SLR: Learning Quadruped Locomotion without Privileged Information

Anonymous Author(s)
Affiliation
Address
email

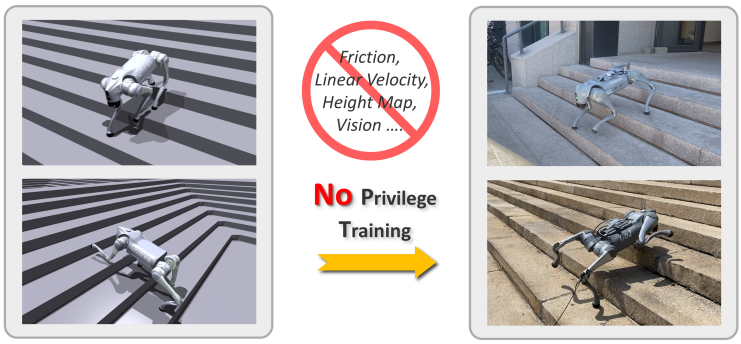

Figure 1: We present a framework designed to train a robust quadruped locomotion policy, eliminating the necessity for privileged information. The robot adeptly maneuvers through diverse terrains in simulation and exhibits comparable performance in corresponding real-world environments, showcasing a remarkable level of proficiency.

Abstract: Traditional reinforcement learning control for quadruped robots often relies on privileged information, demanding meticulous selection and precise estimation, thereby imposing constraints on the development process. This work proposes a Self-learning Latent Representation (SLR) method, which achieves high-performance control policy learning without the need for privileged information. To enhance the credibility of our proposed method's evaluation, SLR is compared with open-source code repositories of state-of-the-art algorithms, retaining the original authors' configuration parameters. Across four repositories, SLR consistently outperforms the reference results. Ultimately, the trained policy and encoder empower the quadruped robot to navigate steps, climb stairs, ascend rocks, and traverse various challenging terrains.

Keywords: Locomotion, Reinforcement Learning, Privileged Learning

1 Introduction

Humans and animals inherently possess locomotion abilities, enabling them to traverse various complex terrains. In contrast, gait control for robots is highly challenging. Model-based methods have achieved some success by leveraging robots' mechanical structures and dynamic principles [1, 2, 3, 4, 5]. However, finding a balance between model accuracy and computational efficiency remains difficult, especially for real-time applications.

Additionally, designing these models requires a deep understanding of a robot dynamics, posing a significant challenge for researchers. As a result, Reinforcement Learning (RL) methods are be-

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

coming increasingly popular. By simulating real-world environments and training policies with customized reward functions, these methods enable robots to perform complex locomotion tasks in real-time [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18].

Most RL applications in quadruped robots rely on privileged learning methods [19]. In real-world scenarios, a robot's interaction with its environment is modeled as a Partially Observable Markov Decision Process (POMDP). Solely relying on proprioceptive sensor measurements, a robot cannot fully perceive external environmental information, limiting its decision-making capabilities. Consequently, many studies leverage the "observability" advantages of simulation platforms. During training, various physical parameters (such as friction coefficients [10, 20, 21], restitution coefficients [9, 21], and scan-dots of the terrain [14, 22]) are artificially added as privileged information to help the robot understand both itself and the external environment.

In contrast, human cognition does not require explicit knowledge of physical parameters to traverse various terrains. Similarly, for robots using neural networks as their strategy, adding specific physical parameters may not yield the anticipated results. This discrepancy arises because we are not solving dynamic equations based on robot models but incorporating them as part of the neural network's input. Hence, these crafted privileged pieces of information may not necessarily provide comprehensibility or interpretability for neural network-based agents.

Therefore, instead of relying on manually chosen physical parameters to construct privileged information, this work explores whether it is possible for robots to learn a latent representation of environmental states by themselves? We believe that sufficiently intelligent robots should possess this capability, and representations learned through self-learning are more suitable for robots than those specified by humans.

To this end, a Self-learning Latent Representation (SLR) algorithm is proposed that generates latent representations guided by the Markov process of reinforcement learning without using any privileged information. Through self-learning, the robot can understand the latent features of the environment, demonstrating generalized locomotion capabilities across various terrains, as shown in Figure 1.

