SLowRL: Safe Low-Rank Adaptation in Reinforcement Learning for Locomotion

Elham Daneshmand[1,2], **Shafeef Omar**[4], **Glen Berseth**[2,3], **Majid Khadiv**[4], **Hsiu-Chin Lin**[1,2]

elham.daneshmand@mail.mcgill.ca, shafeef.omar@tum.de, glen.berseth@mila.quebec, majid.khadiv@tum.de, hsiu-chin.lin@mcgill.ca

[1]McGill University [2]Mila – Quebec AI Institute [3]Université de Montréal [4]Technical University of Munich

Abstract

A simulator is, at best, a coarse low-fidelity model of the real world the agent eventually has to act in. Closing this residual gap on hardware is a canonical instance of operating in a *big world*: the real environment exposes contact dynamics, latencies, and disturbances that the agent was never given the capacity (parameters *or* data) to model during pretraining. Naive on-hardware fine-tuning is risky — the policy can damage the robot before it improves — and full-parameter updates require prohibitive interaction time. We propose $SLowRL$, a continual fine-tuning framework that confronts this big-world adaptation problem with two complementary forms of capacity limitation: (i) a *rank-1* LoRA adapter applied per layer to both actor and critic, restricting each layer's update to a single direction in its image space ($\sim 0.9\%$ of FFT's trainable parameter count) over a frozen sim-trained policy, and (ii) a task-agnostic recovery policy gated by a discrete safety filter that pulls the robot back to a nominal state whenever proprioceptive limits are violated. On sim-to-sim and sim-to-real fine-tuning of a Unitree Go2 quadruped robot, $SLowRL$ reaches the pretraining reward in $38-55\%$ of the wall-clock time of full fine-tuning, eliminates 96% of falls, and *still* matches or exceeds full-parameter baselines on the dynamic jump task — evidence that under big-world distribution shift, a smaller update adapts faster and moves more smoothly, while a lightweight safety filter keeps it safe.

1 Introduction

Reinforcement learning agents trained in simulation are agents whose internal models have far less capacity than the world they are eventually deployed in. The sim-to-real gap is precisely the mismatch between a finite, randomized simulator and an open, partially observed physical environment with hard latencies, contact discontinuities, and unmodeled actuator dynamics. Because no fixed-size policy and no finite simulator can fully represent this target world, every real-robot deployment is, at some level, a *continual* adaptation problem: the agent must keep refining itself from a small budget of on-hardware interaction without catastrophic failure.

This setting exposes two coupled constraints that motivate this work. **(C1) Limited safe interaction budget.** On hardware, every exploratory action can damage the robot; the agent therefore cannot afford the millions of unsafe steps that simulators happily provide. **(C2) Limited update capacity.** Even when interaction is available, large-capacity updates on small data overfit to noise and produce high-frequency, bang-bang torque artefacts that further endanger the system (Mysore et al., 2021).

Standard proximal policy optimization (PPO)-based full fine-tuning ignores both constraints: it adapts every parameter, requires hours of real-world interaction, and routinely falls during exploration.

We argue that both constraints push in the same direction — toward agents that adapt to the big world by changing as little as possible. Concretely, we introduce $SLowRL$ (Figure 1), which fine-tunes a sim-trained policy under two parallel restrictions. Let W_0 denote the parameters of the sim-trained policy. We apply the following two restrictions in parallel:

- A *rank-1 Low-Rank Adaptation (LoRA) adapter* is applied to every dense layer of both actor and critic, while the pretrained weights W_0 remain frozen. Each layer's update is constrained to a rank-1 outer product $\Delta W = ba^\top$, dropping the trainable parameter budget per layer from $d \times k$ to $d + k$ ($\sim 0.9\%$ of full fine-tuning across the network).
- A *task-agnostic recovery policy* π_r, gated by a discrete proprioceptive safety filter, returns the robot to a nominal upright state whenever roll, pitch, or hip height leave a safe envelope — keeping exploration inside the recoverable set.

