models without normalization exhibited >99% performance collapse due to RealAnt's 200:1 observation scale ratio causing gradient domination [17]. Methodological implications: (1) SR2L and DR require normalized observations as infrastructure, (2) fair robustness comparisons require consistent normalization, (3) M1's moderate noise tolerance stems from gradient stabilization enabling learning, not a robustness mechanism. SR2L achieves noise-induced regularization (up to 114% performance at moderate noise) with cross-distribution generalization (106.6% Poisson, 112.8% salt-and-pepper despite Gaussian-only training)—smoothness creates policies that tolerate and benefit from observation perturbations.

E. Deployment Recommendations

Based on 38,000 evaluation episodes across 5 systematic experiments, we recommend:

Universal recommendation: Deploy M3 (DR). Across all scenarios—sensor noise, joint failures, combined stress, and clean environments—M3 achieves best or near-best performance while providing the highest baseline performance (8.26m). Domain randomization acts as regularization, preventing overfitting and delivering robustness without sacrifice.

Key finding on sensor noise: Observation normalization alone (M1) provides substantial robustness to sensor noise (83% retention at 10$\times$ training levels). SR2L training (M2) adds only marginal improvement (+17.7% to 100.7% retention) while sacrificing baseline performance (7.26m vs 8.26m). Notably, M3 (trained only for joint failures) achieves 92.2% noise retention as cross-robustness—better than M1 without specialized noise training. **Implication:** For Gaussian sensor noise matching training distributions, explicit noise training provides minimal benefit over normalization; practitioners should focus on joint failure adaptation. However, SR2L remains valuable when facing unknown noise distributions or non-Gaussian perturbations—its smoothness regularization provides theoretical guarantees for bounded perturbations that pure normalization cannot offer. Consider SR2L when deploying to environments with fundamentally different sensor characteristics than simulation.

Key finding on joint failures: M3 (DR) dominates joint failure robustness (53.3% hip retention, 41.7% ankle retention). M4 (combined) offers no advantage over M3 alone despite training for both noise and failures, confirming that multi-objective training creates interference rather than synergy.

General principle: Train for the most challenging failure mode (joint failures via DR), which provides cross-robustness to other perturbations as a side benefit. Attempting to train for multiple failure modes simultaneously degrades performance across all objectives.

F. Limitations and Future Work

Our study has several limitations. First, **simulation-only evaluation**: we used RealAnt simulation rather than physical hardware; sim-to-real transfer may reveal additional challenges including actuator dynamics, sensing delays, and physical contact modeling that could affect robustness comparisons. Second, **limited hyperparameter exploration**: we used standard hyperparameter settings from prior work (SR2L: $\lambda = 0.001$, $\sigma = 0.01$ from [6]; PPO defaults from Stable-Baselines3 [20]). Due to compute constraints and time limitations (our experiments consumed approximately 800 GPU-hours), comprehensive hyperparameter sweeps were not feasible; alternative parameter regimes might reveal conditions where interference reduces or synergy emerges. Third, our DR curriculum was hand-designed; learned curricula might better balance objectives. Finally, we focused on PPO; other algorithms (SAC, TD3) might exhibit different interaction patterns.

Future work should explore: (1) comprehensive hyperparameter sweeps to identify parameter regimes where method combinations may be more effective, (2) alternative combination strategies that avoid gradient conflict (e.g., sequential training, importance-weighted objectives), (3) theoretical analysis of when robustness methods are compatible vs. conflicting, (4) extension to other locomotion platforms and failure modes, (5) physical robot validation of our deployment recommendations.

VI. Related Work

A. Robustness Methods: SR2L and Domain Randomization

Smooth Regularized Reinforcement Learning (SR2L) [6] is a general RL technique that introduces smoothness constraints through a regularization term $L_{SR2L} = \mathbb{E}[\|\pi_\theta(s) - \pi_\theta(s + \delta)\|^2]$ that penalizes policy sensitivity to observation perturbations, creating robustness to observation perturbations, including sensor noise in robotic applications. Related approaches enforce Lipschitz continuity [21] or adversarial perturbations [22].