Our results indicate that, without relying on any privileged information, the SLR algorithm outperforms traditional privileged learning methods. Moreover, the learned latent representations are highly consistent with the actual terrain conditions across various terrains. To evaluate the proposed algorithm, the SLR algorithm is implemented in the open-source code repositories of previous researchers and compared in the same environments used by the authors. The SLR results demonstrated that this approach achieves state-of-the-art performance both in simulation and real-world deployments.

2 Related Work

Privileged learning in RL-based methods can be divided into explicit estimation and implicit estimation methods based on the target of supervised learning. In this paper, directly fitting specific physical parameters from privileged information is referred to as explicit estimation, while fitting the latent representation of privileged information is classified as implicit estimation.

Explicit Estimation. [6] concurrently trains a policy network and a state estimator, which include real-world parameters that are difficult to obtain accurately, such as linear velocity, foot height, and contact probability. [13] sets friction coefficients and stiffness coefficients as privileged information, inferring these from observation history to assist the robot in domain randomizations. But in the real world, the quadruped robot's foot contact time is very short during fast running, making it difficult to fully perceive ground friction. Therefore, [12] proposes learning information-gathering behaviors by adding an active estimation reward, which increases the accuracy of estimates for privileged information disparities.

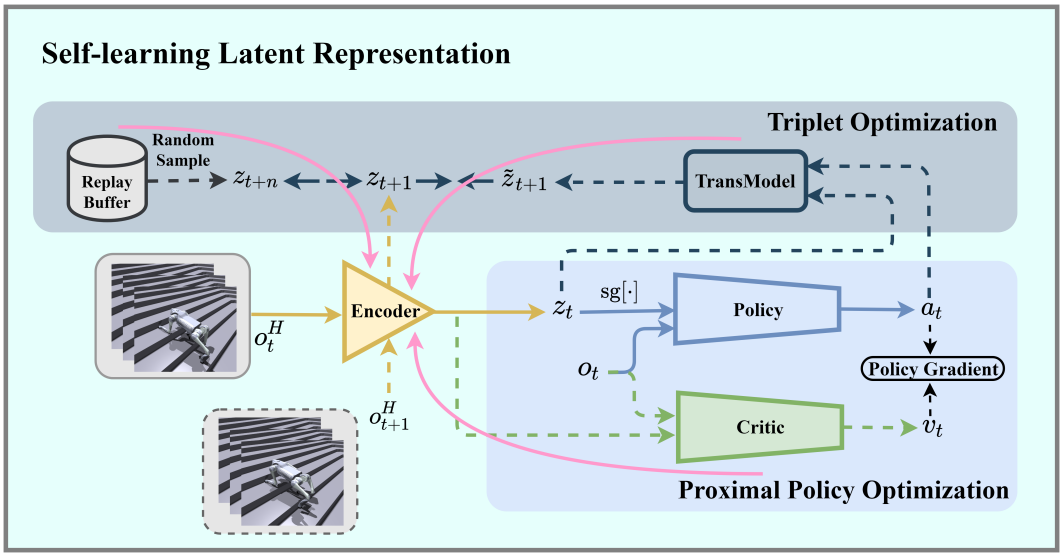

Figure 2: Illustration of SRL training framework. All dashed lines represent the network updating process. The solid pink line indicates the encoder updates through backpropagation from the Critic network, the transition model, and random sampling. The remaining solid lines represent the network's forward inference process.

Implicit Estimation. To train quadruped robots on complex terrain, estimating a large amount of privileged information explicitly is challenging. A common approach is to encode this high-dimensional information into a latent representation. [10] utilizes a teacher encoder to compress data such as friction and terrain height into a latent representation, and then a student adaptation module learn to infer this latent representation from observation history. [11] optimized the teacher-student training process from [10] in one stage by regularizing the teacher's actual privileged latent and supervising the student's estimated privileged latent. The algorithm of [9] also includes a teacher-student policy. During training, the teacher's encoder is used, and the student's adaptation module output aligns with the teacher's encoder. For deployment, the student's adaptation module and the teacher's trained policy are utilized together. [8] employs the Asymmetric Actor-Critic (AAC) [7] method, feeding privileged information to the critic network while using encoder-decoder estimators to assist the actor in imagining this privileged information.