Empirical findings. Across two locomotion tasks (trot and jump) and both sim-to-sim and sim-to-real protocols on a Unitree Go2, $SLowRL$:

- Recovers pretraining reward in 38% (trot) and 55% (jump) of the wall-clock time of Full Fine-Tuning (FFT), while updating only $\sim 0.9\%$ of the parameters;
- Reduces failures during fine-tuning from 14.25/69.0 (FFT w/o safety) and 7.5/17.5 (FFT w/ safety) to 0.0/2.0 averaged over 4 seeds;
- On a real Go2 jump task, improves mean reward from -19 to -7 in 60 minutes of on-robot training while FFT plateaus near -15.

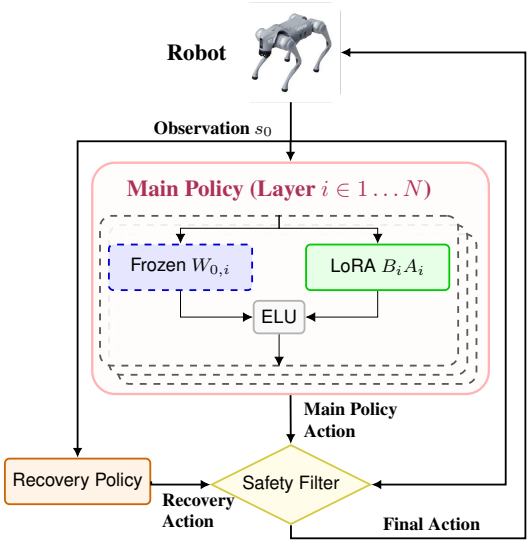


Figure 1: The $SLowRL$ architecture. A frozen sim-trained policy W_0 and a rank-1 LoRA adapter $\Delta W = BAx$ act in parallel. A discrete safety filter monitors proprioceptive state and hands control to the task-agnostic recovery policy whenever the robot leaves the safe envelope.

Contribution. We provide empirical evidence that for structured contact-rich locomotion, the residual sim-to-real gap can be closed by a per-layer rank-1 update over the frozen sim-trained policy — i.e. a low-dimensional submanifold of the parameter space — and that pairing this capacity restriction with a simple state-space safety filter is sufficient for safe, fast continual adaptation on real hardware. This is consistent with the workshop's framing of agents whose capacity is much smaller than the world they must act in.

2 Related Work

Sim-to-real for locomotion has historically been addressed by domain randomization (Tan et al., 2018; Yang and Yin, 2021; Bogdanovic et al., 2021), system identification (Hwangbo et al., 2019), or latent adaptation (Kumar et al., 2021; Miki et al., 2022). These methods assume the real world is in-distribution under a wide simulator ensemble — a strong assumption that ignores the residual big-world gap and tends to yield conservative policies. Latent-adaptation in particular (RMA, Kumar et al. 2021) infers privileged context at deployment but does not update policy weights, so it cannot close residual gaps that no simulator-side latent encodes; residual-RL approaches (Johannink et al., 2019; Ankile et al., 2024) keep the pretrained policy frozen but learn an *unrestricted-capacity* additive correction, preserving prior knowledge yet offering neither a parameter-efficiency nor a safety prior.

$SLowRL$ is complementary to both latent adaptation and residual RL, and can in fact be read as a *low-rank, safety-gated* residual: a structural prior on *how much* the policy may change (rank-1 per layer), paired with a structural prior on *when* on-policy exploration is allowed (the recovery filter).

On-hardware fine-tuning addresses the residual gap directly. Recovery RL (Thananjeyan et al., 2020) and constraint-layer methods (Liu et al., 2023) use safety critics or projection layers, but the safety signal is itself subject to the sim-to-real gap. Yang et al. (2022) report ~ 115 minutes per task, and Smith et al. (2023) reach ~ 20 minutes by restricting to statically stable behaviours — inapplicable to dynamic or underactuated tasks (Daneshmand et al., 2021).

Parameter-efficient adaptation. LoRA (Hu et al., 2022) has reshaped fine-tuning for large language models; recent theoretical work (Schulman and Lab, 2025) suggests that on-policy RL is information-limited and that the effective rank of useful updates is small. To our knowledge, $SLowRL$ is the first to evaluate LoRA (and specifically rank-1 LoRA) for high-frequency continuous control on real legged hardware.