Domain randomization (DR) [4], [5] trains on randomized environment parameters; applications span robotic manipulation, autonomous driving, and legged locomotion. Recent extensions include active domain randomization that automatically tunes randomization ranges [7] and multi-modal randomization combining visual, dynamics, and actuator variations for improved sim-to-real transfer [23].

Curriculum learning [10], which structures training as progression through increasing difficulty levels, has become a standard component of DR for robotics. For quadruped locomotion with actuator failures, curriculum DR typically begins with clean environments to establish baseline gaits, then gradually introduces randomized joint failures at increasing frequencies [11]. This phased approach prevents early training collapse that would occur if full randomization were applied from initialization, significantly improving both training success rates and final robustness. Our M3 and M4 models employ 3-phase curriculum schedules (detailed in Section III) following this established paradigm.

While both SR2L and DR (with curriculum learning) address robustness individually, evaluating their interaction when combined remains an open question.

B. Combining Robustness Methods as Multi-Objective Learning

When training a policy with both SR2L and DR (our M4 model), we create a multi-objective learning problem: the policy must simultaneously (1) move forward quickly, (2) maintain smooth actions under observation noise (SR2L), and (3) adapt to joint failures (DR). The natural hypothesis is that these objectives combine additively—each method handles its designated failure mode without interfering with the others.

However, multi-task learning research shows that objective combinations can produce negative transfer, where training for multiple goals simultaneously yields worse performance than training for each goal separately [18]. We demonstrate this phenomenon in the robustness domain: SR2L and DR create conflicting gradient signals that cause training collapse, making M4 (combined) worse than M3 (DR alone) across all metrics.

C. Quadruped Locomotion and Implicit Robustness

Deep RL has achieved impressive quadruped locomotion performance across platforms including ANYmal [1], MIT Cheetah [9], and Unitree Go1 [2], typically using DR as an implicit robustness mechanism. Recent work demonstrates locomotion over challenging terrain [3], recovery from external perturbations, and zero-shot sim-to-real transfer through extensive randomization. However, systematic studies of robustness methods remain limited.

Observation normalization [16], a standard technique that maintains running statistics to normalize observations to zero mean and unit variance, is typically viewed as a training stabilizer. Our controlled ablation reveals observation normalization is a prerequisite for SR2L and DR training (>99% collapse without normalization, $p < 0.001$), clarifying that normalized observations are required infrastructure with critical implications for fair method comparison [17].

VII. Conclusion

Through approximately 38,000 evaluation episodes across four training methods, we reveal both unexpected mechanisms and important findings. We establish clear evidence that (1) for quadruped locomotion, domain randomization with curriculum training acts as regularization and achieves best baseline performance (M3: 8.26m vs M1: 6.40m, $p < 0.001$, 29% improvement), (2) combining SR2L and DR produces interference rather than synergy (M3 outperforms M4 by 56-145% under combined stress, all $p < 0.001$), (3) observation normalization is a prerequisite for SR2L/DR training (>99% collapse without normalization, $p < 0.001$), and (4) SR2L exhibits noise-induced regularization with remarkable cross-distribution generalization (up to 114% performance at moderate noise levels across distributions).

The overfitting finding fundamentally changes the robustness training cost-benefit analysis. Rather than sacrificing baseline performance for fault tolerance, domain randomization provides "free" performance improvement through regularization—paralleling data augmentation in supervised learning. Multi-seed training curves reveal M3 maintains highest training rewards and achieves best evaluation (8.26m), while M1's moderate training rewards mask severe overfitting (worst evaluation: 6.40m). This demonstrates that domain randomization enables effective learning and generalization simultaneously.

Our negative result that multi-objective robustness training underperforms specialized approaches contradicts the natural hypothesis of additive benefits. Through training dynamics analysis (Figure 3), we demonstrate that this stems from a fundamental incompatibility: SR2L's smoothness penalty opposes DR's adaptation requirement when joint failures induce significant changes in observations, leading to catastrophic collapse during training. This finding has immediate practical implications: practitioners should train specialized models for specific failure modes rather than attempting to handle all robustness dimensions simultaneously.