3 Method

3.1 Problem Formulation

Our goal is to construct a one-stage end-to-end system based on RL, using proprioceptive sensor data as input for measuring and controlling joint movements.

Observation Space. The observations o_t consist of 45 dimensions, including the robot's base angular velocity ω, commands from the joystick c_t=[v_x^{cmd},v_x^{cmd},$\omega_{\text{yaw}}^{\text{cmd}}$], measurement of the gravity vector g_t, joint positions θ and velocities $\dot{\theta}$, and the actions a_{t-1} taken by the robot at the previous time step.

Action Space. The action space is a 12-dimensional vector a_t corresponding to the four legs of the quadruped robot, with each leg having three motor drive units. The neural network's output is converted into actual torque τ through a PD controller.

Reward Functions. The reward functions we used during the training are shown in Table A1, which come from [10, 16, 23].

Domain Randomizations. For the training of our method, domain randomizations are used. The details are shown in Table A2, which come from [8, 21].

3.2 Framework Overview

The proposed training framework fully leverages the Markov Decision Process (MDP), guiding the latent's self-learning based on state transitions, state distinctions, and cumulative rewards, without relying on manually set privileged information constraints. All networks used in this framework are multi-layer perception (MLP). The training framework is shown in Figure 2.

Encoder: In this architecture, the encoder's input consists of the observation history o_t^H, which is composed of proprioceptive information o_t from the previous 10 time steps. The encoder outputs a latent representation z_t of the observation history:

$$z_t = \phi(o_t^H) \tag{1}$$

Actor-Critic: The policy (Actor) and Critic network are trained jointly via Proximal Policy Optimization (PPO) algorithm [24]. The policy takes as input the current proprioceptive observation o_t and the latent representation z_t, and it outputs the joint position a_t. The Critic network, which shares the same input as the policy, outputs the state value v_t. It is worth noting that, we turn off the gradient of z_t in the policy and turn it on in the Critic network, using the backpropagation of the Critic network to update the encoder in the direction of the maximum cumulative reward:

$$a_t = \pi\left(o_t, \mathrm{sg}[z_t]\right) \tag{2}$$

$$v_t = V\left(o_t, z_t\right) \tag{3}$$

where $\mathrm{sg}[\cdot]$ is the stop gradient operator.

State Transition Model: The state transition model simulates the real state transitions of the environment $p\left(s_{t+1} \mid s_t, a_t\right)$ and shares the same network structure as the encoder. Its input is the latent representation z_t and the action a_t, and it outputs the next time step's latent estimation $\tilde{z}_{t+1}$:

$$\tilde{z}_{t+1} = \mu(z_t, a_t) \tag{4}$$

Loss Function: We align the estimated $\tilde{z}_{t+1}$ from the state transition model with the actual latent state z_{t+1} at time $t+1$, while ensuring distinctiveness from other latent states z_{t+n} at different times $t+n$. Drawing inspiration from [25, 26], we formulate a latent representation loss function utilizing a triplet loss, denoted as $\mathcal{L}_{\text{trip}}$.

$$\mathcal{L}_{\text{trip}}\left(z_{t+1}, \tilde{z}_{t+1}, z_{t+n}\right) = \max\left(\|z_{t+1} - \tilde{z}_{t+1}\|_2^2 - \|z_{t+1} - z_{t+n}\|_2^2 + m, 0\right), \quad \text{s.t.} \quad n \neq 1 \tag{5}$$

where m is a margin that is enforced between $\tilde{z}_{t+1}$ and z_{t+n} pairs.

This updating strategy empowers the encoder to comprehend the dynamics of environmental state transitions and extract environmental attributes by assimilating state-action pairs derived from the MDP rollout.