3 Background

We model the robot as a discounted Markov Decision Processes (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ and optimize $J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_t \gamma^t r_t]$ with PPO (Schulman et al., 2017). We assume access to a pretrained policy with parameters $W_0 \in \mathbb{R}^{d\times k}$ obtained in a source simulator with domain randomization. Concrete simulator and training details appear in §5.

LoRA. Standard fine-tuning replaces $W_0 \in \mathbb{R}^{d\times k}$ with $W = W_0 + \Delta W$. LoRA assumes ΔW has low intrinsic rank $\rho \ll \min(d, k)$ and parametrises it as $\Delta W = BA$ with $B \in \mathbb{R}^{d\times\rho}, A \in \mathbb{R}^{\rho\times k}$. Each linear layer becomes

$$h = W_0 x + \tfrac{\alpha}{\rho} BAx, \qquad A \sim \mathcal{N}(0, \sigma^2),\ B = 0, \tag{1}$$

where α is a scaling hyperparameter that controls the magnitude of the update. The factors are initialized as $A \sim \mathcal{N}(0, \sigma^2)$ and $B = 0$, so the initial policy is exactly W_0 and only A, B are optimized during fine-tuning.

4 $SLowRL$: Safe Low-Rank Fine-tuning

$SLowRL$ has two stages. **(1) Sim pretraining**: we train a contact-conditioned base policy following Omar and Khadiv (2025) in IsaacLab, giving frozen weights W_0. **(2) Safe low-rank fine-tuning**: on the target domain (MuJoCo or the real robot) we attach a rank-1 LoRA adapter to every dense layer of both the actor and the critic, freeze W_0, and update only A, B with PPO. The control loop is gated by the recovery policy π_r.

4.1 Rank-1 LoRA-PPO

We apply LoRA with $\rho = 1, \alpha = 4$ to all linear layers of actor and critic. Initializing $B = 0$ guarantees the fine-tune trajectory begins at the sim-trained policy. *What "rank 1" means here.* Per layer, the update $\Delta W = ba^\top \in \mathbb{R}^{d\times k}$ lies in a rank-1 subspace, so its image-space contribution $\Delta W x$ is one-dimensional for any input x; across the L adapted layers the joint update spans an L-dimensional submanifold of the full $\sum_\ell d_\ell k_\ell$ parameter space. Two practical consequences follow:

- **Trainable parameters drop by $\approx 99.09\%$.**[1] With ΔW confined to a rank-1 outer product $ba^\top$, the optimizer searches over $d + k$ rather than $d \times k$ parameters per layer. The smaller landscape empirically converges in fewer rollouts (Section 5).

[1] For the $[82, 512, 256, 128]$ actor and critic MLPs (12-dim action, scalar value head), rank-1 LoRA introduces $\sum_\ell (d_\ell + k_\ell) \approx 3.8$k trainable parameters, versus $\sum_\ell d_\ell k_\ell \approx 413$k for full fine-tuning — i.e. 0.91%, a 99.09% reduction.

- **Bang-bang torque artefacts are suppressed.** A rank-1 update can shift each layer's output only along a single direction in its image space, bounding the per-step change in $\|a_t - a_{t-1}\|$ relative to unconstrained updates. The action rate stays low throughout training (Table 2), an implicit regulariser absent in full fine-tuning.

Adapting the critic is necessary. Under PPO, the actor update is driven by the advantage $\hat{A}_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ where V is the critic's value estimate. If V remains frozen at sim-trained values, it predicts returns under simulator dynamics — but the rollouts now come from the target domain. The resulting advantages systematically misrank actions, and the actor optimizes against a biased signal. Our ablation (Section 5) shows that LoRA-on-actor-only fails to converge and leaves the value loss two orders of magnitude above the actor+critic configuration.