The discovery that SR2L and DR require observation normalization for basic locomotion ($>$99% collapse without normalization) reveals normalization is prerequisite infrastructure, not a robustness mechanism. High-dimensional continuous control with vastly different numerical ranges (a 200:1 ratio in RealAnt) causes gradient domination, which prevents learning without normalization [15]. This clarifies that robustness methods operate on normalized observations, with implications for fair experimental comparison. Future quadruped RL systems should recognize normalization as required infrastructure for SR2L/DR methods.

Deployment Recommendations. For real-world quadruped systems:

1. Deploy M3-style domain randomization universally. In our experiments, achieves best baseline (8.26m), best joint failure robustness (46.0%), and excellent noise tolerance (92.2%) without multi-objective training complexity.

2. Use observation normalization as required infrastructure. Enables SR2L/DR training; $>$99% collapse without it.

3. Train specialized models per failure mode. Avoid combined training to prevent gradient interference—M3 outperforms M4 by 57-145%.

5. Evaluate on test conditions, not just training metrics. M1's moderate training rewards masked severe overfitting.

This work contributes quantitative evidence, methodological frameworks (comprehensive ablation methodology with 38,000 episodes, multi-distribution generalization testing, controlled observation normalization ablation), and theoretical insights (mathematical formulation of gradient conflict, identification of observation normalization as prerequisite for SR2L/DR methods, characterization of noise-induced regularization) that advance understanding of robustness in learned locomotion policies. The finding that specialized training outperforms multi-objective approaches challenges conventional wisdom and provides actionable guidance for deploying robust quadruped systems.

REFERENCES

[1] 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, p. eaau5872, 2019.

[2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, M. Khadiv, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, 2022.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems (RSS)*, 2021, demonstrates adaptation to unseen environments and perturbations.

[4] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[5] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[6] Q. Shen, Y. Li, H. Jiang, Z. Wang, and T. Zhao, "Deep reinforcement learning with robust and smooth policy," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 8707–8718.

[7] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active domain randomization," in *Conference on Robot Learning (CoRL)*. PMLR, 2020, pp. 1162–1176.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[9] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadruped locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*. ACM, 2009, pp. 41–48, foundational paper on training with progressively increasing difficulty.

[11] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *Frontiers in Robotics and AI*, vol. 9, 2022, comprehensive review including curriculum DR.

[12] Farama Foundation, "Gymnasium: A standard interface for reinforcement learning environments," https://gymnasium.farama.org/, 2023, open-source successor to OpenAI Gym.

[13] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 5026–5033.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[15] S. Huang, R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang, "The 37 implementation details of proximal policy optimization," ICLR Blog Track, 2022, documents observation normalization as critical PPO implementation detail. [Online]. Available: https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

[16] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017, vecNormalize implementation.

[17] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, S. Gelly, and O. Bachem, "What matters in on-policy reinforcement learning? a large-scale empirical study," *International Conference on Learning Representations (ICLR)*, 2021, empirically demonstrates observation normalization is critical for PPO performance.

[18] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 9120–9132, negative transfer in multi-task learning.

[19] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019, data augmentation as regularization in supervised learning.

[20] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.
[21] T. Yu, S. Sugano, J. Nakanishi, and J. Morimoto, "L2c2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 25 688–25 707.
[22] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 2817–2826.
[23] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019, multi-modal domain randomization (visual + dynamics + actuator).

Appendix

A. Training Infrastructure

Hardware: All models trained on cluster compute nodes equipped with NVIDIA RTX 8000 GPUs (48GB VRAM), Intel Xeon Gold 6248R CPUs (24 cores @ 3.0GHz), 192GB system RAM.

Software: Python 3.9.7, PyTorch 1.12.0, Stable-Baselines3 1.6.0, Gymnasium 0.28.1, MuJoCo 2.3.0, CUDA 11.6.