4 Experiments

4.1 Ablation Study for the Privileged Learning Encoder

To evaluate the proposed algorithm against traditional privileged learning algorithms, an ablation study was conducted on Implicit and Explicit privileged learning methods, as well as the SLR method. Commonly used privileged information $e_t \in \mathbb{R}^{10}$ were chosen, including parameters such as friction, restitution, foot height, and foot contact. During training, the former two parameters were utilized for domain randomizations to enhance model generalization, while the latter two served as external robot perceptions aiding decision-making. The above two privileged learning methods are as follows:

Implicit: We adapted the teacher algorithm from [9] to encode privileged information into latent $l_t = h(e_t)$, which was integrated into the policy. An adaptation module then estimated the privileged latent from observation history $\tilde{l}_t = \phi(o_t^H)$ and aligned it with the real privileged latent generated by the encoder.

Explicit: The 10-dimensional privileged information e_t was directly estimated as $\tilde{e}_t = \phi(o_t^H)$ and then integrated into the policy.

To ensure the fairness of quantitative comparisons, the configurations used in the experiments, including training hyperparameters, reward function, and other settings, are all based on the default settings in the code repository [27]. Training was conducted for 5000 iterations under multiple terrains by default, and the results are presented in Figure 3.

Figure 3 demonstrates the significant superiority of the proposed algorithm over traditional privileged learning methods. This improvement is attributed to the self-learning encoding mechanism, enabling the robot to discern between various terrains. While certain privileged information, such as friction and restitution used for domain randomization, remains consistent across different terrains, others like foot height and foot contact offer limited terrain differentiation.

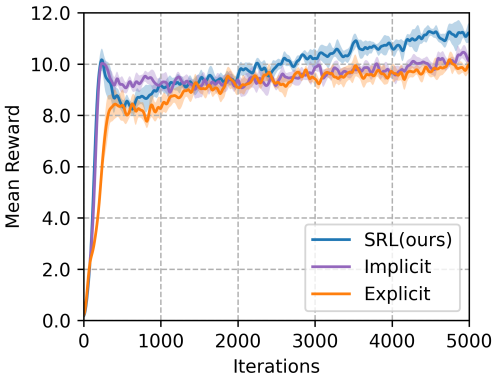

Figure 3: Ablation study training curves, curves are averaged over 3 seeds. The shaded area represents the standard deviation across seeds.

4.2 Latent Representation Analysis

Identifying and distinguishing various terrain types is essential for robotic systems [28]. To assess the efficacy of the self-learned latent representation in this regard, a simulation-based latent representation analysis is conducted. A test environment is designed composed of four sequential terrains: an upward slope, descending stairs, flat ground, and ascending stairs. The trained robot is tasked with navigating these terrains sequentially (Figure 4). During traversal, the latent representation output is recorded at each step. Subsequently, t-distributed stochastic neighbor embedding (t-SNE) analysis is applied to project the complex latent representations of the entire trajectory into a two-dimensional space for analysis. The latent representations from Implicit and SRL are analyzed separately, as depicted in Figure 5.

Different Terrain Representation. As shown in Figure 5. The self-learned latent representations reveal distinct ring-shaped regions (labeled A, B, C, D) for each terrain. Notably, regions A and D exhibit similarity, suggesting the robot perceives up slope and ascending stairs as similar processes. In contrast, representations trained with the Implicit method present scattered points, indicative of overlapping privileged information across terrains.

Terrain Transition Representation. Notably, in the SLR latent representations, when the robot transitions from one terrain to another, all four ring-shaped latent representations show "tails". For instance, when transitioning from an up slope to descending stairs, the latent representation in region A has a tails extending to the right, with the tail's end having a color similar to the lower left corner of region B. This indicates that the robot is near the boundary between terrains at that moment, and our latent representations are effectively indicating such terrain transitions.

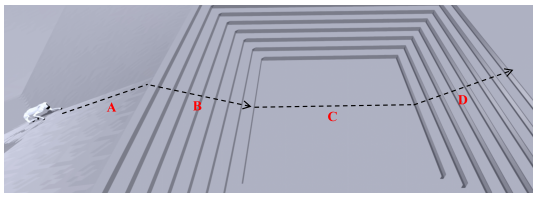

Figure 4: t-SNE test terrain.