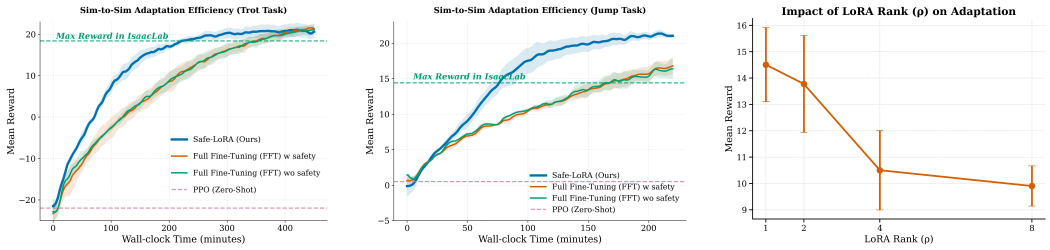
4.2 Discrete Safety Filter and Recovery Policy

Hardware safety is enforced by a discrete trigger acting on proprioceptive state. At every step, the filter checks three proprioceptive signals against hand-tuned envelope limits: roll $|\phi| < 0.6\,\mathrm{rad}$ ($\approx 34°$), pitch $|\theta| < 0.6\,\mathrm{rad}$, and hip height $z_{\text{hip}} > 0.20\,\mathrm{m}$ (against a nominal standing height of $\approx 0.30\,\mathrm{m}$). If any condition is violated, control is handed to a separately-trained, task-agnostic recovery policy π_r that drives the robot back to a nominal upright posture and is held for at least 100 control steps ($\approx 2\,\mathrm{s}$ at $50\,\mathrm{Hz}$) to avoid chattering. π_r is pretrained once in IsaacLab with broad initial-pose randomization (sampled tilt, lateral and angular velocities), the same domain-randomization ranges used for the main controller, and a large fall-termination penalty, so it generalises across the locomotion tasks the main policy is later fine-tuned on without further task-specific training. The filter is intentionally simple: a hand-designed envelope on three signals, not a learned safety critic. This avoids relying on a critic that is itself subject to the sim-to-real gap.

5 Experiments

We ask: **(Q1)** does the safety filter reduce failure rates without slowing learning? **(Q2)** does limiting capacity to rank 1 hurt asymptotic performance? **(Q3)** does the benefit come from training *fewer* parameters, or from adapting *every* layer within a low-rank subspace? **(Q4)** does $SLowRL$ transfer to a real Go2? **(Q5)** does the low-rank advantage depend on an already-converged warm-start?

Setup. We evaluate on a quadruped robot (Unitree Go2) across two locomotion tasks: *trotting* and *jumping*. The actor and critic are 3-layer MLPs with hidden sizes $[512, 256, 128]$ and ELU activations. The policy outputs 12 joint position targets (scaled by 0.35) at 50 Hz with 4 physics substeps at 200 Hz; the observation is $\mathbb{R}^{82}$ (base velocities, projected gravity, contact goals/plan, joint state, end-effector errors, previous action). Pretraining uses IsaacLab (NVIDIA, 2025) with 4096 parallel environments; fine-tuning uses MuJoCo (Todorov et al., 2012) (sim-to-sim) or the physical Go2 with Vicon motion capture (sim-to-real). Both sim-to-sim and sim-to-real are run in real time so per-iteration wall-clock cost is dominated by the rollout phase (1200 steps $\approx$ 24s).
Reward. Following Omar and Khadiv (2025), the reward combines contact-tracking terms (approach, release, maintain contact, foot clearance, goal completion) with regularization penalties (action rate, joint torques, joint acceleration, orientation, hip height); the dominant terms reward tracking the desired foot-placement plan while the regularization terms penalize unsafe or energy-inefficient behaviour. The same reward structure is used for pretraining and fine-tuning, with minor per-term weight adjustments to account for inter-simulator contact-dynamics differences (full specification in the parent paper). The "pretraining reward" reference line in Figs. 2a–2c is the mean Stage-1 reward attained on the source simulator. Sim-to-sim results are averaged over 4 seeds per method. Learning rates are tuned per method (10^{-2} for LoRA, 10^{-3} for FFT); all other PPO hyperparameters are shared. Baselines: *PPO zero-shot* (no fine-tuning), *FFT w/o safety* (full fine-tuning, no safety filter), and *FFT w/ safety* (full fine-tuning with our safety filter and recovery policy π_r).