Random Seeds: All models were trained with 5 random seeds for statistical robustness:

- Seeds used: 42, 123, 456, 789, 999
- Each model (M1, M2, M3, M4) trained with all 5 seeds
- Learning curves show mean ± 1 SD across seeds

B. Training Duration

Total Compute Time (per seed):

- M1 (Baseline, 32M steps): 36.2 hours
- M2 (SR2L, 32M steps): 41.8 hours (15% overhead from smoothness loss)
- M3 (DR, 32M steps): 44.1 hours (22% overhead from environment randomization)
- M4 (Combined, 32M steps): 47.3 hours (combined overheads)

Aggregate Compute: 847 GPU-hours total (169.4 hours per model × 5 seeds), excluding evaluation episodes. This aligns with the 800 GPU-hours reported in Section VI.

C. Hyperparameters

All hyperparameters detailed in Section 3.5 (PPO Training Configuration) and Section 2.4 (SR2L Hyperparameter Selection). Key parameters:

- Learning rate: 3×10^{-4}
- Batch size: 1536
- PPO epochs: 8
- Clip range: 0.15
- GAE λ: 0.97
- Discount γ: 0.995
- SR2L λ: 0.001
- SR2L σ: 0.01

D. Evaluation Protocol

Episodes per condition: 100 episodes for all quantitative comparisons (sufficient for statistical power with Bonferroni correction).

Episode length: 1000 timesteps or until termination (torso height < 0.2m).

Total evaluation episodes: Approximately 38,000 episodes across all experiments.

E. Statistical Methodology and Seed Usage

Training vs. Evaluation Seeds: It is important to distinguish between training seeds and evaluation methodology:

Training Phase (Learning Curves):

- All models trained with 5 independent random seeds (42, 123, 456, 789, 999)
- Training dynamics analysis uses seed-averaged data to show robust trends
- Catastrophic collapse in M4 observed consistently across multiple seeds

Statistical Pairing for Comparisons:

- Paired t-tests use **episode-level pairing**: episode i of model A is paired with episode i of model B
- All models evaluated on **identical environment initializations** (same random seed sequence: 0-99)
- This ensures each model faces the exact same initial robot pose, terrain, and stochastic dynamics
- Pairing reduces variance from environment stochasticity, increasing statistical power
- Bonferroni correction applied with $\alpha = 0.05/n$ where n is the number of comparisons per experiment

Why This Approach:

- Training curves (mean ± SD) demonstrate **reproducibility** across different random initializations
- Final evaluation on median seeds provides **representative single-model performance** for deployment recommendations
- Episode-level pairing with identical environment seeds ensures **fair head-to-head comparisons**
- This methodology balances statistical rigor with practical computational constraints

F. Code and Data Availability

Code, trained models, and evaluation data are available at: https://github.com/anand1221178/robust-quadruped-rl. Implementation based on Stable-Baselines3 with custom environment wrappers for SR2L smoothness regularization and domain randomization.

Wits University Faculty of Science post-graduate student AI declaration

I understand that the use of generative AI tools (such as ChatGPT or similar) without explicitly declaring such use constitutes a form of plagiarism and is classified by Wits University as academic misconduct.

I declare that in the course of conducting the research towards my degree or in the preparation of this thesis/dissertation/research report (select one by marking with an X):

I **did not** make use of generative AI tools [ ]

I **did** make use of generative AI tools for the following (tick all that apply):

1. Idea Generation (research problem/design, hypothesis) [ ]
2. Sourcing Related Work (summarising, identifying sources) [ ]
3. Methods and Experiment Design (experiment setup, model tuning) [ ]
4. Data Analysis (presentation, coding, interpretation) [X]
5. Theoretical Development (theorem proving, conceptual analysis) [ ]
6. Code Development (generating algorithms, writing scripts) [ ]
7. Presentation (rendering graphics, formatting) [ ]
8. Editing (grammar, readability) [X]
9. Writing (text generation, document structuring) [ ]
10. Citation Formatting (structuring, organising) [X]

If other uses were involved, please specify below:

Generative AI tool used (list all)	Used for?

If generative AI tools were used as an integral part of the experimental design or in the direct execution of my research, I confirm that details of this use are clearly outlined in the relevant experimental/methodology chapters of my thesis/dissertation/research report.

Student number: 2561034

Candidate signature: PatelA

Date: 10/11/2025