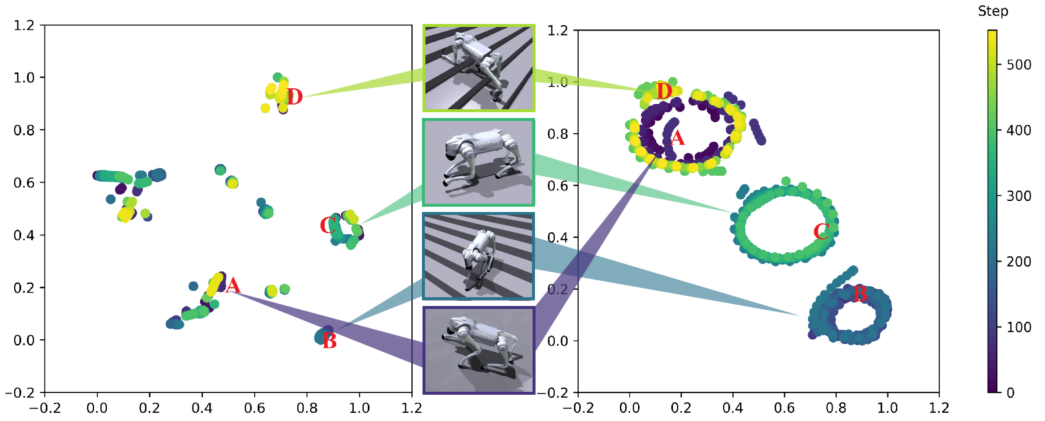

Figure 5: t-SNE visualization of Implicit (left) and SLR (right). Color intensity represents cumulative steps across four terrains. The privileged latent distribution is discrete and weakly correlated with terrain. In contrast, the SLR latent trajectories align precisely with the terrains traversed by the robot, with each ring-like representation accompanied by a "tail", indicating terrain transitions.

5 Results

5.1 Compared Methods

Previous comparative studies of robot algorithms often lack fairness due to the absence of a universal benchmark in robotics. Comparing algorithms within one's own environment is not objective, as reward functions and hyperparameters are typically optimized for the researcher's algorithm. Additionally, replicating previous algorithms in a new code repository can lead to incomplete reproductions, potentially affecting their performance.

In this study, these issues are addressed by directly implementing the SLR algorithm in the open-source code repositories of previous state-of-the-art (SOTA) works [9, 11, 13, 27]. Only the algorithm framework is modified while keeping other variables consistent with the original settings. This approach ensures a fair comparison between our method and the publicly available SOTA algorithms. The algorithms which are evaluated as given below:

- **MoB**: Explicit estimation, trained on flat ground with the Unitree Go1 robot.
- **RLvRL**: Implicit estimation, trained on flat ground with the Cheetah robot.
- **ROA**: Implicit estimation, trained on custom fractal noise environment using a Unitree Go1 robot equipped with a manipulator.
- **Baseline**: No encoder, no privileged information, trained in multi-terrain environments with the Unitree A1 robot.

5.2 Simulation

All four code repositories implemented RL based on the PPO algorithm, with training conducted on the IsaacGym platform [27, 29], utilizing three different random seeds. Training curves are illustrated in Figure 6. Evaluation metrics include linear velocity tracking error (LVTE) and angular velocity tracking error (AVTE), assessed using Mean Square Error (MSE). Target linear velocity ranges from -1.0 to 1.0 m/s, and target angular velocity ranges from -1.0 to 1.0 rad/s. Evaluation results are summarized in Table 1.