(a) Trot, wall-clock (38% time). (b) Jump, wall-clock (55% time). (c) Rank sweep $\rho \in \{1, 2, 4, 8\}$.

Figure 2: **Sim-to-sim fine-tuning efficiency.** (a,b) $SLowRL$ converges to the pretraining reward faster than both FFT variants. Reward scales are task-specific—the dashed PPO zero-shot and IsaacLab-pretraining reference lines differ between trot and jump—so absolute rewards are not comparable across panels (a) and (b). (c) Rank-1 reaches pretraining reward first; higher ranks converge more slowly under the same 75-minute budget and do not surpass rank-1 even after 230 minutes.

5.1 Safety and Smoothness (Q1)

Table 1 reports the mean number of falls/collisions per seed during fine-tuning. $SLowRL$ drives failures to near zero on both tasks; importantly, *adding safety to FFT alone is not enough* (still 7.5/17.5 falls) — low-rank parameterisation contributes most of the reduction. Table 2 corroborates this: the action-rate $\|a_t - a_{t-1}\|$ drops by 88.9% (trot) and 35.7% (jump) over training under $SLowRL$, versus 38.9% and 5.7% for FFT, indicating that the rank constraint actively suppresses bang-bang behaviour. The two restrictions address different failure modes: the rank-1 constraint suppresses the high-frequency action artefacts that make updates unsafe to *apply*, while the discrete safety filter catches the falls that remain — and neither alone suffices, since FFT *with* the same filter still incurs 7.5/17.5 falls (Table 1).

Table 1: Mean failures during fine-tuning (4 seeds).

Method	Trot	Jump
FFT *w/o safety*	14.25	69.0
FFT *w/ safety*	7.5	17.5
$SLowRL$ **(ours)**	**0.0**	**2.0**

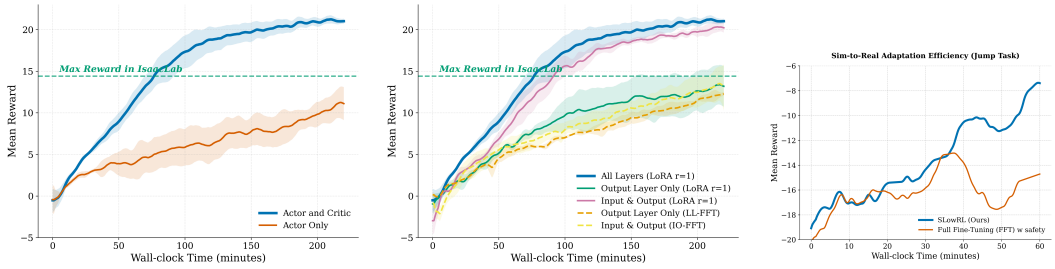
Table 2: Reduction in action rate $\|a_t - a_{t-1}\|$ from the start to the end of training. Larger values indicate stronger suppression of high-frequency oscillation.

Method	Trot	Jump
FFT *w/ safety*	38.9%	5.7%
$SLowRL$ **(ours)**	$\mathbf{88.9\%}$	$\mathbf{35.7\%}$

5.2 Sample Efficiency in Sim-to-Sim (Q2, Q3)

Figure 2 plots reward vs. wall-clock time in the IsaacLab→MuJoCo transfer. $SLowRL$ reaches the pretraining reward in 38% (trot) and 55% (jump) of FFT's time. Because all methods use identical 1200-step rollouts (one gradient update per rollout), each iteration costs the same number of physical interactions, so the gap is not an artefact of cheaper iterations — $SLowRL$ simply requires fewer rollouts and fewer gradient steps to converge (we verify this by replotting the same data against cumulative environment steps; the ordering and gaps are preserved).

The rank-1 hypothesis (Q2). Figure 2c sweeps $\rho \in \{1, 2, 4, 8\}$ at a fixed 75-minute budget. Rank 1 reaches the pretraining reward fastest; higher-rank adapters expand the optimization landscape and amplify gradient noise from contact discontinuities and safety resets, slowing convergence. Extending the budget to 230 minutes does not change the ordering. All ranks in Fig. 2c share the same LoRA learning rate (10^{-2}), so this ordering reflects the rank constraint rather than an optimisation artefact; the per-method learning-rate gap between $SLowRL$ and FFT is examined as a limitation in §6. This



(a) Actor+Critic vs. Actor-only. (b) Layer-wise LoRA placement. (c) Sim-to-real jump.