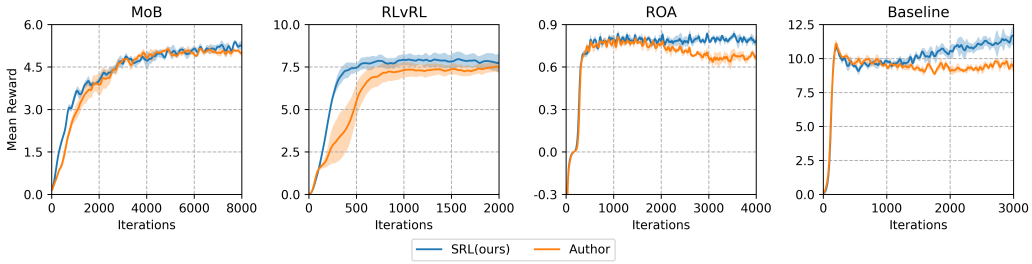

Figure 6: Training curves from various repositories. The SRL's mean reward surpasses those of Baseline, RLvRL, and ROA, and is marginally higher than MoB. The shaded area represents the standard deviation across seeds.

Metrics	Method	Code Repository			
		MoB	RLvRL	ROA	Baseline
LVTE	Author	**0.078**	**0.051**	0.150	0.041
	Ours	0.115	0.080	**0.136**	**0.033**
AVTE	Author	0.130	0.229	1.100	0.053
	Ours	**0.041**	**0.125**	**1.096**	**0.045**

Table 1: Velocity tracking errors in various environments. The SRL algorithm demonstrates superior velocity tracking capabilities compared to the authors' algorithm in most cases. Top performances are highlighted in bold.

Based on the evaluation results of the four code repositories, the proposed SLR algorithm generally achieves higher mean rewards and superior velocity tracking capabilities compared to the original implementations. This indicates that this approach can develop more effective policies without relying on privileged information. Additionally, the experiments with the MoB and ROA repositories demonstrate that the SLR algorithm is robust and generalizable, capable of handling complex tasks involving multiple commands and manipulators. Consequently, it can be inferred that enabling robots to autonomously learn the latent representation yields greater benefits than manually selecting privileged information.

5.3 Deploy in Real-World

The trained policies are deployed on the Unitree Go2 robot in real-world as depicted in Figure 7. Performance evaluation of the policy was conducted across various indoor and outdoor terrains, and comparative analyses were performed against above code repositories and Unitree built-in MPC control method. Each environment was tested ten times. As presented in Table 2, the results demonstrate the superior efficacy of our policy in real-world scenarios.

6 Discussion and Limitations

In this study, we introduce a quadruped locomotion algorithm that operates without relying on privileged information. Unlike previous open-source repositories, our proposed algorithm surpasses the author's algorithm in performance retaining original authors' configuration parameters. Through real-world deployment, our robot demonstrates the capability to navigate through diverse and challenging terrains.

While our blind policy demonstrates robust motion, achieving superior trajectory planning necessitates the use of vision sensors. Moving forward, we aim to enhance quadruped locomotion by integrating visual information for tackling even more complex challenges.

(a) Indoor Scenarios (b) Outdoor Scenarios

Figure 7: Indoor and outdoor experiment settings. The left column (a) showcases indoor scenarios, with the top row depicting the robot navigating a step obstacle and the bottom row featuring the robot on a deformable surface. The right column (b) illustrates outdoor scenarios, depicting the robot moving through a bush, on a cobbled road, and climbing a curb in the top row, while the bottom row showcases the robot encountering a slope, stairs, and a rock.

Scenarios	Terrain	Ours	MoB	RLvRL	Baseline	MPC
Indoor	Step	**10**	4	2	1	**10**
	Deformable Slope	**10**	3	1	0	8
Outdoor	Bush	**10**	9	7	5	**10**
	Cobbled Road	**10**	**10**	4	3	9
	Curb	**10**	2	1	0	4
	Earthen Slope	**10**	4	2	0	6
	Stairs	**10**	0	0	0	2
	Rock	**7**	0	0	0	0

Table 2: Comparison of different deployment strategies across various terrains. Each strategy was evaluated ten times per terrain, with the values in the table indicating the number of successful trials. Top performances are highlighted in bold.

References

[1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252. IEEE, 2018.

[2] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter. Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. *IEEE Robotics and Automation Letters*, 3(3):2261–2268, 2018.

[3] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154–1171, 2021.

[4] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, et al. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525. IEEE, 2020.

[5] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter. Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot. In *Field and Service Robotics: Results of the 12th International Conference*, pages 247–260. Springer, 2021.

[6] G. Ji, J. Mun, H. Kim, and J. Hwangbo. Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robotics and Automation Letters*, 7(2):4630–4637, 2022.

[7] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017.

[8] I. M. A. Nahrendra, B. Yu, and H. Myung. Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5078–5084. IEEE, 2023.

[9] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[10] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. 2021.

[11] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: Learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[12] G. B. Margolis, X. Fu, Y. Ji, and P. Agrawal. Learning to see physical properties with active sensing motor policies. *Conference on Robot Learning*, 2023.

[13] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[14] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[15] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[16] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, pages 403–415. PMLR, 2023.

[17] T. He, C. Zhang, W. Xiao, G. He, C. Liu, and G. Shi. Agile but safe: Learning collision-free high-speed legged locomotion. *arXiv preprint arXiv:2401.17583*, 2024.

[18] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.

[19] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[20] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[21] J. Wu, G. Xin, C. Qi, and Y. Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 2023.

[22] 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.

[23] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. *arXiv preprint arXiv:2111.01674*, 2021.

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

[25] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[26] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

[27] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[28] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone. Sterling: Self-supervised terrain representation learning from unconstrained robot experience. In *7th Annual Conference on Robot Learning*, 2023.

[29] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

A Appendix

A.1 Reward Terms Detail

In Table A1, v is the linear velocity, σ is the tracking shaping scale equal to 0.25 here, h^{target} is the desired base height corresponding to ground, p_z^{target} and p_z^i are the desired feet position and real feet position in z-axis of robot's frame and v_{xy}^i is the feet velocity in xy-plane of robot's frame.

Reward	Equation	Weight
Powers	$\lvert\tau\rVert\dot{\theta}\rvert^T$	-2e-5
Linear velocity tracking	$\exp\left\{-\frac{\lVert v_{xy}^{\text{cmd}}-v_{xy}\rVert_2^2}{\sigma}\right\}$	1.0
Angular velocity tracking	$\exp\left\{-\frac{\left(\omega_{\text{yaw}}^{\text{cmd}}-\omega_{\text{yaw}}\right)^2}{\sigma}\right\}$	0.5
Linear velocity penalty in z-axis	v_z^2	-2.0
Angular velocity penalty	$\lVert\omega_{xy}\rVert_2^2$	-0.05
Joint acceleration penalty	$-\lVert\ddot{\theta}\rVert^2$	-2.5e-7
Base Height penalty	$\left(h^{\text{target}}-h\right)^2$	-10.0
Joint torques	$-\lVert\tau\rVert^2$	1
Action rate	$\lVert a_t-a_{t-1}\rVert_2^2$	-0.01
Action smoothness	$\lVert a_t-2a_{t-1}+a_{t-2}\rVert_2^2$	-0.01
Foot clearance	$\sum_{i=0}^{3}\left(p_z^{\text{target}}-p_z^i\right)^2\cdot v_{xy}^i$	-0.01
Orientation	$\lVert g\rVert_2^2$	-0.2

Table A1: Reward Terms

A.2 Domain Randomizations

Parameters	Range[Min,Max]	Unit
Body Mass	[0.8,1.2]×nominal value	Kg
Link Mass	[0.8,1.2]×nominal value	Kg
CoM	[-0.1,0.1]×[-0.1,0.1]×[-0.1,0.1]	m
Payload Mass	[-1,3]	Kg
Ground Friction	[0.2,2.75]	-
Ground Restitution	[0.0,1.0]	-
Motor Strength	[0.8,1.2]×motor torque	Nm
Joint K_p	[0.8,1.2]×20	-
Joint K_d	[0.8,1.2]×0.5	-
Initial Joint Positions	[0.5,1.5]×nominal value	rad
System Delay	[0,3Δ_t]	s
External Force	[-30,30]×[-30,30]×[-30,30]	N

Table A2: Domain Randomizations and their Respective Range