Figure 3: **Ablations and real-world transfer.** (a) Freezing the critic breaks adaptation. (b) All-Layers > Output-Only at matched parameter count. (c) On the physical Go2 jump task, $SLowRL$ improves mean reward from -19 to -7 in 60 min while FFT w/ safety plateaus near -15.

is empirical support for the workshop's framing that, in big-world adaptation, the optimal update lives on a low-dimensional manifold rather than across the full parameter set.

Is rank-1 a proxy for "fewer parameters" (Q3)? Figure 3b compares three placement strategies for rank-1 LoRA: *Output Only* (matched parameter budget, adapters only on the final layer), *In+Out*, and *All Layers*. All Layers wins, and the parameter-matched Output Only baseline is substantially slower. The benefit is therefore not just "train fewer weights" but "adapt every level of abstraction within a low-rank subspace".

Critic adaptation is necessary. Figure 3a compares LoRA on *Actor Only* vs. *Actor+Critic*. Actor-only fails to converge and the value loss plateaus near 0.2, vs. 0.007 for Actor+Critic, confirming that a frozen sim-trained critic provides biased advantages after domain shift.

5.3 Real-World Transfer (Q4)

We deploy $SLowRL$ on a Unitree Go2 for the dynamic jump task, selected because it is the more safety-critical behavior. Figure 3c reports reward against wall-clock time during a single 60-minute on-robot run per method. $SLowRL$ moves from -19 to -7; FFT w/ safety initially tracks $SLowRL$ but plateaus near -15 and degrades late in training. We attribute this late-stage instability to *critic drift*: unconstrained full-parameter critic updates diverge from true returns, producing deceptive advantages, while safety-triggered resets fragment the on-policy trajectory distribution. The low-rank constraint in $SLowRL$ limits the magnitude of critic updates and preserves the alignment between value predictions and observed returns. Contact-pattern consistency improves by 9% for $SLowRL$ relative to FFT. We report a single 60-minute run per method: on-hardware wall-clock cost makes multi-seed real-robot sweeps impractical, so we treat this result as corroborating the multi-seed sim-to-sim trends (Fig. 2) rather than as an independent statistical claim.

5.4 Robustness to Warm-Start Quality (Q5)

Does the low-rank advantage depend on a pretrained policy? We fine-tune from IsaacLab pretrained policy checkpoints at $50/100/250/2500$ PPO iters for both $SLowRL$ and layer-restricted FFT (3 seeds, 24 runs). Figure 4 shows reward rising monotonically with pretrained policy quality for both methods (no saturation), with $SLowRL$ leading FFT by $+3.6$ to $+6.3$ once the prior is non-degenerate (pretrain $\geq$ 100). At the degenerate extreme (pretrain $= 50$) $SLowRL$ is worse and high-variance—a rank-1 update cannot repair a broken prior—bounding the low-rank hypothesis to priors that have learned the core skill. All

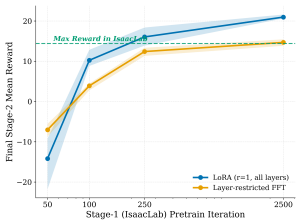


Figure 4: **Warm-start robustness.** Final Stage-2 reward vs. Stage-1 pretrain iters (3 seeds; band = min–max).

24 runs incurred *zero* safety terminations, so the safety filter generalises across the full range of pretraining quality, not just near a single anchor.

6 Discussion and Limitations

Why does "smaller" help here? Two mechanisms compound. (i) *Information regime*: PPO on-policy data carries limited information per rollout, so the largest update direction that can be reliably identified is also low-rank (Schulman and Lab, 2025). (ii) *Inductive prior*: the sim-trained W_0 already encodes the bulk of the locomotion skill; the residual mismatch is geometric (latencies, friction) rather than semantic. Both arguments are consistent with the workshop premise that big-world adaptation should be done by capacity-limited agents that piggyback on coarse but useful models.

Limitations. Our evaluation is restricted to flat-ground trot and jump; highly unstructured terrain (stairs, debris) may exceed the expressive capacity of a rank-1 update, and the hand-designed proprioceptive envelope is not terrain-adaptive. More principled alternatives such as Control Barrier Functions or learned terrain-aware safety critics are natural next steps, but our results suggest that any learned safety boundary should be *trained or adapted on the target domain*, not transferred zero-shot from simulation. We tune the learning rate per method (10^{-2} for LoRA, 10^{-3} for FFT) following standard practice for low-rank vs. full-rank optimization; an LR-matched sweep and an ablation of LoRA without the safety filter would further disentangle the contributions of capacity restriction, optimization, and gating. Broader parameter-efficient and safety-layer baselines (adapters in normalization layers, head-only fine-tuning, CBF/QP projections) are left to future work; the layer-placement ablation (Fig. 3b) already controls for raw parameter count at fixed LoRA structure. Reliance on PPO also limits sample efficiency; off-policy and offline-RL fine-tuners that can replay the (limited) real-world data are a promising direction.

7 Conclusion

We presented $SLowRL$, a continual sim-to-real fine-tuning method that addresses big-world adaptation by deliberately limiting the agent's capacity in two ways: a rank-1 LoRA update over a frozen sim-trained policy and a task-agnostic recovery policy gated by a discrete safety filter. Across sim-to-sim and sim-to-real evaluation on a Unitree Go2, this combination cuts adaptation time by $\sim 46.5\%$ on average, drives falls to near zero, and outperforms full fine-tuning on a real dynamic jump task. The result supports the view that, in big-world settings, useful agent updates are low-dimensional — and that exploiting this property is essential for safe and efficient hardware adaptation.

References

L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal. From imitation to refinement – residual rl for precise visual assembly. *arXiv preprint arXiv:2407.16677*, 2024.

M. Bogdanovic, M. Khadiv, and L. Righetti. Model-free reinforcement learning for robust locomotion using demonstrations from trajectory optimization. *Frontiers in Robotics and AI*, 9, 2021.

E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti. Variable horizon mpc with swing foot dynamics for bipedal walking control. *IEEE Robotics and Automation Letters*, 6(2):2349–2356, Apr. 2021. ISSN 2377-3774. doi: 10.1109/lra.2021.3061381. URL http://dx.doi.org/10.1109/LRA.2021.3061381.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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*, 4(26), Jan. 2019. ISSN 2470-9476.

doi: 10.1126/scirobotics.aau5872. URL http://dx.doi.org/10.1126/scirobotics.aau5872.

T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019. doi: 10.1109/ICRA.2019.8794127.

A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: rapid motor adaptation for legged robots. In D. A. Shell, M. Toussaint, and M. A. Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Z. Liu, Z. Guo, Y. Yao, Z. Cen, W. Yu, T. Zhang, and D. Zhao. Constrained decision transformer for offline safe reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822.

S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1810–1816. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561138.

NVIDIA. Isaac Sim. https://github.com/isaac-sim/IsaacSim, 2025. Version 5.1.0, Apache-2.0 license.

S. Omar and M. Khadiv. Learning to act through contact: A unified view of multi-task robot learning, 2025. URL https://arxiv.org/abs/2510.03599.

J. Schulman and T. M. Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

L. M. Smith, Y. Cao, and S. Levine. Grow your limits: Continuous improvement with real-world rl for robotic locomotion. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10829–10836, 2023.

J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In H. Kress-Gazit, S. S. Srinivasa, T. Howard, and N. Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018. doi: 10.15607/RSS.2018.XIV.010.

B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. P. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg. Recovery rl: Safe reinforcement learning with learned recovery zones. *IEEE Robotics and Automation Letters*, 6:4915–4922, 2020.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

T.-Y. Yang, T. Zhang, L. Luu, S. Ha, J. Tan, and W. Yu. Safe reinforcement learning for legged locomotion. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2454–2461, 2022. doi: 10.1109/IROS47612.2022.9982038.

Z. Yang and Z. Yin. Efficient hyperparameter optimization for physics-based character animation. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 4:1 – 19, 2021.